

IBM® DB2® Universal Database



데이터 이동 유틸리티 안내 및 참조서

버전 7

IBM® DB2® Universal Database



데이터 이동 유틸리티 안내 및 참조서

버전 7

이 책의 정보와 지원하는 제품을 사용하기 전에 반드시 329 페이지의 『부록F. 주의사항』을 읽으십시오.

이 책에는 IBM의 특허 정보가 나와 있습니다. 이 정보는 사용권 계약하에서 제공되며, 저작권법으로 보호받습니다. 이 책에 있는 정보는 어떠한 제품도 보증하지 않으며, 이 책에 제공된 어떤 내용도 이와 같이 해석되어서는 안됩니다.

책에 대한 주문은 IBM 영업 대표나 IBM 해당 지역 사무소를 통해 하시기 바랍니다.

IBM으로 정보를 보내면, IBM은 적절한 방식으로 이를 사용하거나 배포할 수 있으며, 제공한 독자는 이에 대해 책임을 지지 않습니다.

© Copyright International Business Machines Corporation 1999, 2000. All rights reserved.

목차

이 책에 관하여	vii	버퍼된 삽입으로 가져오기 사용	34
이 책의 사용자	vii	식별 컬럼으로 가져오기 사용	34
이 책의 구성	viii	생성된 컬럼으로 가져오기 사용	36
제1장 내보내기	1	내보낸 테이블 재작성	38
내보내기 개요	2	대형 오브젝트(LOB) 가져오기	40
내보내기를 사용하는 데 필요한 특권, 권한 및		사용자 정의 구별 유형(UDT) 가져오기	40
권한 부여	3	가져오기 동안 테이블 잠금	41
내보내기 사용	3	IMPORT 명령	42
내보내기를 사용하기 전에	3	가져오기 API	51
내보내기 호출	4	SQLUIMPT-IN 데이터 구조	64
I 식별 컬럼으로 내보내기 사용	4	SQLUIMPT-OUT 데이터 구조	65
내보낸 테이블 재작성	5	파일 유형 수정자(가져오기)	67
대형 오브젝트(LOB) 내보내기	5	문자 세트 및 NLS 고려사항	77
I 병렬 처리로 데이터 내보내기	6	가져오기 세션 예	77
EXPORT 명령	9	CLP 예	77
내보내기 API	12	API 예	80
SQLUEXPT-OUT 데이터 구조	19	가져오기 성능 최적화	80
파일 유형 수정자(내보내기)	20	제한사항 및 한계점	80
분리 문자 제한사항	21	문제점 해결	81
내보내기 세션 예	23	제3장 로드	83
CLP 예	23	로드 개요	84
API 예	24	I 버전 6 및 버전 7에 소개된 이전 로드 동	
제한사항	28	작에 대한 변경사항	89
문제점 해결	28	병렬 처리 및 로딩	90
제2장 가져오기	29	로드를 사용하는 데 필요한 특권, 권한 및 권	
가져오기 개요	30	한 부여	91
가져오기를 사용하는 데 필요한 특권, 권한 및		로드 사용	92
권한 부여	31	로드를 사용하기 전에	92
가져오기 사용	32	로드 호출	93
가져오기를 사용하기 전에	32	I 식별 컬럼으로 로드 사용	94
가져오기 호출	32	I 생성된 컬럼으로 로드 사용	96
클라이언트/서버 환경에서 가져오기 사용	33	제한조건 위반에 대한 점검	98
		인터럽트된 로드 조각 재시작	100

로드 사본 위치 파일 사용	100	DB2 Data Links Manager 데이터를 이동시 키기 위해 내보내기 유틸리티 사용	204
LOAD 명령	103	DB2 Data Links Manager 데이터를 이동시 키기 위해 가져오기 유틸리티 사용	208
I LOAD QUERY 명령	119	DB2 Data Links Manager 데이터를 이동시 키기 위해 로드 유틸리티 사용	209
로드 API	121	제6장 시스템간 데이터 이동	211
데이터 구조: SQLULOAD-IN	132	플랫폼간 데이터 이동	211
데이터 구조: SQLULOAD-OUT	137	PC/IXF 파일 형식	211
db2LoadQuery - Load Query API	139	컬럼 식별자가 있는 ASCII(DEL) 파일 형 식	212
파일 유형 수정자(로드)	145	WSF 파일 형식	213
예외 테이블	159	db2move를 사용하여 데이터 이동 도구	214
덤프 파일	160	DB2 Connect를 사용한 데이터 이동	220
임시 파일 로드	160	내보내기 및 가져오기 유틸리티 사용	220
로드 유틸리티 로그 레코드	161	입력된 테이블간의 데이터 이동	222
문자 세트와 자국어 지원	161	트래버스 순서	224
로드 세션 예	162	데이터 이동중 선택	225
CLP 예	162	입력된 테이블간의 데이터 이동 예	225
API 예	170	데이터 이동을 위해 복제 사용	228
로드 조작후 보류 상태	175	구성요소에 의한 IBM 복제 도구	229
로드 성능 최적화	176	I 데이터를 이동하기 위해 Data Warehouse I Center 사용	230
제한사항 및 한계점	181	부록A. 구문 도표를 읽는 방법	233
문제점 해결	182	부록B. 가져오기 및 로드 유틸리티의 차이점	237
제4장 AutoLoader	183	부록C. 내보내기/가져오기/로드 유틸리티 파일 형식	239
AutoLoader 개요	183	분리 ASCII(DEL) 파일 형식	240
AutoLoader를 사용하는 데 필요한 특권, 권 한 및 권한 부여	185	샘플 DEL 파일	242
AutoLoader 사용	185	DEL 데이터 유형 설명	243
AutoLoader를 사용하기 전에	185	컬럼 식별자가 없는 ASCII(ASC) 파일 형식	246
AutoLoader 호출	185	샘플 ASC 파일	246
다중 데이터베이스 파티션으로 로드	186	ASC 데이터 유형 설명	248
AutoLoader 옵션	187	IXF 파일 형식의 PC 버전	250
AutoLoader 세션 예	195	PC/IXF 레코드 유형	252
이주 및 백 레플 호환성	198		
AutoLoader 힌트 및 추가 정보	199		
제한사항 및 한계점	200		
AutoLoader 문제점 해결	201		
제5장 DB2 Data Links Manager 데이터 이동	203		

PC/IXF 데이터 유형.	275	PDF 책 인쇄	319
PC/IXF 데이터 유형 설명.	282	인쇄된 책 주문	319
데이터베이스에 PC/IXF 파일 가져오기를 관리하는 일반 규칙	288	DB2 온라인 문서.	321
데이터베이스로 PC/IXF 파일 가져오기를 관리하는 데이터 유형 특정 규칙.	290	온라인 도움말 액세스	321
FORCEIN 옵션	294	정보 온라인 보기.	323
PC/IXF와 버전 0 System/370 IxF간의 차이점	303	DB2 마법사 사용.	325
워크시트 파일 형식(WSF)	304	문서 서버 설정	326
부록D. 경고, 오류 및 완료 메시지	307	정보 온라인 검색.	327
부록E. DB2 라이브러리 사용.	309	부록F. 주의사항	329
DB2 PDF 파일 및 인쇄된 책	309	등록상표.	332
DB2 정보	309	색인	335
		IBM에 문의	341
		제품 정보	341

이 책에 관하여

이 책에서는 다음과 같은 IBM DB2 Universal Database(UDB) 데이터 이동 유틸리티에 관한 정보를 제공하며, 이를 사용하는 방법을 보여줍니다.

- 가져오기 및 내보내기 유틸리티는 테이블이나 뷰 및 다른 데이터베이스나 스프레드시트 프로그램간, DB2 데이터베이스간, DB2 데이터베이스와 DB2 Connect를 사용하는 호스트 데이터베이스간의 데이터를 이동시킵니다. 내보내기 유틸리티가 데이터베이스로부터 운영 체제 파일로 데이터를 이동시키므로, 사용자가 이 파일을 사용하여 다른 데이터베이스로 데이터를 가져오거나 로드할 수 있습니다.
- 로드 유틸리티는 데이터를 테이블로 이동시켜 이미 있는 색인을 확장하고 통계를 작성합니다. 로드 유틸리티는 대량의 데이터가 관련될 경우, 가져오기 유틸리티보다 더 빠르게 데이터를 이동시킵니다. 내보내기 유틸리티를 사용하여 로드 해제된 데이터는 로드 유틸리티를 사용하여 로드될 수 있습니다.
- AutoLoader 유틸리티는 많은 양의 데이터를 파티션하여 이 파티션 데이터를 파티션된 데이터베이스의 다른 파티션에 로드합니다.
- DataPropagator(DPROP)는 다른 DB2 관계형 데이터베이스에 있는 다른 테이블에 대한 테이블 갱신의 자동 복사를 허용하는 DB2 Universal Database의 구성요소입니다.
- DWC(Data Warehouse Center)는 조작 데이터베이스에서 웨어하우스 데이터베이스로 데이터를 이동하는 데 사용할 수 있습니다.

데이터베이스 안팎으로 데이터를 이동시키는 기타 벤더의 제품도 사용 가능하지만, 이 책에서는 언급되지 않습니다.

이 책의 사용자

이 안내서는 데이터베이스 관리자, 응용프로그램 프로그래머 및 다음과 같은 타스크를 수행하는 기타 DB2 UDB 사용자를 위한 책입니다.

- 운영 체제 파일로부터 DB2 테이블로 데이터를 로드함

- DB2 데이터베이스간, DB와 다른 응용프로그램(예: 스프레드시트)간에 데이터를 이동시킴
- 데이터를 아카이브함

사용자가 DB2 Universal Database, SQL(Structured Query Language) 및 DB2 UDB가 수행하고 있는 운영 체제 환경을 잘 알고 있다고 가정합니다. DB2 UDB에 대한 일반적인 정보는 *관리 안내서*에서 자세한 내용을 참조하십시오. SQL에 대해서는 *SQL 참조서*에서 자세한 내용을 참조하십시오. DB2 UDB 명령행 처리기 구성, 호출 및 사용에 관한 정보는 *Command Reference*에서 자세한 내용을 참조하십시오. DB2 UDB API에 대해서는 *Administrative API Reference*에서 자세한 내용을 참조하십시오. 응용프로그램 빌드 안내서에서 DB2 관리 API를 포함하는 응용프로그램 작성에 관한 일반적인 정보를 참조하십시오. 이 안내서는 사용자의 운영 체제에 따른 DB2 설치에 대한 설명은 포함하지 않습니다. 설치 정보는 사용자의 운영 체제에 대한 해당 빠른 시작 책에 들어 있습니다.

이 책의 구성

다음과 같은 내용이 들어 있습니다.

『제1장 내보내기』

DB2 테이블에서 파일로 데이터를 이동시키는 데 사용되는 DB2 내보내기 유틸리티를 설명합니다.

『제2장 가져오기』

파일에서 DB2 테이블이나 뷰로 데이터를 이동시키는 데 사용되는 DB2 가져오기 유틸리티를 설명합니다.

『제3장 로드』

파일에서 DB2 테이블로 많은 양의 데이터를 이동시키는 데 사용되는 DB2 로드 유틸리티를 설명합니다.

『제4장 AutoLoader』

많은 양의 데이터를 파티션하여 파티션된 데이터베이스의 다른 파티션에 파티션 데이터를 로드하는 AutoLoader 유틸리티를 설명합니다.

『제5장 DB2 Data Links Manager 데이터 이동』

DB2 파일 관리 프로그램 데이터를 이동시키기 위해 DB2 내보내기, 가져오기 및 로드 유틸리티를 사용하는 방법을 설명합니다.

『제6장 시스템간 데이터 이동』

플랫폼간 및 DRDA 호스트 데이터베이스간에 데이터를 전송하기 위해 DB2 내보내기, 가져오기 및 로드 유틸리티 사용법에 대해 설명합니다. 엔터프라이즈의 데이터베이스간에 데이터를 이동시키기 위한 또다른 방법인 DataPropagator(DPRO)도 설명됩니다. DWC(Data Warehouse Center)에 대해서도 설명하는데 이는 운영중인 데이터베이스에서 웨어하우스 데이터베이스로 데이터를 이동하는 데 사용할 수 있습니다.

『부록A. 구문 도표를 읽는 방법』

구문 도표에 사용된 규칙을 설명합니다.

『부록B. 가져오기 및 로드 유틸리티의 차이점』

DB2 로드 및 가져오기 유틸리티간의 중요한 차이점을 요약합니다.

『부록C. 내보내기/가져오기/로드 유틸리티 파일 형식』

데이터베이스 관리 프로그램 내보내기, 가져오기 및 로드 유틸리티가 지원하는 외부 파일 형식을 설명합니다.

『부록D. 경고, 오류 및 완료 메시지』

경고 또는 오류 조건 검출시 데이터베이스 관리 프로그램에 의해 생성된 메시지 해석에 대한 정보를 제공합니다.

제1장 내보내기

여기에서는 DB2 UDB 내보내기 유틸리티를 설명하며, 이 유틸리티는 DB2 데이터베이스의 데이터를 데이터베이스 밖에 저장된 하나 이상의 파일에 데이터를 작성하는 데 사용됩니다. 내보낸 데이터는 DB2 가져오기 또는 DB2 로드 유틸리티를 사용하여 다른 DB2 데이터베이스로 가져오거나 로드될 수 있거나 다른 응용 프로그램(예: 스프레드시트)으로 가져올 수 있습니다.

다음과 같은 내용이 들어 있습니다.

- 2 페이지의 『내보내기 개요』
- 3 페이지의 『내보내기를 사용하는 데 필요한 특권, 권한 및 권한 부여』
- 3 페이지의 『내보내기 사용』
- 4 페이지의 『식별 컬럼으로 내보내기 사용』
- 5 페이지의 『내보낸 테이블 재작성』
- 5 페이지의 『대형 오브젝트(LOB) 내보내기』
- 6 페이지의 『병렬 처리로 데이터 내보내기』
- 9 페이지의 『EXPORT 명령』
- 12 페이지의 『내보내기 API』
- 19 페이지의 『SQLUEXT-OUT 데이터 구조』
- 20 페이지의 『파일 유형 수정자(내보내기)』
- 23 페이지의 『내보내기 세션 예』
- 28 페이지의 『제한사항』
- 28 페이지의 『문제점 해결』.

204 페이지의 『DB2 Data Links Manager 데이터를 이동시키기 위해 내보내기 유틸리티 사용』에서 DB2 Data Links Manager 데이터 내보내기에 관한 자세한 정보를 참조하십시오. 222 페이지의 『입력된 테이블간의 데이터 이동』에서 입력된 테이블로부터 데이터 내보내기에 관한 정보를 참조하십시오. 220 페이지의 『DB2

Connect를 사용한 데이터 이동』에서 DRDA 서버 데이터베이스로부터 DB2 Connect 워크스테이션에 있는 파일로 데이터 내보내기 및 이와 반대 방향으로 내보내는 내용을 참조하십시오.

내보내기 개요

내보내기 유틸리티는 데이터베이스로부터 여러 외부 파일 형식 중 하나로 된 운영 체제 파일로 데이터를 내보냅니다.

다음은 데이터를 내보낼 때 필요한 내용입니다.

- 내보낼 데이터를 지정하는 SQL SELECT문
- 내보낸 데이터를 저장할 운영 체제 파일의 경로와 이름
- 입력 파일에 있는 데이터의 형식. 이 형식은 IXF, WSF 또는 DEL이 될 수 있습니다. 239 페이지의 『부록C. 내보내기/가져오기/로드 유틸리티 파일 형식』에서 자세한 내용을 참조하십시오.
- 메시지 파일 이름
- 입력된 테이블을 내보낼 때 계층 구조에 서브테이블 트래버스 순서를 제공할 필요가 있습니다. IXF 형식이 사용될 경우, 기본 순서를 사용하도록 권장됩니다. 순서를 지정할 때, 서브테이블이 PRE-ORDER 형식으로 트래버스되어야 하는 것을 명심하십시오. 입력된 테이블을 내보낼 때 SELECT문을 직접 제공할 수 없습니다. 대신, 목표 서브테이블 이름과 선택적으로 WHERE절을 지정해야 합니다. 내보내기 유틸리티는 필요한 SELECT문을 생성하고 실행시키기 위해 트래버스 순서에 따라 이 정보를 사용합니다. 222 페이지의 『입력된 테이블간의 데이터 이동』에서 자세한 내용을 참조하십시오.

다음과 같이 지정할 수도 있습니다.

- IXF 또는 WSF 파일로 내보낼 때 새로운 컬럼 이름. 새로운 컬럼 이름을 지정하지 않을 경우, 이미 있는 테이블이나 뷰의 컬럼 이름이 내보낸 파일에서 사용됩니다.
- 내보내기 조작을 사용자가 정의하기 위한 추가 옵션(20 페이지의 『파일 유형 수정자(내보내기)』 참조)

다중 데이터베이스 파티션 환경에서 내보내기 유틸리티를 사용하려 할 경우에는 **db2batch**를 사용하여 각 데이터베이스 파티션에서 타스크를 완료할 수 있습니다. SELECT문은 지역적으로 발견된 데이터만 리턴할 수 있어야 합니다. 선택 조건은 다음과 같습니다.

```
SELECT * FROM tablename WHERE NODENUMBER(column-name) = CURRENT NODE
```

Command Reference 또는 *관리 안내서*에서 **db2batch**에 관한 자세한 정보를 참조하십시오.

내보내기를 사용하는 데 필요한 특권, 권한 및 권한 부여

특권은 사용자가 데이터베이스 자원을 작성하거나 액세스할 수 있게 합니다. 권한 레벨은 특권과 상위 레벨 데이터베이스 관리 프로그램 유지보수 및 유틸리티 조작을 그룹으로 분류하는 방법을 제공합니다. 이는 모두 데이터베이스 관리 프로그램과 데이터베이스 오브젝트에 대한 액세스를 제어하기 위한 것입니다. 사용자는 적합한 권한 즉, 필요한 특권이나 권한을 가지고 있는 오브젝트만 액세스할 수 있습니다.

내보내기 조작에 참여하는 각 테이블에 대한 SYSADM이나 DBADM 권한, 또는 CONTROL이나 SELECT 특권이 있어야 합니다.

내보내기 사용

내보내기를 사용하기 전에

내보내기 유틸리티를 호출하기 전에 데이터가 내보낼 데이터베이스로 연결되어야 합니다(또는 내재적으로 연결할 수 있어야 합니다). 유틸리티가 COMMIT문을 발행하므로, 내보내기 유틸리티를 호출하기 전에 COMMIT 또는 ROLLBACK을 수행하여 모든 트랜잭션을 완료하고 모든 잠금을 해제해야 합니다. 다른 연결을 사용하여 테이블을 액세스하는 다른 사용자 응용프로그램은 연결해제할 필요가 없습니다.

내보내기 호출

내보내기 유틸리티는 다음을 통해 호출될 수 있습니다.

- 명령행 처리기(CLP)

다음은 CLP를 통해 발행된 내보내기 명령의 예입니다.

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

- 제어 센터에 있는 내보내기 노트북. 내보내기 노트북을 열려면 다음을 수행하십시오.

1. 제어 센터에서, 테이블 또는 뷰 폴더가 나올 때까지 오브젝트 트리를 확장하십시오.
2. 작업하려는 폴더를 클릭하십시오. 이미 있는 테이블이나 뷰가 창의 오른쪽 파티션 영역에 표시됩니다(내용 영역).
3. 내용 영역에서 원하는 테이블이나 뷰를 오른쪽 마우스 버튼으로 클릭한 후, 팝업 메뉴에서 내보내기를 선택하십시오. 내보내기 노트북이 열립니다.

관리 안내서에서 제어 센터에 관한 일반적인 정보를 참조하십시오. 자세한 정보는 제어 센터의 온라인 도움말 기능을 통해 제공됩니다.

- API, **sqluexpr**. 12 페이지의 『내보내기 API』에서 이 API에 관한 정보를 참조하십시오. 응용프로그램 빌드 안내서에서 DB2 관리 API를 포함하는 응용프로그램 작성에 관한 일반적인 정보를 참조하십시오.

식별 컬럼으로 내보내기 사용

내보내기 유틸리티는 식별 컬럼이 들어 있는 테이블에서 데이터를 내보내는 데 사용할 수 있습니다. 내보내기 조작에 지정된 SELECT문이 "select * from tablename" 양식으로 되어 있고 METHOD 옵션을 사용하지 않은 경우에는, IXF 파일로 식별 컬럼 등록 정보를 내보내는 기능이 지원됩니다. IMPORT 명령의 REPLACE_CREATE 및 CREATE 옵션은 그 해당 식별 컬럼 등록 정보를 포함하여 테이블을 재작성하는 데 사용될 수 있습니다. 유형 GENERATED ALWAYS의 식별 컬럼이 들어 있는 테이블에서 IPX 파일이 작성된 경우에, 데이터 파일을 정상적으로 가져올 수 있는 유일한 방법은 identityignore 식별자를 지정하는 것입니다. 그렇지 않은 경우에는 모든 행들이 거부됩니다(SQL3550W).

내보낸 테이블 재작성

테이블은 내보내기 유틸리티를 사용하고 IXF 파일 형식을 지정하여 저장될 수 있습니다. 그런 다음, 저장된 테이블(색인 포함)은 가져오기 유틸리티를 사용하여 재작성될 수 있습니다(IPX 파일 형식이 테이블에 대해 실행하는 속성 목록을 포함하여, 38 페이지의 『내보낸 테이블 재작성』에서 자세한 내용을 참조하십시오).

내보낼 데이터가 내보내진 파일이 작성될 파일 시스템에서 사용할 수 있는 공간을 초과할 경우, 내보내기 작업은 실패합니다. 이러한 경우, 내보내기 파일이 목표 파일 시스템에 알맞도록 WHERE절에 조건을 지정하여 선택되는 데이터의 양을 제한해야 합니다. 모든 데이터를 내보내기 위해 내보내기 유틸리티를 여러 번 호출할 수 있습니다.

DEL 및 ASC 파일 형식에 목표 테이블의 설명은 들어 있지 않지만 레코드 데이터는 들어 있습니다. 데이터가 이러한 파일 형식으로 저장된 테이블을 재작성하려면 목표 테이블을 작성한 후 로드, AutoLoader 또는 가져오기 유틸리티를 사용하여 이 파일에서 테이블을 상주시키십시오. **db2look**(DB2 Statistics 및 DDL Extraction Tool; *Command Reference* 참조)은 원래 테이블 정의를 캡처하여 해당 DDL을 생성하는 데 사용할 수 있습니다.

대형 오브젝트(LOB) 내보내기

대형 오브젝트(LOB) 컬럼으로부터 데이터를 내보낼 때, 기본 조치는 데이터의 첫 번째 32비트를 선택하여 이 데이터를 컬럼 데이터의 나머지와 같은 파일에 배치하는 것입니다. `lobsinfile` 수정자(20 페이지의 『파일 유형 수정자(내보내기)』 참조)가 지정된 경우, 내보내기 유틸리티는 전체 LOB(2GB까지)를 선택하여 다른 파일에 배치합니다.

주: IXF 파일 형식은 LOB 컬럼이 로그되는지의 여부와 같이 컬럼의 LOB 옵션을 저장하지 않습니다. 이는 가져오기 유틸리티가 1GB 이상으로 정의된 LOB 컬럼이 들어 있는 테이블을 재작성할 수 없음을 의미합니다.

병렬 처리로 데이터 내보내기

데이터를 병렬로 내보내기를 하면 그렇지 않은 경우 보다 훨씬 효과적인 방법으로 노드들을 가로질러 형식화된 출력을 생성할 뿐만 아니라 데이터 전송도 줄고 결과 세트의 기록도 분산됩니다. 데이터를 병렬로 내보낼 때(테이블의 각 파티션에 대해 하나씩, 다중 내보내기 조작을 호출하여), 데이터가 발췌되어 지역 노드로 변환된 후 지역 파일 시스템에 기록됩니다. 반대로, 데이터를 직렬로 내보낼 때는(단일 내보내기 조작을 통해 내보낼 때), 데이터가 병렬로 발췌된 후 클라이언트로 보내지면 그 곳에서 단일 프로세스가 변환을 수행하고 결과 세트를 지역 파일 시스템에 기록합니다.

db2batch 명령은 SQL문의 성능 특성 및 실행 지속시간을 모니터링하는 데 사용됩니다. 이 유틸리티는 다음과 같은 파티션된 데이터베이스 환경에서 병렬 내보내기 기능도 가집니다.

- 내보내질 데이터를 정의하기 위해 조회를 수행하는 환경
- 각 파티션에서 그 파티션에 상주하는 내보내진 데이터가 들어 있는 파일을 작성하는 환경

병렬 내보내기는 어떤 테이블들이 조회되고 그 테이블들이 파티션된 데이터베이스 환경 내에서 어느 곳에 존재하는가에 따라 두 가지 유형으로 나뉩니다.

- 단일 파티션 테이블로부터 병렬 내보내기 또는 나란히 배열된 다중 테이블에서의 조인 또는 부속 조회(**db2batch** 명령에서 `-p s`를 지정하십시오).
테이블은 두 가지 면에서 나란히 배열된 것으로 간주될 수 있습니다.
 - 테이블이 동일한 파티션에 정의된 단일 파티션 노드 그룹 내에 있습니다.
 - 테이블이 동일한 노드 그룹에 있으며 동일한 수와 동일한 유형의 컬럼이 있는 파티션 키를 가지고 있습니다. 파티션 키의 해당 컬럼들은 파티션 호환가능하며 테이블들은 전체 파티션 키 또는 파티션 키의 `sueprset`에 대해 `equi-joined`됩니다.

각각의 경우에 아래에 설명된대로 `NODENUMBER` 함수를 사용하여 해당 파티션의 내보내기 데이터 파일을 생성하기 위해 각 파티션에서 조회를 수행할 수 있습니다. 테이블이 단일 파티션에만 존재하면 데이터가 하나의 파티션에서만 검

색되기 때문에 내보내기 병렬 처리가 취소된다는 점에 유의하십시오. 이러한 경우에 내보내기 병렬 처리를 사용하려면 아래의 내용을 참조하십시오.

- 나란히 배열되지 않은 여러 테이블로부터 **SELECT(db2batch** 명령에 `-p t tablename` 또는 `-p d`를 지정하십시오. 앞의 옵션은 올림 테이블로 사용하도록 기존 테이블의 이름을 지정하는 반면에, 뒤의 옵션은 내보내기 유틸리티가 올림 테이블을 작성하도록 합니다).

내보내기 유틸리티는 내보내기 조회를 통해 데이터가 상주되는 올림 테이블을 사용합니다. 이 올림 테이블은 올림 테이블로 `fullselect` 조회의 `INSERT`를 수행함으로써 "내보내기(`export`)" 결과 세트의 행들을 위치시키는 데 사용됩니다. 일단 올림 테이블이 작성되면, 내보내기 유틸리티는 올림 테이블에 대해 다음을 수행하여 각 파티션에서 내보내기 데이터 파일을 생성합니다.

```
"select * WHERE NODENUMBER(colname) = CURRENT NODE"
```

올림 테이블은 또한 단일 파티션 테이블을 병렬로 내보내는 데 사용될 수 있습니다. 대부분의 경우에 단일 파티션에서 다중 파티션 올림 테이블로 데이터를 전송한 후 모든 파티션에서 올림 테이블을 병렬로 내보내면 단일 파티션 테이블을 직렬로 내보내는 것보다 빠릅니다.

내보내기 유틸리티는 각 파티션에서 병렬로 조회를 수행하여 해당 파티션에서 데이터를 검색합니다. `db2batch -p s` 경우에 원래의 `SELECT` 조회는 병렬로 수행됩니다. `db2batch -p t` 및 `db2batch -p d` 경우에 올림 테이블은 지정된 조회를 사용하여 내보내기 데이터와 함께 로드되며 `SELECT *` 조회는 데이터를 내보내기 위해 각 파티션에서 병렬로 올림 테이블에 대해 수행됩니다. 주어진 파티션에 상주하는 데이터만을 내보내기 위해 **db2batch**가 해당 파티션에서 수행되는 조회의 `WHERE`절에 `NODENUMBER(colname) = CURRENT NODE`를 추가합니다. `colname` 매개변수는 테이블 컬럼의 규정화 또는 비규정화 이름으로 설정되어야 합니다. 내보내기 유틸리티는 원래 조회에 첫번째 컬럼 이름을 사용하여 이 매개변수를 설정합니다.

다음은 내보내기 유틸리티에서 사용하는 조회에 대한 제한사항입니다.

- `db2batch -p s`를 지정할 때, `NODENUMBER colname`에 대해 컬럼 이름이 필요하기 때문에 조회에 컬럼 함수만 들어 있으면 안됩니다.

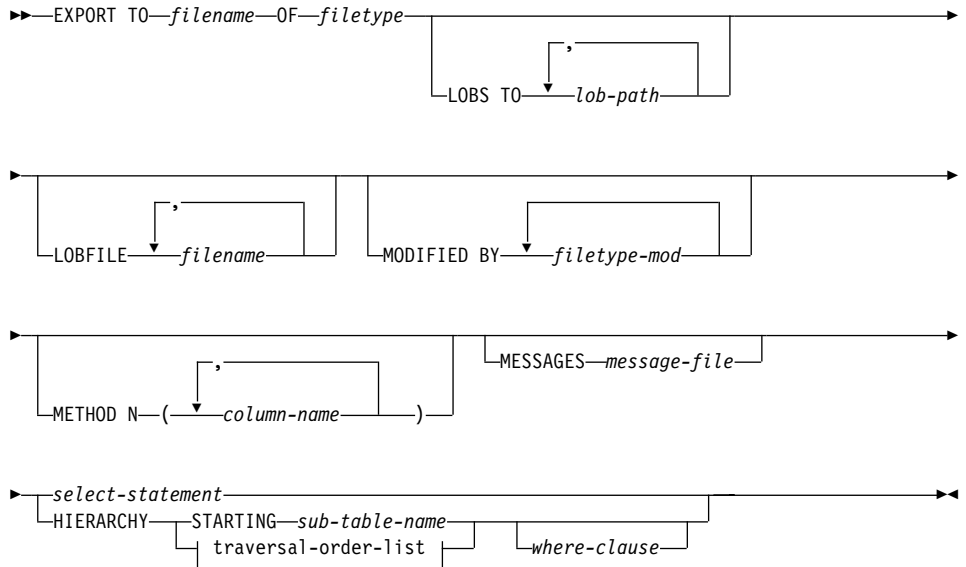
병렬 처리로 데이터 내보내기

- `db2batch -p s`를 지정할 때 총계(예: 최소, 최대 및 평균)는 파티션 키가 포함된 그룹화에 따라 달라야 합니다.
- `db2batch -p t` 또는 `db2batch -p d`를 지정할 때, 조회에 `ORDER BY`를 포함할 수 없는데 이는 DB2 UDB가 `INSERT`문 내의 `fullselect`에서 `ORDER BY`를 지원하지 않기 때문입니다.

db2batch 명령에 `-p s` 지정할 때 사용자가 `-r` 옵션을 사용하여 결과 출력 파일을 작성할 경우, `ORDER BY` 절이 있으면 각 파티션상의 파일들이 정렬된 순서로 있게 됩니다. 단일 저장 파일을 원한다면, 각 파티션상의 정렬 파일을 하나의 정렬 파일로 병합하십시오. 예를 들어, UNIX 기반 시스템에서 명령 `sort -m`을 사용하여 파일들을 단일 정렬 파일로 병합하십시오. NFS 마운트된 파일 시스템으로 출력을 송신할 경우에는 사용자가 `ORDER BY` 절을 지정해도 출력이 정렬되지 않습니다.

EXPORT 명령

명령 구문



traversal-order-list:



명령 매개변수

HIERARCHY traversal-order-list

지정된 트래버스 순서로 부속 계층 구조를 내보냅니다. 모든 서브테이블이 PRE-ORDER 형식으로 나열되어야 합니다. 첫번째 서브테이블 이름이 SELECT문에 대한 목표 테이블 이름으로 사용됩니다.

HIERARCHY STARTING sub-table-name

기본 트래버스 순서를 사용하여(ASC, DEL, WSF 파일의 경우, OUTER 순서 또는 PC/IXF 데이터 파일에 저장된 순서), sub-table-name에서 시작하는 부속 계층 구조를 내보냅니다.

LOBFILE filename

LOB 파일에 대한 하나 이상의 기본 파일 이름을 지정합니다. 첫번째 이름에 대한 이름 공간이 모두 사용되면 두번째 이름이 사용되고, 계속 그 다음 이름이 사용됩니다.

내보내기 조작중 LOB 파일을 작성할 때, 파일 이름은 이 목록의 현재 기본 이름을 현재 경로(*lob-path*로부터)에 추가하고, 3자릿수의 순차 번호를 추가한 다음 구성됩니다. 예를 들어, 현재 LOB 경로가 /u/foo/lob/path 디렉토리이고, LOB 파일 이름이 bar이면, 작성된 LOB 파일은 /u/foo/lob/path/bar.001, /u/foo/lob/path/bar.002 등이 됩니다.

LOBS TO lob-path

LOB 파일이 저장될 디렉토리에 대해 하나 이상의 경로를 지정합니다. 첫 번째 경로에서 파일 공간이 모두 사용되면 두번째 경로가 사용되고, 계속 그 다음 경로가 사용됩니다.

MESSAGES message-file

내보내기 조작중 발생하는 경고 및 오류 메시지의 목적지를 지정합니다. 파일이 이미 있으면, 내보내기 유틸리티가 정보를 첨부합니다. *message-file* 이 생략되면, 메시지는 표준 출력에 작성됩니다.

METHOD N column-name

출력 파일에 사용할 하나 이상의 컬럼 이름을 지정합니다. 이 매개변수가 지정되어 있지 않으면, 테이블에 있는 컬럼 이름이 사용됩니다. 이 매개변수는 WSF 및 IXF 파일에만 유효하며, 계층 구조적 데이터를 내보낼 때에는 유효하지 않습니다.

MODIFIED BY filetype-mod

추가 옵션을 지정합니다(20 페이지의 표2 참조).

OF filetype

출력 파일의 데이터 형식을 다음과 같이 지정합니다.

- 다양한 데이터베이스 관리 프로그램 및 파일 관리 프로그램에서 사용되는 DEL(컬럼 식별자가 있는 ASCII 형식)
- 다음과 같은 프로그램에서 사용되는 WSF(워크시트 형식)
 - Lotus 1-2-3

- Lotus Symphony

주: BIGINT 또는 DECIMAL 데이터를 내보낼 때 DOUBLE 유형의 범위에 해당하는 값만 정확하게 내보낼 수 있습니다. 이 범위에 해당하지 않는 값도 내보내기는 하지만, 이러한 값을 다시 가져오거나 로드할 때 운영 체제에 따라 부정확한 데이터를 만드는 결과를 낼 수 있습니다.

- SELECT문에 컬럼이 지정되어 있는 경우를 제외하고, 이미 있는 모든 색인뿐 아니라 대부분의 테이블 속성이 IXF 파일에 저장되어 있는 IXF(Integrated eXchange Format, PC 버전). 다른 파일 형식의 경우 데이터가 테이블로 가져오기 위해서는 테이블이 먼저 있어야 하는 반면, 이 형식의 경우에는 테이블이 재작성될 수 있습니다. IXF 파일 형식이 테이블에 대해 실행하는 속성 목록에 대해서는 38 페이지의 『내보낸 테이블 재작성』에서 자세한 내용을 참조하십시오.

이 파일의 형식에 대해서는 데이터 이동 유틸리티 안내 및 참조서에서 239 페이지의 『부록C. 내보내기/가져오기/로드 유틸리티 파일 형식』에서 자세한 내용을 참조하십시오.

select-statement

내보낼 데이터를 리턴할 SELECT문을 지정합니다. SELECT문이 오류를 일으키면, 메시지가 메시지 파일(또는 표준 출력)에 작성됩니다. 오류 코드가 SQL0012W, SQL0347W, SQL0360W, SQL0437W 또는 SQL1824W 중 하나일 경우 내보내기 조작이 계속되고, 그렇지 않으면 조작이 중단됩니다.

TO filename

데이터가 내보낼 파일 이름을 지정합니다. 이 파일에 대한 완전 경로가 지정되어 있지 않을 경우, 내보내기 유틸리티는 현재 디렉토리와 기본 드라이브를 목적지로 사용합니다.

이미 있는 파일 이름이 지정될 경우, 내보내기 유틸리티는 파일의 내용을 겹쳐쓰고 정보를 첨부하지 않습니다.

내보내기 API

C API 구문

```
/* File: sqlutil.h */
/* API: Export */
/* ... */
SQL_API_RC SQL_API_FN
sqluexpr (
    char * pDataFileName,
    sqlu_media_list * pLobPathList,
    sqlu_media_list * pLobFileList,
    struct sqldcol * pDataDescriptor,
    struct sqlchar * pActionString,
    char * pFileType,
    struct sqlchar * pFileTypeMod,
    char * pMsgFileName,
    short CallerAction,
    struct sqluexprt_out* pOutputInfo,
    void * pReserved,
    struct sqlca * pSqlca);
/* ... */
```


일반 API 구문

```

/* File: sqlutil.h */
/* API: Export */
/* ... */
SQL_API_RC SQL_API_FN
sqlgexpr (
    unsigned short DataFileNameLen,
    unsigned short FileTypeLen,
    unsigned short MsgFileNameLen,
    char * pDataFileName,
    sqlu_media_list * pLobPathList,
    sqlu_media_list * pLobFileList,
    struct sqldcol * pDataDescriptor,
    struct sqlchar * pActionString,
    char * pFileType,
    struct sqlchar * pFileTypeMod,
    char * pMsgFileName,
    short CallerAction,
    struct sqluexpt_out* pOutputInfo,
    void * pReserved,
    struct sqlca * pSqlca);
/* ... */

```

API 매개변수

DataFileNameLen

입력. 데이터 파일 이름의 길이(바이트)를 나타내는 부호가 없는 2바이트 정수

FileTypeLen

입력. 파일 유형의 길이(바이트)를 나타내는 부호가 없는 2바이트 정수

MsgFileNameLen

입력. 메시지 파일 이름의 길이(바이트)를 나타내는 부호가 없는 2바이트 정수

pDataFileName

입력. 데이터가 내보내질 외부 파일 이름과 경로를 포함하는 문자열

pLobPathList

입력. *media_type* `SQLU_LOCAL_MEDIA`를 사용하는 *sqlu_media_list* 및 LOB 파일이 저장될 클라이언트의 경로를 나열하는 *sqlu_media_entry* 구조

이 목록의 첫번째 경로에서 파일 공간이 모두 사용되면, API는 두번째 경로를 사용하고 다음 경로를 계속해서 사용합니다.

*Administrative API Reference*에 있는 “`SQLU-MEDIA-LIST`”에서 자세한 내용을 참조하십시오.

pLobFileList

입력. *media_type* `SQLU_CLIENT_LOCATION`을 사용하는 *sqlu_media_list* 및 기본 파일 이름을 포함하는 *sqlu_location_entry* 구조

이 목록의 첫번째 이름을 사용하여 이름 공간이 모두 사용되면 API는 두번째 이름을 사용하고, 계속 다음 이름을 사용합니다.

*Administrative API Reference*에 있는 “`SQLU-MEDIA-LIST`”에서 자세한 내용을 참조하십시오.

내보내기 조작중 LOB 파일을 작성할 때, 파일 이름은 이 목록의 현재 기본 이름을 현재 경로(*pLobFilePath*로부터)에 추가하고, 3자릿수의 순차 번호를 추가한 다음 구성됩니다. 예를 들어, 현재 LOB 경로가 `/u/foo/lob/path` 디렉토리이고, 현재 LOB 파일 이름이 `bar`이면, 작성된 LOB 파일은 `/u/foo/lob/path/bar.001`, `/u/foo/lob/pah/bar.002` 등이 됩니다.

pDataDescriptor

입력. 출력 파일에 대한 컬럼 이름을 지정하는 *sqldcol* 구조에 대한 포인터. *dcolmeth* 필드의 값은 이 매개변수에 제공된 정보의 나머지가 내보내기 유틸리티에 의해 해석되는 방법을 결정합니다. 이 매개변수에 대해 유효한 값(*sqlutil*에 정의됨)은 다음과 같습니다.

SQL_METH_N

이름 출력 파일에 사용될 컬럼 이름을 지정합니다.

SQL_METH_D

기본값. 테이블의 기존 컬럼 이름이 출력 파일에 사용됩니다. 이

러한 경우, 컬럼의 수와 컬럼 스펙 배열은 모두 무시됩니다. 컬럼 이름은 *pActionString*에 지정된 SELECT문의 출력으로부터 파생됩니다.

*Administrative API Reference*에 있는 “SQLDCOL”에서 자세한 내용을 참조하십시오.

pActionString

입력. 유효한 동적 SQL SELECT문을 포함하는 *sqlchar* 구조에 대한 포인터. 이 구조에는 SELECT문을 이루는 문자 앞에 2바이트 길이의 필드가 들어 있습니다. SELECT문은 데이터베이스로부터 발췌되어 외부 파일에 기록되는 데이터를 지정합니다.

외부 파일(*pDataDescriptor*로부터)의 컬럼과 SELECT문의 데이터베이스 컬럼은 각각의 목록/구조 위치가 일치합니다. 데이터베이스에서 선택된 데이터의 첫번째 컬럼은 외부 파일의 첫번째 컬럼에 놓여지며, 컬럼 이름은 외부 컬럼 배열의 첫번째 요소에서 가져옵니다.

*Administrative API Reference*에 있는 “SQLCHAR”에서 자세한 내용을 참조하십시오.

주: 입력된 테이블에 사용될 구문은 9 페이지의 『EXPORT 명령』에 설명되어 있습니다.

pFileType

입력. 외부 파일 내의 데이터 형식을 나타내는 문자열. 지원되는 외부 파일 형식(*sqlutil*에 정의됨)은 다음과 같습니다.

SQL_DEL

dBase, BASIC 및 IBM Personal Decision Series 프로그램과 기타 여러 데이터베이스 관리 프로그램 및 파일 관리 프로그램과의 교환을 위한 컬럼 식별자가 있는 ASCII

SQL_WSF

Lotus Symphony 및 1-2-3 프로그램과 교환을 위한 워크시트 형식

SQL_IXF

IXF(Integrated eXchange Format)의 PC 버전. 테이블로부터 데이터 내보내기를 위해 선호되는 방법입니다. 이 파일 형식에 내보낸 데이터는 나중에 같은 테이블 또는 다른 데이터베이스 관리 프로그램 테이블로 가져오거나 로드될 수 있습니다.

pFileTypeMod

입력. 하나 이상의 처리 옵션을 지정하는 문자 배열 앞에 오는 2바이트 길이의 필드가 들어 있는 *sqldcol* 구조에 대한 포인터. 이 포인터가 널(NULL)이거나 지정된 구조에 0 문자가 있으면, 이 조치는 기본 스펙 선택으로 해석됩니다.

모든 옵션이 지원되는 모든 파일 유형에 사용될 수는 없습니다.

*Administrative API Reference*에 있는 “SQLCHAR” 및 20 페이지의 『파일 유형 수정자(내보내기)』에서 자세한 내용을 참조하십시오.

pMsgFileName

입력. 유틸리티에 의해 리턴된 오류, 경고 및 정보용 메시지에 대한 목적지를 포함하는 문자열. 운영 체제 파일이나 표준 장치의 경로 및 이름이 될 수 있습니다. 파일이 이미 있으면, 겹쳐 쓰입니다. 파일이 없으면, 파일이 작성됩니다.

CallerAction

입력. 호출자가 요청한 조치. 유효한 값(*sqlutil*에 정의됨)은 다음과 같습니다.

SQLU_INITIAL

초기 호출. 이 값은 API에 대한 첫번째 호출에 사용되어야 합니다.

초기 호출이나 이후 계속된 다른 호출이 리턴하여 호출 응용프로그램이 요청된 내보내기 작업을 완료하기 전에 일부 조치를 수행하도록 요구할 경우, 호출자 조치는 다음 중 하나로 설정되어야 합니다.

SQLU_CONTINUE

계속 처리. 이 값은 초기 호출이 사용자 입력을 요청하는 유틸리티(예: 테이블이 종료된 상태에 응답하도록) 상태와 함께 리턴한

후, API에 대한 계속된 호출에만 사용될 수 있습니다. 이는 유틸리티가 요청한 사용자 조치가 완료되고, 유틸리티가 초기 요청을 계속 처리할 수 있다는 것을 지정합니다.

SQLU_TERMINATE

처리 종료. 이 값은 초기 호출이 사용자 입력을 요청하는 유틸리티(예: 테이블이 종료된 상태에 응답하도록) 상태와 함께 리턴한 후, API에 대한 계속된 호출에만 사용될 수 있습니다. 이는 유틸리티가 요청한 사용자 조치가 수행되지 않고, 유틸리티가 초기 요청 처리를 종료하려 함을 지정합니다.

pOutputInfo

출력. 목표 파일에 내보낸 레코드 수를 리턴합니다. 19 페이지의 『SQLUEXPT-OUT 데이터 구조』에서 이 구조에 관한 자세한 정보를 참조하십시오.

pReserved

향후 사용을 위해 예약됨.

pSqlca

출력. *sqlca* 구조에 대한 포인터. 이 구조에 대해서는 *Administrative API Reference*에 있는 “SQLCA”에서 자세한 내용을 참조하십시오.

REXX API 구문

```
EXPORT :stmt TO datafile OF filetype
[MODIFIED BY :filetmod] [USING :dcoldata]
MESSAGES msgfile [ROWS EXPORTED :number]
```

```
CONTINUE EXPORT
```

```
STOP EXPORT
```

REXX API 매개변수

stmt 유효한 동적 SQL SELECT문을 포함하는 REXX 호스트 변수. 이 명령문은 데이터베이스로부터 발췌될 데이터를 지정합니다.

내보내기 API

datafile

데이터가 내보내질 파일 이름

filetype

내보내기 파일의 데이터 형식. 지원되는 파일 형식은 다음과 같습니다.

DEL 컬럼 식별자가 있는 ASCII

WSF 워크시트 형식

IXF 통합 교환 형식(IXF: Integrated eXchange Format)의 PC 버전

filemod

추가 처리 옵션을 포함하는 호스트 변수(20 페이지의 『파일 유형 수정자 (내보내기)』 참조)

dcoldata

내보내기 파일에 사용될 컬럼 이름을 포함하는 복합 REXX 호스트 변수. 다음에서, XXX는 호스트 변수 이름을 나타냅니다.

XXX.0

컬럼 수(변수의 나머지에 있는 요소 수).

XXX.1

첫번째 컬럼 이름

XXX.2

두번째 컬럼 이름

XXX.3

계속

이 매개변수가 널(NULL)이거나 *dcoldata*에 대한 값이 지정되어 있지 않으면, 유틸리티는 데이터베이스 테이블의 컬럼 이름을 사용합니다.

msgfile

오류 및 경고 메시지가 송신될 파일, 경로 또는 장치 이름

number

내보낸 행 수를 포함할 호스트 변수

SQLUEXPT-OUT 데이터 구조

이 구조는 12 페이지의 『내보내기 API』로부터 데이터를 전달하는 데 사용됩니다.

표 1. SQLUEXPT-OUT 구조의 필드

필드 이름	데이터 유형	설명
SIZEOFSTRUCT	INTEGER	구조의 크기
ROWSEXPORTED	INTEGER	데이터베이스로부터 목표 파일로 내보낸 레코드의 수

언어 구문

C 구조

```

/* File: sqlutil.h */
/* Structure: SQL-UExPT-OUT */
/* ... */
SQL_STRUCTURE sqluexpt_out
{
    sqluint32      sizeofStruct;
    sqluint32      rowsExported;
};
/* ... */

```

COBOL 구조

```

* File: sqlutil.cbl
01 SQL-UExPT-OUT.
   05 SQL-SIZE-OF-UExPT-OUT  PIC 9(9) COMP-5 VALUE 8.
   05 SQL-ROWSEXPORTED      PIC 9(9) COMP-5 VALUE 0.
*

```

파일 유형 수정자(내보내기)

표 2. 유효한 파일 유형 수정자(내보내기)

수정자	설명
모든 파일 형식	
lobsinfile	<i>lob-path</i> 는 LOB 값을 포함하는 파일에 대한 경로를 지정합니다.
DEL(컬럼 식별자가 있는 ASCII) 파일 형식	
chardelx	<i>x</i> 는 단일 문자열 분리 문자입니다. 기본값은 큰 따옴표(")입니다. 한 문자열을 묶기 위해 큰 따옴표 대신 지정된 문자가 사용됩니다. ^a 작은 따옴표(')도 다음과 같이 한 문자열 분리 문자로 지정될 수 있습니다. modified by chardel''
coldelx	<i>x</i> 는 단일 문자 컬럼 분리 문자입니다. 기본값은 쉼표(.)입니다. 컬럼의 끝을 표시하기 위해 쉼표 대신 지정된 문자가 사용됩니다. ^a 다음 예에서 coldel;이 내보내기 유틸리티가 발견한 모든 세미콜론 (;)을 컬럼 분리 문자로 해석하게 합니다. db2 "export to temp of del modified by coldel; select * from staff where dept = 20"
datesiso	날짜 형식. 모든 날짜 데이터 값이 ISO 형식("YYYY-MM-DD")으로 내보내도록 합니다. ^b
decplusblank	더하기 부호 문자. 양수(십진수) 값 앞에 더하기 부호(+) 대신 공백을 두게 합니다. 기본 조치는 양수(십진수) 값 앞에 더하기 부호를 붙이는 것입니다.
decptx	<i>x</i> 는 마침표 대신 소수점 문자로 사용하는 단일 문자입니다. 기본값은 마침표(.)입니다. 소수점 문자로 마침표 대신 지정된 문자가 사용됩니다. ^a
dldelx	<i>x</i> 는 단일 문자 DATALINK 분리 문자입니다. 기본값은 세미콜론(;)입니다. DATALINK 값에 대한 필드간 분리자로서 세미콜론 대신 지정된 문자가 사용됩니다. 이는 DATALINK 값이 둘 이상의 부속값을 가질 수 있기 때문에 필요합니다. ^a 주: <i>x</i> 는 행, 컬럼 또는 문자열 분리 문자와 같은 문자로 지정되지 않아야 합니다.
nodoubledel	이중 문자 분리 문자의 인식을 억제합니다. 21 페이지의 『분리 문자 제한사항』에서 자세한 내용을 참조하십시오.
WSF 파일 형식	
1	Lotus 1-2-3 릴리스 1 또는 Lotus 1-2-3 릴리스 1a와 호환 가능한 WSF 파일을 작성합니다. ^c 이는 기본값입니다.

표 2. 유효한 파일 유형 수정자(내보내기) (계속)

수정자	설명
2	Lotus Symphony 릴리스 1.0과 호환 가능한 WSF 파일을 작성합니다. ^c
3	Lotus 1-2-3 버전 2 또는 Lotus Symphony 릴리스 1.1과 호환 가능한 WSF 파일을 작성합니다. ^c
4	DBCS 문자를 포함하는 WSF 파일을 작성합니다.
<p>주:</p> <ol style="list-style-type: none"> 1. MODIFIED BY 옵션을 사용하여 지원되지 않는 파일 유형을 사용하려 할 경우, 내보내기 유틸리티는 경고를 발행하지 않습니다. 이를 시도하면, 내보내기 조작이 실패하며 오류 코드가 리턴됩니다. 2. ^a 『분리 문자 제한사항』에서는 분리 문자가 겹쳐쓸 때 사용될 수 있는 문자에 대해 적용되는 제한사항을 나열합니다. 3. ^b 내보내기 유틸리티는 일반적으로 다음을 기록합니다. <ul style="list-style-type: none"> • YYYYMMDD 형식의 날짜 데이터 • "YYYY-MM-DD" 형식의 char(date) 데이터 • "HH.MM.SS" 형식의 시간 데이터 • "YYYY-MM-DD-HH.MM.SS.aaaaaa" 형식의 시간소인 데이터 <p>내보내기 조작에 대한 SELECT문에 지정된 datetime 컬럼에 들어 있는 데이터도 이러한 형식을 취하게 됩니다.</p> 4. ^c 이들 파일은 filetype-mod 매개변수 문자열에서 Lotus 1-2-3에 대해 L을 지정하고, Symphony에 대해 S를 지정하여 특정 제품으로 방향 지시될 수 있습니다. 한 개의 값 또는 제품 지정자만 지정되는 경우가 있을 수 있습니다. 	

분리 문자 제한사항

선택된 분리 문자가 이동될 데이터의 부분이 아닌지를 확인하는 것은 사용자의 책임입니다. 만약 그렇다면, 예기치 않은 오류가 발생할 수 있습니다. 다음의 제한사항은 데이터 이동시 컬럼, 문자열, DATALINK 및 소수점 분리 문자에 대해 적용됩니다.

- 분리 문자는 상호 배타적입니다.
- 분리 문자는 2진 0, 라인 피드 문자, 캐리지 리턴 또는 공백 공간이 될 수 없습니다.

파일 유형 수정자(내보내기)

- 기본 소수점(.)은 문자열 분리 문자가 될 수 없습니다.
- 다음 문자는 ASCII 계열 코드 페이지와 EBCDIC 계열 코드 페이지에 의해 각기 다르게 지정됩니다.
 - Shift-In(0x0F) 및 Shift-Out(0x0E) 문자는 EBCDIC MBCS 데이터 파일에 대한 분리 문자가 될 수 없습니다.
 - MBCS, EUC 또는 DBCS 코드 페이지에 대한 분리 문자는 EBCDIC MBCS 데이터에 대한 기본 소수점인 0x4b를 제외하고는 0x40보다 클 수 없습니다.
 - ASCII 코드 페이지 또는 EBCDIC MBCS 코드 페이지의 데이터 파일에 대한 기본 분리 문자는 다음과 같습니다.
 - " (0x22, 큰 따옴표: 문자열 분리 문자)
, (0x2c, 쉼표: 컬럼 분리 문자)
 - EBCDIC SBCS 코드 페이지에 있는 데이터 파일에 대한 기본 분리 문자는 다음과 같습니다.
 - " (0x7F, 큰 따옴표: 문자열 분리 문자)
, (0x6B, 쉼표; 컬럼 분리 문자)
 - ASCII 데이터 파일에 대한 기본 소수점은 0x2e(마침표)입니다.
 - EBCDIC 데이터 파일에 대한 기본 소수점은 0x4B(마침표)입니다.
 - 서버의 코드 페이지가 클라이언트의 코드 페이지와 다를 경우, 기본값이 아닌 분리 문자의 16진 표현이 지정될 것을 권장합니다. 예를 들면,

```
db2 load from ... modified by charde10x0C colde1X1e ...
```

DEL 파일에서 이중 문자 분리 문자 인식에 대한 지원에 관한 다음 정보는 내보내기, 가져오기 및 로드 유틸리티에 대해 적용됩니다.

- 문자 분리 문자는 DEL 파일의 문자 기준 필드에서 허용됩니다. 이는 CHAR, VARCHAR, LONG VARCHAR 또는 CLOB 유형의 필드에 적용됩니다 (lobsinfile이 지정된 경우 제외). 문자 분리 문자 사이에 있는 모든 문자 분리 문자의 쌍은 데이터베이스로 가져오거나 로드됩니다. 예를 들면,

```
"What a ""nice"" day!"
```

는 다음과 같이 가져오기됩니다.

```
What a "nice" day!
```

내보내기의 경우, 이 규칙은 반대로 적용됩니다. 예를 들면,

```
I am 6" tall.
```

은 다음과 같이 DEL 파일에 내보냅니다.

```
"I am 6"" tall."
```

- DBCS 환경의 경우, 파이프() 문자 분리 문자는 지원되지 않습니다.

내보내기 세션 예

CLP 예

다음 예에서는 SAMPLE 데이터베이스(사용자가 연결되어야 하는)에 있는 STAFF 테이블의 정보를 IXF 형식의 출력으로 myfile.ixf에 내보내는 방법을 보여줍니다. 데이터베이스가 DB2 Connect를 통해 연결되어 있지 않은 경우, 색인 정보(있을 경우)가 출력 파일에 저장됩니다. 그렇지 않으면, 데이터만 저장됩니다.

```
db2 export to myfile.ixf of ixf messages msgs.txt select * from staff
```

다음 예에서는 SAMPLE 데이터베이스(사용자가 연결되어야 함)의 STAFF 테이블에서 부서 20에 있는 직원에 대한 정보를 IXF 형식의 출력으로 awards.ixf에 내보내는 방법을 보여줍니다.

```
db2 export to awards.ixf of ixf messages msgs.txt select * from staff
where dept = 20
```

다음 예에서는 LOB를 DEL 파일로 내보내는 방법을 보여줍니다.

```
db2 export to myfile.del of del lobs to mylobs
lobfile lobs1, lobs2 modified by lobsinfile
select * from emp_photo
```

다음 예에서는 첫번째 디렉토리에 적합하지 않을 수 있는 파일에 대해 두번째 디렉토리를 지정하고 LOB를 DEL 파일로 내보내는 방법을 보여줍니다.

```
db2 export to myfile.del of del
lobs to /db2exp1, /db2exp2 modified by lobsinfile
select * from emp_photo
```

내보내기 세션 예

다음 예에서는 작은 따옴표를 문자열 분리 문자로, 세미콜론을 컬럼 분리 문자로, 쉼표를 소수점으로 사용하고 데이터를 DEL 파일로 내보내는 방법을 보여줍니다. 데이터를 데이터베이스로 다시 가져올 때에도 동일한 규칙을 사용해야 합니다.

```
db2 export to myfile.del of del
      modified by char del'' coldel; decpt,
      select * from staff
```

API 예

다음 샘플 프로그램에서는 다음을 수행하는 방법을 보여줍니다.

- SAMPLE 데이터베이스의 STAFF 테이블의 정보를 EXPTABLE.DEL 파일로 내보내기
- 이 정보를 분리된 텍스트 파일로부터 새로운 테이블 IMPTABLE로 가져오기.

관리 안내서에서 SAMPLE 데이터베이스에 관한 자세한 정보를 참조하십시오.

이 샘플 프로그램에 대한 소스 파일(impexp.sqc)은 \sql1ib\samples\c 디렉토리에서 찾을 수 있습니다. 이는 DB2 API와 Embedded SQL 호출을 모두 포함합니다. 같은 디렉토리에 위치한 스크립트 파일 bldvaemb.cmd에는 샘플 프로그램 및 다른 샘플 프로그램을 빌드하기 위한 명령이 들어 있습니다. 응용프로그램 빌드 안내서에서 DB2 관리 API를 포함하는 응용프로그램 작성에 관한 일반적인 정보 및 컴파일과 링크 옵션에 관한 자세한 정보를 참조하십시오. OS/2상의 impexp.sqc 소스 파일로부터 샘플 프로그램 impexp를 빌드하려면, 다음을 수행하십시오.

1. impexp.sqc, bldvaemb.cmd, util.c 및 util.h 파일을 작업 디렉토리에 복사하십시오.
2. 데이터베이스 관리 프로그램이 수행중이지 않을 경우, db2start 명령을 발행하십시오.
3. bldvaemb impexp sample을 입력하십시오. 다음 파일이 생성됩니다.

```
impexp.bnd
impexp.c
util.obj
impexp.obj
impexp.exe
```

샘플 프로그램(실행 가능한 파일)을 수행시키려면 impexp를 입력하십시오. 생성된 일부 파일(예: 메시지 파일) 및 컬럼 식별자가 있는 ASCII 데이터 파일을 조사하는 데 유용하다는 것을 알게 됩니다.

```

/*****
**
** Source File Name = impexp.sqc 1.4
**
** PURPOSE :
**   This program is an example of how APIs are implemented in order to
**   export and import tables and table data. The order of the program
**   is as follows:
**     - export a table to a comma-delimited text file
**     - import the comma-delimited text file to a DB2 table
**   This program needs the embedded SQL calls in order to connect to
**   an existing database, then to create a temporary table to work with.
**
** STRUCTURES USED :
**   sqldcol
**   sqlchar
**   sqluexpt_out
**   sqluimp_in
**   sqluimp_out
**   sqlca
**
** APIs USED :
**       IMPORT TO          sqluimpt_api
**       EXPORT            sqlgexpt
**
** FUNCTIONS DECLARED :
**   'C' COMPILER LIBRARY :
**     stdio.h - printf
**     string.h - fgets, strncpy
**
**   DBMS LIBRARY :
**     sqlenv.h - see "APIs USED" above
**
**   OTHER :
**     internal :
**       list_dcs :          Displays a directory of databases
**
**     external :
**       check_error :      Checks for SQLCODE error, and prints out any
**       [in UTIL.C]       related information available.
**
** EXTERNAL DEPENDANCIES :
**   - Ensure existence of database (SAMPLE) for precompile purposes.
**   - Precompile with the SQL precompiler (PREP in DB2)
**   - Bind to a database (BIND in DB2)
**   - Compile and link with the IBM Cset++ compiler(AIX and OS/2)
**     or the Microsoft Visual C++ compiler (Windows)
**     or the compiler supported on your platform.
**
** *****/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sqlenv.h>

```

내보내기 세션 예

```
#include <sqlutil.h>
#ifdef DB2MAC
#include <malloc.h>
#endif
#include "util.h"

#ifdef DB268K
/* Need to include ASLM for 68K applications */
#include <LibraryManager.h>
#endif

#define CHECKERR(CE_STR)  if (check_error (CE_STR, &sqlca) != 0) return 1;

EXEC SQL INCLUDE SQLCA;
int main (int argc, char *argv[]) {
    short int      callerAction = 0;
    struct sql_dcol   columnData;
    struct sqlchar   *columnStringPointer;
    struct sqluexpt_out  outputInfo;
    struct sqluimpt_in  impInput;
    struct sqluimpt_out  impOutput;

    char      datafile[] = "EXPTABLE.DEL";
    char      statement[] = "select name, id from staff";
    char      impStatement[] = "insert into imptable (name, id)";
    char      msgfile_x[] = "EXPMSG.TXT";
    char      msgfile_m[] = "IMPMSG.TXT";
    char      fileFormat[] = "DEL";

    EXEC SQL BEGIN DECLARE SECTION;
        char userid[9];
        char passwd[19];
    EXEC SQL END DECLARE SECTION;

#ifdef DB268K
    /* Before making any API calls for 68K environment,
       need to initial the Library Manager */
        InitLibraryManager(0,kCurrentZone,kNormalMemory);
        atexit(CleanupLibraryManager);
#endif

    /* need to preset the size of structure field and counts */
    outputInfo.sizeOfStruct = SQLUEXPT_OUT_SIZE;
    impInput.sizeOfStruct = SQLUIMPT_IN_SIZE;
    impOutput.sizeOfStruct = SQLUIMPT_OUT_SIZE;
    impInput.restartcnt = impInput.commitcnt = 0;

    /******\
    * need to allocate the proper amount of space for the SQL statement *
    \******/
    columnStringPointer = (struct sqlchar *)malloc(strlen(statement)
        + sizeof (struct sqlchar));
    columnStringPointer->length = strlen(statement);
    strncpy (columnStringPointer->data, statement, strlen(statement));
    /* DELimited format can not have specified names, therefore the
       column method is 'D'efault */
    columnData.dcolmeth = 'D';

    if(argc == 1) {
        EXEC SQL CONNECT TO sample;
        CHECKERR ("CONNECT TO SAMPLE");
    }
    else if(argc == 3) {
```

```

        strcpy (userid, argv[1]);
        strcpy (passwd, argv[2]);
        EXEC SQL CONNECT TO sample USER :userid USING :passwd;
        CHECKERR ("CONNECT TO SAMPLE");
    }
    else {
        printf ("\nUSAGE: impexp [userid passwd]\n\n");
        return 1;
    } /* endif */

    printf ("exporting NAME and ID from STAFF table into file '%s'\n", datafile);
    /*****
    * EXPORT API called *
    \*****/
    sqluexpr (datafile, NULL, NULL, &columnData, columnStringPointer,
              fileFormat, NULL, msgfile_x, 0, &outputInfo, NULL, &sqlca);
    CHECKERR ("exporting table");
    printf ("rows exported %d\n", outputInfo.rowsExported);
    free (columnStringPointer);

    /*****
    * need to allocate the proper amount of space for the SQL statement *
    \*****/
    columnStringPointer = (struct sqlchar *)malloc(strlen(impStatement)
        + sizeof (struct sqlchar));
    columnStringPointer->length = strlen(impStatement);
    strncpy (columnStringPointer->data, impStatement, strlen(impStatement));

    printf ("creating a temporary table 'imptable' to import into\n");
    /* create a temporary table to import into */
    EXEC SQL CREATE TABLE imptable (name VARCHAR(15), id INT);
    CHECKERR ("CREATE TABLE");

    printf ("importing the file '%s' into the 'imptable'\n", datafile);
    /*****
    * IMPORT API called *
    \*****/
    sqluimpr (datafile, NULL, &columnData, columnStringPointer, fileFormat,
              NULL, msgfile_m, 0, &impInput, &impOutput, NULL, NULL, &sqlca);
    CHECKERR ("importing table");
    printf ("rows imported %d\nnumber of rows committed %d\n",
            impOutput.rowsInserted, impOutput.rowsCommitted);

    free (columnStringPointer);

    /* drop the table */
    EXEC SQL DROP TABLE imptable;
    CHECKERR ("DROP TABLE");

    EXEC SQL CONNECT RESET;
    CHECKERR ("CONNECT RESET");
}
/* end of program : impexp.sqc */

```

제한사항

다음 제한사항은 내보내기 유틸리티에 적용됩니다.

- 이 유틸리티는 별명의 사용을 지원하지 않습니다.
- 이 유틸리티는 구조화된 유형 컬럼이 있는 테이블을 지원하지 않습니다.

문제점 해결

데이터를 내보내기, 가져오기, 로딩, 바인딩 또는 복원과 같은 DB2 조작을 수행하는 동안 메시지 파일이 이러한 조작과 연관된 오류, 경고 및 정보용 메시지를 포함하여 작성되게 지정할 수 있습니다. MESSAGE 매개변수를 사용하여 이들 파일의 이름을 지정하십시오.

이들 메시지 파일은 표준 ASCII 텍스트 파일입니다. 메시지 파일의 각 메시지는 새로운 행에서 시작하며 DB2 메시지 검색 기능이 제공하는 정보가 들어 있습니다. 이 메시지 파일을 인쇄하려면 운영 체제에 대한 인쇄 절차를 사용하고, 메시지를 보려면 ASCII 편집기를 사용하십시오.

제2장 가져오기

여기에서는 DB2 UDB 가져오기 유틸리티에 대해 설명하며, 이 유틸리티는 SQL INSERT문을 사용하여 입력 파일의 데이터를 테이블이나 뷰에 기록합니다. 목표 테이블이나 뷰에 이미 데이터가 들어 있으면, 기존의 데이터를 바꾸거나 추가시킬 수 있습니다.

다음과 같은 내용이 들어 있습니다.

- 30 페이지의 『가져오기 개요』
- 31 페이지의 『가져오기를 사용하는 데 필요한 특권, 권한 및 권한 부여』
- 32 페이지의 『가져오기 사용』
- 34 페이지의 『버퍼된 삽입으로 가져오기 사용』
- 34 페이지의 『식별 컬럼으로 가져오기 사용』
- 36 페이지의 『생성된 컬럼으로 가져오기 사용』
- 38 페이지의 『내보낸 테이블 재작성』
- 40 페이지의 『대형 오브젝트(LOB) 가져오기』
- 40 페이지의 『사용자 정의 구별 유형(UDT) 가져오기』
- 41 페이지의 『가져오기 동안 테이블 잠금』
- 42 페이지의 『IMPORT 명령』
- 51 페이지의 『가져오기 API』
- 67 페이지의 『파일 유형 수정자(가져오기)』
- 77 페이지의 『문자 세트 및 NLS 고려사항』
- 77 페이지의 『가져오기 세션 예』
- 80 페이지의 『가져오기 성능 최적화』
- 80 페이지의 『제한사항 및 한계점』
- 81 페이지의 『문제점 해결』.

208 페이지의 『DB2 Data Links Manager 데이터를 이동시키기 위해 가져오기 유틸리티 사용』에서 DB2 Data Links Manager 데이터 가져오기에 관한 정보를 참조하십시오. 222 페이지의 『입력된 테이블간의 데이터 이동』에서 입력된 테이블로부터 데이터 가져오기에 관한 정보를 참조하십시오. 220 페이지의 『DB2 Connect 를 사용한 데이터 이동』에서 DB2 Connect 워크스테이션의 파일로부터 DRDA 서버 데이터베이스로 데이터 가져오기 및 이와 반대 방향으로 가져오기에 관한 내용을 참조하십시오.

가져오기 개요

가져오기 유틸리티는 입력 파일의 데이터를 테이블이나 갱신 가능한 뷰로 삽입합니다. 가져온 데이터를 수신하는 테이블이나 뷰에 이미 데이터가 들어 있으면, 이미 있는 데이터를 바꾸거나 이 데이터에 추가시킬 수 있습니다.

다음은 데이터를 가져올 때 필요한 내용입니다.

- 입력 파일의 경로와 이름
- 목표 테이블 또는 뷰의 이름이나 별명
- 입력 파일에 있는 데이터의 형식. 이 형식은 IXF, WSF, DEL 또는 ASC가 될 수 있습니다. 239 페이지의 『부록C. 내보내기/가져오기/로드 유틸리티 파일 형식』에서 자세한 내용을 참조하십시오.
- 입력 데이터가 테이블이나 뷰로 삽입되는지의 여부, 또는 테이블이나 뷰에 있는 기존 데이터가 입력 데이터로 갱신되거나 바뀌게 되는지의 여부
- 유틸리티가 API, `sqluimpr`을 통해 호출될 경우의 메시지 파일 이름
- 입력된 테이블에서 작업할 때, 모든 구성 유형을 통해 진행하기 위한 방법이나 순서를 제공해야 합니다. 계층 구조에 있는 모든 상위 테이블과 서브테이블을 위에서 아래로, 좌에서 우로 처리하는 순서를 *트래버스* 순서라고 합니다. 이 순서는 다른 데이터와 관계하여 데이터가 이동되는 곳을 결정하기 때문에 테이블 계층 구조간에 데이터를 이동시킬 때 중요합니다.

입력된 테이블에서 작업할 때에는 서브테이블 목록을 제공해야 할 수도 있습니다. 이 목록은 데이터를 가져오기 위한 서브테이블과 속성을 표시합니다.

222 페이지의 『입력된 테이블간의 데이터 이동』에서 자세한 내용을 참조하십시오.

다음과 같이 지정할 수도 있습니다.

- 데이터를 가져오기 위해 사용하는 방법: 컬럼 위치(location), 컬럼 이름 또는 상대 컬럼 위치(position)
- 테이블에 대한 변경사항을 요약하기 전에 가져오기 위한 행의 수. 정기적으로 COMMIT를 요청하면 가져오기 조작이 도중에 실패하고 ROLLBACK이 일어날 때 유실되는 행의 수를 줄입니다. 대형 입력 파일을 처리할 때 DB2 로그가 가득 차게 되는 것도 막습니다.
- 가져오기 조작을 시작하기 전에 건너뛰는 파일 레코드의 수. 오류가 발생하면, 제대로 가져오기되고 요약된 마지막 행에 이어 즉시 가져오기 조작을 재시작할 수 있습니다.
- 데이터가 삽입될 테이블이나 뷰에 있는 컬럼 이름

가져오기를 사용하는 데 필요한 특권, 권한 및 권한 부여

특권은 사용자가 데이터베이스 자원을 작성하거나 액세스할 수 있게 합니다. 권한 레벨은 특권과 상위 레벨 데이터베이스 관리 프로그램 유지보수 및 유틸리티 조작을 그룹으로 분류하는 방법을 제공합니다. 이는 모두 데이터베이스 관리 프로그램과 데이터베이스 오브젝트에 대한 액세스를 제어하기 위한 것입니다. 사용자는 적합한 권한 즉, 필요한 특권이나 권한을 가지고 있는 오브젝트만 액세스할 수 있습니다.

새로운 테이블을 작성하기 위해 가져오기 유틸리티를 사용하려면, 데이터베이스에 대한 SYSADM 권한, DBADM 권한 또는 CREATETAB 특권이 있어야 합니다. 이미 있는 테이블이나 뷰에 있는 데이터를 바꾸려면, 그 테이블이나 뷰에 대한 SYSADM 권한, DBADM 권한 또는 CONTROL 특권이 있어야 합니다. 이미 있는 테이블이나 뷰에 데이터를 추가하려면, 그 테이블이나 뷰에 대한 SELECT 및 INSERT 특권이 있어야 합니다.

가져오기 사용

가져오기를 사용하기 전에

가져오기 유틸리티를 호출하기 전에, 데이터가 가져올 데이터베이스에 연결되어야 합니다(또는 내재적으로 연결할 수 있어야 합니다). 유틸리티가 COMMIT 또는 ROLLBACK문을 발행할 것이므로, 가져오기를 호출하기 전에 COMMIT나 ROLLBACK을 수행하여 모든 트랜잭션을 완료하고 모든 잠금을 해제해야 합니다.

가져오기 호출

가져오기 유틸리티는 다음을 통해 호출될 수 있습니다.

- 명령행 처리기(CLP)

다음은 CLP를 통해 발행된 가져오기 명령의 예입니다.

```
db2 import from stafftab.ixf of ixf insert into userid.staff
```

- 제어 센터에 있는 가져오기 노트북. 가져오기 노트북을 열려면 다음을 수행하십시오.

1. 제어 센터에서, 테이블 폴더가 나올 때까지 오브젝트 트리를 확장하십시오.
2. 테이블 폴더를 클릭하십시오. 이미 있는 테이블이 창의 오른쪽에 있는 파티션 영역에 표시됩니다(내용 영역).
3. 내용 영역에서 원하는 테이블을 오른쪽 마우스 버튼으로 클릭한 후, 팝업 메뉴에서 가져오기를 선택하십시오. 가져오기 노트북이 열립니다.

관리 안내서에서 제어 센터에 관한 일반적인 정보를 참조하십시오. 자세한 정보는 제어 센터의 온라인 도움말 기능을 통해 제공됩니다.

- API, [sqluimpr](#). 51 페이지의 『가져오기 API』에서 이 API에 관한 정보를 참조하십시오. 응용프로그램 빌드 안내서에서 DB2 관리 API를 포함하는 응용프로그램 작성에 관한 일반적인 정보를 참조하십시오.

클라이언트/서버 환경에서 가져오기 사용

원격 데이터베이스로 파일을 가져올 때, 서버에서 가져오기를 수행하기 위해 저장 프로시저어가 호출될 수 있습니다. 저장 프로시저어는 다음과 같은 경우에는 호출되지 않습니다.

- 응용프로그램과 데이터베이스 코드 페이지가 다를 경우
- 가져오기 중인 파일이 다중 파트 PC/IXF 파일인 경우
- 데이터를 가져오는 데 사용된 방법이 컬럼 이름 또는 상대 컬럼 위치(position)인 경우
- 제공된 목표 컬럼 목록이 4KB보다 길 경우
- OS/2 또는 DOS 클라이언트가 디스켓에서 파일을 가져오는 경우
- LOBS FROM절 또는 lobsinfile 수정자가 지정된 경우
- NULL INDICATORS절이 ASC 파일에 대해 지정된 경우

가져오기 유틸리티가 저장 프로시저어를 사용할 때, 메시지가 서버에 설치되어 있는 기본 언어를 사용하여 메시지 파일에 작성됩니다. 이 메시지는 클라이언트와 서버에 있는 언어가 같을 경우, 응용프로그램의 언어로 되어 있습니다.

가져오기 유틸리티는 sqllib 디렉토리(또는 지정된 경우, **DB2INSTPROF** 레지스트리 변수가 나타내는 디렉토리)의 tmp 서브디렉토리에 두 개의 임시 파일을 작성합니다. 하나의 파일은 데이터를 위한 파일이고 다른 하나의 파일은 가져오기 유틸리티가 생성한 메시지를 위한 파일입니다.

서버에서 데이터를 작성 또는 열기에 관한 오류를 받게 될 경우, 다음 내용을 확인하십시오.

- 디렉토리가 있음
- 파일에 대해 충분한 디스크 공간이 있음
- 인스턴스 소유자가 디렉토리에서 쓰기 사용권한을 가지고 있음

버퍼된 삽입으로 가져오기 사용

파티션된 데이터베이스 환경에서, 가져오기 유틸리티는 버퍼된 삽입을 사용하도록 작동할 수 있습니다. 이는 데이터 가져오기시 발생하는 메시지 수를 줄이고, 보다 좋은 성능을 낼 수 있습니다. 단, 버퍼된 삽입이 실패할 경우, 세부사항이 리턴되지 않기 때문에 이 옵션은 사용자가 오류 보고에 대해 염려하지 않을 경우에만 작동되어야 합니다.

버퍼된 삽입(insert)을 요청하려면 DB2 바인드 유틸리티를 사용하십시오. 가져오기 패키지, db2uimp.bnd는 INSERT BUF 옵션을 사용하여 데이터베이스에 대해 리바인드되어야 합니다. 예를 들면, 다음과 같습니다.

```
db2 connect to your_database
db2 bind db2uimp.bnd insert buf
```

주: 버퍼된 삽입(insert) 기능은 INSERT_UPDATE 매개변수가 지정되어 있는 모든 가져오기 조작중에는 작동 불가능합니다.

식별 컬럼으로 가져오기 사용

가져오기 유틸리티는 식별 컬럼이 들어 있는 테이블로 데이터를 가져오는 데 사용할 수 있습니다. 식별 관련 파일 유형 수정자를 사용하지 않는 경우에는 유틸리티가 다음 규칙에 따라 기능합니다.

- 식별 컬럼이 GENERATED ALWAYS이면, 입력 파일의 해당 행이 식별 컬럼에 대해 값을 유실하거나 널(NULL) 값이 명시적으로 주어질 때마다 식별 값이 테이블행에 대해 생성됩니다. 식별 컬럼에 대해 비 널(NULL) 값이 지정되면 행이 거부됩니다(SQL3550W).
- 식별 컬럼이 GENERATED BY DEFAULT인 경우에, 가져오기 유틸리티는 사용자 제공 값이 주어진다면 그 값을 사용합니다. 데이터가 유실되거나 명시적으로 널(NULL)인 경우에는 값이 생성됩니다.

가져오기 유틸리티는 식별 컬럼의 데이터 유형(즉, SMALLINT, INT, BIGINT 또는 DECIMAL) 값에 대해 일반적으로 수행되는 정도를 넘어서 사용자 제공 식별 값을 추가로 검증하지 않습니다. 중복값은 보고되지 않습니다. 또한, 식별 컬럼이 있는 테이블로 데이터를 가져올 때는 compound=x 수정자를 사용할 수 없습니다.

가져오기 유틸리티에서는 식별 컬럼이 들어 있는 테이블과의 사용을 간소화하기 위해 두 가지 파일 유형 수정자를 지원합니다.

- `identitymissing` 수정자를 사용하면 입력 데이터 파일에 식별 컬럼에 대한 값이 아무 것도 들어 있지 않은 경우(널(NULL)까지도)에 식별 컬럼이 있는 테이블을 보다 편리하게 가져올 수 있습니다. 예를 들어, 다음과 같은 SQL문으로 정의된 테이블을 고려해 봅시다.

```
create table table1 (c1 char(30),
                    c2 int generated by default as identity,
                    c3 real,
                    c4 char(1))
```

사용자는 파일(`import.del`)에서 `TABLE1`으로 데이터를 가져오려 할 수도 있으며 이 데이터는 식별 컬럼이 없는 테이블로부터 내보내질 수도 있습니다. 다음은 그러한 파일의 예입니다.

```
Robert, 45.2, J
Mike, 76.9, K
Leo, 23.4, I
```

이 파일을 가져오기 위한 한가지 방법은 다음과 같이 `IMPORT` 명령을 통해 가져올 컬럼을 명시적으로 나열하는 것입니다.

```
db2 import from import.del of del replace into table1 (c1, c3, c4)
```

그러나, 여러 컬럼이 있는 테이블의 경우에는 이 구문이 번거롭고 오류를 더 유발할 수도 있습니다. 파일 가져오기의 대체 방법은 다음과 같이 `identitymissing` 파일 유형 수정자를 사용하는 것입니다.

```
db2 import from import.del of del modified by identitymissing replace into table1
```

- `identityignore` 수정자는 일부 방식에서는 `identitymissing` 수정자의 반대입니다. 이는 입력 데이터 파일에 식별 컬럼에 대한 데이터가 들어 있을지라도 데이터를 무시하고 각 행에 대해 식별값을 생성하도록 가져오기 유틸리티에 지정합니다. 예를 들어, 사용자는 위에서 정의한대로 다음과 같은 데이터를 파일(`import.del`)에서 `TABLE1`으로 가져오길 원할 수 있습니다.

```
Robert, 1, 45.2, J
Mike, 2, 76.9, K
Leo, 3, 23.4, I
```

식별 컬럼으로 가져오기 사용

사용자 제공 값 1, 2 및 3이 식별 컬럼에 대해 사용되지 않으면, 사용자가 다음과 같은 `IMPORT` 명령을 발행할 수 있습니다.

```
db2 import from import.del of del method P(1, 3, 4) replace into table1 (c1, c3, c4)
```

또, 이러한 접근 방식도 테이블에 여러 컬럼이 있는 경우에는 번거롭고 오류를 더 유발할 수 있습니다. `identityignore` 수정자는 다음과 같이 구문을 간소화합니다.

```
db2 import from import.del of del modified by identityignore replace into table1
```

식별 컬럼이 있는 테이블을 IXF 파일로 내보낼 때는, 테이블의 식별 컬럼 속성을 포함하여 테이블을 재작성하는 데 `IMPORT` 명령의 `REPLACE_CREATE` 및 `CREATE` 옵션을 사용할 수 있습니다. 유형 `GENERATED ALWAYS`의 식별 컬럼이 들어 있는 테이블에서 그와 같은 IPX 파일이 작성된 경우에, 데이터 파일을 정상적으로 가져올 수 있는 유일한 방법은 `identityignore` 식별자를 지정하는 것입니다. 그렇지 않은 경우에는 모든 행들이 거부됩니다 (SQL3550W).

생성된 컬럼으로 가져오기 사용

가져오기 유틸리티는 생성된 (비식별) 컬럼이 들어 있는 테이블로 데이터를 가져오는 데 사용할 수 있습니다.

생성된 컬럼 관련 파일 유형 수정자를 사용하지 않는 경우에는 가져오기 유틸리티가 다음 규칙에 따라 기능합니다.

- 입력 파일의 해당 행이 컬럼에 대한 값을 유실하거나 널(NULL) 값이 명시적으로 주어질 때마다 생성된 컬럼에 대해 값이 생성됩니다. 생성된 컬럼에 대해 비 널(NULL) 값이 제공되면 행이 거부됩니다(SQL3550W).
- 널(NULL) 입력 가능하지 않은 생성된 컬럼에 대해 서버가 널(NULL) 값을 생성하면, 이 필드가 속해 있는 데이터행이 거부됩니다(SQL0407N). 예를 들어, 널(NULL) 입력 가능하지 않은 생성된 컬럼이 입력 파일의 컬럼에 널(NULL) 값을 제공한 두 테이블 컬럼의 합계로 정의된 경우에는 이러한 상황이 발생할 수 있습니다.

가져오기 유틸리티에서는 생성된 컬럼이 들어 있는 테이블과의 사용을 간소화하기 위해 두 가지 파일 유형 수정자를 지원합니다.

- `generatedmissing` 수정자를 사용하면, 테이블에 나타난 생성된 모든 컬럼에 대한 값이 아무 것도 입력 데이터 파일에 들어 있지 않은 경우(널(NULL)까지도)에, 생성된 컬럼이 있는 테이블로 데이터를 보다 편리하게 가져올 수 있습니다. 예를 들어, 다음과 같은 SQL문으로 정의된 테이블을 고려해 봅시다.

```
create table table1 (c1 int,
                    c2 int,
                    g1 int generated always as (c1 + c2),
                    g2 int generated always as (2 * c1),
                    c3 char(1))
```

사용자는 파일(`load.del`)에서 `TABLE1`으로 데이터를 가져오려 할 수도 있으며 이 데이터는 생성된 컬럼이 없는 테이블로부터 내보내질 수도 있습니다. 다음은 그러한 파일의 예입니다.

```
1, 5, J
2, 6, K
3, 7, I
```

이 파일을 가져오기 위한 한가지 방법은 다음과 같이 `IMPORT` 명령을 통해 가져올 컬럼을 명시적으로 나열하는 것입니다.

```
db2 import from import.del of del replace into table1 (c1, c2, c3)
```

그러나, 여러 컬럼이 있는 테이블의 경우에는 이 구문이 번거롭고 오류를 더 유발할 수도 있습니다. 파일 가져오기의 대체 방법은 다음과 같이 `generatedmissing` 파일 유형 수정자를 사용하는 것입니다.

```
db2 import from import.del of del modified by generatedmissing replace into table1
```

- `generatedignore` 수정자는 일부 방식에서는 `generatedmissing` 수정자의 반대입니다. 이는 입력 데이터 파일에 생성된 모든 컬럼에 대한 데이터가 들어 있을지라도 데이터를 무시하고 각 행에 대해 값을 생성하도록 가져오기 유틸리티에 지정합니다. 예를 들어, 사용자는 위에서 정의한대로 다음과 같은 데이터를 파일(`import.del`)에서 `TABLE1`으로 가져오길 원할 수 있습니다.

```
1, 5, 10, 15, J
2, 6, 11, 16, K
3, 7, 12, 17, I
```

생성된 컬럼으로 가져오기 사용

사용자 제공, 비 널(NULL) 값 10, 11 및 12(g1의 경우) 그리고 15, 16 및 17(g2의 경우)을 사용하면 행이 거부됩니다(SQL3550W). 이를 피하기 위해 사용자가 다음과 같이 IMPORT 명령을 발행할 수 있습니다.

```
db2 import from import.del of del method P(1, 2, 5) replace into table1 (c1, c2, c3)
```

또, 이러한 접근 방식도 테이블에 여러 컬럼이 있는 경우에는 번거롭고 오류를 더 유발할 수 있습니다. generatedignore 수정자는 다음과 같이 구문을 간소화합니다.

```
db2 import from import.del of del modified by generatedignore replace into table1
```

내보낸 테이블 재작성

가져오기 유틸리티를 사용하여 내보내기 유틸리티를 통해 저장된 테이블을 재작성할 수 있습니다. 테이블이 IXF 파일로 내보내져 있어야 하며, 내보내기 조작시 사용된 SELECT문은 특정 조건을 갖지 않아야 합니다(예를 들면, SELECT문에 컬럼 이름을 사용할 수 없습니다. select *만 허용됩니다). IXF 파일로부터 테이블을 작성할 때, 원래 테이블의 모든 속성이 보존되지는 않습니다. 예를 들어, 참조 제한조건, 외부 키 정의 및 사용자 정의 데이터 유형은 보존되지 않습니다. 다음과 같은 원래 테이블의 속성이 보존됩니다.

- 기본 키 이름, 정의
- 고유 제한조건 이름 및 정의 - 다른 유형의 제한조건이나 트리거는 보존되지 않음
- 컬럼 정보:
 - 컬럼 이름
 - 컬럼 데이터 유형(기본 유형으로 보존되는 사용자 정의 구별 유형 포함)
 - 식별 등록 정보
 - 길이(lob_file 유형 제외)
 - 코드 페이지 (적용 가능한 경우)
 - DATALINK 옵션
 - 식별 옵션

- 컬럼이 널(NULL) 입력 가능 또는 널(NULL) 입력 불가능으로 정의되었는지 여부
- 상수에 대한 기본값 - 다른 유형의 기본값은 제외
- 색인 정보:
 - 색인 이름
 - 색인 작성자 이름
 - 컬럼 이름 및 각 컬럼이 오름차순 또는 내림차순으로 정렬되는지 여부
 - 색인이 '고유'로 정의된 상태인지 여부
 - 색인이 클러스터된 것인지 여부
 - 색인이 역 스캔을 허용하는지 여부
 - *pctfree* 값
 - *minpctused* 값

다음과 같은 원래 테이블의 속성은 보존되지 않습니다.

- 소스가 정상 테이블, 요약 테이블, 뷰 또는 이들 소스 모두로부터의 컬럼 세트인지 여부.
- 테이블 정보:
 - 요약 테이블 정의(적용 가능한 경우)
 - 요약 테이블 옵션(적용 가능한 경우)
 - 테이블 공간 옵션 - 그러나 이 정보는 IMPORT 명령을 통해 지정될 수 있습니다.
- 컬럼 정보:
 - 상수값을 제외한 기본값
 - LOB 옵션(가능하다면)
 - 테이블 작성 명령문의 절 참조(가능하다면)
 - 참조 제한조건(가능하다면)
 - 점검 제한조건(가능하다면)
 - 생성된 컬럼 옵션(가능하다면)
- 색인 정보:

- 컬럼 포함 (가능하다면)

대형 오브젝트(LOB) 가져오기

대형 오브젝트(LOB) 컬럼으로 가져올 때, 데이터는 컬럼 데이터의 나머지와 같은 파일 또는 다른 파일에서 올 수 있습니다. 후자의 경우, 각 LOB 인스턴스에 대해 한 개의 파일이 존재합니다.

기본 입력 데이터 파일에 있는 컬럼은 가져오기 데이터(기본값) 또는 가져오기 데이터가 저장된 파일 이름을 포함합니다.

주:

1. LOB 데이터가 기본 입력 데이터 파일에 저장될 때, 32KB를 넘는 데이터는 허용되지 않습니다. 절단 경고는 무시됩니다.
2. 모든 LOB 데이터가 주 파일에 저장되어야 하거나, 각 LOB가 각기 다른 파일에 저장됩니다. 주 파일은 LOB 데이터와 파일 이름을 혼합하여 가질 수 없습니다. LOB 값은 lobsinfile 수정자(67 페이지의 『파일 유형 수정자(가져오기)』 참조) 및 LOBS FROM절(42 페이지의 『IMPORT 명령』 참조)을 사용하여 각기 다른 파일로부터 가져오기됩니다.

사용자 정의 구별 유형(UDT) 가져오기

이 가져오기 유틸리티는 사용자 정의 구별 유형(UDT)을 유사한 기본 데이터 유형으로 자동으로 캐스트합니다. 이는 사용자가 UDT를 기본 데이터 유형으로 명시적으로 캐스트해야 하는 노력을 줄여줍니다. 캐스트를 통해 SQL에서 UDT와 기본 데이터 유형간에 비교를 할 수 있습니다.

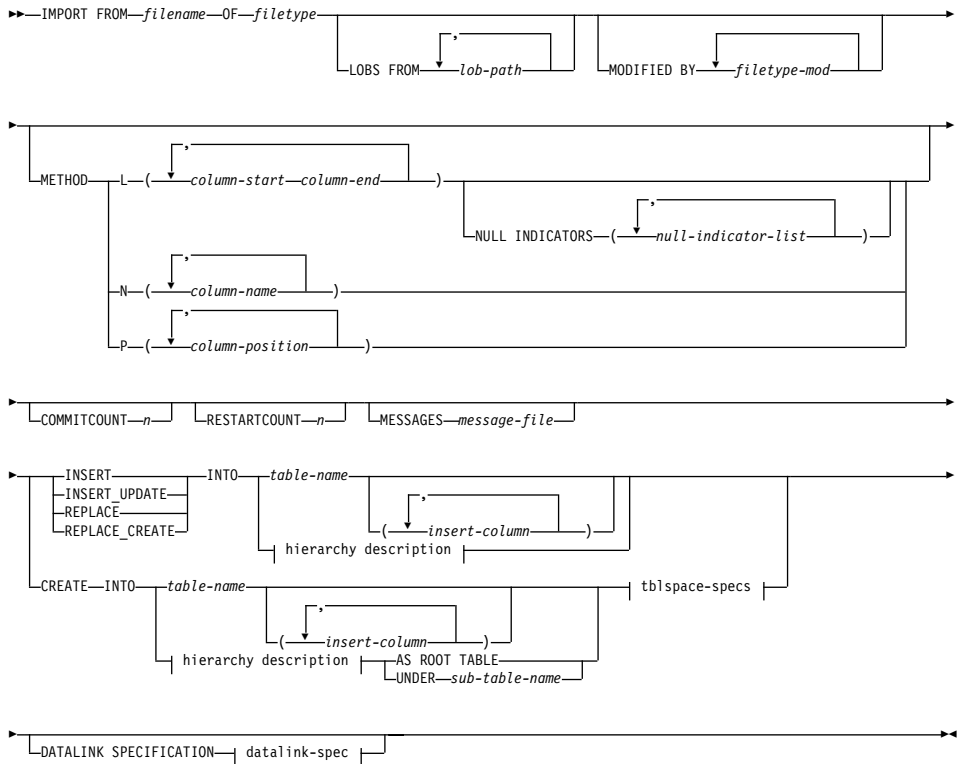
가져오기 동안 테이블 잠금

기본적으로 가져오기 유틸리티는 분리 레벨 RR(반복 읽기)인 데이터베이스로 바운드됩니다. 많은 수의 행을 테이블로 가져오기할 경우에는 기존의 잠금을 독점 잠금으로 확대해야 합니다. 동일한 테이블에서 작업 중인 다른 응용프로그램이 일부 행 잠금을 보유 중인 경우에 잠금을 독점 잠금으로 확대하면 교착 상태가 발생합니다. 이를 피하기 위해 가져오기 유틸리티가 작업 시작시에 테이블에 대해 독점 잠금을 요구합니다.

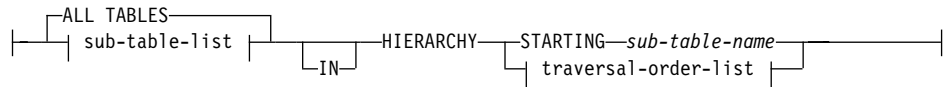
테이블에서 잠금을 보유하는 작업에는 두 가지 의미가 포함됩니다. 첫째, 테이블 잠금 또는 가져오기 목표 테이블에 대한 행 잠금을 보유 중인 다른 응용프로그램이 있는 경우에는, 그 응용프로그램 모두가 변경사항을 확약하거나 구간 복원할 때까지 가져오기 유틸리티가 기다립니다. 둘째, 가져오기를 수행 중인 동안에 잠금을 요청 중인 다른 응용프로그램은 가져오기 조작이 완료될 때까지 기다립니다.

IMPORT 명령

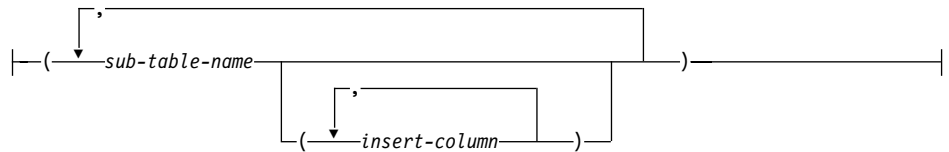
명령 구문



hierarchy description:



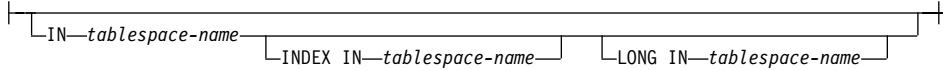
sub-table-list:



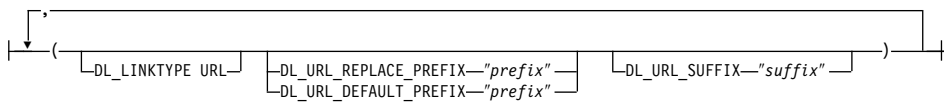
traversal-order-list:



tblspace-specs:



datalink-spec:



명령 매개변수

ALL TABLES

계층 구조 전용을 위한 내재된 키워드. 계층 구조를 가져올 때, 기본값은 트래버스 순서로 지정된 모든 테이블을 가져오는 것입니다.

AS ROOT TABLE

하나 이상의 서브테이블을 독립형 테이블 계층 구조로 작성합니다.

COMMITCOUNT n

모든 *n* 레코드가 가져오기된 후에 COMMIT를 수행합니다.

CREATE

테이블 정의와 행의 내용을 작성합니다. DB2 테이블, 서브테이블 또는 계층 구조로부터 데이터가 내보낸 경우, 색인이 작성됩니다. 이 옵션이 계층 구조에서 작동하고 데이터가 DB2로부터 내보낸 경우에는 유형 계층 구조도 작성됩니다. 이 옵션은 IXF 파일에서만 사용할 수 있습니다.

주: MVS 호스트 데이터베이스에서 내보낸 데이터에 페이지 크기에 산정된 길이가 254 미만인 LONGVAR 필드가 들어 있으면, 행이 너무

IMPORT 명령

길기 때문에 CREATE를 실패합니다. 이러한 경우, 테이블을 수작업으로 작성하고, IMPORT를 INSERT와 함께 호출하거나, 아니면 LOAD 명령을 사용해야 합니다.

DATALINK SPECIFICATION

각 DATALINK 컬럼에 대해서는 괄호로 묶은 하나의 컬럼 스펙을 가질 수 있습니다. 각 컬럼 스펙은 하나 이상의 DL_LINKTYPE, 접두부 및 DL_URL_SUFFIX 스펙으로 구성됩니다. 접두부 스펙은 DL_URL_REPLACE_PREFIX 또는 DL_URL_DEFAULT_PREFIX가 될 수 있습니다.

테이블에 정의된 DATALINK 컬럼 수 만큼의 DATALINK 컬럼 스펙이 있을 수 있습니다. 스펙의 순서는 *insert-column* 목록이나 테이블 정의 (*insert-column* 목록이 정의되어 있지 않을 경우) 내에 있는 DATALINK 컬럼의 순서를 따릅니다.

DL_LINKTYPE

지정된 경우, 컬럼 정의의 LINKTYPE과 일치해야 합니다. 그러므로, DL_LINKTYPE URL은 컬럼 정의가 LINKTYPE URL을 지정할 경우 허용됩니다.

DL_URL_DEFAULT_PREFIX "prefix"

지정된 경우, 같은 컬럼 내의 모든 DATALINK 값에 대한 기본 접두부의 역할을 해야 합니다. 이 경우, 접두부는 URL 스펙의 "스키마 호스트 포트" 부분을 나타냅니다. 분산 파일 시스템(DFS)의 경우, 접두부는 URL 스펙의 "scheme cellname filepath-join" 부분을 참조하십시오.

다음은 접두부의 예입니다.

```
"http://server"  
"file://server"  
"file:"  
"http://server:80"  
"dfs://.../cellname/fs"
```

컬럼의 데이터에 접두부가 들어 있지 않고 기본 접두부가 DL_URL_DEFAULT_PREFIX로 지정되면, 기본 접두부가 컬럼 값(NULL 이 아닐 경우) 앞에 놓입니다.

예를 들어, DL_URL_DEFAULT_PREFIX가 기본 접두부 "http://toronto"를 지정하면,

- 컬럼 입력 값 "/x/y/z"는 "http://toronto/x/y/z"로 저장됩니다.
- 컬럼 입력 값 "http://coyote/a/b/c"는 "http://coyote/a/b/c"로 저장됩니다.
- 컬럼 입력 값 널(NULL)은 널(NULL)로 저장됩니다.

DL_URL_REPLACE_PREFIX "prefix"

이 절은 사용자가 데이터에 있는 호스트 이름을 다른 호스트 이름으로 번역 바꿀 때, 이전에 내보내기 유틸리티에 의해 생성된 데이터를 로드하거나 가져오는 데 유용합니다. 이 접두부를 지정하면, 이는 널(NULL)이 아닌 모든 컬럼 값에 대한 접두부가 됩니다. 컬럼 값에 접두부가 있으면, 이 접두부를 바꿉니다. 컬럼 값에 접두부가 없으면, DL_URL_REPLACE_PREFIX에 의해 지정된 접두부가 이 컬럼 값 앞에 놓입니다. 분산 파일 시스템(DFS)의 경우, 접두부는 URL 스펙의 "scheme cellname filepath-junction" 부분을 참조하십시오.

예를 들면, DL_URL_REPLACE_PREFIX가 접두부 "http://toronto"를 지정하면,

- 컬럼 입력 값 "/x/y/z"는 "http://toronto/x/y/z"로 저장됩니다.
- 컬럼 입력 값 "http://coyote/a/b/c"는 "http://toronto/a/b/c"로 저장됩니다. "toronto"가 "coyote"를 바꾸는 점을 주지하십시오.
- 컬럼 입력 값 널(NULL)은 널(NULL)로 저장됩니다.

DL_URL_SUFFIX "suffix"

이 접미부를 지정하면, 컬럼에 대한 널(NULL)이 아닌 모든 컬럼 값에 첨부됩니다. 실제로는 DATALINK 값의 URL 부분의 "경로" 구성요소에 첨부됩니다.

FROM filename

가져올 데이터를 포함하는 파일을 지정합니다. 경로가 생략되면, 현재 작업 디렉토리를 사용합니다.

HIERARCHY

계층 구조적 데이터가 가져오는 것을 지정합니다.

IMPORT 명령

IN tablespace-name

테이블이 작성될 테이블 공간을 나타냅니다. 테이블 공간이 있어야 하며, REGULAR 테이블 공간이어야 합니다. 다른 테이블 공간이 지정되지 않으면, 모든 테이블 부분이 이 테이블 공간에 저장됩니다. 이 절이 지정되지 않을 경우, 테이블은 권한 부여 ID가 작성한 테이블 공간에 작성됩니다. 테이블 공간이 발견되지 않을 경우, 테이블은 기본 테이블 공간인 USERSPACE1에 작성됩니다. USERSPACE1이 삭제되면, 테이블 작성이 실패합니다.

INDEX IN tablespace-name

테이블상의 색인이 작성될 테이블 공간을 나타냅니다. 이 옵션은 IN절에 지정된 1차 테이블 공간이 DMS 테이블 공간일 때에만 허용됩니다. 지정된 테이블 공간이 있어야 하며, REGULAR DMS 테이블 공간이어야 합니다.

주: 어느 테이블 공간이 색인을 포함할지는 테이블이 작성될 때에만 지정할 수 있습니다.

insert-column

데이터가 삽입될 테이블이나 뷰의 컬럼 이름을 지정합니다.

INSERT

기존 테이블 데이터를 변경하지 않고 가져온 데이터를 테이블에 추가합니다.

INSERT_UPDATE

목표 테이블에 가져온 데이터의 행을 추가하거나 기본 키가 일치하는 기존 행(목표 테이블의)을 갱신합니다.

INTO table-name

데이터가 가져올 데이터베이스 테이블을 지정합니다. 이 테이블은 시스템 테이블, 임시 테이블 또는 요약 테이블이 될 수 없습니다.

완전 규정화되거나 규정화되지 않은 테이블 이름을 사용해야 하는 경우, 하위 레벨 서버의 경우를 제외하고는 INSERT, INSERT_UPDATE 또는

REPLACE에 대한 별명을 사용할 수 있습니다. 규정된 테이블 이름은 *schema.tablename* 형식입니다. *schema*는 테이블이 작성된 사용자 이름입니다.

LOBS FROM lob-path

LOB 파일을 저장하는 하나 이상의 경로를 지정합니다. LOB 데이터 파일의 이름은 주 데이터 파일(ASC, DEL 또는 IXF)에 저장되며, LOB 컬럼에 로드될 컬럼에 들어갑니다. 이 옵션은 *lobsinfile* 수정자가 지정되지 않은 경우에는 무시됩니다.

LONG IN tablespace-name

긴 컬럼(LONG VARCHAR, LONG VARGRAPHIC, LOB 데이터 유형 또는 이를 소스 유형으로 갖는 구별 유형)이 저장된 테이블 공간을 나타냅니다. 이 옵션은 IN절에 지정된 1차 테이블 공간이 DMS 테이블 공간일 때에만 허용됩니다. 테이블 공간이 있어야 하며, LONG DMS 테이블 공간이어야 합니다.

MESSAGES message-file

가져오기 조작중 발생하는 경고 및 오류 메시지의 목적지를 지정합니다. 파일이 이미 있으면, 가져오기 유틸리티가 정보를 추가합니다. 이 파일에 대한 완전 경로가 지정되어 있지 않을 경우, 이 유틸리티는 현재 디렉토리와 기본 드라이브를 목적지로 사용합니다. *message-file*이 생략되면, 메시지는 표준 출력에 작성됩니다.

METHOD

- | **L** 데이터를 가져올 시작 및 끝 컬럼 번호를 지정합니다. 컬럼 번호
- | 는 데이터의 행 시작에서의 바이트 옵셋입니다. 번호는 1부터 매
- | 겨집니다.
- | **주:** 이 방법은 ASC 파일에서만 사용할 수 있으며, 이 파일 유형
- | 에 대해 유일한 유효 옵션입니다.
- | **N** 가져올 컬럼 이름을 지정합니다.
- | **주:** 이 방법은 IXF 파일에서만 사용할 수 있습니다.

IMPORT 명령

P 가져올(import) 입력 데이터 필드의 색인(1부터 번호가 매겨짐)을 지정합니다.

주: 이 방법은 IXF 또는 DEL 파일에서만 사용할 수 있으며, DEL 파일 유형에 대해 유일한 유효 옵션입니다.

MODIFIED BY filetype-mod

추가 옵션을 지정합니다(67 페이지의 표5).

NULL INDICATORS null-indicator-list

이 옵션은 METHOD L 매개변수가 지정될 때 즉, 입력 파일이 ASC 파일일 때만 사용할 수 있습니다. 널(NULL) 표시기 목록은 각 널(NULL) 표시기 필드의 컬럼 번호를 지정하는 양의 정수 목록으로 쉼표로 구분됩니다. 컬럼 번호는 데이터의 행 시작으로부터 널(NULL) 표시기 필드의 바이트 옵셋입니다. METHOD L 매개변수에 정의된 각 데이터 필드에 대한 널(NULL) 표시기 목록에 반드시 하나의 항목이 있어야 합니다. 컬럼 번호 0은 해당 데이터 필드에 항상 데이터가 들어 있음을 나타냅니다.

널(NULL) 표시기 컬럼의 Y 값은 컬럼 데이터가 널(NULL)임을 지정합니다. 널(NULL) 표시기 컬럼에서 Y 이외의 다른 문자는 컬럼 데이터가 널(NULL)이 아니고 METHOD L 옵션에 의해 지정된 컬럼 데이터를 가져오는 것을 나타냅니다.

널(NULL) 표시기 문자는 MODIFIED BY 옵션을 사용하여 변경할 수 있습니다(145 페이지의 표8의 nullindchar 수정자에 대한 설명 참조).

OF filetype

입력 파일에 있는 데이터의 형식을 다음과 같이 지정합니다.

- ASC(컬럼 식별자가 없는 ASCII 형식)
- 다양한 데이터베이스 관리 프로그램 및 파일 관리 프로그램에서 사용되는 DEL(컬럼 식별자가 있는 ASCII 형식)
- 다음과 같은 프로그램에서 사용되는 WSF(워크시트 형식)
 - Lotus 1-2-3
 - Lotus Symphony

- 동일한 DB2 테이블 또는 다른 DB2 테이블에서 내보냈다는 것을 나타내는 IXF(Integrated Exchange Format, PC 버전). IXF 파일은 컬럼이 SELECT문에 지정된 경우를 제외하고는 기존의 모든 색인 정의와 테이블 정의도 포함합니다.

이 파일의 형식에 대해서는 239 페이지의 『부록C. 내보내기/가져오기/로드 유틸리티 파일 형식』에서 자세한 내용을 참조하십시오.

REPLACE

데이터 오브젝트를 절단하여 테이블에서 기존의 모든 데이터를 삭제한 후 가져온 데이터를 삽입합니다. 테이블 정의와 색인 정의는 변경되지 않습니다. 이 옵션은 테이블이 있을 경우에만 사용할 수 있습니다. DATALINK 컬럼이 있는 테이블에 대해서는 유효하지 않습니다. 계층 구조간에 데이터를 이동시킬 때 이 옵션을 사용한 경우, 각 서브테이블이 아닌 전체 계층 구조에 대한 데이터만 바꿀 수 있습니다.

REPLACE_CREATE

테이블이 이미 있으면, 데이터 오브젝트를 절단하여 테이블에서 기존의 모든 데이터를 삭제한 후 테이블 정의나 색인 정의를 변경하지 않고 가져온 데이터를 삽입합니다.

테이블이 존재하지 않으면, 행 내용과 마찬가지로 테이블 및 색인 정의를 작성합니다.

이 옵션은 IXF 파일에서만 사용할 수 있습니다. DATALINK 컬럼이 있는 테이블에 대해서는 유효하지 않습니다. 계층 구조간에 데이터를 이동시킬 때 이 옵션을 사용한 경우, 각 서브테이블이 아닌 전체 계층 구조에 대한 데이터만 바꿀 수 있습니다.

RESTARTCOUNT *n*

가져오기 조작이 레코드 *n* + 1에서 시작되는 것을 지정합니다. 첫번째 *n* 레코드는 생략됩니다.

STARTING *sub-table-name*

*sub-table-name*부터 시작하는 기본 순서를 요청하는, 계층 구조 전용 키워드. PC/IXF 파일의 경우, 기본 순서는 입력 파일에 저장된 순서입니다. PC/IXF 파일 형식에 대해서는 기본 순서만 유효합니다.

IMPORT 명령

sub-table-list

INSERT 또는 INSERT_UPDATE 옵션으로 입력된 테이블의 경우, 데이터가 가져올 서브테이블을 나타내기 위해 서브테이블 이름 목록이 사용됩니다.

traversal-order-list

INSERT, INSERT_UPDATE 또는 REPLACE 옵션으로 입력된 테이블의 경우, 계층 구조에 있는 서브테이블을 가져오는 트래버스 순서를 나타내기 위해 서브테이블 이름 목록이 사용됩니다.

UNDER sub-table-name

하나 이상의 서브테이블을 작성하기 위한 상위 테이블을 지정합니다.

가져오기 API

C API 구문

```
/* File: sqlutil.h */
/* API: Import */
/* ... */
SQL_API_RC SQL_API_FN
sqluimpr (
    char * pDataFileName,
    sqlu_media_list * pLobPathList,
    struct sqldcol * pDataDescriptor,
    struct sqlchar * pActionString,
    char * pFileType,
    struct sqlchar * pFileTypeMod,
    char * pMsgFileName,
    short CallerAction,
    struct sqluimpt_in* pImportInfoIn,
    struct sqluimpt_out* pImportInfoOut,
    sqlint32 * pNullIndicators,
    void * pReserved,
    struct sqlca * pSqlca);
/* ... */
```

일반 API 구문

```
/* File: sqlutil.h */
/* API: Import */
/* ... */
SQL_API_RC SQL_API_FN
sqlgimpr (
    unsigned short DataFileNameLen,
    unsigned short FileTypeLen,
    unsigned short MsgFileNameLen,
    char * pDataFileName,
    sqlu_media_list * pLobPathList,
    struct sqldcol * pDataDescriptor,
    struct sqlchar * pActionString,
    char * pFileType,
    struct sqlchar * pFileTypeMod,
    char * pMsgFileName,
    short CallerAction,
    struct sqluimpt_in* pImportInfoIn,
    struct sqluimpt_out* pImportInfoOut,
    sqlint32 * NullIndicators,
    void * pReserved,
    struct sqlca * pSqlca);
/* ... */
```

API 매개변수

DataFileNameLen

입력. 입력 파일 이름의 길이(바이트)를 나타내는 부호가 없는 2바이트 정수

FileTypeLen

입력. 입력 파일 유형의 길이(바이트)를 나타내는 부호가 없는 2바이트 정수

MsgFileNameLen

입력. 메시지 파일 이름의 길이(바이트)를 나타내는 부호가 없는 2바이트 정수

pDataFileName

입력. 데이터가 가져올 외부 입력 파일 이름과 경로를 포함하는 문자열

pLobPathList

입력. *media_type* SQLU_LOCAL_MEDIA를 사용하는 *sqlu_media_list* 및 LOB 파일을 찾을 수 있는 클라이언트의 경로를 나열하는 *sqlu_media_entry* 구조

*Administrative API Reference*에 있는 “SQLU-MEDIA-LIST”에서 자세한 내용을 참조하십시오.

pDataDescriptor

입력. 외부 파일로부터 가져오기를 위해 선택되는 컬럼에 대한 정보를 포함하는 *sqldcol* 구조에 대한 포인터. *dcolmeth* 필드의 값은 이 매개변수에 제공된 정보의 나머지가 가져오기 유틸리티에 의해 해석되는 방법을 결정합니다. 이 매개변수에 대해 유효한 값(*sqlutil*에 정의됨)은 다음과 같습니다.

SQL_METH_N

이름. 컬럼 이름에 의해 외부 입력 파일로부터 컬럼을 선택합니다.

SQL_METH_P

위치(position). 컬럼 위치에 의해 외부 입력 파일로부터 컬럼을 선택합니다.

SQL_METH_L

위치(location). 컬럼 위치에 의해 외부 입력 파일로부터 컬럼을 선택합니다. 데이터베이스 관리 프로그램은 다음과 같은 조건 때문에 유효하지 않은 위치 쌍을 갖는 가져오기 호출을 거절합니다.

- 시작 또는 끝나는 위치가 1에서부터 가장 큰 2바이트 정수(부호가 있는) 범위에 있지 않음
- 끝나는 위치가 시작 위치보다 작음
- 위치 쌍으로 정의된 입력 컬럼 폭이 목표 컬럼의 유형 및 길이와 호환되지 않음

두 위치가 모두 0인 위치 쌍은 널(NULL) 입력 가능 컬럼이 널(NULL)로 채워짐을 나타냅니다.

SQL_METH_D

기본값. *pDataDescriptor*가 널(NULL)이거나 *SQL_METH_D*로 설정

정되어 있으면, 외부 입력 파일로부터 컬럼 선택이 수행된 것입니다. 이러한 경우, 컬럼의 수와 컬럼 스펙 배열은 모두 무시됩니다. 외부 입력 파일에 있는 데이터의 첫번째 n 컬럼은 컬럼의 원래 순서대로 선택된 것이며, n 은 데이터가 가져올 데이터베이스 컬럼의 수를 나타냅니다.

*Administrative API Reference*에 있는 “SQLDCOL”에서 자세한 내용을 참조하십시오.

pActionString

입력. 데이터가 가져올 컬럼을 식별하는 문자 배열 앞에 오는 2바이트 길이의 필드가 들어 있는 *sqlchar* 구조에 대한 포인터

이 문자 배열은 다음과 같은 형식으로 되어 있습니다.

```
{INSERT | INSERT UPDATE | REPLACE | CREATE | REPLACE_CREATE}
INTO {tname[(tcolumn-list)] |
[{{ALL TABLES | (tname[(tcolumn-list)][, tname[(tcolumn-list)]])}]}}
[IN] HIERARCHY {STARTING tname | (tname[, tname])}
[UNDER sub-table-name | AS ROOT TABLE]}
[ATALINK SPECIFICATION datalink-spec]
```

INSERT

기존 테이블 데이터를 변경하지 않고 가져온 데이터를 테이블에 추가합니다.

INSERT_UPDATE

가져오기된 행의 기본 키 값이 테이블에 없을 경우 가져온 행을 추가하고, 기본 키 값이 발견되면 이 행으로 갱신합니다. 이 옵션은 목표 테이블에 기본 키가 있고, 가져오기되는 목표 컬럼의 지정(또는 암시) 목록에 기본 키에 대한 모든 컬럼이 있을 경우에만 유효합니다. 이 옵션은 뷰에는 적용할 수 없습니다.

REPLACE

테이블 오브젝트를 절단하여 테이블로부터 모든 기존 데이터를 삭제한 후, 가져온 데이터를 삽입합니다. 테이블 정의와 색인 정의는 변경되지 않습니다. (*indexixf*가 *FileTypeMod*에 있고 *FileType*이 *SQL_IXF*이면, 색인이 삭제된 후 바뀔 것입니다). 테이블이 이미 정의되어 있지 않으면 오류가 리턴됩니다.

경고: 기존 데이터가 삭제된 후에 오류가 발생하면 데이터가 유실됩니다.

CREATE

지정된 테이블이 정의되어 있지 않을 경우, 지정된 PC/IXF 파일 정보를 사용하여 테이블 정의와 행 내용을 작성합니다. 파일이 DB2에 의해 이미 내보낸 경우에는 색인도 작성됩니다. 지정된 테이블이 이미 정의되어 있으면 오류가 리턴됩니다. 이 옵션은 PC/IXF 파일 형식에 대해서만 유효합니다.

REPLACE_CREATE

지정된 테이블이 정의되어 있는 경우, PC/IXF 파일의 PC/IXF 행 정보를 사용하여 테이블 내용을 바꿉니다. 테이블이 이미 정의되어 있지 않으면, 지정된 PC/IXF 파일의 정보를 사용하여 테이블 정의와 행 내용이 작성됩니다. PC/IXF 파일이 DB2에 의해 이미 내보낸 경우에는 색인도 작성됩니다. 이 옵션은 PC/IXF 파일 형식에 대해서만 유효합니다.

경고: 기존 데이터가 삭제된 후에 오류가 발생하면 데이터가 유실됩니다.

tname 데이터가 삽입될 테이블, 입력된 테이블, 뷰 또는 오브젝트 뷰의 이름. 규정화되거나 규정화되지 않은 이름이 지정되어야 하는 경우, 하위 레벨 서버의 경우를 제외하고는 REPLACE, INSERT_UPDATE 또는 INSERT에 대한 별명이 지정될 수 있습니다. 뷰일 경우, 읽기 전용 뷰는 될 수 없습니다.

tcolumn-list

데이터가 삽입될 테이블 또는 뷰 컬럼 이름의 목록. 컬럼 이름은 쉼표로 구분되어야 합니다. 컬럼 이름을 지정하지 않으면, CREATE TABLE이나 ALTER TABLE문에 정의된 컬럼 이름이 사용됩니다. 입력된 테이블에 대해 컬럼 목록이 지정되어 있지 않으면, 데이터는 각 서브테이블에 있는 모든 컬럼에 삽입됩니다.

sub-table-name

CREATE 옵션으로 하나 이상의 서브테이블을 작성할 때 상위 테이블을 지정합니다.

ALL TABLES

계층 구조 전용을 위한 내재된 키워드. 계층 구조를 가져올 때, 기본값은 *traversal-order-list*에 지정된 모든 테이블을 가져오는 것입니다.

HIERARCHY

계층 구조적 데이터가 가져오는 것을 지정합니다.

STARTING

계층 구조 전용 키워드. 지정된 서브테이블 이름에서 시작하여 기본 순서가 사용되는 것을 지정합니다.

UNDER

계층 구조 및 CREATE 전용 키워드. 새로운 계층 구조, 부속 계층 구조 또는 서브테이블이 지정된 서브테이블 밑에 작성되는 것을 지정합니다.

AS ROOT TABLE

계층 구조 및 CREATE 전용 키워드. 새로운 계층 구조, 부속 계층 구조 또는 서브테이블이 독립형 계층 구조로 작성되는 것을 지정합니다.

DATALINK SPECIFICATION *datalink-spec*

DB2 데이터 링크에 관한 매개변수를 지정합니다. 이 매개변수는 IMPORT 명령에서와 같은 구문을 사용하여 지정할 수 있습니다(42 페이지의 『IMPORT 명령』 참조).

*tname*과 *tcolumn-list* 매개변수는 SQL INSERT문의 *tablename*과 *colname* 목록에 상응하며, 동일한 제한사항을 가집니다.

*tcolumn-list*의 컬럼과 외부 컬럼(지정되거나 암시된)은 목록이나 구조에서 해당 위치가 일치합니다(*sqldcol* 구조에 지정된 첫번째 컬럼의 데이터가 *tcolumn-list*의 첫번째 요소에 해당하는 테이블이나 뷰 필드로 삽입됩니다).

동일하지 않은 컬럼 수가 지정될 경우, 실제 처리된 컬럼 수는 두 컬럼 수 중 적은 쪽입니다. 이는 오류(널(NULL)을 입력할 수 없는 테이블 필드에 널을 값이 없으므로) 또는 정보용 메시지(여러 외부 파일 컬럼이 무시되었으므로) 결과가 나타날 수 있습니다.

*Administrative API Reference*에 있는 “SQLCHAR”에서 자세한 내용을 참조하십시오.

pFileType

입력. 외부 파일 내의 데이터 형식을 나타내는 문자열. 지원되는 외부 파일 형식(sqlutil에 정의됨)은 다음과 같습니다.

SQL_ASC

컬럼 식별자가 없는 ASCII

SQL_DEL

dBase, BASIC 및 IBM Personal Decision Series 프로그램과 기타 여러 데이터베이스 관리 프로그램 및 파일 관리 프로그램과의 교환을 위한 컬럼 식별자가 있는 ASCII

SQL_IXF

IXF(Integrated Exchange Format)의 PC 버전. 테이블로부터 데이터를 내보내기 위한 우선적 방법이며, 나중에 같은 테이블이나 다른 데이터베이스 관리 프로그램 테이블에 가져올 수 있습니다.

SQL_WSF

Lotus Symphony 및 1-2-3 프로그램과 교환을 위한 워크시트 형식

이 파일의 형식에 대해서는 239 페이지의 『부록C. 내보내기/가져오기/로드 유틸리티 파일 형식』에서 자세한 내용을 참조하십시오.

pFileTypeMod

입력. 하나 이상의 처리 옵션을 지정하는 문자 배열 앞에 오는 2바이트 길이의 필드가 들어 있는 구조에 대한 포인터. 이 포인터가 널(NULL)이거나 지정된 구조에 0 문자가 있으면, 이 조치는 기본 스펙 선택으로 해석됩니다.

모든 옵션이 지원되는 모든 파일 유형에 사용될 수는 없습니다.

*Administrative API Reference*에 있는 “SQLCHAR” 및 67 페이지의 『파일 유형 수정자(가져오기)』에서 자세한 내용을 참조하십시오.

pMsgFileName

입력. 유틸리티에 의해 리턴된 오류, 경고 및 정보용 메시지에 대한 목적지를 포함하는 문자열. 운영 체제 파일이나 표준 장치의 경로 및 이름이 될 수 있습니다. 파일이 이미 있으면, 첨부됩니다. 파일이 없으면, 파일이 작성됩니다.

CallerAction

입력. 호출자가 요청한 조치. 유효한 값(sqlutil에 정의됨)은 다음과 같습니다.

SQLU_INITIAL

초기 호출. 이 값은 API에 대한 첫번째 호출에 사용되어야 합니다.

초기 호출이나 이후 계속된 다른 호출이 리턴하여 호출 응용프로그램이 요청된 내보내기 조작을 완료하기 전에 일부 조치를 수행하도록 요구할 경우, 호출자 조치는 다음 중 하나로 설정되어야 합니다.

SQLU_CONTINUE

계속 처리. 이 값은 초기 호출이 사용자 입력을 요청하는 유틸리티(예: 테이프가 종료된 상태에 응답하도록) 상태와 함께 리턴한 후, API에 대한 계속된 호출에만 사용될 수 있습니다. 이는 유틸리티가 요청한 사용자 조치가 완료되고, 유틸리티가 초기 요청을 계속 처리할 수 있다는 것을 지정합니다.

SQLU_TERMINATE

처리 종료. 이 값은 초기 호출이 사용자 입력을 요청하는 유틸리티(예: 테이프가 종료된 상태에 응답하도록) 상태와 함께 리턴한 후, API에 대한 계속된 호출에만 사용될 수 있습니다. 이는 유틸리티가 요청한 사용자 조치가 수행되지 않고, 유틸리티가 초기 요청 처리를 종료하려 함을 지정합니다.

pImportInfoIn

입력. 추가 입력 매개변수를 포함하는 *sqluimpt_in* 구조에 대한 선택적 포인터. 64 페이지의 『SQLUIMPT-IN 데이터 구조』에서 이 구조에 관한 정보를 참조하십시오.

pImportInfoOut

출력. 추가 출력 매개변수를 포함하는 *sqluimpt_out* 구조에 대한 선택적 포인터. 65 페이지의 『SQLUIMPT-OUT 데이터 구조』에서 이 구조에 관한 정보를 참조하십시오.

NullIndicators

입력. ASC 파일 전용. 컬럼 데이터가 널(NULL) 입력 가능한지의 여부를 나타내는 정수 배열. 이 배열에 있는 요소의 수는 입력 파일의 컬럼 수와 일치해야 합니다. 즉, 이 배열의 요소와 데이터 파일로부터 가져오기되는 컬럼이 1대1로 대응됩니다. 따라서, 요소의 수는 *pDataDescriptor* 매개변수의 *dcolnum* 필드와 동일해야 합니다. 각 배열 요소에는 널(NULL) 표시 필드로 사용될 데이터 파일에 있는 컬럼을 식별하는 숫자를 포함하거나 테이블 컬럼이 널(NULL) 입력 불가능이라는 것을 나타내는 0을 포함합니다. 요소가 0이 아닐 경우, 데이터 파일에서 식별된 컬럼은 Y 또는 N을 포함해야 합니다. Y는 테이블 컬럼 데이터가 널(NULL)임을 나타내고, N은 테이블 컬럼 데이터가 널(NULL)이 아님을 나타냅니다.

pReserved

향후 사용을 위해 예약됨.

pSqlca

출력. *sqlca* 구조에 대한 포인터. 이 구조에 대해서는 *Administrative API Reference*에 있는 “SQLCA”에서 자세한 내용을 참조하십시오.

REXX API 구문

```
IMPORT FROM datafile OF filetype  
[MODIFIED BY :filetmod]  
[METHOD {L|N|P} USING :dcoldata]  
[COMMITCOUNT :commitcnt] [RESTARTCOUNT :restartcnt]  
MESSAGES msgfile  
{INSERT|REPLACE|CREATE|INSERT_UPDATE|REPLACE_CREATE}  
INTO tname [(:columns)]  
[OUTPUT INTO :output]  
  
CONTINUE IMPORT  
  
STOP IMPORT
```

REXX API 매개변수

datafile

데이터가 가져올 파일 이름

filetype

외부 가져오기 파일의 데이터 형식. 지원되는 파일 형식은 다음과 같습니다.

DEL 컬럼 식별자가 있는 ASCII

ASC 컬럼 식별자가 없는 ASCII

WSF 워크시트 형식

IXF 통합 교환 형식(IXF: Integrated Exchange Format)의 PC 버전

filetmod

추가 처리 옵션을 포함하는 호스트 변수(67 페이지의 『파일 유형 수정자(가져오기)』 참조)

LINIP 외부 입력 파일의 컬럼을 선택하는 데 사용될 방법을 지정하는 문자. 유효한 값은 다음과 같습니다.

L 위치(Location)

N 이름

P 위치(Position)**dcoldata**

외부 파일로부터 가져오기를 위해 선택된 컬럼에 대한 정보를 포함하는 복합 REXX 호스트 변수. 이 구조의 내용은 지정된 방법에 따라 다릅니다. 다음에서, XXX는 호스트 변수 이름을 나타냅니다.

- 위치(location) 방법

XXX.0

변수의 나머지에 있는 요소의 수

XXX.1

입력 파일에서 이 컬럼의 시작 위치를 나타내는 수. 이 컬럼은 데이터베이스 테이블에서 첫번째 컬럼이 됩니다.

XXX.2

컬럼의 끝나는 위치를 나타내는 수

XXX.3

입력 파일에서 이 컬럼의 시작 위치를 나타내는 수. 이 컬럼은 데이터베이스 테이블에서 두번째 컬럼이 됩니다.

XXX.4

컬럼의 끝나는 위치를 나타내는 수

XXX.5

계속

- 이름 방법

XXX.0

호스트 변수에 있는 컬럼 이름의 수

XXX.1

첫번째 이름

XXX.2

두번째 이름

XXX.3

계속

가져오기 API

- 위치(position) 방법

XXX.0

호스트 변수에 있는 컬럼 위치(position)의 수

XXX.1

외부 입력 파일의 컬럼 위치

XXX.2

외부 입력 파일의 컬럼 위치

XXX.3

계속

tname 목표 테이블이나 뷰의 이름. 데이터는 일기 전용 뷰에 가져올 수 없습니다.

columns

데이터가 삽입될 테이블이나 뷰의 컬럼 이름을 포함하는 REXX 호스트 변수. 다음에서, XXX는 호스트 변수 이름을 나타냅니다.

XXX.0

컬럼의 수

XXX.1

첫번째 컬럼 이름

XXX.2

두번째 컬럼 이름

XXX.3

계속

msgfile

오류 및 경고 메시지가 송신될 파일, 경로 또는 장치 이름

commitcnt

모든 *commitcnt* 레코드가 가져오기된 후에 COMMIT를 수행합니다.

restartcnt

가져오기 조작이 레코드 *restartcnt* + 1에서 시작되는 것을 지정합니다. 첫 번째 *restartcnt* 레코드는 생략됩니다.

output

가져오기 조작으로부터 정보가 전달되는 복합 REXX 호스트 변수. 다음에서, XXX는 호스트 변수 이름을 나타냅니다.

XXX.1

가져오기 조작중 외부 입력 파일에서 읽은 레코드의 수

XXX.2

삽입 또는 갱신을 시작하기 전에 건너뛴 레코드의 수

XXX.3

목표 테이블로 삽입된 행의 수

XXX.4

가져오기된 레코드의 정보로 갱신된 목표 테이블의 행 수

XXX.5

가져올 수 없는 레코드의 수

XXX.6

삽입, 갱신, 생략 및 거부된 행을 포함하여, 데이터베이스에 가져오기되어 확약된 레코드의 수

SQLUIMPT-IN 데이터 구조

이 구조는 51 페이지의 『가져오기 API』로부터 데이터를 전달하는 데 사용됩니다.

표 3. SQLUIMPT-IN 구조의 필드

필드 이름	데이터 유형	설명
SIZEOFSTRUCT	INTEGER	이 구조의 크기(바이트)
COMMITCNT	INTEGER	데이터베이스에 확약하기 전에 가져오기할 레코드의 수. <i>commitcnt</i> 레코드가 가져올 때마다 COMMIT가 수행됩니다.
RESTARTCNT	INTEGER	레코드 삽입 또는 갱신을 시작하기 전에 건너뛴 레코드의 수. 이 매개변수는 일부 레코드가 데이터베이스에 확약된 후, 레코드를 가져오려는 이전 시도가 실패할 경우 사용되어야 합니다. 지정된 값은 다음 가져오기 조작에 대한 시작점을 나타냅니다.

언어 구문

C 구조

```

/* File: sqlutil.h */
/* Structure: SQLUIMPT-IN */
/* ... */
SQL_STRUCTURE sqluimpt_in
{
    sqluint32      sizeofStruct;
    sqluint32      commitcnt;
    sqluint32      restartcnt;
};
/* ... */
    
```

COBOL 구조

```

* File: sqlutil.cbl
01 SQL-UIMPT-IN.
   05 SQL-SIZE-OF-UIMPT-IN    PIC 9(9) COMP-5 VALUE 12.
   05 SQL-COMMITCNT          PIC 9(9) COMP-5 VALUE 0.
   05 SQL-RESTARTCNT        PIC 9(9) COMP-5 VALUE 0.
*
    
```

SQLUIMPT-OUT 데이터 구조

이 구조는 51 페이지의 『가져오기 API』로부터 데이터를 전달하는 데 사용됩니다.

표 4. SQLUIMPT-OUT 구조의 필드

필드 이름	데이터 유형	설명
SIZEOFSTRUCT	INTEGER	이 구조의 크기(바이트)
ROWSREAD	INTEGER	가져오기중 파일에서 읽은 레코드의 수
ROWSSKIPPED	INTEGER	삽입 또는 갱신을 시작하기 전에 건너뛴 레코드의 수
ROWSINSERTED	INTEGER	목표 테이블로 삽입된 행의 수
ROWSUPDATED	INTEGER	가져오기된 레코드(기본 키 값이 이미 테이블에 있는 레코드)의 정보로 갱신된 목표 테이블의 행의 수
ROWSREJECTED	INTEGER	가져올 수 없는 레코드의 수
ROWSCOMMITTED	INTEGER	데이터베이스에 가져오기되어 확정된 레코드의 수

언어 구문

C 구조

```

/* File: sqlutil.h */
/* Structure: SQLUIMPT-OUT */
/* ... */
SQL_STRUCTURE sqluimpt_out
{
    sqluint32    sizeOfStruct;
    sqluint32    rowsRead;
    sqluint32    rowsSkipped;
    sqluint32    rowsInserted;
    sqluint32    rowsUpdated;
    sqluint32    rowsRejected;
    sqluint32    rowsCommitted;
};
/* ... */

```

SQLUIMPT-OUT 데이터 구조

COBOL 구조

```
* File: sqlutil.cbl
01 SQL-UIMPT-OUT.
   05 SQL-SIZE-OF-UIMPT-OUT PIC 9(9) COMP-5 VALUE 28.
   05 SQL-ROWSREAD          PIC 9(9) COMP-5 VALUE 0.
   05 SQL-ROWSSKIPPED      PIC 9(9) COMP-5 VALUE 0.
   05 SQL-ROWSINSERTED    PIC 9(9) COMP-5 VALUE 0.
   05 SQL-ROWSUPDATED     PIC 9(9) COMP-5 VALUE 0.
   05 SQL-ROWSREJECTED    PIC 9(9) COMP-5 VALUE 0.
   05 SQL-ROWSCOMMITTED   PIC 9(9) COMP-5 VALUE 0.
*
```

파일 유형 수정자(가져오기)

표 5. 유효한 파일 유형 수정자(가져오기)

수정자	설명
모든 파일 형식	
compound= <i>x</i>	<p><i>x</i>는 1부터 100까지의 숫자입니다. 비최소단위 복합 SQL을 사용하여 데이터를 삽입하십시오. <i>x</i>문이 매시간 시도됩니다.</p> <p>이 수정자가 지정되고 트랜잭션 로그 크기가 충분치 않으면 가져오기 작업에 실패하게 됩니다. 트랜잭션 로그는 COMMITCOUNT에 의해 지정된 행수 또는 COMMITCOUNT가 지정되지 않은 경우에는 데이터 파일의 행수를 수용할 정도로 충분히 커야 합니다. 그러므로 트랜잭션 로그 오버플로우를 피하려면 COMMITCOUNT 옵션을 지정하는 것이 바람직합니다.</p> <p>이 수정자는 INSERT_UPDATE 모드, 계층 테이블 및 다음과 같은 수정자 즉, usedefaults, identitymissing, identityignore, generatedmissing 및 generatedignore와는 호환이 되지 않습니다.</p>
generatedignore	이 수정자는 생성된 모든 컬럼에 대한 데이터가 데이터 파일에 존재하지만 이를 무시하도록 가져오기 유틸리티에 통지합니다. 그 결과 생성된 컬럼에 대한 모든 값이 유틸리티에 의해 생성됩니다. 이 수정자는 generatedmissing 수정자와 함께 사용할 수 없습니다.
generatedmissing	이 수정자가 지정되면 유틸리티는 입력 데이터 파일에 생성된 컬럼에 대한 데이터가 없다고(널(NULL) 까지도) 가정하고 각 행에 대해 값을 생성합니다. 이 수정자는 generatedignore 수정자와 함께 사용할 수 없습니다.
identityignore	이 수정자는 식별 컬럼에 대한 데이터가 데이터 파일에 존재하지만 이를 무시하도록 가져오기 유틸리티에 통지합니다. 그 결과 모든 식별값이 유틸리티에 의해 생성됩니다. 작동은 GENERATED ALWAYS 및 GENERATED BY DEFAULT 식별 컬럼 둘다에 대해 동일합니다. 이는 GENERATED ALWAYS 컬럼의 경우에 아무 행도 거부되지 않음을 의미합니다. 이 수정자는 identitymissing 수정자와 함께 사용할 수 없습니다.
identitymissing	이 수정자가 지정되면 유틸리티는 입력 데이터 파일에 식별 컬럼에 대한 데이터가 없다고(널(NULL) 까지도) 가정하고 각 행에 대해 값을 생성합니다. 작동은 GENERATED ALWAYS 및 GENERATED BY DEFAULT 식별 컬럼 둘다에 대해 동일합니다. 이 수정자는 identityignore 수정자와 함께 사용할 수 없습니다.
lobsinfile	<i>lob-path</i> 는 LOB 값을 포함하는 파일에 대한 경로를 지정합니다.

파일 유형 수정자(가져오기)

표 5. 유효한 파일 유형 수정자(가져오기) (계속)

수정자	설명
no_type_id	단일 서브테이블에 가져올 때에만 유효합니다. 일반적인 사용법은 정규 테이블로부터 데이터를 내보낸 다음, 단일 서브테이블로 데이터를 변환하기 위해 가져오기 조작을 호출하는 것입니다(이 수정자를 사용하여).
nodefaults	<p>목표 테이블 컬럼에 대한 소스 컬럼이 명시적으로 지정되지 않고, 테이블 컬럼이 널(NULL) 입력 기능이 아니면, 기본값이 로드되지 않습니다. 이 옵션을 사용하지 않으면, 목표 테이블 컬럼 중 하나에 대한 소스 컬럼이 명시적으로 지정되지 않을 경우, 다음과 같은 경우가 발생합니다.</p> <ul style="list-style-type: none"> • 컬럼에 대해 기본값이 지정될 수 있으면 기본값이 로드됩니다. • 컬럼이 널(NULL) 입력이 가능하지만 해당 컬럼에 대해 기본값이 지정될 수 없으면 NULL이 로드됩니다. • 컬럼이 널(NULL) 입력이 가능하고 기본값이 지정될 수 없으면, 오류가 리턴되고 유틸리티가 처리를 중지합니다.
usedefaults	<p>목표 테이블 컬럼에 대한 소스 컬럼이 지정되어 있지만, 하나 이상의 행 인스턴스에 대한 데이터가 들어 있지 않으면, 기본값이 로드됩니다. 다음은 데이터 생략의 예입니다.</p> <ul style="list-style-type: none"> • DEL 파일의 경우, 컬럼에 대해 ",,"가 지정됩니다. • ASC 파일의 경우, 컬럼에 대해 널(NULL) 표시기가 예로 설정됩니다. • DEL/ASC/WSF 파일의 경우, 컬럼을 충분히 가지고 있지 않거나 원래 스펙에 대해 길이가 충분하지 않은 행 <p>이 옵션을 사용하지 않으면, 소스 컬럼에 행 인스턴스에 대한 데이터가 없을 경우, 다음과 같은 경우가 발생합니다.</p> <ul style="list-style-type: none"> • 컬럼이 널(NULL) 입력 가능하면 널(NULL)이 로드됩니다. • 컬럼이 널(NULL) 입력 불가능하면 유틸리티는 행을 거절합니다.
ASCII 파일 형식(ASC/DEL)	

표 5. 유효한 파일 유형 수정자(가져오기) (계속)

수정자	설명
dateformat="x"	<p>x는 소스 파일에서의 날짜 형식입니다. ^a 유효한 날짜 요소는 다음과 같습니다.</p> <p>YYYY - 년 (0000 - 9999 사이의 4자리수) M - 월 (1 - 12 사이의 한자리 또는 두자리수) MM - 월 (1 - 12 사이의 두자리수, M과는 상호 배타적임) D - 일 (1 - 31 사이의 한자리 또는 두자리수) DD - 일 (1 - 31 사이의 두자리수, D와는 상호 배타적임) DDD - 년도의 일(001 - 366 사이의 세자리수, 다른 일 또는 달 요소와는 상호 배타적임)</p> <p>지정되지 않은 각 요소에 대해 기본값 1이 할당됩니다. 다음은 날짜 형식의 일부 예입니다.</p> <p>"D-M-YYYY" "MM.DD.YYYY" "YYYYDDD"</p>
implieddecimal	<p>내재된 소수점의 위치는 컬럼 정의에 의해 결정됩니다. 그리고 더 이상 값의 끝부분에 있다고 가정되지 않습니다. 예를 들어, 값 12345는 DECIMAL(8,2) 컬럼에 12345.00이 아니라 123.45로 로드됩니다.</p>
noeofchar	<p>선택적인 파일 끝 문자 x'1A'는 파일의 끝으로 인식되지 않습니다. 마치 일반적인 문자인 것처럼 처리가 계속 진행됩니다.</p>

파일 유형 수정자(가져오기)

표 5. 유효한 파일 유형 수정자(가져오기) (계속)

수정자	설명
timeformat="x"	<p>x는 소스 파일에서의 시간 형식입니다. ^a 유효한 시간 요소는 다음과 같습니다.</p> <ul style="list-style-type: none"> H - 시 (12시간 시스템 경우에는 0 - 12 사이, 24시간 시스템 경우에는 0 - 24 사이의 한자리 또는 두자리수) HH - 시 (12시간 시스템 경우에는 0 - 12 사이, 24시간 시스템 경우에는 0 - 24 사이의 두자리수, H와는 상호 배타적임) M - 분 (0 - 59 사이의 한자리 또는 두자리수) MM - 분 (0 - 59 사이의 두자리수, M과는 상호 배타적임) S - 초 (0 - 59 사이의 한자리 또는 두자리수) SS - 초 (0 - 59 사이의 두자리수, S와는 상호 배타적임) SSSSS - 한밤중 이후의 초단위 낮 시간 (00000 - 86399 사이의 5자리수, 다른 시간 요소와는 상호 배타적임) TT - Meridian 표시기(AM 또는 PM) <p>지정되지 않은 각 요소에 대해 기본값 0이 할당됩니다. 시간 형식의 일부 예는 다음과 같습니다.</p> <p>"HH:MM:SS" "HH.MM TT" "SSSSS"</p>

표 5. 유효한 파일 유형 수정자(가져오기) (계속)

수정자	설명
timestampformat="x"	<p>x는 소스 파일에서의 시간소인 형식입니다. ^a 유효한 시간소인 요소는 다음과 같습니다.</p> <p>YYYY - 년 (0000 - 9999 사이의 4자리수) M - 월 (1 - 12 사이의 한자리 또는 두자리수) MM - 월 (1 - 12 사이의 두자리수, M(월)과는 상호 배타적임) D - 일 (1 - 31 사이의 한자리 또는 두자리수) DD - 일 (1 - 31 사이의 두자리수, D와는 상호 배타적임) DDD - 년도의 일(001 - 366 사이의 세자리수, 다른 일 또는 월 요소와는 상호 배타적임) H - 시 (12시간 시스템 경우에는 0 - 12 사이, 24시간 시스템 경우에는 0 - 24 사이의 한자리 또는 두자리수) HH - 시 (12시간 시스템 경우에는 0 - 12 사이, 24시간 시스템 경우에는 0 - 24 사이의 두자리수, H와는 상호 배타적임) M - 분 (0 - 59 사이의 한자리 또는 두자리수) MM - 분 (0 - 59 사이의 두자리수, M(분)과는 상호 배타적임) S - 초 (0 - 59 사이의 한자리 또는 두자리수) SS - 초 (0 - 59 사이의 두자리수, S와는 상호 배타적임) SSSSS - 한밤중 이후의 초단위 낮 시간 (00000 - 86399 사이의 5자리수, 다른 시간 요소와는 상호 배타적임) UUUUUU - 밀리초(000000 - 999999 사이의 6자리수) TT - Meridian 표시기(AM 또는 PM)</p> <p>지정되지 않은 YYYY, M, MM, D, DD 또는 DDD 요소에 대해 기본값 1이 할당됩니다. 지정되지 않은 다른 모든 요소에 대해 기본값 0이 할당됩니다. 다음은 시간소인 형식의 예입니다.</p> <p>"YYYY/MM/DD HH:MM:SS.UUUUUU"</p> <p>다음 예에서는 사용자 정의 날짜 및 시간 형식이 들어 있는 데이터를 schedule이라는 테이블로 가져오는 방법에 대해 설명합니다.</p> <pre>db2 import from delfile2 of del modified by timestampformat="yyyymm.dd hh:mm tt" insert into schedule</pre>
ASC(컬럼 식별자가 없는 ASCII) 파일 형식	

파일 유형 수정자(가져오기)

표 5. 유효한 파일 유형 수정자(가져오기) (계속)

수정자	설명
nochecklengths	nochecklengths를 지정하면, 소스 데이터에 목표 테이블 컬럼의 크기를 초과하는 컬럼 정의가 들어 있더라도 각 행을 가져오려고 시도합니다. 이러한 행은 코드 페이지 변환으로 소스 데이터가 축소될 경우, 올바르게 가져올 수 있습니다. 예를 들어, 소스의 4바이트 EUC 데이터는 목표의 2바이트 DBCS 데이터로 축소할 수 있으며, 필요한 공간을 반으로 줄입니다. 이 옵션은 소스 데이터가 일치되지 않는 컬럼 정의에도 불구하고 모든 경우에 적합할 때 특히 유용합니다.
nullindchar=x	x는 단일 문자입니다. 널(NULL) 값을 표시하는 문자를 x로 변경합니다. x의 기본값은 Y입니다. ^b 이 수정자는 문자가 영문자인 경우를 제외하고는 EBCDIC 데이터 파일에 대해 대소문자가 구별됩니다. 예를 들어, 널(NULL) 표시기 문자가 문자 N이 되게 지정되면, n도 널(NULL) 표시기로 인식됩니다.
reclen=x	x는 32 767의 최대값을 갖는 정수입니다. 각 행에 대해 x 문자를 읽으며, 행의 끝을 나타내는 데 새로운 행 문자가 사용되지 않습니다.
striptblanks	가변 길이 필드에 데이터를 로드할 때 뒤 공백 공간을 모두 절단합니다. 이 옵션이 지정되지 않으면, 공백 공간이 보존됩니다. 다음 예에서 striptblanks는 가져오기 유틸리티가 뒤 공백 공간을 절단하게 합니다. <pre>db2 import from myfile.asc of asc modified by striptblanks method l (1 10, 12 15) messages msgs.txt insert into staff</pre> 이 옵션은 striptnulls와 함께 지정할 수 없습니다. 이는 상호 배타적인 옵션입니다. 주: 이 옵션은 백 레벨 호환성에 대해서만 지원되는 사용되지 않는 t 옵션을 바꿉니다.
striptnulls	가변 길이 필드에 데이터를 로드할 때 뒤 공백 널(NULL)(0x00 문자)을 모두 절단합니다. 이 옵션이 지정되지 않으면, 널(NULL)이 보존됩니다. 이 옵션은 striptblanks와 함께 지정될 수 없습니다. 이는 상호 배타적인 옵션입니다. 주: 이 옵션은 백 레벨 호환성에 대해서만 지원되는 사용되지 않는 padwithzero 옵션을 바꿉니다.
DEL(컬럼 식별자가 있는 ASCII) 파일 형식	

표 5. 유효한 파일 유형 수정자(가져오기) (계속)

수정자	설명
chardelx	<p>x는 단일 문자열 분리 문자입니다. 기본값은 큰 따옴표(")입니다. 문자열을 묶기 위해 큰 따옴표 대신 지정된 문자가 사용됩니다.^{bc}</p> <p>작은 따옴표(')도 문자열 분리 문자로 지정될 수 있습니다. 다음 예에서 charde1''은 내보내기 유틸리티가 발견한 모든 작은 따옴표(')를 문자열 분리 문자로 해석하게 합니다.</p> <pre>db2 "import from myfile.del of del modified by charde1'" method p (1, 4) insert into staff (id, years)"</pre>
coldelx	<p>x는 단일 문자 컬럼 분리 문자입니다. 기본값은 쉼표(,)입니다. 컬럼의 끝을 표시하기 위해 쉼표 대신 지정된 문자가 사용됩니다.^{bc}</p> <p>다음 예에서 colde1;은 가져오기 유틸리티가 발견한 모든 세미콜론(;)을 컬럼 분리 문자로 해석하게 합니다.</p> <pre>db2 import from myfile.del of del modified by colde1; messages msgs.txt insert into staff</pre>
datesiso	날짜 형식. 모든 날짜 데이터 값이 ISO 형식으로 내보내도록 합니다.
decplusblank	더하기 부호 문자. 양수(십진수) 값 앞에 더하기 부호(+) 대신 공백을 두게 합니다. 기본 조치는 양수(십진수) 값 앞에 더하기 부호를 붙이는 것입니다.
decptx	<p>x는 마침표 대신 소수점 문자로 사용하는 단일 문자입니다. 기본값은 마침표(.)입니다. 소수점 문자로 마침표 대신 지정된 문자가 사용됩니다.^{bc}</p> <p>다음 예에서 decpt;는 가져오기 유틸리티가 발견한 모든 세미콜론(;)을 소수점으로 해석하게 합니다.</p> <pre>db2 "import from myfile.del of del modified by charde1'" decpt; messages msgs.txt insert into staff"</pre>

파일 유형 수정자(가져오기)

표 5. 유효한 파일 유형 수정자(가져오기) (계속)

수정자	설명
delprioritychar	<p>분리 문자에 대한 현재 기본 우선순위는 레코드 분리 문자, 문자 분리 문자, 컬럼 분리 문자순입니다. 이 수정자는 분리 문자 우선순위를 문자 분리 문자, 레코드 분리 문자, 컬럼 분리 문자순으로 전환하여 이전 우선순위를 사용하는 기존 응용프로그램을 보호합니다. 구문은 다음과 같습니다.</p> <pre>db2 import ... modified by delprioritychar ...</pre> <p>예를 들어, 다음과 같은 DEL 데이터 파일이 주어질 경우,</p> <pre>"Smith, Joshua",4000,34.98<row delimiter> "Vincent,<row delimiter>, is a manager", 4005,44.37<row delimiter></pre> <p>delprioritychar 수정자의 지정으로 이 데이터 파일에는 두 개의 행만 있게 됩니다. 두번째 <row delimiter>는 두번째 행의 첫번째 데이터 컬럼의 부분으로 해석되고, 첫번째와 세번째 <row delimiter>는 실제 레코드 분리 문자로서 해석됩니다. 이 수정자가 지정되지 않으면, 이 데이터 파일에는 각기 <row delimiter>로 구분된 세 개의 행이 있게 됩니다.</p>
dldelx	<p>x는 단일 문자 DATALINK 분리 문자입니다. 기본값은 세미콜론(:)입니다. DATALINK 값에 대한 필드간 분리자로서 세미콜론 대신 지정된 문자가 사용됩니다. 이는 DATALINK 값이 둘 이상의 부속값을 가질 수 있기 때문에 필요합니다.^{bc}</p> <p>주: x는 행, 컬럼 또는 문자열 분리 문자와 같은 문자로 지정되지 않아야 합니다.</p>
keepblanks	<p>이는 CHAR, VARCHAR, LONG VARCHAR 또는 CLOB 유형의 각 필드에 앞 공백 및 뒤 공백을 보존합니다. 이 옵션이 없으면 문자 분리문자 내부에 있지 않는 모든 앞 공백 및 뒤 공백들은 제거되며 모든 공백 필드에 대해 테이블에 NULL이 삽입됩니다.</p>
nodoubledel	<p>이중 문자 분리 문자의 인식을 억제합니다. 21 페이지의 『분리 문자 제한사항』에서 자세한 내용을 참조하십시오.</p>
IXF 파일 형식	
forcein	<p>유틸리티에게 코드 페이지가 일치하지 않더라도 데이터를 승인하고, 코드 페이지간의 변환을 억제하도록 지시합니다.</p> <p>데이터에 대한 길이가 충분한지 확인하기 위해 고정 길이 목표 필드를 점검합니다. nochecklengths가 지정되면, 점검이 수행되지 않고 각 행을 가져오려고 시도합니다.</p>

표 5. 유효한 파일 유형 수정자(가져오기) (계속)

수정자	설명
indexixf	유틸리티에게 이미 있는 테이블에 현재 정의되어 있는 모든 색인을 삭제하고 PC/IXF 파일에 있는 색인 정의로부터 새로운 색인을 작성하도록 지시합니다. 이 옵션은 테이블 내용이 바뀌는중일 때에만 사용할 수 있습니다. 이 옵션은 뷰에서는 사용할 수 없으며, <i>insert-column</i> 이 지정되어 있을 때에도 사용할 수 없습니다.
indexschema= <i>schema</i>	색인 작성시 색인 이름에 대해 지정된 <i>schema</i> 를 사용합니다. <i>schema</i> 가 지정되지 않으면(대신 <i>indexschema</i> 가 지정되어 있으면), 연결 사용자 ID를 사용합니다. 키워드가 지정되지 않으면, IXF 파일의 스키마를 사용합니다.
nochecklengths	<i>nochecklengths</i> 를 지정하면, 소스 데이터에 목표 테이블 컬럼의 크기를 초과하는 컬럼 정의가 들어 있더라도 각 행을 가져오려고 시도합니다. 이러한 행은 코드 페이지 변환으로 소스 데이터가 축소될 경우, 올바르게 가져올 수 있습니다. 예를 들어, 소스의 4바이트 EUC 데이터는 목표의 2바이트 DBCS 데이터로 축소할 수 있으며, 필요한 공간을 반으로 줄입니다. 이 옵션은 소스 데이터가 일치되지 않는 컬럼 정의에도 불구하고 모든 경우에 적합할 때 특히 유용합니다.

파일 유형 수정자(가져오기)

표 5. 유효한 파일 유형 수정자(가져오기) (계속)

수정자	설명
	<p>주:</p> <ol style="list-style-type: none"> <p>MODIFIED BY 옵션을 사용하여 지원되지 않는 파일 유형을 사용하려 할 경우, 가져오기 유틸리티는 경고를 발행하지 않습니다. 이를 시도하면, 가져오기 조작이 실패하며 오류 코드가 리턴됩니다.</p> <p>^a 날짜 형식 문자열 주위의 큰 따옴표는 필수사항입니다. 필드 분리자에는 a-z, A-Z 및 0-9가 포함될 수 없습니다. 필드 분리자는 DEL 파일 형식의 문자 분리문자 또는 필드 분리문자와 같을 수 없습니다. 요소의 시작 및 끝 위치가 명확한 경우에는 필드 분리자를 생략해도 됩니다. 가변 길이 항목들 때문에 D, H, M 또는 S와 같은 요소(수정자에 따라)가 사용되는 경우에는 애매한 부분이 있을 수 있습니다.</p> <p>시간소인 형식의 경우에는 월 및 분 설명자 모두 영문자 M을 사용하기 때문에 혼돈을 피하려면 주의해서 사용해야 합니다. 월 필드는 다른 날짜 필드와 인접해 있어야 합니다. 분 필드는 다른 시간 필드와 인접해 있어야 합니다. 다음은 불명확한 시간소인 형식입니다.</p> <p>"M" (월 또는 분이 될 수 있음) "M:M" (어느 것인지 구분되지 않음) "M:YYYY:M" (둘다 월로 풀이됨) "S:M:YYYY" (시간값 및 날짜값 둘다에 인접함)</p> <p>불명확한 경우에는 유틸리티가 오류 메시지를 보고하고 조작에 실패하게 됩니다.</p> <p>다음은 명확한 시간소인 형식입니다.</p> <p>"M:YYYY" (월) "S:M" (분) "M:YYYY:S:M" (월....분) "M:H:YYYY:M:D" (분....월)</p> <p>주: 큰따옴표 및 백슬래시와 같은 일부 문자 앞에는 Escape 문자(예: \)가 놓여야 합니다.</p> <p>^b 문자는 소스 데이터의 코드 페이지에 지정되어야 합니다.</p> <p>문자 코드 포인트(문자 부호 대신)는 xJJ 또는 0xJJ(여기서, JJ는 코드 포인트의 16진 표시법) 구문을 사용하여 지정될 수 있습니다. 예를 들어, # 문자를 컬럼 분리 문자로 지정하려면 다음 중 하나를 사용하십시오.</p> <p>... modified by coldel# modified by coldel0x23 modified by coldelX23 ...</p> <p>^c 21 페이지의 『분리 문자 제한사항』에서는 분리 문자가 겹쳐쓸 때 사용될 수 있는 문자에 대해 적용되는 제한사항을 나열합니다.</p>

문자 세트 및 NLS 고려사항

문자 데이터의 확장 또는 축소와 관련하여 동일하지 않은 코드 페이지 상황이 일어날 수 있습니다. 예를 들어, 일본어 또는 대만어 확장 UNIX 코드(EUC)와 2바이트 문자 세트(DBCS)는 같은 문자에 대해서도 다른 길이로 코드화할 수 있습니다. 일반적으로, 목표 컬럼 길이에 대해 입력 데이터 길이를 비교하는 일은 데이터를 읽기 전에 수행됩니다. 입력 길이가 목표 길이보다 길 경우, 이 컬럼에 널(NULL) 입력이 가능하면 컬럼에 널(NULL)이 입력됩니다. 그렇지 않으면, 요청이 무시됩니다. nochecklengths 수정자(67 페이지의 『파일 유형 수정자(가져오기)』 참조)가 지정된 경우, 초기 비교가 수행되지 않고 데이터를 가져오려고 시도합니다. 변환이 완료된 다음 데이터가 너무 길 경우, 이 행은 무시됩니다. 그렇지 않으면, 데이터가 가져오기됩니다.

가져오기 세션 예

CLP 예

예 1

다음의 예는 myfile.ixf에서 STAFF 테이블로 정보를 가져오는 방법을 보여주고 있습니다.

```
db2 import from myfile.ixf of ixf messages msg.txt insert into staff
```

```
SQL3150N The H record in the PC/IXF file has product "DB2 01.00",
date "19970220", and time "140848".
```

```
SQL3153N The T record in the PC/IXF file has name "myfile",
qualifier " ", and source " ".
```

```
SQL3109N The utility is beginning to load data from file "myfile".
```

```
SQL3110N The utility has completed processing. "58" rows were read
from the input file.
```

```
SQL3221W ...Begin COMMIT WORK. Input Record Count = "58".
```

```
SQL3222W ...COMMIT of any database changes was successful.
```

SQL3149N "58" rows were processed from the input file. "58" rows were successfully inserted into the table. "0" rows were rejected.

예 2

다음 예에서는 DEL 형식의 데이터가 들어 있는 입력 파일 delfile1로부터 MOVIEABLE 테이블을 가져오는 방법을 보여줍니다.

```
db2 import from delfile1 of del
modified by dldel|
insert into movietable(Actorname, description, url_making_of, url_movie)
datalink specification (dl_url_default_prefix "http://narang"),
(dl_url_replace_prefix "http://bomdel" dl_url_suffix ".mpeg")
```

주:

1. 이 테이블에는 네 개의 컬럼이 있습니다.

actorname	VARCHAR(n)
description	VARCHAR(m)
url_making_of	DATALINK (with LINKTYPE URL)
url_movie	DATALINK (with LINKTYPE URL)

2. 입력 파일의 DATALINK 데이터에는 부속 필드 분리 문자로 사용하는 세로 막대(|) 문자가 들어 있습니다.
3. url_making_of에 대한 컬럼 값이 접두부 문자 순서를 가지고 있지 않으면, "http://narang"이 사용됩니다.
4. url_movie에 대한 모든 널(NULL) 사용 불가능 컬럼 값은 "http://bomdel"을 접두부로 가져옵니다. 이미 있는 값이 바뀔 것입니다.
5. url_movie에 대한 널(NULL) 사용 불가능 컬럼 값은 경로에 추가되는 ".mpeg"를 가져옵니다. 예를 들면, url_movie의 컬럼 값이 "http://server1/x/y/z"이면 "http://bomdel/x/y/z.mpeg"로 저장되고, 값이 "/x/y/z"이면 "http://bomdel/x/y/z.mpeg"로 저장됩니다.

예 3 (식별 컬럼으로 테이블로 가져오기)

TABLE1은 4개의 컬럼이 있습니다.

- C1 VARCHAR(30)
- C2 INT GENERATED BY DEFAULT AS IDENTITY
- C3 DECIMAL(7,2)

- C4 CHAR(1)

TABLE2는 TABLE1과 같습니다. 다만, C2는 GENERATED ALWAYS 식별 컬럼입니다.

DATAFILE1에서 데이터 레코드 (DEL 형식):

```
"Liszt"
"Hummel",,187.43, H
"Grieg",100, 66.34, G
"Satie",101, 818.23, I
```

DATAFILE2에서 데이터 레코드 (DEL 형식):

```
"Liszt", 74.49, A
"Hummel", 0.01, H
"Grieg", 66.34, G
"Satie", 818.23, I
```

다음과 같은 명령은 행 1과 2에 대해 식별 값을 생성합니다. 그 이유는 식별 값이 이들 행에 대해 DATAFILE1에서 제공되지 않았기 때문입니다. 하지만 상대적으로 행 3과 4는 사용자 제공 식별 값 100과 101을 할당합니다.

```
db2 import from datafile1.del of del replace into table1
```

DATAFILE1을 TABLE1로 가져와서 식별 값이 모든 행에 대해 생성되려면, 다음 명령을 입력하십시오.

```
db2 import from datafile1.del of del method P(1, 3, 4) replace into table1 (c1, c3, c4)
db2 import from datafile1.del of del modified by identityignore replace into table1
```

DATAFILE2를 TABLE1로 가져와서 식별 값이 각각의 행에 대해 생성되려면, 다음 명령을 입력하십시오.

```
db2 import from datafile2.del of del replace into table1 (c1, c3, c4)
db2 import from datafile2.del of del modified by identitymissing replace into table1
```

식별 관련 파일 유형 수정자를 사용하지 않고 DATAFILE1을 TABLE2로 가져오게 되면, 수정자들이 자신 소유의 널(NULL)이 아닌 값을 제공하고 식별 컬럼이 GENERATED ALWAYS이기 때문에, 행 1과 2는 삽입되지만 행 3과 4는 거부됩니다.

API 예

24 페이지의 『API 예』에서 자세한 내용을 참조하십시오.

가져오기 성능 최적화

목표 테이블 컬럼 이름 또는 특정 가져오기 방법을 지정하면 원격 데이터베이스로 가져오는 속도가 느려지게 됩니다.

제한사항 및 한계점

다음 제한사항은 가져오기 유틸리티에 적용됩니다.

- 이 유틸리티는 별명의 사용을 지원하지 않습니다.
- 이미 있는 테이블이 종속 테이블의 외부 키가 참조하는 기본 키를 포함하는 상대 테이블일 경우, 이 테이블의 데이터는 바꿀 수 없고 단지 추가만 가능합니다.
- 지금 새로 고침 모드로 정의된 요약 테이블의 기본이 되는 테이블에 가져오기 바꾸기 조작을 수행할 수 없습니다.
- 시스템 테이블, 요약 테이블 또는 구조화 유형 컬럼이 있는 테이블로 데이터를 가져올 수 없습니다.
- 선언된 임시 테이블로 데이터를 가져올 수 없습니다.
- 가져오기 유틸리티를 통해 뷰를 작성할 수 없습니다.
- 참조 제한조건 및 외부 키 정의는 PC/IXF 파일로부터 테이블을 작성할 때 보존되지 않습니다(데이터가 이전에 SELECT *를 사용하여 내보낸 경우에는 기본 키 정의가 보존됩니다).
- 가져오기 유틸리티가 자체 SQL문을 생성하기 때문에, 일부 경우에는 최대 명령문 크기인 64KB가 초과됩니다.

다음 제한사항은 가져오기 유틸리티에 적용됩니다.

원격 데이터베이스에 대한 가져오기 작업에 의해 생성된 출력 메시지 볼륨이 60KB를 초과한 경우에는, 유틸리티가 첫번째 30KB와 마지막 30KB를 보유합니다.

문제점 해결

데이터를 내보내기, 가져오기, 로딩, 바인딩 또는 복원과 같은 DB2 조작을 수행하는 동안 메시지 파일이 이러한 조작과 연관된 오류, 경고 및 정보용 메시지를 포함하여 작성되게 지정할 수 있습니다. MESSAGE 매개변수를 사용하여 이들 파일의 이름을 지정하십시오.

이들 메시지 파일은 표준 ASCII 텍스트 파일입니다. 메시지 파일의 각 메시지는 새로운 행에서 시작하며 DB2 메시지 검색 기능이 제공하는 정보가 들어 있습니다. 이 메시지 파일을 인쇄하려면 운영 체제에 대한 인쇄 절차를 사용하고, 메시지를 보려면 ASCII 편집기를 사용하십시오.

제3장 로드

여기에서는 파일, Named Pipe 또는 장치로부터 DB2 테이블로 데이터를 이동시키는 DB2 UDB 로드 유틸리티에 대해 설명합니다. 이들 데이터 소스는 데이터베이스가 상주하는 노드 또는 원격으로 연결된 클라이언트에 상주할 수 있습니다. 로드될 테이블이 존재해야 합니다. 새로운 데이터를 받는 테이블에 이미 데이터가 들어 있으면, 기존의 데이터를 바꾸거나 추가시킬 수 있습니다.

다음과 같은 내용이 들어 있습니다.

- 84 페이지의 『로드 개요』
- 90 페이지의 『병렬 처리 및 로딩』
- 91 페이지의 『로드를 사용하는 데 필요한 특권, 권한 및 권한 부여』
- 92 페이지의 『로드 사용』
- 94 페이지의 『식별 컬럼으로 로드 사용』
- 96 페이지의 『생성된 컬럼으로 로드 사용』
- 98 페이지의 『제한조건 위반에 대한 점검』
- 100 페이지의 『인터럽트된 로드 조작 재시작』
- 100 페이지의 『로드 사본 위치 파일 사용』
- 103 페이지의 『LOAD 명령』
- 119 페이지의 『LOAD QUERY 명령』
- 121 페이지의 『로드 API』
- 132 페이지의 『데이터 구조: SQLULOAD-IN』
- 137 페이지의 『데이터 구조: SQLULOAD-OUT』
- 139 페이지의 『db2LoadQuery - Load Query API』
- 145 페이지의 『파일 유형 수정자(로드)』
- 159 페이지의 『예외 테이블』
- 160 페이지의 『덤프 파일』

- 160 페이지의 『임시 파일 로드』
- 161 페이지의 『로드 유틸리티 로그 레코드』
- 161 페이지의 『문자 세트와 자국어 지원』
- 162 페이지의 『로드 세션 예』
- 175 페이지의 『로드 조작후 보류 상태』
- 176 페이지의 『로드 성능 최적화』
- 181 페이지의 『제한사항 및 한계점』
- 182 페이지의 『문제점 해결』.

209 페이지의 『DB2 Data Links Manager 데이터를 이동시키기 위해 로드 유틸리티 사용』에서 DB2 Data Links Manager 데이터 로드와 관한 정보를 참조하십시오.

로드 개요

로드 유틸리티는 새로 작성된 테이블이나 이미 데이터가 들어 있는 테이블에 많은 양의 데이터를 효율적으로 이동시킬 수 있습니다. 이 유틸리티는 대형 오브젝트(LOB)와 사용자 정의 유형(UDT)을 포함하는 모든 데이터 유형을 처리할 수 있습니다. 로드 유틸리티가 데이터베이스에 형식화된 페이지를 직접 작성하는 반면, 가져오기 유틸리티는 SQL INSERT를 수행하기 때문에 로드 유틸리티의 처리 속도가 가져오기 유틸리티보다 빠르게 수행됩니다. 로드 유틸리티는 트리거를 시작하지 않으며, 참조 또는 테이블 제한조건 점검(색인의 고유성 유효화 점검 외에는)을 수행하지 않습니다. 로드 및 가져오기 유틸리티의 상세한 비교는 237 페이지의 『부록B. 가져오기 및 로드 유틸리티의 차이점』에서 자세한 내용을 참조하십시오.

로드 유틸리티는 세 가지 단계로 구성됩니다(85 페이지의 그림1 참조).

- 로드. 이 단계에서 데이터가 테이블에 작성됩니다.

로드 단계 동안 데이터가 테이블에 로드되고, 색인 키와 테이블 통계가 수집됩니다(필요한 경우). 저장 지점 또는 일관성 지점은 LOAD 명령의 SAVECOUNT 매개변수를 통해 지정된 간격에 설정됩니다. 저장 지점 시점에 올바르게 로드된 입력 행 수를 나타내는 메시지가 생성됩니다. FILE LINK CONTROL로 정의된 DATALINK 컬럼의 경우, 널(NULL)이 아닌 컬럼 값에 대해 링크 조작이 수

행됩니다. 이 조작이 실패하면, 로드 조작을 재시작할 수 있습니다. RESTART 옵션은 자동으로 마지막에 성공한 일관성 지점부터 로드 조작을 재시작합니다. TERMINATE 옵션은 실패한 로드 조작을 구간 복원합니다.

로드 조작의 세 가지 단계

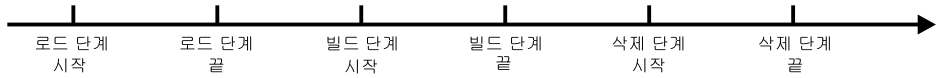


그림 1. 로드 프로세스의 세 단계는 로드, 빌드 및 삭제입니다. 연관된 테이블 공간은 로드 단계의 시작부터 빌드 단계가 끝날 때까지 로드 보류 상태로 되며, 빌드 단계의 끝부터 삭제 단계가 끝날 때까지 삭제 보류 상태로 됩니다.

- 빌드. 이 단계에서 색인이 작성됩니다.
 빌드 단계에서, 로드 단계중 수집된 색인 키를 기준으로 색인이 작성됩니다. 로드 단계에서 색인 키가 정렬되고 색인 통계가 수집됩니다(INDEXES 옵션과 함께 STATISTICS YES로 지정된 경우). 이 통계는 RUNSTATS 명령을 통해 수집된 통계와 유사합니다 (Command Reference 참조). 빌드 단계 도중 실패하면, RESTART 옵션이 자동으로 적합한 지점에서 로드 조작을 재시작합니다.
 - 삭제. 이 단계에서 고유 키 위반이나 DATALINK 위반을 일으킨 행이 테이블에서 삭제됩니다. 고유 키 위반이 지정된 경우에 이는 예외 테이블에 배치되고(159 페이지의 『예외 테이블』 참조), 거절된 행에 관한 메시지가 메시지 파일에 작성됩니다. 로드 프로세스가 완료되면, 이들 메시지를 검토하여 문제점을 해결하고 수정된 행을 테이블에 삽입하십시오.
 로드 유틸리티에 의해 작성된 임시 파일을 삭제하거나 변경하지 마십시오. 삭제 단계에 대해 중요한 일부 임시 파일이 있습니다. 삭제 단계 도중 실패하면, RESTART 옵션이 자동으로 적합한 지점에서 로드 조작을 재시작합니다.
- 주: 모든 삭제 이벤트가 로그됩니다. 고유성 조건을 위반하는 레코드가 많이 있을 경우, 삭제 단계 동안 로그 파일이 가득 차게 될 수 있습니다.

다음은 데이터를 로드할 때 필요한 내용입니다.

- 입력 파일, Named Pipe 또는 장치의 이름과 경로
- 목표 테이블의 이름이나 별명

로드 개요

- 입력 파일에 있는 데이터의 형식. 이 형식은 DEL, ASC 또는 PC/IXF가 될 수 있습니다. 239 페이지의 『부록C. 내보내기/가져오기/로드 유틸리티 파일 형식』에서 자세한 내용을 참조하십시오.
- 입력 데이터가 테이블에 추가될 것인지의 여부 또는 테이블에 이미 있는 데이터를 바꿀 것인지의 여부
- 유틸리티가 API, **sqlload**를 통해 호출될 경우의 메시지 파일 이름

다음과 같이 지정할 수도 있습니다.

- 로드 유틸리티가 원격으로 연결된 클라이언트로부터 호출된 경우에는 로드될 데이터가 클라이언트에 상주함을 나타냅니다.
- 데이터를 로드하기 위해 사용할 방법: 컬럼 위치(location), 컬럼 이름 또는 상대 컬럼 위치(position)
- 유틸리티가 일관성 지점을 설정하려는 빈도. SAVECOUNT 매개변수를 사용하여 이 값을 지정하십시오. 이 매개변수가 지정되면, 로드 재시작 조작은 처음부터 시작하지 않고 마지막 일관성 지점에서 시작합니다.
- 데이터가 삽입될 테이블 컬럼의 이름
- LOB가 저장되는 입력 파일의 경로와 파일 이름. lobsinfile 수정자는 로드 유틸리티에게 모든 LOB 데이터가 파일로부터 로드되고 있다고 알립니다(145 페이지의 『파일 유형 수정자(로드)』 참조).
- 로드 중인 컬럼 값이 내재된 소수점을 가지고 있는지의 여부. implieddecimal 수정자가 로드 유틸리티에게 데이터가 테이블에 들어갈 때 데이터에 대해 소수점이 적용된다고 알려줍니다(145 페이지의 『파일 유형 수정자(로드)』 참조). 예를 들어, 12345 값은 DECIMAL(8,2) 컬럼에 로드될 때 12345.00이 아니라 123.45로 로드됩니다.
- 유틸리티가 테이블이 로드된 다음 사용 가능한 여유 공간의 크기를 수정해야 하는지의 여부. 추가로 여유 공간이 있어야 로드 조작 완료 후 테이블에 대해 INSERT 및 UPDATE를 증가시킬 수 있습니다. 감소된 여유 공간은 관련된 행을 더욱 밀접하게 유지하여, 테이블 성능을 향상시킬 수 있습니다.
 - totalfreepace 수정자는 사용자가 빈 데이터 페이지를 로드된 테이블 끝에 추가할 수 있게 합니다. 이 매개변수로 지정된 숫자는 테이블 끝에 여유 공간으로 추가될 테이블에 있는 전체 페이지의 비율입니다. 예를 들어, 이 매개변수에 숫자 20을 지정하고, 테이블에 100개의 데이터 페이지가 있다면, 20개의 빈 페이지가 추가로 첨부됩니다. 따라서, 테이블에 있는 전체 데이터 페이지 수는 120이 됩니다.

- `pagefreespace` 수정자는 로드된 각 데이터 페이지에 허용될 여유 공간의 크기를 제어할 수 있도록 합니다. 이 매개변수로 지정된 숫자는 여유 공간으로 남게 될 각 데이터 페이지의 비율입니다. 페이지의 첫번째 행은 제한사항 없이 추가됩니다. 따라서, 이 매개변수에 아주 큰 행과 큰 숫자를 지정하면, 이 매개변수로 지정된 값이 나타내는 것보다 적은 여유 공간이 각 페이지에 남게 될 수 있습니다.
- `indexfreespace` 수정자는 로드된 각 색인 페이지에 허용되는 여유 공간의 크기를 제어할 수 있도록 합니다. 이 매개변수로 지정된 숫자는 여유 공간으로 남게 될 각 색인 페이지의 비율입니다. 페이지의 첫번째 색인 항목은 제한사항 없이 추가됩니다. 추가 색인 항목은 색인 페이지에 배치되며, 여유 공간 임계값이 유지보수될 수 있는 백분율이 제공됩니다. 기본값은 CREATE INDEX 시점에서 사용된 값입니다. `indexfreespace` 값은 CREATE INDEX문에 지정된 PCTFREE 값에 우선합니다.

`pagefreespace` 수정자를 지정하고 테이블에 색인이 있으면, `indexfreespace` 지정을 고려해야 합니다. 각 색인 페이지에 대해 남겨둘 여유 공간의 크기를 결정할 때, 테이블에 삽입되는 각 행의 크기가 색인에 삽입된 연관 키의 크기보다 클 것인지를 고려하십시오. 또한, 색인과 테이블에 대한 테이블 공간의 페이지 크기는 서로 다를 수 있습니다.

- 로드 프로세스중 통계가 수집될 것인지의 여부. 이 옵션은 로드 조작이 REPLACE 모드로 수행중일 경우에만 지원됩니다.

데이터가 테이블에 추가될 경우, 통계는 수집되지 않습니다. 추가된 테이블에 대한 현재 통계를 수집하려면, 로드 프로세스 완료에 이어 `runstats` 유틸리티를 호출하십시오. 고유 색인이 있는 테이블에 대한 통계를 수집하고 삭제 단계에서 중복 키가 삭제된 경우, 통계는 삭제된 행에 대해서는 갱신되어 있지 않습니다. 중복된 레코드 수가 너무 많을 것이라고 생각한다면, 로드 조작시 통계를 수집하지 마십시오. 대신, 로드 프로세스 완료 후 `runstats` 유틸리티를 호출하십시오.

- 작성된 변경사항의 사본을 유지할 것인지의 여부. 이는 데이터베이스의 롤 포워드 복구를 작동 가능하게 하기 위해 수행됩니다. 이 옵션은 데이터베이스에 대해 포워드 로그 복구가 작동 불가능할 경우, 즉, `logretain`과 `userexit`가 작동되지 않으면 지원되지 않습니다. 사본이 작성되지 않고 포워드 로그 복구도 작동 불가능할 경우, 테이블 공간은 로드 조작 완료 시점에서 백업 보류 상태로 남게 됩니다(175 페이지의 『로드 조작후 보류 상태』 참조).

전체 복구 가능한 데이터베이스에 대해 기록이 요구됩니다. 로드 유틸리티는 데이터 로딩과 연관된 기록을 거의 완전하게 제거합니다. 기록 대신, 테이블의 로드된 부분의 사본을 작성하는 옵션이 있습니다. 100 페이지의 『로드 사본 위치 파일 사용』에서 DB2

가 로드 사본의 트랙을 유지하는 방법에 관한 정보를 참조하십시오. 실패에 따른 데이터베이스 복구를 허용하는 데이터베이스 환경이 있다면, 다음 중 하나를 수행할 수 있습니다.

- 테이블의 로드된 부분의 사본 작성을 명시적으로 요청함
- 로드 조작이 완료된 후 즉시 테이블이 상주하고 있는 테이블 공간을 백업함

이미 데이터가 들어 있는 테이블을 로드중이며 데이터베이스가 복구 불가능할 경우, 로드 유틸리티를 호출하기 전에, 오류로부터 복구할 수 있도록 데이터베이스 또는 테이블이 로드중인 테이블 공간의 백업 사본을 가지고 있는지 확인하십시오.

복구 가능한 데이터베이스에서 일련의 복수 로드 조작을 수행하려고 할 경우, 각 로드 조작을 복구 불가능으로 지정하고 로드 순서의 맨 끝에서 백업을 하면, COPY YES 옵션을 사용하여 각 로드 조작을 호출하는 것보다 조작 순서가 더욱 빨라집니다. NONRECOVERABLE 옵션을 사용하여 로드 트랜잭션을 복구 불가능으로 표시하고, 다음의 롤 포워드 조치로 복구될 수 없도록 지정할 수 있습니다. 롤 포워드 유틸리티는 이 트랜잭션을 생략하고, 데이터가 로드되고 있는 테이블을 "유효하지 않은 테이블"로 표시합니다. 이 유틸리티는 이 테이블에 대한 모든 다음 트랜잭션도 무시합니다. 롤 포워드가 완료된 후, 이러한 테이블은 삭제만 가능합니다(그림2 참조). 이 옵션의 사용으로, 테이블 공간이 로드 조작 후 백업 보류 상태로 되지 않고 로드된 데이터의 사본이 로드 조작 동안 작성될 필요가 없습니다.

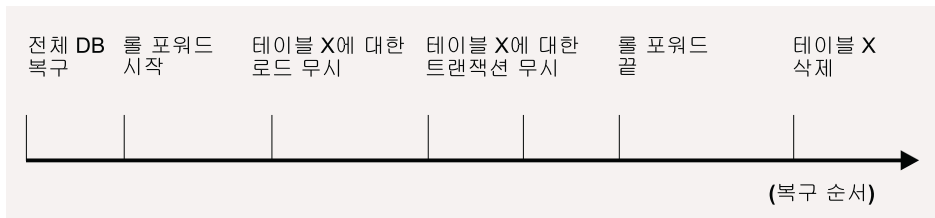


그림2. 롤 포워드 조치 수행중 복구 불가능한 처리

관리 안내서에서 자세한 내용을 참조하십시오.

- 로드 조작중 임시 파일을 작성할 때 사용될 완전한 경로. 이 이름은 LOAD 명령의 TEMPFILES PATH 매개변수에 의해 지정됩니다. 기본값은 데이터베이스 경로입니다. 경로는 서버 머신에 있으며, DB2 인스턴스가 전용으로 액세스합니다. 따라서, 이 매개변수에 지정된 모든 경로 이름 규정은 클라이언트가 아닌 서버의 디렉토리 구조를 반영해야 하며, DB2 인스턴스 소유자는 경로에 대해 읽기 및 쓰기 사용권한을 가지고 있어야 합니다. 이는 사용자가 인스턴스 소유자인 경우에도 적용됩니다. 사용자가 인스턴스

소유자가 아니라면, 인스턴스 소유자가 쓰기 가능한 위치(location)를 지정해야 합니다.
160 페이지의 『임시 파일 로드』에서 임시 파일에 관한 자세한 정보를 참조하십시오.

버전 6 및 버전 7에 소개된 이전 로드 동작에 대한 변경사항

버전 6 및 버전 7 로드 유틸리티는 이전 릴리스와 완전한 백레벨 호환성을 가집니다. 즉, 이 유틸리티는 이전 릴리스의 구문을 받아들여 정상적으로 작동합니다. 다음은 버전 6에 소개된 로드 동작에 대한 변경사항 및 구문 변경사항을 요약한 것입니다.

- 로드 재시작은 더 이상 RESTARTCOUNT 값을 사용하지 않으며 매개변수는 현재 예약된 상태입니다. 이전에 인터럽트된 로드 조작을 재시작하면, 로드 조작이 로드, 빌드 또는 삭제 단계의 최종 일관성 지점부터 자동으로 계속됩니다.
- 색인 작성 동안 색인 키의 정렬에는 DB2 UDB 버전 6 데이터베이스 엔진에서 사용하는 새로운 정렬 알고리즘을 이용합니다. 정렬에 전용된 메모리양은 정렬 힙(sortheap) 데이터베이스 구성 매개변수 및 정렬 힙 임계값(sheaphtres) 데이터베이스 관리자 구성 매개변수의 값에 의해 제어됩니다. 이는 SORT BUFFER에 대한 기존 LOAD 옵션과 USING 절을 통해 지정된 임시 정렬 디렉토리를 이제는 사용하지 않음을 의미합니다. 버전 6에 이 옵션을 지정하면, 정보용 경고 메시지가 리턴되지만 로드 조작은 정상적으로 진행됩니다.

로드 색인 작성 동안 발생하는 정렬 넘침(spill)이 이제는 임시 테이블 공간 내에서 수행됩니다. 정렬 조작은 디스크로 직접 넘치지 않으며 오히려, 임시 테이블 공간과 관련된 버퍼 풀로 넘칩니다. 임시 테이블 공간과 관련된 대형 버퍼 풀이 있으면 색인 작성 시간을 단축시킬 수 있습니다. 버전 6 이전의 로드 정렬 조작(여러 개의 임시 정렬 디렉토리를 지정함으로써)에서 사용가능했던 것과 동일한 유형의 I/O 병렬화를 수행하려면, 각각 다른 디스크 장치에 상주하는 여러 개의 컨테이너로 임시 테이블 공간을 선언하는 것이 바람직합니다. 또한 이 자원들을 사용하지 않을 때는 디스크 자원을 보류시키지 않고 많은 볼륨의 데이터를 수용할 수 있도록 임시 테이블 공간을 SMS(시스템 관리 공간)로 선언하는 것이 바람직합니다.

- REMOTE FILE 옵션의 이름이 바뀌었습니다(임시 파일에 대한 경로를 지정할 때는 유틸리티에서 여전히 REMOTE FILE 매개변수를 허용함). 이는 이 매개

로드 구문 및 동작의 변경사항

변수의 의미와 목적을 보다 잘 반영하기 위한 순전히 문법적인 변경사항입니다. TEMPFILES PATH 매개변수는 파일이 아닌 디렉토리를 반영합니다(160 페이지의 『임시 파일 로드』 참조)

- 로드 유틸리티는 이제 여러 가지 색인화 모드(완전 REBUILD, INCREMENTAL 확장, 로드 조장이 완료될 때까지 색인 유지보수 DEFERRED 및 수행시 전체 리빌드 및 증가 유지보수 사이에 선택하는 AUTOSELECT 모드)를 지원합니다. 전체 리빌드 모드는 버전 6 이전 릴리스의 동작을 이중복사합니다. 버전 6의 기본 동작은 AUTOSELECT 모드입니다.
- 버전 6에서 TERMINATE 옵션은 로드 조장을 구간 복원하는 데 사용할 수 있습니다. 이전에는 이 옵션이 테이블 공간을 복원 보류 중 상태로 놓았습니다. 그러나 실패한 LOAD REPLACE 조작 이후의 TERMINATE 요청에서는 테이블 데이터를 복원하지 않는다는 점에 유의하십시오.

버전 7 로드 유틸리티는 원격으로 연결된 클라이언트에 상주하는 데이터를 완전한 파일 또는 Named Pipe로 로드할 수 있습니다. lobsinfile 파일 유형 수정자가 지정된 경우에는 LOB값이 들어 있는 개별 파일이 서버에 놓여야 합니다.

병렬 처리 및 로딩

로드 유틸리티에는 대칭적 멀티프로세서(SMP) 환경과 같이 여러 개의 프로세서 또는 여러 개의 저장영역 장치가 사용되는 하드웨어 구성의 이점이 있습니다. 로드 유틸리티를 사용하여 많은 양의 데이터를 병렬 처리하는 데에는 여러 가지 방법이 있습니다. 한 가지 방법은 로드 조작시 I/O 병렬 처리를 할 수 있는 복수 저장영역 장치를 사용하는 것입니다(91 페이지의 그림3 참조). 다른 방법은 파티션 내 병렬 처리를 허용하는 SMP 환경에서 여러 개의 프로세서 사용과 관계됩니다(91 페이지의 그림4 참조). 두 가지 방법 모두 더 빠르게 데이터를 로드할 수 있게 합니다. 176 페이지의 『로드 성능 최적화』에서 자세한 내용을 참조하십시오.

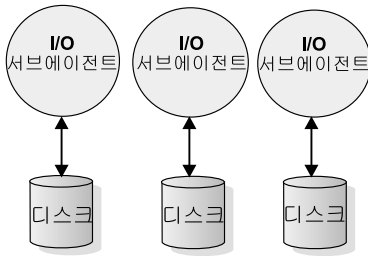


그림 3. 데이터 로드시 I/O 병렬 처리의 이점

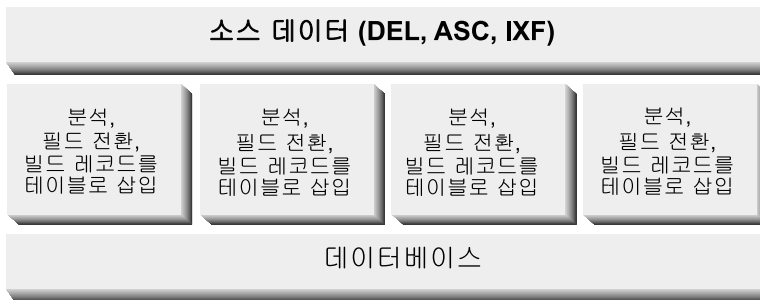


그림 4. 데이터 로드시 파티션 내 병렬 처리의 이점

로드를 사용하는 데 필요한 특권, 권한 및 권한 부여

특권은 사용자가 데이터베이스 자원을 작성하거나 액세스할 수 있게 합니다. 권한 레벨은 특권과 상위 레벨 데이터베이스 관리 프로그램 유지보수 및 유틸리티 조작용 그룹으로 분류하는 방법을 제공합니다. 이는 모두 데이터베이스 관리 프로그램과 데이터베이스 오브젝트에 대한 액세스를 제어하기 위한 것입니다. 사용자는 적합한 권한 즉, 필요한 특권이나 권한을 가지고 있는 오브젝트만 액세스할 수 있습니다.

테이블에 데이터를 로드하려면, 다음 중 하나를 갖추어야 합니다.

- 데이터베이스에 대한 SYSADM 권한
- DBADM 권한
- LOAD 권한 및

로드를 사용하는 데 필요한 권한

- 로드 유틸리티가 INSERT 모드, TERMINATE 모드(이전 로드 삽입 작업을 종료하기 위해) 또는 RESTART 모드(이전 로드 삽입 작업을 재시작하기 위해)로 호출될 때 테이블에 대한 INSERT 특권.
- 로드 유틸리티가 REPLACE 모드, TERMINATE 모드(이전 로드 바꾸기 작업을 종료하기 위해) 또는 RESTART 모드(이전 로드 바꾸기 작업을 재시작하기 위해)로 호출될 때 테이블에 대한 INSERT 및 DELETE 특권.
- 예외 테이블이 로드 조작의 일부분으로 사용된 경우에는 예외 테이블에 대한 INSERT 특권.

모든 로드 프로세스(및 일반적으로 모든 DB2 서버 프로세스)가 인스턴스 소유자에게 소유되고, 이들 모든 프로세스가 인스턴스 소유자의 ID를 사용하여 필요한 파일을 액세스하기 때문에, 인스턴스 소유자에게는 입력 데이터 파일에 대해 읽기 액세스가 있어야 합니다. 이들 입력 데이터 파일은 누가 명령을 호출하는지에 관계없이 인스턴스 소유자가 읽을 수 있어야 합니다.

Windows NT용 DB2가 Windows NT 운영 체제에 대해 서비스로 정의되어 있을 경우, 서비스에는 LAN 자원(드라이브, 디렉토리 및 파일)을 사용하는 데 필요한 읽기/쓰기 파일 사용권한이 있는 사용자 계정이 있어야 합니다.

로드 사용

로드를 사용하기 전에

로드 유틸리티를 호출하기 전에, 데이터가 로드될 데이터베이스에 연결되어야 합니다(또는 내재적으로 연결할 수 있어야 합니다). 유틸리티가 COMMIT문을 발행할 것이므로, 로드 유틸리티를 호출하기 전에 COMMIT 또는 ROLLBACK을 수행하여 모든 트랜잭션을 완료하고 모든 잠금을 해제해야 합니다.

데이터가 입력 파일에 나타난 순서대로 로드되기 때문에 특정 순서가 필요하면, 데이터는 로드 조작이 시도되기 전에 정렬되어야 합니다.

클러스터링이 필요하면, 데이터는 로드되기 전에 클러스터링 색인에 대해 정렬되어야 합니다.

로드 호출

로드 유틸리티는 다음을 통해 호출될 수 있습니다.

- 명령행 처리기(CLP)

다음은 CLP를 통해 발행된 로드 명령의 예입니다.

```
db2 load from stafftab.ixf of ixf messages staff.msgs
      insert into userid.staff copy yes use tsm data buffer 4000
```

이 예에서,

- 모든 경고나 오류 메시지는 staff.msgs 파일에 들어갑니다.
- 작성된 변경사항의 사본이 TSM(Tivoli Storage Manager, 이전에는 ADSM이었음)에 저장됩니다. 관리 안내서에 있는 데이터베이스 복구 장의 “Tivoli Storage Manager” 절에서 TSM에 대한 자세한 내용을 참조하십시오.
- 4,000페이지의 버퍼 공간이 로드 조작중 사용됩니다.

다음은 CLP를 통해 발행된 LOAD 명령의 다른 예입니다.

```
db2 load from stafftab.ixf of ixf messages staff.msgs
      tempfiles path /u/myuser replace into staff
```

이 예에서,

- 테이블 데이터가 바꾸게 됩니다.
- 임시 파일이 작성될 서버 경로로서 /u/myuser를 지정하는 데 TEMPFILE PATH 매개변수가 사용됩니다.

주: 이들 예는 로드 입력 파일에 대해 상대 경로 이름을 사용합니다. 상대 경로 이름은 데이터베이스와 같은 노드상의 클라이언트의 호출에서만 사용할 수 있습니다. 완전한 경로 이름을 사용하도록 권장됩니다.

- 제어 센터에 있는 로드 노트북. 로드 노트북을 열려면 다음을 수행하십시오.
 1. 제어 센터에서, 테이블 폴더가 나올 때까지 오브젝트 트리를 확장하십시오.
 2. 테이블 폴더를 클릭하십시오. 이미 있는 테이블이 창의 오른쪽에 있는 파티션 영역에 표시됩니다(내용 영역).
 3. 내용 영역에서 원하는 테이블을 오른쪽 마우스 버튼으로 클릭한 후, 팝업 메뉴에서 로드를 선택하십시오. 로드 노트북이 열립니다.

관리 안내서에서 제어 센터에 관한 일반적인 정보를 참조하십시오. 자세한 정보는 제어 센터의 온라인 도움말 기능을 통해 제공됩니다.

- API, `sqlload`. 121 페이지의 『로드 API』에서 이 API에 관한 정보를 참조하십시오. 응용프로그램 빌드 안내서에서 DB2 관리 API를 포함하는 응용프로그램 작성에 관한 일반적인 정보를 참조하십시오.

식별 컬럼으로 로드 사용

로드 유틸리티는 식별 컬럼이 들어 있는 테이블로 데이터를 로드하는 데 사용할 수 있습니다. 식별 관련 파일 유형 수정자를 사용하지 않는 경우에는 유틸리티가 다음 규칙에 따라 기능합니다.

- 식별 컬럼이 `GENERATED ALWAYS`이면, 입력 파일의 해당 행이 식별 컬럼에 대해 값을 유실하거나 널(NULL) 값이 명시적으로 주어질 때마다 식별 값이 테이블 행에 대해 생성됩니다. 식별 컬럼에 대해 비 널(NULL) 값이 지정되면 행이 거부됩니다(SQL3550W).
- 식별 컬럼이 `GENERATED BY DEFAULT`인 경우에, 로드 유틸리티는 사용자 제공 값이 주어진다면 그 값을 사용합니다. 데이터가 유실되거나 명시적으로 널(NULL)인 경우에는 값이 생성됩니다.

로드 유틸리티는 식별 컬럼의 데이터 유형(즉, `SMALLINT`, `INT`, `BIGINT` 또는 `DECIMAL`) 값에 대해 일반적으로 수행되는 정도를 넘어서 사용자 제공 식별 값을 추가로 검증하지 않습니다. 중복값은 보고되지 않습니다.

로드 유틸리티에서는 식별 컬럼이 들어 있는 테이블과의 사용을 간소화하기 위해 세 가지(상호 배타적인) 파일 유형 수정자를 지원합니다.

- `identitymissing` 수정자를 사용하면 입력 데이터 파일에 식별 컬럼에 대한 값이 아무 것도 들어 있지 않은 경우(널(NULL)까지도)에 식별 컬럼이 있는 테이블을 보다 편리하게 로드할 수 있습니다. 예를 들어, 다음과 같은 SQL문으로 정의된 테이블을 고려해 봅시다.

```
create table table1 (c1 varchar(30),
                    c2 int generated by default as identity,
                    c3 decimal(7,2),
                    c4 char(1))
```

사용자는 식별 컬럼이 없는 테이블에서 내보내진 파일(load.del)의 데이터가 있는 TABLE1을 로드하려 할 수도 있습니다. 다음은 그러한 파일의 예입니다.

```
Robert, 45.2, J
Mike, 76.9, K
Leo, 23.4, I
```

이 파일을 로드하기 위한 한가지 방법은 다음과 같이 LOAD 명령을 통해 로드될 컬럼을 명시적으로 나열하는 것입니다.

```
db2 load from load.del of del replace into table1 (c1, c3, c4)
```

그러나, 여러 컬럼이 있는 테이블의 경우에는 이 구문이 번거롭고 오류를 더 유발할 수도 있습니다. 파일 로드의 대체 방법은 다음과 같이 identitymissing 파일 유형 수정자를 사용하는 것입니다.

```
db2 load from load.del of del modified by identitymissing replace into table1
```

- identityignore 수정자는 일부 방식에서는 identitymissing 수정자의 반대입니다. 이는 입력 데이터 파일에 식별 컬럼에 대한 데이터가 들어 있을지라도 데이터를 무시하고 각 행에 대해 식별값을 생성하도록 로드 유틸리티에 지정합니다. 예를 들어, 사용자는 위에서 정의한대로 다음과 같은 데이터가 들어 있는 데이터 파일(load.del)에서 TABLE1을 로드하길 원할 수 있습니다.

```
Robert, 1, 45.2, J
Mike, 2, 76.9, K
Leo, 3, 23.4, I
```

사용자 제공 값 1, 2 및 3이 식별 컬럼에 대해 사용되지 않으면, 사용자가 다음과 같은 LOAD 명령을 발행할 수 있습니다.

```
db2 load from load.del of del method P(1, 3, 4) replace into table1 (c1, c3, c4)
```

또, 이러한 접근 방식도 테이블에 여러 컬럼이 있는 경우에는 번거롭고 오류를 더 유발할 수 있습니다. identityignore 수정자는 다음과 같이 구문을 간소화합니다.

```
db2 load from load.del of del modified by identityignore replace into table1
```

- identityoverride 수정자는 GENERATED ALWAYS 식별 컬럼이 있는 테이블로 사용자 제공 값을 로드하는 데 사용됩니다. 이는 다른 데이터베이스 시스템에서 데이터를 이주하고 테이블이 GENERATED ALWAYS로 정의되어야 할 때 또는 ROLLFORWARD DATABASE 명령의 DROPPED TABLE

식별 컬럼으로 로드 사용

RECOVERY 옵션을 사용하여 복구된 데이터로부터 테이블을 로드할 때 매우 유익할 수 있습니다. 이 수정자를 사용할 때는 식별 컬럼에 대한 데이터(또는 NULL 데이터)가 없는 행들은 거부됩니다(SQL3116W).

주: 이 수정자를 사용할 때 GENERATED ALWAYS 컬럼의 고유성 등록 정보를 위반할 수 있습니다.

생성된 컬럼으로 로드 사용

로드 유틸리티는 생성된 (비식별) 컬럼이 들어 있는 테이블로 데이터를 로드하는 데 사용할 수 있습니다. 그러나 현재는 유틸리티 내에서 비식별 컬럼값의 생성을 지원하지 않으므로 그와 같은 컬럼이 있는 테이블로 로드하면 항상 테이블이 점검 보류 상태로 남게 됩니다. 테이블을 점검 보류 상태에서 벗어나게 하여 강제로 값을 생성토록 하려면 다음을 발행하십시오.

```
SET INTEGRITY FOR tablename IMMEDIATE CHECKED FORCE GENERATED;
```

어떤 값이 로드되더라도 허용하려면 다음을 발행하십시오.

```
SET INTEGRITY FOR tablename GENERATED COLUMN IMMEDIATE UNCHECKED;
```

생성된 컬럼 관련 파일 유형 수정자를 사용하지 않는 경우에는 로드 유틸리티가 다음 규칙에 따라 기능합니다.

- 널(NULL) 입력 가능한 생성 컬럼의 경우에 데이터 파일 내의 컬럼에 대해 NULL값이 제공되면 이를 받아들여 테이블에 로드합니다. 널(NULL) 입력 가능하지 않은 생성 컬럼의 경우에 데이터 파일 내의 컬럼에 대해 NULL값이 제공되면 생성된 컬럼의 데이터 유형에 대한 기본값이 로드됩니다. 널(NULL) 입력 가능 및 널(NULL) 입력 가능하지 않은 생성 컬럼 둘다의 경우에 데이터 파일에 비 NULL값이 있으면 행이 거부됩니다(SQL3550W).
- 로드 조작이 완료된 후에 SET INTEGRITY문에 FORCE GENERATED 옵션을 사용하여 테이블이 점검 보류 상태에서 벗어나면 로드된 값이 생성된 값으로 바뀌집니다. FORCE GENERATED 옵션을 사용하지 않으면 다음의 경우에 로드된 값이 남아 있습니다.
 - 생성된 컬럼 제한조건을 위반하지 않은 경우.

– GENERATED COLUMN IMMEDIATE UNCHECKED 옵션을 사용한 경우.

GENERATED COLUMN IMMEDIATE UNCHECKED 옵션을 사용하지 않으면, 생성된 컬럼 제한조건을 위반한 행들이 예외 테이블(제공된 경우)에 놓입니다.

로드 유틸리티에서는 생성된 컬럼이 들어 있는 테이블과의 사용을 간소화하기 위해 세 가지(상호 배타적인) 파일 유형 수정자를 지원합니다.

- `generatedmissing` 수정자를 사용하면, 테이블에 나타난 생성된 모든 컬럼에 대한 아무 값도 입력 데이터 파일에 들어 있지 않은 경우(널(NULL)까지도)에, 생성된 컬럼이 있는 테이블을 보다 편리하게 로드할 수 있습니다. 예를 들어, 다음과 같은 SQL문으로 정의된 테이블을 고려해 봅시다.

```
create table table1 (c1 int,
                    c2 int,
                    g1 int generated always as (c1 + c2),
                    g2 int generated always as (2 * c1),
                    c3 char(1))
```

사용자는 생성된 컬럼이 없는 테이블에서 내보내진 파일(load.del)의 데이터가 있는 TABLE1을 로드하려 할 수도 있습니다. 다음은 그러한 파일의 예입니다.

```
1, 5, J
2, 6, K
3, 7, I
```

이 파일을 로드하기 위한 한가지 방법은 다음과 같이 LOAD 명령을 통해 로드될 컬럼을 명시적으로 나열하는 것입니다.

```
db2 load from load.del of del replace into table1 (c1, c2, c3)
```

그러나, 여러 컬럼이 있는 테이블의 경우에는 이 구문이 번거롭고 오류를 더 유발할 수도 있습니다. 파일 로드의 대체 방법은 다음과 같이 `generatedmissing` 파일 유형 수정자를 사용하는 것입니다.

```
db2 load from load.del of del modified by generatedmissing replace into table1
```

- `generatedignore` 수정자는 일부 방식에서는 `generatedmissing` 수정자의 반대입니다. 이는 목표 테이블에 있는 생성된 모든 컬럼에 대한 데이터가 입력 데이터 파일에 들어 있을지라도 데이터를 무시하고 각 컬럼에 NULL이 로드되도록

생성된 컬럼으로 로드 사용

록 로드 유틸리티에 지정합니다. 예를 들어, 사용자는 위에서 정의한대로 다음과 같은 데이터가 들어 있는 데이터 파일(load.del)에서 TABLE1을 로드하길 원할 수 있습니다.

```
1, 5, 10, 15, J
2, 6, 11, 16, K
3, 7, 12, 17, I
```

사용자 제공, 비 널(NULL) 값 10, 11 및 12(g1의 경우) 그리고 15, 16 및 17(g2의 경우)을 사용하면 행이 거부됩니다(SQL3550W). 이를 피하기 위해 사용자가 다음과 같이 LOAD 명령을 발행할 수 있습니다.

```
db2 load from load.del of del method P(1, 2, 5) replace into table1 (c1, c2, c3)
```

또, 이러한 접근 방식도 테이블에 여러 컬럼이 있는 경우에는 번거롭고 오류를 더 유발할 수 있습니다. generatedignore 수정자는 다음과 같이 구문을 간소화합니다.

```
db2 load from load.del of del modified by generatedignore replace into table1
```

- generatedoverride 수정자는 생성된 컬럼이 있는 테이블로 사용자 제공 값을 로드하는 데 사용됩니다. 이는 다른 데이터베이스 시스템에서 데이터를 이주할 때 또는 ROLLFORWARD DATABASE 명령의 DROPPED TABLE RECOVERY 옵션을 사용하여 복구된 데이터로부터 테이블을 로드할 때 매우 유용할 수 있습니다. 이 수정자를 사용할 때는 널(NULL) 입력 가능하지 않은 생성된 컬럼에 대한 데이터(또는 NULL 데이터)가 없는 행들은 거부됩니다(SQL3116W).

제한조건 위반에 대한 점검

로드 조작이 수행된 다음, 로드된 테이블은 테이블 점검 제한조건이나 참조 무결성 제한조건이 정의되어 있을 경우 점검 보류 상태가 될 수 있습니다. 로드된 테이블에 해당하는 SYSCAT.TABLES 항목의 STATUS 플래그는 테이블의 점검 보류 상태를 나타냅니다. 로드된 테이블이 사용 가능하게 되기 위해서는 STATUS에 정상 상태를 나타내는 N 값이 있어야 합니다.

점검 보류 상태를 제거하려면, SET INTEGRITY문을 사용하십시오(SQL 참조서 참조). SET INTEGRITY문은 테이블의 제한조건 위반을 점검하고 테이블이 점검

보류 상태에서 빠져 나오도록 합니다. 모든 로드 조작이 INSERT 모드에서 수행 될 경우, SET INTEGRITY문은 기본적으로 점차적으로 제한조건을 처리합니다(즉, 테이블의 추가 부분에 대해서만 제한조건 위반을 점검합니다). 예를 들면, 다음과 같습니다.

```
db2 load from infile1.ixf of ixf insert into table1
db2 set integrity for table1 immediate checked
```

TABLE1의 추가 부분만 제한조건 위반을 점검합니다. 추가된 부분만 제한조건 위반을 점검하면, 전체 테이블을 점검하는 것보다 더욱 빠르게 처리됩니다. 데이터가 적게 추가된 대형 테이블일 경우 특히 빠릅니다.

이 명령문을 한 번 호출하여 하나 이상의 테이블을 점검할 수 있습니다. 종속 테이블이 점검될 경우, 상위 테이블이 점검 보류 상태에 있지 않아야 합니다. 참조 무결성 순환의 경우, 이 순환에 관련된 모든 테이블은 SET INTEGRITY문의 단일 호출에 포함되어 있어야 합니다. 이는 종속 테이블이 로드되는 동안 상위 테이블의 제한조건 위반을 점검할 때 편리합니다. 이러한 경우에는 두 개의 테이블이 같은 테이블 공간에 있지 않을 경우에만 발생할 수 있습니다.

SET INTEGRITY문을 발행할 때, INCREMENTAL 옵션을 지정하여 점차적인 처리를 명시적으로 요청할 수 있습니다. 대부분의 경우, 점차적인 처리가 기본 활동이기 때문에 이 옵션은 필요하지 않습니다. 점차적인 처리가 불가능하면, 자동으로 전체 처리가 사용됩니다. INCREMENTAL 옵션이 지정되어 있지만 점차적인 처리가 불가능할 때, 다음의 경우 오류가 발생합니다.

- 두 테이블 모두 점검 보류 상태에 있는 동안 새로운 제한조건이 테이블이나 상위 테이블에 추가됨.
- 테이블에 대한 마지막 무결성 점검 이후에 로드 바꾸기 조작이 발생했거나 NOT LOGGED INITIALLY WITH EMPTY TABLE 옵션이 활성화되었음.
- 상위 테이블이 로드 바꾸기 되거나 비점차적으로 무결성을 점검함.
- 이주 전에 테이블이 점검 보류 상태에 있었음. 이주 후 테이블에서 무결성을 맨 처음 점검할 때는 전체 처리가 필요합니다.
- 테이블 또는 그의 상위가 들어 있는 테이블 공간이 특정 시점으로 롤 포워드되었음.

T 테이블에 SYSCAT.TABLES 카탈로그의 CONST_CHECKED 컬럼에 하나 이상의 W 값이 있으면(W 상태 설명에 대해서는 SQL 참조서의 SET INTEGRITY 문 참조), 시스템은 INCREMENTAL 옵션이 지정되어 있지 않을 경우 전체 테이블에서 제한조건 위반을 점검합니다. 옵션이 지정되어 있으면, 이것이 허용되지만, 모든 데이터가 시스템에 의해 검증되는 것이 아니라는 것을 나타내기 위해 SYSTABLES의 CONST_CHECKED 컬럼이 U로 표시됩니다.

로드 예외 테이블 옵션을 사용하여 제한조건 위반사항이 있는 행에 대한 정보를 캡처합니다(159 페이지의 『예외 테이블』 참조).

SET INTEGRITY문이 제한조건을 위반한 행을 삭제한 결과로 DELETE 트리거를 활성화시키지는 않지만, 테이블이 점검 보류 상태에서 나온 다음 트리거가 활성화됩니다. 이와 같이, 데이터를 수정하여 예외 테이블로부터 로드 테이블로 행을 삽입하면, 테이블에 정의되어 있는 모든 INSERT 트리거가 활성화됩니다. 이것의 결과가 고려되어야 합니다. 하나의 옵션은 INSERT 트리거를 삭제하고 예외 테이블의 행을 삽입한 다음, INSERT 트리거를 재작성하는 것입니다.

인터럽트된 로드 조작 재시작

로드 유틸리티가 존재하지 않는 데이터 파일이나 유효하지 않은 컬럼 이름과 같은 사용자 오류 때문에 재시작할 수 없으면, 처리를 종료하고 테이블 공간을 정상 상태로 남겨둡니다.

데이터를 로드하는 동안 실패하면, 마지막 일관성 지점에서 로드 조작을 재시작하거나(RESTART 옵션 사용) 전체 테이블을 다시 로드할 수 있습니다(REPLACE 옵션 사용). 이전 호출에서 사용한 매개변수와 동일한 매개변수를 사용하여 유틸리티가 필요한 임시 파일을 찾을 수 있도록 하십시오.

로드 사본 위치 파일 사용

로드 사본 위치 정보가 있는 파일을 식별하기 위해 DB2LOADREC 레지스트리 변수가 사용됩니다. 이 파일은 롤 포워드 복구시 로드 사본의 위치를 지정하는 데 사용됩니다. 다음에 관한 정보가 들어 있습니다.

- 매체 유형

- 사용될 매체 장치의 수
- 테이블 로드 조작중 생성된 로드 사본의 위치
- 로드 사본의 파일 이름(적용 가능한 경우)

위치 파일이 존재하지 않거나 일치하는 항목이 파일에서 발견되지 않으면, 로그 레코드의 정보가 사용됩니다.

파일의 정보는 롤 포워드 복구가 수행하기 전에 겹쳐쓰일 수 있습니다.

파티션된 데이터베이스 환경에서, 로드 사본 위치 파일이 각 데이터베이스 파티션 서버에 있어야 하며, 파일 이름(경로 포함)이 같아야 합니다.

다음은 위치 파일의 예입니다. 처음 5개의 매개변수에는 유효한 값이 있어야 하며 로드 사본을 식별하는 데 사용됩니다. 기록된 각 로드 사본에 대해 전체 구조가 반복됩니다.

```

TIMestamp      19950725182542      * Time stamp generated at load time
SCHEMA        PAYROLL              * Schema of table loaded
TABlename     EMPLOYEES             * Table name
DATAbasename  DBT                      * Database name
DB2instance   TORONTO                 * DB2INSTANCE
BUFFernumber  NULL                     * Number of buffers to be used for recovery
SESSionnumber NULL                     * Number of sessions to be used for recovery
TYPEofmedia   L                          * Type of media - L for local device
                                     A for Tivoli Storage Manager (TSM, formerly AD5M)
                                     0 for other vendors

LOCationnumber 3                      * Number of locations
ENTry         /u/toronto/dbt.payroll.employes.001
ENT           /u/toronto/dbt.payroll.employes.002
ENT           /dev/rmt0
TIM           19950725192054
SCH          PAYROLL
TAB          DEPT
DAT          DBT
DB2          TORONTO
SES         NULL
BUF         NULL
TYP         A
TIM         19940325192054
SCH        PAYROLL
TAB        DEPT
DAT        DBT
DB2        TORONTO
SES        NULL
BUF        NULL
TYP        0
SHRlib     /@sys/lib/backup_vendor.a
    
```

주:

1. 각 키워드의 처음 세 문자에는 중요한 의미가 있습니다. 모든 키워드는 지정된 순서로 되어 있어야 합니다. 공백 행은 사용할 수 없습니다.
2. 시간소인은 *yyyymmddhhmmss* 형식입니다.

로드 사본 위치 파일 사용

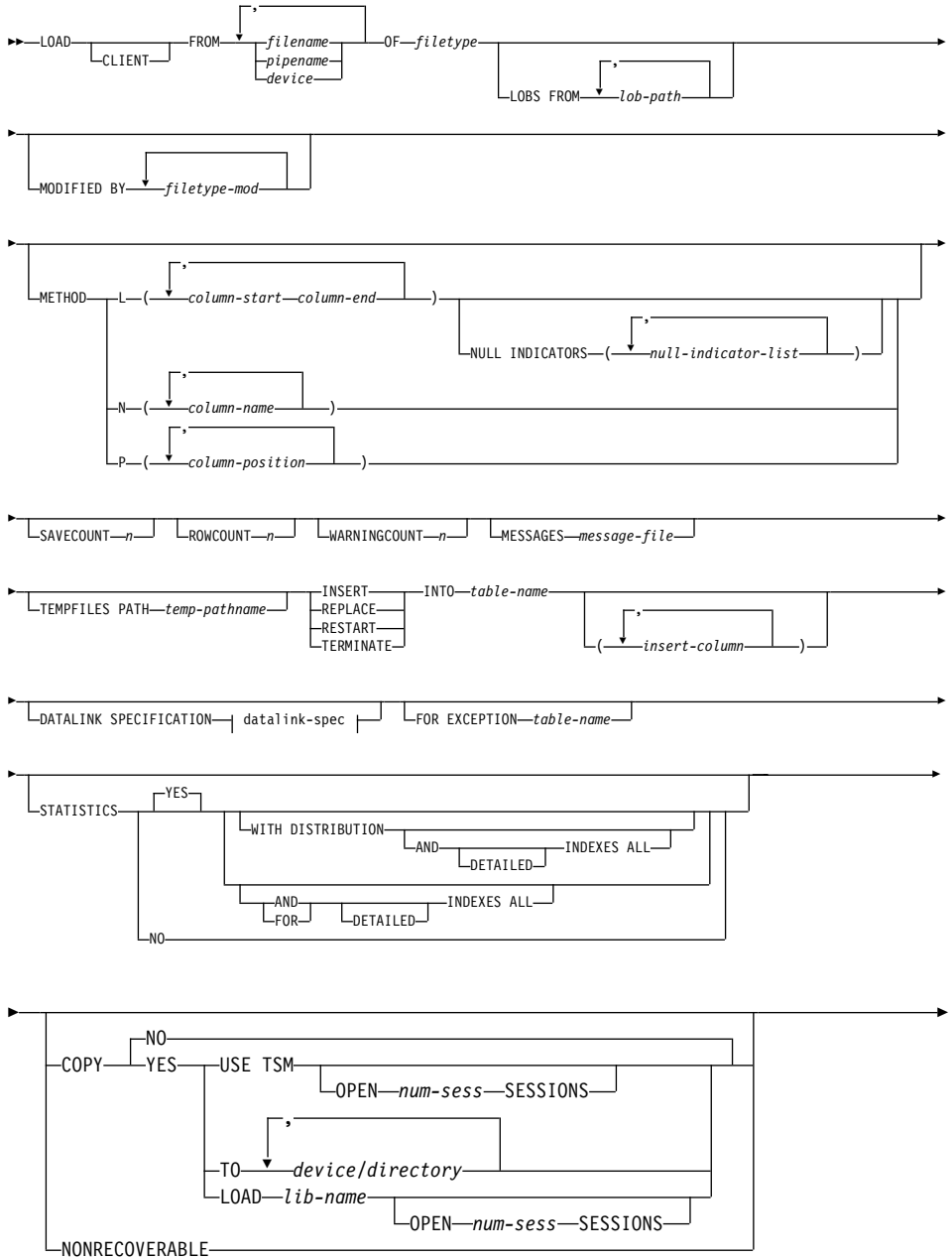
3. 널(NULL) 입력이 가능한 BUF와 SES를 제외한 모든 필드는 필수 필드입니다.
4. 매체 유형은 지역 장치(테이프, 디스크 또는 디스켓의 경우 L), Tivoli Storage Manager(A, 이전에는 ADSM) 또는 기타 벤더(O)를 사용할 수 있습니다. 매체 유형이 L일 경우, 위치 항목 앞에 위치 수가 필요합니다. A일 경우, 더 이상 입력이 필요하지 않습니다. 유형이 O이면, 공유 라이브러리 이름이 필요합니다. 백업 매체로서 Tivoli Storage Manager(TSM) 및 기타 벤더 제품에 대한 자세한 내용은 관리 안내서에 있는 데이터베이스 복구 장의 “Tivoli Storage Manager” 절을 참조하십시오.
5. SHRlib 매개변수는 LOAD COPY 데이터를 저장하기 위한 기능을 하는 라이브러리를 나타냅니다.

LOAD COPY NO를 수행하고 데이터베이스의 백업 사본 또는 로드 조작 수행 후 영향을 받은 테이블 공간이 없을 경우, 로드 조작보다 더 이후에 데이터베이스나 테이블 공간을 한 지점에 복원할 수 없습니다. 즉, 데이터베이스나 테이블 공간을 로드 조작 후의 상태로 재구축하기 위해 롤 포워드 복구를 사용할 수 없습니다. 로드 조작 이전 시간의 지점으로만 데이터베이스나 테이블 공간을 복원할 수 있습니다.

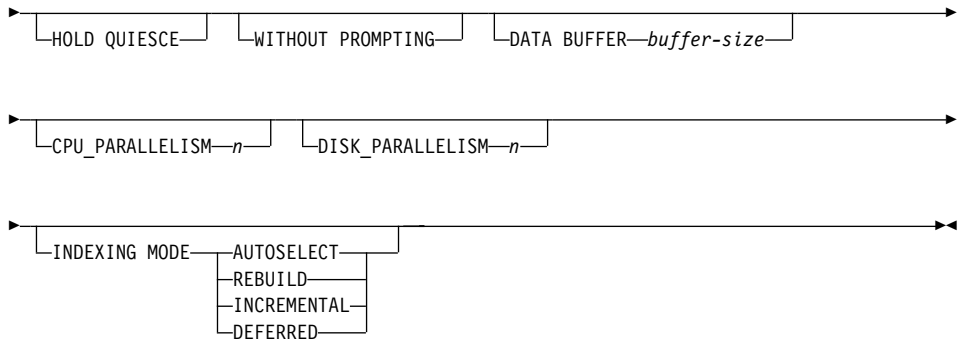
특정 로드 사본을 사용하려고 할 경우에는 로드 시간소인이 데이터베이스에 대한 복구 실행기록 파일에 기록됩니다. 파티션된 데이터베이스 환경에서 복구 실행기록 파일은 각 데이터베이스 파티션에 대해 지역입니다.

LOAD 명령

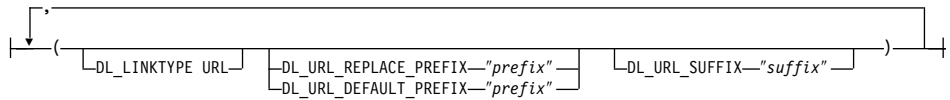
명령 구문



LOAD 명령



datalink-spec:



명령 매개변수

CLIENT

로드될 데이터가 원격으로 연결된 클라이언트에 상주함을 나타냅니다. 이 옵션은 원격 클라이언트에서 로드 작업을 호출하지 않을 경우에는 무시됩니다.

주:

- 다음과 같은 상황에서는 원격으로 연결된 클라이언트에 상주하는 데이터를 로드하는 작업을 지원하지 않습니다.
 - 클라이언트가 연결된 데이터베이스가 DB2 Enterprise - Extended Edition 환경 내에 있습니다.
 - 클라이언트가 연결된 데이터베이스가 이미 카탈로그화된 데이터베이스에 대해 카탈로그화됩니다.
- DUMPFILERE 및 LOBSINFILE 수정자(145 페이지의 표8 참조)는 CLIENT 키워드가 지정되어도 서버에 있는 파일들을 참조합니다.

3. 원격 로드 조작 동안에는 코드 페이지 변환이 수행되지 않습니다. 데이터의 코드 페이지가 서버의 코드 페이지와 다른 경우에는, CODEPAGE 수정자를 사용하여 데이터 코드 페이지를 지정해야 합니다(145 페이지의 표8 참조).

4. 파이프(Named Pipe는 제외)로부터 원격 클라이언트 데이터를 로드할 때는 파일당 상한이 2GB입니다.

다음 예에서는 원격으로 연결된 클라이언트에 상주하는 데이터 파일 (/u/user/data.del)이 서버 데이터베이스에 있는 MYTABLE로 로드됩니다.

```
db2 load client from /u/user/data.del of del
modified by codepage=850 insert into mytable
```

COPY NO

포워드 복구가 작동 가능한 경우(즉, *logretain* 또는 *userexit*가 켜져 있을 경우) 테이블이 상주하는 테이블 공간이 백업 보류 상태에 놓이게 됨을 지정합니다. 테이블 공간 백업이나 전체 데이터베이스 백업이 이루어질 때까지 테이블 공간 내의 테이블에 있는 데이터를 갱신하거나 삭제할 수 없습니다. 그러나, SELECT문을 사용하여 테이블에 있는 데이터를 액세스할 수는 있습니다.

COPY YES

로드된 데이터 사본이 저장되는 것을 지정합니다. 이 옵션은 포워드 복구가 작동 불가능한 경우(즉 *logretain*과 *userexit*가 모두 오프됨)에는 유효하지 않습니다. 이 옵션은 DATALINK 컬럼이 있는 테이블에 대해서는 지원되지 않습니다.

USE TSM

사본이 TSM을 사용하여 저장되는 것을 지정합니다.

OPEN num-sess SESSIONS

TSM 또는 벤더 제품과 함께 사용될 I/O 세션 수. 기본값은 1입니다.

TO device/directory

이미지 사본이 작성될 장치 또는 디렉토리를 지정합니다. 테이프

LOAD 명령

는 OS/2에서 지원되지 않으며, 테이프에 대한 사본은 SCO UnixWare 7에서 수행중인 DB2 서버에 대해 지원되지 않습니다.

LOAD lib-name

사용될 벤더의 백업 및 복원 I/O 기능을 포함하는 공유 라이브러리 이름(OS/2 또는 Windows 운영 체제상의 DLL). 이는 전체 경로를 포함할 수 있습니다. 전체 경로가 지정되지 않으면, User Exit 프로그램이 상주하는 경로를 기본 경로로 사용합니다.

CPU_PARALLELISM n

테이블 오브젝트 빌드시 로드 유틸리티가 레코드를 분석, 변환 및 형식화하기 위해 다량으로 만들어 낼 프로세스 또는 스레드의 수를 지정합니다. 이 매개변수는 파티션 내 병렬 처리를 이용하도록 설계되었습니다. 소스 데이터의 레코드 순서가 보존되기 때문에 사전 정렬된 데이터를 로드할 때 특히 도움이 됩니다. 이 매개변수 값이 0이거나 지정되지 않은 경우, 로드 유틸리티는 런타임시 지능적 기본값(사용가능한 CPU의 수에 기초하여)을 사용합니다.

주:

1. 이 매개변수가 LOB 또는 LONG VARCHAR 필드가 있는 테이블과 함께 사용될 경우, 값은 시스템 CPU의 수나 사용자가 지정한 값에 관계없이 하나가 됩니다.
2. SAVECOUNT 매개변수에 작은 값을 지정하면, 로더가 더 많은 I/O 조작을 수행하여 데이터와 테이블 메타 데이터를 모두 비웁니다. CPU_PARALLELISM이 1보다 크면, 비우기 조작은 비동기적이며, 로더의 CPU 이용을 허용합니다. CPU_PARALLELISM이 1로 설정되면, 로더는 일관성 지점에 있는 동안 I/O를 대기합니다. CPU_PARALLELISM을 2로 설정하고, SAVECOUNT를 10 000으로 설정한 로드 조작은 비록 한 개의 CPU만 있더라도 CPU_PARALLELISM을 1로 설정한 같은 조작보다 빨리 완료됩니다.

DATA BUFFER buffer-size

유틸리티에서 데이터를 전송하기 위한 버퍼 공간으로 사용할 4KB 페이지

의 수(병렬 처리의 등급과 관계없이)를 지정합니다. 지정된 값이 연산상의 최소값보다 작을 경우, 요구된 최소 자원이 사용되며, 경고는 리턴되지 않습니다.

이 메모리는 유틸리티 힙에서 직접 할당되는데, 이 유틸리티 힙의 크기는 `util_heap_sz` 데이터베이스 구성 매개변수를 통해 변경될 수 있습니다.

값이 지정되지 않으면, 런타임 유틸리티에 의해 지능적 기본값이 계산됩니다. 기본값은 테이블의 일부 특성과 같이 로더의 인스턴스 작성 시간에 유틸리티 힙에서 사용 가능한 여유 공간의 비율을 기준으로 합니다.

DATALINK SPECIFICATION

각 DATALINK 컬럼에 대해서는 괄호로 묶은 하나의 컬럼 스펙을 가질 수 있습니다. 각 컬럼 스펙은 하나 이상의 `DL_LINKTYPE`, 접두부 및 `DL_URL_SUFFIX` 스펙으로 구성됩니다. 접두부 스펙은 `DL_URL_REPLACE_PREFIX` 또는 `DL_URL_DEFAULT_PREFIX`가 될 수 있습니다.

테이블에 정의된 DATALINK 컬럼 수 만큼의 DATALINK 컬럼 스펙이 있을 수 있습니다. 스펙의 순서는 `insert-column` 목록이나 테이블 정의 (`insert-column` 목록이 정의되어 있지 않을 경우) 내에 있는 DATALINK 컬럼의 순서를 따릅니다.

DISK_PARALLELISM n

테이블 공간 컨테이너로 데이터를 기록하기 위해 로드 유틸리티가 다량으로 만들어 낼 프로세스 또는 스레드의 수를 지정합니다. 값이 지정되지 않으면, 유틸리티는 테이블 공간 컨테이너의 수와 테이블의 특성을 근거로 하여 지능적 기본값을 선택합니다.

DL_LINKTYPE

지정된 경우, 컬럼 정의의 `LINKTYPE`과 일치해야 합니다. 그러므로, `DL_LINKTYPE URL`은 컬럼 정의가 `LINKTYPE URL`을 지정할 경우 허용됩니다.

DL_URL_DEFAULT_PREFIX "prefix"

지정된 경우, 같은 컬럼 내의 모든 DATALINK 값에 대한 기본 접두부의 역할을 해야 합니다. 이 경우, 접두부는 URL 스펙의 "스키마 호스트

LOAD 명령

포트" 부분을 나타냅니다. 분산 파일 시스템(DFS)의 경우, 접두부는 URL 스펙의 "scheme cellname file-space-junction" 부분을 참조하십시오.

다음은 접두부의 예입니다.

```
"http://server"  
"file://server"  
"file:"  
"http://server:80"  
"dfs://.../cellname/fs"
```

컬럼 데이터에 접두부가 들어 있지 않고 기본 접두부가 DL_URL_DEFAULT_PREFIX로 지정되면, 기본 접두부가 컬럼 값 (NULL이 아닐 경우) 앞에 놓입니다.

예를 들어, DL_URL_DEFAULT_PREFIX가 기본 접두부 "http://toronto"를 지정하면,

- 컬럼 입력 값 "/x/y/z"는 "http://toronto/x/y/z"로 저장됩니다.
- 컬럼 입력 값 "http://coyote/a/b/c"는 "http://coyote/a/b/c"로 저장됩니다.
- 컬럼 입력 값 널(NULL)은 널(NULL)로 저장됩니다.

DL_URL_REPLACE_PREFIX "prefix"

이 절은 사용자가 데이터에 있는 호스트 이름을 다른 호스트 이름으로 전역 바꿀 때, 이전에 내보내기 유틸리티에 의해 생성된 데이터를 로드하거나 가져오는 데 유용합니다. 이 접두부를 지정하면, 이는 널(NULL)이 아닌 모든 컬럼 값에 대한 접두부가 됩니다. 컬럼 값에 접두부가 있으면, 이 접두부를 바꿉니다. 컬럼 값에 접두부가 없으면, DL_URL_REPLACE_PREFIX에 의해 지정된 접두부가 이 컬럼 값 앞에 놓입니다. 분산 파일 시스템(DFS)의 경우, 접두부는 URL 스펙의 "scheme cellname file-space-junction" 부분을 참조하십시오.

예를 들면, DL_URL_REPLACE_PREFIX가 접두부 "http://toronto"를 지정하면,

- 컬럼 입력 값 "/x/y/z"는 "http://toronto/x/y/z"로 저장됩니다.
 - 컬럼 입력 값 "http://coyote/a/b/c"는 "http://toronto/a/b/c"로 저장됩니다.
- 다. "toronto"가 "coyote"를 바꾸는 점을 주지하십시오.

- 컬럼 입력 값 널(NULL)은 널(NULL)로 저장됩니다.

DL_URL_SUFFIX "suffix"

이 접미부를 지정하면, 컬럼에 대한 널(NULL)이 아닌 모든 컬럼 값에 첨부됩니다. 실제로는 DATALINK 값의 데이터 위치 부분의 "경로" 구성요소에 첨부됩니다.

FOR EXCEPTION table-name

오류가 생긴 행이 복사될 예외 테이블을 지정합니다. 고유 색인이나 기본 키 색인이 잘못되어 있는 모든 행이 복사됩니다. DATALINK 예외사항도 예외 테이블에 캡처됩니다. 완전하지 않은 테이블 이름이 지정되면, 테이블은 CURRENT SCHEMA를 사용하여 완전하게 됩니다.

예외 테이블에 작성된 정보는 덤프 파일에 작성되지 않습니다(dumpfile 수정자에 대한 설명은 145 페이지의 표8 참조). 파티션된 데이터베이스 환경에서는 로딩 테이블이 정의된 노드에 대해 예외 테이블이 정의되어 있어야 합니다. 반면, 덤프 파일에는 유효하지 않거나 구문 오류가 있기 때문에 로드될 수 없는 행이 들어 있습니다. 159 페이지의 『예외 테이블』에서 자세한 내용을 참조하십시오.

FROM filename/pipename/device

로드 중인 데이터를 포함하는 파일, 파이프 또는 장치를 지정합니다. CLIENT 옵션이 지정되지 않으면, 이 파일, 파이프 또는 장치는 데이터베이스가 상주하는 노드에 상주해야 합니다. 여러 개의 이름이 지정되어 있으면 순서대로 처리됩니다. 지정된 마지막 항목이 테이프 장치일 경우, 사용자는 다른 테이프에 대해 프롬프트됩니다. 유효한 응답 옵션은 다음과 같습니다.

- c** 계속. 경고 메시지가 생성된 장치를 계속 사용합니다(예를 들어, 새로운 테이프가 마운트된 경우).
- d** 장치 종료. 경고 메시지가 생성된 장치의 사용을 중단합니다(예를 들어, 더 이상 테이프가 없을 경우).
- t** 종료. 모든 장치를 종료합니다.

주:

1. 테이프는 OS/2에서 지원되지 않습니다.

LOAD 명령

2. 완전한 파일 이름이 사용되는 것이 권장됩니다. 서버가 원격 서버일 경우, 완전한 파일 이름이 사용되어야 합니다. 데이터베이스가 호출자와 같은 노드에서 상주하면, 상대 경로가 사용될 수 있습니다.
3. 파일이 물리적으로는 별개이지만 논리적으로 하나의 파일로 된 경우, 여러 개의 IXF 파일로부터 데이터 로드가 지원됩니다. 파일이 논리적 및 물리적으로 모두 다를 경우에는 지원되지 않습니다.
4. OS/2에 *pipename*을 지정할 때 예상된 양의 데이터보다 적게 로드되면, 시스템 자원을 비우고(IPL이 권장됨), LOAD 명령을 재발행하십시오.
5. 클라이언트 머신에 상주하는 데이터를 로드할 경우에는 데이터가 완전한 파일 또는 Named Pipe의 양식으로 되어 있어야 합니다.

HOLD QUIESCE

유틸리티가 로드 조작 후 quiesce 독점 상태로 테이블에 남아야 함을 지정합니다. 테이블 공간을 quiesce 상태에서 해제하려면, 다음 명령을 발행하십시오.

```
db2 quiesce tablespaces for table <tablename> reset
```

주: *phantom quiesces*가 작성되지 않았는지 확인하십시오. *Command Reference*에서 자세한 내용을 참조하십시오.

INDEXING MODE

로드 유틸리티가 색인을 재빌드하고 점차적으로 확장하는지 지정하십시오. 유효한 값은 다음과 같습니다.

AUTOSELECT

로드 유틸리티가 REBUILD 또는 INCREMENTAL 모드 중에서 자동으로 결정합니다.

REBUILD

모든 색인이 재빌드됩니다. 유틸리티에는 구 테이블 데이터와 추가된 테이블 데이터에 대한 모든 색인 키 부분을 정렬하기 위해 충분한 자원이 있어야 합니다.

INCREMENTAL

새로운 데이터로 색인이 확장됩니다. 이 접근법은 색인 여유 공간

을 사용합니다. 삽입된 레코드에 대한 색인 키를 추가하기 위한 충분한 정렬 공간만 필요로 합니다. 이 방법은 색인 오브젝트가 유효하고 로드 조작 시작시 액세스할 수 있는 경우에만 지원됩니다 (예를 들어, DEFERRED 모드가 지정된 로드 조작 직후에는 유효하지 않습니다). 모드가 지정되어 있지만 색인 상태 때문에 지원되지 않을 경우에는 경고가 리턴되며, 로드 조작은 REBUILD 모드로 계속됩니다. 이처럼 로드 재시작 조작이 로드 빌드 단계에서 시작하면, INCREMENTAL 모드는 지원되지 않습니다.

점차적인 색인화는 다음의 모든 조건이 해당될 경우에는 지원되지 않습니다.

- LOAD COPY 옵션이 지정됨(*logretain* 또는 *userexit*가 작동 가능함)
- 테이블이 DMS 테이블 공간에 상주함
- 색인 오브젝트가 로드중인 테이블에 속한 다른 테이블 오브젝트에 의해 공유되는 테이블 공간에 상주함

이 제한사항을 무시하려면, 색인을 다른 테이블 공간에 배치하도록 권장합니다.

DEFERRED

로드 유틸리티는 이 모드가 지정되어 있으면 색인 작성을 시도하지 않습니다. 색인은 새로 고침이 필요하다고 표시됩니다. 로드 조작에 관련되지 않은 이러한 색인에 처음 액세스하면 재빌드가 강제되거나(자세한 정보는 *관리 안내서* 참조), 데이터베이스가 재시작될 때 색인이 재빌드될 수 있습니다. 이 접근 방법은 가장 큰 색인에 대한 모든 키 부분을 위한 충분한 정렬 공간을 요구합니다. 색인 구축을 위해 계속적으로 소요되는 전체 시간은 REBUILD 모드에 요구되는 시간보다 더 깁니다. 따라서, 유예 색인화로 복수 로드 조작을 수행할 때에는 로드하지 않는 최초 액세스때 색인이 재빌드되게 하는 것보다 순서의 마지막 로드 조작이 색인 빌드를 수행하게 하는 것이 유리합니다(성능 관점에서).

LOAD 명령

유예 색인화는 비고유 색인이 있는 테이블에 대해서만 지원되므로, 로드 단계에서 삽입된 중복 키는 로드 조작 후에는 영속적이지 않습니다.

주: DATALINK 컬럼이 있는 테이블에 대해서는 유예 색인화 (deferred indexing)가 지원되지 않습니다.

INSERT

로드 유틸리티가 실행할 수 있는 네 가지 모드 중 하나. 기존 테이블 데이터를 변경하지 않고 로드된 데이터를 테이블에 추가합니다.

insert-column

데이터가 삽입될 테이블 컬럼을 지정합니다.

로드 유틸리티는 이름에 하나 이상의 공백이 들어 있는 컬럼을 분석할 수 없습니다. 예를 들면,

```
db2 load from delfile1 of del modified by noeofchar noheader
method P (1, 2, 3, 4, 5, 6, 7, 8, 9)
insert into table1 (BLOB1, S2, I3, Int 4, I5, I6, DT7, I8, TM9)
```

위 예제는 Int 4 컬럼 때문에 실패합니다. 이에 대한 솔루션은 이 컬럼 이름을 다음과 같이 큰 따옴표로 묶는 것입니다.

```
db2 load from delfile1 of del modified by noeofchar noheader
method P (1, 2, 3, 4, 5, 6, 7, 8, 9)
insert into table1 (BLOB1, S2, I3, "Int 4", I5, I6, DT7, I8, TM9)
```

INTO table-name

데이터가 로드될 데이터베이스 테이블을 지정합니다. 이 테이블은 시스템 테이블이나 임시 테이블이 될 수 없습니다. 별명 또는 완전하거나 완전하지 않은 테이블 이름이 지정될 수 있습니다. 완전한 테이블 이름은 *schema.tablename* 형식입니다. 완전하지 않은 테이블 이름이 지정되면, 테이블은 CURRENT SCHEMA를 사용하여 완전하게 됩니다.

LOBS FROM lob-path

로드될 LOB 값을 포함하는 데이터 파일에 대한 경로. 경로는 슬래시(/)로 끝나야 합니다. CLIENT 옵션을 지정한 경우에는 경로가 완전해야 합니다. LOB 데이터 파일의 이름은 주 데이터 파일(ASC, DEL 또는 IXF)

에 저장되며, LOB 컬럼에 로드될 컬럼에 들어갑니다. 이 옵션은 `lobsinfile`이 `filetype-mod` 문자열에 지정되지 않은 경우에는 무시됩니다(145 페이지의 표8 참조).

MESSAGES message-file

로드 조작중 발생하는 경고 및 오류 메시지의 목적지를 지정합니다. 메시지 파일이 지정되어 있지 않을 경우, 메시지는 표준 출력으로 작성됩니다. 파일에 대한 완전 경로가 지정되어 있지 않을 경우, 로드 유틸리티는 현재 디렉토리와 기본 드라이브를 목적으로 사용합니다. 이미 있는 파일 이름이 지정되면, 이 유틸리티는 정보를 추가합니다.

메시지 파일은 로드 조작의 끝부분에 일반적으로 메시지와 함께 있으며 조작 진행의 모니터링과 같은 데는 적합하지 않습니다. 로드 조작의 실시간 모니터링에 대해서는 119 페이지의 『LOAD QUERY 명령』을 사용하십시오.

METHOD

L 데이터를 로드해 올 시작 및 끝 컬럼 번호를 지정합니다. 컬럼 번호는 데이터의 행 시작에서의 바이트 옵셋입니다. 번호는 1부터 매겨집니다.

주: 이 방법은 ASC 파일에서만 사용할 수 있으며, 이 파일 유형에 대해 유일한 유효 옵션입니다.

N 로드될 데이터 파일에 있는 컬럼 이름을 지정합니다. 이들 컬럼 이름의 대소문자는 시스템 카탈로그에 있는 해당 이름의 대소문자와 일치해야 합니다. 널(NULL)을 입력할 수 없는 각각의 테이블 컬럼은 METHOD N 목록에서 상응하는 항목을 가져야 합니다. 예를 들면, 주어진 데이터 필드 F1, F2, F3, F4, F5 및 F6 그리고 테이블 컬럼 C1 INT, C2 INT NOT NULL, C3 INT NOT NULL 및 C4 INT, `method N (F2, F1, F4, F3)`은 유효한 요청이고 반면에, `method N (F2, F1)`은 유효하지 않습니다.

주: 이 방법은 IXF 파일에서만 사용할 수 있습니다.

P 로드할 입력 데이터 필드의 색인(1부터 번호가 매겨짐)을 지정합니다. null이 될 수 없는 각각의 테이블 컬럼은 METHOD P 목

LOAD 명령

록에서 상응하는 항목을 가져야 합니다. 예를 들면, 주어진 데이터 필드 F1, F2, F3, F4, F5 및 F6 그리고 테이블 컬럼 C1 INT, C2 INT NOT NULL, C3 INT NOT NULL 및 C4 INT, method P (2, 1, 4, 3)는 유효한 요청이고 반면에, method P (2, 1)는 유효하지 않습니다.

주: 이 방법은 IXF 또는 DEL 파일에서만 사용할 수 있으며, DEL 파일 유형에 대해 유일한 유효 옵션입니다.

MODIFIED BY filetype-mod

추가 옵션을 지정합니다 (145 페이지의 표8).

NONRECOVERABLE

로드 트랜잭션이 복구 불가능으로 표시되고, 다음 롤 포워드 조치로 이를 복구할 수 없음을 지정합니다. 롤 포워드 유틸리티는 이 트랜잭션을 생각하고, 데이터가 로드되고 있는 테이블을 "유효하지 않은 테이블"로 표시합니다. 이 유틸리티는 이 테이블에 대한 모든 다음 트랜잭션도 무시합니다. 롤 포워드 조치가 완료되고 나면, 복구 불가능한 로드 조치가 완료된 다음에 나오는 확약점 이후에 취해진 백업(전체 또는 테이블 공간)으로부터 그와 같은 테이블만 드롭되거나 복원될 수 있습니다.

이 옵션의 사용으로, 테이블 공간이 로드 조작 후 백업 보류 상태로 되지 않고 로드된 데이터의 사본이 로드 조작 동안 작성될 필요가 없습니다.

이 옵션은 FILE LINK CONTROL 속성을 갖는 DATALINK 컬럼이 테이블에 있거나 추가될 때 사용되지 않아야 합니다.

NULL INDICATORS null-indicator-list

이 옵션은 METHOD L 매개변수가 지정될 때 즉, 입력 파일이 ASC 파일일 때만 사용할 수 있습니다. 널(NULL) 표시기 목록은 각 널(NULL) 표시기 필드의 컬럼 번호를 지정하는 양의 정수 목록으로 쉼표로 구분됩니다. 컬럼 번호는 데이터의 행 시작으로부터 널(NULL) 표시기 필드의 바이트 옵셋입니다. METHOD L 매개변수에 정의된 각 데이터 필드에 대한 널(NULL) 표시기 목록에 반드시 하나의 항목이 있어야 합니다. 컬럼 번호 0은 해당 데이터 필드에 항상 데이터가 들어 있음을 나타냅니다.

널(NULL) 표시기 컬럼의 Y 값은 컬럼 데이터가 널(NULL)임을 지정합니다. 널(NULL) 표시기 컬럼에서 Y 이외의 다른 문자는 컬럼 데이터가 널(NULL)이 아니고 METHOD L 옵션에 의해 지정된 컬럼 데이터가 로드되는 것을 나타냅니다.

널(NULL) 표시기 문자는 MODIFIED BY 옵션을 사용하여 변경할 수 있습니다(145 페이지의 표8의 nullindchar 수정자에 대한 설명 참조).

OF filetype

입력 파일에 있는 데이터의 형식을 다음과 같이 지정합니다.

- ASC(컬럼 식별자가 없는 ASCII 형식)
- DEL(컬럼 식별자가 있는 ASCII 형식)
- 동일한 DB2 테이블 또는 다른 DB2 테이블에서 내보낸 IXF(integrated exchange format, PC 버전)

이 파일의 형식에 대해서는 239 페이지의 『부록C. 내보내기/가져오기/로드 유틸리티 파일 형식』에서 자세한 내용을 참조하십시오.

REPLACE

로드 유틸리티가 실행할 수 있는 네 가지 모드 중 하나. 테이블에서 모든 기존 데이터를 삭제한 후 로드된 데이터를 삽입합니다. 테이블 정의와 색인 정의는 변경되지 않습니다. 계층 구조간에 데이터를 이동시킬 때 이 옵션을 사용한 경우, 각 서브테이블이 아닌 전체 계층 구조에 대한 데이터만 바꿀 수 있습니다.

이 옵션은 DATALINK 컬럼이 있는 테이블에 대해서는 지원되지 않습니다.

RESTART

로드 유틸리티가 실행할 수 있는 네 가지 모드 중 하나. 이전에 인터럽트된 로드 작업을 재시작합니다. 로드 작업은 로드, 빌드 또는 삭제 단계의 최종 일관성 지점부터 자동으로 계속됩니다.

RESTARTCOUNT

예약됨.

ROWCOUNT n

로드될 파일에 있는 n 물리 레코드의 수를 지정합니다. 사용자가 파일의 첫번째 n 행만 로드할 수 있도록 합니다.

SAVECOUNT n

로드 유틸리티가 모든 n 행 다음에 일관성 지점을 설정하도록 지정합니다. 이 값은 페이지 카운트로 변환되어 extent 크기의 간격으로 반올림됩니다. 메시지가 각 일관성 지점에서 발행되므로, 로드 조작은 119 페이지의 『LOAD QUERY 명령』을 사용하여 모니터링됩니다. n 값이 충분하지 않으면, 각 일관성 지점에서 수행된 활동의 동기화가 성능에 영향을 미치게 됩니다.

기본값은 필요하지 않는 한, 일관성 지점이 설정되지 않았다는 것을 나타내는 0입니다.

SORT BUFFER buffer-size

예약됨.

STATISTICS NO

통계가 수집되지 않으며, 키탈로그에 있는 통계가 변경되지 않음을 나타냅니다. 이는 기본값입니다.

STATISTICS YES

테이블 및 기존 색인에 대해 통계가 수집되는 것을 나타냅니다. 이 옵션은 로드 조작이 REPLACE 모드일 경우에만 지원됩니다.

WITH DISTRIBUTION

분산 통계가 수집되는 것을 지정합니다.

AND INDEXES ALL

테이블 및 색인 통계가 수집되는 것을 지정합니다.

FOR INDEXES ALL

색인 통계만 수집되는 것을 지정합니다.

DETAILED

확장된 색인 통계가 수집되는 것을 지정합니다.

TEMPFILES PATH temp-pathname

로드 조작중 임시 파일을 작성할 때 사용될 경로 이름을 지정하며, 서버 노드에 따라 완전한 이름이어야 합니다.

임시 파일은 파일 시스템 공간을 차지합니다. 때때로, 이러한 공간 요구는 매우 구체적입니다. 다음은 모든 임시 파일에 할당되어야 하는 파일 시스템 공간에 대한 예측입니다.

- DATALINK 값을 포함하는 각 중복 또는 거부된 행에 대해 4바이트
- 로드 유틸리티가 생성하는 각 메시지에 대해 136바이트
- 데이터 파일이 긴 필드 데이터나 LOB를 포함할 경우, 15KB 오버헤드. 이 수량은 INSERT 옵션이 지정되고, 많은 양의 긴 필드나 LOB 데이터가 이미 테이블에 있을 경우 더욱 늘어날 수 있습니다.

160 페이지의 『임시 파일 로드』에서 임시 파일에 관한 자세한 정보를 참조하십시오.

TERMINATE

로드 유틸리티가 실행할 수 있는 네 가지 모드 중 하나. 이전에 인터럽트된 로드 조작을 종료하고, 일관성 지점이 지났더라도 조작이 시작된 시간에 있던 지점으로 조작을 구간 복원합니다. 조작에 관련된 모든 테이블 공간의 상태는 정상으로 리턴하고, 모든 테이블 오브젝트가 일관성 있게 작성됩니다(색인 오브젝트는 유효하지 않은 것으로 표시될 수 있습니다. 이 경우, 다음 액세스시 색인이 자동으로 재빌드됩니다). 종료되는 로드 조작이 로드 REPLACE이면, 테이블은 로드 TERMINATE 조작 후 빈 테이블로 절단됩니다. 종료중인 로드 조작이 로드 INSERT이면, 테이블은 로드 TERMINATE 조작 후 모든 원래 레코드를 보존합니다.

로드 종료 옵션은 테이블 공간으로부터 백업 보류 상태를 제거하지 않습니다.

주: 이 옵션은 DATALINK 컬럼이 있는 테이블에 대해서는 지원되지 않습니다.

USING directory

예약됨.

WARNINGCOUNT *n*

n 경고 후 로드 작업을 중단합니다. 경고가 예상되지는 않지만 올바른 파일 및 테이블이 사용되고 있는지에 대한 검증이 필요할 경우, 이 매개변수를 설정하십시오. *n*이 0이거나 이 옵션이 지정되어 있지 않으면, 로드 작업은 발행된 경고 횟수를 무시하고 계속 진행됩니다. 경고 임계값에 도달했기 때문에 로드 작업이 중단될 경우, 다른 로드 작업을 RESTART 모드에서 시작할 수 있습니다. 로드 작업은 최종 일관성 지점부터 자동으로 계속됩니다. 그렇지 않으면, 다른 로드 작업이 REPLACE 모드로 입력 파일의 처음부터 시작될 수 있습니다.

WITHOUT PROMPTING

데이터 파일 목록에 로드될 모든 파일이 들어 있으며, 나열된 장치나 디렉토리가 전체 로드 작업에 충분하다는 것을 나타냅니다. 로드 작업이 종료되기 전에 입력 파일 계속이 발견되지 않거나 채워지면, 로드 작업이 실패하며 테이블은 로드 보류 상태가 됩니다.

이 옵션이 지정되지 않고, 테이프 장치가 사본 이미지에 대한 테이프의 끝에 이르거나 마지막에 나열된 항목이 테이프 장치일 경우, 사용자는 해당 장치의 새로운 테이프에 대해 프롬프트됩니다. 테이프는 OS/2에서 지원되지 않습니다.

LOAD QUERY 명령

처리중 로드 조작의 상태를 점검합니다. 동일한 데이터베이스에 대한 연결 및 다른 CLP 세션도 이 명령을 올바르게 호출하기 위해 필요합니다. 지역 또는 원격 사용자에 의해 사용될 수 있습니다.

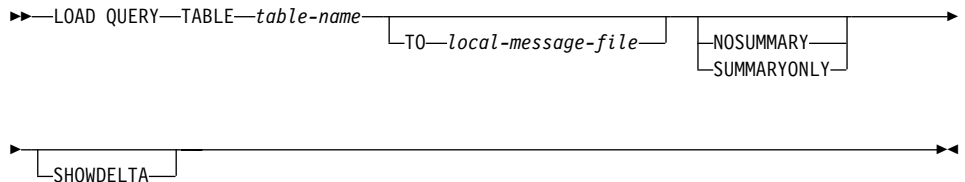
권한 부여

없음

필수 연결

데이터베이스

명령 구문



명령 매개변수

NOSUMMARY

로드 요약 정보(읽은 행, 건너뛴 행, 로드된 행, 거부된 행, 삭제된 행, 확장된 행 및 경고 수)가 보고되지 않음을 지정합니다.

SHOWDELTA

새로운 정보만(LOAD QUERY 명령의 최종 호출 이후 발생한 로드 이벤트에 대한) 보고되는 것을 지정합니다.

SUMMARYONLY

로드 요약 정보만 보고되는 것을 지정합니다.

TABLE table-name

데이터가 현재 로드중인 테이블 이름을 지정합니다. 완전하지 않은 테이블 이름이 지정되면, 테이블은 CURRENT SCHEMA를 사용하여 완전하게 됩니다.

LOAD QUERY 명령

TO local-message-file

로드 조작중 발생하는 경고 및 오류 메시지의 목적지를 지정합니다. 이 파일은 LOAD 명령에 대해 지정된 *message-file*이 될 수 없습니다. 이 파일이 이미 존재하고 있을 경우, 로드 유틸리티가 생성한 모든 메시지가 이 파일에 추가됩니다.

예

STAFF 테이블에 많은 양의 데이터를 로드하는 사용자는 로드 조작의 상태를 점검하는 것을 원합니다. 사용자는 다음과 같이 지정할 수 있습니다.

```
db2 connect to <database>
db2 load query table staff to /u/mydir/staff.tempsmg
```

출력 파일 /u/mydir/staff.tempsmg는 다음과 같이 나타날 수 있습니다.

```
SQL3500W The utility is beginning the "LOAD" phase at time
"02-13-1997 19:40:29.645353".
```

```
SQL3519W Begin Load Consistency Point. Input record count = "0".
```

```
SQL3520W Load Consistency Point was successful.
```

```
SQL3109N The utility is beginning to load data from file
"/u/mydir/data/staffbig.ixf".
```

```
SQL3150N The H record in the PC/IXF file has product "DB2 01.00",
date "19970111", and time "194554".
```

```
SQL3153N The T record in the PC/IXF file has name
"data/staffbig.ixf", qualifier " ", and source " ".
```

```
SQL3519W Begin Load Consistency Point. Input record count =
"111152".
```

```
SQL3520W Load Consistency Point was successful.
```

```
SQL3519W Begin Load Consistency Point. Input record count =
"222304".
```

```
SQL3520W Load Consistency Point was successful.
```

참조

103 페이지의 『LOAD 명령』.

로드 API

C API 구문

```
/* File: sqlutil.h */
/* API: Load */
/* ... */
SQL_API_RC SQL_API_FN
sqluload (
    sqlu_media_list * pDataFileList,
    sqlu_media_list * pLobPathList,
    struct sqldcol * pDataDescriptor,
    struct sqlchar * pActionString,
    char * pFileType,
    struct sqlchar * pFileTypeMod,
    char * pLocalMsgFileName,
    char * pRemoteMsgFileName,
    short CallerAction,
    struct sqluload_in * pLoadInfoIn,
    struct sqluload_out * pLoadInfoOut,
    sqlu_media_list * pWorkDirectoryList,
    sqlu_media_list * pCopyTargetList,
    sqlint32 * pNullIndicators,
    void * pReserved,
    struct sqlca * pSqlca);
/* ... */
```

일반 API 구문

```
/* File: sqlutil.h */
/* API: Load */
/* ... */
SQL_API_RC SQL_API_FN
sqlgload (
    unsigned short FileTypeLen,
    unsigned short LocalMsgFileNameLen,
    unsigned short RemoteMsgFileNameLen,
    sqlu_media_list * pDataFileList,
    sqlu_media_list * pLobPathList,
    struct sqlldcol * pDataDescriptor,
    struct sqlchar * pActionString,
    char * pFileType,
    struct sqlchar * pFileTypeMod,
    char * pLocalMsgFileName,
    char * pRemoteMsgFileName,
    short CallerAction,
    struct sqluload_in * pLoadInfoIn,
    struct sqluload_out * pLoadInfoOut,
    sqlu_media_list * pWorkDirectoryList,
    sqlu_media_list * pCopyTargetList,
    sqlint32 * pNullIndicators,
    void * pReserved,
    struct sqlca * pSqlca);
/* ... */
```

API 매개변수

FileTypeLen

입력. 파일 유형의 길이(바이트)를 나타내는 부호가 없는 2바이트 정수

LocalMsgFileNameLen

입력. 지역 메시지 파일 이름의 길이(바이트)를 나타내는 부호가 없는 2바이트 정수

RemoteMsgFileNameLen

입력. 임시 파일 경로 이름의 길이(바이트)를 나타내는 부호가 없는 2바이트 정수

pDataFileList

입력. 소스 파일, 장치, 벤더 또는 파이프 목록을 제공하는 데 사용하는 *sqlu_media_list* 구조에 대한 포인터. 타입은 OS/2에서 지원되지 않습니다.

이 구조에 제공되는 정보는 *media_type* 필드의 값에 따라 다릅니다. 유효한 값(sqlutil에 정의됨)은 다음과 같습니다.

SQLU_SERVER_LOCATION

media_type 필드가 이 값으로 설정되면, 호출자는 *sqlu_location_entry* 구조를 통해 정보를 제공합니다. *sessions* 필드는 제공된 *sqlu_location_entry* 구조의 수를 나타냅니다. 이는 파일, 장치 및 Named Pipe에 사용됩니다.

SQLU_CLIENT_LOCATION

media_type 필드가 이 값으로 설정되면, 호출자는 *sqlu_location_entry* 구조를 통해 정보를 제공합니다. *sessions* 필드는 제공된 *sqlu_location_entry* 구조의 수를 나타냅니다. 이는 완전한 파일 및 Named Pipe에 사용됩니다. 이 *media_type*이 API가 원격으로 연결된 클라이언트를 통해 호출될 때에만 유효합니다.

SQLU_TSM_MEDIA

media_type 필드가 이 값으로 설정될 경우, *sqlu_vendor* 구조가 사용되는데, 여기서 *filename*은 로드될 데이터에 대한 고유 식별자입니다. 여기에는 *sessions* 값과 관계없이 한 개의 *sqlu_vendor* 항목만 있어야 합니다. *sessions* 필드는 시작할 TSM 세션의 수를 나타냅니다. 로드 유틸리티는 다른 순차 번호를 갖지만, 하나의 *sqlu_vendor* 항목에 같은 데이터를 갖는 세션을 시작합니다.

SQLU_OTHER_MEDIA

media_type 필드가 이 값으로 설정될 경우, *shr_lib*가 공유 라이브러리 이름이고, *filename*이 로드될 데이터에 대한 고유 식별자인 *sqlu_vendor* 구조가 사용됩니다. 여기에는 *sessions* 값과 관계없이 한 개의 *sqlu_vendor* 항목만 있어야 합니다. *sessions* 필드

는 시작할 다른 벤더 세션의 수를 나타냅니다. 로드 유틸리티는 다른 순차 번호를 갖지만, 하나의 *sqlu_vendor* 항목에 같은 데이터를 갖는 세션을 시작합니다.

파일 이름이 제공될 때마다 이 파일 이름은 완전 규정화된 이름이어야 합니다. *Administrative API Reference*에 있는 “SQLU-MEDIA-LIST”에서 자세한 내용을 참조하십시오.

pLobPathList

입력. *sqlu_media_list* 구조에 대한 포인터. IXF, ASC 및 DEL 파일 유형의 경우, 각 LOB 파일이 로드될 위치를 식별하기 위한 완전한 경로나 장치의 목록. 이 파일 이름은 IXF, ASC 또는 DEL 파일에 있으며, 지정된 경로에 첨부되어 있습니다. 타입은 OS/2에서 지원되지 않습니다.

이 구조에 제공되는 정보는 *media_type* 필드의 값에 따라 다릅니다. 유효한 값(*sqlutil*에 정의됨)은 다음과 같습니다.

SQLU_LOCAL_MEDIA

이 값으로 설정되면, 호출자는 *sqlu_media_entry* 구조를 통해 정보를 제공합니다. *sessions* 필드는 제공된 *sqlu_media_entry* 구조의 수를 나타냅니다.

SQLU_TSM_MEDIA

이 값으로 설정되면, *filename*이 로드될 데이터에 대한 고유 식별자인 *sqlu_vendor* 구조가 사용됩니다. 여기에는 *sessions* 값과 관계없이 한 개의 *sqlu_vendor* 항목만 있어야 합니다. *sessions* 필드는 시작할 TSM 세션의 수를 나타냅니다. 로드 유틸리티는 다른 순차 번호를 갖지만, 하나의 *sqlu_vendor* 항목에 같은 데이터를 갖는 세션을 시작합니다.

SQLU_OTHER_MEDIA

media_type 필드가 이 값으로 설정될 경우, *shr_lib*가 공유 라이브러리 이름이고, *filename*이 로드될 데이터에 대한 고유 식별자인 *sqlu_vendor* 구조가 사용됩니다. 여기에는 *sessions* 값과 관계없이 한 개의 *sqlu_vendor* 항목만 있어야 합니다. *sessions* 필드

는 시작할 다른 벤더 세션의 수를 나타냅니다. 로드 유틸리티는 다른 순차 번호를 갖지만, 하나의 *sqlu_vendor* 항목에 같은 데이터를 갖는 세션을 시작합니다.

*Administrative API Reference*에 있는 “SQLU-MEDIA-LIST”에서 자세한 내용을 참조하십시오.

pDataDescriptor

입력. 외부 파일로부터 로딩하기 위해 선택되는 컬럼에 대한 정보를 포함하는 *sqldcol* 구조에 대한 포인터.

pFileType 매개변수가 SQL_ASC로 설정되면, 이 구조의 *dcolmeth* 필드가 SQL_METH_L로 설정되어야 합니다. 사용자는 로드될 각 컬럼에 대한 시작 및 끝 위치를 지정합니다.

파일 유형이 SQL_DEL이면, *dcolmeth*는 SQL_METH_P 또는 SQL_METH_D가 될 수 있습니다. SQL_METH_P일 경우, 사용자는 소스 컬럼 위치 (position)를 제공해야 합니다. SQL_METH_D이면, 이 파일의 첫번째 컬럼이 테이블의 첫번째 컬럼으로 로드되고, 계속 그 다음 컬럼이 로드됩니다.

파일 유형이 SQL_IXF이면, *dcolmeth*는 SQL_METH_P, SQL_METH_D 또는 SQL_METH_N 중 하나가 될 수 있습니다. SQL_METH_N이 파일 컬럼 이름이 *sqldcol* 구조로 제공되는 것을 나타내는 점을 제외하고는 DEL 파일에 대한 규칙이 여기에 적용됩니다.

*Administrative API Reference*에 있는 “SQLDCOL”에서 자세한 내용을 참조하십시오.

pActionString

입력. 테이블에 영향을 주는 조치를 지정하는 문자 배열 앞에 오는 2바이트 길이의 필드가 들어 있는 *sqlchar* 구조에 대한 포인터.

이 문자 배열은 다음과 같은 형식으로 되어 있습니다.

```
"INSERT|REPLACE|RESTART|TERMINATE
INTO tbnam [(column_list)]
[DATA LINK SPECIFICATION datalink-spec]
[FOR EXCEPTION e_tbnam]"
```

INSERT

기존 테이블 데이터를 변경하지 않고 로드된 데이터를 테이블에 추가합니다.

REPLACE

테이블에서 모든 기존 데이터를 삭제한 후 로드된 데이터를 삽입합니다. 테이블 정의와 색인 정의는 변경되지 않습니다.

RESTART

이전에 인터럽트된 로드 작업을 재시작합니다. 로드 작업은 로드, 빌드 또는 삭제 단계의 최종 일관성 지점부터 자동으로 계속됩니다.

TERMINATE

이전에 인터럽트된 로드 작업을 종료하고, 일관성 지점이 지났더라도 작업이 시작된 시간에 있던 지점으로 작업을 구간 복원합니다. 작업에 관련된 모든 테이블 공간의 상태는 정상으로 리턴하고, 모든 테이블 오브젝트가 일관성 있게 작성됩니다(색인 오브젝트는 유효하지 않은 것으로 표시될 수 있습니다. 이 경우, 다음 액세스 시 색인이 자동으로 재빌드됩니다). 테이블이 상주하는 테이블 공간이 로드 보류 상태가 아닐 경우, 이 옵션은 테이블 공간 상태에 영향을 주지 않습니다.

로드 종료 옵션은 테이블 공간으로부터 백업 보류 상태를 제거하지 않습니다.

tbname

데이터가 로드될 테이블 이름. 이 테이블은 시스템 테이블이나 선언된 임시 테이블이 될 수 없습니다. 별명 또는 완전하거나 완전하지 않은 테이블 이름이 지정될 수 있습니다. 완전한 테이블 이름은 *schema.tablename* 형식입니다. 완전하지 않은 테이블 이름이 지정되면, 테이블은 CURRENT SCHEMA를 사용하여 완전하게 됩니다.

(*column_list*)

데이터가 삽입될 테이블 컬럼 이름의 목록. 컬럼 이름은 쉼표로 구분되어야 합니다. 이름에 공간 또는 소문자가 들어 있으면, 작은 따옴표로 묶어야 합니다.

DATALINK SPECIFICATION *datalink-spec*

DB2 데이터 링크에 관한 매개변수를 지정합니다. 이 매개변수는 LOAD 명령어에서와 구문을 사용하여 지정할 수 있습니다(103 페이지의 『LOAD 명령』 참조).

FOR EXCEPTION *e_tbname*

오류가 생긴 행이 복사될 예외 테이블을 지정합니다. 고유 색인이거나 기본 키 색인이 잘못되어 있는 모든 행이 복사됩니다. DATALINK 예외사항도 예외 테이블에 캡처됩니다.

pFileType

입력. 외부 파일 내의 데이터 형식을 나타내는 문자열. 지원되는 외부 파일 형식(sqlutil에 정의됨)은 다음과 같습니다.

SQL_ASC

컬럼 식별자가 없는 ASCII

SQL_DEL

dBase, BASIC 및 IBM Personal Decision Series 프로그램과 기타 여러 데이터베이스 관리 프로그램 및 파일 관리 프로그램과의 교환을 위한 컬럼 식별자가 있는 ASCII

SQL_IXF

IXF(Integrated Exchange Format)의 PC 버전. 테이블로부터 데이터를 내보내기 위한 우선적 방법이며, 나중에 같은 테이블이나 다른 데이터베이스 관리 프로그램 테이블에 가져올 수 있습니다.

파일 형식에 관한 자세한 정보는 239 페이지의 『부록C. 내보내기/가져오기/로드 유틸리티 파일 형식』에서 자세한 내용을 참조하십시오.

pFileTypeMod

입력. 하나 이상의 처리 옵션을 지정하는 문자 배열 앞에 오는 2바이트 길

이의 필드가 들어 있는 구조에 대한 포인터. 이 포인터가 널(NULL)이거나 지정된 구조에 0 문자가 있으면, 이 조치는 기본 스펙 선택으로 해석됩니다.

모든 옵션이 지원되는 모든 파일 유형에 사용될 수는 없습니다.

*Administrative API Reference*에 있는 “SQLCHAR” 및 145 페이지의 『파일 유형 수정자(로드)』에서 자세한 내용을 참조하십시오.

pLocalMsgFileName

입력. 출력 메시지가 기록될 지역 파일 이름이 들어 있는 문자열

pRemoteMsgFileName

입력. 임시 파일을 위해 서버에서 사용될 경로 이름이 들어 있는 문자열. 임시 파일은 메시지, 일관성 지점 및 삭제 단계 정보를 저장하기 위해 작성됩니다. 160 페이지의 『임시 파일 로드』에서 임시 파일에 관한 자세한 정보를 참조하십시오.

CallerAction

입력. 호출자가 요청한 조치. 유효한 값(sqlutil에 정의됨)은 다음과 같습니다.

SQLU_INITIAL

초기 호출. 이 값(또는 SQLU_NOINTERRUPT)은 API에 대한 첫번째 호출에 사용되어야 합니다.

SQLU_NOINTERRUPT

초기 호출. 처리를 일시정지하지 않습니다. 이 값(또는 SQLU_INITIAL)은 API에 대한 첫번째 호출에 사용되어야 합니다.

초기 호출이나 이후 계속된 다른 호출이 리턴하여 호출 응용프로그램이 요청된 로드 작업을 완료하기 전에 일부 조치를 수행하도록 요구할 경우, 호출자 조치는 다음 중 하나로 설정되어야 합니다.

SQLU_CONTINUE

계속 처리. 이 값은 초기 호출이 사용자 입력을 요청하는 유틸리티(예: 테이프가 종료된 상태에 응답하도록) 상태와 함께 리턴한 후, API에 대한 계속된 호출에만 사용될 수 있습니다. 이는 유틸

리티가 요청한 사용자 조치가 완료되고, 유틸리티가 초기 요청을 계속 처리할 수 있다는 것을 지정합니다.

SQLU_TERMINATE

처리 종료. 로드중인 테이블 공간을 LOAD_PENDING 상태로 남겨둔 채, 로드 유틸리티를 미리 종료하게 합니다. 이 옵션은 데이터를 더 이상 처리하지 않을 경우 지정되어야 합니다.

SQLU_ABORT

처리 종료. 로드중인 테이블 공간을 LOAD_PENDING 상태로 남겨둔 채, 로드 유틸리티를 미리 종료하게 합니다. 이 옵션은 데이터를 더 이상 처리하지 않을 경우 지정되어야 합니다.

SQLU_RESTART

처리를 재시작함

SQLU_DEVICE_TERMINATE

단일 장치를 종료함. 이 옵션은 유틸리티가 장치로부터 데이터 읽기를 중단하지만, 데이터를 계속 처리할 경우 지정되어야 합니다.

pLoadInfoIn

입력. 추가 입력 매개변수를 포함하는 *sqluload_in* 구조에 대한 선택적 포인터. 132 페이지의 『데이터 구조: SQLULOAD-IN』에서 이 구조에 관한 정보를 참조하십시오.

pLoadInfoOut

출력. 추가 출력 매개변수를 포함하는 *sqluload_out* 구조에 대한 선택적 포인터. 137 페이지의 『데이터 구조: SQLULOAD-OUT』에서 이 구조에 관한 정보를 참조하십시오.

pWorkDirectoryList

예약됨.

pCopyTargetList

입력. 목표 경로, 장치 또는 사본 이미지가 작성될 공유 라이브러리의 목록을 제공하기 위해 사용되는(사본 이미지가 작성될 경우) *sqlu_media_list* 구조에 대한 포인터.

이 구조에 제공된 값은 *media_type* 필드 값에 따라 다릅니다. 이 필드에 대해 유효한 값(sqlutil에 정의됨)은 다음과 같습니다.

SQLU_LOCAL_MEDIA

이 사본이 지역 매체에 작성될 경우, *media_type*을 이 값으로 설정하고 *sqlu_media_entry* 구조에 목표에 관한 정보를 제공하십시오. *sessions* 필드는 제공된 *sqlu_media_entry* 구조의 수를 지정합니다.

SQLU_TSM_MEDIA

사본이 TSM으로 작성될 경우, 이 값을 사용하십시오. 더 이상 정보가 필요하지 않습니다.

SQLU_OTHER_MEDIA

벤더 제품이 사용될 경우, 이 값을 사용하고 *sqlu_vendor* 구조를 통해 정보를 더 제공하십시오. 이 구조의 *shr_lib* 필드를 벤더 제품의 공유 라이브러리 이름에 설정하십시오. *sessions* 값과 관계 없이 한 개의 *sqlu_vendor* 항목만 제공해야 합니다. *sessions* 필드는 제공된 *sqlu_media_entry* 구조의 수를 지정합니다. 로드 유틸리티는 다른 순차 번호를 갖지만, 하나의 *sqlu_vendor* 항목에 같은 데이터를 갖는 세션을 시작합니다.

자세한 정보는 *Administrative API Reference*에 있는 “SQLU-MEDIA-LIST”에서 자세한 내용을 참조하십시오.

pNullIndicators

입력. ASC 파일 전용. 컬럼 데이터가 널(NULL) 입력 가능한지의 여부를 나타내는 정수 배열. 이 배열의 요소와 데이터 파일에서 로드되는 컬럼은 1대1로 대응됩니다. 따라서, 요소의 수는 *pDataDescriptor* 매개변수의 *dcolnum* 필드와 동일해야 합니다. 각 배열 요소에는 널(NULL) 표시기 필드로 사용될 데이터 파일에 있는 위치를 식별하는 숫자 또는 테이블 컬럼이 널(NULL) 입력 불가능이라는 것을 나타내는 0을 포함합니다. 요소가 0이 아닐 경우, 데이터 파일에서 식별된 위치는 Y 또는 N을 포함해야 합니다. Y는 테이블 컬럼 데이터가 널(NULL)임을 나타내고, N는 테이블 컬럼 데이터가 널(NULL)이 아님을 나타냅니다.

pReserved

향후 사용을 위해 예약됨.

pSqlca

출력. *sqlca* 구조에 대한 포인터. *Administrative API Reference* 및 *SQL 참조서*에서 이 구조에 관한 자세한 정보를 참조하십시오.

REXX API 구문

이 API는 SQLDB2 인터페이스를 통해 REXX로부터 호출될 수 있습니다. 응용 프로그램 개발 안내서에서 자세한 내용을 참조하십시오. 103 페이지의 『LOAD 명령』에서 구문 설명에 대한 자세한 내용을 참조하십시오.

데이터 구조: SQLLOAD-IN

이 구조는 121 페이지의 『로드 API』로의 호출시 정보를 입력하기 위해 사용됩니다.

표 6. SQLLOAD-IN 구조의 필드

필드 이름	데이터 유형	설명
SIZEOFSTRUCT	sqluint32	이 구조의 크기(바이트)
SAVECNT	sqluint32	일관성 지점을 설정하기 전에 로드할 레코드의 수. 이 값은 페이지 카운트로 변환되어 extent 크기의 간격으로 반올림됩니다. 메시지가 각 일관성 지점에서 발행되므로, 139 페이지의 『db2LoadQuery - Load Query API』. <i>savecnt</i> 값이 충분하지 않을 경우, 각 일관성 지점에서 수행된 활동의 동기화는 성능에 영향을 줍니다. 기본값은 필요하지 않는 한, 일관성 지점이 설정되지 않음을 나타내는 0입니다.
RESTARTCNT	sqluint32G	예약됨.
ROWCNT	sqluint32	로드될 물리적 레코드의 수. 사용자가 파일의 첫번째 <i>rowcnt</i> 행만 로드할 수 있도록 합니다.
WARNINGCNT	sqluint32	<i>warningcnt</i> 경고 후 로드 조작을 중단합니다. 경고가 예상되지는 않지만 올바른 파일 및 테이블이 사용되고 있는지에 대한 검증이 필요할 경우, 이 매개변수를 설정하십시오. <i>warningcnt</i> 가 0이거나 이 옵션이 지정되어 있지 않으면, 로드 조작은 발행된 경고 횟수를 무시하고 계속 진행됩니다. 경고의 임계값이 초과되었기 때문에 로드 조작이 중단될 경우, 다른 로드 조작을 RESTART 모드에서 시작할 수 있습니다. 로드 조작은 최종 일관성 지점부터 자동으로 계속됩니다. 그렇지 않으면, 다른 로드 조작이 REPLACE 모드로 입력 파일의 처음부터 시작될 수 있습니다.
DATA_BUFFER_SIZE	sqluint32	유틸리티에서 데이터를 전송하기 위한 버퍼 공간으로 사용할 4KB 페이지의 수(병렬 처리의 등급과 관계없이). 지정된 값이 연산상의 최소값보다 작을 경우, 요구된 최소값이 사용되며 경고는 리턴되지 않습니다. 이 메모리는 유틸리티 힙에서 직접 할당되는데, 이 유틸리티의 힙의 크기는 <i>util_heap_sz</i> 데이터베이스 구성 매개변수를 통해 변경될 수 있습니다. 값이 지정되지 않으면, 런타임 유틸리티에 의해 지능적 기본값이 계산됩니다. 기본값은 테이블의 일부 특성과 같이 로더의 인스턴스 작성 시간에 유틸리티 힙에서 사용 가능한 여유 공간의 비율을 기준으로 합니다.

표 6. SQLULOAD-IN 구조의 필드 (계속)

필드 이름	데이터 유형	설명
SORT_BUFFER_SIZE	sqluint32	예약됨.
HOLD QUIESCE	UNSIGNED SHORT	유틸리티가 로드 조작 후 테이블을 배타적인 quiesce 상태로 두고자 할 경우 값이 TRUE로 설정되고, 그렇지 않으면 FALSE로 설정되는 플래그
RESTARTPHASE	CHAR(1)	예약됨.
STATSOPT	CHAR(1)	수집할 통계의 입도. 이 값에 대해서는 아래를 참조하십시오.
CPU_PARALLELISM	UNSIGNED SHORT	테이블 오브젝트 빌드시 로드 유틸리티가 레코드를 분석, 변환 및 형식화하기 위해 로드 유틸리티가 다량으로 만들어 낼 프로세스 또는 스레드의 수. 이 매개변수는 파티션 내 병렬 처리를 이용하도록 설계되었습니다. 소스 데이터의 레코드 순서가 보존되기 때문에 사전 정렬된 데이터를 로드할 때 특히 도움이 됩니다. 이 매개변수 값이 0일 경우, 유틸리티는 런타임시 지능적 기본값을 사용합니다. 주: 이 매개변수가 LOB 또는 LONG VARCHAR 필드가 있는 테이블과 함께 사용될 경우, 이 값은 시스템 CPU의 수 또는 사용자가 지정한 값과 관계없이 하나가 됩니다.
DISK_PARALLELISM	UNSIGNED SHORT	테이블 공간 컨테이너로 데이터를 기록하기 위해 로드 유틸리티가 다량으로 만들어 낼 프로세스 또는 스레드의 수. 값이 지정되지 않으면, 유틸리티는 테이블 공간 컨테이너의 수와 테이블의 특성을 근거로 하여 지능적 기본값을 선택합니다.
NON_RECOVERABLE	UNSIGNED SHORT	로드 트랜잭션이 복구 불가능으로 표시되고, 다음 물 포워드 조치로 복구할 수 없을 경우, SQLU_NON_RECOVERABLE_LOAD로 설정하십시오. 물 포워드 유틸리티는 이 트랜잭션을 생략하고, 데이터가 로드 중인 테이블을 "유효하지 않은 테이블"로 표시합니다. 이 유틸리티는 이 테이블에 대한 모든 다음 트랜잭션도 무시합니다. 물 포워드가 완료되면, 이러한 테이블은 삭제만 가능합니다. 이 옵션의 사용으로, 테이블 공간이 로드 조작 후 백업 보류 상태로 되지 않고 로드된 데이터의 사본이 로드 조작 동안 작성될 필요가 없습니다. 로드 트랜잭션을 복구 가능으로 표시하려면 SQLU_RECOVERABLE_LOAD로 설정하십시오.
INDEXING_MODE	UNSIGNED SHORT	로드 유틸리티가 색인을 재빌드하고 점차적으로 확장하는지 지정하십시오. 이 값에 대해서는 아래를 참조하십시오.

STATSOPT에 대해 유효한 값(sqlutil에 정의됨)은 다음과 같습니다.

SQLU_STATS_NONE

데이터 구조: SQLULOAD-IN

SQL_STATS_EXTTABLE_ONLY

SQL_STATS_EXTTABLE_INDEX

SQL_STATS_INDEX

SQL_STATS_TABLE

SQL_STATS_EXTINDEX_ONLY

SQL_STATS_EXTINDEX_TABLE

SQL_STATS_ALL

SQL_STATS_BOTH

| *INDEXING_MODE*에 대한 유효한 값(sqlutil에 정의됨)은 다음과 같습니다.

| **SQLU_INX_AUTOSELECT**

| **SQLU_INX_REBUILD**

| **SQLU_INX_INCREMENTAL**

| **SQLU_INX_DEFERRED**

| 이러한 인덱싱 모드의 설명에 대해서는 103 페이지의 『LOAD 명령』에서 자세한
| 내용을 참조하십시오.

언어 구문

C 구조

```

/* File: sqlutil.h */
/* Structure: SQLLOAD-IN */
/* ... */
SQL_STRUCTURE sqlload_in
{
    sqluint32      sizeofStruct;
    sqluint32      savecnt;
    sqluint32      restartcnt;
    sqluint32      rowcnt;
    sqluint32      warningcnt;
    sqluint32      data_buffer_size;
    sqluint32      sort_buffer_size; /* No longer used. */
    unsigned short hold_quiesce;
    char           restartphase;
    char           statsopt;
    unsigned short cpu_parallelism;
    unsigned short disk_parallelism;
    unsigned short non_recoverable;
    unsigned short indexing_mode;
};
/* ... */

```

COBOL 구조

```

* File: sqlutil.cbl
01 SQLLOAD-IN.
   05 SQL-SIZE-OF-STRUCT      PIC 9(9) COMP-5 VALUE 40.
   05 SQL-SAVECNT            PIC 9(9) COMP-5.
   05 SQL-RESTARTCOUNT     PIC 9(9) COMP-5.
   05 SQL-ROWCNT            PIC 9(9) COMP-5.
   05 SQL-WARNINGCNT       PIC 9(9) COMP-5.
   05 SQL-DATA-BUFFER-SIZE  PIC 9(9) COMP-5.
   05 SQL-SORT-BUFFER-SIZE  PIC 9(9) COMP-5. * No longer used.
   05 SQL-HOLD-QUIESCE     PIC 9(4) COMP-5.
   05 SQL-RESTARTPHASE     PIC X.
   05 SQL-STATSOPT        PIC X.
   05 SQL-CPU-PARALLELISM  PIC 9(4) COMP-5.
   05 SQL-DISK-PARALLELISM PIC 9(4) COMP-5.
   05 SQL-NON-RECOVERABLE  PIC 9(4) COMP-5.
   05 SQL-INDEXING-MODE    PIC 9(4) COMP-5.

```

*

데이터 구조: SQLLOAD-IN

데이터 구조: SQLLOAD-OUT

이 구조는 121 페이지의 『로드 API』로의 호출 이후 정보를 입력하기 위해 사용됩니다.

표 7. SQLLOAD-OUT 구조의 필드

필드 이름	데이터 유형	설명
SIZEOFSTRUCT	sqluint32	이 구조의 크기(바이트)
ROWSREAD	sqluint32	로드 조작중 읽은 레코드의 수
ROWSSKIPPED	sqluint32	로드 조작이 시작하기 전에 건너편 레코드의 수
ROWSLOADED	sqluint32	목표 테이블로 로드된 행의 수
ROWSREJECTED	sqluint32	로드될 수 없는 레코드의 수
ROWSDELETED	sqluint32	삭제된 중복 행의 수
ROWSCOMMITTED	sqluint32	처리된 전체 레코드의 수. 데이터베이스에 로드되어 확장된 레코드의 수와 건너편 레코드 및 거부된 레코드의 수를 합한 수

언어 구문

C 구조

```

/* File: sqlutil.h */
/* Structure: SQLLOAD-OUT */
/* ... */
SQL_STRUCTURE sqlload_out
{
    sqluint32    sizeOfStruct;
    sqluint32    rowsRead;
    sqluint32    rowsSkipped;
    sqluint32    rowsLoaded;
    sqluint32    rowsRejected;
    sqluint32    rowsDeleted;
    sqluint32    rowsCommitted;
};
/* ... */

```

데이터 구조: SQLLOAD-OUT

COBOL 구조

```
* File: sqlutil.cbl
01 SQLLOAD-OUT.
   05 SQL-SIZE-OF-STRUCT      PIC 9(9) COMP-5 VALUE 28.
   05 SQL-ROWS-READ          PIC 9(9) COMP-5.
   05 SQL-ROWS-SKIPPED      PIC 9(9) COMP-5.
   05 SQL-ROWS-LOADED       PIC 9(9) COMP-5.
   05 SQL-ROWS-REJECTED     PIC 9(9) COMP-5.
   05 SQL-ROWS-DELETED      PIC 9(9) COMP-5.
   05 SQL-ROWS-COMMITTED    PIC 9(9) COMP-5.
*
```

db2LoadQuery - Load Query API

처리중 로드 조작의 상태를 점검합니다.

권한 부여

없음

필수 연결

데이터베이스

API Include 파일

db2ApiDf.h

C API 구분

```
/* File: db2ApiDf.h */
/* API: Load Query */
/* ... */
SQL_API_RC SQL_API_FN
db2LoadQuery (
    db2UInt32 versionNumber,
    void * pParmStruct,
    struct sqlca *pSqlca);

typedef struct
{
    db2UInt32 iStringType;
    char * piString;
    db2UInt32 iShowLoadMessages;
    db2LoadQueryOutputStruct * poOutputStruct;
    char * piLocalMessageFile;
} db2LoadQueryStruct;

typedef struct
{
    db2UInt32 oRowsRead;
    db2UInt32 oRowsSkipped;
    db2UInt32 oRowsCommitted;
    db2UInt32 oRowsLoaded;
    db2UInt32 oRowsRejected;
    db2UInt32 oRowsDeleted;
    db2UInt32 oCurrentIndex;
    db2UInt32 oNumTotalIndexes;
    db2UInt32 oCurrentMPPNode;
    db2UInt32 oLoadRestarted;
    db2UInt32 oWhichPhase;
    db2UInt32 oWarningCount;
} db2LoadQueryOutputStruct;
/* ... */
```


일반 API 구문

```

/* File: db2ApiDf.h */
/* API: Load Query */
/* ... */
SQL_API_RC SQL_API_FN
db2gLoadQuery (
    db2UInt32 versionNumber,
    void * pParmStruct,
    struct sqlca *pSqlca);

typedef struct
{
    db2UInt32 iStringType;
    db2UInt32 iStringLen;
    char * piString;
    db2UInt32 iShowLoadMessages;
    db2LoadQueryOutputStruct * poOutputStruct;
    db2UInt32 iLocalMessageFileLen;
    char * piLocalMessageFile
} db2gLoadQueryStruct;

typedef struct
{
    db2UInt32 oRowsRead;
    db2UInt32 oRowsSkipped;
    db2UInt32 oRowsCommitted;
    db2UInt32 oRowsLoaded;
    db2UInt32 oRowsRejected;
    db2UInt32 oRowsDeleted;
    db2UInt32 oCurrentIndex;
    db2UInt32 oNumTotalIndexes;
    db2UInt32 oCurrentMPPNode;
    db2UInt32 oLoadRestarted;
    db2UInt32 oWhichPhase;
    db2UInt32 oWarningCount;
} db2LoadQueryOutputStruct;
/* ... */

```

API 매개변수

versionNumber

입력. 통과된 구조의 버전 및 릴리스 레벨을 두번째 매개변수 *pParmStruct* 로서 지정합니다.

pParmStruct

입력. *db2LoadQueryStruct* 구조에 대한 포인터.

pSqlca

출력. *sqlca* 구조에 대한 포인터. *Administrative API Reference* 및 *SQL 참조서*에서 이 구조에 관한 자세한 정보를 참조하십시오.

iStringType

입력. *piString*에 대한 유형을 지정합니다. 유효한 값(*db2ApiDf.h*에 정의됨)은 다음과 같습니다.

DB2LOADQUERY_TABLENAME

db2LoadQuery API의 사용을 위한 테이블 이름 지정을 나타냅니다.

iStringLen

입력. *piString*의 길이(바이트)를 지정합니다.

piString

입력. *iStringType* 값에 따라 임시 파일 경로 이름이나 테이블 이름을 지정합니다.

iShowLoadMessages

입력. 로드 유틸리티에 의해 리턴될 메시지 레벨을 지정합니다. 유효한 값(*db2ApiDf.h*에 정의됨)은 다음과 같습니다.

DB2LOADQUERY_SHOW_ALL_MSGS

모든 로드 메시지를 리턴합니다.

DB2LOADQUERY_SHOW_NO_MSGS

로드 메시지를 리턴하지 않습니다.

DB2LOADQUERY_SHOW_NEW_MSGS

이 API에 대해 마지막 호출이 일어난 후에 생성된 메시지만 리턴합니다.

poOutputStruct

출력. 로드 요약 정보를 포함하는 *db2LoadQueryOutputStruct* 구조에 대한 포인터. 요약이 필요하지 않으면 널(NULL)로 설정합니다.

iLocalMessageFileLen

입력. *piLocalMessageFile*의 길이(바이트)를 지정합니다.

piLocalMessageFile

입력. 출력 메시지에 사용될 지역 파일 이름을 지정합니다.

oRowsRead

출력. 이제까지 로드 유틸리티가 읽은 레코드의 수

oRowsSkipped

출력. 로드 조작을 시작하기 전에 건너뛴 레코드의 수

oRowsCommitted

출력. 이제까지 목표 테이블에 확약된 행의 수

oRowsLoaded

출력. 이제까지 목표 테이블로 로드된 행의 수

oRowsRejected

출력. 이제까지 목표 테이블로부터 거부된 행의 수

oRowsDeleted

출력. 이제까지 목표 테이블에서 삭제된 행의 수(삭제 단계 동안)

oCurrentIndex

출력. 현재 빌드중인 색인(빌드 단계 동안)

oCurrentMPPNode

출력. 어느 노드가 조회되고 있는지를 나타냅니다(MPP 모드에 대해서만).

oLoadRestarted

출력. 조회중인 로드 조작이 로드 재시작 조작일 경우, TRUE 값을 갖는 플래그

db2LoadQuery - Load Query API

oWhichPhase

출력. 조회중인 로드 조작의 현재 단계를 나타냅니다. 유효한 값 (db2ApiDf.h에 정의됨)은 다음과 같습니다.

DB2LOADQUERY_LOAD_PHASE

로드 단계

DB2LOADQUERY_BUILD_PHASE

빌드 단계

DB2LOADQUERY_DELETE_PHASE

삭제 단계

oNumTotalIndexes

출력. 빌드될 전체 색인 수(빌드 단계 동안)

oWarningCount

출력. 이제까지 리턴된 전체 경고 수

REXX API 구문

이 API는 SQLDB2 인터페이스를 통해 REXX로부터 호출될 수 있습니다. 응용 프로그램 개발 안내서에서 자세한 내용을 참조하십시오. 119 페이지의 『LOAD QUERY 명령』에서 구문 설명에 대한 자세한 내용을 참조하십시오.

샘플 프로그램

C \sqllib\samples\c\loadqry.sqc

COBOL \sqllib\samples\cobo\loadqry.sqb

사용시 주의사항

이 API는 *piString*가 지정하는 테이블상의 로드 조작 상태를 읽어 *pLocalMsgFileName*이 지정하는 파일에 상태를 기록합니다.

파일 유형 수정자(로드)

표 8. 유효한 파일 유형 수정자(로드)

수정자	설명
모든 파일 형식	
anyorder	이 수정자는 <code>cpu_parallelism</code> 매개변수와 함께 사용됩니다. SMP 시스템에서 중요한 부가적 성능 이점을 갖는, 소스 데이터 순서를 보존할 필요가 없음을 지정합니다. <code>cpu_parallelism</code> 값이 1일 경우, 이 옵션은 무시됩니다. 이 옵션은 <code>SAVECOUNT > 0</code> 일 경우 지원되지 않습니다. 왜냐하면, 일관성 지점 다음의 고장 복구가 데이터를 순서대로 로드할 것을 요구하기 때문입니다.
fastparse	<p>사용자 제공 컬럼 값에 대해 축소된 구문 점검을 수행하며, 성능이 향상됩니다. 이 옵션하에서 로드된 테이블은 체계적으로 정확하다는 것이 보증되며, 유틸리티는 파티션 위반이나 트랩을 막기 위해 반드시 충분한 데이터 점검을 수행합니다. 정확한 형식으로 된 데이터는 정확하게 로드됩니다.</p> <p>예를 들어, 123qwr4 값이 ASC 파일에 있는 정수 컬럼에 대한 필드 항목으로 인식되면, 로드 유틸리티는 값이 유효 숫자를 나타내지 않기 때문에 대개 구문 오류로 플래그합니다. <code>fastparse</code>의 경우, 구문 오류는 검출되지 않으며, 임의 숫자가 정수 필드에 로드됩니다. 반드시 데이터를 지우고 이 수정자를 사용하도록 주의하십시오. ASCII 데이터에서 이 옵션을 사용하면 성능이 매우 향상될 수 있지만, <code>fastparse</code>는 PC/IXF 데이터에서는 성능을 많이 향상시키지 않습니다. 왜냐하면, IXF가 2진 형식이고 <code>fastparse</code>가 ASCII에서 내부 형식으로의 변환과 분석에 영향을 미치기 때문입니다.</p>
generatedignore	이 수정자는 생성된 모든 컬럼에 대한 데이터가 데이터 파일에 존재하지만 이를 무시하도록 로드 유틸리티에 통지합니다. 널(NULL) 입력 가능한 생성된 컬럼의 경우에는 이 결과 NULL이 컬럼으로 로드되며, 널(NULL) 입력 가능하지 않은 생성된 컬럼의 경우에는 이 결과 생성된 컬럼의 데이터 유형에 대한 기본값이 로드됩니다. 로드 조작의 끝부분에서, SET INTEGRITY문은 로드된 값을 생성된 컬럼 정의에 따라 계산된 값으로 강제로 바꾸기 위해 호출될 수 있습니다. 이 수정자는 <code>generatedmissing</code> 또는 <code>generatedoverride</code> 수정자와 함께 사용할 수 없습니다.
generatedmissing	이 수정자가 지정되면 유틸리티는 입력 데이터 파일에 생성된 컬럼에 대한 데이터가 없다고(널(NULL) 까지도) 가정하고 컬럼에 NULL을 로드합니다. 로드 조작의 끝부분에서, SET INTEGRITY문은 NULL을 생성된 컬럼 정의에 따라 계산된 값으로 강제로 바꾸기 위해 사용될 수 있습니다. 이 수정자는 <code>generatedignore</code> 또는 <code>generatedoverride</code> 수정자와 함께 사용할 수 없습니다.

파일 유형 수정자(로드)

표 8. 유효한 파일 유형 수정자(로드) (계속)

수정자	설명
generatedoverride	<p>이 수정자는 테이블에 있는 생성된 모든 컬럼에 대해 명시적이면서 NULL이 아닌 데이터를 받아들이도록 로드 유틸리티에 지시합니다(이러한 유형의 컬럼에 대한 일반 규칙에 상반됨). 이는 다른 데이터베이스 시스템에서 데이터를 이주할 때 또는 ROLLFORWARD DATABASE 명령의 DROPPED TABLE RECOVERY 옵션을 사용하여 복구된 데이터로부터 테이블을 로드할 때 매우 유익할 수 있습니다. 이 수정자를 사용할 때는 널(NULL) 입력 가능하지 않은 생성된 컬럼에 대해 데이터가 없거나 NULL 데이터가 있는 행들은 거부됩니다(SQL3116W).</p> <p>주: 이 옵션을 사용하면 로드 유틸리티가 생성된 컬럼값의 유효성 검사를 시도하지 않습니다.</p> <p>이 수정자는 generatedmissing 또는 generatedignore 수정자와 함께 사용할 수 없습니다.</p>
identityignore	<p>이 수정자는 식별 컬럼에 대한 데이터가 데이터 파일에 존재하지만 이를 무시하도록 로드 유틸리티에 통지합니다. 그 결과 모든 식별값이 유틸리티에 의해 생성됩니다. 작동은 GENERATED ALWAYS 및 GENERATED BY DEFAULT 식별 컬럼 둘다에 대해 동일합니다. 이는 GENERATED ALWAYS 컬럼의 경우에 아무 행동도 거부되지 않음을 의미합니다. 이 수정자는 identitymissing 또는 identityoverride 수정자와 함께 사용할 수 없습니다.</p>
identitymissing	<p>이 수정자가 지정되면 유틸리티는 입력 데이터 파일에 식별 컬럼에 대한 데이터가 없다고(널(NULL) 까지도) 가정하고 각 행에 대해 값을 생성합니다. 작동은 GENERATED ALWAYS 및 GENERATED BY DEFAULT 식별 컬럼 둘다에 대해 동일합니다. 이 수정자는 identityignore 또는 identityoverride 수정자와 함께 사용할 수 없습니다.</p>
identityoverride	<p>이 수정자는 GENERATED ALWAYS로 정의된 식별 컬럼이 로드될 테이블에 존재할 경우에만 사용해야 합니다. 이는 그러한 컬럼에 대해 명시적이면서 NULL이 아닌 데이터를 받아들이도록 유틸리티에 지시합니다(이러한 유형의 식별 컬럼에 대한 일반 규칙에 상반됨). 이는 다른 데이터베이스 시스템에서 데이터를 이주하고 테이블이 GENERATED ALWAYS로 정의되어야 할 때 또는 ROLLFORWARD DATABASE 명령의 DROPPED TABLE RECOVERY 옵션을 사용하여 복구된 데이터로부터 테이블을 로드할 때 매우 유익할 수 있습니다. 이 수정자를 사용할 때는 식별 컬럼에 대한 데이터가 없거나 NULL 데이터가 있는 행들은 거부됩니다(SQL3116W). 이 수정자는 identitymissing 또는 identityignore 수정자와 함께 사용할 수 없습니다.</p> <p>주: 이 옵션을 사용하면 로드 유틸리티가 테이블의 식별 컬럼에 있는 값의 고유성을 유지보수하거나 검증하려고 시도하지 않습니다.</p>

표 8. 유효한 파일 유형 수정자(로드) (계속)

수정자	설명
indexfreespace= x	<p>x는 0부터 99까지의 정수입니다. 이 값은 색인 로드시 여유 공간으로 남게 될 각 색인 페이지의 비율로 해석됩니다. 페이지의 첫번째 항목은 제한 없이 추가되며, 그 다음 항목은 여유 공간 백분율 임계값이 유지보수될 수 있을 때 추가됩니다. 기본값은 CREATE INDEX 시점에서 사용된 값입니다.</p> <p>이 값은 CREATE INDEX문에 지정된 PCTFREE 값보다 우선적이며 색인 페이지에만 영향을 미칩니다.</p>
lobsinfile	<p><i>lob-path</i>는 LOB 값을 포함하는 파일에 대한 경로를 지정합니다. ASC, DEL 또는 IXF 로드 입력 파일에는 LOB 컬럼에 LOB 데이터를 갖는 파일 이름이 들어 있습니다.</p>
noheader	<p>헤더 검증 코드(단일 노드 노드 그룹에 상주하는 테이블에 대한 로드 조작에만 적용가능함)를 건너뛵니다.</p> <p>AutoLoader 유틸리티(183 페이지의 『제4장 AutoLoader』 참조) 다중 노드의 노드 그룹에 있는 테이블로 데이터를 제공하는 각 파일에 헤더를 작성합니다. 헤더에는 노드 번호, 파티션 맵, 파티션 키 스펙이 들어 있습니다. 로드 유틸리티는 데이터가 정확한 노드에 로드되고 있는지 검증하기 위해 이 정보를 필요로 합니다. 단일 노드 노드그룹에 있는 테이블에 파일을 로드할 때 헤더가 존재하지 않으므로, 이 옵션은 로드 유틸리티가 머릿글 검증 코드를 건너뛰게 합니다.</p>
norowwarnings	<p>거부된 행에 대한 모든 경고를 금지합니다.</p>
pagefreespace= x	<p>x는 0부터 100까지의 정수입니다. 이 값은 여유 공간으로 남게 될 각 데이터 페이지의 비율로 해석됩니다.</p> <p>최소 행 크기 때문에 지정된 값이 유효하지 않으면(예를 들어, 길이가 최소한 3 000 바이트이고, 값이 50인 x 값), 이 행은 새로운 페이지로 갑니다. 100 값이 지정되면, 각 행은 새로운 페이지에 들어갑니다.</p> <p>주: 테이블의 PCTFREE 값은 페이지당 지정된 여유 공간의 크기를 결정합니다. 로드 조작에 대해 pagefreespace 값 또는 테이블상에 PCTFREE 값이 설정되어 있지 않으면 유틸리티는 각 페이지에 대해 가능한 한 많은 공간을 차지합니다. pagefreespace에 의해 설정된 값은 테이블에 대해 지정된 PCTFREE 값을 겹쳐 씩니다.</p>
totalfreespace= x	<p>x는 0부터 100까지의 정수입니다. 이 값은 테이블 끝에 여유 공간으로 추가될 테이블에 있는 전체 페이지의 비율로 해석됩니다. 예를 들어, x가 20이고, 테이블에 100개의 데이터 페이지가 있으면, 20개의 빈 페이지가 추가로 첨부됩니다. 따라서, 테이블에 대한 전체 데이터 페이지 수는 120이 됩니다.</p>

파일 유형 수정자(로드)

표 8. 유효한 파일 유형 수정자(로드) (계속)

수정자	설명
usedefaults	<p>목표 테이블 컬럼에 대한 소스 컬럼이 지정되어 있지만, 하나 이상의 행 인스턴스에 대한 데이터가 들어 있지 않으면, 기본값이 로드됩니다. 다음은 데이터 생략의 예입니다.</p> <ul style="list-style-type: none"> • DEL 파일의 경우, 컬럼에 대해 ",,"가 지정됩니다. • DEL/ASC/WSF 파일의 경우, 컬럼을 충분히 가지고 있지 않거나 원래 스펙에 대해 길이가 충분하지 않은 행 <p>이 옵션을 사용하지 않으면, 소스 컬럼에 행 인스턴스에 대한 데이터가 없을 경우, 다음과 같은 경우가 발생합니다.</p> <ul style="list-style-type: none"> • 컬럼이 널(NULL) 입력 가능하면 널(NULL)이 로드됩니다. • 컬럼이 널(NULL) 입력 불가능하면 유틸리티는 행을 거절합니다.
ASCII 파일 형식(ASC/DEL)	
codepage=x	<p>x는 ASCII 문자열입니다. 이 값은 입력 데이터 세트에 있는 데이터의 코드 페이지로 해석됩니다. 로드 조작중 이 코드 페이지의 문자 데이터(및 문자에 지정된 숫자 데이터)를 데이터베이스 코드 페이지로 변환합니다.</p> <p>다음과 같은 규칙이 적용됩니다.</p> <ul style="list-style-type: none"> • 순수 DBCS(그래픽), 혼합 DBCS 및 EUC의 경우, 분리 문자는 x00부터 x3F까지의 범위로 제한됩니다. • EBCDIC 코드 페이지에 지정된 DEL 데이터의 경우, 분리 문자는 쉬프트인 및 쉬프트아웃 DBCS 문자와 일치하지 않을 수 있습니다. • nullindchar은 코드 포인트 x20과 x7F 사이에 설정된 표준 ASCII에 포함되어 있는 기호를 지정해야 합니다. 이것은 ASCII 기호와 코드 포인트를 참조합니다. EBCDIC 데이터는 코드 포인트가 다르더라도, 상응하는 기호를 사용할 수 있습니다.

표 8. 유효한 파일 유형 수정자(로드) (계속)

수정자	설명
dateformat="x"	<p>x는 소스 파일에서의 날짜 형식입니다. ^a 유효한 날짜 요소는 다음과 같습니다.</p> <p>YYYY - 년 (0000 - 9999 사이의 4자리수) M - 월 (1 - 12 사이의 한자리 또는 두자리수) MM - 월 (1 - 12 사이의 두자리수, M과는 상호 배타적임) D - 일 (1 - 31 사이의 한자리 또는 두자리수) DD - 일 (1 - 31 사이의 두자리수, D와는 상호 배타적임) DDD - 년도의 일(001 - 366 사이의 세자리수, 다른 일 또는 달 요소와는 상호 독점적임)</p> <p>지정되지 않은 각 요소에 대해 기본값 1이 할당됩니다. 다음은 날짜 형식의 일부 예입니다.</p> <p>"D-M-YYYY" "MM.DD.YYYY" "YYYYDDD"</p>

파일 유형 수정자(로드)

표 8. 유효한 파일 유형 수정자(로드) (계속)

수정자	설명
dumpfile = x	<p>x는 거부된 행이 기록될 예외 파일의 완전한 이름(서버 노드에 따라 규정화된)입니다. 최대 32KB의 데이터가 레코드별로 작성됩니다. 다음은 덤프 파일을 지정하는 방법을 나타내는 예입니다.</p> <pre>db2 load from data of del modified by dumpfile = /u/user/filename insert into table_name</pre> <p>주:</p> <ol style="list-style-type: none"> 1. 파티션된 데이터베이스 환경에서, 경로는 동시에 수행하는 로드 조작이 같은 파일에 쓰려고 시도하지 않도록 로딩 노드에 대해 지역적이어야 합니다. 2. 이 파일의 내용은 비동기 버퍼 노드에 있는 디스크에 작성됩니다. 실패하거나 인터럽트된 로드 조작의 이벤트에서, 디스크에 확인된 레코드의 수는 확실하게 알 수 없으며, LOAD RESTART 후 일관성을 보장할 수 없습니다. 이 파일은 한번 통과로 시작하고 완료하는 로드 조작에 대해서만 완료된다고 가정될 수 있습니다. 3. 이 수정자는 복수 파일 확장자를 갖는 파일 이름을 지원하지 않습니다. 예를 들면, <pre>dumpfile = /home/svtdbm6/DUMP.FILE</pre> 은 로드 유틸리티에서 사용 가능하지만, <pre>dumpfile = /home/svtdbm6/DUMP.LOAD.FILE</pre> 은 지원되지 않습니다.
implieddecimal	<p>내재된 소수점의 위치는 컬럼 정의에 의해 결정됩니다. 그리고 더 이상 값의 끝부분에 있다고 가정되지 않습니다. 예를 들어, 값 12345는 DECIMAL(8,2) 컬럼에 12345.00이 아니라 123.45로 로드됩니다.</p>

표 8. 유효한 파일 유형 수정자(로드) (계속)

수정자	설명
timeformat="x"	<p>x는 소스 파일에서의 시간 형식입니다. ^a 유효한 시간 요소는 다음과 같습니다.</p> <ul style="list-style-type: none"> H - 시 (12시간 시스템 경우에는 0 - 12 사이, 24시간 시스템 경우에는 0 - 24 사이의 한자리 또는 두자리수) HH - 시 (12시간 시스템 경우에는 0 - 12 사이, 24시간 시스템 경우에는 0 - 24 사이의 두자리수, H와는 상호 배타적임) M - 분 (0 - 59 사이의 한자리 또는 두자리수) MM - 분 (0 - 59 사이의 두자리수, M과는 상호 배타적임) S - 초 (0 - 59 사이의 한자리 또는 두자리수) SS - 초 (0 - 59 사이의 두자리수, S와는 상호 배타적임) SSSSS - 한밤중 이후의 초단위 낮 시간 (00000 - 86399 사이의 5자리수, 다른 시간 요소와는 상호 배타적임) TT - Meridian 표시기(AM 또는 PM) <p>지정되지 않은 각 요소에 대해 기본값 0이 할당됩니다. 시간 형식의 일부 예는 다음과 같습니다.</p> <p>"HH:MM:SS" "HH.MM TT" "SSSSS"</p>

파일 유형 수정자(로드)

표 8. 유효한 파일 유형 수정자(로드) (계속)

수정자	설명
timestampformat="x"	<p>x는 소스 파일에서의 시간소인 형식입니다. ^a 유효한 시간소인 요소는 다음과 같습니다.</p> <p>YYYY - 년 (0000 - 9999 사이의 네자리수) M - 월 (1 - 12 사이의 한자리 또는 두자리수) MM - 월 (1 - 12 사이의 두자리수, M(월)과는 상호 배타적임) D - 일 (1 - 31 사이의 한자리 또는 두자리수) DD - 일 (1 - 31 사이의 두자리수, D와는 상호 배타적임) DDD - 년도의 일(001 - 366 사이의 세자리수, 다른 일 또는 월 요소와는 상호 독점적임) H - 시 (12시간 시스템 경우에는 0 - 12 사이, 24시간 시스템 경우에는 0 - 24 사이의 한자리 또는 두자리수) HH - 시 (12시간 시스템 경우에는 0 - 12 사이, 24시간 시스템 경우에는 0 - 24 사이의 두자리수, H와는 상호 배타적임) M - 분 (0 - 59 사이의 한자리 또는 두자리수) MM - 분 (0 - 59 사이의 두자리수, M(분)과는 상호 배타적임) S - 초 (0 - 59 사이의 한자리 또는 두자리수) SS - 초 (0 - 59 사이의 두자리수, S와는 상호 배타적임) SSSSS - 한밤중 이후의 초단위 낮 시간 (00000 - 86399 사이의 5자리수, 다른 시간 요소와는 상호 배타적임) UUUUUU - 밀리초(000000 - 999999 사이의 6자리수) TT - Meridian 표시기(AM 또는 PM)</p> <p>지정되지 않은 YYYY, M, MM, D, DD 또는 DDD 요소에 대해 기본값 1이 할당됩니다. 지정되지 않은 다른 모든 요소에 대해 기본값 0이 할당됩니다. 다음은 시간소인 형식의 예입니다.</p> <p>"YYYY/MM/DD HH:MM:SS.UUUUUU"</p> <p>다음 예에서는 사용자 정의 날짜 및 시간 형식이 들어 있는 데이터를 schedule이라는 테이블로 가져오는 방법에 대해 설명합니다.</p> <pre>db2 import from delfile2 of del modified by timestampformat="yyyy.mm.dd hh:mm tt" insert into schedule</pre>

표 8. 유효한 파일 유형 수정자(로드) (계속)

수정자	설명
noeofchar	선택적인 파일 끝 문자 x'1A'는 파일의 끝으로 인식되지 않습니다. 마치 일반적인 문자인 것처럼 처리가 계속 진행됩니다.
ASC(컬럼 식별자가 없는 ASCII) 파일 형식	
binarynumerics	<p>숫자(DECIMAL은 아님) 데이터는 2진 형식이어야 하며, 문자 표현이 아니어야 합니다. 이것은 비용이 많이 드는 변환을 하지 않습니다.</p> <p>이 옵션은 reflen 옵션으로 지정된 고정 길이 레코드를 사용하여 위치적 ASC에서만 지원됩니다. noeofchar 옵션 사용이 가정됩니다.</p> <p>다음과 같은 규칙이 적용됩니다.</p> <ul style="list-style-type: none"> • BIGINT, INTEGER 및 SMALLINT의 예외시, 데이터 유형간의 변환이 수행되지 않습니다. • 데이터 길이는 해당 목표 컬럼 정의와 일치해야 합니다. • FLOAT는 IEEE 부동 소수점 형식이어야 합니다. • 로드 소스 파일의 2진 데이터는 로드 조작이 수행중인 플랫폼과 관계없이 big-endian으로 가정됩니다. <p>주: 이 수정자의 영향을 받는 컬럼에 대한 데이터는 널(NULL)이 존재할 수 없습니다. 공백(일반적으로, 널(NULL)로 해석됨)은 이 수정자가 사용될 때 2진 값으로 해석됩니다.</p>
nochecklengths	nochecklengths를 지정하면, 소스 데이터에 목표 테이블 컬럼의 크기를 초과하는 컬럼 정의가 들어 있더라도 각 행을 로드하기 위한 시도가 이루어집니다. 이러한 행은 코드 페이지 변환으로 소스 데이터가 축소될 경우, 올바르게 로드될 수 있습니다. 예를 들어, 소스의 4바이트 EUC 데이터는 목표의 2바이트 DBCS 데이터로 축소될 수 있으며, 필요한 공간을 반으로 줄입니다. 이 옵션은 소스 데이터가 일치되지 않는 컬럼 정의에도 불구하고 모든 경우에 적합할 때 특히 유용합니다.
nullindchar=x	<p>x는 단일 문자입니다. 널(NULL) 값을 표시하는 문자를 x로 변경합니다. x의 기본 값은 Y입니다.^b</p> <p>이 수정자는 문자가 영문자인 경우를 제외하고는 EBCDIC 데이터 파일에 대해 대소문자가 구별됩니다. 예를 들어, 널(NULL) 표시기 문자가 문자 N이 되게 지정되면, n도 널(NULL) 표시기로 인식됩니다.</p>

파일 유형 수정자(로드)

표 8. 유효한 파일 유형 수정자(로드) (계속)

수정자	설명
packeddecimal	<p>binarynumerics 수정자가 DECIMAL 필드 유형을 포함하지 않으므로 압축 10진 데이터를 직접 로드합니다.</p> <p>이 옵션은 reclen 옵션으로 지정된 고정 길이 레코드를 사용하여 위치적 ASC에서 만 지원됩니다. noeofchar 옵션 사용이 가정됩니다.</p> <p>부호 니블(1/2바이트)에 대해 지원되는 값은 다음과 같습니다.</p> <pre>+ = 0xC 0xA 0xE 0xF - = 0xD 0xB</pre> <p>주: 이 수정자의 영향을 받는 컬럼에 대한 데이터는 널(NULL)이 존재할 수 없습니다. 공백(일반적으로, 널(NULL)로 해석됨)은 이 수정자가 사용될 때 2진 값으로 해석됩니다.</p> <p>서버 플랫폼과 관계없이 로드 소스 데이터에 있는 2진 데이터의 바이트 순서는 big-endian으로 가정됩니다. 즉, OS/2나 Windows 운영 체제에서 이 수정자를 사용할 때 바이트 순서는 전환될 수 없습니다.</p>
reclen=x	<p>x는 32 767의 최대값을 갖는 정수입니다. 각 행에 대해 x 문자를 읽으며, 행의 끝을 나타내는 데 새로운 행 문자가 사용되지 않습니다.</p>
striptblanks	<p>가변 길이 필드에 데이터를 로드할 때 뒤 공백 공간을 모두 절단합니다. 이 옵션이 지정되지 않으면, 공백 공간이 보존됩니다.</p> <p>이 옵션은 striptnulls와 함께 지정할 수 없습니다. 이는 상호 배타적인 옵션입니다.</p> <p>주: 이 옵션은 백 레벨 호환성에 대해서만 지원되는 사용되지 않는 t 옵션을 바꿉니다.</p>
striptnulls	<p>가변 길이 필드에 데이터를 로드할 때 뒤 공백 널(NULL)(0x00 문자)을 모두 절단합니다. 이 옵션이 지정되지 않으면, 널(NULL)이 보존됩니다.</p> <p>이 옵션은 striptblanks와 함께 지정될 수 없습니다. 이는 상호 배타적인 옵션입니다.</p> <p>주: 이 옵션은 백 레벨 호환성에 대해서만 지원되는 사용되지 않는 padwithzero 옵션을 바꿉니다.</p>

표 8. 유효한 파일 유형 수정자(로드) (계속)

수정자	설명
zoneddecimal	<p>BINARYNUMERICs 수정자가 DECIMAL 필드 유형을 포함하지 않으므로 존 10진 데이터를 로드합니다. 이 옵션은 RECLEN 옵션으로 지정된 고정 길이 레코드를 사용하여 위치적 ASC에서만 지원됩니다. NOEOFCHAR 옵션 사용이 가정됩니다.</p> <p>반바이트 기호 값은 다음 중 하나일 수 있습니다.</p> <p>+ = 0xC 0xA 0xE 0xF - = 0xD 0xB</p> <p>지원되는 숫자값은 0x0에서 0x9 사이입니다.</p> <p>지원되는 존 값은 0x3 및 0xF입니다.</p>
DEL(컬럼 식별자가 있는 ASCII) 파일 형식	
chardelx	<p>x는 단일 문자열 분리 문자입니다. 기본값은 큰 따옴표(")입니다. 문자열을 묶기 위해 큰 따옴표 대신 지정된 문자가 사용됩니다.^{bc}</p> <p>작은 따옴표(')도 다음과 같이 한 문자열 분리 문자로 지정될 수 있습니다.</p> <p style="text-align: center;">modified by charde'l''</p>
coldelx	<p>x는 단일 문자 컬럼 분리 문자입니다. 기본값은 쉼표(,)입니다. 컬럼의 끝을 표시하기 위해 쉼표 대신 지정된 문자가 사용됩니다.^{bc}</p>
datesiso	<p>날짜 형식. 모든 날짜 데이터 값이 ISO 형식으로 로드되도록 합니다.</p>
decplusblank	<p>더하기 부호 문자. 양수(십진수) 값 앞에 더하기 부호(+) 대신 공백을 두게 합니다. 기본 조치는 양수(십진수) 값 앞에 더하기 부호를 붙이는 것입니다.</p>
decptx	<p>x는 마침표 대신 소수점 문자로 사용하는 단일 문자입니다. 기본값은 마침표(.)입니다. 소수점 문자로 마침표 대신 지정된 문자가 사용됩니다.^{bc}</p>

파일 유형 수정자(로드)

표 8. 유효한 파일 유형 수정자(로드) (계속)

수정자	설명
delprioritychar	<p>분리 문자에 대한 현재 기본 우선순위는 레코드 분리 문자, 문자 분리 문자, 컬럼 분리 문자순입니다. 이 수정자는 분리 문자 우선순위를 문자 분리 문자, 레코드 분리 문자, 컬럼 분리 문자순으로 전환하여 이전 우선순위를 사용하는 기존 응용프로그램을 보호합니다. 구문은 다음과 같습니다.</p> <pre>db2 load ... modified by delprioritychar ...</pre> <p>예를 들어, 다음과 같은 DEL 데이터 파일이 주어질 경우,</p> <pre>"Smith, Joshua",4000,34.98<row delimiter> "Vincent,<row delimiter>, is a manager", 4005,44.37<row delimiter></pre> <p>delprioritychar 수정자의 지정으로 이 데이터 파일에는 두 개의 행만 있게 됩니다. 두번째 <row delimiter>는 두번째 행의 첫번째 데이터 컬럼의 부분으로 해석되고, 첫번째와 세번째 <row delimiter>는 실제 레코드 분리 문자로서 해석됩니다. 이 수정자가 지정되지 않으면, 이 데이터 파일에는 각기 <row delimiter>로 구분된 세 개의 행이 있게 됩니다.</p>
dldelx	<p>x는 단일 문자 DATALINK 분리 문자입니다. 기본값은 세미콜론(;)입니다. DATALINK 값에 대한 필드간 분리자로서 세미콜론 대신 지정된 문자가 사용됩니다. 이는 DATALINK 값이 둘 이상의 부속값을 가질 수 있기 때문에 필요합니다.^{bcd} 주: x는 행, 컬럼 또는 문자열 분리 문자와 같은 문자로 지정되지 않아야 합니다.</p>
keepblanks	<p>이는 CHAR, VARCHAR, LONG VARCHAR 또는 CLOB 유형의 각 필드에 앞 공백 및 뒤 공백을 보존합니다. 이 옵션이 없으면, 문자 분리문자 내부에 있지 않는 모든 앞 공백 및 뒤 공백들은 제거되며, 모든 공백 필드에 대해 테이블에 NULL이 삽입됩니다.</p> <p>다음 예에서는 데이터 파일에 있는 모든 앞 공간 및 뒤 공간을 보존하면서 TABLE1이라는 테이블로 데이터를 로드하는 방법에 대해 설명합니다.</p> <pre>db2 load from delfile3 of del modified by keepblanks insert into table1</pre>
nodoubledel	<p>이중 문자 분리 문자의 인식을 억제합니다. 21 페이지의 『분리 문자 제한사항』에서 자세한 내용을 참조하십시오.</p>
IXF 파일 형식	

표 8. 유효한 파일 유형 수정자(로드) (계속)

수정자	설명
forcein	<p>유틸리티에게 코드 페이지가 일치하지 않더라도 데이터를 승인하고, 코드 페이지간의 변환을 억제하도록 지시합니다.</p> <p>데이터에 대한 길이가 충분한지 확인하기 위해 고정 길이 목표 필드를 점검합니다. nochecklengths가 지정되면, 점검이 수행되지 않고 각 행을 로드하기 위한 시도가 이루어집니다.</p>
nochecklengths	<p>nochecklengths를 지정하면, 소스 데이터에 목표 테이블 컬럼의 크기를 초과하는 컬럼 정의가 들어 있더라도 각 행을 로드하기 위한 시도가 이루어집니다. 이러한 행은 코드 페이지 변환으로 소스 데이터가 축소될 경우, 올바르게 로드될 수 있습니다. 예를 들어, 소스의 4바이트 EUC 데이터는 목표의 2바이트 DBCS 데이터로 축소될 수 있으며, 필요한 공간을 반으로 줄입니다. 이 옵션은 소스 데이터가 일치되지 않는 컬럼 정의에도 불구하고 모든 경우에 적합할 때 특히 유용합니다.</p>

파일 유형 수정자(로드)

표 8. 유효한 파일 유형 수정자(로드) (계속)

수정자	설명
<p>주:</p>	<p>1. MODIFIED BY 옵션을 사용하여 지원되지 않는 파일 유형을 사용할 경우, 로드 유틸리티는 경고를 발행하지 않습니다. 이를 시도하면, 로드 조작이 실패하며 오류 코드가 리턴됩니다.</p>
<p>2. ^a 날짜 형식 문자열 주위의 큰 따옴표는 필수사항입니다. 필드 분리자에는 a-z, A-Z 및 0-9가 포함될 수 없습니다.</p>	<p>필드 분리자는 DEL 파일 형식의 문자 분리문자 또는 필드 분리문자와 같을 수 없습니다. 요소의 시작 및 끝 위치가 명확한 경우에는 필드 분리자를 생략해도 됩니다. 가변 길이 항목들 때문에 D, H, M 또는 S와 같은 요소(수정자에 따라)가 사용되는 경우에는 애매한 부분이 있을 수 있습니다. 시간소인 형식의 경우에는 월 및 분 설명자 모두 영문자 M을 사용하기 때문에 혼돈을 피하려면 주의해서 사용해야 합니다. 월 필드는 다른 날짜 필드와 인접해 있어야 합니다. 분 필드는 다른 시간 필드와 인접해 있어야 합니다. 다음은 불분명한 시간소인 형식입니다.</p>
<p>"M" (월 또는 분이 될 수 있음)</p>	<p>"M:M" (어느 것인지 구분되지 않음)</p>
<p>"M:YYYY:M" (둘다 월로 풀이됨)</p>	<p>"S:M:YYYY" (시간값 및 날짜값 둘다에 인접함)</p>
<p>불분명한 경우에는 유틸리티가 오류 메시지를 보고하고 조작에 실패하게 됩니다.</p>	<p>다음은 명확한 시간소인 형식입니다.</p>
<p>"M:YYYY" (월)</p>	<p>"S:M" (분)</p>
<p>"M:YYYY:S:M" (월....분)</p>	<p>"M:H:YYYY:M:D" (분....월)</p>
<p>주: 큰따옴표 및 백슬래시와 같은 일부 문자 앞에는 Escape 문자(예: \)가 놓여야 합니다.</p>	<p>3. ^b 문자는 소스 데이터의 코드 페이지에 지정되어야 합니다.</p>
<p>문자 코드 포인트(문자 부호 대신)는 xJJ 또는 0xJJ(여기서, JJ는 코드 포인트의 16진 표시법임) 구문을 사용하여 지정될 수 있습니다. 예를 들어, # 문자를 컬럼 분리 문자로 지정하려면 다음 중 하나를 사용하십시오.</p>	<pre>... modified by coldel# modified by coldel0x23 modified by coldelX23 ...</pre>
<p>4. ^c 21 페이지의 『분리 문자 제한사항』에서는 분리 문자가 겹쳐쓸 때 사용될 수 있는 문자에 대해 적용되는 제한사항을 나열합니다.</p>	<p>5. ^d DATALINK 분리 문자가 URL 구문에서 유효한 문자이더라도, 로드 조작의 영역에서는 이 문자의 특수한 의미를 잃게 됩니다.</p>

예외 테이블

예외 테이블은 로드 중인 테이블의 정의를 반영하는 사용자 작성 테이블로서 몇 가지 추가 컬럼을 포함합니다. 이는 LOAD 명령의 FOR EXCEPTION 절에 의해 지정됩니다. 예외 테이블에 식별 컬럼 또는 다른 유형의 생성된 컬럼이 들어 있지 않을 수도 있습니다. 1차 테이블에 식별 컬럼이 있으면 예외 테이블 내의 해당 컬럼에 컬럼 유형, 길이 및 널(NULL) 입력 가능성 속성만 들어 있어야 합니다. 예외 테이블은 고유 색인 규칙을 위반한 행의 사본을 저장하는 데 사용됩니다. 이 유틸리티는 고유성 위반 이외의 다른 제한조건이나 외부 키 위반에 대해서는 점검하지 않습니다. DATALINK 예외사항도 예외 테이블에 캡처됩니다.

고유 키는 두 값이 동일하지 않은 키입니다. 이 제한조건을 실행하기 위해 사용되는 매커니즘을 고유 색인이라고 합니다. 기본 키는 고유 키의 특수한 경우입니다. 테이블은 둘 이상의 기본 키를 가질 수 없습니다.

주: 색인을 빌드하기 전에 유효하지 않은 데이터 때문에 거부된 모든 행은 예외 테이블에 삽입되지 않습니다.

행은 예외 테이블의 기존 정보에 추가되며, 이는 이전의 로드 조작에서 발생한 유효하지 않은 행도 포함할 수 있습니다. 현재 로드 조작에서 유효하지 않은 행만 원할 경우, 유틸리티를 호출하기 전에 기존 행을 삭제해야 합니다.

로드 유틸리티에서 사용하는 예외 테이블은 SET INTEGRITY 문에서 사용된 예외 테이블과 동일합니다.

고유 색인 및 중복 레코드 가능성이 있는 데이터를 로드할 때에는 예외 테이블이 사용되어야 합니다. 예외 테이블이 지정되어 있지 않은 경우에는 중복 레코드가 발견되면, 로드 조작은 계속 수행되고 삭제된 중복 레코드에 대한 경고 메시지만 발행됩니다. 레코드 자체는 로그되지 않습니다.

로드 조작이 완료되면, 오류 상태에 있는 데이터를 수정하기 위해 예외 테이블에 있는 정보를 사용할 수 있습니다. 그런 다음, 수정된 데이터를 테이블에 삽입할 수 있습니다.

SQL 참조서에서 예외 테이블에 관한 자세한 정보를 참조하십시오.

덤프 파일

dumpfile 수정자를 지정하면 로드 유틸리티에게 거부된 행이 기록될 예외 파일의 이름과 위치를 알려줍니다. 파티션된 데이터베이스 환경에서 수행시, 이 이름은 예외 파일이 생성된 파티션 번호를 나타내는 확장자가 지정됩니다. 예를 들면, 다음과 같습니다.

```
dumpfile = "/u/username/dumpit"
```

파티션 0에서, /u/username/dumpit.000이라고 명명된 파일을 생성합니다. 파티션 5에서는 /u/username/dumpit.005라고 명명된 파일이 생성되고, 계속 그 다음 파일이 명명됩니다.

길이가 32 768 바이트 보다 짧으면 레코드가 덤프 파일로 복사되고, 보다 길면 레코드 플래그먼트(레코드의 최종 바이트 포함)가 파일에 기록됩니다.

145 페이지의 『파일 유형 수정자(로드)』에서 로드 파일 유형 수정자에 관한 자세한 정보를 참조하십시오.

임시 파일 로드

DB2는 로드 처리중 임시 2진 파일을 작성합니다. 이 파일들은 로드 응급 복구, 로드 종료 조작, 경고 및 오류 메시지 및 런타임 제어 데이터에 사용됩니다. 이 임시 파일들은 로드 조작이 오류 없이 완료되면 삭제됩니다.

임시 파일은 LOAD 명령의 *temp-pathname* 매개변수 또는 **sqlload** API의 *pRemoteMsgFileName* 매개변수를 통해 지정될 수 있는 경로에 작성됩니다. 기본 경로는 데이터베이스 디렉토리의 서브디렉토리입니다.

임시 파일 경로는 서버 머신에 상주하며 DB2 인스턴스가 전용으로 액세스합니다. 따라서, *temp-pathname* 매개변수에 지정된 모든 경로 이름 규정은 클라이언트가 아닌 서버의 디렉토리 구조를 반영하며, DB2 인스턴스 소유자에게 경로에 대해 읽기 및 쓰기 사용권한이 있어야 합니다.

주: MPP 시스템에서는 임시 파일 경로가 지역 디스크에 상주하며 NFS 마운트에 상주하지 않습니다. 경로가 NFS 마운트에 있으면, 로드 조작시 성능이 현저하게 떨어집니다.

경고: 이 경로에 작성된 임시 파일은 어떠한 상황에서도 부당하게 변경되지 않아야 합니다. 그렇지 않으면, 로드 조작이 제대로 기능하지 못하게 되고 데이터베이스를 위험에 빠뜨리게 됩니다.

로드 유틸리티 로그 레코드

유틸리티 관리 프로그램은 로드 유틸리티를 포함하여 여러 DB2 유틸리티와 연관된 로그 레코드를 작성합니다. 로그 레코드는 특정 활동의 시작 또는 끝을 표시합니다. 다음 로그 레코드는 로드 조작과 연관된 레코드입니다.

- 로드 시작. 이 로그 레코드는 로드 조작의 시작과 연관됩니다.
- 테이블 로드 삭제 시작. 이 로그 레코드는 로드 조작에서 삭제 단계의 시작과 연관됩니다. 삭제 단계는 중복된 기본 키 값이 있는 경우에만 시작됩니다. 삭제 단계 동안에는 테이블 레코드 또는 색인 키에 대한 각 삭제 조작들이 잠깁니다.
- 로드 삭제 시작 보충. 이 로그 레코드는 로드 조작에서 삭제 단계의 끝과 연관됩니다.
- 로드 보류 목록. 이 로그 레코드는 로드 트랜잭션 확약시 작성됩니다. 보류 목록은 트랜잭션이 확약될 때까지 지연된 복구 불가능한 조작의 링크 목록입니다. 이 트랜잭션 다음에는 확약 로그 레코드가 생기지 않습니다.
- 테이블 공간 extent가 할당되거나 삭제될 때마다 로그 레코드가 기록됩니다(복구 불가능한 로드 조작 동안).

*Administrative API Reference*에서 이들 로그 레코드의 구조에 관한 설명을 참조하십시오.

문자 세트와 자국어 지원

DB2 UDB 데이터 이동 유틸리티는 다음과 같은 자국어 지원(NLS)을 제공합니다.

- 가져오기 및 내보내기 유틸리티는 클라이언트 코드 페이지로부터 서버 코드 페이지로 자동 코드 페이지 변환을 제공합니다.

문자 세트와 자국어 지원

- 로드 및 AutoLoader 유틸리티의 경우에, 데이터는 DEL 및 ASC 파일과 함께 codepage 수정자를 사용하여 임의의 코드 페이지로부터 서버 코드 페이지로 변환될 수 있습니다.
- 모든 유틸리티의 경우에, IXF 데이터는 원래의 코드 페이지(IXF 파일에 저장될 때)에서 서버 코드 페이지로 자동 변환됩니다.

문자 데이터의 확장 또는 축소와 관련하여 동일하지 않은 코드 페이지 상황이 일어날 수 있습니다. 예를 들어, 일본어 또는 대만어 확장 UNIX 코드(EUC)와 2바이트 문자 세트(DBCS)는 같은 문자에 대해서도 다른 길이로 코드화할 수 있습니다. 일반적으로, 목표 컬럼 길이에 대해 입력 데이터 길이를 비교하는 일은 데이터를 읽기 전에 수행됩니다. 입력 길이가 목표 길이보다 길 경우, 이 컬럼에 널(NULL) 입력이 가능하면 컬럼에 널(NULL)이 입력됩니다. 그렇지 않으면, 요청이 무시됩니다. nochecklengths 수정자(145 페이지의 『파일 유형 수정자(로드)』 참조)가 지정된 경우, 초기 비교가 수행되지 않고 데이터를 로드하기 위한 시도가 이루어집니다. 변환이 완료된 다음 데이터가 너무 길 경우, 이 행은 무시됩니다. 그렇지 않으면, 데이터가 로드됩니다.

로드 세션 예

CLP 예

예 1

TABLE1에는 5개의 컬럼이 있습니다.

- COL1 VARCHAR 20 NOT NULL WITH DEFAULT
- COL2 SMALLINT
- COL3 CHAR 4
- COL4 CHAR 2 NOT NULL WITH DEFAULT
- COL5 CHAR 2 NOT NULL

ASCFILE1에는 6개의 요소가 있습니다.

- ELE1 01-20
- ELE2 21-22

- ELE5 23-23
- ELE3 24-27
- ELE4 28-31
- ELE6 32-32
- ELE6 33-40

데이터 레코드는 다음과 같습니다.

```
1...5....10...15...20...25...30...35...40
Test data 1          XXN 123abcdN
Test data 2 and 3   QQY   wxyzN
Test data 4,5 and 6 WVN6789   Y
```

다음 명령은 파일로부터 테이블을 로드합니다.

```
db2 load from ascfile1 of asc modified by striptblanks reclen=40
method L (1 20, 21 22, 24 27, 28 31)
null indicators (0,0,23,32)
insert into table1 (col1, col5, col2, col3)
```

주:

1. MODIFIED BY 매개변수에 striptblanks 지정은 VARCHAR 컬럼에 있는 공백을 절단합니다(예를 들어, 행 1, 2, 3에 각각 11, 17, 19바이트가 들어 있는 COL1).
2. MODIFIED BY 매개변수에 reclen=40 지정은 각 입력 레코드 끝에 새로운 행 문자가 없으며, 각 레코드가 40바이트라는 것을 나타냅니다. 마지막 8바이트는 테이블을 로드하는 데 사용되지 않습니다.
3. 입력 파일에서 COL4가 제공되지 않으므로, 기본값이 TABLE1에 삽입됩니다 (NOT NULL WITH DEFAULT로 정의되어 있음).
4. 위치(position) 23과 32는 TABLE1의 COL2와 COL3이 지정된 행에 널(NULL)이 로드될 것인지를 나타내는 데 사용됩니다. 지정된 레코드에 대한 컬럼의 널(NULL) 표시기 위치에 Y가 있으면, 컬럼은 널(NULL)이 됩니다. N이 있으면, 입력 레코드의 컬럼 데이터 위치에 있는 데이터 값이(L(.....))에 정의된 것과 같이) 행에 대한 컬럼 데이터의 소스로서 사용됩니다. 이 예에서, 행 1에서는 어떠한 컬럼도 NULL이 아니고, 행 2에서는 COL2가 NULL이며, 행 3에서는 COL3이 NULL입니다.

로드 세션 예

5. 이 예에서는 COL1과 COL5에 대한 NULL INDICATORS가 0으로 지정되어 있어 데이터가 널(NULL) 입력이 불가능하다는 것을 나타냅니다.
6. 지정된 컬럼에 대한 NULL INDICATOR는 입력 레코드에서는 어느 곳이나 있을 수 있습니다. 단, 위치가 지정되어야 하며 Y 또는 N 값이 제공되어야 합니다.

예 2(파일로부터 LOB 로드)

TABLE1에는 세 개의 컬럼이 있습니다.

- COL1 CHAR 4 NOT NULL WITH DEFAULT
- LOB1 LOB
- LOB2 LOB

ASCFILE1에는 세 개의 요소가 있습니다.

- ELE1 위치 01-04
- ELE2 위치 06-13
- ELE3 위치 15-22

다음 파일은 /u/user1 또는 /u/user1/bin에 상주합니다.

- ASCFILE2에는 LOB 데이터가 있습니다.
- ASCFILE3에는 LOB 데이터가 있습니다.
- ASCFILE4에는 LOB 데이터가 있습니다.
- ASCFILE5에는 LOB 데이터가 있습니다.
- ASCFILE6에는 LOB 데이터가 있습니다.
- ASCFILE7에는 LOB 데이터가 있습니다.

ASCFILE1에 있는 데이터 레코드는 다음과 같습니다.

```
1...5....10...15...20...25...30.
REC1 ASCFILE2 ASCFILE3
REC2 ASCFILE4 ASCFILE5
REC3 ASCFILE6 ASCFILE7
```

다음 명령은 파일로부터 테이블을 로드합니다.

```
db2 load from ascfile1 of asc
  lobs from /u/user1, /u/user1/bin
  modified by lobsinfile reclen=22
  method L (1 4, 6 13, 15 22)
  insert into table1
```

주:

1. MODIFIED BY 매개변수의 lobsinfile의 스펙은 로더에게 모든 LOB 데이터가 파일로부터 로드될 것임을 전달합니다.
2. MODIFIED BY 매개변수의 reclen=22 스펙은 각 입력 레코드의 끝에 새로운 행 문자가 없으며, 각 레코드가 22바이트라는 것을 나타냅니다.
3. LOB 데이터는 ASCFILE2부터 ASCFILE7까지의 6개의 파일에 들어 있습니다. 각 파일에는 특정 행에 대해 LOB 컬럼을 로드하는 데 사용될 데이터가 들어 있습니다. LOB와 다른 데이터간의 관계는 ASCFILE1에 지정됩니다. 이 파일의 첫번째 레코드는 로더에게 행 1의 COL1에 REC1을 배치하도록 지시합니다. ASCFILE2의 내용은 행 1의 LOB1을 로드하는 데 사용되고, ASCFILE3의 내용은 행 1의 LOB2를 로드하는 데 사용됩니다. 이와 같이, ASCFILE4와 ASCFILE5는 행 2의 LOB1과 LOB2를 로드하는 데 사용되고, ASCFILE6과 ASCFILE7은 행 3의 LOB를 로드하는 데 사용됩니다.
4. LOBS FROM 매개변수에는 로더가 해당 파일을 요구할 때 명명된 LOB 파일을 검색하는 두 개의 경로가 있습니다.
5. ASCFILE1(컬럼 식별자가 없는 ASCII 파일)로부터 LOB를 로드하려면, lobsinfile 수정자를 사용하지 않고 다음과 같은 규칙을 지켜야 합니다.
 - LOB를 포함하여 모든 레코드의 전체 길이가 32KB를 초과할 수 없습니다.
 - 입력 레코드에 있는 LOB 필드는 고정 길이가 되어야 하며, 필요에 따라 공백으로 채워진 LOB 데이터를 가집니다.
 - LOB가 데이터베이스로 삽입될 때 LOB를 채운 공백이 제거될 수 있도록 striptblanks 수정자가 지정되어야 합니다.

예 3(덤프 파일 사용)

테이블 FRIENDS는 다음과 같이 정의됩니다.

```
table friends "( c1 INT NOT NULL, c2 INT, c3 CHAR(8) )"
```

이 테이블에 다음과 같은 데이터 레코드를 로드하려고 할 경우,

```
23, 24, bobby  
, 45, john  
4,, mary
```

두번째 행은 첫번째 INT가 널(NULL)이고 컬럼 정의가 NOT NULL을 지정하기 때문에 거부됩니다. DEL 형식과 일치하지 않는 초기 문자를 포함하는 컬럼은 오류를 일으키며 레코드는 거부됩니다. 이러한 레코드는 덤프 파일에 작성될 수 있습니다(145 페이지의 표8 참조).

문자 분리 문자 밖의 컬럼에 나타나는 DEL 데이터는 무시되지만, 경고를 생성합니다. 예를 들면, 다음과 같습니다.

```
22,34,"bob"
24,55,"sam" sdf
```

유틸리티는 테이블의 세번째 컬럼에 "sam"을 로드하고, 문자 "sdf"는 경고에 플래그됩니다. 레코드는 거부되지 않습니다. 다음은 또다른 예입니다.

```
22 3, 34,"bob"
```

유틸리티는 22,34,"bob"를 로드하고, 22 다음의 컬럼에 있는 일부 데이터가 무시되었다는 경고를 생성합니다. 레코드는 거부되지 않습니다.

예 4(DATALINK 데이터 로드)

다음 명령은 DEL 형식의 데이터가 들어 있는 입력 파일 delfile1로부터 MOVIE TABLE 테이블을 로드합니다.

```
db2 load from delfile1 of del
modified by d1del|
insert into movietable(Actorname, description, url_making_of, url_movie)
datalink specification (dl_url_default_prefix "http://narang"),
(dl_url_replace_prefix "http://bomdel" dl_url_suffix ".mpeg")
for exception excptab
```

주:

1. 이 테이블에는 네 개의 컬럼이 있습니다.

actorname	VARCHAR(n)
description	VARCHAR(m)
url_making_of	DATALINK (with LINKTYPE URL)
url_movie	DATALINK (with LINKTYPE URL)

2. 입력 파일의 DATALINK 데이터에는 부속 필드 분리 문자로 사용하는 세로 막대(|) 문자가 들어 있습니다.

3. url_making_of에 대한 컬럼 값이 접두부 문자 순서를 가지고 있지 않으면, "http://narang"이 사용됩니다.

4. url_movie에 대한 모든 널(NULL) 사용 불가능 컬럼 값은 "http://bomdel"을 접두부로 가져옵니다. 이미 있는 값이 바뀌게 됩니다.

5. url_movie에 대한 널(NULL) 사용 불가능 컬럼 값은 경로에 추가되는 ".mpeg"를 가져옵니다. 예를 들면, url_movie의 컬럼 값이 "http://server1/x/y/z"이면 "http://bomdel/x/y/z.mpeg"로 저장되고, 값이 "/x/y/z"이면 "http://bomdel/x/y/z.mpeg"로 저장됩니다.

6. 테이블을 로드하는 동안 고유 색인이나 DATALINK 예외가 발생하면, 영향을 받은 레코드는 테이블에서 삭제되어 예외 테이블 excptab을 넣습니다.

예 5 (식별 컬럼으로 테이블 로드하기)

TABLE1은 4개의 컬럼이 있습니다.

- C1 VARCHAR(30)
- C2 INT GENERATED BY DEFAULT AS IDENTITY
- C3 DECIMAL(7,2)
- C4 CHAR(1)

TABLE2는 TABLE1과 같습니다. 다만, C2는 GENERATED ALWAYS 식별 컬럼입니다.

DATAFILE1에서 데이터 레코드 (DEL 형식):

```
"Liszt"
"Hummel",,187.43, H
"Grieg",100, 66.34, G
"Satie",101, 818.23, I
```

DATAFILE2에서 데이터 레코드 (DEL 형식):

```
"Liszt", 74.49, A
"Hummel", 0.01, H
"Grieg", 66.34, G
"Satie", 818.23, I
```

주:

1. 다음과 같은 명령은 행 1과 2에 대해 식별 값을 생성합니다. 그 이유는 식별 값이 이들 행에 대해 DATAFILE1에서 제공되지 않았기 때문입니다. 하지만 상대적으로 행 3과 4는 사용자 제공 식별 값 100과 101을 할당합니다.

```
db2 load from datafile1.del of del replace into table1
```

2. DATAFILE1을 TABLE1로 로드해서 식별 값이 모든 행에 대해 생성되려면, 다음 명령을 입력하십시오.

```
db2 load from datafile1.del of del method P(1, 3, 4) replace into table1 (c1, c3, c4)
db2load from datafile1.del of del modified by identityignore replace into table1
```

3. DATAFILE2를 TABLE1로 로드해서 식별 값이 각각의 행에 대해 생성되려면, 다음 명령을 입력하십시오.

```
db2 load from datafile2.del of del replace into table1 (c1, c3, c4)
db2 load from datafile2.del of del modified by identitymissing replace into table1
```

4. DATAFILE1을 TABLE2로 로드해서 100과 101의 식별 값이 행 3과 4로 할당되게 하려면, 다음 명령을 입력하십시오.

```
db2 load from datafile1.del of del modified by identityoverride replace into table2
```

이 경우에, 행 1과 2는 거부됩니다. 그 이유는 유틸리티가 사용자 제공 값에서 시스템이 생성한 식별 값을 겹쳐쓰려고 했기 때문입니다. 만약 사용자 제공 값이 제세되지 않으면 식별 컬럼이 NULL이 아니기 때문에 행은 거부되어야 합니다.

- 5. 식별 관련 파일 유형 수정자를 사용하지 않고 DATAFILE1이 TABLE2로 로드되면, 수정자들이 자신 소유의 비널(NULL) 값을 제공하고 식별 컬럼이 GENERATED ALWAYS이기 때문에, 행 1과 2는 로드되지만 행 3과 4는 거부됩니다.

API 예

다음 샘플 프로그램에서는 다음을 수행하는 방법을 보여줍니다.

- SAMPLE 데이터베이스에 있는 EMP_RESUME 테이블의 정보를 EXPTABLE.DEL 파일로 내보내기
- 이 정보를 구분된 텍스트 파일로부터 새로운 테이블 LOADTABLE로 로드

관리 안내서에서 SAMPLE 데이터베이스에 관한 자세한 정보를 참조하십시오.

이 샘플 프로그램에 대한 소스 파일(tload.sqc)은 Windows 운영 체제 및 OS/2의 sqlllib\samples\c 디렉토리에서 또는 UNIX 기반 시스템의 sqlllib/samples/c 디렉토리에서 찾을 수 있습니다. 이는 DB2 API와 Embedded SQL 호출을 모두 포함합니다. 같은 디렉토리에 위치한 스크립트 파일 bldapp에는 이 샘플 프로그램 및 다른 샘플 프로그램을 빌드하기 위한 명령이 들어 있습니다. 응용프로그램 빌드 안내서에서 DB2 관리 API를 포함하는 응용프로그램 작성에 관한 일반적인 정보 및 컴파일과 링크 옵션에 관한 자세한 정보를 참조하십시오. tload.sqc 소스 파일로부터 샘플 프로그램 tload를 빌드하려면 다음을 수행하십시오.

1. tload.sqc, bldapp*, utilemb.c 및 utilemb.h 파일을 작업 디렉토리에 복사하십시오.
2. 데이터베이스 관리 프로그램이 수행중이지 않을 경우, db2start 명령을 발행하십시오.
3. bldapp tload sample을 입력하십시오. 다음 파일이 생성됩니다.

```
tload.bnd
tload.c
util.obj/util.o
tload.obj/tload.o
tload.exe/tload
```

샘플 프로그램(실행 가능한 파일)을 수행시키려면 tload를 입력하십시오. 메시지 파일과 같이 생성된 일부 파일 및 컬럼 식별자가 있는 ASCII 데이터 파일을 조사하는 데 유용하다는 것을 알게 됩니다.

```

/*****
**
** Source File Name = tload.sqc 1.4
**
** Licensed Materials - Property of IBM
**
** (C) COPYRIGHT International Business Machines Corp. 1995, 1997
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
**
**
** PURPOSE :
** To show the use of the QUIESCE TABLESPACE and the LOAD APIs.
** - EXPORT the EMP_RESUME table into a comma delimited file.
** - create a temporary table ('loadtable').
** - QUIESCE the TABLESPACES, preparing the temporary table to be
**   LOADable.
** - LOAD the comma delimited file into a temporary table ('loadtable').
**
** STRUCTURES USED :
** sqldcol
** sqlchar
** sqluexpt_out
** sqlca
**
** APIs USED :
** EXPORT sqluexpr
** QUIESCE TABLESPACE FOR TABLES sqluvqdp
** LOAD sqluload
**
** FUNCTIONS DECLARED :
** 'C' COMPILER LIBRARY :
** stdio.h - printf
** string.h - fgets, strncpy
**
** DBMS LIBRARY :
** sqlenv.h - see "APIs USED" above
**
** OTHER :
** external :
** check_error : Checks for SQLCODE error, and prints out any
** [in UTIL.C] related information available.
**
** EXTERNAL DEPENDANCIES :
** - Ensure existence of database (SAMPLE) for precompile purposes.
** - Precompile with the SQL precompiler (PREP in DB2)
** - Bind to a database (BIND in DB2)
** - Compile and link with the IBM Cset++ compiler(AIX and OS/2)
** or the Microsoft Visual C++ compiler (Windows)
** or the compiler supported on your platform.
**
**
**
**
*****/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sqlenv.h>
#include <sqlutil.h>
#include <malloc.h>
#include "utilemb.h"

#ifdef DB2AIX
#define WORKDIR "/u/workdir"
#else
#define WORKDIR "."
#endif

```

```

EXEC SQL INCLUDE SQLCA;
int main (int argc, char *argv[]) {
    short int          callerAction = 0;
    struct sql_dcol    DataDescriptor;
    struct sqlchar     *ActionString;
    struct sqlchar     *FileTypeMod;
    struct sqlu_xpt_out outputInfo;
    char               datafile[] = "EXPTABLE.DEL";
    char               statement[] = "SELECT empno, photo format, picture FROM emp_photo";
    char               impStatement[] = "INSERT INTO loadTable (num, format, photo)";

    /*
    char               statement[] = "SELECT empno, photo format FROM emp_photo";
    char               impStatement[] = "INSERT INTO loadTable (num, format)";
    */

    char               msgfile_x[] = "EXPMMSG.TXT";
    char               FileType[] = SQL_DEL;

    char               table_name[18];

    /* Variables for the LOAD API */
    struct sqlu_media_list DataFileList;
    struct sqlu_media_list *pLobPathList;
    struct sqlu_load_in InputInfo;
    struct sqlu_load_out OutputInfo;
    struct sqlu_media_list *pWorkDirectoryList = NULL;
    struct sqlu_media_list *pCopyTargetList;
    char LocalMsgFileName[] = "LOADMSG";
    char RemoteMsgFileName[] = "RLOADMSG";
    short CallerAction;
    sqlint32 *pNullIndicators;
    void *pReserved;

    EXEC SQL BEGIN DECLARE SECTION;
        char userid[9];
        char passwd[19];
    EXEC SQL END DECLARE SECTION;

    printf ("This is sample program 'tload.sqc'\n");

    /* Initialize structures. */
    memset(&DataFileList, 0, sizeof(struct sqlu_media_list));
    memset(&InputInfo, 0, sizeof(struct sqlu_load_in));
    memset(&OutputInfo, 0, sizeof(struct sqlu_load_out));

    /* need to preset the size of structure field and counts */
    outputInfo.sizeOfStruct = SQLU_XPT_OUT_SIZE;

    /******\
    * need to allocate the proper amount of space for the SQL statement *
    \******/
    ActionString = (struct sqlchar *)malloc(strlen(statement)
        + sizeof (struct sqlchar));
    ActionString->length = strlen(statement);
    strncpy(ActionString->data, statement, strlen(statement));

    FileTypeMod = (struct sqlchar *)malloc(strlen("lobsinfile")
        + sizeof (struct sqlchar));
    FileTypeMod->length = strlen("lobsinfile");
    strncpy (FileTypeMod->data, "lobsinfile", FileTypeMod->length);

    /* DELimited format can not have specified names, therefore the
    column method is 'D' default */
    DataDescriptor.dcolmeth = SQL_METH_D;

    if (Argc == 1) {
        EXEC SQL CONNECT TO sample;
        EMB_SQL_CHECK("CONNECT TO SAMPLE");
    }
    else if (Argc == 3) {
        strcpy (userid, argv[1]);
        strcpy (passwd, argv[2]);
        EXEC SQL CONNECT TO sample USER :userid USING :passwd;
        EMB_SQL_CHECK("CONNECT TO SAMPLE");
    }
}

```



```

else {
    printf ("\nUSAGE: tload [userid passwd]\n\n");
    return 1;
} /* endif */

printf ("Exporting EMP_RESUME table into file '%s'\n", datafile);
/*****
 * EXPORT API called *
*****/
sqluexpr (datafile, NULL, NULL, &DataDescriptor, ActionString,
    FileType, FileTypeMod, msgfile_x, 0, &outputInfo, NULL, &sqlca);
EMB_SQL_CHECK("exporting table");
printf ("Rows exported %d\n", outputInfo.rowsExported);

free (ActionString);

/* need to allocate the proper amount of space */
/* for the SQL statement */
ActionString = (struct sqlchar *)malloc(strlen(impStatement)
    + sizeof (struct sqlchar));
ActionString->length = strlen(impStatement);
strcpy(ActionString->data, impStatement, strlen(impStatement));

printf ("Creating a temporary table 'loadtable' to load into\n");
/* create a temporary table to import into */
EXEC SQL CREATE TABLE loadtable (num CHARACTER(6), format
    VARCHAR(10), photo BLOB(100K));
EMB_SQL_CHECK("CREATE TABLE");

/* end the transaction so the program can quiesce the tablespace */
EXEC SQL COMMIT;

printf ("Quiescing tablespaces for table 'loadtable'\n");
/*****
 * QUIESCE TABLESPACE FOR TABLE *
*****/
sqluvdp ("loadtable", SQLU QUIESCEMODE EXCLUSIVE, NULL, &sqlca);
EMB_SQL_CHECK("QUIESCE TABLESPACES FOR TABLE");

printf ("Loading the file '%s' into 'loadtable'\n", datafile);

/* initializing the variables for the LOAD API */
/* the DataFileList structure */
DataFileList.media_type = SQLU_SERVER_LOCATION;
DataFileList.sessions = 1;
DataFileList.target.location = (sqlu_location_entry *) malloc
    (sizeof(sqlu_location_entry) * DataFileList.sessions);
strcpy (DataFileList.target.location->location_entry, datafile);

pLobPathList = NULL;
CallerAction = SQLU_INITIAL;

/* the sqluload input structure */
InputInfo.sizeOfStruct = SQLULOAD_IN_SIZE; /* this should never change */
InputInfo.savecnt = 0; /* consistency points as */
/* infrequently as possible */
/* start at row 1 */
InputInfo.restartcnt = 0; /* load all rows */
InputInfo.rowcnt = 0; /* don't stop for warnings */
InputInfo.warningcnt = 0; /* default data buffer size */
InputInfo.data_buffer_size = 0; /* default warning buffer */
InputInfo.sort_buffer_size = 0; /* size */
InputInfo.hold_quiesce = 0; /* don't hold the quiesce */
InputInfo.restartphase = ' '; /* ignored anyway, but */
/* must be ' ',L,B,D */
InputInfo.statsopt = SQLU_STATS_NONE; /* don't bother collecting */
InputInfo.indexing_mode = SQLU_INX_AUTOSELECT; /* let load choose indexing */
/* mode

/* the sqluload output structure */
OutputInfo.sizeOfStruct = SQLULOAD_OUT_SIZE;

/* the CopyTargetList structure */
pCopyTargetList = NULL;

```

```

OutputInfo.sizeOfStruct = SQLLLOAD_OUT_SIZE;
/*****\
 * LOAD *
 \*****/
sqluload (&DataFileList,
          pLobPathList,
          &DataDescriptor,
          ActionString,
          FileType,
          FileTypeMod,
          LocalMsgFileName,
          RemoteMsgFileName,
          CallerAction,
          &InputInfo,
          &OutputInfo,
          pWorkDirectoryList,
          pCopyTargetList,
          pNullIndicators,
          pReserved,
          &sqlca);
EMB_SQL_CHECK("LOADing table");

printf ("Rows loaded %d\nrows committed %d\n", OutputInfo.rowsLoaded,
        OutputInfo.rowsCommitted);

free (ActionString);

/* drop the table */
EXEC SQL DROP TABLE loadtable;

EXEC SQL CONNECT RESET;
EMB_SQL_CHECK("CONNECT RESET");
}
/* end of program : tload.sqc */

```

loadqry.sqc 샘플 프로그램에 대한 소스 파일은 \sqllib\samples\c 디렉토리에서 찾을 수 있습니다. 이 샘플 프로그램에서는 프로그램이 연결되어 있는 데이터베이스에 대한 로드 조작의 현재 상태를 조회하기 위한 API 사용법을 보여줍니다. 이는 DB2 API와 Embedded SQL 호출을 모두 포함합니다. 같은 디렉토리에 위치한 스크립트 파일 bldvaemb.cmd에는 샘플 프로그램 및 다른 샘플 프로그램을 빌드하기 위한 명령이 들어 있습니다. 응용프로그램 빌드 안내서에서 DB2 관리 API를 포함하는 응용프로그램 작성에 관한 일반적인 정보 및 컴파일과 링크 옵션에 관한 자세한 정보를 참조하십시오. OS/2상의 loadqry.sqc 소스 파일로부터 샘플 프로그램 loadqry를 빌드하려면 다음을 수행하십시오.

1. loadqry.sqc, bldvaemb.cmd, util.c 및 util.h 파일을 작업 디렉토리에 복사하십시오.
2. 데이터베이스 관리 프로그램이 수행중이지 않을 경우, db2start 명령을 발행하십시오.
3. bldvaemb loadqry sample을 입력하십시오. 다음 파일이 생성됩니다.

```
loadqry.bnd
loadqry.c
util.obj
loadqry.obj
loadqry.exe
```

샘플 프로그램(실행 가능한 파일)을 수행시키려면 loadqry를 입력하십시오. 메시지 파일을 조사하는 데 유용하다는 것을 알게 됩니다. 이 파일은 프로그램이 로드 조작이 진행중일 때 수행될 경우에만 정보를 가집니다.

로드 조작후 보류 상태

일반적인 기록이 수행되지 않기 때문에, 로드 유틸리티는 보류 상태를 사용하여 데이터베이스 일관성을 보존합니다. 이러한 상태는 LIST TABLESPACES 명령의 사용으로 점검할 수 있습니다(*Command Reference* 참조).

로드 프로세스의 로드 및 빌드 단계는 모든 연관 테이블 공간을 로드 보류 상태에 넣습니다. 로드 보류 상태를 제거하려면 다음 중 하나를 수행하십시오(로드 조작에 실패했거나 인터럽트된 경우).

- 로드 조작을 재시작하십시오. 먼저, 장애의 원인을 정리해 보십시오. 예를 들어, 로드 유틸리티에 디스크 공간이 부족하면 로드 재시작 조작을 시도하기 전에 테이블 공간에 컨테이너를 추가하십시오.
- 로드 조작을 종료하십시오.
- 로드 조작이 실패한 동일 테이블에 대해 LOAD REPLACE 조작을 호출하십시오.
- 최근의 테이블 공간이나 데이터베이스 백업으로 RESTORE DATABASE 명령을 사용하여 로드하는 테이블에 대한 테이블 공간을 복구한 다음, 계속 필요한 복구 조치를 하십시오.
- 로드중인 테이블에 대한 테이블 공간을 삭제한 다음, 재작성하십시오.

삭제 단계는 연관된 모든 테이블 공간을 삭제 보류 상태에 넣습니다. 이는 로드 조작이 실패했거나 삭제 단계(고유 키 위반 제거) 동안 인터럽트될 때 발생할 수 있습니다. 삭제 기능은 로그되기 때문에 로그 공간이 부족할 때 장애가 발생할 수 있습니다. 로드 보류 상태를 제거하기 위해 취한 것과 동일한 조치가 삭제 보류 상태를 제거하는 데 사용될 수 있습니다.

로드 조작 후 보류 상태

테이블 공간은 로드 프로세스가 완료되고 다음과 같은 경우, 백업 보류 상태가 됩니다.

- 데이터베이스 구성 매개변수 *logretain*이 복구에 설정되거나 *userexit*가 작동 가능함
- 로드 옵션 COPY YES가 지정되지 않음
- 로드 옵션 NONRECOVERABLE이 지정되지 않음

로드 프로세스와 연관된 네번째 가능한 상태(점검 보류 상태)는 참조 및 점검 제한조건, DATALINKS 제한조건, AST 제한조건 또는 생성된 컬럼 제한조건에 관한 것입니다. 예를 들어, 기존의 테이블이 종속 테이블의 외부 키가 참조하는 기본 키를 갖는 상위 테이블일 경우, 상위 테이블에 있는 데이터를 대체하면 종속 테이블(테이블 공간이 아닌)이 점검 보류 상태에 놓이게 됩니다. 참조 무결성과 점검 제한조건에 대해 테이블의 유효성을 검증하려면, 테이블이 점검 보류 상태에 있을 경우, 로드 프로세스가 완료된 후 SET INTEGRITY문을 발행하십시오. 98 페이지의 『제한조건 위반에 대한 점검』에서 점검 보류 상태에 관한 자세한 정보를 참조하십시오.

로드 성능 최적화

로드 유틸리티의 성능은 데이터의 종류와 수량, 색인의 수 및 지정된 로드 옵션에 따라 달라집니다.

고유 색인은 중복된 색인이 있으면 로드 성능을 떨어뜨립니다. 대부분의 경우, 로드 조작이 관련된 후 각 색인에 대해 CREATE INDEX문을 호출하는 것보다 로드 조작중에 색인을 작성하는 것이 여전히 보다 효율적입니다(177 페이지의 그림 5 참조).



그림 5. 동시적인 색인화 및 통계 수집을 통해 로드 성능 향상. 일반적으로 테이블은 세 단계, 즉, 데이터 로드, 색인 빌드 및 통계 수집으로 빌드됩니다. 이 때문에 로드 조작, 색인 작성(각 테이블에 대해 여러 색인이 있을 수 있음) 및 통계 수집(테이블 데이터와 모든 색인에서 I/O를 일으키는) 중 다수의 데이터 I/O가 일어납니다. 훨씬 더 빠르게 수행하기 위한 다른 방법은 로드 유틸리티가 데이터를 한 번 통과하여 이들 작업을 모두 완료하는 것입니다.

색인 작성 성능을 조정할 때, 로드 조작중 색인 키를 정렬하는 데 제공된 메모리의 양은 *sorthheap* 데이터베이스 구성 매개변수에 의해 제어됩니다. 예를 들어, 로드 유틸리티에게 키를 정렬하기 위해 색인당 기본 메모리 4000 페이지를 사용하도록 지시하려면, *sorthheap* 데이터베이스 구성 매개변수를 4000 페이지로 설정하고, 데이터베이스로부터 모든 응용프로그램을 연결해제한 후, LOAD 명령을 발행하십시오. 색인이 너무 커서 메모리에서 정렬할 수 없으면 정렬 넘침이 일어납니다. 즉, 데이터가 여러 "정렬 수행"으로 나뉘지어 나중에 병합될 임시 테이블 공간에 저장됩니다. *sorthheap* 매개변수의 크기를 증가시켜 정렬 넘침을 피하는 방법이 없다면, 임시 테이블 공간에 대한 버퍼 풀이 넘침의 원인이 되는 디스크 입출력량을 최소화할 만큼 충분히 커야 한다는 점에 유의하십시오. 더구나, 정렬 수행의 병합 동안에 I/O 병렬화를 수행하려면, 각각 다른 디스크 장치에 상주하는 여러 개의 컨테이너로 임시 테이블 공간을 선언하는 것이 바람직합니다.

로드 조작중 색인을 작성하기 위해 씨드 파티 벤더의 고성능 정렬 라이브러리를 설치하여 로드 성능을 향상시킬 수 있습니다. 씨드 파티 정렬 제품의 예로는 SyncSort가 있습니다. **DB2SORT** 환경 변수(레지스트리 값)를 사용하여 런타임시 로드될 정렬 라이브러리의 위치를 지정하십시오. 환경 변수에 대해서는 *관리 안내서*에서 자세한 내용을 참조하십시오.

SET INTEGRITY문을 사용하면 테이블을 로드하는 데 필요한 전체 시간을 늘여나게 하고 다시 테이블을 사용 가능하게 만들 수 있습니다. 모든 로드 조작이 INSERT 모드에서 수행될 경우, SET INTEGRITY문은 테이블에 대해 제한조건 위반을 점차적으로 점검함

로드 성능 최적화

니다(테이블의 추가된 부분만 점검함). 테이블이 제한조건 위반에 대해 점차적으로 점검될 수 없다면 전체 테이블이 점검되며, 테이블이 다시 사용 가능해지려면 어느 정도 시간이 걸릴 수 있습니다.

로드 유틸리티는 INSERT 모드와 REPLACE 모드에서 똑같이 잘 수행합니다.

유틸리티는 사용자가 이들 매개변수를 지정하지 않을 경우, DISK_PARALLELISM, CPU_PARALLELISM 및 DATA_BUFFER에 대해 최적 값을 결정하여 가능한 최고 성능을 내려고 시도합니다. 최적화는 유틸리티 힙에서 사용 가능한 여유 공간과 크기에 따라 수행됩니다. 사용자의 특정 요구를 위해 매개변수를 조정하기 전에 로드 유틸리티가 이들 매개변수에 대한 값을 선택할 수 있도록 허용하는 것을 고려하십시오.

다음은 로드 유틸리티를 통해 사용 가능한 여러 가지 옵션의 성능 영향에 관한 내용입니다.

ANYORDER

로드중인 데이터의 순서 보존을 중지하고 성능을 향상시키려면 이 파일 유형 수정자를 지정하십시오. 로드될 데이터가 사전 정렬되어 있을 경우, anyorder가 이미 정렬된 순서를 훼손하므로, 다음 조회때 사전 정렬의 이점을 잃어버리게 됩니다.

BINARY NUMERICS 및 PACKED DECIMAL

숫자 ASC 데이터를 고정 길이 레코드로 로드할 때 성능을 향상시키려면 이 파일 유형 수정자를 사용하십시오.

COPY YES 또는 NO

로드 조작중 입력 데이터의 사본 작성 여부를 지정하려면 이 매개변수를 사용하십시오. 로드중인 모든 데이터가 로드 조작중에 복사되기 때문에(포워드 복구가 작동 가능해야 함) COPY YES는 로드 성능을 떨어뜨립니다. 증가된 I/O 활동이 I/O 바운드 시스템에서 로드 시간을 증가시킬 수 있습니다. 여러 장치 또는 디렉토리를 지정하면(다른 디스크상에) 이 조작의 결과로 일부 성능 패널티를 상쇄할 수 있습니다. 포워드 복구가 작동 가능할 경우 테이블이 백업 보류 상태가 되고, 데이터베이스나 선택된 테이블 공간이 테이블이 액세스되기 전에 백업되어야 하기 때문에, COPY NO는 전체적인 성능을 떨어뜨릴 수 있습니다.

CPU_PARALLELISM

이 매개변수를 사용하여 파티션 내 병렬 처리(사용자 머신의 성능 일부일 경우)를 이용하고 로드 성능을 높이십시오. 이 매개변수는 데이터 레코드를 분석하고

변환 및 형식화하기 위해 로드 유틸리티가 사용하는 프로세스나 스레드의 수를 지정합니다. 허용된 최대수는 30입니다. 지원된 값을 지원하기 위한 메모리가 충분하지 않으면 유틸리티가 값을 조정합니다. 이 매개변수가 지정되지 않을 경우, 로드 유틸리티는 시스템상의 CPU 수를 기준으로 한 기본값을 선택합니다.

이 매개변수의 값에 상관없이 소스 데이터의 레코드 순서가 보존됩니다(그림6 참조).

테이블에 LOB 또는 LONG VARCHAR 데이터가 있으면 CPU_PARALLELISM은 하나로 설정됩니다. 이 경우, 병렬 처리는 지원되지 않습니다.

이 매개변수의 사용이 SMP(대칭적 멀티프로세서) 하드웨어에 제한되지 않더라도, 비SMP 환경에서 이 매개변수를 사용할 때 분간되는 성능 이점을 얻지 못할 수 있습니다.



그림 6. 소스 데이터의 레코드 순서는 로드 조작중 파티션 내 병렬 처리 이용시 보존됩니다.

DATA BUFFER

DATA BUFFER 매개변수는 로드 유틸리티에 버퍼로 할당된 전체 메모리 양을 지정합니다. 이 버퍼를 여러 개의 *extent* 크기로 할 것을 권장합니다. 한 *extent*는 DB2 내의 데이터에 대한 이동 단위이며, *extent* 크기는 하나 이상의 4KB 페이지가 될 수 있습니다. DATA BUFFER 매개변수는 대형 오브젝트(LOB)에서 작업할 때 유용하며 I/O 대기 시간을 줄입니다. 데이터 버퍼는 유틸리티 힙으로부터 할당됩니다. 시스템에서 사용 가능한 저장영역의 크기에 따라 DB2 유틸리티가 사용할 메모리를 더 할당할 것을 고려해야 합니다. 데이터베이스 구성 매개변수 *util_heap_sz*는 적절히 수정될 수 있습니다. *Command Reference*에서 UPDATE DATABASE CONFIGURATION 명령에 관한 정보를 참조하십시오. 유틸리티 힙 크기 구성 매개변수에 대한 기본값은 5 000 4KB 페이지입니다. 로드 유틸리티가 유틸리티 힙의 메모리를 사용하는 여러 유틸리티 중 유일한 유틸리티이기 때문에, 이 매개변수에 의해 정의된 페이지의 50% 이하를 이 로드 유틸리티에 사용할 수 있게 하고, 유틸리티 힙을 충분히 크게 정의할 것을 권장합니다. *util_heap_sz*에 대해서는 관리 안내서에서 자세한 내용을 참조하십시오.

DISK_PARALLELISM

DISK_PARALLELISM 매개변수는 로드 유틸리티가 데이터 레코드를 디스크에 작성하기 위해 사용하는 프로세스나 스레드의 수를 지정합니다. 이 매개변수를 사용하여 데이터 로드시 사용 가능한 컨테이너를 활용하고 로드 성능을 향상시키십시오. 허용된 최대수는 CPU_PARALLELISM 값(로드 유틸리티에 의해 실제 사용된)의 4배 또는 50 중 더 큰 값입니다. 기본적으로, DISK_PARALLELISM 은 이 값이 허용된 최대수를 초과하는 경우를 제외하고는, 로드중인 테이블에 대한 오브젝트가 들어 있는 모든 테이블 공간의 테이블 공간 컨테이너의 합과 동일합니다.

FASTPARSE

fastparse 파일 유형 수정자를 사용하여 사용자 제공 컬럼 값에 대해 수행되는 데이터 점검을 줄이고 성능을 향상시키십시오. 이 옵션은 로드중인 데이터가 유효할 때에만 사용되어야 합니다. 성능을 약 10%나 20% 정도 향상시킬 수 있습니다.

NONRECOVERABLE

테이블에 대해 로드 트랜잭션을 복구할 필요가 없을 경우 이 매개변수를 사용하십시오. 테이블에 데이터 이동하는 것 외에 다른 추가적인 활동이 요구되지 않기 때문에 로드 성능이 향상되고, 테이블 공간을 백업 보류 상태에 두지 않고 로드 조작이 완료됩니다.

주: 차후 복원 및 롤 포워드 복구 도중 이러한 로드 트랜잭션이 발견되면, 이 테이블은 갱신되지 않고 "유효하지 않은" 테이블로 표시됩니다. 이 테이블에 대한 추가 조치는 무시됩니다. 롤 포워드 조작이 완료된 다음에는 테이블은 삭제만 가능합니다.

NOROWWARNINGS

경고의 수가 많아 생길 것이 우려되면, norowwarnings 파일 유형 수정자를 사용하여 거부된 행에 대한 경고 기록을 억제하고 성능을 향상시키십시오.

SAVECOUNT

이 매개변수를 사용하여 로드 조작시 일관성 지점의 설정을 위한 간격을 설정하십시오. 일관성 지점을 확립하기 위해 수행되는 활동 동기화는 시간이 걸립니다. 너무 자주 수행되면 로드 성능이 눈에 띄게 떨어집니다. 아주 많은 수의 행이 로드될 경우, 큰 SAVECOUNT 값을 지정하도록 권장합니다(예를 들어, 1억 개의 레코드와 관련하여 로드 조작할 경우 천만 값).

LOAD RESTART 조작은 자동으로 마지막 일관성 지점부터 계속 수행됩니다.

STATISTICS YES

이 매개변수를 사용하여 로드 조작 자체의 성능이 다소 떨어지더라도 (특히 DETAILED INDEXES ALL이 지정되어 있을 때), 로드 조작 완료 후 runstats 유틸리티를 호출하는 것보다 더 효율적으로 데이터 분배와 색인 통계를 수집하십시오.

최적 성능을 위해 응용프로그램은 가능한 한 최상의 데이터 분배와 색인 통계를 필요로 합니다. 통계가 갱신된 다음에는 응용프로그램은 최신 통계를 기준으로 하여 테이블 데이터에 새로운 액세스 경로를 사용할 수 있습니다. 테이블에 대한 새로운 액세스 경로는 DB2 BIND 명령을 사용하여 응용프로그램 패키지를 리바인드하여 작성될 수 있습니다(*Command Reference* 참조).

대형 테이블로 데이터를 로드할 때에는 *stat_heap_sz*(통계 힙 크기) 데이터베이스 구성 매개변수에 대해 더 큰 값이 지정되도록 권장합니다. *Command Reference* 에서 UPDATE DATABASE CONFIGURATION 명령에 관한 정보를 참조하십시오. *stat_heap_sz*에 대해서는 *관리 안내서*에서 자세한 내용을 참조하십시오.

WARNINGCOUNT

이 매개변수를 사용하여, 로드 조작이 강제로 종료되기 전에 유틸리티가 리턴할 수 있는 경고수를 지정하십시오. 단지 몇 개의 경고가 있거나 경고가 없다고 예상할 때에는 WARNINGCOUNT 매개변수를 대략 사용자가 예상하는 숫자로 설정하거나, 경고가 없을 것으로 예상할 때에는 20으로 설정하십시오. WARNINGCOUNT 숫자에 도달하면 로드 조작이 중지됩니다. 이것은 사용자에게 로드 조작을 완료하기 전에 데이터를 수정할(또는 로드중에 있는 테이블을 삭제한 후 재작성할) 기회를 줍니다. 로드 조작의 성능에 직접적인 영향을 주지는 않지만, WARNINGCOUNT 임계값을 설정하면 문제점이 있다고 판별하기 전에 전체 로드 조작이 완료될 때까지 기다리지 않아도 되게 합니다.

제한사항 및 한계점

다음 제한사항은 로드 유틸리티에 적용됩니다.

- 이 유틸리티는 별명의 사용을 지원하지 않습니다.
- 입력된 테이블 또는 구조화 유형 컬럼이 있는 테이블로 데이터를 로드하는 기능은 지원되지 않습니다.
- 선언된 임시 테이블로 데이터를 로드하는 기능은 지원되지 않습니다.

로드 제한사항 및 한계점

- 로드 보류 상태에 있는 테이블 공간에 테이블을 작성 또는 삭제하려는 시도가 실패하게 됩니다.
- DB2 Connect 또는 DB2 버전 2 이전의 다운레벨 서버를 통해 액세스된 데이터베이스로 데이터를 로드할 수 없습니다. 이 릴리스의 DB2에서만 사용 가능한 옵션은 이전 릴리스의 서버에서는 사용할 수 없습니다.
- LOAD REPLACE 조작중 오류가 발생하면, 테이블에 있는 원래 데이터가 유실됩니다. 로드 조작이 재시작될 수 있도록 입력 데이터의 사본을 보존하십시오.
- 새로 로드된 행에서는 트리거가 활성화되지 않습니다. 트리거와 연관된 비즈니스 규칙은 로드 유틸리티에 의해 시행되지 않습니다.

문제점 해결

데이터를 내보내기, 가져오기, 로딩, 바인딩 또는 복원과 같은 DB2 조작을 수행하는 동안 메시지 파일이 이러한 조작과 연관된 오류, 경고 및 정보용 메시지를 포함하여 작성되게 지정할 수 있습니다. MESSAGE 매개변수를 사용하여 이들 파일의 이름을 지정하십시오.

이들 메시지 파일은 표준 ASCII 텍스트 파일입니다. 이 메시지 파일을 인쇄하려면 운영 체제에 대한 인쇄 절차를 사용하고, 메시지를 보려면 ASCII 편집기를 사용하십시오.

주:

1. 조작이 완료된 다음에만 메시지 파일의 내용을 볼 수 있습니다.
2. 메시지 파일의 각 메시지는 새로운 행에서 시작하며 DB2 메시지 검색 기능이 제공하는 정보가 들어 있습니다.

제4장 AutoLoader

여기에서는 모든 파티션 또는 일부 파티션에서 동시에 데이터를 로드하기 위해 파티션된 데이터베이스 환경에서 사용할 수 있는 DB2 UDB AutoLoader 유틸리티에 대해 설명합니다.

다음과 같은 내용이 들어 있습니다.

- 『AutoLoader 개요』
- 185 페이지의 『AutoLoader를 사용하는 데 필요한 특권, 권한 및 권한 부여』
- 185 페이지의 『AutoLoader 사용』
- 186 페이지의 『다중 데이터베이스 파티션으로 로드』
- 187 페이지의 『AutoLoader 옵션』
- 195 페이지의 『AutoLoader 세션 예』
- 198 페이지의 『이주 및 백 레벨 호환성』
- 199 페이지의 『AutoLoader 힌트 및 추가 정보』
- 200 페이지의 『제한사항 및 한계점』
- 201 페이지의 『AutoLoader 문제점 해결』.

AutoLoader 개요

AutoLoader는 다음을 수행할 수 있는 유틸리티입니다.

- 하나의 시스템(예: MVS)에서 다른 시스템(예: UNIX)으로의 데이터 전송
- 데이터의 병렬 파티션
- 해당 데이터베이스 파티션에 데이터를 동시에 로드

AutoLoader는 다음 네 가지 모드 중 하나에서 수행할 수 있습니다.

- **SPLIT_AND_LOAD**. 데이터가 파티션되며(병렬 처리로) 해당 데이터베이스 파티션에 동시에 로드됩니다.

AutoLoader 개요

- **SPLIT_ONLY.** 데이터가 파티션되며(병렬 처리로) 출력이 지정된 위치에 있는 파일에 작성되거나, AutoLoader의 현재 작업 디렉토리에 있는 파일에 작성됩니다.
- **LOAD_ONLY.** 데이터가 이미 파티션되었다고 가정되며, 파티션 프로세스가 생략되고 데이터는 해당 데이터베이스 파티션에 동시 로드됩니다.
- **ANALYZE.** 모든 데이터베이스 파티션에 대한 분배와 함께 최적 파티션 맵이 생성됩니다.

파티션된 데이터베이스에서 많은 양의 데이터가 여러 파티션에 위치하고 있습니다. 데이터의 각 부분이 상주하는 데이터베이스 파티션을 결정하기 위해 파티션 키가 사용됩니다. 정확한 데이터베이스 파티션에 로드되기 위해서는 데이터가 그 전에 파티션되어야 합니다. AutoLoader 유틸리티는 두 가지 조작을 모두 수행할 수 있습니다(그림7 참조).

AutoLoader 유틸리티는 해싱 알고리즘을 사용하여 테이블이 정의되어 있는 노드 그룹에 데이터베이스 파티션이 있는 만큼 많은 출력 소켓으로 데이터를 파티션합니다. 그런 다음, 노드 그룹의 모든 데이터베이스 파티션 세트에서 동시에 이들 출력 소켓으로부터 로드합니다. 이 유틸리티의 키 기능은 파티션 및 로드 프로세스 동안 요구되는 모든 데이터 전송에 대해 소켓을 사용한 직접 TCP/IP 통신을 사용하는 것입니다. 또한 파티션 단계에 대해 복수 데이터베이스 파티션의 사용을 허용하며, 이를 통해 성능이 매우 향상됩니다.

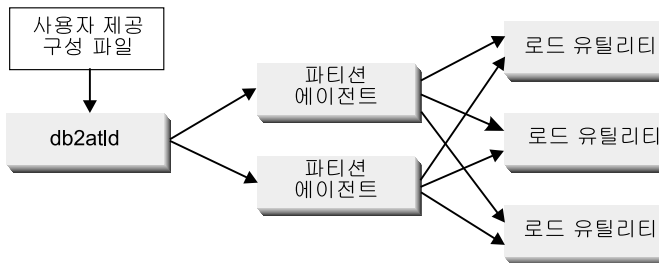


그림7. AutoLoader 개요 이 예에서는 AutoLoader가 소스 데이터를 읽은 후, 두 개의 파티션 에이전트의 각 에이전트로 절반씩 송신하고, 이 에이전트는 데이터를 파티션하여 세 개의 데이터베이스 파티션 중 하나에 송신합니다. 각 파티션의 로드 유틸리티가 데이터를 로드합니다.

AutoLoader를 사용하는 데 필요한 특권, 권한 및 권한 부여

특권은 사용자가 데이터베이스 자원을 작성하거나 액세스할 수 있게 합니다. 권한 레벨은 특권과 상위 레벨 데이터베이스 관리 프로그램 유지보수 및 유틸리티 조작용 그룹으로 분류하는 방법을 제공합니다. 이는 모두 데이터베이스 관리 프로그램과 데이터베이스 오브젝트에 대한 액세스를 제어하기 위한 것입니다. 사용자는 적합한 권한 즉, 필요한 특권이나 권한을 가지고 있는 오브젝트만 액세스할 수 있습니다.

AutoLoader 유틸리티를 사용하는 데 필요한 권한은 로드 유틸리티가 요구하는 권한과 동일합니다(91 페이지의 『로드를 사용하는 데 필요한 특권, 권한 및 권한 부여』 참조). 게다가, 로드 덤프 파일 및 로드 임시 파일은 인스턴스 소유자에 대해 쓰기 액세스 가능해야 합니다.

AutoLoader 사용

AutoLoader를 사용하기 전에

AutoLoader 유틸리티를 호출하기 전에 다음을 수행하십시오.

1. 임시 작업 디렉토리를 작성하십시오. 이 디렉토리는 모든 파티션 데이터베이스 파티션에 액세스할 수 있어야 합니다. 이 디렉토리에서 AutoLoader 유틸리티를 호출하게 됩니다.
2. AutoLoader 구성 파일을 변경하여(195 페이지의 『AutoLoader 세션 예』 참조) 작업 디렉토리로 복사하십시오.
3. *svcname* 데이터베이스 관리 프로그램 구성 매개변수와 **DB2COMM** 프로파일 레지스트리 변수가 올바르게 설정되어 있는지 확인하십시오. 이는 AutoLoader 유틸리티가 작업 파티션(유틸리티를 호출하는)에서 데이터베이스 파티션(테이블이 정의되어 있는)으로 원격 데이터베이스 연결을 작성하기 때문에 중요합니다.

AutoLoader 호출

AutoLoader 유틸리티는 **db2atld** 명령으로 호출됩니다.

```
db2atld [-config config_file] [-restart] [-terminate]
```

여기서 "-config config_file"은 AutoLoader 구성 파일을 지정합니다.

(기본값은 "autoloader.cfg"입니다.); "-restart"는 인터럽트된 AutoLoader 조작의 재시작을 요청하며(구성 파일은 재시작하도록 변경될 필요가 없음), "-terminate"는 인터럽트된 AutoLoader 조작의 종료를 요청합니다.

샘플 구성 파일인 autoloader.cfg는 sqllib/samples/autoloader 디렉토리에 들어 있습니다. 유틸리티를 수행시키려는 조작에 따라 샘플 구성 파일을 복사, 재명명 및 사용자 정의할 것을 권장합니다.

다중 데이터베이스 파티션으로 로드

복수 데이터베이스 파티션 노드 그룹의 테이블로 데이터를 로드할 경우, 로드 유틸리티는 로드될 파일이 파티션되어 정확한 헤더 정보를 갖도록 요구합니다. 로드 유틸리티는 데이터가 정확한 위치로 갔는지 확인하기 위해 AutoLoader 파티션 조작이 각 데이터 파일에 작성한 헤더 정보를 검증합니다.

단일 데이터베이스 파티션 노드 그룹의 테이블로 데이터를 로드할 경우, 테이블이 파티션 키를 갖도록 정의되어 있더라도 파일은 파티션될 필요가 없습니다. 이러한 상황에서는 로드 조작에 대해 noheader 수정자를 지정해야 합니다.

로드 유틸리티는 AutoLoader 파티션 조작이 사용한 파티션 맵이 테이블이 로드될 때 지정된 것과 동일한지를 점검합니다. 동일하지 않을 경우, 오류가 리턴됩니다. 로드 유틸리티는 파일 파티션이 올바른 데이터베이스 파티션에 로드되었는지 및 파티션시 지정된 파티션 키 컬럼의 데이터 유형이 키탈로그에 있는 현재 정의와 일치하는지도 점검합니다. 테이블이 로드된 노드 그룹은 데이터 파일이 파티션된 시간과 해당 데이터베이스 테이블로 부분이 로드된 시간 사이에는 재분배될 수 없습니다. 재분배가 수행된 경우, 유틸리티는 파티션된 데이터를 로드할 수 없습니다.

로드 유틸리티가 다음과 같은 플랫폼 파일 형식을 지원하더라도,

- 컬럼 식별자가 없는 ASCII(ASC)
- 컬럼 식별자가 있는 ASCII(DEL)
- PC/IXF

AutoLoader는 ASC와 DEL 파일을 파티션하는 데에만 사용할 수 있습니다.

PC/IXF 파일은 파티션될 수는 없지만, noheader 수정자를 로드 조작에 사용하

여 단일 데이터베이스 파티션 노드 그룹으로 로드될 수 있습니다. IXF 파일을 다중 파티션 테이블로 로드하려면, 먼저 이를 단일 파티션 테이블로 로드한 다음, 단일 파티션 테이블로부터 SELECT를 사용하여 다중 파티션 테이블로 데이터를 삽입해야 합니다.

LOAD ROWCOUNT절은 AutoLoader 조작에서 지원되지 않습니다. 이 매개변수는 파티션이 아닌 데이터베이스 환경에서만 유효합니다. LOAD SAVECOUNT절은 AutoLoader 조작에서 복수의 파티션이 사용될 경우에는 지원되지 않습니다.

파티션 키의 일부인 컬럼은 유효하지 않거나 거부되며, 해당 행과 연관된 데이터는 로드되지 않습니다. 행이 지정되더라도, 이 행은 덤프 파일에 들어 가지 않습니다. 대신, 레코드가 거부되었다는 것을 나타내는 메시지가 파티션 로그 파일에 기록됩니다. AutoLoader 프로세스가 완료된 후 반드시 파티션 로그 파일을 점검하십시오.

AutoLoader 옵션

다음과 같이 AutoLoader 구성 파일에 지정할 수 있는 여러 옵션이 있습니다.

RELEASE 레벨

이 구성 파일의 릴리스 레벨 구성 파일에 있는 이 행을 삭제하거나 변경하지 마십시오.

LOAD 명령

구성 파일에서 가장 중요한 부분은 LOAD 명령입니다. AutoLoader는 선택한 조작 모드가 로드의 필요성을 제안하지 않더라도, 데이터 처리 지시를 위해 이 LOAD 명령을 필요로 합니다. 예를 들면, AutoLoader는 SPLIT_ONLY 모드 조작을 수행할 경우에도 LOAD 명령을 사용하여 유용한 정보를 추출합니다. LOAD 명령의 스펙은 데이터가 오는 소스와 데이터 유형(예: 컬럼 식별자가 있는 ASCII), 데이터 로드 방법 및 목표 테이블 이름을 나타냅니다.

반드시 스키마 이름, 파일 이름, 파일 유형 및 테이블 이름을 포함하는 완전한 LOAD 명령을 지정하십시오. AutoLoader 유틸리티는 또한 LOAD 명령이 추가로 앞에 붙은 『db2』 키워드를 제외하고는 『db2 -f』 파일의 형식과 일치할 것을 요구합니다. *Command Reference*에서 이에 대한 자세한

한 정보와 기타 명령행 처리기(CLP) 옵션을 참조하십시오. LOAD 명령에 특수 escape 셸 문자를 사용할 필요는 없습니다. 마지막으로, 행의 마지막 문자가 백슬래시(\) 문자이면, 다음 행은 현재 행의 계속을 나타냅니다. 이러한 경우, 백슬래시와 행의 끝(eol) 문자는 무시됩니다.

LOAD 명령에 CLIENT 키워드를 지정한 경우에는 AutoLoader가 오류를 리턴합니다.

103 페이지의 『LOAD 명령』에서 LOAD 명령에 사용할 수 있는 모든 매개변수에 관한 자세한 정보를 참조하십시오.

DATABASE 매개변수

이 매개변수는 데이터가 로드될 데이터베이스를 식별하는 데 사용됩니다. 이름을 지정하지 않으면, SAMPLE을 기본값으로 사용합니다.

HOSTNAME 매개변수

이 매개변수는 데이터 파일이 상주할 원격 머신 이름을 지정합니다. 이 머신은 MVS 호스트나 다른 워크스테이션이 될 수 있습니다. 지정하지 않을 경우, FILE_TRANSFER_CMD 매개변수가 설정되고, 호스트 이름 nohost가 <hostname> 인수의 _TRANSFER_CMD 매개변수로 전달됩니다. 이 매개변수와 관련된 기본값은 없습니다.

FILE_TRANSFER_CMD 매개변수

호스트 파일 전송 개념을 지원하는 AutoLoader의 이전 버전. 여기서, AutoLoad 유틸리티가 원격 호스트에서 데이터 파일을 전송하도록 구성되었습니다. 이 옵션은 FILE_TRANSFER_CMD 옵션으로 바꿨습니다. 이 매개변수는 원격 호스트로부터 데이터를 전송하는 데 사용되는 실행가능한 파일, 일괄처리 파일 또는 스크립트의 완전한 이름을 지정합니다. 경로가 AutoLoader로 액세스할 수 있어야 합니다. 실행 파일 이름을 포함하는 전체 경로는 254자를 초과하지 않아야 합니다.

지정된 파일을 호출하기 전에, AutoLoader는 호스트로부터 송신되는 데이터를 예상하며 Named Pipe를 설정합니다. 작성될 Named Pipe의 수는 LOAD 명령의 FROM절에 나열된 파일 또는 장치 수와 동등합니다. LOAD 명령에서 오는 이 정보는 실행 가능한 파일, 일괄처리 파일 또는 스크립트로 전달될 매개변수를 지정하는 데 사용되기도 합니다.

이 정보를 근거로 AutoLoader가 다음 명령을 생성합니다.

```
<COMMAND> <logpath> <hostname> <basepipeName>
<nummedia> <source media list>
```

여기서,

- <COMMAND>는 호스트로부터 데이터를 이동시키는 데 사용되는 실행 가능한 파일, 일괄처리 파일 또는 스크립트에 대한 완전한 경로입니다.

그리고, 나머지 항목은 명령에 사용할 수 있는 매개변수입니다.

- <logpath>는 AutoLoader 로그 경로입니다. COMMAND 프로그램은 진단 또는 임시 데이터를 기록할 때 경로를 사용할 수 있습니다.
- <hostname>은 HOSTNAME 매개변수로 지정된 호스트 이름입니다.
- <basepipeName>은 AutoLoader가 작성할 Named Pipe에 대한 기본 이름입니다. AutoLoader 유틸리티는 기본 이름을 생성하고 이 이름이 시스템상에서 고유하다는 것을 보장합니다. 기본 이름은 필요한 Named Pipe를 작성하기 위해 유틸리티에 의해 추가됩니다.
- <nummedia>는 데이터를 제공하는 파일이나 장치(LOAD 명령의 FROM 절에 나열된)의 수입니다.
- <source media list>는 데이터를 제공하는 각 파일이나 장치(LOAD 명령의 FROM 절에 나열된)의 이름을 포함합니다. 이들 이름은 이름에 들어 있을 수 있는 특수 문자로 인해 생기는 잠재적인 문제점을 피하기 위해 큰 따옴표로 구분되어 있습니다.

AIX 샘플 파일인 atldftp.driv는 sqllib/samples/autoloader 디렉토리에 들어 있습니다. 샘플에서는 어떻게 FTP가 원격 호스트로부터 데이터를 이동시키는 데 사용될 수 있는지를 보여줍니다.

SPLIT_FILE_LOCATION 매개변수

이 매개변수는 다음과 같은 두 가지 방법으로 사용됩니다.

- 유틸리티가 LOAD_ONLY 모드일 경우, 파티션 파일의 위치에 대해 경로 이름을 제공하기 위해
- 파티션된 파일(유틸리티가 SPLIT_ONLY 모드일 경우)이 놓여질 위치에 대한 경로 이름을 제공하기 위해

이 매개변수에 대한 값이 지정되지 않고 유틸리티가 SPLIT_ONLY 모드에서 조작될 경우 파티션 파일은 현재 작업 디렉토리에 위치되며, 유틸리티가 LOAD_ONLY 모드에서 조작될 경우 유틸리티는 현재 작업 디렉토리에서 파티션 파일을 찾습니다.

SPLIT_FILE_LOCATION이 SPLIT_ONLY 조작 이후에 모든 파티션을 통해 마운트된 NFS 디렉토리를 가리키면, LOAD_ONLY 조작이 사용자 개입없이 분할 데이터 파일을 액세스할 수 있습니다. 그러나, SPLIT_FILE_LOCATION이 SPLIT_ONLY 조작 이후에 NFS 디렉토리를 가리키지 않으면, 사용자가 각 파티션으로부터 LOAD_ONLY 조작이 실행될 디렉토리로 분할 데이터 파일을 수동으로 이동해야 합니다.

OUTPUT_NODES 매개변수

로드 조작이 수행될 데이터베이스 파티션이 이 매개변수에 의해 식별됩니다. 지정된 파티션 번호는 테이블이 정의되어 있는 데이터베이스 파티션의 부속 집합이어야 합니다. 기본값은 '모두'입니다. 즉, 테이블이 정의되어 있는 모든 데이터베이스 파티션으로 데이터가 로드되게 합니다.

SPLIT_NODES 매개변수

파티션 프로세스에 참가한 데이터베이스 파티션은 이 매개변수를 통해 지정됩니다. 이들 데이터베이스 파티션은 로드중인 데이터베이스 파티션과 동일하거나 다를 수 있습니다. 이 매개변수에 대한 값이 지정되지 않으면, AutoLoader는 파티션에 필요한 파티션의 수를 결정하고 어느 파티션이 최적 성능을 달성하는 데 사용될 것인지를 결정합니다. 파티션에 필요한 파티션 수를 결정하기 위해 다음 규칙을 사용합니다.

- LOAD 명령에서 ANYORDER 수정자가 지정되지 않으면 하나의 파티션만 AutoLoader 세션에 사용되며,
 - OUTPUT_NODES 매개변수를 통해 한 파티션만 지정되거나 AutoLoader의 작업 파티션이 OUTPUT_NODES 매개변수에 대해 지정된 값의 요소가 아닐 경우, AutoLoader의 작업 파티션이 파티션으로 사용됩니다.
 - 그렇지 않으면, AutoLoader 작업 파티션 외의 첫번째 파티션 (OUTPUT_NODES에 있는)이 파티션으로 사용됩니다.
- LOAD 명령에서 anyorder 수정자가 지정되어 있을 경우,

1. 파티션 수가 다음과 같이 결정됩니다.

$$(\text{OUTPUT_NODES에 있는 파티션 수})/4 + 1$$

2. 이 파티션 수는 AutoLoader 작업 파티션을 제외하고 OUTPUT_NODES 매개변수에 대해 지정된 파티션으로부터 선택됩니다.

RUN_STAT_NODE 매개변수

LOAD 명령에서 STATISTICS YES 스펙과 결합하여 사용함으로써 통계를 수집하는 데이터베이스 파티션을 지정할 수 있습니다. 공백으로 두거나 -1 값을 지정하면, 기본값이 출력 파티션 목록의 첫번째 데이터베이스 파티션이 됩니다.

MODE 매개변수

이 매개변수는 AutoLoader 유틸리티가 수행하려는 모드를 지정합니다. 유효한 값은 SPLIT_AND_LOAD(기본값), SPLIT_ONLY, LOAD_ONLY 또는 ANALYZE입니다.

SPLIT_AND_LOAD

이 모드에서, 데이터가 파티션된 다음에 올바른 데이터베이스 파티션에 로드됩니다. 데이터는 소켓을 사용하여 직접 TCP/IP 통신을 통해 전송됩니다. 복수 입력 파일을 사용할 수 있습니다.

SPLIT_ONLY

이 모드에서 데이터는 파티션만 됩니다. 지정된 데이터베이스 파티션에 대해 파티션 데이터 파일 세트가 생성됩니다. 각 파티션 데이터 파일 세트를 위한 충분한 저장영역을 가지고 있어야 합니다. 파티션 기능은 SPLIT_FILE_LOCATION 매개변수가 지정하는 위치나 현재 작업 디렉토리에 파일을 작성합니다. 디렉토리 위치는 쓰기 액세스가 가능한 위치여야 합니다. 데이터는 filename.xxx 규칙을 사용하여 명명된 각기 다른 파일로 파티션되며, 여기서 xxx는 파티션 파일이 속하는 파티션의 번호를 나타냅니다. LOAD 명령에 다중 입력 데이터 파일이 있을 경우에는 파일이 모두 파티션됩니다. 단, 각 데이터베이스 파티션에 대해 한 개의 파티션 파일만 생성됩니다. 파티션 파일의 이름은 첫번째 입력 데이터 파일의 이름과 동일합니다.

LOAD_ONLY

이 모드에서는 이전 파티션 데이터가 로드됩니다. 데이터는 *filename.xxx* 또는 *filename.00xxx* 규칙을 사용하여 명명된 각기 다른 파일에 들어 있습니다. 여기서, *xxx*는 분리 파일이 속하는 파티션의 번호를 나타냅니다. AutoLoader는 *SPLIT_FILE_LOCATION* 또는 현재 작업 디렉토리에서 이들 파일을 찾을 것을 예상합니다. 디렉토리 위치는 읽기 액세스가 가능한 위치여야 합니다. 파티션 파일은 해당 파티션에서 동시에 로드됩니다. **LOAD** 명령에 다중 입력 데이터 파일이 있을 경우 (*infile1*, *infile2* 등과 같은), AutoLoader는 *infile1.xxx*가 있으면 파일을 로드합니다. 그렇지 않고 *infile1.00xxx*가 있다면 이를 로드합니다. 두 파일 모두 없을 경우, AutoLoader는 오류를 리턴합니다. 두 파일 모두 있을 경우, AutoLoader는 *infile1.xxx*를 로드합니다. 각 파일 유형(*xxx* 또는 *00xxx*) 중 첫 번째 *infile1*이 로드된 다음, *infile2*에 대한 점검이 시작되고, 이 프로세스가 모든 입력 파일이 로드될 때까지 반복됩니다.

ANALYZE

이 모드에서는 노드 그룹을 위한 사용자 정의 최적 파티션 맵이 생성됩니다. 많은 레코드 수를 갖는 데이터 파일을 입력 파일로 지정할 것을 권장합니다(복수 입력 파일이 허용됨). 이것이 수행되면, 맵은 노드 그룹의 각 데이터베이스 파티션에서 더욱 고르게 데이터 분배를 수행합니다. *MAP_FILE_OUTPUT* 매개변수에 의해 지정된 파일에 출력이 작성됩니다. **REDISTRIBUTE NODEGROUP** 명령(*Command Reference* 참조)은 새로운 파티션 맵이 수행되기 전에 호출되어야 합니다. **SPLIT_AND_LOAD** 모드에서의 계속적인 AutoLoader 호출은 자동으로 새로운 파티션 맵을 사용합니다. *MAP_FILE_INPUT* 매개변수는 노드 그룹의 기본 파티션 맵을 변경하지 않고 새로운 파티션 맵에 따라 데이터를 파티션할 때 사용할 수 있습니다.

LOGFILE 매개변수

이 매개변수는 AutoLoader 유틸리티가 사용하는 임시 및 영구 파일의 기본 이름을 제공하는 데 사용됩니다.

```

<logfile>.split.cfg ...
    모든 파티션에 대한 구성 파일
<logfile>.split.<3-digit-node-number>.log ...
    각 파티션에 대한 구성 파일
<logfile>.pmap.<pid> ...
    내부 임시 파일. <pid>는 AutoLoader 작업의 프로세스 ID임
<logfile>.load.<3-digit-node-number> ...
    LOAD 명령에 지정된 메시지 파일이 없을 경우,
    각 로드 프로세스에 대한 메시지 파일

```

LOGFILE 매개변수에 대한 경로를 지정할 수 있더라도, 해당 경로가 이미 있는지 및 액세스 가능성을 검증해야 합니다. 기본값은 ./autoloader.log입니다.

주: 여러 개의 동시적인 AutoLoader 세션의 경우, 지정된 기본 이름 또는 경로 이름이 고유한지 확인해야 합니다.

AUTHENTICATION 및 PASSWORD 매개변수

이들 매개변수는 로드시 클라이언트/서버 데이터베이스 연결 또는 파티션 프로그램의 원격 호출을 위해 암호가 요구될 경우에 필요합니다. AUTHENTICATION에 대한 기본값은 N0(암호 점검을 하지 않음)이며, PASSWORD 매개변수에 대해 지정된 모든 값이 무시됩니다.

지역 데이터베이스 연결 개념이 지정된 MPP 인스턴스의 모든 노드로부터의 연결을 포함하기 위해 MPP 환경으로 확장되었습니다. 이는 AUTHENTICATION 서버를 사용하여 인스턴스를 구성하더라도, db2nodes.cfg 파일에 정의되어 있는 노드 중 하나에서 연결을 시도할 경우 암호가 요구되지 않음을 의미합니다. AutoLoader는 AutoLoader 구성 파일의 AUTHENTICATION 플래그가 설정되어 있지 않거나 N0로 설정되고, PASSWORD 매개변수에 대한 값이 지정되어 있지 않으면 이 새로운 연결 조작을 사용합니다. AutoLoader에 대한 암호는 사용자 시스템에서 프로그램의 원격 실행을 위해 암호가 필요할 경우에만 필수적입니다. 예를 들어, UNIX 시스템의 .rhosts 파일이 rsh 실행을 작동 가능하게 하도록 적절히 설정되지 않을 경우 암호가 요구됩니다.

이와 달리, 암호가 필요할 경우 사용자가 작성한 암호 파일에 대한 완전한 경로를 정의하는 DB2 레지스트리 변수 DB2ATLD_PWFILE이 설정될 수 있습니다. 암호 파일과 완전한 경로는 모두 AutoLoader 유틸리티

에 액세스할 수 있어야 합니다. 이 변수가 정의되어 있으면, 값이 가리키는 파일의 첫번째 단어가 암호로 됩니다.

MAX_NUM_SPLITTERS 매개변수

이 매개변수는 AutoLoader 작업에서 사용할 수 있는 파티션 프로세스의 최대수를 지정합니다. 기본값은 25입니다.

FORCE 매개변수

이 매개변수는 유틸리티가 (시작 시간에) 일부 목표 파티션 또는 테이블 공간이 오프라인 상태임을 판별한 경우에도 AutoLoader 작업을 강제로 계속하도록 합니다. 값이 NO이고 일부 파티션이 사용 불가능하면, 데이터가 처리되지 않습니다. 값이 YES이면, 사용 가능한 데이터베이스 파티션이 로드되고 다른 모든 데이터베이스 파티션은 무시됩니다. 이 매개변수에 대한 기본값은 NO입니다.

STATUS_INTERVAL 매개변수

이 매개변수는 진행 메시지를 생성하기 전에 로드할 데이터의 메가바이트 (MB) 수를 지정합니다. 1-4000 범위의 모든 숫자가 유효한 값이 됩니다. 기본값은 100입니다.

PORTS 매개변수

이 매개변수는 내부 AutoLoader 통신을 위한 소켓을 작성하는 데 사용되는 TCP 포트의 범위를 지정합니다. 기본 범위는 6000-6063입니다. AutoLoader 호출 시간에 정의되면, **DB2ATLD_PORTS** DB2 레지스트리 변수 값이 이 매개변수에 대해 지정된 모든 값을 바꿉니다.

CHECK_LEVEL 매개변수

이 매개변수는 입력 또는 출력중 레코드 절단에 대해 점검을 수행해야 하는지의 여부를 지정합니다. 유효한 값은 CHECK와 NOCHECK입니다. 기본값은 NOCHECK입니다.

MAP_FILE_INPUT 매개변수

이 매개변수는 사용자 정의 파티션 맵을 포함하는 파일을 가리키는 입력 파일 이름을 지정합니다. 파티션 맵이 사용자 정의(기본값이 아닌) 맵이면, 이 매개변수를 지정해야 합니다. 최적 맵을 생성하기 위해 ANALYZE 모드에서 AutoLoader를 호출하여 사용자 정의 파티션 맵을 가져올 수 있습

니다. 이 맵은 실제 로드를 진행하기 전에 데이터베이스에 있는 각 데이터베이스 파티션으로 이동되어야 합니다.

MAP_FILE_OUTPUT 매개변수

이 매개변수는 AutoLoader가 ANALYZE 모드에서 호출될 때 파티션 맵에 대한 이름을 지정합니다. 최적 파티션 맵이 모든 데이터베이스 파티션에 데이터를 고르게 분배합니다. 이 매개변수에 대한 값을 지정하지 않고 유틸리티가 ANALYZE 모드에서 수행중이면, 오류가 리턴됩니다.

TRACE 매개변수

이 매개변수는 사용자가 모든 데이터 변환 프로세스의 덤프 파일 및 해쉬 값의 출력을 검토할 필요가 있을 때 추적할 레코드 수를 지정합니다. 기본값은 0입니다(추적하지 않음).

NEWLINE 매개변수

이 매개변수는 데이터 파일에 있는 각 레코드를 분리하는 데 사용하는 문자를 지정합니다. 이 매개변수는 입력 데이터 파일이 각 레코드가 새로운 행 문자로 분리되어 있는 고정 길이 ASC 파일이고, LOAD 명령에 reclen 수정자가 지정되어 있을 경우에만 의미가 있습니다. YES 값을 지정할 경우, AutoLoader는 레코드가 새로운 행 문자로 종료되는지 항상 점검합니다. 또한 레코드 길이가 reclen 수정자를 통해 지정된 길이와 일치하는지 점검합니다. 기본값은 NO입니다.

AutoLoader 세션 예

다음은 샘플 AutoLoader 구성 파일입니다(AIX).

```
#####
# release level
#####
RELEASE=V7.00

#####
# CLP load command
#####
db2 load from /home/user/atld_work/test.dat of del replace into user.test

#####
# database name
#####
database=wsdb

#####
# split partition list
```

AutoLoader 세션 예

```
#####
SPLIT_NODES=(0,2)

#####
# running mode
#####
mode=split_and_load

#####
# log file token
#####
logfile=mylog

#####
# frequency of progressive information
#
# print out progressive info every 10
# mega-bytes of data
#####
STATUS_INTERVAL=10
```

이 구성 파일에 대해 발행된 다음 명령에는 각 관련 데이터베이스 파티션에서 액세스할 수 있는 임시 작업 디렉토리와 경로가 들어 있습니다. 구성 파일의 이름, sample.atld.cfg도 다음과 같이 지정됩니다.

```
/home/user/atld_work/ $ db2atld -config sample.atld.cfg
```

이 명령을 호출하면 다음 출력이 생성됩니다.

```
/home/user/atld_work/ $ db2atld -config sample.atld.cfg
Utility program: "db2atld". Version: "07010".
Start reading autoloader configuration file: sample.atld.cfg.
Finish reading autoloader configuration file: sample.atld.cfg.
Start initializing autoloader process.
Finish initializing autoloader process.
The AutoLoader is now issuing all LOAD requests.
The LOAD operation has begun on partition "2".
The LOAD operation has begun on partition "0".
The LOAD operation has begun on partition "1".
The AutoLoader is now issuing all split requests.
Start db2split on node "0" in background.
Start db2split on node "2" in background.
The AutoLoader is waiting for all splitters to complete.
The utility has read "10" megabytes from the source data.
The utility has read "20" megabytes from the source data.
The utility has read "30" megabytes from the source data.
The utility has read "40" megabytes from the source data.
The utility has read "50" megabytes from the source data.
The utility has read "60" megabytes from the source data.
The utility has completed reading "62" megabytes from the user data.
The remote execution of the splitter utility on partition "0"
```


finished with remote execution code "0".
 The AutoLoader is waiting for all LOAD operations to complete.
 The remote execution of the splitter utility on partition "2"
 finished with remote execution code "0".

Operation	Node	SQL Code	Result
LOAD	002	+00000000	Success.
LOAD	000	+00000000	Success.
LOAD	001	+00000000	Success.
SPLIT	000	+00000000	Success.
SPLIT	002	+00000000	Success.
PRE-SPLIT	000	+00000000	Success.
RESULTS:	3 of 3 LOADs completed successfully.		

Summary of Splitters:

Rows Read 500003
 Rows Rejected 125214
 Rows Partitioned 374789

Summary of LOADs:

Rows Read 374789
 Rows Skipped 0
 Rows Loaded 374789
 Rows Rejected 0
 Rows Deleted 0
 Rows Committed 374789

AutoLoader가 생성한 메시지의 주요 부분은 관련 데이터베이스 파티션의 초기화에 관한 내용입니다. 분할 및 로딩 프로세스 양쪽 모두의 진행 상황이 개별적인 텍스트 파일에 기록됩니다. AutoLoader 프로세스의 종료도 기록됩니다.

수행된 조작의 요약 테이블, 사용된 파티션, 리턴된 SQL 코드 및 확보된 결과도 생성됩니다. 0 이외의 SQL 코드가 리턴되면, 메시지 파일 검토시 기록된 특정 경고나 오류가 나타납니다.

AutoLoader 작업에 대한 레코드 요약으로 출력이 완료됩니다.

이주 및 백 레벨 호환성

AutoLoader 유틸리티와 연관된 이주 및 백 레벨 호환성 문제가 있습니다.

- AutoLoader 유틸리티의 이전 버전은 **db2autold** 명령으로 호출되었습니다. 그러나, 현재 버전은 **db2atld** 명령을 통해 호출됩니다.
- **db2atld**는 내부 통신 채널로 소켓을 사용하고(Named Pipe와 반대로), 6063에서 6000까지의 기본 범위에서 TCP 포트 번호를 선택합니다. 단, 시스템이 다른 응용프로그램에 대해 이 범위를 필요로 할 경우, 이주시 다음 두 가지 옵션을 사용할 수 있습니다.
 - AutoLoader 구성 파일에 있는 PORTS 매개변수는 기본값 외의 다른 포트 범위를 지정하는 데 사용할 수 있습니다.
 - **DB2ATLD_PORTS** DB2 레지스트리 변수는 다음과 같이 범위를 지정하여 정의할 수 있습니다.

<lower-port-number>:<higher-port-number>

TCP 포트 범위를 결정하기 위한 우선순위의 순서는 **DB2ATLD_PORTS** DB2 레지스트리 변수, PORTS AutoLoader 구성 매개변수 및 기본값 순입니다.

- 클라이언트에서 서버로 데이터베이스 연결을 위해 암호가 필요할 경우, 이주할 때 사용할 수 있는 다음 두 가지 옵션이 있습니다.
 - AutoLoader 구성 매개변수 AUTHENTICATION과 PASSWORD를 사용할 수 있습니다. AUTHENTICATION이 YES로 설정되고 PASSWORD가 정의되어 있으면, 암호가 인증에 대해 사용됩니다. AUTHENTICATION이 YES로 설정되고 PASSWORD가 정의되어 있지 않으면, 암호에 대해 프롬프트됩니다.
 - DB2 레지스트리 값 **DB2ATLD_PWFILE**은 암호가 저장되어 있는 파일을 가리키도록 설정될 수 있습니다. 이를 지정하면, 파일 내용을 평가하여 최초 공백 분리 문자열이 암호로 사용됩니다. 레지스트리 값이 마지막으로 평가된 이후 정의되면, 다른 암호 값을 겹쳐쓰는 데 사용됩니다.

AutoLoader 힌트 및 추가 정보

다음은 AutoLoader 유틸리티를 사용하기 전에 고려해야 할 몇 가지 정보입니다.

- 소량의 데이터로 AutoLoader 유틸리티를 사용함으로써 AutoLoader 조작에 익숙해 지십시오.
- 입력 데이터가 이미 정렬되거나 선택한 순서로 되어 있으며, 로드 프로세스시 이 순서를 유지보수하려 할 경우, 파티션을 위해 데이터베이스 파티션만 사용해야 합니다. 병렬 처리 파티션은 데이터가 수신되었을 때와 같은 순서로 로드될 것이라는 것을 보장할 수 없습니다. anyorder 수정자가 LOAD 명령에 지정되지 않은 경우에는 AutoLoader가 기본값으로 단일 분할기(splitter)를 선택합니다.
- LOB가 각기 다른 파일에서 로드중일 경우(즉, 로드 유틸리티를 통해 lobinfile 수정자를 사용하고 있는 경우), LOB 파일이 들어 있는 모든 디렉토리는 로딩이 일어나고 있는 모든 데이터베이스 파티션에 읽기 액세스가 가능해야 합니다. LOAD *lob-path* 매개변수는 LOB로 작업할 때에는 완전히 규정되어야 합니다.
- 모든 임시 AutoLoader 파일은 LOGFILE AutoLoader 구성 매개변수를 사용하여 지정된 디렉토리에 상주합니다. 이 디렉토리는 파티션이 수행될 모든 파티션에 대한 읽기 및 쓰기 액세스와 함께 네트워크 액세스가 가능해야 합니다. 임시 파일에 대해 다른 디렉토리를 지정하여, 다른 테이블 공간에 있는 별도의 테이블에 데이터를 로드하기 위해 여러 개의 동시적인 AutoLoader를 수행할 수 있습니다.
- AutoLoader 작업에서 활동 데이터베이스 연결의 최대 수는 OUTPUT_NODES AutoLoader 구성 매개변수에 정의된 로딩 파티션의 수입입니다. *maxxappls*(활동 응용프로그램의 최대수) 데이터베이스 구성 매개변수가 충분히 설정되어 있는지 확인하십시오.
- AutoLoader 구성 파일에 FORCE=YES를 지정함으로써 AutoLoader가 (시동시) 일부 로딩 파티션이나 연관된 테이블 공간이 오프라인 상태인 것을 검출하더라도, AutoLoader 작업을 강제로 계속 수행하도록 할 수 있습니다.
- STATUS_INTERVAL AutoLoader 구성 매개변수를 사용하여 AutoLoader 작업의 진행을 모니터링하십시오. AutoLoader는 처리된 데이터의 메가바이트 수를 나타내는 메시지를 지정된 간격으로 리턴합니다.

AutoLoader 힌트 및 추가 정보

- 파티션(SPLIT_NODES 매개변수에 의해 정의됨)이 로딩 파티션(OUTPUT_NODES 매개변수에 의해 정의됨)과 다를 때 더 나은 성능을 기대할 수 있습니다. 이는 CPU 주기에 대한 경합이 더 적기 때문입니다. AutoLoader 유틸리티 자체는 파티션 또는 로드 조작에 참가하지 않은 데이터베이스 파티션에서 호출되어야 합니다. SMP 시스템에서 사용 가능한 각 CPU에 대해 최소한 한 개의 파티션 태스크가 있도록 하여 성능을 향상시킬 수 있습니다.
- AutoLoader는 LOAD 명령에 있는 MESSAGES 매개변수를 무시하고 LOAD 명령에 의한 모든 메시지를 load_log.XXX 파일로 보냅니다. 이 파일에는 데이터베이스 파티션 XXX상의 로드 프로세스로부터의 메시지가 들어 있습니다. AutoLoader는 spl_t_log.XXX 파일도 작성합니다. 이 파일에는 데이터베이스 파티션 XXX상의 파티션 프로세스로부터의 메시지가 들어 있습니다. 유틸리티는 주 AutoLoader 스크립트의 메시지가 들어 있는 autoload.log 파일도 작성합니다. 이 파일을 점검하여 모든 파이프 및 임시 디렉토리가 올바르게 설정되었는지 확인하십시오.
- AutoLoader는 통계를 수집할 출력 데이터베이스 파티션을 한 개만 선택합니다. 이 파티션을 지정하기 위해 RUN_STAT_NODE AutoLoader 구성 매개변수를 사용할 수 있습니다.
- AutoLoader의 다중 호출을 사용하여 각기 다른 테이블에 데이터를 동시에 로드할 수 있습니다. 다음을 확인하십시오.
 - 테이블이 별도의 테이블 공간에 상주함
 - 모든 AutoLoader 조작이 별도의 디렉토리에서 호출됨
 - 임시 파일을 작성하는 데 사용된 데이터 파일 이름이 각 AutoLoader 조작에 대해 고유함

제한사항 및 한계점

다음 제한사항은 AutoLoader 유틸리티에 적용됩니다.

- 로드 조작에 대한 입력 파일의 위치는 테이프 장치가 될 수 없습니다.
- AutoLoader는 LOAD 명령에서 CLIENT 옵션을 지원하지 않습니다.
- AutoLoader는 LOAD 명령에서 ROWCOUNT 옵션을 지원하지 않습니다.

- 데이터를 파티션하여 로드하기 위해 여러 개의 데이터베이스 파티션을 사용할 때, LOAD 명령에서 0보다 큰 SAVECOUNT의 사용은 지원되지 않습니다.
- AutoLoader를 사용하여 구조화 유형 컬럼이 있는 테이블로 데이터를 로드하는 기능은 지원되지 않습니다.

AutoLoader에도 적용되는 로드 제한사항에 대해서는 181 페이지의 『제한사항 및 한계점』에서 자세한 내용을 참조하십시오.

AutoLoader 문제점 해결

AutoLoader 유틸리티가 정지된 상태로 나타나면 다음을 수행할 수 있습니다.

- AutoLoader 구성 파일의 STATUS_INTERVAL 매개변수를 사용하여 AutoLoader 작업의 진행을 모니터하십시오.
- <logfile>.split.<3-digit-node-number>.log 파일을 점검하여 파티션중 인 각 데이터베이스 파티션에서 파티션의 프로세스 상태를 확인하십시오. 작업이 잘 진행되고 있고 AutoLoader 구성 파일의 TRACE 매개변수가 설정되어 있으면, 이들 로그 파일에 일정한 수의 레코드에 대한 추적 메시지가 있게 됩니다.
- LOAD 메시지 파일이나 <logfile>.load.<3-digit-node-number> 파일을 점검하여 로드 오류 메시지가 있는지 확인하십시오.
- AutoLoader 프로세스 중 하나가 오류를 만났다는 것을 알리는 오류를 발견하면 현재 AutoLoader 작업을 인터럽트하십시오.

AutoLoader 유틸리티가 계속 실패할 경우, 다음을 수행할 수 있습니다.

1. AutoLoader 구성 파일의 MODE 매개변수를 SPLIT_ONLY로 설정한 후, 유틸리티를 다시 호출하십시오.
2. 파티션 데이터 파일을 점검하여 파일 내에 비정상적인 내용이 있는지 확인하십시오. 파티션 파일이 정확하게 보이면, 올바른 데이터베이스 파티션에서 이들 파티션 파일 중 하나를 수작업으로 로드하십시오.
3. 데이터가 올바르게 로드하면, 추가 AutoLoader 문제점이나 데이터베이스 시스템 문제점이 있을 수 있습니다. IBM 서비스 부서에 문의하십시오.

AutoLoader 문제점 해결

AIX 버전 4.2에서 다음과 같은 라이브러리들(지정된 레벨 이상에 있는)이 없으면 **db2atld**는 정지될 수도 있습니다.

bos.rte.libc	4.2.1.13 (PTF U458582)
bos.rte.libpthreads	4.2.1.5 (PTF U458538)

다음 내용은 Windows NT용 IBM DB2 Universal Database Enterprise - Extended Edition에 대한 오류 시나리오에 적용됩니다.

여러 개의 네트워크 카드가 설치되어 있는 멀티 홈 머신에서 **db2atld**가 수행할 때, AutoLoader가 수행중인 머신에 **hostname** 명령을 입력한 다음, 같은 머신으로부터 이 호스트 이름을 ping하여 머신이 정확하게 구성되어 있는지 확인하십시오. 리턴된 IP 주소는 DB2 MPP 노드 목록에 있는 다른 머신으로부터 이 호스트 이름이 ping될 때 리턴된 주소와 동일해야 합니다. 머신이 올바르게 구성되어 있지 않으면, 유틸리티가 SQL6555N 오류를 리턴하고, AutoLoader 구성 파일의 OUTPUTNODES 매개변수에 의해 정의된 로딩 노드의 db2diag.log 파일에서 errno = 10061(연결이 거부됨) 오류 메시지가 있는 것을 보게 됩니다.

Windows NT 머신의 경우, 지역 호스트 이름에 대해 리턴된 IP 주소는 DNS나 hosts 파일에서 검색되지 않고 제어 패널 네트워크 아이콘에 지역적으로 구성된 정보로부터 검색됩니다. Windows NT 버전 4.0의 결함은 멀티 홈 머신에 리턴된 IP 주소 순서가 제어 패널 아이콘이 구성된 바인딩 순서를 무시하게 합니다. 이러한 문제점 해결에 도움을 주는 정보는 Microsoft Support 온라인 조항 Q171320을 참조하십시오.

제5장 DB2 Data Links Manager 데이터 이동

이 장에서는 DB2 Data Links Manager 데이터를 이동시키기 위해 DB2 내보내기, 가져오기 및 로드 유틸리티를 사용하는 방법에 대해 설명합니다.

239 페이지의 『부록C. 내보내기/가져오기/로드 유틸리티 파일 형식』에서 이 유틸리티에서 사용할 수 있는 파일 형식에 관한 정보를 참조하십시오.

DB2 Data Links Manager 빠른 시작에서 DB2 Data Links Manager에 관한 자세한 정보를 참조하십시오.

다음과 같은 내용이 들어 있습니다.

- 204 페이지의 『DB2 Data Links Manager 데이터를 이동시키기 위해 내보내기 유틸리티 사용』
- 208 페이지의 『DB2 Data Links Manager 데이터를 이동시키기 위해 가져오기 유틸리티 사용』
- 209 페이지의 『DB2 Data Links Manager 데이터를 이동시키기 위해 로드 유틸리티 사용』.

DB2 Data Links Manager 데이터를 이동시키기 위해 내보내기 유틸리티 사용

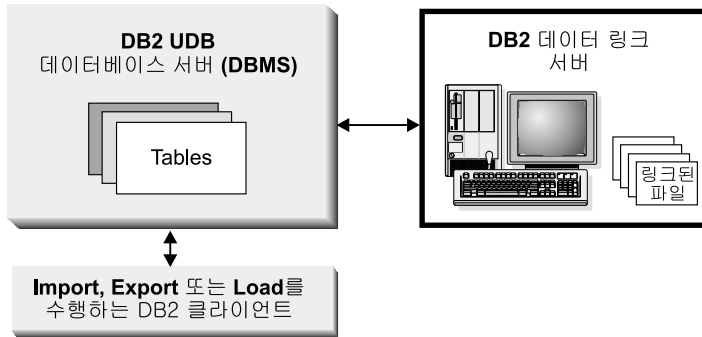


그림 8. DB2 Data Links Manager 데이터 이동. 테이블 데이터가 데이터베이스에 상주하고, DATALINK 컬럼이 참조하는 파일이 데이터 링크 서버에 상주하므로, 내보내기, 가져오기 및 로드 유틸리티는 데이터베이스 데이터와 해당 데이터 링크 서버의 데이터 파일을 모두 이동시켜야 합니다.

DB2 Data Links Manager 데이터를 이동시키기 위해 내보내기 유틸리티 사용

테이블 데이터가 데이터베이스에 상주하고, DATALINK 컬럼에 의해 참조된 파일이 데이터 링크 서버에 상주하므로, 내보내기 유틸리티는 데이터베이스 데이터와 해당 데이터 링크 서버의 데이터 파일을 모두 이동시켜야 합니다(그림 8 참조). 이를 수행하기 위해 내보내기 유틸리티가 데이터 링크 서버당 한 개의 제어 파일을 작성합니다. 제어 파일의 이름은 데이터 링크 서버의 이름과 동일합니다. 새로운 디렉토리에 작성된 제어 파일의 `d1fm/YYYYMMDD/HHMMSS`이며, 여기서 `YYYYMMDD`는 *YearMonthDay*를 나타내고 `HHMMSS`는 *HourMinuteSecond*를 나타냅니다. 이 디렉토리는 내보내기 데이터 파일이 작성된 디렉토리와 같은 디렉토리 아래에 작성됩니다. 제어 파일은 내보낸 행의 DATALINK 컬럼에 의해 참조된 상응하는 DB2 Data Links Manager 파일 이름을 나열합니다.

DFS(분산 파일 시스템) 또는 AIX 운영 체제에서 내보내기 유틸리티는 셀 내의 모든 데이터 링크 서버에 대해 한 개의 제어 파일만 작성합니다. 각 셀에 대해 하나의 제어 파일이 있게 됩니다. 제어 파일의 이름은 셀의 이름과 동일합니다. 제어 파일은 내보낸 행의 DATALINK 컬럼에 의해 참조된 셀 내에 있는 모든 DB2 Data Links Manager 파일의 URL을 나열합니다.

DB2 Data Links Manager 데이터를 이동시키기 위해 내보내기 유틸리티 사용

WINDOWS NT 운영 체제에서 내보내기 유틸리티는 모든 데이터 링크 서버에 대해 한 개의 제어 파일만 작성합니다. 이 제어 파일의 이름은 `ctrlfile.lst`입니다. 이는 `dlfm\YYYYMMDD\HHMMSS`라는 새로운 디렉토리에 작성됩니다. 이 디렉토리는 내보내기 데이터 파일이 작성된 디렉토리와 같은 디렉토리 아래에 작성됩니다. 제어 파일은 내보낸 행의 DATALINK 컬럼에 의해 참조된 모든 DB2 Data Links Manager 파일의 URL을 나열합니다.

NO LINK CONTROL 등록 정보를 갖는 DATALINK 값은 제어 파일에 배치되지 않습니다.

제어 파일은 각각의 해당 데이터 링크 서버로 전송되어야 합니다. DFS 경우에 각 셀에 대한 제어 파일은 셀 내의 데이터 링크 서버들 중 하나로 전송되어야 합니다. Windows NT 운영 체제에서, 단일 제어 파일은 참조된 모든 데이터 링크 서버로 전송되어야 합니다. **dlfm_export** 유틸리티는 제어 파일 이름을 지정하는 각 데이터 링크 서버에서 수행되어야 합니다. 이 유틸리티는 해당 데이터 링크 서버에 대한 제어 파일에 나열된 파일의 아카이브 파일을 작성합니다. DFS 경우에 **dlfm_export** 유틸리티는 제어 파일이 전송된 데이터 링크 서버에서 수행되어야 합니다. 이 유틸리티는 셀 내의 모든 데이터 링크 서버에 대한 제어 파일에 나열된 파일들의 아카이브를 작성합니다. **dlfm_export** 유틸리티를 수행하기 전에 **DLFM_FS_ENVIRONMENT** 레지스터리 변수를 알맞게 설정하십시오.

DATALINK 컬럼에 의해 참조된 해당 파일과 테이블의 일치 사본이 복사되었는지 확인하려면, 다음 단계를 수행하십시오.

1. 다음 명령을 발행하여 내보내기 조작이 수행하고 있을 때 진행중인 갱신 트랜잭션이 없다는 것을 확인하십시오.

```
db2 quiesce tablespaces for table tablename share
```

2. 내보내기 유틸리티를 호출하십시오.

3. 각 데이터 링크 서버에서 루트 권한을 갖는 **dlfm_export**를 수행하십시오. 이것은 데이터 링크 파일 관리자가 액세스 권한을 갖지 않을 수 있는 파일을 보존합니다. DFS 경우에 **dlfm_export** 유틸리티는 제어 파일에 나열된 파일들을 아카이브하기 전에 DCE 네트워크 루트 신임장을 가져 옵니다. 유틸리티는 아카이브된 파일들의 ACL 정보를 캡처하지 않습니다. **dlfm_export**에 입력할 때, 내보내기 유틸리티가 생성한 제어 파일 이름을 지정하십시오.

DB2 Data Links Manager 데이터를 이동시키기 위해 내보내기 유틸리티 사용

4. 다음 명령을 발행하여 테이블을 갱신 가능하게 하십시오.

```
db2 quiesce tablespaces for table tablename reset
```

내보내기 유틸리티가 SQL 응용프로그램으로 실행합니다. SELECT문의 조건에 맞는 행과 컬럼이 데이터베이스로부터 발췌됩니다. DATALINK 컬럼의 경우, SELECT문은 스칼라 함수를 지정하지 않아야 합니다.

내보내기 유틸리티는 다음과 같은 파일을 생성합니다.

- 내보내기 데이터 파일. 이 파일의 DATALINK 컬럼 값에는 가져오기 및 로드 유틸리티에서 사용된 것과 같은 형식이 있습니다. DATALINK 컬럼 값이 널(NULL)일 경우, 이는 다른 널(NULL) 컬럼과 같은 방법으로 처리됩니다.
- 각 데이터 링크 서버에 대한 제어 파일. 제어 파일은 데이터 링크 서버로부터 내보낼 모든 파일 이름과 완전한 경로를 나열합니다. DFS 경우에 각 셀에 대해 하나의 제어 파일이 있게 됩니다. Windows NT 운영 체제에는 DATALINK 컬럼 값이 참조하는 모든 데이터 링크 서버에 대해 한 개의 제어 파일만 있습니다.

dlfm_export 유틸리티를 사용하여 다음과 같이 하나 이상의 데이터 링크 서버로부터 파일을 내보내십시오.

```
dlfm_export control-file-name archive-file-name
```

여기서, *control-file-name*은 DB2 클라이언트에서 내보내기 유틸리티를 수행하여 생성된 제어 파일 이름이고, *archive-file-name*은 생성될 아카이브 파일 이름입니다. 기본 *archive-file-name*은 `export.tar`이며, 현재 작업 디렉토리에 위치합니다.

dlfm_import라는 보충 유틸리티는 **dlfm_export**가 생성한 아카이브 파일로부터 파일을 검색 및 복원하기 위해 제공됩니다. 이 유틸리티는 아카이브된 파일이 같은 데이터 링크 서버에 복원되든지 또는 다른 데이터 링크 서버에 복원되든지 관계없이 사용되어야 합니다.

dlfm_import 유틸리티를 사용하여 다음과 같이 아카이브 파일에서 파일을 검색 하십시오.

```
dlfm_import archive-file-name [LISTFILES]
```

DB2 Data Links Manager 데이터를 이동시키기 위해 내보내기 유틸리티 사용

여기서, *archive-file-name*은 파일을 복원하는 데 사용될 아카이브 파일의 이름입니다. 기본 *archive-file-name*은 `export.tar`입니다. LISTFILES은 선택적인 키워드로서 이를 지정하면 유틸리티가 아카이브 내에 포함된 파일 목록을 리턴시킵니다. 각 데이터 링크 서버에서 루트 권한이 있는 **dlfm_import** 유틸리티를 수행하십시오. 이는 사용자가 **dlfm_export** 유틸리티가 수행된 데이터 링크 서버와 동일한 디렉토리 구조와 사용자 ID를 갖지 않을 수 있는 다른 데이터 링크 서버에 아카이브 파일을 복원하고자 할 수 있기 때문입니다. DFS 경우에 목표 셀 내의 데이터 링크 서버 중 하나에서 **dlfm_import** 유틸리티를 수행하십시오. 이 유틸리티는 아카이브에서 파일을 발췌하기 전에 DCE 네트워크 루트 신임장을 가져 옵니다. **dlfm_import** 유틸리티를 수행하기 전에 **DLFM_FS_ENVIRONMENT** 레지스터리 변수를 알맞게 설정하십시오.

주: **dlfm_export** 유틸리티가 수행된 데이터 링크 서버와 다른 데이터 링크 서버에서 **dlfm_import** 유틸리티를 수행할 때, 파일이 올바른 경로에 복원됩니다. 이 파일은 일부 사용자 ID가 가져오기 중인 머신에 없을 경우, 루트가 소유하게 됩니다. 이들 파일을 데이터베이스에 삽입하기 전에 모든 파일이 올바른 사용권한을 가지고 있고 올바른 사용자 ID에 속하는지 확인하십시오. DFS 경우에 **dlfm_import** 유틸리티를 수행하기 전에 필요한 파일 세트를 작성하고 필요한 ACL들을 설정하십시오. 이 유틸리티는 파일을 발췌하는 데 필요한 디렉토리(존재하지 않을 경우)를 작성합니다.

다음 표에서는 SystemA라는 데이터베이스가 참조하는 파일과 DB2 데이터를 SystemB라는 데이터베이스로 내보내는 방법을 (비DFS 경우에 대해) 보여줍니다. SystemA는 데이터 링크 서버 DLFM1과 DLFM2를 사용합니다. SystemB는 데이터 링크 서버 DLFMX와 DLFMY를 사용합니다. DLFM1의 파일은 DLFMX로 내보내고, DLFM2의 파일은 DLFMY로 내보냅니다.

데이터 링크 서버 DLFM1과 DLFM2를 사용하는 데이터베이스 SystemA			단계
파일의 DB2 데이터	DLFM1에 대한 파일 이름이 있는 File1	DLFM2에 대한 파일 이름이 있는 File2	1) 두 데이터 링크 서버에서 <code>dlfm_export</code> 명령을(루트로서) 수행하십시오. 이 명령은 두 데이터 링크 서버에서 아카이브 파일을 생성합니다.
데이터 링크 서버 DLFMX와 DLFMY를 사용하는 데이터베이스 SystemB			

DB2 Data Links Manager 데이터를 이동시키기 위해 내보내기 유틸리티 사용

데이터 링크 서버 DLFM1 과 DLFM2 를 사용하는 데이터베이스 SystemA			단계
	DLFMX에서, 아카이브 파일로부터 복원하십시오.	DLFMY에서, 아카이브 파일로부터 복원하십시오.	2) 두 데이터 링크 서버에서 <code>dlfm_import</code> 명령을(루트로서) 수행하십시오.
			3) 내보낸 각 파일에 대해 적합한 데이터 링크 서버를 지정하려면, 매개변수 <code>DL_URL_REPLACE_PREFIX</code> 를 사용하여 SystemB에서 <code>IMPORT</code> 명령을 수행하십시오.
SystemB에서 <code>IMPORT</code> 명령을 수행하면, <code>DATALINK</code> 컬럼이 참조하는 모든 파일과 SystemA 데이터가 가져오기됩니다.			

DB2 Data Links Manager 데이터를 이동시키기 위해 가져오기 유틸리티 사용

테이블 데이터가 데이터베이스에 상주하고, `DATALINK` 컬럼이 참조하는 파일이 데이터 링크 서버에 상주하므로, 가져오기 유틸리티는 데이터베이스 데이터와 해당 데이터 링크 서버의 데이터 파일을 모두 이동시켜야 합니다(204 페이지의 그림8 참조).

목표 데이터베이스에 대해 가져오기 유틸리티를 수행하기 전에 다음을 수행하십시오.

1. 적합한 데이터 링크 서버에 참조될 파일을 복사하십시오. `dlfm_import` 유틸리티는 `dlfm_export` 유틸리티가 생성한 보존 파일로부터 파일을 발췌하는 데 사용될 수 있습니다.
2. 데이터 링크 서버의 데이터 링크 파일 관리자에 대한 하나 이상의 접두부 이름을 정의하십시오(데이터베이스 등록과 같이 기타 관리적인 작업을 수행하고자 할 수 있습니다).
3. 필요하다면, `SQL` 테이블에 대해 내보낸 데이터로부터 `URL`(`DATALINK` 컬럼의)에 있는 데이터 링크 서버 정보를 갱신하십시오(원래 구성의 데이터 링크 서버가 대상 위치에 있는 데이터 링크 서버와 같으면, 데이터 링크 서버 이름

DB2 Data Links Manager 데이터를 이동시키기 위해 가져오기 유틸리티 사용

이 갱신될 필요가 없습니다). DFS 경우 필요하다면, SQL 테이블에 대해 내보낸 데이터로부터 URL(DATALINK 컬럼의)에 있는 셀 이름 정보를 갱신하십시오.

4. DB2 Data Links Manager 구성 파일에 있는 목표 구성에 데이터 링크 서버를 정의하십시오. DFS 경우에 DB2 Data Links Manager 구성 파일에 있는 목표 구성에 셀을 정의하십시오. **dlfm server_conf** 명령과 **dlfm client_conf** 명령을 사용하여 목표 셀 내에 DB2 Data Links Manager를 등록하십시오.

가져오기 유틸리티가 목표 데이터베이스에 대해 수행할 때, DATALINK 컬럼 데이터에 의해 참조된 파일은 적합한 데이터 링크 서버에 링크됩니다.

DB2 Data Links Manager 데이터를 이동시키기 위해 로드 유틸리티 사용

FILE LINK CONTROL로 정의된 DATALINK 컬럼이 있는 테이블로 데이터를 로드할 경우, 로드 유틸리티를 호출하기 전에 다음 단계를 수행하십시오(모든 DATALINK 컬럼이 NO LINK CONTROL로 정의되면, 이 단계는 필요하지 않습니다).

1. DB2 Data Links Manager가 DATALINK 컬럼 값에 의해 참조될 데이터 링크 서버에 설치되어 있는지 확인하십시오. DFS 경우에 목표 셀 내의 DB2 Data Links Manager가 등록된 상태인지 확인하십시오.
2. 데이터베이스가 DB2 Data Links Manager와 함께 등록되었는지 확인하십시오.
3. DATALINK 값으로 삽입될 모든 파일을 적합한 데이터 링크 서버에 복사하십시오.
4. 데이터 링크 서버의 DB2 Data Links Manager에 대해 접두부 이름(들)을 정의하십시오.
5. DATALINK 데이터(로드될)에 의해 참조된 데이터 링크 서버를 DB2 Data Links Manager 구성 파일에 등록하십시오. DFS 경우에, DATALINK 데이터(로드될)에 의해 참조된 목표 구성에 있는 셀들을 DB2 Data Links Manager 구성 파일에 등록하십시오.

DB2 Data Links Manager 데이터를 이동시키기 위해 로드 유틸리티 사용

DB2와 데이터 링크 서버간 연결은 로드 유틸리티 수행중 실패할 수 있으며, 로드 조작이 실패하도록 합니다. 이러한 상황이 발생하면,

1. 데이터 링크와 DB2 Data Links Manager를 시작하십시오.
2. LOAD RESTART 명령을 발행하십시오(103 페이지의 『LOAD 명령』 참조).

로드 조작중 실패한 링크는 데이터 통합성이 위반되는 것으로 고려되며, 고유 색인 위반과 같은 방법으로 처리됩니다. 따라서, 하나 이상의 DATALINK 컬럼이 있는 테이블 로딩에 대해 특별한 예외가 정의되어 있습니다. 추가 정보는 *SQL* 참조서에 있는 예외 설명을 참조하십시오.

다음은 DATALINK 컬럼이 있는 테이블들을 로드할 때 지원되지 않는 로드 유틸리티 기능입니다.

- CPU_PARALLELISM (값이 1로 강제 설정됨)
- LOAD REPLACE
- LOAD TERMINATE
- LOAD COPY

제6장 시스템간 데이터 이동

여기에서는 플랫폼간 및 DRDA 호스트 데이터베이스간에 데이터를 전송하기 위해 DB2 내보내기, 가져오기 및 로드 유틸리티를 사용하는 방법에 대해 설명합니다. 엔터프라이즈의 데이터베이스간에 데이터를 이동시키기 위한 또다른 방법인 DB2 DataPropagator에 대해서도 설명합니다. DWC(Data Warehouse Center)에 대해서도 설명하는데 이는 운영중인 데이터베이스에서 웨어하우스 데이터베이스로 데이터를 이동하는 데 사용할 수 있습니다.

다음과 같은 내용이 들어 있습니다.

- 『플랫폼간 데이터 이동』
- 214 페이지의 『db2move를 사용하여 데이터 이동 도구』
- 220 페이지의 『DB2 Connect를 사용한 데이터 이동』
- 222 페이지의 『입력된 테이블간의 데이터 이동』
- 228 페이지의 『데이터 이동을 위해 복제 사용』
- 230 페이지의 『데이터를 이동하기 위해 Data Warehouse Center 사용』.

플랫폼간 데이터 이동

플랫폼간에 데이터를 내보내기, 가져오기 또는 로드할 때에는 호환성이 중요합니다. 다음 절은 다른 운영 체제간에 데이터를 이동시킬 때 PC/IXF, 컬럼 식별자가 있는 ASCII(DEL) 및 WSF 파일 형식의 고려사항에 대해 설명합니다. 239 페이지의 『부록C. 내보내기/가져오기/로드 유틸리티 파일 형식』에서 DB2 데이터 이동 유틸리티에서 사용할 수 있는 파일 형식에 관한 자세한 내용을 참조하십시오.

PC/IXF 파일 형식

PC/IXF는 플랫폼간에 데이터를 전송하기 위해 권장되는 파일 형식입니다. PC/IXF 파일은 로드 유틸리티나 가져오기 유틸리티가 머신 독립형 형식에서 숫자 데이터를 처리할 수 있도록 합니다(일반적으로 머신 종속형). 예를 들어, 숫자 데이터는 Intel과 기타 하드웨어 아키텍처에서 서로 다르게 저장되고 처리됩니다.

DB2 제품군의 모든 제품 사이에 PC/IXF 파일의 호환성을 제공하기 위해, 내보내기 유틸리티가 Intel 형식의 숫자 데이터가 있는 파일을 작성하고 가져오기 유틸리티도 이 형식의 파일을 예상합니다.

하드웨어 플랫폼에 따라 DB2 제품은 내보내기 및 가져오기 조작을 수행하는 동안 Intel과 Intel이 아닌 형식 사이에서 숫자 값을 변환합니다(바이트 전환 사용).

DB2의 UNIX 기본 구현은 내보내기 수행중 다중 파트 PC/IXF 파일을 작성하지 않습니다. 단, 사용자에게 DB2가 작성한 다중 파트 PC/IXF 파일을 가져올 수 있도록 허용합니다. 이 유형의 파일을 가져올 때, 모든 파트가 같은 디렉토리에 있어야 합니다. 그렇지 않으면, 오류가 리턴됩니다.

DB2 내보내기 유틸리티의 UNIX 기본 구현에 의해 작성된 단일 파트 PC/IXF 파일은 OS/2용 DB2 또는 Windows NT용 DB2에 의해 가져올 수 있습니다.

컬럼 식별자가 있는 ASCII(DEL) 파일 형식

DEL 파일들은 각기 작성된 운영 체제에 따라 차이가 있습니다. 그 차이점은 다음과 같습니다.

- 행 분리자 문자
 - UNIX 기본 텍스트 파일은 라인 피드(LF) 문자를 사용합니다.
 - UNIX가 아닌 기본 텍스트 파일은 캐리지 리턴/라인 피드(CRLF) 순서를 사용합니다.
- 파일 끝 문자
 - UNIX 기본 텍스트 파일에는 파일 끝 문자가 없습니다.
 - UNIX가 아닌 기본 텍스트 파일에는 파일 끝 문자(X'1A')가 있습니다.

DEL 내보내기 파일이 텍스트 파일이기 때문에 이 파일은 하나의 운영 체제에서 다른 운영 체제로 전송될 수 있습니다. 파일 전송 프로그램은 파일을 텍스트 모드로 전송할 경우, 운영 체제적 차이점을 처리할 수 있습니다. 행 분리자와 파일 끝 문자의 변환은 2진 모드에서는 수행되지 않습니다.

주: 문자 데이터 필드에 행 분리자 문자가 들어 있으면, 이는 파일 전송시 변환됩니다. 이러한 변환은 데이터에 예기치 않은 변경을 초래합니다. 이러한 이유 때

문에, 플랫폼간에 데이터를 이동시킬 때에는 DEL 내보내기 파일을 사용하지 않는 것이 좋습니다. 대신, PC/IXF 파일 형식을 사용하십시오.

WSF 파일 형식

WSF 형식 파일의 숫자 데이터는 Intel 머신 형식으로 저장됩니다. 이 형식은 Lotus WSF 파일이 다른 Lotus 운영 체제(예를 들어, Intel 기본 및 UNIX 기본 시스템)에서 사용되고 전송될 수 있도록 합니다.

내부 형식의 이러한 일관성의 결과로, DB2 제품에서 내보낸 WSF 파일은 다른 플랫폼에서 수행중인 Lotus 1-2-3 또는 Symphony에 의해 사용될 수 있습니다. DB2 제품은 다른 플랫폼에서 작성된 WSF 파일도 가져올 수 있습니다.

운영 체제간에 WSF 파일을 2진(텍스트가 아닌) 모드로 전송하십시오.

주: 데이터 유실이 일어날 수 있으므로, 다른 플랫폼과 DB2 데이터베이스간에 데이터를 전송하는 데 WSF 파일 형식을 사용하지 마십시오. 대신, PC/IXF 파일 형식을 사용하십시오.

db2move를 사용하여 데이터 이동 도구

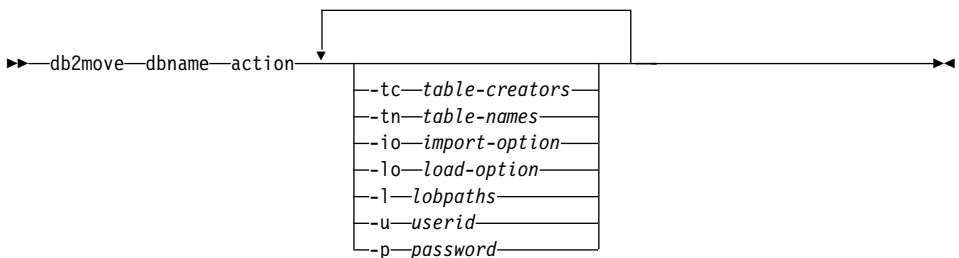
이 도구는 워크스테이션에 위치한 DB2 데이터베이스간의 여러 테이블의 이동을 원활하게 만듭니다. 이 도구는 특정 데이터베이스에 대한 시스템 카탈로그 테이블을 조회하고 모든 사용자 테이블의 목록을 컴파일합니다. 그런 다음, 이들 테이블을 PC/IXF 형식으로 내보냅니다. PC/IXF 파일은 같은 시스템상에 있는 다른 지역 DB2 데이터베이스로 가져오거나 로드될 수 있습니다. 또는, 다른 워크스테이션 플랫폼으로 전송되어 해당 플랫폼에 있는 DB2 데이터베이스로 가져오거나 로드될 수 있습니다.

주: 구조화된 유형 컬럼이 있는 테이블은 이 도구가 사용될 때 이동하지 않습니다.

권한 부여

이 도구는 사용자가 요청한 조치에 따라 DB2 내보내기, 가져오기 및 로드 API를 호출합니다. 따라서, 요청하는 사용자 ID에는 해당 API가 필요로 하는 올바른 권한 부여가 있어야 합니다. 그렇지 않으면, 요청이 실패합니다.

명령 구문



명령 매개변수

dbname

데이터베이스 이름

action EXPORT, IMPORT 또는 LOAD 중 하나여야 합니다.

-tc table-creators. 기본값은 모든 작성자입니다.

이는 단지 EXPORT 조치입니다. 지정될 경우, 이 옵션과 함께 나열된 작성자가 작성한 테이블만 내보냅니다. 지정되지 않을 경우, 기본값은 모든 작성자를 사용하는 것입니다. 여러 작성자를 지정할 때, 각 작성자는 쉼표로 구분되어야 합니다. 작성자 ID 사이에 공백을 사용할 수 없습니다. 지정될 수 있는 최대 작성자 수는 10입니다. 이 옵션은 내보내기용 테이블을 선택하기 위해 『-tn』 table-names 옵션과 함께 사용될 수 있습니다.

문자열에서 어느 곳에나 들어갈 수 있는 와일드카드 문자로 별표(*)를 사용할 수 있습니다.

-tn table-names. 기본값은 모든 사용자 테이블입니다.

이는 단지 EXPORT 조치입니다. 지정될 경우, 이름이 지정된 문자열에 있는 이름과 실제로 일치하는 테이블만 내보냅니다. 지정되지 않을 경우, 기본값은 모든 사용자 테이블을 사용하는 것입니다. 여러개의 테이블 이름을 지정할 때, 각 테이블 이름은 쉼표로 구분되어야 합니다. 테이블 이름 사이에 공백을 사용할 수 없습니다. 지정될 수 있는 최대 테이블 이름 수는 10입니다. 이 옵션은 내보내기용 테이블을 선택하기 위해 『-tc』 table-creators 옵션과 함께 사용되어야 합니다. db2move는 해당 이름이 지정된 테이블 이름과 일치하고 작성자가 지정된 테이블 작성자와 일치하는 테이블만 내보냅니다.

문자열에서 어느 곳에나 들어갈 수 있는 와일드카드 문자로 별표(*)를 사용할 수 있습니다.

-io import-option. 기본값은 REPLACE_CREATE입니다.

유효한 옵션은 INSERT, INSERT_UPDATE, REPLACE, CREATE 및 REPLACE_CREATE입니다.

-lo load-option. 기본값은 INSERT입니다.

유효한 옵션은 INSERT 및 REPLACE입니다.

-l lobpaths. 기본값은 현재 디렉토리입니다.

이 옵션은 LOB 파일이 작성되거나(EXPORT의 부분으로서) 검색되는 (IMPORT 또는 LOAD의 부분으로서) 완전한 경로 이름을 지정합니다. 다중 LOB 경로를 지정할 때, 각 경로 이름은 쉼표로 구분되어야 합니다. LOB 경로 사이에 공백을 사용할 수 없습니다. 첫번째 경로가 공간을 모

db2move를 사용하여 데이터 이동 도구

두 사용하거나(EXPORT 조작중), 파일이 경로에서 발견되지 않으면 (IMPORT나 LOAD 조작중) 두번째 경로가 사용되고, 계속 그 다음 경로가 사용됩니다.

EXPORT 조치시 LOB 경로가 지정되어 있으면, LOB 경로 디렉토리에 있는 모든 파일이 삭제되고 디렉토리가 삭제된 후, 새로운 디렉토리가 작성됩니다. 지정되어 있지 않으면, 현재 디렉토리가 LOB 경로에 사용됩니다.

-u 사용자 ID. 기본값은 로그인되어 있는 사용자 ID입니다.

사용자 ID 및 암호는 모두 선택적입니다. 단, 하나가 지정되면 다른 하나도 지정되어야 합니다. 원격 서버에 연결하는 클라이언트에서 명령이 수행되는 경우, 사용자 ID와 암호가 지정되어야 합니다.

-p 암호. 기본값은 로그인된 암호입니다.

사용자 ID 및 암호는 모두 선택적입니다. 단, 하나가 지정되면 다른 하나도 지정되어야 합니다. 원격 서버에 연결하는 클라이언트에서 명령이 수행되는 경우, 사용자 ID와 암호가 지정되어야 합니다.

예

- db2move sample export

이 명령은 SAMPLE 데이터베이스에 있는 모든 테이블을 내보냅니다. 모든 옵션에 대해 기본값이 사용됩니다.

- db2move sample export -tc userid1,us*rid2 -tn tname1,*tname2

이 명령은 『userid1』 또는 사용자 ID LIKE 『us%rid2』에 의해 작성되고, 『tname1』 또는 테이블 이름 LIKE 『%tname2』를 갖는 모든 테이블을 내보냅니다.

- db2move sample import -l D:\LOBPATH1,C:\LOBPATH2

이 예는 OS/2 또는 Windows 운영 체제에만 적용 가능합니다. 이 명령은 SAMPLE 데이터베이스에 있는 모든 테이블을 가져옵니다. LOB 파일에 대해 LOB 경로 『D:\LOBPATH1』과 『C:\LOBPATH2』를 검색합니다.

- db2move sample load -l /home/userid/lobpath,/tmp

이 예는 UNIX 기본 시스템에만 적용 가능합니다. 이 명령은 SAMPLE 데이터베이스에 있는 모든 테이블을 로드합니다. LOB 파일에 대해 /home/userid/lobpath 서브디렉토리와 tmp 서브디렉토리를 모두 검색합니다.

- `db2move sample import -io replace -u userid -p password`

이 명령은 REPLACE 모드에서 SAMPLE 데이터베이스에 있는 모든 테이블을 가져옵니다. 지정된 사용자 ID와 암호가 사용됩니다.

사용시 주의사항

이 도구는 사용자 작성 테이블을 내보내기, 가져오기 또는 로드합니다. 한 운영 체제로부터 다른 운영 체제로 데이터베이스를 복제할 경우, **db2move**는 테이블의 이동을 도와줍니다. 테이블과 연관된 다른 모든 오브젝트 즉, 별명, 뷰, 트리거, 사용자 정의 기능 등을 이동시키는 데에도 필요합니다. **db2look**(DB2 통계 및 DDL 추출 도구; *Command Reference* 참조)은 데이터베이스로부터 DDL을 발췌하여 이들 오브젝트의 일부를 이동시키는 데 도움을 줄 수 있습니다.

내보내기, 가져오기 또는 로드 API가 **db2move**에 의해 호출될 때 FileTypeMod 매개변수는 lobsinfile로 설정됩니다. 즉, LOB 데이터는 PC/IXF 파일과 다른 파일에 보관됩니다. 26 000 파일 이름이 LOB 파일에 대해 사용 가능합니다.

LOAD 조치는 데이터베이스 및 데이터 파일이 상주하는 머신에서 지역적으로 수행되어야 합니다. **db2move**에 의해 로드 API가 호출되면, CopyTargetList 매개변수는 널(NULL)로 설정됩니다. 즉, 복사가 수행되지 않습니다. *logretain*이 켜지면, 나중에 로드 조치가 롤 포워드될 수 없습니다. 로드된 테이블이 상주하는 테이블 공간이 백업 보류 상태로 되고 액세스할 수 없게 됩니다. 전체 데이터베이스 백업 또는 테이블 공간 백업을 하려면, 테이블 공간을 백업 보류 상태에서부터 나오게 해야 합니다.

버전 5.2 클라이언트에서 버전 6 데이터베이스에 대해 발행하면, 길이가 18자 상인 테이블 또는 컬럼 이름을 이 도구에서 지원하지 않습니다.

EXPORT 사용시 요구/생성된 파일

- 입력: 없음
- 출력:

db2move를 사용하여 데이터 이동 도구

EXPORT.out EXPORT 조치의 요약된 결과

db2move.lst 원래 테이블 이름, 이 테이블의 상응하는 PC/IXF 파일 이름 (tabnnn.ixf) 및 메시지 파일 이름(tabnnn.msg)의 목록. 이 목록, 내보낸 PC/IXF 파일 및 LOB 파일(tabnnnc.yyy)은 **db2move** IMPORT 또는 LOAD 조치에 대한 입력으로 사용 됩니다.

tabnnn.ixf 특정 테이블의 내보낸 PC/IXF 파일

tabnnn.msg 상응하는 테이블의 내보내기 메시지 파일

tabnnnc.yyy 특정 테이블의 내보낸 LOB 파일

『nnn』은 테이블 번호입니다. 『c』는 하나의 영문자입니다. 『yyy』는 001-999 범위의 한 숫자입니다.

이들 파일은 내보내기중인 테이블에 LOB 데이터가 있을 경우에만 작성됩니다. 작성된 경우, 이들 LOB 파일은 『lobpath』 디렉토리에 위치합니다. LOB 파일에 대해 모두 26 000의 이름이 사용 가능합니다.

system.msg 파일이나 디렉토리 명령을 작성 또는 삭제에 위한 시스템 메시지가 들어 있는 메시지 파일. 이 파일은 EXPORT 조치 및 LOB 경로가 지정된 경우에만 사용됩니다.

IMPORT 사용시 요구/생성된 파일

- 입력:

db2move.lst EXPORT 조치의 출력 파일

tabnnn.ixf EXPORT 조치의 출력 파일

tabnnnc.yyy EXPORT 조치의 출력 파일

- 출력:

IMPORT.out IMPORT 조치의 요약된 결과

tabnnn.msg 상응하는 테이블의 가져오기 메시지 파일

LOAD 사용시 요구/생성된 파일

- 입력:

db2move.lst EXPORT 조치의 출력 파일

tabnnn.ixf EXPORT 조치의 출력 파일

tabnnnc.yyy EXPORT 조치의 출력 파일

• 출력:

LOAD.out LOAD 조치의 요약된 결과

tabnnn.msg 상응하는 테이블의 로드 메시지 파일

DB2 Connect를 사용한 데이터 이동

호스트 데이터베이스 시스템과 워크스테이션간에 데이터를 이동시킬 필요가 있는 복잡한 환경에서 작업할 경우, 워크스테이션에서 호스트로 데이터 전송뿐 아니라 호스트에서 워크스테이션으로 데이터 전송을 위한 게이트웨이로 DB2 Connect를 사용할 수 있습니다(그림9 참조).

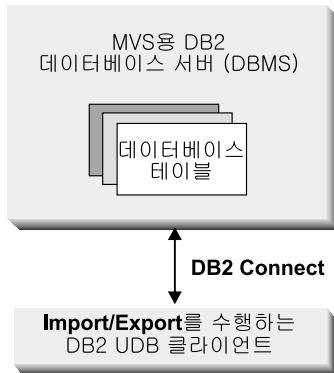


그림 9. DB2 Connect를 통한 가져오기/내보내기

다음 절에서는 DB2 Connect를 사용하여 데이터를 내보내기 또는 가져오기하는 내용을 설명합니다.

내보내기 및 가져오기 유틸리티 사용

DB2 내보내기 및 가져오기 유틸리티는 사용자가 DRDA 서버 데이터베이스로부터 DB2 Connect 워크스테이션상의 파일로 데이터 이동 및 이와 반대 방향으로 이동시킬 수 있도록 합니다. 그런 다음, 이 내보내기 또는 가져오기 형식을 지원하는 관계형 데이터베이스 관리 시스템 또는 기타 모든 응용프로그램에서 데이터를 사용할 수 있습니다. 예를 들어, MVS/ESA용 DB2에서 컬럼 식별자가 있는 ASCII 파일로 데이터를 내보낸 후, OS/2용 DB2 데이터베이스에 가져올 수 있습니다.

데이터베이스 클라이언트나 DB2 Connect 워크스테이션에서 내보내기 및 가져오기 조작을 수행할 수 있습니다.

주:

1. 내보내거나 가져올 데이터는 두 데이터베이스 모두에 적용할 수 있는 크기 및 데이터 유형 제한사항에 따라야 합니다.
2. 가져오기 성능을 향상시키기 위해 복합 SQL을 사용할 수 있습니다. 가져오기 유틸리티에 있는 복합 파일 유형 수정자를 지정하여 지정된 수의 SQL문을 블록으로 그룹화하십시오(67 페이지의 『파일 유형 수정자(가져오기)』 참조). 이는 네트워크 오버헤드를 줄이고 응답 시간을 향상시킵니다.
3. 9 페이지의 『EXPORT 명령』 및 42 페이지의 『IMPORT 명령』에서 내보내기 및 가져오기 유틸리티의 구문에 관한 자세한 정보를 참조하십시오.

워크스테이션에서 DRDA 서버로 데이터 이동

DRDA 서버 데이터베이스로 데이터를 이동시키려면, 다음을 수행하십시오.

1. DB2 테이블에서 PC/IXF 파일로 데이터를 내보내십시오.
2. INSERT 옵션을 사용하여 PC/IXF 파일을 DRDA 서버 데이터베이스의 호환 가능한 테이블로 가져오십시오.

DRDA 서버에서 워크스테이션으로 데이터 이동

DRDA 서버 데이터베이스에서 데이터를 이동시키려면, 다음을 수행하십시오.

1. DRDA 서버 데이터베이스 테이블에서 PC/IXF 파일로 데이터를 내보내십시오.
2. PC/IXF 파일을 DB2 테이블로 가져오십시오.

제한사항

DB2 Connect에서, 내보내기 및 가져오기 조작은 다음 조건을 충족시켜야 합니다.

- 파일이 PC/IXF 유형이어야 합니다.
- 가져오기 전에 데이터와 호환성이 있는 속성을 갖는 테이블이 있어야 합니다. DB2 Connect를 통해 가져올 경우, 테이블을 작성할 수 없습니다. 단지, INSERT만 지원되는 옵션이기 때문입니다.
- 가져오기 조작에 대해 확약 카운트 간격이 지정되어 있지 않아야 합니다.

이러한 조건 중 하나라도 충족되지 않으면, 조작이 실패하고 오류 메시지가 리턴됩니다.

주: 색인 정의는 내보내기에 저장되지 않거나 가져오기에 사용됩니다.

DB2 Connect를 사용한 데이터 이동

1바이트 및 2바이트 혼합 데이터

혼합 데이터(1바이트와 2바이트 데이터를 모두 포함하는 컬럼)를 내보내기 또는 가져오기할 경우, 다음을 고려하십시오.

- 데이터를 EBCDIC로 저장하는 시스템에서(MVS, OS/390, OS/400, VM 및 VSE), 쉬프트인과 쉬프트아웃 문자는 2바이트 데이터의 시작과 끝을 표시합니다. 데이터베이스 테이블에 대해 컬럼 길이를 정의할 때에는 반드시 이들 문자를 위한 공간을 허용하십시오.
- 컬럼 데이터가 일치된 패턴을 갖지 않는 한, 가변 길이 문자 컬럼을 사용하는 것이 좋습니다.

입력된 테이블간의 데이터 이동

입력된 테이블의 밖으로 및 테이블 안으로 데이터를 이동시키기 위해 DB2 내보내기 및 가져오기 유틸리티를 사용할 수 있습니다. 입력된 테이블은 계층 구조로 되어 있을 수 있습니다. 계층 구조간 데이터 이동은 다음과 같습니다.

- 한 계층 구조에서 동일한 계층 구조로 이동
- 한 계층 구조에서 더 큰 계층 구조의 부속 섹션으로 이동
- 큰 계층 구조의 부속 섹션에서 다른 계층 구조로 이동

IMPORT CREATE 옵션을 사용하여 테이블 계층 구조와 유형 계층 구조를 모두 작성할 수 있습니다.

계층 구조에서 유형의 실체는 데이터베이스에 따라 다릅니다. 이는 서로 다른 데이터베이스에 있는 동일한 유형이 다른 식별자를 갖는다는 것을 의미합니다. 따라서, 이러한 데이터베이스간에 데이터를 이동시킬 때에는 데이터가 올바르게 이동되었다는 것을 확인하기 위해 동일 유형의 맵핑이 수행되어야 합니다.

입력된 각 행이 내보내기 조작중 작성되기 전에 식별자는 색인 값으로 변환됩니다. 이 색인 값은 1부터 계층 구조에 있는 상대 유형의 번호 중 모든 숫자가 될 수 있습니다. 색인 값은 특정 순서로 된 계층 구조를 통해 이동할 때 각 유형에 번호를 매김으로써 생성됩니다. 이 순서를 트래버스 순서라고 합니다. 이는 계층 구조에 있는 모든 상위 테이블과 서브테이블을 위에서 아래로, 좌에서 우로 처리하는

순서입니다. 트레이스 순서는 다른 데이터와 관련하여 데이터가 이동되는 곳을 결정하기 때문에 테이블 계층 구조간에 데이터를 이동시킬 때 중요합니다.

하나의 방법은 계층 구조의 맨 위(또는 루트 테이블)부터 진행하여 맨 아래 서브 테이블까지 계층 구조(서브테이블)를 내려간 다음, 상위 테이블에 백업하고, 다음 번 『맨 오른쪽』 하나 이상의 서브테이블로 내려간 다음, 다음으로 높은 상위 테이블에 백업하고 서브테이블로 내려가는 순서입니다.

다음 그림에서는 네 가지 유효한 트레이스 순서로 된 계층 구조를 보여줍니다.

- Person, Employee, Manager, Architect, Student.
- Person, Student, Employee, Manager, Architect(이 트레이스 순서는 점선으로 표시됨)
- Person, Employee, Architect, Manager, Student.
- Person, Student, Employee, Architect, Manager.

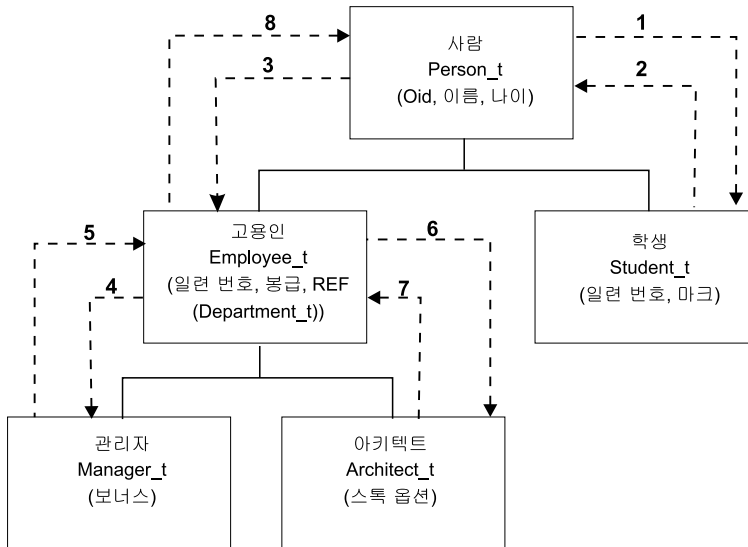


그림 10.

트래버스 순서

트래버스 순서에 있는 모든 상대 유형은 계층 구조에서 지정된 시작점으로부터 계층 구조에서 도달 가능한 모든 유형을 참조합니다. 기본 순서는 계층 구조에 있는 모든 테이블을 포함하고, 각 테이블은 OUTER 순서 술어에 사용된 스키마에 의해 순서가 지정됩니다. 또한, 사용자가 사용될 상대 유형을 정의한(트래버스 순서 목록에서) 사용자 정의 트래버스 순서도 있습니다. 내보내기 유틸리티 및 가져오기 유틸리티를 호출할 때에는 같은 트래버스 순서를 사용해야 합니다.

트래버스 순서를 지정할 경우, 서브테이블이 PRE-ORDER 형식으로 트래버스되어야 합니다(즉, 계층 구조의 각 브랜치는 새로운 브랜치가 시작되기 전에 맨 아래로 트래버스되어야 합니다).

기본 트래버스 순서

기본 트래버스 순서는 다른 파일 형식에서 사용될 때에는 다르게 작용합니다. 다음에서는 동일한 테이블 계층 구조와 유형 관계를 가정합니다.

PC/IXF 파일 형식에 데이터를 내보내면, 모든 상대 유형과 정의 및 상대 테이블을 작성합니다. 내보내기는 또한 각 테이블에 대한 색인 값의 맵핑을 완성합니다. 이 맵핑은 가져오기 조작시 목표 데이터베이스에 대한 정확한 데이터 이동을 확인하는 데 사용됩니다. PC/IXF 파일 형식으로 작업할 경우, 기본 트래버스 순서를 사용해야 합니다.

ASC, DEL 또는 WSF 파일 형식의 경우, 소스와 목표 계층 구조가 구조적으로 동일하더라도 입력된 행과 입력된 테이블이 작성된 순서가 다를 수 있습니다. 이 결과는 기본 트래버스 순서가 계층 구조를 통해 진행할 때 식별할 시간차로 나타납니다. 각 유형별 작성 시간은 기본 트래버스 순서를 사용하여 소스와 목표의 계층 구조를 통과하는 순서를 결정합니다. 소스와 목표의 두 계층 구조에 있는 각 유형의 작성 순서가 동일하며, 소스와 목표가 구조적으로 동일하지 확인하십시오. 이러한 조건이 충족될 수 없으면, 사용자 지정 트래버스 순서를 선택하십시오.

사용자 지정 트래버스 순서

계층 구조에서 트래버스 순서를 제어하려 할 경우, 내보내기 및 가져오기 유틸리티에 대해 같은 트래버스 순서가 사용되는지 확인하십시오. 즉, 다음과 같은 경우입니다.

- 소스 및 목표 데이터베이스에 있는 서브테이블 정의가 동일함
- 소스 및 목표 데이터베이스에 있는 서브테이블간의 계층 구조적 관계가 동일함
- 동일한 트래버스 순서

가져오기 유틸리티가 목표 데이터베이스에 정확한 데이터 이동을 보장합니다.

트래버스 순서를 정의할 때 계층 구조 아래에 시작점과 경로를 지정하더라도, 각 브랜치는 끝까지 트래버스되어야만 계층 구조의 다음 브랜치를 시작할 수 있습니다. 내보내기 및 가져오기 유틸리티는 지정된 트래버스 순서에서 이 조건이 위반된 곳을 찾습니다.

데이터 이동중 선택

입력된 테이블의 하나의 계층적 구조에서 다른 구조로의 데이터 이동은 특정 트래버스 순서와 중간 플랫폼 파일을 통해 수행됩니다. 내보내기 유틸리티(트래버스 순서와 관련하여)는 해당 파일에 배치되는 내용을 제어합니다. 사용자는 목표 테이블 이름과 WHERE절만 지정하면 됩니다. 내보내기 유틸리티는 이러한 선택 기준을 사용하여 적합한 중간 파일을 작성합니다.

가져오기 유틸리티는 목표 데이터베이스에 배치되는 내용을 제어합니다. 목표 데이터베이스로 이동된 속성을 제한하기 위해 각 서브테이블 이름 끝에 속성 목록을 지정할 수 있습니다. 속성 목록을 사용하지 않으면, 각 서브테이블에 있는 모든 컬럼이 이동됩니다.

가져오기 유틸리티는 CREATE, INTO 테이블 이름, UNDER 및 AS ROOT TABLE 매개변수를 통해 이동되는 계층 구조의 크기 및 배치를 조정합니다. 42 페이지의 『IMPORT 명령』에서 IMPORT 명령 매개변수에 관한 자세한 정보를 참조하십시오.

입력된 테이블간의 데이터 이동 예

이 절에 나와 있는 예는 다음은 계층적 구조를 기준으로 합니다.

입력된 테이블간의 데이터 이동

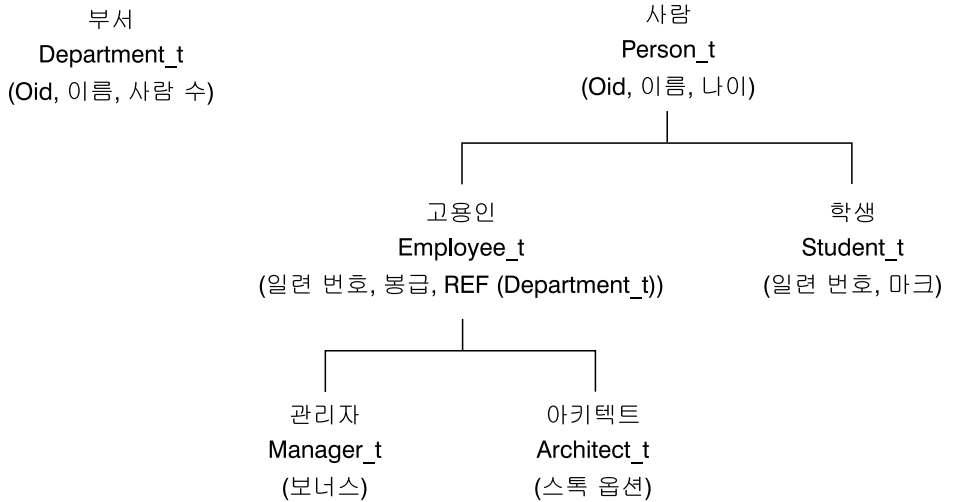


그림 11.

예 1

전체 계층 구조를 내보낸 다음 가져오기 조사를 통해 재작성하려면, 다음을 수행하십시오.

```

DB2 CONNECT TO Source_db
DB2 EXPORT TO entire_hierarchy.ixf OF IXF HIERARCHY STARTING Person
DB2 CONNECT TO Target_db
DB2 IMPORT FROM entire_hierarchy.ixf OF IXF CREATE INTO
HIERARCHY STARTING Person AS ROOT TABLE
  
```

계층 구조에 있는 각 유형은 유형이 없을 경우 작성됩니다. 이들 유형이 이미 있을 경우에는 소스 데이터베이스에 있는 정의와 동일한 정의가 목표 데이터베이스에 있어야 합니다. 이들 유형이 동일하지 않으면, SQL 오류(SQL20013N)가 리턴됩니다. 새로운 계층 구조를 작성하지 않기 때문에, 목표 데이터베이스(Target_db)로 이동되는 데이터 파일에 정의된 서브테이블이 존재할 수 없습니다. 소스 데이터베이스 계층 구조에 있는 각 테이블이 작성됩니다. 소스 데이터베이스의 데이터가 목표 데이터베이스의 정확한 서브테이블에 가져오기됩니다.

예 2

좀더 복잡한 예에서, 소스 데이터베이스의 전체 계층 구조를 내보내 목표 데이터베이스에 가져오려고 합니다. 21살 이상의 사람에 대한 모든 데이터를 내보내더라도, 선택된 데이터만 목표 데이터베이스에 가져옵니다.

```
DB2 CONNECT TO Source_db
DB2 EXPORT TO entire_hierarchy.del OF DEL HIERARCHY (Person,
Employee, Manager, Architect, Student) WHERE Age>=20
DB2 CONNECT TO Target_db
DB2 IMPORT FROM entire_hierarchy.del OF DEL INSERT INTO (Person,
Employee(Salary), Architect) IN HIERARCHY (Person, Employee,
Manager, Architect, Student)
```

목표 테이블 Person, Employee 및 Architect가 모두 존재해야 합니다. Person, Employee 및 Architect 서브테이블에 데이터가 가져오기됩니다. 즉, 다음과 같이 가져옵니다.

- Person에 있는 모든 컬럼을 Person에
- Employee의 Person과 Salary에 있는 모든 컬럼을 Employee에
- Employee의 Person과 Salary에 있는 모든 컬럼과, Architect에 있는 모든 컬럼을 Architect에

컬럼 SerialNum 및 REF(Employee_t)는 Employee 또는 서브테이블(데이터가 가져오기 되는 유일한 서브테이블인 Architect)에 가져오지 않습니다.

주: Architect가 Employee의 서브테이블이고 Employee에 대해 지정된 유일한 가져오기 컬럼이 Salary이기 때문에, Salary는 Architect에 가져온 유일한 Employee 특정 컬럼이 됩니다. 즉, SerialNum이나 REF(Employee_t) 컬럼은 모두 Employee 또는 Architect 행에 가져오지 않습니다.

Manager와 Student 테이블에 대한 데이터는 가져오지 않습니다.

예 3

이 예에서는 일반 테이블로부터 내보내 계층 구조에 있는 단일 서브테이블 자격으로 가져옵니다. EXPORT 명령은 일반(입력되지 않은) 테이블에서 조작하므로, 데이터 파일에 Type_id 컬럼이 존재하지 않습니다. 이를 나타내기 위해 수정자 no_type_id가 사용되므로, 가져오기 유틸리티는 첫번째 컬럼이 Type_id 컬럼이 될 것이라고 예상하지 않습니다.

```
DB2 CONNECT TO Source_db
DB2 EXPORT TO Student_sub_table.del OF DEL SELECT * FROM
Regular_Student
DB2 CONNECT TO Target_db
DB2 IMPORT FROM Student_sub_table.del OF DEL METHOD P(1,2,3,5,4)
MODIFIED BY NO_TYPE_ID INSERT INTO HIERARCHY (Student)
```

입력된 테이블간의 데이터 이동

이 예에서는 목표 테이블 Student가 존재해야 합니다. Student가 서브테이블이기 때문에, 첫번째 컬럼에 Type_id가 존재하지 않는다는 것을 나타내기 위해 수정자 no_type_id가 사용됩니다. 단, Student 테이블에 있는 다른 모든 속성 외에도 Object_id가 있는지 확인해야 합니다. Object-id는 Student 테이블에 가져오기된 각 행에서 첫번째 컬럼이 됩니다. METHOD절은 마지막 두 속성의 순서를 바꾸어 놓습니다.

데이터 이동을 위해 복제 사용

복제를 사용하여 일반적인 체제상의 데이터를 복수 원격 데이터베이스로 복사할 수 있습니다. 마스터 데이터베이스에 대한 갱신사항을 다른 데이터베이스로 자동으로 복사하게 할 필요가 있는 경우, DB2의 복제 기능을 사용하여 복사되어야 할 데이터와 데이터가 복사되어야 할 데이터베이스 테이블 및 갱신사항이 복사되어야 하는 주기를 지정할 수 있습니다. DB2에 있는 복제 기능은 크고 작은 엔터프라이즈에서 데이터를 복제하기 위한 대형 IBM 솔루션의 일부입니다.

IBM 복제 도구는 분산 관계형 데이터베이스 관리 시스템간에 데이터를 복사하는 DB2 DataPropagator 프로그램 및 DB2 Universal Database 도구의 세트입니다.

- DB2 Universal Database 플랫폼간
- DB2 Universal Database 플랫폼 및 DRDA 연결성을 지원하는 호스트 데이터베이스
- DRDA 연결성을 지원하는 호스트 데이터베이스간

DB2 DataJoiner를 통해 IBM이 아닌 관계형 데이터베이스 관리 시스템으로 데이터를 복제할 수도 있습니다.

IBM 복제 도구를 사용하여 엔터프라이즈 내의 모든 데이터에 대해 단일 제어점으로부터 복사 조작을 정의, 동기화, 자동화 및 관리할 수 있습니다. DB2 Universal Database의 복제 도구는 관계형 데이터베이스간에서만 복제 기능을 제공합니다. 이들은 또한 IMS DataPropagator(이전에는 DPropNR)와 결합하여 IMS 및 VSAM 데이터를 복제하고 Lotus NotesPump와 결합하여 Lotus Notes 데이터베이스로(부터) 복제합니다.

복제 기능을 사용하여 실행 데이터베이스에 추가 로드를 하지 않고도 일반 사용자 및 응용프로그램이 실행 데이터를 액세스하게 할 수 있습니다. 사용자나 응용프로그램이 데이터를 원격으로 액세스하게 하는 것보다는 지역 데이터베이스에 데이터

를 복사할 수 있습니다. 일반적인 복제 시나리오에는 하나 이상의 원격 데이터베이스에 사본을 가지고 있는 소스 테이블이 관련됩니다. 예를 들면, 중앙 은행과 지점입니다. 예정된 시간에, DB2 데이터베이스의 자동 갱신이 일어나고 소스 데이터베이스에 대한 모든 변경사항이 목표 데이터베이스 테이블에 복사됩니다.

복제 도구를 사용하여 사본 테이블 구조를 사용자 정의할 수 있습니다. 데이터 복사를 향상시키기 위해 목표 데이터베이스에 복사할 때 SQL을 사용할 수 있습니다. 소스 테이블을 복사하고, 적절한 시기에 지정된 지점에서 데이터를 캡처하고, 변경사항의 실행기록을 제공하거나 추가 목표 테이블에 복사될 데이터를 준비하는 읽기 전용 사본을 작성할 수 있습니다. 또한, 일반 사용자나 응용프로그램에 의해 갱신될 수 있는 읽기 및 쓰기 사본을 작성하여 복제된 변경사항을 마스터 테이블로 다시 돌려 보낼 수 있습니다. 소스 테이블의 뷰 또는 사본의 뷰를 복제할 수 있습니다. 이벤트 지향의 복제도 가능합니다.

사용자는 다음의 플랫폼 즉, AIX, AS/400, HP-UX, Linux, Microsoft Windows 95, 98, 2000 및 NT, OS/2, OS/390, SCO UnixWare, Sun Solaris, Sequent, VM 및 VSE에서 DB2 데이터베이스간에 데이터를 복제할 수 있습니다. 또한 DB2와 다음의 비DB2 데이터베이스(DB2 DataJoiner가 있는) 즉, Informix, Microsoft Jet, Microsoft SQL Server, Oracle, Sybase 및 Sybase SQLAnywhere 간에 데이터를 복제할 수 있습니다. 사용자는 다른 IBM 제품과 결합해서 DB2 데이터를 IMS, VSAM 또는 Lotus Notes로(부터) 복제할 수 있습니다. 마지막으로 Windows CE 또는 Palm OS 장치의 DB2 Everywhere로 데이터를 복제할 수 있습니다.

구성요소에 의한 IBM 복제 도구

IBM 복제 도구 솔루션의 두 가지 구성요소는 Capture 프로그램과 Apply 프로그램입니다. DB2 제어 센터를 사용하여 이들 구성요소를 설정할 수 있습니다. 이들 구성요소의 조작 및 모니터링은 제어 센터 밖에서 수행됩니다.

Capture 프로그램은 소스 테이블에 대한 변경사항을 캡처합니다. 소스 테이블은 다음이 될 수 있습니다.

- 파일 시스템 또는 DB2 DataPropagator 외부에 로드된 관계형이 아닌 데이터베이스 관리 프로그램의 SQL 데이터가 들어 있는 외부 테이블.
- 데이터베이스에 있는 기존 테이블.

데이터 이동을 위해 복제 사용

- Apply 프로그램에 의해 이미 갱신된 테이블. 변경사항을 소스 테이블에 다시 복사하거나 다른 목표 테이블에 복사하는 것을 허용합니다.

변경사항은 데이터 변경 테이블에 복사되며, 목표 시스템이 이 변경사항을 복사할 준비가 될 때까지 이 테이블에 저장됩니다. 그런 다음, Apply 프로그램이 데이터 변경 테이블로부터 이 변경사항을 가져다가 목표 테이블에 복사합니다.

제어 센터를 사용하여 다음을 수행하십시오.

- 복제 환경을 설정하십시오.
- 소스 및 목표 테이블을 정의하십시오.
- 자동 복사 시간을 지정하십시오.
- 데이터에 대한 SQL 추가 처리를 지정하십시오.
- 소스와 목표 테이블간의 관계를 지정하십시오.

복제 안내 및 참조서에서 자세한 정보를 참조하십시오.

데이터를 이동하기 위해 Data Warehouse Center 사용

DWC(Data Warehouse Center)를 사용하여 조작가능한 데이터베이스로부터 웨어하우스 데이터베이스로 데이터를 이동할 수 있으며 이는 일반 사용자들이 결정 지원을 위해 조회할 수 있습니다. 또한 DWC를 사용하여 *sources*라는 조작가능한 데이터베이스의 구조를 정의할 수 있습니다. 그런 다음, 조작가능한 데이터가 이동되어 웨어하우스용으로 변환되는 방법을 지정할 수 있습니다. *targets*이라는, 웨어하우스 데이터베이스에 있는 테이블의 구조를 모델로 하거나 데이터 이동 조작을 정의하는 프로세스의 일부분으로 테이블을 자동 빌드할 수 있습니다.

Data Warehouse Center에서는 다음의 DB2 기능을 사용하여 데이터를 이동 및 변환합니다.

- SQL

SQL을 사용하여 소스에서 데이터를 선택하여 목표에 데이터를 삽입할 수 있습니다. 또한 SQL을 사용하여 웨어하우스 형식으로 데이터를 변환할 수 있습니다. 사용자는 Data Warehouse Center를 사용하여 SQL을 생성하거나 자신의 SQL을 작성할 수 있습니다.

- 로드 및 내보내기 유틸리티

이 DB2 유틸리티를 사용하여 소스에서 데이터를 내보낸 후 목표에 데이터를 로드할 수 있습니다. 이 유틸리티들은 많은 양의 데이터를 이동해야 할 경우에 도움이 됩니다. Data Warehouse Center는 다음과 같은 유형의 로드 및 내보내기 조작을 지원합니다.

DB2 데이터 내보내기

지역 DB2 데이터베이스에서 분리 파일로 데이터를 내보냅니다.

ODBC 데이터 내보내기

ODBC로 등록된 데이터베이스 내의 테이블에서 데이터를 선택한 후, 분리 파일에 데이터를 기록합니다.

DB2 로드

분리 파일에서 DB2 테이블로 데이터를 로드합니다.

DB2 UDB EEE 데이터베이스로 DB2 로드(AIX 전용)

테이블에 있는 기존 데이터를 새로운 데이터로 대체하여 분리 파일로부터 DB2 Universal Database Enterprise - Extended Edition으로 데이터를 로드합니다. 이 조작에서는 데이터베이스에 대한 목표 파티션 맵을 확보하여, 각 파일이 데이터베이스 파티션에 로드될 수 있도록 입력 파일을 분할한 후, 모든 파티션에 대해 원격 로드 조작을 수행합니다.

- 복제

또한 복제 기능을 사용하여 웨어하우스 소스에서 웨어하우스 목표로 많은 양의 데이터를 복사한 후 소스 데이터에 대한 후속 변경사항들을 캡처할 수 있습니다. Data Warehouse Center는 다음과 같은 유형의 복제를 지원합니다.

기본 총계

사용자 테이블에 대한 총계 데이터가 들어 있으며 지정된 간격으로 추가되는 목표 테이블을 작성합니다.

변경 총계

총계 데이터가 들어 있고 소스 테이블에 대해 기록된 변경사항에 근거한 목표 테이블을 작성합니다.

데이터 이동을 위해 복제 사용

특정 시점

소스 테이블과 일치하는 목표 테이블을 작성하고 시간소인 컬럼을 목표 테이블에 추가합니다.

올림 테이블

갱신된 데이터에 대한 소스로서 여러 목표 테이블에 대해 사용할 수 있는 "consistent-change-data" 테이블을 작성합니다.

사용자 복사

복사를 수행할 때 소스 테이블과 일치하는 목표 테이블을 작성합니다.

이 옵션들은 모든 DB2 Universal Database 워크스테이션 운용 환경, OS/390용 DB2 Universal Database, AS/400용 DB2 및 DataJoiner에서 지원됩니다.

- 변환기 저장 프로시듀어

사용자는 Data Warehouse Center를 사용하여 OLAP(온라인 분석 처리) 데이터베이스로 데이터를 이동할 수 있습니다. 데이터가 웨어하우스에 있으면 변환기 저장 프로시듀어를 사용하여 데이터를 크린업한 후 이를 사실 및 차원 테이블에 쌓을 수 있습니다. 또한 변환기를 사용하여 통계 데이터를 생성할 수 있습니다. 데이터가 크린업되고 변환되고 나면, 이를 OLAP 큐브로 로드하거나 때로 *datamarts*라고 불리는 부문 서버로 이를 복제합니다. 변환기는 일부 DB2 제품품에만 포함됩니다. 보다 자세한 내용은 담당 IBM 대표에게 문의하십시오.

Data Warehouse Center 관리 안내서에서 Data Warehouse Center에 대한 자세한 내용을 참조하십시오.

부록A. 구문 도표를 읽는 방법

구문 도표에서는 운영 체제가 입력된 명령을 정확하게 해석할 수 있도록 명령이 지정되는 방법을 보여줍니다.

구문 도표는 왼쪽에서 오른쪽으로, 위에서 아래로, 수평 행(기본 경로)을 따라 읽으십시오(주 경로). 이 행이 화살표로 끝나면 명령 구문이 계속되는 것이며, 다음 행도 화살표로 시작됩니다. 수직 막대는 명령 구문의 끝을 표시합니다.

구문 도표로부터 정보를 입력할 때에는 반드시 따옴표 및 등호와 같이 구두점을 포함하도록 하십시오.

매개변수는 키워드나 변수로 분류됩니다.

- 키워드는 상수를 나타내며 대문자로 표시됩니다. 단, 명령 프롬프트에서는 키워드가 대문자, 소문자 또는 대소문자로 입력될 수 있습니다. 명령 이름은 키워드의 예입니다.
- 변수는 사용자가 제공하는 이름과 값을 나타내며 소문자로 표시됩니다. 단, 명령 프롬프트에서는 대소문자 제한사항이 명시되지 않는 한, 변수는 대문자, 소문자 또는 대소문자로 입력될 수 있습니다. 파일 이름은 변수의 한 예입니다.

매개변수는 키워드와 변수의 조합이 될 수 있습니다.

필수 매개변수는 주 경로에 표시됩니다.

▶—COMMAND—*required parameter*—▶

선택적 매개변수는 주 경로 아래에 표시됩니다.

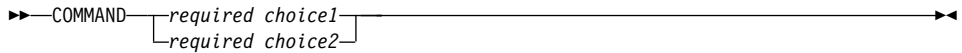
▶—COMMAND—
└ *optional parameter* ┘—▶

매개변수의 기본값은 경로 위에 표시됩니다.

구문 도표를 읽는 방법



주 경로상에 표시된 첫번째 매개변수와 함께, 매개변수 스택은 선택되어야 하는 매개변수 중 하나를 나타냅니다.

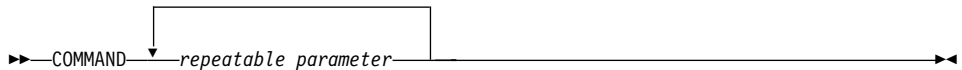


주 경로 아래에 표시된 첫번째 매개변수와 함께, 매개변수 스택은 선택될 수 있는 매개변수 중 하나를 나타냅니다.



왼쪽으로 돌아와서 경로 위에 표시된 화살표는 다음과 같은 규칙에 따라 반복될 수 있는 항목을 나타냅니다.

- 화살표가 인터럽트되지 않을 경우, 항목은 항목이 공백으로 구분되어 있는 목록에서 반복될 수 있습니다.



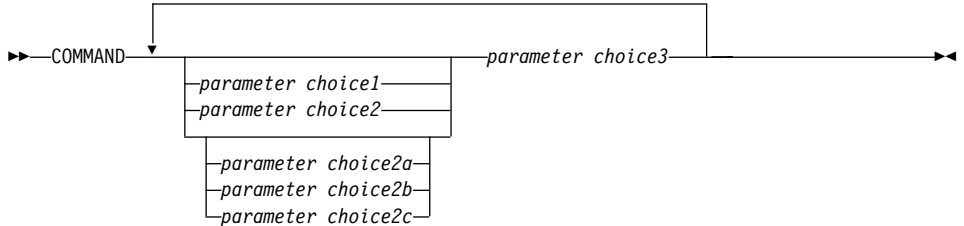
- 화살표에 쉼표가 있으면, 항목은 항목이 쉼표로 구분되어 있는 목록에서 반복될 수 있습니다.



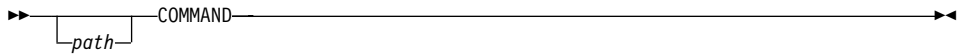
매개변수 스택의 항목은 앞에서 언급된 필수 및 선택적 매개변수에 대한 스택 규칙에 따라 반복될 수 있습니다.

일부 구문 도표는 다른 매개변수 스택 내에 매개변수 스택을 포함합니다. 스택의 항목만 앞에서 언급된 규칙에 따라서 반복될 수 있습니다. 즉, 내부 스택이 위에

반복 화살표를 가지고 있지 않고 외부 스택이 가지고 있다면, 내부 스택에서 하나의 매개변수만 선택하여 외부 스택의 모든 매개변수와 조합될 수 있으며, 이 조합은 반복될 수 있습니다. 예를 들어, 다음 도표는 *choice2a* 매개변수를 *choice2* 매개변수와 조합한 다음, 이 조합(*choice2*와 *choice2a*)을 다시 반복할 수 있다는 것을 나타냅니다.

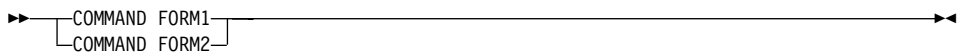


일부 명령은 선택적 경로 매개변수가 앞에 옵니다.



이 매개변수를 제공하지 않으면, 시스템은 현재 디렉토리에서 명령을 검색합니다. 명령을 찾을 수 없을 경우, 시스템은 *.profile*에 나열된 경로상의 모든 디렉토리에서 명령을 계속 검색합니다.

일부 명령은 기능적으로 동등한 구문적 다양성을 가집니다.



구문 도표를 읽는 방법

부록B. 가져오기 및 로드 유틸리티의 차이점

다음 표는 DB2 로드 및 가져오기 유틸리티간의 중요한 차이점을 요약합니다.

가져오기 유틸리티	로드 유틸리티
많은 양의 데이터를 이동시킬 때에는 처리 속도가 느립니다.	많은 양의 데이터를 이동시킬 때에는 로드 유틸리티가 데이터베이스에 형식화된 페이지를 직접 작성하기 때문에 가져오기 유틸리티보다 처리 속도가 빠릅니다.
파티션 내 병렬 처리의 사용이 제한됩니다.	파티션 내 병렬 처리의 사용. 일반적으로, 대칭적 멀티프로세서(SMP) 머신을 필요로 합니다.
FASTPARSE가 지원되지 않습니다.	사용자 제공 데이터의 데이터 점검이 감소되는 FASTPARSE 지원.
CODEPAGE가 지원되지 않습니다.	로드 조작중 이 코드 페이지의 문자 데이터(및 문자에 지정된 숫자 데이터)를 데이터베이스 코드 페이지로 변환하는 CODEPAGE 지원
계층 구조적 데이터를 지원합니다.	계층 구조적 데이터를 지원하지 않습니다.
PC/IXF 형식으로 지원되는 테이블, 계층 구조 및 색인을 작성합니다.	테이블과 색인이 이미 있어야 합니다.
요약 테이블로 가져오기가 지원되지 않습니다.	요약 테이블로 로드를 지원합니다.
WSF 형식이 지원됩니다.	WSF 형식이 지원되지 않습니다.
BINARYNUMERICS가 지원되지 않습니다.	BINARYNUMERICS가 지원됩니다.
PACKEDDECIMAL이 지원되지 않습니다.	PACKEDDECIMAL이 지원됩니다.
ZONEDDECIMAL이 지원되지 않습니다.	ZONEDDECIMAL이 지원됩니다.
GENERATED ALWAYS로 정의된 컬럼을 겹쳐 쓸 수 없습니다.	GENERATEDIGNORE 및 IDENTITYIGNORE 파일 유형 수정자를 사용하여 GENERATED ALWAYS 컬럼을 대체할 수 있습니다.
테이블 및 뷰에 대한 가져오기를 지원합니다.	테이블에 대한 로드만 지원합니다.
테이블과 색인이 상주하는 테이블 공간은 가져오기 조작 동안 온라인 상태입니다.	테이블과 색인이 상주하는 테이블 공간은 로드 조작 동안 오프라인 상태입니다.
모든 행이 로그됩니다.	최소한의 로깅이 수행됩니다.
트리거가 지원됩니다.	트리거가 지원되지 않습니다.

가져오기 및 로드 유틸리티의 차이점

가져오기 유틸리티	로드 유틸리티
가져오기 조작이 인터럽트된 경우 <i>commitcount</i> 가 지정되면, 테이블이 사용 가능하며, 마지막 COMMIT까지 로드된 행을 포함합니다. 사용자는 가져오기 조작을 재시작하거나 테이블을 있는 그대로 사용할 수 있습니다.	로드 조작이 인터럽트된 경우 <i>savecount</i> 가 지정되면, 테이블은 로드 보류 상태로 남고, 로드 조작이 재시작되고 로드 종료 조작이 호출될 때까지, 또는 로드 조작이 시도되기 전에 작성된 백업 이미지로부터 테이블 공간이 복원될 때까지 사용할 수 없습니다.
필요한 공간은 대략 가장 큰 색인에 10%를 더한 크기와 동등합니다. 이 공간은 데이터베이스 내의 임시 테이블 공간으로부터 확보됩니다.	필요한 공간은 대략 테이블에 정의되어 있는 모든 색인의 크기 합계와 동등하며, 이 크기의 두 배 만큼 될 수 있습니다. 이 공간은 데이터베이스 내의 임시 공간으로부터 확보됩니다.
모든 제한조건이 가져오기 조작 동안 유효화됩니다.	고유성은 로드 조작 동안 검증되지만, 다른 모든 제한조건은 SET INTEGRITY문을 사용하여 점검되어야 합니다.
키 값은 가져오기 조작 동안 한 번에 하나씩 색인에 삽입됩니다.	데이터가 로드된 후 키 값이 정렬되고 색인이 빌드됩니다.
갱신된 통계가 필요할 경우, 가져오기 조작 후 <i>runstats</i> 유틸리티가 수행되어야 합니다.	통계는 테이블에 있는 모든 데이터가 바꾸게될 경우, 로드 조작 동안 수집될 수 있습니다.
DB2 Connect를 통해 호스트 데이터베이스로 가져올 수 있습니다.	호스트 데이터베이스로 로드할 수 없습니다.
가져오기 파일은 가져오기 유틸리티가 호출된 노드에 상주해야 합니다.	파티션된 데이터베이스 환경에서 로드 파일 또는 파이프들은 데이터베이스가 들어 있는 노드 상에 상주해야 합니다. 파티션되지 않은 데이터베이스 환경에서 로드 파일 또는 파이프들은 데이터베이스가 들어 있는 노드 상에 또는 로드 유틸리티가 호출된 원격으로 연결된 클라이언트 상에 상주할 수 있습니다.
백업 이미지는 요구되지 않습니다. 가져오기 유틸리티가 SQL 삽입을 사용하고, DB2가 활동을 로그하며, 조작이 실패할 경우, 이 조작을 복구하기 위해 백업이 필요하지 않기 때문입니다.	백업 이미지는 로드 조작중 작성될 수 있습니다.

부록C. 내보내기/가져오기/로드 유틸리티 파일 형식

DB2 내보내기, 가져오기 및 로드 유틸리티가 지원하는 네 가지의 운영 체제 파일 형식은 다음과 같이 기술됩니다.

DEL 다양한 데이터베이스 관리 프로그램 및 파일 관리 프로그램간의 데이터 교환을 위한 컬럼 식별자가 있는 ASCII. 데이터를 저장하기 위한 이러한 일반적인 접근법은 컬럼 값을 구분하기 위해 특수 문자 분리 문자를 사용합니다.

ASC 정렬된 컬럼 데이터를 갖는 플랫폼 텍스트 파일을 작성하는 다른 응용프로그램으로부터 데이터를 가져오거나 로드하기 위한 컬럼 식별자가 없는 ASCII

PC/IXF

통합 교환 형식(IXF)의 PC 버전. 데이터베이스 관리 프로그램 내에서 데이터 교환을 위해 선호되는 방법. PC/IXF는 내부 테이블의 외부적 표현을 포함하는 데이터베이스 테이블의 구조화된 설명입니다.

WSF Lotus 1-2-3 및 Symphony와 같은 제품과 데이터 교환을 위한 워크시트 형식. 로드 유틸리티는 이 파일 형식을 지원하지 않습니다.

DEL, WSF 또는 ASC 데이터 파일 형식을 사용할 때는 파일을 가져오기 전에 테이블 컬럼 이름과 데이터 유형을 포함하여 테이블을 정의하십시오. 운영 체제 파일 필드에 있는 데이터 유형은 데이터베이스 테이블에서 그에 상응하는 데이터의 유형으로 변환됩니다. 가져오기 유틸리티는 채우기 또는 절단 기능을 가지고 가져온 문자 데이터 및 다른 유형의 숫자 필드로 가져온 숫자 데이터를 포함하는 등의 사소한 비호환성 문제점이 있는 데이터를 승인합니다.

PC/IXF 데이터 파일 형식 사용시 테이블은 가져오기 조작을 시작하기 전에 미리 있어야 할 필요는 없습니다. 사용자 정의 구별 유형(UDT)이 새로운 테이블 컬럼 유형의 부분으로 작성되지 않습니다. 대신, 기본 유형이 사용됩니다. 마찬가지로, PC/IXF 데이터 파일 형식으로 내보낼 때 UDT는 PC/IXF 파일에 기본 데이터 유형으로 저장됩니다.

분리 ASCII(DEL) 파일 형식

분리 ASCII(DEL) 파일은 행과 컬럼 분리 문자가 있는 순차적 ASCII 파일입니다. 각 DEL 파일은 행으로 정렬된 후 컬럼순으로 정렬된 셀 값으로 구성된 ASCII 문자열입니다. 데이터 스트림의 행은 행 분리 문자로 분리되고, 각 행에서는 각 셀 값이 컬럼 분리 문자로 분리됩니다.

다음 테이블은 가져올 수 있거나 내보내기 조치의 결과로 생성될 수 있는 DEL 파일의 형식에 대해 설명합니다.

```
DEL 파일 ::= 행 1 데이터 || 행 분리 문자 ||
           행 2 데이터 || 행 분리 문자 ||
           .
           .
           행 n 데이터 || 선택적인 행 분리 문자
```

```
행 i 데이터 ::= 셀 값 (i,1) || 컬럼 분리 문자 ||
               셀 값(i,2) || 컬럼 분리 문자 ||
               .
               .
               셀 값(i,m)
```

행 분리 문자 ::= ASCII 라인 피드 스퀘스^a

컬럼 분리 문자 ::= 기본값 ASCII 쉼표 (,)^b

```
셀 값(i,j) ::= 앞 공백
              || ASCII 값 표시(숫자 값)
              (정수, 십진수 또는 유동 소수점)
              || 분리된 문자열
              || 분리되지 않은 문자열
              || 뒷 공백
```

분리되지 않은 문자열 ::= 행 분리 문자 또는 컬럼 분리 문자를 제외한 다른 문자 세트

```
분리된 문자열 ::= 문자열 분리 문자 ||
                확장된 문자열 ||
                문자열 분리 문자 ||
                트레일링 가베지
```

트레일링 가베지 ::= 행 분리 문자 또는 컬럼 분리 문자를 제외한 다른 문자 세트

문자열 분리 문자 ::= 기본값 ASCII 큰 따옴표
표시 (")^c

확장된 문자열 ::= || NODOUBLEDEL 수정자가 지정되면,
행 분리 문자 또는 문자열
분리 문자를 제외한 모든
문자 세트
|| 문자열이 2개의 연속적인
문자열 분리 문자의 일부가
아니면, 행 분리 문자 또는
문자열 분리 문자를 제외한
모든 문자 세트
|| 문자열 분리 문자가 2개의
연속적인 문자열 분리 문자의
일부가 아니고 DELPRIORITYCHAR
수정자가 지정되면, 문자열
분리 문자를 제외한 모든 문자
세트

파일 끝(EOF) 문자 ::= 16진수 '1A' (OS/2 또는 Windows 운영 체제에서만)

ASCII 값 표시(숫자 값)^d ::= 선택적인 기호 '+' 또는 '-'
|| 숫자 앞, 뒤 또는 그 사이에 선택적인 소수점이
있는 1에서 31 자리 10진수
|| 선택적인 지수

지수 ::= 문자 'E' 또는 'e'
|| 선택적인 기호 '+' 또는 '-'
|| 소수점이 없는 1에서 3 자리 10진수

10진수 ::= 문자 '0', '1', ... '9'

소수점 ::= 기본값 ASCII 점 (.)^e

- ^a 레코드 분리 문자는 새로운 행 문자, ASCII x0A로 가정됩니다. OS/2 또는 Windows 운영 체제에서 생성된 데이터는 0x0D0A의 캐리지 리턴/라인 피드 (CRLF) 2바이트 표준을 사용할 수 있습니다. EBCDIC 코드 페이지의 데이터는 레코드 분리 문자로서 EBCDIC LF 문자(0x25)를 사용해야 합니다(EBCDIC 데이터는 LOAD 명령에서 CODEPAGE 옵션을 사용하여 로드될 수 있습니다).
- ^b 컬럼 분리 문자는 COLDEL 옵션으로 지정될 수 있습니다.
- ^c 문자열 분리 문자는 CHARDEL 옵션으로 지정될 수 있습니다.

주: 분리 문자의 기본 우선순위는 다음과 같습니다.

분리 ASCII(DEL) 파일 형식

1. 레코드 분리 문자
2. 문자 분리 문자
3. 컬럼 분리 문자

145 페이지의 표8에 있는 delprioritychar 수정자 설명을 참조하십시오.

- ^d 숫자 값의 ASCII 표현에 지수가 포함되어 있으면, 이는 FLOAT 상수입니다. 소수점이 있지만 지수가 없는 경우, DECIMAL 상수입니다. 소수점과 지수가 없는 경우, INTEGER 상수입니다.
- ^e 소수점 문자는 DECPT 옵션으로 지정될 수 있습니다.

샘플 DEL 파일

다음은 DEL 파일의 예입니다. 각 행은 라인 피드 순서로 종료됩니다(OS/2 또는 Windows 운영 체제에서, 각 행은 캐리지 리턴/라인 피드(CRLF) 순서로 종료됩니다).

```
"Smith, Bob",4973,15.46  
"Jones, Bill",12345,16.34  
"Williams, Sam",452,193.78
```

다음 예는 비분리 문자열의 사용을 설명합니다. 문자 데이터에 쉼표가 들어 있기 때문에 컬럼 분리 문자가 세미콜론으로 변경되었습니다.

```
Smith, Bob;4973;15.46  
Jones, Bill;12345;16.34  
Williams, Sam;452;193.78
```

주:

1. 공간(X'20')은 결코 유효한 분리 문자가 아닙니다.
2. 첫번째 문자 앞에 오거나 셀 값의 마지막 문자 뒤에 오는 공간을 가져오는 동안 버립니다. 셀 값에 삽입된 공간은 버리지 않습니다.
3. 마침표(.)는 시간소인 값에 있는 마침표와 상충되기 때문에 유효한 문자열 분리 문자가 되지 않습니다.
4. 순수 DBCS(그래픽), 혼합 DBCS 및 EUC의 경우, 분리 문자는 x00부터 x3F까지의 범위로 제한됩니다.

5. EBCDIC 코드 페이지에 지정된 DEL 데이터의 경우, 분리 문자는 쉬프트인 및 쉬프트아웃 DBCS 문자와 일치하지 않을 수 있습니다.
6. OS/2 또는 Windows 운영 체제에서, 문자 분리 문자 안에 포함되지 않은 파일 끝 문자(X'1A')의 처음 발생은 파일의 끝을 나타냅니다. 그 다음에 있는 모든 데이터는 가져오기되지 않습니다.
7. 널(NULL) 값은 일반적으로 생길 수 있는 셀 값의 부재 또는 공간 문자열에 의해 지정됩니다.
8. 일부 제품이 문자 필드를 254 또는 255바이트로 제한하기 때문에 254바이트를 초과하는 최대 길이의 문자 컬럼이 내보내도록 선택될 때마다 내보내기 유틸리티가 경고 메시지를 생성합니다. 가져오기 유틸리티는 가장 긴 LONG CHAR 및 LONG VARCHAR 컬럼만큼 긴 필드를 수용합니다.

DEL 데이터 유형 설명

표 9. DEL 파일 형식에 사용할 수 있는 데이터 유형 형식

데이터 유형	내보내기 유틸리티에 의해 생성된 파일에 있는 형식	가져오기 유틸리티에 사용 가능한 형식
BIGINT	-9 223 372 036 854 775 808에서 9 223 372 036 854 775 807까지 범위에 있는 INTEGER 상수.	-9 223 372 036 854 775 808에서 9 223 372 036 854 775 807까지 범위에 있는 숫자 값의 ASCII 표현. 십진수 및 부동 소수는 정수 값으로 절단됩니다.
BLOB, CLOB	문자 분리자(예: 큰 따옴표)로 둘러싸인 문자 데이터	분리 또는 비분리 문자열. 이 문자열은 데이터베이스 컬럼 값으로 사용됩니다.
BLOB_FILE, CLOB_FILE	각 BLOB/CLOB 컬럼에 대한 문자 데이터는 각 파일에 저장되고, 파일 이름은 문자 분리 문자로 둘러싸입니다.	데이터를 보유하는 파일의 분리 또는 비분리 파일 이름
CHAR	문자 분리자(예: 큰 따옴표)로 둘러싸인 문자 데이터	분리 또는 비분리 문자열. 문자열은 필요할 경우, 데이터베이스 컬럼의 폭과 일치시키기 위해 절단되거나 공간(X'20')으로 채워집니다.

분리 ASCII(DEL) 파일 형식

표 9. DEL 파일 형식에 사용할 수 있는 데이터 유형 형식 (계속)

데이터 유형	내보내기 유틸리티에 의해 생성된 파일에 있는 형식	가져오기 유틸리티에 사용 가능한 형식
DATE	문자 분리 문자가 없는 <code>yyyymmdd</code> (년 월 일). 예: 19931029 또는 모든 날짜 값을 ISO 형식으로 내보내도록 지정하기 위해 <code>DATESISO</code> 옵션을 사용할 수 있습니다.	목표 데이터베이스의 국가 코드와 일치하는 ISO 형식으로 된 날짜 값을 포함하는 분리 또는 비분리 문자열, 또는 <code>yyyymmdd</code> 형식의 비분리 문자열
DBCLOB(DBCS 전용)	그래픽 데이터는 분리된 문자열로 내보내집니다.	작은 바이트 길이의 분리 또는 비분리 문자열. 이 문자열은 데이터베이스 컬럼 값으로 사용됩니다.
DBCLOB_FILE(DBCS 전용)	각 DBCLOB 컬럼에 대한 문자 데이터는 각 파일에 저장되고, 파일 이름은 문자 분리자로 둘러싸입니다.	데이터를 보유하는 파일의 분리 또는 비분리 파일 이름
DECIMAL	내보내는 필드의 정밀도와 스케일이 있는 DECIMAL 상수. <code>DECPLUSBLANK</code> 옵션은 양수(십진수) 값 앞에 더하기 부호(+) 대신 공백 공간을 두도록 지정하는 데 사용될 수 있습니다.	파일이 가져오는 데이터베이스 컬럼의 범위를 넘지 않는 숫자 값의 ASCII 표현. 입력 값에 데이터베이스 컬럼이 수용할 수 있는 것보다 더 많은 숫자가 소수점 뒤에 있으면, 초과된 숫자는 절단됩니다.
FLOAT(long)	-10E307에서 10E307까지 범위에 있는 FLOAT 상수	-10E307에서 10E307까지 범위에 있는 숫자 값의 ASCII 표현
GRAPHIC(DBCS 전용)	그래픽 데이터는 분리된 문자열로 내보내집니다.	작은 바이트 길이의 분리 또는 비분리 문자열. 문자열은 데이터베이스 컬럼의 폭과 일치시키기 위해 필요할 경우, 절단되거나 공간('X'8140')으로 채워집니다.
INTEGER	-2 147 483 648부터 2 147 483 647까지 범위에 있는 INTEGER 상수	-2 147 483 648부터 2 147 483 647까지 범위에 있는 숫자 값의 ASCII 표현. 십진수 및 부동 소수는 정수 값으로 절단됩니다.

표 9. DEL 파일 형식에 사용할 수 있는 데이터 유형 형식 (계속)

데이터 유형	내보내기 유틸리티에 의해 생성된 파일에 있는 형식	가져오기 유틸리티에 사용 가능한 형식
LONG VARCHAR	문자 분리자(예: 큰 따옴표)로 둘러싸인 문자 데이터	분리 또는 비분리 문자열. 이 문자열은 데이터베이스 컬럼 값으로 사용됩니다.
LONG VARGRAPHIC(DBCS 전용)	그래픽 데이터는 분리된 문자열로 내보내집니다.	짝수 바이트 길이의 분리 또는 비분리 문자열. 이 문자열은 데이터베이스 컬럼 값으로 사용됩니다.
SMALLINT	-32 768에서 32 767까지 범위에 있는 INTEGER 상수	-32 768에서 32 767까지 범위에 있는 숫자 값의 ASCII 표현. 십진수 및 부동 소수는 정수 값으로 절단됩니다.
TIME	hh.mm.ss(시간 분 초). 문자 분리자로 둘러싸인 ISO 형식의 시간 값. 예: "09.39.43"	목표 데이터베이스의 국가 코드와 일치하는 형식으로 된 시간 값을 포함하는 분리 또는 비분리 문자열
TIMESTAMP	yyyy-mm-dd-hh.mm.ss.nnnnnn(년 월 일 시간 분 초 마이크로초). 문자 분리자로 둘러싸인 날짜와 시간을 표시하는 문자열	데이터베이스의 저장영역에 대해 사용 가능한 시간소인 값을 포함하는 분리 또는 비분리 문자열
VARCHAR	문자 분리자(예: 큰 따옴표)로 둘러싸인 문자 데이터	분리 또는 비분리 문자열. 필요할 경우, 데이터베이스 컬럼의 최대 폭과 일치시키기 위해 문자열이 절단됩니다.
VARGRAPHIC(DBCS 전용)	그래픽 데이터는 분리된 문자열로 내보내집니다.	짝수 바이트 길이의 분리 또는 비분리 문자열. 필요할 경우, 데이터베이스 컬럼의 최대 폭과 일치시키기 위해 문자열이 절단됩니다.

컬럼 식별자가 없는 ASCII(ASC) 파일 형식

컬럼 식별자가 없는 ASCII(ASC) 파일은 행 분리 문자가 있는 순차적 ASCII 파일입니다. 워드 프로세서를 포함하여 데이터에 대한 컬럼 형식을 갖는 모든 ASCII 제품과 데이터 교환을 위해 사용할 수 있습니다. 각 ASC 파일은 행과 컬럼으로 정렬된 데이터 값으로 구성된 ASCII 문자열입니다. 데이터 스트림의 행은 행 분리 문자로 분리됩니다. 행에 있는 각 컬럼은 시작-끝 위치 쌍(IMPORT 매개변수로 지정된)에 의해 정의됩니다. 각 쌍은 바이트 위치(position)로 지정된 행의 위치(location)를 나타냅니다. 행의 첫번째 위치는 바이트 위치 1입니다. 각 위치 쌍의 첫번째 요소는 컬럼이 시작하는 바이트이며, 각 위치 쌍의 두번째 요소는 컬럼이 끝나는 바이트입니다. 컬럼은 부분적으로 겹쳐질 수 있습니다. ASC 파일에 있는 모든 행은 동일한 컬럼 정의를 가집니다.

ASC 파일은 다음과 같이 정의됩니다.

```
ASC 파일 ::= 행 1 데이터 || 행 분리 문자 ||
           행 2 데이터 || 행 분리 문자 ||
           .
           .
           .
           행 n 데이터
```

행 i 데이터 ::= ASCII 문자 || 행 분리 문자

행 분리 문자 ::= ASCII 라인 피드 시퀀스^a

- ^a 레코드 분리 문자는 새로운 행 문자, ASCII x0A로 가정됩니다. OS/2 또는 Windows 운영 체제에서 생성된 데이터는 0x0D0A의 캐리지 리턴/라인 피드(CRLF) 2바이트 표준을 사용할 수 있습니다. EBCDIC 코드 페이지의 데이터는 레코드 분리 문자로서 EBCDIC LF 문자(0x25)를 사용해야 합니다(EBCDIC 데이터는 LOAD 명령에서 CODEPAGE 옵션을 사용하여 로드될 수 있습니다). 레코드 분리 문자는 결코 데이터 필드의 부분으로 해석되지 않습니다.

샘플 ASC 파일

다음은 ASC 파일의 예입니다. 각 행은 라인 피드 순서로 종료됩니다(OS/2 또는 Windows 운영 체제에서, 각 행은 캐리지 리턴/라인 피드(CRLF) 순서로 종료됩니다).

Smith, Bob	4973	15.46
Jones, Suzanne	12345	16.34
Williams, Sam	452123	193.78

주:

1. ASC 파일은 컬럼 이름을 포함하지 않는 것으로 가정됩니다.
2. 문자열은 분리 문자로 둘러싸여 있지 않습니다. ASC 파일에 있는 컬럼의 데이터 유형은 데이터베이스 테이블에 있는 목표 컬럼의 데이터 유형에 의해 결정됩니다.
3. 널(NULL)은 다음과 같은 경우에 널(NULL) 입력 가능 데이터베이스 컬럼에 가져오기됩니다.
 - 숫자, 날짜, 시간 또는 시간소인 데이터베이스 컬럼에 대해 공백 필드가 목표 지정된 경우
 - 시작 및 끝 위치 쌍이 지정되지 않은 필드
 - 시작 및 끝 위치 쌍 = 0이 지정된 위치 쌍
 - 데이터의 행이 너무 짧아서 목표 컬럼에 대한 유효 값을 포함할 수 없는 경우
 - NULL INDICATORS 로드 옵션이 사용되고, 널(NULL) 표시기 컬럼에 N(또는 사용자가 지정한 다른 값)이 있는 경우
4. 목표 컬럼이 널(NULL) 입력 불가능 컬럼일 경우, 숫자, 날짜, 시간 또는 시간소인 컬럼에 공백 필드를 가져오면 행이 거부됩니다.
5. 입력 데이터가 목표 컬럼과 호환되지 않고 컬럼이 널(NULL) 입력 가능이면, 오류가 발견된 위치에 따라 널(NULL)을 가져오거나 행이 거부됩니다. 컬럼이 널(NULL) 입력 불가능이면 행이 거부됩니다. 메시지가 메시지 파일에 작성되며, 검출된 비호환성을 지정합니다.

ASC 데이터 유형 설명

표 10. ASC 파일 형식에 사용할 수 있는 데이터 유형 형식

데이터 유형	가져오기 유틸리티에 사용 가능한 형식
BIGINT	<p>모든 숫자 유형(SMALLINT, INTEGER, BIGINT, DECIMAL 또는 FLOAT)에 있는 상수는 사용 가능합니다. 각 값은 -9 223 372 036 854 775 808에서 9 223 372 036 854 775 807까지 범위에 있지 않을 경우 거부됩니다. 십진수는 정수 값으로 절단됩니다. 쉼표, 마침표, 콜론은 소수점으로 인식됩니다. 방대한 분리 문자는 허용되지 않습니다.</p> <p>시작 및 끝 위치는 폭이 50바이트를 초과하지 않는 필드를 지정해야 합니다. 정수, 십진수 및 부동 소수의 가수는 31자릿수를 초과할 수 없습니다. 부동 소수의 지수는 3자릿수를 초과할 수 없습니다.</p>
BLOB/CLOB	<p>문자열. 이 문자열은 목표 컬럼의 최대 길이와 일치시키기 위해 필요할 경우, 오른쪽에서 절단됩니다. ASC 공백 절단 옵션이 실행되면, 뒤 공백은 원래 문자열이나 절단된 문자열로부터 제거됩니다.</p>
BLOB_FILE, CLOB_FILE, DBCLOB_FILE(DBCS 전용)	<p>데이터를 보유하는 파일의 분리 또는 비분리 파일 이름</p>
CHAR	<p>문자열. 이 문자열은 목표 컬럼의 폭과 일치시키기 위해 필요할 경우, 오른쪽이 절단되거나 공간으로 채워집니다.</p>
DATE	<p>목표 데이터베이스의 국가 코드와 일치하는 형식으로 된 날짜 값을 표시하는 문자열</p> <p>시작 및 끝 위치는 날짜의 외부적 표현을 위한 범위에 해당하는 필드 폭을 지정해야 합니다.</p>
DBCLOB(DBCS 전용)	<p>작은 바이트의 문자열. 홀수 바이트 문자열은 유효하지 않으며 사용되지 않습니다. 유효한 문자열은 목표 컬럼의 최대 길이와 일치시키기 위해 필요할 경우, 오른쪽이 절단됩니다.</p>
DECIMAL	<p>모든 숫자 유형(SMALLINT, INTEGER, BIGINT, DECIMAL 또는 FLOAT)에 있는 상수는 사용 가능합니다. 각 값은 가져오는 데이터베이스 컬럼 범위에 있지 않을 경우 거부됩니다. 입력 값에 데이터베이스 컬럼의 스케일보다 더 많은 숫자가 소수점 뒤에 들어 있으면, 초과된 숫자는 절단됩니다. 쉼표, 마침표, 콜론은 소수점으로 인식됩니다. 방대한 분리 문자는 허용되지 않습니다.</p> <p>시작 및 끝 위치는 폭이 50바이트를 초과하지 않는 필드를 지정해야 합니다. 정수, 십진수 및 부동 소수의 가수는 31자릿수를 초과할 수 없습니다. 부동 소수의 지수는 3자릿수를 초과할 수 없습니다.</p>

표 10. ASC 파일 형식에 사용할 수 있는 데이터 유형 형식 (계속)

데이터 유형	가져오기 유틸리티에 사용 가능한 형식
FLOAT(long)	<p>모든 숫자 유형(SMALLINT, INTEGER, BIGINT, DECIMAL 또는 FLOAT)에 있는 상수는 사용 가능합니다. 모든 값은 유효합니다. 쉼표, 마침표, 콜론은 소수점으로 인식됩니다. 대문자 또는 소문자 E는 FLOAT 상수의 지수 부분의 시작으로 인식됩니다.</p> <p>시작 및 끝 위치는 폭이 50바이트를 초과하지 않는 필드를 지정해야 합니다. 정수, 십진수 및 부동 소수의 가수는 31자릿수를 초과할 수 없습니다. 부동 소수의 지수는 3자릿수를 초과할 수 없습니다.</p>
GRAPHIC(DBCS 전용)	<p>짝수 바이트의 문자열. 홀수 바이트 문자열은 유효하지 않으며 사용되지 않습니다. 유효 문자열은 목표 컬럼의 최대 길이와 일치시키기 위해 필요할 경우, 오른쪽이 절단되거나 2바이트 공간(0x8140)으로 채워집니다.</p>
INTEGER	<p>모든 숫자 유형(SMALLINT, INTEGER, BIGINT, DECIMAL 또는 FLOAT)에 있는 상수는 사용 가능합니다. 각 값은 -2 147 483 648에서 2 147 483 647까지 범위에 있지 않을 경우 거부됩니다. 십진수는 정수 값으로 절단됩니다. 쉼표, 마침표, 콜론은 소수점으로 인식됩니다. 방대한 분리 문자는 허용되지 않습니다.</p> <p>시작 및 끝 위치는 폭이 50바이트를 초과하지 않는 필드를 지정해야 합니다. 정수, 십진수 및 부동 소수의 가수는 31자릿수를 초과할 수 없습니다. 부동 소수의 지수는 3자릿수를 초과할 수 없습니다.</p>
LONG VARCHAR	<p>문자열. 이 문자열은 목표 컬럼의 최대 길이와 일치시키기 위해 필요할 경우, 오른쪽에서 절단됩니다. ASC 공백 절단 옵션이 실행되면, 뒤 공백은 원래 문자열이나 절단된 문자열로부터 제거됩니다.</p>
LONG VARGRAPHIC (DBCS 전용)	<p>짝수 바이트의 문자열. 홀수 바이트 문자열은 유효하지 않으며 사용되지 않습니다. 유효한 문자열은 목표 컬럼의 최대 길이와 일치시키기 위해 필요할 경우, 오른쪽이 절단됩니다.</p>
SMALLINT	<p>모든 숫자 유형(SMALLINT, INTEGER, BIGINT, DECIMAL 또는 FLOAT)에 있는 상수는 사용 가능합니다. 각 값은 -32 768에서 32 767까지 범위에 있지 않을 경우 거부됩니다. 십진수는 정수 값으로 절단됩니다. 쉼표, 마침표, 콜론은 소수점으로 인식됩니다. 방대한 분리 문자는 허용되지 않습니다.</p> <p>시작 및 끝 위치는 폭이 50바이트를 초과하지 않는 필드를 지정해야 합니다. 정수, 십진수 및 부동 소수의 가수는 31자릿수를 초과할 수 없습니다. 부동 소수의 지수는 3자릿수를 초과할 수 없습니다.</p>

컬럼 식별자가 없는 ASCII(ASC) 파일 형식

표 10. ASC 파일 형식에 사용할 수 있는 데이터 유형 형식 (계속)

데이터 유형	가져오기 유틸리티에 사용 가능한 형식
TIME	목표 데이터베이스의 국가 코드와 일치하는 형식으로 된 시간 값을 표시하는 문자열 시작 및 끝 위치는 시간의 외부적 표현을 위한 범위에 해당하는 필드 폭을 지정해야 합니다.
TIMESTAMP	데이터베이스의 저장영역에 대해 사용 가능한 시간소인 값을 나타내는 문자열 시작 및 끝 위치는 시간소인의 외부 표현을 위한 범위에 해당하는 필드 폭을 지정해야 합니다.
VARCHAR	문자열. 이 문자열은 목표 컬럼의 최대 길이와 일치시키기 위해 필요할 경우, 오른쪽에서 절단됩니다. ASC 공백 절단 옵션이 실행되면, 뒤 공백은 원래 문자열이나 절단된 문자열로부터 제거됩니다.
VARGRAPHIC(DBCS 전용)	작수 바이트의 문자열. 홀수 바이트 문자열은 유효하지 않으며 사용되지 않습니다. 유효한 문자열은 목표 컬럼의 최대 길이와 일치시키기 위해 필요할 경우, 오른쪽이 절단됩니다.

IXF 파일 형식의 PC 버전

IXF(PC/IXF) 파일 형식의 PC 버전은 통합 교환 형식(IXF) 데이터 상호교환 아키텍처의 데이터베이스 관리 프로그램에 적합합니다. IXF 아키텍처는 관계형 데이터베이스 구조와 데이터의 교환이 작동 가능하도록 설계되어 있습니다. PC/IXF 아키텍처는 데이터베이스 관리 프로그램이 수신 제품의 요구사항 및 특이성을 염려하지 않고 데이터베이스를 내보낼 수 있게 합니다. 마찬가지로, PC/IXF 파일을 가져오는 제품은 단지 PC/IXF 아키텍처만 이해하면 됩니다. 파일을 가져오기한 제품의 특성은 관련되지 않습니다. PC/IXF 파일 아키텍처는 내보내기 및 가져오기 데이터베이스 시스템의 독립성을 유지보수합니다.

IXF 아키텍처는 특정 관계형 데이터베이스 제품에 의해 지원되지 않을 수 있는 몇 가지 유형을 포함하여, 여러 관계형 데이터 유형 세트를 지원하는 일반적 관계형 데이터베이스 교환 형식입니다. PC/IXF 파일 형식은 이러한 유연성을 보존합니다. 예를 들어, PC/IXF 아키텍처는 1바이트 문자열(SBCS)과 2바이트 문자열(DBCS)

데이터 유형을 모두 지원합니다. 모든 구현이 모든 PC/IXF 데이터 유형을 지원하지는 않습니다. 단, 제한된 구현이 가져오기중 지원되지 않은 데이터 유형의 검출 및 제거를 지원합니다.

일반적으로, PC/IXF 파일은 가변 길이 레코드의 중단되지 않은 배열로 구성됩니다. 이 파일에는 다음 순서로 된 레코드 유형이 들어 있습니다.

- 레코드 유형 H의 하나의 헤더 레코드
- 레코드 유형 T의 하나의 테이블 레코드
- 레코드 유형 C의 다중 컬럼 설명자 레코드(테이블의 각 컬럼에 대해 하나의 레코드)
- 레코드 유형 D의 다중 데이터 레코드(테이블의 각 행은 하나 이상의 D 레코드로 표시됨)

PC/IXF 파일은 H 레코드 다음에 어느 곳이나 레코드 유형 A의 응용프로그램 레코드도 포함할 수 있습니다. 이들 레코드는 PC/IXF 파일에 포함될 수 있으며, 응용프로그램이 PC/IXF 형식에 의해 정의되지 않은 추가 데이터를 PC/IXF 파일에 포함시킬 수 있도록 합니다. A 레코드는 A 레코드에 있는 응용프로그램 식별자가 의미하는 데이터 형식 및 내용에 대해 특정 지식을 갖지 않는 PC/IXF 파일을 읽는 모든 프로그램에서 무시됩니다.

PC/IXF 파일의 모든 레코드는 레코드 길이 식별자를 사용하여 시작합니다. 이것은 레코드 길이 표시기 뒤에 오는 PC/IXF 레코드 부분의 길이(바이트)를 나타내는 정수 값의 6바이트 오른쪽으로 정렬된 문자 표현입니다. 즉, 전체 레코드 크기에서 6바이트를 뺍니다. PC/IXF 파일을 읽는 프로그램은 이 레코드 길이를 사용하여 현재 레코드의 끝과 다음 레코드의 시작을 위치시켜야 합니다. H, T 및 C 레코드는 정의된 모든 필드를 포함할 수 있을 만큼 충분히 커야 하며, 레코드 길이 필드도 물론 실제 길이와 맞아야 합니다. 단, 추가 데이터(예: 새로운 필드)가 이 중 하나의 레코드의 끝에 추가될 경우, PC/IXF 파일을 읽는 기존 프로그램은 추가 데이터를 무시해야 하며 경고 메시지 이상은 생성하지 않습니다. 그러나 PC/IXF 파일을 작성하는 프로그램은 정의된 모든 필드를 포함하기 위해 정확한 길이가 필요한 H, T, C 레코드를 작성해야 합니다.

IXF 파일 형식의 PC 버전

PC/IXF 파일 레코드는 문자 데이터를 포함하는 필드로 구성됩니다. 가져오기 및 내보내기 유틸리티는 목표 데이터베이스의 CPGID를 사용하여 이 문자 데이터를 해석합니다. 여기에는 다음 두 가지 예외가 있습니다.

- A 레코드의 IXFADATA 필드

IXFADATA 필드에 들어 있는 문자 데이터의 코드 페이지 환경은 특정 A 레코드를 작성하고 처리하는 응용프로그램에 의해 설정됩니다. 즉, 이 환경은 구현에 따라 다릅니다.

- D 레코드의 IXFDCOLS 필드

IXFDCOLS 필드에 들어 있는 문자 데이터의 코드 페이지 환경은 특정 컬럼과 데이터를 정의하는 C 레코드에 있는 정보의 함수입니다.

H, T, C 레코드 및 D와 A 레코드의 접두부에 있는 숫자 필드는 정수 값의 오른쪽으로 정렬된 1바이트 문자 표현이어야 하며, 앞에 0 또는 공백으로 채워집니다. 0 값은 공백이 아닌, 최소한 하나의(오른쪽 정렬) 0 문자로 표시되어야 합니다. 데이터 유형에 의해 길이가 암시된 필드에서 이 숫자 필드 중 하나가 사용되지 않으면(예: IXFCLENG), 이 필드는 공백으로 채워져야 합니다. 이들 숫자 필드는 다음과 같습니다.

IXFHRECL, IXFTRECL, IXFCRECL, IXFDRECL, IXFARECL,
IXFHHCNT, IXFHSBCP, IXFHDBCP, IXFTCCNT, IXFTNAML,
IXFCLENG, IXFCDRID, IXFCPOSN, IXFCNAML, IXFCTYPE,
IXFCSBCP, IXFCDBCP, IXFCNDIM, IXFCDSIZ, IXFDRID

주: 데이터베이스 관리 프로그램 PC/IXF 파일 형식은 System/370 IXF 형식과 일치하지 않습니다(303 페이지의 『PC/IXF와 버전 0 System/370 IXF간의 차이점』 참조).

PC/IXF 레코드 유형

5개의 기본 PC/IXF 레코드 유형이 있습니다.

- 헤더
- 테이블
- 컬럼 설명자
- 데이터
- 응용프로그램

그리고 DB2 UDB가 사용하는 6개의 응용프로그램 부속 유형이 있습니다.

- 색인
- 계층
- 서브테이블
- 계속
- 종료
- 식별

각 PC/IXF 레코드 유형은 필드의 배열로 정의되며, 이들 필드는 필수적이며 표시된 순서로 나타나야 합니다.

헤더 레코드

필드 이름	길이	유형	비고
IXFHRECL	06-BYTE	CHARACTER	레코드 길이
IXFHRECT	01-BYTE	CHARACTER	레코드 유형 = 'H'
IXFHID	03-BYTE	CHARACTER	IXF 식별자
IXFHVERS	04-BYTE	CHARACTER	IXF 버전
IXFHPROD	12-BYTE	CHARACTER	제품
IXFHDATE	08-BYTE	CHARACTER	작성 날짜
IXFHTIME	06-BYTE	CHARACTER	작성 시간
IXFHHCNT	05-BYTE	CHARACTER	헤딩 레코드 카운트
IXFHSBCP	05-BYTE	CHARACTER	1바이트 코드 페이지
IXFHDBC	05-BYTE	CHARACTER	2바이트 코드 페이지
IXHFHIL1	02-BYTE	CHARACTER	예약됨

헤더 레코드에는 다음 필드가 들어 있습니다.

IXFHRECL

레코드 길이 표시기. 레코드 길이 표시기 뒤에 오는 PC/IXF 레코드 부분의 길이(바이트)를 나타내는 정수 값의 6바이트 문자 표현입니다. 즉 전체 레코드 크기에서 6바이트를 뺍니다. H 레코드는 정의된 모든 필드를 포함할 수 있을 만큼 충분히 길어야 합니다.

IXFHRECT

IXF 레코드 유형. 이 레코드에 대해 H로 설정됩니다.

IXFHID

파일 형식 식별자. 이 파일에 대해 IXF로 설정됩니다.

IXF 파일 형식의 PC 버전

IXFHVERS

파일 작성시 사용된 PC/IXF 형식 레벨. '0002'로 설정됩니다.

IXFHPROD

자신을 식별하기 위해 파일을 작성하는 프로그램이 사용할 수 있는 필드. 이 필드가 채워지면, 처음 6바이트는 파일을 작성하는 제품을 식별하는 데 사용되고, 마지막 6바이트는 작성중인 제품의 버전이나 릴리스를 식별하는 데 사용됩니다. 데이터베이스 관리 프로그램은 이 필드를 사용하여 데이터베이스 관리 프로그램 특정 데이터가 있는지를 신호합니다.

IXFHDATE

yyyymmdd 형식으로 된 파일 작성 날짜.

IXFHTIME

hhmmss 형식으로 된 파일 작성 시간. 이 필드는 선택적이며 공백으로 남을 수 있습니다.

IXFHHCNT

첫번째 데이터 레코드 앞에 오는 이 파일에 있는 H, T, C 레코드의 수. A 레코드는 이 수에 포함되지 않습니다.

IXFHSBCP

1바이트 코드 페이지 필드. SBCS CPGID 또는 '00000'의 1바이트 문자 표시를 포함합니다.

내보내기 유틸리티는 이 필드를 내보낸 데이터베이스 테이블의 SBCS CPGID와 동일하게 설정합니다. 예를 들어, 테이블 SBCS CPGID가 850 일 경우에는 이 필드에 '00850'이 포함됩니다.

IXFHDBCP

2바이트 코드 페이지 필드. DBCS CPGID 또는 '00000'의 1바이트 문자 표시를 포함합니다.

내보내기 유틸리티는 이 필드를 내보낸 데이터베이스 테이블의 DBCS CPGID와 동일하게 설정합니다. 예를 들어, 테이블 DBCS CPGID가 301 일 경우에는 이 필드에 '00301'이 포함됩니다.

IXFHFIL1

호스트 IXF 파일에 있는 예약 필드와 일치시키기 위해 두 자리 공백으로 설정된 여분 필드.

테이블 레코드

필드 이름	길이	유형	비고
-----	-----	-----	-----
IXFTRECL	006-BYTE	CHARACTER	레코드 길이
IXFTRECT	001-BYTE	CHARACTER	레코드 유형 = 'T'
IXFTNAML	003-BYTE	CHARACTER	이름 길이
IXFTNAME	256-BYTE	CHARACTER	데이터 이름
IXFTQULL	003-BYTE	CHARACTER	규정자 길이
IXFTQUAL	256-BYTE	CHARACTER	규정자
IXFTSRC	012-BYTE	CHARACTER	데이터 소스
IXFTDATA	001-BYTE	CHARACTER	데이터 규정 = 'C'
IXFTFORM	001-BYTE	CHARACTER	데이터 형식 = 'M'
IXFTFRM	005-BYTE	CHARACTER	머신 형식 = 'PC'
IXFTLOC	001-BYTE	CHARACTER	데이터 위치 = 'I'
IXFTCNT	005-BYTE	CHARACTER	'C' 레코드 계수
IXFTFIL1	002-BYTE	CHARACTER	예약됨
IXFTDESC	030-BYTE	CHARACTER	데이터 설명
IXFTPKNM	257-BYTE	CHARACTER	기본 키 이름
IXFTDSPC	257-BYTE	CHARACTER	예약됨
IXFTISPC	257-BYTE	CHARACTER	예약됨
IXFTLSPC	257-BYTE	CHARACTER	예약됨

테이블 레코드에는 다음 필드가 들어 있습니다.

IXFTRECL

레코드 길이 표시기. 레코드 길이 표시기 뒤에 오는 PC/IXF 레코드 부분의 길이(바이트)를 나타내는 정수 값의 6바이트 문자 표현입니다. 즉 전체 레코드 크기에서 6바이트를 뺍니다. T 레코드는 정의된 모든 필드를 포함할 수 있을 만큼 충분히 길어야 합니다.

IXFTRECT

IXF 레코드 유형. 이 레코드에 대해 T로 설정됩니다.

IXFTNAML

IXFTNAME 필드에 있는 테이블 이름의 길이(바이트).

IXFTNAME

테이블 이름. 각 파일에 한 개의 테이블만 있으면, 이는 단지 정보용 필드

IXF 파일 형식의 PC 버전

입니다. 데이터베이스 관리 프로그램은 데이터를 가져올 때 이 필드를 사용하지 않습니다. PC/IXF 파일을 작성할 때, 데이터베이스 관리 프로그램은 이 필드에 DOS 파일 이름(및 경로 정보)을 기록합니다.

IXFTQULL

IXFTQUAL 필드에 있는 테이블 이름 규정자의 길이(바이트).

IXFTQUAL

테이블 이름 규정자. 관계형 시스템에 있는 테이블의 작성자를 식별합니다. 이는 단지 정보용 필드입니다. 파일을 작성하는 프로그램에 이 필드에 기록할 데이터가 없을 경우, 공백 값이 우선적으로 채워집니다. 파일을 읽는 프로그램이 이 필드를 인쇄 또는 표시하거나 정보용 필드에 저장할 수 있습니다. 단, 연산은 이 필드의 내용에 따라야 합니다.

IXFTSRC

데이터의 원래 소스를 지정하는 데 사용됩니다. 이는 단지 정보용 필드입니다. 파일을 작성하는 프로그램에 이 필드에 기록할 데이터가 없을 경우, 공백 값이 우선적으로 채워집니다. 파일을 읽는 프로그램이 이 필드를 인쇄 또는 표시하거나 정보용 필드에 저장할 수 있습니다. 단, 연산은 이 필드의 내용에 따라야 합니다.

IXFTDATA

데이터를 기술하는 데 사용되는 규정. 이 필드는 가져오기 및 내보내기에 대해 C로 설정되어야 합니다. 이는 각 컬럼 속성이 다음 컬럼 설명자(C) 레코드에 기술되며 데이터가 PC/IXF 규정을 따른다는 것을 나타냅니다.

IXFTFORM

숫자 데이터를 저장하는 데 사용되는 규정. 이 필드는 M으로 설정되어야 하며, 데이터(D) 레코드에 있는 숫자 데이터가 IXFTMFRM 필드에 의해 지정된 머신(내부) 형식으로 저장되는 것을 나타냅니다.

IXFTMFRM

PC/IXF 파일에 있는 모든 머신 데이터의 형식. 데이터베이스 관리 프로그램은 이 필드가 PCbbb로 설정될 경우에만 파일을 읽거나 작성합니다. 여기서 b는 공백을 나타내며, PC는 PC/IXF 파일에 있는 데이터가 IBM PC 머신 형식으로 되어 있다는 것을 지정합니다.

IXFTLOC

데이터의 위치(location). 데이터베이스 관리 프로그램은 I의 값만 지원하며, 데이터가 이 파일에 대해 내부적이라는 것을 의미합니다.

IXFTCCNT

이 테이블에 있는 C 레코드의 수. 정수 값의 오른쪽 정렬 문자 표시입니다.

IXFTFIL1

호스트 IXF 파일에 있는 예약 필드와 일치시키기 위해 두 자리 공백으로 설정된 여분 필드.

IXFTDESC

테이블에 대한 설명적 데이터. 이는 단지 정보용 필드입니다. 파일을 작성하는 프로그램에 이 필드에 기록할 데이터가 없을 경우, 공백 값이 우선적으로 채워집니다. 파일을 읽는 프로그램이 이 필드를 인쇄 또는 표시하거나 정보용 필드에 저장할 수 있습니다. 단, 연산은 이 필드의 내용에 따라야 합니다. 컬럼의 기본값이 널(NULL)이 아니고 테이블 이름이 워크스테이션 데이터베이스에서 주어지지 않을 경우, 이 필드에는 NOT NULL WITH DEFAULT가 들어갑니다.

IXFTPKNM

테이블에 정의된 기본 키의 이름(가능하다면). 이름은 널(NULL) 종료 문자열로 저장됩니다.

IXFTDSPC

이 필드는 향후 사용을 위해 예약되었습니다.

IXFTISPC

이 필드는 향후 사용을 위해 예약되었습니다.

IXFTLSPC

이 필드는 향후 사용을 위해 예약되었습니다.

컬럼 설명자 레코드

필드 이름	길이	유형	비고
-----	-----	-----	-----
IXFCRECL	006-BYTE	CHARACTER	레코드 길이
IXFCRECT	001-BYTE	CHARACTER	레코드 유형 = 'C'

IXF 파일 형식의 PC 버전

	IXFCNAML	003-BYTE	CHARACTER	컬럼 이름 길이
	IXFCNAME	256-BYTE	CHARACTER	컬럼 이름
	IXFCNULL	001-BYTE	CHARACTER	컬럼이 널을 허용함
	IXFCDEF	001-BYTE	CHARACTER	컬럼이 기본값을 지님
	IXFCSLCT	001-BYTE	CHARACTER	컬럼이 플래그를 선택함
	IXFCKPOS	002-BYTE	CHARACTER	기본 키에서의 위치
	IXFCCLAS	001-BYTE	CHARACTER	데이터 클래스
	IXFCTYPE	003-BYTE	CHARACTER	데이터 유형
	IXFCSBCP	005-BYTE	CHARACTER	1바이트 코드 페이지
	IXFCBCP	005-BYTE	CHARACTER	2바이트 코드 페이지
	IXFCLENG	005-BYTE	CHARACTER	컬럼 데이터 길이
	IXFCDRID	003-BYTE	CHARACTER	'D' 레코드 식별자
	IXFCPOSN	006-BYTE	CHARACTER	컬럼 위치
	IXFCDESC	030-BYTE	CHARACTER	컬럼 설명
	IXFCLOBL	020-BYTE	CHARACTER	LOB 컬럼 길이
	IXFCUDTL	003-BYTE	CHARACTER	UDT 이름 길이
	IXFCUDTN	256-BYTE	CHARACTER	UDT 이름
	IXFCDEFL	003-BYTE	CHARACTER	기본 값 길이
	IXFCDEFV	254-BYTE	CHARACTER	기본 값
	IXFCDLPR	010-BYTE	CHARACTER	dataLink 특성
	IXFCREF	001-BYTE	CHARACTER	참조 유형
	IXFCNDIM	002-BYTE	CHARACTER	차원의 수
	IXFCDSIZ	varying	CHARACTER	각 차원의 크기

| 컬럼 설명자 레코드에는 다음 필드가 들어 있습니다.

| IXFCRECL

| 레코드 길이 표시기. 레코드 길이 표시기 뒤에 오는 PC/IXF 레코드 부분
| 의 길이(바이트)를 나타내는 정수 값의 6바이트 문자 표현입니다. 즉 전체
| 레코드 크기에서 6바이트를 뺍니다. C 레코드는 정의된 모든 필드를 포함
| 할 수 있을 만큼 충분한 길이어야 합니다.

| IXFCRECT

| IXF 레코드 유형. 이 레코드에 대해 C로 설정됩니다.

| IXFCNAML

| IXFCNAME 필드에 있는 컬럼 이름의 길이(바이트).

| IXFCNAME

| 컬럼 이름.

| IXFCNULL

| 이 컬럼에 널(NULL)이 허용되는지를 지정합니다. 유효한 설정값은 Y 또
| 는 N입니다.

IXFCDEF

이 필드에 대해 기본값이 정의되는지 지정합니다. 유효한 설정값은 Y 또는 N입니다.

IXFCSLCT

데이터에 있는 컬럼의 부속 집합의 선택을 허용하기 위해 사용되지 않는 필드. PC/IXF 파일을 작성하는 프로그램은 이 필드에 항상 Y를 저장해야 합니다. PC/IXF 파일을 읽는 프로그램은 이 필드를 무시해야 합니다.

IXFCKPOS

기본 키의 한부분으로서 컬럼의 위치(position). 컬럼이 기본 키의 일부가 아닌 경우에는 01에서 16 사이 또는 N 값이 유효합니다.

IXFCCLAS

IXFCTYPE 필드에서 사용될 데이터 유형의 클래스. 데이터베이스 관리 프로그램은 관계형 유형(R)만 지원합니다.

IXFCTYPE

컬럼에 대한 데이터 유형. 275 페이지의 『PC/IXF 데이터 유형』에서 데이터 유형에 대한 자세한 정보를 참조하십시오.

IXFCSBCP

SBCS CPGID의 1바이트 문자 표시가 들어 있습니다. 이 필드는 이 컬럼에 대한 D 레코드의 IXFDCLS 필드에서 발생하는 1바이트 문자 데이터에 대한 CPGID를 지정합니다.

이 필드의 의미는 컬럼에 대한 데이터 유형에 따라 다릅니다(IXFCTYPE 필드에 지정됨).

- 문자열 컬럼의 경우, 이 필드는 일반적으로 H 레코드에 있는 IXFHSBCP 필드 값과 동일한 0 외의 값을 가져야 합니다. 단, 다른 값은 허용됩니다. 이 값이 0일 경우, 컬럼은 비트 문자열 데이터를 포함하는 것으로 해석됩니다.
- 숫자 컬럼의 경우, 이 필드는 의미가 없습니다. 내보내기 유틸리티에 의해 0으로 설정되고 가져오기 유틸리티는 이 필드를 무시합니다.

IXF 파일 형식의 PC 버전

- 날짜 또는 시간 컬럼의 경우, 이 필드는 의미가 없습니다. 내보내기 유틸리티가 이 필드를 IXFHSBCH 필드의 값으로 설정하고 가져오기 유틸리티는 이 필드를 무시합니다.
- 그래픽 컬럼의 경우, 이 필드는 0이 되어야 합니다.

또한 281 페이지의 표12에서도 자세한 내용을 참조하십시오.

IXFCDBCP

DBCS CPGID의 1바이트 문자 표시가 들어 있습니다. 이 필드는 이 컬럼에 대한 D 레코드의 IXFDCOLS 필드에서 발생하는 2바이트 문자 데이터에 대한 CPGID를 지정합니다.

이 필드의 의미는 컬럼에 대한 데이터 유형에 따라 다릅니다(IXFCTYPE 필드에 지정됨).

- 문자열 컬럼의 경우, 이 필드는 H 레코드에 있는 IXFHDBCP 필드의 값과 동일한 값을 갖거나 0이 되어야 합니다. 단, 다른 값은 허용됩니다. IXFCSBCP 필드에 있는 값이 0일 경우, 이 필드의 값도 0이 되어야 합니다.
- 숫자 컬럼의 경우, 이 필드는 의미가 없습니다. 내보내기 유틸리티에 의해 0으로 설정되고 가져오기 유틸리티는 이 필드를 무시합니다.
- 날짜 또는 시간 컬럼의 경우, 이 필드는 의미가 없습니다. 내보내기 유틸리티에 의해 0으로 설정되고 가져오기 유틸리티는 이 필드를 무시합니다.
- 그래픽 컬럼의 경우, 이 필드는 IXFHDBCP 필드의 값과 동일한 값을 가져야 합니다.

또한 281 페이지의 표12에서도 자세한 내용을 참조하십시오.

IXFCLENG

설명중인 컬럼의 크기에 대한 정보를 제공합니다. 일부 데이터 유형의 경우, 이 필드는 사용되지 않으며 공백이 포함되어야 합니다. 다른 데이터 유형의 경우, 이 필드는 컬럼 길이를 지정하는 정수의 오른쪽 정렬 데이터 표시를 포함합니다. 또다른 데이터 유형의 경우, 이 필드는 두 개의 부속 필드 즉, 정밀도를 위한 3바이트와 스케일을 위한 2바이트로 나뉩니다. 이 부속 필드는 모든 오른쪽 정렬 문자 표시의 정수입니다.

IXFCDRID

D 레코드 식별자. 이 필드에는 정수 값의 오른쪽 정렬 문자 표시가 들어 있습니다. PC/IXF 파일에 있는 데이터의 각 행을 포함하기 위해 여러 개의 D 레코드가 사용될 수 있습니다. 이 필드는 어느 D 레코드(데이터의 행에 사용된 여러 D 레코드 중)가 컬럼에 대한 데이터를 포함하는지를 지정합니다. 1 값(예: 001)은 컬럼에 대한 데이터가 데이터 행의 첫번째 D 레코드에 있다는 것을 나타냅니다. 첫번째 C 레코드는 1의 IXFCDRID 값을 가져야 합니다. 다음에 있는 모든 C 레코드는 앞에 있는 C 레코드의 값 또는 더 높은 C 레코드의 값과 동일한 IXFCDRID 값을 가져야 합니다.

IXFCPOSN

이 필드의 값은 테이블 데이터의 행을 나타내는 D 레코드 중 하나에 컬럼에 대한 데이터를 위치시키는 데 사용됩니다. 이는 D 레코드의 IXFDCOLS 필드 내에서의 이 컬럼에 대한 데이터의 시작 위치입니다. 컬럼에 널(NULL)을 입력할 수 있을 경우, IXFCPOSN은 널(NULL) 표시기를 가리키고, 그렇지 않으면 데이터 자체를 가리킵니다. 컬럼에 가변 길이의 데이터가 들어 있으면, 데이터 자체는 현재 길이 표시기에서 시작합니다. D 레코드의 IXFDCOLS 필드에 있는 첫번째 바이트에 대한 IXFCPOSN 값은 1입니다(0이 아님). 컬럼이 새로운 D 레코드에 있을 경우, IXFCPOSN의 값은 1이 되어야 합니다. 그렇지 않으면, IXFCPOSN 값은 컬럼마다 늘어나서 데이터 값이 겹치지 않는 정도까지 되어야 합니다.

IXFCDESC

컬럼에 대한 설명적 정보. 이는 단지 정보용 필드입니다. 파일을 작성하는 프로그램에 이 필드에 기록할 데이터가 없을 경우, 공백 값이 우선적으로 채워집니다. 파일을 읽는 프로그램이 이 필드를 인쇄 또는 표시하거나 정보용 필드에 저장할 수 있습니다. 단, 연산은 이 필드의 내용에 따라야 합니다.

IXFCLOBL

이 컬럼에 정의된 long 또는 LOB의 길이(바이트). 이 컬럼이 long 또는 LOB가 아니면, 이 필드의 값은 000입니다.

IXF 파일 형식의 PC 버전

IXFCUDTL

IXFCUDTN 필드에 있는 UDT(사용자 정의 유형)의 길이(바이트). 이 컬럼의 유형이 UDT가 아니면, 이 필드의 값은 000입니다.

IXFCUDTN

이 컬럼에 대한 데이터 유형으로 사용된 사용자 정의 유형의 이름.

IXFCDEFL

IXFCDEFV 필드에 있는 기본값의 길이(바이트). 이 컬럼에 기본값이 없으면, 이 필드의 값은 000입니다.

IXFCDEFV

이 컬럼에 대해 기본값을 지정합니다(기본값이 정의된 경우).

IXFCDLPR

컬럼이 DATALINK 컬럼이면 이 필드에서 다음과 같은 등록 정보를 서술합니다.

- 첫번째 문자는 링크 유형을 나타내며 값은 U입니다.
- 두번째 문자는 링크 제어 유형을 나타냅니다. 제어가 없는 경우에는 N 값, 파일 제어의 경우에는 F값이 유효합니다.
- 세번째 문자는 무결성 레벨을 나타내며 값은 A(모든 DATALINK 값을 제어하는 데이터베이스 관리 프로그램의 경우)입니다.
- 네번째 문자는 읽기 사용권한을 나타냅니다. 데이터베이스 판별 사용권한인 경우에는 D값, 파일 시스템 판별 사용권한인 경우에는 F값이 유효합니다.
- 다섯번째 문자는 읽기 사용권한을 나타냅니다. 블록화 액세스인 경우에는 B값, 파일 시스템 판별 사용권한인 경우에는 F값이 유효합니다.
- 여섯번째 문자는 복구 옵션을 나타냅니다. 유효한 값은 Y(DB2가 이 컬럼에 참조된 파일의 특정 시점 복구를 지원함) 및 N(지원하지 않음)입니다.
- 일곱번째 값은 데이터 파일이 링크해제될 때 취할 조치를 나타냅니다. 복원인 경우에는 R값, 파일 삭제의 경우에는 D값이 유효합니다.

IXFCREF

컬럼이 계층의 일부라면, 이 필드는 컬럼이 데이터 컬럼(D) 또는 참조 컬럼(R)인지 여부를 지정합니다.

IXFCNDIM

컬럼에 있는 차원의 수. 이 버전의 PC/IXF에서는 배열이 지원되지 않습니다. 따라서, 이 필드에는 0 정수 값의 문자 표현이 들어 있어야 합니다.

IXFCDSIZ

각 차원의 크기 또는 범위. 이 필드의 길이는 차원당 5바이트입니다. 배열이 지원되지 않기 때문에(즉, 차원의 수가 0이 되어야 함), 이 필드의 길이는 0이며, 실제로 존재하지 않습니다.

데이터 레코드

필드 이름	길이	유형	비고
IXFDRECL	06-BYTE	CHARACTER	레코드 길이
IXFDRECT	01-BYTE	CHARACTER	레코드 유형 = 'D'
IXFDRID	03-BYTE	CHARACTER	'D' 레코드 식별자
IXFDFIL1	04-BYTE	CHARACTER	예약됨
IXFDCOLS	가변	변수	컬럼 데이터

데이터 레코드에는 다음 필드가 들어 있습니다.

IXFDRECL

레코드 길이 표시기. 레코드 길이 표시기 뒤에 오는 PC/IXF 레코드 부분의 길이(바이트)를 나타내는 정수 값의 6바이트 문자 표현입니다. 즉 전체 레코드 크기에서 6바이트를 뺍니다. 각 D 레코드는 레코드에 저장된 마지막 데이터 컬럼의 현재 발생에 대해 모든 중요한 데이터를 포함할 수 있을 만큼 충분히 길어야 합니다.

IXFDRECT

이 레코드에 대해 D로 설정된 IXF 레코드 유형이며, 테이블에 대한 데이터 값을 포함하는 것을 나타냅니다.

IXFDRID

데이터의 행에 제공되는 여러 D 레코드의 배열에서 특정 D 레코드를 식별하는 레코드 식별자. 데이터 행의 첫번째 D 레코드의 경우, 이 필드는

IXF 파일 형식의 PC 버전

1 값을 가지며, 두번째 D 레코드의 경우 2 값을 가지며, 계속 그 다음 값을 가집니다. 각 데이터 행에서, C 레코드에서 호출된 모든 D 레코드 식별자는 실제로 존재해야 합니다.

IXDFIL1

예약된 필드와 일치되기 위해 네 개의 공백으로 설정된 여분 필드이며, 호스트 IXF 파일에 있을 수 있는 쉬프트아웃 문자를 위한 자리를 보유합니다.

IXFDCOLS

컬럼 데이터를 위한 영역. 데이터 레코드(D 레코드)의 데이터 영역은 하나 이상의 컬럼 항목으로 구성됩니다. D 레코드와 동일한 D 레코드 식별자를 갖는 각 컬럼 설명자 레코드에 대해 하나의 컬럼 항목을 가집니다. D 레코드에서, 컬럼 항목의 시작 위치는 C 레코드의 IXFCPOSN 값에 의해 지정됩니다.

컬럼 항목 데이터의 형식은 컬럼의 널(NULL) 입력 가능 여부에 따라 결정됩니다.

- 컬럼이 널(NULL) 입력 가능이면(IXFCNULL 필드가 Y로 설정됨), 컬럼 항목 데이터는 널(NULL) 표시기를 포함합니다. 컬럼이 널(NULL) 입력 불가능일 경우, 표시기 뒤에 실제 데이터베이스 값을 포함하는 데이터 유형 특정 정보가 표시됩니다. 널(NULL) 표시기는 널(NULL)이 아닐 경우 x'0000'으로 설정된 2바이트 값이며, 널(NULL)일 경우 x'FFFF'입니다.
- 컬럼이 널(NULL) 입력 불가능일 경우, 컬럼 항목 데이터는 실제 데이터베이스 값이 있는 데이터 유형 특정 정보만 포함합니다.

가변 길이 데이터 유형의 경우, 데이터 유형 특정 정보는 현재 길이 표시기를 포함합니다. 현재 길이 표시기는 IXFTMFRM 필드에 의해 지정된 형식의 2바이트 정수입니다.

D 레코드의 데이터 영역 길이는 32 771바이트를 초과할 수 없습니다.

응용프로그램 레코드

필드 이름	길이	유형	비고
IXFARECL	06-BYTE	CHARACTER	레코드 길이
IXFARECT	01-BYTE	CHARACTER	레코드 유형 = 'A'
IXFAPPID	12-BYTE	CHARACTER	응용프로그램 식별자
IXFADATA	가변	변수	응용프로그램 특정 데이터

응용프로그램 레코드에는 다음 필드가 들어 있습니다.

IXFARECL

레코드 길이 표시기. 레코드 길이 표시기 뒤에 오는 PC/IXF 레코드 부분의 길이(바이트)를 나타내는 정수 값의 6바이트 문자 표현입니다. 즉 전체 레코드 크기에서 6바이트를 뺍니다. 각 A 레코드는 최소한 전체 IXFAPPID 필드를 포함할 수 있을 만큼 충분히 길어야 합니다.

IXFARECT

이 레코드에 대해 A로 설정된 IXF 레코드 유형이며, 응용프로그램 레코드임을 나타냅니다. 이들 레코드는 응용프로그램 식별자가 나타내는 데이터의 형식 및 내용에 관해 특정 지식이 없는 프로그램에서는 무시됩니다.

IXFAPPID

A 레코드를 작성하는 응용프로그램을 식별하는 응용프로그램 식별자. 데이터베이스 관리 프로그램에 의해 작성된 PC/IXF 파일은 이 필드의 처음 6자가 데이터베이스 관리 프로그램을 식별하는 상수로 설정되고, 마지막 6자가 데이터베이스 관리 프로그램 또는 A 레코드를 작성하는 다른 응용프로그램의 릴리스나 버전을 식별하는 A 레코드를 가질 수 있습니다.

IXFADATA

이 필드에는 응용프로그램 종속 보충 데이터가 들어 있습니다. 이 데이터의 형식과 내용은 A 레코드를 작성하는 프로그램 및 A 레코드를 처리할 것 같은 다른 프로그램에게만 알려집니다.

DB2 INDEX RECORD

필드 이름	길이	유형	비고
IXFARECL	006-BYTE	CHARACTER	레코드 길이
IXFARECT	001-BYTE	CHARACTER	레코드 유형 = 'A'
IXFAPPID	012-BYTE	CHARACTER	응용프로그램 식별자 = 'DB2 02.00'
IXFAITYP	001-BYTE	CHARACTER	응용프로그램 특정 데이터 유형 = 'I'
IXFADATE	008-BYTE	CHARACTER	'H' 레코드에서 작성된 데이터

IXF 파일 형식의 PC 버전

IXFATIME	006-BYTE	CHARACTER	'H' 레코드에서 작성된 시간
IXFANDXL	002-BYTE	SHORT INT	색인 이름 길이
IXFANDXN	256-BYTE	CHARACTER	색인 이름
IXFANCL	002-BYTE	SHORT INT	색인 작성자 이름 길이
IXFANCN	256-BYTE	CHARACTER	색인 작성자 이름
IXFATABL	002-BYTE	SHORT INT	테이블 이름 길이
IXFATABN	256-BYTE	CHARACTER	테이블 이름
IXFATCL	002-BYTE	SHORT INT	테이블 작성자 이름 길이
IXFATCN	256-BYTE	CHARACTER	테이블 작성자 이름
IXFAUNIQ	001-BYTE	CHARACTER	고유한 규칙
IXFACNT	002-BYTE	CHARACTER	컬럼 개수
IXFAREVS	001-BYTE	CHARACTER	역 스캔 플래그 허용
IXFAPCTF	002-BYTE	CHARACTER	여유 pct 양
IXFAPCTU	002-BYTE	CHARACTER	minpctused 양
IXFAEXTI	001-BYTE	CHARACTER	예약됨
IXFACNML	002-BYTE	SHORT INT	컬럼 이름 길이
IXFACOLN	varying	CHARACTER	색인에서 컬럼 이름

각 사용자 정의 색인에 대해 이러한 유형의 레코드 하나가 지정됩니다. 이 레코드는 테이블에 대한 모든 C 레코드 뒤에 놓입니다. DB2 색인 레코드에는 다음 필드가 들어 있습니다.

IXFARECL

레코드 길이 표시기. 레코드 길이 표시기 뒤에 오는 PC/IXF 레코드 부분의 길이(바이트)를 나타내는 정수 값의 6바이트 문자 표현입니다. 즉 전체 레코드 크기에서 6바이트를 뺍니다. 각 A 레코드는 최소한 전체 IXFAPPID 필드를 포함할 수 있을 만큼 충분히 길어야 합니다.

IXFARECT

이 레코드에 대해 A로 설정된 IXF 레코드 유형이며, 응용프로그램 레코드를 나타냅니다. 이들 레코드는 응용프로그램 식별자가 나타내는 데이터의 형식 및 내용에 관해 특정 지식이 없는 프로그램에서는 무시됩니다.

IXFAPPID

이 A 레코드를 작성하는 응용프로그램으로서 DB2를 식별하는 응용프로그램 식별자.

IXFAITYP

이것이 DB2 응용프로그램 레코드의 부속 유형 "I"임을 지정합니다.

IXFADATE

yyyymmdd 형식으로 된 파일 작성 날짜. 이 필드는 IXFHDATE와 같은 값을 가져야 합니다.

IXFATIME

hhmmss 형식으로 된 파일 작성 시간. 이 필드는 IXFHTIME와 같은 값을 가져야 합니다.

IXFANDXL

IXFANDXN 필드에 있는 색인 이름의 길이(바이트).

IXFANDXN

색인 이름.

IXFANCL

IXFANCN 필드에 있는 색인 작성자 이름의 길이(바이트).

IXFANCN

색인 작성자 이름.

IXFATABL

IXFATABN 필드에 있는 테이블 이름의 길이(바이트).

IXFATABN

테이블 이름.

IXFATCL

IXFATCN 필드에 있는 테이블 작성자 이름의 길이(바이트).

IXFATCN

테이블 작성자 이름.

IXFAUNIQ

색인의 유형을 지정합니다. 유효한 값은 기본 키의 경우에는 P, 고유 색인의 경우에는 U, 고유하지 않은 색인의 경우에는 D입니다.

IXFACCNT

색인 정의에 컬럼수를 지정합니다.

IXFAREVS

이 색인에 대해 역 스캔이 허용되는지 여부를 지정합니다. 유효한 값은 역 스캔의 경우에는 Y, 역이 아닌 스캔의 경우에는 N입니다.

IXF 파일 형식의 PC 버전

IXFAPCTF

여유 공간으로 남게될 색인 페이지의 퍼센트를 지정합니다. 유효한 값의 범위는 -1에서 99입니다. -1 또는 0 값이 지정되면 시스템 기본값이 사용됩니다.

IXFAPCTU

두 색인 페이지가 병합되기 전에 자유 공간 상태로 있어야 할 색인 페이지의 최소 백분율을 지정합니다. 유효한 값의 범위는 00에서 99입니다.

IXFAEXTI

향후 사용을 위해 예약됨.

IXFACNML

IXFACOLN 필드에 있는 컬럼 이름의 길이(바이트).

IXFACOLN

이 색인의 일부분인 컬럼의 이름. 유효한 값은 *+name-name...* 형식으로 되어 있으며 여기서, +는 컬럼에 대한 오름차순 정렬을 지정하고 -는 컬럼에 대한 내림차순 정렬을 지정합니다.

DB2 계층 레코드

필드 이름	길이	유형	비고
IXFARECL	006-BYTE	CHARACTER	레코드 길이
IXFARECT	001-BYTE	CHARACTER	레코드 유형 = 'A'
IXFAPPID	012-BYTE	CHARACTER	응용프로그램 식별자 = 'DB2 02.00'
IXFAXTYP	001-BYTE	CHARACTER	응용프로그램 특정 데이터 유형 = 'X'
IXFADATE	008-BYTE	CHARACTER	'H' 레코드에서 작성된 데이터
IXFATIME	006-BYTE	CHARACTER	'H' 레코드에서 작성된 시간
IXFAYCNT	010-BYTE	CHARACTER	이 계층에 대한 'Y' 레코드 계수
IXFAYSTR	010-BYTE	CHARACTER	이 계층의 시작 컬럼

계층을 서술하기 위해 이러한 유형의 레코드 하나가 사용됩니다. 모든 서브테이블 레코드(아래 참조)는 계층 레코드 바로 뒤에 놓여야 하고, 계층 레코드는 테이블에 대한 모든 C 레코드 뒤에 놓입니다. DB2 계층 레코드에는 다음 필드가 들어 있습니다.

IXFARECL

레코드 길이 표시기. 레코드 길이 표시기 뒤에 오는 PC/IXF 레코드 부분의 길이(바이트)를 나타내는 정수 값의 6바이트 문자 표현입니다. 즉 전체

레코드 크기에서 6바이트를 뺍니다. 각 A 레코드는 최소한 전체 IXFAPPID 필드를 포함할 수 있을 만큼 충분히 길어야 합니다.

IXFARECT

이 레코드에 대해 A로 설정된 IXF 레코드 유형이며, 응용프로그램 레코드를 나타냅니다. 이들 레코드는 응용프로그램 식별자가 나타내는 데이터의 형식 및 내용에 관해 특정 지식이 없는 프로그램에서는 무시됩니다.

IXFAPPID

이 A 레코드를 작성하는 응용프로그램으로서 DB2를 식별하는 응용프로그램 식별자.

IXFAXTYP

이것이 DB2 응용프로그램 레코드의 부속 유형 "X"임을 지정합니다.

IXFADATE

yyyymmdd 형식으로 된 파일 작성 날짜. 이 필드는 IXFHDATE와 같은 값을 가져야 합니다.

IXFATIME

hhmmss 형식으로 된 파일 작성 시간. 이 필드는 IXFHTIME와 같은 값을 가져야 합니다.

IXFAYCNT

이 계층 레코드 뒤에 예상되는 서브테이블 레코드의 수를 지정합니다.

IXFAYSTR

내보내진 데이터의 시작 부분에 서브테이블 레코드의 색인을 지정합니다. 계층의 내보내기를 루트가 아닌 서브테이블에서 시작한 경우에는 이 서브테이블의 모든 상위 테이블들이 내보내집니다. IXF 파일 내부에서의 이 서브테이블 위치가 또한 이 필드에 저장됩니다. 첫번째 X 레코드는 색인이 0인 컬럼을 나타냅니다.

DB2 SUBTABLE RECORD

필드 이름	길이	유형	비고
IXFARECL	006-BYTE	CHARACTER	레코드 길이
IXFARECT	001-BYTE	CHARACTER	레코드 유형 = 'A'
IXFAPPID	012-BYTE	CHARACTER	응용프로그램 식별자 = 'DB2 02.00'
IXFAXTYP	001-BYTE	CHARACTER	응용프로그램 특정 데이터 유형 = 'Y'
IXFADATE	008-BYTE	CHARACTER	'H' 레코드에서 작성된 데이터

IXF 파일 형식의 PC 버전

IXFATIME	006-BYTE	CHARACTER	'H' 레코드에서 작성된 시간
IXFASCHL	003-BYTE	CHARACTER	유형 스키마 이름 길이
IXFASCHN	256-BYTE	CHARACTER	유형 스키마 이름
IXFATYPL	003-BYTE	CHARACTER	유형 이름 길이
IXFATYPN	256-BYTE	CHARACTER	유형 이름
IXFATABL	003-BYTE	CHARACTER	테이블 이름 길이
IXFATABN	256-BYTE	CHARACTER	테이블 이름
IXFAPNDX	010-BYTE	CHARACTER	상위 테이블의 서브테이블 색인
IXFASNDX	005-BYTE	CHARACTER	현재 테이블의 시작 컬럼 색인
IXFAENDX	005-BYTE	CHARACTER	현재 테이블의 끝 컬럼 색인

서브테이블을 계층의 일부분으로 서술하기 위해 이러한 유형의 레코드 하나가 사용됩니다. 계층에 속해 있는 모든 서브테이블 레코드들은 해당 계층 레코드 바로 뒤에 함께 저장되어야 합니다. 서브테이블은 하나 이상의 컬럼으로 구성되며 각 컬럼은 컬럼 레코드에 서술됩니다. 서브테이블에 있는 각 컬럼은 C 레코드의 연속 세트에 서술되어야 합니다. DB2 서브테이블 레코드에는 다음 필드가 들어 있습니다.

IXFARECL

레코드 길이 표시기. 레코드 길이 표시기 뒤에 오는 PC/IXF 레코드 부분의 길이(바이트)를 나타내는 정수 값의 6바이트 문자 표현입니다. 즉 전체 레코드 크기에서 6바이트를 뺍니다. 각 A 레코드는 최소한 전체 IXFAPPID 필드를 포함할 수 있을 만큼 충분히 길어야 합니다.

IXFARECT

이 레코드에 대해 A로 설정된 IXF 레코드 유형이며, 응용프로그램 레코드를 나타냅니다. 이들 레코드는 응용프로그램 식별자가 나타내는 데이터의 형식 및 내용에 관해 특정 지식이 없는 프로그램에서는 무시됩니다.

IXFAPPID

이 A 레코드를 작성하는 응용프로그램으로서 DB2를 식별하는 응용프로그램 식별자.

IXFAYTYP

이것이 DB2 응용프로그램 레코드의 부속 유형 "Y"임을 지정합니다.

IXFADATE

yyyymmdd 형식으로 된 파일 작성 날짜. 이 필드는 IXFHDATE와 같은 값을 가져야 합니다.

IXFATIME

hhmmss 형식으로 된 파일 작성 시간. 이 필드는 IXFHTIME와 같은 값을 가져야 합니다.

IXFASCHL

IXFASCHN 필드에 있는 서브테이블 스키마 이름의 길이(바이트).

IXFASCHN

서브테이블 스키마의 이름.

IXFATYPL

IXFATYPN 필드에 있는 서브테이블 이름의 길이(바이트).

IXFATYPN

서브테이블 이름.

IXFATABL

IXFATABN 필드에 있는 테이블 이름의 길이(바이트).

IXFATABN

테이블 이름.

IXFAPNDX

상위 서브테이블의 서브테이블 레코드 색인. 이 서브테이블이 계층의 루트인 경우에는 이 필드에 값 -1이 포함됩니다.

IXFASNDX

이 서브테이블을 구성하는 컬럼 레코드의 시작 색인.

IXFAENDX

이 서브테이블을 구성하는 컬럼 레코드의 끝 색인.

DB2 CONTINUATION RECORD

필드 이름	길이	유형	비고
IXFARECL	006-BYTE	CHARACTER	레코드 길이
IXFARECT	001-BYTE	CHARACTER	레코드 유형 = 'A'
IXFAPPID	012-BYTE	CHARACTER	응용프로그램 식별자 = 'DB2 02.00'
IXFACTYP	001-BYTE	CHARACTER	응용프로그램 특정 데이터 유형 = 'C'
IXFADATE	008-BYTE	CHARACTER	'H' 레코드에서 작성된 데이터
IXFATIME	006-BYTE	CHARACTER	'H' 레코드에서 작성된 시간
IXFALAST	002-BYTE	SHORT INT	마지막 디스켓 볼륨 번호
IXFATHIS	002-BYTE	SHORT INT	이번 디스켓 볼륨 번호
IXFANEXT	002-BYTE	SHORT INT	다음 디스켓 볼륨 번호

IXF 파일 형식의 PC 버전

파일이 마지막 볼륨이 아닌 이상, 이 레코드는 다중 볼륨 IXF 파일의 일부분인 각 파일의 끝 부분에 있습니다. 파일이 첫번째 볼륨이 아닌 이상, 이 레코드는 또한 다중 볼륨 IXF 파일의 일부분인 각 파일의 시작 부분에서 찾을 수 있습니다. 이 레코드의 목적은 파일 순서를 추적하는 것입니다. DB2 계속 레코드(DB2 continuation records)에는 다음 필드가 들어 있습니다.

IXFARECL

레코드 길이 표시기. 레코드 길이 표시기 뒤에 오는 PC/IXF 레코드 부분의 길이(바이트)를 나타내는 정수 값의 6바이트 문자 표현입니다. 즉 전체 레코드 크기에서 6바이트를 뺍니다. 각 A 레코드는 최소한 전체 IXFAPPID 필드를 포함할 수 있을 만큼 충분히 길어야 합니다.

IXFARECT

이 레코드에 대해 A로 설정된 IXF 레코드 유형이며, 응용프로그램 레코드임을 나타냅니다. 이들 레코드는 응용프로그램 식별자가 나타내는 데이터의 형식 및 내용에 관해 특정 지식이 없는 프로그램에서는 무시됩니다.

IXFAPPID

이 A 레코드를 작성하는 응용프로그램으로서 DB2를 식별하는 응용프로그램 식별자.

IXFACTYP

이것이 DB2 응용프로그램 레코드의 부속 유형 "C"임을 지정합니다.

IXFADATE

yyyymmdd 형식으로 된 파일 작성 날짜. 이 필드는 IXFHDATE와 같은 값을 가져야 합니다.

IXFATIME

hhmmss 형식으로 된 파일 작성 시간. 이 필드는 IXFHTIME와 같은 값을 가져야 합니다.

IXFALAST

이 필드는 little-endian 형식의 2진 필드입니다. 값은 IXFATHIS의 값보다 하나 적어야 합니다.

IXFATHIS

이 필드는 little-endian 형식의 2진 필드입니다. 연속적인 블록에서는 이 필드값 또한 연속되어야 합니다. 첫번째 블록 값은 1입니다.

IXFANEXT

이 필드는 little-endian 형식의 2진 필드입니다. 레코드가 파일의 시작 부분에 있지 않는 한 값은 IXFATHIS에 있는 값보다 하나 커야 하며, 시작 부분에 있는 경우에는 값이 0이어야 합니다.

DB2 TERMINATE RECORD

필드 이름	길이	유형	비고
IXFARECL	006-BYTE	CHARACTER	레코드 길이
IXFARECT	001-BYTE	CHARACTER	레코드 유형 = 'A'
IXFAPPID	012-BYTE	CHARACTER	응용프로그램 식별자 = 'DB2 02.00'
IXFAETYP	001-BYTE	CHARACTER	응용프로그램 특정 데이터 유형 = 'E'
IXFADATE	008-BYTE	CHARACTER	'H' 레코드에서 작성된 데이터
IXFATIME	006-BYTE	CHARACTER	'H' 레코드에서 작성된 시간

이 레코드는 IXF 파일의 끝부분에 있는 파일끝 표시문자입니다. DB2 종료 레코드에는 다음 필드가 들어 있습니다.

IXFARECL

레코드 길이 표시기. 레코드 길이 표시기 뒤에 오는 PC/IXF 레코드 부분의 길이(바이트)를 나타내는 정수 값의 6바이트 문자 표현입니다. 즉 전체 레코드 크기에서 6바이트를 뺍니다. 각 A 레코드는 최소한 전체 IXFAPPID 필드를 포함할 수 있을 만큼 충분히 길어야 합니다.

IXFARECT

이 레코드에 대해 A로 설정된 IXF 레코드 유형이며, 응용프로그램 레코드를 나타냅니다. 이들 레코드는 응용프로그램 식별자가 나타내는 데이터의 형식 및 내용에 관해 특정 지식이 없는 프로그램에서는 무시됩니다.

IXFAPPID

이 A 레코드를 작성하는 응용프로그램으로서 DB2를 식별하는 응용프로그램 식별자.

IXFAETYP

이것이 DB2 응용프로그램 레코드의 부속 유형 "E"임을 지정합니다.

IXF 파일 형식의 PC 버전

IXFADATE

yyyymmdd 형식으로 된 파일 작성 날짜. 이 필드는 IXFHDATE와 같은 값을 가져야 합니다.

IXFATIME

hhmmss 형식으로 된 파일 작성 시간. 이 필드는 IXFHTIME와 같은 값을 가져야 합니다.

DB2 식별 레코드

필드 이름	길이	유형	비고
IXFARECL	06-BYTE	CHARACTER	레코드 길이
IXFARECT	01-BYTE	CHARACTER	레코드 유형 = 'A'
IXFAPPID	12-BYTE	CHARACTER	응용프로그램 식별자
IXFATYPE	01-BYTE	CHARACTER	응용프로그램 특정 레코드 유형 = 'S'
IXFADATE	08-BYTE	CHARACTER	응용프로그램 레코드 작성 날짜
IXFATIME	06-BYTE	CHARACTER	응용프로그램 레코드 작성 시간
IXFACOLN	06-BYTE	CHARACTER	식별 컬럼의 컬럼 번호
IXFAITYP	01-BYTE	CHARACTER	항상 작성됨 ('Y' 또는 'N')
IXFASTRT	33-BYTE	CHARACTER	식별 START AT 값
IXFAINCR	33-BYTE	CHARACTER	식별 INCREMENT BY 값
IXFACACH	10-BYTE	CHARACTER	식별 CACHE 값

DB2 식별 레코드에는 다음 필드가 들어 있습니다.

IXFARECL

레코드 길이 표시기. 레코드 길이 표시기 뒤에 오는 PC/IXF 레코드 부분의 길이(바이트)를 나타내는 정수 값의 6바이트 문자 표현입니다. 즉 전체 레코드 크기에서 6바이트를 뺍니다. 각 A 레코드는 최소한 전체 IXFAPPID 필드를 포함할 수 있을 만큼 충분히 길어야 합니다.

IXFARECT

이 레코드에 대해 A로 설정된 IXF 레코드 유형이며, 응용프로그램 레코드를 나타냅니다. 이들 레코드는 응용프로그램 식별자가 나타내는 데이터의 형식 및 내용에 관해 특정 지식이 없는 프로그램에서는 무시됩니다.

IXFAPPID

이 A 레코드를 작성하는 응용프로그램으로서 DB2를 식별하는 응용프로그램 식별자.

IXFATYPE

응용프로그램 특정 레코드 유형. 이 필드는 항상 "S" 값을 가져야 합니다.

IXFADATE

yyyymmdd 형식으로 된 파일 작성 날짜. 이 필드는 IXFHDATE와 같은 값을 가져야 합니다.

IXFATIME

hhmmss 형식으로 된 파일 작성 시간. 이 필드는 IXFHTIME와 같은 값을 가져야 합니다.

IXFACOLN

테이블에 있는 식별 컬럼의 컬럼 번호.

IXFAITYP

식별 컬럼의 유형. "Y" 값은 식별 컬럼이 항상 GENERATED임을 나타냅니다. 다른 모든 값들은 컬럼 유형이 GENERATED BY DEFAULT 인 것으로 풀이됩니다.

IXFASTRT

테이블 작성시에 CREATE TABLE문에 제공되었던 식별 컬럼에 대한 START AT 값.

IXFAINCR

테이블 작성시에 CREATE TABLE문에 제공되었던 식별 컬럼에 대한 INCREMENT BY 값.

IXFACACH

테이블 작성시에 CREATE TABLE문에 제공되었던 식별 컬럼에 대한 CACHE 값. "1" 값은 NO CACHE 옵션에 해당됩니다.

PC/IXF 데이터 유형

표 11. PC/IXF 데이터 유형

이름	IXFCTYPE 값	설명
BIGINT	492	IXFTMFRM에 의해 지정된 형식의 8바이트 정수. -9 223 372 036 854 775 808에서 9 223 372 036 854 775 807까지 사이의 모든 수를 나타냅니다. IXFCSBCP와 IXFCDBCP는 중요하지 않으므로 0이 되어야 합니다. IXFCLENG는 사용되지 않으며 공백이 되어야 합니다.

IXF 파일 형식의 PC 버전

표 11. PC/IXF 데이터 유형 (계속)

이름	IXFCTYPE 값	설명
BLOB, CLOB	404, 408	<p>가변 길이 문자열. 이 문자열의 최대 길이는 컬럼 설명자 레코드의 IXFCLENG 필드에 들어 있으며, 32 767바이트를 초과할 수 없습니다. 문자열 앞에는 문자열의 길이(바이트)를 지정하는 4 바이트 정수인 현재 길이 표시기가 붙습니다. 이 문자열은 IXFCSBCP가 지정하는 코드 페이지에 있습니다.</p> <p>다음은 2진 대형 오브젝트(BLOB)에만 적용됩니다. IXFCSBCP가 0일 경우, 문자열은 비트 데이터이며 모든 변환 프로그램에 의해 변환되지 않아야 합니다.</p> <p>다음은 CLOB에만 적용됩니다. IXFCDBCP가 0이 아닐 경우, 문자열은 IXFCDBCP가 지정한 코드 페이지에 있는 2바이트 문자도 포함할 수 있습니다.</p>
BLOB_FILE, CLOB_FILE, DBCLOB_FILE	916, 920, 924	<p><i>name_length</i> 및 <i>name</i> 필드가 채워져 있는 SQLFILE 구조를 포함하는 고정 길이 필드. 이 구조의 길이는 컬럼 설명자 레코드의 IXFCLENG 필드에 들어 있으며, 255바이트를 초과할 수 없습니다. 이 파일 이름은 IXFCSBCP가 지정하는 코드 페이지에 있습니다. IXFCDBCP가 0이 아닐 경우, 파일 이름은 IXFCDBCP가 지정한 코드 페이지에 있는 2바이트 문자도 포함할 수 있습니다. IXFCSBCP가 0일 경우, 파일 이름은 비트 데이터이며 모든 변환 프로그램에서 변환되지 않아야 합니다.</p> <p>구조의 길이가 IXFCLENG에 저장되므로 원래 LOB의 실제 길이는 없어집니다. BLOB_FILE, CLOB_FILE 또는 DBCLOB_FILE 유형의 컬럼이 있는 IXF 파일은 LOB 필드를 재작성하는데 사용되어서는 안 됩니다. 왜냐하면, LOB가 <i>sql_lobfile_len</i> 길이로 작성되기 때문입니다.</p>

표 11. PC/IXF 데이터 유형 (계속)

이름	IXFCTYPE 값	설명
CHAR	452	고정 길이 문자열. 이 문자열 길이는 컬럼 설명자 레코드의 IXFCLENG 필드에 들어 있으며, 254바이트를 초과할 수 없습니다. 이 문자열은 IXFCSBCP가 지정하는 코드 페이지에 있습니다. IXFCDBCP가 0이 아닐 경우, 문자열은 IXFCDBCP가 지정한 코드 페이지에 있는 2바이트 문자도 포함할 수 있습니다. IXFCSBCP가 0일 경우, 문자열은 비트 데이터이며 모든 변환 프로그램에서 변환되지 않아야 합니다.
DATE	384	그레고리력에 따른 시점. 각 날짜는 ISO 형식의 10바이트 문자열인 yyyy-mm-dd입니다. 연도 부분의 범위는 0001에서 9999까지입니다. 월 부분의 범위는 01에서 12까지입니다. 일 부분의 범위는 01에서 n까지이며, n은 월에 따라 다르며, 월과 윤년의 날짜에 대한 통상적인 규칙을 사용합니다. 모든 부분의 앞에 들어 있는 0은 생략할 수 없습니다. IXFCLENG는 사용되지 않으며 공백이 되어야 합니다. DATE의 유효 문자는 모든 PC ASCII 코드 페이지에서 일정합니다. 따라서, IXFCSBCP와 IXFCDBCP는 중요하지 않으므로 0이 되어야 합니다.
DBCLOB	412	2바이트 문자의 가변 길이 문자열. 컬럼 설명자 레코드의 IXFCLENG 필드는 문자열에 있는 2바이트 문자의 최대수를 지정하며 16 383을 초과할 수 없습니다. 문자열 앞에는 2바이트 문자로 된 문자열의 길이를 지정하는 4바이트 정수인 현재 길이 표시기가 들어갑니다(즉, 이 정수의 값은 문자열 길이(바이트)의 절반입니다). 문자열은 C 레코드의 IXFCDBCP가 지정한 대로 DBCS 코드 페이지에 들어 있습니다. 문자열이 2바이트 문자 데이터로만 구성되므로, IXFCSBCP는 0이 되어야 합니다. 여기에서는 쉬프트인이나 쉬프트아웃 문자가 사용되지 않습니다.

IXF 파일 형식의 PC 버전

표 11. PC/IXF 데이터 유형 (계속)

이름	IXFCTYPE 값	설명
DECIMAL	484	정밀도 P(컬럼 설명자 레코드에 있는 IXFCLENG의 처음 3바이트로 지정된)와 스케일 S(EXFCLENG의 마지막 두 바이트로 지정된)로 된 압축시킨 수(packed decimal). 압축시킨 수의 길이(바이트)는 (P+2)/2입니다. 정밀도는 1과 31 사이의 홀수여야 합니다. 압축시킨 수는 IXFTMFRM에 의해 지정된 내부 형식으로 되어 있으며, PC에 대한 압축시킨 수는 System/370에 대한 압축시킨 수와 동일하게 정의됩니다. IXFCSBCP와 IXFCDBCP는 중요하지 않으므로 0이 되어야 합니다.
FLOATING POINT	480	IXFCLENG가 9 또는 4로 설정되는 것에 따라, 긴(8바이트) 또는 짧은(4바이트) 부동 소수. 이 데이터는 IXFTMFRM에 의해 지정된 대로 내부 머신 형식으로 되어 있습니다. IXFCSBCP와 IXFCDBCP는 중요하지 않으므로 0이 되어야 합니다. 4바이트 부동 소수는 데이터베이스 관리 프로그램에서 지원되지 않습니다.
GRAPHIC	468	2바이트 문자의 가변 길이 문자열. 컬럼 설명자 레코드의 IXFCLENG 필드는 문자열에 있는 2바이트 문자의 수를 지정하며, 127을 초과할 수 없습니다. 이 문자열의 실제 길이는 IXFCLENG 필드 값(바이트)의 두 배입니다. 문자열은 C 레코드의 IXFCDBCP가 지정한 대로 DBCS 코드 페이지에 들어 있습니다. 문자열이 2바이트 문자 데이터로만 구성되므로, IXFCSBCP는 0이 되어야 합니다. 여기에서는 쉬프트인이나 쉬프트아웃 문자가 사용되지 않습니다.
INTEGER	496	IXFTMFRM에 의해 지정된 형식의 4바이트 정수. -2 147 483 648과 147 483 647 사이의 모든 수를 나타냅니다. IXFCSBCP와 IXFCDBCP는 중요하지 않으므로 0이 되어야 합니다. IXFCLENG는 사용되지 않으며 공백이 되어야 합니다.

표 11. PC/IXF 데이터 유형 (계속)

이름	IXFCTYPE 값	설명
LONGVARCHAR	456	가변 길이 문자열. 이 문자열의 최대 길이는 컬럼 설명자 레코드의 IXFCLENG 필드에 들어 있으며, 32 767바이트를 초과할 수 없습니다. 문자열 앞에는 문자열의 길이(바이트)를 지정하는 2바이트 정수인 현재 길이 표시기가 들어 있습니다. 이 문자열은 IXFCSBCP가 지정하는 코드 페이지에 있습니다. IXFCDBCP가 0이 아닐 경우, 문자열은 IXFCDBCP가 지정한 코드 페이지에 있는 2바이트 문자도 포함할 수 있습니다. IXFCSBCP가 0일 경우, 문자열은 비트 데이터이며 모든 변환 프로그램에서 변환되지 않아야 합니다.
LONG VARGRAPHIC	472	2바이트 문자의 가변 길이 문자열. 컬럼 설명자 레코드의 IXFCLENG 필드는 문자열에 대한 2바이트 문자의 최대수를 지정하며 16 383을 초과할 수 없습니다. 문자열 앞에는 2바이트 문자로 된 문자열의 길이를 지정하는 2바이트 정수인 현재 길이 표시기가 들어갑니다(즉, 이 정수의 값은 문자열 길이(바이트)의 절반입니다). 문자열은 C 레코드의 IXFCDBCP가 지정한 대로 DBCS 코드 페이지에 들어 있습니다. 문자열이 2바이트 문자 데이터로만 구성되므로, IXFCSBCP는 0이 되어야 합니다. 여기에서는 쉬프트인이나 쉬프트아웃 문자가 사용되지 않습니다.
SMALLINT	500	IXFTMFRM에 의해 지정된 형식의 2바이트 정수. -32 768과 +32 767 사이의 모든 수를 나타냅니다. IXFCSBCP와 IXFCDBCP는 중요하지 않으므로 0이 되어야 합니다. IXFCLENG는 사용되지 않으며 공백이 되어야 합니다.

IXF 파일 형식의 PC 버전

표 11. PC/IXF 데이터 유형 (계속)

이름	IXFCTYPE 값	설명
TIME	388	24시간제에 따른 시점. 각 시간은 ISO 형식의 8 바이트 문자열인 <i>hh.mm.ss</i> 입니다. 시간 부분의 범위는 00에서 24까지이며, 다른 부분의 범위는 00에서 59까지입니다. 시간이 24일 경우, 다른 부분은 00입니다. 최소 시간은 00.00.00이고, 최대 시간은 24.00.00입니다. 모든 부분의 앞에 들어 있는 0은 생략할 수 없습니다. IXFCLENG는 사용되지 않으며 공백이 되어야 합니다. TIME의 유효 문자는 모든 PC ASCII 코드 페이지에서 일정합니다. 따라서, IXFCSBCP와 IXFCDBCP는 중요하지 않으므로 0이 되어야 합니다.
TIMESTAMP	392	마이크로초 정밀도에 따른 날짜와 시간. 각 시간 소인은 <i>yyyy-mm-dd-hh.mm.ss.nnnnnn</i> (년 월 일 시간 분 초 마이크로초) 형식의 문자열입니다. IXFCLENG는 사용되지 않으며 공백이 되어야 합니다. TIMESTAMP의 유효 문자는 모든 PC ASCII 코드 페이지에서 일정합니다. 따라서, IXFCSBCP와 IXFCDBCP는 중요하지 않으므로 0이 되어야 합니다.
VARCHAR	448	가변 길이 문자열. 문자열의 최대 길이(바이트)는 컬럼 설명자 레코드의 IXFCLENG 필드에 들어 있으며 254바이트를 초과할 수 없습니다. 문자열 앞에는 문자열의 길이(바이트)를 지정하는 2 바이트 정수로 된 컬럼 길이 표시기가 들어 있습니다. 이 문자열은 IXFCSBCP가 지정하는 코드 페이지에 있습니다. IXFCDBCP가 0이 아닐 경우, 문자열은 IXFCDBCP가 지정한 코드 페이지에 있는 2바이트 문자도 포함할 수 있습니다. IXFCSBCP가 0일 경우, 문자열은 비트 데이터이며 모든 변환 프로그램에서 변환되지 않아야 합니다.

표 11. PC/IXF 데이터 유형 (계속)

이름	IXFCTYPE 값	설명
VARGRAPHIC	464	2바이트 문자의 가변 길이 문자열. 컬럼 설명자 레코드의 IXFCLENG 필드는 문자열에 있는 2바이트 문자의 최대수를 지정하며 127을 초과할 수 없습니다. 문자열 앞에는 2바이트 문자로 된 문자열의 길이를 지정하는 2바이트 정수인 현재 길이 표시기가 들어갑니다(즉, 이 정수의 값은 문자열 길이(바이트)의 절반입니다). 문자열은 C 레코드의 IXFCDBCP가 지정한 대로 DBCS 코드 페이지에 들어 있습니다. 문자열이 2바이트 문자 데이터로만 구성되므로, IXFCSBCP는 0이 되어야 합니다. 여기에서는 쉬프트인이나 쉬프트아웃 문자가 사용되지 않습니다.

PC/IXF 문자 또는 그래픽 컬럼에 대한 IXFCSBCP와 IXFCDBCP 값의 모든 조합이 유효하지는 않습니다. 유효하지 않은(IXFCSBCP,IXFCDBCP) 조합이 있는 PC/IXF 문자나 그래픽 컬럼은 유효하지 않은 데이터 유형입니다.

표 12. 유효한 PC/IXF 데이터 유형

PC/IXF 데이터 유형	유효한 (IXFCSBCP, IXFCDBCP) 쌍	유효하지 않은 (IXFCSBCP,IXFCDBCP) 쌍
CHAR, VARCHAR 또는 LONG VARCHAR	(0,0), (x,0) 또는 (x,y)	(0,y)
BLOB	(0,0)	(x,0), (0,y) 또는 (x,y)
CLOB	(x,0), (x,y)	(0,0), (0,y)
GRAPHIC, VARGRAPHIC, LONG VARGRAPHIC 또는 DBCLOB	(0,y)	(0,0), (x,0) 또는 (x,y)
주: x와 y는 0이 아닙니다.		

PC/IXF 데이터 유형 설명

표 13. PC/IXF 파일 형식에 사용할 수 있는 데이터 유형 형식.

데이터 유형	내보내기 유틸리티에 의해 생성된 파일에 있는 형식	가져오기 유틸리티에 사용 가능한 형식
BIGINT	데이터베이스 컬럼과 동일한 BIGINT 컬럼이 작성됩니다.	모든 숫자 유형(SMALLINT, INTEGER, BIGINT, DECIMAL 또는 FLOAT)에 있는 컬럼이 사용 가능합니다. 각 값은 -9 223 372 036 854 775 808에서 9 223 372 036 854 775 807까지 범위에 있지 않을 경우 거부됩니다.
BLOB	PC/IXF BLOB 컬럼이 작성됩니다. 데이터베이스 컬럼의 최대 길이, SBCS CPGID 값 및 DBCS CPGID 값은 컬럼 설명자 레코드에 복사됩니다.	PC/IXF CHAR, VARCHAR, LONG VARCHAR, BLOB 또는 BLOB_FILE 컬럼은 다음 경우에 사용 가능합니다. <ul style="list-style-type: none"> • 데이터베이스 컬럼이 FOR BIT DATA로 표시된 경우 • PC/IXF 컬럼 1바이트 코드 페이지 값이 데이터베이스 컬럼의 SBCS CPGID와 동일하고 PC/IXF 컬럼 2바이트 코드 페이지 값이 0이거나 데이터베이스 컬럼의 DBCS CPGID와 동일할 경우 PC/IXF GRAPHIC, VARGRAPHIC, LONG VARGRAPHIC BLOB 컬럼도 사용 가능합니다. PC/IXF 컬럼이 고정 길이일 경우, 이 길이는 데이터베이스 컬럼의 최대 길이와 호환 가능해야 합니다. 294 페이지의 『FORCEIN 옵션』에서 자세한 내용을 참조하십시오.

표 13. PC/IXF 파일 형식에 사용할 수 있는 데이터 유형 형식. (계속)

데이터 유형	내보내기 유틸리티에 의해 생성된 파일에 있는 형식	가져오기 유틸리티에 사용 가능한 형식
CHAR	PC/IXF CHAR 컬럼이 작성됩니다. 데이터베이스 컬럼 길이, SBCS CPGID 값 및 DBCS CPGID 값은 PC/IXF 컬럼 설명자 레코드에 복사됩니다.	PC/IXF CHAR, VARCHAR 또는 LONG VARCHAR 컬럼은 다음 경우에 사용 가능합니다. <ul style="list-style-type: none"> • 데이터베이스 컬럼이 FOR BIT DATA로 표시된 경우 • PC/IXF 컬럼 1바이트 코드 페이지 값이 데이터베이스 컬럼의 SBCS CPGID와 동일하고 PC/IXF 컬럼 2바이트 코드 페이지 값이 0이거나 데이터베이스 컬럼의 DBCS CPGID와 동일할 경우 데이터베이스 컬럼이 FOR BIT DATA로 표시될 경우, PC/IXF GRAPHIC, VARGRAPHIC 또는 LONG VARGRAPHIC 컬럼도 사용 가능합니다. 모든 경우에서, PC/IXF 컬럼이 고정 길이일 경우, 이 길이는 데이터베이스 컬럼의 길이와 호환 가능해야 합니다. 이 데이터는 필요한 경우, 오른쪽에 1바이트 공간(x'20')으로 채워집니다. 294 페이지의 『FORCEIN 옵션』에서 자세한 내용을 참조하십시오.
CLOB	PC/IXF CLOB 컬럼이 작성됩니다. 데이터베이스 컬럼의 최대 길이, SBCS CPGID 값 및 DBCS CPGID 값은 컬럼 설명자 레코드에 복사됩니다.	PC/IXF 컬럼 1바이트 코드 페이지 값이 데이터베이스 컬럼의 SBCS CPGID와 동일하고 PC/IXF 컬럼 2바이트 코드 페이지 값이 0이거나 데이터베이스 컬럼의 DBCS CPGID와 동일할 경우, PC/IXF CHAR, VARCHAR, LONG VARCHAR, CLOB 또는 CLOB_FILE 컬럼이 사용 가능합니다. PC/IXF 컬럼이 고정 길이일 경우, 이 길이는 데이터베이스 컬럼의 최대 길이와 호환 가능해야 합니다. 294 페이지의 『FORCEIN 옵션』에서 자세한 내용을 참조하십시오.

IXF 파일 형식의 PC 버전

표 13. PC/IXF 파일 형식에 사용할 수 있는 데이터 유형 형식. (계속)

데이터 유형	내보내기 유틸리티에 의해 생성된 파일에 있는 형식	가져오기 유틸리티에 사용 가능한 형식
DATE	데이터베이스 컬럼과 동일한 DATE 컬럼이 작성됩니다.	DATE 유형의 PC/IXF 컬럼은 일반적인 입력입니다. 가져오기 유틸리티는 비호환적인 길이를 갖는 컬럼을 제외하고는 모든 문자 유형의 컬럼을 사용하려고 시도하기도 합니다. PC/IXF 파일에 있는 문자 컬럼은 목표 데이터베이스의 국가 코드와 일치하는 형식의 날짜를 포함해야 합니다.
DBCLOB	PC/IXF DBCLOB 컬럼이 작성됩니다. 데이터베이스 컬럼의 최대 길이, SBCS CPGID 값 및 DBCS CPGID 값은 컬럼 설명자 레코드에 복사됩니다.	PC/IXF GRAPHIC, VARGRAPHIC, LONG VARGRAPHIC, DBCLOB 또는 DBCLOB_FILE 컬럼은 PC/IXF 컬럼 2바이트 코드 페이지 값이 데이터베이스 컬럼의 값과 동일할 때 사용할 수 있습니다. PC/IXF 컬럼이 고정 길이일 경우, 이 길이는 데이터베이스 컬럼의 최대 길이와 호환 가능해야 합니다. 294 페이지의 『FORCEIN 옵션』에서 자세한 내용을 참조하십시오.
DECIMAL	데이터베이스 컬럼과 동일한 DECIMAL 컬럼이 작성됩니다. 컬럼의 정밀도와 스케일은 컬럼 설명자 레코드에 저장되어 있습니다.	모든 숫자 유형(SMALLINT, INTEGER, BIGINT, DECIMAL 또는 FLOAT)에 있는 컬럼이 사용 가능합니다. 각 값은 가져오기 중인 DECIMAL 컬럼의 범위에 있지 않을 경우 거부됩니다.
FLOAT	데이터베이스 컬럼과 동일한 FLOAT 컬럼이 작성됩니다.	모든 숫자 유형(SMALLINT, INTEGER, BIGINT, DECIMAL 또는 FLOAT)에 있는 컬럼이 사용 가능합니다. 모든 값은 범위 내에 있습니다.
GRAPHIC(DBCS 전용)	PC/IXF GRAPHIC 컬럼이 작성됩니다. 데이터베이스 컬럼 길이, SBCS CPGID 값 및 DBCS CPGID 값은 컬럼 설명자 레코드에 복사됩니다.	PC/IXF GRAPHIC, VARGRAPHIC 또는 LONG VARGRAPHIC 컬럼은 PC/IXF 컬럼 2바이트 코드페이지 값이 데이터베이스 컬럼의 값과 동일할 경우 사용 가능합니다. PC/IXF 컬럼이 고정 길이일 경우, 이 길이는 데이터베이스 컬럼 길이와 호환 가능해야 합니다. 데이터는 필요한 경우, 오른쪽에 2바이트 공간(x'8140')으로 채워집니다. 294 페이지의 『FORCEIN 옵션』에서 자세한 내용을 참조하십시오.

표 13. PC/IXF 파일 형식에 사용할 수 있는 데이터 유형 형식. (계속)

데이터 유형	내보내기 유틸리티에 의해 생성된 파일에 있는 형식	가져오기 유틸리티에 사용 가능한 형식
INTEGER	데이터베이스 컬럼과 동일한 INTEGER 컬럼이 작성됩니다.	모든 숫자 유형(SMALLINT, INTEGER, BIGINT, DECIMAL 또는 FLOAT)에 있는 컬럼이 사용 가능합니다. 각 값은 -2 147 483 648에서 2 147 483 647까지 범위에 있지 않을 경우 거부됩니다.
LONG VARCHAR	PC/IXF LONG VARCHAR 컬럼이 작성됩니다. 데이터베이스 컬럼의 최대 길이, SBCS CPGID 값 및 DBCS CPGID 값은 컬럼 설명자 레코드에 복사됩니다.	PC/IXF CHAR, VARCHAR 또는 LONG VARCHAR 컬럼은 다음 경우에 사용 가능합니다. <ul style="list-style-type: none"> • 데이터베이스 컬럼이 FOR BIT DATA로 표시된 경우 • PC/IXF 컬럼 1바이트 코드 페이지 값이 데이터베이스 컬럼의 SBCS CPGID와 동일하고 PC/IXF 컬럼 2바이트 코드 페이지 값이 0이거나 데이터베이스 컬럼의 DBCS CPGID와 동일할 경우 <p>데이터베이스 컬럼이 FOR BIT DATA로 표시될 경우, PC/IXF GRAPHIC, VARGRAPHIC 또는 LONG VARGRAPHIC 컬럼도 사용 가능합니다. 모든 경우에서, PC/IXF 컬럼이 고정 길이일 경우, 이 길이는 데이터베이스 컬럼의 최대 길이와 호환 가능해야 합니다. 294 페이지의 『FORCEIN 옵션』에서 자세한 내용을 참조하십시오.</p>
LONG VARGRAPHIC (DBCS 전용)	PC/IXF LONG VARGRAPHIC 컬럼이 작성됩니다. 데이터베이스 컬럼의 최대 길이, SBCS CPGID 값 및 DBCS CPGID 값은 컬럼 설명자 레코드에 복사됩니다.	PC/IXF GRAPHIC, VARGRAPHIC 또는 LONG VARGRAPHIC 컬럼은 PC/IXF 컬럼 2바이트 코드페이지 값이 데이터베이스 컬럼의 값과 동일할 경우 사용 가능합니다. PC/IXF 컬럼이 고정 길이일 경우, 이 길이는 데이터베이스 컬럼의 최대 길이와 호환 가능해야 합니다. 294 페이지의 『FORCEIN 옵션』에서 자세한 내용을 참조하십시오.

IXF 파일 형식의 PC 버전

표 13. PC/IXF 파일 형식에 사용할 수 있는 데이터 유형 형식. (계속)

데이터 유형	내보내기 유틸리티에 의해 생성된 파일에 있는 형식	가져오기 유틸리티에 사용 가능한 형식
SMALLINT	데이터베이스 컬럼과 동일한 SMALLINT 컬럼이 작성됩니다.	모든 숫자 유형(SMALLINT, INTEGER, BIGINT, DECIMAL 또는 FLOAT)에 있는 컬럼이 사용 가능합니다. 각 값은 -32 768에서 32 767까지 범위에 있지 않을 경우 거부됩니다.
TIME	데이터베이스 컬럼과 동일한 TIME 컬럼이 작성됩니다.	TIME 유형의 PC/IXF 컬럼은 일반적인 입력입니다. 가져오기 유틸리티는 비호환적인 길이를 갖는 컬럼을 제외하고는 모든 문자 유형의 컬럼을 사용하려고 시도하기도 합니다. PC/IXF 파일에 있는 문자 컬럼은 목표 데이터베이스의 국가 코드와 일치하는 형식의 시간 데이터를 포함해야 합니다.
TIMESTAMP	데이터베이스 컬럼과 동일한 TIMESTAMP 컬럼이 작성됩니다.	TIMESTAMP 유형의 PC/IXF 컬럼은 일반적인 입력입니다. 가져오기 유틸리티는 비호환적인 길이를 갖는 컬럼을 제외하고는 모든 문자 유형의 컬럼을 사용하려고 시도하기도 합니다. PC/IXF 파일에 있는 문자 컬럼은 시간소인을 위한 입력 형식의 데이터를 포함해야 합니다.

표 13. PC/IXF 파일 형식에 사용할 수 있는 데이터 유형 형식. (계속)

데이터 유형	내보내기 유틸리티에 의해 생성된 파일에 있는 형식	가져오기 유틸리티에 사용 가능한 형식
VARCHAR	<p>데이터베이스 컬럼의 최대 길이가 254 이하일 경우, PC/IXF VARCHAR 컬럼이 작성됩니다. 데이터베이스 컬럼의 최대 길이가 254 보다 클 경우, PC/IXF LONG VARCHAR 컬럼이 작성됩니다. 데이터베이스 컬럼의 최대 길이, SBCS CPGID 값 및 DBCS CPGID 값은 컬럼 설명자 레코드에 복사됩니다.</p>	<p>PC/IXF CHAR, VARCHAR 또는 LONG VARCHAR 컬럼은 다음 경우에 사용 가능합니다.</p> <ul style="list-style-type: none"> • 데이터베이스 컬럼이 FOR BIT DATA로 표시된 경우 • PC/IXF 컬럼 1바이트 코드 페이지 값이 데이터베이스 컬럼의 SBCS CPGID와 동일하고 PC/IXF 컬럼 2바이트 코드 페이지 값이 0이거나 데이터베이스 컬럼의 DBCS CPGID와 동일할 경우 <p>데이터베이스 컬럼이 FOR BIT DATA로 표시될 경우, PC/IXF GRAPHIC, VARGRAPHIC 또는 LONG VARGRAPHIC 컬럼도 사용 가능합니다. 모든 경우에서, PC/IXF 컬럼이 고정 길이일 경우, 이 길이는 데이터베이스 컬럼의 최대 길이와 호환 가능해야 합니다. 294 페이지의 『FORCEIN 옵션』에서 자세한 내용을 참조하십시오.</p>
VARGRAPHIC (DBCS 전용)	<p>데이터베이스 컬럼의 최대 길이가 127 이하일 경우, PC/IXF VARGRAPHIC 컬럼이 작성됩니다. 데이터베이스 컬럼의 최대 길이가 127 보다 클 경우, PC/IXF LONG VARGRAPHIC 컬럼이 작성됩니다. 데이터베이스 컬럼의 최대 길이, SBCS CPGID 값 및 DBCS CPGID 값은 컬럼 설명자 레코드에 복사됩니다.</p>	<p>PC/IXF GRAPHIC, VARGRAPHIC 또는 LONG VARGRAPHIC 컬럼은 PC/IXF 컬럼 2바이트 코드페이지 값이 데이터베이스 컬럼의 값과 동일할 경우 사용 가능합니다. PC/IXF 컬럼이 고정 길이일 경우, 이 길이는 데이터베이스 컬럼의 최대 길이와 호환 가능해야 합니다. 294 페이지의 『FORCEIN 옵션』에서 자세한 내용을 참조하십시오.</p>

데이터베이스에 PC/IXF 파일 가져오기를 관리하는 일반 규칙

데이터베이스 관리 프로그램 가져오기 유틸리티는 SBCS 또는 DBCS 환경에서 PC/IXF 파일을 가져올 때 다음 일반 규칙을 적용합니다.

- 가져오기 유틸리티는 PC/IXF 형식 파일만 사용합니다(IXFHID = 'IXF'). 다른 형식의 IXF 파일은 가져올 수 없습니다.
- 가져오기 유틸리티는 1024 컬럼이 넘는 PC/IXF 파일을 거부합니다.
- PC/IXF H 레코드에 있는 IXFHSBCP의 값이 SBCS CPGID와 동일하거나 IXFHSBCP/IXFHDBCP와 목표 데이터베이스의 SBCS/DBCS CPGID 사이에 변환 테이블이 있어야 합니다. IXFHDBCP의 값은 '00000' 또는 목표 데이터베이스의 DBCS CPGID와 동일해야 합니다. 이들 조건 중 하나라도 충족되지 않은 경우, 가져오기 유틸리티는 294 페이지의 『FORCEIN 옵션』이 지정되지 않는 한 PC/IXF 파일을 거부합니다.

- 유효하지 않은 데이터 유형 — 새로운 테이블

새로운 테이블로 PC/IXF 파일 가져오기는 IMPORT 명령에 있는 CREATE 또는 REPLACE_CREATE 키워드로 지정됩니다. 유효하지 않은 데이터 유형(유효한 데이터 유형이 275 페이지의 『PC/IXF 데이터 유형』에 정의되어 있음)의 PC/IXF 컬럼을 새로운 테이블에 가져오기 위해 선택하면 가져오기 유틸리티가 종료됩니다. 모든 PC/IXF 파일이 거부되고, 테이블이 작성되지 않으며, 데이터가 가져오기되지 않습니다.

- 유효하지 않은 데이터 유형 — 기존 테이블

기존 테이블로 PC/IXF 파일 가져오기는 IMPORT 명령에 있는 INSERT, INSERT_UPDATE 또는 REPLACE_CREATE 키워드에 의해 지정됩니다. 유효하지 않은 데이터 유형의 PC/IXF 컬럼을 기존 테이블로 가져오기 위해 선택할 경우, 다음 두 조치 중 하나가 가능합니다.

- 목표 테이블 컬럼에 널(NULL) 입력이 가능할 경우, 유효하지 않은 PC/IXF 컬럼에 대한 모든 값이 무시되고 테이블 컬럼 값은 널(NULL)로 설정됩니다.
- 목표 테이블 컬럼에 널(NULL) 입력이 불가능하면, 가져오기 유틸리티는 종료됩니다. 모든 PC/IXF 파일이 거부되고 데이터는 가져오기되지 않습니다. 기존 테이블은 변경되지 않습니다.

- 새로운 테이블로 가져올 때, 널(NULL) 입력 가능 PC/IXF 컬럼은 널(NULL) 입력 가능 데이터베이스 컬럼을 생성하고, 널(NULL) 입력 불가능 PC/IXF 컬럼은 널(NULL) 입력이 불가능한 데이터베이스 컬럼을 생성합니다.
- 널(NULL) 입력 불가능 PC/IXF 컬럼은 널(NULL) 입력 가능 데이터베이스 컬럼에 가져올 수 있습니다.
- 널(NULL) 입력 가능 PC/IXF 컬럼은 널(NULL) 입력 불가능 데이터베이스 컬럼에 가져올 수 있습니다. PC/IXF 컬럼에서 널(NULL) 값이 발견되면, 가져오기 유틸리티는 널(NULL) 값이 있는 PC/IXF 행의 모든 컬럼 값을 거부하고 다음 PC/IXF 행으로 가서 처리를 계속합니다. 즉, 목표 테이블 컬럼(널에 대한)이 널(NULL) 입력 불가능일 경우, 널(NULL) 값이 들어 있는 PC/IXF 행에서는 데이터가 가져오기되지 않습니다.

- 호환되지 않는 컬럼 — 새로운 테이블

새로운 데이터베이스로 가져올 때 목표 데이터베이스 컬럼과 호환되지 않는 PC/IXF 컬럼이 선택되면, 가져오기 유틸리티가 종료됩니다. 모든 PC/IXF 파일이 거부되고, 테이블이 작성되지 않으며, 데이터가 가져오기되지 않습니다.

주: IMPORT 294 페이지의 『FORCEIN 옵션』은 호환 가능한 컬럼의 영역을 확장합니다.

- 호환되지 않는 컬럼 — 기존 테이블

기존 테이블로 가져올 때 목표 데이터베이스 컬럼과 호환되지 않는 PC/IXF 컬럼이 선택되면, 다음 두 가지 조치 중 하나가 가능합니다.

- 목표 테이블 컬럼이 널(NULL) 입력 가능하면, PC/IXF 컬럼에 대한 모든 값이 무시되고 테이블 컬럼 값이 널(NULL)로 설정됩니다.
- 목표 테이블 컬럼에 널(NULL) 입력이 불가능하면, 가져오기 유틸리티는 종료됩니다. 모든 PC/IXF 파일이 거부되고 데이터는 가져오기되지 않습니다. 기존 테이블은 변경되지 않습니다.

주: IMPORT 294 페이지의 『FORCEIN 옵션』은 호환 가능한 컬럼의 영역을 확장합니다.

- 유효하지 않은 값

IXF 파일 형식의 PC 버전

가져오기 도중, 목표 데이터베이스 컬럼에 대해 유효하지 않은 PC/IXF 컬럼 값이 발견되면, 가져오기 유틸리티는 유효하지 않은 값이 들어 있는 PC/IXF 행의 모든 컬럼 값을 거부하고(전체 행이 거부됨), 다음 PC/IXF 행에서 처리를 계속합니다.

- DBCS 데이터가 들어 있는 PC/IXF 파일을 가져오거나 로드할 때 상응하는 변환 파일(sqllib\conv에 위치함)이 클라이언트 머신에 설치되어 있어야 합니다. 이들 변환 파일의 이름은 소스와 목표 코드 페이지 번호를 모두 포함합니다. 확장자는 항상 .cnv입니다. 예를 들어, 파일 09320943.cnv는 코드 페이지 932를 943으로 변환하기 위한 변환 테이블을 포함하고 있습니다.

클라이언트 머신에 적합한 변환 파일이 들어 있지 않으면, 서버 머신에서 클라이언트 머신의 sqllib\conv 디렉토리로 복사할 수 있습니다. 반드시 호환 가능한 플랫폼으로부터 파일을 복사하도록 하십시오. 예를 들어, 클라이언트가 UNIX 기본 운영 체제에서 수행중이면, UNIX 기본 운영 체제에서 수행하는 서버에서도 파일을 복사하십시오.

데이터베이스로 PC/IXF 파일 가져오기를 관리하는 데이터 유형 특정 규칙

- 유효한 PC/IXF 숫자 컬럼은 호환적인 모든 숫자 데이터베이스 컬럼에 가져올 수 있습니다. 4바이트 부동 소수 데이터가 있는 PC/IXF 컬럼은 유효하지 않은 데이터 유형이기 때문에 가져오기되지 않습니다.
- 데이터베이스 날짜/시간 컬럼은 컬럼 길이와 값, 호환성, 제한조건에 따라, 일치된 PC/IXF 날짜/시간 컬럼(DATE, TIME, TIMESTAMP)의 값 뿐만 아니라, PC/IXF 문자 컬럼(CHAR, VARCHAR, LONG VARCHAR)의 값도 승인할 수 있습니다.
- 유효한 PC/IXF 문자 컬럼(CHAR, VARCHAR 또는 LONG VARCHAR)은 FOR BIT DATA로 표시된 기존 데이터베이스 문자 컬럼에 항상 가져올 수 있습니다. 그렇지 않으면,
 - IXFCSBCP와 SBCS CPGID가 일치해야 합니다.
 - IXFCSBCP/IXFCDBCP 및 SBCS/DBCS에 대한 변환 테이블이 있어야 합니다.
 - 하나의 세트가 모두 0이어야 합니다(FOR BIT DATA).

IXFCSBCP가 0이 아닐 경우, IXFCDBCP 값은 0이거나 목표 데이터베이스 컬럼의 DBCS CPGID와 동일해야 합니다.

이들 조건 중 하나라도 충족되지 않으면, PC/IXF와 데이터베이스 컬럼이 호환되지 않습니다.

유효한 PC/IXF 문자 컬럼을 새로운 데이터베이스 테이블로 가져올 때, IXFCSBCP의 값은 0 또는 데이터베이스의 SBCS CPGID와 동일하거나 변환 테이블이 있어야 합니다. IXFCSBCP가 0일 경우, IXFCDBCP도 0이 되어야 하며(그렇지 않으면, PC/IXF컬럼이 유효하지 않은 데이터 유형이 됨), IMPORT는 새로운 테이블에 FOR BIT DATA로 표시된 문자 컬럼을 작성합니다. IXFCSBCP가 0이 아니고, 데이터베이스의 SBCS CPGID와 동일할 경우, IXFCDBCP의 값은 0 또는 데이터베이스의 DBCS CPGID와 동일해야 합니다. 이 경우, 유틸리티는 새로운 테이블에 데이터베이스와 동일한 SBCS 및 DBCS CPGID 값을 갖는 문자 컬럼을 작성합니다. 이들 조건이 충족되지 않으면, PC/IXF와 데이터베이스 컬럼이 호환되지 않습니다.

294 페이지의 『FORCEIN 옵션』을 사용하여 코드 페이지 동일성 점검을 겹쳐 쓸 수 있습니다. 단, IXFCSBCP가 0이고 IXFCDBCP가 0이 아닌 PC/IXF 문자 컬럼은 유효하지 않는 데이터 유형이며, FORCEIN이 지정되어 있더라도 가져올 수 없습니다.

- 유효한 PC/IXF 그래픽 컬럼(GRAPHIC, VARGRAPHIC 또는 LONG VARGRAPHIC)은 FOR BIT DATA로 표시된 기존 데이터베이스 컬럼에 항상 가져올 수 있지만, 다른 모든 데이터베이스 컬럼과는 호환되지 않습니다. 294 페이지의 『FORCEIN 옵션』을 사용하여 이 제한사항을 완화시킬 수 있습니다. 단, IXFCSBCP가 0이 아니거나 IXFCDBCP가 0인 PC/IXF 그래픽 컬럼은 유효하지 않은 데이터 유형이며, FORCEIN이 지정되어 있더라도 가져올 수 없습니다.

유효한 PC/IXF 그래픽 컬럼을 데이터베이스 그래픽 컬럼으로 가져올 때, IXFCDBCP의 값은 목표 데이터베이스 컬럼의 DBCS CPGID와 동일해야 합니다(즉, 두 컬럼의 2바이트 코드 페이지가 일치해야 합니다).

- 기존 데이터베이스 테이블에 PC/IXF 파일을 가져오는 동안, 길이가 목표 컬럼의 최대 길이보다 큰 고정 길이 문자열 컬럼(CHAR 또는 GRAPHIC)이 선택 되면, 이들 컬럼은 서로 호환되지 않습니다.

IXF 파일 형식의 PC 버전

- 기존 데이터베이스 테이블에 PC/IXF 파일을 가져오는 동안, 길이가 목표 컬럼의 최대 길이보다 큰 가변 길이 문자열 컬럼(VARCHAR, LONG VARCHAR, VARGRAPHIC 또는 LONG VARGRAPHIC)이 선택되면, 이들 컬럼은 서로 호환됩니다. 각 값은 데이터베이스 관리 프로그램 INSERT문을 관리하는 호환성 규칙에 따라 처리되며, 목표 데이터베이스 컬럼에 대해 너무 긴 PC/IXF 값은 유효하지 않습니다.
- 고정 길이 데이터베이스 문자 컬럼(즉, CHAR 컬럼)에 가져온 PC/IXF 값은 데이터베이스 컬럼과 동일한 길이의 값을 확보하기 위해 필요할 경우, 오른쪽에 1바이트 공간(0x20)으로 채워집니다. 고정 길이 데이터베이스 그래픽 컬럼(즉, GRAPHIC 컬럼)에 가져온 PC/IXF 값은 데이터베이스 컬럼과 동일한 길이의 값을 확보하기 위해 필요할 경우, 오른쪽에 2바이트 공간(0x8140)으로 채워집니다.
- PC/IXF VARCHAR 컬럼이 254바이트의 최대 길이를 가지므로, 최대 길이 n 의 데이터베이스 VARCHAR 컬럼은 $254 < n < 4001$ 일 때 최대 길이 n 의 PC/IXF LONG VARCHAR 컬럼으로 가져와야 합니다.
- PC/IXF LONG VARCHAR 컬럼이 32 767바이트의 최대 길이를 가지며, 데이터베이스 LONG VARCHAR 컬럼이 32 700바이트의 최대 길이 제한을 가지더라도, 길이가 32 700바이트보다 큰 PC/IXF LONG VARCHAR 컬럼은 (32 768바이트 보다는 적은) 계속 유효하며, 데이터베이스 LONG VARCHAR 컬럼에 가져올 수 있습니다. 단, 데이터가 유실될 수 있습니다.
- PC/IXF VARGRAPHIC 컬럼이 127바이트의 최대 길이를 가지므로, 최대 길이 n 의 데이터베이스 VARGRAPHIC 컬럼은 $127 < n < 2001$ 일 때 최대 길이 n 의 PC/IXF LONG VARGRAPHIC 컬럼에 가져올 수 있습니다.
- PC/IXF LONG VARGRAPHIC 컬럼이 16 383바이트의 최대 길이를 가지고, 데이터베이스 LONG VARGRAPHIC 컬럼이 16 350의 최대 길이 제한을 가지더라도, 길이가 16 350바이트보다 큰 PC/IXF LONG VARGRAPHIC 컬럼은 (16 384바이트 보다는 적은) 계속 유효하며, 데이터베이스 LONG VARGRAPHIC 컬럼에 가져올 수 있습니다. 단, 데이터가 유실될 수 있습니다.

293 페이지의 표14에는 FORCEIN 옵션을 사용하지 않고 새로운 또는 기존 데이터베이스 테이블에 PC/IXF 파일을 가져오는 것을 요약한 내용이 있습니다.

표 14. FORCEIN 옵션을 사용하지 않은 PC/IXF 파일 가져오기의 요약

PC/IXF 컬럼 데이터 유형	데이터베이스 컬럼 데이터 유형											
	숫자					문자			그래프	날짜 시간		
	SMALL INT	INT	BIGINT	DEC	FLT	(0,0)	(SBCS, 0) ^d	(SBCS, DBCS) ^b	^b	DATE	TIME	TIME STAMP
숫자												
-SMALLINT	N											
	E	E	E	E ^a	E							
-INTEGER		N										
	E ^a	E	E	E ^a	E							
-BIGINT			N									
	E ^a	E ^a	E	E ^a	E							
-DECIMAL				N								
	E ^a	E ^a	E ^a	E ^a	E							
-FLOAT					N							
	E ^a	E ^a	E ^a	E ^a	E							
문자												
-(0,0)						N						
						E				E ^c	E ^c	E ^c
-(SBCS,0)							N	N				
						E	E	E		E ^c	E ^c	E ^c
-(SBCS, DBCS)								N		E ^c	E ^c	E ^c
						E		E				
그래픽												
									N			
						E			E			
날짜시간												
-DATE										N		
										E		
-TIME											N	
											E	
-TIME STAMP												N
												E

IXF 파일 형식의 PC 버전

표 14. FORCEIN 옵션을 사용하지 않은 PC/IXF 파일 가져오기의 요약 (계속)

PC/IXF 컬럼 데이터 유형	데이터베이스 컬럼 데이터 유형											
	숫자					문자			그래프	날짜 시간		
	SMALL INT	INT	BIGINT	DEC	FLT	(0,0)	(SBCS, 0) ^d	(SBCS, DBCS) ^b	^b	DATE	TIME	TIME STAMP
<p>주:</p> <p>1. 이 표는 모든 유효한 PC/IXF 및 데이터베이스 관리 프로그램 데이터 유형의 행렬입니다. PC/IXF 컬럼이 데이터베이스 컬럼에 가져오기될 수 있는 경우, PC/IXF 데이터 유형 행렬 행과 데이터베이스 관리 프로그램 데이터 유형 행렬 컬럼이 교차하는 행렬 셀에 문자가 표시됩니다. 'N'은 유틸리티가 새로운 데이터베이스 테이블을 작성하는 것을 나타냅니다(지정된 데이터 유형의 데이터베이스 컬럼이 작성됩니다). 'E'는 유틸리티가 기존 데이터베이스 테이블에 데이터를 가져오는 것을 나타냅니다(지정된 데이터 유형의 데이터베이스 컬럼이 유효한 목표입니다).</p> <p>2. 문자열 데이터 유형은 코드 페이지 속성에 의해 구별됩니다. 이들 속성은 다음과 같이 정렬된 쌍(SBCS,DBCS)으로 표시됩니다.</p> <ul style="list-style-type: none"> SBCS는 0이거나 문자 데이터 유형의 1바이트 코드 페이지 속성의 0 이외의 값을 표시합니다. DBCS는 0이거나 문자 데이터 유형의 2바이트 코드 페이지 속성의 0 이외의 값을 표시합니다. <p>3. 테이블이 PC/IXF 문자 컬럼이 데이터베이스 문자 컬럼에 가져오기될 수 있다고 지정하면, 각 코드 페이지 속성 쌍의 값은 코드 페이지 동일성을 관리하는 규칙을 충족시킵니다.</p> <p>^a 각 값은 목표 숫자 데이터 유형에 대한 범위에 있지 않을 경우 거부됩니다.</p> <p>^b 데이터 유형은 DBCS 환경에서만 사용 가능합니다.</p> <p>^c 각 값은 유효한 날짜 또는 시간 값이 아닐 경우 거부됩니다.</p> <p>^d 데이터 유형이 DBCS 환경에서 사용 불가능합니다.</p>												

FORCEIN 옵션

FORCEIN 옵션은 PC/IXF 파일에 있는 데이터와 목표 데이터베이스간의 코드 페이지 차이점에도 불구하고 PC/IXF 파일의 가져오기를 허용합니다. 호환 가능한 컬럼의 정의에 추가 융통성을 부여합니다.

FORCEIN 일반 의미론

다음 일반 의미론은 SBCS 또는 DBCS 환경에서 FORCEIN 옵션을 사용할 때 적용됩니다.

- FORCEIN 옵션은 신중하게 사용되어야 합니다. 일반적으로, 이 옵션을 사용하지 않고 가져오기를 시도하는 것이 좋습니다. 단, PC/IXF 데이터 교환 아키텍처의 일반적 특성 때문에, 일부 PC/IXF 파일은 조정되어야만 가져올 수 있는 데이터 유형이나 값을 가지고 있을 수 있습니다.

- FORCEIN을 사용하여 새로운 테이블로 가져올 경우, 기존 테이블에 가져오는 경우와 다른 결과를 낼 수 있습니다. 기존 테이블에는 각 PC/IXF 데이터 유형에 대해 사전 정의된 목표 데이터 유형이 있습니다.
- LOB 데이터가 LOBSINFILE 옵션을 사용하여 내보내지고, 파일이 다른 코드 페이지를 사용하는 다른 클라이언트로 이동하면, 다른 데이터와 달리, 각기 다른 파일에 있는 CLOB와 DBCLOBS는 데이터베이스에 가져오거나 로드될 때 클라이언트 코드 페이지로 변환되지 않습니다.

FORCEIN 코드 페이지 의미론

다음 코드 페이지 의미론은 SBCS 또는 DBCS 환경에서 FORCEIN 옵션을 사용할 때 적용됩니다.

- FORCEIN 옵션은 모든 가져오기 유틸리티 코드 페이지 비교를 불가능하게 합니다.

이 규칙은 기존 또는 새로운 데이터베이스 테이블에 가져올 때, 컬럼 레벨 및 파일 레벨에서의 코드 페이지 비교에 대해 적용됩니다. 컬럼(예: 데이터 유형) 레벨의 경우, 이 규칙은 다음 데이터베이스 관리 프로그램 및 PC/IXF 데이터 유형 즉, 문자(Char, VARCHAR, LONG VARCHAR) 및 그래픽(Graphic, VARGRAPHIC, LONG VARGRAPHIC)에만 적용됩니다. 이러한 제한점은 다른 데이터 유형의 코드 페이지 속성이 데이터 유형 값의 해석에 타당하지 않다는 사실에 따른 것입니다.

- FORCEIN 옵션은 데이터 유형을 판별하기 위한 코드 페이지 속성 조사를 불가능하게 하지 않습니다.

예를 들어, 데이터베이스 관리 프로그램은 CHAR 컬럼이 FOR BIT DATA 속성을 갖도록 선언될 수 있게 합니다. 이러한 선언문은 컬럼의 SBCS CPGID와 DBCS CPGID를 모두 0으로 설정합니다. 이것은 컬럼 값을 비트 문자열(문자 열보다는)로 식별하는 이들 CPGID의 0 값입니다.

- FORCEIN 옵션은 코드 페이지 변환을 포함하지 않습니다.

FORCEIN 옵션에 대해 민감한 데이터 유형 값은 "그대로" 복사합니다. 코드 페이지 환경의 변화를 나타내기 위해 코드 페이지 맵핑이 사용되지 않습니다. 가져온 값을 공간으로 채우는 일은 고정 길이 목표 컬럼의 경우에 필요할 수 있습니다.

- FORCEIN 옵션을 사용하여 기존 테이블로 데이터를 가져오면

IXF 파일 형식의 PC 버전

- 목표 데이터베이스 테이블과 컬럼의 코드 페이지 값이 항상 사용됩니다.
- PC/IXF 파일과 컬럼의 코드 페이지 값은 무시됩니다.

이 규칙은 FORCEIN 옵션 사용과 관계없이 적용됩니다. 데이터베이스 관리 프로그램은 데이터베이스가 작성된 후에는 데이터베이스 또는 컬럼 코드 페이지 값을 변경할 수 없게 합니다.

- FORCEIN 옵션을 사용하여 새로운 테이블로 가져오면
 - 목표 데이터베이스의 코드 페이지 값이 사용됩니다.
 - IXFCSBCP = IXFCDBCP = 0인 PC/IXF 문자 컬럼은 FOR BIT DATA 라고 표시된 테이블 컬럼을 생성합니다.
 - 다른 모든 PC/IXF 문자 컬럼은 데이터베이스의 값과 동일한 SBCS 및 DBCS CPGID 값을 갖는 테이블 문자 컬럼을 생성합니다.
 - PC/IXF 그래픽 컬럼은 "정의되지 않은" SBCS CPGID 및 데이터베이스의 값과 동일한 DBCS CPGID 값(DBCS 환경에서만)을 갖는 테이블 그래픽 컬럼을 생성합니다.

FORCEIN 예

IXFCSBCP = '00897' 및 IXFCDBCP = '00301'인 PC/IXF CHAR 컬럼을 고려하십시오. SBCS CPGID = '00850' 및 DBCS CPGID = '00000'인 데이터베이스 CHAR 컬럼으로 이 컬럼을 가져옵니다. FORCEIN 옵션을 사용하지 않을 경우, 유틸리티는 종료되며, 데이터는 가져오지 않거나 PC/IXF 컬럼 값이 무시됩니다. 그리고 데이터베이스 컬럼에는 널(NULL)이 들어갑니다(데이터베이스 컬럼이 널(NULL) 입력 가능일 경우). FORCEIN을 사용하면, 유틸리티는 코드 페이지 비호환성을 무시하고 계속 처리합니다. 다른 데이터 유형 비호환성(예: 길이)이 없을 경우, PC/IXF 컬럼의 값은 "그대로" 가져오고 데이터베이스 컬럼 코드 페이지 환경에서 해석할 수 있게 됩니다.

다음에 나와 있는 테이블은 다음 내용을 나타냅니다.

- 지정된 코드 페이지 속성이 있는 PC/IXF 파일 데이터 유형을 가져올 때 새로운 데이터베이스 테이블에 작성된 컬럼의 코드 페이지 속성
- PC/IXF 데이터 유형이 유효하지 않거나 호환되지 않을 경우, 가져오기 유틸리티는 이를 거부합니다.

표 15. 가져오기 유틸리티 코드 페이지 의미론의 요약(새로운 테이블). 이 테이블에서는 a와 x 사이에 변환 테이블이 존재하지 않음을 가정합니다. 테이블이 있으면, 항목 3과 4가 FORCEIN 옵션을 사용하지 않아도 됩니다.

PC/IXF 데이터 유형의 코드 페이지 속성	데이터베이스 테이블 컬럼의 코드 페이지 속성	
	FORCEIN을 사용하지 않을 경우	FORCEIN을 사용할 경우
SBCS		
(0,0)	(0,0)	(0,0)
(a,0)	(a,0)	(a,0)
(x,0)	거부	(a,0)
(x,y)	거부	(a,0)
(a,y)	거부	(a,0)
(0,y)	거부	(0,0)
DBCS		
(0,0)	(0,0)	(0,0)
(a,0)	(a,b)	(a,b)
(x,0)	거부	(a,b)
(a,b)	(a,b)	(a,b)
(x,y)	거부	(a,b)
(a,y)	거부	(a,b)
(x,b)	거부	(a,b)
(0,b)	(-,b)	(-,b)
(0,y)	거부	(-,b)

IXF 파일 형식의 PC 버전

표 15. 가져오기 유틸리티 코드 페이지 의미론의 요약(새로운 테이블) (계속). 이 테이블에서는 a와 x 사이에 변환 테이블이 존재하지 않음을 가정합니다. 테이블이 있으면, 항목 3과 4가 FORCEIN 옵션을 사용하지 않아도 됩니다.

PC/IXF 데이터 유형의 코드 페이지 속성	데이터베이스 테이블 컬럼의 코드 페이지 속성	
	FORCEIN을 사용하지 않을 경우	FORCEIN을 사용할 경우
<p>주:</p> <ol style="list-style-type: none"> 1. PC/IXF 데이터 유형의 코드 페이지 속성은 정렬된 쌍으로 표시됩니다. 여기서 x는 0이 아닌 1 바이트 코드 페이지 값을 나타내고, y는 0이 아닌 2바이트 코드 페이지 값을 나타냅니다. '-'은 정의되지 않은 코드 페이지 값을 나타냅니다. 2. 여러 코드 페이지 속성 쌍에 다른 문자를 사용하는 것은 고려해 보아야 합니다. 다른 문자는 다른 값을 의미합니다. 예를 들어, PC/IXF 데이터 유형이 (x,y)로 표시되고, 데이터베이스 컬럼은 (a,y)로 표시된 경우, x는 a와 같지 않습니다. 그러나 PC/IXF 파일과 데이터베이스는 동일한 2바이트 코드 페이지 값 y를 가지고 있습니다. 3. 문자 및 그래픽 데이터 유형만이 FORCEIN 코드 페이지 의미론의 영향을 받습니다. 4. 새로운 테이블이 들어 있는 데이터베이스는 (a,0)의 코드 페이지 속성을 가지는 것으로 가정됩니다. 따라서, 새로운 테이블에 있는 모든 문자 컬럼은 (0,0) 또는 (a,0)의 코드 페이지 속성을 가져야 합니다. DBCS 환경에서는 새로운 테이블이 들어 있는 데이터베이스는 (a,b)의 코드 페이지 속성을 가지는 것으로 가정됩니다. 따라서, 새로운 테이블에 있는 모든 그래픽 컬럼은 (-,b)의 코드 페이지 속성을 가져야 하고, 모든 문자 컬럼은 (a,b)의 코드 페이지 속성을 가져야 합니다. SBCS CPGID는 그래픽 데이터 유형에 대해 정의되어 있지 않기 때문에 '-'으로 표시됩니다. 5. 데이터 유형의 결과는 300 페이지의 『FORCEIN 데이터 유형 의미론』에 기술된 규칙에 의해 결정됩니다. 6. 거부 결과는 유효하지 않거나 호환되지 않는 데이터 유형에 대한 규칙의 결과입니다(288 페이지의 『데이터베이스에 PC/IXF 파일 가져오기를 관리하는 일반 규칙』 참조). 		

다음에 나와 있는 테이블은 다음 내용을 나타냅니다.

- 가져오기 유틸리티가 다양한 코드 페이지 속성을 갖는 PC/IXF 데이터 유형을 지정된 코드 페이지 속성을 갖는 기존 테이블 컬럼(목표 컬럼)에 가져오는 것을 허용합니다.

- 가져오기 유틸리티가 특정 코드 페이지 속성을 갖는 PC/IXF 데이터 유형이 표시된 코드 페이지 속성을 갖는 기존 테이블 컬럼에 가져오는 것을 허용하지 않습니다. PC/IXF 데이터 유형이 유효하지 않거나 호환되지 않을 경우, 가져오기 유틸리티는 이를 거부합니다.

표 16. 가져오기 유틸리티 코드 페이지 의미론의 요약(기존 테이블). 이 테이블에서는 a와 x 사이에 변환 테이블이 존재하지 않음을 가정합니다.

PC/IXF 데이터 유형의 코드 페이지 속성	목표 데이터베이스 컬럼의 코드 페이지 속성	가져오기 결과	
		FORCEIN을 사용하지 않을 경우	FORCEIN을 사용할 경우
SBCS			
(0,0)	(0,0)	허용	허용
(a,0)	(0,0)	허용	허용
(x,0)	(0,0)	허용	허용
(x,y)	(0,0)	허용	허용
(a,y)	(0,0)	허용	허용
(0,y)	(0,0)	허용	허용
(0,0)	(a,0)	NULL 또는 거부	허용
(a,0)	(a,0)	허용	허용
(x,0)	(a,0)	NULL 또는 거부	허용
(x,y)	(a,0)	NULL 또는 거부	허용
(a,y)	(a,0)	NULL 또는 거부	허용
(0,y)	(a,0)	NULL 또는 거부	NULL 또는 거부
DBCS			
(0,0)	(0,0)	허용	허용
(a,0)	(0,0)	허용	허용
(x,0)	(0,0)	허용	허용
(a,b)	(0,0)	허용	허용
(x,y)	(0,0)	허용	허용
(a,y)	(0,0)	허용	허용
(x,b)	(0,0)	허용	허용
(0,b)	(0,0)	허용	허용
(0,y)	(0,0)	허용	허용

IXF 파일 형식의 PC 버전

표 16. 가져오기 유틸리티 코드 페이지 의미론의 요약(기존 테이블) (계속). 이 테이블에서는 a와 x 사이에 변환 테이블이 존재하지 않음을 가정합니다.

PC/IXF 데이터 유형 의 코드 페이지 속성	목표 데이터베이스 컬 럼의 코드 페이지 속성	가져오기 결과	
		FORCEIN을 사용하 지 않을 경우	FORCEIN을 사용할 경우
(0,0)	(a,b)	NULL 또는 거부	허용
(a,0)	(a,b)	허용	허용
(x,0)	(a,b)	NULL 또는 거부	허용
(a,b)	(a,b)	허용	허용
(x,y)	(a,b)	NULL 또는 거부	허용
(a,y)	(a,b)	NULL 또는 거부	허용
(x,b)	(a,b)	NULL 또는 거부	허용
(0,b)	(a,b)	NULL 또는 거부	NULL 또는 거부
(0,y)	(a,b)	NULL 또는 거부	NULL 또는 거부
(0,0)	(-,b)	NULL 또는 거부	허용
(a,0)	(-,b)	NULL 또는 거부	NULL 또는 거부
(x,0)	(-,b)	NULL 또는 거부	NULL 또는 거부
(a,b)	(-,b)	NULL 또는 거부	NULL 또는 거부
(x,y)	(-,b)	NULL 또는 거부	NULL 또는 거부
(a,y)	(-,b)	NULL 또는 거부	NULL 또는 거부
(x,b)	(-,b)	NULL 또는 거부	NULL 또는 거부
(0,b)	(-,b)	허용	허용
(0,y)	(-,b)	NULL 또는 거부	허용

주:

1. 297 페이지의 표15에 대한 주를 참조하십시오.
2. NULL 또는 거부 결과는 유효하지 않거나 호환되지 않는 데이터 유형에 대한 규칙의 결과입니다(288 페이지의 『데이터베이스에 PC/IXF 파일 가져오기를 관리하는 일반 규칙』 참조).

FORCEIN 데이터 유형 의미론

FORCEIN 옵션은 특정 PC/IXF 컬럼을 동일하지 않거나 호환되지 않는 데이터 유형의 목표 데이터베이스 컬럼으로 가져올 수 있게 합니다. 다음 데이터 유형 의미론은 SBCS 또는 DBCS 환경에서 FORCEIN 옵션을 사용할 때 적용됩니다(주석이 표시된 곳은 제외).

- SBCS 환경에서, FORCEIN 옵션은 다음을 가져올 수 있도록 합니다.
 - PC/IXF BIT 데이터 유형(PC/IXF 문자 컬럼에 대해 IXFCBCP = 0 = IXFCDBC)을 데이터베이스 문자 컬럼(0이 아닌 SBCS CPGID 및 DBCS CPGID = 0)에 기존 테이블만 허용.
 - PC/IXF MIXED 데이터 유형(0이 아닌 IXFCBCP 및 IXFCDBC)을 데이터베이스 문자 컬럼에 기존 및 새로운 테이블 모두 허용.
 - PC/IXF GRAPHIC 데이터 유형을 데이터베이스 FOR BIT DATA 컬럼 (SBCS CPGID = 0 = DBCS CPGID)에 새로운 테이블만 허용(기존 테이블에는 항상 허용됨).
- FORCEIN 옵션은 유효한 PC/IXF 데이터 유형의 영역을 확장하지 않습니다. 275 페이지의 『PC/IXF 데이터 유형』에서 유효한 것으로 정의되어 있지 않는 데이터 유형이 있는 PC/IXF 컬럼은 FORCEIN 옵션을 사용 또는 사용하지 않은 가져오기에 대해 유효하지 않습니다.
- DBCS 환경에서, FORCEIN 옵션은 다음을 가져올 수 있도록 합니다.
 - PC/IXF BIT 데이터 유형을 데이터베이스 문자 컬럼에 가져오기.
 - PC/IXF BIT 데이터 유형을 데이터베이스 그래픽 컬럼에 가져오기. 단, PC/IXF BIT 컬럼이 고정 길이일 경우, 이 길이는 짝수여야 합니다. 홀수 길이로 된 고정 길이 PC/IXF BIT 컬럼은 데이터베이스 그래픽 컬럼과 호환되지 않습니다. 가변 길이 컬럼의 홀수 길이 값이 데이터베이스 그래픽 컬럼에 가져오는 데 유효하지 않은 값이기는 하지만, 가변 길이 PC/IXF BIT 컬럼은 길이가 짝수 또는 홀수이든 상관없이 호환됩니다.
 - PC/IXF MIXED 데이터 유형을 데이터베이스 문자 컬럼에 가져오기.

표17에서는 FORCEIN 옵션을 사용하여 PC/IXF 파일을 새로운 또는 기존 데이터베이스 테이블에 가져오는 것을 요약합니다.

표 17. FORCEIN을 사용한 PC/IXF 파일 가져오기의 요약

PC/IXF 컬럼 데이터 유형	데이터베이스 컬럼 데이터 유형											
	숫자					문자			그래프	날짜 시간		
	SMALL INT	INT	BIGINT	DEC	FLT	(0,0)	(SBCS, 0) ^c	(SBCS, DBCS) ^b	^b	DATE	TIME	TIME STAMP
숫자												
-SMALLINT	N											

IXF 파일 형식의 PC 버전

표 17. FORCEIN을 사용한 PC/IXF 파일 가져오기의 요약 (계속)

PC/IXF 컬럼 데이터 유형	데이터베이스 컬럼 데이터 유형											
	숫자					문자			그래프	날짜 시간		
	SMALL INT	INT	BIGINT	DEC	FLT	(0,0)	(SBCS, 0) ^e	(SBCS, DBCS) ^b	^b	DATE	TIME	TIME STAMP
	E	E	E	E ^a	E							
-INTEGER		N										
	E ^a	E	E	E ^a	E							
-BIGINT			N									
	E ^a	E ^a	E	E ^a	E							
-DECIMAL				N								
	E ^a	E ^a	E ^a	E ^a	E							
-FLOAT					N							
	E ^a	E ^a	E ^a	E ^a	E							
문자												
-(0,0)						N						
						E	E w/F	E w/F	E w/F	E ^c	E ^c	E ^c
-(SBCS,0)							N	N				
						E	E	E		E ^c	E ^c	E ^c
-(SBCS, DBCS)							N w/F ^d	N		E ^c	E ^c	E ^c
						E	E w/F	E				
그래픽												
						N w/F ^d			N			
						E			E			
날짜시간												
-DATE										N		
										E		
-TIME											N	
											E	
-TIME STAMP												N
												E

표 17. FORCEIN을 사용한 PC/IXF 파일 가져오기의 요약 (계속)

PC/IXF 컬럼 데이터 유형	데이터베이스 컬럼 데이터 유형											
	숫자					문자			그래프	날짜 시간		
	SMALL INT	INT	BIGINT	DEC	FLT	(0,0)	(SBCS, 0) ^c	(SBCS, DBCS) ^b	'b'	DATE	TIME	TIME STAMP
<p>주: PC/IXF 컬럼을 FORCEIN 옵션만 사용하여 데이터베이스 컬럼으로 가져올 수 있을 경우, 문자열 'w/F'가 'N' 또는 'E'와 함께 표시됩니다. 'N'은 유틸리티가 새로운 데이터베이스를 작성중임을 나타내고, 'E'는 유틸리티가 기존 데이터베이스 테이블에 데이터를 가져오는 중임을 나타냅니다. FORCEIN 옵션은 문자와 그래픽 데이터 유형의 호환성에만 영향을 줍니다.</p> <p>^a 각 값은 목표 숫자 데이터 유형에 대한 범위에 있지 않을 경우 거부됩니다.</p> <p>^b 데이터 유형은 DBCS 환경에서만 사용 가능합니다.</p> <p>^c 각 값은 유효한 날짜 또는 시간 값이 아닐 경우 거부됩니다.</p> <p>^d 소스 PC/IXF 데이터 유형이 목표 데이터베이스에서 지원되지 않을 경우에만 적용됩니다.</p> <p>^e 데이터 유형이 DBCS 환경에서 사용 불가능합니다.</p>												

PC/IXF와 버전 0 System/370 IXF간의 차이점

다음 사항에서는 데이터베이스 관리 프로그램에 의해 사용되는 PC/IXF와 여러 호스트 데이터베이스 제품에 의해 사용되는 버전 0 System/370 IXF간의 차이점에 대해 설명합니다.

- PC/IXF 파일은 EBCDIC가 아닌 ASCII 파일입니다. PC/IXF 파일은 확장된 코드 페이지를 가지고 있으며, H 레코드에 새로운 코드 페이지 식별자를 포함하며, 컬럼 설명자 레코드에서 실제 코드 페이지 값을 사용합니다. 또한, FOR BIT DATA와 같은 문자 데이터의 컬럼을 표시하기 위한 메커니즘도 있습니다. FOR BIT DATA 컬럼은 특별한 의미가 있습니다. 왜냐하면, PC/IXF 파일 형식을 다른 IXF나 데이터베이스 파일 형식으로 변환 또는 이 반대로 변환하는 변환 규칙이 FOR BIT DATA 컬럼에 있는 값에 대해서는 어떠한 코드 페이지 변환도 수행할 수 없기 때문입니다.
- 머신 데이터 형식만 허용됩니다. 즉, IXFTFORM 필드는 항상 M 값을 포함해야 합니다. 그밖에, 머신 데이터는 PC 형식으로 되어야 합니다. 즉, IXFTMFRM 필드는 PC 값을 포함해야 합니다. 이는 PC/IXF 데이터 레코드의 데이터 부분에 있는 정수, 부동 소수 및 십진수가 PC 형식으로 되어야 하는 것을 의미합니다.

IXF 파일 형식의 PC 버전

- 응용프로그램(A) 레코드는 PC/IXF 파일에서 H 레코드 다음에 어느 곳에나 들어 갈 수 있습니다. 이 레코드는 IXFHHCNT 필드 값이 계산될 때 카운트되지 않습니다.
- 모든 PC/IXF 레코드는 레코드 길이 식별자를 사용하여 시작합니다. 이는 레코드 길이 표시기를 포함하지 않는 PC/IXF 레코드의 길이(바이트)를 포함하는 정수 값의 6바이트 문자 표현입니다. 즉, 전체 레코드 길이에서 6바이트를 뺍니다. 이 레코드 길이 필드의 목적은 PC 프로그램이 레코드 경계를 식별할 수 있도록 하는 것입니다.
- 가변 길이 데이터의 압축 저장영역을 촉진하고, 데이터가 여러 레코드로 파티션될 때 복잡한 처리를 피하기 위해, PC/IXF는 버전 0 IXF X 레코드를 지원하지 않으나 D 레코드 식별자를 지원합니다. 가변 길이 필드나 널(NULL) 입력 가능 필드가 데이터 D 레코드의 마지막 필드가 될 때마다, 이 필드의 전체 최대 길이를 PC/IXF 파일에 기록할 필요는 없습니다.

워크시트 파일 형식(WSF)

Lotus 1-2-3과 Symphony 제품은 각 새로운 릴리스에 추가적인 기능이 추가된 동일한 기본 형식을 사용합니다. 데이터베이스 관리 프로그램은 모든 Lotus 제품에 대해 동일한 워크시트 레코드의 부속 집합을 지원합니다. 즉, 데이터베이스 관리 프로그램이 지원하는 Lotus 1-2-3 및 Symphony 제품의 릴리스에 대해 세 문자 확장자를 갖는 모든 파일 이름이 사용 가능합니다(예: WKS, WK1, WRK, WR1, WJ2).

각 WSF 파일은 하나의 워크시트를 나타냅니다. 데이터베이스 관리 프로그램은 다음 규칙을 사용하여 워크시트를 해석하고 내보내기 조작으로 생성된 워크시트가 일관성을 갖도록 합니다.

- 첫번째 행(ROW 값 0)에 있는 셀은 전체 워크시트에 대한 설명적 정보를 위해 예약됩니다. 이 행에 있는 모든 데이터는 선택적이며, 가져오는 동안 무시됩니다.
- 두번째 행(ROW 값 1)에 있는 셀은 컬럼 레이블을 위해 사용됩니다.
- 나머지 행은 데이터 행입니다(레코드 또는 테이블의 데이터 행).

- 모든 컬럼 헤딩 아래 있는 셀 값은 해당 특정 컬럼 또는 필드에 대한 값입니다.
- 널(NULL) 값은 셀 내용 레코드의 행에 있는 특정 컬럼에 대한 실제 셀 내용 레코드(예: 비 정수, 번호, 레이블 또는 공식 레코드)가 없을 때 지정됩니다.

주: 널(NULL) 행은 가져오거나 내보내지 않습니다.

내보내기 조작중 WSF 형식에 따른 파일을 작성하려면, 일부 데이터가 유실될 수도 있습니다.

WSF 파일은 DB2가 지원하는 기존 코드 페이지와 동일할 필요가 없는 Lotus 코드 포인트 매핑을 사용합니다. 결과적으로, WSF 파일을 가져오거나 내보낼 때, 데이터는 Lotus 코드 포인트로부터 응용프로그램 코드 페이지에 의해 사용된 코드 포인트로 변환되거나 반대 방향으로 변환됩니다. DB2는 Lotus 코드 포인트와 코드 페이지 437, 819, 850, 860, 863, 865가 정의한 코드 포인트간의 변환을 지원합니다.

주: 다중 바이트 문자 세트 사용자의 경우, 변환이 수행되지 않습니다.

워크시트 파일 형식(WSF)

부록D. 경고, 오류 및 완료 메시지

여러 가지 데이터 이동 유틸리티에 의해 생성된 메시지는 SQL 메시지 중에 포함됩니다. 이러한 메시지는 경고나 오류 조건 검출시 데이터베이스 관리 프로그램에 의해 생성됩니다. 각 메시지에는 접두부(SQL)와 4 또는 5 자릿수의 메시지 번호로 구성된 메시지 식별자가 들어 있습니다. 메시지에는 세 가지 유형이 있는데 통지, 경고 및 중요한 메시지 유형입니다. N으로 끝나는 메시지 식별자는 오류 메시지를 나타냅니다. W로 끝나는 메시지 식별자는 경고 또는 정보용 메시지를 나타냅니다. C로 끝나는 메시지 식별자는 중요한 시스템 오류를 나타냅니다.

메시지 번호는 *SQLCODE*로도 참조됩니다. *SQLCODE*는 메시지 유형(N, W, C)에 따라 양수 또는 음수로 응용프로그램에 보내집니다. N과 C는 음수 값을 내며, W는 양수 값을 냅니다. DB2가 응용프로그램에 *SQLCODE*를 리턴하므로, 응용프로그램이 *SQLCODE*가 있는 메시지를 받을 수 있습니다. DB2는 또한 SQL문의 결과가 될 수 있는 조건에 대한 *SQLSTATE* 값도 리턴합니다. 일부 *SQLCODE* 값은 연관된 *SQLSTATE* 값을 갖습니다.

메시지 참조서에서 모든 DB2 메시지에 관한 자세한 정보를 참조하십시오. 이 책에 있는 정보를 사용하여 오류나 문제점을 식별하고, 적합한 복구 조치를 사용하여 문제점을 해결할 수 있습니다. 이 정보는 메시지가 생성되고 기록되는 위치를 이해하는 데에도 사용할 수 있습니다.

SQL 메시지 및 *SQLSTATE* 값과 연관된 메시지 텍스트는 운영 체제 명령행에서도 액세스할 수 있습니다. 이러한 오류 메시지의 도움말에 액세스하려면, 운영 체제 명령 프롬프트에서 다음과 같이 입력하십시오.

```
db2 ? SQLnnnnn
```

여기서, *nnnnn*은 메시지 번호를 나타냅니다.

db2 명령에 대한 매개변수로 사용되는 메시지 식별자는 대소문자를 구별하고, 끝나는 문자를 필요로 하지 않습니다. 따라서, 다음 명령은 같은 결과를 내게 됩니다.

메시지

```
db2 ? SQL0000N
db2 ? sq10000
db2 ? SQL0000n
```

메시지 텍스트가 화면에서 너무 긴 경우, 다음 명령을 사용하십시오(UNIX 기본 운영 체제 및 "more" 파이프를 지원하는 기타 운영 체제에서).

```
db2 ? SQLnnnnn | more
```

또한, 출력을 파일로 경로 재지정하여 열람되게 할 수 있습니다.

도움말은 대화식 입력 모드에서도 호출할 수 있습니다. 이러한 모드로 액세스하려면, 운영 체제 명령 프롬프트에서 다음을 입력하십시오.

```
db2
```

이 모드로 DB2 메시지 도움말을 보려면, 명령 프롬프트(db2 =>)에서 다음을 입력하십시오.

```
? SQLnnnnn
```

SQLSTATE와 연관된 메시지 텍스트는 다음 명령으로 검색될 수 있습니다.

```
db2 ? nnnnn
또는
db2 ? nn
```

여기서, *nnnnn*은 5자리 문자의 SQLSTATE 값(영숫자)이고, *nn*은 두 자릿수의 SQLSTATE 클래스 코드(SQLSTATE 값의 처음 두 자릿수)입니다.

부록E. DB2 라이브러리 사용

DB2 Universal Database 라이브러리는 온라인 도움말, 책(PDF 및 HTML) 및 샘플 프로그램이 HTML 형식으로 구성됩니다. 이 절에서는 제공되는 정보 및 액세스하는 방법을 설명합니다.

온라인으로 제품 정보에 액세스하려면 정보 센터를 사용하면 됩니다. 324 페이지의 『정보 센터로 정보에 액세스』에서 자세한 내용을 참조하십시오. 타스크 정보, DB2 책, 문제점 해결 정보, 샘플 프로그램, 웹에 있는 DB2 정보를 볼 수 있습니다.

DB2 PDF 파일 및 인쇄된 책

DB2 정보

다음의 테이블은 DB2 책을 4개의 범주로 나눕니다.

DB2 안내 및 참조 정보

이 책에는 모든 플랫폼에 공통적인 DB2 정보가 들어 있습니다.

DB2 설치 및 구성 정보

이 책에는 특정 플랫폼의 DB2를 위한 것입니다. 예를 들어, OS/2, Windows 및 UNIX 플랫폼에서의 DB2용으로 각각 다른 빠른 시작 책이 있습니다.

플랫폼간 샘플 프로그램(HTML)

이 샘플들은 응용프로그램 개발 클라이언트와 함께 설치된 샘플 프로그램의 HTML 버전입니다. 이들은 단지 정보용으로서 실제 프로그램을 대체하지는 않습니다.

릴리스 정보

이 파일에는 DB2 책에 포함될 수 없었던 최신 정보가 포함되어 있습니다.

설치 매뉴얼, 릴리스 정보 및 지습서는 제품 CD-ROM의 HTML 디렉토리에서 볼 수 있습니다. 대부분의 책은 단지 보기용으로 제품 CD-ROM에서 HTML 형식으로 제공되고 보기와 인쇄용으로 제품 CD-ROM에서 PDF 형식으로 제공됩니다. 또한 IBM에서 인쇄된 책을 주문하려면 319 페이지의 『인쇄된 책 주문』에서 자세한 내용을 참조하십시오. 다음 테이블에는 주문할 수 있는 책을 보여줍니다.

OS/2 및 Windows 플랫폼에서는 `sqllib\doc\html` 디렉토리에 HTML 파일을 설치할 수 있습니다. DB2 정보는 여러 나라 언어로 번역되었습니다. 하지만, 모든 정보가 모든 나라의 언어로 번역된 것은 아닙니다. 정보가 특정 나라의 언어로 사용할 수 없을 경우에는 영문으로 제공됩니다.

UNIX 플랫폼에서는 `doc/%L/html` 디렉토리에 여러 나라 언어 버전의 HTML 파일을 설치할 수 있습니다. 여기서 `%L`은 해당 언어의 로케일을 나타냅니다. 빠른 시작 책에서 보다 자세한 내용을 참조하십시오.

다음의 여러 가지 방법으로 DB2 책을 구하고 정보를 액세스할 수 있습니다.

- 323 페이지의 『정보 온라인 보기』
- 327 페이지의 『정보 온라인 검색』
- 319 페이지의 『인쇄된 책 주문』
- 319 페이지의 『PDF 책 인쇄』

표 18. DB2 정보

이름	설명	문서 번호	HTML 디렉토리
		PDF 파일 이름	
DB2 안내 및 참조 정보			
관리 안내서	관리 안내서: 계획에서는 데이터베이스의 개념에 대한 개요, 논리적 또는 물리적인 데이터베이스 설계와 같은 설계에 대한 정보 그리고 고가용성에 대한 정보를 제공합니다.	SA30-0990 db2d1x70	db2d0
	관리 안내서: 구현에서는 사용자의 설계, 데이터베이스 액세스, 감사, 백업 및 복구와 같은 구현에 대한 정보를 제공합니다.	SA30-0988	
	관리 안내서: 성능에서는 데이터베이스의 환경, 응용프로그램 성능 평가 및 성능 조정에 대한 정보를 제공합니다.	db2d2x70	
	사용자는 문서 번호 SBOF-8934를 사용하여 세 권으로 된 관리 안내서 책을 주문할 수 있습니다.	SA30-0989 db2d3x70	
Administrative API Reference	데이터베이스를 관리하는 데 사용할 수 있는 DB2 API와 데이터 구조에 대해 설명합니다. 또한 응용프로그램에서 API를 호출하는 방법을 설명합니다.	SC09-2947 db2b0x70	db2b0
응용프로그램 빌드 안내서	환경 설정 정보와 Windows에서 DB2 응용프로그램을 컴파일, 링크 및 수행하기 위한 지침이 단계별로 제공되어 있습니다.	SA30-0991 db2axx70	db2ax
APPC, CPI-C, and SNA Sense Codes	DB2 Universal Database 제품을 사용할 때 발생할 수 있는 APPC, CPI-C 및 SNA 센스 코드에 관한 일반 정보를 제공합니다. HTML 형식으로만 사용할 수 있습니다.	문서 번호가 없습니다. db2apx70	db2ap
응용프로그램 개발 안내서	Embedded SQL 또는 Java(JDBC 및 SQLJ)를 사용하여 DB2 데이터베이스를 액세스하는 응용프로그램을 개발하는 방법을 설명합니다. 저장 프로시저어 작성, 사용자 정의 함수 작성, 사용자 정의 유형 작성, 트리거 사용, 파티션된 환경 또는 연합 시스템에서 응용프로그램을 개발하는 등의 다양한 주제가 다루어집니다.	SA30-0992 db2a0x70	db2a0

표 18. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
PDF 파일 이름			
<i>CLI Guide and Reference</i>	DB2 콜 레벨 인터페이스와 Microsoft ODBC 스펙과 호환 가능한 호출 가능 SQL 인터페이스를 사용하여 DB2 데이터베이스에 액세스하는 응용프로그램의 개발 방법에 대해 설명합니다.	SC09-2950 db210x70	db210
<i>Command Reference</i>	명령행 프로세서를 사용하는 방법을 설명하고 데이터베이스를 관리하기 위해 사용할 수 있는 DB2 명령을 설명합니다.	SC09-2951 db2n0x70	db2n0
연결성 보충 설명서	AS/400용 DB2, OS/390용 DB2, MVS용 DB2 또는 VM용 DB2를 DB2 Universal Database 서버와의 DRDA 응용프로그램 리퀘스터로 사용하는 방법에 대한 참조 정보 및 설치 정보를 제공합니다. 또한 DB2 Connect AR(응용프로그램 리퀘스터)과 함께 DRDA AS(응용프로그램 서버)를 사용하는 방법에 대해서도 상세히 설명합니다. HTML 및 PDF 형식으로만 사용할 수 있습니다.	문서 번호가 없습니다. db2h1x70	db2h1
데이터 이동 유틸리티 안내 및 참조서	Import, Export, Load, AutoLoader 및 DPROP와 같이 데이터 이동을 용이하게 해 주는 DB2 UDB 유틸리티의 사용 방법에 대해 설명합니다.	SA30-0994 db2dmx70	db2dm
<i>Data Warehouse Center</i> 관리 안내서	Data Warehouse Center를 사용하여 데이터 웨어하우스를 구축 및 유지보수하는 방법을 제공합니다.	SA30-1000 db2ddx70	db2dd
<i>Data Warehouse Center</i> 응용프로그램 통합 안내서	프로그래머들이 Data Warehouse Center 및 Information Catalog Manager를 응용프로그램과 통합하는 데 도움을 주는 정보를 제공합니다.	SA30-1001 db2adx70	db2ad
<i>DB2 Connect</i> 사용자 안내서	DB2 Connect 제품에 대한 개념, 프로그래밍 및 일반 사용 정보를 제공합니다.	SA30-0993 db2c0x70	db2c0
<i>DB2 Query Patroller Administration Guide</i>	DB2 Query Patroller 시스템의 조작 개요, 특정 조작 및 관리 정보, 관리 그래픽 사용자 인터페이스 유틸리티에 대한 타스크 정보를 제공합니다.	SC09-2958 db2dwx70	db2dw

표 18. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
		PDF 파일 이름	
<i>DB2 Query Patroller User's Guide</i>	DB2 Query Patroller의 도구 및 함수를 사용하는 방법을 설명합니다.	SC09-2960	db2ww
		db2wwx70	
용어집	DB2에서 사용되는 용어와 그 구성요소에 대한 정의를 제공합니다.	문서 번호가 없습니다.	db2t0
	HTML 형식과 <i>SQL</i> 참조서에서 사용할 수 있습니다.	db2t0x70	
<i>Image, Audio 및 Video Extenders</i> 관리 및 프로그래밍	DB2 Extender에 대한 일반적인 정보와 이미지, 오디오 및 비디오(IAV)의 관리 및 구성에 대한 정보 그리고 IAV extenders를 사용한 프로그램에 대한 정보를 제공합니다. 여기에는 참조 정보, 진단 정보(메시지 포함) 및 샘플도 들어 있습니다.	SA30-1043	dmbu7
		dmbu7x70	
<i>Information Catalog Manager Administration Guide</i>	정보 카탈로그 관리에 대한 지침을 제공합니다.	SC26-9995	db2di
		db2dix70	
<i>Information Catalog Manager Programming Guide and Reference</i>	Information Catalog Manager에 대한 아키텍처 인터페이스에 대한 정의를 제공합니다.	SC26-9997	db2bi
		db2bix70	
<i>Information Catalog Manager 사용자 안내서</i>	Information Catalog Manager 사용자 인터페이스 사용에 대한 정보를 제공합니다.	SA30-1002	db2ai
		db2aix70	
설치 및 구성 보충 설명서	플랫폼 특정 DB2 클라이언트의 플랜, 설치 및 설정에 대해 설명합니다. 또한 바인딩, 클라이언트 및 서버 통신의 설정, DB2 GUI 도구, DRDA AS, 분산 설치 및 이중 데이터 소스에 대한 분산 요구와 액세스 방식의 구성에 대한 정보가 들어 있습니다.	GA30-0975	db2iy
		db2iyx70	
메시지 참조서	DB2, Information Catalog Manager 및 Data Warehouse Center에서 발행하는 메시지와 코드를 나열하고 수행해야 할 조치에 대해 설명합니다.	볼륨 1 GA30-0986	db2m0
		db2m1x70	
	문서 번호(SBOF-8932)를 사용하여 두 권으로 된 메시지 참조서 책을 모두 주문할 수 있습니다.	볼륨 2 GA30-0987	
		db2m2x70	

표 18. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
PDF 파일 이름			
<i>OLAP Integration Server Administration Guide</i>	OLAP 통합 서버의 관리 프로그램 구성요소를 사용하는 방법을 설명합니다.	SC27-0782	n/a
		db2dpx70	
<i>OLAP Integration Server Metaoutline User's Guide</i>	표준 OLAP Metaoutline 인터페이스 (Metaoutline Assistant가 아닌)를 사용하여 OLAP Metaoutlines을 작성하고 사용하는 방법을 설명합니다.	SC27-0784	n/a
		db2upx70	
<i>OLAP Integration Server Model User's Guide</i>	표준 OLAP 모델 인터페이스(Model Assistant가 아닌)를 사용하여 OLAP 모델을 작성하는 방법을 설명합니다.	SC27-0783	n/a
		db2lpx70	
<i>OLAP 설치 및 사용자 안내서</i>	OLAP Starter Kit에 대한 구성 및 설치 정보를 제공합니다.	SA30-1074	db2ip
		db2ipx70	
<i>Excel용 OLAP SpreadSheet Add-in 사용자 안내서</i>	Excel 스프레드시트 프로그램을 사용하여 OLAP 데이터를 분석하는 방법을 설명합니다.	SA30-0564	db2ep
		db2epx70	
<i>Lotus 1-2-3용 OLAP SpreadSheet Add-in 사용자 안내서</i>	Lotus 1-2-3 스프레드시트 프로그램을 사용하여 OLAP 데이터를 분석하는 방법을 설명합니다.	SA30-0565	db2tp
		db2tpx70	
<i>복제 안내 및 참조서</i>	DB2와 함께 제공된 IBM 복제 도구에 관한 플랜, 구성, 관리 및 사용 정보를 제공합니다.	SA30-1003	db2e0
		db2e0x70	
<i>Spatial Extender 사용자 안내 및 참조서</i>	Spatial Extender 설치, 구성, 관리, 프로그래밍 및 문제 해결에 대한 정보를 제공합니다. 또한 공간 데이터 개념에 대한 설명을 제공하고 Spatial Extender에만 고유하게 적용되는 참조 정보(메시지 및 SQL)를 제공합니다.	SA30-1045	db2sb
		db2sbx70	
<i>SQL 시작하기</i>	SQL 개념을 소개하고, 많은 구조와 타스크에 관한 예를 보여줍니다.	SA30-0996	db2y0
		db2y0x70	
<i>SQL 참조서, 볼륨 1 및 볼륨 2</i>	SQL 구문, 의미 그리고 언어 규칙에 대해 설명합니다. 또한 릴리스 간 비호환성, 제품 제한 사항 및 카탈로그 뷰에 대한 정보도 들어 있습니다.	볼륨 1 SA30-0997	db2s0
		db2s1x70	
	SBOF-8933 문서 번호를 사용하여 SQL 참조서 책을 주문할 수 있습니다.	볼륨 2 SA30-0998	
		db2s2x70	

표 18. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
PDF 파일 이름			
시스템 모니터 안내 및 참조 서	데이터베이스와 데이터베이스 관리 프로그램에 관한 여러 종류의 정보를 수집하는 방법에 대해 설명합니다. 이 책은 데이터베이스 활동을 이해하고, 성능을 향상시키고, 문제점의 원인을 판별하기 위한 정보를 사용하는 방법을 설명합니다.	SA30-0995 db2f0x70	db2f0
Text Extender 관리 및 프로그래밍	DB2 extenders에 관한 일반적인 정보와 Text extenders 관리 및 구성에 관한 정보, Text extenders를 사용한 프로그래밍에 관한 정보를 제공합니다. 여기에는 참조 정보, 진단 정보(메시지 포함) 및 샘플도 들어 있습니다.	SA30-1044 desu9x70	desu9
문제점 해결 안내서	오류의 출처를 판별하고 문제점으로부터 회복하고, DB2 고객 서비스와 상담하여 진단 도구를 사용하는 것을 도와줍니다.	GA30-0704 db2p0x70	db2p0
새로운 기능	DB2 Universal Database, 버전 7의 새로운 특성, 기능 및 향상된 내용을 설명합니다.	SA30-0999 db2q0x70	db2q0
DB2 설치 및 구성 정보			
OS/2 및 Windows용 DB2 Connect Enterprise Edition 빠른 시작	OS/2 및 Windows 32 비트 운영 체제에서 DB2 Connect Enterprise Edition에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0974 db2c6x70	db2c6
UNIX용 DB2 Connect Enterprise Edition 빠른 시작	UNIX 기반 플랫폼에서의 DB2 Connect Enterprise Edition에 대한 플랜, 이주, 설치, 구성 및 타스크 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0973 db2cyx70	db2cy
DB2 Connect Personal Edition 빠른 시작	OS/2 및 Windows 32 비트 운영 체제에서 DB2 Connect Personal Edition에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다. 또한 지원되는 모든 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0981 db2c1x70	db2c1
DB2 Connect Personal Edition Quick Beginnings for Linux	지원되는 모든 Linux에서 DB2 Connect Personal Edition에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다.	GC09-2962 db2c4x70	db2c4

표 18. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
		PDF 파일 이름	
<i>DB2 Data Links Manager</i> 빠른 시작	AIX 및 Windows 32 비트 운영 체제용 DB2 Data Links Manager에 대한 플랜, 설치, 구성 및 타스크 정보를 제공합니다.	GA30-0980 db2z6x70	db2z6
<i>UNIX용 DB2 Enterprise - Extended Edition</i> 빠른 시작	UNIX 기반 플랫폼에서의 DB2 Enterprise - Extended Edition 플랜, 설치 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0978 db2v3x70	db2v3
<i>Windows용 DB2 Enterprise - Extended Edition</i> 빠른 시작	Windows 32 비트 운영 체제용 DB2 Enterprise - Extended Edition에 관한 플랜, 설치 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0977 db2v6x70	db2v6
<i>OS/2용 DB2</i> 빠른 시작	OS/2 운영 체제에서의 DB2 Universal Database에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0982 db2i2x70	db2i2
<i>UNIX용 DB2</i> 빠른 시작	UNIX 기반 플랫폼에서의 DB2 Universal Database에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0984 db2ixx70	db2ix
<i>Windows용 DB2</i> 빠른 시작	Windows 32 비트 운영 체제에서 DB2 Universal Database에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0985 db2i6x70	db2i6
<i>DB2 Personal Edition</i> 빠른 시작	OS/2 및 Windows 32 비트 운영 체제에서의 DB2 Universal Database Personal Edition에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다.	GA30-0983 db2i1x70	db2i1
<i>DB2 Personal Edition Quick Beginnings for Linux</i>	지원되는 모든 Linux에서 DB2 Universal Database Personal Edition에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다.	GC09-2972 db2i4x70	db2i4
<i>DB2 Query Patroller</i> 설치 안내서	DB2 Query Patroller에 관한 설치 정보를 제공합니다.	GA30-0976 db2iw70	db2iw

표 18. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
		PDF 파일 이름	
<i>DB2 Warehouse Manager</i> 설치 안내서	웨어하우스 에이전트, 웨어하우스 변환기 및 Information Catalog Manager에 관한 설치 정보를 제공합니다.	GA30-1027 db2idx70	db2id
HTML 형식의 플랫폼 공통 샘플 프로그램			
HTML 샘플 프로그램	DB2가 지원하는 모든 플랫폼에서 프로그래밍 언어에 대한 샘플 프로그램이 HTML 형식으로 제공됩니다. 이 샘플 프로그램은 정보용으로만 사용됩니다. 모든 샘플을 모든 프로그램 언어로 사용할 수 있는 것은 아닙니다. DB2 응용프로그램 개발 클라이언트가 설치될 때만 HTML 샘플을 사용할 수 있습니다. 프로그램에 대한 정보는 응용프로그램 빌드 안내서에서 자세한 내용을 참조하십시오.	문서 번호가 없습니다.	db2hs
릴리스 정보			
<i>DB2 Connect</i> 릴리스 정보	DB2 Connect 책에는 포함될 수 없었던 최신 정보를 제공합니다.	#2를 참조하십시오.	db2cr
<i>DB2</i> 설치 정보	DB2 책에는 포함될 수 없었던 최신 설치 정보를 제공합니다.	제품 CD-ROM에서만 사용할 수 있습니다.	
<i>DB2</i> 릴리스 정보	DB2 책에는 포함될 수 없었던 모든 DB2 제품 및 기능에 대한 최신 정보를 제공합니다.	#2를 참조하십시오.	db2ir

주:

1. 파일 이름의 6번째 자리에 있는 문자 *x*는 책의 언어 버전을 나타냅니다. 예를 들면, 파일 이름 db2d0e70은 관리 안내서 책의 영문 버전을 나타내며 db2d0k70은 같은 책의 한글 버전을 나타냅니다. 다음 문자는 언어 버전을 나타내기 위해 파일 이름의 6번째 자리에 사용됩니다.

언어	식별자
브라질 포르투갈어	b
불가리아어	u
체코어	x
덴마크어	d
네덜란드어	q
영어	e

핀란드어	y
프랑스어	f
독일어	g
그리스어	a
헝가리어	h
이탈리아어	i
일본어	j
한글	k
노르웨이어	n
폴란드어	p
포르투갈어	v
러시아어	r
중국어	c
슬로베니아어	l
스페인어	z
스웨덴어	s
대만어	t
터키어	m

2. DB2 책에 포함되어 있지 않을 수 있는 최신 정보는 릴리스 정보에서 HTML 형식과 ASCII 파일로 사용할 수 있습니다. HTML 버전은 정보 센터와 제품 CD-ROM에서 사용할 수 있습니다. ASCII 파일을 보려면,

- UNIX 기반 플랫폼의 경우에는 Release.Notes 파일을 참조하십시오. 이 파일은 DB2DIR/Readme/%L 디렉토리에 있으며 여기서, %L은 로케일 이름이고 DB2DIR은 다음과 같습니다.
 - AIX에서는 /usr/lpp/db2_07_01
 - HP-UX, PTX, Solaris 및 Silicon Graphics IRIX에서는 /opt/IBMdb2/V7.1
 - Linux에서는 /usr/IBMdb2/V7.1
- 다른 플랫폼의 경우에는 RELEASE.TXT 파일을 참조하십시오. 이 파일은 제품이 설치된 디렉토리에 있습니다. OS/2 플랫폼에서는 **IBM DB2** 폴더를 더블 클릭하고 릴리스 정보 아이콘을 더블 클릭할 수 있습니다.

PDF 책 인쇄

책의 사본을 원하는 경우 DB2 책 CD-ROM에 있는 PDF 파일을 인쇄할 수 있습니다. Adobe Acrobat Reader를 사용하여 책 전체나 특정 페이지를 인쇄할 수 있습니다. 라이브러리에 있는 각 책의 파일 이름에 대해서는 311 페이지의 표18에서 자세한 내용을 참조하십시오.

Adobe 웹 사이트인 <http://www.adobe.com>에서 Adobe Acrobat Reader의 최신 버전을 얻을 수 있습니다.

PDF 파일은 파일 확장자가 PDF로서 DB2 책 CD-ROM에 들어 있습니다. PDF 파일을 액세스하려면,

1. DB2 책 CD-ROM을 삽입하십시오. UNIX 기반의 플랫폼에서는 DB2 책 CD-ROM을 마운트해야 합니다. 마운트 절차에 대해서는 *빠른 시작* 책에서 자세한 내용을 참조하십시오.
2. Acrobat Reader를 시작하십시오.
3. 다음 위치에서 원하는 PDF 파일을 여십시오.
 - OS/2 및 Windows 플랫폼에서:
`x:\doc\language` 디렉토리. 여기서 *x*는 CD-ROM 드라이브를 나타내며 *language*는 사용자 언어를 나타내는 2문자 국가 코드를 나타냅니다
 - UNIX 기반 플랫폼에서:
`/cdrom/doc/%L` 디렉토리. 여기서 */cdrom*은 CD-ROM의 마운트 위치이고 *%L*은 원하는 로케일의 이름입니다.

또한 PDF 파일을 CD-ROM에서 지역이나 네트워크로 파일을 복사하고 거기서 읽을 수도 있습니다.

인쇄된 책 주문

인쇄된 DB2 책은 책 주문 번호(SBOF)를 사용하여 세트나 낱권으로 주문할 수 있습니다. 인쇄본을 주문하려면, IBM 협력업체 또는 영업 대표에게 문의하십시오. 또한 웹 사이트 <http://www.elink.ibm.com/pbl/pbl>에서도 책을 주문할 수 있습니다.

두 종류의 책 세트를 사용할 수 있습니다. SBOF-8935는 DB2 Warehouse Manager에 대한 참조 및 사용에 관한 정보를 제공합니다. SBOF-8931은 다른 모든 DB2 Universal Database 제품과 특징에 대한 참조 및 사용 정보를 제공합니다. 각 SBOF의 내용은 다음 테이블에 나열되어 있습니다.

표 19. 인쇄된 책 주문

SBOF 번호	포함된 책
SBOF-8931	<ul style="list-style-type: none"> • 관리 안내서: 계획 • 관리 안내서: 구현 • 관리 안내서: 성능 • Administrative API Reference • 응용프로그램 빌드 안내서 • 응용프로그램 개발 안내서 • CLI Guide and Reference • Command Reference • 데이터 이동 유틸리티 안내 및 참조서 • Data Warehouse Center 관리 안내서 • Data Warehouse Center 응용프로그램 통합 안내서 • DB2 Connect 사용자 안내서 • 설치 및 구성 보충 설명서 • Image, Audio 및 Video Extenders 관리 및 프로그래밍 • 메시지 참조서, 볼륨 1 및 2 • OLAP Integration Server Administration Guide • OLAP Integration Server Metaoutline User's Guide • OLAP Integration Server Model User's Guide • OLAP Integration Server User's Guide • OLAP 설정 및 사용자 안내서 • Excel용 OLAP Spreadsheet Add-in 사용자 안내서 • Lotus 1-2-3용 OLAP Spreadsheet Add-in 사용자 안내서 • 복제 안내 및 참조서 • Spatial Extender 관리 및 프로그래밍 안내서 • SQL 시작하기 • SQL 참조서, 볼륨 1 및 2 • 시스템 모니터 안내 및 참조서 • Text Extender 관리 및 프로그래밍 • 문제점 해결 안내서 • 새로운 기능
SBOF-8935	<ul style="list-style-type: none"> • Information Catalog Manager Administration Guide • Information Catalog Manager 사용자 안내서 • Information Catalog Manager Programming Guide and Reference • Query Patroller Administration Guide • Query Patroller User's Guide

DB2 온라인 문서

온라인 도움말 액세스

온라인 도움말은 모든 DB2 구성요소에서 사용할 수 있습니다. 다음의 테이블에서는 다양한 도움말 유형을 설명합니다.

도움말의 유형	내용	액세스하는 방법
명령 도움말	명령행 처리기 내의 명령 구문을 설명합니다.	대화식 모드인 명령행 처리기에서, 다음을 입력하십시오. <i>? command</i> 여기서, <i>command</i> 는 키워드이거나 전체 명령입니다. 예를 들어, <i>? catalog</i> 는 모든 CATALOG 명령에 대한 도움말을 표시하고, <i>? catalog database</i> 는 CATALOG DATABASE 명령에 대한 도움말을 표시합니다.
클라이언트 구성 지원 프로그램 도움말	창 또는 노트북에서 수행할 수 있는 작업을 설명합니다. 도움말은 알아야 할 개요와 전체조건 정보를 포함하고, 창 또는 노트북 제어를 사용하는 방법을 설명합니다.	창이나 노트북에서, 도움말 버튼을 누르거나 F1 키를 누르십시오.
명령 센터 도움말		
제어 센터 도움말		
Data Warehouse Center 도움말		
이벤트 분석기 도움말		
Information Catalog Manager 도움말		
위성 관리 센터 도움말		
스크립트 센터 도움말		

도움말의 유형	내용	액세스하는 방법
메시지 도움말	메시지의 원인과 사용자가 취해야 할 조치를 설명합니다.	<p>대화식 모드인 명령행 처리기에서, 다음을 입력하십시오.</p> <p><code>? XXXnnnnn</code></p> <p>여기서, <code>XXXnnnnn</code>은 유효한 메시지 식별자입니다.</p> <p>예를 들어, <code>? SQL30081</code>은 <code>SQL30081</code> 메시지에 대한 도움말을 표시합니다.</p> <p>한 번에 한 화면씩 메시지 도움말을 보려면, 다음을 입력하십시오.</p> <p><code>? XXXnnnnn more</code></p> <p>파일에 메시지 도움말을 저장하려면, 다음을 입력하십시오.</p> <p><code>? XXXnnnnn > filename.ext</code></p> <p>여기서, <code>filename.ext</code>는 메시지 도움말을 저장하려는 파일입니다.</p>
SQL 도움말	SQL문의 구문을 설명합니다.	<p>대화식 모드인 명령행 처리기에서, 다음을 입력하십시오.</p> <p><code>help statement</code></p> <p>여기서, <code>statement</code>는 SQL문입니다.</p> <p>예를 들어, <code>help SELECT</code>는 <code>SELECT</code>문에 대한 도움말을 표시합니다.</p> <p>주: SQL 도움말은 UNIX 기반 플랫폼에서 사용할 수 없습니다.</p>
SQLSTATE 도움말	SQL 상태 및 클래스 코드를 설명합니다.	<p>대화식 모드인 명령행 처리기에서, 다음을 입력하십시오.</p> <p><code>? sqlstate</code> 또는 <code>? class code</code></p> <p>여기서, <code>sqlstate</code>는 유효한 5자리 숫자로 된 SQL 상태이고 <code>class code</code>는 SQL 상태의 처음 2자리 숫자입니다.</p> <p>예를 들어, <code>? 08003</code>은 <code>08003</code> SQL 상태에 대한 도움말을 표시하고, <code>? 08</code>은 <code>08</code> 클래스 코드에 대한 도움말을 표시합니다.</p>

정보 온라인 보기

이 제품에 들어 있는 책은 HTML(Hypertext Markup Language) 소프트웨어 형식으로 제공됩니다. 소프트웨어는 정보를 검색할 수 있게 하고 관련된 정보로 링크하는 하이퍼텍스트를 제공합니다. 또한, 사이트에서 라이브러리를 공유하는 것도 더 쉬워집니다.

HTML 버전 3.2 스펙을 따르는 브라우저로 온라인 책 또는 샘플 프로그램을 볼 수 있습니다.

온라인 책 또는 샘플 프로그램을 보려면:

- DB2 관리 도구를 수행할 경우, 정보 센터를 사용하십시오.
- 브라우저에서, 파일 →페이지 열기를 클릭하십시오. 열린 페이지에 DB2 정보에 대한 설명과 링크가 들어 있습니다.
 - UNIX 기반 플랫폼에서는 다음과 같은 페이지를 여십시오.

```
INSTHOME/sql1lib/doc/%L/html/index.htm
```

여기서 %L은 로케일 이름입니다.

- 다른 플랫폼에서는 다음 페이지를 여십시오

```
sql1lib\doc\html\index.htm
```

이 경로는 DB2가 설치된 드라이브에 있습니다.

정보 센터를 설치하지 않은 경우, **DB2 정보** 아이콘을 더블 클릭하여 페이지를 열 수 있습니다. 사용하는 시스템에 따라, 주 제품 폴더나 Windows 시작 메뉴에 아이콘이 있습니다.

Netscape 브라우저 설치

웹 브라우저를 설치하지 않은 경우, 제품 상자에 있는 Netscape CD-ROM에서 Netscape를 설치할 수 있습니다. 설치하는 방법에 대한 자세한 지시 사항은 다음을 수행하십시오.

1. Netscape CD-ROM을 삽입하십시오.
2. UNIX 기반의 플랫폼에서는 CD-ROM을 마운트해야 합니다. 마운트 절차에 대해서는 빠른 시작 책에서 자세한 내용을 참조하십시오.

3. 설치 지침서는 CDNAVnn.txt 파일을 참조하십시오. 여기서, nn은 2문자로 된 언어 식별자입니다. 파일은 CD-ROM의 루트 디렉토리에 있습니다.

정보 센터로 정보에 액세스

정보 센터는 DB2 제품 정보로의 빠른 액세스를 제공합니다. 정보 센터는 DB2 관리 도구를 사용할 수 있는 모든 플랫폼에서 사용할 수 있습니다.

정보 센터 아이콘을 더블 클릭하여 정보 센터를 열 수 있습니다. 사용하는 시스템에 따라 아이콘은 주 제품 폴더나 Windows 시작 메뉴의 정보 폴더에 있습니다.

또한 DB2 Windows 플랫폼에서 도구 모음이나 도움말 메뉴를 사용하여 정보 센터를 액세스할 수 있습니다.

정보 센터는 6개 유형의 정보를 제공합니다. 적당한 탭을 클릭하여 그 유형에서 지원하는 주제를 보십시오.

타스크 DB2를 사용하여 수행할 수 있는 키 타스크.

참조 키워드, 명령 및 API와 같은 DB2 참조 정보.

책 DB2 책.

문제점 해결 오류 메시지의 종류와 복구 조치.

샘플 프로그램 DB2 응용프로그램 개발 클라이언트와 함께 제공되는 샘플 프로그램. DB2 응용프로그램 개발 클라이언트를 설치하지 않은 경우, 이 탭은 표시되지 않습니다.

웹 월드 와이드 웹에서의 DB2 정보. 이 정보에 액세스하려면, 사용자의 시스템으로부터 웹으로의 연결이 있어야 합니다.

목록 중 하나에서 항목을 선택할 때, 정보 센터는 정보를 표시하기 위해 표시기를 시작합니다. 표시기는 사용자가 선택하는 정보의 종류에 따라, 시스템 도움말 표시기, 편집기 또는 웹브라우저가 될 수 있습니다.

정보 센터는 찾기 기능을 제공하므로 목록을 찾지 않고도 특정 주제를 찾을 수 있습니다.

전체 텍스트 검색을 위해서는 **DB2 온라인 정보 검색** 검색 양식으로 연결된 정보 센터의 하이퍼텍스트 링크를 따라 검색하십시오.

HTML 검색 서버는 보통 자동으로 시작됩니다. HTML 정보에서 검색 기능이 작동하지 않으면, 다음 방법 중 하나를 사용하여 검색 서버를 시작할 수 있습니다.

Windows의 경우:

시작을 클릭하고 프로그램 → IBM DB2 → 정보 → HTML 검색 서버 시작을 선택하십시오.

OS/2 경우:

OS/2용 DB2 폴더를 더블 클릭하고 HTML 검색 서버 시작 아이콘을 더블 클릭하십시오.

HTML 정보를 검색하면서 다른 문제가 생길 경우, 릴리스 정보를 참조하십시오.

주: 검색 기능은 Linux, PTX 및 Silicon Graphics IRIX 환경에서는 작동하지 않습니다.

DB2 마법사 사용

마법사는 한 번에 한 단계씩 각 작업을 수행하게 함으로써 특정 관리 작업을 완료하는 데 도움을 줍니다. 마법사는 제어 센터 및 클라이언트 구성 지원 프로그램을 통해 사용할 수 있습니다. 다음 테이블에서는 마법사를 나열하고 그 기능을 설명합니다.

주: 데이터베이스 작성, 색인 작성, 다중 사이트 갱신 구성 및 성능 구성 마법사는 파티션된 데이터베이스 환경에서 사용할 수 있습니다.

마법사	도움 대상	액세스하는 방법
데이터베이스 추가	클라이언트 워크스테이션의 데이터베이스를 카탈로그화합니다.	클라이언트 구성 지원 프로그램에서 추가를 클릭하십시오.
데이터베이스 백업	백업 계획을 결정하고, 작성하고, 일정을 세웁니다.	제어 센터에서 백업하려는 데이터베이스를 마우스의 오른쪽 버튼으로 클릭한 다음 백업 → 마법사를 사용한 데이터베이스 백업을 선택하십시오.
다중 사이트 갱신 구성	다중 사이트 갱신, 분선 트랜잭션 또는 2 단계 확약을 구성합니다.	제어 센터에서 데이터베이스 폴더를 마우스의 오른쪽 버튼으로 클릭하고 다중 사이트 갱신을 선택하십시오.

마법사	도움 대상	액세스하는 방법
데이터베이스 작성	데이터베이스를 작성한 다음, 몇 가지 기본적인 구성 작업을 수행합니다.	제어 센터에서 데이터베이스 폴더를 마우스의 오른쪽 버튼으로 클릭한 다음 작성 → 마법사를 사용한 데이터베이스 작성을 선택하십시오.
테이블 작성	기본 데이터 유형을 선택한 다음, 테이블에 대한 기본 키를 작성합니다.	제어 센터에서 테이블 아이콘을 마우스의 오른쪽 버튼으로 클릭하고 작성 → 마법사를 사용한 테이블을 선택하십시오.
테이블 공간 작성	새로운 테이블 공간을 작성합니다.	제어 센터에서 테이블 공간 아이콘을 마우스의 오른쪽 버튼으로 선택하고 작성 → 마법사를 사용한 테이블 공간을 선택하십시오.
색인 작성	사용자의 모든 조회를 작성하고 삭제하기 위해 색인화합니다.	제어 센터에서 색인 아이콘을 마우스의 오른쪽 버튼으로 클릭하고 작성 → 마법사를 사용한 색인을 선택하십시오.
성능 구성	업무 요구조건에 맞게 구성 매개변수를 갱신하여 데이터베이스의 성능을 조정합니다.	제어 센터에서 성능을 조정하려는 데이터베이스를 마우스의 오른쪽 버튼으로 클릭하고 마법사를 사용한 성능 구성을 선택하십시오. 파티션된 데이터베이스에 대해 데이터베이스 파티션 뷰로부터 성능을 조정하려는 첫번째 파티션을 마우스의 오른쪽 버튼으로 클릭하고 마법사를 사용한 성능 구성을 선택하십시오.
데이터베이스 복원	실패 후에 데이터베이스를 복구합니다. 사용할 백업 위치 및 재작동할 로그 기록을 이해하는 데 도움을 줍니다.	제어 센터에서 복원하려는 데이터베이스를 마우스의 오른쪽 버튼으로 클릭한 다음 복원 → 마법사를 사용한 데이터베이스를 선택하십시오.

문서 서버 설정

기본 값으로 DB2 정보는 지역 시스템에 설치됩니다. 이는 DB2 정보에 액세스해야 하는 모든 사람이 동일한 파일을 설치해야 함을 의미합니다. DB2 정보를 한 위치에 저장하려면, 다음과 같이 하십시오.

1. 지역 시스템의 `\sqllib\doc\html`에 있는 모든 파일과 서브디렉토리를 웹 서버로 복사하십시오. 각 책은 책을 구성하는 데 필요한 모든 HTML 및 GIF 파일이 들어 있는 서브디렉토리를 가집니다. 디렉토리 구조가 변경되지 않게 하십시오.
2. 새로운 위치에 있는 파일을 찾으려면 웹 서버를 구성하십시오. 보다 자세한 정보는 `설치 및 구성 보충 설명서`의 부록 `NetQuestion`을 참조하십시오.
3. Java 버전의 정보 센터를 이용하는 경우, 모든 HTML 파일에 대한 기본 URL을 지정할 수 있습니다. 책 목록에 대해서는 URL을 사용해야 합니다.
4. 책 파일을 열람할 수 있게 되면, 다음과 같이 자주 열람하는 주제 항목에 대해서는 북마크를 설정할 수 있습니다. 다음의 페이지들을 북마크로 설정해 두면 도움이 될 것입니다.
 - 책 목록
 - 자주 이용하는 책의 목차
 - ALTER TABLE 주제와 같은 자주 참조하는 항목
 - 검색 양식

DB2 Universal Database 온라인 문서 파일을 중앙 시스템에서 제공하는 방법에 대한 정보를 보려면 `설치 및 구성 보충 설명서`의 부록 `NetQuestion`을 참조하십시오.

정보 온라인 검색

HTML 파일에서 정보를 찾으려면, 다음 방법 중 하나를 사용하십시오.

- 맨 위 프레임에서 검색을 클릭하십시오. 특정 주제를 찾으려면 검색 형식을 사용하십시오. 이 기능은 Linux, PTX 또는 Silicon Graphics IRIX 환경에서는 사용할 수 없습니다.
- 맨 위 프레임에서 색인을 클릭하십시오. 책에서 특정 주제를 찾으려면 색인을 사용하십시오.
- 책에서 특정 주제를 찾으려면 목차나 도움말의 색인 또는 HTML 책을 표시하고 웹 브라우저의 찾기 기능을 사용하십시오.
- 특정 주제로 빨리 리턴하려면 웹 브라우저의 북마크 기능을 사용하십시오.

- 특정 주제를 찾으려면 정보 센터의 검색 기능을 사용하십시오. 324 페이지의 『정보 센터로 정보에 액세스』에서 자세한 내용을 참조하십시오.

부록F. 주의사항

IBM은 다른 나라에서는 이 책에 언급된 제품, 서비스 또는 기능들을 제공하지 않을 수도 있습니다. 각 지역에서 현재 사용 가능한 제품 및 서비스에 대해서는 해당 지역의 IBM 영업 대표에 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 IBM 제품, 프로그램 또는 서비스만 사용할 수 있다는 의미는 아닙니다. IBM의 지적 재산을 침해하지 않는 한 기능적으로 동일한 제품, 프로그램 또는 서비스를 대신 사용할 수 있습니다. 그러나 IBM 이외의 제품, 프로그램 또는 서비스 조작에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 문서에서 논의되는 주제에 대해 특허를 갖고 있거나 현재 출원 중일 수 있습니다. 이 책을 제공한다고 해서 그러한 특허에 대한 사용권까지를 부여하는 것은 아닙니다. 특허 사용권에 대한 문의는 다음 주소로 하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩
한국 아이.비.엠 주식회사
지적 재산권부

2바이트(DBCS) 정보에 관한 사용권 문의는 사용자 국가의 IBM 지적 재산권부나 다음 주소로 서면 문의하십시오.

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

다음 사항은 영국이나 이 조항이 현지법과 상충되는 나라에는 적용되지 않습니다. IBM에서는 이 책을 명시적 또는 암시적인 어떠한 종류의 보증없이 『있는 그대로』 제공하므로, 판매 가능성을 보장하거나 특정 목적에 적합한지 여부에 대해서는 책임질 수 없습니다. 일부 국가에서는 특정 거래의 명시적 또는 암시적인 보증을 부인하는 문장을 허용하지 않으므로, 이 사항이 사용자에게 적용되지 않을 수도 있습니다.

이 책에는 기술상 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 책의 내용은 정기적으로 변경되며, 이들 변경사항은 개정판에 통합됩니다. IBM은 사전 통지없이 언제든지 이 책에 설명된 제품과 프로그램을 개선 및 변경할 수 있습니다.

이 책에서 타사의 웹 사이트를 언급한 것은 단지 편의를 위해서일 뿐이며 이런 웹 사이트를 추천하려는 의도는 아닙니다. 이런 웹 사이트의 데이터가 이 IBM 제품에 대한 데이터의 일부는 아니므로 이런 웹 사이트 사용에 대한 책임은 사용자가 져야 합니다.

IBM은 독자가 제공한 정보를 적절한 방식으로 사용하거나 배포할 수 있으며, 제공한 독자는 이에 대해 책임을 지지 않습니다.

이 프로그램의 사용권자가 (i) 독립적으로 작성된 프로그램과 다른 프로그램(이 프로그램을 포함한) 사이의 정보 교환과 (2) 교환된 정보의 공동 사용을 목적으로 그 프로그램에 대한 정보를 원하는 경우, 다음 주소로 문의하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩
한국 아이.비.엠 주식회사
소프트웨어 사업본부

이러한 정보는 특정한 기간 및 조건하에 사용가능하며 어떤 경우에는 사용료를 지불해야 합니다.

이 책에 기술된 사용권 프로그램 및 이 프로그램에 사용가능한 모든 사용권 데이터는 IBM 고객 협약, IBM 국제 프로그래밍 사용권 협약 또는 이와 동등한 모든 협약 조건하에 IBM에서 제공됩니다.

여기에 제시된 어떠한 성능 데이터는 주위 환경에 따라 결정될 수 있습니다. 따라서, 다른 운영 체제에서 제시된 결과 값과 다를 수 있습니다. 몇몇 측정값은 개발 단계에서 얻은 값일 수 있습니다. 따라서 일반적인 사용자 시스템에서 얻은 값과 다를 수 있습니다. 또한 몇몇 측정값은 보외법을 통해 측정된 값입니다. 실제 값과는 다를 수 있습니다. 이 책의 사용자는 사용자의 특정 환경에 맞게 적용가능한 데이터를 변경해야 합니다.

타사 제품과 관련된 정보는 해당 제품의 공급자, 공개 발표 또는 기타 공개적으로 사용가능한 소스에서 확보한 것입니다. IBM은 이들 제품을 검사하지 않았으므로, 성능의 정확성이나 호환성 또는 타사 제품과 관련한 기타 여러 가지 요구들을 확인할 수 없습니다. 타사 제품의 성능에 관한 문제는 해당 제품의 공급자에게 제기되어야 합니다.

IBM이 제시하는 방향 또는 의도에 관한 어떠한 언급도 특별한 통지없이 변경될 수 있습니다.

이 정보는 일상적인 비즈니스 처리에 사용되는 데이터와 보고서의 예가 들어 있을 수 있습니다. 보다 구체적으로 예를 나타내기 위해 특정 개인, 회사, 상표 또는 제품 이름이 언급되는 경우가 있습니다. 여기서 언급된 이름은 가상의 이름이며 실제 비즈니스 업체가 사용하는 이름 및 주소와 유사하다면 우연인 것입니다.

사용권:

이 정보에는 여러 운영 체제에서 프로그래밍 소스 언어로 예제 응용프로그램이 들어 있을 수 있습니다. 사용자는 이들 예제 프로그램을 IBM에게 비용을 지급하지 않고 복사, 수정 및 분배할 수 있습니다. 이들 예제 프로그램은 모든 조건에서 철저히 검사되지 않았습니다. 따라서 IBM은 이들 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 함축할 수 없습니다.

이들 예제 프로그램의 각각의 복사본이나 특정 부분은 다음과 같은 사용권 주의 사항을 포함해야 합니다.

© (사용자 회사 이름) (년도). 이 코드의 일부는 IBM Corp. 예제 프로그램에서 발췌된 것입니다. © Copyright IBM Corp. (년도 입력). All rights reserved.

등록상표

별표(*)로 표시된 다음의 용어는 전세계에서 IBM의 상표입니다.

ACF/VTAM	IBM
AISPO	IMS
AIX	IMS/ESA
AIX/6000	LAN DistanceMVS
AIXwindows	MVS/ESA
AnyNet	MVS/XA
APPN	Net.Data
AS/400	OS/2
BookManager	OS/390
CICS	OS/400
C Set++	PowerPC
C/370	QBIC
DATABASE 2	QMF
DataHub	RACF
DataJoiner	RISC System/6000
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/DS
DB2 Extenders	SQL/400
DB2 OLAP Server	System/370
DB2 Universal Database	System/390
Distributed Relational Database Architecture	SystemView
DRDA	VisualAge
eNetwork	VM/ESA
Extended Services	VSE/ESA
FFST	VTAM
First Failure Support Technology	WebExplorer
	WIN-OS/2

다음 용어는 해당 회사의 상표 또는 등록 상표입니다.

Microsoft, Windows 및 Windows NT는 Microsoft Corporation의 상표 또는 등록 상표입니다.

Java 또는 모든 Java 관련 상표 및 로고 그리고 Solaris는 전세계에서 Sun Microsystems, Inc.의 상표입니다.

Tivoli 및 NetView는 전세계에서 Tivoli Systems Inc.의 상표입니다.

UNIX는 전세계에서 X/Open Company Limited가 독점권을 갖는 등록 상표입니다.

두 개의 별표(**)가 붙은 기타 회사 이름, 제품 이름 또는 서비스 이름은 해당 회사의 상표이거나 서비스 표시입니다.

색인

[가]

가져오기

- 에 대한 파일 유형 수정자 67
- 테이블에 PC/IXF 파일 43
- PC/IXF 파일의, forcein을 사용한 301

가져오기 메시지 파일 28, 81, 182

가져오기 유틸리티

- 개요 30
- 내보낸 테이블 재작성 38
- 대형 오브젝트(LOB) 40
- 로드 유틸리티와 비교 237
- 버퍼된 삽입 34
- 사용자 정의 구별 유형(UDT) 40
- 사용하는 데 필요한 권한 및 특권 31
- 생성된 컬럼 36
- 성능 80
- 성능 최적화 80
- 식별 컬럼 34
- 원격 데이터베이스 33
- 제한사항 80
- 코드 페이지 고려사항 77
- 클라이언트/서버 33
- 테이블 잠금 41
- 한계점 80
- 호스트와 워크스테이션간 데이터 전송 220

DB2 Data Links Manager 208

가져오기 유틸리티 파일 형식 239

가져오기 파일 유형으로 ASC 45

검색

온라인 정보 324, 327

경고 메시지 307

계속 레코드, PC/IXF 271

계속 레코드, PC/IXF 268

구문 도표 233

구조

- 컬럼 식별자가 없는 ASCII(ASC) 파일 246
- 컬럼 식별자가 있는 ASCII(DEL) 파일 240

권한

- 가져오기 유틸리티에 필요한 31
- 내보내기 유틸리티에 필요한 3
- 로드 유틸리티에 필요한 91
- AutoLoader 유틸리티에 필요한 185

[나]

내보내기

- 에 대한 파일 유형 수정자 20
- 컬럼 이름 지정 14

내보내기 메시지 파일 28, 81, 182

내보내기 유틸리티

- 개요 2
- 내보낸 테이블 재작성 5
- 대형 오브젝트(LOB) 5
- 사용하는 데 필요한 권한 및 특권 3
- 식별 컬럼 4
- 제한사항 28
- 호스트와 워크스테이션간 데이터 전송 220

DB2 Data Links Manager 204

db2batch를 사용하여 병렬 내보내기 6

내보내기 유틸리티 파일 형식 239

내보내기 조각

- Data Warehouse Center에서 지원 230

내보낸 테이블 재작성

- 가져오기 유틸리티 38
- 내보내기 유틸리티 5
- IXF 파일에 저장되지 않은 테이블 속성 39
- IXF 파일에 저장된 테이블 속성 38

[다]

다중 사이트 갱신 구성 마법사 325

대형 오브젝트(LOB)

- 가져오기 유틸리티 40
- 내보내기 유틸리티 5

덤프 파일

로드 유틸리티 160

데이터 레코드, PC/IXF 263

데이터 로드

- 데이터베이스 파티션에서 데이터를 로드하기 위한 AutoLoader 유틸리티 183

데이터 유형

PC/IXF 275

데이터 유형 설명

- ASC 248
- DEL 243
- PC/IXF 282

데이터 이동

플랫폼간 211

데이터 전송

- 플랫폼간 211
- 호스트와 워크스테이션간 220

AutoLoader 유틸리티 183

데이터 파티션 183

AutoLoader 유틸리티 183

데이터베이스
 복구 가능한 및 복구 불가능한, 사용할 로드 옵션 87
 웨어하우스 230
 데이터베이스 백업 마법사 325
 데이터베이스 이동 도구 214
 데이터베이스 작성 마법사 325
 데이터베이스 추가 마법사 325, 326

[라]

레코드 길이 표시기 251
 레코드 유형
 PC/IXF 251, 252
 레코드 유형, PC/IXF
 계속 271
 계층 268
 데이터 263
 색인 265
 서브테이블 269
 식별 274
 응용프로그램 265
 종료 273
 컬럼 설명자 257
 테이블 255
 헤더 253
 로그 레코드
 로드 유틸리티 161
 로드
 에 대한 파일 유형 수정자 145
 로드 메시지 파일 28, 81, 182
 로드 보류 목록 로그 레코드 161
 로드 삭제 시작 보충 로그 레코드 161
 로드 시작 로그 레코드 161
 로드 유틸리티
 가져오기 유틸리티와 비교 237
 개요 84
 덤프 파일 160
 데이터베이스 복구 87
 로그 레코드 161
 로드 단계 84

로드 유틸리티 (계속)
 변경된 구분 및 동작 89
 병렬 처리 90
 빌드 단계 85
 사용하는 데 필요한 권한 및 특권 91
 삭제 단계 85
 생성된 컬럼 96
 성능 최적화 176
 식별 컬럼 94
 실패로부터 복구 100
 예외 테이블 159
 임시 파일 160
 제한사항 181
 코드 페이지 고려사항 161
 프로세스 개요 84
 한계점 181
 DB2 Data Links Manager 209
 로드 유틸리티 파일 형식 239
 로드 조작

Data Warehouse Center에서 지원 230
 릴리스 정보 318

[마]

마법사
 다중 사이트 갱신 구성 325
 데이터베이스 백업 325
 데이터베이스 복원 326
 데이터베이스 작성 325
 데이터베이스 추가 325, 326
 색인 326
 성능 구성 326
 태스크 완료 325
 테이블 공간 작성 326
 테이블 작성 326
 매개변수
 에 대한 구분 233
 메시지 307

메시지 파일: 내보내기, 가져오기 및 로드 28, 81, 182
 명령 구분
 해석 233
 문서 서버 설정 326
 문자열 분리 문자 242

[바]

버퍼된 삽입
 가져오기 유틸리티 34
 변경된 구분 및 동작
 로드 유틸리티 89
 변수
 에 대한 구분 233
 변환기 저장 프로시저 232
 병렬 내보내기
 db2batch 사용 6
 병렬 처리
 로드 유틸리티 90
 보기
 온라인 정보 323
 보류 상태 175
 복구 불가능한 데이터베이스
 사용할 로드 옵션 87
 복구가능한 데이터베이스
 사용할 로드 옵션 87
 복원 마법사 326
 복제
 Data Warehouse Center가 지원하는
 유형 231
 분리 문자
 문자열 242
 [사]
 사용자 정의 구별 유형(UDT)
 가져오기 유틸리티 40
 상태
 로드 보류 175
 백업 보류 175

상태 (계속)
 삭제 보류 175
 점검 보류 175
 색인 레코드, PC/IXF 265
 색인 마법사 326
 샘플 프로그램
 크로스 플랫폼 317
 HTML 317
 샘플 ASC 파일 246
 샘플 DEL 파일 242
 생성된 비식별 컬럼 36, 96
 생성된 컬럼 36, 96
 서브테이블 레코드, PC/IXF 269
 설치
 Netscape 브라우저 323
 성능
 가져오기 유틸리티 80
 로드 유틸리티 176
 성능 구성 마법사 326
 수정자, 파일 유형
 가져오기 유틸리티에 대한 67
 내보내기 유틸리티에 대한 20
 로드 유틸리티에 대한 145
 식별 레코드, PC/IXF 274
 식별 컬럼 4, 34, 94

[아]

언어 식별자
 책 317
 예
 forcein 296
 예외 테이블
 로드 유틸리티 159
 오류 메시지 307
 온라인 도움말 321
 온라인 정보
 검색 327
 보기 323
 옵션
 forcein 294

완료 메시지 307
 요약 테이블
 가져오기 제한사항 80
 워크시트 파일 형식(WSF) 304
 유틸리티 파일 형식 239
 유효하지 않은 PC/IXF 데이터 유형
 281
 유효하지 않은 PC/IXF 컬럼 값 289
 유효한 PC/IXF 데이터 유형 281
 응용프로그램 레코드, PC/IXF 265
 의미론
 forcein, 데이터 유형 300
 forcein, 일반 294
 forcein, 코드 페이지 295
 입시 파일
 로드 유틸리티 117, 160
 입력된 테이블
 가져오기 유틸리티 222
 간 데이터 이동 222
 내보내기 유틸리티 222
 데이터 이동 예 225
 데이터 이동중 선택 225
 트래버스 순서 30, 224

[자]

잠금
 가져오기 유틸리티 41
 저장 프로시저
 변환기 232
 정보 센터 324
 제한조건 점검 98
 종료 레코드, PC/IXF 273

[차]

책 309, 319
 최신 정보 318

[카]

컬럼
 가져오기를 위해 지정 56

컬럼 값, 유효하지 않은 289
 컬럼 설명자 레코드, PC/IXF 257
 컬럼 식별자가 없는 ASCII(ASC) 파일
 형식 246
 컬럼 식별자가 있는 ASCII(DEL) 파일
 형식 240
 플랫폼간 데이터 이동 212
 컬럼, 호환되지 않는 289
 코드 페이지 고려사항
 가져오기 유틸리티 77
 로드 유틸리티 161
 코드 페이지 변환
 파일 290
 PC/IXF 데이터를 가져오거나 로드할
 때 290
 키워드
 에 대한 구문 233

[타]

테이블
 내보내기(export), 재작성 38
 테이블 공간 작성 마법사 326
 테이블 레코드, PC/IXF 255
 테이블 로드 삭제 시작 로그 레코드 161
 테이블 작성 마법사 326
 통합 교환 형식(IXF) 250
 트래버스 순서 30
 기본 224
 사용자 지정 224
 입력된 테이블 224
 특권
 가져오기 유틸리티에 필요한 31
 내보내기 유틸리티에 필요한 3
 로드 유틸리티에 필요한 91
 AutoLoader 유틸리티에 필요한 185

[파]

파일 유형 수정자

가져오기 유틸리티 67

내보내기 유틸리티 20

로드 유틸리티 145

파일 형식

위크쉬트(WSF) 304

컬럼 식별자가 없는

ASCII(ASC) 246

컬럼 식별자가 있는

ASCII(DEL) 240

테이블에 파일을 가져오기 위한 48

테이블을 파일로 내보내기 위한 10

IXF(PC/IXF)의 PC 버전 250

파티션 키 183

표시기

레코드 길이 251

[하]

해명 알고리즘 184

헤더 레코드, PC/IXF 253

호환되지 않는 컬럼 289

환경 변수

DB2LOADREC 100

A

anyorder 145

ASC 데이터 유형 설명 248

ASC 파일

샘플 246

형식 246

AutoLoader 유틸리티

개요 183

문제점 해결 201

사용하는 데 필요한 권한 및 특권

185

제한사항 200

AutoLoader 유틸리티 (계속)

한계점 200

B

binarynumerics 153

C

chardel 20, 73, 155

codepage 148

codel 20, 73, 155

compound 67

D

Data Warehouse Center (DWC)

개요 230

dateformat 69, 149

datesiso 20, 73, 155

DB2 Data Links Manager

가져오기 유틸리티 208

내보내기 유틸리티 204

로드 유틸리티 209

로드 유틸리티 문제점 해결 210

인스턴스간 내보내기 207

DB2 라이브러리

구조 309

마법사 325

문서 서버 설정 326

온라인 도움말 321

온라인 정보 검색 327

온라인 정보 보기 323

인쇄된 책 주문 319

정보 센터 324

책 309

책에 대한 언어 식별자 317

최신 정보 318

PDF 책 인쇄 319

db2LoadQuery - 로드 조회 139

DB2LOADREC 100

db2move 214

decplusblank 20, 73, 155

decpt 20, 73, 155

DEL 데이터 유형 설명 243

DEL 파일

샘플 242

형식 240

delprioritychar 74, 156

dldel 20, 74, 156

dumpfile 150

DWC (Data Warehouse Center) 230

F

fastparse 145

forcein 74, 157

데이터 유형 의미론 300

예 296

옵션 294

을 사용한 PC/IXF 파일 가져오기의

요약 301

일반 의미론 294

코드 페이지 의미론 295

G

generatedignore 67, 145

generatedmissing 67, 145

generatedoverride 146

H

HTML

샘플 프로그램 317

I

IBM Relational Data 복제 도구

개요 228

구성요소 229

identityignore 67, 146

identitymissing 67, 146
identityoverride 146
implieddecimal 69, 150
indexfreespace 147
indexixf 75
indexschema 75
IXF(PC/IXF) 파일 형식의 PC 버전
250

K

keepblanks 74, 156

L

LOAD
임시 파일 117
LOAD QUERY 119
LOAD QUERY (db2LoadQuery) 139
lobsinfile 20, 67, 147

N

Netscape 브라우저
설치 323
nochecklengths 72, 75, 153, 157
nodefaults 68
nodoubledel 20, 74, 156
noeofchar 69, 153
noheader 147
norowwarnings 147
no_type_id 68
nullindchar 72, 153

P

packeddecimal 154
pagefreespace 147
PC/IXF
데이터 유형 275
데이터 유형 설명 282

PC/IXF (계속)
레코드 유형 251, 252
유효하지 않은 데이터 유형 281,
288
유효하지 않은 컬럼 값 289
유효한 데이터 유형 281
코드 페이지 변환 파일 290
System/370 IXF와 대조됨 303

PC/IXF 레코드 유형

계속 271
계층 268
데이터 263
색인 265
서브테이블 269
식별 274
응용프로그램 265
종료 273
컬럼 설명자 257
테이블 255
헤더 253

PC/IXF 파일

형식 250

PC/IXF 파일 가져오기

데이터 유형 특정 규칙 290
일반 규칙 288
forcein을 사용한 301

PC/IXF 파일 가져오기를 관리하는 규칙
288, 290

PC/IXF 파일 형식

플랫폼간 데이터 이동 211

PC/IXF 파일의 가져오기

데이터 유형 특정 규칙 290
일반 규칙 288

PC/IXF와 System/370 IXF의 차이점
303

PDF 319

PDF 책 인쇄 319

R

reclen 72, 154

S

SELECT문
EXPORT 명령에 있는 11
SmartGuides
마법사 325
SQL 메시지 307
SQLCODE 307
SQLSTATE 307
SQLUIMPT-IN 구조 64
SQLUIMPT-OUT 구조 65
SQLULOAD-IN 구조 132
SQLULOAD-OUT 구조 137
SQL-UEXPT-OUT 구조 19
striptblanks 72, 154
striptnulls 72, 154
System/370 IXF 303
PC/IXF와 대조됨 303

T

timeformat 70, 151
timestampformat 71, 152
totalfreespace 147

U

usedefaults 68, 148

W

WSF 파일
형식 304
WSF 파일 형식
플랫폼간 데이터 이동 213

Z

zoneddecimal 155

IBM에 문의

기술적인 문제가 발생한 경우에는 DB2 고객 지원 센터에 문의하기 전에 문제점 해결 안내서에서 제안한 조치를 검토하고 실행해 보십시오. 이것은 DB2 고객 지원 부서로 하여금 사용자를 보다 더 잘 지원할 수 있도록 하기 위함입니다.

DB2 Universal Database 제품에 대한 정보나 주문은 그 지역의 IBM 영업 대표나 공인 IBM 소프트웨어 재판매업자에게 문의하십시오.

미국에 사시는 분은 다음 번호 중 하나를 선택하여 전화하십시오.

- 고객 지원을 받으려면, 1-800-237-5511.
- 사용가능한 서비스 옵션을 알려면, 1-888-426-4343.

제품 정보

미국에 사시는 분은 다음 번호 중 하나를 선택하여 전화하십시오.

- 제품 주문이나 일반 정보를 얻으려면, 1-800-IBM-CALL (1-800-426-2255)이나 1-800-3IBM-OS2 (1-800-342-6672).
- 책에 대한 주문은 1-800-879-2755.

<http://www.ibm.com/software/data/>

DB2 월드 와이드 웹 페이지에는 새로운 소식, 제품 설명, 교육 일정 등에 관한 현재의 DB2 정보를 제공합니다.

<http://www.ibm.com/software/data/db2/library/>

DB2 제품 및 서비스 기술 라이브러리는 빈도 높은 질문(FAQ), 수정사항(fixes), 책 및 최신 DB2 기술 정보에 대한 액세스를 제공합니다.

주: 이러한 정보는 영어로만 제공됩니다.

<http://www.elink.ibm.com/pbl/pbl/>

여기에서는 책을 웹 사이트에서 주문할 수 있는 방법을 제공합니다.

<http://www.ibm.com/education/certify/>

IBM 웹 사이트에서 기술 전문 인증 프로그램은 DB2를 포함하여 다른 IBM 제품의 기술 전문 인증 테스트 정보를 제공합니다.

<ftp.software.ibm.com>

anonymous로 로그인하십시오. /ps/products/db2 디렉토리에서, DB2와 많은 관련 제품에 관한 데이터, 수정사항, 도구 등을 찾을 수 있습니다.

[comp.databases.ibm-db2](#), [bit.listserv.db2-l](#)

이러한 인터넷 뉴스 그룹으로 사용자는 DB2 제품에 대한 자신의 사용 경험을 토론할 수 있습니다.

Compuserve에서, GO IBMDB2

IBM DB2 계열 포럼을 액세스하려면 이 명령을 입력하십시오. 모든 DB2 제품이 이러한 포럼을 통해 지원됩니다.

미국 외 지역에서 IBM에 연락하는 방법에 관한 정보는 *IBM Software Support Handbook*의 Appendix A를 참조하십시오. 이 문서에 액세스하려면, 웹 사이트 <http://www.ibm.com/support/>로 가서 페이지 밑에 있는 IBM Software Support Handbook 링크로 가십시오.

주: 일부 국가에서는 IBM 공인 딜러는 IBM 지원 센터 대신 해당 딜러 지원 부서에 연락해야 합니다.

IBM 한글 지원에 관한 설문



FAX : (02) 781-7778

보내 주시는 의견은 더 나은 고객 지원 체제를 위한 귀중한 자료가 됩니다.
독자 여러분의 좋은 의견을 기다립니다.

책 제목: IBM[®] DB2[®] Universal Database
데이터 이동 유틸리티 안내 및 참조서
버전 7

책 번호: SA30-0994-00

성 명		직위/담당업무	
회 사 명		부 서 명	
주 소			
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 대해 주셔서 감사합니다.
귀하의 의견은 저희에게 매우 소중한 것이며, 고객 여러분들께 보다 좋은 제품을 제공해 드리기 위해 최선을 다하겠습니다.



Printed in Australia

SA30-0994-00

