

IBM® DB2® Universal Database



릴리스 정보

버전 7.2/버전 7.1 *FixPak 3*

IBM® DB2® Universal Database



릴리스 정보

버전 7.2/버전 7.1 *FixPak 3*

목차

DB2 Universal Database 7의 사용을 환영합니다! xiii

제1부 특별 주의사항 1

특별 주의사항 3

DB2 UDB 버전 7의 액세스 가능성 기능. 3

키보드 입력 및 탐색. 3

액세스 가능한 표시에 대한 기능. 3

대체 정보 신호 4

지원 기술과의 호환성 4

액세스 가능한 문서 4

추가 필수 Solaris 패치 레벨 4

Solaris용 DB2 버전 7에서 지원되는 CPU 4

파티션 데이터베이스에 노드 추가 시 문제점 4

이주 중 오류 7

Red Flag Linux에서 중국어 로케일 수정. 7

이동 가능한 드라이브가 접속되지 않은 경우 DB2 설

치 정지 7

일본어 및 중국어 Linux 환경에서 Linux용 DB2에

대한 추가 로케일 설정 7

Microsoft Internet Explorer의 제어 센터 문제점 7

Windows 환경에서의 Information Catalog

Manager와 Sybase 사이의 비호환 8

제어 센터 기능 손실. 9

DB2 UDB가 포함되어 있지 않은 Netscape CD 9

XML Readme 파일 오류 9

S/390용 Linux에서의 데이터 손실 가능성 9

Windows 2000에서의 DB2 UDB. 10

온라인 문서(HTML, PDF 및 검색) 11

Windows 2000 운영 체제에서 지원되는 웹 브라우

저 11

Solaris에서 DB2 온라인 정보 검색 11

TCP/IP를 사용하도록 OS/2용 NetQuestion 전환. 12

Netscape 시작 시 발생하는 오류 메시지 12

Unix 기반 시스템에서 Adobe Acrobat Reader에

대한 구성 요구사항. 13

하나의 PDF 파일에 제공되는 SQL 참조서. 13

제2부 설치 및 구성 15

일반 설치 정보 17

지원되는 모든 DB2 클라이언트의 설치 패키지 다운

로드. 17

Windows 2000에 DB2 설치 17

특수 레지스터로 정의된 뷰와 관련된 이주 발행 17

Windows 2000에서의 IPX/SPX 프로토콜 지원 17

DB2의 이전 버전을 업그레이드하기 전에 DB2 프로

세스 중지. 18

다른 DB2 제품이 이미 설치된 경우의 DB2 설치 이

후 db2iupdt 수행 18

DB2 제어 센터를 수행할 Linux 환경 설정. 18

S/390 상의 Linux용 DB2 Universal Database

Enterprise Edition 및 DB2 Connect Enterprise

Edition. 19

UNIX용 DB2 Universal Database Enterprise -

Extended Edition 빠른 시작 19

HP-UX용 shmseg 커널 매개변수. 20

IBM Visual Warehouse 제어 데이터베이스 이주

웨어하우스 제어 데이터베이스에 액세스. 20

Data Links Manager 빠른 시작. 21

메시지 "접두부에 대한 afsfid 확보 시 오류 발생"으

로 Dlfm 시작이 실패함 21

아카이브 파일에 대한 Tivoli Storage Manager 클

래스 설정. 21

DFS Client Enabler에 대한 디스크 공간 요구사항

21

AIX에서 데이터 링크 파일 관리 프로그램 백엔드 프

로세스 모니터 21

AIX용 DB2 Data Links Manager 설치 및 구성:

DCE-DFS 환경에서의 추가 설치 고려사항 22

실패한 "dlfm add_prefix" 명령 22

AIX용 DB2 Data Links Manager 설치 및 구성:

db2setup 유틸리티를 사용하여 AIX에 DB2 Data

Links Manager 설치 23

AIX용 DB2 Data Links Manager 설치 및 구성:

DCE-DFS 사후 설치 타스크 23

AIX용 DB2 Data Links Manager 설치 및 구성:

Smit를 사용하여 수동으로 DB2 Data Links

Manager 설치 24

DB2 Data Links DFS Client Enabler 설치 및 구

성 24

Solaris용 DB2 Data Links Manager 설치 및 구성

AIX에서 DB2 Data Links Manager에 대한 백업 메소드 선택	24
Solaris 운영 환경에서 DB2 Data Links Manager용 백업 방법 선택	26
Windows NT에서 DB2 Data Links Manager에 대 한 백업 메소드 선택	27
AIX에서 저널 파일 시스템 백업	29
Windows NT에서 데이터 링크의 관리자 그룹 특권 Data Links File System Filter(DLFF) 설치에 대한 로깅 최소화	30
설치 후 메시지 로깅	31
SMIT May Remove Additional Filesets를 사용하 여 DLFM 구성요소 설치 제거	32
호스트 이름을 시작/판별하기 전에	32
Data Links File Manager 작업: DB2 데이터베이스 에서 DB2 Data Links Manager 삭제 후 정리	33
DLFM1001E (새 오류 메시지).	34
DLFM 설정 구성 파일 옵션.	34
AIX에서 Data Links/DFS Script dmapp_prestart 수행 시 오류.	35
Tivoli Space Manager와 Data Links 통합	35
제한사항 및 한계	36
AIX용 DB2 Data Links Manager 설치 및 구성 공통적인 설치 고려사항	37
설치 및 구성 보충 설명서	39
UNIX 운영 체제에 DB2 클라이언트 설치	39
HP-UX 커널 구성 매개변수.	39
사용자 자신의 응용프로그램 수행	39
Run-time Client를 사용한 데이터베이스 유틸리티 바인딩	39
ODBC를 사용한 DB2에 대한 UNIX 클라이언트 액세스	39
여러 데이터 소스에 액세스하기 위해 연합 시스템 설 정	39
연합 시스템	39
DB2 Relational Connect 설치.	40
Oracle 데이터 소스에 액세스	43
문서 오류.	43
Sybase 데이터 소스에 액세스(새 장).	44
Sybase 데이터 소스를 연합 서버에 추가.	44
Sybase 코드 페이지 지정.	51
ODBC를 사용하여 Microsoft SQL Server 데이터 소스에 액세스(새 장)	51
연합 서버에 Microsoft SQL Server 데이터 소스 추가.	52
Microsoft SQL Server 코드 페이지 검토	57

제3부 관리	59
관리 안내서: 계획	63
물리적 데이터베이스 설계.	63
파티션 키	63
노드 그룹 설계	63
분산 데이터베이스 설계	63
다중 데이터베이스 갱신	63
Windows NT 환경의 고가용성.	64
DB2MSCS 유틸리티 수행 전에 머신 재부트	64
Sun Cluster 2.2 상의 DB2와 고가용성.	64
Solaris에서 Veritas 지원.	64
이름 지정 규칙	66
8자를 초과하는 사용자 ID 및 스키마 이름에 대 한 주의사항	66
사용자 ID 및 암호.	66
릴리스 간의 비호환성	66
Norton's Utilities와 Windows NT DLFS 비호 환	66
SET CONSTRAINTS를 SET INTEGRITY로 교체.	67
자국어 지원	67
DB2 버전 7 자국어 버전.	67
DAS(DB2 Administrator Server)에 대한 로케일 설정.	71
DB2 UDB는 Windows 플랫폼에서 Baltic Rim 코드 페이지(MS-1257)를 지원합니다.	71
코드 페이지 값 추론	71
국가 코드 및 코드 페이지 지원.	71
문자 세트.	71
관리 안내서: 구현	73
DMS 컨테이너 추가 또는 확장(새로운 프로세스).	73
GUI 도구를 사용하여 DB2 관리	73
데이터베이스 작성	74
테이블 공간 작성	74
순차 작성	75
IDENTITY 컬럼과 순차 비교	77
색인, 색인 확장 또는 색인 스펙 작성	78
데이터베이스 변경	78
파티션의 SMS 테이블 공간에 컨테이너 추가	78
식별 컬럼 변경	78
순차 변경	79
순차 삭제	79
테이블 공간의 상태 전환	80
DMS 테이블 공간의 컨테이너 수정	80
데이터베이스 액세스 제어	80

순차 특권	80	정렬 힙 임계값(sheapthres).	114
데이터 암호화	81	레벨 자동 업그레이드 전의 잠금 목록 최대 퍼센트(maxlocks)	114
데이터베이스 복구	82	DB2/DB2 Data Links Manager/Data Links 액세스 토큰 만료 간격 (dl_expint) 구성	115
중지된 입출력 사용 방법	82	MIN_DEC_DIV_3 데이터베이스 구성 매개변수	115
중분 백업 및 복구	84	응용프로그램 제어 힙 크기(app_ctl_heap_sz)	116
병렬 복구	86	데이터베이스 시스템 모니터 힙(heap) 크기 (mon_heap_sz)	117
Named Pipe에 백업	87	실행 중인 응용프로그램의 최대 수 (maxappls)	117
분리 이미지로부터 백업	87	복구 범위 및 소프트 점점점 간격(softmax)	117
요구에 따른 로그 아카이브	90	수정된 페이지 추적 사용 (trackmod)	117
로그 미러링	90	데이터베이스 로그 경로 변경 (newlogpath)	117
Sun Solaris와 HP에 대한 교차 플랫폼 백업 및 복원 지원	91	로그 파일의 위치 (logpath)	119
DB2 Data Links Manager 고려사항/백업 유틸리티 고려사항	91	잠금 목록용 최대 저장영역 (locklist)	119
DB2 Data Links Manager 고려사항/복원 및 롤포워드 유틸리티 고려사항.	91	DB2 레지스트리 및 환경 변수	119
롤 포워드하지 않고 오프라인 백업으로부터 데이터베이스 복원	93	새로운 레지스트리 변수 및 변경된 레지스트리 변수의 표	119
데이터베이스와 테이블 공간 복원 및 로그 끝까지 롤포워드	94	SQL Explain 도구	124
DB2 Data Links Manager 및 복구 상호작용.	94	Administering Satellites 안내 및 참조서	125
조정이 필요한 상황 감지	98	버전 7.2 DB2 Personal Edition 및 DB2 Workgroup Edition을 위성으로 설정	125
데이터베이스 복구를 위한 User Exit	98	전제조건	125
다중 데이터베이스 파티션 서버로 명령 발행	99	동기화하기 위해 버전 7.2 시스템 구성	128
고속 노드간 통신	100	버전 6 Enterprise Edition 시스템에 FixPak 2 이상 설치	129
VI를 사용하여 수행할 수 있는 DB2	100	버전 6 제어 센터 및 위성 관리 센터 업그레이드	137
관리 안내서: 성능.	103	Command Reference	139
응용프로그램 고려사항	103	db2batch - 벤치마크 도구	139
분리 레벨 지정.	103	db2cap(새 명령)	139
최적화 클래스 조정	103	db2cap - CLI/ODBC 정적 패키지 바인딩 도구	140
동적 복합 텍스트 명령문	104	db2ckrst(새 명령).	141
환경 고려사항	104	db2ckrst - 중분 복원 이미지 순차 점점	142
큰 색인 키 사용	105	db2gncol(새 명령)	143
시스템 카탈로그 통계	105	db2gncol - 생성된 컬럼 값 갱신.	144
분산 통계 수집 및 사용.	105	db2inidb - 미러된 데이터베이스 초기화	145
카탈로그 통계 갱신 규칙	106	db2look - DB2 통계 추출 도구.	146
부속 구성요소 통계	106	db2updv7 - 버전 7 현재 수정 레벨로 데이터베이스 갱신	146
SQL 컴파일러 이해	109	새 명령행 처리기 옵션(-x, 컬럼 표제의 인쇄 억제)	148
복제된 요약 테이블	109	DB2 CLP에 대한 트루 타입 글꼴 요구사항	148
데이터 액세스 개념 및 최적화.	110	ADD DATALINKS MANAGER	148
조작 성능	111	ARCHIVE LOG (새 명령)	148
데이터베이스 버퍼 풀 관리.	111	Archive Log	149
다중 데이터베이스 버퍼 풀 관리	112	BACKUP DATABASE	151
조정자(governor) 사용	113		
DB2 구성	113		
정렬 힙 크기(sortheap)	113		

	구문 도표	151		DATALINK 복제.	175
	DB2 Data Links Manager 고려사항	151		LOB 제한사항.	176
	BIND.	152		복제 계획	176
	CALL	153		복제 환경 설정.	176
	DROP DATALINKS MANAGER (새 명령)	153		update-anywhere 전제조건.	176
	DROP DATALINKS MANAGER.	154		복제 환경 설정.	176
	EXPORT	158		문제점 판별.	177
	GET DATABASE CONFIGURATION	158		AS/400용 Capture 및 Apply.	178
	GET ROUTINE (새 명령)	159		OS/390용 Capture 및 Apply.	179
	GET ROUTINE	160		OS/390용 DB2 DataPropagator의 전제조건	179
	GET SNAPSHOT	160		OS/390에서의 유니코드 및 ASCII 코드화 체계	179
	IMPORT	161		UNIX 플랫폼용 Capture 및 Apply.	181
	LIST HISTORY	161		UNIX 및 Windows용 Capture 및 Apply에 대	
	LOAD	162		한 환경 변수 설정.	181
	PING (새 명령)	163		테이블 구조.	181
	PING.	164		Capture 및 Apply 메시지	181
	PUT ROUTINE (새 명령)	165		응용프로그램 내에서 Capture 및 Apply 프로그램	
	PUT ROUTINE	166		시작	182
	RECONCILE	166		시스템 모니터 안내 및 참조서.	183
	REORGANIZE TABLE	167		db2ConvMonStream.	183
	RESTORE DATABASE	168		문제점 해결 안내서	185
	구문	168		사용자가 로그인되지 않았을 때 Windows 95,	
	DB2 Data Links Manager 고려사항	169		Windows 98 및 Windows ME에서 DB2 시작.	185
	ROLLFORWARD DATABASE	169		DB2 Universal Database Server 문제점 해결	189
	CLP 리턴 코드의 문서 오류	169			
	데이터 이동 유틸리티 안내 및 참조서	171		64 비트 플랫폼에서 DB2 Universal Database 사	
	가져오기	171		용	191
	버퍼된 삽입이 있는 가져오기 사용	171		구성	191
	로드	171		LOCKLIST.	191
	로드 조작 이후 보류 상태	171		shmsys:shminfo_shmmax	191
	로드 제한사항 및 한계	171		제한사항	191
	totalfreespace 파일 유형 수정자	171		XML Extender 관리 및 프로그래밍	193
	자동 로드 프로그램	172		MQSeries	195
	Authentication=YES일 경우 자동 로드 프로그램			DB2 MQSeries 함수 설치 및 구성.	195
	램 수행에 rexecd 필요	172		MQSeries 설치	195
	복제 안내 및 참조서.	173		MQSeries AMI 설치	196
	복제 및 IBM이 아닌 서버.	173		DB2 MQSeries 함수 사용.	196
	Windows 2000에서의 복제	173		MQSeries 메시지 스타일	197
	SQL 파일 저장 시 알려진 오류	173		메시지 구조.	197
	DB2 유지보수.	173		MQSeries 기능 개요.	198
	웹의 Data Difference 유틸리티	173		제한사항	199
	데이터 복제 시나리오	174		오류 코드	199
	복제 시나리오	174		사용 시나리오	199
	복제 계획	175		기본 메시지.	200
	테이블 및 컬럼 이름.	175			

	메시지 전송	201
	메시지 검색	202
	응용프로그램간 연결성	204
	enable_MQFunctions	208
	enable_MQFunctions	209
	disable_MQFunctions	210
	disable_MQFunctions	211

제4부 관리 도구 213

	제어 센터	215
	VSE 및 VM 서버용 DB2 서버 관리 능력	215
	제어 센터에 대한 Java 1.2 지원	215
	Windows 운영 체제에서 온라인 도움말 사용 시 "유효하지 않은 단축 아이콘" 오류	216
	OS/2의 Java 제어 센터	216
	Windows 운영 체제의 저널에서 완료된 작업을 보려고 할 때 "파일 액세스 거부" 오류	216
	멀티사이트 갱신 테스트 연결	216
	OS/390용 DB2 제어 센터	217
	OS/390에 대한 제어 센터의 필수 수정	219
	공간 계층 작성 대화 상자로 변경	219
	DB2 제어 센터에 대한 문제점 해결 정보	219
	UNIX 기반 시스템에서 제어 센터 문제점 해결	219
	OS/2에서 가능한 정보 팝업 상자 문제점	220
	jdk11_path 구성 매개변수에 대한 도움말	220
	스크립트 센터 또는 저널 사용 시 Solaris 시스템 오류(SQL10012N)	220
	DPREPL.DFT 파일에 대한 도움말	220
	하나 이상의 제어 센터 애플릿 시작	221
	애플릿으로 수행되는 제어 센터에 대한 온라인 도움말	221
	애플릿 모드에서 제어 센터 수행(Windows 95)	221
	대형 조회 결과로 작업	221
	정보 센터	223
	Windows 운영 체제에서 "유효하지 않은 단축 아이콘" 오류	223
	Netscape가 이미 열려 있을 때 Netscape Navigator에서 외부 웹 링크 열기(Unix 기반 시스템).	223
	정보 센터 시작 문제점	223
	마법사	225
	데이터베이스 작성 마법사에서 Extent 크기 설정	225
	MQSeries 지원 마법사	225
	OLE DB 지원 마법사	225

제5부 비즈니스 인텔리전스 227

	비즈니스 인텔리전스 자습서	231
	개정된 비즈니스 인텔리전스 자습서	231
	Data Warehouse Center 관리 안내서	233
	문제점 해결	233
	Excel을 웨어하우스 소스로 설정	233
	프로세스 정의 및 수행	233
	메타데이터 내보내기 대화 상자	233
	OS/390 JCL 작업열 제출 프로그램의 값 정의	234
	데이터 웨어하우스 샘플 부록 변경사항	234
	Data Warehouse Center 메시지	234
	DB2 OLAP 통합 서버에서 윤곽 작성 및 데이터 로드	234
	Data Warehouse Center에서 Classic Connect 사용	234
	Data Warehouse Center 환경 구조	235
	역변환기 사용	235
	DB2 버전 7 웨어하우스 에이전트를 사용하여 DB2 버전 5 데이터에 액세스	236
	DB2 버전 5 서버 이주	236
	에이전트 구성 변경	236
	IBM ERwin 메타데이터 추출 프로그램	238
	목차	238
	소프트웨어 요구사항	239
	프로그램 파일	239
	태그 언어 파일 작성	239
	태그 언어 파일을 Data Warehouse Center로 가져오기	241
	태그 언어 파일을 Information Catalog Manager로 가져오기	242
	문제점 해결	243
	ERwin을 DB2 Data Warehouse Center로 맵핑	245
	Data Warehouse Center에서 이름 및 주소 정리	247
	Trillium Batch System JCL 파일 작성	252
	UNIX 및 Windows에서 Trillium Batch System 스크립트 파일 작성	253
	Trillium Batch System 단계 정의	253
	Trillium Batch System 사용자 정의 프로그램 사용	253
	오류 처리	254
	MQSeries와 Data Warehouse Center 통합	256
	MQSeries 메시지용 뷰 작성	256
	MQSeries 메시지 및 XML 메타데이터 가져오기	257

Microsoft OLE DB 및 Data Transaction Services 지원	260
OLE DB 테이블 함수용 뷰 작성	261
DTS 패키지용 뷰 작성	261
바꾸기와 함께 증가 확약 사용.	262
구성요소 추적 데이터 파일 이름	262
AIX 및 Solaris 운영 환경에서 Sybase 소스에 대해 필요한 Open Client	262
정정된 샘플 항목	263
웨어하우스 소스 설정.	263
Microsoft Access의 Memo 필드를 웨어하우스 소스에 맵핑.	263
웨어하우스 데이터베이스 유지보수	263
DB2 UDB RUNSTATS 프로그램에 대한 단계 부속 유형에 테이블 링크	263
기본 웨어하우스 제어 데이터베이스	264
웨어하우스 제어 데이터베이스 관리 창	264
활동 중인 웨어하우스 제어 데이터베이스 변경	264
웨어하우스 제어 데이터베이스 작성 및 초기화	265
편집된 SQL 단계 작성	266
프로세스 모델 편집기 창에서 소스 및 목표 변경	266
Data Warehouse Center 오브젝트에 설명 추가 샘플 내용 수행.	266
Create DDL SQL문 편집	267
Visual Warehouse 비즈니스 뷰 이주	267
목표 테이블 및 기본 키 생성	267
Merant ODBC 드라이버 사용	267
새로운 ODBC 드라이버.	268
OS/2 데이터베이스에서 웨어하우스 소스 또는 목표 정의	268
웨어하우스 제어 데이터베이스의 상태 모니터.	268
TBC_MD 샘플 데이터베이스에서 SQL 지원 사용	268
FormatDate 함수 사용	268
언어 설정 변경.	269
키 테이블 생성 변환기 사용	269
데이터베이스에 연결 유지보수.	269
원격 Data Warehouse Center 클라이언트 설정	270
VM용 DB2 웨어하우스 소스 정의	270
VM용 DB2 또는 VSE용 DB2 목표 테이블 정의	270
분리 식별자 지원 사용	270
Data Joiner가 바인드 문제점을 지시합니다	270
Data Warehouse Center를 사용한 복제 설정 및 수행	271
문제점 해결 추가 정보	274
소스 및 목표에 액세스	275

지원되는 IBM 이외 데이터베이스 소스에 대한 추가 사항.	276
Data Warehouse Center에 수동으로 데이터 소스 작성	277
CWMI(Common Warehouse Metadata Interchange)를 사용하여 메타데이터 가져오기 및 내보내기	278
소개	278
메타데이터 가져오기	278
가져오기 유틸리티 수행 후 메타데이터 갱신	280
메타데이터 내보내기	280
OS/390 Runstats 유틸리티 단계	282
OS/390 로드 유틸리티 단계	283
CWM(Common Warehouse Metamodel) XML 지원	283
프로세스 모델 편집기.	284
스키마 모델 편집기	284
필수 필드	284
Data Warehouse Center 런치패드 향상	284
단계 정보를 파일로 인쇄	285
Data Warehouse Center 응용프로그램 통합 안내서	287
추가 메타데이터 템플릿	287
Commit.tag	287
ForeignKey.tag	288
ForeignKeyAdditional.tag	290
PrimaryKey.tag	292
PrimaryKeyAdditional.tag	293
Data Warehouse Center 온라인 도움말.	297
복제를 위한 테이블 또는 뷰 정의	297
AS/400 에이전트를 사용하여 Essbase VWP 수행	297
Data Warehouse Center 발행 창 및 관련 등록 정보 창 사용	297
외부 키	297
복제 노트북.	297
태그 언어 가져오기	297
데이터 추가를 위한 링크	298
테이블 가져오기	298
RUNSTATS 및 REORGANIZE TABLE 온라인 도움말에 대한 정정	298
통지 페이지(웨어하우스 등록정보 노트북 및 스케줄 노트북)	298
에이전트 사이트 노트북의 에이전트 모듈 필드	298
DB2 OLAP Starter Kit	299

OLAP 서버 웹 사이트	299
지원되는 운영 체제 서비스 레벨	299
UNIX에서 DB2 OLAP Starter Kit 설정 완료	299
OLAP Starter Kit용 ODBC 구성	300
UNIX 시스템에서 데이터 소스 구성.	301
UNIX 시스템에서 OLAP 메타데이터 카탈로그	
구성	302
Windows 시스템에서 데이터 소스 구성	303
Windows 시스템에서 OLAP 메타데이터 카탈로	
그 구성	304
데이터 소스를 구성한 후의 작업	305
OLAP Starter Kit Desktop에서 로그인	306
Starter Kit 로그인 예	307
OLAP Starter Kit에 대한 샘플 데이터베이스 수동	
작성 및 구성	307
응용프로그램을 OLAP Starter Kit 버전 7.2로 이	
주	309
알려진 문제점 및 제한사항.	309
OLAP Spreadsheet Add-in EQD 파일 누락	311
Information Catalog Manager 관리 안내서	313
Information Catalog Manager 초기화 유틸리티	313
사용권 문제.	313
설치 문제	313
DB2 버전 7 Information Catalog Manager를 사	
용하여 DB2 버전 5 정보 카탈로그에 액세스.	314
정보 카탈로그 설정	315
기타 제품과 메타데이터 교환	316
flgnxoln 명령을 사용하여 메타데이터 교환	316
MDISDGC 명령을 사용하여 메타데이터 교환	316
프로그램 호출	316
Information Catalog Manager Programming	
Guide and Reference	319
Information Catalog Manager 이유 코드.	319
Information Catalog Manager 사용자 안내서	321
Information Catalog Manager: 온라인 메시지	323
메시지 FLG0260E	323
메시지 FLG0051E	323
메시지 FLG0003E	323
메시지 FLG0372E	323
메시지 FLG0615E	323
Information Catalog Manager: 온라인 도움말	325
웹에 대한 Information Catalog Manager.	325

DB2 Warehouse Manager 설치 안내서.	327
웨어하우스 변환기에 대한 소프트웨어 요구사항	327
SAP R/3용 커넥터	327
설치 요구사항	327
웹용 커넥터.	327
설치 요구사항	327
Query Patroller 관리 안내서	329
DB2 Query Patroller 클라이언트는 별도의 구성요	
소임	329
dqpmigrate를 사용하여 DB2 Query Patroller로부	
터 이주	329
조회 관리 사용.	329
제어 테이블용 테이블 공간의 위치	330
dqpstart 명령에 사용할 수 있는 새로운 매개변수	330
iwmcmd 명령에 사용할 수 있는 새로운 매개변수	331
새 레지스트리 변수:	
DQP_RECOVERY_INTERVAL	331
조회 관리자 시작	331
사용자 관리.	331
작업 대기행렬 작성	332
명령행 인터페이스 사용	332
Query Enabler 참고.	332
DB2 Query Patroller 추적 프로그램이 공백 컬럼	
페이지를 리턴합니다	332
Query Patroller 및 복제 도구.	333
DB2 Query Patroller 문제점 해결	333
<hr/>	
제6부 응용프로그램 개발	335
Administrative API Reference	339
db2ArchiveLog (새 API)	339
db2ArchiveLog	340
db2ConvMonStream.	342
db2DatabasePing(새 API)	342
db2DatabasePing - Ping 데이터베이스.	343
db2HistData	345
db2HistoryOpenScan	346
db2XaGetInfo(새 API)	346
db2XaGetInfo - 자원 관리 프로그램에 대한 정	
보 얻기	347
db2XaListIndTrans(sqlxphqr을 대신하는 새 API)	348
db2XaListIndTrans - 2 단계 확약 중 이상 실	
패 트랜잭션 목록	349
db2GetSnapshot - 스냅샷 가져오기.	353
Forget 로그 레코드	354
sqlaintp - 오류 메시지 가져오기.	354

sqlbctcq - 테이블 공간 컨테이너 조회 단기	354	Linux Java 루틴에서 DECIMAL 유형 실패	378
sqlubkp - 데이터베이스 백업	354	반복적인 저장 프로시저에서 커서 사용	378
sqlureot - 테이블 재구성	355	OLE 자동 저장 프로시저어 작성.	379
sqlurestore - 데이터베이스 복원	355	복잡한 오브젝트 사용: 사용자 정의 구조화 유형	379
AIX Extended Shared Memory		컬럼에 구조화 유형 속성 삽입.	379
Support(EXTSHM)에 관한 문서 오류	355	대형 오브젝트(LOB) 사용	380
SQLFUPD	355	연합 데이터베이스 시스템의 대형 오브젝트	
locklist	355	(LOB) 지원.	380
SQLEDBDESC	356	시스템 조정.	382
SQLFUPD 문서 오류	356	DB2 프로그래밍 고려사항	382
응용프로그램 빌드 안내서	357	IBM DB2 OLE DB Provider	382
소개	357	C 및 C++ 프로그래밍	382
지원되는 소프트웨어	357	저장 프로시저어, 함수 및 메소드에 대한 C/C++	
샘플 프로그램	359	유형	383
DB2 응용프로그램 빌드에 대한 일반 정보	360	Java 프로그래밍	384
파일, Makefile 및 오류 확인 유틸리티 빌드	360	PARAMETER STYLE JAVA 프로시저어 및	
Java 애플릿 및 응용프로그램 빌드	360	함수에서의 Java 메소드 서명	384
환경 설정	360	JDBC 애플릿 서버에 연결.	384
SQL 프로시저어 빌드	363	샘플 프로그램	385
SQL 프로시저어 환경 설정.	363	CLI 안내서 및 참조서	387
컴파일러 환경 변수 설정	363	Run-time Client를 사용한 데이터베이스 유틸리티	
컴파일 명령 사용자 정의	364	바인딩.	387
중간 파일 보유.	368	CLI 응용프로그램에서 정적 SQL 사용.	387
백업 및 복원	369	JDBC/ODBC/CLI 정적 프로파일의 한계	387
SQL 프로시저어 작성	369	ADT 변환	388
저장 프로시저어 호출	370	고급 기능 사용.	389
컴파일된 SQL 프로시저어 분산	370	다중 스레드 응용프로그램 작성	389
HP-UX 응용프로그램 빌드.	371	화면 이동 커서.	389
HP-UX C	371	복합 SQL 사용	391
HP-UX C++	371	저장 프로시저어 사용	391
OS/2 응용프로그램 빌드	371	CLI/ODBC 구성 및 샘플 응용프로그램 수행.	392
OS/2용 VisualAge C++ 버전 4.0	371	구성 키워드.	392
PTX 응용프로그램 빌드.	372	DB2 CLI 함수	392
ptx/C++	372	SQLBindFileToParam - LOB 파일 참조를	
Solaris 응용프로그램 빌드	372	LOB 매개변수에 바인드.	392
SPARCompiler C++	372	SQLNextResult - 다음 결과 세트를 다른 명령	
Windows 32 비트 운영 체제용 응용프로그램 빌드	373	문 핸들과 연결.	392
VisualAge C++ 버전 4.0	373	Extended Scalar Functions	395
응용프로그램 개발 안내서	375	날짜 및 시간 함수	395
DB2 응용프로그램 코딩.	375	DB2 CLI/ODBC/JDBC 추적 기능 사용	395
Microsoft Visual C++용 IBM DB2 Universal		메시지 참조서	397
Database Project 및 Tool Add-in 활성화	375	메시지 및 SQLSTATE 도움말 보기.	397
Common DB2 응용프로그램 기법	375	DB2 Connect의 SQLCODE 재맵핑 변경.	397
순차값 생성.	375	새 메시지 및 변경된 메시지	397
저장 프로시저어	378	콜 레벨 인터페이스(CLI) 메시지.	397

DB2 메시지.	398	복합 텍스트 명령문(프로시듀어)에 대한 복합 텍	
DBI 메시지.	398	스트 명령문 변경사항.	485
Data Warehouse Center(DWC) 메시지	398	RETURN	487
SQL 메시지.	400	SIGNAL.	489
정정된 SQLSTATES.	402	SQL 제한사항	492
SQL 참조서	403	카탈로그 뷰.	492
하나의 PDF 파일에 제공되는 SQL 참조서	403	SYSCAT.SEQUENCES	493
언어 요소	403	DB2 Stored Procedure Builder	495
이름 지정 규칙 및 내재된 오브젝트 이름 규정	403	DB2 Stored Procedure Builder에 대한 Java 1.2	
데이터 링크 할당	403	지원	495
표현식.	403	DB2 저장 프로시듀어의 원격 디버그	496
함수	410	Windows, OS/2 또는 UNIX 플랫폼에서 SQL 프	
새로운 함수 및 프로시듀어 사용	410	로시듀어 빌드	496
스칼라 함수.	410	Solaris 플랫폼에서 DB2 Stored Procedure	
테이블 함수.	437	Builder 사용	496
프로시듀어	441	알려진 문제점 및 제한사항.	497
조회	444	대만어 로케일에서 DB2 Stored Procedure	
select-statement/구문 도표.	444	Builder 사용	497
select-statement/fetch-first-clause	445	UNIX(AIX, Sun Solaris, Linux) 설치 및 Stored	
SQL문	445	Procedure Builder	497
파티션 키 갱신 지원	445	OS/390에서 SQL 저장 프로시듀어 빌드	498
유니코드 데이터베이스의 더 큰 색인 키	446	SQL 저장 프로시듀어	498
ALTER SEQUENCE	446	Java 저장 프로시듀어 내보내기	498
ALTER TABLE	450	OS/390에서 저장 프로시듀어 삽입	498
복합 SQL(Embedded)	454	워크스테이션 서버에서 SQL 저장 프로시듀어에 대	
복합 텍스트 명령문(동적)	454	한 빌드 옵션 설정.	499
CREATE FUNCTION(소스 또는 템플릿) 457		OS/390에서 빌드된 저장 프로시듀어의 WLM 주소	
CREATE FUNCTION(SQL 스칼라, 테이블 또		공간을 자동으로 새로 고침.	499
는 행).	458	OS/390에서 Java 저장 프로시듀어 개발	499
CREATE METHOD	460	MQSeries 및 OLE DB용 DB2 테이블 사용자 정	
CREATE SEQUENCE.	462	의 함수(UDF) 빌드	500
CREATE TRIGGER	467	유니코드 갱신	501
CREATE WRAPPER	470	소개	501
DECLARE CURSOR	470	DB2 유니코드 데이터베이스 및 응용프로그램	
DELETE	471	문서 갱신	503
DROP	472	SQL 참조서.	504
GRANT(순차 특권)	472	언어 요소	504
INSERT	473	함수	508
SELECT INTO	474	CLI 안내서 및 참조서	508
SET ENCRYPTION PASSWORD	475	고급 기능 사용.	508
SET 전이 변수.	476	DB2 CLI 및 ODBC	514
UPDATE	480	데이터 이동 유틸리티 안내 및 참조서	515
SQL 프로시듀어를 이제는 SQL 제어 명령문이라고		내보내기가져오기/로드 유틸리티 파일 형식	515
합니다.	480		
SQL 프로시듀어 명령문.	481		
FOR	483	제7부 호스트 시스템에 연결	517

연결성 보충 설명서	519
VM 환경에서 응용프로그램 서버 설정	519
CLI/ODBC/JDBC 구성 PATCH1 및 PATCH2 설 정	519

제8부 일반 정보 523

일반 정보	525
DB2 Universal Database 비즈니스 인텔리전스 Quick Tour.	525
DB2 Everywhere는 현재 DB2 Everyplace를 의미 함	526
마우스 필요.	526
DB2 Run-Time Client에서 바인드 시도시 "바인드 파일이 없습니다"라는 오류를 리턴	526
검색 발견	526
HP-UX 11에 대한 메모리 창.	526
dlfm_client_conf 실패에 대한 사용자 조치	528
복사 디먼이 dlfm stop으로 중지되지 않는 드문 이 벤트	528
DB2 DFS Client Enabler 설치 제거	528
Windows NT에서의 클라이언트 인증	528
fork 중에 자동 로드 프로그램이 정지할 수도 있음	529
DATALINK 복원	529
Windows NT용 IBM 통신 서버(CS/NT)에서 사용 자 ID 및 암호 정의	529
노드 정의	529
연합 시스템 제한사항	530
DataJoiner 제한사항	531
Windows NT용 히브리어 Information Catalog Manager.	531
Microsoft SNA Server 및 SNA Multisite Update(2단계 협약) 지원	531
DB2의 SNA SPM이 Windows 부팅 후에 실패함	532
DAS(DB2 Administrator Server)에 대한 로케일 설정	532
단축키가 작동하지 않음	533
Windows NT 및 Windows 2000에서 DB2에 대 한 서비스 계정 요구사항	533
버전 6에서 작성된 Query Patroller 사용자의 EXECUTE 특권 상실	533
Query Patroller 제한사항	534
DWC(Data Warehouse Center)에서 사용될 모든 사용자 정의 프로그램에 협약 필요	534

Data Warehouse Center 명령행 내보내기를 위한 새 옵션	534
백업 서비스 API(XBSA)	535
OS/390 에이전트	535
설치 개요	536
설치 세부사항	536
추가 에이전트 기능 설정	539
트리거 프로그램(XTClient)으로 웨어하우스 단계 스케줄링	541
변환기.	542
DB2 계열 외의 데이터베이스 액세스	546
OS/390용 DB2 유틸리티 수행	548
복제	550
에이전트 로그	551
Windows NT에서 클라이언트쪽 캐칭	552
Enterprise Edition UNIX CD-ROM에서의 시험 사용판.	552
DB2 Connect Enterprise Edition UNIX CD-ROM에서의 시험 사용판	553
Data Links Manager 삭제.	554
SMIT May Remove Additional Filesets를 사용하 여 DLFM 구성요소 설치 제거	554
Windows 2000에서 CLP 사용시 SQL1035N 오류 발생	554
SQL 지원에 대해 개선된 기능	555
Linux에서 DB2용 Gnome 및 KDE 바탕 화면 통 합	555
Windows 2000 Terminal Server의 관리 모드에서 DB2 실행	555
백업 및 복원 명령에 대한 온라인 도움말	556
"Warehouse Manager"는 "DB2 Warehouse Manager"이어야 함	556

제9부 추가 정보 557

추가 정보	559
DB2 Universal Database 및 DB2 Connect 온라 인 지원	559
DB2 잡지	559

제10부 부록 561

부록. 주의사항	563
등록상표 및 서비스 상표	565
색인	567

DB2 Universal Database 7의 사용을 환영합니다!

주: 이 릴리스 정보를 보다 잘 볼 수 있도록 글꼴을 monospace로 설정하십시오.

DB2 Universal Database 및 DB2 Connect 지원 사이트는 규칙적으로 갱신됩니다. 최신 정보가 있는지 <http://www.ibm.com/software/data/db2/udb/winos2unix/support>를 확인하십시오.

이 파일에는 인쇄된 DB2 매뉴얼에 포함되어 있지 않은 다음의 제품에 대한 정보가 들어 있습니다.

- IBM DB2 Universal Database Personal Edition, 버전 7.2
- IBM DB2 Universal Database Workgroup Edition, 버전 7.2
- IBM DB2 Universal Database Enterprise Edition, 버전 7.2
- IBM DB2 Data Links Manager, 버전 7.2
- IBM DB2 Universal Database Enterprise - Extended Edition, 버전 7.2
- IBM DB2 Query Patroller, 버전 7.2
- IBM DB2 Personal Developer's Edition, 버전 7.2
- IBM DB2 Universal Developer's Edition, 버전 7.2
- IBM DB2 Data Warehouse Manager, 버전 7.2
- IBM DB2 Relational Connect, 버전 7.2

한편, 다음 제품에는 별도의 릴리스 정보 파일 READCON.TXT가 제공됩니다.

- IBM DB2 Connect Personal Edition, 버전 7.2
- IBM DB2 Connect Enterprise Edition, 버전 7.2

새로운 기능 책에는 버전 7.2의 주요 DB2 향상 내용 중 일부에 대한 개요가 들어 있습니다. 7.2 버전의 새로운 기능 책이 없으면 <http://www.ibm.com/software/data/db2/udb/winos2unix/support>에서 내용을 열람하거나 다운로드할 수 있습니다.

주: 페이지의 왼쪽의 갱신 막대(())는 첫 릴리스 정보가 발행된 이후에 추가되거나 수정된 행을 나타냅니다.

제1부 특별 주의사항

특별 주의사항

DB2 UDB 버전 7의 액세스 가능성 기능

DB2 UDB 제품군은 장애자들이 보다 쉽게 제품에 액세스할 수 있도록 해주는 여러 기능을 제공합니다. 다음과 같은 기능이 제공됩니다.

- 키보드 입력 및 탐색을 용이하게 하는 기능
- 표시 등록정보를 향상시키는 기능
- 오디오 및 비주얼 경보 신호에 대한 옵션
- 지원 기술과의 호환성
- 운영 체제의 액세스 가능성 기능과의 호환성
- 액세스 가능한 문서 형식

키보드 입력 및 탐색

키보드 입력

DB2 제어 센터는 키보드만을 사용하여 작동시킬 수 있습니다. 메뉴 항목과 제어는 사용자들이 제어를 활성화하거나 키보드에서 직접 메뉴 항목을 선택할 수 있게 하는 액세스 키를 제공합니다. 이들 키는 액세스 키가 나타나는 제어나 메뉴에서 해당 액세스 키에 밑줄이 그어져서 자체적으로 설명됩니다.

키보드 초점

UNIX 기반 시스템에서, 키보드 초점의 위치가 강조표시되어 활동 중인 창의 영역과 사용자의 키 누름이 영향을 주는 위치를 나타냅니다.

액세스 가능한 표시에 대한 기능

DB2 제어 센터에는 사용자 인터페이스를 향상시키고 시력이 좋지 않은 사용자를 위해 액세스 가능성을 높이는 여러 가지 기능들이 있습니다. 이러한 액세스 가능성을 높인 기능으로는 고대비 설정 및 사용자 정의가 가능한 글꼴 등록정보에 대한 지원이 있습니다.

고대비 모드

제어 센터 인터페이스는 운영 체제에서 제공하는 고대비 모드 옵션을 지원합니다. 이 기능은 배경 색상과 전경 색상 간의 높은 대비 차가 필요한 사용자들에게 도움이 됩니다.

글꼴 설정

제어 센터 인터페이스로 사용자는 메뉴와 대화 상자 창에서 텍스트의 색상, 크기 및 글꼴을 선택할 수 있습니다.

색상에 관계 없음

이 제품의 어떤 기능을 사용하기 위해 색상을 구별할 필요는 없습니다.

대체 경보 신호

사용자는 오디오 또는 비주얼 신호를 통해 경보를 수신하도록 선택할 수 있습니다.

지원 기술과의 호환성

DB2 제어 센터 인터페이스는 Via Voice와 같은 화면 판독기 응용프로그램과 호환됩니다. 응용프로그램 모드에 있으면, 제어 센터 인터페이스는 이러한 액세스 가능성을 높인 응용프로그램에 필요한 등록정보를 제공하여 화면상의 정보를 시각 장애 사용자가 사용할 수 있게 해줍니다.

액세스 가능한 문서

DB2 제품군의 문서는 HTML 형식으로 사용할 수 있습니다. 따라서, 사용자는 브라우저에 설정된 표시 기본 설정에 따라 문서를 볼 수 있습니다. 또한, 화면 판독기 및 다른 지원 기술도 사용 가능합니다.

추가 필수 Solaris 패치 레벨

Solaris 버전 2.6용 DB2 Universal Database 버전 7은 UNIX용 DB2 빠른 시작 매뉴얼에 나열된 패치 이외에 패치 106285-02 이상이 필요합니다.

Solaris용 DB2 버전 7에서 지원되는 CPU

UltraSparc 이전의 CPU 버전은 지원되지 않습니다.

파티션 데이터베이스에 노드 추가 시 문제점

페이지 크기가 기본 페이지 크기(4 KB)와 다른 하나 이상의 시스템 임시 테이블 공간이 있는 파티션 데이터베이스에 노드를 추가하면, 오류 메시지 『SQL6073N 노드 추가 조작이 실패했습니다』와 SQLCODE가 나타납니다. 이러한 상태는 노드를 작성할 때 페이지 크기가 4 KB인 IBMDEFAULTBP 버퍼 풀만이 있기 때문에 발생합니다.

예를 들어, **db2start** 명령을 사용하여 현재 파티션 데이터베이스에 노드를 추가할 수 있습니다.

```
DB2START NODENUM 2 ADDNODE HOSTNAME newhost PORT 2
```

파티션 데이터베이스에 기본 페이지 크기의 시스템 임시 테이블 공간이 있는 경우, 다음의 메시지가 리턴됩니다.

SQL6075W 데이터베이스 관리 프로그램 시작 조작이 정상적으로 노드를 추가하였습니다. 모든 노드가 중지되고 다시 시작될 때까지 노드가 활성화되지 않습니다.

그러나, 파티션 데이터베이스에 기본 페이지 크기가 아닌 시스템 임시 테이블 공간이 있는 경우, 다음 메시지가 리턴됩니다.

```
SQL6073N 노드 추가 조작이 실패했습니다. SQLCODE = "<-902>"
```

이와 유사한 예로, 새로운 노드 설명으로 db2nodes.cfg 파일을 수동 갱신한 후에 ADD NODE 명령을 사용할 수 있습니다. 파일을 편집하고 기본 페이지 크기의 시스템 임시 테이블 공간이 있는 파티션 데이터베이스로 ADD NODE 명령을 수행하면 다음의 메시지가 리턴됩니다.

```
DB20000I ADD NODE 명령이 정상적으로 완료되었습니다.
```

그러나, 파티션 데이터베이스에 기본 페이지 크기가 아닌 시스템 임시 테이블 공간이 있는 경우, 다음 메시지가 리턴됩니다.

```
SQL6073N 노드 추가 조작이 실패했습니다. SQLCODE = "<-902>"
```

이와 같은 문제점을 방지하기 위한 한 가지 방법은 다음을 수행하는 것입니다.

```
DB2SET DB2_HIDDENBP=16
```

단, **db2start** 또는 ADD NODE 명령을 발행하기 전에 이를 수행해야 합니다. 이 레지스트리 변수는 DB2가 각각 기본값과 다른 페이지 크기를 사용하여 16 페이지의 숨겨진 버퍼 풀을 할당할 수 있도록 합니다. 따라서, ADD NODE 조작이 성공적으로 완료됩니다.

이들 문제점을 방지하기 위한 다른 방법은 ADD NODE 또는 **db2start** 명령에 WITHOUT TABLESPACES 절을 지정하는 것입니다. 이와 같이 한 후에 CREATE BUFFERPOOL 명령문을 사용하여 버퍼 풀을 작성하고 ALTER TABLESPACE 명령문을 사용하여 시스템 임시 테이블 공간을 버퍼 풀에 연관시켜야 합니다.

페이지 크기가 기본 페이지 크기(4 KB)와 다른 하나 이상의 테이블 공간이 있는 기존의 노드 그룹에 노드를 추가하면, 『SQL0647N ''' 버퍼 풀이 현재 활동 중 상태가 아닙니다.』라는 오류 메시지가 표시됩니다. 이러한 상태는 새로운 노드에서 작성된 기본 값이 아닌 페이지 크기 버퍼 풀이 테이블 공간에 대해 활성화되지 않았기 때문에 발생 합니다.

예를 들어, ALTER NODEGROUP 명령문을 사용하여 노드 그룹에 노드를 추가할 수 있습니다.

```
DB2START
CONNECT TO mpp1
ALTER NODEGROUP ng1 ADD NODE (2)
```

노드 그룹에 기본 페이지 크기의 테이블 공간이 있는 경우, 다음의 메시지가 리턴됩니다.

SQL1759W "<ng1>" 노드 그룹에서 오브젝트에 대한 데이터 위치지정을 변경하여 추가된 일부 노드를 포함시키거나 일부 삭제 노드를 제외시키려면 노드 그룹 재분배가 필요합니다.

그러나, 노드 그룹에 기본 페이지 크기가 아닌 테이블 공간이 있는 경우, 리턴되는 메시지는 다음과 같습니다.

```
SQL0647N "" 버퍼 풀이 현재 활동 중 상태가 아닙니다.
```

이 문제점을 방지하기 위한 한 가지 방법은 ALTER NODEGROUP 명령문을 발행하기 전에 각 페이지 크기에 대해 버퍼 풀을 작성하고 데이터베이스에 다시 연결하는 것입니다.

```
DB2START
CONNECT TO mpp1
CREATE BUFFERPOOL bp1 SIZE 1000 PAGESIZE 8192
CONNECT RESET
CONNECT TO mpp1
ALTER NODEGROUP ng1 ADD NODE (2)
```

이 문제점을 방지하기 위한 두 번째 방법은 다음을 수행하는 것입니다.

```
DB2SET DB2_HIDDENBP=16
```

단, **db2start** 명령과 CONNECT 및 ALTER NODEGROUP 명령문을 발행하기 전에 이를 수행하십시오.

ALTER TABLESPACE 명령문을 사용하여 노드에 테이블 공간을 추가하는 경우에는, 다른 문제점이 발생할 수 있습니다. 예를 들어, 다음과 같습니다.

```
DB2START
CONNECT TO mpp1
ALTER NODEGROUP ng1 ADD NODE (2) WITHOUT TABLESPACES
ALTER TABLESPACE ts1 ADD ('ts1') ON NODE (2)
```

이 일련의 명령과 명령문은 오류 메시지 SQL0647N(예상하는 메시지 SQL1759W가 아님)을 생성합니다.

이 변경을 올바르게 완료하려면, ALTER NODEGROUP... WITHOUT TABLESPACES 명령문 이후에 데이터베이스에 다시 연결해야 합니다.

```
DB2START
CONNECT TO mpp1
ALTER NODEGROUP ng1 ADD NODE (2) WITHOUT TABLESPACES
CONNECT RESET
CONNECT TO mpp1
ALTER TABLESPACE ts1 ADD ('ts1') ON NODE (2)
```

이 문제점을 방지하기 위한 다른 방법은 다음을 수행하는 것입니다.

```
DB2SET DB2_HIDDENBP=16
```

단, **db2start** 명령과 CONNECT, ALTER NODEGROUP 및 ALTER TABLESPACE 명령문을 발행하기 전에 이를 수행하십시오.

이주 중 오류

이주가 성공된 경우라도 이주 중에 db2diag.log 파일에 오류 항목(데이터베이스가 이주되지 않았습다)이 나타나는데, 이는 무시해도 되는 오류입니다.

Red Flag Linux에서 중국어 로케일 수정

중국어 Red Flag Linux 서버 버전 1.1을 사용하는 경우, 중국어 로케일 수정사항을 받기 위해 Red Flag에 접속하십시오. 버전 1.1용 중국어 로케일 수정이 없으면, DB2는 중국어의 코드 페이지가 1386임을 인식하지 못합니다.

이동 가능한 드라이브가 접속되지 않은 경우 DB2 설치 정지

접속되지 않은 제거 가능한 드라이브가 있는 컴퓨터를 사용하여 DB2를 설치하는 경우, 설치 유형을 선택한 후 설치가 정지될 수 있습니다. 이 문제점을 해결하려면, -a 옵션을 지정하여 설치 프로그램을 수행하십시오.

```
setup.exe -a
```

일본어 및 중국어 Linux 환경에서 Linux용 DB2에 대한 추가 로케일 설정

추가 로케일 설정은 일본어 또는 중국어 Linux 시스템에서 제어 센터와 같은 Java GUI 도구를 사용하려고 할 때 필요합니다. 일본어 또는 중국어 문자는 이 설정 없이는 올바르게 표시될 수 없습니다. 다음 설정을 사용자 프로파일에 포함시키거나, 제어 센터를 호출하기 전에 항상 명령행에서 이를 수행하십시오.

```
일본어 시스템의 경우  
export LC_ALL=ja_JP
```

```
중국어 시스템의 경우  
export LC_ALL=zh_CN
```

Microsoft Internet Explorer의 제어 센터 문제점

Internet Explorer(IE) 보안 옵션 설정으로 인한 문제점이 있습니다. 제어 센터에서는 서명하지 않은 jar를 사용하므로, 시스템 정보로의 액세스는 보안 관리 프로그램이 사용할 수 없습니다.

이 문제점을 제거하려면, 다음과 같이 IE 보안 옵션을 다시 구성하십시오.

1. 보기 메뉴(IE4) 또는 도구 메뉴(IE5)에서 인터넷 옵션을 선택하십시오.
2. 보안 페이지에서, 신뢰할 수 있는 사이트를 선택하십시오.
3. 사이트를 클릭하십시오

4. 제어 센터 웹 서버를 신뢰할 수 있는 사이트 목록에 추가하십시오. 제어 센터 웹 서버가 같은 도메인에 있으면(도메인 이름없이) 웹 서버 이름만을 추가하는 것이 유용할 수 있습니다. 예를 들어, 다음과 같습니다.

```
http://ccWebServer.ccWebServerDomain
http://ccWebServer
```

5. 확인을 클릭하십시오.
6. 사용자 정의 수준...을 클릭하십시오.
7. 아래로 화면 이동하여 **Java** -> **Java** 권한으로 가서 사용자 정의를 선택하십시오.
8. **Java** 사용자 정의 설정...을 클릭하십시오.
9. 권한 편집 페이지를 선택하십시오.
10. 아래로 화면 이동하여, 서명하지 않은 콘텐츠 -> 서명하지 않은 콘텐츠 실행 -> 추가 서명하지 않은 권한 -> 시스템 정보로 가서 사용을 선택하십시오.
11. 열린 각 창마다 확인을 클릭하십시오.

Windows 환경에서의 Information Catalog Manager와 Sybase 사이의 비호환

Sybase Open Client가 탑재된 Windows NT 또는 Windows 2000에 ICM(Information Catalog Manager) 버전 7을 설치하면 오류가 발생하고 Sybase Utilities는 작동을 중지합니다. 아래와 유사한 오류 메시지가 발생합니다.

```
LIBTCL.DLL을 초기설정할 수 없습니다. SYBASE 환경 변수가
올바르게 설정되었는지 확인하십시오.
```

LC_ALL 환경 매개변수를 Windows 환경 매개변수에서 제거하여 이런 오류가 발생하지 않도록 하십시오. LC_ALL은 로케일 범주 매개변수입니다. 로케일 범주는 지역화 루틴이 프로그램이 로케일 정보의 어떤 부분을 사용할지를 지정하기 위해 사용하는 정확한 상수입니다. 로케일은 프로그램의 특정 부분에 대해 사용자 정의해야 하는 지역(또는 국가)을 의미합니다. 로케일 종속 영역에는 예를 들어 날짜 형식 또는 화폐 단위 표시형식 등이 포함됩니다. LC_ALL은 모든 로케일 특정 작동(모든 범주)에 영향을 줍니다.

ICM이 Windows NT 플랫폼에서 Sybase와 공존하도록 LC_ALL 환경 매개변수를 제거하면, 다음 기능이 더 이상 작동하지 않습니다.

- Information Catalog User
- Information Catalog Administrator
- Information Catalog Manager

제어 센터 기능 손실

DB2 서버에 FixPak 2를 적용하면 이전 레벨 제어 센터 클라이언트에 대한 문제점이 없어야 합니다. 하지만, DB2 버전 7.2에서, 이전 레벨 제어 센터 클라이언트는 거의 모든 기능을 상실합니다. 이 경우 이전 레벨이란 FixPak 6 이전의 클라이언트 버전 6 그리고 FixPak 2 이전의 클라이언트 버전 7을 말합니다. 클라이언트 버전 5는 영향을 받지 않습니다.

수정 방법은 영향을 받는 클라이언트를 업그레이드하는 것입니다. 버전 6 클라이언트는 FixPak 6 이상으로 업그레이드해야 하고, 버전 7 클라이언트는 FixPak 2 이상으로 업그레이드해야 합니다.

DB2 UDB가 포함되어 있지 않은 Netscape CD

이제 Netscape CD에는 DB2 UDB가 포함되어 있지 않습니다. Netscape 제품은 <http://www.netscape.com>에서 사용 가능합니다.

XML Readme 파일 오류

DB2 XML Extender용 버전 7.1 README.TXT 파일의 "고려사항" 부분을 보면 다음과 같은 내용이 있습니다.

3. DB2 UDB 기본 버전은 DB2 UDB 버전 7.1입니다. DB2 UDB 버전 6.1을 AIX와 Solaris에서 사용하려면 DB2 UDB V6.1 인스턴스 및 DB2 UDB V6.1 라이브러리가 함께 실행되고 있는지 확인해야 합니다.

이는 잘못된 정보입니다. **DB2 XML Extender**는 **DB2 버전 7.1 및 7.2**에만 지원됩니다.

readme.aix, readme.nt 및 readme.sun 파일에 나타난 소프트웨어 요구사항은 다음과 같습니다.

- FP1_U465423 이상인(AIX) DB2 UDB 6.1
- FixPak 3이 설치된(NT) DB2 Universal Database 버전 6.1 이상
- FixPak FP1_U465424 이상인(Sun) DB2 UDB 버전 6.1

이는 잘못된 정보입니다. **DB2 XML Extender**는 **DB2 버전 7.1**이나 **7.2**에서 사용 가능합니다.

S/390용 Linux에서의 데이터 손실 가능성

S/390용 Linux에서 2.2 시리즈 커널과 함께 DB2를 사용할 때 Linux 머신에서 사용할 가능한 RAM은 1 GB 미만으로 제한되어야 합니다. RAM을 1 GB로 제한해야 Linux 커널 버그로 인한 DB2에서의 데이터 손실이 발생하지 않습니다.

| 이러한 제한은 S/390용 Linux의 DB2에만 해당되며 Intel의 Linux에는 해당되지 않습
| 니다.

| 1 GB 이상의 RAM도 사용 가능한 [http://www10.software.ibm.com/developerworks/
| opensource/linux390/alpha_src.html](http://www10.software.ibm.com/developerworks/opensource/linux390/alpha_src.html) 에서 커널 패치를 사용할 수 있습니다.

| **Windows 2000에서의 DB2 UDB**

| 이 릴리스 정보에서 Windows NT에 대한 참조는 달리 지정하지 않는 한 Windows
| 2000에도 해당됩니다.

온라인 문서(HTML, PDF 및 검색)

Windows 2000 운영 체제에서 지원되는 웹 브라우저

Windows 2000에서는 Microsoft Internet Explorer를 사용하십시오.

Netscape를 사용할 경우에는, 다음 사항에 유의하십시오.

- Netscape를 사용하여 Windows 2000에서 DB2 온라인 정보 검색을 완료하는 데는 시간이 아주 오래 걸릴 수도 있습니다. Netscape는 사용 가능한 모든 CPU 자원을 사용하며 무한정으로 수행될 것으로 나타납니다. 검색 요청 후 검색 결과가 표시되는 동안, 다른 창을 클릭하여 활성화 창을 변경하는 것이 좋습니다. 그러면, 적당한 시간 내에 검색 결과가 리턴됩니다.
- 제어 센터에서, Netscape 브라우저 창에 도움말이 표시되도록 요청한 후, 브라우저 창을 열린 채로 두고 제어 센터의 다른 부분에서 나중에 도움말을 요청하면, 브라우저는 이전 도움말을 표시한 채로 변경되지 않을 수도 있습니다. 브라우저를 닫고 도움말을 다시 요청하면, 올바른 도움말이 제공됩니다. 12 페이지의 『Netscape 시작 시 발생하는 오류 메시지』의 단계를 수행해도 이 문제점을 수정할 수 있습니다. 제어 센터에 대한 도움말을 요청하기 전에 브라우저를 닫아서 문제점을 해결할 수도 있습니다.
- 제어 센터 도움말이나, 정보 센터로부터 주제를 요청할 경우, 오류 메시지가 표시될 수도 있습니다. 이를 수정하려면, 12 페이지의 『Netscape 시작 시 발생하는 오류 메시지』의 단계를 수행하십시오.

Solaris에서 DB2 온라인 정보 검색

Solaris에서 DB2 온라인 정보를 검색하는 데 문제가 있으면 /etc/system에서 시스템의 커널 매개변수를 확인하십시오. 다음은 DB2의 검색 시스템인 NetQuestion에서 필요한 최소 커널 매개변수입니다.

```
semsys:seminfo_semmni 256
semsys:seminfo_semmap 258
semsys:seminfo_semmns 512
semsys:seminfo_semmnu 512
semsys:seminfo_semmsl 50
shmsys:shminfo_shmmax 6291456
shmsys:shminfo_shmseg 16
shmsys:shminfo_shmmni 300
```

커널 매개변수를 설정하려면, /etc/system의 끝에 다음 행을 추가하십시오.

```
set <semaphore_name> = value
```

새로운 값이나 변경된 값이 적용되려면 시스템을 재부트해야 합니다.

TCP/IP를 사용하도록 OS/2용 NetQuestion 전환

OS/2 시스템에서 TCP/IP를 사용하기 위해 NetQuestion을 전환하는 데 대한 지시사항이 완전하지 않습니다. 그 지시사항에 언급된 *.cfg 파일의 위치가 NetQuestion 설치 디렉토리의 데이터 서브디렉토리입니다. 다음 명령 중 하나를 입력하여 NetQuestion 설치 디렉토리를 판별할 수 있습니다.

```
echo %IMNINSTRV% //for SBCS installations
echo %IMQINSTRV% //for DBCS installations
```

Netscape 시작 시 발생하는 오류 메시지

Netscape를 시작하려고 할 때 다음과 같은 오류 메시지가 표시될 경우,

```
<file path> 파일(또는 해당되는 구성요소 중 하나)을 찾을 수 없습니다.
경로와 파일 이름이 올바르고, 필요한 모든 라이브러리가
사용 가능한지 확인하십시오.
```

```
"D:\Program Files\SQLLIB\CC\..\doc\html\db2he1p\XXXXX.htm"을 열 수 없습니다.
```

Windows NT, 95 또는 98에서 이 문제점을 정정하려면 다음 단계를 수행하십시오 (Windows 2000에서의 수행 사항에 대해서는 아래를 참조하십시오).

1. 시작 메뉴에서 프로그램 → Windows 탐색기를 선택하십시오. Windows 탐색기가 열립니다.
2. Windows 탐색기에서 보기 → 옵션을 선택하십시오. 옵션 노트북이 열립니다.
3. 파일 형식 탭을 클릭하십시오. 파일 형식 페이지가 열립니다.
4. 등록된 파일 형식 필드에서 Netscape Hypertext Document를 강조표시하고 편집을 클릭하십시오. 파일 형식 편집 창이 열립니다.
5. 작업 필드에서 "Open"을 강조표시하십시오.
6. 편집 버튼을 클릭하십시오. 명령 편집 창이 열립니다.
7. **DDE** 메시지 선택란을 선택 취소하십시오.
8. 작업을 수행할 응용프로그램 필드에서, 문자열의 맨 끝에 "%1"이 표시되는지 확인하십시오(물음표와, 첫번째 물음표 앞의 공백도 확인하십시오).

Windows 2000에서 해당 오류 메시지가 표시될 경우에는, 다음 단계를 수행해야 합니다.

1. 시작 메뉴에서 **Windows** 탐색기를 선택하십시오. Windows 탐색기가 열립니다.
2. Windows 탐색기에서 도구 → 폴더 옵션을 선택하십시오. 폴더 옵션 노트북이 열립니다.
3. 파일 형식 탭을 클릭하십시오.
4. 파일 유형 페이지의 등록된 파일 유형 필드에서 HTM Netscape Hypertext Document를 강조표시한 후 고급을 누르십시오. 파일 형식 편집 창이 열립니다.
5. 명령 필드에서 "open"을 강조표시하십시오.

6. 편집 버튼을 클릭하십시오. 명령 편집 창이 열립니다.
7. **DDE** 메시지 선택란을 선택 취소하십시오.
8. 작업을 수행할 응용프로그램 필드에서, 문자열의 맨 끝에 "%1"이 표시되는지 확인하십시오(물음표와, 첫번째 물음표 앞의 공백도 확인하십시오).
9. 확인을 클릭하십시오.
10. HTML Netscape Hypertext Document와 SHTML Netscape Hypertext Document 파일 형식에 대해 4 - 8 단계를 반복하십시오.

Unix 기반 시스템에서 Adobe Acrobat Reader에 대한 구성 요구사항

Acrobat Reader는 Unix 기반 플랫폼에서 영어로만 제공되며 영어 이외의 언어 로케일로 PDF 파일을 열려고 시도하면 오류가 리턴될 수 있습니다. 이들 오류는 PDF 파일의 글꼴 액세스 또는 추출 문제점을 제시하지만 실제로는 영문 Acrobat Reader가 UNIX 영어 외의 언어 로케일 내에서는 제대로 기능을 수행할 수 없으므로 인해 발생합니다.

이런 경우 PDF 파일을 보려면, 영문 Acrobat Reader를 시작하기 전에 다음의 단계 중 하나를 수행하여 영어 로케일로 전환하십시오.

- 시작 스크립트 파일에서 `#!/bin/sh` 명령문 이후에 다음의 행을 추가하여 Acrobat Reader의 시작 스크립트를 편집하십시오.

```
LANG=C;export LANG
```

이렇게 하면 Netscape Navigator 등의 다른 응용프로그램이나 응용프로그램 도움말 메뉴에서 Acrobat Reader를 시작할 때 올바르게 작동할 것입니다.

- 명령 프롬프트에 `LANG=C`를 입력하여 Acrobat Reader의 응용프로그램 환경을 영어로 설정하십시오.

자세한 내용은 Adobe Systems(<http://www.Adobe.com>)에 문의하십시오.

하나의 PDF 파일에 제공되는 SQL 참조서

각 책의 "DB2 라이브러리 사용" 부록은 SQL 참조서가 두 개의 다른 볼륨으로 된 PDF 형식으로 사용할 수 있다고 설명하고 있는데 이는 잘못된 정보입니다.

인쇄된 책이 두 볼륨으로 나타나고 두 개의 해당 양식 번호가 올바르게라도 하나의 PDF 파일만이 있으며 여기에는 두 볼륨이 모두 들어 있습니다. PDF 파일 이름은 db2s0x70입니다.

제2부 설치 및 구성

일반 설치 정보	17	AIX에서 DB2 Data Links Manager에 대한 백업	메소드 선택	24
지원되는 모든 DB2 클라이언트의 설치 패키지 다운		Solaris 운영 환경에서 DB2 Data Links Manager용	백업 방법 선택	26
로드.	17	Windows NT에서 DB2 Data Links Manager에 대	한 백업 메소드 선택	27
Windows 2000에 DB2 설치	17	AIX에서 저널 파일 시스템 백업		29
특수 레지스터로 정의된 뷰와 관련된 이주 발행 . . .	17	Windows NT에서 데이터 링크의 관리자 그룹 특권		30
Windows 2000에서의 IPX/SPX 프로토콜 지원 . . .	17	Data Links File System Filter(DLFF) 설치에 대한	로깅 최소화	30
DB2의 이전 버전을 업그레이드하기 전에 DB2 프로		설치 후 메시지 로깅		31
세스 중지.	18	SMIT May Remove Additional Filesets를 사용하	여 DLFM 구성요소 설치 제거.	32
다른 DB2 제품이 이미 설치된 경우의 DB2 설치 이		호스트 이름을 시작/판별하기 전에		32
후 db2iupdt 수행	18	Data Links File Manager 작업: DB2 데이터베이스	에서 DB2 Data Links Manager 삭제 후 정리 . . .	33
DB2 제어 센터를 수행할 Linux 환경 설정.	18	DLFM1001E (새 오류 메시지).		34
S/390 상의 Linux용 DB2 Universal Database		DLFM 설정 구성 파일 옵션.		34
Enterprise Edition 및 DB2 Connect Enterprise		AIX에서 Data Links/DFS Script dmapp_prestart	수행 시 오류.	35
Edition.	19	Tivoli Space Manager와 Data Links 통합		35
UNIX용 DB2 Universal Database Enterprise -		제한사항 및 한계		36
Extended Edition 빠른 시작	19	AIX용 DB2 Data Links Manager 설치 및 구성		37
HP-UX용 shmseg 커널 매개변수	20	공통적인 설치 고려사항		37
IBM Visual Warehouse 제어 데이터베이스 이주	20	DB2 File Manager 버전 5.2에서 DB2 Data	Links Manager 버전 7로 이주.	37
웨어하우스 제어 데이터베이스에 액세스.	20	설치 및 구성 보충 설명서		39
Data Links Manager 빠른 시작.	21	UNIX 운영 체제에 DB2 클라이언트 설치		39
메시지 "접두부에 대한 afsfid 확보 시 오류 발생"으		HP-UX 커널 구성 매개변수.		39
로 Dlfm 시작이 실패함	21	사용자 자신의 응용프로그램 수행		39
아카이브 파일에 대한 Tivoli Storage Manager 클		Run-time Client를 사용한 데이터베이스 유틸리티	바인딩	39
래스 설정	21	ODBC를 사용한 DB2에 대한 UNIX 클라이언트	액세스	39
DFS Client Enabler에 대한 디스크 공간 요구사항	21	여러 데이터 소스에 액세스하기 위해 연합 시스템 설	정	39
AIX에서 데이터 링크 파일 관리 프로그램 백엔드 프		연합 시스템		39
로세스 모니터	21	제한사항		40
AIX용 DB2 Data Links Manager 설치 및 구성:		DB2 Relational Connect 설치.		40
DCE-DFS 환경에서의 추가 설치 고려사항	22	Windows NT 서버에 DB2 Relational		
실패한 "dlfm add_prefix" 명령	22	Connect 설치		40
AIX용 DB2 Data Links Manager 설치 및 구성:				
db2setup 유틸리티를 사용하여 AIX에 DB2 Data				
Links Manager 설치	23			
AIX용 DB2 Data Links Manager 설치 및 구성:				
DCE-DFS 사후 설치 타스크	23			
AIX용 DB2 Data Links Manager 설치 및 구성:				
Smit를 사용하여 수동으로 DB2 Data Links				
Manager 설치	24			
DB2 Data Links DFS Client Enabler 설치 및 구				
성	24			
Solaris용 DB2 Data Links Manager 설치 및 구성	24			

	AIX, Linux 및 Solaris 운영 환경 서버에	
	DB2 Relational Connect 설치.	41
	Oracle 데이터 소스에 액세스	43
	문서 오류	43
	Sybase 데이터 소스에 액세스(새 장).	44
	Sybase 데이터 소스를 연합 서버에 추가.	44
	1 단계: 환경 변수 설정 및 프로파일 레지스트	
	리 갱신.	45
	2 단계: DB2를 Sybase 클라이언트 소프트웨어	
	어에 링크(AIX 및 Solaris만 해당)	46
	3 단계: DB2 인스턴스 재순환	46
	4 단계: 인터페이스 파일 작성 및 설정	46
	5 단계: 랩퍼 작성	47
	6 단계: DB2_DJ_COMM 환경 변수 설정	47
	7 단계: 서버 작성	48
	선택적: 8 단계: CONNECTSTRING 서버 옵션	
	설정.	48
	9 단계: 사용자 맵핑 작성	50
	10 단계: 테이블 및 뷰에 대한 별명 작성	50
	Sybase 코드 페이지 지정.	51
	ODBC를 사용하여 Microsoft SQL Server 데이터	
	소스에 액세스(새 장)	51
	연합 서버에 Microsoft SQL Server 데이터 소스	
	추가.	52
	1 단계: 환경 변수 설정(AIX만 해당).	52
	2 단계: 셸 스크립트 수행(AIX만 해당)	53
	3 단계: 선택적: DB2_DJ_COMM 환경 변수	
	설정.	54
	4 단계: DB2 인스턴스 재순환(AIX만 해당)	54
	5 단계: 랩퍼 작성	54
	6 단계: 서버 작성	55
	7 단계: 사용자 맵핑 작성	56
	8 단계: 테이블 및 뷰에 대한 별명 작성.	56
	9 단계: ODBC 추적 얻기(선택적).	57
	Microsoft SQL Server 코드 페이지 검토	57

일반 설치 정보

지원되는 모든 DB2 클라이언트의 설치 패키지 다운로드

버전 7 이전 버전의 모든 클라이언트를 비롯하여 지원되는 모든 DB2 클라이언트의 설치 패키지를 다운로드하려면, IBM DB2 클라이언트 응용프로그램 인에이블러 팩 웹 사이트인 <http://www.ibm.com/software/data/db2/db2tech/clientpak.html>로 가십시오.

Windows 2000에 DB2 설치

Windows 2000에서, 이전 버전의 DB2 위에 설치하거나 현재 버전을 재설치하는 경우, 모든 DB2 서비스의 복구 옵션이 "동작하지 않음"으로 설정되었는지 확인하십시오.

특수 레지스터로 정의된 뷰와 관련된 이주 발행

특수 레지스터 USER 또는 CURRENT SCHEMA가 뷰 컬럼을 정의하는 데 사용된 경우, 데이터베이스 이주 이후에 뷰는 사용할 수 없게 됩니다. 예를 들어, 다음과 같습니다.

```
create view v1 (c1) as values user
```

버전 5에서, USER 및 CURRENT SCHEMA는 데이터 유형이 CHAR(8)이었지만 버전 6 이후로는 VARCHAR(128)로 정의되었습니다. 이 예에서, 뷰가 버전 5에서 작성된 경우, 컬럼 c1의 데이터 유형은 CHAR이고 데이터베이스 이주 이후에 CHAR로 남아 있게 됩니다. 뷰가 이주 이후에 사용되면, 런타임에 컴파일되지만 데이터 유형이 일치하지 않으므로 실패합니다.

이를 해결하려면 뷰를 삭제하고 다시 작성하십시오. 뷰를 삭제하기 전에, SYSCAT.VIEWS 카탈로그 뷰를 조회하여 뷰 생성에 사용된 구문을 캡처하십시오. 예를 들어, 다음과 같습니다.

```
select text from syscat.views where viewname='<>'
```

Windows 2000에서의 IPX/SPX 프로토콜 지원

이 정보는 "가능한 클라이언트 서버 연결 시나리오"라는 절에 있는 빠른 시작 책의 설치 계획 장을 참조합니다.

공개된 프로토콜 지원 도표는 완전하게 올바르지는 않습니다. Windows 2000 클라이언트는 IPX/SPX를 사용하는 OS/2 또는 Unix 기반 서버에 연결되지 않습니다. IPX/SPX를 사용하는 Windows 2000 서버에 연결된 OS/2 또는 Unix 기반 클라이언트는 지원되지 않습니다.

DB2의 이전 버전을 업그레이드하기 전에 DB2 프로세스 중지

이 정보는 Windows용 DB2 빠른 시작 책의 이주 정보를 참조합니다.

Windows 머신에서 수행 중인 DB2의 이전 버전을 업그레이드할 경우, 설치 프로그램은 메모리에 DB2 DLL을 보유하는 프로세스 목록을 포함하는 경고를 제공합니다. 이때, 그 목록에 표시되는 프로세스를 수동으로 중지하거나 설치 프로그램이 이 프로세스를 자동으로 종료할 수 있도록 옵션을 사용할 수 있습니다. 데이터 유실을 피하려면 설치하기 전에 모든 DB2 프로세스를 수동으로 중지하는 것이 좋습니다. DB2 프로세스가 수행되지 않도록 하는 최상의 방법은 Windows 서비스 패널을 통해 시스템의 프로세스를 보는 것입니다. Windows 서비스 패널에서, DB2 서비스, OLAP 서비스 또는 데이터 웨어하우스 서비스가 수행되고 있는지 확인하십시오.

주: Windows 플랫폼에서는 한 번에 하나의 DB2 버전만 수행할 수 있습니다. 예를 들어, 같은 Windows 머신에서 DB2 버전 7과 DB2 버전 6을 동시에 수행할 수 없습니다. DB2 버전 6이 설치된 머신에서 DB2 버전 7을 설치할 경우, 설치 프로그램은 설치하는 동안 DB2 버전 6을 삭제합니다. DB2 이전 버전으로부터의 이주에 대해서는 해당되는 빠른 시작 매뉴얼에서 자세한 내용을 참조하십시오.

다른 DB2 제품이 이미 설치된 경우의 DB2 설치 이후 db2iupdt 수행

다음 정보는 빠른 시작 설치 문서에서 제공되어야 합니다.

UNIX 기반 시스템에서 DB2 UDB 버전 7을 설치할 때 이미 DB2 제품이 설치되어 있으면, **db2iupdt** 명령을 실행하여 이 제품의 새 기능을 사용할 인스턴스를 갱신해야 합니다. 일부 기능은 이 명령을 수행할 때까지 사용할 수 없습니다.

DB2 제어 센터를 수행할 Linux 환경 설정

이 정보는 빠른 시작 책의 "DB2 제어 센터 설치" 장에 포함되어 있어야 합니다.

Linux에서 DB2 설치 프로그램을 나가서 터미널 창으로 리턴한 뒤, 다음 명령을 입력하여 DB2 제어 센터를 수행할 수 있는 올바른 환경을 설정하십시오.

```
su -l <instance name>
export JAVA_HOME=/usr/jdk118
export DISPLAY=<your machine name>:0
```

그리고 나서, 다른 터미널 창을 열고 다음을 입력하십시오.

```
su root
xhost +<your machine name>
```

터미널 창을 닫고 인스턴스 소유자 ID로 로그인한 터미널에 리턴한 후 다음 명령을 입력하십시오.

```
db2cc
```

S/390 상의 Linux용 DB2 Universal Database Enterprise Edition 및 DB2 Connect Enterprise Edition

DB2 Universal Database Enterprise Edition 및 DB2 Connect Enterprise Edition이 이제 S/390의 Linux에서 사용 가능합니다. S/390 머신에 Linux를 설치하기 전에, 소프트웨어 및 하드웨어 요구사항을 알아야 합니다.

하드웨어

S/390 9672 Generation 5 이상, Multiprise 3000.

소프트웨어

- S/390용 SuSE Linux v7.0 또는 zSeries 및 S/390용 Turbolinux Server 6
- S/390용 패치와 함께 커널 레벨 2.2.16(아래를 참조하십시오)
- glibc 2.1.3
- libstdc++ 6.1

S/390 Linux에는 다음 패치가 필요합니다.

- 지금은 패치가 필요하지 않습니다.

최근 갱신 사항을 보려면, <http://www.software.ibm.com/data/db2/linux> 웹 사이트로 가십시오.

주:

1. 32 비트 Intel 기반 Linux 및 S/390 상의 Linux에서만 지원됩니다.
2. 다음은 DB2 버전 7에서 Linux/390에서 사용할 수 없습니다.
 - DB2 UDB Enterprise - Extended Edition
 - DB2 Extenders
 - Data Links Manager
 - DB2 Administrative Client
 - 암호 변경 지원
 - LDAP 지원

UNIX용 DB2 Universal Database Enterprise - Extended Edition 빠른 시작

Linux에서 DB2 Universal Database 설치 및 구성에서는 Linux EEE 클러스터의 각 물리 노드의 커널, glibc 및 libstdc++ 레벨이 같아야 함을 나타내야 합니다.

Linux용 DB2 EEE의 시험 사용판을 다음 웹 사이트에서 다운로드받을 수 있습니다.
<http://www6.software.ibm.com/dl/db2udbd1/db2udbd1-p>

HP-UX용 shmseg 커널 매개변수

빠른 시작 책에 제공된 HP-UX 커널 구성 매개변수 갱신에 대한 정보가 올바르지 않습니다. HP-UX용 shmseg 커널 매개변수에 대한 권장 값은 무시해야 합니다.

기본 HP-UX 값(120)을 대신 사용해야 합니다.

IBM Visual Warehouse 제어 데이터베이스 이주

Windows용 DB2 Universal Database 빠른 시작은 Windows NT와 Windows 2000에서 DB2 Universal Database 버전 7의 일반 설치를 수행하는 동안 활동 중인 웨어하우스 제어 데이터베이스가 이주되는 방법에 대한 정보를 제공합니다. 하나 이상의 웨어하우스 제어 데이터베이스를 이주시켜야 하는 경우, 웨어하우스 제어 데이터베이스 관리 창을 사용하여 추가 데이터베이스를 이주시켜야 합니다. 한 번에 하나의 웨어하우스 제어 데이터베이스만이 활동 상태가 될 수 있습니다. 이주시킨 마지막 데이터베이스가 다음에 Data Warehouse Center에 로그인할 때 사용하려는 것이 아닌 경우, 웨어하우스 제어 데이터베이스 관리 창을 사용하여 사용하려는 데이터베이스를 등록해야 합니다.

웨어하우스 제어 데이터베이스에 액세스

Windows NT에서의 DB2 버전 7의 일반 설치에서, DB2 버전 7 웨어하우스 제어 데이터베이스가 웨어하우스 서버와 함께 작성됩니다. Visual Warehouse 웨어하우스 제어 데이터베이스가 있는 경우, 웨어하우스 제어 데이터베이스의 메타데이터를 DB2 버전 7 Data Warehouse Center에서 사용할 수 있도록 이주시키기 전에 웨어하우스 제어 데이터베이스가 포함된 DB2 서버를 DB2 버전 7로 업그레이드해야 합니다. 계속 사용할 웨어하우스 제어 데이터베이스를 버전 7로 이주해야 합니다. 사용 중인 웨어하우스 제어 데이터베이스의 메타데이터는 DB2 버전 7 설치 프로세스 중에 버전 7로 이주됩니다. 추가 웨어하우스 제어 데이터베이스의 메타데이터를 이주시키려면, Windows NT에서 시작 → 프로그램 → IBM DB2 → 웨어하우스 제어 데이터베이스 관리를 선택하여 시작하는 웨어하우스 제어 데이터베이스 이주 유틸리티를 사용하십시오. DB2 Universal Database for Windows 빠른 시작에서 웨어하우스 제어 데이터베이스 이주에 대한 내용을 참조하십시오.

Data Links Manager 빠른 시작

메시지 "접두부에 대한 afsfid 확보 시 오류 발생"으로 Dlfm 시작이 실패함

DCE-DFS 환경에서 수행 중인 Data Links Manager의 경우, dlfm 시작이 다음 오류로 실패하면, IBM 서비스에 문의하십시오.

접두부에 대한 afsfid를 가져오는 중 오류가 발생했습니다

"dlfm add_prefix"를 사용하여 Data Links Manager에 등록된 DFS 파일세트가 삭제된 경우 오류가 발생할 수 있습니다.

아카이브 파일에 대한 Tivoli Storage Manager 클래스 설정

아카이브 파일에 사용할 TSM 관리 클래스를 지정하려면, DLFM_TSM_MGMTCLASS DB2 레지스트리 항목을 적합한 관리 클래스 이름으로 설정하십시오.

DFS Client Enabler에 대한 디스크 공간 요구사항

DFS Client Enabler는 DB2 Universal Database 클라이언트 또는 서버 설치 중에 선택할 수 있는 선택 구성요소입니다. DFS Client Enabler가 DB2 UDB 클라이언트 또는 서버를 필요로 하지 않고 자체적으로 수행되더라도 DB2 Universal Database 클라이언트 또는 서버 제품을 설치하지 않고는 DFS Client Enabler를 설치할 수 없습니다. DFS Client Enabler 코드에 필요한 2 MB의 디스크 공간 이외에 DB2 Run-Time Client 설치의 일부로 DFS Client Enabler를 설치하는 경우에는 추가로 40 MB를 확보해야 합니다. DB2 Administration Client 또는 DB2 서버 설치의 일부로 DFS Client Enabler를 설치하는 경우, 보다 많은 디스크 공간이 필요합니다. DB2 Universal Database 제품의 디스크 공간 요구사항에 관한 추가 정보는, UNIX용 DB2 빠른 시작 설명서를 참조하십시오.

AIX에서 데이터 링크 파일 관리 프로그램 백엔드 프로세스 모니터

dlfm see 명령의 출력이 변경되었습니다. AIX에서 데이터 링크 파일 관리 프로그램 백엔드 프로세스를 모니터링하기 위해 이 명령을 발행하면, 리턴되는 출력은 다음과 유사합니다.

PID	PPID	PGID	RUNAME	UNAME	ETIME	DAEMON NAME
17500	60182	40838	dlfm	root	12:18	dlfm_copyd_(dlfm)
41228	60182	40838	dlfm	root	12:18	dlfm_chownd_(dlfm)
49006	60182	40838	dlfm	root	12:18	dlfm_upcall_(dlfm)
51972	60182	40838	dlfm	root	12:18	dlfm_gcd_(dlfm)
66850	60182	40838	dlfm	root	12:18	dlfm_retrieved_(dlfm)

```
67216    60182    40838    dlfm     dlfm     12:18 dlfm_de1grp (dlfm)
60182         1    40838    dlfm     dlfm     12:18 dlfm_d_ (dlfm)
```

DLFM SEE 요청이 성공하였습니다.

괄호 안의 이름은 dlfm 인스턴스의 이름이며 이 경우에는 "dlfm"입니다.

AIX용 DB2 Data Links Manager 설치 및 구성: DCE-DFS 환경에서의 추가 설치 고려사항

"설치 요구사항" 절에 다음 정보가 추가되어야 합니다.

DFS 3.1용 e-fix 또는 PTF 세트 1(사용할 수 있는 경우) 중 하나를 설치해야 합니다. e-fix는 다음에서 사용할 수 있습니다.
http://www.transarc.com/Support/dfs/datalinks/efix_dfs31_main_page.html

또한,

dfs 클라이언트는 Data Links Manager를 설치하기 전에 수행 중이어야 합니다. db2setup이나 smitty를 사용하십시오.

"키템 파일" 절에는 다음과 같이 정정해야 하는 오류가 있습니다.

이 키템 파일에는 핵심부와 암호 정보가 포함되어 있고 datalink.ktb라고 하며

올바른 이름: datalink.ktb가 아래 예에서 사용됩니다. 이 파일의 작성은 DLMADMIN 인스턴스가 작성되지 않으면 발생할 수 없기 때문에 "키템 파일" 절은 "DCE-DFS 사 후 설치 태스크" 아래로 이동되어야 합니다.

"Data Links Manager 서버 및 클라이언트" 절에 Data Links Manager 서버는 Data Links Manager 클라이언트 이전에 설치되어야 한다는 것이 명시되어야 합니다.

새로운 절인 "백업 디렉토리"가 추가되어야 합니다.

백업 메소드가 지역 파일 시스템에 대한 것일 경우, 이는 DFS 파일 시스템의 디렉토리여야 합니다. 이 DFS 파일 세트가 DFS 관리자에 의해 작성되었는지 확인하십시오. 이것은 DMLFS 파일 세트가 아니어야 합니다.

실패한 "dlfm add_prefix" 명령

DCE/DFS 환경에서 수행 중인 Data Links Manager의 경우, dlfm add_prefix 명령은 -2061 리턴 코드(백업 실패)로 실패할 수도 있습니다. 이러한 상황이 발생할 경우, 다음 단계를 수행하십시오.

1. **dlfm stop** 명령을 발행하여 Data Links Manager 디먼 프로세스를 중지하십시오.
2. **dlfm stopdbm** 명령을 발행하여 DB2 프로세스를 중지하십시오.
3. **dce_login root** 명령을 발행하여 dce 루트 신임장을 확보하십시오.
4. **dlfm startdbm** 명령을 발행하여 DB2 프로세스를 시작하십시오.

5. **dlfm add_prefix** 명령을 발행하여 Data Links Manager로 파일 세트를 등록하십시오.
6. **dlfm start** 명령을 발행하여 Data Links Manager 디먼 프로세스를 시작하십시오.

AIX용 DB2 Data Links Manager 설치 및 구성: db2setup 유틸리티를 사용하여 AIX에 DB2 Data Links Manager 설치

"DB2 데이터베이스 DLFM_DB가 작성됨" 절에서, DLFM_DB는 DCE_DFS 환경에서 작성되지 않습니다. 이는 사후 설치 단계로 수행해야 합니다.

"DMAPP에 대한 DCE-DFS 사전 시작 등록" 절에서, 2 단계는 다음으로 변경되어야 합니다.

2. DFS가 시작될 때 DMAPP가 시작되도록 명령이 /opt/dce1ocal/tcl/user_cmd.tcl에 추가됩니다.

AIX용 DB2 Data Links Manager 설치 및 구성: DCE-DFS 사후 설치 타스크

다음의 새로운 절인 "Data Links Manager 설치 완료"가 추가되어야 합니다.

Data Links Manager 서버에서, 다음 단계를 수행하여 설치를 완료해야 합니다.

1. "AIX용 DB2 Data Links Manager 설치 및 구성" 장의 "DCE-DFS 환경에서의 추가 설치 고려사항" 절에 있는 "키템 파일"에 요약된 대로 키템 파일을 작성하십시오.
2. 루트로써 다음 명령을 입력하여 DMAPP를 시작하십시오.


```
stop.dfs all
start.dfs all
```
3. 다음과 같이 dce 루트 신임장을 사용하여 "dlfm setup"을 수행하십시오.
 - a. Data Links Manager 관리자 DLMADMIN으로 로그인하십시오.
 - b. 루트로써 dce_login을 발행하십시오.
 - c. dlfm setup 명령을 입력하십시오.

Data Links Manager 클라이언트에서, 다음 단계를 수행하여 설치를 완료하십시오.

1. "AIX용 DB2 Data Links Manager 설치 및 구성" 장의 "DCE-DFS 환경에서의 추가 설치 고려사항" 절에 있는 "키템 파일"에 요약된 대로 키템 파일을 작성하십시오.
2. 루트로써 다음 명령을 입력하여 DMAPP를 시작하십시오.


```
stop.dfs all
start.dfs all
```

AIX용 DB2 Data Links Manager 설치 및 구성: Smit를 사용하여 수동으로 DB2 Data Links Manager 설치

"SMIT 사후 설치 타스크" 절의 7 단계에서 명령 "dce_login root"가 "dlfm setup" 이전에 발행되어야 한다고 내용을 수정하십시오. 11 단계는 필요하지 않습니다. 이 단계는 6 단계(dlfm server_conf)나 8 단계(dlfm client_conf)가 수행되면 자동으로 수행됩니다. 또한, 12 단계도 제거하십시오(dlfm start). 설치를 완료하려면, 다음 단계를 수행하십시오.

1. "AIX용 DB2 Data Links Manager 설치 및 구성" 장의 "DCE-DFS 환경에서의 추가 설치 고려사항" 절에 있는 "키템 파일"에 요약된 대로 키템 파일을 작성하십시오.
2. 루트로서 다음 명령을 입력하여 DMAPP를 시작하십시오.

```
stop.dfs all
start.dfs all
```

DB2 Data Links DFS Client Enabler 설치 및 구성

"DFS Client Enabler 구성" 절에서 다음 정보를 2 단계에 추가하십시오.

"secval" 명령을 수행하면 항상 구성이 완료됩니다.
그러나, 머신을 재부트하려면 이 명령도 필요할 수 있습니다.
READ PERMISSION DB 파일에 액세스할 때 문제점이 발생할 경우,
DB2 DFS Client Enabler가 설치된 머신을 재부트하십시오.

Solaris용 DB2 Data Links Manager 설치 및 구성

다음 조치는 Solaris용 DB2 Data Links Manager를 설치한 후에 수행되어야 합니다.

1. 다음 세 행을 /etc/system 파일에 추가하십시오.

```
set dlfsdrv:glob_mod_pri=0x100800
set dlfsdrv:glob_mesg_pri=0xff
set dlfsdrv:ConfigDlfsUid=UID
```

여기서 *UID*는 id dlfm의 사용자 ID를 나타냅니다.

2. 머신을 재부트하여 변경사항을 활성화하십시오.

AIX에서 DB2 Data Links Manager에 대한 백업 메소드 선택

디스크 복사 및 XBSA 이외에도, 데이터 링크 서버에 상주하는 파일들을 백업하기 위해 Tivoli Storage Manager(TSM)를 사용할 수도 있습니다.

아카이브 서버로 Tivoli Storage Manager를 사용하려면 다음을 수행하십시오.

1. Data Links 서버에서 Tivoli Storage Manager를 설치하십시오. 자세한 내용은 Tivoli Storage Manager 제품 문서를 참조하십시오.

2. Tivoli Storage Manager 서버를 사용하여 Data Links 서버 클라이언트 응용프로그램을 등록하십시오. 자세한 내용은 Tivoli Storage Manager 제품 문서를 참조하십시오.
3. 다음 환경 변수를 Data Links Manager 관리자의 db2profile 또는 db2cshrc 스크립트 파일에 추가하십시오.

```
(Bash, Bourne 또는 Korn 셸의 경우)
export DSMI_DIR=/usr/tivoli/tsm/client/api/bin
export DSMI_CONFIG=$HOME/tsm/dsm.opt
export DSMI_LOG=$HOME/dldump
export PATH=$PATH:$DSMI_DIR
```

```
(C 셸의 경우)
setenv DSMI_DIR /usr/tivoli/tsm/client/api/bin
setenv DSMI_CONFIG ${HOME}/tsm/dsm.opt
setenv DSMI_LOG ${HOME}/dldump
setenv PATH=${PATH}:$DSMI_DIR
```

4. dsm.sys TSM 시스템 옵션 파일이 \$DSMI_DIR 디렉토리에 있는지 확인하십시오.
5. dsm.opt TSM 사용자 옵션 파일이 *INSTHOME*/tsm 디렉토리에 있는지 확인하십시오. *INSTHOME*은 Data Links Manager 관리자의 홈 디렉토리입니다.
6. /usr/tivoli/tsm/client/api/bin/dsm.sys Tivoli Storage Manager 시스템 옵션 파일에서 *PASSWORDACCESS* 옵션을 generate로 설정하십시오.
7. 처음으로 Data Links File Manager를 시작하기 전에 generate 옵션으로 TSM 암호를 등록하십시오. 이 방법을 사용하면 Data Links File Manager가 TSM 서버에 대한 연결을 초기화할 때 암호를 제공하지 않아도 됩니다. 자세한 내용은 해당되는 TSM 제품 문서를 참조하십시오.
8. *DLFM_BACKUP_TARGET* 레지스트리 변수를 TSM으로 설정하십시오. *DLFM_BACKUP_DIR_NAME* 레지스트리 변수의 값은 이 경우에 무시됩니다. 이렇게 하면 Tivoli Storage Manager 백업 옵션이 활성화됩니다.

주:

- a. 수행 시 *DLFM_BACKUP_TARGET* 레지스트리 변수의 설정을 TSM과 디스크 사이에 변경할 경우, 아카이브된 파일이 새로 지정된 아카이브 위치로 이동되지 않는다는 점에 유의해야 합니다. 예를 들어, *DLFM_BACKUP_TARGET* 레지스트리 값을 TSM으로 설정하여 데이터 링크 파일 관리 프로그램을 시작하고, 그 레지스트리 값을 디스크 위치로 변경할 경우, 새로 아카이브된 모든 파일은 디스크상의 새 위치에 저장됩니다. 이전에 TSM으로 아카이브된 파일들은 새 디스크 위치로 이동되지 않습니다.
 - b. 기본 TSM 관리 클래스를 대체하려면 *DLFM_TSM_MGMTCLASS*라고 하는 새 레지스트리 변수가 있습니다. 이 레지스트리 변수를 설정되지 않은 상태로 두면, 기본 TSM 관리 클래스가 사용됩니다.
9. **dlfm stop** 명령을 입력하여 Data Links File Manager를 중지하십시오.
 10. **dlfm start** 명령을 입력하여 Data Links File Manager를 시작하십시오.

Solaris 운영 환경에서 DB2 Data Links Manager용 백업 방법 선택

디스크 복사 및 XBSA 이외에도, 데이터 링크 서버에 상주하는 파일들을 백업하기 위해 Tivoli Storage Manager(TSM)를 사용할 수도 있습니다.

아카이브 서버로 Tivoli Storage Manager를 사용하려면 다음을 수행하십시오.

1. Data Links 서버에서 Tivoli Storage Manager를 설치하십시오. 자세한 내용은 Tivoli Storage Manager 제품 문서를 참조하십시오.
2. Tivoli Storage Manager 서버를 사용하여 Data Links 서버 클라이언트 응용프로그램을 등록하십시오. 자세한 내용은 Tivoli Storage Manager 제품 문서를 참조하십시오.
3. 다음 환경 변수를 Data Links Manager 관리자의 db2profile 또는 db2cshrc 스크립트 파일에 추가하십시오.

```
(Bash, Bourne 또는 Korn 셸의 경우)
export DSMI_DIR=/opt/tivoli/tsm/client/api/bin
export DSMI_CONFIG=$HOME/tsm/dsm.opt
export DSMI_LOG=$HOME/dldump
export PATH=$PATH:/opt/tivoli/tsm/client/api/bin
```

```
(C 셸의 경우)
setenv DSMI_DIR /opt/tivoli/tsm/client/api/bin
setenv DSMI_CONFIG ${HOME}/tsm/dsm.opt
setenv DSMI_LOG ${HOME}/dldump
setenv PATH=${PATH}:/opt/tivoli/tsm/client/api/bin
```

4. dsm.sys TSM 시스템 옵션 파일이 /opt/tivoli/tsm/client/api/bin 디렉토리에 있는지 확인하십시오.
5. dsm.opt TSM 사용자 옵션 파일이 *INSTHOME*/tsm 디렉토리에 있는지 확인하십시오. *INSTHOME*은 Data Links Manager 관리자의 홈 디렉토리입니다.
6. /opt/tivoli/tsm/client/api/bin/dsm.sys Tivoli Storage Manager 시스템 옵션 파일에서 *PASSWORDACCESS* 옵션을 generate로 설정하십시오.
7. 처음으로 Data Links File Manager를 시작하기 전에 generate 옵션으로 TSM 암호를 등록하십시오. 이 방법을 사용하면 Data Links File Manager가 TSM 서버에 대한 연결을 초기화할 때 암호를 제공하지 않아도 됩니다. 자세한 내용은 해당되는 TSM 제품 문서를 참조하십시오.
8. *DLFM_BACKUP_TARGET* 레지스트리 변수를 TSM으로 설정하십시오. *DLFM_BACKUP_DIR_NAME* 레지스트리 변수의 값은 이 경우에 무시됩니다. 이렇게 하면 Tivoli Storage Manager 백업 옵션이 활성화됩니다.

주:

- a. 수행 시 *DLFM_BACKUP_TARGET* 레지스트리 변수의 설정을 TSM과 디스크 사이에 변경할 경우, 아카이브된 파일이 새로 지정된 아카이브 위치로 이동되지 않는다는 점에 유의해야 합니다. 예를 들어,
DLFM_BACKUP_TARGET 레지스트리 값을 TSM으로 설정하여 데이터 링

크 파일 관리 프로그램을 시작하고, 그 레지스트리 값을 디스크 위치로 변경할 경우, 새로 아카이브된 모든 파일은 디스크상의 새 위치에 저장됩니다. 이전에 TSM으로 아카이브된 파일들은 새 디스크 위치로 이동되지 않습니다.

- b. 기본 TSM 관리 클래스를 대체하려면 DLFM_TSM_MGMTCLASS라고 하는 새 레지스트리 변수가 있습니다. 이 레지스트리 변수를 설정되지 않은 상태로 두면, 기본 TSM 관리 클래스가 사용됩니다.

9. **dlfm stop** 명령을 입력하여 Data Links File Manager를 중지하십시오.

10. **dlfm start** 명령을 입력하여 Data Links File Manager를 시작하십시오.

Windows NT에서 DB2 Data Links Manager에 대한 백업 메소드 선택

복구에 대해 정의된 DATALINK 컬럼이 있는 테이블로 DATALINK 값이 삽입될 때마다, Data Links 서버에서 해당되는 DATALINK 파일은 아카이브 서버로 백업되도록 스케줄이 정해집니다. 현재, 디스크 복사(기본 메소드) 및 Tivoli Storage Manager는 아카이브 서버로의 파일 백업에 대해 지원되는 두 가지 옵션입니다. Windows NT용 DB2 Data Links Manager의 차후 릴리스에서는 다른 벤더의 백업 미디어와 소프트웨어가 지원됩니다.

디스크 복사(기본 메소드)

DB2 서버에 **backup** 명령이 입력될 경우, command is entered on 데이터베이스의 링크된 파일들은 Data Links 서버에서 DLFM_BACKUP_DIR_NAME 환경 변수에 지정된 디렉토리로 백업됩니다. 이 변수에 대한 기본 값은 c:\dlfmbackup입니다. 여기서, c:\는 Data Links Manager 백업 설치 드라이브를 나타냅니다.

이 변수를 c:\dlfmbackup으로 설정하려면, 다음 명령을 입력하십시오.

```
db2set -g DLFM_BACKUP_DIR_NAME=c:\dlfmbackup
```

DLFM_BACKUP_DIR_NAME 환경 변수에 지정된 위치는 Data Links Filesystem Filter를 사용하고 백업 파일에 대해 사용자가 지정한 디렉토리에 필수 공간이 사용 가능한 파일 시스템에 있어야 합니다.

또한, 다음 명령을 입력하여 DLFM_BACKUP_TARGET 변수가 LOCAL로 설정되도록 하십시오.

```
db2set -g DLFM_BACKUP_TARGET=LOCAL
```

이 변수를 설정 또는 변경한 후, **dlfm stop** 및 **dlfm start** 명령을 사용하여 Data Links File Manager를 중지하고 재시작하십시오.

Tivoli Storage Manager

아카이브 서버로 Tivoli Storage Manager를 사용하려면 다음을 수행하십시오.

1. Data Links 서버에서 Tivoli Storage Manager를 설치하십시오. 자세한 내용은 Tivoli Storage Manager 제품 문서를 참조하십시오.
2. Tivoli Storage Manager 서버를 사용하여 Data Links 서버 클라이언트 응용프로그램을 등록하십시오. 자세한 내용은 Tivoli Storage Manager 제품 문서를 참조하십시오.
3. 시작을 클릭한 후 설정 → 제어판 → 시스템을 선택하십시오. 시스템 등록 정보 창이 열립니다. 환경 탭을 선택한 후 다음 환경 변수와 해당되는 값을 입력하십시오.

변수	값
DSMI_DIR	c:\tsm\baclient
DSMI_CONFIG	c:\tsm\baclient\dsm.opt
DSMI_LOG	c:\tsm\dldump

4. dsm.sys TSM 시스템 옵션 파일이 c:\tsm\baclient 디렉토리에 있는지 확인하십시오.
5. dsm.opt TSM 사용자 옵션 파일이 c:\tsm\baclient 디렉토리에 있는지 확인하십시오.
6. c:\tsm\baclient\dsm.sys Tivoli Storage Manager 시스템 옵션 파일에서 *PASSWORDACCESS* 옵션을 generate로 설정하십시오.
7. 처음으로 Data Links File Manager를 시작하기 전에 generate 옵션으로 TSM 암호를 등록하십시오. 이 방법을 사용하면 Data Links File Manager가 TSM 서버에 대한 연결을 초기화할 때 암호를 제공하지 않아도 됩니다. 자세한 내용은 해당되는 TSM 제품 문서를 참조하십시오.
8. 다음 명령을 사용하여 DLFM_BACKUP_TARGET 환경 변수를 TSM으로 설정하십시오.

```
db2set -g DLFM_BACKUP_TARGET=TSM
```

DLFM_BACKUP_DIR_NAME 환경 변수의 값은 이 경우에 무시됩니다. 이렇게 하면 Tivoli Storage Manager 백업 옵션이 활성화됩니다.

주:

- a. 수행 시 DLFM_BACKUP_TARGET 환경 변수의 설정을 TSM과 LOCAL 사이에 변경할 경우, 아카이브된 파일이 새로 지정된 아카이브 위치로 이동되지 않는다는 점에 유의해야 합니다. 예를 들어, DLFM_BACKUP_TARGET 환경 변수를 TSM으로 설정하여 데이터 링크 파일 관리 프로그램을 시작하고 값을 LOCAL로 변경할 경우, 새로 아카이브된 모든 파일은 디스크의 새 위치에 저장됩니다. 이전에 TSM으로 아카이브된 파일들은 새 디스크 위치로 이동되지 않습니다.

b. 기본 TSM 관리 클래스를 대체하려면 `DLFM_TSM_MGMTCLASS` 라고 하는 새 환경 변수가 있습니다. 이 변수를 설정되지 않은 상태로 두면, 기본 TSM 관리 클래스가 사용됩니다.

9. `dlfm stop` 명령을 입력하여 Data Links File Manager를 중지하십시오.

10. `dlfm start` 명령을 입력하여 Data Links File Manager를 시작하십시오.

AIX에서 저널 파일 시스템 백업

책에서는 Data Links Manager를 중지해야 하고 파일 시스템을 오프라인 백업해야 한다고 언급하고 있습니다. 더 높은 사용 가능성을 필요로 하는 사용자의 경우에는 이 Data Links Manager를 중지하는 요구사항을 제거하는 다음의 접근 방식이 제안됩니다.

1. CLI 소스 파일 `quiesce.c` 및 셸 스크립트 `online.sh`에 액세스하십시오. 이 파일들은 `/samples/dlfm` 디렉토리에 있습니다.

2. `quiesce.c`를 컴파일하십시오.

```
xlc -o quiesce -L$HOME/sqlllib/lib -I$HOME/sqlllib/include -c quiesce.c
```

3. 루트로 DLFS 파일 시스템이 있는 노드에서 스크립트를 수행하십시오.

셸 스크립트 `online.sh`는 Data Link Manager로 등록된 각 데이터베이스에 대해 Data Link Manager 노드에 카탈로그 항목을 가지고 있는 것으로 가정합니다. 또한, `/etc/filesystems`은 DLFS 파일 시스템에 대해 완전한 항목을 가지고 있는 것으로 가정합니다. 셸 스크립트는 다음을 수행합니다.

- Data Links Manager로 등록된 데이터베이스에서 모든 테이블을 Quiesce합니다. 그러면, 새로운 활동이 중지됩니다.
- 파일 시스템을 마운트 해제하고 읽기 전용 파일 시스템으로 다시 마운트합니다.
- 파일 시스템 백업을 수행합니다.
- 파일 시스템을 마운트 해제하고 읽기/쓰기 파일 시스템으로 다시 마운트합니다.
- DB2 테이블을 재설정합니다. 즉, Quiesce 상태를 벗어나게 합니다.

스크립트는 다음과 같이 환경에 적합하도록 수정해야 합니다.

1. 백업 명령을 선택하고 스크립트의 `do_backup` 함수에 기록하십시오.

2. 스크립트 내에서 다음 환경 변수를 설정하십시오,

- `DLFM_INST`: DLFM 인스턴스 이름으로 설정하십시오.
- `PATH_OF_EXEC`: "quiesce" 실행 파일 경로로 설정하십시오.

다음과 같이 스크립트를 호출하십시오.

```
online.sh <filesystem_name>
```

Windows NT에서 데이터 링크의 관리자 그룹 특권

Windows NT에서, 관리자 그룹에 속하는 사용자는 DataLinks를 사용하여 링크된 파일에 관해서는 UNIX의 루트 사용자와 대부분의 함수에 대해 동일한 특권을 소유하게 됩니다. 다음 표는 둘을 비교합니다.

작업	Unix(루트)	Windows NT(관리자)
이름 바꾸기	예	예
토큰 없이 파일 액세스	예	예
삭제	예	아니오(아래 주 참조)
갱신	예	아니오(아래 주 참조)

주: NTFS는 읽기 전용 파일에 대해 이 작업들을 허용하지 않습니다. 관리자 사용자는 파일에 대한 쓰기 사용 권한을 이용하여 이 작업들을 수행할 수 있습니다.

Data Links File System Filter(DLFF) 설치에 대한 로깅 최소화

d_lfs_cfg 파일을 변경하여 Data Links File System Filter(DLFF) 설치에 대한 로깅을 최소화할 수 있습니다. d_lfs_cfg 파일은 드라이버 및 구성 매개변수를 로드하기 위해 strload 루틴으로 전달됩니다. 파일은 /usr/lpp/db2_07_01/cfg/ 디렉토리에 있습니다. 기호 링크를 통해, 파일을 /etc 디렉토리에서 찾을 수도 있습니다. d_lfs_cfg 파일은 다음 형식이 있습니다.

```
d 'driver-name' 'vfs number' 'dlfm id' 'global message priority'  
'global module priority' - 0 1
```

여기서,

d d 매개변수는 드라이버가 로드된다는 것을 지정합니다.

driver-name

driver-name은 로드될 드라이버의 전체 경로입니다. 예를 들어, DB2 버전 2의 전체 경로는 /usr/lpp/db2_07_01/bin/dlfsdrv입니다. 드라이버의 이름은 dlfsdrv입니다.

vfs number

/etc/vfs의 DLFS에 대한 vfs 항목입니다.

dlfm id

DataLinks Manager 관리자의 사용자 ID입니다.

global message priority

전역 메시지 우선순위입니다

global module priority

전역 모듈 우선순위입니다

0 1은 이 드라이버에 대한 비복제 노드 작성을 위한 부속 번호입니다. 노드 이름은 복제된 드라이버 노드 이름에 부 번호를 첨부하여 작성됩니다. 다섯 개의 부속 번호 이상은 제공될 수 없습니다(0-4).

실제 예는 다음과 같습니다.

```
d /usr/lpp/db2_07_01/bin/dlfsdrv 14,208,255,-1 - 0 1
```

로그되는 메시지는 전역 메시지 우선순위와 전역 모듈 우선순위에 따라 다릅니다. 로깅을 최소화하기 위해, 전역 메시지 우선순위 값을 변경할 수 있습니다.

네 개의 메시지 우선순위 값을 사용할 수 있습니다.

```
#define LOG_EMERGENCY      0x01
#define LOG_TRACING        0x02
#define LOG_ERROR          0x04
#define LOG_TROUBLESHOOT  0x08
```

DLFF의 메시지 대부분의 메시지 우선순위는 LOG_TROUBLESHOOT입니다. 다음은 몇 가지 대체 구성 예입니다.

비상 메시지와 오류 메시지가 필요할 경우, dlfs_cfg 구성 파일에서 전역 메시지 우선순위를 5(1+4)로 설정하십시오.

```
d /usr/lpp/db2_07_01/bin/dlfsdrv 14,208,5,-1 - 0 1
```

오류 메시지만 필요할 경우, 글로벌 메시지 우선순위를 4로 설정하십시오.

```
d /usr/lpp/db2_07_01/bin/dlfsdrv 14,208,4,-1 - 0 1
```

DLFS에 대한 로깅이 필요하지 않으면, 글로벌 메시지 우선순위를 0으로 설정하십시오.

```
d /usr/lpp/db2_07_01/bin/dlfsdrv 14,208,0,-1 - 0 1
```

설치 후 메시지 로깅

설치 후에 비상, 오류 및 문제 해결 메시지를 로그하려면, dlfs_cfg 파일을 수정해야 합니다. dlfs_cfg 파일은 /usr/lpp/db2_07_01/cfg 디렉토리에 있습니다. 전역 메시지 우선순위는 255(최대 우선순위) 또는 13(8+4+1)으로 설정해야 합니다. 우선순위를 13(8+4+1)으로 설정하면 비상, 오류 및 문제 해결 정보를 로그합니다.

전역 메시지 우선순위를 설정한 후, DLFS 필터 파일 시스템을 마운트 해제하고 dlfsdrv 드라이버를 재로드하여 새 우선순위 값이 로드 시간에 설정되도록 하십시오. dlfsdrv 드라이버를 재로드한 후, DLFS 필터 파일 시스템을 다시 마운트해야 합니다.

주: dlfs_cfg의 설정은 dlfs_cfg 파일을 다시 변경하기 전까지 dlfsdrv 드라이버 로드 때마다 유지됩니다.

SMIT May Remove Additional Filesets를 사용하여 DLFM 구성요소 설치 제거

Data Links Manager가 설치되어 있는 AIX 머신에서 DB2(버전 5, 6 또는 7)를 설치 제거하기 전에, 다음 단계를 따르십시오.

1. 루트로서, 다음 명령을 사용하여 /etc/vfs를 복사하십시오.
2. DB2를 설치 제거하십시오.
3. 루트로서, /etc/vfs를 단계 1에서 작성된 백업으로 바꾸십시오.

```
cp -p /etc/vfs /etc/vfs.bak
```

```
cp -p /etc/vfs.bak /etc/vfs
```

호스트 이름을 시작/판별하기 전에

각 DB2 서버와 Data Links 서버의 이름을 판별해야 합니다. 설치를 검증하려면 이러한 호스트 이름을 알아야 합니다. DB2 Data Links File Manager에 연결할 때, DB2 UDB 서버는 내부적으로 다음 정보를 DLFM에 보냅니다.

- 데이터베이스 이름
- 인스턴스 이름
- 호스트 이름

그러면 DLFM은 이 정보를 내부 테이블과 비교하여 연결 허용 여부를 결정합니다. 데이터베이스 이름, 인스턴스 이름 및 호스트 이름의 조합이 `dlfm add_db` 명령을 사용하여 등록된 경우에만 연결을 허용합니다. `dlfm add_db` 명령에 사용된 호스트 이름은 DB2 UDB 서버가 내부적으로 보낸 호스트 이름과 정확히 일치해야 합니다.

호스트 이름을 다음과 같이 사용하십시오.

1. DB2 서버에서 `hostname` 명령을 입력하십시오. 예를 들어, 이 명령이 `db2server`를 리턴할 것입니다.
2. 플랫폼에 따라, 다음 중 하나를 수행하십시오.
 - AIX에서는, `host db2server` 명령을 입력하십시오. 여기서 `db2server`는 이전 단계에서 구해진 이름입니다. 이 명령은 다음과 유사한 출력을 리턴해야 합니다.

```
db2server.services.com is 9.11.302.341, Aliases: db2server
```

- Windows NT에서는, `nslookup db2server` 명령을 입력하십시오. 여기서 `db2server`는 이전 단계에서 구해진 이름입니다. 이 명령은 다음과 유사한 출력을 리턴해야 합니다.

```
Server: dnsserv.services.com
Address: 9.21.14.135
Name: db2server.services.com
Address: 9.21.51.178
```


- Solaris에서는, `cat /etc/hosts | grep 'hostname'`을 입력하십시오. `/etc/hosts`에서 도메인 이름 없이 호스트 이름을 지정할 경우, 명령은 다음과 유사한 출력을 리턴해야 합니다.

```
9.112.98.167 db2server loghost
```

도메인 이름과 함께 호스트 이름을 지정할 경우, 명령은 다음과 유사한 출력을 리턴해야 합니다.

```
9.112.98.167 db2server.services.com loghost
```

`dlfm add_db` 명령을 사용하여 DB2 UDB 데이터베이스를 등록할 때, 호스트 이름에 `db2server.services.com`을 사용하십시오. `dlfm add_db` 명령에서 다른 별명을 사용하면 DLFM과의 DB2 서버의 내부 연결이 실패합니다.

Data Links 서버는 DB2 "`add datalinks manager for database database_alias using node hostname port port_number`" 명령을 사용하여 DB2 데이터베이스에 등록됩니다.

`hostname`은 Data Links 서버의 이름입니다. Data Links 서버의 별명을 이 명령에서 사용할 수 있습니다. 이 Data Links 서버를 참조하는 DATALINK 값은 URL 값에 호스트 이름을 지정해야 합니다. 즉, URL 값을 DATALINK 컬럼에 지정할 때 "`add datalinks manager`" 명령에 사용된 정확한 이름을 사용해야 합니다. 다른 별명을 사용하면 SQL문 실패를 초래합니다.

Data Links File Manager 작업: DB2 데이터베이스에서 DB2 Data Links Manager 삭제 후 정리

`DROP DATALINKS MANAGER` 명령을 사용하여 DB2 Data Links Manager 삭제 시, 명령 자체는 DB2 Data Links Manager에 대한 해당 정보를 정리하지 않습니다. 사용자가 명시적으로 데이터베이스에 링크된 파일의 링크해제 및 백업 정보의 가비지 수집을 실시할 수 있습니다. `dlfm drop_dlm` 명령을 사용하여 이를 수행합니다. 이 명령은 특정 데이터베이스에 대한 모든 정보의 비동기 삭제를 시작합니다. 이 명령이 성공하려면 DB2 Data Links Manager가 수행 중이어야 합니다. 반드시 DB2 Data Links Manager를 삭제한 후에 이 명령을 사용해야 합니다. 그렇지 않으면, DB2 Data Links Manager에 관한 중요 정보가 유실되어 복구할 수 없습니다.

특정 데이터베이스에 대해 링크해제 처리 및 백업 정보의 가비지 수집을 실시하려면, 다음을 수행하십시오.

1. DB2 Data Links Manager 관리자로서 시스템에 로그인하십시오.
2. 다음 명령을 실행하십시오.

```
dlfm drop_dlm database instance hostname
```

여기서,

database는 원격 DB2 UDB 데이터베이스의 이름입니다.
instance는 데이터베이스가 상주하는 인스턴스입니다.
hostname은 데이터베이스가 상주하는 DB2 UDB 서버의
호스트 이름입니다.

3. 로그오프하십시오.

이 명령에 사용되는 문맥을 보여주는 완전한 사용 시나리오는, *Command Reference*를 참조하십시오.

새 오류 코드가 이 명령에 대해 작성되었습니다(『DLFM1001E (새 오류 메시지)』 참조).

DLFM1001E (새 오류 메시지)

DLFM1001E: drop_dlm 처리에 오류.

원인:

Data Links Manager가 지정된 데이터베이스에 대해 링크해제 및 가비지 수집 처리를 시작할 수 없습니다. 다음 중 하나가 이 오류의 원인일 수 있습니다.

- Data Links Manager가 수행 중이 아닙니다.
- 명령에 유효하지 않은 데이터베이스, 인스턴스 및 호스트 이름이 지정되었습니다.
- Data Links Manager의 구성요소 서비스 중 하나에서 실패했습니다.

조치:

다음 단계를 수행하십시오.

1. Data Links Manager가 수행 중인지 확인하십시오. 이미 수행 중이 아니라면 Data Links Manager를 시작하십시오.
2. 데이터베이스, 인스턴스 및 호스트 이름의 조합이 등록된 데이터베이스를 식별하는지 확인하십시오. Data Links Manager에서 "dlfm list registered databases" 명령을 사용하여 이를 확인할 수 있습니다.
3. 오류가 계속 발생하면, db2diag.log 파일의 정보를 참조하여 구성요소 서비스(예를 들어, 연결 관리 서비스, 트랜잭션 관리 서비스, 등등)가 실패했는지 확인하십시오. db2diag.log에서 오류 코드를 보고, 오류 코드 아래에 제안된 적절한 조치를 취하십시오.

DLFM 설정 구성 파일 옵션

dlfm setup dlfm.cfg 옵션이 제거되었습니다. 문서에서 이에 대한 언급은 무시하십시오.

AIX에서 Data Links/DFS Script dmapp_prestart 수행 시 오류

```
/usr/sbin/cfgdmepl -a "/usr/lib/drivers/dmlfs.ext"
```

이 Data Links/DFS Script dmapp_prestart를 수행할 때 위의 명령이 리턴 코드 1로 실패하면, DFS 3.1 ptfset1을 설치하여 cfgdmepl을 수정하십시오.

Tivoli Space Manager와 Data Links 통합

DB2 Data Links Manager는 Tivoli Space Manager의 장점을 활용할 수 있습니다. Tivoli Space Manager HSM(Hierarchical Storage Manager) 클라이언트 프로그램은 자동으로 적합한 파일을 저장영역으로 이주시켜 지역 파일 시스템에서 특정 레벨의 여유 공간을 관리합니다. 이 프로그램은 이주된 파일을 액세스할 때 자동으로 다시 호출되어 사용자가 특정 파일을 이주하고 다시 호출하도록 허용합니다.

파일 시스템 공간을 규칙적으로 관리해야 하는 제3의 저장영역으로 정기적으로 이동해야 하는 대형 파일이 있는 파일 시스템을 가지고 있는 사용자는 이 새로운 기능으로 편리하게 작업할 수 있습니다. Tivoli Space Manager는 현재 많은 사용자를 위해 제3의 저장영역을 관리할 수 있는 방법을 제공합니다. Tivoli Space Manager의 새로운 DB2 Data Links Manager 지원을 통해 DATALINK 파일 공간을 융통성있게 관리할 수 있습니다. Tivoli Space Manager를 사용하면 DB2 Data Links Manager 파일 시스템에 저장될 모든 파일에 대해 충분한 공간을 미리 할당하지 않아도, 파일 시스템을 사용할 때 실수로 가득 채울 위험없이 일정 기간 이상 Data Links 관리 파일 시스템 할당을 조정할 수 있습니다.

Data Links 및 HSM 지원을 파일 시스템에 모두 추가

HSM(Hierarchical Storage Management)으로 파일 시스템을 등록할 때 먼저 HSM으로 등록한 다음, DataLinks File Manager로 등록하십시오.

1. "dsmmigfs add /fs" 명령을 사용하여 HSM으로 등록하십시오.
2. "dlfmfsmd /fs" 명령을 사용하여 DLM으로 등록하십시오.

Data Links 파일 시스템 지원은 다음 항목들을 통해 HSM 파일 시스템용 /etc/filesystems의 절에 영향을 줍니다.

```
vfs = d1fs
mount = false
options = rw,Basefs=fsm
nodename = -
```

기존 HSM 파일 시스템에 Data Links 지원 추가

"dlfmfsmd /fs" 명령을 사용하여 DLM으로 등록하십시오.

기존 Data Links 파일 시스템에 HSM 지원 추가

1. "dsmmigfs add /fs" 명령을 사용하여 HSM으로 등록하십시오.
2. "dlfmfsmd /fs" 명령을 사용하여 DLM으로 등록하십시오.

Data Links-HSM 파일 시스템에서 Data Links 지원 제거

"dlfmfsmd -j /fs" 명령을 사용하여 Data Links 지원을 제거하십시오.

Data Links-HSM 파일 시스템에서 HSM 지원 제거

1. "dsmmigfs remove /fs" 명령을 사용하여 HSM 지원을 제거하십시오.
2. "dlfmfsmd -j /fs" 명령을 사용하여 Data Links 지원을 제거하십시오.
3. "dlfmfsmd /fs" 명령을 사용하여 DLM으로 등록하십시오.

Data Links-HSM 파일 시스템에서 Data Links 및 HSM 지원 제거

1. "dsmmigfs remove /fs" 명령을 사용하여 HSM 지원을 제거하십시오.
2. "dlfmfsmd -j /fs" 명령을 사용하여 Data Links 지원을 제거하십시오.

제한사항 및 한계

이 기능은 현재 AIX에만 지원됩니다.

선택적 이주(dsmmigrate) 및 FC(Read permission DB) 링크 파일의 재호출은 루트 사용자만 수행할 수 있습니다.

선택적 이주는 Read Permission DB 파일이 DataLink Manager Administrator(dlfm)인 경우, 파일 소유자만 수행할 수 있습니다. 이들 파일에 액세스하려면 호스트 데이터베이스에 있는 토큰이 필요합니다. 토큰이 필요하지 않은 사용자는 "루트" 사용자 뿐입니다. "루트" 사용자가 선택적 이주 및 Read Permission DB 파일의 재호출을 수행하는 것이 더 쉽습니다. dlfm 사용자는 처음에만 유효한 토큰을 사용하여 FC 파일을 이주할 수 있습니다. 재호출 후 두 번째로 이주를 시도하면 오류 메시지 "ANS1028S 내부 프로그램 오류. 서비스 대표에게 문의하십시오."가 나타나면서 조작이 실패합니다. FC 파일에서 루트 사용자가 아닌 사용자가 dsmmigrate를 수행하면 성공하지 못합니다. 일반적으로 파일 서버의 파일에 관리자가 액세스하므로 이러한 제한은 큰 문제가 되지 않습니다.

stat 및 statfs 시스템 호출은 dlfs가 fsm에 마운트된 경우에도 Vfs-type을 dlfs가 아니라 fsm으로 표시합니다.

위의 동작은 dsmrealld 디먼의 일반적인 기능으로 파일 시스템에서 statfs를 수행하여 Vfs-type이 fsm인지 아닌지 확인합니다.

최소한의 inode를 가진 파일이 FC(Read permission DB) 링크인 경우, "dsmls"는 출력을 표시하지 않습니다.

dsmls 명령은 ls 명령과 비슷하며 TSM이 관리하는 파일들을 나열합니다. 사용자 조치는 필요하지 않습니다.

AIX용 DB2 Data Links Manager 설치 및 구성

공통적인 설치 고려사항

DB2 File Manager 버전 5.2에서 DB2 Data Links Manager 버전 7로 이주

단계 3의 정보가 잘못되었습니다. 3 단계는 다음과 같아야 합니다.

"3. DLFM 관리자로서, `/usr/lpp/db2_07_01/adm/db2dlmmg` 명령을 수행하십시오.

설치 및 구성 보충 설명서

UNIX 운영 체제에 DB2 클라이언트 설치

HP-UX 커널 구성 매개변수

HP-UX 커널 매개변수 설정의 권장사항에서 **msgmbn** 및 **msgmax**를 65535 이상으로 설정하라는 것은 잘못된 것입니다. 두 매개변수는 정확하게 65535로 설정되어야 합니다.

사용자 자신의 응용프로그램 수행

Run-time Client를 사용한 데이터베이스 유틸리티 바인딩

Run-time Client는 데이터베이스 유틸리티(가져오기, 내보내기, 재구성, 명령행 처리기)와 DB2 CLI 바인드 파일을 이 데이터베이스와 사용하기 전에 각 데이터베이스에 바인드하는 데 사용할 수 없습니다. 그 대신 DB2 Administration Client나 DB2 응용 프로그램 개발 클라이언트를 사용해야 합니다.

이들 데이터베이스 유틸리티와 DB2 CLI 바인드 파일을 해당 데이터베이스와 함께 사용하려면 각 데이터베이스에 바인드해야 합니다. 네트워크 환경에서, 다른 운영 체제에서 수행되거나 DB2의 버전이나 서비스 레벨이 다른 여러 클라이언트를 사용하는 경우, 각 운영 체제와 DB2 버전 조합에 대해 한번씩 유틸리티를 바인드해야 합니다.

ODBC를 사용한 DB2에 대한 UNIX 클라이언트 액세스

"사용자 자신의 응용프로그램 수행"에서는 ODBC 클라이언트 응용프로그램이나 ODBC SDK와 함께 ODBC 드라이버 관리자를 설치할 경우, `odbcinst.ini`를 갱신해야 한다고 설명하고 있습니다. 이는 부분적으로 잘못된 정보입니다. Merant ODBC 드라이버 관리자 제품을 설치할 경우에는 `odbcinst.ini`를 갱신할 필요가 없습니다.

여러 데이터 소스에 액세스하기 위해 연합 시스템 설정

연합 시스템

DB2 연합 시스템은 특별한 유형의 분산 데이터베이스 관리 시스템(DBMS)입니다. 연합 시스템을 사용하면 Oracle, Sybase 및 Microsoft SQL Server 등 다른 DBMS에 있는 데이터를 조회하고 검색할 수 있습니다. SQL문은 하나의 명령문에서 각각의 데이터베이스나 여러 DBMS를 참조할 수 있습니다. 예를 들어, DB2 Universal Database 테이블, Oracle 테이블 및 Sybase 뷰에 있는 데이터를 조인할 수 있습니다. 지원되는

DBMS는 Oracle, Sybase, Microsoft SQL Server(AIX 및 Windows NT용) 및 DB2 Universal Database 계열의 구성원(OS/390용 DB2, AS/400용 DB2 및 Windows용 DB2)입니다.

DB2 연합 시스템은 하나 이상의 데이터 소스 및 DB2 인스턴스(연합 데이터베이스가 될 데이터베이스)가 있는 서버로 구성됩니다. 연합 데이터베이스에는 데이터 소스 및 그 특성을 식별하는 카탈로그 항목이 있습니다. 데이터 소스는 DBMS와 데이터로 구성됩니다. DB2 Universal Database에는 이들 데이터 소스에 액세스할 때 사용하는 랩퍼라는 프로토콜이 있습니다. 랩퍼는 연합 서버가 데이터 소스와 통신하고 데이터 소스에서 데이터를 검색할 때 사용하는 메커니즘입니다. 별명은 데이터 소스에 있는 테이블 및 뷰를 참조할 때 사용합니다. 응용프로그램은 다른 DB2 데이터베이스와 마찬가지로 연합 데이터베이스에 연결합니다. 사용하는 랩퍼는 DB2 Universal Database가 수행되는 플랫폼에 따라 달라집니다.

연합 시스템이 설정된 후 데이터 소스의 정보는 하나의 대형 데이터베이스에 있는 것처럼 액세스될 수 있습니다. 사용자 및 응용프로그램은 하나의 연합 데이터베이스에 조회를 보내면 이 데이터베이스가 데이터 소스에서 데이터를 검색합니다.

DB2 연합 시스템이 작동되는 데는 제한이 있습니다. 분산 요청(DR)은 DB2 버전 7의 읽기 전용 조작으로만 제한됩니다. 또한 별명에 대한 유틸리티 조작(LOAD, REORG, REORGCHK, IMPORT, RUNSTATS 등)을 실행할 수 없습니다. 그러나 DDL 및 DML 명령문을 데이터 소스와 연관된 SQL Dialect를 사용하여 직접 DBMS로 제출하는 통과 기능을 사용할 수 있습니다.

제한사항

버전 7.2의 새 랩퍼(Linux 및 Solaris의 Oracle, AIX 및 Solaris의 Sybase, NT 및 AIX의 Microsoft SQL Server 등)는 FixPak 4에서는 제공되지 않습니다. DB2 Relational Connect 버전 7.2를 구입해야 합니다.

DB2 Relational Connect 설치

이 절에는 연합 시스템 서버로 사용할 서버에 DB2 Relational Connect를 설치하는 과정이 나와 있습니다.

Windows NT 서버에 DB2 Relational Connect 설치

Windows NT 연합 서버에 DB2 Relational Connect를 설치하기 전에 다음을 확인하십시오.

- DB2 Universal Database Enterprise Edition이나 DB2 Universal Database Enterprise -- Extended Edition이 연합 서버에 설치되어 있는지 확인합니다. 분산 요청에 DB2 계열 데이터베이스를 포함하려면 DB2 Universal Database를 설치할 때 **DB2용 Distributed Join** 데이터 소스 옵션을 선택해야 합니다. 이 옵션이 구현되었는지 검증하려면 FEDERATED 매개변수가 YES로 설정되어 있는지 확인합니다. **GET**

DATABASE MANAGER CONFIGURATION 명령을 실행하여 이 설정을 확인할 수 있는데, 이 명령은 모든 매개변수 및 현재 설정을 표시합니다.

- 연합 서버에 데이터 소스에 대한 클라이언트 소프트웨어(예: Sybase Open Client)를 설치했는지 확인합니다.

1. 설치하기 위해 작성한 사용자 계정으로 시스템에 로그인합니다.
2. 설치 프로그램이 필요한 파일을 갱신할 수 있도록 수행 중인 프로그램을 종료합니다.
3. 설치 프로그램을 호출합니다. 설치 프로그램은 자동 또는 수동으로 호출할 수 있습니다. 설치 프로그램이 자동으로 시작되지 않거나 다른 언어로 설치하려면 설치 프로그램을 수동으로 호출합니다.

- 설치 프로그램을 자동으로 호출하려면 DB2 Relational Connect CD를 드라이브에 넣습니다. 자동 수행 기능이 자동으로 설치 프로그램을 시작합니다. 시스템 언어가 결정되고 이 언어에 해당하는 설치 프로그램이 시작됩니다.

- 설치 프로그램을 수동으로 호출하려면 다음을 수행하십시오.

- a. 시작을 누르고 실행 옵션을 선택합니다.
- b. 열기 필드에 다음 명령을 입력합니다.

```
x:\setup /i language
```

여기서,

x: CD-ROM 드라이브를 나타냅니다.

language

사용자 언어의 국가 코드를 나타냅니다(예: 영어는 EN).

- c. 확인을 클릭하십시오.

설치 런치패드가 열립니다.

4. 설치를 눌러 설치 프로세스를 시작합니다.

5. 설치 프로그램의 프롬프트에 따릅니다.

프로그램이 완료되면 DB2 Relational Connect가 다른 DB2 제품과 함께 설치 디렉토리에 설치됩니다.

AIX, Linux 및 Solaris 운영 환경 서버에 DB2 Relational Connect 설치

DB2 Relational Connect를 AIX, Linux 및 Solaris 운영 환경 연합 서버에 설치하기 전에 다음을 수행하십시오.

- DB2 Universal Database Enterprise Edition이나 DB2 Universal Database -- Extended Edition이 연합 서버에 설치되어 있는지 확인합니다. 분산 요청에 DB2 계열 데이터베이스를 포함하려면 DB2 Universal Database를 설치할 때 **DB2용 Distributed Join** 데이터 소스 옵션을 선택해야 합니다. 이 옵션이 구현되었는지 검

증하려면 FEDERATED 매개변수가 YES로 설정되어 있는지 확인합니다. **GET DATABASE MANAGER CONFIGURATION** 명령을 실행하여 이 설정을 확인할 수 있는데, 이 명령은 모든 매개변수 및 현재 설정을 표시합니다.

- 연합 서버에 데이터 소스에 대한 클라이언트 소프트웨어(예: Sybase Open Client)를 설치했는지 확인합니다.

DB2 Relational Connect를 AIX, Linux 및 Solaris 운영 환경 서버에 설치하려면 db2setup 유틸리티를 사용하십시오.

1. 루트 권한이 있는 사용자로 로그인합니다.
2. DB2 제품 CD-ROM을 넣고 마운트합니다. CD-ROM을 마운트하는 방법에 대한 자세한 정보는 AIX용 빠른 시작을 참조하십시오.
3. **cd *lcdrom*** 명령을 입력하여 CD-ROM이 마운트된 디렉토리로 변경합니다. 여기서, *cdrom*은 제품 CD-ROM의 마운트 지점입니다.
4. **./db2setup** 명령을 입력합니다. 잠시 뒤 DB2 설치 유틸리티 창이 열립니다.
5. 설치를 선택합니다. DB2 V7 설치 창이 열립니다.
6. 클라이언트용 DB2 Relational Connect 제품(예: Sybase용 Relational Connect)을 탐색하고 스페이스바를 눌러 선택합니다. 옵션을 선택하면 옆에 별표가 나타납니다.
7. 확인을 선택합니다. DB2 Services 작성 창이 열립니다.
8. DB2 인스턴스를 작성하도록 선택할 수 있습니다. 확인을 선택합니다. 요약 보고서 설치 창이 열립니다. Oracle용 Distributed Join 및 DB2 Relational Connect용 Product Signature의 두 항목이 자동으로 설치됩니다. Product Signature는 Sybase 데이터 소스에 연결할 경우 필요합니다.
9. 계속을 선택합니다. Relational Connect 설치를 중지할 수 있는 마지막 기회임을 지시하는 창이 나타납니다. 확인을 눌러 설치를 계속합니다. 설치를 완료하려면 몇 분 정도 걸립니다.
10. 설치가 성공적으로 완료되었음을 알려주면 확인을 누릅니다. 각각의 설치된 옵션이 성공했는지 실패했는지를 나타내는 요약 보고서 창이 나타납니다. 다시 확인을 누릅니다.

설치가 끝나면 DB2 Relational Connect가 다른 DB2 제품과 함께 디렉토리에 설치됩니다. AIX에서는 /usr/lpp/db2_07_01 디렉토리에 설치됩니다. Solaris에서는 /opt/IBMDB2/V7.1 디렉토리에 설치됩니다. Linux에서는 /usr/IBMDB2/V7.1 디렉토리에 설치됩니다.

Oracle 데이터 소스에 액세스

DB2 Universal Database는 AIX 및 Windows NT에서 랩퍼를 지원할 뿐 아니라 이제 Linux 및 Solaris 운영 환경에서 Oracle 랩퍼도 지원합니다. 이러한 지원은 Oracle 버전 8로 제한됩니다. 이들 플랫폼에 대한 랩퍼에 액세스하려면 V7.2 DB2 Relational Connect CD를 넣고 Oracle 데이터 소스용 DB2 Relational Connect를 선택합니다.

일단 DB2 Relational Connect를 설치하면 연합 서버에 Oracle 데이터 소스를 추가할 수 있습니다.

1. DB2 연합 서버에서 Oracle 클라이언트 소프트웨어를 설치하고 구성합니다.
2. db2dj.ini 파일을 수정하고 **db2set** 명령을 실행하여 데이터 소스 환경 변수를 설정합니다.
3. UNIX 플랫폼에서 수행되는 DB2 연합 서버인 경우 djxlink 스크립트를 실행하여 DB2 연합 서버에 Oracle SQL*Net 또는 Net8 라이브러리를 링크 편집합니다.
4. SQL*Net 또는 Net8 tnsnames.ora 파일이 갱신되는지 확인합니다.
5. DB2 인스턴스를 재순환합니다.
6. 랩퍼를 작성합니다.
7. 선택적으로 DB2_DJ_COMM 환경 변수를 설정합니다.
8. 서버를 작성합니다.
9. 사용자 맵핑을 작성합니다.
10. 테이블 및 뷰에 대한 별명을 작성합니다.

환경 변수 설정을 포함하여 이 단계에 대한 자세한 지침은 *DB2 설치 및 구성 보충 설명서*의 Oracle 데이터 소스에 액세스하기 위해 연합 시스템 설정에 나와 있습니다.

문서 오류

『Oracle 데이터 소스를 연합 시스템에 추가』 절에 다음 오류가 있습니다.

- 프로시듀어에서 한 단계가 누락되었습니다. 올바른 단계는 아래와 같습니다.
 1. Oracle이 제공하는 문서를 사용하여 DB2 연합 서버에 Oracle 클라이언트 소프트웨어를 설치 및 구성하십시오.
 2. db2dj.ini 파일을 수정하고 **db2set** 명령을 실행하여 데이터 소스 환경 변수를 설정하십시오. **db2set** 명령은 사용자 설정으로 DB2 프로파일 레지스트리를 갱신합니다.
환경 변수 설정에 대한 자세한 지침은 *DB2 설치 및 구성 보충 설명서*를 참조하십시오.

3. UNIX 플랫폼에서 수행되는 DB2 연합 서버의 경우, djxlink 스크립트를 수행하여 Oracle SQL*Net 또는 Net8 라이브러리를 DB2 연합 서버에 링크 편집하십시오. 플랫폼에 따라 djxlink 스크립트는 다음 위치에 있습니다.

/usr/lpp/db2_07_01/bin on AIX

/opt/IBMdb2/V7.1/bin Solaris

/usr/IBMdb2/V7.1/bin Linux

Oracle의 클라이언트 소프트웨어를 DB2 연합 서버에 설치한 후에만 djxlink 스크립트를 수행하십시오.

- 문서에서 다음과 같이 설정하도록 지시하고 있습니다.

```
DB2_DJ_INI = sqllib/cfg/db2dj.ini
```

이는 잘못되었습니다. 다음과 같이 설정해야 합니다.

```
DB2_DJ_INI = $INSTHOME/sqllib/cfg/db2dj.ini
```

Sybase 데이터 소스에 액세스(새 장)

연합 서버에 Sybase 데이터 소스를 추가하기 전에 DB2 연합 서버에 Sybase Open Client 소프트웨어를 설치 및 구성해야 합니다. Open Client 소프트웨어의 설치에 관해 자세한 사항은 Sybase에 포함된 문서에서 설치 절차를 참조하십시오. 설치의 일부로 카탈로그 저장 프로시저 및 Sybase Open Client 라이브러리가 포함되는지 확인하십시오.

연합 서버를 설정하여 Sybase 데이터 소스에 저장된 데이터에 액세스하려면 다음을 수행해야 합니다.

1. DB2 Relational Connect 버전 7.2를 설치합니다. 40 페이지의 『DB2 Relational Connect 설치』를 참조하십시오.
2. Sybase 데이터 소스를 연합 서버에 추가합니다.
3. Sybase 코드 페이지를 지정합니다.

이 장은 2 단계 및 3 단계를 설명합니다.

이 장의 지침은 Windows NT, AIX 및 Solaris 운영 환경에 적용됩니다. 플랫폼별 차이는 해당 부분에서 설명합니다.

Sybase 데이터 소스를 연합 서버에 추가

Sybase 데이터 소스를 연합 서버에 추가하려면 다음을 수행해야 합니다.

1. 환경 변수를 설정하고 프로파일 레지스트리를 갱신합니다.
2. DB2를 Sybase 클라이언트 소프트웨어에 링크합니다(AIX 및 Solaris만 해당).
3. DB2 인스턴스를 재순환합니다.

4. 인터페이스 파일을 작성하고 설정합니다.
5. 랩퍼를 작성합니다.
6. 선택적으로 DB2_DJ_COMM 환경 변수를 설정합니다.
7. 서버를 작성합니다.
8. 선택적으로 CONNECTSTRING 서버 옵션을 설정합니다.
9. 사용자 맵핑을 작성합니다.
10. 테이블 및 뷰에 대한 별명을 작성합니다.

이 절은 각 단계의 자세한 사항을 설명합니다.

1 단계: 환경 변수 설정 및 프로파일 레지스트리 갱신

db2dj.ini 파일을 수정하고 **db2set** 명령을 실행하여 데이터 소스 환경 변수를 설정하십시오. db2dj.ini 파일에는 연합 서버에 설치된 Sybase 클라이언트 소프트웨어의 구성 정보가 들어 있습니다. **db2set** 명령은 사용자 설정으로 DB2 프로파일 레지스트리를 갱신합니다.

파티션된 데이터베이스 시스템의 특정 인스턴스에서 모든 노드에 하나의 db2dj.ini 파일을 사용하거나 특정 인스턴스에서 하나 이상의 노드에 고유한 db2dj.ini 파일을 사용할 수 있습니다. 파티션되지 않은 데이터베이스 시스템에는 인스턴스별로 단 하나의 db2dj.ini 파일만 있습니다.

환경 변수를 설정하려면 다음을 수행하십시오.

1. sql1lib/cfg에 있는 db2dj.ini 파일을 편집하고 다음 환경 변수를 설정합니다.

```
SYBASE="<sybase home directory>"
```

여기서 *<sybase home directory>*는 Sybase 클라이언트가 설치된 디렉토리입니다.

2. DB2 인스턴스의 .profile 파일을 Sybase 환경 변수로 갱신합니다. 다음 명령을 실행하여 이를 수행할 수 있습니다.

```
export PATH=『$SYBASE/bin:$PATH』
export SYBASE="<sybase home directory>"
```

여기서 *<sybase home directory>*는 Sybase 클라이언트가 설치된 디렉토리입니다.

3. 다음을 입력하여 DB2 인스턴스 .profile을 실행합니다.

```
. .profile
```

4. **db2set** 명령을 실행하여 DB2 프로파일 레지스트리를 변경 내용으로 갱신합니다. **db2set** 명령의 구문은 데이터베이스 시스템 구조에 따라 다릅니다. 이 단계는 다음 데이터베이스 시스템 구조에서 db2dj.ini 파일을 사용하는 경우에만 필요합니다. 파티션되지 않은 데이터베이스 시스템에서 db2dj.ini 파일을 사용하거나 db2dj.ini 파일을 현재 노드에만 적용하려면 다음을 실행합니다.

```
db2set DB2_DJ_INI = sql1lib/cfg/db2dj.ini
```

파티션된 데이터베이스 시스템에서 db2dj.ini 파일을 사용하고 있고, db2dj.ini 파일의 값을 이 인스턴스의 모든 노드에 적용하려면 다음을 실행합니다.

```
db2set -g DB2_DJ_INI = sqllib/cfg/db2dj.ini
```

db2dj.ini 파일을 파티션된 데이터베이스 시스템에서 사용하고 있고, db2dj.ini 파일의 값을 특정 노드에 적용하려면 다음을 실행합니다.

```
db2set -i INSTANCEX 3 DB2_DJ_INI = sqllib/cfg/node3.ini
```

여기서,

INSTANCEX

인스턴스 이름입니다.

3 db2nodes.cfg 파일에 있는 노드 번호입니다.

node3.ini

db2dj.ini 파일의 이름이 바뀐 수정된 버전입니다.

2 단계: DB2를 Sybase 클라이언트 소프트웨어에 링크(AIX 및 Solaris만 해당)

Sybase 데이터 소스에 액세스하려면 DB2 연합 서버가 클라이언트 라이브러리에 링크 편집되어야 합니다. 링크 편집 프로세스는 연합 서버가 통신하는 각 데이터 소스의 랩 퍼를 작성합니다. djsxlink 스크립트를 실행하면 랩퍼 라이브러리가 작성됩니다. djsxlink 스크립트를 실행하려면 다음을 입력하십시오.

```
djsxlink
```

3 단계: DB2 인스턴스 재순환

프로그램에서 환경 변수를 설정하려면 DB2 인스턴스를 재순환합니다. 인스턴스를 재순환하면 DB2 인스턴스를 새로 고쳐서 변경 내용을 적용합니다.

다음 명령을 실행하여 DB2 인스턴스를 재순환 하십시오.

Windows NT 서버용 DB2의 경우,

```
NET STOP instance_name  
NET START instance_name
```

AIX 및 Solaris 서버용 DB2의 경우,

```
db2stop  
db2start
```

4 단계: 인터페이스 파일 작성 및 설정

인터페이스 파일을 작성하고 설정하려면 파일을 작성한 후 파일에 액세스할 수 있게 해야 합니다.

1. Sybase 제공 유틸리티를 사용하여 액세스할 모든 Sybase Open Servers용 데이터를 포함하는 인터페이스 파일을 작성합니다. 유틸리티 사용에 관해 자세한 정보는 Sybase 설치 문서를 참조하십시오.

Windows NT는 일반적으로 이 파일의 이름을 sql.ini으로 지정합니다. 지금 작성한 파일의 이름을 sql.ini에서 interfaces로 바꿔서 모든 플랫폼에서 파일 이름을 동일하게 사용합니다. sql.ini의 이름을 interfaces로 바꾸지 않으려면 8 단계에서 설명한 CONNECTSTRING 옵션이나 IFILE 매개변수를 사용해야 합니다.

AIX 및 Solaris 시스템에서 이 파일 이름은 <instance home>/sqllib/interfaces 입니다.

2. DB2가 인터페이스 파일에 액세스할 수 있게 합니다.

Windows NT 서버용 DB2의 경우,

파일을 DB2 인스턴스의 %DB2PATH% 디렉토리에 둡니다.

AIX 및 Solaris 서버용 DB2의 경우,

파일을 DB2 인스턴스의 \$HOME/sqllib 디렉토리에 둡니다. **ln** 명령을 사용하여 DB2 인스턴스의 \$HOME/sqllib 디렉토리에서 파일에 링크합니다. 예를 들어, 다음과 같습니다.

```
ln -s -f /home/sybase/interfaces /home/db2djinst1/sqllib
```

5 단계: 래퍼 작성

CREATE WRAPPER 문을 사용하여 Sybase 데이터 소스에 액세스할 때 사용할 래퍼를 지정합니다. 래퍼는 연합 서버가 데이터 소스와 통신하고 데이터 소스에서 데이터를 검색할 때 사용하는 메커니즘입니다. DB2의 Sybase용 래퍼 두 가지는 CTLIB 및 DBLIB입니다. 다음은 CREATE WRAPPER 문의 예입니다.

```
CREATE WRAPPER CTLIB
```

여기서 CTLIB는 Sybase Open Client 소프트웨어에서 사용하는 기본 래퍼 이름입니다. CTLIB 래퍼는 Windows NT, AIX 및 Solaris 서버에서 사용할 수 있습니다.

기본 래퍼 이름을 선택한 이름으로 대체할 수 있습니다. 그러나 이 경우 CREATE WRAPPER 문에 연합 서버용 래퍼 라이브러리 이름 및 LIBRARY 매개변수를 포함해야 합니다. 래퍼 라이브러리 이름 관련 자세한 정보는 *DB2 SQL 참조서*의 CREATE WRAPPER 문을 참조하십시오.

6 단계: DB2_DJ_COMM 환경 변수 설정

Sybase 데이터 소스에 액세스할 때 성능을 개선하려면 DB2_DJ_COMM 환경 변수를 설정합니다. 이 변수는 연합 서버를 초기화할 때 래퍼가 로드될 것인지 결정합니다. 앞 단계에서 지정한 래퍼의 해당 래퍼 라이브러리를 포함하도록 DB2_DJ_COMM 환경 변수를 설정합니다. 예를 들어 다음과 같습니다.

Windows NT 서버용 DB2의 경우,

```
db2set DB2_DJ_COMM='ctlib.dll'
```

AIX 서버용 DB2의 경우,

```
db2set DB2_DJ_COMM='libctlib.a'
```

Solaris 서버용 DB2의 경우,

```
db2set DB2_DJ_COMM='libctlib.so'
```

등호(=) 앞 뒤에 공백이 없는지 확인하십시오.

랩퍼 라이브러리 이름에 대한 자세한 정보는 *DB2 SQL* 참조서를 참조하십시오.

DB2_DJ_COMM 환경 변수에 대한 자세한 정보는 *관리 안내서*를 참조하십시오.

7 단계: 서버 작성

다음과 같이 CREATE SERVER 문을 사용하여 액세스할 데이터 소스가 있는 각 Sybase 서버를 정의합니다.

```
CREATE SERVER SYBSERVER TYPE SYBASE VERSION 12.0 WRAPPER CTLIB  
OPTIONS (NODE 'sybnode', DBNAME 'sybdb')
```

여기서,

SYBSERVER

Sybase 서버에 할당할 이름입니다. 이 이름은 고유해야 합니다.

SYBASE

액세스를 구성할 데이터 소스 유형입니다. Sybase가 지원되는 유일한 데이터 소스입니다.

12.0 액세스하는 Sybase 버전입니다. 지원되는 버전은 10.0, 11.0, 11.1, 11.5, 11.9 및 12.0입니다.

CTLIB CREATE WRAPPER 문에서 지정한 랩퍼 이름입니다.

'*sybnode*'

*SYBSERVER*가 있는 노드 이름입니다. 인터페이스 파일에서 노드 값을 얻으십시오. 이 값은 대소문자를 구별합니다.

노드 이름이 옵션으로 지정되었지만 Sybase 데이터 소스인 경우에는 필수입니다. 추가 옵션에 대한 자세한 정보는 *DB2 SQL* 참조서를 참조하십시오.

'*sybdb*'

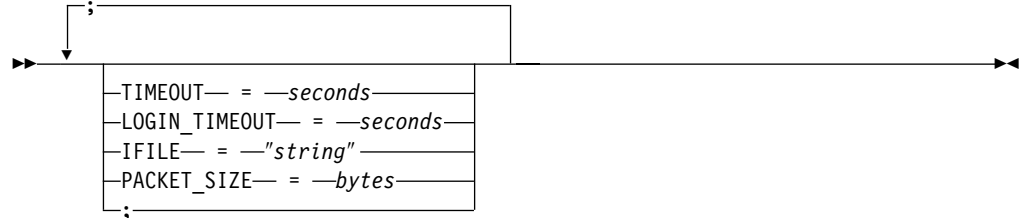
액세스할 Sybase 데이터베이스 이름입니다.

선택적: 8 단계: **CONNECTSTRING** 서버 옵션 설정

시간종료 임계값, 인터페이스 파일의 경로 및 이름, 그리고 인터페이스 파일의 패킷 크기를 지정합니다. Sybase Open Client는 시간종료 임계값을 사용하여 너무 오랫동안 수행되는 응답 및 조회를 인터럽트합니다. CREATE SERVER OPTION DDL 문에

서 CONNECTSTRING 옵션을 사용하여 DB2에서 이 임계값을 설정할 수 있습니다. CONNECTSTRING 옵션을 사용하여 다음 값을 지정합니다.

- SQL 조회의 시간종료 지속기간
- 로그인 응답의 시간종료 지속기간
- 인터페이스 파일의 경로 및 이름
- 패킷 크기



TIMEOUT

SQL문에 대해 DB2 Universal Database가 Sybase Open Client의 응답을 기다리는 초 수를 지정합니다. *seconds*의 값은 DB2 Universal Database 정수 범위의 양수입니다. 지정할 시간종료 값은 사용하는 랩퍼에 따라 다릅니다. Windows NT, AIX 및 Solaris 서버는 모두 DBLIB 랩퍼를 사용할 수 있습니다. DBLIB 랩퍼의 기본값은 0입니다. Windows NT, AIX 및 Solaris 서버에서 DBLIB의 기본값을 사용하면 DB2 Universal Database가 응답을 무한정 기다리게 됩니다.

LOGIN_TIMEOUT

DB2 Universal Database가 로그인 요청에 대해 Sybase Open Client의 응답을 기다리는 초 수를 지정합니다. 기본값은 TIMEOUT과 동일합니다.

IFILE

Sybase Open Client 인터페이스 파일의 경로 및 이름을 지정합니다. 문자열에 지정된 경로에는 큰 따옴표를 쳐야 합니다. Windows NT 서버인 경우 기본값은 %DB2PATH%입니다. AIX 및 Solaris 서버인 경우 기본값은 DB2 Universal Database 인스턴스의 홈 디렉토리에서 sqllib/interfaces입니다.

PACKET_SIZE

인터페이스 파일의 패킷 크기를 바이트 단위로 지정합니다. 데이터 소스가 지정된 패킷 크기를 지원하지 않으면 연결에 실패합니다. 각 레코드가 매우 클 경우 패킷 크기를 늘리면(예를 들어, 큰 테이블에 행을 삽입할 때) 성능이 크게 향상됩니다. 바이트 크기는 숫자 값입니다. 자세한 정보는 Sybase 참조 매뉴얼을 참조하십시오.

예:

Windows NT 서버인 경우 시간종료 값을 60초로 설정하고 인터페이스 파일을 C:\etc\interfaces로 설정하려면 다음을 사용하십시오.

```
CREATE SERVER OPTION connectstring FOR SERVER sybase1
SETTING 'TIMEOUT=60;LOGIN_TIMEOUT=5;IFILE="C:\etc\interfaces";'
```

AIX 및 Solaris 서버인 경우 시간종료 값을 60초로 설정하고 인터페이스 파일을 /etc/interfaces로 설정하려면 다음을 사용하십시오.

```
CREATE SERVER OPTION connectstring FOR SERVER sybase1
SETTING 'TIMEOUT=60;PACKET_SIZE=4096;IFILE="/etc/interfaces";'
```

9 단계: 사용자 맵핑 작성

연합 서버의 사용자 ID나 암호가 Sybase 데이터 소스의 사용자 ID나 암호와 다르면 다음과 같이 CREATE USER MAPPING 문을 사용하여 지역 사용자 ID를 Sybase 데이터 소스에 정의된 사용자 ID 및 암호로 맵핑합니다.

```
CREATE USER MAPPING FOR DB2USER SERVER SYBSERVER
OPTIONS ( REMOTE_AUTHID 'sybuser', REMOTE_PASSWORD 'dayllte')
```

여기서,

DB2USER

Sybase 데이터 소스에 정의된 사용자 ID로 맵핑할 지역 사용자 ID입니다.

SYBSERVER

CREATE SERVER 문에서 정의한 Sybase 데이터 소스 이름입니다.

'*sybuser*'

DB2USER를 맵핑할 Sybase 데이터 소스의 사용자 ID입니다. 이 값은 대소문자를 구별합니다.

'*dayllte*'

'*sybuser*'와 연관된 암호입니다. 이 값은 대소문자를 구별합니다.

추가 옵션에 대한 자세한 정보는 *DB2 SQL* 참조서를 참조하십시오.

10 단계: 테이블 및 뷰에 대한 별명 작성

Sybase 데이터 소스에 있는 각각의 뷰나 테이블에 대한 별명을 지정합니다. Sybase 데이터 소스를 조회할 때 이 별명을 사용합니다. Sybase 별명은 대소문자를 구별합니다. 스키마 및 테이블 이름에 큰 따옴표를 쳐야 합니다. 다음은 CREATE NICKNAME 문의 예입니다.

```
CREATE NICKNAME SYBSALES FOR SYBSERVER."salesdata"."europe"
```

여기서,

SYBSALES

Sybase 테이블이나 뷰의 고유한 별명입니다.

SYBSERVER."salesdata"."europe"**

세 부분으로 구성된 식별자인데 다음 형식을 따릅니다.

data_source_name."remote_schema_name"."remote_table_name"**

별명을 작성할 각각의 테이블이나 뷰에 대해 이 단계를 반복하십시오. 별명을 작성할 때 DB2는 연결을 통해 데이터 소스 카탈로그를 조회합니다. 이 조회에서 데이터 소스와의 연결을 테스트합니다. 연결이 작동되지 않으면 오류 메시지가 나타납니다.

CREATE NICKNAME 문에 대한 자세한 정보는 *DB2 SQL 참조서*를 참조하십시오. 일반적인 별명 및 데이터 유형 매핑을 검증하는 데 대한 자세한 정보는 *DB2 관리 안내서*를 참조하십시오.

Sybase 코드 페이지 지정

이 단계는 DB2 연합 서버 및 Sybase 서버가 다른 코드 페이지에서 수행되고 있을 때만 필요합니다. DB2와 같은 코드 세트를 사용하는 데이터 소스는 변환할 필요가 없습니다. 다음 표는 일반적인 NLS(National Language Support) 코드 페이지에 동등한 Sybase 옵션을 제공합니다. 이들 동등한 값에 대응하여 Sybase 데이터 소스를 구성해야 하거나, 클라이언트 코드가 불일치를 발견하여 오류 플래그를 표시하거나 자체 의미를 사용하여 데이터를 매핑할 수 있어야 합니다. 소스 코드 페이지에서 목표 코드 페이지로 변환하는 변환 테이블이 없으면 DB2는 오류 메시지를 표시합니다. 자세한 정보는 Sybase 문서를 참조하십시오.

표 1. Sybase 코드 페이지 옵션

코드 페이지	동등한 Sybase 옵션
850	cp850
897	sjis
	iso_1
912	iso_2
1089	iso_6
813	iso_7
916	iso_8
920	iso_9

ODBC를 사용하여 Microsoft SQL Server 데이터 소스에 액세스(새 장)

Microsoft SQL Server 데이터 소스를 DB2 연합 서버에 추가하기 전에 연합 서버에서 ODBC 드라이버를 설치하고 구성해야 합니다. ODBC 드라이버 설치 방법에 대한 구체적인 자세한 정보는 ODBC 드라이버에 포함된 문서에 나타난 설치 절차를 참조하십시오.

연합 서버를 설정하여 Microsoft SQL Server 데이터 소스에 저장된 데이터에 액세스하려면 다음을 수행해야 합니다.

1. 연합 서버에서 ODBC 드라이버를 설치하고 구성합니다. ODBC 드라이버 설치에 관해 자세한 정보는 ODBC 드라이버에 포함된 문서에서 설치 절차를 참조하십시오.

Windows NT 서버용 DB2의 경우,

ODBC 장치 관리 프로그램을 사용하여 DSN 시스템을 구성합니다.

AIX 서버용 DB2의 경우,

MERANT가 제공한 라이브러리의 스테드된 버전을 설치하고 MERANT 라이브러리 디렉토리를 LIBPATH의 첫번째 항목으로 지정하고 .odbc.ini 파일을 설정합니다. 홈 디렉토리에 .odbc.ini 파일을 작성합니다.

2. DB2 Relational Connect 버전 7.2를 설치합니다. 40 페이지의 『DB2 Relational Connect 설치』를 참조하십시오.
3. Microsoft SQL Server 데이터 소스를 연합 서버에 추가합니다.
4. Microsoft SQL Server 코드 페이지를 지정합니다.

이 장에서는 3 단계 및 4 단계를 설명합니다.

이 장의 지침은 Windows NT 및 AIX 플랫폼에 적용됩니다. 플랫폼별 차이는 해당 부분에서 설명합니다.

연합 서버에 Microsoft SQL Server 데이터 소스 추가

ODBC 드라이버 및 DB2 Relational Connect를 설치한 후, 다음 단계에 따라 Microsoft SQL Server 데이터 소스를 연합 서버에 추가합니다.

1. 환경 변수를 설정합니다(AIX만 해당).
2. 셸 스크립트를 수행합니다(AIX만 해당).
3. 선택적으로 DB2_DJ_COMM 환경 변수를 설정합니다.
4. DB2 인스턴스를 재순환합니다(AIX만 해당).
5. 랩퍼를 작성합니다.
6. 서버를 작성합니다.
7. 사용자 맵핑을 작성합니다.
8. 테이블 및 뷰의 별명을 작성합니다.
9. 선택적으로 ODBC 추적을 연습니다.

이 단계는 다음 절에 자세히 설명합니다.

1 단계: 환경 변수 설정(AIX만 해당)

db2dj.ini 파일을 수정하고 **db2set** 명령을 실행하여 데이터 소스 환경 변수를 설정하십시오. db2dj.ini 파일에는 Microsoft SQL Server 데이터 소스에 연결하는 구성 정보가 들어 있습니다. **db2set** 명령은 사용자 설정으로 DB2 프로파일 레지스트리를 갱신합니다.

파티션된 데이터베이스 시스템의 특정 인스턴스에서 모든 노드에 하나의 db2dj.ini 파일을 사용하거나 특정 인스턴스에서 하나 이상의 노드에 고유한 db2dj.ini 파일을 사용할 수 있습니다. 파티션되지 않은 데이터베이스 시스템에는 인스턴스별로 단 하나의 db2dj.ini 파일만 있습니다.

환경 변수를 설정하려면 다음을 수행하십시오.

1. \$HOME/sqlllib/cfg/에 있는 db2dj.ini 파일을 편집하고 다음 환경 변수를 설정합니다.

```
ODBCINI=$HOME/.odbc.ini
DJX_ODBC_LIBRARY_PATH=<path to the Merant driver>/lib
LIBPATH=<path to the Merant driver>/lib
DB2ENVLIST=LIBPATH
```

db2set 명령을 실행하여 DB2 프로파일 레지스트리를 변경 내용으로 갱신합니다. **db2set** 구문은 데이터베이스 시스템 구조에 따라 다릅니다.

- 파티션되지 않은 데이터베이스 시스템에서 db2dj.ini 파일을 사용하거나, 파티션된 데이터베이스 시스템에서 db2dj.ini 파일을 사용하고 있고, db2dj.ini 파일의 값을 현재 노드에만 적용하려면 다음 명령을 실행합니다.

```
db2set DB2_DJ_INI=<path to ini file>/db2dj.ini
```

- 파티션된 데이터베이스 시스템에서 db2dj.ini 파일을 사용하고 있고, db2dj.ini 파일의 값을 이 인스턴스 내의 모든 노드에 적용하려면 다음 명령을 실행합니다.

```
db2set -g DB2_DJ_INI=<path to ini file>/db2dj.ini
```

- 파티션된 데이터베이스 시스템에서 db2dj.ini 파일을 사용하고 있고, db2dj.ini 파일의 값을 특정 노드에 적용하려면 다음 명령을 실행합니다.

```
db2set -i INSTANCEX 3 DB2_DJ_INI=$HOME/sqlllib/cfg/node3.ini
```

여기서,

INSTANCEX

인스턴스 이름입니다.

3 db2nodes.cfg 파일에 있는 노드 번호입니다.

node3.ini

db2dj.ini 파일의 이름이 바뀐 수정된 버전입니다.

2 단계: 셸 스크립트 수행(AIX만 해당)

djxlink.sh 셸 스크립트는 클라이언트 라이브러리를 랩퍼 라이브러리에 링크합니다. 셸 스크립트를 수행하려면 다음 명령을 실행하십시오.

```
djxlink
```

3 단계: 선택적: DB2_DJ_COMM 환경 변수 설정

Microsoft SQL Server 데이터 소스에 액세스할 때 시간이 너무 많이 걸리면 데이터 소스에 액세스할 때가 아니라 연합 서버를 초기화할 때 랩퍼를 로드하기 위해 DB2_DJ_COMM 환경 변수를 설정하여 성능을 개선할 수 있습니다. 다음과 같이 5 단계에서 지정한 랩퍼에 해당하는 랩퍼 라이브러리를 포함하는 DB2_DJ_COMM 환경 변수를 설정합니다.

Windows NT 서버용 DB2의 경우,

```
db2set DB2_DJ_COMM=djxmssql3.d11
```

AIX 서버용 DB2의 경우,

```
db2set DB2_DJ_COMM=libmssql3.a
```

등호(=) 앞 뒤에 공백이 없는지 확인하십시오.

랩퍼 라이브러리 이름에 관해 자세한 정보는 *DB2 SQL 참조서*를 참조하십시오.

4 단계: DB2 인스턴스 재순환(AIX만 해당)

프로그램에서 환경 변수를 설정하려면 DB2 인스턴스를 재순환합니다. 인스턴스를 재순환하면 DB2 인스턴스를 새로 고쳐서 변경 내용을 적용합니다. 다음 명령을 실행하여 DB2 인스턴스를 재순환합니다.

```
db2stop  
db2start
```

5 단계: 랩퍼 작성

DB2 Universal Database에는 Microsoft SQL Server 데이터 소스에 액세스할 때 사용할 수 있는 랩퍼라는 두 개의 서로 다른 프로토콜이 있습니다. 랩퍼는 연합 서버가 데이터 소스와 통신하고 데이터 소스에서 데이터를 검색할 때 사용하는 메커니즘입니다. 사용하는 랩퍼는 DB2 Universal Database가 수행되는 플랫폼에 따라 달라집니다. 표 2를 지침으로 적절한 랩퍼를 선택하십시오.

표 2. ODBC 드라이버

ODBC 드라이버	플랫폼	랩퍼 이름
ODBC 3.0(또는 이상) 드라이버	Windows NT	DJXMSSQL3
MERANT DataDirect Connect ODBC 3.6 드라이버	AIX	MSSQLODBC3

CREATE WRAPPER 문을 사용하여 Microsoft SQL Server 데이터 소스에 액세스할 때 사용할 랩퍼를 지정합니다. 다음은 CREATE WRAPPER 문의 예입니다.

```
CREATE WRAPPER DJXMSSQL3
```

여기서 DJXMSSQL3은 (ODBC 3.0 드라이버를 사용하여) Windows NT용 DB2에서 사용되는 기본 랩퍼 이름입니다. AIX 서버용 DB2가 있는 경우 MSSQLODBC3 랩퍼 이름을 지정합니다.

기본 래퍼 이름을 선택한 이름으로 대체할 수 있습니다. 그러나 이 경우 CREATE WRAPPER 문에 연합 서버 플랫폼용 래퍼 라이브러리 이름 및 LIBRARY 매개변수도 포함되어야 합니다. 예를 들어, 다음과 같습니다.

Windows NT 서버용 DB2의 경우,

```
CREATE WRAPPER wrapper_name LIBRARY 'djmssql3.dll'
```

여기서 *wrapper_name*은 래퍼에 지정할 이름이며 'djmssql3.dll'은 라이브러리 이름입니다.

AIX 서버용 DB2의 경우,

```
CREATE WRAPPER wrapper_name LIBRARY 'libmssql3.a'
```

여기서 *wrapper_name*은 래퍼에 지정할 이름이며 'libdjmssql.a'는 라이브러리 이름입니다.

래퍼 라이브러리 이름에 대한 자세한 정보는 *DB2 SQL 참조서*의 CREATE WRAPPER 문을 참조하십시오.

6 단계: 서버 작성

CREATE SERVER 문을 사용하여 연결할 Microsoft SQL Server 데이터 소스를 각각 정의합니다. 예를 들어, 다음과 같습니다.

```
CREATE SERVER sqlserver TYPE MSSQLSERVER VERSION 7.0 WRAPPER djmssql3  
OPTIONS (NODE 'sqlnode', DBNAME 'database_name')
```

여기서,

sqlserver

Microsoft SQL Server 서버에 할당할 이름입니다. 이 이름은 고유해야 합니다.

MSSQLSERVER

액세스를 구성할 데이터 소스 유형입니다.

7.0 액세스하는 Microsoft SQL Server 버전입니다. DB2 Universal Database는 Microsoft SQL Server 버전 6.5 및 7.0을 지원합니다.

DJXMSSQL3

CREATE WRAPPER 문에서 정의한 래퍼 이름입니다.

'*sqlnode*'

액세스하는 Microsoft SQL Server의 버전을 참조하는 시스템 DSN 이름입니다. 이 값은 대소문자를 구별합니다. DB2 Universal Database는 Microsoft SQL Server 버전 6.5 및 7.0을 지원합니다.

CREATE SERVER 문에는 노드 이름(시스템 DSN 이름)이 옵션으로 지정되었지만 Microsoft SQL Server 데이터 소스의 경우는 필수입니다.

CREATE WRAPPER 문에서 사용할 수 있는 추가 옵션은 *DB2 SQL 참조서*를 참조하십시오.

'database_name'

연결할 데이터베이스 이름입니다.

CREATE SERVER 문에는 데이터베이스 이름이 옵션으로 지정되었지만 Microsoft SQL Server 데이터 소스의 경우는 필수입니다.

7 단계: 사용자 맵핑 작성

연합 서버의 사용자 ID나 암호가 Microsoft SQL Server 데이터 소스의 사용자 ID나 암호와 다르면 다음과 같이 CREATE USER MAPPING 문을 사용하여 지역 사용자 ID를 Microsoft SQL Server 데이터 소스에 정의된 사용자 ID 및 암호로 맵핑합니다.

```
CREATE USER MAPPING FOR db2user SERVER server_name
OPTIONS (REMOTE_AUTHID 'mssqluser', REMOTE_PASSWORD 'dayllite')
```

여기서,

db2user

Microsoft SQL Server 데이터 소스에 정의된 사용자 ID로 맵핑할 지역 사용자 ID입니다.

server_name

CREATE SERVER 문에서 정의한 서버 이름입니다.

'mssqluser'

*db2user*를 맵핑할 Microsoft SQL Server 데이터 소스의 사용자 ID입니다. 이 값은 대소문자를 구별합니다.

'dayllite'

*'mssqluser'*와 연관된 암호입니다. 이 값은 대소문자를 구별합니다.

CREATE USER MAPPING 문에서 사용할 수 있는 추가 옵션은 *DB2 SQL 참조서*를 참조하십시오.

8 단계: 테이블 및 뷰에 대한 별명 작성

액세스할 Microsoft SQL Server 데이터 소스의 뷰나 테이블 각각의 별명을 지정합니다. Microsoft SQL Server 데이터 소스를 조회할 때 이 별명을 사용합니다. CREATE NICKNAME 문을 사용하여 별명을 지정합니다. 별명은 대소문자를 구별합니다. 다음은 CREATE NICKNAME 문의 예입니다.

```
CREATE NICKNAME mssqlsales FOR server_name.salesdata.europe
```

여기서,

mssqlsales

Microsoft SQL Server 테이블이나 뷰의 고유한 별명입니다.

server_name.salesdata.europe

세 부분으로 구성된 식별자인데 다음 형식을 따릅니다.

data_source_server_name.remote_schema_name.remote_table_name

별명의 *remote_schema_name* 및 *remote_table_name* 부분은 큰 따옴표를 사용하는 것이 좋습니다.

별명을 만들 때 DB2는 데이터 소스 시스템 카탈로그 테이블(Microsoft SQL Server에서는 시스템 테이블이라고 함)에 액세스하려고 합니다. 여기서 데이터 소스와 연결을 테스트합니다. 연결되지 않으면 오류 메시지가 나타납니다.

별명을 작성할 모든 데이터베이스 테이블 및 뷰는 이 단계를 반복하십시오.

CREATE NICKNAME 문에 관해 자세한 정보는 *DB2 SQL 참조서*를 참조하십시오. 일반적인 별명 및 데이터 유형 매핑의 검증에 관해 자세한 정보는 *DB2 관리 안내서*를 참조하십시오.

9 단계: ODBC 추적 얻기(선택적)

데이터 소스 액세스에 문제가 있으면 ODBC 추적 정보를 얻어서 이 문제를 분석하고 해결할 수 있습니다. ODBC 추적 작업을 적절하게 수행하려면 ODBC 데이터 소스 관리자가 제공하는 추적 도구를 사용합니다. 추적을 활성화하면 시스템 성능에 영향을 주므로 문제를 해결한 후에는 추적을 해제해야 합니다.

Microsoft SQL Server 코드 페이지 검토

Microsoft SQL Server는 DB2 UDB와 많은 부분 공통되는 자국어 지원(NLS)의 코드 페이지 옵션을 지원합니다. DB2와 같은 코드 세트를 사용하는 데이터 소스는 변환할 필요가 없습니다. 표 3에는 DB2 Universal Database와 Microsoft SQL Server가 모두 지원하는 코드 페이지가 나와 있습니다.

표 3. DB2 UDB 및 Microsoft SQL Server 코드 페이지 옵션

코드 페이지	지원되는 언어
1252	ISO 문자 세트
850	다국어
437	영어
874	태국어
932	일본어
936	중국어
949	한국어
950	대만어
1250	중앙 유럽어
1251	시릴어
1253	그리스어
1254	터키어

표 3. DB2 UDB 및 Microsoft SQL Server 코드 페이지 옵션 (계속)

코드 페이지	지원되는 언어
1255	히브리어
1256	아랍어

DB2 연합 서버 및 Microsoft SQL Server가 서로 다른 자국어 지원(NLS) 코드 페이지에서 수행되는 경우, 동등한 값에 상응하도록 Microsoft SQL Server 데이터 소스를 구성하거나 클라이언트 코드가 불일치를 발견하여 오류 플래그를 표시하거나 자체 의미를 사용하여 데이터를 맵핑해야 합니다. 소스 코드 페이지에서 목표 코드 페이지로 변환하는 변환 테이블이 없으면 DB2는 오류 메시지를 표시합니다. 자세한 정보는 Microsoft SQL Server 문서를 참조하십시오.

제3부 관리

관리 안내서: 계획	63	순차 변경	79
물리적 데이터베이스 설계	63	순차 삭제	79
파티션 키	63	테이블 공간의 상태 전환	80
노드 그룹 설계	63	DMS 테이블 공간의 컨테이너 수정	80
분산 데이터베이스 설계	63	데이터베이스 액세스 제어	80
다중 데이터베이스 갱신	63	순차 특권	80
Windows NT 환경의 고가용성	64	데이터 암호화	81
DB2MSCS 유틸리티 수행 전에 머신 재부트	64	데이터베이스 복구	82
Sun Cluster 2.2 상의 DB2와 고가용성	64	중지된 입출력 사용 방법	82
Solaris에서 Veritas 지원	64	중분 백업 및 복구	84
이름 지정 규칙	66	중분 백업 이미지로부터 복원	85
8자를 초과하는 사용자 ID 및 스키마 이름에 대		병렬 복구	86
한 주의사항	66	Named Pipe에 백업	87
사용자 ID 및 암호	66	분리 이미지로부터 백업	87
릴리스 간의 비호환성	66	요구에 따른 로그 아카이브	90
Norton's Utilities와 Windows NT DLFS 비호		로그 미러링	90
환	66	Sun Solaris와 HP에 대한 교차 플랫폼 백업 및	
SET CONSTRAINTS를 SET INTEGRITY로		복원 지원	91
교체	67	DB2 Data Links Manager 고려사항/백업 유틸리	
자국어 지원	67	티 고려사항	91
DB2 버전 7 자국어 버전	67	DB2 Data Links Manager 고려사항/복원 및 롤	
제어 센터 및 문서 파일 세트	69	포워드 유틸리티 고려사항	91
DAS(DB2 Administrator Server)에 대한 로케일		롤 포워드하지 않고 오프라인 백업으로부터 데이	
설정	71	터베이스 복원	93
DB2 UDB는 Windows 플랫폼에서 Baltic Rim		데이터베이스와 테이블 공간 복원 및 로그 끝까지	
코드 페이지(MS-1257)를 지원합니다	71	롤 포워드	94
코드 페이지 값 추론	71	DB2 Data Links Manager 및 복구 상호작용	94
국가 코드 및 코드 페이지 지원	71	조정이 필요한 상황 감지	98
문자 세트	71	데이터베이스 복구를 위한 User Exit	98
관리 안내서: 구현	73	다중 데이터베이스 파티션 서버로 명령 발행	99
DMS 컨테이너 추가 또는 확장(새로운 프로세스)	73	고속 노드간 통신	100
GUI 도구를 사용하여 DB2 관리	73	VI를 사용하여 수행할 수 있는 DB2	100
데이터베이스 작성	74	관리 안내서: 성능	103
테이블 공간 작성	74	응용프로그램 고려사항	103
Linux에서 원시 입출력 사용	74	분리 레벨 지정	103
순차 작성	75	최적화 클래스 조정	103
IDENTITY 컬럼과 순차 비교	77	동적 복합 텍스트 명령문	104
색인, 색인 확장 또는 색인 스펙 작성	78	환경 고려사항	104
데이터베이스 변경	78	큰 색인 키 사용	105
파티션의 SMS 테이블 공간에 컨테이너 추가	78	시스템 카탈로그 통계	105
식별 컬럼 변경	78	분산 통계 수집 및 사용	105

카탈로그 통계 갱신 규칙	106	db2ckrst(새 명령)	141
부속 구성요소 통계	106	db2ckrst - 증분 복원 이미지 순차 점검	142
SQL 컴파일러 이해	109	db2gncol(새 명령)	143
복제된 요약 테이블	109	db2gncol - 생성된 컬럼 값 갱신.	144
데이터 액세스 개념 및 최적화.	110	db2inidb - 미러된 데이터베이스 초기화	145
조작 성능	111	db2look - DB2 통계 추출 도구.	146
데이터베이스 버퍼 풀 관리.	111	db2updv7 - 버전 7 현재 수정 레벨로 데이터베이	
다중 데이터베이스 버퍼 풀 관리.	112	스 갱신	146
조정자(governor) 사용	113	새 명령행 처리기 옵션(-x, 컬럼 표제의 인쇄 억제)	148
DB2 구성	113	DB2 CLP에 대한 트루 타입 글꼴 요구사항	148
정렬 힙 크기(sortheap)	113	ADD DATALINKS MANAGER	148
정렬 힙 임계값(sheapthres).	114	ARCHIVE LOG (새 명령)	148
레벨 자동 업그레이드 전의 잠금 목록 최대 퍼센		Archive Log	149
트(maxlocks)	114	BACKUP DATABASE	151
DB2/DB2 Data Links Manager/Data Links 액		구문 도표	151
세스 토큰 만료 간격 (dl_expint) 구성	115	DB2 Data Links Manager 고려사항	151
MIN_DEC_DIV_3 데이터베이스 구성 매개변수	115	BIND.	152
응용프로그램 제어 힙 크기(app_ctl_heap_sz)	116	CALL	153
데이터베이스 시스템 모니터 힙(heap) 크기		DROP DATALINKS MANAGER (새 명령)	153
(mon_heap_sz)	117	DROP DATALINKS MANAGER.	154
실행 중인 응용프로그램의 최대 수 (maxappls)	117	EXPORT	158
복구 범위 및 소프트 점검점 간격(softmax)	117	GET DATABASE CONFIGURATION	158
수정된 페이지 추적 사용 (trackmod)	117	GET ROUTINE (새 명령)	159
데이터베이스 로그 경로 변경 (newlogpath)	117	GET ROUTINE	160
로그 파일의 위치 (logpath)	119	GET SNAPSHOT	160
잠금 목록용 최대 저장영역 (locklist)	119	IMPORT	161
DB2 레지스트리 및 환경 변수	119	LIST HISTORY	161
새로운 레지스트리 변수 및 변경된 레지스트리		LOAD	162
변수의 표	119	PING (새 명령)	163
SQL Explain 도구	124	PING.	164
Administering Satellites 안내 및 참조서	125	PUT ROUTINE (새 명령)	165
버전 7.2 DB2 Personal Edition 및 DB2		PUT ROUTINE	166
Workgroup Edition을 위성으로 설정	125	RECONCILE	166
전제조건	125	REORGANIZE TABLE	167
설치 고려사항	126	RESTORE DATABASE	168
동기화하기 위해 버전 7.2 시스템 구성	128	구문	168
버전 6 Enterprise Edition 시스템에 FixPak 2		DB2 Data Links Manager 고려사항	169
이상 설치	129	ROLLFORWARD DATABASE	169
DB2 제어 서버로 사용하기 위해 버전 6		CLP 리턴 코드의 문서 오류	169
DB2 Enterprise Edition 업그레이드	129	데이터 이동 유틸리티 안내 및 참조서	171
버전 6 제어 센터 및 위성 관리 센터 업그레이드	137	가져오기	171
Command Reference	139	버퍼된 삽입이 있는 가져오기 사용	171
db2batch - 벤치마크 도구	139	로드	171
db2cap(새 명령)	139	로드 조작 이후 보류 상태	171
db2cap - CLI/ODBC 정적 패키지 바인딩 도구	140	로드 제한사항 및 한계	171
		totalfreespace 파일 유형 수정자	171

자동 로드 프로그램	172	제한사항	191
Authentication=YES일 경우 자동 로드 프로그		XML Extender 관리 및 프로그래밍	193
램 수행에 rexecd 필요	172	MQSeries	195
복제 안내 및 참조서	173	DB2 MQSeries 함수 설치 및 구성	195
복제 및 IBM이 아닌 서버	173	MQSeries 설치	195
Windows 2000에서의 복제	173	MQSeries AMI 설치	196
SQL 파일 저장 시 알려진 오류	173	DB2 MQSeries 함수 사용	196
DB2 유지보수	173	MQSeries 메시지 스타일	197
웹의 Data Difference 유틸리티	173	메시지 구조	197
데이터 복제 시나리오	174	MQSeries 기능 개요	198
복제 시나리오	174	제한사항	199
복제 계획	175	오류 코드	199
테이블 및 컬럼 이름	175	사용 시나리오	199
DATALINK 복제	175	기본 메시지	200
LOB 제한사항	176	메시지 전송	201
복제 계획	176	메시지 검색	202
복제 환경 설정	176	응용프로그램간 연결성	204
update-anywhere 전제조건	176	요청/응답 통신	204
복제 환경 설정	176	Publish/Subscribe	205
문제점 판별	177	enable_MQFunctions	208
AS/400용 Capture 및 Apply	178	enable_MQFunctions	209
OS/390용 Capture 및 Apply	179	disable_MQFunctions	210
OS/390용 DB2 DataPropagator의 전제조건	179	disable_MQFunctions	211
OS/390에서의 유니코드 및 ASCII 코드화 체계	179		
코드화 체계 선택	179		
코드화 체계 설정	180		
UNIX 플랫폼용 Capture 및 Apply	181		
UNIX 및 Windows용 Capture 및 Apply에 대			
한 환경 변수 설정	181		
테이블 구조	181		
Capture 및 Apply 메시지	181		
응용프로그램 내에서 Capture 및 Apply 프로그램			
시작	182		
시스템 모니터 안내 및 참조서	183		
db2ConvMonStream	183		
문제점 해결 안내서	185		
사용자가 로그인되지 않았을 때 Windows 95,			
Windows 98 및 Windows ME에서 DB2 시작	185		
DB2 Universal Database Server 문제점 해결	189		
64 비트 플랫폼에서 DB2 Universal Database 사			
용	191		
구성	191		
LOCKLIST	191		
shmsys:shminfo_shmmax	191		

관리 안내서: 계획

물리적 데이터베이스 설계

파티션 키

『노드 그룹 설계』 절 아래의 『노드 그룹 설계 고려사항』 서브 섹션에서, 파티션 키 정의 시 고려해야 할 사항을 설명하는 절에서 DB2_UPDATE_PART_KEY=ON일 경우에만 다음 텍스트를 삭제해야 합니다.

주: DB2_UPDATE_PART_KEY=OFF일 경우, 제한사항이 계속 적용됩니다.

주: FixPak 3 이후에서, 기본 값은 ON입니다.

- 테이블의 행에 대해 파티션 키 컬럼 값을 갱신할 수 없습니다.
- 파티션 키 컬럼 값을 삭제 또는 삽입만 할 수 있습니다.

노드 그룹 설계

『노드 그룹 설계』 절 아래의 『노드 그룹 설계 고려사항』 및 『복제된 요약 테이블』 서브 섹션에서, 두 번째 단락의 마지막 문장인 다음 문장을 무시하십시오.

REPLICATED 키워드는 REFRESH DEFERRED 옵션으로 정의된 요약 테이블에만 지정될 수 있습니다.

분산 데이터베이스 설계

다중 데이터베이스 갱신

"다중 데이터베이스 갱신" 절에서, 설정 단계 목록이 정확하지 않습니다. 4 단계의 내용이 다음과 같이 나타나 있습니다.

응용프로그램 개발 안내서에 설명된 대로
응용프로그램을 사전 처리 컴파일하도록 유형 2 연결을 지정하고
(PRECOMPILE PROGRAM 명령에 CONNECT 2 지정),
일단계 확약을 지정하십시오
(PRECOMPILE PROGRAM 명령에 SYNCPOINT ONEPHASE 지정).

이 문장은 다음과 같이 바뀌어야 합니다.

응용프로그램 개발 안내서에 설명된 대로
응용프로그램을 사전 처리 컴파일하도록 유형 2 연결을 지정하고
(PRECOMPILE PROGRAM 명령에 CONNECT 2 지정),
일단계 확약을 지정하십시오
(PRECOMPILE PROGRAM 명령에 SYNCPOINT ONEPHASE 지정).

Windows NT 환경의 고가용성

DB2MSCS 유틸리티 수행 전에 머신 재부트

DB2MSCS 유틸리티는 Fail-Over 지원을 위한 DB2를 Microsoft Cluster Service 환경 하에서 사용하기 위해 필요한 설정 수행에 사용됩니다. DB2MSCS 유틸리티가 성공적으로 수행되려면, Cluster Service가 %ProgramFiles%\SQLLIB\bin 디렉토리 아래에 있는 자원 DLL, db2wolf.dll을 찾을 수 있어야 합니다.

DB2 UDB 버전 7 설치 프로그램은 PATH 시스템 환경 변수가 %ProgramFiles%\SQLLIB\bin 디렉토리를 가리키도록 설정합니다. 하지만, Windows 2000 운영 체제를 수행할 경우에는 설치 후에 머신을 재시작하지 않아도 됩니다.

DB2MSCS 유틸리티를 수행하려면, PATH 환경 변수가 Cluster Service에 대해 갱신 되도록 머신을 재부트해야 합니다.

Sun Cluster 2.2 상의 DB2와 고가용성

DB2 Connect는 다음과 같은 경우에 Sun Cluster 2.2에서 지원됩니다.

- 호스트에 대한 프로토콜이 TCP/IP인 경우(SNA가 아님)
- 2 단계 확약이 사용되지 않는 경우. 이 제한사항은 사용자가 SPM 로그를 공유 디스크에 위치하도록 구성하고(이 구성은 *spm_log_path* 데이터베이스 관리 프로그램 구성 매개변수를 통해 수행할 수 있음) 실패 복구 머신에 동일한 TCP/IP 구성이 있는 경우(동일한 호스트 이름, IP 주소 등)에 완화됩니다.

Solaris에서 Veritas 지원

이제 DB2는 Solaris에서 DB2 고가용성을 위해 클러스터를 지원하는 Veritas를 지원 합니다.

설명 DB2 UDB 인스턴스를 온라인, 오프라인 및 모니터링합니다.

입력 지점

온라인 db2start를 사용하여 인스턴스를 활성화합니다.

오프라인

db2stop을 사용하여 인스턴스를 작동 종료합니다.

모니터 지정된 DB2 인스턴스가 작동 중인지 판별합니다. 적절한 프로세스 모니터링 및 (선택적) 데이터베이스 모니터링을 사용합니다.

제거 DB2 인스턴스 자원을 제거합니다.

속성	유형	정의
probeDatabase	string	모니터할 데이터베이스

instanceOwner	string	인스턴스 소유자 이름
instanceHome	string	인스턴스 소유자의 홈 디렉토리
probeTable	string	모니터할 probeDatabase의 테이블
monitorLevel	integer	1은 프로세스 모니터링을 의미하며, 2는 db 모니터링을 의미합니다
nodeNumber	integer	시작할 인스턴스의 nodeNumber(설정 제거는 EE)

유형 정의

```

type DB2UDB (
    static int CleanTimeout = 240
    static int MonitorTimeout = 30
    static int OfflineTimeout = 240
    static int OnlineRetryLimit = 2
    static int OnlineTimeout = 120
    static int OnlineWaitLimit = 1
    static int RestartLimit = 3
    static int ToleranceLimit = 1
    static str ArgList[] =
    { probeDatabase, instanceOwner, instanceHome, probeTable, monitorLevel, nodeNumber }
    NameRule = resource.db2udb
    str probeDatabase
    str instanceOwner
    str instanceHome
    str probeTable
    int monitorLevel
    int nodeNumber
)

```

샘플 구성

```

DB2UDB db2_resource_n0 (
    probeDatabase = sample
    probeTable = vcstable
    instanceOwner = stevera
    instanceHome = "/export/home/stevera"
    monitorLevel = 2
)

```

설치

- 디렉토리 /opt/VRTSvcs/bin/DB2UDB를 작성하십시오.
- 파일 online, offline, monitor, clean, DB2UDBAgent를 /opt/VRTSvcs/bin/DB2UDB에 복사하고 실행으로 표시되었는지 확인하십시오.
- 파일 db2udb.type.cf를 /etc/VRTSvcs/conf/config에 복사하십시오.
- 클러스터를 중지하십시오(예를 들어, hastop -all).
- include db2udb.type.cf 행을 파일 main.cf의 include types.cf 행 뒤에 추가하십시오.
- /opt/VRTSvcs/bin/hacf -verify /etc/VRTSvcs/conf/config로 클러스터 구성을 검증하십시오.

이제 DB2 인스턴스 제어에 필요한 DB2 자원을 작성할 준비가 되었습니다.

이름 지정 규칙

8자를 초과하는 사용자 ID 및 스키마 이름에 대한 주의사항

- Windows 32 비트 플랫폼의 DB2 버전 7 제품이 지원하는 사용자 ID 최대 길이는 30자입니다. Windows NT 및 Windows 2000이 20자 이상의 ID를 지원하지 않으므로 사용자 ID는 실제로 20자까지 지정할 수 있습니다.
- DB2 버전 7은 사용자 ID와 암호가 명시적으로 지정되는 경우에 Windows NT 및 Windows 2000에 연결하는 비 Windows 32 비트 클라이언트에 대해 8자 이상의 긴 사용자 ID를 지원합니다. 클라이언트 또는 DCE 인증을 사용한 연결은 이에 해당되지 않습니다.
- 모든 플랫폼에서 DCE 인증은 8자 이하의 사용자 ID를 사용해야만 합니다.
- 성공한 CONNECT 또는 ATTACH로부터 SQLCA에 리턴된 AuthID는 8자에서 잘립니다. 이런 경우, SQLWARN 필드에는 경고가 표시됩니다. 자세한 정보는 *SQL* 참조서에서 CONNECT 명령문에 대한 설명을 참조하십시오.
- 명령행 처리기(CLP)에서 성공한 CONNECT 또는 ATTACH로부터 리턴된 AuthID는 8자에서 잘립니다. 생략 부호(...)가 AuthID에 추가되어 ID가 잘렸다는 것을 표시해 줍니다.
- DB2 버전 7은 다음의 경우를 제외하고는 최대 30 바이트의 스키마 이름을 지원합니다.
 - 18 바이트보다 긴 스키마 이름이 있는 테이블은 복제할 수 없습니다.
 - 사용자 정의 유형(UDT)의 스키마 이름은 8 바이트보다 길 수 없습니다.

사용자 ID 및 암호

『사용자 ID 및 암호』 절 아래에서, 『A through Z』에 대한 언급을 다음과 같이 변경하십시오.

1 바이트 대문자 및 소문자 라틴 문자(A...Z,a...z). 다른 글자 및 문자 지원은 사용 중인 코드 페이지에 따라 다릅니다. 코드 페이지 지원에 대한 자세한 정보는 『자국 언어 지원(NLS)』 부록을 참조하십시오.

릴리스 간의 비호환성

Norton's Utilities와 Windows NT DLFS 비호환

Windows NT Data Links File System은 Norton Utilities와 호환되지 않습니다. DLFS가 제어하는 드라이브에서 파일을 삭제하면 다음 커널 예외가 발생합니다 - 오류 0x1E(커널 모드 예외가 처리되지 않았습니다). 0xC0000005 도중 예외가 발생했습니다(액세스 위반). 이 액세스 위반은 DLFS 필터 드라이버가 로드된 후 Norton Utilities 드라이버가 로드되므로 일어납니다.

일시적인 해결책은 Norton Utilities 드라이버가 로드된 후에, DLFS D 드라이버를 로드하는 것입니다. 이 일시적인 해결책은 DLFS D 드라이버 시작을 수동으로 변경하여 수행할 수 있습니다. 시작을 클릭한 후 설정 → 제어판 → 장치 → **DLFS D**를 선택하십시오.

시스템 시작시 DLFS D 드라이버와 DLFM Service를 로드하는 배치 파일을 작성하여, 시작 폴더에 추가할 수 있습니다. 배치 파일의 내용은 다음과 같습니다.

```
net start dlfsd
net start "dlfm service"
```

이 배치 파일의 이름을 start_dlfs.bat로 지정하고, C:\WINNT\Profiles\Administrator\Start Menu\Programs\Startup 디렉토리로 복사하십시오.

관리자만 DLFS 필터 드라이버와 DLFM 서비스를 로드할 특권을 갖습니다.

SET CONSTRAINTS를 SET INTEGRITY로 교체

SET CONSTRAINTS 문이 SET INTEGRITY 문으로 교체되었습니다. 이전과의 호환을 위해, 두 명령문 모두 DB2UDBV7에서 승인됩니다.

자국어 지원

DB2 버전 7 자국어 버전

DB2 버전 7은 영어, 프랑스어, 독일어, 이탈리아어, 스페인어, 브라질 포르투갈어, 일본어, 한국어, 중국어, 대만어, 덴마크어, 핀란드어, 노르웨이어, 스웨덴어, 체코어, 네덜란드어, 헝가리어, 폴란드어, 터키어, 러시아어, 불가리아어 및 슬로베니아어로 사용할 수 있습니다.

UNIX 기반 플랫폼에서, DB2 제품 메시지 및 라이브러리를 서로 다른 언어로 설치할 수 있습니다. DB2 설치 유틸리티는 다음 표에 예시된 것처럼 메시지 카탈로그 파일 세트를 주어진 플랫폼에서 가장 흔히 사용되는 로케일 디렉토리에 저장합니다. 표4는 AIX, HP-UX 및 Solaris에 대한 정보를 제공합니다. 69 페이지의 표5는 Linux, Linux/390, SGI 및 Dynix에 대한 정보를 제공합니다.

표 4. AIX, HP-UX, Solaris

운영 체제	AIX		HP-UX		Solaris	
언어	로케일	코드 페이지	로케일	코드 페이지	로케일	코드 페이지
프랑스어	fr_FR	850	fr_FR.iso88591	1051	fr	
	Fr_FR		fr_FR.roman8			

표 4. AIX, HP-UX, Solaris (계속)

독일어	de_DE	850	de_DE.iso88591	1051	de	
	De_DE		de_DE.roman8			
이탈리아어	it_IT	850	it_IT.iso88591	1051	it	
	It_IT		it_IT.roman8			
스페인어	es_ES	850	es_ES.iso88591	1051		
	Es_ES		es_ES.roman8			
브라질 포르투갈어	pt_BR				pt_BR	
일본어	ja_JP	954	ja_JP.eucJP	954	ja	954
	Ja_JP	932				
한국어	ko_KR	970	ko_KR.eucKR	970	ko	970
중국어	zh_CN	1383	zh_CN.hp15CN	1383	zh	1383
	Zh_CN.GBK	1386				
대만어	zh_TW	964	zh_TW.eucTW	964	zh_TW	964
	Zh_TW	950	zh_TW.big5	950	zh_TW.BIG5	950
덴마크어	da_DK	850	da_DK.iso88591	1051	da	
	Da_DK		da_DK.roman8			
핀란드어	fi_FI	850	fi_FI.iso88591	1051	fi	
	Fi_FI		fi_FI.roman8			
노르웨이어	no_NO	850	no_NO.iso88591	1051	no	
	No_NO		no_NO.roman8			
스웨덴어	sv_SE	850	sv_SE.iso88591	1051	sv	
	Sv_SE		sv_SE.roman8			
체코어	cs_CZ	912				
헝가리어	hu_HU	912				
폴란드어	pl_PL	912				
네덜란드어	nl_NL	850				
	Nl_NL					
터키어	tr_TR	920				
러시아어	ru_RU	915				
불가리아어	bg_BG	915	bg_BG.iso88595	915		
슬로비니아어	sl_SI	912	sl_SI.iso88592	912	sl_SI	912

표 5. Linux, Linux/390, SGI, Dynix

운영 체제	Linux		Linux/390		SGI		Dynix	
언어	로케일	코드 페이지	로케일	코드 페이지	로케일	코드 페이지	로케일	코드 페이지
프랑스어	fr		fr				fr	
독일어	de		de				de	
이탈리아어								
스페인어								
브라질 포르투갈어								
일본어	ja_JP.ujis	954	ja_JP.ujis	954			ja_JP.EUC	954
한국어	ko	970	ko	970	ko_KO.euc	970		
중국어	zh_zh_CN.GBK	1386	zh_zh_CN.GBK	1386				
대만어	zh_TW.Big5	950	zh_TW.Big5	950				
덴마크어								
핀란드어								
노르웨이어								
스웨덴어								
체코어								
헝가리어								
폴란드어								
네덜란드어							nl	
터키어								
러시아어								
불가리아어								
슬로베니아어								

시스템이 동일한 코드 페이지를 사용하지만 위에 제공된 것과 다른 로케일 이름을 사용하는 경우, 여전히 적합한 메시지 디렉토리에 대한 링크를 작성하여 번역된 메시지를 볼 수 있습니다.

예를 들어, AIX 머신 기본 로케일이 ja_JP.IBM-eucJP이고 ja_JP.IBM-eucJP의 코드 페이지가 954일 경우, 다음 명령을 실행하여 /usr/lpp/db2_07_01/msg/ja_JP.IBM-eucJP로부터 /usr/lpp/db2_07_01/msg/ja_JP로 링크를 작성할 수 있습니다.

ln -s /usr/lpp/db2_07_01/msg/ja_JP /usr/lpp/db2_07_01/msg/ja_JP.IBM-eucJP

이 명령을 실행하고 나면, 모든 DB2 메시지가 일본어로 바뀝니다.

제어 센터 및 문서 파일 세트

제어 센터, 제어 센터 도움말 및 문서 파일 세트는 목표 워크스테이션에서 다음 디렉토리에 있습니다.

- AIX용 DB2

- /usr/lpp/db2_07_01/cc/%L
- /usr/lpp/db2_07_01/java/%L
- /usr/lpp/db2_07_01/doc/%L
- /usr/lpp/db2_07_01/qp/\$L
- /usr/lpp/db2_07_01/spb/%L

- HP-UX용 DB2

- /opt/IBMd2/V7.1/cc/%L
- /opt/IBMd2/V7.1/java/%L
- /opt/IBMd2/V7.1/doc/%L

- Linux용 DB2

- /usr/IBMd2/V7.1/cc/%L
- /usr/IBMd2/V7.1/java/%L
- /usr/IBMd2/V7.1/doc/%L

- Solaris용 DB2

- /opt/IBMd2/V7.1/cc/%L
- /usr/IBMd2/V7.1/java/%L
- /opt/IBMd2/V7.1/doc/%L

제어 센터 파일 세트는 유니코드 코드 페이지에 있습니다. 문서 및 제어 센터 도움말 파일 세트는 브라우저 인식 코드 세트로 되어 있습니다. 시스템이 제공된 것과 다른 로케일 이름을 사용하는 경우, 여전히 적합한 언어 디렉토리에 대한 링크를 작성하여 제어 센터의 변환된 버전을 수행하고 도움말의 변환된 버전을 볼 수 있습니다.

예를 들어, AIX 머신 기본 로케일이 ja_JP.IBM-eucJP일 경우, 다음 명령을 실행하여 /usr/lpp/db2_07_01/cc/ja_JP.IBM-eucJP to /usr/lpp/db2_07_01/cc/ja_JP and from /usr/lpp/db2_07_01/doc/ja_JP.IBM-eucJP to /usr/lpp/db2_07_01/doc/ja_JP로 링크를 작성할 수 있습니다.

- **ln -s /usr/lpp/db2_07_01/cc/ja_JP /usr/lpp/db2_07_01/cc/ja_JP.IBM-eucJP**
- **ln -s /usr/lpp/db2_07_01/doc/ja_JP /usr/lpp/db2_07_01/doc/ja_JP.IBM-eucJP**

이들 명령을 실행하면, 제어 센터와 도움말 텍스트가 일본어로 표시됩니다.

주: 웹 제어 센터는 Linux/390 또는 NUMA-Q에서 본래 수행이 지원되지 않습니다. 이러한 플랫폼의 데이터베이스를 관리하기 위해 클라이언트 워크스테이션에서 웹 제어 센터를 사용할 수 있습니다.

DAS(DB2 Administrator Server)에 대한 로케일 설정

DAS(DB2 Administrator Server) 인스턴스의 로케일이 DB2 인스턴스 로케일과 호환되는지 확인하십시오. 그렇지 않으면, DB2 인스턴스는 DAS와 통신할 수 없습니다.

LANG 환경 변수가 DAS의 사용자 프로파일에 설정되지 않은 경우, DAS는 기본 시스템 로케일로 시작됩니다. 기본 시스템 로케일이 정의되어 있지 않으면, 코드 페이지 819로 DAS가 시작됩니다. DB2 인스턴스가 DBCS 로케일 중 하나를 사용하고, DAS가 코드 페이지 819로 시작될 경우, 인스턴스는 DB2 관리 서버와 통신할 수 없게 됩니다. DAS의 로케일과 DB2 인스턴스의 로케일은 호환 가능해야 합니다. 예를 들어, 중국어 Linux 시스템에서 DAS의 사용자 프로파일에 "LANG=zh_CN"가 설정되어야 합니다.

DB2 UDB는 Windows 플랫폼에서 Baltic Rim 코드 페이지(MS-1257)를 지원합니다

DB2 UDB는 Windows 32 비트 운영 체제에서, Baltic Rim 코드 페이지, MS-1257을 지원합니다. 이 코드 페이지는 라트비아어, 리투아니아어 및 에스토니아어에 사용됩니다.

코드 페이지 값 추론

『코드 페이지 값 추론』이라는 절에서, 첫번째 단락은 다음과 같습니다.

하지만, DB2가 운영 체제로부터 적절한 코드 페이지 값을 판별하므로 DB2CODEPAGE 레지스트리 변수를 반드시 설정할 필요는 없습니다.

이 단락을 다음과 같이 변경하십시오. 보통, DB2가 운영 체제로부터 코드 페이지 정보를 자동으로 추론하므로 DB2CODEPAGE 레지스트리 변수를 설정할 필요는 없습니다.

국가 코드 및 코드 페이지 지원

『국가 코드 및 코드 페이지 지원』이라는 절에서, 다음 정보를 테이블에 추가하십시오.

코드 페이지	그룹	코드-세트	Tr.	국가 코드	로케일	OS	국가 이름
943	D-1	IBM-943	JP	81	ja_JP.PCK	Sun	Japan

문자 세트

『문자 세트』라는 절과 서브섹션 『식별자에 대한 문자 세트』에서, 첫번째 단락의 마지막 두 문장을 다음으로 바꾸십시오.

특수 문자 #, @ 및 \$는 NLS 호스트(EBCDIC) 불변 문자 세트에 포함되어 있지 않으므로 주의해서 사용하십시오. 사용 중인 코드 페이지에 따라, 확장 문자 세트의 문자도 사용할 수 있습니다. 다중 코드 페이지 환경에서 데이터베이스를 사용할 경우, 모든 코드 페이지가 사용하려는 확장 문자 세트의 요소를 지원하는지 확인해야 합니다.

관리 안내서: 구현

DMS 컨테이너 추가 또는 확장(새로운 프로세스)

DMS 컨테이너(파일 컨테이너 및 원시 장치 컨테이너 모두)는 이제 프리페처를 통해 병렬로 추가(테이블 공간 작성 중 또는 그 후) 또는 확장될 수 있습니다. 이런 컨테이너 작성/크기 조정 작업을 더 많이 병렬로 처리하려면, 시스템에서 수행되는 프리페처의 수를 증가시킬 수 있습니다. 병렬로 처리되지 않는 유일한 프로세스는 이런 조치들의 로깅 그리고 컨테이너 작성의 경우에 컨테이너의 태그 기록입니다.

주: prefetcher의 수가 추가되는 컨테이너의 수와 같을 때는 CREATE TABLESPACE / ALTER TABLESPACE의 병렬 처리(기존 테이블 공간에 새 컨테이너 추가에 관하여)는 더 이상 증가되지 않습니다.

GUI 도구를 사용하여 DB2 관리

『경보 센터』라는 절에서, 절의 마지막 두 문장을 제거하십시오.

『성능 모니터』 절에서, 서브섹션 『특정 시점에서 성능 모니터링』의 『성능 변수 정의』 목록에서 두 번째 bullet 항목을 제거하십시오.

또한 같은 서브섹션의 뒷부분에서, 절의 마지막 몇 개의 문단을 다음과 같이 다시 작성해야 합니다.

각각에 대해, 다양한 성능 변수를 모니터할 수 있습니다. 스냅샷 모니터 창의 도움말 메뉴에 있는 성능 변수 참조 도움말은 모든 성능 변수의 설명을 제공합니다. 이러한 변수는 범주로 조직화됩니다. 다음과 같은 범주가 있습니다.

- 인스턴스: 에이전트, 연결, 정렬
- 데이터베이스: 잠금 및 교착 상태, 버퍼 풀 및 I/O, 연결, 정렬, SQL문 활동
- 테이블: 테이블
- 테이블 공간: 버퍼 풀 및 I/O
- 데이터베이스 연결: 버퍼 풀 및 I/O, 잠금 및 교착 상태, 정렬, SQL문 활동

스냅샷 생성 방법의 세부사항 정보는, 온라인 도움말을 참조하십시오.

같은 절에서, 『오브젝트가 경보 센터에 나타날 때 필요한 조치』라는 서브섹션의 마지막 문장을 제거하십시오.

데이터베이스 작성

테이블 공간 작성

Linux에서 원시 입출력 사용

Linux에는 원시 입출력이 수행되려면 블록 장치에 바인드되어야 하는 원시 장치 노드 풀이 있습니다. 원시와 블록 장치 바인딩 정보의 중앙 저장소로서 기능하는 원시 장치 제어기가 있습니다. 바인딩은 raw라는 유틸리티를 사용하여 수행되며, 이 유틸리티는 Linux 공급자가 제공합니다.

Linux에 원시 입출력을 설정하기 전에, 다음이 필요합니다.

- 하나 이상의 사용 가능한 IDE 또는 SCSI 디스크 파티션
- Linux 커널 2.4.0 이상(하지만, 일부 Linux 분산은 2.2 커널에서 원시 입출력을 제공합니다.)
- 이름이 /dev/rawctl 또는 /dev/raw인 원시 장치 제어기. 없을 경우, 기호 링크를 작성하십시오.

```
# ln -s /dev/your_raw_dev_ctrl /dev/rawctl
```
- Linux 분산과 함께 제공되는 원시 유틸리티
- DB2 버전 7.1 FixPak 3 이상

주: 현재 원시 입출력을 지원하는 분산 중에, 원시장치 노드의 이름 지정이 서로 다릅니다.

분산	원시 장치 노드	원시 장치 제어기
RedHat 6.2	/dev/raw/raw1 to 255	/dev/rawctl
SuSE 7.0	/dev/raw1 to 63	/dev/raw

DB2는 위의 원시 장치 제어기 및 원시 장치 노드의 대부분의 다른 이름을 지원합니다. 원시 장치는 Linux/390의 DB2에서는 지원되지 않습니다.

Linux에서 원시 입출력을 구성하려면 다음을 수행하십시오.

이 예에서, 사용되는 원시 파티션은 /dev/sda5입니다. 중요한 데이터가 들어 있어서는 안 됩니다.

단계 1. 이 파티션에서 4 096 바이트 페이지의 수를 계산하되, 필요할 경우 남는 바이트는 버리십시오. 예를 들어, 다음과 같습니다.

```
# fdisk /dev/sda
명령(도움말은 m): p

Disk /dev/sda: 255 헤드, 63 섹터, 1106 실린더
단위 = 16065 * 512 바이트의 실린더
```

장치	부트	시작	종료	블록	Id	시스템
/dev/sda1		1	523	4200997	83	Linux
/dev/sda2		524	1106	4682947+	5	확장
/dev/sda5		524	1106	4682947	83	Linux

```
명령 (도움말은 m): q
#
```

/dev/sda5의 페이지 수는

```
num_pages = floor( ((1106-524+1)*16065*512)/4096 )
num_pages = 11170736 입니다.
```

단계 2. 미사용 원시 장치 노드를 이 파티션에 바인드하십시오. 머신을 재부트할 때마다 이를 수행해야 하며, 루트 액세스가 필요합니다. raw -a를 사용하여 어떤 원시 장치 노드가 이미 사용 중인지 보십시오.

```
# raw /dev/raw/raw1 /dev/sda5
/dev/raw/raw1: bound to major 8, minor 5
```

단계 3. 원시 장치 제어기 및 디스크 파티션에 전역 읽기 권한을 설정하십시오. 원시 장치에 전역 읽기 및 쓰기 권한을 설정하십시오.

```
# chmod a+r /dev/rawctl
# chmod a+r /dev/sdb1
# chmod a+rw /dev/raw/raw1
```

단계 4. 디스크 파티션이 아닌, 원시 장치를 지정하여, DB2에 테이블 공간을 작성하십시오. 예를 들어, 다음과 같습니다.

```
CREATE TABLESPACE dms1
MANAGED BY DATABASE
USING (DEVICE '/dev/raw/raw1' 11170736)
```

원시 장치의 테이블 공간은 DB2가 지원하는 모든 다른 페이지 크기에도 지원됩니다.

순차 작성

『새 테이블에 식별 컬럼 정의』라는 절 뒤에, 다음 절 "순차 작성"을 추가하십시오.

순차는 값의 자동 생성을 허용하는 데이터베이스 오브젝트입니다. 순차는 고유한 키 값을 생성하는 task에 매우 적합합니다. 응용프로그램은 순차를 사용하여, 데이터베이스 외부에서 고유한 카운터(counter) 생성의 결과로 나타날 수 있는 동시성 및 성능 문제점을 방지할 수 있습니다.

식별 컬럼 속성과는 달리, 순차는 특정 테이블 컬럼에 연결되지 않고 고유한 테이블 컬럼에 바인드되지 않으며 해당 테이블 컬럼을 통해서만 액세스 가능합니다.

순차는 작성 또는 변경할 수 있으므로, 다음 방법 중 하나로 값을 생성합니다.

- 바인드되지 않고 단조롭게 증분 또는 감소
- 사용자 정의 한계까지 단조롭게 증분 또는 감소하고 중지

- 사용자 정의 한계까지 단조롭게 증분 또는 감소 그리고 다시 시작 부분으로 돌아가 다시 시작

다음은 순차 오브젝트 작성의 예입니다.

```
CREATE SEQUENCE order_seq
  START WITH 1
  INCREMENT BY 1
  NOMAXVALUE
  NOCYCLE
  CACHE 24
```

이 예에서, 순차의 이름은 `order_seq`입니다. 1에서 시작하여 상한없이 1씩 증가합니다. 지정된 상한이 없으므로 시작 부분으로 돌아가 1부터 다시 시작할 필요가 없습니다. CACHE 매개변수와 연관된 숫자는 데이터베이스 관리자가 사전 할당하여 메모리에 보존하는 순차 값의 최대 수를 지정합니다.

생성된 순차 번호에는 다음 특성이 있습니다.

- 스케일이 영(0)인 정확한 숫자 데이터 유형이 값이 될 수 있습니다. 이와 같은 데이터 유형에는 SMALLINT, BIGINT, INTEGER 및 DECIMAL이 포함됩니다.
- 연속되는 값은 지정된 정수 증분에 따라 다를 수 있습니다. 기본 증분 값은 1입니다.
- 카운터(counter) 값은 복구 가능합니다. 카운터 값은 복구가 필요할 때 로그로부터 재구성됩니다.
- 성능을 향상시키기 위해 값을 캐쉬할 수 있습니다. 캐쉬에서 값을 사전할당하고 저장하면 순차에 대한 값이 생성될 때 로그에 대한 동기 입출력을 감소시킵니다. 시스템 장에서, 확장되지 않은 캐쉬된 모든 값은 전혀 사용되지 않으며 유실된 것으로 간주됩니다. CACHE에 지정하는 값은 유실될 수 있는 순차 값의 최대수입니다.

하나 이상의 순차가 들어 있는 데이터베이스가 이전 시점으로 복구될 경우, 일부 순차에 대해 중복 값 생성을 초래할 수 있습니다. 중복 값을 방지하려면, 순차가 있는 데이터베이스를 이전 시점으로 복구해서는 안됩니다.

순차는 단일 노드 데이터베이스에서만 지원됩니다.

순차에는 두 개의 표현식이 사용됩니다.

PREVVAL 표현식은 현재 세션 내에서 이전 문장에 대해 지정된 순차의 가장 최근에 생성된 값을 리턴합니다.

NEXTVAL 표현식은 지정된 순차의 다음 값을 리턴합니다. NEXTVAL 표현식이 순차의 이름을 지정하면 새로운 순차 번호가 생성됩니다. 그러나, 조회 내에서 동일한 순차 이름을 지정하는 NEXTVAL 표현식의 여러 인스턴스가 있는 경우, 순차의 카운터(counter)는 결과의 각 행에 대해 한번만 증가됩니다.

별개의 두 테이블에서 동일한 순차 번호를 고유 키로 사용할 수 있습니다. 첫번째 테이블에 대해서는 NEXTVAL 표현식으로 순차 번호를 참조하고, 다른 테이블에 대해서는 PREVVAL 표현식으로 참조합니다.

예를 들어, 다음과 같습니다.

```
INSERT INTO order (orderno, custno)
VALUES (NEXTVAL FOR order_seq, 123456);
INSERT INTO line_item (orderno, partno, quantity)
VALUES (PREVVAL FOR order_seq, 987654, 1)
```

NEXTVAL 또는 PREVVAL 표현식을 다음 위치에서 사용할 수 있습니다.

- INSERT 문, VALUES 절
- SELECT 문, SELECT 목록
- SET 지정 문
- UPDATE 문, SET 절
- VALUES 또는 VALUES INTO 문

IDENTITY 컬럼과 순차 비교

『순차 작성』이라는 새 절 뒤에, 다음 절을 추가하십시오.

IDENTITY 컬럼과 순차 사이에 유사점이 있는 반면, 차이점도 있습니다. 데이터베이스와 응용프로그램 설계시 각각의 특성을 사용할 수 있습니다.

식별 컬럼에는 다음 특성이 있습니다.

- 식별 컬럼은 테이블이 작성된 때에만 테이블의 일부로서 정의될 수 있습니다. 일단 테이블이 작성되면, 그것을 변경하거나 식별 컬럼을 추가할 수 없습니다(하지만, 기존 식별 컬럼 특성은 변경될 수 있습니다).
- 식별 컬럼은 자동으로 단일 테이블에 대한 값을 생성합니다.
- 식별 컬럼은 GENERATED ALWAYS로 정의하면, 사용되는 값은 데이터베이스 관리 프로그램에 의해 항상 생성됩니다. 테이블 내용의 수정 중에 응용프로그램은 자신의 값을 제공할 수 없습니다.

순차 오브젝트에는 다음 특성이 있습니다.

- 순차 오브젝트는 어떤 한 테이블에 연결되지 않은 데이터베이스 오브젝트입니다.
- 순차 오브젝트는 어떤 SQL문에서도 사용될 수 있는 순차 값을 생성합니다.
- 응용프로그램에서 순차 오브젝트를 사용할 수 있으므로, 지정된 순차에서 다음 값 그리고 실행 중인 문 이전에 생성된 값의 검색을 제어하는 데 사용되는 두가지 표현식이 있습니다. PREVVAL 표현식은 현재 세션 내에서 이전 문장에 대해 지정된 순차의 가장 최근에 생성된 값을 리턴합니다. NEXTVAL 표현식은 지정된 순차의 다음 값을 리턴합니다. 이들 표현식을 사용하여, 여러 테이블 내의 여러 SQL문에 걸쳐 동일한 값을 사용할 수 있습니다.

이들이 두 항목의 모든 특성은 아니지만, 이런 특성은 데이터베이스 설계 및 데이터베이스를 사용하는 응용프로그램에 따라 어떤 것을 사용해야 할지 결정하는데 도움이 될 것입니다.

색인, 색인 확장 또는 색인 스펙 작성

『색인 작성, 색인 확장 또는 색인 스펙』이라는 절 내에서, 『색인 키의 일부인 컬럼은 255 바이트로 제한됩니다』라는 문장으로 시작하는 단락에 다음 주를 추가하십시오.

주: DB2_INDEX_2BYTEVARLEN 레지스트리 변수는 길이가 255 바이트 이상인 컬럼을 색인 키의 일부로 지정할 수 있도록 하기 위해 사용됩니다.

데이터베이스 변경

『테이블 공간 변경』 절에 다음의 새로운 절이 추가되었습니다.

파티션의 SMS 테이블 공간에 컨테이너 추가

현재 컨테이너가 없는 파티션(또는 노드)의 SMS 테이블 공간에 컨테이너를 추가할 수 있습니다.

테이블 공간의 내용은 모든 컨테이너에서 다시 균형을 유지합니다. 테이블 공간에 대한 액세스는 재균형 유지 중에 제한되지 않습니다. 하나 이상의 컨테이너를 추가해야 하는 경우, 모두 동시에 추가해야 합니다.

명령행을 사용하여 SMS 테이블 공간에 컨테이너를 추가하려면, 다음을 입력하십시오.

```
ALTER TABLESPACE <name>
  ADD ('<path>')
  ON NODE (<partition_number>)
```

번호로 지정된 파티션과 파티션 범위 내의 모든 파티션(또는 노드)은 테이블 공간이 정의된 노드 그룹에 반드시 있어야 합니다. `partition_number`는 명시적으로만 나타나거나 명령문에 대해 정확히 하나의 `on-nodes-clause`에 나타날 수 있습니다.

다음의 예는 Unix 기반 운영 체제의 테이블 공간 『계획』에서 사용하는 노드 그룹의 파티션 번호 3에 새로운 컨테이너를 추가하는 방법을 보여줍니다.

```
ALTER TABLESPACE plans
  ADD ('/dev/rhdisk0')
  ON NODE (3)
```

『테이블 속성 변경』 절 뒤에, 다음 절을 추가하십시오.

식별 컬럼 변경

기존 식별 컬럼의 속성을 ALTER TABLE 문으로 수정하십시오. 이 구문을 포함하여, 이 명령문에 대한 자세한 정보는, SQL 참조서를 참조하십시오.

식별 컬럼이 순차의 특성 일부를 갖도록 식별 컬럼을 수정하는 여러 방법이 있습니다.

ALTER TABLE 및 식별 컬럼에 고유한 여러 태스크가 있습니다.

- RESTART는 식별 컬럼과 연관된 순차를 식별 컬럼이 본래 작성될 때 시작 값으로 암시적 또는 명시적으로 지정된 값으로 재설정합니다.
- RESTART WITH <숫자 상수>는 식별 컬럼과 연관된 순차를 정확한 숫자 상수 값으로 재설정합니다. 숫자 상수는 식별 컬럼에 지정될 수 있는 소수점 오른쪽에 영(0)이 아닌 숫자가 없는 양수 또는 음수입니다.

순차 변경

기존 순차의 속성을 ALTER SEQUENCE 문으로 수정하십시오. 이 구문을 포함하여, 이 명령문에 대한 자세한 정보는, *SQL 참조서*를 참조하십시오.

수정할 수 있는 순차의 속성은 다음과 같습니다.

- 미래의 값 사이의 증분 변경
- 새로운 최소 또는 최대 값 설정
- 캐쉬되는 순차 번호의 수 변경
- 순차가 순환하는지 여부 변경
- 순차 번호를 요청 순서로 생성해야 하는지 여부 변경
- 순차 재시작

순차 작성의 일부에서 찾을 수 없는 두 태스크가 있습니다. 다음과 같습니다.

- RESTART. 순차를 순차가 작성될 때 시작 값으로 암시적 또는 명시적으로 지정된 값으로 재설정합니다.
- RESTART WITH 숫자 상수. 정확한 숫자 상수 값으로 순차를 재설정합니다. 숫자 상수는 소수점 오른쪽에 영(0)이 아닌 숫자가 없는 양수 또는 음수입니다.

순차를 재시작하거나 CYCLE로 변경한 후에는, 중복 순차 번호가 생성될 가능성이 있습니다. 장래 순차 번호만이 ALTER SEQUENCE 문에 의해 영향을 받습니다.

순차의 데이터 유형은 변경될 수 없습니다. 대신, 현재 순차를 삭제한 뒤 새로운 데이터 유형을 지정하여 새 순차를 작성할 수 있습니다.

DB2에 의해 사용되지 않는 캐쉬된 모든 순차 값은 순차가 변경될 때 유실됩니다.

순차 삭제

순차를 삭제하려면, DROP 문을 사용하십시오. 이 구문을 포함하여, 이 명령문에 대한 자세한 정보는, *SQL 참조서*를 참조하십시오.

다음 명령을 사용하여 특정 순차를 삭제할 수 있습니다.

```
DROP SEQUENCE sequence_name
```

여기서 `sequence_name`은 삭제될 순차의 이름이고 기존 순차를 정확하게 식별하기 위해 암시적 또는 명시적 스키마 이름을 포함합니다.

IDENTITY 컬럼에 대해 시스템 생성된 순차는 DROP SEQUENCE 문을 사용하여 삭제할 수 없습니다.

순차를 삭제하면, 순차에 대한 모든 특권 또한 삭제됩니다.

테이블 공간의 상태 전환

ALTER TABLESPACE 명령문의 SWITCH ONLINE 절은 OFFLINE 상태에 있는 테이블 공간과 연관된 컨테이너가 액세스 가능하게 되면 이 테이블 공간을 ONLINE 상태로 이동시키는 데 사용할 수 있습니다. 이 테이블 공간이 ONLINE 상태로 이동되어도 데이터베이스의 나머지 부분은 여전히 시동되어 사용되고 있습니다.

이 절을 사용하는 대신 데이터베이스로부터 모든 응용프로그램을 연결해제한 다음, 응용프로그램을 데이터베이스에 다시 연결할 수도 있습니다. 그러면, 테이블 공간이 OFFLINE 상태에서 ONLINE 상태로 이동됩니다.

명령행을 사용하여 테이블 공간을 ONLINE 상태로 전환하려면, 다음과 같이 입력하십시오.

```
ALTER TABLESPACE <name>
    SWITCH ONLINE
```

DMS 테이블 공간의 컨테이너 수정

이제 DMS 테이블 공간을 병렬로 작성하고 크기를 조정할 수 있습니다. 이는 성능 면에서 이점이 있습니다. 병렬 처리 수준은 프리페처 수 더하기 1과 같습니다.

데이터베이스 액세스 제어

『색인 특권』 절 뒤에, 다음 절을 추가하십시오.

순차 특권

순차 작성자는 자동으로 USAGE 특권을 받습니다. 순차에 대해 NEXTVAL 및 PREVVAL 표현식을 사용하려면 USAGE 특권이 필요합니다. 다른 사용자들이 NEXTVAL 및 PREVVAL 표현식을 사용하도록 하려면, 순차 특권을 PUBLIC으로 부여해야 합니다. 이렇게 하면 모든 사용자가 지정된 순차가 있는 표현식을 사용하도록 허용됩니다.

『감사 기능을 사용하여 데이터 액세스 모니터』라는 절 뒤에, 다음 절을 추가하십시오.

데이터 암호화

보안 계획의 일부에는 데이터 암호화가 관련될 것입니다. 이를 수행하기 위해, 암호화 및 암호해독 내장 함수, ENCRYPT, DECRYPT_BIN, DECRYPT_CHAR 및 GETHINT를 사용할 수 있습니다. 구문을 포함하여, 이들 함수에 대한 자세한 정보는, 릴리스 정보에서 SQL 참조서 절을 참조하십시오.

ENCRYPT 함수는 암호 기반 암호화 방법을 사용하여 데이터를 암호화합니다. 또한 이들 함수는 암호 힌트를 캡슐화할 수 있도록 해줍니다. 암호 힌트는 암호화된 데이터에 삽입됩니다. 일단 암호화되면, 데이터를 암호해독하는 유일한 방법은 정확한 암호를 사용하는 것입니다. 이런 함수를 사용하도록 선택하는 개발자는 암호를 잊었을 경우와 사용 불가능한 데이터의 관리를 위한 계획을 세워야 합니다.

ENCRYPT 함수의 결과는 첫번째 인수와 동일한 데이터 유형입니다.

VARCHAR만 암호화될 수 있습니다.

결과의 선언된 길이는 다음 중 하나입니다.

- 선택적 힌트 매개변수가 지정되었을 경우 데이터 인수의 길이 더하기 42.
- 선택적 힌트 매개변수가 지정되지 않았을 경우 데이터 인수의 길이 더하기 10.

DECRYPT_BIN and DECRYPT_CHAR 함수는 암호 기반 암호해독을 사용하여 데이터를 암호해독합니다.

DECRYPT_BIN 및 DECRYPT_CHAR 함수의 결과는 첫번째 인수와 동일한 데이터 유형입니다.

결과의 선언된 길이는 원래 데이터의 길이입니다.

GETHINT 함수는 캡슐화된 암호 힌트를 리턴합니다. 암호 힌트는 데이터 소유자가 암호를 기억하도록 도와주는 구문입니다. 예를 들어, 『Ocean』이라는 단어는 암호 "Pacific"을 기억하기 위한 힌트로서 사용될 수 있습니다.

데이터 암호화에 사용되는 암호는 두 방법 중 하나로 결정됩니다.

- 암호 인수. 암호는 ENCRYPT 함수가 호출될 때 명시적으로 전달되는 문자열입니다. 데이터는 부여된 암호로 암호화 및 암호해독됩니다.
- 특수 레지스터 암호. SET ENCRYPTION PASSWORD 문은 암호 값을 암호화하여 암호화된 암호를 특수 레지스터에 저장하기 위해 데이터베이스 관리자에게 보냅니다. 암호 매개변수 없이 호출된 ENCRYPT, DECRYPT_BIN 및 DECRYPT_CHAR 함수는 ENCRYPTION PASSWORD 특수 레지스터의 값을 사용합니다.

특수 레지스터를 위한 초기 또는 기본 값은 비어 있는 문자열입니다.

암호 유효 길이는 6에서 127 사이입니다(6과 127 포함). 힌트의 유효한 길이는 0에서 32입니다(0과 32 포함).

ENCRYPTION PASSWORD 특수 레지스터를 클라이언트로부터 설정할 경우, 암호는 클라이언트에서 암호화되고, 데이터베이스 서버로 전송된 뒤, 암호해독됩니다. 암호가 읽기 가능한 상태로 남아있지 않도록 보장하기 위해, 데이터베이스 서버에서 또 재암호화됩니다. DECRYPT_BIN과 DECRYPT_CHAR 함수는 사용하기 전에 특수 레지스터를 암호해독해야 합니다. ENCRYPTION PASSWORD에서 발견된 값 또한 읽기 가능한 상태가 아닙니다. 게이트웨이 보안은 지원되지 않습니다.

데이터베이스 복구

중지된 입출력 사용 방법

『데이터베이스 복구』에 중지된 입출력 기능 사용에 관한 다음 새 절이 추가되었습니다.

주: db2inidb 유틸리티에 대한 아래 정보는 버전 7.2 새로운 기능 책의 정보를 대체합니다.

db2inidb는 DB2와 함께 제공되는 새로운 도구로서 응급 복구를 수행하고 데이터베이스를 롤 포워드 보류 상태로 만듭니다.

중단된 입출력은 온라인 분리 미리 처리를 위한 완전한 구현을 제공함으로써, 즉 데이터베이스를 종료하지 않고 미러를 분리함으로써 지속적인 시스템 사용 가능성을 지원합니다. 고객이 대형 데이터베이스에 관한 오프라인 또는 온라인 백업을 수행할 수 없을 때, 중단된 입출력 및 분리 미러를 사용하여 백업 또는 시스템 복사를 미리 이미지에서 수행할 수 있습니다.

저장 장치가 미러링된 방식에 따라, db2inidb의 사용이 달라집니다. 다음 사용은 전체 데이터베이스가 저장 시스템을 통해 계속 미러링된다고 가정합니다.

다중 노드 환경에서, 분할 이미지를 파티션에서 사용하기 전에 db2inidb 도구가 모든 파티션에서 수행되어야 합니다. db2inidb 도구는 동시에 모든 파티션에서 수행될 수 있습니다.

1. 복제 데이터베이스 작성

여기서의 목적은 기본 데이터베이스의 복제본이 읽기 전용으로 사용되도록 하는 것입니다. 다음 프로시듀어는 복제 데이터베이스 작성 방법을 설명합니다.

- a. 다음 명령을 입력하여 기본 시스템에서 입출력을 중단하십시오.

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

- b. 운영 체제 레벨 명령을 사용하여 기본 데이터베이스로부터 미러를 분리하십시오.
- c. 다음 명령을 입력하여 기본 시스템에서 입출력을 재개하십시오.

db2 set write resume for database

명령 수행 후, 기본 시스템의 데이터베이스는 정상 상태로 돌아가야 합니다.

- d. 다른 머신에서 미러된 데이터베이스에 접속하십시오.
- e. 다음 명령을 입력하여 데이터베이스 인스턴스를 시작하십시오.

db2start

- f. 다음 명령을 입력하여 DB2 응급 복구를 시작하십시오.

db2inidb *database_name* AS SNAPSHOT

주: 이 명령은 분리 시기에 1단계 확약 중 실패한 트랜잭션에 의한 변경사항을 구간 복원합니다.

이 프로세스를 오프라인 백업에도 사용할 수 있지만, 기본 시스템에서 복원될 경우, 이 백업은 로그 체인이 일치하지 않으므로 롤 포워드 사용될 수 없습니다.

2. 분리 미러를 대기 데이터베이스로 사용

미러링된(대기) 데이터베이스는 계속해서 로그를 통해 롤 포워드하므로, 기본 데이터베이스에 의해 작성된 새 로그를 기본 시스템에서 계속 가져옵니다. 다음 프로시저는 분리 미러를 대기 데이터베이스로 사용하는 방법을 설명합니다.

- a. 기본 데이터베이스에서 입출력 쓰기를 중단합니다.
- b. 기본 시스템에서 미러를 분리합니다.
- c. 기본 데이터베이스에서 입출력 쓰기를 재개하여 기본 데이터베이스가 정상 처리로 돌아가도록 합니다.
- d. 미러된 데이터베이스를 다른 인스턴스에 접속합니다.
- e. 미러링을 롤 포워드 보류 상태로 만들고 롤 포워드합니다. db2inidb 도구를 수행하여(db2inidb <db_alias> as standby) 쓰기 일시중지 상태를 제거하고 롤 포워드 보류 상태의 미러된 데이터베이스를 찾습니다.
- f. 기본 시스템에서 로그 파일을 검색하는 User Exit 프로그램을 설정하여 로그를 복사하여 이 미러된 데이터베이스가 최근 로그를 사용하도록 보장합니다.
- g. 데이터베이스를 로그 끝까지 롤 포워드합니다.
- h. f 단계로 돌아가서 기본 데이터베이스가 작업 정지될 때까지 이 프로세스를 반복합니다.

3. 분리 미러를 백업 이미지로 사용

다음 프로시저는 미러 시스템을 기본 시스템 복원을 위한 백업 이미지로 사용하는 방법을 설명합니다.

- a. 운영 체제 명령을 사용하여 미러된 데이터 및 로그를 기본 시스템 위에 복사하십시오.
- b. 다음 명령을 입력하여 데이터베이스 인스턴스를 시작하십시오.

db2start

- c. 다음 명령을 수행하여 롤 포워드 보류 상태의 미러된 데이터베이스를 찾고 쓰기 일시중지 상태를 제거합니다.

```
db2inidb database_alias AS MIRROR
```

- d. 데이터베이스를 로그 끝까지 롤 포워드합니다.

증분 백업 및 복구

『데이터베이스 복구』에서, 다음은 증분 백업 및 복구에 관한 새로운 절입니다.

데이터베이스의 크기 그리고 특히 웨어하우스는 테라바이트와 페타바이트 범위로 계속 확장되므로, 이러한 데이터베이스 백업과 복구에 필요한 시간 및 하드웨어 자원 또한 실질적으로 증가합니다. 데이터베이스가 클 경우, 전체 데이터베이스와 테이블 공간 백업은 항상 최선의 방법은 아닙니다. 그런 데이터베이스의 여러 사본에 필요한 저장영역의 요구사항이 막대하기 때문입니다. 다음 문제점들을 고려하십시오.

- 웨어하우스의 데이터의 소수만 변경될 경우, 전체 데이터베이스를 백업할 필요가 없습니다.
- 기본 데이터베이스에 테이블 공간을 추가한 뒤 테이블 공간만 백업하는 것은 위험합니다. 백업된 테이블 공간 외부의 데이터가 변경될 수 있기 때문입니다.

이제 DB2는 증분 백업 및 복구를 지원합니다(긴 필드 또는 큰 오브젝트 데이터는 아닙니다). 증분 백업이란 이전 백업이 수행된 이후에 갱신된 페이지만 들어 있는 백업 이미지입니다. 갱신된 데이터 및 색인 페이지에 덧붙여, 각각의 증분 백업 이미지에는 일반적으로 전체 백업 이미지에 저장되는 모든 초기 데이터베이스 메타데이터(데이터베이스 구성, 테이블 공간 정의, 데이터베이스 실행기록 등과 같은)가 들어 있습니다.

두 가지 유형의 증분 백업이 지원됩니다.

- **증분.** 증분 백업 이미지는 가장 최근의 성공적인 전체 백업 조작 이후에 변경된 데이터베이스 데이터 모두의 사본입니다. 오랜 시간 동안 수행되는 일련의 증분 백업에는 각각 이전 증분 백업 이미지의 내용이 들어 있기 때문에 누적 백업 이미지라고도 합니다. 증분 백업 이미지의 선임 백업은 항상 동일한 오브젝트의 가장 최근의 성공적 전체 백업입니다.
- **델타.** 델타 또는 증분 델타, 백업 이미지는 문제의 테이블 공간의 마지막 성공적인 백업(전체, 증분 또는 델타) 이후에 변경된 모든 데이터베이스 데이터의 사본입니다. 이를 차별 또는 비누적 백업 이미지라고도 합니다. 델타 백업 이미지의 선임 백업은 델타 백업 이미지의 테이블 공간의 각각의 사본이 들어 있는 가장 최근의 성공적인 백업입니다.

증분과 델타 백업 이미지 사이의 핵심 차이점은 계속해서 변경되는 오브젝트를 연달아 백업할 때 취하는 조치입니다. 각각의 연속되는 증분 이미지에는 이전 증분 이미지의 전체 내용과, 이전 백업이 생성된 이후의 변경되거나 새로운 데이터가 들어 있습니다. 델타 백업 이미지에는 이전 이미지가 생성된 이후에 변경된 페이지만 들어 있습니다.

데이터베이스와 테이블 공간 증분 백업의 조합은, 온라인 및 오프라인 조작 모드에서 모두 허용됩니다. 데이터베이스와 테이블 공간 증분 백업의 결합은 데이터베이스 백업(여러 테이블 공간의 테이블 공간 백업)이 반드시 단일 이미지일 필요는 없지만, 다른 시간에 수행된 이전의 데이터베이스와 테이블 공간 백업의 고유한 세트일 수 있음을 암시하므로, 백업 전략을 계획할 때 주의하십시오.

데이터베이스 또는 테이블 공간을 일치된 상태로 재구축하려면, 복구 프로세스가 복원될 전체 오브젝트(데이터베이스 또는 테이블 공간)의 일치된 이미지로 시작해야 하며, 적합한 증분 백업 이미지 각각을 아래 설명된 순서로 적용해야 합니다("복원 방법" 절을 참조하십시오).

데이터베이스 갱신을 추적하기 위해, DB2는 새로운 데이터베이스 구성 매개 변수 TRACKMOD를 지원하며, 매개변수는 두 가능한 값 중 하나를 갖습니다.

- NO. 증분 백업이 이 구성에서는 허용되지 않습니다. 데이터베이스 페이지 갱신이 어떤 방식으로든 추적 또는 기록되지 않습니다.
- YES. 이 구성에 증분 백업이 허용됩니다. 갱신 추적을 사용하면, 인스턴스의 데이터베이스에 최초로 성공적인 연결이 있으면, 변경이 유효해집니다. 증분 백업하기 전에 전체 데이터베이스 백업이 필요합니다.

기존 데이터베이스에 대한 기본 TRACKMOD 설정은 NO이고, 새 데이터베이스에 대해서는 YES입니다.

추적의 세분성은 SMS 및 DMS 테이블 공간 모두에 대한 테이블 공간 레벨에 있습니다.

최소값이더라도, 데이터베이스의 갱신 추적은 데이터를 갱신 또는 삽입하는 트랜잭션의 런타임 성능에 영향을 줄 수 있습니다.

증분 백업 이미지로부터 복원

증분 백업 이미지로부터의 복원 조작은 항상 다음 단계들로 구성됩니다.

1. 증분 목표 이미지 식별. DBA는 먼저 복원된 최종 이미지를 결정하고, DB2 복원 유틸리티로부터 증분 복원 조작을 요청해야 합니다. 이 이미지가 복원될 최종 이미지가 되므로, 이를 증분 복원의 목표 이미지라고 합니다. 이 이미지에 대한 증분 복원 명령은 이 목표 이미지의 구성 및 테이블 공간 정의가 있는 새로운 데이터베이스의 작성을 시작할 수 있습니다. 증분 목표 이미지는 RESTORE DATABASE 명령에서 TAKEN AT 매개변수를 사용하여 지정됩니다.
2. 후속 증분 백업 이미지가 적용될 수 있는 베이스라인을 구축하기 위해 가장 최근의 전체 데이터베이스 또는 테이블 공간 이미지 복원.
3. 필요한 전체 또는 테이블 공간 증분 백업 이미지 각각을 생성된 순서로, 2 단계에서 복원된 베이스라인 이미지의 맨 위에 복원.

4. 1 단계의 목표 이미지를 두 번째 읽을 때까지 3 단계 반복. 목표 이미지는 완전한 증분 복원 조작 중에 두 번 액세스됩니다. 첫번째 액세스 중에, 오직 초기 데이터만 이미지로부터 읽을 수 있습니다. 사용자 데이터는 읽히지 않습니다. 두 번째 액세스 중에만 완전한 이미지가 읽히고 처리됩니다.

복원 조작 중에 작성될 데이터베이스에 대해 데이터베이스가 초기에 올바른 실행 기록, 데이터베이스 구성 및 테이블 공간 정의로 구성되었는지 확인하기 위해 증분 복원 조작의 목표 이미지를 두 번 액세스해야 합니다. 최초 전체 데이터베이스 백업 이미지가 수행된 이후에 테이블 공간이 삭제되었을 경우, 해당 이미지에 대한 테이블 공간 데이터는 백업 이미지로부터 읽히지만 증분 복원 처리 중에 무시됩니다.

예를 들어, 다음과 같습니다.

1. `db2 restore database sample incremental taken at <ts>`

여기서,
<ts>는 복원될 최종 증분 백업 이미지를 가리킵니다.

2. `db2 restore database sample incremental taken at <ts1>`

여기서,
<ts1>은 초기 전체 데이터베이스(또는 테이블 공간) 이미지를 가리킵니다.

3. `db2 restore database sample incremental taken at <tsX>`

여기서,
<tsX>는 작성 순차의 각 증분 백업 이미지를 가리킵니다.

4. 이미지 <ts>까지를 포함하여 각 증분 백업 이미지를 복원하여 3단계를 반복하십시오.

데이터베이스 복원 조작을 시도하는 중에, 테이블 공간 증분 백업이 생성되었을 경우, 테이블 공간 이미지는 백업 시간소인의 연대 순으로 복원되어야 합니다.

병렬 복구

이제 DB2는 여러 에이전트를 사용하여 응급 복구와 데이터베이스 롤 포워드 복구를 모두 수행합니다. 특히 대칭 다중 프로세서(SMP) 머신에서 이러한 조작 중의 성능 개선을 기대할 수 있습니다. 데이터베이스 복구 중에 여러 에이전트를 사용하면 SMP 머신에서 사용 가능한 여분의 CPU를 이용합니다.

이 개선점에 도입된 새로운 에이전트 유형이 db2agnsc입니다. DB2는 머신의 CPU 수에 기반하여 데이터베이스 복구에 사용될 에이전트의 수를 선택합니다. SMP 머신의 경우, 사용되는 에이전트 수는 (CPU 수 + 1)입니다. 단일 CPU 머신에서는, 보다 효율적인 로그 읽기, 로그 레코드 처리 및 데이터 페이지 프리페치를 위해 사용됩니다.

DB2는 이러한 로그 레코드들이 적합할 경우 동시에 다시 적용될 수 있도록 분산합니다. 로그 레코드 처리가 페이지 레벨에서 병렬로 이루어지므로(같은 데이터 페이지의 로그 레코드는 같은 에이전트에 의해 처리됨) 모든 작업이 한 테이블에서 수행되더라도 성능이 향상됩니다.

Named Pipe에 백업

UNIX 기반 시스템에서 지역 named pipe에 데이터베이스 백업(그리고 데이터베이스 복원)에 대한 지원이 이제 사용 가능합니다. Named Pipe의 기록기와 관독기가 모두 동일한 머신에 있어야 합니다. 파이프는 반드시 존재해야 하며 지역 파일 시스템에 위치해야 합니다. Named Pipe는 지역 장치로서 취급되므로, 목표가 Named Pipe임을 지정할 필요가 없습니다. 다음은 AIX 예입니다.

1. Named Pipe를 작성합니다.
`mkfifo /u/dbuser/mypipe`
2. 이 파이프를 데이터베이스 백업 조작의 목표로 사용합니다.
`db2 backup db sample to /u/dbuser/mypipe`
3. 데이터베이스를 복원합니다.
`db2 restore db sample into mynewdb from /u/dbuser/mypipe`

분리 이미지로부터 백업

이제 DB2는 데이터베이스의 분리 미러된 사본에서 전체 오프라인 데이터베이스 백업을 지원합니다. 롤 포워드 보류 상태에 있는 데이터베이스는 사용 불가능하므로, 온라인 백업은 지원되지 않고 필요하지 않습니다. 분리 미러된 백업 이미지가 복원될 때, 분리가 발생했을 때 활동 중인 트랜잭션이 있었을 수 있으므로 롤 포워드되어야 합니다.

주: DB2 버전 7.1 FixPak 3 및 DB2 버전 7.2의 경우, 이 지원은 DMS 테이블 공간만 포함된 데이터베이스에만 국한됩니다. 분리 후에 데이터베이스 백업을 시도하고 데이터베이스에 SMS 테이블 공간이 있을 경우, 백업은 실패합니다.

한번 데이터베이스가 분리되면, **db2inidb** 유틸리티를 사용하여 다음 옵션 중 하나를 지정해야 합니다.

- **Snapshot.** 이는 응급 복구를 실시하여, 데이터베이스를 일치되게 합니다. 새로운 로그 체인이 시작되며, 데이터베이스는 본래 데이터베이스의 로그를 통해 롤 포워드할 수 없습니다. 데이터베이스는 백업을 포함한 어떤 조작에도 사용 가능합니다.
- **Standby.** 이것은 데이터베이스를 롤 포워드 보류 상태로 만듭니다. 응급 복구는 수행되지 않으며, 데이터베이스는 불일치 상태로 남아 있습니다.
- **Mirror.** 이것은 데이터베이스의 미러된 사본이 원래 데이터베이스를 바꾸도록 만듭니다. 데이터베이스는 롤 포워드 보류 상태가 되고, **WRITE SUSPEND** 상태는 해제됩니다. 응급 복구는 수행되지 않으며, 데이터베이스는 불일치 상태로 남아 있습니다.

다음은 몇가지 사용 시나리오입니다.

- 데이터베이스 복제 작성.

이것의 목적은 예를 들어 보고서 작성에 사용될 수 있는 기본 데이터베이스의 읽기 전용 복제를 만드는 것입니다. 그러려면, 다음 단계를 수행하십시오.

1. 기본 시스템에서 입출력을 일시중단하십시오.

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

2. 미러를 분리하십시오. 운영 체제 레벨 명령을 사용하여 기본 데이터베이스로부터 미러를 분리하십시오.

3. 기본 시스템에서 입출력을 재개하십시오.

```
db2 set write resume for database
```

기본 시스템의 데이터베이스가 이제 정상 상태로 돌아가야 합니다.

4. 데이터베이스의 분리 미러를 다른 호스트에 마운트하십시오.

5. 인스턴스를 시작하십시오.

```
db2start
```

6. DB2 응급 복구를 시작하십시오.

```
db2inidb <db_name> as snapshot
```

이 프로세스를 오프라인 백업에도 사용할 수 있지만, 기본 시스템에서 복원될 경우, 이 백업은 로그 체인이 일치하지 않으므로 롤 포워드 사용될 수 없습니다.

- 분리 미러를 대기 데이터베이스로 사용.

여기서의 아이디어는 미러된(대기) 데이터베이스가 로그를 통해 계속해서 롤 포워드 하고, 기본 데이터베이스가 작성하는 새로운 로그조차 기본 시스템으로부터 계속 가져온다는 것입니다. 분리 미러를 대기 데이터베이스로 사용하려면, 다음 단계를 수행하십시오.

1. 기본 시스템에서 입출력을 일시중단하십시오.

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

2. 미러를 분리하십시오. 운영 체제 레벨 명령을 사용하여 기본 데이터베이스로부터 미러를 분리하십시오.

3. 기본 시스템에서 입출력을 재개하십시오.

```
db2 set write resume for database
```

기본 시스템의 데이터베이스가 이제 정상 상태로 돌아가야 합니다.

4. 데이터베이스의 분리 미러를 다른 호스트에 마운트하십시오.

5. 쓰기 일시중지 상태를 제거하고 미러된 데이터베이스를 롤 포워드 보류 상태에 두십시오.

```
db2inidb <db_name> as standby
```


6. 로그를 복사하십시오. 이 미러된 데이터베이스에 대해 최근 로그를 사용할 수 있도록 기본 시스템의 아카이브 위치에서 로그 파일을 검색하는 User Exit 프로그램을 설정하십시오.

7. 미러를 로그 끝까지 롤 포워드합니다.

```
db2 rollforward db <db_name> to end of logs
```

8. 기본 데이터베이스가 작동이 중지될 때까지 단계 6부터 프로세스를 반복하십시오.

- 분리 미러를 사용하여 기본 시스템 복구.

다음 프로시더는 미러된 시스템을 백업 이미지로 사용하여 기본 시스템을 복원하는 방법을 설명합니다.

1. 복사하십시오. 운영 체제 명령을 사용하여 미러된 데이터 및 로그를 기본 시스템 위에 복사하십시오.

2. 인스턴스를 시작하십시오.

```
db2start
```

3. 복원된 미러를 롤 포워드 보류 상태에 두고 미러를 로그 끝까지 롤 포워드하십시오.

```
db2inidb <dbname> as mirror
```

- 응급 복구를 수행하지 않고 백업.

응급 복구를 수행하지 않고 분리 미러에서 오프라인 백업을 수행한다는 것은 이 백업 이미지를 기본 시스템의 맨위에 복원할 수 있다는 의미입니다. 그러려면, 다음 단계를 수행하십시오.

1. 기본 시스템에서 입출력을 일시중단하십시오.

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

2. 미러를 분리하십시오. 운영 체제 레벨 명령을 사용하여 기본 데이터베이스로부터 미러를 분리하십시오.

3. 기본 시스템에서 입출력을 재개하십시오.

```
db2 set write resume for database
```

기본 시스템의 데이터베이스가 이제 정상 상태로 돌아가야 합니다.

4. 데이터베이스의 분리 미러를 다른 호스트에 마운트하십시오.

5. 인스턴스를 시작하십시오.

```
db2start
```

6. 미러된 데이터베이스를 롤 포워드 보류 상태에 두십시오.

```
db2inidb <db_name> as standby
```

7. 데이터베이스 백업 조작을 호출하십시오.

```
db2 backup database <db_name>
```

이 결과 내재된 데이터베이스 연결이 설정되지만, DB2 응급 복구를 시작하지는 않습니다.

요구에 따른 로그 아카이브

DB2는 이제 어느 때든 복구 가능한 데이터베이스에 대한 활동 중인 로그의 닫기(그리고, User Exit 옵션을 사용할 경우, 아카이브)를 지원합니다. 이는 알려진 지점까지 로그 파일의 완전한 세트를 수집한 뒤 이 로그 파일을 사용하여 대기 데이터베이스를 갱신할 수 있도록 해줍니다.

주: 요구에 따른 아카이브는 로그 파일이 즉시 아카이브됨을 보장하지 않습니다. 로그 파일을 절단하고 아카이브 요청을 발행하지만, User Exit 프로그램과 연관되어 여전히 지연될 수 있습니다.

새로운 DB2 ARCHIVE 명령을 호출하여 또는 새로운 **db2ArchiveLog** API를 호출하여 요구에 따른 아카이브를 실시할 수 있습니다.

로그 미러링

『데이터베이스 복구』에서, 일시중단된 입출력 기능 사용에 관한 다음 새 절이 추가됩니다.

DB2는 이제 데이터베이스 레벨에서 로그 미러링을 지원합니다. 로그 파일 미러링은 다음으로부터 데이터베이스를 보호하는 데 도움이 됩니다.

- 사용 중인 로그를 실수로 삭제
- 하드웨어 장애로 인한 데이터 손상

사용 중인 로그의 손상(디스크 손상의 결과로)이 염려된다면, 로그가 저장된 볼륨을 미러링하여 사용 중인 로그의 사본을 관리하기 위해 새로운 DB2 레지스트리 변수, DB2_NEWLOGPATH2를 사용하여 데이터베이스에 2차 경로를 지정하는 것을 고려해 보십시오.

DB2_NEWLOGPATH2 레지스트리 변수는 데이터베이스가 로그 파일의 동일한 두 번째 사본을 다른 경로에 기록하도록 해줍니다. 물리적으로 별도의 디스크에(또한 다른 디스크 제어기에 있으면 더욱 좋습니다) 2차 로그 경로를 배치하도록 권합니다. 그렇게 하면, 디스크 제어기가 단일 실패 지점이 될 수 없습니다.

주: Windows NT와 OS/2는 중재 경로 이름 아래에 장치를 "마운팅"하는 것을 허용하지 않으므로, 별도 장치에 2차 경로를 지정할 수 없습니다(이들 플랫폼에서는).

DB2_NEWLOGPATH2는 사용 가능(1로 설정) 또는 사용 불가능(0으로 설정)이 될 수 있습니다. 기본 값은 영(0)입니다. 이 변수를 1로 설정할 경우, 2차 경로 이름은 문자 2가 연결된 LOGPATH 변수의 현재 값입니다. 예를 들어, SMP 환경에서 LOGPATH가 /u/dbuser/sqllogdir/logpath이면, 2차 로그 경로는 /u/dbuser/sqllogdir/

logpath2가 됩니다. MPP 환경에서, LOGPATH가 /u/dbuser/sqllogdir/logpath 이면, DB2는 노드 표시기를 경로에 추가하고 /u/dbuser/sqllogdir/logpath/NODE0000을 기본 로그 경로로 사용합니다. 이 경우, 2차 로그 경로는 /u/dbuser/sqllogdir/logpath2/NODE0000입니다.

처음 DB2_NEWLOGPATH2가 사용 가능화되면, 다음 데이터베이스 시작시에 현재 로그 파일이 완료되기까지는 실제로 사용되지 않습니다. 이는 NEWLOGPATH가 현재 사용되는 방식과 유사합니다.

기본 또는 2차 로그 경로에 쓰기 중에 오류가 있으면, 데이터베이스는 실패 경로를 "bad"로 표시하고, 메시지를 db2diag.log 파일에 쓰고, 이어지는 로그 레코드를 남아 있는 "good" 로그 경로에만 씁니다. DB2는 현재 로그 파일이 완료될 때까지 "bad" 경로를 다시 사용하지 않습니다. DB2가 다음 로그 파일을 열어야 할 때에는, 이 경로가 유효한지 검증하고, 유효하다면 사용하기 시작합니다. 유효하지 않을 경우, DB2는 다음 로그 파일이 처음 액세스되기 전에는 경로를 다시 사용하지 않습니다. 로그 경로를 동기화하려는 시도가 없지만, DB2는 로그 파일이 아카이브될 때 올바른 경로가 사용되도록 발생하는 액세스 오류에 관한 정보를 보존합니다. 남아 있는 "good" 경로에 쓰는 동안 실패가 발생하면, 데이터베이스는 이상종료됩니다.

Sun Solaris와 HP에 대한 교차 플랫폼 백업 및 복원 지원

이제 Sun Solaris와 HP 사이에 교차 플랫폼 백업 및 복원 지원이 사용 가능합니다. 시스템 간에 백업 이미지를 전송할 때, 2진 모드로 전송해야 합니다. 목표 시스템에서, 데이터베이스는 원래 데이터베이스가 작성된 시스템과 동일한 코드 페이지/영역으로 작성되어야 합니다.

DB2 Data Links Manager 고려사항/백업 유틸리티 고려사항

이 절의 두 번째 단락을 다음으로 바꾸십시오.

파일들이 링크될 때, Data Links 서버는 파일이 ADSM과 같은 아카이브 서버 또는 디스크에 비동기로 복사되도록 스케줄합니다. 백업 유틸리티가 수행될 때, DB2는 복사를 위해 스케줄된 모든 파일이 복사되었는지 확인합니다. 백업 처리를 시작할 때, DB2는 DB2 구성 파일에 지정된 모든 Data Links 서버에 접속합니다. Data Links 서버에 하나 이상의 파일이 링크되어 있고, 수행 중이지 않거나, 백업 조작 중에 수행이 중지되면, 백업에는 완전한 DATALINK 정보가 들어 있지 않습니다. 백업 조작은 성공적으로 완료됩니다. Data Links 서버가 다시 데이터베이스에 사용 가능으로 표시되려면, 모든 미결된 백업에 대한 백업 처리가 성공적으로 완료되어야 합니다. Data Links 서버에서 완료되기를 기다리는 미결 백업의 수가 이미 num_db_backups (아래 참조) 값의 두 배일 때 백업이 시작되면, 백업 조작은 실패합니다. 추가 백업이 허용되려면 Data Links 서버를 재시작하고 미결 백업이 완료되어야 합니다.

DB2 Data Links Manager 고려사항/복원 및 롤 포워드 유틸리티 고려사항

다음으로 시작하는 단락,

데이터베이스 또는 테이블 공간을 복원하고
WITHOUT DATALINK를 지정하지 않을 경우...

그리고

데이터베이스 또는 테이블 공간을 복원하고
WITHOUT DATALINK 옵션을 지정하지 않을 경우...

이 단락을 다음으로 바꾸십시오.

데이터베이스 또는 테이블 공간을 복원할 때, 복원 조작이
성공하려면 다음 조건이 만족되어야 합니다.

- o 백업 파일에 기록된 Data Links 서버가 수행 중이지 않을 경우,
복원 조작은 그래도 성공적으로 완료됩니다.

누락된 Data Links 서버의 영향을 받는 DATALINK 컬럼 정보가 있는
테이블은 복원 조작(또는 롤 포워드 조작, 사용되었을 경우) 후에
데이터 링크 조정 보류 상태가 됩니다.

Data Links 서버가 데이터베이스에 다시 사용 가능으로 표시되려면,
이 복원 처리가 성공적으로 완료되어야 합니다.

- o 백업 파일에 기록된 Data Links Server가 복원 조작 중에 수행이
중지되면, 복원 조작은 실패합니다. 복원은 Data Links Server가
작동 중지되어도 재시작될 수 있습니다(위를 참조).

- o 이전 데이터베이스 복원 작업이 Data Links 서버에서 아직
완료되지 않았을 경우, 이어지는 데이터베이스 또는 테이블 공간
복원 조작은 그러한 Data Links 서버가 재시작되고, 완료되지
않은 복원이 완료될 때까지 실패합니다.

- o 백업 파일에 기록된 모든 DATALINK 컬럼에 대한 정보는
적절한 Data Links 서버의 등록 테이블에 존재해야 합니다.

IDATALINK에 관한 모든 정보가 등록 테이블에 기록되지 않으면,
누락된 DATALINK 컬럼 정보가 있는 테이블은 복원 조작(또는
롤 포워드 조작, 사용되었을 경우) 후에 데이터 링크 조정 불가능
상태가 됩니다.

백업이 등록 테이블에 기록되지 않는다면, 제공된 백업 파일이
num_db_backup의 값보다 이전 값이고 이미 "가비지 수집되었음"을
의미할 수 있습니다. 이것은 이전 백업의 아카이브된 파일이
제거되었고 복원될 수 없음을 의미합니다. DATALINK 컬럼이
있는 모든 테이블은 데이터 링크 조정 보류 상태가
됩니다.

백업 파일이 등록 테이블에 기록되지 않는다면, Data Links
서버가 수행 중이지 않기 때문에 백업 처리가 아직 완료되지
않았음을 의미할 수 있습니다. DATALINK 컬럼이 있는 모든
테이블은 데이터 링크 조정 보류 상태가 됩니다. Data Links
테이블은 데이터 링크 조정 보류 상태가 됩니다. Data Links
서버가 재시작될 때, 백업 처리는 복원 처리 전에 완료됩니다.

테이블은 사용자가 계속 사용 가능하지만, DATALINK 컬럼의 값은
정확하게 파일을 참조하지 않을 수도 있습니다(예를 들어, DATALINK
컬럼의 값과 일치하는 파일을 찾지 못할 수도 있습니다).

이런 경우를 원치 않으면,
"SET CONSTRAINTS for tablename TO DATALINK RECONCILE PENDING" 문을
실행하여 테이블을 점검 보류 상태로 만들 수 있습니다.

복원 조작 후에, 데이터 링크 조정 불가능 상태에 있는 테이블이 있을 경우,
"Datalink_Reconcile_Not_Possible 상태에서 테이블 제거" 아래에 제안된 방법중 하
나로 DATALINK 컬럼 데이터를 수정할 수 있습니다.

첫번째 단락 하단의 참고는 동일합니다.

다음은 이 절의 끝에 추가하십시오.

데이터베이스 백업 이미지의 datalink.cfg 파일은
백업 시점의 datalink.cfg 파일만 반영하므로, 비정상적인 복구 사례를
다루기 위해 datalink.cfg 파일을 아카이브해둘 것을 강력히
권합니다. 모든 복구 사례에 가장 최근 datalink.cfg 파일이
필요합니다. 따라서, 매번 ADD DATALINKS MANAGER 또는
DROP DATALINKS MANAGER 명령 호출 후에 datalink.cfg 파일을
백업해야 합니다.

최근 datalink.cfg 파일이 디스크에 없을 경우,
가장 최근 datalink.cfg 파일을 검색하는 데 도움이 됩니다.

가장 최근 datalink.cfg 파일이 디스크에 없을 경우, 기존
datalink.cfg 파일(백업 이미지로부터 복원된)을 롤 포워드 조작을
수행하기 전에 아카이브된 최근 datalink.cfg로 바꾸십시오.
데이터베이스가 복원된 후에 이를 수행하십시오.

롤 포워드하지 않고 오프라인 백업으로부터 데이터베이스 복원

테이블 공간 레벨이 아닌, 데이터베이스 레벨에서만 롤 포워드없이 복원할 수 있습니다.
롤 포워드하지 않고 데이터베이스를 복원하려면, 복구 불가능 데이터베이스(즉, 순환 로
그를 사용하는 데이터베이스)를 복원하거나, RESTORE DATABASE 명령에
WITHOUT ROLLING FORWARD 매개변수를 지정하십시오.

WITHOUT DATALINK 옵션으로 복원 유틸리티를 사용할 경우, DATALINK 컬럼
이 있는 모든 테이블은 데이터 링크 조정 보류(DRP) 상태가 되고, 복원 조작 중에 Data
Links 서버로 조정이 수행되지 않습니다.

WITHOUT DATALINK 옵션을 사용하지 않고, 백업 파일에 기록된 Data Links 서
버가 더 이상 데이터베이스에 정의되어 있지 않을 경우(즉, DROP DATALINKS
MANAGER 명령을 사용하여 삭제된 경우), 삭제된 Data Links 서버를 참조하는
DATALINK 데이터가 들어 있는 테이블은 복원 유틸리티에 의해 DRP 상태가 됩니
다.

WITHOUT DATALINK 옵션을 사용하지 않을 경우, 모든 Data Links 서버가 사용
가능하며, DATALINK 컬럼에 대한 모든 정보가 등록 테이블에 전체가 기록되고, 백
업 파일에 기록된 각 Data Links 서버에 대해 다음이 발생합니다.

- 데이터베이스 복원 조작에 사용된 백업 이미지 후에 링크된 모든 파일은 링크해제로
표시됩니다(백업 이미지에 링크된 것으로 기록되지 않기 때문입니다).

- 백업 이미지 후에 링크해제되었지만, 백업 이미지를 취하기 전에 링크되었던 모든 파일은 링크된 것으로 표시됩니다(백업 이미지에 링크된 것으로 기록되었기 때문입니다). 파일이 연속해서 다른 데이터베이스의 다른 테이블에 링크되었을 경우, 복원 테이블은 데이터 링크 조정 보류 상태가 됩니다.

주: 데이터베이스 복원 조작에 사용된 백업 이미지가 최소한 하나의 Data Links 서버가 수행 중이 아닐 때 취해졌다면, 백업의 DATALINK 정보가 불완전하므로 위의 내용을 수행할 수 없습니다. 또한 데이터베이스 복원 조작에 사용된 백업 이미지를 포워드 가 있거나 없는 데이터베이스 복원 후에 취해졌을 경우에 위의 내용이 수행되지 않습니다. 두 경우 모두, DATALINK 컬럼이 있는 모든 테이블은 데이터 링크 조정 보류 상태가 되고, 복원 조작 중에 Data Links 서버에서 아무 조정도 수행되지 않습니다.

데이터베이스와 테이블 공간 복원 및 로그 끝까지 롤 포워드

데이터베이스나 테이블 공간을 복원한 뒤, 로그 끝까지(모든 로그가 제공됨을 의미) 롤 포워드할 경우, 백업 파일에 기록된 Data Links 서버 중 최소한 하나가 복원 조작 중에 수행되지 않는 한 조정 점검이 필요하지 않습니다. 롤 포워드 조작에 대해 모든 로그가 제공되는지 확인할 수 없거나, DATALINK 값을 조정할 필요가 있다고 생각되면, 다음을 수행하십시오.

1. 관련된 테이블(또는 테이블들)에 대해 SQL문을 발행하십시오.

```
SET CONSTRAINTS FOR tablename TO DATALINK RECONCILE PENDING
```

이 명령문은 테이블을 데이터 링크 조정 보류 및 점검 보류 상태로 만듭니다.

2. 테이블이 점검 보류 상태가 되길 원치 않을 경우, 다음 SQL문을 발행하십시오.

```
SET CONSTRAINTS FOR tablename IMMEDIATE CHECKED
```

그러면 테이블이 점검 보류 상태에서는 벗어나지만, 데이터 링크 조정 보류 상태에 남아 있습니다. 테이블이 이 상태에서 벗어나게 하려면, 조정 유틸리티를 사용해야 합니다.

데이터베이스에서 삭제된 DB2 Data Links Manager를 참조하는 DATALINK 데이터가 백업 파일에 포함될 수 있습니다(즉, 백업이 수행될 때 DB2 Data Links Manager가 데이터베이스에 등록되어 있었습니다). 롤 포워드 중인 각 테이블 공간에 삭제된 DB2 DataLinks Manager를 참조하는 DATALINK 데이터가 있는 테이블이 최소한 하나 포함되어있을 경우, 모든 테이블은 롤 포워드 유틸리티에 의해 DRP 상태가 됩니다.

DB2 Data Links Manager 및 복구 상호작용

다음 테이블은 수행할 수 있는 복구의 서로 다른 유형, 복원 및 롤 포워드 처리중에 발생하는 DB2 Data Links Manager 처리 그리고 복구가 완료된 후 조정 유틸리티를 수행해야 하는지 여부를 보여줍니다.

복구 유형	복원 중 DB2 Data Links Manager 처리	롤 포워드 중 DB2 Data Links Manager 처리	조정
복구 불가능한 데이터베이스 (logretain=NO)			
완전한 백업의 데이터베이스 복원, 모든 Data Links Server 작동	빠른 조정이 수행됩니다.	N/A	파일 링크에서의 문제가 의심될 경우 선택적으로 수행할 수 있습니다.
WITHOUT DATALINK 옵션을 사용하여 데이터베이스 복원	테이블이 <i>Datalink_Reconcile_Pending</i> 상태가 됨	N/A	필수
완전한 백업의 데이터베이스 복원, 최소한 하나의 Data Links Server 작동 중지	작동이 중지된 Data Links 서버에 링크가 없는 테이블 공간의 테이블들에만 빠른 조정이 수행되고, 기타 테이블은 <i>Datalink_Reconcile_Pending</i> 상태가 됩니다.	NA	작동이 중지된 Data Links 서버에 링크가 있는 테이블 공간의 테이블에 필요
불완전한 백업의 데이터베이스 복원, 모든 Data Links 서버 작동	빠른 조정은 수행되지 않고, DATALINK 컬럼이 있는 모든 테이블은 <i>Datalink_Reconcile_Pending</i> 상태가 됩니다.	NA	필수
복구 가능한 데이터베이스 (logretain=YES)			
WITHOUT ROLLING FORWARD 옵션을 사용하는 데이터베이스 복원, 완전한 백업 사용, 모든 Data Links 서버 작동	빠른 조정이 수행됩니다.	N/A	선택적
WITHOUT ROLLING FORWARD 및 WITHOUT DATALINK 옵션을 사용하는 데이터베이스 복원, 완전 또는 불완전 백업 사용, Data Links 서버가 작동 또는 작동 중지	테이블이 <i>Datalink_Reconcile_Pending</i> 상태가 됨	N/A	필수
WITHOUT ROLLING FORWARD 옵션을 사용하여 데이터베이스 복원, 완전한 백업 사용, 최소한 하나의 Data Links 서버가 작동 중지	작동이 중지된 Data Links 서버에 링크가 없는 테이블 공간의 테이블에만 빠른 조정이 수행되고, 기타 테이블은 <i>Datalink_Reconcile_Pending</i> 상태가 됩니다.	N/A	작동이 중지된 Data Links 서버에 링크가 있는 테이블 공간의 테이블에 필요
WITHOUT ROLLING FORWARD 옵션을 사용하는 데이터베이스 복원, 불완전한 백업 사용, Data Links 서버 작동 또는 작동 중지	빠른 조정은 수행되지 않고, DATALINK 컬럼이 있는 모든 테이블은 <i>Datalink_Reconcile_Pending</i> 상태가 됩니다.	N/A	필수
데이터베이스 복원 및 로그 끝까지 롤 포워드, 완전한 백업 사용, 모든 Data Links 서버 작동	조치 없음	조치 없음	선택적

복구 유형	복원 중 DB2 Data Links Manager 처리	롤 포워드 중 DB2 Data Links Manager 처리	조정
데이터베이스 복원 및 로그 끝까지 롤 포워드, 완전한 백업 사용, 최소한 하나의 Data Links 서버가 롤 포워드 처리 중에 작동 중지	조치 없음	조치 없음	선택적
데이터베이스 복원 및 로그 끝까지 롤 포워드, 완전한 또는 불완전한 백업 사용, 복원 중 Data Links 서버가 작동 중지	조치 없음	DATALINK 컬럼이 있는 모든 테이블은 Datalink_Reconcile_Pending 상태가 됩니다.	DATALINK 컬럼이 있는 모든 테이블에 필요
데이터베이스 복원 및 로그 끝까지 롤 포워드, 불완전한 백업 사용, 복원 중 모든 Data Links 서버가 작동	조치 없음	조치 없음	선택적
데이터베이스 복원 및 로그 끝까지 롤 포워드, 완전한 또는 불완전한 백업 사용, 모든 Data Links 서버가 작동, 어떤 Data Links 서버에서 백업을 알 수 없음	조치 없음	백업을 알 수 없는 Data Links 서버에 링크가 있는 테이블 공간의 모든 테이블이 Datalink_Reconcile_Pending 상태가 됩니다.	필수
테이블 공간 복원 및 로그 끝까지 롤 포워드, 완전한 백업 사용, 모든 Data Links 서버 작동	조치 없음	조치 없음	선택적
테이블 공간 복원 및 로그 끝까지 롤 포워드, 완전한 백업 사용, 최소한 하나의 Data Links 서버가 롤 포워드 처리 중에 작동 중지	조치 없음	조치 없음	선택적
테이블 공간 복원 및 로그 끝까지 롤 포워드, 완전한 또는 불완전한 백업 사용, 복원 처리 중 Data Links 서버가 작동 중지	조치 없음	작동 중지된 Data Links 서버에 링크가 있는 테이블 공간의 모든 테이블은 Datalink_Reconcile_Pending 상태가 됩니다.	작동이 중지된 Data Links 서버에 링크가 있는 테이블 공간의 테이블에 필요
테이블 공간 복원 및 로그 끝까지 롤 포워드, 불완전한 백업 사용, 모든 Data Links 서버 작동	조치 없음	조치 없음	선택적
테이블 공간 복원 및 로그 끝까지 롤 포워드, 완전한 또는 불완전한 백업 사용, 복원 및/또는 롤 포워드 처리 중 Data Links 서버가 작동 중지	조치 없음	테이블이 Datalink_Reconcile_Pending 상태가 됨	필수

복구 유형	복원 중 DB2 Data Links Manager 처리	롤 포워드 중 DB2 Data Links Manager 처리	조정
테이블 공간 복원 및 어느 시점까지 롤 포워드, 완전한 또는 불완전한 백업 사용, 복원 및/또는 롤 포워드 처리 중 Data Links 서버가 작동 또는 작동 중지	조치 없음	테이블이 <i>Datalink_Reconcile_Pending</i> 상태가 됨	필수
1의 롤 포워드 없이 다른 데이터베이스 이름, 별명, 호스트 이름 또는 인스턴스로 데이터베이스 복원	테이블은 <i>Datalink_Reconcile_Not_Possible</i> 상태가 됩니다.	N/A	선택적, 하지만 <i>Datalink_Reconcile_Not_Possible</i> 상태의 테이블은 수동으로 수정해야 합니다.
다른 데이터베이스 이름, 별명, 호스트 이름 또는 인스턴스로 데이터베이스 복원 및 롤 포워드	조치 없음	테이블은 <i>Datalink_Reconcile_Not_Possible</i> 상태가 됩니다.	선택적, 하지만 <i>Datalink_Reconcile_Not_Possible</i> 상태의 테이블은 수작업으로 수정해야 합니다.
사용할 수 없는 백업(Data Links 서버에서 이미지가 가비지 수집되었습니다)으로부터 롤 포워드 1 없이 데이터베이스 복원, WITHOUT DATALINK 옵션 지정 또는 비지정	테이블이 <i>Datalink_Reconcile_Pending</i> 상태가 됨	조치 없음	필수
사용할 수 없는 백업(Data Links 서버에서 이미지가 가비지 수집되었습니다)으로부터 데이터베이스 복원 및 롤 포워드, WITHOUT DATALINK 옵션 지정 또는 비지정	조치 없음	테이블이 <i>Datalink_Reconcile_Pending</i> 상태가 됨	필수
사용할 수 없는 백업(Data Links 서버에서 이미지가 가비지 수집되었습니다)으로부터 테이블 공간 복원 및 롤 포워드	조치 없음	테이블이 <i>Datalink_Reconcile_Pending</i> 상태가 됨	필수

주:

- 오프라인 백업을 사용한 복원 및 WITHOUT ROLLING FORWARD 옵션(*logretain* 설정) 또는 오프라인 백업을 사용한 복원(*logretain* 해제).
- 완전한 백업은 모든 필수 Data Links 서버가 수행 중일 때 취해진 백업입니다. 불완전한 백업은 최소한 하나의 필수 Data Links 서버가 수행 중이지 않을 때 취해진 백업입니다.
- 데이터베이스 복원 조작에 사용된 백업 이미지가 데이터베이스 복원 이후에 롤 포워드를 설정 또는 해제하여 수행된 경우 빠른 조정 처리는 수행되지 않습니다. 이 경우, DATALINK 컬럼이 있는 모든 테이블은 *Datalink_Reconcile_Pending* 상태가 됩니다.

조정이 필요한 상황 감지

다음은 조정 유틸리티를 수행해야 하는 몇가지 상황입니다.

- 전체 데이터베이스가 복원되어 특정 시점으로 롤 포워드되었습니다. 전체 데이터베이스가 확장된 트랜잭션으로 롤 포워드되었으므로, 점검 보류 상태가 되는 테이블은 없습니다(참조 제한조거 또는 점검 제한조건으로 인해). 데이터베이스의 모든 데이터는 일치된 상태가 됩니다. 하지만 DATALINK 컬럼은 DB2 Data Links Manager의 메타데이터와 동기화되지 않을 수 있으므로, 조정이 필요합니다.

이런 상황에서, DATALINK 데이터가 있는 테이블은 이미 DRP 상태가 됩니다. 이러한 테이블 각각에 대해 조정 유틸리티를 호출해야 합니다.

- DB2 Data Links Manager를 수행하는 특정 Data Links 서버는 메타데이터의 추적을 상실합니다. 이는 여러 이유로 발생할 수 있습니다. 예를 들어, 다음과 같습니다.

- Data Links 서버가 콜드 시동되었습니다.
- Data Links 서버 메타데이터가 백 레벨 상태로 복원되었습니다.

SQL UPDATE 및 DELETE 도중과 같은 일부 상황에서, DB2가 Data Links 서버의 메타데이터에 문제가 있음을 발견할 수 있습니다. 이 경우, SQL문은 실패합니다. SET CONSTRAINTS 문을 사용하여 테이블을 DRP 상태로 만든 뒤, 해당 테이블에 조정 유틸리티를 수행하십시오.

- 파일 시스템이 사용 불가능하고(예를 들어, 디스크 손상으로 인해) 현재 상태로 복원되지 않습니다. 이 경우, 파일이 누락될 수 있습니다.
- DB2 Data Links Manager가 데이터베이스에서 삭제되고, 그 DB2 Data Links Manager를 참조하는 DATALINK FILE LINK CONTROL 값이 있습니다. 그와 같은 테이블에서 조정 유틸리티를 수행해야 합니다.

데이터베이스 복구를 위한 User Exit

『아카이브 및 검색 고려사항』에서 다음의 문단은 올바르지 못한 내용으로 목록에서 제거해야 합니다.

User Exit는 원격 클라이언트가 DB2 서버로의 연결을 잃어버린 경우 인터럽트될 수 있습니다. 즉, User Exit를 통한 로그의 아카이브 처리 중에 다른 SNA 연결 클라이언트 중 하나가 죽었거나 전원이 차단되어 신호(SIGUSR1)가 서버로 보내지는 결과를 가져올 수 있습니다. 이 서버는 신호를 User Exit에 보내어 인터럽트가 발생하도록 합니다. User Exit 프로그램은 인터럽트 점검을 위해 수정될 수 있으며, 이후 계속될 수 있습니다.

오류 처리 절에는 참고 3의 내용을 다음의 정보로 바꿔야 하는 참고 목록이 있습니다.

- User Exit 프로그램 요청이 5분간 보류됩니다. 이 시간 동안 리턴 코드를 작성한 로그 파일 요청을 비롯한 모든 요청이 무시됩니다.

5분의 요청 처리 보류 이후에 다음 요청이 처리됩니다. 이 요청 처리에서 오류가 발생하지 않으면, 새로운 User Exit 프로그램 요청의 처리가 계속되며 DB2가 이전 아카이브에 실패했거나 보류되었던 로그 파일에 대한 아카이브 요청을 다시 발행합니다. 8보다 큰 리턴 코드가 재시도 중에 생성된 경우, 요청은 추가로 5분간 보류됩니다. 5분의 보류는 문제점이 정정되거나 데이터베이스를 중지하고 다시 시작할 때까지 계속됩니다.

모든 응용프로그램이 데이터베이스에서 연결해제되고 데이터베이스가 다시 열렸으면, DB2는 이전 데이터베이스 사용에서 성공적으로 아카이브하지 못한 로그 파일에 대한 아카이브 요청을 발행합니다.

User Exit 프로그램이 로그 파일을 아카이브하는 데 실패한 경우, 디스크는 로그 파일들로 가득 차고 이러한 로그 파일들을 형식화하기 위한 추가 작업 때문에 성능이 저하될 수 있습니다. 디스크가 가득 차면, 데이터베이스 관리 프로그램은 데이터베이스 변경에 대한 더 이상의 응용프로그램 요청을 수용하지 않습니다.

로그 파일을 검색하기 위해 User Exit 프로그램을 호출한 경우, 롤 포워드 복구가 보류되지만 ROLLFORWARD DATABASE 유틸리티에서 중지를 지정하지 않은 경우에는 중지되지 않습니다. 중지가 지정되지 않은 경우, 문제점을 정정하고 복구를 재개할 수 있습니다.

다중 데이터베이스 파티션 서버로 명령 발행

『수행할 명령 지정』 절의 맨 아래에 다음을 추가합니다.

백그라운드로 stdin에서 읽어오는 작업이 포함된 korn 셸 셸 스크립트를 수행할 때, stdin의 경로를 프로세스가 터미널에서 중단되지 않고 읽을 수 있는 소스로 명시적으로 재지정해야 합니다(SIGTTIN 메시지). stdin의 경로를 재지정하기 위해 다음과 같은 스크립트를 실행할 수 있습니다.

```
shell_script </dev/null &
```

입력이 제공되지 않는 경우

마찬가지로 백그라운드로 db2_all을 수행할 때 항상 </dev/null을 지정해야 합니다. 예를 들어, 다음과 같습니다.

```
db2_all ";run_this_command" </dev/null &
```

이를 수행하여 stdin의 경로를 재지정하고 터미널에서 중단되는 것을 피할 수 있습니다.

또 다른 방법은 원격 명령의 출력을 고려하지 않는 경우, db2_all 접두부에서 『daemonize』 옵션을 사용하는 것입니다.

```
db2_all ";daemonize_this_command" &
```

고속 로드간 통신

다음 절이 갱신되었습니다.

VI를 사용하여 수행할 수 있는 DB2

Windows용 DB2 Enterprise - Extended Edition 빠른 시작에서 자세한 설치 내용을 참조하십시오.

Windows용 DB2 Enterprise - Extended Edition 빠른 시작에 설명된 대로 DB2 설치를 완료한 후, 다음 DB2 레지스트리 변수를 설정하고 인스턴스의 각 데이터베이스 파티션 서버에 대해 다음 작업을 수행하십시오.

- Set DB2_VI_ENABLE=ON

db2set 명령을 사용하여 레지스트리 변수의 값을 수정하십시오. **db2_all** 명령을 사용하여 인스턴스의 모든 데이터베이스 파티션에서 **db2set** 명령을 수행하십시오. **db2_all** 명령을 수행하려면 Administrators 그룹의 구성원인 사용자 계정으로 로그인되어 있어야 합니다.

다음의 예에서, 큰 따옴표 안에 있는 ; 문자는 인스턴스의 모든 데이터베이스 파티션 서버에서 요청을 동시에 수행할 수 있게 합니다.

```
db2_all ";db2set DB2_VI_ENABLE=ON"
```

db2_all 명령에 대한 자세한 정보는, *관리 안내서: 구현에서 "다중 데이터베이스 파티션 서버에 명령 발행"*을 참조하십시오.

- Set DB2_VI_DEVICE=nic0

예를 들어, 다음과 같습니다.

```
db2_all ";db2set DB2_VI_DEVICE=nic0"
```

주: Synfinity Interconnect가 있으면, 이 변수를 DB2_VI_DEVICE=VINIC로 설정해야 합니다. 장치 이름(VINIC)은 대문자이어야 합니다.

- Set DB2_VI_VIPL=vipl.dll

예를 들어, 다음과 같습니다.

```
db2_all ";db2set DB2_VI_VIPL=vipl.dll"
```

주: 이 예에 사용된 값은 레지스트리 변수의 기본값입니다. *관리 안내서: 성능에서 레지스트리 변수에 대한 자세한 내용을 참조하십시오.*

- MPP 인스턴스에서 db2start를 입력하십시오.
- db2diag.log 파일을 검토하십시오. 각 파티션마다 『VI가 사용 가능 상태』라고 언급하는 메시지가 있어야 합니다.
- FCM(Fast Communications Manager) 구성 매개변수를 갱신해야 할 수도 있습니다. FCM과 관련된 자원 제한사항의 결과로 문제점이 발생된 경우, FCM 구성 매개

변수의 값을 높여야 합니다. FCM 구성 매개변수의 값을 증가시킨 다른 고속 상호 연결 환경으로부터 이동하는 경우, 이들 값을 낮춰야 할 수도 있습니다. 또한, Windows NT에서는 DB2 기본값을 겹쳐쓰도록 DB2NTMEMSIZE 레지스트리 변수를 설정해야 합니다. *관리 안내서: 성능에서* 레지스트리 변수에 대한 자세한 내용을 참조하십시오.

관리 안내서: 성능

응용프로그램 고려사항

분리 레벨 지정

『분리 레벨 지정』이라는 절에서, 다음 단락을 추가하십시오.

응용프로그램을 준비 또는 바인딩할 때 패키지 레벨에서 분리 레벨 설정에 추가하여, 명령문 레벨에서 분리 레벨을 설정할 수 있습니다. 명령문 레벨 분리 레벨은 WITH 절을 사용하여 지정됩니다.

다음 SQL문은 명령문 레벨 분리를 지원합니다.

- SELECT 문
- SELECT INTO
- 검색된 DELETE
- INSERT
- 검색된 UPDATE
- DECLARE CURSOR

명령문 레벨 분리와 연관된 몇가지 조건이 있습니다.

- WITH 절은 부속 조회에서는 사용될 수 없습니다.
- WITH UR 옵션은 읽기 전용 조작에만 적용됩니다. 다른 상황에 사용되면, 명령문은 자동으로 『UR』에서 『CS』로 변경됩니다.
- 명령문에 대한 기본 분리 레벨은 명령문이 바인딩된 패키지의 분리 레벨입니다.
- 명령문 레벨 분리 레벨은 명령문이 나타나는 패키지에 대해 지정된 분리 레벨보다 우선합니다.

최적화 클래스 조정

『최적화 클래스 조정』 절에서, 현재 레벨 2 정보를 다음으로 바꾸십시오.

- 2 - 이 클래스는 최적화 알고리즘이 클래스 1을 상당히 개선하는 최적화 등급을 사용하는 반면, 복잡한 조회에 대해 클래스 3 이상보다 컴파일 비용을 상당히 낮게 유지하도록 지시합니다. 특히 다음 경우에 그렇습니다.
 - 빈도 및 Quantile 비통합 분산 통계를 포함하여, 모든 사용 가능한 통계가 활용됩니다.
 - 매우 드문 경우에만 적용되는 컴퓨터 집약적 규칙을 제외하고, 요약 테이블에 조회 라우팅을 포함하여 모든 조회 재작성 규칙이 적용됩니다.

- Greedy 조인 열거가 사용됩니다.
- 목록 프리페치 및 요약 테이블 라우팅을 포함하여 광범위한 액세스 메소드가 고려됩니다.
- 적용 가능하다면, 스타 조인이 고려됩니다.

최적화 클래스 2는 동적 프로그래밍이 아닌 Greedy 조인 열거를 사용한다는 점을 제외하고 클래스 5와 유사합니다. 이 클래스에는 복잡한 조회에 거의 다른 대안을 고려하지 않고, 따라서 클래스 3 이상보다 적은 컴파일 시간을 소비하는 Greedy 조인 열거 알고리즘을 사용하는 모든 최적화 클래스의 최적화 대부분이 있습니다. 따라서 의사 결정 또는 온라인 분석 처리(OLAP) 환경에서 매우 복잡한 조회에 권장됩니다. 그런 경우, 동일한 조회가 드물게 실행될 경우가 많으므로, 그 액세스 플랜은 다음 조회 발생까지 캐쉬에 남아있을 가능성이 적습니다.

동적 복합 텍스트 명령문

『복합 텍스트 SQL』 절 뒤에 이 새 절을 추가해야 합니다.

동적 복합 텍스트 명령문은 다른 SQL문을 하나의 실행 블록으로 그룹화합니다. 동적 복합 텍스트 명령문 안에서 SQL 변수를 선언하고, SQLSTATE와 연관된 조건을 선언하고, 하나 이상의 SQL 프로시저 명령문을 사용할 수 있습니다. 동적 복합 텍스트 명령문에서 오류가 발생하면, 모든 이전 SQL문은 구간 복원되고 동적복합 텍스트 명령문의 나머지 SQL문은 처리되지 않습니다.

동적 복합 텍스트 명령문은 트리거, SQL 함수, SQL 메소드에 embed될 수 있고, 동적 SQL문의 사용을 통해 발행될 수 있습니다. 이 실행 명령문은 동적으로 준비될 수 있습니다. 명령문을 호출하는 데 특권이 필요하지 않지만 명령문과 연관된 권한부여 ID에는 복합 텍스트 명령문 내의 Embedded SQL문을 호출하기 위해 필요한 특권이 있어야 합니다.

변수는 변수 선언의 부속 명령문에 있습니다. 조건 선언의 SQLSTATE 값에 기반한 부속 명령문에 조건이 있습니다. 동적 복합 텍스트 명령문은 DB2에 의해 하나의 명령문으로 컴파일됩니다. 이 명령문은 작은 제어 플로우 논리이지만 중요한 데이터 플로우에 관련된 짧은 스크립트에 효과적으로 사용될 수 있습니다. 중첩된 복합 텍스트 제어 플로우가 있는 큰 구성의 경우, SQL 프로시저어 사용을 고려해야 합니다.

동적 복합 텍스트 명령문 안에서 사용될 수 있는 여러 제어 흐름 논리 명령문이 있습니다. 여기에는 FOR 문, IF 문, ITERATE 문 및 WHILE 문이 포함됩니다. 이들 명령문에 대한 세부사항 및 기타 지원되는 명령문은 *SQL 참조서*에 있습니다

환경 고려사항

『색인 보조 프로그램 사용』이라는 절 뒤에, 다음 절을 추가하십시오.

큰 색인 키 사용

길이가 255 바이트 이상인 컬럼을 색인 키의 일부로 지정할 수 있습니다. DB2_INDEX_2BYTEVARLEN 레지스트리 변수는 색인 키 길이 저장에 1 바이트 대신 2 바이트를 사용하도록 허용합니다.

레지스트리 변수 변경의 영향을 받는 여러 SQL문이 있습니다. 다음과 같습니다.

- CREATE TABLE. 변수 키 부분이 있는 기본, 외부 및 고유 키가 255 바이트보다 큰 크기를 가질 수 있습니다.
- CREATE INDEX. 변수 키 부분이 있는 고유 색인 및 포함 컬럼을 포함한 모든 색인은 255 바이트보다 큰 크기를 가질 수 있습니다.
- ALTER TABLE. 변수 키 부분이 있는 기본, 외부 및 고유 키가 255 바이트보다 큰 크기를 가질 수 있습니다. 변수 키 부분이 있는 고유 색인 및 포함 컬럼을 포함한 모든 색인은 255 바이트보다 큰 크기를 가질 수 있습니다.

255 바이트로 외부 키 제한은 레지스트리 변수의 값에 관계없이 제거되었습니다. 외부 키에 해당하는 기본 키에 대한 참 조건은 제한 또는 한계를 강요합니다.

더 큰 색인 키를 사용하도록 기존 색인을 전환하려면, 색인을 삭제하고, DB2_INDEX_2BYTEVARLEN 레지스트리 변수를 ON으로 설정한 뒤, 색인을 다시 작성하십시오(더 큰 컬럼을 사용하여).

구문 설명을 포함하여 SQL문에 대한 자세한 정보는, SQL 참조서를 참조하십시오.

시스템 카탈로그 통계

분산 통계 수집 및 사용

『등호 술어에 대한 영향의 예』라고 하는 하위 절에는 술어 $C \leq 10$ 에 대한 설명이 있습니다. 여기서 오류가 -86%라고 되어 있습니다. 이는 잘못된 정보입니다. 문단의 끝에 있는 문장은 다음과 같아야 합니다.

데이터가 고르게 분산되어 있다고 가정하고, 공식 (1)을 사용하면 해당 술어를 충족시키는 행의 수가 1로 예측되며 오류는 -87.5%입니다.

『등호 술어에 대한 영향의 예』라고 하는 하위 절에는 술어 $C > 8.5 \text{ AND } C \leq 10$ 에 대한 설명이 있습니다. 선형 삽입을 사용한 r_2 값의 예상 값은 다음과 같이 변경해야 합니다.

$$r_2 *= \frac{10 - 8.5}{100 - 8.5} \times (\text{값이 } > 8.5 \text{ 및 } \leq 100.0 \text{인 행 수})$$
$$r_2 *= \frac{10 - 8.5}{100 - 8.5} \times (10 - 7)$$

$$r_2 = \frac{1.5}{91.5} \times (3)$$

$$r_2 = 0$$

이 새로운 예의 다음에 오는 문단도 다음과 같이 수정해야 합니다.

최종 예측은 $r_1 + r_2 = 70$ 이고,
오류는 단지 -12.5%입니다.

키탈로그 통계 갱신 규칙

『컬럼 통계 갱신 규칙』 절 안에서, 첫번째 목록 항목의 마지막 bullet 목록 항목을 다음으로 바꾸어야 합니다.

HIGH2KEY는 해당 컬럼에 구별 값이 네 개 이상 있을 때마다 LOW2KEY보다 커야 합니다. 컬럼에 3이하의 구별 값이 있을 경우, HIGH2KEY는 LOW2KEY와 같을 수 있습니다.

부속 구성요소 통계

FixPak 1에서는, 부속 구성요소 통계를 수집하고 사용하기 위한 옵션이 제공됩니다. 이들은 데이터가 공백으로 분리되는 서브필드 또는 부속 구성요소 시리즈 양식으로 된 구조인 경우 컬럼에 있는 데이터 내용에 대한 통계입니다.

예를 들어, 각 행이 하나의 문서에 대해 설명하는 DOCUMENTS 테이블이 데이터베이스에 들어 있다고 가정하고, 이 DOCUMENTS에는 텍스트 검색을 위해 이 문서와 관련된 관련 키워드 목록이 들어 있는 KEYWORDS 컬럼이 있다고 가정합니다. KEYWORDS에 있는 값은 다음과 같을 수 있습니다.

```
'database simulation analytical business intelligence'
'simulation model fruitfly reproduction temperature'
'forestry spruce soil erosion rainfall'
'forest temperature soil precipitation fire'
```

이 예에서, 각 컬럼 값은 5개의 부속 구성요소로 구성되며, 이들 각각에는 공백을 사용하여 다른 단어와 구분되는 하나의 단어(키워드)가 있습니다.

다음과 같이 % match_all 문자를 사용하여 그러한 컬럼에서 LIKE 술어를 지정하는 조회의 경우,

```
SELECT .... FROM DOCUMENTS WHERE KEYWORDS LIKE '%simulation%'
```

최적화 알고리즘이 컬럼의 부속 구성요소 구조에 대한 기본적인 통계를 아는 것이 좋습니다.

SUB_COUNT

부속 구성요소의 평균 수.

SUB_DELIM_LENGTH

부속 구성요소를 구분하는 각 분리문자의 평균 길이. 이 문맥에서 분리문자는 하나 이상의 연속적인 공백 문자입니다.

KEYWORDS 컬럼 예에서, SUB_COUNT는 5이고, SUB_DELIM_LENGTH는 1입니다. 각 분리문자가 하나의 공백 문자이기 때문입니다.

FixPak 1에서, 시스템 관리자는 DB2_LIKE_VARCHAR 레지스트리 변수의 확장 방식으로 이러한 통계의 수집과 사용을 제어합니다. 이 레지스트리 변수는 DB2 UDB 최적화 알고리즘이 다음 양식의 술어를 처리하는 방법에 영향을 줍니다.

```
COLUMN LIKE '%xxxxxx'
```

여기서 xxxxxx는 임의의 문자열입니다. 즉, 검색 값이 % 문자로 시작하는 LIKE 술어입니다(이것은 % 문자로 종료될 수도 있고, 안될 수도 있습니다). 이들을 아래에 있는 "와일드카드 LIKE 술어"라고도 합니다. 모든 술어에서, 최적화 알고리즘은 술어와 일치하는 행 수를 측정해야 합니다. 와일드카드 LIKE 술어의 경우, 최적화 알고리즘은 일치하는 COLUMN에 일련의 구성요소들이 전체 컬럼을 구성하기 위해 함께 연결되는 구조가 있는 것으로 가정하고, 선행 또는 후속 % 문자를 제외하고 문자열의 길이를 근거로 각 요소의 길이를 추정합니다. 새 구문은 다음과 같습니다.

```
db2set DB2_LIKE_VARCHAR=[Y|N|S|num1][,Y|N|num2]
```

여기서

- 첫번째 항목(쉼표 앞에 오는)은 다음을 의미하지만, 부속 구성요소 통계가 양수가 아닌 컬럼에만 해당됩니다.

S	DB2 버전 2에서 사용했던 알고리즘을 사용합니다.
N	고정 길이 부속 구성요소 알고리즘을 사용합니다.
Y (기본값)	알고리즘 매개변수에 대한 기본값과 함께 가변 길이 부속 구성요소 알고리즘을 사용합니다.
num1	가변 길이 부속 구성요소 알고리즘을 사용하고, 알고리즘 매개변수로서 num1을 사용합니다.

- 두 번째 항목(쉼표 뒤에 오는)은 다음을 의미합니다.

N (기본값)	부속 구성요소 통계를 수집하거나 사용하지 않습니다.
Y	부속 구성요소 통계를 수집합니다. 양의 부속 구성요소 통계를 가진 컬럼에 대한 알고리즘 매개변수의 기본값과 함께 해당 통계를 사용하는 변수 길이 부속 구성요소 알고리즘.
num2	부속 구성요소 통계를 수집합니다. 양의 부속 구성요소 통계를 가진 컬럼에 대한 알고리즘 매개변수의 기본값과 함께 해당 통계를 사용하는 변수 길이 부속 구성요소 알고리즘.

DB2_LIKE_VARCHAR 값에 첫번째 항목만 들어 있는 경우, 부속 구성요소 통계는 수집되지 않으며, 이전에 수집했던 통계도 무시됩니다. 지정된 값은 최적화 알고리즘이 이전과 같은 방식으로 와일드카드 LIKE 술어의 선택도를 계산하는 방법에 영향을 줍니다.

- 값이 S인 경우, 최적화 알고리즘은 DB2 버전 2에서 사용했던 것과 같은 알고리즘을 사용하며, 이것은 부속 구성요소 모델을 미리 가정하지 않습니다.
- 값이 N인 경우, 최적화 알고리즘은 부속 구성요소 모델을 미리 가정하는 알고리즘을 사용하며, COLUMN이 가변 길이로 정의되는 경우라도 이를 고정 길이로 간주합니다.
- 값이 Y(기본값) 또는 부동 소수점 상수인 경우, 최적화 알고리즘은 부속 구성요소 모델을 미리 가정하는 알고리즘을 사용하며, COLUMN이 정의된 경우 가변 길이로 간주합니다. 또한 데이터에서 아닌 조회 자체에서 부속 구성요소 통계를 참조합니다. 이 알고리즘에는 % 문자로 묶여진 문자열 대신 구성요소가 존재하는 기간을 지정하는 매개변수("알고리즘 매개변수")가 포함됩니다.
- 값이 Y인 경우, 최적화 알고리즘에서는 알고리즘 매개변수로 기본값 1.9를 사용합니다.
- 값이 부동 소수점 상수인 경우, 최적화 알고리즘에서는 알고리즘 매개변수에 대해 지정된 값을 사용합니다. 이 상수는 0 - 6.2 범위내에 있어야 합니다.

DB2_LIKE_VARCHAR 값에 두 개의 항목이 들어 있고 두 번째가 Y 또는 부동 소수점 상수인 경우, CHAR, VARCHAR, GRAPHIC 또는 VARGRAPHIC 유형의 1 바이트 문자 세트 문자열 컬럼에 있는 부속 문자열 통계가 RUNSTATS 동작 동안 수집되고, 와일드카드 LIKE 술어가 포함된 조회를 컴파일하는 동안 사용됩니다. 최적화 알고리즘에서는 부속 구성요소 모델을 미리 가정하는 알고리즘을 사용하고, 알고리즘 매개변수뿐만 아니라 SUB_COUNT 및 SUB_DELIM_LENGTH 통계를 사용하여 술어의 선택도를 계산합니다. 알고리즘 매개변수는 참조 알고리즘이 지정되는 것과 같은 방식으로 지정됩니다. 즉,

- 값이 Y인 경우, 최적화 알고리즘에서는 알고리즘 매개변수로 기본값 1.9를 사용합니다.
- 값이 부동 소수점 상수인 경우, 최적화 알고리즘에서는 알고리즘 매개변수에 대해 지정된 값을 사용합니다. 이 상수는 0 - 6.2 범위내에 있어야 합니다.

컴파일하는 동안 최적화 알고리즘은 조회에 포함된 컬럼에서 부속 구성요소 통계가 수집되지 않았음을 발견하고, "추론" 부속 구성요소 알고리즘 즉, DB2_LIKE_VARCHAR의 첫번째 항목만이 지정될 때 사용되었던 알고리즘을 사용합니다. 그러므로, 부속 구성요소 통계가 최적화 알고리즘에서 사용되도록 하려면, DB2_LIKE_VARCHAR의 두 번째 항목이 RUNSTATS 및 컴파일 동안 설정되어야 합니다.

부속 구성요소 통계 값들은 SYSIBM.SYSCOLUMNS를 조회하여 볼 수도 있습니다. 예를 들어, 다음과 같습니다.

```
select substr(NAME,1,16), SUB_COUNT, SUB_DELIM_LENGTH
from sysibm.syscolumns where tname = 'DOCUMENTS'
```

SUB_COUNT 및 SUB_DELIM_LENGTH 컬럼은 SYSSTAT.COLUMNS 통계 뷰에서 표시되지 않으므로, 갱신할 수 없습니다.

주: 이 옵션이 사용되면 RUNSTATS에 오랜 시간이 소요될 수 있습니다. 예를 들어, RUNSTATS는 DETAILED 및 DISTRIBUTION 옵션이 사용되지 않는 경우 문자 컬럼이 5개인 테이블에서 15 - 40% 이상 더 소요될 수 있습니다. DETAILED 또는 DISTRIBUTION 옵션이 지정되면, 오버헤드의 절대 양이 같은 경우라도 오버헤드 비율은 줄어듭니다. 이 옵션 사용에 대해 고려하는 경우, 조회 성능의 향상에 대비하여 이 오버헤드를 고려해야 합니다.

SQL 컴파일러 이해

다음 절을 변경해야 합니다.

복제된 요약 테이블

다음의 정보는 이 절에 이미 있는 기존의 정보를 바꾸거나 추가합니다.

복제된 요약 테이블은 조인의 조합을 지원하는 데 사용할 수 있습니다. 예를 들어, 20개의 노드에 대형 사실 테이블이 분산되어 있는 스타 스키마가 있는 경우, 사실 테이블과 차원 테이블 사이의 조인이 이들 테이블을 조합시키는 데 가장 효율적입니다.

모든 테이블을 동일한 노드 그룹에 배치하면 많아야 조합 조인에 대해 올바르게 파티션된 하나의 차원 테이블이 있을 것입니다. 사실 테이블에서의 컬럼 조인은 사실 테이블의 파티션 키에 해당하지 않으므로 다른 모든 차원 테이블은 조합 조인에서 사용할 수 없습니다.

예를 들어, C1에서 파티션된 FACT(C1, C2, C3, ...)라고 하는 테이블과 C1에서 파티션된 DIM1(C1, dim1a, dim1b, ...)이라고 하는 테이블과 C2에서 파티션된 DIM2(C2, dim2a, dim2b, ...)라고 하는 테이블 등이 있을 수 있습니다.

이 예에서 술어 DIM1.C1 = FACT.C1이 조합되기 때문에 FACT와 DIM1 사이의 조인은 완벽함을 알 수 있습니다. 이 두 테이블 모두 컬럼 C1에서 파티션됩니다.

FACT가 컬럼 C2가 아닌 컬럼 C1에서 파티션되기 때문에 DIM2와 술어 WHERE DIM2.C2 = FACT.C2와의 조인을 조합할 수 없습니다.

이 경우, 사실 테이블의 노드 그룹에 DIM2를 복제하는 것이 좋을 것입니다. 이러한 방식으로 각 파티션에서 지역적으로 조인을 수행할 수 있습니다.

주: 여기에서 언급한 복제된 요약 테이블은 데이터베이스 내의 복제와 상관이 없습니다. 데이터베이스 간 복제는 복사 작업 내역, 제어 테이블 및 다른 데이터베이스와 다른 운영 체제에 위치한 데이터와 관련이 있습니다. 데이터베이스 간 복제에 관심이 있는 경우, 복제 안내 및 참조서에서 자세한 내용을 참조하십시오.

복제된 요약 테이블을 작성할 때, 소스 테이블은 단일 노드 노드 그룹 테이블이거나 다중 노드 노드 그룹 테이블일 수 있습니다. 대부분의 경우, 테이블은 작고 단일 노드 노드 그룹에 위치할 수 있습니다. 복제된 요약 테이블을 작성할 때 테이블에서 컬럼의 부분 집합만을 지정하거나 사용되는 술어를 통해 행 수를 제한하거나 두 가지 방법을 모두 사용하여 복제할 데이터에 한계를 설정할 수 있습니다.

주: 데이터 캡처 옵션은 복제된 요약 테이블이 제 기능을 수행하는 데 반드시 필요하지는 않습니다.

복제된 요약 테이블은 다중 노드 노드 그룹에서도 작성할 수 있습니다. 이 노드 그룹은 대형 테이블들을 배치한 노드 그룹과 동일합니다. 이 경우, 소스 테이블의 사본이 노드 그룹의 모든 파티션에서 작성됩니다. 대형 사실 테이블과 차원 테이블 간의 조인은 소스 테이블을 모든 파티션에 브로드캐스트해야 한다기보다는 이 환경에서 지역적으로 수행할 가능성이 큽니다.

복제된 테이블의 색인은 자동으로 작성되지 않습니다. 색인이 작성되며 소스 테이블에 식별된 것과 다를 수 있습니다.

주: 복제된 테이블에서는 고유 색인을 작성하거나 제한조건을 설정할 수 없습니다. 따라서, 소스 테이블에 없는 제한조건 위반이 방지됩니다. 이러한 제한조건들은 소스 테이블에 동일한 제한조건이 있더라도 허용되지 않습니다.

REFRESH 명령문을 사용한 후에 다른 테이블과 같이 복제된 테이블에서 RUNSTATS를 수행해야 합니다.

복제된 테이블은 조회 안에 직접 참조할 수 있습니다. 그러나, 특정 파티션에서 테이블 데이터를 보기 위해 NODENUMBER() 술어를 복제된 테이블과 함께 사용할 수 없습니다.

작성한 복제된 요약 테이블이 사용되었는지 확인하려면(소스 테이블을 참조한 조회가 있는 것으로 가정), EXPLAIN 기능을 사용할 수 있습니다. 먼저, EXPLAIN 테이블이 있는지 확인합니다. 그런 다음, 관심이 있는 SELECT문에 대한 Explain 플랜을 작성합니다. 마지막으로, db2exfmt 유틸리티를 사용하여 EXPLAIN 출력을 형식화합니다.

최적화 알고리즘에 의해 선택된 액세스 플랜은 조인해야 하는 정보에 따라 복제된 요약 테이블을 사용할 수도 있고 사용하지 않을 수도 있습니다. 복제된 요약 테이블을 사용하지 않는 경우는 최적화 알고리즘이 원래의 소스 테이블을 노드 그룹의 다른 파티션에 브로드캐스트하는 것이 보다 저렴한 것으로 판별한 경우에 발생할 수 있습니다.

데이터 액세스 개념 및 최적화

『색인 스캔 개념』의 『다중 색인 액세스』 절이 변경되었습니다.

절의 끝에 있는 주의 이전에 다음 정보를 추가하십시오.

여러 색인을 스캔할 때 동적 비트맵의 성능 이점을 실현하려면, 정렬 힙 크기(*sortheap*) 데이터베이스 구성 매개변수와 정렬 힙 임계값(*sheapthres*) 데이터베이스 관리 프로그램 구성 매개변수의 값을 변경해야 할 수도 있습니다.

동적 비트맵이 액세스 플랜에서 사용될 경우 추가 정렬 힙(heap) 공간이 필요합니다. *sheapthres*가 *sortheap*에 상대적으로 근접하도록 설정될 경우(즉, 동시 조회당 두 번이나 세 번보다 적을 경우), 여러 색인 액세스를 가지고 있는 동적 비트맵은 참여하는 최적화 알고리즘보다 적은 양의 메모리를 사용하여 작업해야 합니다.

해결책은 *sortheap*에 상대적으로 *sheapthres*의 값을 증가시키는 것입니다.

『술어에 관련된 용어』의 『스타 조인의 검색 전략』 절이 변경되었습니다.

절의 끝에 다음 정보를 추가하십시오.

스타 조인 기술의 일부로 작성되어 사용되는 동적 비트맵은 정렬 힙(heap) 메모리를 사용합니다. 정렬 힙 크기(*sortheap*) 데이터베이스 구성 매개변수에 관한 추가 정보는 관리 안내서: 성능에서 "DB2 구성"을 참조하십시오.

조작 성능

데이터베이스 버퍼 풀 관리

『데이터베이스 버퍼 풀 관리』라는 절에서, 『버퍼 풀 작성시, 기본값으로 페이지 크기는 4 KB입니다』로 시작하는 단락 뒤에 다음 정보를 추가하십시오.

Windows 2000으로 작업시, 최대 64 GB의 버퍼 풀 크기는 DB2 및 운영 체제의 크기보다 작게 지원됩니다(이는 DB2가 시스템에서 기본 제품임을 가정합니다). 이 지원은 Microsoft Address Windowing Extensions(AWE)를 통해 사용 가능합니다.

AWE는 어떤 버퍼 풀 크기로도 사용될 수 있지만, 큰 버퍼 풀에서 AWE가 필요하다면, 기타 권장 Windows 제품이 있습니다. Windows 2000 Advanced Server는 최대 8 GB의 메모리를 지원합니다. Windows 2000 Data Center Server는 최대 64 GB의 메모리를 지원합니다.

DB2 및 Windows 2000은 AWE 버퍼 풀을 지원하도록 올바르게 구성되어야 합니다. AWE를 이용하는 버퍼 풀이 데이터베이스에 존재해야 합니다.

3 GB 사용자 공간을 할당하려면, /3 GB Windows 2000 부트 옵션을 사용하십시오. 그러면 큰 AWE 창 크기를 사용할 수 있습니다. AWE 메모리 인터페이스를 통해 4 GB 이상의 메모리에 액세스할 수 있으려면, /PAE Windows 2000 부트 옵션을 사용하십시오. 올바른 부트 옵션을 선택했는지 검증하려면, 제어에서 시스템을 선택한 뒤, 『시작 및 복구』를 선택하십시오. 드롭다운 목록에서 사용 가능한 부트 옵션을 볼 수 있습니다. 원하는 부트 옵션(/3 GB 또는 /PAE)이 선택되어 있으면, AWE 지원 설정의 다

음 task로 진행할 준비가 되었습니다. 선택할 수 있는 옵션 중에 원하는 것이 없다면, 시스템 드라이브의 boot.ini 파일에 옵션을 추가해야 합니다. boot.ini 파일에는 운영 체제가 시작될 때 수행되어야 하는 조치 목록이 들어 있습니다. /3 GB 또는 /PAE 또는 둘 다를(공백으로 분리) 기존 매개변수 목록의 끝에 추가하십시오. 이 변경된 파일을 저장했으면, 위에 언급된 대로 올바른 부트 옵션을 검증하고 선택할 수 있습니다.

또한 『메모리의 잠금 페이지』-권한이 DB2가 설치될 사용자와 연결되도록 Windows 2000을 수정해야 합니다. 『메모리의 잠금 페이지』-권한을 설정하려면, DB2를 설치한 사용자로 Windows 2000에 로그인하고, Windows 2000의 시작 메뉴에서 『관리 도구』 폴더 그리고 『지역 보안 규정』 프로그램을 선택하십시오. 지역 규정 아래에서, 『메모리의 잠금 페이지』-권한에 대한 사용자 권한 지정을 선택할 수 있습니다.

DB2는 DB2_AWE 레지스트리 변수 설정을 요구합니다. 이 레지스트리 변수를 올바르게 설정하려면, AWE 지원을 허용하려는 버퍼 풀의 버퍼 풀 ID를 알아야 합니다. 또한 할당할 물리적 페이지 및 주소 창 페이지의 수를 알아야 합니다. 할당할 물리적 페이지의 수는 총 사용 가능한 물리적 페이지 보다 적은 어떤 값이어야 합니다. 선택되는 실제 수는 작업 환경에 따라 다릅니다. 예를 들어, DB2와 데이터베이스 응용프로그램만 시스템에서 사용되는 환경일 경우, DB2_AWE 변수에 사용되는 값으로 물리적 페이지의 총 크기보다 1/2 또는 1 GB 작도록 선택할 수 있습니다. 기타 비데이터베이스 응용프로그램이 시스템을 사용하는 환경일 경우, 기타 응용프로그램에 보다 많은 페이지를 허용하기 위해 총계에서 빼는 값을 늘려야 합니다. DB2_AWE 레지스트리 변수에 사용되는 수는 AWE 지원에 그리고 DB2가 사용할 물리적 페이지의 수입니다. 주소 창 페이지의 상한은 /3 GB Windows 부트 옵션이 사용 가능할 경우, 1.5 GB 또는 2.5 GB입니다.

DB2 레지스트리 변수 DB2_AWE 설정에 대한 자세한 정보는, 이 절 후반의 『DB2 레지스트리 및 환경 변수』에서 신규 및 변경된 레지스트리 변수 테이블을 참조하십시오.

다중 데이터베이스 버퍼 풀 관리

『다중 데이터베이스 버퍼 풀 관리』라는 절 안에서, 『데이터베이스 설계시, 페이지 크기가 8 KB인 테이블이 가장 좋다고 판단했을 수 있습니다』로 시작하는 문단위에 다음 문단을 추가하십시오.

Windows 2000에서 작업시, DB2_AWE 레지스트리 변수는 카탈로그와 구성 파일의 버퍼 풀 크기 설정값에 우선할 수 있습니다. 이 레지스트리 변수를 사용하면 버퍼 풀 크기가 최대 대략 64 GB까지 허용됩니다.

같은 절에서, 참고 직전의 문단을 다음으로 바꾸십시오.

데이터베이스 관리 프로그램이 최소 크기 값으로 시작하도록 하는 이유는 데이터베이스에 연결을 허용하는 것입니다. 그런 뒤 버퍼 풀 크기를 재구성하거나, 올바른 버퍼 크

기로 데이터베이스를 재시작할 목적으로 기타 중요한 작업을 수행할 수 있습니다. 그런 상태에서 확장된 시간에 데이터베이스 운영을 고려하지 마십시오.

『카탈로그 및 사용자 테이블 재구성』 절에서, 『REORG 유틸리티를 사용하여 임시 테이블 공간을 지정할 수 있습니다...』로 시작하는 문단의 마지막 문장(짧은 목록과 함께)을 다음으로 바꾸십시오.

같은 테이블 공간을 사용하여 테이블을 재구성하는 것이 더 빠르지만 많은 로깅이 발생하므로 재구성된 테이블에 충분한 공간이 있어야 합니다. 임시 테이블 공간을 지정할 경우, SMS 임시 테이블 공간을 지정하도록 일반적으로 권장됩니다. DMS 임시 테이블 공간은 이 유형의 테이블 공간을 사용하는 REORG만 진행될 수 있으므로 권장되지 않습니다.

『메모리 확장』 절에서, 이 절의 세 번째 문단 뒤에 다음 문단을 추가하십시오.

DB2_AWE 레지스트리 변수를 사용하여 Windows 2000 Address Windowing Extensions(AWE) 버퍼 풀을 할당할 때, 확장 저장영역 캐쉬는 사용할 수 없습니다.

조정자(governor) 사용

『조정자(governor) 구성 파일 작성』 절에서, 스케줄 조치 설명 뒤의 첫 번째 문단에서 첫 번째 문장은 다음으로 바뀌어야 합니다.

응용프로그램에 하나 이상의 규칙이 적용될 경우, 모든 규칙이 적용됩니다. 규칙 및 설정된 한계에 따라, 첫 번째 만난 규칙 한계와 연관된 조치가 최초로 적용되는 조치입니다.

DB2 구성

다음 매개변수를 변경해야 합니다.

정렬 힙 크기(sortheap)

『권장사항』 절이 변경되었습니다. 여기의 정보는 다음과 같아야 합니다.

정렬 힙(heap)에 대해 작업할 때, 다음을 고려해야 합니다.

- 적절한 색인은 정렬 힙의 사용을 최소화할 수 있습니다.
- 해쉬 조인 버퍼와 동적 비트맵(색인 ANDing 및 스타 조인에 사용됨)은 정렬 힙 메모리를 사용합니다. 이 기술을 사용할 때 이 매개변수의 크기를 증가시키십시오.
- 자주 대규모 정렬이 필요할 경우 이 매개변수의 크기를 증가시키십시오.
- ...(항목의 나머지는 변경되지 않습니다).

정렬 힙 임계값(sheaphtres)

이 매개변수에 대한 설명에서 뒤에서 두 번째 문장이 변경되었습니다. 그 문장은 다음과 같아야 합니다.

정렬 힙(heap)을 사용하는 조작들의 예로는 정렬, 동적 비트맵(색인 ANDing 및 스타 조인에 사용됨) 그리고 테이블이 메모리에 있는 조작이 있습니다.

다음의 정보를 이 매개변수의 설명에 추가합니다.

단일 노드에서 다중 노드 환경으로 이동할 때 이 매개변수의 값을 증가시킬 이유가 없습니다. 단일 노드(DB2 EE에서) 환경에서 데이터베이스 및 데이터베이스 관리 프로그램 구성 매개변수를 조정했으면, 다중 노드(DB2 EEE에서) 환경에서 동일한 값이 대부분의 경우 제대로 작동됩니다.

정렬 힙 임계값 매개변수는 데이터베이스 관리 프로그램 구성 매개변수로서 전체 DB2 인스턴스에 적용됩니다. 다른 노드나 파티션에서 이 매개변수에 다른 값을 설정하는 유일한 방법은 하나 이상의 DB2 인스턴스를 작성하는 것입니다. 그러면, 다른 노드 그룹에서 다른 DB2 데이터베이스를 관리해야 합니다. 이러한 배열은 파티션 데이터베이스 환경의 여러 가지 장점의 실패를 초래합니다.

레벨 자동 업그레이드 전의 잠금 목록 최대 퍼센트(maxlocks)

다음 변경사항은 "레벨 자동 업그레이드 전의 잠금 목록 최대 퍼센트(maxlocks)" 데이터베이스 구성 매개변수의 권장사항 절에 해당됩니다.

권장사항: 다음 공식은 응용프로그램이 평균 잠금 수의 두 배를 보유하도록 허용하는 *maxlocks*를 설정할 수 있게 해줍니다.

$$\text{maxlocks} = 2 * 100 / \text{maxappls}$$

여기서 2는 평균의 두 배를 이루기 위해 사용되고 100은 허용되는 최대 퍼센트 값을 나타냅니다. 동시에 수행되는 응용프로그램의 수가 적다면, 다음 공식을 첫번째 공식 대신 사용할 수 있습니다.

$$\text{maxlocks} = 2 * 100 / (\text{동시에 실행하는 응용프로그램의 평균 수})$$

maxlocks 설정시 고려해야 할 사항 중 하나는 잠금 목록의 크기(*locklist*)와 관련하여 사용하는 것입니다. 잠금 레벨 자동 업그레이드가 발생하기 전에 응용프로그램이 보유하는 잠금 수의 실제 한계는 다음과 같습니다.

$$\text{maxlocks} * \text{locklist} * 4096 / (100 * 36)$$

여기서 4096은 페이지의 바이트 수, 100은 *maxlocks*에 허용되는 최대 퍼센트 값 그리고 36은 잠금 당 바이트 수입니다. 1000 개의 잠금을 요구하는 응용프로그램이 있을

경우, 결과가 1000보다 크도록 이 공식에서 *maxlocks* 및 *locklist*에 대한 값을 선택해야 합니다. *maxlocks*에 10, *locklist*에 100을 사용하면 이 공식의 결과는 필요한 1000보다 커집니다.

*maxlocks*를 너무 낮게 설정하면, 다른 동시 응용프로그램을 위한 잠금 공간이 충분해도 잠금 레벨 자동 업그레이드가 일어납니다. *maxlocks*를 너무 높게 설정하면, 적은 수의 응용프로그램이 잠금 공간 대부분을 소비하고, 다른 응용프로그램은 잠금 레벨 자동 업그레이드를 수행해야 합니다. 이 경우의 잠금 레벨 자동 업그레이드 필요는 낮은 동시성을 초래합니다.

데이터베이스 시스템 모니터를 사용하여 이 구성 매개변수를 추적하여 조정할 수 있습니다.

DB2/DB2 Data Links Manager/Data Links 액세스 토큰 만료 간격 (dl_expint) 구성

문서와는 반대로, *dl_expint*를 "-1"로 설정하면, 액세스 제어 토큰이 만료됩니다. 이에 대한 일시적인 해결책은 *dl_expint*를 최대 값, 31536000 (초)로 설정하는 것입니다. 이는 일 년의 만료 시간에 해당하며, 모든 응용프로그램에 대해 적합해야 합니다.

MIN_DEC_DIV_3 데이터베이스 구성 매개변수

SQL에서 소수 나눗셈에 대한 스케일 계산 변경을 사용하는 빠른 방법으로 *MIN_DEC_DIV_3* 데이터베이스 구성 매개변수가 추가되었습니다. *MIN_DEC_DIV_3*은 YES 또는 NO로 설정할 수 있습니다. *MIN_DEC_DIV_3*의 기본값은 NO입니다.

MIN_DEC_DIV_3 데이터베이스 구성 매개변수는 그 결과인 나눗셈을 포함한 소수 연산의 스케일을 변경합니다. 값이 NO이면, 스케일은 31-p+s'로 계산됩니다. 추가 정보는, *SQL* 참조서, "SQL의 소수 연산"을 참조하십시오. YES로 설정하면, 스케일은 MAX(3, 31-p+s')로 계산됩니다. 이는 소수 나눗셈의 결과가 항상 최소한 3의 스케일을 갖도록 만듭니다. 정확도는 항상 31입니다.

이 데이터베이스 구성 매개변수를 변경하면 기존 데이터베이스에 대한 응용프로그램에 변경을 초래할 수 있습니다. 이 데이터베이스 구성 매개변수를 변경함으로써 소수 나눗셈에 대한 결과 스케일이 영향을 받을 때 이런 경우가 발생합니다. 아래 나열된 것은 응용프로그램에 영향을 줄 수 있는 몇몇 가능한 시나리오입니다. 이런 시나리오는 기존 데이터베이스가 있는 데이터베이스 서버에서 *MIN_DEC_DIV_3*을 변경하기 전에 반드시 고려해야 합니다.

- 뷰 컬럼의 결과 스케일이 변경되면, 하나의 설정으로 환경에 정의되어 있는 뷰를 데이터베이스 구성 매개변수가 변경된 후에 참조할 때 *SQLCODE* -344로 실패할 수 있습니다. 메시지 *SQL0344N*은 순환 공통 테이블 표현식을 의미하지만, 오브젝트 이름(첫번째 토큰)이 뷰일 경우 이 오류를 피하려면 뷰를 삭제하고 다시 작성해야 합니다.

- 정적 패키지는 패키지가 암시적 또는 명시적으로 리바인드되기 전에는 작동을 변경하지 않습니다. 예를 들어, 값을 NO에서 YES로 변경한 후, 추가 스케일 자리 수는 리바인드가 발생하기 전까지 결과에 포함되지 않을 수 있습니다. 변경된 정적 패키지에 대해, 명시적 리바인드 명령을 사용하여 리바인드를 강제 수행할 수 있습니다.
- 소수 나눗셈과 관련된 점검 제한조건은 이전에 받아들여졌던 일부 값들을 제한할 수 있습니다. 이제 그런 행은 제한조건을 위반하지만 점검 제한조건에 관련된 컬럼들 중 하나가 갱신되거나 **SET INTEGRITY** 명령과 **IMMEDIATE CHECKED** 옵션이 처리되기까지 발견되지 않습니다. 그런 제한조건 점검을 강제수행하려면, **ALTER TABLE** 명령을 수행하여 점검 제한조건을 삭제한 뒤 **ALTER TABLE** 명령을 수행하여 제한조건을 다시 추가하십시오.

주: DB2 버전 7에는 또한 다음 제한사항이 있습니다.

1. **GET DB CFG FOR DBNAME** 명령이 MIN_DEC_DIV_3 설정을 표시하지 않습니다. 현재 설정을 판단하는 가장 좋은 방법은 소수 나눗셈 결과의 부가 효과를 관찰하는 것입니다. 예를 들어, 다음 명령문에 대해 생각해 봅시다.
VALUES (DEC(1,31,0)/DEC(1,31,5))

이 명령문이 sqlcode SQL0419N을 리턴하면 데이터베이스에 MIN_DEC_DIV_3 지원이 없거나 OFF로 설정되어 있습니다. 명령문이 1.000을 리턴하면 MIN_DEC_DIV_3은 ON으로 설정되어 있습니다.
2. MIN_DEC_DIV_3은 ? **UPDATE DB CFG** 명령을 수행할 때 구성 키워드의 목록에 나타나지 않습니다.

응용프로그램 제어 힙 크기(app_ctl_heap_sz)

이 매개변수의 텍스트는 다음과 같아야 합니다.

내부 병렬을 사용하는(intra_parallel=ON) 파티션된 데이터베이스와 파티션되지 않은 데이터베이스의 경우, 이것은 응용프로그램 제어 힙(heap)에 할당되는 공유 메모리 영역의 크기입니다. 내부 병렬을 사용하지 않는(intra_parallel=OFF) 파티션되지 않은 데이터베이스의 경우, 이것은 힙(heap)에 할당될 최대 개인용 메모리입니다. 파티션당 연결당 하나의 응용프로그램 제어 힙(heap)이 있습니다.

응용프로그램 제어 힙(heap)은 주로 같은 요청에 대해 작업하는 에이전트들 사이에 정보 공유를 위해 그리고 파티션된 데이터베이스 환경에서 SQL문을 표시하는 실행 섹션 저장을 위해 필요합니다. 이 힙(heap)의 사용은 병렬 정도가 1보다 작거나 같은 조회 수행시 파티션되지 않은 데이터베이스의 최소한도입니다.

또한 이 힙(heap)은 선언된 임시 테이블에 대한 설명자 정보 저장에 사용됩니다. 명시적으로 삭제되지 않은 모든 선언된 임시 테이블에 대한 설명자 정보는 이 힙(heap)의 메모리에 보존되고 선언된 임시 테이블이 삭제될 때까지 삭제될 수 없습니다.

『권장사항』 부분은 변경되지 않았습니다.

데이터베이스 시스템 모니터 힙(heap) 크기 (mon_heap_sz)

지역 및 원격 클라이언트가 있는 OS/2 및 Windows NT 데이터베이스 서버 그리고 지역 클라이언트가 있는 위성 데이터베이스 서버에 대한 기본값이 24에서 32로 변경되었습니다. 범위는 변경되지 않았습니다.

실행 중인 응용프로그램의 최대 수 (maxappls)

모든 플랫폼에 대한 상한이 64 000에서 60 000으로 변경되었습니다. 기본값은 변경되지 않았습니다.

복구 범위 및 소프트 점검점 간격(softmax)

측정 단위가 한 기본 로그 파일의 백분율로 변경되었습니다.

수정된 페이지 추적 사용 (trackmod)

구성 유형: 데이터베이스

매개변수 유형: 구성용

기본값 [범위]: Off [On; Off]

이 매개변수를 ON으로 설정하면, 데이터베이스 관리 프로그램은 가장 최근에 전체 백업이 수행된 이후에 데이터베이스의 어떤 페이지가 변경되었는지 추적합니다. 이는 백업 유틸리티가 매 페이지를 개별적으로 검토하지 않고 어떤 페이지를 증분 백업에 포함시켜야 할지 결정하도록 해줍니다. SMS 테이블 공간의 경우, 이 추적의 세분성은 테이블 페이스 레벨에 있습니다. DMS 테이블 공간의 경우, 세분성은 데이터에 대한 extent 레벨에 있고, 기타 페이지 유형에 대해서는 테이블 공간 레벨에 있습니다. 이 매개변수를 ON으로 설정한 후, 증분 백업의 베이스라인이 되도록 전체 데이터베이스를 백업해야 합니다.

데이터베이스 로그 경로 변경 (newlogpath)

구성 유형: 데이터베이스

매개변수 유형: 구성용

기본값 [범위]: Null [유효한 경로 또는 장치]

관련 매개변수: 로그 파일의 위치(logpath), 데이터베이스가 일치함(database_consistent)

이 매개변수로, 로그 파일이 저장된 위치를 변경하기 위해 최대 242 바이트의 문자열을 지정할 수 있습니다. 문자열은 경로 이름 또는 원시 장치를 가리킬 수 있습니다. 문자열이 경로 이름을 가리킬 경우, 상대 경로 이름이 아닌 완전한 경로 이름이어야 합니다.

주: 파티션된 데이터베이스 환경에서, 노드 번호가 자동으로 경로에 추가됩니다. 이는 다중 논리 노드 구성에서 경로의 고유성을 유지하기 위해 수행됩니다.

장치를 지정하려면, 운영 체제가 장치로 식별하는 문자열을 지정하십시오. 예를 들어, Windows NT에서 다음과 같이 지정하십시오.

```
\\.\d: 또는 \\.\PhysicalDisk5
```

주: 장치에 로그를 쓰려면 서비스 팩 3이 설치된 Windows NT 버전 4.0이 있어야 합니다.

UNIX 기반 플랫폼에서는 다음과 같이 지정하십시오.

```
/dev/rdblog8
```

주: AIX, Windows 2000, Windows NT, Solaris, HP-UX, NUMA-Q 및 Linux 플랫폼에서만 장치를 지정할 수 있습니다.

다음 두 경우가 모두 발생할 때까지 새로운 설정은 logpath의 값이 되지 않습니다.

- database_consistent 매개변수에 의해 지시된 대로 데이터베이스는 일치된 상태에 있습니다.
- 모든 사용자가 데이터베이스로부터 연결해제됩니다.

첫번째 새로운 연결이 데이터베이스에 작성되면, 데이터베이스 관리 프로그램은 logpath에 의해 지정된 새로운 위치로 로그를 이동합니다.

이전 로그 경로에 로그 파일이 있어야 합니다. 이러한 로그 파일은 아카이브되지 않을 수도 있습니다. 이러한 로그 파일은 수동으로 아카이브해야 합니다. 또한, 이 데이터베이스에서 복제를 수행 중일 경우, 복제가 여전히 로그 경로 변경 전의 로그 파일을 필요로 할 수 있습니다. 데이터베이스가 User Exit 사용 가능(userexit) 데이터베이스 구성 매개변수가 "Yes"로 설정되어 구성되었을 경우 그리고 모든 로그파일이 DB2에 의해 자동으로 또는 사용자에게 의해 수동으로 아카이브되었다면, DB2는 로그 파일을 검색하여 복제 프로세스를 완료할 수 있습니다. 그렇지 않으면, 이전 로그 경로에서 새 로그 경로로 파일을 복사해야 합니다.

권장사항: 이상적으로, 로그 파일은 입출력이 많지 않은 물리적 디스크에 있습니다. 예를 들어, 로그를 운영 체제나 볼륨이 큰 데이터베이스와 같은 디스크에 두지 마십시오. 이는 입출력 대기와 같은 오버헤드가 최소인 효율적인 로그 활동이 가능하도록 해줍니다.

데이터베이스 시스템 모니터를 사용하여 데이터베이스 로그와 관련된 입출력의 수를 추적할 수 있습니다.

자세한 정보는, 시스템 모니터 안내서 및 참조서에서 다음 모니터 요소 설명을 참조하십시오.

- log_reads (읽은 로그 페이지 수)
- log_writes (쓴 로그 페이지 수)

앞의 데이터 요소는 데이터베이스 로그와 관련된 입출력 활동의 양을 리턴합니다. 운영 체제 모니터 도구를 사용하여 기타 디스크 입출력 활동에 대한 정보를 수집한 뒤, 두 유형의 입출력 유형을 비교할 수 있습니다.

로그 파일의 위치 (logpath)

구성 유형: 데이터베이스

매개변수 유형: 정보용

관련 매개변수: 데이터베이스 로그 경로 변경 (newlogpath)

이 매개변수에는 로그 목적으로 사용되는 현재 경로가 들어 있습니다. 이 매개변수는 newlogpath 매개변수에 대한 변경이 유효하게 된 후에 데이터베이스 관리 프로그램에 의해 설정되므로, 사용자가 직접 변경할 수 없습니다. 데이터베이스가 작성될 때, 그에 대한 복구 로그 파일이 데이터베이스가 있는 디렉토리의 서브디렉토리에 작성됩니다. 기본값은 데이터베이스가 작성된 디렉토리 아래에 SQLOGDIR이라는 서브디렉토리입니다.

잠금 목록용 최대 저장영역 (locklist)

최대 값이 60 000에서 524 288로 증가되었습니다.

DB2 레지스트리 및 환경 변수

다음의 레지스트리 값은 새로운 값이거나 변경이 필요합니다.

새로운 레지스트리 변수 및 변경된 레지스트리 변수의 표

표 6. 레지스트리 변수

변수 이름 설명	운영 체제	값
DB2MAXFSCRSEARCH	모두	기본값=5 값: -1, 1 - 33554

테이블에 레코드를 추가할 때 검색할 여유 공간 레코드의 수를 지정합니다. 기본값은 5개의 여유 공간 제어 레코드를 검색하는 것입니다. 이 값을 수정하면, 삽입 속도를 공간 재사용과 균형을 유지할 수 있습니다. 공간 재사용을 위해 최적화하려면 큰 값을 사용하십시오. 삽입 속도를 위해 최적화하려면 작은 값을 사용하십시오. 이 값을 -1로 설정하면 데이터베이스 관리 프로그램은 모든 여유 공간 제어 레코드를 검색해야 합니다.

표 6. 레지스트리 변수 (계속)

변수 이름	운영 체제	값
설명		
DLFM_TSM_MGMTCLASS	AIX, Windows NT, Solaris	기본값: 기본 TSM 관리 클래스 값: 유효한 TSM 관리 클래스
링크된 파일을 아카이브하고 검색하는 데 사용할 TSM 관리 클래스를 지정합니다. 이 변수에 값이 설정되지 않았으면, 기본 TSM 관리 클래스가 사용됩니다.		
DB2_CORRELATED_PREDICATES	모두	기본값=YES 값: YES 또는 NO
이 변수의 기본값은 YES입니다. 조인의 상관 컬럼에 고유 색인이 있으며 이 레지스트리 변수가 YES인 경우, 최적화 알고리즘은 조인 술어의 상관을 발견하여 보상하려고 시도합니다. 이 레지스트리 변수가 YES인 경우, 최적화 알고리즘은 고유 색인 통계의 KEYCARD 정보를 사용하여 상관 경우를 발견하고 상관된 술어들의 결합된 선택을 동적으로 조정하여 조인 크기와 비용의 보다 정확한 예상 값을 산출합니다.		
DB2_VI_DEVICE	Windows NT	기본값=널(NULL) 값: nic0 또는 VINIC
NIC(Network Interface Card)와 연관된 장치나 가상 인터페이스 제공자 인스턴스의 기호 이름을 지정합니다. 독립된 각 하드웨어 밴더(IHV)가 자체의 NIC를 작성합니다. Windows NT 머신 당 하나의 NIC만이 허용되며 동일한 물리적 머신에서 여러 논리 노드가 동일한 NIC를 공유합니다. 기호 장치 이름 『VINIC』는 대문자이어야 하며 Synfinity Interconnect와 함께만 사용할 수 있습니다. 현재 지원되는 다른 모든 구현은 기호 장치 이름으로 『nic0』을 사용합니다.		
DB2_SELECTIVITY	ALL	기본값=NO 값: YES 또는 NO
이 레지스트리 값은 SELECTIVITY 절이 사용될 수 있는 곳을 제어합니다. SELECTIVITY 절에 관한 완전한 세부사항은 SQL 참조, 언어 요소, 검색 조건을 보십시오. 이 레지스트리 값이 YES로 설정되면, 술어가 기본 술어이고 최소한 하나의 표현식에 호스트 변수가 포함되어 있을 때 SELECTIVITY 절을 지정할 수 있습니다.		
DB2_UPDATE_PART_KEY	ALL	기본값=YES 값: YES 또는 NO
FixPak 3 이후에서, 기본 값은 YES입니다. 이 레지스트리 변수는 파티션 키 갱신 허용 여부를 지정합니다.		
DB2_BLOCK_ON_LOG_DISK_FULL	ALL	기본값=NO 값: YES 또는 NO
DB2가 사용 중인 로그 경로에서 새 로그 파일을 작성할 수 없을 때 "디스크 가득참" 오류가 생성되는 것을 막기 위해 이 DB2 레지스트리 변수를 설정할 수 있습니다. 대신, DB2는 성공할 때까지 5분마다 로그 파일 작성을 시도합니다. 각 시도 후에, DB2는 db2diag.log 파일에 메시지를 씁니다. 로그 디스크 가득참 조건 때문에 응용프로그램이 정지되었는지 확인할 수 있는 유일한 방법은 db2diag.log 파일을 모니터링하는 것입니다. 로그 파일이 성공적으로 작성될 때까지, 테이블 데이터 갱신을 시도하는 사용자 응용프로그램은 트랜잭션을 확약할 수 없습니다. 읽기 전용 조회는 직접적으로 영향을 받지 않을 수 있습니다. 하지만, 조회가 갱신 요청에 의해 잠긴 데이터에 액세스해야 하거나, 응용프로그램 갱신에 의해 버퍼 풀에 고정된 데이터 페이지에 액세스해야 할 경우, 읽기 전용 조회 또한 정지된 것으로 보입니다.		
DB2_INDEX_2BYTEVARLEN	모두	기본값=NO 값: YES 또는 NO

표 6. 레지스트리 변수 (계속)

변수 이름	운영 체제	값
설명		
<p>이 레지스트리 변수는 길이가 255 바이트 이상인 컬럼을 색인 키의 일부로 지정할 수 있도록 합니다. 이 레지스트리 변수를 YES로 설정하기 전에 이미 작성된 색인은 계속 255 키 제한이 있습니다. 이 레지스트리 변수를 YES로 설정한 후에 작성된 색인은 레지스트리 변수가 다시 NO로 설정되더라도 2 바이트 색인으로서 가능합니다.</p> <p>이 레지스트리 변수를 변경하면, CREATE TABLE, CREATE INDEX 및 ALTER TABLE을 포함하여 여러 SQL문이 영향을 받습니다. 이러한 명령문에 대한 자세한 정보는, <i>SQL 참조서</i>에 대해 문서화된 변경사항을 참조하십시오.</p>		
DB2_FORCE_FCM_BP	AIX	기본값=NO 값: YES 또는 NO
<p>빠른 통신 관리 프로그램(FCM) 자원이 할당되었는지 지정합니다. 데이터베이스 관리 프로그램 공유 메모리 세그먼트 또는 별도 세그먼트로부터 자원이 할당될 수 있습니다. 같은 머신에 여러 논리적 노드가 있으면, 이 레지스트리 변수를 사용해야 합니다. 대칭 다중 처리(SMP)를 사용하는 파티션된 데이터베이스 시스템에서, 이 레지스트리 변수의 설정은 통신이 발생하는 방법에 아무 영향이 없습니다. 이 경우, 통신은 항상 공유 메모리를 통합합니다. 하지만, DB2가 사용할 공유 메모리 세그먼트의 수에 영향을 줍니다.</p>		
DB2_AWE	Windows 2000	기본값=NULL 값: <항목>[; <항목>;...] 여기서 <항목>=<버퍼 풀 ID>, <물리적 페이지 수>, <주소 창 의 수>
<p>Windows 2000의 DB2 UDB가 최대 64 GB 메모리를 사용하는 버퍼 풀을 할당하도록 허용합니다. AWE(Address Windowing Extensions) 버퍼 풀을 지원하려면 Windows 2000이 올바르게 구성되어야 합니다. 여기에는 『lock pages in memory』-right를 Windows 2000의 사용자와 연결하는 것과 DB2에서 이 레지스트리 변수를 설정하는 것이 포함됩니다. 이 변수 설정에서 AWE 지원에 사용될 버퍼 풀 ID를 알아야 합니다. 또한 할당할 물리적 페이지 수 및 주소 창 의 수를 알아야 합니다.</p> <p>할당할 물리적 페이지의 수 및 주소 창 의 수 편별에 대한 정보는, 이 절 앞부분의 『조작 성능』에서 『데이터베이스 버퍼 풀 관리』를 참조하십시오.</p> <p>주: AWE 지원을 사용할 경우, 확장 저장영역(ESTORE)을 데이터베이스의 버퍼 풀에 사용할 수 없습니다. 이 변수에 의해 참조되는 버퍼 풀은 이미 SYSIBM.SYSBUFFERPOOLS에 존재해야 합니다.</p>		
DB2_STPROC_LOCKUP_FIRST	모두	기본값=NO 값: YES 또는 NO
<p>이 레지스트리 변수는 DB2_DARI_LOOKUP_ALL로부터 이름이 변경되었습니다.</p>		
DB2MEMDISCLAIM	AIX	기본값=YES 값: YES 또는 NO
<p>AIX에서, DB2가 사용하는 메모리에는 연관된 페이징 공간이 있습니다. 이 페이징 공간은 연관된 메모리가 사용 가능하게 되더라도 예약된 상태로 남아 있습니다. 페이징 공간의 예약은 AIX 시스템의 조정 가능 가상 메모리 관리 할당 규정에 따라 다릅니다. 이 레지스트리 변수는 DB2 에이전트가 AIX가 사용 가능한 메모리로부터 예약된 페이징 공간의 연관을 해제하도록 명시적으로 요청하는 지 여부를 제어합니다.</p> <p>『YES』를 설정하면 페이징 공간 요구가 작아지고, 페이징으로부터 디스크 활동이 적습니다. 『NO』를 설정하면 페이징 공간 요구가 더 크고, 페이징으로부터 디스크 활동이 더 많습니다. 페이징 공간이 풍부하고, 실제 메모리가 풍부하여 페이징이 결코 발생하지 않는 경우와 같은 상황에서는, NO를 설정하면 작은 성능 개선이 있습니다.</p>		
DB2MEMMAXFREE	모두	기본값=8 388 608 바이트 값: 0에서 2 ³² -1 바이트
<p>이 레지스트리 변수는 DB2 프로세스가 보유하는 바이트 단위 미사용 메모리의 최대량을 제어합니다.</p>		

표 6. 레지스트리 변수 (계속)

변수 이름 설명	운영 체제	값
DB2_ANTIJOIN	모두	EEE 환경에서 기본값=NO 비 EEE 환경에서 기본값=YES 값: YES 또는 NO
DB2 Universal Database EEE 환경의 경우 YES를 지정하면, 최적화 알고리즘은 NOT EXISTS 부속조회를 DB2에 의해 보다 효율적으로 처리할 수 있는 안티조인으로 변형할 기회를 검색합니다. 비 EEE 환경의 경우 NO를 지정하면, 최적화 알고리즘은 NOT EXISTS 부속조회를 안티조인으로 변형하는 기회를 제한합니다.		
NEWLOGPATH2	UNIX	기본값=NO 값: YES 또는 NO
이 매개변수는 이중 로그를 구현하기 위해 2차 경로가 사용되어야 하는지 여부를 지정하도록 합니다. 사용될 경로는 'LOGPATH'의 현재 경로에 문자 '2'를 추가하여 생성됩니다.		
DB2DOMAINLIST	Windows NT	기본값=NULL 값: 하나 이상의 유효한 Windows NT 도메인(각각 쉼표로 분리)
하나 이상의 Windows NT 도메인을 정의합니다. 이들 도메인에 속한 사용자만 연결되거나 접속 요청이 승인됩니다. 이 레지스트리 변수는 DB2 Universal Database 버전 7.1(또는 그 이상)을 수행하는 DB2 서버와 클라이언트가 있는 순수한 Windows NT 도메인 환경 하에서만 사용되어야 합니다.		
DB2_LIKE_VARCHAR	모두	기본값=Y,N 값: Y, N, S, 0에서 6.2 사이의 부동 소수점 상수

컬렉션 및 부속요소 통계를 제어합니다. 이들은 데이터가 공백으로 분리되는 서브필드 또는 부속 구성요소 시리즈 양식으로 된 구조인 경우 컬럼에 있는 데이터 내용에 대한 통계입니다.

이 레지스트리 변수는 최적화 알고리즘이 다음 양식의 술어를 처리하는 방법에 영향을 줍니다.

COLUMN LIKE '%xxxxxx%'

여기서 xxxxxx는 임의의 문자열입니다.

다음은 이 레지스트리 변수가 사용되는 방법을 보여주는 구문입니다.

db2set DB2_LIKE_VARCHAR=[Y|N|S|num1] [,Y|N|S|num2]

여기서

- 쉼표 앞의 항목 또는 술어 오른쪽의 유일한 항목은 양의 부속요소 통계가 없는 컬럼에 대해서만 다음을 의미합니다.
 - S - 최적화 알고리즘이 % 문자 사이의 문자열의 길이에 기반하여 함께 연결되어 컬럼을 형성하는 일련의 요소들에서 각 요소의 길이를 평가합니다.
 - Y - 기본값. 알고리즘 매개변수에 기본값 1.9를 사용합니다. 알고리즘 매개변수와 함께 가변 길이 부속 요소 알고리즘을 사용합니다.
 - N - 고정 길이 부속 요소 알고리즘을 사용합니다.
 - num1 - 가변 길이 부속 요소 알고리즘과 num1 값을 알고리즘 매개변수로 사용합니다.
- 쉼표 뒤의 항목은 다음을 의미합니다.
 - N - 기본값. 사용 부속 요소 통계를 수집하지 않습니다.
 - Y - 부속 요소 통계를 수집합니다. 양의 부속 요소 통계가 있는 컬럼의 경우 알고리즘 매개변수에 1.9 기본값과 함께 수집된 통계를 사용하는 가변 길이 부속 요소 알고리즘을 사용합니다.
 - num2 - 부속 요소 통계를 수집합니다. 양의 부속 요소 통계가 있는 컬럼의 경우 알고리즘 매개변수인 num2 값과 함께 수집된 통계를 사용하는 가변 길이 부속 요소 알고리즘을 사용합니다.

DB2_PINNED_BP	AIX, HP-UX	기본값=NO 값: YES 또는 NO
<p>이 변수는 일부 AIX 운영 체제에서 메인 메모리의 데이터베이스와 연관된 데이터베이스 전역 메모리(버퍼 풀 포함)를 보유하기 위해 사용됩니다. 이 데이터베이스 전역 메모리를 시스템 메인 메모리에 보존하면 데이터베이스 성능이 보다 일관됩니다.</p> <p>예를 들어, 버퍼 풀을 시스템 메인 메모리에서 스왑아웃하면, 데이터베이스 성능이 저하됩니다. 버퍼 풀을 시스템 메모리에 두어 디스크 I/O를 줄이면 데이터베이스 성능을 향상시킵니다. 메인 메모리를 더 많이 요구하는 기타 응용프로그램이 있을 경우, 시스템 메인 메모리 요구사항에 따라 데이터베이스 전역 메모리가 메인 메모리에서 스왑아웃되도록 허용하려 할 것입니다.</p> <p>64 비트 환경에서 HP-UX로 작업시, 이 레지스트리 변수 수정과 더불어, DB2 인스턴스 그룹에 MLOCK 특권이 부여되어야 합니다. 이는 루트 액세스 권한이 있는 사용자가 다음을 수행함으로써 이루어집니다.</p> <p>1. DB2 인스턴스 그룹을 /etc/privgroup 파일에 추가하십시오. 예를 들어, DB2 인스턴스 그룹이 db2iadm1 그룹에 속한다면, 다음 행을 /etc/privgroup 파일에 추가해야 합니다.</p> <p>db2iadm1 MLOCK</p> <p>2. 다음 명령을 실행하십시오.</p> <p>setprivgrp -f /etc/privgroup</p>		
DB2_RR_TO_RS	모두	기본값=NO 값: YES 또는 NO

다음 키 잠금은 모든 INSERT 및 DELETE 문에 대한 다음 키와 SELECT 문의 결과 세트보다 더 높은 다음 키 값을 자동으로 잠궈서 반복 읽기(RR)를 보장합니다. 색인의 키 부분을 변경하는 UPDATE 문의 경우 원래 색인 키가 삭제되고 새 키 값이 삽입됩니다. 다음 키 잠금은 키 삽입과 키 삭제 시 모두 수행됩니다. 다음 키 잠금은 ANSI 및 SQL92 표준 RR을 보장해야 하며 이는 DB2 기본값입니다.

응용프로그램이 중지된 경우, 응용프로그램의 스냅샷 정보를 검사해야 합니다. 다음 키 잠금에 문제가 있으면 두 가지 조건을 기준으로 DB2_RR_TO_RS 레지스트리 변수를 설정할 수 있습니다. 반복 읽기(RR) 동작에 의존하는 응용프로그램이 없고 검색에서 확장되지 않은 삭제를 건너뛸 수 있는 경우, DB2_RR_TO_RS를 설정할 수 있습니다. 건너뛰기 동작은 RR, 읽기 안정성(RS) 및 커서 안정성(CS) 분리 레벨에 영향을 줍니다. 미확약 읽기(UR) 분리 레벨에 대한 행 잠금은 없습니다.

DB2_RR_TO_RS가 설정되면 다음 키 잠금이 색인 키 삽입 및 삭제 시 수행되지 않으므로 사용자 테이블에서 검색에 대한 RR 동작을 보장할 수 없습니다. 카탈로그 테이블은 이 옵션의 영향을 받지 않습니다.

동작의 다른 변경은 DB2_RR_TO_RS를 설정하는 것인데, 행이 검색 규정에 맞지도 삭제되고 확장되지 않은 행을 검색할 경우에는 건너뛸니다.

SQL Explain 도구

제목이 『db2expln 및 dynexpln 수행』인 절에서 마지막 문단을 다음의 내용으로 바꿔야 합니다.

db2expln을 수행하려면, db2expln 패키지에 대한 EXECUTE 권한과 함께 시스템 카탈로그 뷰에 대해 SELECT 특권이 있어야 합니다. dynexpln을 수행하려면, 데이터베이스에 대해 BINDADD 권한을 가지고 있어야 하며 데이터베이스에 연결하는데 사용하는 스키마가 존재해야 하거나 데이터베이스에 대한 EXPLICIT_SCHEMA 권한이 있어야 하고 설명되는 SQL문에 필요한 특권이 있어야 합니다(SYSADM 또는 DBADM 권한이 있는 경우, 이러한 모든 권한 부여 레벨을 자동으로 가지게 됩니다).

Administering Satellites 안내 및 참조서

버전 7.2 DB2 Personal Edition 및 DB2 Workgroup Edition을 위성으로 설정

다음 절에서는 Windows 기반 버전 7.2 DB2 Personal Edition 및 DB2 Workgroup Edition 시스템을 위성 환경에서 모든 기능을 발휘하는 위성으로 사용하도록 설정하는 방법을 설명합니다. 이 설명에서 사용되는 용어 및 개념에 대한 자세한 정보는 *Administering Satellites* 안내 및 참조서에서 참조하십시오. 다음 URL에 이 책이 있습니다.

http://www-4.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/v6pubs.d2w/en_main

Administering Satellites 안내 및 참조서의 정보를 보충하는 Technotes는 다음 URL을 참조하십시오.

<http://www-4.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/browse.d2w/report?type=tech5udb&tech5udb=Y>

전제조건

DB2 Personal Edition이나 DB2 Workgroup Edition을 위성으로 설정하려면 다음이 필요합니다.

1. DB2 제어 서버

DB2 제어 서버는 Windows NT 또는 AIX에서 실행되는 DB2 Enterprise Edition 시스템으로 Control Server 구성요소가 설치되어 있습니다. 사용하는 DB2 Enterprise Edition 시스템은 FixPak 2 이상이 있는 버전 6이거나 FixPak이 있는 버전 7이어야 합니다.

- DB2 제어 서버로 사용하려는 버전 6 Enterprise Edition 시스템이 있는 경우, 129 페이지의 『버전 6 Enterprise Edition 시스템에 FixPak 2 이상 설치』에서 참조하십시오.
- 버전 7을 사용하며 제어 서버 구성요소가 설치되어 있지 않은 경우, 이 구성요소를 설치하고 이미 설치되어 있는 FixPak을 다시 설치한 후, DB2 제어 서버 인스턴스 및 위성 제어 데이터베이스를 작성하십시오. 이들 오브젝트 작성에 대한 지침은 *Administering Satellites* 안내 및 참조서에서 참조하십시오.

주: 버전 7.2 Enterprise Edition 시스템을 DB2 제어 서버로 사용하기 위해 Windows NT에 설치하고 응답 파일 설치를 수행하려는 경우, 응답 파일에 지정된 키워드에 대한 자세한 정보는 *DB2 제어 서버 응답 파일 키워드*라는 제목의 Technote를 참조하십시오.

2. DB2 제어 서버 인스턴스 및 위성 제어 데이터베이스

DB2 제어 서버 인스턴스는 일반적으로 DB2CTLSV라고 하며 위성 제어 데이터베이스는 SATCTLDB라고 합니다. DB2 제어 서버 인스턴스 및 위성 제어 데이터베이스는 Enterprise Edition 시스템에 있으며 Windows NT에서는 제어 서버 구성 요소와 함께 DB2를 설치할 때 자동으로 작성됩니다. AIX에 DB2를 설치할 경우, DB2 제어 서버 및 위성 제어 데이터베이스 작성에 대한 자세한 내용은 *Administering Satellites* 안내 및 참조서에서 참조하십시오.

3. 위성 관리 센터

위성 관리 센터는 위성 환경을 설정하고 관리할 때 사용하는 GUI 도구의 집합입니다. 제어 센터에서 이 도구 세트에 액세스합니다. 위성 관리 센터 및 위성 환경에 대한 자세한 정보는 *Administering Satellites* 안내 및 참조서 및 위성 관리 센터에서 제공되는 온라인 도움말을 참조하십시오. 버전 6 제어 센터를 수행하고 있는 경우, 137 페이지의 『버전 6 제어 센터 및 위성 관리 센터 업그레이드』에서 참조하십시오.

위성 환경을 설정하고 위성 관리 센터의 새 위성을 나타내는 오브젝트를 작성하기 위해 위성 관리 센터를 사용하지 않은 경우, 위성을 설치하기 전에 사용해야 합니다. 자세한 정보는 *Administering Satellites* 안내 및 참조서의 위성 환경 설정 및 테스트 방법에 대한 설명을 참조하십시오.

4. 위성으로 사용하려는 버전 7.2 Personal Edition 또는 Workgroup Edition 시스템

설치 고려사항

DB2 Personal Edition이나 DB2 Workgroup Edition을 설치할 경우, 시스템을 동기화 하기 위해 특별한 구성요소를 선택할 필요가 없습니다. 응답 파일 설치를 수행하는 경우 버전 7.2 시스템을 설치할 때 지정해야 하는 키워드에 대해서는 『응답 파일 설치 수행』을 참조하십시오. 버전 7.2 시스템의 대화식 설치를 수행하는 경우 버전 7.2 시스템을 동기화하기 위해 설정해야 하는 값에 대해서는 DB2를 설치한 후 128 페이지의 『동기화하기 위해 버전 7.2 시스템 구성』을 참조하십시오.

응답 파일 설치 수행: 버전 7.2 DB2 Personal Edition이나 DB2 Workgroup Edition의 응답 파일 설치를 수행하는 경우, 응답 파일에서 다음 키워드를 설정할 수 있습니다.

응답 파일 설치 시 이들 키워드 중 하나 이상을 지정하지 않을 경우, 버전 7.2 시스템을 동기화하기 위해 DB2를 설치한 후 수행해야 하는 추가 단계에 대해서는 128 페이지의 『동기화하기 위해 버전 7.2 시스템 구성』을 참조하십시오. 또한 응답 파일 설치 시 지정한 값을 변경하려면 이 절의 지침을 사용할 수 있습니다.

db2.db2satelliteid

시스템에서 위성 ID를 설정합니다.

주: 키워드를 지정하지 않으면 위성 ID가 DB2를 설치할 때 사용된 사용자 ID로 자동으로 설정됩니다. 사용자 ID를 위성 ID로 사용하려면 키워드 값을 지정할 필요가 없습니다.

db2.db2satelliteappver

시스템의 응용프로그램 버전을 설정합니다.

주: 키워드를 지정하지 않으면 위성의 응용프로그램 버전이 자동으로 V1R0M00으로 설정됩니다. 이 값을 응용프로그램 버전으로 사용하려면 키워드의 값을 지정할 필요가 없습니다.

db2.satctldb_username

위성 제어 데이터베이스에 연결하기 위해 시스템에서 사용할 사용자 이름을 설정합니다.

db2.satctldb_password

사용자 이름이 위성 제어 데이터베이스에 연결될 때 사용자 이름이 DB2 제어 서버로 전달할 암호를 설정합니다.

응답 파일 설치를 완료하면 버전 7.2 시스템은 동기화할 준비가 됩니다. 위성에서 **db2sync -t** 명령을 실행하여 위성에서 지정한 값이 맞는지 그리고 위성이 위성 제어 데이터베이스에 연결될 수 있는지 확인해야 합니다.

응답 파일 설치에 대한 추가 정보는 *Administering Satellites* 안내 및 참조서에서 참조하십시오.

주:

1. 버전 7에서 사용자 ID 및 암호는 Windows NT 및 Windows 2000에서 모든 서비스를 작성하기 위해 필요합니다. 사용자 ID 및 암호는 응답 파일에서 키워드 쌍으로 지정됩니다. 해당 서비스에 대해 특정 키워드 쌍을 지정하여 서비스를 대체하도록 지정하지 않는 한 응답 파일에서 발견되는 첫번째 키워드 쌍이 모든 서비스의 기본 사용자 ID 및 암호가 됩니다.

버전 6에서 **admin.userid** 및 **admin.password** 키워드는 Remote Command Service에서 사용될 사용자 ID 및 암호를 지정하기 위해 DB2 Satellite Edition의 응답 파일 설치 시 지정될 수 있습니다. 버전 7.2 Personal Edition 및 Workgroup Edition의 경우, 이들 키워드를 지정하면 버전 7.2 시스템의 DB2DAS00 인스턴스에 대해 사용됩니다. DB2 버전 7.2 시스템의 경우, Remote Command Service는 시스템의 DB2 인스턴스가 사용하는 사용자 ID 및 암호를 사용합니다. **db2.userid** 및 **db2.password**의 값을 지정하지 않으면 위에서 설명한 기본 규칙이 적용됩니다.

2. 버전 6에서는 응답 파일 설치를 사용하여 DB2 Satellite Edition을 설치할 때 데이터베이스를 작성할 수 있었습니다. 위성으로 사용하려는 버전 7.2 Personal Edition

이나 Workgroup Edition 시스템에서는 응답 파일 설치 시 데이터베이스를 작성할 수 없습니다. 다음 키워드(*Administering Satellites* 안내 및 참조서에서 설명)는 지원되지 않습니다.

- **db2.userdb_name**
- **db2.userdb_recoverable**
- **db2.userdb_rep_src**

동기화하기 위해 버전 7.2 시스템 구성

버전 7.2 시스템을 대화식으로 설치할 경우, DB2를 설치한 후 시스템을 동기화하기 전에 DB2 Personal Edition이나 DB2 Workgroup Edition 시스템에서 몇 가지 값을 설정해야 합니다.

주: 시스템에서 운영 체제 실행 스크립트를 사용하여 위성이 위성 제어 데이터베이스에 연결할 때 사용하는 사용자 ID와 암호를 제외한 모든 값을 위성에서 설정할 수 있습니다(129 페이지의 4 단계 참조).

1. **db2set** 명령을 사용하여 위성 ID를 설정합니다.

DB2 Personal Edition이나 DB2 Workgroup Edition을 대화식으로 설치하는 경우, 위성 ID는 자동으로 DB2를 설치할 때 사용된 사용자 ID로 설정됩니다. 사용자 ID를 위성 ID로 사용하려면 이 단계를 수행할 필요가 없습니다. 위성 ID 설정에 대한 자세한 정보는 *Administering Satellites* 안내 및 참조서에서 참조하십시오.

2. **db2sync -s** 명령을 사용하여 위성에서 응용프로그램 버전을 설정합니다.

DB2 Personal Edition이나 DB2 Workgroup Edition을 대화식으로 설치하는 경우, 위성의 응용프로그램 버전은 자동으로 V1R0M00으로 설정됩니다. 이 값을 응용프로그램 버전으로 사용하려면 이 단계를 수행할 필요가 없습니다.

위성에서 **db2sync -g** 명령을 수행하여 응용프로그램 버전의 현재 설정을 볼 수 있습니다. 이 값을 변경하려면 **db2sync -s** 명령을 실행하십시오. 응용프로그램 버전에 대해 새 값을 제공하라는 프롬프트가 표시됩니다. 응용프로그램 버전 설정에 대한 자세한 정보는 *Administering Satellites* 안내 및 참조서에서 참조하십시오.

3. **catalog node** 및 **catalog database** 명령을 위성에서 실행하여 위성에서 DB2 제어 서버 인스턴스 및 위성 제어 데이터베이스 SATCTLDB를 카탈로그화할 수 있습니다.

위성에서 **db2sync -t** 명령을 사용하여 테스트 모드로 DB2 Synchronizer 응용 프로그램을 열 수도 있습니다. 명령을 실행할 때 위성에서 SATCTLDB 데이터베이스의 카탈로그가 만들어지지 않으면 카탈로그 제어 데이터베이스 창이 열립니다. 카탈로그 제어 데이터베이스 창에서 제공되는 DB2 발견 기능을 사용하여 DB2 제어 서버 및 SATCTLDB 데이터베이스의 카탈로그를 만들거나 이 창에서 호스트 이름

및 서버 이름을 입력할 수 있습니다. 4 단계에서 설명한대로 위성이 위성 제어 데이터베이스에 연결할 때 사용할 사용자 ID 및 암호를 지정하라는 프롬프트도 나타납니다.

주: 버전 7.2 DB2 Personal Edition이나 DB2 Workgroup Edition을 대화식으로 설치한 후, DB2 Synchronizer는 테스트 모드로 자동으로 시작되지 않습니다 (버전 6 DB2 Satellite Edition의 경우와 동일).

4. 위성에서 **db2sync -t** 명령을 실행하여 다음을 수행합니다.

- 위성이 위성 제어 데이터베이스에 연결할 때 사용할 사용자 ID 및 암호를 지정합니다.

동기화 자격 증명이 아직 위성에 저장되지 않았으면 Connect to Control 데이터베이스 창이 열립니다. 이 창에서 위성이 위성 제어 데이터베이스에 연결할 때 사용할 사용자 ID 및 암호를 지정해야 합니다.

- 위성에 설정된 값이 올바른지 검증합니다.
- 위성을 위성 제어 데이터베이스에 연결할 수 있는지 검증합니다.

이 구성 작업을 완료하면 버전 7.2 시스템은 동기화할 준비가 됩니다.

버전 6 Enterprise Edition 시스템에 FixPak 2 이상 설치

다음 절에서는 DB2 제어 서버로 사용하기 위해 Windows NT 또는 AIX에서 버전 6 Enterprise Edition 시스템을 업그레이드할 때 수행해야 하는 작업에 대해 설명합니다. 버전 6 제어 센터를 사용하고 있는 경우에는 137 페이지의 『버전 6 제어 센터 및 위성 관리 센터 업그레이드』의 단계도 수행하여 위성 환경을 관리하는 올바른 레벨의 제어 센터 및 위성 관리 센터가 있음을 검증하십시오.

DB2 제어 서버로 사용하기 위해 버전 6 DB2 Enterprise Edition 업그레이드

버전 6 DB2 Enterprise Edition 시스템을 DB2 제어 서버로 사용하려면 제어 서버 구성요소와 함께 설치되어야 하며 DB2 Enterprise Edition의 레벨이 FixPak 2 서비스 레벨 이상이어야 합니다. DB2 제어 서버 구성요소가 설치되어 있는지 여부에 따라 그리고 DB2 Enterprise Edition의 서비스 레벨에 따라 다음 작업 중 하나를 수행해야 합니다.

- DB2 제어 서버 구성요소를 기존 DB2 Enterprise Edition V6.1 시스템에 설치하고 FixPak 2 이상을 설치합니다. 그런 다음 시스템의 위성 제어 데이터베이스 (SATCTLDB)를 갱신합니다.
- 이미 설치된 DB2 제어 서버를 FixPak 2 레벨 이상으로 업그레이드합니다.

다음 정보를 참조하여 앞의 두 작업 중 수행해야 할 작업, 그리고 상황에 적용해야 할 단계를 식별하십시오. 다음은 수행해야 할 단계를 요약한 것입니다.

1. 먼저 현재의 DB2 Enterprise Edition 설치 상태를 살펴봅니다. 제어 서버 구성요소 및 DB2 서비스 레벨이 설치되어 있는지 파악합니다.
2. 두 번째로 파악한 상태 정보에 따라 수행할 작업을 판단합니다.
3. 세 번째로 DB2 Enterprise Edition을 업그레이드하기 위해 필요한 단계를 수행합니다.

DB2 제어 서버는 Windows NT 및 AIX용 DB2 Enterprise Edition에서만 실행될 수 있습니다. 플랫폼에 적합한 지침에 따라 계속하십시오.

- 『Windows NT에서 DB2 Enterprise Edition 업그레이드』
- 134 페이지의 『AIX에서 DB2 Enterprise Edition 업그레이드』

Windows NT에서 DB2 Enterprise Edition 업그레이드: 다음 절의 정보를 참조하여 버전 6 DB2 Enterprise Edition 시스템의 현재 서비스 레벨 및 시스템을 FixPak 2 서비스 레벨 이상으로 갱신하기 위해 수행해야 할 단계를 파악합니다. 다음 절 중 하나 이상의 단계를 수행해야 합니다.

- 『Windows NT의 DB2 Enterprise Edition 평가』
- 131 페이지의 『수행할 조치 판단』
- 131 페이지의 『Windows NT에 제어 서버 구성요소 설치』
- 132 페이지의 『Windows NT에 FixPak 2 이상 설치』
- 133 페이지의 『Windows NT에서 SATCTLDB 업그레이드』

Windows NT의 DB2 Enterprise Edition 평가: DB2 Enterprise Edition이 Windows NT에 설치되어 있는 경우, 다음 단계를 수행하십시오.

1. 제어 서버 구성요소가 설치되어 있는지 확인합니다. Registry Editor를 사용하여 설치된 구성요소 목록을 표시합니다.
 - a. 명령 프롬프트에서 regedit를 입력합니다.
 - b. HKEY_LOCAL_MACHINE\SOFTWARE\IBM\DB2\Components registry key에서 제어 서버가 목록에 있는지 확인합니다. 목록에 없으면 제어 서버가 설치되지 않은 것입니다.
2. DB2 Enterprise Edition의 서비스 레벨을 파악합니다. 명령 프롬프트에서 db2level 명령을 실행합니다. 다음 테이블을 참조하여 출력 내용을 해석합니다.

db2level 출력의 키 필드 값			사용자 DB2 시스템
릴리스	레벨	정보용 토큰	
SQL06010	01010104	db2_v6, n990616	버전 6.1 기본
SQL06010	01020104	DB2 V6.1.0.1, n990824, WR21136	버전 6.1 plus FixPak 1
SQL06010	01030104	DB2 V6.1.0.6, s991030, WR21163 또는 DB2 V6.1.0.9, s000101, WR21173	버전 6.1 plus FixPak 2

주: 레벨이 01030104 이상인 경우, 사용자 시스템은 FixPak 2 이상의 FixPak 레벨입니다.

3. 찾은 정보를 기록한 후 『수행할 조치 판단』을 계속합니다.

수행할 조치 판단: 수집한 정보를 참조하여 다음 테이블에서 상황에 맞는 행을 찾아 FixPak 2 레벨 이상의 DB2 제어 서버를 지원하도록 DB2 Enterprise Edition 시스템을 준비하기 위해 필요한 단계를 수행합니다.

테이블 뒤의 절에서 필요한 단계를 수행하기 위한 지침을 제공합니다. 각 단계를 수행할 때마다 표시하십시오. 상황에 맞는 단계만 수행하십시오.

제어 서버 구성요소가 설치됨	DB2 Enterprise Edition 시스템의 서비스 레벨	DB2 Enterprise Edition 시스템을 준비하기 위해 필요한 단계
아니오	버전 6.1 기본, 또는 버전 6.1 plus FixPak 1, 또는 버전 6.1 plus FixPak 2 이상	다음 단계를 수행하십시오. 1. 『Windows NT에 제어 서버 구성요소 설치』 2. 132 페이지의 『Windows NT에 FixPak 2 이상 설치』 3. 133 페이지의 『Windows NT에서 SATCTLDB 업그레이드』
예	버전 6.1 기본 또는 버전 6.1 plus FixPak 1	다음 단계를 수행하십시오. 1. 132 페이지의 『Windows NT에 FixPak 2 이상 설치』 2. 133 페이지의 『Windows NT에서 SATCTLDB 업그레이드』
예	버전 6.1, plus FixPak 2 이상	다음 단계를 수행하십시오. 1. 133 페이지의 『Windows NT에서 SATCTLDB 업그레이드』

Windows NT에 제어 서버 구성요소 설치: Windows NT에 제어 서버 구성요소를 설치하려면 다음을 수행하십시오.

1. 계속하기 전에 시스템의 모든 데이터베이스 활동이 완료되었는지 확인합니다.
2. DB2 Universal Database Enterprise Edition 버전 6.1 CD를 CD 드라이브에 넣으십시오.
설치 프로그램이 자동으로 시작하지 않으면 CD의 루트에서 setup 명령을 수행하여 설치를 시작합니다.
3. 프롬프트가 표시되면 DB2를 사용하는 모든 프로세스를 종료합니다.
4. 환영 창에서 다음을 선택합니다.
5. 제품 선택 창에서 DB2 Enterprise Edition이 선택되어 있는지 확인합니다.
6. 설치 유형 선택 패널에서 사용자 설치를 누릅니다.

7. 구성요소 선택 패널에서 제어 서버 구성요소가 선택되어 있는지 확인하고 다음을 누릅니다.

주: 시스템에 아직 설치되지 않은 다른 구성요소를 선택하면 이 구성요소도 설치됩니다. DB2가 설치된 드라이브나 디렉토리를 변경할 수 없습니다.

8. Configure DB2 Services 패널에서 프로토콜 값 및 제어 서버 인스턴스의 시작 옵션을 수정하거나 기본값을 사용할 수 있습니다. 기본값을 수정하고 다음을 누르거나 다음을 눌러 기본값을 사용할 수 있습니다.

9. 파일 복사 시작 창에서 다음을 눌러 설치를 시작합니다.

10. 파일 복사 프로세스가 완료되면 시스템을 재부트할 수 있는 옵션이 나타납니다. 이제 재부트해야 합니다. 제어 서버에 맞게 시스템을 변경한 내용은 시스템을 재부트할 때까지 적용되지 않습니다.

설치가 완료되고 시스템을 재부트했으면 제어 서버 설치 과정의 일부로 작성된 위성 제어 데이터베이스(SATCTLDB)를 DB2 인스턴스에서 카탈로그화해야 시스템에서 지역적으로 제어 센터 및 위성 관리 센터를 사용할 수 있습니다. SATCTLDB 데이터베이스를 카탈로그화하려면 다음을 수행하십시오.

1. 시작 → 프로그램 → **Windows NT용 DB2** → 명령 창을 선택하여 DB2 명령 창을 엽니다.

2. db2 인스턴스를 사용하고 있는지 확인합니다.

set 명령을 실행하고 db2instance 값을 확인합니다. 값이 db2가 아니면 다음 명령을 실행합니다.

```
set db2instance=db2
```

3. 다음 명령을 입력하여 db2ctlsv 인스턴스를 카탈로그화합니다.

```
db2 catalog local node db2ctlsv instance db2ctlsv
```

4. 다음 명령을 입력하여 SATCTLDB 데이터베이스를 카탈로그화합니다.

```
db2 catalog database satctldb at node db2ctlsv
```

5. 다음 명령을 입력하여 카탈로그 조치를 확약합니다.

```
db2 terminate
```

6. DB2 명령 창을 닫으십시오.

Windows NT에 FixPak 2 이상 설치: Windows NT에서 버전 6 DB2 Enterprise Edition 시스템을 FixPak 2 이상으로 업그레이드하려면 다음 중 하나를 수행하십시오.

- Windows NT용 DB2 Enterprise Edition V6.1의 최신 FixPak을 readme와 함께 웹에서 다운로드합니다. 지침에 따라 다음 URL에서 FixPak을 다운로드할 수 있습니다.

<http://www-4.ibm.com/software/data/db2/db2tech/version61.html>

readme.txt 파일의 지침에 따라 FixPak을 설치합니다.

- FixPak 2 레벨 이상의 Windows NT용 DB2 Universal Database, 버전 6.1 FixPak CD를 사용하여 CD의 WINNT95 디렉토리의 readme.txt 파일 지침에 따라 설치를 완료하십시오.

Windows NT에서 SATCTLDB 업그레이드: Windows NT에서 SATCTLDB 데이터베이스를 업그레이드하려면 다음을 수행하십시오.

1. SATCTLDB 데이터베이스의 레벨을 파악합니다.

- Windows NT 시스템에서 지역 관리자 권한을 가진 사용자 ID로 로그인합니다.
- 시작 → 프로그램 → Windows NT용 DB2 → 명령 창을 선택하여 DB2 명령 창을 엽니다.
- 다음 명령을 입력하여 SATCTLDB에 연결합니다.

```
db2 connect to satctldb
```

- 다음 조회를 실행하여 데이터베이스에 I_BATCHSTEP_TRGSCR 트리거가 있는지 파악합니다.

```
db2 select name from sysibm.systriggers where name='I_BATCHSTEP_TRGSCR'
```

리턴된 행 수를 기록합니다.

- 다음 명령을 입력하여 데이터베이스와의 연결을 끊습니다.

```
db2 connect reset
```

1d 단계에서 하나의 행을 리턴하면 데이터베이스 레벨이 정확한 것입니다. 이 경우 2 단계를 건너뛰고 134 페이지의 3 단계로 진행합니다. 0개의 행이 리턴되면 데이터베이스 레벨이 잘못된 것이므로 134 페이지의 3 단계를 수행하기 전에 2 단계에서 설명한대로 업그레이드해야 합니다.

2. SATCTLDB 데이터베이스를 업그레이드하려면 다음 단계를 수행하십시오. DB2 명령 창에서 모든 명령을 입력하십시오.

- <db2path>\misc 디렉토리로 변경합니다. 여기서 <db2path>는 설치 드라이브 및 경로입니다(예: c:\sql11b).
- db2ctlsv 인스턴스를 사용하고 있는지 확인합니다.

set 명령을 실행하고 db2instance 값을 확인합니다. 값이 db2ctlsv가 아니면 다음 명령을 실행합니다.

```
set db2instance=db2ctlsv
```

- 다음 명령을 입력하여 SATCTLDB 데이터베이스를 삭제합니다.

```
db2 drop database satctldb
```

- 다음 명령을 입력하여 새로운 SATCTLDB 데이터베이스를 작성합니다.

```
db2 -tf satctldb.dd1 -z satctldb.log
```

- 다음 명령을 실행하십시오.

```
db2 terminate
```

3. db2satcs.dll 저장 프로시저를 SATCTLDB 데이터베이스에 바인드합니다. 다음 단계를 수행하십시오.

a. 다음 명령을 입력하여 SATCTLDB 데이터베이스에 연결합니다.

```
db2 connect to satctldb
```

b. <db2path>\bnd 디렉토리로 변경합니다. 여기서 <db2path>는 설치 드라이브 및 경로입니다(예: c:\sql1lib).

c. 다음과 같이 bind 명령을 실행합니다.

```
db2 bind db2satcs.bnd
```

4. 다음 명령을 입력하여 데이터베이스와의 연결을 끊습니다.

```
db2 connect reset
```

5. DB2 명령 창을 닫으십시오.

AIX에서 DB2 Enterprise Edition 업그레이드: 다음 절의 정보를 참조하여 버전 6 DB2 Enterprise Edition 시스템의 현재 서비스 레벨 및 시스템을 FixPak 2 서비스 레벨 이상으로 갱신하기 위해 수행해야 할 단계를 파악합니다. 다음 절 중 하나 이상의 단계를 수행해야 합니다.

- 『AIX에서 DB2 Enterprise Edition 평가』
- 135 페이지의 『수행할 조치 판단』
- 135 페이지의 『AIX에 제어 서버 구성요소 설치』
- 136 페이지의 『AIX에 FixPak 2 이상 설치』
- 136 페이지의 『AIX에서 SATCTLDB 업그레이드』

AIX에서 DB2 Enterprise Edition 평가: 버전 6 DB2 Enterprise Edition이 AIX에 설치되어 있는 경우, 다음 단계를 수행하십시오.

1. 제어 서버 구성요소가 설치되어 있는지 확인합니다. 다음 명령을 입력하십시오.

```
ls1pp -1 | grep db2_06_01.ctr
```

리턴된 데이터가 없으면 제어 서버 구성요소가 설치되지 않은 것입니다.

2. DB2 Enterprise Edition의 서비스 레벨을 파악합니다. DB2 인스턴스 소유자로 로그인하고 db2level 명령을 실행합니다. 다음 표를 참조하여 출력 내용을 해석합니다.

db2level 출력의 키 필드 값			사용자 DB2 시스템
릴리스	레벨	정보용 토큰	
SQL06010	01010104	db2_v6, n990616	버전 6.1 기본
SQL06010	01020104	DB2 V6.1.0.1, n990824, U465423	버전 6.1 plus FixPak 1
SQL06010	01030104	DB2 V6.1.0.6, s991030, U468276 또는 DB2 V6.1.0.9, s000101, U469453	버전 6.1 plus FixPak 2

주: 레벨이 01030104 이상인 경우, 사용자 시스템은 FixPak 2 이상의 FixPak 레벨입니다.

3. 찾은 정보를 기록한 후 『수행할 조치 판단』을 계속합니다.

수행할 조치 판단: 수집한 정보를 참조하여 다음 테이블에서 상황에 맞는 행을 찾아 FixPak 2 레벨의 DB2 제어 서버를 지원하도록 DB2 Enterprise Edition 시스템을 준비하기 위해 필요한 단계를 수행합니다.

테이블 뒤의 절에서 필요한 단계를 수행하기 위한 지침을 제공합니다. 각 단계를 수행할 때마다 표시하십시오. 상황에 맞는 단계만 수행하십시오.

제어 서버 구성요소가 설치됨	DB2 Enterprise Edition 시스템의 서비스 레벨	DB2 Enterprise Edition 시스템을 준비하기 위해 필요한 단계
아니오	버전 6.1 기본, 또는 버전 6.1 plus FixPak 1, 또는 버전 6.1 plus FixPak 2 이상	다음 단계를 수행하십시오. 1. 『AIX에 제어 서버 구성요소 설치』 2. 136 페이지의 『AIX에 FixPak 2 이상 설치』 3. 136 페이지의 『AIX에서 SATCTLDB 업그레이드』
예	버전 6.1 기본 또는 버전 6.1 plus FixPak 1	다음 단계를 수행하십시오. 1. 136 페이지의 『AIX에 FixPak 2 이상 설치』 2. 136 페이지의 『AIX에서 SATCTLDB 업그레이드』
예	버전 6.1, plus FixPak 2 이상	다음 단계를 수행하십시오. 1. 136 페이지의 『AIX에서 SATCTLDB 업그레이드』

AIX에 제어 서버 구성요소 설치: AIX에 제어 서버 구성요소를 설치하려면 다음을 수행하십시오.

1. 루트 권한이 있는 사용자로 로그인합니다.
2. DB2 Universal Database Enterprise Edition 버전 6.1 CD를 CD 드라이브에 넣으십시오.
3. CD가 마운트된 디렉토리로 변경합니다(예: cd /cdrom).
4. 다음 명령을 입력하여 DB2 설치 프로그램을 시작합니다.
./db2setup
5. DB2 설치 프로그램 창이 열리면 탭 키를 사용하여 설치 옵션을 선택하고 Enter를 누릅니다.
6. Enterprise Edition 계열을 찾아 탭 키를 사용하여 옆의 사용자 정의 옵션을 선택합니다. Enter를 누르십시오.

7. DB2 제어 서버 구성요소를 선택하고 탭 키를 사용하여 확인을 선택한 후 Enter를 누릅니다.
8. 남은 창의 지침에 따라 DB2 제어 서버 구성요소의 설치를 완료합니다.

설치가 완료되면 DB2CTLSV 인스턴스 및 SATCTLDB 데이터베이스를 작성합니다. 이 작업을 수행하려면 *Administering Satellites* 안내 및 참조서 "AIX에서 DB2 제어 서버 설정"의 자세한 지침을 따르십시오.

AIX에 FixPak 2 이상 설치: 기존 DB2 Enterprise Edition 시스템 AIX를 FixPak 2 이상으로 업그레이드하려면 다음 중 하나를 수행하십시오.

- AIX용 DB2 Enterprise Edition V6.1의 최신 FixPak을 FixPak Readme와 함께 웹에서 다운로드합니다. 지침에 따라 다음 URL에서 FixPak을 다운로드할 수 있습니다.

<http://www-4.ibm.com/software/data/db2/db2tech/version61.html>

FixPak readme 파일의 지침에 따라 FixPak을 설치합니다.

- FixPak 2 레벨 이상의 AIX용 DB2 Universal Database, 버전 6.1 FixPak CD를 사용하여 CD의 readme 디렉토리의 지침에 따라 설치를 완료하십시오.

FixPak readme 파일의 지침에 따라 db2iupdt 명령을 실행하여 DB2CTLSV 인스턴스를 갱신했는지 확인합니다.

AIX에서 SATCTLDB 업그레이드: AIX에서 SATCTLDB 데이터베이스를 업그레이드하려면 다음을 수행하십시오.

1. SATCTLDB 데이터베이스의 레벨을 파악합니다.
 - a. db2ctlsv로 로그인합니다.
 - b. 데이터베이스 서버가 시작되었는지 확인합니다. 서버가 시작되지 않았으면 db2start 명령을 실행합니다.
 - c. 다음 명령을 입력하여 SATCTLDB 데이터베이스에 연결합니다.


```
db2 connect to satctldb
```
 - d. 다음 조회를 실행하여 데이터베이스에 I_BATCHSTEP_TRGSCR 트리거가 있는지 파악합니다.


```
db2 "select name from sysibm.systriggers where name='I_BATCHSTEP_TRGSCR'"
```

 리턴된 행 수를 기록합니다.
 - e. 다음 명령을 입력하여 데이터베이스와의 연결을 끊습니다.


```
db2 connect reset
```

1d 단계에서 하나의 행을 리턴하면 데이터베이스 레벨이 정확한 것입니다. 이 경우 2 단계를 건너뛰고 137 페이지의 3 단계로 진행합니다. 0개의 행이 리턴되

면 데이터베이스 레벨이 잘못된 것이므로 3 단계를 수행하기 전에 2 단계에서 설명한대로 업그레이드해야 합니다.

2. SATCTLDB 데이터베이스를 FixPak 2 레벨로 업그레이드하려면 다음 단계를 수행하십시오. DB2 명령 창에서 모든 명령을 입력하십시오.

a. \$HOME/sql1lib/misc 디렉토리로 변경합니다.

b. 다음 명령을 입력하여 SATCTLDB 데이터베이스를 삭제합니다.

```
db2 drop database satctldb
```

c. 다음 명령을 입력하여 새로운 SATCTLDB 데이터베이스를 작성합니다.

```
db2 -tf satctldb.dd1 -z $HOME/satctldb.log
```

d. 다음 명령을 실행하십시오.

```
db2 terminate
```

3. db2satcs.dll 저장 프로시듀어를 SATCTLDB 데이터베이스에 바인드합니다. 다음 단계를 수행하십시오.

a. 다음 명령을 입력하여 SATCTLDB 데이터베이스에 연결합니다.

```
db2 connect to satctldb
```

b. \$HOME/sql1lib/bnd 디렉토리로 변경합니다.

c. 다음과 같이 bind 명령을 실행합니다.

```
db2 bind db2satcs.bnd
```

4. 다음 명령을 입력하여 데이터베이스와의 연결을 끊습니다.

```
db2 connect reset
```

버전 6 제어 센터 및 위성 관리 센터 업그레이드

버전 6 제어 센터 및 위성 관리 센터를 FixPak 2 이상으로 업그레이드된 버전 6 DB2 제어 서버 및 위성 제어 데이터베이스(SATCTLDB)와 함께 사용하려면 도구도 FixPak 2 이상으로 업그레이드해야 합니다.

제어 센터 및 위성 관리 센터가 DB2 제어 서버와 같은 시스템에서 실행되고 있으면 DB2 Enterprise Edition 시스템이 FixPak 2로 업그레이드될 때 함께 업그레이드된 것입니다. 그러나 이들 도구를 다른 시스템에서 실행하려면 이 시스템을 FixPak 2 레벨 이상으로 업그레이드해야 합니다.

이 시스템을 FixPak 2 이상으로 업그레이드하려면 다음을 수행하십시오.

- V6.1의 사용자 제품용 최신 FixPak을 readme와 함께 웹에서 다운로드합니다. 지침에 따라 다음 URL에서 FixPak을 다운로드할 수 있습니다.

```
http://www-4.ibm.com/software/data/db2/db2tech/version61.html
```

readme 파일의 지침에 따라 FixPak을 설치합니다.

- FixPak 2 레벨 이상의 사용하는 운영 체제용 DB2 Universal Database, 버전 6.1
FixPak CD를 사용하여 readme의 지침에 따라 설치를 완료하십시오.

Command Reference

db2batch - 벤치마크 도구

PERF_DETAIL 매개변수의 설명에서 마지막 문장은 다음과 같아야 합니다.

1 이상의 값은 DB2 버전 2 및 DB2 UDB 서버에서만 유효하며,
현재 호스트 머신에서는 지원되지 않습니다.

db2cap(새 명령)

db2cap - CLI/ODBC 정적 패키지 바인딩 도구

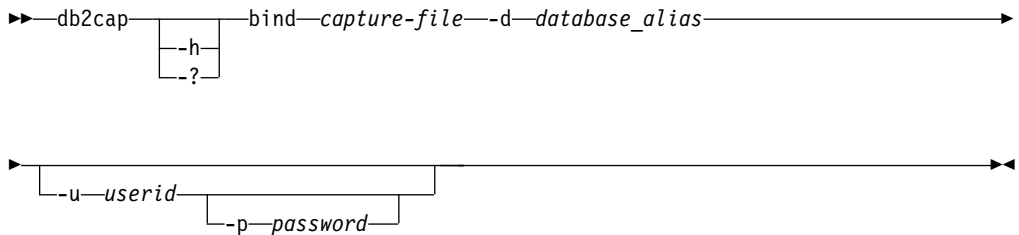
캡처 파일을 바인드하여 하나 이상의 정적 패키지를 생성하십시오. 캡처 파일은 CLI/ODBC/JDBC 응용프로그램의 정적 프로파일링 세션 중에 생성되며 응용프로그램이 수행하면서 캡처한 SQL문이 들어 있습니다. 이 유틸리티는 응용프로그램에 대한 정적 SQL을 실행할 때 CLI/ODBC/JDBC 드라이버가 사용할 수 있도록 캡처 파일을 처리합니다.

CLI/ODBC/JDBC 응용프로그램에서 정적 SQL의 사용법에 대해서는 *CLI Guide and Reference*에 있는 정적 프로파일링 기능을 참조하십시오.

권한 부여

- 캡처 파일에 기록된 SQL문이 참조한 데이터베이스 오브젝트에 대한 액세스 특권.
- **db2cap** 명령을 호출하는 데 사용되는 연결 ID와 다른 경우, OWNER 및 QUALIFIER와 같은 바인드 옵션을 설정하기에 충분한 권한.
- 패키지를 처음으로 바인드하는 경우에는 BINDADD 권한이 필요하며 그렇지 않은 경우에는 BIND 권한이 필요합니다.

명령 구문



명령 매개변수

-h/-? 명령 구문의 도움말 텍스트를 표시합니다.

bind capture-file

캡처 파일의 명령문들을 바인드하고 하나 이상의 패키지를 작성합니다.

-d database_alias

하나 이상의 패키지가 포함될 데이터베이스에 대한 데이터베이스 별명을 지정합니다.

-u userid

데이터 소스에 연결하는 데 사용할 사용자 ID를 지정합니다.

주: 사용자 ID가 지정되지 않은 경우, 신뢰하는 권한 부여 ID를 시스템으로부터 확보합니다.

-p password

데이터 소스에 연결하는 데 사용할 암호를 지정합니다.

사용 참고

명령은 UNIX 플랫폼에서 소문자로 입력해야 하지만 Windows 운영 체제와 OS/2에서는 소문자나 대문자로 입력할 수 있습니다.

유틸리티는 캡처 파일에서 찾을 수 있는 여러 사용자 지정 바인드 옵션을 지원합니다. 성능 및 보안 상의 이유로 텍스트 편집기를 사용하여 파일을 조사하고 편집하여 이들 옵션을 변경할 수 있습니다.

SQLERROR(CONTINUE) 및 VALIDATE(RUN) 바인드 옵션은 패키지를 작성하는데 사용할 수 있습니다.

이 유틸리티를 사용하여 패키지를 작성할 때, 정적 프로파일을 사용 불가능하게 해야 합니다.

작성되는 패키지 수는 캡처 파일에 기록된 SQL문에 사용된 분리 레벨에 따라 다릅니다. 패키지 이름은 캡처 파일의 패키지 키워드 중 처음 최대 7자로 구성되며, 다음 단일 문자 중의 하나가 접미부입니다.

- 0 - 미확약 읽기(UR)
- 1 - 커서 안정성(CS)
- 2 - 읽기 안정성(RS)
- 3 - 반복 읽기(RR)
- 4 - 확약 안함(NC)

패키지에 대한 특정 정보를 얻기 위해 사용자는 다음을 수행할 수 있습니다.

- 캡처 파일에 있는 COLLECTION 및 PACKAGE 키워드를 사용하여 해당 SYSIBM 카탈로그 테이블을 조회하십시오.
- 캡처 파일을 검토하십시오.

db2ckrst(새 명령)

db2ckrst - 증분 복원 이미지 순차 점검

데이터베이스 실행기록을 조회하고 증분 복원에 필요한 백업 이미지의 시간소인 목록을 생성합니다. 수동 증분 복원을 위한 간소화된 복원 구문 또한 생성됩니다.

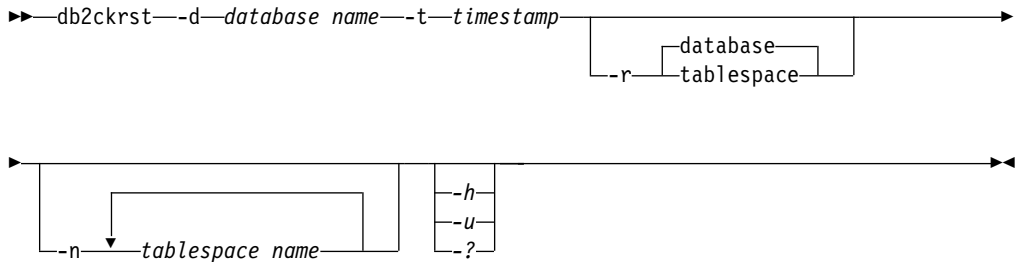
권한 부여

없음

필수 연결

없음

명령 구문



명령 매개변수

-d database namefile-name

복원될 데이터베이스의 별명 이름을 지정합니다.

-t timestamp

증분식으로 복원될 백업 이미지의 시간소인을 지정합니다.

-r 실행될 복원 유형을 지정합니다. 기본값은 database입니다.

주: tablespace를 선택하고 테이블 공간 이름을 제공하지 않으면, 유틸리티는 지정된 이미지의 실행기록 항목을 보고 나열된 테이블 공간 이름을 사용하여 복원을 수행합니다.

-n tablespace name

복원될 테이블 공간 이름을 하나 이상 지정합니다.

주: 데이터베이스 복원 유형을 선택하고 테이블 공간 이름의 목록을 지정하면, 유틸리티는 제공된 테이블 공간 이름을 사용하여 테이블 공간 복원을 계속합니다.

-h/-u/-?

도움말 정보를 표시합니다. 이 옵션이 지정된 경우, 다른 모든 옵션은 무시되며 도움말 정보만이 표시됩니다.

예

```
db2ckrst -d mr -t 20001015193455 -r database
db2ckrst -d mr -t 20001015193455 -r tablespace
db2ckrst -d mr -t 20001015193455 -r tablespace -n tbspl tbsp2

> db2 backup db mr

백업 성공. 이 백업 이미지의 시간소인: 20001016001426

> db2 backup db mr incremental

백업 성공. 이 백업 이미지의 시간소인: 20001016001445

> db2ckrst -d mr -t 20001016001445

데이터베이스 mr에 대한 시간소인 20001016001445를 사용한 이미지의 제안된 복원 순서.
=====
db2 restore db mr incremental taken at 20001016001445
db2 restore db mr incremental taken at 20001016001426
db2 restore db mr incremental taken at 20001016001445
=====

> db2ckrst -d mr -t 20001016001445 -r tablespace -n userspace1

데이터베이스 mr에 대한 시간소인 20001016001445를 사용한 이미지의 제안된 복원 순서.
=====
db2 restore db mr tablespace ( USERSPACE1 ) incremental taken at 20001016001445
db2 restore db mr tablespace ( USERSPACE1 ) incremental taken at 20001016001426
db2 restore db mr tablespace ( USERSPACE1 ) incremental taken at 20001016001445
=====
```

사용 참고

데이터베이스 실행기록은 이 유틸리티가 사용될 순서로 존재해야 합니다. 데이터베이스 실행기록이 존재하지 않으면, 이 유틸리티를 사용하기 전에 RESTORE 명령에 HISTORY FILE 옵션을 지정하십시오.

PRUNE HISTORY 명령의 FORCE 옵션을 사용할 경우, 가장 최근의 전체 데이터베이스 백업 이미지로부터 복구에 필요한 항목을 삭제하는 것이 가능합니다. PRUNE HISTORY 명령의 기본 조작용 필수 항목이 삭제되는 것을 방지하는 것입니다. PRUNE HISTORY 명령의 FORCE 옵션을 사용하지 않도록 권장합니다.

백업 기록을 잘 보존하고 이 유틸리티를 지침으로 사용하도록 권장합니다.

db2gncol(새 명령)

db2gncol - 생성된 컬럼 값 갱신

점검 보류 모드에 있으며 로그 공간이 제한된 테이블에서 생성된 컬럼을 갱신합니다. 이 도구는 표현식에 의해 생성된 컬럼이 있는 테이블에서 SET INTEGRITY 명령문을 위해 준비하는 데 사용됩니다.

권한 부여

다음 중 하나입니다.

- *sysadm*
- *dbadm*

명령 구문

```
▶▶db2gncol -d database -s schema_name -t table_name -c commit_count▶▶  
  
└─┬─userid─┬─p─password┬─h─┘
```

명령 매개변수

-d database

테이블이 위치하는 데이터베이스의 별명 이름을 지정합니다.

-s schema_name

테이블의 스키마 이름을 지정합니다. 스키마 이름은 대소문자를 구별합니다.

-t table_name

표현식에 의해 생성된 새로운 컬럼 값을 계산할 테이블을 지정합니다. 테이블 이름은 대소문자를 구별합니다.

-c commit_count

확약 간에 갱신된 행 수를 지정합니다. 이 매개변수는 컬럼 값을 생성하는 데 필요한 로그 공간의 크기에 영향을 줍니다.

-u userid

시스템 관리자 또는 데이터베이스 관리자 특권이 있는 사용자 ID를 지정합니다. 이 옵션이 생략된 경우, 현재 사용자로 가정합니다.

-p password

지정된 사용자 ID의 암호를 지정합니다.

-h 도움말 정보를 표시합니다. 이 옵션이 지정된 경우, 다른 모든 옵션은 무시되며 도움말 정보만이 표시됩니다.

사용 참고

SET INTEGRITY 명령문에서 FORCE GENERATED 옵션 대신 이 도구를 사용하는 것은 테이블이 크고 다음의 상태가 존재하는 경우에 필요합니다.

- 생성된 컬럼의 생성 표현식을 변경한 후에 모든 컬럼 값을 다시 생성해야 합니다.
- 생성된 컬럼에 사용된 외부 UDF가 변경되어 많은 컬럼 값들이 변경됩니다.
- 생성된 컬럼이 테이블에 추가되었습니다.
- 생성된 컬럼에 값을 제공하지 않은 대형 로드 또는 로드 추가가 수행되었습니다.
- 장기 수행 중인 동시 트랜잭션 또는 테이블의 크기로 인해 로그 공간이 너무 작습니다.

이 도구는 표현식을 기초로 작성된 모든 컬럼 값을 다시 생성합니다. 테이블이 갱신되는 동안 로그 공간을 모두 사용하는 것을 방지하기 위해 종종 확약을 수행합니다. **db2gncol**이 수행되었으면, 테이블은 SET INTEGRITY 명령문을 사용하여 점점 보류 모드에서 벗어날 수 있습니다.

db2inidb - 미러된 데이터베이스 초기화

분리된 미러 환경에서 이 명령은 다른 목적으로 미러링된 데이터베이스를 초기화합니다.

권한 부여

다음 중 하나이어야 합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*

필요한 연결

없음

명령 구문

```

▶▶—db2inidb—database_alias—AS—SNAPSHOT—
|—STANDBY—
|—MIRROR—

```

명령 매개변수

database_alias

초기화될 데이터베이스의 별명을 지정합니다.

SNAPSHOT

미러된 데이터베이스가 기본 데이터베이스의 복제로 초기화되도록 지정합니다. 이 데이터베이스는 읽기 전용입니다.

STANDBY

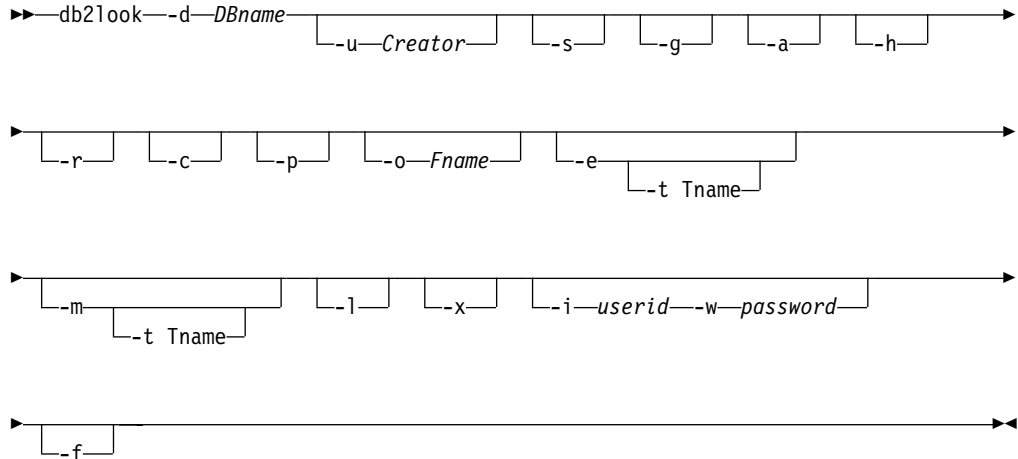
데이터베이스가 롤 포워드 보류 상태에 있도록 지정합니다. 기본 데이터베이스의 새 로그를 가져와 대기 데이터베이스에 적용합니다. 대기 데이터베이스는 기본 데이터베이스가 중단될 경우, 대신 사용할 수 있습니다.

MIRROR

미러된 데이터베이스를 기본 데이터베이스를 복원할 수 있는 백업 이미지로 사용할 수 있도록 지정합니다.

db2look - DB2 통계 추출 도구

구문 도표는 다음과 같이 표시되어야 합니다.



-td x 매개변수가 -c 매개변수 뒤에 추가되었습니다. 그 정의는 다음과 같습니다.

db2look에 의해 생성된 SQL문에 대한 명령문 분리문자를 지정합니다. 이 옵션을 지정하지 않으면, 기본값은 세미콜론 ';'입니다. -e 옵션이 지정되었을 경우 이 옵션을 사용하도록 권장합니다. 이 경우, 추출된 오브젝트에는 트리거 또는 SQL 루틴이 포함될 수 있습니다.

또한 다음 예가 추가되었습니다.

데이터베이스 DEPARTMENT의 모든 사용자에 의해 작성된 오브젝트에 대해 DDL문을 생성합니다. db2look 출력이 파일 db2look.sql에 전송됩니다.

```
db2look -d department -a -e -td % -o db2look.sql  
db2 -td% -f db2look.sql
```

db2updv7 - 버전 7 현재 수정 레벨로 데이터베이스 갱신

이 명령은 데이터베이스의 시스템 카탈로그가 현재 FixPak을 지원하도록 다음 방법으로 갱신합니다.

- 새로운 내장 함수(ABS, DECRYPT_BIN, DECRYPT_CHAR, ENCRYPT, GETHINT, MULTIPLY_ALT 및 ROUND)의 사용을 가능하게 합니다.
- 새로운 내장 프로시저(GET_ROUTINE_SAR 및 PUT_ROUTINE_SAR)의 사용을 가능하게 합니다.

- Windows 및 OS/2 데이터베이스에서 WEEK_ISO 및 DAYOFWEEK_ISO 함수에 정정사항을 추가 또는 적용합니다.
- 버전 2에서 버전 6으로 이주한 테이블에 대한 테이블 팩 설명자에 정정을 적용합니다.
- 뷰 SYSCAT.SEQUENCES를 작성합니다.

권한 부여

sysadm

필요한 연결

데이터베이스. 이 명령은 지정된 데이터베이스에 자동으로 연결을 설정합니다.

명령 구문

```

▶▶ db2upd7 --d database_name -u userid -p password

```

-h

명령 매개변수

-d database-name

갱신될 데이터베이스의 이름을 지정합니다.

-u userid

사용자 ID를 지정합니다.

-p password

사용자의 암호를 지정합니다.

-h 도움말 정보를 표시합니다. 이 옵션이 지정된 경우, 다른 모든 옵션은 무시되며 도움말 정보만이 표시됩니다.

예 FixPak 설치 후, 다음 명령을 사용하여 샘플 데이터베이스의 시스템 카탈로그를 갱신하십시오.

```
db2upd7 -d sample
```

사용법 주의사항

이 도구는 최소한 FixPak 2가 설치된 DB2 버전 7.1 또는 버전 7.2를 수행하는 데이터베이스에서만 사용할 수 있습니다. 명령을 한 번 이상 수행하면, 오류가 보고되지 않고 각각의 카탈로그 갱신은 단 한번 적용됩니다.

새로운 내장 함수를 사용하려면, 모든 응용프로그램과 이 데이터베이스와의 연결을 해제하고 데이터베이스가 활성화되어 있을 경우 비활성화해야 합니다.

새 명령행 처리기 옵션(-x, 컬럼 포제의 인쇄 억제)

새로운 옵션인 -x는 컬럼 이름을 비롯하여 헤더없이 데이터를 리턴하도록 명령행 처리기에 지시합니다. 이 명령 옵션의 기본 설정은 OFF입니다.

DB2 CLP에 대한 트루 타입 글꼴 요구사항

DB2 명령행 처리기(CLP) 창에서 1 바이트(SBCS) 언어에 대한 자국어 문자를 올바르게 표시하려면 글꼴을 트루 타입으로 변경하십시오.

ADD DATALINKS MANAGER

이 명령에 필요한 권한 부여 레벨은 다음 중 하나입니다.

- *sysadm*
- *sysctrl*
- *sysmaint*

다음 사용 참고가 추가되어야 합니다.

이 명령은 모든 응용프로그램이 데이터베이스로부터 연결해제된 후에만 유효합니다. 이 명령이 성공하려면 DB2 Data Links Manager가 완전히 설정되어 수행 중이어야 합니다. 또한 데이터베이스가 `d1fm add_db` 명령을 사용하여 DB2 Data Links Manager에 등록되어 있어야 합니다. 데이터베이스에 추가될 수 있는 DB2 Data Links Manager의 최대 수는 16입니다.

ARCHIVE LOG (새 명령)

Archive Log

복구 가능한 데이터베이스에 사용 중인 로그 파일을 닫고 절단합니다. User Exit를 사용할 경우, 아카이브 요청을 실행하십시오.

권한 부여

다음 중 하나입니다.

- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*

필수 연결

이 명령은 지정된 데이터베이스에 자동으로 연결을 설정합니다. 연결이 이미 존재할 경우 오류가 리턴됩니다.

명령 구문

```
►► ARCHIVE LOG FOR DATABASE database-alias
                        DB
                        |
                        | On Node clause
                        |
USER username          |
  USING password      |
```

On Node clause:

```
ON Node List clause
  ALL NODES EXCEPT Node List clause
```

Node List clause:

```
NODE ( node number TO node number )
NODES
```

명령 매개변수

DATABASE database-alias

사용 중인 로그가 아카이브될 데이터베이스의 별명을 지정합니다.

USER username

연결이 시도될 사용자 이름을 식별합니다.

USING password

사용자 이름을 인증하는 암호를 지정합니다.

ON ALL NODES

db2nodes.cfg 파일의 모든 노드에 발행되어야 하는 명령을 지정합니다. 노드 절을 지정하지 않을 경우 이것이 기본값입니다.

EXCEPT

노드 목록에 지정된 것을 제외하고, db2nodes.cfg 파일의 모든 노드에 발행되어야 하는 명령을 지정합니다.

ON NODE/ON NODES

노드 집합에서 지정된 데이터베이스에 대해 아카이브되어야 하는 로그를 지정합니다.

node number

노드 목록의 노드 번호를 지정합니다.

TO node number

로그가 아카이브되어야 하는 노드 범위를 지정할 때 사용됩니다. 지정된 첫 번째 노드 번호부터 지금까지의 모든 노드 그리고, 지정된 두 번째 노드 번호가 노드 목록에 포함됩니다.

사용 참고

이 명령을 사용하여 알려진 지점까지 로그 파일의 완전한 세트를 수집할 수 있습니다. 로그 파일은 대기 데이터베이스 갱신에 사용될 수 있습니다.

이 함수는 지정된 데이터베이스에 데이터베이스 연결이 없을 때에만 실행됩니다. 이는 사용자가 확약되지 않은 트랜잭션으로 명령을 실행하는 것을 방지합니다. 그와 같이, ARCHIVE LOG 명령은 사용자의 불완전한 트랜잭션을 강제로 확약하지 않습니다. 지정된 데이터베이스와의 데이터베이스 연결이 이미 존재하고 이 명령이 실행될 경우, 명령은 종료되고 오류를 리턴합니다. 이 명령이 실행될 때 다른 응용프로그램에 지정된 데이터베이스와 진행 중인 트랜잭션이 있을 경우, 명령이 로그 버퍼를 디스크로 비우므로 약간의 성능 저하가 있습니다. 버퍼에 로그 레코드 쓰기를 시도하는 다른 트랜잭션은 비우기가 완료될 때까지 기다려야 합니다.

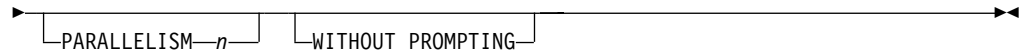
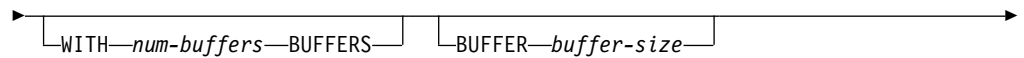
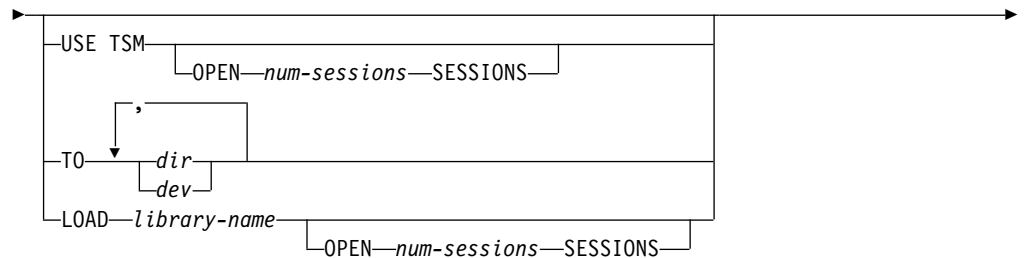
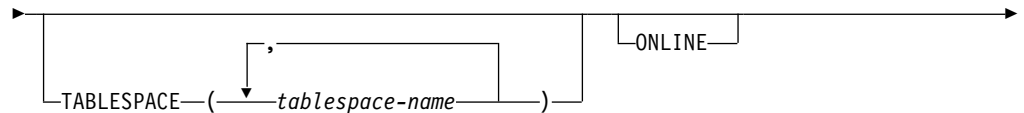
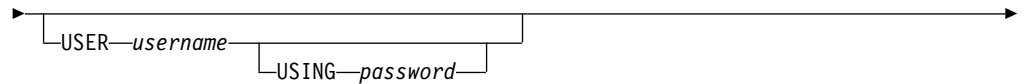
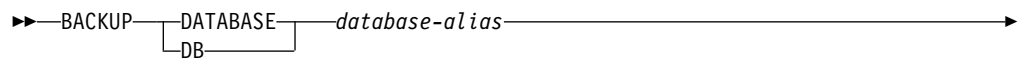
MPP 환경에서 사용되는 경우, 노드의 부속 집합은 노드 절을 사용하여 지정됩니다. 노드 절이 지정되지 않으면, 기본적으로 이 명령은 모든 노드에서 사용 중인 로그를 닫고 아카이브합니다.

이 명령을 사용하면 데이터베이스가 관련 LSN 공간 부분을 유실하도록 하며, 그 때문에 유효한 LSN의 완전 소모가 촉진됩니다.

BACKUP DATABASE

구문 도표

BACKUP DATABASE의 구문 도표는 새로운 INCREMENTAL 매개변수와 선택적 DELTA 인수를 반영하기 위해 갱신됩니다. INCREMENTAL 옵션만 지정하면 누적 백업 이미지를 생성합니다. 선택적 DELTA 인수는 비누적 백업 이미지 생성 지정에 사용될 수 있습니다.



DB2 Data Links Manager 고려사항

데이터베이스에 하나 이상의 Data Links 서버가 구성될 경우, Data Links 서버를 사용할 수 없더라도 백업 조작은 성공합니다. Data Links 서버가 재시작될 때, Data Links 서버가 데이터베이스에 다시 사용 가능해지기 전에 백업 처리가 해당 Data Links 서버에서 완료됩니다.

주: 데이터베이스에 대한 실행기록 파일에 보유한 것의(데이터베이스 구성 매개변수 num_db_backups) 두 배만큼의 백업이 사용 가능하지 않은 Data Links 서버를 여전히 대기 중이라면, 백업 조작이 실패합니다.

BIND

DB2의 명령 구문은 연합 매개변수를 표시하도록 다음과 같이 수정되어야 합니다.

```
FEDERATED---NO---  
      '-YES--'
```

FEDERATED

패키지의 정적 SQL 명령문이 별명 또는 연합 뷰를 참조하는지 지정합니다. 이 옵션을 지정하지 않고 패키지의 정적 SQL 명령문이 별명 또는 연합 뷰를 참조하면, 경고가 리턴되고 패키지가 작성됩니다.

NO 별명이나 연합 뷰는 패키지의 정적 SQL 명령문에서 참조되지 않습니다. 이 패키지의 준비 또는 바인드 중에 정적 SQL 명령문에서 별명 또는 연합 뷰를 만나면, 오류가 리턴되고 패키지가 작성되지 않습니다.

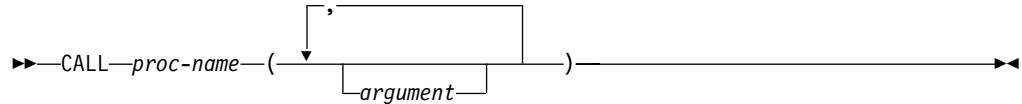
YES 패키지의 정적 SQL 명령문에서 별명이나 연합 뷰를 참조할 수 있습니다. 패키지의 준비 또는 바인드 중에 정적 SQL 명령문에서 별명이나 연합 뷰를 만나지 않으면, 오류 또는 경고가 리턴되지 않고 패키지가 작성됩니다.

주: 버전 7 FixPak 2에서, FEDERATED 옵션에 값을 지정하지 않고 소스 파일을 사전 처리 컴파일하거나 바인드 파일을 바인딩할 때 서버에 의해 SQL1179W 경고 메시지가 생성됩니다. 소스 파일 또는 바인드 파일에 별명을 참조하는 정적 SQL이 포함되어 있을 때 같은 메시지가 생성됩니다. 두 예외사항이 있습니다.

- 버전 7 FixPak 2 이전의 FixPak인 클라이언트 또는 아래 레벨의 클라이언트의 경우, sqlaprep() API가 메시지 파일에 이 SQL1179W 경고를 보고하지 않습니다. 명령행 처리기 PRECOMPILE 명령 역시 이 경우에 경고를 출력하지 않습니다.
- 버전 7 FixPak 2 이전의 FixPak인 클라이언트 또는 아래 레벨의 클라이언트의 경우, sqlabndx API가 메시지 파일에 이 SQL1179W 경고를 보고합니다. 하지만, 또한 메시지 파일에 아무 패키지도 작성되지 않았음을 의미하는 SQL0092N 메시지가 포함됩니다. 실제로 패키지가 작성되므로 이는 옳지 않습니다. 명령행 처리기 BIND 명령이 동일한 잘못된 경고를 리턴합니다.

CALL

CALL 명령의 구문은 다음과 같이 나타나야 합니다.



argument 매개변수의 설명은 다음과 같이 변경되었습니다.

Specifies one or more arguments for the stored procedure. All input and output arguments must be specified in the order defined by the procedure. Output arguments are specified using the "?" character. For example, a stored procedure foo with one integer input parameter and one output parameter would be invoked as "call foo (4, ?)".

주:

1. 운영 체제 프롬프트에서 이 유틸리티를 호출할 때, 다음과 같이 명령을 분리해야 합니다.

```
"call DEPT_MEDIAN (51)"
```

작은 따옴표(')를 사용할 수도 있습니다.

2. 호출되는 저장 프로시저는 데이터베이스에서 고유하게 이름을 지정해야 합니다.
3. 저장 프로시저는 카탈로그화되어야 합니다. 카탈로그화되지 않은 프로시저가 호출된 경우, DB21036 오류 메시지가 리턴됩니다.
4. 명령행에 매개변수가 충분히 지정되지 않았거나 명령행 매개변수들의 순서가 올바르지 않은 경우(입력, 출력), 저장 프로시저 정의에 따라 DB21101E 메시지가 리턴됩니다.
5. 결과 컬럼에 최대 1023자가 있습니다.
6. LOBS 및 2진 데이터(FOR BIT DATA, VARBINARY, LONGVARBINARY, GRAPHIC, VARGAPHIC 또는 LONGVARGRAPHIC)는 지원되지 않습니다.
7. CALL은 결과 세트를 지원합니다.
8. 지원되지 않는 유형의 OUTPUT 변수가 있는 SP가 호출될 경우, CALL은 실패하고 메시지 DB21036이 리턴됩니다.
9. CALL에 대한 INPUT 매개변수의 최대 길이는 1024자입니다.

DROP DATALINKS MANAGER (새 명령)

DROP DATALINKS MANAGER

지정된 데이터베이스에 대해 등록된 DB2 Data Links Manager의 목록에서 DB2 Data Links Manager를 삭제합니다.

권한 부여

다음 중 하나입니다.

- *sysadm*
- *sysctrl*
- *sysmaint*

명령 구문

```
▶ DROP DATALINKS MANAGER FOR DATABASE dbname USING name ▶  
                                  |                                  |  
                                  DB                                  
```

명령 매개변수

DATABASE dbname

데이터베이스 이름을 지정합니다.

USING name

LIST DATALINKS MANAGER 명령으로 표시되는 DB2 Data Links Manager의 이름을 지정합니다.

예

예 1

일부 데이터베이스 테이블에 micky.almaden.ibm.com에 링크가 있을 때 호스트 bramha.almaden.ibm.com에 상주하는 인스턴스 validate 하의 데이터베이스 TEST에서 DB2 Data Links Manager micky.almaden.ibm.com 삭제. *DB2 Data Links Manager* 삭제시 다음 단계 수행이 매우 중요합니다.

1. 데이터베이스 TEST에 대해 데이터베이스를 백업합니다.
2. micky.almaden.ibm.com에 링크가 있다면, 링크해제하십시오.
 - a. SYSADM_GROUP에 속하는 사용자 ID로 로그인하고 데이터베이스 TEST에 독점 모드 연결을 획득하십시오.

```
connect to test in exclusive mode
```

이것이 해당 사용자 ID를 사용하는 test와의 유일한 연결인지 확인하십시오. 이는 다른 새 링크가 작성되는 것을 방지합니다.

- b. 데이터베이스에서 모든 FILE LINK CONTROL DATALINK 컬럼의 목록 및 이 컬럼이 들어 있는 테이블의 목록을 획득하십시오.

```
select tabname, colname from syscat.columns where substr(dl_features, 2, 1) = 'F'
```

- c. 목록의 각 FILE LINK CONTROL DATALINK 컬럼에 대해, SQL SELECT를 발행하여 micky.almaden.ibm.com과의 링크가 있는지 판별하십시오. 예를 들어, 테이블 t의 DATALINK 컬럼 c의 경우, SELECT 문은 다음과 같을 것입니다.

```
select count(*) from t where dlurlserver(t.c) = 'MICKY.ALMADEN.IBM.COM'
```

- d. 그와 같은 링크가 들어 있는 각 FILE LINK CONTROL DATALINK 컬럼에 대해, SQL UPDATE를 발행하여 micky.almaden.ibm.com과의 링크 값을 링크해제하십시오. 예를 들어, 테이블 t의 DATALINK 컬럼 c의 경우, UPDATE 문은 다음과 같을 것입니다.

```
update t set t.c = null where dlurlserver(t.c) = 'MICKY.ALMADEN.IBM.COM'
```

t.c에 널(NULL) 입력이 불가능하면, 대신 다음을 사용할 수 있습니다.

```
update t set t.c = dlvalue('') where dlurlserver(t.c) = 'MICKY.ALMADEN.IBM.COM'
```

- e. 이 SQL UPDATE를 요약하십시오.

```
commit
```

3. DROP DATALINKS MANAGER 명령을 발행하십시오.

```
drop datalinks manager for db test using node micky.almaden.ibm.com
```

4. 변경을 유효화하고 데이터베이스에 다른 연결을 허용하기 위해 독점 모드 연결을 종료하십시오.

```
terminate
```

5. micky.almaden.ibm.com의 TEST에 대해 링크해제 처리 및 백업 정보 가비지 수집을 시작하십시오. DB2 Data Links Manager 관리자로서, micky.almaden.ibm.com에서 다음 명령을 발행하십시오.

```
dlfm drop_dlm test validate bramha.almaden.ibm.com
```

이는 사용자가 단계 3을 호출하기 전에 링크해제를 하지 않았을 경우에만 데이터베이스 TEST에 아직 링크되어 있는 파일을 링크해제합니다. micky.almaden.ibm.com에 이전에 데이터베이스 TEST에 링크되었던 파일에 대한 백업 정보가 있을 경우(예: 아카이브 파일, 메타데이터), 이 명령은 해당 정보의 가비지 수집을 시작합니다. 실제 링크해제 및 가비지 수집은 비동기로 수행됩니다.

예 2

관리 프로그램이 이미 데이터베이스 TEST에서 삭제되었을 때 micky.almaden.ibm.com이라는 DB2 Data Links Manager의 파일과의 링크인 DATALINK 값 삭제. micky.almaden.ibm.com을 삭제하는 중에 예 1의 단계들을 수행하지 않았을 경우 이것이 필요할 수 있습니다. SQL DELETE, SELECT 및 UPDATE 문은 그와 같은 DATALINK 값(SQL0368)에 대해서는 성공적이지 않습니다. 사용자가 그런 DATALINK 값이 있는 테이블 각각에 대해 조정 조작을 수행해야 합니다. micky.almaden.ibm.com과의 링크였던 각 DATALINK 값은 널(NULL) 또는 길이가 0인 DATALINK 값으로 갱신됩니다. 그런 값이 들어 있는 행은 예외 테이블(지정되어

있을 경우)에 삽입됩니다. 그러나, DATALINK 값에는 접두부 이름이 포함되지 않습니다. micky.almaden.ibm.com이 삭제되었기 때문에 원래 DATALINK 값의 접두부 이름은 시스템에 의해 더 이상 구할 수가 없습니다. 예를 들어, 원래 DATALINK 값이 'http://host.com/dlfs/x/y/a.b'이고 '/dlfs'가 접두부 이름이라면, 예외 테이블의 DATALINK 값은 'http://host.com/x/y/a.b'가 됩니다. 이러한 DATALINK 값에 의해 참조되는 파일은 DB2 Data Links Manager에서 링크된 상태로 남아 있습니다. dlfm drop_dlm 명령을 micky.almaden.ibm.com에서 발행하여 이들 파일에 대한 링크해제 처리를 실시할 수 있습니다. micky.almaden.ibm.com에 이전에 데이터베이스 TEST에 링크되었던 파일에 대한 백업 정보가 있을 경우(예: 아카이브 파일, 메타데이터), 이 명령은 해당 정보의 가비지 수집을 시작합니다. 실제 링크해제 및 가비지 수집은 비동기로 수행됩니다.

예 3

데이터베이스 TEST에 대한 DB2 Data Links Manager micky.almaden.ibm.com의 다중 실행. 이 시나리오는 DB2 Data Links Manager가 삭제된 후 다시 등록되고, 그 후에는 완전히 새 DB2 Data Links Manager로서 취급되는 것을 예시합니다. 다음 단계는 단지 가능한 시나리오를 예시하는 것입니다. 권장된 대로 micky.almaden.ibm.com 삭제를 위한 예1의 단계를 수행한다면, micky.almaden.ibm.com의 이전 실행은 없습니다. 즉, 아래 7단계에서 오류 SQL0368이 나타나지 않습니다.

1. micky.almaden.ibm.com을 데이터베이스 TEST에 등록하십시오.

```
add datalinks manager for db test using node micky.almaden.ibm.com port 14578
```

2. micky.almaden.ibm.com의 파일에 링크를 작성하십시오.

```
connect to test
create table t(c1 int, c2 datalink linktype url file link control mode db2options)
insert into t values(1, dlvalue('file://micky.almaden.ibm.com/pictures/yosemite.jpg'))
commit
terminate
```

3. micky.almaden.ibm.com을 데이터베이스 TEST에서 삭제하십시오.

```
drop datalinks manager for db test using micky.almaden.ibm.com
```

4. DATALINK 값을 선택하십시오.

```
connect to test
select * from t
terminate
```

다음과 같은 내용이 보일 것입니다.

```
SQL0368 The DB2 Data Links Manager "MICKY.ALMADEN.IBM.COM"이
데이터베이스에 등록되지 않았습니다. SQLSTATE=55022.
```

5. micky.almaden.ibm.com을 데이터베이스 TEST에 다시 등록하십시오.

```
add datalinks manager for db test using node micky.almaden.ibm.com port 14578
```

6. 더 많은 DATALINK 값을 삽입하십시오.

```
connect to test
insert into t values(2, d1value('file://micky.almaden.ibm.com/pictures/tahoe.jpg'))
commit
```

7. DATALINK 값을 선택하십시오.

```
select c2 from t where c1 = 2
```

선택된 값이 micky.almaden.ibm.com의 현재 등록된 실행과의 링크이므로 이 명령은 성공적입니다.

```
select c2 from t where c1 = 1
```

다음은 리턴합니다.

```
SQL0368 The DB2 Data Links Manager "MICKY.ALMADEN.IBM.COM"이
데이터베이스에 등록되지 않았습니다. SQLSTATE=55022.
```

선택된 값이 위의 3 단계에서 삭제된 micky.almaden.ibm.com의 현재 등록된 실행과의 링크이기 때문입니다.

사용 참고

DROP DATALINKS MANAGER 명령의 효과는 구간 복원될 수 없습니다. DROP DATALINKS MANAGER 명령 사용시 예 1에 설명된 단계를 따르는 것이 매우 중요합니다.

이 명령은 모든 응용프로그램이 데이터베이스로부터 연결해제된 후에만 유효합니다.

명령이 성공적으로 완료되면, 사용자는 (DB210201I)DB2 Data Links Manager에 아무 처리도 수행되지 않았다는 정보를 받습니다. DB2 Data Links Manager를 삭제하기 전에, 사용자는 데이터베이스가 해당 DB2LData Links Manager의 파일과 링크를 갖고 있는지 확인해야 합니다. DB2 Data Links Manager가 삭제된 후에 링크가 데이터베이스에 존재할 경우, 사용자는 링크를 제거하기 위해 조정 유틸리티를 수행해야 합니다. 조정 유틸리티는 이러한 링크를 널(NULL)로 설정하거나(DATALINK 컬럼이 널(NULL)이 가능하다면), 길이가 0인 DATALINK 값으로 설정합니다.

데이터베이스와 삭제된 DB2 Data Links Manager 사이의 링크에 해당하는 파일은 링크된 상태로 남아 있습니다. 즉, 읽기, 쓰기, 이름 바꾸기, 삭제, 권한 변경 또는 소유자 변경과 같은 조작에 액세스 불가능합니다.

DB2 Data Links Manager의 링크해제된 파일의 아카이브 사본은 이 명령에 의해 가비지 수집되지 않습니다. 그러나, 사용자가 DB2 Data Links Manager에서 dlfm drop_dlm 명령을 사용하여 명시적으로 링크해제 처리 및 가비지 수집을 시작할 수 있습니다.

DB2 Data Links Manager를 삭제하기 전에 데이터베이스를 백업하도록 권장합니다. 더불어, 모든 복제 복사 작업 내역이 이 DB2 Data Links Manager와 관련된 모든 변경사항을 복제했는지 확인하십시오.

데이터베이스에서 DB2 Data Links Manager를 삭제하기 전에 백업했고, DB2 DataLinks Manager가 삭제된 후 그 백업 이미지가 복원에 사용될 경우, 복원 또는 롤포워드 처리가 특정 테이블을 데이터 링크 조정 보류(DRP) 상태로 만듭니다.

EXPORT

"DB2 Data Links Manager 고려사항" 절에서, DATALINK 컬럼이 참조하는 테이블과 해당 파일이 내보내기를 위해 복사되는지를 확인하는 절차의 3 단계는 다음과 같아야 합니다.

3. Run the `dlfm_export` utility at each Data Links server. Input to the `dlfm_export` utility is the control file name, which is generated by the export utility. This produces a tar (or equivalent) archive of the files listed within the control file. For Distributed File Systems (DFS), the `dlfm_export` utility will get the DCE network root credentials before archiving the files listed in the control file. `dlfm_export` does not capture the ACLs information of the files that are archived.

동일한 절에서, "Successful execution of EXPORT results in the generation of the following files" 문장은 아래와 같이 수정해야 합니다.

첫번째 사항에서 두 번째 문장은 다음과 같아야 합니다.

A DATALINK column value in this file has the same format as that used by the import and load utilities.

두 번째 사항에서 첫번째 문장은 다음과 같아야 합니다.

Control files `server_name`, which are generated for each Data Links server. (On the Windows NT operating system, a single control file, `ctrlfile.lst`, is used by all Data Links servers. For DFS, there is one control file for each cell.)

다음의 문장을 표5 앞의 문단에 추가해야 합니다.

For more information about `dlfm_export`, refer to the "Data Movement Utilities Guide and Reference" under "Using Export to move DB2 Data Links Manager Data".

GET DATABASE CONFIGURATION

DL_TIME_DROP 구성 매개변수의 설명은 다음과 같이 변경되어야 합니다.

Applies to DB2 Data Links Manager only. This parameter specifies the number of days files would be retained on an archive server (such as a TSM server) after a DROP DATABASE command is issued.

새 매개변수 TRACKMOD가 GET DATABASE CONFIGURATION 명령에 추가됩니다. 구문은 다음과 같습니다.

▶ GET DATABASE CONFIGURATION FOR *database-alias*

```
▶ GET DATABASE CONFIGURATION FOR database-alias
```



매개변수 설명이 다음과 같이 추가됩니다.

TRACKMOD

중분 백업을 허용하기 위해 DB2가 데이터베이스에서 수정된 페이지를 추적해야 하는지 여부를 지시합니다.

OFF 변경된 페이지가 추적되지 않도록 지정합니다. 버전 7.1 FixPak 3 이전에 작성된 데이터베이스에 대한 기본값입니다.

ON 변경된 페이지가 추적되도록 지정합니다. 이 매개변수가 설정되면, 데이터베이스의 중분 백업이 이루어집니다. 버전 7.1 FixPak 3 이후에 작성된 데이터베이스에 대한 기본값입니다.

GET ROUTINE (새 명령)

GET ROUTINE

지정된 SQL 루틴에 대한 루틴 SQL 아카이브(SAR) 파일을 검색합니다.

권한 부여

dbadm

필수 연결

데이터베이스. 내재된 연결이 사용될 경우, 기본 데이터베이스와 연결이 설정됩니다.

명령 구문

```
▶▶ GET ROUTINE INTO file_name FROM SPECIFIC PROCEDURE routine_name ▶▶
```

명령 매개변수

INTO *file-name*

루틴 SQL 아카이브(SAR)가 저장된 파일의 이름.

FROM

검색된 루틴 스펙의 시작을 지시합니다.

SPECIFIC

지정된 *routine-name*은 고유 이름으로 제공됩니다.

PROCEDURE

루틴이 SQL 프로시저어입니다.

routine-name

프로시저어의 이름. SPECIFIC을 지정하면 프로시저어의 특정한 이름입니다. 이름이 스키마 이름으로 규정되지 않으면, CURRENT SCHEMA가 루틴의 스키마 이름으로 사용됩니다. *routine-name*은 SQL 프로시저어로 정의된 기존 프로시저어어야 합니다.

예

```
GET ROUTINE INTO procs/proc1.sar FROM PROCEDURE myapp1.proc1;
```

GET SNAPSHOT

FCM FOR ALL NODES 매개변수의 설명은 다음과 같아야 합니다.

GET SNAPSHOT문이 발행된 노드 및 EEE 인스턴스의 다른 노드에 대해 노드 사이의 FCM(Fast Communication Manager) 통계를 제공합니다.

IMPORT

"DB2 Data Links Manager 고려사항" 절에서, 다음의 문장을 3 단계에 추가해야 합니다.

For Distributed File Systems (DFS), update the cell name information in the URLs (of the DATALINK columns) from the exported data for the SQL table, if required.

다음의 문장은 4 단계에 추가해야 합니다.

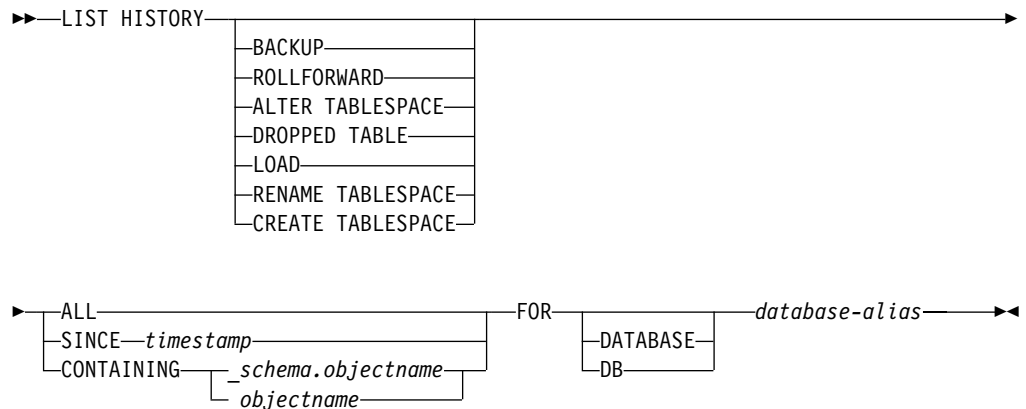
For DFS, define the cells at the target configuration in the DB2 Data Links Manager configuration file.

4 단계 다음의 문단은 아래와 같아야 합니다.

When the import utility runs against the target database, files referred to by DATALINK column data are linked on the appropriate Data Links servers.

LIST HISTORY

CREATE TABLESPACE 매개변수가 LIST HISTORY 명령에 추가됩니다. 구문 도표는 다음과 같습니다.



매개변수 설명이 다음과 같이 추가됩니다.

CREATE TABLESPACE

모든 CREATE TABLESPACE 및 DROP TABLESPACE 조작을 나열합니다.

사용 참고가 다음과 같이 갱신됩니다.

다음 기호가 보고서의 조작 섹션에 추가됩니다.

- A - 테이블 공간 작성
- O - 테이블 공간 삭제

- U - 로드 해제

보고서의 유형 섹션 아래의 기호가 다음과 같이 재구성됩니다.

- 백업 유형
 - F - 오프라인
 - N - 온라인
 - I - 증분 오프라인
 - O - 증분 온라인
 - D - 델타 오프라인
 - E - 델타 온라인
- 롤 포워드 유형
 - E - 로그의 끝
 - P - 특정 시점
- 로드 유형
 - I - 삽입
 - R - 바꾸기
- 테이블 공간 변경 유형
 - C - 컨테이너 추가
 - R - 재균형(Rebalance)
- Quiesce 유형
 - S - Quiesce 공유
 - U - Quiesce 갱신
 - X - Quiesce 독점
 - Z - Quiesce 재설정

LOAD

정의된 "DB2 Data Links Manager 고려사항" 절에서, FILE LINK CONTROL로 DATALINK 컬럼이 있는 테이블에 데이터를 로드하는 경우, 로드 유틸리티를 호출하기 전에 수행할 절차의 1 단계에 다음의 문장을 추가하십시오.

For Distributed File Systems (DFS), ensure that the DB2 Data Links Managers within the target cell are registered.

다음의 문장은 5 단계에 추가해야 합니다.

For DFS, register the cells at the target configuration referred to by DATALINK data (to be loaded) in the DB2 Data Links Manager configuration file.

"입력 파일에 DATALINK 정보 표시" 절에서, `urlname`에 대한 매개변수 설명 다음의 첫번째 참고는 아래와 같아야 합니다.

Currently "http", "file", "unc", and "dfs" are permitted as a schema name.

두 번째 참고의 첫번째 문장은 다음과 같아야 합니다.

The prefix (schema, host, and port) of the URL name is optional. For DFS, the prefix refers to the schema cellname filespace-junction portion.

컬럼 식별자가 있는 ASCII(DEL) 파일 형식과 컬럼 식별자가 없는 ASCII(ASC) 파일 형식 모두에 대한 DATALINK 데이터 예에서, 3번째 예는 제거되어야 합니다.

컬럼의 로드 또는 가져오기 스펙이 `DL_URL_DEFAULT_PREFIX`로 가정되는 DATALINK 데이터 예를 지우고 다음과 같이 바꿔야 합니다.

다음은 컬럼의 로드 또는 가져오기 스펙이 `DL_URL_REPLACE_PREFIX("http://qso")`로 가정되는 DATALINK 데이터 예입니다.

- `http://www.almaden.ibm.com/mrep/intro.mpeg`

이 샘플 URL은 다음 부분으로 저장됩니다.

- 스키마 = http
- 서버 = qso
- 경로 = /mrep/intro.mpeg
- 주석 = NULL 문자열

- `/u/me/myfile.ps`

다음의 부분으로 저장됩니다.

- 스키마 = http
- 서버 = qso
- 경로 = /u/me/myfile.ps
- 주석 = NULL 문자열

PING (새 명령)

PING

연결을 설정하기 위해 DB2 Connect가 사용되는 클라이언트와 데이터베이스 서버 사이에서 기초 연결의 네트워크 응답 시간을 테스트합니다.

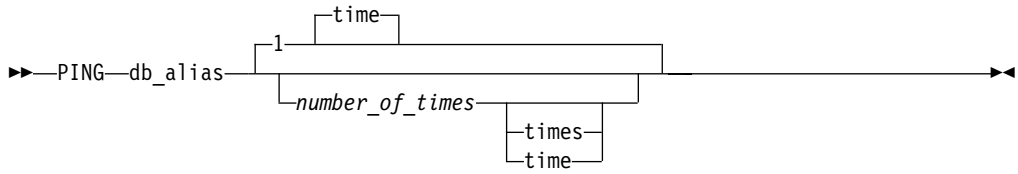
권한 부여

없음

필수 연결

데이터베이스

명령 구문



명령 매개변수

db_alias

Ping이 전송되는 DRDA 서버의 데이터베이스에 대한 데이터베이스 별명을 지정합니다.

주: 필수라도 이 매개변수는 현재 사용되지 않습니다. 앞으로 사용되기 위해 예약된 것입니다. 유효한 모든 데이터베이스 별명을 지정할 수 있습니다.

number of times

이 테스트의 반복 수를 지정합니다. 값은 1 - 32767 이내여야 합니다. 기본값은 1입니다. 각 반복마다 1회가 리턴됩니다.

예

호스트 데이터베이스 서버 hostdb로의 연결에 대한 네트워크 응답 시간을 테스트하려면 다음을 수행하십시오.

```
db2 ping hostdb 1
또는
db2 ping hostdb
```

명령은 다음과 같은 출력을 표시합니다.

```
Elapsed time: 7221 microseconds
```

호스트 데이터베이스 서버 hostdb로의 5번 연결에 대한 네트워크 응답 시간을 테스트하려면 다음을 수행하십시오.

```
db2 ping hostdb 5
또는
db2 ping hostdb 5 times
```

명령은 다음과 같은 출력을 표시합니다.

```
Elapsed time: 8412 microseconds  
Elapsed time: 11876 microseconds  
Elapsed time: 7789 microseconds  
Elapsed time: 10124 microseconds  
Elapsed time: 10988 microseconds
```

사용 참고

이 명령을 호출하려면 데이터베이스에 연결되어 있어야 하며, 그렇지 않으면 오류가 발생합니다.

리턴된 경과 시간은 DB2 Connect를 통한 클라이언트와 DRDA 서버 데이터베이스와의 연결을 위한 것입니다.

PUT ROUTINE (새 명령)

PUT ROUTINE

지정된 루틴 SQL 아카이브(SAR) 파일을 사용하여 데이터베이스에 루틴을 정의합니다.

권한 부여

dbadm

필수 연결

데이터베이스 내재된 연결이 사용될 경우, 기본 데이터베이스와 연결이 설정됩니다.

명령 구문

```
▶▶ PUT ROUTINE FROM file-name [OWNER new-owner] [USE REGISTERS]
```

명령 매개변수

FROM file-name

루틴 SQL 아카이브(SAR)가 저장된 파일의 이름.

OWNER new-owner

루틴의 권한 부여 점검에 사용될 루틴의 새 권한 부여 이름을 지정합니다. 새 소유자는 정의될 루틴에 대해 필요한 특권을 소유해야 합니다. OWNER 절을 지정하지 않으면, 처음에 루틴을 정의했던 권한 부여 이름이 사용됩니다.

USE REGISTERS

CURRENT SCHEMA 및 CURRENT PATH 특수 레지스터가 루틴 정의에 사용됨을 의미합니다. 이 절을 지정하지 않으면, 기본 스키마 및 SQL 경로의 설정값은 루틴이 원래 정의될 때 사용된 설정값입니다. CURRENT SCHEMA는 루틴 정의(루틴 이름 포함)에서 규정되지 않은 스키마 이름에 사용되고, CURRENT PATH는 루틴 정의에서 규정되지 않은 루틴 및 데이터 유형 결정에 사용됩니다.

예

```
PUT ROUTINE FROM procs/procl.sar;
```

사용 참고

제공된 스키마 아래에 하나 이상의 프로시저어를 동시에 설치할 수 없습니다.

RECONCILE

다음 사용 참고가 명령 설명에 추가되어야 합니다.

조정 중에, 테이블 데이터에 따라 존재하지만, Data Links Manager 메타데이터에 따라서는 존재하지 않는 파일들을 다른 충돌이 없다면 링크하려는 시도가 이루어집니다. 필요한 DB2 Data Links Manager는 테이블에 참조된 DATALINK 값이 있는 것입니다. 필요한 DB2 Data Links Manager가 사용 불가능하고, 또 데이터베이스에 구성되었지만 테이블 데이터의 일부가 아닌 경우에도 조정을 실행할 수 있습니다.

예외 테이블이 지정되어 있지 않으면, 파일 참조가 재구축될 수 없는 DATALINK 컬럼 값 각각에 대한 호스트 이름, 파일 이름, 컬럼 ID 및 이유 코드가 예외 보고서 파일 (*filename.exp*)에 있습니다. DB2 Data Links Manager 자체가 DROP DATALINKS MANAGER 명령을 사용하여 데이터베이스에서 삭제되었기 때문에 파일 참조가 재구축될 수 없을 경우, 예외 보고서 파일에 보고된 파일 이름은 전체 파일 이름이 아닙니다. 즉 접두부 부분이 누락되었습니다. DB2 Data Links Manager가 삭제되었거나 사용 불가능한 DATALINK 값에 대한 예외 테이블에서, DATALINK 값의 파일 이름은 전체 파일 이름이 아닙니다. 접두부 부분이 누락되었습니다. 예를 들어, 원래 DATALINK 값이 'http://host.com/dlfs/x/y/a.b'이면, 예외 테이블에 보고되는 값은 'http://host.com/x/y/a.b'입니다. 즉, 접두부 이름 'dlfs'가 포함되지 않습니다. 이 경우 예외 보고서 파일에는 'x/y/a.b'가 들어 갑니다. 즉 접두부 이름 'dlfs'가 포함되지 않습니다.

조정 프로세스 종료시, 조정 처리가 모든 필수 DB2 Data Links Manager에서 완료되었을 경우에만 테이블이 데이터 링크 조정 보류(DRP) 상태에서 벗어납니다. 조정 처리가 필수 DB2 Data Links Manager(사용 불가능하기 때문에)에서 보류 중이면, 테이블은 DRP 상태에 남아 있거나 DRP 상태가 됩니다.

다음은 가능한 위반 목록에 추가해야 합니다.

```
00010-DB2 Data Links Manager referenced by the DATALINK value
has been dropped from the database using the DROP DATALINKS
MANAGER command. In this case, the corresponding DATALINK value
in the exception table will not contain the prefix name.
For example, if the original DATALINK value was
'http://host.com/dlfs/prfx/x/y/a.b', and '/dlfs/prfx' is
the prefix name, the exception table will contain
'http://host.com/x/y/a.b'.
```

REORGANIZE TABLE

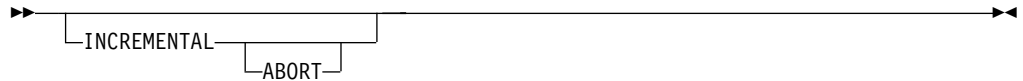
다음 문장이 사용 참고에 추가됩니다.

REORGANIZE TABLE은 색인 확장자에 기반한 색인을 사용할 수 없습니다.

RESTORE DATABASE

구문

다음 옵션이 TABLESPACE/TABLESPACE ONLINE/HISTORY FILE 옵션 뒤의 RESTORE DATABASE 명령 구문에 추가됩니다.



매개변수 설명이 다음과 같이 추가됩니다.

INCREMENTAL

데이터베이스의 수동 누적 복원이 이루어집니다. 사용자는 각각의 복원 명령을 발행합니다.

ABORT

이 매개변수는 성공적인 완료 전에 증분 복원을 종료하는 데에 사용되어야 합니다.

다음 예가 또한 추가됩니다.

다음은 복구 가능한 데이터베이스에서 주간 증분 백업 전략 샘플입니다. 전체 백업은 일주일에 한번, 델타는 매일, 증분은 주중에 스케줄되었습니다.

```
(Sun) backup db kdr use adsm
(Mon) backup db kdr online incremental delta use adsm
(Tue) backup db kdr online incremental delta use adsm
(Wed) backup db kdr online incremental use adsm
(Thu) backup db kdr online incremental delta use adsm
(Fri) backup db kdr online incremental delta use adsm
(Sat) backup db kdr online incremental use adsm
```

위에서 금요일 아침에 작성된 이미지의 수동 데이터베이스 복원에 대해, 다음 명령을 실행하십시오.

```
restore db kdr incremental taken at (Thu)
restore db kdr incremental taken at (Sun)
restore db kdr incremental taken at (Wed)
restore db kdr incremental taken at (Thu)
```

주: form db2 restore db <name>의 RESTORE 명령은 복원되는 이미지가 데이터베이스 이미지이거나 테이블 공간 이미지이거나 상관없이 전체 데이터베이스 복원을 수행합니다. db2 restore db <name> tablespace 형태의 RESTORE 명령은 이미지에서 발견된 테이블 공간의 테이블 공간 복원을 수행합니다. 테이블 공간 목록이 제공되는 RESTORE 명령은 명시적으로 나열된 모든 테이블 공간에 대해 복원을 수행합니다.

DB2 Data Links Manager 고려사항

"DB2 Data Links Manager 고려사항" 절의 두 번째 문단을 다음으로 바꾸어야 합니다.

If one or more Data Links servers are configured for the database, the restore operation will succeed, even if a Data Links server is not available. When the Data Links server restarts, restore processing will be completed on that Data Links server before it becomes available to the database again.
NOTE: If a database restore operation is still waiting for an unavailable Data Links server, any subsequent database or table space restore operations will fail.

ROLLFORWARD DATABASE

"DB2 Data Links Manager 고려사항" 절의 두 번째 문단을 다음으로 바꾸어야 합니다.

If one or more Data Links servers are configured for the database, the rollforward operation will succeed, even if a Data Links server is not available. When the Data Links server restarts, rollforward processing will be completed on that Data Links server before it becomes available to the database again.

CLP 리턴 코드의 문서 오류

명령행 처리기 리턴 코드 절에서, 두 번째 문단은 다음과 같아야 합니다.

예를 들어, 다음 Bourne 셸 스크립트는 GET DATABASE MANAGER CONFIGURATION 명령을 실행한 후, CLP 리턴 코드를 검사합니다.

```
db2 get database manager configuration
if [ "$?" = "0" ]
then echo "OK!"
fi
```

데이터 이동 유틸리티 안내 및 참조서

가져오기

버퍼된 삽입이 있는 가져오기 사용

이 절 끝의 참고는 다음과 같아야 합니다.

주: EEE를 제외한 모든 환경에서, INSERT_UPDATE 매개변수가 지정된 가져오기 조
작 중에 버퍼된 삽입 기능이 사용 불가능합니다.

로드

로드 조작 이후 보류 상태

이 절의 마지막 문단에서 처음 두 문장이 다음과 같이 변경되었습니다.

로드 프로세스와 관련된 네번째 가능한 상태(점검 보류 상태)는
참조 및 점검 제한조건, DATALINKS 제한 조건, AST 제한조건
또는 생성된 컬럼 제한조건에 관계됩니다. 예를 들어 기존 테이블이
종속 테이블의 외부 키에 의해 참조되는 기본 키가 들어 있는
상위 테이블일 경우, 상위 테이블에서 데이터를 바꾸면 두 테이블
모두(테이블 공간이 아님) 점검 보류 상태가 됩니다.

로드 제한사항 및 한계

다음의 제한사항이 생성된 컬럼과 로드 유틸리티에 적용됩니다.

- 생성된 컬럼이 색인의 "포함 컬럼"이 아니거나 generatedoverride 파일 유형 수
정자가 사용되지 않은 경우, 생성된 컬럼이 고유 색인에 있는 테이블을 로드할 수 없
습니다. 이 수정자가 사용된 경우, 컬럼의 모든 값이 입력 데이터 파일에 제공될 것
으로 예상됩니다.
- generatedoverride 파일 유형 수정자가 사용되지 않은 경우, 파티션 키에 생성된
컬럼이 있는 테이블을 로드할 수 없습니다. 이 수정자가 사용된 경우, 컬럼의 모든
값이 입력 데이터 파일에 제공될 것으로 예상됩니다.

totalreespace 파일 유형 수정자

totalreespace 파일 유형 수정자(LOAD)가 0에서 2 147 483 647 사이의 값을 승인
하도록 수정되었습니다.

자동 로드 프로그램

Authentication=YES일 경우 자동 로드 프로그램 수행에 rexecd 필요

자동 로드 프로그램 옵션 절에서 다음 참고가 AUTHENTICATION 및 PASSWORD 매개변수 설명에 추가됩니다.

Linux 환경에서, authentication 옵션을 YES로 설정하여 자동 로드 프로그램을 실행할 경우, 모든 머신에서 rexecd를 사용해야 합니다. rexecd가 사용되지 않으면, 다음 오류 메시지가 생성됩니다.

```
openbreeze.torolab.ibm.com: Connection refused
SQL6554N  원격 프로세스 실행을 시도하는 중 오류가 발생했습니다.
```

다음 오류 메시지는 db2diag.log 파일에 생성됩니다.

```
2000-10-11-13.04.16.832852 Instance:svtdbm Node:000
PID:19612(db2at1d) Appid:
oper_system_services sqloRemoteExec Probe:31
```

복제 안내 및 참조서

복제 및 IBM이 아닌 서버

DataJoiner 버전 2 이상을 사용하여 Informix, Microsoft SQL Server, Oracle, Sybase 및 Sybase SQL Anywhere와 같은 IBM이 아닌 서버와의 사이에 데이터를 복제해야 합니다. DB2 Relational Connect 버전 7에는 갱신 기능이 없으므로 이 복제 유형으로 관계 연결 기능을 사용할 수 없습니다. 또한, DJRA(DataJoiner Replication Administration)를 사용하여 기존의 모든 DB2 및 DataJoiner 버전에 대해 모든 플랫폼(AS/400, OS/2, OS/390, UNIX 및 Windows)에서 이중 복제를 관리해야 합니다.

Windows 2000에서의 복제

DB2 DataPropagator 버전 7은 Windows 2000 운영 체제와 호환할 수 있습니다.

SQL 파일 저장 시 알려진 오류

DB2 Connect Personal Edition에서 제어 센터를 사용하는 경우, SQL 파일을 저장할 수 없습니다. SQL 파일을 저장하려고 하면 DAS(Database Administration Server)가 실행 중이 아님을 나타내는 오류 메시지가 표시됩니다. 실제로 DAS는 DB2 Connect PE를 포함하고 있지 않으므로 사용할 수 없습니다.

DB2 유지보수

복제 환경에서 사용하는 다양한 DB2 제품용 최신 DB2 유지보수를 설치하는 것이 좋습니다.

웹의 Data Difference 유틸리티

<ftp://ftp.software.ibm.com/ps/products/datapropagator/fixes/>에서 Data Difference 유틸리티(DDU)를 다운로드할 수 있습니다. DDU는 동일한 파일의 두 버전을 비교하여 차이를 나타내는 출력 파일을 만드는 샘플 유틸리티입니다. 자세한 내용은 샘플 유틸리티에 포함된 README 파일을 참조하십시오.

데이터 복제 시나리오

복제 시나리오

새로운 이질형 데이터 복제 시나리오에 대해서는 DataPropagator 웹 사이트 (<http://www.ibm.com/software/data/dpropr/>)의 라이브러리 페이지를 참조하십시오. 이 시나리오에서 단계를 수행하여 AIX의 Oracle 데이터베이스에 있는 복제-소스 테이블에서 Windows NT용 DB2에 있는 데이터베이스의 목표 테이블로 변경사항을 복사하십시오. 이 시나리오에서는 DB2 DataJoiner 복제 관리(DJRA) 도구, Capture 트리거, Apply 프로그램 및 DB2 DataJoiner를 사용합니다.

책의 44 페이지에서, 암호 파일 작성을 위한 6 단계는 다음과 같습니다.

6 단계: 암호 파일 작성

Apply 프로그램은 소스 서버에 연결되어야 하므로, 사용자 인증을 위한 암호 파일을 작성해야 합니다. Apply 프로그램을 수행할 사용자 ID가 암호 파일을 읽을 수 있도록 하십시오.

암호 파일 작성하기

1. Windows NT 명령 프롬프트 창에서 C:\scripts 디렉토리를 변경하십시오.
2. DEPTQUAL.PWD 디렉토리에서 새 파일을 작성하십시오. 메모장과 같은 텍스트 편집기를 사용하여 이 파일을 작성할 수 있습니다. 암호 파일의 이름 지정 규칙은 *applyqual.pwd*입니다. 여기서 *applyqual*은 복사 작업 내역 세트를 작성할 때 사용했던 Apply 규정자의 대소문자와 값이 반드시 일치해야 하는 대소문자 구분 문자열입니다. 이 시나리오의 경우, Apply 규정자는 DEPTQUAL입니다.

주: 버전 5 DB2 DataPropagator의 파일 이름 지정 규칙도 지원됩니다.

3. 암호 파일 내용에는 다음 두가지 형식이 있습니다.

```
SERVER=server USER=userid PWD=password
```

여기서,

server

복사 작업 내역 세트 테이블에서 표시된 그대로의 소스, 목표 또는 제어 서버의 이름. 이 시나리오의 경우, 이들 이름들은 SAMPLE 및 COPYDB입니다.

userid

특별한 데이터베이스를 관리할 때 사용하려는 사용자 ID. 이 값은 Windows NT 및 UNIX 운영 체제에서 대소문자가 구별됩니다.

password

해당 사용자 ID와 연관된 암호. 이 값은 Windows NT 및 UNIX 운영 체제에서 대소문자가 구별됩니다.

이 파일에 공백행이나 주석행을 두지 마십시오. 서버 이름, 사용자 ID 및 암호 정보만을 추가하십시오.

4. 암호 파일 내용은 다음과 비슷해야 합니다.

```
SERVER=SAMPLE USER=subina PWD=subpw  
SERVER=COPYDB USER=subina PWD=subpw
```

DB2 인증 및 보안에 대해서는, *IBM DB2 관리 안내서*를 참조하십시오.

복제 계획

테이블 및 컬럼 이름

복제에서는 테이블 및 컬럼 이름에 공백을 지원하지 않습니다.

DATALINK 복제

DATALINK 복제는 버전 7.1 FixPak 1의 일부로서 Solaris에서 사용 가능합니다. 여기에는 소스 및 목표 DATALINK 파일 시스템에서 수행되고 제공된 파일의 마지막 수정 시간을 표시하는 MDTM(modtime) 명령을 지원하는 FTP 디먼이 필요합니다. Solaris 운영 체제의 버전 2.6 또는 MDTM용 FTP 지원이 포함되지 않는 기타 버전을 사용하는 경우, WU-FTPD같은 추가 소프트웨어가 필요합니다.

AS/400의 DB2 데이터베이스와 다른 플랫폼의 DB2 데이터베이스 사이에서 DATALINK 컬럼을 복제할 수 없습니다.

AS/400 플랫폼에서는 DATALINK 값의 속성인 "주석"의 복제가 지원되지 않습니다.

AIX 4.2를 수행하는 경우, 기본 User Exit 프로그램(ASNDLCOPY)을 수행하기 전에 APAR IY03101에 대한 PTF를 설치해야 합니다(AIX 4210-06 RECOMMENDED MAINTENANCE FOR AIX 4.2.1). 이 PTF에는 FTP 디먼의 "modtime/MDTM" 명령에 대한 Y2K 수정이 들어 있습니다. 수정을 확인하려면, "modtime <file>" 명령에서 리턴된 최종 수정 시간을 확인하십시오. 여기서 <file>은 2000년 1월 1일 이후에 수정된 파일입니다.

목표 테이블이 외부 CCD 테이블인 경우, DB2 DataPropagator는 ASNDLCOPY 루틴을 호출하여 DATALINK 파일을 복제합니다. 각 프로그램의 소스 코드에 대한 서론 절에서 ASNDLCOPY 및 ASNDLCOPYD 프로그램의 사용 방법에 대한 최신 정보를 참조하십시오. 다음의 제한사항이 적용됩니다.

- 내부 CCD 테이블에는 DATALINK 값이 아닌 DATALINK 표시기가 포함될 수 있습니다.
- 압축 외부 CCD 테이블에 DATALINK 값이 포함될 수 있습니다.
- 비압축 CCD 목표 테이블에는 DATALINK 컬럼이 포함될 수 없습니다.

- 소스 및 목표 서버가 동일한 경우, 복사 작업 내역 세트에는 DATALINK 컬럼이 있는 구성원이 포함되면 안됩니다.

LOB 제한사항

압축 내부 CCD 테이블에는 LOB 컬럼이나 LOB 표시기에 대한 참조가 포함될 수 없습니다.

복제 계획

"연결성"에는 다음의 사실이 포함되어야 합니다.

Apply 프로그램이 제어 서버에 연결될 수 없는 경우,
Apply 프로그램은 종료됩니다.

AS/400에 데이터 블로킹을 사용할 때에는 간격 중에 복제한 총 데이터 양이 "4백만 행"을 초과하지 않도록 해야 합니다. 책에서 언급된 "4 MB"는 틀린 정보입니다.

복제 환경 설정

update-anywhere 전제조건

충돌 검출 및 복사 작업 내역 세트에서 150개 이상의 복사 작업 내역 세트 구성원으로 update-anywhere 복제를 설정하려면, 다음의 DDL을 수행하여 제어 서버에 ASN.IBMSNAP_COMPENSATE 테이블을 작성해야 합니다.

```
CREATE TABLE ASN.IBMSNAP_COMPENSATE (
    APPLY_QUAL char(18) NOT NULL,
    MEMBER SMALLINT,
    INTENTSEQ CHAR(10) FOR BIT DATA,
    OPERATION CHAR(1));
```

복제 환경 설정

"CD 테이블, 색인 및 테이블 공간 이름 사용자 정의"에서 DPREPL.DFT 파일이 \sqllib\bin 디렉토리 또는 \sqllib\java 디렉토리에 있다고 서술하고 있습니다. 이는 틀린 서술로, DPREPL.DFT는 \sqllib\cc 디렉토리에 있습니다.

"보유 제한 설명"에서 Capture warm starts 또는 Capture prune 명령을 사용할 경우에만 행을 제거하기 위해 보유 제한을 사용해야 함을 명시해야 합니다. auto-pruning 옵션과 함께 Capture를 시작하면 행을 제거할 때 보유 제한을 사용하지 않습니다.

문제점 판별

복제 분석기가 Windows 32 비트 시스템과 AIX에서 수행됩니다. AIX에서 분석기를 수행하려면, PATH 환경 변수에서 sqlllib/bin 디렉토리가 /usr/local/bin 앞에 나타나도록 하여 /usr/local/bin/analyze와의 충돌을 방지해야 합니다.

복제 분석기에는 CT와 AT의 두 가지 추가 선택 키워드가 있습니다.

CT=*n* 캡처 추적 테이블에서 *n*일 이후의 항목들만을 표시합니다. 이 키워드는 선택적입니다. 이 키워드를 지정하지 않으면, 기본값은 7일입니다.

AT=*n* 적용 추적 테이블에서 *n*일보다 새로운 항목들만을 표시합니다. 이 키워드는 선택적입니다. 이 키워드를 지정하지 않으면, 기본값은 7일입니다.

예:

```
analyze mydb1 mydb2 f=mydirectory ct=4 at=2 deepcheck q=applyqual1
```

복제 분석기의 경우, 다음의 키워드 정보가 갱신됩니다.

deepcheck

분석기가 CD 및 UOW 테이블 제거 정보, OS/390용 DB2 테이블 공간 파티션 및 압축 세부사항, 복사 작업 내역 키, 복사 작업 내역 타임라인 및 복사 작업 내역 세트 SQL문 오류와 관련된 목표 색인의 분석을 비롯한 보다 완전한 분석을 수행하도록 지정합니다. 이 분석에는 모든 서버가 포함됩니다. 이 키워드는 선택적입니다.

lightcheck

보고서에서 ASN.IBMSNAP_SUBS_COLS 테이블의 모든 컬럼 세부사항, 복사 작업 내역 오류 또는 비정상 또는 생략 및 잘못된되거나 비효율적인 색인 정보를 제외하도록 지정합니다. 이러한 정보의 제거는 자원을 절약하고 보다 작은 HTML 출력 파일을 작성합니다. 이 키워드는 선택 사항이며 deepcheck 키워드와 함께 사용할 수 없습니다.

분석기 도구는 AS/400 플랫폼에서의 복제를 위해 PTF에서 사용할 수 있습니다. 이들 도구는 복제 환경에 대한 정보를 수집하고 문제점 판별을 돕기 위해 IBM 서비스 담당자에게 보낼 수 있는 HTML 파일을 작성합니다. AS/400 도구를 가져오려면, 적합한 PTF를 다운로드하십시오. 예를 들어, 제품 5769DP2의 경우, PTF SF61798이나 해당 최신 교체 PTF를 다운로드해야 합니다.

“문제점 해결” 절에 다음의 문제점과 해결책을 추가하십시오.

문제점: Apply 프로그램이 변경사항 복제 없이 루핑됩니다. Apply 트레일 테이블은 STATUS=2를 보여줍니다.

복사 작업 내역 세트에는 여러 소스 테이블이 포함됩니다. 세트의 한 소스 테이블의 중요점 처리를 향상하기 위해, 해당 소스 테이블에 대한 내부 CCD 테이블이 정의되지만, 다른 복사 작업 내역 세트에 있습니다. 소스

테이블 갱신이 이루어지지만 내부 CCD 테이블을 채우는 Apply 프로세스는 비동기로 수행됩니다(예를 들어, Apply 프로그램이 시작되지 않거나 이벤트가 트리거되지 않거나 할 수 있습니다). 소스 테이블에서 목표 테이블로 갱신 사항을 복제하는 Apply 프로그램은 내부 CCD 테이블이 갱신되기를 기다리기 때문에 루핑됩니다.

루핑을 중지하려면, 내부 CCD 테이블에 대해 Apply 프로그램을 시작하십시오(또는 복제를 초래하는 이벤트를 트리거하십시오) Apply 프로그램은 내부 CCD 테이블을 채우고 루핑 Apply 프로그램이 모든 소스 테이블의 변경사항을 처리하도록 합니다.

여러 Apply 프로그램이 상주하는 내부 CCD 테이블과 함께 소스 테이블이 있는 복사 작업 내역 세트에서 유사한 경우가 발생할 수 있습니다.

AS/400용 Capture 및 Apply

"작업 관리에 대한 주의사항"은 다음과 같아야 합니다.

기본 정의를 변경하거나 사용자 고유 정의를 제공할 수 있습니다. 자신의 서브시스템 설명을 작성하는 경우, 그 이름을 서브시스템 QZSNDPR로 지정하고, 이를 QDPR 이외의 라이브러리에서 작성해야 합니다. 해당 정의 변경에 대한 자세한 정보는 "OS/400 Work Management V4R3", SC41-5306을 참조하십시오.

"AS/400용 DB2 DataPropagator의 설치 확인 및 사용자 정의"에 다음의 내용을 추가하십시오.

트랜잭션의 수가 많아 잠금 경합에 문제가 발생할 경우, 기본 대기 시간종료 값을 30에서 120으로 증가시킬 수 있습니다. Capture 작업이 시작될 때마다 작업을 변경할 수 있고 또는 다음 프로시저어를 사용하여 서브시스템에서 수행되는 모든 작업의 시간종료 값을 변경할 수 있습니다.

1. 다음 명령을 발행하여 QGPL/QBATCH를 복제함으로써 새 클래스 오브젝트를 작성하십시오.

```
CRTDUPOBJ OBJ(QBATCH) FROMLIB(QGPL) OBJTYPE(*CLS) TOLIB(QDPR) NEWOBJ(QZSNDPR)
```

2. 새로 작성된 클래스의 시간종료 대기값을 변경하십시오(예를 들면 300초로).

```
CHGCLS CLS(QDPR/QZSNDPR) DFTWAIT(300)
```

3. 서브시스템 설명 QDPR/QZSNDPR에 있는 라우팅 항목을 갱신하여 새로 작성된 클래스를 사용하십시오.

```
CHGRTGE SBS(DQDPR/QZSNDPR) SEQNBR(9999) CLS(QDPR/QZSNDPR)
```

194 페이지의 "delete journal receiver exit 루틴 사용"에는 다음과 같은 문장이 포함되어야 합니다. delete journal receiver exit 루틴에 대한 등록을 제거할 경우 소스 테이블에 사용된 모든 저널에 DLTRCV(*NO)가 있어야 합니다.

195 페이지에서, ADDEXITPGM 명령 매개변수는 다음과 같아야 합니다.

```
ADDEXITPGM EXITPNT(QIBM_QJO_DLT_JRNRVC)
            FORMAT(DRCV0100)
            PGM(QDPR/QZSNDREP)
            PGMNBR(*LOW)
            CRTEXITPNT(*NO)
            PGM(DTA(65535 10 QSYS)
```

OS/390용 Capture 및 Apply

다음 단락이 갱신되었습니다.

OS/390용 DB2 DataPropagator의 전제조건

OS/390용 DB2 DataPropagator 버전 7(V7)을 수행하려면 OS/390용 DB2 버전 5, OS/390용 DB2 버전 6 또는 OS/390용 DB2 버전 7이 있어야 합니다.

OS/390에서의 유니코드 및 ASCII 코드화 체계

OS/390용 DB2 DataPropagator V7은 유니코드 및 ASCII 코드화 체계를 지원합니다. 새로운 코드화 체계를 사용하려면 OS/390용 DB2 V7이 필요하며 DB2 DataPropagator 소스, 목표 및 제어 테이블을 다음 절에서 설명하는 것처럼 수동으로 작성하거나 변환해야 합니다. 그러나 기존 복제 환경에서는 코드화 체계를 수정하지 않아도 OS/390용 DB2 DataPropagator V7로 작업할 수 있습니다.

코드화 체계 선택

소스, CD 및 목표 테이블에서 같은 코드화 체계를 사용하는 경우, 복제 환경에서 데이터 변환의 필요성을 최소화할 수 있습니다. 테이블에 대한 코드화 체계를 선택할 때 단일 CCSID 규칙을 따르십시오. 테이블 공간의 문자 데이터를 ASCII, 유니코드 또는 EBCDIC으로 코드화할 수 있습니다. 테이블 공간의 모든 테이블은 동일한 코드화 체계를 사용해야 합니다. SQL문의 모든 테이블 코드화 체계는 동일해야 합니다. 또한 뷰 및 조인에서 사용하는 모든 테이블은 동일한 코드화 체계를 사용해야 합니다.

단일 CCSID 규칙을 따르지 않으면 DB2에서 위반을 발견하여 바인드나 실행 시에 SQLCODE -873을 리턴합니다. ASCII 또는 유니코드를 어떤 테이블에 사용해야 하는지는 클라이언트/서버 구성에 따라 달라집니다. 특히 테이블에 대한 코드화 체계를 선택할 때 이 규칙을 따르십시오.

- OS/390용 DB2의 소스 또는 목표 테이블은 EBCDIC, ASCII 또는 유니코드일 수 있습니다. 이들 테이블은 지원되는 DBMS(DB2 계열 또는 DataJoiner가 있는 DB2가 아닌 DBMS)에서 동일하거나 다른 코드화 체계를 사용하는 테이블 간에 복사될 수 있습니다.
- OS/390용 DB2 소스 서버에서 한 서버의 모든 CD, UOW, 레지스터 및 제거 제어 테이블은 동일한 코드화 체계를 사용해야 합니다. 일관성 유지를 위해 항상 명시적으로 코드화 체계를 지정하십시오.
- 동일한 제어 서버의 모든 제어 테이블(ASN.IBMSNAP_SUBS_XXXX)은 동일한 코드화 체계를 사용해야 합니다.
- 다른 제어 테이블은 다른 코드화 체계를 사용할 수 있지만 ASN.IBMSNAP_CRITSEC 테이블은 EBCDIC을 사용하는 것이 좋습니다.

코드화 체계 설정

테이블에 대해 적절한 코드화 체계를 지정하려면 테이블을 생성할 때 사용한 SQL을 수정합니다.

- 적합한 코드화 체계를 사용하여 소스 및 목표 테이블을 작성하거나 기존 목표 및 소스 테이블의 코드화 체계를 변경하십시오. 기존 테이블의 코드화 체계를 변경하기 전에 Capture 및 Apply 프로그램을 중지하는 것이 좋으며 Capture 프로그램을 콜드 시동한 후 Apply 프로그램을 다시 시작합니다. 기존 테이블의 코드화 체계를 변경하려면 다음을 수행하십시오.

1. Reorg 유틸리티를 사용하여 기존 테이블을 복사합니다.
2. 기존 테이블을 삭제합니다.
3. 새로운 코드화 체계를 지정하는 테이블을 다시 작성합니다.
4. 로드 유틸리티를 사용하여 이전 데이터를 새 테이블로 로드합니다.

Reorg 및 로드 유틸리티에 대한 자세한 정보는 *DB2 Universal Database for OS/390 Utility Guide and Reference*를 참조하십시오.

- 적절한 코드화 체계를 사용하여 새로운 제어 테이블을 작성하거나 기존 테이블의 코드화 체계를 수정하십시오.

DPCNTL.MVS는 OS/390용 DB2의 sqlllib\samples\repl을 포함하고 있으며 제어 테이블을 작성하는 몇 개의 CREATE TABLE 문이 들어 있습니다. ASCII 또는 유니코드가 필요한 테이블(예: ASN.IBMSNAP_REGISTER 및 ASN.IBMSNAP_PRUNCNTL)의 경우, 다음 예와 같이 CCSID ASCII 또는 CCSID UNICODE 키워드를 추가하십시오.

```
CREATE TABLE ASN.IBMSNAP_PRUNCNTL (  
  TARGET_SERVER      CHAR( 18)          NOT NULL,  
  TARGET_OWNER      CHAR( 18)          NOT NULL,  
  TARGET_TABLE      CHAR( 18)          NOT NULL,  
  SYNCHTIME         TIMESTAMP,  
  SYNCHPOINT        CHAR( 10)          FOR BIT DATA,  
  SOURCE_OWNER      CHAR( 18)          NOT NULL,  
  SOURCE_TABLE      CHAR( 18)          NOT NULL,  
  SOURCE_VIEW_QUAL  SMALLINT          NOT NULL,  
  APPLY_QUAL        CHAR( 18)          NOT NULL,  
  SET_NAME          CHAR( 18)          NOT NULL,  
  CNTL_SERVER       CHAR( 18)          NOT NULL,  
  TARGET_STRUCTURE  SMALLINT          NOT NULL,  
  CNTL_ALIAS        CHAR( 8)           NOT NULL,  
) CCSID UNICODE  
  DATA CAPTURE CHANGES  
  IN TSSNAP02;
```

기존 제어 테이블 및 CD 테이블을 수정하려면 Reorg 및 로드 유틸리티를 사용하십시오.

- 새로운 복제 소스나 복사 작업 내역 세트를 작성할 경우, 관리 도구로 만든 SQL 과일을 수정하여 적절한 코드화 체계를 지정하십시오. SQL에는 복제 소스 및 복사 작

업 내역 세트용 CD 및 목표 테이블을 작성할 때 사용되는 몇 개의 CREATE TABLE 문이 있습니다. 적절한 위치에 CCSID ASCII 또는 CCSID UNICODE 키워드를 추가하십시오. 예를 들어, 다음과 같습니다.

```
CREATE TABLE user1.cdtable1 (  
    employee_name varchar,  
    employee_age decimal  
) CCSID UNICODE;
```

*DB2 UDB for OS/390 SQL Reference*에는 CCSID에 대한 자세한 내용이 있습니다.

UNIX 플랫폼용 Capture 및 Apply

UNIX 및 Windows용 Capture 및 Apply에 대한 환경 변수 설정

기본 코드 페이지 값 이외의 코드 페이지를 사용하여 소스 데이터베이스를 작성한 경우, DB2CODEPAGE 환경 변수를 해당 코드 페이지로 설정합니다. DB2CODEPAGE를 설정하기 전에 코드 페이지 값을 가져오는 자세한 정보는 *DB2 관리 안내서*를 참조하십시오. Capture는 데이터를 캡처할 데이터베이스와 같은 코드 페이지에서 수행해야 합니다. DB2는 Capture가 수행되고 있는 활동 중인 환경에서 Capture 코드 페이지를 가져옵니다. DB2CODEPAGE가 설정되지 않으면 DB2는 운영 체제에서 코드 페이지 값을 가져옵니다. 데이터베이스를 작성할 때 기본 코드 페이지를 사용한 경우, 운영 체제에서 가져온 값이 Capture에 대해 올바른 값입니다.

테이블 구조

339 페이지에서, 값 "2"에 대한 STATUS 컬럼 설명에 다음의 문장을 추가하십시오.

내부 CCD 테이블을 사용하고 Apply 트레일 테이블의 상태 컬럼에서 반복적으로 "2" 값을 받으면, "문제점 판별"로 이동하여 "문제점: Apply 프로그램이 변경사항을 복제하지 않고 루프합니다. Apply 트레일 테이블에서는 상태 -2를 표시합니다"를 참조하십시오.

Capture 및 Apply 메시지

ASN0017E 메시지는 다음과 같아야 합니다.

ASN0017E

Capture 프로그램에서 심각한 내부 오류가 발생했으며 올바른 오류 메시지를 발행할 수 없습니다. 루틴 이름은 "routine"입니다. 리턴 코드는 "return_code"입니다.

ASN1027S 메시지를 추가해야 합니다.

ASN1027S

대형 오브젝트(LOB) 컬럼이 너무 많이 지정되었습니다. 오류 코드는 "<error_code>"입니다.

설명: 복사 작업 내역 세트 구성원에 대해 대형 오브젝트(BLOB, CLOB 또는 DBCLOB) 컬럼이 너무 많이 지정되었습니다. 허용되는 최대 컬럼 수는 10입니다.

사용자 조치: 복사 작업 내역 세트 구성원에서 초과된 대형 오브젝트 컬럼을 제거하십시오.

ASN1048E 메시지는 다음과 같아야 합니다.

ASN1048E

Apply 순환 실행에 실패했습니다. 자세한 내용은 Apply 추적 테이블을 참조하십시오.
"<text>"

설명: Apply 순환에 실패했습니다. 메시지에서 "<text>"는 "<target_server>", "<target_owner, target_table, stmt_number>" 및 "<cntl_server>"를 식별합니다.

사용자 조치: 감사 추적 테이블의 APPERRM 필드를 확인하여 Apply 순환이 실패한 이유를 살펴보십시오.

응용프로그램 내에서 Capture 및 Apply 프로그램 시작

책의 399 페이지에서, Capture 및 Apply 프로그램을 시작하는 샘플 루틴의 주석에 몇 개의 오류가 나타나지만 샘플의 코드는 올바릅니다. 주석은 샘플의 뒤쪽 부분이 Capture 매개변수에 속하는 것으로 나타내지만 샘플의 뒤쪽 부분은 Apply 매개변수에 속합니다.

다음의 디렉토리에서 Apply 및 Capture API의 샘플과 각각의 makefile을 가져올 수 있습니다.

NT의 경우 - sql1lib\samples\repl
UNIX의 경우 - sql1lib/samples/repl

시스템 모니터 안내 및 참조서

db2ConvMonStream

사용 주의사항에서, 스냅샷 변수 데이터 스트림 유형 SQLM_ELM_SUBSECTION의 구조는 sqlm_subsection이어야 합니다.

문제점 해결 안내서

사용자가 로그인되지 않았을 때 Windows 95, Windows 98 및 Windows ME에서 DB2 시작

Windows 95, Windows 98 또는 Windows Millennium Edition(ME) 환경에서 db2start 명령이 성공하려면, 다음 중 하나를 수행해야 합니다.

- Windows 로그인 창이나 Microsoft 네트워킹 로그인 창을 사용하여 로그인하십시오
- **db2logon** 명령을 발행하십시오(186 페이지의 1에서 **db2logon** 명령에 대한 정보를 참조).

또한, 로그인 중에 지정되었거나 **db2logon** 명령에 대해 지정된 사용자 ID는 DB2의 요구사항을 만족시켜야 합니다(187 페이지의 2 참조).

db2start 명령이 시작되면, 먼저 사용자가 로그인되었는지 확인합니다. 사용자가 로그인된 경우, **db2start** 명령은 이 사용자의 ID를 사용합니다. 사용자가 로그인되지 않은 경우, **db2start** 명령은 **db2logon** 명령이 수행되었는지 확인하고 수행되었으면, **db2start** 명령은 **db2logon** 명령에 대해 지정된 사용자 ID를 사용합니다. **db2start** 명령이 유효한 사용자 ID를 찾을 수 없는 경우, 이 명령은 종료됩니다.

Windows 95, Windows 98 및 Windows ME에서 DB2 Universal Database 버전 7 설치 중에, 기본적으로 설치 소프트웨어는 시스템이 부트될 때 **db2start** 명령을 수행하는 단축 아이콘을 시작 폴더에 추가합니다(자세한 정보는 186 페이지의 1을 참조하십시오). 시스템의 사용자가 로그인되지도 않고 **db2logon** 명령을 발행하지도 않은 경우, **db2start** 명령은 종료됩니다.

사용자가 Windows나 네트워크에 정상적으로 로그인하지 않았으면, 다음과 같이 배치 파일에서 명령을 수행하여 **db2start** 명령보다 먼저 **db2logon** 명령을 발행하기 위한 요구사항을 숨길 수 있습니다.

1. **db2logon** 명령 다음에 **db2start.exe** 명령을 발행하는 배치 파일을 작성하십시오. 예를 들어, 다음과 같습니다.

```
@echo off
db2logon db2local /p:password
db2start
cls
exit
```

2. 배치 파일의 이름을 db2start.bat로 지정하고 DB2를 설치한 드라이브와 경로에 있는 /bin 디렉토리에 저장하십시오. 운영 체제가 배치 파일로의 경로를 찾을 수 있도록 이 위치에 배치 파일을 저장합니다.

DB2가 설치된 드라이브와 경로는 DB2 레지스트리 변수 DB2PATH에 저장됩니다. DB2를 설치한 드라이브와 경로를 찾으려면, 다음의 명령을 발행하십시오.

```
db2set -g db2path
```

db2set 명령이 c:\sql1lib 값을 리턴하는 것으로 가정하십시오. 이 경우, 다음과 같이 배치 파일을 저장하게 됩니다.

```
c:\sql1lib\bin\db2start.bat
```

3. 시스템이 부트될 때 DB2를 시작하려면, 시작 폴더의 단축 아이콘에서 배치 파일을 수행해야 합니다. 두 개의 옵션이 있습니다.

- **db2start.exe** 대신 배치 파일을 수행하도록 DB2 설치 프로그램이 작성한 단축 아이콘을 수정하십시오. 앞의 예에서, 단축 아이콘은 이제 db2start.bat 배치 파일을 수행합니다. DB2 설치 프로그램에 의해 작성된 단축 아이콘을 **DB2 - DB2.lnk**라고 하며 대부분의 시스템에서 c:\WINDOWS\Start Menu\Programs\Start\DB2 - DB2.lnk에 위치합니다.
- 배치 파일을 수행하는 자신의 단축 아이콘을 추가하고, DB2 설치 프로그램에 의해 추가된 단축 아이콘을 삭제하십시오. 다음의 명령을 사용하여 DB2 단축 아이콘을 삭제하십시오.

```
del "C:\WINDOWS\Start Menu\Programs\Startup\DB2 - DB2.lnk"
```

사용자 자신의 단축 아이콘을 사용하기로 결정한 경우, 단축 아이콘에 대해 종료 시 닫기 속성을 설정해야 합니다. 이 속성을 설정하지 않으면, DOS 명령 프롬프트가 **db2start** 명령이 성공적으로 완료된 이후에도 task 표시줄에 남아 있습니다. DOS 창이 **db2start** 프로세스 중에 열려 있는 것을 방지하려면, 최소화하여 수행하도록 설정된 이 단축 아이콘(및 이를 수행하는 DOS 창)을 작성할 수 있습니다.

주: 시스템을 부트하는 동안 DB2를 시작하는 대신, DB2를 사용하는 응용프로그램을 수행하기 전에 DB2를 시작할 수 있습니다. 자세한 내용은 188 페이지의 5을 참조하십시오.

db2start 명령이 수행되기 전에 배치 파일을 사용하여 **db2logon** 명령을 발행하고 사용자가 때때로 로그인하는 경우, **db2start** 명령은 계속해서 작동되며 DB2가 로그인한 사용자의 사용자 ID를 사용하는 것이 유일한 차이점이 됩니다. 참고 1에서 추가 세부 사항을 참조하십시오.

주:

1. **db2logon** 명령은 사용자 로그인을 시뮬레이트합니다. db2logon 명령의 형식은 다음과 같습니다.

```
db2logon userid /p:password
```

명령에 대해 지정된 사용자 ID는 DB2 이름 지정 요구사항을 만족시켜야 합니다(참고 2에서 자세한 내용 참조). 명령이 사용자 ID와 암호 없이 발행된 경우, 사용자에게 사용자 ID와 암호를 묻는 프롬프트가 표시된 창이 열립니다. 유일하게 제공되는 매개변수가 사용자 ID인 경우, 사용자에게 암호를 묻는 프롬프트를 표시하지 않으며 아래에 설명된 특정 상황 하에서는 암호가 필요합니다.

db2logon 명령에 의해 설정된 사용자 ID와 암호는 사용자가 Windows 로그인 창이나 Microsoft 네트워킹 로그인 창을 사용하여 로그인하지 않은 경우에만 사용됩니다. 사용자가 로그인했으며 **db2logon** 명령이 발행되었으면, **db2logon** 명령의 사용자 ID는 모든 DB2 조치에 사용되지만 **db2logon** 명령에 지정된 암호는 무시됩니다.

사용자가 Windows 로그인 창이나 Microsoft Networking 로그인 창을 사용하여 로그인하지 않은 경우, **db2logon** 명령을 통해 제공된 사용자 ID와 암호는 다음과 같이 사용됩니다.

- **db2start** 명령은 시작 시에 사용자 ID를 사용하며 암호를 요구하지 않습니다.
- 테이블 작성과 같은 조치에 고급 규정자가 누락된 경우, 사용자 ID를 고급 규정자로 사용합니다. 예를 들어, 다음과 같습니다.
 - a. `db2logon db2local`을 발행하십시오.
 - b. `create table tab1`을 발행하십시오.
테이블은 고급 규정자로 `db2local.tab1`을 사용하여 작성됩니다.

테이블과 다른 오브젝트의 스키마 이름과 동일한 사용자 ID를 사용해야 합니다.

- 시스템이 서버에 대한 클라이언트의 역할을 수행하며 사용자가 사용자 ID와 암호없이 CONNECT 명령문(예: `CONNECT TO TEST`)을 발행하고 인증이 server로 설정된 경우, **db2logon** 명령의 사용자 ID와 암호는 원격 서버에서 사용자의 유효성을 확인하는 데 사용됩니다. 사용자가 명시적 사용자 ID와 암호를 사용하여 연결하는 경우(예: `CONNECT TO TEST USER userID USING password`), CONNECT 명령문에 대해 지정된 값이 사용됩니다.
2. 버전 7에서, 로그인하는 데 사용되거나 **db2logon** 명령에 대해 지정된 사용자 ID는 다음 DB2 요구사항을 준수해야 합니다.
- USERS, ADMINS, GUESTS, PUBLIC, LOCAL 또는 SQL 참조서에 나열된 SQL 예약어가 아니어야 합니다.
 - SQL, SYS 또는 IBM으로 시작할 수 없습니다.
 - 문자에 다음이 포함될 수 있습니다.
 - A - Z(Windows 95, Windows 98 및 Windows ME는 대소문자 구별 사용자 ID를 지원합니다)
 - 0 - 9
 - @, # 또는 \$

3. 사용자 정의된 대화식 설치 중에 또는 응답 파일 설치를 수행하고 DB2. AUTOSTART=NO 옵션을 지정한 경우에 시작 폴더에서 **db2start** 단축 아이콘의 작성을 방지할 수 있습니다. 이들 옵션을 사용하는 경우, 시작 폴더에는 db2start 단축 아이콘이 없으며 db2start.bat 파일을 수행하기 위한 사용자 자신의 단축 아이콘을 추가해야 합니다.
4. **Windows 98** 및 **Windows ME**에서 Windows 98 또는 Windows ME가 시작되어 있으면 항상 로그인되는 사용자 ID를 지정하는 데 사용할 수 있는 옵션이 있습니다. 이 경우, Windows 로그인 창은 나타나지 않습니다. 이 옵션을 사용하는 경우, 사용자 ID가 DB2 요구사항을 만족시키면 사용자가 로그인되고 **db2start** 명령이 성공됩니다(참고 187 페이지의 2에서 자세한 내용 참조). 이 옵션을 사용하지 않으면, 사용자에게는 항상 로그인 창이 제시됩니다. 사용자가 로그인하지 않고 이 창을 취소하면, 위에 설명된 대로 **db2logon** 명령이 이전에 발행되지 않았거나 배치 파일에서 호출되지 않은 경우에 **db2start** 명령은 실패합니다.
5. 시스템 부트 중에 DB2를 시작하지 않으면, 응용프로그램이 DB2를 시작할 수 있습니다. DB2를 사용하는 응용프로그램 초기화의 일부로 db2start.bat 파일을 수행할 수 있습니다. 이 방법을 사용하면 DB2를 사용할 응용프로그램이 시작되었을 때에만 DB2가 시작됩니다. 사용자가 응용프로그램을 종료하면, **db2stop** 명령을 발행하여 DB2를 중단할 수 있습니다. DB2가 시스템 부트 중에 시작되지 않은 경우, 비즈니스 응용프로그램은 이러한 방식으로 DB2를 시작할 수 있습니다.

DB2 동기화 응용프로그램을 사용하거나 응용프로그램에서 동기화 API를 호출하려면, 실행하기 위해 다운로드된 스크립트에 지역 인스턴스나 지역 데이터베이스에 대해 작동되는 명령이 포함된 경우, DB2를 시작해야 합니다. 이들 명령은 데이터베이스 스크립트, 인스턴스 스크립트에 있거나 운영 체제(OS) 스크립트에 내재될 수 있습니다. OS 스크립트에 인스턴스나 데이터베이스를 사용하는 명령행 처리기 명령이나 DB2 API가 포함되지 않은 경우, DB2를 시작하지 않고 수행할 수 있습니다. 동기화 프로세스 중에 스크립트에서 수행될 명령을 미리 알려주기 어렵기 때문에 일반적으로 DB2는 동기화가 시작되기 전에 시작해야 합니다.

응용프로그램에서 **db2sync** 명령이나 동기화 API를 호출하는 경우, 응용프로그램의 초기화 중에 DB2를 시작하게 됩니다. 사용자가 Windows용 DB2 폴더의 DB2 동기화 단축 아이콘을 사용하여 동기화를 시작하는 경우, db2sync.bat 파일을 수행하도록 DB2 동기화 단축 아이콘을 수정해야 합니다. DB2가 동기화가 시작되기 전에 수행되도록 하려면, 배치 파일에 다음의 명령이 포함되어야 합니다.

```
@echo off
db2start.bat
db2sync.exe
db2stop.exe
cls
exit
```

이 예에서, db2start.bat 파일은 **db2logon** 및 **db2start** 명령을 위에 설명된 대로 호출하는 것으로 가정합니다.

응용프로그램이 시작되면 DB2를 시작하기로 결정한 경우, DB2의 설치가 시작 폴더에 DB2를 시작하기 위한 단축 아이콘을 추가하지 않는지 확인하십시오. 참고 188 페이지의 3에서 자세한 내용을 참조하십시오.

DB2 Universal Database Server 문제점 해결

『잠금 및 교착 상태』 절의 『응용프로그램이 느리거나 정지된 것처럼 보임』 서브섹션에서 『잠금 대기 또는 교착 상태가 다음 키 잠금으로 발생하지 않음』의 설명을 다음과 같이 바꾸십시오.

다음 키 잠금은 모든 INSERT 및 DELETE 문에 대한 다음 키와 SELECT 문의 결과 세트보다 더 높은 다음 키 값을 자동으로 잠워서 반복 읽기(RR)를 보장합니다. 색인의 키 부분을 변경하는 UPDATE 문의 경우, 원래 색인 키가 삭제되고 새 키 값이 삽입됩니다. 다음 키 잠금은 키 삽입과 키 삭제 시 모두 수행됩니다. ANSI 및 SQL92 표준 RR을 보장해야 하며 이는 DB2 기본값입니다.

응용프로그램에 대한 스냅샷 정보를 검사하십시오. 다음 키 잠금에 문제가 있으면 반복 읽기(RR) 동작에 의존하는 응용프로그램이 없고 검색에서 파악되지 않은 삭제를 건너뛸 수 있는 경우, DB2_RR_TO_RS 옵션을 설정할 수 있습니다.

DB2_RR_TO_RS가 설정되면 다음 키 잠금이 색인 키 삽입 및 삭제 시 수행되지 않으므로 사용자 테이블에서의 검색에 대한 RR 동작을 보장할 수 없습니다. 카탈로그 테이블은 이 옵션의 영향을 받지 않습니다.

동작의 다른 변경은 DB2_RR_TO_RS를 설정하는 것인데 행이 검색 규정에 맞지도 삭제되고 파악되지 않은 행을 검색할 경우에는 건너뛹니다.

예를 들어, 트랜잭션 A가 컬럼1=10인 행을 삭제하고 트랜잭션 B가 컬럼1>8이고 컬럼1<12인 행을 검색하는 시나리오를 생각해 봅니다.

DB2_RR_TO_RS를 해제하면 트랜잭션 B는 트랜잭션 A가 파악하거나 구간 복원할 때까지 기다립니다. 구간 복원하는 경우, 컬럼1=10인 행이 트랜잭션 B의 조회 결과 세트에 포함됩니다.

DB2_RR_TO_RS가 설정되면 트랜잭션 B는 트랜잭션 A가 파악하거나 구간 복원할 때까지 기다리지 않습니다. 삭제된 행을 포함하지 않은 조회 결과를 즉시 받습니다.

ANSI 및 SQL92 표준 RR이 필요한 경우나 검색 시 파악되지 않은 삭제를 건너뛰지 않을 경우에는 이 옵션을 사용하지 마십시오.

64 비트 플랫폼에서 DB2 Universal Database 사용

구성

LOCKLIST

다음 정보를 테이블 2에 추가해야 합니다.

매개변수	이전 상한	현재 상한
LOCKLIST	60000	524288

shmsys:shminfo_shmmax

64 비트 Solaris 운영 체제의 DB2 사용자들은 대형 데이터베이스 공유 메모리 세트를 할당할 수 있도록 /etc/system에 있는 "shmsys:shminfo_shmmax" 값을 필요한 만큼 증가해야 합니다. UNIX용 DB2 빠른 시작 책에서는 이 매개변수를 "바이트로 머신에서 물리적 RAM의 90%"로 설정하도록 권합니다. 이 권장사항은 64 비트 구현에서도 유효합니다.

그러나, UNIX용 DB2 빠른 시작 책에 있는 다음 권장사항은 잘못된 것입니다. 4 GB 이상의 RAM(Solaris 운영 체제에서 사용할 수 있는 전체 양의 최대 64 GB)이 있는 32 비트 시스템의 경우, 사용자가 shmmax 값을 4 GB 이상으로 설정하고 32 비트 커널을 사용하면, 커널은 숫자의 하위 32 비트만을 참조하므로, shmmax가 아주 작은 값이 되는 경우가 있습니다.

제한사항

현재 64 비트 운영 체제에서는 LDAP 지원이 없습니다.

32 비트 및 64 비트 데이터베이스는 같은 경로에서 작성될 수 없습니다. 예를 들어 32 비트 데이터베이스가 <somepath>에 있는 경우,

```
db2 create db <somedb> on <somepath>
```

64 비트 인스턴스에서 발행되면, "SQL10004C 데이터베이스 디렉토리에 액세스하는 동안 입출력 오류가 발생했습니다."와 함께 실패합니다.

XML Extender 관리 및 프로그래밍

IBM DB2 XML Extender에 대한 릴리스 정보는 DB2 XML 웹 사이트, <http://www-4.ibm.com/software/data/db2/extenders/xmlext/library.html>에서 찾을 수 있습니다.

MQSeries

이 절에서는 DB2 및 MQSeries를 사용하여 메시지 및 데이터베이스 액세스를 결합하는 응용프로그램을 구성하는 방법에 대해 설명합니다. 이 절에서는 사용자 정의 함수(UDF)와 비슷한 함수 집합을 중심으로 설명합니다. 이 함수 집합은 DB2 Universal Database, 버전 7.2에서 선택적으로 사용 가능합니다. 이 기본 함수를 사용하여 단순한 이벤트 통지에서부터 데이터 웨어하우스에 이르는 광범위한 응용프로그램을 지원할 수 있습니다.

데이터 웨어하우스 응용프로그램에 대한 자세한 정보는 256 페이지의 『MQSeries와 Data Warehouse Center 통합』에서 참조하십시오.

DB2 MQSeries 함수 설치 및 구성

이 절에서는 DB2 MQSeries 함수를 사용하기 위해 DB2 환경을 구성하는 방법에 대해 설명합니다. 다음 절차를 완료하면 SQL 내에서 DB2 MQSeries 함수를 사용할 수 있습니다. 이들 함수에 대한 설명은 릴리스 정보의 SQL 참조서 절에 있습니다. 최신 문서, 힌트 및 팁 등의 추가 정보는 <http://www.ibm.com/software/data/integration/MQSeries>에 있습니다.

DB2 MQSeries 함수를 구성하고 사용하기 위한 기본 절차는 다음과 같습니다.

1. MQSeries를 설치합니다.
2. MQSeries AMI를 설치합니다.
3. DB2 MQSeries 함수를 사용하여 구성합니다.

또한 DB2 MQSeries 함수에서 제공하는 publish/subscribe 기능을 사용하려면 MQSeries Integrator 또는 MQSeries Publish/Subscribe 함수도 설치해야 합니다. MQSeries Integrator에 대한 정보는 <http://www.ibm.com/software/ts/mqseries/integrator>에 있습니다. MQSeries Publish/Subscribe 기능에 대한 정보는 <http://www.ibm.com/software/ts/mqseries/txppacs>의 범주 3에 있습니다.

MQSeries 설치

첫 단계는 MQSeries 버전 5.2가 DB2 서버에 설치되어 있는지 확인하는 것입니다. 이 버전의 MQSeries가 이미 설치되어 있으면 다음 단계인 "MQSeries AMI 설치"로 건너뛰십시오. DB2 버전 7.2에는 DB2와 함께 사용할 MQSeries 서버의 사본이 들어 있습니다. MQSeries를 설치하거나 기존 MQSeries 설치를 업그레이드하기 위한 플랫폼

별 지침은 <http://www.ibm.com/software/ts/mqseries/library/manuals>의 플랫폼별 빠른 시작 책에 있습니다. 설치를 계속할 경우, 기본 대기행렬 관리 프로그램을 설정해야 합니다.

MQSeries AMI 설치

다음 단계는 MQSeries Application Messaging Interface(AMI)를 설치하는 것입니다. 이는 관리 및 프로그래밍 작업을 완전하게 분리하는 MQSeries 프로그래밍 인터페이스를 확장한 것입니다. DB2 MQSeries 함수를 사용하려면 이 인터페이스를 설치해야 합니다. MQSeries AMI가 DB2 서버에 이미 설치되어 있으면 다음 단계인 "DB2 MQSeries 함수 사용"으로 건너뛰십시오. MQSeries AMI가 설치되어 있지 않으면 DB2 7.2와 함께 제공된 설치 패키지에서 설치하거나 <http://www.ibm.com/software/ts/mqseries/txppacs>의 MQSeries Support Pacs 웹 사이트에서 복사하십시오. AMI는 "범주 3 - 제품 확장"에 있습니다. 사용자 편의를 위해 MQSeries AMI 사본을 DB2와 함께 제공하고 있습니다. 이 파일은 `sqlllib/cfg` 디렉토리에 있습니다. 파일 이름은 운영 체제에 따라 달라집니다.

AIX 버전 4.3 이상	ma0f_ax.tar.Z
HP-UX	ma0f_hp.tar.Z
Solaris 운영 환경	ma0f_sol7.tar.Z
Windows 32 비트	ma0f_nt.zip

압축 설치 이미지에 포함된 AMI readme 파일에서 설명하는 일반적인 AMI 설치 절차를 따르십시오.

DB2 MQSeries 함수 사용

이 단계에서 DB2 MQSeries 함수를 구성하고 사용하게 됩니다. **enable_MQFunctions** 유틸리티는 먼저 적절한 MQSeries 환경이 설정되었는지 확인한 후, DB2 MQSeries 함수에 대한 기본 구성을 설치 및 작성하고 이들 함수를 사용하여 지정된 데이터베이스를 사용 가능하게 하여 구성이 작동되는지 확인합니다.

1. Windows NT 또는 Windows 2000인 경우 단계 5로 진행하십시오.
2. UNIX에서 그룹 설정: UNIX에서 이 함수를 사용 가능하게 하려면 먼저 DB2 인스턴스 소유자(주로 `db2inst1`) 및 분리(`fenced`)된 UDF(주로 `db2fenc1`)와 연관된 사용자 ID를 MQSeries 그룹 `mqm`에 추가해야 합니다. DB2 함수가 MQSeries를 액세스하기 위해 필요합니다.
3. UNIX에서 DB2 환경 변수 설정: `AMT_DATA_PATH` 환경 변수를 DB2가 인식한 목록에 추가합니다. `$INSTHOME/sqlllib/profile.env` 파일을 편집하고 `AMT_DATA_PATH`를 `DB2ENVLIST`에 추가할 수 있습니다. `db2set` 명령을 사용할 수도 있습니다.

4. UNIX에서 데이터베이스 인스턴스 재시작: 변경한 환경 변수를 적용하려면 데이터베이스 인스턴스를 재시작해야 합니다.
5. UNIX일 경우에는 디렉토리를 \$INSTHOME/sql1lib/cfg로, Windows일 경우에는 %DB2PATH%/cfg로 변경합니다.
6. **enable_MQFunctions** 명령을 수행하여 DB2 MQSeries 함수용 데이터베이스를 구성하고 사용 가능하게 합니다. 이 명령에 대한 자세한 설명은 208 페이지의 『enable_MQFunctions』를 참조하십시오. 일반적인 몇 가지 예가 아래에 있습니다. 성공적으로 완료한 후 지정된 데이터베이스는 사용 가능하며 구성이 테스트됩니다.
7. 명령행 처리기를 사용하여 이 함수를 테스트하려면 사용 가능한 데이터베이스에 연결한 후 다음 명령을 실행합니다.

```
values DB2MQ.MQSEND('a test')
values DB2MQ.MQRECEIVE()
```

첫번째 명령문은 "a test" 메시지를 DB2MQ_DEFAULT_Q 대기행렬로 보내며 두 번째 명령문은 이 메시지를 다시 받습니다.

주: **enable_MQFunctions**을 수행하면 기본 MQSeries 환경이 설정됩니다. MQSeries 대기행렬 관리 프로그램 DB2MQ_DEFAULT_MQM 및 기본 대기행렬 DB2MQ_DEFAULT_Q가 작성됩니다. amt.xml, amthost.xml 및 amt.dtd 파일은 이 파일들이 AMT_DATA_PATH가 가리키는 디렉토리에 없는 경우, 작성됩니다. amthost.xml 파일이 없고 connectionDB2MQ의 정의가 포함되어 있지 않으면 적절한 정보와 함께 파일에 한 행이 추가됩니다. 원본 파일의 사본이 DB2MQSAVE.amthost.xml로 저장됩니다.

MQSeries 메시지 스타일

DB2 MQSeries 함수는 데이터그램, publish/subscribe(p/s) 및 요청/응답(r/r)의 세 가지 메시지 모델을 지원합니다.

데이터그램으로 전송된 메시지는 예상되는 응답없이 단일 목적지로 전송됩니다. p/s 모델에서는 하나 이상의 subscriber에게 메시지를 배포하는 publication 서비스에게 하나 이상의 publisher가 메시지를 보냅니다. 요청/응답은 데이터그램과 비슷하지만 보낸 사람이 응답을 받을 것으로 예상합니다.

메시지 구조

MQSeries 자체는 전송하는 메시지의 특정한 구조화를 위임하거나 지원하지 않습니다.

MQSeries Integrator(MQSI) 등의 다른 제품이 C, Cobol 또는 XML 문자열로 형성된 메시지를 지원합니다. MQSI 내의 구조화된 메시지는 메시지 저장소가 정의합니다. XML 메시지는 일반적으로 자체 설명적인 메시지 구조로서 저장소를 통해 관리될 수도 있습니다. 메시지는 구조화되지 않을 수도 있는데 이 경우 메시지 내용을 분석하고

구성하는 사용자 코드가 필요합니다. 이러한 메시지는 주로 반구조화되는데, 즉 메시지 내의 필드를 구분하기 위해 바이트 위치나 분리분자를 사용합니다. 이러한 반구조화된 메시지는 MQSeries 지원 마법사가 지원합니다. XML 메시지는 DB2 XML Extender의 새로운 일부 기능이 지원합니다.

MQSeries 기능 개요

MQSeries 함수 세트는 DB2 UDB 버전 7.2과 함께 제공되며 SQL문에 메시지 조작이 포함될 수 있습니다. 따라서 이런 지원은 데이터베이스 인터페이스를 사용하는 C, Java, SQ와 같은 지원되는 언어로 작성된 응용프로그램에서 사용할 수 있습니다. 아래 예는 SQL로 작성한 것입니다. 이 SQL은 다른 프로그래밍 언어에서 표준 방식으로 사용할 수 있습니다. 위에서 설명한 모든 MQSeries 메시지 스타일이 지원됩니다. MQSeries 함수에 대한 자세한 정보는 릴리스 정보의 SQL 참조서 절을 참조하십시오.

기본 구성에서 MQSeries 서버는 DB2와 함께 데이터베이스 서버 머신에 있습니다. MQSeries 함수는 DB2에 설치되며 MQSeries 서버에 대한 액세스를 제공합니다. DB2 클라이언트는 DB2 서버에 액세스할 수 있는 모든 머신에 설치될 수 있습니다. 여러 클라이언트가 데이터베이스를 통해 MQSeries 함수에 동시에 액세스할 수 있습니다. 제공한 함수로 DB2 클라이언트는 SQL문 내에서 메시지 조작을 수행할 수 있습니다. 이 메시지 조작으로 DB2 응용프로그램은 서로 통신하거나 다른 MQSeries 응용프로그램과 통신할 수 있습니다.

enable_MQFunctions 명령을 사용하여 MQSeries 함수에 대해 DB2 데이터베이스를 사용 가능하게 할 수 있습니다. 이 명령은 클라이언트 응용프로그램이 추가 관리 조치 없이 사용하는 단순한 기본 구성을 설정합니다. 자세한 설명은 **enable_MQFunctions** 및 **disable_MQFunctions**을 참조하십시오. 기본 구성을 사용하여 응용프로그램 프로그래머는 개발을 쉽게 시작할 수 있고 개발용으로 간단한 인터페이스를 만들 수 있습니다. 추가 기능은 필요에 따라 계속 구성할 수 있습니다.

예 1: 기본 구성으로 단순한 메시지를 보내는 SQL문은 다음과 같습니다.

```
VALUES DB2MQ.MQSEND('simple message')
```

이 SQL문은 simple message라는 메시지를 MQSeries 대기행렬 관리 프로그램 및 기본 구성에 지정된 대기행렬로 보냅니다.

MQSeries의 Application Messaging Interface(AMI)는 메시지 조치와 이 조치를 수행하는 방법에 대해 설명하는 정의를 명확하게 구분합니다. 이 정의는 외부 저장소 파일에 보존하며 AMI 관리 도구로 관리합니다. 따라서 AMI 응용프로그램을 개발하고 관리하기가 쉬워집니다. DB2와 함께 제공된 MQSeries 함수는 AMI MQSeries 인터페이스를 기준으로 합니다. AMI는 구성 정보를 저장하기 위해 AMI Repository라는 외부 구성 파일을 사용하도록 지원합니다. 기본 구성에는 DB2와 함께 사용하도록 구성된 MQSeries AMI Repository가 포함됩니다.

MQSeries AMI의 두 가지 주요 개념인 서비스 지점 및 규정이 DB2 MQSeries 함수로 전달됩니다. 서비스 지점은 메시지를 보내거나 받는 논리적인 끝점입니다. AMI 저장소에서 각 서비스 지점은 MQSeries 대기행렬 이름 및 대기행렬 관리 프로그램과 함께 정의됩니다. 규정은 주어진 메시지 조작에 사용해야 하는 서비스 옵션의 질을 정의합니다. 서비스의 주요 질에는 메시지 우선순위 및 지속성이 포함됩니다. 기본 서비스 지점 및 규정 정의가 제공되며 개발 프로그램에서 응용프로그램을 더욱 단순하게 만들기 위해 사용될 수 있습니다. 예 1은 기본 서비스 지점 및 규정 이름을 명시적으로 지정하기 위해 다음과 같이 다시 작성할 수 있습니다.

예 2:

```
VALUES DB2MQ.MQSEND('DB2.DEFAULT.SERVICE', 'DB2.DEFAULT.POLICY', 'simple message')
```

대기행렬 및 응용프로그램이 있는 서버에서 하나 이상의 응용프로그램이 대기행렬에 서비스를 제공할 수 있습니다. 많은 구성에서 다른 응용프로그램 및 목적을 지원하기 위해 복수의 대기행렬을 정의합니다. 이러한 이유로 MQSeries 요청을 작성할 때 서로 다른 서비스 지점을 정의하는 것이 중요합니다. 다음은 이러한 예를 보여줍니다.

예 3:

```
VALUES DB2MQ.MQSEND('ODS_Input', 'simple message')
```

주: 이 예에서는 규정이 지정되지 않았으므로 기본 규정이 사용됩니다.

제한사항

MQSeries는 메시지 조작과 데이터베이스 조작을 원자 트랜잭션인 단일 작업 단위(UOW)로 결합하는 기능을 제공합니다. 이 기능이 처음부터 Unix 및 Windows의 MQSeries 함수에 의해 지원되는 것은 아닙니다.

송수신 함수를 사용할 때 메시지의 최대 길이는 4000문자입니다. 이는 MQPublish를 사용하여 메시지를 발행할 때의 최대 메시지 크기이기도 합니다.

오류 코드

MQSeries 함수가 리턴한 리턴 코드가 MQSeries Application Messaging Interface Manual의 부록 B에 나와 있습니다.

사용 시나리오

MQSeries 함수를 다양한 시나리오로 사용할 수 있습니다. 이 절에서는 기본 메시지, 응용프로그램 연결성 및 데이터 책을 포함한 일반적인 시나리오를 몇 가지 검토합니다.

기본 메시지

모든 데이터베이스 응용프로그램이 같은 DB2 서버에 연결되어 있을 때 MQSeries DB2 함수와의 가장 기본적인 형태의 메시지 교환이 이루어집니다. 클라이언트는 데이터베이스 서버에 있거나 네트워크 환경에 분산되어 있습니다.

간단한 시나리오에서 클라이언트 A는 MQSEND 함수를 호출하여 사용자 정의 문자열을 기본 서비스 위치로 보냅니다. 그러면 MQSeries 함수가 데이터베이스 서버의 DB2에서 실행됩니다. 잠시 후에 클라이언트 B가 MQRECEIVE 함수를 호출하여 기본 서비스에서 정의한 대기행렬 헤드에서 메시지를 제거하여 이를 클라이언트에게 리턴합니다. 이 작업을 수행할 MQSeries 함수가 다시 DB2에 의해 실행됩니다.

데이터베이스 클라이언트는 간단한 메시지를 다양한 방법으로 사용할 수 있습니다. 일반적으로 다음과 같은 경우에 메시징을 사용합니다.

- 데이터 콜렉션 -- 하나 이상의 다양한 정보 소스에서 메시지 형태로 정보를 받습니다. 정보 소스는 SAP나 사내에서 개발된 응용프로그램과 같은 상업용 응용프로그램입니다. 대기행렬에서 이러한 데이터를 받아서 추가로 처리하거나 분석하기 위해 데이터베이스 테이블에 저장합니다.
- 워크로드 분산 -- 작업 요청이 동일한 응용프로그램의 여러 인스턴스가 공유하는 대기행렬에 포스트됩니다. 인스턴스가 일부 작업을 수행할 준비가 되면 수행할 작업 요청이 있는 대기행렬의 최상단에서 메시지를 받습니다. 이러한 기법을 사용하면 요청으로 채워진 단일 대기행렬로 표현되는 워크로드를 여러 인스턴스가 공유할 수 있습니다.
- 응용프로그램 신호 -- 여러 프로세스가 공동 작업하는 경우, 서로 협조하기 위해 메시지를 주로 사용합니다. 이 메시지에는 수행할 작업에 대한 명령이나 요청이 포함됩니다. 일반적으로 이러한 종류의 신호는 단방향입니다. 즉, 메시지를 보내는 개시자가 응답을 기대하지 않습니다. 자세한 정보는 204 페이지의 『요청/응답 통신』을 참조하십시오.
- 응용프로그램 통지 -- 통지는 데이터를 보내는 개시자가 응답을 기대하지 않는다는 점에서 신호와 비슷합니다. 그러나 일반적으로 통지에는 발생한 비즈니스 이벤트에 대한 데이터가 포함됩니다. 205 페이지의 『Publish/Subscribe』는 통지의 개선된 형태입니다.

다음 시나리오는 원격 메시지를 통합하기 위해 위에서 설명한 단순한 시나리오를 확장한 것입니다. 즉, 머신 A와 머신 B 사이에서 메시지가 전송됩니다. 각 단계는 다음과 같습니다.

1. DB2 클라이언트가 머신 B의 원격 대기행렬을 나타내기 위해 정의된 대상 서비스를 지정하는 MQSEND 호출을 실행합니다.
2. MQSeries DB2 함수는 메시지를 보내는 실제 MQSeries 작업을 수행합니다. 머신 A의 MQSeries 서버는 메시지를 받아들이고 이 메시지가 머신 A의 현재 MQSeries 구성 및 서비스 지점 정의로 정의한 목적지로 배달되도록 보장합니다. 서

머신 B의 대기행렬임을 파악한 후 메시지를 머신 B의 MQSeries 서버로 배달하려고 시도합니다. 이 때 필요하면 투명하게 재시도합니다.

3. 머신 B의 MQSeries서버는 머신 A의 서버에서 보낸 메시지를 받고 이 메시지를 머신 B의 목적지 대기행렬에 갖다 놓습니다.
4. 머신 B의 MQSeries 클라이언트는 대기행렬의 헤드에서 메시지를 요청합니다.

메시지 전송

MQSEND를 사용하여 DB2 사용자나 개발자는 전송할 데이터, 전송할 위치 및 전송 시기를 선택합니다. 업계에서는 이를 "보내고 잊어버리기" 즉, 보낸 사람은 단지 메시지를 보내기만 하고 메시지가 목적지에 도달했는지 확인하는 것은 MQSeries 배달 프로토콜이라고 합니다. 다음은 이를 보여주는 예입니다.

예 4: 사용자 정의 문자열을 서비스 지점 myplace로 highPriority 규정과 함께 보냅니다.

```
VALUES DB2MQ.MQSEND('myplace','highPriority','test')
```

여기서 highPriority 규정은 MQSeries 우선순위를 가장 높은 레벨로 설정하고 지속성과 같은 서비스 질도 동일하게 조정하는 AMI Repository에 정의된 규정을 참조합니다.

메시지 내용은 SQL 및 사용자 지정 데이터의 적법한 결합으로 구성될 수 있습니다. 여기에는 중첩된 함수, 연산자 및 캐스트가 포함됩니다. 예를 들어, VARCHAR 컬럼 LASTNAME, FIRSTNAME 및 DEPARTMENT이 있는 EMPLOYEE 테이블에서 DEPARTMENT 5LGA의 각 직원에 대해 이 정보를 포함하는 메시지를 보내려면 다음을 수행해야 합니다.

예 5:

```
SELECT DB2MQ.MQSEND(LASTNAME || ' ' || FIRSTNAME || ' ' || DEPARTMENT)
FROM EMPLOYEE
WHERE DEPARTMENT = '5LGA'
```

이 테이블에 정수 AGE 컬럼도 있으면 다음이 포함됩니다.

예 6:

```
SELECT DB2MQ.MQSEND(LASTNAME || ' ' || FIRSTNAME || ' ' || DEPARTMENT || ' ' || char(AGE))
FROM EMPLOYEE
WHERE DEPARTMENT = '5LGA'
```

마지막으로 다음 예는 유효한 SQL 표현식을 사용하여 메시지 내용을 추출하는 방법을 보여줍니다. 두 번째 테이블 DEPT에는 varchar 컬럼 DEPT_NO 및 DEPT_NAME이 있고 직원의 LASTNAME 및 DEPT_NAME을 포함하는 메시지를 보낼 수 있습니다.

예 7:

```
SELECT DB2MQ.MQSEND(e.LASTNAME || ' ' || d.DEPTNAME) FROM EMPLOYEE e, DEPT d
WHERE e.DEPARTMENT = d.DEPTNAME
```

메시지 검색

MQSeries DB2 함수를 사용하여 메시지를 받거나 읽을 수 있습니다. 읽는 것과 받는 것의 차이는 읽는 경우 대기행렬의 헤드에서 메시지를 제거하지 않고 메시지를 리턴하지만 받는 경우에는 메시지가 대기행렬에서 제거됩니다. 받는 작업을 사용하여 검색된 메시지는 한 번만 검색할 수 있지만 읽는 작업을 사용하여 검색된 메시지는 여러 번 검색할 수 있습니다. 다음은 이를 보여주는 예입니다.

예 8:

```
VALUES DB2MQ.MQREAD()
```

이 예에서는 기본 서비스 질 규정을 사용하는 기본 서비스가 정의한 대기행렬의 헤드에서 메시지를 포함하는 VARCHAR 문자열을 리턴합니다. 읽을 메시지가 없을 경우, 널(NULL) 값이 리턴된다는 점에 주의하십시오. 이 작업으로 대기행렬은 변경되지 않습니다.

예 9:

```
VALUES DB2MQ.MQRECEIVE('Employee_Changes')
```

위의 예는 기본 규정을 사용하는 Employee_Changes 서비스가 정의한 대기행렬의 헤드에서 메시지를 제거하는 방법을 보여줍니다.

DB2의 강력한 기능 중 하나는 사용자 정의(또는 DB2 제공) 함수에서 테이블을 생성하는 기능입니다. 이 테이블 함수 기능을 이용하여 대기행렬의 내용을 DB2 테이블로 구체화할 수 있습니다. 다음은 이를 보여주는 간단한 예입니다.

예 10:

```
SELECT t.* FROM table ( DB2MQ.MQREADALL()) t
```

이 조회는 이 메시지에 대한 메타데이터 및 기본 서비스에서 정의한 대기행렬의 모든 메시지로 구성된 테이블을 리턴합니다. 리턴된 테이블 구조의 전체 정의는 부록에 정의되어 있지만 첫 컬럼은 메시지 내용을 반영하며 나머지 컬럼에는 메타데이터가 포함됩니다. 메시지만 리턴하려면 위의 예를 다음과 같이 바꿀 수 있습니다.

예 11:

```
SELECT t.MSG FROM table (DB2MQ.MQREADALL()) t
```

테이블 함수가 리턴한 테이블은 데이터베이스에서 직접 검색한 테이블과 다르지 않습니다. 따라서 다양한 방법으로 이 테이블을 사용할 수 있습니다. 예를 들어, 테이블 내용을 다른 테이블과 조인하거나 대기행렬의 메시지 수를 셀 수 있습니다.

예 12:

```

SELECT t.MSG, e.LASTNAME
FROM table (DB2MQ.MQREADALL() ) t, EMPLOYEE e
WHERE t.MSG = e.LASTNAME

```

예 13:

```

SELECT COUNT(*) FROM table (DB2MQ.MQREADALL()) t

```

테이블 함수에 대한 뷰를 작성하여 테이블 소스가 대기행렬이라는 것을 숨길 수도 있습니다. 예를 들어, 다음 예에서는 NEW_EMPLOYEES 서비스가 참조하는 대기행렬에 대해 NEW_EMP라는 뷰를 작성합니다.

예 14:

```

CREATE VIEW NEW_EMP (msg) AS
SELECT t.msg FROM table (DB2MQ.MQREADALL()) t

```

이 경우 뷰는 전체 메시지를 포함하는 단일 컬럼과 함께 정의됩니다. 고정된 길이의 두 필드가 포함된 경우와 같이 메시지가 단순하게 구성되면 메시지를 두 개의 컬럼으로 분석하기 위해 DB2 내장 함수를 사용하는 것이 편합니다. 예를 들어, 특정 대기행렬로 전송되는 메시지는 항상 18문자의 성과 그 뒤에 18문자의 이름이 나온다는 것을 알고 있는 경우, 다음과 같이 별도의 컬럼으로 각 필드를 포함하고 있는 뷰를 정의할 수 있습니다.

예 15:

```

CREATE VIEW NEW_EMP2 AS
SELECT left(t.msg,18) AS LNAME, right(t.msg,18) AS FNAME
FROM table(DB2MQ.MQREADALL()) t

```

DB2 Stored Procedure Builder의 새로운 기능인 MQSeries 지원 마법사를 사용하여 분리된 메시지 구조를 컬럼으로 맵핑하는 새로운 DB2 테이블 함수 및 뷰를 작성할 수 있습니다.

이는 하나 이상의 메시지 내용을 데이터베이스로 저장할 경우에 좋습니다. 메시지를 내용을 조작하고 저장하는 SQL의 전체 성능을 사용하여 수행할 수 있습니다. 다음은 가장 간단한 예입니다.

예 16:

```

INSERT INTO MESSAGES
SELECT t.msg FROM table (DB2MQ.MQRECEIVEALL()) t

```

하나의 컬럼 VARCHAR(2000)가 있는 MESSAGES 테이블에서 위의 명령문을 사용하면 기본 서비스 대기행렬의 메시지를 테이블에 삽입합니다. 이 방법을 수정하여 다양한 경우에 사용할 수 있습니다.

응용프로그램간 연결성

응용프로그램 통합은 많은 솔루션에서 일반적인 요소입니다. 구입한 응용프로그램을 기존 구조에 통합하거나 새로 개발한 응용프로그램만 기존 환경에 통합할 경우, 모두 이 기종 서브시스템 콜렉션이 전체적으로 작업할 수 있도록 함께 통합해야 하는 문제에 직면하게 됩니다. MQSeries는 일반적으로 응용프로그램을 통합하는 필수적인 도구입니다. 대부분의 하드웨어, 소프트웨어 및 언어 환경에서 액세스할 수 있는 MQSeries는 이 기종 응용프로그램 콜렉션을 서로 연결하는 방법을 제공합니다.

이 절에서는 몇 개의 응용프로그램 통합 시나리오와 DB2와 함께 사용하는 방법에 대해 설명합니다. 주제가 광범위하므로 응용프로그램 통합에 대한 포괄적인 접근은 이 범위에 포함되지 않습니다. 따라서 요청/응답 통신과 MQSeries Integrator 및 Publish/Subscribe의 두 가지 간단한 항목에 초점을 맞춥니다.

요청/응답 통신

요청/응답(R/R) 통신 방법은 하나의 응용프로그램이 다른 응용프로그램의 서비스를 요청하는 일반적인 방법입니다. 이를 수행하는 한 방법은 리퀘스터가 서비스 제공자에게 수행할 작업을 요청하는 메시지를 보내는 것입니다. 작업이 완료되면 제공자는 결과(또는 완료 확인만)를 다시 리퀘스터에게 보낼 수 있습니다. 위에서 설명한 기본 메시지 전송 방법 이외에는 보낸 사람의 요청을 서비스 제공자의 응답과 연결하는 방법이 없습니다. 계속 진행하기 전에 리퀘스터가 응답을 기다리지 않는 경우, 각 응답을 리퀘스터와 연결하는 메커니즘이 사용되어야 합니다. 개발자가 이러한 메커니즘을 만들 필요 없이 MQSeries는 교환 시 메시지 상관을 허용하는 상관 ID를 제공합니다.

여러 가지 방법으로 이 메커니즘을 사용할 수 있지만 가장 간단한 방법은 리퀘스터가 다음과 같은 방법 등을 사용하여 알려진 상관 ID로 메시지를 표시하는 것입니다.

예 17:

```
DB2MQ.MQSEND ('myRequester','myPolicy','SendStatus:cust1','Req1')
```

이 명령문은 요청의 상관 ID를 나타내기 위해 위의 MQSEND 명령문에 마지막 매개 변수 Req1을 추가합니다.

이 특정 요청에 대한 응답을 받으려면 다음과 같이 해당되는 MQRECEIVE 문을 사용하여 이 상관과 일치하는 표시된 서비스가 정의한 첫 메시지를 선택적으로 검색합니다.

예 18:

```
DB2MQ.MQRECEIVE('myReceiver','myPolicy','Req1')
```

요청을 서비스하는 응용프로그램이 사용 중이거나 리퀘스터가 응답을 받기 전에 위의 MQRECEIVE를 실행하면 이 상관 ID와 일치하는 메시지를 찾을 수 없습니다.

서비스 요청 및 상관 ID를 모두 받으려면 다음과 같은 명령문을 사용합니다.

예 19:

```
SELECT msg, correlid FROM table (DB2MQ.MQRECEIVEALL('aServiceProvider','myPolicy',1)) t
```

이 명령문은 서비스 aServiceProvider에서 첫번째 요청의 상관 ID 및 메시지를 리턴합니다.

일단 서비스가 수행되면 aRequester가 설명하는 대기행렬로 응답 메시지를 보냅니다. 그동안 서비스 리퀘스터는 다른 작업을 할 수 있습니다. 실제로 설정된 시간 내에 초기 서비스 요청에 대한 응답을 받는다는 보장이 없습니다. 이와 같은 응용프로그램 레벨 시간종료는 개발자가 관리해야 합니다. 리퀘스터는 응답을 발견하기 위해 폴링해야 합니다.

이러한 시간과 무관한 비동기 처리의 장점은 리퀘스터 및 서비스 제공자가 서로 완전히 독립적으로 실행할 수 있다는 점입니다. 또한 응용프로그램이 간헐적으로만 연결되는 환경 및 여러 요청이나 응답이 처리되기 전에 누적되는 배치 위주의 환경에서 사용될 수 있습니다. 이러한 누적은 데이터 웨어하우스나 조작 데이터 저장소를 정기적으로 갱신하는 데이터 웨어하우스 환경에서 주로 사용됩니다.

Publish/Subscribe

단순 데이터 publication: 응용프로그램 통합의 또 다른 일반적인 시나리오는 한 응용프로그램이 다른 응용프로그램에게 이벤트를 통지하는 것입니다. 이는 다른 응용프로그램이 모니터링하는 대기행렬로 메시지를 보냄으로써 쉽게 수행할 수 있습니다. 메시지 내용은 사용자 정의 문자열이나 데이터베이스 컬럼으로 구성될 수 있습니다. 단순한 메시지인 경우, 주로 MQSEND 함수를 사용해서 보내야 합니다. 이러한 메시지를 여러 수신자에게 동시에 보내야 할 경우, MQSeries AMI의 분산 목록 기능을 사용할 수 있습니다.

분산 목록은 AMI 관리 도구로 정의됩니다. 분산 목록은 개별 서비스 목록으로 구성됩니다. 분산 목록으로 보낸 메시지는 목록에서 정의된 모든 서비스로 전달됩니다. 이는 일부 서비스가 항상 모든 메시지와 관련이 있다는 것을 알고 있을 경우 특히 유용합니다. 다음 예는 분산 목록 interestedParties에 메시지를 보내는 경우입니다.

예 20:

```
DB2MQ.MQSEND('interestedParties','information of general interest');
```

특정 서비스를 받아야 하는 메시지를 더 잘 제어해야 할 경우, Publish/Subscribe 기능이 필요합니다. Publish/Subscribe 시스템은 일반적으로 여러 subscribers에게 메시지를 받기 위해 여러 subscribers가 등록할 수 있는 확장 가능한 보안 환경을 제공합니다. 이 기능을 지원하기 위해 MQSeries Integrator 또는 MQSeries Publish/Subscribe 기능과 함께 MQPublish 인터페이스를 사용할 수 있습니다.

MQPublish를 사용하여 사용자는 선택적으로 메시지와 연관된 주제를 지정할 수 있습니다. 주제를 사용하여 subscriber는 받을 메시지를 더욱 명확하게 지정할 수 있습니다. 다음과 같은 단계를 수행합니다.

1. MQSeries 관리자가 MQSeries Integrator publish/subscribe 기능을 구성합니다.
2. 관심있는 응용프로그램은 MQSI 구성에서 정의한 subscription 지점에 subscribe하여 선택적으로 관심있는 주제를 지정합니다. 각 subscriber는 관련 주제를 선택하고 MQSeries Integrator V2의 내용에 기초한 subscription을 이용할 수도 있습니다. 서비스 이름으로 표시된 대기행렬이 subscriber를 정의함에 주의하십시오.
3. DB2 응용프로그램은 서비스 지점 Weather에 메시지를 발행합니다. 메시지는 weather가 Austin 주제의 경우 Sleet임을 나타내므로 관심있는 subscribers에게 Austin의 weather가 Sleet임을 통지합니다.
4. 실제로 메시지를 발행하는 메커니즘은 DB2가 제공하는 MQSeries 함수에 의해 처리됩니다. 메시지는 Weather라는 이름의 서비스를 사용하여 MQSeries Integrator로 보내집니다.
5. MQSI는 Weather 서비스에서 메시지를 받아 MQSI 구성에서 정의한 처리를 수행하며 충족시키는 subscriptions를 판별합니다. 그런 다음 MQSI는 메시지를 기준에 맞는 subscriber 대기행렬에 전달합니다.
6. Weather 서비스에 subscribe하고 Austin에 관심이 있다고 등록한 응용프로그램은 수신하는 서비스에서 Sleet 메시지를 받습니다.

모든 기본값 및 널(NULL) 주제를 사용하여 이 데이터를 발행하려면 다음 명령문을 사용해야 합니다.

예 21:

```
SELECT DB2MQ.MQPUBLISH(LASTNAME || ' ' || FIRSTNAME || ' ' || DEPARTMENT || ' ' ||char(AGE))
FROM EMPLOYEE
WHERE DEPARTMENT = '5LGA'
```

모든 매개변수를 완벽하게 지정하고 LASTNAME만 포함하도록 메시지를 단순화하면 명령문은 다음과 같습니다.

예 22:

```
SELECT DB2MQ.MQPUBLISH('HR_INFO_PUB', 'SPECIAL_POLICY', LASTNAME,
'ALL_EMP:5LGA', 'MANAGER')
FROM EMPLOYEE
WHERE DEPARTMENT = '5LGA'
```

이 명령문은 SPECIAL_POLICY 서비스를 사용하여 메시지를 HR_INFO_PUB publication 서비스에 발행합니다. 메시지는 보낸 사람이 MANAGER 주제임을 나타냅니다. 주제 문자열은 ‘:’를 사용하여 연결한 여러 주제를 지정할 수 있음을 보여줍니다. 이 예에서 두 개의 주제를 사용함으로써 subscribers는 ALL_EMP 또는 단지 5LGA에만 등록하여 이 메시지를 받을 수 있습니다.

발행된 메시지를 받으려면 먼저 주어진 주제를 포함하는 메시지에 대한 관련 사항을 등록하고 메시지를 받아야 하는 subscriber 서비스 이름을 표시해야 합니다. AMI subscriber 서비스는 중개인 서비스 및 받는 사람 서비스를 정의합니다. 중개인 서비스는 subscriber가 publish/subscribe 중개인과 통신하는 방법이며 받는 사람 서비스는 subscription 요청과 일치하는 메시지를 보낼 위치입니다. 다음 명령문은 ALL_EMP 주제에 대한 관련 사항을 등록합니다.

예 23:

```
DB2MQ.MQSUBSCRIBE('aSubscriber', 'ALL_EMP')
```

일단 응용프로그램이 subscribe하면 ALL_EM 주제와 함께 발행된 메시지가 subscriber 서비스에 의해 정의된 받는 사람 서비스로 전달됩니다. 응용프로그램은 동시에 여러 개의 subscriptions를 가질 수 있습니다. subscription에 맞는 메시지를 받기 위해 표준 메시지 검색 함수를 사용할 수 있습니다. 예를 들어 subscriber 서비스 aSubscriber가 받는 사람 서비스를 aSubscriberReceiver로 정의하면 다음 명령문은 첫번째 메시지를 손상시키지 않으면서 읽습니다.

예 24:

```
DB2MQ.MQREAD('aSubscriberReceiver')
```

메시지 및 메시지가 발행될 주제를 파악하기 위해 테이블 함수 중 하나를 사용할 수 있습니다. 다음 명령문은 aSubscriberReceiver에서 처음 다섯 개의 메시지를 받아 메시지 및 주제를 모두 표시합니다.

예 25:

```
SELECT t.msg, t.topic FROM table (DB2MQ.MQRECEIVEALL('aSubscriberReceiver',5)) t
```

주제가 ALL_EMP인 모든 메시지를 읽기 위해 SQL의 기능을 이용하여 다음을 실행할 수 있습니다.

예 26:

```
SELECT t.msg FROM table (DB2MQ.MQREADALL('aSubscriberReceiver')) t  
WHERE t.topic = 'ALL_EMP'
```

주: MQRECEIVEALL이 제한조건과 함께 사용되면 ALL_EMP 주제로 발행된 메시지뿐만 아니라 전체 대기행렬이 소비됨에 주의해야 합니다. 이는 제한조건이 적용되기 전에 테이블 함수가 수행되기 때문입니다.

특정 주제에 subscribe하는 데 더 이상 관련 사항이 없으면 다음과 같은 명령문을 사용하여 명시적으로 unsubscribe해야 합니다.

예 27:

```
DB2MQ.MQUNSUBSCRIBE('aSubscriber', 'ALL_EMP')
```

일단 이 명령문이 실행되면 publish/subscribe 중개인은 이 subscription과 일치하는 메시지를 더 이상 배달하지 않습니다.

자동 publication: 데이터베이스 메시지의 또 다른 중요한 기법은 자동 publication입니다. DB2의 트리거 기능을 사용하여 트리거 호출의 일부로 메시지를 자동으로 발행할 수 있습니다. 자동 데이터 publication을 위한 다른 방법들도 있지만 트리거에 기반한 접근 방법을 사용하면 관리자나 개발자가 메시지 내용을 더 자유롭게 구성할 수 있으며 트리거 조치를 더 융통성 있게 정의할 수 있습니다. 트리거를 사용할 경우에는 빈도 및 실행 비용에 주의해야 합니다. 다음은 MQSeries DB2 함수와 함께 트리거를 사용하는 방법을 보여주는 예입니다.

아래 예는 새 직원이 고용될 때마다 쉽게 메시지를 발행하는 방법을 보여줍니다. NEW_EMP에 대한 관심을 등록하고 HR_INFO_PUB 서비스에 subscribe한 사용자나 응용프로그램은 각각의 새 직원의 날짜, 이름 및 부서를 포함하는 메시지를 받게 됩니다.

예 28:

```
CREATE TRIGGER new_employee AFTER INSERT ON employee REFERENCING NEW AS n
FOR EACH ROW MODE DB2SQL
VALUES DB2MQ.MQPUBLISH('HR_INFO_PUB&', 'NEW_EMP',
current date || ' ' || LASTNAME || ' ' || DEPARTMENT)
```

enable_MQFunctions

enable_MQFunctions

지정된 데이터베이스에 대한 DB2 MQSeries 함수를 사용 가능하게 하고 DB2 MQSeries 함수가 적절히 실행될 수 있음을 확인합니다. MQSeries 및 MQSeries AMI 가 설치되거나 구성되어 있지 않으면 명령은 실패합니다.

권한 부여

다음 중 하나입니다.

- *sysadm*
- *dbadm*
- 데이터베이스의 IMPLICIT_SCHEMA(함수의 명시적 또는 내재된 스키마 이름이 없는 경우)
- 스키마에서 CREATEIN 특권(스키마 이름 DB2MQ가 있는 경우)

명령 구문

```
▶ enable_MQFunctions -n database -u userid -p password [force]
[noValidate]
```

명령 매개변수

-n database

사용 가능한 데이터베이스 이름을 지정합니다.

-u userid

데이터베이스에 연결할 사용자 ID를 지정합니다.

-p password

사용자 ID의 암호를 지정합니다.

-force 재설치 시 발생한 경고를 무시하도록 지정합니다.

-noValidate

SDB2 MQSeries 함수가 유효화되지 않도록 지정합니다.

예

다음 예에서는 DB2MQ 함수가 작성됩니다. 사용자는 SAMPLE 데이터베이스에 연결합니다. 기본 스키마 DB2MQ가 사용됩니다.

```
enable_MQFunctions -n sample -u user1 -p password1
```

사용 참고

DB2 MQ 함수는 이 명령에서 자동으로 작성되는 DB2MQ 스키마에서 실행됩니다.

이 명령을 실행하기 전에 다음을 수행하십시오.

- MQ 및 AMI가 설치되어 있고 MQSeries 버전이 5.2 이상인지 확인하십시오.
- 환경 변수 \$AMT_DATA_PATH가 정의되어 있는지 확인하십시오.
- 디렉토리를 DB2PATH의 서브디렉토리 cfg로 변경하십시오.

UNIX인 경우 다음을 수행하십시오.

- db2set을 사용하여 AMT_DATA_PATH를 DB2ENVLIST에 추가합니다.
- UDF 실행과 연관된 사용자 계정이 mqm 그룹의 구성원인지 확인하십시오.
- 이 명령을 호출할 사용자가 mqm 그룹의 구성원인지 확인하십시오.

주: AIX 4.2는 MQSeries 5.2에서 지원되지 않습니다.

disable_MQFunctions

disable_MQFunctions

지정된 데이터베이스에 대해 DB2 MQSeries 함수를 사용하지 않습니다.

권한 부여

다음 중 하나입니다.

- *sysadm*
- *dbadm*
- 데이터베이스의 IMPLICIT_SCHEMA(함수의 명시적 또는 내재된 스키마 이름이 없는 경우)
- 스키마에서 CREATEIN 특권(스키마 이름 DB2MQ가 있는 경우)

명령 구문

```
▶▶ disable_MQFunctions -n database -u userid -p password ▶▶
```

명령 매개변수

-n database

데이터베이스 이름을 지정합니다.

-u userid

데이터베이스에 연결할 사용자 ID를 지정합니다.

-p password

사용자 ID의 암호를 지정합니다.

예

다음 예에서는 DB2MQ 함수를 SAMPLE 데이터베이스에 대해 사용하지 않습니다.

```
disable_MQFunctions -n sample -u user1 -p password1
```


제4부 관리 도구

제어 센터	215
VSE 및 VM 서버용 DB2 서버 관리 능력	215
제어 센터에 대한 Java 1.2 지원.	215
Windows 운영 체제에서 온라인 도움말 사용 시 "유효하지 않은 단축 아이콘" 오류	216
OS/2의 Java 제어 센터.	216
Windows 운영 체제의 저널에서 완료된 작업을 보려고 할 때 "파일 액세스 거부" 오류	216
멀티사이트 갱신 테스트 연결	216
OS/390용 DB2 제어 센터.	217
OS/390에 대한 제어 센터의 필수 수정.	219
공간 계층 작성 대화 상자로 변경	219
DB2 제어 센터에 대한 문제점 해결 정보.	219
UNIX 기반 시스템에서 제어 센터 문제점 해결	219
OS/2에서 가능한 정보 팝업 상자 문제점	220
jdk11_path 구성 매개변수에 대한 도움말	220
스크립트 센터 또는 저널 사용 시 Solaris 시스템 오류(SQL10012N)	220
DPREPL.DFT 파일에 대한 도움말	220
하나 이상의 제어 센터 애플릿 시작.	221
애플릿으로 수행되는 제어 센터에 대한 온라인 도움말	221
애플릿 모드에서 제어 센터 수행(Windows 95)	221
대형 조회 결과로 작업	221
정보 센터	223
Windows 운영 체제에서 "유효하지 않은 단축 아이콘" 오류.	223
Netscape가 이미 열려 있을 때 Netscape Navigator에서 외부 웹 링크 열기(Unix 기반 시스템).	223
정보 센터 시작 문제점	223
마법사.	225
데이터베이스 작성 마법사에서 Extent 크기 설정	225
MQSeries 지원 마법사	225
OLE DB 지원 마법사	225

제어 센터

VSE 및 VM 서버용 DB2 서버 관리 능력

DB2 Universal Database 버전 7 제어 센터는 VSE 및 VM용 DB2 서버 데이터베이스의 지원을 향상시켰습니다. 모든 VSE 및 VM용 DB2 서버 데이터베이스 오브젝트는 제어 센터에서 볼 수 있습니다. CREATE INDEX, REORGANIZE INDEX 및 UPDATE STATISTICS 명령문과 REBIND 명령에 대한 지원도 있습니다. REORGANIZE INDEX 및 REBIND에서는 VSE 및 VM용 DB2 서버 호스트에서 저장 프로시저가 수행되고 있어야 합니다. 이 저장 프로시저는 VSE 및 VM용 DB2 서버의 VSE 및 VM용 제어 센터에서 제공됩니다.

완전히 통합된 제어 센터는 사용자가 DB2 서버가 수행되는 플랫폼에 상관없이 DB2를 관리할 수 있게 합니다. VSE 및 VM용 DB2 서버 오브젝트는 DB2 Universal Database 오브젝트와 함께 제어 센터 기본 창에 표시됩니다. 이들 오브젝트를 관리하기 위한 해당 조치와 유틸리티는 오브젝트를 선택하여 호출합니다. 예를 들어, 사용자는 특정 데이터베이스의 색인을 나열하고 색인 중 하나를 선택하여 재구성할 수 있습니다. 사용자는 데이터베이스의 테이블을 나열하고 통계를 갱신하거나 테이블을 복제 소스로 정의할 수도 있습니다.

DB2 Connect 사용자 안내서 또는 설치 및 구성 보충 설명서에서 VSE 및 VM용 DB2 Server 오브젝트에서 관리 작업을 수행하기 위한 제어 센터 구성에 대한 자세한 내용을 참조하십시오.

제어 센터에 대한 Java 1.2 지원

제어 센터는 Java 1.2에서 bi-di 지원을 사용하여 아랍어와 히브리어 같은 양방향 언어를 지원합니다. 이 지원은 Windows NT 플랫폼에서만 제공됩니다.

Java 1.2는 제어 센터가 이를 인식하고 사용할 수 있도록 설치해야 합니다.

1. JDK 1.2.2는 DB2 UDB CD의 DB2\bi-di\NT 디렉토리에서 사용할 수 있습니다. `ibm-inst-n122p-win32-x86.exe`는 설치 프로그램이고 `ibm-jdk-n122p-win32-x86.exe`는 JDK 분산 프로그램입니다. 두 파일을 모두 하드 드라이브의 임시 디렉토리에 복사한 다음, 거기에서 설치 프로그램을 수행하십시오.
2. <DB2PATH>\java\Java12에 설치하십시오. 여기서 <DB2PATH>는 DB2의 설치 경로입니다.
3. JDK/JRE 설치에서 프롬프트를 표시할 때 시스템 VM으로 JDK/JRE를 선택하지 마십시오.

Java 1.2가 성공적으로 설치되면, Java 1.2를 사용하여 보통 방식으로 제어 센터를 시작합니다.

Java 1.2의 사용을 중지하려면, <DB2PATH>\java\Java12에서 JDK/JRE를 설치 제거하거나 <DB2PATH>\java\Java12 서브디렉토리의 이름을 바꾸십시오.

주: <DB2PATH>\java\Java12를 <DB2PATH>\Java12와 혼동하지 마십시오.
<DB2PATH>\Java12는 DB2 설치의 일부이며 Java 1.2에 대한 JDBC 지원이 포함됩니다.

Windows 운영 체제에서 온라인 도움말 사용 시 "유효하지 않은 단축 아이콘" 오류

제어 센터 온라인 도움말을 사용할 때, "유효하지 않은 단축 아이콘"과 같은 오류가 발생할 수 있습니다. 최근에 새로운 웹 브라우저나 웹 브라우저의 새로운 버전을 설치한 경우, HTML 및 HTM 문서가 올바른 브라우저와 연관되었는지 확인하십시오. Windows 도움말 항목 "파일을 열 때 시작되는 프로그램 변경"을 참조하십시오.

OS/2의 Java 제어 센터

제어 센터는 HPFS로 포맷된 드라이브에 설치해야 합니다.

Windows 운영 체제의 저널에서 완료된 작업을 보려고 할 때 "파일 액세스 거부" 오류

Windows NT용 DB2 Universal Database에서, 스크립트 센터에서 작성된 작업의 세부사항을 보기 위해 저널을 열려고 시도하면 "파일 액세스 거부" 오류가 발생합니다. 작업 상태는 완료를 나타냅니다. 이 경우는 스크립트 센터에서 작성된 작업에 START 명령이 포함되면 발생합니다. 이 오류를 방지하려면, 두 배치 파일과 작업 자체에서 START 대신 START/WAIT를 사용하십시오.

멀티사이트 갱신 테스트 연결

버전 7 제어 센터의 멀티사이트 갱신 테스트 연결 기능은 목표 인스턴스의 버전에 제한을 받습니다. "원격" 테스트 연결 기능을 수행하려면 목표 인스턴스는 최소한 버전 7이어야 합니다. 버전 6에서 멀티사이트 갱신 테스트 연결 기능을 수행하려면, 목표 인스턴스에서 제어 센터를 지역적으로 시작하고 수행해야 합니다.

OS/390용 DB2 제어 센터

OS/390용 DB2 UDB 제어 센터는 사용권이 부여된 IBM DB2 유틸리티의 사용을 관리할 수 있게 합니다. 별도로 주문할 수 있는 OS/390용 DB2 UDB 기능인 유틸리티 기능은 DB2 제어 센터가 관리하기 전에 사용자의 환경에서 사용권을 부여받고 설치해야 합니다.

OS/390용 DB2 서브시스템을 구성할 때 제어 센터를 통해 정의된 "CC390" 데이터베이스는 제어 센터의 내부 지원용으로 사용됩니다. 이 데이터베이스는 수정하지 마십시오.

OS/390용 DB2 버전 7.1이 제어 센터 목차나 정보 센터 탭 정보에 특별히 언급되지 않았더라도 이 문서는 OS/390용 DB2 버전 7.1 기능을 지원합니다. 많은 OS/390용 DB2 버전 6 고유 기능도 OS/390용 DB2 버전 7.1과 관련되어 있으며 목차의 OS/390용 DB2 버전 7.1의 일부 고유 기능에는 버전 지정이 없습니다. 제어 센터에서 OS/390용 DB2 버전 7.1 서브시스템을 구성한 경우, 이 버전의 모든 문서에 액세스할 수 있습니다.

OS/390용 DB2 제어 센터에서 DDL 생성 함수에 액세스하여 사용하려면, DDL 생성 함수를 설치해야 합니다.

- 버전 5의 경우, OS/390용 DB2 버전 5와 함께 DB2Admin 2.0을 설치하십시오.
- 버전 6의 경우, OS/390용 DB2 버전 6의 DB2 Admin 기능에 대한 PTF로 사용 가능한 작은 프로그래밍 향상 기능을 설치하십시오.
- 버전 7.1의 경우, DDL 생성 함수는 별도로 가격이 설정된 OS/390용 DB2 버전 7.1의 DB2 관리 기능의 일부입니다.

Stored Procedure Builder는 DB2 UDB 제어 센터를 시작하기 전에 설치되어야 제어 센터에서 액세스할 수 있습니다. 이것은 DB2 Application Development Client의 일부입니다.

워크스테이션에서 직접 OS/390용 DB2 서브시스템을 카탈로그화하려면, 클라이언트 구성 지원 프로그램 도구를 사용하도록 선택하십시오.

1. 소스 페이지에서, 데이터베이스에 연결 수동 구성 단일선택 버튼을 지정하십시오.
2. 프로토콜 페이지에서, 적합한 통신 정보를 완성하십시오.
3. 데이터베이스 페이지에서, 데이터베이스 이름 피드에 서브시스템 이름을 지정하십시오.
4. 노드 옵션 페이지에서, 노드 옵션 구성(선택적) 선택란을 선택하십시오.
5. 운영 체제 필드의 목록에서 MVS/ESA, OS/390을 선택하십시오.
6. 완료 버튼을 눌러 구성을 완료하십시오.

게이트웨이 머신을 통해 OS/390용 DB2 서브시스템을 카탈로그화하려면, 게이트웨이 머신에서 위의 단계 1-6을 따르고 다음과 같이 수행하십시오.

1. 클라이언트 머신에서, 제어 센터를 시작하십시오.
2. 시스템 폴더를 마우스 오른쪽 버튼으로 클릭하고 추가를 선택하십시오.
3. 시스템 추가 대화 상자에서, 시스템 이름 필드에 게이트웨이 머신 이름을 입력하십시오.
4. DB2DAS00을 원격 인스턴스 필드에 입력하십시오.
5. TCP/IP 프로토콜의 경우, 프로토콜 매개변수에서 호스트 이름 필드에 게이트웨이 머신의 호스트 이름을 지정하십시오.
6. 523을 서비스 이름 필드에 입력하십시오.
7. 확인을 클릭하여 시스템을 추가하십시오. 이제 시스템 폴더 아래에 추가된 게이트웨이 머신이 표시될 것입니다.
8. 게이트웨이 머신 이름을 확장하십시오.
9. 인스턴스 폴더를 마우스 오른쪽 버튼으로 클릭하고 추가를 선택하십시오.
10. 인스턴스 추가 대화 상자에서, 새로 고침을 클릭하여 게이트웨이 머신에서 사용 가능한 인스턴스를 나열하십시오. 게이트웨이 머신이 Windows NT 시스템인 경우, OS/390용 DB2 서브시스템은 인스턴스 DB2에 카탈로그화되었을 것입니다.
11. 인스턴스를 선택하십시오. 이 인스턴스에 대한 프로토콜 매개변수는 자동으로 채워집니다.
12. 확인을 클릭하여 인스턴스를 추가하십시오.
13. 인스턴스 폴더를 열어 방금 추가한 인스턴스를 확인하십시오.
14. 인스턴스를 확장하십시오.
15. 데이터베이스 폴더를 마우스 오른쪽 버튼으로 클릭하고 추가를 선택하십시오.
16. 새로 고침을 눌러 게이트웨이 머신 상의 지역 데이터베이스를 표시하십시오. 데이터베이스 추가 대화 상자에서 DB2 서브시스템을 추가하는 경우, 데이터베이스 이름 필드에 서브시스템 이름을 입력하십시오. 서브시스템(또는 데이터베이스)의 지역 별명 이름을 입력하십시오(선택적).
17. 확인을 클릭하십시오.

이제 제어 센터에서 서브시스템을 성공적으로 추가했습니다. 데이터베이스를 열면, OS/390용 DB2 서브시스템이 표시되는 것을 확인할 수 있습니다.

『제어 센터 390』 절의 첫번째 문단이 다음과 같이 기술되어 있습니다.

OS/390용 DB2 UDB 제어 센터는 사용권이 부여된 IBM DB2 유틸리티의 사용을 관리할 수 있게 합니다. OS/390용 DB2 UDB의 별도로 주문 가능한 기능의 요소들인 유틸리티 기능은 DB2 제어 센터가 관리하기 전에 사용자의 환경에서 사용권을 부여받고 설치해야 합니다.

이 절은 이제 다음과 같아야 합니다.

OS/390용 DB2 제어 센터는 사용권이 부여된 IBM DB2 유틸리티의 사용을 관리할 수 있게 합니다. 별도로 주문 가능한 제품의 요소인 유틸리티 기능을 DB2 제어 센터가 관리하려면 사용자의 환경에서 사용권을 부여받고 설치되어야 합니다.

OS/390에 대한 제어 센터의 필수 수정

APAR PQ36382를 OS/390용 DB2 버전 5 및 OS/390용 DB2 버전 6의 390 Enablement 기능에 적용하여 버전 7용 DB2 UDB 제어 센터를 사용하여 이들 서버 시스템을 관리해야 합니다. 이 수정을 사용하지 않으면 버전 7용 DB2 UDB 제어 센터를 사용하여 이들 서브시스템용 유틸리티를 실행할 수 없습니다.

APAR은 다음의 FMID에 적용해야 합니다.

OS/390용 DB2 버전 5 390 Enablement: FMID JDB551D
OS/390용 DB2 버전 6 390 Enablement: FMID JDB661D

공간 계층 작성 대화 상자로 변경

"<<" 및 ">>" 버튼이 공간 계층 작성 대화 상자에서 제거되었습니다.

DB2 제어 센터에 대한 문제점 해결 정보

빠른 시작 책의 "제어 센터 설치 및 구성" 장에서 "문제점 해결 정보"는 제어 센터를 애플릿으로 수행할 때 문제가 있으면 명령 창에서 클라이언트 브라우저의 CLASSPATH 설정을 해제하도록 알려줍니다. 이 절에서는 또한 같은 명령 창에서 브라우저를 시작하라고 합니다. 그러나, 브라우저를 시작하는 명령은 제공되지 않습니다. Internet Explorer를 시작하려면, start iexplore를 입력하고 **Enter**를 누르십시오. Netscape를 시작하려면, start netscape를 입력하고 **Enter**를 누르십시오. 이 명령들은 브라우저가 PATH에 있는 것으로 가정한 것입니다. 그렇지 않으면, 이를 PATH에 추가하거나 브라우저의 설치 디렉토리로 전환한 후 **start** 명령을 다시 발행하십시오.

UNIX 기반 시스템에서 제어 센터 문제점 해결

UNIX 기반 시스템에서 제어 센터를 시작할 수 없는 경우, JAVA_HOME 환경 변수가 Java 분산을 가리키도록 설정하십시오.

- Java가 /usr/jdk118에 설치된 경우, JAVA_HOME을 /usr/jdk118로 설정하십시오.
- sh, ksh 또는 bash 셸의 경우
export JAVA_HOME=/usr/jdk118.
- csh 또는 tcsh 셸의 경우
setenv JAVA_HOME /usr/jdk118

OS/2에서 가능한 정보 팝업 상자 문제점

화면 크기가 1024x768이고 256 색상이며 Workplace Shell Palette Awareness가 사용 가능한 상태에서 OS/2에서 제어 센터를 수행하는 경우, 현재 창의 경계를 넘어서 확장되는 정보 팝업 상자는 검정색 배경에 검정색 텍스트로 표시될 수 있습니다. 이 문제점을 수정하려면, 표시 설정을 256 색상 이상으로 변경하거나 Workplace Shell Palette Awareness를 사용 불가능하게 하십시오.

jdk11_path 구성 매개변수에 대한 도움말

제어 센터 도움말에서, Java Development Kit 1.1 설치 경로(jdk11_path) 구성 매개변수의 설명에 하위 표제 적용 대상 아래에 한 행이 누락되었습니다. 적용 대상의 완전한 목록은 다음과 같습니다.

- 지역 및 원격 클라이언트가 있는 데이터베이스 서버
- 클라이언트
- 지역 클라이언트가 있는 데이터베이스 서버
- 지역 및 원격 클라이언트가 있는 파티션 데이터베이스 서버
- 지역 클라이언트가 있는 위성 데이터베이스 서버

스크립트 센터 또는 저널 사용 시 Solaris 시스템 오류(SQL10012N)

스크립트 센터나 저널에서 Solaris 시스템을 선택할 때, 다음 오류가 발생할 수 있습니다.

```
SQL10012N - 지정된 라이브러리 "/udbprod/db2as/sqlllib/function/unfenced/db2scdar!ScheduleInfoOpenScan"을 로드하는 동안 예상하지 못한 시스템 오류가 수신되었습니다. SQLSTATE=42724.
```

이것은 Solaris 런타임 링커에서의 버그에 의해 야기됩니다. 이 문제점을 정정하려면, 다음 패치를 적용하십시오.

Solaris 2.6의 경우 105490-06(107733은 105490을 대체함)

DPREPL.DFT 파일에 대한 도움말

제어 센터에서, 도구 설정 노트북의 복제 페이지에 대한 도움말에서 5d 단계가 다음과 같습니다.

시스템이 기본 파일로 사용할 수 있도록 파일을 제어 센터에 대한 작업 디렉토리(예를 들어, SQLLIB\BIN)에 저장하십시오.

5d 단계는 다음과 같아야 합니다.

시스템이 기본 파일로 사용할 수 있도록 파일을 제어 센터에 대한 작업 디렉토리(SQLLIB\CC)에 저장하십시오.

하나 이상의 제어 센터 애플릿 시작

동일한 머신에서 동시에 하나 이상의 제어 센터 애플릿을 시작할 수 없습니다. 이 제한 사항은 지원되는 모든 브라우저에서 수행되는 제어 센터 애플릿에 적용됩니다.

애플릿으로 수행되는 제어 센터에 대한 온라인 도움말

제어 센터가 애플릿으로 수행될 경우, F1 키는 정보 팝업 상자가 있는 창과 노트북에 서만 작동합니다.

F1 키를 눌러서 다음과 같은 구성요소에서 정보 팝업 상자를 불러올 수 있습니다.

- OS/390용 DB2 Universal Database
- 마법사

제어 센터의 나머지 구성요소에서, F1은 도움말을 불러오지 않습니다. 다른 구성요소에 대한 도움말을 표시하려면, 도움말 누름 버튼이나 도움말 풀다운 메뉴를 사용하십시오.

애플릿 모드에서 제어 센터 수행(Windows 95)

유효하지 않은 사용자 ID와 암호가 지정된 경우, 스크립트 센터를 열려는 시도가 실패할 수 있습니다. 제어 센터를 시작할 때 유효한 사용자 ID와 암호를 입력하였는지 확인하십시오.

대형 조회 결과로 작업

사용자는 많은 행 수를 리턴하는 조회를 작성하는 것이 쉽습니다. 사용자가 실제로 리턴할 행 수를 예상하는 것은 쉽지 않습니다. 수천개(또는 수백만개)의 행을 리턴할 수 있는 조회를 사용할 경우 두 가지 문제점이 있습니다.

1. 결과를 검색하는 데 시간이 많이 걸립니다.
2. 결과를 보유할 대용량의 클라이언트 메모리가 필요합니다.

이 프로세스를 편리하게 처리하기 위해 DB2는 대형 결과 세트를 청크로 나눕니다. 조회 결과를 한 번에 한 청크씩 검색하여 표시합니다.

따라서,

1. 남은 청크가 검색되는 동안 조회의 첫 청크를 표시할 수 있으므로 표시 시간이 줄어듭니다.
2. 주어진 시간에 조회 결과의 단 한 개의 청크만 클라이언트에 저장되므로 클라이언트의 메모리 요구사항이 줄어듭니다.

메모리의 조회 결과 행 수를 제어하려면 다음을 수행하십시오.

1. 도구 설정 메모장의 일반 페이지를 엽니다.
2. 최대 크기 절에서 다음을 선택합니다.
 - 샘플 내용을 선택하면 샘플 내용 창에 표시된 결과 행 수가 제한됩니다. 항목 필드에 결과 세트의 청크 크기(행 수)를 지정합니다.
 - 명령 센터를 선택하면 명령 센터의 조회 결과 페이지에 표시된 결과 행 수가 제한됩니다. 항목 필드에 결과 세트의 청크 크기(행 수)를 지정합니다.

샘플 내용 창의 조회 결과나 명령 센터의 조회 결과 페이지로 작업할 때 메모리의 행 필드는 조회용 메모리에 있는 행 수를 나타냅니다. 이 수는 설정된 최대 크기보다 커서는 안됩니다. 다음을 눌러 결과 세트의 다음 청크를 검색합니다. 다음이 비활동으로 되어 있으면 결과 세트의 끝에 도달한 것입니다.

정보 센터

Windows 운영 체제에서 "유효하지 않은 단축 아이콘" 오류

정보 센터를 사용할 때, "유효하지 않은 단축 아이콘"과 같은 오류가 발생할 수 있습니다. 최근에 새로운 웹 브라우저나 웹 브라우저의 새로운 버전을 설치한 경우, HTML 및 HTM 문서가 올바른 브라우저와 연관되었는지 확인하십시오. Windows 도움말 항목 "파일을 열 때 시작되는 프로그램 변경"을 참조하십시오.

Netscape가 이미 열려 있을 때 Netscape Navigator에서 외부 웹 링크 열기 (Unix 기반 시스템)

Netscape Navigator가 이미 열려 있어서 지역 DB2 HTML 문서나 외부 웹 사이트를 표시하고 있을 경우, 정보 센터로부터 외부 웹 사이트를 열려고 하면, Netscape 오류가 발생합니다. "Netscape가 <외부 웹 사이트>라는 파일이나 디렉토리를 찾을 수 없습니다"라는 오류가 표시됩니다.

이 문제점을 해결하려면, 외부 웹 사이트를 열기 전에 열려 있는 Netscape 브라우저를 닫으십시오. Netscape는 다시 시작하여 외부 웹 사이트를 엽니다.

Netscape가 이미 열려 있는 상태에서 지역 DB2 HTML 문서를 열려고 할 때에는 이 오류가 발생하지 않습니다.

정보 센터 시작 문제점

일부 시스템에서, 시작 메뉴, 첫 단계 또는 **db2ic** 명령을 사용하여 정보 센터를 호출할 경우 시작하는데 시간이 오래 걸릴 수도 있습니다. 이러한 문제점이 발생하면, 제어 센터를 시작한 후 도움말 --> 정보 센터를 선택하십시오.

마법사

데이터베이스 작성 마법사에서 Extent 크기 설정

데이터베이스 작성 마법사를 사용하면 새로운 데이터베이스의 사용자 테이블 공간에 대한 Extent 크기와 프리페치 크기 매개변수를 설정할 수 있습니다(그러나 카탈로그 또는 임시 테이블에 대해서는 설정할 수 없습니다). 이 기능은 마법사의 "사용자 테이블" 페이지에서 사용자 테이블 공간에 대해 최소한 하나의 컨테이너가 지정된 경우에만 사용 가능합니다.

MQSeries 지원 마법사

DB2 버전 7.2에는 새로운 MQSeries 지원 마법사가 있습니다. 이 마법사는 버전 7.2에서도 새로운 기능인 DB2 MQSeries 함수를 사용하여 MQSeries 대기행렬을 읽는 테이블 함수를 작성합니다. 이 마법사는 스펙에 따라 각 MQSeries 메시지를 구분된 문자열이나 고정된 길이의 컬럼 문자열로 다룰 수 있습니다. 작성된 테이블 함수는 스펙에 따라 문자열을 분석하고 각 MQSeries 메시지를 테이블 함수의 행으로 리턴합니다. 또한 마법사를 사용하여 테이블 함수 최상위에서 뷰를 작성하고 MQSeries 메시지 및 테이블 함수 결과를 미리 열람할 수 있습니다. 마법사는 Stored Procedure Builder나 Data Warehouse Center에서 시작할 수 있습니다.

마법사를 실행하기 위해서 필요한 요구사항은 다음과 같습니다.

- MQSeries 버전 5.2
- MQSeries Application Messaging Interface(AMI)
- DB2 MQSeries 함수

요구사항에 대한 자세한 내용은 195 페이지의 『MQSeries』를 참조하십시오.

샘플 및 MQSeries 지원 마법사 자습서를 보려면 <http://www.ibm.com/software/data/db2/udb/ide>의 자습서 절을 참조하십시오.

OLE DB 지원 마법사

이 마법사를 사용하여 Microsoft OLE DB 표준을 지원하는 다른 데이터베이스 제공자에서 데이터를 읽는 테이블 함수를 작성할 수 있습니다. OLE DB 테이블 함수로 읽는 데이터가 포함된 DB2 테이블을 선택적으로 작성할 수 있으며 OLE DB 테이블 함수에 대한 뷰를 작성할 수 있습니다. 마법사는 Stored Procedure Builder나 Data Warehouse Center에서 시작할 수 있습니다.

| 마법사를 실행하기 위해서 필요한 요구사항은 다음과 같습니다.

- | • OLE DB 제공자(예:Oracle, Microsoft SQL Server)
- | • OLE DB 지원 함수

| 샘플 및 OLE DB 지원 마법사 자습서를 보려면 <http://www.ibm.com/software/data/db2/udb/ide>의 자습서 절을 참조하십시오.

제5부 비즈니스 인텔리전스

비즈니스 인텔리전스 자습서	231	Trillium 메타데이터 가져오기	250
개정된 비즈니스 인텔리전스 자습서	231	메타데이터 맵핑	251
Data Warehouse Center 관리 안내서	233	제한사항	252
문제점 해결	233	Trillium Batch System JCL 파일 작성	252
Excel을 웨어하우스 소스로 설정	233	UNIX 및 Windows에서 Trillium Batch System 스크립트 파일 작성	253
프로세스 정의 및 수행	233	Trillium Batch System 단계 정의	253
메타데이터 내보내기 대화 상자	233	Trillium Batch System 사용자 정의 프로그램 사용	253
OS/390 JCL 작업열 제출 프로그램의 값 정의	234	오류 처리	254
데이터 웨어하우스 샘플 부록 변경사항	234	오류 리턴 코드	255
Data Warehouse Center 메시지	234	로그 파일	256
DB2 OLAP 통합 서버에서 윤곽 작성 및 데이터 로드	234	MQSeries와 Data Warehouse Center 통합	256
Data Warehouse Center에서 Classic Connect 사용	234	MQSeries 메시지용 뷰 작성	256
Data Warehouse Center 환경 구조	235	요구사항	256
역변환기 사용	235	제한사항	257
DB2 버전 7 웨어하우스 에이전트를 사용하여 DB2 버전 5 데이터에 액세스	236	MQSeries 메시지용 뷰 작성	257
DB2 버전 5 서버 이주	236	MQSeries 메시지 및 XML 메타데이터 가져오기	257
에이전트 구성 변경	236	요구사항	257
UNIX 웨어하우스 에이전트	236	제한사항	257
Microsoft Windows NT, Windows 2000 및 OS/2 웨어하우스 에이전트	236	MQSeries 메시지 및 XML 메타데이터 가져오기	258
IBM ERwin 메타데이터 추출 프로그램	238	MQSeries 사용자 정의 프로그램 사용	259
목차	238	오류 리턴 코드	259
소프트웨어 요구사항	239	오류 로그 파일	260
프로그램 파일	239	Microsoft OLE DB 및 Data Transaction Services 지원	260
태그 언어 파일 작성	239	OLE DB 테이블 함수용 뷰 작성	261
태그 언어 파일을 Data Warehouse Center로 가져오기	241	DTS 패키지용 뷰 작성	261
태그 언어 파일을 Information Catalog Manager로 가져오기	242	바꾸기와 함께 증가 확약 사용	262
문제점 해결	243	구성요소 추적 데이터 파일 이름	262
ERwin을 DB2 Data Warehouse Center로 맵핑	245	AIX 및 Solaris 운영 환경에서 Sybase 소스에 대해 필요한 Open Client	262
ERwin을 Information Catalog Manager로 맵핑	245	정정된 샘플 항목	263
Data Warehouse Center에서 이름 및 주소 정리	247	웨어하우스 소스 설정	263
요구사항	247	Microsoft Access의 Memo 필드를 웨어하우스 소스에 맵핑	263
Trillium Software System 구성요소	248	웨어하우스 데이터베이스 유지보수	263
Trillium Batch System을 Data Warehouse Center와 함께 사용	249	DB2 UDB RUNSTATS 프로그램에 대한 단계 부속 유형에 테이블 링크	263
		기본 웨어하우스 제어 데이터베이스	264

웨어하우스 제어 데이터베이스 관리 창	264	필수 필드	284
활동 중인 웨어하우스 제어 데이터베이스 변경	264	Data Warehouse Center 런치패드 향상	284
웨어하우스 제어 데이터베이스 작성 및 초기화	265	단계 정보를 파일로 인쇄	285
편집된 SQL 단계 작성	266	Data Warehouse Center 응용프로그램 통합 안내	
프로세스 모델 편집기 창에서 소스 및 목표 변경	266	서	287
Data Warehouse Center 오브젝트에 설명 추가	266	추가 메타데이터 템플릿	287
샘플 내용 수행.	266	Commit.tag	287
Create DDL SQL문 편집	267	토큰	288
Visual Warehouse 비즈니스 뷰 이주	267	값의 예	288
목표 테이블 및 기본 키 생성	267	ForeignKey.tag	288
Merant ODBC 드라이버 사용	267	토큰	289
새로운 ODBC 드라이버.	268	값의 예	290
OS/2 데이터베이스에서 웨어하우스 소스 또는 목표		ForeignKeyAdditional.tag	290
정의	268	토큰	291
웨어하우스 제어 데이터베이스의 상태 모니터.	268	값의 예	292
TBC_MD 샘플 데이터베이스에서 SQL 지원 사용	268	PrimaryKey.tag	292
FormatDate 함수 사용	268	토큰	292
언어 설정 변경.	269	값의 예	293
키 테이블 생성 변환기 사용	269	PrimaryKeyAdditional.tag	293
데이터베이스에 연결 유지보수.	269	토큰	294
원격 Data Warehouse Center 클라이언트 설정	270	값의 예	294
VM용 DB2 웨어하우스 소스 정의	270	Data Warehouse Center 온라인 도움말.	297
VM용 DB2 또는 VSE용 DB2 목표 테이블 정의	270	복제를 위한 테이블 또는 뷰 정의	297
분리 식별자 지원 사용	270	AS/400 에이전트를 사용하여 Essbase VWP 수행	297
Data Joiner가 바인드 문제점을 지시합니다	270	Data Warehouse Center 발행 창 및 관련 등록 정	
Data Warehouse Center를 사용한 복제 설정 및		보 창 사용	297
수행	271	외부 키	297
문제점 해결 추가 정보	274	복제 노트북.	297
소스 및 목표에 액세스	275	태그 언어 가져오기	297
지원되는 IBM 이외 데이터베이스 소스에 대한 추		데이터 추가를 위한 링크	298
가사항.	276	테이블 가져오기	298
Data Warehouse Center에 수동으로 데이터 소스		RUNSTATS 및 REORGANIZE TABLE 온라인	
작성	277	도움말에 대한 정정	298
CWMI(Common Warehouse Metadata		통지 페이지(웨어하우스 등록정보 노트북 및 스케줄	
Interchange)를 사용하여 메타데이터 가져오기 및		노트북)	298
내보내기	278	에이전트 사이트 노트북의 에이전트 모듈 필드	298
소개	278	DB2 OLAP Starter Kit	299
메타데이터 가져오기	278	OLAP 서버 웹 사이트	299
가져오기 유틸리티 수행 후 메타데이터 갱신	280	지원되는 운영 체제 서비스 레벨	299
메타데이터 내보내기	280	UNIX에서 DB2 OLAP Starter Kit 설정 완료	299
OS/390 Runstats 유틸리티 단계	282	OLAP Starter Kit용 ODBC 구성	300
OS/390 로드 유틸리티 단계	283	UNIX 시스템에서 데이터 소스 구성.	301
CWM(Common Warehouse Metamodel) XML		ODBC 환경 변수 구성	301
지원	283	odbc.ini 파일 편집	301
프로세스 모델 편집기.	284		
스키마 모델 편집기	284		

	odbc.ini 파일에 데이터 소스 추가	301		SAP R/3용 커넥터	327
	DB2용 ODBC 설정 예	302		설치 요구사항	327
	Oracle용 ODBC 설정 예	302		웹용 커넥터	327
	UNIX 시스템에서 OLAP 메타데이터 카탈로그			설치 요구사항	327
	구성	302			
	Windows 시스템에서 데이터 소스 구성	303			
	Windows 시스템에서 OLAP 메타데이터 카탈로그				
	구성	304			
	데이터 소스를 구성한 후의 작업	305			
	OLAP Starter Kit Desktop에서 로그인	306			
	Starter Kit 로그인 예	307			
	OLAP Starter Kit에 대한 샘플 데이터베이스 수동				
	작성 및 구성	307			
	응용프로그램을 OLAP Starter Kit 버전 7.2로 이				
	주	309			
	알려진 문제점 및 제한사항	309			
	OLAP Spreadsheet Add-in EQD 파일 누락	311			
	Information Catalog Manager 관리 안내서	313			
	Information Catalog Manager 초기화 유틸리티	313			
	사용권 문제	313			
	설치 문제	313			
	DB2 버전 7 Information Catalog Manager를 사				
	용하여 DB2 버전 5 정보 카탈로그에 액세스	314			
	정보 카탈로그 설정	315			
	기타 제품과 메타데이터 교환	316			
	flngxoln 명령을 사용하여 메타데이터 교환	316			
	MDISDGC 명령을 사용하여 메타데이터 교환	316			
	프로그램 호출	316			
	Information Catalog Manager Programming				
	Guide and Reference	319			
	Information Catalog Manager 이유 코드	319			
	Information Catalog Manager 사용자 안내서	321			
	Information Catalog Manager: 온라인 메시지	323			
	메시지 FLG0260E	323			
	메시지 FLG0051E	323			
	메시지 FLG0003E	323			
	메시지 FLG0372E	323			
	메시지 FLG0615E	323			
	Information Catalog Manager: 온라인 도움말	325			
	웹에 대한 Information Catalog Manager	325			
	DB2 Warehouse Manager 설치 안내서	327			
	웨어하우스 변환기에 대한 소프트웨어 요구사항	327			

비즈니스 인텔리전스 자습서

개정된 비즈니스 인텔리전스 자습서

FixPak 2에는 버전 7.1에 있는 여러 문제점을 정정한 개정된 비즈니스 인텔리전스 자습서 및 Data Warehouse Center 샘플 데이터베이스가 포함되어 있습니다. 개정된 Data Warehouse Center 샘플 데이터베이스를 적용하려면, 다음을 수행해야 합니다.

샘플 데이터베이스를 아직 설치하지 않았을 경우, 첫 단계 시작 패드를 사용하여 새 샘플 데이터베이스를 작성하십시오. 시작을 클릭하고 프로그램 → IBM DB2 → 첫 단계를 선택하십시오.

이전에 샘플 데이터베이스를 설치했을 경우, 샘플 데이터베이스 DWCTBC, TBC_MD 및 TBC를 삭제하십시오. 보관하고자 하는 데이터를 샘플 데이터베이스에 추가했다면, 샘플 데이터베이스를 삭제하기 전에 백업하십시오. 세 개의 샘플 데이터베이스를 삭제하려면 다음을 수행하십시오.

1. DB2 명령 창을 열려면, 시작을 클릭하고 프로그램 → IBM DB2 → 명령 창을 선택하십시오.
2. DB2 명령 창에서, 다음 세 명령을 각각 입력하고 하나를 입력한 뒤 Enter를 누르십시오.

```
db2 drop database dwctbc  
db2 drop database tbc_md  
db2 drop database tbc
```
3. DB2 명령 창을 닫으십시오.
4. 첫 단계 시작 패드를 사용하여 새 샘플 데이터베이스를 작성하십시오. 시작을 클릭하고 프로그램 → IBM DB2 → 첫 단계를 선택하십시오.

Data Warehouse Center 관리 안내서

문제점 해결

Data Warehouse Center 문제점 해결 정보가 DB2 문제점 해결 안내서로 이동되었습니다.

Excel을 웨어하우스 소스로 설정

웨어하우스 소스 설정"에서, Microsoft Excel에 관한 절인 "Windows NT에서 비 DB2 데이터베이스 웨어하우스 소스 설정" 절에 한 단계가 누락되어 있습니다. 새 단계는 아래에 3단계로 표시되어 있습니다.

Microsoft Excel 95/97 ODBC 드라이버를 사용하여 Excel 스프레드시트에 액세스할 경우, 스프레드시트 내의 각 워크시트에 이름이 지정된 테이블을 작성해야 합니다. 각 워크시트에 대해 이름이 지정된 테이블을 작성하려면 다음을 수행하십시오.

1. 원하는 컬럼과 행을 선택하십시오.
2. Excel → 삽입 → 이름 → 정의를 누르십시오.
3. 이름 정의 창의 "참조" 필드에 1단계에서 선택한 셀이 들어 있는지 확인하십시오. 들어 있지 않으면 "참조" 필드 맨 오른쪽의 아이콘을 눌러 선택한 모든 셀을 포함시키십시오.
4. 표시된 데이터에 이름을 입력하십시오(또는 기본 이름을 사용하십시오).
5. 확인을 클릭하십시오.

프로세스 정의 및 수행

"프로세스 정의 및 수행"의 "Data Warehouse Center 외부에서 단계 시작" 절에서, 2 바이트 이름이 있는 단계를 시작하면 JDK 1.1.8 이상이 웨어하우스 서버 워크스테이션 및 에이전트 사이트에서 필요합니다.

메타데이터 내보내기 대화 상자

"데이터 웨어하우스 센터 메타데이터 내보내기 및 가져오기"절에서 서브섹션 "메타데이터를 태그 언어 파일로 내보내기"의 5단계는 다음과 같아야 합니다.

내보내는 프로세스와 관련된 스케줄 정보를 내보내지 않으려면 스케줄 포함 선택란을 지우십시오.

OS/390 JCL 작업열 제출 프로그램의 값 정의

180 페이지의 "OS/390 JCL 작업열 제출(VWPMVS) 프로그램에 대한 값 정의" 절에서, 단계 8은 JES 파일과 동일한 디렉토리에서 .netrc 파일을 정의해야 한다고 언급합니다. 그 대신, 프로그램은 .netrc 파일을 작성합니다. 파일이 없으면, 프로그램은 홈 디렉토리에 파일을 작성합니다. 홈 디렉토리에 .netrc 파일이 이미 있으면, 프로그램은 기존의 파일 이름을 바꾸고 새로운 파일을 작성합니다. 프로그램이 처리를 완료하면, 작성한 새로운 .netrc 파일을 삭제하고 원래의 파일을 .netrc로 이름을 바꿉니다.

데이터 웨어하우스 샘플 부록 변경사항

- 데이터 웨어하우징 샘플 부록의 "샘플 메타데이터 보기 및 수정" 절에서, GEOGRAPHIES 테이블이 소스 테이블의 목록에 포함되어야 합니다.
- 데이터 웨어하우징 샘플 부록의 "단계 승격" 절에서, 생산 모드로의 단계 승격에 대한 절차에서 다음의 내용은 올바르지 않습니다. 테스트 모드로 단계를 승격시켰을 때 목표 테이블이 작성되었기 때문입니다.

Data Warehouse Center가 목표 테이블 작성을 시작하고, 진행 창을 표시합니다.

Data Warehouse Center 메시지

Microsoft Windows NT 및 Windows 2000에서, Data Warehouse Center는 시스템 이벤트 로그에 이벤트를 기록합니다. 이벤트 ID는 Data Warehouse Center 메시지 번호에 해당됩니다. Data Warehouse Center 메시지는 메시지 참조서에서 자세한 내용을 참조하십시오.

DB2 OLAP 통합 서버에서 윤곽 작성 및 데이터 로드

315 페이지의 그림 20에 있는 예에는 오류가 있습니다. 다음 명령이 맞습니다.

```
"C:\IS\bin\olapicmd" < "C:\IS\Batch\my_script.script" >
"C:\IS\Batch\my_script.log"
```

"C:\IS\bin\olapicmd"를 둘러싸는 큰 따옴표는(프로그램 파일처럼) 경로에서 디렉토리의 이름에 공백이 있을 경우 필요합니다.

Data Warehouse Center에서 Classic Connect 사용

- "부록 F. Data Warehouse Center와 함께 Classic Connect 사용"의 388 페이지에 있는 "CROSS ACCESS ODBC 드라이버 설치" 절은 다음 정보로 교체되었습니다.

DB2 Warehouse Manager 버전 7의 사용자 설치를 수행하고, Classic Connect Drivers 구성요소를 선택하여 CROSS ACCESS ODBC 드라이버를 설치하십시오. 드라이버는 DB2 Warehouse Manager의 일반 설치의 일부로서 설치되지 않습니다.

CROSS ACCESS ODBC 드라이버는 SQLLIB 디렉토리의 ODBC32 서브디렉토리에 설치됩니다. 설치가 완료된 후, 드라이버의 경로를 (예를 들어, C:\Program Files\SQLLIB\ODBC32) PATH 시스템 환경 변수에 수동으로 추가해야 합니다. 다른 버전의 CROSS ACCESS ODBC 드라이버가 이미 설치되어 있으면, ...\SQLLIB\ODBC32\를 다른 버전의 경로 앞에 두십시오. 운영 체제는 CROSS ACCESS ODBC 드라이버가 들어 있는 경로의 첫번째 디렉토리를 사용합니다.

- 다음 프로시저가 "부록 F, Data Warehouse Center와 함께 Classic Connect 사용"에 추가되어야 합니다.

Classic Connect ODBC 드라이버를 설치하려면 다음을 수행하십시오.

1. Warehouse Manager CD-ROM을 CD-ROM 드라이브에 넣으십시오. 런치패드 가 열립니다.
2. 런치패드에서 설치를 누르십시오.
3. 제품 선택 창에서, **DB2 Warehouse Manager** 선택란이 선택되어 있는지 확인한 후, 다음을 누르십시오.
4. 설치 유형 선택 창에서, 사용자 설치를 선택한 후, 다음을 누르십시오.
5. 구성요소 선택 창에서, **Classic Connect** 드라이버 및 웨어하우스 에이전트를 선택하고, 다른 모든 선택란은 지운 후, 다음을 누르십시오.
6. 파일 복사 시작 창에서, 선택사항을 검토하십시오. 선택사항을 변경하려면, 뒤로 를 눌러 선택했던 창으로 돌아가십시오. 다음을 눌러 설치를 시작하십시오.

Data Warehouse Center 환경 구조

401 페이지에 있는 "부록 G. Data Warehouse Center 환경 구조"에서, 테이블에 잘못된 항목이 있습니다. C:\Program Files\SQLLIB\ODBC32는 PATH 환경 변수에 추가되지 않습니다. PATH 환경 변수의 유일한 갱신 내용은 C:\Program Files\SQLLIB\BIN 입니다.

역변환기 사용

책에서는 역변환기가 매개변수를 기본으로 목표 테이블을 작성할 수 있지만, 생성된 목표 테이블에 원하는 출력 컬럼이 없는 포인트를 놓쳤다고 설명합니다. 이 출력 컬럼은 목표 테이블에서 명시적으로 작성되어야 합니다.

DB2 버전 7 웨어하우스 에이전트를 사용하여 DB2 버전 5 데이터에 액세스

DB2 버전 7 웨어하우스 에이전트는 DB2 버전 7 설치 프로세스에 의해 구성된대로 DB2 버전 6과 DB2 버전 7 데이터에 대한 액세스를 지원합니다. DB2 버전 5 데이터에 액세스해야 하는 경우, 다음의 두 가지 중 하나를 사용해야 합니다.

- DB2 버전 5 서버를 DB2 버전 6 또는 DB2 버전 7로 이주시키십시오.
- DB2 버전 5 데이터에 대한 액세스를 허용하도록 적합한 운영 체제에서 에이전트 구성을 수정하십시오.

DB2 버전 7 웨어하우스 에이전트는 DB2 버전 2 또는 다른 이전 버전의 데이터에 대한 액세스를 지원하지 않습니다.

DB2 버전 5 서버 이주

운영 체제의 *DB2 Universal Database* 빠른 시작에서 DB2 버전 5 서버의 이주에 대한 내용을 참조하십시오.

에이전트 구성 변경

다음의 정보는 각 운영 체제에서 에이전트 구성을 변경하는 방법을 설명한 것입니다. DB2 서버를 DB2 버전 6 이상으로 이주시키려면, 구성에 대한 변경사항을 제거하십시오.

UNIX 웨어하우스 에이전트

CLI나 ODBC 액세스를 통해 DB2 버전 5에서 데이터에 액세스하도록 UNIX 웨어하우스 에이전트를 설정하려면, 다음과 같이 수행하십시오.

1. DB2 버전 6 Run-time Client를 설치하십시오. 다음의 URL에서 클라이언트 다운로드를 선택하여 Run-time Client를 얻을 수 있습니다.

<http://www.ibm.com/software/data/db2/udb/support>

2. DB2INSTANCE 환경 변수가 DB2 버전 6 인스턴스를 지정하도록 웨어하우스 에이전트 구성 파일을 갱신하십시오.
3. DB2 버전 6 인스턴스에서 웨어하우스 에이전트가 액세스할 모든 데이터베이스를 카탈로그화하십시오.
4. 에이전트 디먼 프로세스 ID로 kill 명령을 발행하여 에이전트 디먼 프로세스를 종료하십시오. 그러면, 에이전트 디먼은 자동으로 재시작됩니다. 프로세스를 종료하려면 루트 권한이 필요합니다.

Microsoft Windows NT, Windows 2000 및 OS/2 웨어하우스 에이전트

DB2 버전 5에서 데이터에 액세스하도록 Microsoft NT, Windows 2000 또는 OS/2 웨어하우스 에이전트를 설정하려면, 다음과 같이 수행하십시오.

1. DB2 버전 7 웨어하우스 에이전트가 설치되지 않은 워크스테이션에 DB2 Connect Enterprise Edition 버전 6을 설치하십시오.

DB2 Connect Enterprise Edition은 DB2 Universal Database Enterprise Edition 및 DB2 Universal Database Enterprise - Extended Edition의 일부로 포함됩니다. 이들 DB2 제품 중 하나의 버전 6이 설치된 경우, DB2 Connect를 별도로 설치할 필요가 없습니다.

제한사항: 동일한 Windows NT 또는 OS/2 워크스테이션에 여러 DB2 버전을 설치할 수 없습니다. 다른 Windows NT 워크스테이션이나 OS/2 또는 UNIX 워크스테이션에 DB2 Connect를 설치할 수 있습니다.

2. DB2 버전 5 데이터에 액세스할 수 있도록 웨어하우스 에이전트와 DB2 Connect 버전 6을 구성하십시오. *DB2 Connect* 사용자 안내서에서 자세한 내용을 참조하십시오. 다음의 단계는 필수적인 단계의 개요입니다.

- a. DB2 버전 5 시스템에서 DB2 명령행 처리기를 사용하여 웨어하우스 에이전트가 액세스할 버전 5 데이터베이스를 카탈로그화하십시오.
- b. DB2 Connect 시스템에서 DB2 명령행 처리기를 사용하여 다음을 카탈로그화하십시오.
 - DB2 버전 5 시스템에 대한 TCP/IP 노드
 - DB2 버전 5 시스템에 대한 데이터베이스
 - DB2 버전 5 시스템에 대한 DCS 항목
- c. 웨어하우스 에이전트 워크스테이션에서 DB2 명령행 처리기를 사용하여 다음을 카탈로그화하십시오.
 - DB2 Connect 시스템에 대한 TCP/IP 노드
 - DB2 Connect 시스템에 대한 데이터베이스

DB2 Universal Database 설치 및 구성 보충 설명서에서 데이터베이스 카탈로그화에 대한 내용을 참조하십시오.

3. 웨어하우스 에이전트 워크스테이션에서 DB2 Connect를 통해 액세스할 각 데이터베이스에 DB2 CLI 패키지를 바인드하십시오.

다음의 DB2 명령은 가정한 DB2 버전 5 데이터베이스인 v5database로의 바인딩 예를 제공합니다. DB2 명령행 처리기를 사용하여 다음의 명령을 발행하십시오. db2cli.lst와 db2ajgrt는 \sqllib\bnd 디렉토리에 위치합니다.

```
db2 connect to v5database user userid using password
db2 bind db2ajgrt.bnd
db2 bind @db2cli.lst blocking all grant public
```

여기서 *userid*는 V5 데이터베이스의 사용자 ID이고 *password*는 사용자 ID의 암호입니다.

db2cli.list가 DB2 버전 5 데이터베이스에 바인드되면 오류가 발생합니다. 이 오류는 이 구성에서 대형 오브젝트(LOB)가 지원되지 않기 때문에 발생합니다. 이 오류는 DB2 버전 5 데이터베이스에 대한 웨어하우스 에이전트의 액세스에 영향을 주지 않습니다.

2000년 6월에 사용할 수 있는 DB2 Universal Database 버전 5의 FixPak 14는 DB2 Connect를 통해 DB2 버전 5 데이터에 액세스하는 데 필수적입니다. 해당 FixPak에서 APAR 번호 JR14507을 참조하십시오.

IBM ERwin 메타데이터 추출 프로그램

목차

소프트웨어 요구사항

프로그램 파일

태그 언어 파일 작성

태그 언어 파일을 Data Warehouse Center로 가져오기

태그 언어 파일을 Information Catalog Manager로 가져오기

문제점 해결

ERwin을 Data Warehouse Center로 맵핑

ERwin을 Information Catalog Manager로 맵핑

이 절에서는 IBM ERwin 메타데이터 추출 프로그램을 사용하여 ER1 파일에서 메타데이터를 추출하여 DB2 Data Warehouse Center 또는 Information Catalog Manager(DataGuide) 태그 언어 파일을 작성하는 방법을 설명합니다.

메타데이터 추출 프로그램은 입력 ER1 파일에 저장된 데이터베이스, 테이블 및 컬럼 등의 모든 물리적 오브젝트를 추출하여 메타데이터 모델을 Data Warehouse Center나 Information Catalog Manager 태그 언어 파일로 작성합니다. 엔터티 및 속성으로 구성된 Information Catalog Manager의 논리적 모델도 또한 추출되어 작성되는데, 데이터베이스와 테이블 사이 및 테이블과 엔터티 사이 등과 같이 오브젝트들 사이에서 관련된 모든 관계 태그가 작성됩니다. 데이터베이스가 없는 테이블인 경우 이름이 DATABASE인 기본 데이터베이스가 작성됩니다. 스키마가 없는 테이블인 경우 기본 스키마 USERID가 사용됩니다. 모델 이름에 대해서는 ER1 파일 이름이 사용됩니다. ER1 속성을 Data Warehouse Center나 Information Catalog Manager 태그로 맵핑하는 데 대한 자세한 정보는 "ERwin을 DB2 Data Warehouse Center에 맵핑" 및 "ERwin을 Information Catalog Manager에 맵핑"을 참조하십시오.

메타데이터 추출 프로그램은 DB2, Informix, Oracle, Sybase, ODBC 데이터 소스 및 Microsoft SQL Server를 포함하여 관계형 데이터베이스가 있는 모든 ER1 모델을 지원합니다.

소프트웨어 요구사항

메타데이터 추출 프로그램을 수행하려면 다음 소프트웨어 요구사항을 갖춰야 합니다.

- Windows NT 4.0 이상
- ERwin 3.5.2와 Service Pack 3 Build 466

ERwin 태그 언어 파일을 가져오려면 다음 소프트웨어 요구사항을 갖춰야 합니다.

Data Warehouse Center인 경우:

IBM DB2 Universal Database 버전 7.2

Information Catalog Manager인 경우:

IBM DB2 Warehouse Manager 7.2

템플릿 태그 언어 파일(.tag)은 VWS_TEMPLATES 환경 변수가 가리키는 디렉토리에 있어야 합니다.

유형 태그 언어 파일(.typ)은 DGWPATH 환경 변수가 가리키는 디렉토리에 있어야 합니다.

프로그램 파일

메타데이터 추출 프로그램은 IBM DB2 디렉토리의 sqllib\bin 서브디렉토리에 설치되어야 합니다. 프로그램은 디렉토리에 다음 파일들을 설치합니다.

flgerwin.exe

기본 이주 프로그램

erwext.dll

태그 언어 파일 생성 프로그램 DLL

cdmerwsn.dll

ERwin API 래퍼 클래스 DLL

추출 프로그램을 시작하려면 명령 프롬프트에서 flgerwin 명령을 실행하십시오.

태그 언어 파일 작성

Data Warehouse Center 또는 Information Catalog Manager 태그 언어 파일을 작성하려면 flgerwin.exe 프로그램을 실행할 때 두 개의 기본 매개변수를 제공하십시오. 첫 번째 매개변수는 추출할 메타데이터가 있는 ER1 파일입니다. 두 번째 매개변수는 출력

태그 언어 파일입니다. 기본값으로 추출 프로그램은 MERGE 매개변수를 Data Warehouse Center 태그 언어 파일에 추가합니다.

명령 구문은 다음과 같습니다.

```
flgerwin inputFile.er1 outputFile.tag [-dwc] [-icm] [-m] [-u] [-a] [-d]
```

스타 스키마를 작성하려면 명령 구문이 다음과 같아야 합니다.

```
flgerwin inputFile.er1 outputFile.tag [-dwc] [-starschema]
```

-dwc Data Warehouse Center 태그 언어 파일을 작성합니다. -dwc에 대해 사용할 수 있는 선택적 매개변수는 -m 및 -starschema입니다.

-icm Information Catalog Manager 태그 언어 파일을 작성합니다. -icm에 대해 사용할 수 있는 선택적 매개변수는 -m, -u, -a 및 -d입니다.

-starschema

ERwin 모델 스타 스키마 태그 언어 파일을 작성합니다.

-m 오브젝트의 조치를 MERGE로 지정합니다.

-u 오브젝트의 조치를 UPDATE로 지정합니다.

-a 오브젝트의 조치를 ADD로 지정합니다.

-d 오브젝트의 조치를 DELETE로 지정합니다.

메타데이터 추출 프로그램은 데이터가 아니라 메타데이터로 작업합니다. ERwin 태그 언어 파일 가져오기를 완료한 후 목표 테이블을 사용하기 전에 암호 및 사용자 ID가 일치해야 합니다. 메타데이터를 기존 데이터베이스 데이터와 병합하려면 다음을 수행합니다.

등록 정보 --> 데이터베이스 --> 사용자 ID의 병합된 데이터베이스 사용자 ID 및 암호와 일치하도록 Data Warehouse Center 사용자 ID 및 암호를 변경합니다.

메타데이터 추출 프로그램을 사용하여 태그 언어 파일을 목표로 가져올 수 있습니다. 새로 가져온 메타데이터에서 테이블에 아직 데이터가 입력되지 않았습니다. 이들 테이블의 논리적 또는 물리적 표현으로 보고, ERwin에서 가져온 테이블 정의에 데이터를 입력하기 위한 웨어하우스 단계를 작성할 수 있습니다.

입력 ER1 파일은 쓸 수 있는 상태여야 합니다. 메타데이터 추출 프로그램이 수행된 후 ER1 파일은 읽기 전용으로 바뀝니다. 파일을 읽기/쓰기 모드로 변경하려면 다음 예와 같은 명령을 사용하십시오.

```
attrib -r erwinsimplemode.er1
```

여기서 erwinsimplemode.er1은 ERwin 플랫폼 파일 이름입니다.

메타데이터 추출 프로그램은 ER1 파일이 현재 세션에서 사용 중이거나 오류 조건을 발견한 경우 이 파일을 읽기 전용 상태로 저장합니다. ER1 파일이 읽기 전용 상태인 경우 비정상적인 프로그램 종료 메시지가 나타날 수 있습니다. 메타데이터 추출 프로그램은 현재 처리 중인 테이블 이름을 표시합니다. 메타데이터 추출 프로그램의 처리가 끝나면 이를 나타내는 메시지가 표시됩니다.

차원 테이블을 사실 테이블에 자동조인하여 스타 스키마를 작성할 경우 사용하는 테이블 수에 따라 처리 시간이 오래 걸릴 수 있습니다. 처리되는 동안 자동조인 행은 녹색으로 표시됩니다. 저장된 후에는 행이 검은색으로 변합니다.

자동으로 생성된 제한조건 이름을 사용하여 제한조건 이름이 고유한지 확인하십시오.

처리되는 동안 "중복된 컬럼이 발견되었습니다. 컬럼이 추출되지 않습니다"라는 메시지가 나타날 수 있습니다. 이는 정보 메시지이며 추출 프로그램이 정상적으로 완료되는 데에는 영향을 주지 않습니다. 이 메시지는 외부 키의 실제 이름이 현재 처리되고 있는 테이블 컬럼의 실제 이름과 같은 경우에 표시됩니다.

태그 언어 파일을 Data Warehouse Center로 가져오기

두 가지 방법으로 태그 언어 파일을 Data Warehouse Center로 가져올 수 있습니다. Data Warehouse Center를 사용하거나 명령행을 사용할 수 있습니다.

Data Warehouse Center를 사용하여 태그 언어 파일을 가져오려면 다음을 수행하십시오.

1. 시작 --> 프로그램 --> **IBM DB2** --> 제어 센터를 누르십시오. DB2 제어 센터가 열립니다.
2. Data Warehouse Center를 열고 로그인합니다.
3. 웨어하우스를 오른쪽 마우스 단추로 누릅니다. 가져오기 창이 열립니다.
4. 메타데이터 가져오기 -> **ERwin**을 누르십시오. 메타데이터 가져오기 창이 열립니다.
5. 입력 파일 필드에서 입력 태그 언어 파일을 입력하고 확인을 누릅니다.
6. 스타 스키마 추출 선택란을 선택하여 ERwin 스타 스키마 메타데이터 모델을 웨어하우스 스키마로 정의합니다.

가져오기를 완료한 후 보기 --> 새로 고침을 눌러 새로운 단계를 볼 수 있습니다.

명령행을 사용하여 태그 언어 파일을 가져오려면 다음 명령을 입력하십시오.

```
iwh2imp2 tag-filename log-pathname  
target-control-db userid password
```

tag-filename

태그 언어 파일의 전체 경로 및 파일 이름

log-pathname

로그 파일의 전체 경로 이름

target-control-db

가져올 목표 데이터베이스 이름

userid 제어 데이터베이스에 액세스할 때 사용할 사용자 ID

password

제어 데이터베이스에 액세스할 때 사용할 암호

DB2 데이터베이스 정의를 변경하여 Data Warehouse Center의 소스가 되게 하려면 태그 언어 파일을 변경합니다.

- 소스로 사용할 각 데이터베이스에 대해 ISWH 태그를 ISWH(Y)에서 ISWH(N)으로 변경합니다.
- 소스로 사용할 각 데이터베이스에 대해 관계 태그를 **:RELTYPE.TYPE(LINK) SOURCETYPE(SCGTARIR) TARGETYPE(DATABASE)**에서 **:RELTYPE.TYPE(LINK) SOURCETYPE(SCGSR CIR) TARGETYPE(DATABASE)**로 변경합니다.

태그 언어 파일을 가져올 때 다음 메시지가 나타날 수 있습니다.

메시지: DWC13238E "DBNAME(____)
OWNER(____) TABLE(____) COLUMNS(____)"으로 식별된 "COLUMN" 유형의 오
브젝트가
태그 언어 파일에서 두 번 정의되었습니다.

이는 정보 메시지일 뿐이며 가져오기는 성공적으로 완료되었습니다. 같은 이름의 외부 키가 있는 엔터티가 있거나, 절단에 의해 변경된 이름이 비슷한 컬럼이 있는 엔터티가 있거나 또는 이와 비슷한 상황에서 이 메시지가 나타납니다. 중복된 컬럼 이름이 있는지 모델을 확인하고 적절하게 조정합니다.

태그 언어 파일을 Information Catalog Manager로 가져오기

두 가지 방법으로 태그 언어 파일을 Information Catalog Manager로 가져올 수 있습니다. Information Catalog Administrator를 사용하거나 명령행을 사용합니다.

Information Catalog Administrator를 사용하여 태그 언어 파일을 가져오려면 다음을 수행하십시오.

1. 시작 --> 프로그램 --> DB2 --> **Information Catalog Manager**를 누르십시오.
2. 카탈로그 --> 가져오기를 누르십시오. 가져오기 창이 열립니다.
3. 찾기를 눌러 태그 언어 파일을 검색한 후 가져오기를 누릅니다.

가져오기를 완료한 후 주제 아이콘을 더블 클릭하면 가져온 모든 모델 및 데이터베이스를 표시하는 창이 열립니다.

명령 인터페이스를 사용하여 태그 언어 파일을 가져오려면 다음 명령을 입력하십시오.

```
DGUIDE /USERID userid /PASSWORD password  
/DGNAME dgname /IMPORT filename /LOGFILE  
filename /ADMIN /RESTART (B|C)
```

/USERID

제어 데이터베이스에 액세스할 때 사용할 사용자 ID

/PASSWORD

이 사용자 ID의 암호

/DGNAME

Information Catalog Manager 이름

/IMPORT

태그 언어 파일의 전체 경로 및 파일 이름

/LOGFILE

로그 파일의 전체 경로 이름

/ADMIN

관리자로 로그인 합니다.

/RESTART

태그 언어 파일 시작부분에서 가져오기를 시작하거나(선택 B) 마지막 확약 지점에서 시작함(선택 C, 기본값)을 나타냅니다.

문제점 해결

오류 메시지가 나타나면 여기서 메시지를 찾아 오류를 해결하기 위해 취할 수 있는 조치가 있는지 확인하십시오.

ER1 입력 파일이나 태그 출력 파일이 없습니다.

메타데이터 추출 프로그램은 특정한 순서로 두 개의 매개변수를 요구합니다. 첫번째 매개변수는 ER1 파일의 이름이며 두 번째 매개변수는 태그 출력 파일 이름입니다. 기존 태그 언어 파일 이름을 지정하면 파일을 겹쳐씹니다.

Windows 시스템에서 프로그램이 비정상적으로 종료되었습니다.

입력 ER1 파일이 읽기 전용 상태일 것입니다. ER1 파일을 저장할 때 문제가 있는 경우 이 메시지가 나타나며 메타데이터 추출 프로그램은 파일을 읽기 전용 모드로 만듭니다. 명령 셸에서 다음

```
attrib -r inputFile.er1
```

명령을 실행하여 ER1 파일 상태를 읽기/쓰기로 변경합니다.

태그 언어 파일 ... 열 수 없습니다.

현재 드라이브에서 파일을 작성할 수 없거나 열 수 없는 시스템 문제가 있는지 확인합니다.

템플릿 파일 경로를 찾을 수 없습니다.

환경 변수 VWS_TEMPLATES가 설정되지 않았습니다. Data Warehouse Center가 설치되었는지 확인합니다.

유형 파일 경로를 찾을 수 없습니다.

환경 변수 DGWPATH가 설정되지 않았습니다. Data Warehouse Center가 설치되었는지 확인합니다.

지원되지 않는 서버 버전: ...

추출할 입력 ER1 파일이 프로그램에서 지원하지 않는 목표 서버에 저장되어 있습니다. ERwin을 시작하고 ER1 파일을 연 후 서버 --> 목표 서버 및 적절한 버전을 누릅니다 [소프트웨어 요구사항 참조]. ER1 파일을 저장합니다.

알 수 없는 ERwAPI 오류입니다.

ERwin API 오류가 발생했으며 프로그램은 오류에 대한 추가 정보를 얻을 수 없었습니다. ERwin 3.5.2가 설치되어 있는지 확인합니다. ERwin API를 등록해야 합니다.

ERwin API를 등록하려면 ERwin 프로그램 파일이 설치되어 있는 디렉토리에서 **regsvr32er2api32.dll** 명령을 수행합니다. "er2api32.dll의 DllRegisterServer가 성공했습니다"라는 메시지가 나타납니다. Data Warehouse Center에서 추출 프로그램을 시작하거나 명령 셸에서 flgerwin 명령을 실행하여 시작합니다.

추출 프로그램 오류: ...

오류 메시지를 확인하고 적절한 조치를 취합니다. 대부분의 경우처럼 이는 추출 프로그램 내부 오류이며 IBM 대표에게 문제를 보고해야 합니다.

알 수 없는 추출 프로그램 오류입니다.

알 수 없는 오류가 발생했습니다. 대부분의 경우처럼 이는 내부 오류이며 IBM 대표에게 문제를 보고해야 합니다.

오류로 인해 추출 프로그램이 종료되었습니다.

추출 프로그램이 완료되지 못하게 하는 오류가 발생했습니다. 문제를 해결하려면 위해 추가 오류 메시지를 참조하거나 IBM 대표에게 문의하십시오.

ERwin을 DB2 Data Warehouse Center로 맵핑

이 절에서는 기본 ERwin 오브젝트 속성이 Data Warehouse Center 태그에 대응되는 방법을 설명합니다.

데이터베이스 - WarehouseDatabase.tag 또는 SourceDatabase.tag

ERwin	명령행 태그	Data Warehouse Center
다이어그램 이름	NAME	웨어하우스 소스 및 웨어하우스 목표 이름
다이어그램 작성자	RESPNSBL	문의처
데이터베이스 이름	DBNAME	데이터베이스 이름
데이터베이스 버전	DBTYPE	데이터베이스 유형
다이어그램 설명	SHRTDESC	설명

테이블 - Table.tag

ERwin	명령행 태그	Data Warehouse Center
테이블 이름	NAME	테이블 이름
테이블 이름	TABLES	테이블 이름
데이터베이스 이름	DBNAME	n/a
테이블 소유자	OWNER	테이블 스키마
테이블 주석	SHRTDESC	설명

컬럼 - Column.tag

ERwin	명령행 태그	Data Warehouse Center
컬럼 이름	NAME	컬럼 이름
데이터 유형	NATIVEDT	데이터 유형
길이	LENGTH	길이
스케일	SCALE	스케일
null 옵션	NULLABLE	널(NULL) 허용(선택란)
위치	POSNO	n/a
기본 키	KEYPOSNO	n/a
데이터베이스 이름	DBNAME	n/a
테이블 소유자	OWNER	n/a
테이블 이름	TABLES	n/a
컬럼 주석	SHRTDESC	설명

ERwin을 Information Catalog Manager로 맵핑

이 절에서는 기본 ERwin 오브젝트 속성이 Information Catalog Manager 태그에 대응되는 방법을 설명합니다.

데이터베이스 - Database.tag

ERwin	명령행 태그	Information Catalog Manager 인터페이스
다이어그램 이름	NAME	데이터베이스 이름
다이어그램 작성자	RESPNSBL	데이터베이스 소유자
데이터베이스 이름	DBNAME	데이터베이스 이름
데이터베이스 버전	DBTYPE	데이터베이스 유형
다이어그램 설명	SHRTDESC	간단한 설명

테이블 - TableOrView.tag

ERwin	명령행 태그	Information Catalog Manager 인터페이스
테이블 이름	NAME	테이블 이름
테이블 이름	TABLES	테이블 이름
데이터베이스 이름	DBNAME	데이터베이스 이름
테이블 소유자	OWNER	테이블 소유자
테이블 주석	SHRTDESC	간단한 설명
ERwin API	TABLVIEW	정의는 뷰를 나타냄

컬럼 - ColumnOrField.tag

ERwin	명령행 태그	Information Catalog Manager 인터페이스
컬럼 이름	NAME	컬럼 이름
데이터 유형	DATATYPE	컬럼 데이터 유형
길이	LENGTH	컬럼 길이
스케일	SCALE	컬럼 스케일
null 옵션	널(NULL)	컬럼이 널(NULL)이 될 수 있는지 여부(?)
위치	POSNO	컬럼 위치
기본 키	KEYPOSNO	기본 키의 컬럼 위치
ERwin API	ISKEY	컬럼이 키의 일부인지 여부(?)
ERwin API	UNIKEY	컬럼이 고유한 키인지 여부(?)
데이터베이스 이름	DBNAME	데이터베이스 이름
테이블 소유자	OWNER	테이블 소유자
테이블 이름	TABLES	테이블 이름
컬럼 주석	SHRTDESC	간단한 설명
ERwin	ISTEXT	데이터가 텍스트인지 여부(?)
ERwin API	IDSRES	데이터 분석

모델 - Model.tag

ERwin	명령행 태그	Information Catalog Manager 인터페이스
ER1 파일 이름	NAME	모델 이름
다이어그램 작성자	RESPNSBL	자세한 정보...
다이어그램 설명	SHRTDESC	간단한 설명

엔터티 - Entity.tag

ERwin	명령행 태그	Information Catalog Manager 인터페이스
엔터티 이름	NAME	엔터티 이름
참고	SHRTDESC	간단한 설명
정의	LONGDESC	자세한 설명
엔터티 소유자	RESPNSBL	자세한 정보...

속성 - Attribute.tag

ERwin	명령행 태그	Information Catalog Manager 인터페이스
속성 이름	NAME	속성 이름
참고	SHRTDESC	간단한 설명
정의	LONGDESC	자세한 설명
데이터 유형	DATATYPE	구성원 데이터 유형
길이	LENGTH	구성원 길이

Data Warehouse Center에서 이름 및 주소 정리

Data Warehouse Center 및 Trillium Software System을 사용하여 이름 및 주소 데이터를 정리합니다. Trillium Software System은 이름 및 주소 데이터를 다시 포맷하고, 표준화 및 검증하는 이름 및 주소 정리 제품입니다. 사용자 정의 프로그램에서 Trillium Batch System 프로그램을 시작하여 Trillium Software System을 Data Warehouse Center에서 사용할 수 있습니다. Trillium Batch System 스크립트나 JCL에서 메타데이터를 가져올 때 사용자 정의 프로그램이 웨어하우스 트리에 추가됩니다.

Data Warehouse Center는 Vality and Evolutionary Technologies, Inc의 도구와 통합될 수 있습니다.

요구사항

- Trillium Software System을 웨어하우스 에이전트 사이트나 원격 호스트에 설치해야 합니다.
- UNIX 및 Windows 플랫폼인 경우 Trillium Software System의 bin 디렉토리 경로를 시스템 환경 변수 PATH에 추가하여 에이전트가 Trillium Batch System 프

로그래를 수행할 수 있게 해야 합니다. UNIX에서는 vwdaemon 프로세스를 시작하기 전에 IWH.environment 파일에서 PATH 변수를 추가하여 이를 수행해야 합니다.

- 사용자는 Trillium 소프트웨어 작업 지식이 있어야 합니다.

다음 표에서는 소프트웨어 요구사항을 설명합니다.

운영 체제	필수 소프트웨어
UNIX	Trillium Software System 버전 4.0 Data Warehouse Manager 버전 7.2 웨어하우스 에이전트
Windows NT 및 Windows 2000	Trillium Software System 버전 4.0 Data Warehouse Manager 버전 7.2 웨어하우스 에이전트 원격으로 액세스하려면 호스트에 ftpd 및 rexecd 디먼이 설치되어 있어야 합니다.
OS/390	원격 OS/390 호스트에 설치된 Trillium Software System 버전 4.0 UNIX, Windows NT에 설치된 Data Warehouse Manager 버전 7.2 TCP/IP 3.2 이상이 설치되어 있어야 함 OS/390 운영 체제는 원격 호스트로서만 지원됨

Trillium Software System 구성요소

Trillium Software System은 변환 프로그램, 분석 프로그램, 공간형 변환 프로그램 및 일치 프로그램의 네 가지 기본 구성 요소로 구성됩니다. 이름 및 주소 정리 조작을 수행하기 위한 함수 세트로 구성요소를 사용합니다. 사용자 정의 프로그램인, Trillium Batch System에서 구성요소를 수행합니다.

변환 프로그램

변환 프로그램을 사용하여 소스 데이터를 지정된 출력 형식으로 표준화하여 변환합니다.

분석 프로그램

분석 프로그램을 사용하여 이름 및 주소 소스 데이터를 해석하고 소스 데이터에 대한 메타데이터를 작성합니다.

공간형 변환 프로그램

공간형 변환 프로그램으로 우편 서비스 데이터와 소스 데이터를 비교하여 courier 또는 ZIP+4 코드 등의 누락된 정보를 제공합니다. 공간형 변환 프로그램은 또한 United States Census 데이터와 일치시키는 조작도 수행합니다.

일치 프로그램

일치 프로그램으로 비슷한 이름 및 주소를 비교하여 중복된 레코드를 식별합니다. 한 레코드를 레코드 그룹과 비교하는 일치 프로그램을 사용하여 참조 일치를 수행할 수 있습니다.

Trillium Batch System을 Data Warehouse Center와 함께 사용

Data Warehouse Center에서 Trillium Batch System 메타데이터를 가져와서 사용자 정의 프로그램 단계를 작성할 수 있습니다. 이 단계를 지역 웨어하우스 에이전트 또는 원격 웨어하우스 에이전트 사이트의 Trillium Batch System 스크립트라고 합니다. Data Warehouse Center에서 Trillium Batch System 스크립트는 소스 및 목표 파일이 있는 단계입니다. 소스 파일은 첫번째 Trillium Batch System 명령에 대해 사용된 입력 데이터 파일입니다. 목표 파일은 스크립트의 마지막 Trillium 명령에 의해 작성된 출력 데이터 파일입니다. 그런 다음 이 단계를 다른 단계에서 사용하기 위해 다른 프로세스 로 복사할 수 있습니다.

다음 그림은 Trillium Batch System 입력 및 출력 데이터 파일과 Data Warehouse Center의 소스 및 목표 파일 사이의 관계를 보여줍니다.

```
REM Running the converter
pfcondrv -parmfile c:\tril40\us_proj\parms\pfcondrv.par
REM Running the parser
pfprsdv -parmfile c:\tril40\us_proj\parms\pfprsdv.par
REM Running the Matcher
cfmatdrv -parmfile c:\tril40\us_proj\parms\pfmatdrv.par
```

그림 1. 샘플 Trillium 스크립트 파일

```
INP_FNAME01 c:\tril40\us_proj\data\convinp
INP_DDL01 c:\tril40\us_proj\dict\input.ddl
```

그림 2. pfcondrv.par 파일의 내용

```
OUT_DDNAME c:\tril40\us_proj\data\maout
DDL_OUT_FNAME c:\tril40\us_proj\dict\parseout.ddl
```

그림 3. pfmatdrv.par 파일의 내용

```
c:\Tril40\us_proj\data\convinp (source file) ->
Trillium Batch System Step -> c:\tril40\us_proj\data\maout (target file)
```

그림 4. Trillium Batch System 단계 정의

Trillium 메타데이터 가져오기

Trillium 메타데이터를 Data Warehouse Center로 가져오려면 다음을 수행하십시오.

1. Trillium Batch System 스크립트나 JCL을 작성합니다. 스크립트나 JCL 작성 도구를 사용하여 스크립트나 JCL 파일을 작성할 수 있습니다.
2. 웨어하우스를 오른쪽 마우스 단추로 누르고 메타데이터 가져오기 -> Trillium을 눌러 Trillium Batch System 창을 엽니다.
3. 스크립트나 JCL 필드에 수행할 Trillium Batch System 스크립트나 JCL 파일을 입력합니다.
4. 입력 파일 필드에 지정된 스크립트나 JCL 파일에서 제일 먼저 수행할 Trillium Batch System 프로그램의 입력 데이터 파일 이름을 입력합니다.
5. 입력 DDL 필드에 입력 데이터 파일을 설명하는 입력 DDL 파일 이름을 입력합니다. 이 파일은 웨어하우스 에이전트 사이트에서만 사용 가능합니다.
6. 출력 파일 필드에 스크립트나 JCL 파일의 마지막 Trillium Batch System 프로그램에 대한 출력 데이터 파일 이름을 입력합니다.
7. 출력 DDL 필드에 출력 데이터 파일을 설명하는 출력 DDL 파일 이름을 입력합니다. 이 파일은 웨어하우스 에이전트 사이트에서만 사용 가능합니다.
8. 선택적: 출력 오류 파일 필드에 사용할 출력 오류 파일 이름을 입력합니다. 이 오류 파일은 Trillium Batch System 프로그램에서 런타임 오류를 캡처합니다. 이 오류는 stderr 로그에 기록됩니다. 지역 호스트인 경우 여기서 이름을 지정하지 않으면 기본 출력 오류 파일이 작성됩니다. 출력 오류 파일에 대한 자세한 정보는 "오류 조절" 주제를 참조하십시오.
9. 연결 탭을 누르십시오.
10. 가져오는 Trillium 메타데이터가 웨어하우스 에이전트 사이트에 있으면 지역 호스트를 누릅니다.
가져오는 Trillium 메타데이터가 웨어하우스 에이전트 사이트에 없으면 원격 호스트를 누르고 원격 호스트를 지정합니다. 이 절 마지막의 "원격 호스트 지정"을 참조하십시오.
11. 확인을 눌러 Trillium 메타데이터를 가져오고 노트북을 닫습니다.
12. 스크립트나 JCL이 기본 에이전트 사이트에서 수행되지 않는 경우 Trillium Batch System 단계를 위해 등록정보 노트북에서 사용하고 있는 웨어하우스 에이전트 사이트를 지정합니다.

가져오기 조작이 완료되면 다음 웨어하우스 오브젝트가 웨어하우스 트리에 추가됩니다.

- Trillium Batch System.*scriptName* 템플릿(여기서 *scriptName*은 스크립트나 JCL 파일 이름입니다.)
- Trillium Batch System 프로세스
- 사용자 정의 프로그램을 수행하는 Trillium Batch System 단계

- 메타데이터를 가져올 때 지정한 웨어하우스 파일 소스 및 웨어하우스 파일 목표. 파일 소스 및 파일 목표는 고정 파일입니다.
- Trillium Batch System 프로그램 그룹.

원격 호스트 지정

원격 호스트를 지정하려면 다음을 수행하십시오.

1. 원격 호스트를 누르고 가져올 메타데이터가 있는 원격 시스템의 TCP/IP 호스트 이름을 입력합니다. 원격 호스트를 선택하면 원격 목표 파일이 지원되지 않으므로 목표 파일이 지역 파일로 작성됩니다. 원격 파일을 얻는 FTP 단계를 지정된 지역 목표 파일에 추가할 수 있습니다.
2. 원격 운영 체제 목록에서 액세스할 원격 호스트의 운영 체제를 누릅니다.
3. 원격 사용자 ID 필드에 액세스하는 원격 호스트의 사용자 ID를 입력합니다.
4. 암호 옵션 목록에서 액세스할 원격 호스트에 대해 사용할 암호 옵션을 선택합니다.

암호가 필요하지 않음

원격 호스트의 메타데이터에 액세스할 때 암호가 필요하지 않도록 지정합니다.

암호 검색

사용자 정의 프로그램에서 암호를 검색하도록 지정합니다.

암호 프로그램 필드에 암호를 검색할 암호 프로그램 이름을 입력합니다. 프로그램은 웨어하우스 에이전트 사이트에 있어야 하며 첫 행에서 암호를 출력 파일로 작성해야 합니다.

프로그램 매개변수 필드에 암호 프로그램에 대한 매개변수를 입력합니다. 첫 번째 매개변수는 암호가 쓰여진 출력 파일이어야 합니다.

나중에 암호 입력

나중에 암호를 입력하도록 지정합니다.

Trillium Batch System 프로그램을 수행하는 단계에 대한 등록정보 노트 북에 암호를 입력합니다.

메타데이터 맵핑

소스 및 목표 파일에 대한 메타데이터를 작성하기 위해 Trillium은 Trillium DDL 파일을 읽습니다. DDL 파일은 다음과 같은 Data Warehouse Center 데이터 유형으로 변환됩니다.

DDL 데이터 유형	Warehouse 데이터 유형
------------	------------------

ASCII CHARACTER	CHARACTER(<i>n</i>)
ASCII NUMERIC	
EBCDIC CHARACTER	
EBCDIC NUMERIC	
기타 유형	NUMERIC
주: EBCDIC CHARACTER 및 EBCDIC NUMERIC 데이터 유형은 Trillium Software System이 OS/390 운영 체제에서 수행되고 있을 경우에만 지원됩니다.	

변수 *n*은 문자열의 문자 수입니다.

제한사항

Data Warehouse Center의 메타데이터 가져오기 조작 및 Trillium DDL을 사용하여 입력 및 출력 DDL 파일에 겹치는 필드를 지정할 수 있습니다. 그러나 해당하는 웨어하우스 소스 및 웨어하우스 목표 파일은 SQL 단계 또는 샘플 내용과 함께 Data Warehouse Center에 사용될 수 없습니다. 메타데이터 가져오기 조작은 전체 레코드에 걸쳐 겹치는 필드를 무시하므로 이들 필드를 지정할 수는 있지만 결과 소스 및 목표 파일에서는 컬럼으로 사용될 수 없습니다.

오류 파일이 지정되면 스크립트 이름에 공백이 포함될 수 없습니다.

Trillium Batch System JCL 파일 작성

Trillium Batch System JCL 파일을 작성하려면 다음 요구사항을 만족해야 합니다.

- 작업 이름은 사용자 ID에 한 문자를 추가한 것이어야 합니다.
- 작업은 오류 출력 클래스로 경로가 지정되어야 합니다.
- Trillium Batch System 프로그램을 수행하는 각 작업 단계에는 영구 데이터 세트를 정의하는 SYSTEM DD 문이 포함되어야 합니다. 데이터 세트에는 Trillium Batch System 프로그램의 오류가 들어 있습니다. 이 데이터 세트는 JCL이 제출되기 전에 자동으로 삭제됩니다. 오류 처리 및 보고에 대한 자세한 정보는 "오류 조절" 주제를 참조하십시오.

스크립트나 JCL이 원격 호스트에서 수행될 때 출력 오류 파일이 지정되어야 합니다. 그렇지 않으면 메시지가 캡처되지 않아서 Data Warehouse Center로 리턴됩니다. UNIX나 Windows인 경우 오류를 캡처하는 가장 간단한 방법은 Trillium Batch System 스크립트를 호출하고 표준 오류를 출력 파일로 파이프하는 다른 스크립트를 작성하는 것입니다.

```
//SYSTEM DD UNIT=&UNIT,
//          DISP=(MOD,CATLG,KEEP),
//          SPACE=(400,(20,20),,,ROUND),
//          DSN=&PROJPREF.&TRILVER.&PROJECT.STDERR;
```

그림 5. SYSTEM DD 문을 포함하는 작업 단계의 예

UNIX 및 Windows에서 Trillium Batch System 스크립트 파일 작성

Trillium Batch System 스크립트나 매개변수 파일에 입력 파일의 관련 경로가 있으면 사용자는 스크립트 파일 시작 부분의 cd 문을 스크립트 파일 디렉토리에 두어야 합니다.

Trillium Batch System 단계 정의

Trillium Batch System 단계를 정의하기 전에 프로세스에서 사용할 Trillium 메타데이터를 가져와야 합니다. Trillium Batch System 단계를 프로세스에 추가하려면 다음을 수행하십시오.

1. 프로세스 모델 편집기에서 프로세스를 엽니다.
2. 파레트에서 **Trillium Batch System** 아이콘을 누릅니다.
3. **Trillium Batch System** 프로그램 -> *programName*을 누릅니다. 여기서 *programName*은 사용할 Trillium Batch System 프로그램 이름입니다.
4. 단계가 표시될 위치를 캔버스에서 누릅니다.
5. *DB2 Universal Database* 도움말에서 "사용자 정의 프로그램을 수행하는 단계 정의" 주제의 단계를 완료합니다.

Trillium Batch System 사용자 정의 프로그램 사용

Trillium Batch System 사용자 정의 프로그램은 Windows NT 및 UNIX용 DB2 Data Warehouse Center 버전 7.2에 포함됩니다. Trillium 메타데이터를 가져올 때 작성되는 Trillium Batch System 단계는 Trillium Batch System 사용자 정의 프로그램을 수행합니다. 사용자 정의 프로그램은 Trillium Batch System 스크립트나 JCL을 호출합니다. 다음 테이블에는 Trillium Batch System 스크립트나 JCL에 대한 매개변수가 포함되어 있습니다.

매개변수	값
원격 호스트	<ul style="list-style-type: none"> • localhost가 기본값입니다. Trillium Batch System 이 웨어하우스 에이전트 사이트에 설치되어 있는 경우 이 값을 사용하십시오. • Trillium Batch System이 원격 운영 체제에 설치되어 있는 경우에는 원격 호스트 이름
스크립트나 JCL	스크립트나 JCL 이름

원격 운영 체제	원격 호스트의 운영 체제 이름. 원격 호스트 매개변수 값이 localhost인 경우 이 매개변수는 무시됩니다. 유효한 값은 다음과 같습니다. <ul style="list-style-type: none"> • OS/390 운영 체제인 경우 MVS • AIX, SUN Solaris, HP-UX 및 NUMA/Q 운영 체제인 경우 UNIX • Windows NT 또는 2000 운영 체제인 경우 WIN
원격 사용자 ID	원격 명령을 실행할 권한이 있는 사용자 ID. RemoteHostName이 localhost인 경우 이 매개변수는 무시됩니다.
암호 옵션	암호를 얻는 방법. 유효한 값은 다음과 같습니다. <p>ENTERPASSWORD</p> <p>암호가 다음 매개변수로 전달되는 경우 이 값을 사용합니다.</p> <p>PASSWORDNOTREQUIRED</p> <p>암호가 필요하지 않은 경우 이 값을 사용합니다.</p> <p>GETPASSWORD</p> <p>프로그램 이름이 다음 매개변수로 전달되는 경우 이 값을 사용합니다.</p> <p>제한사항:</p> <ul style="list-style-type: none"> • 프로그램은 에이전트 사이트에 있고, 첫 행에서 암호를 출력 파일로 작성해야 하며 성공적으로 수행된 경우 0을 리턴해야 합니다. • Password 매개변수 값은 암호 프로그램 이름이어야 합니다. • Program 매개변수 매개변수 값은 큰 따옴표가 있는 문자열이어야 합니다. • 문자열의 첫 매개변수는 암호를 쓸 출력 파일 이름이어야 합니다.
암호	유효한 값은 암호 또는 암호 프로그램 이름입니다. 암호 프로그램은 웨어하우스 에이전트 사이트에 대해 지역적이어야 합니다.
프로그램 매개변수	암호 프로그램의 매개변수
출력 오류 파일	출력 오류 파일 이름.
주: 이 테이블의 모든 매개변수의 데이터 유형은 CHARACTER입니다.	

오류 처리

Trillium Batch System 프로그램은 Windows NT 및 UNIX 운영 체제에서는 표준 오류(stderr) 파일에 오류 메시지를 쓰며 OS/390 운영 체제에서는 SYSTEMR 데이터 세트에 씁니다.

Windows NT나 UNIX 운영 체제의 Trillium Batch System 프로그램에서 오류를 캡처하려면 표준 오류의 경로를 출력 오류 파일로 재지정해야 합니다.

OS/390 운영 체제의 Trillium Batch System 프로그램에서 오류를 캡처하려면 JCL에 SYSTEM DD 문이 있어야 합니다.

Import Metadata 창에서 출력 오류 파일 이름을 지정한 경우 표준 오류 출력을 오류 파일로 저장하거나 경로를 재지정해야 합니다. Data Warehouse Center는 파일을 읽어 오류 메시지로 문자열 ERROR가 포함된 모든 행을 보고합니다. 모든 Trillium Batch System 프로그램 오류 메시지에는 문자열 ERROR가 들어 있습니다.

출력 오류 파일이 웨어하우스 에이전트 사이트에서 수행 중인 스크립트나 JCL에서 지정되지 않으면 Data Warehouse Center는 자동으로 파일 이름을 만들고 표준 오류 출력의 경로를 이 파일로 재지정합니다. 오류가 발생하면 오류 파일이 삭제되지 않습니다. 오류 파일은 환경 변수 VWS_LOGGING에 의해 지정된 디렉토리에 저장됩니다. 파일 이름은 tbsudp-date- time.err입니다. 여기서 date는 파일이 작성된 시스템 날짜이며 time은 파일이 작성된 시스템 시간입니다. 다음 파일 이름은 출력 오류 파일 이름 형식을 나타냅니다.

tbsudp-021501-155606.err

오류 리턴 코드

오류 번호	설명
0	성공
4	경고. 암호 파일을 지울 수 없거나 Trillium Batch System 사용자 정의 프로그램이 임시 파일에 액세스할 때 내부 오류가 발생했습니다. 환경 변수 VWS_LOGGING으로 지정된 디렉토리에 작성된 모든 임시 파일이나 암호 파일의 상태를 확인하십시오.
8	매개변수의 값이나 수가 올바르지 않습니다. 정확한 구문을 보려면 문서나 로그 파일을 읽으십시오.
12	Trillium Batch System 사용자 정의 프로그램이 FTP를 통해 원격 호스트에 연결되어 있는 동안 문제가 발생했습니다. FTP 연결이나 호스트 이름, 사용자 ID 및 암호를 확인하십시오.
16	Trillium Batch System 사용자 정의 프로그램이 로그 또는 내부 파일을 작성할 수 없습니다. 사용자가 올바른 권한을 갖고 있는지 확인하고 디스크가 가득 차지 않는지 확인합니다.

20	OS/390 JCL이 실행될 수 없거나 Trillium Batch System 사용자 정의 프로그램이 삭제되거나 FTP를 통해 OS/390에서 파일을 가져오는 동안 오류가 발생했습니다. JESLogFile을 점검하여 이유를 확인합니다.
48	환경 변수 VWS_LOGGING을 찾을 수 없거나 로그 파일을 작성할 수 없습니다. 자세한 내용은 로그 파일을 확인하십시오.
56	Windows NT 또는 UNIX 스크립트가 실행될 수 없거나 Trillium Batch System 사용자 정의 프로그램이 원격 호스트에 연결되어 있는 동안 오류가 발생했습니다. 연결이나 호스트 이름, 사용자 ID 및 암호를 확인하십시오.
500	스크립트나 JCL 파일이 오류를 리턴하거나, 오류를 리턴하지는 않지만 오류 파일에 데이터가 있습니다. 자세한 내용은 로그 파일을 확인하십시오. OS/390인 경우 JESLogFile도 확인하십시오.

로그 파일

Data Warehouse Center는 Trillium Batch System 사용자 정의 프로그램이 수행될 때 모든 진단 정보를 로그 파일에 저장합니다. 로그 파일 이름은 `tbsudp-date-time.log`입니다. 여기서 `date`는 파일이 작성된 시스템 날짜이며 `time`은 파일이 작성된 시스템 시간입니다. 로그 파일은 에이전트 사이트에서 환경 변수 `VWS_LOGGING`에 의해 지정된 디렉토리에 작성됩니다. Trillium Batch System 사용자 정의 프로그램이 성공적으로 수행되면 로그 파일이 삭제됩니다.

MQSeries와 Data Warehouse Center 통합

Data Warehouse Center를 사용하여 이제 MQSeries 메시지 대기행렬의 데이터에 DB2 데이터베이스 뷰로 액세스할 수 있습니다. 데이터를 액세스할 수 있는 DB2 테이블 함수 및 DB2 뷰를 작성하는 마법사가 제공됩니다. 각각의 MQSeries 메시지는 분리된 문자열로 취급되는데, 사용자 스펙에 따라 분석되어 결과 행으로 리턴됩니다. 또한 XML 문서인 MQSeries 메시지는 웨어하우스 소스로 액세스할 수 있습니다. Data Warehouse Center를 사용하여 DB2 XML Extender DAD(Document Access Definition) 파일 및 MQSeries 메시지 대기행렬에서 메타데이터를 가져올 수 있습니다.

MQSeries 메시지용 뷰 작성

요구사항

DB2 Universal Database 버전 7.2.

DB2 Warehouse Manager 버전 7.2

MQSeries를 지원합니다. MQSeries 요구사항에 대한 자세한 정보는 195 페이지의 『MQSeries』을(를) 참조하십시오. 웨어하우스 소스 설정에 대한 자세한 정보는 사용자 정의 함수(UDF) 설정 절을 참조하십시오.

제한사항

- 웨어하우스 소스 데이터베이스를 카탈로그화할 때 에이전트 머신에서 데이터베이스 별명도 카탈로그화됩니다. 그러나 MQSeries 및 XML 뷰를 작성할 때 Data Warehouse Center는 클라이언트 머신에서도 데이터베이스 별명이 정의되어 있다고 가정하며 웨어하우스 소스 데이터베이스 사용자 ID 및 암호를 사용하여 연결하려고 시도합니다. 성공하면 마법사가 호출되어 뷰를 작성할 수 있습니다. 성공하지 못하면 경고 메시지가 표시되고 마법사에서 다른 데이터베이스 별명을 카탈로그화하거나 선택해야 합니다.
- MQ 메시지의 최대 길이에 대해서는 릴리스 정보의 SQL 참조서 절을 참조하십시오.

MQSeries 메시지용 뷰 작성

MQSeries 메시지에 대한 뷰를 작성하려면 다음을 수행하십시오.

1. Data Warehouse Center 창에서 웨어하우스 소스 트리를 확장합니다.
2. 뷰를 포함할 웨어하우스 소스를 확장합니다.
3. 뷰 폴더를 오른쪽 마우스 단추로 누르고 **MQSeries 메시지용 작성...**을 누릅니다.
MQSeries 마법사가 열립니다. 마법사를 완료하면 Data Warehouse Center에 새로운 뷰가 작성됩니다. 뷰가 선택되면 MQSeries 대기행렬이 액세스되고 마법사의 사용자 스펙에 따라 분리된 문자열로 각각의 메시지가 분석됩니다.

MQSeries 메시지 및 XML 메타데이터 가져오기

요구사항

DB2 Universal Database 버전 7.2.

DB2 XML Extender 버전 7.2.

MQSeries를 지원합니다. MQSeries 요구사항에 대한 자세한 정보는 195 페이지의 『MQSeries』을(를) 참조하십시오. 웨어하우스 소스 설정에 대한 자세한 정보는 사용자 정의 함수(UDF) 설정 절을 참조하십시오.

제한사항

목표 테이블에 기본 키 및 외부 키가 있으면 가져오기에 실패합니다. 가져오기 전에 데이터 웨어하우스에서 이들 키 정의를 수동으로 삭제해야 합니다.

MQSeries 메시지 및 XML 메타데이터 가져오기

MQSeries 메타데이터를 Data Warehouse Center로 가져오려면 다음을 수행하십시오.

1. 웨어하우스 목표 데이터베이스를 준비합니다.
 - 레지스터뿐 아니라 웨어하우스 목표도 정의하고 변환기를 사용 가능하도록 해야 합니다.
 - DB2 XML Extender에 대한 웨어하우스 목표를 사용 가능하게 해야 합니다. 자세한 정보는 DB2 XML Extender 버전 7.2 릴리스 정보를 참조하십시오.
 - XML Extender DAD(Data Access Definition) 파일을 작성하여 XML 문서 내용을 웨어하우스 테이블에 맵핑하는 방법을 Data Warehouse Center에 알려줍니다. 데이터베이스용 DAD 파일을 사용하여 XML 컬렉션을 사용 가능하게 합니다. 자세한 정보는 DB2 XML Extender 버전 7.2 릴리스 정보를 참조하십시오.
2. 웨어하우스를 오른쪽 마우스 단추로 누르고 메타데이터 가져오기 -> MQSeries를 눌러 메타데이터 가져오기 창을 엽니다.
3. AMI 서비스 필드에 메시지를 보내거나 검색할 서비스 지점을 입력합니다.
4. AMI 규정 필드에 메시지 시스템이 조작을 수행하기 위해 사용할 규정을 입력합니다.
5. DAD 파일 필드에 DB2 XML Extender DAD 파일 이름을 입력하거나 생략 부호(...)를 눌러서 선택할 파일을 검색합니다. 이 파일은 지역적이어야 합니다.
6. 웨어하우스 목표 필드의 콤보 상자 목록에서 단계를 수행할 웨어하우스 목표 이름을 선택합니다. 웨어하우스 목표는 이미 정의되어 있어야 합니다.
7. 스키마 필드에 규정자가 없는 DAD 파일의 테이블 이름을 규정하는 스키마 이름을 입력합니다. 기본 스키마는 이미 선택한 웨어하우스 목표의 로그인 사용자 ID로 간주됩니다.
8. 목표 옵션을 선택합니다.

이 단계에서 런타임에 목표 테이블 내용을 바꾸려면 테이블 내용 바꾸기 단일선택 버튼을 누릅니다.

이 단계에서 런타임 시 목표 테이블 내용에 추가하려면 테이블 내용 추가 단일선택 버튼을 누릅니다.
9. 확인을 클릭하십시오.

메타데이터 가져오기 창이 닫힙니다.

가져오기 조작이 완료되면 다음 웨어하우스 오브젝트가 웨어하우스 트리에 추가됩니다.

- 이름이 MQSeries 및 XML인 주제 영역
- 이름이 MQSeries 및 XML인 프로세스 이름
- 이름이 MQSeries 및 XML인 사용자 정의 프로그램 그룹
- DAD 파일에 설명된 모든 웨어하우스 목표 테이블 정의

- <ServiceName>.<DAD 파일 기본 이름>.<웨어하우스 목표 이름 > 단계.
- <ServiceName>.<DAD 파일 기본 이름> 프로그램 템플릿

웨어하우스 목표 에이전트 사이트가 지역 머신과 다르면 단계 매개변수를 변경해야 합니다.

1. 단계를 오른쪽 마우스 단추로 누르고 등록정보를 선택합니다. 등록정보 노트북에서 매개변수 탭을 누릅니다.
2. DAD 파일 이름 매개변수를 원격 웨어하우스 목표 에이전트 사이트의 동일한 DAD 파일 이름으로 변경합니다.
3. 처리 옵션 탭의 에이전트 사이트에 원하는 에이전트 사이트가 있는지 확인하십시오.

MQSeries 사용자 정의 프로그램 사용

MQSeries 및 XML 저장 프로시더는 **MQXMLXF**라고 하며 Windows NT 및 UNIX용 DB2 Data Warehouse Center 버전 7.2에 포함되어 있습니다. MQSeries 및 XML 메타데이터를 가져올 때 작성된 단계는 저장 프로시더를 수행합니다. 매개변수에 대해서는 다음 테이블에서 설명합니다.

매개변수	값
MQSeries ServiceName	메시지를 보내거나 검색할 서비스 지점 이름
MQSeries PolicyName	메시지 시스템이 조작을 수행하기 위해 사용할 규정 이름
DAD 파일 이름	DB2 XML Extender DAD 파일 이름
TargetTableList	덱표로 구분된 단계의 목표 테이블 목록
옵션	REPLACE 또는 APPEND
수행 ID	스텝 판 번호(로그에 기록하기 위한 번호)
주: 이 테이블의 모든 매개변수의 데이터 유형은 CHARACTER입니다.	

저장 프로시더는 옵션에 REPLACE 값이 있으면 목표 테이블의 모든 행을 삭제합니다. 또한 저장 프로시더는 DB2 XML Extender 저장 프로시더를 호출하여 모든 기존 MQSeries 메시지용 목표 테이블에 데이터를 입력합니다.

오류 리턴 코드

단계를 수행할 때 저장 프로시더는 오류 코드 SQLCODE -443 및 SQLSTATE 38600을 리턴할 수도 있습니다. 오류를 진단하려면 가능한 진단 텍스트가 있는 다음 테이블을 참조하십시오.

오류 번호	설명
AMIRC=xxxx;<log file name>	xxxx는 AMI 계층의 리턴 코드입니다. 자세한 정보는 MQSeries 문서를 참조하십시오. <로그 파일 이름>은 로그 파일의 위치를 나타냅니다.
XMLRC=xxxx;<log file name>	xxxx는 DB2 XML Extender 계층의 리턴 코드입니다. 리턴 코드에 대한 설명은 DB2 XML Extender 문서를 참조하십시오. <로그 파일 이름>은 로그 파일의 위치를 나타냅니다.

SQLCODE=xxxxx;<log file name>	xxxxx는 SQL 요청이 수행될 때 리턴된 0이 아닌 SQLCODE입니다. <로그 파일 이름>은 로그 파일의 위치를 나타냅니다.
-------------------------------	--

모든 오류에 대해 자세한 내용은 로그 파일을 참조하십시오.

오류 로그 파일

Data Warehouse Center는 MQXMLXF가 수행될 때 모든 진단 정보를 로그 파일에 저장합니다. 로그 파일 이름은 mqxf<nnnnnnnnn>.log입니다. 여기서 <nnnnnnnnn>은 저장 프로시저로 전달된 RunID입니다. Data Warehouse Center는 VWS_LOGGING 환경 변수가 나타내는 디렉토리에 파일을 작성합니다. 이 환경 변수가 정의되지 않았으면 로그 파일은 임시 디렉토리에 작성됩니다.

VWS_LOGGING 환경 변수를 Unix 플랫폼의 저장 프로시저에서 볼 수 있게 하려면 **db2start** 명령 전에 **db2set** 명령을 사용하여 VWS_LOGGING을 DB2ENVLIST 환경 변수에 추가해야 합니다. 아래 그림은 환경 명령의 예입니다.

```
db2set DB2ENVLIST="AMT_DATA_PATH VWS_LOGGING"
```

그림 6. 환경 변수 명령 예

단계가 성공적으로 수행되면 로그 파일이 삭제됩니다.

Microsoft OLE DB 및 Data Transaction Services 지원

Data Warehouse Center를 사용하여 이제 OLE DB Provider의 데이터에 DB2 데이터베이스 뷰로 액세스할 수 있습니다. Data Warehouse Center와 함께 제공된 OLE DB Assist 마법사를 사용하여 데이터를 액세스할 수 있는 DB2 OLE DB 테이블 함수와 DB2 뷰를 작성할 수 있습니다.

Microsoft DTS(Data Transformation Services)를 사용하여 OLE DB 소스 및 목표 사이에서 데이터를 가져오고, 내보내고, 변환하여 데이터 웨어하우스 및 데이터 마트를 빌드할 수 있습니다. DTS는 Microsoft SQL Server와 함께 설치됩니다. 모든 DTS 타스크는 DTS 패키지용 Microsoft OLE DB Provider를 사용하여 수행하고 액세스할 수 있는 DTS 패키지에 저장됩니다. OLE DB 소스로 DTS의 패키지에 액세스할 수 있으므로 OLE DB 데이터 소스와 같은 방법으로 DTS 패키지용 OLE DB Assist 마법사로 뷰도 작성할 수 있습니다. 런타임으로 뷰에 액세스하면 DTS 패키지가 실행되고 DTS 패키지의 타스크의 목표 테이블이 작성된 뷰가 됩니다.

Data Warehouse Center에서 뷰를 작성한 후 다른 뷰와 마찬가지로 사용할 수 있습니다. 예를 들어, SQL 단계의 OLE DB 소스와 DB2 테이블을 조인할 수 있습니다. SQL 단계에서 작성된 뷰를 사용하면 DTS 제공자가 호출되고 DTS 패키지가 수행됩니다.

소프트웨어 요구사항

- 웨어하우스 목표 데이터베이스인 Windows NT용 DB2 Universal Database 버전 7.2
- DB2 Warehouse Manager 버전 7.2
- 버전 7.2보다 먼저 웨어하우스 목표 데이터베이스가 작성되면 Windows NT용 DB2 UDB 버전 7.2를 설치한 후 db2updv7 명령을 수행해야 합니다.
- 웨어하우스 소스 데이터베이스를 카탈로그화할 때 웨어하우스 에이전트 머신에서 데이터베이스 별명도 카탈로그화됩니다. 그러나 마법사를 시작할 때 Data Warehouse Center는 클라이언트 워크스테이션에도 데이터베이스 별명이 정의되어 있다고 가정하며 웨어하우스 소스 데이터베이스 사용자 ID 및 암호를 사용하여 연결하려고 시도합니다. 연결에 성공하면 마법사가 시작되어 뷰를 작성할 수 있습니다. 연결에 성공하지 못하면 경고 메시지가 표시되고 마법사에서 다른 데이터베이스 별명을 카탈로그화하거나 선택해야 합니다.
- DTS 패키지에서 특정 테이블을 식별하려면 목표 테이블을 작성하는 DataPumpTask의 워크플로 등록정보 창의 옵션 탭에서 **DSO 행 세트 제공자** 선택란을 선택해야 합니다. 복수 DSO 행 세트 제공자 속성을 설정하면 처음 선택된 단계의 결과만이 사용됩니다. 뷰가 선택되면 이 뷰의 목표 테이블의 행 세트가 리턴되며 다음 단계에서 작성한 다른 행 세트는 모두 무시됩니다.
- 마법사에 대한 테이블 이름을 입력할 때 task의 워크플로 등록정보 노트부의 옵션 페이지에 나타난 단계 이름을 사용합니다.
- DTS 패키지 연결 문자열에는 dtsrun 명령과 동일한 구문이 있습니다.

OLE DB 테이블 함수용 뷰 작성

OLE DB 테이블 함수에 대한 뷰를 작성하려면 다음을 수행하십시오.

1. Data Warehouse Center 창에서 웨어하우스 소스 트리를 확장합니다.
2. 뷰를 포함할 웨어하우스 소스를 확장합니다.
3. 뷰 폴더를 오른쪽 마우스 단추로 누르고 **OLE DB 테이블 함수용** 작성을 누릅니다.

OLE DB Assist 마법사가 열립니다. 마법사의 단계에 따라 웨어하우스 소스 데이터베이스에서 새로운 뷰를 작성하는 task를 수행합니다.

DTS 패키지용 뷰 작성

DTS 패키지용 뷰를 작성하려면 다음을 수행하십시오.

1. Data Warehouse Center 창에서 웨어하우스 소스 트리를 확장합니다.
2. 뷰를 포함할 웨어하우스 소스를 확장합니다.
3. 뷰 폴더를 오른쪽 마우스 단추로 누르고 **DTS 패키지용 Microsoft OLE DB Provider**를 누릅니다.

OLE DB Assist 마법사가 열립니다. 마법사의 단계에 따라 웨어하우스 소스 데이터베이스에서 새로운 뷰를 작성하는 작업을 수행합니다.

DTS에 대한 자세한 정보는 마법사가 DTS 제공자에 연결하기 위해 필요한 제공자 문자열을 빌드하는 방법에 대한 자세한 설명이 포함된 Microsoft Platform SDK 2000 문서를 참조하십시오.

바꾸기와 함께 증가 확약 사용

데이터 처리 유형이 바꾸기인 단계에서 증가 확약은 새 데이터가 삽입될 때만 사용됩니다. 이전 데이터는 단일 확약 범위에서 삭제됩니다. 로그 레코드를 작성하지 않고 데이터를 삭제해야 할 경우 추가 데이터 처리 유형으로 SQL 단계를 수행하기 전에 빈 작업을 로드하는 단계를 수행하십시오.

구성요소 추적 데이터 파일 이름

Data Warehouse Center는 Windows NT에서 다음과 같은 파일을 작성합니다.

AGNTnnnn.Log에는 추적 정보가 들어 있습니다. *nnnn*은 운영 체제에 따라 4자리 또는 5자리가 되는 숫자로 된 프로세스 ID입니다.

AGNTnnnn.Set에는 에이전트에 대한 환경 설정이 포함되어 있습니다. *nnnn*은 운영 체제에 따라 4자리 또는 5자리가 되는 숫자로 된 프로세스 ID입니다.

기본 디렉토리는 *x:\program files\sql\lib\logging*입니다. 여기서 *x*는 DB2가 설치되어 있는 드라이브입니다.

AIX 및 Solaris 운영 환경에서 Sybase 소스에 대해 필요한 Open Client

웨어하우스 소스 설정, AIX, 테이블 3. AIX에서 지원되는 데이터 소스에 대한 연결 요구사항, Solaris 운영 환경인 경우에는 테이블 4. 연결 Solaris 운영 환경에서 지원되는 데이터 소스에 대한 연결 요구사항에서 Sybase 항목의 "연결 방법:" 컬럼에는 추가 단계가 포함되어야 합니다. 추가 단계는 아래의 3 단계입니다.

3. Open Client 설치

Windows NT나 Windows 2000 플랫폼에서 Sybase 소스에 연결하기 위해 Open Client가 필요합니다.

정정된 샘플 항목

*Data Warehouse Center Administration Guide*의 "웨어하우스 소스 설정"의 그림 6, 8, 10 및 11에는 드라이버 속성의 경로가 잘못 지정되어 있습니다. 다음 경로가 맞습니다.

그림 6

Driver=/home/db2_07_01/3.6/odbc/lib/ivinf12.so

그림 8

Driver=/home/db2_07_01/3.6/odbc/lib/ivsyb1112.so

그림 10

Driver=/home/db2_07_01/3.6/lib/ivor814.so

그림 11

Driver=/home/db2_07_01/3.6/odbc/lib/ivmsss14.so

웨어하우스 소스 설정

Microsoft Access의 Memo 필드를 웨어하우스 소스에 맵핑

Microsoft Access 데이터베이스의 Memo 필드는 Data Warehouse Center에서 컬럼 크기가 1 GB를 초과하는 LONG VARCHAR 데이터 유형으로 표시됩니다. 실제 시스템 구성을 지원하기 위해, Data Warehouse Center는 128 KB를 초과하는 값을 절단합니다. 웨어하우스 소스에서 Memo 필드 값이 절단되는 것을 방지하기 위해, 단계에서 테이블을 사용하기 전에 Memo 필드 데이터를 받는 컬럼의 데이터 유형을 LONGVARCHAR에서 CLOB로 변경하십시오. 컬럼의 데이터 유형을 변경하지 않으면, 128 KB 보다 큰 값은 절단됩니다.

CLOB 데이터 유형을 위한 DRDA 지원이 OS/390 및 OS/400에 필요합니다. CLOB 데이터 유형은 DB2 버전 6부터 OS/390에 지원됩니다. CLOB 데이터 유형은 OS/400 버전 4, 릴리스 4 그리고 DB FixPak 4 이상(PTF SF99104)인 OS/400에 지원됩니다. OS/400의 경우, 날짜가 1999년 2월인 설치 디스크 버전 4, 릴리스 4에도 CLOB 유형에 대한 지원이 들어 있습니다.

웨어하우스 데이터베이스 유지보수

DB2 UDB RUNSTATS 프로그램에 대한 단계 부속 유형에 테이블 링크

RUNSTATS 프로그램에 대한 단계 부속 유형은 웨어하우스 목표로부터 읽고 씁니다. 단계에 대한 값을 정의하기 전에 프로세스 모델 창의 단계 부속 유형에 목표를 링크하십시오.

기본 웨어하우스 제어 데이터베이스

Windows NT나 Windows 2000에서 일반 DB2 설치를 수행하는 동안, DB2는 Windows NT 레지스트리에 활동 중인 웨어하우스 제어 데이터베이스가 식별되지 않았으면 Data Warehouse Center에 대한 기본 웨어하우스 제어 데이터베이스를 작성하고 초기화합니다. 초기화는 Data Warehouse Center가 Data Warehouse Center 메타 데이터를 저장하는 데 필요한 제어 테이블을 작성하는 프로세스입니다.

기본 웨어하우스 제어 데이터베이스의 이름은 DWCTRLDB로 지정됩니다. 로그인하면, Data Warehouse Center는 기본적으로 웨어하우스 제어 데이터베이스로서 DWCTRLDB를 지정합니다. 사용할 웨어하우스 제어 데이터베이스의 이름을 보려면, Data Warehouse Center 로그인 창에서 고급 버튼을 클릭하십시오.

웨어하우스 제어 데이터베이스 관리 창

웨어하우스 제어 데이터베이스 관리 창은 Windows NT나 Windows 2000에서 DB2 일반 설치를 수행하는 동안에 설치됩니다. 이 창을 사용하여 활동 중인 웨어하우스 제어 데이터베이스를 변경하고 새로운 웨어하우스 제어 데이터베이스를 작성 및 초기화하고 IBM Visual Warehouse와 함께 사용된 웨어하우스 제어 데이터베이스를 이주시킬 수 있습니다. 다음 절에서는 이들 각 활동을 설명합니다.

웨어하우스 제어 데이터베이스 관리 창을 사용하기 전에 웨어하우스 서버를 중지하십시오.

활동 중인 웨어하우스 제어 데이터베이스 변경

활동 중인 웨어하우스 제어 데이터베이스 이외의 웨어하우스 제어 데이터베이스를 사용하려면, 웨어하우스 제어 데이터베이스 관리 창을 사용하여 이 데이터베이스를 활동 중인 제어 데이터베이스로 등록하십시오. Data Warehouse Center에 로그인할 때 활동 중인 웨어하우스 제어 데이터베이스 이외의 이름을 지정하면, 지정한 데이터베이스가 웨어하우스 서버에서 지정한 데이터베이스와 일치하지 않음을 나타내는 오류를 받게 됩니다.

데이터베이스를 등록하려면, 다음과 같이 수행하십시오.

1. 시작 --> 프로그램 --> IBM DB2 --> Warehouse 제어 데이터베이스 관리를 누르십시오.
2. 새 제어 데이터베이스 필드에 사용하려는 제어 데이터베이스의 이름을 입력하십시오.
3. 스키마 필드에 데이터베이스에 대해 사용할 스키마의 이름을 입력하십시오.
4. 사용자 ID 필드에, 데이터베이스에 액세스하는 데 필요한 사용자 ID를 입력하십시오.

5. 암호 필드에 이 사용자 ID에 대한 암호를 입력하십시오.
6. 암호 확인 필드에 암호를 다시 입력하십시오.
7. 확인을 클릭하십시오.

창이 열린 상태로 남아 있습니다. 메시지 필드는 등록 프로세스의 상태를 나타내는 메시지를 표시합니다.

8. 프로세스가 완료된 후에 창을 닫으십시오.

웨어하우스 제어 데이터베이스 작성 및 초기화

기본값 이외의 웨어하우스 제어 데이터베이스를 작성하려면, 설치 프로세스 중이나 설치 이후에 웨어하우스 제어 데이터베이스 관리 창을 사용하여 작성할 수 있습니다. 설치 프로세스를 사용하여 웨어하우스 서버와 동일한 워크스테이션이나 다른 워크스테이션에서 데이터베이스를 작성할 수 있습니다.

설치 중에 작성되는 웨어하우스 제어 데이터베이스의 이름을 변경하려면, 사용자 정의 설치를 수행하고 지역 웨어하우스 제어 데이터베이스 정의 창에서 이름을 변경해야 합니다. 설치 프로세스는 지정한 이름으로 데이터베이스를 작성하고 Data Warehouse Center와 함께 사용하도록 데이터베이스를 초기화하고 활동 중인 웨어하우스 제어 데이터베이스로 데이터베이스를 등록합니다.

웨어하우스 서버가 설치되지 않은 워크스테이션에서 설치하는 동안 웨어하우스 제어 데이터베이스를 작성하려면, 사용자 정의 설치 중에 웨어하우스 지역 제어 데이터베이스를 선택하십시오. 설치 프로세스는 데이터베이스를 작성합니다. 설치 이후에 264 페이지의 『활동 중인 웨어하우스 제어 데이터베이스 변경』의 단계에 따라 웨어하우스 서버 워크스테이션에서 웨어하우스 제어 데이터베이스 관리 창을 사용해야 합니다. 설치 중에 지정한 데이터베이스 이름을 지정하십시오. 이 데이터베이스는 Data Warehouse Center와 함께 사용하도록 초기화되고 활동 중인 웨어하우스 제어 데이터베이스로 등록됩니다.

설치 프로세스 이후에 웨어하우스 제어 데이터베이스를 작성하고 초기화하려면, 웨어하우스 서버 워크스테이션에서 웨어하우스 제어 데이터베이스 관리 창을 사용하십시오. 새로운 웨어하우스 제어 데이터베이스가 웨어하우스 서버 워크스테이션에 없으면, 이 데이터베이스를 먼저 작성하고 웨어하우스 서버 워크스테이션에서 카탈로그화해야 합니다. 그런 다음, 264 페이지의 『활동 중인 웨어하우스 제어 데이터베이스 변경』의 단계를 따르십시오. 설치 중에 지정한 데이터베이스 이름을 지정하십시오.

Data Warehouse Center에 로그인할 때, 고급 버튼을 클릭하고 활동 중인 웨어하우스 제어 데이터베이스의 이름을 입력하십시오.

편집된 SQL 단계 작성

사용에 기초하여 편집된 SQL 단계를 작성할 때, 편집 삭제 성능을 높이기 위해 편집 컬럼에서 고유하지 않은 색인을 작성할 수 있습니다. 이 점은 대형 웨어하우스 테이블의 경우에만 고려하십시오. 삽입의 성능은 적은 수의 행을 삽입할 때 영향을 받을 수 있기 때문입니다.

프로세스 모델 편집기 창에서 소스 및 목표 변경

프로세스 모델 창에서, 소스나 목표를 변경할 경우, 변경한 사항은 즉시 자동으로 저장됩니다. 단계 추가와 같은 다른 변경 작업을 수행한 경우, 변경한 사항을 명시적으로 저장하여 변경사항을 영구적으로 보유해야 합니다. 변경사항을 저장하려면, 프로세스 —> 저장을 클릭하십시오.

Data Warehouse Center 오브젝트에 설명 추가

Data Warehouse Center에서 노트북의 설명 필드에 최대 254자를 지정할 수 있습니다. 이 최대값은 온라인 도움말에 지정된 최대 길이를 바꿉니다.

샘플 내용 수행

- 플랫폼 파일 소스에서 AS/400 에이전트를 사용하는 샘플 내용 요청을 성공적으로 수행할 수 없습니다. 플랫폼 파일을 작성하고 AS/400 에이전트를 사용하여 샘플 내용 요청을 발행하도록 시도할 수 있지만 이 요청은 실패합니다.
- 프로세스 모델 편집기를 통해 웨어하우스 목표에서 샘플 내용을 수행하면 오류가 발생할 수 있습니다. 이 오류는 웨어하우스 소스, 웨어하우스 목표 및 프로세스의 단계에 공통되는 에이전트 사이트의 사용 가능성과 관련이 있습니다. 단계에 대해 사용 가능한 에이전트 사이트의 목록은 웨어하우스 소스 IR 에이전트 사이트, 웨어하우스 목표 IR 에이전트 사이트 및 이 특정 단계에 사용 가능한 에이전트 사이트의 교차를 통해 확보합니다(단계는 에이전트 사이트 등록정보 노트북의 마지막 페이지에서 선택합니다). 예를 들어, FTP Put 프로그램을 수행하는 프로세스에 대해 샘플 내용을 보려고 합니다(VWPRCPY). 이 프로세스에서 사용되는 단계는 에이전트 사이트 정의에 있는 에이전트 사이트에 대해 선택되어야 합니다. 목표 파일에 대해 샘플 내용을 수행하면, 선택된 목록에서 주로 첫번째 에이전트 사이트가 사용됩니다. 그러나, 데이터베이스 유지보수 조작은 나열된 에이전트 사이트의 순서에 영향을 줄 수 있습니다. 선택된 에이전트 사이트가 소스나 목표 파일과 동일한 시스템에 상주하지 않으면 샘플 내용이 실패합니다.

Create DDL SQL문 편집

개발 모드의 단계에 대해 목표 테이블의 Create DDL SQL문을 편집하려고 하면, 다음의 잘못된 메시지가 표시됩니다. 『Create DDL SQL문의 변경사항은 테이블 정의나 실제 물리적 테이블에 반영되지 않습니다. 계속하시겠습니까?』

변경사항은 실제 물리적 테이블에 반영됩니다. 이 메시지를 무시하고 Create DDL 명령문의 변경을 계속하십시오.

개발 모드의 단계에 대해 이 메시지의 정정된 버전은 다음과 같아야 합니다. 『Create DDL SQL문의 변경사항은 테이블 정의에 반영되지 않습니다. 계속하시겠습니까?』

테스트 또는 생산 모드의 단계에서 메시지가 올바릅니다. Data Warehouse Center는 단계를 테스트 모드로 승격했을 때 작성된 물리적 목표 테이블을 변경하지 않습니다.

Visual Warehouse 비즈니스 뷰 이주

Visual Warehouse 메타데이터 동기화 비즈니스 뷰를 Data Warehouse Center로 이주하려면, 웨어하우스 제어 데이터베이스를 이주하기 전에 비즈니스 뷰를 생산 상태로 승격하십시오. 비즈니스 뷰가 생산 상태에 있으면, 해당되는 스케줄이 Data Warehouse Center로 이주됩니다. 비즈니스 뷰가 생산 상태에 있지 않으면, 스케줄 없이 테스트 상태로 이주됩니다. 이주된 단계들은 생산 상태로 승격할 수 없습니다. Data Warehouse Center에서 다시 동기화 단계들을 작성하고 이주된 단계들은 삭제해야 합니다.

목표 테이블 및 기본 키 생성

Data Warehouse Center가 단계에 대한 목표 테이블을 생성할 경우, 그 목표 테이블에 대한 기본 키를 생성하지 않습니다. 이동 평균과 같은 일부 변환기는 생성된 테이블을 소스 테이블로 사용하므로 소스 테이블이 기본 키를 가지고 있어야 합니다. 변환기에서 생성된 테이블을 사용하기 전에, DB2 제어 센터에서 테이블을 마우스 오른쪽 버튼으로 클릭하고 변경을 클릭하여 테이블에 대한 기본 키를 정의하십시오.

Merant ODBC 드라이버 사용

Merant ODBC 드라이버를 사용하여 Windows NT에서 Microsoft SQL Server에 액세스하려면, 시스템 경로에 sqllib\odbc32 디렉토리가 있는지 확인하십시오.

새로운 ODBC 드라이버

Merant ODBC 소스에 액세스하기 위해 링크된 Data Warehouse Center AIX 또는 Sun 에이전트를 사용하려 하고 DB2 데이터베이스에도 액세스하려면, .odbc.ini 파일의 DB2 소스 섹션에서 "Driver=" 속성의 값을 다음과 같이 변경하십시오.

AIX에서 드라이버 이름은 /usr/lpp/db2_07_01/lib/db2_36.o입니다.

AIX용 샘플 ODBC 소스 항목은 다음과 같습니다.

```
[SAMPLE] Driver=/usr/lpp/db2_07_01/lib/db2_36.o
Description=DB2 ODBC Database
Database=SAMPLE
```

Sun에서 드라이버 이름은 /opt/IBMdbs2/V7.1/lib/libdb2_36.so입니다.

Sun용 샘플 ODBC 소스 항목은 다음과 같습니다.

```
[SAMPLE] Driver=/opt/IBMdbs2/V7.1/lib/libdb2_36.so
Description=DB2 ODBC Database
Database=SAMPLE
```

OS/2 데이터베이스에서 웨어하우스 소스 또는 목표 정의

OS/2 데이터베이스에 대해 웨어하우스 소스나 웨어하우스 목표를 정의할 경우, 데이터베이스 이름을 대문자로 입력하십시오.

웨어하우스 제어 데이터베이스의 상태 모니터

DB2 제어 센터나 명령행 처리기는 웨어하우스 제어 데이터베이스가 불일치 상태에 있음을 나타낼 수 있습니다. 이것은 웨어하우스 서버가 해당되는 초기 시작 메시지를 웨어하우스 로그 프로그램에 대해 확약하지 않았음을 나타냅니다.

TBC_MD 샘플 데이터베이스에서 SQL 지원 사용

TBC_MD 데이터베이스에 포함된 데이터 웨어하우스 샘플에서, SQL 지원을 사용하여 시나리오 SQL 선택 단계에서 SQL을 변경할 수 없습니다. SQL이 SQL 지원에 의해 생성된 이후에 편집되었기 때문입니다.

FormatDate 함수 사용

FormatDate 함수를 사용하려면, SQL 단계에 대한 등록정보 노트북의 SQL문 페이지에서 SQL 빌드를 클릭하십시오.

FormatDate 함수 출력의 데이터 유형은 varchar(255)입니다. 함수 매개변수 - FormatDate 페이지의 범주 목록에서 날짜, 시간 또는 날짜/시간을 선택하여 데이터 유형을 변경할 수 없습니다.

언어 설정 변경

AIX 및 Solaris 운영 환경에서, 설치 프로세스는 정보 카탈로그에 대해 발행할 언어를 설정하고, OLAP Integration Server로 내보내기를 설정합니다. 설치 중에 설정된 언어 이외의 다른 언어로 이 함수들을 사용할 경우, 다음 명령을 한 행에 입력하여 다음의 소프트 링크를 작성하십시오.

AIX에서

```
/usr/bin/ln -sf /usr/lpp/db2_07_01/msg/locale/flgnxolv.str  
/usr/lpp/db2_07_01/bin/flgnxolv.str  
locale xx_yy 형식의 언어 로케일 이름
```

Solaris 운영 환경에서

```
/usr/bin/ln -sf /opt/IBMDB2/V7.1/msg/locale/flgnxolv.str  
/opt/IBMDB2/V7.1/bin/flgnxolv.str  
locale xx_yy 형식의 언어 로케일 이름
```

키 테이블 생성 변환기 사용

키 테이블 생성 변환기의 키 컬럼에서 값 갱신 옵션을 사용할 경우, 변환기는 키 값을 가지고 있지 않은 테이블의 해당 행들만 갱신합니다(즉, 값이 널(NULL)입니다). 추가 행이 테이블에 삽입될 경우, 키 값은 변환기를 다시 수행할 때까지 널(NULL)입니다.

이러한 문제점을 피하려면, 다음 방법을 사용하십시오.

- 변환기의 초기 수행 후에는 모든 값 대체 옵션을 사용하여 모든 행에 대해 키를 다시 작성하십시오.

데이터베이스에 연결 유지보수

웨어하우스 서버는 데이터베이스를 관리하는 DB2 서버가 중지되고 다시 시작될 때 지역 또는 원격 데이터베이스에 대한 연결을 유지보수하지 않습니다. DB2를 중지하고 다시 시작할 경우, 웨어하우스 서비스도 중지한 후 다시 시작하십시오.

원격 Data Warehouse Center 클라이언트 설정

DB2 Administration Client와 데이터 웨어하우징 도구를 설치하여 웨어하우스 서버를 포함하는 워크스테이션과 다른 워크스테이션에 Data Warehouse Center 관리 클라이언트를 설정할 경우, 웨어하우스 서버 워크스테이션이 클라이언트 워크스테이션에 대해 서비스 파일을 청취하는 TCP/IP 포트 번호를 추가해야 합니다. 다음과 같이 서비스 파일에 항목을 추가하십시오.

```
vwkernel          11000/tcp
```

VM용 DB2 웨어하우스 소스 정의

DRDA 게이트웨이를 통해 액세스되는 VM용 DB2 데이터베이스에 대한 웨어하우스 소스를 정의할 경우, CLOB 및 BLOB 데이터 유형의 사용에 대해 제한사항이 있습니다.

- 샘플 내용 함수로는 CLOB 및 BLOB 데이터 유형의 데이터를 볼 수 없습니다.
- SQL 단계에서 CLOB 및 BLOB 데이터 유형 컬럼을 사용할 수 없습니다.

이 제한사항은 LOB 오브젝트가 DRDA를 사용하여 DB2 버전 7 클라이언트로 전송될 수 없는 VM용 DB2 버전 5.2에 대한 알려진 제한사항입니다.

VM용 DB2 또는 VSE용 DB2 목표 테이블 정의

VM용 DB2나 VSE용 DB2 목표 테이블을 Data Warehouse Center에서 정의할 경우, 공용에 관한 부여 선택란을 선택하지 마십시오. Data Warehouse Center가 생성하는 GRANT 명령 구문은 VM용 DB2 및 VSE용 DB2에서 지원되지 않습니다.

분리 식별자 지원 사용

Windows NT에서 Sybase 및 Microsoft SQL Server에 대한 분리 식별자 지원이 가능하도록 하려면, ODBC 드라이버 설정 노트북의 고급 페이지에서 인용된 식별자 사용 가능 선택란을 선택하십시오.

UNIX에서 Sybase에 대한 분리 식별자 지원이 가능하게 하려면, .odbc.ini 파일에서 연결 속성 EQI=1을 포함하도록 Sybase 데이터 소스를 편집하십시오.

Data Joiner가 바인드 문제점을 지시합니다

DB2 버전 7.1 + FixPak 2 이상과 함께 DataJoiner를 사용하는 고객은 바인드 문제점을 알리는 오류를 받을 수 있습니다.

예를 들어, Data Warehouse Center 버전 7 에이전트에서 DataJoiner를 사용할 때, 다음과 같은 오류를 받을 수 있습니다.

```
DWC07356E An agent's processing of a command of type "importTableNames"
failed for edition "0" of step "?".
SQL0001N Binding or precompilation did not complete successfully.
SQL0001N Package "NULLID.SQLL6D05" was not found.
SQLSTATE=51002 RC = 7356 RC2 = 8600
```

문제점을 해결하려면, 다음 행을 db2cli.ini 파일에 추가하십시오.

```
[COMMON]
DYNAMIC=1
```

UNIX 시스템에서, db2cli.ini 파일은 ../sqllib/cfg 디렉토리에 있습니다. Windows NT에서, db2cli.ini 파일은 ../sqllib 디렉토리에 위치합니다.

Data Warehouse Center를 사용한 복제 설정 및 수행

1. Data Warehouse Center를 사용하여 복제를 설정하고 수행하려면 웨어하우스 제어 데이터베이스와 웨어하우스 목표 데이터베이스에 모두 복제 제어 테이블이 있어야 합니다.

복제를 수행하려면 제어 및 목표 데이터베이스에 모두 복제 제어 테이블이 있어야 합니다. 복제 제어 테이블은 ASN 스키마에서 찾을 수 있으며 모두 IBMSNAP로 시작합니다. 복제 제어 테이블은 제어 테이블이 아직 존재하지 않는 경우에 제어 센터를 통해 복제 소스를 정의하면 데이터베이스에서 자동으로 작성됩니다. 제어 테이블은 목표 DB에도 존재해야 한다는 점을 유의하십시오. 목표 DB에서 작성된 제어 테이블 세트를 얻으려면, 제어 센터를 사용하여 복제 소스를 작성한 다음, 제어 테이블을 제 위치에 그대로 두는 것만으로 복제 소스를 제거할 수 있습니다. 또는, DJRA, Data Joiner 복제 관리 제품을 사용하여 제어 테이블을 정의할 수도 있습니다.

2. DJRA 설치 및 사용

제어 테이블을 정의하기 위해 DJRA를 사용하기를 원하거나 사용해야 하는 경우, 이를 먼저 설치해야 합니다. DJRA는 DB2의 일부로 제공됩니다. DJRA를 설치하려면, d:\sqllib\djra 디렉토리(DB2가 설치된 위치)로 가서 djra.exe 패키지를 클릭하십시오. 그러면, 시스템에 DJRA가 설치됩니다. 그 후에 DJRA에 액세스하려면, Windows NT의 시작 메뉴에서 Windows NT용 DB2를 클릭하고 복제를 선택한 다음, 복제 관리 도구를 선택하십시오. DJRA 인터페이스는 보통 NT 응용 프로그램과 약간 다릅니다. 수행하는 각 기능을 위해 수행할 SQL 세트를 작성하지만 실행하지는 않습니다. 사용자는 생성된 SQL을 수동으로 저장하고 SQL을 수행하려면 Execute SQL 함수를 선택해야 합니다.

3. Capture 및 Apply를 수행하기 위한 설정

테스트하고 있는 시스템에 대한 복제 안내서 및 참조 설명서에서 Capture 및 Apply 프로그램을 수행하기 위한 시스템 구성에 대한 지시사항을 참조하십시오. Capture 및 Apply 프로그램을 사용할 각 데이터베이스에 바인드해야 합니다. 암호 파일은

작성할 필요가 '없다'는 점을 유의하십시오. Data Warehouse Center가 복제 복사 작업 내역을 위해 암호 파일을 자동으로 작성합니다.

4. 제어 센터에서 복제 소스 정의

제어 센터를 사용하여 복제 소스를 정의하십시오. Data Warehouse Center는 사용자 복사, 특정 시점, 기본 총계, 변경 총계 및 올림 테이블(CCD 테이블)의 5가지 유형의 복제를 지원합니다. 사용자 복사, 특정 시점 및 압축 올림 테이블 유형의 경우, 복제 소스 테이블에 기본 키가 있어야 합니다. 다른 복제 유형은 그렇지 않습니다. 복제 소스로 정의할 입력 테이블을 선택할 때 이 점을 기억해두십시오. 복제 소스는 실제로는 목표 테이블에 이동되기 전에 데이터 변경사항을 보유하기 위한 원본 소스 테이블 및 작성된 CD(데이터 변경)의 정의입니다. 제어 센터에서 복제 소스를 정의하면, 소스 및 해당 CD 테이블을 정의하기 위한 레코드가 ASN.IBMSNAP_REGISTER에 기록됩니다. 이 CD 테이블은 동시에 작성되지만 처음에는 아무 데이터도 없습니다. 복제 소스를 정의할 때 사후 이미지 컬럼만을 포함시키거나 사전 및 사후 이미지 컬럼을 모두 포함시키도록 선택할 수 있습니다. 이러한 선택은 제어 센터 복제 소스 인터페이스의 선택란을 통해 이루어집니다. 선택한 사전 및 사후 이미지 컬럼은 새로운 CD 테이블에 작성된 컬럼으로 변환됩니다. CD 테이블에서, 사후 이미지 컬럼의 이름은 원본 소스 테이블 컬럼 이름과 동일합니다. 사후 이미지 컬럼에는 컬럼 이름의 첫번째 문자로 'X'가 있습니다.

5. Data Warehouse Center에 복제 소스 가져오기

제어 센터에서 복제 소스를 작성했으면, 이를 Data Warehouse Center로 가져올 수 있습니다. 소스를 가져올 때에는 "복제할 수 있는 테이블"이라고 하는 선택란을 반드시 클릭하십시오. 이것은 복제 소스로 정의된 테이블을 확인하기 위해 ASN.IBMSNAP_REGISTER 테이블의 레코드를 보도록 Data Warehouse Center에 지시합니다.

6. Data Warehouse Center에서 복제 단계 정의

프로세스 모델 편집기에서, 5가지 복제 유형인 기본 총계, 변경 총계, 특정 시점, 올림 테이블 또는 사용자 복사 중 하나를 선택하십시오. 기본 총계 또는 변경 총계 복제 유형을 정의하려면, Data Warehouse Center에서 기본 총계 또는 변경 총계 복제를 설정하는 방법에 대한 아래의 절을 참조하십시오. 복제 유형에 대해 적합한 복제 소스를 선택하십시오. 위에 언급한 바와 같이 사용자 복사, 특정 시점 및 압축 올림 테이블 복제 유형의 경우에는 입력 소스에 기본 키가 있어야 합니다. 복제 소스를 복제 단계에 연결하십시오. 복제 단계에서 등록정보를 여십시오. 매개변수 탭으로 가십시오. 원하는 컬럼을 선택하십시오. 목표 테이블을 작성할 선택란을 선택하십시오. 웨어하우스 목표를 선택하십시오. 처리 옵션으로 가서 매개변수를 입력하십시오. 확인을 누르십시오.

7. Capture 프로그램 시작

DOS 창에서, ASNCCP source-database COLD PRUNE을 입력하십시오.

COLD 매개변수는 콜드 시동을 나타내며 CD 테이블에서 기존의 데이터를 삭제합니다. PRUNE 매개변수는 IBMSNAP_PRUNCNTL 테이블을 유지보수하도록 Capture 프로그램에 지시합니다. Capture 프로그램을 수행 중인 상태로 두십시오. 종료할 시간이 되면, DOS 창에서 Ctrl-Break를 사용하여 중지할 수 있습니다. Apply 프로그램을 시작하기 전에 Capture 프로그램을 시작해야 한다는 점을 기억하십시오.

8. 복제 단계 테스트로의 승격

다시 Data Warehouse Center에서, 정의된 복제 단계에 대해 단계를 테스트 모드로 승격하십시오. 그러면, 복제 복사 작업 내역 정보가 복제 제어 테이블에 기록됩니다. 복사 작업 내역을 지원하기 위한 IBMSNAP_SUBS_SET, IBMSNAP_SUBS_MEMBR, IBMSNAP_SUBS_COLS 및 IBMSNAP_SUBS_EVENT에 추가된 것을 확인하게 됩니다. 목표 테이블도 목표 데이터베이스에 작성됩니다. 복제 유형이 사용자 복사, 특정 시점 또는 압축 올림 테이블이면, 목표 테이블에서 기본 키가 필요합니다. 제어 센터로 가서 기본 키를 작성하십시오. 일부 복제 목표 테이블은 다양한 컬럼에서 고유 색인을 요구한다는 점도 유의하십시오. Data Warehouse Center에는 테이블 작성 시에 이들 고유 색인을 작성하기 위한 코드가 있어서 사용자 자신이 이들을 작성할 필요가 없습니다. 그렇지만 제어 센터에서 기본 키를 정의하며 이 컬럼에 대해 이미 고유 색인이 존재하는 경우, 기본 키를 작성하면 경고 메시지가 표시됩니다. 이 경고 메시지는 무시하십시오.

9. 복제 단계 생산으로의 승격

생산으로의 승격 중에는 복제 복사 작업 내역이 변경되지 않습니다. 이것은 다른 단계와 마찬가지로 엄격한 Data Warehouse Center 조작입니다.

10. 복제 단계 수행

복제 단계가 테스트 모드로 승격된 후에 수행할 수 있습니다. 소스 테이블을 변경하기 전에 초기 수행을 완료하십시오. 진행 중인 작업(WIP) 절로 가서 복제 단계를 선택하십시오. 이를 수행하십시오. 단계가 수행되면, IBMSNAP_SUBS_EVENT 테이블의 이벤트 레코드가 갱신되고 IBMSNAP_SUBS_SET의 복사 작업 내역 레코드가 활동 중인 것으로 게시됩니다. 복사 작업 내역을 즉시 수행해야 합니다. 복사 작업 내역이 수행되면, 에이전트는 Apply 프로그램을 호출하여 활동 중인 복사 작업 내역을 처리합니다. 이 시점 이후로 원본 소스 테이블을 갱신하면, 변경된 데이터가 CD 테이블로 이동됩니다. 그 이후에 Apply 프로그램의 재수행과 같이 복제 단계를 수행하면, 변경된 데이터는 CD 테이블에서 목표 테이블로 이동됩니다.

11. 복제 단계 테스트로의 강등

테스트로의 강등 중에는 복제 복사 작업 내역이 변경되지 않습니다. 이것은 다른 단계와 마찬가지로 엄격한 Data Warehouse Center 조작입니다.

12. 복제 단계 개발로의 강등

복제 단계를 개발로 강등시키면, 복사 작업 내역 정보가 복제 제어 테이블에서 제거됩니다. 개발로의 강등이 완료된 이후에 특정 복사 작업 내역에 대한 레코드는 복제 제어 테이블에 남아 있지 않습니다. 목표 테이블도 이 시점에서 삭제됩니다. CD 테이블은 복제 소스의 정의에 속하므로 제 위치에 남아 있습니다.

13. Data Warehouse Center에서 기본 총계 또는 변경 총계 복제를 설정하는 방법.
 - 입력 테이블. GROUP BY 명령문과 함께 사용할 수 있는 입력 테이블을 선택하십시오. 그 예로써 SALES, REGION, DISTRICT 컬럼이 있는 입력 테이블을 사용합니다.
 - 복제 단계. 기본 또는 변경 총계를 선택하십시오. 단계 등록정보를 여십시오.
 - Apply 프로그램이 수행되면, SELECT SUM(SALES), REGION, DISTRICT GROUP BY REGION, DISTRICT와 유사한 SELECT 문을 실행해야 합니다. 따라서 선택된 출력 컬럼에서 REGION, DISTRICT 및 하나의 계산 결과 컬럼인 SUM(SALES)를 선택해야 합니다. 계산 결과 컬럼 추가 버튼을 사용하십시오. 예에서는 표현식 필드에 SUM(SALES)을 입력하십시오. 이를 저장하십시오.
 - Where 절. GROUP BY 절만이 필요한 복제 단계를 설정할 때 1=1과 같은 더미 Where 절도 제공해야 한다는 복제 요구사항이 있습니다. 이 Where 절에는 단어 "WHERE"를 포함시키지 마십시오. 따라서, 기본 총계에 대한 Data Warehouse Center GUI에는 WHERE 절 입력 필드만이 있습니다. 예로써 이 필드에 1=1 GROUP BY REGION, DISTRICT를 입력하십시오. 변경 총계의 경우, Where 절과 GROUP BY 입력 필드가 둘 다 있습니다. Where 절 필드에 1=1을 입력하고 GROUP BY 필드에 GROUP BY REGION, DISTRICT를 입력하십시오.
 - 다른 복제 유형에서와 같이 나머지 단계 등록정보를 설정하십시오. 확인을 눌러 단계를 저장하고 목표 테이블 오브젝트를 작성하십시오.
 - 목표 테이블 오브젝트를 여십시오. 이제 계산 결과 컬럼 표현식에 대한 출력 컬럼의 이름을 유효한 컬럼 이름으로 바꾸고 컬럼에 대해 유효한 데이터 유형을 지정해야 합니다. 목표 테이블 오브젝트를 저장하십시오.
 - 복제 단계에서 테스트로의 승격을 수행하십시오. 목표 테이블이 작성됩니다. 기본 키는 필요하지 않습니다.
 - 다른 복제 단계와 같은 단계를 수행하십시오.

문제점 해결 추가 정보

- Apply 프로그램에 대한 추적을 작동시키려면, 웨어하우스 등록정보 패널에서 에이전트 추적 값 = 4를 설정하십시오. 에이전트는 에이전트 추적 = 4이면 Apply에 대한 완전한 추적을 작동시킵니다.

CD 테이블에서 데이터를 볼 수 없는 경우, Capture 프로그램이 시작되지 않았거나 일부 변경된 데이터를 작성하기 위해 원본 소스 테이블을 갱신하지 않았기 때문일 것입니다.

- 스케줄 노트북의 통지 페이지에 있는 메일 서버 필드가 온라인 도움말에서 누락되었습니다.
- 메일 서버는 Data Warehouse Center 통지를 작동시키기 위해 ESMTP를 지원해야 합니다. 진행 중인 작업 창 도움말에서, 웨어하우스 → 웨어하우스 센터 외에서 진행 중인 작업 → 진행 중인 작업을 클릭하십시오.

소스 및 목표에 액세스

다음의 표는 Data Warehouse Center가 지원하는 소스와 목표의 버전과 릴리스 레벨을 나열한 것입니다.

표 7. 지원되는 IBM 웨어하우스 소스의 버전 및 릴리스 레벨

소스	버전/릴리스
IMS	5.1
Windows NT용 DB2 Universal Database	5.2 - 7.1
DB2 Universal Database Enterprise-Extended Edition	5.2 - 7.1
OS/2용 DB2 Universal Database	5.2 - 7.1
AS/400용 DB2 Universal Database	3.7 - 4.5
AIX용 DB2 Universal Database	5.2 - 7.1
Solaris 운영 체제용 DB2 Universal Database	5.2 - 7.1
OS/390용 DB2 Universal Database	4.1 - 7.1
DB2 DataJoiner	2.1.1
VM용 DB2	5.3.4 이상
VSE용 DB2	7.1

소스	Windows NT	AIX
Informix	7.2.2 - 8.2.1	7.2.4 - 9.2.0
Oracle	7.3.2 - 8.1.5	8.1.5
Microsoft SQL Server	7.0	
Microsoft Excel	97	
Microsoft Access	97	
Sybase	11.5	11.9.2

표 8. 지원되는 IBM 웨어하우스 목표의 버전 및 릴리스 레벨

목표	버전/릴리스
Windows NT용 DB2 Universal Database	6 - 7
DB2 Universal Database Enterprise-Extended Edition	6 - 7

표 8. 지원되는 IBM 웨어하우스 목표의 버전 및 릴리스 레벨 (계속)

목표	버전/릴리스
OS/2용 DB2 Universal Database	6 - 7
AS/400용 DB2 Universal Database	3.1-4.5
AIX용 DB2 Universal Database	6 -7
Solaris 운영 체제용 DB2 Universal Database	6 -7
OS/390용 DB2 Universal Database	4.1 - 7
DB2 DataJoiner	2.1.1
DB2 DataJoiner/Oracle	8
VM용 DB2	3.4 - 5.3.4
VSE용 DB2	3.2, 7.1
CA/400	3.1.2

지원되는 IBM 이외 데이터베이스 소스에 대한 추가사항

다음 표에는 지원되는 IBM 이외 데이터베이스 소스에 대한 추가사항이 있습니다.

데이터베이스	운영 체제	데이터베이스 클라이언트 요구사항
Informix	AIX	Informix-Connect 및 ESQL/C 버전 9.1.4 이상
Informix	Solaris 운영 환경	Informix-Connect 및 ESQL/C 버전 9.1.3 이상
Informix	Windows NT	Windows 플랫폼용 Informix-Connect 2.x 또는 Windows 플랫폼용 Informix-Client Software Developer's Kit 2.x
Oracle 7	AIX	Oracle7 SQL*Net 및 Oracle7 SQL*Net 공유 라이브러리 (genclntsh 스크립트에 의해 빌드)
Oracle 7	Solaris 운영 환경	Oracle7 SQL*Net 및 Oracle7 SQL*Net 공유 라이브러리 (genclntsh 스크립트에 의해 빌드)
Oracle 7	Windows NT	현재 SQL*Net 버전에 적절한 DLL과 OCIW32.DLL. 예를 들어, SQL*Net 2.3에는 ORA73.DLL, CORE35.DLL, NLSRTL32.DLL, CORE350.DLL 및 OCIW32.DLL 이 필요합니다.
Oracle 8	AIX	Oracle8 Net8 및 Oracle8 SQL*Net 공유 라이브러리 (genclntsh8 스크립트에 의해 빌드)
Oracle 8	Solaris 운영 환경	Oracle8 Net8 및 Oracle8 SQL*Net 공유 라이브러리 (genclntsh8 스크립트에 의해 빌드)

데이터베이스	운영 체제	데이터베이스 클라이언트 요구사항
Oracle 8	Windows NT	<p>버전 8.0.3 레벨 이상에서 원격 Oracle8 데이터베이스 서버에 액세스하려면, Oracle Net8 클라이언트 버전 7.3.4.x, 8.0.4 또는 그 이상을 설치하십시오.</p> <p>Intel 시스템에서, 경로에 Oracle Net8 클라이언트에 대해 적절한 DLL(예: Ora804.DLL, PLS804.DLL 및 OCI.DLL)을 설치하십시오.</p>
Sybase	AIX	<p>DCE 이외 환경(ibsyb15 ODBC 드라이버)에서 libct library</p> <p>DCE 환경(ibsyb1115 ODBC 드라이버)에서 Sybase 11.1 client library libct_r</p>
Sybase	Solaris 운영 환경	<p>DCE 이외 환경(ibsyb15 ODBC 드라이버)에서 libct library</p> <p>DCE 환경(ibsyb1115 ODBC 드라이버)에서 Sybase 11.1 client library libct_r</p>
Sybase	Windows NT	Sybase Open Client-Library 10.0.4 이상과 적절한 Sybase Net-Library.

Data Warehouse Center에 수동으로 데이터 소스 작성

Relational Connect 및 『Create Nickname』 문을 사용하여 데이터 소스가 작성될 때, Data Warehouse Center에서 테이블 가져오기에 관련된 함수에서 데이터 소스를 사용할 수 없습니다. 데이터 소스를 소스 또는 목표 테이블로 사용하려면, 다음 단계를 수행하십시오.

1. 테이블을 가져오지 말고 소스/목표를 정의하십시오.
2. Data Warehouse Center의 기본 창에서 웨어하우스 소스/목표를 확장하고, 원하는 소스/목표에 대해 『테이블』을 마우스 오른쪽 버튼으로 클릭하십시오.
3. 정의를 클릭하십시오.
4. 열리는 노트북을 사용하여 데이터 소스를 정의하고 각 데이터 소스에 대해 컬럼이 정의되었는지 확인하십시오.

추가 정보는, 정보 센터에서 『웨어하우스 소스 테이블 정의』 또는 『웨어하우스 목표 테이블 정의』를 보십시오.

CWMI(Common Warehouse Metadata Interchange)를 사용하여 메타데이터 가져오기 및 내보내기

소개

태그 언어 파일에 대한 기존 지원에 추가하여, Data Warehouse Center는 이제 CWM(Common Warehouse Metamodel) 표준을 준수하는 XML 파일로부터 메타데이터를 가져오기 및 내보내기할 수 있습니다. 이들 CWM 준수 XML 파일 가져오기 및 내보내기를 CWMI(Common Warehouse Metadata Interchange)라고 합니다.

다음 Data Warehouse Center 오브젝트로부터 메타데이터를 가져오기 및 내보내기할 수 있습니다.

- 웨어하우스 소스
- 웨어하우스 목표
- 프로세스, 소스, 목표 및 단계를 포함한 주요 영역
- 사용자 정의 프로그램

CWMI 가져오기 및 내보내기 유틸리티는 현재 일정한 종류의 메타데이터는 지원하지 않습니다. 여기에는 스케줄, 웨어하우스 스키마, 단축 아이콘 단계, 연쇄 관계, 사용자 및 그룹이 포함됩니다.

Data Warehouse Center는 가져오기 및 내보내기 프로세스의 결과가 들어 있는 로그 파일을 작성합니다. 일반적으로, x:\program files\sql11lib\logging 디렉토리에(여기서 x:는 DB2를 설치한 드라이브) 또는 VWS_LOGGING 환경 변수로서 지정한 디렉토리에 로그 파일이 작성됩니다. 로그 파일은 일반 텍스트입니다. 어떤 텍스트 편집기로도 볼 수 있습니다.

메타데이터 가져오기

Data Warehouse Center 내에서 또는 명령행에서 메타데이터를 가져올 수 있습니다.

가져오기 프로세스를 통해 작성된 새 오브젝트는 기본 Data Warehouse Center 보안 그룹에 할당됩니다. 추가 정보는, 이들 릴리스 정보에서 "가져오기 후 보안 갱신"을 보십시오.

단계에 관한 메타데이터를 가져올 경우, 여러 파일이 단계와 연관될 수 있습니다. 단계에 관한 메타데이터는 XML 파일에 저장되지만, 때때로 단계에 BLOB으로 저장된 연관 데이터가 있습니다. BLOB 메타데이터의 파일 이름은 XML 파일과 같지만, 번호가

부여된 확장자가 있는 별도 파일로 되어 있습니다. 가져오기를 수행할 때 모든 관련된 단계 파일은 같은 디렉토리에 있어야 합니다.

테스트 또는 생산 모드에 있을 때 단계 갱신

단계가 개발 모드에 있어야 Data Warehouse Center가 단계의 메타데이터를 갱신할 수 있습니다. 단계가 테스트 또는 생산 모드에 있으면, 메타데이터를 가져오기 전에 단계를 개발 모드로 강등하십시오.

1. Data Warehouse Center에 로그인하십시오.
2. 강등하려는 단계를 마우스 오른쪽 버튼으로 클릭하고, 모드를 클릭하십시오.
3. 개발을 클릭하십시오.

이제 단계가 개발 모드에 있습니다. 메타데이터를 가져온 후 단계를 테스트 또는 생산 모드로 되돌리십시오.

Data Warehouse Center에서 데이터 가져오기

Data Warehouse Center 내에서 메타데이터를 가져올 수 있습니다.

1. Data Warehouse Center에 로그인하십시오.
2. 왼쪽 페인에서, 웨어하우스를 클릭하십시오.
3. 선택 항목 -> 메타데이터 가져오기 -> 파일 교환...을 누르십시오.
4. 메타데이터 가져오기 창에서, 가져오려는 메타데이터가 들어 있는 파일 이름을 지정하십시오. 파일 이름을 입력하거나 파일을 열람할 수 있습니다.
 - 위치를 알 경우, 가져오려는 파일의 완전한 경로와 파일 이름을 입력하십시오. 메타데이터를 XML 형식으로 가져오도록 지정하려면 .xml 파일 확장자를 포함시키십시오.
 - 파일을 찾아보려면 다음을 수행하십시오.
 - a. 생략(...) 누름 버튼을 클릭하십시오.
 - b. 파일 창에서, 파일 유형을 XML로 변경하십시오.
 - c. 올바른 디렉토리로 가서 가져오려는 파일을 선택하십시오.

주: 파일 확장자는 .xml이어야 합니다.

- d. 확인을 클릭하십시오.
5. 메타데이터 가져오기 창에서, 확인을 눌러 완료하십시오. Data Warehouse Center가 파일을 가져오는 동안 진행 창이 표시됩니다.

명령행을 사용하여 메타데이터 가져오기

또한 명령행을 사용하여 메타데이터를 가져올 수 있습니다. 다음은 가져오기 명령 구문입니다.

CWMImport XML_file dwcControlDB dwcUserId dwcPW [PREFIX = DWCtbschema]

XML_file	가져오려는 XML 파일의 완전한 경로 및 파일 이름 (드라이브와 디렉토리 포함). 이 매개변수는 필수입니다.
dwcControlDB	가져온 메타데이터를 저장할 웨어하우스 제어 데이터베이스의 이름. 이 매개변수는 필수입니다.
dwcUserId	웨어하우스 제어 데이터베이스에 로그인하기 위해 사용하는 사용자 ID. 이 매개변수는 필수입니다.
dwcPW	웨어하우스 제어 데이터베이스에 로그인하기 위해 사용하는 암호. 이 매개변수는 필수입니다.
[PREFIX=DWCtbschema]	Data Warehouse Center 시스템 테이블에 대한 데이터베이스 스키마 이름. 때로 테이블 접두부라고 합니다. PREFIX=에 값을 지정하지 않으면, 기본 스키마 이름은 IWH입니다. 이 매개변수는 선택적입니다.

가져오기 유틸리티 수행 후 메타데이터 갱신

가져오기 후 보안 갱신

보안상의 문제로, Data Warehouse Center는 암호를 가져오거나 내보내지 않습니다. 필요하다면 새 오브젝트에서 암호를 갱신해야 합니다. 가져오기 고려사항에 관한 세부사항은, *Data Warehouse Center 관리 안내서*, "Data Warehouse Center 메타데이터 내보내기 및 가져오기"를 보십시오.

메타데이터를 가져올 때, 모든 오브젝트는 기본 보안 그룹에 할당됩니다. 오브젝트에 액세스 권한을 소유하는 그룹을 변경할 수 있습니다.

1. Data Warehouse Center에 로그인하십시오.
2. 변경하려면 오브젝트가 들어 있는 폴더를 마우스 오른쪽 버튼으로 클릭하십시오.
3. 등록정보를 클릭한 뒤, 보안 탭을 클릭하십시오.
4. 선택된 웨어하우스 그룹 목록에서 그룹을 제거하거나 사용 가능한 웨어하우스 그룹 목록에 그룹을 추가하십시오.
5. 확인을 클릭하십시오.

메타데이터 내보내기

Data Warehouse Center 내에서 또는 명령행에서 메타데이터를 내보낼 수 있습니다.

일부 단계에는 BLOB으로 저장된 메타데이터가 있습니다. BLOB 메타데이터는 단계의 XML 파일과 같은 이름의 별도 파일로 내보내지지만, 번호가 부여된 확장자가 붙습니다(.1, .2 등등).

Data Warehouse Center에서 데이터 내보내기

Data Warehouse Center 내에서 메타데이터를 내보낼 수 있습니다.

1. Data Warehouse Center에 로그인하십시오.
 2. 왼쪽 페인에서, 웨어하우스를 클릭하십시오.
 3. 선택 항목 -> 메타데이터 내보내기-> 교환 파일을 누르십시오.
 4. 메타데이터 내보내기 창에서, 내보내려는 메타데이터가 들어 있는 파일 이름을 지정하십시오. 파일 이름을 입력하거나 파일을 찾아볼 수 있습니다.
 - 사용할 파일의 완전한 경로와 파일 이름을 알 경우, 파일 이름 입력 필드에 입력하십시오. 메타데이터를 XML 형식으로 내보내도록 지정하려면 .xml 파일 확장자를 포함시키십시오.
 - 파일을 찾아보려면 다음을 수행하십시오.
 - a. 생략(...) 누름 버튼을 클릭하십시오.
 - b. 파일 창에서, 파일 유형을 XML로 변경하십시오.
 - c. 올바른 디렉토리로 가서 내보낸 메타데이터를 보관할 파일을 선택하십시오.
- 주: 선택한 기존 파일은 내보내기된 메타데이터로 겹쳐 씌어집니다.
- d. 확인을 클릭하십시오.
5. 메타데이터 내보내기 창에 올바른 파일 이름이 표시될 때, 메타데이터를 내보낼 사용 가능한 오브젝트 목록에서 오브젝트를 클릭하십시오.
 6. > 기호를 클릭하여 선택된 오브젝트를 사용 가능한 오브젝트 목록에서 선택된 오브젝트 목록으로 이동하십시오. 내보내고자 하는 모든 오브젝트가 선택된 오브젝트 목록에 속할 때까지 반복하십시오.
 7. 확인을 클릭하십시오.

Data Warehouse Center는 내보내기 위해 선택한 Data Warehouse Center 오브젝트에 관한 정보가 들어 있는 입력 파일을 작성한 후, 그 오브젝트에 관한 메타데이터를 내보냅니다. Data Warehouse Center가 메타데이터를 내보내는 동안 진행 창이 표시됩니다.

내보내기 프로세스가 완료되면, 내보내기 프로세스에 관한 정보메시지를 받게 됩니다. 리턴 코드 0은 내보내기가 성공적임을 의미합니다. 또한 보다 상세한 정보를 알기 위해 로그 파일을 볼 수 있습니다.

명령행을 사용하여 메타데이터 내보내기

명령행에서 메타데이터를 내보내려면, 먼저 입력 파일을 작성해야 합니다. 입력 파일은 확장자가 .INP인 텍스트 파일이고, 내보내려는 오브젝트 유형 별로 모든 오브젝트를 나열합니다. Data Warehouse Center 내에서 내보내기를 수행할 때는 입력파일이 자동으로 작성되지만, 명령행에서 내보내기를 수행할 때는 먼저 입력 파일을 작성해야 합니다. 어떤 텍스트 편집기로도 입력 파일을 작성할 수 있습니다. Data Warehouse Center

에 나타나는 대로 모든 오브젝트 이름을 입력하십시오. 반드시 파일을 읽기/쓰기 디렉토리에 작성하십시오. 내보내기 유틸리티를 수행할 때, Data Warehouse Center는 입력 파일이 있는 곳과 같은 디렉토리에 XML 파일을 씁니다.

다음은 샘플 입력 파일입니다.

```
<PROC>
Tutorial Fact Table Process
<IR>
Tutorial file source
Tutorial target
<UDP>
New Program group
```

<PROC>(processes) 섹션에, 내보낼 모든 프로세스를 나열하십시오. <IR>(정보 자원) 섹션에, 내보낼 모든 웨어하우스 소스 및 목표를 나열하십시오. Data Warehouse Center는 이들 소스 및 목표와 연관된 테이블과 컬럼을 자동으로 포함시킵니다. <UDP>(사용자 정의 프로그램) 섹션에서, 내보낼 모든 프로그램 그룹을 나열하십시오.

메타데이터를 내보내려면, DOS 명령 프롬프트에서 다음 명령을 입력하십시오.

```
CWMSExport INPcontrol_file dwcControlDB dwcUserID dwcPW
[PREFIX=DWCtbschema]
```

<i>INPcontrol_file</i>	내보내려는 오브젝트가 들어 있는 .INP 파일의 완전한 경로 및 파일 이름(드라이브와 디렉토리 포함). 이 매개변수는 필수입니다.
<i>dwcControlDB</i>	오브젝트를 가져올 웨어하우스 제어 데이터베이스의 이름. 이 매개변수는 필수입니다.
<i>dwcUserID</i>	웨어하우스 제어 데이터베이스에 로그인하기 위해 사용하는 사용자 ID. 이 매개변수는 필수입니다.
<i>dwcPW</i>	웨어하우스 제어 데이터베이스에 로그인하기 위해 사용하는 암호. 이 매개변수는 필수입니다.
<i>[PREFIX=DWCtbschema]</i>	Data Warehouse Center 시스템 테이블에 대한 데이터베이스 스키마 이름. 때로 테이블 접두부라고 합니다. PREFIX=에 값을 지정하지 않으면, 기본값은 IWH입니다. 이 매개변수는 선택적입니다.

OS/390 Runstats 유틸리티 단계

DWC에서 OS/390 Runstats 유틸리티 단계를 정의할 때, 사용자는 단계 등록 정보 대화상자의 매개변수 탭에서 다음을 알아야 합니다. 테이블 공간 필드에, 사용자는 이름을 대문자로 입력해야 합니다. 테이블 공간이 DSNCB04에 있지 않다면, 테이블 공간 이름이 들어 있는 데이터베이스에 맞게 규정해야 합니다. 예를 들어, SAMPLE.EMPLOYEE를 입력하십시오. 현재 도움말은 이 필드에 대해 언급하지 않습니다.

OS/390 로드 유틸리티 단계

DWC에서 OS/390 로드 유틸리티 단계를 정의할 때, 사용자는 단계 등록 정보 대화상자의 매개변수 탭에서 다음을 알아야 합니다. 로드가 작동하려면, 사용자가 항상 고급 버튼을 선택해야 합니다. 선택하지 않으면 로드 문의 INTO 절이 생성되지 않아서 로드가 수행될 때 실패합니다.

또한, FixPak 3에는 로드 데이터 세트를 둘러싸는 큰 따옴표를 제거하는 수정사항이 포함되어 있습니다. 이 수정사항이 없으면, 로드는 작동하지 않습니다.

CWM(Common Warehouse Metamodel) XML 지원

버전 7.2 CWM 툴킷은 JDK 1.2.2 또는 1.3에서 작동됩니다.

다음 CWM XML 오브젝트를 가져오거나 내보낼 수 있습니다.

다른 프로세스의 단축 아이콘 단계

다른 프로세스(단축 아이콘)의 단계와 관련된 단계가 포함된 프로세스를 내보내면 두 프로세스 모두 내보내지며 관계도 유지됩니다.

조건부 연쇄 관계

CHILD, SUCCESS, FAILURE 및 UNCONDITIONAL이 포함된 단계 사이의 서로 다른 연쇄 관계를 가져오거나 내보낼 수 있습니다.

뷰 오브젝트로서의 웨어하우스 소스

내보낼 때 웨어하우스 소스를 뷰 오브젝트로 정의할 수 있습니다. 뷰 오브젝트는 테이블 오브젝트와 동일하게 처리됩니다.

컬럼 및 필드에 대한 SQLDataType

컬럼 및 필드에 대해 SQLDataType을 사용할 수 있습니다.

동일한 테이블에 대한 복수 상관 이름

가져오거나 내보내는 동안 여러 개의 상관 이름을 사용할 수 있습니다. 이 때 각각의 상관 이름은 동일한 테이블에 대해 자체 컬럼을 맵핑합니다.

새로운 SAP 및 WSA(WebSphere Site Analyzer) 소스 지원

새 소스 지원 태그가 추가되어 이제 웨어하우스에 저장된 SAP 및 WSA 정보를 내보낼 수 있습니다.

프로세스 모델 편집기

화면에 맞게 프로세스 모델 편집기 파레트 크기를 조정할 수 있습니다. 파레트의 아이콘은 복수 컬럼 파레트에 맞게 자동으로 위치가 조정됩니다.

파레트 아이콘을 누르면 파레트 오브젝트에 제목이 나타납니다.

보기 메뉴에서 비즈니스 이름 표시 메뉴를 선택하여 테이블 및 파일 오브젝트를 볼 수 있습니다. 백분을 설정을 조절하고 확대/축소 옵션을 선택하여 프로세스 뷰의 크기를 변경할 수 있습니다.

파레트 내에서 오브젝트가 겹치면 오브젝트를 눌러 화면 최상단에 표시할 수 있습니다. 또한 파레트 공간을 절약하기 위해 오브젝트 이름이 여러 행으로 표시됩니다.

Delete 키를 사용하여 오브젝트를 제거할 수 있습니다. 프로세스를 저장하면 테이블 변경 내용도 저장됩니다. 선택 동작은 자동이 아닙니다. 웨어하우스 소스 또는 목표 및 프로세스에서 테이블, 파일 또는 뷰를 제거하려면 오브젝트가 웨어하우스 소스에 있는 경우에는 오른쪽 마우스 단추를 누르고 소스에서 제거 조치를 선택하거나 오브젝트가 웨어하우스 목표에 있는 경우에는 목표에서 제거 조치를 선택합니다.

커서는 파레트 선택 상태를 표시합니다. 또한 상태 표시줄에는 커서가 위치한 오브젝트 이름이 나타납니다.

스키마 모델 편집기

스키마 모델 편집기에서 테이블을 최소화하거나 최대화할 수 있습니다. 테이블을 최소화하면 아이콘으로 변합니다. 더 정확하게 표시하기 위해 스타 스키마 배치를 작성할 수 있습니다.

필수 필드

Data Warehouse Center는 필수 필드에 빨강색 경계선을 표시합니다. 빨강색 경계선은 Data Warehouse Center 오브젝트를 정의하기 위해 필요한 데이터베이스 이름, 사용자 ID 또는 암호 등의 필수정보를 상기시킵니다. 필수 정보를 입력하면 경계선이 사라집니다.

Data Warehouse Center 런치패드 항상

런치패드의 Data Warehouse Center 오브젝트를 작성하면 탐색 트리가 확장되어 새 오브젝트 위치가 표시됩니다.

단계 정보를 파일로 인쇄

단계에 대한 정보(주제 영역, 소스 테이블 이름 및 목표 테이블 이름 등)를 텍스트 파일로 인쇄할 수 있습니다. 단계 정보를 파일로 인쇄하려면 프로세스 모델 편집기에서 단계 아이콘을 오른쪽 마우스 단추로 누르고 인쇄 → 파일로 인쇄를 누른 후 정보를 인쇄할 파일 이름을 지정합니다.

Data Warehouse Center 응용프로그램 통합 안내서

메타데이터 템플릿에서 테이블 16은 컬럼 태그 토큰을 설명합니다. 참조서의 정보에 "*ColumnPositionNumber"가 "1"로 시작되어야 함이 서술되어 있어야 합니다. 참조서가 시작 문자를 "0"으로 잘못 제공하고 있습니다.

뒤의 테이블 42에서 해당하는 SourceDataBase.tag의 DatabaseType 토큰에 대해 지정된 유형이 ISV_IR_FFLLan인 경우 TableTypeIfFile 토큰이 필요합니다. 지정되지 않은 경우에는 오류가 발생합니다.

Data Warehouse Center 메타데이터에서, POSNO 컬럼 오브젝트 등록정보에 대한 설명은 다음과 같이 변경되어야 합니다.

색인으로서, 1에서부터 시작하며, 테이블 또는 파일의 행에 있는 컬럼 또는 필드의 색인입니다.

Information Catalog Manager 오브젝트 유형에서, 오브젝트 유형을 정의하기 위한 태그 언어가 포함된 .TYP 파일을 찾을 수 있는 디렉토리가 \SQLLIB\DGWIN\TYPES로 변경되었습니다.

추가 메타데이터 템플릿

메타데이터 템플릿에 다음 메타데이터 템플릿이 포함되어야 합니다.

표 9. Data Warehouse Center와 함께 제공된 새 메타데이터 템플릿

템플릿	설명	참조:
commit.tag		『Commit.tag』
Foreignkey.tag		288 페이지의 『ForeignKey.tag』
Foreignkeyadditional.tag		290 페이지의 『ForeignKeyAdditional.tag』
Primarykey.tag		292 페이지의 『PrimaryKey.tag』
Primarykeyadditional.tag		293 페이지의 『PrimaryKeyAdditional.tag』

Commit.tag

이 템플릿을 사용하여 대형 태그 언어 파일을 사용할 때 성능을 개선합니다. 확장 템플릿을 여기서 설명하는 템플릿 그룹 사이에 삽입할 수 있습니다. 확장 템플릿은 그룹 내의 템플릿 사이에는 삽입할 수 없습니다. 예를 들어, AgentSite.tag와 VWPGroup.tag 사이에 확장 템플릿을 삽입하는 것은 유효하지만 VWPProgramTemplate.tag와 VWPProgramTemplateParameter.tag 사이에 확장 태그를 삽입하는 것은 유효하지 않습니다. 확장 템플릿이 잘못 사용되면 가져오기에서 오류를 보고할 수 있습니다.

- AgentSite.tag
- VWPGroup.tag
- VWPProgramTemplate.tag, VWPProgramTemplateParameter.tag
- SourceDatabase.tag
- WarehouseDatabase.tag
- Table.tag, Column.tag
- SubjectArea.tag
- Process.tag
- Step.tag, StepInputTable.tag, StepOutputTable.tag, StepVWPOutputTable.tag, StepVWPProgramInstance.tag, VWPProgramInstanceParameter.tag
- StepCascade.tag
- StarSchema.tag, StarSchemaInputTable.tag
- PrimaryKey.tag, PrimaryKeyAdditional.tag
- ForeignKey.tag, ForeignKeyAdditional.tag

확약 템플리트를 사용하는 것은 선택적입니다.

토큰

표10은(는) 템플리트의 각 토큰에 대한 정보를 제공합니다.

표 10. Commit.tag 토큰

토큰	설명	허용되는 값
	관계 매개변수	
*CurrentCheckPointID++	0부터 시작되는 색인으로, 토큰에서 대 체될 때마다 증가합니다.	숫자 값
	이 토큰은 필수입니다.	

값의 예

표11은(는) 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 나타내는 각 토큰에 대한 값의 예를 보여줍니다.

표 11. Commit.tag 토큰 값 예

토큰	값의 예
*CurrentCheckPointID++	1

ForeignKey.tag

이 템플리트를 사용하여 테이블의 외부 키 제한조건을 정의합니다. ForeignKey.tag 템플리트는 제한조건이 정의되는 컬럼 및 테이블과의 관계를 정의합니다. 이 템플리트는 또한 참조되는 기본 키의 컬럼 및 테이블과의 관계도 정의합니다. ForeignKey.tag 템

플리트를 사용하기 전에 외부 키 제한조건을 정의할 테이블과 컬럼(Table.tag 및 Column.tag 템플릿 사용) 및 기본 키 제한조건(PrimaryKey.tag 템플릿 사용)을 정의해야 합니다.

토큰

표12은(는) 템플릿의 각 토큰에 대한 정보를 제공합니다.

표 12. ForeignKey.tag 토큰

토큰	설명	허용되는 값
엔터티 매개변수		
*ConstraintName	제한조건 이름 이름은 테이블이나 필드에서 고유해야 합니다. 이 토큰은 필수입니다.	텍스트 문자열. 최대 80 바이트
*ForeignKeyColumnName	외부 키 제한조건이 정의되는 컬럼 이름	텍스트 문자열. 최대 254 바이트
*ForeignKeyID	외부 키를 고유하게 식별하는 키. 키 수는 태그 언어 파일의 다른 모든 키와 달라야 합니다. 팁: 키 값을 증가시키기 전에 ForeignKeyAdditional.tag 템플릿 처리를 완료하십시오. 이 토큰은 필수입니다.	숫자 값
*MapID	교환 파일의 다른 모든 키와 다른 임의의 숫자 팁: 이 토큰의 값을 증가시키기 전에 ForeignKey.tag 템플릿 처리를 완료하십시오. 이 토큰은 필수입니다.	숫자 값
*PrimaryColumnName	참조된 컬럼의 컬럼 이름	텍스트 문자열. 최대 80 바이트
*ReferencedPrimaryKeyID	기본 키를 고유하게 식별하는 키. 키 수는 태그 언어 파일의 다른 모든 키와 달라야 합니다. 팁: 키 값을 증가시키기 전에 ForeignKey.tag 템플릿 처리를 완료하십시오. 이 토큰은 필수입니다.	숫자 값
관계 매개변수		
*DatabaseName	웨어하우스 소스 또는 웨어하우스 목표의 비즈니스 이름 이 토큰은 필수입니다.	텍스트 문자열. 최대 40 바이트

표 12. ForeignKey.tag 토큰 (계속)

토큰	설명	허용되는 값
*ForeignTablePhysicalName	다른 테이블의 키를 참조하는 외부 키를 포함하는 실제 테이블의 데이터베이스 정의 이름	텍스트 문자열. 최대 254 바이트
*PrimaryTablePhysicalName	외부 키에 의해 참조되는 키를 포함하는 실제 테이블의 데이터베이스 정의 이름	텍스트 문자열. 최대 80 바이트
*PrimaryTableOwner	참조되는 기본 키 컬럼을 포함하는 테이블의 스키마, 소유자, 고급 규정자 또는 콜렉션	텍스트 문자열. 최대 128 바이트
	이 토큰은 필수입니다.	
*ForeignTableOwner	외부 키 제한조건 컬럼을 포함하는 테이블의 스키마, 소유자, 고급 규정자 또는 콜렉션	텍스트 문자열. 최대 128 바이트
	이 토큰은 필수입니다.	

값의 예

표13은(는) 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 나타내는 각 토큰에 대한 값의 예를 보여줍니다.

표 13. ForeignKey.tag 토큰 값 예

토큰	값의 예
*ConstraintName	부서
*DatabaseName	재정 웨어하우스
*ForeignColumnName	Country_code
*ForeignKeyID	07011
*ForeignTablePhysicalName	지리
*MapID	02568
*PrimaryColumnName	State_code
*Referenced PrimaryKeyID	이름
*PrimaryTablePhysicalName	도시
*PrimaryTableOwner	DB2ADMIN
*ForeignTableOwner	IWH

ForeignKeyAdditional.tag

이 템플릿을 사용하여 합성 외부 키를 정의합니다. ForeignKeyAdditional.tag 템플릿을 사용하기 전에 첫번째 컬럼에서 제한조건을 정의해야 합니다(ForeignKey.tag 템플릿 사용). 그런 다음 추가할 각 컬럼에 대해 이 템플릿을 사용하여 컬럼을 추가할 수 있습니다.

토큰

표14은(는) 템플리트의 각 토큰에 대한 정보를 제공합니다.

표 14. ForeignKeyAdditional.tag 토큰

토큰	설명	허용되는 값
엔터티 매개변수		
*ForeignColumnName	외부 키 제한조건이 정의되는 컬럼 이름	텍스트 문자열. 최대 80 바이트
*ForeignKeyID	외부 키를 고유하게 식별하는 키. 키는 태그 언어 파일의 다른 모든 키와 달라야 합니다. 팁: 키 값을 증가시키기 전에 ForeignKey.tag 템플리트 처리를 완료하십시오. 이 토큰은 필수입니다.	숫자 값
*MapID	교환 파일의 다른 모든 키와 다른 의미의 숫자 팁: 이 값을 증가시키기 전에 ForeignKeyAdditional.tag 템플리트 처리를 완료하십시오. 이 토큰은 필수입니다.	숫자 값
*MapSeqNo	외부 키 제한조건에 합성 키의 일부로 추가된 각각의 추가 컬럼을 나타내는 숫자	2부터 시작되는 증가하는 연속적인 고유한 숫자
*PrimaryColumnName	참조된 컬럼의 컬럼 이름	텍스트 문자열. 최대 80 바이트
관계 매개변수		
*DatabaseName	웨어하우스 소스 또는 웨어하우스 목표의 비즈니스 이름 이 토큰은 필수입니다.	텍스트 문자열. 최대 40 바이트
*ForeignTablePhysicalName	다른 테이블의 키를 참조하는 외부 키를 포함하는 실제 테이블의 데이터베이스 정의 이름	텍스트 문자열. 최대 80 바이트
*PrimaryTablePhysicalName	외부 키에 의해 참조되는 키를 포함하는 실제 테이블의 데이터베이스 정의 이름	텍스트 문자열. 최대 80 바이트
*PrimaryTableOwner	참조되는 기본 키 컬럼을 포함하는 테이블의 스키마, 소유자, 고급 규정자 또는 컬렉션 이 토큰은 필수입니다.	텍스트 문자열. 최대 128 바이트
*ForeignTableOwner	외부 키 제한조건 컬럼을 포함하는 테이블의 스키마, 소유자, 고급 규정자 또는 컬렉션 이 토큰은 필수입니다.	텍스트 문자열. 최대 128 바이트

값의 예

표15은(는) 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 나타내는 각 토큰에 대한 값의 예를 보여줍니다.

표 15. *ForeignKeyAdditional.tag* 토큰 값 예

토큰	값의 예
<i>*DatabaseName</i>	재정 웨어하우스
<i>*ForeignKeyColumnName</i>	Country_code
<i>*ForeignKeyID</i>	07011
<i>*ForeignKeyPhysicalName</i>	지리
<i>*MapID</i>	22578
<i>*MapSeqNo</i>	2
<i>*PrimaryColumnName</i>	State_code
<i>*PrimaryTablePhysicalName</i>	도시
<i>*PrimaryTableOwner</i>	DB2ADMIN
<i>*ForeignKeyTableOwner</i>	IWH

PrimaryKey.tag

이 템플리트를 사용하여 테이블의 기본 키 제한조건을 정의합니다. 이 템플리트는 또한 제한조건이 정의되는 컬럼 및 테이블과의 관계도 정의합니다. *PrimaryKey.tag* 템플리트를 사용하기 전에 기본 키 제한조건을 정의할 테이블과 컬럼(*Table.tag* 및 *Column.tag* 템플리트 사용)을 정의해야 합니다.

토큰

표16은(는) 템플리트의 각 토큰에 대한 정보를 제공합니다.

표 16. *PrimaryKey.tag* 토큰

토큰	설명	허용되는 값
	엔터티 매개변수	
<i>*ColumnName</i>	필드나 컬럼 이름 이름은 테이블이나 필드에서 고유해야 합니다. 이 토큰은 필수입니다.	텍스트 문자열. 최대 80 바이트
<i>*MapID</i>	교환 파일의 다른 모든 키와 다른 임 숫자 값의 숫자 팁: 이 토큰의 값을 증가시키기 전에 <i>PrimaryKey.tag</i> 템플리트 처리를 완료하십시오. 이 토큰은 필수입니다.	

표 16. PrimaryKey.tag 토큰 (계속)

토큰	설명	허용되는 값
*PrimaryKeyID	기본 키를 고유하게 식별하는 키. 키는 태그 언어 파일의 다른 모든 키와 달라야 합니다. 팁: 키 값을 증가시키기 전에 PrimaryKey.tag 템플릿 처리를 완료하십시오. 이 토큰은 필수입니다.	숫자 값
관계 매개변수		
*DatabaseName	웨어하우스 소스 또는 웨어하우스 목표의 비즈니스 이름 이 토큰은 필수입니다.	텍스트 문자열. 최대 40 바이트
*TableOwner	컬럼을 포함하는 테이블의 스키마, 소유자, 고급 규정자 또는 콜렉션 이 토큰은 필수입니다.	텍스트 문자열. 최대 128 바이트
*TablePhysicalName	데이터베이스 관리 프로그램이나 파일 시스템에 정의된, 컬럼을 포함하는 테이블 및 파일의 실제 이름 이 토큰은 필수입니다.	텍스트 문자열. 최대 80 바이트

값의 예

표 17은(는) 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 나타내는 각 토큰에 대한 값의 예를 보여줍니다.

표 17. PrimaryKey.tag 토큰 값 예

토큰	값의 예
*ColumnName	Country_code
*DatabaseName	재정 웨어하우스
*MapID	54627
*PrimaryKeyID	74622
*TableOwner	DB2ADMIN
*TablePhysicalName	GEOGRAPHY

PrimaryKeyAdditional.tag

이 템플릿을 사용하여 합성 기본 키를 정의합니다. PrimaryKeyAdditional.tag 템플릿을 사용하기 전에 PrimaryKey.tag 템플릿을 사용하여 첫번째 컬럼에서 제한조건을 정의해야 합니다. 그런 다음 이 템플릿을 사용하여 추가 컬럼을 추가할 수 있습니다. 또한 템플릿은 PrimaryKey.tag를 사용하여 정의된 첫번째 기본 키와 추가 기본 키를 관련시킵니다.

토큰

표18은(는) 템플리트의 각 토큰에 대한 정보를 제공합니다.

표 18. *PrimaryKeyAdditional.tag* 토큰

토큰	설명	허용되는 값
엔터티 매개변수		
<i>*ColumnName</i>	필드나 컬럼 이름 이름은 테이블이나 필드에서 고유해야 합니다. 이 토큰은 필수입니다.	텍스트 문자열. 최대 80 바이트
<i>*FirstPrimaryKeyID</i>	기본 키를 고유하게 식별하는 키. 키 숫자 값은 태그 언어 파일의 다른 모든 키와 달라야 합니다. 팁: 키 값을 증가시키기 전에 <i>PrimaryKeyAdditional.tag</i> 템플리트 처리를 완료하십시오. 이 토큰은 필수입니다.	
<i>*MapID</i>	교환 파일의 다른 모든 키와 다른 임의의 숫자 팁: 이 값을 증가시키기 전에 <i>PrimaryKeyAdditional.tag</i> 템플리트 처리를 완료하십시오. 이 토큰은 필수입니다.	
<i>*MapSeqNo</i>	기본 키 제한조건에 합성 키의 일부로 2부터 시작되는 증가하는 연속적인 고유한 숫자 추가된 각각의 추가 컬럼을 나타내는 숫자	
관계 매개변수		
<i>*DatabaseName</i>	웨어하우스 소스 또는 웨어하우스 목 표의 비즈니스 이름 이 토큰은 필수입니다.	텍스트 문자열. 최대 40 바이트
<i>*TableOwner</i>	컬럼을 포함하는 테이블의 스키마, 소 유자, 고급 규정자 또는 콜렉션 이 토큰은 필수입니다.	텍스트 문자열. 최대 15 바이트
<i>*TablePhysicalName</i>	데이터베이스 관리 프로그램이나 파일 시스템에 정의된, 컬럼을 포함하는 테이블 및 파일의 실제 이름 이 토큰은 필수입니다.	텍스트 문자열. 최대 80 바이트

값의 예

295 페이지의 표19은(는) 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 나타내는 각 토큰에 대한 값의 예를 보여줍니다.

표 19. PrimaryKeyAdditional.tag 토큰 값 예

토큰	값의 예
*ColumnName	Country_code
*DatabaseName	재정 웨어하우스
*MapID	99542
*MapSeqNo	2
*FirstPrimaryKeyID	07801
*TableOwner	DB2ADMIN
*TablePhysicalName	GEOGRAPHY

Data Warehouse Center 온라인 도움말

복제를 위한 테이블 또는 뷰 정의

테이블이나 뷰가 Data Warehouse Center에서 복제 소스로 사용되기 전에 DB2 제어 센터를 사용하여 복제를 위한 테이블이나 뷰를 정의해야 합니다.

AS/400 에이전트를 사용하여 Essbase VWP 수행

AS/400 에이전트를 통해 Essbase VWP를 수행하기 전에 ARBORLIB 및 ARBORPATH를 *sys 환경 변수로 설정해야 합니다. 이들을 설정하려면, 사용자 ID는 *jobctl 권한을 가지고 있어야 합니다. 이들 환경 변수는 Essbase가 설치된 라이브러리를 지정해야 합니다.

Data Warehouse Center 발행 창 및 관련 등록 정보 창 사용

태스크 도움말의 단계 10에는 한계 값을 1로 지정하고(트리에서 오브젝트의 레벨 제한) 프로세스를 발행한 경우, 이 프로세스로부터 1 단계만이 발행 및 표시된다는 점을 언급한 예가 있습니다. 이 예는 모든 경우에 올바르지 않습니다. 단계 8의 두 번째 점 표시 항목에서, 첫번째 사항이 잘못되었습니다. "정보 카탈로그 소스 컬럼과 목표 컬럼 사이에 변환 오브젝트를 생성하려면 컬럼 레벨을 클릭하십시오."가 맞습니다.

외부 키

온라인 도움말에서 "외부 키"에 대한 참조는 "웨어하우스 외부 키"이어야 합니다.

복제 노트북

온라인 도움말에서 "복제 정의 노트북"에 대한 참조는 "복제 단계 노트북"이어야 합니다.

태그 언어 가져오기

태그 언어 온라인 도움말 가져오기: 공통 가져오기 오류를 나타내는 점 표시 목록에서, 이 목록의 한 항목이 "제대로 내보내지 않은 태그 언어 파일을 가져오고 있습니다"입니다. 이 항목은 공통 입력 오류의 목록에 적용할 수 없습니다.

데이터 추가를 위한 링크

온라인 도움말의 『데이터 추가』 주제에서, "프로세스에 소스 테이블 추가" 및 "프로세스에 목표 테이블 추가" 주제에 대한 링크가 끊어져 있습니다. 도움말 색인에서 이 주제들을 찾을 수 있습니다.

테이블 가져오기

"소스 테이블 및 뷰를 웨어하우스 소스로 가져오기" 및 "웨어하우스 소스로 목표 테이블 가져오기" 도움말 항목에는 와일드카드 문자에 관한 잘못된 내용이 들어 있습니다. 문장은 다음과 같습니다.

예를 들어, XYZ*는 스키마가 있는 테이블 및 뷰를 리턴합니다.
테이블 및 뷰를 리턴합니다.

다음과 같이 바뀌어야 합니다.

예를 들어, XYZ%는 이들 문자로 시작되는 스키마가 있는
테이블 및 뷰를 리턴합니다.

RUNSTATS 및 REORGANIZE TABLE 온라인 도움말에 대한 정정

이 유틸리티에 대한 온라인 도움말은 통계를 수행하거나 재구성될 테이블을 소스 및 목표 모두로 링크해야 한다는 점을 언급합니다. 그러나, 이 단계에서는 소스에 기록하기 때문에 소스에서 단계로만 링크해야 합니다.

통지 페이지(웨어하우스 등록정보 노트북 및 스케줄 노트북)

웨어하우스 등록정보 노트북의 통지 페이지에서

보낸 사람 입력 필드는 문자열 <현재 사용자 로그인 ID>로 초기화됩니다.

이 문장은 다음과 같이 바뀌어야 합니다.

보낸 사람 입력 필드는 문자열 <현재 로그인 사용자 전자우편 주소>로 초기화됩니다.

스케줄 노트북의 통지 페이지에서 보낸 사람은 웨어하우스 등록정보 노트북에서 설정된 것으로 초기화됩니다. 어떤 것도 설정하지 않으면, 현재 로그인 사용자 전자우편 주소로 초기화됩니다. 로그인 사용자와 연관된 전자우편 주소가 없을 경우, 보낸 사람은 로그인 사용자 ID로 설정됩니다.

에이전트 사이트 노트북의 에이전트 모듈 필드

에이전트 사이트 노트북의 에이전트 모듈 필드는 웨어하우스 에이전트 디먼이 웨어하우스 에이전트를 발생시키면 프로그램의 이름을 제공합니다. IBM이 지시하지 않는 한 이 필드의 이름을 변경하지 마십시오.

DB2 OLAP Starter Kit

IBM DB2 OLAP Starter Kit 7.2는 몇몇 운영 체제 플랫폼에서 Oracle, MS-SQL, Sybase, Informix RDBMS(관계형 데이터베이스 관리 시스템)를 추가로 지원합니다. 버전 7.2에는 DB2를 포함하여 지원되는 모든 RDBMS용 스크립트와 도구가 포함됩니다. 몇 가지 제한사항이 있으며, 309 페이지의 『알려진 문제점 및 제한사항』에서 자세한 내용을 참조하십시오.

DB2 Universal Database용 DB2 OLAP Starter Kit 버전 7.2의 서비스 레벨은 Hyperion Essbase 6.1용 패치 2와 HIS(Hyperion Integration Server) 2.0용 패치 2를 더한 것과 같습니다.

OLAP 서버 웹 사이트

DB2 OLAP Starter Kit에 대한 최근 설치 및 사용법 추가 정보는 다음 DB2 OLAP 서버 웹 사이트의 라이브러리 페이지를 보십시오.

<http://www.ibm.com/software/data/db2/db2olap/library.html>

지원되는 운영 체제 서비스 레벨

버전 7.2용 OLAP Starter Kit의 서버 구성요소는 다음 운영 체제 및 서비스 레벨을 지원합니다.

- Windows NT 4.0 서버 SP 5 및 Windows 2000
- AIX 버전 4.3.3 이상
- Solaris 운영 체제 버전 2.6, 7, 8(Sun OS 5.6, 5.7, 5.8)

클라이언트 구성요소는 Windows 95, Windows 98, Windows NT 4.0 SP5, Windows 2000에서 수행됩니다.

UNIX에서 DB2 OLAP Starter Kit 설정 완료

DB2 OLAP Starter Kit 설치에 대한 UNIX에 대한 DB2 Universal Database 설치의 기본 프로시저를 따릅니다. 설치 프로그램은 다음 시스템 디렉토리에 제품 파일을 설치합니다(AIX: /usr/lpp/db2_07_01, Solaris: /opt/IBMDB2/V7.1).

그러면 인스턴스 작성 단계 중 두 개의 DB2 OLAP 디렉토리(essbase 및 is)가 sqllib 아래의 인스턴스 사용자의 홈 디렉토리에 작성됩니다. 한 번에 OLAP Server의 한 인스턴스만 머신에서 수행될 수 있습니다. 설정을 완료하려면, 사용자는 시스템에서 is/bin

디렉토리에 링크되지 않도록 수동으로 is/bin 디렉토리를 설정해야 합니다. 이것은 인스턴스의 홈 디렉토리 내에서 쓰기 가능한 디렉토리로 링크되어야 합니다.

Solaris에 대한 설정을 완료하려면, 인스턴스 ID를 사용하여 로그인하고, sqllib/is 디렉토리로 변경한 후 다음을 입력하십시오.

```
rm bin
mkdir bin
cd bin
ln -s /opt/IBMDB2/V7.1/is/bin/ismesg.mdb ismesg.mdb
ln -s /opt/IBMDB2/V7.1/is/bin/olapicmd olapicmd
ln -s /opt/IBMDB2/V7.1/is/bin/olapisvr olapisvr
ln -s /opt/IBMDB2/V7.1/is/bin/essbase.mdb essbase.mdb
ln -s /opt/IBMDB2/V7.1/is/bin/libolapams.so libolapams.so
```

OLAP Starter Kit용 ODBC 구성

IBM DB2 OLAP Starter Kit 7.2에서는 OLAP 통합 서버에서 관계형 데이터 소스와 OLAP 메타데이터 카탈로그의 ODBC(Open Database Connectivity) 연결 작업을 하려면 ODBC.ini 파일이 있어야 합니다.

- Windows 시스템에서는 이 파일이 레지스트리의 HKEY_LOCAL_MACHINE/SOFTWARE/ODBC 아래에 있습니다. 관계형 데이터 소스에 연결하는 방법에 대한 정보를 저장하려면 ODBC 데이터 소스 관리자를 사용하십시오.
- UNIX 시스템에서는 설치 프로그램이 모델 odbc.ini 파일을 작성합니다. 관계형 데이터 소스에 연결하는 방법에 대한 정보를 저장하려면 선호하는 편집기를 사용하여 이 파일을 편집하십시오.

ODBC.ini 파일은 ODBC 소프트웨어 패키지에 사용 가능하며, Microsoft Office 소프트웨어에도 포함되어 있습니다. ODBC 드라이버나 ODBC 관리자를 설치하는 응용 프로그램에 대한 자세한 정보는 <http://support.microsoft.com/support/kb/articles/Q1113/1/08.asp> 웹 사이트를 참조하십시오.

AIX 머신에서 Oracle을 사용하는 사용자가 ODBC를 Oracle용으로 구성하려면 MERANT 3.6 드라이버를 가리키도록 ODBC.ini 파일을 갱신해야 합니다.

버전 7.2에서는 OLAP Starter Kit가 관계형 데이터 소스 및 OLAP 메타데이터 카탈로그의 ODBC 연결을 관리합니다. 이러한 ODBC 연결을 위해 OLAP Starter Kit는 Windows NT 4.0, Windows 2000, AIX, Solaris에서 ODBC 드라이버를 사용합니다.

- DB2 Universal Database 버전 6 데이터베이스 클라이언트: Windows NT 4.0 SP5 또는 Windows 2000, AIX 4.3.3, Solaris 운영 체제 2.6, 7, 8(Sun OS 5.6, 5.7, 5.8)에서 DB2 버전 6 ODBC 드라이버 사용

- DB2 Universal Database 7.1 데이터베이스 클라이언트: Windows NT 4.0 SP5 또는 Windows 2000, AIX 4.3.3, Solaris 운영 체제 2.6, 7, 8(Sun OS 5.6, 5.7, 5.8)에서 DB2 버전 7 ODBC 드라이버 사용
- Oracle 8.04 및 8i SQL*Net 8.0 데이터베이스 클라이언트: Windows NT 4.0 SP5 또는 Windows 2000, AIX 4.3.3, Solaris 운영 체제 2.6, 7, 8(Sun OS 5.6, 5.7, 5.8)에서 MERANT 3.6 ODBC 드라이버 사용
- MS SQL Server 6.5.201(데이터베이스 클라이언트가 필요하지 않음): Windows NT 4.0 SP5 또는 Windows 2000에서 MS SQL Server 6.5 ODBC 드라이버 사용
- MS SQL Server 7.0(데이터베이스 클라이언트가 필요하지 않음): Windows NT 4.0 SP5 또는 Windows 2000에서 MS SQL Server 7.0 ODBC 드라이버 사용

UNIX 시스템에서 데이터 소스 구성

AIX와 Solaris에서는 ODBC용 환경 변수를 수동으로 설정하고 `odbc.ini` 파일을 편집하여 관계형 데이터 소스 및 OLAP 메타데이터 카탈로그를 구성해야 합니다. 새 드라이버 또는 데이터 소스를 추가하거나, 드라이버나 데이터 소스를 변경한 경우에는 반드시 `odbc.ini` 파일을 편집해야 합니다.

ODBC 환경 변수 구성

UNIX 시스템에서는 ODBC 핵심 구성요소를 액세스할 수 있도록 환경 변수를 설정해야 합니다. 필요한 변수를 설정하는 `is.sh` 및 `is.csh` 셸 스크립트는 Starter Kit의 홈 디렉토리에 있습니다. ODBC를 통해 데이터 소스에 연결하기 전에 이 스크립트 중 하나를 실행해야 합니다. 또한 사용할 사용자 이름으로 OLAP Starter Kit을 수행하려면 로그인 스크립트에 이 스크립트들을 포함해야 합니다.

odbc.ini 파일 편집

`odbc.ini` 파일에서 데이터 소스를 구성하려면 ODBC 데이터 소스에 대한 설명과 이름을 추가하고, 데이터 소스 이름에 사용하기 위해 작성한 별도의 섹션에 ODBC 드라이버 경로, 파일 이름, 기타 드라이버 설정을 입력해야 합니다. 설치 프로그램은 샘플 `odbc.ini` 파일을 `ISHOME` 디렉토리에 설치합니다. 이 파일에는 지원되는 ODBC 드라이버에 대한 일반 ODBC 연결과 구성 정보가 있습니다. 이 파일을 시작점으로 하여 사용할 ODBC 드라이버를 관계형 데이터 소스 및 OLAP 메타데이터 카탈로그에 맵핑하십시오.

`odbc.ini`가 아닌 다른 파일을 사용할 경우에는 `ODBCINI` 환경 변수를 사용할 파일 이름으로 설정하십시오.

odbc.ini 파일에 데이터 소스 추가

1. OLAP Starter Kit 서버가 실행되는 시스템에서 `vi` 등의 편집기를 사용하여 `odbc.ini` 파일을 여십시오.

2. [ODBC Data Sources]로 시작되는 섹션을 찾은 다음 mydata=data source for analysis와 같이 데이터 소스 이름과 설명이 있는 새 줄을 추가하십시오. 혼동하지 않도록 하려면 데이터 소스의 이름을 RDBMS의 데이터베이스 이름과 동일하게 지정하십시오.
3. [mydata]와 같이 꺾쇠 괄호로 묶인 새로운 데이터 소스 이름이 있는 새 줄을 작성하여 파일에 새 섹션을 추가하십시오.
4. 데이터 소스 이름 다음의 줄에는 이 데이터 소스에 필요한 ODBC 드라이버의 전체 경로와 파일 이름 및 기타 필요한 ODBC 드라이버 정보를 추가하십시오. 다음 절에 나오는 예를 지침으로 삼아 RDBMS의 데이터 소스에 매핑하십시오. Driver= 설정에 지정한 위치에 실제로 ODBC 드라이버 파일이 있는지 확인하십시오.
5. odbc.ini 편집을 마치면 파일을 저장하고 텍스트 편집기를 종료하십시오.

DB2용 ODBC 설정 예

다음 예는 AIX에 설치된 DB2 Universal Database 버전 6.1에서 IBM DB2 원시 ODBC 드라이버를 사용하여 db2data 관계형 데이터 소스에 연결하기 위해 odbc.ini 편집 방법을 보여줍니다. vi 편집기에서 \$ODBCINI 명령을 사용하여 odbc.ini 편집 후 다음 명령문을 삽입하십시오.

```
[ODBC Data Sources]
db2data=DB2 Source Data on AIX
...
[db2data]
Driver=/home/db2inst1/sql1lib/lib/db2.o
Description=DB2 Data Source - AIX, native
```

Oracle용 ODBC 설정 예

다음 예는 Oracle 버전 8(Solaris)에서 MERANT 버전 3.6 ODBC 드라이버를 사용하여 oradata 관계형 데이터 소스에 연결하기 위해 odbc.ini 편집 방법을 보여줍니다. 이 예에서 LogonID와 Password에는 OLAP Starter Kit의 사용자 이름과 암호에 사용된 실제 값을 사용했습니다.

```
[ODBC Data Sources]
oradata=Oracle8 Source Data on Solaris
...
[myoracle] Driver=
/export/home/users/dkendric/is200/odbc/lib/ARor815.so
Description=my oracle source
```

UNIX 시스템에서 OLAP 메타데이터 카탈로그 구성

AIX와 Solaris에서 OLAP 메타데이터 카탈로그를 구성하는 것은 데이터 소스를 구성하는 것과 비슷합니다. OLAP 메타데이터 카탈로그 데이터베이스에 대해 301 페이지의 『odbc.ini 파일 편집』에 설명된 대로 odbc.ini 파일에 데이터 소스 이름과 섹션을 추가하십시오. 이 외 다른 부분은 변경할 필요가 없습니다.

ODBC 데이터 소스로 구성하기 전에 먼저 지원되는 RDBMS에 OLAP 메타데이터 카탈로그 데이터베이스를 작성해야 합니다.

다음 예는 DB2 버전 6.1(Solaris)에서 원시 ODBC 드라이버를 사용하여 OLAP 메타데이터 카탈로그, TBC_MD에 연결하기 위해 odbc.ini를 편집하는 방법을 보여줍니다.

```
[ODBC Data Sources]
ocd6a5a=db2 v6
...
[ocd6a5a]
Driver=/home/db2inst1/sqllib/lib/db2.0
Description=db2
```

Windows 시스템에서 데이터 소스 구성

Windows NT나 Windows 2000 시스템에서 관계형 데이터 소스를 구성하려면 ODBC 관리자를 시작한 다음 OLAP 모델 및 metaoutline 작성에 사용할 데이터 소스에 대한 연결을 작성해야 합니다. Windows 제어판에서 ODBC 관리자 유틸리티를 실행하십시오. 다음 예는 DB2 데이터 소스를 작성합니다. 다른 RDBMS에서는 대화 상자가 다릅니다.

ODBC 관리자를 사용하여 관계형 데이터 소스를 구성하려면 다음 단계를 완료하십시오.

1. Windows 바탕 화면에서 제어판 창을 여십시오.
2. 제어판 창에서 다음 단계 중 하나를 수행하십시오.
 - a. Windows NT에서는 **ODBC** 아이콘을 더블 클릭하여 ODBC 데이터 소스 관리자 대화 상자를 여십시오.
 - b. Windows 2000에서는 관리 도구 아이콘을 더블 클릭한 다음 데이터 소스 (**ODBC**) 아이콘을 두 번 클릭하여 ODBC 데이터 소스 관리자 대화 상자를 여십시오.
3. ODBC 데이터 소스 관리자 대화 상자에서 시스템 **DSN** 탭을 누르십시오.
4. 추가를 눌러 새 데이터 소스 작성 대화 상자를 여십시오.
5. ODBC 관리자의 새 데이터 소스 작성 대화 상자에 있는 드라이버 목록에서 IBM DB2 ODBC 드라이버와 같은 적절한 드라이버를 선택하고 완료를 눌러 ODBC IBMDB2 드라이버 - 추가 대화 상자를 여십시오.
6. ODBC IBM DB2 드라이버 - 추가 대화 상자의 데이터베이스 별명 드롭 다운 목록에서 관계형 데이터 소스(예: 샘플 응용프로그램의 TBC)에 대한 데이터베이스 이름을 선택하십시오.
7. 설명 텍스트 상자에 드라이버 사용 방법을 나타내는 선택적 설명을 입력하고 추가를 누르십시오. 예를 들어, 다음 단어를 입력하여 My Business 데이터베이스를 설명할 수 있습니다.

Customers, products, markets

다음 단어를 입력하여 샘플 응용프로그램의 데이터베이스를 설명할 수 있습니다.

Sample relational data source

OLAP Starter Kit Desktop에서 연결할 때 이러한 설명을 보고 선택한 항목에 사용 가능한 데이터 소스를 식별할 수 있습니다.

8. 확인을 눌러 ODBC 데이터 소스 관리자 대화 상자로 돌아가십시오. 입력한 데이터 소스 이름과 이 데이터 소스에 맵핑한 드라이버가 시스템 DSN 탭의 시스템 데이터 소스 목록에 표시됩니다.

데이터 소스의 구성 정보를 편집하려면

1. 데이터 소스 이름을 선택하고 구성을 눌러 ODBC IBM DB2 - 추가 대화 상자를 여십시오.
2. 변경할 정보가 있으면 알맞게 정정하십시오.
3. 확인을 두 번 눌러 대화 상자를 닫으십시오.

Windows 시스템에서 OLAP 메타데이터 카탈로그 구성

Windows NT나 Windows 2000에서 OLAP 메타데이터 카탈로그를 구성하려면 ODBC 관리자를 시작한 다음 OLAP 메타데이터 카탈로그 데이터베이스가 포함된 데이터 소스에 대한 연결을 작성하십시오.

다음 예는 DB2 데이터 소스를 작성합니다. 다른 RDBMS에서는 대화 상자가 다릅니다. OLAP 메타데이터 카탈로그용 데이터 소스를 작성하려면 다음 단계를 완료하십시오.

1. 바탕 화면에서 제어판 창을 여십시오.
2. 제어판 창에서 다음 단계 중 하나를 수행하십시오.
 - a. Windows NT에서는 ODBC 아이콘을 더블 클릭하여 ODBC 데이터 소스 관리자 대화 상자를 여십시오.
 - b. Windows 2000에서는 관리 도구 아이콘을 더블 클릭한 다음 데이터 소스 (ODBC) 아이콘을 두 번 클릭하여 ODBC 데이터 소스 관리자 대화 상자를 여십시오.
3. ODBC 데이터 소스 관리자 대화 상자에서 시스템 DSN 탭을 누르십시오.
4. 추가를 눌러 새 데이터 소스 작성 대화 상자를 여십시오.
5. ODBC 관리자의 새 데이터 소스 작성 대화 상자에 있는 드라이버 목록에서 IBM DB2 ODBC 드라이버와 같은 적절한 드라이버를 선택하고 완료를 눌러 ODBC IBMDB2 드라이버 - 추가 대화 상자를 여십시오.

6. ODBC IBM DB2 드라이버 - 추가 대화 상자의 데이터베이스 별명 드롭 다운 목록에서 OLAP 메타데이터 카탈로그(예: 샘플 응용프로그램의 TBC_MD)에 대한 데이터베이스 이름을 선택하십시오. 선택한 데이터베이스의 이름이 데이터 소스 이름 텍스트 상자에 자동으로 표시됩니다.

7. 데이터 소스의 이름을 변경하려면 데이터 소스 이름 텍스트 상자에 표시된 이름을 선택한 다음 드라이버 사용 방법을 나타내는 새 이름을 입력하고 추가를 누르십시오. 예를 들어, 다음 이름을 입력하여 첫번째 OLAP 메타데이터 카탈로그에 연결할 때 이 드라이버를 사용한다는 것을 나타낼 수 있습니다.

OLAP Catalog first

다음 이름을 입력하여 샘플 응용프로그램의 OLAP 메타데이터 카탈로그 데이터베이스에 연결한다는 것을 나타낼 수 있습니다.

TBC_MD

8. 설명 텍스트 상자에 드라이버 사용 방법을 나타내는 설명을 입력하십시오. 예를 들어, 다음 단어를 입력하여 OLAP 메타데이터 카탈로그를 설명할 수 있습니다.

My first models and metaoutlines

다음 단어를 입력하여 샘플 응용프로그램의 OLAP 메타데이터 카탈로그 데이터베이스를 설명할 수 있습니다.

Sample models and metaoutlines

이 설명을 보고 OLAP Starter Kit Desktop에서 OLAP 메타데이터 카탈로그에 연결할 때 선택할 카탈로그를 쉽게 식별할 수 있습니다.

9. 확인을 눌러 ODBC 데이터 소스 관리자 대화 상자로 돌아가십시오. 입력한 데이터 소스 이름과 이 데이터 소스에 맵핑한 드라이버가 시스템 DSN 탭의 시스템 데이터 소스 목록에 표시됩니다.

데이터 소스의 구성 정보를 편집하려면

1. 데이터 소스 이름을 선택하고 구성을 눌러 ODBC IBM DB2 - 추가 대화 상자를 여십시오.

2. 변경할 정보가 있으면 알맞게 수정하십시오.

3. 확인을 두 번 눌러 대화 상자를 닫으십시오.

데이터 소스를 구성한 후의 작업

관계형 데이터 소스와 OLAP 메타데이터 카탈로그를 구성했으면 OLAP Starter Kit에서 관계형 데이터 소스와 OLAP 메타데이터 카탈로그에 연결할 수 있습니다. 연결한 다음 OLAP 모델과 metaoutline을 작성, 수정, 저장할 수 있습니다.

SQL Server 데이터베이스를 호출하는 동안 SQL Server ODBC 드라이버가 시간종료될 수 있습니다. 데이터베이스가 사용 중이지 않으면 다시 시도하십시오. 드라이버 시간종료 시간을 늘려 이 문제를 방지할 수 있습니다. 자세한 정보는 사용 중인 드라이버의 ODBC 문서를 참조하십시오.

ODBC 연결 문제점 및 솔루션에 대한 자세한 정보는 *OLAP Integration Server* 시스템 관리자 안내서를 참조하십시오.

OLAP Starter Kit Desktop에서 로그인

OLAP Starter Kit Desktop을 사용하여 OLAP 모델과 metaoutline을 작성하려면, 클라이언트 소프트웨어를 DB2 OLAP Integration Server와 DB2 OLAP Server에 연결해야 합니다. 로그인 대화 상자는 데스크탑이 이들 두 서버에 연결하는 데 필요한 정보를 묻는 프롬프트를 표시합니다. 대화 상자의 왼쪽에 DB2 OLAP Integration Server에 대한 정보를 입력하십시오. 오른쪽에는 DB2 OLAP Server에 대한 정보를 입력하십시오.

DB2 OLAP Integration Server에 연결하려면, 다음과 같이 수행하십시오.

- 서버: 통합 서버의 호스트 이름이나 IP 주소를 입력하십시오. 데스크탑과 동일한 워크스테이션에 통합 서버를 설치한 경우, 일반 값은 "localhost" 또는 "127.0.0.1"입니다.
- OLAP 메타데이터 카탈로그: OLAP Integration Server에 연결할 때에는 메타데이터 카탈로그도 지정해야 합니다. OLAP Integration Server는 작성한 OLAP 모델 및 metaoutline에 대한 정보를 메타데이터 카탈로그로 알려진 관계형 데이터베이스에 저장합니다. 이 관계형 데이터베이스는 ODBC에 대해 등록해야 합니다. 카탈로그 데이터베이스에는 OLAP Integration Server가 인식하는 특정 관계형 테이블 세트가 포함됩니다. 로그인 대화 상자에서, 통합 서버를 지정한 다음 OLAP 메타데이터 카탈로그 필드에 대한 풀다운 메뉴를 펼쳐서 OLAP Integration Server에 알려진 ODBC 데이터 소스의 목록을 볼 수 있습니다. 메타데이터 카탈로그 테이블이 포함된 ODBC 데이터베이스를 선택하십시오.
- 사용자 이름 및 암호: OLAP Integration Server는 이 패널에서 지정한 사용자 이름과 암호를 사용하여 메타데이터 카탈로그에 연결합니다. 이것은 서버에 있는 로그인 계정입니다(서버와 클라이언트가 동일한 머신에서 수행되지 않으면 클라이언트에는 이 계정이 없습니다). 사용자 이름은 OLAP 메타데이터 카탈로그를 작성한 사용자의 이름이어야 합니다. 그렇지 않은 경우, OLAP Integration Server는 카탈로그 데이터베이스에서 관계 테이블을 찾지 않습니다. 테이블 스키마 이름들이 다르기 때문입니다.

DB2 OLAP Server 정보는 선택적이므로 로그인 대화 상자의 오른쪽에 있는 입력 필드는 공백으로 둘 수 있습니다. 그러나, 데스크탑 및 Administration Manager의 일부 조작에서는 DB2 OLAP Server에 연결할 것을 요구합니다. 이들 필드를 공백으로 두

면, 데스크탑은 요청한 조작을 완료하기 위해 통합 서버가 DB2 OLAP Server에 연결해야 하는 경우에 로그인 대화 상자를 다시 표시합니다. 로그인 대화 상자에서 DB2 OLAP Server 필드를 항상 채우는 것이 좋습니다.

DB2 OLAP Server에 연결하려면, 다음과 같이 수행하십시오.

- 서버: DB2 OLAP Server의 호스트 이름이나 IP 주소를 입력하십시오. OLAP Starter Kit을 수행하고 있는 경우, OLAP Server와 통합 서버는 동일합니다. 통합 서버와 OLAP Server가 다른 호스트에 설치된 경우, OLAP Integration Server에 정의된 호스트 이름이나 IP 주소를 입력하십시오.
- 사용자 이름 및 암호: OLAP Integration Server는 이 패널에서 지정한 사용자 이름과 암호를 사용하여 DB2 OLAP Server에 연결합니다. 이 사용자 이름과 암호는 DB2 OLAP Server에 이미 정의되어 있어야 합니다. OLAP Server는 호스트 운영 체제에서 자체의 사용자 이름과 암호를 별도로 관리합니다.

Starter Kit 로그인 예

다음의 예는 OLAP 샘플을 작성하고 관리자 사용자 ID로 *db2admin*을 선택하고 OLAP Starter Kit 설치 중에 관리자 암호로 *password*를 선택한 것으로 가정합니다.

- OLAP Integration Server의 경우: 서버는 *localhost*, OLAP 메타데이터 카탈로그는 *TBC_MD*, 사용자 이름은 *db2admin*, 암호는 *password*입니다.
- DB2 OLAP Server의 경우: 서버는 *localhost*, 사용자 이름은 *db2admin*입니다.

OLAP Starter Kit에 대한 샘플 데이터베이스 수동 작성 및 구성

샘플 데이터베이스는 OLAP Starter Kit를 설치하면 자동으로 작성됩니다. 다음의 지시 사항은 필요한 경우에 카탈로그 및 샘플 데이터베이스를 수동으로 설정하는 방법을 설명합니다.

1. Windows에서는 시작 → 프로그램 → Windows NT용 DB2 → 명령 창을 클릭하여 명령 센터를 여십시오.
2. 카탈로그 데이터베이스를 작성하십시오.
 - a. `db2 create db OLAP_CAT`를 입력하십시오.
 - b. `db2 connect to OLAP_CAT`를 입력하십시오.
3. 데이터베이스에서 테이블을 작성하십시오.
 - a. `\SQLLIB\IS\ocscript\ocdb2.sql`을 찾으십시오.
 - b. `db2 -tf ocdb2.sql`을 입력하십시오.
4. 샘플 소스 데이터베이스를 작성하십시오.
 - a. `db2 connect reset`를 입력하십시오.
 - b. `db2 create db TBC`를 입력하십시오.
 - c. `db2 connect to TBC`를 입력하십시오.

5. 데이터베이스에서 테이블을 작성하십시오.
 - a. \SQLLIB\IS\samples\를 찾으십시오.
 - b. tbcdb2.sql을 \SQLLIB\samples\db2sampl\tbc에 복사하십시오.
 - c. lddb2.sql을 \SQLLIB\samples\db2sampl\tbc에 복사하십시오.
 - d. \SQLLIB\samples\db2sampl\tbc를 찾으십시오.
 - e. db2 -tf tbcdb2.sql을 입력하십시오.
 - f. db2 -vf lddb2.sql을 입력하여 테이블에 샘플 소스 데이터를 로드하십시오.
6. 샘플 카탈로그 데이터베이스를 작성하십시오.
 - a. db2 connect reset를 입력하십시오.
 - b. db2 create db TBC_MD를 입력하십시오.
 - c. db2 connect to TBC_MD를 입력하십시오.
7. 데이터베이스에서 테이블을 작성하십시오.
 - a. \SQLLIB\IS\samples\tbc_md를 찾으십시오.
 - b. ocdb2.sql을 \SQLLIB\samples\db2sampl\tbcmd에 복사하십시오.
 - c. lcdb2.sql을 \SQLLIB\samples\db2sampl\tbcmd에 복사하십시오.
 - d. \SQLLIB\samples\db2sampl\tbcmd를 찾으십시오.
 - e. db2 -tf ocdb2.sql을 입력하십시오.
 - f. db2 -vf lcdb2.sql을 입력하여 테이블에 샘플 메타데이터를 로드하십시오.
8. TBC_MD, TBC, AND OLAP_CAT용 ODBC를 구성하십시오.
 - a. 시작 → 설정 → 제어판을 눌러 NT 제어판을 여십시오.
 - b. 목록에서 ODBC(또는 ODBC 데이터 소스)를 선택하십시오.
 - c. 시스템 DSM 탭을 선택하십시오.
 - d. 추가를 클릭하십시오. 새 데이터 소스 작성 창이 열립니다.
 - e. 목록에서 IBM DB2 ODBC 드라이버를 선택하십시오.
 - f. 완료를 클릭하십시오. ODBC IBM DB2 드라이버 - 추가 창이 열립니다.
 - g. 데이터 소스 이름 필드에 데이터 소스의 이름(OLAP_CAT)을 입력하십시오.
 - h. 데이터베이스 별명 필드에 별명 이름을 입력하거나 아래 화살표를 클릭하고 목록에서 OLAP_CAT를 선택하십시오.
 - i. 확인을 클릭하십시오.
 - j. TBC_MD 및 TBC 데이터베이스에 대해 이들 단계를 반복하십시오.

응용프로그램을 OLAP Starter Kit 버전 7.2로 이주

설치 프로그램이 OLAP Starter Kit 샘플 응용프로그램, 데이터베이스, 데이터 파일을 다시 설치하지 않습니다. 기존의 응용프로그램 및 데이터베이스는 전혀 영향을 받지 않습니다. 그러나 설치하기 전에 항상 응용프로그램과 데이터베이스를 백업하는 것이 좋습니다.

응용프로그램을 열면 자동으로 버전 7.2로 이주됩니다.

알려진 문제점 및 제한사항

이 절에서는 DB2 OLAP Starter Kit에 대해 알려진 한계를 나열합니다.

Windows 플랫폼용 Merant 드라이버와의 Informix RDBMS 호환성

Windows 플랫폼용 Merant 드라이버가 Informix RDBMS와 작동하려면 다음 두 항목을 PATH 문에 추가해야 합니다.

- C:\Informix
- C:\Informix\bin

두 항목 모두 PATH 시작 부분에 있어야 합니다.

OLAP 모델과 연관된 metaoutline 간에 발생할 수 있는 차원 불일치

몇몇 경우에는 OLAP 모델에 일치하는 차원이 없는 차원을 metaoutline에 작성할 수 있습니다. 다음과 같은 시나리오에서 가능합니다.

1. 새 OLAP 모델을 작성하고 저장하십시오.
2. 이 모델에 바탕을 두고 metaoutline을 작성한 후 metaoutline을 저장하지 마십시오.
3. OLAP 모델로 돌아가서 metaoutline 차원 중 하나를 작성할 때 기준으로 사용한 차원을 삭제하십시오.
4. metaoutline으로 돌아가서 저장하고 닫은 다음 다시 여십시오. 그러면 이 metaoutline에 OLAP 모델에 일치하는 차원이 없는 차원이 포함됩니다.

OLAP Starter Kit는 이와 같이 작성된 일치하지 않는 차원과 metaoutline에 있는 사용자 정의 차원을 구별하지 못합니다. 따라서 metaoutline에 일치하지 않는 차원이 표시되더라도 OLAP 모델에 일치하는 차원이 없기 때문에 metaoutline은 이를 사용자 정의 차원으로 간주합니다.

Windows 2000 플랫폼에서 구성원 및 데이터를 로드할 수 없게 하는 TMP 환경 변수 설정

Windows 2000과 Windows NT에서는 TMP에 대한 기본 시스템 환경 변수 설정과 사용자 환경 변수 설정이 다르기 때문에, Windows 2000에서 OLAP Starter Kit을 실행하면 구성원 및 데이터를 로드할 수 없습니다. 이 경우 temp

파일을 작성할 수 없다는 오류 메시지가 나타납니다. 다음 단계를 따라서 Windows 2000에서 이러한 제한사항으로 인한 문제를 일시적으로 해결할 수 있습니다.

1. C:\TEMP라는 디렉토리를 작성하십시오.
2. 시스템과 사용자의 TMP 환경 변수를 모두 TMP=C:\TEMP로 설정하십시오.

ODBC를 설치했는데 기존 Merant 드라이버가 바뀌지 않을 경우

이렇게 설치했을 때는 기존 3.6 Merant ODBC 드라이버가 갱신되지 않습니다. OLAP Starter Kit 버전 7.1, FixPak 2 이전 버전에서 업그레이드할 경우에는 이전에 설치한 ODBC 드라이버를 사용해야 합니다.

UNIX 플랫폼에서 Merant Informix ODBC 드라이버 사용

UNIX 플랫폼에서 Merant Informix ODBC 드라이버를 사용하려면 다음 중 하나를 수행하십시오.

- Starter Kit을 시작하기 전에 LANG 환경 변수를 "en_US"로 설정하십시오. 예를 들어, 콘 셸에 다음과 같이 입력하십시오.

```
export LANG='en_US'
```

OLAP Starter Kit을 시작할 때마다 이 변수를 설정하십시오.

- LANG 환경 변수가 다른 값으로 설정되어 있지 않은 경우에는 설치한 후 다음 기호 링크를 만드십시오.

```
ln -s $ISHOME/locale/en_US $ISHOME/locale/$LANG
```

OLAP 클라이언트 및 서버의 서비스 레벨 혼용

IBM은 DB2 OLAP Starter Kit의 클라이언트 및 서버 구성요소를 동일한 버전 및 FixPak 레벨로 유지할 것을 권장합니다. 그러나 몇몇 경우에는 클라이언트 및 서버 구성요소를 서로 다른 수준으로 섞어서 사용할 수 있습니다.

같은 버전의 클라이언트와 서버를 다른 서비스 레벨에서 사용

IBM은 이전 버전의 서버에 새 버전의 클라이언트를 사용하는 것을 지원하지 않거나 권장하지 않습니다. 그러나 IBM이 지원하지는 않지만 새 버전의 서버에 이전 버전의 클라이언트를 사용할 수는 있습니다. 이 경우 몇 가지 문제점이 발생할 수 있습니다. 예를 들어, 다음과 같습니다.

- 서버의 메시지가 정확하지 않습니다. 서버의 레벨과 일치하도록 클라이언트의 message.MDB 파일을 업그레이드하여 이 문제를 일시적으로 해결할 수 있습니다.
- 서버의 새로운 기능이 작동하지 않습니다. 새로운 기능을 사용하려고 하면 클라이언트나 서버 또는 둘 다 실패합니다.
- 클라이언트와 서버가 제대로 연결되지 않습니다.

한 버전에서 여러 서버에 단일 클라이언트 사용

클라이언트 하나를 다른 머신이나 운영 체제에 있는 여러 개의 OLAP 서버에 연결해야 할 경우, IBM은 모두 동일한 버전과 서비스 레벨을 사용할 것을 권장합니다. 클라이언트 레벨은 적어도 가장 낮은 서버 레벨과 동일해야 합니다. 문제가 발생하면 다른 클라이언트 머신을 사용하여 해당 호스트와 일치하도록 하거나, 모든 클라이언트와 서버를 동일한 레벨로 업그레이드해야 합니다.

여러 버전의 클라이언트와 서버 혼용

IBM은 OLAP Starter Kit 버전 7.1의 클라이언트 및 서버를 버전 7.2의 클라이언트 및 서버와 사용하는 것을 지원하지 않습니다. IBM OLAP 제품을 새로운 버전 레벨로 업그레이드하면, 대개 클라이언트와 서버를 동일한 버전 레벨로 설정하기 위해 네트워크가 갱신되고 데이터 형식이 변경됩니다.

IBM 제품(DB2 OLAP Starter Kit)을 Hyperion 제품(Hyperion Essbase 및 Hyperion Integration Server)과 혼용

IBM은 IBM의 OLAP 클라이언트 및 서버를 Hyperion Solutions의 클라이언트 및 서버와 혼용하는 것을 지원하지 않습니다. 이 둘 사이에는 기능 상의 몇 가지 차이가 있기 때문에, 이들 혼용 요소들이 같은 상황에서 작동하더라도 문제가 발생할 수 있습니다.

OLAP Spreadsheet Add-in EQD 파일 누락

DB2 OLAP Starter Kit에서, Spreadsheet Add-in에는 Query Designer(EQD)라고 하는 구성요소가 있습니다. EQD에 대한 온라인 도움말 중 지습서라는 버튼을 누르면 아무런 내용도 표시되지 않습니다. EQD 지습서에 표시되어야 하는 내용은 *Excel용 OLAP Spreadsheet Add-in 사용자 안내서*와 *1-2-3용 OLAP Spreadsheet Add-in 사용자 안내서*의 2장 부속 집합입니다. EQD 지습서의 모든 정보는 정보 센터에서 이 서적들의 HTML 버전 및 PDF 버전으로 사용할 수 있습니다.

Information Catalog Manager 관리 안내서

Information Catalog Manager 초기화 유틸리티

Information Catalog Manager(ICM) 유틸리티로, 이제 다음 명령을 사용하여 SQL문을 CREATE TABLE 문의 끝에 추가할 수 있습니다.

```
CREATEIC \DBTYPE dbtype \DGNAME dgname \USERID userid \PASSWORD password
\KA1 userid \TABOPT "directory:tabopt.file"
```

DB2가 설치된 디렉토리에서 CREATEIC 유틸리티의 TABOPT 키워드를 지정할 수 있습니다. TABOPT 키워드 뒤의 값은 전체 경로가 있는 *tabopt.file* 파일 이름입니다. 디렉토리 이름에 공백이 포함되어 있으면, 이름을 인용부호로 묶으십시오. *tabopt.file* 파일의 내용에는 CREATE TABLE 문에 추가할 정보가 들어 있어야 합니다. 아래의 SQL 문 중 하나를 사용하여 이 *tabopt.file* 파일에 쓸 수 있습니다 ICM 유틸리티는 이 파일을 읽은 뒤 그것을 CREATE TABLE 문에 추가합니다.

표 20. SQL문

IN MYTABLESPACE	MYTABLESPACE의 데이터로 테이블을 작성합니다.
DATA CAPTURE CHANGES	테이블을 작성하고 SQL 변경사항을 확장된 형식으로 로그합니다.
IN ACCOUNTING INDEX IN ACCOUNT_IDX	ACCOUNTING의 데이터와 ACCOUNT_IDX의 색인으로 테이블을 작성합니다.

내용 파일의 최대 크기는 1000개의 1 바이트 문자입니다.

이 새로운 기능은 Windows와 UNIX 시스템에서만 사용 가능합니다.

사용권 문제

다음 메시지가 표시될 경우,

```
FLG0083E: IBM Information Catalog Manager 초기화 유틸리티에 대해 유효한
사용권을 갖고 있지 않습니다. 해당 지역의 소프트웨어 사업부나 IBM
영업 대표에게 문의하십시오.
```

DB2 Warehouse Manager나 IBM DB2 OLAP Server를 구입하고 Information Catalog Manager 구성요소를 설치해야 합니다. 이 구성요소에 Information Catalog Manager 초기화 유틸리티가 있습니다.

설치 문제

DB2 Warehouse Manager나 IBM DB2 OLAP Server를 설치한 후 다른 Information Catalog Manager 관리자 구성요소를 같은 워크스테이션에 설치한 경우(DB2 Universal Database CD-ROM을 사용하여), Information Catalog 초기화 유틸리티 위에 겹쳐쓰

기 하였을 수도 있습니다. 그러한 경우, \sqllib\bin 디렉토리에서, createic.bak 및 flgnmwcr.bak 파일을 찾고 이름을 각각 createic.exe 및 flgnmwcr.exe로 변경하십시오.

DB2 Universal Database에서 추가 Information Catalog Manager 구성요소를 설치할 경우, 그 구성요소들은 Data Warehouse Manager를 설치한 곳과 다른 별도의 워크스테이션에 있어야 합니다. 자세한 내용은 DB2 Warehouse Manager 설치 안내서에서, Information Catalog Manager 구성요소 설치를 참조하십시오.

DB2 버전 7 Information Catalog Manager를 사용하여 DB2 버전 5 정보 카탈로그에 액세스

DB2 버전 7 Information Catalog Manager 하위 구성요소는 DB2 버전 7 설치 프로세스에 의해 구성된대로 DB2 버전 6과 DB2 버전 7 데이터베이스에 저장된 정보 카탈로그에 대한 액세스를 지원합니다. DB2 버전 5 데이터베이스에 저장된 정보 카탈로그에 액세스하기 위해 하위 구성요소의 구성을 수정할 수 있습니다. DB2 버전 7 Information Catalog Manager 하위 구성요소는 DB2 버전 2 또는 다른 이전 버전에서의 데이터에 대한 액세스를 지원하지 않습니다.

DB2 버전 5 데이터베이스에 저장된 정보 카탈로그에 액세스하도록 정보 카탈로그 관리자, 정보 카탈로그 사용자 및 정보 카탈로그 초기화 유틸리티를 설정하려면, 다음과 같이 수행하십시오.

1. DB2 버전 7 Information Catalog Manager가 설치되지 않은 워크스테이션에 DB2 Connect Enterprise Edition 버전 6을 설치하십시오.

DB2 Connect Enterprise Edition은 DB2 Universal Database Enterprise Edition 및 DB2 Universal Database Enterprise - Extended Edition의 일부로 포함됩니다. 이들 DB2 제품 중 하나의 버전 6이 설치된 경우, DB2 Connect를 별도로 설치할 필요가 없습니다.

제한사항: 동일한 Windows NT 또는 OS/2 워크스테이션에 여러 DB2 버전을 설치할 수 없습니다. 다른 Windows NT 워크스테이션이나 OS/2 또는 UNIX 워크스테이션에 DB2 Connect를 설치할 수 있습니다.

2. DB2 버전 5 데이터에 액세스할 수 있도록 Information Catalog Manager와 DB2 Connect 버전 6을 구성하십시오. *DB2 Connect* 사용자 안내서에서 자세한 내용을 참조하십시오. 다음의 단계는 필수적인 단계의 개요입니다.
 - a. DB2 버전 5 시스템에서 DB2 명령행 처리기를 사용하여 Information Catalog Manager가 액세스할 버전 5 데이터베이스를 카탈로그화하십시오.
 - b. DB2 Connect 시스템에서 DB2 명령행 처리기를 사용하여 다음을 카탈로그화하십시오.
 - DB2 버전 5 시스템에 대한 TCP/IP 노트

- DB2 버전 5 시스템에 대한 데이터베이스
 - DB2 버전 5 시스템에 대한 DCS 항목
- c. Information Catalog Manager에서 DB2 명령행 처리기를 사용하여 다음을 카탈로그화하십시오.
- DB2 Connect 시스템에 대한 TCP/IP 노드
 - DB2 Connect 시스템에 대한 데이터베이스

DB2 Universal Database 설치 및 구성 보충 설명서에서 데이터베이스 카탈로그화에 대한 내용을 참조하십시오.

3. Information Catalog Manager가 있는 웨어하우스에서 DB2 Connect를 통해 액세스할 각 데이터베이스에 DB2 CLI 패키지를 바인드하십시오.

다음의 DB2 명령은 가정한 DB2 버전 5 데이터베이스인 v5database로의 바인딩 예를 제공합니다. DB2 명령행 처리기를 사용하여 다음의 명령을 실행하십시오. db2cli.lst와 db2ajgrt는 \sqllib\bnd 디렉토리에 위치합니다.

```
db2 connect to v5database user userid using password
db2 bind db2ajgrt.bnd
db2 bind @db2cli.lst blocking all grant public
```

여기서 *userid*는 v5database의 사용자 ID이며 *password*는 사용자 ID의 암호입니다.

db2cli.list가 DB2 버전 5 데이터베이스에 바인드되면 오류가 발생합니다. 이 오류는 이 구성에서 대형 오브젝트(LOB)가 지원되지 않기 때문에 발생합니다. 이 오류는 DB2 버전 5 데이터베이스에 대한 웨어하우스 에이전트의 액세스에 영향을 주지 않습니다.

2000년 6월에 사용할 수 있는 DB2 Universal Database 버전 5의 FixPak 14는 DB2 Connect를 통해 DB2 버전 5 데이터에 액세스하는 데 필수적입니다. 해당 FixPak에서 APAR 번호 JR14507을 참조하십시오.

정보 카탈로그 설정

"정보 카탈로그 설정"의 첫번째 절에 있는 단계 2는 다음과 같습니다.

```
When you install either the DB2 Warehouse Manager
or the DB2 OLAP Server, a default information catalog
is created on DB2 Universal Database for Windows NT.
```

이것은 잘못된 내용입니다. 새로운 정보 카탈로그를 정의해야 합니다. "Information Catalog 작성" 절에서 자세한 내용을 참조하십시오.

기타 제품과 메타데이터 교환

"기타 제품과 메타데이터 교환"의 "발표할 OLAP 오브젝트 식별" 절의 두 번째 구문입니다.

When you publish DB2 OLAP Integration Server metadata, a linked relationship is created between an information catalog "dimensions within a multi-dimensional database" object type and a table object in the OLAP Integration Server.

이 내용은 다음과 같이 바뀌어야 합니다.

When you publish DB2 OLAP Integration Server metadata, a linked relationship is created between an information catalog "dimensions within a multi-dimensional database object and a table object".

이 내용은 "Metadata mappings between the Information Catalog Manager and OLAP Server" 절에 있는 부록 C, "Metadata mappings"에도 나타납니다.

flgnxoln 명령을 사용하여 메타데이터 교환

"Exchanging Metadata"에서, "Identifying OLAP objects to publish" 절이 있습니다. 이 절의 끝에서는 flgnxoln 명령을 사용하여 정보 카탈로그로 OLAP 서버 메타데이터를 발행하는 예가 있습니다. 이 예에서 db2olap.ctl 및 db2olap.ff 파일의 디렉토리는 x:\Program Files\sqllib\logging으로 나타나는데, 이는 잘못된 정보입니다. 디렉토리 이름은 87 페이지에 설명된 대로 x:\Program Files\sqllib\exchange이어야 합니다.

MDISDGC 명령을 사용하여 메타데이터 교환

기타 제품과 메타데이터 교환에서 97 페이지의 "Converting MDIS-conforming metadata into a tag language file"절이 있습니다. MS-DOS 명령 프롬프트에서는 MDISDGC 명령을 발행할 수 없습니다. DB2 명령 창에서 MDISDGC 명령을 발행해야 합니다. "Converting a tag language file into MDIS-conforming metadata" 절의 첫번째 문장은 또한 MS-DOS 명령 프롬프트에서 DGMDISC 명령을 발행해야 한다고 합니다. DB2 명령 창에서 DGMDISC 명령을 발행해야 합니다.

프로그램 호출

정보 카탈로그 관리 안내서에 있는 예제에서는 디렉토리 이름 프로그램 파일이 들어 있는 명령을 보여줍니다. 경로 이름의 일부로서 프로그램 파일이 들어있는 프로그램을 호출할 경우, 큰 따옴표로 프로그램 호출을 묶어야 합니다. 예를 들어 부록 B, "Predefined Information Catalog Manager object types"에는 제목이 "Initializing your information catalog with the predefined object types"인 절에 예가 있습니다. 이 절에 있는 예를 사용하는 경우, DOS 프롬프트에서 실행하면 오류를 받게 됩니다. 올바른 예는 다음과 같습니다.

```
"X:Program Files\SQLLIB\SAMPLES\SAMPDATA\DGWDEMO"  
/T userid password dgname
```

Information Catalog Manager Programming Guide and Reference

Information Catalog Manager 이유 코드

부록 D: Information Catalog Manager 이유 코드에서, 다음의 이유 코드에 대한 가장 오른쪽 컬럼에서 일부 텍스트가 잘릴 수 있습니다. 31014, 32727, 32728, 32729, 32730, 32735, 32736, 32737, 33000, 37507, 37511 및 39206. 텍스트가 잘린 경우, HTML 버전 책을 보고 완전한 컬럼인지 확인하십시오.

Information Catalog Manager 사용자 안내서

"서버 노드 및 원격 정보 카탈로그 등록" 절이 있습니다. 이 절에서는 Information Catalog Manager를 사용하여 원격 정보 카탈로그를 등록하기 전에 호출에서 완료할 수 있는 단계를 나열하고 있습니다. 이 절의 마지막 문단에서는 DB2 제어 센터에서 단계 세트를 완료한 뒤(시스템 추가, 인스턴스 추가 및 데이터베이스 추가), Information Catalog Manager를 열기 전에 제어 센터를 종료해야 합니다. 이 내용은 잘못된 것입니다. Information Catalog Manager를 열기 전에 제어 센터를 종료할 필요가 없습니다.

같은 수정 내용이 온라인 도움말 태스크 "서버 노드 및 원격 정보 카탈로그 등록" 및 레지스터 서버 노드 및 정보 카탈로그 창의 온라인 도움말에도 적용됩니다.

Information Catalog Manager: 온라인 메시지

메시지 FLG0260E

메시지 설명의 두 번째 문장은 다음과 같아야 합니다.

오류로 인해 정보 카탈로그가 구간 복원을 시도했지만, 실패했습니다.
정보 카탈로그가 안정 상태에 있지 않으므로, 변경할 수 없습니다.

메시지 FLG0051E

메시지 설명에서 두 번째 점(bullet) 항목은 다음과 같아야 합니다.

정보 카탈로그에 너무 많은 오브젝트 또는 오브젝트 유형이 있습니다.

관리자 조치는 다음과 같아야 합니다.

Delete some objects or object types from the current
information catalog using the import function.

메시지 FLG0003E

메시지 설명은 다음과 같아야 합니다.

정보 카탈로그는 이를 사용하기 전에 등록되어야 합니다.
정보 카탈로그가 제대로 등록되지 않을 수 있습니다.

메시지 FLG0372E

메시지 설명의 첫번째 문장은 다음과 같아야 합니다.

The ATTACHMENT-IND value was ignored for an object
because that object is an Attachment object.

메시지 FLG0615E

메시지의 두 번째 구문은 다음과 같아야 합니다.

Information Catalog Manager가 예상치 못한 데이터베이스
오류를 발견했거나, 현재 디렉토리 또는 경로에서 바인드
파일을 찾을 수 없습니다.

Information Catalog Manager: 온라인 도움말

정보 카탈로그 창: 선택 항목 메뉴 열기 항목에 대한 "선택된 오브젝트를 엽니다"라는 온라인 도움말은 잘못된 것입니다. 이를 "검색 정의 창을 엽니다"로 수정하십시오.

웹에 대한 Information Catalog Manager

OS/390용 DB2 UDB 시스템에 위치한 정보 카탈로그를 사용할 때 대소문자를 구별하지 않는 검색은 사용할 수 없습니다. 이 점은 간단한 검색과 고급 검색 모두에 해당합니다. 온라인 도움말은 OS/390용 DB2 UDB 정보 카탈로그에 대한 모든 검색이 간단한 검색의 경우 대소문자를 구별함을 설명하지 않습니다. 게다가 모든 그룹화 범주 오브젝트는 아래에 오브젝트가 없어도 펼칠 수 있습니다.

DB2 Warehouse Manager 설치 안내서

웨어하우스 변환기에 대한 소프트웨어 요구사항

웨어하우스 변환기를 사용할 데이터베이스에 JDK(Java Developer's Kit) 버전 1.1.8 이상을 설치해야 합니다.

SAP R/3용 커넥터

SAP R/3 비즈니스 오브젝트 필드의 컬럼을 DB2 테이블에 맵핑하면 생성되는 컬럼 이름 중 일부가 30문자보다 긴 경우가 있습니다. 이 경우 생성된 컬럼 이름은 SAP 필드 이름의 처음 30문자만 반영합니다. 생성된 이름이 원하는 이름이 아니면 테이블의 등록정보 노트북을 사용하여 변경할 수 있습니다.

설치 요구사항

RFC_INI 환경 변수를 참조하십시오(예: Set RFC_INI=c:\rfcapl.ini). 이 변수를 설정한 후 머신을 재부트해야 합니다.

웹용 커넥터

웹용 커넥터 실행 중 문제가 발생하면 IBM Service에 커넥터에 대한 추적을 보내십시오.

웹용 커넥터에 추적을 사용하려면 웨어하우스 센터의 에이전트 추적을 0보다 높은 레벨로 설정하십시오. 추적 파일의 이름은 WSApid.log이고, 여기서 pid는 에이전트에 대한 Windows 프로세스 ID입니다. 추적 파일은 \sqllib\logging 디렉토리에 작성됩니다.

설치 요구사항

JRE(Java run-time environment)나 JVM(Java virtual machine) 버전 1.2.2 이상을 설치하고, 이 버전을 기본값으로 설정하십시오. JRE 버전을 기본값으로 설정하려면 1.2.2 JRE의 경로를 시스템의 PATH 변수에 추가하십시오 (예: C:\JDKs\IBM\java12\bin;). 기본 JRE를 변경한 후에는 머신을 재부트해야 합니다. Java를 설치하지 않은 경우에는 데이터 웨어하우스 커넥터 설치 CD에서 설치할 수 있습니다.

Query Patroller 관리 안내서

DB2 Query Patroller 클라이언트는 별도의 구성요소임

DB2 Query Patroller 클라이언트는 DB2 Administration Client의 일부가 아닌 별도의 구성요소입니다. 이것은 Query Patroller 설치 안내서에 설명된 대로 DB2 Administration Client의 설치 중에 함께 설치되지 않는다는 의미입니다. 대신 Query Patroller 클라이언트를 별도로 설치한다는 것을 의미합니다.

Query Patroller 클라이언트와 Query Patroller 서버의 버전이 동일해야 합니다.

dqpmigrate를 사용하여 DB2 Query Patroller로부터 이주

버전 7 Query Patroller Server가 버전 6 Query Patroller Server 위에 설치되었을 경우 dqpmigrate 명령을 반드시 사용해야 합니다. FixPak 2 이상의 경우, FixPak을 설치하면 이 명령을 대신 수행하므로 직접 dqpmigrate를 수행하지 않아도 됩니다. 이 명령을 사용하지 않으면, v6에 정의된 기본 사용자는 버전 7에서 추가된 여러 새 저장 프로시저에 EXECUTE 특권이 없습니다.

주: dqpmigrate.bnd는 sqllib/bnd 디렉토리에 그리고 dqpmigrate.exe는 sqllib/bin 디렉토리에 있습니다.

dqpmigrate를 수동으로 사용하여 EXECUTE 특권을 부여하려면, FixPak을 설치한 후 다음을 수행하십시오.

1. 다음 명령을 입력하여 /sqllib/bnd/dqpmigrate.bnd 패키지 파일을 Query Patroller 서버가 설치된 데이터베이스에 바인드하십시오.

```
db2 bind dqpmigrate.bnd
```

2. 다음을 입력하여 **dqpmigrate**를 실행하십시오.

```
dqpmigrate dbalias userid passwd
```

조회 관리 사용

"Getting Started" 장의 "Enabling Query Management"에서, 문장은 다음과 같아야 합니다.

```
You must be the owner of the data base, or you must have SYSADM,  
SYSCTRL, or SYSMAINT authority to set database configuration parameters.
```

제어 테이블용 테이블 공간의 위치

"시스템 개요"의 DB2 Query Patroller 제어 테이블에서 다음 텍스트가 이 절의 첫 번째 단락 끝에 추가되었습니다.

DB2 Query Patroller 제어 테이블용 테이블 공간은 단일 노드로 된 노드 그룹에 있어야 하며, 그렇지 않으면 DB2 Query Patroller가 제대로 작동하지 않습니다.

dqpstart 명령에 사용할 수 있는 새로운 매개변수

시작하기의 DB2 Query Patroller 시작 및 중지서 다음 텍스트가 마지막 단락 뒤에 추가되었습니다.

dqpstart 명령에 사용할 수 있는 새로운 매개변수는 다음과 같습니다.

RESTART 매개변수:

호스트 이름 및/또는 dqpnodes.cfg 파일에 지정된 노드의 노드 유형을 바꿀 수 있도록 합니다. DB2 Query Patroller는 이 노드에서 시작됩니다.

주: RESTART 매개변수를 지정하여 DQPSTART 명령을 수행하기 전에 다음 사항을 확인하십시오.

1. 바뀔 호스트에서 DB2 Query Patroller를 중지했어야 합니다.
2. 새 호스트에서 DB2 Query Patroller가 실행되고 있지 않아야 합니다.

구문은 다음과 같습니다.

```
dqpstart nodenum node_num restart hostname server | agent | none
```

ADDNODE 매개변수:

dqpnodes.cfg 파일에 새로운 노드를 추가할 수 있도록 합니다. dqpnodes.cfg 파일에 새로운 노드 항목을 추가하면 DB2 Query Patroller는 추가한 노드에서 시작됩니다. 구문은 다음과 같습니다.

```
dqpstart nodenum node_num addnode hostname server | agent | none
```

DROPNODE 매개변수:

dqnodes.cfg 파일에서 노드를 삭제할 수 있도록 합니다. dqpnodes.cfg 파일에서 노드 항목을 삭제하기 전에는 DB2 Query Patroller가 이 노드에서 시작됩니다. 구문은 다음과 같습니다.

```
dqpstop nodenum node_num droppnode
```

iwm_cmd 명령에 사용할 수 있는 새로운 매개변수

새로운 **-v** 매개변수가 **iwm_cmd** 명령에 추가되어 지정한 노드에서 수행 중이었던 작업 상태를 복구할 수 있게 되었습니다. 비활동 노드에 있는 작업만 복구할 수 있습니다. 이 명령은 노드 실패시 이 노드에서 수행 중이었던거나 취소 중이었던 작업이 있으면 발행됩니다. "Running" 상태에 있었던 작업은 다시 제출되고 "Queued" 상태로 되돌려집니다. "Cancelling" 상태에 있었던 작업은 "Cancelled" 상태로 설정됩니다.

부분 구문은 다음과 같습니다.

```
▶▶ iwm_cmd -u user_id -p password -v node_id_to_recover
```

node_id_to_recover

작업이 복구될 노드를 지정합니다.

새 레지스트리 변수: DQP_RECOVERY_INTERVAL

iwm_scheduler가 복구 파일을 검색하는 시간 간격을 분 단위로 설정하는 데 사용할 수 있는 DQP_RECOVERY_INTERVAL이라는 새로운 레지스트리 변수가 추가되었습니다. 기본값은 60분입니다.

조회 관리자 시작

"Using QueryAdministrator to Administer DB2 Query Patroller" 장에서는 Windows의 시작 메뉴에서 조회 관리자를 시작하기 위한 지시사항을 제공합니다. 첫 단계는 다음의 텍스트 내용을 포함하고 있습니다.

```
If you are using Windows, you can select DB2  
Query Patroller --> QueryAdministrator  
from the IBM DB2 program group.
```

이 문장은 다음과 같이 바뀌어야 합니다.

```
DB2 Query Patroller --> QueryAdmin.
```

사용자 관리

"Using QueryAdministrator to Administer DB2 Query Patroller" 장의 "User Administration" 절에서, 최대 경과 시간 매개변수에 대한 정의는 값이 0 또는 -1로 설정된 경우, 항상 조회가 완료될 때까지 수행된다고 나와있습니다. 이 내용은 잘못된 것입니다. 이 매개변수는 음수 값으로 설정할 수 없습니다. 따라서 값이 "0"으로 설정된 경우, 항상 조회 완료될 때까지 수행됩니다.

Max Queries 매개변수는 DB2 Query Patroller가 동시에 수행할 최대 작업 수를 지정합니다. Max Queries는 0 - 32767 범위 내의 정수여야 합니다.

작업 대기행렬 작성

"Using QueryAdministrator to Administer DB2 Query Patroller" 장의 "Job Queue Administration" 절에서, "작업 큐 작성"에 대한 단계의 화면 캡처는 두 번째 단계 이후에 표시되어야 합니다. 새로운 작업 대기행렬에 대한 정보 창은 조회 관리자 도구의 작업 대기행렬 관리 페이지에서 새로 작성을 클릭하면 열립니다.

작업 대기행렬 페이지나 작업 대기행렬 탭은 작업 대기행렬 관리 페이지와 작업 대기행렬 관리 탭으로 바뀌어야 합니다.

명령행 인터페이스 사용

DB2 Query Patroller 시스템에서 사용자 권한이 있는 사용자가 조회를 제출하고 결과 테이블을 작성하려면, 이 사용자는 데이터베이스에서 CREATETAB 권한이 필요합니다. 사용자는 DQP_RES_TBLSPC 프로파일 변수가 설정되지 않은 상태로 남아 있거나 DQP_RES_TBLSPC 프로파일 변수가 기본 테이블 공간의 이름으로 설정된 경우에는 데이터베이스에서 CREATETAB 권한이 필요하지 않습니다. 사용자가 기본 테이블 공간에서 테이블을 작성하는 권한을 가지고 있기 때문에, 이 경우, 결과 테이블은 정상적으로 작성됩니다.

Query Enabler 참고

- 키 세트 커서를 사용하는 씨드 파티 조회 도구를 사용할 경우, 조회는 인터셉트되지 않습니다. Query Enabler가 이러한 조회를 인터셉트하도록 하려면, 다음을 포함하도록 db2cli.ini 파일을 수정해야 합니다.

```
[common]
DisableKeySetCursor=1
```

- AIX 클라이언트의 경우, 환경 변수 LIBPATH가 설정되어 있는지 확인하십시오. JDK와 함께 제공되는 라이브러리 libXext.a는 /usr/lib/X11 서브디렉토리의 라이브러리와 호환되지 않습니다. 이것은 Query Enabler GUI에 문제점을 야기합니다.

DB2 Query Patroller 추적 프로그램이 공백 컬럼 페이지를 리턴합니다

FixPak 3에 DB2 Query Patroller 추적 프로그램에 대한 수정사항이 포함되어 있습니다. 추적 프로그램은 이제 아무 컬럼에도 해당되지 않는 조회를 올바르게 보고합니다. 그런 조회의 예가 "SELECT COUNT(*) FROM ..."입니다. 이 종류의 조회는 테이블에서 아무 컬럼에도 해당되지 않으므로, 추적 프로그램은 컬럼 페이지에 공백 페이지를 나타냅니다. 이 공백 컬럼 페이지는 결함이 아닙니다.

Query Patroller 및 복제 도구

Query Patroller 버전 7은 복제 도구(asnapply, asncpp, djra and analyze)의 조회를 인터셉트하므로 이러한 도구가 제대로 작동하지 않습니다. 이 문제를 해결하려면 이 도구를 실행할 때 동적 조회 관리를 해제하십시오.

DB2 Query Patroller 문제점 해결

부록 B, DB2 Query Patroller 클라이언트 문제점 해결에서, Common Query Enabler Problems, 문제점 #2 절의 첫번째 bullet의 텍스트를 다음으로 바꾸어야 합니다.

경로 설정에 jre가 포함되어 있는지 확인하십시오.

제6부 응용프로그램 개발

Administrative API Reference	339	중간 파일 보유.	368
db2ArchiveLog (새 API)	339	백업 및 복원	369
db2ArchiveLog	340	SQL 프로시저어 작성	369
db2ConvMonStream.	342	저장 프로시저어 호출	370
db2DatabasePing(새 API).	342	컴파일된 SQL 프로시저어 분산	370
db2DatabasePing - Ping 데이터베이스.	343	HP-UX 응용프로그램 빌드.	371
db2HistData	345	HP-UX C	371
db2HistoryOpenScan	346	HP-UX C++	371
db2XaGetInfo(새 API)	346	OS/2 응용프로그램 빌드	371
db2XaGetInfo - 자원 관리 프로그램에 대한 정		OS/2용 VisualAge C++ 버전 4.0	371
보 얻기	347	PTX 응용프로그램 빌드.	372
db2XaListIndTrans(sqlxphqr을 대신하는 새 API)	348	ptx/C++	372
db2XaListIndTrans - 2 단계 예약 중 이상 실		Solaris 응용프로그램 빌드	372
패 트랜잭션 목록	349	SPARCompiler C++	372
db2GetSnapshot - 스냅샷 가져오기.	353	Windows 32 비트 운영 체제용 응용프로그램 빌드	373
Forget 로그 레코드	354	VisualAge C++ 버전 4.0	373
sqlaintp - 오류 메시지 가져오기.	354	응용프로그램 개발 안내서	375
sqlbctcq - 테이블 공간 컨테이너 조회 닫기	354	DB2 응용프로그램 코딩.	375
sqlubkp - 데이터베이스 백업	354	Microsoft Visual C++용 IBM DB2 Universal	
sqlureot - 테이블 재구성	355	Database Project 및 Tool Add-in 활성화	375
sqlurestore - 데이터베이스 복원	355	Common DB2 응용프로그램 기법	375
AIX Extended Shared Memory		순차값 생성.	375
Support(EXTSHM)에 관한 문서 오류	355	순차 활동 제어.	376
SQLFUPD	355	순차 오브젝트에서 성능 개선	377
locklist	355	순차 오브젝트 및 식별 컬럼 비교.	378
SQLEDBDESC	356	저장 프로시저어	378
SQLFUPD 문서 오류	356	Linux Java 루틴에서 DECIMAL 유형 실패	378
응용프로그램 빌드 안내서	357	반복적인 저장 프로시저어에서 커서 사용	378
소개	357	OLE 자동 저장 프로시저어 작성.	379
지원되는 소프트웨어	357	복잡한 오브젝트 사용: 사용자 정의 구조화 유형	379
샘플 프로그램	359	컬럼에 구조화 유형 속성 삽입.	379
DB2 응용프로그램 빌드에 대한 일반 정보	360	대형 오브젝트(LOB) 사용	380
파일, Makefile 및 오류 확인 유틸리티 빌드	360	연합 데이터베이스 시스템의 대형 오브젝트	
Java 애플릿 및 응용프로그램 빌드	360	(LOB) 지원.	380
환경 설정	360	DB2의 LOB 검색 방법.	380
OS/2에서의 JDK 레벨	362	응용프로그램의 LOB 위치 지정자 사용 방법	381
HP-UX에 설치된 Java2.	362	LOB에 대한 제한사항	381
SQL 프로시저어 빌드	363	LOB 데이터와 비 LOB 데이터 유형 간 맵핑	381
SQL 프로시저어 환경 설정.	363	시스템 조정.	382
컴파일러 환경 변수 설정	363	DB2 프로그래밍 고려사항	382
컴파일 명령 사용자 정의	364	IBM DB2 OLE DB Provider	382

C 및 C++ 프로그래밍	382	DB2 메시지.	398
저장 프로시저, 함수 및 메소드에 대한 C/C++		DBI 메시지.	398
유형	383	Data Warehouse Center(DWC) 메시지	398
Java 프로그래밍	384	SQL 메시지.	400
PARAMETER STYLE JAVA 프로시저 및		정정된 SQLSTATES.	402
함수에서의 Java 메소드 서명	384	SQL 참조서	403
JDBC 애플릿 서버에 연결.	384	하나의 PDF 파일에 제공되는 SQL 참조서	403
샘플 프로그램	385	언어 요소	403
CLI 안내서 및 참조서	387	이름 지정 규칙 및 내재된 오브젝트 이름 규정	403
Run-time Client를 사용한 데이터베이스 유틸리티		데이터 링크 할당	403
바인딩.	387	표현식.	403
CLI 응용프로그램에서 정적 SQL 사용.	387	구문 도표	403
JDBC/ODBC/CLI 정적 프로파일의 한계	387	OLAP 함수.	404
ADT 변환	388	순차 참조	407
고급 기능 사용.	389	함수	410
다중 스레드 응용프로그램 작성	389	새로운 함수 및 프로시저 사용	410
화면 이동 커서.	389	스칼라 함수.	410
OS/390용 서버측 화면 이동 커서 지원.	389	ABS 또는 ABSVAL	410
복합 SQL 사용	391	DECRYPT_BIN 및 DECRYPT_CHAR	411
저장 프로시저 사용	391	ENCRYPT	412
CLI로 저장 프로시저 작성	391	GETHINT	415
CLI 저장 프로시저 및 자동 바인드	391	IDENTITY_VAL_LOCAL	415
CLI/ODBC 구성 및 샘플 응용프로그램 수행.	392	LCASE 및 UCASE(유니코드)	420
구성 키워드.	392	MQPUBLISH	420
DB2 CLI 함수	392	MQREAD	422
SQLBindFileToParam - LOB 파일 참조를		MQRECEIVE	423
LOB 매개변수에 바인드.	392	MQSEND	425
SQLNextResult - 다음 결과 세트를 다른 명령		MQSUBSCRIBE	426
문 핸들과 연결.	392	MQUNSUBSCRIBE.	428
목적	392	MULTIPLY_ALT	429
구문	392	REC2XML	430
함수 인수	392	ROUND.	435
사용법.	393	WEEK_ISO	437
리턴 코드	394	테이블 함수.	437
진단	394	MQREADALL	437
제한사항	394	MQRECEIVEALL	439
참조서.	394	프로시저	441
Extended Scalar Functions	395	GET_ROUTINE_SAR	442
날짜 및 시간 함수	395	PUT_ROUTINE_SAR	442
DB2 CLI/ODBC/JDBC 추적 기능 사용	395	조회	444
메시지 참조서	397	select-statement/구문 도표	444
메시지 및 SQLSTATE 도움말 보기.	397	select-statement/fetch-first-clause	445
DB2 Connect의 SQLCODE 재매핑 변경.	397	SQL문	445
새 메시지 및 변경된 메시지	397	파티션 키 갱신 지원	445
콜 레벨 인터페이스(CLI) 메시지	397	명령문: ALTER TABLE	445
		명령문: CREATE TABLE.	445

명령문: DECLARE GLOBAL	SIGNAL.	489
TEMPORARY TABLE PARTITIONING	SIGNAL.	490
KEY (column-name,...).	SQL 제한사항	492
명령문: UPDATE.	카탈로그 뷰.	492
유니코드 데이터베이스의 더 큰 색인 키	SYSCAT.SEQUENCES	493
ALTER TABLE	DB2 Stored Procedure Builder	495
CREATE INDEX	DB2 Stored Procedure Builder에 대한 Java 1.2	
CREATE TABLE	지원	495
ALTER SEQUENCE	DB2 저장 프로시저의 원격 디버그	496
ALTER SEQUENCE	Windows, OS/2 또는 UNIX 플랫폼에서 SQL 프	
ALTER TABLE	로시저어 빌드	496
복합 SQL(Embedded)	Solaris 플랫폼에서 DB2 Stored Procedure	
복합 텍스트 명령문(동적)	Builder 사용	496
복합 텍스트 명령문(동적)	알려진 문제점 및 제한사항.	497
CREATE FUNCTION(소스 또는 템플릿)	대만어 로케일에서 DB2 Stored Procedure	
CREATE FUNCTION(SQL 스칼라, 테이블 또	Builder 사용	497
는 행).	UNIX(AIX, Sun Solaris, Linux) 설치 및 Stored	
CREATE METHOD	Procedure Builder	497
CREATE METHOD	OS/390에서 SQL 저장 프로시저어 빌드	498
CREATE SEQUENCE.	SQL 저장 프로시저어	498
CREATE SEQUENCE.	Java 저장 프로시저어 내보내기	498
CREATE TRIGGER	OS/390에서 저장 프로시저어 삽입	498
CREATE TRIGGER	워크스테이션 서버에서 SQL 저장 프로시저어에 대	
CREATE WRAPPER	한 빌드 옵션 설정.	499
DECLARE CURSOR	OS/390에서 빌드된 저장 프로시저어의 WLM 주소	
DELETE	공간을 자동으로 새로 고침.	499
DROP	OS/390에서 Java 저장 프로시저어 개발	499
GRANT(순차 특권)	MQSeries 및 OLE DB용 DB2 테이블 사용자 정	
GRANT(순차 특권)	의 함수(UDF) 빌드	500
INSERT	유니코드 갱신	501
SELECT INTO	소개	501
SET ENCRYPTION PASSWORD	DB2 유니코드 데이터베이스 및 응용프로그램	
SET ENCRYPTION PASSWORD	문서 갱신	503
SET 전이 변수.	SQL 참조서.	504
SET 변수	언어 요소	504
UPDATE	데이터 유형의 승급	504
SQL 프로시저어를 이제는 SQL 제어 명령문이라고	데이터 유형 사이의 유형 변환(CAST)	504
합니다.	할당 및 비교	505
SQL 프로시저어 명령문.	결과 데이터 유형 규칙	507
SQL 프로시저어 명령문.	문자열 변환 규칙	507
FOR	표현식.	508
FOR	술어	508
복합 텍스트 명령문(프로시저어)에 대한 복합 텍	함수	508
스트 명령문 변경사항.	스칼라 함수.	508
RETURN	CLI 안내서 및 참조서	508
RETURN		

	고급 기능 사용.	508
	DB2 CLI 유니코드 응용프로그램 작성. . .	508
	DB2 CLI 및 ODBC	514
	ODBC 유니코드 응용프로그램	514
	데이터 이동 유틸리티 안내 및 참조서	515
	내보내기가져오기/로드 유틸리티 파일 형식 . . .	515

Administrative API Reference

| **db2ArchiveLog (새 API)**

|

db2ArchiveLog - 사용 중인 로그 아카이브

db2ArchiveLog

복구 가능한 데이터베이스에 사용 중인 로그 파일을 닫고 뒤를 자릅니다. User Exit를 사용할 경우, 아카이브 요청을 발행하십시오.

권한 부여

다음 중 하나입니다.

- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*

필수 연결

이 API는 지정된 데이터베이스에 자동으로 연결을 설정합니다. 지정된 데이터베이스와 연결이 이미 존재할 경우, API는 오류를 리턴합니다.

참고

db2ApiDf.h

C API 구문

```
/* File: db2ApiDf.h */
/* API: Archive Active Log */
SQL_API_RC SQL_API_FN
db2ArchiveLog (
    db2Uint32 version,
    void *pDB2ArchiveLogStruct,
    struct sqlca * pSqlca);
typedef struct
{
    char          *piDatabaseAlias;
    char          *piUserName;
    char          *piPassword;
    db2Uint16     iAllNodeFlag;
    db2Uint16     iNumNodes;
    SQL_PDB_NODE_TYPE *piNodeList;
    db2Uint32     iOptions;
}
```

일반 API 구문

```

/* File: db2ApiDf.h */
/* API: Archive Active Log */
SQL_API_RC SQL_API_FN
db2gArchiveLog (
    db2UInt32 version,
    void *pDB2ArchiveLogStruct,
    struct sqlca * pSqlca);
typedef struct
{
    db2UInt32          iAliasLen;
    db2UInt32          iUserNameLen;
    db2UInt32          iPasswordLen;
    char              *piDatabaseAlias;
    char              *piUserName;
    char              *piPassword;
    db2UInt16          iAllNodeFlag;
    db2UInt16          iNumNodes;
    SQL_PDB_NODE_TYPE *piNodeList;
    db2UInt32          iOptions;
}

```

API 매개변수**version**

입력. 두 번째 매개변수, *pDB2ArchiveLogStruct*로서 전달된 변수의 버전 및 릴리스 레벨을 지정합니다.

pDB2ArchiveLogStruct

입력. *db2ArchiveLogStruct* 구조를 가리키는 포인터.

pSqlca

출력. *sqlca* 구조를 가리키는 포인터.

iAliasLen

입력. 데이터베이스 별명의 길이를 바이트 단위로 표시하는 4 바이트 부호 없는 정수.

iUserNameLen

사용자 이름의 길이를 바이트 단위로 표시하는 4 바이트 부호 없는 정수. 사용자 이름을 사용하지 않을 경우 영(0)으로 설정하십시오.

iPasswordLen

입력. 암호의 길이를 바이트 단위로 표시하는 4 바이트 부호 없는 정수. 암호를 사용하지 않을 경우 영(0)으로 설정하십시오.

piDatabaseAlias

입력. 사용 중인 로그가 아카이브될 데이터베이스의 별명(시스템 데이터베이스 디렉토리에 카탈로그화된 대로)이 들어 있는 문자열.

db2ArchiveLog - 사용 중인 로그 아카이브

piUserName

입력. 연결을 시도할 때 사용될 사용자 이름이 들어 있는 문자열.

piPassword

입력. 연결을 시도할 때 사용될 암호가 들어 있는 문자열.

iAllNodeFlag

MPP 전용. 입력. 작업이 db2nodes.cfg 파일에 나열된 모든 노드에 적용될지 여부를 표시하는 플래그. 유효한 값은 다음과 같습니다.

DB2ARCHIVELOG_NODE_LIST

*piNodeList*에 전달되는 노드 목록의 노드에 적용합니다.

DB2ARCHIVELOG_ALL_NODES

모든 노드에 적용합니다. *piNodeList*는 널(NULL)이어야 합니다. 이것은 기본값입니다.

DB2ARCHIVELOG_ALL_EXCEPT

*piNodeList*에 전달된 노드 목록에 있는 노드를 제외한 모든 노드에 적용합니다.

iNumNodes

MPP 전용. 입력. *piNodeList* 배열의 노드 수를 지정합니다.

piNodeList

MPP 전용. 입력. 아카이브 로그 작업을 적용할 노드 번호의 배열을 가리키는 포인터.

iOptions

입력. 앞으로 사용을 위해 예약되어 있습니다.

db2ConvMonStream

사용 주의사항에서, 스냅샷 변수 데이터 스트림 유형 SQLM_ELM_SUBSECTION의 구조는 sqlm_subsection이어야 합니다.

db2DatabasePing(새 API)

db2DatabasePing - Ping 데이터베이스

클라이언트와 데이터베이스 서버 사이에서 기초적인 연결성의 네트워크 응답 시간을 테스트합니다. 이 API는 응용프로그램이 호스트 데이터베이스 서버에 DB2 Connect를 통해(직접 또는 게이트웨이를 통해) 액세스할 때 사용할 수 있습니다.

권한 부여

없음

필수 연결

데이터베이스

참고

db2ApiDf.h

C API 구문

```

/* File: db2ApiDf.h */
/* API: Ping Database */
/* ... */
SQL_API_RC SQL_API_FN
db2DatabasePing (
    db2UInt32 versionNumber,
    void      *pParmStruct,
    struct sqlca * pSqlca);
/* ... */
typedef SQL_STRUCTURE db2DatabasePingStruct
{
    char          iDbAlias[SQL_ALIAS_SZ + 1];
    db2UInt16     iNumIterations;
    db2UInt32     *poElapsedTime;
}

```

일반 API 구문

```

/* File: db2ApiDf.h */
/* API: Ping Database */
/* ... */
SQL_API_RC SQL_API_FN
db2gDatabasePing (
    db2UInt32 versionNumber,
    void      *pParmStruct,
    struct sqlca * pSqlca);
/* ... */
typedef SQL_STRUCTURE db2gDatabasePingStruct
{
    db2UInt16     iDbAliasLength;
    char          iDbAlias[SQL_ALIAS_SZ];
    db2UInt16     iNumIterations;
    db2UInt32     *poElapsedTime;
}

```

API 매개변수

versionNumber

입력. 응용프로그램이 실행하는 DB2 Universal Database 또는 DB2 Connect 제품의 버전 및 릴리스.

주: db2Version710 상수 또는 그 이상이 DB2 버전 7.1 이상에서 사용되어야 합니다.

iDbAliasLength

입력. 데이터베이스 별명 길이.

주: 이 매개변수는 현재 사용되지 않습니다. 앞으로 사용되기 위해 예약된 것입니다.

iDbAlias

입력. 데이터베이스 별명.

주: 이 매개변수는 현재 사용되지 않습니다. 앞으로 사용되기 위해 예약된 것입니다.

iNumIterations

입력. 테스트 요청 반복 수. 값은 1 - 32767 이내여야 합니다.

poElapsedTime

출력. 요소 개수가 iNumIterations과 같은 32 비트 정수 배열에 대한 포인터. 배열에 있는 각 요소에는 한 테스트 요청 반복에 소요되는 마이크로 단위의 시간이 들어갑니다.

주: 응용프로그램은 이 API를 호출하기 전에 이 배열에 메모리를 할당해야 합니다.

pSqlca

출력. *sqlca* 구조를 가리키는 포인터. *Administrative API Reference*에서 이 구조에 대한 자세한 내용을 참조하십시오.

사용 참고

이 API를 호출하기 전에 데이터베이스 연결이 있어야 하며, 그렇지 않으면 오류가 발생합니다.

이 기능은 또한 PING 명령을 사용하여 호출될 수 있습니다. 이 명령에 대한 설명은 *Command Reference*를 참조하십시오.

db2HistData

다음 항목을 테이블 11에 추가해야 합니다. db2HistData 구조의 필드는 다음과 같습니다.

필드 이름	데이터 유형	설명
oOperation	char	테이블 12 참조.
oOptype	char	테이블 13 참조.

다음 테이블이 다음 테이블 11에 추가됩니다.

테이블 12. db2HistData 구조에서 oOperation에 대해 유효한 이벤트 값

값	설명	C 정의	COBOL/FORTRAN 정의
A	테이블 공간 추가	DB2HISTORY_OP_ADD_TABLESPACE	DB2HIST_OP_ADD_TABLESPACE
B	백업	DB2HISTORY_OP_BACKUP	DB2HIST_OP_BACKUP
C	로드 사본	DB2HISTORY_OP_LOAD_COPY	DB2HIST_OP_LOAD_COPY
D	삭제된 테이블	DB2HISTORY_OP_DROPPED_TABLE	DB2HIST_OP_DROPPED_TABLE
F	롤 포워드	DB2HISTORY_OP_ROLLFWD	DB2HIST_OP_ROLLFWD
G	테이블 재구성	DB2HISTORY_OP_REORG	DB2HIST_OP_REORG
L	로드	DB2HISTORY_OP_LOAD	DB2HIST_OP_LOAD
N	테이블 공간 이름 바꾸기	DB2HISTORY_OP_REN_TABLESPACE	DB2HIST_OP_REN_TABLESPACE
O	테이블 공간 삭제	DB2HISTORY_OP_DROP_TABLESPACE	DB2HIST_OP_DROP_TABLESPACE
Q	quiesce	DB2HISTORY_OP_QUIESCE	DB2HIST_OP_QUIESCE
R	복원	DB2HISTORY_OP_RESTORE	DB2HIST_OP_RESTORE
S	통계 수행	DB2HISTORY_OP_RUNSTATS	DB2HIST_OP_RUNSTATS
T	테이블 공간 변경	DB2HISTORY_OP_ALT_TABLESPACE	DB2HIST_OP_ALT_TBS
U	로드 해제	DB2HISTORY_OP_UNLOAD	DB2HIST_OP_UNLOAD

다음 테이블도 추가됩니다.

테이블 13. 유효 oOptype 값 db2HistData 구조

oOperation	oOptype	설명	C/COBOL/FORTRAN 정의
B	F	오프라인	DB2HISTORY_OPTYPE_OFFLINE
	N	온라인	DB2HISTORY_OPTYPE_ONLINE
	I	증분 오프라인	DB2HISTORY_OPTYPE_INCR_OFFLINE
	O	증분 온라인	DB2HISTORY_OPTYPE_INCR_ONLINE
	D	델타 오프라인	DB2HISTORY_OPTYPE_DELTA_OFFLINE
	E	델타 온라인	DB2HISTORY_OPTYPE_DELTA_ONLIN
F	E	로그의 끝	DB2HISTORY_OPTYPE_EOL
	P	특정 시점	DB2HISTORY_OPTYPE_PIT

db2ArchiveLog - 사용 중인 로그 아카이브

L	I	삽입	DB2HISTORY_OPTYPE_INSERT
	R	바꾸기	DB2HISTORY_OPTYPE_REPLACE
Q	S	Quiesce 공유	DB2HISTORY_OPTYPE_SHARE
	U	Quiesce 갱신	DB2HISTORY_OPTYPE_UPDATE
	X	Quiesce 독점	DB2HISTORY_OPTYPE_EXCL
	Z	Quiesce 재설정	DB2HISTORY_OPTYPE_RESET
R	F	오프라인	DB2HISTORY_OPTYPE_OFFLINE
	N	온라인	DB2HISTORY_OPTYPE_ONLINE
	I	증분 오프라인	DB2HISTORY_OPTYPE_INCR_OFFLINE
	O	증분 온라인	DB2HISTORY_OPTYPE_INCR_ONLINE
T	C	컨테이너 추가	DB2HISTORY_OPTYPE_ADD_CONT
	R	재균형(Rebalance)	DB2HISTORY_OPTYPE_REB

db2HistoryOpenScan

다음 값이 **iCallerAction** 매개변수에 추가됩니다.

DB2HISTORY_LIST_CRT_TABLESPACE

다른 필터를 전달하는 CREATE TABLESPACE 및 DROP TABLESPACE
레코드만 선택하십시오.

db2XaGetInfo(새 API)

db2XaGetInfo - 자원 관리 프로그램에 대한 정보 얻기

xa_open 호출이 작성되면 특정 자원 관리 프로그램에 대한 정보를 추출합니다.

권한 부여

없음

필수 연결

데이터베이스

참고

sqlxa.h

C API 구문

```

/* File: sqlxa.h */
/* API: Get Information for Resource Manager */
/* ... */
SQL_API_RC SQL_API_FN
db2XaGetInfo (
    db2UInt32 versionNumber,
    void * pParmStruct,
    struct sqlca * pSqlca);
typedef SQL_STRUCTURE db2XaGetInfoStruct
{
    db2int32 iRmid;
    struct sqlca oLastSqlca;
} db2XaGetInfoStruct;

```

API 매개변수**versionNumber**

입력. 두 번째 매개변수 *pParmStruct*로 전달되는 구조의 버전 및 릴리스 레벨을 지정합니다.

pParmStruct

입력. *db2XaGetInfoStruct* 구조를 가리키는 포인터.

pSqlca

출력. *sqlca* 구조를 가리키는 포인터. *Administrative API Reference*에서 이 구조에 대한 자세한 내용을 참조하십시오.

iRmid 입력. 정보가 필요한 자원 관리 프로그램을 지정합니다.

oLastSqlca

출력. 마지막 XA API 호출에 대한 *sqlca*가 포함됩니다.

주: 마지막으로 실패한 XA API로부터 작성된 *sqlca*만을 검색할 수 있습니다.

db2XaListIndTrans(sqlxphqr을 대신하는 새 API)

db2XaListIndTrans - 2 단계 확약 중 이상 실패 트랜잭션 목록

현재 연결된 데이터베이스에 대한 모든 2단계 확약 중 이상 실패 트랜잭션의 목록을 제공합니다.

범위

이 API는 발행된 노드에만 영향을 줍니다.

권한 부여

다음 중 하나입니다.

- *sysadm*
- *dbadm*

필수 연결

데이터베이스

참고

db2ApiDf.h

C API 구문

```
/* File: db2ApiDf.h */
/* API: List Indoubt Transactions */
/* ... */
SQL_API_RC SQL_API_FN
db2XaListIndTrans (
    db2Uint32 versionNumber,
    void * pParmStruct,
    struct sqlca * pSqlca);

typedef SQL_STRUCTURE db2XaListIndTransStruct
{
    db2XaRecoverStruct * piIndoubtData;
    db2Uint32          iIndoubtDataLen;
    db2Uint32          oNumIndoubtsReturned;
    db2Uint32          oNumIndoubtsTotal;
    db2Uint32          oReqBufferLen;
} db2XaListIndTransStruct;
typedef SQL_STRUCTURE db2XaRecoverStruct
{
    sqluint32         timestamp;
    SQLXA_XID         xid;
    char              dbalias[SQLXA_DBNAME_SZ];
    char              applid[SQLXA_APPLID_SZ];
    char              sequence_no[SQLXA_SEQ_SZ];
    char              auth_id[SQL_USERID_SZ];
    char              log_full;
    char              connected;
    char              indoubt_status;
    char              originator;
    char              reserved[8];
} db2XaRecoverStruct;
```

API 매개변수

versionNumber

입력. 두 번째 매개변수 *pParmStruct*로 전달되는 구조의 버전 및 릴리스 레벨을 지정합니다.

pParmStruct

입력. *db2XaListIndTransStruct* 구조를 가리키는 포인터.

pSqlca

출력. *sqlca* 구조를 가리키는 포인터. *Administrative API Reference*에서 이 구조에 대한 자세한 내용을 참조하십시오.

piIndoubtData

입력. 2단계 확약 중 이상 실패 데이터가 리턴될 응용프로그램 제공 버퍼를 가리키는 포인터. 2단계 확약 중 이상 실패의 형식은 *db2XaRecoverStruct*입니다.

db2ArchiveLog - 사용 중인 로그 아카이브

응용프로그램은 이 매개변수가 제공한 주소에서 시작하는 *db2XaRecoverStruct* 구조의 크기를 사용하여 2단계 확약 중 이상 실패 트랜잭션의 목록을 통과할 수 있습니다.

값이 널(NULL)이면, DB2는 필요한 버퍼의 크기를 계산하여 이 값을 *oReqBufferLen*에 리턴합니다. *oNumIndoubtsTotal*에는 2단계 확약 중 이상 실패 트랜잭션의 총 수가 포함됩니다. 응용프로그램은 필요한 버퍼 크기를 할당하고 API를 다시 발행할 수 있습니다.

oNumIndoubtsReturned

출력. *pIndoubtData*에 의해 지정된 버퍼에 리턴된 2단계 확약 중 이상 실패 트랜잭션 레코드의 수.

oNumIndoubtsTotal

출력. API 호출 시에 사용 가능한 총 2단계 확약 중 이상 실패 트랜잭션 레코드의 수. *piIndoubtData* 버퍼가 모든 레코드를 수용하기에 너무 작은 경우, *oNumIndoubtsTotal*은 *oNumIndoubtsReturned*에 대한 총계보다 커집니다. 응용프로그램은 모든 레코드를 확보하기 위해 API를 다시 발행할 수 있습니다.

주: 이 수는 자동 또는 경험적 2단계 확약 중 이상 실패 트랜잭션 재동기화의 결과나 2단계 확약 중 이상 실패 상태에 들어가는 다른 트랜잭션의 결과로 API 호출 간에 변경될 수 있습니다.

oReqBufferLen

출력. API 호출 시에 모든 2단계 확약 중 이상 실패 트랜잭션 레코드를 보유할 필수 버퍼 길이. 응용프로그램은 *pIndoubtData*를 널(NULL)로 설정하여 API를 호출함으로써 필수 버퍼 크기를 판별하는 데 이 값을 사용할 수 있습니다. 그런 다음, 이 값은 필수 버퍼를 할당하는 데 사용할 수 있으며 API는 *pIndoubtData*를 할당된 버퍼의 주소로 설정하여 발행할 수 있습니다.

주: 필수 버퍼 크기는 자동 또는 경험적 2단계 확약 중 이상 실패 트랜잭션 재동기화의 결과나 2단계 확약 중 이상 실패 상태에 들어가는 다른 트랜잭션의 결과로 API 호출들 간에 변경될 수 있습니다. 응용프로그램은 이 점을 고려하여 보다 큰 버퍼를 할당할 수 있습니다.

timestamp

출력. 트랜잭션이 2단계 확약 중 이상 실패 상태에 들어간 시간을 지정합니다.

xid

출력. 전역 트랜잭션을 고유 식별하기 위해 트랜잭션 관리 프로그램이 지정하는 XA 식별자를 지정합니다.

dbalias

출력. 2단계 확약 중 이상 실패 트랜잭션이 발견된 데이터베이스의 별명을 지정합니다.

db2ArchiveLog - 사용 중인 로그 아카이브

applid 출력. 데이터베이스 관리 프로그램이 이 트랜잭션에 지정한 응용프로그램 식별자를 지정합니다.

sequence_no

출력. 데이터베이스 관리 프로그램이 확장자로서 *applid*에 지정한 순차 번호를 지정합니다.

auth_id

출력. 트랜잭션을 수행한 사용자의 권한 부여 ID를 지정합니다.

log_full

출력. 이 트랜잭션으로 인해 로그가 가득 찼는지 여부를 나타냅니다. 유효한 값은 다음과 같습니다.

SQLXA_TRUE

이 2단계 확약 중 이상 실패 트랜잭션으로 인해 로그가 가득 찼습니다.

SQLXA_FALSE

이 2단계 확약 중 이상 실패 트랜잭션으로 인해 로그가 가득 차지는 않았습니다.

connected

출력. 응용프로그램의 연결 여부를 나타냅니다. 유효한 값은 다음과 같습니다.

SQLXA_TRUE

트랜잭션이 정상 동기 지점 처리 중이며, 2단계 확약 중 두 번째 단계를 기다리는 중입니다.

SQLXA_FALSE

이전 실패에 의해 트랜잭션이 2단계 확약 중 이상 실패 상태로 남았으며 이제 트랜잭션 관리 프로그램으로부터의 재동기화를 기다리고 있습니다.

indoubt_status

출력. 이 2단계 확약 중 이상 실패 트랜잭션의 상태를 나타냅니다. 유효한 값은 다음과 같습니다.

SQLXA_TS_PREP

트랜잭션이 준비되었습니다. 연결된 매개변수는 트랜잭션이 정상 확약 처리의 두 번째 단계를 기다리고 있는지 또는 오류가 발생하여 트랜잭션 관리 프로그램과의 재동기화가 필요한 지를 판별하는 데 사용할 수 있습니다.

SQLXA_TS_HCOM

트랜잭션이 경험적으로 확약되었습니다.

SQLXA_TS_HROL

트랜잭션이 경험적으로 구간 복원되었습니다.

SQLXA_TS_MACK

트랜잭션에서 파티션 데이터베이스의 노드로부터 확약 승인이 누락되었습니다.

SQLXA_TS_END

트랜잭션이 이 데이터베이스에서 종료되었습니다. 이 트랜잭션은 나중에 다시 활성화되거나 확약 또는 구간 복원됩니다. 트랜잭션 관리 프로그램에 오류가 발생하여 트랜잭션이 완료되지 않을 가능성도 있습니다. 이와 같은 경우, 이 트랜잭션에는 경험적 조치가 필요한데, 잠금을 보유하고 있으며 다른 응용프로그램이 데이터에 액세스하는 것을 방지할 수 있기 때문입니다.

사용 참고

일반 응용프로그램은 현재 연결을 데이터베이스 또는 파티션 데이터베이스 조정자 노드로 설정한 후에 다음의 단계를 수행합니다.

1. *piIndoubtData*를 널(NULL)로 설정하여 **db2XaListIndTrans**를 호출하십시오. 그러면, *oReqBufferLen* 및 *oNumIndoubtsTotal*에 값이 리턴됩니다.
2. *oReqBufferLen*에 리턴된 값을 사용하여 버퍼를 할당하십시오. 이 API의 초기 호출은 *oReqBufferLen*을 확보하기 위한 것이므로 추가 2단계 확약 중 이상 실패 트랜잭션이 있으면 이 버퍼는 크기가 충분하지 않을 수도 있습니다. 응용프로그램은 *oReqBufferLen*보다 큰 버퍼를 제공합니다.
3. 모든 2단계 확약 중 이상 실패 트랜잭션 레코드가 확보되었는지 판별하십시오. 이 작업은 *oNumIndoubtsReturned*를 *oNumIndoubtTotal*과 비교하여 수행할 수 있습니다. *oNumIndoubtsTotal*이 *oNumIndoubtsReturned*보다 크면, 응용프로그램은 위의 단계를 반복할 수 있습니다.

참조

*Administrative API Reference*에서 "sqlxhfrg - Forget Transaction Status", "sqlxphcm - Commit an Indoubt Transaction" 및 "sqlxphrl - Roll Back an Indoubt Transaction"을 참조하십시오.

db2GetSnapshot - 스냅샷 가져오기

db2GetSnapshot API의 구문은 다음과 같아야 합니다.

```
int db2GetSnapshot( unsigned char version;
db2GetSnapshotData *data,
struct sqlca *sqlca);
```

데이터에 설명된 매개변수는 다음과 같습니다.

```
typedef struct db2GetSnapshotData{
```

db2ArchiveLog - 사용 중인 로그 아카이브

```
sqlma *piSqlmaData;  
sqlm_collected *poCollectedData  
void *poBuffer;  
db2uint32 iVersion;  
db2int32 iBufferSize;  
db2uint8 iStoreResult;  
db2uint16 iNodeNumber;  
db2uint32 *poOutputFormat;  
  
}db2GetSnapshotData;
```

Forget 로그 레코드

다음 정보는 부록 F의 MPP Subordinator Prepare 절 다음에 추가됩니다.

이 로그 레코드는 2단계 확약중 이상 트랜잭션이 구간 복원되거나 2단계 확약이 확약된 후 기록되며, 트랜잭션의 종료와 보유된 모든 로그 자원이 해제되었음을 표시합니다. 트랜잭션을 종료하려면 트랜잭션이 경험적으로 완료된 상태에 있어야 합니다.

표 21. Forget 로그 레코드 구조

설명	유형	오프셋(바이트)
로그 헤더	LogManagerLogRecordHeader	0(20)
time	sqluint64	20(8)
총 길이: 28 바이트		

sqlaintp - 오류 메시지 가져오기

다음의 사용법 주의사항이 이 API의 설명에 추가됩니다.

In a multi-threaded application, sqlaintp must be attached to a valid context; otherwise, the message text for SQLCODE -1445 cannot be obtained.

sqlbctcq - 테이블 공간 컨테이너 조회 달기

이 API에 대해 로드는 유효 권한 부여 레벨이 아닙니다.

sqlubkp - 데이터베이스 백업

BackupType 매개변수에 대해 SQLUB_FULL 값이 SQLUB_DB로 바뀌었습니다. 데이터베이스의 모든 테이블 공간을 백업합니다.

새 증분 백업 기능을 지원하기 위해, SQLUB_INCREMENTAL 및 SQLUB_DELTA 매개변수도 추가되었습니다. 증분 백업 이미지는 가장 최근의 성공적인 전체 백업 이후에 변경된 모든 데이터베이스 데이터 모두의 사본입니다. 델타 백업 이미지는 어떤 유형이든 가장 최근의 성공적인 백업 이후에 변경된 모든 데이터베이스 데이터의 사본입니다.

sqlureot - 테이블 재구성

다음 문장이 사용 참고에 추가됩니다.

REORGANIZE TABLE은 색인 확장자에 기반한 색인을 사용할 수 없습니다.

sqlurestore - 데이터베이스 복원

RestoreType 매개변수에 대해 SQLUD_FULL 값이 SQLUD_DB로 바뀌었습니다. 데이터베이스의 모든 테이블 공간을 복원합니다. 이는 오프라인으로 수행됩니다.

새 증분 복원 기능을 지원하기 위해, SQLUD_INCREMENTAL 매개변수도 추가되었습니다.

증분 백업 이미지는 가장 최근의 성공적인 전체 백업 이후에 변경된 모든 데이터베이스 데이터 모두의 사본입니다.

AIX Extended Shared Memory Support(EXTSHM)에 관한 문서 오류

"부록 E. 동시적인 액세스와 스레드된 응용프로그램", 참고 2가 다음과 같아야 합니다.

2. 기본값으로 AIX에서는 32 비트 응용프로그램에 프로세스 당 11개 이상의 공유 메모리 세그먼트를 연결할 수 없으며, 이 중 최대 10개의 공유 메모리 세그먼트를 지역 DB2 연결에 사용할 수 있습니다.

DB2가 있는 EXTSHM을 사용하려면 다음을 수행하십시오.

클라이언트 세션에서:

```
export EXTSHM=ON
```

DB2를 시작할 때:

```
export EXTSHM=ON
db2set DB2ENVLIST=EXTSHM
db2start
```

EEE에서 sqllib/db2profile에 다음 줄을 추가하십시오.

```
EXTSHM=ON
export EXTSHM
```

SQLFUPD

locklist

토큰의 이름이 SQLF_DBTN_LOCKLIST에서 SQLF_DBTN_LOCK_LIST로 변경되었습니다. locklist 매개변수는 SMALLINT에서 64 비트 부호없는 INTEGER로 변경

db2ArchiveLog - 사용 중인 로그 아카이브

되었습니다. 갱신 가능한 데이터베이스 구성 매개변수의 테이블에 다음을 추가해야 합니다.

매개변수 이름	토큰	토큰 값	데이터 유형
locklist	SQLF_DBTN_LOCK_LIST	704	Uint64

이 매개변수에 대한 새 최대값은 524 288입니다.

SQLLEDBDESC

SQLDBCSS(sqlenv에 정의)에 유효한 값의 목록에 두 개의 값이 추가됩니다. 다음과 같습니다.

SQL_CS_SYSTEM_NLSCHAR

문자 유형에 대해 NLS 버전의 비교 루틴을 사용하여 시스템으로부터 순차를 조합.

SQL_CS_USER_NLSCHAR

문자 유형에 대해 NLS 버전의 비교 루틴을 사용하여 사용자로부터 순차를 조합.

SQLFUPD 문서 오류

『Data Structures』의 표53. Updatable Database Configuration Parameters에 dbheap에 대한 토큰 값이 701로 잘못나와 있습니다. 올바른 값은 58입니다.

응용프로그램 빌드 안내서

소개

지원되는 소프트웨어

주: **PHP**. 이제 PHP로 웹 기반 응용프로그램에서 DB2를 액세스할 수 있습니다. PHP는 HTML이 포함된 서버측의 플랫폼 간 스크립트 언어입니다. Unified-ODBC 액세스 방법을 사용하여 DB2를 액세스할 수 있도록 지원하며, 이 경우 사용자 레벨 PHP가 ODBC 호출을 사용하여 DB2와 통신합니다. 표준 ODBC와는 달리 Unified-ODBC 방법을 사용하면 ODBC 레이어를 통하지 않고 DB2 CLI 레이어로 직접 통신이 이루어집니다. DB2에 PHP를 사용하는 것에 대한 자세한 내용은 www.ibm.com/software/data/db2/udb/win02unix/support의 DB2 지원 사이트를 검색하십시오.

AIX

C 및 C++ 컴파일러에 대해 대해서는 다음 버전이 나열되어야 합니다.

AIX 버전 3.6.6용 IBM C 및 C++ 컴파일러(64 비트용 버전 3.6.6.3)
AIX용 IBM C 4.4
IBM VisualAge C++ 버전 4.0

주: 다음 주소에서 이들 컴파일러 버전에 대해 사용할 수 있는 최신 FixPak을 다운로드할 수 있습니다.

<http://www.ibm.com/software/ad/vacpp/service/csd.html>

Micro Focus COBOL 컴파일러에 대해서는 다음 버전이 나열되어야 합니다.

AIX 4.2.1

Micro Focus COBOL 버전 4.0.20 (PRN 12.03 이상)
Micro Focus COBOL 버전 4.1.10 (PRN 13.04 이상)

AIX 4.3

Micro Focus COBOL Server Express 버전 1.0

주: Micro Focus COBOL용 DB2 지원 저장 프로시저 및 AIX 4.3에서의 UDF에 대해서는 DB2 Application Development 웹 페이지를 참조하십시오.

<http://www.ibm.com/software/data/db2/udb/ad>

AIX용 IBM XL Fortran 버전 5.1.0 컴파일러로 64 비트 응용프로그램을 빌드하려면, 컴파일러 및 링크 단계에서 "-q64" 옵션을 사용하십시오. 64 비트 응용프로그램은 이 컴파일러의 이전 버전에서는 지원되지 않음을 기억하십시오.

HP-UX

C++ 컴파일러에 대해 나열된 버전은 다음과 같아야 합니다.

HP aC++, 버전 A.03.25

주: HP는 이전 컴파일러와 새 컴파일러로 컴파일된 오브젝트들 사이의 2진 호환성을 지원하지 않으므로, 이것은 HP-UX의 DB2에 액세스하도록 빌드된 C++ 응용프로그램의 재컴파일을 강제 수행할 것입니다. C++ 응용프로그램은 또한 이 새 컴파일러를 사용하여 예외를 처리하도록 빌드되어야 합니다.

aCC 전이 안내에 관한 URL은 <http://www.hp.com/esy/lang/cpp/tguide>입니다. C++ 비호환성 부분은 다음에 있습니다.

<http://www.hp.com/esy/lang/cpp/tguide/transcontent.html#RN.CVT.1.2>
<http://www.hp.com/esy/lang/cpp/tguide/transcontent.html#RN.CVT.3.3>

C 대 C++ 부분은 다음에 있습니다.

<http://www.hp.com/esy/lang/cpp/tguide/transcontent.html#RN.CVT.3.3.1>

C와 aCC가 호환 가능하더라도, 두 가지의 다른 데이터 유형을 사용할 경우, "main"을 포함하는 오브젝트가 aCC로 컴파일되고 최종 실행 파일이 aCC에 링크되어야 합니다.

Linux

Linux용 DB2에서는 다음 REXX 버전을 지원합니다.

Object REXX Interpreter for Linux Version 2.1

Linux/390

Linux/390용 DB2는 Java, C 및 C++만 지원합니다.

OS/2

C/C++ 컴파일러에 대해 나열된 버전은 다음과 같아야 합니다.

OS/2용 IBM VisualAge C++ 버전 3.6.5 및 버전 4.0

주: 다음 주소에서 이들 컴파일러 버전에 대해 사용할 수 있는 최신 FixPak을 다운로드할 수 있습니다.

<http://www.ibm.com/software/ad/vacpp/service/csd.html>

VisualAge C++ 컴파일러에 대한 이후 서비스 지원에 대한 제한에 대해서는 다음 사이트에서 새로 추가된 절을 참조하십시오.

<http://www-4.ibm.com/software/ad/vacpp/>

Solaris

Micro Focus COBOL 컴파일러에 대해 나열된 버전은 다음과 같아야 합니다.

Micro Focus COBOL Server Express 버전 1.0

Windows 32 비트 운영 체제

IBM VisualAge C++ 컴파일러에 대해 나열된 버전은 다음과 같아야 합니다.

Windows용 IBM VisualAge C++ 버전 3.6.5 및 4.0

주: 다음 주소에서 이들 컴파일러 버전에 대해 사용할 수 있는 최신 FixPak을 다운로드할 수 있습니다.

<http://www.ibm.com/software/ad/vacpp/service/csd.html>

VisualAge C++ 컴파일러에 대한 이후 서비스 지원에 대한 제한에 대해서는 다음 사이트에서 새로 추가된 절을 참조하십시오.

<http://www-4.ibm.com/software/ad/vacpp/>

Micro Focus COBOL 컴파일러에 대해 나열된 버전은 다음과 같아야 합니다.

Micro Focus COBOL 버전 4.0.20

Micro Focus COBOL Net Express 버전 3.0

샘플 프로그램

"오브젝트 링크 및 포함 샘플" 절에 다음의 내용을 추가해야 합니다.

salarycltv

Visual Basic 저장 프로시저어인 salarysrv를 호출하는 Visual C++ DB2 CLI 샘플.

SALSVADO

테이블 staff2에서 중간 급여를 계산하는 32 비트 Visual Basic 및 ADO에 구현된 샘플 OLE 자동 저장 프로시저어(SALSVADO) 및 SALCLADO 클라이언트(SALCLADO).

"Log Management User Exit Samples" 절에 다음의 내용을 추가해야 합니다.

3.1.6 레벨 이상의 ADSM API 클라이언트를 사용하는 AIX의 응용프로그램은 단일 스레드인 경우에도 xlc나 xlc가 아닌 xlc_r이나 xlc_r 컴파일러를 사용하여 빌드해야 합니다. 그러면, 라이브러리의 스레드가 보안됩니다. 이 점은 Log Management User Exit Samples인 db2uext2.cadsm에 적용됩니다.

스레드가 보안되지 않은 라이브러리로 컴파일된 응용프로그램이 있는 경우, 수정 테스트 IC21925E를 적용하거나 응용프로그램 공급자에게 문의할 수 있습니다. 수정 테스트는 index.storsys.ibm.com anonymous ftp 서버에서 사용할 수 있습니다. 이것은 ADSM API 레벨을 3.1.3으로 되돌립니다.

DB2 응용프로그램 빌드에 대한 일반 정보

파일, Makefile 및 오류 확인 유틸리티 빌드

표16의 b1devm에 대한 항목은 다음과 같습니다.

b1devm

이벤트 모니터 샘플 프로그램, evm(AIX, OS/2 및 Windows 32 비트 운영 체제에서만 사용할 수 있음).

표17에는 다음의 항목이 포함되어야 합니다.

bldmevm

Microsoft Visual C++ 컴파일러를 사용하는 이벤트 모니터 샘플 프로그램 evm.

bldvevm

VisualAge C++ 컴파일러를 사용하는 이벤트 모니터 샘플 프로그램 evm.

Java 애플릿 및 응용프로그램 빌드

환경 설정

지원되는 플랫폼에서 IBM JDK 1.1.8을 사용하여 SQLJ 프로그램을 빌드할 경우, JDK 빌드 날짜가 1999년 11월 24일(또는 이후)이어야 합니다. 그렇지 않으면, 컴파일 도중 JNI panic 오류가 발생할 수 있습니다.

지원되는 플랫폼에서 IBM JDK 1.2.2을 사용하여 SQLJ 프로그램을 빌드할 경우, JDK 빌드 날짜가 2000년 4월 17일(또는 이후)이어야 합니다. 그렇지 않으면, 컴파일 도중 유효하지 않은 Java 유형 오류가 발생할 수 있습니다.

하위 절인 **AIX, HP-UX, Linux** 및 **Solaris**에서, JDBC 2.0에 대한 정보를 다음으로 바꾸십시오.

Java 응용프로그램과 함께 JDBC 2.0 드라이버 사용

JDBC 1.22 드라이버는 여전히 모든 운영 체제에서 기본 드라이버입니다. JDBC 2.0의 새로운 기능을 활용하려면, JDK 1.2 지원을 설치해야 합니다. JDBC 2.0의 새로운 기능을 활용하는 응용프로그램을 실행하기 전에 sqllib/java12 디렉토리에서 usejdbc2 명령을 발행하여 환경을 설정해야 합니다. 응용프로그램이 항상 JDBC 2.0 드라이버를

사용할 수 있게 하려면, .profile과 같은 로그인 프로파일이나 .bashrc, .cshrc 또는 .kshrc와 같은 셸 초기화 스크립트에 다음의 행을 추가해보십시오.

```
. sql1lib/java12/usejdbc2
```

usejdbc2는 db2profile 이후에 수행해야 하므로 이 명령은 db2profile을 수행하기 위한 명령 이후에 배치되도록 하십시오.

JDBC 1.22 드라이버로 다시 전환하려면, sql1lib/java12 디렉토리에서 다음의 명령을 실행하십시오.

```
. usejdbc1
```

Java 저장 프로시저 및 UDF와 함께 JDBC 2.0 드라이버 사용

JDBC 2.0 드라이버를 Java 저장 프로시저 및 UDF와 함께 사용하려면, 인스턴스의 분리 사용자 ID를 위한 환경을 설정해야 합니다. 기본 분리 사용자 ID는 db2fenc1입니다. 분리 사용자 ID를 위한 환경을 설정하려면, 다음의 단계를 수행하십시오.

1. .profile과 같은 사용자 ID 프로파일이나 .bashrc, .cshrc 또는 .kshrc와 같은 사용자 ID 셸 초기화 스크립트에 다음의 행을 추가하십시오.

```
. sql1lib/java12/usejdbc2
```

2. CLP에서 다음의 명령을 발행하십시오.

```
db2set DB2_USE_JDK12=1
```

Java UDF 및 저장 프로시저에 대한 JDBC 1.22 드라이버 지원으로 다시 전환하려면, 다음의 단계를 수행하십시오.

1. .profile과 같은 분리 사용자 ID 프로파일이나 .bashrc, .cshrc 또는 .kshrc와 같은 사용자 ID 셸 초기화 스크립트에서 다음의 행을 제거하십시오.

```
. sql1lib/java12/usejdbc2
```

2. CLP에서 다음의 명령을 발행하십시오.

```
db2set DB2_USE_JDK12=
```

응용프로그램이 항상 JDBC 2.0 드라이버를 사용할 수 있게 하려면, .profile과 같은 로그인 프로파일이나 .bashrc, .cshrc 또는 .kshrc와 같은 셸 초기화 스크립트에 다음의 행을 추가할 수 있습니다.

```
. sql1lib/java12/usejdbc2
```

usejdbc2는 db2profile 이후에 수행해야 하므로 이 명령은 db2profile을 수행하기 위한 명령 이후에 배치되도록 하십시오.

HP-UX

Java 저장 프로시저 및 사용자 정의 함수(UDF)는 JDK 1.1이 있는 HP-UX용 DB2에서는 지원되지 않습니다.

Silicon Graphics IRIX

JDK 1.2.2에 대해 Java JIT 컴파일러를 사용하여 오브젝트 유형이 -o32인 SQLJ 응용프로그램을 빌드할 때, SQLJ 변환기가 세그먼트화 결함으로 실패할 경우, 다음 명령을 사용하여 JIT 컴파일러의 사용을 중단하십시오.

```
export JAVA_COMPILER=NONE
```

JDK 1.2.2은 Silicon Graphics IRIX에서 Java SQLJ 프로그램을 빌드하는 데 필요합니다.

Windows 32 비트 운영 체제

Java 저장 프로시저어 및 UDF와 함께 JDBC 2.0 드라이버 사용

JDBC 2.0 드라이버를 Java 저장 프로시저어 및 UDF와 함께 사용하려면, 다음의 단계를 수행하여 환경을 설정해야 합니다.

1. sqllib\java12 디렉토리에서 다음의 명령을 발행하십시오.

```
usejdbc2
```

2. CLP에서 다음의 명령을 발행하십시오.

```
db2set DB2_USE_JDK12=1
```

Java UDF 및 저장 프로시저어에 대한 JDBC 1.22 드라이버 지원으로 다시 전환하려면, 다음의 단계를 수행하십시오.

1. sqllib\java12 디렉토리에서 다음의 명령을 발행하십시오.

```
usejdbc2
```

2. CLP에서 다음의 명령을 발행하십시오.

```
db2set DB2_USE_JDK12=
```

OS/2에서의 JDK 레벨

일부 메시지는 1999년 9월 이전에 발표된 JDK 1.1.8 버전을 수행하는 OS/2에서 표시되지 않습니다. 최신 JDK 버전 1.1.8을 가지고 있는지 확인하십시오.

HP-UX에 설치된 Java2

Java2 저장 프로시저어를 수행하기 위한 공유 라이브러리 경로가 다음과 유사하게 변경되었습니다.

```
export SHLIB_PATH=$JAVADIR/jre/lib/PA_RISC:$JAVADIR/  
jre/lib/PA_RISC/classic:$HOME/sqllib/lib:/usr/lib:$SHLIB_PATH
```

\$JAVADIR은 Java2 SDK가 설치된 위치입니다.

SQL 프로시저어 빌드

SQL 프로시저어 환경 설정

이들 명령어들은 "설정"에서 DB2 환경을 설정하는 명령어의 추가 내용입니다.

SQL 프로시저어 지원의 경우, Application Development Client를 서버에 설치해야 합니다. Application Development Client 설치에 대해서는 사용자 플랫폼의 빠른 시작 서적을 참조하십시오. 사용자 플랫폼에서 DB2가 지원하는 C 및 C++ 컴파일러에 대해서는 "플랫폼별 지원 소프트웨어"를 참조하십시오.

주: OS/2 FAT 파일 시스템에서는 8자 이하로 SQL 프로시저어의 스키마 이름이 제한됩니다. 8자 이상인 스키마 이름은 HPFS 파일 시스템을 사용해야 합니다.

컴파일러 구성은 두 부분 즉, 컴파일러에 대한 환경 변수 설정과, 컴파일 명령 정의로 구성됩니다. 환경 변수에서는 컴파일러의 2진, 라이브러리 및 포함 파일로의 경로를 제공합니다. 컴파일 명령은 DB2가 SQL 프로시저어에 대해 생성한 C 파일들을 컴파일할 때 사용할 전체 명령입니다.

컴파일러 환경 변수 설정

아래 설명하는 것처럼 OS/2, Windows 및 UNIX 기반 운영 체제 시스템에서 환경을 구성하는 규칙은 모두 다릅니다. 어떤 경우는 구성이 필요하지 않습니다. 또 어떤 경우 DB2_SQLROUTINE_COMPILER_PATH DB2 레지스트리 변수는 해당 환경 변수를 설정하는 실행가능 스크립트를 지적하도록 설정되어야 합니다.

주: **db2set** 명령어나 SQL 저장 프로시저어 빌드 옵션 대화 상자를 Stored Procedure Builder에서 사용하여 이 DB2 레지스트리 변수 값을 설정할 수 있습니다. SQL 저장 프로시저어 빌드 옵션 대화 상자를 사용하면 물리적으로 데이터베이스 서버에 액세스하거나 변경사항을 유효하게 하기 위해 데이터베이스 서버를 재시작할 필요가 없습니다.

OS/2에서,

OS/2용 IBM VisualAge C++ 버전 3.6의 경우

```
db2set DB2_SQLROUTINE_COMPILER_PATH="c:\ibmcxxo\bin\setenv.cmd"
```

OS/2용 IBM VisualAge C++ 버전 4의 경우

```
db2set DB2_SQLROUTINE_COMPILER_PATH="c:\ibmcpp40\bin\setenv.cmd"
```

주: 이들 명령어에서는, c: 드라이브에 C++ 컴파일러가 설치된 것으로 가정합니다. 필요하다면 시스템에서 C++ 컴파일러의 위치를 반영하도록 드라이브 또는 경로를 변경하십시오.

Windows 32 비트 운영 체제에서 컴파일러의 환경 변수가 SYSTEM 변수로 설정되면, 아무 구성도 필요하지 않습니다. 그렇지 않으면 다음과 같이 DB2_SQLROUTINE_COMPILER_PATH DB2 레지스트리 변수를 설정하십시오.

Microsoft Visual C++ 버전 5.0의 경우

```
db2set DB2_SQLROUTINE_COMPILER_PATH="c:\devstudio\vc\bin\vcvars32.bat"
```

Microsoft Visual C++ 버전 6.0의 경우

```
db2set DB2_SQLROUTINE_COMPILER_PATH="c:\Microsoft\vc98\bin\vcvars32.bat"
```

Windows용 IBM VisualAge C++ 버전 3.6의 경우

```
db2set DB2_SQLROUTINE_COMPILER_PATH="c:\ibmcxxw\bin\setenv.bat"
```

Windows용 IBM VisualAge C++ 버전 4의 경우

```
db2set DB2_SQLROUTINE_COMPILER_PATH="c:\ibmcppw40\bin\setenv.bat"
```

주: 이들 명령에서는, c: 드라이브에 C++ 컴파일러가 설치된 것으로 가정합니다. 필요하다면 시스템에서 C++ 컴파일러의 위치를 반영하도록 드라이브 또는 경로를 변경하십시오.

UNIX 기반 운영 체제에서, DB2는 저장 프로시저를 처음 컴파일할 때 실행가능한 스크립트 파일 \$HOME/sqllib/function/routine/sr_cpath(컴파일러 환경 변수에 대한 기본값이 들어있음)를 생성합니다. 기본값이 사용자 컴파일러에 적합하지 않으면 이 파일을 편집할 수 있습니다. 또는, 원하는 설정값을 지정하는 다른 실행가능 스크립트의 전체 경로 이름이 들어가도록 DB2_SQLROUTINE_COMPILER_PATH DB2 레지스트리 변수를 설정할 수 있습니다(위 예제 참조).

컴파일 명령 사용자 정의

응용프로그램 개발 클라이언트의 설치에서는 각 플랫폼에서 지원되는 적어도 하나의 컴파일러에 대해 작업하는 기본 컴파일 명령을 제공합니다.

```
AIX: AIX용 IBM C Set++ 버전 3.6.6
Solaris: SPARCompiler C++ 버전 4.2 및 5.0
HP-UX: HP-UX C++ 버전 A.12.00
Linux: GNU/Linux g++ 버전 egcs-2.90.27 980315(egcs-1.0.2 릴리스)
PTX: ptx/C++ 버전 5.2
OS/2: OS/2용 IBM VisualAge C++ 버전 3
Windows NT 및 Windows 2000: Microsoft Visual C++ 버전 5.0 및 6.0
```

다른 컴파일러를 사용하려면, 다음과 같은 명령으로

DB2_SQLROUTINE_COMPILE_COMMAND DB2 레지스트리 변수를 설정해야 합니다.

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=compilation_command
```

여기서, *compilation_command*는 C 또는 C++ 컴파일 명령이며, 여기에는 저장 프로시저를 작성하는 데 필요한 옵션 및 매개변수가 포함됩니다.

컴파일 명령에서는 키워드 SQLROUTINE_FILENAME을 사용하여, 생성된 SQC, C, PDB, DEF, EXP, 메시지 로그 및 공유 라이브러리 파일들의 이름을 교체하십시오. AIX의 경우, 키워드 SQLROUTINE_ENTRY를 사용하여 항목 지정 이름을 바꾸십시오.

주: **db2set** 명령이나 SQL 저장 프로시저 빌드 옵션 대화 상자를 Stored Procedure Builder에서 사용하여 이 DB2 레지스트리 변수 값을 설정할 수 있습니다. SQL 저장 프로시저 빌드 옵션 대화 상자를 사용하면 물리적으로 데이터베이스 서버에 액세스하거나 변경사항을 유효하게 하기 위해 데이터베이스 서버를 재시작할 필요가 없습니다.

다음은 지원되는 서버 플랫폼의 C 또는 C++ 컴파일러에 대한 DB2_SQLROUTINE_COMPILE_COMMAND의 기본값입니다.

AIX

AIX용 IBM C 버전 3.6.6을 사용할 경우

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=x1c -H512 -T512 \  
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.c -bE:SQLROUTINE_FILENAME.exp \  
-e SQLROUTINE_ENTRY -o SQLROUTINE_FILENAME -L$HOME/sql1lib/11b -lc -ldb2
```

AIX용 IBM C Set++ 버전 3.6.6을 사용할 경우

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=x1C -H512 -T512 \  
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.c -bE:SQLROUTINE_FILENAME.exp \  
-e SQLROUTINE_ENTRY -o SQLROUTINE_FILENAME -L$HOME/sql1lib/11b -lc -ldb2
```

이것은 DB2_SQLROUTINE_COMPILE_COMMAND DB2 레지스트리 변수가 설정되지 않은 경우의 기본 컴파일 명령입니다.

주: AIX에서 64 비트 SQL 프로시저를 컴파일하려면, 위의 명령에 -q64 옵션을 추가하십시오.

AIX용 IBM VisualAge C++ 버전 4를 사용할 경우

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="vacbld"
```

vacbld 명령 다음에 구성 파일을 지정하지 않을 경우, DB2는 처음 SQL 프로시저를 작성하려고 할 때 다음의 기본 구성 파일을 작성합니다.

```
$HOME/sql1lib/function/routine/sqlproc.icc
```

DB2_SQLROUTINE_COMPILE_COMMAND에 대해 DB2 레지스트리 값을 설정할 때 자신의 고유한 구성 파일을 지정하면 됩니다.

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="vacbld  
%DB2PATH%/function/sqlproc.icc"
```

HP-UX

HP C 컴파일러 버전 A.11.00.03을 사용할 경우

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=cc +DAportable +ul -Aa +z \  
-I$HOME/sql1lib/include -c SQLROUTINE_FILENAME.c; \  
ld -b -o SQLROUTINE_FILENAME SQLROUTINE_FILENAME.o \  
-L$HOME/sql1lib/lib -ldb2
```

HP-UX C++ 버전 A.12.00을 사용할 경우

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=CC +DAportable +a1 +z -ext \  
-I$HOME/sql1lib/include -c SQLROUTINE_FILENAME.c; \  
ld -b -o SQLROUTINE_FILENAME SQLROUTINE_FILENAME.o \  
-L$HOME/sql1lib/lib -ldb2
```

이것은 DB2_SQLROUTINE_COMPILE_COMMAND DB2 레지스트리 변수가 설정되지 않은 경우의 기본 컴파일 명령입니다.

Linux

GNU/Linux gcc 버전 2.7.2.3을 사용할 경우

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=cc \  
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.c \  
-shared -o SQLROUTINE_FILENAME -L$HOME/sql1lib/lib -ldb2
```

GNU/Linux g++ 버전 egcs-2.90.27 980315(egcs-1.0.2 릴리스)를 사용할 경우

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=g++ \  
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.c \  
-shared -o SQLROUTINE_FILENAME -L$HOME/sql1lib/lib -ldb2
```

이것은 DB2_SQLROUTINE_COMPILE_COMMAND DB2 레지스트리 변수가 설정되지 않은 경우의 기본 컴파일 명령입니다.

PTX

ptx/C 버전 4.5를 사용할 경우

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=cc -KPIC \  
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.c \  
-G -o SQLROUTINE_FILENAME.so -L$HOME/sql1lib/lib -ldb2 ; \  
cp SQLROUTINE_FILENAME.so SQLROUTINE_FILENAME
```

ptx/C++ 버전 5.2를 사용할 경우

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=c++ -KPIC \  
-D RWSTD_COMPILE_INSTANTIATE=0 \  
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.c \  
-G -o SQLROUTINE_FILENAME.so -L$HOME/sql1lib/lib -ldb2 ; \  
cp SQLROUTINE_FILENAME.so SQLROUTINE_FILENAME
```

이것은 DB2_SQLROUTINE_COMPILE_COMMAND DB2 레지스트리 변수가 설정되지 않은 경우의 기본 컴파일 명령입니다.

OS/2

OS/2용 IBM VisualAge C++ 버전 3을 사용할 경우

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="icc -Ge- -Gm+ -w2  
-I%DB2PATH%\include SQLROUTINE_FILENAME.c  
/B\"/NOFREE /NOI /ST:64000\" SQLROUTINE_FILENAME.def  
%DB2PATH%\lib\db2api.lib"
```

이것은 DB2_SQLROUTINE_COMPILE_COMMAND DB2 레지스트리 변수가 설정되지 않은 경우의 기본 컴파일 명령입니다.

OS/2용 IBM VisualAge C++ 버전 4를 사용할 경우

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="vacbld"
```

vacbld 명령 다음에 구성 파일을 지정하지 않을 경우, DB2는 처음 SQL 프로시저어를 작성하려고 할 때 다음의 기본 구성 파일을 작성합니다.

```
%DB2PATH%\function\routine\sqlproc.icc
```

DB2_SQLROUTINE_COMPILE_COMMAND에 대해 DB2 레지스트리 값을 설정할 때 자신의 고유한 구성 파일을 지정하면 됩니다.

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="vacbld  
%DB2PATH%\function\sqlproc.icc"
```

Solaris

SPARCCompiler C 버전 4.2 및 5.0을 사용할 경우

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=cc -xarch=v8plusa -Kpic \  
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.c \  
-G -o SQLROUTINE_FILENAME -L$HOME/sql1lib/lib \  
-R$HOME/sql1lib/lib -ldb2
```

SPARCCompiler C++ 버전 4.2 및 5.0을 사용할 경우

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=CC -xarch=v8plusa -Kpic \  
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.c \  
-G -o SQLROUTINE_FILENAME -L$HOME/sql1lib/lib \  
-R$HOME/sql1lib/lib -ldb2
```

이것은 DB2_SQLROUTINE_COMPILE_COMMAND DB2 레지스트리 변수가 설정되지 않은 경우의 기본 컴파일 명령입니다.

주:

1. 컴파일러 옵션 -xarch=v8plusa가 기본 컴파일러 명령에 추가되었습니다. 372 페이지의 『Solaris 응용프로그램 빌드』에서 이 옵션이 추가된 이유에 대한 자세한 내용을 참조하십시오.
2. Solaris에서 64 비트 SQL 프로시저어를 컴파일하려면, -xarch=v8plusa 옵션을 삭제하고 위의 명령에 -xarch=v9 옵션을 추가하십시오.

Windows NT 및 Windows 2000

주: SQL 프로시저어는 Windows 98 또는 Windows 95에서 지원되지 않습니다.

Microsoft Visual C++ 버전 5.0 및 6.0을 사용할 경우

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=c1 -Od -W2 /TC -D_X86_=1
-I%DB2PATH%\include SQLROUTINE_FILENAME.c /link -dll
-def:SQLROUTINE_FILENAME.def /out:SQLROUTINE_FILENAME.dll
%DB2PATH%\lib\db2api.lib
```

이것은 DB2_SQLROUTINE_COMPILE_COMMAND DB2 레지스트리 변수가 설정되지 않은 경우의 기본 컴파일 명령입니다.

Windows용 IBM VisualAge C++ 버전 3.6을 사용할 경우

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="ilib /GI
SQLROUTINE_FILENAME.def & icc -Ti -Ge- -Gm+ -W2
-I%DB2PATH%\include SQLROUTINE_FILENAME.c
/B\"/ST:64000 /PM:VIO /DLL\" SQLROUTINE_FILENAME.exp
%DB2PATH%\lib\db2api.lib"
```

Windows용 IBM VisualAge C++ 버전 4를 사용할 경우

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="vacbld"
```

vacbld 명령 다음에 구성 파일을 지정하지 않을 경우, DB2는 처음 SQL 프로시저어를 작성하려고 할 때 다음의 기본 구성 파일을 작성합니다.

```
%DB2PATH%\function\routine\sqlproc.icc
```

DB2_SQLROUTINE_COMPILE_COMMAND에 대해 DB2 레지스트리 값을 설정할 때 자신의 고유한 구성 파일을 지정하면 됩니다.

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="vacbld
%DB2PATH%\function\sqlproc.icc"
```

기본 컴파일러 옵션으로 되돌리려면 다음 명령을 사용하여

DB2_SQLROUTINE_COMPILE_COMMAND의 DB2 레지스트리 값을 널(NULL)로 설정하십시오.

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=
```

중간 파일 보유

SQL 프로시저어가 성공적으로 작성되지 않을 경우에 남아 있을 수 있는 중간 파일들은 수동으로 삭제해야 합니다. 이 파일들은 다음 디렉토리에 있습니다.

UNIX \$DB2PATH/function/routine/sqlproc/\$DATABASE/\$SCHEMA/tmp

여기서 \$DB2PATH는 인스턴스가 작성된 디렉토리를 나타내고, \$DATABASE는 데이터베이스 이름을 나타내며, \$SCHEMA는 SQL 프로시저어가 작성된 스키마 이름을 나타냅니다.

OS/2 및 Windows

```
%DB2PATH%\function\routine\sqlproc%\DATABASE%\SCHEMA%\tmp
```


여기서 %DB2PATH%는 인스턴스가 작성되었던 디렉토리를 나타내며, %DATABASE%는 데이터베이스 이름을 나타내고, %SCHEMA%는 SQL 프로시저어가 작성되었던 스키마 이름을 나타냅니다.

백업 및 복원

SQL 프로시저어가 작성되면, 생성된 공유 라이브러리/DLL이 2 MB 이하인 경우 카탈로그 테이블에 생성된 공유 라이브러리/DLL이 카탈로그 테이블에 보관됩니다. 데이터베이스를 백업 및 복원할 때, 생성된 라이브러리/DLL이 2 MB 이하인 모든 SQL 프로시저어들은 카탈로그 테이블에 보관된 버전으로 백업 및 복원됩니다. 생성된 공유 라이브러리/DLL이 2 MB 이상인 SQL 프로시저어의 경우, 데이터베이스 백업 및 복원과 함께 파일 시스템 백업 및 복원도 수행하십시오. 그렇지 않으면, syscat.procedures 카탈로그 테이블에 있는 소스를 사용하여 직접 SQL 프로시저어의 공유 라이브러리/DLL을 다시 작성해야 합니다.

주: 데이터베이스 복구시, 복구되는 데이터베이스에 속하는 파일 시스템의 모든 SQL 프로시저어 실행 파일들은 제거됩니다. 색인 작성 구성 매개변수(indexrec)가 RESTART로 설정되면, 모든 SQL 프로시저어 실행 파일들을 카탈로그 테이블에서 추출하여, 다음 번 연결시 파일 시스템에 다시 놓습니다. 그렇지 않으면 처음 SQL 프로시저어를 실행할 때 SQL 실행 파일들이 추출됩니다.

실행 파일들은 다음 디렉토리에 다시 놓여집니다.

UNIX \$DB2PATH/function/routine/sqlproc/\$DATABASE

여기서 \$DB2PATH는 인스턴스가 작성된 디렉토리를 나타내고, \$DATABASE는 SQL 프로시저어가 작성된 데이터베이스 이름을 나타냅니다.

OS/2 및 Windows

%DB2PATH%\function\routine\sqlproc\%DATABASE%

여기서 %DB2PATH%는 인스턴스가 작성되었던 디렉토리를 나타내고, %DATABASE%는 SQL 프로시저어가 작성되었던 데이터베이스 이름을 나타냅니다.

SQL 프로시저어 작성

SQL 프로시저어를 개발하려면 데이터베이스 관리 프로그램 구성 매개변수 KEEPDIARY를 'NO'로 설정하십시오. SQL 프로시저어가 실행되고 나서 로드된 채로 보존될 경우, 라이브러리는 새로 고칠 수 없고 실행 파일을 파일 시스템에서 삭제할 수 없으므로, 같은 이름의 저장 프로시저어를 삭제하고 재작성하는 데 문제점이 발생할 수도 있습니다. 또한, 실행 파일을 삭제할 수 없으므로 변경사항을 구간 복원하거나 데이터베이스를 삭제하려고 할 때 문제점이 발생하게 됩니다.

KEEPDIARY 매개변수 설정에 대해서는 "응용프로그램 빌드 안내서"의 "설정"에서 '데이터베이스 관리 프로그램 구성 파일 갱신'을 참조하십시오.

주: SQL 프로시저어는 매개변수에 대해 다음과 같은 데이터 유형을 지원하지 않습니다.

- LONG VARCHAR
- 2진 대형 오브젝트(BLOB)
- 문자 대형 오브젝트(CLOB)
- 2 바이트 문자 대형 오브젝트(DBCLOB)

저장 프로시저어 호출

‘CALL 명령 사용’의 첫번째 문단은 다음과 같아야 합니다.

호출 명령을 사용하려면, 저장 프로시저어 이름과 IN 또는 INOUT 매개변수와 더불어 각 OUT 매개변수의 플레이스 홀더로 ‘?’를 입력해야 합니다. CALL 명령 구문에 대해서는 153 페이지의 『CALL』에서 자세한 내용을 참조하십시오.

컴파일된 SQL 프로시저어 분산

주: DB2 서버 간에 컴파일된 SQL 프로시저어를 분산시키려면, 컴파일된 SQL 프로시저어의 소스나 목적지로서 사용되는 모든 DB2 서버에 대해 다음 단계를 수행해야 합니다.

단계 1. FixPak 3을 설치하십시오.

단계 2. **db2updv7** 명령을 발행하여 DB2가 컴파일된 SQL 프로시저어를 추출하고 설치할 수 있게 하십시오.

```
db2updv7 -d database_name
```

SQL 프로시저어를 정의할 때, 프로시저어는 C 프로그램으로 변환되어 사전 처리 컴파일되고, 목표 데이터베이스에 대해 바인드되고, 컴파일되고 링크되어 공유 라이브러리를 작성합니다. 컴파일 및 링크 단계에서는 C 또는 C++ 컴파일러가 데이터베이스 서버 머신에서 사용 가능해야 합니다. 하지만, 일단 SQL 프로시저어를 정의하면, 같은 플랫폼에서 수행되지만 C 또는 C++ 컴파일러를 반드시 액세스할 필요가 없는 DB2 데이터베이스에 컴파일된 형태로 분산될 수 있습니다. DB2는 사용자가 컴파일된 형태의 SQL 프로시저어를 한 데이터베이스에서 추출하여 그것을 컴파일된 형태로 다른 데이터베이스에 설치하도록 허용합니다.

DB2는 추출 및 설치 조작에 대해 명령행 인터페이스 및 프로그래밍 인터페이스를 모두 제공합니다. 명령행 인터페이스는 두 개의 CLP 명령(GET ROUTINE 및 PUT ROUTINE)으로 구성됩니다. 프로그램 인터페이스는 두 개의 내장 저장 프로시저어(GET_ROUTINE_SAR 및 PUT_ROUTINE_SAR)로 구성됩니다. 명령행 인터페이스에 대한 자세한 정보는, Command Reference를 참조하십시오. 프로그래밍 인터페이스에 대한 자세한 정보는, SQL 참조서를 참조하십시오.

컴파일된 SQL 프로시저를 하나의 데이터베이스 서버에서 또다른 데이터베이스 서버로 분산시키려면, 다음 단계를 수행하십시오.

단계 1. 응용프로그램의 일부인 SQL 프로시저 정의를 포함, 응용프로그램을 개방하십시오.

단계 2. 프로시저를 테스트한 후, 각 프로시저의 컴파일된 버전을 다른 파일로 추출하십시오. 자세한 정보는, Command Reference에서 GET ROUTINE 명령이나 SQL 참조서에서 GET_ROUTINE_SAR 저장 프로시저를 참조하십시오.

단계 3. PUT ROUTINE 명령을 발행하거나, PUT_ROUTINE_SAR 저장 프로시저를 호출하여, 2에서 작성한 파일을 사용하여 각 프로시저의 컴파일된 버전을 각 서버에 설치하십시오. 각 데이터베이스 서버의 운영 체제와 DB2 레벨이 동일해야 합니다.

HP-UX 응용프로그램 빌드

HP-UX C

"다중 스레드 응용프로그램"에서, bldmt 스크립트 파일이 다른 컴파일 옵션으로 바뀌었습니다. 새로운 버전은 sqllib/samples/c 디렉토리에 있습니다.

HP-UX C++

빌드 스크립트에서, C++ 컴파일러 변수 CC가 HP aC++ 컴파일러에 대해 aCC로 바뀌었습니다. 바뀐 빌드 스크립트는 sqllib/samples/cpp 디렉토리에 있습니다.

"+u1" 컴파일 옵션은 aCC 컴파일러로 저장 프로시저 및 UDF를 빌드할 때 사용되어야 합니다. 이 옵션을 사용하면 정렬되지 않은 데이터에 액세스할 수 있습니다. HP-UX용 DB2와 함께 제공된 샘플 빌드 스크립트 bldsrv 및 bldudf 그리고 샘플 makefile은 이 옵션을 갱신되지 않습니다. 이들은 이 옵션을 사용하기 전에 이 옵션을 추가하여 교정되어야 합니다. 다음은 bldsrv 및 bldudf 스크립트의 새 컴파일 단계입니다.

```
aCC +DAportable +u1 -Aa +z -ext -I$DB2PATH/include -c $1.C
```

"다중 스레드 응용프로그램"에서, bldmt 스크립트 파일이 다른 컴파일 옵션으로 바뀌었습니다. 새로운 버전은 sqllib/samples/cpp 디렉토리에 있습니다.

OS/2 응용프로그램 빌드

OS/2용 VisualAge C++ 버전 4.0

OS/2 및 Windows의 경우 이 절에 설명된 export 명령 대신 set 명령을 사용하십시오. 예를 들어, set CLI=tbinfo와 같습니다.

‘DB2 CLI 응용프로그램’의 하위 절 ‘Embedded SQL 응용프로그램 빌드 및 수행’에서 OS/2 및 Windows의 경우에는 Embedded SQL 응용프로그램이 cliapi.icc에 의해 링크된 db2api.lib 라이브러리를 필요로 하므로 cli.icc 파일 대신 cliapi.icc 파일을 사용해야 합니다.

PTX 응용프로그램 빌드

ptx/C++

저장 프로시저어와 사용자 정의 함수를 빌드하려면 라이브러리가 -shared 옵션으로 링크되어야 합니다. sqllib/samples 디렉토리에서, makefile, 빌드 스크립트 bldsrv 및 bldudf가 bldsrv의 다음의 링크 단계에서처럼 이 옵션을 포함하도록 갱신되었습니다.

```
c++ -shared -G -o $1 $1.o -L$DB2PATH/lib -ldb2
```

Solaris 응용프로그램 빌드

SPARCompiler C++

Solaris에서 C/C++ 응용프로그램을 실행하고 SQL 프로시저어를 수행할 경우의 문제점

Sun WorkShop Compiler C/C++를 사용할 때, 다음과 같은 오류를 얻는 실행 파일과 관련된 문제가 발생할 수 있습니다.

1. 1행에 구문 오류: ‘(’가 없어야 합니다.
2. ksh: <응용프로그램 이름>:을 실행할 수 없습니다(여기서, 응용프로그램 이름은 컴파일된 실행 파일의 이름입니다).

컴파일러가 libdb2.so와 링크할 때 유효한 실행 파일을 작성하지 않은 문제가 있을 수 있습니다. 이 문제점을 수정하기 위한 한 가지 제안은 컴파일 및 링크 명령에 다음의 컴파일러 옵션을 추가하는 것입니다.

```
-xarch=v8plusa
```

예를 들어, dynamic.sqc 샘플 응용프로그램을 컴파일할 때,

```
embprep dynamic sample
embprep utilemb sample
cc -c utilemb.c -xarch=v8plusa -I/export/home/db2inst1/sqllib/include
cc -o dynamic dynamic.c utilemb.o -xarch=v8plusa -I/export/home/db2inst1/sqllib/include \
-L/export/home/db2inst1/sqllib/lib -R/export/home/db2inst1/sqllib/lib -l db2
```

주:

1. Solaris에서 SQL 프로시저어를 사용하고 있고 DB2_SQLROUTINE_COMPILE_COMMAND 프로파일 변수를 통해 사용자 자

신의 컴파일 문자열을 사용하는 경우, 위에 제공된 컴파일러 옵션을 반드시 포함시키도록 하십시오. 기본 컴파일러 명령에는 이 옵션이 포함됩니다.

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="cc -# -Kpic -xarch=v8plusa -I$HOME/sql1lib/include \  
SQLROUTINE_FILENAME.c -G -o SQLROUTINE_FILENAME -L$HOME/sql1lib/lib -R$HOME/sql1lib/lib -ldb2
```

2. Solaris에서 64 비트 SQL 프로시저어를 컴파일하려면, `-xarch=v8plusa` 옵션을 삭제하고 위의 명령에 `-xarch=v9` 옵션을 추가하십시오.

Windows 32 비트 운영 체제용 응용프로그램 빌드

VisualAge C++ 버전 4.0

OS/2 및 Windows의 경우 이 절에 설명된 `export` 명령 대신 `set` 명령을 사용하십시오. 예를 들어, `set CLI=tbinfo`와 같습니다.

‘DB2 CLI 응용프로그램’의 하위 절 ‘Embedded SQL 응용프로그램 빌드 및 수행’에서 OS/2 및 Windows의 경우에는 Embedded SQL 응용프로그램이 `cliapi.icc`에 의해 링크된 `db2api.lib` 라이브러리를 필요로 하므로 `cli.icc` 파일 대신 `cliapi.icc` 파일을 사용해야 합니다.

응용프로그램 개발 안내서

DB2 응용프로그램 코딩

Microsoft Visual C++용 IBM DB2 Universal Database Project 및 Tool Add-in 활성화

db2vccmd 명령을 수행하기 전에(1 단계), 현재 로그인 ID를 사용하여 Visual C++를 한 번 이상 시작하였다가 중지하도록 하십시오. 처음 Visual C++를 수행할 경우, 사용자 ID에 대한 프로파일이 작성되고 그것은 **db2vccmd** 명령에 의해 갱신됩니다. 한 번도 시작하지 않은 상태에서 **db2vccmd**를 수행하려고 하면, 다음과 같은 오류가 표시될 수 있습니다.

```
"DB2 Project add-in 등록 중... 실패했습니다(rc = 2)."
```

Common DB2 응용프로그램 기법

순차값 생성

순차값을 생성하는 것은 일반적인 데이터베이스 응용프로그램 개발 문제점입니다. 해당 문제점의 최적 솔루션은 순차 오브젝트 및 순차 표현식을 SQL로 사용하는 것입니다. 각 순차 오브젝트는 순차 표현식에 의해서만 액세스될 수 있는 고유하게 명명된 데이터베이스 오브젝트입니다. 두 개의 순차 표현식(PREVVAL 표현식 및 NEXTVAL 표현식)이 있습니다. PREVVAL 표현식은 이전 명령문의 지정 순차에 대해 가장 최근에 생성된 값을 리턴합니다. NEXTVAL 순차 표현식은 순차 오브젝트의 값을 증가시키며 순차 오브젝트의 새 값을 리턴합니다.

순차 오브젝트를 작성하려면, CREATE SEQUENCE 문을 발행하십시오. 예를 들어, 기본 속성을 사용하여 id_values이라는 순차 오브젝트를 작성하려면, 다음 명령문을 발행하십시오.

```
CREATE SEQUENCE id_values
```

순차 오브젝트의 현재 값을 표시하려면, PREVVAL 표현식을 사용하여 VALUES 문을 발행하십시오.

```
VALUES PREVVAL FOR id_values
1
-----
1
1 record(s) selected.
```

반복적으로 순차 오브젝트의 현재 값을 검색할 수 있으며, NEXTVAL 표현식을 발행할 때까지 순차 오브젝트가 리턴하는 값이 변경되지 않습니다. 다음 예에서, NEXTVAL 표현식이 순차 오브젝트의 값을 증가시킬 때까지 PREVVAl 표현식은 1 값을 리턴합니다.

```
VALUES PREVVAl FOR id_values
1
-----
1
1 record(s) selected.VALUES PREVVAl FOR id_values
1
-----
1
1 record(s) selected.VALUES NEXTVAL FOR id_values
1
-----
2
1 record(s) selected.VALUES PREVVAl FOR id_values
1
-----
2
1 record(s) selected.
```

순차 오브젝트의 다음 값으로 컬럼 값을 갱신하려면, 다음과 같이 UPDATE 문에 NEXTVAL 표현식을 포함시키십시오.

```
UPDATE staff
SET id = NEXTVAL FOR id_values
WHERE id = 350
```

순차 오브젝트의 다음 값을 사용하여 테이블에 새로운 행을 삽입하려면, 다음과 같이 INSERT 명령문에 NEXTVAL 표현식을 포함시키십시오.

```
INSERT INTO staff (id, name, dept, job)
VALUES (NEXTVAL FOR id_values, 'Kandil', 51, 'Mgr')
```

PREVVAl 및 NEXTVAL 표현식에 대한 자세한 정보는 *SQL 참조서*를 참조하십시오.

순차 활동 제어

사용자 응용프로그램의 필요성에 맞게 순차 오브젝트의 활동을 조정할 수 있습니다. CREATE SEQUENCE 문을 발행하여 새로운 순차 오브젝트를 작성하고, 기존 순차 오브젝트에 대해 ALTER SEQUENCE 문을 발행할 때 순차 오브젝트의 속성을 변경합니다. 다음은 지정할 수 있는 순차 오브젝트의 일부 속성입니다.

데이터 유형

CREATE SEQUENCE 문의 AS 절은 순차 오브젝트의 숫자 데이터 유형을 지정합니다. *SQL 참조서*의 “SQL Limits” 부록에 지정된 것처럼 데이터 유형은 순차 오브젝트의 가능한 최소 및 최대 값을 결정합니다. 순차 오브젝트의 데

이더 유형을 변경할 수 없으며, 대신 DROP SEQUENCE 문을 발행하고 새로운 데이터 유형으로 CREATE SEQUENCE 문을 발행하여 순차 오브젝트를 삭제해야 합니다.

시작값 CREATE SEQUENCE 문의 START WITH 절은 순차 오브젝트의 초기값을 설정합니다. ALTER SEQUENCE 문의 RESTART WITH 절은 순차 오브젝트의 값을 지정 값으로 재설정합니다.

최소값 MINVALUE 절은 순차 오브젝트의 최소값을 설정합니다.

최대값 MAXVALUE 절은 순차 오브젝트의 최대값을 설정합니다.

증분값 INCREMENT BY 절은 각 NEXTVAL 표현식이 순차 오브젝트에 추가하는 값을 설정합니다. 순차 오브젝트의 값을 감소시키려면, 음의 값을 지정하십시오.

순차 순환

CYCLE 절은 최대 또는 최소 값에 도달한 순차 오브젝트의 값이 다음 NEXTVAL 표현식에서 시작 값으로 리턴하도록 만듭니다.

예를 들어, 0 값으로 시작하며, 최대값 1000을 가지며, 각 NEXTVAL 표현식에서 2씩 증가하며, 최대값에 도달할 때 시작값으로 리턴하는 id_values이라는 순차 오브젝트를 작성하려면, 다음 명령문을 발행하십시오.

```
CREATE SEQUENCE id_values
  START WITH 0
  INCREMENT BY 2
  MAXVALUE 1000
  CYCLE
```

CREATE SEQUENCE 및 ALTER SEQUENCE 문에 대한 자세한 정보는 *SQL 참조서*를 참조하십시오.

순차 오브젝트에서 성능 개선

식별 컬럼과 같이, 순차 오브젝트를 사용하여 값을 생성하면 대체 접근방식과 비교시 일반적으로 응용프로그램의 성능을 개선시킵니다. 오브젝트 순서화와 다른 방법은 현재 값을 저장하는 단일 컬럼 테이블을 작성하고 트리거를 사용하거나 응용프로그램의 제어 아래에서 해당 값을 증가시키는 것입니다. 응용프로그램이 동시에 단일 컬럼 테이블을 액세스하는 분산 환경에서, 테이블에 대해 일련화된 액세스를 강제하는 데 필요한 잠금은 성능에 크게 영향을 미칠 수 있습니다.

순차 오브젝트는 단일 컬럼 테이블 접근과 연관된 잠금 문제를 피하고 메모리에서 순차 값을 캐쉬하여 DB2 응답 시간을 개선시킬 수 있습니다. 순차 오브젝트를 사용하는 응용프로그램의 성능을 최대화하려면, 순차 오브젝트가 순차 값의 적절한 양을 캐쉬하는지 확인하십시오. CREATE SEQUENCE 및 ALTER SEQUENCE 문의 CACHE 절은 DB2가 메모리에서 생성하고 저장하는 순차 값의 최대 수를 지정합니다.

순차 오브젝트가 시스템 장애나 데이터베이스 비활성화로 인해 해당 순서에 값을 도입하지 않고 순서대로 값을 생성해야 하는 경우, CREATE SEQUENCE 문에서 ORDER 및 NO CACHE 절을 사용하십시오. NO CACHE 절은 순차 오브젝트가 새로운 값을 생성할 때마다 데이터베이스 로그에 강제로 쓰도록하므로 일부 사용자 응용프로그램 성능이 저하되지만, 생성된 값에 갭이 표시되지 않게 합니다.

순차 오브젝트 및 식별 컬럼 비교

순차 오브젝트 및 식별 컬럼이 DB2 응용프로그램에 대해 유사한 역할을 하는 것처럼 보이지만, 몇가지 중요한 차이점이 있습니다.

- 식별 컬럼은 단일 테이블에서 컬럼에 대한 값을 자동으로 생성합니다. 순차 오브젝트는 SQL문에서 사용될 수 있는 순차 값을 생성합니다.
- 식별 컬럼은 고유한 값을 생성합니다. CREATE SEQUENCE 또는 ALTER SEQUENCE 문에 CYCLE 절을 포함하면 순차 오브젝트가 중복된 값을 생성하게 합니다.

저장 프로시저어

Linux Java 루틴에서 DECIMAL 유형 실패

이 문제점은 Java용 IBM Developer Kit이 /usr/lib 디렉토리의 그 라이브러리에 대한 링크를 작성하지 않으므로 발생합니다. DB2 루틴에 대한 보안 모델에서는 표준 시스템 라이브러리 외부에서 라이브러리에 액세스하도록 허용하지 않습니다. Linux상의 Java 루틴에서 DECIMAL 지원을 사용할 수 있도록 하려면 다음을 수행하십시오.

1. 루트 권한으로 다음 명령을 발행하여, Java용 IBM Developer Kit에서 /usr/lib/으로 기호적 링크를 작성하십시오.

Java용 IBM Developer Kit 1.1.8의 경우

```
ln -sf /usr/jdk118/lib/linux/native_threads/* /usr/lib/
```

Java용 IBM Developer Kit 1.3의 경우

```
ln -sf /opt/IBMJava2-13/jre/bin/*.so /usr/lib/
```

2. **ldconfig** 명령을 발행하여 시스템 범용 라이브러리 목록을 갱신하십시오.

반복적인 저장 프로시저어에서 커서 사용

Embedded SQL로 작성된 SQL 프로시저어 또는 저장 프로시저어를 사용할 때 오류를 방지하려면, 반복적인 CALL 문을 발행하기 전에 모든 열린 커서를 닫으십시오.

예를 들어, 저장 프로시저어 MYPROC에 다음 코드 부분이 있다고 가정합니다.

```
OPEN c1;  
CALL MYPROC();  
CLOSE c1;
```

MYPROC이 반복적인 CALL 문을 발행할 때 커서 c1이 여전히 열려 있으므로 DB2는 MYPROC이 호출될 때 오류를 리턴합니다. DB2가 리턴하는 특정 오류는 MYPROC 이 커서에서 수행하는 조치에 따라 다릅니다.

MYPROC 호출이 성공하려면, 다음 예에서 보여주는 것과 같이 중첩된 CALL문 앞의 열린 커서를 닫도록 MYPROC을 재작성하십시오.

```
OPEN c1;  
CLOSE c1;  
CALL MYPROC();
```

오류를 방지하기 위해 중첩된 CALL 문 발행 전에 모든 열린 커서를 닫으십시오.

OLE 자동 저장 프로시저어 작성

다음 문단에서 마지막 문장이 "OLE 자동 저장 프로시저어" 절의 두 번째 문단에 추가 되어야 합니다.

```
OLE 자동 오브젝트를 코드화한 후, CREATE PROCEDURE 문을 사용하여 오브젝트의 메소드를 저장 프로시저어로 등록하십시오.  
OLE 자동 저장 프로시저어를 등록하려면,  
CREATE PROCEDURE 문을 LANGUAGE OLE 절과 함께 실행하십시오. 외부 이름은 OLE 자동 오브젝트를 식별하는 OLE progID와 !(느낌표)로 구분되는 메소드 이름으로 구성됩니다. OLE 자동 오브젝트는 in-process 서버(.DLL)로 구현되어야 합니다.
```

복잡한 오브젝트 사용: 사용자 정의 구조화 유형

컬럼에 구조화 유형 속성 삽입

다음 규칙은 Embedded 정적 SQL문에 적용됩니다. 사용자 정의 구조화 유형의 속성을 속성과 동일한 유형의 컬럼에 삽입하려면, 그 유형의 인스턴스를 나타내는 호스트 변수를 괄호로 묶고 이중 점 연산자와 속성 이름을 닫는 괄호에 추가하십시오. 예를 들어, 다음과 같은 상황이 있을 수 있습니다.

- PERSON_T 유형 VARCHAR(30)의 속성 NAME을 포함하는 구조화 유형입니다.
- T1은 유형 VARCHAR(30)의 컬럼 C1을 포함하는 테이블입니다.
- personhv는 프로그래밍 언어로 유형 PERSON_T에 대해 선언된 호스트 변수입니다.

컬럼 C1에 NAME 속성을 삽입하기 위한 적절한 구문은 다음과 같습니다.

```
EXEC SQL INSERT INTO T1 (C1) VALUES ((:personhv)..NAME)
```

대형 오브젝트(LOB) 사용

연합 데이터베이스 시스템의 대형 오브젝트(LOB) 지원

DB2는 세 유형의 대형 오브젝트(LOB), 즉 문자 대형 오브젝트(CLOB), 2 바이트 문자 대형 오브젝트(DBCLOB), 2진 대형 오브젝트(BLOB)의 지원합니다. 다음 DB2 책에서 DB2 LOB 지원에 대한 일반적인 정보를 참조하십시오.

- *DB2 응용프로그램 개발 안내서*
- *DB2 SQL 참조서*
- *DB2 관리 안내서 계획*

연합 데이터베이스 시스템에서는 원격 데이터 소스에 있는 LOB를 액세스하고 조작할 수 있습니다. LOB의 크기가 대형일 경우, 원격 데이터 소스에서 LOB를 전송할 때 시간이 많이 소요됩니다. 그러면 DB2 연합 데이터베이스는 데이터 소스에서 전송할 LOB 데이터를 최소화하고, 요청된 LOB 데이터를 데이터 소스에서 요청하는 응용프로그램으로 직접 전달하며, DB2에서 LOB를 구체화하지 않습니다.

이 절에서는 다음에 대해 설명합니다.

- DB2의 LOB 검색 방법
- 응용프로그램의 LOB 위치 지정자 사용 방법
- LOB에 대한 제한사항
- LOB 데이터와 비 LOB 데이터 유형 간 맵핑
- 시스템 조정

DB2의 LOB 검색 방법

DB2 연합 시스템은 LOB 스트리밍과 LOB 구체화라는 두 메커니즘을 사용하여 LOB를 검색합니다.

LOB 스트리밍: LOB 스트리밍에서는 LOB 데이터가 단계별로 검색됩니다. DB2에서는 완전히 검색된 조회 결과 세트의 데이터에 LOB 스트리밍을 사용합니다. 예를 들어, 다음과 같은 조회가 있다고 가정해 보십시오.

```
SELECT empname, picture FROM orc_emp_table WHERE empno = '01192345'
```

여기서 *picture*는 LOB 컬럼을 나타내고 *orc_emp_table*은 직원 데이터가 포함된 Oracle 테이블을 참조하는 별명을 나타냅니다. Oracle 데이터 소스에 대해 전체 조회를 수행할 경우, DB2 조회 프로세서가 *picture* 컬럼을 스트리밍으로 표시합니다. 실행 시간에 DB2가 스트리밍으로 표시된 LOB가 있음을 발견하면 데이터 소스에서 LOB를 단계별로 검색한 다음, 이 데이터를 응용프로그램의 메모리 공간으로 전송합니다.

LOB 구체화: LOB 구체화에서는 DB2가 원격 LOB 데이터를 검색하고 연합 서버에 지역적으로 저장합니다. DB2는 다음의 경우에 LOB 구체화를 사용합니다.

- LOB 컬럼을 지연하거나 스트리밍할 수 없을 경우
- 데이터를 전송하기 전에 LOB 컬럼에 지역적으로 함수를 적용해야 할 경우. 이는 DB2가 원격 데이터 소스에서는 사용 불가능한 함수를 보완할 경우 발생합니다. 예를 들어, Microsoft SQL Server는 LOB 컬럼에 SUBSTR 함수를 제공하지 않기 때문에, 이를 보완하기 위해 DB2는 LOB 컬럼을 지역적으로 구체화하고 검색된 LOB에 DB2 SUBSTR 함수를 적용합니다.

응용프로그램의 LOB 위치 지정자 사용 방법

응용프로그램은 원격 데이터 소스에 저장된 LOB에 대해 LOB 위치 지정자를 요청할 수 있습니다. LOB 위치 지정자는 호스트 변수에 저장된 4 바이트 값으로, 프로그램이 데이터베이스 시스템에 보유한 LOB 값(또는 LOB 표현식)을 참조할 때 사용할 수 있습니다. 프로그램은 LOB 위치 지정자를 사용하여 LOB 값이 일반 호스트 변수에 저장되었을 때와 같이 LOB 값을 조작할 수 있습니다. LOB 위치 지정자를 사용할 경우, 차이점은 LOB 값을 서버에서 응용프로그램으로 전송하거나 응용프로그램에서 서버로 다시 전송할 필요가 없다는 것입니다. DB2 응용프로그램 개발 안내서에서 LOB 위치 지정자에 대한 자세한 정보를 참조하십시오.

DB2는 원격 데이터 소스에서 LOB를 검색하고 검색한 LOB를 DB2에 저장한 다음, 저장한 LOB에 대해 LOB 위치 지정자를 발행할 수 있습니다. LOB 위치 지정자는 다음의 경우에 해제됩니다.

- 응용프로그램이 "FREE LOCATOR" SQL문을 발행한 경우
- 응용프로그램이 COMMIT문을 발행한 경우
- DB2를 다시 시작한 경우

LOB에 대한 제한사항

LOB를 사용하고 검색할 때는 다음을 고려하십시오.

- DB2는 원격 LOB를 파일 참조 변수에 바인드할 수 없습니다.
- 통과 모드에서는 LOB가 지원되지 않습니다.

LOB 데이터와 비 LOB 데이터 유형 간 맵핑

DB2 LOB 데이터 유형을 데이터 소스에 있는 비 LOB 데이터 유형과 맵핑할 수 있는 경우도 몇 가지 있습니다. DB2 LOB 유형으로 된 컬럼과 데이터 소스에 있는 대응되는 컬럼 간에 맵핑을 작성해야 할 때는 가능하면 대응되는 열에 LOB 데이터 유형을 사용하는 것이 좋습니다.

맵핑을 작성하려면 CREATE TYPE MAPPING DDL문을 사용하십시오. 예를 들어, 다음과 같습니다.

```
CREATE TYPE MAPPING my_oracle_lob FROM sysibm.clob TO SERVER TYPE oracle TYPElong
```

여기서,

| `my_oracle_lob`

| 는 유형 맵핑의 이름입니다.

| `sysibm.clob`

| 은 DB2 CLOB 데이터 유형입니다.

| `oracle` 은 연결할 서버 유형입니다.

| `long` 은 Oracle의 대응되는 데이터 유형입니다.

시스템 조정

원격 LOB를 검색하는 응용프로그램이 명령문을 처리할만한 충분한 시스템 자원이 없음을 나타내는 오류 메시지를 리턴하면 데이터베이스 구성 파일에서 응용프로그램 힙 크기 매개변수인 APPLHEAPSZ의 값을 늘리십시오. 예를 들어, 다음과 같습니다.

```
DB2 UPDATE DB CFG FOR EMPLOYEE USING APPLHEAPSZ 512
```

여기서 *EMPLOYEE*는 조정할 데이터베이스의 이름이고, *512*는 응용프로그램 힙 크기 매개변수의 값입니다.

DB2 프로그래밍 고려사항

IBM DB2 OLE DB Provider

IBM DB2 버전 7.1 FixPak 1 이상을 설치하면 DB2에서 다음 오류가 발생하는 원인이 수정됩니다.

초기설정 제공 프로그램에서의 오류로 인해 연결 테스트에 실패했습니다.
IBM OLE DB Provider를 사용할 수 없습니다.
자세한 내용은 `readme` 파일을 참조하십시오.

DB2용 IBM OLE DB Provider의 사용에 대한 자세한 정보는 <http://www.ibm.com/software/data/db2/udb/ad/v71/oledb.html>을 참조하십시오.

C 및 C++ 프로그래밍

다음 표는 "저장 프로시저어", "사용자 정의 함수 및 메소드 작성" 및 『C 및 C++ 프로그래밍』에 포함된 정보를 보충합니다. 이 표에는 저장 프로시저어, UDF 및 메소드에 대한 SQL 데이터 유형과 C 데이터 유형 사이의 지원되는 맵핑이 나열됩니다.

저장 프로시저어, 함수 및 메소드에 대한 C/C++ 유형

표 22. C/C++ 선언에 맵핑되는 SQL 데이터 유형

SQL 컬럼 유형	C/C++ 데이터 유형	SQL 컬럼 유형 설명
SMALLINT (500 또는 501)	sqlint16	16 비트의 부호가 있는 정수
INTEGER(496 또는 497)	sqlint32	32 비트의 부호가 있는 정수
BIGINT (492 또는 493)	sqlint64	64 비트의 부호가 있는 정수
REAL (480 또는 481)	float	단정밀도 부동 소수점
DOUBLE (480 또는 481)	double	배정밀도 부동 소수점
DECIMAL(p,s) (484 또는 485)	지원되지 않음	십진수 값을 전달하려면, 매개변수가 DECIMAL로부터 유형 변환(CAST)될 수 있는 데이터 유형(예: CHAR 또는 DOUBLE)이 되도록 정의하고 명시적으로 인수를 이 유형으로 유형 변환(CAST)하십시오.
CHAR(n) (452 또는 453)	char[n+1] 여기서 n은 데이터를 보유할만큼 충분히 큼니다. 1<=n<=254	고정 길이의 널로 종료되는 문자열
CHAR(n) FOR BIT DATA (452 또는 453)	char[n+1] 여기서 n은 데이터를 보유할만큼 충분히 큼니다. 1<=n<=254	고정 길이 문자열
VARCHAR(n) (448 또는 449) (460 또는 461)	char[n+1] 여기서 n은 데이터를 보유할만큼 충분히 큼니다. 1<=n<=32 672	널로 종료되는 가변 길이 문자열
VARCHAR(n) FOR BIT DATA (448 또는 449)	struct { sqluint16 length; char[n] }	널로 종료되지 않는 가변 길이 문자열
	1<=n<=32 672	
LONG VARCHAR (456 또는 457)	struct { sqluint16 length; char[n] }	널로 종료되지 않는 가변 길이 문자열
	32 673<=n<=32 700	
CLOB(n) (408 또는 409)	struct { sqluint32 length; char data[n]; }	4 바이트 문자열 길이 표시기가 있는, 널로 종료되지 않는 가변 길이 문자열
	1<=n<=2 147 483 647	
BLOB(n) (404 또는 405)	struct { sqluint32 length; char data[n]; }	4 바이트 문자열 길이 표시기가 있는, 널로 종료되지 않는 가변 길이 문자열
	1<=n<=2 147 483 647	
DATE (384 또는 385)	char[11]	널 종료 문자 양식

표 22. C/C++ 선언에 맵핑되는 SQL 데이터 유형 (계속)

SQL 컬럼 유형	C/C++ 데이터 유형	SQL 컬럼 유형 설명
TIME (388 또는 389)	char[9]	널 종료 문자 양식
TIMESTAMP (392 또는 393)	char[27]	널 종료 문자 양식
주: 다음의 데이터 유형은 WCHARTYPE NOCONVERT 옵션으로 사전에 컴파일된 경우에 DBCS 또는 EUC 환경에서만 사용할 수 있습니다.		
GRAPHIC(n) (468 또는 469)	sqldbchar[n+1] 여기서 n은 데이터를 보유할 고정 길이의 널로 종료되는 2 바이트 문자열 만큼 충분히 큼. 1<=n<=127	
VARGRAPHIC(n) (400 또는 401)	sqldbchar[n+1] 여기서 n은 데이터를 보유할 널로 종료되지 않는 가변 길이 2 바이트 문자열 만큼 충분히 큼. 1<=n<=16 336	
LONG VARGRAPHIC (472 또는 473)	struct { sqluint16 length; sqldbchar[n] }	널로 종료되지 않는 가변 길이 2 바이트 문자열
	16 337<=n<=16 350	
DBCLOB(n) (412 또는 413)	struct { sqluint32 length; sqldbchar data[n]; }	4 바이트 문자열 길이 표시기가 있는, 널로 종료되지 않는 가변 길이 문자열
	1<=n<=1 073 741 823	

Java 프로그래밍

PARAMETER STYLE JAVA 프로시저 및 함수에서의 Java 메소드 서명

CREATE PROCEDURE 또는 CREATE FUNCTION 명령문의 EXTERNAL NAME 절에서 Java 메소드 이름 뒤에 지정한 경우, Java 메소드 서명은 프로시저어나 함수 이름 다음에 지정된 서명에 대한 기본 Java 유형 맵핑에 해당되어야 합니다. 예를 들어, SQL 유형 INTEGER의 기본 Java 맵핑은 "java.lang.Integer"가 아닌 "int"입니다.

JDBC 애플릿 서버에 연결

Java 애플릿이 사용하는 db2java.zip 파일은 JDBC 애플릿 서버와 같은 FixPak 레벨이어야 합니다. 보통 환경에서, db2java.zip은 책의 그림 22에서와 같이 JDBC 애플릿 서버와 수행 중인 웹 서버로부터 로드됩니다. 따라서 레벨이 일치하게 됩니다. 하지만, db2java.zip을 다른 위치로부터 로드하는 Java 애플릿이 구성에 있다면, 불일치가 발생할 수 있습니다. FixPak 2 이전에는, 이것이 예상치 못한 장애를 초래할 수 있습니다. FixPak 2에서는, 두 파일 간의 FixPak 레벨 일치가 연결 시에 엄격히 강화됩니다. 불일치가 발견되면, 연결이 거부되고, 클라이언트는 다음 예외 중 하나를 받게 됩니다.

- db2java.zip이 FixPak 2 이상일 경우

```
COM.ibm.db2.jdbc.DB2Exception: [IBM][JDBC Driver]
CLI0621E Unsupported JDBC server configuration.
```


- db2java.zip이 FixPak 2 이전일 경우

```
COM.ibm.db2.jdbc.DB2Exception: [IBM][JDBC Driver]
CLI0601E Invalid statement handle or statement is closed.
SQLSTATE=S1000
```

불일치가 발생하면, JDBC 애플릿 서버는 다음 메시지 중 하나를 jdbcerr.log 파일에 로그합니다.

- JDBC 애플릿 서버가 FixPak 2 이상일 경우

```
jdbcFSQLConnect: JDBC Applet Server and client (db2java.zip)
versions do not match. Unable to proceed with connection., einfo= -111
```

- JDBC 애플릿 서버가 FixPak 2 이전일 경우

```
jdbcServiceConnection(): Invalid Request Received., einfo= 0
```

샘플 프로그램

"오브젝트 링크 및 포함 샘플" 절에 다음의 내용을 추가해야 합니다.

salarytlvc Visual Basic 저장 프로시저, salarysrv를 호출하는
Visual C++ DB2 CLI 샘플.

SALSVADO 샘플 OLE 자동 저장 프로시저(SALSVADO) 및

SALCLADO 클라이언트(SALCLADO)로, 32 비트 Visual Basic 및 ADO에서
구현되며, staff2 테이블에서 중간 급여를 계산합니다.

CLI 안내서 및 참조서

Run-time Client를 사용한 데이터베이스 유틸리티 바인딩

Run-time Client는 데이터베이스 유틸리티(가져오기, 내보내기, 재구성, 명령행 처리기)와 DB2 CLI 바인드 파일을 이 데이터베이스와 사용하기 전에 각 데이터베이스에 바인드하는 데 사용할 수 없습니다. 그 대신 DB2 Administration Client나 DB2 응용 프로그램 개발 클라이언트를 사용해야 합니다.

이들 데이터베이스 유틸리티와 DB2 CLI 바인드 파일을 해당 데이터베이스와 함께 사용하려면 각 데이터베이스에 바인드해야 합니다. 네트워크 환경에서, 다른 운영 체제에서 수행되거나 DB2의 버전이나 서비스 레벨이 다른 여러 클라이언트를 사용하는 경우, 각 운영 체제와 DB2 버전 조합에 대해 한번씩 유틸리티를 바인드해야 합니다.

CLI 응용프로그램에서 정적 SQL 사용

CLI 응용프로그램에서의 정적 SQL 사용에 대한 자세한 내용은 <http://www.ibm.com/software/data/db2/udb/staticcli/> 웹 사이트를 참조하십시오.

JDBC/ODBC/CLI 정적 프로파일의 한계

JDBC/ODBC/CLI 정적 프로파일은 현재 간단한 응용프로그램을 목표로 합니다. 여러 기능적 구성요소가 있는 복잡한 응용프로그램과 실행 중의 복잡한 프로그램 로직을 위한 것이 아닙니다.

SQL문을 프로파일 세션에서 캡처하려면 성공적으로 실행해야 합니다. 명령문 일치 세션에서, 일치하지 않는 동적 명령문은 동적 JDBC/ODBC/CLI 호출로 계속 실행됩니다.

SQL문은 일치하는 명령문의 유효한 후보로 캡처 및 바인드된 것과 문자마다 동일해야 합니다. 공백 또한 의미를 가집니다. 예를 들어, "COL = 1"은 "COL=1"과 다르게 간주됩니다. 일치 확률을 높이려면 리터럴 대신에 매개변수 표시문자를 사용하십시오.

사전 바인드된 정적 SQL문과 함께 응용프로그램을 실행할 때, 동적 명령문 활동을 제어하는 동적 레지스터는 정적으로 변환되는 명령문에 대해 아무 영향을 주지 않습니다.

응용프로그램이 후속 DML 명령문에 참조된 오브젝트에 대해 DDL 명령문을 발행하면, 캡처 파일에서 이들 모든 명령문을 찾을 수 있습니다. JDBC/ODBC/CLI 정적 프로파일 바인드 도구는 이들을 바인드하려고 시도합니다. 바인드 시도는

VALIDATE(RUN) 바인드 옵션을 지원하는 DBMS를 사용하면 성공하지만 이를 지원하는 않는 것을 사용하면 실패합니다. 이 경우, 응용프로그램은 정적 프로파일을 사용해야 합니다.

데이터베이스 관리자는 응용프로그램 특정 요구사항에 따라 SQL문을 추가, 변경 또는 제거하여 캡처 파일을 편집할 수 있습니다.

ADT 변환

다음은 책의 기존 정보를 대체합니다.

- 새로운 설명자 유형(smallint) SQL_DESC_USER_DEFINED_TYPE_CODE가 다음 값과 함께 있습니다.

```
SQL_TYPE_BASE 0 (이것은 USER_DEFINED_TYPE이 아님)
SQL_TYPE_DISTINCT 1
SQL_TYPE_STRUCTURED 2
이 값은 SQLColAttribute 또는 SQLGetDescField로
조회할 수 있습니다(IRD 전용).
다음 속성은 실제 유형 이름을 확보하기 위해 추가됩니다.
SQL_DESC_REFERENCE_TYPE
SQL_DESC_STRUCTURED_TYPE
SQL_DESC_USER_TYPE
위의 값은 SQLColAttribute 또는 SQLGetDescField를
사용하여 조회할 수 있습니다(IRD 전용).
```

- 응용프로그램에서 필요로 할 경우, SQL_DESC_BASE_TYPE을 추가하십시오. 예를 들어, 응용프로그램이 구조화 유형을 인식하지 못하지만 이를 폐치하거나 삽입하기 위해 다른 코드가 세부사항을 처리하도록 할 수도 있습니다.
- SQL_ATTR_TRANSFORM_GROUP이라고 하는 새 연결 속성을 추가하여 응용프로그램이 변환 그룹을 설정할 수 있도록 하십시오(SQL "SET CURRENT DEFAULT TRANSFORM GROUP"문을 사용하기 보다는).
- CLI가 값 SQL_DESC_USER_DEFINED_TYPE_CODE를 유효한 SQL 유형으로 리턴하도록 하는 SQLSetConnectAttr을 사용하여 설정하거나 조회할 수 있는 SQL_ATTR_RETURN_USER_DEFINED_TYPES라고 하는 새 명령문/연결 속성을 추가하십시오. 이 속성은 변환을 사용하기 전에 필요합니다.
 - 기본적으로, 그 속성은 해제되어 있으며 기본 유형 정보가 SQL 유형으로 리턴되도록 합니다.
 - 이 속성을 설정하면 SQL_DESC_USER_DEFINED_TYPE_CODE가 SQL_TYPE으로 리턴됩니다. 응용프로그램은 SQL_DESC_USER_DEFINED_TYPE_CODE에 대해 확인한 후 적절한 유형 이름을 검색할 것으로 예상됩니다. 이는 SQLColAttribute, SQLDescribeCol 및 SQLGetDescField에 대해 사용 가능하게 됩니다.
- SQLBindParameter는 SQL_C_DEFAULT를 바인드할 때 오류를 제공하지 않습니다. SQLBindParameter가 유형 SQL_USER_DEFINED_TYPE을 지정할 수 있도록

록 허용하는 코드가 없기 때문입니다. 서버에 유입되는 기본 SQL 유형을 근거로 표준 기본 C 유형이 사용됩니다. 예를 들어, 다음과 같습니다.

```
sqlrc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR, SQL_VARCHAR, 30,
                          0, &c2, 30, NULL);
```

고급 기능 사용

다중 스레드 응용프로그램 작성

"다중 스레드 혼합 응용프로그램" 절에 다음의 내용을 추가해야 합니다.

주: 기본 스택 크기를 사용하지 않고, 대신 스택 크기를 최소한 256 000으로 늘리는 것이 좋습니다. DB2에서는 DB2 함수를 호출할 때 최소 스택 크기는 256 000이어야 합니다. 따라서 스택의 총 크기를 할당할 때, 응용프로그램과 DB2 함수 호출을 위한 최소 요구사항을 충족시킬 수 있을 정도의 크기를 할당해야 합니다.

화면 이동 커서

다음 정보는 "화면 이동 커서" 절에 추가되어야 합니다.

OS/390용 서버측 화면 이동 커서 지원

Unix, Windows 및 OS/2 플랫폼용 UDB 클라이언트는 OS/390 버전 7 데이터베이스에 대해 수행할 때 갱신가능한 서버측 화면 이동 커서를 지원합니다. 3층(three-tier) 환경에서 OS/390 화면 이동 커서를 액세스하려면, 클라이언트 및 게이트웨이가 DB2 UDB 버전 7.1, FixPak 3 이상을 실행 중이어야 합니다.

화면 이동 커서를 액세스할 수 있는 2개의 응용프로그램 사용 인터페이스(ODBC 및 JDBC)가 있습니다. JDBC 인터페이스는 정적 화면 이동 커서만을 액세스할 수 있지만, ODBC 인터페이스는 정적 및 keyset 위주(keyset-driven) 서버측 화면 이동 커서를 액세스할 수 있습니다.

커서 속성: 아래 테이블은 ODBC에서 OS/390 버전 7 커서에 대한 기본 속성을 나열합니다.

표 23. ODBC에서 OS/390 커서에 대한 기본 속성

커서 유형	커서 민감성	커서 갱신가능	커서 동시성	커서 화면 이동
포워드 전용 ^a	미지정	갱신가능하지 않음	읽기 전용 동시성	화면 이동 가능하지 않음
정적	민감하지 않음	갱신가능하지 않음	읽기 전용 동시성	화면 이동
keyset 위주(keyset-driven)	민감함	갱신가능	값 동시성	화면 이동

^a 포워드 전용은 FOR UPDATE 절이 없는 화면 이동 커서의 기본 활동입니다. 포워드 전용 커서에서 FOR UPDATE를 지정하면 갱신가능한, 잠금 동시성, 화면 이동 가능하지 않은 커서를 작성합니다.

지원되는 페치 방향: 모든 ODBC 페치 방향은 SQLFetchScroll 또는 SQLExtendedFetch 인터페이스를 통해 지원됩니다.

keyset 위주(keyset-driven) 커서 갱신: keyset 위주(keyset-driven) 커서는 갱신 가능한 커서입니다. CLI 드라이버는 조회가 SELECT ... FOR READ ONLY 조회로서 발행되거나 FOR UPDATE 절이 이미 존재하는 경우를 제외하고 FOR UPDATE 절을 조회에 추가합니다. OS/390용 DB2에서 구현되는 keyset 위주(keyset-driven) 커서는 값 동시성 커서입니다. 값 동시성 커서는 최적 잠금의 결과를 가져오며, 여기서 값은 갱신이나 삭제가 시도될 때까지 보유되지 않습니다. 갱신 또는 삭제가 시도되면, 데이터베이스 서버는 응용프로그램이 검색한 이전 값과 주요 테이블의 현재 값을 비교합니다. 값이 일치하는 경우, 갱신 또는 삭제가 성공합니다. 값이 일치하지 않는 경우, 조치가 실패합니다. 실패한 경우, 응용프로그램은 다시 값을 조회하고 여전히 적용가능하다면 갱신 또는 삭제를 다시 발행해야 합니다.

응용프로그램은 두가지 방식으로 keyset 위주(keyset-driven) 커서를 갱신할 수 있습니다.

- SQLExecute() 또는 SQLExecDirect()가 있는 SQLPrepare()를 사용하여 UPDATE WHERE CURRENT OF "<cursor name>" 또는 DELETE WHERE CURRENT OF "<cursor name>"을 발행하십시오.
- SQLSetPos() 또는 SQLBulkOperations()를 사용하여 결과세트에 행을 추가, 갱신 또는 삭제하십시오.

주: SQLSetPos() 또는 SQLBulkOperations()를 통해 결과 세트에 추가된 행은 서버에서 테이블로 삽입되지만, 서버의 결과 세트에 추가되지는 않습니다. 그러므로, 이러한 행은 갱신가능하지 않으며 다른 트랜잭션이 작성한 변경사항에 민감하지도 않습니다. 그러나, 삽입된 행은 클라이언트에 캐쉬되므로 결과 세트의 일부로 나타납니다. 삽입된 행에 적용되는 모든 트리거는 적용되지 않았던 것처럼 응용프로그램에 나타납니다. 삽입된 행을 갱신가능하고, 민감하게 하고, 적용가능한 트리거의 결과를 보려면, 응용프로그램이 조회를 다시 발행하여 결과 세트를 재생성해야 합니다.

화면 이동 커서 지원 이전에 작성된 응용프로그램의 문제점 해결: 화면 이동 커서 지원은 새로운 기능이므로, OS/390용 UDB 또는 Unix, Windows 및 OS/2용 UDB는 행동 또는 성능 변경이 발생할 수 있습니다. 화면 이동 커서가 지원되기 전에는, 화면 이동 커서를 요청한 응용프로그램이 포워드 전용 커서를 받기 때문에 이런 경우가 발생합니다. 화면 이동 커서 지원하기 전의 응용프로그램의 이전 활동을 복원하려면, 다음과 같은 구성 키워드를 db2cli.ini 파일에 설정하십시오.

표 24. 화면 이동 커서가 지원하기 전에 응용프로그램 활동을 복원하는 구성 키워드 값

구성 키워드 설정	설명
PATCH2=6	화면 이동 커서(keyset 위주(keyset-driven) 및 정적 모두)가 지원되지 않음을 나타내는 메시지를 리턴합니다. CLI는 자동으로 화면 이동 커서에 대한 모든 요청을 포워드 전용 커서로 낮춥니다.
DisableKeysetCursor=1	서버측 및 클라이언트측 keyset 위주(keyset-driven) 화면 이동 커서 모두를 사용하지 않습니다. 이것은 keyset 위주(keyset-driven) 커서가 요청될 때 CLI 드라이버가 응용프로그램에 정적 커서를 부여하도록 강제하는 데 사용될 수 있습니다.
UseServerKeysetCursor=0	클라이언트측 keyset 위주(keyset-driven) 커서 라이브러리를 사용 중인 응용프로그램의 서버측 keyset 위주(keyset-driven) 커서를 사용하지 않으므로 keyset 위주(keyset-driven) 커서를 시뮬레이트합니다. 클라이언트측 커서가 많은 양의 오버헤드를 발생시키고 일반적으로 서버측 커서보다 성능이 나쁘므로 서버측 keyset 위주(keyset-driven) 커서에서 문제점이 발생할 때만 이 옵션을 사용하십시오.

복합 SQL 사용

다음의 주의사항이 책에서 누락되었습니다.

Any SQL statement that can be prepared dynamically, other than a query, can be executed as a statement inside a compound statement.

Note: Inside Atomic Compound SQL, savepoint, release savepoint, and rollback to savepoint SQL statements are also disallowed. Conversely, Atomic Compound SQL is disallowed in savepoint.

저장 프로시저어 사용

CLI로 저장 프로시저어 작성

다음은 CLI 저장 프로시저어에 대해 문서화되지 않은 제한사항입니다.

If you are making calls to multiple CLI stored procedures, the application must close the open cursors from one stored procedure before calling the next stored procedure. More specifically, the first set of open cursors must be closed before the next stored procedure tries to open a cursor.

CLI 저장 프로시저어 및 자동 바인드

다음은 이 책의 정보를 보충하는 내용입니다.

CLI/ODBC 드라이버는 사용자가 적합한 특권 또는 권한을 갖고 있는 경우, 보통 CLI/ODBC 응용프로그램이 데이터베이스에 대해 SQL을 처음 실행할 때 CLI 패키지를 자동으로 바인드합니다. CLI 패키지의 자동 바인드는 저장 프로시저어내에서 수행될 수 없으므로, 응용프로그램이 수행하는 제일 처음 동작이 CLI 저장 프로시저어를 호출하는 것인 경우 발생하지 않습니다. 그러므로, 새 데이터베이스에 대해 CLI 저장 프로시저어를 호출하는 CLI 응용프로그램을 실행하기 전에, 이 명령으로 CLI 패키지를 한번 바인드해야 합니다.

UNIX

```
db2 bind <BNDPATH>/@db2cli.lst blocking all
```

Windows 및 OS/2

```
db2bind "%DB2PATH%\bnd\@db2cli.lst" blocking
```

권장하는 접근 방법은 런타임시 자동 바인드를 예방하기 위해 데이터베이스 작성시 이들 패키지를 항상 바인드하는 것입니다. 사용자에게 특권이 없거나 다른 응용프로그램이 동시에 자동 바인드를 시도하는 경우 자동 바인드가 실패할 수 있습니다.

CLI/ODBC 구성 및 샘플 응용프로그램 수행

구성 키워드

CURRENTFUNCTIONPATH 키워드에서 마지막 단락을 무시하십시오. 올바른 정보는 다음과 같습니다.

이 키워드는 현재 사용자의 스키마 이외의 스키마 이름으로 정의될 수 있었던 저장 프로시저 및 비규정 함수를 분석하는 프로세스의 일부로서 사용됩니다. 스키마 이름의 순서는 함수 및 프로시저 이름이 분석될 순서를 결정합니다. 함수 및 프로시저 분석에 대한 자세한 정보는, SQL 참조서를 참조하십시오.

DB2 CLI 함수

SQLBindFileToParam - LOB 파일 참조를 LOB 매개변수에 바인드

SQLBindFileToParam() CLI 함수의 마지막 매개변수인 IndicatorValue는 현재 "output(deferred)"으로 설명됩니다. 이것은 "input(deferred)"이어야 합니다.

SQLNextResult - 다음 결과 세트를 다른 명령문 핸들과 연결

다음 텍스트가 『DB2 CLI 함수』에 추가되어야 합니다.

목적

스펙: DB2 CLI 7.x

구문

```
SQLRETURN SQLNextResult (SQLHSTMT StatementHandle1  
                          SQLHSTMT StatementHandle2);
```

함수 인수

표 25. SQLNextResult 인수

데이터 유형	인수	사용	설명
SQLHSTMT	StatementHandle	입력	명령문 핸들.

표 25. *SQLNextResult* 인수 (계속)

데이터 유형	인수	사용	설명
SQLHSTMT	<i>StatementHandle</i>	입력	명령문 핸들.

사용법

저장 프로시저는 나감 후에 하나 또는 그 이상의 커서를 열어 두어 여러 개의 결과 세트를 리턴합니다. 첫번째 결과 세트는 저장 프로시저를 호출했던 명령문 핸들을 사용하여 항상 액세스됩니다. 여러 개의 결과 세트가 리턴될 경우, *SQLMoreResults()* 또는 *SQLNextResult()*를 사용하여 결과 세트를 설명하고 가져 올 수 있습니다.

*SQLMoreResults()*를 사용하여 첫번째 결과 세트를 닫고 다음 결과 세트가 처리되도록 하는 반면, *SQLNextResult()*는 *StatementHandle1*의 커서를 닫지 않고 다음 결과 세트를 *StatementHandle2*로 이동합니다. 가져올 결과 세트가 없으면 두 기능은 *SQL_NO_DATA_FOUND*를 리턴합니다.

*SQLNextResult()*를 사용하면, 결과 세트들이 한번 다른 명령문 핸들로 전송되면 어떤 순서로든 처리되도록 합니다. *SQLMoreResults()* 및 *SQLNextResult()*에 대한 혼합된 호출은 *StatementHandle1*에 더 이상의 커서(열린 결과 세트)가 없을 때까지 허용됩니다.

*SQLNextResult()*가 *SQL_SUCCESS*를 리턴하면, 다음 결과 세트는 *StatementHandle1*과 더 이상 연결되지 않습니다. 대신, *SQLExecDirect()* 호출이 *StatementHandle2*에 대한 조회에서 방금 성공적으로 실행된 것 처럼 다음 결과 세트가 *StatementHandle2*와 연결됩니다. 따라서, 커서는 *SQLNumResultSets()*, *SQLDescribeCol()* 또는 *SQLColAttribute()*를 사용하여 설명될 수 있습니다.

*SQLNextResult()*가 호출된 후, 지금 *StatementHandle2*과 연결된 결과 세트는 남은 결과 세트 체인에서 제거되고 *SQLNextResult()* 또는 *SQLMoreResults()*에서 다시 사용될 수 없습니다. 이는 'n' 결과 세트, *SQLNextResult()*가 최대 'n-1' 횟수만큼 성공적으로 호출될 수 있음을 의미합니다.

*SQLFreeStmt()*가 *SQL_CLOSE* 옵션과 함께 호출되거나, *SQLFreeHandle()*이 *HandleType*가 *SQL_HANDLE_STMT*로 설정되어 호출될 경우, 이 명령문 핸들에 대한 모든 보류 중인 결과 세트가 버려집니다.

*SQLNextResult()*는 *StatementHandle2*에 열린 커서가 있거나 *StatementHandle1* 및 *StatementHandle2*가 동일한 연결 상에 있지 않을 경우 *SQL_ERROR*를 리턴합니다. 오류나 경고가 리턴되면, *SQLError()*가 *StatementHandle1*에서 항상 호출되어야 합니다.

주: SQLMoreResults()는 또한 입력 매개변수 값이 SQLParamOptions() 및 SQLBindParameter()로 지정된 배열이 있는 매개변수화된 조회와 함께 작업합니다. 하지만, SQLNextResult()는 이를 지원하지 않습니다.

리턴 코드

- SQL_SUCCESS
- SQL_SUCCESS_WITH_INFO
- SQL_STILL_EXECUTING
- SQL_ERROR
- SQL_INVALID_HANDLE
- SQL_NO_DATA_FOUND

진단

표 26. SQLNextResult SQLSTATE

SQLSTATE	설명	자세한 설명
40003 08S01	통신 링크 실패.	응용프로그램과 데이터 소스 사이의 통신 링크가 함수가 완료되기 전에 실패했습니다.
58004	예상하지 못한 시스템 실패.	복구할 수 없는 시스템 오류입니다.
HY001	메모리 할당 실패.	DB2 CLI가 함수 실행 또는 완료 지원에 필요한 메모리를 할당할 수 없습니다.
HY010	함수 순차 오류.	data-at-execute (SQLParamData(), SQLPutData()) 조작 중에 함수가 호출되었습니다. StatementHandle2에 그와 연결된 열린 커서가 있습니다. BEGIN COMPOUND 및 END COMPOUND SQL 작업 내에 있는 동안 함수가 호출되었습니다.
HY013	예상하지 못한 메모리 처리 오류.	DB2 CLI가 함수 실행 또는 완료 지원에 필요한 메모리에 액세스할 수 없습니다.
HYT00	시간종료 만기.	데이터 소스가 결과 세트를 리턴하기 전에 시간종료 기간이 만기되었습니다. 시간종료는 Windows 3.1 및 Macintosh System 7과 같은 비 멀티태스킹 시스템에서만 지워집니다. 시간종료 기간은 SQLSetConnectAttr()에 대한 SQL_ATTR_QUERY_TIMEOUT 속성을 사용하여 설정할 수 있습니다.

제한사항

SQLMoreResults()만 매개변수화된 조회에 대해 사용할 수 있습니다.

참조서

- 535 페이지의 "SQLMoreResults - 추가 결과 세트가 있는지 판단"
- 120 페이지의 "저장 프로시저로부터 결과 세트 리턴"

Extended Scalar Functions

날짜 및 시간 함수

다음 함수는 "Extended Scalar Functions"의 날짜 및 시간 함수 절에서 누락된 것입니다.

DAYOFWEEK_ISO(date_exp)

*date_exp*에 있는 요일을 1-7 범위내의 정수값으로 리턴합니다. 여기서 1은 월 요일을 나타냅니다. 이 함수와 DAYOFWEEK() 함수의 차이점을 기억하십시오. DAYOFWEEK() 함수에서는 1이 일요일을 나타냅니다.

WEEK_ISO(date_exp)

*date_exp*에 있는 일년중 주수를 1-53 범위내의 정수값으로서 리턴하십시오. 1주는 목요일이 들어가는 연도의 첫번째 주로서 정의됩니다. 그러므로, 1주는 1월 4일 들어가는 첫번째 주와 같으므로, 월요일이 주의 첫번째 날로서 간주됩니다.

WEEK_ISO()는 WEEK()의 현재 정의와 다름을 기억하십시오. 이 WEEK()는 최대 54까지의 값을 리턴합니다. WEEK() 함수의 경우, 주 1은 첫번째 토요일이 들어있는 주입니다. 이것은 주에 1일만 있는 경우라도 1월 1일이 들어있는 주와 같습니다.

DAYOFWEEK_ISO() 및 WEEK_ISO()는 버전 7에서 작성된 데이터베이스에서는 자동으로 사용 가능합니다. 버전 7 이전에서 작성된 데이터베이스에서는 이러한 함수를 사용할 수 없습니다. 그러한 데이터베이스에서 DAYOFWEEK_ISO() 및 WEEK_ISO() 함수를 사용 가능하게 하려면, **db2updb** 시스템 명령어를 사용하십시오. **db2updb**에 대한 내용은, 이 릴리스 정보에 있는 "Command Reference" 절을 참조하십시오.

DB2 CLI/ODBC/JDBC 추적 기능 사용

이 부록의 절이 갱신되었습니다. 이 추적 기능에 관한 최근 정보는 문제점 해결 안내서의 『추적』 장을 참조하십시오.

메시지 참조서

메시지 및 SQLSTATE 도움말 보기

명령행 처리기에서 사용 가능한 도움말에는 메시지 참조서에서는 사용 가능하지 않은 SQLSTATE 값과 메시지에 대해 갱신된 새로운 도움말이 포함됩니다.

명령행 처리기에서 도움말을 표시하려면 운영 체제의 명령 프롬프트에서 다음 명령을 입력하십시오.

```
db2 "? XXXnnnnn"
```

여기서 XXX는 메시지 접두부를 나타내고 nnnnn은 메시지 번호를 나타냅니다.

예를 들어, db2 "? SQL30081"은 SQL30081 메시지에 대한 도움말을 표시합니다.

명령행 처리기에서 SQLSTATE 텍스트를 표시하려면 운영 체제의 명령 프롬프트에서 다음 명령을 입력하십시오.

```
db2 "? XXXXX"
```

여기서 XXXXX는 SQLSTATE 값을 나타냅니다.

예를 들어, db2 "? 428F1"은 SQLSTATE 428F1에 대한 텍스트를 표시합니다.

DB2 Connect의 SQLCODE 재매핑 변경

버전 7.2에서는 DB2 Connect에 대한 기본 SQLCODE 재매핑이 변경되었습니다. 이제 호스트 데이터베이스가 SQLCODE 값 -567을 리턴하면, DB2 Connect는 DB2 클라이언트로 리턴하기 전에 SQLCODE 값을 -551로 재매핑합니다.

새 메시지 및 변경된 메시지

다음 목록에는 DB2 버전 7.1용 메시지 참조서가 출판된 이후 변경된 메시지의 메시지 번호가 있습니다. DB2 사용 중 이러한 메시지가 나타나면 갱신된 메시지가 제대로 설치된 것이지만, 이 메시지가 메시지 참조서의 정보와 일치하지는 않습니다.

콜 레벨 인터페이스(CLI) 메시지

CLI0645E

CLI0646E

CLI0647E

DB2 메시지

DB21086I
DB210060E
DB210061E
DB210062E
DB210113E
DB210114E
DB210115E
DB210116E
DB210117E
DB210118E
DB210120E
DB210121E
DB210200I
DB210201I

DBI 메시지

DBI1172E
DBI1793W
DBI1794E
DBI1795E
DBI1796W
DBI1797I

Data Warehouse Center(DWC) 메시지

DWC0000I
DWC03504E
DWC08900E
DWC08901E
DWC08902E
DWC08903E
DWC08904E
DWC08907C
DWC08908C

DWC08909C
DWC08910E
DWC08911E
DWC08912E
DWC08913E
DWC08914E
DWC08915E
DWC08917E
DWC08919I
DWC08930E
DWC08931E
DWC08932E
DWC08933E
DWC08934E
DWC08935E
DWC08936W
DWC08937I
DWC08938I
DWC08939I
DWC08940I
DWC08941I
DWC08960I
DWC08961I
DWC08962I
DWC08963I
DWC08964I
DWC08965I
DWC08966E
DWC08967E
DWC08968E
DWC13239E
DWC13300E
DWC13301E
DWC13302E

DWC13304E
DWC13603E
DWC13700E
DWC13701E
DWC13702E
DWC13703E
DWC13705E
DWC13706E
DWC13707E

SQL 메시지

SQL0017N
SQL0056N
SQL0057N
SQL0058N
SQL0097N
SQL0224N
SQL0225N
SQL0227N
SQL0228N
SQL0231W
SQL0243N
SQL0244N
SQL0270N
SQL0301N
SQL0303N
SQL0336N
SQL0348N
SQL0349N
SQL0357N
SQL0358N
SQL0368N
SQL0408N
SQL0423N

SQL0590N
SQL0670N
SQL0845N
SQL0846N
SQL1179W
SQL1186N
SQL1550N
SQL1551N
SQL1552N
SQL1553N
SQL1704N
SQL2077W
SQL2078N
SQL2417N
SQL2426N
SQL2571N
SQL2572N
SQL2573N
SQL2574N
SQL2575N
SQL2576N
SQL4942N
SQL5012N
SQL6583N
SQL20005N
SQL20117N
SQL20121N
SQL20133N
SQL20134N
SQL20135N
SQL20143N
SQL20144N
SQL20145N
SQL20146N

SQL20147N

SQL20148N

SQL20153N

SQL21000N

정정된 SQLSTATES

표 27.

42630	이 문맥에서는 SQLSTATE나 SQLCODE 변수가 유효하지 않습니다.
42631	표현식은 SQL 함수의 RETURN 문에 지정해야 합니다.
42632	SQL 함수나 메소드에는 RETURN 문이 있어야 합니다.
428F2	정수 표현식은 SQL 프로시저의 RETURN 문에 지정해야 합니다.
560B7	복수 행 INSERT의 경우, NEXTVAL 순차 표현식은 각 행에 동일하게 사용해야 합니다.

SQL 참조서

하나의 PDF 파일에 제공되는 SQL 참조서

각 책의 "DB2 라이브러리 사용" 부록은 SQL 참조서가 두 개의 다른 볼륨으로 된 PDF 형식으로 사용할 수 있다고 설명하고 있는데 이는 잘못된 정보입니다.

인쇄된 책이 두 볼륨으로 나타나고 두 개의 해당 양식 번호가 올바르게라도, 하나의 PDF 파일만이 있으며 여기에는 두 볼륨이 모두 들어 있습니다. PDF 파일 이름은 db2s0x70입니다.

언어 요소

이름 지정 규칙 및 내재된 오브젝트 이름 규정

이 절에 다음 주의사항을 추가하십시오.

SQL 프로시저어 구문에서 사용했던 다음 이름들은 이름이 구분되는 경우라도, 일반 식별자에서 허용되는 문자로 제한됩니다.

- condition-name
- label
- parameter-name
- procedure-name
- SQL-variable-name
- statement-name

데이터 링크 할당

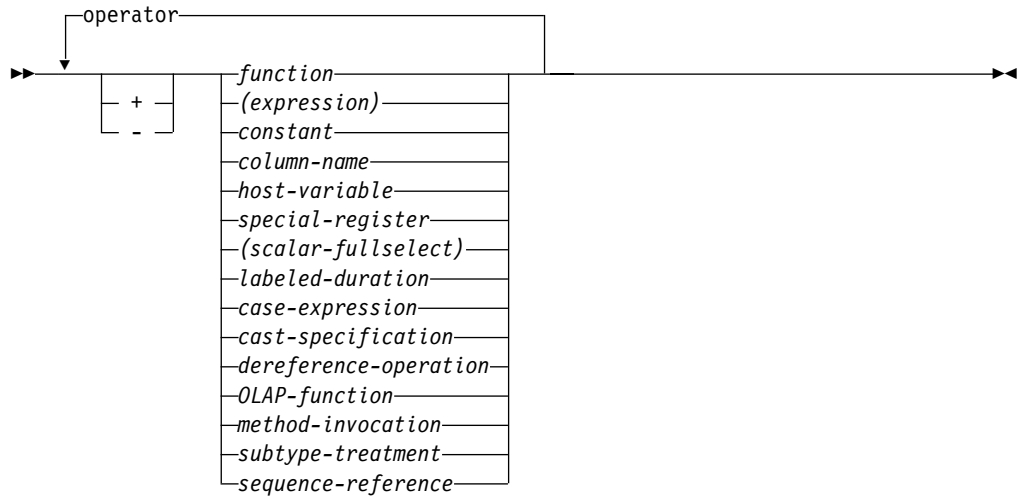
이 절에 있는 단락이 다음과 같이 변경되었습니다.

URL 매개변수 또는 함수 결과는 입력 및 출력 모두에서 동일하며 DATALINK 컬럼의 길이에 의해 바인드됩니다. 그러나, 어떤 경우에 리턴된 URL 값에 액세스 토큰이 접속되어 있습니다. 이것이 가능한 상황에서, 출력 위치는 액세스 토큰과 DATALINK 컬럼의 길이를 위한 충분한 저장 공간을 가져야 합니다. 그러므로, 입력에서 제공된 전체 확장 양식에서 주석 및 URL의 실제 길이는 출력 기억 공간을 수용하도록 제한되어야 합니다. 제한된 길이가 초과되면, 이 오류가 발생합니다.

표현식

구문 도표

구문 도표는 다음과 같이 변경되었습니다.



operator:



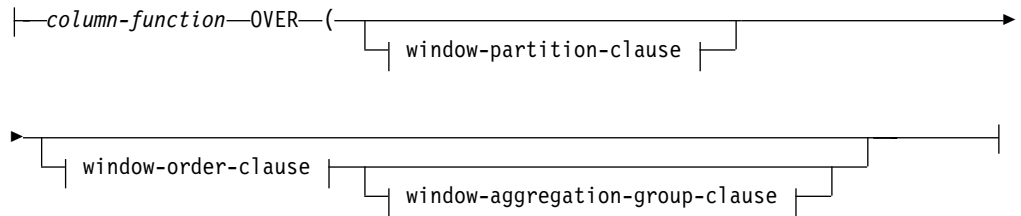
주:

1 ||는 CONCAT에 대한 동의어로서 사용될 수 있습니다.

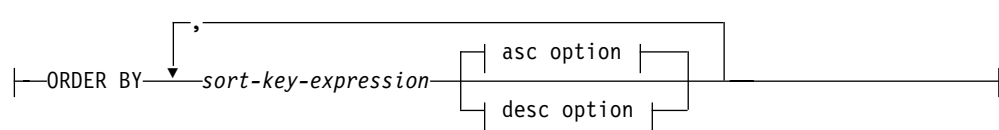
OLAP 함수

다음은 "표현식"에서 "OLAP 함수" 절에 대한 정정을 나타냅니다.

aggregation-function:



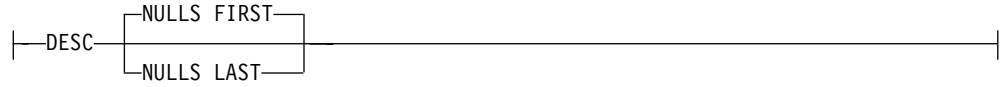
window-order-clause:



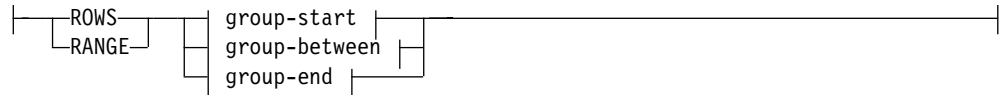
asc option:



desc option:



window-aggregation-group-clause:



group-end:



window-order-clause 설명

NULLS FIRST

창 순서화는 정렬 순서에서 널(NULL)이 아닌 모든 값 이전에 널(NULL) 값을 고려합니다.

NULLS LAST

창 순서화는 정렬 순서에서 널(NULL)이 아닌 모든 값 이후에 널(NULL) 값을 고려합니다.

window-aggregation-group-clause 설명

window-aggregation-group-clause

R 행의 총계 그룹은 R의 파티션 행들의 순서화에서 R에 비교하여 정의된 행 세트입니다. 이 절은 총계 그룹을 지정합니다. 이 절이 지정되지 않은 경우, 기본값은 RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW와 동일하며 누적 총계 결과를 제공합니다.

ROWS

총계 그룹이 행을 계수하여 정의됨을 나타냅니다.

RANGE

총계 그룹이 정렬 키로부터의 오프셋에 의해 정의됨을 나타냅니다.

group-start

총계 그룹의 시작점을 지정합니다. 총계 그룹 끝은 현재 행입니다. *group-start* 절의 스펙은 형식 "BETWEEN *group-start* AND CURRENT ROW"의 *group-between* 절과 동등합니다.

group-between

ROWS 또는 RANGE에 기초하여 총계 그룹 시작과 끝을 지정합니다.

group-end

총계 그룹의 종료점을 지정합니다. 총계 그룹 시작은 현재 행입니다. *group-end* 절의 스펙은 형식 "BETWEEN CURRENT ROW AND *group-end*"의 *group-between* 절과 동등합니다.

UNBOUNDED PRECEDING

현재 행보다 앞에 있는 전체 파티션을 포함시킵니다. ROWS 또는 RANGE를 통해 지정할 수 있습니다. 또한, *window-order-clause*에서 여러 *sort-key-expressions*를 통해 지정할 수도 있습니다.

UNBOUNDED FOLLOWING

현재 행의 뒤에 있는 전체 파티션을 포함시킵니다. ROWS 또는 RANGE를 통해 지정할 수 있습니다. 또한, *window-order-clause*에서 여러 *sort-key-expressions*를 통해 지정할 수도 있습니다.

CURRENT ROW

현재 행에 기초하여 총계 그룹의 시작이나 끝을 지정합니다. ROWS가 지정된 경우, 현재 행은 총계 그룹 경계입니다. RANGE가 지정된 경우, 총계 그룹 경계에는 현재 행으로 *sort-key-expressions*에 대해 동일한 값이 있는 행 세트가 포함됩니다. 이 절은 *group-bound1*이 *value* FOLLOWING을 지정한 경우에 *group-bound2*에 지정할 수 없습니다.

value PRECEDING

현재 행보다 앞에 있는 행의 범위나 수를 지정합니다. ROWS가 지정된 경우, *value*는 행 수를 나타내는 양의 정수입니다. RANGE가 지정된 경우, *value*의 데이터 유형은 *window-order-clause*의 *sort-key-expression* 유형과 비교할만 해야 합니다. *sort-key-expression*은 하나만 있을 수 있으며 *sort-key-expression*의 데이터 유형은 감산을 허용해야 합니다. 이 절은 *group-bound1*이 CURRENT ROW 또는 *value* FOLLOWING인 경우에 *group-bound2*에 지정할 수 없습니다.

value FOLLOWING

현재 행보다 뒤에 있는 행의 범위나 수를 지정합니다. ROWS가 지정된 경우, *value*는 행 수를 나타내는 양의 정수입니다. RANGE가 지정된 경우, *value*의 데이터 유형은 *window-order-clause*의 *sort-key-expression* 유형과 비교할만 해야 합니다. *sort-key-expression*은 하나만 있을 수 있으며 *sort-key-expression*의 데이터 유형은 가산을 허용해야 합니다.

순차 참조

다음 정보는 표현식 섹션의 끝에 추가되어야 합니다("Subtype Treatment" 이후).

sequence-reference:

nextval-expression
prevval-expression

nextval-expression:

NEXTVAL FOR *sequence-name*

prevval-expression:

PREVVAL FOR *sequence-name*

NEXTVAL FOR *sequence-name*

NEXTVAL 표현식은 *sequence-name*에 의해 지정된 순차에 대한 다음 값을 리턴합니다.

PREVVAL FOR *sequence-name*

PREVVAL 표현식은 현재 세션 내에서 이전 명령문의 지정 순차에 대해 가장 최근에 생성된 값을 리턴합니다. 이 값은 순차의 이름을 지정하는 PREVVAL 표현식을 사용하여 반복적으로 참조될 수 있습니다. 단일 명령문 내에서 동일한 순차 이름을 지정하는 PREVVAL 표현식의 여러 인스턴스가 있을 수 있으며 모두 동일한 값을 리턴합니다.

PREVVAL 표현식은 동일한 순차 이름을 지정하는 NEXTVAL 표현식이 현재 사용자 세션에서 이미 참조된 경우에만 사용될 수 있습니다(현재 또는 이전 트랜잭션에서)(SQLSTATE 51035).

주:

- 새 순차 번호는 NEXTVAL 표현식이 순차 이름을 지정할 때 생성됩니다. 그러나, 조회 내에서 동일한 순차 이름을 지정하는 NEXTVAL 표현식의 여러 인스턴스가 있는 경우, 순차의 카운터(counter)는 결과의 각 행에 대해 한번만 증가됩니다.
- 순차에 대해 가장 최근에 생성된 값은 순차의 이름을 지정하는 PREVVAL 표현식을 사용하여 반복적으로 참조될 수 있습니다. 단일 명령문 내에서 동일한 순차 이름을 지정하는 PREVVAL 표현식의 여러 인스턴스가 있을 수 있습니다.
- 아래 표시된 것처럼, 동일한 순차 번호는 첫번째 행에 대한 NEXTVAL 표현식(이것은 순차 값을 생성합니다)과, 다른 행에 대한 PREVVAL 표현식(PREVVAL

의 이 인스턴스는 이전 명령문에서 NEXTVAL 표현식으로 생성된 순차 값을 참조합니다)에서 순차 번호를 참조하여 별도의 두 테이블에서 고유한 키 값으로서 사용될 수 있습니다.

```
INSERT INTO order(orderno, custno)
VALUES (NEXTVAL FOR order_seq, 123456);
```

```
INSERT INTO line_item (orderno, partno, quantity)
VALUES (PREVVAL FOR order_seq, 987654, 1);
```

- NEXTVAL 및 PREVVAL 표현식이 지정될 수 있는 예는 다음과 같습니다.
 - select-statement 또는 SELECT INTO 문: 명령문이 DISTINCT 키워드, GROUP BY 절, ORDER BY 절, UNION 키워드, INTERSECT 키워드 또는 EXCEPT 키워드를 포함하지 않는 한 select-clause 내에서
 - INSERT 명령문: VALUES 절
 - INSERT 명령문: fullselect의 select-clause 내에서
 - UPDATE 문: SET 절에 있는 표현식 fullselect의 select-clause 내에서 (검색되거나 위치 지정된 UPDATE 문 중 하나)
 - VALUES INTO 문: 표현식 fullselect의 select-clause 내에서
- NEXTVAL 및 PREVVAL 표현식이 지정될 수 없는 예(SQLSTATE 428F9)는 다음과 같습니다.
 - 완전 외부 조인의 조인 조건
 - CREATE TABLE 또는 ALTER TABLE 문에서 컬럼에 대한 DEFAULT 값
 - CREATE TABLE 또는 ALTER TABLE 문에서 생성된 컬럼 정의
 - CHECK 제한조건의 조건
 - CREATE TRIGGER 문
 - CREATE VIEW 문
 - CREATE METHOD 문
 - CREATE FUNCTION 문
- 그리고, NEXTVAL 표현식은 다음에서 지정될 수 없습니다(SQLSTATE 428F9).
 - CASE 표현식
 - 총계 함수의 매개변수 목록
 - 부속 조회
 - DISTINCT 연산자가 들어 있는 SELECT 문
 - 조인의 조인 조건
 - SELECT 문의 GROUP BY 절
 - UNION, INTERSECT 또는 EXCEPT 집합 연산자를 사용하는 또다른 SELECT 문과 조합된 SELECT 문

- 중첩 테이블 표현식
- 테이블 함수의 매개변수 목록
- SELECT, DELETE 또는 UPDATE 문의 WHERE 절
- ORDER BY 절
- CALL 문의 매개변수 목록

- 값이 순차에 대해 생성되면, 해당 값이 소모되며, 값이 필요한 다음번에 새 값이 생성됩니다. 이것은 NEXTVAL 표현식이 들어 있는 명령문이 실패할 때에도 참입니다.
- INSERT 명령문이 컬럼에 대한 VALUES 목록에 NEXTVAL 표현식을 포함하는 경우, INSERT의 실행 중 어떤 지점에서 어떤 오류가 발생하는 경우(다음 순차 값 생성시 문제점이 되거나, 또다른 컬럼에 대한 값에서 문제점이 될 수 있습니다), 삽입 실패가 발생하며, 순차에 대해 생성된 값은 소비된 것으로 간주됩니다. 어떤 경우에, 동일한 INSERT 명령문을 재발행하면 성공할 수도 있습니다.

예를 들어, NEXTVAL이 사용된 컬럼에 대한 고유한 색인의 존재의 결과로 오류가 발생하고 생성된 순차 값이 이미 색인에 존재하는 경우가 있습니다. 순차에 대해 생성된 다음 값이 색인에 존재하지 않는 값이므로 차후 INSERT가 성공할 수 있습니다.

- 순차에 대한 값 생성에서 순차에 대한 최대 값이 초과되고(또는 내림차순 순차에 대한 최소 값) 순환이 허용되지 않으면, 오류가 발생합니다(SQLSTATE 23522). 이 경우, 사용자는 순차를 ALTER하여 채택 가능한 값의 범위를 확장하거나, 순차의 순환을 사용 가능하게 하거나, 더 큰 범위의 값을 가진 다른 데이터 유형을 가진 새 순차를 DROP 및 CREATE할 수 있습니다.

예를 들어, SMALLINT의 데이터 유형으로 순차가 정의되어 있고, 순차가 지정 가능한 값을 벗어나 수행합니다. INTEGER로서 순차를 재정의하려면, 새 정의로 순차를 삭제 및 재작성하는 것이 필요합니다.

- 커서의 SELECT 문에 있는 PREVVAL에 대한 참조는 커서를 열기 전에 지정된 순차에 대해 생성된 값을 참조합니다. 그러나 커서를 닫으면 그 이후의 명령문에서는 지정된 순차에 대해 PREVVAL이 리턴하는 값에 영향을 줄 수 있으며, 커서를 다시 연 이벤트의 동일한 명령문에서도 마찬가지입니다. 이는 커서의 SELECT 문에 동일한 순차 이름에 대한 NEXTVAL에 대한 참조가 포함되었을 때 발생합니다.

예: 이 예에서는 "order"라고 하는 테이블이 있으며 "order_seq"라고 하는 순차가 다음과 같이 작성됨을 가정합니다.

```
CREATE SEQUENCE order_seq
START WITH 1
INCREMENT BY 1
NOMAXVALUE
NOCYCLE
CACHE 24
```

- 위에서 작성된 순서에 대한 NEXTVAL 표현식으로 "order_seq" 순차 번호를 생성하는 방법에 대한 몇가지 예는 다음과 같습니다.

```
INSERT INTO order(orderno, custno)
VALUES (NEXTVAL FOR order_seq, 123456);
```

또는,

```
UPDATE order
SET orderno = NEXTVAL FOR order_seq
WHERE custno = 123456;
```

또는,

```
VALUES NEXTVAL FOR order_seq INTO :hv_seq;
```

함수

새로운 함수 및 프로시저어 사용

버전 7 FixPaks는 새로운 SQL 내장 스칼라 함수를 제공합니다. 이들 새 함수의 설명은 SQL 참조서 갱신내용을 참조하십시오. 새 함수는 데이터베이스 서버 코드가 새 서비스 레벨로 업그레이드될 때 각 데이터베이스에서 자동으로 사용 가능하지 않습니다. 이들 새 함수를 사용하려면, 시스템 관리자가 서버에서 각 데이터베이스를 지정하여 **db2updv7** 명령을 발행해야 합니다. 이 명령은 이 명령 실행 이전에 작성된 데이터베이스 오브젝트가 새로운 함수 서명과 일치하는 기존 함수 서명을 사용하도록 보장하는 항목을 데이터베이스에 작성합니다.

MQSeries 함수(MQDB2 스키마에 정의된 함수) 사용에 대한 자세한 정보는 195 페이지의 『MQSeries』를 참조하십시오.

스칼라 함수

ABS 또는 ABSVAL

▶ ABS (expression) ▶
 └─ ABSVAL ─┘

스키마는 SYSIBM입니다.

이 함수는 버전 7.1의 FixPak 2에서 처음 사용 가능합니다.

주: ABS(또는 ABSVAL) 함수의 SYSFUN 버전을 계속 사용할 수 있습니다.

인수의 절대값을 리턴합니다.

인수는 내장 숫자 데이터 유형의 값을 리턴하는 표현식입니다.

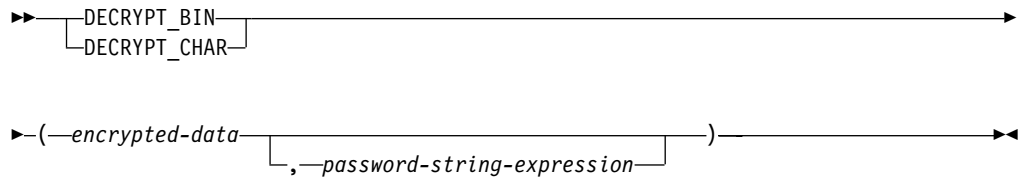
함수의 결과에 인수와 동일한 데이터 유형 및 길이 속성이 있습니다. 인수가 널(NULL)이 될 수 있거나 데이터베이스가 DFT_SQLMATHWARN을 yes로 설정하여 구성되어 있는 경우, 결과는 널(NULL)이 될 수 있습니다. 인수가 널(NULL)이면, 결과는 널(NULL) 값입니다.

예를 들어, 다음과 같습니다.

```
ABS(-51234)
```

값이 51234인 INTEGER를 리턴합니다.

DECRYPT_BIN 및 DECRYPT_CHAR



스키마는 SYSIBM입니다.

이 함수는 버전 7.1의 FixPak 3에서 처음 사용 가능합니다.

DECRYPT_BIN 및 DECRYPT_CHAR 함수는 *encrypted-data*를 암호해독한 결과 값을 리턴합니다. 암호해독에 사용되는 암호는 *password-string-expression* 값 또는 ENCRYPTION PASSWORD 값 중 하나입니다(SET ENCRYPTION PASSWORD 문을 사용하여 지정). DECRYPT_BIN 및 DECRYPT_CHAR 함수는 ENCRYPT 함수를 사용하여 암호화된 값만을 암호해독할 수 있습니다(SQLSTATE 428FE).

encrypted-data

ENCRYPT 함수를 사용한, 완전히 암호화된 데이터 문자열인 CHAR FOR BIT DATA 또는 VARCHAR FOR BITDATA 값을 리턴하는 표현식.

password-string-expression

최소한 6 바이트이고 127 바이트를 초과하지 않는 CHAR 또는 VARCHAR 값을 리턴하는 표현식(SQLSTATE 428FC). 이것은 데이터를 암호화하는 데 사용된 것과 동일한 암호이어야 합니다. 그렇지 않으면 암호해독이 오류를 초래합니다(SQLSTATE 428FD). 암호 인수의 값이 널(NULL)이거나 제공되지 않으면, 데이터는 세션에 대해 설정되어야 하는 ENCRYPTION PASSWORD 값을 사용하여 암호해독됩니다(SQLSTATE 51039).

DECRYPT_BIN 함수의 결과는 VARCHAR FOR BIT DATA입니다. DECRYPT_CHAR 함수의 결과는 VARCHAR입니다. *encrypted-data*가 힌트를 포함한 경우, 힌트는 함수에 의해 리턴되지 않습니다. 결과의 길이 속성은 *encrypted-data* 데이터 유형의 길이 속성에서 8 바이트를 뺀 것입니다. 함수에서 리턴한 값의 실제 길이는 암호화된 원래 문자열의 길이와 일치합니다. *encrypted-data*가 암호화된 문자열 이외의 바이트를 포함하는 경우, 이 바이트는 함수에 의해 리턴되지 않습니다. 첫번째 인수가 널(NULL)일 수 있는 경우, 결과가 널(NULL)이 될 수 있으며, 첫번째 인수가 널(NULL)인 경우 결과는 널(NULL) 값입니다.

암호화가 발생한 코드 페이지 이외의 코드 페이지를 사용하는 다른 시스템에서 데이터가 암호해독되는 경우, 암호해독된 값을 데이터베이스 코드 페이지로 변환할 때 확장이 일어날 수 있습니다. 그러한 상황에서, *encrypted-data* 값은 많은 수의 바이트로 VARCHAR 문자열로 유형변환(cast)되어야 합니다.

또한 이 함수 사용에 대한 자세한 정보는 『ENCRYPT』 및 415 페이지의 『GETHINT』를 참조하십시오

예:

예 1: 이 예는 ENCRYPTION PASSWORD 값을 사용하여 암호화 암호를 보유합니다.

```
SET ENCRYPTION PASSWORD = 'Ben123';
INSERT INTO EMP (SSN) VALUES ENCRYPT('289-46-8832');
SELECT DECRYPT_CHAR(SSN)
FROM SSN;
```

리턴된 값은 '289-46-8832'입니다.

예 2: 이 예는 명시적으로 암호화 암호를 전달합니다.

```
SELECT DECRYPT_CHAR(SSN, 'Ben123')
FROM SSN;
```

리턴된 값은 '289-46-8832'입니다.

ENCRYPT

```

▶▶—ENCRYPT—————▶
▶—(—data-string-expression—, —password-string-expression—, —hint-string-expression—)—▶▶

```

스키마는 SYSIBM입니다.

이 함수는 버전 7.1의 FixPak 3에서 처음 사용 가능합니다.

ENCRYPT 함수는 *data-string-expression* 암호화의 결과 값을 리턴합니다. 암호화에 사용되는 암호는 *password-string-expression* 값 또는 ENCRYPTION PASSWORD 값 중 하나입니다(SET ENCRYPTION PASSWORD 문을 사용하여 지정).

data-string-expression

암호화되는 CHAR 또는 VARCHAR 값을 리턴하는 표현식. *data-string-expression* 의 데이터 유형에 대한 길이 속성은 *hint-string-expression* 인수가 없으면 32663 으로 제한되며 *hint-string-expression* 인수가 지정될 때는 32631로 제한됩니다 (SQLSTATE 42815).

password-string-expression

최소한 6 바이트이고 127 바이트를 초과하지 않는 CHAR 또는 VARCHAR 값을 리턴하는 표현식(SQLSTATE 428FC). 값은 *data-string-expression* 을 암호화하는 데 사용되는 암호를 나타냅니다. 암호 인수의 값이 널(NULL)이거나 제공되지 않으면, 데이터는 세션에 대해 설정되어야 하는 ENCRYPTION PASSWORD 값을 사용하여 암호화됩니다(SQLSTATE 51039).

hint-string-expression

데이터 소유자가 암호를 기억하도록 돕는 최대 32 바이트까지의 CHAR 또는 VARCHAR 값을 리턴하는 표현식(예: 'Pacific'을 기억하도록 돕는 힌트로서 'Ocean'). 힌트 값이 제공되면, 힌트는 결과에 추가되며 GETHINT 함수를 사용하여 검색될 수 있습니다. 이 인수가 널(NULL)이거나 제공되지 않으면, 결과에 힌트가 추가되지 않습니다.

함수의 결과 데이터 유형은 VARCHAR FOR BIT DATA입니다.

결과의 길이 속성은 다음과 같습니다.

- 선택적 힌트 매개변수가 지정되면, 암호화되지 않은 데이터 + 8 바이트 + 다음 8 바이트 경계까지의 바이트 수 + 힌트 길이 32 바이트의 길이 속성.
- 힌트 매개변수가 없는 경우, 암호화되지 않은 데이터 + 8 바이트 + 다음 8 바이트 경계까지의 바이트 수의 길이 속성.

첫번째 인수가 널(NULL)일 수 있는 경우, 결과가 널(NULL)이 될 수 있으며, 첫번째 인수가 널(NULL)인 경우 결과는 널(NULL) 값입니다.

암호화된 결과가 *data-string-expression* 값보다 깁니다. 그러므로, 암호화된 값을 지정할 때, 전체 암호화 값을 포함하는 충분한 크기로 목표가 선언되었는지 확인하십시오.

참고:

- **암호화 알고리즘:** 사용된 내부 암호화 알고리즘은 패딩(padding)이 있는 RC2 블록 암호이며, 128 비트 비밀 키는 MD2 메시지 요약을 사용하여 암호에서 추출됩니다.
- **암호화 암호 및 데이터:** 암호 관리를 수행하는 것은 사용자 책임입니다. 데이터가 암호화되면 데이터를 암호화하기 위해 사용된 암호만이 암호해독에 사용될 수 있습니다.

다(SQLSTATE 428FD). 암호 값을 설정하기 위해 공백으로 채워질 수 있는 CHAR 문자를 사용할 때는 주의하십시오. 암호화된 결과에는 널(NULL) 종료자와 다른 인 쇄가능하지 않은 문자가 포함될 수 있습니다.

- **테이블 컬럼 정의:** 암호화된 데이터를 포함하기 위해 컬럼 및 유형을 정의할 때 항상 다음과 같은 길이 속성을 계산합니다. 힌트 없는 암호화된 데이터의 경우:

암호화되지 않은 데이터의 최대 길이 + 8 바이트 + 다음 8 바이트 경계까지의 바이트 수 = 암호화된 데이터 컬럼 길이.

추가된 힌트가 있는 암호화된 데이터의 경우:

암호화되지 않은 데이터의 최대 길이 + 8 바이트 + 다음 8 바이트 경계까지의 바이트 수 + 힌트 길이의 32 바이트 = 암호화된 데이터 컬럼 길이.

제시된 데이터 길이보다 짧은 길이로의 지정이나 유형변환(cast)은 나중에 암호해독 실패 및 데이터 유실의 결과를 초래할 수 있습니다. 공백은 너무 짧은 컬럼에 저장 될 때 절단될 수 있는 유효한 암호화 데이터 값입니다.

샘플 컬럼 길이 계산

암호화되지 않은 데이터의 최대 길이	6 바이트
8 바이트	8 바이트
다음 8 바이트 경계까지의 바이트 수	2 바이트

암호화 데이터 컬럼 길이	16 바이트
암호화되지 않은 데이터의 최대 길이	32 바이트
8 바이트	8 바이트
다음 8 바이트 경계까지의 바이트 수	8 바이트

암호화 데이터 컬럼 길이	48 바이트

- **암호화된 데이터의 관리:** 암호화 데이터는 ENCRYPT 함수에 해당하는 암호해독 함수를 지원하는 서버에서만 암호해독될 수 있습니다. 그러므로, 암호화된 데이터를 가진 컬럼 복제는 DECRYPT_BIN 또는 DECRYPT_CHAR 함수를 지원하는 서버에만 수행되어야 합니다.

또한 이 함수 사용에 대한 자세한 정보는 411 페이지의 『DECRYPT_BIN 및 DECRYPT_CHAR』 및 415 페이지의 『GETHINT』를 참조하십시오.

예:

예 1: 이 예는 ENCRYPTION PASSWORD 값을 사용하여 암호화 암호를 보유합니다.

```
SET ENCRYPTION PASSWORD = 'Ben123';
INSERT INTO EMP (SSN) VALUES ENCRYPT('289-46-8832');
```

예 2: 이 예는 명시적으로 암호화 암호를 전달합니다.

```
INSERT INTO EMP (SSN) VALUES ENCRYPT('289-46-8832', 'Ben123', '');
```

예 3: 힌트 'Ocean'은 암호화 암호 'Pacific'을 기억하도록 돕기 위해 저장됩니다.

```
INSERT INTO EMP (SSN) VALUES ENCRYPT('289-46-8832','Pacific','Ocean');
```

GETHINT

▶▶—GETHINT—(—*encrypted-data*—)—————▶▶

스키마는 SYSIBM입니다.

이 함수는 버전 7.1의 FixPak 3에서 처음 사용 가능합니다.

GETHINT 함수는 *encrypted-data*에서 발견된 경우 암호 힌트를 리턴합니다. 암호 힌트는 데이터 소유자가 암호를 기억하도록 돕는 구입니다(예: 'Pacific'을 기억하도록 돕는 힌트로서 'Ocean').

encrypted-data

ENCRYPT 함수를 사용한, 완전히 암호화된 데이터 문자열인 CHAR FOR BIT DATA 또는 VARCHAR FOR BITDATA 값을 리턴하는 표현식(SQLSTATE 428FE).

함수의 결과는 VARCHAR(32)입니다. 결과는 널(NULL)일 수 있습니다. 힌트 매개변수가 ENCRYPT 함수에 의해 *encrypted-data*에 추가되지 않았거나 첫번째 인수가 널(NULL)인 경우, 결과는 널(NULL) 값입니다.

또한 이 함수 사용에 대한 자세한 정보는 411 페이지의 『DECRYPT_BIN 및 DECRYPT_CHAR』 및 412 페이지의 『ENCRYPT』를 참조하십시오.

예:

이 예에서 힌트 'Ocean'은 사용자가 암호화 암호 'Pacific'을 기억하도록 돕기 위해 저장됩니다.

```
INSERT INTO EMP (SSN) VALUES ENCRYPT('289-46-8832', 'Pacific','Ocean');
SELECT GETHINT(SSN)
FROM EMP;
```

리턴된 값은 'Ocean'입니다.

IDENTITY_VAL_LOCAL

▶▶—IDENTITY_VAL_LOCAL—(—)—————▶▶

스키마는 SYSIBM입니다.

이 프로시저는 버전 7.1의 FixPak 3에서 처음 사용 가능합니다.

IDENTITY_VAL_LOCAL 함수는 식별 컬럼에 가장 최근에 할당된 값을 리턴하는 비결정적인 함수이며 여기서 할당은 VALUES 절을 사용하는 단일 행 INSERT 문의 결과로서 발생합니다. 이 함수에는 입력 매개변수가 없습니다.

해당 식별 컬럼의 실제 데이터 유형과 상관없이 결과는 DECIMAL(31,0)입니다.

함수에서 리턴되는 값은 최신 단일 행 INSERT 명령문에 식별된 테이블의 식별 컬럼에 지정된 값입니다. INSERT 문은 식별 컬럼이 들어 있는 테이블에서 VALUES 절을 사용하여 작성되어야 합니다. 또한, INSERT 명령문은 동일한 레벨로 발행해야 합니다.¹ 즉, 값은 다음에 지정된 값으로 바뀔 때까지 지정된 레벨로 지역적으로 사용 가능합니다.

지정된 값은 사용자가 제공한 값이거나(식별 컬럼이 GENERATED BY DEFAULT로 정의된 경우) DB2가 생성한 식별 값일 수 있습니다.

함수는 다음 상황에서 널(NULL) 값을 리턴합니다.

- 식별 컬럼이 들어 있는 테이블에 대해 현재 처리 레벨에서 VALUES 절이 있는 단일 행 INSERT 명령문이 발행되지 않은 경우.
- 값을 지정한 최신 INSERT 명령문 이후로 작업 단위(UOW)의 COMMIT 또는 ROLLBACK이 발생한 경우².

함수의 결과는 다음 명령문에 의해 영향을 받지 않습니다.

- 식별 컬럼이 없는 테이블에 대해 VALUES 절이 있는 단일 행 INSERT 명령문.
- VALUES 절이 있는 복수 행 INSERT 명령문.
- fullselect가 있는 INSERT 명령문.
- ROLLBACK TO SAVEPOINT 명령문.

참고:

- INSERT 명령문의 VALUES 절에 있는 표현식은 INSERT 명령문의 목표 컬럼에 대한 지정 이전에 평가됩니다. 따라서, INSERT 명령문의 VALUES 절에서 IDENTITY_VAL_LOCAL 함수의 호출은 이전 INSERT 문에 대해 식별 컬럼에 가장 최근에 할당된 값을 사용합니다. 식별 컬럼이 들어 있는 테이블에 대한 VALUES 절이 있는 이전 단일 행 INSERT 명령문이 IDENTITY_VAL_LOCAL 함수와 동일한 레벨 내에서 실행된 경우, 함수는 널(NULL) 값을 리턴합니다.
- 트리거가 정의된 테이블의 식별 컬럼 값은 식별 컬럼에 대한 트리거 전이 변수를 참조하여 트리거 내에서 판별할 수 있습니다.

1. 새로운 레벨은 트리거, 함수 또는 저장 프로시저가 호출될 때마다 시작됩니다.

2. 자동 확약이 해제되지 않은 경우, 각 명령문에 함수가 호출되면 각 명령문 이후에 자동으로 확약하는 인터페이스는 널(NULL) 값을 리턴합니다.

- 삽입 트리거의 트리거 조건 내에서 IDENTITY_VAL_LOCAL 함수를 호출하는 결과는 널(NULL) 값입니다.
- 여러 이전 또는 이후 삽입 트리거가 테이블에 대해 존재하는 것이 가능합니다. 이 경우 각 트리거는 별도로 처리되며, 하나의 트리거 조치가 할당한 식별 값은 IDENTITY_VAL_LOCAL 함수를 사용하는 다른 트리거 조치에 대해 사용 불가능합니다. 여러 트리거 조치가 개념적으로 동일 레벨에서 정의되지만 이것은 참입니다.
- 이전 삽입 트리거의 본문에서 IDENTITY_VAL_LOCAL 함수를 사용하는 것은 일반적으로 권장되지는 않습니다. 이전 삽입 트리거의 트리거 조치 내에서 IDENTITY_VAL_LOCAL 함수를 호출하는 결과는 널(NULL) 값입니다. 트리거가 정의된 테이블의 식별 컬럼에 대한 값은 이전 삽입 트리거의 트리거 조치 내에서 IDENTITY_VAL_LOCAL 함수를 호출하여 획득될 수 없습니다. 그러나, 식별 컬럼에 대한 값은 식별 컬럼에 대한 트리거 전이 변수를 참조하여 트리거 조치에서 구할 수 있습니다.
- 이후 삽입 트리거의 트리거 조치 내에서 IDENTITY_VAL_LOCAL 함수를 호출하는 결과는 3 식별 컬럼이 들어 있는 테이블의 VALUES 절이 있는 최신 단일 행 INSERT 명령문에 식별된 테이블의 식별 컬럼에 지정된 값입니다. IDENTITY_VAL_LOCAL 함수의 호출 이전에, 식별 컬럼이 들어 있는 테이블에 대한 VALUES 절이 있는 단일 행 INSERT 명령문이 동일한 트리거된 조치 내에서 실행되지 않은 경우, 함수는 널(NULL) 값을 리턴합니다.
- IDENTITY_VAL_LOCAL 함수의 결과가 결정적이지 아니므로, 커서의 SELECT 문 내에서 IDENTITY_VAL_LOCAL 함수의 호출 결과는 각 FETCH 문에 대해 다를 수 있습니다.
- 지정된 값은 식별 컬럼에 실제로 지정된 값 즉, 후속 SELECT 문에 리턴되는 값입니다. 이 값은 반드시 INSERT 명령문의 VALUES 절에 제공된 값이거나 DB2에 의해 생성된 값일 필요는 없습니다. 지정된 값은 식별 컬럼과 연관된 트리거 전이 변수에 대해 이전 삽입 트리거 본문 내의 SET 전이 변수 명령문에 지정된 값일 수 있습니다.
- 함수에 의해 리턴되는 값은 실패한 VALUES 절이 있는 단일 행 INSERT를 따라 식별 컬럼이 있는 테이블로 가면서 예상할 수 없습니다. 이 값은 실패한 INSERT 이전에 호출한 함수에서 리턴된 값이거나 성공된 INSERT에서 지정된 값일 수 있습니다. 리턴되는 실제 값은 실패 지점에 따라 다르므로 예상할 수 없습니다.

예:

예 1: IVAR 변수를 EMPLOYEE 테이블의 식별 컬럼에 지정된 값으로 설정하십시오. 이 삽입이 EMPLOYEE 테이블에서 첫번째이면, IVAR의 값은 1이 됩니다.

3. 이것은 FOR EACH ROW 및 FOR EACH STATEMENT 이후 삽입 트리거 둘다에 적용됩니다.

```

CREATE TABLE EMPLOYEE
(EMPNO INTEGER GENERATED ALWAYS AS IDENTITY,
 NAME CHAR(30),
 SALARY DECIMAL(5,2),
 DEPTNO SMALLINT)

```

예 2: INSERT 명령문에서 호출된 IDENTITY_VAL_LOCAL 함수는 이전 단일 행 INSERT 문과 연관된 값을 식별 컬럼이 있는 테이블에 대한 VALUES 절과 함께 리턴합니다. 이 예에서는 T1과 T2의 두 테이블이 있다고 가정합니다. T1과 T2에는 모두 이름이 C1로 지정된 식별 컬럼이 있습니다. DB2는 테이블 T1의 C1 컬럼에 대해 1로 시작하여 순서대로 값을 생성하고 테이블 T2의 C1 컬럼에 대해 10으로 시작하여 순서대로 값을 생성합니다.

```

CREATE TABLE T1
(C1 INTEGER GENERATED ALWAYS AS IDENTITY,
 C2 INTEGER),
CREATE TABLE T2
(C1 DECIMAL(15,0) GENERATED BY DEFAULT AS IDENTITY
 (START WITH 10),
 C2 INTEGER),
INSERT INTO T1 (C2) VALUES (5),
INSERT INTO T1 (C2) VALUES (6),
SELECT * FROM T1

```

다음 결과를 제공합니다.

C1	C2
1	5
2	6

이제, IVAR 변수에 대한 함수를 선언합니다.

```
VALUES IDENTITY_VAL_LOCAL() INTO :IVAR
```

이 때, IDENTITY_VAL_LOCAL 함수는 IVAR 에 값 2를 리턴하는데, 이 값이 DB2 가 할당한 가장 최근의 값이기 때문입니다. 다음 INSERT 문은 단일 행을 T2에 삽입 하며 여기서 컬럼 C2는 IDENTITY_VAL_LOCAL 함수로부터 값 2를 얻습니다.

```

INSERT INTO T2 (C2) VALUES (IDENTITY_VAL_LOCAL());
SELECT * FROM T2
WHERE C1 = DECIMAL(IDENTITY_VAL_LOCAL(),15,0)

```

다음 결과를 리턴합니다.

C1	C2
10.	2

이 삽입 후에 IDENTITY_VAL_LOCAL 함수를 호출하면, 값 10이 작성되며 이는 DB2가 T2의 컬럼 C1에 대해 생성한 값입니다.

더 낮은 레벨에서 지정된 식별이 있을지라도, 트리거를 포함하는 중첩 환경에서 IDENTITY_VAL_LOCAL 함수를 사용하여 특정 레벨에서 지정된 식별 값을 검색하십시오. EMPLOYEE, EMP_ACT 및 ACCT_LOG 테이블이 있다고 가정하십시오. EMP_ACT 및 ACCT_LOG 테이블에 추가적인 삽입 결과를 가져오는 EMPLOYEE에 정의된 이후 삽입 트리거가 있습니다.

```

CREATE TABLE EMPLOYEE
  (EMPNO SMALLINT GENERATED ALWAYS AS IDENTITY (START WITH 1000),
   NAME CHAR(30),
   SALARY DECIMAL(5,2),
   DEPTNO SMALLINT);
CREATE TABLE EMP_ACT
  (ACNT_NUM SMALLINT GENERATED ALWAYS AS IDENTITY (START WITH 1),
   EMPNO SMALLINT);
CREATE TABLE ACCT_LOG
  (ID SMALLINT GENERATED ALWAYS AS IDENTITY (START WITH 100),
   ACNT_NUM SMALLINT,
   EMPNO SMALLINT);
CREATE TRIGGER NEW_HIRE
  AFTER INSERT ON EMPLOYEE
  REFERENCING NEW AS NEW_EMP
  FOR EACH ROW MODE DB2SQL
  BEGIN ATOMIC
    INSERT INTO EMP_ACT (EMPNO)
    VALUES (NEW_EMP.EMPNO);
    INSERT INTO ACCT_LOG (ACNT_NUM EMPNO)
    VALUES (IDENTITY_VAL_LOCAL(), NEW_EMP.EMPNO);
  END

```

첫번째 트리거된 INSERT 명령문은 행을 EMP_ACT 테이블에 삽입합니다. 이 INSERT 명령문은 EMPLOYEE 테이블의 EMPNO 컬럼에 대한 트리거 전이 변수를 사용하여, EMPLOYEE 테이블의 EMPNO 컬럼에 대한 식별 값이 EMP_ACT 테이블의 EMPNO 컬럼에 복사됨을 나타냅니다. IDENTITY_VAL_LOCAL 함수는 EMPLOYEE 테이블의 EMPNO 컬럼에 지정된 값을 획득하기 위해 사용될 수 없습니다. 이것은 INSERT 명령문이 이 중첩 레벨에서 발행되지 않았기 때문이며, 그와 같이 IDENTITY_VAL_LOCAL 함수가 EMP_ACT에 대한 INSERT의 VALUES 절에서 호출된 경우, 널(NULL) 값을 리턴합니다. EMP_ACT 테이블에 대한 INSERT 명령문도 ACNT_NUM 컬럼에 대한 새 식별 컬럼 값의 생성 결과를 가져옵니다.

두 번째 트리거된 INSERT 명령문은 행을 ACCT_LOG 테이블에 삽입합니다. 이 명령문은 IDENTITY_VAL_LOCAL 함수를 호출하여, 트리거된 조치에서 이전 INSERT 명령문에서 EMP_ACT 테이블의 ACNT_NUM 컬럼에 지정된 식별 값이 ACCT_LOG 테이블의 ACNT_NUM 컬럼에 복사됨을 나타냅니다. EMPNO 컬럼에는 EMPLOYEE 테이블의 EMPNO 컬럼과 동일한 값이 지정됩니다.

호출하는 응용프로그램에서(즉, EMPLOYEE에 대한 INSERT가 발행된 레벨), 원래 INSERT 명령문에 의해 EMPLOYEE 테이블의 EMPNO 컬럼에 지정된 값으로 IVAR 변수를 설정하십시오.

```
INSERT INTO EMPLOYEE (NAME, SALARY, DEPTNO)
VALUES ('Rupert', 989.99, 50);
```

원래의 INSERT 명령문을 처리한 후에 세 개의 테이블의 내용과 모든 트리거된 조치는 다음과 같습니다.

```
SELECT EMPNO, SUBSTR(NAME,10) AS NAME, SALARY, DEPTNO
FROM EMPLOYEE;
EMPNO      NAME              SALARY              DEPTNO
-----
      1000 Rupert              989.99              50

SELECT ACNT_NUM, EMPNO
FROM EMP_ACT;
ACNT_NUM    EMPNO
-----
          1      1000

SELECT * FROM ACCT_LOG;
ID          ACNT_NUM    EMPNO
-----
          100          1      1000
```

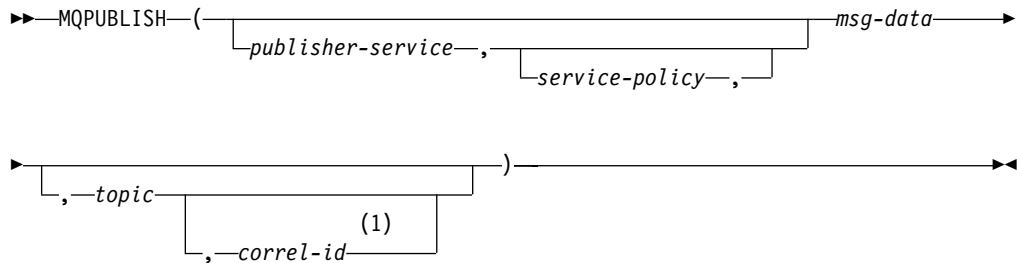
IDENTITY_VAL_LOCAL 함수의 결과는 동일한 중첩 레벨에서 식별 컬럼에 대한 최신 지정 값입니다. 원래의 INSERT 명령문과 모든 트리거된 조치를 처리한 후에, IDENTITY_VAL_LOCAL 함수는 EMPLOYEE 테이블의 EMPNO 컬럼에 지정된 값인 1000 값을 리턴합니다. 다음과 같은 VALUES 문은 IVAR을 1000으로 설정합니다. EMP_ACT 테이블로의 삽입(EMPLOYEE 테이블로의 삽입 이후에 더 낮은 중첩 레벨에서 발생)은 이 IDENTITY_VAL_LOCAL 함수의 호출로 리턴된 사항에 영향을 미치지 않습니다.

```
VALUES IDENTITY_VAL_LOCAL() INTO :IVAR;
```

LCASE 및 UCASE(유니코드)

유니코드 데이터베이스에서, 유니코드 문자의 전체 레퍼터리는 이들 문자의 유니코드 등 록정보에 기초한 대문자(또는 소문자)입니다. 로마 숫자와 함께, ASCII 문자의 이중 폭 버전 역시 이제 올바른 대문자 또는 소문자로 변환됩니다.

MQPUBLISH



주:

1 service와 policy를 이전에 정의하지 않은 경우, correl-id를 지정할 수 없습니다.

스키마는 MQDB2입니다.

MQPUBLISH 함수는 데이터를 MQSeries에 발행합니다. 이 함수를 사용하려면 MQSeries Publish/Subscribe나 MQSeries Integrator 중 하나를 설치해야 합니다. 자세한 정보는 www.ibm.com/software/MQSeries를 참조하십시오.

MQPUBLISH 함수는 *service-policy*가 정의한 서비스 규정 품질을 사용하여 *msg-data*에 포함된 데이터를 *publisher-service*에 지정된 MQSeries 발행자에 발행합니다. 선택적으로 메시지에 주제를 지정할 수 있고, 사용자 정의 메시지 상관 ID를 지정할 수도 있습니다. 이 함수는 성공할 경우 "1"값을 리턴하고, 성공하지 못할 경우 "0"값을 리턴합니다.

publisher-service

메시지가 송신될 논리적 MQSeries 목적지가 포함된 문자열. 이 옵션을 지정하면 *publisher-service*가 AMT.XML 저장소 파일에 정의된 publisher Service Point를 참조해야 합니다. 서비스 지점은 메시지가 송신되거나 수신되는 논리적 끝점입니다. 서비스 지점 정의에는 MQSeries 대기행렬 관리자 이름 및 대기행렬 이름이 포함됩니다. 자세한 정보는 MQSeries Application Messaging Interface를 참조하십시오. *publisher-service*가 지정되어 있지 않으면 DB2.DEFAULT.PUBLISHER가 사용됩니다. *publisher-service*의 최대 크기는 48문자입니다.

service-policy

이 메시지를 처리할 때 사용할 MQSeries AMI Service Policy가 포함된 문자열. 이 옵션을 지정하면 *service-policy*가 AMT.XML 저장소 파일에 정의된 규정을 참조해야 합니다. 서비스 규정은 이 메시징 작업에 적용되어야 할 서비스 옵션의 품질 세트를 정의합니다. 이 옵션에는 메시지 우선순위와 메시지 지속성이 포함됩니다. 자세한 정보는 MQSeries Application Messaging Interface를 참조하십시오. *service-policy*가 지정되어 있지 않으면 기본 DB2.DEFAULT.POLICY가 사용됩니다. *service-policy*의 최대 크기는 48문자입니다.

msg-data

MQSeries를 통해 전송할 데이터가 포함된 문자열 표현식. 최대 크기는 4000문자입니다.

topic

메시지 발행에 대한 주제가 포함된 문자열 표현식. 주제가 지정되어 있지 않으면 이 메시지에 주제가 연관되지 않습니다. *topic*의 최대 크기는 40문자입니다. 문자열 하나에 주제를 여러 개(40문자까지) 지정할 수 있습니다. 각 주제는 콜론으로 구분해야 합니다. 예를 들어, "t1:t2:the third topic"은 이 메시지가 t1, t2, "the third topic"이라는 세 주제와 연관되어 있음을 나타냅니다.

correl-id

이 메시지와 연관시킬 상관 ID가 포함된 선택적 문자열 표현식. 대개 *correl-id*를 요청과 응답 시나리오에 지정하여 요청과 응답을 연관시킵니다. 이 옵션을 지정하지 않으면 메시지에 상관 ID가 추가되지 않습니다. *correl-id*의 최대 크기는 24문자입니다.

예

예 1: 이 예에서는 기본 규정(DB2.DEFAULT.POLICY)을 사용하여 문자열 "Testing 123"을 기본 publisher 서비스(DB2.DEFAULT.PUBLISHER)에 발행합니다. 메시지에 상관 ID나 주제는 지정되지 않습니다.

```
VALUES MQPUBLISH('Testing 123')
```

예 2: 이 예에서는 "TESTS"라는 주제 하에 문자열 "Testing 345"를 "MYPUBLISHER" publisher 서비스에 발행합니다. 기본 규정이 사용되고 상관 ID는 지정되지 않습니다.

```
VALUES MQPUBLISH('MYPUBLISHER','Testing 345', 'TESTS')
```

예 3: 이 예에서는 상관 ID를 "TEST1"로 지정하고 "MYPOLICY" 규정을 사용하여 문자열 "Testing 678"을 "MYPUBLISHER" publisher 서비스에 발행합니다. 이 메시지는 "TESTS"라는 주제로 발행됩니다.

```
VALUES MQPUBLISH('MYPUBLISHER','MYPOLICY','Testing 678','TESTS','TEST1')
```

예 4: 이 예에서는 상관 ID는 지정하지 않고 기본 규정(DB2.DEFAULT.POLICY)을 사용하여 "TESTS"라는 주제 하에 문자열 "Testing 901"을 "MYPUBLISHER" publisher 서비스에 발행합니다.

```
VALUES MQPUBLISH('Testing 901','TESTS')
```

성공할 경우, 모든 예는 '1'값을 리턴합니다.

MQREAD

```
MQREAD ( [receive-service] [,service-policy] )
```

스키마는 MQDB2입니다.

MQREAD 함수는 *service-policy*가 정의한 서비스 규정 품질을 사용하여 *receive-service*가 지정한 MQSeries 위치에서 메시지를 리턴합니다. 이 작업을 실행하면 대기행렬에서 *receive-service*와 연관된 메시지가 제거되지 않고, 대기행렬 헤드에 있는 메시지가 리턴됩니다. 리턴 값은 메시지가 포함된 VARCHAR(4000)입니다. 리턴할 메시지가 없을 때는 널(NULL)이 리턴됩니다.

receive-service

메시지가 수신될 논리적 MQSeries 목적지가 포함된 문자열. 이 옵션을 지정하면 *receive-service*가 AMT.XML 저장소 파일에 정의된 Service Point를 참조해야 합니다. 서비스 지점은 메시지가 송신되거나 수신되는 논리적 끝점입니다. 서비스 지점 정의에는 MQSeries 대기행렬 관리자 이름 및 대기행렬 이름이 포함됩니다. 자세한 정보는 MQSeries Application Messaging Interface를 참조하십시오.

*receive-service*가 지정되어 있지 않으면 DB2.DEFAULT.SERVICE가 사용됩니다.
*receive-service*의 최대 크기는 48문자입니다.

service-policy

이 메시지를 처리할 때 사용할 MQSeries AMI Service Policy가 포함된 문자열.
이 옵션을 지정하면 *service-policy*가 AMT.XML 저장소 파일에 정의된 규정을 참조해야 합니다. 서비스 규정은 이 메시징 작업에 적용되어야 할 서비스 옵션의 품질 세트를 정의합니다. 이 옵션에는 메시지 우선순위와 메시지 지속성이 포함됩니다. 자세한 정보는 MQSeries Application Messaging Interface를 참조하십시오.
*service-policy*가 지정되어 있지 않으면 기본 DB2.DEFAULT.POLICY가 사용됩니다.
*service-policy*의 최대 크기는 48문자입니다.

예:

예 1: 이 예에서는 기본 규정(DB2.DEFAULT.POLICY)을 사용하여 기본 서비스(DB2.DEFAULT.SERVICE)가 지정한 대기행렬 헤드에 있는 메시지를 읽습니다.

VALUES MQREAD()

예 2: 이 예에서는 기본 규정(DB2.DEFAULT.POLICY)을 사용하여 "MYSERVICE" 서비스가 지정한 대기행렬 헤드에 있는 메시지를 읽습니다.

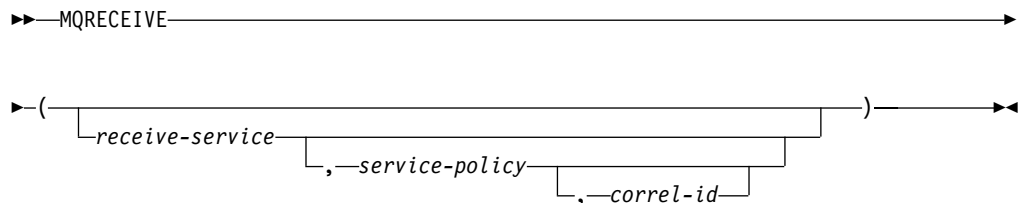
VALUES MQREAD('MYSERVICE')

예 3: 이 예에서는 "MYPOLICY" 규정을 사용하여 "MYSERVICE" 서비스가 지정한 대기행렬 헤드에 있는 메시지를 읽습니다.

VALUES MQREAD('MYSERVICE', 'MYPOLICY')

성공할 경우, 모든 예는 메시지의 내용을 VARCHAR(4000)로 리턴합니다. 리턴할 메시지가 없을 때는 널(NULL)이 리턴됩니다.

MQRECEIVE



스키마는 MQDB2입니다.

MQRECEIVE 함수는 *service-policy*가 정의한 서비스 규정 품질을 사용하여 *receive-service*가 지정한 MQSeries 위치에서 메시지를 리턴합니다. 이 작업을 수행하면 *receive-service*와 연관된 대기행렬에서 메시지가 제거됩니다. *correl-id*가 지정되어 있으면 일치하는 상관 ID가 있는 첫 번째 메시지가 리턴됩니다. *correl-id*가 지정되어

있지 않으면 대기행렬 헤드에 있는 메시지가 리턴됩니다. 리턴 값은 메시지가 포함된 VARCHAR(4000)입니다. 리턴할 메시지가 없을 때는 널(NULL)이 리턴됩니다.

receive-service

메시지가 수신될 논리적 MQSeries 목적지가 포함된 문자열. 이 옵션을 지정하면 *receive-service*가 AMT.XML 저장소 파일에 정의된 Service Point를 참조해야 합니다. 서비스 지점은 메시지가 송신되거나 수신되는 논리적 끝점입니다. 서비스 지점 정의에는 MQSeries 대기행렬 관리자 이름 및 대기행렬 이름이 포함됩니다. 자세한 정보는 MQSeries Application Messaging Interface를 참조하십시오. *receive-service*가 지정되어 있지 않으면 DB2.DEFAULT.SERVICE가 사용됩니다. *receive-service*의 최대 크기는 48문자입니다.

service-policy

이 메시지를 처리할 때 사용할 MQSeries AMI Service Policy가 포함된 문자열. 이 옵션을 지정하면 *service-policy*가 AMT.XML 저장소 파일에 정의된 규정을 참조해야 합니다.⁴ *service-policy*가 지정되어 있지 않으면 기본 DB2.DEFAULT.POLICY가 사용됩니다. *service-policy*의 최대 크기는 48문자입니다.

correl-id

이 메시지와 연관시킬 선택적 상관 ID가 포함된 문자열. 대개 *correl-id*를 요청과 응답 시나리오에 지정하여 요청과 응답을 연관시킵니다. 이 옵션을 지정하지 않으면 상관 ID가 사용되지 않습니다. *correl-id*의 최대 크기는 24문자입니다.

예:

예 1: 이 예에서는 기본 규정(DB2.DEFAULT.POLICY)을 사용하여 기본 서비스(DB2.DEFAULT.SERVICE)가 지정한 대기행렬 헤드에 있는 메시지를 수신합니다.

VALUES MQRECEIVE()

예 2: 이 예에서는 기본 규정(DB2.DEFAULT.POLICY)을 사용하여 "MYSERVICE" 서비스가 지정한 대기행렬 헤드에 있는 메시지를 수신합니다.

VALUES MQRECEIVE('MYSERVICE')

예 3: 이 예에서는 "MYPOLICY" 규정을 사용하여 "MYSERVICE" 서비스가 지정한 대기행렬 헤드에 있는 메시지를 수신합니다.

VALUES MQRECEIVE('MYSERVICE', 'MYPOLICY')

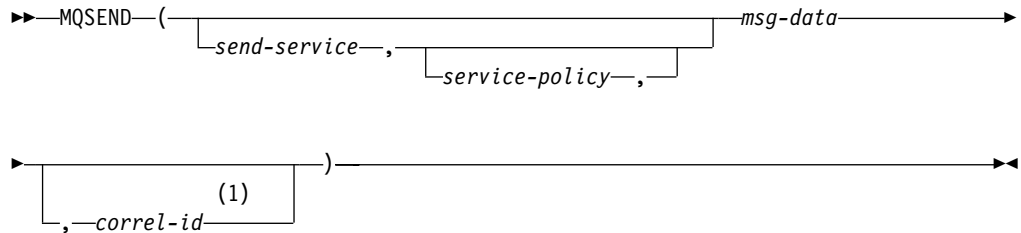
예 4: 이 예에서는 "MYPOLICY" 규정을 사용하여 "MYSERVICE" 서비스가 지정한 대기행렬 헤드에서 '1234'와 일치하는 상관 ID가 있는 첫 번째 메시지를 수신합니다.

4. 서비스 규정은 이 메시징 작업에 적용되어야 할 서비스 옵션 품질 세트를 정의합니다. 이 옵션에는 메시지 우선순위와 메시지 지속성이 포함됩니다. 자세한 정보는 MQSeries Application Messaging Interface를 참조하십시오.

VALUES MQRECEIVE('MYSERVICE',MYPOLICY','1234')

성공할 경우, 모든 예는 메시지의 내용을 VARCHAR(4000)로 리턴합니다. 리턴할 메시지가 없을 때는 널(NULL)이 리턴됩니다.

MQSEND



주:

1 *service*와 *policy*를 이전에 정의하지 않은 경우, *correl-id*를 지정할 수 없습니다.

스키마는 MQDB2입니다.

MQSEND 함수는 *service-policy*가 정의한 서비스 규정 품질을 사용하여 *msg-data*에 포함된 데이터를 *send-service*가 지정한 MQSeries로 송신합니다. *correl-id*로 선택적 사용자 정의 메시지 상관 ID를 지정할 수도 있습니다. 이 함수는 성공할 경우 "1"값을 리턴하고, 성공하지 못할 경우 "0"값을 리턴합니다.

msg-data

MQSeries를 통해 전송할 데이터가 포함된 문자열 표현식. 최대 크기는 4000문자입니다.

send-service

메시지가 송신될 논리적 MQSeries 목적지가 포함된 문자열. 이 옵션을 지정하면 *send-service*가 AMT.XML 저장소 파일에 정의된 service point를 참조해야 합니다. 서비스 지점은 메시지가 송신되거나 수신되는 논리적 끝점입니다. 서비스 지점 정의에는 MQSeries 대기행렬 관리자 이름 및 대기행렬 이름이 포함됩니다. 자세한 정보는 MQSeries Application Messaging Interface를 참조하십시오. *send-service*가 지정되어 있지 않으면 DB2.DEFAULT.SERVICE가 사용됩니다. *send-service*의 최대 크기는 48문자입니다.

service-policy

이 메시지를 처리할 때 사용할 MQSeries AMI Service Policy가 포함된 문자열. 이 옵션을 지정하면 *service-policy*가 AMT.XML 저장소 파일에 정의된 서비스 규정을 참조해야 합니다. 서비스 규정은 이 메시징 작업에 적용되어야 할 서비스 옵션의 품질 세트를 정의합니다. 이 옵션에는 메시지 우선순위와 메시지 지속성이 포함됩니다. 자세한 정보는 MQSeries Application Messaging Interface를 참조하십시오.

시오. *service-policy*가 지정되어 있지 않으면 기본 DB2.DEFAULT.POLICY가 사용됩니다. *service-policy*의 최대 크기는 48문자입니다.

correl-id

이 메시지와 연관시킬 상관 ID가 포함된 선택적 문자열. 대개 *correl-id*를 요청과 응답 시나리오에 지정하여 요청과 응답을 연관시킵니다. 이 옵션을 지정하지 않으면 상관 ID가 송신되지 않습니다. *correl-id*의 최대 크기는 24문자입니다.

예:

예 1: 이 예에서는 상관 ID를 지정하지 않고 기본 규정(DB2.DEFAULT.POLICY)을 사용하여 문자열 "Testing 123"을 기본 서비스(DB2.DEFAULT.SERVICE)에 송신합니다.

```
VALUES MQSEND('Testing 123')
```

예 2: 이 예에서는 상관 ID를 지정하지 않고 "MYPOLOCY" 규정을 사용하여 문자열 "Testing 345"를 "MYSERVICE" 서비스에 송신합니다.

```
VALUES MQSEND('MYSERVICE','MYPOLICY','Testing 345')
```

예 3: 이 예에서는 상관 ID를 "TEST3"으로 지정하고 "MYPOLOCY" 규정을 사용하여 문자열 "Testing 678"을 "MYSERVICE" 서비스에 송신합니다.

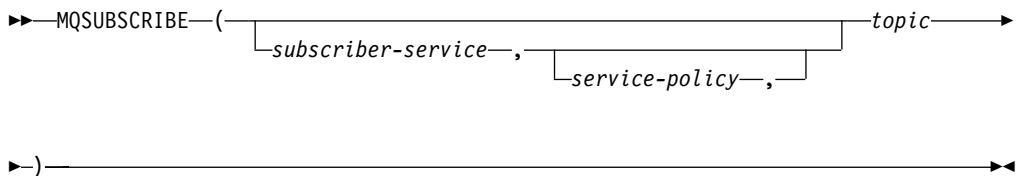
```
VALUES MQSEND('MYSERVICE','MYPOLICY','Testing 678','TEST3')
```

예 4: 이 예에서는 상관 ID를 지정하지 않고 기본 규정(DB2.DEFAULT.POLICY)을 사용하여 문자열 "TESTING 901"을 "MYSERVICE" 서비스에 송신합니다.

```
VALUES MQSEND('MYSERVICE','Testing 901')
```

성공할 경우, 모든 예는 '1' 스칼라 값을 리턴합니다.

MQSUBSCRIBE



스키마는 MQDB2입니다.

MQSUBSCRIBE 함수는 지정된 주제로 발행된 MQSeries 메시지에 관련 사항을 등록하는 데 사용됩니다. *subscriber-service*는 지정된 주제와 일치하는 메시지의 논리적 목적지를 지정합니다. *topic*과 일치하는 메시지가 *subscriber-service*가 정의한 대기행렬에 놓이면, MQREAD, MQRECEIVE, MQREADALL, MQRECEIVEALL을 호출하여 메시지를 읽거나 수신할 수 있습니다. 이 함수를 사용하려면 MQSeries

Integrator나 MQSeries Publish/Subscribe와 같은 MQSeries 기반 publish 및 subscribe 시스템을 설치하고 구성해야 합니다. 자세한 정보는 www.ibm.com/software/MQSeries 를 참조하십시오.

이 함수는 성공할 경우 "1"값을 리턴하고, 성공하지 못할 경우 "0"값을 리턴합니다. 이 함수가 성공적으로 실행되면 publish 및 subscribe 서버가 주제와 일치하는 메시지를 *subscriber-service*가 정의한 서비스 지점으로 전달합니다.

subscriber-service

*topic*과 일치하는 메시지가 송신될 논리적 MQSeries subscription 지점이 포함된 문자열. 이 옵션을 지정하면 *subscriber-service*가 AMT.XML 저장소 파일에 정의된 Subscriber Service Point를 참조해야 합니다. 서비스 지점 정의에는 MQSeries 대기행렬 관리자 이름 및 대기행렬 이름이 포함됩니다. 자세한 정보는 MQSeries Application Messaging Interface를 참조하십시오. *subscriber-service*가 지정되어 있지 않으면 DB2.DEFAULT.SUBSCRIBER가 사용됩니다. *subscriber-service*의 최대 크기는 48문자입니다.

service-policy

메시지를 처리할 때 사용할 MQSeries AMI Service Policy가 포함된 문자열. 이 옵션을 지정하면 *service-policy*가 AMT.XML 저장소 파일에 정의된 규정을 참조해야 합니다. 서비스 규정은 이 메시징 작업에 적용되어야 할 서비스 옵션의 품질 세트를 정의합니다. 이 옵션에는 메시지 우선순위와 메시지 지속성이 포함됩니다. 자세한 정보는 MQSeries Application Messaging Interface를 참조하십시오. *service-policy*가 지정되어 있지 않으면 기본 DB2.DEFAULT.POLICY가 사용됩니다. *service-policy*의 최대 크기는 48문자입니다.

topic

수신할 메시지 유형을 정의하는 문자열. 이 subscription은 지정된 주제로 발행된 메시지만 수신합니다. 여러 subscription이 공존할 수 있습니다. 주제의 최대 크기는 40문자입니다. 문자열 하나에 주제를 여러 개(40문자까지) 지정할 수 있습니다. 각 주제는 콜론으로 구분해야 합니다. 예를 들어, "t1:t2:the third topic"은 이 메시지가 t1, t2, "the third topic"이라는 세 주제와 연관되어 있음을 나타냅니다.

예:

예 1: 이 예에서는 "Weather"라는 주제가 포함된 메시지에 관련 사항을 등록합니다. 기본 *subscriber-service*(DB2.DEFAULT.SUBSCRIBER)가 subscriber로 등록되고, 기본 *service-policy*(DB2.DEFAULT.POLICY)가 서비스 품질을 지정합니다.

```
VALUES MQSUBSCRIBE('Weather')
```

예 2: 이 예에서는 subscriber가 "Stocks"가 포함된 메시지에 관련 사항을 등록하는 것을 보여 줍니다. subscriber는 "BASIC-POLOCY" 규정을 사용하여 "PORTFOLIO-UPDATES"로 등록되어 있습니다.

VALUES MQSUBSCRIBE('PORTFOLIO-UPDATES','BASIC-POLICY','Stocks')

성공할 경우, 모든 예는 '1' 스칼라 값을 리턴합니다.

MQUNSUBSCRIBE

```
▶—MQUNSUBSCRIBE—(—  
    └──subscriber-service—, └──service-policy—, └──  
    └──topic—)——▶
```

스키마는 MQDB2입니다.

MQUNSUBSCRIBE 함수는 기존의 메시지 subscription의 등록을 취소하는 데 사용됩니다. 어떤 subscription을 취소했는지 식별하기 위해 *subscriber-service*, *service-policy*, *topic*이 사용됩니다. 이 함수를 사용하려면 MQSeries Integrator나 MQSeries Publish/Subscribe와 같은 MQSeries 기반 publish 및 subscribe 시스템을 설치하고 구성해야 합니다. 자세한 정보는 www.ibm.com/software/MQSeries를 참조하십시오.

이 함수는 성공할 경우 "1"값을 리턴하고, 성공하지 못할 경우 "0"값을 리턴합니다. 이 함수가 성공적으로 실행되면 publish 및 subscribe 서버가 주어진 매개변수로 정의한 subscription을 제거합니다. 지정된 *topic*으로 된 메시지는 더 이상 *subscriber-service*가 정의한 논리적 목적지로 송신되지 않습니다.

subscriber-service

이 옵션을 지정하면 *subscriber-service*가 AMT.XML 저장소 파일에 정의된 Subscriber Service Point를 참조해야 합니다. 서비스 지점 정의에는 MQSeries 대기행렬 관리자 이름 및 대기행렬 이름이 포함됩니다. 자세한 정보는 MQSeries Application Messaging Interface를 참조하십시오. *subscriber-service*가 지정되어 있지 않으면 DB2.DEFAULT.SUBSCRIBER가 사용됩니다. *subscriber-service*의 최대 크기는 48문자입니다.

service-policy

이 옵션을 지정하면 *service-policy*가 AMT.XML 저장소 파일에 정의된 규정을 참조해야 합니다. 서비스 규정은 이 메시징 작업에 적용되어야 할 서비스 옵션의 품질 세트를 정의합니다. 자세한 정보는 MQSeries Application Messaging Interface를 참조하십시오. *service-policy*가 지정되어 있지 않으면 기본 DB2.DEFAULT.POLICY가 사용됩니다. *service-policy*의 최대 크기는 48문자입니다.

topic

수신하지 않을 메시지의 제목을 지정하는 문자열. *topic*의 최대 크기는 40문자입니다. 문자열 하나에 주제를 여러 개(40문자까지) 지정할 수 있습니다. 각 주제는 콜론으로 구분해야 합니다. 예를 들어, "t1:t2:the third topic"은 이 메시지가 t1, t2, "the third topic"이라는 세가지 주제와 연관되어 있음을 나타냅니다.

예:

예 1: 이 예에서는 "Weather"라는 주제가 포함된 메시지에 대한 관련 사항을 취소합니다. 기본 subscriber-service(DB2.DEFAULT.SUBSCRIBER)가 unsubscriber로 등록되고, 기본 service-policy(DB2.DEFAULT.POLICY)가 서비스 품질을 지정합니다.

VALUES MQUNSUBSCRIBE('Weather')

예 2: 이 예에서는 subscriber가 "Stocks"가 포함된 메시지에 대한 관련 사항을 취소하는 것을 보여 줍니다. subscriber는 "BASIC-POLICY" 규정을 사용하여 "PORTFOLIO-UPDATES"로 등록되어 있습니다.

VALUES MQUNSUBSCRIBE('PORTFOLIO-UPDATES','BASIC-POLICY','Stocks')

성공할 경우, 모든 예는 '1' 스칼라 값을 리턴하고, 성공하지 못할 경우 "0" 스칼라 값을 리턴합니다.

MULTIPLY_ALT

►—MULTIPLY_ALT—(*exact_numeric_expression*, *exact_numeric_expression*)—►

스키마는 SYSIBM입니다.

이 함수는 버전 7.1의 FixPak 2에서 처음 사용 가능합니다.

MULTIPLY_ALT 스칼라 함수는 두 인수의 결과를 소수 값으로 리턴합니다. 특히 인수의 정밀도의 합이 31을 초과할 때, 곱셈 연산자의 대안으로 제공됩니다.

인수는 내장된 정확한 숫자 데이터 유형(DECIMAL, BIGINT, INTEGER 또는 SMALLINT)이 될 수 있습니다.

함수의 결과는 DECIMAL입니다. 결과의 정밀도와 스케일은 기호 p와 s를 사용하여 첫번째 인수의 정밀도와 스케일을 표시하고, 기호 p'와 s'를 사용하여 두 번째 인수의 정밀도와 스케일을 표시하는 방법으로 결정됩니다.

- 정밀도는 $\text{MIN}(31, p + p')$ 입니다.
- 스케일은 다음과 같습니다.
 - 두 인수의 스케일이 0이면 0
 - $p+p'$ 가 31과 같거나 작을 경우 $\text{MIN}(31, s+s')$
 - $p+p'$ 가 31보다 클 경우 $\text{MAX}(\text{MIN}(3, s+s'), 31-(p-s+p'-s'))$

최소한 한 인수가 널(NULL)이거나 데이터베이스가 DFT_SQLMATHWARN이 yes로 설정되어 구성되어 있을 경우 결과가 널(NULL)일 수 있습니다. 인수의 하나가 널(NULL)이면 결과가 널(NULL) 값입니다.

최소한 3의 스케일이 필요하고 정밀도의 합이 31을 초과하는 소수 연산을 수행할때 곱셈 연산자 대신 MULTIPLY_ALT 함수를 선택하는 것이 좋습니다. 이런 경우, 내부 계산이 수행되어 오버플로우가 방지됩니다. 최종 결과는 스케일과 일치하기 위해 필요한 경우 절단을 사용하여 결과 데이터 유형에 할당됩니다. 스케일이 3일 때도 최종 결과의 오버플로우가 가능함을 유의하십시오.

다음은 MULTIPLY_ALT와 곱셈 연산자를 사용하여 결과 유형을 비교하는 샘플입니다.

인수 유형 1	인수 유형 2	MULTIPLY_ALT를 사용한 결과	곱셈 연산자를 사용한 결과
DECIMAL(31,3)	DECIMAL(15,8)	DECIMAL(31,3)	DECIMAL(31,11)
DECIMAL(26,23)	DECIMAL(10,1)	DECIMAL(31,19)	DECIMAL(31,24)
DECIMAL(18,17)	DECIMAL(20,19)	DECIMAL(31,29)	DECIMAL(31,31)
DECIMAL(16,3)	DECIMAL(17,8)	DECIMAL(31,9)	DECIMAL(31,11)
DECIMAL(26,5)	DECIMAL(11,0)	DECIMAL(31,3)	DECIMAL(31,5)
DECIMAL(21,1)	DECIMAL(15,1)	DECIMAL(31,2)	DECIMAL(31,2)

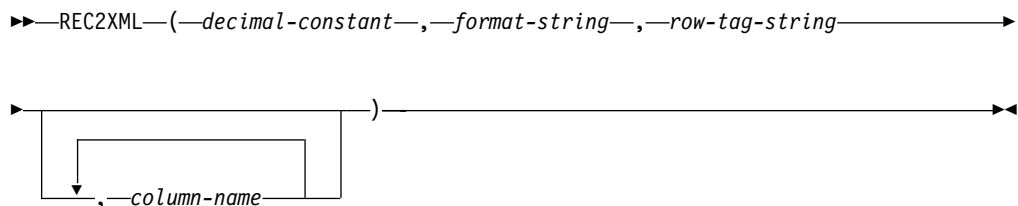
예:

첫번째 인수의 데이터 유형이 DECIMAL(26, 3)이고 두 번째 인수의 데이터 유형이 DECIMAL(9,8)일 때 두 값을 곱하십시오. 결과의 데이터 유형은 DECIMAL(31,7)입니다.

```
values multiply_alt(98765432109876543210987.654,5.43210987)
1
-----
536504678578875294857887.5277415
```

이들 두 수의 완전한 결과는 536504678578875294857887.52774154498이지만 마지막 4 자리는 결과 데이터 유형의 스케일과 일치시키기 위해 절단되었음을 유의하십시오. 같은 값들로 곱셈 연산자를 사용하면 결과 데이터 유형이 DECIMAL(31,11)이고 결과 값이 소수점 왼쪽 24 자리이지만, 결과 데이터 유형은 20 자리만 지원하므로 산술 오버플로우가 발생합니다.

REC2XML



스키마는 SYSIBM입니다.

REC2XML 함수는 XML 태그로 형식화되고 컬럼 이름과 컬럼 값이 포함된 문자열을 리턴합니다.

decimal-constant

컬럼 값 문자를 바꾸기 위한 확장 인수. 소수 값은 0.0 이상 6.0 미만이어야 합니다(SQLSTATE 42820).

decimal-constant 값은 함수의 결과 길이를 계산하는 데 사용됩니다. 문자 데이터 유형으로 된 모든 컬럼에서, 컬럼의 길이 속성은 결과 길이에 추가되기 전에 이 확장 인수로 곱해집니다.

확장을 지정하지 않으려면 1.0 값을 사용하십시오. 1.0 미만의 값을 지정하면 계산된 결과 길이가 줄어듭니다. 결과 문자열의 실제 길이가 함수에 대해 계산된 결과 길이보다 크면 오류(SQLSTATE 22001)가 발생합니다.

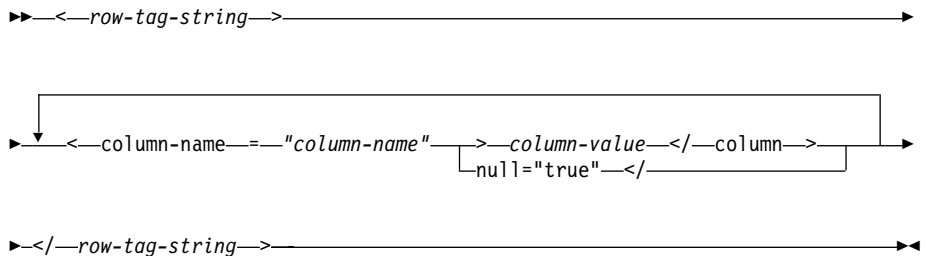
format-string

함수 실행 중 함수가 사용할 형식을 지정하는 문자열 상수

*format-string*은 대소문자를 구별하므로 다음 값은 대문자로 지정해야 인식됩니다.

COLATTVAL 또는 COLATTVAL_XML

이러한 형식은 컬럼의 문자열을 속성 값으로 리턴합니다.



컬럼은 유효한 XML 속성 값일 수도 있고 그렇지 않을 수도 있습니다. 유효한 XML 속성 값이 아닌 컬럼 이름에 대해서는, 컬럼 이름을 결과 문자열에 포함하기 전에 컬럼 이름에서 문자 바꾸기가 수행됩니다.

컬럼 값은 유효한 XML 요소 값일 수도 있고 그렇지 않을 수도 있습니다.

format-string COLATTVAL이 지정되어 있으면, 유효한 XML 요소 값이 아닌 열 값에 대해 열 값을 결과 문자열에 포함하기 전에 컬럼 값에서 문자 바꾸기가 수행됩니다. *format-string* COLATTVAL_XML이 지정되어 있으면, 컬럼 값에 대해 문자 바꾸기가 수행되지 않습니다(컬럼 이름에 대해서는 문자 바꾸기가 수행됨).

row-tag-string

각 행에 사용할 태그를 지정하는 문자열 상수

빈 문자열이 지정되어 있으면 'row' 값을 사용합니다.

하나 이상의 빈 문자로 된 문자열이 지정되어 있으면 결과 문자열에 시작하는 *row-tag-string*이나 끝나는 *row-tag-string*(꺀쇠 괄호 분리문자 포함)이 나타나지 않습니다.

column-name

테이블 컬럼의 규정된 이름이나 규정되지 않은 이름. 컬럼은 다음 데이터 유형 중 하나로 되어 있어야 합니다(SQLSTATE 42815).

- 숫자(SMALLINT, INTEGER, BIGINT, DECIMAL, NUMERIC, REAL, DOUBLE)
- 문자열(Char, VARCHAR)⁵
- 날짜/시간(DATE, TIME, TIMESTAMP)
- 위 유형에 바탕을 둔 사용자 정의 유형

동일한 컬럼 이름을 한 번 이상 지정할 수 없습니다(SQLSTATE 42734).

함수의 결과는 VARCHAR입니다. 최대 길이는 32672 바이트(SQLSTATE 54006)입니다.

다음 호출을 검토해 보십시오.

REC2XML (dc, fs, rt, c₁, c₂, ..., c_n)

fs의 값이 "COLATTVAL"이나 "COLATTVAL_XML"이면

결과는 다음 표현식과 동일합니다.

```
'<' CONCAT rt CONCAT '>' CONCAT y1 CONCAT y2 CONCAT ... CONCAT yn  
CONCAT '</' CONCAT rt CONCAT '>'
```

여기서 y_n은 다음과 동일합니다.

```
'<column name="' CONCAT xvcn CONCAT vn
```

v_n은 다음과 동일합니다.

```
'">' CONCAT rn CONCAT '</column>'
```

이는 컬럼이 널(NULL)이 아니고,

```
'" null="true"/>'
```

컬럼 값이 널(NULL)인 경우 적용됩니다.

xvc_n은 c_n 컬럼 이름의 문자열 표현식과 동일하고, 이 경우 433 페이지의 표29에 나타나는 모든 문자는 해당 표현으로 바뀝니다. 이는 결과 문자열이 유효한 XML 속성이나 요소 값 토큰이 되도록 하기 위해서입니다.

5. BIT DATA 부속 유형으로 된 문자열은 허용되지 않습니다.

r_n 은 표28에 표시된 문자열 표현과 동일합니다.

결과 컬럼 값:

컬럼의 데이터 유형 및 지정된 실제 *format-string*에 따라 결과 문자열에 연결되기 전에 테이블의 결과 값이 변환됩니다. 다음 표에서 컬럼 값에 이루어지는 변환을 볼 수 있습니다.

표 28. 컬럼 값 문자열 결과

c_n 데이터 유형	r_n
CHAR, VARCHAR	값은 문자열입니다. <i>format-string</i> 이 "_XML" 문자로 끝나지 않으면, c_n 의 각 문자가 표29에 표시된 해당 대체 표현으로 바뀝니다. 길이 속성은 c_n 의 dc * 길이 속성입니다.
SMALLINT, INTEGER, BIGINT, DECIMAL, NUMERIC, REAL, DOUBLE	값은 LTRIM(RTRIM(CHAR(c_n)))입니다. 길이 속성은 CHAR(c_n)의 결과 길이입니다. 소수점 표시 문자는 항상 마침표입니다.
DATE	값은 CHAR(c_n , ISO)입니다. 길이 속성은 CHAR(c_n , ISO)의 결과 길이입니다.
TIME	값은 CHAR(c_n , JIS)입니다. 길이 속성은 CHAR(c_n , JIS)의 결과 길이입니다.
TIMESTAMP	값은 CHAR(c_n)입니다. 길이 속성은 CHAR(c_n)의 결과 길이입니다.

문자 바꾸기:

*format-string*에 지정된 값에 따라 컬럼 이름이 유효한 XML 속성 값을 이루고, 컬럼 값은 유효한 XML 요소 값을 이루도록 컬럼 이름과 컬럼 값에 있는 특정 문자가 바뀝니다.

표 29. XML 속성 값 및 요소 값에 대한 문자 바꾸기

<는 <로 바뀝니다.
>는 >로 바뀝니다.
"는 "로 바뀝니다.
&는 &로 바뀝니다.
'는 '로 바뀝니다.

예:

- DEPARTMENT 테이블의 DEPTNAME과 LOCATION 열을 제외한 다른 열에서 DEPARTMENT 'D01'이 유효한 XML 문자열이 되도록 DEPARTMENT 테이블 행의 형식을 지정하십시오. 데이터에 바뀌어야 할 문자가 포함되어 있지 않으므로 확장 인수는 1.0(확장 안함)이 됩니다. 또한 이 행에 대한 MGRNO 값은 널(NULL)입니다.

```

SELECT REC2XML (1.0, 'COLATTVAL', '', DEPTNO, MGRNO, ADMRDEPT)
FROM DEPARTMENT
WHERE DEPTNO = 'D01'

```

이 예는 다음과 같은 VARCHAR(117) 문자열을 리턴합니다.

```

<row>
  <column name="DEPTNO">D01</column>
  <column name="MGRNO" null="true"/>
  <column name="ADMRDEPT">A00</column>
</row>

```

주: REC2XML은 출력에 새로운 줄 문자를 삽입하지 않습니다. 위의 예 출력은 읽기 쉽게 형식화된 것입니다.

- 5일로 된 대학 시간표에서, CL_SCHED라는 테이블에 '&43<FIE'라는 수업이 도입되었고, CLASS_CODE 컬럼에는 새로운 형식이 지정되었습니다. 이 예에서 REC2XML 함수를 사용하면 새로운 수업 데이터에서 수업 종료 시간을 제외한 다른 데이터에 XML 문자열 형식이 지정됩니다.

확장 인수가 1.0인 REC2XML 호출(아래 참조)의 길이 속성은 128이 됩니다 ('<row>' '</row>' 오버헤드에 11, 컬럼 이름에 21, '<column name=', '>', '</column>' 큰따옴표에 75, CLASS_CODE 데이터에 7, DAY 데이터에 6, STARTING 데이터에 8). '&'와 '<' 문자는 다른 표현으로 바뀌므로 확장 인수를 1.0으로 지정하는 것으로는 충분하지 않습니다. 이 함수의 길이 속성은 새로운 형식의 CLASS_CODE 데이터에 대해 7 - 14문자의 증가를 지원해야 합니다.

그러나 DAY 값은 1자리 이상이 되지 않으므로 DAY 길이에 대해 계산된 나머지 5자리는 사용되지 않습니다. 따라서 확장은 2의 증가분만 처리하면 됩니다. CLASS_CODE만 인수 목록에 있는 유일한 문자열 컬럼이므로, 이 컬럼 값에만 확장 인수가 적용됩니다. 길이에 2의 증가분을 확보하려면 9/7(약 1.2857)의 확장 인수가 필요합니다. 따라서 1.3 확장 인수가 사용됩니다.

```

SELECT REC2XML (1.3, 'COLATTVAL', 'record', CLASS_CODE, DAY, STARTING)
FROM CL_SCHED
WHERE CLASS_CODE = '&43<FIE'

```

다음 문자열을 리턴합니다.

```

<record>
  <column name="CLASS_CODE">&43<FIE</column>
  <column name="DAY">5</column>
  <column name="STARTING">06:45:00</column>
</record>

```

주: REC2XML은 출력에 새로운 줄 문자를 삽입하지 않습니다. 위의 예 출력은 읽기 쉽게 형식화된 것입니다.

- 이 예는 컬럼 이름에서 바뀐 문자를 보여 줍니다.

```

SELECT REC2XML (1.3,'COLATTVAL', '', Class, "time<noon")
FROM (SELECT Class_code, Starting
      FROM C1_sched
      WHERE Starting < '12:00:00')
AS Early (Class, "time<noon")

```

이 조회는 다음 문자열을 리턴합니다.

```

<row>
  <column name="CLASS">&43&lt;FIE</column>
  <column name="time&lt;noon">06:45:00</column>
</row>

```

주: REC2XML은 출력에 새로운 줄 문자를 삽입하지 않습니다. 위의 예 출력은 읽기 쉽게 형식화된 것입니다.

- 이력서를 유효한 XML 문자열로 저장하는 EMP_RESUME 테이블에 새로운 행이 추가되었다고 가정해 봅시다. 문자 바꾸기가 수행되지 않도록 COLATTVAL_XML *format-string*이 사용됩니다. 모든 이력서는 길이가 3500문자를 넘지 않습니다. 다음 조회를 사용하여 EMP_RESUM 테이블에서 이력서의 XML 버전을 선택한 다음 XML 문서 단편화로 형식화할 수 있습니다.

```

SELECT REC2XML (1.0,'COLATTVAL_XML', '', EMPNO, RESUME_XML)
FROM (SELECT EMPNO, CAST(RESUME as VARCHAR(3500)) as RESUME_XML
      FROM EMP_RESUME
      WHERE RESUME_FORMAT = 'XML')
AS EMP_RESUME_XML

```

ROUND

▶▶ ROUND—(*expression1*, *expression2*)—▶▶

스키마는 SYSIBM입니다.

이 함수는 버전 7.1의 FixPak 2에서 처음 사용 가능합니다.

주: ROUND 함수의 SYSPFUN 버전을 계속 사용할 수 있습니다.

ROUND 함수는 *expression2*가 양수이면 소수점 오른쪽으로 또는 *expression2*가 0 또는 음수이면 소수점 왼쪽으로 *expression2* 자리로 반올림된 *expression1*을 리턴합니다.

*expression1*이 양수이면, 값 5는 다음 높은 양수로 반올림됩니다. 예를 들어, ROUND(3.5,0) = 4입니다. *expression1*이 음수이면, 값 5는 다음 낮은 음수로 반올림됩니다. 예를 들어, ROUND(-3.5,0) = -4입니다.

expression1

내장 숫자 데이터 유형의 값을 리턴하는 표현식.

expression2

작은 또는 큰 정수를 리턴하는 표현식. *expression2*의 값이 음수가 아닐 때, 소

수점 오른쪽의 그 숫자만큼의 자리로 반올림하도록 지정합니다. *expression2*의 값이 음수이면, 소수점 왼쪽으로 *expression2* 자리의 절대값으로 반올림하도록 지정합니다.

*expression2*가 음수가 아니면, *expression1*은 소수점 오른쪽으로 *expression2* 수의 자리의 절대값으로 반올림됩니다. *expression2*의 값이 *expression1*의 스케일보다 클 경우, 경과 값이 1만큼 큰 정밀도를 가질 때를 제외하고는 값이 변경되지 않습니다. 예를 들어, $\text{ROUND}(748.58,5) = 748.58$ 에서 정밀도가 6 이고 스케일은 계속 2입니다.

*expression2*가 음수이면, *expression1*은 소수점 왼쪽으로 *expression2*+1 수의 자리의 절대값으로 반올림됩니다.

음수 *expression2*의 절대값이 소수점 왼쪽 자리 수보다 크면, 결과는 0입니다. 예를 들어, $\text{ROUND}(748.58,-4) = 0$ 입니다.

*expression1*이 DECIMAL 또는 NUMERIC이고 정밀도가 31보다 작을 경우 정밀도가 1만큼 증가되는 경우를 제외하고, 결과의 데이터 유형 및 길이 속성은 첫번째 인수의 데이터 유형 및 길이 속성과 같습니다. 예를 들어, 데이터 유형이 DECIMAL(5,2)인 인수의 결과는 DECIMAL(6,2)입니다. 데이터 유형이 DECIMAL(31,2)인 인수의 결과는 DECIMAL(31,2)입니다. 스케일은 첫번째 인수의 스케일과 같습니다.

인수가 널(NULL)이 될 수 있거나 데이터베이스가 DFT_SQLMATHWARN을 yes로 설정하여 구성되어 있을 경우, 결과는 널(NULL)이 될 수 있습니다. 인수가 널(NULL)이면, 결과는 널(NULL) 값입니다.

예:: 각각 2, 1, 0, -1, -2, -3 및 -4 소수 자리로 라운드되는 숫자 873.726을 계산하십시오.

```
VALUES (ROUND(873.726, 2),
        ROUND(873.726, 1),
        ROUND(873.726, 0),
        ROUND(873.726,-1),
        ROUND(873.726,-2),
        ROUND(873.726,-3),
        ROUND(873.726,-4) )
```

이 예는 다음을 리턴합니다.

1	2	3	4	5	6	7
873.730	873.700	874.000	870.000	900.000	1000.000	0.000

양수와 음수를 모두 계산하십시오.

```
VALUES (ROUND(3.5, 0),
        ROUND(3.1, 0),
        ROUND(-3.1, 0),
        ROUND(-3.5,0) )
```

이 예는 다음을 리턴합니다.

```
  1   2   3   4
  ---
 4.0  3.0 -3.0 -4.0
```

WEEK_ISO

이 함수의 설명을 다음과 같이 변경하십시오.

스키마는 SYSFUN입니다.

인수의 일년 중 주수를 1-53 범위의 정수 값으로 리턴합니다. 한 주는 월요일에서 시작 하며 항상 7일을 포함합니다. 제 1주는 목요일이 포함된 일년 중 첫번째 주로서, 1월 4일이 포함된 첫번째 주와 같습니다. 따라서 연초의 최대 3일이 이전 연도의 마지막 주 로 나타날 수 있습니다. 반대로 연말의 최대 3일이 다음 연도의 첫번째 주에 나타날 수 있습니다.

인수는 날짜, 시간소인, CLOB이거나 LONG VARCHAR가 아닌 날짜 또는 시간소인 의 유효한 문자열 표현입니다.

함수의 결과는 INTEGER입니다. 결과가 널(NULL)일 수 있습니다. 즉, 인수가 널 (NULL)이면 결과가 널(NULL) 값이 됩니다.

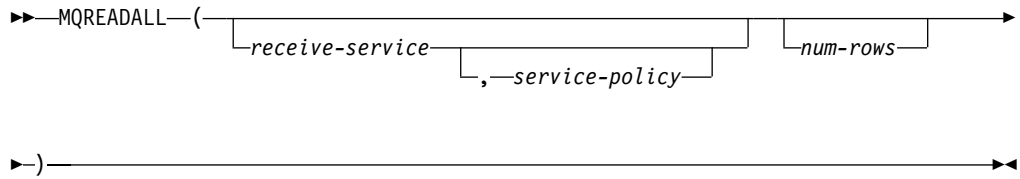
예:

다음 목록에서는 WEEK_ISO 및 DAYOFWEEK_ISO의 결과 예를 보여줍니다.

DATE	WEEK_ISO	DAYOFWEEK_ISO
-----	-----	-----
1997-12-28	52	7
1997-12-31	1	3
1998-01-01	1	4
1999-01-01	53	5
1999-01-04	1	1
1999-12-31	52	5
2000-01-01	52	6
2000-01-03	1	1

테이블 함수

MQREADALL



스키마는 MQDB2입니다.

MQRECEIVE 함수는 *service-policy* 서비스 규정 품질을 사용하여 *receive-service*가 지정한 MQSeries 위치에서 메시지와 메시지 메타데이터가 포함된 테이블을 리턴합니다. 이 작업을 수행하면 대기행렬에서 *receive-service*와 연관된 메시지가 제거되지 않습니다.

*num-rows*가 지정되어 있으면 *num-rows* 메시지의 최대 크기가 리턴됩니다. *num-rows*가 지정되어 있지 않으면 사용 가능한 모든 메시지가 리턴됩니다. 리턴된 테이블에는 다음 컬럼이 포함됩니다.

- MSG - MQSeries 메시지의 내용이 포함된 VARCHAR(4000) 컬럼
- CORRELID - 메시지를 연결할 때 사용되는 상관 ID가 포함된 VARCHAR(24) 컬럼
- TOPIC - 사용 가능한 경우, 메시지와 함께 발행된 제목이 포함된 VARCHAR(40) 컬럼
- QNAME - 메시지가 수신된 대기행렬의 이름이 포함된 VARCHAR(48) 컬럼
- MSGID - 메시지에 할당된 MQSeries 고유 식별자가 포함된 CHAR(24) 컬럼
- MSGFORMAT - MQSeries가 정의한 메시지 형식이 포함된 VARCHAR(8) 컬럼
일반적인 문자열의 형식은 MQSTR입니다.

receive-service

메시지를 읽을 논리적 MQSeries 목적지가 포함된 문자열. 이 옵션을 지정하면 *receive-service*가 AMT.XML 저장소 파일에 정의된 서비스 지점을 참조해야 합니다. 서비스 지점은 메시지가 송신되거나 수신되는 논리적 끝점입니다. 서비스 지점 정의에는 MQSeries 대기행렬 관리자 이름 및 대기행렬 이름이 포함됩니다. 자세한 정보는 MQSeries Application Messaging Interface를 참조하십시오. *receive-service*가 지정되어 있지 않으면 DB2.DEFAULT.SERVICE가 사용됩니다. *receive-service*의 최대 크기는 48문자입니다.

service-policy

이 메시지를 처리할 때 사용할 MQSeries AMI Service Policy가 포함된 문자열. 이 옵션을 지정하면 *service-policy*가 AMT.XML 저장소 파일에 정의된 규정을 참조해야 합니다. 서비스 규정은 이 메시징 작업에 적용되어야 할 서비스 옵션의 품질 세트를 정의합니다. 이 옵션에는 메시지 우선순위와 메시지 지속성이 포함됩니다. 자세한 정보는 MQSeries Application Messaging Interface를 참조하십시오.

*service-policy*가 지정되어 있지 않으면 기본 DB2.DEFAULT.POLICY가 사용됩니다. *service-policy*의 최대 크기는 48문자입니다.

num-rows

이 함수가 리턴할 수 있는 메시지의 최대 개수가 포함된 양의 정수

예:

예 1: 이 예에서는 기본 규정(DB2.DEFAULT.POLICY)을 사용하여 기본 서비스(DB2.DEFAULT.SERVICE)가 지정한 대기행렬에서 모든 메시지를 수신합니다. 메시지와 모든 메타데이터가 테이블로 리턴됩니다.

```
SELECT *  
FROM table (MQREADALL()) T
```

예 2: 이 예에서는 기본 규정(DB2.DEFAULT.POLICY)을 사용하여 MYSERVICE 서비스가 지정한 대기행렬의 헤드에서 모든 메시지를 수신합니다. MSG와 CORRELID 컬럼만 리턴됩니다.

```
SELECT T.MSG, T.CORRELID  
FROM table (MQREADALL('MYSERVICE')) T
```

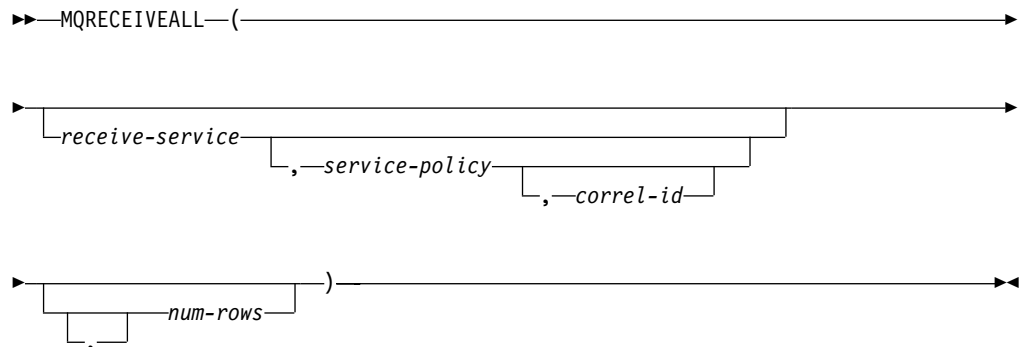
예 3: 이 예에서는 기본 규정(DB2.DEFAULT.POLICY)을 사용하여 기본 서비스(DB2.DEFAULT.SERVICE)가 지정한 대기행렬의 헤드를 읽습니다. CORRELID가 '1234'인 메시지와 모든 컬럼이 리턴됩니다.

```
SELECT *  
FROM table (MQREADALL()) T  
WHERE T.CORRELID = '1234'
```

예 4: 이 예에서는 기본 규정(DB2.DEFAULT.POLICY)을 사용하여 기본 서비스(DB2.DEFAULT.SERVICE)가 지정한 대기행렬 헤드에 있는 처음 10개의 메시지를 수신합니다. 모든 컬럼이 리턴됩니다.

```
SELECT *  
FROM table (MQREADALL(10)) T
```

MQRECEIVEALL



스키마는 MQDB2입니다.

MQRECEIVEALL 함수는 *service-policy* 서비스 규정 품질을 사용하여 *receive-service* 가 지정한 MQSeries 위치에서 메시지와 메시지 메타데이터가 포함된 테이블을 리턴합니다. 이 작업을 수행하면 *receive-service*와 연관된 대기행렬에서 메시지가 제거됩니다.

*correl-id*가 지정되어 있으면 일치하는 상관 ID가 있는 메시지만 리턴됩니다. *correl-id*가 지정되어 있지 않으면 대기행렬 헤드에 있는 메시지가 리턴됩니다.

*num-rows*가 지정되어 있으면 *num-rows* 메시지의 최대 크기가 리턴됩니다. *num-rows*가 지정되어 있지 않으면 사용 가능한 모든 메시지가 리턴됩니다. 리턴된 테이블에는 다음 컬럼이 포함됩니다.

- MSG - MQSeries 메시지의 내용이 포함된 VARCHAR(4000) 컬럼
- CORRELID - 메시지를 연결할 때 사용되는 상관 ID가 포함된 VARCHAR(24) 컬럼
- TOPIC - 사용 가능한 경우, 메시지와 함께 발행된 제목이 포함된 VARCHAR(40) 컬럼
- QNAME - 메시지가 수신된 대기행렬의 이름이 포함된 VARCHAR(48) 컬럼
- MSGID - 메시지에 할당된 MQSeries 고유 식별자가 포함된 CHAR(24) 컬럼
- MSGFORMAT - MQSeries가 정의한 메시지 형식이 포함된 VARCHAR(8) 컬럼
일반적인 문자열의 형식은 MQSTR입니다.

receive-service

메시지가 수신될 논리적 MQSeries 목적지가 포함된 문자열. 이 옵션을 지정하면 *receive-service*가 AMT.XML 저장소 파일에 정의된 서비스 지점을 참조해야 합니다. 서비스 지점은 메시지가 송신되거나 수신되는 논리적 끝점입니다. 서비스 지점 정의에는 MQSeries 대기행렬 관리자 이름 및 대기행렬 이름이 포함됩니다. 자세한 정보는 MQSeries Application Messaging Interface를 참조하십시오. *receive-service*가 지정되어 있지 않으면 DB2.DEFAULT.SERVICE가 사용됩니다. *receive-service*의 최대 크기는 48문자입니다.

service-policy

이 메시지를 처리할 때 사용할 MQSeries AMI Service Policy가 포함된 문자열. 이 옵션을 지정하면 *service-policy*가 AMT.XML 저장소 파일에 정의된 규정을 참조해야 합니다. 서비스 규정은 이 메시징 작업에 적용되어야 할 서비스 옵션의 품질 세트를 정의합니다. 이 옵션에는 메시지 우선순위와 메시지 지속성이 포함됩니다. 자세한 정보는 MQSeries Application Messaging Interface를 참조하십시오. *service-policy*가 지정되어 있지 않으면 기본 DB2.DEFAULT.POLICY가 사용됩니다. *service-policy*의 최대 크기는 48문자입니다.

correl-id

이 메시지와 연관시킬 상관 ID가 포함된 선택적 문자열. 대개 *correl-id*를 요청과

응답 시나리오에 지정하여 요청과 응답을 연관시킵니다. 이 옵션을 지정하지 않으면 상관 ID가 지정되지 않습니다. `correl-id`의 최대 크기는 24문자입니다.

num-rows

이 함수가 리턴할 수 있는 메시지의 최대 개수가 포함된 양의 정수

예:

예 1: 이 예에서는 기본 규정(DB2.DEFAULT.POLICY)을 사용하여 기본 서비스(DB2.DEFAULT.SERVICE)가 지정한 대기행렬에서 모든 메시지를 수신합니다. 메시지와 모든 메타데이터가 테이블로 리턴됩니다.

```
SELECT *  
FROM table (MQRECEIVEALL()) T
```

예 2: 이 예에서는 기본 규정(DB2.DEFAULT.POLICY)을 사용하여 MYSERVICE 서비스가 지정한 대기행렬의 헤드에서 모든 메시지를 수신합니다. MSG와 CORRELID 컬럼만 리턴됩니다.

```
SELECT T.MSG, T.CORRELID  
FROM table (MQRECEIVEALL('MYSERVICE')) T
```

예 3: 이 예에서는 "MYPOLICY" 규정을 사용하여 "MYSERVICE" 서비스가 지정한 대기행렬 헤드에서 모든 메시지를 수신합니다. CORRELID가 '1234'인 메시지와 MSG와 CORRELID 컬럼만 리턴됩니다.

```
SELECT T.MSG, T.CORRELID  
FROM table (MQRECEIVEALL('MYSERVICE','MYPOLICY','1234')) T
```

예 4: 이 예에서는 기본 규정(DB2.DEFAULT.POLICY)을 사용하여 기본 서비스(DB2.DEFAULT.SERVICE)가 지정한 대기행렬 헤드에 있는 처음 10개의 메시지를 수신합니다. 모든 컬럼이 리턴됩니다.

```
SELECT *  
FROM table (MQRECEIVEALL(10)) T
```

프로시저

GET_ROUTINE_SAR

▶▶ GET_ROUTINE_SAR (—sarblob—, —type—, —routine_name_string—) ▶▶

스키마는 SYSFUN입니다.

이 프로시저는 버전 7.1의 FixPak 3에서 처음 사용 가능합니다.

GET_ROUTINE_SAR 프로시저는 동일한 운영 체제에서 동일한 레벨을 수행하는 다른 데이터베이스 서버에 동일한 루틴을 설치하기 위해 필요한 정보를 검색합니다. 정보는 SQL 아카이브 파일을 나타내는 하나의 BLOB 문자열로 검색됩니다. GET_ROUTINE_SAR 프로시저 호출자는 반드시 DBADM 권한을 소유해야 합니다.

sarblob

루틴 SAR 파일 목차가 들어 있는 BLOB(3M) 유형의 출력 인수.

type

다음 값 중 하나를 사용하여, 루틴의 유형을 지정하는 유형 CHAR(2)의 입력 인수.

- 프로시저의 경우 **P**.
- 프로시저의 특정 이름의 경우 **SP**.

routine_name_string

루틴의 규정된 이름을 지정하는 VARCHAR(257) 유형의 입력 인수. 스키마 이름을 지정하지 않으면 루틴이 처리될 때 기본값은 CURRENT SCHEMA입니다.

주: *routine_name_string*은 큰 따옴표 문자(")를 포함할 수 없습니다.

루틴의 규정된 이름을 사용하여 검색할 루틴을 결정합니다. 발견된 루틴은 SQL 루틴이어야 하며 그렇지 않으면 오류가 발생합니다(SQLSTATE 428F7). 특정 이름을 사용하지 않을 경우, 둘 이상의 루틴이 검색되어 오류가 발생합니다(SQLSTATE 42725). 이럴 경우, 루틴의 특정한 이름을 사용하여 루틴을 찾아야 합니다.

SAR 파일에는 서버에서 사용 가능하지 않은 바인드 파일이 포함되어 있어야 합니다. 바인드 파일을 찾을 수 없거나 SAR 파일에 저장되어 있는 경우, 오류가 발생합니다(SQLSTATE 55045).

PUT_ROUTINE_SAR

▶▶ PUT_ROUTINE_SAR (—sarblob—, —new_owner—, —use_register_flag—) ▶▶

스키마는 SYSFUN입니다.

이 프로시저는 버전 7.1의 FixPak 3에서 처음 사용 가능합니다.

PUT_ROUTINE_SAR 프로시저는 서버에서 SQL 루틴 작성에 필요한 파일을 전달한 뒤 루틴을 정의합니다. PUT_ROUTINE_SAR 프로시저 호출자는 반드시 DBADM 권한을 소유해야 합니다.

sarblob

루틴 SAR 파일 목차가 들어 있는 BLOB(3M) 유형의 입력 인수.

new_owner

루틴의 권한 부여 점검에 사용되는 권한 부여 이름이 들어 있는 VARCHAR(128) 유형의 입력 인수. *new-owner*는 정의될 루틴에 필요한 특권을 가져야 합니다. *new-owner*가 지정되지 않은 경우, 원래 루틴 정의자의 *authorization-name*이 사용됩니다.

use_register_flag

루틴 정의에 CURRENT SCHEMA 및 CURRENT PATH 특수 레지스터 사용 여부를 지시하는 INTEGER 유형의 입력 인수. 특수 레지스터가 사용되지 않으면, 기본 스키마 및 SQL 경로의 설정값은 루틴이 원래 정의될 때 사용된 설정값입니다. *use-register-flag*에 대해 가능한 값은 다음과 같습니다.

0 현재 환경의 특수 레지스터를 사용하지 않습니다.

1 CURRENT SCHEMA 및 CURRENT PATH 특수 레지스터를 사용합니다.

값이 1이면, CURRENT SCHEMA는 루틴 정의(루틴 이름 포함)에서 규정되지 않은 오브젝트 이름에 사용되고, CURRENT PATH는 루틴 정의에서 규정되지 않은 루틴 및 데이터 유형 결정에 사용됩니다. *use-registers-flag*가 지정하지 않으면, 0 값이 지정된 경우와 같습니다.

입력이 환경에 대해 적절한지 확인하기 위해 *sarblob*에 들어 있는 식별 정보를 점검하고, 그렇지 않으면 오류가 발생합니다(SQLSTATE 55046). 그런 다음 PUT_ROUTINE_SAR 프로시저는 서버에서 루틴을 정의하기 위해 *sarblob*의 내용을 사용합니다.

sarblob 인수의 내용은 SQL 아카이브 파일을 구성하는 별도의 파일로 추출됩니다. 공유 라이브러리와 바인드 파일은 임시 디렉토리의 파일에 기록됩니다. 루틴 정의 명령문 처리가 컴파일 및 링크가 필요하지 않음을 알고 공유 라이브러리 및 바인드 파일의 위치가 사용 가능하도록 환경이 설정됩니다. 그런 뒤 DDL 파일의 내용을 사용하여 동적으로 루틴 정의 명령문을 실행합니다.

주: 제공된 스키마 아래에 하나 이상의 프로시저를 동시에 설치할 수 없습니다.

이 명령문을 실행하면 다른 인터페이스를 사용하여 루틴 정의 명령문을 실행하는 것과 같은 오류가 발생할 수 있습니다. 루틴 정의 처리 중에, 공유 라이브러리와 바인드 파일

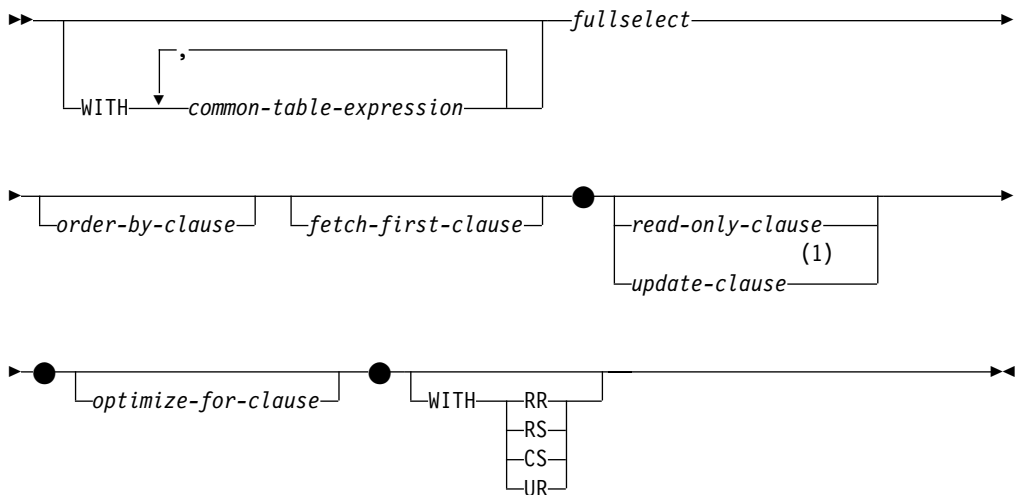
의 존재가 인식되고 사전 처리 컴파일, 컴파일 및 링크 단계가 생략됩니다. 바인드 파일은 바인드 처리 중에 사용되고 두 파일의 내용은 SQL 루틴을 위한 보통 라이브러리에 복사됩니다.

주: GET ROUTINE 또는 PUT ROUTINE 조작(또는 해당 프로시저)가 성공적으로 실행되지 못하면, 항상 실패 원인에 대한 정보를 제공하는 진단 텍스트와 함께 오류(SQLSTATE 38000)를 리턴합니다. 예를 들어, GET ROUTINE에 대해 제공된 프로시저 이름이 SQL 프로시저를 식별하지 않으면, 진단 "100" 및 "02000"가 리턴되며, 여기서 "100" 및 "02000"은 각각 SQLCODE 및 SQLSTATE로 문제점의 원인을 식별합니다. 이 예에서 SQLCODE 및 SQLSTATE는 주어진 프로시저 이름에 지정된 행을 카탈로그 테이블에서 찾지 못했음을 나타냅니다.

조회

select-statement/구문 도표

구문 도표는 다음과 같이 변경됩니다.



주:

1 update-clause 및 order-by-clause은 둘다 동일한 select-statement에 지정될 수 없습니다.

다음과 같은 단락을 구문 도표 아래의 설명에 추가하십시오.

선택적 WITH 절은 select 문이 실행되는 분리 레벨을 지정합니다.

- RR - 반복 읽기
- RS - 읽기 안정성
- CS - 커서 안정성
- UR - 미확약 읽기

명령문의 기본 분리 레벨은 명령문이 바인드된 패키지의 분리 레벨입니다.

select-statement/fetch-first-clause

fetch-first-clause의 설명에서 다음의 마지막 문장은

SELECT 문에 있는 첫번째 페치 절의 스펙으로 인해 커서를 삭제할 수 없게 되었습니다(읽기 전용). 이 절은 FOR UPDATE 절과 함께 지정될 수 없습니다.

올바르지 않으므로 제거해야 합니다.

SQL문

파티션 키 갱신 지원

파티션 키 갱신이 이제 지원됩니다. DB2_UPDATE_PART_KEY=ON:일 경우에만 여러 명령문에서 다음 텍스트를 삭제해야 합니다.

주: DB2_UPDATE_PART_KEY=OFF인 경우, 제한사항은 여전히 적용됩니다.

명령문: ALTER TABLE

규칙

- 테이블의 파티션 키 컬럼을 갱신할 수 없습니다(SQLSTATE42997).
- 관계가 ON DELETE SET NULL(SQLSTATE 42997)로 정의되어 있을 때 파티션 키의 널(NULL) 입력 가능 컬럼은 외부 키 컬럼으로 포함될 수 없습니다.

명령문: CREATE TABLE

규칙

- 테이블의 파티션 키 컬럼을 갱신할 수 없습니다(SQLSTATE42997).
- 관계가 ON DELETE SET NULL(SQLSTATE 42997)로 정의되어 있을 때 파티션 키의 널(NULL) 입력 가능 컬럼은 외부 키 컬럼으로 포함될 수 없습니다.

명령문: DECLARE GLOBAL TEMPORARY TABLE PARTITIONING KEY (column-name,...)

주: 파티션 키 컬럼을 갱신할 수 없습니다(SQLSTATE 42997).

명령문: UPDATE

주석

- 108 파티션 키의 컬럼을 갱신할 수 없습니다(SQLSTATE 42997). 파티션 키에서 컬럼을 변경하려면 데이터 행을 삭제하고 삽입해야 합니다.

유니코드 데이터베이스의 더 큰 색인 키

ALTER TABLE

1차 및 고유 키를 포함하는 색인의 일부인 변수 길이 컬럼의 길이는 레지스트리 변수 DB2_INDEX_2BYTEVARLEN이 설정되었을 때 정의되며, 255 바이트보다 긴 길이로 변경될 수 있습니다. 가변 길이 컬럼이 외부 키와 관련되어 있다는 사실은 레지스트리 변수 설정과 관계없이 더 이상 해당 컬럼의 길이가 255 바이트보다 더 크게 변경되지 않도록 하지는 못합니다. 그러나, 255보다 큰 길이는 해당 기본 키에 있는 컬럼이 255 바이트보다 큰 길이를 가지고 있는 경우를 제외하고는 테이블로 삽입될 수 없습니다. 이것은 기본 키가 레지스트리 변수 ON으로 작성된 경우에만 가능합니다.

CREATE INDEX

레지스트리 변수 DB2_INDEX_2BYTEVARLEN이 ON인 경우 길이가 255 바이트보다 큰 변수 길이 컬럼에 색인이 정의될 수 있습니다.

CREATE TABLE

변수 keypart가 있는 1차 및 고유 키는 레지스트리 변수 DB2_INDEX_2BYTEVARLEN이 ON인 경우 255보다 큰 크기를 가질 수 있습니다. 외부 키는 길이가 255 바이트보다 큰 변수 길이 컬럼에 정의될 수 있습니다.

ALTER SEQUENCE

ALTER SEQUENCE

ALTER SEQUENCE 문은 다음 기준으로 순차 속성을 수정합니다.

- 순차 재시작
- 장래 순차 값 사이의 증분 변경
- 새로운 최소 또는 최대 값 설정
- 캐쉬된 순차 번호 수 변경
- 순차가 순환될 수 있는지 여부 변경
- 요청 순서 대로 순차 번호가 생성되어야 하는지 여부 변경

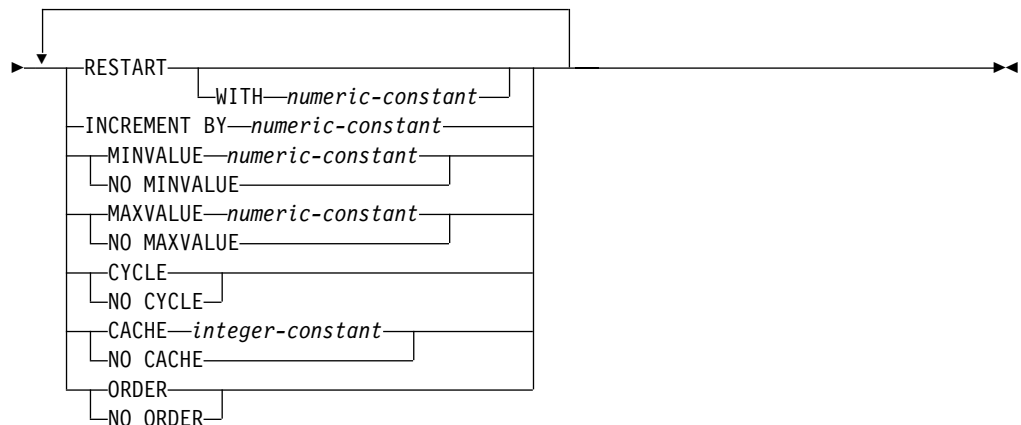
호출: 이 명령문은 응용프로그램에 추가되거나 동적 SQL문의 사용을 통해 발행될 수 있습니다. 동적으로 준비될 수 있는 실행 명령문입니다. 그러나, 바인드 옵션 DYNAMICRULES BIND가 적용되는 경우, 명령문은 동적으로 준비될 수 없습니다 (SQLSTATE 42509).

권한 부여: 명령문의 권한 부여 ID가 보유한 특권은 다음 중 하나를 포함해야 합니다.

- 순차 정의자
- 내재적으로 또는 명시적으로 지정된 스키마의 ALTERIN 특권
- SYSADM 또는 DBADM 권한

구문:

▶▶ ALTER SEQUENCE *sequence-name* →



설명:

sequence-name

특정 순차를 식별합니다. 이름의 조합, 내재적이거나 명시적 스키마 이름은 현재 서

버에서 기존 순차를 식별해야 합니다. 명시적으로나 내재적으로 지정된 스키마 이름의 순차가 없다면, 오류(SQLSTATE 42704)가 발생합니다.

RESTART

순차를 재시작합니다. *numeric-constant*가 지정되지 않은 경우, 원래 순차를 작성한 CREATE SEQUENCE 문에서 시작 값으로서 내재적으로나 명시적으로 지정된 값으로 순차가 재시작됩니다.

WITH *numeric-constant*

지정된 값으로 순차를 재시작합니다. 이 값은 소수점의 오른쪽(SQLSTATE 42894)에 0이 아닌 숫자가 없는 한, 순차(SQLSTATE 42820)와 연관된 데이터 유형의 컬럼에 지정될 수 있는 양 또는 음의 값일 수 있습니다.

INCREMENT BY

연속된 순차 값 사이의 간격을 지정합니다. 이 값은 소수점의 오른쪽(SQLSTATE 428FA)에 0이 아닌 숫자가 없이, 순차(SQLSTATE 42820)와 연관된 데이터 유형의 컬럼에 지정될 수 있는 양 또는 음의 값일 수 있으며 큰 정수 상수(SQLSTATE 42815)의 값을 초과하지 않습니다.

이 값이 음수이면, 값의 순차는 내림차순입니다. 이 값이 양수이면, 값의 순차는 오름차순입니다. 이 값이 0이거나 MINVALUE 및 MAXVALUE가 정의한 범위보다 크면, 오직 하나의 값이 생성되지만, 그렇지 않으면 순차는 오름차순 순차로서 처리됩니다.

MINVALUE 또는 NO MINVALUE

내림차순 순차가 값 생성을 순환 또는 중지하거나, 오름차순 순차가 최대 값에 도달한 이후에 순환하는 최소 값을 지정합니다.

MINVALUE *numeric-constant*

최소 값인 숫자 상수를 지정합니다. 이 값은 소수점의 오른쪽(SQLSTATE 428FA)에 0이 아닌 숫자가 없이, 순차(SQLSTATE 42820)와 연관된 데이터 유형의 컬럼에 지정될 수 있는 양 또는 음의 값일 수 있으나 값은 최대 값(SQLSTATE 42815) 이하이어야 합니다.

NO MINVALUE

오름차순 순차의 경우, 값은 START WITH 값이거나, START WITH가 지정되지 않은 경우 1입니다. 내림차순 순차의 경우, 값은 순차와 연관된 데이터 유형의 최소 값입니다. 이 값이 기본값입니다.

MAXVALUE 또는 NO MAXVALUE

오름차순 순차가 값 생성을 순환 또는 중지하거나, 내림차순 순차가 최소 값에 도달한 이후에 순환하는 최대 값을 지정합니다.

MAXVALUE *numeric-constant*

최대 값인 숫자 상수를 지정합니다. 이 값은 소수점의 오른쪽(SQLSTATE 428FA)에 0이 아닌 숫자가 없이, 순차(SQLSTATE 42820)와 연관된 데이터

유형의 컬럼에 지정될 수 있는 양 또는 음의 값일 수 있으나 값은 최소 값 (SQLSTATE 42815) 이상이어야 합니다.

NO MAXVALUE

오름차순 순차의 경우, 값은 순차와 연관된 데이터 유형의 최대 값입니다. 내림차순 순차의 경우, 값은 START WITH 값이거나, START WITH가 지정되지 않은 경우 -1입니다. 이 값이 기본값입니다.

CYCLE 또는 NO CYCLE

최대 또는 최소 값 중 하나에 도달한 이후에 순차가 값을 계속 생성해야 하는가를 지정합니다. 다음 값이 범위 조건에 정확히 맞도록 또는 순환이 허용될 경우 START WITH 값으로 돌아가 다음 값이 결정되는 경우에 범위 조건을 지나쳐 순차의 범위에 도달할 수 있습니다.

CYCLE

최대 또는 최소 값에 도달한 후에 이 순차에 대해 계속 값이 생성되도록 지정합니다. 옵션이 사용되면, 오름차순 순차가 최대 값에 도달한 후에 최소값을 생성하며, 내림차순 순차가 최소 값에 도달한 후에 최대 값을 생성합니다. 순차에 대한 최대 및 최소 값이 순환에 사용되는 범위를 결정합니다.

CYCLE이 유효하면, 중복된 값이 순차에 대해 생성될 수 있습니다.

NO CYCLE

순차에 대한 최대 또는 최소 값에 도달하면 순차에 대한 값이 생성되지 않음을 지정합니다. 이 값이 기본값입니다.

CACHE 또는 NO CACHE

더 빠른 액세스를 위해 메모리에 일부 사전할당된 값을 유지할 지 여부를 지정합니다. 이것은 성능 및 조정 옵션입니다.

CACHE integer-constant

사전할당되어 메모리에 유지된 순차의 최대 수를 지정합니다. 캐쉬에서 값을 사전할당하고 저장하면 순차에 대한 값이 생성될 때 로그에 대한 동기 입출력을 감소시킵니다.

시스템 실패시, 예약된 명령문에서 사용되지 않은 캐쉬된 모든 순차 값은 유실됩니다(즉, 결코 사용되지 않습니다). CACHE 옵션에 대해 지정된 값은 시스템 실패시 유실될 수 있는 순차 값의 최대 수입니다.

최소값은 2(SQLSTATE 42815)입니다. 기본값은 CACHE 20입니다.

NO CACHE

순차의 값은 사전할당되지 않음을 지정합니다. 시스템 실패, 종료 또는 데이터베이스 비활성화의 경우에 값이 유실되지 않습니다. 이 옵션이 지정되면, 순차 값은 캐쉬에 저장되지 않습니다. 이 경우, 순차의 새 값에 대한 모든 요청은 로그에 동기 입출력되는 결과를 가져옵니다.

NO ORDER 또는 ORDER

순차 번호가 요청 순서대로 생성될 지 여부를 지정합니다.

ORDER

순차 번호가 요청 순서대로 생성되도록 지정합니다.

NO ORDER

순차 번호가 요청 순서대로 생성될 필요가 없음을 지정합니다. 이 값이 기본값입니다.

순차를 재시작하거나 CYCLE로 변경한 후에, 순차 번호가 이전에 순차에 의해 생성된 순차 번호들의 중복된 값일 수 있습니다.

참고:

- 장래 순차 번호만이 ALTER SEQUENCE 문에 의해 영향을 받습니다.
- 순차의 데이터 유형은 변경될 수 없습니다. 대신, 새 순차에 대해 원하는 데이터 유형을 지정하는 순차를 삭제하고 재작성하십시오.
- 모든 캐시 값은 순차가 변경될 때 상실됩니다.

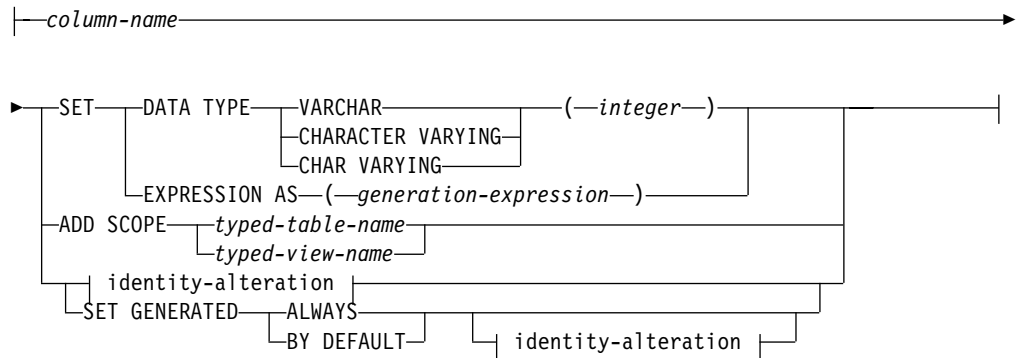
예: 예 1: 숫자 값 없이 RESTART를 지정하는 것이 가능한 이유는 순차를 START WITH 값으로 재설정하는 것입니다. 이 예에서, 목표는 1에서 테이블의 행 수까지 생성하는 것이며 그런 다음 임시 테이블을 사용하여 테이블에 추가된 컬럼으로 숫자를 삽입합니다. 또다른 용도는 모든 결과 행의 번호가 지정되는 결과를 얻는 것입니다.

```
ALTER SEQUENCE org_seq
RESTART
SELECT NEXTVAL for org_seq, org.*
FROM org
```

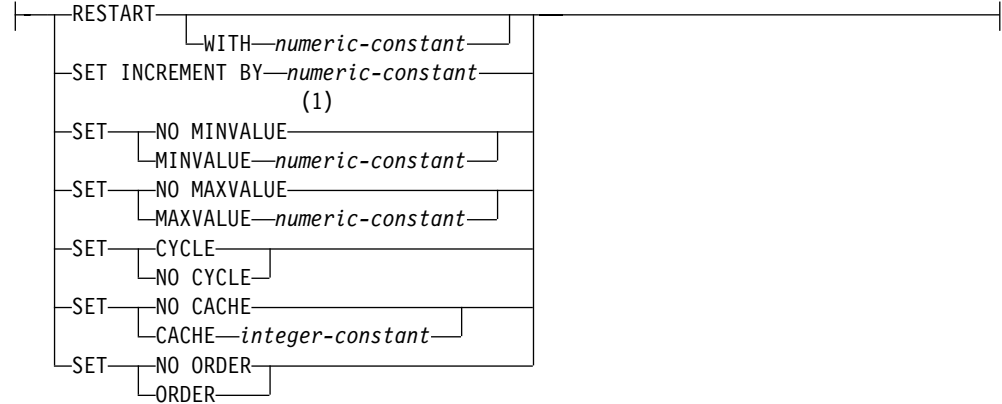
ALTER TABLE

구문 단편화에 대한 변경사항:

column-alteration:



identity-alteration:



주:

- 1 다음 매개변수는 공백없이 지정될 수 있습니다. NOMINVALUE, NOMAXVALUE, NOCYCLE, NOCACHE 및 NOORDER. 이 단일 단어 버전은 두 개의 단어 버전에 대해 모두 채택 가능한 대안입니다.

다음 매개변수를 추가하십시오.

SET GENERATED

값이 컬럼에 대해 항상 생성될 지 아니면 기본값이 필요할 때에만 생성될 지 여부를 지정합니다.

ALWAYS

행이 테이블에 삽입되거나 갱신될 때 컬럼에 대한 값이 항상 생성됩니다. 컬럼은 이미 생성된 컬럼으로서 정의되어 있어야 합니다(SQLSTATE 42837).

BY DEFAULT

값이 지정된 경우를 제외하고는, 행이 테이블에 삽입될 때 컬럼에 대한 값이 생성됩니다. 컬럼은 이미 생성된 컬럼으로서 정의되어 있어야 합니다(SQLSTATE 42837).

RESTART 또는 RESTART WITH numeric-constant

Identity 컬럼과 연관된 순차의 상태를 재설정합니다. WITH numeric-constant 가 지정되지 않으면, 식별 컬럼이 원래 작성되었을 때 시작값으로서 내재적으로 또는 명시적으로 지정된 값에서 식별 컬럼에 대한 순차가 재시작됩니다. numeric-constant는 소수점 오른쪽(SQLSTATE 42894)에 0이 아닌 숫자가 없는 한 이 컬럼(SQLSTATE 42820)에 지정될 수 있는 양수 또는 음수 값의 정확한 숫자 상수입니다. 컬럼은 IDENTITY 속성으로 미리 정의되어 있어야 합니다(SQLSTATE 42837). numeric-constant는 컬럼에 대한 다음 값으로 사용됩니다.

SET INCREMENT BY numeric-constant

연속된 식별 컬럼 값 사이의 간격을 지정합니다. 컬럼은 IDENTITY 속성으로

미리 정의되어 있어야 합니다(SQLSTATE 42837). 이 값은 소수점의 오른쪽(SQLSTATE 428FA)에 0이 아닌 숫자가 없는 한, 이 컬럼(SQLSTATE 42820)에 지정될 수 있고 큰 정수 상수(SQLSTATE 42815)의 값을 초과하지 않는 양 또는 음의 값일 수 있습니다(SQLSTATE 42894).

이 값이 음수이면, 이 식별 컬럼의 값 순차는 내림차순입니다. 이 값이 양수이면, 이 식별 컬럼의 값 순차는 오름차순입니다. 이 값이 0이거나 MINVALUE 및 MAXVALUE가 정의한 범위보다 크면, 오직 하나의 값이 생성되지만, 그렇지 않으면 순차는 오름차순 순차로서 처리됩니다.

SET MINVALUE *numeric-constant* 또는 NO MINVALUE

내림차순 식별 컬럼이 값 생성을 순환 또는 중지하거나, 오름차순 식별 컬럼이 최대 값에 도달한 이후에 순환하는 최소 값을 지정합니다. 컬럼은 IDENTITY 속성으로 미리 정의되어 있어야 합니다(SQLSTATE 42837).

MINVALUE *numeric-constant*

최소 숫자 상수 값을 지정합니다. 이 값은 소수점의 오른쪽(SQLSTATE 28F92)에 0이 아닌 숫자가 없이, 이 컬럼(SQLSTATE 42820)에 지정될 수 있는 양 또는 음의 값일 수 있으나 값은 최대 값(SQLSTATE 42815)보다 작아야 합니다.

NO MINVALUE

오름차순 순차의 경우, 값은 START WITH 값이거나, START WITH가 지정되지 않은 경우 1입니다. 내림차순 순차의 경우, 값은 컬럼에 대한 데이터 유형의 최소 값입니다.

SET MAXVALUE *numeric-constant* 또는 NO MAXVALUE

오름차순 식별 컬럼이 값 생성을 순환 또는 중지하거나, 내림차순 식별 컬럼이 최소 값에 도달한 이후에 순환하는 최대 값을 지정합니다. 컬럼은 IDENTITY 속성으로 미리 정의되어 있어야 합니다(SQLSTATE 42837).

MAXVALUE *numeric-constant*

최대 값인 숫자 상수를 지정합니다. 이 값은 소수점의 오른쪽(SQLSTATE 28F92)에 0이 아닌 숫자가 없이, 이 컬럼(SQLSTATE 42820)에 지정될 수 있는 양 또는 음의 값일 수 있으나 값은 최소 값(SQLSTATE 42815)보다 커야 합니다.

NO MAXVALUE

오름차순 순차의 경우, 값은 컬럼에 대한 데이터 유형의 최대 값입니다. 내림차순 순차의 경우, 값은 START WITH 값이거나, START WITH가 지정되지 않은 경우 -1입니다.

SET CYCLE 또는 NO CYCLE

식별 컬럼이 최대 또는 최소 값 중 하나를 생성하는 값을 계속 생성해야 하는가를 지정합니다. 컬럼은 IDENTITY 속성으로 미리 정의되어 있어야 합니다 (SQLSTATE 42837).

CYCLE

최대 또는 최소 값에 도달한 후에 이 컬럼에 대해 계속 값이 생성되도록 지정합니다. 옵션이 사용되면, 오름차순 식별 컬럼이 최대 값에 도달한 후에 최소값을 생성하며, 내림차순 순차가 최소 값에 도달한 후에 최대 값을 생성합니다. 식별 컬럼에 대한 최대 및 최소 값은 순환에 사용되는 범위를 결정합니다.

CYCLE이 유효하면, 중복된 값이 식별 컬럼에 대해 생성될 수 있습니다. 필수적이지는 않지만, 고유 값이 필요한 경우, 식별 컬럼을 사용하여 정의된 단일 컬럼 고유의 색인이 고유성을 보장합니다. 고유한 색인이 그러한 식별 컬럼에 존재하고 고유하지 않은 값이 생성되면, 오류가 발생합니다 (SQLSTATE 23505).

NO CYCLE

최대 또는 최소 값에 도달되면 식별 컬럼에 대한 값이 생성되지 않음을 지정합니다.

SET CACHE *integer-constant* 또는 NO CACHE

더 빠른 액세스를 위해 메모리에 일부 사전할당된 값을 유지할 지 여부를 지정합니다. 이것은 성능 및 조정 옵션입니다. 컬럼은 IDENTITY 속성으로 미리 정의되어 있어야 합니다(SQLSTATE 42837).

CACHE *integer-constant*

얼마나 많은 식별 순차 값이 사전할당되어 메모리에 유지되는지 지정합니다. 식별 컬럼에 대한 값이 생성되면, 캐쉬에서 값을 사전할당하고 저장하면 로그에 대한 동기 입출력을 감소시킵니다.

식별 컬럼에 대한 새 값이 필요하며 캐쉬에서 사용 가능한 미사용 값이 없으면, 값의 할당을 위해 로그에 대한 입출력을 기다려야합니다. 그러나, 새 값이 식별 컬럼에 필요하며 캐쉬에 미사용 값이 있으면, 로그에 대한 입출력을 피함으로써 해당 식별 값의 할당이 더 빠르게 발생하게 할 수 있습니다.

데이터베이스 관리 프로그램이 중지되면(예: 데이터베이스 비활성화, 시스템 실패 또는 종료), 확장된 명령문에서 사용되지 않은 캐쉬된 모든 값은 유실됩니다(즉, 결코 사용되지 않습니다). CACHE 옵션에 대해 지정된 값은 시스템 실패시 유실될 수 있는 식별 컬럼에 대한 값의 최대 수입니다.

최소값은 2(SQLSTATE 42615)입니다.

NO CACHE

식별 컬럼의 값은 사전할당되지 않음을 지정합니다.

이 옵션이 지정되면, 식별 컬럼의 값은 캐쉬에 저장되지 않습니다. 이 경우, 새 식별 값에 대한 모든 요청이 로그에 동기 입출력되는 결과를 가져옵니다.

SET ORDER 또는 NO ORDER

식별 컬럼 값이 요청 순서대로 생성될 지 여부를 지정합니다. 컬럼은 IDENTITY 속성으로 미리 정의되어 있어야 합니다(SQLSTATE 42837).

ORDER

식별 컬럼이 요청 순서대로 생성되도록 지정합니다.

NO ORDER

식별 컬럼이 요청 순서대로 생성될 필요가 없음을 지정합니다.

복합 SQL(Embedded)

준비된 COMMIT 문은 최소단위 복합 SQL문에서 허용되지 않습니다.

복합 텍스트 명령문(동적)

복합 텍스트 명령문(동적)

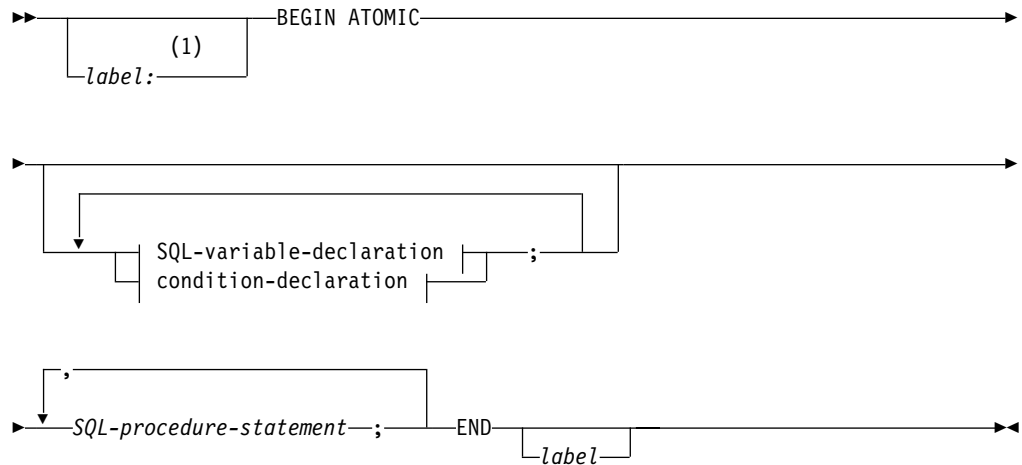
복합 텍스트 명령문은 다른 명령문들을 실행 블록으로 함께 그룹화합니다. 동적으로 준비된 최소단위 복합 텍스트문 내에서 SQL 변수를 선언할 수 있습니다.

호출: 이 명령문은 트리거, SQL 함수 또는 SQL 메소드에서 추가되거나 동적 SQL 문을 사용하여 발행될 수 있습니다. 동적으로 준비될 수 있는 실행 명령문입니다.

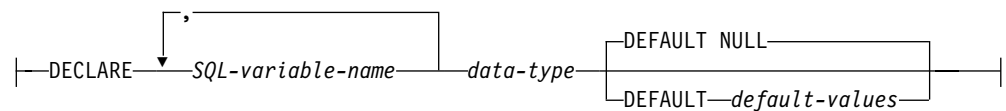
권한 부여: 동적 복합 텍스트 명령문을 호출하는 데 특권은 필요하지 않습니다. 그러나, 복합 텍스트 명령문의 권한 부여 ID는 복합 텍스트 명령문에 추가된 SQL문 호출에 필요한 특권을 보유해야 합니다.

구문:

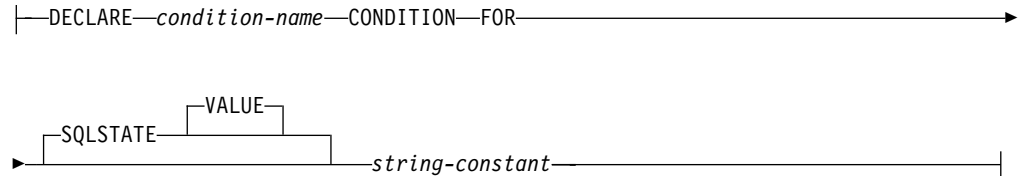
dynamic-compound-statement



SQL-variable-declaration:



condition-declaration:



주:

1 명령문이 함수, 메소드 또는 트리거 정의일 때에만 레이블을 지정할 수 있습니다.

설명:

label

코드 블록에 대한 레이블을 정의합니다. 시작 레이블이 지정되면, 동적 복합 텍스트 명령문에서 선언된 SQL 변수를 규정하는 데 사용될 수 있으며 LEAVE 문에 지정될 수 있습니다. 종료 레이블이 지정된 경우, 시작 레이블과 동일해야 합니다.

ATOMIC

오류가 복합 텍스트 명령문에서 발생한 경우, 복합 텍스트 명령문에 있는 모든 SQL 문이 롤백되며 복합 텍스트 명령문에서 나머지 모든 SQL 문은 처리되지 않습니다.

SQL-procedure-statement

다음과 같은 *SQL-control-statements* 목록은 동적 복합 텍스트 명령문 내에서 사용될 수 있습니다.

- FOR 문
- GET DIAGNOSTICS 문
- IF 문
- ITERATE 문
- LEAVE 문
- SIGNAL 문
- WHILE 문

발행될 수 있는 SQL 문은 다음과 같습니다.

- fullselect⁶
- 검색된 UPDATE
- 검색된 DELETE
- INSERT
- SET 변수 명령문

SQL-variable-declaration

동적 복합 텍스트 명령문에 지역적인 변수를 선언합니다.

SQL-variable-name

지역 변수의 이름을 정의합니다. DB2는 모든 SQL 변수 이름을 대문자로 변환합니다. 이름에는 다음 제한사항이 따릅니다.

- 동일한 복합 텍스트 문 내에서 또다른 SQL 변수와 같을 수 없습니다.
- 매개변수 이름과 같을 수 없습니다.
- 컬럼 이름과 같을 수 없습니다.

SQL 문에 SQL 변수 및 컬럼 참조와 동일한 이름을 가진 식별자가 들어 있는 경우, DB2는 식별자를 컬럼으로서 해석합니다.

6. common-table-expression은 fullselect를 선행할 수 있습니다.

data-type

변수의 데이터 유형을 지정합니다.

DEFAULT *default-values* 또는 **NULL**

SQL 변수의 기본값을 정의합니다. 동적 복합 텍스트 명령문이 호출될 때 변수가 초기화됩니다. 기본값이 지정되지 않으면, 변수는 NULL로 초기화됩니다.

condition-declaration

조건 이름 및 해당 SQLSTATE 값을 선언합니다.

condition-name

조건의 이름을 지정합니다. 조건 이름은 프로시저어 본문 내에서 고유해야 하며 선언된 복합 텍스트 명령문 내에서만 참조될 수 있습니다.

FOR SQLSTATE *string-constant*

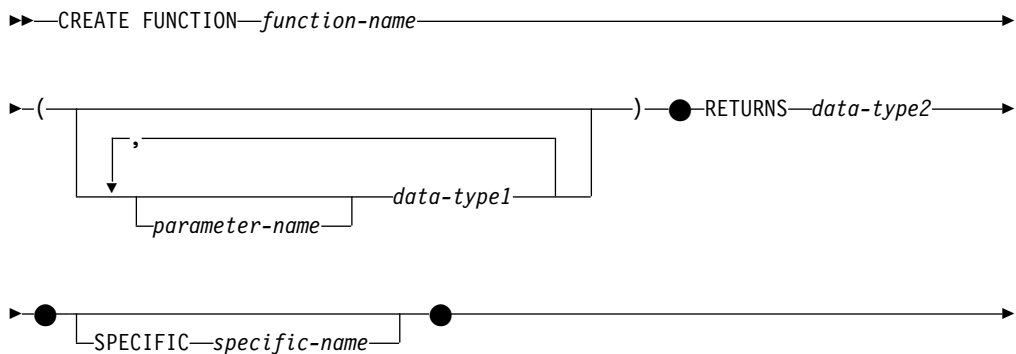
조건과 연관된 SQLSTATE를 지정합니다. *string-constant*는 작은 따옴표로 묶인 5문자로서 지정되어야 하며 '00000'일 수 없습니다.

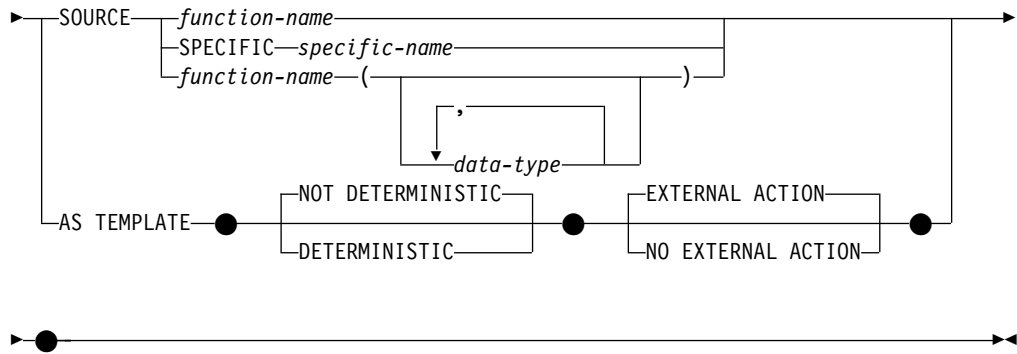
참고:

- 동적 복합 텍스트 명령문은 하나의 단일 명령문으로서 DB2에 의해 컴파일됩니다. 이 명령문은 작은 제어 흐름 논리이지만 중요한 데이터 흐름을 포함하는 짧은 스크립트에 효과적입니다. 중첩 복합 제어 흐름이 있는 큰 구성의 경우, SQL 프로시저어 사용에 대한 세부사항에 대해 SQL 프로시저어를 사용하는 것이 더 좋습니다.

CREATE FUNCTION(소스 또는 템플릿)

구문 도표는 다음과 같이 변경됩니다.





다음은 "설명" 섹션에 추가하십시오.

DETERMINISTIC 또는 NOT DETERMINISTIC

이 선택적 절은 주어진 인수 값에 대해 함수가 항상 동일한 결과를 리턴하는지 (DETERMINISTIC) 여부나 결과에 영향을 미치는 일부 상태 값에 의해 함수가 좌우되는지 (NOT DETERMINISTIC) 여부를 지정합니다. 즉, DETERMINISTIC 함수는 동일한 입력으로 연속 호출시 동일한 테이블을 항상 리턴해야 합니다. 동일한 입력이 항상 동일한 결과를 생성한다는 사실이 이점인 최적화는 NOT DETERMINISTIC을 지정하여 방지됩니다.

NOT DETERMINISTIC은 함수의 본문이 특수 레지스터를 액세스하거나 또다른 비결정 함수를 호출하는 경우 명시적으로나 내재적으로 지정되어야 합니다.

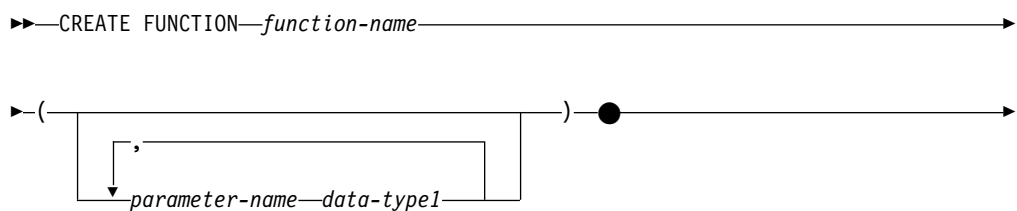
NO EXTERNAL ACTION 또는 EXTERNAL ACTION

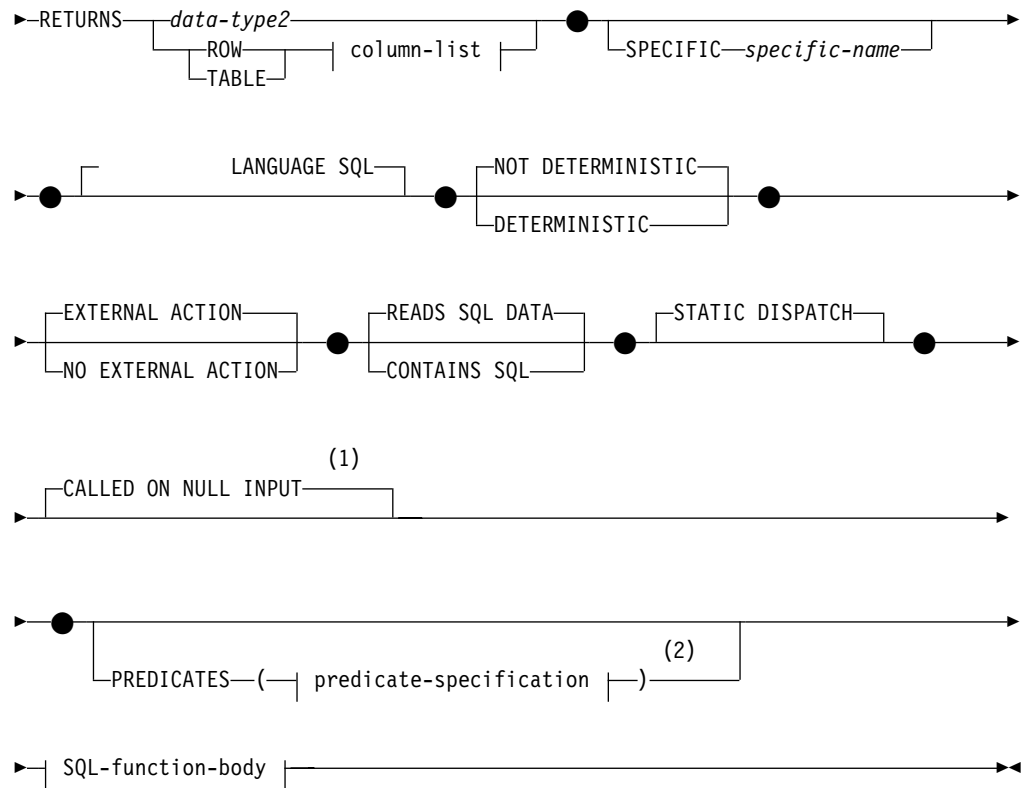
이 선택적 절은 데이터베이스 관리 프로그램이 관리하지 않는 오브젝트의 상태를 변경하는 일부 조치를 함수가 취할 지 여부를 지정합니다. NO EXTERNAL ACTION을 지정하여, 시스템은 함수가 외부 영향을 갖지 않음을 가정하는 특정 최적화를 사용할 수 있습니다.

함수의 본문이 외부 조치가 있는 또다른 함수를 호출하는 경우 EXTERNAL ACTION은 명시적으로 또는 내재적으로 지정되어야 합니다(SQLSTATE 428C2).

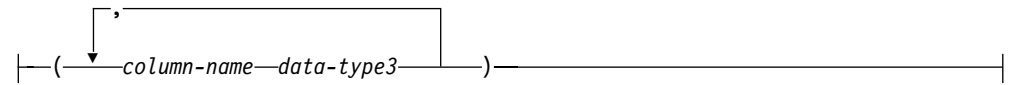
CREATE FUNCTION(SQL 스칼라, 테이블 또는 행)

구문 도표는 다음과 같이 변경됩니다.

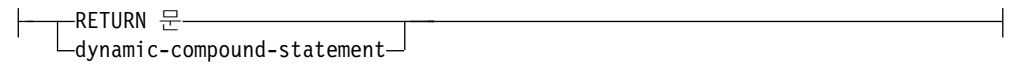




column-list:



SQL-function-body:



주:

- 1 NULL CALL은 CALLED ON NULL INPUT 대신 지정될 수 있습니다.
- 2 RETURNS가 스칼라 결과를 지정하는 경우에만 유효합니다(data-type2).

다음 매개변수를 변경하십시오.

LANGUAGE SQL

SQL을 사용하여 함수가 작성됨을 지정합니다.

이 매개변수는 "RETURN 표현식, NULL, WITH common-table-expression, fullselect" 매개변수 섹션을 바꿉니다.

SQL-function-body

함수의 본문을 지정합니다. 매개변수 이름은 SQL-function-body에서 참조될 수 있습니다. 매개변수 이름은 불명확한 참조를 피하기 위해 함수 이름으로 규정화될 수 있습니다.

SQL-function-body가 동적 복합 텍스트 명령문인 경우, 최소한 하나의 RETURN 문을 포함해야 하며 함수가 호출될 때 RETURN 문이 실행되어야 합니다 (SQLSTATE 42632). 함수가 테이블 또는 행 함수인 경우, 동적 복합 텍스트에 있는 마지막 명령문인 오직 하나의 RETURN 문만을 포함할 수 있습니다(SQLSTATE 429BD).

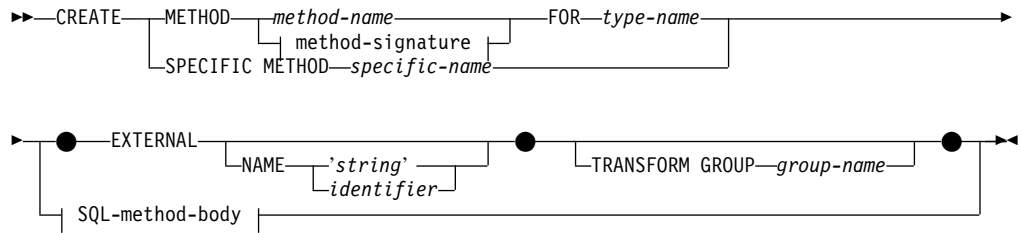
자세한 정보는 455 페이지의 『복합 텍스트 명령문(동적)』 및 488 페이지의 『RETURN』을 참조하십시오.

CREATE METHOD

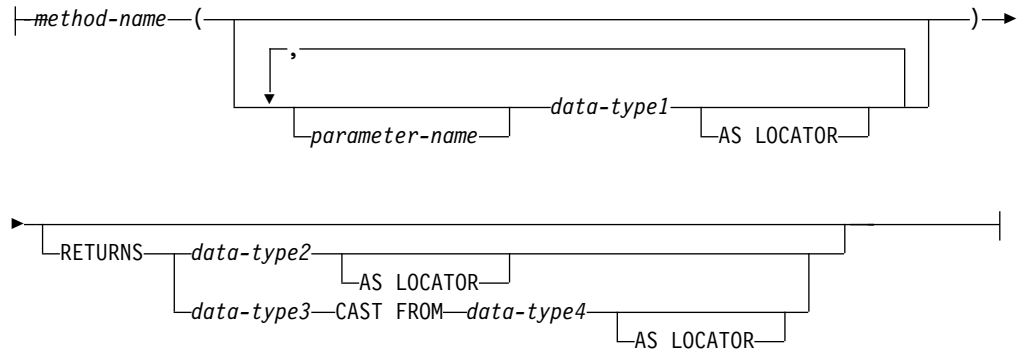
구문 도표는 다음과 같이 변경됩니다.

CREATE METHOD

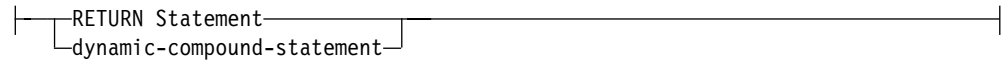
구문:



method-signature:



SQL-method-body:



다음 매개변수는 "RETURN scalar-expression 또는 NULL" 섹션을 바꿉니다.

SQL-method-body

SQL-method-body는 CREATE TYPE에 있는 메소드 스펙이 LANGUAGE SQL 인 경우 메소드가 구현되는 방식을 정의합니다.

SQL-method-body는 다음과 같은 메소드 스펙 부분과 일치해야 합니다.

- DETERMINISTIC 또는 NOT DETERMINISTIC(SQLSTATE 428C2)
- EXTERNAL ACTION 또는 NO EXTERNAL ACTION(SQLSTATE 428C2)
- CONTAINS SQL 또는 READS SQL DATA(SQLSTATE 42985)

매개변수 이름은 SQL-method-body에서 참조될 수 있습니다. 메소드의 주제는 SELF 라는 내재된 첫번째 매개변수로서 메소드 구현에 전달됩니다.

자세한 정보는 455 페이지의 『복합 텍스트 명령문(동적)』 및 488 페이지의 『RETURN』을 참조하십시오.

CREATE SEQUENCE

CREATE SEQUENCE

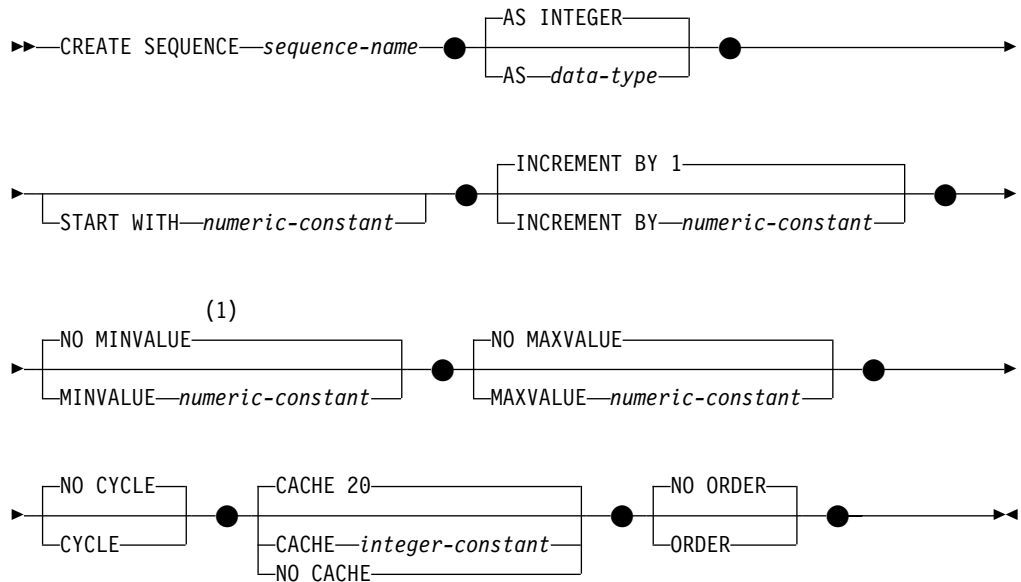
CREATE SEQUENCE 문은 응용프로그램 서버(AS)에서 순차를 작성합니다.

호출: 이 명령문은 응용프로그램에 추가되거나 동적 SQL문의 사용을 통해 발행될 수 있습니다. 동적으로 준비될 수 있는 실행 명령문입니다. 그러나, 바인드 옵션 DYNAMICRULES BIND가 적용되는 경우, 명령문은 동적으로 준비될 수 없습니다 (SQLSTATE 42509).

권한 부여: 명령문의 권한 부여 ID가 보유한 특권은 다음 중 하나를 포함해야 합니다.

- 내재적으로 또는 명시적으로 지정된 스키마에 대한 CREATEIN 특권
- SYSADM 또는 DBADM 권한

구문:



주:

- 1 다음 매개변수는 공백없이 지정될 수 있습니다. NOMINVALUE, NOMAXVALUE, NOCYCLE, NOCACHE 및 NOORDER. 이 단일 단어 버전은 두 개의 단어 버전에 대해 모두 채택 가능한 대안입니다.

설명:

sequence-name

순차의 이름을 지정합니다. 이름의 조합, 내재적이거나 명시적 스키마 이름은 현재 서버에서 기존 순차를 식별하지 말아야 합니다(SQLSTATE 42710).

*sequence-name*의 비규정 양식은 SQL 식별자입니다. 규정 양식은 규정자 다음에 마침표 그리고 SQL식별자가 뒤에 옵니다. 규정자는 스키마 이름입니다.

순차 이름이 스키마 이름으로 명시적으로 규정되면, 스키마 이름은 'SYS'로 시작할 수 없으며 오류(SQLSTATE 42939)가 발생합니다.

AS *data-type*

순차 값에 사용되는 데이터 유형을 지정합니다. 데이터 유형은 0 스케일을 갖는 정확한 숫자 유형(SMALLINT, INTEGER, BIGINT 또는 DECIMAL)이거나 소스 유형이 0 스케일을 갖는 완전한 숫자 유형인 사용자 정의 구별 유형일 수 있습니다(SQLSTATE 42815). 기본값은 INTEGER입니다.

START WITH *numeric-constant*

순차의 첫번째 값을 지정합니다. 이 값은 소수점의 오른쪽(SQLSTATE 428FA)에 0이 아닌 숫자가 없이, 순차(SQLSTATE 42820)와 연관된 데이터 유형의 컬럼에 지정될 수 있는 양 또는 음의 값일 수 있습니다. 기본값은 오름차순 순차에 대해 MINVALUE이며 내림차순 순차에 대해 MAXVALUE입니다.

이 값은 반드시 순차의 최대 또는 최소 값에 도달한 이후에 순차가 순환될 값을 아닙니다. START WITH 절은 순환에 사용되는 범위 외부에 있는 순차를 시작하는데 사용될 수 있습니다. 순환에 사용되는 범위는 MINVALUE 및 MAXVALUE에 의해 정의됩니다.

INCREMENT BY *numeric-constant*

연속된 순차 값 사이의 간격을 지정합니다. 이 값은 소수점의 오른쪽(SQLSTATE 428FA)에 0이 아닌 숫자가 없이, 순차(SQLSTATE 42820)와 연관된 데이터 유형의 컬럼에 지정될 수 있는 양 또는 음의 값일 수 있으며 큰 정수 상수(SQLSTATE 42815)의 값을 초과하지 않습니다.

이 값이 음수이면, 값의 순차는 내림차순입니다. 이 값이 양수이면, 값의 순차는 오름차순입니다. 이 값이 0이거나 MINVALUE 및 MAXVALUE가 정의한 범위보다 크면, 오직 하나의 값이 생성되지만, 그렇지 않으면 순차는 오름차순 순차로서 처리됩니다. 기본값은 1입니다.

MINVALUE 또는 **NO MINVALUE**

내림차순 순차가 값 생성을 순환 또는 중지하거나, 오름차순 순차가 최대 값에 도달한 이후에 순환하는 최소 값을 지정합니다.

MINVALUE *numeric-constant*

최소 값인 숫자 상수를 지정합니다. 이 값은 소수점의 오른쪽(SQLSTATE 428FA)에 0이 아닌 숫자가 없이, 순차(SQLSTATE 42820)와 연관된 데이터 유형의 컬럼에 지정될 수 있는 양 또는 음의 값일 수 있으나 값은 최대 값(SQLSTATE 42815) 이하이어야 합니다.

NO MINVALUE

오름차순 순차의 경우, 값은 START WITH 값이거나, START WITH가 지정되지 않은 경우 1입니다. 내림차순 순차의 경우, 값은 순차와 연관된 데이터 유형의 최소 값입니다. 이 값이 기본값입니다.

MAXVALUE 또는 NO MAXVALUE

오름차순 순차가 값 생성을 순환 또는 중지하거나, 내림차순 순차가 최소 값에 도달한 이후에 순환하는 최대 값을 지정합니다.

MAXVALUE *numeric-constant*

최대 값인 숫자 상수를 지정합니다. 이 값은 소수점의 오른쪽(SQLSTATE 428FA)에 0이 아닌 숫자가 없이, 순차(SQLSTATE 42820)와 연관된 데이터 유형의 컬럼에 지정될 수 있는 양 또는 음의 값일 수 있으나 값은 최소 값(SQLSTATE 42815) 이상이어야 합니다.

NO MAXVALUE

오름차순 순차의 경우, 값은 순차와 연관된 데이터 유형의 최대 값입니다. 내림차순 순차의 경우, 값은 START WITH 값이거나, START WITH가 지정되지 않은 경우 -1입니다. 이 값이 기본값입니다.

CYCLE 또는 NO CYCLE

최대 또는 최소 값 중 하나에 도달한 이후에 순차가 값을 계속 생성해야 하는가를 지정합니다. 다음 값이 정확하게 범위 조건에 맞거나, 범위 조건을 초과시켜 순차 범위에 도달할 수 있습니다.

CYCLE

최대 또는 최소 값에 도달한 후에 이 순차에 대해 계속 값이 생성되도록 지정합니다. 옵션이 사용되면, 오름차순 순차가 최대 값에 도달한 후에 최소값을 생성하며, 내림차순 순차가 최소 값에 도달한 후에 최대 값을 생성합니다. 순차에 대한 최대 및 최소 값이 순환에 사용되는 범위를 결정합니다.

CYCLE이 유효하면, 중복된 값이 순차에 대해 생성될 수 있습니다.

NO CYCLE

순차에 대한 최대 또는 최소 값에 도달하면 순차에 대한 값이 생성되지 않음을 지정합니다. 이 값이 기본값입니다.

CACHE 또는 NO CACHE

더 빠른 액세스를 위해 메모리에 일부 사전할당된 값을 유지할 지 여부를 지정합니다. 이것은 성능 및 조정 옵션입니다.

CACHE *integer-constant*

사전할당되어 메모리에 유지된 순차의 최대 수를 지정합니다. 캐쉬에서 값을 사전할당하고 저장하면 순차에 대한 값이 생성될 때 로그에 대한 동기 입출력을 감소시킵니다.

시스템 실패시, 예약된 명령문에서 사용되지 않은 캐쉬된 모든 순차 값은 유실됩니다(즉, 결코 사용되지 않습니다). CACHE 옵션에 대해 지정된 값은 시스템 실패시 유실될 수 있는 순차 값의 최대 수입니다.

최소값은 2(SQLSTATE 42815)입니다. 기본값은 CACHE 20입니다.

NO CACHE

순차의 값은 사전할당되지 않음을 지정합니다. 시스템 실패, 종료 또는 데이터베이스 비활성화의 경우에 값이 유실되지 않습니다. 이 옵션이 지정되면, 순차 값은 캐쉬에 저장되지 않습니다. 이 경우, 순차의 새 값에 대한 모든 요청은 로그에 동기 입출력되는 결과를 가져옵니다.

NO ORDER 또는 ORDER

순차 번호가 요청 순서대로 생성될 지 여부를 지정합니다.

ORDER

순차 번호가 요청 순서대로 생성되도록 지정합니다.

NO ORDER

순차 번호가 요청 순서대로 생성될 필요가 없음을 지정합니다. 이 값이 기본값입니다.

참고:

- 항상 상수 값을 리턴하는 상수 순차를 정의하는 것이 가능합니다. 이것은 MINVALUE 또는 MAXVALUE에 대해 동일한 값을 지정하거나, 0의 INCREMENT 값을 지정하여 수행될 수 있습니다. 어느 경우에도, NEXTVAL가 두번 이상 동일한 값을 생성하도록 하려면, CYCLE이 지정되어야 합니다. 상수 순차는 숫자 전역 변수로서 사용될 수 있습니다. ALTER SEQUENCE는 상수 순차에 대해 생성될 값을 조정하는 데 사용될 수 있습니다.
- 순차는 수동으로, ALTER SEQUENCE 문을 사용하여 순환될 수 있습니다. NO CYCLE이 내재적으로 또는 명시적으로 지정된 경우, 순차는 ALTER SEQUENCE 문을 사용하여 재시작되거나 확장되어 순차에 대한 최대 또는 최소 값에 도달한 후에 값이 계속 생성되도록 할 수 있습니다.
- 순차 번호 캐싱은 순차 번호의 범위가 빠른 액세스를 위해 메모리에 유지될 수 있음을 나타냅니다. 캐쉬에서 다음 순차 번호를 할당할 수 있는 순차를 응용프로그램이 액세스할 때, 순차 번호 할당이 빠르게 발생할 수 있습니다. 그러나, 캐쉬에서 다음 순차 번호를 할당할 수 없는 순차를 응용프로그램이 액세스하는 경우, 순차 번호 할당을 위해 지속적인 저장영역에 입출력 조작을 기다려야 할 수 있습니다. 성능 및 응용프로그램 요구사항 tradeoff에 유의하여 CACHE의 값을 선택해야 합니다.
- 소유자는 새 순차에 대한 ALTER 및 USAGE 특권을 갖습니다. USAGE 특권만이 소유자에 의해 PUBLIC으로만 권한 부여됩니다.
- NOMINVALUE, NOMAXVALUE, NOCYCLE, NOCACHE, NOORDER와 같은 구문도 지원됩니다.

예: 예 1: org_seq라고 하는 순차 작성:

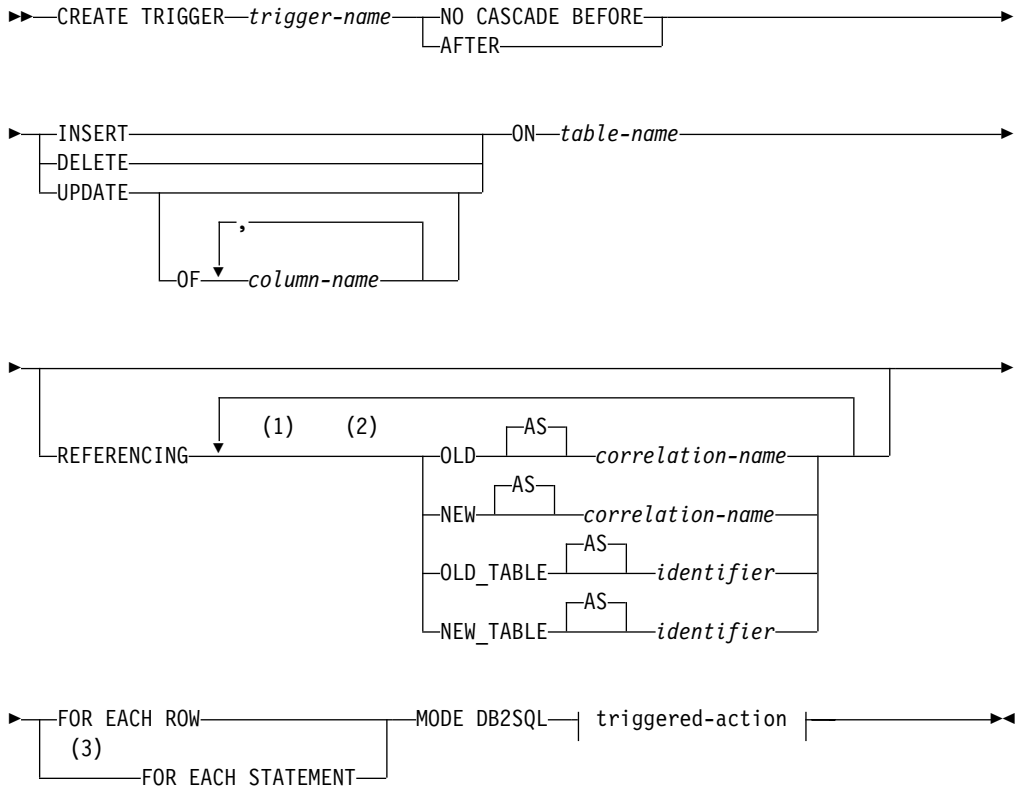
```
| CREATE SEQUENCE org_seq
| START WITH 1
| INCREMENT BY 1
| NO MAXVALUE
| NO CYCLE
| CACHE 24
```

| **CREATE TRIGGER**

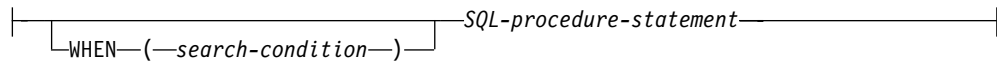
|

CREATE TRIGGER

구문:



triggered-action:



주:

- 1 OLD 및 NEW는 각각 한 번만 지정될 수 있습니다.
- 2 OLD_TABLE 및 NEW_TABLE은 각각 한 번만 AFTER 트리거에 대해서만 지정될 수 있습니다.
- 3 FOR EACH STATEMENT는 BEFORE 트리거에 대해서 지정될 수 없습니다.

"triggered-action"의 설명을 다음으로 바꾸십시오.

triggered-action

트리거가 활성화될 때 수행되는 조치를 지정합니다. triggered-action은 SQL-procedure-statement와 SQL-procedure-statement의 실행을 위한 선택적 조건으로 구성됩니다.

WHEN(*search-condition*)

참, 거짓 또는 알 수 없음 조건을 지정합니다. *search-condition*은 특정 트리거 조치가 실행되어야 하는지 여부를 결정하는 기능을 제공합니다.

지정된 검색 조건이 참으로 평가하는 경우에만 연관된 조치가 수행됩니다. WHEN 절이 누락되면, 연관된 *SQL-procedure statement*는 항상 수행됩니다.

SQL-procedure-statement

*SQL-procedure-statement*에는 동적 복합 텍스트 명령문이나 455 페이지의 『복합 텍스트 명령문(동적)』에 나열된 SQL 제어 명령문 중 하나가 들어 있을 수 있습니다.

트리거가 BEFORE 트리거인 경우, *SQL-procedure-statement*는 fullselect나 SET 변수 명령문을 포함할 수 있습니다(SQLSTATE 42987).

트리거가 AFTER 트리거인 경우, *SQL-procedure-statement*는 다음 중 하나를 포함할 수 있습니다(SQLSTATE 42987).

- INSERT SQL문
- 검색된 UPDATE SQL문
- 검색된 DELETE SQL문
- SET 변수 명령문
- fullselect ⁷

*SQL-procedure-statement*는 정의되지 않은 전이 변수 (SQLSTATE 42703)나 선언된 임시 테이블(SQLSTATE 42995)을 참조할 수 없습니다.

BEFORE 트리거에서 *SQL-procedure-statement*는 REFRESH IMMEDIATE으로 정의된 요약 테이블을 참조할 수 없습니다(SQLSTATE 42997).

BEFORE 트리거에서 *SQL-procedure-statement*는 새 전이 변수에 있는 식별 컬럼 이외에 생성된 컬럼을 참조할 수 없습니다(SQLSTATE 42989).

참고 섹션은 다음과 같이 변경됩니다.

- *SQL-procedure-statement*에 지정된 fullselect의 결과는 트리거 내부나 외부에서 사용 불가능합니다.
- **작동 불능 트리거:** 작동 불능 트리거는 더 이상 사용 가능하지 않으므로 활성화되지 못합니다. 다음 경우에 트리거가 작동 불능 상태가 됩니다.
 - 트리거 작성자가 트리거를 실행하는 데 필요한 특권이 취소됩니다.
 - 트리거된 조치가 종속되는 테이블, 뷰 또는 별명과 같은 오브젝트가 삭제됩니다.
 - 트리거된 조치가 종속적인 뷰가 작동 불능 상태가 됩니다.

7. common-table-expression은 fullselect를 선행할 수 있습니다.

- 트리거의 주제 테이블인 별명이 삭제됩니다.

실제적인 용어로, 작동 불능 트리거는 DROP 또는 REVOKE 문의 연쇄 규칙의 결과로서 트리거 정의가 삭제된 것입니다. 예를 들어, 뷰가 삭제되면, 해당 뷰를 사용하여 정의된 *SQL-procedure-statement*는 작동 불능이 됩니다.

트리거가 작동 불능이면, 트리거를 활성화 중이었던 조작을 수행하는 명령문을 가진 모든 패키지는 유효하지 않은 것으로 표시됩니다. 패키지가 리바인드되면(명시적으로 또는 내재적으로) 작동 불능 트리거가 완전히 무시됩니다. 마찬가지로, 트리거를 활성화 중이었던 응용프로그램을 수행하는 동적 SQL문을 가진 응용프로그램도 모든 작동 불능 트리거를 완전히 무시합니다.

트리거 이름은 여전히 DROP TRIGGER 및 COMMENT ON TRIGGER 문에서 지정될 수 있습니다.

작동 불능 트리거는 작동 불능 트리거의 정의 텍스트를 사용하여 CREATE TRIGGER 문을 발행하여 재작성될 수 있습니다. 이 트리거 정의 텍스트는 SYSCAT.TRIGGERS의 TEXT 컬럼에 저장됩니다. 작동 불능 트리거를 재작성하기 위해 명시적으로 삭제할 필요가 없음을 기억하십시오. 작동 불능 트리거와 동일한 *trigger-name*을 가진 CREATE TRIGGER 문을 발행하면 작동 불능 트리거가 경고로 바뀝니다(SQLSTATE 01595).

작동 불능 트리거는 SYSCAT.TRIGGERS 카탈로그 뷰의 VALID 컬럼에서 X에 의해 표시됩니다.

- **트리거 실행 오류:** 트리거된 SQL문 실행 중 발생한 오류는 심각한 오류가 아니면 SQLSTATE 09000을 사용하여 리턴됩니다. 오류가 심각한 경우, 심각한 오류 SQLSTATE가 리턴됩니다. 심각하지 않은 오류에 대한 SQLCA의 SQLERRMC 필드는 트리거 이름, SQLCODE, SQLSTAT와 실패 토큰에서 적합한 많은 토큰을 포함합니다.

*SQL-procedure-statement*는 SIGNAL SQLSTATE 문을 포함하거나 RAISE_ERROR 함수를 포함할 수 있습니다. 이 경우 모두에서, 리턴된 SQLSTATE는 SIGNAL SQLSTATE 문에 지정된 것이거나 RAISE_ERROR 조건입니다.

CREATE WRAPPER

Linux에서는 LIBDRDA.A 및 LIBSQLNET.A가 아닌 LIBDRDA.SO 및 LIBSQLNET.SO 라이브러리를 사용합니다.

DECLARE CURSOR

『DECLARE CURSOR』 문 내에서 주의사항 섹션의 끝에 있는 다음 문장이 아래와 같이 변경되어야 합니다.

An ambiguous cursor is considered read-only if the BLOCKING bind option is ALL, otherwise it is considered deletable.

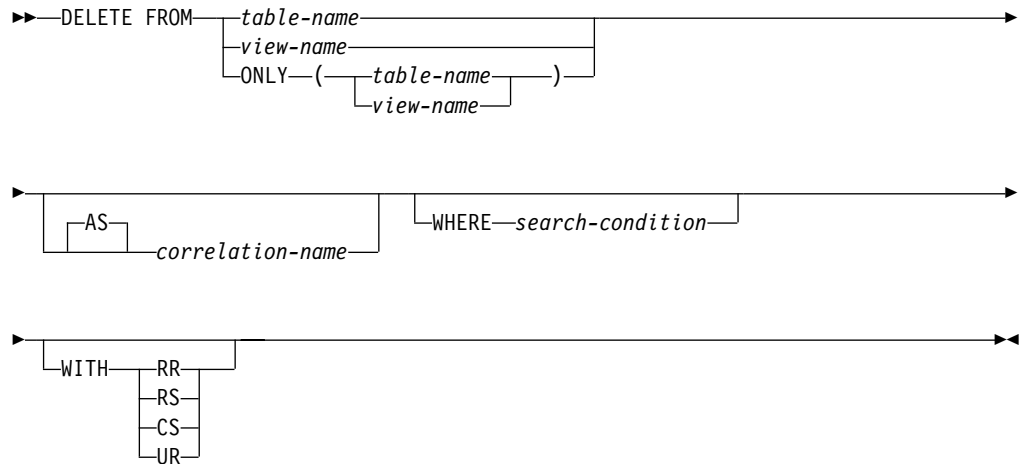
다음으로 변경하십시오.

An ambiguous cursor is considered read-only if the BLOCKING bind option is ALL; otherwise, it is considered updatable.

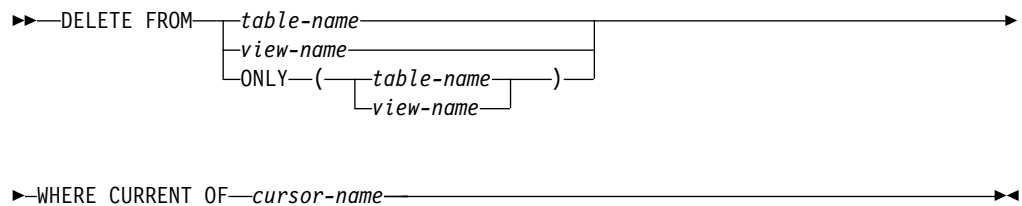
『deletable』이 『updatable』로 변경되었습니다.

DELETE

검색된 DELETE 구문 도표는 다음과 같이 변경됩니다.



위지지정된 DELETE:



다음을 설명 섹션에 추가하십시오.

WITH

삭제될 행을 찾을 때 사용되는 분리 레벨을 지정합니다.

RR

반복 읽기(RR)

RS

읽기 안정성(RS)

CS

커서 안정성(CS)

UR

미확약 읽기(UR)

명령문의 기본 분리 레벨은 명령문이 바인드된 패키지의 분리 레벨입니다.

DROP

다음 옵션을 추가하십시오.

```
►►—SEQUENCE—sequence-name—RESTRICT—◄◄
```

다음 매개변수를 추가하십시오.

SEQUENCE *sequence-name* RESTRICT

삭제되는 특정 순차를 식별합니다. 내재되거나 명시적 스키마 규정자와 함께 *sequence-name*은 현재 서버에서 기존 순차를 식별해야 합니다. 명시적으로나 내재적으로 지정된 스키마에 이 이름의 순차가 없다면, 오류(SQLSTATE42704)가 발생합니다.

RESTRICT 키워드는 테이블 컬럼의 정의가 순차를 참조하는 경우(IDENTITY 컬럼을 통해) 순차가 삭제되지 않는 규칙을 강화합니다.

주:

- IDENTITY 컬럼의 시스템 작성 순차는 DROP 순차 명령을 사용하여 삭제될 수 없습니다.
- 순차가 삭제되면, 순차에 대한 모든 특권도 삭제됩니다.

오브젝트가 서로 가지는 종속성을 표시하는 테이블은 다음과 같이 갱신될 필요가 있습니다(테이블 27).

새 행: DROP SEQUENCE 새 행 "DROP SEQUENCE" 및 컬럼 "PACKAGE"의 교차점에 있는 항목은 "A"입니다. 이 새 행에서의 나머지 항목은 "-"입니다

GRANT(순차 특권)

GRANT(순차 특권)

GRANT 문의 이 양식은 사용자 정의 순차에 특권을 권한 부여합니다.

호출: 이 명령문은 응용프로그램에 추가되거나 동적 SQL문의 사용을 통해 발행될 수 있습니다. 동적으로 준비될 수 있는 실행 명령문입니다. 그러나, 바인드 옵션 DYNAMICRULES BIND가 적용되는 경우, 명령문은 동적으로 준비될 수 없습니다 (SQLSTATE 42509).

권한 부여: 명령문의 권한 부여 ID가 보유한 특권은 다음 중 하나를 포함해야 합니다.

- 순차의 소유자
- SYSADM 또는 DBADM 권한

구문:

```
▶▶ GRANT USAGE ON SEQUENCE sequence-name TO PUBLIC
```

설명:

USAGE

순차의 USAGE 특권을 권한 부여합니다.

ON SEQUENCE *sequence-name*

USAGE 특권이 권한 부여되는 순차를 식별합니다. 내재되거나 명시적 스키마 규정자를 포함하여, *sequence-name*은 현재 서버에서 기존 순차를 고유하게 식별해야 합니다. 이 이름에 의한 순차가 지정된 스키마에 없다면, 오류(SQLSTATE 42704)가 발생합니다.

TO PUBLIC

모든 사용자에게 USAGE 특권을 권한 부여합니다.

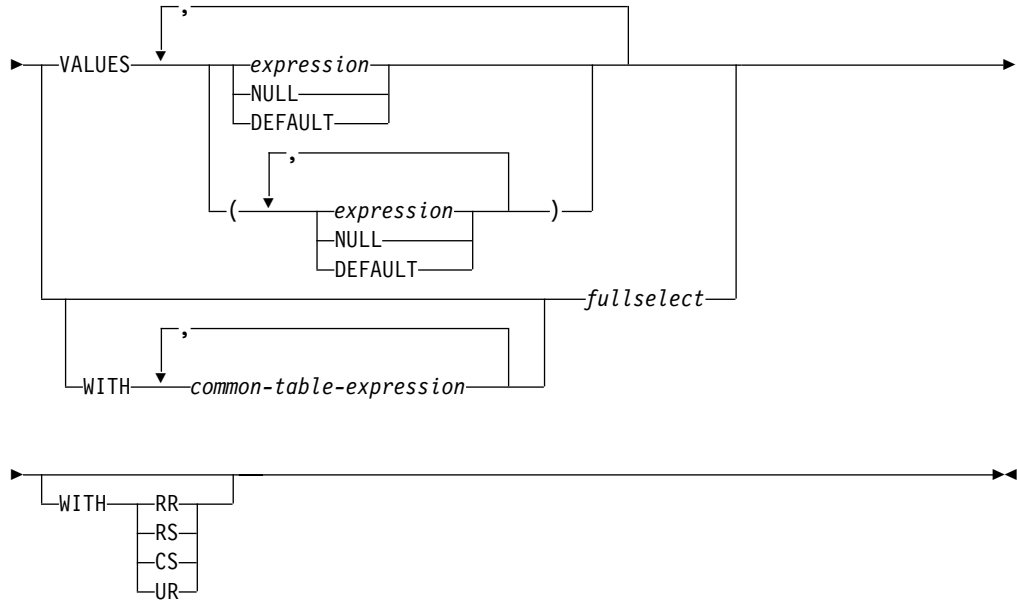
예: 예 1: MYNUM이라는 순차에서 임의 사용자에게 특권 부여

```
GRANT USAGE ON SEQUENCE MYNUM TO PUBLIC
```

INSERT

구문 도표는 다음과 같이 변경됩니다.

```
▶▶ INSERT INTO table-name | view-name ( column-name )
```



다음을 설명 섹션에 추가하십시오.

WITH

`fullselect`가 실행되는 분리 레벨을 지정합니다.

RR

반복 읽기(RR)

RS

읽기 안정성(RS)

CS

커서 안정성(CS)

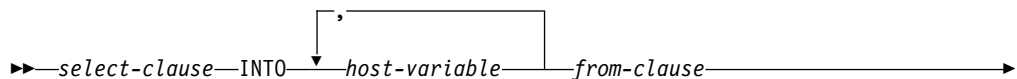
UR

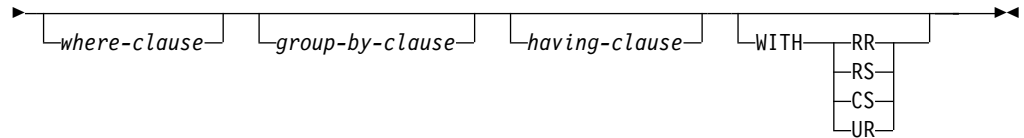
미확약 읽기(UR)

명령문의 기본 분리 레벨은 명령문이 바인드된 패키지의 분리 레벨입니다.

SELECT INTO

구문 도표는 다음과 같이 변경됩니다.





다음을 설명 섹션에 추가하십시오.

WITH

SELECT INTO가 실행되는 분리 레벨을 지정합니다.

RR

반복 읽기(RR)

RS

읽기 안정성(RS)

CS

커서 안정성(CS)

UR

미확약 읽기(UR)

명령문의 기본 분리 레벨은 명령문이 바인드된 패키지의 분리 레벨입니다.

SET ENCRYPTION PASSWORD

SET ENCRYPTION PASSWORD

SET ENCRYPTION PASSWORD 문은 암호화 및 암호해독 함수에서 사용될 암호를 설정합니다. 암호는 DB2 인증에 묶이지 않으며, 데이터 암호화에만 사용됩니다.

이 명령문은 트랜잭션 제어 하에 있지 않습니다.

호출: 이 명령문은 응용프로그램에 추가되거나 대화식으로 발행될 수 있습니다. 동적으로 준비될 수 있는 실행 명령문입니다.

권한 부여: 이 명령문을 실행하는 데에는 권한 부여가 필요하지 않습니다.

구문:

```
►► SET ENCRYPTION PASSWORD [ ] [host-variable | string-constant] ►►
```

설명: ENCRYPTION PASSWORD는 암호 기반 암호화에 대한 ENCRYPT, DECRYPT_BIN 및 DECRYPT_CHAR 내장 함수에 의해 사용될 수 있습니다. 길이는 6 및 127 이내이어야 합니다. 모든 문자는 대문자로의 변환이 없으므로 의도하는 정확한 대소문자로 지정되어야 합니다.

host-variable

유형 CHAR 또는 VARCHAR의 변수. 호스트 변수의 내용 길이는 6 및 127 이내이어야 합니다(SQLSTATE 428FC). 널(NULL)로 설정될 수 없습니다. 모든 문자는 대문자로의 변환이 없으므로 의도하는 정확한 대소문자로 지정되어야 합니다.

string-constant

문자열 상수. 길이는 6 및 127 이내이어야 합니다(SQLSTATE 428FC).

규칙:

- 초기 ENCRYPTION PASSWORD 값은 빈 문자열('')입니다.
- 호스트 변수 또는 문자열 상수는 보통 DB2 메커니즘을 사용하여 데이터베이스 서버로 전송됩니다.

참고:

- 이 명령문 사용에 대한 자세한 정보는 412 페이지의 『ENCRYPT』 및 411 페이지의 『DECRYPT_BIN 및 DECRYPT_CHAR』를 참조하십시오.

예: 예 1: 다음 명령문은 ENCRYPTION PASSWORD를 설정합니다.

```
SET ENCRYPTION PASSWORD = 'bubbalu'
```

SET 전이 변수

이 섹션은 다음과 같이 변경됩니다.

SET 변수

SET 변수 문은 값을 지역 변수나 새 전이 변수에 지정합니다. 이 명령문은 트랜잭션 제어 하에 있습니다.

호출: 이 명령문은 동적 복합 텍스트 명령문, 트리거, SQL 함수 또는 SQL 메소드 중 하나에서 SQL문으로서만 사용될 수 있습니다.

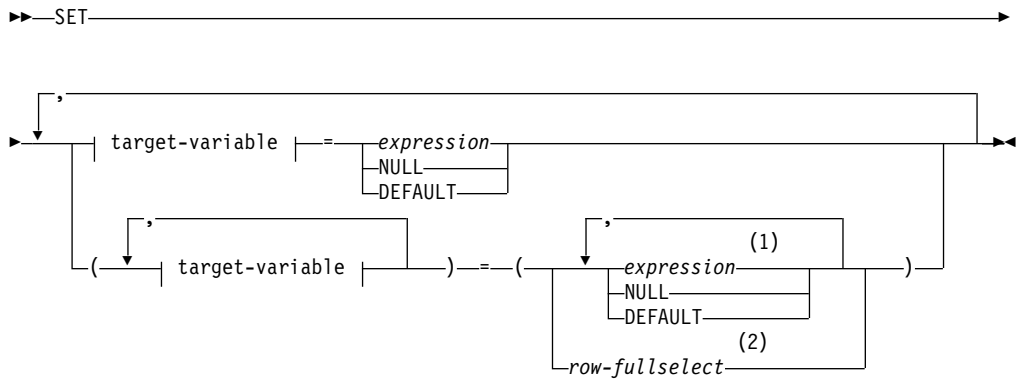
권한 부여: 전이 변수를 참조하려면, 트리거 작성자의 권한 부여 ID가 보유한 특권은 다음 중 하나를 포함해야 합니다.

- 왼쪽 지정에서 참조된 컬럼의 UPDATE 및 오른쪽에서 참조된 임의의 컬럼에 대한 SELECT.
- 테이블에 있는 CONTROL 특권(트리거의 주제 테이블).
- SYSADM 또는 DBADM 권한.

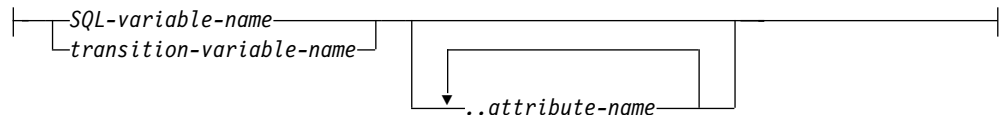
오른쪽 지정에서와 같이 *row-fullselect*로 이 명령문을 실행하려면, 트리거 정의자나 동적 복합 텍스트 명령문 소유자 중 하나의 권한 부여 ID에 의해 보유된 특권은 참조된 각 테이블이나 뷰 중 하나에 대해 최소한 다음 중 하나를 포함해야 합니다.

- SELECT 특권
- CONTROL 특권
- SYSADM 또는 DBADM

구문:



target-variable:



주:

- 1 표현식, NULL 및 DEFAULT의 수가 *target-variable*의 수와 일치해야 합니다.

2 선택된 목록에 있는 컬럼 수는 *target-variable*의 수와 일치해야 합니다.

설명:

target-variable

지정의 목표 변수를 식별합니다. 동일한 변수를 나타내는 *target-variable*는 두 번 이상 지정되지 않아야 합니다(SQLSTATE 42701).

SQL-variable-name

지정 목표인 SQL 변수를 식별합니다. SQL 변수는 사용되기 전에 선언되어야 합니다. SQL 변수는 동적 복합 텍스트 명령문에서 정의될 수 있습니다.

transition-variable-name

전이 행에서 갱신되는 컬럼을 식별합니다. *transition-variable-name*은 주제 테이블이나 트리거에서 컬럼을 식별해야 하며, 새 값을 식별하는 상관 이름에 의해 선택적으로 규정됩니다(SQLSTATE 42703).

..attribute name

설정된 구조화 유형의 속성을 지정합니다(속성 지정이라고 합니다). 지정된 *SQL-variable-name* 또는 *transition-variable-name*은 사용자 정의 구조화 유형으로 정의되어야 합니다(SQLSTATE 428DP). *attribute-name*은 구조화 유형의 속성이어야 합니다(SQLSTATE 42703). *..attribute name* 절을 포함하지 않은 지정을 관습적인 지정이라고 합니다.

expression

*target-variable*의 새 값을 나타냅니다. 표현식은 SQL 참조서의 설명하는 유형의 표현식입니다. 표현식이 스칼라 fullselect 내에서 발생할 때를 제외하고는 컬럼 함수를 포함할 수 없습니다(SQLSTATE 42903). CREATE TRIGGER 문의 문맥에서, 표현식에는 OLD 및 NEW 전이 변수에 대한 참조가 들어 있을 수 있으며 *correlation-name*에 의해 규정화되어 어떤 전이 변수인지 지정해야 합니다(SQLSTATE 42702).

NULL

널(NULL) 값을 지정하며 널(NULL) 입력 가능 컬럼에 대해서만 지정될 수 있습니다(SQLSTATE 23502). NULL은 속성의 데이터 유형으로 특별히 유형변환(cast)된 경우를 제외하고는, 속성 지정에 있는 값일 수 없습니다(SQLSTATE 429B9).

DEFAULT

기본값이 사용되어야 함을 지정합니다.

*target-variable*이 컬럼이면, 삽입된 값은 컬럼이 테이블에서 정의된 방식에 의해 좌우됩니다.

- 컬럼이 WITH DEFAULT 절을 사용하여 정의된 경우, 값은 컬럼에 대해 정의된 기본값으로 설정됩니다.

- 컬럼이 IDENTITY 절을 사용하여 정의된 경우, 값은 데이터베이스 관리 프로그램에 의해 생성됩니다.
- 컬럼이 WITH DEFAULT 절, IDENTITY 절 또는 NOT NULL을 지정하지 않고 정의된 경우, 값은 NULL입니다.
- 컬럼이 NOT NULL 절을 사용하여 정의되었으며, IDENTITY 절이 사용되지 않은 경우 또는 WITH DEFAULT 절이 사용되지 않았거나 DEFAULT NULL이 사용된 경우, DEFAULT 키워드는 해당 컬럼에 대해 지정될 수 없습니다 (SQLSTATE 23502).

*target-variable*가 SQL 변수인 경우, 삽입된 값은 변수 선언에서 지정되거나 암시된 기본값입니다.

row-fullselect

지정에 대해 지정된 *target-variables*의 수에 해당하는 컬럼 수와 함께 단일 행을 리턴하는 *fullselect*. 값은 각 해당 *target-variable*에 지정됩니다. *row-fullselect*의 결과에 행이 없으면, 널(NULL) 값이 지정됩니다. CREATE TRIGGER 문의 문맥에서 *row-fullselect*은 사용할 전이 변수를 지정하기 위해 *correlation-name*으로 규정화되어야 하는 OLD 및 NEW 전이 변수에 대한 참조를 포함할 수 있습니다 (SQLSTATE 42702). 결과에 둘 이상의 행이 있는 경우 오류가 리턴됩니다 (SQLSTATE 21000).

규칙:

- 표현식에서 지정되는 값의 수, NULL 및 DEFAULT 또는 *row-fullselect*는 지정에서 지정된 *target-variables*의 수와 일치해야 합니다(SQLSTATE 42802).
- SET 변수 명령문은 SQL 변수 및 전이 변수를 하나의 명령문에서 지정할 수 없습니다(SQLSTATE 42997).
- 값은 SQL 참조서의 기술된 지정 규칙 아래의 *target-variables*에 지정됩니다. 명령문이 BEFORE UPDATE 트리거에서 사용되고, 레지스트리 변수 DB2_UPDATE_PART_KEY=OFF이면, *target-variable*로서 지정된 *transition-variable*은 파티션 키 컬럼일 수 없습니다(SQLSTATE 42997).

참고:

- 둘 이상의 지정이 포함되면, 모든 표현식 및 *row-fullselect*는 지정이 수행되기 전에 평가됩니다. 표현식이나 행 *fullselect*에서 *target-variables*에 대한 참조는 항상 단일 SET 문의 지정을 선행하는 *target-variable*의 값입니다.
- 구별 유형으로서 정의된 식별 컬럼이 갱신되면, 전체 계산은 소스 유형으로 수행되며, 값이 실제로 컬럼에 지정되기 전에 결과는 구별 유형으로 유형변환(cast)됩니다.

8

8. 계산에 앞서 이전 값을 소스 유형으로의 이전 값의 유형변환은 없습니다.

- 식별 컬럼에 대한 SET 문에서 DB2가 값을 생성하게 하려면, DEFAULT 키워드를 사용하십시오.

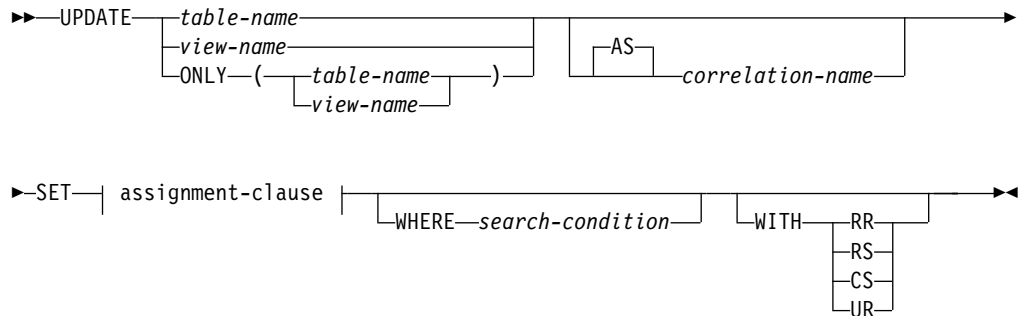
```
SET NEW.EMPNO = DEFAULT
```

이 예에서, NEW.EMPNO는 식별 컬럼으로서 정의되며, 이 컬럼을 갱신하기 위해 사용되는 값은 DB2에 의해 생성됩니다.

이 명령문의 예는 동일합니다.

UPDATE

검색된 UPDATE 구문 도표는 다음과 같이 변경됩니다.



다음은 설명 섹션에 추가하십시오.

WITH

UPDATE 문이 실행되는 분리 레벨을 지정합니다.

RR

반복 읽기(RR)

RS

읽기 안정성(RS)

CS

커서 안정성(CS)

UR

미확약 읽기(UR)

명령문의 기본 분리 레벨은 명령문이 바인드된 패키지의 분리 레벨입니다.

SQL 프로시저어를 이제는 SQL 제어 명령문이라고 합니다

제어 명령문은 구조화 프로그래밍 언어에서 프로그램을 작성하는 것과 유사한 방법으로 SQL을 사용하도록 허용하는 SQL문입니다. SQL 제어 명령문은 루틴, 트리거 또는 동적 복합 텍스트 명령문의 본문에서 사용될 수 있습니다.

| 이 장에는 SQL-procedure-statement문과 함께 지원되는 SQL 제어 명령문의 구문 및
| 설명이 들어 있습니다.

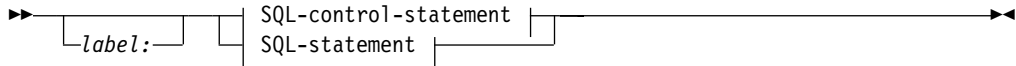
| **SQL 프로시저어 명령문**

| SQL 프로시저어 명령문 정보는 다음과 같이 변경됩니다.

SQL 프로시저어 명령문

이 장에는 구문 도표, 의미론 설명, 규칙이 들어 있으며, SQL 루틴, 트리거 또는 동적 복합 텍스트 명령문을 구성하는 명령문 사용의 예가 들어 있습니다.

구문:



SQL-control-statement:

<i>ALLOCATE CURSOR statement</i>	(1)
<i>assignment statement</i>	(1)
<i>ASSOCIATE LOCATORS statement</i>	(1)
<i>CASE statement</i>	(1)
<i>dynamic-compound statement</i>	(2)
<i>FOR statement</i>	
<i>GET DIAGNOSTICS statement</i>	(1)
<i>GOTO statement</i>	
<i>IF statement</i>	
<i>ITERATE statement</i>	
<i>LEAVE statement</i>	(1)
<i>LOOP statement</i>	(1)
<i>procedure-compound statement</i>	(1)
<i>REPEAT statement</i>	(1)
<i>RESIGNAL statement</i>	(1)
<i>RETURN statement</i>	
<i>SIGNAL statement</i>	
<i>WHILE statement</i>	

주:

- 1 이 명령문은 SQL 프로시저어의 범위에서만 지원됩니다.
- 2 이 명령문은 트리거, SQL 함수 또는 SQL 메소드 내에서만 지원됩니다. 가장 바깥의 명령문이어야 합니다.

설명:

label:

SQL 프로시저어 명령문의 레이블을 지정합니다. 레이블은 목록 내에서 중첩된 모든 복합 텍스트 명령문을 포함하여 SQL 프로시저어 명령문 목록 내에서 고유해야

합니다. 중첩되지 않은 복합 텍스트 명령문은 동일한 레이블을 사용할 수 있습니다.
SQL 프로시저 명령문 목록은 몇 개의 SQL 제어 명령문에서 가능합니다.

트리거, SQL 함수나 메소드 또는 동적 복합 텍스트 명령문의 문맥에서, 동적 복합
텍스트 명령문, FOR 명령문 및 WHILE 명령문만이 레이블될 수 있습니다.

SQL-statement

SQL 프로시저의 본문에서, 모든 실행 SQL문이 포함되며, 다음과 같은 예외를
갖습니다.

- CONNECT
- 색인, 테이블 또는 뷰 이외의 오브젝트 CREATE
- DESCRIBE
- DISCONNECT
- 색인, 테이블 또는 뷰 이외의 오브젝트 DROP
- FLUSH EVENT MONITOR
- REFRESH TABLE
- RELEASE (연결만)
- RENAME TABLE
- RENAME TABLESPACE
- REVOKE
- SET CONNECTION
- SET INTEGRITY

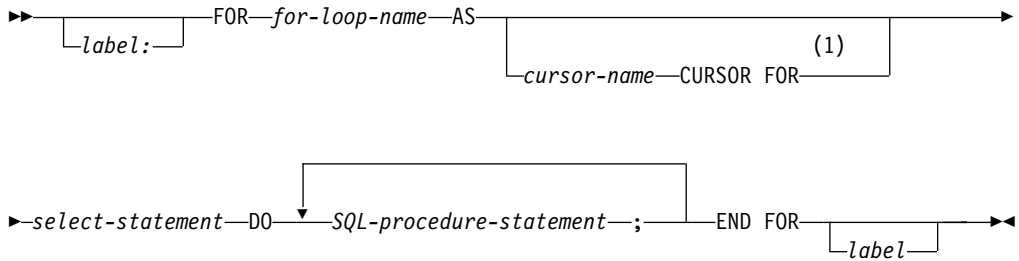
주: SQL 프로시저 본문 내에서 CALL 문을 포함할 수 있지만, 이 CALL 문은
또다른 SQL 프로시저나 C 프로시저만을 호출할 수 있습니다. SQL 프
시저 본문 내의 CALL 문은 저장 프로시저의 다른 유형을 호출할 수 없
습니다.

FOR

FOR

FOR 문은 테이블의 각 행에 대한 명령문이나 명령문 그룹을 실행합니다.

구문:



주:

1 이 옵션은 SQL 프로시저의 문맥에서만 사용될 수 있습니다.

설명:

label

FOR 문에 대한 레이블을 지정합니다. 시작 레이블이 지정되면, 해당 레이블은 LEAVE 및 ITERATE 명령문에서 사용될 수 있습니다. 종료 레이블이 지정된 경우, 시작 레이블과 동일해야 합니다.

for-loop-name

FOR 문을 구현하기 위해 생성된 내재된 복합 텍스트 명령문에 대한 레이블을 지정합니다. FOR 문 내에서 ITERATE 또는 LEAVE 문과 함께 사용될 수 없다는 것만 제외하고 복합 텍스트 명령문의 레이블에 대한 규칙을 따릅니다. *for-loop-name* 은 지정된 *select-statement*가 리턴한 컬럼 이름을 규정하는 데 사용됩니다.

cursor-name

SELECT 문에서 결과 테이블로부터 행을 선택하기 위해 사용된 커서의 이름을 지정합니다. 지정되지 않은 경우, DB2는 고유한 커서 이름을 생성합니다.

select-statement

커서의 SELECT 문을 지정합니다. 선택 목록에 있는 모든 컬럼은 이름을 가져야 하며 동일한 이름을 가진 두 개의 컬럼이 있을 수 없습니다.

트리거, 함수, 메소드 또는 동적 복합 텍스트 명령문에서, *select-statement*는 선택적 공통 테이블 표현식을 가진 *fullselect*으로 구성되어야 합니다.

SQL-procedure-statement

테이블의 각 행에 대해 호출되는 명령문(또는 명령문들)을 지정합니다.

규칙:

- 선택 목록은 고유한 컬럼 이름으로 구성되어야 하며 선택 목록에서 지정된 테이블은 프로시저가 작성될 때 존재해야 하거나, 이전 SQL 프로시저 명령문에서 작성된 테이블이어야 합니다.
- for-statement에서 지정된 커서는 for-statement 외부에서 참조될 수 없으며 OPEN, FETCH, CLOSE 문에서 지정될 수 없습니다.

예: 다음 예에서, for-statement는 전체 employee 테이블에서 반복하기 위해 사용됩니다. 테이블에 있는 각 행의 경우, SQL 변수 fullname은 사원의 성씨, 그 다음에는 성표, 이름, 공백 및 중간 이름의 첫자로 설정됩니다. fullname에 대한 각 값은 tnames 테이블로 삽입됩니다.

```

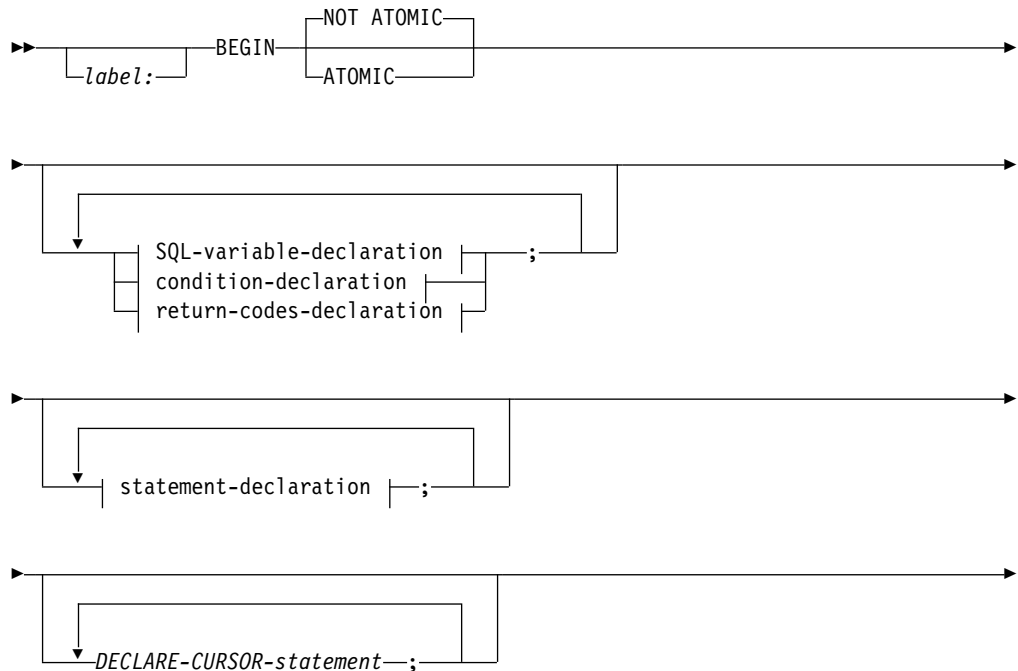
BEGIN
  DECLARE fullname CHAR(40);
  FOR v1 AS
    SELECT firstnme, midinit, lastname FROM employee
    DO
      SET fullname = lastname || ',' || firstnme || ' ' || midinit;
      INSERT INTO tnames VALUE (fullname);
    END FOR
END

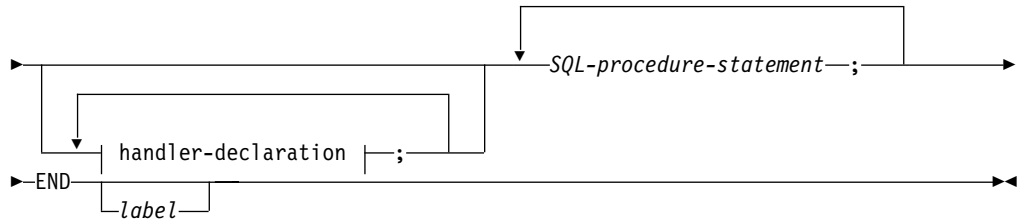
```

복합 텍스트 명령문(프로시저)에 대한 복합 텍스트 명령문 변경사항

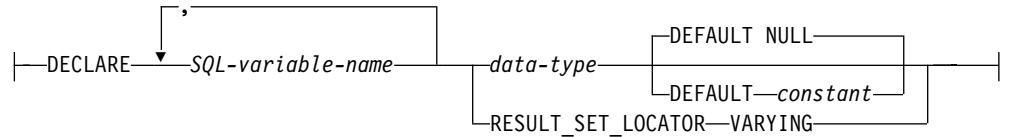
프로시저 복합 텍스트 명령문은 SQL 프로시저에서 다른 명령문을 함께 그룹화합니다. SQL 변수, 커서 및 조건 핸들러를 복합 텍스트 명령문 내에서 선언할 수 있습니다.

구문 도표는 이제 procedure-compound-statement라는 제목을 갖습니다.

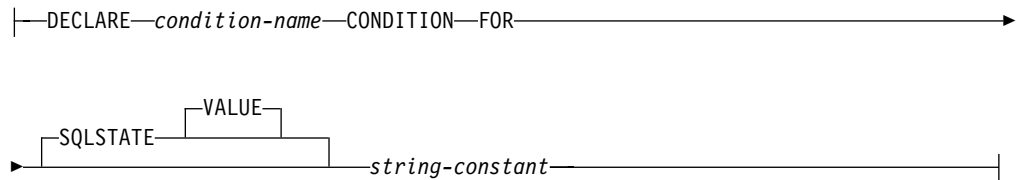




SQL-variable-declaration:



condition-declaration:



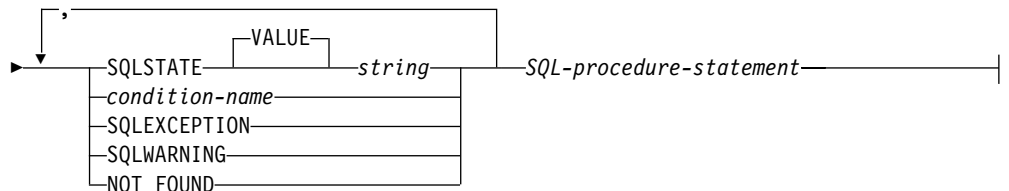
statement-declaration:



return-codes-declaration:



handler-declaration:



| **statement-declaration**

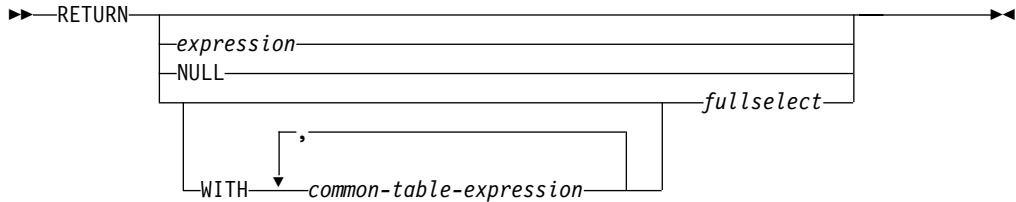
| *statement-declaration*은 복합 명령문에 지역적인 하나 이상의 이름 목록을 선언합
| 니다. 명령문 이름은 동일한 복합 명령문 내에서 다른 명령문 이름과 같을 수 없습
| 니다.

| **RETURN**

RETURN

RETURN 문은 루틴으로부터 리턴하기 위해 사용됩니다. SQL 함수 또는 메소드의 경우, 함수 또는 메소드의 결과를 리턴합니다. SQL 프로시저의 경우, 선택적으로 정수 상태 값을 리턴합니다.

구문:



설명:

expression

루틴으로부터 리턴된 값을 지정합니다.

- 루틴이 함수 또는 메소드인 경우, *expression* 중 하나, NULL 또는 *fullselect*가 지정되어야 하며(SQLSTATE 42630) 결과의 데이터 유형은 루틴의 RETURNS 유형에 지정 가능해야 합니다(SQLSTATE 42866).
- 스칼라 표현식(스칼라 *fullselect* 이외)은 테이블 함수에 대해 지정될 수 없습니다 (SQLSTATE 428F1).
- 루틴이 프로시저인 경우, *expression*의 데이터 유형은 INTEGER이어야 합니다(SQLSTATE 428E2). 프로시저는 NULL 또는 *fullselect*를 리턴할 수 없습니다.

NULL

함수 또는 메소드는 RETURNS 절에 정의된 데이터 유형의 널(NULL) 값을 리턴함을 지정합니다. NULL은 프로시저에서 RETURN에 대해 지정될 수 없습니다.

WITH *common-table-expression*

다음에 오는 *fullselect*와 함께 사용할 공통 테이블 표현식을 정의합니다.

fullselect

함수에 대해 리턴되는 행을 지정합니다. *fullselect*에 있는 컬럼 수는 함수 결과에 있는 컬럼 수와 일치해야 합니다(SQLSTATE 42811). 그리고, *fullselect*의 정적 컬럼 유형은 컬럼에 대한 지정 규칙을 사용하여, 함수 결과의 선언된 컬럼 유형에 지정 가능해야 합니다(SQLSTATE 42866).

*fullselect*는 프로시저에서 RETURN에 대해 지정될 수 없습니다.

루틴이 스칼라 함수 또는 메소드인 경우, *fullselect*는 하나의 컬럼(SQLSTATE 42823)과 최대한 하나의 행(SQLSTATE 21000)을 리턴해야 합니다.

루틴이 행 루틴인 경우, 최대한 하나의 행(SQLSTATE 21505)을 리턴해야 합니다.

루틴이 테이블 함수인 경우, 하나 이상의 컬럼을 가진 0개 이상의 행을 리턴할 수 있습니다.

규칙:

- SQL 함수 또는 메소드의 실행은 RETURN으로 종료되어야 합니다(SQLSTATE 42632).
- SQL 테이블이나 *dynamic-compound-statement*를 사용하는 행 함수에서, 허용된 유일한 RETURN 명령문은 복합 텍스트 문의 끝에 있는 명령문입니다(SQLSTATE 429BD).

참고:

- 값이 프로시저에서 리턴되면, 호출자는 다음을 사용하여 값을 액세스할 수 있습니다.
 - SQL 프로시저가 또다른 SQL 프로시저에서 호출되었을 때 RETURN_STATUS를 검색하기 위한 GET DIAGNOSTICS 문
 - CLI 응용프로그램에서 escape 절 CALL 구문(=?=CALL...)에서 리턴 값 매개변수 표시문자에 바인드된 매개변수
 - SQLCODE가 0보다 작지 않을 때 SQLERRD[0]의 값을 검색하여 SQL 프로시저의 CALL 처리에서 리턴된 SQLCA에서 직접(SQLCODE가 0보다 작을 때 -1 값 가정).

예: 성공적인 경우 상태 값 0을 가진 SQL 저장 프로시저에서 리턴할 RETURN 문을 사용하고, 그렇지 않은 경우 -200을 사용하십시오.

```
BEGIN
...
  GOTO FAIL
...
  SUCCESS: RETURN 0
  FAIL: RETURN -200
END
```

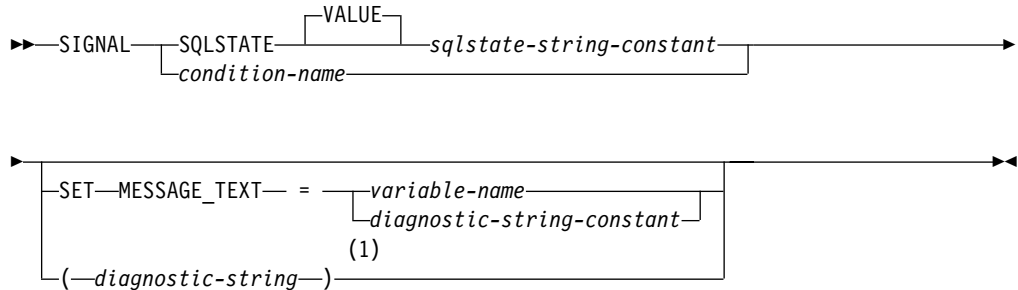
SIGNAL

SIGNAL SQLSTATE 문은 이 용도로 더 이상 사용되지 않습니다.

SIGNAL

SIGNAL 문은 오류나 경고 조건을 신호하기 위해 사용됩니다. 선택적 메시지 텍스트와 함께 지정된 SQLSTATE와 함께 오류나 경로가 리턴되도록 합니다.

구문:



주:

- 1 이 옵션은 DB2의 이전 버전과의 호환성을 위해 CREATE TRIGGER 문의 범위 내에서만 제공됩니다.

설명:

SQLSTATE VALUE *sqlstate-string-constant*

지정된 문자열 상수는 SQLSTATE를 나타냅니다. SQLSTATE에 대한 규칙을 따르는 정확히 5문자를 가진 문자열 상수이어야 합니다.

- 각 문자는 숫자 세트('0' - '9') 또는 강조되지 않은 대문자('A' - 'Z')에 속해야 합니다.
- '00'는 성공적인 완료를 나타내므로, SQLSTATE 클래스(처음 두 문자)는 이것 이 될 수 없습니다.

동적 복합 텍스트 명령문, 트리거, SQL 함수 또는 SQL 메소드 중 하나의 문맥에서 다음 규칙도 적용되어야 합니다.

- '01' 또는 '02'는 오류 클래스가 아니므로, SQLSTATE 클래스(처음 두 문자)는 이들이 될 수 없습니다
- SQLSTATE 클래스가 '0' - '6'까지의 번호나 'A' - 'H'까지의 문자로 시작되면, 그 서브클래스(마지막 세 문자)는 'I' - 'Z' 범위에 있는 문자로 시작되어야 합니다.
- SQLSTATE 클래스가 숫자 '7', '8', '9' 또는 문자 'I' - 'Z'로 시작하는 경우 서브클래스는 '0' - '9' 또는 'A' - 'Z' 중 하나일 수 있습니다.

SQLSTATE가 이 규칙에 적합하지 않는 경우, 오류가 발생합니다(SQLSTATE 428B3).

condition-name

조건의 이름을 지정합니다. 조건 이름은 프로시저어 내에서 고유해야 하며 선언된 복합 텍스트 명령문 내에서만 참조될 수 있습니다.

SET MESSAGE_TEXT=

오류나 경고를 설명하는 문자열을 지정합니다. 문자열은 SQLCA의 SQLERRMC 필드에서 리턴됩니다. 실제 문자열이 70 바이트보다 길지 않은 경우, 경고 없이 절단됩니다. SQLSTATE 또는 *condition-name*도 지정된 경우에만 이 절이 지정될 수 있습니다(SQLSTATE 42601).

variable-name

복합 텍스트 명령문 내에서 선언되어야 하는 SQL 변수를 식별합니다. SQL 변수는 CHAR 또는 VARCHAR 데이터 유형으로서 정의되어야 합니다.

diagnostic-string-constant

메시지 텍스트가 들어 있는 문자열 상수를 지정합니다.

diagnostic-string

오류 조건을 설명하기 위해 최대 70 바이트의 문자열을 리턴하는 CHAR 또는 VARCHAR의 유형을 가진 표현식. 문자열이 70 바이트보다 긴 경우, 절단됩니다. 이 옵션은 DB2의 이전 버전과의 호환성을 위해 CREATE TRIGGER 문의 범위 내에서만 제공됩니다. 정규 사용은 권장되지 않습니다.

참고:

- SIGNAL 문이 발행된 경우, 지정된 SQLCODE는 다음과 같습니다.
 - SQLSTATE가 '01' 또는 '02'로 시작된 경우 +438
 - 또는 -438
- SQLSTATE 또는 조건이 예외('01' 또는 '02' 이외의 SQLSTATE 클래스) 신호를 나타내는 경우,
 - 핸들러가 신호 명령문과 동일한 복합 텍스트 명령문(또는 외부 복합 텍스트 명령문)에 있고, 복합 텍스트 명령문에 지정된 SQLSTATE, *condition-name* 또는 SQLEXCEPTION에 대한 핸들러가 들어 있다면 예외가 핸들되고 제어는 핸들러로 이전됩니다.
 - 예외가 핸들될 수 없는 경우, 제어는 즉시 복합 텍스트 명령문의 끝으로 리턴됩니다.
- SQLSTATE 또는 조건이 경고(SQLSTATE 클래스 '01') 또는 찾지 못함 조건(SQLSTATE 클래스 '02') 신호를 나타내는 경우,
 - 핸들러가 신호 명령문과 동일한 복합 텍스트 명령문(또는 외부 복합 텍스트 명령문)에 존재하며, 복합 텍스트 명령문에는 지정된 SQLSTATE, *condition-name*, SQLWARNING(SQLSTATE 클래스가 '01'인 경우) 또는 NOT FOUND

(SQLSTATE 클래스가 '02'인 경우) 또는 NOT에 대한 핸들러가 들어 있다면 경고 또는 찾지 못함 조건이 핸들되고 제어가 핸들러로 이전됩니다.

- 경고가 핸들될 수 없는 경우, 처리는 다음 명령문으로 계속합니다.

- SQLSTATE 값은 두 문자 클래스 코드 값과 그 뒤에 오는 세 문자 서브클래스 코드 값으로 구성됩니다. 클래스 코드 값은 성공 및 실패 실행 조건의 클래스를 나타냅니다.

임의의 유효한 SQLSTATE 값은 SIGNAL 문에서 사용될 수 있습니다. 그러나, 프로그래머가 응용프로그램용으로 예약된 범위에 근거하여 새로운 SQLSTATE를 정의하도록 권장합니다. 이것은 장래 릴리스에서 데이터베이스 관리 프로그램에 의해 정의될 수 있는 SQLSTATE 값의 의도하지 않은 사용을 방지합니다.

- 문자 '7' - '9' 또는 'I' - 'Z'로 시작하는 SQLSTATE 클래스가 정의될 수 있습니다. 이 클래스 내에서, 어떤 클래스도 정의될 수 있습니다.

- 문자 '0' - '6' 또는 'A' - 'H'로 시작하는 SQLSTATE 클래스는 데이터베이스 관리 프로그램을 위해 예약되어 있습니다. 이 클래스 내에서, 문자 '0' - 'H'로 시작하는 서브클래스는 데이터베이스 관리 프로그램에 대해 예약됩니다. 문자 'I' - 'Z'로 시작하는 서브클래스가 정의될 수 있습니다.

예: 응용프로그램이 고객 번호를 알 수 없을 때 응용프로그램 오류를 신호하는 주문 시스템의 SQL 프로시저. ORDERS 테이블은 주문이 삽입되기 전에 CUSTNO가 존재하도록 요구하며, CUSTOMER 테이블에 외부 키를 포함합니다.

```
CREATE PROCEDURE SUBMIT_ORDER
  (IN ONUM INTEGER, IN CNUM INTEGER,
   IN PNUM INTEGER, IN QNUM INTEGER)
  SPECIFIC SUBMIT_ORDER
  MODIFIES SQL DATA
  LANGUAGE SQL
  BEGIN
    DECLARE EXIT HANDLER FOR SQLSTATE VALUE '23503'
      SIGNAL SQLSTATE '75002'
      SET MESSAGE_TEXT = '고객 번호를 알 수 없습니다';
    INSERT INTO ORDERS (ORDERNO, CUSTNO, PARTNO, QUANTITY)
      VALUES (ONUM, CNUM, PNUM, QNUM);
  END
```

SQL 제한사항

Database Manager Limits의 표 33에 변경된 사항이 있습니다. 레지스트리 변수 DB2_INDEX_2BYTEVARLEN이 ON으로 설정되었기 때문에 이제 가장 긴 변수 색인 키 부분이 255(바이트로) 이상이 될 수 있습니다.

카탈로그 뷰

새로운 카탈로그 뷰가 추가되었습니다.

SYSCAT.SEQUENCES

SYSCAT.SEQUENCES 뷰는 FixPak 3 이상으로 작성된 데이터베이스에 대해 자동으로 생성됩니다. FixPak 3 이전에 작성된 데이터베이스의 경우, **db2updv7** 명령을 수행하여 뷰를 데이터베이스에 추가하십시오. 세부사항에 대해서는 릴리스 정보에서 Command Reference 갱신을 참조하십시오.

SQL 데이터 정의 명령문, 환경 루틴 및 특정 유틸리티에 응답하여 일반 조작 중 이 카탈로그 뷰가 갱신됩니다. 카탈로그 뷰에 있는 데이터는 일반 SQL 조회 기능을 통해 사용 가능합니다. 컬럼은 설명하는 오브젝트의 유형에 근거하여 일관된 이름을 갖습니다.

표 30. SYSCAT.SEQUENCES 카탈로그 뷰에 있는 컬럼

컬럼 이름	데이터 유형	널(NULL) 입력 가능	설명
SEQSCHEMA	VARCHAR(128)		순차의 스키마.
SEQNAME	VARCHAR(128)		순차 이름(식별 컬럼에 대한 DB2에 의해 생성).
DEFINER	VARCHAR(128)		순차의 정의자.
OWNER	VARCHAR(128)		순차의 소유자.
SEQID	INTEGER		순차의 내부 ID.
SEQTYPE	CHAR(1)		순차 유형 S - 일반 순차
INCREMENT	DECIMAL(31,0)		증분값.
START	DECIMAL(31,0)		시작값.
MAXVALUE	DECIMAL(31,0)	예	최대 값.
MINVALUE	DECIMAL(31,0)		최소 값.
CYCLE	CHAR(1)		경계에 도달하면 순환이 발생합니다. Y - 순환 발생 N - 순환이 발생하지 않음
CACHE	INTEGER		더 빠른 액세스를 위해 메모리에 사전할당하려는 순차 값의 수. 0은 값이 사전할당되지 않음을 나타냅니다.
ORDER	CHAR(1)		순차 번호가 요청 순서대로 생성되어야 하는지 여부. Y - 순차 번호는 요청 순서 대로 생성되어야 합니다. N - 순차 번호는 요청 순서 대로 생성되도록 요구되지 않습니다.
DATATYPEID	INTEGER		내장 유형의 경우, 내장 유형의 내부 ID. 구별 유형의 경우, 구별 유형의 내부 ID.
SOURCETYPEID	INTEGER		내장 유형의 경우, 이것은 0 값을 갖습니다. 구별 유형의 경우, 이것은 구별 유형의 소스 유형인 내장 유형의 내부 ID입니다.
CREATE_TIME	TIMESTAMP		순차가 작성된 시간.
ALTER_TIME	TIMESTAMP		마지막 ALTER SEQUENCE 문이 이 순차에 대해 실행되었을 때의 시간.

표 30. SYSCAT.SEQUENCES 카탈로그 뷰에 있는 컬럼 (계속)

컬럼 이름	데이터 유형	널(NULL) 입력 가능	설명
PRECISION	SMALLINT		십진수 또는 숫자 유형을 갖는 순차에 대해 정의된 정밀도. 값은 SMALLINT의 경우 1, INTEGER의 경우 10 그리고 BIGINT의 경우 19입니다.
ORIGIN	CHAR(1)		순차 원점 U - 사용자 생성 순차 S - 시스템 생성 순차
REMARKS	VARCHAR(254)	예	사용자 제공 주석 또는 널(NULL)

DB2 Stored Procedure Builder

DB2 Stored Procedure Builder에 대한 Java 1.2 지원

DB2 Stored Procedure Builder는 Java 1.2 기능을 사용한 Java Stored Procedure Builder를 지원합니다. 그 외에도, Stored Procedure Builder는 Java 1.2에서 bi-di 지원을 사용하여 아랍어와 히브리어 같은 양방향 언어를 지원합니다.

이 지원은 Windows NT 플랫폼에서만 제공됩니다.

Stored Procedure Builder가 Java 1.2 기능을 인식하고 사용할 수 있도록 Java 1.2를 설치해야 합니다.

Java 1.2를 설치하려면, 다음과 같이 수행하십시오.

1. JDK 1.2.2는 DB2 UDB CD의 DB2\bi\NT 디렉토리에서 사용할 수 있습니다. `ibm-inst-n122p-win32-x86.exe`는 설치 프로그램이고 `ibm-jdk-n122p-win32-x86.exe`는 JDK 분산 프로그램입니다. 두 파일을 모두 하드 드라이브의 임시 디렉토리에 복사한 다음, 거기에서 설치 프로그램을 수행하십시오.
2. <DB2PATH>\java\Java12에 설치하십시오. 여기서 <DB2PATH>는 DB2의 설치 경로입니다.
3. JDK/JRE 설치에서 프롬프트를 표시할 때 시스템 VM으로 JDK/JRE를 선택하지 마십시오.

Java 1.2가 성공적으로 설치된 후에, 보통 방식으로 Stored Procedure Builder를 시작하십시오.

JDK 1.2 지원을 사용하여 Java 저장 프로시저를 실행하려면, 다음의 명령을 사용하여 데이터베이스 서버 환경 변수 `DB2_USE_JDK12`를 TRUE로 설정하십시오.

```
DB2SET DB2_USE_JDK12=TRUE
```

또한, Java 1.2 지원이 설치된 디렉토리를 지정하도록 `JDK11_PATH`를 설정하십시오. 다음의 명령을 사용하여 이 경로를 설정하십시오.

```
DB2 UPDATE DBM CFG USING JDK11_PATH
```

Java 1.2의 사용을 중지하려면, <DB2PATH>\java\Java12에서 JDK/JRE를 설치 제거하거나 간단하게 <DB2PATH>\java\Java12 서브디렉토리의 이름을 바꿀 수 있습니다.

중요: <DB2PATH>\java\Java12를 <DB2PATH>\Java12와 혼동하지 마십시오.
<DB2PATH>\Java12는 DB2 설치의 일부이며 Java 1.2에 대한 JDBC 지원이 포함됩니다.

DB2 저장 프로시저의 원격 디버그

UNIX 및 Windows 플랫폼에서 Java 및 C 저장 프로시저에 대한 원격 디버그 기능을 사용하려면, IBM Distributed Debugger를 설치해야 합니다. IBM Distributed Debugger는 Java용 Visual Age Professional Edition CD에 있습니다. 디버거 클라이언트는 Windows 플랫폼에서만 수행됩니다. 지원되는 서버 플랫폼은 Windows, AIX 및 Solaris입니다. Stored Procedure Builder 내장 SQL 디버그 기능을 사용하여 Windows 및 UNIX 플랫폼의 지역 및 원격 SQL 저장 프로시저를 디버그하십시오. OS/2 플랫폼에 대한 지원은 현재 사용 불가능합니다. OS/390용 DB2 Stored Procedure Builder에 대한 자세한 정보는 웹 사이트 <http://www-4.ibm.com/software/data/db2/os390/spb/exciting>를 참조하십시오.

OS/390 플랫폼에서 SQL 프로시저를 디버그하려면, OS/390 R1 제품에 대한 IBM C/C++ Productivity Tools도 있어야 합니다. 웹 사이트 <http://www.ibm.com/software/ad/c390/pt>에서 OS/390 R1의 IBM C/C++ Productivity Tools에 대한 자세한 내용을 참조하십시오.

Windows, OS/2 또는 UNIX 플랫폼에서 SQL 프로시저 빌드

Stored Procedure Builder를 사용하여 Windows, OS/2 또는 UNIX 데이터베이스에서 SQL 프로시저를 성공적으로 빌드하려면, SQL 프로시저에 대해 서버를 구성해야 합니다. SQL 프로시저의 사용자 서버 구성 방법에 대한 정보는, 360 페이지의 『Java 애플릿 및 응용프로그램 빌드』를 참조하십시오.

Solaris 플랫폼에서 DB2 Stored Procedure Builder 사용

Solaris 플랫폼에서 Stored Procedure Builder를 사용하려면, 다음과 같이 수행하십시오.

1. JDK 1.1.8을 다운로드하고 설치하십시오. JavaSoft 웹 사이트에서 JDK 1.1.8을 다운로드할 수 있습니다.
2. 환경 변수 JAVA_HOME을 JDK를 설치한 위치로 설정하십시오.
3. DB2 JDK11_PATH를 JDK를 설치한 디렉토리로 설정하십시오.
DB2 JDK11_PATH를 설정하려면, 다음의 명령을 사용하십시오.
DB2 UPDATE DBM CFG USING JDK11_PATH.

알려진 문제점 및 제한사항

- SQL 프로시저어는 현재 Windows 98에서 지원되지 않습니다.
- Java 저장 프로시저어의 경우, JAR ID, 클래스 이름 및 메소드 이름은 ASCII 이외 문자를 포함할 수 없습니다.
- AS/400에서 다음 V4R4 PTF는 OS/400 V4R4에 적용되어야 합니다.
 - SF59674
 - SF59878
- 문자 부속 유형이 FOR MIXED DATA 또는 FOR SBCS DATA인 저장 프로시저어 매개변수는 저장 프로시저어가 데이터베이스로부터 복원될 때 편집기 분할창에 소스 코드에 표시되지 않습니다.
- 현재, Java 소스가 데이터베이스에서 검색될 때 문제점이 있습니다. 검색 시, 코드의 주석은 축소되어 제공됩니다. 이는 ASCII 이외 코드 페이지에서 작업하는 DB2 Stored Procedure Builder 사용자와 클라이언트 및 서버가 다른 코드 페이지에 있는 사용자에게 영향을 줍니다.

대만어 로케일에서 DB2 Stored Procedure Builder 사용

대만어 로케일에서 Java Development Kit 또는 Java Runtime 1.1.8을 사용할 경우 문제점이 발생합니다. Stored Procedure Builder 프로그램의 그래픽 측면(메뉴, 편집기 텍스트, 메시지 등)이 적절하게 표시되지 않습니다. 해결책은 다음 디렉토리 중 하나 또는 둘 다에 표시되는 font.properties.zh_TW 파일로 변경하는 것입니다.

```
sql1lib/java/jdk/lib  
sql1lib/java/jre/lib
```

다음은

```
monospaced.0=\u7d30\u660e\u9ad4,CHINESEBIG5_CHARSET,NEED_CONVERTED
```

다음과 같이 변경하십시오.

```
monospaced.0=Courier New,ANSI_CHARSET
```

UNIX(AIX, Sun Solaris, Linux) 설치 및 Stored Procedure Builder

Sun Solaris 설치의 경우, UDB와 함께 AIX에서 설치된 것과 다른 Java Development Kit 또는 Runtime을 사용할 경우, Java가 설치된 경로(즉, /bin 및 /lib 디렉토리를 포함하는 디렉토리)로 환경 변수 JAVA_HOME을 설정해야 합니다. Stored Procedure Builder는 Linux에서 지원되지 않지만 지원되는 플랫폼에서 사용하여 Linux용 DB2 UDB 시스템에서 저장 프로시저어를 빌드하고 수행할 수 있습니다.

지원되는 플랫폼은 클라이언트에 대한 AIX, Solaris 및 NT와 서버에 대한 AIX, Solaris, NT, Linux, OS/2, HP-UX 및 NUMA-Q를 포함합니다.

OS/390에서 SQL 저장 프로시저어 빌드

DB2 Stored Procedure Builder는 OS/390용 DB2 UDB V7 서버에서 SQL 저장 프로시저어 빌드를 지원합니다.

SQL 저장 프로시저어

Windows 및 UNIX 플랫폼에서 SQL 저장 프로시저어의 디버거는 이제 직접 DB2 저장 프로시저어 빌더로 통합됩니다. KEEP DARI 데이터베이스 관리 프로그램 구성 옵션은 비분리(unfenced, 트러스트된) SQL 프로시저어를 디버거할 때 YES 또는 NO로 설정될 수 있습니다. 그러나 분리(fenced, 트러스트되지 않은) SQL 프로시저어를 디버거할 때 YES(기본값)로 설정되어야 합니다. 통합된 디버거 사용에 대한 자세한 정보는 Stored Procedure Builder 도움말을 참조하십시오.

Java 저장 프로시저어 내보내기

DB2 Stored Procedure Builder는 이제 내보내는 Java 저장 프로시저어를 지원합니다. Java 저장 프로시저어를 내보내려면 다음을 참조하십시오.

1. 저장 프로시저어 폴더를 마우스 오른쪽 버튼으로 누르고, Java 저장 프로시저어 내보내기를 눌러 Java 저장 프로시저어 내보내기 창을 여십시오.
2. 내보내려는 저장 프로시저어를 선택하고, "선택된 저장 프로시저어" 컬럼으로 이동하십시오.
3. 선호하는 옵션을 선택한 다음, 확인을 누르십시오.

OS/390에서 저장 프로시저어 삽입

OS/390에서 실행하는 DB2 저장 프로시저어 버전 5 이상의 경우, 마법사를 사용하여 저장 프로시저어를 삽입하고 WLM 환경 옵션이 없음을 나타내는 경우, 생성된 코드에는 NO WLM ENVIRONMENT이라는 텍스트가 들어 있습니다. 이 코드 라인은 예상된 대로 SPAS 주소 공간에서 저장 프로시저어를 실행하도록 야기합니다. 이러한 수정사항은 DB2 저장 프로시저어 빌더 버전 6 이상에 존재하는 문제점을 해결합니다.

수정 이후에 생성된 코드는 다음과 같습니다.

```
CREATE PROCEDURE SYSPROC.Proc2 ( )
    RESULT SETS 1
    LANGUAGE SQL
    MODIFIES SQL DATA
    COLLID TEST
    NO WLM ENVIRONMENT
    ASUTIME NO LIMIT
    RUN OPTIONS 'NOTEST(ALL,*, ,VADTCPIP&9.112.14.91:*)'
-----
-- SQL Stored Procedure
-----
```

```

P1: BEGIN
    -- Declare cursor
    DECLARE cursor1 CURSOR WITH RETURN FOR
        SELECT * FROM SYSIBM.SYSPROCEDURES;

    -- Cursor left open for client application
    OPEN cursor1;

END P1

```

워크스테이션 서버에서 SQL 저장 프로시저에 대한 빌드 옵션 설정

UNIX 및 Windows 플랫폼에서 DB2 저장 프로시저를 사용하면, 모든 SQL 저장 프로시저에 대해 빌드 옵션을 설정할 수 있습니다. 이 빌드 옵션은 다음과 같은 컴파일러 및 사전 처리 컴파일러 DB2 레지스트리 변수를 포함합니다.

- DB2_SQLROUTINE_PREPOPTS
- DB2_SQLROUTINE_COMPILER_PATH
- DB2_SQLROUTINE_COMPILE_COMMAND
- DB2_SQLROUTINE_KEEP_FILES

db2set 명령을 사용하여 이 레지스트리 변수를 설정하는 것이 가능하지만, Stored Procedure Builder를 사용하면 물리적으로 데이터베이스 서버를 액세스하여 명령을 실행하거나 변경사항이 유효하게 하기 위해 서버를 중지한 다음 재시작할 필요성을 줄입니다.

SQL 저장 프로시저 빌드 옵션 창을 열려면, 프로젝트 보기에서 데이터베이스 연결을 마우스 오른쪽 버튼으로 누르고, SQL 저장 프로시저 빌드 옵션을 누르십시오. 이 옵션 설정에 대한 자세한 정보는, DB2 저장 프로시저 도움말을 참조하십시오.

OS/390에서 빌드된 저장 프로시저의 WLM 주소 공간을 자동으로 새로 고침

WLM에서 수행할 OS/390에서 저장 프로시저를 정상적으로 빌드한 후에, DB2 저장 프로시저 빌더는 WLM 주소 공간을 자동으로 새로 고칩니다.

OS/390에서 Java 저장 프로시저 개발

DB2 Stored Procedure Builder는 OS/390용 DB2 UDB 버전 6 이상에서 Java 저장 프로시저 개발을 지원합니다. 이제 Java 저장 프로시저를 새로 작성하거나 기존 Java 저장 프로시저를 변경할 수 있습니다.

MQSeries 및 OLE DB용 DB2 테이블 사용자 정의 함수(UDF) 빌드

DB2 저장 프로시저는 MQSeries 및 OLE DB용 테이블 UDF 작성을 지원하는 마법사를 제공합니다. OLE DB 테이블 UDF 작성 마법사를 사용하여 OLE DB 데이터 공급자를 액세스할 수 있습니다. 이 마법사는 OLE 테이블 UDF 및 선택적 테이블 뷰를 작성합니다. MQSeries 테이블 UDF 작성 마법사를 사용하여 선택적 테이블 뷰가 있는 테이블 UDF를 작성한 다음 MQSeries 메시지를 액세스하고 테이블 형식으로 된 데이터를 분석할 수 있습니다.

유니코드 갱신

소개

유니코드 표준은 문자 및 텍스트에 대한 공통 문자 코드화 체계입니다. 유니코드는 문자의 복수 바이트 표현입니다. 국제적으로 텍스트 데이터 교환을 가능케 하고 전역 소프트웨어의 기초를 만드는 복수 언어 텍스트 코드화의 일관된 방법입니다.

유니코드는 다음의 두 코드화 체계를 제공합니다. 기본 코드화 체계는 UTF-16으로, 16 비트 코드화 체계입니다. UCS-2는 한 문자를 표현하기 위해 두 바이트를 사용하는 UTF-16의 부속 집합입니다. UCS-2는 일반적으로 모든 기존 하나 또는 두 바이트 코드 페이지로부터 모든 필요한 문자를 표시할 수 있는 보편적인 코드 페이지로 인정됩니다. UCS-2는 IBM에 코드 페이지 1200으로 등록되어 있습니다.

다른 유니코드 코드화 체계는 UTF-8로, 바이트 지향으로 기본 ASCII 기반 시스템에서 사용하기 쉽게 고안되었습니다. UTF-8은 다양한 수의 바이트(보통 1-3, 때로는 4)를 사용하여 각 문자를 저장합니다. 상이하지 않은 ASCII 문자는 한 바이트로 저장됩니다. 모든 기타 문자는 여러 바이트를 사용하여 저장됩니다. 일반적으로, 복수 바이트 코드 페이지를 위해 설계되지 않은 코드는 UTF-8 데이터를 확장 ASCII 데이터로 취급합니다. UTF-8은 IBM에 코드 페이지 1208으로 등록되어 있습니다.

데이터는 지역 코드 페이지, UCS-2와 UTF-8 사이에서 변환되므로 응용프로그램은 데이터의 요구사항을 고려해야 합니다. 예를 들어, 원래의 코드 페이지 및 사용된 문자에 따라 20 문자는 UCS-2에서 정확히 40 바이트 그리고 UTF-8에서는 20에서 60 바이트 사이의 바이트가 필요합니다.

DB2 유니코드 데이터베이스 및 응용프로그램

UTF-8 코드 세트로 작성된 Unix, Windows 또는 OS/2용 DB2 Universal Database를 사용하여 데이터를 UCS-2 및 UTF-8 형식 모두로 저장할 수 있습니다. 그런 데이터베이스를 유니코드 데이터베이스라고 합니다. SQL CHAR 데이터는 UTF-8을 사용하여 그리고 SQL GRAPHIC 데이터는 UCS-2를 사용하여 코드화됩니다. 1 바이트(SBCS) 및 복수 바이트(MBCS) 코드 세트를 CHAR 컬럼에 그리고 2 바이트(DBCS) 코드 세트를 GRAPHIC 컬럼에 저장하는 것과 같습니다.

응용프로그램의 코드 페이지는 DB2가 데이터 저장에 사용하는 코드 페이지와 일치하지 않을 수 있습니다. 비 유니코드 데이터베이스에서, 코드 페이지가 같지 않으면, 데이터베이스 관리 프로그램은 클라이언트와 서버 사이에 전송되는 문자 및 그래픽(순수 DBCS) 데이터를 변환합니다. 유니코드 데이터베이스에서는, 클라이언트 코드 페이지와

UTF-8 사이의 문자 데이터 변환이 데이터베이스 관리 프로그램에 의해 자동으로 수행
되지만, 모든 그래픽(UCS-2) 데이터는 클라이언트와 서버 사이의 변환없이 전달됩니다.

데이터베이스 관리 프로그램이 수행하는 코드 페이지 변환

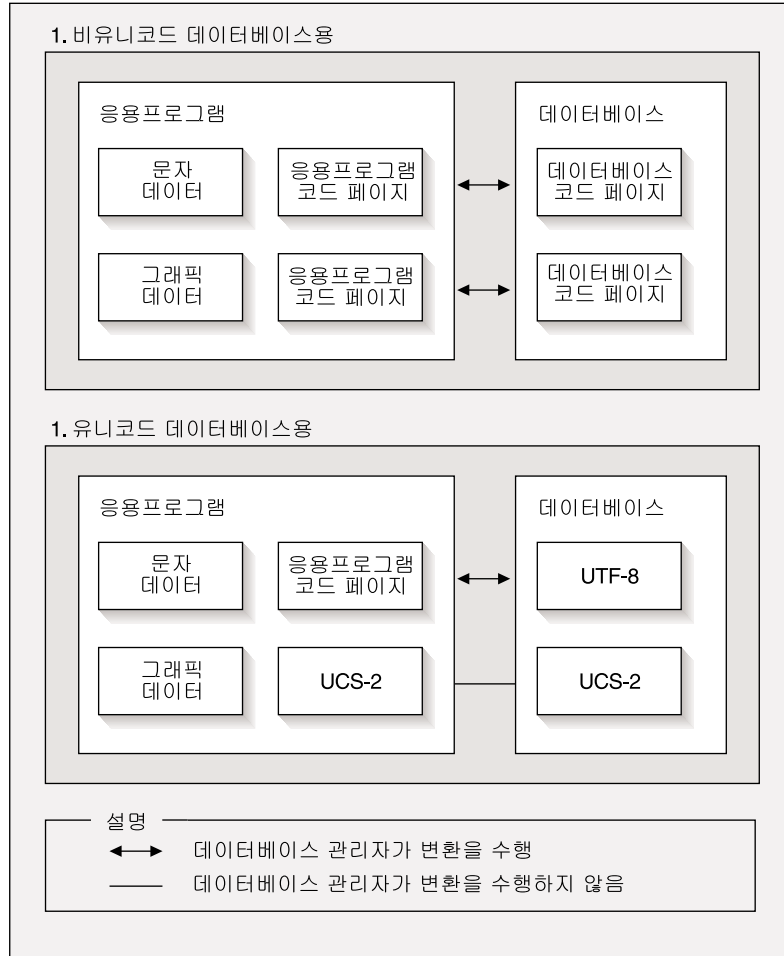


그림 7. 데이터베이스 관리 프로그램이 수행하는 코드 페이지 변환

주:

1. 유니코드 데이터베이스에 연결할 때, 응용프로그램이 DB2CODEPAGE=1208을 설정하고, 지역 코드 페이지가 UTF-8이면 아무 코드 페이지 변환도 필요하지 않습니다.
2. 유니코드 데이터베이스에 연결될 때, CLI 응용프로그램은 문자 데이터를 그래픽 데이터로 그리고 그래픽 데이터를 문자 데이터로 수신할 수도 있습니다.

응용프로그램이 UTF-8 코드 페이지를 지정하면, 모든 그래픽 데이터를 UCS-2로 그리고 문자 데이터를 UTF-8로 송수신함을 의미할 수 있습니다. 이 응용프로그램 코드 페이지는 유니코드 데이터베이스에만 지원됩니다.

유니코드 사용시 기타 고려사항:

1. 데이터베이스 코드 페이지는 데이터베이스가 작성될 때 결정되고, 기본으로 그 값은 운영 체제 로케일(또는 코드 페이지)로부터 결정됩니다. CODESET 및 TERRITORY 키워드를 사용하여 유니코드 DB2 데이터베이스를 명시적으로 작성할 수 있습니다. 예를 들어, 다음과 같습니다.

```
CREATE DATABASE unidb USING CODESET UTF-8 TERRITORY US
```

2. 또한 응용프로그램 코드 페이지는 지역 코드 페이지의 기본값을 설정하지만, 이는 두 방법 중 하나로 UTF-8에 의해 겹쳐써여 집니다.

- 다음 명령으로 응용프로그램 코드 페이지를 UTF-8(1208)로 설정하십시오.

```
db2set DB2CODEPAGE=1208
```

- CLI/ODBC 응용프로그램의 경우, SQLSetConnectAttr()를 호출하고 SQL_ATTR_ANSI_APP를 SQL_AA_FALSE로 설정하십시오. 기본 설정값은 SQL_AA_TRUE입니다.

3. GRAPHIC 컬럼의 데이터는 각 유니코드 문자에 대해 정확히 두 바이트를 차지하는 반면, CHAR 컬럼의 데이터는 각 유니코드 문자에 대해 1에서 3 바이트를 차지합니다. GRAPHIC 컬럼과 관련된 문자에서의 SQL 한계는 일반적으로 CHAR 컬럼에 대한 것의 반이지만, 바이트 면에서는 동등합니다. CHAR 컬럼의 최대 문자 길이는 254입니다. 그래픽 컬럼의 최대 문자 길이는 127입니다. 추가 정보는, SQL 참조서의 "기능" 장에서 MAX를 보십시오.

4. 그래픽 리터럴은 G 접두부로 인해 문자 리터럴과 차별화됩니다. 예를 들어, 다음과 같습니다.

```
SELECT * FROM mytable WHERE mychar = 'utf-8 data' AND mygraphic = G'ucs-2 data'
```

주: G 접두부는 유니코드 데이터베이스에 필요하지 않습니다.

추가 정보 및 갱신된 지원은 512 페이지의 『유니코드 데이터베이스의 리터럴』에서 참조하십시오.

5. CLI/ODBC 및 JDBC 응용프로그램에 대한 지원은 삽입된 응용프로그램에 대한 지원과 다릅니다. CLI/ODBC 지원에 특정한 정보는, 508 페이지의 『CLI 안내서 및 참조서』를 보십시오.

6. UCS-2 데이터의 바이트 순서는 플랫폼들 간에 다를 수 있습니다. 내부적으로, DB2는 big-endian 형식을 사용합니다.

문서 갱신

이 문서는 DB2 버전 7.1에서 유니코드 사용에 관한 다음 정보를 갱신합니다.

- *SQL 참조서*:

언어 요소

함수

- *CLI Guide and Reference*:

고급 기능 사용

DB2 CLI 및 ODBC

- 데이터 이동 유틸리티 안내 및 참조서, 내보내기/가져오기/로드 유틸리티 파일 형식
- DB2에서 유니코드 사용에 관한 추가 정보는, 관리 안내서, National Language Support (NLS): “DB2 UDB에서 유니코드/UCS-2 및 UTF-8 지원”을 보십시오.

SQL 참조서

언어 요소

데이터 유형의 승급

이 절에서 테이블 5는 각 데이터 유형에 대한 프로시저 목록을 보여줍니다. 다음에 주의하십시오.

1. 유니코드 데이터베이스의 경우, 다음은 데이터 유형과 같은 것으로 간주됩니다.
 - CHAR 및 GRAPHIC
 - VARCHAR 및 VARGRAPHIC
 - LONG VARCHAR 및 LONG VARGRAPHIC
 - CLOB 및 DBCLOB
2. 유니코드 데이터베이스에서, 예를 들어, foo(CHAR(8))과 foo(GRAPHIC(8))과 같이 함수 서명에 유일한 차이점이 있는 CHAR와 GRAPHIC 데이터 유형 사이에 있는 함수를 작성할 수 있습니다. 향후 릴리스로 이주할 때 이주를 진행하려면 그 중 하나를 삭제해야 하므로 이와 같은 중복 함수를 정의하지 말도록 강력히 권장합니다.

그런 중복 함수가 존재할 경우, 어느 것을 호출할 지는 두 전달 알고리즘에 의해 결정됩니다. 첫번째 전달은 비 유니코드 데이터베이스에서 함수 결정에 사용되는 것과 같은 알고리즘을 사용하여 일치하는 것을 찾으려 합니다. 일치하는 것을 찾을 수 없으면, CHAR 및 GRAPHIC 문자열에 대한 다음 승급 선행을 고려한 두 번째 전달이 수행됩니다.

GRAPHIC→CHAR→VARGRAPHIC→VARCHAR→LONG VARGRAPHIC→LONG VARCHAR→DBCLOB→CLOB

데이터 유형 사이의 유형 변환(CAST)

“구별 유형과 관련된 다음 유형 변환(CAST)이 지원됩니다”라고 소개된 목록에 다음 항목이 추가되었습니다.

- 유니코드 데이터베이스의 경우, VARCHAR 또는 VARGRAPHIC으로부터 소스 데이터 유형이 CHAR 또는 GRAPHIC인 구별 유형 DT로 유형 변환(CAST)하십시오.

다음은 표20 내장 데이터 유형 사이에서 지원되는 유형 변환(CAST)의 갱신입니다. 영향을 받는 표 행만 포함되었습니다.

표 31. 내장 데이터 유형 사이에서 지원되는 유형 변환(CAST)

목표 데이터 유형 →	C	V	L	C	G	V	L	D
	H	A	O	L	R	A	O	B
	A	R	N	O	A	R	N	C
	R	C	G	B	P	G	G	L
		H	V		H	R	V	O
		A	A		I	A	A	B
		R	R		C	P	R	
소스 데이터 유형 ↓			C			H	G	
			H			I	R	
			A			C	A	
			R				P	
							H	
							I	
							C	
CHAR	Y	Y	Y	Y	Y ¹	Y ¹	-	-
VARCHAR	Y	Y	Y	Y	Y ¹	Y ¹	-	-
LONGVARCHAR	Y	Y	Y	Y	-	-	Y ¹	Y ¹
CLOB	Y	Y	Y	Y	-	-	-	Y ¹
GRAPHIC	Y ¹	Y ¹	-	-	Y	Y	Y	Y
VARGRAPHIC	Y ¹	Y ¹	-	-	Y	Y	Y	Y
LONGVARGRAPHIC	-	-	Y ¹	Y ¹	Y	Y	Y	Y
DBCLOB	-	-	-	Y ¹	Y	Y	Y	Y

¹ 유형 변환(CAST)은 유니코드 데이터베이스에만 지원됩니다.

할당 및 비교

문자 및 그래픽 데이터 모두와 관련된 할당 및 비교는 문자열 중 하나가 리터럴일 때만 지원됩니다. 함수 결정의 경우, 그래픽 리터럴과 문자 리터럴은 모두 문자 및 그래픽 함수 매개변수에 일치합니다.

다음 표는 할당 및 비교를 위한 데이터 유형 호환성의 갱신입니다. 영향을 받는 표의 행 그리고 새로운 주석 ⁶만 포함되었습니다.

피연산자	2진 정수	소수	부동 소수점	문자열	그래픽 문자열	날짜	시간	시간소인	2진 문자열	UDT
문자열	아니오	아니오	아니오	예	예 ⁶	¹	¹	¹	아니오 ³	²
그래픽 문자열	아니오	아니오	아니오	예 ⁶	예	아니오	아니오	아니오	아니오	²

⁶ 유니코드 데이터베이스에만 지원됩니다.

문자열 할당:

저장영역 할당

이 서브섹션의 마지막 문단이 다음과 같이 수정되었습니다.

문자열이 고정 길이 컬럼에 할당되고 문자열의 길이가 목표의 길이 속성보다 작을 때, 문자열은 필요한 수의 1 바이트, 2 바이트 또는 UCS-2² 공백으로 오른쪽으로 패드(pad)됩니다. 컬럼이 FOR BIT DATA 속성으로 정의되었다더라도 패드 문자는 항상 공백입니다.

검색 할당

이 서브섹션의 세번째 문단이 다음과 같이 수정되었습니다.

문자열이 고정 길이 변수에 할당되고 문자열의 길이가 목표의 길이 속성보다 작을 때, 문자열은 필요한 수의 1 바이트, 2 바이트 또는 UCS-2² 공백으로 오른쪽으로 패드(pad)됩니다. 문자열 FOR BIT DATA 속성으로 정의되었다더라도 패드 문자는 항상 공백입니다.

² UCS-2는 다른 등록정보로 여러 SPACE 문자를 정의합니다. 유니코드 데이터베이스의 경우, 데이터베이스 관리 프로그램은 항상 ASCII SPACE를 위치 x'0020'에서 UCS-2 공백으로 사용합니다. EUC 데이터베이스의 경우, x'3000' 위치의 IDEOGRAPHIC SPACE가 패딩 GRAPHIC 문자열로 사용됩니다.

문자열 할당 변환 규칙

다음 문단이 이 서브섹션 끝에 추가되었습니다.

유니코드 데이터베이스의 경우, 문자열은 그래픽 컬럼에 할당될 수 있고 그래픽 문자열은 문자 컬럼에 할당될 수 있습니다.

그래픽 문자열 할당을 위한 DBCS 고려사항

이 서브섹션의 첫번째 문단이 다음과 같이 수정되었습니다.

그래픽 문자열 할당은 문자열에 대한 것과 비슷한 방식으로 처리됩니다. 비 유니코드 데이터베이스의 경우, 그래픽 문자열 데이터 유형은 다른 그래픽 문자열 데이터 유형과만 호환되고, 숫자, 문자열 또는 날짜시간 데이터 유형과는 호환되지 않습니다. 유니코드 데이터베이스의 경우, 그래픽 문자열 데이터 유형은 문자열 데이터 유형과 호환됩니다.

문자열 비교:

비교를 위한 변환 규칙

이 서브섹션은 다음과 같이 수정되었습니다.

두 문자열이 비교될 때, 필요하면 문자열 중 하나가 코드화 체계 또는 다른 문자열의 코드 페이지로 먼저 변환됩니다. 세부사항은, SQL 참조서의 언어 요소의 “문자열 변환 규칙”을 보십시오.

결과 데이터 유형 규칙

유니코드 데이터베이스의 문자 및 그래픽 문자열: 이것은 서브섹션 "그래픽 문자열" 뒤에 삽입되는 새로운 서브섹션입니다.

유니코드 데이터베이스에서, 문자열 및 그래픽 문자열은 호환됩니다.

피연산자가 ...이고	기타 피연산자가 ...일 경우	결과 데이터 유형은 ...입니다
GRAPHIC(x)	CHAR(y) 또는 GRAPHIC(y)	GRAPHIC(z) 여기서 $z = \max(x,y)$
VARGRAPHIC(x)	CHAR(y) 또는 VARCHAR(y)	VARGRAPHIC(z) 여기서 $z = \max(x,y)$
VARCHAR(x)	GRAPHIC(y) 또는 VARGRAPHIC(y)	VARGRAPHIC(z) 여기서 $z = \max(x,y)$
LONG VARGRAPHIC(x)	CHAR(y) 또는 VARCHAR(y)	LONG VARGRAPHIC(z) 여기서 $z = \max(x,y)$
LONG VARCHAR(x)	GRAPHIC(y) 또는 VARGRAPHIC(y)	LONG VARGRAPHIC(z) 여기서 $z = \max(x,y)$
DBCLOB(x)	CHAR(y) 또는 VARCHAR(y)	DBCLOB(z) 여기서 $z = \max(x,y)$
DBCLOB(x)	LONG VARCHAR(y)	DBCLOB(z) 여기서 $z = \max(x,16350)$
CLOB(x)	GRAPHIC(y) 또는 VARGRAPHIC(y)	DBCLOB(z) 여기서 $z = \max(x,y)$
CLOB(x)	LONG VARGRAPHIC(y)	DBCLOB(z) 여기서 $z = \max(x,16350)$

문자열 변환 규칙

세번째 항목이 이 섹션의 다음 목록에 추가되었습니다.

각 코드 페이지 쌍에 대해, 결과는 다음 규칙을 순차적으로 적용하여 결정됩니다.

- 코드 페이지가 같으면, 결과는 그 코드 페이지입니다.
- 한 코드 페이지가 BIT DATA(코드 페이지 0)이면, 결과 코드 페이지는 BIT DATA입니다.
- 유니코드 데이터베이스에서, 한 코드 페이지가 다른 코드 페이지와 다른 코드화 체계로 데이터를 표시하면, 결과는 UTF-8 위의 UCS-2입니다(즉, 문자 데이터 유형 위의 그래픽 데이터 유형입니다).¹
- 그렇지 않으면, 결과 코드 페이지는 SQL 참조서의 언어 요소의 "문자열 변환 규칙" 테이블 8에 의해 결정됩니다. 테이블의 '첫번째' 항목은 첫번째 피연산자의 코드 페이지가 선택되었음을 의미하고 '두 번째'는 두 번째 피연산자의 코드 페이지가 선택되었음을 의미합니다.

¹ 비 유니코드 데이터베이스에서, 다른 코드화 체계 사이의 변환은 지원되지 않습니다.

표현식

다음에 추가되었습니다.

유니코드 데이터베이스에서, 문자 또는 그래픽 문자열을 승인하는 표현식은 변환이 지원되는 모든 문자열 유형을 승인합니다.

병합 연산자: 다음이 이 서브섹션 끝에 추가되었습니다.

유니코드 데이터베이스에서, 문자열 피연산자와 그래픽 문자열 피연산자 모두와 관련된 병합은 먼저 문자 피연산자를 그래픽 피연산자로 변환합니다. 비 유니코드 데이터베이스에서, 병합에 문자 및 그래픽 피연산자 모두 포함될 수 없음을 주의하십시오.

술어

"다음 규칙이 모든 술어 유형에 적용됩니다"로 소개된 목록에 다음 항목이 추가되었습니다.

- 유니코드 데이터베이스에서, 문자 또는 그래픽 문자열을 승인하는 모든 술어는 변환이 지원되는 모든 문자열 유형을 승인합니다.

함수

스칼라 함수

다음 문장이 이 섹션 끝에 추가되었습니다.

유니코드 데이터베이스에서, character GRAPHIC 또는 graphic 문자열을 승인하는 모든 스칼라 함수는 변환이 지원되는 이들 문자열을 승인합니다.

CLI 안내서 및 참조서

고급 기능 사용

다음은 이 장의 새로운 절입니다.

DB2 CLI 유니코드 응용프로그램 작성

DB2 CLI 유니코드 응용프로그램 지원의 두 주요 영역이 있습니다.

1. ANSI 문자열 인수 대신에 유니코드 문자열 인수를 승인할 수 있는 함수 세트 추가.
2. 데이터를 ANSI 또는 유티코드 데이터로 설명하는 새로운 C 및 SQL 데이터 유형 추가. 다음 절에는 이들 두 영역에 대한 추가 정보가 있습니다. 유니코드 응용프로그램으로 취급되려면, 연결하기 전에 응용프로그램이 SQL_ATTR_ANSI_APP 연결 속성을 SQL_AA_FALSE로 설정해야 합니다. 이는 CLI가 유니코드 클라이언트로서 연결되도록 보장하고, 모든 유니코드 데이터는 CHAR 데이터의 경우 UTF-8로 또는 GRAPHIC 데이터의 경우 UCS-2로 보내집니다.

유니코드 함수: 다음은 유니코드(W) 및 ANSI(A) 버전을 모두 지원하는 ODBC API 함수의 목록입니다(유니코드의 경우 함수 이름에 W가 있습니다).

SQLBrowseConnect	SQLForeignKeys	SQLPrimaryKeys
SQLColAttribute	SQLGetConnectAttr	SQLProcedureColumns
SQLColAttributes	SQLGetConnectOption	SQLProcedures
SQLColumnPrivileges	SQLGetCursorName	SQLSetConnectAttr
SQLColumns	SQLGetDescField	SQLSetConnectOption
SQLConnect	SQLGetDescRec	SQLSetCursorName
SQLDataSources	SQLGetDiagField	SQLSetDescField
SQLDescribeCol	SQLGetDiagRec	SQLSetStmtAttr
SQLDriverConnect	SQLGetInfo	SQLSpecialColumns
SQLDrivers	SQLGetStmtAttr	SQLStatistics
SQLError	SQLNativeSQL	SQLTablePrivileges
SQLExecDirect	SQLPrepare	SQLTables

문자열 길이인수를 리턴 또는 취하는 유니코드 함수는 문자수로서 전달됩니다. 서버 데이터에 대한 길이 정보를 리턴하는 함수의 경우, 표시 크기와 정밀도는 문자 수로 설명합니다. 길이(데이터의 전송 크기)가 문자열 또는 비문자열 데이터를 참조할 수 있는 경우, 길이는 8진수로 설명됩니다. 예를 들어, `SQLGetInfoW`는 길이를 바이트수로 취하지만, `SQLExecDirectW`는 문자수를 사용합니다. CLI는 응용프로그램의 바인딩에 따라 결과 세트를 유니코드 또는 ANSI로 리턴합니다. 응용프로그램이 `SQL_C_CHAR`에 바인드할 경우, 드라이버는 `SQL_WCHAR` 데이터를 `SQL_CHAR`로 변환합니다. 드라이버 관리 프로그램은 ANSI 드라이버를 위해 `SQL_C_WCHAR`을 `SQL_C_CHAR`에 맵핑하지만 유니코드 드라이버를 위해서는 맵핑하지 않습니다.

새로운 데이터 유형 및 유효한 변환: 두 새로운 CLI 또는 ODBC 정의 데이터 유형, `SQL_C_WCHAR`과 `SQL_WCHAR`이 있습니다. `SQL_C_WCHAR`은 C 버퍼에 UCS-2 데이터가 있음을 의미합니다. `SQL_WCHAR`은 특정 컬럼 또는 매개변수 표시문자에 유니코드 데이터가 들어 있음을 의미합니다. DB2 유니코드 서버의 경우, 그래픽 컬럼은 `SQL_WCHAR`로 설명됩니다. 변환은 그래픽 데이터 유형 사이에서 뿐만 아니라 `SQL_C_WCHAR`과 `SQL_CHAR`, `SQL_VARCHAR`, `SQL_LONGVARCHAR` 및 `SQL_CLOB` 사이에서 허용됩니다.

표 32. 지원되는 데이터 변환

SQL 데이터 유형	S Q L - C - C H A R	S Q L - C - W C H A R	S Q L - C - L O N G	S Q L - C - S H O R T	S Q L - C - T I N Y I N T	S Q L - C - F L O A T	S Q L - C - D O U B L E	S Q L - C - T Y P E - D A T E	S Q L - C - T Y P E - T I M E	S Q L - C - T Y P E - T I M E S T A M P	S Q L - C - B I N A R Y	S Q L - C - B I T	S Q L - C - D B C H A R	S Q L - C - C L O B - L O C A T O R	S Q L - C - B L O B - L O C A T O R	S Q L - C - D B C L O B - L O C A T O R	S Q L - C - B I G I N T	S Q L - C - N U M E R I C
BLOB	X	X									D				X			
CHAR	D	X	X	X	X	X	X	X	X	X	X	X					X	X
CLOB	D	X									X			X				
DATE	X	X						D		X								
DBCLOB		X									X		D			X		
DECIMAL	D	X	X	X	X	X	X				X	X					X	X
DOUBLE	X	X	X	X	X	X	D					X					X	X
FLOAT	X	X	X	X	X	X	D					X					X	X
GRAPHIC(비 유니코드)	X	X											D					
GRAPHIC(유니코드)	X	X	X	X	X	X	X	X	X	X	X	X	D				X	
INTEGER	X	X	D	X	X	X	X					X					X	X
LONG VARCHAR	D	X									X							
LONG VARGRAPHIC (비 유니코드)	X	X									X		D					
LONG VARGRAPHIC(유니코드)	X	X									X		D					
NUMERIC	D	X	X	X	X	X	X					X						X
REAL	X	X	X	X	X	D	X					X						X
SMALLINT	X	X	X	D	X	X	X					X					X	X
BIGINT	X	X	X	X	X	X	X				X	X					D	X
TIME	X	X							D	X								
TIMESTAMP	X	X						X	X	D								

표 32. 지원되는 데이터 변환 (계속)

SQL 데이터 유형	S Q L - C - C H A R	S Q L - C - W C H A R	S Q L - C - L O N G	S Q L - C - S H O R T	S Q L - C - T I N Y I N T	S Q L - C - F L O A T	S Q L - C - D O U B L E	S Q L - C - T Y P E - D A T E	S Q L - C - T Y P E - T I M E	S Q L - C - T Y P E - T I M E S T A M P	S Q L - C - B I N A R Y	S Q L - C - B I T	S Q L - C - D B C H A R	S Q L - C - C L O B	S Q L - C - B L O B	S Q L - C - D B C L O B	S Q L - C - B I G I N T	S Q L - C - N U M E R I C	
VARCHAR	D	X	X	X	X	X	X	X	X	X	X	X						X	X
VARGRAPHIC (비 유니코드)	X	X											D						
VARGRAPHIC(유니코드)	X	X	X	X	X	X	X	X	X	X	X	X	D					X	

주:

D 변환이 지원됩니다. SQL 데이터 유형에 대한 기본 변환입니다.

X 모든 IBM DBMS가 변환을 지원합니다.

공백 IBM DBMS가 변환을 지원하지 않습니다.

- 데이터가 LOB 위치 지정자로 변환되지 않고, 위치 지정자는 데이터 값을 나타냅니다. 추가 정보는 대형 오브젝트(LOB) 사용을 참조하십시오.
- SQL_C_NUMERIC은 32 비트 Windows 운영 체제에서만 사용 가능합니다.

폐기된 키워드/매치 값: 유니코드 응용프로그램이 지원되기 전에, 1 바이트 문자 데이터로 작업하도록 작성된 응용프로그램은 GRAPHIC=1,2 또는 3, Patch2=7 등과 같이, 일련의 cliini 파일 키워드에 의해 2 바이트 그래픽 데이터로 작업하도록 만들 수 있었습니다.

이들 키워드는 유니코드 응용프로그램에 더 이상 필요하지 않으며, 실제로 심각한 부작용이 있으므로 사용해서는 안됩니다. 이를 알고 있지 않으면, 특정 응용프로그램이 유니코드 응용프로그램일 경우, 그래픽 데이터 처리에 영향을 주는 키워드 없이 시도하도록 권장합니다.

유니코드 데이터베이스의 리터럴: 비 유니코드 데이터베이스에서 LONG VARCHAR 및 LONG VARCHAR 컬럼의 데이터는 비교될 수 없습니다. 내재된 코드 페이지 변환이 지원되지 않으므로 GRAPHIC/VARGRAPHIC 및 CHAR/VARCHAR 컬럼의 데이터는 명시적 유형변환(cast) 함수를 사용하여 서로 비교하거나 할당될 수만 있었습니다. 여기에는 GRAPHIC/VARGRAPHIC 리터럴이 CHAR/VARCHAR 리터럴과 G 접두부에 의해 차별화되는 GRAPHIC/VARGRAPHIC 및 CHAR/VARCHAR 리터럴이 포함됩니다.

유니코드 데이터베이스의 경우, GRAPHIC/VARGRAPHIC과 CHAR/VARCHAR 리터럴 사이의 유형변환(CAST)은 필요하지 않습니다. 또한, G 접두부가 GRAPHIC/VARGRAPHIC 리터럴 앞에 필요하지 않습니다. 최소한 한 인수가 리터럴이면, 내재된 변환이 발생합니다. 이는 G 접두부가 있거나 없는 리터럴을 SQLPrepareW() 또는 SQLExecDirect()를 사용하는 명령문 내에서 사용할 수 있도록 해줍니다. LONG VARCHAR의 리터럴은 G 접두부를 가져야합니다.

자세한 정보는, *SQL 참조서*의 언어 요소에서 "데이터 유형 사이의 유형변환(cast)"을 보십시오.

새로운 CLI 구성 키워드: 다음 세 개의 키워드를 추가하여 유니코드 응용프로그램이 데이터베이스에 연결될 때 여분의 오버헤드를 피할 수 있도록 하였습니다.

1. DisableUnicode

키워드 설명

유니코드에 대한 기본 지원 사용안함

db2cli.ini 키워드 구문

DisableUnicode = 0 | 1

기본 설정

0 (false)

DB2 CLI/ODBC 설정 탭

이 키워드는 CLI/ODBC 설정 노트북을 사용하여 설정할 수 없습니다. 이 키워드를 사용하려면 db2cli.ini 파일을 직접 수정해야 합니다.

사용법 주의사항:

유니코드 지원이 사용 가능하고, 유니코드 응용프로그램이 호출한 경우, 코드 페이지 변환에 의한 불필요한 자료 손실이 없도록 보장하기 위하여 CLI는 최선의 클라이언트 코드 페이지를 사용하여 데이터베이스에 연결하려고 시도합니다. 이는 코드 페이지가 교환되므로 연결 시간을 증가시키거나 또는 이러한 지원이 추가되기 이전에는 발생치 않았던 클라이언트에서의 코드 페이지 변환을 초래할 수 있습니다.

이 키워드를 True로 설정하면, 데이터가 서버에 송신되기 이전에 우선적으로 모든 유니코드 데이터가 응용프로그램의 국지 코드 페이지로 변환되도록 합니다. 이는 국지 코드 페이지에서 대표될 수 없는 데이터의 데이터 손실을 초래할 수 있습니다.

2. ConnectCodepage

키워드 설명

여분의 연결 오버헤드를 피하기 위해 데이터 소스에 연결하는 경우, 사용할 구체적 코드 페이지를 지정합니다.

db2cli.ini 키워드 구문

ConnectCodepage = 0 | 1 <any valid db2 codepage>

기본 설정

0

DB2 CLI/ODBC 설정 탭

이 키워드는 CLI/ODBC 설정 노트북을 사용하여 설정할 수 없습니다. 이 키워드를 사용하려면 db2cli.ini 파일을 직접 수정해야 합니다.

사용법 주의사항:

비 유니코드 응용프로그램은 언제나 응용프로그램의 지역 코드페이지 또는 DB2Codepage 환경설정을 사용하여 데이터베이스에 연결합니다. 기본값으로 CLI는 유니코드 응용프로그램이 UTF-8 및 UCS-2 코드 페이지를 사용하여 유니코드 데이터베이스에 연결되도록 보장하며, 데이터베이스의 코드 페이지를 사용하여 비 유니코드 데이터베이스에 연결됩니다. 이는 코드 페이지 변환에 따른 불필요한 자료 손실이 발생하지 않도록 보장합니다.

연결시에 여분의 오버헤드를 피하기 위하여 비 유니코드 데이터베이스에 연결하는 경우, 사용자는 이 키워드를 사용하여 데이터베이스의 코드 페이지를 지정할 수 있습니다.

값 1을 지정하여 SQLDriverConnect()가 출력 연결 문자열에서 올바른 값을 리턴하도록 하여 해당 값이 미래의 SQLDriverConnect() 호출에 사용될 수 있도록 합니다.

3. 유니코드 서버

키워드 설명

데이터 소스가 유니코드 서버임을 표시 설정 ConnectCodepage=1208 과 동일.

db2cli.ini 키워드 구문

UnicodeServer = 0 | 1

기본 설정

0

DB2 CLI/ODBC 설정 탭

이 키워드는 CLI/ODBC 설정 노트북을 사용하여 설정할 수 없습니다. 이 키워드를 사용하려면 db2cli.ini 파일을 직접 수정해야 합니다.

사용법 주의사항:

이 키워드는 ConnectCodepage=1208와 동일하며 편리성을 위해 추가하였습니다. OS/390 버전 7 또는 그 이상의 DB2에 연결하는 경우, 이 키워드를 사용하여 여러분의 연결 오버헤드를 피할 수 있습니다. 여러분의 처리가 필요하지 않으므로, 이 키워드를 Windows의 DB2, Unix의 DB2 또는 OS/2 데이터베이스의 DB2에 사용할 필요는 없습니다.

DB2 CLI 및 ODBC

다음은 추가된 새 절입니다.

ODBC 유니코드 응용프로그램

유니코드 ODBC 응용프로그램은 주로 UCS-2로 문자 데이터를 전송하고 검색합니다. ODBC 함수('W' 접두부)의 유니코드 버전을 호출하여 그리고 유니코드 데이터 유형을 지시하여 이를 수행합니다. 응용프로그램은 명시적으로 지역 코드 페이지를 지정하지 않습니다. 응용프로그램은 계속 ANSI 함수를 호출하고 지역 코드 페이지 문자열을 전달합니다.

예를 들어, 응용프로그램은 SQLConnectW()를 호출하고 DSN 사용자 ID와 암호를 유니코드 인수로 전달합니다. 그런 뒤 SQLExecDirectW()를 호출하고 유니코드 SQL문에 전달한 뒤, ANSI 지역 코드 페이지 버퍼(SQL_C_CHAR)와 유니코드 버퍼(SQL_C_WCHAR)의 조합을 바인드합니다. 데이터베이스 데이터 유형은 지역 코드 페이지 또는 UCS-2 및 UTF-8이어야 합니다.

CLI 응용프로그램이 SQL_ATTR_ANSI_APP를 SQL_AA_FALSE로 설정하여 SQLConnectW를 호출하거나 SQLSetConnectAttr을 호출할 경우, 응용프로그램은 유니코드 응용프로그램으로 간주됩니다. 이는 모든 CHAR 데이터가 UTF-8 형식으로 데이터베이스로 송수신됨을 의미합니다. 그런 뒤 응용프로그램은 CHAR 데이터를 지역 코드 페이지의 SQL_C_CHAR 버퍼로(데이터 유실 가능), 데이터 유실없이 UCS-2의 SQL_C_WCHAR 버퍼로 가져옵니다.

응용프로그램이 위 두 호출 중 어느 것도 수행하지 않으면, CHAR 데이터는 서버에서 응용프로그램 지역 코드 페이지로 변환됩니다. 이는 SQL_C_WCHAR로 가져온 CHAR 데이터에 데이터 유실이 있을 수 있음을 의미합니다.

DB2CODEPAGE 인스턴스 변수가 코드 페이지 1208(UTF-8)로 설정되어 있으면 (db2set을 사용하여), 응용프로그램은 UTF-8이 이제 지역 코드 페이지이므로 모든 CHAR 데이터를 UTF-8로 수신합니다. 응용프로그램은 또한 모든 CHAR 입력 데이

터 역시 UTF-8인지 확인해야 합니다. 또한 ODBC는 모든 SQL_C_WCHAR 데이터가 원시 endian 형식이라고 가정합니다. CLI는 SQL_C_WCHAR에 필요한 바이트 리버설을 수행합니다.

ODBC 유니코드 대 비 유니코드 응용프로그램: 이 DB2 Universal Database 릴리스에는 SQLConnectW() API가 포함되어 있습니다. 유니코드 드라이버는 드라이버 관리 프로그램이 유니코드 드라이버로 인식하는 순서대로 SQLConnectW를 내보내야 합니다. 많은 응용프로그램(Microsoft Access 및 Visual Basic과 같이)이 SQLConnectW()를 호출함을 유의해야 합니다. 이전 릴리스의 DB2 Universal Database에서는, DB2 CLI가 이 API를 지원하지 않았고, 따라서 ODBC 드라이버에 의해 유니코드 드라이버로 인식될 수 없었습니다. 이는 ODBC 드라이버 관리 프로그램이 모든 유니코드 데이터를 응용프로그램의 지역 코드 페이지로 변환하도록 만들었습니다. SQLConnectW() 함수의 지원이 추가되어, 이들 응용프로그램이 이제 유니코드 응용프로그램으로 연결되고 DB2 CLI가 모든 필요한 데이터 변환을 관장합니다.

DB2 CLI는 이제 유니코드 API(접미부 "W") 및 정규 ANSI API를 승인합니다. ODBC는 접미부 "A"로 함수 세트를 정의하지만, 드라이버 관리 프로그램은 접미부가 "A"인 ANSI 함수를 드라이버로 전달하지 않습니다. 대신, 이 함수들을 접미부가 없는 ANSI 함수 호출로 변환한 뒤, 그것을 드라이버로 전달합니다.

SQLConnectW() API를 호출하는 ODBC 응용프로그램을 유니코드 응용프로그램이라고 합니다. ODBC 드라이버 관리 프로그램은 호출된 응용프로그램 버전에 상관없이 항상 SQLConnectW() API를 호출하므로, ODBC는 응용프로그램이 ANSI 또는 UNICODE 응용프로그램으로 취급되어야 할 경우 드라이버에게 알리기 위해 SQL_ATTR_ANSI_APP 연결 속성을 도입했습니다. SQL_ATTR_ANSI_APP가 SQL_AA_FALSE로 설정되어 있지 않으면, DB2 CLI는 모든 유니코드 데이터를 서버로 보내기 전에 지역 코드 페이지로 변환합니다.

데이터 이동 유틸리티 안내 및 참조서

내보내기/가져오기/로드 유틸리티 파일 형식

다음 갱신사항이 추가되었습니다.

내보내기, 가져오기 및 로드 유틸리티는 비 유니코드 데이터베이스에 연결된 유니코드 클라이언트에서 사용될 때 지원되지 않습니다. 유니코드 클라이언트 파일은 유니코드 클라이언트가 유니코드 데이터베이스에 연결된 경우에만 지원됩니다.

제7부 호스트 시스템에 연결

연결성 보충 설명서	519
VM 환경에서 응용프로그램 서버 설정	519
CLI/ODBC/JDBC 구성 PATCH1 및 PATCH2 설정	519

연결성 보충 설명서

VM 환경에서 응용프로그램 서버 설정

"네트워크 정보 제공" 절, "응용프로그램 서버 정의" 하위 절의 하나의 문장 다음에 다음 문장을 추가하십시오.

RDB_NAME은 SQLSTART EXEC에서 DBNAME 매개변수로 제공됩니다.

CLI/ODBC/JDBC 구성 PATCH1 및 PATCH2 설정

CLI/ODBC/JDBC 드라이버는 클라이언트 구성 지원 프로그램이나 ODBC 드라이버 관리자(시스템에 설치된 경우)를 통해 구성하거나 db2cli.ini 파일을 수동으로 편집하여 구성할 수 있습니다. 설치 및 구성 보충 설명서 또는 *CLI Guide and Reference*에서 자세한 내용을 참조하십시오.

DB2 CLI/ODBC 드라이버 기본 활동은 db2cli.ini 파일을 통해서나 SQLDriverConnect() 또는 SQLBrowseConnect() CLI API를 통해서 PATCH1 및 PATCH2 키워드 모두에 값을 지정하여 수정할 수 있습니다.

PATCH1 키워드는 사용자가 설정하려는 모든 키워드를 지정합니다. 예를 들어, 패치 1, 2 및 8이 지정된 경우, PATCH1의 값은 11입니다. 다음은 각 키워드 값의 설명과 드라이버에 미치는 영향입니다.

- 1 - "count(exp)"에 대한 드라이버 검색을 하고 이를 "count(distinct exp)"로 대체합니다. 일부 DB2 버전에서 "count(exp)" 구문을 지원하고, 그 구문이 일부 ODBC 응용프로그램에서 생성되기 때문에 필요합니다. 서버가 "count(exp)" 구문을 지원하지 않을 경우 Microsoft 응용프로그램에서 필요로 합니다.
- 2 - LITERAL PREFIX 또는 LITERAL SUFFIX 컬럼에 대한 SQLGetTypeInfo() 함수에 SQL NULL DATA가 리턴될 때 몇몇 ODBC 응용프로그램이 트랩됩니다. 이는 드라이버가 빈 문자열을 대신 리턴하도록 강제 수행합니다. Impromptu 2.0에서 필요합니다.
- 4 - 시간소인에서 시간과 분수 부분이 0일 경우, 드라이버가 입력 시간소인 데이터를 날짜 데이터로 처리하도록 강제 수행합니다. Microsoft Access에서 필요합니다.
- 8 - 시간소인에서 날짜 부분이 1899-12-30일 경우, 드라이버가 입력 시간소인 데이터를 시간 데이터로 처리하도록 강제 수행합니다. Microsoft Access에서 필요합니다.
- 16 - 사용되지 않습니다.
- 32 - 드라이버가 SQL_LONGVARCHAR, SQL_LONGVARIABLE 및 SQL_LONGVARGRAPHIC 컬럼에 대한 정보를 리턴하지 않도록 강제 수행합니다. 응용프로그램에게는 긴 필드가 지원되지 않는 것처럼 나타납니다. Lotus 123에서 필요합니다.
- 64 - 드라이버가 그래픽 출력 문자열을 널(NULL) 종료하도록 강제 수행합니다. 2 바이트 환경에서 Microsoft Access에 필요합니다.
- 128 - 드라이버가 "SELECT Config, nValue FROM MSystemConf" 조회가 서버로 가게 하도록 강제 수행합니다. 현재 드라이버는 연관된 SQLSTATE 값 S00002(테이블을 찾을 수 없음)의 오류를 리턴합니다. 사용자가 이 구성 테이블을 데이터베이스에 작성하고 응용프로그램이 그것에 액세스하도록 할 경우 필요합니다.
- 256 - 드라이버가 SQLStatistics() 호출에서 기본 키 컬럼을 먼저 리턴하도록 강제 수행합니다. 현재, 드라이버는 표준 ODBC 동작인 색인 이름별로 정렬된 색인들을 리턴합니다.

- 512 - 드라이버가 SQL API SQLTABLEPRIVILEGES 및 SQL API SQLCOLUMNPRIVILEGES SQL API SQLTABLEPRIVILEGES 및 SQL API SQLCOLUMNPRIVILEGES.
- 1024 - 드라이버가 실행된 UPDATE 또는 DELETE 문이 어떤 행에도 영향을 못미칠 경우, SQLExecute() 또는 SQLExecDirect()에 SQL_NO_DATA_FOUND 대신 SQL_SUCCESS를 리턴하도록 강제 수행합니다. Visual Basic 응용프로그램에 필요합니다.
- 2048 - 사용되지 않습니다.
- 4096 - 드라이버가 자동 확약 모드일 때 커서를 닫은 후 COMMIT문을 발행하지 못하도록 강제 수행합니다.
- 8192 - 드라이버가 저장 프로시저를 호출한 후 추가 결과 세트를 리턴하도록 강제 수행합니다. 이 결과 세트는 한 행의 결과 세트로, 저장 프로시저의 출력 값들로 구성됩니다. Powerbuild 응용프로그램에서 액세스할 수 있습니다.
- 32768 - 드라이버가 Microsoft Query 응용프로그램을 DB2 MVS 동의어에 대해 작업하도록 강제 수행합니다.
- 65536 - 드라이버가 사실상 그래픽 리터럴인 문자 리터럴 앞에 수동으로 "G"를 삽입하도록 강제 수행합니다. 이 패치는 2 바이트 환경에서 작업할 때 항상 제공되어야 합니다.
- 131072 - 시간소인 컬럼이 고유 색인의 일부일 경우 드라이버가 대신 시간소인 컬럼을 CHAR(26) 컬럼으로 설명하도록 강제 수행합니다. Microsoft 응용프로그램에서 필요합니다.
- 262144 - 드라이버가 SYSCAT.PROCEDURES 및 SYSCAT.PROCPARMS 테이블 대신 의사 카탈로그 테이블인 db2cli.procedures를 사용하도록 강제 수행합니다.
- 524288 - DB2/400 V3.x 시스템에 대한 시스템 테이블을 조회할 때 드라이버가 TABLE SCHEMA 대신 SYSTEM TABLE SCHEMA를 사용하도록 강제 수행합니다. 그 결과 성능이 향상됩니다.
- 1048576 - 드라이버가 길이가 0인 문자열을 SQLPutData()를 통해 SQL_NULL_DATA로 처리하도록 강제 수행합니다.

PATCH2 키워드는 PATCH1 키워드와 다릅니다. 이 경우, 쉼표 구분 문자를 사용하여 여러 패치를 지정합니다. 예를 들어, 패치 1, 4 및 5가 지정된 경우, PATCH2의 값은 "1,4,5"입니다. 다음은 각 키워드 값의 설명과 드라이버에 미치는 영향입니다.

- 1 - 드라이버가 CALL문에서 저장 프로시저의 이름을 대문자로 바꾸도록 강제 수행합니다.
- 2 - 사용되지 않습니다.
- 3 - 드라이버가 스키마 호출에 대한 모든 인수를 대문자로 변환하도록 강제 수행합니다.
- 4 - 드라이버가 스키마 호출에 대해 버전 5 유사 결과 세트 대신 버전 2.1.2 유사 결과 세트(즉, SQLColumns(), SQLProcedureColumns() 등)를 리턴하도록 강제 수행합니다.
- 5 - 드라이버가 입력 VARCHAR 컬럼의 처리를 최적화하지 않도록 강제 수행합니다. 이 때, 데이터에 대한 포인터와 길이에 대한 포인터는 메모리에서 연속됩니다.
- 6 - 드라이버가 화면 이동이 가능한 커서가 지원되지 않는 메시지를 리턴하도록 강제 수행합니다. DB2 클라이언트가 버전 5이고 서버가 DB2 UDB 버전 5일 경우 Visual Basic 프로그램에서 필요합니다.
- 7 - 드라이버가 모든 GRAPHIC 컬럼 데이터 유형을 CHAR 컬럼 데이터 유형에 맵핑하도록 강제 수행합니다. 이는 2 바이트 환경에서 필요합니다.
- 8 - 드라이버가 스키마 호출에서 카탈로그 검색 인수를 무시하도록 강제 수행합니다.
- 9 - 커서의 초기 단기에 대해 확약하지 않습니다.
- 10 - 사용되지 않습니다.
- 11 - 카탈로그 이름이 지원됨을 보고합니다(VB 저장 프로시저).
- 12 - 스키마 호출 인수에서 큰 따옴표를 제거합니다(Visual Interdev).
- 13 - db2cli.ini에서 출력 연결 문자열에 키워드를 추가하지 않습니다.
- 14 - SQLProcedures() 및 SQLProcedureColumns()에서 스키마 이름을 무시합니다.
- 15 - 문자 출력에서 소수점에 대해 항상 마침표를 사용합니다.
- 16 - 열 때마다 설명 정보를 리턴하도록 강제 수행합니다.
- 17 - 설명에서 컬럼 이름을 리턴하지 않습니다.
- 18 - 리터럴을 매개변수 표시문자로 대체하려고 합니다.
- 19 - 현재, DB2 MVS V4.1은 외부 조인 절의 ON 절에서 괄호가 허용되는 ODBC 구문을 지원하지 않습니다. 이 PATCH2를 켜면 IBM DB2 ODBC 드라이버는 외부 조인 절이 ODBC escape 순서에 있을 때 괄호를 제거합니다. 이 PATCH2는 DB2 MVS 4.1대해서 일때만 사용해야 합니다.
- 20 - 현재, MVS의 DB2는 매개변수 표시문자가 모두 피연산자인 BETWEEN 술어(expression ? BETWEEN ?)를 지원하지 않습니다. 이 패치를 켜면 IBM ODBC 드라이버가 술어를 (expression >= ? and expression <= ?)로 재작성하도록 만듭니다.
- 21 - 저장 프로시저에 대한 모든 OUTPUT을 SQL_NULL_DATA로 설정합니다.
- 22 - PATCH2는 IBM ODBC 드라이버가 외부 조인을 지원되지 않는 것으로 보고하도록 만듭니다. 이는 col1 길이가 254 문자 보다 큰 외부 조인

- 명령문을 사용하는 SELECT DISTINCT col1 또는 ORDER BY col1을 생성하여 DB2 UDB가 오류를 리턴하는(DB2 UDB는 254 바이트 이상의 컬럼을 이 사용법에서 지원하지 않으므로) 응용프로그램에 해당됩니다.
- 23 - cbColDef=0과 바인드되는 매개변수에 대한 입력을 최적화하지 않습니다.
 - 24 - 시간 값을 문자로 맵핑하기 위한 일시적인 해결책에 액세스합니다.
 - 25 - 십진수 컬럼에 대한 일시적인 해결책에 액세스하여, 문자 표현에서 후미 0들을 제거합니다.
 - 26 - sqlcode 464를 응용프로그램에 리턴하지 않습니다. 결과 세트가 리턴됨을 나타냅니다.
 - 27 - 응용프로그램이 유효한 값을 지정할지라도, SQLTable이 TABLETYPE 키워드 값을 사용하도록 강제 수행합니다.
 - 28 - 실수 컬럼을 DOUBLE 컬럼으로 설명합니다.
 - 29 - 십진수 컬럼에 대한 ADO 일시적인 해결책 - $1 > x > -1$ 일 때, x값에 사용된 맨 앞의 영(zero)을 제거 (몇몇 MDAC 버전에만 필요합니다)
 - 30 - 저장 프로시저어 캐싱 최적화를 사용할 수 없게 합니다.
 - 31 - SQLStatistics 호출에서 별명에 대한 통계를 보고합니다.
 - 32 - sqlcode -727 이유 코드 4 처리를 대체합니다.
 - 33 - char로 바뀌었을 때 시간소인의 ISO 버전을 리턴합니다. (ODBC 버전의 반대로)
 - 34 - CHAR FOR BIT DATA 컬럼을 CHAR로 보고합니다.
 - 35 - SQL_DESC BASE TABLE_NAME이 요청되었을 때 유효하지 않은 TABLENAME을 보고합니다 - ADO 읽기전용 최적화
 - 36 - 예약됨
 - 37 - 예약됨

제8부 일반 정보

일반 정보	525		설치 세부사항	536
DB2 Universal Database 비즈니스 인텔리전스 Quick Tour.	525		추가 에이전트 기능 설정	539
DB2 Everywhere는 현재 DB2 Everyplace를 의미 함	526		트리거 프로그램(XTClient)으로 웨어하우스 단계 스케줄링	541
마우스 필요.	526		변환기.	542
DB2 Run-Time Client에서 바인드 시도시 "바인드 파일이 없습니다"라는 오류를 리턴	526		DB2 계열 외의 데이터베이스 액세스	546
검색 발견	526		OS/390용 DB2 유틸리티 수행	548
HP-UX 11에 대한 메모리 창.	526		복제	550
dlfm client_conf 실패에 대한 사용자 조치	528		에이전트 로그	551
복사 디먼이 dlfm stop으로 중지되지 않는 드문 이 벤트	528		Windows NT에서 클라이언트쪽 캐칭	552
DB2 DFS Client Enabler 설치 제거	528		Enterprise Edition UNIX CD-ROM에서의 시험 사용판.	552
Windows NT에서의 클라이언트 인증	528		DB2 Connect Enterprise Edition UNIX CD-ROM에서의 시험 사용판	553
fork 중에 자동 로드 프로그램이 정지할 수도 있음	529		Data Links Manager 삭제.	554
DATALINK 복원	529		SMIT May Remove Additional Filesets를 사용하 여 DLFM 구성요소 설치 제거	554
Windows NT용 IBM 통신 서버(CS/NT)에서 사용 자 ID 및 암호 정의	529		Windows 2000에서 CLP 사용시 SQL1035N 오류 발생	554
노드 정의	529		SQL 지원에 대해 개선된 기능	555
연합 시스템 제한사항	530		Linux에서 DB2용 Gnome 및 KDE 바탕 화면 통 합	555
DataJoiner 제한사항.	531		Windows 2000 Terminal Server의 관리 모드에서 DB2 실행	555
Windows NT용 히브리어 Information Catalog Manager.	531		백업 및 복원 명령에 대한 온라인 도움말	556
Microsoft SNA Server 및 SNA Multisite Update(2단계 확약) 지원	531		"Warehouse Manager"는 "DB2 Warehouse Manager"이어야 함	556
DB2의 SNA SPM이 Windows 부팅 후에 실패함	532			
DAS(DB2 Administrator Server)에 대한 로케일 설정	532			
단축키가 작동하지 않음.	533			
Windows NT 및 Windows 2000에서 DB2에 대 한 서비스 계정 요구사항	533			
버전 6에서 작성된 Query Patroller 사용자의 EXECUTE 특권 상실	533			
Query Patroller 제한사항	534			
DWC(Data Warehouse Center)에서 사용될 모든 사용자 정의 프로그램에 확약 필요	534			
Data Warehouse Center 명령행 내보내기를 위한 새 옵션	534			
백업 서비스 API(XBSA)	535			
OS/390 에이전트	535			
설치 개요	536			

일반 정보

DB2 Universal Database 비즈니스 인텔리전스 Quick Tour

Quick Tour는 Linux 또는 Linux/390용 DB2에서 사용할 수 없습니다.

Quick Tour는 작은 시스템 글꼴로 수행하도록 최적화되어 있습니다. OS/2에서 Quick Tour를 올바르게 보려면 웹 브라우저의 글꼴 크기를 조정해야 할 수도 있습니다. 글꼴 크기 조정에 대해서는 사용하는 웹 브라우저에 대한 도움말을 참조하십시오. Quick Tour를 올바르게 보려면(SBCS 전용) 8 포인트 Helv 글꼴을 사용하는 것이 좋습니다. 일본어나 한국어 사용자의 경우, 8 포인트 Mincho 글꼴을 사용하는 것이 좋습니다. 기본 글꼴을 설정할 때, 환경설정 창의 글꼴 페이지에서 "문서에서 지정한 글꼴을 무시하고 기본 글꼴 사용" 옵션을 선택하십시오.

어떤 경우에는 Quick Tour가 2차 브라우저 창 뒤에서 시작될 수도 있습니다. 이러한 문제점을 정정하려면, Quick Tour를 닫고 12 페이지의 『Netscape 시작 시 발생하는 오류 메시지』의 단계를 수행하십시오.

Quick Tour를 시작할 때, 다음과 같은 JavaScript 오류가 표시될 수도 있습니다.

파일:/C/Program Files/SQLLIB/doc/html/db2qt/index4e.htm, 65행:

창이 정의되어 있지 않습니다.

이 JavaScript 오류가 발생하면 Quick Tour가 시작된 후 Quick Tour 시작 페이지인 index4e.htm이 자동으로 닫히지 않습니다. index4e.htm이 표시되는 브라우저 창을 닫으면 Quick Tour 시작 페이지를 닫을 수 있습니다.

"새로운 기능" 절의 데이터 관리 주제에서는 "on-demand 로그 아카이브 지원"이 버전 7.1에서 지원된다고 설명합니다. 이것은 잘못된 내용입니다. 또한 다음 문장도 포함되어 있습니다.

로그 파일의 크기가 4 GB에서 32 GB로 증가되었습니다.

이 문장은 다음과 같이 바뀌어야 합니다.

사용 중인 전체 로그 공간이 4 GB에서 32 GB로 증가되었습니다.

DB2 Data Links Manager에 대해 설명하는 절에는 다음과 같은 문장이 있습니다.

또한, NetBackup을 사용한 백업 및 복원을 위한 Veritas XBSA 인터페이스 사용을 지원합니다.

이 문장은 다음과 같이 바뀌어야 합니다.

또한, 파일 아카이브 및 복원을 위한 XBSA 인터페이스도 지원됩니다. XBSA 인터페이스를 지원하는 저장영역 관리자에는 Legato NetWorker 및 Veritas NetBackup이 포함됩니다.

DB2 Everywhere는 현재 DB2 Everyplace를 의미함

DB2 Everywhere의 이름이 DB2 Everyplace로 변경되었습니다.

마우스 필요

Windows를 제외한 모든 플랫폼의 경우, 도구를 사용하려면 마우스가 필요합니다.

DB2 Run-Time Client에서 바인드 시도시 "바인드 파일이 없습니다"라는 오류를 리턴

DB2 Run-Time Client에는 완전한 바인드 파일 세트가 없기 때문에 GUI 도구의 바인딩은 DB2 Run-Time Client에서 수행할 수 없으며 DB2 Administration Client에서만 수행할 수 있습니다.

검색 발견

검색 발견은 브로드캐스트 미디어에서만 지원됩니다. 예를 들어, 검색 발견은 ATM 어댑터를 통해서 기능 수행하지 않습니다. 그러나, 이 제한사항은 알려진 발견에는 적용되지 않습니다.

HP-UX 11에 대한 메모리 창

메모리 창은 대형 HP 64 비트 머신에 있으며 32 비트 응용프로그램에 대한 공유 메모리의 1.75 GB보다 많은 장점을 활용하고자 하는 사용자를 위한 것입니다. DB2의 64 비트 버전을 수행 중인 경우 메모리 창이 필요하지 않습니다. 메모리 창은 프로세스 당 또는 프로세스 그룹 당 별도로 1 GB씩을 사용할 수 있게 합니다. 그러면, 인스턴스는 자체적으로 1 GB의 공유 메모리와 0.75 GB의 전역 메모리를 가질 수 있습니다. 사용자가 이러한 장점을 활용하려는 경우, 여러 인스턴스를 각각의 창에서 수행할 수 있습니다. 다음은 메모리 창의 사용을 위한 전제조건과 조건입니다.

- DB2 EE 환경
 - 패치: 확장 소프트웨어 12/98 및 PHKL_17795.
 - 인스턴스에 대해 \$DB2INSTANCE 변수를 설정해야 합니다.
 - /etc/services.window 파일에 메모리 창에서 수행하고자 하는 각 DB2 인스턴스에 대한 항목이 있어야 합니다. 예를 들어, 다음과 같습니다.

```
db2instance1 50
db2instance2 60
```

주: 이름과 ID 사이에는 단 하나의 공백만 둘 수 있습니다.

- 서버에서 수행하려고 하는, 둘 이상의 단일 명령문이 필요한 DB2 명령은 TCPIP 루프백 메소드를 사용하여 수행해야 합니다. 이것은 메모리 창이 첫번째 명령문의 처리를 완료하면 셸이 종료되기 때문입니다. DB2 서비스는 이를 수행하는 방법을 알고 있습니다.

- 메모리 창에서 수행 중인 인스턴스에 대해 수행하려는 DB2 명령은 앞에 db2win(sqllib/bin에 위치)이 붙어야 합니다. 예를 들어, 다음과 같습니다.

```
db2win db2start
db2win db2stop
```

- 메모리 창 외부에서 수행되는 DB2 명령(그러나 메모리 창이 수행 중인 경우)은 1042를 리턴합니다. 예를 들어, 다음과 같습니다.

```
db2win db2start <== OK
db2 connect to db <==SQL1042
db2stop <==SQL1042
db2win db2stop <== OK
```

- DB2 EEE 환경

- 패치: 확장 소프트웨어 12/98 및 PHKL_17795.
- 인스턴스에 대해 \$DB2INSTANCE 변수를 설정해야 합니다.
- DB2_ENABLE_MEM_WINDOWS 레지스트리 변수는 TRUE로 설정해야 합니다.
- /etc/services.window 파일에 메모리 창에서 수행하고자 하는 각 인스턴스의 각 논리 노드에 대한 항목이 있어야 합니다. 각 항목의 첫번째 필드는 포트 번호로 연결된 인스턴스 이름이어야 합니다. 예를 들어, 다음과 같습니다.

```
=== db2instance1에 대한 $HOME/sqllib/db2nodes.cfg ===
5 host1 0
7 host1 1
9 host2 0
=== db2instance2에 대한 $HOME/sqllib/db2nodes.cfg ===
1 host1 0
2 host2 0
3 host2 1
=== host1에서의 /etc/services.window ===
db2instance10 50
db2instance11 55
db2instance20 60
=== host2에서의 /etc/services.window ===
db2instance10 30
db2instance20 32
db2instance21 34
```

- DB2 명령의 앞에 EE 환경에서만 사용되는 db2win을 붙여서는 안됩니다.

dlfm client_conf 실패에 대한 사용자 조치

DLFM 클라이언트에서, dlfm client_conf가 실패했다면, DB2 카탈로그에 있는 "오래된" 항목 때문일 수도 있습니다. 이를 해결하려면 다음 명령을 발행하십시오.

```
db2 uncatalog db <dbname>
db2 uncatalog node <node alias>
db2 terminate
```

그런 다음, dlfm client_conf를 다시 시도하십시오.

복사 디먼이 dlfm stop으로 중지되지 않는 드문 이벤트

사용자가 dlfm stop을 발행하거나 비정상적인 종료에 있는 경우에 dlfm_copyd(복사 디먼)가 중지하지 않는 상황이 가끔 발생하기도 합니다. 이러한 상황이 발생하면, dlfm의 재시작을 시도하기 전에 dlfm shutdown을 발행하십시오.

DB2 DFS Client Enabler 설치 제거

DB2 DFS Client Enabler가 설치 제거되기 전에 루트는 DFS 파일을 사용하고 있지 않으며 사용자가 DFS 파일 공간에 셸을 열어 두지 않았는지 확인해야 합니다. 루트로서 다음의 명령을 발행하십시오.

```
stop.dfs dfs_cl
```

/...가 더 이상 마운트되지 않았는지 확인하십시오.

```
mount | grep -i dfs
```

이 작업이 수행되지 않았으며 DB2 DFS Client Enabler가 설치 제거된 경우, 머신을 재부트해야 합니다.

Windows NT에서의 클라이언트 인증

Windows NT 환경에서 기존의 클라이언트 인증 메커니즘을 구현하기 위해 새 DB2 레지스트리 변수 DB2DOMAINLIST가 도입되었습니다. 이 변수는 Windows NT용 DB2 서버에서 하나 이상의 Windows NT 도메인을 정의하기 위해 사용됩니다. 이 목록에 정의된 도메인에 속한 사용자의 연결 또는 접속 요청만이 수용됩니다.

이 레지스트리 변수는 버전 7(또는 그 이상)에서 DB2 서버와 클라이언트가 수행 중인 순수한 Windows NT 도메인 환경에서만 사용해야 합니다.

관리 안내서: 성능의 "DB2 레지스트리 및 환경 변수" 절에서 이 레지스트리 변수의 설정에 대한 자세한 내용을 참조하십시오.

fork 중에 자동 로드 프로그램이 정지할 수도 있음

AIX 4.3.3에서는 fork 중에 자동 로드 프로그램을 정지시킬 가능성이 있는 libc 문제점의 수정사항을 제공합니다. 자동 로드 프로그램은 다중 스레드 프로그램입니다. 스레드 중 하나가 다른 프로세스를 fork 해제합니다. 하위 프로세스의 fork는 상위 메모리의 이미지를 하위에서 작성되게 합니다.

동일한 프로세스 내의 힙(heap)에서 메모리를 할당하는 여러 스레드를 관리하기 위해 libc.a가 사용하는 잠금이 비 fork 스레드에 의해 보류되었을 수 있습니다. 비 fork 스레드가 하위 프로세스에는 없으므로 이 잠금은 하위 프로세스에서 해제되지 않고 때때로 상위 프로세스가 정지하게 됩니다.

DATALINK 복원

롤 포워드의 사용에 상관없이 데이터베이스 복원 이후에 취한 오프라인 백업의 복원에서는 빠른 조정 처리가 수반되지 않습니다. 이러한 경우, 파일 링크 제어 하에 DATALINK 컬럼이 있는 모든 테이블은 데이터 링크 조정 보류(DRP) 상태에 있게 됩니다.

Windows NT용 IBM 통신 서버(CS/NT)에서 사용자 ID 및 암호 정의

DB2 서버에 연결하기 위해 원격 DB2 클라이언트에 대한 통신 프로토콜로 APPC를 사용하며 SNA 제품으로 CS/NT를 사용하는 경우, CS/NT 구성 파일에서 다음의 키워드가 올바르게 설정되었는지 확인하십시오. 이 파일은 x:\ibmcs\private 디렉토리에서 주로 찾을 수 있습니다.

노드 정의

TG_SECURITY_BEHAVIOR

이 매개변수는 TP가 보안을 위해 구성되지 않은 경우에 노드가 ATTACH에 제시된 보안 정보를 처리하는 방법을 사용자가 판별할 수 있게 합니다.

IGNORE_IF_NOT_DEFINED

이 매개변수는 사용자가 ATTACH에 보안 매개변수가 제시되었는지의 여부를 판별하고 TP가 보안을 위해 구성되지 않은 경우에 이를 무시할 수 있게 합니다.

IGNORE_IF_NOT_DEFINED를 사용하는 경우, CS/NT에서 사용자 ID와 암호를 정의할 필요가 없습니다.

VERIFY_EVEN_IF_NOT_DEFINED

이 매개변수는 사용자가 ATTACH에 보안 매개변수가 제시되었는지를 판별하고 TP가 보안을 위해 구성되지 않은 경우라도 이를 확인할 수 있게 합니다. 이 값이 기본값입니다.

VERIFY_EVEN_IF_NOT_DEFINED를 사용하는 경우 CS/NT에서 사용자 ID와 암호를 정의해야 합니다.

CS/NT 사용자 ID와 암호를 정의하려면, 다음의 단계를 수행하십시오.

1. 시작 --> 프로그램 --> IBM 통신 서버 --> SNA 노드 구성. 통신 서버 구성 환영 창이 열립니다.
2. 수정하려는 구성 파일을 선택하십시오. 다음을 클릭하십시오. 구성 시나리오 선택 창이 열립니다.
3. CPI-C, APPC 또는 5250 에뮬레이션을 강조표시하십시오. 완료를 클릭하십시오. 통신 서버 SNA 노드 창이 열립니다.
4. CPI-C 및 APPC 옆의 [+]를 클릭하십시오.
5. LU6.2 보안 옆의 [+]를 클릭하십시오.
6. 사용자 암호를 마우스 오른쪽 버튼으로 클릭하고 작성을 선택하십시오. 사용자 ID 암호 정의 창이 열립니다.
7. 사용자 ID와 암호를 입력하십시오. 확인을 클릭하십시오. 완료를 클릭하여 변경사항을 적용하십시오.

연합 시스템 제한사항

다음은 연합 시스템에 적용되는 제한사항입니다.

- Oracle 데이터 유형 NCHAR, NVARCHAR2, NCLOB 및 BFILE은 별명을 포함하는 조회에서 지원되지 않습니다.
- Create Server Option, Alter Server Option 및 Drop Server Option 명령은 제어 센터에서 지원되지 않습니다. 이러한 명령을 발행하려면, 명령행 처리기(CLP)를 사용해야 합니다.
- 별명을 포함하는 조회의 경우, DB2 UDB는 항상 DFT_SQLMATHWARN 데이터 베이스 구성 옵션을 따르지는 않습니다. 그 대신, DB2 UDB는 DFT_SQLMATHWARN 설정에 관계없이 원격 데이터 소스에서 직접 산술 오류나 경고를 리턴합니다.
- CREATE SERVER OPTION문은 조합 순서에서 대소문자가 구별되는 데이터 소스에 대해 COLSEQ 서버 옵션이 'I'로 설정되지 않도록 합니다.
- ALTER NICKNAME문은 유효하지 않은 옵션이 지정될 때 SQL0901N을 리턴합니다.
- Oracle, Microsoft SQL Server 및 Sybase 데이터 소스의 경우, 숫자 데이터 유형은 DB2의 BIGINT 데이터 유형으로 맵될 수 없습니다. 기본적으로, Oracle의 숫자 (p,s) 데이터 유형(10 <= p <= 18, s = 0)은 DB2의 DECIMAL 데이터 유형에 맵핑됩니다.

DataJoiner 제한사항

연합 환경 내에서 발행된 DR(Distributed requests)은 읽기 전용 조작으로 제한됩니다.

Windows NT용 히브리어 Information Catalog Manager

Information Catalog Manager는 히브리어로 사용 가능하며 Windows NT용 DB2 Warehouse Manager CD에 제공됩니다.

히브리어 변환은 zip 파일인 IL_ICM.ZIP으로 제공되며 Windows NT용 DB2 Warehouse Manager CD의 DB2\IL 디렉토리에 있습니다.

Information Catalog Manager의 히브리어 판을 설치하려면, 먼저 Windows NT용 DB2 Warehouse Manager 영어 버전과 모든 전제조건을 Windows NT의 Hebrew Enabled 버전에 설치하십시오.

Windows NT용 DB2 Warehouse Manager가 설치되고 나면, DB2\IL 디렉토리에서 Windows NT용 DB2 Warehouse Manager가 설치된 디렉토리에 IL_ICM.ZIP 파일의 압축을 해제하십시오. zip 파일에 포함된 디렉토리 구조를 작성하려면 unzip 프로그램에 올바른 옵션을 제공하십시오.

파일 압축이 해제되면, 전역 환경 변수 LC_ALL을 En_US에서 Iw_IL로 변경하십시오. 설정을 변경하려면 다음을 수행하십시오.

1. Windows NT 제어판을 열고 시스템 아이콘을 두 번 클릭하십시오.
2. 시스템 등록정보 창에서, 환경 탭을 클릭한 뒤, 시스템 변수 섹션에서 LC_ALL 변수를 찾으십시오.
3. 변수를 클릭하여 값 편집 상자에 값을 표시하십시오. 값을 En_US에서 Iw_IL로 값을 변경하십시오.
4. 설정 버튼을 클릭하십시오.
5. 시스템 등록정보 창과 제어판을 닫으십시오.

이제 Information Catalog Manager의 히브리어 버전을 설치하십시오.

Microsoft SNA Server 및 SNA Multisite Update(2단계 확약) 지원

호스트와 AS/400 응용프로그램은 SNA 제품으로 Microsoft SNA Server를 사용 중일 때 SNA 2단계 확약을 사용하여 DB2 UDB 서버에 액세스할 수 없습니다. 이것이 지원된다는 DB2 UDB 서적의 내용은 잘못된 것입니다. Windows NT용 IBM 통신 서버 버전 5.02 이상이 필요합니다.

주: Windows용 DB2 UDB를 사용하여 호스트 및 AS/400 데이터베이스 서버에 액세스하는 응용프로그램은 Microsoft SNA Server 버전 4 서비스 팩 3 이상을 사용하는 SNA two commit를 사용할 수 있습니다.

DB2의 SNA SPM0 | Windows 부팅 후에 실패함

Microsoft SNA Server 버전 4 SP3 이상을 사용할 경우, DB2의 SNA SPM이 재부트 후에 적절하게 시작되었는지 확인하십시오. 다음과 유사한 항목에 대해 `\sqllib\\db2diag.log` 파일을 확인하십시오.

```
2000-04-20-13.18.19.958000 Instance:DB2 Node:000
PID:291(db2syscs.exe) TID:316 Appid:none
common communication sqlccspmconnmgr APPC init Probe:19
SPM0453C Microsoft SNA Server가 시작하지 않았으므로 동기 지점 관리 프로그램이 시작하지 않았습니다.
```

```
2000-04-20-13.18.23.033000 Instance:DB2 Node:000
PID:291(db2syscs.exe) TID:302 Appid:none
common communication sqlccsna start listen Probe:14
DIA300TE "SNA SPM" 프로토콜이 정상적으로 시작되지 않았습니다.
```

```
2000-04-20-13.18.23.603000 Instance:DB2 Node:000
PID:291(db2syscs.exe) TID:316 Appid:none
common communication sqlccspmconnmgr listener Probe:6
DIA3103E APPC 프로토콜 지원에 오류가 발생했습니다. APPC 명령어 "APPC(DISPLAY 1
BYTE)". 1차 rc는 "F004"이며 2차 rc는 "00000000"입니다.
```

db2diag.log에 이러한 항목이 있고, 시간소인이 최근 재부트 시간과 일치할 경우, 다음을 수행해야 합니다.

1. db2stop을 실행하십시오.
2. SnaServer 서비스를 시작하십시오(아직 시작되지 않은 경우).
3. db2start를 실행하십시오.

db2diag.log 파일을 다시 확인하여 항목이 더 이상 추가되지 않는지 확인하십시오.

DAS(DB2 Administrator Server)에 대한 로케일 설정

DAS(DB2 Administrator Server) 인스턴스의 로케일이 DB2 인스턴스 로케일과 호환되는지 확인하십시오. 그렇지 않으면, DB2 인스턴스는 DAS와 통신할 수 없습니다.

LANG 환경 변수가 DAS의 사용자 프로파일에 설정되지 않은 경우, DAS는 기본 시스템 로케일로 시작됩니다. 기본 시스템 로케일이 정의되어 있지 않으면, 코드 페이지 819로 DAS가 시작됩니다. DB2 인스턴스가 DBCS 로케일 중 하나를 사용하고, DAS가 코드 페이지 819로 시작될 경우, 인스턴스는 DB2 관리 서버와 통신할 수 없게 됩니다. DAS의 로케일과 DB2 인스턴스의 로케일은 호환 가능해야 합니다. 예를 들어, 중국어 Linux 시스템에서 DAS의 사용자 프로파일에 "LANG=zh_CN"가 설정되어야 합니다.

단축키가 작동하지 않음

일부 언어에서는 Unix 기반 시스템과 OS/2의 제어 센터에 대해 일부 키보드 단축키가 작동되지 않습니다. 마우스를 사용하여 옵션을 선택하십시오.

Windows NT 및 Windows 2000에서 DB2에 대한 서비스 계정 요구사항

Windows NT 또는 Windows 2000용 DB2를 설치하는 동안, 설치 프로그램은 여러 Windows 서비스를 작성하고, 각 서비스에 서비스 계정을 할당합니다. DB2가 제대로 실행되도록, 설치 프로그램이 다음 사용자 권한을 DB2 서비스와 연관된 서비스 계정에 부여합니다.

- 운영 체제의 일부로서 동작
- 토큰 오브젝트 작성
- 할당량 증가
- 서비스로 로그인
- 프로세스 레벨 토큰 바꾸기

DB2 서비스에 대해 다른 서비스 계정을 사용하려는 경우, 이들 사용자 권한을 서비스 계정에 부여해야 합니다.

이들 사용자 권한 뿐만 아니라, 서비스 계정도 DB2 제품이 설치된 디렉토리로의 쓰기 액세스 권한을 가져야 합니다.

DB2 관리 서버 서비스(DB2DAS00 서비스)에 대한 서비스 계정도 다른 DB2 서비스를 시작 및 중지할 권한을 가지고 있어야 하며(즉, 서비스 계정은 POWER Users 그룹에 속해야 합니다), 이것이 관리하는 DB2 인스턴스에 대해 DB2 SYSADM 권한을 갖고 있어야 합니다.

버전 6에서 작성된 Query Patroller 사용자의 EXECUTE 특권 상실

Query Patroller 버전 7에 추가된 새로운 저장 프로시저(IWM.DQPGROUP, IWM.DQPVALUR, IWM.DQPCALCT 및 IWM.DQPINJOB)로 인해, Query Patroller 버전 6에서 작성한 기존 사용자들은 이들 패키지에 대해 EXECUTE 권한을 갖지 않습니다. 이 문제점을 자동으로 수정하는 응용프로그램이 FixPak 1에 추가되었습니다.

DQP Query Admin을 사용하여 DQP 사용자 정보를 수정하려 할 때는 사용자 목록에서 기존 사용자를 제거하지 마십시오.

Query Patroller 제한사항

JVM (Java Virtual Machine) 플랫폼 제한사항으로 인해, Query Enabler는 HP-UX 및 NUMA-Q에서 지원되지 않습니다. 뿐만 아니라, Query Patroller 추적 프로그램은 NUMA-Q에서 지원되지 않습니다. 모든 Query Patroller 클라이언트 도구가 필수인 경우, 다른 플랫폼(Windows NT와 같은)을 사용하여 HP-UX 또는 NUMA-Q 서버에서 이들 도구를 실행하십시오.

DWC(Data Warehouse Center)에서 사용될 모든 사용자 정의 프로그램에 요약 필요

DWC(Data Warehouse Center)에서 사용자 정의 프로그램으로서 DB2 저장 프로시저어가 작성한 저장 프로시저어를 사용하려는 경우, 저장 프로시저어에서 `con.close();` 명령문 앞에 다음 명령문을 삽입해야 합니다.

```
con.commit();
```

이 명령문이 삽입되지 않으면, DWC에서 저장 프로시저어를 실행할 때 저장 프로시저어가 수행한 변경사항이 구간 복원됩니다.

DWC에 있는 모든 사용자 정의 프로그램에 대해서는, 변경사항이 데이터베이스에 적용되도록 포함된 모든 DB2 함수들을 명시적으로 요약해야 합니다. 즉, COMMIT 명령문을 사용자 정의 프로그램에 추가해야 합니다.

Data Warehouse Center 명령행 내보내기를 위한 새 옵션

태그 파일로의 명령행 내보내기에는 새 옵션 /B가 있습니다. 이 옵션은 Data Warehouse Center 인터페이스에서는 사용할 수 없습니다.

`iwh2exp2` 명령의 새 구문은 다음과 같습니다.

```
iwh2exp2 filename.INP dbname userid password [PREFIX=table_schema] [/S] [/R] [/B]
```

여기서

- filename.INP는 INP 파일의 전체 경로 이름입니다.
- dbname은 Data Warehouse Center 제어 데이터베이스 이름입니다.
- userid는 데이터베이스로 로그인하는 데 사용되는 사용자 ID입니다.
- password는 데이터베이스로 로그인하는 데 사용되는 암호입니다.
- 선택적 매개변수는 다음과 같습니다.
 - PREFIX=table schema: 제어 데이터베이스 테이블에 대한 테이블 스키마 (기본값은 `IWH`입니다).
 - /S: 선택된 단계의 스케줄을 내보내기합니다.
 - /R: 선택된 단계의 웨어하우스 소스를 내보내기하지 않습니다.
 - /B: 선택된 단계의 기여 단계를 내보내기하지 않습니다.

주: /R 또는 /B가 지정된 경우, 결과 태그 파일을 가져오기할 때 이미 웨어하우스 소스 또는 기여 단계가 이미 존재해야 합니다. 그렇지 않으면 오류가 리턴됩니다.

백업 서비스 API(XBSA)

백업 서비스 API(XBSA)는 백업 또는 아카이브 목적을 위해 데이터 저장영역 관리가 필요한 응용프로그램 또는 기능간의 개방형 응용프로그램 프로그래밍 인터페이스로서 영국에서 Open Group에 의해 정의된 것입니다. 이에 대해서는 "Open Group Technical Standard System Management: Backup Services API(XBSA)", 문서번호 C425(ISBN: 1-85912-056-3)에서 설명합니다.

이를 지원하기 위해, 두 개의 새 DB2 레지스트리 변수가 작성되고 현재는 AIX, HP, Solaris 및 Windows NT에서 지원됩니다.

DB2_VENDOR_INI

모든 벤더 고유의 환경 설정이 들어 있는 파일을 지시합니다. 데이터베이스 관리 프로그램이 시작될 때 값을 가져 옵니다.

DB2_XBSA_LIBRARY

벤더 지원 XBSA 라이브러리를 지시합니다. AIX에서, shr.o로 지정되지 않았으면 설정에서 공유 오브젝트를 포함해야 합니다. HP, Solaris 및 Windows NT는 공유 오브젝트 이름을 요구하지 않습니다. 예를 들어 Legato의 DB2용 NetWorker Business Suite 모듈을 사용하려면, 레지스트리 변수가 다음과 같이 설정되어야 합니다.

```
db2set DB2_XBSA_LIBRARY="/usr/lib/libxdb2.a(bsashr10.o)"
```

XBSA 인터페이스는 BACKUP DATABASE 또는 RESTORE DATABASE 명령을 통해 호출될 수 있습니다. 예를 들어, 다음과 같습니다.

```
db2 backup db sample use XBSA
db2 restore db sample use XBSA
```

OS/390 에이전트

이 문서는 OS/390 및 기능 설치에 관한 내용입니다.

이 문서는 OS/390 에이전트 설치 방법에 대한 설명과 기능에 관한 정보를 제공합니다. 설치 프로세스에 대한 빠른 개요는 "설치 개요"를, 자세한 설명은 "설치 세부사항"을 참조하십시오. 에이전트 기능 관련 정보는 "추가 에이전트 기능 설정", "변환기", "DB2 계열 이외의 데이터베이스 액세스"를 참조하십시오.

개요

DB2 웨어하우스 센터에는 OS/390 에이전트가 포함됩니다. 에이전트를 사용하여 OS/390용 DB2 Universal Database와 다른 플랫폼의 DB2 데이터베이스 및 비 DB2 데이터베이스를 포함한 다른 데이터베이스 사이에서 통신할 수 있습니다. 에이전트는 ODBC 연결을 사용하며 지원되는 데이터 소스와 통신할 수 있습니다. 에이전트는

OS/390 UNIX 시스템 서비스 하에서 실행됩니다. OS/390 V2R6 이상이 필요하며 OS/390용 DB2 Universal Database 버전 5, 6, 7과 호환됩니다.

OS/390 에이전트는 다음 작업을 지원합니다.

- 소스 DB2 데이터베이스에서 목표 DB2 데이터베이스로 데이터 복사
- 테이블 또는 파일의 내용 샘플링
- 사용자 정의 프로그램 수행
- IBM DataJoiner를 통한 DB2 이외의 데이터베이스 액세스
- Cross Access의 Classic Connect를 통한 VSAM 또는 IMS 데이터 액세스
- OS/390용 DB2 Universal Database 유틸리티 수행
- IBM Data Propagator에 적용 작업 수행

설치 개요

다음 단계들은 설치 프로세스를 요약한 것입니다. "설치 세부사항" 절에 이 단계에 관해 자세한 설명이 있습니다.

1. OS/390용 DB2 Universal Database 테이프에서 OS/390 에이전트를 설치하십시오.
2. 프로파일 파일의 환경 변수를 갱신하십시오.
3. 다음 항목 사이에 연결을 설정하십시오.
 - 커널과 에이전트 디먼 사이.
 - 에이전트와 액세스할 데이터베이스 사이.
4. CLI를 지역 및 원격 데이터베이스로 바인드하십시오.
5. ODBC 초기화 파일을 설정하십시오.
6. 사용자가 다음 작업을 할 수 있도록 권한을 설정하십시오.
 - 에이전트 디먼 실행
 - DSNAOCLI 플랜에 대한 실행 권한 확보
 - 필요하면 로그 및 ODBC 추적 디렉토리에 읽기 및 쓰기 권한 부여
7. 에이전트 디먼을 시작하십시오.

설치 세부사항

OS/390 에이전트 설치

OS/390 에이전트는 OS/390용 DB2 Universal Database 버전 7 테이프에 포함되어 있습니다. OS/390 에이전트 설치에 관해 자세한 정보는 테이프에 포함된 프로그램 디렉토리를 보십시오.

OS/390 에이전트를 설치하기 전에 DB2 서브시스템에 apar PQ36585나 PQ36586을 적용해야 합니다.

프로파일 파일의 환경 변수 갱신

이 변수들은 에이전트가 다양한 DB2 라이브러리 및 출력 디렉토리 등을 가리키도록 합니다.

다음은 샘플 .profile 파일의 내용입니다. .profile 파일은 환경 변수를 정의하며, 에이전트 디먼을 시작한 사용자의 홈 디렉토리에 있어야 합니다.

```
export VWS_LOGGING=/usr/lpp/DWC/logs
export VWP_LOG=/usr/lpp/DWC/vwp.log
export VWS_TEMPLATES=/usr/lpp/DWC/
export DSNAOINI=/usr/lpp/DWC/dsnaoini
export LIBPATH=/usr/lpp/DWC/:$LIBPATH
export PATH=/usr/lpp/DWC/:$PATH
export STEPLIB=DSN710.SDSNEXIT:DSN710.SDSNLOAD
```

연결 설정

커널 및 디먼 연결을 설정하려면 /etc/services 파일이나 TCPIP.ETC.SERVICES 파일에 다음 줄을 추가하십시오.

```
wkernel 11000/tcp
vwd 11001/tcp
vwlogger 11002/tcp
```

OS/390 에이전트와 데이터베이스 사이에서 연결을 설정하려면, 원격 데이터베이스를 OS/390 통신 데이터베이스(CDB)에 추가하십시오. 다음이 샘플 CDB 삽입 내용입니다.

```
INSERT INTO SYSIBM.LOCATIONS
(LOCATION, LINKNAME, PORT)
VALUES
('NTDB', 'VWNT704', '60002');
INSERT INTO SYSIBM.IPNAMES
(LINKNAME, SECURITY_OUT, USERNAMES, IPADDR)
VALUES
('VWNT704', 'P', 'O', 'VWNT704.STL.IBM.COM');
INSERT INTO SYSIBM.USERNAMES
(TYPE, AUTHID, LINKNAME, NEWAUTHID, PASSWORD)
VALUES
('O', 'MVSUID', 'VWNT704', 'NTUID', 'NTPW');
```

연결 설정 및 통신 데이터베이스 갱신에 관한 자세한 정보는 *DB2 UDB for OS/390 Installation Guide*, GC26-9008-00의 "Connecting Distributed Database Systems" 을 참조하십시오.

CLI 바인딩

OS/390 에이전트는 CLI를 사용하여 DB2와 통신하므로, 사용자의 CLI 플랜을 에이전트가 액세스하는 모든 원격 데이터베이스와 바인드해야 합니다. 다음은 OS/390용 지역 DB2 데이터베이스의 샘플 bind package문입니다.

```

BIND PACKAGE (DWC6CLI) MEMBER(DSNCLICS) ISO(CS)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLINC) ISO(NC)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIRR) ISO(RR)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIRS) ISO(RS)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIUR) ISO(UR)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIMS)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIC1)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIC2)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIF4)

```

다음은 Windows NT에서 실행되는 DB2 데이터베이스를 위한 샘플 bind package문입니다.

```

BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLICS) ISO(CS)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLINC) ISO(NC)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIRR) ISO(RR)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIRS) ISO(RS)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIUR) ISO(UR)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIC1)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIC2)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIQR)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIF4)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIV1)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIV2)

```

다음은 플랜에서 CLI 패키지들을 함께 바인드하는 샘플 bind문입니다.

```
BIND PLAN(DWC6CLI) PKLIST(*.DWC6CLI.* )
```

ODBC 초기설정 파일 설정

샘플 ODBC 초기설정 파일인 inisamp는 usr/lpp/DWC/ 디렉토리에 포함되어 있습니다. 사용자의 시스템에서 작업하도록 이 파일을 편집하거나, 자신의 파일을 작성할 수 있습니다. 파일이 제대로 작업하도록 하기 위해, 다음이 올바르게 구성되었는지 확인하십시오.

- DSNAOINI 환경 변수가 초기화 파일을 가리켜야 합니다.
- 파일 이름에 dsnaoini.location_name 이름 지정 규칙을 사용해야 합니다.
- 파일에 CONNECTTYPE=2 및 MVSATTACHTYPE=CAF 매개변수가 포함되어야 합니다.

CLI와 DSNAOINI 파일 바인드에 관한 자세한 정보는 *DB2 UDB for OS/390 ODBC Guide and Reference*의 SC26-9005를 참조하십시오.

권한 설정

OS/390 에이전트는 디먼 프로세스입니다. 정규 UNIX 보안 또는 OS/390 UNIX 보안으로 에이전트 디먼을 실행할 수 있습니다.

에이전트는 디먼 권한을 요구하므로 이러한 에이전트 실행 파일은 RACF 프로그램 제어에 정의하십시오.

- libt1s4d.dll
- iwhcomnt.dll
- vwd

실행 프로그램을 RACF 프로그램 제어에 정의하려면 Data Warehouse Center 실행 파일이 저장된 디렉토리로 변경하고 다음 명령을 수행하십시오.

```
extattr +p libt1s4d.dll
extattr +p iwhcomnt.dll
extattr +p vwd
```

extattr 명령에 +p 매개변수를 사용하려면, BPX.FILEATTR.PROGCTL FACILITY 클래스에 최소 읽기 액세스 권한이 있어야 합니다. 다음 예는 사용자 ID SMORG에 이 권한을 부여하는 데 사용되는 RACF 명령입니다.

```
RDEFINE FACILITY BPX.FILEATTR.PROGCTL UACC(NONE)
PERMIT BPX.FILEATTR.PROGCTL CLASS(FACILITY) ID(SMORG) ACCESS(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

권한 부여에 관한 자세한 정보는 *OS/390 UNIX System Services Planning*, SC28-1890 을 참조하십시오.

에이전트 디먼 시작

시스템 구성을 마치면 에이전트 디먼을 시작하십시오.

1. OS/390 호스트 이름 및 USS 포트를 통해 OS/390에서 UNIX 시스템 서비스로 텔넷하십시오.
2. 에이전트 디먼을 시작하십시오.
 - 디먼을 포그라운드(foreground)에서 시작하려면 명령행에 vwd를 입력하십시오.
 - 디먼을 백그라운드에서 시작하려면 다음을 입력하십시오.

```
vwd>/usr/lpp/DWC/logs/vwd.log 2>&1 &
```

OS/390 에이전트가 실행되는지 확인하려면 UNIX 셸 명령행에 다음 명령을 입력하십시오.

```
ps -e | grep vwd
```

또는, OS/390 콘솔에서 D 0MVS,a=all을 입력하여 문자열 vwd를 검색하십시오.

추가 에이전트 기능 설정

DB2 Warehouse Manager 패키지는 다음 사용자 정의 프로그램을 포함합니다.

- vwppftp: FTP 명령 파일 수행.
- vwpmvs: JCL 작업스트림 제출.
- vwprepy: FTP를 사용하여 파일 복사.

- XTClient: 클라이언트 트리거 프로그램.
- etidmvs: ETI(Evolutionary Technologies International)의 유틸리티이며, MVS에서 파일 삭제
- etircmvs: ETI의 유틸리티이며, MVS 호스트에서 FTP 실행
- etiexmvs: ETI의 유틸리티이며, MVS에서 JCL 실행

뿐만 아니라, Data Warehouse Center에서 사용자 정의 프로그램 및 저장 프로시저어를 작성할 수 있습니다. OS/390 에이전트는 UNIX 시스템 서비스에서 실행될 수 있는 모든 실행 프로그램을 지원합니다.

사용자 정의 프로그램은 하나 이상의 단계로 지정됩니다. 사용자 정의된 프로그램을 실행할 때, 다음 조치가 발생합니다.

- 에이전트가 시작됩니다.
- 에이전트가 사용자 정의 프로그램을 수행합니다.
- 사용자 정의 프로그램이 리턴 코드와 피드백 파일을 에이전트에게 리턴합니다.
- 에이전트가 커널에 결과를 리턴합니다.

OS/390에서 ETI 프로그램을 실행하려면 먼저 DB2 Universal Database 버전 7.1에 FixPak 2를 적용해야 합니다.

VWP_LOG 환경 변수를 사용하여 사용자 정의 프로그램이 출력을 작성할 수 있는 디렉토리를 정의하십시오.

FTP를 사용하는 작업을 제출하기 위해 사용자 정의 프로그램을 사용할 경우, 먼저 제출할 JCL과 데이터를 작성해야 합니다. JCL의 작업 이름은 USERIDx이어야 하고 여기서 x는 한 문자 또는 숫자입니다(예제: MYUSERA). JCL이 들어 있는 MSGCLASS 및 SYSOUT 파일의 출력 클래스는 JES 보유 출력 클래스를 지정해야 합니다.

제한사항: 제출된 작업의 최대 LRECL은 254 문자입니다. JES는 JCL의 처음 72 문자만 스캔합니다.

FTP 지원을 위한 Data Warehouse Center 템플릿 변경

Data Warehouse Center는 FTP를 사용한 파일 전송을 위한 JCL 템플릿을 설치합니다. OS/390 에이전트가 FTP 명령 GET 또는 PUT을 사용하여 파일을 OS/390 호스트에서 다른 원격 호스트로 전송하도록 하려면, JCL 템플릿의 계정 정보를 사용자의 OS/390 시스템에 맞도록 수정해야 합니다.

1. /usr/lpp/DWC 디렉토리의 파일을 복사 및 갱신할 수 있는 권한을 소유한 ID로 로그인하십시오.
2. ftp.jcl을 찾아 새 파일 이름 systemname.ftp.jcl로 복사하십시오. 여기서 systemname은 OS/390 시스템의 이름입니다.

3. 변환 프로그램 `vwpmvs` 또는 `ETI extract`를 수행하려는 각 OS/390 시스템에 대해 이 파일의 사본을 작성하십시오. 예를 들어, 이 프로그램들 중 하나를 `STLMVS1`에서 수행하려면, `STLMVS1.ftp.jcl`이라는 파일 사본을 작성하십시오.
4. 텍스트 편집기를 사용하여 사이트의 요구 사항에 맞도록 `JCL`을 사용자 정의하십시오. `MVS` 시스템에 대한 표준 계정 정보와 일치하도록 계정 정보를 변경하십시오. `[USERID]` 및 `[FTPFILE]`과 같이, 대괄호 안에 들어 있는 매개변수는 수정하지 마십시오(대괄호는 각각 16진 문자 `x'AD'`와 `x'BD'`입니다). `SPF` 옵션 0에서 `TSO` 터미널 유형을 `3278A`로 설정하지 않은 경우에는 이 값이 대괄호가 아닌 특수 문자로 표시됩니다. `x'AD'`나 `x'BD'` 또는 문자 사이의 데이터를 수정하지 않는다면, 이는 문제가 되지 않습니다.
5. 환경 변수 `VWS_TEMPLATES`가 복사된 템플릿 파일의 디렉토리를 가리키도록 갱신하십시오.

`Data Warehouse Center`에는 다음 샘플 `JCL` 템플릿이 포함됩니다.

```
//[USERID]A JOB , 'PUT/GET',
// CLASS=A,
// USER=&SYSUID,
// NOTIFY=&SYSUID,
// TIME=(,30),
// MSGCLASS=H
//STEP1 EXEC PGM=FTP,PARM='( EXIT'
//INPUT DD DSN=[FTPFILE],DISP=SHR
//OUTPUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
```

테이블 또는 파일의 내용 샘플링

`OS/390` 에이전트를 사용하면, `UNIX` 시스템 서비스 파일 및 `OS/390` 원시 플랫폼 파일과 같은 플랫폼 파일과 `DB2` 테이블의 내용을 샘플링할 수 있습니다. 또한 `OS/390` 에이전트를 사용하는 `Classic Connect`를 통해 `IMS` 또는 `VSAM` 파일의 내용을 샘플링할 수 있습니다. 플랫폼 파일의 경우, 에이전트는 파일 정의의 등록정보에 있는 매개변수를 보고 파일 형식을 판단합니다.

트리거 프로그램(XTClient)으로 웨어하우스 단계 스케줄링

트리거 프로그램을 사용하여 `OS/390` 플랫폼에서 웨어하우스 단계를 계획합니다. 직접 `Data Warehouse Center`에서 단계를 트리거하는 작업을 제출할 수도 있고 `OS/390` 작업 스케줄러가 이를 수행할 수도 있습니다. 단계가 성공하면 `JCL`의 트리거 단계는 0 리턴 코드를 리턴합니다.

트리거 프로그램을 사용하려면 `Java Development Kit (JDK) 1.1.8` 또는 그 이상이 `OS/390 UNIX Systems Services`에 설치되어 있어야 합니다. 트리거를 시작하려면, 먼저 웨어하우스 서버가 수행 중인 머신에서 `XTServer` 서버를 시작하십시오. 이 프로세스는 *Data Warehouse Center 관리 안내서*, "Data Warehouse Center 외부에서 단

계 시작" 주제에 설명되어 있습니다. XTServer가 시작되면 OS/390에서 XTClient를 시작하십시오. 다음 예는 트리거를 시작하기 위한 샘플 JCL입니다.

```
//DBA1A JOB 1,'XTCLIENT',CLASS=A,MSGCLASS=H,  
//      MSGLEVEL=(1,1),REGION=4M,NOTIFY=&SYSUID  
//*****  
//* submit iwhetrig  
//*****  
//BRADS EXEC PGM=BPXBATCH,  
//      PARM=('sh cd /usr/lpp/DWC/; java XTClient 9.317.171.133 1100x  
//      9 drummond pw bvmvs2nt 1 1 100')  
//STDOUT DD PATH='/tmp/xtclient.stdout',  
//      PATHOPTS=(OWRONLY,OCREAT),  
//      PATHMODE=SIRWXU  
//STDERR DD PATH='/tmp/xtclient.stderr',  
//      PATHOPTS=(OWRONLY,OCREAT),  
//      PATHMODE=SIRWXU  
//
```

주: 위 샘플 JCL 코드는 매개변수를 새 행에서 계속하는 방법을 보여줍니다. 그러려면, 매개변수를 71 컬럼까지 입력하고, 72 컬럼에 'X'를 입력한 뒤 다음 행의 16 컬럼에서 계속하십시오.

매개변수의 첫번째 부분은 OS/390 에이전트가 설치된 디렉토리로 변경하는 명령문(cd /usr/lpp/DWC/;)입니다. 매개변수의 두번째 부분은 XTClient를 시작하고 다음 8개의 매개변수를 전달합니다.

- Data Warehouse Center 서버 호스트 이름 또는 IP 주소
- Data Warehouse Center 서버 포트(보통 11009)
- Data Warehouse Center 사용자 ID
- Data Warehouse Center 암호
- 수행할 단계의 이름
- Data Warehouse Center 서버 명령, 여기서,
 - 1 = 단계 시작
 - 2 = 단계를 테스트 모드로 승격
 - 3 = 단계를 생산 모드로 승격
 - 4 = 단계를 테스트 모드로 강등
 - 5 = 단계를 개발 모드로 강등
- 단계 완료를 기다릴지 여부를 지정하는 옵션(여기서 1은 yes, 0은 no)
- 최대 행 개수(모든 행을 패치하려면 0이나 공백 사용)

변환기

소개

다음 12개의 변환기는 기본적인 데이터 변환을 제공하는 Java 저장 프로시저어입니다. 변환기를 실행하려면 먼저 서브시스템에 Java 저장 프로시저어를 설정해야 합니다. 변환기에 관한 자세한 정보는 *IBM DB2 Universal Database Data Warehouse Center 관리 안내서 버전 7, SA30-1000-00*을 참조하십시오.

IWH.CLEAN
IWH.PERIODTABLE
IWH.KEYTABLE
IWH.CHISQUARE
IWH.CORRELATION
IWH.STATISTICS
IWH.INVERTDATA
IWH.PIVOTDATA
IWH.REGRESSION
IWH.ANOVA
IWH.SUBTOTAL
IWH.MOVINGAVERAGE

Java 저장 프로시저어 설정

다음은 Java 저장 프로시저어 설정 방법의 간략한 설명입니다. 더 자세한 설명은 *Application Programming Guide and Reference for Java™*, SC26-9018을 참조하십시오.

1. PTFs UQ46170 및 UQ46114를 DB2 서브시스템에 적용하십시오.
2. OS/390 시스템에 Visual Age for Java 2.0을 설치하십시오.
3. DB2에 JDBC를 설치하고 DB2 서브시스템에서 JDBC 패키지를 바인드하십시오.
4. DB2 서브시스템에 RRS 및 DB2 WLM 저장 프로시저어를 설정하십시오.
5. DB2용 Java 저장 프로시저어를 설정하십시오. 이 단계에서는 Java 프로시저어 주소 공간에 대한 WLM 시작 프로시저어를 작성해야 합니다.
6. WLM에서 Java WLM 시작 프로시저어를 WLM 환경 이름과 연결하십시오. "응용프로그램 환경 작성"이라는 WLM 응용프로그램 환경 패 널을 사용하여 환경 이름을 JCL 프로시저어와 연결하십시오.
7. CREATE나 ALTER PROCEDURE의 WLM_ENVIRONMENT 옵션에 대한 WLM 응용프로그램 환경 이름을 지정하여 저장 프로시저어나 사용자 정의 함수를 응용프로그램 환경과 연결하십시오.
8. DB2에서 시작된 task의 소유자가 Java WML 시작 프로시저어의 라이브러리에 대한 액세스 권한이 있는지 확인하십시오.

웨어하우스 변환기 설정 단계

다음은 웨어하우스 변환기 설정을 간략히 설명합니다. 더 자세한 설명은 *IBM DB2 Universal Database Data Warehouse Center 관리 안내서 버전 7 SA30-1000-00*을 참조하십시오.

1. FixPak 3을 NT 버전 7용 DB2 Universal Database에 적용하거나 웨어하우스 제어 데이터베이스를 TRANSREGISTERED = 1 및 TRANSFENCED=1로 갱신하십시오. 웨어하우스 제어 데이터베이스를 갱신하려면 DB2 Universal Database 명령행 처리기에 다음 SQL을 입력하십시오.

```
CONNECT TO your_vw_control_database
UPDATE IWH.INFORESOURCE SET TRANSREGISTERED = '1' WHERE SUBDBTYPE = 'DB2 MVS'
UPDATE IWH.INFORESOURCE SET TRANSFENCED = '1' WHERE SUBDBTYPE = 'DB2 MVS'
```

2. 변환기를 DB2에 정의하십시오.
 - OS/390용 DB2 버전 7을 사용하는 경우, /usr/lpp/DWC/createXfSQLV7에서 SQL문을 사용하십시오.
 - OS/390용 DB2 버전 6을 사용하는 경우, /usr/lpp/DWC/createXfSQL에서 SQL문을 사용하십시오.
 - OS/390용 DB2 버전 5를 사용하는 경우, /usr/lpp/DWC/createXfSQL에서 주석 처리된 SQL문을 사용하십시오. 모든 CREATE PROCEDURE문을 주석 처리하고, INSERT INTO SYSIBM.SYSPROCEDURES문의 주석을 삭제하여 OS/390용 DB2 버전 5의 변환기를 정의하십시오.

Java 저장 프로시저를 설정했으면 WLM을 사용하여 Java WLM 시작 프로시저를 WLM 환경 이름과 연결하십시오. 환경 이름은 CREATE PROCEDURE문의 WLM ENVIRONMENT 옵션에 지정되어 있습니다. DSNWLMJ는 위에 설명한 변환기 정의와 함께 포함된 WLM 환경 이름입니다. DSNWLMJ의 WLM 연결 이름을 추가하거나 각 변환기 정의의 WLM ENVIRONMENT 옵션을 시작 프로시저와 연결해 둔 이름으로 변경할 수 있습니다.

3. UNIX 시스템 서비스에서 IWH710.SIWHLOAD의 변환기 로드 모듈로 연결되는 링크를 설정하십시오.
 - OS/390 호스트 시스템에서 UNIC 시스템 서비스로 텔넷하십시오.
 - OS/390 에이전트를 설치한 디렉토리로 변경하십시오. 기본 설치 디렉토리는 /usr/lpp/DWC입니다.
 - DB2 V7을 사용할 경우는 단계 4로 건너뛰고, DB2 V5나 V6을 사용할 경우는 설치된 디렉토리의 trlinks 데이터 세트를 편집하십시오. 컬럼 1에 파운드 기호(#)를 넣어 이 줄을 주석 처리하십시오.

```
ln -e IWHXFV7 xf.j11;
```

컬럼 1에서 파운드 기호(#)를 제거하여 이 줄의 주석 처리를 취소하십시오.

```
#ln -e IWHXF xf.j11;
```

변경사항을 저장하십시오.

- trlinks를 입력하고 Enter를 누르십시오. 그러면 에이전트가 IWHXF나 IWHXFV7 중 하나를 로드하도록 지정하는 xf.j11 링크가 해당 디렉토리에 작성됩니다.

4. APF-authorize IWH710.SIWHPDSE, 다음 DB2 Java 저장 프로시저에 있는 STEPLIB 병합에 추가하십시오.
5. xf.j11 링크가 있는 디렉토리(기본값: /usr/lpp/DWC)를 WML 환경 데이터 세트의 CLASSPATH 및 LIBPATH 환경 변수에 추가하십시오.
 - WLM 환경 데이터 세트의 위치를 모를 경우 DB2 Java 저장 프로시저의 시작 프로시저를 검토하십시오. WLM 환경 데이터 세트는 JAVAENV DD 카드가 가리키는 데이터 세트입니다.
6. 저장 프로시저를 시작한 다음 웨어하우스 단계를 작성하고 수행하십시오.

Java 저장 프로시저의 제한사항

저장 프로시저의 서명으로 된 Java 오브젝트는 OS/390용 DB2 버전 7에서만 지원됩니다. 이 때문에 변환기가 OS/390용 DB2 버전 5나 6에서 매개변수에 널(NULL) 값을 지원하지 않습니다. 버전 5나 6에서 널(NULL) 매개변수를 전달하면 0을 지정한 것과 같습니다. 버전 5와 6의 변환기는 제로(0) 매개변수를 널(NULL) 문자열과 동일한 것으로 간주합니다.

OS/390용 DB2 버전 7에서만 저장 프로시저에서 COMMIT SQL문이 지원됩니다. INVERTDATA 저장 프로시저는 저장 프로시저 내에 테이블을 삭제하고 다시 작성하므로 COMMIT문이 필요합니다. 따라서 OS/390용 DB2 버전 5나 6은 IWH.INVERTDATA가 지원하지 않습니다.

OS/390용 DB2는 Java 사용자 정의 함수(UDF)를 지원하지 않으므로 390 플랫폼에서 IWH.FORMATDATE를 지원하지 않습니다.

Java 저장 프로시저용 샘플 시작 프로시저

(“DB2 for OS/390 Application Programming Guide and Reference for Java”에 설명됨).

```
//DSNWLMJ PROC DB2SSN=DSN,NUMTCB=5,APPLENV=DSNWLMJ <-- WLM ENVIRONMENT value in CREATE PROC
//*****
//* THIS PROC IS USED TO START THE WLM-ESTABLISHED SPAS *
//* ADDRESS SPACE FOR THE DSNWLMJ APPLICATION ENVIRONMENT *
//* V WLM,APPLENV=DSNWLMJ,RESUME *
//*****
//DSNWLMJ EXEC PGM=DSNX9WLM,TIME=1440,REGION=0M,
// PARM='&DB2SSN, &NUMTCB, &APPLENV'
//STEPLIB DD DSN=DSN.TESTLIB,DISP=SHR
// DD DSN=IWH710.SIWHPDSE,DISP=SHR <-- This has the transformers in it
// DD DSN=DSN.HPJSP.PDSE.JDBC,DISP=SHR <-- HPJ DLLs from HPJ setup
// DD DSN=SYS1.PP.PDSELINK,DISP=SHR <-- HPJ runtime libraries
// DD DSN=DSN710.SDSNEXIT,DISP=SHR
// DD DSN=DSN710.SDSNLOAD,DISP=SHR
// DD DSN=SYS1.SCEERUN,DISP=SHR
// DD DSN=DSN.PDSE,DISP=SHR <-- HPJ setup info
//JAVAENV DD DSN=DSN.WLMENVJ.JSPENV,DISP=SHR <-- Environment variables, see below
//CEEDUMP DD SYSOUT=A
//DSSPRINT DD SYSOUT=A
//JSPDEBUG DD SYSOUT=A
//SYSABEND DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
```

샘플 환경 변수 데이터 세트

("DB2 for OS/390 Application Programming Guide and Reference for Java"에 설명됨).

```
ENVAR("TZ=PST07",  
"DB2SQLJPROPERTIES=/usr/lpp/db2/jdbc/db2710/classes/db2sqljdbc.properties",  
"LIBPATH=/usr/lpp/DWC",  
"VWSPATH=/usr/lpp/DWC",  
"CLASSPATH=/usr/lpp/db2/jdbc/db2710/classes:/usr/lpp/DWC:/usr/lpp/hpj/lib"),  
MSGFILE(JSPDEBUG)
```

변환기에 자국어 지원

OS/390 에이전트가 만든 대부분의 메시지는 NT 플랫폼으로 전송되어 해석되므로 대개 메시지 언어는 NT에 UDB를 설치한 방법에 따라 다릅니다. 변환기는 예외입니다. OS/390 에이전트에는 변환기용으로 다음 메시지 파일이 제공됩니다.

파일 이름:	언어:
Xf.properties_Fi_FI	핀란드어(핀란드)
Xf.properties_No_NO	노르웨이어(노르웨이)
Xf.properties_Ru_RU	러시아어(러시아)
Xf.properties_Zh_CN	중국어(중국)
Xf.properties_Zh_TW	대만어
Xf.properties_Da_DK	덴마크어(덴마크)
Xf.properties_De_DE	독일어(독일)
Xf.properties_En_US	영어(미국)
Xf.properties_Es_ES	스페인어(스페인)
Xf.properties_Fr_FR	프랑스어(프랑스)
Xf.properties_It_IT	이탈리아어(이탈리아)
Xf.properties_Ja_JP	일본어(일본)
Xf.properties_Ko_KR	한국어(한국)
Xf.properties_Pt_BR	포르투갈어(브라질)
Xf.properties_Sv_SE	스웨덴어(스웨덴)

변환기 메시지를 영어 외의 언어로 표시하려면 파일 중 하나를 선택한 다음 내용을 Xf.properties에 복사하십시오.

DB2 계열 외의 데이터베이스 액세스

DB2 Universal Database 시스템 이외에 액세스하기 위해, OS/390 에이전트는 DataJoiner를 사용합니다. DataJoiner를 사용하면 에이전트가 에이전트로 들어오는 일반 DRDA 흐름을 UDB 데이터베이스인 것처럼 사용할 수 있습니다. ODBC 요청이 직접 DB2 계열 데이터베이스 이외의 소스로 전송되는 경우, DataJoiner는 추가 계층의 코드를 호출하여 외부 데이터베이스에 액세스합니다.

DataJoiner는 Oracle, Sybase, Informix, Microsoft SQL Server, Teradata 등 ODBC 드라이버를 Windows NT, AIX 또는 Sun의 Solaris 운영 환경에서 실행시키는 데이터베이스에 액세스할 수 있습니다. OS/390 에이전트는 소스로서 DataJoiner에 액세스할 수 있지만, 목표로서는 액세스할 수 없습니다. DataJoiner는 2단계 확약을 지원하지 않습니다. DataJoiner가 버전 2.1과 2.1.1에서 TCP/IP를 응용프로그램 요청자로서 지

원하지만, 응용프로그램 서버(AS)는 없습니다. OS/390 에이전트가 TCP/IP를 사용하려면 응용프로그램 서버가 있어야 하므로 OS/390에서 DataJoiner에 액세스하려면 대신 SNA 연결을 사용하십시오.

OS/390에서 IMS 및 VSAM 액세스

Classic Connect는 웨어하우스 에이전트와 별도로 구입하여 설치됩니다. OS/390 에이전트는 Classic Connect ODBC 드라이버를 통해 IMS 및 VSAM을 액세스할 수 있습니다. Classic Connect를 사용하면 IMS 및 VSAM 데이터 세트에 DB2와 유사한 정의를 설정하고 ODBC를 사용하여 액세스할 수 있습니다.

OS/390 에이전트에는 요청이 Classic Connect로 전송되는지 DB2로 전송되는지에 따라 알맞은 ODBC 드라이버를 로드합니다. DB2 소스를 액세스하는 경우, 에이전트는 DB2 ODBC 드라이버를 로드합니다. VSAM 또는 IMS 소스에 액세스하는 경우 에이전트는 Classic Connect ODBC 드라이버를 로드합니다. 그런 다음 에이전트의 요청이 처리됩니다.

Classic Connect ODBC 드라이버 및 웨어하우스 액세스 설정

Classic Connect는 OS/390 에이전트와 별도로 구입하여 설치됩니다. Classic Connect는 단일 파일이나 파일의 일부를 하나 이상의 관계형 테이블로 볼 수 있습니다. 데이터에 액세스하려면 Classic Connect의 IMS 및 VSAM 데이터를 맵핑해야 합니다. 직접 데이터를 맵핑하거나 Microsoft Windows Classic Connect 비관계형 데이터 매퍼를 사용하여 맵핑할 수 있습니다.

1. OS/390에 Classic Connect 데이터 서버 설치
2. 선택적: NT에 Classic Connect Data Mapper 설치
3. Classic Connect가 데이터를 관계형으로 액세스할 수 있도록 Classic Connect의 논리적 테이블 정의를 정의합니다. 데이터 매퍼를 사용하여 IMS 및 VSAM 구조의 정의를 작성하거나 직접 정의할 수 있습니다.
4. Classic Connect를 설정한 후 웨어하우스의 액세스를 설정할 수 있습니다.

- a. Classic Connect .ini 파일을 작성하십시오.

ca.ini 샘플 Classic Connect 응용프로그램 구성 파일은 /usr/lpp/DWC/ 디렉토리에 있습니다. 다음은 파일을 다시 만든 것입니다.

```
* national language for messages
NL = US English
* resource master file
NL CAT = usr/lpp/DWC/v4r1m00/msg/engcat
FETCH BUFFER SIZE = 32000
DEFLOC = CXASAMP
USERID = uid
USERPASSWORD = pwd
DATASOURCE = DJX4DWC tcp/9.112.46.200/1035
MESSAGE POOL SIZE = 1000000
```

- b. .ini 파일의 DATASOURCE 행을 갱신하십시오. 이 줄에는 데이터 소스 이름 및 프로토콜 주소가 들어 있습니다. 데이터 소스 이름은 Classic Connect 데이터 서버에서 정의된 조회 프로세서와 상응해야 하며, 이름은 데이터 서버 구성 파일의 QUERY PROCESSOR SERVICE INFO ENTRY에 있습니다. 프로토콜 주소는 같은 파일에 있는 TCP/IP SERVICE INFO 항목에서 찾을 수 있습니다. 이 파일에 있는 USERID 및 USERPASSWORD는 웨어하우스 데이터 소스를 정의할 때 사용됩니다.
- c. .ini 파일과 같은 디렉토리에 있는 Classic Connect 프로그램 파일로 CXA_CONFIG 환경 변수를 내보내십시오.
- d. .ini 파일과 같은 디렉토리에 있는 Classic Connect 프로그램 파일에 대한 경로가 포함되도록 LIBPATH 환경 변수를 갱신하십시오.
- e. 선택적: cxasamp 테스트 프로그램을 사용하여 설치를 확인하십시오. .ini 파일이 있는 디렉토리에서 cxasamp를 입력하십시오. location/uid/pwd는 .ini 파일에서 정의된 name/userid/userpassword 데이터 소스입니다.
- f. DB2 데이터 소스를 정의하는 방법으로 데이터 소스를 웨어하우스에 정의하십시오.

OS/390용 DB2에는 드라이버 관리 프로그램이 없으므로 dsnaoini 파일을 갱신할 필요가 없습니다. Classic Connect용 드라이버 관리 프로그램이 OS/390 에이전트에서 빌드되었습니다.

OS/390용 DB2 유틸리티 수행

에이전트를 사용하여 유틸리티를 수행하려면 OS/390 에이전트에 apar PQ44904를 적용해야 합니다.

DSNUTILS는 WLM 및 RRS 환경에서 수행되는 OS/390용 DB2 저장 프로시저어입니다. 사용자 정의 저장 프로시저어 인터페이스를 사용하여 설치된 DB2 유틸리티를 DSNUTILS를 사용하여 실행할 수 있습니다. OS/390용 DB2의 LOAD, REORG, RUNSTATS 유틸리티에 있는 등록정보 슈트를 사용하여 유틸리티 실행 방법을 변경할 수 있습니다. 유틸리티의 등록정보를 변경하려면 프로세스 모델 편집기 창에서 유틸리티를 마우스 오른쪽 버튼으로 누르고 등록정보를 누르십시오.

Warehouse Manager는 DSNUTILS 인터페이스가 있으므로 Warehouse Manager 단계에 DB2 유틸리티를 포함할 수 있습니다.

DSNUTILS 저장 프로시저어를 설정하려면 다음을 수행하십시오.

1. DB2를 설치할 때 DSNTIJSJG 작업을 수행하여 DSNUTILS 저장 프로시저어를 설정하고 바인드하십시오. DSNUTILS 정의에 PARAMETER STYLE GENERAL이 있는지 확인하십시오.
2. WLM 관리 저장 프로시저어를 사용하도록 설정하십시오.

3. RRS 및 WLM 환경을 설정하십시오.
4. DB2에 제공되는 샘플 배치 DSNUTILS 프로그램을 실행하십시오(권장, 필수는 아님).
5. CLI가 저장 프로시저를 호출할 수 있도록 DSNUTILS 계획과 DSNCLI 계획을 바인드하십시오.

```
BIND PLAN(DSNAOCLI) PKLIST(*.DSNAOCLI.*, *.DSNUTILS.*)
```
6. Warehouse Manager를 사용하여 단계를 설정하고 그 단계를 수행하십시오. 데이터 상주 유형은 APPEND이어야 합니다. 아니면, Warehouse Manager가 유틸리티를 실행하기 전에 테이블 내용을 모두 삭제합니다.

LOAD 유틸리티를 사용하여 OS/390용 DB2 테이블 간에 데이터 복사

테이블을 플랫폼 파일에 로드 해제한 다음 플랫폼 파일을 다른 테이블에 로드하여 테이블을 복사하는 경우입니다. 일반적으로 데이터를 로드 해제하고 로드 해제로 작성된 로드 제어 명령문을 편집한 다음 데이터를 로드해야 합니다. 웨어하우스를 사용하면 각 단계마다 중지할 필요 없이 다른 테이블에 재로드하고 직접 제어 명령문을 편집하도록 지정할 수 있습니다. 방법은 다음과 같습니다. Reorg/Generic 인터페이스에서 UNLOAD나 REORG TABLESPACE 유틸리티를 사용하여 파일을 로드 해제하는 단계를 작성합니다. 두 가지 유틸리티는 두 개의 데이터 세트 즉, 테이블 데이터가 있는 데이터 세트와 LOAD에 입력 가능한 유틸리티 제어 명령문이 있는 데이터 세트를 작성합니다. 유틸리티가 생성한 제어 명령문에서 INTO TABLE 테이블 이름은 로드 해제된 테이블의 이름입니다. 다음은 Reorg Unload 단계에 사용할 수 있는 DSNUTILS 매개변수 예입니다.

표 33. Reorg Unload 단계의 등록정보

UTILITY_ID	REORGULX
RESTART	NO
UTSTMT	REORG TABLESPACE DBVW.USAINENT UNLOAD EXTERNAL
UTILITY_NAME	REORG TABLESPACE
RECDSN	DBVW.DSNURELD.RECDSN
RECDEVT	SYSDA
RECSpace	50
PNCHDSN	DBVW.DSNURELD.PNCHDSN
PNCHDEVT	SYSDA
PNCHSPACE	3

Reorg/Generic DSNUTILS 인터페이스를 사용하여 로드 단계를 작성하십시오. 일반적으로 DSNUTILS 유틸리티 명령문 매개변수는 유틸리티 제어 명령문을 지정합니다. 웨어하우스 유틸리티 인터페이스는 또한 유틸리티 명령문 필드에 파일 이름을 허용합니다. :FILE: 키워드를 사용하여 유효한 제어 명령문이 있는 파일을 지정하고, :TABLE: 키

위드를 사용하여 로드할 테이블 이름을 지정할 수 있습니다. LOAD 유틸리티를 사용하여 이전 예의 출력으로 작업하려면 LOAD 등록정보에 다음 매개변수 값을 적용하십시오.

주: UTSTMT 필드에서 로드 명령문이나, UNLOAD EXTERNAL 옵션을 지정하여 REORG 유틸리티를 실행했을 때 출력된 파일 이름을 입력하십시오.

표 34. LOAD 단계 등록정보

UTILITY_ID	LOADREORG
RESTART	NO
UTSTMT	:FILE:DBVW.DSNURELD.PNCHDSN:TABLE:[DBVW]. INVENTORY
UTILITY_NAME	LOAD
RECDSN	DBVW.DSNURELD.RECDSN
RECDEVT	SYSDA

동일한 DB2 서브시스템이나 다른 DB2 서브시스템에서 모든 OS/390용 DB2 소스 및 목표 테이블에 적용됩니다. 제어 명령문 플랫폼 파일은 HFS나 원시 MVS 파일입니다.

OS/390에 사용 가능한 DSNUTILS 및 DB2 유틸리티에 관해 자세한 정보는 DB2 for OS/390 Utility Guide and Reference를 참조하십시오.

복제

OS/390 에이전트를 사용하여 Data Propagator 복제 적용 단계를 자동화할 수 있습니다. 복제에는 소스 데이터베이스, 제어 데이터베이스 및 목표 데이터베이스가 필요합니다. 이들은 다른 또는 같은 데이터베이스일 수 있습니다. 캡처 작업에서는 DB2 로그를 읽고 소스 데이터베이스에서 추가, 갱신 또는 변경된 행을 판별한 다음, 변경사항을 변경 데이터 테이블에 기록합니다. 그런 다음 목표 데이터베이스에 변경사항을 적용하기 위해 적용 작업이 수행됩니다. DB2 Warehouse Manager 패키지는 복제 단계를 작성하여 적용 작업의 실행을 자동화할 수 있습니다. Warehouse Manager를 사용하여 수행할 적용 작업의 유형 및 수행 시기를 정의하십시오. SASNLINK 라이브러리를 steplib 환경 변수로 내보내야 합니다.

Data Warehouse Center 템플릿에 복제 지원 추가

Data Warehouse Center는 복제 지원을 위한 JCL 템플릿을 포함합니다. OS/390 에이전트를 사용하여 Apply 프로그램을 수행하려면 템플릿의 계정과 데이터 세트 정보를 사용자의 OS/390 시스템에 맞도록 수정해야 합니다. 템플릿을 변경하려면 다음을 수행하십시오.

1. /usr/lpp/DWC/ 디렉토리의 파일을 복사 및 갱신할 수 있는 권한을 가진 ID로 로그인하십시오.

2. apply.jcl을 찾아 이 파일을 systemname.apply.jcl로 복사하십시오. 여기서 systemname은 MVS 시스템의 이름입니다. 예를 들어 STLMVS1에 있는 파일 사본의 이름은 STLMVS1.apply.jcl입니다.
3. 텍스트 편집기를 사용하여 사이트의 요구 사항에 맞도록 JCL을 사용자 정의하십시오. MVS 시스템의 표준 계정 정보와 일치하도록 계정 정보를 수정하고 STEPLIB DD 및 MSGS DD의 데이터 세트를 수정하십시오.
4. 필요하다면, EXEC 카드의 프로그램 이름을 변경하십시오. 프로그램 이름 변경에 관한 세부사항은, DB2 복제 안내 및 참조서를 보십시오. [USERID] 및 [APPLY_PARMS]와 같이 대괄호 안에 있는 매개변수는 수정하지 마십시오(대괄호는 각각 16진 문자 x'AD'와 x'BD'입니다). SPF 옵션 0에서 TSO 터미널 유형을 3278A로 설정하지 않은 경우에는 이 값이 대괄호가 아닌 특수 문자로 표시됩니다. x'AD'나 x'BD' 또는 문자 사이의 데이터를 수정하지 않는다면, 이는 문제가 되지 않습니다.
5. 환경 변수 VWS_TEMPLATES가 복사된 템플릿 파일의 디렉토리를 가리키도록 갱신하십시오.

다음 예는 Data Warehouse Center와 함께 제공되는 JCL 템플릿입니다.

Apply JCL 템플릿

```
//[USERID]A JOB ,MSGCLASS=H,MSGLEVEL=(1,1),
// REGION=2M,TIME=1440,NOTIFY=&SYSUID
//* DON'T CHANGE THE FIRST LINE OF THIS TEMPLATE.
//* THE REMAINING JCL SHOULD BE MODIFIED FOR YOUR SITE.
/*****
//* RUN APPLY/MVS ON OS/390 DB2 6.1 *
/*****
//ASRARUN EXEC PGM=ASRAPV66,REGION=10M,
// [APPLY_PARMS]
//STEPLIB DD DISP=SHR,DSN=DPROPR.V6R1M0.SASNLINK
// DD DISP=SHR,DSN=DSN610.SDSNLOAD
//MSGS DD DSN=DPROPR.V2R1M0A.MSGS,DISP=SHR
//ASNASPL DD DSN=&&ASNASPL,DISP=(NEW,DELETE,DELETE),
// UNIT=SYSDA,SPACE=(CYL,(10,1)),
// DCB=(RECFM=VB,BLKSIZE=6404)
//SYSTEM DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//
```

에이전트 로그

서버, 로그 프로그램, 에이전트 및 Data Warehouse Center 프로그램은 VWS_LOGGING 환경 변수에 지정된 로그 디렉토리에 로그를 기록합니다. 로그 파일은 일반 텍스트입니다.

Data Warehouse Center에서 에이전트 로그를 시작할 수 있습니다. 오른쪽 창에서 **Warehouse**를 누르고 등록정보를 누르십시오. 추적 레벨 탭에서 원하는 추적 레벨로 설정을 변경하십시오. 에이전트 추적에서는 0-4 레벨을 지원합니다.

- 레벨 1 - 시작/종료 추적
- 레벨 2 - 레벨 1 + 디버그 추적
- 레벨 3 - 레벨 2 + 데이터 추적
- 레벨 4 - 내부 버퍼 추적

추적 레벨이 1보다 높으면, 성능이 저하됩니다. 디버그 목적으로만 추적을 설정해야 합니다. 추적 정보는 AGNTxxx.LOG 파일에 저장되고 환경 정보는 AGNTxxx.SET 파일에 저장됩니다.

Windows NT에서 클라이언트쪽 캐싱

사용자가 공유 드라이브를 사용하여 DB2 Datalinks를 설치한 Windows NT 서버 머신에 있는 READ PERM DB 파일을 유효한 토큰을 사용하여 액세스하려 할 경우, 예상대로 파일이 열립니다. 하지만, 그 후에, 동일한 토큰을 사용하는 이어지는 열기 요청은 서버에 실제로 도달하지 않고, 클라이언트의 캐쉬로부터 서비스를 받습니다. 심지어 토큰이 만료된 후에도, 항목이 캐쉬에 아직 있으므로 사용자가 파일의 내용을 계속 볼 수 있습니다. 이 문제는 파일이 Windows NT 워크스테이션에 있을 경우에는 발생하지 않습니다.

솔루션은 Windows NT 서버에서 레지스트리 항목

`\\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Lanmanserver\Parameters\EnableOpLocks`를 영(0)으로 설정하는 것입니다. 이 레지스트리를 설정하면, 클라이언트가 공유 드라이브를 통해 서버에 상주하는 파일을 액세스할 때마다, 클라이언트 캐쉬에서 서비스되지 않고 요청이 항상 서버에 도달합니다. 따라서, 토큰은 모든 요청에 대해 재검증됩니다.

이 솔루션은 공유 드라이브를 통한 서버로부터의 모든 파일 액세스의 전반적인 성능에 영향을 준다는 단점이 있습니다. 이렇게 설정하더라도, 다른 클라이언트 머신으로부터의 액세스와는 달리 서버 자체에 있는 공유 드라이브 맵핑을 통해 파일을 액세스할 경우, 요청은 여전히 캐쉬로부터 서비스 받는 것처럼 보입니다. 따라서, 토큰 만기에 영향을 받지 않습니다.

주: 어떤 경우라도 파일 액세스가 지역 액세스이고 공유 드라이브를 통하지 않으면, 토큰 검증 및 이어지는 토큰 만기는 예상대로 발생합니다.

Enterprise Edition UNIX CD-ROM에서의 시험 사용판

UNIX 플랫폼용 DB2 Universal Database(UDB) Enterprise Edition(EE) CD-ROM 버전 6 및 버전 7에는 DB2 Connect Enterprise Edition(CEE)의 90일 시험 사용판이 들어 있습니다. DB2 Connect 기능성은 DB2 UDB EE 제품으로 빌드되므로, DB2 Connect 기능성을 사용하기 위해 DB2 UDB EE가 설치된 시스템에 DB2 CEE 제품

을 설치할 필요가 없습니다. DB2 CEE의 90일 시험 사용판을 설치하고 사용권 버전으로 업그레이드하려고 결정한 경우, DB2 CEE 제품을 구입하고 DB2 CEE 사용권 키를 설치해야 합니다. 제품을 재설치할 필요가 없습니다. 사용권 키를 설치하는 명령어는 UNIX용 DB2 EE 또는 DB2 CEE 빠른 시작 책에서 제공됩니다.

EE 설치와 함께 시험 CEE 제품을 설치한 경우, 다음 명령어로 CEE 90일 시험 사용판을 제거할 수 있습니다. Connect EE의 시험 사용판을 제거하면, DB2 CEE와 함께 사용할 수 있는 DB2 Connect를 여전히 갖습니다.

DB2 Connect 버전 7을 제거하려면, 각 플랫폼에서 다음 Fileset를 설치 제거하십시오.

- AIX에서, db2_07_01.clic Fileset를 설치 제거하십시오.
- NUMA-Q 및 Solaris 운영 환경에서, db2clic71 패키지를 설치 제거하십시오.
- Linux에서, db2clic71-7.1.0-x RPM을 설치 제거하십시오.
- HP-UX에서, DB2V7CONN.clic Fileset를 설치 제거하십시오.

DB2 Connect 버전 6을 제거하려면, 각 플랫폼에서 다음 Fileset를 설치 제거하십시오.

- AIX에서, db2_06_01.clic Fileset를 설치 제거하십시오.
- NUMA-Q 및 Solaris 운영 환경에서, db2cplic61 패키지를 설치 제거하십시오.
- Linux에서, db2cplic61-6.1.0-x RPM을 설치 제거하십시오.
- HP-UX에서, DB2V6CONN.clic Fileset를 설치 제거하십시오.

DB2 Connect Enterprise Edition UNIX CD-ROM에서의 시험 사용판

UNIX 플랫폼용 DB2 Connect Enterprise Edition(EE) CD-ROM 버전 6 및 버전 7에는 DB2 Universal Database(UDB)의 90일 시험 사용판이 들어 있습니다. DB2 UDB EE 90일 시험 사용판은 평가를 위해 제공되지만, 작업할 DB2 Connect에는 필요하지 않습니다.

DB2 UDB EE의 90일 시험 사용판을 설치하고 사용권 버전으로 업그레이드하려고 결정한 경우, DB2 UDB 제품을 구입하고 DB2 UDB EE 제품을 구입하고 DB2 UDB EE 사용권 키를 설치해야 합니다. 제품을 재설치할 필요가 없습니다. 사용권 키를 설치하는 명령어는 UNIX용 DB2 EE 또는 DB2 CEE 빠른 시작 책에서 제공됩니다. Connect EE 설치와 함께 시험 UDB EE 제품을 설치한 경우, 다음 명령어로 EE 90일 시험 사용판을 제거할 수 있습니다. DB2 UDB EE의 시험 사용판을 제거한 경우, DB2 Connect EE의 기능성에 영향을 미치지 않습니다.

DB2 UDB EE 버전 7을 제거하려면, 각 플랫폼에서 다음 Fileset를 설치 제거하십시오.

- AIX에서, db2_07_01.elic Fileset를 설치 제거하십시오.
- NUMA-Q 및 Solaris 운영 환경에서, db2elic71 패키지를 설치 제거하십시오.

- Linux에서, db2elic71-7.1.0-x RPM을 설치 제거하십시오.
- HP-UX에서, DB2V7ENTP.elic Fileset를 설치 제거하십시오.

DB2 UDB EE 버전 6을 제거하려면, 각 플랫폼에서 다음 Fileset를 설치 제거하십시오.

- AIX에서, db2_06_01.elic Fileset를 설치 제거하십시오.
- NUMA-Q 및 Solaris 운영 환경에서, db2elic61 패키지를 설치 제거하십시오.
- Linux에서, db2elic61-6.1.0-x RPM을 설치 제거하십시오.
- HP-UX에서, DB2V6ENTP.elic Fileset를 설치 제거하십시오.

Data Links Manager 삭제

이제 지정된 데이터베이스에 대한 DB2 Data Links Manager를 삭제할 수 있습니다. 백업/복원과 같은 유틸리티와 마찬가지로 일부 Data Links 관련 SQL 요청의 처리는 데이터베이스에 구성된 모든 DLM과의 통신을 포함합니다. 이전에, 비록 DB2는 조작 가능하지 않았지만 구성된 DLM을 삭제하는 기능을 갖지 않았습니다. 이것은 SQL 및 유틸리티 처리에 추가적인 오버헤드를 가져옵니다. DLM이 추가되었으면, 일부 SQL 요청(예를 들어, 테이블/테이블공간/데이터베이스 삭제)의 실패를 가져올 수 있는 요청 처리에서 엔진이 DLM과 통신했습니다.

SMIT May Remove Additional Filesets를 사용하여 DLFM 구성요소 설치 제거

Data Links Manager가 설치되어 있는 AIX 머신에서 DB2(버전 5, 6 또는 7)를 설치 제거하기 전에, 다음 단계를 따르십시오.

1. 루트로서, 명령을 사용하여 /etc/vfs의 사본을 작성하십시오.


```
cp -p /etc/vfs /etc/vfs.bak
```
2. DB2를 설치 제거하십시오.
3. 루트로서, 단계 1에서 작성한 백업 사본으로 /etc/vfs를 바꾸십시오.


```
cp -p /etc/vfs.bak /etc/vfs
```

Windows 2000에서 CLP 사용시 SQL1035N 오류 발생

일부 사용자(예를 들어, 관리자)만이 쓰기 권한을 가진 디렉토리에 DB2가 설치된 경우, DB2 명령행 처리기를 사용하려고 시도할 때 일반 사용자는 오류 SQL1035N가 발생할 수 있습니다.

이 문제점을 해결하려면, DB2는 모든 사용자가 쓰기 액세스를 가진 디렉토리에 설치되어야 합니다.

SQL 지원에 대해 개선된 기능

SQL 지원 도구는 이제 사용자가 테이블 조인에 대해 "=" 이외의 조인 연산자를 지정하게 합니다. SQL 지원 도구의 조인 페이지에서 조인 유형 버튼을 눌러 시작된 조인 유형 대화 상자는 조인 연산자의 드롭 다운 목록을 포함하도록 확장되었습니다.

사용 가능한 연산자는 "=", "<>", "<", ">", "<=" 및 ">="입니다. SQL 지원은 단순 SQL문 작성시 사용자를 돕는 도구입니다. Command Center(대화식 탭)에서 사용 가능하며, Control Center(뷰 작성 및 트리거 작성 대화 상자, Stored Procedure Builder("SQL Stored Procedure 삽입" 마법사) 및 Data Warehouse Center(SQL 프로세스 단계)에서 사용 가능합니다.

Linux에서 DB2용 Gnome 및 KDE 바탕 화면 통합

DB2는 이제 지원되는 Intel 기반 Linux 분산 프로그램용 Gnome 및 KDE 바탕 화면에 가장 일반적으로 사용되는 DB2 도구를 시작하는 DB2 폴더 및 아이콘을 작성하기 위한 유틸리티 세트가 제공됩니다. DB2 버전 7.2에서 기본 설치되며, 설치한 유틸리티를 사용하여 하나 이상 선택된 사용자의 바탕 화면 아이콘을 작성하거나 제거할 수 있습니다.

하나 이상 선택된 사용자의 바탕 화면 아이콘 세트를 추가하려면 다음 명령을 사용하십시오.

```
db2icons <user1> [<user2> <user3>...]
```

주: Gnome이나 KDE 바탕 화면 환경을 실행할 때 아이콘이 생성된 경우에는 바탕 화면을 직접 새로 고쳐야 새 아이콘이 표시됩니다.

하나 이상 사용자의 바탕 화면에서 바탕 화면 아이콘 세트를 제거하려면 다음 명령을 사용하십시오.

```
db2rmicons <user1> [<user2> <user3>...]
```

주: 다른 사용자의 아이콘을 생성하거나 제거하려면 충분한 권한이 있어야 합니다. 일반 사용자는 대개 **db2icons** 및 **db2rmicons**를 사용하여 자신의 아이콘을 작성하거나 제거할 수 있고, 루트 사용자나 지정 사용자의 홈 디렉토리에 대한 쓰기 권한을 가진 사용자만 다른 사용자의 아이콘을 작성하거나 제거할 수 있습니다.

Windows 2000 Terminal Server의 관리 모드에서 DB2 실행

DB2 UDB 버전 7.1, FixPak 3 이상을 사용할 때는 DB2를 Windows 2000 Terminal Server의 관리 모드에서 실행할 수 있습니다. 이전 버전에서는 Windows 2000 Terminal Server의 관리 모드에서 클라이언트 세션 하에 DB2를 실행할 수 없습니다.

백업 및 복원 명령에 대한 온라인 도움말

db2 ? backup을 입력하면 잘못된 정보가 나타납니다. 올바른 정보는 다음과 같습니다.

```
BACKUP DATABASE database-alias [USER username [USING password]]
[TABLESPACE (tblspace-name [ {,tblspace-name} ... ])] [ONLINE]
[INCREMENTAL [DELTA]] [USE TSM [OPEN num-sess SESSIONS]] |
TO dir/dev [ {,dir/dev} ... ] | LOAD lib-name [OPEN num-sess SESSIONS]]
[WITH num-buff BUFFERS] [BUFFER buffer-size] [PARALLELISM n]
[WITHOUT PROMPTING]
```

db2 ? restore를 입력하면 잘못된 정보가 나타납니다. 올바른 정보는 다음과 같습니다.

```
RESTORE DATABASE source-database-alias { restore-options | CONTINUE | ABORT };
restore-options:";
[USER username [USING password]] [{TABLESPACE [ONLINE] |};
TABLESPACE (tblspace-name [ {,tblspace-name} ... ]) [ONLINE] |";
HISTORY FILE [ONLINE]]] [INCREMENTAL [ABORT]]";
[USE TSM [OPEN num-sess SESSIONS] |";
FROM dir/dev [ {,dir/dev} ... ] | LOAD shared-lib";
[OPEN num-sess SESSIONS]]] [TAKEN AT date-time] [TO target-directory]";
[INTO target-database-alias] [NEWLOGPATH directory]";
[WITH num-buff BUFFERS] [BUFFER buffer-size]";
[DLREPORT file-name] [REPLACE EXISTING] [REDIRECT] [PARALLELISM n]";
[WITHOUT ROLLING FORWARD] [WITHOUT DATALINK] [WITHOUT PROMPTING]";
```

"Warehouse Manager"는 "DB2 Warehouse Manager"이어야 함

제품 화면과 제품 문서에서 나타나는 모든 문구 "Warehouse Manager"는 "DB2 Warehouse Manager"를 의미합니다.

제9부 추가 정보

추가 정보

DB2 Universal Database 및 DB2 Connect 온라인 지원

이 문서가 발행된 이후로 발견된 발행에 대한 정보를 비롯하여 완전하고 최신의 DB2 정보 소스를 보려면, <http://www.ibm.com/software/data/db2/udb/winos2unix/support>에 있는 DB2 Universal Database 및 DB2 Connect 온라인 지원 웹 사이트를 사용하십시오.

DB2 잡지

DB2 제품군에 대한 최신 정보를 보려면, "DB2 잡지"를 무료로 구독하십시오. 잡지의 온라인 판은 <http://www.db2mag.com>에서 사용할 수 있으며 구독 요청을 위한 지시사항도 이 사이트에 게시되어 있습니다.

제10부 부록

부록. 주의사항

IBM은 다른 나라에서는 이 문서에서 언급한 제품, 서비스 또는 부속 기능을 제공하지 않을 수도 있습니다. 현재 사용 가능한 제품 및 서비스에 대한 정보는 한국 IBM 영업 대표에 문의하십시오. IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 반드시 IBM 제품, 프로그램 또는 서비스만을 사용하라는 의미는 아닙니다. IBM의 지적 재산을 침해하지 않는 한, 타사의 기능상 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스 운용에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 문서에서 다루어지는 내용에 대해 특허를 보유하고 있거나 특허 출원 중일 수 있습니다. 이 문서를 제공한다고 해서 특허에 대한 사용권까지 제공하는 것은 아닙니다. 사용권에 대한 의문사항은 다음으로 문의하시기 바랍니다.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

전화번호: 080-023-8080

2 바이트(DBCS) 정보에 관한 사용권 의문사항은 한국 IBM 고객만족센터 또는 다음 주소로 서면 문의하시기 바랍니다.

IBM World Trade Asia Corporation

Licensing

2-31 Roppongi 3-chome, Minato-ku

Tokyo 106, Japan

다음 단락은 해당 규정이 현지법과 상충되는 영국이나 기타 나라에서는 적용되지 않습니다. IBM은 이 책을 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여(단, 이에 한하지 않음) 어떠한 종류의 묵시적 또는 명시적 보증없이 『현상태대로』 제공합니다. 일부 국가에서는 특정 거래에 있어서 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 정기적으로 변경되며 변경사항은 신판에 통합됩니다. IBM은 이 책에 기술된 제품 및/또는 프로그램을 사전 통고 없이 언제든지 변경하거나 개선할 수 있습니다.

이 정보에서 언급되는 비IBM 웹 사이트는 편의상 제공된 것으로 이들 웹 사이트를 옹호하기 위한 것은 절대 아닙니다. 이들 웹 사이트에 나오는 자료들은 이 IBM 제품에 대한 자료의 일부가 아니며 이들 웹 사이트의 사용은 사용자 본인의 책임입니다.

IBM은 적당하다고 생각하는 방식으로 사용자가 제공하는 정보를 사용하거나 배포할 수 있으며 사용자는 이에 대해 책임을 지지 않습니다.

(i) 독자적으로 작성된 프로그램과 다른 프로그램(이 프로그램을 포함) 사이의 정보 교환 및 (ii) 교환된 정보의 상호 이용을 목적으로 정보를 필요로 하는 이 프로그램의 사용권자는 다음으로 문의해야 합니다.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

이러한 정보는 해당 조항 및 조건(예를 들면, 사용료 지불 등)에 따라 사용할 수 있습니다.

이 정보에 기술된 사용권 프로그램과 이에 사용할 수 있는 모든 사용권 자료는 IBM 기본 계약, IBM 프로그램 사용권 계약 또는 이에 준하는 당사와의 계약을 통해 제공됩니다.

여기에 있는 모든 성능 데이터는 제한된 환경에서 산출된 것입니다. 따라서, 다른 기동 여건에서 얻어진 결과는 상당히 다를 수 있습니다. 일부 측정치는 개발 레벨 시스템에서 작성되었을 수 있으며, 이러한 측정치가 일반적으로 사용 가능한 시스템에서도 동일하다고는 보장하지 않습니다. 더우기, 일부 측정치는 추정을 통해 추측되었을 수도 있습니다. 실제 결과는 다를 수 있습니다. 이 책의 사용자는 본인의 특정 환경에 적용할 수 있는 데이터를 검증해야 합니다.

비IBM 제품에 대한 정보는 이들 제품의 공급업체, 해당 업체의 공개자료 또는 기타 범용 자료로부터 입수한 것입니다. IBM은 이들 제품을 테스트하지 않았으며 비IBM 제품과 관련한 배상 청구, 호환성 또는 성능의 정확성 등에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대해서는 이들 제품의 공급업체에 문의하십시오.

IBM이 제시하는 방향 또는 의도에 관한 어떠한 언급도 특별한 통지없이 변경될 수 있습니다.

이 정보에는 일상의 업무에서 사용되는 자료와 보고의 예제가 포함되어 있을 수 있습니다. 가능한 완벽하게 설명하기 위해 개인, 회사, 상표 및 제품의 이름이 예제에 들어 있습니다. 이들 이름은 모두 가공의 것이며 실제 기업의 이름 및 주소와 유사하더라도 이는 전적으로 우연입니다.

저작권:

이 정보에는 여러 가지 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 응용프로그램이 포함되어 있을 수 있습니다. 샘플 응용프로그램의 작성 기준이 된 운영 플랫폼의 응용프로그램 프로그래밍 인터페이스에 부합하는 응용프로그램을 개발, 사용,

마케팅 또는 배포를 목적으로 이들 샘플 프로그램을 복사, 수정 및 배포할 수 있으며 IBM에 대한 지불 의무는 없습니다. 이러한 예제는 모든 조건하에서 철저히 테스트된 것은 아닙니다. 따라서 IBM은 이들 프로그램의 신뢰성, 실용성 또는 기능에 대해 보증할 수 없습니다.

각 사본이나 이들 샘플 프로그램의 일부 또는 파생본에는 다음과 같은 저작권 주의사항을 포함시켜야 합니다.

© (귀하의 회사명) (연도). 이 코드 부분은 IBM 샘플 프로그램에 나와 있습니다. © Copyright IBM Corp. _연도 입력_. All rights reserved.

등록상표 및 서비스 상표

별표(*)로 표시된 다음 용어는 미국과 기타 국가에서 IBM Corporation의 등록상표입니다.

ACF/VTAM	IBM
AISPO	IMS
AIX	IMS/ESA
AIX/6000	LAN DistanceMVS
AIXwindows	MVS/ESA
AnyNet	MVS/XA
APPN	Net.Data
AS/400	OS/2OS/390
BookManager CICS	OS/400
C Set++	PowerPC
C/370	QBIC
DATABASE 2	QMF
DataHub	RACF
DataJoiner	RISC System/6000
DataPropagator	RS/6000S/370
DataRefresher	SP
DB2	SQL/DS
DB2 ConnectDB2 Extenders	SQL/400
DB2 OLAP Server	System/370
DB2 Universal Database	System/390
Distributed Relational	SystemView
Database Architecture	VisualAge
DRDA	VM/ESA
eNetwork	VSE/ESA
Extended Services	VTAM
FFST	WebExplorer
First Failure Support Technology	WIN-OS/2

다음 용어는 기타 회사의 등록상표입니다.

Microsoft, Windows 및 Windows NT는 Microsoft Corporation의 상표 또는 등록상표입니다.

Java 또는 모든 Java 관련 상표와 로고 및 Solaris는 미국과 기타 국가에서 Sun Microsystems, Inc.의 등록상표입니다.

Tivoli 및 NetView는 미국과 기타 국가에서 Tivoli Systems Inc.의 등록상표입니다.

UNIX는 미국과 기타 국가에서 X/Open Company Limited가 독점적인 사용권을 가진 등록상표입니다.

이중 별표(**)로 표시된 기타 회사, 제품 또는 서비스 이름은 다른 회사의 등록상표 또는 서비스 상표입니다.

색인

[가]

가져오기, 가져오기, 태그 언어 파일,
DWC 241
가져오기, 태그 언어 파일, ICM, Information
Catalog Manager 242
검색 발견 526
게이트웨이
3층(three-tier) 환경 389
경보 센터 73
교착 상태 189
구성 매개변수
app_ctl_heap_sz 116
maxappls 117
maxlocks 114
MIN_DEC_DIV_3 115
mon_heap_sz 117
softmax 117
구성요소 추적 데이터 파일 이름 262
권한 부여
순차에서 공용 작성 473
순차에서 권한 부여 작성 473

[나]

높은 사용 가능성 82

[다]

다음 키 잠금 189
다중 사이트 갱신 테스트 연결
인스턴스 버전 비호환성 216
다중 색인 액세스 110
데이터 암호화 81
데이터 유형
유니코드 데이터베이스에서의 승격 504
데이터 이동
유니코드 클라이언트의 한계 515
데이터 정리, 이름 및 주소 정리 247
데이터베이스 작성 마법사
Extent 크기 설정 225
동적 복합 텍스트 명령문 104, 455

[라]

런타임 클라이언트를 사용하여 데이터베이스 유
틸리티 바인딩 387
레지스트리 변수
백업 서비스 API를 지원 535
신규 및 변경 119
DB2DOMAINLIST 528
DB2_AWE 112
로그
미러링 90
요구에 따라 아카이브 90
로케일
관리 서버 및 인스턴스 간의 호환성 532
관리 서버와 인스턴스 사이의 호환성 71
자국 언어 지원 67
Linux의 일본어 및 중국어 7
Red Flag Linux의 중국어 7
로케일 설정
CLI를 사용하는 응용프로그램 71
루틴
프로시저어 441

[마]

마법사
데이터베이스 작성 225
MQSeries 지원 225
OLE DB 지원 225
메모리 창
HP-UX 11 526
메시지
캡처 및 적용 181
메시지, MQ Series 256, 257
명령문 레벨 분리 103
문자 변환
문자열 결합 작업 규칙 507
문자열 비교시 규칙 507
문자열
할당, 개요 505
미러링
로그 90

[바]

바꾸기와 함께 증가 확약 262
발트어
코드 페이지 지원 71
백업
오프라인
분리 미리 이미지로부터 87
증분 84
Named Pipe에 87
백업 서비스 API
DB2 지원 535
버퍼 풀
AWE 111
변환 규칙
문자열 결합 작업 507
문자열 변환 507
병렬 복구 86
병합
연산자 508
복구
병렬 86
증분 84
DB2 Data Links Manager와 상호작용
94
복사 다면 528
복원
증분 84
DATALINK 컬럼에서의 문제점 529
복제
문제점 관별 177
비IBM 서버 173
시나리오 174
테이블 및 컬럼 이름에 대한 제한사항
175
플랜 176
DATALINK 175
update-anywhere 전제조건 176
복제 데이터베이스
작성 82
복제 분석기 177
복제된 요약 테이블 63, 109
복합 텍스트 명령문
동적 104

복합 텍스트 명령문(동적)
 변수 455
 복합 SQL, 사용 391
 부속 구성요소 통계 106
 분리 레벨
 명령문 레벨 103
 INSERT 문 471, 474
 SELECT INTO 문 475
 SELECT 문 444
 UPDATE 문 480
 분리 미리
 대기 데이터베이스 83
 분리 미리 조절
 온라인 82
 분산 데이터베이스
 갱신을 위한 설정 63
 불명확한 커서 470
 뷰
 이주 후 사용 불가능 17
 빠른 안내 525

[사]

사용자 ID
 문자 한계 66
 색인
 다중 액세스 110
 큰 키 105
 색인 키
 큰 키 105
 샘플 내용, 보기 221
 설명 정보
 DECRYPT 함수 411
 설치
 제거 가능한 드라이브 사용시 정지 7
 DB2 프로세스 중지 18
 Windows에 필요한 서비스 계정 533
 설치 요구사항
 OS/390용 Linux에서 UDB EE 및
 CEE 19
 설치 전 중지 18
 설치해제
 DB2 DFS Client Enabler 528
 설치, 설치, 이주, 실행 239
 성능 모니터 73
 소프트웨어 요구사항, ERwin 태그 언어 파일,
 Data Warehouse Center, DWC,
 Information Catalog Manager, ICM 239
 순차 77

568 릴리스 정보

순차 77 (계속)
 변경 79
 삭제 79
 작성 75
 특권 80
 호출 407
 nextval-expression 407
 prevval-expression 407
 순차 호출 407
 순차, 설명 375
 스킴라 함수
 DAYOFWEEK_ISO 395
 WEEK_ISO 395
 스키마 모델 편집기
 Data Warehouse Center 284
 스키마 이름
 문자 한계 66
 시스템 카탈로그 통계
 분산 통계 수집 105
 식별 컬럼
 변경 78
 식별 컬럼 값 리턴
 IDENTITY_VAL_LOCAL 함수 415

[아]

아카이브 및 검색
 User Exit 프로그램 98
 암호화
 데이터 81
 암호화 정보
 ENCRYPT 함수 412
 GETHINT 함수 415
 액세스 기능 533
 액세스하기 쉬운 기능 3
 양방향 언어 지원
 Windows NT의 제어 센터 215
 연합 시스템
 랩퍼 39
 별명 39
 제한사항 39
 연합 시스템, 제한사항 530
 오류 메시지
 바인드 파일이 없음 526
 유효하지 않은 단축 아이콘 216
 이주 중 7
 트리거 실행 470
 파일 액세스 거부 216
 오류 메시지 (계속)
 파티션된 데이터베이스에 노트 추가 4
 SQL10012N 220
 온라인 정보 검색
 Solaris 운영 체제 11
 요구에 따라 로그 아카이브 90
 요구에 따른 로그 아카이브 90
 원격 클라이언트 액세스
 NT용 통신 서버 구성 529
 웹 브라우저
 Windows 2000용 권장사항 11
 위성
 대화식 설치 128
 설치 고려사항 126
 응답 파일 설치 126
 DB2 Personal Edition 125
 DB2 Workgroup Edition 125
 위성 관리 센터
 버전 6, 위성 환경을 관리하기 위해 사용
 137
 위성 환경
 버전 6 위성 관리 센터를 사용하여 관리
 137
 설치 고려사항 126
 위성의 대화식 설치 128
 위성의 응답 파일 설치 126
 전제조건 125
 DB2 Personal Edition 위성 125
 DB2 Workgroup Edition 위성 125
 유니코드
 데이터베이스 및 응용프로그램 501
 문서 갱신사항 503
 함수 508
 ODBC 응용프로그램 514
 유니코드 클라이언트
 데이터 이동 한계 515
 유효하지 않은 단축 아이콘 오류 216
 응급 복구
 db2imidb 도구 사용 82
 응답 파일 설치
 위성별 키워드 126
 DB2 제어 서버 키워드 125
 이름 및 주소 정리, Trillium 247
 이름 지정 규칙 66
 문자 한계 66
 이주
 뷰의 문제 17
 오류 7

이주 (계속)
 인스턴스 갱신을 위해 db2iupdt 수행 18
 Windows 2000 17
 이중 로그 90
 인증
 DB2DOMAINLIST를 사용하는 도메인
 528
 일본어
 Linux에 로케일 설정 7
 일시중지된 입출력 기능
 연속 사용 가능성 지원 82

[자]

자국 언어 지원 71
 자동 로드 프로그램
 fork 중 정지 529
 작동 불능 트리거
 세부 설명 469
 잠금 대기 189
 잠금, 최적의 390
 저장 프로시저어
 Data Warehouse Center 534
 정렬 힙 임계값 114
 정렬 힙 크기 113
 정적 프로파일, JDBC/ODBC/CLI 응용프로그
 램 387
 제어 센터
 도구 설정 노트북 220
 동시적인 애플릿에 대한 제한사항 221
 문제점 해결 219
 버전 6, 위성 환경을 관리하기 위해 사용
 137
 변환된 버전 수행 69
 양방향 지원
 Windows NT 215
 Internet Explorer에서 오류 7
 OS/2에서 infopops 표시 문제점 220
 OS/390 플랫폼 217
 OS/390에 수정 필요 219
 UNIX 플랫폼에서 문제점 해결 219
 VM 및 VSE 데이터베이스 관리에 사용
 215
 Windows 95에서 스크립트 센터 문제점
 221
 조정자(governor) 구성 파일 113
 조회 결과, 보기 221

중국어
 Linux에 로케일 설정 7
 증분 백업 및 복구 84
 [카]
 카탈로그 뷰
 SEQUENCES 493
 캡처 및 적용
 오류 메시지 181
 응용프로그램 내에서 시작 182
 AS/400에서 178

커서
 값 동시성 390
 갱신가능 389
 동시성 389
 민감성 389
 속성, 기본값 389
 유형 389
 화면 이동 389
 문제점 해결 응용프로그램 390
 서버측 389
 정적 389
 keyset 위주(keyset-driven) 389
 OS/390 389
 keyset 위주(keyset-driven)
 갱신 390

컨테이너
 SMS 테이블 공간에 추가 78
 컬럼
 문자열 할당, 기본 규칙 505
 컬럼에 문자열 할당, 규칙 505
 코드 페이지
 로케일에 맵핑 67
 발트어 71
 CLI를 사용하는 응용프로그램 71
 클라이언트
 다운로드 17
 3층(three-tier) 환경 389

[타]

태그 언어 파일, DWC, ICM, ERwin 파일,
 ER1 파일 239
 테이블 공간
 ONLINE 상태로 설정 80
 테이블 구조 181

템플릿
 Commit.tag
 값의 예 288
 토큰 288
 ForeignKeyAdditional.tag
 값의 예 292
 토큰 291
 ForeignKey.tag
 값의 예 290
 토큰 289
 PrimaryKeyAdditional.tag
 값의 예 294
 토큰 294
 PrimaryKey.tag
 값의 예 293
 토큰 292

트리거
 오류 메시지 470
 작동 불능 469
 트리거 SQL문
 SET 변수 명령문 477
 특권
 USAGE 80

[파]

파일 액세스 거부 오류 216
 파티션 키
 갱신 63
 파티션된 데이터베이스
 노드 추가시 오류 4
 패치 레벨
 Solaris 버전 2.6 4
 표현식
 병합 연산자 508
 순차 407
 NEXTVAL 76
 PREVVAl 76
 string 508
 프로세스 모델 편집기
 Data Warehouse Center 284
 프리페처
 병렬 DMS 컨테이너 작성 및 확장 73
 피연산자
 string 508

[하]

할당

문자열, 기본 규칙 505

함수

스칼라

DECRYPT 411
ENCRYPT 412
GETHINT 415
IDENTITY_VAL_LOCAL 415
MQPUBLISH 420
MQREAD 422
MQRECEIVE 423
MQSEND 425
MQSUBSCRIBE 426
MQUNSUBSCRIBE 428
REC2XML 430

유니코드 데이터베이스 508

테이블

MQREADALL 437
MQRECEIVEALL 439

프로시저어 441

DECRYPT 81

ENCRYPT 81

GETHINT 81

A

Address Windowing Extensions

(AWE) 111

Adobe Acrobat Reader

영어 이외의 로케일에서 액세스 13

ADT 변환 388

AIX 4.3.3

fork 이후에 자동 로드 프로그램 정지
529

ALTER SEQUENCE 문

세부 설명 447

app_ctl_heap_sz 116

Archive Log 149

AS/400

캡처 및 적용 178

C

CLI

유니코드 응용프로그램 작성 508

CLI 인터페이스

코드 페이지 설정 71

CLI 저장 프로시저어

자동 바인드 391

한계 391

CLI 함수

SQLBindFileToParam 392

SQLNextResult 392

Commit.tag 템플릿

값의 예 288

토큰 288

Common Warehouse Metamodel

XML 지원 283

CREATE PROCEDURE 문

동적 복합 텍스트 명령문 455

FOR 문 484

RETURN 문 488

SIGNAL 문 490

SQL 프로시저어 명령문 482

CREATE SEQUENCE 문

세부 설명 463

CWM

XML 지원 283

D

Data Links File System

Norton Utilities와 비호환 66

Data Warehouse Center

저장 프로시저어 사용 534

iwh2exp2 명령 구문 534

OLE DB 지원 260

DataJoiner

제한사항 531

DATALINK

복원시 문제점 529

복제 175

DAYOFWEEK_ISO 스칼라 함수 395

DB2 Connect

Sun Cluster 2.2의 전제조건 64

DB2 Data Links Manager

복구 상호작용 94

DB2 Enterprise Edition

버전 6, DB2 제어 서버로 사용 129

위성으로 설정 125

전제조건 125

DB2 Personal Edition

위성으로 설정 125

DB2 Personal Edition (계속)

전제조건 125

DB2 Relational Connect

AIX, Solaris 및 Linux에 설치 41

Windows NT에 설치 40

DB2 시작

Windows 95, 98 및 ME에서 db2start사
용 185

DB2 제어 서버

버전 6, 업그레이드 129

응답 파일 설치 125

최소 요구사항 125

db2ArchiveLog 340

DB2CODEPAGE 레지스트리 변수

CLI를 사용하는 응용프로그램 71

db2expln

수행 124

db2inidb 도구 82

db2iupdt 명령, DB2 업그레이드 후 수행 18

DB2MSPCS 유틸리티, PATH 설정을 위해 머

신 재부트 64

db2start

Windows 95, 98 및 ME 185

db2가 아닌 소스, Sybase용 샘플 항목 263

DB2가 아닌 웨어하우스 소스, AIX 262

DB2가 아닌 웨어하우스 소스, Solaris 운영 환
경 262

db2_all 99

DB2_AWE 레지스트리 변수 112

DB2_INDEX_2BYTEVARLEN 78

DECLARE CURSOR 470

DECRYPT 함수

값 및 인수 411

세부 설명 411

DFS Client Enabler

설치해제 528

dlfm client_conf

실패 이유 528

dI_expint 설정 권장사항 115

DMS 컨테이너

병렬로 작성 및 확장 73

DMS 테이블 공간

병렬 작성 및 크기 조정 80

DWC

CWM XML 지원 283

OLE DB 지원 260

dynexpln

수행 124

E

enable_MQFunctions 209, 211

ENCRYPT 함수

값 및 인수 412

세부 설명 412

ERwin, 메타데이터, 추출, 가져오기 238

Extent 크기

데이터베이스 작성 마법사에서 설정 225

F

FOR statement 484

ForeignKeyAdditional.tag 템플릿

값의 예 292

토론 291

ForeignKey.tag 템플릿

값의 예 290

토론 289

G

GETHINT 함수

값 및 인수 415

세부 설명 415

GRANT

CREATE ON SEQUENCE 473

GRANT(순차 특권) 문

세부 설명 473

gui 도구

바인딩 오류 526

H

HP 및 Sun Solaris

백업 및 복원 지원 91

HP-UX

Query Enabler에 대한 지원 없음 534

HP-UX 11

메모리 창 526

I

IDENTITY 컬럼 77

IDENTITY_VAL_LOCAL 함수

값 및 인수 415

세부 설명 415

Information Catalog Manager

유틸리티 프로그램 313

히브리어 531

Internet Explorer

제어 센터에서 오류 7

IPX/SPX 프로토콜 지원, Windows 2000 17

iwh2exp2 명령

새 옵션 534

J

Java 메소드

CREATE PROCEDURE 또는 CREATE
FUNCTION 384

JDBC

화면 이동 커서, 액세스 389

JDBC/ODBC/CLI 정적 프로파일, 한계 387

JDK 1.1

설치 경로 220

L

Linux

일본어 및 중국어 로케일 설정 7

제어 센터용 환경 설정 18

locklist

maxlocks 구성 매개변수 114

M

maxappls 117

Microsoft SQL Server 데이터 소스

성능 개선 54

코드 페이지 57

환경 변수 52

DB2에 링크 53

DSN 이름 55

MERANT 라이브러리 51

MERANT 랩퍼 54

ODBC 드라이버 51

ODBC 추적 57

mon_heap_sz 117

MQ 함수 209, 211

MQPUBLISH 함수

값 및 인수 420

세부 설명 420

MQREAD 함수

값 및 인수 422

MQREAD 함수 (계속)

세부 설명 422

MQREADALL 함수

값 및 인수 437

세부 설명 437

MQRECEIVE 함수

값 및 인수 423

세부 설명 423

MQRECEIVEALL 함수

값 및 인수 439

세부 설명 439

MQSEND 함수

값 및 인수 425

세부 설명 425

MQSeries 지원 마법사 225

MQSUBSCRIBE 함수

값 및 인수 426

세부 설명 426

MQUNSUBSCRIBE 함수

값 및 인수 428

세부 설명 428

N

Named Pipe

백업 87

Netscape

오류 메시지 12, 223

온라인 정보 액세스에 사용 11

NEXTVAL 76

nextval-expression 407

NLV 지원

유니코드 501

Norton Utilities

Data Links File System과 비호환 66

NUMA-Q

Query Enabler 또는 추적 프로그램에 대한
지원 없음 534

O

ODBC

페이지 방향 390

화면 이동 커서, 액세스 389

ODBC 유니코드 응용프로그램 514

OLE DB 지원 260

OLE DB 지원 마법사 225

Open Client, DB2가 아닌 웨어하우스 소스 및 262
Oracle 데이터 소스
환경 변수 43
Linux 램퍼 43
Solaris 램퍼 43
OS/2
메시지에 필요한 JDK 레벨 362
제어 센터 설치 216

P

PREVVAL 76
prevval-expression 407
PrimaryKeyAdditional.tag 템플릿
값의 예 294
토큰 294
PrimaryKey.tag 템플릿
값의 예 293
토큰 292

Q

Query Enabler
HP-UX 및 NUMA-Q에서 지원되지 않음 534
Query Patroller
이후 이후의 사용자 특권 상실 533
Query Patroller 추적 프로그램
NUMA-Q에서 지원되지 않음 534

R

REC2XML 함수
값 및 인수 430
세부 설명 430
Red Flag Linux
중국어 로케일 7
RETURN statement 488

S

SET CONSTRAINTS 문
SET INTEGRITY로 바꿈 67
SET ENCRYPTION PASSWORD 81
SET ENCRYPTION PASSWORD 문
세부 설명 476
SET INTEGRITY 문
SET CONSTRAINTS를 바꿈 67

572 릴리스 정보

SET 변수 명령문
세부 설명 477
SIGNAL statement 490
SIGTTIN 99
SMS 테이블 공간
컨테이너 추가 78
SNA SPM
재부트 이후에 시작 검증 532
SNA 서버
호스트 또는 AS/400 응용프로그램에서
UDB 액세스 531
softmax 117
Solaris
UltraSparc 이전 버전에 대한 지원 부족 4
V2.6에 필요한 패치 레벨 4
Solaris 운영 체제
온라인 정보 검색 11
64 비트
구성 191
제한사항 191
Solaris에서의 고가용성 64
SQL 변수 455
SQL 프로시저
동적 복합 텍스트 명령문 455
변수 455
DECLARE 문 455
FOR statement 484
RETURN statement 488
SIGNAL statement 490
SQLBindFileToParam CLI 함수, 정정 392
SQLNextResult 함수 392
SQL문
ALTER SEQUENCE 447
CREATE SEQUENCE 463
GRANT(순차 특권) 473
SET ENCRYPTION PASSWORD 476
SET 변수 477
SQL, 복합 텍스트 391
stdin 99
string
표현식 508
피연산자 508
Sun Cluster 2.2
DB2 Connect 전제조건 64
Sun Solaris 및 HP
백업 및 복원 지원 91
SWITCH ONLINE 절 80

Sybase
Windows에서ICM과 비호환 8
Sybase Open Client 44
Sybase 데이터 소스 44
성능 개선 47
코드 페이지 51
환경 변수 44
DB2에 링크 46

U

USAGE 특권 80

V

Veritas
Solaris에서의 고가용성 64
VI
DB2에서 사용하도록 설정 100
VM 및 VSE용 DB2 서버
제어 센터에서 데이터베이스 관리 215

W

WEEK_ISO 스칼라 함수 395
Windows
Sybase와 ICM 사이의 비호환 8
Windows 2000
권장 웹 브라우저 11
이주 17
IPXX/SPX 프로토콜 지원 17

X

XML 지원
CWM(Common Warehouse
Metamodel)용 283

IBM 한글 지원에 관한 설문



FAX : (02) 3787-0123

보내 주시는 의견은 더 나은 고객 지원 체계를 위한 귀중한 자료가 됩니다.
 독자 여러분의 좋은 의견을 기다립니다.

책 제목: IBM® DB2® Universal Database
 릴리스 정보
 버전 7.2/버전 7.1 FixPak 3

성명		직위/담당업무	
회사명		부서명	
주소			
전화번호		팩스번호	
전자우편 주소			
사용중인 시스템	<input type="checkbox"/> 중대형 서버 <input type="checkbox"/> UNIX 서버 <input type="checkbox"/> PC 및 PC 서버		

- IBM에서 제공하는 한글 책자와 영문 책자 중 어느 것을 더 좋아하십니까? 그 이유는 무엇입니까?
 한글 책자 영문 책자
 (이유: _____)
- 본 책자와 해당 소프트웨어에서 사용된 한글 용어에 대한 귀하의 평가 점수는?
 수 우 미 양 가
- 본 책자와 해당 소프트웨어에서 번역 품질에 대한 귀하의 평가 점수는?
 수 우 미 양 가
- 본 책자의 인쇄 상태에 대한 귀하의 평가 점수는?
 수 우 미 양 가
- 한글 소프트웨어 및 책자가 지원되는 분야에 대해 귀하는 어떻게 생각하십니까?
 한글 책자를 늘려야 함 현재 수준으로 만족
 그다지 필요성을 느끼지 않음
- IBM은 인쇄물 형식(hardcopy)과 화면 형식(softcopy)의 두 종류로 책자를 제공합니다. 어느 형식을 더 좋아하십니까?
 인쇄물 형식(hardcopy) 화면 형식(softcopy) 둘 다

☞ IBM 한글 지원 서비스에 대해 기타 제안사항이 있으시면 적어주십시오.

◎ 설문에 답해 주셔서 감사합니다.
 귀하의 의견은 저희에게 매우 소중한 것이며, 고객 여러분들께 보다 좋은 제품을 제공해 드리기 위해 최선을 다하겠습니다.

IBM