



## 명령어 참조서





## 명령어 참조서

**주!**

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 1333 페이지의 부록 D 『주의사항』의 정보를 읽으십시오.

**개정판 주의사항**

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

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

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

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

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

---

## 목차

이 책에 대한 정보 . . . . .	xi
이 책의 사용자 . . . . .	xi
이 책의 구성 . . . . .	xi
강조표시 규칙 . . . . .	xii

구문 다이어그램을 읽는 방법 . . . . .	xiii
---------------------------	------

---

### 제 1 부 명령행 처리기(CLP) . . . . . 1

제 1 장 명령행 처리기 기능 . . . . .	3
제 2 장 db2 - 명령행 처리기 호출 . . . . .	11
제 3 장 명령행 처리기 옵션. . . . .	13
제 4 장 명령행 처리기 리턴 코드. . . . .	23
제 5 장 명령행 처리기에서 명령 도움말 호출 . . . . .	25
제 6 장 명령행 처리기에서 메시지 도움말 호출 . . . . .	27

---

### 제 2 부 명령행 SQL문 및 XQuery 명령문 사용. . . . . 29

---

### 제 3 부 명령 구문 도움말을 읽는 방법 . . . 37

---

### 제 4 부 CLP 명령 . . . . . 41

제 7 장 ACTIVATE DATABASE. . . . .	43
제 8 장 ADD CONTACT. . . . .	45
제 9 장 ADD CONTACTGROUP. . . . .	47
제 10 장 ADD DBPARTITIONNUM . . . . .	49
제 11 장 ADD XMLSCHEMA DOCUMENT . . . . .	53
제 12 장 ARCHIVE LOG . . . . .	55
제 13 장 ATTACH . . . . .	59
제 14 장 AUTOCONFIGURE . . . . .	63
제 15 장 BACKUP DATABASE . . . . .	67

제 16 장 BIND . . . . .	77
제 17 장 CATALOG DATABASE . . . . .	99
제 18 장 CATALOG DCS DATABASE . . . . .	103
제 19 장 CATALOG LDAP DATABASE . . . . .	107
제 20 장 CATALOG LDAP NODE . . . . .	111
제 21 장 CATALOG LOCAL NODE. . . . .	113
제 22 장 CATALOG NAMED PIPE NODE . . . . .	115
제 23 장 CATALOG ODBC DATA SOURCE . . . . .	117
제 24 장 CATALOG TCPIP/TCPIP4/TCPIP6 NODE . . . . .	119
제 25 장 CHANGE DATABASE COMMENT . . . . .	123
제 26 장 CHANGE ISOLATION LEVEL . . . . .	125
제 27 장 COMPLETE XMLSCHEMA . . . . .	127
제 28 장 CREATE DATABASE. . . . .	129
제 29 장 CREATE TOOLS CATALOG. . . . .	153
제 30 장 DEACTIVATE DATABASE. . . . .	157
제 31 장 DECOMPOSE XML DOCUMENT . . . . .	159
제 32 장 DECOMPOSE XML DOCUMENT . . . . .	161
제 33 장 DEREGISTER. . . . .	163
제 34 장 DESCRIBE . . . . .	165
제 35 장 DETACH. . . . .	177
제 36 장 DROP CONTACT . . . . .	179
제 37 장 DROP CONTACTGROUP . . . . .	181
제 38 장 DROP DATABASE . . . . .	183
제 39 장 DROP DBPARTITIONNUM VERIFY . . . . .	185

제 40 장 DROP TOOLS CATALOG . . . . .	187	제 64 장 HISTORY . . . . .	281
제 41 장 ECHO . . . . .	189	제 65 장 IMPORT . . . . .	283
제 42 장 EDIT . . . . .	191	제 66 장 INITIALIZE TAPE . . . . .	315
제 43 장 EXPORT . . . . .	193	제 67 장 INSPECT . . . . .	317
제 44 장 FORCE APPLICATION . . . . .	207	제 68 장 LIST ACTIVE DATABASES . . . . .	327
제 45 장 GET ADMIN CONFIGURATION . . . . .	209	제 69 장 LIST APPLICATIONS . . . . .	329
제 46 장 GET ALERT CONFIGURATION . . . . .	211	제 70 장 LIST COMMAND OPTIONS . . . . .	333
제 47 장 GET CLI CONFIGURATION . . . . .	217	제 71 장 LIST DATABASE DIRECTORY . . . . .	335
제 48 장 GET CONNECTION STATE . . . . .	219	제 72 장 LIST DATABASE PARTITION GROUPS . . . . .	339
제 49 장 GET CONTACTGROUP . . . . .	221	제 73 장 LIST DBPARTITIONNUMS . . . . .	343
제 50 장 GET CONTACTGROUPS . . . . .	223	제 74 장 LIST DCS APPLICATIONS . . . . .	345
제 51 장 GET CONTACTS . . . . .	225	제 75 장 LIST DCS DIRECTORY . . . . .	347
제 52 장 GET DATABASE CONFIGURATION . . . . .	227	제 76 장 LIST DRDA INDOUBT TRANSACTIONS . . . . .	349
제 53 장 GET DATABASE MANAGER CONFIGURATION . . . . .	233	제 77 장 LIST HISTORY . . . . .	351
제 54 장 GET DATABASE MANAGER MONITOR SWITCHES . . . . .	239	제 78 장 LIST INDOUBT TRANSACTIONS . . . . .	355
제 55 장 GET DESCRIPTION FOR HEALTH INDICATOR . . . . .	243	제 79 장 LIST NODE DIRECTORY . . . . .	359
제 56 장 GET HEALTH NOTIFICATION CONTACT LIST . . . . .	245	제 80 장 LIST ODBC DATA SOURCES . . . . .	363
제 57 장 GET HEALTH SNAPSHOT . . . . .	247	제 81 장 LIST PACKAGES/TABLES . . . . .	365
제 58 장 GET INSTANCE . . . . .	251	제 82 장 LIST TABLESPACE CONTAINERS . . . . .	369
제 59 장 GET MONITOR SWITCHES . . . . .	253	제 83 장 LIST TABLESPACES . . . . .	371
제 60 장 GET RECOMMENDATIONS FOR HEALTH INDICATOR . . . . .	257	제 84 장 LIST UTILITIES . . . . .	385
제 61 장 GET ROUTINE . . . . .	261	제 85 장 LOAD . . . . .	387
제 62 장 GET SNAPSHOT . . . . .	263	제 86 장 LOAD QUERY . . . . .	435
제 63 장 HELP . . . . .	279	제 87 장 UPGRADE DATABASE . . . . .	443
		제 88 장 PING . . . . .	445
		제 89 장 PRECOMPILE . . . . .	449

제 90 장 PRUNE HISTORY/LOGFILE . . . . .	479	제 114 장 RUNCMD . . . . .	611
제 91 장 PUT ROUTINE . . . . .	481	제 115 장 RUNSTATS . . . . .	613
제 92 장 QUERY CLIENT. . . . .	483	제 116 장 SET CLIENT. . . . .	631
제 93 장 QUIESCE . . . . .	485	제 117 장 SET RUNTIME DEGREE . . . . .	635
제 94 장 QUIESCE TABLESPACES FOR TABLE . . . . .	489	제 118 장 SET SERVEROUTPUT 명령 . . . . .	637
제 95 장 QUIT . . . . .	493	제 119 장 SET TABLESPACE CONTAINERS . . . . .	639
제 96 장 REBIND . . . . .	495	제 120 장 SET TAPE POSITION . . . . .	643
제 97 장 RECOVER DATABASE . . . . .	499	제 121 장 SET UTIL_IMPACT_PRIORITY . . . . .	645
제 98 장 REDISTRIBUTE DATABASE PARTITION GROUP . . . . .	507	제 122 장 SET WORKLOAD 명령 . . . . .	649
제 99 장 REFRESH LDAP. . . . .	521	제 123 장 SET WRITE . . . . .	651
제 100 장 REGISTER . . . . .	523	제 124 장 START DATABASE MANAGER . . . . .	653
제 101 장 REGISTER XMLSCHEMA . . . . .	527	제 125 장 START HADR . . . . .	663
제 102 장 REGISTER XSROBJECT . . . . .	531	제 126 장 STOP DATABASE MANAGER . . . . .	667
제 103 장 REORG INDEXES/TABLE . . . . .	533	제 127 장 STOP HADR. . . . .	671
제 104 장 REORGCHK . . . . .	549	제 128 장 TAKEOVER HADR . . . . .	673
제 105 장 RESET ADMIN CONFIGURATION . . . . .	561	제 129 장 TERMINATE. . . . .	677
제 106 장 RESET ALERT CONFIGURATION . . . . .	563	제 130 장 UNCATALOG DATABASE . . . . .	679
제 107 장 RESET DATABASE CONFIGURATION . . . . .	565	제 131 장 UNCATALOG DCS DATABASE . . . . .	681
제 108 장 RESET DATABASE MANAGER CONFIGURATION . . . . .	567	제 132 장 UNCATALOG LDAP DATABASE . . . . .	683
제 109 장 RESET MONITOR . . . . .	569	제 133 장 UNCATALOG LDAP NODE . . . . .	685
제 110 장 RESTART DATABASE . . . . .	571	제 134 장 UNCATALOG NODE . . . . .	687
제 111 장 RESTORE DATABASE . . . . .	573	제 135 장 UNCATALOG ODBC DATA SOURCE . . . . .	689
제 112 장 REWIND TAPE. . . . .	595	제 136 장 UNQUIESCE . . . . .	691
제 113 장 ROLLFORWARD DATABASE . . . . .	597	제 137 장 UPDATE ADMIN CONFIGURATION . . . . .	693
		제 138 장 UPDATE ALERT CONFIGURATION . . . . .	695

제 139 장 UPDATE ALTERNATE SERVER FOR DATABASE . . . . .	701
제 140 장 UPDATE ALTERNATE SERVER FOR LDAP DATABASE . . . . .	703
제 141 장 UPDATE CLI CONFIGURATION	705
제 142 장 UPDATE COMMAND OPTIONS	707
제 143 장 UPDATE CONTACT . . . . .	709
제 144 장 UPDATE CONTACTGROUP . . . . .	711
제 145 장 UPDATE DATABASE CONFIGURATION . . . . .	713
제 146 장 UPDATE DATABASE MANAGER CONFIGURATION . . . . .	719
제 147 장 UPDATE HEALTH NOTIFICATION CONTACT LIST . . . . .	723
제 148 장 UPDATE HISTORY . . . . .	725
제 149 장 UPDATE LDAP NODE . . . . .	729
제 150 장 UPDATE MONITOR SWITCHES	731
제 151 장 UPDATE XMLSCHEMA . . . . .	735
<hr/>	
제 5 부 시스템 명령어 . . . . .	737
제 152 장 dasauto - DB2 Administration Server 자동 시작. . . . .	739
제 153 장 dasCRT - DB2 Administration Server 작성 . . . . .	741
제 154 장 dasdrop - DB2 Administration Server 제거 . . . . .	743
제 155 장 dasmigr - DB2 Administration Server 이주 . . . . .	745
제 156 장 dasupdt - DAS 갱신 . . . . .	747
제 157 장 db2_deinstall - DB2 제품, 기능 또는 언어 설치 제거. . . . .	749
제 158 장 db2_install - DB2 제품 설치 . . . . .	753

제 159 장 db2_local_ps - Linux/UNIX의 DB2 프로세스 상태 . . . . .	757
제 160 장 db2addicons - DB2 도구 명령의 주메뉴 항목 작성 . . . . .	759
제 161 장 db2admin - DB2 Administration Server . . . . .	761
제 162 장 db2adutil - TSM에서 DB2 오브젝트 관리 . . . . .	763
제 163 장 db2advis - DB2 디자인 어드바이저	773
제 164 장 db2audit - 감사 기능 관리자 도구	781
제 165 장 db2batch - 벤치마크 도구. . . . .	793
제 166 장 db2acsutil - DB2 스냅샷 백업 오브젝트 관리 명령 . . . . .	803
제 167 장 db2bfd - 바인드 파일 설명 도구. . . . .	807
제 168 장 db2ca - 구성 지원 프로그램 시작 . . . . .	809
제 169 장 db2cap - CLI/ODBC 정적 패키지 바인딩 도구 . . . . .	811
제 170 장 db2cat - 시스템 카탈로그 분석 . . . . .	813
제 171 장 db2cc - 제어 센터 시작 . . . . .	817
제 172 장 db2cfexp - 연결성 구성 익스포트 도구	819
제 173 장 db2cfimp - 연결성 구성 임포트 도구	823
제 174 장 db2chglbpath - 임베디드(embedded) 런타임 라이브러리 검색 경로 수정 . . . . .	825
제 175 장 db2chgpath - 임베디드(Embedded) 런타임 경로 변경 . . . . .	827
제 176 장 db2ckbkp - 백업 점검 . . . . .	829
제 177 장 db2ckupgrade - 데이터베이스 업그레이드 점검 . . . . .	835
제 178 장 db2ckrst - 증분 리스토어 이미지 순서 확인 . . . . .	839
제 179 장 db2cli - DB2 대화식 CLI . . . . .	843
제 180 장 db2cmd - DB2 명령 창 열기 . . . . .	845



제 181 장 db2cptsa - DB2 HA 스크립트 설치 또는 갱신 명령 . . . . .	847	제 203 장 db2gov - DB2 조정자 . . . . .	937
제 182 장 db2dart - 데이터베이스 분석 및 보고 도구 . . . . .	849	제 204 장 db2govlg - DB2 조정자 로그 쿼리	939
제 183 장 db2daslevel - DAS 레벨 표시 . . . . .	855	제 205 장 db2gpmap - 분산 맵 가져오기 . . . . .	941
제 184 장 db2dclgn - 선언 생성기 . . . . .	857	제 206 장 db2hc - Health Center 시작 . . . . .	943
제 185 장 db2diag - db2diag 로그 분석 도구	861	제 207 장 db2iauto - 인스턴스 자동 시작 . . . . .	945
제 186 장 db2drdat - DRDA 추적 . . . . .	879	제 208 장 db2iclus - Microsoft Cluster Server	947
제 187 장 db2drvmp - DB2 데이터베이스 드라이브 맵 . . . . .	881	제 209 장 db2icrt - 인스턴스 작성 . . . . .	951
제 188 장 db2empfa - 다중페이지 파일 할당 사용 . . . . .	883	제 210 장 db2idrop - 인스턴스 제거 . . . . .	957
제 189 장 db2envar.bat - 현재 명령 창의 환경 설정 . . . . .	885	제 211 장 db2IdentifyType1 - 유형 1 인텍스 식별 도구 . . . . .	961
제 190 장 db2eva - 이벤트 분석기 . . . . .	887	제 212 장 db2ilist - 인스턴스 목록 . . . . .	965
제 191 장 db2evmon - 이벤트 모니터 제작 도구	889	제 213 장 db2iupgrade - 인스턴스 업그레이드	967
제 192 장 db2evtbl - 이벤트 모니터 목표 테이블 정의 생성 . . . . .	891	제 214 장 db2inidb - 미러된 데이터베이스 초기화 . . . . .	971
제 193 장 db2exfmt - Explain 테이블 형식 . . . . .	893	제 215 장 db2inspf - 검사 결과 형식화 . . . . .	973
제 194 장 db2exmig - Explain 테이블 이주 . . . . .	897	제 216 장 db2iprone - 설치 이미지 크기 줄이기 명령 . . . . .	975
제 195 장 db2expln - SQL 및 XQuery Explain . . . . .	899	제 217 장 db2isetup - 인스턴스 작성 인터페이스 시작 . . . . .	977
제 196 장 db2extsec - DB2 오브젝트에 대한 권한 설정 . . . . .	909	제 218 장 db2iupdt - 인스턴스 갱신 . . . . .	979
제 197 장 db2flsn - 로그 시퀀스 번호 찾기 . . . . .	913	제 219 장 db2jdbcbind - DB2 JDBC 패키지 바인더 . . . . .	985
제 198 장 db2fm - DB2 결함 모니터 . . . . .	917	제 220 장 db2ldcfg - LDAP 환경 구성 . . . . .	987
제 199 장 db2fmcu - DB2 결함 모니터 제어기 명령 . . . . .	921	제 221 장 db2level - DB2 서비스 레벨 표시	989
제 200 장 db2fodc - DB2 FODC(First Occurrence Data Collection) 명령 . . . . .	923	제 222 장 db2licm - 라이선스 관리 도구 . . . . .	991
제 201 장 db2fs - 첫 단계 . . . . .	931	제 223 장 db2listvolumes - 모든 디스크 볼륨의 GUID 표시 . . . . .	995
제 202 장 db2gcf - DB2 인스턴스 제어 . . . . .	933	제 224 장 db2logsforrwd - 롤 포워드 복구에 필요한 로그 나열 . . . . .	997
		제 225 장 db2look - DB2 통계 및 DDL 추출 도구 . . . . .	999

제 226 장 db2ls - 설치된 DB2 제품 및 기능 나열 . . . . .	1011	제 245 장 db2relocatedb - 데이터베이스 재배치	1157
제 227 장 db2move - 데이터베이스 이동 도구	1015	제 246 장 db2rfpen - 롤 포워드 보류 상태 재 설정 . . . . .	1163
제 228 장 db2mqdsn - MQ 리스너 . . . . .	1027	제 247 장 db2rmicons - 주메뉴 명령에서 DB2 도구 제거 . . . . .	1165
제 229 장 db2mscs - Windows 장애 복구 유틸리티 설정 . . . . .	1031	제 248 장 db2rspgn - 응답 파일 생성 프로그램	1167
제 230 장 db2mtrk - 메모리 추적 프로그램	1035	제 249 장 db2sampl - 샘플 데이터베이스 작성	1169
제 231 장 db2nchg - 데이터베이스 파티션 서버 구성 변경 . . . . .	1041	제 250 장 db2schex - Active Directory 스키마 확장 명령 . . . . .	1173
제 232 장 db2ncrt - 인스턴스에 데이터베이스 파티션 서버 추가 . . . . .	1043	제 251 장 db2set - DB2 프로파일 레지스트리	1175
제 233 장 db2ndrop - 인스턴스에서 데이터베이스 파티션 서버 삭제 . . . . .	1045	제 252 장 db2setup - DB2 설치 . . . . .	1179
제 234 장 db2nrcfg - 루트 서버가 아닌 설치 구성 도구 명령 . . . . .	1047	제 253 장 db2sqljbind - SQLJ 프로파일 바인더 . . . . .	1181
제 235 장 db2rfe - 루트 서버가 아닌 설치에 루트 기능 사용 명령 . . . . .	1049	제 254 장 db2sqljcustomize - SQLJ 프로파일 사용자 정의 프로그램 . . . . .	1187
제 236 장 db2nrupdt - 루트 서버가 아닌 설치 인스턴스 갱신 명령 . . . . .	1051	제 255 장 db2sqljprint - SQLJ 프로파일 프린터 . . . . .	1201
제 237 장 db2nrupgrade - 루트 서버가 아닌 인스턴스 업그레이드 명령 . . . . .	1053	제 256 장 db2start - DB2 시작 . . . . .	1203
제 238 장 db2osconf - 커널 매개변수 값 유틸리티 . . . . .	1055	제 257 장 db2stat - Windows의 DB2 프로세스 상태 . . . . .	1205
제 239 장 db2pd - DB2 데이터베이스 모니터 및 문제점 해결 . . . . .	1059	제 258 장 db2stop - Stop DB2 . . . . .	1207
제 240 장 db2pdcfg - 문제점 판별 동작을 위한 DB2 데이터베이스 구성 . . . . .	1141	제 259 장 db2support - 문제점 분석 및 환경 콜렉션 도구 . . . . .	1209
제 241 장 db2perfc - 데이터베이스 성능 값 재 설정 . . . . .	1149	제 260 장 db2swtch - 디폴트 DB2 사본 및 데이터베이스 클라이언트 인터페이스 사본 전환 . . . . .	1215
제 242 장 db2perfi - 성능 카운터 등록 유틸리티 . . . . .	1151	제 261 장 db2sync - DB2 동기 프로그램 시작	1217
제 243 장 db2perfr - 성능 모니터 등록 도구	1153	제 262 장 db2systray - DB2 시스템 트레이 시작 . . . . .	1219
제 244 장 db2rbind - 모든 패키지 리바인드	1155	제 263 장 db2tapemgr - 테이프의 로그 파일 관리 . . . . .	1223
		제 264 장 db2tbst - 테이블 스페이스 상태 확보	1227
		제 265 장 db2tdbmgr - 도구 카탈로그 데이터베이스 이주 명령 . . . . .	1229

제 266 장 db2trc - 추적 . . . . .	1231	제 285 장 db2ts DISABLE DATABASE FOR TEXT . . . . .	1293
제 267 장 db2unins - DB2 데이터베이스 제품, 기능 또는 언어 설치 제거. . . . .	1239	제 286 장 db2ts DROP INDEX . . . . .	1297
제 268 장 db2untag - 컨테이너 태그 릴리스	1241	제 287 장 db2ts ENABLE DATABASE FOR TEXT . . . . .	1299
제 269 장 db2updserv - 제품 갱신사항 표시	1243	제 288 장 db2ts HELP. . . . .	1301
제 270 장 db2val - DB2 사본 유효성 확인 도구 명령. . . . .	1245	제 289 장 db2ts START FOR TEXT . . . . .	1305
제 271 장 db2xdbmig - XSR 오브젝트 이주 명령. . . . .	1247	제 290 장 db2ts STOP FOR TEXT . . . . .	1307
제 272 장 db2xpvt - 트랩 파일 형식화. . . . .	1249	제 291 장 db2ts UPDATE INDEX . . . . .	1309
제 273 장 disable_MQFunctions - WebSphere MQ 함수 비활성화 명령 . . . . .	1251	<hr/>	
제 274 장 doce_deinstall - DB2 정보 센터 설치 제거. . . . .	1253	제 7 부 부록 . . . . .	1313
제 275 장 doce_install - DB2 정보 센터 설치	1255	부록 A. 이름 지정 규칙 . . . . .	1315
제 276 장 enable_MQFunctions - WebSphere MQ 함수 사용 명령 . . . . .	1257	이름 지정 규칙 . . . . .	1315
제 277 장 installFixPack - 설치된 DB2 제품 갱신. . . . .	1261	부록 B. 파일 유형 수정자 및 분리문자 . . . . .	1317
제 278 장 setup - DB2 설치 . . . . .	1265	익스포트, 임포트 및 로드 유틸리티의 파일 유형 수정자 . . . . .	1317
제 279 장 sqlj - SQLJ 변환기 . . . . .	1267	데이터 이동 시 분리문자 고려사항. . . . .	1317
<hr/>		부록 C. DB2 기술 정보 개요 . . . . .	1321
제 6 부 DB2 텍스트 검색 명령 . . . . .	1271	DB2 기술 라이브러리(하드카피 또는 PDF 형식)	1322
제 280 장 db2ts ALTER INDEX . . . . .	1273	인쇄된 DB2 서적 주문. . . . .	1324
제 281 장 db2ts CLEANUP FOR TEXT	1277	명령행 처리기에서 SQL 상태 도움말 표시 . . . . .	1325
제 282 장 db2ts CLEAR COMMAND LOCKS . . . . .	1279	DB2 정보 센터의 다른 버전에서 액세스 . . . . .	1326
제 283 장 db2ts CLEAR EVENTS FOR INDEX. . . . .	1281	DB2 정보 센터에서 원하는 언어로 항목 표시	1326
제 284 장 db2ts CREATE INDEX . . . . .	1285	컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 갱신 . . . . .	1327
		컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 수동 갱신 . . . . .	1328
		DB2 자습서 . . . . .	1330
		DB2 문제점 해결 정보. . . . .	1331
		이용약관 . . . . .	1331
		부록 D. 주의사항 . . . . .	1333
		색인 . . . . .	1337



---

## 이 책에 대한 정보

이 책에서는 데이터베이스 관리 기능을 실행하는 IBM® DB2® 명령행 처리기(CLP) 및 시스템 명령에 대한 사용 정보를 제공합니다.

---

## 이 책의 사용자

이 책의 사용자는 데이터베이스 관리에 대해 이해하고 구조적 쿼리 언어(SQL)에 대한 지식이 있다고 가정합니다.

---

## 이 책의 구성

이 책은 CLP, 시스템 및 DB2 텍스트 검색 명령을 사용하는 데 필요한 참조 정보를 제공합니다.

이 책에서 설명하는 주요 주제 영역은 다음과 같습니다.

### 명령행 처리기(CLP) 사용

- 제1장 『명령행 처리기(CLP)』에서는 명령행 처리기를 호출하고 사용하는 방법과 CLP 옵션에 대해 설명합니다.
- 제2장 『명령행 SQL문 및 XQuery문 사용』에서는 명령행에서 SQL문을 사용하는 방법에 대해 설명합니다.

### 명령 사용법

- 제3장 『명령 도움말 화면 읽는 방법』에서는 명령 도움말 화면을 호출하는 방법과 명령 도움말 화면 구문 규칙에 대해 설명합니다.

### CLP 명령

- 제4장 『CLP 명령』에서는 알파벳 순으로 나열된 데이터베이스 관리 프로그램 명령에 대해 설명합니다.

### 시스템 명령어

- 제5장 『시스템 명령』에서는 데이터베이스 관리 프로그램에 액세스하는 셸 스크립트에 있거나 운영 체제 명령 프롬프트에 입력할 수 있는 알파벳 순으로 나열된 명령에 대해 설명합니다.

### DB2 텍스트 검색 명령

- 제6장 『DB2 텍스트 검색 명령』에서는 앞에 db2ts가 붙은 운영 체제 명령 프롬프트에 입력할 수 있는 알파벳 순으로 나열된 텍스트 검색 명령에 대해 설명합니다.

### 부록

- 부록 A 『이름 지정 규칙』에서는 데이터베이스, 테이블 등의 오브젝트 이름을 지정하는 데 사용된 규칙에 대해 설명합니다.
- 부록 B 『파일 유형 수정자 및 분리문자』에서는 로드, импорт 및 익스포트 유틸리티를 위한 파일 유형 수정자에 대해 설명합니다. 데이터 이동을 위한 분리문자 주의 사항에 대해서도 설명합니다.

---

## 강조표시 규칙

이 책에서 사용되는 강조표시 규칙은 다음과 같습니다.

굵게	명령, 키워드 및 시스템에 의해 사전 정의된 기타 아이템 이름을 표시합니다. 대문자로 작성된 명령은 CLP 명령이고 소문자로 작성된 명령은 시스템 명령입니다.
기울임꼴	다음 중 하나를 표시합니다. <ul style="list-style-type: none"> <li>• 사용자가 제공해야 되는 이름이나 값(변수)</li> <li>• 일반 강조</li> <li>• 새 용어 소개</li> <li>• 다른 정보 소스에 대한 참조</li> </ul>
모노스페이스	다음 중 하나를 표시합니다. <ul style="list-style-type: none"> <li>• 파일 및 디렉토리</li> <li>• 명령 프롬프트나 창에 입력하도록 지시한 정보</li> <li>• 특정 데이터 값의 예</li> <li>• 시스템에서 표시할 수 있는 텍스트와 유사한 텍스트 예</li> <li>• 시스템 메시지 예</li> <li>• 프로그래밍 코드의 샘플</li> </ul>

---

## 구문 다이어그램을 읽는 방법

구문은 다음과 같은 구조를 사용하여 설명합니다.

라인의 경로를 따라서 왼쪽에서 오른쪽으로 및 위에서 아래로 구문 다이어그램을 읽으십시오.

▶— 기호는 구문 다이어그램의 시작을 표시합니다.

—▶ 기호는 구문이 다음 라인에서 계속됨을 표시합니다.

▶— 기호는 구문이 이전 라인에서 계속됨을 표시합니다.

—▶ 기호는 구문 다이어그램의 끝을 표시합니다.

구문 조각은 |— 기호로 시작하고 —| 기호로 끝납니다.

필수 항목은 수평선(주 경로)에 나타납니다.

▶—*required\_item*—————▶

선택 항목은 기본 경로 아래에 나타납니다.

▶—*required\_item*—*optional\_item*—————▶

선택적 항목이 주 경로 위에 나타나는 경우 해당 항목은 실행 시 적용되지 않으며 관독성을 위해서만 사용됩니다.

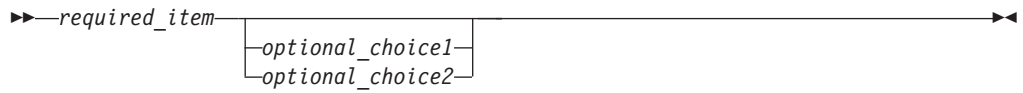
▶—*required\_item*—*optional\_item*—————▶

둘 이상의 항목에서 선택할 수 있는 경우 해당 항목은 스택으로 나타납니다.

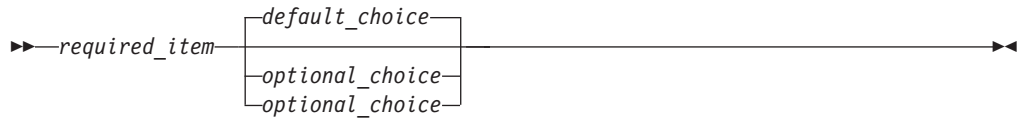
항목 중 하나를 반드시 선택해야 하는 경우 스택 중 하나의 항목이 주 경로에 나타납니다.

▶—*required\_item*—*required\_choice1*  
—*required\_choice2*—————▶

항목 중 하나의 선택이 선택적인 경우 전체 스택이 주 경로 아래에 나타납니다.



항목 중 하나가 디폴트인 경우 해당 항목은 주 경로 위에 나타나고 나머지 선택사항은 아래에 표시됩니다.



주 라인 위에서 왼쪽으로 돌아오는 화살표는 반복될 수 있는 항목을 표시합니다. 이런 경우, 반복되는 항목은 하나 이상의 공백으로 분리되어야 합니다.



반복 화살표가 쉼표를 포함하는 경우 반복되는 항목을 쉼표로 구분해야 합니다.

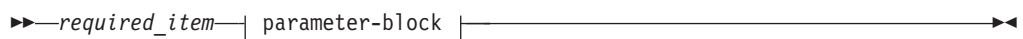


스택 위에 있는 반복 화살표는 스택된 항목에서 둘 이상을 선택하거나 단일 선택사항을 반복할 수 있음을 표시합니다.

키워드는 대문자로 나타냅니다(예: FROM). 키워드는 표시된 대로 정확하게 입력해야 합니다. 변수는 소문자로 나타냅니다(예: column-name). 구문에서 사용자가 제공하는 이름이나 값을 나타냅니다.

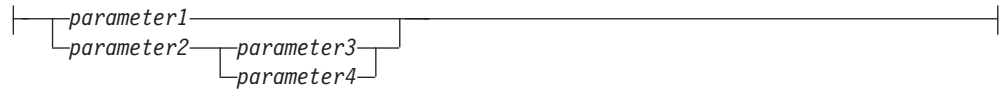
구두점, 괄호, 산술 연산자 또는 기호 같은 기타가 표시되는 경우 이들을 구문의 일부로 입력해야 합니다.

가끔 단일 변수가 구문의 더 큰 조각을 나타냅니다. 예를 들어 다음 다이어그램에서 parameter-block 변수는 **parameter-block**으로 레이블되는 전체 구문 조각을 나타냅니다.





**parameter-block:**



『큰 클머리표』(●) 사이에 발생하는 인접한 세그먼트는 임의의 순서로 지정할 수 있습니다.



위의 다이어그램은 item2 및 item3이 어느 순서로든 지정할 수 있음을 표시합니다. 다음의 두 가지가 모두 유효합니다.

```
required_item item1 item2 item3 item4  
required_item item1 item3 item2 item4
```



---

## 제 1 부 명령행 처리기(CLP)



---

## 제 1 장 명령행 처리기 기능

명령행 처리기는 다음과 같이 실행됩니다.

- CLP 명령(대소문자 구분 안함)이 명령 프롬프트에 입력됩니다.
- 명령은 ENTER 키를 눌러 명령 셸로 보냅니다.
- 출력 방향은 자동으로 표준 출력 디바이스로 지정됩니다.
- 파이핑과 경로 재지정이 지원됩니다.
- 성공 및 실패 완료가 사용자에게 통지됩니다.
- 명령을 실행하면 제어가 운영 체제 명령 프롬프트로 리턴되어 사용자가 명령을 더 입력할 수 있습니다.
- 파일 입력 옵션으로 CLP가 호출되면 CLIENT APPLNAME 특수 레지스터를 자동으로 CLP *filename*으로 설정합니다.

다음 중 하나로 명령행 처리기를 시작할 수 있습니다.

- db2 명령을 입력하거나
- Linux<sup>®</sup> 운영 체제에서 주메뉴를 누르고 **IBM DB2** → 명령행 처리기를 선택하십시오.

특정 CLP 명령과 SQL문에서는 서버 인스턴스가 실행 중이고 데이터베이스 연결이 존재해야 합니다. 다음 중 하나를 수행하여 데이터베이스에 연결하십시오.

- 다음 SQL문을 실행합니다.  
`db2 connect to database`
- DB2 레지스트리 변수 **DB2DBDFT**로 정의한 디폴트 데이터베이스에 대한 내재된 연결을 설정합니다.

명령이 명령 프롬프트에서 허용되는 문자 한계를 초과하면 백슬래시(#)를 라인 연속 문자로서 사용할 수 있습니다. 명령행 처리기가 라인 연속 문자를 만나면 다음 라인을 읽고 두 라인 모두에 포함된 문자를 병합합니다. 또는 **-t** 옵션을 사용하여 다른 라인 종료 문자를 설정할 수 있습니다.

명령행 처리기는 NULL 문자열을 널(NULL) 문자열로 인식합니다. 이전에 특정 값으로 설정된 필드를 나중에 NULL로 설정할 수 있습니다. 예를 들면, 다음과 같습니다.

```
db2 update database manager configuration using tm_database NULL
```

**tm\_database** 필드를 NULL로 설정합니다. 이 조작은 대소문자를 구분합니다. 소문자 null은 널(NULL) 문자열로 해석되지 않고 null 문자가 포함된 문자열로 해석됩니다.

## 명령행 처리기 사용자 정의

**DB2\_CLPPROMPT** 레지스트리 변수를 사용하여 대화식 입력 프롬프트를 사용자 정의할 수 있습니다. 이 레지스트리 변수는 최대 길이가 100인 임의의 텍스트 문자열로 설정할 수 있으며 %i, %ia, %d, %da 및 %n 토큰을 포함할 수 있습니다. 런타임에서 이러한 토큰이 특정 값으로 대체됩니다.

표 1. DB2\_CLPPROMPT 토큰 및 런타임 값

DB2_CLPPROMPT 토큰	런타임 값
%ia	현재 인스턴스 첨부 권한 부여 ID
%i	현재 첨부된 인스턴스의 로컬 별명. 인스턴스 첨부가 없는 경우에는 <b>DB2INSTANCE</b> 레지스트리 변수 값. Windows® 플랫폼 전용: <b>DB2INSTANCE</b> 레지스트리 변수가 설정되지 않은 경우에는 <b>DB2INSTDEF</b> 레지스트리 변수 값.
%da	현재 데이터베이스 연결의 권한 부여 ID
%d	현재 연결된 데이터베이스의 로컬 별명. 데이터베이스 연결이 없는 경우 <b>DB2DBDFT</b> 레지스트리 변수 값.
%n	줄 바꾸기

- 런타임에서 토큰이 값과 연관되지 않으면 해당 토큰이 비어 있는 문자열로 대체됩니다.
- 대화식 입력 프롬프트는 항상 권한 부여 ID, 데이터베이스 이름 및 인스턴스 이름이 프롬프트에 표시된 연결 및 첨부 정보와 일치하도록 대문자로 표시합니다.
- **DB2\_CLPPROMPT** 레지스트리 변수가 CLP 대화식 모드에서 변경되는 경우 CLP 대화식 모드가 닫힌 후 다시 열려야 **DB2\_CLPPROMPT**의 새 값이 적용됩니다.

**DB2\_CLPHISTSZIE** 레지스트리 변수를 사용하여 명령 실행기록에 저장할 명령 수를 지정할 수 있습니다. HISTORY 명령을 사용하면 CLP 대화식 모드 세션에서 실행하는 명령 실행기록의 콘텐츠에 액세스할 수 있습니다.

**DB2\_CLP\_EDITOR** 레지스트리 변수를 사용하여 EDIT 명령을 발행할 때 열리는 편집기를 지정할 수도 있습니다. CLP 대화식 세션에서 EDIT 명령을 사용하여 사용자가 지정한 명령이 사전 로드된 편집기를 열어서 편집 및 실행할 수 있습니다.

예:

**DB2\_CLPPROMPT**가 (%ia@%i, %da@%d)로 정의되는 경우 입력 프롬프트는 다음과 같은 값을 갖습니다.

- 인스턴스 첨부와 데이터베이스 연결이 없음. **DB2INSTANCE**가 DB2로 설정. **DB2DBDFT**는 설정되지 않음.  
(@DB2, @)
- (Windows) 인스턴스 첨부와 데이터베이스 연결이 없음. **DB2INSTANCE** 및 **DB2DBDFT**는 설정되지 않음. **DB2INSTDEF**가 DB2로 설정.

- (@DB2, @)
- 인스턴스 첨부과 데이터베이스 연결이 없음. **DB2INSTANCE**가 DB2로 설정. **DB2DBDFT**가 "SAMPLE"로 설정.  
(@DB2, @SAMPLE)
- 권한 부여 ID "keon14"를 사용하는 인스턴스 "DB2"에 대한 인스턴스 첨부. **DB2INSTANCE**가 DB2로 설정. **DB2DBDFT**가 "SAMPLE"로 설정.  
(KEON14@DB2, @SAMPLE)
- 권한 부여 ID "horton7"을 사용하는 "sample" 데이터베이스에 대한 데이터베이스 연결. **DB2INSTANCE**가 DB2로 설정. **DB2DBDFT**가 SAMPLE로 설정.  
(@DB2, HORTON7@SAMPLE)
- 권한 부여 ID "keon14"를 사용하는 인스턴스 "DB2"에 대한 인스턴스 첨부. 권한 부여 ID "horton7"을 사용하는 "sample" 데이터베이스에 대한 데이터베이스 연결. **DB2INSTANCE**가 DB2로 설정. **DB2DBDFT**는 설정되지 않음.  
(KEON14@DB2, HORTON7@SAMPLE)

## 명령 파일에서 명령행 처리기 사용

데이터베이스 관리 프로그램에 대한 CLP 요청은 셸 스크립트 명령 파일에 임베드(imbed)될 수 있습니다. 다음 예는 셸 스크립트 명령 파일에 CREATE TABLE문을 입력하는 방법을 나타냅니다.

```
db2 "create table mytable (name VARCHAR(20), color CHAR(10))"
```

명령 및 명령 파일에 대한 자세한 정보는 해당 운영 체제 매뉴얼을 참조하십시오.

## 명령행 처리기 디자인

명령행 처리기는 두 개의 프로세스로 구성됩니다. 프론트엔드 프로세스(DB2 명령)는 사용자 인터페이스로 사용되며 백엔드 프로세스(db2bp)는 데이터베이스 연결을 유지합니다.

### 데이터베이스 연결 유지

db2가 호출될 때마다 새 프론트엔드 프로세스가 시작됩니다. 백엔드 프로세스는 첫 번째 db2 호출로 시작되며 TERMINATE로 명시적으로 종료될 수 있습니다. 순위가 같은 모든 프론트엔드 프로세스는 단일 백엔드 프로세스로 지원되므로 단일 데이터베이스 연결을 공유합니다.

예를 들어, 동일한 운영 체제 명령 프롬프트에서 다음 db2를 호출하면 개별 프론트엔드 프로세스가 단일 백엔드 프로세스를 공유하여 데이터베이스 연결 처리량이 유지됩니다.

- db2 'connect to sample',

- db2 'select \* from org',
- . foo (여기서 foo는 DB2 명령을 포함하는 쉘 스크립트) 및
- db2 -tf myfile.clp

동일한 운영 체제 프롬프트에서 다음을 호출하면 개별 데이터베이스 연결이 생성됩니다. 각 연결은 개별 상위 프로세스를 갖기 때문입니다. 따라서 개별 백엔드 프로세스는 다음과 같습니다.

- foo
- . foo &
- foo &
- sh foo

### 프론트엔드 및 백엔드 프로세스 사이의 통신

프론트엔드 프로세스와 백엔드 프로세스는 세 개 메시지 큐(요청 큐, 입력 큐 및 출력 큐)를 통해 통신합니다.

### 환경 변수

다음 환경 변수는 두 프로세스 간 통신 구성 방법을 제공합니다.

표 2. 환경 변수

변수	최소값	최대값	디폴트값
<b>DB2BQTIME</b>	1초	5294967295	1초
<b>DB2BQTRY</b>	0회 시도	5294967295	60회 시도
<b>DB2RQTIME</b>	1초	5294967295	5초
<b>DB2IQTIME</b>	1초	5294967295	5초

### DB2BQTIME

명령행 처리가 호출되면 프론트엔드 프로세스가 백엔드 프로세스를 이미 사용 중인지 여부를 점검합니다. 백엔드 프로세스를 사용하는 경우 프론트엔드 프로세스는 해당 연결을 재설정합니다. 백엔드 프로세스를 사용하지 않는 경우 프론트엔드 프로세스가 활성화합니다. 그러면 프론트엔드 프로세스는 **DB2BQTIME** 변수로 지정한 지속기간 동안 유휴 상태를 유지하고 다시 점검합니다. 프론트엔드 프로세스는 **DB2BQTRY** 변수로 지정된 횟수만큼 점검합니다. 이 횟수를 초과한 후에도 백엔드 프로세스가 활성화되지 않으면 시간종료되고 오류 메시지가 리턴됩니다.

### DB2BQTRY

**DB2BQTIME** 변수와 함께 사용되며 프론트엔드 프로세스가 백엔드 프로세스 활성화 여부를 판별하기 위해 시도하는 횟수를 지정합니다.



쿼리 시간의 최적화를 위해 최대 사용 시간 동안은 **DB2BQTIME** 및 **DB2BQTRY** 값을 늘릴 수 있습니다.

### **DB2RQTIME**

백엔드 프로세스가 시작되면 프론트엔드의 요청에 대한 해당 요청 큐에서 대기합니다. 또한 명령 프롬프트에서 시작된 요청 간에 요청 큐에서 대기합니다.

**DB2RQTIME** 변수는 백엔드 프로세스가 프론트엔드 프로세스의 요청을 대기하는 시간을 지정합니다. 이 시간이 종료되고 요청 큐에 요청이 없으면 백엔드 프로세스가 프론트엔드 프로세스의 순위가 계속 있는지 여부를 점검하고 순위가 없으면 자동으로 종료됩니다. 그렇지 않으면 요청 큐에서 계속 대기합니다.

### **DB2IQTIME**

백엔드 프로세스에서 프론트엔드 프로세스의 요청을 수신하면 입력 큐를 통해 입력을 수신할 수 있음을 나타내는 수신확인을 프론트엔드 프로세스로 보냅니다. 그런 다음 백엔드 프로세스가 입력 큐에서 대기합니다. 또한 일괄처리 파일 (-f 옵션으로 지정됨)이 실행될 때와 사용자가 대화식 모드일 때 입력 큐에서 대기합니다.

**DB2IQTIME** 변수는 프론트엔드 프로세스가 명령을 전달하도록 백엔드 프로세스가 입력 큐를 대기하는 시간을 지정합니다. 이 시간이 경과되면 백엔드 프로세스가 프론트엔드 프로세스를 사용 중인지 여부를 점검하고 프론트엔드 프로세스가 더 이상 존재하지 않으면 요청 큐에서 다시 대기합니다. 또는 백엔드 프로세스가 프론트엔드 프로세스의 입력을 계속 대기합니다.

이러한 환경 변수 값을 보려면 LIST COMMAND OPTIONS를 사용하십시오.

백엔드 환경 변수는 백엔드 프로세스가 시작될 때 프론트엔드 프로세스에서 설정된 값을 상속합니다. 그러나 프론트엔드 환경 변수가 변경되면 백엔드 프로세스가 이러한 변경사항을 상속하지 않습니다. 백엔드 프로세스를 먼저 종료한 다음 db2 명령을 실행하여 재시작해야 변경된 값을 상속합니다.

다음 시나리오는 백엔드 프로세스를 종료해야 하는 경우의 한 예입니다.

1. 사용자 A가 로그인하여 몇몇 CLP 명령을 실행한 다음 TERMINATE를 실행하지 않고 로그오프합니다.
2. 사용자 B가 동일한 창을 사용하여 로그인합니다.
3. 사용자 B가 특정 CLP 명령을 실행하면 DB21016 메시지(시스템 오류)와 함께 실패합니다.

사용자 A가 시작한 백엔드 프로세스는 사용자 B가 CLP를 사용하기 시작할 때도 계속 사용됩니다. 사용자 B의 프론트엔드 프로세스의 순위(명령이 실행된 운영 시스템 창)를 계속 사용하기 때문입니다. 백엔드 프로세스는 사용자 B가 실행한 새 명령을 지원하도록 시도합니다. 그러나 사용자 B의 프론트엔드 프로세스에는 백엔드 프로세스의 메시지 큐를 사용할 수 있는 충분한 권한이 없습니다. 해당 백엔드 프로세스를 작성한 사

용자 A의 권한이 필요하기 때문입니다. CLP 세션은 사용자가 동일한 운영 체제 창을 사용하여 새 CLP 세션을 시작하기 전에 TERMINATE 명령으로 끝나야 합니다. 이 세션은 각각의 새로운 사용자에게 대해 백엔드 프로세스를 작성하여 권한으로 인한 문제를 방지하며 새 사용자의 백엔드 프로세스에서 환경 변수(예: **DB2INSTANCE**)에 대한 올바른 값을 설정합니다.

## CLP 사용 시 참고사항

명령은 명령 프롬프트에서 대문자 또는 소문자로 입력할 수 있습니다. 그러나 DB2에서 대소문자를 구분하는 매개변수는 원하는 정확한 대소문자로 입력해야 합니다. 예를 들어, **CHANGE DATABASE COMMENT** 명령의 WITH 절에 있는 *comment-string* 은 대소문자를 구분하는 매개변수입니다.

SQL문에서는 분리 ID가 허용됩니다.

CLP 명령에서는 특수 문자 또는 메타 문자(예: \$ & \* ( ) ; < > ? # ' ")가 허용됩니다. 이러한 문자를 CLP 대화식 모드 또는 CLP 일괄처리 입력 모드 외부에서 사용하면 운영 체제 셸이 해석합니다. 셸이 특수 조치를 수행하지 않는 경우 인용 부호 또는 Escape 문자가 필요합니다.

예를 들어, AIX® 콘 셸 환경에서 실행되는 경우,

```
db2 select * from org where division > 'Eastern'
```

위의 내용은 "select <the names of all files> from org where division"으로 해석됩니다. 결과인 SQL 구문 오류의 경로는 Eastern 파일로 재지정됩니다. 다음 구문은 올바른 출력을 생성합니다.

```
db2 "select * from org where division > 'Eastern'"
```

특수 문자는 플랫폼마다 다릅니다. AIX 콘 셸에서는 Escape 문자(#)(예: #\*, #> 또는 #')를 사용하여 위의 예를 다시 쓸 수 있습니다.

대부분의 운영 체제 환경에서는 입출력 경로를 재지정할 수 있습니다. 예를 들어, SAMPLE 데이터베이스에 연결한 경우 다음 요청은 STAFF 테이블을 쿼리하고 mydata 디렉토리의 staflist.txt 파일로 출력을 보냅니다.

```
db2 "select * from staff" > mydata/staflist.txt
```

출력 경로 재지정이 지원되지 않는 환경의 경우 CLP 옵션을 사용할 수 있습니다. 예를 들어, 요청은 다음과 같이 다시 쓸 수 있습니다.

```
db2 -r mydata#staflist.txt "select * from staff"
```

```
db2 -z mydata#staflist.txt "select * from staff"
```

명령행 처리기는 프로그래밍 언어가 아닙니다. 예를 들어, 호스트 변수를 지원하지 않으며 다음 명령문은

db2 connect to :HostVar in share mode

구문이 올바르지 않습니다. :HostVar은 유효한 데이터베이스 이름이 아니기 때문입니다.

명령행 처리기는 SQL NULL 값을 하이픈(-)으로 나타냅니다. 컬럼이 숫자인 경우 하이픈은 컬럼 오른쪽에 있습니다. 컬럼이 숫자가 아닌 경우 하이픈은 왼쪽에 있습니다.

DB2 명령행 처리기 창에서 1바이트의 자국 문자(SBCS) 언어를 올바르게 표시하려면 트루 타입 글꼴을 선택해야 합니다. 예를 들어, Windows 환경의 경우 명령 창 등록 정보 노트북을 열고 Lucinda Console과 같은 글꼴을 선택하십시오.

명령행 처리기는 파일 경로 이름에서 자국어 지원(NLS) 문자를 지원하지 않습니다. 이는 특히 문제점이 있는 파일 경로 이름이 가장 자주 나타나는 IMPORT, EXPORT 및 REGISTER XMLSCHEMA와 같은 명령에 영향을 줍니다.



## 제 2 장 db2 - 명령행 처리기 호출

db2 명령은 CLP(명령행 처리기)를 시작합니다. CLP는 데이터베이스 유틸리티, SQL문 및 온라인 도움말을 실행하는 데 사용됩니다. 또한 다음 모드로 시작할 수 있는 다양한 명령 옵션을 제공합니다.

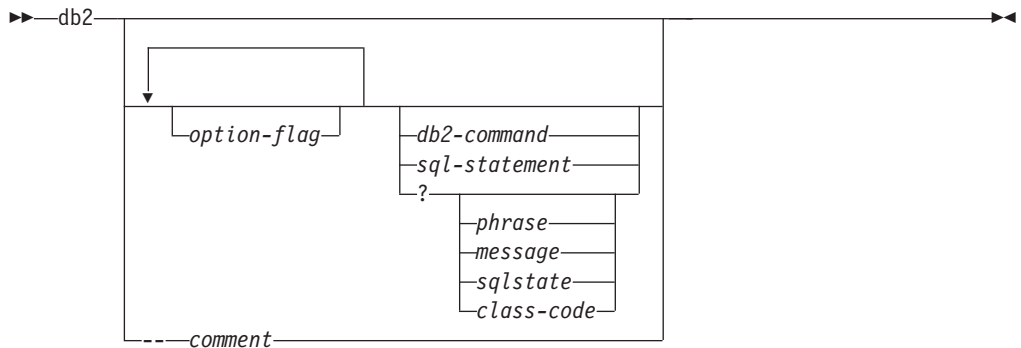
- **db2 =>** 입력 프롬프트 특성을 갖는 대화식 입력 모드
- 각 명령 앞에 db2가 붙어야 하는 명령 모드
- **-f** 파일 입력 옵션을 사용하는 일괄처리 모드.

Windows 운영 체제에서 db2cmd는 CLP 사용 DB2 창을 열고 DB2 명령행 환경을 초기화합니다. 이 명령을 실행하는 것은 DB2 명령 창 아이콘을 누르는 것과 같습니다.

QUIT는 명령행 처리기를 중지합니다. TERMINATE 또한 명령행 처리기를 중지하지만 연관된 백엔드 프로세스를 제거하고 사용되는 메모리를 비웁니다. TERMINATE는 모든 STOP DATABASE MANAGER(db2stop) 명령 이전에 실행되어야 합니다. 또한 이러한 변경사항을 적용하려면 데이터베이스 구성 매개변수가 변경된 후 TERMINATE를 실행해야 합니다. 기존 연결은 CLP를 종료하기 전에 재설정해야 합니다.

셸 명령(!)을 사용하면 UNIX® 기반 시스템 및 Windows 운영 체제에서 운영 체제 명령을 대화식 또는 일괄처리 모드로 실행할 수 있습니다(예: UNIX의 경우 !ls, Windows 운영 체제의 경우 !dir).

### 명령 구문



#### option-flag

CLP 옵션 플래그를 지정합니다.

#### db2-command

DB2 명령을 지정합니다.

**sql-statement**

SQL문을 지정합니다.

? CLP 일반 도움말을 요청합니다.

**? phrase**

지정된 명령 또는 주제와 연관된 도움말 텍스트를 요청합니다. 데이터베이스 관리 프로그램이 요청된 정보를 찾을 수 없는 경우 일반 도움말 화면을 표시합니다.

? options는 CLP 옵션의 현재 설정과 설명을 요청합니다. ? help는 온라인 도움말 구문 다이어그램 읽기에 대한 정보를 요청합니다.

**? message**

유효한 SQLCODE로 지정된 메시지에 대한 도움말을 요청합니다(예를 들어, ? sql10007n).

**? sqlstate**

유효한 SQLSTATE로 지정된 메시지에 대한 도움말을 요청합니다.

**? class-code**

유효한 클래스 코드로 지정된 메시지에 대한 도움말을 요청합니다.

**-- comment**

주석 문자 --로 시작되는 입력은 명령행 처리기에서 주석으로 간주됩니다.

모든 경우 공백으로 변수 이름과 물음표(?)를 구분해야 합니다.

## 제 3 장 명령행 처리기 옵션

CLP 명령 옵션은 명령행 처리기 DB2OPTIONS 환경 변수(대문자여야 함)를 설정하거나 명령행 플래그를 사용하여 지정할 수 있습니다.

사용자는 DB2OPTIONS를 사용하여 전체 세션에 대한 옵션을 설정할 수 있습니다.

LIST COMMAND OPTIONS를 사용하여 DB2OPTIONS 값과 옵션 플래그에 대한 현재 설정을 봅니다. UPDATE COMMAND OPTIONS를 사용하여 대화식 입력 모드 또는 명령 파일에서 옵션 설정을 변경합니다.

명령행 처리기는 다음 순서로 옵션을 설정합니다.

1. 디폴트 옵션을 설정합니다.
2. DB2OPTIONS를 읽어 디폴트값을 겹쳐씹니다.
3. 명령행을 읽어 DB2OPTIONS를 겹쳐씹니다.
4. UPDATE COMMAND OPTIONS의 입력을 최종 대화식 겹쳐쓰기로 승인합니다.

표 3은 CLP 옵션 플래그를 요약 설명합니다. 이러한 옵션은 순서와 조합에 관계 없이 지정할 수 있습니다. 옵션을 활성화하려면 해당 옵션 문자에 마이너스 부호(-)를 접두부로 추가하십시오. 옵션을 비활성화하려면 옵션 문자에 마이너스 부호를 접두부로 추가하고 옵션 문자 뒤에 마이너스 부호를 더 추가하거나 옵션 문자에 플러스 부호(+)를 접두부로 추가하십시오. 예를 들어, -c는 자동 커밋 옵션을 활성화하고 -c- 또는 +c는 옵션을 비활성화합니다. 이러한 옵션은 대소문자를 구분하지 않습니다. 즉, -a와 -A는 같습니다.

표 3. CLP 명령 옵션

옵션 플래그	설명	디폴트 설정
-a	이 옵션은 명령행 처리기에서 SQLCA 데이터를 표시하도록 지시합니다.	OFF
-c	이 옵션은 명령행 처리기에서 SQL문을 자동으로 커밋하도록 지시합니다.	ON
-d	이 옵션은 명령행 처리기에서 XML 데이터의 XML 선언을 검색 및 표시하도록 지시합니다.	OFF
-e{cls}	이 옵션은 명령행 처리기에서 SQLCODE 또는 SQLSTATE를 표시하도록 지시합니다. 이 옵션은 상호 배타적입니다.	OFF
-f <i>filename</i>	이 옵션은 명령행 처리기에서 표준 입력 대신 파일의 명령 입력을 읽도록 지시합니다.	OFF
-i	이 옵션은 명령행 처리기에서 올바른 들여쓰기로 XML 데이터를 '고급 인쇄'하도록 지시합니다. 이 옵션은 XQuery 명령문의 결과 세트에만 영향을 줍니다.	OFF

표 3. CLP 명령 옵션 (계속)

옵션 플래그	설명	디폴트 설정
-l <i>filename</i>	이 옵션은 명령행 처리기에서 실행기록 파일에 명령을 로깅하도록 지시합니다.	OFF
-m	이 옵션은 명령행 처리기에서 INSERT/DELETE/UPDATE/MERGE에 대해 영향을 받은 행 수를 인쇄하도록 지시합니다.	OFF
-n	단일 분리 토큰에서 개행 문자를 제거합니다. 이 옵션이 지정되지 않으면 개행 문자가 공백으로 바뀝니다. 이 옵션은 -t 옵션과 함께 사용해야 합니다.	OFF
-o	이 옵션은 명령행 처리기에서 출력 데이터와 메시지를 표준 출력에 표시하도록 지시합니다.	ON
-p	이 옵션은 명령행 처리기에서 명령행 처리기 프롬프트를 대화식 입력 모드로 표시하도록 지시합니다.	ON
-q	이 옵션은 명령행 처리기에서 문자열의 공백과 개행 문자를 작은 따옴표 또는 큰따옴표로 구분하여 보존하도록 지시합니다. q 옵션이 ON이면 n 옵션은 무시됩니다.	OFF
-r <i>filename</i>	이 옵션은 명령행 처리기에서 명령으로 생성된 보고서를 파일에 기록하도록 지시합니다.	OFF
-s	이 옵션은 명령을 일괄처리 파일 또는 대화식 모드로 실행하는 동안 오류가 발생하는 경우 명령행 처리기에서 실행을 중지하도록 지시합니다.	OFF
-t	이 옵션은 명령행 처리기에서 세미콜론(;)을 명령문 종료 문자로 사용하도록 지시합니다.	OFF
-tdx 또는 -tdxx	이 옵션은 명령행 처리기에서 x 또는 xx를 하나 이상의 명령문 종료 문자(1자 또는 2자)로 정의하고 사용하도록 지시합니다.	OFF
-v	이 옵션은 명령행 처리기에서 표준 출력으로 명령 텍스트를 반향하도록 지시합니다.	OFF
-w	이 옵션은 명령행 처리기에서 FETCH/SELECT 경고 메시지를 표시하도록 지시합니다.	ON
-x	이 옵션은 명령행 처리기에서 컬럼 이름을 포함하여 헤더 없이 데이터를 리턴하도록 지시합니다. 이 플래그는 일부 명령에만 영향을 줍니다. 이 플래그는 LIST TABLES와 같은 SQL문을 기반으로 하는 명령과 SQL문에 적용됩니다.	OFF
-z <i>filename</i>	이 옵션은 명령행 처리기에서 모든 출력 경로를 파일로 재지정하도록 지시합니다. 이 옵션은 -r 옵션과 유사하지만 출력에 메시지 또는 오류 코드가 포함됩니다.	OFF

예 :

다음 AIX 명령은

```
export DB2OPTIONS='+ta -c +ec -o -p'
```

세션에 대해 다음과 같은 디폴트 설정을 설정합니다.



```
Display SQLCA - off
Auto Commit - on
Display SQLCODE - off
Display Output - on
Display Prompt - on
```

다음은 이 옵션에 대한 자세한 설명입니다.

#### SQLCA 데이터 옵션 표시(-a):

DB2 명령 또는 SQL문을 실행한 후 SQLCA 데이터를 표준 출력에 표시합니다. SQLCA 데이터는 오류 또는 성공 메시지 대신 표시됩니다.

이 명령 옵션의 디폴트 설정은 OFF(+a 또는 -a-)입니다.

-o 및 -r 옵션은 -a 옵션에 영향을 줍니다. 자세한 내용은 옵션 설명을 참조하십시오.

#### 자동 커밋 옵션(-c):

이 옵션은 각 명령 또는 명령문이 독립적으로 처리되는지 여부를 지정합니다. ON(-c)으로 설정되면 각 명령 또는 명령문이 자동으로 커밋되거나 롤백됩니다. 명령 또는 명령문이 성공하면 해당 명령 또는 명령문과 이전에 자동 커밋 OFF(+c 또는 -c-)로 실행된 모든 성공 명령 및 명령문이 커밋됩니다. 그러나 명령 또는 명령문이 실패하면 해당 명령 또는 명령문과 이전에 자동 커밋 OFF로 실행된 모든 성공 명령 및 명령문이 롤백됩니다. OFF(+c 또는 -c-)가 설정된 경우 COMMIT 또는 ROLLBACK이 명시적으로 실행되거나, 자동 커밋 ON(-c)과 함께 다음 명령이 실행된 경우 이러한 조치 중 하나가 발생합니다.

이 명령 옵션의 디폴트 설정은 ON입니다.

자동 커밋 옵션은 다른 명령행 처리기 옵션에 영향을 주지 않습니다.

예: 다음 시나리오를 고려하십시오.

1. db2 create database test
2. db2 connect to test
3. db2 +c "create table a (c1 int)"
4. db2 select c2 from a

4단계의 SQL문은 테이블 A에 C2 컬럼이 없으므로 실패합니다. 해당 명령문은 자동 커밋 ON(디폴트값)으로 실행되었으므로 4단계의 명령문 뿐만 아니라 3단계의 명령문도 롤백합니다. 후자의 경우 자동 커밋 OFF로 실행되었기 때문입니다. 다음 명령은

```
db2 list tables
```

비어 있는 목록을 리턴합니다.

#### XML 선언 옵션(-d):

-d 옵션은 명령행 처리기에 XML 데이터의 XML 선언을 검색 및 표시하는지 여부를 알려줍니다.

ON(-d)으로 설정되면 XML 선언을 검색하고 표시합니다. OFF(+d 또는 -d-)로 설정되면 XML 선언이 검색 및 표시되지 않습니다. 이 명령 옵션의 디폴트 설정은 OFF입니다.

XML 선언 옵션은 다른 명령행 처리기 옵션에 영향을 주지 않습니다.

#### SQLCODE/SQLSTATE 옵션(-e) 표시:

-e{c|s} 옵션은 명령행 처리기에서 SQLCODE(-ec) 또는 SQLSTATE(-es)를 표준 출력에 표시하도록 지시합니다. -ec 및 -es 옵션은 CLP 대화식 모드에서 유효하지 않습니다.

이 명령 옵션의 디폴트 설정은 OFF(+e 또는 -e-)입니다.

-o 및 -r 옵션은 -e 옵션에 영향을 줍니다. 자세한 내용은 옵션 설명을 참조하십시오.

SQLCODE/SQLSTATE 표시 옵션은 다른 명령행 처리기 옵션에 영향을 주지 않습니다.

예: AIX에서 실행 중인 명령행 처리기에서 SQLCODE를 검색하려면 다음을 입력하십시오.

```
sqlcode=`db2 -ec +o db2-command`
```

#### 입력 파일에서 읽기 옵션(-f):

-f *filename* 옵션은 명령행 처리기에서 표준 입력이 아닌 지정된 파일의 입력을 읽도록 지시합니다. *Filename*은 파일의 디렉토리 경로를 포함할 수 있는 절대 또는 상대 파일 이름입니다. 디렉토리 경로가 지정되지 않으면 현재 디렉터리가 사용됩니다.

파일 입력 옵션으로 CLP가 호출되면 CLIENT APPLNAME 특수 레지스터를 자동으로 CLP *filename*으로 설정합니다.

다른 옵션과 -l 옵션이 결합되는 경우 -l 옵션을 마지막으로 지정해야 합니다. 예를 들면, 다음과 같습니다.

```
db2 -tvf filename
```

이 옵션은 대화식 모드에서 변경할 수 없습니다.

이 명령 옵션에 대한 디폴트 설정은 OFF (+f 또는 -f-)입니다.

명령은 QUIT 명령 또는 TERMINATE 명령이 실행되거나 EOF가 나타날 때까지 처리됩니다.

이 옵션과 데이터베이스 명령이 모두 지정되면 명령행 처리기가 명령을 처리하지 않고 오류 메시지가 리턴됩니다.

주석 문자 --로 시작되는 입력 파일 라인은 명령행 처리기에서 주석으로 간주됩니다. 주석 문자는 한 라인에서 첫 번째 비공백 문자여야 합니다.

(=로 시작되는 입력 파일 라인은 주석 블록 시작으로 간주됩니다. =)로 끝나는 라인은 주석 블록 끝을 표시합니다. (=로 시작되고 =)로 끝나는 입력 라인 블록은 명령행 처리기에서 연속 주석으로 간주됩니다. (= 앞과 =) 뒤의 공백은 허용됩니다. 주석은 중첩될 수 있으며 명령문에 중첩되어 사용될 수 있습니다. 명령 종료 문자(;는 =) 뒤에서 사용할 수 없습니다.

-f *filename* 옵션이 지정되면 -p 옵션은 무시됩니다.

입력 파일에서 읽기 옵션은 다른 명령행 처리기 옵션에 영향을 주지 않습니다.

#### 고급 인쇄 옵션(-i):

-i 옵션은 명령행 처리기에서 올바른 들여쓰기로 XML 데이터를 '고급 인쇄' 하도록 지시합니다. 이 옵션은 XQuery 명령문의 결과 세트에만 영향을 줍니다.

이 명령 옵션의 디폴트 설정은 OFF(+i 또는 -i-)입니다.

고급 인쇄 옵션은 다른 명령행 처리기 옵션에 영향을 주지 않습니다.

#### 실행기록 파일에 명령 로그 옵션(-l):

-l *filename* 옵션은 명령행 처리기에서 지정된 파일에 명령을 로깅하도록 지시합니다. 이 실행기록 파일에는 실행된 명령과 명령의 완료 상태에 대한 레코드가 들어 있습니다. *Filename*은 파일의 디렉토리 경로를 포함할 수 있는 절대 또는 상대 파일 이름입니다. 디렉토리 경로가 지정되지 않으면 현재 디렉토리가 사용됩니다. 지정된 파일 또는 디폴트 파일이 이미 있는 경우 새 로그 항목이 해당 파일에 추가됩니다.

다른 옵션과 -l 옵션이 결합되는 경우 -l 옵션을 마지막으로 지정해야 합니다. 예를 들면, 다음과 같습니다.

```
db2 -tv1 filename
```

이 명령 옵션의 디폴트 설정은 OFF(+i 또는 -i-)입니다.

실행기록 파일에 명령 로깅 옵션은 다른 명령행 처리기 옵션에 영향을 주지 않습니다.

#### 영향받은 행 수 표시 옵션(-m):

-m 옵션은 INSERT, 명령행 처리기에서 DELETE, UPDATE 또는 MERGE에 대해 영향을 받은 행 수를 인쇄할지 여부를 지시합니다.

ON (-m)으로 설정되면 영향을 받은 행 수가 INSERT/DELETE/UPDATE/MERGE 명령문에 대해 표시됩니다. OFF (+m or -m-)로 설정되면 영향을 받은 행 수가 표시되지 않습니다. 다른 명령문에서는 이 옵션이 무시됩니다. 이 명령 옵션의 디폴트 설정은 OFF입니다.

-o 및 -r 옵션은 -m 옵션에 영향을 줍니다. 자세한 내용은 옵션 설명을 참조하십시오.

#### 개행 문자 제거 옵션(-n):

단일 분리 토큰에서 개행 문자를 제거합니다. 이 옵션이 지정되지 않으면 개행 문자가 공백으로 바뀝니다. 이 옵션은 대화식 모드에서 변경할 수 없습니다.

이 명령 옵션의 디폴트 설정은 OFF(+n 또는 -n-)입니다.

이 옵션은 -t 옵션과 함께 사용해야 합니다. 자세한 내용은 옵션 설명을 참조하십시오.

#### 출력 표시 옵션(-o):

-o 옵션은 명령행 처리기에서 출력 데이터와 메시지를 표준 출력으로 보내도록 지시합니다.

이 명령 옵션의 디폴트 설정은 ON입니다.

대화식 모드 시작 정보는 이 옵션의 영향을 받지 않습니다. 출력 데이터는 사용자 지정 명령을 실행한 보고서 출력과 SQLCA 데이터(요청된 경우)로 구성됩니다.

다음 옵션은 +o 옵션의 영향을 받습니다.

- -r *filename*: 대화식 모드 시작 정보가 저장되지 않습니다.
- -e: +o가 지정되더라도 SQLCODE 또는 SQLSTATE가 표준 출력에 표시됩니다.
- -a: +o가 지정된 경우에는 영향을 주지 않습니다. -a, +o 및 -r*filename*이 지정되면 SQLCA 정보가 파일에 기록됩니다.

-o 및 -e 옵션이 모두 지정되면 데이터와 SQLCODE 또는 SQLSTATE가 화면에 표시됩니다.

-o 및 -v 옵션이 모두 지정되면 데이터가 표시되고 실행된 각 명령의 텍스트가 화면에 반향됩니다.

출력 표시 옵션은 다른 명령행 처리기 옵션에 영향을 주지 않습니다.

#### DB2 대화식 프롬프트 표시 옵션(-p):

-p 옵션은 사용자가 대화식 모드인 경우 명령행 처리기에서 명령행 처리기 프롬프트를 표시하도록 지시합니다.

이 명령 옵션의 디폴트 설정은 ON입니다.

프롬프트 해제는 명령이 명령행 처리기로 파이핑될 때 유용합니다. 예를 들어, CLP 명령이 포함된 파일은 다음을 실행하여 실행할 수 있습니다.

```
db2 +p < myfile.clp
```

-p 옵션은 -f *filename* 옵션이 지정된 경우 무시됩니다.

DB2 대화식 프롬프트 표시 옵션은 다른 명령행 처리기 옵션에 영향을 주지 않습니다.

**공백 및 줄 바꾸기 보존 옵션(-q):**

-q 옵션은 명령행 처리기에서 문자열의 공백과 줄 바꾸기를 작은 따옴표 또는 큰따옴표로 구분하여 보존하도록 지시합니다.

이 명령 옵션의 디폴트 설정은 OFF(+q 또는 -q-)입니다.

-q 옵션이 ON이면 -n 옵션은 무시됩니다.

**보고서 파일에 저장 옵션(-r):**

-r *filename* 옵션은 명령으로 생성된 모든 출력 데이터가 지정된 파일에 작성되도록 지시하므로 화면에 나타나지 않는 보고서를 캡처하는 데 유용합니다. 메시지 또는 오류 코드는 파일에 작성되지 않습니다. *Filename*은 파일의 디렉토리 경로를 포함할 수 있는 절대 또는 상대 파일 이름입니다. 디렉토리 경로가 지정되지 않으면 현재 디렉토리가 사용됩니다. 새 보고서 항목은 파일에 추가됩니다.

이 명령 옵션의 디폴트 설정은 OFF(+r 또는 -r-)입니다.

-a 옵션이 지정되면 SQLCA 데이터가 파일에 기록됩니다.

-r 옵션은 -e 옵션에 영향을 주지 않습니다. -e 옵션이 지정되면 SQLCODE 또는 SQLSTATE가 파일이 아닌 표준 출력에 기록됩니다.

-r *filename*이 DB2OPTIONS에 설정되는 경우 사용자는 명령행에서 +r(또는 -r-) 옵션을 설정하여 특정 명령 호출에 대한 출력 데이터가 파일에 작성되지 않도록 방지할 수 있습니다.

보고서 파일에 저장 옵션은 다른 명령행 처리기 옵션에 영향을 주지 않습니다.

**명령 오류 시 실행 중지 옵션(-s):**

명령이 대화식 모드로 또는 출력 파일에서 실행되고 구문 또는 명령 오류가 발생하는 경우 -s 옵션은 명령행 처리기에서 실행을 중지하고 표준 출력에 오류 메시지를 작성하도록 지시합니다.

이 명령 옵션의 디폴트 설정은 OFF(+s 또는 -s-)입니다. 이 설정은 명령행 처리기에서 오류 메시지를 표시하고 나머지 명령을 계속 실행하며 시스템 오류가 발생하는 경우에만(리턴 코드 8) 실행을 중지하도록 지시합니다.

다음 표에는 이러한 동작이 요약 설명되어 있습니다.

표 4. CLP 리턴 코드 및 명령 실행

리턴 코드	-s 옵션 설정	+s 옵션 설정
0(성공)	계속 실행	계속 실행
1(0개 행 선택)	계속 실행	계속 실행
2(경고)	계속 실행	계속 실행

표 4. CLP 리턴 코드 및 명령 실행 (계속)

리턴 코드	-s 옵션 설정	+s 옵션 설정
4(DB2 또는 SQL 오류)	실행 중지	계속 실행
8(시스템 오류)	실행 중지	실행 중지

**명령문 종료 문자 옵션(-t 및 -tdx 또는 -tdxx):**

-t 옵션은 명령행 처리기에서 세미콜론(;)을 명령문 종료 문자로 사용하도록 지시하며 백슬래시(\) 라인 연속 문자를 사용하지 않습니다. 이 옵션은 대화식 모드에서 변경할 수 없습니다.

이 명령 옵션의 디폴트 설정은 OFF(+t 또는 -t-)입니다.

주: CLP를 사용하여 CLP문을 실행하는 경우 세미콜론이 아닌 종료 문자를 선택하는 것이 가장 좋습니다. 이렇게 하면 네임 스페이스 선언을 사용하는 명령문 또는 쿼리가 잘못 해석되지 않습니다. 네임 스페이스 선언도 세미콜론으로 종료되기 때문입니다.

종료 문자 길이를 1자 또는 2자로 정의하려면 -td와 선택한 하나 이상의 문자를 차례로 사용하십시오. 예를 들어, -td%%는 %%를 명령문 종료 문자로 설정합니다. 또는 입력 파일에서 --#SET TERMINATOR 지시문을 사용하여 명령문 종료 문자를 설정하십시오. 예를 들어, 다음과 같습니다.

```
db2 -td%% -f file1.txt
```

또는

```
db2 -f file2.txt
```

여기서 file2.txt에는 파일의 첫 번째 명령문으로 다음이 포함됩니다.

```
--#SET TERMINATOR %%
```

이 명령 옵션의 디폴트 설정은 OFF입니다.

하나 이상의 종료 문자를 사용하여 명령행에서 여러 명령문을 병합할 수 없습니다. 종료 기호 점은 각 입력 라인의 마지막 하나 또는 두 개의 비공백 문자에 대해서만 수행되기 때문입니다.

명령문 종료 문자는 다른 명령행 처리기 옵션에 영향을 주지 않습니다.

**상세 출력 옵션(-v):**

-v 옵션은 명령행 처리기에서 출력을 표시하기 전에 사용자가 입력한 명령 텍스트와 해당 명령의 메시지를 표준 출력에 반향하도록 지시합니다. ECHO는 이 옵션에서 제외됩니다.

이 명령 옵션의 디폴트 설정은 OFF(+v 또는 -v-)입니다.

+o(또는 -o-)가 지정되면 -v 옵션이 영향을 주지 않습니다.

상세 출력 옵션은 다른 명령행 처리기 옵션에 영향을 주지 않습니다.

#### 경고 메시지 표시 옵션(-w):

-w 옵션은 명령행 처리기에서 쿼리 중에 발생할 수 있는 경고 메시지를 표시할지 여부를 지시합니다(FETCH/SELECT). 경고는 쿼리 실행의 다양한 단계에서 발생할 수 있으며 결과적으로 데이터 리턴 이전 또는 이후나 리턴 중에 메시지가 표시됩니다. 리턴된 데이터에 경고 메시지 텍스트가 포함되지 않도록 하려면 이 플래그를 사용할 수 있습니다.

이 명령 옵션의 디폴트 설정은 ON입니다.

#### 컬럼 표제 인쇄 제외 옵션(-x):

-x 옵션은 명령행 처리기에서 컬럼 이름을 포함하여 헤더 없이 데이터를 리턴하도록 지시합니다. 이 플래그는 일부 명령에만 영향을 줍니다. 이 플래그는 LIST TABLES와 같은 SQL문을 기반으로 하는 명령과 SQL문에 적용됩니다.

이 명령 옵션의 디폴트 설정은 OFF입니다.

#### 파일에 모든 출력 저장 옵션(-z):

-z *filename* 옵션은 명령으로 생성된 모든 출력이 지정된 파일에 작성되도록 지시하므로 화면에 나타나지 않는 보고서를 캡처하는 데 유용합니다. 이 옵션은 -r 옵션과 유사하지만 이러한 경우 메시지, 오류 코드 및 기타 정보 출력 또한 파일에 작성됩니다. *Filename*은 파일의 디렉토리 경로를 포함할 수 있는 절대 또는 상대 파일 이름입니다. 디렉토리 경로가 지정되지 않으면 현재 디렉토리가 사용됩니다. 새 보고서 항목은 파일에 추가됩니다.

이 명령 옵션의 디폴트 설정은 OFF(+z 또는 -z-)입니다.

-a 옵션이 지정되면 SQLCA 데이터가 파일에 기록됩니다.

-z 옵션은 -e 옵션에 영향을 주지 않습니다. -e 옵션이 지정되면 SQLCODE 또는 SQLSTATE가 파일이 아닌 표준 출력에 기록됩니다.

-z *filename*이 DB2OPTIONS에 설정되는 경우 사용자는 명령행에서 +z(또는 -z-) 옵션을 설정하여 특정 명령 호출에 대한 출력 데이터가 파일에 작성되지 않도록 방지할 수 있습니다.

파일에 모든 출력 저장 옵션은 다른 명령행 처리기 옵션에 영향을 주지 않습니다.





---

## 제 4 장 명령행 처리기 리턴 코드

명령행 처리기가 명령 또는 SQL문 처리를 완료하면 리턴(또는 종료) 코드를 리턴합니다. 이러한 코드는 명령행에서 CLP 함수를 실행하는 사용자에게는 투명하지만 해당 함수가 셸 스크립트에서 실행될 때 검색할 수 있습니다.

예를 들어, 다음 본 셸 스크립트는 GET DATABASE MANAGER CONFIGURATION 명령을 실행한 다음 CLP 리턴 코드를 검사합니다.

```
db2 get database manager configuration
if [ "$?" = "0" ]
then echo "OK!"
fi
```

리턴 코드는 다음 중 하나입니다.

### 코드 설명

- 0 DB2 명령 또는 SQL문이 성공적으로 실행됨
- 1 SELECT 또는 FETCH문이 행을 리턴하지 않음
- 2 DB2 명령 또는 SQL문 경고
- 4 DB2 명령 또는 SQL문 오류
- 8 명령행 처리기 시스템 오류

명령행 처리기는 사용자가 대화식 모드에서 명령문을 실행하거나 -f 옵션을 사용하여 파일에서 입력을 읽는 동안 리턴 코드를 제공하지 않습니다.

리턴 코드는 사용자가 대화식 모드를 종료한 후 또는 입력 파일 처리가 종료되어야 사용할 수 있습니다. 이러한 경우 리턴 코드는 해당 지점에서 실행된 개별 명령 또는 명령문에서 리턴된 개별 코드의 논리적 OR입니다.

예를 들어, 대화식 모드의 사용자가 명령을 실행하여 리턴 코드가 0, 1 및 2인 경우 사용자가 대화식 모드를 종료한 후 리턴 코드 3이 리턴됩니다. 개별 코드 0, 1 및 2는 리턴되지 않습니다. 리턴 코드 3은 사용자에게 대화식 모드 처리 중에 하나 이상의 명령이 1을 리턴하고 하나 이상의 명령이 2를 리턴하도록 지시합니다.

리턴 코드 4는 DB2 명령 또는 SQL문이 음수 SQLCODE를 리턴한 경우에 리턴됩니다. 리턴 코드 8은 명령행 처리기에 시스템 오류가 있는 경우에만 리턴됩니다.

입력 파일 또는 대화식 모드에서 명령이 실행되고 명령행 처리기에 시스템 오류가 발생하는 경우(리턴 코드 8) 명령 실행이 즉시 중지됩니다. 하나 이상의 DB2 명령 또는 SQL문이 오류로 종료되는 경우(리턴 코드 4) -s(명령 실행 중지 오류) 옵션이 설정되면 명령 실행이 중지되고 그렇지 않으면 계속 실행됩니다.



---

## 제 5 장 명령행 처리기에서 명령 도움말 호출

명령 도움말은 명령행 처리기에서 명령 구문을 설명합니다.

명령 도움말을 호출하려면 명령행 처리기를 열고 다음을 입력하십시오.

```
? command
```

여기서 *command*는 키워드 또는 전체 명령을 나타냅니다.

예를 들어, ? catalog는 모든 CATALOG 명령의 도움말을 표시하고 ? catalog database는 CATALOG DATABASE 명령의 도움말만 표시합니다.



---

## 제 6 장 명령행 처리기에서 메시지 도움말 호출

메시지 도움말은 메시지의 원인과 오류에 대한 응답으로 수행해야 하는 조치를 설명합니다.

메시지 도움말을 호출하려면 명령행 처리기를 열고 다음을 입력하십시오.

? *XXXnnnnn*

여기서 *XXXnnnnn*은 유효한 메시지 ID를 나타냅니다.

예를 들어, ? SQL30081은 SQL30081 메시지에 대한 도움말을 표시합니다.



## 제 2 부 명령행 SQL문 및 XQuery 명령문 사용

이 섹션에서는 명령행에서 구조적 쿼리 언어(SQL)문 사용에 대한 정보를 제공합니다. 이들 명령문은 운영 체제 명령 프롬프트에서 직접 실행할 수 있으며, 명령이 응용프로그램에 기록된 것과 동일한 방식으로 데이터베이스 테이블, 인덱스 또는 뷰에 저장된 정보를 정의하고 조작하는 데 사용할 수 있습니다. 정보를 추가, 삭제 또는 갱신할 수 있으며 테이블의 콘텐츠에서 보고서를 생성할 수 있습니다.

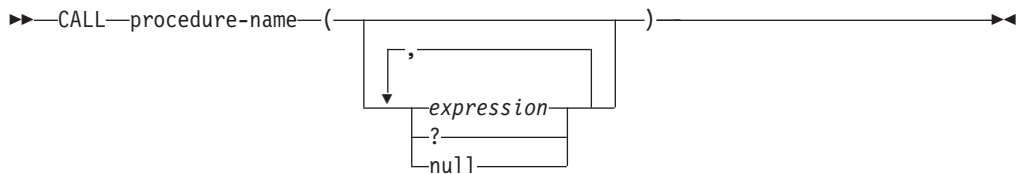
명령행에서 SQL문을 사용할 수 있으며 스토어드 프로시저(SYSPROC.ADMIN\_CMD())를 사용하여 SQL을 통해 일부 CLP 명령을 실행할 수 있습니다. 이 스토어드 프로시저 사용 방법에 대한 자세한 정보는 SQL Administrative Routines를 참조하십시오.

CLP에서 XQuery 명령문을 발행하려면 명령문 앞에 XQUERY 키워드를 붙이십시오.

**주:** CLP를 사용하여 XQuery 명령문을 발행하는 경우 세미콜론 이외의 종료 문자(-t 옵션)를 선택하는 것이 가장 좋습니다. 이것은 이름 스페이스 선언도 세미콜론으로 종료되므로 이름 스페이스 선언을 사용하는 명령문 또는 쿼리가 잘못 해석되지 않도록 합니다.

명령행 처리기를 통해 실행될 수 있는 모든 SQL문은 33 페이지의 표 5의 CLP 컬럼에 나열됩니다. 모든 SQL문의 구문은 명령행에서 실행되거나 소스 프로그램에 임베디드(embedded)되는지 여부와 상관없이 SQL 참조서에서 설명합니다. 많은 Embedded SQL 문 및 CLP SQL문의 구문은 동일합니다. 그러나 호스트 변수, 매개변수 표시문자, 디스크립터 이름 및 명령문 이름은 Embedded SQL에만 적용할 수 있습니다. CALL, CLOSE, CONNECT, DECLARE CURSOR, FETCH 및 OPEN의 구문은 이들 명령문이 CLP를 통해 임베디드(embedded) 또는 실행되는지 여부에 따라 다릅니다. 이들 명령문의 CLP 구문은 아래에서 제공합니다.

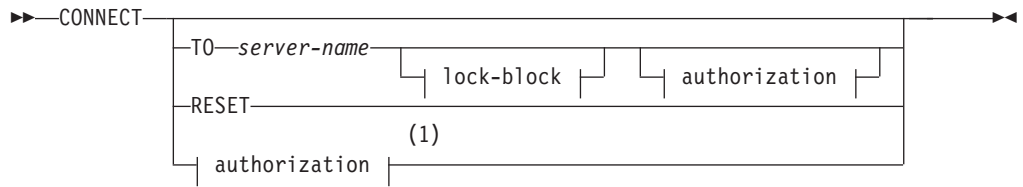
### CALL



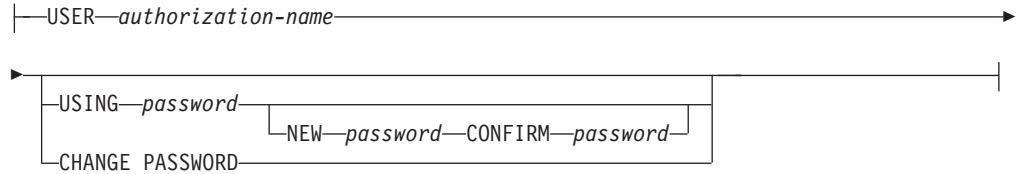
### CLOSE



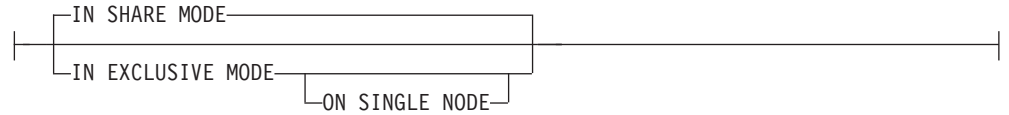
### CONNECT



**authorization:**



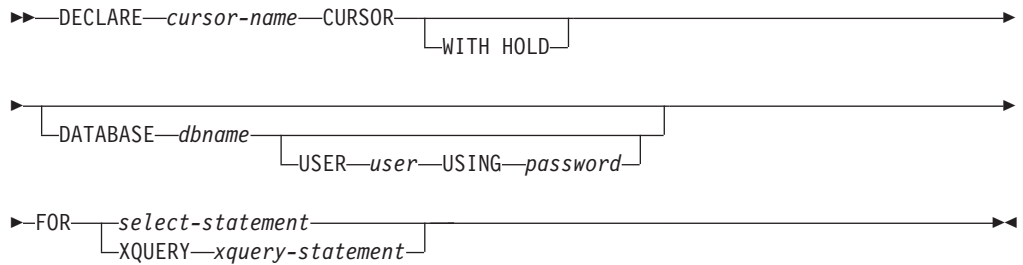
**lock-block:**



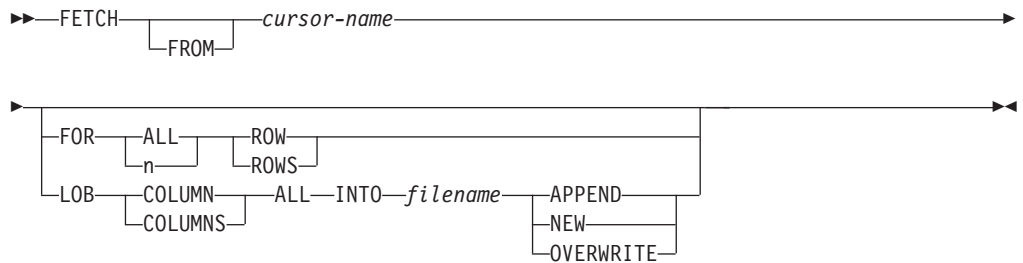
**주:**

1 이 형식은 내재적 연결이 설정된 경우에만 유효합니다.

**DECLARE CURSOR**



**FETCH**



**OPEN**



주:

1. CALL이 발행될 때,

- 프로시저의 각 IN 또는 INOUT 매개변수에 대한 표현식을 사용해야 합니다. INOUT 매개변수의 경우 표현식은 단일 리터럴 값이어야 합니다. INOUT XML 매개변수는 NULL(널(NULL) 입력 가능한 경우)이거나 XMLPARSE(DOCUMENT *string*) 형식이어야 합니다. XMLPARSE에 대한 인수에 있는 *string*은 문자열 리터럴이어야 하며 CURRENT IMPLICIT XMLPARSE OPTION 특수 레지스터에 종속됩니다. 표현식일 수 없습니다.
- 물음표(?)가 프로시저의 각 OUT 매개변수에 사용되어야 합니다.
- 스토어드 프로시저는 카탈로그되어야 합니다. 카탈로그 해제된 프로시저가 호출되면 SQL0440N 오류 메시지가 리턴됩니다.

다음 CLP 스크립트는 XML 컬럼 C1을 갖는 테이블을 작성한 후 PROC4라는 프로시저를 작성합니다. 세 개의 XML 매개변수 IN(PARM1), INOUT(PARM2) 및 OUT(PARM3)을 사용하며, XML 데이터를 갖는 결과 세트를 리턴합니다.

```
CREATE TABLE TAB4(C1 XML)
CREATE PROCEDURE PROC4(IN PARM1 XML, INOUT PARM2 XML, OUT PARM3 XML)
  LANGUAGE SQL
  BEGIN
    DECLARE STMT CLOB(1M) DEFAULT '';
    DECLARE C1 CURSOR WITH RETURN FOR S1;
    SET STMT = 'SELECT C1 FROM TAB4';

    /* INSERT PARM1 */
    INSERT INTO TAB4 VALUES(PARM1);

    /* MANIPULATE PARM2 */

    /* SET PARM3 AND INSERT */
    SET PARM3 = XMLPARSE(DOCUMENT '<a>333</a>');
    INSERT INTO TAB4 VALUES(PARM3);

    /* RETURN A RESULT SET WITH XML DATA */
    PREPARE S1 FROM STMT;
    OPEN C1;
  END
```

명령행 처리기에서 프로시저 PROC4를 호출하려면 CALL문을 발행하십시오.

```
CALL PROC4(XMLPARSE(DOCUMENT '<a>111</a>'), XMLPARSE(DOCUMENT '<a>222</a>'), ?)
```

2. CONNECT의 CLP 버전은 사용자가 다음 매개변수를 사용하여 암호를 변경할 수 있도록 허용합니다.

### **NEW password**

사용자 이름에 지정될 새 암호를 지정합니다. 암호 길이는 최대 18자일 수 있습니다. 암호를 변경할 시스템은 사용자 인증이 설정된 방식에 따라 다릅니다.

### **CONFIRM password**

새 암호와 동일해야 하는 문자열입니다. 이 매개변수는 입력 오류를 포착하는 데 사용됩니다.

### **CHANGE PASSWORD**

이 옵션이 지정되면 사용자에게 현재 암호, 새 암호 및 새 암호의 확인을 위해 프롬프트됩니다. 암호는 입력 시에 표시되지 않습니다.

3. DECLARE CURSOR문의 DATABASE절은 커서가 커서 조작에서 후속 로드에서 사용되고 있을 때만 적용할 수 있습니다.
4. XQuery 명령문에서 DECLARE CURSOR문을 사용하려면 사용자는 XQuery 명령문 앞에 키워드 XQUERY를 명시적으로 추가해야 합니다.
5. FETCH가 명령행 처리기를 통해 발행될 때, 지역의 10진수 분리문자와 함께 10진수 및 부동 소수점 숫자가 표시되며, 미국, 캐나다 및 영국에서는 마침표(.), 대부분의 다른 국가/지역에서는 쉼표(,)입니다. 그러나 테이블을 갱신하기 위해 INSERT, UPDATE, CALL 및 기타 SQL문이 명령행 처리기를 통해 발행될 때는 동일한 목적으로 쉼표를 사용하는 국가/지역에서도 마침표를 10진수 분리문자로 사용해야 합니다.
6. FETCH가 명령행 처리기를 통해 발행될 때, 널(NULL) 값은 일반적으로 하이픈(-)으로 표시됩니다. DFT\_SQLMATHWARN YES로 구성된 데이터베이스의 경우, 산술 오류를 생성하는 표현식은 널(NULL) 값으로 처리됩니다. 그런 산술 오류 NULL 값은 더하기(+)로 표시됩니다.

예를 들어 다음과 같이 테이블 t1을 작성하고 채우십시오.

```
create table t1 (i1 int , i2 int);
insert into t1 values (1,1),(2,0),(3,null);
```

명령문 select i1/i2 from t1은 다음 결과를 생성합니다.

```
1
---
1
+
-
3개의 레코드가 선택되었습니다.
```

7. 새 LOB 옵션이 FETCH에 추가되었습니다. LOB절이 지정되는 경우 새 행만 폐치됩니다.
  - SELECT가 LOB 컬럼이 들어있는 테이블을 쿼리하기 위해 명령행 처리기를 통해 발행될 때, 모든 컬럼은 출력에서 8KB로 절단됩니다.

- 각 LOB 컬럼 값은 이름 *filename.xxx*를 갖는 파일에 폐치되며, 여기서 *filename*은 LOBLOB절에서 지정되고 *xxx*는 001부터 999까지의 파일 확장자입니다(001은 대응하는 DECLARE CURSOR문의 선택 목록에 있는 첫 번째 LOB 컬럼이고 002는 두 번째 LOB 컬럼이며 999는 999번째 컬럼임). 파일에 폐치할 수 있는 LOB 컬럼의 최대 수는 999입니다.
  - 데이터가 들어있는 파일의 이름이 LOB 컬럼에 표시됩니다.
8. 명령행 처리기는 16진수 표시로 BLOB 컬럼을 표시합니다.
  9. 적합한 변환 기능을 사용할 수 없는 경우 구조화된 유형 컬럼에 대한 참조를 포함하는 SQL문을 발행할 수 없습니다.
  10. SQL문 구성요소가 들어있는 SQL문 및 CLP 명령에 대한 64K의 CLP 부과 제한은 현재 제거되었습니다.
  11. SELECT, CALL 또는 XQuery를 통해 검색되는 XML 데이터는 출력에서 4000 바이트로 절단됩니다.

CLP가(CLP를 통해 SQL문을 사용하여 데이터베이스를 쿼리할 때) 데이터를 표시하는 방법을 변경하려면 쿼리되는 데이터베이스에 대해 CLP 바인드 파일을 리바인드하십시오. 예를 들어 ISO 형식으로 날짜 및 시간을 표시하려면 다음을 수행하십시오.

1. CLP 바인드 파일의 이름이 들어있는 텍스트 파일을 작성하십시오. 이 파일은 하나의 BIND 명령으로 다중 파일을 바인드하기 위한 목록 파일로 사용됩니다. 이 예에서 파일 이름은 *clp.lst*이며 콘텐츠는 다음과 같습니다.

```
db2clpcs.bnd +
db2clpr.r.bnd +
db2clpur.bnd +
db2clprs.bnd +
db2clpns.bnd
```

2. 데이터베이스에 연결하십시오.
3. 다음 명령을 발행하십시오.

```
db2 bind @clp.lst collection nullid datetime iso
```

표 5. SQL문(DB2)

SQL문	동적 <sup>1</sup>	명령행 처리기 (CLP)	콜 레벨 인터페이스 <sup>3</sup> (CLI)	SQL 프로시저
ALLOCATE CURSOR				X
지정 명령문				X
ASSOCIATE LOCATORS				X
ALTER { BUFFERPOOL, NICKNAME, <sup>9</sup> NODEGROUP, SERVER, <sup>9</sup> TABLE, TABLESPACE, USER MAPPING, <sup>9</sup> TYPE, VIEW }	X	X	X	
BEGIN DECLARE SECTION <sup>2</sup>				
CALL	X	X	X	X

표 5. SQL문(DB2) (계속)

SQL문	동적 <sup>1</sup>	명령행 처리기 (CLP) 콜 레벨 인터페이스 <sup>3</sup> (CLI)		SQL 프로시저
CASE문				X
CLOSE		X	SQLCloseCursor(), SQLFreeStmt()	X
COMMENT ON	X	X	X	X
COMMIT	X	X	SQLEndTran(), SQLTransact()	X
Compound SQL(임베디드)			X <sup>4</sup>	
복합 명령문				X
CONNECT(유형 1)		X	SQLBrowseConnect(), SQLConnect(), SQLDriverConnect()	
CONNECT(유형 2)		X	SQLBrowseConnect(), SQLConnect(), SQLDriverConnect()	
CREATE { ALIAS, BUFFERPOOL, DISTINCT TYPE, EVENT MONITOR, FUNCTION, FUNCTION MAPPING <sup>9</sup> , GLOBAL TEMPORARY TABLE, INDEX, INDEX EXTENSION, METHOD, NICKNAME, <sup>9</sup> NODEGROUP, PROCEDURE, SCHEMA, SERVER, TABLE, TABLESPACE, TRANSFORM, TYPE MAPPING, <sup>9</sup> TRIGGER, USER MAPPING, <sup>9</sup> TYPE, VIEW, WRAPPER <sup>9</sup> }	X	X	X	X <sup>10</sup>
DECLARE CURSOR <sup>2</sup>		X	SQLAllocStmt()	X
DECLARE GLOBAL TEMPORARY TABLE	X	X	X	X
DELETE	X	X	X	X
DESCRIBE <sup>8</sup>		X	SQLColAttributes(), SQLDescribeCol(), SQLDescribeParam() <sup>6</sup>	
DISCONNECT		X	SQLDisconnect()	
DROP	X	X	X	X <sup>10</sup>
END DECLARE SECTION <sup>2</sup>				
EXECUTE			SQLExecute()	X
EXECUTE IMMEDIATE			SQLExecDirect()	X
EXPLAIN	X	X	X	X
FETCH		X	SQLExtendedFetch(), SQLFetch(), SQLFetchScroll()	X
FLUSH EVENT MONITOR	X	X	X	
FOR문				X
FREE LOCATOR			X <sup>4</sup>	X
GET DIAGNOSTICS				X
GOTO문				X

표 5. SQL문(DB2) (계속)

SQL문	동적 <sup>1</sup>	명령행 처리기 (CLP)      콜 레벨 인터페이스 <sup>3</sup> (CLI)		SQL 프로시저
GRANT	X	X	X	X
IF문				X
INCLUDE <sup>2</sup>				
INSERT	X	X	X	X
ITERATE				X
LEAVE문				X
LOCK TABLE	X	X	X	X
LOOP문				X
OPEN		X	SQLExecute(), SQLExecDirect()	X
PREPARE			SQLPrepare()	X
REFRESH TABLE	X	X	X	
RELEASE		X		X
RELEASE SAVEPOINT	X	X	X	X
RENAME TABLE	X	X	X	
RENAME TABLESPACE	X	X	X	
REPEAT문				X
RESIGNAL문				X
RETURN문				X
REVOKE	X	X	X	
ROLLBACK	X	X	SQLEndTran(), SQLTransact()	X
SAVEPOINT	X	X	X	X
SELECT	X	X	X	X
SELECT INTO				X
SET CONNECTION		X	SQLSetConnection()	
SET CURRENT DEFAULT TRANSFORM GROUP	X	X	X	X
SET CURRENT DEGREE	X	X	X	X
SET CURRENT EXPLAIN MODE	X	X	X, SQLSetConnectAttr()	X
SET CURRENT EXPLAIN SNAPSHOT	X	X	X, SQLSetConnectAttr()	X
SET CURRENT PACKAGESET				
SET CURRENT QUERY OPTIMIZATION	X	X	X	X
SET CURRENT REFRESH AGE	X	X	X	X
SET EVENT MONITOR STATE	X	X	X	X
SET INTEGRITY	X	X	X	
SET PASSTHRU <sup>9</sup>	X	X	X	X
SET PATH	X	X	X	X
SET SCHEMA	X	X	X	X
SET SERVER OPTION <sup>9</sup>	X	X	X	X

표 5. SQL문(DB2) (계속)

SQL문	동적 <sup>1</sup>	명령행 처리기		SQL 프로시저
		(CLP)	콜 레벨 인터페이스 <sup>3</sup> (CLI)	
SET transition-variable <sup>5</sup>	X	X	X	X
SIGNAL문				X
SIGNAL SQLSTATE <sup>5</sup>	X	X	X	
UPDATE	X	X	X	X
VALUES INTO				X
WHENEVER <sup>2</sup>				
WHILE문				X

주:

1. 이 목록의 모든 명령문을 정적 SQL로 코드화할 수 있지만 X로 표시된 것만 동적 SQL로 코드화할 수 있습니다.
2. 이 명령문을 실행할 수 없습니다.
3. X는 SQLExecDirect() 또는 SQLPrepare() 및 SQLExecute()를 사용하여 이 명령문을 실행할 수 있음을 표시합니다. 동등한 DB2 CLI 함수가 있는 경우 함수 이름이 나열됩니다.
4. 이 명령문이 동적은 아니지만, DB2 CLI를 사용하여 SQLExecDirect() 또는 SQLPrepare() 중 하나와 SQLExecute()를 호출할 때 이 명령문을 지정할 수 있습니다.
5. CREATE TRIGGER문에서만 사용할 수 있습니다.
6. SQL DESCRIBE문만을 사용하여 출력을 설명할 수 있는 반면, DB2 CLI를 사용하여 입력을 설명할 수도 있습니다 (SQLDescribeParam() 함수를 사용하여).
7. SQL FETCH문을 사용하면 한 방향으로 한 번에 하나의 행을 페치할 수 있는 반면, DB2 CLI SQLExtendedFetch() 및 SQLFetchScroll() 함수를 사용하면 배열로 페치할 수 있습니다. 또한, 모든 방향으로, 결과 세트의 어느 위치에나 페치할 수 있습니다.
8. DESCRIBE SQL문은 CLP DESCRIBE 명령의 구문과 다른 구문을 갖습니다.
9. 명령문은 페더레이티드 데이터베이스 서버에 대해서만 지원됩니다.
10. SQL 프로시저는 인덱스, 테이블 및 뷰에 대해 CREATE 및 DROP문만 발행할 수 있습니다.

---

## 제 3 부 명령 구문 도움말을 읽는 방법

특정 명령에 유효한 옵션을 기억하지 못하는 경우가 있습니다. 모든 명령행 처리기(CLP) 명령은 명령 키워드 앞에 물음표(?)를 추가하여 CLP 프롬프트에서 도움말 화면을 호출할 수 있습니다. 많은 시스템 명령의 경우 명령 키워드 다음에 *help* 옵션을 실행하여 요약 도움말 화면을 표시할 수 있습니다. 명령 창에 나타나는 유용한 명령 도움말 화면 출력은 여기에서 설명하는 구문 규칙을 사용합니다.

### 도움말 호출

#### CLP 명령

CLP 명령 도움말 화면을 표시하려면 BACKUP DATABASE 명령에 대한 아래 예와 같이 db2 대화식 모드 프롬프트에서(db2 =>) 명령 키워드 앞에 물음표를 추가하십시오.

```
db2 => ? backup database
```

또는 'db2' 대화식 모드가 아닌 경우 BACKUP DATABASE 명령에 대한 아래 예와 같이 각 명령 도움말 화면 호출 앞에 db2를 추가하십시오.

```
=> db2 ? backup database
```

#### 시스템 명령어

대부분의 시스템 명령은 시스템 명령 키워드 다음에 *help* 옵션을 추가하여 명령 도움말 화면을 표시할 수 있습니다. 많은 시스템 명령이 일반 *help* 옵션을 사용하지만 다른 및/또는 추가 *help* 옵션을 사용하는 시스템 명령도 있습니다. 최초 시도하는 경우 명령의 이전 *help* 옵션을 검색하지 않고 명령 도움말 화면을 호출할 수 있는 다음 일반 옵션을 시도하십시오.

#### 도움말 옵션

- -h
- -?
- -help
- 명령 키워드 다음에 아무 것도 입력하지 않습니다.

주: 명령 키워드 다음에 아무 것도 입력하지 않으면 옵션이 필요하지 않은 경우 실제로 명령을 실행할 수 있습니다.

### 도움말 화면 구문 규칙

[ ] 선택적 매개변수를 묶음

{ } 필수 매개변수를 묶음

- | 두 개 이상의 항목을 분리, 하나를 선택할 수 있음
- ... 반복할 수 있는 매개변수 표시
- ( ) 반복할 수 있는 매개변수 분리문자(항상 사용되지는 않음)

KEYWORDS 명령은 대문자로 표시함

적절한 입력을 결정하고 입력하는 데 필요한 변수는 소문자로 표시함

### 명령 도움말 화면 출력 예

다음은 UPDATE MONITOR SWITCHES 명령의 CLP 명령 도움말 화면입니다.

```
db2 => ? update monitor
UPDATE MONITOR SWITCHES USING {switch-name {ON | OFF} ...}
[AT DBPARTITIONNUM db-partition-number | GLOBAL]
```

```
switch-name:
BUFFERPOOL, LOCK, SORT, STATEMENT, TABLE, TIMESTAMP, UOW
```

다음은 db2look 명령의 시스템 명령 도움말 화면입니다. 이 명령은 지정된 -h 도움말 옵션으로 호출되지 않았습니다.

C:\Program Files\IBM\SQLLIB\BIN>db2look

```
Syntax: db2look -d DBname [-e] [-xs] [-xdir Path] [-u Creator] [-z Schema]
[-t Tname1 Tname2...TnameN] [-tw Tname] [-h]
[-o Fname] [-a] [-m] [-c] [-r] [-l] [-x] [-xd] [-f]
[-fd] [-td x] [-noview] [-i userID] [-w password]
[-v Vname1 Vname2 ... VnameN] [-dp] [-ct]
[-wrapper WrapperName] [-server ServerName] [-nofed]
[-wlm] [-ap]

[-wrapper WrapperName] [-server ServerName] [-fedonly] [-nofed]
```

db2look [-h]

- d: 데이터베이스 이름: 반드시 지정해야 합니다.
- e: 데이터베이스 중복을 위해 필요한 DDL 파일 추출
- xs: XSR 오브젝트 익스포트 및 DDL문이 포함된 스크립트 생성
- xdir: 경로 이름: XSR 오브젝트가 저장될 디렉토리
- u: 작성자 ID: -u와 -a 모두 지정되지 않은 경우, \$USER가 사용됩니다.
- z: 스키마 이름: -z 및 -a가 모두 지정되면, -z는 무시됩니다.
- t: 지정된 테이블에 대해 통계 생성
- tw: 이름이 테이블 이름의 패턴 기준(와일드카드 문자)과 일치하는 테이블의 DDL 생성
- ap: AUDIT USING문 생성
- wlm: WLM 특정 DDL문 생성
- h: 추가 세부사항 도움말 메시지
- o: 제공된 파일 이름으로 출력 경로 재지정
- a: 모든 작성자에 대한 통계 생성
- m: 가상 갱신 모드에서 db2look 유틸리티 실행
  - c: 가상 갱신을 위해 COMMIT문 생성 안함
  - r: 가상 갱신을 위해 RUNSTATS문 생성 안함
- l: 데이터베이스 레이아웃 생성: 데이터베이스 파티션 그룹, 버퍼 풀 및 테이블 스페이스
- x: 오브젝트의 원래 정의자를 제외한 권한 부여 명령문 DDL 생성
- xd: 오브젝트의 원래 정의자를 포함한 권한 부여 명령문 DDL 생성



- f: 구성 매개변수 및 환경 변수 추출
- td: x를 명령문 분리문자로 지정합니다(디폴트값은 세미콜론(;))임).
- i: 데이터베이스가 상주하는 서버로 로그인하기 위한 사용자 ID
- w: 데이터베이스가 상주하는 서버로 로그인하기 위한 암호
- noview: CREATE VIEW DDL문 생성 안함
- wrapper: 이 래퍼에 적용되는 페더레이티드 오브젝트에 대한 DDL을 생성합니다.
- server: 이 서버에 적용되는 페더레이티드 오브젝트에 대한 DDL을 생성합니다.
- FEDONLY: 작성된 페더레이티드 DDL문만
- nofed: 페더레이티드 DDL 생성 안함
- fd: 기타 cfg 및 env 매개변수와 함께 opt\_buffpage 및 opt\_sortheap에 대한 db2fopt문을 생성하십시오.
- v: 뷰 전용 DDL을 생성하십시오. 이 옵션은 -t가 지정되면 무시됩니다.
- dp: CREATE문 이전에 DROP문 생성
- ct: 오브젝트 작성 시간별로 DDL문 생성

주: 일반적으로 시스템 명령 도움말 화면이 CLP 명령 도움말 화면보다 자세한 정보를 제공합니다.

## 명령 입력 예

다음 UPDATE MONITOR SWITCHES 명령 도움말 화면 사용 예에서는

```
db2 => ? update monitor
UPDATE MONITOR SWITCHES USING {switch-name {ON | OFF} ...}
[AT DBPARTITIONNUM db-partition-number | GLOBAL]
```

```
switch-name:
  BUFFERPOOL, LOCK, SORT, STATEMENT, TABLE, TIMESTAMP, UOW
```

유효한 명령 입력은 다음과 같습니다.

```
UPDATE MONITOR SWITCHES USING LOCK OFF
UPDATE MONITOR SWITCHES USING LOCK OFF TIMESTAMP ON
UPDATE MONITOR SWITCHES USING STATEMENT ON AT DBPARTITIONNUM 1
UPDATE MONITOR SWITCHES USING SORT ON GLOBAL
```

유효하지 않은 명령 입력은 다음과 같습니다.

```
UPDATE MONITOR SWITCHES LOCK OFF
UPDATE MONITOR SWITCHES USING LOCK GLOBAL
UPDATE MONITOR SWITCHES USING STATEMENT ON AT DBPARTITIONNUM 1 GLOBAL
```

## 메모

온라인 정보 센터를 검색하지 않고 명령 도움말 화면 구문 규칙을 확인하려면 CLP 프롬프트에서 다음 명령을 실행하십시오.

```
db2 => ? 도움말
```

또는 시스템 명령 프롬프트에서 다음을 입력하십시오.

```
=> db2 ? 도움말
```



---

## 제 4 부 CLP 명령



---

## 제 7 장 ACTIVATE DATABASE

지정된 데이터베이스를 활성화하고 모든 필요한 데이터베이스 서비스를 시작하여 응용 프로그램이 데이터베이스에 연결하고 사용할 수 있도록 합니다.

### 범위

이 명령은 시스템의 모든 노드의 지정된 데이터베이스를 활성화합니다. 이들 노드 중 하나 이상에서 데이터베이스 활성화 중에 오류가 발생하면 경고가 리턴됩니다. 데이터베이스는 명령이 성공한 모든 노드에서 활성화 상태를 유지합니다.

### 권한 부여

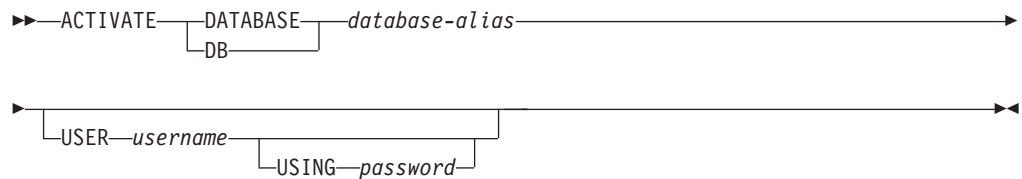
다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

*database-alias*

시작할 데이터베이스의 별명을 지정합니다.

**USER** *username*

데이터베이스를 시작하는 사용자를 지정합니다.

**USING** *password*

사용자 이름의 암호를 지정합니다.

## 사용 시 참고사항

데이터베이스가 시작되지 않았고 응용프로그램에서 CONNECT TO(또는 내재된 연결)가 발행되는 경우 응용프로그램은 해당 데이터베이스에 대해 작업할 수 있기 전에 데이터베이스 관리 프로그램이 필수 데이터베이스를 시작하는 동안 기다려야 합니다. 그러나 데이터베이스가 시작된 후에는 기타 응용프로그램이 시작하는 데 시간을 소비하지 않고 간단히 연결하고 사용할 수 있습니다.

데이터베이스 관리자가 ACTIVATE DATABASE를 사용하여 선택한 데이터베이스를 시작할 수 있습니다. 이는 데이터베이스 초기화에서 응용프로그램 시간이 소비되지 않게 합니다.

ACTIVATE DATABASE로 초기화된 데이터베이스는 DEACTIVATE DATABASE 명령을 사용하거나 db2stop 명령을 사용하여 종료할 수 있습니다.

데이터베이스가 CONNECT TO(또는 내재된 연결)로 시작되었고 뒤에 동일한 데이터베이스에 대해 ACTIVATE DATABASE가 발행되는 경우 DEACTIVATE DATABASE를 사용하여 해당 데이터베이스를 종료해야 합니다. ACTIVATE DATABASE를 사용하여 데이터베이스를 시작하지 않은 경우 데이터베이스는 마지막 응용프로그램의 연결이 끊길 때 종료합니다.

ACTIVATE DATABASE는 재시작이 필요한 데이터베이스(예를 들어 불일치 상태에 있는 데이터베이스)에 대해 작업할 때 CONNECT TO(또는 내재된 연결)와 비슷한 방식으로 동작합니다. 데이터베이스는 ACTIVATE DATABASE로 초기화할 수 있기 전에 재시작됩니다. 재시작은 데이터베이스가 AUTORESTART ON을 갖도록 구성되는 경우에만 수행됩니다.

ACTIVATE DATABASE 명령을 발행하는 응용프로그램은 어떤 데이터베이스에도 활성 데이터베이스 연결을 가질 수 없습니다.

## 제 8 장 ADD CONTACT

이 명령은 시스템 또는 전역 목록에서 로컬로 정의할 수 있는 문의처 목록에 문의처를 추가합니다. 문의처는 스케줄러 및 Health Monitor와 같은 프로세스가 메시지를 보낼 수 있는 사용자입니다. DAS(Database Administration Server) **contact\_host** 구성 매개변수의 설정은 목록이 로컬인지 또는 전역인지를 판별합니다.

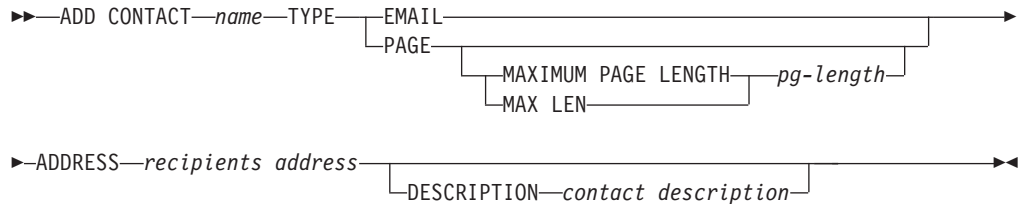
### 권한 부여

없음

### 필수 연결

없음. 로컬 실행 전용: 이 명령은 리모트 연결에 사용할 수 없습니다.

### 명령 구문



### 명령 매개변수

#### ADD CONTACT *name*

추가될 문의처 이름. DB2 관리 서버 구성 매개변수 **contact\_host**가 다른 시스템을 지정하는 경우를 제외하고는 문의처는 디폴트로 로컬 시스템에 추가됩니다.

**TYPE** 문의처의 메소드로서 다음 중 하나가 되어야 합니다.

#### EMAIL

이 문의처는 (ADDRESS)에서 전자 우편에 의해 통지됩니다.

#### PAGE

이 문의처는 ADDRESS로 보내진 페이지에 의해 통지됩니다.

#### MAXIMUM PAGE LENGTH *pg-length*

페이징 서비스에 메시지 길이 제한이 있는 경우 이 예에서 문자로 지정됩니다.

통지 시스템은 SMTP 프로토콜을 사용하여 DB2 관리 서버 구성 매개변수 **smtp\_server** 에 의해 지정된 메일 서버로 통지를 보냅니다. 전자 우편을 보내거나 호출기를 호출하는 것은 SMTP 서버의 역할입니다.

**ADDRESS** *recipients-address*

받는 사람의 SMTP 편지함 주소(예: joe@somewhere.org). **smtp\_server** DAS 구성 매개변수는 SMTP 서버의 이름으로 설정되어야 합니다.

**DESCRIPTION** *contact description*

문의처의 텍스트 설명. 최대 128 문자를 사용합니다.



---

## 제 9 장 ADD CONTACTGROUP

로컬 시스템에 정의된 그룹 목록에 새 문의처 그룹을 추가합니다. 문의처 그룹은 스케줄러 및 Health Monitor와 같은 모니터링 프로세스가 메시지를 보낼 수 있는 사용자 및 그룹의 목록입니다. DAS(Database Administration Server) **contact\_host** 구성 매개변수의 설정은 목록이 로컬인지 또는 전역인지를 판별합니다.

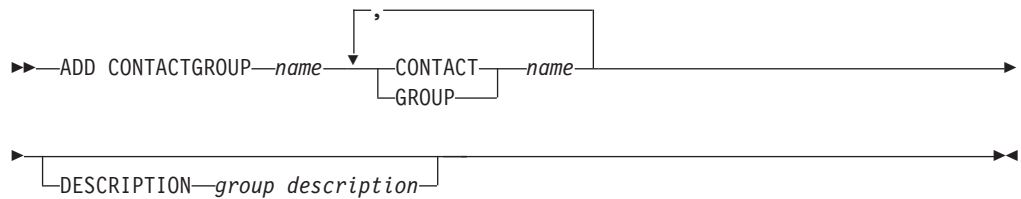
### 권한 부여

없음

### 필수 연결

없음. 로컬 실행 전용: 이 명령은 리모트 연결에 사용할 수 없습니다.

### 명령 구문



### 명령 매개변수

#### **ADD CONTACTGROUP** *name*

새 문의처 그룹 이름으로, 시스템에 있는 그룹 세트 중에서 고유해야 합니다.

#### **CONTACT** *name*

그룹의 구성원인 문의처 이름. 문의처를 그룹에 추가한 후 ADD CONTACT 명령을 사용하여 정의할 수 있습니다.

#### **GROUP** *name*

이 그룹이 구성원인 문의처 그룹 이름

#### **DESCRIPTION** *group description*

선택사항. 문의처 그룹의 텍스트 설명.



---

## 제 10 장 ADD DBPARTITIONNUM

데이터베이스 파티션 서버에 데이터베이스 파티션을 추가합니다.

### 범위

이 명령은 실행된 데이터베이스 파티션 서버에만 영향을 줍니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*

### 필수 연결

없음

### 명령 구문

```
▶▶—ADD DBPARTITIONNUM—┬──────────────────────────────────────────────────────────────────────────────────┬──▶  
                        └─LIKE DBPARTITIONNUM—db-partition-number—┘  
                        └─WITHOUT TABLESPACES—┘
```

### 명령 매개변수

#### LIKE DBPARTITIONNUM *db-partition-number*

새 시스템 임시 테이블 스페이스의 컨테이너가 *db-partition-number*에 지정된 데이터베이스 파티션 서버에 있는 데이터베이스의 컨테이너와 같음을 지정합니다. 지정된 데이터베이스 파티션 서버는 이미 *db2nodes.cfg* 파일에 정의되어 있어야 합니다.

자동 스토리지를 사용하도록 정의된 시스템 임시 테이블 스페이스(즉, CREATE TABLESPACE문의 MANAGED BY AUTOMATIC STORAGE절로 작성된 테이블 스페이스 또는 MANAGED BY절이 지정되지 않은 테이블 스페이스)의 경우, 컨테이너는 지정된 파티션의 컨테이너와 반드시 일치할 필요는 없습니다. 대신, 컨테이너는 데이터베이스와 연관되는 스토리지 경로를 기반으로 하는 데이터베이스 관리 프로그램에 의해 자동으로 지정됩니다. 이로써 두 파티션에서 동일한 컨테이너가 사용될 수도, 그렇지 않을 수도 있습니다.

#### WITHOUT TABLESPACES

데이터베이스 파티션에 대해 시스템 임시 테이블 스페이스 컨테이너가 작성되

지 않음을 지정합니다. ALTER TABLESPACE문을 사용하여 데이터베이스를 사용하기 전에 각 데이터베이스 파티션에 시스템 임시 테이블 스페이스를 추가해야 합니다.

옵션을 지정하지 않은 경우 시스템 임시 테이블 스페이스의 컨테이너는 각 데이터베이스의 카탈로그 파티션에 있는 컨테이너와 동일하게 됩니다. 카탈로그 파티션은 파티션된 데이터베이스 환경에서 데이터베이스마다 다른 데이터베이스 파티션이 될 수 있습니다. 자동 스토리지를 사용하도록 정의된 시스템 임시 테이블 스페이스(즉, CREATE TABLESPACE문의 MANAGED BY AUTOMATIC STORAGE 절로 작성되었거나 MANAGED BY절이 전혀 지정되지 않은 시스템 임시 테이블 스페이스)에 대해서는 이 옵션이 무시됩니다. 이 테이블 스페이스의 경우 컨테이너 작성을 지연할 방법이 없습니다. 컨테이너는 데이터베이스에 연결된 스토리지 경로를 기반으로 하는 데이터베이스 관리 프로그램에 의해 자동으로 할당됩니다.

## 사용 시 참고사항

이 명령은 데이터베이스 파티션 서버가 하나의 데이터베이스를 포함하며 파티션 추가 조작 시에 데이터베이스가 카탈로그되지 않은 환경에 추가되는 경우에만 사용해야 합니다. 이 상황에서는 데이터베이스가 카탈로그되지 않기 때문에 파티션 추가 조작이 데이터베이스를 인식하지 못하고 새 데이터베이스 파티션 서버의 데이터베이스에 대해 데이터베이스 파티션을 작성하지 않습니다. 새 데이터베이스 파티션 서버에서 데이터베이스 파티션 연결 시도는 오류를 초래합니다. 데이터베이스는 ADD DBPARTITIONNUM 명령을 사용하여 새 데이터베이스 파티션 서버에서 데이터베이스에 대한 데이터베이스 파티션을 작성하기 전에 우선 카탈로그해야 합니다.

이 명령은 둘 이상의 데이터베이스가 있고 이 데이터베이스 중 하나 이상이 파티션 추가 조작 시에 카탈로그된 경우 사용하면 안됩니다. 이 상황에서는 CREATE DATABASE 명령의 AT DBPARTITIONNUM 매개변수를 사용하여 파티션 추가 조작 시에 카탈로그되지 않은 각 데이터베이스에 대해 데이터베이스 파티션을 작성하십시오. 카탈로그되지 않은 각 데이터베이스는 CREATE DATABASE 명령을 사용하여 새 데이터베이스 파티션 서버에서 데이터베이스에 대해 데이터베이스 파티션을 작성하기 전에 우선 카탈로그해야 합니다.

새 데이터베이스 파티션을 추가하기 전에 작성해야 하는 컨테이너에 필요한 스토리지가 충분한지 확인하십시오.

데이터베이스 파티션 서버 추가 조작은 인스턴스에 있는 모든 데이터베이스에 대해 비어 있는 데이터베이스 파티션을 작성합니다. 새 데이터베이스 파티션의 구성 매개변수는 디폴트값으로 설정됩니다.

주: 카탈로그 해제된 데이터베이스는 새 데이터베이스 파티션을 추가할 때 인식되지 않습니다. 카탈로그 해제된 데이터베이스는 새 데이터베이스 파티션에 표시되지 않습니다. 새 데이터베이스 파티션에서 데이터베이스에 연결하려고 하면 오류 메시지 SQL1013N 이 리턴됩니다.

로컬에서 데이터베이스 파티션 작성 시 데이터베이스 파티션 서버 노드 추가 조작이 실패하면 제거 단계가 되고 이 경우 작성된 모든 데이터베이스가 로컬에서 삭제됩니다. 즉, 데이터베이스 파티션은 추가 중인 데이터베이스 파티션 서버에서만 제거됩니다. 기존 데이터베이스 파티션은 모든 다른 데이터베이스 파티션 서버에서 이전과 동일하게 유지됩니다. 제거 단계가 실패하면 추가적인 제거는 수행되지 않으며 오류가 리턴됩니다.

새 데이터베이스 파티션 서버의 데이터베이스 파티션은 ALTER DATABASE PARTITION GROUP 문을 사용하여 데이터베이스 파티션을 데이터베이스 파티션 그룹에 추가할 때까지 사용자 데이터를 포함할 수 없습니다.

이 명령은 데이터베이스 작성 또는 데이터베이스 삭제 조작이 진행 중인 경우 실패합니다. 경쟁 조작이 완료되면 명령을 다시 실행할 수 있습니다.

데이터베이스의 자동 스토리지 사용 가능 여부를 판별하기 위해 ADD DBPARTITIONNUM이 인스턴스의 각 데이터베이스에 대해 카탈로그 파티션과 통신해야 합니다. 자동 스토리지가 사용 가능한 경우 스토리지 경로 정의는 통신 중에 검색됩니다. 이와 유사하게 시스템 임시 테이블 스페이스가 데이터베이스 파티션과 함께 작성되는 경우 ADD DBPARTITIONNUM은 해당 서버에 있는 데이터베이스 파티션의 테이블 스페이스 정의를 검색하기 위해 다른 데이터베이스 파티션 서버와 통신해야 합니다. **start\_stop\_time** 데이터베이스 관리 프로그램 구성 매개변수를 사용하여 다른 데이터베이스 파티션 서버가 자동 스토리지 및 테이블 스페이스 정의에 응답해야 하는 시간을 분 단위로 지정합니다. 이 시간이 초과되면 명령은 실패합니다. 이와 같은 상황이 발생하면 **start\_stop\_time** 값을 증가시키고 명령을 다시 실행하십시오.

## 호환성

버전 8 이전 버전과의 호환성:

- NODE 키워드는 DBPARTITIONNUM을 대체할 수 있습니다.



## 제 11 장 ADD XMLSCHEMA DOCUMENT

등록을 완료하기 전에 기존의 불완전한 XML 스키마에 하나 이상의 XML 스키마 문서를 추가합니다.

### 권한 부여

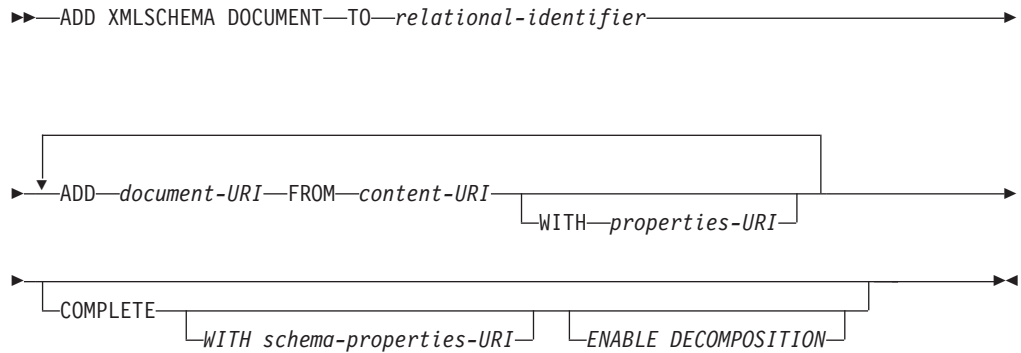
다음 권한이 필요합니다.

- 사용자 ID는 카탈로그 뷰 SYSCAT.XSROBJECTS에 기록된 대로 XSR 오브젝트의 소유자여야 합니다.

### 필수 연결

데이터베이스

### 명령 구문



### 설명

#### TO relational-identifier

추가 스키마 문서가 추가되는, 등록되었지만 불완전한 XML 스키마의 관계형 이름을 지정합니다.

#### ADD document-URI

문서가 다른 XML 문서에서 참조되어, 해당 스키마에 추가될 XML 스키마 문서의 URI를 지정합니다.

#### FROM content-URI

XML 스키마 문서가 위치해 있는 URI를 지정합니다. 파일 스킴 URI만 지원됩니다.

**WITH** *properties-URI*

XML 스키마의 등록 정보 문서 URI를 지정합니다. 파일 스킴 URI만 지원됩니다.

**COMPLETE**

추가할 XML 스키마 문서가 더 이상 없음을 표시합니다. 지정하는 경우, 스키마의 유효성이 확인되고 오류가 발견되지 않으면 사용 가능한 것으로 표시됩니다.

**WITH** *schema-properties-URI*

XML 스키마의 등록 정보 문서 URI를 지정합니다. 파일 스킴 URI만 지원됩니다.

**ENABLE DECOMPOSITION**

스키마가 XML 문서 분석에 사용됨을 지정합니다.

예 :

```
ADD XMLSCHEMA DOCUMENT TO JOHNDOE.PRODSHEMA
  ADD 'http://myPOschema/address.xsd'
  FROM 'file:///c:/TEMP/address.xsd'
```



---

## 제 12 장 ARCHIVE LOG

복구 가능한 데이터베이스의 활성 로그 파일을 닫고 절단합니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*

### 필수 연결

없음. 이 명령은 명령 지속기간 동안 데이터베이스 연결을 설정합니다.

### 명령 구문

```
▶▶ ARCHIVE LOG FOR DATABASE database-alias
   DB
USER username USING password
On Database Partition Number Clause
```

#### On Database Partition Number Clause:

```
ON ALL DBPARTITIONNUMS Database Partition Number List Clause
EXCEPT Database Partition Number List Clause
```

#### Database Partition Number List Clause:

```
DBPARTITIONNUM
DBPARTITIONNUMS
( db-partition-number TO db-partition-number )
```

## 명령 매개변수

### **DATABASE** *database-alias*

사용 중인 로그가 아카이브될 데이터베이스의 별명을 지정합니다.

### **USER** *username*

연결을 시도할 사용자 이름을 식별합니다.

### **USING** *password*

사용자 이름을 인증할 암호를 지정합니다.

### **ON ALL DBPARTITIONNUMS**

db2nodes.cfg 파일에서 모든 데이터베이스 파티션에 대해 명령이 실행되어야 하는 것으로 지정합니다. 데이터베이스 파티션 번호 절이 지정되지 않는 경우 디폴트값입니다.

### **EXCEPT**

데이터베이스 파티션 번호 목록에 지정된 파티션을 제외하고 db2nodes.cfg 파일에서 모든 데이터베이스 파티션에 대해 명령이 실행되어야 하는 것으로 지정합니다.

### **ON DBPARTITIONNUM | ON DBPARTITIONNUMS**

데이터베이스 파티션 세트에 지정된 데이터베이스에 대해 로그가 아카이브되어야 하는 것으로 지정합니다.

### *db-partition-number*

데이터베이스 파티션 번호 목록의 데이터베이스 파티션 번호를 지정합니다.

### **TO** *db-partition-number*

로그를 아카이브해야 하는 데이터베이스 파티션의 범위를 지정할 때 사용됩니다. 지정된 두 번째 데이터베이스 파티션 번호를 포함하여 지정된 첫 번째 데이터베이스 파티션 번호의 모든 데이터베이스 파티션은 데이터베이스 파티션 번호 목록에 포함됩니다.

## 사용 시 참고사항

이 명령은 로그 파일의 전체 세트를 알려진 지점까지 수집하는 데 사용할 수 있습니다. 로그 파일은 대기 데이터베이스를 갱신하는 데 사용할 수 있습니다.

이 명령은 호출 응용프로그램 또는 셸에 지정된 데이터베이스에 대한 데이터베이스 연결이 없는 경우에만 실행할 수 있습니다. 이 경우 사용자가 커밋되지 않은 트랜잭션으로 명령을 실행할 수 없습니다. 따라서 ARCHIVE LOG 명령은 사용자의 불완전 트랜잭션을 강제로 커밋하지 않습니다. 호출 응용프로그램 또는 셸에 이미 지정된 데이터베이스에 대한 데이터베이스 연결이 있는 경우 이 명령이 종료되고 오류를 리턴합니다. 이 명령이 실행될 때 다른 응용프로그램에 지정된 데이터베이스 관련 트랜잭션이 진

행 중인 경우 명령이 로그 버퍼를 디스크로 비우므로 성능이 약간 저하됩니다. 로그 레코드를 버퍼에 쓰려는 다른 트랜잭션 시도는 비우기가 완료될 때까지 대기해야 합니다.

파티션된 데이터베이스 환경에서 사용하는 경우 데이터베이스 파티션 번호 절을 사용하여 데이터베이스 파티션 서브셋을 지정할 수 있습니다. 데이터베이스 파티션 번호 절이 지정되지 않는 경우 이 명령의 디폴트 동작은 모든 데이터베이스 파티션에서 사용 중인 로그를 닫고 아카이브하는 것입니다.

이 명령을 사용하면 사용 중인 로그 파일이 잘리므로 사용 중인 로그 공간의 일부를 사용하게 됩니다. 사용 중인 로그 공간은 잘린 로그가 비활성화되면 이전 크기를 다시 시작합니다. 이 명령을 자주 사용하면 트랜잭션에 사용할 수 있는 사용 중인 로그 공간의 양이 현저히 감소할 수 있습니다.

## 호환성

버전 8 이전 버전과의 호환성:

- NODE 키워드는 DBPARTITIONNUM을 대체할 수 있습니다.
- NODES 키워드는 DBPARTITIONNUMS를 대체할 수 있습니다.



## 제 13 장 ATTACH

응용프로그램이 인스턴스 레벨 명령(예: CREATE DATABASE 및 FORCE APPLICATION)이 실행될 인스턴스를 지정할 수 있게 합니다. 이 인스턴스는 현재 인스턴스, 동일한 워크스테이션의 다른 인스턴스 또는 리모트 워크스테이션의 인스턴스일 수 있습니다.

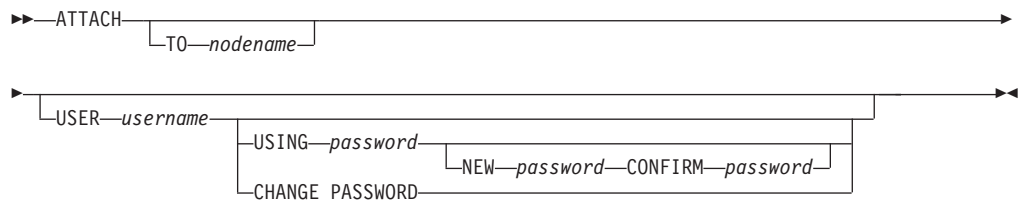
### 권한 부여

없음

### 필수 연결

없음. 이 명령은 인스턴스 접속을 설정합니다.

### 명령 구문



### 명령 매개변수

#### TO *nodename*

사용자가 접속하려는 인스턴스의 별명입니다. 이 인스턴스는 로컬 노드 디렉토리에 일치하는 항목이 있어야 합니다. 이에 대한 유일한 예외는 접속의 오브젝트로 지정할 수 있지만 노드 디렉토리에서 노드 이름으로 사용할 수 없는 로컬 인스턴스(DB2INSTANCE 환경 변수로 지정되는)입니다.

#### USER *username*

인증 ID를 지정합니다. Windows 운영 체제의 DB2 데이터베이스 인스턴스에 접속할 때, 사용자 이름은 Microsoft® SAM(Security Account Manager)과 호환 가능한 형식으로 지정할 수 있습니다. 규정자는 최대 길이가 15문자인 플랫폼 스타일(NetBIOS형) 이름이어야 합니다. 예: *domainname#username*.

#### USING *password*

사용자 이름의 암호를 지정합니다. 사용자 이름이 지정되지만 암호가 지정되지 않는 경우, 사용자에게 현재 암호를 입력하도록 프롬프트합니다. 암호는 입력 시에 표시되지 않습니다.

### **NEW password**

사용자 이름에 지정될 새 암호를 지정합니다. 암호를 변경할 시스템은 사용자 인증이 설정된 방식에 따라 다릅니다.DB2 데이터베이스 시스템은 AIX, Linux 및 Windows 운영 체제에서 암호 변경 지원을 제공하고 사용자 자신이 작성한 플러그인에 대해 최대 255문자를 지원합니다. 암호에 대한 추가 정보는 암호 규칙을 참조하십시오.

### **CONFIRM password**

새 암호와 동일해야 하는 문자열입니다. 이 매개변수는 입력 오류를 포착하는데 사용됩니다.

### **CHANGE PASSWORD**

이 옵션이 지정되면 사용자에게 현재 암호, 새 암호 및 새 암호의 확인이 프롬프트됩니다. 암호는 입력 시에 표시되지 않습니다.

### **예:**

두 리모트 노드를 카탈로그하십시오.

```
db2 catalog tcpip node node1 remote freedom server server1
db2 catalog tcpip node node2 remote flash server server1
```

첫 번째 노드에 접속하고, 모든 사용자를 강제 종료시킨 후 접속 해제하십시오.

```
db2 attach to node1
db2 force application all
db2 detach
```

두 번째 노드에 접속하여 활동 중인 사용자를 확인하십시오.

```
db2 attach to node2
db2 list applications
```

명령이 에이전트 ID 1, 2 및 3을 리턴할 때 1과 3을 강제 종료시킨 후 접속 해제하십시오.

```
db2 force application (1, 3)
db2 detach
```

현재 인스턴스에 접속하고(필수는 아니며, 내재됨), 모든 사용자를 강제 종료시킨 후 접속 해제하십시오(AIX만 해당).

```
db2 attach to $DB2INSTANCE
db2 force application all
db2 detach
```

### **사용 시 참고사항**

*nodename*이 명령에서 생략된 경우 접속의 현재 상태에 대한 정보가 리턴됩니다.

ATTACH가 실행되지 않은 경우 인스턴스 레벨 명령은 **DB2INSTANCE** 환경 변수로 지정되는 현재 인스턴스에 대해 실행됩니다.





## 제 14 장 AUTOCONFIGURE

이들 권장 값을 적용하는 옵션을 사용하여 버퍼 풀 크기, 데이터베이스 구성 및 데이터베이스 관리 프로그램 구성 매개변수에 대한 초기값을 개선하여 표시합니다.

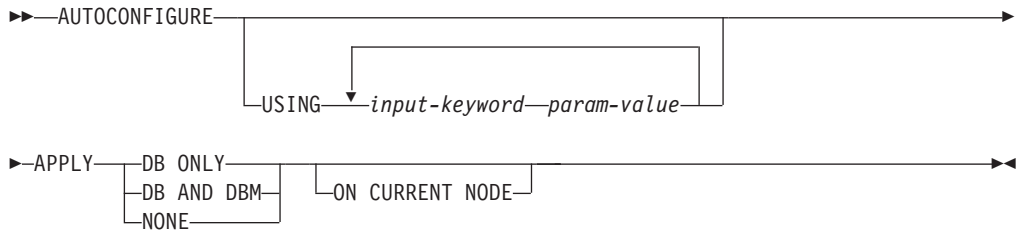
### 권한 부여

*sysadm*

### 필수 연결

데이터베이스

### 명령 구문



### 명령 매개변수

**USING** *input-keyword param-value*

표 6. 유효한 입력 키워드 및 매개변수 값

키워드	유효한 값	디폴트값	설명
mem_percent	1-100	25	전용으로 사용되는 인스턴스 메모리의 초기 백분율. 시스템에서 현재 메모리 사용에 따라 연속 데이터베이스에 대해 구성 어드바이저가 이 값을 조정합니다.
workload_type	simple, mixed, complex	mixed	단순 워크로드는 I/O 집중적이고 주로 트랜잭션인 경향이 있는 반면, 복합 워크로드는 CPU 집중적이며 주로 쿼리로 이루어지는 경향이 있습니다.
num_stmts	1-1 000 000	10	작업 단위(UOW) 당 명령문 수
tpm	1-200 000	60	분 당 트랜잭션 수

표 6. 유효한 입력 키워드 및 매개변수 값 (계속)

키워드	유효한 값	디폴트값	설명
admin_priority	performance, recovery, both	both	성능 향상(분 당 더 많은 트랜잭션 수) 또는 보다 짧은 복구 시간을 위한 최적화
is_populated	yes, no	yes	데이터베이스가 데이터로 채워지는지 여부
num_local_apps	0-5 000	0	연결된 로컬 응용프로그램 수
num_remote_apps	0-5 000	10	연결된 리모트 응용프로그램 수
isolation	RR, RS, CS, UR	RR	이 데이터베이스에 연결 중인 응용프로그램의 최대 분리 레벨(반복 읽기, 읽기 안정성, 커서 안정성, 언커미트 읽기). 다른 구성 매개변수의 값을 판별하는 데만 사용됩니다. 응용프로그램을 특정 분리 레벨로 제한하려면 아무 것도 설정하지 않으며 디폴트값을 사용하는 것이 안전합니다.
bp_resizeable	yes, no	yes	버퍼 풀의 크기를 조정할 수 있는지 여부

## APPLY

### DB ONLY

현재 데이터베이스 관리 프로그램 구성에 기초하여 데이터베이스 구성 및 버퍼 풀 설정에 대한 권장 값을 표시합니다. 데이터베이스 구성 및 버퍼 풀 설정에 대한 권장 변경사항을 적용합니다.

### DB AND DBM

데이터베이스 관리 프로그램 구성, 데이터베이스 구성 및 버퍼 풀 설정에 대한 권장 변경사항을 표시하고 적용합니다.

### NONE

권장 변경사항을 표시하지만 이를 적용하지 않습니다.

## ON CURRENT NODE

DPF(Database Partitioning Feature)에서 구성 어드바이저가 디폴트로 모든 노드의 데이터베이스 구성을 갱신합니다. ON CURRENT NODE 옵션을 사용하여 실행하면 어드바이저가 권장 데이터베이스 구성을 코디네이터(연결) 노드에만 적용합니다.

버퍼 풀 변경사항은 시스템 키탈로그에 항상 적용됩니다. 따라서 모든 노드가 영향을 받습니다. ON CURRENT NODE 옵션은 버퍼 풀 권장사항에 중요하지 않습니다.

## 사용 시 참고사항

- 다중 로컬 파티션이 있는 시스템에서 **mem\_percent** 매개변수는 모든 논리 파티션에서 사용될 메모리의 백분율을 나타냅니다. 예를 들어, DB2가 시스템에서 25%의 메모리를 사용할 경우, 논리 파티션 수에 관계없이 25%를 지정하십시오. 그러나 데이터베이스 구성 권장사항은 한 논리 파티션에 대해 조정됩니다.
- 이 명령은 현재 연결된 데이터베이스만 시스템에서 사용 중이라고 가정하고 이 데이터베이스에 대한 구성 권장사항을 작성합니다. 시스템에서 두 개 이상의 데이터베이스를 사용 중인 경우 **mem\_percent** 매개변수를 조정하여 현재 데이터베이스의 메모리 공유를 반영하십시오. 예를 들어, DB2 데이터베이스가 시스템 메모리의 80%를 사용하며 시스템에 있는 두 개의 사용 중인 데이터베이스가 자원을 동일하게 공유해야 하는 경우 **mem\_percent** 매개변수에 대해 40%(80%를 두 개의 데이터베이스로 나눔)를 지정하십시오.
- AUTOCONFIGURE 명령으로 구성 어드바이저를 명시적으로 호출할 때 DB2\_ENABLE\_AUTOCONFIG\_DEFAULT 레지스트리 변수의 설정을 무시합니다.
- 데이터베이스에 대해 AUTOCONFIGURE 명령을 실행시 자체 성능 조정 메모리 관리자를 사용 가능하게 할 것을 권장합니다. 그러나 SHEAPTHRES가 0이 아닌 인스턴스에서 데이터베이스에 대해 AUTOCONFIGURE 명령을 실행할 경우, 정렬 메모리 조정(SORTHEAP)은 자동으로 사용할 수 있게 되지 않습니다. 정렬 메모리 조정(SORTHEAP)을 사용할 수 있게 하려면 UPDATE DATABASE MANAGER CONFIGURATION 명령을 사용하여 SHEAPTHRES를 0으로 설정해야 합니다. SHEAPTHRES의 값을 변경하면 이전의 기존 데이터베이스에서의 정렬 메모리 사용에 영향을 줄 수 있다는 점을 참고하십시오.



## 제 15 장 BACKUP DATABASE

데이터베이스 또는 테이블 스페이스의 백업 사본을 작성합니다.

다른 운영 체제 및 하드웨어 플랫폼 간에 DB2 데이터베이스 시스템에서 제공하는 백업 조작에 대한 정보를 보려면 『다른 운영 체제 및 하드웨어 플랫폼 간의 백업 및 리스토어 조작』을 참조 하십시오.

### 범위

파티션된 데이터베이스 환경에서 데이터베이스 파티션이 지정되지 않으면, 이 명령은 명령이 실행되는 데이터베이스 파티션에만 영향을 미칩니다.

파티션된 백업을 수행하는 옵션을 지정하면, 카탈로그 노드에서만 명령을 호출할 수 있습니다. 모든 데이터베이스 파티션 서버를 백업하도록 옵션을 지정한 경우, db2nodes.cfg 파일에 나열된 모든 데이터베이스 파티션에 영향을 미칩니다. 그렇지 않으면, 명령에서 지정하는 데이터베이스 파티션 서버에 영향을 미칩니다.

### 권한 부여

다음 중 하나가 필요합니다.

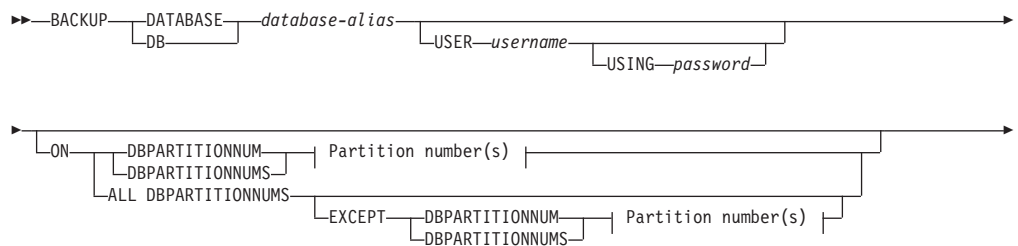
- *sysadm*
- *sysctrl*
- *sysmaint*

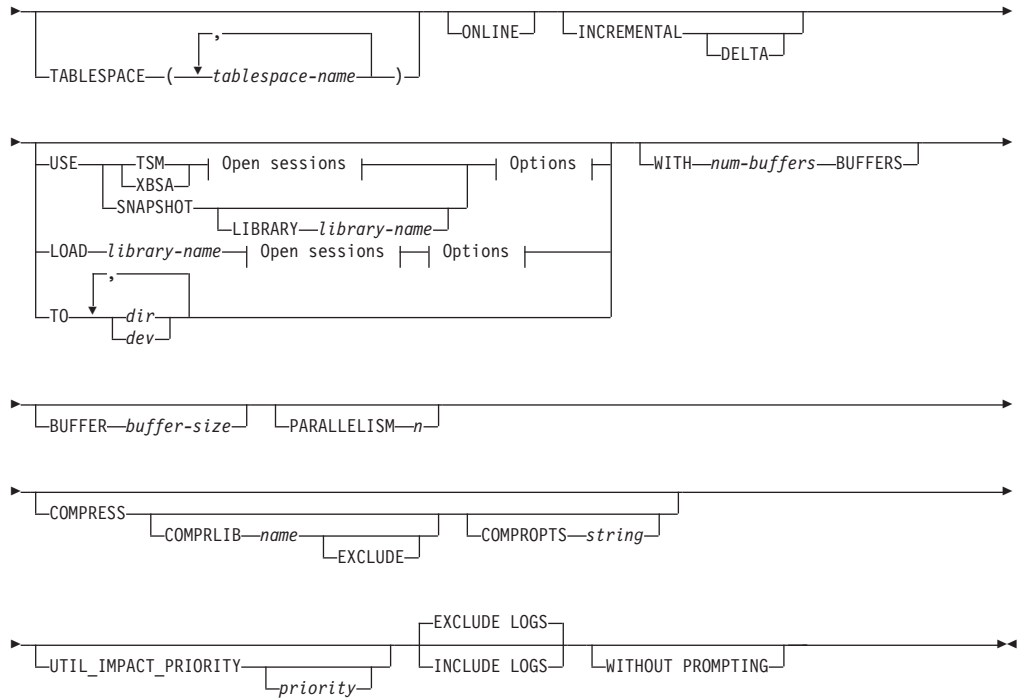
### 필수 연결

데이터베이스. 이 연결은 지정된 데이터베이스에 대한 연결을 자동으로 설정합니다.

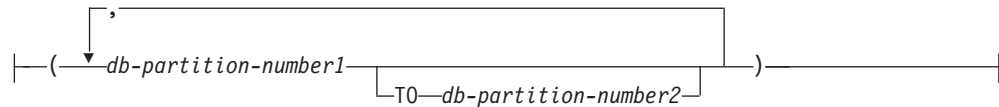
주: 지정된 데이터베이스로 이미 연결되어 있는 경우 이 연결을 종료하고 백업 조작에 특정하게 새 연결을 설정합니다. 이 연결은 백업 조작이 완료될 때 종료됩니다.

### 명령 구문

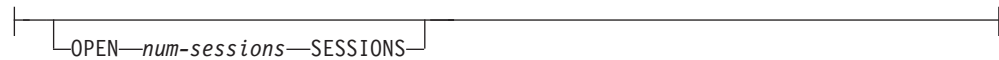




**Partition number(s):**



**Open sessions:**



**Options:**



**명령 매개변수**

**DATABASE | DB** *database-alias*

백업할 데이터베이스의 별명을 지정합니다.

**USER** *username*

데이터베이스를 백업할 사용자 이름을 식별합니다.

**USING** *password*

사용자 이름을 인증하는 데 사용되는 암호. 암호를 생략하면 사용자가 이를 입력하도록 프롬프트가 표시됩니다.

**ON** 데이터베이스 파티션 세트의 데이터베이스를 백업합니다.

**DBPARTITIONNUM** *db-partition-number1*

데이터베이스 파티션 목록의 데이터베이스 파티션 번호를 지정합니다.

**DBPARTITIONNUMS** *db-partition-number1 TO db-partition-number2*

데이터베이스 파티션 번호 범위를 지정하여, *db-partition-number1*부터 *db-partition-number2*까지의 모든 파티션을 데이터베이스 파티션 목록에 포함합니다.

**ALL DBPARTITIONNUMS**

*db2nodes.cfg* 파일에 지정된 모든 파티션에서 데이터베이스를 백업하도록 지정합니다.

**EXCEPT**

데이터베이스 파티션 목록에 지정된 파티션을 제외하고, *db2nodes.cfg* 파일에 지정된 모든 파티션에서 데이터베이스를 백업하도록 지정합니다.

**DBPARTITIONNUM** *db-partition-number1*

데이터베이스 파티션 목록의 데이터베이스 파티션 번호를 지정합니다.

**DBPARTITIONNUMS** *db-partition-number1 TO db-partition-number2*

데이터베이스 파티션 번호 범위를 지정하여, *db-partition-number1*부터 *db-partition-number2*까지의 모든 파티션을 데이터베이스 파티션 목록에 포함합니다.

**TABLESPACE** *tablespace-name*

백업될 테이블 스페이스를 지정하는 데 사용되는 이름 목록.

**ONLINE**

온라인 백업을 지정합니다. 디폴트값은 오프라인 백업입니다. 온라인 백업은 *logretain* 또는 *userexit*를 사용할 수 있도록 하여 구성된 데이터베이스에만 사용할 수 있습니다. 온라인 백업 중, DB2는 SMS 테이블 스페이스에 존재하는 모든 테이블에 대해 처리 시 IN(의도 없음) 잠금을 확보합니다. 온라인 백업 중 SMS 테이블 스페이스의 LOB 데이터에 대해서는 S(공유 잠금)를 보유하지 않습니다.

## INCREMENTAL

누적(증분) 백업 이미지를 지정합니다. 증분 백업 이미지는 가장 최근에 완료된 전체 백업 조작 이후 변경된 모든 데이터베이스 데이터의 사본입니다.

## DELTA

비누적(델타) 백업 이미지를 지정합니다. 델타 백업 이미지는 가장 최근에 완료된 모든 유형의 백업 조작 이후 변경된 모든 데이터베이스 데이터의 사본입니다.

## USE

**TSM** 백업에 TSM(Tivoli® Storage Manager) 출력이 사용되도록 지정합니다.

**XBSA** XBSA 인터페이스가 사용되도록 지정합니다. 백업 서비스 API(XBSA)는 백업 또는 아카이브용으로 데이터 스토리지 관리를 필요로 하는 응용프로그램 또는 기능에 대한 개방 API입니다.

## SNAPSHOT

스냅샷 백업이 사용되도록 지정합니다.

다음 매개변수가 있는 SNAPSHOT 매개변수를 사용할 수 없습니다.

- TABLESPACE
- INCREMENTAL
- WITH *num-buffers* BUFFERS
- BUFFER
- PARALLELISM
- COMPRESS
- UTIL\_IMPACT\_PRIORITY
- SESSIONS

스냅샷 백업의 디폴트 동작은 모든 경로의 FULL DATABASE OFFLINE 백업으로, 모든 컨테이너, 로컬 볼륨 디렉토리, 데이터베이스 경로(DBPATH) 및 1차 로그 및 미러 로그 경로를 작성합니다 (EXCLUDE LOGS가 명시적으로 지정된 경우를 제외하고 INCLUDE LOGS는 모든 스냅샷 백업의 디폴트임).

## LIBRARY *library-name*

IBM Data Server에 통합되며 다음 스토리지 하드웨어의 DB2 ACS API 드라이버입니다.

- IBM TotalStorage SAN 볼륨 제어기
- IBM 엔터프라이즈 스토리지 서버 모델 800
- IBM 시스템 스토리지 DS6000



- IBM 시스템 스토리지 DS8000
- IBM 시스템 스토리지 N 시리즈
- NetApp V-series
- NetApp FAS

다른 스토리지 하드웨어가 있으며 해당 스토리지 하드웨어의 DB2 ACS API 드라이버가 있는 경우, LIBRARY 매개변수를 사용하여 DB2 ACS API 드라이버를 지정할 수 있습니다.

LIBRARY 매개변수의 값은 완전한 라이브러리 파일 이름입니다.

## OPTIONS

### *"options-string"*

백업 작업에 사용되는 옵션을 지정합니다. 이 문자열은 큰따옴표 없이 입력된 그대로 정확히 DB2 ACS API 드라이버로 전달됩니다. **VENDOROPT** 데이터베이스 구성 매개변수를 사용하여 스냅샷 백업 작업을 위한 벤더 고유의 옵션을 지정할 수 없습니다. 대신 백업 유틸리티의 **OPTIONS** 매개변수를 사용해야 합니다.

### *@ file-name*

백업 작업에 사용될 옵션이 DB2 서버에 있는 파일에 들어 있음을 지정합니다. 이 문자열은 벤더 지원 라이브러리로 전달됩니다. 파일은 완전한 파일 이름이어야 합니다.

## OPEN *num-sessions* SESSIONS

DB2와 TSM 또는 다른 백업 벤더 제품 사이에 작성될 I/O 세션 수입니다. 이 매개변수는 테이프, 디스크 또는 다른 로컬 디바이스에 백업할 때는 유효하지 않습니다.

## TO *dir | dev*

디렉토리 또는 테이프 디바이스 이름 목록입니다. 디렉토리가 있는 전체 경로를 지정해야 합니다. USE TSM, TO 및 LOAD가 생략되면, 백업 이미지의 디폴트 대상 디렉토리는 클라이언트 컴퓨터의 현재 작업 디렉토리입니다. 이 목표 디렉토리 또는 디바이스는 데이터베이스 서버에 있어야 합니다.

파티션된 데이터베이스에서, 대상 디렉토리나 디바이스는 모든 데이터베이스 파티션에 있어야 하며, 선택적으로 공유 경로일 수 있습니다. 디렉토리나 디바이스는 데이터베이스 파티션 표현식을 사용하여 지정될 수 있습니다. 데이터베이스 파티션 표현식에 대한 자세한 정보는 *자동 스토리지 데이터베이스*를 참조하십시오.

백업 이미지가 위치할 목표 디렉토리 및 디바이스를 지정하기 위해 이 매개변수를 반복할 수 있습니다. 두 개 이상의 목표가 지정될 경우(예: target1, target2,

target3) target1이 첫 번째로 열려집니다. 미디어 헤더 및 특수 파일(구성 파일, 테이블 스페이스 테이블 및 실행기록 파일을 포함하여)은 target1에 넣어집니다. 백업 조작 중 나머지 모든 목표를 열고 병렬로 사용합니다. Windows 운영 체제에서는 일반 테이프가 지원되지 않으므로 각 유형의 테이프 디바이스에는 고유 디바이스 드라이버가 필요합니다.

테이프 디바이스 또는 플로피 디스크를 사용하면 사용자 입력에 대한 메시지 및 프롬프트가 생성될 수 있습니다. 유효한 응답은 다음과 같습니다.

- c** 계속. 경고 메시지를 생성한 디바이스를 계속 사용하십시오(예를 들어, 새 테이프가 마운트되었을 때).
- d** 디바이스 종료. 경고 메시지를 생성한 디바이스만 사용을 중지하십시오(예를 들어, 추가 테이프가 없을 때).
- t** 종료. 백업 조작을 중단하십시오.

테이프 시스템이 백업 이미지를 고유하게 참조할 수 있는 기능을 지원하지 않는 경우, 같은 데이터베이스의 여러 백업 사본을 동일한 테이프에 보존하지 않는 것이 바람직합니다.

#### **LOAD** *library-name*

사용될 벤더 백업 및 리스토어 I/O 기능이 들어 있는 공유 라이브러리 이름(Windows 운영 체제의 DLL). 여기에는 전체 경로가 포함될 수 있습니다. 전체 경로가 제공되지 않을 경우 디폴트 경로는 User Exit 프로그램이 있는 경로입니다.

#### **WITH** *num-buffers* **BUFFERS**

사용될 버퍼 수. DB2는 사용자가 명시적으로 값을 입력하는 경우를 제외하고는 자동으로 이 매개변수에 대한 최적 값을 선택합니다. 그러나 다중 위치로 백업을 생성할 때 성능을 개선하기 위해 더 많은 버퍼 수를 사용할 수 있습니다.

#### **BUFFER** *buffer-size*

백업 이미지를 빌드할 때 사용되는 버퍼의 크기(4KB 페이지 단위). DB2는 사용자가 명시적으로 값을 입력하는 경우를 제외하고는 자동으로 이 매개변수에 대한 최적 값을 선택합니다. 이 매개변수의 최소값은 8페이지입니다.

가변 블록 크기를 갖는 테이프를 사용할 경우, 버퍼 크기를 테이프 디바이스가 지원하는 범위 내로 줄이십시오. 그렇지 않으면, 백업 조작은 계속할 수 있지만 결과 이미지를 복구하지 못할 수 있습니다.

Linux의 대부분의 버전에서 SCSI 테이프 디바이스로 백업 조작에 DB2의 디폴트 버퍼 크기를 사용하면 SQL2025N 오류, 이유 코드 75가 리턴됩니다. Linux 내부 SCSI 버퍼가 오버플로우되지 않게 하려면 다음 공식을 사용하십시오.

$$\text{bufferpages} \leq \text{ST\_MAX\_BUFFERS} * \text{ST\_BUFFER\_BLOCKS} / 4$$

여기서 *bufferpages*는 BUFFER 매개변수에 대해 사용하려는 값이고 ST\_MAX\_BUFFERS 및 ST\_BUFFER\_BLOCKS는 drivers/scsi 디렉토리 아래의 Linux 커널에 정의됩니다.

#### **PARALLELISM** *n*

백업 유틸리티에서 병렬로 읽을 수 있는 테이블 스페이스 수를 결정합니다. DB2는 사용자가 명시적으로 값을 입력하는 경우를 제외하고는 자동으로 이 매개변수에 대한 최적 값을 선택합니다.

#### **UTIL\_IMPACT\_PRIORITY** *priority*

백업이 지정된 우선순위를 사용하여 조절 모드로 실행하도록 지정합니다. 조절을 사용하면 백업 조작의 성능 영향을 조정할 수 있습니다. 우선순위는 1 - 100의 숫자를 사용할 수 있는데, 1은 최저 우선순위를 나타내고 100은 최고 우선순위를 나타냅니다. UTIL\_IMPACT\_PRIORITY 키워드를 우선순위 없이 지정할 경우 백업은 디폴트 우선순위인 50으로 실행됩니다.

UTIL\_IMPACT\_PRIORITY를 지정하지 않으면 백업은 조절되지 않은 모드로 실행됩니다. 백업을 조절 모드로 실행되게 하려면 *util\_impact\_lim* 구성 매개변수를 설정하여 영향 규정을 정의해야 합니다.

#### **COMPRESS**

백업을 압축할 것을 표시합니다.

#### **COMPRLIB** *name*

압축을 수행하는 데 사용될 라이브러리 이름을 표시합니다(예: Windows의 경우 db2compr.dll, Linux/UNIX 시스템의 경우 libdb2compr.so). 이 이름은 서버에 있는 파일을 나타내는 완전한 경로이어야 합니다. 이 매개변수를 지정하지 않으면 디폴트 DB2 압축 라이브러리를 사용합니다. 지정된 라이브러리를 로드할 수 없으면 백업에 실패합니다.

#### **EXCLUDE**

압축 라이브러리가 백업 이미지에 저장되지 않을 것을 표시합니다.

#### **COMPROPTS** *string*

압축 라이브러리의 초기화 루틴으로 전달될 2진 데이터 블록을 설명합니다. DB2는 클라이언트에서 서버로 직접 이 문자열을 전달하므로 모든 바이트 리버설 또는 코드 페이지 변환 문제는 압축 라이브러리에서 처리되어야 합니다. 데이터 블록의 첫 번째 문자가 '@'일 경우, DB2는 이 데이터의 나머지 부분을 서버에 있는 파일 이름으로 해석합니다. 그런 다음 DB2는 문자열의 내용을 이 파일의 내용과 바꾼 후 대신 이 새 값을 초기화 루틴으로 전달합니다. *string*의 최대 길이는 1024바이트입니다.

## EXCLUDE LOGS

백업 이미지가 로그 파일을 포함해서는 안된다는 것을 지정합니다. 오프라인 백업 작업을 수행할 때, 스냅샷 백업 예외를 사용하여 이 옵션이 지정되었는지 여부에 관계없이 로그를 제외합니다.

## INCLUDE LOGS

백업 이미지가 이 이미지를 어떤 일관된 특정 시점으로 리스토어 및 롤 포워드하는 데 필요한 일련의 로그 파일을 반드시 포함하도록 지정합니다. 명시적으로 제외하도록 명시한 경우를 제외하고 이 옵션이 디폴트인 경우, 스냅샷 백업의 예외가 있는 오프라인 백업에는 이 옵션이 유효하지 않습니다.

## WITHOUT PROMPTING

백업이 자동으로 실행되고 일반적으로 사용자 개입이 필요한 모든 조치는 오류 메시지를 리턴할 것을 지정합니다.

주:

1. 파티션된 데이터베이스에서 어떤 파티션이 백업되는지 백업 명령에서 표시하는 경우, 프롬프트하지 않고 내재적으로 백업 작업이 수행됩니다.
2. 파티션된 데이터베이스 환경에서 BACKUP DATABASE 명령에 지정된 옵션은 개별적으로 각 파티션에서 적용됩니다. 예를 들어, 백업 작업이 USE TSM OPEN 3 SESSIONS로 지정되면, DB2는 각 파티션에서 세 개의 TSM 세션을 엽니다.

예:

1. 다음 예에서 데이터베이스 WSDb는 번호가 0부터 3까지인 모두 네 개의 데이터베이스 파티션에 정의됩니다. 경로 /dev3/backup은 모든 데이터베이스 파티션에서 액세스할 수 있습니다. 데이터베이스 파티션 0은 카탈로그 파티션입니다. 모든 WSDb 데이터베이스 파티션을 /dev3/backup으로 오프라인 백업을 수행하려면 데이터베이스 파티션 0에서 다음 명령을 발행하십시오.

```
db2 BACKUP DATABASE wsdb ON ALL DBPARTITIONNUMS TO /dev3/backup
```

모든 파티션에서 동시에 백업이 수행됩니다. 네 개의 데이터베이스 파티션 백업 이미지가 모두 /dev3/backup 디렉토리에 저장되며, 이 디렉토리는 둘 이상의 파티션에서 액세스할 수 있는 공유 디렉토리이거나, 각 파티션에서 개별적으로 액세스할 수 있는 로컬로 마운트되는 디렉토리이거나, 양쪽을 조합한 것일 수 있습니다.

2. 다음 예에서 데이터베이스 SAMPLE은 두 개의 동시 TSM 클라이언트 세션을 사용하여 TSM 서버로 백업됩니다. DB2는 이 환경에 대한 최적 버퍼 크기를 계산합니다.

```
db2 backup database sample use tsm open 2 sessions with 4 buffers
```

3. 다음 예에서, 데이터베이스 payro11의 테이블 스페이스(syscatspace, userspace1)에 대한 테이블 스페이스 레벨 백업이 테이프에 수행됩니다.

```
db2 backup database payroll tablespace (syscatspace, userspace1) to
/dev/rmt0, /dev/rmt1 with 8 buffers without prompting
```

4. 백업 조작에 사용할 TSM 정보를 지정하기 위해 USE TSM OPTIONS 키워드를 사용할 수 있습니다. 다음 예는 USE TSM OPTIONS 키워드를 사용하여 완전한 파일 이름을 지정하는 방법을 보여 줍니다.

```
db2 backup db sample use TSM options @/u/dmcinnis/myoptions.txt
```

myoptions.txt 파일에는 다음 정보가 들어 있습니다. -fromnode=bar  
-fromowner=dmcinnis

5. 다음은 복구 가능한 데이터베이스의 주별 증분 백업 전략 샘플입니다. 주별 전체 데이터베이스 백업 조작과 일별 비누적(델타) 백업 조작 및 주중 누적(증분) 백업 조작이 포함됩니다.

```
(Sun) db2 backup db sample use tsm
(Mon) db2 backup db sample online incremental delta use tsm
(Tue) db2 backup db sample online incremental delta use tsm
(Wed) db2 backup db sample online incremental use tsm
(Thu) db2 backup db sample online incremental delta use tsm
(Fri) db2 backup db sample online incremental delta use tsm
(Sat) db2 backup db sample online incremental use tsm
```

6. 다음 예에서는 데이터베이스 SAMPLE에 대한 백업 조작에 세 개의 동일한 목표 디렉토리가 지정됩니다. 목표 파일 시스템이 여러 실제 디스크로 구성된 경우 이를 수행하려 할 수 있습니다.

```
db2 backup database sample to /dev3/backup, /dev3/backup, /dev3/backup
```

데이터는 세 개의 목표 디렉토리에 동시 백업되고, 세 개의 백업 이미지는 .001, .002, .003과 같은 확장자를 사용하여 생성됩니다.

7. 다음 예에서 데이터베이스 WSDb는 번호가 0부터 3까지인 모두 네 개의 데이터베이스 파티션에 정의됩니다. 데이터베이스 파티션 0은 카탈로그 파티션입니다. 데이터베이스 파티션 1과 2에서 테이블 스페이스 USERSPACE1의 온라인 백업을 수행하고, 백업 이미지를 TSM 서버에 저장하려면, 파티션 0에서 다음 명령을 발  
행하십시오.

```
db2 BACKUP DATABASE wsdb ON DBPARTITIONNUMS (1, 2) TABLESPACE (USERSPACE1)
ONLINE USE TSM
```

8. 각 파티션에 의해 리턴된 sqlcode를 표시하기 위해 생성된 샘플 출력.

#### 예 1 모든 파티션 성공(sqlcode >= 0)

```
$ db2 backup db foo on all dbpartitionnums tablespace(T1)
Part Result
-----
0 DB20000I The BACKUP DATABASE command completed successfully.
1 SQL2430W The database backup succeeded, but the following
table spaces do not exist on this database partition: "T1".
```

Backup successful. 이 백업 이미지에 대한 시간소인은 20040908010203입니다.

#### 예 2: 하나 이상의 파티션 실패(sqlcode < 0)

```

$ db2 backup db foo on all dbpartitionnums to /backups
Part Result
-----
0 DB20000I The BACKUP DATABASE command completed successfully.
1 SQL2419N The target disk "/backups" has become full.

SQL2429N The database backup failed. The following database
partitions returned errors: "1".

```

9. 다음 백업은 작성된 이미지에 로그 디렉토리를 포함합니다.

```

db2 backup db sample use snapshot
db2 backup db sample online use snapshot
db2 backup db sample use snapshot INCLUDE LOGS
db2 backup db sample online use snapshot INCLUDE LOGS

```

10. 다음 백업은 작성된 이미지에 로그 디렉토리를 포함하지 않습니다.

```

db2 backup db sample use snapshot EXCLUDE LOGS
db2 backup db sample online use snapshot EXCLUDE LOGS

```

## 사용 시 참고사항

백업의 데이터는 데이터베이스 서버의 보호를 받을 수 없습니다. 특히 백업에 LBAC 보호 데이터가 포함된 경우 백업이 올바르게 보호되는지 확인하십시오.

테이프에 백업시 가변 블록 크기의 사용은 현재 지원되지 않습니다. 이 옵션을 사용해야 하는 경우, 가변 블록 크기로 작성된 백업 이미지를 사용하여 성공적으로 복구할 수 있는 제대로 테스트된 프로시저가 있는지 확인하십시오.

가변 블록 크기 사용시 사용 중인 테이프 디바이스에 대해 최대 한계 이하의 백업 버퍼 크기를 지정해야 합니다. 최적 성능을 위해서는 버퍼 크기를 사용 중인 디바이스의 최대 블록 크기 한계와 동일하게 지정해야 합니다.

파일러/스토리지 시스템에서 실패하는 경우 스냅샷 백업은 일반 디스크 백업으로 보완되어야 합니다.

정기적으로 데이터베이스를 백업할 때, 매우 큰 데이터베이스 백업 이미지, 여러 데이터베이스 로그 및 로드 사본 이미지를 축적할 수 있으며, 이는 모두 많은 디스크 스페이스를 차지할 수 있습니다. 이러한 복구 오버젝트를 관리하는 방법에 대한 정보는 『복구 오버젝트 관리』를 참조하십시오.

---

## 제 16 장 BIND

프리컴파일러에서 생성된 바인드 파일에 저장된 SQL 문을 준비하고 데이터베이스에 저장되는 패키지를 작성하는 바인드 유틸리티를 호출합니다.

### 범위

이 명령은 db2nodes.cfg에서 모든 데이터베이스 파티션에서 발행될 수 있습니다. 카탈로그 데이터베이스 파티션에서 데이터베이스 카탈로그를 갱신합니다. 모든 데이터베이스 파티션에 갱신사항이 적용됩니다.

### 권한 부여

다음 권한 부여 중 하나여야 합니다.

- *dbadm* 권한
- SQLERROR CHECK 또는 EXPLAIN ONLY가 지정된 경우, EXPLAIN 또는 SQLADM 권한 중 하나면 충분합니다.
- 패키지가 없는 경우, BINDADD 권한과 다음 권한이 필요합니다.
  - 패키지의 스키마 이름이 없는 경우, 데이터베이스에 대한 IMPLICIT\_SCHEMA 권한
  - 패키지의 스키마 이름이 없는 경우, 스키마에 대한 CREATEIN 특권
- 패키지가 존재하는 경우, 다음 특권 중 하나여야 합니다.
  - 스키마에 대한 ALTERIN 특권
  - 패키지에 대한 BIND 특권

또한, 사용자는 응용프로그램에서 정적 SQL문을 컴파일하는 데 필요한 모든 특권이 필요합니다. 그룹에 부여된 특권은 정적 명령문의 권한 부여 검사에는 사용되지 않습니다.

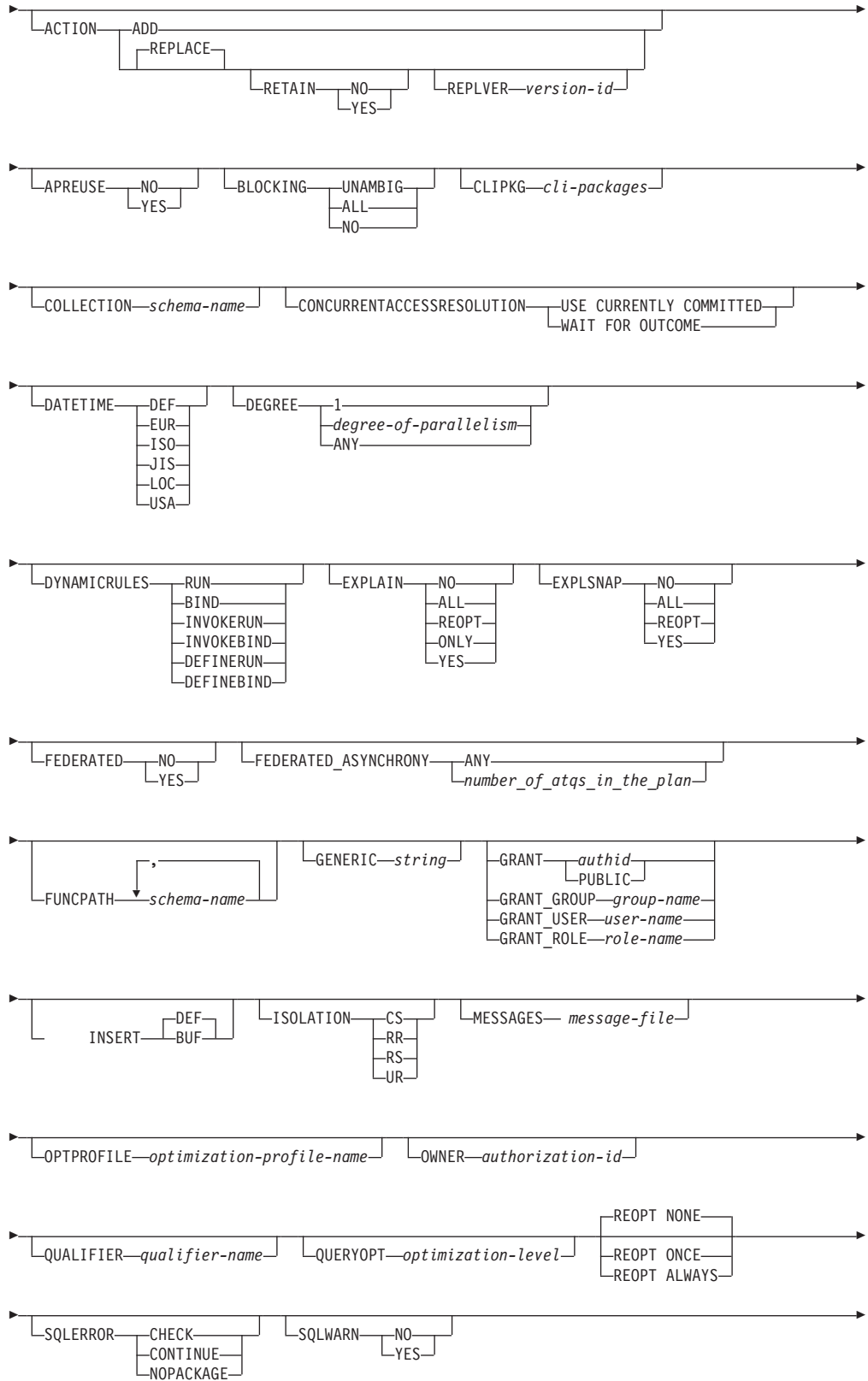
### 필수 연결

데이터베이스 내재된 연결이 사용 가능한 경우 디폴트 데이터베이스에 연결됩니다.

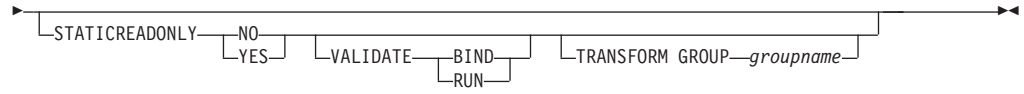
### 명령 구문

#### Linux, Windows 및 UNIX용 DB2

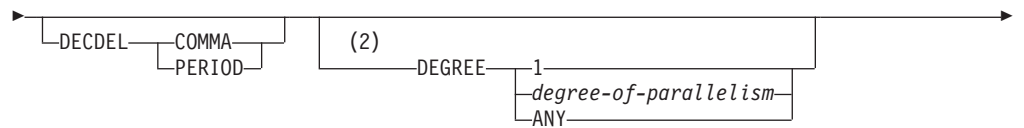
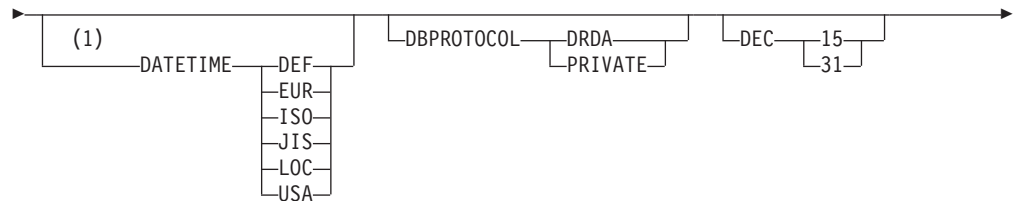
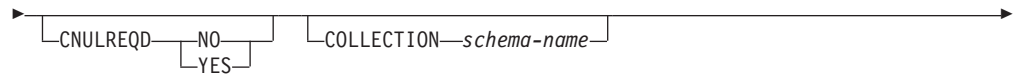
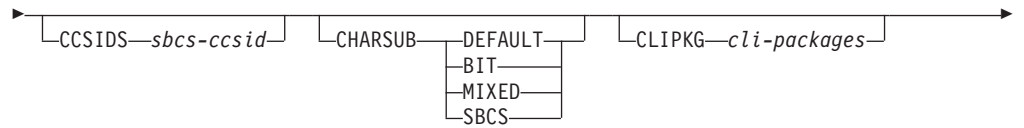
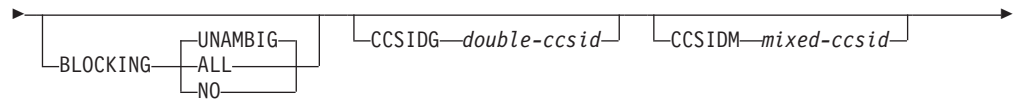
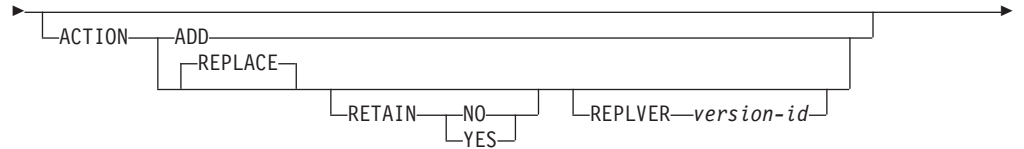
▶▶—BIND—*filename*—————▶▶

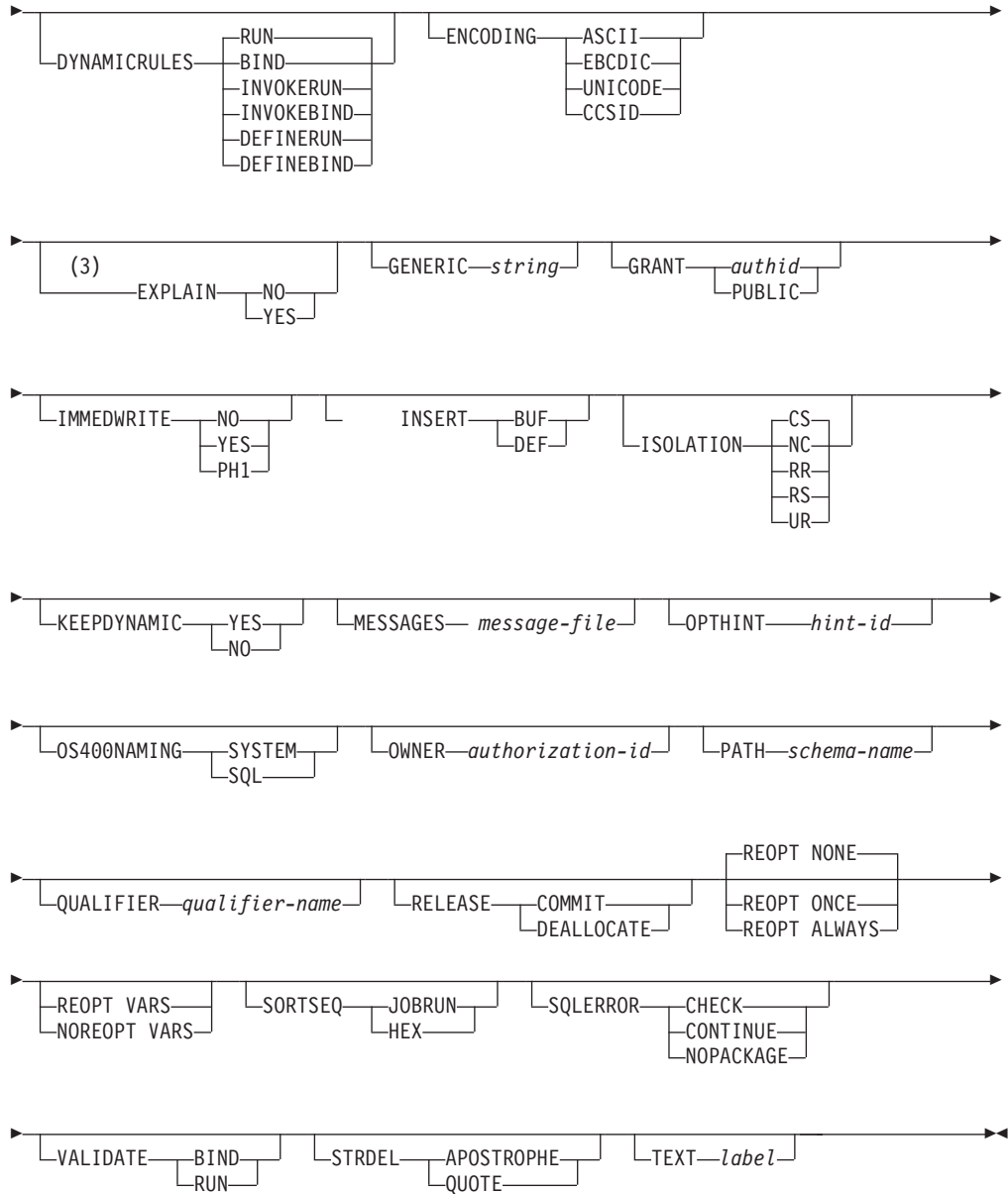






**Linux, Windows 및 UNIX가 아닌 다른 서버의 DB2**





**주:**

- 1 서버가 DATETIME DEF 옵션을 지원하지 않는 경우, DATETIME ISO로 맵핑됩니다.
- 2 DEGREE 옵션은 DRDA® 레벨 2 응용프로그램 서버에서만 지원됩니다.
- 3 DRDA는 YES 또는 NO 값을 갖도록 EXPLAIN 옵션을 정의합니다. 서버가 EXPLAIN YES 옵션을 지원하지 않으면 값은 EXPLAIN ALL에 맵핑됩니다.

**명령 매개변수**

*filename*

응용프로그램이 프리컴파일될 때 생성된 바인드 파일의 이름이나, 몇몇 바인드

파일의 이름을 포함하는 목록 파일을 지정합니다. 바인드 파일의 확장자는 .bnd  
입니다. 전체 경로 이름을 지정할 수 있습니다.

목록 파일을 지정한 경우 목록 파일 이름의 첫 번째 문자는 @ 문자여야 합니  
다. 목록 파일은 몇 개의 바인드 파일 이름 행을 포함할 수 있습니다. 동일한  
행에 나열된 바인드 파일은 더하기(+) 문자로 구분해야 하지만 각 행에 나열된  
첫 번째 파일의 앞이나 나열된 마지막 바인드 파일 뒤에는 +가 있으면 안됩니  
다. 예를 들어, 다음과 같습니다.

```
/u/smith/sql1lib/bnd/@all.lst
```

이 예는 다음 바인드 파일을 포함하는 목록 파일입니다.

```
mybind1.bnd+mybind.bnd2+mybind3.bnd+  
mybind4.bnd+mybind5.bnd+  
mybind6.bnd+  
mybind7.bnd
```

## ACTION

패키지가 추가 또는 교체될 수 있는지 여부를 표시합니다.

**ADD** 이름 지정된 패키지가 없으며, 새 패키지가 작성됨을 표시합니다. 패키  
지가 이미 존재하는 경우, 실행이 중지하며 진단 오류 메시지가 리턴됩  
니다.

## REPLACE

기존 패키지는 동일한 패키지 이름 및 작성자가 있는 새 패키지로 교  
체됨을 표시합니다. 이것은 ACTION 옵션의 디폴트값입니다.

## RETAIN

패키지가 교체될 때 BIND 및 EXECUTE 권한이 유지되는지  
여부를 표시합니다. 패키지 변경의 소유권이 변경되면, 새 소유  
자는 BIND 및 EXECUTE 권한을 이전 패키지 소유자에게 부  
여합니다.

**NO** 패키지가 교체될 때 BIND 및 EXECUTE 권한을 유  
지하지 않습니다. 이 값은 DB2에서 지원되지 않습니다.

**YES** 패키지가 교체될 때 BIND 및 EXECUTE 권한을 유  
지합니다. 이것은 디폴트값입니다.

## REPLVER *version-id*

패키지의 특정 버전을 교체합니다. 버전 ID는 어떤 패키지 버  
전이 교체되는지 지정합니다. 지정된 버전이 없으면, 오류가 리  
턴됩니다. REPLACE의 REPLVER 옵션이 지정되지 않고 바  
인드 중인 패키지의 패키지 이름, 작성자 및 버전이 일치하는

패키지가 이미 존재하는 경우 해당 패키지는 교체되며, 그렇지 않은 경우 새 패키지가 추가됩니다.

### APREUSE

정적 SQL 액세스 플랜이 재사용되는지 여부를 지정합니다. 이 옵션이 사용 가능하면, 쿼리 컴파일러는 바인드 동안 그리고 향후 내재된 리바인드와 명시적 리바인드 동안 기존 패키지에서 명령문의 액세스 플랜을 재사용하려고 시도합니다.

**YES** 쿼리 컴파일러가 패키지에서 명령문의 액세스 플랜을 재사용하려고 시도합니다. 기존 패키지가 있는 경우, 쿼리 컴파일러는 새 바인드 파일에서 명령문과 일치하는 모든 명령문의 액세스 플랜을 재사용하도록 시도합니다. 일치하는 명령문의 경우, 명령문 텍스트가 동일해야 하며 기존 패키지의 명령문에 대한 섹션 번호는 새 패키지의 명령문에 대한 섹션 번호와 일치해야 합니다.

**NO** 쿼리 컴파일러가 패키지에서 명령문의 액세스 플랜을 재사용하려고 시도하지 않습니다. 이는 디폴트 설정입니다.

### BLOCKING

커서에 대해 행 블로킹 유형을 지정합니다. LOB 컬럼 데이터 유형에 대한 참조를 포함하는 행 데이터의 블로킹은 Database Partitioning Feature(DPF)가 사용 가능한 환경에서도 지원됩니다.

**ALL FOR READ ONLY** 절로 지정된 커서나 **FOR UPDATE**로 지정되지 않은 커서의 경우, 블로킹이 발생합니다.

앰비규어스 커서는 읽기 전용입니다.

**NO** 임의 커서에 대한 블로킹은 발생하지 않습니다.

읽기 전용 커서 및 앰비규어스 커서의 정의는 *DECLARE CURSOR* 문을 참조하십시오.

앰비규어스 커서는 갱신 가능합니다.

### UNAMBIG

**FOR READ ONLY** 절로 지정된 커서의 경우, 블로킹이 발생합니다.

앰비규어스가 아니고 읽기 전용인 **FOR READ ONLY** 또는 **FOR UPDATE** 절로 선언되지 않은 커서는 차단됩니다. 앰비규어스 커서는 차단되지 않습니다.

앰비규어스 커서는 갱신 가능합니다.

### CCSIDG *double-ccsid*

**CREATE** 및 **ALTER TABLE SQL**문의 문자 컬럼 정의에서(특정 **CCSID** 절 없이) 2바이트 문자에 사용되는 코드화된 문자 세트 ID(**CCSID**)를 지정하는 정

수. 이 옵션은 Linux, UNIX 및 Windows용 DB2 데이터베이스에서 지원되지 않습니다. 이 옵션이 지정되지 않은 경우 DRDA 서버는 시스템 정의 디폴트값을 사용합니다.

#### **CCSIDM** *mixed-ccsid*

CREATE 및 ALTER TABLE SQL문의 문자 컬럼 정의에서(특정 CCSID 절 없이) 혼합 바이트 문자에 사용되는 코드화된 문자 세트 ID(CCSID)를 지정하는 정수. 이 옵션은 Linux, UNIX 및 Windows용 DB2 데이터베이스에서 지원되지 않습니다. 이 옵션이 지정되지 않은 경우 DRDA 서버는 시스템 정의 디폴트값을 사용합니다.

#### **CCSIDS** *sbc-ccsid*

CREATE 및 ALTER TABLE SQL문의 문자 컬럼 정의에서(특정 CCSID 절 없이) 1바이트 문자에 사용되는 코드화된 문자 세트 ID(CCSID)를 지정하는 정수. 이 옵션은 Linux, UNIX 및 Windows용 DB2 데이터베이스에서 지원되지 않습니다. 이 옵션이 지정되지 않은 경우 DRDA 서버는 시스템 정의 디폴트값을 사용합니다.

#### **CHARSUB**

CREATE 및 ALTER TABLE SQL문에서 컬럼 정의에 사용되는 디폴트 문자 부속 유형을 지정합니다. 이 DRDA 프리컴파일/바인드 옵션은 Linux, UNIX 및 Windows용 DB2 데이터베이스에서 지원되지 않습니다.

**BIT** 명시적 부속 유형이 지정되지 않은 모든 새 문자 컬럼에서 FOR BIT DATA SQL 문자 부속 유형을 사용합니다.

#### **DEFAULT**

명시적 부속 유형이 지정되지 않은 모든 새 문자 컬럼에서 목표 시스템 정의된 디폴트를 사용합니다.

#### **MIXED**

명시적 부속 유형이 지정되지 않은 모든 새 문자 컬럼에서 FOR MIXED DATA SQL 문자 부속 유형을 사용합니다.

**SBCS** 명시적 부속 유형이 지정되지 않은 모든 새 문자 컬럼에서 FOR SBCS DATA SQL 문자 부속 유형을 사용합니다.

#### **CLIPKG** *cli-packages*

데이터베이스에 대해 CLI 바인드 파일을 바인드할 때 작성할 CLI 대형 패키지 수를 지정하는 3 및 30 사이의 정수입니다.

#### **CNULREQD**

이 옵션은 DRDA에서 지원되지 않는 LANGLEVEL 프리컴파일 옵션과 관련됩니다. 바인드 파일이 C 또는 C++ 응용프로그램에서 작성된 경우에만 유효합니다. 이 DRDA 바인드 옵션은 Linux, UNIX 및 Windows용 DB2 데이터베이스에서 지원되지 않습니다.

- NO** 응용프로그램이 C 문자열 호스트 변수에서 널(NULL) 종료자에 대해 LANGLEVEL SAA1 프리컴파일 옵션에 기준하여 코딩되었습니다.
- YES** 응용프로그램이 C 문자열 호스트 변수에서 널(NULL) 종료자에 대해 LANGLEVEL MIA 프리컴파일 옵션에 기준하여 코딩되었습니다.

**COLLECTION** *schema-name*

패키지에 대해서 128바이트 컬렉션 ID를 지정합니다. 지정되지 않은 경우, 패키지 처리하는 사용자의 권한 부여 ID가 사용됩니다.

**CONCURRENTACCESSRESOLUTION**

패키지에서 명령문에 사용할 동시 액세스 분석을 지정합니다.

**USE CURRENTLY COMMITTED**

데이터베이스 관리 프로그램이 갱신 또는 삭제 중인 프로세스에 있을 때 데이터베이스 관리 프로그램이 적용 가능한 스캔을 위해 현재 커미트된 데이터 버전을 사용할 수 있음을 지정합니다. 삽입 중인 프로세스의 행은 건너뛴 수 있습니다. 유효한 분리 레벨이 커서 안정성 또는 읽기 안정성(읽기 안정성의 경우 커미트되지 않은 삽입만 건너뛴)이며 그렇지 않은 경우 무시되는 경우 이 절이 적용됩니다. 적용 가능한 스캔은 읽기 전용이 아닌 명령문과 읽기 전용 명령문의 부분인 읽기 전용 스캔을 포함합니다. 레지스트리 변수 **DB2\_EVALUNCOMMITTED**, **DB2\_SKIPDELETED** 및 **DB2\_SKIPINSERTED**의 설정은 더 이상 적용되지 않습니다.

**WAIT FOR OUTCOME**

갱신 중인 프로세스에서 데이터 발생시 커미트 또는 롤백을 대기하기 위해 커서 안정성 및 상위 스캔을 지정합니다. 삽입 또는 삭제 중인 프로세스의 행은 건너뛴 수 없습니다. 레지스트리 변수 **DB2\_EVALUNCOMMITTED**, **DB2\_SKIPDELETED** 및 **DB2\_SKIPINSERTED**의 설정은 더 이상 적용되지 않습니다.

**DATETIME**

사용되는 날짜 및 시간 형식을 지정합니다.

- DEF** 데이터베이스의 지역 코드와 연관된 날짜 및 시간 형식을 사용합니다.
- EUR** 유럽 날짜 및 시간 형식의 IBM 표준을 사용합니다.
- ISO** 국제 표준 조직의 날짜 및 시간 형식을 사용합니다.
- JIS** 일본 산업 표준의 날짜 및 시간 형식을 사용합니다.
- LOC** 데이터베이스의 지역 코드와 연관된 로컬 양식의 날짜 및 시간 형식을 사용합니다.
- USA** 미국 날짜 및 시간 형식의 IBM 표준을 사용합니다.

## DBPROTOCOL

세 파트 이름 명령문으로 식별되는 리모트 사이트에 연결시 사용할 프로토콜을 지정합니다. OS/390® 전용 DB2에 의해 지원됩니다. 지원되는 옵션 값 목록은 OS/390용 DB2 문서를 참조하십시오.

**DEC** 10진수 산술 연산에서 사용되는 최대 정밀도를 지정합니다. 이 DRDA 프리컴 파일/바인드 옵션은 Linux, UNIX 및 Windows용 DB2 데이터베이스에서 지원되지 않습니다. 이 옵션이 지정되지 않은 경우 DRDA 서버는 시스템 정의 디폴트값을 사용합니다.

**15** 15자리 정밀도가 10진수 산술 연산에서 사용됩니다.

**31** 31자리 정밀도가 10진수 산술 연산에서 사용됩니다.

## DECDEL

마침표(.) 또는 쉼표(,)가 10진수 및 부동 소수점 리터럴에서 소수점 표시기로 사용되는지 여부를 지정합니다. 이 DRDA 프리컴파일/바인드 옵션은 Linux, UNIX 및 Windows용 DB2 데이터베이스에서 지원되지 않습니다. 이 옵션이 지정되지 않은 경우 DRDA 서버는 시스템 정의 디폴트값을 사용합니다.

## COMMA

소수점 표시기로서 쉼표(,)를 사용합니다.

## PERIOD

소수점 표시기로서 마침표(.)를 사용합니다.

## DEGREE

SMP 시스템에서 정적 SQL문의 실행을 위한 병렬 처리 수준을 지정합니다. 이 옵션은 CREATE INDEX 병렬 처리에 영향을 미치지 않습니다.

**1** 명령문의 실행은 병렬 처리를 사용하지 않습니다.

### *degree-of-parallelism*

명령문이 실행될 수 있는 병렬 처리 수준의 등급을 지정합니다(2 - 32767의 값).

**ANY** 데이터베이스 관리 프로그램에 의해 판별된 등급을 사용하여 명령문 실행이 병렬 처리를 포함할 수 있는지 지정합니다.

## DYNAMICRULES

권한 부여 ID 및 규정되지 않은 오브젝트 참조의 내재된 자격에 대해 사용되는 값의 초기 설정에 대해 런타임에 어떤 규칙이 동적 SQL에 적용되는지 정의합니다.

**RUN** 패키지를 실행하는 사용자의 권한 부여 ID가 동적 SQL문의 권한 부여 검사에 사용됨을 지정합니다. 또한, 권한 부여 ID는 동적 SQL문 내 규정되지 않은 오브젝트 참조의 내재된 자격의 디폴트 패키지 규정자로서 권한 부여 ID가 사용됩니다. 이것은 디폴트값입니다.

**BIND** 권한 부여 및 자격에 대해 정적 SQL에 적용하는 모든 규칙이 런타임에 사용됨을 지정합니다. 즉, 패키지 소유자의 권한 부여 ID는 동적 SQL문의 권한 부여 검사에 사용되며 디폴트 패키지 규정자가 동적 SQL문 내 규정되지 않은 오브젝트 참조의 내재된 자격에는 디폴트 패키지 규정자가 사용됩니다.

#### **DEFINERUN**

패키지가 루틴 컨텍스트 내에서 사용되면, 루틴 정의자의 권한 부여 ID는 권한 부여 검사와 루틴 내 동적 SQL문 내에서 규정되지 않은 오브젝트 참조의 내재된 자격에 사용됩니다.

패키지가 독립형 응용프로그램으로 사용되면, 동적 SQL문은 패키지가 DYNAMICRULES RUN으로 바인드된 것처럼 처리됩니다.

#### **DEFINEBIND**

패키지가 루틴 컨텍스트 내에서 사용되면, 루틴 정의자의 권한 부여 ID는 권한 부여 검사와 루틴 내 동적 SQL문 내에서 규정되지 않은 오브젝트 참조의 내재된 자격에 사용됩니다.

패키지가 독립형 응용프로그램으로 사용되면, 동적 SQL문은 패키지가 DYNAMICRULES BIND로 바인드된 것처럼 처리됩니다.

#### **INVOKERUN**

패키지가 루틴 컨텍스트 내에서 사용되면, 루틴이 호출되었을 때 유효한 현재 명령문 권한 부여 ID가 동적 SQL문의 권한 부여 검사와 해당 루틴 내 동적 SQL문 내에서 규정되지 않은 오브젝트 참조의 내재된 자격에 사용됩니다.

패키지가 독립형 응용프로그램으로 사용되면, 동적 SQL문은 패키지가 DYNAMICRULES RUN으로 바인드된 것처럼 처리됩니다.

#### **INVOKEBIND**

패키지가 루틴 컨텍스트 내에서 사용되면, 루틴이 호출되었을 때 유효한 현재 명령문 권한 부여 ID가 동적 SQL문의 권한 부여 검사와 해당 루틴 내 동적 SQL문 내에서 규정되지 않은 오브젝트 참조의 내재된 자격에 사용됩니다.

패키지가 독립형 응용프로그램으로 사용되면, 동적 SQL문은 패키지가 DYNAMICRULES BIND로 바인드된 것처럼 처리됩니다.

동적 SQL문은 바인드 동작을 나타내는 패키지에서 패키지 소유자의 권한 부여 ID를 사용하므로, 패키지의 사용자가 받아서는 안되는 권한을 패키지 바인더에 부여하지 않아야 합니다. 마찬가지로, 동적 명령문은 루틴 정의자의 권한 부여 ID를 사용하므로 정의 동작을 나타내는 루틴을 정의할 때, 루틴의 정의자에게 패키지 사용자가 받아서는 안되는 권한을 부여해서는 안됩니다.



다음과 같은 동적 Prepared SQL문은 DYNAMICRULES RUN: GRANT, REVOKE, ALTER, CREATE, DROP, COMMENT ON, RENAME, SET INTEGRITY 및 SET EVENT MONITOR STATE로 바인드되지 않은 패키지 내에서 사용될 수 없습니다.

## ENCODING

플랜 또는 패키지에서 정적 명령문의 모든 호스트 변수에 대한 인코딩을 지정합니다. OS/390 전용 DB2에 의해 지원됩니다. 지원되는 옵션 값 목록은 OS/390용 DB2 문서를 참조하십시오.

## EXPLAIN

패키지의 각 SQL문에 선택된 액세스 플랜에 대한 Explain 테이블에 정보를 저장합니다. DRDA는 이 옵션의 ALL 값을 지원하지 않습니다.

**NO** Explain 정보는 캡처되지 않습니다.

**YES** Explain 테이블에는 정적 명령문의 경우 prep/바인드 시간에 그리고 증분식 바인드 명령문의 경우 런타임에 선택된 액세스 플랜에 대한 정보가 채워집니다.

패키지가 루틴에 사용되며 패키지가 증분식 바인드 명령문을 포함하는 경우, 루틴은 MODIFIES SQL DATA로 정의되어야 합니다. 그렇지 않으면, 패키지의 증분식 바인드 명령문이 런타임 오류(SQLSTATE 42985)를 유발합니다.

## REOPT

재최적화가 가능한 각 증분식 바인드 SQL문의 Explain 정보는 런타임 시 Explain 테이블에 놓입니다. 또한, CURRENT EXPLAIN MODE 레지스터가 NO로 설정되어도 런타임 시 재최적화가 가능한 동적 SQL문에 대한 Explain 정보가 수집됩니다.

패키지가 루틴에 사용되면 루틴이 MODIFIES SQL DATA로 정의되어야 하며, 그렇지 않으면 패키지의 증분식 바인드 및 동적문이 런타임 오류(SQLSTATE 42985)를 유발합니다.

## ONLY

ONLY 옵션은 특권 없이 Explain문을 실행하게 합니다. Explain 테이블이 채워지지만 지속적 패키지는 작성되지 않습니다. 바인드 프로세스 동안 동일한 이름과 버전을 가진 기존 패키지가 있는 경우, ACTION REPLACE를 지정해도 기존 패키지는 삭제되거나 교체되지 않습니다. Explain 테이블을 채우는 동안 오류가 발생하는 경우, 오류를 리턴한 명령문이나 다음에 오는 명령문에 대한 explain 정보는 추가되지 않습니다.

**ALL** 각 적격 정적 SQL문의 Explain 정보는 prep/바인드 시간에 Explain 테이블에 위치됩니다. 적합한 각 증분식 바인드 SQL문의 Explain 정

보는 런타임에 Explain 테이블에 위치됩니다. 또한, CURRENT EXPLAIN MODE 레지스터가 NO로 설정되는 경우에도 런타임 시 적격의 동적 SQL문에 대한 Explain 정보가 수집됩니다.

패키지가 루틴에 사용되면 루틴이 MODIFIES SQL DATA로 정의되어야 하며, 그렇지 않으면 패키지의 증분식 바인드 및 동적문이 런타임 오류(SQLSTATE 42985)를 유발합니다. EXPLAIN에 대한 이 값은 DRDA에서 지원되지 않습니다.

## EXPLSNAP

Explain 스냅샷 정보를 Explain 테이블에 저장합니다. 이 DB2 프리컴파일/바인드 옵션은 DRDA에서 지원되지 않습니다.

**NO** Explain 스냅샷은 캡처되지 않습니다.

**YES** 각 적격 정적 SQL문의 Explain 정보는 정적 명령문의 경우 prep/바인드 시간에 그리고 증분식 바인드 명령문의 경우 런타임에 Explain 테이블에 위치됩니다.

패키지가 루틴에 사용되고 패키지가 증분식 바인드 명령문을 포함하는 경우, 루틴이 MODIFIES SQL DATA로 정의되어야 하며, 그렇지 않으면 패키지의 증분식 바인드 명령문이 런타임 오류(SQLSTATE 42985)를 유발합니다.

## REOPT

재최적화가 가능한 각 증분식 바인드 SQL문의 Explain 스냅샷 정보는 런타임 시 Explain 테이블에 놓입니다. 또한, CURRENT EXPLAIN SNAPSHOT 레지스터가 NO로 설정되어도 런타임 시 재최적화가 가능한 동적 SQL문에 대한 Explain 스냅샷 정보가 수집됩니다.

패키지가 루틴에 사용되면 루틴이 MODIFIES SQL DATA로 정의되어야 하며, 그렇지 않으면 패키지의 증분식 바인드 및 동적문이 런타임 오류(SQLSTATE 42985)를 유발합니다.

**ALL** 각 적격 정적 SQL문의 Explain 스냅샷 정보는 prep/바인드 시간에 Explain 테이블에 위치됩니다. 각 적격 증분식 바인드 SQL문의 Explain 스냅샷 정보는 런타임 시 Explain 테이블에 놓입니다. 또한, CURRENT EXPLAIN SNAPSHOT 레지스터가 NO로 설정되는 경우에도 런타임 시 적격의 동적 SQL문에 대한 Explain 스냅샷 정보가 수집됩니다.

패키지가 루틴에 사용되면 루틴이 MODIFIES SQL DATA로 정의되어야 하며, 그렇지 않으면 패키지의 증분식 바인드 및 동적 명령문이 런타임 오류(SQLSTATE 42985)를 유발합니다.

## FEDERATED

패키지의 정적 SQL문이 별칭 또는 페더레이티드 뷰를 참조하는지 여부를 지정

합니다. 이 옵션이 지정되지 않고 패키지의 정적 SQL문이 별칭이나 페더레이티드 뷰를 참조하는 경우, 경고가 리턴되며 패키지가 작성됩니다. 이 옵션은 DRDA에 대해 지원되지 않습니다.

**NO** 별칭이나 페더레이티드 뷰는 패키지의 정적 SQL문에서 참조되지 않습니다. 이 패키지의 준비 또는 바인드 단계 동안 정적 SQL문에서 별칭이나 페더레이티드 뷰를 직면하는 경우, 오류가 리턴되며 패키지는 작성되지 않습니다.

**YES** 별칭이나 페더레이티드 뷰는 패키지의 정적 SQL문에서 참조될 수 있습니다. 패키지의 준비 또는 바인드 동안 정적 SQL문에서 별칭이나 페더레이티드 뷰를 직면하지 않는 경우, 오류나 경고가 리턴되지 않으며 패키지가 작성됩니다.

### **FEDERATED\_ASYNCRONY**

Embedded SQL을 사용하는 프로그램의 액세스 플랜에서 페더레이티드 서버가 지원하는 비동기 테이블 큐(ATQ)의 최대 수를 지정합니다.

**ANY** 옵티마이저는 액세스 플랜의 ATQ 수를 판별합니다. 옵티마이저는 플랜에서 모든 적격 SHIP이나 리모트 푸시다운 운영자에게 ATQ를 지정합니다. DB2\_MAX\_ASYNC\_REQUESTS\_PER\_QUERY 서버 옵션에 지정된 값은 비동기 요청 수를 제한합니다.

*number\_of\_atqs\_in\_the\_plan*

플랜에서의 ATQ 수. 0 - 32767 범위의 수를 지정합니다.

### **FUNCPATH**

사용자 정의 구별 유형에서 사용되는 함수 경로와 정적 SQL의 함수를 지정합니다. 이 옵션이 지정되지 않으면, 디폴트 함수 경로는 "SYSIBM","SYSFUN",USER이며 여기서 USER는 USER 특수 레지스터의 값입니다. 이 DB2 프리컴파일/바인드 옵션은 DRDA에서 지원되지 않습니다.

*schema-name*

응용프로그램 서버(AS)에 존재하는 스키마를 식별하는 SQL ID로, 일반 또는 구분된 유형이 있습니다. 프리컴파일 또는 바인드 시간에는 스키마가 존재하는지 확인되지 않습니다. 동일한 스키마는 함수 경로에 두 번 이상 나타날 수 없습니다. 함수 경로에 스키마 이름 SYSPUBLIC을 지정할 수 없습니다. 지정될 수 있는 스키마 수는 결과 함수 경로의 길이로 제한되며, 2048바이트를 초과할 수 없습니다. 스키마 SYSIBM은 명시적으로 지정될 필요가 없으며, 함수 경로에 포함되지 않는 경우 내재적으로 최초 스키마로 간주됩니다.

### **GENERIC string**

목표 데이터베이스에 정의된 새 옵션의 바인딩을 지원하지만, DRDA에 의해서 지원되지 않습니다. BIND 또는 PRECOMPILED에 정의된 바인드 옵션을

패스하려면 이 옵션을 사용하지 마십시오. 이 옵션은 동적 SQL 성능을 크게 개선시킬 수 있습니다. 구문은 다음과 같습니다.

```
generic "option1 value1 option2 value2 ..."
```

각 옵션 및 값은 하나 이상의 공백으로 구분되어야 합니다. 예를 들어, 목표 DRDA 데이터베이스가 DB2 Universal Database, 버전 8이면,

```
generic "explsnap all queryopt 3 federated yes"
```

을 사용하여 각 EXPLSNAP, QUERYOPT 및 FEDERATED 옵션을 바인드할 수 있습니다.

문자열의 최대 길이는 32768바이트입니다.

## GRANT

주: GRANT, GRANT\_GROUP, GRANT\_USER 및 GRANT\_ROLE 옵션 중 두 개 이상을 지정한 경우, 지정한 마지막 옵션만 실행됩니다.

*authid* 지정된 사용자 이름, 역할 이름 또는 그룹 ID에 EXECUTE 및 BIND 특권을 부여합니다. GRANT문에서 권한 받은 사용자의 유형을 지정하는 데 USER, GROUP 또는 ROLE 중 어느 것도 지정하지 않은 경우 권한 ID 유형을 판별하기 위해 SQL GRANT 문 및 해당 규칙을 사용합니다. 규칙에 대해서는 *GRANT(역할)*문을 참조하십시오.

### PUBLIC

EXECUTE 및 BIND 특권을 PUBLIC에 부여합니다.

### GRANT\_GROUP *group-name*

EXECUTE 및 BIND 특권을 지정된 그룹 이름에 부여합니다.

### GRANT\_USER *user-name*

EXECUTE 및 BIND 특권을 지정된 사용자 이름에 부여합니다.

### GRANT\_ROLE *role-name*

EXECUTE 및 BIND 특권을 지정된 역할 이름에 부여합니다.

## INSERT

DB2 Enterprise Server Edition 서버에 대해 프리컴파일되거나 바인드되는 프로그램이 성능을 향상시키기 위해 해당 데이터 삽입을 버퍼 지정하도록 요청하게 합니다.

**BUF** 응용프로그램에서의 삽입이 버퍼 지정되도록 지정합니다.

**DEF** 응용프로그램에서의 삽입이 버퍼 지정되지 않도록 지정합니다.

## ISOLATION

이 패키지에 바인드되는 프로그램이 다른 실행 프로그램의 효과로부터 얼마나 분리되는지 정도를 판별합니다.

- CS** 분리 레벨로 커서 안정성을 지정합니다.
- NC** 커미트가 없습니다. 커미트 제어가 사용되지 않음을 지정합니다. 이 분리 레벨은 Linux, UNIX 및 Windows용 DB2 데이터베이스에서 지원되지 않습니다.
- RR** 반복 읽기를 분리 레벨로 지정합니다.
- RS** 읽기 안정성을 분리 레벨로 지정합니다. 패키지의 SQL문 실행이 행 읽기를 위한 다른 응용프로그램 프로세스에서 분리되며 응용프로그램에 의해 변경되도록 읽기 안정성이 보장합니다.
- UR** 커미트되지 않은 읽기를 분리 레벨로 지정합니다.

#### **IMMEDWRITE**

그룹 버퍼 풀(GBP) 종속 페이지세트나 데이터베이스 파티션에 작성되는 갱신에 즉시 쓰기가 수행되는지 여부를 표시합니다. OS/390 전용 DB2에 의해 지원됩니다. 지원되는 옵션 값 목록은 OS/390용 DB2 문서를 참조하십시오.

#### **KEEPDYNAMIC**

동적 SQL문이 커미트 지점 이후에 보존되는지 여부를 지정합니다. OS/390 전용 DB2에 의해 지원됩니다. 지원되는 옵션 값 목록은 OS/390용 DB2 문서를 참조하십시오.

#### **MESSAGES** *message-file*

경고, 오류 및 완료 상태 메시지의 대상을 지정합니다. 바인드가 정상적인지 여부의 메시지 파일이 작성됩니다. 메시지 파일 이름이 지정되지 않으면 메시지가 표준 출력에 작성됩니다. 파일에 대한 전체 경로가 지정되지 않으면, 전체 경로가 사용됩니다. 기존 파일의 이름이 지정되지 않으면, 파일의 콘텐츠가 겹쳐쓰입니다.

#### **OPTHINT**

쿼리 최적화 힌트가 정적 SQL에 사용되는지 여부를 제어합니다. OS/390 전용 DB2에 의해 지원됩니다. 지원되는 옵션 값 목록은 OS/390용 DB2 문서를 참조하십시오.

#### **OPTPROFILE** *optimization-profile-name*

패키지에서 모든 정적문에 사용되는 기존 최적화 프로파일의 이름을 지정합니다. 옵션의 디폴트값은 빈 문자열입니다. CURRENT OPTIMIZATION PROFILE 특수 레지스터가 널(NULL)인 DML문의 동적 준비를 위한 디폴트값으로 해당 값을 적용합니다. 지정된 이름이 규정되지 않은 경우, QUALIFIER 바인드 옵션에 의해 내재적으로 자격이 부여된 SQL ID입니다.

BIND 명령은 최적화 파일을 처리하지 않지만, 이름이 구문적으로 유효하지만 확인합니다. 그러므로 최적화 프로파일이 없거나 유효하지 않은 경우, DML문이 해당 최적화 프로파일을 사용하여 최적화될 때까지 이유 코드가 13인 SQL0437W 경고는 발생하지 않습니다.

## OS400NAMING

DB2 for System i<sup>®</sup> 데이터 액세스 시 이름 지정 옵션이 사용됨을 지정합니다. DB2 for System i에 의해서만 지원됩니다. 지원되는 옵션 값 목록은 System i용 DB2의 문서를 참조하십시오.

구분자로 사용되는 슬래시 때문에, 유틸리티가 OS400NAMING SYSTEM 옵션으로 바인드되거나 프리컴파일된 경우에도 DB2 유틸리티는 System i 시스템 이름 지정 규칙을 사용하는 특정 SQL문에 대해 실행 시 구문 오류를 보고할 수 있습니다. 예를 들어, System i 시스템 이름 지정 규칙이 사용된 경우, OS400NAMING SYSTEM 옵션을 사용하여 프리컴파일 또는 바인드되었는지에 관계없이 명령행 처리기는 SQL CALL문에 대해 구문 오류를 보고합니다.

### OWNER *authorization-id*

패키지 소유자의 128바이트 권한 부여 ID를 지정합니다. 소유자는 패키지에 포함된 SQL문을 실행하는 데 필요한 특권을 가져야 합니다. DBADM 권한을 가진 사용자만이 사용자 ID 이외의 권한 부여 ID를 지정할 수 있습니다. 디폴트 값은 프리컴파일/바인드 프로세스 호출자의 권한 부여 ID입니다. SYSIBM, SYSCAT 및 SYSSTAT는 이 옵션에 대해 유효한 값이 아닙니다. *authorization-id*는 사용자여야 합니다. 역할 또는 그룹은 OWNER 옵션을 사용하여 지정할 수 없습니다.

### PATH

사용자 정의 구별 유형에서 사용되는 함수 경로와 정적 SQL의 함수를 지정합니다. 이 옵션이 지정되지 않으면, 디폴트 함수 경로는 "SYSIBM","SYSFUN",USER이며 여기서 USER는 USER 특수 레지스터의 값입니다.

#### *schema-name*

응용프로그램 서버(AS)에 존재하는 스키마를 식별하는 SQL ID로, 일반 또는 구분된 유형이 있습니다. 프리컴파일 또는 바인드 시간에는 스키마가 존재하는지 확인되지 않습니다.

### QUALIFIER *qualifier-name*

패키지에 포함된 규정되지 않은 오브젝트의 128바이트 내재된 규정자를 제공합니다. 소유자가 명시적으로 지정되었는지 여부와 관계없이, 디폴트는 소유자의 권한 부여 ID입니다.

### QUERYOPT *optimization-level*

패키지에 포함된 모든 정적 SQL문의 원하는 최적화 레벨을 표시합니다. 디폴트 값은 5입니다. SET CURRENT QUERY OPTIMIZATION문은 사용 가능한 최적화 레벨의 전체 범위를 기술합니다. 이 DB2 프리컴파일/바인드 옵션은 DRDA에서 지원되지 않습니다.

### RELEASE

자원이 각 커밋 지점에서 릴리스되었는지 여부와 응용프로그램이 종료하는 시

기를 표시합니다. 이 DRDA 프리컴파일/바인드 옵션은 Linux, UNIX 및 Windows용 DB2 데이터베이스에서 지원되지 않습니다.

### **COMMIT**

각 커밋 지점에서 자원을 릴리스합니다. 동적 SQL문에 사용됩니다.

### **DEALLOCATE**

응용프로그램 종료 시에만 자원을 릴리스합니다.

### **SORTSEQ**

System i 시스템에서 사용할 정렬 시퀀스 테이블을 지정합니다. DB2 for System i에 의해서만 지원됩니다. 지원되는 옵션 값 목록은 System i용 DB2의 문서를 참조하십시오.

### **SQLERROR**

오류 발생 시 패키지나 바인드 파일을 작성할지 여부를 표시합니다.

### **CHECK**

목표 시스템이 바인드 중인 SQL문의 모든 구문과 시맨틱 점검을 수행하도록 지정합니다. 패키지는 이 프로세스의 파트로 작성되지 않습니다. 바인드 중, 동일한 이름과 버전을 가진 기존 패키지가 있으면, ACTION REPLACE가 지정된 경우에도 기존 패키지는 삭제되거나 교체되지 않습니다.

### **CONTINUE**

SQL문 바인드 시 오류가 발생하는 경우에도 패키지를 작성합니다. VALIDATE RUN이 지정된 경우 권한 부여 또는 기존 이유로 바인드에 실패한 명령문은 실행 시간에 증분식으로 바인드될 수 있습니다. 런타임에 실행하면 오류(SQLCODE -525, SQLSTATE 51015)를 생성합니다.

### **NOPACKAGE**

오류 발생 시 패키지나 바인드 파일이 작성되지 않습니다.

### **REOPT**

호스트 변수, 매개변수 표시문자, 전역 변수 및 특수 레지스터의 값을 사용하여 DB2가 런타임 시 액세스 경로를 최적화하도록 할 것인지 여부를 지정합니다. 가능한 값은 다음과 같습니다.

#### **NONE**

호스트 변수, 매개변수 표시문자, 전역 변수 또는 특수 레지스터를 포함하는 제공된 SQL문의 액세스 경로는 실제 값을 사용하여 최적화되지 않습니다. 이들 변수의 디폴트 추정치가 사용되며, 플랜은 캐시되어 이후에 사용됩니다. 디폴트값입니다.

#### **ONCE**

제공된 SQL문의 액세스 경로는 쿼리가 처음 실행될 때 호스트 변수,

매개변수 표시문자, 전역 변수 또는 특수 레지스터의 실제 값을 사용하여 최적화됩니다. 이 플랜은 캐시되어 차후에 사용됩니다.

### **ALWAYS**

제공된 SQL문의 액세스 경로는 항상 쿼리가 실행될 때마다 알려지는 호스트 변수, 매개변수 표시문자, 전역 변수 또는 특수 레지스터의 값을 사용하여 컴파일되고 재최적화됩니다.

### **REOPT | NOREOPT VARS**

이 옵션은 REOPT ALWAYS 및 REOPT NONE에 의해 교체되었지만, 이전 릴리스와의 호환성을 위해 여전히 지원됩니다. 호스트 변수, 매개변수 표시문자, 전역 변수 및 특수 레지스터의 값을 사용하여 런타임에 DB2가 액세스 경로를 판별할지 여부를 지정합니다. OS/390 전용 DB2에 의해 지원됩니다. 지원되는 옵션 값 목록은 OS/390용 DB2 문서를 참조하십시오.

### **SQLWARN**

동적 SQL문 컴파일(PREPARE 또는 EXECUTE IMMEDIATE를 통한), 또는 설명 처리(PREPARE...INTO 또는 DESCRIBE를 통한)에서 경고가 리턴되는지 여부를 표시합니다.

**NO** 경고는 SQL 컴파일러에서 리턴되지 않습니다.

**YES** 경고는 SQL 컴파일러에서 리턴됩니다.

SQLCODE +236, +237 및 +238은 예외입니다. SQLWARN 옵션 값과 무관하게 리턴됩니다.

### **STATICREADONLY**

정적 커서가 READ ONLY로 처리되는지 여부를 판별합니다. 이 DB2 프리컴파일/바인드 옵션은 DRDA에서 지원되지 않습니다.

**NO** 모든 정적 커서는 제공된 명령문 텍스트와 LANGLEVEL 프리컴파일 옵션 설정이 정상적으로 생성될 때 속성을 가져옵니다. 이것은 디폴트 값입니다.

**YES** FOR UPDATE 또는 FOR READ ONLY절을 포함하고 있지 않은 모든 정적 커서는 READ ONLY로 간주됩니다.

### **STRDEL**

아포스트로피(') 또는 큰따옴표(")가 SQL문 내에서 문자열 분리문자로 사용될지 여부를 지정합니다. 이 DRDA 프리컴파일/바인드 옵션은 Linux, UNIX 및 Windows용 DB2 데이터베이스에서 지원되지 않습니다. 이 옵션이 지정되지 않은 경우 DRDA 서버는 시스템 정의 디폴트값을 사용합니다.

### **APOSTROPHE**

문자열 분리문자로 아포스트로피(')를 사용합니다.



## QUOTE

문자열 분리문자로 큰따옴표(")를 사용합니다.

## TEXT *label*

패키지의 설명. 최대 길이는 255문자입니다. 디폴트값은 공백입니다. 이 DRDA 프리컴파일/바인드 옵션은 Linux, UNIX 및 Windows용 DB2 데이터베이스에서 지원되지 않습니다.

## TRANSFORM GROUP

사용자 정의 구조화된 유형 값을 호스트 프로그램과 교환하기 위해 동적 SQL 문에 사용되는 변환 그룹의 이름을 지정합니다. 이 변환 그룹은 동적 SQL문이나 매개변수와 결과를 외부 함수 또는 메소드와 교환하기 위해 사용되지 않습니다. 이 옵션은 DRDA에서 지원되지 않습니다.

### *groupname*

길이가 최대 18바이트인 SQL ID입니다. 그룹 이름은 규정자 접두부를 포함할 수 없으며 데이터베이스 사용을 위해 예약되어 있으므로 접두부 SYS로 시작할 수 없습니다. 호스트 변수와 상호작용하는 정적 SQL문에서, 구조화된 유형의 값 교환을 위해 사용되는 변환 그룹의 이름은 다음과 같습니다.

- TRANSFORM GROUP 바인드 옵션에 있는 그룹 이름(있는 경우).
- 원래 프리컴파일 시간에 지정된 대로 TRANSFORM GROUP prep 옵션에 있는 그룹 이름(있는 경우).
- 그룹 이름이 DB2\_PROGRAM인 유형에 대한 변환이 존재하는 경우, DB2\_PROGRAM 그룹
- 위의 조건 중 하나도 존재하지 않는 경우 변환 그룹 없음이 사용됩니다.

다음 오류는 정적 SQL문의 바인드 동안 가능합니다.

- SQLCODE yyyyy, SQLSTATE xxxxx: 변환이 필요하지만, 정적 변환 그룹이 선택되지 않았습니다.
- SQLCODE yyyyy, SQLSTATE xxxxx: 선택된 변환 그룹이 교환해야 하는 데이터 유형에 대해 필요한 변환(입력 변수의 경우 TO SQL, 출력 변수의 경우 FROM SQL)을 포함하지 않습니다.
- SQLCODE yyyyy, SQLSTATE xxxxx: FROM SQL 변환의 결과 유형이 출력 변수의 유형과 호환 가능하지 않거나, TO SQL 변환의 매개변수 유형이 입력 변수의 유형과 호환 가능하지 않습니다.

이 오류 메시지에서, yyyyy는 SQL 오류 코드로 교체되며 xxxxx는 SQL 상태 코드로 교체됩니다.

## VALIDATE

데이터베이스 관리 프로그램가 권한 부여 오류와 오브젝트를 찾을 수 없음 오류를 점검하는 시기를 판별합니다. 패키지 소유자 권한 부여 ID가 유효성 점검에 사용됩니다.

**BIND** 유효성 확인은 프리컴파일/바인드 시간에 수행됩니다. 모든 오브젝트가 존재하지 않거나, 모든 권한이 보유되지 않으면, 오류 메시지가 생성됩니다. `SQLERROR CONTINUE`가 지정되면, 오류 메시지와 관계없이 패키지/바인드 파일이 생성되지만 오류에 있는 명령문은 실행할 수 없습니다.

**RUN** 유효성 확인이 바인드 시간에 시도됩니다. 모든 오브젝트가 존재하며 모든 권한이 보유되지만, 실행 시간에 더 이상의 점검은 수행되지 않습니다.

모든 오브젝트가 존재하지 않거나 프리컴파일/바인드 시간에 모든 권한이 보유되지 않으면, 경고 메시지가 생성되며, `SQLERROR CONTINUE` 옵션 설정과 관계없이 패키지가 정상적으로 바인드됩니다. 그러나 프리컴파일/바인드 프로세스 동안 해당 점검에 실패한 SQL문의 존재 점검과 권한 점검은 실행 시간에 재실행될 수 있습니다.

## 예:

다음 예는 `myapp.bnd(myapp.sqc` 프로그램이 프리컴파일될 때 생성된 바인드)를 연결된 데이터베이스에 바인드합니다.

```
db2 bind myapp.bnd
```

바인드 프로세스의 결과 메시지는 표준 출력으로 보냅니다.

## 사용 시 참고사항

`ONCE` 또는 `ALWAYS` 값이 지정된 `REOPT` 옵션을 사용하여 패키지를 바인드하면 정적문과 동적문의 컴파일 및 성능이 변경될 수 있습니다.

응용프로그램 소스 파일에 대한 프리컴파일 프로세스 중 또는 이후에 별도의 단계를 수행하여 바인드를 완료할 수 있습니다. 별도의 프로세스로 바인드를 수행할 때는 `BIND`를 사용하십시오.

패키지 작성 시에 사용한 이름은 바인드 파일에 저장되고 생성된 소스 파일을 기반으로 합니다(기존 경로나 확장자는 버림). 예를 들어 프리컴파일된 소스 파일인 `myapp.sql`은 디폴트 바인드 파일 `myapp.bnd`와 디폴트 패키지 이름 `MYAPP`를 생성합니다. 그러나 바인드 파일 이름 및 패키지 이름은 `BINDFILE` 및 `PACKAGE` 옵션을 사용하여 프리컴파일 시에 겹쳐줄 수 있습니다.

아직 존재하지 않는 스키마 이름으로 패키지를 바인드하면 해당 스키마가 내재적으로 작성됩니다. 스키마 소유자는 SYSIBM입니다. 스키마에 대한 CREATEIN 특권은 PUBLIC에 권한 부여됩니다.

BIND는 시작된 트랜잭션에서 실행됩니다. 바인드 수행 후, BIND는 COMMIT 또는 ROLLBACK을 실행하여 현재 트랜잭션을 종료하고 다른 트랜잭션을 시작합니다.

치명적 오류나 100개 이상의 오류가 발생하면 바인딩이 중지됩니다. 치명적 오류가 발생한 경우, 유틸리티가 바인딩을 중지하고 모든 파일을 닫은 후 패키지를 버립니다.

패키지가 바인드 동작을 표시하면, 다음이 참이 됩니다.

1. BIND 옵션 OWNER의 내재된 또는 명시적 값이 동적 SQL문의 권한 부여 검사에 사용됩니다.
2. BIND 옵션 QUALIFIER의 내재된 또는 명시적 값이 동적 SQL문 내에서 규정되지 않은 자격의 내재된 규정자로서 사용됩니다.
3. 특수 레지스터 CURRENT SCHEMA의 값은 자격에 영향을 미치지 않습니다.

단일 연결 동안 여러 패키지가 참조될 경우 해당 패키지에서 준비한 모든 동적 SQL문은 사용되는 특정 패키지 및 환경에 대해 DYNAMICRULES 옵션에서 지정한 동작을 제시합니다.

SQL0020W 메시지에 표시된 매개변수는 올바르게 오류로 명시되지만 메시지에 표시된 대로 무시됩니다.

SQL문에 오류가 있는 것으로 발견되고 BIND 옵션 SQLERROR CONTINUE가 지정된 경우, 명령문은 유효하지 않은 것으로 표시됩니다. SQL문의 상태를 변경하려면 다른 BIND를 실행해야 합니다. 내재된 리바인드 및 명시적 리바인드는 유효하지 않은 명령문의 상태를 변경하지 않습니다. VALIDATE RUN으로 바인드된 패키지에서, 오브젝트 존재 여부나 리바인드 중 권한 문제점 존재 여부에 따라 내재된 리바인드 및 명시적 리바인드에 걸쳐 명령문은 정적에서 증분식 바인드로 또는 증분식 바인드에서 정적 바인드로 변경할 수 있습니다.

패키지를 바인드하기 위해 사용된 권한 부여 ID(OWNER 바인드 옵션의 값)나 PUBLIC에 부여된 역할의 특권은 패키지를 바인드할 때 고려됩니다. 패키지를 바인드하기 위해 사용된 권한 부여 ID가 구성원인 그룹을 통해 획득된 역할은 사용되지 않습니다.

Embedded SQL 프로그램의 경우, 바인드 옵션이 명시적으로 지정되지 않으면 FEDERATED\_ASYNC 구성 매개변수를 사용하여 패키지의 정적문이 바인드됩니다. FEDERATED\_ASYNCHRONY 바인드 옵션이 명시적으로 지정되면, 해당 값은 패키지를 바인딩하는 데 사용되며 특수 레지스터의 초기값이기도 합니다. 그렇지 않으면, 데

이터베이스 관리 프로그램 구성 매개변수의 값이 특수 레지스터의 초기 값으로 사용됩니다. FEDERATED\_ASYNCRONY 바인드 옵션은 명시적으로 설정될 경우에만 동적 SQL에 영향을 미칩니다.

FEDERATED\_ASYNCRONY 바인드 옵션의 값은 SYSCAT.PACKAGES 카탈로그 테이블의 FEDERATED\_ASYNCRONY 컬럼에 기록됩니다. 바인드 옵션을 명시적으로 지정하지 않은 경우 FEDERATED\_ASYNC 구성 매개변수의 값이 사용되고 카탈로그는 FEDERATED\_ASYNCRONY 컬럼에 대해 -2 값을 표시합니다.

패키지가 바인드될 때 FEDERATED\_ASYNCRONY 바인드 옵션이 명시적으로 지정되지 않고 이 패키지가 내재적으로나 명시적으로 리바인드되는 경우, 패키지는 FEDERATED\_ASYNC 구성 매개변수의 현재 값을 사용하여 리바인드됩니다.

## 제 17 장 CATALOG DATABASE

데이터베이스 위치 정보를 시스템 데이터베이스 디렉토리에 저장합니다. 로컬 워크스테이션이나 리모트 데이터베이스 파티션 서버 중 하나에 데이터베이스를 위치시킬 수 있습니다.

### 범위

파티션된 데이터베이스 환경에서 로컬 데이터베이스를 시스템 데이터베이스 디렉토리로 카탈로그할 때, 데이터베이스가 있는 서버의 데이터베이스 파티션에서 이 명령을 발행해야 합니다.

### 권한 부여

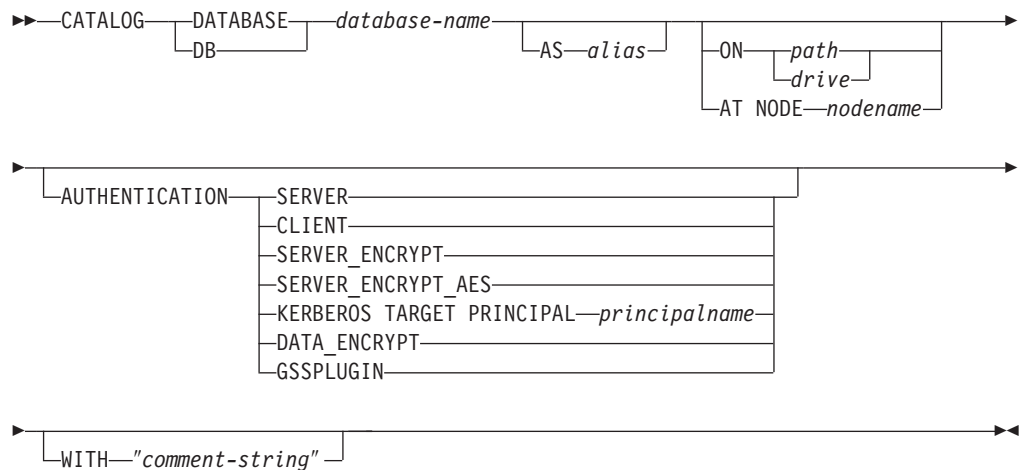
다음 중 하나가 필요합니다.

- SYSADM
- SYSCTRL

### 필수 연결

없음. 디렉토리 조작은 로컬 디렉토리에만 영향을 미칩니다.

### 명령 구문



### 명령 매개변수

**DATABASE** *database-name*

데이터베이스 이름을 카탈로그에 지정합니다.

### **AS alias**

카탈로그되는 데이터베이스의 대체 이름으로 별명을 지정합니다. 별명이 지정되지 않으면, 데이터베이스 관리 프로그램은 별명으로 *database-name*을 사용합니다.

### **ON path | drive**

카탈로그되는 데이터베이스가 있는 경로를 지정합니다. Windows 운영 체제에서, 카탈로그되는 데이터베이스가 있는 드라이브명을 대신 지정할 수 있습니다 (특정 경로가 아닌 드라이브에서 작성된 경우).

### **AT NODE nodename**

카탈로그되는 데이터베이스가 있는 데이터베이스 파티션 서버의 이름을 지정합니다. 이 이름은 노드 디렉토리에 있는 항목 이름과 일치해야 합니다. 지정된 노드 이름이 노드 디렉토리에 없는 경우, 경고가 리턴되지만 데이터베이스는 시스템 데이터베이스 디렉토리에 카탈로그됩니다. 카탈로그된 데이터베이스로의 연결이 필요한 경우 노드 이름이 노드 디렉토리에서 카탈로그되어야 합니다.

## **AUTHENTICATION**

리모트 데이터베이스의 인증 값은 저장되지만(LIST DATABASE DIRECTORY 명령의 출력에 표시) 로컬 데이터베이스의 인증 값은 저장되지 않습니다.

인증 유형을 지정하면 성능 이점을 가져올 수 있습니다.

### **SERVER**

목표 데이터베이스를 포함하는 데이터베이스 파티션 서버에서 인증이 발생하도록 지정합니다.

### **CLIENT**

응용프로그램이 호출되는 데이터베이스 파티션 서버에서 인증이 발생하도록 지정합니다.

### **SERVER\_ENCRYPT**

목표 데이터베이스를 포함하는 데이터베이스 파티션 서버에서 인증이 발생하도록 지정하며, 사용자 ID 및 암호가 소스에서 암호화됩니다. 소스에서 카탈로그된 인증 유형에서 지정한 대로 사용자 ID 및 암호가 목표에서 암호 해독됩니다.

## **KERBEROS**

Kerberos 보안 메커니즘을 사용하여 인증이 발생함을 지정합니다.

### **TARGET PRINCIPAL principalname**

목표 서버의 완전한 Kerberos 핵심부 이름, 즉 name/instance@REALM 양식으로 된 DB2 인스턴스 소유자의 완전한 Kerberos 핵심부. Windows 2000, Windows XP 및 Windows Server 2003의 경우, 이것은 *userid@DOMAIN*,

`userid@xxx.xxx.xxx.com` 또는 `domain#userid` 양식으로 된 DB2 서버 서비스의 로그인 어카운트입니다.

### **DATA\_ENCRYPT**

목표 데이터베이스를 포함하는 데이터베이스 파티션 서버에서 인증이 발생하도록 지정하며, 해당 연결은 데이터 암호화를 사용해야 합니다.

### **GSSPLUGIN**

외부 GSS API 기반 플러그인 보안 메커니즘을 사용하여 인증이 발생함을 지정합니다.

### **SERVER\_ENCRYPT\_AES**

목표 데이터베이스가 들어있는 데이터베이스 파티션 서버에서 인증이 발생하고, 사용자 ID 및 암호가 소스에서 AES(Advanced Encryption Standard) 암호화 알고리즘을 사용하여 암호화되고 목표에서 암호 해독되도록 지정합니다.

### **WITH "comment-string"**

시스템 데이터베이스 디렉토리의 데이터베이스 디렉토리 항목이나 데이터베이스를 설명합니다. 주석 문자열의 최대 길이는 30문자입니다. 캐리지 리턴이나 줄 바꾸기 문자는 허용되지 않습니다. 주석 텍스트는 큰따옴표로 묶어야 합니다.

### **예:**

```
db2 catalog database sample on /databases/sample  
with "Sample Database"
```

## **사용 시 참고사항**

CATALOG DATABASE를 사용하여 로컬 또는 리모트 데이터베이스 파티션 서버에 있는 데이터베이스를 카탈로그하거나, 이전에 카탈로그되지 않은 데이터베이스를 재카탈로그하거나, 하나의 데이터베이스에 대해 다중 별명을 유지할 수 있습니다(데이터베이스 위치에 관계없이).

DB2에서 데이터베이스 작성 시 자동으로 카탈로그. 데이터베이스의 항목을 로컬 데이터베이스 디렉토리에서 카탈로그하고 다른 항목을 시스템 데이터베이스 디렉토리에서 카탈로그합니다. 데이터베이스가 리모트 클라이언트(또는 동일한 머신의 다른 인스턴스에서 실행 중인 클라이언트)에서 작성되는 경우, 클라이언트 인스턴스의 시스템 데이터베이스 디렉토리에서 항목이 작성됩니다.

경로 또는 데이터베이스 파티션 서버 이름 중 어느 것도 지정되지 않으면, 데이터베이스는 로컬로 가정되며, 데이터베이스의 위치는 데이터베이스 관리 프로그램 구성 매개변수 `dftdbpath`에 지정된 것으로 가정됩니다.

데이터베이스 관리 프로그램 인스턴스와 동일한 데이터베이스 파티션 서버의 데이터베이스는 간접 항목으로 카탈로그됩니다. 다른 데이터베이스 파티션 서버의 데이터베이스는 리모트 항목으로 카탈로그됩니다.

CATALOG DATABASE는 시스템 데이터베이스 디렉토리가 없는 경우 자동으로 작성합니다. 시스템 데이터베이스 디렉토리는 사용 중인 데이터베이스 관리 프로그램 인스턴스를 포함하는 경로에 저장되며, 데이터베이스 외부에 유지됩니다.

LIST DATABASE DIRECTORY 명령을 사용하여 시스템 데이터베이스 디렉토리의 콘텐츠를 나열하십시오. 로컬 데이터베이스 디렉토리의 콘텐츠를 나열하려면 LIST DATABASE DIRECTORY ON *path*를 사용하십시오. 여기서 *path*는 데이터베이스가 작성된 위치입니다.

디렉토리 캐싱이 사용 가능하면, 데이터베이스, 노드 및 DCS 디렉토리 파일은 메모리에 캐시됩니다. 디렉토리 캐싱이 사용 가능한지 확인하려면, GET DATABASE MANAGER CONFIGURATION 명령에서 출력의 *dir\_cache* 디렉토리 캐시 지원 구성 매개변수의 값을 점검하십시오. 응용프로그램 디렉토리 캐시는 최초 디렉토리 찾아보기 중 작성됩니다. 캐시는 응용프로그램이 임의의 디렉토리 파일을 수정할 때에만 새로 고쳐지므로, 다른 응용프로그램으로 작성된 디렉토리 변경사항은 응용프로그램이 재시작될 때까지 유효하지 않습니다.

CLP의 디렉토리 캐시를 새로 고치려면, TERMINATE 명령을 사용하십시오. 데이터베이스 관리 프로그램의 공유 캐시를 새로 고치려면, 데이터베이스 관리 프로그램을 중지(db2stop)한 후 재시작(db2start)하십시오. 다른 응용프로그램의 디렉토리 캐시를 새로 고치려면, 해당 응용프로그램을 중지한 후 재시작하십시오.



## 제 18 장 CATALOG DCS DATABASE

리모트 호스트 또는 System i 데이터베이스에 대한 정보를 데이터베이스 연결 서비스 (DCS) 디렉토리에 저장합니다. 이 데이터베이스는 DB2® Connect™와 같은 응용프로그램 리퀘스터(AR)를 통해 액세스됩니다. 시스템 데이터베이스 디렉토리에 있는 데이터베이스 이름과 일치하는 데이터베이스 이름을 가진 DCS 디렉토리 항목이 있으면 지정된 AR을 호출하여 데이터베이스가 있는 리모트 서버로 SQL 요청을 전달합니다.

### 권한 부여

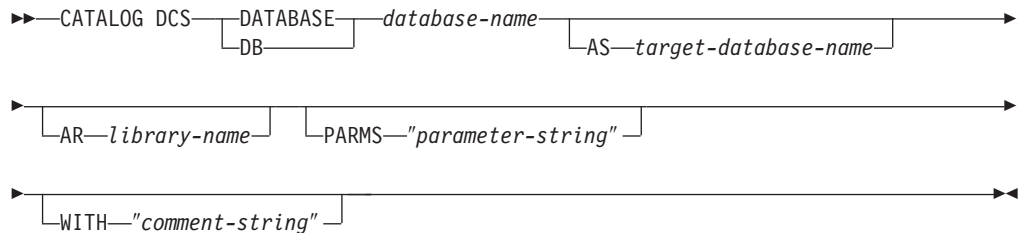
다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

#### **DATABASE** *database-name*

목표 데이터베이스의 별명을 카탈로그에 지정합니다. 이 이름은 리모트 데이터베이스 파티션 서버와 연관된 데이터베이스 디렉토리에 있는 항목의 이름과 일치해야 합니다.

#### **AS** *target-database-name*

목표 호스트나 System i 데이터베이스의 이름을 카탈로그에 지정합니다.

#### **AR** *library-name*

DCS 디렉토리에 나열된 리모트 데이터베이스를 액세스하기 위해 로드되고 사용되는 응용프로그램 리퀘스터(AR) 라이브러리의 이름을 지정합니다.

DB2 Connect AR을 사용하는 경우, 라이브러리 이름을 지정하지 마십시오. 디폴트값은 DB2 Connect가 호출되게 합니다.

DB2 Connect를 사용 중이지 않으면, AR의 라이브러리 이름을 지정하고 데이터베이스 관리 프로그램 라이브러리와 동일한 경로에 해당 라이브러리를 위치시키십시오. Windows 운영 체제에서, 경로는 drive:\sql1lib\bin입니다. UNIX 기반 시스템에서, 경로는 인스턴스 소유자의 \$HOME/sql1lib/lib입니다.

**PARMS** *"parameter-string"*

호출 시 AR로 전달되는 매개변수 문자열을 지정합니다. 매개변수 문자열은 큰 따옴표로 묶어야 합니다. 매개변수 문자열에 대한 자세한 정보는 관련 개념 섹션의 "DCS 디렉토리 값"을 참조하십시오.

**WITH** *"comment-string"*

DCS 디렉토리 항목을 설명합니다. 이 디렉토리에 카탈로그된 데이터베이스를 설명하도록 돕는 주석을 입력할 수 있습니다. 최대 길이는 30문자입니다. 캐리지 리턴이나 줄 바꾸기 문자는 허용되지 않습니다. 주석 텍스트는 큰따옴표로 묶어야 합니다.

**예:**

다음 예는 z/OS®용 DB2 데이터베이스인 DB1 데이터베이스에 대한 정보를 DCS 디렉토리에 카탈로그합니다.

```
db2 catalog dcs database db1 as dsn_db_1
with "DB2/z/OS location name DSN_DB_1"
```

**사용 시 참고사항**

DB2 Connect 프로그램은 다음과 같이 DRDA 응용프로그램 서버(AS)에 대한 연결을 제공합니다.

- OS/390용 DB2 또는 System/370™ 및 System/390® 구조 호스트 컴퓨터의 z/OS 데이터베이스.
- System/370 및 System/390 구조 호스트 컴퓨터에서의 VM 및 VSE 데이터베이스의 DB2.
- Application System/400® (System i) 및 System i 컴퓨터의 System i 데이터베이스.

데이터베이스 관리 프로그램은 데이터베이스 연결 서비스 디렉토리가 없는 경우 작성합니다. 이 디렉토리는 사용 중인 데이터베이스 관리 프로그램 인스턴스를 포함하는 경로에 저장됩니다. DCS 디렉토리는 데이터베이스 외부에 유지됩니다.

데이터베이스는 시스템 데이터베이스 디렉토리에서 리모트 데이터베이스로서 카탈로그되어야 합니다.

LIST DCS DIRECTORY 명령을 사용하여 DCS 디렉토리의 콘텐츠를 나열하십시오.

디렉토리 캐싱이 사용 가능하면, 데이터베이스, 노드 및 DCS 디렉토리 파일은 메모리에 캐시됩니다. 응용프로그램 디렉토리 캐시는 최초 디렉토리 찾아보기 중 작성됩니다. 캐시는 응용프로그램이 임의의 디렉토리 파일을 수정할 때에만 새로 고쳐지므로, 다른 응용프로그램으로 작성된 디렉토리 변경사항은 응용프로그램이 재시작될 때까지 유효하지 않습니다.

CLP의 디렉토리 캐시를 새로 고치려면, TERMINATE 명령을 사용하십시오. DB2의 공유 캐시를 새로 고치려면, 데이터베이스 관리 프로그램을 중지(db2stop)한 후 재시작(db2start)하십시오. 다른 응용프로그램의 디렉토리 캐시를 새로 고치려면, 해당 응용프로그램을 중지한 후 재시작하십시오.



## 제 19 장 CATALOG LDAP DATABASE

데이터베이스를 LDAP(Lightweight Directory Access Protocol)에 등록합니다.

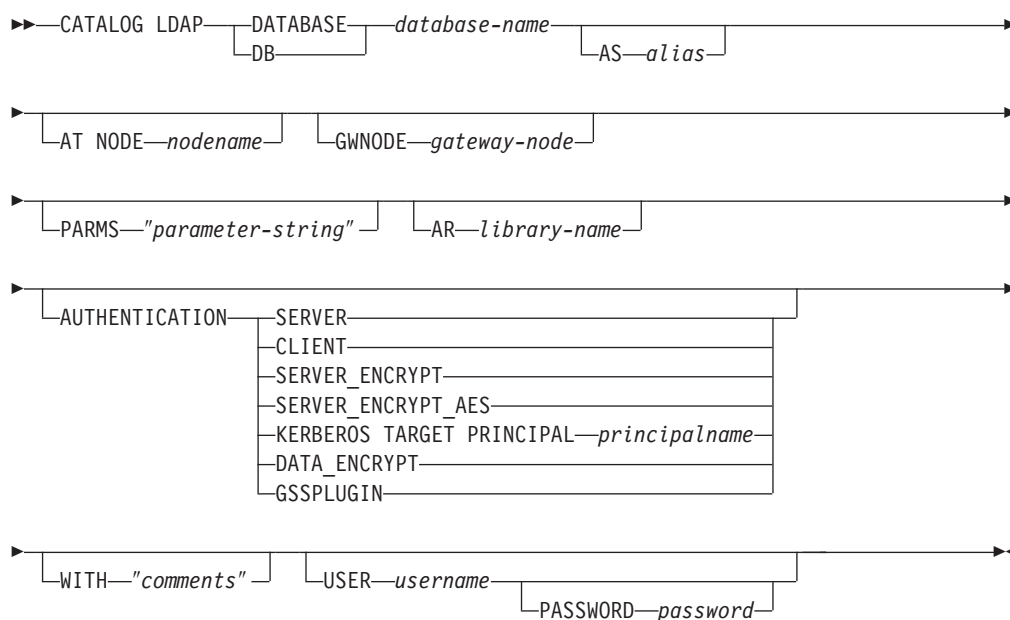
### 권한 부여

없음

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

#### **DATABASE** *database-name*

카탈로그할 데이터베이스의 이름을 지정합니다.

#### **AS** *alias*

카탈로그될 데이터베이스에 대한 대체 이름으로 별명을 지정합니다. 별명이 지정되지 않는 경우 데이터베이스 이름이 별명으로 사용됩니다.

#### **AT NODE** *nodename*

데이터베이스가 있는 데이터베이스 서버에 대한 LDAP 노드 이름을 지정합니다. 이 매개변수는 리모트 서버에서 데이터베이스를 등록할 때 지정되어야 합니다.

## **GWNODE** *gateway-node*

게이트웨이 서버의 LDAP 노드 이름을 지정합니다.

## **PARMS** "*parameter-string*"

DCS 데이터베이스에 액세스할 때 응용프로그램 리퀘스터(AR)로 전달되는 매개변수 문자열을 지정합니다. 암호 변경 *sym\_dest\_name*은 매개변수 문자열에 지정되지 않아야 합니다. LDAP에 DB2 서버를 등록할 때 암호 변경 LU 이름을 지정하려면 키워드 **CHGPWDLU**를 사용하십시오.

## **AR** *library-name*

로드되고 DCS 디렉토리에 나열되는 리모트 데이터베이스에 액세스하는 데 사용되는 응용프로그램 리퀘스터(AR) 라이브러리의 이름을 지정합니다.

DB2 Connect AR을 사용 중인 경우 라이브러리 이름을 지정하지 마십시오. 디폴트값은 DB2 Connect가 호출되도록 합니다.

DB2 Connect를 사용하지 않는 경우, AR의 라이브러리 이름을 지정하고 해당 라이브러리를 데이터베이스 관리 프로그램 라이브러리와 동일한 경로에 배치하십시오. Windows 운영 체제에서 경로는 `drive:\sqllib\wdll`입니다. UNIX 운영 체제에서 경로는 인스턴스 소유자의 `$HOME/sqllib/lib`입니다.

## **AUTHENTICATION**

인증 레벨을 지정합니다. 가능한 값은 다음과 같습니다.

### **SERVER**

목표 데이터베이스가 들어있는 노드에서 인증이 발생하도록 지정합니다.

### **CLIENT**

응용프로그램이 호출되는 노드에서 인증이 발생하도록 지정합니다.

### **SERVER\_ENCRYPT**

목표 데이터베이스를 포함하는 데이터베이스 파티션 서버에서 인증이 발생하도록 지정하며, 사용자 ID 및 암호가 소스에서 암호화됩니다. 사용자 ID 및 암호는 소스에서 카탈로그되는 인증 유형에 의해 지정되는 대로 목표에서 암호 해독됩니다.

### **SERVER\_ENCRYPT\_AES**

목표 데이터베이스가 들어있는 데이터베이스 파티션 서버에서 인증이 발생하고, 사용자 ID 및 암호가 소스에서 AES(Advanced Encryption Standard) 암호화 알고리즘을 사용하여 암호화되고 목표에서 암호 해독되도록 지정합니다.

### **KERBEROS**

인증이 Kerberos 보안 메커니즘을 사용하여 발생하도록 지정합니다.

### **TARGET PRINCIPAL** *principalname*

목표 서버에 대한 완전한 Kerberos 핵심부 이름. 즉,

`userid@xxx.xxx.xxx.com` 또는 `domain#userid` 양식으로 DB2 서버 서비스의 로그인 어카운트입니다.

#### **DATA\_ENCRYPT**

목표 데이터베이스가 들어있는 노드에서 인증이 발생하고 연결이 데이터 암호화를 사용해야 하도록 지정합니다.

#### **GSSPLUGIN**

인증은 외부 GSS API 기반 플러그인 보안 메커니즘을 사용하여 발생하도록 지정합니다.

#### **WITH "comments"**

DB2 서버를 설명합니다. 네트워크 디렉토리에 등록된 서버를 설명하는 데 도움이 되는 모든 주석을 입력할 수 있습니다. 최대 길이는 30문자입니다. 캐리지 리턴 또는 줄 바꿈기 문자를 사용할 수 없습니다. 주석 텍스트는 큰따옴표로 묶어야 합니다.

#### **USER *username***

사용자의 LDAP 식별 이름(DN)을 지정합니다. LDAP 사용자 DN은 LDAP 디렉토리에 오브젝트를 작성하기 위한 충분한 권한이 있어야 합니다. 사용자의 LDAP DN이 지정되지 않은 경우 현재 로그인 사용자의 증명서가 사용됩니다. 사용자의 LDAP DN 및 암호가 `db2ldcfg`를 사용하여 지정된 경우, 사용자 이름 및 암호를 여기에 지정할 필요는 없습니다.

#### **PASSWORD *password***

어카운트 암호. 사용자의 LDAP DN 및 암호가 `db2ldcfg`를 사용하여 지정된 경우, 사용자 이름 및 암호를 여기에 지정할 필요는 없습니다.

### **사용 시 참고사항**

노드 이름이 지정되지 않는 경우 DB2는 현재 머신의 DB2 서버를 표시하는 LDAP의 첫 번째 노드를 사용합니다.

다음 경우에는 데이터베이스를 수동으로 등록(카탈로그)해야 할 수 있습니다.

- 데이터베이스 서버가 LDAP를 지원하지 않습니다. 관리자는 LDAP를 지원하는 클라이언트가 각 클라이언트 머신에서 로컬로 데이터베이스를 카탈로그할 필요 없이 데이터베이스에 액세스할 수 있도록 LDAP에서 각 데이터베이스를 수동으로 등록해야 합니다.
- 응용프로그램이 다른 이름을 사용하여 데이터베이스에 연결하려고 합니다. 이 경우 관리자는 다른 별명 이름을 사용하여 데이터베이스를 카탈로그할 수 있습니다.
- 데이터베이스가 호스트 또는 System i 데이터베이스 서버에 상주합니다. 이 경우 관리자는 LDAP에서 데이터베이스를 등록하고 **GWNODE** 매개변수를 통해 게이트웨이 노드를 지정할 수 있습니다.

- CREATE DATABASE IN LDAP 중에 데이터베이스 이름이 이미 LDAP에 존재합니다. 데이터베이스가 여전히 로컬 시스템에 존재하지만(또한 로컬 응용프로그램이 액세스할 수 있지만), LDAP의 기존 항목이 새 데이터베이스를 반영하도록 수정되지 않습니다. 이 경우 관리자는 다음을 수행할 수 있습니다.
  - LDAP에서 기존 데이터베이스 항목을 제거하고 수동으로 새 데이터베이스를 LDAP에 등록합니다.
  - 다른 별명 이름을 사용하여 LDAP에 새 데이터베이스를 등록합니다.



## 제 20 장 CATALOG LDAP NODE

LDAP(Lightweight Directory Access Protocol)에서 새 노드 항목을 카탈로그합니다.

### 권한 부여

없음

### 필수 연결

없음

### 명령 구문

```
►► CATALOG LDAP NODE nodename AS nodealias
└── USER username ──┬──┐
                    │   └── PASSWORD password ──┘
```

### 명령 매개변수

#### **NODE** *nodename*

DB2 서버의 LDAP 노드 이름을 지정합니다.

#### **AS** *nodealias*

LDAP 노드 항목에 대한 새 별명 이름을 지정합니다.

#### **USER** *username*

사용자의 LDAP 식별 이름(DN)을 지정합니다. LDAP 사용자 DN은 LDAP 디렉토리에 오브젝트를 작성하기 위한 충분한 권한이 있어야 합니다. 사용자의 LDAP DN이 지정되지 않은 경우 현재 로그인 사용자의 증명서가 사용됩니다.

#### **PASSWORD** *password*

여카운트 암호

### 사용 시 참고사항

CATALOG LDAP NODE 명령은 DB2 서버를 표시하는 노드에 대해 다른 별명 이름을 지정하는 데 사용됩니다.



---

## 제 21 장 CATALOG LOCAL NODE

동일한 머신에 있는 인스턴스에 대한 로컬 별명을 작성합니다. 사용자의 클라이언트에서 액세스할 동일한 워크스테이션에 둘 이상의 인스턴스가 있을 때 로컬 노드를 카탈로그해야 합니다. 프로세스간 통신(IPC)이 로컬 노드에 액세스하는 데 사용됩니다.

### 권한 부여

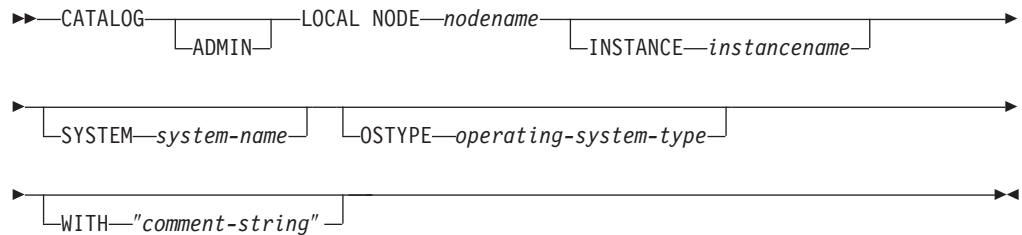
다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

#### ADMIN

로컬 Administration Server 노드가 카탈로그되도록 지정합니다.

#### INSTANCE *instancename*

액세스할 로컬 인스턴스의 이름.

#### SYSTEM *system-name*

서버 머신을 식별하는 데 사용되는 DB2 시스템 이름을 지정합니다.

#### OSTYPE *operating-system-type*

서버 머신의 운영 체제 유형을 지정합니다. 올바른 값은 AIX, WIN, HPUX, SUN, OS390, OS400, VM, VSE, SNI, SCO, LINUX 및 DYNIX입니다.

## 예:

워크스테이션 A에 두 서버 인스턴스 `inst1` 및 `inst2`가 있습니다. 단일 CLP 세션의 두 인스턴스 모두에서 데이터베이스를 작성하려면 다음의 일련의 명령을 발행하십시오 (DB2INSTANCE 환경 변수가 `inst1`로 설정된다고 가정).

1. `inst1`에 로컬 데이터베이스를 작성하십시오.

```
db2 create database mydb1
```

2. 이 워크스테이션의 다른 서버 인스턴스를 카탈로그하십시오.

```
db2 catalog local node mynode2 instance inst2
```

3. `mynode2`에 데이터베이스를 작성하십시오.

```
db2 attach to mynode2  
db2 create database mydb2
```

## 사용 시 참고사항

디렉토리 캐싱이 사용 가능한 경우 데이터베이스, 노드 및 DCS 디렉토리 파일이 메모리에 캐시됩니다. 응용프로그램의 디렉토리 캐시는 첫 번째 디렉토리 찾아보기 중에 작성됩니다. 캐시는 응용프로그램이 디렉토리 파일 중 하나를 수정할 때만 새로 고쳐지므로, 다른 응용프로그램이 작성한 디렉토리 변경은 응용프로그램이 재시작될 때까지 적용되지 않을 수 있습니다.

CLP의 디렉토리 캐시를 새로 고치려면 `TERMINATE`를 사용하십시오. DB2의 공유 캐시를 새로 고치려면 데이터베이스 관리 프로그램을 중지한 후(`db2stop`) 재시작하십시오(`db2start`). 다른 응용프로그램에 대한 디렉토리 캐시를 새로 고치려면 해당 응용프로그램을 중지한 후 재시작하십시오.

---

## 제 22 장 CATALOG NAMED PIPE NODE

노드 디렉토리에 Named Pipe 노드 항목을 추가합니다. Named Pipe는 리모트 노드에 액세스하는 데 사용됩니다.

이 명령은 Windows에서만 사용할 수 있습니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*

### 필수 연결

없음

### 명령 구문

```
▶▶ CATALOG [ADMIN] NPIPE NODE nodename REMOTE computername
▶▶ INSTANCE instancename [SYSTEM system-name]
▶▶ [OSTYPE operating-system-type] [WITH "comment-string"]
```

### 명령 매개변수

#### ADMIN

NPIPE Administration Server 노드가 카탈로그되도록 지정합니다.

#### REMOTE *computername*

목표 데이터베이스가 있는 노드의 컴퓨터 이름입니다. 최대 길이는 15문자입니다.

#### INSTANCE *instancename*

목표 데이터베이스가 있는 서버 인스턴스의 이름입니다. 리모트 노드와 통신하는 데 사용되는 리모트 Named Pipe의 이름과 동일합니다.

#### SYSTEM *system-name*

서버 머신을 식별하는 데 사용되는 DB2 시스템 이름을 지정합니다.

## **OSTYPE** *operating-system-type*

서버 머신의 운영 체제 유형을 지정합니다. 올바른 값은 AIX, WIN, HPUNIX, SUN, OS390, OS400, VM, VSE, SNI, SCO 및 LINUX입니다.

### **예:**

```
db2 catalog npipe node db2np1 remote nphost instance db2inst1
with "A remote named pipe node."
```

## **사용 시 참고사항**

데이터베이스 관리 프로그램은 첫 번째 노드가 카탈로그될 때(즉, 첫 번째 CATALOG...NODE 명령이 발행될 때) 노드 디렉토리를 작성합니다. Windows 클라이언트에서는 클라이언트가 설치되는 인스턴스 서브디렉토리에 노드 디렉토리를 저장하고 유지합니다. AIX 클라이언트에서는 DB2 설치 디렉토리에 노드 디렉토리를 작성합니다.

LIST NODE DIRECTORY 명령을 사용하여 로컬 노드 디렉토리의 콘텐츠를 나열하십시오.

디렉토리 캐싱이 사용 가능한 경우(GET DATABASE MANAGER CONFIGURATION 명령의 **dir\_cache** 구성 매개변수 참조), 데이터베이스, 노드 및 DCS 디렉토리 파일이 메모리에 캐시됩니다. 응용프로그램의 디렉토리 캐시는 첫 번째 디렉토리 찾아보기 중에 작성됩니다. 캐시는 응용프로그램이 디렉토리 파일 중 하나를 수정할 때만 새로 고쳐지므로, 다른 응용프로그램이 작성한 디렉토리 변경은 응용프로그램이 재시작될 때까지 적용되지 않을 수 있습니다.

CLP의 디렉토리 캐시를 새로 고치려면 TERMINATE 명령을 사용하십시오. DB2의 공유 캐시를 새로 고치려면 데이터베이스 관리 프로그램을 중지한 후(db2stop) 재시작하십시오(db2start). 다른 응용프로그램에 대한 디렉토리 캐시를 새로 고치려면 해당 응용프로그램을 중지한 후 재시작하십시오.

---

## 제 23 장 CATALOG ODBC DATA SOURCE

사용자 또는 시스템 ODBC 데이터 소스를 카탈로그합니다.

ODBC(Open Database Connectivity) 용어에서 데이터 소스는 특정 데이터베이스 또는 파일 시스템에 대한 사용자 정의 이름입니다. 해당 이름이 ODBC API를 통해 데이터베이스 또는 파일 시스템에 액세스하는 데 사용됩니다. 사용자 또는 시스템 데이터 소스를 카탈로그할 수 있습니다. 사용자 데이터 소스는 이를 카탈로그한 사용자만 볼 수 있는 반면, 시스템 데이터 소스는 다른 모든 사용자가 보고 사용할 수 있습니다.

이 명령은 Windows 플랫폼에서만 사용할 수 있습니다.

### 권한 부여

없음

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

**USER** 사용자 데이터 소스를 카탈로그합니다. 이것은 키워드를 지정하지 않는 경우 디폴트입니다.

#### SYSTEM

시스템 데이터 소스를 카탈로그합니다.

#### ODBC DATA SOURCE *data-source-name*

카탈로그될 데이터 소스의 이름을 지정합니다. 데이터 소스의 이름 및 데이터베이스의 이름은 동일해야 합니다. 따라서, 데이터 소스의 이름은 데이터베이스 이름의 최대 길이로 제한됩니다.

#### ALL DATA SOURCES

모든 로컬 데이터베이스 별명을 ODBC 데이터 소스(DSN)로 카탈로그하도록 지정합니다.

## 사용 시 참고사항

Microsoft Windows Vista 이상 버전에서는 전체 관리자 특권을 갖고 실행 중인 DB2 명령 창에서 CATALOG SYSTEM ODBC DATA SOURCE 명령을 실행해야 합니다.

ALL DATA SOURCES 매개변수를 지정하면 dbalias 매개변수를 로컬 데이터베이스 디렉토리에 있는 데이터베이스의 별명과 일치하는 값으로 설정한 기존 ODBC DSN이 갱신되지 않습니다.

### 예 1

"MyProdDatabase"라는 기존 ODBC DSN이 있다고 가정합니다. dbalias 매개변수는 "PRODDB"로 설정됩니다. 또한 로컬 디렉토리에 별명이 "PRODDB"인 데이터베이스도 있다고 가정하십시오. DSN이 데이터베이스 별명과 일치하지 않기 때문에 CATALOG ODBC DATA SOURCE myproddb 명령 또는 CATALOG ALL DATA SOURCES 명령을 실행해도 "MyProdDatabase" DSN이 변경되지 않습니다. 대신 dbalias가 "PRODDB"로 설정된 "PRODDB"에 대한 ODBC DSN 항목이 작성됩니다. 데이터베이스 별명과 동일한 이름을 갖는 기존 ODBC DSN이 있는 경우 기존 ODBC DSN의 dbalias 매개변수가 데이터베이스 별명으로 갱신됩니다. 모든 연관된 CLI 매개변수 및 값은 변경되지 않은 채로 있습니다.

### 예 2:

dbalias 매개변수가 "salesdb"로 설정된 "MYDB"라는 기존 DSN이 있다고 가정합니다. "MYDB"라는 로컬 디렉토리에 데이터베이스가 있는 경우 CATALOG ODBC DATA SOURCE mydb 명령 또는 CATALOG ALL DATA SOURCES 명령이 DSN의 dbalias 매개변수를 "MYDB"로 변경합니다.



## 제 24 장 CATALOG TCPIP/TCPIP4/TCPIP6 NODE

TCP/IP 데이터베이스 파티션 서버 항목을 노드 디렉토리에 추가합니다. TCP/IP 통신 프로토콜은 리모트 데이터베이스 파티션 서버에 액세스하기 위해 사용됩니다. CATALOG TCPIP/TCPIP4/TCPIP6 NODE 명령은 클라이언트에서 실행됩니다.

### 권한 부여

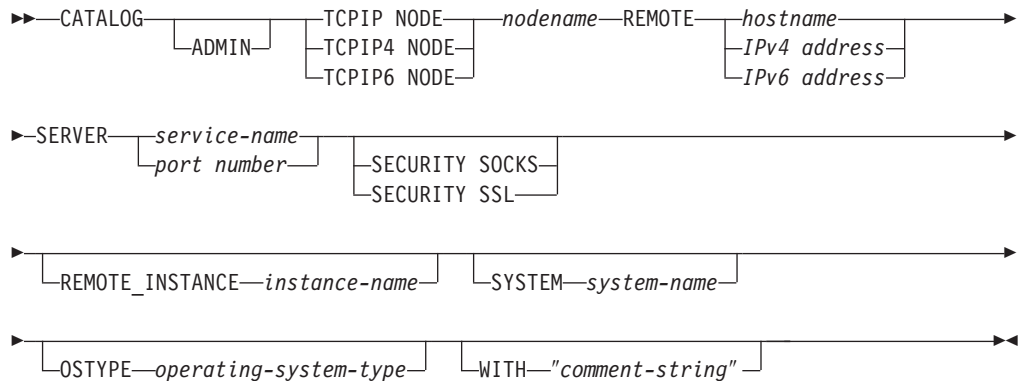
다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*

### 필수 연결

없음. 디렉토리 조작은 로컬 디렉토리에만 영향을 줍니다.

### 명령 구문



### 명령 매개변수

#### ADMIN

TCP/IP 관리 서버 노드가 카탈로그됨을 지정합니다. 이 매개변수는 SECURITY SOCKS 매개변수가 지정된 경우 지정될 수 없습니다.

#### TCPIP NODE *nodename*

TCPIP, TCPIP4 또는 TCPIP6 데이터베이스 파티션 서버는 카탈로그할 데이터베이스를 포함하는 머신에 대해 설정할 수 있는 로컬 별칭을 나타냅니다. IPv4 IP 주소를 지정할 때 TCPIP4만 지정하고 IPv6 IP 주소를 지정할 때 TCPIP6만 지정하십시오.

#### REMOTE *hostname* | *IPv4 address* | *IPv6 address*

목표 데이터베이스가 있는 노드의 호스트 이름 또는 IP 주소. *IP address*는 IPv4

또는 IPv6 주소가 될 수 있습니다. 호스트 이름은 TCP/IP 네트워크에 알려진 데이터베이스 파티션 서버의 이름입니다. 호스트 이름의 최대 길이는 255자입니다.

#### **SERVER** *service-name* | *port number*

서버 데이터베이스 관리 프로그램 인스턴스의 서비스 이름 또는 포트 번호를 지정합니다. 최대 길이는 14자입니다. 이 매개변수는 대소문자가 구분됩니다.

서비스 이름을 지정한 경우 클라이언트의 *services* 파일은 서비스 이름을 포트 번호에 매핑하는 데 사용됩니다. 서비스 이름을 서버의 데이터베이스 관리 프로그램 구성 파일에 지정한 경우 서버의 *services* 파일은 이 서비스 이름을 포트 번호에 매핑하는 데 사용됩니다. 클라이언트 및 서버의 포트 번호는 일치해야 합니다.

서비스 이름 대신 포트 번호를 서버의 데이터베이스 관리 프로그램 구성 파일에 지정할 수 있지만 권장되지 않습니다. 포트 번호를 지정한 경우 서비스 이름은 로컬 *services* 파일에서 지정하지 않아도 됩니다.

이 매개변수는 ADMIN 노드의 경우 지정할 필요가 없지만 ADMIN이 아닌 노드에 대해서는 필수입니다. ADMIN 노드에서의 값은 항상 523입니다.

#### **SECURITY SOCKS**

노드가 SOCKS 사용 가능 상태가 됨을 지정합니다. 이 매개변수는 IPv4에서만 지원됩니다. CATALOG TCP/IP NODE가 사용되고 SECURITY SOCKS를 지정한 경우, DB2 데이터베이스 제품은 IPv4를 사용하여 연결을 설정합니다. ADMIN 매개변수를 지정한 경우 이 매개변수를 지정할 수 없습니다.

다음 환경 변수는 필수이므로 반드시 SOCKS를 사용하도록 설정해야 합니다.

##### **SOCKS\_NS**

SOCKS 서버의 호스트 주소를 분석하는 도메인 이름 서버. 호스트 이름이나 IPv4 주소여야 합니다.

##### **SOCKS\_SERVER**

SOCKS 서버의 완전한 호스트 이름 또는 IPv4 주소. SOCKS가 IBM Data Server Client가 완전한 호스트 이름을 분석할 수 없도록 지정된 경우 SOCKS는 IPv4 주소가 입력된 것으로 가정합니다.

다음 조건 중 하나가 참이어야 합니다.

- SOCKS 서버가 도메인 이름 서버를 통해 도달 가능합니다.
- 호스트 이름이 *hosts* 파일에 나열되어 있습니다. 이 파일의 위치는 TCP/IP 문서에 설명됩니다.
- IPv4 주소가 지정되어 있습니다.

이 명령이 *db2start* 다음에 발행되는 경우, 명령이 적용되도록 하려면 *TERMINATE* 명령을 발행해야 합니다.

## SECURITY SSL

노드가 SSL 사용 가능 상태임을 지정합니다. ADMIN 매개변수도 지정하는 경우 SECURITY SSL 절을 지정할 수 없습니다.

## REMOTE\_INSTANCE *instance-name*

데이터베이스가 상주하고, 접속 또는 연결되는 서버 인스턴스의 이름을 지정합니다.

## SYSTEM *system-name*

서버 머신을 식별하는 데 사용되는 DB2 시스템 이름을 지정합니다. 실제 머신, 서버 시스템 또는 워크스테이션의 이름입니다.

## OSTYPE *operating-system-type*

서버 머신의 운영 체제 유형을 지정합니다. 유효한 값은 AIX, WIN, HPUX, SUN, OS390, OS400, VM, VSE 및 LINUX입니다.

## 예:

CATALOG TCPIP NODE 명령을 사용하여 호스트 이름을 지정하려면 다음을 발행하십시오.

```
db2 catalog tcpip node db2tcp1 remote hostname server db2inst1
with "Look up IPv4 or IPv6 address from hostname"
```

CATALOG TCPIP4 NODE 명령을 사용하여 IPv4 주소를 지정하려면 다음을 발행하십시오.

```
db2 catalog tcpip4 node db2tcp2 remote 192.0.32.67 server db2inst1
with "Look up IPv4 address from 192.0.32.67"
```

이 예는 IPv4 주소를 지정합니다. CATALOG TCPIP4 NODE 명령에서 IPv6 주소를 지정하면 안됩니다. IPv6 주소를 지정하면 카탈로그는 실패하지 않지만 카탈로그 중에 유효하지 않은 주소가 지정되었으므로 그 뒤의 접속 또는 연결이 실패합니다.

CATALOG TCPIP6 NODE 명령을 사용하여 IPv6 주소를 지정하려면 다음을 발행하십시오.

```
db2 catalog tcpip6 node db2tcp3 1080:0:0:0:8:800:200C:417A server 50000
with "Look up IPv6 address from 1080:0:0:0:8:800:200C:417A"
```

이 예는 server의 IPv6 주소 및 포트 번호를 지정합니다. CATALOG TCPIP4 NODE 명령에서 IPv6 주소를 지정하면 안됩니다. IPv6 주소를 지정하면 카탈로그는 실패하지 않지만 카탈로그 중에 유효하지 않은 주소가 지정되었으므로 그 뒤의 접속 또는 연결이 실패합니다.

다음 예는 SSL 연결에 대해 노드를 카탈로그합니다(서버 호스트 이름은 *hostname*이고, *ssl\_port*는 데이터베이스 서버가 SSL 프로토콜을 사용하는 리모트 클라이언트의 통신을 기다리는 포트 번호임).

```
db2 catalog tcpip node db2tcp4 remote hostname server ssl_port
```

## 사용 시 참고사항

데이터베이스 관리 프로그램은 첫 번째 노드가 카탈로그될 때(즉, 첫 번째 CATALOG...NODE 명령이 발행될 때) 노드 디렉토리를 작성합니다. Windows 클라이언트에서는 클라이언트가 설치되는 인스턴스 서브디렉토리에 노드 디렉토리를 저장하고 유지합니다. AIX 클라이언트에서는 DB2 설치 디렉토리에 노드 디렉토리를 작성합니다.

LIST NODE DIRECTORY 명령을 사용하여 로컬 노드 디렉토리의 콘텐츠를 나열하십시오.

디렉토리 캐싱이 사용 가능한 경우 데이터베이스, 노드 및 DCS 디렉토리 파일이 메모리에 캐시됩니다. 응용프로그램의 디렉토리 캐시는 첫 번째 디렉토리 찾아보기 중에 작성됩니다. 캐시는 응용프로그램이 디렉토리 파일 중 하나를 수정할 때만 새로 고쳐지므로, 다른 응용프로그램이 작성한 디렉토리 변경은 응용프로그램이 재시작될 때까지 적용되지 않을 수 있습니다.

CLP의 디렉토리 캐시를 새로 고치려면 TERMINATE 명령을 사용하십시오. DB2의 공유 캐시를 새로 고치려면 데이터베이스 관리 프로그램을 중지한 후(db2stop) 재시작하십시오(db2start). 다른 응용프로그램의 디렉토리 캐시를 새로 고치려면, 해당 응용프로그램을 중지한 후 재시작하십시오.

DB2가 IPv6에서 대기(listen)하도록 하려면 운영 체제 및 서버가 먼저 IPv6에 맞게 구성되어야 합니다. 시스템 관리자에게 IPv6 TCPIP 노드를 카탈로그하기 전에 이 구성이 완료되었는지 확인하도록 요청하십시오. IPv4가 구성된 IPv6으로 업그레이드에 따라 AIX 5.3에서 수행할 수 있는 방법을 참조하십시오.

---

## 제 25 장 CHANGE DATABASE COMMENT

시스템 데이터베이스 디렉토리 또는 로컬 데이터베이스 디렉토리에서 데이터베이스 주석을 변경합니다. 새 주석 텍스트는 주석과 현재 연관된 텍스트를 대체할 수 있습니다.

### 범위

이 명령은 명령이 실행되는 데이터베이스 파티션에만 영향을 미칩니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*

### 필수 연결

없음

### 명령 구문

```
▶ CHANGE DATABASE database-alias COMMENT ON path | drive WITH "comment-string" ▶
```

### 명령 매개변수

#### **DATABASE** *database-alias*

주석이 변경되는 데이터베이스의 별명을 지정합니다. 시스템 데이터베이스 디렉토리에서 주석을 변경하려면, 데이터베이스의 별명을 지정하십시오. 로컬 데이터베이스 디렉토리에서 주석을 변경하려면, 데이터베이스가 있는 경로를 지정하고 (*path* 매개변수 사용) 데이터베이스의 이름(별명이 아님)을 입력하십시오.

#### **ON** *path* | *drive*

데이터베이스가 있는 경로를 지정하며 로컬 데이터베이스 디렉토리에서 주석을 변경합니다. 경로가 지정되지 않으면, 시스템 데이터베이스 디렉토리의 항목에 대한 데이터베이스 주석이 변경됩니다. Windows 운영 체제에서, 데이터베이스가 있는 드라이브명을 대신 지정할 수 있습니다(특정 경로가 아닌 드라이브에서 작성된 경우).

#### **WITH** "*comment-string*"

시스템 데이터베이스 디렉토리 또는 로컬 데이터베이스 디렉토리에서 항목을 설

명합니다. 카탈로그된 데이터베이스를 설명하도록 돕는 주석을 입력할 수 있습니다. 주석 문자열의 최대 길이는 30문자입니다. 캐리지 리턴이나 줄 바꾸기 문자는 허용되지 않습니다. 주석 텍스트는 큰따옴표로 묶어야 합니다.

## 예:

다음 예는 "Test 2 - Holding"에서 "Test 2 - Add employee inf rows"로 SAMPLE 데이터베이스의 시스템 데이터베이스 디렉토리 주석 텍스트를 변경합니다.

```
db2 change database sample comment
with "Test 2 - Add employee inf rows"
```

## 사용 시 참고사항

새 주석 텍스트가 기존 텍스트를 교체합니다. 정보를 추가하려면, 이전 텍스트 다음에 새 텍스트를 입력하십시오.

데이터베이스 별명과 연관된 항목의 주석만 수정됩니다. 동일한 데이터베이스 이름을 가지지만 다른 별명을 갖는 다른 항목은 영향을 받지 않습니다.

경로가 지정된 경우, 데이터베이스 별명은 로컬 데이터베이스 디렉토리에 카탈로그되어야 합니다. 경로가 지정되지 않은 경우, 데이터베이스 별명은 시스템 데이터베이스 디렉토리에 카탈로그되어야 합니다.

---

## 제 26 장 CHANGE ISOLATION LEVEL

데이터베이스가 액세스되는 동안 DB2가 다른 프로세스에서 데이터를 분리하는 방식을 변경합니다.

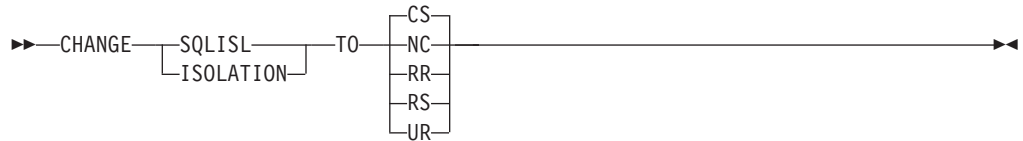
### 권한 부여

없음

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

#### TO

**CS** 커서 안정성을 분리 레벨로 지정합니다.

**NC** 커밋 없음을 분리 레벨로 지정합니다. DB2에 의해 지원되지 않습니다.

**RR** 반복 읽기를 분리 레벨로 지정합니다.

**RS** 읽기 안정성을 분리 레벨로 지정합니다.

**UR** 커밋되지 않은 읽기를 분리 레벨로 지정합니다.

### 사용 시 참고사항

DB2는 데이터베이스에서 데이터 무결성을 유지하기 위해 분리 레벨을 사용합니다. 분리 레벨은 현재 실행 중인 다른 응용프로그램 프로세스에 의해 작성된 변경사항으로부터 응용프로그램 프로세스가 분리되는 등급을 정의합니다.

선택된 분리 레벨을 데이터베이스에서 지원하지 않는 경우, 연결 시 자동으로 지원되는 레벨로 자동 확대됩니다.

유형 1 연결로 데이터베이스에 연결되어 있는 동안 분리 레벨 변경은 허용되지 않습니다. 분리 레벨이 변경되기 전에 백엔드 프로세스를 종료해야 합니다.

```
db2 terminate
db2 change isolation to ur
db2 connect to sample
```

유형 2 연결을 사용하여 변경이 허용되지만, 주의하여야 합니다. 왜냐하면 동일한 명령 행 처리기 백엔드 프로세스에서 작성한 모든 연결에 변경이 적용되기 때문입니다. 사용하는 어떤 연결된 데이터베이스에 어떤 분리 레벨을 적용하는지 확인해야 합니다.

다음 예에서, 사용자는 SAMPLE 데이터베이스의 작성을 따르는 DB2 대화식 모드에 있습니다.

```
update command options using c off
catalog db sample as sample2

set client connect 2

connect to sample
connect to sample2

change isolation to cs
set connection sample
declare c1 cursor for select * from org
open c1
fetch c1 for 3 rows

change isolation to rr
fetch c1 for 2 rows
```

c1은 이 분리 레벨의 준비된 상태에 있지 않으므로 SQL0514N 오류가 발생합니다.

```
change isolation to cs
set connection sample2
fetch c1 for 2 rows
```

c1은 이 데이터베이스의 준비된 상태에 있지 않으므로 SQL0514N 오류가 발생합니다.

```
declare c1 cursor for select division from org
```

커서 c1이 이미 선언되어 열려 있으므로 DB21029E 오류가 발생합니다.

```
set connection sample
fetch c1 for 2 rows
```

원래 데이터베이스(SAMPLE)가 원래 분리 레벨(CS)에서 사용되었으므로 작동합니다.



---

## 제 27 장 COMPLETE XMLSCHEMA

이 명령은 XML 스키마 저장소(XSR)에 XML 스키마를 등록하는 프로세스를 완료합니다.

### 권한 부여

- 사용자 ID는 카탈로그 뷰 SYSCAT.XSROBJECTS에 기록된 대로 XSR 오브젝트의 소유자여야 합니다.

### 필수 연결

데이터베이스

### 명령 구문

```
▶▶ COMPLETE XMLSCHEMA relational-identifier [WITH schema-properties-URI]  
[ENABLE DECOMPOSITION]
```

### 설명

#### *relational-identifier*

이전에 REGISTER XMLSCHEMA 명령으로 등록된 XML 스키마의 관계형 이름을 지정합니다. 관계형 이름은 SQL 스키마와 XML 스키마 이름으로 구성되는 두 파트 SQL ID로 지정될 수 있으며 형식은 *SQLschema.name*입니다. 스키마를 지정하지 않는 경우 디폴트 SQL 스키마(CURRENT SCHEMA 특수 레지스터에 정의된)가 사용됩니다.

#### WITH *schema-properties-URI*

XML 스키마의 등록 정보 문서 URI(Uniform Resource Identifier)를 지정합니다. 파일 스킴 URI로 지정된 로컬 파일만 지원됩니다. 스키마 등록 정보 문서는 XML 스키마 등록의 완료 단계에서만 지정할 수 있습니다.

#### ENABLE DECOMPOSITION

XML 인스턴스 문서 분석에 스키마를 사용할 수 있음을 표시합니다.

### 예 :

```
COMPLETE XMLSCHEMA user1.POschema WITH 'file:///c:/TEMP/schemaProp.xml'
```

## 사용 시 참고사항

XML 스키마 등록 프로세스가 완료될 때까지 유효성 확인 또는 어노테이션에 XML 스키마를 참조하거나 사용할 수 없습니다. 이 명령은 REGISTER XMLSCHEMA 명령으로 시작된 XML 스키마에 대한 XML 스키마 등록 프로세스를 완료합니다.

---

## 제 28 장 CREATE DATABASE

CREATE DATABASE 명령은 선택적 사용자 정의 조합 시퀀스로 새 데이터베이스를 초기화하며, 세 개의 초기 테이블 스페이스를 작성하며, 시스템 테이블을 작성하며, 복구 로그 파일을 할당합니다. 새 데이터베이스를 초기화하면, AUTOCONFIGURE 명령이 디폴트로 발행됩니다.

주: 인스턴스 및 데이터베이스 디렉토리가 DB2 데이터베이스 관리 프로그램에 의해 작성되면, 권한이 정확하고 변경되지 않아야 합니다.

CREATE DATABASE 명령이 발행되면, 구성 어드바이저도 자동으로 실행합니다. 이것은 시스템 자원에 따라 데이터베이스 구성 매개변수가 자동으로 조정됨을 의미합니다. 또한 자동화된 Runstats가 사용 가능합니다. 데이터베이스 작성 시 구성 어드바이저를 사용 안하려면, *db2\_enable\_autoconfig\_default* 레지스트리 변수를 참조하십시오. 자동화된 Runstats를 사용 안하려면, *auto\_runstats* 데이터베이스 구성 매개변수를 참조하십시오.

단일 파티션 데이터베이스의 적응 자체 성능 조정 메모리도 디폴트로 사용 가능합니다. 디폴트로 적응 자체 성능 조정 메모리를 사용 안하려면, *self\_tuning\_mem* 데이터베이스 구성 매개변수(*self\_tuning\_mem* - 자체 성능 조정 메모리 구성 매개변수 참조)를 참조하십시오. 다중 파티션 데이터베이스의 경우, 적응 자체 성능 조정 메모리는 디폴트로 사용하지 않습니다.

CREATE DATABASE 명령에서 코드 세트가 지정되지 않으면, 허용되는 조합은 다음과 같습니다. SYSTEM, IDENTITY\_16BIT, UCA400\_NO, UCA400\_LSK, UCA400\_LTH, *language-aware-collation* 및 *locale-aware-collation*(SQLCODE -1083). 데이터베이스의 디폴트 코드 세트는 UTF-8입니다. 특정 코드 세트와 지역이 데이터베이스에 필요한 경우, 원하는 코드 세트와 지역이 CREATE DATABASE 명령에 지정되어야 합니다.

이 명령은 클라이언트에서 유효하지 않습니다.

### 범위

파티션된 데이터베이스 환경에서, 이 명령은 *db2nodes.cfg* 파일에 나열된 모든 데이터베이스 파티션에 영향을 미칩니다.

이 명령이 발행된 데이터베이스 파티션은 새 데이터베이스의 카탈로그 데이터베이스 파티션이 됩니다.

## 권한 부여

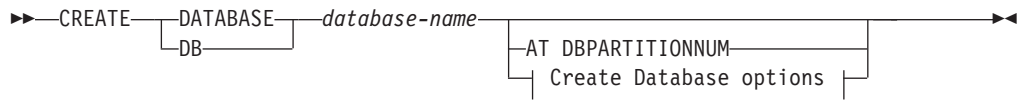
다음 중 하나여야 합니다.

- *sysadm*
- *sysctrl*

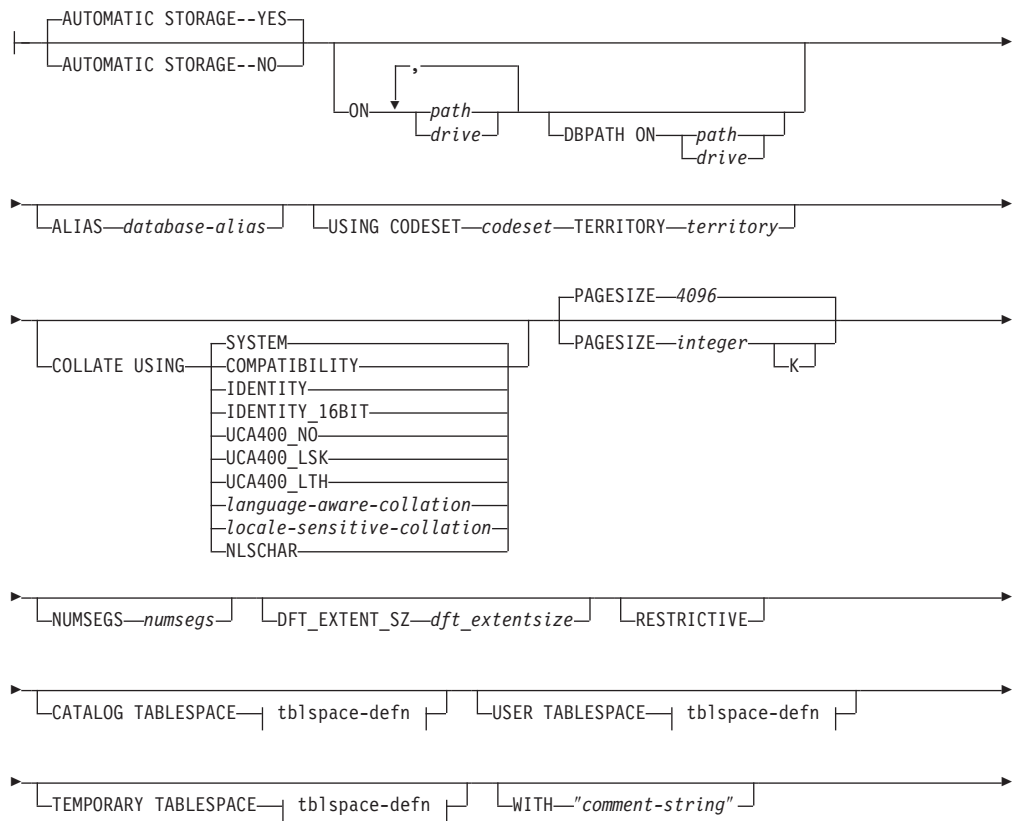
## 필수 연결

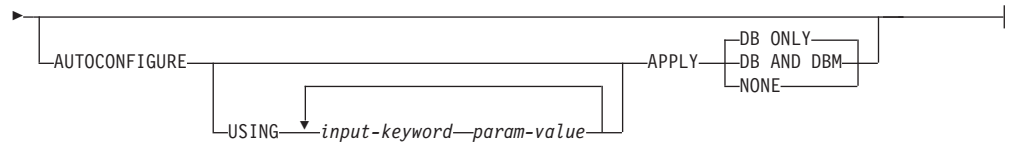
인스턴스. 다른 (리모트) 데이터베이스 파티션 서버에서 데이터베이스를 작성하려면, 우선 해당 서버에 접속해야 합니다. 처리하는 동안 이 명령에 의해 임시로 데이터베이스 연결이 설정됩니다.

## 명령 구문

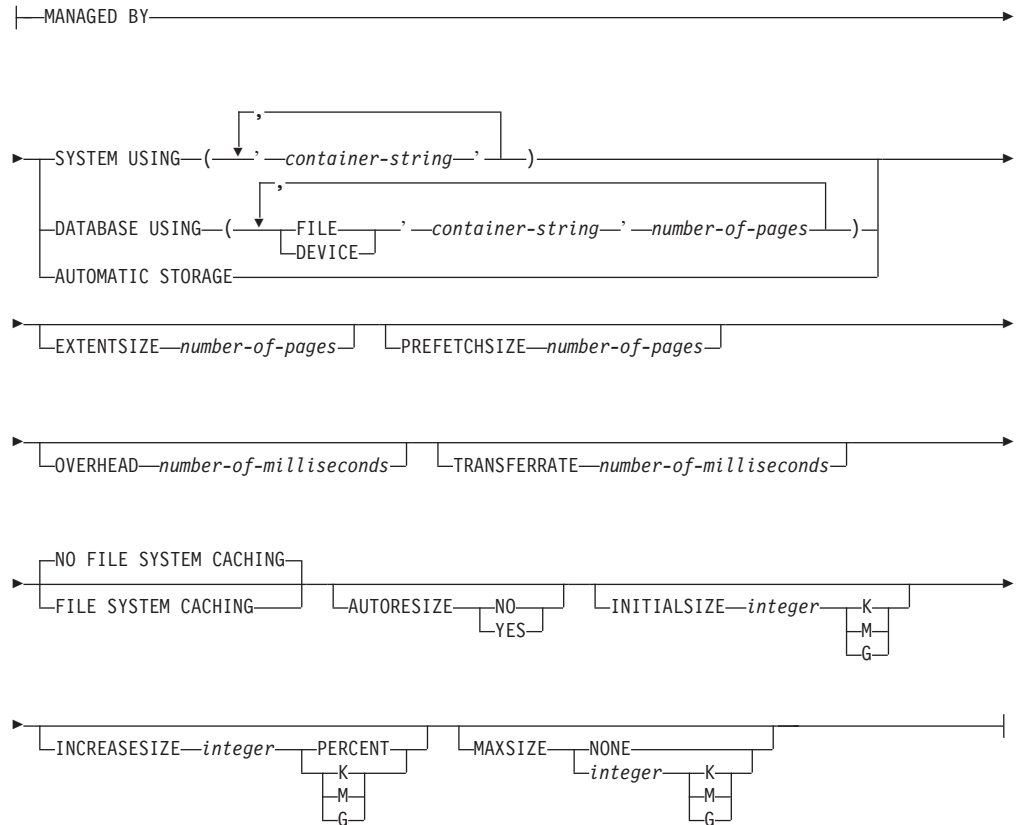


### Create Database options:





**tblspace-defn:**



**주:**

1. 코드 세트 및 지역 값의 조합이 유효해야 합니다.
2. 모든 조합 시퀀스가 모든 코드 세트 및 지역 조합에서 유효하지는 않습니다.
3. CREATE DATABASE에서 지정된 테이블 스페이스 정의는 데이터베이스가 작성 중인 모든 데이터베이스 파티션에 적용합니다. 각 데이터베이스 파티션마다 개별적으로 해당 정의를 지정할 수 없습니다. 테이블 스페이스 정의가 특정 데이터베이스 파티션에서 다르게 작성되면, CREATE TABLESPACE문을 사용해야 합니다.

테이블 스페이스에 대한 컨테이너 정의 시, \$N을 사용할 수 있습니다. 컨테이너가 실제로 작성되면 \$N은 데이터베이스 파티션 번호로 교체됩니다. 다중 논리적 파티션 데이터베이스에서 컨테이너를 지정하려는 경우 필요합니다.

4. AUTOCONFIGURE 옵션은 *sysadm* 권한을 필요로 합니다.

## 명령 매개변수

### **DATABASE** *database-name*

새 데이터베이스에 지정되는 이름. 이것은 로컬 데이터베이스 디렉토리 또는 시스템 데이터베이스 디렉토리 중 하나로 임의의 다른 데이터베이스에서 데이터베이스를 구별짓는 고유 이름이어야 합니다. 이 이름은 데이터베이스의 이름 지정 규칙과 일치해야 합니다. 특히, 이름에 공백 문자가 포함되지 않아야 합니다.

### **AT DBPARTITIONNUM**

명령을 발행하는 데이터베이스 파티션에만 데이터베이스를 작성하도록 지정합니다. 새 데이터베이스 작성 시 이 옵션을 지정하지 않습니다. 손상되어 삭제한 데이터베이스 파티션을 재작성할 때 사용할 수 있습니다. AT DBPARTITIONNUM 옵션과 함께 CREATE DATABASE 명령을 사용한 이후에, 이 데이터베이스 파티션에서의 데이터베이스는 리스토어 보류 상태입니다. 이 데이터베이스 파티션 서버에서 즉시 데이터베이스를 리스토어해야 합니다. 이 매개변수는 일반 용도가 아닙니다. 예를 들어, 데이터베이스 파티션 서버에서의 데이터베이스 파티션이 손상되어 재작성해야 하는 경우 RESTORE DATABASE 명령과 함께 사용되어야 합니다. 이 매개변수의 부적절한 사용은 시스템에서 불일치를 유발할 수 있으므로, 주의해야 합니다.

이 변수가 삭제된 데이터베이스 파티션을 재작성하기 위해 사용되는 경우(손상되었기 때문에), 이 데이터베이스 파티션의 데이터베이스는 리스토어 보류 상태가 됩니다. 데이터베이스 파티션을 재작성한 이후, 즉시 데이터베이스를 이 데이터베이스 파티션에 리스토어해야 합니다.

### **AUTOMATIC STORAGE NO | YES**

데이터베이스의 자동 스토리지를 명시적으로 사용 안하거나 사용 가능하게 함을 지정합니다. 디폴트값은 YES입니다. AUTOMATIC STORAGE 절이 지정되지 않으면, 디폴트로 자동 스토리지가 내재적으로 사용 가능합니다.

**NO** 데이터베이스의 자동 스토리지가 사용 가능하지 않습니다.

**YES** 데이터베이스의 자동 스토리지가 사용 가능합니다.

### **ON** *path* 또는 *drive*

이 옵션의 의미는 AUTOMATIC STORAGE 옵션의 값에 따릅니다.

- AUTOMATIC STORAGE NO가 지정되면, 데이터베이스의 자동 스토리지를 사용하지 않습니다. 이 경우, 오직 하나의 경로가 ON 옵션의 파트로 포함될 수 있으며, 데이터베이스를 작성하기 위해 경로를 지정합니다. 경로가 지정되지 않으면, 데이터베이스 관리 프로그램 구성 파일에 지정된 디폴트 데이터베이스 경로에 데이터베이스가 작성됩니다(**dftdbpath** 매개변수). 이 동작은 DB2® Universal Database™ 버전 8.2 이하의 동작과 일치합니다.

- 그렇지 않은 경우, 디폴트로 데이터베이스의 자동 스토리지가 사용 가능합니다. 이 경우, 다중 경로가 여기에 나열되며 쉼표로 구분됩니다. 이것은 스토리지 경로로 참조되며 자동 스토리지 테이블 스페이스의 테이블 스페이스 컨테이너를 보유하기 위해 사용됩니다. 다중 파티션 데이터베이스의 경우 동일한 스토리지 경로가 모든 파티션에서 사용됩니다.

DBPATH ON 옵션은 데이터베이스를 작성하기 위한 경로에 지정합니다. DBPATH ON 옵션이 지정되지 않으면, ON 옵션에 나열된 최초 경로에 데이터베이스가 작성됩니다. 경로가 ON 옵션으로 지정되지 않으면, 데이터베이스 관리 프로그램 구성 파일에 지정된 디폴트 데이터베이스 경로에 데이터베이스가 작성됩니다(`dfidbpath` 매개변수). 이것은 데이터베이스와 연관된 단일 스토리지 경로의 위치로 사용됩니다.

데이터베이스 경로는 계층 디렉토리 구조가 작성된 위치입니다. 구조는 데이터베이스 조작에 필요한 다음 파일을 보유합니다.

- 버퍼 풀 정보
- 테이블 스페이스 정보
- 스토리지 경로 정보
- 데이터베이스 구성 정보
- 백업, 리스토어, 테이블 로딩, 테이블 재구성, 테이블 스페이스의 변경 및 다른 데이터베이스 변경사항에 대한 실행기록 파일 정보
- 사용 중인 로그에 대한 정보가 있는 로그 제어 파일

DBPATH ON 옵션은 데이터베이스 데이터가 보존된 스토리지 경로에서 분리된 디렉토리에 해당 파일과 정보를 위치시키기 위해 사용될 수 있습니다. 데이터베이스 데이터에서 분리된 데이터베이스 정보를 보존하기 위해 자동 스토리지가 사용 가능할 때 DBPATH ON 옵션을 사용하는 것이 좋습니다.

경로의 최대 길이는 175문자입니다.

MPP 시스템의 경우, NFS 마운트 디렉토리에서 데이터베이스를 작성하지 말아야 합니다. 경로가 지정되지 않으면, `dfidbpath` 데이터베이스 관리 프로그램 구성 매개변수가 NFS 마운트 경로에 설정되지 않습니다(예를 들어, UNIX 기반 시스템에서 인스턴스 소유자의 \$HOME 디렉토리를 지정하지 않아야 함). MPP 시스템에서 이 명령에 지정되는 경로는 상대적 경로일 수 없습니다. 또한, ON 옵션의 파트로 지정되는 모든 경로는 모든 데이터베이스 파티션에 존재해야 합니다.

주어진 데이터베이스 경로 또는 스토리지 경로가 존재해야 하며 각 데이터베이스 파티션에서 액세스 가능해야 합니다.

#### **DBPATH ON** *path* 또는 *drive*

자동 스토리지가 사용 가능하면, DBPATH ON 옵션은 데이터베이스를 작성하

기 위한 경로에 지정합니다. 자동 스토리지가 사용 가능하며 DBPATH ON 옵션이 지정되지 않으면, ON 옵션에 나열된 최초 경로에 데이터베이스가 작성됩니다.

데이터베이스 경로의 최대 길이는 215문자이며 스토리지 경로의 최대 길이는 175 문자입니다.

#### **ALIAS** *database-alias*

시스템 데이터베이스 디렉토리에 있는 데이터베이스의 별명. 별명을 제공하지 않으면, 지정된 데이터베이스 이름이 사용됩니다.

#### **USING CODESET** *codeset*

이 데이터베이스에 입력되는 데이터에 사용되는 코드 세트를 지정합니다. 데이터베이스를 작성한 후에, 지정된 코드 세트를 변경할 수 없습니다.

#### **TERRITORY** *territory*

이 데이터베이스에 입력되는 데이터에 사용되는 지역 ID 또는 로케일 ID를 지정합니다. 데이터베이스를 작성한 후에, 지정된 지역을 변경할 수 없습니다. 코드 세트 및 지역 또는 로케일 값의 조합이 유효해야 합니다.

#### **COLLATE USING**

데이터베이스에 사용되는 조합 시퀀스의 유형을 식별합니다. 데이터베이스가 작성되면, 조합 시퀀스를 변경할 수 없습니다.

유니코드 데이터베이스에서, COLLATE USING 절에 지정된 조합과 무관하게, 카탈로그 테이블 및 뷰는 IDENTITY 조합으로 작성됩니다. 유니코드가 아닌 데이터베이스에서, 카탈로그 테이블 및 뷰는 데이터베이스 조합으로 작성됩니다.

#### **COMPATIBILITY**

DB2 버전 2 조합 시퀀스. 일부 조합 테이블이 확장되었습니다. 이 옵션은 이전 버전의 해당 테이블이 사용됨을 지정합니다.

#### **IDENTITY**

문자열이 바이트 단위로 비교되는 ID 조합 시퀀스. 이것은 유니코드 데이터베이스의 디폴트입니다.

#### **IDENTITY\_16BIT**

유니코드 협회 웹 사이트([www.unicode.org](http://www.unicode.org))에서 사용 가능한 유니코드 기술 보고서 #26에서 지정한 대로의 CESU-8(Compatibility Encoding Scheme for UTF-16: 8-Bit) 조합 시퀀스. 이 옵션은 유니코드 데이터베이스 작성 시에만 지정될 수 있습니다.

#### **UCA400\_NO**

정규화가 내재적으로 ON으로 설정된 유니코드 표준 버전 4.0.0에 근거한 UCA(Unicode Collation Algorithm) 조합 시퀀스. UCA의 세부 사항은 유니코드 협회 웹 사이트([www.unicode.org](http://www.unicode.org))에서 사용 가능한



유니코드 기술 보고서 #10에 있습니다. 이 옵션은 유니코드 데이터베이스 작성 시에만 사용될 수 있습니다.

#### **UCA400\_LSK**

유니코드 표준 버전 4.0.0에 근거하지만 적절한 순서로 Slovak 문자를 정렬하는 UCA(Unicode Collation Algorithm) 조합 시퀀스. UCA의 세부사항은 유니코드 협회 웹 사이트 ([www.unicode.org](http://www.unicode.org))에서 사용 가능한 유니코드 기술 보고서 #10에 있습니다. 이 옵션은 유니코드 데이터베이스 작성 시에만 사용될 수 있습니다.

#### **UCA400\_LTH**

유니코드 표준 버전 4.0.0에 근거하지만 Royal 태국어 사전 순서에 따라 모든 태국어 문자를 정렬하는 UCA(Unicode Collation Algorithm) 조합 시퀀스. UCA의 세부사항은 유니코드 협회 웹 사이트 ([www.unicode.org](http://www.unicode.org))에서 사용 가능한 유니코드 기술 보고서 #10에 있습니다. 이 옵션은 유니코드 데이터베이스 작성 시에만 사용될 수 있습니다. 이 조합기는 NLSCHAR 조합기 옵션에서 태국어 데이터를 다르게 순서화할 수 있습니다.

#### *language-aware-collation*

이 옵션은 유니코드 데이터베이스에만 사용될 수 있습니다. 데이터베이스 조합 시퀀스는 유니코드가 아닌 데이터베이스의 SYSTEM 조합에 근거합니다. 이 문자열은 *codepage\_territory* 형식이어야 합니다. 제공되는 문자열이 유효하지 않은 경우, 데이터베이스 작성에 실패합니다 (SQLCODE -204, 오브젝트가 없음). 자세한 정보와 시스템 기반 조합의 이름 지정에 대해서는 *유니코드 데이터의 언어 인식 조합*을 참조하십시오.

주: 버전 9.0 서버에 대해 CREATE DATABASE 명령 수행 시, 이 옵션을 사용할 수 없습니다. 디폴트로, 그러한 서버에서 유니코드 데이터베이스는 SYSTEM 조합으로 작성됩니다.

#### *locale-sensitive-collation*

이 옵션은 유니코드 데이터베이스에만 사용될 수 있습니다. 로케일 구분 UCA 기반 조합의 이름 지정 정보와 자세한 정보는 UCA(Unicode Collation Algorithm) 기반 조합을 참조하십시오. 제공된 조합 이름이 유효하지 않은 경우, CREATE DATABASE 명령 실행이 실패합니다 (SQLCODE -204).

#### **NLSCHAR**

특정 코드 세트/지역의 고유 조합 규칙을 사용하는 시스템 정의 조합 시퀀스.

이 옵션은 오직 태국어 코드 페이지(CP874)와 함께 사용할 수 있습니다. 이 옵션이 태국어가 아닌 환경에서 지정되는 경우, 명령이 실패하며 이유 코드가 4인 SQL1083N 오류를 리턴합니다.

## SYSTEM

데이터베이스 작성 시 이것이 디폴트 옵션입니다. 유니코드가 아닌 데이터베이스의 경우, 조합 시퀀스는 데이터베이스 지역에 근거합니다. 유니코드 데이터베이스의 경우, 이 옵션은 클라이언트 코드 세트 및 지역에 근거하여 언어 인식 조합으로 맵핑합니다. 적절한 언어 인식 조합이 사용 불가능하면, IDENTITY 조합이 사용됩니다.

## PAGESIZE *integer*

데이터베이스 작성 시 초기 테이블 스페이스(SYSCATSPACE, TEMPSPACE1, USERSPACE1)와 함께 디폴트 버퍼 풀의 페이지 크기를 지정합니다. 또한 이것은 모든 추후 CREATE BUFFERPOOL 및 CREATE TABLESPACE문의 디폴트 페이지 크기를 표시합니다. 접미부 K가 없는 정수의 올바른 값은 4 096, 8 192, 16 384 또는 32 768입니다. 접미부 K가 있는 integer에 유효한 값은 4, 8, 16 또는 32입니다. 정수 및 접미부 K 간에는 최소 하나의 스페이스가 필요합니다. 디폴트는 페이지 크기가 4 096바이트(4K)입니다.

## NUMSEGS *numsegs*

디폴트 SMS 테이블 스페이스의 데이터베이스 테이블 파일을 저장하기 위해 작성되고 사용되는 디렉토리(테이블 스페이스 컨테이너) 수를 지정합니다. 이 매개변수는 자동 스토리지 테이블 스페이스, DMS 테이블 스페이스, 명시적 작성 특성을 갖는 SMS 테이블 스페이스(데이터베이스가 작성될 때 작성됨) 또는 데이터베이스가 작성된 이후에 명시적으로 작성된 SMS 테이블 스페이스에 영향을 미치지 않습니다.

## DFT\_EXTENT\_SZ *dft\_extentsize*

테이블 스페이스의 디폴트 Extent 크기를 데이터베이스에 지정합니다.

## RESTRICTIVE

RESTRICTIVE 옵션이 제공되면, RESTRICT\_ACCESS 데이터베이스 구성 매개변수가 YES로 설정되며 어떤 특권이나 권한도 자동으로 PUBLIC에 권한 부여되지 않습니다. RESTRICTIVE 옵션이 제공되지 않으면

RESTRICT\_ACCESS 데이터베이스 구성 매개변수는 NO로 설정되며 다음의 모든 특권은 PUBLIC에 권한 부여됩니다.

- NULLID 스키마에서 작성된 모든 패키지에서의 BIND
- BINDADD
- CONNECT
- 스키마 SQLJ에서의 CREATEIN
- 스키마 NULLID에서의 CREATEIN

- CREATETAB
- 스키마 SQLJ에 있는 모든 프로시저에서 GRANT로 EXECUTE
- 스키마 SYSFUN에 있는 모든 기능과 프로시저에서 GRANT로 EXECUTE
- 스키마 SYSPROC에 있는 모든 기능과 프로시저에서 GRANT로 EXECUTE(감사 루틴 제외)
- 스키마 SYSIBM에 있는 모든 테이블 함수에서 EXECUTE
- 스키마 SYSIBM에 있는 다른 모든 프로시저에서 EXECUTE
- NULLID 스키마에서 작성된 모든 패키지에서 EXECUTE
- IMPLICIT\_SCHEMA
- SYSIBM 카탈로그 테이블에 대한 SELECT 액세스
- SYSCAT 카탈로그 뷰에 대한 SELECT 액세스
- SYSSTAT 카탈로그 뷰에 대한 SELECT 액세스
- SYSSTAT 카탈로그 뷰에 대한 UPDATE 액세스
- SYSDEFAULTUSERWORKLOAD 워크로드에서의 USAGE
- 테이블 스페이스 USERSPACE1에서의 USE

#### CATALOG TABLESPACE tblspace-defn

카탈로그 테이블, SYSCATSPACE를 보유할 테이블 스페이스의 정의를 지정합니다. 이 옵션이 지정되지 않고 데이터베이스에 대한 자동 스토리지가 사용 가능하지 않은 경우, SYSCATSPACE는 컨테이너로 *numsegs*개의 디렉토리가 있으며 Extent 크기가 *dft\_extentsize*인 시스템 관리 스페이스(SMS) 테이블 스페이스로서 작성됩니다. 예를 들어, *numsegs*가 5로 지정되면 다음 컨테이너가 작성됩니다.

```

/u/smith/smith/NODE0000/SQL00001/SQLT0000.0
/u/smith/smith/NODE0000/SQL00001/SQLT0000.1
/u/smith/smith/NODE0000/SQL00001/SQLT0000.2
/u/smith/smith/NODE0000/SQL00001/SQLT0000.3
/u/smith/smith/NODE0000/SQL00001/SQLT0000.4

```

이 옵션이 지정되지 않고 데이터베이스에 대한 자동 스토리지가 사용 가능한 경우, 정의된 스토리지 경로에 작성된 컨테이너가 있는 자동 스토리지 테이블 스페이스로서 SYSCATSPACE가 작성됩니다. 이 테이블 스페이스의 Extent 크기는 4입니다. AUTORESIZE, INITIALSIZE, INCREASESIZE 및 MAXSIZE의 적절한 값이 자동으로 설정됩니다.

테이블 스페이스 정의 필드에 대한 자세한 정보는 *CREATE TABLESPACE*문을 참조하십시오.

파티션된 데이터베이스 환경에서, 카탈로그 테이블 스페이스는 카탈로그 데이터베이스 파티션(CREATE DATABASE 명령이 발행된 데이터베이스 파티션)에서만 작성됩니다.

## USER TABLESPACE **tblspace-defn**

초기 사용자 테이블 스페이스(USERSPACE1)를 지정합니다. 이 옵션이 지정되지 않고 데이터베이스에 대한 자동 스토리지가 사용 가능하지 않은 경우, USERSPACE1은 컨테이너로 *numsegs*개의 디렉토리가 있는 SMS 테이블 스페이스로서 작성되며 Extent 크기가 *dft\_extentsize*입니다. 예를 들어, *numsegs*가 5로 지정되면 다음 컨테이너가 작성됩니다.

```
/u/smith/smith/NODE0000/SQL00001/SQLT0001.0  
/u/smith/smith/NODE0000/SQL00001/SQLT0002.1  
/u/smith/smith/NODE0000/SQL00001/SQLT0002.2  
/u/smith/smith/NODE0000/SQL00001/SQLT0002.3  
/u/smith/smith/NODE0000/SQL00001/SQLT0002.4
```

이 옵션이 지정되지 않고 데이터베이스에 대한 자동 스토리지가 사용 가능한 경우, 정의된 스토리지 경로에 작성된 컨테이너가 있는 자동 스토리지 테이블 스페이스로서 USERSPACE1이 작성됩니다. 이 테이블 스페이스의 Extent 크기는 *dft\_extentsize*입니다. AUTORESIZE, INITIALSIZE, INCREASESIZE 및 MAXSIZE의 적절한 값이 자동으로 설정됩니다.

테이블 스페이스 정의 필드에 대한 자세한 정보는 *CREATE TABLESPACE*문을 참조하십시오.

## TEMPORARY TABLESPACE **tblspace-defn**

초기 시스템 임시 테이블 스페이스(TEMPSPACE1)를 지정합니다. 이 옵션이 지정되지 않고 데이터베이스에 대한 자동 스토리지가 사용 가능하지 않은 경우, TEMPSPACE1은 컨테이너로 *numsegs*개의 디렉토리가 있는 SMS 테이블 스페이스로서 작성되며 Extent 크기가 *dft\_extentsize*입니다. 예를 들어, *numsegs*가 5로 지정되면 다음 컨테이너가 작성됩니다.

```
/u/smith/smith/NODE0000/SQL00001/SQLT0002.0  
/u/smith/smith/NODE0000/SQL00001/SQLT0001.1  
/u/smith/smith/NODE0000/SQL00001/SQLT0001.2  
/u/smith/smith/NODE0000/SQL00001/SQLT0001.3  
/u/smith/smith/NODE0000/SQL00001/SQLT0001.4
```

이 옵션이 지정되지 않고 데이터베이스에 대한 자동 스토리지가 사용 가능한 경우, 정의된 스토리지 경로에 작성된 컨테이너가 있는 자동 스토리지 테이블 스페이스로서 TEMPSPACE1이 작성됩니다. 이 테이블 스페이스의 Extent 크기는 *dft\_extentsize*입니다.

테이블 스페이스 정의 필드에 대한 자세한 정보는 *CREATE TABLESPACE*문을 참조하십시오.

## **tblspace-defn**

다음과 같은 명령 매개변수를 통해 다양한 테이블 스페이스 정의를 지정할 수 있습니다.

### **MANAGED BY**

## SYSTEM USING *container-string*

테이블 스페이스가 SMS 테이블 스페이스임을 지정합니다. 테이블 스페이스의 유형이 지정되지 않은 경우에는 일반 테이블 스페이스 작성이 디폴트 동작입니다.

SMS 테이블 스페이스의 경우, 해당 테이블 스페이스에 속하면서 해당 테이블 스페이스의 데이터가 저장될 컨테이너를 식별합니다. *container-string*은 240바이트를 초과할 수 없습니다.

각 *container-string*은 절대 또는 상대 디렉토리 이름이 될 수 있습니다.

절대 디렉토리가 아닌 경우 디렉토리 이름은 데이터베이스 디렉토리에 대해 상대적이며, 데이터베이스 디렉토리와 실제적으로 연결되지 않은 스토리지로 가는 경로 이름 별명(UNIX 시스템의 기호 링크)이 될 수 있습니다. 예를 들어

*dbdir/work/c1*은 별도의 파일 시스템으로의 기호 링크가 될 수 있습니다.

디렉토리 이름의 구성요소가 존재하지 않으면, 데이터베이스 관리 프로그램이 이를 작성합니다. 테이블 스페이스가 삭제되면, 데이터베이스 관리 프로그램이 작성한 모든 구성요소도 삭제됩니다. *container-string*에 의해 식별된 디렉토리가 존재하는 경우 그 디렉토리에 파일 또는 서브디렉토리가 들어 있으면 안 됩니다(SQLSTATE 428B2).

*container-string*의 형식은 운영 체제에 따라 다릅니다. Windows 운영 체제에서는 절대 디렉토리 경로 이름이 드라이브 이름 및 콜론(:)으로 시작하고, UNIX 시스템에서는 슬래시(/)로 시작합니다. 모든 플랫폼에서 상대 경로 이름은 운영 체제와 상관없는 문자로 시작합니다.

리모트 자원(예: LAN 경로 재지정된 드라이브 또는 NFS 마운트 파일 시스템)은 현재 Network Appliance Filers, IBM iSCSI, IBM Network Attached Storage, Network Appliance iSCSI, NEC iStorage S2100, S2200 또는 S4100 또는 Windows DB2 서버가 있는 NEC Storage NS 시리즈를 사용할 경우에만 지원됩니다. NEC Storage NS 시리즈는 UPS(Uninterrupted Power Supply)를 사용할 경우에만 지원됩니다. 대기(Standby)보다는 연속 UPS를 사용하도록 권장합니다. AIX의 NFS 마운트 파일 시스템은 `-o nointr` 옵션을 사용하여 무정전 모드에서 마운트되어 있어야 합니다.

## DATABASE USING

테이블 스페이스가 DMS 테이블 스페이스임을 지정합니다. 테

이블 스페이스의 유형이 지정되지 않은 경우에는 대형 테이블 스페이스 작성이 디폴트 동작입니다.

DMS 테이블 스페이스의 경우, 해당 테이블 스페이스에 속하면서 해당 테이블 스페이스의 데이터가 저장될 컨테이너를 식별합니다. 컨테이너의 유형(FILE 또는 DEVICE) 및 크기(PAGESIZE 페이지)가 지정됩니다. FILE과 DEVICE 컨테이너를 혼합하여 지정할 수 있습니다. *container-string*은 254바이트를 초과할 수 없습니다.

리모트 자원(예: LAN 경로 재지정된 드라이브 또는 NFS 마운트 파일 시스템)은 현재 Network Appliance Filers, IBM iSCSI, IBM Network Attached Storage, Network Appliance iSCSI, NEC iStorage S2100, S2200 또는 S4100 또는 Windows DB2 서버가 있는 NEC Storage NS 시리즈를 사용할 경우에만 지원됩니다. NEC Storage NS 시리즈는 UPS(Uninterrupted Power Supply)를 사용할 경우에만 지원됩니다. 대기(Standby)보다는 연속 UPS를 사용하도록 권장합니다.

모든 컨테이너는 모든 데이터베이스에서 고유해야 합니다. 하나의 컨테이너는 단 하나의 테이블 스페이스에만 속할 수 있습니다. 컨테이너의 크기는 다를 수 있으나 모든 컨테이너가 동일한 크기일 때 최적의 성능을 얻을 수 있습니다. *container-string*의 정확한 형식은 운영 체제에 따라 다릅니다.

#### **FILE** *container-string number-of-pages*

FILE 컨테이너의 경우, *container-string*은 절대 또는 상대 파일 이름이어야 합니다. 절대 경로가 아니면 파일 이름은 데이터베이스 디렉토리에 상대적입니다. 디렉토리 이름의 구성요소가 존재하지 않으면, 데이터베이스 관리 프로그램이 이를 작성합니다. 파일이 존재하지 않으면, 파일이 작성되어 데이터베이스 관리 프로그램이 지정하는 크기로 초기화됩니다. 테이블 스페이스가 삭제되면, 데이터베이스 관리 프로그램이 작성한 모든 구성요소도 삭제됩니다.

주: 파일이 존재하면 겹쳐쓰여지고, 지정된 크기보다 작으면 확장됩니다. 지정된 크기보다 크더라도, 파일은 잘리지 않습니다.

#### **DEVICE** *container-string number-of-pages*

DEVICE 컨테이너의 경우, *container-string*은 디바이스 이름이어야 합니다. 디바이스는 이미 존재하고 있어야 합니다.

#### **AUTOMATIC STORAGE**

테이블 스페이스가 자동 스토리지 테이블 스페이스임을 지정합니다. 데이터베이스에 대해 자동 스토리지가 정의되지 않은 경우 오류가 리턴됩니다(SQLSTATE 55060).

자동 스토리지 테이블 스페이스가 시스템 관리 스페이스(SMS) 테이블 스페이스 또는 데이터베이스 관리 스페이스(DMS) 테이블 스페이스로 작성됩니다. DMS가 선택되고 테이블 스페이스의 유형이 지정되지 않은 경우에는 대형 테이블 스페이스 작성이 디폴트 동작입니다. 자동 스토리지 테이블 스페이스에서 데이터베이스 관리 프로그램은 데이터베이스와 연관된 스토리지 경로에 기초하여 테이블 스페이스에 지정된 컨테이너를 결정합니다.

#### **EXTENTSIZE** *number-of-pages*

다음 컨테이너로 건너뛰기 전에 컨테이너에 기록될 PAGESIZE 페이지의 수를 지정합니다. Extent 크기 값은 정수로 지정될 수 있으며 숫자 뒤에 K(킬로바이트의 경우), M(메가바이트의 경우)이 붙습니다. 이런 방법으로 지정될 경우, 바이트 수를 페이지 크기로 나눈 최저값은 Extent 크기에 대한 값을 결정하는데 사용됩니다. 데이터베이스 관리 프로그램은 데이터가 저장됨에 따라 컨테이너를 반복적으로 순환합니다.

디폴트값은 유효한 범위가 있는 DFT\_EXTENT\_SZ 데이터베이스 구성 매개변수에 의해 제공되며, 유효한 범위는 2 - 256 페이지입니다.

#### **PREFETCHSIZE** *number-of-pages*

데이터 프리페치가 수행되는 동안 테이블 스페이스로부터 읽어 들일 PAGESIZE 페이지의 수를 지정합니다. 프리페치 크기 값은 정수로도 지정할 수 있으며 숫자 뒤에 K(킬로바이트의 경우), M(메가바이트의 경우) 또는 G(기가바이트의 경우)가 붙습니다. 이런 방식으로 지정될 경우, 바이트 수를 페이지 크기로 나눈 최저값은 프리페치 크기에 대한 페이지 수를 결정하는데 사용됩니다.

#### **OVERHEAD** *number-of-milliseconds*

입출력 제어기 오버헤드와 디스크 탐색 및 대기 시간을 지정합니다. 이 값은 쿼리 최적화 시 I/O의 비용을 계산할 때 사용

됩니다. *number-of-milliseconds* 값은 모든 숫자 리터럴(정수, 10진수 또는 부동 소수점)입니다. 이 값이 모든 컨테이너에 대해 동일하지 않은 경우 이 숫자는 테이블 스페이스에 속하는 모든 컨테이너에 대한 평균이어야 합니다.

버전 9 이상에서 작성된 데이터베이스의 경우 디폴트 입출력 제어기 오버헤드와 디스크 검색 및 대기 시간은 7.5 밀리초입니다. DB2 이전 버전에서 버전 9 이상으로 업그레이드된 데이터베이스의 경우 디폴트값은 12.67밀리초입니다.

#### **TRANSFERRATE** *number-of-milliseconds*

1페이지를 메모리로 읽는 시간을 지정합니다. 이 값은 쿼리 최적화 시 I/O의 비용을 계산할 때 사용됩니다. *number-of-milliseconds* 값은 모든 숫자 리터럴(정수, 10진수 또는 부동 소수점)입니다. 이 값이 모든 컨테이너에 대해 동일하지 않은 경우 이 숫자는 테이블 스페이스에 속하는 모든 컨테이너에 대한 평균이어야 합니다.

버전 9 이상에서 작성된 데이터베이스의 경우 1페이지를 메모리로 읽는 디폴트 시간은 0.06 밀리초입니다. DB2 이전 버전에서 버전 9 이상으로 업그레이드된 데이터베이스의 경우 디폴트값은 0.18밀리초입니다.

#### **NO FILE SYSTEM CACHING**

모든 I/O 조작이 파일 시스템 레벨 캐시를 생략하도록 지정합니다. 자세한 정보는 *파일 시스템 캐싱*이 없는 테이블 스페이스를 참조하십시오. 이것은 대부분 구성에서의 디폴트 옵션입니다. 자세한 정보는 *파일 시스템 캐싱* 구성을 참조하십시오.

#### **FILE SYSTEM CACHING**

목표 테이블 스페이스의 모든 I/O 조작이 파일 시스템 레벨에서 캐시되도록 지정합니다. 자세한 정보는 *파일 시스템 캐싱*이 없는 테이블 스페이스를 참조하십시오. 이것은 일부 구성에서의 디폴트 옵션입니다. 자세한 정보는 *파일 시스템 캐싱* 구성을 참조하십시오.

#### **AUTORESIZE**

DMS 테이블 스페이스 또는 자동 스토리지 테이블 스페이스의 자동 크기 조정 성능의 사용 여부를 지정합니다. 자동 크기 조정 가능 테이블 스페이스는 가득 차게 되면 자동으로 크기가 증가됩니다. 디폴트값은 DMS 테이블 스페이스의 경우 NO이고 자동 스토리지 테이블 스페이스의 경우 YES입니다.



**NO** DMS 테이블 스페이스 또는 자동 스토리지 테이블 스페이스의 자동 크기 조정 성능의 사용 안함 여부를 지정합니다.

**YES** DMS 테이블 스페이스 또는 자동 스토리지 테이블 스페이스의 자동 크기 조정 성능의 사용을 지정합니다.

#### **INITIALSIZE** *integer*

자동 스토리지 테이블 스페이스의 데이터베이스 파티션별 초기 크기를 지정합니다. 이 옵션은 자동 스토리지 테이블 스페이스에 대해서만 유효합니다. 정수 값은 K(킬로바이트), M(메가바이트) 또는 G(기가바이트) 앞에 나와야 합니다. 데이터베이스 관리 프로그램이 테이블 스페이스의 컨테이너에서 크기를 일관적으로 유지하려고 하기 때문에 사용되는 실제 값은 지정한 값보다 약간 작을 수 있습니다. 또한 테이블 스페이스가 자동 크기 조정이 가능하고 최초 크기가 새 테이블 스페이스에 추가되어야 하는 메타데이터를 포함할 정도로 충분히 크지 않은 경우에는 데이터베이스 관리 프로그램이 충분한 스페이스를 확보할 때까지 **INCREASESIZE**를 사용하여 테이블 스페이스 확장을 계속합니다. **INITIALSIZE** 절이 지정되지 않은 경우, 데이터베이스 관리 프로그램은 적절한 값을 판별합니다. *integer*의 값은 최소 48K이어야 합니다.

**K** K(KB).

**M** M(MB).

**G** G(GB).

#### **INCREASESIZE** *integer*

테이블 스페이스가 가득 차고 스페이스 요청이 있으면 자동 크기 조정이 사용 가능한 테이블 스페이스가 자동으로 증가되는 양을 데이터베이스 파티션별로 지정합니다. 정수 값 다음에 다음 중 하나가 와야 합니다.

- 스페이스 요청시 테이블 스페이스 크기 백분율로 양을 지정하는 PERCENT. PERCENT가 지정되면 정수 값은 0 - 100 사이에 있어야 합니다(SQLSTATE 42615).
- 바이트 단위로 양을 지정하는 K(킬로바이트), M (메가바이트) 또는 G(기가바이트)

데이터베이스 관리 프로그램이 테이블 스페이스의 컨테이너에서 증가량을 일관적으로 유지하려고 하기 때문에 사용되는 실제 값은 지정한 값보다 약간 작거나 클 수 있습니다. 테이블

스페이스가 자동 크기 조정 가능하지만 INCREASESIZE 절이 지정되지 않은 경우, 데이터베이스 관리 프로그램이 적절한 값을 결정합니다.

#### **PERCENT**

0 - 100의 퍼센트.

**K** K(KB).

**M** M(MB).

**G** G(GB).

#### **MAXSIZE**

자동 크기 조정이 사용 가능한 테이블 스페이스가 자동으로 증가될 수 있도록 최대 크기를 지정합니다. 테이블 스페이스가 자동 크기 조정 가능하지만 MAXSIZE 절이 지정되지 않은 경우, 디폴트값은 NONE입니다.

#### **NONE**

테이블 스페이스가 파일 시스템 용량 또는 최대 테이블 스페이스 크기까지 증가될 수 있도록 지정합니다.

#### *integer*

DMS 테이블 스페이스 또는 자동 스토리지 테이블 스페이스가 자동으로 증가될 수 있도록 데이터베이스 파티션별로 크기에 대한 하드 한계를 지정합니다. 정수 값은 K(킬로바이트), M(메가바이트) 또는 G(기가바이트) 앞에 나와야 합니다. 데이터베이스 관리 프로그램이 테이블 스페이스의 컨테이너에서 증가량을 일관적으로 유지하려고 하기 때문에 사용되는 실제 값은 지정한 값보다 약간 작을 수 있습니다.

**K** K(KB).

**M** M(MB).

**G** G(GB).

#### **WITH** *comment-string*

데이터베이스 디렉토리의 데이터베이스 항목을 설명합니다. 데이터베이스를 설명하도록 돕는 주석을 입력할 수 있습니다. 최대 길이는 30문자입니다. 캐리지 리턴이나 줄 바꾸기 문자는 허용되지 않습니다. 주석 텍스트는 작은따옴표나 큰따옴표로 묶어야 합니다.

#### **AUTOCONFIGURE**

사용자 입력을 기본으로, 버퍼 풀 크기, 데이터베이스 구성 및 데이터베이스 관리 프로그램 구성의 권장 설정값을 계산하고 선택적으로 적용합니다. CREATE

DATABASE 명령 발행 시 디폴트로 구성 어드바이저가 실행됩니다. AUTOCONFIGURE 옵션은 권장사항을 조정하려는 경우에만 필요합니다.

**USING** *input-keyword param-value*

표 7. 유효한 입력 키워드 및 매개변수 값

키워드	유효한 값	디폴트값	설명
mem_percent	1 - 100	25	전용으로 사용되는 인스턴스 메모리의 초기 백분율. 시스템에서 현재 메모리 사용에 따라 연속 데이터베이스에 대해 구성 어드바이저가 이 값을 조정합니다.
workload_type	simple, mixed, complex	mixed	단순 워크로드는 I/O 집중적이고 주로 트랜잭션인 경향이 있는 반면, 복합 워크로드는 CPU 집중적이며 주로 쿼리로 이루어지는 경향이 있습니다.
num_stmts	1 - 1 000 000	25	작업 단위(UOW) 당 명령문 수
tpm	1 - 200 000	60	분 당 트랜잭션 수
admin_priority	performance, recovery, both	both	성능 향상(분 당 더 많은 트랜잭션 수) 또는 보다 짧은 복구 시간을 위한 최적화
num_local_apps	0 - 5 000	0	연결된 로컬 응용프로그램 수
num_remote_apps	0 - 5 000	100	연결된 리모트 응용프로그램 수
isolation	RR, RS, CS, UR	RR	이 데이터베이스에 연결 중인 응용프로그램의 분리 레벨(반복 읽기, 읽기 안정성, 커서 안정성, 언커미트 읽기).
bp_resizeable	yes, no	yes	버퍼 풀의 크기를 조정할 수 있는지 여부

## APPLY

### DB ONLY

현재 데이터베이스 관리 프로그램 구성에 기초하여 데이터베이스 구성 및 버퍼 풀 설정에 대한 권장 값을 표시합니다. 데이터베이스 구성 및 버퍼 풀 설정에 대한 권장 변경사항을 적용합니다.

## DB AND DBM

데이터베이스 관리 프로그램 구성, 데이터베이스 구성 및 버퍼 풀 설정에 대한 권장 변경사항을 표시하고 적용합니다.

## NONE

구성 어드바이저를 사용하지 않습니다(디폴트로 사용 가능함).

- AUTOCONFIGURE 키워드가 CREATE DATABASE 명령으로 지정된 경우, DB2\_ENABLE\_AUTOCONFIG\_DEFAULT 변수값은 고려되지 않습니다. 적응 자체 성능 조정 메모리 및 자동 Runstats가 사용 가능해지며 구성 어드바이저는 APPLY DB 또는 APPLY DBM 옵션에서 표시한 대로 데이터베이스 구성 및 데이터베이스 관리 프로그램 구성 매개변수를 조정합니다.
- 데이터베이스에서 CREATE DATABASE 명령으로 AUTOCONFIGURE 옵션 지정 시 자체 성능 조정 메모리 관리자를 사용 가능하게 할 것을 권장합니다. 그러나 SHEAPTHRES가 0이 아닌 인스턴스에서 데이터베이스에 대해 AUTOCONFIGURE 명령을 실행할 경우, 정렬 메모리 조정(SORTHEAP)은 자동으로 사용할 수 있게 되지 않습니다. 정렬 메모리 조정(SORTHEAP)을 사용할 수 있게 하려면 UPDATE DATABASE MANAGER CONFIGURATION 명령을 사용하여 SHEAPTHRES를 0으로 설정해야 합니다. SHEAPTHRES의 값을 변경하면 이전의 기존 데이터베이스에서의 정렬 메모리 사용에 영향을 줄 수 있다는 점을 참고하십시오.

## 예:

CREATE DATABASE 명령의 몇 가지 예가 있습니다.

예 1:

```
CREATE DATABASE TESTDB3
AUTOMATIC STORAGE YES
```

데이터베이스 관리 프로그램 구성 매개변수 *dftdbpath*의 값인 드라이브에서 데이터베이스 TESTDB3이 작성됩니다. 값이 *dftdbpath*인 단일 스토리지 경로로 자동 스토리지가 사용 가능합니다.

예 2:

```
CREATE DATABASE TESTDB7 ON C:.,D:
```

데이터베이스 TESTDB7이 드라이브 C:(스토리지 경로 목록에서의 최초 드라이브)에 작성됩니다. 자동 스토리지가 내재적으로 사용 가능하며 스토리지 경로는 C: 및 D:입니다.

예 3:

```
CREATE DATABASE TESTDB15
AUTOMATIC STORAGE YES
ON C:.,D: DBPATH ON E:
```

데이터베이스 TESTDB15가 드라이브 E:(DBPATH로서 명시적으로 나열됨)에 작성됩니다. 자동 스토리지가 명시적으로 사용 가능하며 스토리지 경로는 C: 및 D:입니다.

## 사용 시 참고사항

CREATE DATABASE 명령:

- 지정된 서브디렉토리에서 데이터베이스를 작성합니다. 파티션된 데이터베이스 환경에서, db2nodes.cfg에 나열된 모든 데이터베이스 파티션에서 데이터베이스를 작성하며, 각 데이터베이스 파티션에서 지정된 서브디렉토리 아래에 \$DB2INSTANCE/NODExxxx 디렉토리를 작성합니다. 단일 파티션 데이터베이스 환경에서, 지정된 서브디렉토리 아래에 \$DB2INSTANCE/NODE0000 디렉토리를 작성합니다.
- 시스템 카탈로그 테이블 및 복구 로그를 작성합니다.
- 다음 데이터베이스 디렉토리에서 데이터베이스를 카탈로그합니다.
  - path에서 표시하는 경로에 있는 서버의 로컬 데이터베이스 디렉토리, 또는 dfidbpath 매개변수에 의해 데이터베이스 관리 프로그램 시스템 구성 파일에 정의된 디폴트 데이터베이스 경로. 로컬 데이터베이스 디렉토리는 데이터베이스가 있는 각 파일 시스템에 상주합니다.
  - 접속된 인스턴스의 서버 시스템 데이터베이스 디렉토리. 결과 디렉토리 항목은 데이터베이스 이름과 데이터베이스 별명을 포함합니다.

명령이 리모트 클라이언트에서 발행되면, 클라이언트의 시스템 데이터베이스 디렉토리도 데이터베이스 이름 및 별명으로 갱신됩니다.

어느 것도 존재하지 않는 경우, 시스템 또는 로컬 데이터베이스 디렉토리를 작성합니다. 지정된 경우에는 주석 및 코드 세트 값이 양쪽 디렉토리에 위치합니다.

주: 경로 구성 매개변수 **newlogpath**에 의해 데이터베이스 변경이 설정되지 않은 경우, 로그 파일 구성 매개변수 **logpath**의 위치 디폴트는 DBPATH ON 옵션으로 표시된 경로입니다. 데이터베이스 데이터에서 분리된 데이터베이스 정보를 보존하기 위해 자동 스토리지가 사용 가능할 때 DBPATH ON 옵션을 사용하는 것이 좋습니다.

- 지정된 코드 세트, 지역 및 조합 시퀀스를 저장합니다. 조합 시퀀스가 고유 가중치로 구성되거나 ID 시퀀스인 경우, 구성 파일에 플래그가 설정됩니다.
- 소유자가 SYSIBM인 SYSCAT, SYSFUN, SYSIBM 및 SYSSTAT라는 스키마를 작성합니다. 이 명령이 발행된 데이터베이스 파티션 서버는 새 데이터베이스의 카탈로그 데이터베이스 파티션이 됩니다. 다음과 같은 두 개의 데이터베이스 파티션 그룹이 자동으로 작성됩니다. IBMDEFAULTGROUP 및 IBMCATGROUP.

- 이전에 정의된 데이터베이스 관리 프로그램 바인드 파일을 데이터베이스에 바인드합니다(유틸리티 바인드 파일 목록 (db2ubind.lst)에 나열됨). 하나 이상의 파일이 정상적으로 바인드하지 않는 경우, CREATE DATABASE는 SQLCA에서 경고를 리턴하며, 실패한 바인드에 대한 정보를 제공합니다. 바인드에 실패하면, 사용자는 정정 조치를 취하고 수동으로 실패 파일을 바인드할 수 있습니다. 어떤 경우든지 데이터베이스가 작성됩니다. RESTRICTIVE 옵션이 선택되지 않으면, PUBLIC에 CREATEIN 특권 부여로 바인드 수행 시 NULLID라는 스키마가 내재적으로 작성됩니다.

유틸리티 바인드 파일 목록은 서버의 이전 버전에 대해 바인드될 수 없는 두 개의 바인드 파일을 포함합니다.

- db2ugtpi.bnd는 DB2 버전 2 서버에 대해 바인드될 수 없습니다.

- db2dropv.bnd는 DB2 병렬 개정판 버전 1 서버에 대해 바인드될 수 없습니다.

최신 레벨에 없는 서버에 대해 db2ubind.lst가 바인드되면, 두 개의 파일에 속하는 경고가 리턴되며, 무시될 수 있습니다.

- SYSCATSPACE, TEMPSPACE1 및 USERSPACE1 테이블 스페이스를 작성합니다. SYSCATSPACE 테이블 스페이스는 카탈로그 데이터베이스 파티션에서만 작성됩니다.
- 다음을 권한 부여합니다.
  - SYSFUN 스키마의 모든 기능에서 PUBLIC에 EXECUTE WITH GRANT 특권 부여
  - SYSIBM 스키마의 모든 프로시저에서 PUBLIC에 EXECUTE 특권 부여
  - 데이터베이스 작성자에 DBADM, CONNECT, CREATETAB, BINDADD, CREATE\_NOT\_FENCED, IMPLICIT\_SCHEMA 및 LOAD 권한 부여
  - PUBLIC에 CONNECT, CREATETAB, BINDADD 및 IMPLICIT\_SCHEMA 권한 부여
  - PUBLIC에 USERSPACE1 TEMPORARY 테이블 스페이스에 대한 USE 특권 부여
  - PUBLIC에 각 시스템 카탈로그에서 SELECT 특권 부여
  - 정상적으로 바인드된 유틸리티에 대해 PUBLIC에 BIND 및 EXECUTE 특권 부여.
  - SYSFUN 스키마의 모든 기능에서 PUBLIC에 EXECUTE WITH GRANT 특권 부여.
  - SYSIBM 스키마의 모든 프로시저에서 PUBLIC에 EXECUTE 특권 부여.

주: RESTRICTIVE 옵션이 제공되면, RESTRICT\_ACCESS 데이터베이스 구성 매개변수가 YES로 설정되며 어떤 특권이나 권한도 자동으로 PUBLIC에 권한 부여되지 않습니다. 추가 정보는 CREATE DATABASE 명령의 RESTRICTIVE 옵션을 참조하십시오.

자동 스토리지는 명시적으로 컨테이너 정의를 지정하지 않고 테이블 스페이스가 작성될 수 있는 데이터베이스와 연관된 스토리지 경로의 콜렉션입니다(자세한 정보는 CREATE TABLESPACE문 참조). 자동 스토리지는 디폴트로 사용 가능하지만, 작성 시 데이터베이스에 대해 명시적으로 사용 안할 수 있습니다. AUTOMATIC STORAGE NO 옵션을 지정하여 데이터베이스 작성 시 자동 스토리지를 사용 안할 수 있습니다.

자동 스토리지는 데이터베이스 작성 시에만 사용 가능하며 데이터베이스가 작성된 후에는 사용 가능하지 않음을 참고하는 것이 중요합니다. 또한, 자동 스토리지는 이를 사용하도록 데이터베이스가 정의되고 나면 자동 스토리지를 사용 안할 수 없습니다.

주어진 데이터베이스 파티션의 자동 스토리지 경로에 대한 여유 공간을 계산할 때, 데이터베이스 관리 프로그램은 다음 디렉토리 또는 마운트 지점이 존재하는지 점검한 후 발견된 첫 번째 디렉토리 또는 마운트 지점을 사용합니다. 이를 수행할 때, 파일 시스템은 스토리지 경로 아래 임의의 위치에서 마운트할 수 있으며, 데이터베이스 관리 프로그램은 테이블 스페이스 컨테이너에 사용 가능한 실제 양이 스토리지 경로 디렉토리 자체와 연관된 양과 동일하지 않을 수도 있음을 인식합니다.

1. <storage path>/<instance name>/NODE####/<database name>
2. <storage path>/<instance name>/NODE####
3. <storage path>/<instance name>
4. <storage path>/<

여기서

- <storage path>는 데이터베이스와 연관된 스토리지 경로입니다.
- <instance name>은 데이터베이스가 있는 인스턴스입니다.
- NODE####는 데이터베이스 파티션 번호에 해당합니다(예를 들어, NODE0000 또는 NODE0001).
- <database name>은 데이터베이스의 이름입니다.

두 개의 논리적 데이터베이스 파티션이 하나의 물리적 시스템에 존재하며 단일 스토리지 경로(/db2data)로 데이터베이스가 작성되는 예를 고려하십시오. 각각의 데이터베이스 파티션은 해당 스토리지 경로를 사용하지만, 자체 파일 시스템 내의 각 파티션에서 데이터를 분리하려고 할 수도 있습니다. 이 경우, 각각의 파티션에 대해 별도의 파일 시스템을 작성하여 /db2data/<instance>/NODE####에 마운트할 수 있습니다. 스토리지 경로에 컨테이너를 작성하고 여유 공간을 판별할 때, 데이터베이스 관리 프로그램은

/db2data에 대한 여유 공간 정보를 검색하지 않지만 대신 해당하는 /db2data/<instance>/NODE#### 디렉토리에 대한 여유 공간을 검색합니다.

일반적으로 멀티파티션 데이터베이스의 각 파티션에 대해 동일한 스토리지 경로를 사용해야 하며 CREATE DATABASE 명령 실행 이전에 존재해야 합니다. 한 가지 예외는 데이터베이스 파티션 표현식을 스토리지 경로 내에서 사용하는 경우입니다. 이를 수행하면 데이터베이스 파티션 번호를 스토리지 경로에 반영하여 결과로 생성되는 경로 이름이 파티션마다 다릅니다.

인수 " \$N" ([blank]\$N)을 사용하여 데이터베이스 파티션 표현식을 나타낼 수 있습니다. 데이터베이스 파티션 표현식은 스토리지 경로의 어디에서든 사용할 수 있으며 데이터베이스 파티션 표현식을 여러 개 지정할 수도 있습니다. 데이터베이스 파티션 표현식은 스페이스 문자로 끝내십시오. 스페이스 다음에 오는 문자는 데이터베이스 파티션 표현식을 평가한 후 스토리지 경로에 추가됩니다. 스토리지 경로에서 데이터베이스 파티션 표현식 뒤에 스페이스 문자가 없으면, 나머지 문자열은 표현식의 일부로 간주됩니다. 인수는 다음 형식 중 한가지 형식으로만 사용할 수 있습니다.

연산자는 왼쪽에서 오른쪽으로 계산됩니다. %는 모듈러스 연산자를 표시합니다. 예에서 데이터베이스 파티션 번호는 10으로 가정합니다.

구문	예 :	값
[blank]\$N	" \$N"	10
[blank]\$N+[number]	" \$N+100"	110
[blank]\$N#[number]	" \$N%5"	0
[blank]\$N+[number]#[number]	" \$N+1%5"	1
[blank]\$N#[number]+[number]	" \$N%4+2"	4
<sup>a</sup> %는 모듈러스입니다.		

dbadm 권한을 사용하여, 다른 사용자나 PUBLIC에 해당 특권을 부여(그리고 권한 취소)할 수 있습니다. 데이터베이스에서 sysadm 또는 dbadm 권한을 가진 다른 관리자가 해당 특권을 취소하더라도, 데이터베이스 작성자는 이 특권을 보유하고 있습니다.

MPP 환경에서, 데이터베이스 관리 프로그램은 모든 데이터베이스 파티션에서 지정된 경로나 디폴트 경로 아래에 서브디렉토리 \$DB2INSTANCE/NODExxxx를 작성합니다. xxxx는 db2nodes.cfg 파일에 정의된 데이터베이스 파티션 번호입니다(즉, 데이터베이스 파티션 0은 NODE0000). 서브디렉토리 SQL00001 - SQLnnnnn이 이 경로에 상주합니다. 이로 인해 다른 데이터베이스 파티션과 연관된 데이터베이스 오브젝트가 다른 디렉토리에 저장됩니다(지정된 경로나 디폴트 경로 아래의 서브디렉토리 \$DB2INSTANCE가 모든 디렉토리 파티션에 의해 공유되는 경우에도).

LDAP(Lightweight Directory Access Protocol) 지원이 현재 머신에서 사용 가능하면, 데이터베이스는 자동으로 LDAP 디렉토리에 등록됩니다. 동일한 이름의 데이터베이스 오브젝트가 이미 LDAP 디렉토리에 존재하는 경우, 데이터베이스는 여전히 로컬



시스템에 작성되지만, 경고 메시지가 리턴되며 이름 지정 충돌이 있음을 표시합니다. 이 경우, 사용자는 CATALOG LDAP DATABASE 명령을 사용하여 수동으로 LDAP 데이터베이스를 카탈로그할 수 있습니다.

응용프로그램이 이미 데이터베이스에 연결되어 있으면 CREATE DATABASE가 실패합니다.

데이터베이스가 작성되면, 자세한 교착 상태 이벤트 모니터가 작성됩니다. 모니터 사용 시, 이 이벤트 모니터와 연관된 일부 오버헤드가 있습니다. DROP EVENT MONITOR 명령을 발행하여 교착 상태 이벤트 모니터를 삭제(drop)할 수 있습니다.

새 데이터베이스의 다른 별명 이름을 정의하려면 CATALOG DATABASE를 사용하십시오.

코드 세트 및 지역 값의 조합이 유효해야 합니다. 지원되는 조합 목록은 지원되는 지역 코드 및 코드 페이지를 참조하십시오.

Windows 시스템에서 데이터베이스 경로(드라이브 대신)를 지정하려면, DB2 레지스트리 변수를 다음과 같이 설정해야 합니다. DB2\_CREATE\_DB\_ON\_PATH=YES.

## 호환성

버전 8 이전 버전과의 호환성:

- DBPARTITIONNUM 대신 NODE 키워드를 사용할 수 있습니다.



---

## 제 29 장 CREATE TOOLS CATALOG

기존 또는 새 데이터베이스에서 DB2 도구 카탈로그 테이블을 작성합니다. 데이터베이스는 로컬 데이터베이스여야 합니다.

도구 카탈로그에는 태스크 센터 및 제어 센터와 같은 도구를 사용하여 구성하는 관리 태스크에 대한 정보가 들어 있습니다.

이 명령은 모든 응용프로그램을 선택적으로 강제 실행하며 도구 카탈로그에 대한 새 테이블 스페이스가 작성되는 경우 데이터베이스 관리 프로그램을 중지하고 재시작합니다. 이 명령은 또한 DAS(DB2 Administration Server) 구성을 갱신하고 스케줄러를 활성화 합니다.

이 명령은 IBM Data Server Client에서는 유효하지 않습니다.

### 범위

이 명령이 실행되는 노드는 새 데이터베이스의 카탈로그 노드가 됩니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*

사용자에게는 또한 DB2 Administration Server 구성 매개변수를 갱신하기 위해 DASADM이 필요합니다.

### 필수 연결

이 명령 처리 중 임시로 데이터베이스 연결이 설정됩니다. 새 테이블 스페이스가 작성되는 경우 이 명령은 데이터베이스 관리 프로그램을 선택적으로 중지하고 재시작합니다.

### 명령 구문

```
▶▶—CREATE TOOLS CATALOG—catalog-name—————▶
▶—CREATE NEW DATABASE—database-name—————▶
└—USE EXISTING—┬—TABLESPACE—tablespace-name—IN—┐ DATABASE—database-name—┘
▶┬—FORCE—┘ ┬—KEEP INACTIVE—┘—————▶▶
```

## 명령 매개변수

### **CATALOG** *catalog-name*

DB2 도구 카탈로그를 고유하게 식별하는 데 사용되는 이름. 카탈로그 테이블은 이 스키마 이름으로 작성됩니다.

### **NEW DATABASE** *database-name*

새 데이터베이스에 지정될 이름. 이 이름은 데이터베이스와 로컬 데이터베이스 디렉토리 또는 시스템 데이터베이스 디렉토리의 다른 데이터베이스를 구분하는 고유 이름이어야 합니다. 이름은 데이터베이스의 이름 지정 규칙을 준수해야 합니다.

### **EXISTING DATABASE** *database-name*

도구 카탈로그를 호스팅할 기존 데이터베이스의 이름. 해당 데이터베이스는 로컬 데이터베이스여야 합니다.

### **EXISTING TABLESPACE** *tablespace-name*

DB2 도구 카탈로그 테이블을 작성하는 데 사용되는 기존 32K 페이지 테이블 스페이스를 지정하는 데 사용될 이름. 테이블을 작성하려면 32K 페이지 크기 임시 테이블 스페이스 또한 존재해야 합니다.

### **FORCE**

새 테이블 스페이스에서 도구 카탈로그를 작성하는 경우 데이터베이스 관리 프로그램을 재시작해야 하며 이 경우 연결된 응용프로그램이 없어야 합니다. FORCE 옵션을 사용하면 데이터베이스에 연결된 응용프로그램이 없는지 확인할 수 있습니다. 응용프로그램이 연결된 경우 기존 테이블 스페이스를 지정하지 않으면 도구 카탈로그 작성에 실패합니다.

### **KEEP INACTIVE**

이 옵션은 DB2 Administration Server 매개변수를 갱신하거나 스케줄러를 사용하지 않습니다.

## 예:

```
db2 create tools catalog cc create new database toolsdb
db2 create tools catalog catalog1 use existing database toolsdb force
db2 create tools catalog foobar use existing tablespace user32Ksp
  in database toolsdb
db2 create tools catalog toolscat use existing database toolsdb keep inactive
```

## 사용 시 참고사항

- 도구 카탈로그 테이블에는 두 개의 32K 페이지 테이블 스페이스(일반 및 임시)가 필요합니다. 또한 기존 테이블 스페이스를 지정하지 않는 경우 테이블 스페이스에 대해 새 32K 버퍼 풀이 작성됩니다. 이 경우 데이터베이스 관리 프로그램을 재시작해야 합니다. 데이터베이스 관리 프로그램이 재시작되어야 하는 경우 모든 기존 응용

프로그램을 비활성화해야 합니다. 새 테이블 스페이스는 디폴트 데이터베이스 디렉토리 경로에 각각 단일 컨테이너가 있는 상태로 작성됩니다.

- 이 명령을 실행하기 전에 이 이름을 사용하는 활성 카탈로그가 존재하는 경우 해당 카탈로그는 사용하지 않고 새 카탈로그가 활성 카탈로그가 됩니다.
- 여러 DB2 도구 카탈로그를 동일한 데이터베이스에서 작성할 수 있으며 카탈로그는 카탈로그 이름으로 고유하게 식별됩니다.
- DAS(DB2 Administration Server) 구성에서 **jdk\_path** 구성 매개변수를 Java™에 대해 지원되는 최소 SDK 레벨로 설정해야 합니다.
- DAS 구성 매개변수를 갱신하려면 DB2 Administration Server에 대한 *dasadm* 권한이 필요합니다.
- KEEP INACTIVE 옵션을 지정하지 않는 경우 이 명령은 DB2 도구 카탈로그 데이터베이스 구성과 관련된 로컬 DAS 구성 매개변수를 갱신하고 로컬 DAS 서버에서 스케줄러를 사용합니다.
- 32비트 및 64비트 인스턴스를 모두 지원하는 플랫폼(AIX, HP-UX 및 Solaris) 중 하나의 64비트 인스턴스에 대해 도구 카탈로그를 작성하는 경우 **jdk\_64\_path** 구성 매개변수를 설정해야 합니다.
- 데이터베이스 파티셔닝 기능(DPF)을 사용하는 복수 파티션 환경의 경우 카탈로그 파티션에 32KB REGULAR 테이블 스페이스가 존재해야 합니다. 그렇지 않으면 테이블 스페이스가 지정될 때 아래와 같은 명령이 실패합니다.

```
db2 create tools catalog foobar use existing tablespace user32Ksp
in database toolsdb
```



---

## 제 30 장 DEACTIVATE DATABASE

지정된 데이터베이스를 중지합니다.

### 범위

MPP 시스템에서 이 명령은 시스템의 모든 데이터베이스 파티션에 있는 지정된 데이터베이스를 비활성화합니다. 이들 데이터베이스 파티션 중 하나 이상에서 오류가 발생하면 경고가 리턴됩니다. 데이터베이스는 일부 데이터베이스 파티션에서 성공적으로 비활성화되지만 오류가 발생하는 노드에서는 계속 활성 상태일 수 있습니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*

### 필수 연결

없음

### 명령 구문

```
►► DEACTIVATE DATABASE database-alias
    └── DB ───┘
└── USER username ───┘
    └── USING password ───┘
```

### 명령 매개변수

**DATABASE** *database-alias*

중지할 데이터베이스의 별명을 지정합니다.

**USER** *username*

데이터베이스를 중지하는 사용자를 지정합니다.

**USING** *password*

사용자 ID의 암호를 지정합니다.

## 사용 시 참고사항

ACTIVATE DATABASE로 초기화되는 데이터베이스는 DEACTIVATE DATABASE 또는 db2stop으로 종료할 수 있습니다. 데이터베이스가 ACTIVATE DATABASE로 초기화된 경우, 해당 데이터베이스에서 연결을 끊는 마지막 응용프로그램이 데이터베이스를 종료하지 않으며 DEACTIVATE DATABASE를 사용해야 합니다. (이 경우에 db2stop도 데이터베이스를 종료합니다.)

DEACTIVATE DATABASE 명령을 발행하는 응용프로그램은 어떤 데이터베이스에도 활성 데이터베이스 연결을 가질 수 없습니다.



---

## 제 31 장 DECOMPOSE XML DOCUMENT

이 명령은 스토어드 프로시저를 호출하여 등록되고 분석 가능한 XML 스키마를 사용하여 단일 XML 문서를 분석합니다.

### 권한 부여

다음의 특권 또는 권한 그룹 중 하나가 필요합니다.

- 다음 권한 부여 중 하나여야 합니다.
  - 어노테이션이 있는 스키마 문서 세트에 참조된 모든 목표 테이블에 대한 CONTROL 특권
  - DATAACCESS 권한
- 다음의 모든 특권:
  - 조치 파일에 지정된 조작에 필요한, 목표 테이블에 대한 INSERT 특권
  - db2-xdb:expression 또는 db2-xdb:condition 어노테이션에서 참조된 테이블에 대한 적용 가능한 SELECT, INSERT, UPDATE 또는 DELETE 특권

VALIDATE 옵션을 지정한 경우, XML 스키마에 대한 USAGE 특권도 필요합니다.

### 필수 연결

데이터베이스

### 명령 구문

```
▶▶—DECOMPOSE XML DOCUMENT—xml-document-name—XMLSCHEMA—xml-schema-name————▶▶  
▶▶—VALIDATE—▶▶
```

### 명령 매개변수

**DECOMPOSE XML DOCUMENT** *xml-document-name*

*xml-document-name*은 분석할 입력 XML 문서의 파일 경로 및 파일 이름입니다.

**XMLSCHEMA** *xml-schema-name*

*xml-schema-name*은 문서 분석에 사용되도록 XML 스키마 저장소에 등록된 기존 XML 스키마의 이름입니다. *xml-schema-name*은 뒤에 마침표와 XML 스키마 이름이 오는 선택적 SQL 스키마 이름으로 구성되는 규정된 SQL ID입니다. SQL 스키마 이름을 지정하지 않은 경우, DB2 특수 레지스터 CURRENT SCHEMA의 값으로 간주됩니다.

## VALIDATE

이 매개변수는 입력 XML 문서가 먼저 유효성 확인을 받고 문서가 유효한 경우에만 분석됨을 표시합니다. VALIDATE를 지정하지 않은 경우, 입력 XML 문서는 분석 이전에 유효성 확인을 받지 않습니다.

## 예:

다음 예는 XML 문서 ./gb/document1.xml의 유효성을 확인하고 등록된 XML 스키마 DB2INST1.GENBANKSCHEMA로 분석됨을 지정합니다.

```
DECOMPOSE XML DOCUMENT ./gb/document1.xml
                XMLSCHEMA DB2INST1.GENBANKSCHEMA
                VALIDATE
```

다음 예는 XML 문서 ./gb/document2.xml이 등록된 XML 스키마 DB2INST2."GENBANK SCHEMA1"에 대해 유효성 확인 없이 분석됨을 지정합니다. 이 때 DB2 특수 레지스터 CURRENT SCHEMA의 값이 DB2INST2로 설정된 것으로 가정합니다.

```
DECOMPOSE XML DOCUMENT ./gb/document2.xml
                XMLSCHEMA "GENBANK SCHEMA1"
```

---

## 제 32 장 DECOMPOSE XML DOCUMENT

이 명령은 스토어드 프로시저를 호출하여 등록되고 분석 가능한 XML 스키마를 사용하여 단일 XML 문서를 분석합니다.

### 권한 부여

다음의 특권 또는 권한 그룹 중 하나가 필요합니다.

- 다음 권한 부여 중 하나여야 합니다.
  - 어노테이션이 있는 스키마 문서 세트에 참조된 모든 목표 테이블에 대한 CONTROL 특권
  - DATAACCESS 권한
- 다음의 모든 특권:
  - 조치 파일에 지정된 조작에 필요한, 목표 테이블에 대한 INSERT 특권
  - db2-xdb:expression 또는 db2-xdb:condition 어노테이션에서 참조된 테이블에 대한 적용 가능한 SELECT, INSERT, UPDATE 또는 DELETE 특권

VALIDATE 옵션을 지정한 경우, XML 스키마에 대한 USAGE 특권도 필요합니다.

### 필수 연결

데이터베이스

### 명령 구문

```
▶▶—DECOMPOSE XML DOCUMENT—xml-document-name—XMLSCHEMA—xml-schema-name————▶▶  
▶▶—VALIDATE—▶▶
```

### 명령 매개변수

**DECOMPOSE XML DOCUMENT** *xml-document-name*

*xml-document-name*은 분석할 입력 XML 문서의 파일 경로 및 파일 이름입니다.

**XMLSCHEMA** *xml-schema-name*

*xml-schema-name*은 문서 분석에 사용되도록 XML 스키마 저장소에 등록된 기존 XML 스키마의 이름입니다. *xml-schema-name*은 뒤에 마침표와 XML 스키마 이름이 오는 선택적 SQL 스키마 이름으로 구성되는 규정된 SQL ID입니다. SQL 스키마 이름을 지정하지 않은 경우, DB2 특수 레지스터 CURRENT SCHEMA의 값으로 간주됩니다.

## VALIDATE

이 매개변수는 입력 XML 문서가 먼저 유효성 확인을 받고 문서가 유효한 경우에만 분석됨을 표시합니다. VALIDATE를 지정하지 않은 경우, 입력 XML 문서는 분석 이전에 유효성 확인을 받지 않습니다.

## 예:

다음 예는 XML 문서 ./gb/document1.xml의 유효성을 확인하고 등록된 XML 스키마 DB2INST1.GENBANKSCHEMA로 분석됨을 지정합니다.

```
DECOMPOSE XML DOCUMENT ./gb/document1.xml
                XMLSCHEMA DB2INST1.GENBANKSCHEMA
                VALIDATE
```

다음 예는 XML 문서 ./gb/document2.xml이 등록된 XML 스키마 DB2INST2."GENBANK SCHEMA1"에 대해 유효성 확인 없이 분석됨을 지정합니다. 이 때 DB2 특수 레지스터 CURRENT SCHEMA의 값이 DB2INST2로 설정된 것으로 가정합니다.

```
DECOMPOSE XML DOCUMENT ./gb/document2.xml
                XMLSCHEMA "GENBANK SCHEMA1"
```

---

## 제 33 장 DEREGISTER

네트워크 디렉토리 서버에서 DB2 서버를 등록 해제합니다.

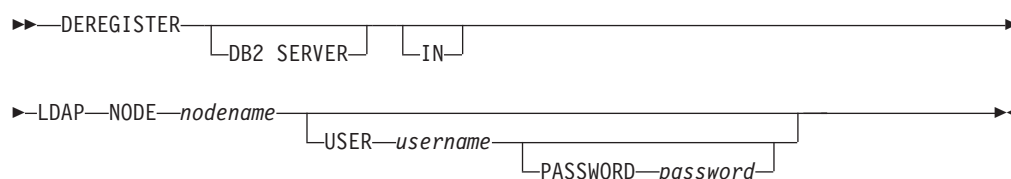
### 권한 부여

없음

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

**IN** DB2 서버를 등록 해제할 네트워크 디렉토리 서버를 지정합니다. 올바른 값은 LDAP(Lightweight Directory Access Protocol) 디렉토리 서버의 경우 LDAP입니다.

#### **USER** *username*

이것은 사용자의 LDAP 식별 이름(DN)입니다. LDAP 사용자 DN은 LDAP 디렉토리에서 오브젝트를 삭제하기 위한 충분한 권한이 있어야 합니다. LDAP에서 등록 해제할 때 사용자 이름은 선택적입니다. 사용자의 LDAP DN이 지정되지 않은 경우 현재 로그인 사용자의 증명서가 사용됩니다.

#### **PASSWORD** *password*

어카운트 암호

#### **NODE** *nodename*

노드 이름은 DB2 서버가 LDAP에 등록되었을 때 지정된 값입니다.

### 사용 시 참고사항

이 명령은 LDAP 환경에 있을 때 리모트 머신에 대해서만 발행되어야 합니다. 리모트 머신에 대해 발행될 때 리모트 서버의 노드 이름을 지정해야 합니다.

인스턴스가 삭제될 때 DB2 서버가 자동으로 등록 해제됩니다.



---

## 제 34 장 DESCRIBE

DESCRIBE 명령은 다음 항목에 대한 정보를 표시합니다.

- SELECT, CALL 또는 XQuery문의 출력
- 테이블 또는 뷰의 컬럼
- 테이블 또는 뷰의 인덱스
- 테이블 또는 뷰의 데이터 파티션

### 권한 부여

필요한 권한 부여는 DESCRIBE 명령을 사용하여 표시하려는 정보 유형에 따릅니다.

- SYSTOOLSTMPSPACE 테이블 스페이스가 있는 경우, 다음 표에 표시된 권한 중 하나가 필요합니다.

정보 표시를 위한 오브젝트	필요한 특권 또는 권한
SELECT문 또는 XQuery문의 출력	SELECT문에서 참조되는 각 테이블이나 뷰에 대해 다음 특권이나 권한 중 하나: <ul style="list-style-type: none"><li>• SELECT 특권</li><li>• DATAACCESS 권한</li><li>• DBADM 권한</li><li>• SQLADM 권한</li><li>• EXPLAIN 권한</li></ul>
CALL문의 출력	다음 특권이나 권한 중 하나: <ul style="list-style-type: none"><li>• DATAACCESS 권한</li><li>• 스토어드 프로시저에 대한 EXECUTE 특권</li></ul>

정보 표시를 위한 오브젝트	필요한 특권 또는 권한
<p>테이블 또는 뷰의 컬럼</p>	<p>SYSCAT.COLUMNS 시스템 카탈로그 테이블에 대해 다음 특권이나 권한 중 하나:</p> <ul style="list-style-type: none"> <li>• SELECT 특권</li> <li>• ACCESSCTRL 권한</li> <li>• DATAACCESS 권한</li> <li>• DBADM 권한</li> <li>• SECADM 권한</li> <li>• SQLADM 권한</li> </ul> <p>SHOW DETAIL 매개변수를 사용하려는 경우, SYSCAT.DATAPARTITIONEXPRESSION 시스템 카탈로그 테이블에 대해서도 해당 특권이나 권한이 필요합니다.</p> <p>PUBLIC은 선언된 임시 테이블에 대한 모든 특권을 소유하므로 사용자는 이 명령을 사용하여 해당 연결 내에 존재하는 선언된 임시 테이블에 대한 정보를 표시할 수 있습니다.</p>
<p>테이블 또는 뷰의 인덱스</p>	<p>SYSCAT.INDEXES 시스템 카탈로그 테이블에 대해 다음 특권이나 권한 중 하나:</p> <ul style="list-style-type: none"> <li>• SELECT 특권</li> <li>• ACCESSCTRL 권한</li> <li>• DATAACCESS 권한</li> <li>• DBADM 권한</li> <li>• SECADM 권한</li> <li>• SQLADM 권한</li> </ul> <p>SHOW DETAIL 매개변수를 사용하려는 경우, GET_INDEX_COLNAMES() UDF에 대해서도 EXECUTE 특권이 필요합니다.</p> <p>PUBLIC은 선언된 임시 테이블에 대한 모든 특권을 소유하므로 사용자는 이 명령을 사용하여 해당 연결 내에 존재하는 선언된 임시 테이블에 대한 정보를 표시할 수 있습니다.</p>



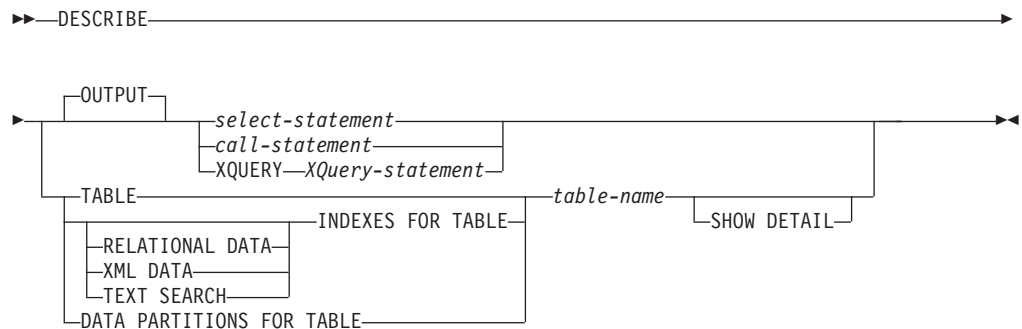
정보 표시를 위한 오브젝트	필요한 특권 또는 권한
테이블 또는 뷰의 데이터 파티션	<p>SYSCAT.DATAPARTITIONS 시스템 카탈로그 테이블에 대해 다음 특권이나 권한 중 하나:</p> <ul style="list-style-type: none"> <li>• SELECT 특권</li> <li>• ACCESSCTRL 권한</li> <li>• DATAACCESS 권한</li> <li>• DBADM 권한</li> <li>• SECADM 권한</li> <li>• SQLADM 권한</li> </ul> <p>PUBLIC은 선언된 임시 테이블에 대한 모든 특권을 소유하므로 사용자는 이 명령을 사용하여 해당 연결 내에 존재하는 선언된 임시 테이블에 대한 정보를 표시할 수 있습니다.</p>

- SYSTOOLSTMPSPACE 테이블 스페이스가 없는 경우, 위의 권한 중 하나 뿐만 아니라 SYSADM 또는 SYSCTRL 권한이 필요합니다.

## 필수 연결

데이터베이스. 내재된 연결이 사용 가능한 경우 디폴트 데이터베이스에 연결됩니다.

## 명령 구문



## 명령 매개변수

### OUTPUT

명령문의 출력을 설명해야 함을 나타냅니다. 이 키워드는 선택적입니다.

*select-statement* | *call-statement* | **XQUERY** *XQuery-statement*

필요한 정보에 대한 명령문을 식별합니다. 명령문은 CLP에 의해 자동으로 준비됩니다. XQuery문을 식별하려면 명령문 앞에 키워드 XQUERY를 붙이십시오. DESCRIBE OUTPUT문은 기술된 쿼리의 최종 결과 테이블의 SELECT 목록 파트로 컬럼이 명시적으로 지정된 경우 내재적으로 숨겨진 컬럼에 대한 정보만을 리턴합니다.

## TABLE *table-name*

설명될 테이블 또는 뷰를 지정합니다. *schema.table-name* 양식의 완전한 이름을 사용해야 합니다. 실제 테이블 대신 테이블의 별명을 사용할 수 없습니다. 내재적으로 숨겨진 컬럼에 대한 정보가 리턴됩니다.

DESCRIBE TABLE 명령은 각 컬럼에 대해 다음 정보를 나열합니다.

- 컬럼 이름
- 유형 스키마
- 유형 이름
- 길이
- 스케일
- 널(NULL)(예/아니오)

## INDEXES FOR TABLE *table-name*

인덱스가 설명되어야 하는 테이블 또는 뷰를 지정합니다. *schema.table-name* 양식의 완전한 이름을 사용하거나 *table-name*만 지정할 수 있습니다. 이 경우 자동으로 디폴트 스키마가 사용됩니다. 실제 테이블 대신 테이블의 별명을 사용할 수 없습니다.

DESCRIBE INDEXES FOR TABLE 명령은 테이블 또는 뷰의 각 인덱스에 대해 다음 정보를 나열합니다.

- 인덱스 스키마
- 인덱스 이름
- 고유한 규칙
- 컬럼 수
- 인덱스 유형

DESCRIBE INDEXES FOR TABLE 명령이 SHOW DETAIL 옵션과 함께 지정되면, 인덱스 이름이 18바이트보다 큰 경우 잘립니다. 인덱스 유형 옵션이 지정되지 않으면, 모든 인덱스 유형의 정보가 나열됩니다(관계형 데이터 인덱스, XML 데이터에 있는 인덱스 및 텍스트 검색 인덱스). 출력은 다음과 같은 추가 정보를 포함합니다.

- 관계형 데이터 인덱스, XML 경로 인덱스, XML 영역 인덱스 또는 XML 데이터에 있는 인덱스의 인덱스 ID
- XML 데이터에 있는 인덱스의 데이터 유형
- XML 데이터에 있는 인덱스의 해시 정보
- XML 데이터에 있는 인덱스의 최대 VARCHAR 길이
- XML 데이터에 있는 인덱스의 지정 XML 패턴
- 텍스트 검색 인덱스의 코드 페이지

- 텍스트 검색 인덱스의 언어
- 텍스트 검색 인덱스의 지정 형식
- 텍스트 검색 인덱스의 갱신 최소치
- 텍스트 검색 인덱스의 갱신 빈도
- 텍스트 검색 인덱스의 콜렉션 디렉토리
- 컬럼 이름

특정 인덱스 유형 전용 정보를 나열하려면 인덱스 유형을 지정하십시오. 다중 인덱스 유형 지정은 지원되지 않습니다.

### **RELATIONAL DATA**

RELATIONAL DATA 인덱스 유형 옵션이 SHOW DETAIL 옵션 없이 지정되면, 다음 정보만이 나열됩니다.

- 인덱스 스키마
- 인덱스 이름
- 고유한 규칙
- 컬럼 수

SHOW DETAIL이 지정되면, 컬럼 이름 정보도 나열됩니다.

### **XML DATA**

XML DATA 인덱스 유형 옵션이 SHOW DETAIL 옵션 없이 지정되면, 다음 정보만이 나열됩니다.

- 인덱스 스키마
- 인덱스 이름
- 고유한 규칙
- 컬럼 수
- 인덱스 유형

SHOW DETAIL이 지정되면, XML 데이터에 있는 인덱스의 다음 정보도 나열됩니다.

- 인덱스 ID
- 데이터 유형
- 해시
- 최대 Varchar 길이
- XML 패턴
- 컬럼 이름

## TEXT SEARCH

TEXT SEARCH 인덱스 유형 옵션이 SHOW DETAIL 옵션 없이 지정되면, 다음 정보만이 나열됩니다.

- 인덱스 스키마
- 인덱스 이름

SHOW DETAIL이 지정되면, 다음과 같은 텍스트 검색 인덱스 정보도 나열됩니다.

- 컬럼 이름
- 코드 페이지
- 언어
- 형식
- 갱신 최소치
- 갱신 빈도
- 콜렉션 디렉토리

TEXT SEARCH 옵션이 지정되고 텍스트 검색 옵션이 설치되지 않거나 적절하게 구성되지 않은 경우, 오류(SQLSTATE 42724)가 리턴됩니다.

컬럼에 나열된 정보는 DB2 텍스트 검색을 참조하십시오.

## DATA PARTITIONS FOR TABLE *table-name*

데이터 파티션이 설명되어야 하는 테이블 또는 뷰를 지정합니다. 테이블의 각 데이터 파티션에 대해 표시되는 정보는 파티션 ID 및 파티션 간격을 포함합니다. 결과는 파티션 ID 순서에 따라 정렬됩니다. *schema.table-name* 양식의 완전한 이름을 사용해야 합니다. 실제 테이블 대신 테이블의 별명을 사용할 수 없습니다. *schema*는 테이블 또는 뷰가 작성된 사용자 이름입니다.

DESCRIBE DATA PARTITIONS FOR TABLE 명령의 경우 출력에 다음 추가 정보가 있는 2차 테이블이 포함되도록 지정합니다.

- 데이터 파티션 시퀀스 ID
- SQL의 데이터 파티션 표현식

## SHOW DETAIL

DESCRIBE TABLE 명령의 경우, 출력에 와 함께 다음 추가 정보가 포함되도록 지정합니다.

- CHARACTER, VARCHAR 또는 LONG VARCHAR 컬럼이 FOR BIT DATA로 정의되었는지 여부
- 컬럼 번호
- 분산 키 순서

- 코드 페이지
- 디폴트값
- 테이블 파티션 유형(범위로 파티션된 테이블의 경우 이 출력은 원본 출력 아래에 표시됨)
- 파티션 키 컬럼(범위로 파티션된 테이블의 경우 이 출력은 원본 출력 아래에 표시됨)
- 인덱스에 사용되는 테이블 스페이스의 ID

예:

### SELECT문의 출력 설명

다음 예는 SELECT문을 설명하는 방법을 보여 줍니다.

```
db2 describe output select * from staff
```

컬럼 정보

컬럼 수: 7

데이터 유형	길이	컬럼 이름	이름 길이
500 SMALLINT	2	ID	2
449 VARCHAR	9	NAME	4
501 SMALLINT	2	DEPT	4
453 CHARACTER	5	JOB	3
501 SMALLINT	2	YEARS	5
485 DECIMAL	7,2	SALARY	6
485 DECIMAL	7,2	COMM	4

### CALL문의 출력 설명

명령문으로 작성된 스토어드 프로시저:

```
CREATE PROCEDURE GIVE_BONUS (IN EMPNO INTEGER,
                             IN DEPTNO INTEGER,
                             OUT CHEQUE INTEGER,
                             INOUT BONUS DEC(6,0))
...

```

...

다음 예는 CALL문의 출력을 설명하는 방법을 보여 줍니다.

```
db2 describe output call give_bonus(123456, 987, ?, 15000.)
```

컬럼 정보

컬럼 수: 2

데이터 유형	길이	컬럼 이름	이름 길이
497 INTEGER	4	CHEQUE	6
485 DECIMAL	6, 0	BONUS	5

프로시저에 하나 이상의 배열 유형 매개변수가 있으면, DESCRIBE 명령의 출력에는 배열 매개변수의 최대 카디널리티(cardinality)를 표시하는 하나의 추가적인 컬럼이 있습니다. 비어 있는 값은 매개변수가 배열이 아님을 표시합니다.

다음 명령문으로 배열 유형 및 프로시저를 작성합니다.

```
CREATE TYPE PRODUCT_LIST AS INTEGER ARRAY[100]
CREATE TYPE CUSTOMER_LIST AS INTEGER ARRAY[1000]

CREATE PROCEDURE DISCONTINUE_PROD (IN PROD_LIST PRODUCT_LIST,
                                   IN EFFECTIVE_DATE DATE,
                                   OUT NUM_PENDING_ORDERS INTEGER,
                                   OUT CUST_LIST CUSTOMER_LIST)
...

```

다음 예는 배열 매개변수가 있는 CALL문의 출력을 설명하는 방법을 보여 줍니다. 이전 예와 다른 유일한 형식 차이는 Max cardinality 컬럼입니다.

```
db2 describe output call discontinue_prod(ARRAY[12, 34, 26], '04/13/2006',?)
```

컬럼 정보

컬럼 수: 2

SQL 유형	유형 길이	컬럼 이름	이름 길이	최대 카디널리티
497 INTEGER	4	NUM_PENDING_ORDERS	17	
497 INTEGER	10	CUSTOMER_LIST	13	1000

### XQuery문의 출력 설명

XML 데이터 유형의 INFO 컬럼이 있는 CUSTOMER 테이블이 주어진 경우, 다음 예는 XQuery문을 설명하는 방법을 보여 줍니다.

```
db2 describe xquery for $cust in db2-fn:xmlcolumn("CUSTOMER.INFO") return $cust
```

컬럼 정보

컬럼 수: 1

SQL 유형	유형 길이	컬럼 이름	이름 길이
998 XML	0 1		1

키워드 XQUERY를 지정하지 않으면 SQL0104N이 리턴됩니다.

```
db2 describe for $cust in db2-fn:xmlcolumn("CUSTOMER.INFO") return $cust
SQL0104N "DESCRIBE" 다음에 예기치 않은 토큰 "for"이(가) 있었습니다.
예상되는 토큰으로는 "OUTPUT"이(가) 있습니다. SQLSTATE=42601
```

XQUERY 옵션을 지원하지 않는 하위 레벨 서버에 대해 DESCRIBE XQUERY 명령을 발행하면 해당 기능이 하위 레벨 서버에서 지원되지 않음을 표시하는 메시지 DB21108E가 리턴됩니다.

### 테이블 설명

다음 예는 테이블을 설명하는 방법을 보여 줍니다.

```
db2 describe table user1.department
```

Table: USER1.DEPARTMENT

컬럼	유형	유형
----	----	----

이름	스키마	이름	길이	스케일	널 (Null)
AREA	SYSIBM	SMALLINT		2	0 No
DEPT	SYSIBM	CHARACTER		3	0 No
DEPTNAME	SYSIBM	CHARACTER		20	0 Yes

다음 예는 세부사항과 함께 테이블을 설명하는 방법을 보여 줍니다. 이 예에서와 같이 테이블이 파티션된 경우 기존 출력 아래에 추가 세부사항이 표시됩니다. 파티션되지 않은 테이블의 경우 추가 테이블 표제를 표시하지 않습니다.

```
db2 describe table user1.employee show detail
```

컬럼 이름	유형 스키마	컬럼 이름	유형 이름	길이
FIRST	SYSIBM		0 CHARACTER	10
LAST	SYSIBM		1 CHARACTER	10

테이블은 범위로 파티션됨(다음 컬럼에서 순서가 지정됨):

```
-----
LAST
FIRST
```

### 테이블 인덱스 설명

다음 예는 테이블 인덱스를 설명하는 방법을 보여 줍니다. 해당 명령은 2개의 관계형 데이터 인덱스, 6개의 xml 데이터 인덱스, 2개의 텍스트 검색 인덱스 및 시스템 인덱스를 나열합니다.

```
db2 describe indexes for table user1.department
```

인덱스 스키마	인덱스 이름	고유 규칙	컬럼 수	인덱스 유형
SYSIBM	SQL070531145253450	D	-	XML DATA - REGIONS
SYSIBM	SQL070531145253620	U	1	XML DATA - PATH
USER1	RELIDX1	D	1	RELATIONAL DATA
USER1	RELIDX2	D	2	RELATIONAL DATA
SYSIBM	SQL070531145253650	P	1	RELATIONAL DATA
USER1	XMLIDX1	D	1	XML DATA - VALUES LOGICAL
SYSIBM	SQL070531154625650	D	1	XML DATA - VALUES PHYSICAL
USER1	XMLIDX2	D	1	XML DATA - VALUES LOGICAL
SYSIBM	SQL070531154626000	D	1	XML DATA - VALUES PHYSICAL
USER1	XMLIDX3	D	1	XML DATA - VALUES LOGICAL
SYSIBM	SQL070531154626090	D	1	XML DATA - VALUES PHYSICAL
USER1	XMLIDX4	D	1	XML DATA - VALUES LOGICAL
SYSIBM	SQL070531154626190	D	1	XML DATA - VALUES PHYSICAL
USER1	XMLIDX5	D	1	XML DATA - VALUES LOGICAL
SYSIBM	SQL070531154626290	D	1	XML DATA - VALUES PHYSICAL
USER1	XMLIDX6	D	1	XML DATA - VALUES LOGICAL
SYSIBM	SQL070531154626400	D	1	XML DATA - VALUES PHYSICAL
USER1	TXTIDX1	-	1	TEXT SEARCH
USER1	TXTIDX2	-	1	TEXT SEARCH

다음 명령은 테이블 USER1.DEPARTMENT의 관계형 데이터 인덱스를 나열합니다.

```
db2 describe relational data indexes for table user1.department
```

인덱스 스키마	인덱스 이름	고유 규칙	컬럼 수
-----	-----	-----	-----

SYSIBM	SQL070531145253650	P	1
USER1	RELIDX1	D	1
USER1	RELIDX2	D	2

다음 명령은 테이블 USER1.DEPARTMENT의 XML 데이터 인덱스를 나열합니다.

```
db2 describe xml data indexes for table user1.department
```

인덱스 스키마	인덱스 이름	고유 규칙	컬럼 수	인덱스 유형
SYSIBM	SQL070531145253450	D	-	XML DATA - REGIONS
SYSIBM	SQL070531145253620	U	1	XML DATA - PATH
USER1	XMLIDX1	D	1	XML DATA - VALUES LOGICAL
SYSIBM	SQL070531154625650	D	1	XML DATA - VALUES PHYSICAL
USER1	XMLIDX2	D	1	XML DATA - VALUES LOGICAL
SYSIBM	SQL070531154626000	D	1	XML DATA - VALUES PHYSICAL
USER1	XMLIDX3	D	1	XML DATA - VALUES LOGICAL
SYSIBM	SQL070531154626090	D	1	XML DATA - VALUES PHYSICAL
USER1	XMLIDX4	D	1	XML DATA - VALUES LOGICAL
SYSIBM	SQL070531154626190	D	1	XML DATA - VALUES PHYSICAL
USER1	XMLIDX5	D	1	XML DATA - VALUES LOGICAL
SYSIBM	SQL070531154626290	D	1	XML DATA - VALUES PHYSICAL
USER1	XMLIDX6	D	1	XML DATA - VALUES LOGICAL
SYSIBM	SQL070531154626400	D	1	XML DATA - VALUES PHYSICAL

다음 명령은 테이블 USER1.DEPARTMENT의 텍스트 검색 인덱스 정보를 나열합니다.

```
db2 describe text search indexes for table user1.department
```

인덱스 스키마	인덱스 이름
USER1	TXTIDX1
USER1	TXTIDX2

다음 명령은 파티션된 테이블 myDpartT의 파티션된 인덱스와 파티션되지 않은 인덱스 둘 다에 대한 정보를 나열합니다.

```
db2 describe indexes for table myDPartT
```

인덱스 스키마	인덱스 이름	고유 규칙	컬럼 컬럼	인덱스 파티션
NEWTON	IDXNDP	D	1	N
NEWTON	IDXDP	D	1	P

## 데이터 파티션 설명

다음 예는 데이터 파티션을 설명하는 방법을 보여 줍니다.

```
db2 describe data partitions for table user1.sales
```

PartitionId	Inclusive (y/n) Low Value	Inclusive (y/n) High Value
0	Y 2001,1	Y 2001,3
1	N 2001,3	Y 2001,6
3	N 2001,6	Y 2001,9



세부사항이 있는 데이터베이스 파티션 설명은 이전 예와 같은 출력을 리턴하며, 데이터 파티션에 대한 데이터가 저장된 파티션 ID 및 테이블 스페이스, 그리고 인덱스가 저장된 테이블 스페이스의 ID를 포함합니다.

```
db2 describe data partitions for table user1.employee show detail
```

PartitionId	Inclusive (y/n) Low Value	Inclusive (y/n) High Value
0	Y MINVALUE,MINVALUE	Y 'beck','kevin'
1	N 'beck','kevin'	N 'treece','jeff'
2	Y 'treece','jeff'	Y 'zhang','liping'
3	Y 'zyzyck',MINVALUE	Y MAXVALUE,MAXVALUE

PartitionId	PartitionName	TableSpId	LongTblSpId	IndexTblSpId	AccessMode	Status
0	PARTx	3	43	50	F	
1	PARTNew	13	13	13	N	A
2	PART3	31	33	35	F	
3	PART4	23	34	23	N	A



---

## 제 35 장 DETACH

논리적 DBMS 인스턴스 접속을 제거하고, 이 계층을 사용한 다른 논리적 연결이 없는 경우 물리적 통신 연결을 종료합니다.

### 권한 부여

없음

### 필수 연결

없음. 기존 인스턴스 접속을 제거합니다.

### 명령 구문

▶▶—DETACH—◀◀

### 명령 매개변수

없음



---

## 제 36 장 DROP CONTACT

로컬 시스템에 정의된 문의처 목록에서 문의처를 제거합니다. 문의처는 스케줄러 및 Health Monitor가 메시지를 보내는 사용자입니다. DAS(Database Administration Server) **contact\_host** 구성 매개변수의 설정은 목록이 로컬인지 또는 전역인지를 판별합니다.

### 권한 부여

없음

### 필수 연결

없음. 로컬 실행 전용: 이 명령은 리모트 연결에 사용할 수 없습니다.

### 명령 구문

►►—DROP CONTACT—*name*—————◄◄

### 명령 매개변수

**CONTACT** *name*

로컬 시스템에서 삭제될 문의처 이름



---

## 제 37 장 DROP CONTACTGROUP

로컬 시스템에 정의된 문의처 목록에서 문의처 그룹을 제거합니다. 문의처 그룹에는 스케줄러 및 Health Monitor가 메시지를 보내는 사용자 목록이 들어 있습니다. DAS(Database Administration Server) **contact\_host** 구성 매개변수의 설정은 목록이 로컬인지 또는 전역인지를 판별합니다.

### 권한 부여

없음

### 필요한 연결

없음

### 명령 구문

►►—DROP CONTACTGROUP—*name*—◄◄

### 명령 매개변수

**CONTACTGROUP** *name*

로컬 시스템에서 삭제될 문의처 그룹의 이름





---

## 제 38 장 DROP DATABASE

데이터베이스의 모든 로그 파일과 데이터베이스 콘텐츠를 삭제하고, 데이터베이스를 카탈로그 해제하고 데이터베이스 서브디렉토리를 삭제합니다.

### 범위

디폴트로, 이 명령은 db2nodes.cfg 파일에 나열된 모든 데이터베이스 파티션에 영향을 미칩니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*

### 필수 연결

인스턴스. 명시적 접속은 필요하지 않습니다. 데이터베이스가 리모트로 나열된 경우, 이 명령의 지속 기간 동안 리모트 데이터베이스 파티션 서버에 대한 인스턴스 접속이 설정됩니다.

### 명령 구문

```
➤➤ DROP DATABASE database-alias AT DBPARTITIONNUM
```

### 명령 매개변수

#### **DATABASE** *database-alias*

삭제될 데이터베이스의 별명을 지정합니다. 데이터베이스는 시스템 데이터베이스 디렉토리에 카탈로그되어야 합니다.

#### **AT DBPARTITIONNUM**

데이터베이스는 DROP DATABASE 명령으로 발행된 데이터베이스 파티션에서만 삭제됩니다. 이 매개변수는 DB2 웨어하우스 개정판 및 DPF(Database Partitioning Feature)에서 제공하는 유틸리티에 의해 사용되며, 일반 용도가 아닙니다. 이 매개변수의 부적절한 사용은 시스템에서 불일치를 유발할 수 있으므로, 주의해야 합니다.

## 예:

다음 예는 데이터베이스 별명 SAMPLE에 의해 참조되는 데이터베이스를 삭제합니다.

```
db2 drop database sample
```

## 사용 시 참고사항

DROP DATABASE는 모든 사용자 데이터 및 로그 파일을 삭제하며 데이터베이스의 모든 백업 및 리스토어 실행기록도 삭제합니다. 리스토어 조작이나, 데이터베이스를 리스토어하는 데 필요한 백업 실행기록 이후에 롤 포워드 복구를 위한 로그 파일이 필요한 경우, 이 명령을 발행하기 전에 해당 파일을 저장해야 합니다.

데이터베이스는 사용 중이지 않아야 하며, 데이터베이스를 삭제하기 전에 모든 사용자는 데이터베이스에서 연결을 끊어야 합니다.

데이터베이스를 삭제하려면, 데이터베이스가 시스템 데이터베이스 디렉토리에 카탈로그 되어야 합니다. 지정된 데이터베이스 별명만 시스템 데이터베이스 디렉토리에서 제거됩니다. 동일한 데이터베이스 이름을 가진 다른 별명이 있는 경우, 해당 항목이 남습니다. 삭제되는 데이터베이스가 로컬 데이터베이스 디렉토리의 마지막 항목인 경우, 로컬 데이터베이스 디렉토리는 자동 삭제됩니다.

DROP DATABASE가 리모트 클라이언트에서(또는 동일한 머신의 다른 인스턴스에서) 발행되는 경우, 지정된 별명은 시스템 데이터베이스 디렉토리에서 제거됩니다. 해당하는 데이터베이스 이름은 서버의 시스템 데이터베이스 디렉토리에서 제거됩니다.

## 호환성

버전 8 이전 버전과의 호환성:

- NODE 키워드를 DBPARTITIONNUM 대신 사용할 수 있습니다.

---

## 제 39 장 DROP DBPARTITIONNUM VERIFY

데이터베이스 파티션이 데이터베이스의 데이터베이스 파티션 그룹에 존재하는지 및 데이터베이스 파티션에서 이벤트 모니터가 정의되는지 여부를 검증합니다. 이 명령은 파티션된 데이터베이스 환경에서 데이터베이스 파티션을 삭제하기 전에 사용되어야 합니다.

### 범위

이 명령은 명령이 발행되는 데이터베이스 파티션에만 영향을 줍니다.

### 권한 부여

*sysadm*

### 명령 구문

▶▶—DROP DBPARTITIONNUM VERIFY—◀◀

### 명령 매개변수

없음

### 사용 시 참고사항

데이터베이스 파티션이 사용 중이 아님을 표시하는 메시지가 리턴되는 경우, STOP DATABASE MANAGER 명령을 DROP DBPARTITIONNUM과 함께 사용하여 db2nodes.cfg 파일에서 데이터베이스 파티션에 대한 항목을 제거하십시오. 그러면 데이터베이스 시스템에서 데이터베이스 파티션이 제거됩니다.

데이터베이스 파티션이 사용 중임을 표시하는 메시지가 리턴되는 경우 다음 조치를 수행해야 합니다.

1. 데이터베이스 파티션에 데이터가 들어있는 경우 데이터를 재분배하여 REDISTRIBUTE DATABASE PARTITION GROUP을 사용하여 데이터베이스 파티션에서 제거하십시오. REDISTRIBUTE DATABASE PARTITION GROUP 명령 또는 ALTER DATABASE PARTITION GROUP문에서 DROP DBPARTITIONNUM 옵션을 사용하여 데이터베이스에 대한 모든 데이터베이스 파티션 그룹에서 데이터베이스 파티션을 제거하십시오. 데이터베이스 파티션 그룹에 데이터베이스 파티션을 포함하는 각 데이터베이스에 대해 이 작업을 수행해야 합니다.
2. 데이터베이스 파티션에서 정의되는 모든 이벤트 모니터를 삭제(Drop)하십시오.

3. DROP DBPARTITIONNUM VERIFY를 다시 실행하여 데이터베이스가 더 이상 사용 중이 아닌지 확인하십시오.

## 호환성

버전 8 이전 버전과의 호환성:

- DBPARTITIONNUM에 키워드 NODE를 대체할 수 있습니다.

---

## 제 40 장 DROP TOOLS CATALOG

지정된 데이터베이스에서 지정된 카탈로그의 DB2 도구 카탈로그 테이블을 삭제(drop)합니다. 이 명령은 IBM Data Server Client에서는 유효하지 않습니다.

**경고:** 활성 도구 카탈로그를 삭제(drop)하면 더 이상 태스크를 스케줄할 수 없고 스케줄된 태스크는 실행되지 않습니다. 스케줄러를 활성화하려면 이전 도구 카탈로그를 활성화하거나 새로 작성해야 합니다.

### 범위

이 명령은 데이터베이스에 영향을 줍니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*

사용자에게는 또한 DAS(DB2 Administration Server) 구성 매개변수를 갱신하기 위해 DASADM이 필요합니다.

### 필수 연결

이 명령 처리 중 임시로 데이터베이스 연결이 설정됩니다.

### 명령 구문

```
►►—DROP TOOLS CATALOG—catalog-name—IN DATABASE—database-name—┐FORCE┘◄◄
```

### 명령 매개변수

**CATALOG** *catalog-name*

DB2 도구 카탈로그를 고유하게 식별하는 데 사용되는 이름. 카탈로그 테이블은 이 스키마에서 삭제됩니다.

**DATABASE** *database-name*

카탈로그 테이블이 들어 있는 로컬 데이터베이스에 연결하는 데 사용되는 이름.

**FORCE**

강제 옵션은 DB2 Administration Server 스케줄러를 강제로 중지하는 데 사용됩니다. 이 옵션이 지정되지 않고 스케줄러를 중지할 수 없으면 도구 카탈로그가 삭제되지 않습니다.

예:

```
db2 drop tools catalog cc in database toolsdb
```

```
db2 drop tools catalog in database toolsdb force
```

### 사용 시 참고사항

- DAS(DB2 Administration Server) 구성에서 **jdk\_path** 구성 매개변수를 Java에 대해 지원되는 최소 SDK 레벨로 설정해야 합니다.
- 이 명령은 로컬 DAS에서 스케줄러를 사용 불가능하게 하며 DB2 도구 카탈로그 데이터베이스 구성과 관련된 DAS 구성 매개변수를 재설정합니다.

---

## 제 41 장 ECHO

사용자가 표준 출력에 문자열을 쓸 수 있도록 허용합니다.

### 권한 부여

없음

### 필수 연결

없음

### 명령 구문

▶▶ ECHO character-string ▶▶

### 명령 매개변수

*character-string*

모든 문자열.

### 사용 시 참고사항

입력 파일이 표준 입력으로 사용되거나 주석이 명령 셸에 의해 해석되지 않고 인쇄되어야 하는 경우 ECHO 명령은 문자열을 표준 출력에 직접 인쇄합니다.

ECHO가 발행될 때마다 한 라인이 인쇄됩니다.

ECHO 명령은 상세(-v) 옵션의 영향을 받지 않습니다.





---

## 제 42 장 EDIT

지정된 편집 명령으로 사용자 지정 편집기를 시작합니다. 사용자가 편집을 완료하고 편집기 내용을 저장하고 편집기를 종료하면 사용자가 CLP 대화식 모드에서 명령을 실행할 수 있습니다.

### 범위

이 명령은 CLP 대화식 모드에서만 실행될 수 있습니다. 특히 CLP 명령 모드 또는 CLP 일괄처리 모드에서 실행할 수 없습니다.

### 권한 부여

없음

### 필수 연결

없음

### 명령 구문

▶ EDIT [E] [EDITOR=*editor*] [num] ▶▶

### 명령 매개변수

#### EDITOR

편집을 위해 지정된 편집기를 시작합니다. 이 매개변수가 지정되지 않으면 사용될 편집기가 다음 순서로 결정됩니다.

1. DB2\_CLP\_EDITOR 레지스트리 변수로 지정되는 편집기
2. VISUAL 환경 변수로 지정되는 편집기
3. EDITOR 환경 변수로 지정되는 편집기
4. 메모장 편집기(Windows 운영 체제), vi 편집기(UNIX 운영 체제)

*num* *num*이 양수인 경우 *num*에 해당하는 명령으로 편집기를 시작합니다. *num*이 음수인 경우 명령 실행기록의 최신 명령부터 역방향으로 카운팅하여 *num*에 해당하는 명령으로 편집기를 시작합니다. 0은 *num*에 유효한 값이 아닙니다. 이 매개변수가 지정되지 않으면 가장 최근에 실행된 명령으로 편집기를 시작합니다. 이는 *num*에 값 -1을 지정하는 것과 같습니다.

### 사용 시 참고사항

1. 지정된 편집기는 운영 체제의 PATH에 포함된 유효한 편집기여야 합니다.

2. HISTORY 명령을 실행하여 편집에 사용할 수 있는 가장 최근에 실행된 명령의 목록을 볼 수 있습니다.
3. EDIT 명령은 명령 실행기록에 기록되지 않습니다. 그러나 EDIT 명령을 사용하여 편집된 명령을 실행하도록 선택하는 경우 이 명령은 명령 실행기록에 기록됩니다.

## 제 43 장 EXPORT

데이터베이스의 데이터를 여러 외부 파일 형식 중 하나로 익스포트합니다. 사용자는 SQL SELECT문을 제공하거나 유형이 지정된 테이블에 대한 계층 정보를 제공하여 익스포트할 데이터를 지정합니다.

199 페이지의 『익스포트 유틸리티의 파일 유형 수정자』로의 빠른 링크.

### 권한 부여

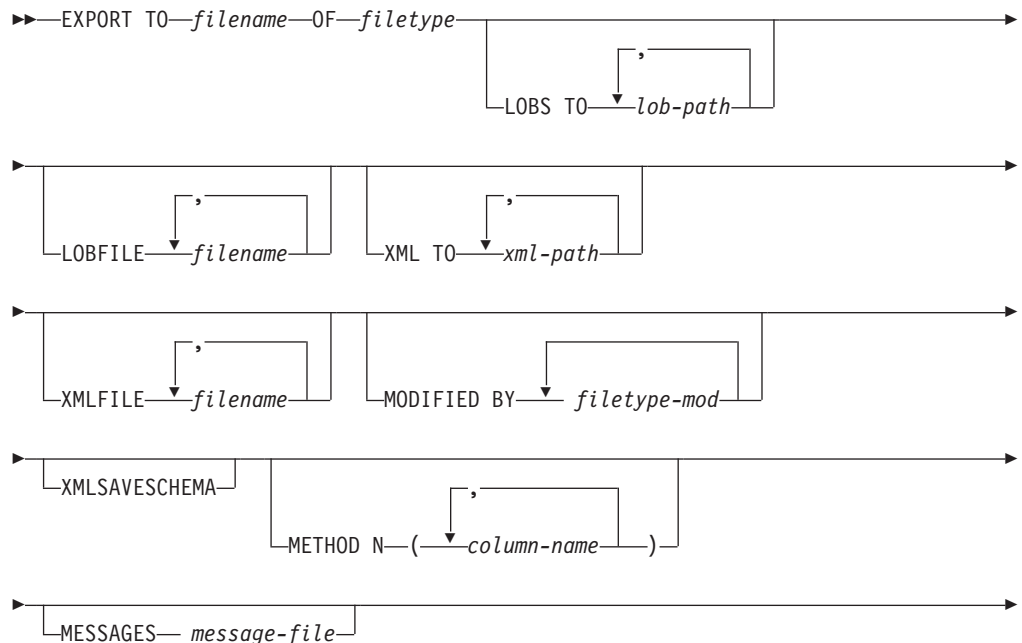
다음 중 하나가 필요합니다.

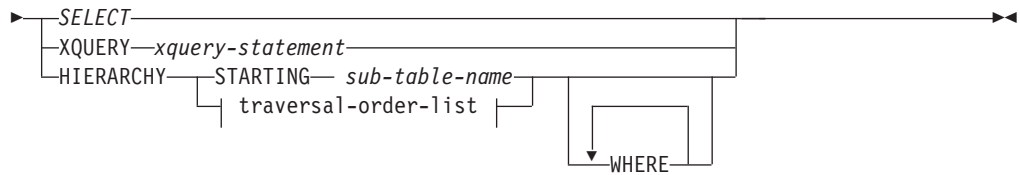
- `dataaccess` 권한
- 참여 중인 각 테이블 또는 뷰에 대한 CONTROL 또는 SELECT 특권

### 필수 연결

데이터베이스. 내재된 연결이 사용 가능한 경우 디폴트 데이터베이스에 연결됩니다. Linux, UNIX 또는 Windows 클라이언트로부터 Linux, UNIX 또는 Windows 데이터베이스 서버에 대한 유틸리티 액세스는 DB2 Connect 게이트웨이 또는 루프 백 환경을 통하지 않고 엔진을 통해 직접 연결되어야 합니다.

### 명령 구문





**traversal-order-list:**



**명령 매개변수**

**HIERARCHY traversal-order-list**

지정된 트래버스 순서를 사용하여 하위 계층 구조를 익스포트합니다. 모든 서브테이블은 PRE-ORDER 형식으로 나열되어야 합니다. 첫 번째 서브테이블 이름을 SELECT문의 목표 테이블 이름으로 사용합니다.

**HIERARCHY STARTING sub-table-name**

디폴트 트래버스 순서(ASC, DEL 또는 WSF 파일의 경우 OUTER 순서 또는 PC/IXF 데이터 파일에 저장된 순서)를 사용하여 sub-table-name에서 시작하는 하위 계층 구조를 익스포트하십시오.

**LOBFILE filename**

LOB 파일에 대한 하나 이상의 기본 파일을 지정합니다. 첫 번째 이름에 대한 이름 공간이 모두 사용되었으면 두 번째 이름을 사용하고 계속 이와 같이 합니다. 이는 내재적으로 LOBSINFILE 동작을 활성화합니다.

익스포트 조작 중에 LOB 파일을 작성할 경우, 파일 이름은 해당 목록의 현재 기본 이름을 현재 경로(lob-path의)에 추가한 후 시작할 3자리 시퀀스 번호 및 3자의 ID(lob)를 추가하여 구성됩니다. 예를 들어, 현재 LOB 경로가 /u/foo/lob/path/ 디렉토리이고 현재 LOB 파일 이름이 bar이면 작성되는 LOB 파일은 /u/foo/lob/path/bar.001.lob, /u/foo/lob/path/bar.002.lob 등입니다. 999가 사용되고 나면 LOB 파일 이름의 3자리 시퀀스 번호는 4자리가 되며, 9999가 사용되고 나면 4자리는 5자리가 되는 식으로 진행됩니다.

**LOBS TO lob-path**

LOB 파일이 저장될 디렉토리에 대한 하나 이상의 경로를 지정하십시오. LOB 경로 당 최소한 하나의 파일이 있으며 각 파일에는 최소한 하나의 LOB가 있습니다. 지정할 수 있는 최대 경로 수는 999입니다. 이는 내재적으로 LOBSINFILE 동작을 활성화합니다.

**MESSAGES message-file**

익스포트 조작 중 발생하는 경고 및 오류 메시지에 대한 목적지를 지정합니다.

파일이 이미 존재할 경우 익스포트 유틸리티는 정보를 추가합니다. *message-file* 이 생략될 경우 메시지는 표준 출력으로 쓰여집니다.

**METHOD N** *column-name*

출력 파일에서 사용될 하나 이상의 컬럼 이름을 지정합니다. 이 매개변수를 지정하지 않으면 테이블에 있는 컬럼 이름이 사용됩니다. 이 매개변수는 WSF 및 IXF 파일에 대해서만 유효하며 계층 데이터를 익스포트할 때는 유효하지 않습니다.

**MODIFIED BY** *filetype-mod*

파일 유형 수정자 옵션을 지정합니다. 199 페이지의 『익스포트 유틸리티의 파일 유형 수정자』를 참조하십시오.

**OF** *filetype*

출력 파일에 있는 데이터의 형식을 지정합니다.

- DEL(컬럼 식별자가 있는 ASCII 형식) - 여러 데이터베이스 관리 프로그램 및 파일 관리자 프로그램에서 사용됩니다.
- WSF(작업시트 형식) - 다음과 같은 프로그램에서 사용됩니다.
  - Lotus® 1-2-3®
  - Lotus Symphony

BIGINT 또는 DECIMAL 데이터를 익스포트할 때 DOUBLE 유형의 범위에 해당하는 값만 정확히 익스포트할 수 있습니다. 이 범위에 해당하지 않는 값은 익스포트되기는 하지만 이들 값을 다시 임포트하거나 로드할 때 운영 체제에 따라 올바르지 않은 데이터가 나타날 수 있습니다.

주: WSF 파일 형식에 대한 지원은 사용되지 않으므로 추후 릴리스에서 제거될 수 있습니다. 지원이 제거되기 전에 WSF 파일 대신 지원되는 파일 형식을 사용하여 시작하는 것이 좋습니다.

- IXF(Integration Exchange Format, PC 버전)는 독점 2진 형식입니다.

**SELECT**

익스포트될 데이터를 리턴하는 SELECT 또는 XQUERY문을 지정합니다. 명령문이 오류를 일으킬 경우 메시지 파일(또는 표준 출력)에 메시지가 쓰여집니다. 오류 코드가 SQL0012W, SQL0347W, SQL0360W, SQL0437W 또는 SQL1824W 중 하나이면 익스포트 작업을 계속 수행하고 그렇지 않으면 중지합니다.

**TO** *filename*

데이터가 익스포트될 파일 이름을 지정합니다. 파일에 대한 전체 경로를 지정하지 않으면 익스포트 유틸리티는 현재 디렉토리 및 디폴트 드라이브를 목적으로 사용합니다.

이미 존재하는 파일 이름을 지정할 경우 익스포트 유틸리티는 파일 내용을 겹쳐쓰고 정보를 추가하지 않습니다.

#### **XMLFILE filename**

XML 파일에 대한 하나 이상의 기본 파일 이름을 지정합니다. 첫 번째 이름에 대한 이름 공간이 모두 사용되었으면 두 번째 이름을 사용하고 계속 이와 같이 합니다.

익스포트 조작 중 XML 파일을 작성할 때 파일 이름은 이 목록의 현재 기본 이름을 현재 경로(*xml-path*로부터)에 추가하고 3자리 숫자 시퀀스 번호를 추가한 후 3자 ID xml을 추가합니다. 예를 들어, 현재 XML 경로가 /u/foo/xml/path/ 디렉토리이고 현재 XML 파일 이름이 bar이면 작성되는 XML 파일은 /u/foo/xml/path/bar.001.xml, /u/foo/xml/path/bar.002.xml 등입니다.

#### **XML TO xml-path**

XML 파일이 저장될 디렉토리에 대한 하나 이상의 경로를 지정하십시오. XML 경로당 최소한 하나의 파일이 있으며 각 파일에는 최소한 하나의 XDM(XQuery Data Model) 인스턴스가 있습니다. 두 개 이상의 경로를 지정하면 XDM 인스턴스는 해당 경로에 균일하게 분산됩니다.

#### **XMLSAVESCHEMA**

모든 XML 컬럼에 대해 XML 스키마 정보가 저장되어야 함을 지정합니다. 각 익스포트된 XM 문서가 삽입될 때 XML 스키마에 대해 유효성이 확인되었으면 해당 스키마의 완전한 SQL ID는 해당 XDS(XML Data Specifier) 내에 (SCH) 속성으로 저장됩니다. 익스포트된 문서가 XML 스키마에 대해 유효성이 확인되지 않았거나 데이터베이스에 스키마 오브젝트가 더 이상 존재하지 않으면 해당 XDS에 SCH 속성이 포함되지 않습니다.

SQL ID의 스키마 및 이름 부분은 XML 스키마에 해당하는 SYSCAT.XSROBJECTS 카탈로그 테이블 행에 "OBJECTSCHEMA" 및 "OBJECTNAME" 값으로 저장됩니다.

XMLSAVESCHEMA 옵션은 올바른 형식의 XML 문서를 생성하지 않는 XQuery 시퀀스와는 호환 가능하지 않습니다.

#### **예:**

다음 예는 SAMPLE 데이터베이스에 있는 STAFF 테이블의 정보를 myfile.ixf 파일로 익스포트하는 방법을 보여 줍니다. 해당 출력은 IXF 형식으로 생성됩니다. 명령을 발행하기 전에 먼저 SAMPLE 데이터베이스에 연결해야 합니다. DB2 Connect를 통해 데이터베이스 연결을 작성할 때 외에는 인덱스 정의(있는 경우)가 출력 파일에 저장됩니다.

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

다음 예는 SAMPLE 데이터베이스에 있는 STAFF 테이블에서 Department 20의 직원에 대한 정보를 익스포트하는 방법을 보여 줍니다. 출력은 IXF 형식으로 awards.ixf 파일로 쓰여집니다. 명령을 발행하기 전에 먼저 SAMPLE 데이터베이스에 연결해야 합니다. 또한, 테이블의 실제 컬럼 이름은 'department'가 아니라 'dept'입니다.

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

다음 예는 LOB를 DEL 파일로 익스포트하는 방법을 보여 줍니다.

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

다음 예는 첫 번째 디렉토리에 맞지 않을 수 있는 파일에 대해 2차 디렉토리를 지정하고 LOB를 DEL 파일로 익스포트하는 방법을 보여 줍니다.

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

다음 예는 작은따옴표를 문자열 분리문자로 사용하고 세미콜론을 컬럼 분리문자로 사용하며 쉼표를 소수점으로 사용하여 데이터를 DEL 파일로 익스포트하는 방법을 보여 줍니다. 데이터를 다시 데이터베이스로 임포트할 때도 동일한 규칙을 사용해야 합니다.

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

## 사용 시 참고사항

- 익스포트 작업을 시작하기 전에 반드시 모든 테이블 작업을 완료하고 모든 잠금을 릴리스하십시오. 이는 WITH HOLD로 열린 모든 커서를 닫고 COMMIT를 발행하거나 ROLLBACK을 발행하여 수행할 수 있습니다.
- SELECT문에서 테이블 별명을 사용할 수 있습니다.
- 메시지 파일에 넣어진 메시지는 메시지 검색 서비스에서 리턴된 정보를 포함합니다. 각 메시지는 새 행에서 시작합니다.
- 익스포트 유틸리티는 DEL 형식 파일로 익스포트하기 위해 길이가 254를 초과하는 문자 컬럼을 선택할 때마다 경고 메시지를 생성합니다.
- 데이터베이스 간에 데이터를 이동하려면 PC/IXF 임포트를 사용해야 합니다. 행 구분자가 있는 문자 데이터를 컬럼 식별자가 있는 ASCII(DEL) 파일로 익스포트하며 텍스트 전송 프로그램으로 이를 처리할 경우 행 구분자가 있는 필드는 줄어들거나 확장됩니다.
- 동일한 클라이언트에서 소스 또는 목표 데이터베이스에 모두 액세스할 수 있는 경우 파일 복사 단계는 필요하지 않습니다.

- DB2 Connect은 OS/390용 DB2, VM 및 VSE용 DB2, OS/400®용 DB2와 같은 DRDA 서버에서 테이블을 익스포트하는 데 사용할 수 있습니다. PC/IXF 익스포트만 지원됩니다.
- IXF 형식으로 익스포트할 때, ID가 IXF 형식에서 지원하는 최대 크기를 초과하는 경우, 익스포트에 성공하지만 결과 데이터 파일은 CREATE 모드를 사용하는 추후 임포트 조작에서는 사용될 수 없습니다. SQL27984W가 리턴됩니다.
- Windows의 디스켓으로 익스포트하고 단일 디스켓의 용량보다 큰 데이터가 있는 테이블을 익스포트할 때, 시스템은 다른 디스켓을 위해 프롬프트를 표시하며, 다중 파트 PC/IXF 파일(다중 볼륨 PC/IXF 파일 또는 논리 분할 PC/IXF 파일이라고도 함)이 생성되어 별도의 디스켓에 저장됩니다. 마지막의 경우를 제외하고 각 파일에는, 파일이 논리적으로 분할되었음을 표시하고 다음 파일을 찾을 곳을 나타내기 위해 쓰여진 DB2 CONTINUATION RECORD(또는 짧게 "AC" 레코드)가 있습니다. AIX 시스템으로 해당 파일을 전송하여, 임포트 및 로드 유틸리티에서 읽을 수 있습니다. 익스포트 유틸리티는 AIX 시스템에서 호출될 때 다중 파트 PC/IXF 파일을 작성하지 않습니다. 자세한 사용을 위해서는 IMPORT 명령 또는 LOAD 명령을 참조하십시오.
- 익스포트 유틸리티는 제공된 SELECT문이 SELECT \* FROM tablename 형식인 경우 해당 테이블의 NOT NULL WITH DEFAULT 속성을 IXF 파일에 저장합니다.
- 유형이 지정된 테이블을 익스포트할 때 subselect문은 단지 목표 테이블 이름과 WHERE 절을 지정하여 표현할 수 있습니다. 계층 구조를 익스포트할 때는 Fullselect 및 select-statement를 지정할 수 없습니다.
- IXF 이외의 파일 형식에서는 트래버스 순서 목록을 지정할 것을 권장합니다. 이는 이 순서가 DB2에게 계층 구조를 트래버스하는 방법과 어떤 서브테이블을 익스포트할지를 알려주기 때문입니다. 이 목록을 지정하지 않으면 해당 계층 구조에 있는 모든 테이블이 익스포트되고 디폴트 순서는 OUTER 순서가 됩니다. 대안은 OUTER 기능이 제공하는 순서인 디폴트 순서를 사용하는 방법입니다.
- 임포트 조작 동안 동일한 트래버스 순서를 사용하십시오. 로드 유틸리티는 계층 구조 또는 하위 계층 구조를 지원하지 않습니다.
- 행이 보호 설정된 테이블에서 데이터를 익스포트할 때 세션 권한 부여 ID가 보유하는 LBAC 증명서가 익스포트되는 행을 제한할 수 있습니다. 세션 권한 부여 ID가 읽기 권한을 가지고 있지 않은 행은 익스포트되지 않습니다. 오류나 경고는 표시되지 않습니다.
- 세션 권한 부여 ID가 보유하는 LBAC 증명서가 익스포트에 포함된 하나 이상의 보호 설정된 컬럼에서 읽기를 허용하지 않을 경우 익스포트를 실패하고 오류(SQLSTATE 42512)가 리턴됩니다.
- 익스포트 패키지는 DATETIME ISO 형식을 사용하여 바운드되므로 문자열 표현식으로 캐스트할 때 모든 날짜/시간/시간소인 값은 ISO 형식으로 변환됩니다. CLP 패키지



지는 DATETIME LOC 형식(로케일 특정 형식)을 사용하여 바운드리프르 CLP DATETIME 형식이 ISO 형식과 다른 경우 CLP 및 익스포트 간에 동작이 불일치 할 수 있습니다. 예를 들어, 다음 SELECT문에서는 예상대로 결과가 리턴됩니다.

```
db2 select col2 from tab1 where char(col2)='05/10/2005';
COL2
-----
05/10/2005
05/10/2005
05/10/2005
3개의 레코드가 선택되었습니다.
```

그러나 동일한 Select절을 사용한 Export 명령에서는 예상대로 결과가 리턴되지 않습니다.

```
db2 export to test.del of del select col2 from test
where char(col2)='05/10/2005';
익스포트된 행 수: 0
```

이제 LOCALE 날짜 형식을 ISO 형식으로 바꾸면 예상대로 결과가 리턴됩니다.

```
db2 export to test.del of del select col2 from test
where char(col2)='2005-05-10';
익스포트된 행 수: 3
```

## 익스포트 유틸리티의 파일 유형 수정자

표 8. 익스포트 유틸리티의 유효한 파일 유형 수정자: 모든 파일 형식

수정자	설명
lobsinfile	<p><i>lob-path</i>는 LOB 데이터를 포함하는 파일에 대한 경로를 지정합니다.</p> <p>각 경로는 LLS(Lob Location Specifier)에 의해 포인트되는 최소한 하나의 LOB를 포함하는 최소한 하나의 파일을 데이터 파일에 포함합니다. LLS는 LOB 파일 경로에 저장된 파일에서 LOB 위치의 문자열 표시입니다. LLS의 형식은 <i>filename.ext.nnn.mmm</i>이며, 여기서 <i>filename.ext</i>는 LOB를 포함하는 파일의 이름이며, <i>nnn</i>은 파일 내에서 LOB의 오프셋을 바이트로 나타낸 것이며, <i>mmm</i>은 LOB의 길이를 바이트로 나타낸 것입니다. 예를 들어, 문자열 db2exp.001.123.456/가 데이터 파일에 저장되는 경우, LOB는 db2exp.001 파일에서 오프셋 123에 위치하며 456바이트 길이입니다.</p> <p>EXPORT 사용 시 『lobsinfile』 수정자를 지정하는 경우, LOB 데이터는 LOBS TO 절에서 지정하는 위치에 있습니다. 그렇지 않으면 LOB 데이터는 데이터 파일 디렉토리로 보냅니다. LOBS TO 절은 LOB 파일이 저장될 디렉토리에 대한 하나 이상의 경로를 지정합니다. LOB 경로 당 최소한 하나의 파일이 있으며 각 파일에는 최소한 하나의 LOB가 있습니다. LOBS TO 또는 LOBFILE 옵션은 내재적으로 LOBSINFILE 동작을 활성화합니다.</p> <p>널(NULL) LOB를 표시하려면, 크기를 -1로 입력하십시오. 크기가 0으로 지정되면, 길이가 0인 LOB로 처리됩니다. 길이가 -1인 널(NULL) LOBS의 경우, 오프셋 및 파일 이름은 무시됩니다. 예를 들어, 널(NULL) LOB의 LLS는 db2exp.001.7.-1/입니다.</p>
xmlinsefiles	<p>각 XDM(XQuery Data Model) 인스턴스는 별도의 파일에 쓰여집니다. 디폴트로 복수의 값은 동일한 파일에서 함께 병합됩니다.</p>
lobsinsefiles	<p>각 LOB 값은 별도의 파일에 쓰여집니다. 디폴트로 복수의 값은 동일한 파일에서 함께 병합됩니다.</p>

표 8. 익스포트 유틸리티의 유효한 파일 유형 수정자: 모든 파일 형식 (계속)

수정자	설명
xmlnodeclaration	XDM 인스턴스는 XML 선언 태그 없이 쓰여집니다. 디폴트로 XDM 인스턴스는 처음에 인코딩 속성을 포함하는 XML 선언 태그를 가지고 익스포트됩니다.
xmlchar	XDM 인스턴스는 문자 코드 페이지에 쓰여집니다. 문자 코드 페이지는 codepage 파일 유형 수정자에서 지정한 값이거나, 지정되지 않은 경우 응용프로그램 코드 페이지입니다. 디폴트로, XDM 인스턴스는 유니코드로 쓰여집니다.
xmlgraphic	xmlgraphic 수정자가 EXPORT 명령에서 지정된 경우, 익스포트된 XML 문서는 응용프로그램 코드 페이지나 codepage 파일 유형 수정자와 관계없이 UTF-16 코드 페이지로 인코딩됩니다.

표 9. 익스포트 유틸리티의 유효한 파일 유형 수정자: DEL(컬럼 식별자가 있는 ASCII) 파일 형식

수정자	설명
chardelx	x는 단일 문자열 분리문자입니다. 디폴트값은 큰따옴표(")입니다. 문자열을 묶기 위해 큰따옴표 대신 지정된 문자가 사용됩니다. <sup>2</sup> 명시적으로 큰따옴표를 문자열 분리문자로 지정하려는 경우 다음과 같이 지정해야 합니다.  modified by charde1"  다음과 같이 작은따옴표(')를 문자열 분리문자로 지정할 수도 있습니다.  modified by charde1''
codepage=x	x는 ASCII 문자열입니다. 해당 값은 출력 데이터 세트에서 데이터의 코드 페이지로서 해석됩니다. 익스포트 조작 중 응용프로그램 코드 페이지로부터 이 코드 페이지로 문자 데이터를 변환합니다.  순수 DBCS(그래픽), 혼합 DBCS 및 EUC의 경우, 분리문자는 x00 - x3F의 범위로 제한됩니다. codepage 수정자는 lobsinfile 수정자와 함께 사용될 수 없습니다.
coldelx	x는 단일 문자 컬럼 분리문자입니다. 디폴트값은 쉼표(,)입니다. 지정된 문자가 쉼표 대신 사용되어 컬럼 끝을 나타냅니다. <sup>2</sup>  다음 예에서, coldel;은 익스포트 유틸리티가 익스포트된 데이터의 컬럼 분리문자로 세미콜론 문자(;)를 사용하게 합니다.  db2 "export to temp of del modified by coldel; select * from staff where dept = 20"
decplusblank	플러스 부호 문자. 플러스 부호(+) 대신 공백을 양의 10진수 값 앞에 붙입니다. 디폴트 조치는 플러스 부호를 양의 10진수 값 앞에 붙이는 것입니다.
decptx	x는 소수점 문자로 마침표를 대신하는 단일 문자입니다. 디폴트값은 마침표(.)입니다. 마침표 대신 지정된 문자가 소수점 문자로 사용됩니다. <sup>2</sup>
nochardel	컬럼 데이터는 문자 분리문자 안에 두지 않습니다. DB2를 사용하여 데이터를 임포트하거나 로드하려는 경우 이 옵션을 지정하지 말아야 합니다. 이 옵션은 문자 분리문자가 없는 벤더 데이터 파일을 지원하기 위해 제공됩니다. 부적절한 사용은 데이터 유실이나 손상을 초래할 수 있습니다.  chardelx 또는 nodoublede1에는 이 옵션을 지정할 수 없습니다. 이들은 상호 배타적인 옵션입니다.
nodoublede1	2바이트 분리문자를 인식하지 않습니다. <sup>2</sup>

표 9. 익스포트 유틸리티의 유효한 파일 유형 수정자: DEL(컬럼 식별자가 있는 ASCII) 파일 형식 (계속)

수정자	설명
stripzeros	<p>익스포트된 모든 10진수 컬럼에서 선행 영(0)을 제거합니다.</p> <p>다음 예를 고려하십시오.</p> <pre>db2 create table decimalTable ( c1 decimal( 31, 2 ) ) db2 insert into decimalTable values ( 1.1 )</pre> <pre>db2 export to data of del select * from decimalTable</pre> <pre>db2 export to data of del modified by STRIPLZEROS select * from decimalTable</pre> <p>첫 번째 익스포트 조작에서, 익스포트된 파일 데이터의 콘텐츠는 +0000000000000000000000000000001.10입니다. stripzeros 수정자를 제외하고 첫 번째와 동일한 두 번째 조작에서, 익스포트된 파일 데이터의 콘텐츠는 +1.10입니다.</p>

표 9. 익스포트 유틸리티의 유효한 파일 유형 수정자: DEL(컬럼 식별자가 있는 ASCII) 파일 형식 (계속)

수정자	설명
timestampformat="x"	<p>x는 소스 파일에서 시간소인의 형식입니다.<sup>4</sup> 유효한 시간소인 요소는 다음과 같습니다.</p> <p>YYYY - Year(0000 - 9999 사이의 4자리 숫자)  M - Month(1 - 12 사이의 1 또는 2자리 숫자)  MM - Month(01 - 12 사이의 2자리 숫자,  M 및 MMM은 상호 독점적)  MMM - Month(월 이름으로 사용되는 3자리 문자로 대소문자를  구분하지 않으며,  M 및 MM은 상호 독점적)  D - Day(1 - 31 사이의 1 또는 2자리 숫자)  DD - Day(1 - 31 사이의 2자리 숫자, D와는 상호 독점적)  DDD - Day of the year(001 - 366 사이의 3자리 숫자,  다른 일 또는 월 요소와 상호 독점적)  H - Hour(12시간 시스템의 경우 0 - 12 범위의 1 또는 2자리 숫자,  24시간 시스템의 경우 0 - 24)  HH - Hour(12시간 시스템의 경우 0 - 12 범위의 2자리 숫자,  24시간 시스템의 경우 0 - 24,  H와는 상호 독점적)  M - Minute(0 - 59 사이의 1 또는 2자리 숫자)  MM - Minute(0 - 59 사이의 2자리 숫자,  M, 분과 상호 독점적)  S - Second(0 - 59 사이의 1 또는 2자리 숫자)  SS - Second(0 - 59 사이의 2자리 숫자,  S와 상호 독점적)  SSSSS - Second of the day after midnight(  00000 - 86399 사이의 5자리 숫자, 다른 시간 요소와  상호 독점적)  U(1 - 12회)  - Fractional seconds(U 어커런스 수는 각 자리가  0 - 9 사이의 자리 수를 나타냄)  TT - Meridian indicator(AM 또는 PM)</p> <p>다음은 시간소인 형식의 예입니다.</p> <p>"YYYY/MM/DD HH:MM:SS.UUUUUU"</p> <p>MMM 요소는 다음 값을 생성합니다. 'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug',  'Sep', 'Oct', 'Nov' 및 'Dec'. 'Jan'은 1월이며, 'Dec'는 12월과 같습니다.</p> <p>다음 예는 'schedule'이라는 테이블에서 사용자 정의 시간소인을 포함하는 데이터를 익스포트  하는 방법을 설명합니다.</p> <pre>db2 export to delfile2 of del modified by timestampformat="yyyymm.dd hh:mm tt" select * from schedule</pre>

표 10. 익스포트 유틸리티의 유효한 파일 유형 수정자: IXF 파일 형식

수정자	설명
codepage=x	<p>x는 ASCII 문자열입니다. 해당 값은 출력 데이터 세트에서 데이터의 코드 페이지로서 해석  됩니다. 익스포트 조작 중 이 코드 페이지로부터 응용프로그램 코드 페이지로 문자 데이터를  변환합니다.</p> <p>순수 DBCS(그래픽), 혼합 DBCS 및 EUC의 경우, 분리문자는 x00 - x3F의 범위로 제한되  니다. codepage 수정자는 lobsinfile 수정자와 함께 사용될 수 없습니다.</p>

표 11. 익스포트 유틸리티의 유효한 파일 유형 수정자: WSF 파일 형식<sup>6</sup>

수정자	설명
1	Lotus 1-2-3 릴리스 1 또는 Lotus 1-2-3 릴리스 1a와 호환 가능한 WSF 파일을 작성합니다. <sup>5</sup> 이것이 디폴트입니다.
2	Lotus Symphony 릴리스 1.0과 호환 가능한 WSF 파일을 작성합니다. <sup>5</sup>
3	Lotus 1-2-3 버전 2 또는 Lotus Symphony 릴리스 1.1과 호환 가능한 WSF 파일을 작성합니다. <sup>5</sup>
4	DBCS 문자를 포함하는 WSF 파일을 작성합니다.

주:

1. MODIFIED BY 옵션과 함께 지원되지 않는 파일 유형을 사용하려고 시도하는 경우 익스포트 유틸리티는 경고를 발행하지 않습니다. 이런 경우, 익스포트 조작에 실패하며 오류 코드가 리턴됩니다.

2. 데이터 이동을 위한 분리문자 고려사항은 분리문자를 겹쳐쓰기할 때 사용할 수 있는 문자에 적용하는 제한사항을 나열합니다.

3. 정상적인 익스포트 유틸리티 쓰기

- YYYYMMDD 형식의 날짜 데이터
- "YYYY-MM-DD" 형식의 문자(날짜) 데이터
- "HH.MM.SS" 형식의 시간 데이터
- "YYYY-MM-DD-HH .MM.SS.aaaaaa" 형식의 시간 소인 데이터

익스포트 조작의 SELECT문에서 지정된 날짜 및 시간 컬럼에 포함된 데이터도 다음 형식입니다.

4. 시간소인 형식의 경우, 문자 M을 사용하는 month 및 minute 디스크립터 간의 모호함을 피하려면 주의해야 합니다. month 필드는 다른 날짜 필드와 인접해야 합니다. minute 필드는 다른 시간 필드와 인접해야 합니다. 다음은 앰비규어스 시간 소인 형식입니다.

- "M"(month 또는 minute일 수 있음)
- "M:M"(month 및 minute 구분 가능?)
- "M:YYYY:M"(둘 다 month로 해석됨)
- "S:M:YYYY"(둘 다 시간 값 및 날짜 값에 인접)

앰비규어스 경우, 유틸리티는 오류 메시지를 발행하며, 조작에 실패합니다.

다음은 명확한 시간소인 형식입니다.

- "M:YYYY" (Month)
- "S:M" (Minute)
- "M:YYYY:S:M" (Month....Minute)
- "M:H:YYYY:M:D" (Minute....Month)

5. 또한 이러한 파일은 Lotus 1-2-3의 경우 L, Symphony의 경우 S를 filetype-mod 매개변수 문자열에 지정하여 특정 제품으로 방향지정할 수 있습니다. 하나의 값 또는 제품 지정자만 지정할 수 있습니다. WSF 파일 형식에 대한 지원은 사용되지

않으므로 추후 릴리스에서 제거될 수 있습니다. 지원이 제거되기 전에 WSF 파일 대신 지원되는 파일 형식을 사용하여 시작하는 것이 좋습니다.

6. WSF 파일 형식은 XML 컬럼에서 지원되지 않습니다. 이 파일 형식에 대한 지원은 사용되지 않으므로 추후 릴리스에서 제거될 수 있습니다. 지원이 제거되기 전에 WSF 파일 대신 지원되는 파일 형식을 사용하여 시작하는 것이 좋습니다.
7. XMLFILE 또는 XML TO 절 중 어느 하나도 지정되지 않아도 모든 XDM 인스턴스는 기본 데이터 파일에서 분리된 XML 파일에 쓰여집니다. 디폴트로, XML 파일은 익스포트된 데이터 파일의 경로에 쓰여집니다. 디폴트로 XML 파일의 기본 이름은 익스포트된 데이터 파일의 이름으로 확장자가 ".xml"입니다.
8. XMLNODEDECLARATION 파일 유형 수정자가 지정된 경우를 제외하고, 모든 XDM 인스턴스는 처음에 인코딩 속성을 포함하는 XML 선언을 이용하여 쓰여집니다.
9. XMLCHAR 또는 XMLGRAPHIC 파일 유형 수정자가 지정된 경우를 제외하고, 디폴트로 모든 XDM 인스턴스는 유니코드로 쓰여집니다.
10. XML 데이터 및 LOB 데이터의 디폴트 경로는 기본 데이터 파일 경로입니다. 디폴트 XML 파일 기본 이름이 주 데이터 파일입니다. 디폴트 LOB 파일 기본 이름이 주 데이터 파일입니다. 예를 들어, 주 데이터 파일이

```
/mypath/myfile.del
```

이고 XML 데이터 및 LOB 데이터의 디폴트 경로는

```
/mypath"
```

이며 디폴트 XML 파일 기본 이름은

```
myfile.del
```

이고 디폴트 LOB 파일 기본 이름은 다음과 같은 경우,

```
myfile.del
```

LOB 파일을 생성하려면 LOBSINFILE 파일 유형 수정자를 지정해야 합니다.

11. 익스포트 유틸리티는 각 LOB 파일이나 XML 파일에 숫자 ID를 추가합니다. ID는 0으로 패드된 시퀀스 값인 3자리 숫자로 시작하며

```
.001
```

에서 시작합니다. 999번째 LOB 파일 또는 XML 파일 이후 ID는 더 이상 0으로 패드되지 않습니다. 예를 들어, 1000번째 LOG 파일이나 XML 파일의 확장자는

```
.1000
```

입니다. 숫자 ID 다음에는 데이터 유형을 나타내는 세 자 유형 ID가 표시됩니다(

```
.lob
```

또는

.xml을 나타냄)

. 예를 들어, 생성된 LOB 파일 이름의 형식은  
myfile.del.001.lob

이고 생성된 XML 파일 이름의 형식은 다음과 같습니다.

myfile.del.001.xml

12. XQuery를 지정하여 잘 양식화된 문서가 아닌 익스포트 유틸리티 익스포트 XDM 인스턴스를 갖는 것이 가능합니다. 그러나 XML 컬럼은 완전한 문서만을 포함하므로, 해당 익스포트된 문서를 직접 XML 컬럼에 임포트하거나 로드할 수 없습니다.





---

## 제 44 장 FORCE APPLICATION

강제로 로컬 또는 리모트 사용자 또는 응용프로그램이 시스템을 해제하도록 하여 서버에서 유지보수를 할 수 있도록 합니다.

**경고:** 인터럽트할 수 없는 조작(예를 들어, RESTORE DATABASE)이 강제 수행된 경우 데이터베이스를 사용할 수 있게 되기 전에 먼저 이 조작을 재실행해야 합니다.

### 범위

이 명령은 \$HOME/sql1lib/db2nodes.cfg 파일에 나열된 모든 데이터베이스 파티션에 영향을 줍니다.

파티션된 데이터베이스 환경에서는 이 명령을 강제되고 있는 응용프로그램의 코디네이터 데이터베이스 파티션에서 발행하지 않아도 됩니다. 파티션된 데이터베이스 환경에 있는 모든 노드(데이터베이스 파티션 서버)에서 발행할 수 있습니다.

### 권한 부여

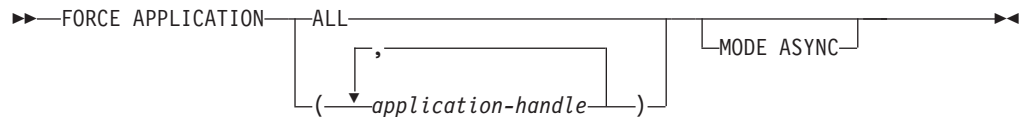
다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*

### 필수 연결

인스턴스. 강제로 사용자가 리모트 서버를 해제하게 하려면 먼저 이 서버에 연결되어야 합니다. 연결되어 있지 않은 경우 이 명령은 로컬로 실행됩니다.

### 명령 구문



### 명령 매개변수

#### FORCE APPLICATION

**ALL** 모든 응용프로그램이 데이터베이스에서 연결이 끊어집니다.

### *application-handle*

종료될 에이전트를 지정합니다. LIST APPLICATIONS 명령을 사용하여 값을 나열하십시오.

### MODE ASYNC

이 명령은 모든 지정된 사용자가 종료될 때까지 대기하지 않고 바로 리턴합니다. 함수가 발행되거나 오류(예: 유효하지 않은 구문)가 발견되면 즉시 리턴합니다.

현재 이 모드만 지원됩니다.

### 예:

다음 예는 *application-handle* 값이 41408 및 55458인 두 사용자가 강제로 데이터베이스에서 연결이 끊어지도록 합니다.

```
db2 force application ( 41408, 55458 )
```

### 사용 시 참고사항

데이터베이스 관리 프로그램은 활동 상태로 남아 db2start를 수행할 필요가 없이 후속 데이터베이스 관리 프로그램 조작을 처리할 수 있도록 합니다.

데이터베이스 무결성을 보존하기 위해 유휴 중이거나 인터럽트할 수 있는 데이터베이스 조작을 실행하는 사용자만 종료할 수 있습니다.

다음과 같은 유형의 사용자 및 응용프로그램은 강제할 수 없습니다.

- 데이터베이스를 작성 중인 사용자
- 시스템 응용프로그램

이러한 유형의 사용자 및 응용프로그램을 성공적으로 강제하려면, 데이터베이스를 비활성화하거나 인스턴스를 재시작해야 합니다.

FORCE APPLICATION을 발행한 후 데이터베이스는 연결 요청을 계속 승인합니다. 모든 사용자를 완전히 강제로 해제하려면 추가 강제가 필요할 수 있습니다.

---

## 제 45 장 GET ADMIN CONFIGURATION

시스템의 관리 노드에 있는 개별 DB2 Administration Server(DAS) 구성 매개변수의 값을 리턴합니다. DAS는 DB2 서버의 리모트 관리를 가능하게 하는 특수 관리 도구입니다. DAS 구성 매개변수의 목록에 대해서는 UPDATE ADMIN CONFIGURATION 명령의 설명을 참조하십시오.

### 범위

이 명령은 사용자가 접속되어 있거나 FOR NODE 옵션에서 지정하는 시스템의 관리 노드에 있는 DAS 구성 매개변수에 관한 정보를 리턴합니다.

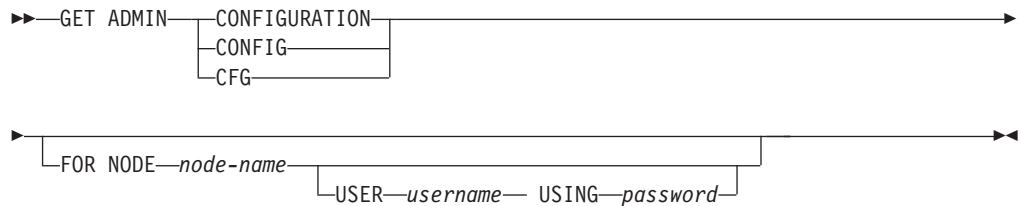
### 권한 부여

없음

### 필수 연결

노드 리모트 시스템에 대한 DAS 구성을 표시하려면 먼저 해당 시스템에 연결하거나 FOR NODE 옵션을 사용하여 시스템의 관리 노드를 지정하십시오.

### 명령 구문



### 명령 매개변수

#### FOR NODE *node-name*

DAS 구성 매개변수를 볼 관리 노드의 이름을 입력하십시오.

#### USER *username* USING *password*

노드에 연결할 때 사용자 이름 및 암호가 필요한 경우 이 정보를 입력하십시오.

### 예:

다음은 GET ADMIN CONFIGURATION의 샘플 출력입니다.

Administration Server 구성

인증 유형 DAS(AUTHENTICATION)

= SERVER\_ENCRYPT

DAS 관리 권한 그룹 이름(DASADM_GROUP)	= ADMINISTRATORS
DAS 발견 모드(DISCOVER)	= SEARCH
DB2 서버 시스템 이름(DB2SYSTEM)	= swalkty
Java Development Kit 설치 경로 DAS(JDK_PATH)	= e:#sql1lib#java#jdk
DAS 코드 페이지(DAS_CODEPAGE)	= 0
DAS 지역(DAS_TERRITORY)	= 0
문의처 목록 위치(CONTACT_HOST)	= hostA.ibm.ca
실행 만기 태스크(EXEC_EXP_TASK)	= NO
스케줄러 모드(SCHED_ENABLE)	= ON
SMTP 서버(SMTP_SERVER)	= smtp1.ibm.ca
데이터베이스 카탈로그 도구(TOOLSCAT_DB)	= CCMD
데이터베이스 카탈로그 인스턴스 도구(TOOLSCAT_INST)	= DB2
데이터베이스 카탈로그 스키마 도구(TOOLSCAT_SCHEMA)	= TOOLSCAT
스케줄러 사용자 ID	= db2admin

## 사용 시 참고사항

오류가 발생하는 경우 리턴된 정보는 유효하지 않습니다. 구성 파일이 유효하지 않은 경우 오류 메시지가 리턴됩니다. 복구하려면 사용자가 DAS를 다시 설치해야 합니다.

구성 매개변수를 DAS와 함께 제공되는 디폴트값으로 설정하려면 RESET ADMIN CONFIGURATION 명령을 사용하십시오.

## 제 46 장 GET ALERT CONFIGURATION

특정 인스턴스에 대한 Health 표시기의 경보 구성 설정을 리턴합니다.

**중요사항:** Health Monitor가 버전 9.7에서 사용되지 않으므로 이 명령 또는 API는 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 버전 9.7의 새로운 내용 책에 있는 『Health Monitor는 사용되지 않음』 주제를 참조하십시오.

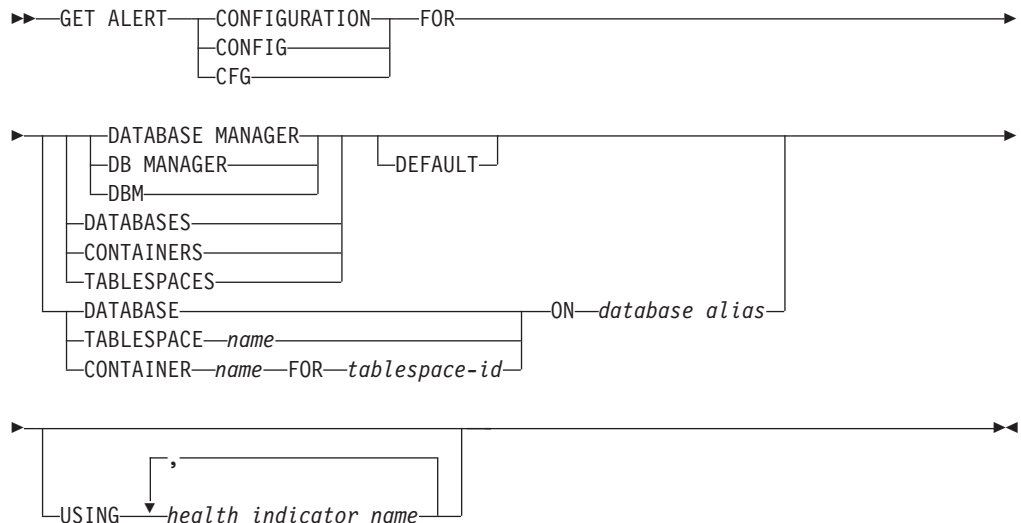
### 권한 부여

없음

### 필수 연결

인스턴스. 명시적 접속은 필요하지 않습니다.

### 명령 구문



### 명령 매개변수

#### DATABASE MANAGER

데이터베이스 관리 프로그램의 경보 설정을 검색합니다.

#### DATABASES

데이터베이스 관리 프로그램이 관리하는 모든 데이터베이스의 경보 설정을 검색합니다. 이들 설정값은 사용자 정의 설정값이 없는 모든 데이터베이스에 적용됩니다. 사용자 정의 설정은 DATABASE ON *database alias* 절을 사용하여 정의됩니다.

## CONTAINERS

데이터베이스 관리 프로그램이 관리하는 모든 테이블 스페이스 컨테이너의 정보 설정을 검색합니다. 이들 설정값은 사용자 정의 설정값이 없는 모든 테이블 스페이스 컨테이너에 적용됩니다. 사용자 정의 설정은 `CONTAINER name ON database alias` 절을 사용하여 정의됩니다.

## TABLESPACES

데이터베이스 관리 프로그램이 관리하는 모든 테이블 스페이스의 정보 설정을 검색합니다. 이들 설정값은 사용자 정의 설정값이 없는 모든 테이블 스페이스에 적용됩니다. 사용자 정의 설정은 `TABLESPACE name ON database alias` 절을 사용하여 정의됩니다.

## DEFAULT

설치 디폴트값이 검색되도록 지정합니다.

## DATABASE ON *database alias*

`ON database alias` 절을 사용하여 지정된 데이터베이스에 대한 정보 설정을 검색합니다. 이 데이터베이스에 사용자 정의 설정이 없는 경우 인스턴스의 모든 데이터베이스에 대한 설정이 리턴됩니다. 이는 `DATABASES` 매개변수를 사용하는 것과 같습니다.

## CONTAINER *name* FOR *tablespace-id* ON *database alias*

`ON database alias` 절을 사용하여 지정된 데이터베이스에서 `FOR tablespace-id` 절을 사용하여 지정된 테이블 스페이스의 *name* 테이블 스페이스 컨테이너에 대한 정보 설정을 검색합니다. 이 테이블 스페이스 컨테이너에 사용자 정의 설정이 없는 경우 데이터베이스의 모든 테이블 스페이스 컨테이너에 대한 설정이 리턴됩니다. 이는 `CONTAINERS` 매개변수를 사용하는 것과 같습니다.

## TABLESPACE *name* ON *database alias*

`ON database alias` 절을 사용하여 지정된 데이터베이스에서 *name* 테이블 스페이스에 대한 정보 설정을 검색합니다. 이 테이블 스페이스에 사용자 정의 설정이 없는 경우 데이터베이스의 모든 테이블 스페이스에 대한 설정이 리턴됩니다. 이는 `TABLESPACES` 매개변수를 사용하는 것과 같습니다.

## USING *health*

정보 구성 정보가 리턴될 Health 표시기 세트를 지정합니다. Health 표시기 이름은 2자 오브젝트 ID 다음에 표시기가 측정하는 내용을 설명하는 이름으로 구성됩니다(예: `db.sort_privmem_util`). 이 절은 선택적 절이므로 사용하지 않는 경우 지정된 오브젝트 또는 오브젝트 유형의 모든 Health 표시기가 리턴됩니다.

## 예:

다음은 데이터베이스 관리 프로그램 정보의 요청 결과에 따른 일반 출력입니다.

DB2 GET ALERT CFG FOR DBM

```

경보 구성
표시기 이름           = db2.db2_op_status
  디폴트               = 예
  유형                 = State-based
  민감도               = 0
  공식                 = db2.db2_status;
  조치                 = 사용 불가능
  임계값 또는 상태 점검 = 사용 가능

표시기 이름           = db2.sort_privmem_util
  디폴트               = 예
  유형                 = 임계값 기반
  경고                 = 90
  경고                 = 100
  단위                 = %
  민감도               = 0
  공식                 = ((db2.sort_heap_allocated/sheapthres)
                        *100);
  조치                 = 사용 불가능
  임계값 또는 상태 점검 = 사용 가능

표시기 이름           = db2.mon_heap_util
  디폴트               = 예
  유형                 = 임계값 기반
  경고                 = 85
  경고                 = 95
  단위                 = %
  민감도               = 0
  공식                 = ((db2.mon_heap_cur_size/
                        db2.mon_heap_max_size)*100);
  조치                 = 사용 불가능
  임계값 또는 상태 점검 = 사용 가능

```

다음은 구성 정보의 요청 결과에 따른 일반 출력입니다.

DB2 GET ALERT CFG FOR DATABASES

```

경보 구성
표시기 이름           = db.db_op_status
  디폴트               = 예
  유형                 = 상태 기반
  민감도               = 0
  공식                 = db.db_status;
  조치                 = 사용 불가능
  임계값 또는 상태 점검 = 사용 가능

표시기 이름           = db.sort_shrmem_util
  디폴트               = 예
  유형                 = 임계값 기반
  경고                 = 70
  경고                 = 85
  단위                 = %
  민감도               = 0
  공식                 = ((db.sort_shrheap_allocated/sheapthres_shr)
                        *100);
  조치                 = 사용 불가능
  임계값 또는 상태 점검 = 사용 가능

표시기 이름           = db.spilled_sorts
  디폴트               = 예
  유형                 = 임계값 기반
  경고                 = 30
  경고                 = 50
  단위                 = %
  민감도               = 0
  공식                 = ((delta(db.sort_overflows,10))/

```

조치	(delta(db.total_sorts,10)+1)*100);
임계값 또는 상태 점검	= 사용 불가능 = 사용 가능
표시기 이름	= db.max_sort_shrmem_util
디폴트	= 예
유형	= 임계값 기반
경고	= 60
알람	= 30
단위	= %
민감도	= 0
공식	= ((db.max_shr_sort_mem/ sheapthres_shr)*100);
조치	= 사용 불가능
임계값 또는 상태 점검	= 사용 가능
표시기 이름	= db.log_util
디폴트	= 예
유형	= 임계값 기반
경고	= 75
알람	= 85
단위	= %
민감도	= 0
공식	= (db.total_log_used/ (db.total_log_used+db.total_log_available) ) *100;
조치	= 사용 불가능
임계값 또는 상태 점검	= 사용 가능
표시기 이름	= db.log_fs_util
디폴트	= 예
유형	= 임계값 기반
경고	= 75
알람	= 85
단위	= %
민감도	= 0
공식	= ((os.fs_used/os.fs_total)*100);
조치	= 사용 불가능
임계값 또는 상태 점검	= 사용 가능
표시기 이름	= db.deadlock_rate
디폴트	= 예
유형	= 임계값 기반
경고	= 5
알람	= 10
단위	= 시간당 교착 상태
민감도	= 0
공식	= delta(db.deadlocks);
조치	= 사용 불가능
임계값 또는 상태 점검	= 사용 가능
표시기 이름	= db.locklist_util
디폴트	= 예
유형	= 임계값 기반
경고	= 75
알람	= 85
단위	= %
민감도	= 0
공식	= (db.lock_list_in_use/(locklist*4096)) *100;
조치	= 사용 불가능
임계값 또는 상태 점검	= 사용 가능
표시기 이름	= db.lock_escal_rate
디폴트	= 예
유형	= 임계값 기반
경고	= 5
알람	= 10
단위	= 시간당 잠금 에스컬레이션



민감도	= 0
공식	= delta(db.lock_escals);
조치	= 사용 불가능
임계값 또는 상태 점검	= 사용 가능
표시기 이름	= db.apps_waiting_locks
디폴트	= 예
유형	= 임계값 기반
경고	= 50
알람	= 70
단위	= %
민감도	= 0
공식	= (db.locks_waiting/db.appls_cur_cons)*100;
조치	= 사용 불가능
임계값 또는 상태 점검	= 사용 가능
표시기 이름	= db.pkgcache_hitratio
디폴트	= 예
유형	= 임계값 기반
경고	= 80
알람	= 70
단위	= %
민감도	= 0
공식	= (1-(db.pkg_cache_inserts/db.pkg_cache_lookups)*100);
조치	= 사용 불가능
임계값 또는 상태 점검	= 사용 불가능
표시기 이름	= db.catcache_hitratio
디폴트	= 예
유형	= 임계값 기반
경고	= 80
알람	= 70
단위	= %
민감도	= 0
공식	= (1-(db.cat_cache_inserts/db.cat_cache_lookups)*100);
조치	= 사용 불가능
임계값 또는 상태 점검	= 사용 불가능
표시기 이름	= db.shrworkspace_hitratio
디폴트	= 예
유형	= 임계값 기반
경고	= 80
알람	= 70
단위	= %
민감도	= 0
공식	= ((1-(db.shr_workspace_section_inserts/db.shr_workspace_section_lookups)*100);
조치	= 사용 불가능
임계값 또는 상태 점검	= 사용 불가능
표시기 이름	= db.db_heap_util
디폴트	= 예
유형	= 임계값 기반
경고	= 85
알람	= 95
단위	= %
민감도	= 0
공식	= ((db.db_heap_cur_size/db.db_heap_max_size)*100);
조치	= 사용 불가능
임계값 또는 상태 점검	= 사용 가능

표시기 이름	= db.tb_reorg_req
디폴트	= 예
유형	= 컬렉션 상태 기반
민감도	= 0
조치	= 사용 불가능
임계값 또는 상태 점검	= 사용 불가능
표시기 이름	= db.hadr_op_status
디폴트	= 예
유형	= State-based
민감도	= 0
공식	= db.hadr_connect_status;
조치	= 사용 불가능
임계값 또는 상태 점검	= 사용 가능
표시기 이름	= db.hadr_delay
디폴트	= 예
유형	= 임계값 기반
경고	= 10
알람	= 15
단위	= 분
민감도	= 0
공식	= (db.hadr_log_gap*var.refresh_rate/60) DIV(delta(db.hadr_secondary_log_pos));
조치	= 사용 불가능
임계값 또는 상태 점검	= 사용 가능
표시기 이름	= db.db_backup_req
디폴트	= 예
유형	= State-based
민감도	= 0
조치	= 사용 불가능
임계값 또는 상태 점검	= 사용 불가능
표시기 이름	= db.fed_nicknames_op_status
디폴트	= 예
유형	= 컬렉션 상태 기반
민감도	= 0
조치	= 사용 불가능
임계값 또는 상태 점검	= 사용 불가능
표시기 이름	= db.fed_servers_op_status
디폴트	= 예
유형	= 컬렉션 상태 기반
민감도	= 0
조치	= 사용 불가능
임계값 또는 상태 점검	= 사용 불가능
표시기 이름	= db.tb_runstats_req
디폴트	= 예
유형	= 컬렉션 상태 기반
민감도	= 0
조치	= 사용 불가능
임계값 또는 상태 점검	= 사용 불가능

---

## 제 47 장 GET CLI CONFIGURATION

db2cli.ini 파일의 콘텐츠를 나열합니다. 이 명령은 전체 파일 또는 지정된 섹션을 나열할 수 있습니다.

db2cli.ini 파일은 DB2 콜 레벨 인터페이스(CLI) 구성 파일로 사용됩니다. DB2 CLI 및 이를 사용하는 응용프로그램의 동작을 수정하는 데 사용할 수 있는 다양한 키워드 및 값이 들어 있습니다. 이 파일은 섹션으로 나뉘어지며, 각 섹션은 데이터베이스 별명 이름에 대응합니다.

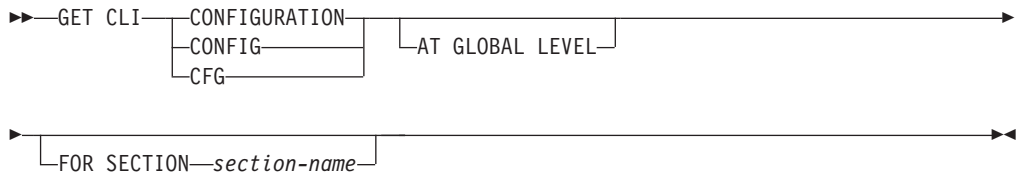
### 권한 부여

없음

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

#### AT GLOBAL LEVEL

LDAP 디렉토리의 디폴트 CLI 구성 매개변수를 표시합니다. 이 매개변수는 Windows 운영 체제에서만 유효합니다.

#### FOR SECTION *section-name*

키워드가 나열될 섹션의 이름입니다. 지정되지 않으면 모든 섹션이 나열됩니다.

### 예:

다음 샘플 출력은 두 섹션을 갖는 db2cli.ini 파일의 콘텐츠를 표시합니다.

```
[tstcli1x]
uid=userid
pwd=password
autocommit=0
TableType="'TABLE','VIEW','SYSTEM TABLE'"
```

```
[tstcli2x]
SchemaList="'OWNER1','OWNER2',CURRENT SQLID"
```

## 사용 시 참고사항

이 명령에 지정된 섹션 이름은 대소문자를 구분하지 않습니다. 예를 들어 db2cli.ini 파일의 섹션 이름(대괄호로 구분됨)이 소문자이고 명령에 지정된 섹션 이름이 대문자인 경우 올바른 섹션이 나열됩니다.

PWD(암호) 키워드의 값은 나열되지 않습니다. 대신 5개의 별표(\*\*\*\*\*)가 나열됩니다.

LDAP(Lightweight Directory Access Protocol)가 사용 가능할 때 CLI 구성 매개변수는 시스템 레벨뿐 아니라 사용자 레벨에서 설정할 수 있습니다. 사용자 레벨에서의 CLI 구성은 LDAP 디렉토리에서 유지됩니다. 지정된 섹션이 사용자 레벨에서 존재하는 경우, 사용자 레벨에서 해당 섹션에 대한 CLI 구성이 리턴됩니다. 그렇지 않으면 시스템 레벨의 CLI 구성이 리턴됩니다.

사용자 레벨에서의 CLI 구성은 LDAP 디렉토리에 유지되며 로컬 시스템에 캐시됩니다. 사용자 레벨에서 CLI 구성을 읽을 때 DB2는 항상 캐시에서 읽습니다. 캐시는 다음 상황에서 새로 고쳐집니다.

- 사용자가 CLI 구성을 갱신합니다.
- 사용자가 명시적으로 REFRESH LDAP 명령을 사용하여 CLI 구성을 강제로 새로 고칩니다.

LDAP 환경에서 사용자는 LDAP 디렉토리에 카탈로그된 데이터베이스에 대한 디폴트 CLI 설정 세트를 구성할 수 있습니다. LDAP 카탈로그된 데이터베이스가 구성 지원 프로그램(CA) 또는 CLI/ODBC 구성 유틸리티를 사용하여 데이터 소스 이름(DSN)으로 추가될 때 모든 디폴트 CLI 설정은 LDAP 디렉토리에 존재하는 경우 로컬 시스템에서 해당 DSN에 대해 구성됩니다. 디폴트 CLI 설정을 표시하려면 AT GLOBAL LEVEL 절을 지정해야 합니다.

---

## 제 48 장 GET CONNECTION STATE

연결 상태를 표시합니다. 가능한 상태는 다음과 같습니다.

- 연결 가능하고 연결되어 있음
- 연결 가능하지만 연결되어 있지 않음
- 연결 불가능하지만 연결되어 있음
- 내재적 연결 가능(내재적 연결이 사용 가능한 경우)

이 명령은 또한 다음 정보를 리턴합니다.

- 데이터베이스 연결 모드(SHARE 또는 EXCLUSIVE)
- 연결이 존재하는 데이터베이스의 이름과 별명(존재하는 경우)
- 연결이 TCP/IP를 사용 중인 경우 연결의 호스트 이름 및 서비스 이름

### 권한 부여

없음

### 필수 연결

없음

### 명령 구문

▶▶—GET CONNECTION STATE—▶▶

### 명령 매개변수

없음

### 예:

다음은 GET CONNECTION STATE의 샘플 출력입니다.

데이터베이스 연결 상태

연결 상태	= 연결 가능하고 연결되어 있음
연결 모드	= SHARE
로컬 데이터베이스 별명	= SAMPLE
데이터베이스 이름	= SAMPLE
호스트 이름	= montero
서비스 이름	= 29384

## 사용 시 참고사항

이 명령은 유형 2 연결에 적용되지 않습니다.

---

## 제 49 장 GET CONTACTGROUP

로컬 시스템에 정의된 단일 문의처 목록에 포함된 연락처를 리턴합니다. 문의처는 스케줄러 및 Health Monitor가 메시지를 보내는 사용자입니다. ADD CONTACTGROUP 명령을 사용하면 이름 지정된 문의처 그룹을 작성할 수 있습니다.

### 권한 부여

없음

### 필수 연결

없음. 로컬 실행 전용: 이 명령은 리모트 연결에 사용할 수 없습니다.

### 명령 구문

► GET CONTACTGROUP *name* ◀

### 명령 매개변수

**CONTACTGROUP** *name*

문의처를 검색할 그룹의 이름.

### 예:

GET CONTACTGROUP 지원

설명

-----

Foo Widgets broadloom support unit

Name	Type
-----	-----
joe	contact
support	contact group
joline	contact





---

## 제 50 장 GET CONTACTGROUPS

이 명령은 시스템에 로컬로 정의되거나 글로벌 목록에 정의될 수 있는 문의처 그룹 목록을 제공합니다. 문의처 그룹은 스케줄러 및 Health Monitor와 같은 모니터링 프로세스가 메시지를 보낼 수 있는 주소 목록입니다. DAS(Database Administration Server) **contact\_host** 구성 매개변수의 설정은 목록이 로컬인지 또는 전역인지를 판별합니다. ADD CONTACTGROUP 명령을 사용하면 이름 지정된 문의처 그룹을 작성할 수 있습니다.

### 권한 부여

없음

### 필요한 연결

없음

### 명령 구문

▶▶ GET CONTACTGROUPS ◀◀

### 명령 매개변수

없음

### 예:

다음 예에서는 GET CONTACTGROUPS 명령이 실행되며 그 결과는 다음과 같습니다.

이름	설명
support	Foo Widgets broadloom support unit
service	Foo Widgets service and support unit



---

## 제 51 장 GET CONTACTS

로컬 시스템에 정의된 문의처 목록을 리턴합니다. 문의처는 스케줄러 및 Health Monitor와 같은 모니터링 프로세스가 통지 또는 메시지를 보내는 사용자입니다.

문의처를 작성하려면 ADD CONTACT 명령을 사용하십시오.

### 권한 부여

없음

### 필수 연결

없음

### 명령 구문

▶▶ GET CONTACTS ◀◀

### 명령 매개변수

없음

### 예:

#### GET CONTACTS

이름	유형	주소	최대 페이지 길이	설명
joe	e-mail	joe@somewhere.com	-	-
joline	e-mail	joline@somewhereelse.com	-	-
john	page	john@relay.org	50	Support 24x7



---

## 제 52 장 GET DATABASE CONFIGURATION

특정 데이터베이스 구성 파일에 있는 개별 항목의 값을 리턴합니다.

### 범위

이 명령은 명령이 실행되는 데이터베이스 파티션의 정보만 리턴합니다.

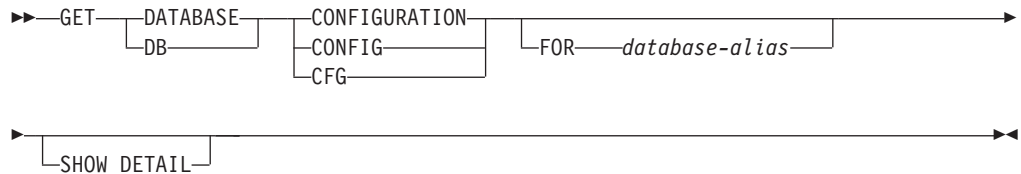
### 권한 부여

없음

### 필수 연결

인스턴스. 명시적 접속이 필요하지 않지만, SHOW DETAIL 절 사용 시 데이터베이스에 대한 연결이 필요합니다. 데이터베이스가 리모트로 나열된 경우, 이 명령의 지속 기간 동안 리모트 노드에 대한 인스턴스 접속이 설정됩니다.

### 명령 구문



### 명령 매개변수

#### FOR *database-alias*

구성을 표시할 데이터베이스의 별명을 지정합니다. 데이터베이스에 대한 연결이 이미 존재하는 경우 별명을 지정할 필요가 없습니다.

#### SHOW DETAIL

다음 번 데이터베이스를 활성화할 때 매개변수의 값과 데이터베이스 구성 매개변수의 현재 값을 나타내는 상세 정보를 표시합니다. 이 옵션을 사용하여 구성 매개변수에 대한 동적 변경 결과를 볼 수 있습니다.

이것은 CLPPlus 인터페이스에서 조작 시 디폴트 절입니다. CLPPlus 프로세서 사용 시 SHOW DETAIL을 호출할 필요가 없습니다.

예:

주:

1. 다른 플랫폼의 출력은 플랫폼별 매개변수를 반영하는 작은 변형을 표시할 수 있습니다.
2. 괄호로 묶은 키워드가 있는 매개변수는 UPDATE DATABASE CONFIGURATION 명령으로 변경될 수 있습니다.
3. 키워드를 포함하지 않는 필드는 데이터베이스 관리 프로그램에 의해 유지되며 갱신될 수 없습니다.

다음은 GET DATABASE CONFIGURATION(Windows에서 발행됨)의 샘플 출력입니다.

데이터베이스에 대한 데이터베이스 구성

```

데이터베이스 구성 릴리스 레벨                = 0x0c00
데이터베이스 릴리스 레벨                    = 0x0c00

데이터베이스 지역                          = US
데이터베이스 코드 페이지                  = 1252
데이터베이스 코드 세트                    = IBM-1252
데이터베이스 국가/지역 코드              = 1
데이터베이스 조합 시퀀스                  = UNIQUE
대체 조합 시퀀스                          (ALT_COLLATE) =
데이터베이스 페이지 크기                  = 8192

동적 SQL 쿼리 관리                        (DYN_QUERY_MGMT) = DISABLE

이 데이터베이스에 대한 발견 지원          (DISCOVER_DB) = ENABLE

액세스 제한                              = NO
디폴트 쿼리 최적화 클래스                  (DFT_QUERYOPT) = 5
병렬 처리 등급                            (DFT_DEGREE) = 1
산술 예외시 계속                         (DFT_SQLMATHWARN) = NO
디폴트 새로 고침 유효 기간                (DFT_REFRESH_AGE) = 0
opt용 디폴트 유지보수 테이블 유형        (DFT_MTTB_TYPES) = SYSTEM
유지되어 자주 사용되는 값의 수          (NUM_FREQVALUES) = 10
유지된 Quantile 수                        (NUM_QUANTILES) = 20

10진수 부동 소수점 반올림 모드          (DECFLT_ROUNDING) = ROUND_HALF_EVEN

백업 보류                                = NO

커미트된 트랜잭션이 모두 디스크에 기록됨 = YES|NO
롤 포워드 보류                            = NO
리스트어 보류                             = NO

다중 페이지 파일 할당 작동              = YES

복구를 위한 로그 유지 상태                = NO
로깅에 대한 User Exit 상태              = NO

자체 성능 조정 메모리                     (SELF_TUNING_MEM) = OFF
데이터베이스 공유 메모리 크기(4KB)      (DATABASE_MEMORY) = AUTOMATIC
데이터베이스 메모리 임계값                (DB_MEM_THRESH) = 10
잠금 목록용 최대 스토리지(4KB)           (LOCKLIST) = 50
Percent. of lock lists per application   (MAXLOCKS) = 22
패키지 캐시 크기(4KB)                    (PCKCACHESZ) = (MAXAPPLS*8)
공유 정렬의 정렬 힙 임계값(4KB)         (SHEAPTHRES_SHR) = 5000

```

정렬 목록 힙(4KB)	(SORTHEAP) = 256
데이터베이스 힙(4KB)	(DBHEAP) = AUTOMATIC
카탈로그 캐시 크기(4KB)	(CATALOGCACHE_SZ) = (MAXAPPLS*4)
로그 버퍼 크기(4KB)	(LOGBUFSZ) = 8
유틸리티 힙 크기(4KB)	(UTIL_HEAP_SZ) = 5000
버퍼 풀 크기(페이지)	(BUFFPAGE) = 250
SQL문 힙(4KB)	(STMHEAP) = AUTOMATIC
디폴트 응용프로그램 힙(4KB)	(APPLHEAPSZ) = AUTOMATIC
응용프로그램 메모리 크기(4KB)	(APPL_MEMORY) = AUTOMATIC
통계 힙 크기(4KB)	(STAT_HEAP_SZ) = AUTOMATIC
교착 상태 점검 간격(밀리초)	(DLCHKTIME) = 10000
잠금 시간종료(초)	(LOCKTIMEOUT) = -1
변경된 페이지 임계값	(CHNGPGS_THRESH) = 60
비동기 페이지 클리너 수	(NUM_IOCLEANERS) = AUTOMATIC
I/O 서버 수	(NUM_IOSERVERS) = AUTOMATIC
인덱스 정렬 플래그	(INDEXSORT) = YES
시퀀스 검출 플래그	(SEQDETECT) = YES
디폴트 프리페치 크기(페이지)	(DFT_PREFETCH_SZ) = AUTOMATIC
트랙 수정 페이지	(TRACKMOD) = OFF
디폴트 컨테이너 수	= 1
디폴트 테이블 스페이스 Extent 크기(페이지)	(DFT_EXTENT_SZ) = 32
실행 중인 최대 프로그램 수	(MAXAPPLS) = AUTOMATIC
실행 중인 응용프로그램의 평균 수	(AVG_APPLS) = AUTOMATIC
응용프로그램당 열린 최대 DB 파일 수	(MAXFILOP) = 32768
Log file size (4KB)	(LOGFILSIZ) = 1000
1차 로그 파일 수	(LOGPRIMARY) = 3
2차 로그 파일 수	(LOGSECOND) = 2
로그 파일에 대한 변경된 경로	(NEWLOGPATH) =
로그 파일에 대한 경로	= D:\#DB2\#NODE0000\#SQL00001\#SQLOGDIR\#
오버플로우 로그 경로	(OVERFLOWLOGPATH) =
미러 로그 경로	(MIRRORLOGPATH) =
처음에 사용되는 로그 파일	=
디스크가 가득 차면 로그 블록화	(BLK_LOG_DSK_FUL) = NO
트랜잭션에서 사용 중인 최대 1차 로그 스페이스 백분율	(MAX_LOG) = 0
1개의 활성 UOW에 대해 사용 중인 로그 파일 수	(NUM_LOG_SPAN) = 0
그룹 커밋 계수	(MINCOMMIT) = 1
소프트 체크포인트 전에 수정된 로그 파일의 백분율	(SOFTMAX) = 100
복구를 위해 로그 유지 작동	(LOGRETAIN) = OFF
로깅에 대한 User Exit 작동	(USEREXIT) = OFF
HADR 데이터베이스 역할	= STANDARD
HADR 로컬 호스트 이름	(HADR_LOCAL_HOST) =
HADR 로컬 서비스 이름	(HADR_LOCAL_SVC) =
HADR 리모트 호스트 이름	(HADR_REMOTE_HOST) =
HADR 리모트 서비스 이름	(HADR_REMOTE_SVC) =
리모트 서버의 HADR 인스턴스 이름	(HADR_REMOTE_INST) =
HADR 시간종료 값	(HADR_TIMEOUT) = 120
HADR 로그 쓰기 동기화 모드	(HADR_SYNCMODE) = NEARSYNC
HADR 피어 창 지속기간(초)	(HADR_PEER_WINDOW) = 0

```

첫 번째 로그 아카이브 메소드      (LOGARCHMETH1) = OFF
logarchmeth1 옵션                  (LOGARCHOPT1) =
두 번째 로그 아카이브 메소드      (LOGARCHMETH2) = OFF
logarchmeth2 옵션                  (LOGARCHOPT2) =
장애 복구 로그 아카이브 경로      (FAILARCHPATH) =
오류 시 로그 아카이브 재시도 수    (NUMARCHRETRY) = 5
로그 아카이브 재시도 대기 시간(초) (ARCHRETRYDELAY) = 20
벤더 옵션                          (VENDOROPT) =

자동 재시작 사용                   (AUTORESTART) = ON
인덱스 재작성 시간 및 인덱스 빌드 다시 실행 (INDEXREC) = SYSTEM (RESTART)
인덱스 빌드 중 로그 페이지 수     (LOGINDEXBUILD) = OFF
디폴트 loadrec 세션 수            (DFT_LOADREC_SES) = 1
유지할 데이터베이스 백업 수       (NUM_DB_BACKUPS) = 12
복구 실행기록 보유(일)            (REC_HIS_RETENTN) = 366
복구 오브젝트 자동 삭제           (AUTO_DEL_REC_OBJ) = OFF

TSM 관리 클래스                    (TSM_MGMTCLASS) =
TSM 노드 이름                     (TSM_NODENAME) =
TSM 소유자                        (TSM_OWNER) =
TSM 암호                          (TSM_PASSWORD) =

자동 유지보수                     (AUTO_MAINT) = ON
  자동 데이터베이스 백업          (AUTO_DB_BACKUP) = OFF
  자동 테이블 유지보수            (AUTO_TBL_MAINT) = ON
  자동 runstats                   (AUTO_RUNSTATS) = ON
  자동 통계 프로파일              (AUTO_STATS_PROF) = OFF
  자동 프로파일 갱신              (AUTO_PROF_UPD) = OFF
  자동 재구성                     (AUTO_REORG) = OFF

자동 유효성 다시 확인              (AUTO_REVAL) = DISABLED
현재 커미트                       (CUR_COMMIT) = ON
입력이 DECIMAL인 CHAR 출력        (DEC_TO_CHAR_FMT) = NEW
XML 문자 조작 사용                 (ENABLE_XMLCHAR) = YES
WLM 콜렉션 간격(분)              (WLM_COLLECT_INT) = 0

```

다음 예는 SHOW DETAIL 옵션 지정 시 명령 출력의 분할 영역을 표시합니다. 대기 값 컬럼의 값은 다음 번 인스턴스를 시작할 때 적용될 값입니다.

데이터베이스에 대한 데이터베이스 구성

설명	매개변수	현재값	지연된 값
데이터베이스 구성 릴리스 레벨		= 0x0c00	
데이터베이스 릴리스 레벨		= 0x0c00	
데이터베이스 지역		= US	
데이터베이스 코드 페이지		= 1252	
데이터베이스 코드 세트		= IBM-1252	
데이터베이스 국가/지역 코드		= 1	
데이터베이스 조합 시퀀스		= UNIQUE	UNIQUE
대체 조합 시퀀스	(ALT_COLLATE) =		
데이터베이스 페이지 크기		= 8192	8192
동적 SQL 쿼리 관리	(DYN_QUERY_MGMT) =	DISABLE	DISABLE
이 데이터베이스에 대한 발견 지원	(DISCOVER_DB) =	ENABLE	ENABLE
액세스 제한		= NO	
디폴트 쿼리 최적화 클래스	(DFT_QUERYOPT) =	5	5
병렬 처리 등급	(DFT_DEGREE) =	1	1
산술 예외시 계속	(DFT_SQLMATHWARN) =	NO	NO
디폴트 새로 고침 유효 기간	(DFT_REFRESH_AGE) =	0	0
opt용 디폴트 유지보수 테이블 유형	(DFT_MTTB_TYPES) =	SYSTEM	SYSTEM



유지되어 자주 사용되는 값의 수	(NUM_FREQVALUES) = 10	10
유지된 Quantile 수	(NUM_QUANTILES) = 20	20
10진수 부동 소수점 반올림 모드	(DECFLT_ROUNDING) = ROUND_HALF_EVEN	ROUND_HALF_EVEN
백업 보류	= NO	
데이터베이스가 일관성이 있음	= NO	
롤 포워드 보류	= NO	
리스토어 보류	= NO	
다중 페이지 파일 할당 작동	= YES	
복구를 위한 로그 유지 상태	= NO	
로그에 대한 User Exit 상태	= NO	
자체 성능 조정 메모리	(SELF_TUNING_MEM) = OFF	OFF
데이터베이스 공유 메모리 크기(4KB)	(DATABASE_MEMORY) = AUTOMATIC(18096)	AUTOMATIC(18096)
데이터베이스 메모리 임계	(DB_MEM_THRESH) = 10	10
잠금 목록용 최대 스토리지(4KB)	(LOCKLIST) = 50	50
Percent. of lock lists per application	(MAXLOCKS) = 22	22
패키지 캐시 크기(4KB)	(PCKCACHESZ) = (MAXAPPLS*8)	(MAXAPPLS*8)
공유 정렬의 정렬 힙 임계값(4KB)	(SHEAPTHRES_SHR) = 5000	5000
정렬 목록 힙(4KB)	(SORTHEAP) = 256	256
데이터베이스 힙(4KB)	(DBHEAP) = AUTOMATIC(600)	AUTOMATIC(600)
카탈로그 캐시 크기(4KB)	(CATALOGCACHE_SZ) = (MAXAPPLS*4)	(MAXAPPLS*4)
로그 버퍼 크기(4KB)	(LOGBUFSZ) = 8	8
유틸리티 힙 크기(4KB)	(UTIL_HEAP_SZ) = 5000	5000
버퍼 풀 크기(페이지)	(BUFFPAGE) = 250	250
SQL문 힙(4KB)	(STMTHEAP) = AUTOMATIC(2048)	AUTOMATIC(2048)
디폴트 응용프로그램 힙(4KB)	(APPLHEAPSZ) = AUTOMATIC(256)	AUTOMATIC(256)
응용프로그램 메모리 크기(4KB)	(APPL_MEMORY) = AUTOMATIC(496)	AUTOMATIC(10000)
통계 힙 크기(4KB)	(STAT_HEAP_SZ) = AUTOMATIC(4384)	AUTOMATIC(4384)
교착 상태 점검 간격(밀리초)	(DLCHKTIME) = 10000	10000
잠금 시간종료(초)	(LOCKTIMEOUT) = -1	-1
변경된 페이지 임계값	(CHNGPGS_THRESH) = 60	60
비동기 페이지 클리너 수	(NUM_IOCLEANERS) = AUTOMATIC(1)	AUTOMATIC(1)
I/O 서버 수	(NUM_IOSERVERS) = AUTOMATIC(3)	AUTOMATIC(3)
인덱스 정렬 플래그	(INDEXSORT) = YES	YES
시퀀스 검출 플래그	(SEQDETECT) = YES	YES
디폴트 프리페치 크기(페이지)	(DFT_PREFETCH_SZ) = AUTOMATIC	AUTOMATIC
트랙 수정 페이지	(TRACKMOD) = NO	NO
디폴트 컨테이너 수	= 1	1
디폴트 테이블 스페이스 Extent 크기(페이지)	(DFT_EXTENT_SZ) = 32	32
실행 중인 최대 프로그램 수	(MAXAPPLS) = AUTOMATIC(40)	AUTOMATIC(40)
실행 중인 응용프로그램의 평균 수	(AVG_APPLS) = AUTOMATIC(1)	AUTOMATIC(1)
응용프로그램당 열린 최대 DB 파일 수	(MAXFILOP) = 32768	32768
로그 파일 크기(4KB)	(LOGFILSIZ) = 1000	1000
1차 로그 파일 수	(LOGPRIMARY) = 3	3
2차 로그 파일 수	(LOGSECOND) = 2	2
로그 파일에 대한 변경된 경로	(NEWLOGPATH) =	
로그 파일에 대한 경로	= D:\WDB2\NODE0000	D:\WDB2\NODE0000
	WSQL00001\WSQLOGDIRW	WSQL00001\WSQLOGDIRW
오버플로우 로그 경로	(OVERFLOWLOGPATH) =	
미러 로그 경로	(MIRRORLOGPATH) =	
처음에 사용되는 로그 파일	=	
디스크가 가득 차면 로그 블록화	(BLK_LOG_DSK_FUL) = NO	NO
트랜잭션에서 사용 중인 최대 1차 로그 스페이스 백분율	(MAX_LOG) = 0	0
1개의 활성 UOW에 대해 사용 중인 로그 파일 수	(NUM_LOG_SPAN) = 0	0
그룹 커밋 계수	(MINCOMMIT) = 1	1
소프트 체크포인트 전에 수정된 로그 파일의 백분율	(SOFTMAX) = 100	100
복구를 위해 로그 유지 작동	(LOGRETAIN) = OFF	OFF
로그에 대한 User Exit 작동	(USEREXIT) = OFF	OFF
HADR 데이터베이스 역할	= STANDARD	STANDARD
HADR 로컬 호스트 이름	(HADR_LOCAL_HOST) =	
HADR 로컬 서비스 이름	(HADR_LOCAL_SVC) =	
HADR 리모트 호스트 이름	(HADR_REMOTE_HOST) =	
HADR 리모트 서비스 이름	(HADR_REMOTE_SVC) =	

리모트 서버의 HADR 인스턴스 이름	(HADR_REMOTE_INST) =	
HADR 시간종료 값	(HADR_TIMEOUT) = 120	120
HADR 로그 쓰기 동기화 모드	(HADR_SYNCMODE) = NEARSYNC	NEARSYNC
HADR 피어 창 지속시간(초)	(HADR_PEER_WINDOW) = 0	0
첫 번째 로그 아카이브 메소드	(LOGARCHMETH1) = OFF	OFF
ogarchmeth1 옵션	(LOGARCHOPT1) =	
두 번째 로그 아카이브 메소드	(LOGARCHMETH2) = OFF	OFF
logarchmeth2 옵션	(LOGARCHOPT2) =	
장애 복구 로그 아카이브 경로	(FAILARCHPATH) =	
오류 시 로그 아카이브 재시도 수	(NUMARCHRETRY) = 5	5
로그 아카이브 재시도 대기 시간(초)	(ARCHRETRYDELAY) = 20	20
벤더 옵션	(VENDOROPT) =	
자동 재시작 사용	(AUTORESTART) = ON	ON
인덱스 재작성 시간 및 인덱스 빌드 다시 실행	(INDEXREC) = SYSTEM	SYSTEM (RESTART)
인덱스 빌드 중 로그 페이지 수	(LOGINDEXBUILD) = OFF	OFF
디폴트 loadrec 세션 수	(DFT_LOADREC_SES) = 1	1
유지할 데이터베이스 백업 수	(NUM_DB_BACKUPS) = 12	12
복구 실행기록 보유(일)	(REC_HIS_RETENTN) = 366	366
복구 오브젝트 자동 삭제	(AUTO_DEL_REC_OBJ) = OFF	OFF
TSM 관리 클래스	(TSM_MGMTCLASS) =	
TSM 노드 이름	(TSM_NODENAME) =	
TSM 소유자	(TSM_OWNER) =	
TSM 암호	(TSM_PASSWORD) =	
자동 유지보수	(AUTO_MAINT) = ON	ON
자동 데이터베이스 백업	(AUTO_DB_BACKUP) = OFF	OFF
자동 테이블 유지보수	(AUTO_TBL_MAINT) = ON	ON
자동 runstats	(AUTO_RUNSTATS) = ON	ON
자동 통계 프로파일	(AUTO_STATS_PROF) = OFF	OFF
자동 프로파일 갱신	(AUTO_PROF_UPD) = OFF	OFF
자동 재구성	(AUTO_REORG) = OFF	OFF
자동 유효성 다시 확인	(AUTO_REVAL) = DISABLED	DISABLED
현재 커밋	(CUR_COMMIT) = ON	ON
입력이 DECIMAL인 CHAR 출력	(DEC_TO_CHAR_FMT) = NEW	NEW
XML 문자 조작 사용	(ENABLE_XMLCHAR) = ON	ON
WLM 콜렉션 간격(분)	(WLM_COLLECT_INT) = 0	0

## 사용 시 참고사항

오류가 발생하면, 리턴된 정보는 유효하지 않습니다. 구성 파일이 유효하지 않으면, 오류 메시지가 리턴됩니다. 백업 버전에서 데이터베이스를 리스토어해야 합니다.

데이터베이스 구성 매개변수를 데이터베이스 관리 프로그램 디폴트로 설정하려면, RESET DATABASE CONFIGURATION 명령을 사용하십시오.

모든 데이터베이스 파티션에서 정보를 검색하려면, SYSIBMADM.DBCFG 관리 뷰를 사용하십시오.

## 제 53 장 GET DATABASE MANAGER CONFIGURATION

데이터베이스 관리 프로그램 구성 파일에 있는 개별 항목의 값을 리턴합니다.

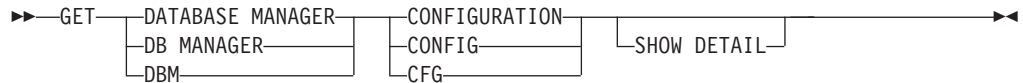
### 권한 부여

없음

### 필수 연결

없음 또는 인스턴스. 로컬 DBM 구성 작업을 수행하는 데는 인스턴스 접속이 필요하지 않지만 리모트 DBM 구성 작업을 수행하는 데는 인스턴스 접속이 필요합니다. 리모트 인스턴스에 대한 데이터베이스 관리 프로그램 구성을 표시하려면 먼저 이 인스턴스에 접속해야 합니다. SHOW DETAIL 절은 인스턴스 접속을 필요로 합니다.

### 명령 구문



### 명령 매개변수

#### SHOW DETAIL

다음 번 데이터베이스 관리 프로그램을 시작할 때 매개변수의 값과 데이터베이스 관리 프로그램 구성 매개변수의 현재 값을 나타내는 상세 정보를 표시합니다. 이 옵션을 사용하여 구성 매개변수에 대한 동적 변경 결과를 볼 수 있습니다.

이것은 CLPPlus 인터페이스에서 조작 시 디폴트 절입니다. CLPPlus 프로세서 사용 시 SHOW DETAIL을 호출할 필요가 없습니다.

### 예:

노드 유형 및 플랫폼 둘 다 어떤 구성 매개변수가 나열되는지 판별합니다.

다음은 GET DATABASE MANAGER CONFIGURATION(Windows에서 발행됨)의 샘플 출력입니다.

```
데이터베이스 관리 프로그램 구성
노드 유형 = 로컬 및 리모트 클라이언트가 있는 Enterprise Server Edition
데이터베이스 관리 프로그램 구성 릴리스 레벨 = 0x0c00
전체 열린 최대 파일 수(MAXTOTFILOP) = 16000
```

```

CPU 속도(밀리초/명령어)(CPUSPEED) = 4.251098e-007
통신 대역폭(MB/초)(COMM_BANDWIDTH) = 1.000000e+002

현재 사용 중인 최대 데이터베이스 수(NUMDB) = 8
페더레이티드 데이터베이스 시스템 지원(FEDERATED) = NO
트랜잭션 프로세서 모니터 이름(TP_MON_NAME) =

디폴트 접미부 어카운트(DFT_ACCOUNT_STR) =

Java Development Kit 설치 경로(JDK_PATH) =

진단 오류 캡처 레벨(DIAGLEVEL) = 3
통지 레벨(NOTIFYLEVEL) = 3
진단 데이터 디렉토리 경로(DIAGPATH) =

디폴트 데이터베이스 모니터 스위치
버퍼 풀(DFT_MON_BUFPOOL) = OFF
잠금(DFT_MON_LOCK) = OFF
정렬(DFT_MON_SORT) = OFF
명령문(DFT_MON_STMT) = OFF
테이블(DFT_MON_TABLE) = OFF
시간소인(DFT_MON_TIMESTAMP) = ON
작업 단위(DFT_MON_UOW) = OFF
인스턴스 및 데이터베이스의 Health 모니터(HEALTH_MON) = ON

SYSADM 그룹 이름(SYSADM_GROUP) =
SYSCTRL 그룹 이름(SYSCTRL_GROUP) =
SYSMAINT 그룹 이름(SYSMAINT_GROUP) =
SYSMON 그룹 이름(SYSMON_GROUP) =

클라이언트 Userid-Password 플러그인(CLNT_PW_PLUGIN) =
클라이언트 Kerberos 플러그인(CLNT_KRB_PLUGIN) = IBMkrb5
그룹 플러그인(GROUP_PLUGIN) =
로컬 권한 부여를 위한 GSS 플러그인(LOCAL_GSSPLUGIN) =
서버 플러그인 모드(SRV_PLUGIN_MODE) = UNFENCED
클라이언트 Userid-Password 플러그인(CLNT_PW_PLUGIN) =
서버 Userid-Password 플러그인(SRVCON_PW_PLUGIN) =
서버 연결 인증(SRVCON_AUTH) = NOT_SPECIFIED
클러스터 관리 프로그램(CLUSTER_MGR) =

데이터베이스 관리 프로그램 인증(AUTHENTICATION) = SERVER
권한 없이 허용되는 카탈로그(CATALOG_NOAUTH) = NO
모든 클라이언트 신뢰(TRUST_ALLCLNTS) = YES
신뢰성 있는 클라이언트 인증(TRUST_CLNTAUTH) = CLIENT
페더레이티드 인증 생략(FED_NOAUTH) = NO

디폴트 데이터베이스 경로(DFTDBPATH) = C:

데이터베이스 모니터 힙 크기(4KB)(MON_HEAP_SZ) = AUTOMATIC
JVM 힙 크기(4KB)(JAVA_HEAP_SZ) = 2048
감사 버퍼 크기(4KB)(AUDIT_BUF_SZ) = 0
인스턴스 공유 메모리 크기(4KB)(INSTANCE_MEMORY) = AUTOMATIC
백업 버퍼 디폴트 크기(4KB)(BACKBUFSZ) = 1024
리스토어 버퍼의 디폴트 크기(4KB)(RESTBUFSZ) = 1024

에이전트 스택 크기(AGENT_STACK_SZ) = 16
최소 커밋 개인용 메모리(4KB)(MIN_PRIV_MEM) = 32
개인용 메모리 임계값(4KB)(PRIV_MEM_THRESH) = 20000

```

```

정렬 힙 임계값(4KB) (SHEAPTHRES) = 0
디렉토리 캐시 지원(DIR_CACHE) = YES

응용프로그램 지원 계층 힙 크기(4KB) (ASLHEAPSZ) = 15
리퀘스터 I/O 블록의 최대 크기(바이트) (RQRIOBLK) = 32767
쿼리 힙 크기(4KB) (QUERY_HEAP_SZ) = 1000

조정 유틸리티의 영향을 받은 워크로드 (UTIL_IMPACT_LIM) = 10

에이전트 우선순위(AGENTPRI) = SYSTEM
에이전트 풀 크기(NUM_POOLAGENTS) = AUTOMATIC
풀에 있는 초기 에이전트 수(NUM_INITAGENTS) = 0
최대 코디네이팅 에이전트 수(MAX_COORDAGENTS) = AUTOMATIC
최대 클라이언트 연결 수(MAX_CONNECTIONS) = AUTOMATIC

분리 프로세스 보존(KEEPFENCED) = YES
풀된 분리 프로세스 수(FENCED_POOL) = AUTOMATIC
초기 분리 프로세스 수(NUM_INITFENCED) = 0

인덱스 재작성 시간 및 인덱스 빌드 다시 실행(INDEXREC) = RESTART

트랜잭션 관리 프로그램 데이터베이스 이름(TM_DATABASE) = 1ST_CONN
트랜잭션 재동기화 간격(초) (RESYNC_INTERVAL) = 180

SPM 이름(SPM_NAME) = KEON14
SPM 로그 크기(SPM_LOG_FILE_SZ) = 256
SPM 재동기화 에이전트 한계(SPM_MAX_RESYNC) = 20
SPM 로그 경로(SPM_LOG_PATH) =

NetBIOS 워크스테이션 이름(NNAME) =

TCP/IP 서비스 이름(SVCENAME) = db2c_DB2
발견 모드(DISCOVER) = SEARCH
발견 서버 인스턴스(DISCOVER_INST) = ENABLE

병렬 처리 등급의 최대 쿼리(MAX_QUERYDEGREE) = ANY
파티션내 병렬 처리 사용(INTRA_PARALLEL) = NO

쿼리 당 최대 비동기 TQ 수(FEDERATED_ASYNC) = 0

int. 통신 버퍼 수(4KB) (FCM_NUM_BUFFERS) = AUTOMATIC
int. 통신 채널 수(FCM_NUM_CHANNELS) = AUTOMATIC
노드 연결 경과 시간(초) (CONN_ELAPSE) = 10
최대 노드 연결 재시도 수(MAX_CONNRETRIES) = 5
노드간 최대 시간차(분) (MAX_TIME_DIFF) = 60

db2start/db2stop 시간종료(분) (START_STOP_TIME) = 10

```

다음 출력 샘플은 SHOW DETAIL 옵션 지정 시 표시되는 정보를 나타냅니다. 대기 값 컬럼에 표시되는 값은 다음 번 데이터베이스 관리 프로그램 인스턴스를 시작할 때 유효한 값입니다.

```
db2 => get dbm cfg show detail
```

데이터베이스 관리 프로그램 구성

노드 유형 = 로컬 및 리모트 클라이언트가 있는 Enterprise Server Edition

설명	매개변수	현재 값	지연된 값
데이터베이스 관리 프로그램 구성 릴리스 레벨		= 0x0c00	
전체 열린 최대 파일 수(MAXTOTALFILP)		= 16000	
CPU 속도(밀리초/명령어)(CPUSPEED)		= 4.251098e-007	4.251098e-007
통신 대역폭(MB/초)(COMM_BANDWIDTH)		= 1.000000e+002	1.000000e+002
현재 사용 중인 최대 데이터베이스 수(NUMDB)		= 8	8
페더레이티드 데이터베이스 시스템 지원(FEDERATED)		= NO	NO
트랜잭션 프로세서 모니터 이름(TP_MON_NAME)		=	
디폴트 접미부 어카운트(DFT_ACCOUNT_STR)		=	
Java Development Kit 설치 경로(JDK_PATH)		=	
진단 오류 캡처 레벨(DIAGLEVEL)		= 3	3
통지 레벨(NOTIFYLEVEL)		= 3	3
진단 데이터 디렉토리 경로(DIAGPATH)		=	
디폴트 데이터베이스 모니터 스위치			
버퍼 풀(DFT_MON_BUFPOOL)		= OFF	OFF
잠금(DFT_MON_LOCK)		= OFF	OFF
정렬(DFT_MON_SORT)		= OFF	OFF
명령문(DFT_MON_STMT)		= OFF	OFF
테이블(DFT_MON_TABLE)		= OFF	OFF
시간소인(DFT_MON_TIMESTAMP)		= ON	ON
작업 단위(DFT_MON_UOW)		= OFF	OFF
인스턴스 및 데이터베이스의 Health 모니터(HEALTH_MON)		= ON	ON
SYSADM 그룹 이름(SYSADM_GROUP)		=	
SYSCTRL 그룹 이름(SYSCTRL_GROUP)		=	
SYSMAINT 그룹 이름(SYSMAINT_GROUP)		=	
SYSMON 그룹 이름(SYSMON_GROUP)		=	
클라이언트 Userid-Password 플러그인(CLNT_PW_PLUGIN)		=	
클라이언트 Kerberos 플러그인(CLNT_KRB_PLUGIN)		= IBMkrb5	IBMkrb5
그룹 플러그인(GROUP_PLUGIN)		=	
로컬 권한 부여를 위한 GSS 플러그인(LOCAL_GSSPLUGIN)		=	
서버 플러그인 모드(SRV_PLUGIN_MODE)		= UNFENCED	UNFENCED
클라이언트 Userid-Password 플러그인(CLNT_PW_PLUGIN)		=	
서버 Userid-Password 플러그인(SRVCON_PW_PLUGIN)		=	
서버 연결 인증(SRVCON_AUTH)		= NOT_SPECIFIED	NOT_SPECIFIED
클러스터 관리 프로그램(CLUSTER_MGR)		=	
데이터베이스 관리 프로그램 인증(AUTHENTICATION)		= SERVER	SERVER
권한 없이 허용되는 카탈로그(CATALOG_NOAUTH)		= NO	NO
모든 클라이언트 신뢰(TRUST_ALLCLNTS)		= YES	YES
신뢰성 있는 클라이언트 인증(TRUST_CLNTAUTH)		= CLIENT	CLIENT
페더레이티드 인증 생략(FED_NOAUTH)		= NO	NO
디폴트 데이터베이스 경로(DFTDBPATH)		= C:	C:
데이터베이스 모니터 힙 크기(4KB)(MON_HEAP_SZ)		= AUTOMATIC(66)	AUTOMATIC(66)
JVM 힙 크기(4KB)(JAVA_HEAP_SZ)		= 2048	2048
감사 버퍼 크기(4KB)(AUDIT_BUF_SZ)		= 0	0
인스턴스 공유 메모리 크기(4KB)(INSTANCE_MEMORY)		= AUTOMATIC(73728)	AUTOMATIC(73728)
백업 버퍼 디폴트 크기(4KB)(BACKBUFSZ)		= 1024	1024
리스터 버퍼의 디폴트 크기(4KB)(RESTBUFSZ)		= 1024	1024
에이전트 스택 크기(AGENT_STACK_SZ)		= 16	16
정렬 힙 임계값(4KB)(SHEAPHTHRES)		= 0	0
디렉토리 캐시 지원(DIR_CACHE)		= YES	YES
응용프로그램 지원 계층 힙 크기(4KB)(ASLHEAPSZ)		= 15	15
리퀘스터 I/O 블록의 최대 크기(바이트)(RQRIOBLK)		= 32767	32767

쿼리 힙 크기(4KB) (QUERY_HEAP_SZ)	= 1000	1000
조정 유틸리티의 영향을 받은 워크로드(UTIL_IMPACT_LIM)	= 10	10
에이전트 우선순위 (AGENTPRI)	= SYSTEM	SYSTEM
에이전트 풀 크기(NUM_POOLAGENTS)	= AUTOMATIC(100)	AUTOMATIC(100)
풀에 있는 초기 에이전트 수(NUM_INITAGENTS)	= 0	0
최대 코디네이팅 에이전트 수(MAX_COORDAGENTS)	= AUTOMATIC(200)	AUTOMATIC(200)
최대 클라이언트 연결 수(MAX_CONNECTIONS)	= AUTOMATIC(MAX_COORDAGENTS)	AUTOMATIC(MAX_COORDAGENTS)
분리 프로세스 보존(KEEPFENCED)	= YES	YES
풀된 분리 프로세스 수(FENCED_POOL)	= AUTOMATIC(MAX_COORDAGENTS)	AUTOMATIC(MAX_COORDAGENTS)
초기 분리 프로세스 수(NUM_INITFENCED)	= 0	0
인덱스 재작성 시간 및 인덱스 빌드 다시 실행(INDEXREC)	= RESTART	RESTART
트랜잭션 관리 프로그램 데이터베이스 이름(TM_DATABASE)	= 1ST_CONN	1ST_CONN
트랜잭션 재동기화 간격(초) (RESYNC_INTERVAL)	= 180	180
SPM 이름 (SPM_NAME)	= KEON14	KEON14
SPM 로그 크기 (SPM_LOG_FILE_SZ)	= 256	256
SPM 재동기화 에이전트 한계 (SPM_MAX_RESYNC)	= 20	20
SPM 로그 경로 (SPM_LOG_PATH) =		
NetBIOS 워크스테이션 이름 (NNAME)	=	
TCP/IP 서비스 이름 (SVCENAME)	= db2c_DB2	db2c_DB2
발견 모드 (DISCOVER)	= SEARCH	SEARCH
발견 서버 인스턴스 (DISCOVER_INST)	= ENABLE	ENABLE
병렬 처리 등급의 최대 쿼리 (MAX_QUERYDEGREE)	= ANY	ANY
파티션내 병렬 처리 사용 (INTRA_PARALLEL)	= NO	NO
쿼리 당 최대 비동기 TQ 수 (FEDERATED_ASYNC)	= 0	0
int. 통신 버퍼 수(4KB) (FCM_NUM_BUFFERS)	= AUTOMATIC(4096)	AUTOMATIC(4096)
int. 통신 채널 수 (FCM_NUM_CHANNELS)	= AUTOMATIC(2048)	AUTOMATIC(2048)
노드 연결 경과 시간(초) (CONN_ELAPSE)	= 10	10
최대 노드 연결 재시도 수 (MAX_CONNRETRIES)	= 5	5
노드간 최대 시간차(분) (MAX_TIME_DIFF)	= 60	60
db2start/db2stop 시간종료(분) (START_STOP_TIME)	= 10	10

## 사용 시 참고사항

- 리모트 인스턴스나 다른 로컬 인스턴스에 대한 접속이 존재하는 경우, 접속된 서버의 데이터베이스 관리 프로그램 구성 매개변수가 리턴되며, 그렇지 않은 경우 로컬 데이터베이스 관리 프로그램 구성 매개변수가 리턴됩니다.
- 오류가 발생하면, 리턴된 정보는 유효하지 않습니다. 구성 파일이 유효하지 않으면, 오류 메시지가 리턴됩니다. 사용자는 복구할 인스턴스를 삭제하고 재작성해야 합니다.
- 구성 매개변수를 데이터베이스 관리 프로그램에서 제공하는 디폴트 값으로 설정하려면, RESET DATABASE MANAGER CONFIGURATION 명령을 사용하십시오.
- FCM\_NUM\_BUFFERS 및 FCM\_NUM\_CHANNELS의 GET DATABASE MANAGER CONFIGURATION SHOW DETAIL에 표시된 AUTOMATIC 값은 인스턴스 시작 시 초기 값이며 런타임 동안 발생하는 자동 증가/감소를 반영하지 않습니다.
- 구성 매개변수 **max\_connections**, **max\_coordagents** 및 **num\_poolagents**는 AUTOMATIC으로 설정됩니다.
- 구성 매개변수 **maxagents** 및 **maxcagents**는 사용되지 않습니다. 다음과 같이 사용되지 않는 기능이 있습니다.

- CLP 및 db2CfgSet API는 해당 매개변수에 대한 갱신을 허용하지만, 이 갱신은 DB2에서 무시됩니다.
- 클라이언트 및 서버가 DB2 v9.5 코드 페이지에 있으면 CLP는 해당 데이터베이스 구성 매개변수를 더 이상 표시하지 않습니다. 서버가 DB2 v9.5이면, 이전 버전 클라이언트에는 해당 매개변수에 대해 0 값이 표시됩니다. 클라이언트가 DB2 v9.5이지만, 서버가 DB2 v9.5 이전이면, 해당 매개변수는 지정된 값으로 표시됩니다.
- db2CfgGet API는 SQLF\_KTN\_MAXAGENTS 및 SQLF\_KTN\_MAXCAGENTS의 요청을 허용하지만, 서버가 DB2 v9.5이면 0을 리턴합니다.
- db2AutoConfig API의 동작은 API에 패스된 db2VersionNumber에 따라 다릅니다. 버전이 DB2 v9.5 이상인 경우, **maxagents**는 리턴되지 않지만, 이 이전 버전에 대해서는 리턴됩니다.
- AUTOCONFIGURE CLP 명령은 이전 버전 클라이언트의 요청이 있는 **maxagents**의 값을 표시합니다(현재 및 권장 값 0). 현재 버전 클라이언트 요청의 경우, **maxagents**는 적절한 값으로 표시됩니다.
- AUTOCONFIGURE ADMIN\_CMD는 서버가 DB2 v9.5 이상일 때 **maxagents**에 대한 정보를 리턴하지 않습니다.
- ADMIN\_CMD를 이용한 **maxagents** 또는 **maxcagents**에 대한 갱신은 정상적으로 리턴하지만 서버가 DB2 v9.5 이상이면 서버에 영향을 미치지 않습니다.
- 서버가 DB2 v9.5 이상이면, DBMCFG 관리 뷰를 사용한 데이터베이스 관리 프로그램 매개변수의 쿼리는 **maxagents** 또는 **maxcagents**의 행을 리턴하지 않습니다.

추후 릴리스에서, 해당 구성 매개변수는 완전히 제거될 수 있습니다.



---

## 제 54 장 GET DATABASE MANAGER MONITOR SWITCHES

데이터베이스 시스템 모니터 스위치 상태를 표시합니다. 모니터 스위치는 데이터베이스 활동 정보를 수집하도록 데이터베이스 시스템 관리 프로그램에 지시합니다. 데이터베이스 시스템 모니터 인터페이스를 사용하는 각 응용프로그램은 자신의 모니터 스위치 세트를 갖습니다. 임의의 모니터링 응용프로그램이 켜지면 데이터베이스 관리 프로그램 레벨 스위치가 설정됩니다. 데이터베이스 시스템 모니터가 현재 모니터링 응용프로그램의 데이터를 수집하고 있는지 여부를 판별하기 위해 이 명령이 사용됩니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*
- *sysmon*

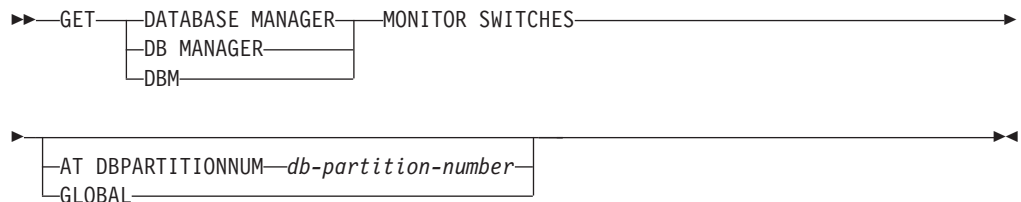
### 필수 연결

인스턴스 또는 데이터베이스:

- 인스턴스에 대한 접속이나 데이터베이스에 대한 연결이 없는 경우, 디폴트 인스턴스 접속이 작성됩니다.
- 인스턴스에 대한 접속과 데이터베이스 연결이 둘 다 있는 경우, 인스턴스 접속이 사용됩니다.

리모트 인스턴스나 다른 로컬 인스턴스에 대한 설정을 표시하려면, 먼저 해당 인스턴스에 대한 접속이 필요합니다.

### 명령 구문



## 명령 매개변수

### AT DBPARTITIONNUM *db-partition-number*

데이터베이스 관리 프로그램 모니터 스위치 상태가 표시되는 데이터베이스 파티션을 지정합니다.

### GLOBAL

파티션된 데이터베이스 환경의 모든 데이터베이스 파티션에 대한 집계 결과를 리턴합니다.

### 예:

다음은 GET DATABASE MANAGER MONITOR SWITCHES에서의 샘플 출력입니다.

```
수집된 DBM 시스템 모니터 정보
Switch list for db partition number 1
Buffer Pool Activity Information (BUFFERPOOL) = ON    06-11-2003 10:11:01.738377
Lock Information (LOCK) = OFF
Sorting Information (SORT) = ON    06-11-2003 10:11:01.738400
SQL Statement Information (STATEMENT) = OFF
Table Activity Information (TABLE) = OFF
Take Timestamp Information (TIMESTAMP) = ON    06-11-2003 10:11:01.738525
Unit of Work Information (UOW) = ON    06-11-2003 10:11:01.738353
```

## 사용 시 참고사항

디폴트로 기록 스위치 BUFFERPOOL, LOCK, SORT, STATEMENT, TABLE 및 UOW가 해제되지만, UPDATE MONITOR SWITCHES 명령을 사용하여 스위치가 설정될 수 있습니다. 이 스위치가 설정된 경우, 이 명령은 스위치가 켜졌을 때의 시간소인을 표시합니다.

기록 스위치 TIMESTAMP가 디폴트로 설정되어 있지만 UPDATE MONITOR SWITCHES를 사용하여 해제될 수 있습니다. 이 스위치가 설정되어 있으면 시스템은 시간소인 모니터 요소의 정보 수집 시 시간소인 호출을 발행합니다. 해당 요소의 예는 다음과 같습니다.

- agent\_sys\_cpu\_time
- agent\_usr\_cpu\_time
- appl\_con\_time
- con\_elapsed\_time
- con\_response\_time
- conn\_complete\_time
- db\_conn\_time
- elapsed\_exec\_time
- gw\_comm\_error\_time
- gw\_con\_time

- gw\_exec\_time
- host\_response\_time
- last\_backup
- last\_reset
- lock\_wait\_start\_time
- network\_time\_bottom
- network\_time\_top
- prev\_uow\_stop\_time
- rf\_timestamp
- ss\_sys\_cpu\_time
- ss\_usr\_cpu\_time
- status\_change\_time
- stmt\_elapsed\_time
- stmt\_start
- stmt\_stop
- stmt\_sys\_cpu\_time
- stmt\_usr\_cpu\_time
- uow\_elapsed\_time
- uow\_start\_time
- uow\_stop\_time

TIMESTAMP 스위치가 해제된 경우, 해당 요소를 판별하기 위해 시간소인 운영 체제 호출이 발행되지 않으며 해당 요소는 0을 포함합니다. 이 스위치를 해제하면 CPU 사용이 100%에 근접하므로 중요해집니다. 이런 경우, 시간소인을 발행하는 데 필요한 CPU 시간이 극적으로 증가합니다.

## 호환성

버전 8 이전 버전과의 호환성:

- NODE 키워드를 DBPARTITIONNUM 대신 사용할 수 있습니다.



---

## 제 55 장 GET DESCRIPTION FOR HEALTH INDICATOR

지정된 Health 표시기에 대한 설명을 리턴합니다. Health 표시기는 데이터베이스 시스템의 특정 상태, 용량 또는 동작의 Health 상태를 측정합니다. 상태는 데이터베이스 오브젝트 또는 자원이 정상적으로 작동하는지 여부를 정의합니다.

### 권한 부여

없음

### 필수 연결

인스턴스. 인스턴스 첨부이 없는 경우 디폴트 인스턴스 첨부이 작성됩니다.

리모트 인스턴스의 스냅샷을 얻으려면 먼저 해당 인스턴스에 접속해야 합니다.

### 명령 구문

►►—GET DESCRIPTION FOR HEALTH INDICATOR—*shortname*—◀◀

### 명령 매개변수

**HEALTH** 표시기 *shortname*

설명을 검색할 Health 표시기의 이름. Health 표시기 이름은 2자 또는 3자 오브젝트 ID 다음에 표시기가 측정하는 내용을 설명하는 이름으로 구성됩니다. 예를 들면, 다음과 같습니다.

db.sort\_privmem\_util

### 예:

다음은 GET DESCRIPTION FOR HEALTH INDICATOR 명령의 샘플 출력입니다.

```
GET DESCRIPTION FOR HEALTH INDICATOR db2.sort_privmem_util
```

```
DESCRIPTION FOR db2.sort_privmem_util
```

Sorting is considered healthy if there is sufficient heap space in which to perform sorting and sorts do not overflow unnecessarily. This indicator tracks the utilization of the private sort memory. If db2.sort\_heap\_allocated (system monitor data element) >= SHEAPTHRES (DBM configuration parameter), sorts may not be getting full sort heap as defined by the SORTHEAP parameter and an alert may be generated. The indicator is calculated using the formula: (db2.sort\_heap\_allocated / SHEAPTHRES) \* 100. The Post Threshold Sorts snapshot monitor element measures the number of sorts that have requested heaps after the sort heap threshold has been exceeded. The value of this indicator, shown in the Additional Details, indicates the degree of severity of the problem for this health indicator. The Maximum Private Sort Memory Used snapshot monitor element maintains a private sort memory high-water mark for the instance. The value of

this indicator, shown in the Additional Information, indicates the maximum amount of private sort memory that has been in use at any one point in time since the instance was last recycled. This value can be used to help determine an appropriate value for SHEAPTHRES.

---

## 제 56 장 GET HEALTH NOTIFICATION CONTACT LIST

인스턴스 상태를 알려주는 문의처와 문의처 그룹의 목록을 리턴합니다. 문의처 목록은 인스턴스 또는 그 데이터베이스 오브젝트에 대해 비정상적인 Health 조건이 나타나는 경우 통지를 받을 개인의 전자 우편 주소 또는 호출기 인터넷 주소로 구성됩니다.

### 권한 부여

없음

### 필요한 연결

인스턴스. 명시적 접속은 필요하지 않습니다.

### 명령 구문

```
▶▶ GET HEALTH NOTIFICATION CONTACT LIST ◀◀  
      └── NOTIFICATION ───┘
```

### 명령 매개변수

없음

### 예:

GET NOTIFICATION LIST 명령을 실행하면 다음과 유사한 보고서가 나타납니다.

이름	유형
-----	-----
Joe Brown	Contact
Support	Contact group





## 제 57 장 GET HEALTH SNAPSHOT

데이터베이스 관리 프로그램과 해당 데이터베이스의 Health 상태 정보를 검색합니다. 리턴된 정보는 명령 실행 시점의 Health 상태 스냅샷을 나타냅니다.

**중요사항:** Health Monitor가 버전 9.7에서 사용되지 않으므로 이 명령 또는 API는 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 *버전 9.7의 새로운 내용* 책에 있는 『Health Monitor는 사용되지 않음』 주제를 참조하십시오.

### 범위

파티션된 데이터베이스 환경의 경우 이 명령은 db2nodes.cfg 파일에 정의된 데이터베이스 파티션에서 호출할 수 있습니다. 디폴트로, 이 명령은 호출된 데이터베이스 파티션에만 적용됩니다. GLOBAL 옵션을 사용하는 경우 모든 데이터베이스 파티션에서 통합 정보를 추출합니다.

### 권한 부여

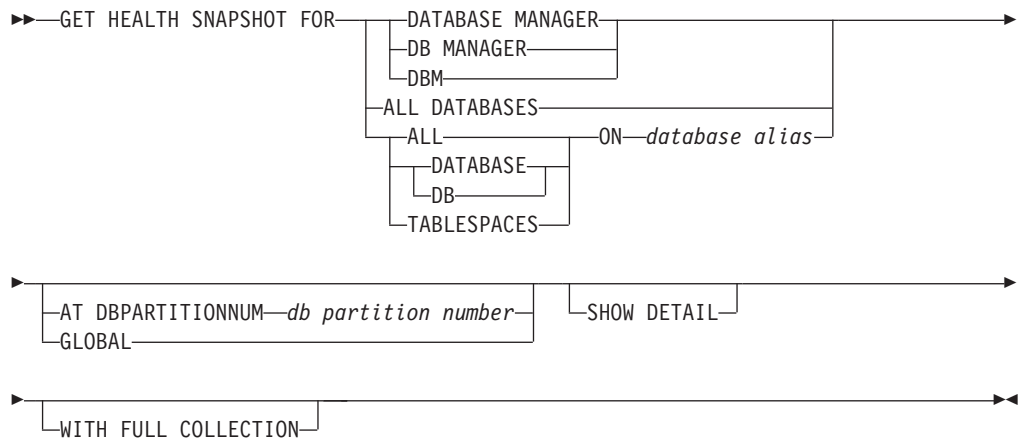
없음

### 필수 연결

인스턴스. 인스턴스 접속이 없는 경우에는 디폴트 인스턴스 접속이 작성됩니다.

리모트 인스턴스의 스냅샷을 얻으려면 먼저 해당 인스턴스에 접속해야 합니다.

### 명령 구문



## 명령 매개변수

### DATABASE MANAGER

활성 데이터베이스 관리 프로그램 인스턴스의 통계를 제공합니다.

### ALL DATABASES

현재 데이터베이스 파티션에 있는 모든 활성 데이터베이스의 Health 상태를 제공합니다.

### ALL ON *database-alias*

지정된 데이터베이스의 모든 테이블 스페이스 및 버퍼 풀에 대한 정보와 Health 상태를 제공합니다.

### DATABASE ON *database-alias*

### TABLESPACES ON *database-alias*

지정된 데이터베이스의 테이블 스페이스에 대한 정보를 제공합니다.

### AT DBPARTITIONNUM *db-partition-number*

지정된 데이터베이스 파티션의 결과를 리턴합니다.

### GLOBAL

파티션된 데이터베이스 환경의 모든 데이터베이스 파티션에 대한 집계 결과를 리턴합니다.

### SHOW DETAIL

출력에 각 Health Monitor 데이터 요소의 실행기록 데이터가 {(Timestamp, Value, Formula)} 양식으로 포함되어야 함을 지정합니다. 여기서 괄호로 묶은 매개변수(Timestamp, Value, Formula)는 리턴되는 각 실행기록 레코드에 반복됩니다. 예를 들면, 다음과 같습니다.

```
(03-19-2002 13:40:24.138865,50,((1-(4/8))*100)),  
(03-19-2002 13:40:13.1386300,50,((1-(4/8))*100)),  
(03-19-2002 13:40:03.1988858,0,((1-(3/3))*100))
```

컬렉션 오브젝트 실행기록은 ATTENTION 또는 AUTOMATE 상태의 모든 컬렉션 오브젝트에 대해 리턴됩니다.

SHOW DETAIL 옵션은 또한 연관된 Health 표시기의 값 및 경고 상태를 이해하는 데 유용한 추가 컨텍스트 정보를 제공합니다. 예를 들어, 테이블 스페이스 스토리지 활용도 Health 표시기를 사용하여 테이블 스페이스가 채워진 정도를 판별하는 경우 테이블 스페이스가 증가하는 속도 또한 SHOW DETAIL 이 제공합니다.

### WITH FULL COLLECTION

모든 컬렉션 상태 기반 Health 표시기의 전체 컬렉션 정보가 리턴되는 것으로 지정합니다. 이 옵션은 이름 및 크기 필터 기준을 모두 고려합니다. 사용자가 전체 컬렉션과 함께 Health 스냅샷을 요청하는 경우 보고서에는 규정의 이름 및 크기 기준을 충족시키는 모든 테이블이 표시됩니다. 지정된 새로 고침 주기

에서 평가될 테이블의 유효성을 확인하는 데 사용할 수 있습니다. 이 옵션이 지정될 때 리턴되는 출력은 NORMAL, AUTOMATED, ATTENTION 또는 AUTOMATE FAILED 상태의 컬렉션 오브젝트에 대한 출력입니다. 이 옵션은 SHOW DETAIL 옵션과 함께 지정할 수 있습니다.

이 옵션을 사용하지 않으면 자동 재구성에 대해 평가되고 수동 개입이 필요한 (즉, 수동 재구성 또는 자동화 실패) 테이블만 Health 스냅샷 가져오기 보고서에 표시됩니다.

## 예:

다음은 데이터베이스 관리 프로그램 정보의 요청 결과에 따른 일반 출력입니다.

```
D:\#>DB2 GET HEALTH SNAPSHOT FOR DBM
```

```

                                     데이터베이스 관리 프로그램 Health 스냅샷
노드 이름                             =
Node type                             = Enterprise Server Edition
                                         with local and remote clients
인스턴스 이름                         = DB2
스냅샷 시간소인                       = 02/17/2004 12:39:44.818949

DB2 인스턴스의 데이터베이스 파티션 수 = 1
데이터베이스 관리 프로그램 시작 시간소인 = 02/17/2004 12:17:21.000119
인스턴스 높은 심각도 경보 상태       = 정상

```

Health 표시기:

```

표시기 이름                           = db2.db2_op_status
값                                     = 0
평가 시간소인                         = 02/17/2004 12:37:23.393000
경보 상태                             = 정상

표시기 이름                           = db2.sort_privmem_util
값                                     = 0
단위                                   = %
평가 시간소인                         = 02/17/2004 12:37:23.393000
경보 상태                             = 정상

표시기 이름                           = db2.mon_heap_util
값                                     = 6
단위                                   = %
평가 시간소인                         = 02/17/2004 12:37:23.393000
경보 상태                             = 정상

```



---

## 제 58 장 GET INSTANCE

DB2INSTANCE 환경 변수의 값을 리턴합니다.

권한 부여

없음

필수 연결

없음

명령 구문

▶▶—GET INSTANCE—◀◀

명령 매개변수

없음

예:

다음은 GET INSTANCE의 샘플 출력입니다.

현재 데이터베이스 관리 프로그램 인스턴스: smith



## 제 59 장 GET MONITOR SWITCHES

현재 세션의 데이터베이스 시스템 모니터 스위치 상태를 표시합니다. 모니터 스위치는 데이터베이스 활동 정보를 수집하도록 데이터베이스 시스템 관리 프로그램에 지시합니다. 데이터베이스 시스템 모니터 인터페이스를 사용하는 각 응용프로그램은 자신의 모니터 스위치 세트를 갖습니다. 이 명령은 이 세트를 표시합니다. 데이터베이스 관리 프로그램 레벨 스위치를 표시하려면, GET DBM MONITOR SWITCHES 명령을 사용하십시오.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*
- *sysmon*

### 필수 연결

인스턴스. 인스턴스가 접속되지 않은 경우, 디폴트 인스턴스 접속이 작성됩니다.

리모트 인스턴스나 다른 로컬 인스턴스에 대한 설정을 표시하려면, 먼저 해당 인스턴스에 대한 접속이 필요합니다.

### 명령 구문

```
▶▶—GET MONITOR SWITCHES—▶▶
└──AT DBPARTITIONNUM—db-partition-number—┘
└──GLOBAL──────────────────────────────────┘
```

### 명령 매개변수

**AT DBPARTITIONNUM** *db-partition-number*

모니터 스위치 상태가 표시되는 데이터베이스 파티션을 지정합니다.

**GLOBAL**

파티션된 데이터베이스 환경의 모든 데이터베이스 파티션에 대한 집계 결과를 리턴합니다.

**예:**

다음은 GET MONITOR SWITCHES에서의 샘플 출력입니다.

## 모니터 기록 스위치

```
Switch list for db partition number 1
Buffer Pool Activity Information (BUFFERPOOL) = ON 02-20-2003 16:04:30.070073
Lock Information (LOCK) = OFF
Sorting Information (SORT) = OFF
SQL Statement Information (STATEMENT) = ON 02-20-2003 16:04:30.070073
Table Activity Information (TABLE) = OFF
Take Timestamp Information (TIMESTAMP) = ON 02-20-2003 16:04:30.070073
Unit of Work Information (UOW) = ON 02-20-2003 16:04:30.070073
```

## 사용 시 참고사항

기록 스위치 `TIMESTAMP`가 디폴트로 설정되어 있지만 `UPDATE MONITOR SWITCHES`를 사용하여 해제될 수 있습니다. 이 스위치가 설정되어 있으면 시스템은 시간소인 모니터 요소의 정보 수집 시 시간소인 호출을 발행합니다.

기록 스위치 `TIMESTAMP`가 디폴트로 설정되어 있지만 `UPDATE MONITOR SWITCHES`를 사용하여 해제될 수 있습니다. 이 스위치가 해제된 경우, 이 명령은 스위치가 꺼졌을 때의 시간소인을 표시합니다. 이 스위치가 설정되어 있으면 시스템은 시간소인 모니터 요소의 정보 수집 시 시간소인 호출을 발행합니다. 해당 요소의 예는 다음과 같습니다.

- `agent_sys_cpu_time`
- `agent_usr_cpu_time`
- `appl_con_time`
- `con_elapsed_time`
- `con_response_time`
- `conn_complete_time`
- `db_conn_time`
- `elapsed_exec_time`
- `gw_comm_error_time`
- `gw_con_time`
- `gw_exec_time`
- `host_response_time`
- `last_backup`
- `last_reset`
- `lock_wait_start_time`
- `network_time_bottom`
- `network_time_top`
- `prev_uow_stop_time`
- `rf_timestamp`



- ss\_sys\_cpu\_time
- ss\_usr\_cpu\_time
- status\_change\_time
- stmt\_elapsed\_time
- stmt\_start
- stmt\_stop
- stmt\_sys\_cpu\_time
- stmt\_usr\_cpu\_time
- uow\_elapsed\_time
- uow\_start\_time
- uow\_stop\_time

TIMESTAMP 스위치가 해제된 경우, 해당 요소를 판별하기 위해 시간소인 운영 체제 호출이 발행되지 않으며 해당 요소는 0을 포함합니다. 이 스위치를 해제하면 CPU 사용이 100%에 근접하므로 중요해집니다. 이런 경우, 시간소인을 발행하는 데 필요한 CPU 시간이 극적으로 증가합니다.

## 호환성

버전 8 이전 버전과의 호환성:

- NODE 키워드를 DBPARTITIONNUM 대신 사용할 수 있습니다.



## 제 60 장 GET RECOMMENDATIONS FOR HEALTH INDICATOR

지정된 Health 표시기가 모니터링하는 데이터베이스 시스템 측면의 Health 상태를 개선하기 위한 권장사항에 대한 설명을 리턴합니다. 특정 오브젝트에 대해 정보 상태인 Health 표시기에 대한 권장사항이 리턴되거나 지정된 Health 표시기의 전체 권장사항 세트를 쿼리할 수 있습니다.

**중요사항:** Health Monitor가 버전 9.7에서 사용되지 않으므로 이 명령 또는 API는 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 버전 9.7의 새로운 내용 책에 있는 『Health Monitor는 사용되지 않음』 주제를 참조하십시오.

### 범위

파티션된 데이터베이스 환경의 경우 이 명령은 db2nodes.cfg 파일에 정의된 데이터베이스 파티션에서 호출할 수 있습니다. 이 명령은 GLOBAL 매개변수가 지정되지 않은 경우 해당 데이터베이스 파티션에만 적용됩니다.

### 권한 부여

없음

### 필수 연결

인스턴스. 인스턴스 첨부이 없는 경우 디폴트 인스턴스 첨부이 작성됩니다. 리모트 인스턴스의 권장사항을 검토하려면 먼저 해당 인스턴스에 접속해야 합니다.

### 명령 구문

```
➤ GET RECOMMENDATIONS FOR HEALTH INDICATOR—health-indicator-name—➤
├── FOR ───┬── DBM ───────────────────────────────────────────────────────────┬──
│         ├── TABLESPACE—tblspacename ───────────────────────────────────┬── ON—database-alias ───┬──
│         ├── CONTAINER—containername—FOR TABLESPACE—tblspacename ───┬──
│         └── DATABASE ─────────────────────────────────────────────────────────┘
└── AT DBPARTITIONNUM—db-partition-number ───────────────────────────────────┘
    └── GLOBAL ───────────────────────────────────────────────────────────────────┘
```

### 명령 매개변수

#### HEALTH 표시기 health-indicator-name

권장사항을 검색할 Health 표시기의 이름. Health 표시기 이름은 2자 또는 3자 오브젝트 ID 다음에 표시기가 측정하는 내용을 설명하는 이름으로 구성됩니다.

**DBM** 경고 상태가 된 데이터베이스 관리 프로그램 Health 표시기의 권장사항을 리턴합니다.

**TABLESPACE** *tblspacename*

지정된 테이블 스페이스 및 데이터베이스에 대해 경고 상태가 된 Health 표시기의 권장사항을 리턴합니다.

**CONTAINER** *containername*

지정된 테이블 스페이스 및 데이터베이스의 지정된 컨테이너에 대해 경고 상태가 된 Health 표시기의 권장사항을 리턴합니다.

**DATABASE**

지정된 데이터베이스에 대해 경고 상태가 된 Health 표시기의 권장사항을 리턴합니다.

**ON** *database-alias*

데이터베이스를 지정합니다.

**AT DBPARTITIONNUM**

Health 표시기가 경고 상태가 된 데이터베이스 파티션 번호를 지정합니다. 데이터베이스 파티션 번호가 지정되지 않고 GLOBAL이 지정되지 않으면 명령이 올바르게 연결된 데이터베이스 파티션에 대한 정보를 리턴합니다.

**GLOBAL**

모든 데이터베이스 파티션에서 지정된 Health 표시기의 권장사항을 검색합니다. 다른 데이터베이스 파티션에서 권장사항이 같은 경우, 해당 권장사항은 영향을 받은 데이터베이스 파티션의 Health 표시기를 해석하는 단일 권장사항 세트로서 리턴됩니다.

**예:**

```
db2 get recommendations for health indicator db.db_heap_util
for database on sample
```

Problem:

Indicator Name	= db.db_heap_util
Value	= 42
Evaluation timestamp	= 11/25/2003 19:04:54
Alert state	= Alarm
Additional information	=

Recommendations:

Recommendation: Increase the database heap size.

Rank: 1

Increase the database configuration parameter dbheap sufficiently to move utilization to normal operating levels. To increase the value, set the new value of dbheap to be equal to  $(\text{pool\_cur\_size} / (4096 * U))$  where U is the desired utilization rate. For example, if your desired utilization rate is 60% of the warning

threshold level, which you have set at 75%, then  
 $U = 0.6 * 0.75 = 0.45$  (or 45%).

Take one of the following actions:

Execute the following scripts at the DB2 server:

```
CONNECT TO SAMPLE;  
UPDATE DB CFG USING DBHEAP 149333;  
CONNECT_RESET;
```

Launch DB2 tool: Database Configuration Window

The Database Configuration window can be used to view and update database configuration parameters.

To open the Database Configuration window:

1. From the Control Center, expand the object tree until you find the databases folder.
2. Click the databases folder. Any existing database are displayed in the contents pane on the right side of the window.
3. Right-click the database that you want in the contents pane, and click Configure Parameters in the pop-up menu. The Database Configuration window opens.

On the Performance tab, update the database heap size parameter as suggested and click OK to apply the update.

Recommendation: Investigate memory usage of database heap.  
Rank: 2

There is one database heap per database and the database manager uses it on behalf of all applications connected to the database. The data area is expanded as needed up to the maximum specified by dbheap.

For more information on the database heap, refer to the DB2 Information Center.

Investigate the amount of memory that was used for the database heap over time to determine the most appropriate value for the database heap configuration parameter. The database system monitor tracks the highest amount of memory that was used for the database heap.

Take one of the following actions:

Launch DB2 tool: Memory Visualizer

The Memory Visualizer is used to monitor memory allocation within a DB2 instance. It can be used to monitor overall memory usage, and to update configuration parameters for individual memory components.

To open the Memory Visualizer:

1. From the Control Center, expand the object tree until you find the instances folder.
2. Click the instances folder. Any existing instances are displayed

- in the contents pane on the right side of the window.
3. Right-click the instance that you want in the contents pane, and click View Memory Usage in the pop-up menu. The Memory Visualizer opens.

To start the Memory Visualizer from the command line issue the db2memvis command.

The Memory Visualizer displays a hierarchical list of memory pools for the database manager. Database Heap is listed under the Database Manager Memory group for each database. On Windows, it is listed under the Database Manager Shared Memory group.

Click the check box on the Show Plot column for the Database Heap row to add the element to the plot.

## 사용 시 참고사항

GET RECOMMENDATIONS FOR HEALTH INDICATOR 명령은 다음 두 가지 방법으로 사용할 수 있습니다.

- 가능한 모든 권장사항의 정보 목록을 가져오기 위해 Health 표시기만 사용합니다. 오브젝트가 지정되지 않으면 이 명령은 지정된 Health 표시기의 정보를 해석하는 데 사용할 수 있는 모든 권장사항의 전체 목록을 리턴합니다.
- 해당 오브젝트에 대한 특정 정보를 해석하는 오브젝트를 지정합니다. 오브젝트(예를 들어, 데이터베이스 또는 테이블 스페이스)가 지정되는 경우 리턴되는 권장사항은 식별된 오브젝트에 대한 정보에 따라 다릅니다. 이러한 경우 권장사항이 보다 구체적이며 정보 해석에 대한 자세한 정보가 포함됩니다. 식별된 Health 표시기가 지정된 오브젝트에 대해 정보 상태가 아닌 경우에는 권장사항이 리턴되지 않습니다.

---

## 제 61 장 GET ROUTINE

지정된 SQL 루틴의 루틴 SQL 아카이브(SAR) 파일을 검색합니다.

### 권한 부여

SYSCAT.ROUTINES에 대한 SELECT 특권.

### 필수 연결

데이터베이스. 내재된 연결이 사용 가능한 경우 디폴트 데이터베이스에 연결됩니다.

### 명령 구문

```
▶▶ GET ROUTINE INTO file_name FROM SPECIFIC PROCEDURE routine_name ▶▶  
  
▶ HIDE BODY ▶▶
```

### 명령 매개변수

#### INTO *file\_name*

루틴 SQL 아카이브(SAR)가 저장되는 파일의 이름을 지정합니다.

#### FROM

검색될 루틴의 스펙 시작을 표시합니다.

#### SPECIFIC

지정된 루틴 이름이 특정 이름으로서 제공됩니다.

#### PROCEDURE

루틴은 SQL 프로시저입니다.

#### *routine\_name*

프로시저의 이름. SPECIFIC이 지정된 경우 프로시저의 특정 이름입니다. 이름이 스키마 이름으로 규정되지 않는 경우 CURRENT SCHEMA는 루틴의 스키마 이름으로 사용됩니다. *routine-name*은 SQL 프로시저로 정의된 기존 프로시저여야 합니다.

#### HIDE BODY

루틴 텍스트가 카탈로그에서 추출될 때 루틴 본문이 비어 있는 본문으로 교체되어야 하는 것으로 지정합니다.

컴파일된 코드에는 영향을 주지 않고 텍스트에만 영향을 줍니다.

예:

```
GET ROUTINE INTO procs/proc1.sar FROM PROCEDURE myapp1.proc1;
```

## 사용 시 참고사항

GET ROUTINE 또는 PUT ROUTINE 조작(또는 해당 프로시저)을 실행할 수 없는 경우 실패 원인에 대한 정보를 제공하는 진단 텍스트와 함께 항상 오류를 리턴합니다 (SQLSTATE 38000). 예를 들어, GET ROUTINE에 제공된 프로시저 이름이 SQL 프로시저를 식별하지 않는 경우, 진단 "-204, 42704" 텍스트가 리턴됩니다. 여기서, "-204" 및 "42704"는 각각 문제점의 원인을 식별하는 SQLCODE 및 SQLSTATE입니다. 이 예의 SQLCODE와 SQLSTATE는 GET ROUTINE 명령에 제공된 프로시저 이름이 정의되지 않음을 나타냅니다.



---

## 제 62 장 GET SNAPSHOT

상태 정보를 수집하고 사용자에게 대한 출력을 형식화합니다. 리턴된 정보는 명령이 발행되었을 때의 데이터베이스 관리 프로그램 작동 상태 스냅샷을 나타냅니다.

### 범위

파티션된 데이터베이스 환경의 경우 이 명령은 db2nodes.cfg 파일에 정의된 데이터베이스 파티션에서 호출할 수 있습니다. 이 명령은 해당 데이터베이스 파티션에만 작동합니다.

### 권한 부여

다음 중 하나가 필요합니다.

- SYSADM
- SYSCTRL
- SYSMOINT
- SYSMON

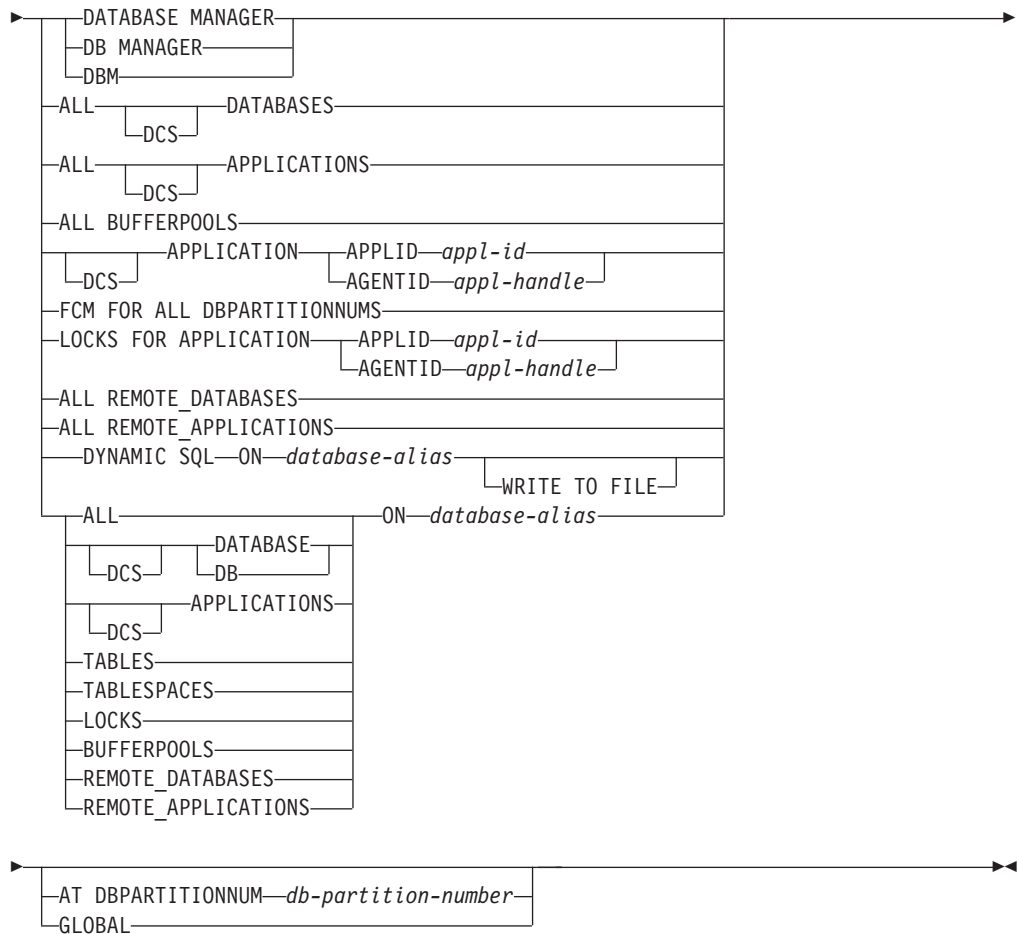
### 필수 연결

인스턴스. 인스턴스 접속이 없는 경우에는 디폴트 인스턴스 접속이 작성됩니다.

리모트 인스턴스의 스냅샷을 얻으려면 먼저 해당 인스턴스에 접속해야 합니다.

### 명령 구문

▶▶ GET SNAPSHOT FOR —————▶▶



일부 통계를 수집하려면 모니터 스위치를 켜야 합니다.

## 명령 매개변수

### DATABASE MANAGER

활성 데이터베이스 관리 프로그램 인스턴스의 통계를 제공합니다.

### ALL DATABAS

현재 데이터베이스 파티션에 있는 모든 활성 데이터베이스의 일반 통계를 제공합니다.

### ALL APPLICATIONS

현재 데이터베이스 파티션의 데이터베이스에 연결된 모든 활성 응용프로그램에 대한 정보를 제공합니다.

### ALL BUFFERPOOLS

모든 활성 데이터베이스의 버퍼 풀 활동에 대한 정보를 제공합니다.

### APPLICATION APPLID *appl-id*

ID가 지정된 응용프로그램에 대한 정보만 제공합니다. 특정의 응용프로그램 ID를 가져오려면 LIST APPLICATIONS 명령을 사용하십시오.

**APPLICATION AGENTID** *appl-handle*

응용프로그램 핸들이 지정된 응용프로그램에 대한 정보만 제공합니다. 응용프로그램 핸들은 현재 실행 중인 응용프로그램을 고유하게 식별하는 32비트 숫자입니다. 특정의 응용프로그램 핸들을 가져오려면 LIST APPLICATIONS 명령을 사용하십시오.

**FCM FOR ALL DBPARTITIONNUMS**

GET SNAPSHOT 명령이 발행된 데이터베이스 파티션과 파티션된 데이터베이스 환경의 다른 데이터베이스 파티션 사이의 FCM(Fast Communication Manager) 통계를 제공합니다.

**LOCKS FOR APPLICATION APPLID** *appl-id*

응용프로그램 ID로 식별되는 지정된 응용프로그램이 보유하는 모든 잠금에 대한 정보를 제공합니다.

**LOCKS FOR APPLICATION AGENTID** *appl-handle*

응용프로그램 핸들로 식별되는 지정된 응용프로그램이 보유하는 모든 잠금에 대한 정보를 제공합니다.

**ALL REMOTE\_DATABASES**

현재 데이터베이스 파티션에 있는 모든 활성 리모트 데이터베이스에 대한 일반 통계를 제공합니다.

**ALL REMOTE\_APPLICATIONS**

현재 데이터베이스 파티션에 연결된 모든 활성 리모트 응용프로그램에 대한 정보를 제공합니다.

**ALL ON** *database-alias*

지정된 데이터베이스의 모든 응용프로그램, 테이블, 테이블 스페이스, 버퍼 풀 및 잠금에 대한 일반 통계와 정보를 제공합니다.

**DATABASE ON** *database-alias*

지정된 데이터베이스에 대한 일반 통계를 제공합니다.

**APPLICATIONS ON** *database-alias*

지정된 데이터베이스에 연결된 모든 응용프로그램에 대한 정보를 제공합니다.

**TABLES ON** *database-alias*

지정된 데이터베이스에 있는 테이블에 대한 정보를 제공합니다. 여기에는 TABLE 기록 스위치가 켜진 후 액세스한 테이블만 포함됩니다.

**TABLESPACES ON** *database-alias*

지정된 데이터베이스의 테이블 스페이스에 대한 정보를 제공합니다.

**LOCKS ON** *database-alias*

지정된 데이터베이스에 연결된 각 응용프로그램이 보유하는 모든 잠금에 대한 정보를 제공합니다.

### **BUFFERPOOLS ON *database-alias***

지정된 데이터베이스의 버퍼 풀 활동에 대한 정보를 제공합니다.

### **REMOTE\_DATABASES ON *database-alias***

지정된 데이터베이스의 모든 활성 리모트 데이터베이스에 대한 일반 통계를 제공합  
니다.

### **REMOTE\_APPLICATIONS ON *database-alias***

지정된 데이터베이스의 리모트 응용프로그램에 대한 정보를 제공합니다.

### **DYNAMIC SQL ON *database-alias***

데이터베이스에 대한 SQL문 캐시 콘텐츠의 특정 시점 그림을 리턴합니다.

### **WRITE TO FILE**

스냅샷 결과가 서버에서 파일에 저장되고 다시 클라이언트로 전달됨을 지정합  
니다. 이 명령은 데이터베이스에 연결되었을 때만 유효합니다. 스냅샷 데이터는  
호출될 때의 연결 동안 테이블 함수 SYSFUN.SQLCACHE\_SNAPSHOT을 통  
해 쿼리할 수 있습니다.

**DCS** DCS가 지정된 절에 따라서, 이 키워드는 다음에 대한 통계를 요청합니다.

- DB2 Connect 게이트웨이에서 현재 실행 중인 특정 DCS 응용프로그램
- 모든 DCS 응용프로그램
- 현재 특정 DCS 데이터베이스에 연결된 모든 DCS 응용프로그램
- 특정 DCS 데이터베이스
- 모든 DCS 데이터베이스

### **AT DBPARTITIONNUM *db-partition-number***

지정된 데이터베이스 파티션의 결과를 리턴합니다.

### **GLOBAL**

파티션된 데이터베이스 환경의 모든 데이터베이스 파티션에 대한 집계 결과를  
리턴합니다.

### **예:**

- 데이터베이스 관리 프로그램에 대한 스냅샷 정보를 요청하려면 다음 명령을 발행하  
십시오.

```
get snapshot for database manager
```

다음은 위의 명령으로 인해 표시되는 샘플 출력입니다.

데이터베이스 관리 프로그램 스냅샷

노드 이름	=	
노드 유형	=	로컬 및 리모트 클라이언트가 있는 Enterprise Server Edition
인스턴스 이름	=	DB2
DB2 인스턴스의 데이터베이스 파티션 수	=	1
데이터베이스 관리 프로그램 상태	=	활성

```

제품 이름 = DB2 v9.5.0.535
서비스 레벨 = s070101 (NT32)

할당된 총 개인용 정렬 힙 수 = 0
개인용 정렬 힙 상위 워터 마크(water mark) = 0
포스트 임계값 정렬 수 = 수집되지 않음
요청된 파이프 정렬 수 = 0
수락된 파이프 정렬 수 = 0

데이터베이스 관리 프로그램 시작 시간소인 = 01/10/2007 15:18:36.241035
최종 재설정 시간소인 =
스냅샷 시간소인 = 01/10/2007 15:28:26.989789

db 관리 프로그램과의 리모트 연결 = 3
db 관리 프로그램에서 실행 중인 리모트 연결 수 = 0
로컬 연결 수 = 1
db 관리 프로그램에서 실행 중인 로컬 연결 수 = 0
활동 중인 로컬 데이터베이스 수 = 1

등록된 에이전트의 상위 워터 마크(water mark) = 0
등록된 에이전트 수 = 8
유휴 에이전트 = 0

커미트된 개인용 메모리(바이트) = 8912896

DB 파티션 번호 0의 전환 목록
버퍼 풀 활동 정보(BUFFERPOOL) = OFF
잠금 정보(LOCK) = ON 01/10/2007 15:22:43.145437
정렬 정보(SORT) = OFF
SQL문 정보(STATEMENT) = OFF
테이블 활동 정보(TABLE) = OFF
시간소인 정보 가져오기(TIMESTAMP) = ON 01/10/2007 15:18:36.241035
작업 단위 정보(UOW) = OFF

폴로부터 할당된 에이전트 수 = 3
비어 있는 폴로부터 작성된 에이전트 = 11
다른 응용프로그램에서 임포트한 에이전트 수 = 0
코디네이팅 에이전트의 상위 워터 마크(water mark) = 9
힙 임계값이 초과된 후의 해시 조인 수 = 0
힙(heap) 임계값이 초과된 후의 OLAP 기능 = 0

전체 게이트웨이 연결 수 = 0
현재 게이트웨이 연결 수 = 0
호스트 응답을 기다리는 게이트웨이 연결 수 = 0
클라이언트 응답을 기다리는 게이트웨이 연결 수 = 0
소실된 게이트웨이 연결 풀 에이전트 수 = 0

해당 노드 FCM 정보 = 0
여유 FCM 버퍼 = 128
여유 FCM 버퍼 하위 워터 마크(water mark) = 128
여유 FCM 채널 = 128
여유 FCM 채널 하위 워터 마크(water mark) = 128

데이터베이스 관리 프로그램의 메모리 사용:

노드 번호 = 0
메모리 풀 유형 = 기타 메모리
현재 크기(바이트) = 11534336
상위 워터 마크(water mark)(바이트) = 11599872
구성된 크기(바이트) = 34275328

노드 번호 = 0
메모리 풀 유형 = 데이터베이스 모니터 힙
현재 크기(바이트) = 65536
상위 워터 마크(water mark)(바이트) = 65536
구성된 크기(바이트) = 327680

```

```

노드 번호 = 0
메모리 풀 유형 = FCMBP 힙
현재 크기(바이트) = 655360
상위 워터 마크(water mark)(바이트) = 655360
구성된 크기(바이트) = 851968

```

- 에이전트 ID가 29인 응용프로그램에 대한 스냅샷 정보를 요청하려면 다음을 발행하십시오.

```
get snapshot for application agentid 29
```

다음은 위의 명령으로 인해 표시되는 샘플 출력입니다. 이 때 잠금 및 명령문 모니터 스위치가 ON으로 설정된 것으로 가정합니다.

#### 응용프로그램 스냅샷

```

응용프로그램 핸들 = 29
응용프로그램 상태 = 잠금 대기
상태 변경 시간 = 수집되지 않음
응용프로그램 코드 페이지 = 819
응용프로그램 국가/지역 코드 = 1
DUOW 상관 토큰 = *LOCAL.jwr.070222182152
응용프로그램 이름 = db2bp
응용프로그램 ID = *LOCAL.jwr.070222182152
시퀀스 번호 = 00001
TP 모니터 클라이언트 사용자 ID =
TP 모니터 클라이언트 워크스테이션 이름 =
TP 모니터 클라이언트 응용프로그램 이름 =
TP 모니터 클라이언트 어카운팅 문자열 =

연결 요청 시작 시간소인 = 02/22/2007 13:21:52.587168
연결 요청 완료 시간소인 = 02/22/2007 13:21:53.291779
응용프로그램 유효 시간 =
CONNECT 권한 부여 ID = JWR
클라이언트 로그인 ID = jwr
클라이언트의 구성 NNAME = gilera
클라이언트 데이터베이스 관리 프로그램 제품 ID = SQL09050
클라이언트 응용프로그램의 프로세스 ID = 843852
클라이언트 응용프로그램의 플랫폼 = AIX 64BIT
클라이언트의 통신 프로토콜 = 로컬 클라이언트

인바운드 통신 주소 = *LOCAL.jwr

데이터베이스 이름 = SAMPLE
데이터베이스 경로 = /home/jwr/jwr/NODE0000/SQL00001/
클라이언트 데이터베이스 별명 = SAMPLE
입력 데이터베이스 별명 =
최종 재설정 시간소인 =
스냅샷 시간소인 = 02/22/2007 13:22:39.766300
권한 부여된 권한 부여 레벨 =
  사용자 권한:
    DBADM 권한
    CREATETAB 권한
    BINDADD 권한
    CONNECT 권한
    CREATE_NOT_FENC 권한
    LOAD 권한
    IMPLICIT_SCHEMA 권한
    CREATE_EXT_RT 권한
    QUIESCE_CONN 권한
  그룹 권한:
    SYSADM 권한
    CREATETAB 권한
    BINDADD 권한
    CONNECT 권한
    IMPLICIT_SCHEMA 권한
코디네이팅 데이터베이스 파티션 번호 = 0
현재 데이터베이스 파티션 번호 = 0

```

코디네이터 에이전트 프로세스 또는 스레드 ID	= 1801	
현재 워크로드 ID	= 1	
유실된 에이전트 수	= 0	
잠금 대기 중인 에이전트 수	= 1	
최대 연관 에이전트 수	= 1	
작업 중인 응용 프로그램 에이전트의 우선순위	= 0	
우선순위 유형	= 동적	
잠금 시간종료(초)	= -1	
응용 프로그램별 보유된 잠금 수	= 4	
연결 이후의 잠금 대기 수	= 1	
응용 프로그램의 잠금 대기 시간(밀리초)	= 20268	
검출된 교착 상태 수	= 0	
잠금 에스컬레이션 수	= 0	
배타적 잠금 에스컬레이션 수	= 0	
연결된 이후의 잠금 시간종료 수	= 0	
UOW의 총 잠금 대기 시간(밀리초)	= 수집되지 않음	
총 정렬 수	= 0	
총 정렬 시간(밀리초)	= 수집되지 않음	
총 정렬 오버플로우 수	= 0	
버퍼 풀 데이터 논리적 읽기 수	= 수집되지 않음	
버퍼 풀 데이터 실제 읽기 수	= 수집되지 않음	
버퍼 풀 임시 데이터 논리적 읽기 수	= 수집되지 않음	
버퍼 풀 임시 데이터 실제 읽기 수	= 수집되지 않음	
버퍼 풀 데이터 쓰기 수	= 수집되지 않음	
버퍼 풀 인덱스 논리적 읽기 수	= 수집되지 않음	
버퍼 풀 인덱스 실제 읽기 수	= 수집되지 않음	
버퍼 풀 임시 인덱스 논리적 읽기 수	= 수집되지 않음	
버퍼 풀 임시 인덱스 실제 읽기 수	= 수집되지 않음	
버퍼 풀 인덱스 쓰기 수	= 수집되지 않음	
버퍼 풀 xda 논리적 읽기 수	= 수집되지 않음	
버퍼 풀 xda 실제 읽기 수	= 수집되지 않음	
버퍼 풀 임시 xda 논리적 읽기 수	= 수집되지 않음	
버퍼 풀 임시 xda 실제 읽기 수	= 수집되지 않음	
버퍼 풀 xda 쓰기 수	= 수집되지 않음	
전체 버퍼 풀 읽기 시간(밀리초)	= 수집되지 않음	
전체 버퍼 풀 쓰기 시간(밀리초)	= 수집되지 않음	
프리페치를 위한 대기 시간(밀리초)	= 수집되지 않음	
읽지 않은 프리페치 페이지 수	= 수집되지 않음	
직접 읽기 수	= 수집되지 않음	
직접 쓰기 수	= 수집되지 않음	
직접 읽기 요청 수	= 수집되지 않음	
직접 쓰기 요청 수	= 수집되지 않음	
직접 읽기 경과 시간(밀리초)	= 수집되지 않음	
직접 쓰기 경과 시간(밀리초)	= 수집되지 않음	
마지막 커밋 후의 SQL 요청 수	= 3	
커밋문 수	= 0	
롤백문 수	= 0	
시도된 동적 SQL문 수	= 3	
시도된 정적 SQL문 수	= 0	
실패한 명령문 조작 수	= 0	
실행된 Select SQL문 수	= 1	
실행된 Xquery문	= 0	
실행된 Update/Insert/Delete문 수	= 0	
실행된 DDL문 수	= 0	
비활성 명령문 실행기록 메모리 사용(바이트)	= 0	
내부 자동 리바인드 수	= 0	
삭제된 내부 행 수	= 0	
삽입된 내부 행 수	= 0	
갱신된 내부 행 수	= 0	
내부 커밋 수	= 1	
내부 롤백 수	= 0	
교착 상태로 인한 내부 롤백 수	= 0	
시도된 바인드/프리컴파일 수	= 0	
삭제된 행 수	= 0	
삽입된 행 수	= 0	
갱신된 행 수	= 0	
선택된 행 수	= 0	

읽은 행 수	= 95	
기록된 행 수	= 0	
사용된 UOW 로그 스페이스(바이트)	= 수집되지 않음	
이전의 UOW 완료 시간소인	= 수집되지 않음	
마지막 완료된 uow의 경과 시간(sec.ms)	= 수집되지 않음	
UOW 시작 시간소인	= 수집되지 않음	
UOW 중지 시간소인	= 수집되지 않음	
UOW 완료 상태	= 수집되지 않음	
열린 리모트 커서 수	= 0	
블로킹으로 열린 리모트 커서 수	= 0	
거부된 리모트 커서 블로킹 요청 수	= 0	
수락된 리모트 커서 블로킹 요청 수	= 1	
열린 로컬 커서 수	= 1	
블로킹으로 열린 로컬 커서 수	= 1	
에이전트가 사용하는 전체 사용자 CPU 시간	= 0.019150	
에이전트가 사용하는 전체 시스템 CPU 시간	= 0.001795	
호스트 실행 경과 시간	= 0.012850	
패키지 캐시 찾아보기 수	= 2	
패키지 캐시 삽입 수	= 1	
응용프로그램 섹션 찾아보기 수	= 3	
응용프로그램 섹션 삽입 수	= 1	
카탈로그 캐시 찾아보기 수	= 11	
카탈로그 캐시 삽입 수	= 8	
카탈로그 캐시 오버플로우 수	= 0	
카탈로그 캐시 상위 워터 마크(water mark)	= 0	
작업 스페이스 정보		
공유 상위 워터 마크(water mark)	= 0	
총 공유 오버플로우 수	= 0	
총 공유 섹션 삽입 수	= 0	
총 공유 섹션 찾아보기 수	= 0	
개인용 상위 워터 마크(water mark)	= 0	
총 개인용 오버플로우 수	= 0	
총 개인용 삽입 수	= 0	
총 개인용 섹션 찾아보기 수	= 0	
가장 최근의 조작	= 페치	
커서 이름	= SQLCUR201	
가장 최근의 조작 시작 시간소인	= 02/22/2007 13:22:19.497439	
가장 최근의 조작 중지 시간소인	=	
응용프로그램과 연관된 에이전트 수	= 1	
해시 조인 수	= 0	
해시 루프 수	= 0	
해시 조인 오버플로우 수	= 0	
작은 해시 조인 오버플로우 수	= 0	
OLAP 기능 수	= 0	
OLAP 기능 오버플로우 수	= 0	
명령문 유형	= 동적 SQL문	
명령문	= 페치	
섹션 번호	= 201	
응용프로그램 작성자	= NULLID	
패키지 이름	= SQLC2G11	
일관성 토큰	= AAAAANBX	
패키지 버전 ID	=	
커서 이름	= SQLCUR201	
명령문 데이터베이스 파티션 번호	= 0	
명령문 시작 시간소인	= 02/22/2007 13:22:19.497439	
명령문 중지 시간소인	=	
마지막 완료된 stmt의 경과 시간(sec.ms)	= 0.000289	
전체 명령문 사용자 CPU 시간	= 0.002172	
전체 명령문 시스템 CPU 시간	= 0.001348	
SQL 컴파일러 비용 추정치(timeron)	= 14	
SQL 컴파일러 카디널리티 추정치	= 57	
요청된 병렬 처리 등급	= 1	
명령문에서 작업 중인 에이전트 수	= 1	



```

명령문에 대해 작성된 서브에이전트 수 = 1
명령문 정렬 수 = 0
총 정렬 시간 = 0
정렬 오버플로 = 0
읽은 행 수 = 0
기록된 행 수 = 0
삭제된 행 수 = 0
갱신된 행 수 = 0
삽입된 행 수 = 0
폐치된 행 수 = 0
버퍼 풀 데이터 논리적 읽기 수 = 수집되지 않음
버퍼 풀 데이터 실제 읽기 수 = 수집되지 않음
버퍼 풀 임시 데이터 논리적 읽기 수 = 수집되지 않음
버퍼 풀 임시 데이터 실제 읽기 수 = 수집되지 않음
버퍼 풀 인덱스 논리적 읽기 수 = 수집되지 않음
버퍼 풀 인덱스 논리적 읽기 수 = 수집되지 않음
버퍼 풀 임시 인덱스 논리적 읽기 수 = 수집되지 않음
버퍼 풀 임시 인덱스 실제 읽기 수 = 수집되지 않음
버퍼 풀 xda 논리적 읽기 수 = 수집되지 않음
버퍼 풀 xda 실제 읽기 수 = 수집되지 않음
버퍼 풀 임시 xda 논리적 읽기 수 = 수집되지 않음
버퍼 풀 임시 xda 실제 읽기 수 = 수집되지 않음
블로킹 커서 = YES
동적 SQL문 텍스트:
select * from org

```

```

에이전트 프로세스/스레드 ID = 1801

```

응용프로그램의 메모리 사용:

```

메모리 풀 유형 = 응용프로그램 힙
현재 크기(바이트) = 65536
상위 워터 마크(water mark)(바이트) = 65536
구성된 크기(바이트) = 1048576

```

```

에이전트 프로세스/스레드 ID = 1801

```

```

에이전트 잠금 시간종료(초) = -1
에이전트의 메모리 사용:

```

```

메모리 풀 유형 = 기타 메모리
현재 크기(바이트) = 589824
상위 워터 마크(water mark)(바이트) = 786432
구성된 크기(바이트) = 34359738368

```

```

잠금을 보유한 에이전트 I = 34
잠금을 보유한 응용프로그램 = *LOCAL.jwr.070222182158
잠금 이름 = 0x0002000E0000000000000000054
잠금 속성 = 0x00000000
릴리스 플래그 = 0x00000001
잠금 오브젝트 = 테이블
잠금 모드 = 배타적 잠금(X)
요청된 잠금 모드 = 공유 잠금 가능(IS)
잠금을 보유한 테이블 스페이스 이름 = USERSPACE1
잠금을 보유한 테이블의 스키마 = JWR
잠금을 보유한 테이블 이름 = ORG
잠금을 보유한 테이블의 데이터 파티션 ID = 0
잠금 대기 시작 시간소인 = 02/22/2007 13:22:19.497833

```

- 모든 데이터베이스에 대한 스냅샷 정보를 요청하려면 다음 명령을 발행하십시오.

```
get snapshot for all databases
```

다음은 위의 명령으로 인해 표시되는 샘플 출력입니다.

#### Database Snapshot

```

데이터베이스 이름 = SAMPLE
데이터베이스 경로 = C:\WDB2\NODE0000\SQL00001W
입력 데이터베이스 별명 =
데이터베이스 상태 = 활동 중

```

```

카탈로그 데이터베이스 파티션 번호 = 0
카탈로그 네트워크 노드 이름 =
데이터베이스 서버에서 실행 중인 운영 체제 = NT
데이터베이스 위치 = 로컬
첫 번째 데이터베이스 연결 시간소인 = 06/21/2007 14:46:49.771064
최종 재설정 시간소인 =
최종 백업 시간소인 =
스냅샷 시간소인 = 06/21/2007 14:51:50.235993

자동 스토리지 경로 수 = 1
자동 스토리지 경로 = C:
노드 번호 = 0

연결에 대한 상위 워터 마크(water mark) = 6
응용프로그램 연결 수 = 4
총 2차 연결 수 = 4
현재 연결된 응용프로그램 수 = 1
db 관리 프로그램에서 현재 실행 중인 응용프로그램 수 = 0
응용프로그램과 연관된 에이전트 수 = 5
응용프로그램과 연관된 최대 에이전트 수 = 6
최대 코디네이팅 에이전트 수 = 6

임계값 위반 수 = 0
현재 보유한 잠금 수 = 0
잠금 대기 수 = 0
데이터베이스의 잠금 대기 시간(밀리초) = 수집되지 않음
사용 중인 잠금 목록 메모리(바이트) = 2256
검출된 교착 상태 수 = 0
잠금 에스컬레이션 수 = 0
배타적 잠금 에스컬레이션 수 = 0
현재 잠금 대기 중인 에이전트 수 = 0
잠금 시간종료 수 = 0
인다우트(indoubt) 트랜잭션 수 = 0

할당된 총 개인용 정렬 힙 수 = 0
할당된 총 공유 정렬 힙 = 0
공유 정렬 힙 상위 워터 마크(water mark) = 0
포스트 임계값 정렬(공유 메모리) = 수집되지 않음
총 정렬 수 = 0
총 정렬 시간(밀리초) = 수집되지 않음
정렬 오버플로 = 0
사용 중인 정렬 수 = 0

버퍼 풀 데이터 논리적 읽기 수 = 수집되지 않음
버퍼 풀 데이터 실제 읽기 수 = 수집되지 않음
버퍼 풀 임시 데이터 논리적 읽기 수 = 수집되지 않음
버퍼 풀 임시 데이터 실제 읽기 수 = 수집되지 않음
비동기식 풀 데이터 페이지 읽기 수 = 수집되지 않음
버퍼 풀 데이터 쓰기 수 = 수집되지 않음
비동기식 풀 데이터 페이지 쓰기 수 = 수집되지 않음
버퍼 풀 인덱스 논리적 읽기 수 = 수집되지 않음
버퍼 풀 인덱스 실제 읽기 수 = 수집되지 않음
버퍼 풀 임시 인덱스 논리적 읽기 수 = 수집되지 않음
버퍼 풀 임시 인덱스 실제 읽기 수 = 수집되지 않음
비동기 풀 인덱스 페이지 읽기 수 = 수집되지 않음
버퍼 풀 인덱스 쓰기 수 = 수집되지 않음
비동기식 풀 인덱스 페이지 쓰기 수 = 수집되지 않음
버퍼 풀 xda 논리적 읽기 수 = 수집되지 않음
버퍼 풀 xda 실제 읽기 수 = 수집되지 않음
버퍼 풀 임시 xda 논리적 읽기 수 = 수집되지 않음
버퍼 풀 임시 xda 실제 읽기 수 = 수집되지 않음
버퍼 풀 xda 쓰기 수 = 수집되지 않음
비동기 풀 xda 페이지 읽기 수 = 수집되지 않음
비동기 풀 xda 페이지 쓰기 수 = 수집되지 않음
전체 버퍼 풀 읽기 시간(밀리초) = 수집되지 않음
전체 버퍼 풀 쓰기 시간(밀리초) = 수집되지 않음
비동기식 읽기의 전체 경과 시간 = 수집되지 않음
비동기식 쓰기의 전체 경과 시간 = 수집되지 않음
비동기식 데이터 읽기 요청 수 = 수집되지 않음
비동기식 인덱스 읽기 요청수 = 수집되지 않음
비동기 xda 읽기 요청 수 = 수집되지 않음

```

사용 가능한 희생 버퍼 없음	= 수집되지 않음
LSN Gap 클리너 트리거 수	= 수집되지 않음
더티 페이지 스틸 클리너 트리거 수	= 수집되지 않음
더티 페이지 임계값 클리너 트리거 수	= 수집되지 않음
프리페치를 위한 대기 시간(밀리초)	= 수집되지 않음
읽지 않은 프리페치 페이지 수	= 수집되지 않음
직접 읽기 수	= 수집되지 않음
직접 쓰기 수	= 수집되지 않음
직접 읽기 요청 수	= 수집되지 않음
직접 쓰기 요청 수	= 수집되지 않음
직접 읽기 경과 시간(밀리초)	= 수집되지 않음
직접 쓰기 경과 시간(밀리초)	= 수집되지 않음
닫힌 데이터베이스 파일 수	= 수집되지 않음
벡터된 IO 수	= 수집되지 않음
벡터된 IO의 페이지 수	= 수집되지 않음
블록 IO 수	= 수집되지 않음
블록 IO의 페이지 수	= 수집되지 않음
호스트 실행 경과 시간	= 수집되지 않음
시도된 커미트문 수	= 0
시도된 롤백문 수	= 0
시도된 동적 명령문 수	= 6
시도된 정적 명령문 수	= 3
실패한 명령문 조작 수	= 0
실행된 Select SQL문 수	= 0
실행된 Xquery문	= 0
실행된 Update/Insert/Delete문 수	= 0
실행된 DDL문 수	= 0
비활성 명령문 실행기록 메모리 사용(바이트)	= 0
내부 자동 리바인드 수	= 0
삭제된 내부 행 수	= 0
삽입된 내부 행 수	= 0
갱신된 내부 행 수	= 0
내부 커미트 수	= 6
내부 롤백 수	= 0
교착 상태로 인한 내부 롤백 수	= 0
정리 보류 중인 MDC 테이블 블록 수	= 0
삭제된 행 수	= 0
삽입된 행 수	= 0
갱신된 행 수	= 0
선택된 행 수	= 0
읽은 행 수	= 98
시도된 바인드/프리컴파일 수	= 0
데이터베이스에 사용 가능한 로그 스페이스(바이트)	= 20400000
데이터베이스에서 사용 중인 로그 스페이스(바이트)	= 0
사용된 최대 2차 로그 스페이스(바이트)	= 0
사용된 최대 전체 로그 스페이스(바이트)	= 0
현재 할당된 2차 로그 수	= 0
읽은 로그 페이지 수	= 0
로그 읽기 시간(sec.ns)	= 0.000000004
기록된 로그 페이지 수	= 0
로그 쓰기 시간(sec.ns)	= 0.000000004
숫자 쓰기 로그 IO	= 0
숫자 읽기 로그 IO	= 0
숫자 부분 페이지 로그 IO	= 0
숫자 로그 버퍼 전체	= 0
버퍼에서 로그 데이터 찾기	= 0
가장 오래된 트랜잭션을 보류 중인 응용프로그램 ID	= 93
복구를 위해 다시 실행할 로그(바이트)	= 0
더티 페이지로 인한 로그 어카운트(바이트)	= 0
노드 번호	= 0
처음 사용 중인 로그의 파일 번호	= 0
마지막 사용 중인 로그의 파일 번호	= 2
현재 사용 중인 로그의 파일 번호	= 0
아카이브 되고 있는 로그 파일 번호	= 적용되지 않음

```

패키지 캐시 찾아보기 수 = 6
패키지 캐시 삽입 수 = 0
패키지 캐시 오버플로우 수 = 0
패키지 캐시 상위 워터 마크(water mark)(바이트) = 196608
응용프로그램 섹션 찾아보기 수 = 6
응용프로그램 섹션 삽입 수 = 0

```

```

카탈로그 캐시 찾아보기 수 = 37
카탈로그 캐시 삽입 수 = 10
카탈로그 캐시 오버플로우 수 = 0
카탈로그 캐시 상위 워터 마크(water mark) = 65536
카탈로그 캐시 통계 크기 = 0

```

작업 스페이스 정보

```

공유 상위 워터 마크(water mark) = 0
해당 공유 오버플로우 수 = 0
총 공유 섹션 삽입 수 = 0
총 공유 섹션 찾아보기 수 = 0
개인용 상위 워터 마크(water mark) = 0
해당 개인용 오버플로우 수 = 0
총 개인용 삽입 수 = 0
총 개인용 섹션 찾아보기 수 = 0

```

```

해시 조인 수 = 0
해시 루프 수 = 0
해시 조인 오버플로우 수 = 0
작은 해시 조인 오버플로우 수 = 0
포스트 임계값 해시 조인 수(공유 메모리) = 0
활동 중인 해시 조인 수 = 0

```

```

OLAP 기능 수 = 0
OLAP 기능 오버플로우 수 = 0
활성 OLAP 기능 = 0

```

```

통계 가공 = 수집되지 않음
동기 Runstats = 수집되지 않음
비동기 Runstats = 수집되지 않음
총 통계 가공 시간(밀리초) = 수집되지 않음
총 동기 Runstats 시간(밀리초) = 수집되지 않음

```

데이터베이스의 메모리 사용:

```

노드 번호 = 0
메모리 풀 유형 = 백업/리스트오어/유틸리티 힙
현재 크기(바이트) = 65536
상위 워터 마크(water mark)(바이트) = 65536
구성된 크기(바이트) = 20512768

```

```

노드 번호 = 0
메모리 풀 유형 = 패키지 캐시 힙
현재 크기(바이트) = 196608
상위 워터 마크(water mark)(바이트) = 196608
구성된 크기(바이트) = 402653184

```

```

노드 번호 = 0
메모리 풀 유형 = 기타 메모리
현재 크기(바이트) = 131072
상위 워터 마크(water mark)(바이트) = 131072
구성된 크기(바이트) = 20971520

```

```

노드 번호 = 0
메모리 풀 유형 = 카탈로그 캐시 힙
현재 크기(바이트) = 65536
상위 워터 마크(water mark)(바이트) = 65536
구성된 크기(바이트) = 402653184

```

```

노드 번호 = 0
메모리 풀 유형 = 버퍼 풀 힙
2차 ID = 1

```

현재 크기(바이트)	= 2424832
상위 워터 마크(water mark)(바이트)	= 2424832
구성된 크기(바이트)	= 402653184
노드 번호	= 0
메모리 풀 유형	= 버퍼 풀 힙
2차 ID	= 시스템 32k 버퍼 풀
현재 크기(바이트)	= 851968
상위 워터 마크(water mark)(바이트)	= 851968
구성된 크기(바이트)	= 402653184
노드 번호	= 0
메모리 풀 유형	= 버퍼 풀 힙
2차 ID	= 시스템 16k 버퍼 풀
현재 크기(바이트)	= 589824
상위 워터 마크(water mark)(바이트)	= 589824
구성된 크기(바이트)	= 402653184
노드 번호	= 0
메모리 풀 유형	= 버퍼 풀 힙
2차 ID	= 시스템 8k 버퍼 풀
현재 크기(바이트)	= 458752
상위 워터 마크(water mark)(바이트)	= 458752
구성된 크기(바이트)	= 402653184
노드 번호	= 0
메모리 풀 유형	= 버퍼 풀 힙
2차 ID	= 시스템 4k 버퍼 풀
현재 크기(바이트)	= 393216
상위 워터 마크(water mark)(바이트)	= 393216
구성된 크기(바이트)	= 402653184
노드 번호	= 0
메모리 풀 유형	= 공유 정렬 힙
현재 크기(바이트)	= 0
상위 워터 마크(water mark)(바이트)	= 0
구성된 크기(바이트)	= 20512768
노드 번호	= 0
메모리 풀 유형	= 잠금 관리 프로그램 힙
현재 크기(바이트)	= 327680
상위 워터 마크(water mark)(바이트)	= 327680
구성된 크기(바이트)	= 393216
노드 번호	= 0
메모리 풀 유형	= 데이터베이스 힙
현재 크기(바이트)	= 10551296
상위 워터 마크(water mark)(바이트)	= 10551296
구성된 크기(바이트)	= 12582912
노드 번호	= 0
메모리 풀 유형	= 응용프로그램 힙
2차 ID	= 97
현재 크기(바이트)	= 65536
상위 워터 마크(water mark)(바이트)	= 65536
구성된 크기(바이트)	= 1048576
노드 번호	= 0
메모리 풀 유형	= 응용프로그램 힙
2차 ID	= 96
현재 크기(바이트)	= 65536
상위 워터 마크(water mark)(바이트)	= 65536
구성된 크기(바이트)	= 1048576
노드 번호	= 0
메모리 풀 유형	= 응용프로그램 힙
2차 ID	= 95
현재 크기(바이트)	= 65536
상위 워터 마크(water mark)(바이트)	= 65536
구성된 크기(바이트)	= 1048576

```

노드 번호                = 0
메모리 풀 유형          = 응용프로그램 힙
  2차 ID                = 94
  현재 크기(바이트)     = 65536
  상위 워터 마크(water mark)(바이트) = 65536
  구성된 크기(바이트)   = 1048576

```

```

노드 번호                = 0
메모리 풀 유형          = 응용프로그램 힙
  2차 ID                = 93
  현재 크기(바이트)     = 65536
  상위 워터 마크(water mark)(바이트) = 65536
  구성된 크기(바이트)   = 1048576

```

```

노드 번호                = 0
메모리 풀 유형          = 응용프로그램 공유 힙
  현재 크기(바이트)     = 65536
  상위 워터 마크(water mark)(바이트) = 65536
  구성된 크기(바이트)   = 20512768

```

사용자 권한은 사용자에게 부여된 모든 권한 및 역할을 나타내고, 그룹 권한은 그룹에 부여된 모든 권한 및 역할을 나타냅니다.

- SAMPLE 데이터베이스에 연결된 응용프로그램 핸들 765가 있는 특정 응용프로그램에 대한 스냅샷 정보를 요청하려면 다음을 발행하십시오.

```
get snapshot for application agentid 765
```

- SAMPLE 데이터베이스에 대한 동적 SQL 스냅샷 정보를 요청하려면 다음 명령을 발행하십시오.

```
get snapshot for dynamic sql on sample
```

## 사용 시 참고사항

- 데이터베이스에 대해 쓰기 일시중단이 ON인 경우, 쓰기 일시중단이 OFF로 설정될 때까지 해당 데이터베이스에 대해 스냅샷을 발행할 수 없습니다. 쓰기 일시중단이 ON으로 설정된 데이터베이스에 대해 스냅샷이 발행된 경우, 진단 프로브는 db2diag 로그 파일에 기록되고 해당 데이터베이스는 건너뛵니다.
- 리모트 인스턴스(또는 다른 로컬 인스턴스)에서 스냅샷을 확보하려면 우선 해당 인스턴스에 접속해야 합니다. 다른 인스턴스에 있는 데이터베이스 별명을 지정하는 경우에는 오류 메시지가 리턴됩니다.
- 일부 통계를 얻으려면 데이터베이스 시스템 모니터 스위치를 켜야 합니다. 기록 스위치 TIMESTAMP가 OFF로 설정된 경우, 시간소인 관련 요소는 "Not Collected"를 보고합니다.
- 다음 사항 중 하나라도 해당하면 테이블 정보가 요청되어도 데이터가 리턴되지 않습니다.
  - TABLE 기록 스위치가 꺼집니다.
  - 스위치를 켜 후 액세스한 테이블이 없습니다.
  - 마지막 RESET MONITOR 명령이 발행된 이후에 액세스한 테이블이 없습니다.

그러나 REORG TABLE이 수행되거나 이 기간 동안 수행된 경우, 일부 필드가 표시되지 않아도 일부 정보는 리턴됩니다.

- 모든 데이터베이스 파티션의 스냅샷 정보(모든 파티션의 집계 결과와 다른)를 얻으려면 스냅샷 관리 뷰를 사용해야 합니다.
- 파티션된 데이터베이스 환경에서, **GLOBAL** 옵션과 함께 명령을 지정하면 모든 노드 중 각각의 상위 워터 마크(water mark)의 합이 아니라 모든 노드 간의 연결에 대해서 가장 높은 상위 워터 마크(water mark)를 나타내는 연결에 대한 상위 워터 마크(water mark) 매개변수 값을 리턴합니다. 예를 들어, 다음과 같습니다.
  - 노드 A에는 현재 연결된 5개의 응용프로그램이 있고 연결에 대한 상위 워터 마크(water mark)는 5입니다.
  - 노드 B에는 현재 연결된 4개의 응용프로그램이 있고 연결에 대한 상위 워터 마크(water mark)는 6입니다.

위의 예에서. High water mark for connections 값은 6이고 Applications connected currently 값은 9입니다.

## 호환성

버전 8 이전 버전과의 호환성:

- **DBPARTITIONNUM**에 키워드 **NODE**를 대체할 수 있습니다.
- **DBPARTITIONNUMS**에 키워드 **NODES**를 대체할 수 있습니다.
- 버전 9.5에서의 새 레지스트리 변수인 **DB2\_SYSTEM\_MONITOR\_SETTINGS**는 Linux에서 CPU 사용을 모니터링하는 동작에 영향을 줍니다. Linux에서 시스템 및 사용자 CPU 사용 시간 모두를 리턴하는 CPU 사용 읽기 메소드를 사용해야 하는 경우 다음 조치 중 하나를 수행하십시오.

Linux의 RHEL4 및 SLES9에서:

```
db2set DB2_SYSTEM_MONITOR_SETTINGS=DISABLE_CPU_USAGE:TRUE
```

Linux의 RHEL5 및 SLES10에서:

```
db2set DB2_SYSTEM_MONITOR_SETTINGS=OLD_CPU_USAGE:TRUE
```





---

## 제 63 장 HELP

사용자가 정보 센터에서 도움말을 호출할 수 있도록 허용합니다.

이 명령은 UNIX 기반 시스템에서 사용할 수 없습니다.

### 권한 부여

없음

### 필수 연결

없음

### 명령 구문

▶▶—HELP—◀◀

### 예:

다음 예는 HELP 명령을 사용하는 방법을 나타냅니다.

- db2 help

이 명령은 DB2 정보 센터를 여는데, 여기에는 태스크, 참조, 책 등과 같은 범주로 나뉘어진 DB2에 관한 정보가 들어 있습니다. 이것은 매개변수 없이 db2ic 명령을 호출하는 것과 동등합니다.

### 사용 시 참고사항

명령행 처리기는 명령의 성공 여부를 모르며 오류 조건을 보고할 수 없습니다.



## 제 64 장 HISTORY

CLP 대화식 모드 세션에서 실행되는 명령 실행기록을 표시합니다.

범위

이 명령은 CLP 대화식 모드에서만 실행될 수 있습니다. 특히 CLP 명령 모드 또는 CLP 일괄처리 모드에서 실행할 수 없습니다.

권한 부여

없음

필수 연결

없음

명령 구문



명령 매개변수

**REVERSE | R**

명령 실행기록을 최근에 실행된 명령을 먼저 나열하여 반대 순서로 표시합니다. 이 매개변수가 지정되지 않으면 최근에 실행된 명령이 마지막에 나열되어 순서대로 명령이 나열됩니다.

*num* 최신 *num* 명령만 표시합니다. 이 매개변수가 지정되지 않으면 최대 20개 명령만 표시됩니다. 그러나 표시되는 명령 수는 명령 실행기록에 저장되는 명령의 수로도 제한됩니다.

사용 시 참고사항

1. DB2\_CLP\_HISTSZIE 레지스트리 변수 값은 명령 실행기록에 저장될 최대 명령 수를 지정합니다. 이 레지스트리 변수는 1과 500 사이의 임의의 값으로 설정할 수 있습니다. 이 레지스트리 변수가 설정되지 않거나 유효한 범위 밖의 값으로 설정되는 경우 최대 20개 명령이 명령 실행기록에 저장됩니다.
2. HISTORY 명령은 항상 명령 실행기록에 나열되므로 표시되는 최대 명령 수는 항상 사용자 지정 최대값보다 하나 더 많습니다.
3. 명령 실행기록은 CLP 대화식 모드 세션에서 지속되지 않습니다. 즉, 명령 실행기록은 대화식 모드 세션 종료 시 저장되지 않습니다.

4. 동시에 실행되는 여러 CLP 대화식 모드 세션의 명령 실행기록은 서로 독립적입니다.

---

## 제 65 장 IMPORT

지원되는 파일 형식을 갖는 외부 파일의 데이터를 테이블, 계층 구조, 뷰 또는 별칭으로 데이터를 삽입합니다. LOAD가 더 빠른 방법이기도 하지만 로드 유틸리티는 계층 구조 레벨에서 데이터 로드를 지원하지 않습니다.

304 페이지의 『임포트 유틸리티의 파일 유형 수정자』로의 빠른 링크.

### 권한 부여

- **INSERT** 옵션을 사용하여 IMPORT하려면 다음 중 하나가 필요합니다.
  - *dataaccess* 권한
  - 각 참여 중인 테이블, 뷰 또는 별칭에 대한 CONTROL 특권
  - 각 참여 중인 테이블 또는 뷰에 대한 INSERT 및 SELECT 특권
- **INSERT\_UPDATE** 옵션을 사용하여 기존 테이블로 IMPORT하려면 다음 중 하나가 필요합니다.
  - *dataaccess* 권한
  - 각 참여 중인 테이블, 뷰 또는 별칭에 대한 CONTROL 특권
  - 각 참여 중인 테이블 또는 뷰에 대한 INSERT, SELECT, UPDATE 및 DELETE 특권
- **REPLACE** 또는 **REPLACE\_CREATE** 옵션을 사용하여 기존 테이블로 IMPORT하려면 다음 중 하나가 필요합니다.
  - *dataaccess* 권한
  - 테이블 또는 뷰에 대한 CONTROL 특권
  - 테이블 또는 뷰에 대한 INSERT, SELECT 및 DELETE 특권
- **CREATE** 또는 **REPLACE\_CREATE** 옵션을 사용하여 새 테이블로 IMPORT하려면 다음 중 하나가 필요합니다.
  - *dbadm* 권한
  - 데이터베이스에 대한 CREATETAB 권한 및 테이블 스페이스에 대한 USE 특권은 다음 경우 중 하나입니다.
    - 테이블의 내재적 또는 명시적 스키마 이름이 존재하지 않을 경우, 데이터베이스에 대한 IMPLICIT\_SCHEMA 권한
    - 테이블의 스키마 이름이 기존의 스키마를 참조할 경우, 스키마에 대한 CREATEIN 특권
- **CREATE** 또는 **REPLACE\_CREATE** 옵션을 사용하여 존재하지 않는 계층 구조로 IMPORT하려면 다음 중 하나가 필요합니다.

- *dbadm* 권한
  - 데이터베이스에 대한 **CREATETAB** 권한과 테이블 스페이스에 대한 **USE** 특권 및 다음 중 하나가 필요합니다.
    - 테이블의 스키마 이름이 존재하지 않을 경우, 데이터베이스에 대한 **IMPLICIT\_SCHEMA** 권한
    - 테이블의 스키마가 존재할 경우, 스키마에 대한 **CREATEIN** 특권
    - 전체 계층 구조에 대한 **REPLACE\_CREATE** 옵션이 사용될 경우, 계층 구조의 모든 서브테이블에 대한 **CONTROL** 특권
- **REPLACE** 옵션을 사용하여 기존 계층 구조로 **IMPORT**하려면 다음 중 하나가 필요합니다.
  - *dataaccess* 권한
  - 계층 구조에 있는 모든 서브테이블에 대한 **CONTROL** 특권
- 컬럼이 보호 설정된 테이블로 데이터를 임포트하려면 세션 권한 부여 ID가 테이블에 있는 보호 설정된 모든 컬럼에 대해 쓰기 액세스할 수 있도록 하는 **LBAC** 증명서를 가지고 있어야 합니다. 그렇지 않으면 임포트를 실패하고 오류(**SQLSTATE 42512**)가 리턴됩니다.
- 행이 보호 설정된 테이블로 데이터를 임포트하려면 세션 권한 부여 ID가 다음 기준을 충족하는 **LBAC** 증명서를 보유해야 합니다.
  - 테이블을 보호하는 보안 규정에 포함됨
  - 세션 권한 부여 ID에 쓰기 액세스에 대한 권한이 부여됨
 삽입할 행에 대한 레이블, 사용자의 **LBAC** 신임, 보안 규정 정의 및 **LBAC** 규칙이 행에 대한 레이블을 판별합니다.
- **REPLACE** 또는 **REPLACE\_CREATE** 옵션을 지정할 경우 세션 권한 부여 ID는 테이블을 삭제할 수 있는 권한이 있어야 합니다.
- 데이터를 별칭으로 임포트하려면 세션 권한 부여 ID에 **pass-through** 모드에서 지정된 데이터 소스를 액세스하고 사용할 특권이 있어야 합니다.

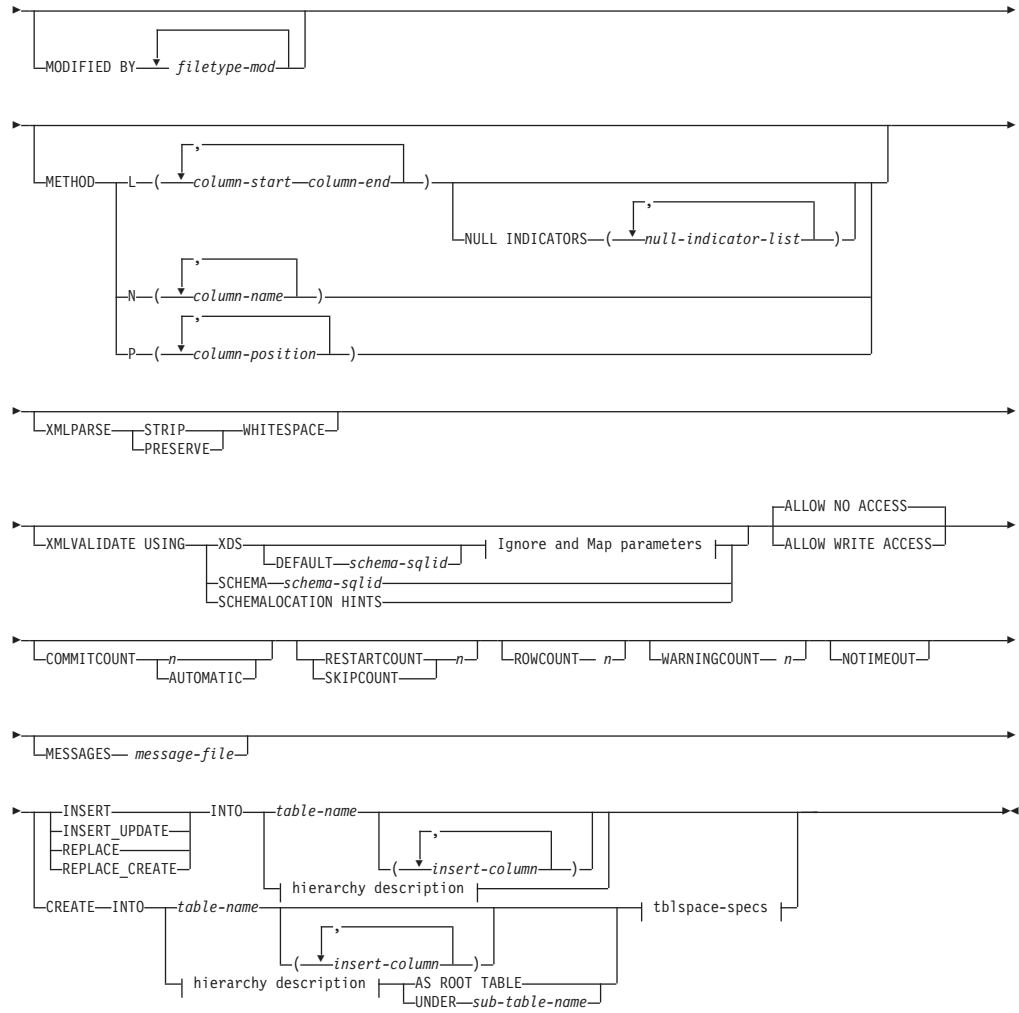
## 필수 연결

데이터베이스. 내재된 연결이 사용 가능한 경우 디폴트 데이터베이스에 연결됩니다. Linux, UNIX 또는 Windows 클라이언트로부터 Linux, UNIX 또는 Windows 데이터베이스 서버에 대한 유틸리티 액세스는 **DB2 Connect** 게이트웨이 또는 루프 백 환경을 통하지 않고 엔진을 통해 직접 연결되어야 합니다.

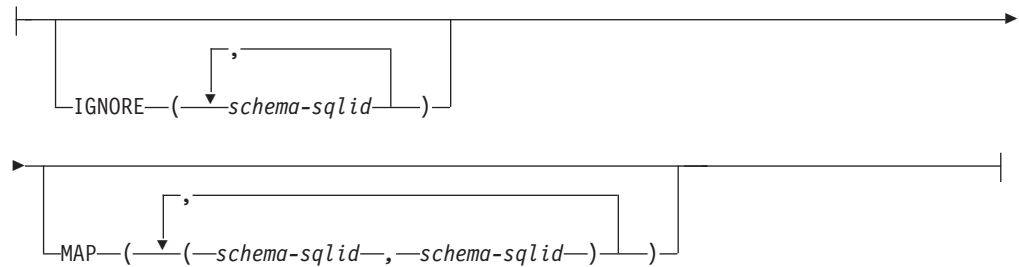
## 명령 구문

```

▶▶—IMPORT FROM—filename—OF—filetype—
       LOBS FROM—lob-path—
       XML FROM—xml-path—
  
```



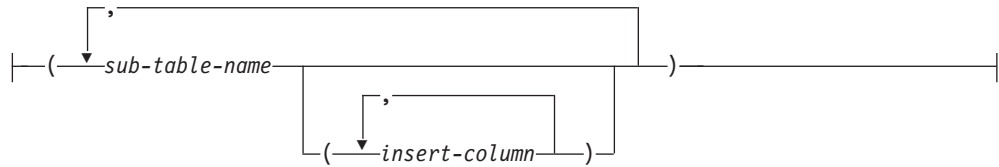
**Ignore and Map parameters:**



**hierarchy description:**



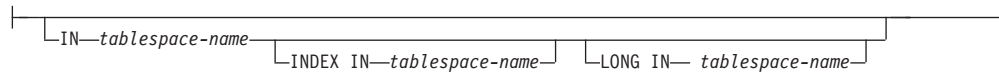
### sub-table-list:



### traversal-order-list:



### tblspace-specs:



## 명령 매개변수

### ALL TABLES

계층 구조 전용 내재적 키워드. 계층 구조를 임포트할 때 디폴트로 트래버스 순서로 지정된 모든 테이블을 임포트합니다.

### ALLOW NO ACCESS

오프라인 모드로 임포트를 실행합니다. 행이 삽입되기 전에 목표 테이블에 대해 독점(X) 잠금을 획득합니다. 이렇게 하면 동시 응용프로그램이 테이블 데이터에 액세스하지 못하게 합니다. 이는 디폴트 임포트 동작입니다.

### ALLOW WRITE ACCESS

온라인 모드로 임포트를 실행합니다. 첫 번째 행이 삽입될 때 목표 테이블에 대해 의도를 가진 독점(IX) 잠금을 획득합니다. 이렇게 하면 동시 판독기 및 기록기가 테이블 데이터에 액세스할 수 있습니다. 온라인 모드는 **REPLACE**, **CREATE** 또는 **REPLACE\_CREATE** 임포트 옵션과 호환 가능하지 않습니다. 온라인 모드는 버퍼링된 삽입과 함께 지원되지 않습니다. 임포트 조작은 테이블 잠금에 대한 잠금 에스컬레이션을 방지하고 사용 중인 로그 스페이스 외부에서 실행되지 않도록 하기 위해 삽입된 데이터를 주기적으로 커밋합니다. 이러한 커밋은 **COMMITCOUNT** 옵션을 사용하지 않는 경우에도 수행됩니다. 각 커밋을 수행하는 동안 임포트는 해당 IX 테이블 잠금을 유실하므로 커밋 후 이를 다시 획득하려고 시도합니다. 별칭으로 임포트할 때 이 매개변수가 필요하며 유효한 숫자를 사용하여 **COMMITCOUNT**를 지정해야 합니다(AUTOMATIC은 유효한 옵션으로 간주하지 않음).



## AS ROOT TABLE

하나 이상의 서브테이블을 독립형 테이블 계층 구조로 작성합니다.

## COMMITCOUNT *n* | AUTOMATIC

*n*개의 레코드가 모두 임포트된 후 COMMIT를 수행합니다. 숫자 *n*을 지정하면 임포트는 *n*개의 레코드가 모두 임포트된 후 COMMIT를 수행합니다. 복합 삽입을 사용할 때, 사용자 지정 커밋 빈도 *n*은 복합 계수 값의 첫 번째 정수 배수로 받아들여집니다. AUTOMATIC을 지정하면 임포트는 내부적으로 커밋을 수행해야 하는 때를 판별합니다. 이 유틸리티는 다음 두 가지 이유 중 하나에 대해 커밋합니다.

- 사용 중인 로그 스페이스 외부에서 실행하지 않도록 하기 위해
- 행 레벨에서 테이블 레벨로 잠금 에스컬레이션되지 않도록 하기 위해

ALLOW WRITE ACCESS 옵션을 지정하고 COMMITCOUNT 옵션을 지정하지 않으면, 임포트 유틸리티는 COMMITCOUNT AUTOMATIC이 지정된 것과 같이 커밋을 수행합니다.

사용 중인 로그 스페이스를 모두 사용해버리지 않도록 하기 위한 임포트 조작의 기능은 DB2 레지스트리 변수 DB2\_FORCE\_APP\_ON\_MAX\_LOG에 의해 영향을 받습니다.

- DB2\_FORCE\_APP\_ON\_MAX\_LOG가 FALSE로 설정되고 COMMITCOUNT AUTOMATIC 명령 옵션이 지정된 경우, 임포트 유틸리티는 자동으로 사용 중인 로그 스페이스를 모두 사용해버리지 않도록 할 수 있습니다.
- DB2\_FORCE\_APP\_ON\_MAX\_LOG가 FALSE로 설정되고 COMMITCOUNT *n* 명령 옵션이 지정된 경우, 레코드를 삽입하거나 갱신하는 중 SQL0964C(트랜잭션 로그가 가득참) 발생 시 임포트 유틸리티는 로그 가득참 조건을 해결하려고 합니다. 무조건 커밋을 수행한 후 레코드를 삽입하거나 갱신하려고 재시도합니다. 이것이 로그 가득참 조건(로그 가득참이 데이터베이스의 다른 활동에 영향을 주는 경우)을 해결하는 데 유용하지 않을 경우, IMPORT 명령은 예상대로 실패합니다. 그러나 커밋되는 행 수는 COMMITCOUNT *n* 값의 배수가 되지 않을 수 있습니다. 임포트 조작 재시도 시 이미 커밋된 행 처리를 피하려면, RESTARTCOUNT 또는 SKIPCOUNT 명령 매개변수를 사용하십시오.
- DB2\_FORCE\_APP\_ON\_MAX\_LOG가 TRUE로 설정된 경우(디폴트), 레코드를 삽입하거나 갱신하는 중 SQL0964C 발생 시 임포트 조작이 실패합니다. COMMITCOUNT AUTOMATIC 또는 COMMITCOUNT *n* 지정 여부와 관계 없이 이런 현상이 일어날 수 있습니다.

응용프로그램에서 데이터베이스가 강제 해제되고 현재 작업 단위(UOW)가 롤백됩니다. импорт 조작 재시도 시 이미 커밋된 행 처리를 피하려면, **RESTARTCOUNT** 또는 **SKIPCOUNT** 명령 매개변수를 사용하십시오.

## CREATE

주: **CREATE** 매개변수는 사용되지 않으며 추후 릴리스에서 제거될 수 있습니다. 추가적인 세부사항은 『IMPORT 명령 옵션 **CREATE** 및 **REPLACE\_CREATE**가 사용되지 않음』을 참조하십시오.

데이터베이스의 코드 페이지에 테이블 정의 및 행 내용을 작성합니다. DB2 테이블, 서브테이블 또는 계층 구조에서 데이터를 익스포트한 경우, 인덱스가 작성됩니다. 이 옵션이 계층 구조에서 작동하고 DB2에서 데이터를 익스포트한 경우, 유형 계층 구조도 작성됩니다. 이 옵션은 IXF 파일에 대해서만 사용할 수 있습니다.

별칭으로 импорт할 때 이 매개변수는 유효하지 않습니다.

주: MVS™ 호스트 데이터베이스에서 데이터를 익스포트하고 호스트 페이지 크기에서 계산된 길이가 254를 초과하는 LONGVAR 필드를 포함할 경우, 행이 너무 길기 때문에 **CREATE**에 실패할 수 있습니다. 제한사항 목록에 대해서는 『인포트된 테이블 재작성』을 참조하십시오. 이 경우, 테이블을 직접 작성한 후 **INSERT**를 사용하여 **IMPORT**를 호출하거나 **LOAD** 명령을 사용해야 합니다.

### DEFAULT *schema-sqlid*

이 옵션은 **USING XDS** 매개변수가 지정된 경우에만 사용할 수 있습니다. **DEFAULT** 절을 통해 지정된 스키마는 인포트된 XML 문서의 XDS(XML Data Specifier)가 XML 스키마를 식별하는 SCH 속성을 포함하지 않을 때 유효성 확인에 사용할 스키마를 식별합니다.

**DEFAULT** 절은 **IGNORE** 및 **MAP** 절보다 우선순위를 갖습니다. XDS가 **DEFAULT** 절을 충족시키면 **IGNORE** 및 **MAP** 스펙은 무시됩니다.

### FROM *filename*

인포트될 데이터가 들어 있는 파일을 지정합니다. 경로가 생략된 경우 현재 작업 디렉토리를 사용합니다.

### HIERARCHY

계층 구조 데이터가 인포트되도록 지정합니다.

### IGNORE *schema-sqlid*

이 옵션은 **USING XDS** 매개변수가 지정된 경우에만 사용할 수 있습니다. SCH 속성에 의해 식별될 경우 **IGNORE** 절은 무시할 하나 이상의 스키마 목록을 지정합니다. 인포트된 XML 문서에 대한 XDS(XML Data Specifier)에 SCH

속성이 존재하고 SCH 속성에 의해 식별된 스키마가 무시할 스키마 목록에 포함된 경우, 이 임포트된 XML 문서에 대해서는 스키마 유효성 확인이 발생하지 않습니다.

**IGNORE** 절에 스키마가 지정되어 있으면 이 스키마는 또한 **MAP** 절에 있는 스키마 쌍의 왼쪽에 존재할 수 없습니다.

**IGNORE** 절은 XDS에만 적용됩니다. **MAP** 절에 의해 맵핑된 스키마는 **IGNORE** 절에 의해 지정된 경우 계속 무시되지 않습니다.

#### **IN** *tablespace-name*

테이블이 작성될 테이블 스페이스를 식별합니다. 테이블 스페이스가 있어야 하며 이 테이블 스페이스는 **REGULAR** 테이블 스페이스이어야 합니다. 다른 테이블 스페이스를 지정하지 않으면 모든 테이블 부분이 이 테이블 스페이스에 저장됩니다. 이 절을 지정하지 않으면 테이블은 권한 부여 ID가 작성한 테이블 스페이스에 작성됩니다. 테이블 스페이스를 찾을 수 없으면 테이블은 디폴트 테이블 스페이스 **USERSPACE1**에 넣어집니다. **USERSPACE1**가 삭제된 경우 테이블 작성에 실패합니다.

#### **INDEX IN** *tablespace-name*

테이블의 인덱스가 작성될 테이블 스페이스를 식별합니다. 이 옵션은 **IN** 절에 지정된 1차 테이블 스페이스가 **DMS** 테이블 스페이스일 경우에만 허용됩니다. 지정된 테이블 스페이스가 있어야 하며 이 테이블 스페이스는 **REGULAR** 또는 **LARGE DMS** 테이블 스페이스이어야 합니다.

주: 인덱스를 포함할 테이블 스페이스는 테이블이 작성되었을 때만 지정할 수 있습니다.

#### *insert-column*

데이터가 삽입될 테이블 또는 뷰에 있는 컬럼 이름을 지정합니다.

#### **INSERT**

기존 테이블 데이터를 변경하지 않고 임포트된 데이터를 테이블에 추가합니다.

#### **INSERT\_UPDATE**

임포트된 데이터 행을 목표 테이블에 추가하거나 기본 키가 일치하는 기존 행 (목표 테이블의)을 갱신합니다.

#### **INTO** *table-name*

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

완전한 테이블 이름 또는 규정되지 않은 테이블 이름을 사용해야 하는 이전 서버의 경우를 제외하고는 **INSERT**, **INSERT\_UPDATE** 또는 **REPLACE**에

별명을 사용할 수 있습니다. 규정된 테이블 이름의 형식은 *schema.tablename* 입니다. *schema*는 테이블이 작성된 사용자 이름입니다.

#### **LOBS FROM** *lob-path*

LOB 파일을 저장하는 하나 이상의 경로를 지정합니다. LOB 데이터 파일의 이름은 기본 데이터 파일(ASC, DEL 또는 IXF)에 저장되는데 LOB 컬럼으로 로드될 컬럼에 저장됩니다. 지정할 수 있는 최대 경로 수는 999입니다. 이는 내재적으로 LOBSINFILE 동작을 활성화합니다.

별칭으로 임포트할 때 이 매개변수는 유효하지 않습니다.

#### **LONG IN** *tablespace-name*

긴 컬럼(LONG VARCHAR, LONG VARGRAPHIC, LOB 데이터 유형 또는 이들 유형 중 하나를 소스 유형으로 갖는 구별 유형)의 값이 저장될 테이블 공간을 식별합니다. 이 옵션은 **IN** 절에 지정된 1차 테이블 공간이 DMS 테이블 공간일 경우에만 허용됩니다. 테이블 공간이 있어야 하며 LARGE DMS 테이블 공간이어야 합니다.

#### **MAP** *schema-sqlid*

이 옵션은 **USING XDS** 매개변수가 지정된 경우에만 사용할 수 있습니다. 임포트된 각 XML 문서에 대한 XDS(XML Data Specifier)의 SCH 속성이 지정하는 스키마 대신 사용할 대체 스키마를 지정하려면 이 **MAP** 절을 사용하십시오. **MAP** 절은 하나 이상의 스키마 쌍의 목록을 지정하며, 여기서 각 쌍은 한 스키마 대 다른 스키마의 매핑을 나타냅니다. 쌍에서 첫 번째 스키마는 XDS에 있는 SCH 속성에 의해 참조되는 스키마를 나타냅니다. 쌍에서 두 번째 스키마는 스키마 유효성 확인을 수행하는 데 사용되어야 하는 스키마를 나타냅니다.

**MAP** 절에 있는 스키마 쌍의 왼쪽에 있는 스키마는 **IGNORE** 절에 지정될 수 없습니다.

스키마 쌍 매핑이 적용된 후, 최종 결과가 됩니다. 매핑 조작은 전이되지 않으므로 선택된 스키마는 다른 스키마 상 매핑에 계속 적용되지 않습니다.

스키마가 두 번 이상 매핑될 수 없다는 것은 쌍의 왼쪽에 두 번 이상 나타날 수 없다는 것을 의미합니다.

#### **MESSAGES** *message-file*

임포트 조작 중 발생하는 경고 및 오류 메시지에 대한 목적지를 지정합니다. 파일이 이미 존재할 경우 임포트 유틸리티는 정보를 추가합니다. 파일에 대한 전체 경로를 지정하지 않으면 유틸리티는 현재 디렉토리 및 디폴트 드라이브를 목적으로 사용합니다. *message-file*이 생략될 경우 메시지는 표준 출력으로 쓰여집니다.

#### **METHOD**

**L** 데이터를 임포트할 시작 및 끝 컬럼 번호를 지정합니다. 컬럼 번호는 데이터 행이 시작되는 바이트 오프셋입니다. 1부터 번호가 매겨집니다.

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

**N** 임포트할 데이터 파일의 컬럼 이름을 지정합니다. 이 컬럼 이름의 대소문자는 시스템 카탈로그의 해당 이름의 대소문자와 일치해야 합니다. 널(NULL) 입력이 가능하지 않은 각 테이블 컬럼에는 **METHOD N** 목록에 해당 항목이 있어야 합니다. 예를 들면, 주어진 데이터 필드 F1, F2, F3, F4, F5 및 F6과 테이블 컬럼 C1 INT, C2 INT NOT NULL, C3 INT NOT NULL 및 C4 INT, method N (F2, F1, F4, F3)은 유효한 요청인 반면, method N (F2, F1)은 유효하지 않습니다.

주: 이 방법은 IXF 파일에 대해서만 사용할 수 있습니다.

**P** 임포트될 입력 데이터 필드의 필드 번호를 지정합니다.

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

#### **MODIFIED BY** *filetype-mod*

파일 유형 수정자 옵션을 지정합니다. 304 페이지의 『임포트 유틸리티의 파일 유형 수정자』을 참조하십시오.

#### **NOTIMEOUT**

임포트 유틸리티가 잠금 대기 중 시간종료되지 않도록 지정합니다. 이 옵션은 **locktimeout** 데이터베이스 구성 매개변수를 대체합니다. 다른 응용프로그램에는 영향을 주지 않습니다.

#### **NULL INDICATORS** *null-indicator-list*

이 옵션은 **METHOD L** 매개변수가 지정된 경우에만 사용할 수 있습니다. 즉, 입력 파일은 ASC 파일입니다. 널(NULL) 표시기 목록은 각 널(NULL) 표시기 필드의 컬럼 번호를 지정하는 쉽표로 구분된 양의 정수 목록입니다. 컬럼 번호는 데이터 행이 시작되는 널(NULL) 표시기 필드의 바이트 오프셋입니다. **METHOD L** 매개변수에 정의된 각 데이터 필드에 대해 한 개의 항목이 널(NULL) 표시기 목록에 있어야 합니다. 컬럼 번호 0은 해당 데이터 필드에 데이터가 항상 들어 있음을 나타냅니다.

널(NULL) 표시기의 Y 값은 컬럼 데이터를 널(NULL)로 지정합니다. 널(NULL) 표시기 컬럼에서 Y 이외의 모든 문자는 컬럼 데이터가 널(NULL)이 아니며 **METHOD L** 옵션이 지정하는 컬럼 데이터가 임포트됨을 지정합니다.

nullindchar 파일 유형 수정자와 함께 **MODIFIED BY** 옵션을 사용하여 널(NULL) 표시기 문자를 변경할 수 있습니다.

## OF filetype

입력 파일에 있는 데이터의 형식을 지정합니다.

- ASC(컬럼 식별자가 없는 ASCII 형식)
- DEL(컬럼 식별자가 있는 ASCII 형식) - 여러 데이터베이스 관리 프로그램 및 파일 관리자 프로그램에서 사용됩니다.
- WSF(작업시트 형식) - 다음과 같은 프로그램에서 사용됩니다.
  - Lotus 1-2-3
  - Lotus Symphony
- IXF(Integration Exchange Format, PC 버전)는 DB2에 의해 독점 사용되는 2진 형식입니다.

**중요사항:** WSF 파일 형식에 대한 지원은 사용되지 않으므로 추후 릴리스에서 제거될 수 있습니다. 지원이 제거되기 전에 WSF 파일 대신 지원되는 파일 형식을 사용하여 시작하는 것이 좋습니다.

별칭으로 임포트할 때 WSF 파일 유형은 지원되지 않습니다.

## REPLACE

데이터 오브젝트를 잘라내어 테이블에서 모든 기존 데이터를 삭제한 후 임포트된 데이터를 삽입합니다. 테이블 정의 및 인덱스 정의는 변경되지 않습니다. 이 옵션은 테이블이 존재할 경우에만 사용할 수 있습니다. 계층 구조 간에 데이터를 이동할 때 이 옵션을 사용하면 개별적 서브테이블이 아닌 전체 계층 구조에 대한 데이터만 바꿀 수 있습니다.

별칭으로 임포트할 때 이 매개변수는 유효하지 않습니다.

이 옵션은 CREATE TABLE문의 NOT LOGGED INITIALLY(NLI)절이나 ALTER TABLE문의 ACTIVE NOT LOGGED INITIALLY절을 인정하지 않습니다.

NLI절이 호출된 CREATE TABLE 또는 ALTER TABLE문과 동일한 트랜잭션에서 **REPLACE** 옵션을 사용한 임포트를 수행할 경우, 임포트는 NLI절을 인정하지 않습니다. 모든 삽입은 로그됩니다.

### 일시적인 해결책 1

DELETE문을 사용하여 테이블의 내용을 삭제한 후 INSERT문을 사용하여 임포트를 호출하십시오.

### 일시적인 해결책 2

테이블을 삭제하고 재작성한 후 INSERT문을 사용하여 임포트를 호출하십시오.

이 제한사항은 DB2 Universal Database 버전 7 및 DB2 UDB 버전 8에 적용됩니다.

## REPLACE\_CREATE

주: **REPLACE\_CREATE** 매개변수는 사용되지 않으며 추후 릴리스에서 제거될 수 있습니다. 추가적인 세부사항은 『IMPORT 명령 옵션 CREATE 및 REPLACE\_CREATE가 사용되지 않음』을 참조하십시오.

테이블이 존재할 경우, 데이터 오브젝트를 잘라내어 테이블에서 모든 기존 데이터를 삭제한 후 테이블 정의 또는 인덱스 정의를 변경하지 않고 импорт된 데이터를 삽입합니다.

테이블이 존재하지 않을 경우, 데이터베이스의 코드 페이지에 행 내용과 함께 테이블 및 인덱스 정의를 작성합니다. 제한사항 목록에 대해서는 **인포트된 테이블 재작성**을 참조하십시오.

이 옵션은 IXF 파일에 대해서만 사용할 수 있습니다. 계층 구조 간에 데이터를 이동하는 경우에 이 옵션을 사용하면 개별 부속 테이블이 아닌 전체 계층 구조에 대한 데이터만 바뀔 수 있습니다.

별칭으로 임포트할 때 이 매개변수는 유효하지 않습니다.

## RESTARTCOUNT *n*

인포트 조작이 레코드  $n + 1$ 에서 시작되도록 지정합니다. 첫 번째  $n$  레코드는 건너뜁니다. 이 솔루션은 기능적으로 **SKIPCOUNT**와 동일합니다. **RESTARTCOUNT** 및 **SKIPCOUNT**는 상호 독점적입니다.

## ROWCOUNT *n*

파일에 있는  $n$ 개의 실제 레코드가 인포트(삽입되거나 갱신됨)되도록 지정합니다. 사용자가 **SKIPCOUNT** 또는 **RESTARTCOUNT** 옵션에 의해 판별된 레코드로부터  $n$ 개의 행만을 파일에서 인포트할 수 있게 합니다. **SKIPCOUNT** 또는 **RESTARTCOUNT** 옵션이 지정되지 않은 경우 첫 번째  $n$  행만 인포트합니다. **SKIPCOUNT  $m$**  또는 **RESTARTCOUNT  $m$** 이 지정된 경우  $m+1$ 에서  $m+n$ 까지의 행을 인포트합니다. 복합 삽입을 사용할 때, **ROWCOUNT  $n$** 은 복합 계수 값의 첫 번째 정수 배수로 반올림됩니다.

## SKIPCOUNT *n*

인포트 조작이 레코드  $n + 1$ 에서 시작되도록 지정합니다. 첫 번째  $n$  레코드는 건너뜁니다. 이 옵션은 기능적으로 **RESTARTCOUNT**와 동일합니다. **SKIPCOUNT** 및 **RESTARTCOUNT**는 상호 독점적입니다.

## STARTING *sub-table-name*

*sub-table-name*부터 시작하고 디폴트 순서를 요청하며 계층 구조 전용 키워드입니다. PC/IXF 파일의 경우 디폴트 순서는 입력 파일에 저장된 순서입니다. PC/IXF 파일 형식에는 디폴트 순서만 사용할 수 있습니다.

### *sub-table-list*

**INSERT** 또는 **INSERT\_UPDATE** 옵션을 사용하는 유형이 지정된 테이블의 경우, 데이터가 임포트될 서브테이블을 표시하기 위해 서브테이블 이름 목록을 사용합니다.

### *traversal-order-list*

**INSERT**, **INSERT\_UPDATE** 또는 **REPLACE** 옵션을 사용하는 유형이 지정된 테이블의 경우, 계층 구조에서 임포트 중인 서브테이블의 트래버스 순서를 표시하기 위해 서브테이블 이름 목록을 사용합니다.

### **UNDER** *sub-table-name*

하나 이상의 서브테이블을 작성하기 위한 상위 테이블을 지정합니다.

### **WARNINGCOUNT** *n*

*n*번의 경고 후 임포트 조작을 중지합니다. 경고가 예상되지는 않지만 올바른 파일 및 테이블이 사용되고 있는지에 대한 검증을 원할 경우 이 매개변수를 설정하십시오. 임포트 파일 또는 목표 테이블이 올바르지 않게 지정될 경우 임포트 유틸리티는 임포트하려는 각 행에 대해 경고를 생성하므로 임포트를 실패하게 됩니다. *n*이 0이거나 이 옵션이 지정되지 않은 경우 임포트 조작은 발행된 경고 수에 관계없이 계속 수행됩니다.

### **XML FROM** *xml-path*

XML 파일이 들어 있는 하나 이상의 경로를 지정합니다.

### **XMLPARSE**

XML 문서가 구문 분석되는 방법을 지정합니다. 이 옵션을 지정하지 않을 경우, XML 문서에 대한 구문 분석 동작은 CURRENT XMLPARSE OPTION 특수 레지스터의 값으로 판별됩니다.

### **STRIP WHITESPACE**

XML 문서가 구문 분석될 때 공백을 제거하도록 지정합니다.

### **PRESERVE WHITESPACE**

XML 문서가 구문 분석될 때 공백을 제거하지 않도록 지정합니다.

### **XMLVALIDATE**

XML 문서가 스키마에 대해 유효성이 확인되도록 지정합니다(해당되는 경우).

### **USING XDS**

기본 데이터 파일의 XDS(XML Data Specifier)에 의해 식별된 XML 스키마에 대해 XML 문서의 유효성을 확인합니다. **USING XDS**와 함께 **XMLVALIDATE** 옵션을 호출한 경우, 유효성 확인을 수행하는 데 사용된 스키마는 디폴트로 XDS의 SCH 속성에 의해 판별됩니다. SCH 속성이 XDS에 존재하지 않을 경우, **DEFAULT** 절로 디폴트 스키마를 지정하지 않으면 스키마 유효성 확인이 발생하지 않습니다.



**DEFAULT, IGNORE** 및 **MAP** 절은 스키마 판별 동작을 수정하는데 사용될 수 있습니다. 이들 세 개의 선택적 절은 XDS의 권장 스펙에 직접적으로 적용되며 서로에게는 적용되지 않습니다. 예를 들어, 한 스키마가 **DEFAULT** 절에서 지정되어 선택되었으면 이 스키마는 **IGNORE** 절에 지정되어도 무시되지 않습니다. 마찬가지로 한 스키마가 **MAP** 절에서 첫 번째 파트 쌍으로 지정되어 선택되면 다른 **MAP** 절 쌍의 두 번째 파트에 지정되어도 다시 맵핑되지 않습니다.

#### **USING SCHEMA** *schema-sqlid*

XML 문서가 지정된 SQL ID가 있는 XML 스키마에 대해 유효성이 확인됩니다. 이 경우 모든 XML 컬럼에 대해 XDS(XML Data Specifier)의 SCH 속성이 무시됩니다.

#### **USING SCHEMALOCATION HINTS**

XML 문서가 소스 XML 문서의 XML 스키마 위치 힌트에 의해 식별된 스키마에 대해 유효성이 확인됩니다. XML 문서에서 schemaLocation 속성을 찾을 수 없으면 유효성 확인이 발생하지 않습니다. **USING SCHEMALOCATION HINTS** 절을 지정하면 모든 XML 컬럼에 대해 XDS(XML Data Specifier)의 SCH 속성이 무시됩니다.

아래의 **XMLVALIDATE** 옵션 예를 참조하십시오.

**예:**

#### **예 1**

다음 예는 myfile.ixf에서 STAFF 테이블로 정보를 임포트하는 방법을 보여 줍니다.

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

SQL3150N PC/IXF 파일의 H 레코드는 제품이 "DB2 01.00", 날짜가 "19970220", 시간이 "140848"입니다.

SQL3153N PC/IXF 파일에서 T 레코드의 이름은 "myfile", 규정자는 " "이고, 소스는 " "입니다.

SQL3109N 유틸리티가 파일 "myfile"에서 데이터를 로드하기 시작합니다.

SQL3110N 유틸리티가 처리를 완료했습니다.  
입력 파일에서 "58"개의 행을 읽었습니다.

SQL3221W ...COMMIT WORK 시작. 입력 레코드 계수 = "58".

SQL3222W 데이터베이스 변경사항의 ...COMMIT가 성공적이었습니다.

SQL3149N 입력 파일에서 "58"개의 행이 처리되었습니다.  
"58"개의 행이 테이블에 삽입되었습니다. "0"개의 행이 거부되었습니다.

#### **예 2(식별 컬럼이 있는 테이블로 임포트)**

TABLE1에는 다음과 같이 4개의 컬럼이 있습니다.

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

TABLE2는 TABLE1과 동일한데 다른 점은 C2가 ENERATED ALWAYS 식별 컬럼인 점입니다.

DATAFILE1의 데이터 레코드(DEL 형식):

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

DATAFILE2의 데이터 레코드(DEL 형식):

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

다음 명령은 DATAFILE1에 1행과 2행에 대한 식별 값이 제공되지 않았기 때문에 이들 행에 대한 식별 값을 생성합니다. 그러나 3행과 4행에는 각각 사용자 제공 ID 값 100과 101이 지정됩니다.

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

DATAFILE1을 TABLE1으로 임포트하여 식별 값이 모든 행에 대해 생성되도록 하려면 다음 명령 중 하나를 발행하십시오.

```
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 TABLE1으로 임포트하여 식별 값이 각 행에 대해 생성되도록 하려면 다음 명령 중 하나를 발행하십시오.

```
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 임포트하면 1행과 2행은 삽입되지만 3행과 4행은 거부됩니다. 이는 이들 행이 자체의 널(NULL)이 아닌 값을 제공하며 식별 컬럼이 GENERATED ALWAYS이기 때문입니다.

## **XMLVALIDATE절 사용의 예**

### **예 1(XMLVALIDATE USING XDS)**

다음 XMLVALIDATE 절의 경우:

```
XMLVALIDATE USING XDS
  IGNORE (S1.SCHEMA_A)
  MAP ((S1.SCHEMA_A, S2.SCHEMA_B))
```

이 임포트는 S1.SCHEMA\_A의 **IGNORE**가 S1.SCHEMA\_A 대 S2.SCHEMA\_B의 **MAP**과 충돌하므로 유효하지 않은 구문 때문에 실패합니다.

### **예 2(XMLVALIDATE USING XDS)**

다음 XMLVALIDATE 절의 경우:

```
XMLVALIDATE USING XDS
  DEFAULT S8.SCHEMA_H
  IGNORE (S9.SCHEMA_I, S10.SCHEMA_J)
  MAP ((S1.SCHEMA_A, S2.SCHEMA_B), (S3.SCHEMA_C, S5.SCHEMA_E),
      (S6.SCHEMA_F, S3.SCHEMA_C), (S4.SCHEMA_D, S7.SCHEMA_G))
```

다음 XDS를 포함하는 XML 컬럼의 경우:

```
<XDS FIL='xmlfile.001.xml' />
```

"xmlfile.001.xml" 파일의 문서를 유효성 확인하는 데 SQL ID "S8.SCHEMA\_H"를 사용 하는 XML 스키마를 사용합니다. 이는 "S8.SCHEMA\_H"가 디폴트 스키마로 사용되도록 지정되었기 때문입니다.

다음 XDS를 포함하는 XML 컬럼의 경우:

```
<XDS FIL='xmlfile.002.xml' OFF='10' LEN='500' SCH='S10.SCHEMA_J' />
```

"xmlfile.002.xml" 파일의 문서에 대해 스키마 유효성 확인이 발생하지 않습니다. 이는 XDS가 "S10.SCHEMA\_J"를 스키마로 사용하도록 지정하지만 이 스키마가 **IGNORE** 절에 포함되어 있기 때문입니다. 문서 내용은 파일의 바이트 오프셋 10(11 번째 바이트를 의미함)에서 찾을 수 있으며 길이는 500바이트입니다.

다음 XDS를 포함하는 XML 컬럼의 경우:

```
<XDS FIL='xmlfile.003.xml' SCH='S6.SCHEMA_F' />
```

"xmlfile.003.xml" 파일의 문서를 유효성 확인하는 데 SQL ID "S3.SCHEMA\_C"를 사용하는 XML 스키마를 사용합니다. 이는 **MAP** 절이 "S6.SCHEMA\_F" 스키마가 "S3.SCHEMA\_C" 스키마로 맵핑되도록 지정하기 때문입니다. 추가 맵핑이 일어나지 않으므로 이 경우 "S3.SCHEMA\_C" 스키마 대 "S5.SCHEMA\_E" 스키마의 맵핑이 적용되지 않는 점을 참고하십시오.

다음 XDS를 포함하는 XML 컬럼의 경우:

```
<XDS FIL='xmlfile.004.xml' SCH='S11.SCHEMA_K' />
```

"xmlfile.004.xml" 파일의 문서를 유효성 확인하는 데 SQL ID "S11.SCHEMA\_K" 를 사용하는 XML 스키마를 사용합니다. 이 경우 **DEFAULT**, **IGNORE** 또는 **MAP** 스펙 중 어느 것도 적용되지 않음을 참고하십시오.

### 예 3(XMLVALIDATE USING XDS)

다음 **XMLVALIDATE** 절의 경우:

```
XMLVALIDATE USING XDS
  DEFAULT S1.SCHEMA_A
  IGNORE (S1.SCHEMA_A)
```

다음 XDS를 포함하는 XML 컬럼의 경우:

```
<XDS FIL='xmlfile.001.xml' />
```

"xmlfile.001.xml" 파일의 문서를 유효성 확인하는 데 SQL ID "S1.SCHEMA\_A" 를 사용 하는 XML 스키마를 사용합니다. 이는 "S1.SCHEMA\_1"이 디폴트 스키마로 사용되도록 지정되었기 때문입니다.

다음 XDS를 포함하는 XML 컬럼의 경우:

```
<XDS FIL='xmlfile.002.xml' SCH='S1.SCHEMA_A' />
```

"xmlfile.002" 파일의 문서에 대해 스키마 유효성 확인이 발생하지 않습니다. 이는 XDS가 "S1.SCHEMA\_A"를 스키마로 사용하도록 지정하지만 이 스키마가 **IGNORE** 절에 포함되어 있기 때문입니다.

### 예 4(XMLVALIDATE USING XDS)

다음 **XMLVALIDATE** 절의 경우:

```
XMLVALIDATE USING XDS
  DEFAULT S1.SCHEMA_A
  MAP ((S1.SCHEMA_A, S2.SCHEMA_B), (S2.SCHEMA_B, S1.SCHEMA_A))
```

다음 XDS를 포함하는 XML 컬럼의 경우:

```
<XDS FIL='xmlfile.001.xml' />
```

"xmlfile.001.xml" 파일의 문서를 유효성 확인하는 데 SQL ID "S1.SCHEMA\_A" 를 사용 하는 XML 스키마를 사용합니다. 이는 "S1.SCHEMA\_1"이 디폴트 스키마로 사용되도록 지정되었기 때문입니다.**DEFAULT** 절이 적용되었으므로 **MAP** 절이 뒤에 적용되지 않음을 참고하십시오. 따라서 이 경우 "S1.SCHEMA\_A" 스키마 대 "S2.SCHEMA\_B" 스키마의 맵핑이 적용되지 않습니다.

다음 XDS를 포함하는 XML 컬럼의 경우:

```
<XDS FIL='xmlfile.002.xml' SCH='S1.SCHEMA_A' />
```

"xmlfile.002.xml" 파일의 문서를 유효성 확인하는 데 SQL ID "S2.SCHEMA\_B" 를 사용하는 XML 스키마를 사용합니다. 이는 "S1.SCHEMA\_A" 스키마가 "S2.SCHEMA\_B" 스키마로 맵핑되도록 **MAP** 절이 지정하기 때문입니다. 추가 맵핑이 일어나지 않으므로 이 경우 "S2.SCHEMA\_B" 스키마 대 "S1.SCHEMA\_A" 스키마의 맵핑이 적용되지 않습니다.

다음 XDS를 포함하는 XML 컬럼의 경우:

```
<XDS FIL='xmlfile.003.xml' SCH='S2.SCHEMA_B' />
```

"xmlfile.003.xml" 파일의 문서를 유효성 확인하는 데 SQL ID "S1.SCHEMA\_A" 를 사용하는 XML 스키마를 사용합니다. 이는 **MAP** 절이 "S2.SCHEMA\_B" 스키마가 "S1.SCHEMA\_A" 스키마로 맵핑되도록 지정하기 때문입니다. 추가 맵핑이 일어나지 않으므로 이 경우 "S1.SCHEMA\_A" 스키마 대 "S2.SCHEMA\_B" 스키마의 맵핑이 적용되지 않는 점을 참고하십시오.

#### 예 5(XMLVALIDATE USING SCHEMA)

다음 **XMLVALIDATE** 절의 경우:

```
XMLVALIDATE USING SCHEMA S2.SCHEMA_B
```

다음 XDS를 포함하는 XML 컬럼의 경우:

```
<XDS FIL='xmlfile.001.xml' />
```

SQL ID "S2.SCHEMA\_B"를 사용하는 XML 스키마를 사용하여 xmlfile.001.xml 파일의 문서를 유효성 확인합니다.

다음 XDS를 포함하는 XML 컬럼의 경우:

```
<XDS FIL='xmlfile.002.xml' SCH='S1.SCHEMA_A' />
```

SQL ID "S2.SCHEMA\_B"를 사용하는 XML 스키마를 사용하여 "xmlfile.002.xml" 파일의 문서를 유효성 확인합니다. **USING SCHEMA** 절에서 지정하는 스키마를 사용하여 유효성 확인을 수행하므로 SCH 속성이 무시됨을 참고하십시오.

#### 예 6(XMLVALIDATE USING SCHEMALOCATION HINTS)

다음 XDS를 포함하는 XML 컬럼의 경우:

```
<XDS FIL='xmlfile.001.xml' />
```

사용된 XML 스키마는 문서 내용의 schemaLocation 속성에 의해 판별되며 스키마가 존재하지 않으면 유효성 확인이 일어나지 않습니다.

다음 XDS를 포함하는 XML 컬럼의 경우:

```
<XDS FIL='xmlfile.002.xml' SCH='S1.SCHEMA_A' />
```

사용된 XML 스키마는 문서 내용의 schemaLocation 속성에 의해 판별되며 스키마가 존재하지 않으면 유효성 확인이 일어나지 않습니다. **SCHEMALOCATION HINTS** 를 사용하여 유효성 확인을 수행하므로 SCH 속성이 무시됨을 참고하십시오.

## 사용 시 참고사항

임포트 작업을 시작하기 전에 반드시 모든 테이블 작업을 완료하고 모든 잠금을 릴리스하십시오. 이는 WITH HOLD로 열려진 모든 커서를 닫고 COMMIT를 발행하거나 ROLLBACK을 발행하여 수행할 수 있습니다.

임포트 유틸리티는 SQL INSERT문을 사용하여 목표 테이블에 행을 추가합니다. 이 유틸리티는 입력 파일에 있는 각 데이터 행마다 하나의 INSERT문을 발행합니다. INSERT문을 실패하면 다음 두 가지 조치 중 하나가 발생합니다.

- 후속 INSERT문이 성공할 수 있을 것 같으면 메시지 파일에 경고 메시지를 작성하고 처리를 계속합니다.
- 후속 INSERT문이 실패할 것 같으며 데이터베이스 손상이 생길 가능성이 있으면 메시지 파일에 오류 메시지를 작성하고 처리를 정지합니다.

이 유틸리티는 **REPLACE** 또는 **REPLACE\_CREATE** 조작 중 이전 행이 삭제된 후 자동 COMMIT를 수행합니다. 따라서 테이블 오브젝트가 잠려진 후 응용프로그램이 데이터베이스 관리 프로그램을 인터럽트하거나 시스템이 실패할 경우 이전 데이터가 모두 유실됩니다. 이들 옵션을 사용하기 전에 이전 데이터가 더 이상 필요하지 않은지 확인하십시오.

**CREATE**, **REPLACE** 또는 **REPLACE\_CREATE** 조작 중 로그가 가득차게 되면 이 유틸리티는 삽입된 레코드에 대해 자동 COMMIT를 수행합니다. 자동 COMMIT 후 응용프로그램이 데이터베이스 관리 프로그램을 인터럽트하거나 시스템이 실패할 경우 부분 데이터가 있는 테이블이 데이터베이스에 남아 있습니다. **REPLACE** 또는 **REPLACE\_CREATE** 옵션을 사용하여 전체 임포트 작업을 재실행하거나 성공적으로 임포트된 행 수로 설정된 **RESTARTCOUNT** 매개변수를 사용하여 **INSERT**를 사용하십시오.

디폴트로 **INSERT** 또는 **INSERT\_UPDATE** 옵션에 대해서는 자동 COMMIT가 수행되지 않습니다. 단, **COMMITCOUNT** 매개변수가 0이 아닌 경우에는 수행됩니다. 자동 COMMIT가 수행되지 않으면 가득찬 로그는 ROLLBACK됩니다.

다음 조건 중 하나라도 해당될 경우 오프라인 임포트는 자동 COMMIT를 수행하지 않습니다.

- 목표가 테이블이 아니고 뷰인 경우
- 복합 텍스트 삽입이 사용된 경우
- 버퍼링된 삽입이 사용된 경우

디폴트로 온라인 임포트는 자동 COMMIT를 수행하여 사용 중인 로그 스페이스와 잠금 목록을 모두 해제합니다. 자동 COMMIT는 COMMITCOUNT 값이 0으로 지정된 경우에만 수행되지 않습니다.

임포트 유틸리티가 COMMIT를 수행할 때마다 메시지 파일에 두 개의 메시지가 작성되는데, 하나는 커밋되는 레코드 수를 나타내고 다른 하나는 COMMIT 완료 후 작성됩니다. 실패 후 임포트 조작을 재시작할 때 마지막 완료된 COMMIT에서 판별된 대로 생략할 레코드 수를 지정하십시오.

임포트 유틸리티는 사소한 비호환성 문제점이 있는 입력 데이터는 승인합니다(예를 들어, 문자 데이터가 채우기 또는 절단을 사용하여 임포트되고 숫자 데이터가 다른 숫자 데이터 유형으로 임포트될 수 있습니다). 그러나 주요한 비호환성 문제점이 있는 데이터는 승인되지 않습니다.

오브젝트 테이블이 자체 테이블 외에 하위 테이블이 있는 경우 이 오브젝트 테이블을 REPLACE 또는 REPLACE\_CREATE 할 수 없으며, 기본 테이블에 하위 테이블(자체 테이블을 포함하여)이 있는 경우 해당 오브젝트 뷰를 REPLACE 또는 REPLACE\_CREATE 할 수 없습니다. 이러한 테이블 또는 뷰를 바꾸려면 다음을 수행하십시오.

1. 해당 테이블이 상위 테이블인 모든 외부 키를 삭제하십시오.
2. 임포트 유틸리티를 실행하십시오.
3. 테이블을 변경하여 외부 키를 재작성하십시오.

외부 키를 재작성하는 동안 오류가 발생하면 데이터를 수정하여 참조 무결성을 유지보수하십시오.

PC/IXF 파일에서 테이블을 재작성할 때 참조 제한조건 및 외부 키 정의는 보존되지 않습니다. (기본 키 정의는 데이터가 이전에 SELECT \*를 사용하여 익스포트된 경우에 보존됩니다.)

리모트 데이터베이스로 임포트하려면 입력 데이터 파일의 사본, 출력 메시지 파일 및 잠재적인 데이터베이스 크기 확장에 필요한 디스크 스페이스가 충분해야 합니다.

리모트 데이터베이스에 대해 임포트 조작이 수행되고 출력 메시지 파일이 매우 긴 경우(60KB 초과), 클라이언트의 사용자에게 리턴되는 메시지 파일은 임포트 조작 중간부터 메시지가 누락될 수 있습니다. 메시지 정보의 처음 30KB와 메시지 정보의 마지막 30KB는 항상 보유됩니다.

PC/IXF 파일이 디스켓 대신 하드 드라이브에 있을 경우 PC/IXF 파일을 리모트 데이터베이스로 임포트하면 훨씬 더 빨리 수행됩니다.

ASC, DEL 또는 WSF 파일 형식의 데이터를 임포트하려면 먼저 데이터베이스 테이블 또는 계층 구조가 존재해야 합니다. 그러나 테이블이 이미 존재하지 않을 경우

**IMPORT CREATE** 또는 **IMPORT REPLACE\_CREATE**가 PC/IXF 파일에서 데이터를 임포트할 때 테이블을 작성합니다. 유형이 지정된 테이블에서 **IMPORT CREATE**는 유형 계층 구조와 테이블 계층 구조를 작성할 수 있습니다.

데이터베이스 간에 데이터를 이동하려면(계층 데이터를 포함하여) PC/IXF 임포트를 사용해야 합니다. 행 구분자가 있는 문자 데이터를 컬럼 식별자가 있는 ASCII(DEL) 파일로 익스포트하며 텍스트 전송 프로그램으로 이를 처리할 경우 행 구분자가 있는 필드는 줄어들거나 확장됩니다. 동일한 클라이언트에서 소스 또는 목표 데이터베이스에 모두 액세스할 수 있는 경우 파일 복사 단계는 필요하지 않습니다.

ASC 및 DEL 파일의 데이터는 임포트를 수행하는 클라이언트 응용프로그램의 코드 페이지에 있는 것으로 간주됩니다. 서로 다른 코드 페이지의 데이터를 임포트할 때는 서로 다른 코드 페이지에 사용할 수 있는 PC/IXF 파일을 사용할 것을 권장합니다. PC/IXF 파일과 임포트 유틸리티가 동일한 코드 페이지에 있을 경우 일반 응용프로그램에 대해 서와 같은 처리가 발생합니다. 두 코드 페이지가 서로 다르고 **FORCEIN** 옵션이 지정된 경우, 임포트 유틸리티는 PC/IXF 파일의 데이터가 임포트를 수행 중인 응용프로그램과 동일한 코드 페이지를 가지고 있다고 간주합니다. 이는 두 코드 페이지에 대한 변환 테이블이 있는 경우에도 마찬가지입니다. 두 코드 페이지가 서로 다르고, **FORCEIN** 옵션이 지정되지 않았으며 변환 테이블이 있는 경우, PC/IXF 파일의 모든 데이터는 파일 코드 페이지에서 응용프로그램 코드 페이지로 변환됩니다. 두 코드 페이지가 서로 다르고 **FORCEIN** 옵션이 지정되지 않았으며 변환 테이블이 없는 경우 임포트 조작은 실패합니다. 이는 AIX 운영 체제의 DB2 클라이언트에 있는 PC/IXF 파일에만 적용됩니다.

1012 컬럼의 한계에 가까운 8KB 페이지에 있는 테이블 오브젝트의 경우, PC/IXF 데이터 파일을 임포트하면 DB2가 오류를 리턴할 수 있는데, 이는 SQL문의 최대 크기를 초과했기 때문입니다. 이러한 상황은 컬럼 유형이 CHAR, VARCHAR 또는 CLOB인 경우에만 발생할 수 있습니다. DEL 또는 ASC 파일을 임포트하는 데는 이러한 제한 사항이 적용되지 않습니다. PC/IXF 파일을 사용하여 새 테이블을 작성할 경우 다른 방법은 db2look을 사용하여 테이블을 작성한 DDL문을 덤프한 후 CLP를 통해 이 명령문을 발행하는 것입니다.

DB2 Connect은 OS/390용 DB2, VM 및 VSE용 DB2, OS/400용 DB2와 같은 DRDA 서버로 데이터를 임포트하는 데 사용할 수 있습니다. PC/IXF 임포트(**INSERT** 옵션)만 지원됩니다. **RESTARTCOUNT** 매개변수도 지원되지만 **COMMITCOUNT** 매개변수는 지원되지 않습니다.

유형이 지정된 테이블에서 **CREATE** 옵션을 사용할 때 PC/IXF 파일에 정의된 모든 서브테이블을 작성하십시오. 서브테이블 정의는 변경할 수 없습니다. 유형이 지정된 테이블에서 **CREATE** 이외의 옵션을 사용할 때, 트래버스 순서 목록을 통해 해당 옵션이 트래버스 순서를 지정하므로 트래버스 순서 목록은 익스포트 조작 시 사용된 옵션과



일치해야 합니다. PC/IXF 파일 형식의 경우, 옵션은 목표 서버테이블 이름만 지정해야 하며 파일에 저장된 트레이스 순서를 사용해야 합니다.

임포트 유틸리티를 사용하여 이전에 PC/IXF 파일로 익스포트된 테이블을 복구할 수 있습니다. 이 테이블은 익스포트될 때의 상태로 리턴합니다.

데이터는 시스템 테이블, 작성된 임시 테이블, 선언된 임시 테이블 또는 요약 테이블로 임포트될 수 없습니다.

뷰는 임포트 유틸리티를 통해 작성될 수 없습니다.

개별적 파트가 Windows 시스템에서 AIX 시스템으로 복사되는 다중 파트 PC/IXF 파일 임포트가 지원됩니다. 첫 번째 파일의 이름만 IMPORT 명령에 지정되어야 합니다. 예를 들어, IMPORT FROM data.ixf OF IXF INSERT INTO TABLE1. 파일 data.002, etc가 data.ixf와 동일한 디렉토리에서 사용 가능해야 합니다.

Windows 운영 체제에서:

- 논리적으로 분할된 PC/IXF 파일 임포트는 지원되지 않습니다.
- 잘못된 형식의 PC/IXF 또는 WSF 파일 임포트는 지원되지 않습니다.

내부 형식으로 된 보안 레이블에는 줄 바꾸기 문자가 포함될 수 있습니다. DEL 파일 형식을 사용하여 파일을 임포트할 경우 이들 줄 바꾸기 문자는 분리문자로 오인될 수 있습니다. 이러한 문제점이 발생하면 IMPORT 명령에 delprioritychar 파일 유형 수정자를 지정하여 분리문자에 대해 이전 디폴트 우선순위를 사용하십시오.

## 페더레이티드 고려사항

IMPORT 명령 및 INSERT, UPDATE 또는 INSERT\_UPDATE 명령 매개변수를 사용할 때 참여 중인 별칭에 대해 CONTROL 특권을 가지고 있는지 확인해야 합니다. 임포트 조작을 수행할 때 사용할 별칭이 이미 존재하는지 확인해야 합니다. 또한 IMPORT 명령 매개변수 섹션에서 표시된 바와 같이 여러 가지 제한사항을 유념해야 합니다.

ODBC와 같은 일부 데이터 소스는 별칭으로의 임포트를 지원하지 않습니다.

## 임포트 유틸리티의 파일 유형 수정자

표 12. 임포트 유틸리티의 유효한 파일 유형 수정자: 모든 파일 형식

수정자	설명
compound=x	<p>x는 1 - 100의 숫자입니다. nonatomic 복합 SQL을 사용하여 데이터를 삽입하며, x 명령문이 매번 시도됩니다.</p> <p>수정자가 지정되고 트랜잭션 로그가 충분히 크지 않으면, 임포트 조치가 실패합니다. <b>COMMITCOUNT</b>에서 지정한 행 수나, <b>COMMITCOUNT</b>가 지정되지 않은 경우 데이터 파일의 행 수를 수용할 정도로 트랜잭션 로그가 커야 합니다. 그러므로 트랜잭션 로그 오버플로우를 피하기 위해 <b>COMMITCOUNT</b> 옵션을 지정하도록 권장합니다.</p> <p>이 수정자는 <b>INSERT_UPDATE</b> 모드, 계층 테이블 및 다음 수정자와 호환되지 않습니다. <b>usedefaults</b>, <b>identitymissing</b>, <b>identityignore</b>, <b>generatedmissing</b> 및 <b>generatedignore</b>.</p>
generatedignore	<p>이 수정자는 생성된 모든 컬럼의 데이터가 데이터 파일에 표시되지만 무시해야 함을 임포트 유틸리티에 알립니다. 이로 인해 생성된 컬럼의 모든 값이 유틸리티에 의해 생성됩니다. 이 수정자는 <b>generatedmissing</b> 수정자와 함께 사용될 수 없습니다.</p>
generatedmissing	<p>이 수정자가 지정된 경우, 유틸리티는 입력 데이터 파일에는 생성된 컬럼의 데이터가 포함되지 않는 것으로(NULL 값도 비포함) 가정하므로 각 행의 값을 생성합니다. 이 수정자는 <b>generatedignore</b> 수정자와 함께 사용될 수 없습니다.</p>
identityignore	<p>이 수정자는 ID 컬럼의 데이터가 데이터 파일에 표시되지만 무시해야 함을 임포트 유틸리티에 알립니다. 이로 인해 유틸리티에 의해 모든 ID 값이 생성됩니다. <b>GENERATED ALWAYS</b> 및 <b>GENERATED BY DEFAULT ID</b> 컬럼 둘 다의 동작이 동일합니다. <b>GENERATED ALWAYS</b> 컬럼의 경우, 어떤 행도 거부되지 않음을 의미합니다. 이 수정자는 <b>identitymissing</b> 수정자와 함께 사용될 수 없습니다.</p>
identitymissing	<p>이 수정자가 지정된 경우, 유틸리티는 입력 데이터 파일에는 ID 컬럼의 데이터가 포함되지 않는 것으로(NULL 값도 비포함) 가정하므로 각 행의 값을 생성합니다. <b>GENERATED ALWAYS</b> 및 <b>GENERATED BY DEFAULT ID</b> 컬럼 둘 다의 동작이 동일합니다. 이 수정자는 <b>identityignore</b> 수정자와 함께 사용될 수 없습니다.</p>
lobsinfile	<p><i>lob-path</i>는 LOB 데이터를 포함하는 파일에 대한 경로를 지정합니다.</p> <p>각 경로는 LLS(Lob Location Specifier)에 의해 포인트되는 최소한 하나의 LOB를 포함하는 최소한 하나의 파일을 데이터 파일에 포함합니다. LLS는 LOB 파일 경로에 저장된 파일에서 LOB 위치의 문자열 표시입니다. LLS 형식은 <i>filename.ext.nnn.mmm</i>이며, 여기서 <i>filename.ext</i>는 LOB를 포함하는 파일의 이름이며, <i>nnn</i>은 파일 내에서 LOB의 오프셋을 바이트로 나타낸 것이며, <i>mmm</i>은 LOB의 길이를 바이트로 나타낸 것입니다. 예를 들어, 문자열 db2exp.001.123.456/가 데이터 파일에 저장되는 경우, LOB는 db2exp.001 파일에서 오프셋 123에 위치하며 456바이트 길이입니다.</p> <p><b>LOBS FROM</b> 절은 『lobsinfile』 수정자가 사용될 때 LOB 파일이 위치하는 곳을 지정합니다. <b>LOBS FROM</b> 절은 내재적으로 <b>LOBSINFILE</b> 동작을 활성화합니다. <b>LOBS FROM</b> 절은 데이터 임포트 중 LOB 파일을 검색하기 위해 경로 목록을 <b>IMPORT</b> 유틸리티로 전달합니다.</p> <p>널(NULL) LOB를 표시하려면, 크기를 -1로 입력하십시오. 크기가 0으로 지정되면, 길이가 0인 LOB로 처리됩니다. 길이가 -1인 널(NULL) LOBS의 경우, 오프셋 및 파일 이름은 무시됩니다. 예를 들어, 널(NULL) LOB의 LLS는 db2exp.001.7.-1/입니다.</p>
no_type_id	<p>단일 서브테이블로 임포트할 때에만 유효합니다. 일반 설치의 일반 테이블에서 데이터를 익스포트한 후 임포트 조작을 호출하여(이 수정자 사용) 데이터를 단일 서브테이블로 변환합니다.</p>

표 12. 임포트 유틸리티의 유효한 파일 유형 수정자: 모든 파일 형식 (계속)

수정자	설명
nodefaults	<p>목표 테이블 컬럼의 소스 컬럼이 명시적으로 지정되지 않고, 테이블 컬럼이 널(NULL) 입력 가능하지 않으면, 디폴트값이 로드되지 않습니다. 목표 테이블 컬럼 중 하나의 소스 컬럼이 명시적으로 지정되지 않은 경우, 이 옵션이 없으면 다음 중 하나가 발생합니다.</p> <ul style="list-style-type: none"> <li>• 컬럼의 디폴트값이 지정될 수 있는 경우, 디폴트값이 로드됩니다.</li> <li>• 컬럼이 널(NULL) 입력 가능하고 해당 컬럼의 디폴트값을 지정할 수 없는 경우, NULL이 로드됩니다.</li> <li>• 컬럼이 널(NULL) 입력 가능하고 디폴트값을 지정할 수 없는 경우, 오류가 리턴되며 유틸리티가 처리를 중지합니다.</li> </ul>
norowwarnings	<p>거부된 행에 대한 모든 경고를 제외시킵니다.</p>
rowchangetimestampignore	<p>이 수정자는 행 변경 시간소인 컬럼의 데이터가 데이터 파일에 표시되지만 무시해야 함을 임포트 유틸리티에 알립니다. 이로 인해 유틸리티에 의해 모든 ROW CHANGE TIMESTAMP가 생성됩니다. GENERATED ALWAYS 및 GENERATED BY DEFAULT 컬럼 둘다의 동작이 동일합니다. GENERATED ALWAYS 컬럼의 경우, 어떤 행도 거부되지 않음을 의미합니다. 이 수정자는 rowchangetimestampmissing 수정자와 함께 사용될 수 없습니다.</p>
rowchangetimestampmissing	<p>이 수정자가 지정된 경우, 유틸리티는 입력 데이터 파일에는 행 변경 시간소인 컬럼의 데이터를 포함하지 않음(NULL 값도 비포함)을 가정하므로 각 행의 값을 생성합니다. GENERATED ALWAYS 및 GENERATED BY DEFAULT 컬럼 둘 다의 동작이 동일합니다. 이 수정자는 rowchangetimestampignore 수정자와 함께 사용될 수 없습니다.</p>
seclabelchar	<p>입력 소스 파일의 보안 레이블이 디폴트로 인코딩된 숫자 형식이 아니라 보안 레이블 값의 문자열 형식임을 표시합니다. IMPORT는 각 보안 레이블을 로드된 대로의 내부 형식으로 변환합니다. 문자열이 적절한 형식으로 되어 있지 않은 경우 행은 로드되지 않으며 경고(SQLSTATE 01H53)가 리턴됩니다. 문자열이 테이블을 보호하는 보안 규정 파트인 유효한 보안 레이블을 나타내지 않는 경우, 행은 로드되지 않으며 경고(SQLSTATE 01H53, SQLCODE SQL3243W)가 리턴됩니다.</p> <p>seclabelname 수정자가 지정된 경우 이 수정자를 지정할 수 없으며, 그렇지 않으면 임포트에 실패하며 오류(SQLCODE SQL3525N)가 리턴됩니다.</p>
seclabelname	<p>입력 소스 파일의 보안 레이블이 디폴트로 인코딩된 숫자 형식이 아니라 이름으로 표시됨을 나타냅니다. IMPORT는 이름이 있는 경우 적절한 보안 레이블로 변환합니다. 테이블을 보호하는 보안 규정에 대해 표시된 이름이 있는 보안 레이블이 없는 경우, 행은 로드되지 않으며 경고(SQLSTATE 01H53, SQLCODE SQL3244W)가 리턴됩니다.</p> <p>seclabelchar 수정자가 지정된 경우 이 수정자를 지정할 수 없으며, 그렇지 않으면 임포트에 실패하며 오류(SQLCODE SQL3525N)가 리턴됩니다.</p> <p>주: 파일 유형이 ASC인 경우, 보안 레이블의 이름 다음에 오는 모든 스페이스는 이름의 일부로 해석됩니다. 이를 피하려면 striptblanks 파일 유형 수정자를 사용하여 스페이스가 제거되었는지 확인하십시오.</p>

표 12. 임포트 유틸리티의 유효한 파일 유형 수정자: 모든 파일 형식 (계속)

수정자	설명
usedefaults	<p>목표 테이블 컬럼의 소스 컬럼이 지정되었지만 하나 이상의 행 인스턴스에 대한 데이터를 포함하지 않은 경우, 디폴트값이 로드됩니다. 누락된 데이터의 예:</p> <ul style="list-style-type: none"> <li>• DEL 파일: 임의 수의 스페이스(" ", ",")로 구분되는 2개의 인접 컬럼 분리문자나 2개의 인접 컬럼 분리문자(",.")가 컬럼 값으로 지정됩니다.</li> <li>• DEL/ASC/WSF 파일: 행의 컬럼이 충분하지 않거나 행의 길이가 원래 스펙만큼 충분하지 않습니다.</li> </ul> <p>주: ASC 파일의 경우, NULL 컬럼 값은 명시적으로 누락된 것으로 간주되지 않으며 디폴트가 NULL 컬럼 값을 대신하지 않습니다. 숫자, 날짜, 시간 및 /시간소인 컬럼의 모든 공백 문자로 NULL 컬럼 값을 표시하거나, 컬럼이 NULL임을 표시하기 위해 모든 유형의 컬럼에 널(NULL) 표시기를 사용함으로써 NULL 컬럼 값을 표시합니다.</p> <p>이 옵션이 없는 경우, 소스 컬럼이 행 인스턴스의 데이터를 포함하지 않으면 다음 중 하나가 발생합니다.</p> <ul style="list-style-type: none"> <li>• DEL/ASC/WSF 파일: 컬럼에 널(NULL) 입력 가능한 경우 NULL이 로드됩니다. 컬럼이 널(NULL) 입력 가능하지 않으면 유틸리티가 행을 거부합니다.</li> </ul>

표 13. 임포트 유틸리티의 유효한 파일 유형 수정자: ASCII 파일 형식(ASC/DEL)

수정자	설명
codepage=x	<p>x는 ASCII 문자열입니다. 해당 값은 입력 데이터 세트에서 데이터의 코드 페이지로서 해석됩니다. 임포트 조작 중 이 코드 페이지로부터 응용프로그램 코드 페이지로 문자 데이터를 변환합니다.</p> <p>다음 규칙이 적용됩니다.</p> <ul style="list-style-type: none"> <li>• 순수 DBCS(그래픽), 혼합 DBCS 및 EUC의 경우, 분리문자는 x00 - x3F의 범위로 제한됩니다.</li> <li>• nullindchar은 x20 - x7F 코드 포인트의 표준 ASCII 세트에 포함된 기호를 지정해야 합니다. 이것은 ASCII 기호 및 코드 포인트를 나타냅니다.</li> </ul> <p>주:</p> <ol style="list-style-type: none"> <li>1. codepage 수정자는 lobsinfile 수정자와 함께 사용될 수 없습니다.</li> <li>2. 응용프로그램 코드 페이지에서 데이터베이스 코드 페이지로 코드 페이지를 변환할 때 데이터 확장이 발생하면, 데이터가 절단되고 데이터 유실이 발생할 수 있습니다.</li> </ol>

표 13. 임포트 유틸리티의 유효한 파일 유형 수정자: ASCII 파일 형식(ASC/DEL) (계속)

수정자	설명
dateformat="x"	<p>x는 소스 파일에서 날짜의 형식입니다.<sup>2</sup> 유효한 날짜 요소는 다음과 같습니다.</p> <p>YYYY - Year(0000 - 9999 범위의 4자리 숫자)  M - Month(1 - 12 범위의 1 또는 2자리 숫자)  MM - Month(1 - 12 범위의 2자리 숫자,  M과 상호 독점적)  D - Day(1 - 31 범위의 1 또는 2자리 숫자)  DD - Day(1 - 31 범위의 2자리 숫자,  D와 상호 독점적)  DDD - Day of the year(001 - 366 범위의 3자리 숫자,  다른 day 또는 month 요소와 상호 독점적)</p> <p>지정되지 않은 각 요소에 대해 디폴트값 1이 지정됩니다. 날짜 형식의 예:</p> <p>"D-M-YYYY"  "MM.DD.YYYY"  "YYYYDDD"</p>
implieddecimal	<p>내포된 소수점의 위치는 컬럼 정의로 판별되며, 값의 끝으로 가정되지 않습니다. 예를 들어, 12345 값은 12345.00이 아닌 123.45로 DECIMAL(8,2) 컬럼에 로드됩니다.</p>
timeformat="x"	<p>x는 소스 파일에서 시간의 형식입니다.<sup>2</sup> 유효한 시간 요소는 다음과 같습니다.</p> <p>H - Hour(12시간 시스템의 경우 0 - 12 범위의  1 또는 2자리 숫자,  24시간 시스템의 경우 0 - 24)  HH - Hour(12시간 시스템의 경우  0 - 12 범위의 2자리 숫자,  24시간 시스템의 경우 0 - 24.  H와 상호 독점적)  M - Minute(0 - 59 사이의 1 또는 2자리 숫자)  MM - Minute(0 - 59 사이의 2자리 숫자,  M과 상호 독점적)  S - Second(0 - 59 사이의 1 또는 2자리 숫자)  SS - Second(0 - 59 사이의 2자리 숫자,  S와 상호 독점적)  SSSSS - Second of the day after midnight(  00000 - 86399 사이의 5자리 숫자, 다른 시간 요소와  상호 독점적)  TT - Meridian indicator(AM 또는 PM)</p> <p>지정되지 않은 각 요소에 대해 디폴트값 0이 지정됩니다. 시간 형식의 예:</p> <p>"HH:MM:SS"  "HH.MM TT"  "SSSSS"</p>

표 13. 임포트 유틸리티의 유효한 파일 유형 수정자: ASCII 파일 형식(ASC/DEL) (계속)

수정자	설명
timestampformat="x"	<p>x는 소스 파일에서 시간소인의 형식입니다.<sup>2</sup> 유효한 시간소인 요소는 다음과 같습니다.</p> <p>YYYY - Year(0000 - 9999 사이의 4자리 숫자)  M - Month(1 - 12 사이의 1 또는 2자리 숫자)  MM - Month(01 - 12 사이의 2자리 숫자,  M 및 MM은 상호 독점적)  MMM - Month(월 이름으로 사용되는 3자리 문자로 대소문자를 구분하지 않으며,  M 및 MM은 상호 독점적)  D - Day(1 - 31 사이의 1 또는 2자리 숫자)  DD - Day(1 - 31 사이의 2자리 숫자, D와는 상호 독점적)  DDD - Day of the year(001 - 366 사이의 3자리 숫자,  다른 day 또는 month 요소와 상호 독점적)  H - Hour(12시간 시스템의 경우 0 - 12 범위의 1 또는 2자리 숫자,  24시간 시스템의 경우 0 - 24)  HH - Hour(12시간 시스템의 경우 0 - 12 범위의 2자리 숫자,  24시간 시스템의 경우 0 - 24,  H와는 상호 독점적)  M - Minute(0 - 59 사이의 1 또는 2자리 숫자)  MM - Minute(0 - 59 사이의 2자리 숫자,  M, minute와 상호 독점적)  S - Second(0 - 59 사이의 1 또는 2자리 숫자)  SS - Second(0 - 59 사이의 2자리 숫자,  S와 상호 독점적)  SSSSS - Second of the day after midnight(  00000 - 86399 사이의 5자리 숫자, 다른 시간 요소와 상호 독점적)  U(1 - 12회)  - Fractional seconds(U 어커런스 수는 각 자리가 0 - 9 사이의 자리 수를 나타냄)  TT - Meridian indicator(AM 또는 PM)</p> <p>다폴트값 1이 미지정된 YYYY, M, MM, D, DD 또는 DDD 요소에 지정됩니다. 다폴트값 'Jan'이 미지정된 MMM 요소에 지정됩니다. 미지정된 다른 모든 요소에 다폴트값 0이 지정됩니다. 다음은 시간소인 형식의 예입니다.</p> <p>"YYYY/MM/DD HH:MM:SS.UUUUUU"</p> <p>MMM 요소의 올바른 값은 다음을 포함합니다. 'jan', 'feb', 'mar', 'apr', 'may', 'jun', 'jul', 'aug', 'sep', 'oct', 'nov' 및 'dec'. 이들 값은 대소문자가 구분되지 않습니다.</p> <p>다음 예는 사용자 정의 날짜 및 시간 형식을 포함하는 데이터를 schedule이라는 테이블로 임포트하는 방법을 설명합니다.</p> <pre>db2 import from delfile2 of del modified by timestampformat="yyyy.mm.dd hh:mm tt" insert into schedule</pre>

표 13. импорт 유틸리티의 유효한 파일 유형 수정자: ASCII 파일 형식(ASC/DEL) (계속)

수정자	설명
usegraphiccodepage	<p>usegraphiccodepage가 제공되면, 그래픽 또는 2바이트 문자 대형 오브젝트(DBCLOB) 데이터 필드로 импорт되는 데이터는 그래픽 코드 페이지에 있음이 가정됩니다. 나머지 데이터는 문자 코드 페이지에 있다고 가정합니다. 그래픽 코드 페이지는 문자 코드 페이지와 연관됩니다. IMPORT는 codepage 수정자가 지정된 경우 이를 통해 문자 코드 페이지를 판별하고 codepage 수정자가 지정되지 않은 경우 응용프로그램의 코드 페이지를 통해 문자 코드 페이지를 판별합니다.</p> <p>복구 중인 테이블에 그래픽 데이터가 있는 경우에만 삭제(drop) 테이블 복구로 생성되는 구분된 데이터 파일과 결합하여 이 수정자를 사용해야 합니다.</p> <p><b>제한사항</b></p> <p>이들 파일이 오직 하나의 코드 페이지에 인코딩된 데이터를 포함하면, usegraphiccodepage 수정자는 EXPORT 유틸리티로 작성된 DEL 파일과 함께 지정되지 않아야 합니다. usegraphiccodepage 수정자는 파일의 2바이트 문자 대형 오브젝트(DBCLOB)에서 무시됩니다.</p>
xmlchar	<p>XML 문서가 문자 코드 페이지에서 인코딩됨을 지정합니다.</p> <p>이 옵션은 지정된 문자 코드 페이지에서 인코딩되었지만 인코딩 선언을 포함하지 않는 XML 문서 처리에 유용합니다.</p> <p>각 문서에서 선언 태그가 존재하고 인코딩 속성을 포함하는 경우 인코딩은 문자 코드 페이지와 일치해야 하며, 그렇지 않으면 문서를 포함하는 행이 거부됩니다. 문자 코드 페이지는 codepage 파일 유형 수정자에서 지정한 값이거나, 지정되지 않은 경우 응용프로그램 코드 페이지입니다. 디폴트로 문서가 유니코드로 인코딩되거나 인코딩 속성을 가진 선언 태그를 포함합니다.</p>
xmlgraphic	<p>XML 문서가 지정된 그래픽 코드 페이지에서 인코딩됨을 지정합니다.</p> <p>이 옵션은 특정 그래픽 코드 페이지에서 인코딩되었지만 인코딩 선언을 포함하지 않는 XML 문서 처리에 유용합니다.</p> <p>각 문서에서 선언 태그가 존재하고 인코딩 속성을 포함하는 경우 인코딩은 그래픽 코드 페이지와 일치해야 하며, 그렇지 않으면 문서를 포함하는 행이 거부됩니다. 그래픽 코드 페이지는 codepage 파일 유형 수정자에서 지정한 값의 그래픽 구성요소이거나, 지정되지 않은 경우 응용프로그램 코드 페이지의 그래픽 구성요소입니다. 디폴트로 문서가 유니코드로 인코딩되거나 인코딩 속성을 가진 선언 태그를 포함합니다.</p> <p>주: xmlgraphic 수정자가 IMPORT 명령으로 지정된 경우, импорт되는 XML 문서는 UTF-16 코드 페이지로 인코딩되어야 합니다. 그렇지 않으면, XML 문서는 구문 분석 오류로 거부되거나 데이터 손상이 있는 테이블로 импорт될 수 있습니다.</p>

표 14. импорт 유틸리티의 유효한 파일 유형 수정자: ASC(컬럼 식별자가 없는 ASCII) 파일 형식

수정자	설명
nochecklengths	<p>nochecklengths가 지정되면, 목표 테이블 컬럼의 크기를 초과하는 컬럼 정의가 소스 데이터에 있는 경우에도 각 행을 импорт하려고 시도합니다. 코드 페이지 변환으로 소스 데이터가 축소되는 경우 그러한 행은 정상적으로 импорт될 수 있습니다. 예를 들어, 소스에서 4바이트 EUC 데이터는 목표에서 2바이트 DBCS 데이터로 축소되어 반 정도의 스페이스가 필요합니다. 불일치 컬럼 정의에도 불구하고 소스 데이터가 모든 경우에 맞는 경우 특히 이 옵션이 유용합니다.</p>

표 14. 임포트 유틸리티의 유효한 파일 유형 수정자: ASC(컬럼 식별자가 없는 ASCII) 파일 형식 (계속)

수정자	설명
nullindchar=x	x는 단일 문자입니다. 널(NULL) 값을 나타내는 문자를 x로 변경합니다. x의 디폴트값은 Y입니다. <sup>3</sup>  문자가 영문자인 경우를 제외하고 EBCDIC 데이터 파일의 경우 이 수정자의 대소문자를 구분합니다. 예를 들어, 널(NULL) 표시기 문자가 N 문자가 되도록 지정되는 경우, n은 널(NULL) 표시기로 인식됩니다.
reclen=x	x는 최대값이 32,767인 정수입니다. 각 행에 대해 x 문자가 읽히지며 행의 끝을 표시하기 위한 줄 바꾸기 문자는 사용되지 않습니다.
striptblanks	데이터를 변수 길이 필드로 로드할 때 뒤 공백을 절단합니다. 이 옵션이 지정되지 않으면, 공백이 보존됩니다.  다음 예에서, striptblanks를 사용하면 임포트 유틸리티가 뒤 공백을 절단합니다. <pre>db2 import from myfile.asc of asc modified by striptblanks method 1 (1 10, 12 15) messages msgs.txt insert into staff</pre> 이 옵션은 striptnulls와 함께 지정될 수 없습니다. 이들은 상호 독점 옵션입니다. 이 옵션은 이전 호환성을 위해서만 지원되는 사용 안하는 t 옵션을 교체합니다.
striptnulls	데이터를 변수 길이 필드로 로드할 때 뒤 NULL 값(0x00 문자)을 절단합니다. 이 옵션이 지정되지 않으면, NULL 값이 보존됩니다.  이 옵션은 striptblanks와 함께 지정될 수 없습니다. 이들은 상호 독점 옵션입니다. 이 옵션은 이전 호환성을 위해서만 지원되는 사용 안하는 padwithzero 옵션을 교체합니다.

표 15. 임포트 유틸리티의 유효한 파일 유형 수정자: DEL(컬럼 식별자가 있는 ASCII) 파일 형식

수정자	설명
chardelx	x는 단일 문자열 분리문자입니다. 디폴트값은 큰따옴표(")입니다. 큰따옴표 대신 지정된 문자를 사용하여 문자열을 묶습니다. <sup>34</sup> 명시적으로 큰따옴표를 문자열 분리문자로 지정하려는 경우 다음과 같이 지정해야 합니다. <pre>modified by charde1"</pre> 작은따옴표(')를 문자열 분리문자로 지정할 수도 있습니다. 다음 예에서, charde1''는 임포트 유틸리티가 작은따옴표를 문자열 분리문자로 해석하게 합니다. <pre>db2 "import from myfile.del of del modified by charde1'" method p (1, 4) insert into staff (id, years)"</pre>
coldelx	x는 단일 문자 컬럼 분리문자입니다. 디폴트값은 쉼표(,)입니다. 지정된 문자가 쉼표 대신 사용되어 컬럼 끝을 나타냅니다. <sup>34</sup>  다음 예에서, colde1;은 임포트 유틸리티가 세미콜론을 컬럼 분리문자로 해석하게 합니다. <pre>db2 import from myfile.del of del modified by colde1; messages msgs.txt insert into staff</pre>
decplusblank	플러스 부호 문자. 플러스 부호(+) 대신 공백을 양의 10진수 값 앞에 붙입니다. 디폴트 조치는 플러스 부호를 양의 10진수 값 앞에 붙이는 것입니다.



표 15. 임포트 유틸리티의 유효한 파일 유형 수정자: DEL(컬럼 식별자가 있는 ASCII) 파일 형식 (계속)

수정자	설명
decptx	<p>x는 소수점 문자로 마침표를 대신하는 단일 문자입니다. 디폴트값은 마침표(.)입니다. 마침표 대신 지정된 문자가 소수점 문자로 사용됩니다.<sup>34</sup></p> <p>다음 예에서, decpt;는 임포트 유틸리티가 세미콜론(; )을 소수점으로 해석하게 합니다.</p> <pre>db2 "import from myfile.del of del       modified by chardel'"       decpt; messages msgs.txt insert into staff"</pre>
delprioritychar	<p>분리문자의 현재 디폴트 우선순위: 레코드 구분 문자, 문자 분리문자, 컬럼 분리문자. 이 수정자는 분리문자 우선순위를 문자 분리문자, 레코드 구분 문자, 컬럼 분리문자로 되돌림으로써 이전 우선순위에 따른 기존 응용프로그램을 보호합니다. 구문:</p> <pre>db2 import ... modified by delprioritychar ...</pre> <p>예를 들어, 다음과 같은 DEL 데이터 파일이 있습니다.</p> <pre>"Smith, Joshua",4000,34.98&lt;row delimiter&gt; "Vincent,&lt;row delimiter&gt;, is a manager", ... ... 4005,44.37&lt;row delimiter&gt;</pre> <p>delprioritychar 수정자를 지정하고, 이 데이터 파일에는 두 개의 행만이 있습니다. 두 번째 &lt;행 분리문자&gt;는 두 번째 행의 첫 번째 데이터 컬럼의 파트로 해석되지만, 첫 번째 및 세 번째 &lt;행 분리문자&gt;는 실제 레코드 구분 문자로 해석됩니다. 이 수정자가 지정되지 않은 경우, 이 데이터 파일에는 세 개의 행이 있으며, 각 행은 &lt;행 분리문자&gt;로 구분됩니다.</p>
keepblanks	<p>유형 CHAR, VARCHAR, LONG VARCHAR 또는 CLOB의 각 필드에 앞뒤 공백을 둡니다. 이 옵션이 없으면, 문자 분리문자 내에 없는 앞 공백과 뒤 공백은 모두 제거되며 공백 필드의 테이블로 NULL이 삽입됩니다.</p>
nochardel	<p>임포트 유틸리티는 컬럼 분리문자 간의 모든 바이트를 컬럼 데이터의 파트로 가정합니다. 문자 분리문자는 컬럼 데이터의 파트로 구문 분석됩니다. DB2를 사용하여 데이터를 익스포트한 경우(익스포트 시 nochardel을 지정한 경우를 제외하고) 이 옵션을 지정하지 말아야 합니다. 이 옵션은 문자 분리문자가 없는 벡터 데이터 파일을 지원하기 위해 제공됩니다. 부적절한 사용은 데이터 유실이나 손상을 초래할 수 있습니다.</p> <p>이 옵션은 chardelx, delprioritychar 또는 nodoubledel과 함께 지정될 수 없습니다. 이들은 상호 독점 옵션입니다.</p>
nodoubledel	<p>2바이트 분리문자를 인식하지 않습니다.</p>

표 16. 임포트 유틸리티의 유효한 파일 유형 수정자: IXF 파일 형식

수정자	설명
forcein	<p>코드 페이지 불일치와 관계없이 데이터를 승인하고 코드 페이지 간에 변환하지 않도록 유틸리티에 지시합니다.</p> <p>데이터의 고정 길이 대상 필드가 충분히 큰지 검증하도록 고정 길이 대상 필드를 점검합니다. nochecklengths가 지정된 경우, 검사가 수행되지 않으며 각 행을 임포트하려고 시도합니다.</p>
indexixf	<p>기존 테이블에 현재 정의된 모든 인덱스를 삭제(drop)하고 PC/IXF 파일의 인덱스 정의에서 새로운 인덱스를 작성하도록 유틸리티에 지시합니다. 이 옵션은 목차가 교체될 때에만 사용될 수 있습니다. 뷰 또는 insert-column과 함께 사용될 수 없습니다.</p>
indexschema= schema	<p>색인 작성 중 인덱스 이름에 지정된 schema를 사용합니다. schema가 지정되지 않은 경우 (그러나 키워드 indexschema가 지정됨), 연결 사용자 ID를 사용합니다. 해당 키워드가 지정되지 않은 경우, IXF 파일에서 스키마를 사용합니다.</p>

표 16. импорт 유틸리티의 유효한 파일 유형 수정자: IXF 파일 형식 (계속)

수정자	설명
nochecklengths	nochecklengths가 지정되면, 목표 테이블 컬럼의 크기를 초과하는 컬럼 정의가 소스 데이터에 있는 경우에도 각 행을 импорт하려고 시도합니다. 코드 페이지 변환으로 소스 데이터가 축소되는 경우 그러한 행은 정상적으로 импорт될 수 있습니다. 예를 들어, 소스에서 4바이트 EUC 데이터는 목표에서 2바이트 DBCS 데이터로 축소되어 반 정도의 스페이스가 필요합니다. 불일치 컬럼 정의에도 불구하고 소스 데이터가 모든 경우에 맞는 경우 특히 이 옵션이 유용합니다.
forcecreate	인포트 조작 중 SQL3311N 리턴 후에 제한된 정보나 가능한 누락 정보로 테이블이 작성되어야 함을 지정합니다.

표 17. codepage 및 usegraphiccodepage 사용 시 IMPORT 동작

codepage=N	usegraphiccodepage	IMPORT 동작
Absent	Absent	파일의 모든 데이터는 응용프로그램 코드 페이지에 있는 것으로 가정됩니다.
Present	Absent	파일의 모든 데이터는 코드 페이지 N에 있는 것으로 가정됩니다.  경고: N이 1바이트 코드 페이지이면 데이터베이스로 임포트될 때 그래픽 데이터가 손상됩니다.
Absent	Present	파일의 문자 데이터는 응용프로그램 코드 페이지에 있는 것으로 가정됩니다. 그래픽 데이터는 응용프로그램 그래픽 데이터의 코드 페이지에 있는 것으로 가정됩니다.  응용프로그램 코드 페이지가 1바이트이면, 모든 데이터는 응용프로그램 코드 페이지에 있는 것으로 가정됩니다.  경고: 응용프로그램 코드 페이지가 1바이트이면, 데이터베이스가 그래픽 컬럼을 포함해도 데이터베이스로 임포트될 때 그래픽 데이터가 손상됩니다.
Present	Present	문자 데이터는 코드 페이지 N에 있는 것으로 가정됩니다. 그래픽 데이터는 N의 그래픽 코드 페이지에 있는 것으로 가정됩니다.  N이 1바이트 또는 2바이트 코드 페이지인 경우, 모든 데이터는 코드 페이지 N에 있는 것으로 가정됩니다.  경고: N이 1바이트 코드 페이지이면 데이터베이스로 임포트될 때 그래픽 데이터가 손상됩니다.

주:

1. **MODIFIED BY** 옵션과 함께 지원되지 않는 파일 유형을 사용하려고 시도하는 경우 импорт 유틸리티는 경고를 발행하지 않습니다. 이런 경우, импорт 조작에 실패하며 오류 코드가 리턴됩니다.
2. 날짜 출력 문자열을 둘러싼 큰따옴표는 필수입니다. 필드 구분자는 a - z, A - Z 및 0 - 9를 포함할 수 없습니다. 필드 구분자는 DEL 파일 형식의 필드 분리문자나 문자 분리문자와 같지 않아야 합니다. 요소의 시작 및 종료 위치가 명확한 경

우 필드 구분자가 선택적입니다. D, H, M 또는 S와 같은 요소가 사용되는 경우 (수정자에 따라) 항목의 변수 길이 때문에 모호함이 있을 수 있습니다.

시간소인 형식의 경우, 문자 M을 사용하는 month 및 minute 디스크립터 간의 모호함을 피하려면 주의해야 합니다. month 필드는 다른 날짜 필드와 인접해야 합니다. minute 필드는 다른 시간 필드와 인접해야 합니다. 다음은 앰비규어스 시간소인 형식입니다.

```
"M" (month 또는 minute일 수 있음)
"M:M" (month 및 minute 구분 가능?)
"M:YYYY:M" (둘 다 month로 해석됨)
"S:M:YYYY" (둘 다 시간 값 및 날짜 값에 인접)
```

앰비규어스 경우, 유틸리티는 오류 메시지를 발행하며, 조작에 실패합니다.

다음은 명확한 시간소인 형식입니다.

```
"M:YYYY" (Month)
"S:M" (Minute)
"M:YYYY:S:M" (Month....Minute)
"M:H:YYYY:M:D" (Minute....Month)
```

큰따옴표 및 백슬래시와 같은 일부 문자는 Escape 문자(예: #)가 앞에 와야 합니다.

3. chardel, coldel 또는 decpt 파일 유형에 제공되는 문자 값은 소스 데이터의 코드 페이지에 지정되어야 합니다.

문자 코드 포인트(문자 기호 대신)는 구문 xJJ 또는 0xJJ를 사용하여 지정될 수 있으며, 여기서 JJ는 코드 포인트의 16진수를 나타냅니다. 예를 들어, 컬럼 분리문자로 # 문자를 지정하려면, 다음 중 하나를 사용하십시오.

```
... modified by coldel# ...
... modified by coldel0x23 ...
... modified by coldelX23 ...
```

4. 데이터 이동을 위한 분리문자 고려사항은 분리문자를 겹쳐쓰기할 때 사용할 수 있는 문자에 적용하는 제한사항을 나열합니다.
5. 별칭으로 импорт할 때 다음 파일 유형 수정자는 허용되지 않습니다.

- indexixf
- indexschema
- dldel filetype
- nodefaults
- usedefaults
- no\_type\_id filetype
- generatedignore
- generatedmissing

- identityignore
  - identitymissing
  - lobsinfile
6. **WSF** 파일 형식은 XML 컬럼에서 지원되지 않습니다. 또한 이 파일 형식에 대한 지원은 사용되지 않으므로 추후 릴리스에서 제거될 수 있습니다. 지원이 제거되기 전에 WSF 파일 대신 지원되는 파일 형식을 사용하여 시작하는 것이 좋습니다.
  7. **CREATE** 모드는 XML 컬럼에서 지원되지 않습니다.
  8. 모든 XML 데이터는 주 데이터 파일에서 분리된 XML 파일에 있어야 합니다. 주 데이터 파일의 각 XML 컬럼에 대해 XDS(XML Data Specifier)(또는 NULL 값)가 있어야 합니다.
  9. XMLCHAR 또는 XMLGRAPHIC 파일 유형 수정자가 지정된 경우를 제외하고, XML 문서는 유니코드 형식으로 되어 있거나 인코딩 속성을 포함하는 선언 태그를 포함한다고 가정합니다.
  10. 잘 양식화되지 않은 문서를 포함하는 행은 거부됩니다.
  11. **XMLVALIDATE** 옵션이 지정된 경우, 일치하는 스키마에 대해 정상적으로 유효성을 확인하는 문서는 삽입될 때 스키마 정보로 주석을 표시합니다. 일치하는 스키마에 대해 유효성을 확인하는 데 실패하는 문서를 포함하는 행은 거부됩니다. 정상적으로 유효성을 확인하려면, 임포트를 호출하는 사용자가 보유한 특권이 최소한 다음 중 하나를 포함해야 합니다.
    - DBADM 권한
    - 유효성 확인에 사용되는 XML 스키마의 USAGE 특권
  12. 내재적으로 숨겨진 행 변경 시간소인 컬럼을 포함하는 테이블로 임포트할 때, 내재적으로 숨겨진 컬럼의 등록 정보는 무시됩니다. 그러므로, 컬럼의 데이터가 임포트되는 데이터에 표시되지 않으며 명시적 컬럼 목록이 표시되지 않으면 rowchangetimestampmissing 파일 유형 수정자가 임포트 명령에 지정되어야 합니다.

---

## 제 66 장 INITIALIZE TAPE

백업 및 리스토어 작업을 위한 테이프를 스트리밍 테이프 디바이스로 초기화합니다. 이 명령은 Windows 운영 체제에서만 지원됩니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*

### 필수 연결

없음

### 명령 구문

```
▶▶—INITIALIZE TAPE—┬──ON— device—┬──USING— blksize—┬──▶▶
```

### 명령 매개변수

#### ON *device*

유효한 테이프 디바이스 이름을 지정합니다. 디폴트값은 `WWW.WTAPE0`입니다.

#### USING *blksize*

디바이스에 대한 블록 크기를 바이트 단위로 지정합니다. 지정된 블록 크기 값이 디바이스에 대한 블록 크기의 지원 범위에 해당될 경우 디바이스는 이 지정된 블록 크기를 사용하도록 초기화됩니다.

BACKUP DATABASE 명령 및 RESTORE DATABASE에 대해 지정된 버퍼 크기는 여기에 지정된 블록 크기로 나눌 수 있어야 합니다.

이 매개변수의 값을 지정하지 않으면 디바이스는 해당 디폴트 블록 크기를 사용하도록 초기화됩니다. 0값을 지정하면 디바이스는 가변 길이 블록 크기를 사용하도록 초기화되고, 디바이스가 가변 길이 블록 코드를 지원하지 않을 경우 오류가 리턴됩니다.

테이프에 백업시 가변 블록 크기의 사용은 현재 지원되지 않습니다. 이 옵션을 사용해야 하는 경우, 가변 블록 크기로 작성된 백업 이미지를 사용하여 성공적으로 복구할 수 있는 제대로 테스트된 프로시저가 있는지 확인하십시오.

가변 블록 크기 사용시 사용 중인 테이프 디바이스에 대해 최대 한계 이하의 백업 버퍼 크기를 지정해야 합니다. 최적 성능을 위해서는 버퍼 크기를 사용 중인 디바이스의 최대 블록 크기 한계와 동일하게 지정해야 합니다.

## 제 67 장 INSPECT

구조적인 무결성을 위해 데이터베이스를 검사하고 페이지 일관성을 위해 데이터베이스 페이지를 점검합니다. INSPECT 명령은 테이블 오브젝트의 구조와 테이블 스페이스의 구조가 유효한지 점검합니다. 상호 오브젝트 유효성 확인은 온라인 인덱스 대 데이터 일관성 검사를 수행합니다.

### 범위

단일 파티션 데이터베이스 환경에서 범위에는 단일 파티션만 포함됩니다. 파티션된 데이터베이스 환경에서 이는 db2nodes.cfg에 정의된 모든 논리적 파티션의 컬렉션입니다. 파티션된 테이블의 경우, CHECK DATABASE 및 CHECK TABLESPACE 옵션에는 각 데이터 파티션 및 파티션되지 않은 인덱스가 포함됩니다. CHECK TABLE 옵션은 파티션된 테이블에 대해서도 사용 가능하지만, 단일 데이터 파티션 또는 인덱스를 검사하기 보다는 하나의 테이블에 있는 모든 데이터 파티션 및 인덱스를 검사합니다.

### 권한 부여

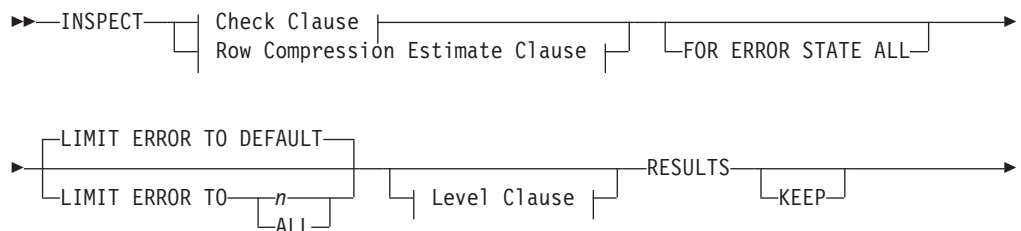
INSPECT CHECK의 경우 다음 중 하나이어야 합니다.

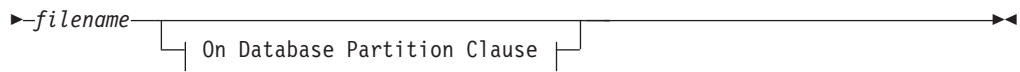
- *sysadm*
- *dbadm*
- *sysctrl*
- *sysmaint*
- 단일 테이블의 경우 CONTROL 특권

### 필요한 연결

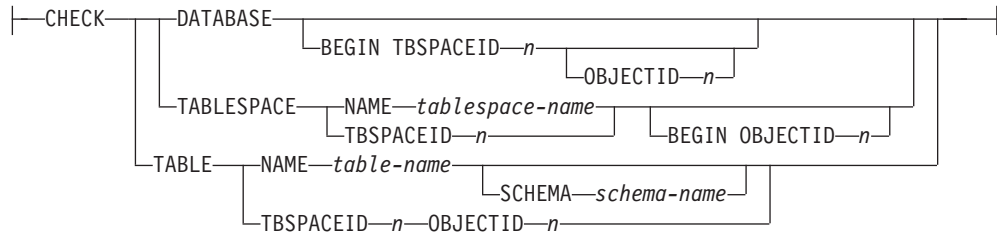
데이터베이스

### 명령 구문

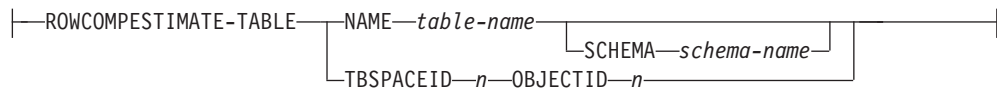




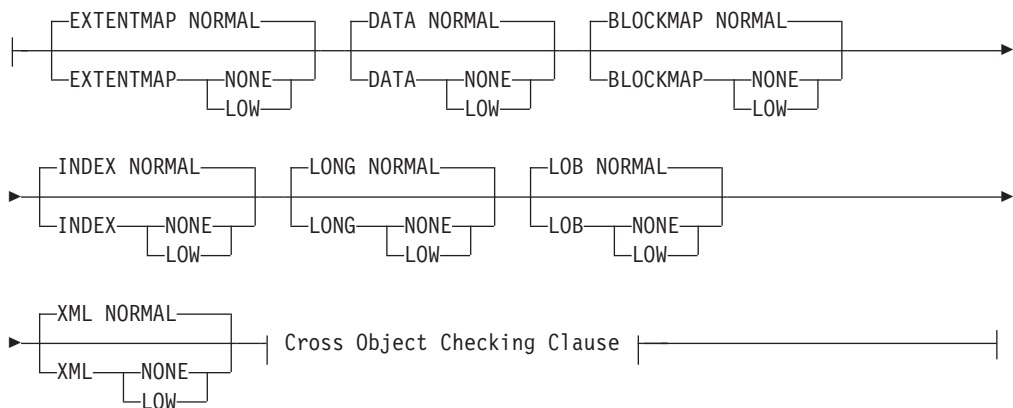
**Check Clause:**



**Row Compression Estimate Clause:**



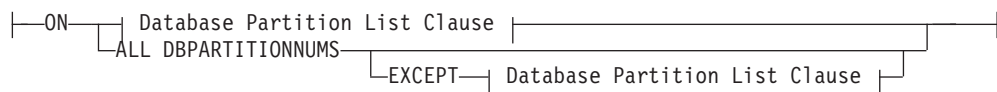
**Level Clause:**



**Cross Object Checking Clause:**

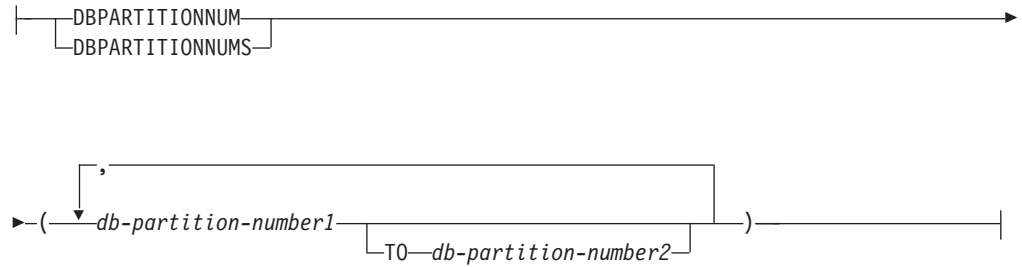


**On Database Partition Clause:**





### Database Partition List Clause:



### 명령 매개변수

#### CHECK

점검 처리를 지정합니다.

#### DATABASE

전체 데이터베이스를 지정합니다.

#### BEGIN TBSPACEID *n*

제공된 테이블 스페이스 ID 번호를 가지고 있는 테이블 스페이스에서 처리가 시작되도록 지정합니다.

#### OBJECTID *n*

제공된 테이블 스페이스 ID 번호 및 오브젝트 ID 번호를 가지고 있는 테이블 스페이스에서 처리가 시작되도록 지정합니다.

#### TABLESPACE

##### NAME *tablespace-name*

제공된 테이블 스페이스 이름을 가지고 있는 단일 테이블 스페이스를 지정합니다.

##### TBSPACEID *n*

제공된 테이블 스페이스 ID 번호를 가지고 있는 단일 테이블 스페이스를 지정합니다.

##### BEGIN OBJECTID *n*

제공된 오브젝트 ID 번호를 가지고 있는 테이블에서 처리가 시작되도록 지정합니다.

#### TABLE

##### NAME *table-name*

제공된 테이블 이름을 가지고 있는 테이블을 지정합니다.

##### SCHEMA *schema-name*

단일 테이블 조작에 대한 지정된 테이블 이름의 스키마 이름을 지정합니다.

## TBSPACEID *n* OBJECTID *n*

제공된 테이블 스페이스 ID 번호 및 오브젝트 ID 번호를 가지고 있는 테이블을 지정합니다.

## ROWCOMPESTIMATE

테이블에 대한 행 압축의 효과를 추정합니다. 또한 이 조작을 수행할 데이터베이스 파티션을 지정할 수도 있습니다.

이 도구는 테이블 데이터의 샘플을 취하고 그 샘플에서 사전을 빌드할 수 있습니다. 그러면 이 사전을 사용하여 샘플에 포함된 레코드에 대해 압축을 테스트할 수 있습니다. 이 테스트 압축에서, 다음과 같이 추정되는 데이터가 수집됩니다.

- 압축을 통해 절약되는 바이트의 백분율
- 압축을 통해 절약되는 페이지의 백분율
- 압축 사전 크기
- 확장 사전 크기

INSPECT는 테이블에 대해 COMPRESS YES 속성이 설정되고 테이블에 대해 아직 사전이 존재하지 않은 경우 압축 추정 값을 수집하기 위해 빌드된 사전을 삽입합니다. INSPECT는 테이블에 액세스하는 다른 응용프로그램에 동시에 사전을 삽입하려고 합니다. 사전을 삽입하려면 독점 테이블 변경 잠금 및 독점 테이블에 대한 의도 잠금이 필요합니다. INSPECT는 데이터 행 압축을 지원하는 테이블에 사전을 삽입하지만 합니다. 파티션된 테이블의 경우 각 파티션에서 별도의 사전이 빌드되어 삽입됩니다.

테이블 행 데이터를 샘플링하고 테이블에 대한 압축 사전을 빌드할 때 INSPECT 명령은 테이블 오브젝트에서 테이블 행 데이터만 지원합니다. 테이블에 XML 컬럼이 포함된 경우, 데이터는 샘플링되지 않고 테이블의 XML 스토리지 오브젝트에서 XML 데이터에 대해 압축 사전이 빌드되지 않습니다.

ROWCOMPESTIMATE 옵션은 인덱스 압축 추정을 제공하지 않습니다. 대신 테이블 함수를 사용하십시오.

## RESULTS

결과 출력 파일을 지정합니다. 파일은 진단 데이터 디렉토리 경로에서 기록됩니다. 점검 처리 중에 오류가 발견되지 않은 경우 결과 출력 파일은 INSPECT 조작 종료 시 지워집니다. 점검 처리 중에 오류가 발견되면 결과 출력 파일은 INSPECT 조작 종료 시 지워지지 않습니다.

**KEEP** 항상 결과 출력 파일을 보존할 것을 지정합니다.

*file-name*

결과 출력 파일의 이름을 지정합니다. 파일은 진단 데이터 디렉토리 경로에서 작성해야 합니다.

## **ALL DBPARTITIONNUMS**

db2nodes.cfg 파일에 지정된 모든 데이터베이스 파티션에서 조작이 수행될 것을 지정합니다. 이는 노드 절이 지정되지 않는 경우 디폴트입니다.

## **EXCEPT**

노드 목록에서 지정된 것을 제외하고 db2nodes.cfg 파일에서 지정되는 모든 데이터베이스 파티션에서 조작이 수행될 것을 지정합니다.

## **ON DBPARTITIONNUM | ON DBPARTITIONNUMS**

데이터베이스 파티션 세트에 대해 조작을 수행합니다.

*db-partition-number1*

데이터베이스 파티션 목록의 데이터베이스 파티션 번호를 지정합니다.

*db-partition-number2*

*db-partition-number2*를 포함하는 *db-partition-number1*의 모든 데이터베이스 파티션이 데이터베이스 파티션 목록에 포함되도록 두 번째 데이터베이스 파티션 번호를 지정합니다.

## **FOR ERROR STATE ALL**

내부 상태가 이미 오류 상태를 표시하는 테이블 오브젝트의 경우, 점검은 단지 이 상태를 보고하고 오브젝트를 통해 스캔하지 않습니다. 이 옵션을 지정하면 내부 상태가 이미 오류 상태를 나열하는 경우에도 오브젝트를 통한 스캔 처리가 수반됩니다.

INDEXDATA 옵션을 같이 사용하는 경우 인덱스나 데이터 오브젝트가 오류 상태인 동안 온라인 인덱스 대 데이터 일관성 검사가 수행되지 않습니다.

## **LIMIT ERROR TO *n***

오브젝트에 대해 오류 상태에 있는 페이지 중 보고가 제한되는 페이지 수. 오브젝트에 대해 오류 상태에 있는 이 페이지 수 한계에 도달하면, 나머지 오브젝트에 대한 점검 처리가 중지됩니다.

INDEXDATA 옵션을 같이 사용하는 경우 *n*은 온라인 인덱스 대 데이터 일관성 검사가 수행되는 동안 보고가 제한되는 오류 수를 나타냅니다.

## **LIMIT ERROR TO DEFAULT**

오브젝트에 대한 오류 보고를 제한할 디폴트 페이지 수. 이 값은 오브젝트의 Extent 크기입니다. 이 매개변수는 디폴트입니다.

INDEXDATA 옵션을 같이 사용하는 경우 DEFAULT는 온라인 인덱스 대 데이터 일관성 검사가 수행되는 동안 보고가 제한되는 디폴트 오류 수를 나타냅니다.

## **LIMIT ERROR TO ALL**

보고되는 오류 페이지 수에 대해 제한이 없습니다.

INDEXDATA 옵션을 같이 사용하는 경우 ALL은 온라인 인덱스 대 데이터 일관성 검사가 수행되는 동안 보고되는 오류 수에 대한 제한이 없음을 나타냅니다.

## **EXTENTMAP**

### **NORMAL**

처리 레벨이 Extent Map에 대해 일반적임을 지정합니다. 디폴트.

### **NONE**

처리 레벨이 Extent Map에 대해 없음을 지정합니다.

**LOW** 처리 레벨이 Extent Map에 대해 낮음을 지정합니다.

## **DATA**

### **NORMAL**

처리 레벨이 데이터 오브젝트에 대해 일반적임을 지정합니다. 디폴트.

### **NONE**

처리 레벨이 데이터 오브젝트에 대해 없음을 지정합니다.

**LOW** 처리 레벨이 데이터 오브젝트에 대해 낮음을 지정합니다.

## **BLOCKMAP**

### **NORMAL**

처리 레벨이 블록 맵 오브젝트에 대해 일반적임을 지정합니다. 디폴트.

### **NONE**

처리 레벨이 블록 맵 오브젝트에 대해 없음을 지정합니다.

**LOW** 처리 레벨이 블록 맵 오브젝트에 대해 낮음을 지정합니다.

## **INDEX**

### **NORMAL**

처리 레벨이 인덱스 오브젝트에 대해 일반적임을 지정합니다. 디폴트.

### **NONE**

처리 레벨이 인덱스 오브젝트에 대해 없음을 지정합니다.

**LOW** 처리 레벨이 인덱스 오브젝트에 대해 낮음을 지정합니다.

## **LONG**

### **NORMAL**

처리 레벨이 Long 오브젝트에 대해 일반적임을 지정합니다. 디폴트.

### **NONE**

처리 레벨이 Long 오브젝트에 대해 없음을 지정합니다.

**LOW** 처리 레벨이 Long 오브젝트에 대해 낮음을 지정합니다.

## LOB

### NORMAL

처리 레벨이 LOB 오브젝트에 대해 일반적임을 지정합니다. 디폴트.

### NONE

처리 레벨이 LOB 오브젝트에 대해 없음을 지정합니다.

**LOW** 처리 레벨이 LOB 오브젝트에 대해 낮음을 지정합니다.

## XML

### NORMAL

처리 레벨이 XML 컬럼 오브젝트에 대해 일반적임을 지정합니다. 디폴트. 대부분의 불일치에 대해 XML 오브젝트 페이지를 점검합니다. 실제 XML 데이터는 검사하지 않습니다.

### NONE

처리 레벨이 XML 컬럼 오브젝트에 대해 없음을 지정합니다. XML 오브젝트는 전혀 검사하지 않습니다.

**LOW** 처리 레벨이 XML 컬럼 오브젝트에 대해 낮음을 지정합니다. 일부 불일치에 대해 XML 오브젝트 페이지를 점검합니다. 실제 XML 데이터는 검사하지 않습니다.

## INDEXDATA

인덱스 대 데이터 일관성 검사를 수행하기 위해 지정됩니다. INDEXDATA 검사는 디폴트로 수행되지 않습니다.

## 예:

- 현재 검사된 오브젝트를 포함하여 모든 오브젝트에 대한 읽기/쓰기 액세스를 허용하는 인덱스 대 데이터 일관성 검사를 수행하려면 다음 명령을 발행하십시오.

```
inspect check table name fea3 indexdata results keep fea3high.out
```

- 현재 검사 중인 오브젝트를 포함하여, 모든 오브젝트에 대한 읽기 또는 쓰기 액세스를 허용하는 인덱스 대 데이터 일관성 검사를 수행하려면 다음 명령을 발행하십시오.

```
INSPECT CHECK TABLE NAME car SCHEMA vps INDEXDATA RESULTS KEEP table1.out
```

- 테이블 EMPLOYEE의 데이터가 압축되는 경우 절약되는 스토리지 스페이스 양을 추정하려면 다음 명령을 발행하십시오.

```
INSPECT ROWCOMPESTIMATE TABLE NAME car SCHEMA vps RESULTS table2.out
```

## 사용법 참고

1. 테이블 오브젝트에 대한 CHECK 조작의 경우, 오브젝트에 대해 처리 레벨을 지정할 수 있습니다. 디폴트는 NORMAL 레벨이며, 오브젝트에 대해 NONE을 지정하면 그 오브젝트는 제외됩니다. LOW를 지정하면 NORMAL에 대해 수행된 점검의 서브세트가 수행됩니다.

2. 테이블 스페이스나 테이블을 식별하기 위한 ID 값을 지정하여 특정 테이블 스페이스에서 또는 특정 테이블에서 시작하도록 CHECK DATABASE 옵션을 지정할 수 있습니다.
3. 테이블을 식별하기 위한 ID 값을 지정하여 특정 테이블에서 시작하도록 CHECK TABLESPACE 옵션을 지정할 수 있습니다.
4. 테이블 스페이스 처리는 테이블 스페이스에 있는 오브젝트에만 영향을 줍니다. INDEXDATA 옵션을 사용하는 경우에는 예외입니다. INDEXDATA는 인덱스 오브젝트가 테이블 스페이스에 있는 한 인덱스 대 데이터 일관성을 검사합니다. 이는 다음을 의미합니다.
  - 인덱스 오브젝트가 있는, 검사할 지정된 테이블 스페이스와 다른 테이블 스페이스에 데이터 오브젝트가 있는 경우에도, 계속 INDEXDATA 검사 혜택을 받을 수 있습니다.
  - 파티션된 테이블의 경우 각 인덱스는 다른 테이블 스페이스에 상주할 수 있습니다. 지정한 테이블 스페이스에 있는 인덱스만 데이터의 인덱스 검사에 활용할 수 있습니다. 하나의 테이블에 대해 모든 인덱스를 검사하려면 CHECK TABLE 옵션이나 CHECK DATABASE 옵션을 사용하십시오.
5. 온라인 검사 처리에서는 분리 레벨의 언커미트 읽기를 사용하여 데이터베이스 오브젝트에 액세스합니다. COMMIT 처리는 INSPECT 처리 중에 수행됩니다. INSPECT를 호출하기 전에 COMMIT 또는 ROLLBACK을 발행하여 작업 단위(UOW)를 종료하는 것이 좋습니다.
6. 온라인 검사 점검 처리 중에 형식화되지 않은 검사 데이터 결과가 지정된 결과 파일로 작성됩니다. 파일은 진단 데이터 디렉토리 경로에서 기록됩니다. 점검 처리 중에 오류가 발견되지 않은 경우 결과 출력 파일은 INSPECT 조작 종료 시 지워집니다. 점검 처리 중에 오류가 발견되면 결과 출력 파일은 INSPECT 조작 종료 시 지워지지 않습니다. 점검 처리 완료 후, 검사 세부사항을 보려면 검사 결과 데이터가 db2inspf 유틸리티로 형식화되어야 합니다. 결과 파일의 확장자는 데이터베이스 파티션 번호입니다.
7. 파티션된 데이터베이스 환경에서, 데이터베이스 파티션은 확장자가 데이터베이스 파티션 번호인 고유한 결과 출력 파일을 생성합니다. 결과 출력 파일의 출력 위치는 데이터베이스 관리 프로그램 진단 데이터 디렉토리 경로가 됩니다. 이미 존재하는 파일의 이름이 지정되면 조작은 처리되지 않습니다. 파일 이름을 지정하기 전에 파일을 제거해야 합니다.
8. 일반 온라인 검사 처리에서는 분리 레벨의 커미트되지 않은 읽기를 사용하여 데이터베이스 오브젝트에 액세스합니다. 테이블에 압축 사전 삽입 시 쓰기 잠금을 획득하려고 합니다. 사전 삽입 잠금에 대한 세부사항은 ROWCOMPESTIMATE 옵션을 참조하십시오. 검사 처리 중에 커미트 처리가 완료됩니다. 검사 조작을 시작하기 전에 COMMIT 또는 ROLLBACK을 발행하여 작업 단위(UOW)를 종료하는 것이 좋습니다.

9. INDEXDATA 옵션은 인덱스 및 데이터 사이의 논리적 불일치만 조사합니다. 따라서 INDEXDATA 점검을 실행하기 전에 먼저 INDEX 및 DATA 점검을 별도로 실행하여 실제 손상은 제외하는 것이 좋습니다.
10. INDEXDATA 매개변수와 함께 지정된 INSPECT 명령은 현재 검사되는 오브젝트/테이블을 포함하여 모든 오브젝트/테이블에 대해 읽기/쓰기 액세스를 허용하는 동안 인덱스 대 데이터 일관성 검사를 수행합니다. INSPECT INDEXDATA 옵션에는 다음 검사가 포함됩니다.
  - 제공된 인덱스 항목에 대한 데이터 행의 존재 여부
  - 데이터 값에 대한 키 검증

INDEXDATA 옵션을 지정한 경우:

- 디폴트로, 명시적으로 지정된 레벨 절 옵션의 값만 사용됩니다. 명시적으로 지정되지 않은 레벨 절 옵션의 경우 디폴트 레벨은 NORMAL에서 NONE으로 대체됩니다. 예를 들어 INDEXDATA가 지정된 유일한 레벨 절 옵션인 경우, 디폴트로 인덱스 대 데이터 검사만 수행됩니다.
11. BLOCKMAP 옵션은 비어 있는 다차원적으로 클러스터된(MDC) 테이블 블록을 재개하기 위해 재구성 다음에 테이블 스페이스에 사용하도록 블록이 재개되었는지 여부를 포함하는 정보를 리턴합니다.





---

## 제 68 장 LIST ACTIVE DATABASES

GET SNAPSHOT FOR ALL DATABASES 명령으로 나열되는 정보의 서브세트를 표시합니다. 활성 데이터베이스가 응용프로그램에 의한 연결 및 사용에 사용할 수 있습니다. 각 활성 데이터베이스의 경우 이 명령은 다음을 표시합니다.

- 데이터베이스 이름
- 데이터베이스에 현재 연결된 응용프로그램 수
- 데이터베이스 경로.

### 범위

이 명령은 \$HOME/sql1lib/db2nodes.cfg에 나열되는 모든 데이터베이스 파티션에서 실행할 수 있습니다. 이들 데이터베이스 파티션의 어디에서나 동일한 정보를 리턴합니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*
- *sysmon*

### 명령 구문

```
▶▶—LIST ACTIVE DATABASES—┐
                             └─AT DBPARTITIONNUM—db-partition-number—┘
                             └─GLOBAL—┘
```

### 명령 매개변수

**AT DBPARTITIONNUM** *db-partition-number*

모니터 스위치의 상태가 표시될 데이터베이스 파티션을 지정합니다.

**GLOBAL**

파티션된 데이터베이스 환경의 모든 노드에 대한 집계 결과를 리턴합니다.

**예:**

다음은 LIST ACTIVE DATABASES 명령의 샘플 출력입니다.

활동 중인 데이터베이스

```
Database name                = TEST
Applications connected currently = 0
```

Database path	= /home/smith/smith/NODE0000/SQL00002/
Database name	= SAMPLE
Applications connected currently	= 1
Database path	= /home/smith/smith/NODE0000/SQL00001/

## 호환성

버전 8 이전 버전과의 호환성:

- DBPARTITIONNUM에 키워드 NODE를 대체할 수 있습니다.

## 제 69 장 LIST APPLICATIONS

모든 사용 중인 데이터베이스 응용프로그램의 이름, 권한부여 ID(사용자 이름), 응용프로그램 핸들, 응용프로그램 ID 및 데이터베이스 이름을 표준 출력으로 표시합니다. 이 명령은 또한 선택적으로 응용프로그램의 시퀀스 번호, 상태, 상태 변경 시간 및 데이터베이스 경로를 표시합니다.

### 범위

이 명령은 명령이 발행된 데이터베이스 파티션의 정보만 리턴합니다.

### 권한 부여

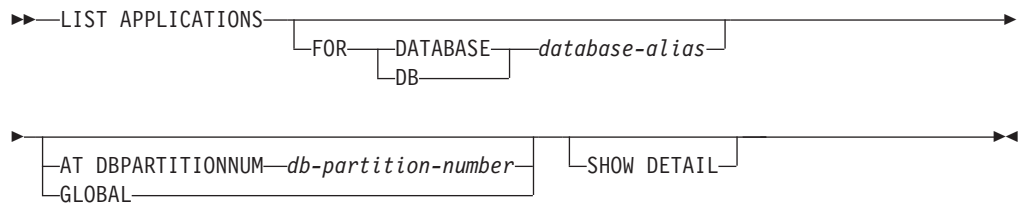
다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*
- *sysmon*

### 필수 연결

인스턴스. 리모트 인스턴스에 대한 응용프로그램을 나열하려면 먼저 이 인스턴스에 접속해야 합니다.

### 명령 구문



### 명령 매개변수

#### FOR DATABASE *database-alias*

지정된 데이터베이스에 연결된 각 응용프로그램의 정보가 표시됩니다. 데이터베이스 이름 정보는 표시되지 않습니다. 이 옵션이 지정되지 않으면, 사용자가 현재 접속한 데이터베이스 파티션에서 현재 데이터베이스에 연결된 각 응용프로그램의 정보를 표시합니다.

디폴트 응용프로그램 정보는 다음과 같이 구성되어 있습니다.

- 권한 부여 ID
- 응용프로그램 이름
- 응용프로그램 핸들
- 응용프로그램 ID
- 데이터베이스 이름
- 에이전트 수

**AT DBPARTITIONNUM** *db-partition-number*

모니터 스위치 상태가 표시되는 데이터베이스 파티션을 지정합니다.

**GLOBAL**

파티션된 데이터베이스 환경의 모든 데이터베이스 파티션에 대한 집계 결과를 리턴합니다.

**SHOW DETAIL**

일부 추가적인 출력 정보는 다음을 포함합니다.

- CONNECT 권한 ID
- 시퀀스 번호
- DB 파티션 번호 조정
- 코디네이터 pid/스레드
- 상태
- 상태 변경 시간
- 노드
- 데이터베이스 경로

이 옵션이 지정되면, 출력을 파일로 경로 재지정하고 편집기를 사용하여 보고서를 보는 것이 좋습니다. 출력 라인이 화면에 표시되면 래핑합니다.

**예:**

SAMPLE 데이터베이스에 연결된 응용프로그램에 대한 세부 정보를 나열하려면, 다음을 발행하십시오.

```
list applications for database sample show detail
```

**사용 시 참고사항**

데이터베이스 관리자는 이 명령의 출력을 문제점 판별을 위해 사용할 수 있습니다. 또한 데이터베이스 관리자가 GET SNAPSHOT 명령이나 FORCE APPLICATION 명령을 응용프로그램에서 사용하려면 이 정보가 필요합니다.

리모트 인스턴스(또는 다른 로컬 인스턴스)에서 응용프로그램을 나열하려면, 해당 인스턴스에 우선 접속하는 것이 필요합니다. 접속되어 있을 때 FOR DATABASE가 지정되고, 현재 접속과 다른 인스턴스에 데이터베이스가 있으면 명령은 실패합니다.

LIST APPLICATIONS는 사용자 응용프로그램만을 표시하지만 LIST APPLICATIONS SHOW DETAIL은 시스템 응용프로그램을 포함해서 모든 응용프로그램을 표시합니다. 이벤트 모니터는 시스템 응용프로그램의 예입니다. 시스템 응용프로그램은 보통 "db2"(예: db2stmm, db2taskd)로 시작하는 응용프로그램 이름으로 스냅샷 출력에 표시됩니다. 추가 정보는 is\_system\_appl 모니터 요소에 대한 [관련 링크](#) 절을 참조하십시오.

## 호환성

버전 8 이전 버전과의 호환성:

- NODE 키워드를 DBPARTITIONNUM 대신 사용할 수 있습니다.



---

## 제 70 장 LIST COMMAND OPTIONS

환경 변수의 현재 설정을 나열합니다.

- DB2BQTIME
- DB2DQTRY
- DB2RQTIME
- DB2IQTIME
- DB2OPTIONS.

### 권한 부여

없음

### 필수 연결

없음

### 명령 구문

▶—LIST COMMAND OPTIONS—▶

### 명령 매개변수

없음

### 예:

다음은 LIST COMMAND OPTIONS의 샘플 출력입니다.

명령행 처리기(CLP) 옵션 설정값

백엔드 프로세스 대기 시간(초)	(DB2BQTIME) = 1
백엔드에 대한 연결 재시도 수	(DB2BQTRY) = 60
요청 큐 대기 시간(초)	(DB2RQTIME) = 5
입력 큐 대기 시간(초)	(DB2IQTIME) = 5
명령 옵션	(DB2OPTIONS) =

옵션	설명	현재 설정값
-a	SQLCA 표시	OFF
-c	자동 커미트	ON
-d	XML 선언 검색 및 표시	OFF
-e	SQLCODE/SQLSTATE 표시	OFF
-f	입력 파일에서 읽기	OFF
-l	실행기록 파일에 명령 로그	OFF
-n	줄 바꾸기 문자 제거	OFF
-o	출력 표시	ON

-p	대화식 입력 프롬프트 표시	ON
-r	보고서 파일에 출력 저장	OFF
-s	명령 오류시 실행 중지	OFF
-t	명령문 종료 문자 설정	OFF
-v	현재 명령 반향 출력	OFF
-w	FETCH/SELECT 경고 메시지 표시	ON
-z	출력 파일에 모든 출력 저장	OFF



---

## 제 71 장 LIST DATABASE DIRECTORY

시스템 데이터베이스 디렉토리의 콘텐츠를 나열합니다. 경로가 지정된 경우, 로컬 데이터베이스 디렉토리의 콘텐츠가 나열됩니다.

### 범위

이 명령이 ON *path* 매개변수 없이 발행되면, 시스템 데이터베이스 디렉토리가 리턴됩니다. 이 정보는 모든 데이터베이스 파티션에서 동일합니다.

ON *path* 매개변수가 지정된 경우, 해당 경로에서의 로컬 데이터베이스 디렉토리가 리턴됩니다. 이 정보는 모든 데이터베이스 파티션에서 동일하지 않습니다.

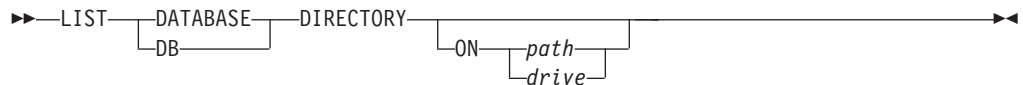
### 권한 부여

없음

### 필수 연결

없음. 디렉토리 조작은 로컬 디렉토리에만 영향을 미칩니다.

### 명령 구문



### 명령 매개변수

#### ON *path* | *drive*

정보를 나열할 로컬 데이터베이스 디렉토리를 지정합니다. 지정되지 않으면, 시스템 데이터베이스 디렉토리의 콘텐츠가 나열됩니다. 인스턴스 이름이 경로에 포함됩니다. 경로의 파트로 인스턴스 이름을 지정하지 마십시오.

### 예:

시스템 데이터베이스 디렉토리의 샘플 출력은 다음과 같습니다.

시스템 데이터베이스 디렉토리

디렉토리의 항목 수 = 2

데이터베이스 항목 1:

데이터베이스 별명 = SAMPLE

데이터베이스 이름 = SAMPLE

로컬 데이터베이스 디렉토리 = /home/smith

데이터베이스 릴리스 레벨 = 8.00

```

주석 =
디렉토리 항목 유형 = 간접
카탈로그 데이터베이스 파티션 번호 = 0
대체 서버 호스트 이름 = montero
대체 서버 포트 번호 = 29384

```

```

데이터베이스 항목 2:
데이터베이스 별명 = TC004000
데이터베이스 이름 = TC004000
노드 이름 = PRINODE
데이터베이스 릴리스 레벨 = a.00
주석 =
디렉토리 항목 유형 = LDAP
카탈로그 데이터베이스 파티션 번호 = -1
게이트웨이 노드 이름 = PRIGW
대체 서버 노드 이름 =
대체 서버 게이트웨이 노드 이름 = ALTGW

```

다음은 로컬 데이터베이스 디렉토리의 샘플 출력을 표시합니다.

/u/smith의 로컬 데이터베이스 디렉토리

디렉토리의 항목 수 = 1

데이터베이스 항목 1:

```

데이터베이스 별명 = SAMPLE
데이터베이스 이름 = SAMPLE
데이터베이스 디렉토리 = SQL00001
데이터베이스 릴리스 레벨 = 8.00
주석 =
디렉토리 항목 유형 = 홈
카탈로그 데이터베이스 파티션 번호 = 0
데이터베이스 파티션 번호 = 0

```

해당 필드는 다음과 같이 식별됩니다.

#### 데이터베이스 별명

데이터베이스가 작성되거나 카탈로그될 때 *alias* 매개변수의 값. 데이터베이스가 카탈로그될 때 별명이 입력되지 않은 경우, 데이터베이스 관리 프로그램은 데이터베이스가 카탈로그되었을 때의 *database-name* 매개변수 값을 사용합니다.

#### 데이터베이스 이름

데이터베이스가 카탈로그되었을 때의 *database-name* 매개변수의 값. 이 이름은 보통 데이터베이스가 작성되었을 때의 이름입니다.

#### 로컬 데이터베이스 디렉토리

데이터베이스가 있는 경로. 시스템 데이터베이스 디렉토리가 스캔된 경우에만 이 필드가 채워집니다.

### 데이터베이스 디렉토리

데이터베이스가 있는 디렉토리의 이름. 로컬 데이터베이스 디렉토리가 스캔된 경우에만 이 필드가 채워집니다.

### 노드 이름

리모트 노드의 이름. 이 이름은 데이터베이스 및 노드가 카탈로그되었을 때 *nodename* 매개변수용으로 입력된 값에 해당합니다.

### 데이터베이스 릴리스 레벨

데이터베이스에서 조작할 수 있는 데이터베이스 관리 프로그램의 릴리스 레벨.

주석 카탈로그되었을 때 입력된 데이터베이스와 연관된 모든 주석.

### 디렉토리 항목 유형

데이터베이스의 위치:

- 다른 노드에 있는 데이터베이스를 설명하는 리모트 항목.
- 간접 항목은 로컬인 데이터베이스를 설명합니다. 시스템 데이터베이스 디렉토리와 동일한 노드에 있는 데이터베이스는 간접적으로 홈 항목을 (로컬 데이터베이스 디렉토리로) 참조하며 간접 항목으로 간주됩니다.
- 홈 항목은 데이터베이스 디렉토리가 로컬 데이터베이스 디렉토리와 동일한 경로에 있음을 표시합니다.
- LDAP 항목은 데이터베이스 위치 정보가 LDAP 서버에 저장됨을 표시합니다.

시스템 데이터베이스 디렉토리의 모든 항목은 리모트 또는 간접 중 하나입니다. 로컬 데이터베이스 디렉토리의 모든 항목은 간접 항목으로서 시스템 데이터베이스 디렉토리에서 식별됩니다.

인증 클라이언트에서 카탈로그된 인증 유형.

### 핵심부 이름

완전한 Kerberos 핵심부 이름을 지정합니다.

### 카탈로그 데이터베이스 파티션 번호

어떤 노드가 카탈로그 데이터베이스 파티션인지 지정합니다. 이것은 CREATE DATABASE 명령이 발행된 데이터베이스 파티션입니다.

### 데이터베이스 파티션 번호

db2nodes.cfg에 지정된 번호를 명령이 발행된 노드에 지정합니다.

### 대체 서버 호스트 이름

데이터베이스로의 연결 시 통신에 실패하면 사용되는 대체 서버의 IP 주소나 호스트 이름을 지정합니다. 이 필드는 시스템 데이터베이스 디렉토리에만 표시됩니다.

#### 대체 서버 포트 번호

데이터베이스로의 연결 시 통신에 실패하면 사용되는 대체 서버의 포트 번호를 지정합니다. 이 필드는 시스템 데이터베이스 디렉토리에만 표시됩니다.

#### 대체 서버 노드 이름

디렉토리 항목 유형이 LDAP인 경우, 데이터베이스로의 연결 시 통신에 실패하면 사용되는 대체 서버의 노드 번호를 지정합니다.

#### 대체 서버 게이트웨이 노드 이름

디렉토리 항목 유형이 LDAP인 경우, 데이터베이스로의 연결 시 통신에 실패하면 사용되는 대체 게이트웨이의 게이트웨이 노드 이름을 지정합니다.

### 사용 시 참고사항

프로세스당 열린 데이터베이스 디렉토리가 최대 8개입니다. 단일 DB2 세션 내에서 8개를 초과하는 LIST DATABASE DIRECTORY 명령을 발행하는 일괄처리 파일의 경우 이 제한을 극복하려면, 일괄처리 파일을 셸 스크립트로 변환하십시오. "db2" 접두부는 각 명령마다 새 DB2 세션을 생성합니다.

---

## 제 72 장 LIST DATABASE PARTITION GROUPS

현재 데이터베이스와 연관된 모든 데이터베이스 파티션 그룹을 나열합니다.

### 범위

이 명령은 \$HOME/sql1lib/db2nodes.cfg에 나열되는 모든 데이터베이스 파티션에서 실행할 수 있습니다. 이들 데이터베이스 파티션의 어디에서나 동일한 정보를 리턴합니다.

### 권한 부여

시스템 카탈로그 SYSCAT.DBPARTITIONGROUPS 및 SYSCAT.DBPARTITIONGROUPDEF의 경우, 다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*
- *sysmon*
- *dbadm*
- CONTROL 특권
- SELECT 특권

### 필수 연결

데이터베이스

### 명령 구문

```
▶▶—LIST DATABASE PARTITION GROUPS—┬──────────────────────────────────────────▶▶  
└─SHOW DETAIL─┘
```

### 명령 매개변수

#### SHOW DETAIL

출력은 다음 정보를 포함해야 함을 지정합니다.

- 분산 맵 ID
- 데이터베이스 파티션 번호
- 사용 중 플래그

### 예:

다음은 LIST DATABASE PARTITION GROUPS 명령의 샘플 출력입니다.

DATABASE PARTITION GROUP NAME

-----  
IBMCATGROUP  
IBMDEFAULTGROUP

2개의 레코드가 선택되었습니다.

다음은 LIST DATABASE PARTITION GROUPS SHOW DETAIL 명령의 샘플 출력입니다.

DATABASE PARTITION GROUP NAME	PMAP_ID	DATABASE PARTITION NUMBER	IN_USE
IBMCATGROUP	0		0 Y
IBMDEFAULTGROUP	1		0 Y

2개의 레코드가 선택되었습니다.

필드는 다음과 같이 식별됩니다.

**DATABASE PARTITION GROUP NAME**

데이터베이스 파티션 그룹 이름. 이름은 데이터베이스 파티션 그룹의 데이터베이스 파티션마다 반복됩니다.

**PMAP\_ID**

분산 맵의 ID. ID는 데이터베이스 파티션 그룹의 데이터베이스 파티션마다 반복됩니다.

**DATABASE PARTITION NUMBER**

데이터베이스 파티션의 번호.

**IN\_USE**

네 개의 값 중 하나:

- Y** 데이터베이스 파티션 그룹에서 데이터베이스 파티션이 사용됩니다.
- D** REDISTRIBUTE DATABASE PARTITION GROUP 조작의 결과로 데이터베이스 파티션 그룹에서 데이터베이스 파티션이 삭제되려고 합니다. 조작이 완료되면, 데이터베이스 파티션은 LIST DATABASE PARTITION GROUPS의 보고서에 포함되지 않습니다.
- A** 데이터베이스 파티션 그룹에 데이터베이스 파티션이 추가되었지만 아직 분산 맵에 추가되지 않았습니다. 데이터베이스 파티션 그룹에 있는 테이블 스페이스에 대한 컨테이너가 이 데이터베이스 파티션에 추가되었습니다. 값은 REDISTRIBUTE DATABASE PARTITION GROUP 조작이 완료될 때 Y로 변경됩니다.
- T** 데이터베이스 파티션 그룹에 데이터베이스 파티션이 추가되었지만 아직 분산 맵에 추가되지 않았습니다. 데이터베이스 파티션 그룹에 있는 테이블 스페이스에 대한 컨테이너가 이 데이터베이스 파티션에 추가되지 않았습니다. 테이블 스페이스 컨테이너는 데이터베이스 파티션 그룹의

각 테이블 스페이스에 대한 새 데이터베이스 파티션에서 추가해야 합니다. 값은 컨테이너가 추가되었을 때 A로 변경됩니다.

## 호환성

버전 8 이전 버전과의 호환성:

- NODEGROUPS 키워드는 DATABASE PARTITION GROUPS로 대체될 수 있습니다.





---

## 제 73 장 LIST DBPARTITIONNUMS

현재 데이터베이스와 연관된 모든 데이터베이스 파티션을 나열합니다.

### 범위

이 명령은 \$HOME/sql1lib/db2nodes.cfg에 나열되는 모든 데이터베이스 파티션에서 실행할 수 있습니다. 이들 데이터베이스 파티션의 어디에서나 동일한 정보를 리턴합니다.

### 권한 부여

없음

### 필수 연결

데이터베이스

### 명령 구문

▶—LIST DBPARTITIONNUMS—▶

### 명령 매개변수

없음

### 예:

다음은 LIST DBPARTITIONNUMS 명령의 출력 샘플입니다.

```
DATABASE PARTITION NUMBER
-----
                                0
                                2
                                5
                                7
                                9
```

5개의 레코드가 선택되었습니다.

### 호환성

버전 8 이전 버전과의 호환성:

- DBPARTITIONNUMS에 키워드 NODES를 대체할 수 있습니다.



---

## 제 74 장 LIST DCS APPLICATIONS

DB2 Connect Enterprise Edition을 통해 호스트 데이터베이스에 연결되는 응용프로그램에 대한 표준 출력 정보로 표시합니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*
- *sysmon*

### 필수 연결

인스턴스. 리모트 인스턴스에서 DCS 응용프로그램을 나열하려면, 해당 인스턴스에 먼저 접속하는 것이 필요합니다.

### 명령 구문

```
▶▶—LIST DCS APPLICATIONS—▶▶  
    SHOW DETAIL  
    EXTENDED
```

### 명령 매개변수

#### LIST DCS APPLICATIONS

디폴트 응용프로그램 정보는 다음을 포함합니다.

- 호스트 권한 부여 ID(*username*)
- 응용프로그램 이름
- 응용프로그램 핸들
- 아웃바운드 응용프로그램 ID(*luwid*).

#### SHOW DETAIL

출력이 다음 추가 정보를 포함하도록 지정합니다.

- 클라이언트 응용프로그램 ID
- 클라이언트 시퀀스 번호
- 클라이언트 데이터베이스 별명
- 클라이언트 노드 이름(*nname*)
- 클라이언트 릴리스 레벨

- 클라이언트 코드 페이지
- 아웃바운드 시퀀스 번호
- 호스트 데이터베이스 이름
- 호스트 릴리스 레벨.

### EXTENDED

확장 보고서를 생성합니다. 이 보고서는 SHOW DETAIL 옵션이 지정될 때 나열되는 모든 필드 이외에 다음과 같은 추가적인 필드를 포함합니다.

- DCS 응용프로그램 상태
- 상태 변경 시간
- 클라이언트 플랫폼
- 클라이언트 프로토콜
- 클라이언트 코드 페이지
- 클라이언트 응용프로그램의 프로세스 ID
- 호스트 코드화된 문자 세트 ID(CCSID).

주:

1. 응용프로그램 상태 필드에는 다음 값 중 하나가 들어 있습니다.

#### 연결 보류 - 아웃바운드

호스트 데이터베이스를 연결하기 위한 요청이 발행되었으며 DB2 Connect가 연결되기를 대기 중임을 나타냅니다.

#### 요청 대기 중

호스트 데이터베이스 연결이 설정되었으며, DB2 Connect가 클라이언트 응용프로그램에서 SQL문을 대기 중임을 나타냅니다.

#### 응답 대기 중

SQL문을 호스트 데이터베이스로 보냈음을 나타냅니다.

2. 이 상태 변경 시간은 처리 중에 시스템 모니터 UOW 스위치가 켜진 경우에만 표시됩니다. 그렇지 않으면, 수집되지 않음이 표시됩니다.

### 사용 시 참고사항

데이터베이스 관리자는 이 명령을 사용하여 게이트웨이로의 클라이언트 응용프로그램 연결을 게이트웨이로부터의 해당 호스트 연결과 일치시키기 위해 이 명령을 사용할 수 있습니다.

또한 데이터베이스 관리자는 에이전트 ID 정보를 사용하여 지정된 응용프로그램이 DB2 Connect 서버를 해제시키도록 강제할 수 있습니다.

---

## 제 75 장 LIST DCS DIRECTORY

데이터베이스 연결 서비스(DCS) 디렉토리의 콘텐츠를 나열합니다.

### 권한 부여

없음

### 필수 연결

없음

### 명령 구문

▶—LIST DCS DIRECTORY—▶

### 명령 매개변수

없음

### 예:

다음은 LIST DCS DIRECTORY의 샘플 출력입니다.

데이터베이스 연결 서비스(DCS) 디렉토리

디렉토리의 항목 수 = 1

DCS 1 entry:

로컬 데이터베이스 이름	= DB2
목표 데이터베이스 이름	= DSN_DB_1
응용프로그램 리퀘스터 이름	=
DCS 매개변수	=
주석	= DB2/MVS Location name DSN_DB_1
DCS 디렉토리 릴리스 레벨	= 0x0100

이들 필드는 다음과 같이 식별됩니다.

#### 로컬 데이터베이스 이름

목표 호스트 데이터베이스의 로컬 별명을 지정합니다. 이것은 호스트 데이터베이스가 DCS 디렉토리에서 카탈로그되었을 때 입력된 *database-name* 매개변수에 해당합니다.

### 목표 데이터베이스 이름

액세스할 수 있는 호스트 데이터베이스의 이름을 지정합니다. 이것은 호스트 데이터베이스가 DCS 디렉토리에서 카탈로그되었을 때 입력된 *target-data-base-name* 매개변수에 해당합니다.

### 응용프로그램 리퀘스터(AR) 이름

응용프로그램 리퀘스터(AR) 또는 서버에 있는 프로그램의 이름을 지정합니다.

### DCS 매개변수

응용프로그램 리퀘스터(AR)에 작업하기 위한 연결 및 운영 환경 매개변수가 들어있는 문자열입니다. 호스트 데이터베이스가 카탈로그되었을 때 입력된 매개변수 문자열에 대응합니다. 문자열은 큰따옴표로 묶어야 하며 매개변수는 쉼표로 구분해야 합니다.

주석 데이터베이스 항목을 설명합니다.

### DCS 디렉토리 릴리스 레벨

데이터베이스가 작성된 Distributed Database Connection Services® 프로그램의 버전 번호를 지정합니다.

## 사용 시 참고사항

DCS 디렉토리는 처음 CATALOG DCS DATABASE 명령이 호출될 때 작성됩니다. DB2가 설치된 경로/드라이브에서 유지되며, DB2 Connect 프로그램이 설치된 경우 위크스테이션이 액세스할 수 있는 호스트 데이터베이스에 관한 정보를 제공합니다. 호스트 데이터베이스는 다음일 수 있습니다.

- OS/390 및 z/OS 호스트의 DB2 데이터베이스
- System i 호스트의 DB2 데이터베이스
- VSE & VM 호스트의 DB2 데이터베이스

---

## 제 76 장 LIST DRDA INDOUBT TRANSACTIONS

DRDA 리퀘스터와 DRDA 서버 사이에서 인다우트(Indoubt)인 트랜잭션 목록을 제공합니다. DRDA 커밋 프로토콜이 사용되는 경우, DRDA 동기점 관리 프로그램 사이의 인다우트(Indoubt) 트랜잭션을 나열합니다.

### 권한 부여

없음

### 필수 연결

인스턴스

### 명령 구문

```
▶—LIST DRDA INDOUBT TRANSACTIONS—▶  
    └──WITH PROMPTING──┘
```

### 명령 매개변수

#### WITH PROMPTING

인다우트(Indoubt) 트랜잭션이 프로세스됨을 표시합니다. 이 매개변수가 지정된 경우, 대화식 대화 상자 모드가 시작되어, 사용자가 인다우트(Indoubt) 트랜잭션을 커밋 또는 롤백할 수 있게 합니다. 이 매개변수가 지정되지 않은 경우, 인다우트(Indoubt) 트랜잭션은 표준 출력 디바이스에 쓰여지며 대화식 대화 상자 모드는 시작되지 않습니다.

무시 옵션은 지원되지 않습니다. 인다우트 트랜잭션이 커밋 또는 롤백되는 경우, 트랜잭션은 자동으로 무시됩니다.

대화식 대화 상자 모드는 사용자가 다음을 수행하도록 허용합니다.

- 모든 인다우트(Indoubt) 트랜잭션 나열(1 입력)
- 인다우트(Indoubt) 트랜잭션 번호  $x$  나열(1와, 이어서 유효한 트랜잭션 번호 입력)
- 종료(q 입력)
- 트랜잭션 번호  $x$  커밋(c, 이어서 유효한 트랜잭션 번호 입력)
- 트랜잭션 번호  $x$  롤백(r, 이어서 유효한 트랜잭션 번호 입력)

공백은 인수에서 명령 문자를 분리시켜야 합니다.

트랜잭션을 커밋 또는 롤백하기 전에, 트랜잭션 데이터가 표시되며, 사용자에게 조치를 확인하도록 요청됩니다.

## 사용 시 참고사항

분산 작업 단위(DUOW)에서 코디네이터와 구성원(participant) 사이의 통신이 단절되면 DRDA 인다우트(Indoubt) 트랜잭션이 발생합니다. 분산 작업 단위(DUOW)를 사용하여 사용자나 응용프로그램은 단일 작업 단위(UOW) 내의 여러 위치에서 데이터를 읽고 갱신할 수 있습니다. 그러한 작업을 위해서는 2단계의 커미트가 필요합니다.

첫 번째 단계에서는 모든 구성원(participant)이 커미트를 준비하도록 요청합니다. 두 번째 단계에서는 트랜잭션을 커미트하거나 롤백합니다. 코디네이터나 구성원(participant)이 첫 번째 단계 이후에 사용 불가능하게 되면 분산 트랜잭션은 인다우트(Indoubt) 상태입니다.

LIST DRDA INDOUBT TRANSACTIONS 명령을 실행하기 전에 응용프로그램 프로세스를 DB2 동기점 관리 프로그램(SPM) 인스턴스에 연결해야 합니다. **spm\_name** 데이터베이스 관리 프로그램 구성 매개변수를 CONNECT 문의 *dbalias*로 사용하십시오.

커미트를 조정하기 위해 SPM을 사용하는 TCP/IP 연결은 DRDA 2단계 커미트 프로토콜을 사용합니다.



## 제 77 장 LIST HISTORY

실행기록 파일의 항목을 나열합니다. 실행기록 파일에는 복구 및 관리 이벤트의 레코드가 들어 있습니다. 복구 이벤트에는 전체 데이터베이스 및 테이블 스페이스 레벨 백업, 증분식 백업, 리스토어 및 롤 포워드 조작이 포함됩니다. 추가 로그된 이벤트는 테이블 스페이스 작성, 변경, 삭제(drop) 또는 이름 바꾸기, 테이블 재구성, 테이블 삭제 및 로드가 포함됩니다.

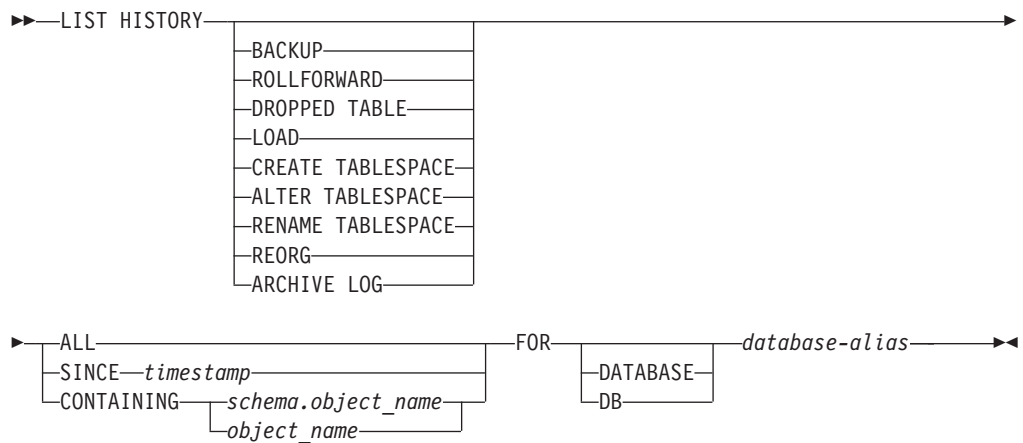
### 권한 부여

없음

### 필수 연결

인스턴스. 이 명령을 리모트 데이터베이스에 대해 실행하려면 모든 리모트 데이터베이스에 접속해야 합니다. 로컬 데이터베이스의 경우 명시적 접속이 필요하지 않습니다.

### 명령 구문



### 명령 매개변수

#### HISTORY

실행기록 파일에 현재 로그된 모든 이벤트를 나열합니다.

#### BACKUP

백업 및 리스토어 조작을 나열합니다.

#### ROLLFORWARD

롤 포워드 조작을 나열합니다.

#### DROPPED TABLE

삭제된 테이블 레코드를 나열합니다. 삭제된 테이블 레코드는 테이블이 삭제되

고 테이블이 들어있는 테이블 스페이스에서 DROPPED TABLE RECOVERY 옵션이 사용 가능할 때만 작성됩니다. 파티션된 테이블에 대한 CREATE TABLE 구문을 리턴하고 삭제된 테이블에 대한 데이터가 들어있는 테이블 스페이스를 표시합니다.

## LOAD

로드 조작을 나열합니다.

## CREATE TABLESPACE

테이블 스페이스 작성 및 삭제(drop) 조작을 나열합니다.

## RENAME TABLESPACE

테이블 스페이스 이름 바꾸기 조작을 나열합니다.

## REORG

재구성 조작을 나열합니다. 파티션된 테이블의 재구성된 각 데이터 파티션에 대한 정보를 포함합니다.

## ALTER TABLESPACE

테이블 스페이스 변경 조작을 나열합니다.

## ARCHIVE LOG

아카이브 로그 조작 및 아카이브된 로그를 나열합니다.

**ALL** 실행기록 파일의 지정된 유형의 모든 항목을 나열합니다.

**SINCE** *timestamp*

완료 시간소인(형식 *yyyymmddhhmmss*) 또는 초기 접두부(최소 *yyyy*)를 지정할 수 있습니다. 제공되는 시간소인보다 크거나 같은 시간소인을 갖는 모든 항목이 나열됩니다.

**CONTAINING** *schema.object\_name*

이 자격이 부여된 이름이 테이블을 고유하게 식별합니다.

**CONTAINING** *object\_name*

이 규정되지 않은 이름은 테이블 스페이스를 고유하게 식별합니다.

**FOR DATABASE** *database-alias*

복구 실행기록 파일이 나열될 데이터베이스를 식별하는 데 사용됩니다.

**예:**

```
db2 list history since 19980201 for sample
db2 list history backup containing userspace1 for sample
db2 list history dropped table all for db sample

db2 -v "list history reorg all for wsdb"
```

Op	Obj	시간소인+순서	유형	Dev	최초 로그	현재 로그	백업 ID
----	-----	---------	----	-----	-------	-------	-------

G	T	20080924101408	N		S0000000.LOG	S0000000.LOG	
---	---	----------------	---	--	--------------	--------------	--

테이블: "ZHMFENG"."T1"

-----  
주식: REORG RECALIM  
시작 시간: 20080924101408  
종료 시간: 20080924101409  
상태: A

## 사용 시 참고사항

SYSIBMADM.DB\_HISTORY 관리 뷰를 사용하여 모든 데이터베이스 파티션에서 데이터를 검색할 수 있습니다.

이 명령으로 생성되는 보고서에는 다음 기호가 들어 있습니다.

### 조작

- A - 테이블 스페이스 작성
- B - 백업
- C - 로드 사본
- D - 삭제된 테이블
- F - 롤 포워드
- G - 테이블 재구성
- L - 로드
- N - 테이블 스페이스 이름 바꾸기
- O - 테이블 스페이스 삭제
- Q - Quiesce
- R - 리스토어
- T - 테이블 스페이스 변경
- U - 언로드
- X - 아카이브 로그

### 유형

#### 아카이브 로그 유형:

- P - 1차 로그 경로
- M - 2차(미러) 로그 경로
- N - 아카이브 로그 명령
- F - 장애 복구 아카이브 경로
- 1 - 1차 로그 아카이브 메소드
- 2 - 2차 로그 아카이브 메소드

#### 백업 유형:

- F - 오프라인
- N - 온라인
- I - 증분 오프라인
- O - 증분 온라인
- D - 델타 오프라인
- E - 델타 온라인
- R - 재빌드

#### 롤 포워드 유형:

- E - 로그의 끝
- P - 특정 시점

#### 로드 유형:

I - 삽입  
R - 바꾸기

테이블 스페이스 변경 유형:

C - 컨테이너 추가  
R - 재조정

Quiesce 유형:

S - Quiesce 상태 공유  
U - Quiesce 상태 갱신  
X - Quiesce 독점  
Z - Quiesce 재설정

## 제 78 장 LIST INDOUBT TRANSACTIONS

인다우트(Indoubt)된 트랜잭션의 목록을 제공합니다. 사용자는 대화식으로 인다우트(Indoubt) 트랜잭션을 커밋, 롤백 또는 무시할 수 있습니다.

2단계 커밋 프로토콜은 다음으로 이루어집니다.

1. 자원 관리 프로그램이 로그 페이지를 디스크에 씬으로써 COMMIT 또는 ROLLBACK 프리미티브 중 하나로 응답할 수 있는 PREPARE 단계.
2. 트랜잭션이 실제로 커밋되거나 롤백되는 COMMIT(또는 ROLLBACK) 단계.

경험적으로 완료된 트랜잭션(경험적으로 커밋되거나 롤백된 트랜잭션)에 의해 보유된 트랜잭션 릴리스 자원 무시. 인다우트(Indoubt) 트랜잭션은 준비되었으나, 아직 커밋되거나 롤백되지 않은 트랜잭션입니다.

### 범위

이 명령은 실행된 노드에서 인다우트(Indoubt) 트랜잭션 목록을 리턴합니다.

### 권한 부여

없음

### 필수 연결

데이터베이스. 내재된 연결이 사용 가능한 경우 디폴트 데이터베이스에 연결됩니다.

### 명령 구문

```
»—LIST INDOUBT TRANSACTIONS—  
   |_____|  
   WITH PROMPTING
```

### 명령 매개변수

#### WITH PROMPTING

인다우트(Indoubt) 트랜잭션이 프로세스됨을 표시합니다. 이 매개변수가 지정된 경우, 대화식 대화 상자 모드가 시작되어, 사용자가 인다우트(Indoubt) 트랜잭션을 커밋, 롤백 또는 무시하게 합니다. 이 매개변수가 지정되지 않은 경우, 인다우트(Indoubt) 트랜잭션은 표준 출력 디바이스에 쓰여지며 대화식 대화 상자 모드는 시작되지 않습니다.

대화식 대화 상자 모드는 사용자가 다음을 수행하도록 허용합니다.

- 모든 인다우트(Indoubt) 트랜잭션 나열(1 입력)

- 인다우트(Indoubt) 트랜잭션 번호  $x$  나열(1와, 이어서 유효한 트랜잭션 번호 입력)
- 종료(q 입력)
- 트랜잭션 번호  $x$  커밋(c, 이어서 유효한 트랜잭션 번호 입력)
- 트랜잭션 번호  $x$  롤백(r, 이어서 유효한 트랜잭션 번호 입력)
- 트랜잭션 번호  $x$  무시(f, 이어서 유효한 트랜잭션 번호 입력)

공백은 인수에서 명령 문자를 분리시켜야 합니다.

트랜잭션을 커밋, 롤백 또는 무시하기 전에, 트랜잭션 데이터가 표시되며, 사용자에게 조치를 확인하도록 요청됩니다.

LIST INDOUBT TRANSACTIONS 명령은 각 인다우트(Indoubt) 트랜잭션에서 데이터베이스의 역할을 표시하기 위해 *type* 정보를 리턴합니다.

**TM** 인다우트(Indoubt) 트랜잭션이 트랜잭션 관리 프로그램 데이터베이스로서 데이터베이스를 사용 중임을 표시합니다.

**RM** 인다우트(Indoubt) 트랜잭션이 자원 관리 프로그램으로서 데이터베이스를 사용 중임을 표시하며, 이는 트랜잭션에서 참여 중인 데이터베이스 중 하나이지만 트랜잭션 관리 프로그램 데이터베이스는 아님을 나타냅니다.

## 사용 시 참고사항

인다우트(Indoubt) 트랜잭션은 인다우트(Indoubt) 상태로 남은 전역 트랜잭션입니다. 이것은 트랜잭션 관리 프로그램(TM)이나 최소한 하나의 자원 관리 프로그램(RM) 중 하나가 2단계 커밋 프로토콜 중 첫 번째 단계(즉, PREPARE 단계)를 정상적으로 완료한 후에 사용 불가능하게 될 때 발생합니다. 해당 관리 프로그램이 다시 사용 가능해지면 TM이 인다우트(Indoubt) 상태 정보로 자신의 로그를 통합하기까지 RM은 트랜잭션의 분기를 롤백 또는 커밋할지 여부를 알지 못합니다. 또한 인다우트(Indoubt) 트랜잭션은 MPP 환경에 존재할 수 있습니다.

현재 연결된 데이터베이스에 대해 LIST INDOUBT TRANSACTIONS이 발행되면, 해당 데이터베이스의 인다우트(Indoubt) 트랜잭션에 대한 정보를 리턴합니다.

상태가 인다우트(i), 누락 커밋 승인(m), 또는 누락 페더레이티드 커밋 승인(d)인 트랜잭션만 커밋될 수 있습니다.

상태가 인다우트(i), 누락 페더레이티드 롤백 승인(b) 또는 종료(e)인 트랜잭션만 롤백될 수 있습니다.

상태가 커밋(c), 롤백(r), 누락 페더레이티드 커밋 승인(d) 또는 누락 페더레이티드 롤백 승인(b)인 트랜잭션만 무시될 수 있습니다.

2단계 커미트의 커미트 단계에서, 코디네이터 노드는 커미트 승인을 대기합니다. 하나 이상의 노드가 응답하지 않는 경우(예를 들어, 노트 실패로 인해), 트랜잭션은 누락 커미트 승인 상태에 놓입니다.

인다우트(Indoubt) 트랜잭션 정보는 명령이 발행된 시간에만 유효합니다. 일단 대화식 대화 상자 모드에서, 트랜잭션 상태는 외부 활동 때문에 변경됩니다. 이런 경우, 그리고 더 이상 적절한 상태가 아닌 인다우트(Indoubt) 트랜잭션을 처리하려고 시도하면, 오류 메시지가 표시됩니다.

이러한 유형의 오류가 발생하면, 사용자는 대화식 대화 상자를 종료(q)하고 LIST INDOUBT TRANSACTIONS WITH PROMPTING 명령을 재발행하여 표시된 정보를 새로 고쳐야 합니다.





---

## 제 79 장 LIST NODE DIRECTORY

노드 디렉토리의 콘텐츠를 나열합니다.

### 권한 부여

없음

### 필수 연결

없음

### 명령 구문

```
▶▶ LIST [ADMIN] NODE DIRECTORY [SHOW DETAIL] ▶▶
```

### 명령 매개변수

#### ADMIN

관리 서버 노드를 지정합니다.

#### SHOW DETAIL

출력은 다음 정보를 포함해야 함을 지정합니다.

- 리모트 인스턴스 이름
- 시스템
- 운영 체제 유형

### 예:

다음은 LIST NODE DIRECTORY의 샘플 출력입니다.

```
노드 디렉토리
디렉토리의 항목 수           = 2
```

#### Node 1 entry:

```
노드 이름           = LANNODE
주석                =
디렉토리 항목 유형 = LDAP
프로토콜           = TCP/IP
호스트 이름        = LAN.db2ntd3.torolab.ibm.com
서비스 이름        = 50000
```

#### Node 2 entry:

```
노드 이름           = TLBA10ME
```

```

주석 =
디렉토리 항목 유형 = LOCAL
프로토콜 = TCPIP
호스트 이름 = t1ba10me
서비스 이름 = 447

```

다음은 LIST ADMIN NODE DIRECTORY의 샘플 출력입니다.

```

노드 디렉토리
디렉토리의 항목 수 = 2

```

Node 1 entry:

```

노드 이름 = LOCALADM
주석 =
디렉토리 항목 유형 = LOCAL
프로토콜 = TCPIP
호스트 이름 = jaguar
서비스 이름 = 523

```

Node 2 entry:

```

노드 이름 = MYDB2DAS
주석 =
디렉토리 항목 유형 = LDAP
프로토콜 = TCPIP
호스트 이름 = peng.torolab.ibm.com
서비스 이름 = 523

```

일반 필드는 다음과 같이 식별됩니다.

#### 노드 이름

리모트 노드의 이름. 이것은 노드가 카탈로그될 때 *nodename* 매개변수용으로 입력된 이름에 해당합니다.

**주석** 노드가 카탈로그되었을 때 입력된 노드와 연관된 주석. 노드 디렉토리에서 주석을 변경하려면, 노드를 카탈로그 해제한 후 새 주석으로 다시 카탈로그하십시오.

#### 디렉토리 항목 유형

LOCAL은 항목이 로컬 노드 디렉토리 파일에 있음을 의미합니다. LDAP은 LDAP 서버나 LDAP 캐시에 항목이 있음을 의미합니다.

#### 프로토콜

노드용으로 카탈로그된 통신 프로토콜.

특정 노드 유형과 연관된 필드 정보는, 적용 가능한 CATALOG...NODE 명령을 참조하십시오.

## 사용 시 참고사항

노드 디렉토리가 각 IBM Data Server Runtime Client에서 작성되고 유지됩니다. 클라이언트가 액세스할 수 있는 데이터베이스가 있는 각 리모트 워크스테이션의 항목을 포함합니다. DB2 클라이언트는 데이터베이스 연결이나 인스턴스 접속이 요청될 때마다 노드 디렉토리의 통신 엔드 포인트 정보를 사용합니다.

데이터베이스 관리 프로그램은 노드 항목을 작성하여 CATALOG...NODE 명령을 처리할 때마다 노드 디렉토리에 추가합니다. 노드에서 사용하는 통신 프로토콜에 따라 항목은 다를 수 있습니다.

노드 디렉토리는 다음과 같은 유형의 노드 항목을 포함할 수 있습니다.

- LDAP
- 로컬
- Named pipe
- TCPIP
- TCPIP4
- TCPIP6



---

## 제 80 장 LIST ODBC DATA SOURCES

모든 사용 가능한 사용자 또는 시스템 ODBC 데이터 소스를 나열합니다.

ODBC(Open Database Connectivity) 용어에서 데이터 소스는 특정 데이터베이스에 대한 사용자 정의 이름입니다. 해당 이름이 ODBC API를 통해 데이터베이스 또는 파일 시스템에 액세스하는 데 사용됩니다. Windows에서는 사용자 또는 시스템 데이터 소스를 카탈로그할 수 있습니다. 사용자 데이터 소스는 이를 카탈로그한 사용자만 볼 수 있는 반면, 시스템 데이터 소스는 다른 모든 사용자가 보고 사용할 수 있습니다.

이 명령은 Windows에서만 사용할 수 있습니다.

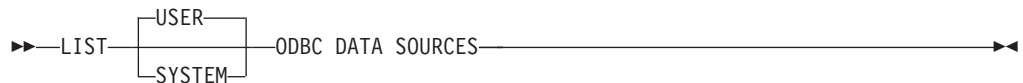
### 권한 부여

없음

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

**USER** 사용자 ODBC 데이터 소스만 나열합니다. 이것은 키워드를 지정하지 않는 경우 디폴트입니다.

### SYSTEM

시스템 ODBC 데이터 소스만 나열합니다.

### 예:

다음은 LIST ODBC DATA SOURCES 명령의 샘플 출력입니다.

사용자 ODBC 데이터소스

Data source name	Description
SAMPLE	IBM DB2 ODBC DRIVER



---

## 제 81 장 LIST PACKAGES/TABLES

현재 데이터베이스와 연결된 패키지 또는 테이블이 표시됩니다.

### 권한 부여

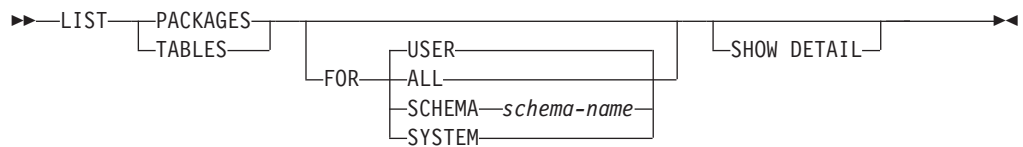
시스템 카탈로그 SYSCAT.PACKAGES(LIST PACKAGES) 및 SYSCAT.TABLES(LIST TABLES)의 경우, 다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*
- *sysmon*
- *dbadm*
- CONTROL 특권
- SELECT 특권

### 필수 연결

데이터베이스 내재된 연결이 사용 가능한 경우 디폴트 데이터베이스에 연결됩니다.

### 명령 구문



### 명령 매개변수

**FOR** FOR절을 지정하지 않은 경우, USER에 대한 패키지 또는 테이블이 표시됩니다.

**ALL** 데이터베이스의 모든 패키지 또는 테이블이 표시됩니다.

**SCHEMA** *schema-name*

지정된 스키마에 대한 데이터베이스의 모든 패키지 또는 테이블이 표시됩니다.

**SYSTEM**

데이터베이스의 모든 시스템 패키지 또는 테이블이 표시됩니다.

**USER** 현재 사용자에게 대한 데이터베이스의 모든 패키지 또는 테이블이 표시됩니다.

## SHOW DETAIL

LIST TABLES 명령에서 이 옵션을 선택하는 경우, 전체 테이블 이름 및 스키마 이름이 표시됩니다. 이 옵션을 지정하지 않는 경우, 테이블 이름은 30자로 절단되고, 31번째 컬럼의 ">" 기호는 테이블 이름의 절단된 부분을 표시합니다. 스키마 이름은 14자로 절단되고 15번째 컬럼의 ">" 기호는 스키마 이름의 절단된 부분을 표시합니다. LIST PACKAGES 명령과 함께 이 옵션을 선택하는 경우, 전체 패키지 스키마(작성자), 버전 및 권한 ID별 바운드가 표시되고, 패키지 unique\_id(16진수로 표시되는 일관성 토큰)가 표시됩니다. 이 옵션을 지정하지 않는 경우, 스키마 이름은 8자로 절단되고, 9번째 컬럼의 ">" 기호는 스키마 또는 ID별 바운드의 절단된 부분을 표시합니다. 버전은 10자로 절단되고 11번째 컬럼의 ">" 기호는 버전의 절단된 부분을 표시합니다.

예:

다음은 LIST PACKAGES의 샘플 출력입니다.

패키지	스키마	버전	바인드 방법	총 섹션	유효	형식	분리 레벨	블로킹
F4INS	USERA	VER1	SNOWBELL		221 Y	0	CS	U
F4INS	USERA	VER2.0	SNOWBELL		201 Y	0	RS	U
F4INS	USERA	VER2.3	SNOWBELL		201 N	3	CS	U
F4INS	USERA	VER2.5	SNOWBELL		201 Y	0	CS	U
PKG12	USERA		USERA		12 Y	3	RR	B
PKG15	USERA		USERA		42 Y	3	RR	B
SALARY	USERT	YEAR2000	USERT		15 Y	3	CS	N

다음은 LIST TABLES의 샘플 출력입니다.

테이블/뷰	스키마	유형	작성 시간
DEPARTMENT	SMITH	T	1997-02-19-13.32.25.971890
EMP_ACT	SMITH	T	1997-02-19-13.32.27.851115
EMP_PHOTO	SMITH	T	1997-02-19-13.32.29.953624
EMP_RESUME	SMITH	T	1997-02-19-13.32.37.837433
EMPLOYEE	SMITH	T	1997-02-19-13.32.26.348245
ORG	SMITH	T	1997-02-19-13.32.24.478021
PROJECT	SMITH	T	1997-02-19-13.32.29.300304
SALES	SMITH	T	1997-02-19-13.32.42.973739
STAFF	SMITH	T	1997-02-19-13.32.25.156337

9개의 레코드가 선택되었습니다.

## 사용 시 참고사항

LIST PACKAGES 및 LIST TABLES 명령을 사용하여 시스템 테이블과의 빠른 인터페이스를 제공할 수 있습니다.

다음 SELECT문은 시스템 테이블에서 찾은 정보를 리턴합니다. 이 명령문은 시스템 테이블이 제공하는 추가 정보를 선택하기 위해 확장할 수 있습니다.



```
select tabname, tabschema, type, create_time
from syscat.tables
order by tabschema, tabname;
```

```
select pkgname, pkgschema, pkgversion, unique_id, boundby, total_sect,
       valid, format, isolation, blocking
from syscat.packages
order by pkgschema, pkgname, pkgversion;
```

```
select tabname, tabschema, type, create_time
from syscat.tables
where tabschema = 'SYSCAT'
order by tabschema, tabname;
```

```
select pkgname, pkgschema, pkgversion, unique_id, boundby, total_sect,
       valid, format, isolation, blocking
from syscat.packages
where pkgschema = 'NULLID'
order by pkgschema, pkgname, pkgversion;
```

```
select tabname, tabschema, type, create_time
from syscat.tables
where tabschema = USER
order by tabschema, tabname;
```

```
select pkgname, pkgschema, pkgversion, unique_id, boundby, total_sect,
       valid, format, isolation, blocking
from syscat.packages
where pkgschema = USER
order by pkgschema, pkgname, pkgversion;
```



---

## 제 82 장 LIST TABLESPACE CONTAINERS

지정된 테이블 스페이스의 컨테이너를 나열합니다.

테이블 스페이스 스냅샷에는 LIST TABLESPACE CONTAINERS 명령이 표시하는 모든 정보가 포함됩니다.

### 범위

이 명령은 명령이 실행되는 노드에 대한 정보만 리턴합니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*
- *sysmon*
- *dbadm*

### 필수 연결

데이터베이스

### 명령 구문

```
▶▶—LIST TABLESPACE CONTAINERS FOR—tablespace-id—┐—SHOW DETAIL—┘▶▶
```

### 명령 매개변수

#### **FOR** *tablespace-id*

현재 데이터베이스에서 사용하는 테이블 스페이스를 고유하게 나타내는 정수. 현재 데이터베이스에서 사용되는 모든 테이블 스페이스의 목록을 가져오려면 LIST TABLESPACES 명령을 사용하십시오.

#### **SHOW DETAIL**

이 옵션을 지정한 경우 각 컨테이너에 대한 다음의 기본 정보가 제공됩니다.

- 컨테이너 ID
- 이름
- 유형(파일, 디스크 또는 경로)

이 옵션을 지정한 경우 각 컨테이너에 대한 다음의 추가 정보가 제공됩니다.

- 총 페이지 수
- 사용할 수 있는 페이지 수
- 액세스 가능 여부(yes 또는 no)

## 예:

다음은 LIST TABLESPACE CONTAINERS FOR 0의 샘플 출력입니다.

테이블 스페이스 0에 대한 테이블 스페이스 컨테이너

컨테이너 ID	= 0
이름	= /home/smith/smith/NODE0000/SQL00001/SQLT0000.0
유형	= 경로

다음은 지정된 LIST TABLESPACE CONTAINERS FOR 0 SHOW DETAIL의 샘플 출력입니다.

테이블 스페이스 0에 대한 테이블 스페이스 컨테이너

컨테이너 ID	= 0
이름	= /home/smith/smith/NODE0000/SQL00001/SQLT0000.0
유형	= 경로
전체 페이지 수	= 895
사용 가능한 페이지 수	= 895
액세스 가능	= 예

---

## 제 83 장 LIST TABLESPACES

테이블 스페이스와 현재 데이터베이스의 테이블 스페이스에 대한 정보를 나열합니다.

이 명령으로 표시되는 정보는 테이블 스페이스 스냅샷에서도 사용 가능합니다.

### 범위

이 명령은 명령이 실행되는 데이터베이스 파티션의 정보만 리턴합니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*
- *sysmon*
- *dbadm*
- LOAD 권한

### 필수 연결

데이터베이스

### 명령 구문

```
▶—LIST TABLESPACES—┐  
                        └SHOW DETAIL┘
```

### 명령 매개변수

#### SHOW DETAIL

이 옵션을 지정하지 않은 경우 각 테이블 스페이스에 대한 다음의 기본 정보가 제공됩니다.

- 테이블 스페이스 ID
- 이름
- 유형(시스템 관리 스페이스 또는 데이터베이스 관리 스페이스)
- 콘텐츠(모든 데이터, long 또는 인덱스 데이터, 또는 임시 데이터)
- 상태(현재 테이블 스페이스 상태를 표시하는 16진수 값). 외부에서 볼 수 있는 테이블 스페이스 상태는 특정 상태 값의 16진수 합으로 구성됩니다. 예

를 들어, 상태가 "Quiesce 상태: EXCLUSIVE" 및 "로드 보류"이면, 값은 0x0004 + 0x0008이고 0x000c가 됩니다. db2tbst(테이블 스페이스 상태 확보) 명령을 사용하여 제공된 16진수 값과 연관되는 테이블 스페이스 상태를 확보할 수 있습니다. 다음은 sqlutil.h에 나열된 비트 정의입니다.

0x0	일반
0x1	Quiesce됨: SHARE
0x2	Quiesce됨: UPDATE
0x4	Quiesce됨: EXCLUSIVE
0x8	로드 보류
0x10	삭제 보류
0x20	백업 보류
0x40	롤 포워드 진행 중
0x80	롤 포워드 보류
0x100	리스토어 보류
0x100	복구 보류(사용되지 않음)
0x200	작동 불가능화 보류
0x400	Reorg 진행 중
0x800	백업 진행 중
0x1000	스토리지가 정의되어야 함
0x2000	리스토어 진행 중
0x4000	오프라인 및 액세스 불가능
0x8000	삭제 보류
0x20000	로드 진행 중
0x2000000	스토리지가 정의되어야 함
0x4000000	StorDef가 '최종' 상태에 있음
0x8000000	StorDef가 롤 포워드 이전에 변경되었음
0x10000000	DMS 재조정 진행 중
0x20000000	테이블 스페이스 삭제 진행 중
0x40000000	테이블 스페이스 작성 진행 중

이 옵션을 지정한 경우 각 테이블 스페이스에 대한 다음 추가 정보가 제공됩니다.

- 총 페이지 수
- 사용할 수 있는 페이지 수
- 사용된 페이지 수
- 여유 페이지 수
- 상위 워터 마크(water mark)(페이지 수)
- 페이지 크기(바이트)
- Extent 크기(페이지 수)
- 프리페치 크기(페이지 수)
- 컨테이너 수
- 최소 복구 시간(0이 아닌 경우에만 표시됨)
- 상태 변경 테이블 스페이스 ID(테이블 스페이스 상태가 "로드 보류" 또는 "삭제 보류"인 경우에만 표시됨)
- 상태 변경 오브젝트 ID(테이블 스페이스 상태가 "로드 보류" 또는 "삭제 보류"인 경우에만 표시됨)

- Quiescer 수(테이블 스페이스 상태가 "Quiesce 상태: SHARE", "Quiesce 상태: UPDATE" 또는 "Quiesce 상태: EXCLUSIVE"인 경우에만 표시됨)
- 각 Quiescer의 테이블 스페이스 ID 및 오브젝트 ID(Quiescer 수가 0보다 큰 경우에만 표시됨)

**예:**

다음은 LIST TABLESPACES SHOW DETAIL의 두 가지 샘플 출력입니다.

```

현재 데이터베이스에 대한 테이블 스페이스
테이블 스페이스 ID          = 0
이름                       = SYSCATSPACE
유형                       = 데이터베이스 관리 스페이스
내용                       = 모든 데이터
상태                       = 0x0000
  세부사항 설명:
    정상
전체 페이지 수            = 895
사용 가능한 페이지 수    = 895
사용된 페이지 수         = 895
사용 가능한 페이지 수    = 적용되지 않음
상위 워터 마크(water mark)(페이지) = 적용되지 않음
페이지 크기(바이트)      = 4096
Extent 크기(페이지)      = 32
프리페치 크기(페이지)   = 32
컨테이너 수              = 1

테이블 스페이스 ID          = 1
이름                       = TEMPSPACE1
유형                       = 시스템 관리 스페이스
내용                       = 임시 데이터
상태                       = 0x0000
  세부사항 설명:
    정상
전체 페이지 수            = 1
사용 가능한 페이지 수    = 1
사용된 페이지 수         = 1
사용 가능한 페이지 수    = 적용되지 않음
상위 워터 마크(water mark)(페이지) = 적용되지 않음
페이지 크기(바이트)      = 4096
Extent 크기(페이지)      = 32
프리페치 크기(페이지)   = 32
컨테이너 수              = 1

테이블 스페이스 ID          = 2
이름                       = USERSPACE1
유형                       = 데이터베이스 관리 스페이스
내용                       = 모든 데이터
상태                       = 0x000c
  세부사항 설명:
    Quiesce됨: EXCLUSIVE
    로드 보류
전체 페이지 수            = 337
사용 가능한 페이지 수    = 337
사용된 페이지 수         = 337
사용 가능한 페이지 수    = 적용되지 않음
상위 워터 마크(water mark)(페이지) = 적용되지 않음
페이지 크기(바이트)      = 4096
Extent 크기(페이지)      = 32
프리페치 크기(페이지)   = 32
컨테이너 수              = 1
상태 변경 테이블 스페이스 ID = 2
상태 변경 오브젝트 ID     = 3

```

```

Quiescer 수 = 1
  Quiescer 1:
    테이블 스페이스 ID = 2
    오브젝트 ID = 3
파티션된 데이터베이스 서버 환경에서는 현재 노드의 테이블 스페이스만 나열됩니다.

```

```

현재 데이터베이스에 대한 테이블 스페이스
테이블 스페이스 ID = 0
이름 = SYSCATSPACE
유형 = 시스템 관리 스페이스
내용 = 모든 데이터
상태 = 0x0000
  세부사항 설명:
    정상
  전체 페이지 수 = 1200
  사용 가능한 페이지 수 = 1200
  사용된 페이지 수 = 1200
  사용 가능한 페이지 수 = 적용되지 않음
  상위 워터 마크(water mark)(페이지) = 적용되지 않음
  페이지 크기(바이트) = 4096
  Extent 크기(페이지) = 32
  프리페치 크기(페이지) = 32
  컨테이너 수 = 1

```

```

테이블 스페이스 ID = 1
이름 = TEMPSPACE1
유형 = 시스템 관리 스페이스
내용 = 임시 데이터
상태 = 0x0000
  세부사항 설명:
    정상
  전체 페이지 수 = 1
  사용 가능한 페이지 수 = 1
  사용된 페이지 수 = 1
  사용 가능한 페이지 수 = 적용되지 않음
  상위 워터 마크(water mark)(페이지) = 적용되지 않음
  페이지 크기(바이트) = 4096
  Extent 크기(페이지) = 32
  프리페치 크기(페이지) = 32
  컨테이너 수 = 1

```

```

테이블 스페이스 ID = 2
이름 = USERSPACE1
유형 = 시스템 관리 스페이스
내용 = 모든 데이터
상태 = 0x0000
  세부사항 설명:
    정상
  전체 페이지 수 = 1
  사용 가능한 페이지 수 = 1
  사용된 페이지 수 = 1
  사용 가능한 페이지 수 = 적용되지 않음
  상위 워터 마크(water mark)(페이지) = 적용되지 않음
  페이지 크기(바이트) = 4096
  Extent 크기(페이지) = 32
  프리페치 크기(페이지) = 32
  컨테이너 수 = 1

```

```

테이블 스페이스 ID = 3
이름 = DMS8K
유형 = 데이터베이스 관리 스페이스
내용 = 모든 데이터
상태 = 0x0000
  세부사항 설명:
    정상
  전체 페이지 수 = 2000
  사용 가능한 페이지 수 = 1952

```



```

사용된 페이지 수                = 96
사용 가능한 페이지 수          = 1856
상위 워터 마크(water mark)(페이지)      = 96
페이지 크기(바이트)            = 8192
Extent 크기(페이지)            = 32
프리페치 크기(페이지)         = 32
컨테이너 수                     = 2

테이블 스페이스 ID            = 4
이름                            = TEMP8K
유형                            = 시스템 관리 스페이스
내용                            = 임시 데이터
상태                            = 0x0000
세부사항 설명:
정상
전체 페이지 수                = 1
사용 가능한 페이지 수          = 1
사용된 페이지 수              = 1
사용 가능한 페이지 수          = 적용되지 않음
상위 워터 마크(water mark)(페이지)      = 적용되지 않음
페이지 크기(바이트)            = 8192
Extent 크기(페이지)            = 32
프리페치 크기(페이지)         = 32
컨테이너 수                     = 1

```

파티션된 데이터베이스 서버 환경에서는 현재 노드의 테이블 스페이스만 나열됩니다.

## 사용 시 참고사항

파티션된 데이터베이스 환경에서는 이 명령이 데이터베이스의 모든 테이블 스페이스를 리턴하지는 않습니다. 모든 테이블 스페이스 목록을 확보하려면 SYSCAT.TABLESPACES를 쿼리하십시오.

테이블 스페이스 재조정 중에 사용 가능한 페이지 수에는 새로 추가된 컨테이너 페이지가 포함되지만 이 새 페이지는 재조정이 완료될 때까지 사용 가능한 페이지 수에 반영되지 않습니다. 테이블 스페이스 재조정이 진행 중 상태가 아니면 사용된 페이지 수 + 여유 페이지 수가 사용할 수 있는 페이지 수와 동일하게 됩니다.

현재 IBM DB2 데이터베이스 제품이 지원하는 최소 25개 테이블 또는 테이블 스페이스가 있습니다. 이들 상태는 특정 상황에서 데이터에 대한 액세스를 제어하거나 특정 사용자 조치를 유추하거나 필요한 경우 데이터베이스의 무결성을 보호하는 데 사용됩니다. 대부분은 로드 유틸리티 또는 백업 및 리스토어 유틸리티 같은 DB2 유틸리티 중 하나의 조작과 관련된 이벤트의 결과입니다.

다음 표는 지원되는 각 테이블 스페이스 상태를 설명합니다. 또한 데이터베이스를 관리하는 중에 발생할 수 있는 상태를 해석하고 응답하는 방법을 정확하게 보여주는 작업 예를 제공합니다. 예는 AIX에서 실행된 명령 스크립트에서 제공됩니다. 사용자가 직접 복사, 붙여넣기 및 실행할 수 있습니다. UNIX가 아닌 시스템에서 DB2 데이터베이스 제품을 실행 중인 경우 모든 경로가 시스템에 맞는 형식인지 확인하십시오. 대부분의 예는 DB2 데이터베이스 제품과 함께 제공되는 SAMPLE 데이터베이스의 테이블을 기초로 합니다. 몇 가지 예는 SAMPLE 데이터베이스의 파트가 아닌 시나리오가 필요하지만, SAMPLE 데이터베이스에 대한 연결을 시작점으로 사용할 수 있습니다.

표 18. 지원되는 테이블 스페이스 상태

상태	16진수 상태 값	설명	예:
백업 보류	0x20	<p>테이블 스페이스는 특정 시점 테이블 스페이스 롤 포워드 작업 후, 또는 COPY NO 옵션을 지정하는 로드 작업(복구 가능한 데이터베이스에 대한) 후 이 상태가 됩니다. 테이블 스페이스(또는 전체 데이터베이스)는 테이블 스페이스를 사용하기 전에 백업해야 합니다. 테이블 스페이스를 백업하지 않은 경우, 테이블 스페이스 내의 테이블은 쿼리할 수 있지만 갱신할 수는 없습니다.</p> <p>주: 데이터베이스는 또한 롤 포워드 복구에 사용 가능하도록 설정된 후 즉시 백업해야 합니다. 데이터베이스는 <b>logretain</b> 데이터베이스 구성 매개변수를 RECOVERY로 설정하거나, <b>userexit</b> 데이터베이스 구성 매개변수를 YES로 설정한 경우에 복구할 수 있습니다. 데이터베이스가 백업될 때까지 (<b>backup_pending</b> 정보용 데이터베이스 구성 매개변수 값이 NO로 설정될 때) 데이터베이스를 활성화하거나 연결할 수 없습니다.</p>	<p>1. 다음 콘텐츠가 있는 로드 입력 파일 staff_data.del을 가정합니다.</p> <pre>11,"Melnyk",20,"Sales",10,70000,15000: update db cfg for sample using logretain recovery; backup db sample; connect to sample; load from staff_data.del of del messages load.msg insert into staff copy no; update staff set salary = 69000 where id = 11;</pre> <p>2.</p> <pre>update db cfg for sample using logretain recovery; connect to sample;</pre>
백업 진행 중	0x800	<p>백업 작업 중에만 적용되는 임시 상태입니다.</p>	<p>온라인 BACKUP DATABASE 명령을 발행하십시오.</p> <pre>backup db sample online;</pre> <p>백업 작업이 실행 중인 동안 다른 세션에서 다음 스크립트를 실행하십시오.</p> <pre>connect to sample;</pre> <p>1.</p> <pre>list tablespaces show detail;</pre> <p>또는</p> <p>2.</p> <pre>get snapshot for tablespaces on sample; connect reset;</pre> <p>USERSPACE1에 대해 리턴되는 정보는 테이블 스페이스가 백업 진행 중 상태임을 표시합니다.</p>

표 18. 지원되는 테이블 스페이스 상태 (계속)

상태	16진수 상태 값	설명	예:
DMS 재조정 진행 중	0x10000000	데이터 재조정 작업 중에만 적용되는 임시 상태입니다. 데이터베이스 관리 스페이스(DMS)로 정의된 테이블 스페이스에 새 컨테이너가 추가되거나 기존 컨테이너가 확장되는 경우, 테이블 스페이스 데이터의 재조정이 발생할 수 있습니다. 재조정은 데이터를 스트라이프된 상태로 보존하려고 할 때 하나의 위치에서 다른 위치로 테이블 스페이스 Extent를 이동하는 프로세스입니다. Extent는 컨테이너 스페이스 단위(페이지 수로 측정)이며, 스트라이프는 테이블 스페이스에 대한 컨테이너 세트 사이의 Extent 계층입니다.	<p>상당한 양의 데이터(예를 들어, 20000 이상의 레코드)를 갖는 로드 입력 파일 staffdata.del을 가정합니다.</p> <pre>connect to sample; create tablespace ts1 managed by database using (file '/home/melnyk/melnyk/NODE00000/SQL00001/ts1c1' 1024); create table newstaff like staff in ts1; load from staffdata.del of del insert into newstaff nonrecoverable; alter tablespace ts1 add (file '/home/melnyk/melnyk/NODE00000/SQL00001/ts1c2' 1024); list tablespaces; connect reset;</pre> <p>TS1에 대해 리턴되는 정보는 테이블 스페이스가 DMS 재조정 진행 중 상태임을 표시합니다.</p>
작동 불가능화 보류	0x200	테이블 스페이스는 데이터베이스 롤 포워드 작업 중에 이 상태가 될 수 있으며, 롤 포워드 작업을 종료하여 더 이상 이 상태에 있으면 안됩니다. 상태는 테이블 스페이스가 오프라인이 되어 트랜잭션에 대한 보상 로그 레코드가 기록되지 않도록 하는 조건으로 트리거됩니다. 이 테이블 스페이스 상태의 출현 및 후속 소멸은 사용자에게 표시되지 않습니다.	이 테이블 스페이스 상태를 보여주는 예는 이 문서의 범위를 벗어납니다.
삭제 보류	0x8000	해당 컨테이너 중 하나 이상이 데이터베이스 재시작 작업 중에 문제점이 있는 것으로 발견되는 경우 테이블 스페이스는 이 상태에 있습니다. (이 데이터베이스가 있는 이전 세션이 비정상적으로 종료된 경우(예: 전원 고장 시) 데이터베이스를 재시작해야 합니다.) 테이블 스페이스가 삭제 보류 상태에 있으면 사용할 수 없게 되고 삭제만 가능합니다.	이 테이블 스페이스 상태를 보여주는 예는 이 문서의 범위를 벗어납니다.

표 18. 지원되는 테이블 스페이스 상태 (계속)

상태	16진수 상태 값	설명	예:
Load in Progress	0x20000	COPY NO 옵션을 지정하는 로드 작업(복구 가능한 데이터 베이스에 대한) 중에만 적용되는 임시 상태입니다. 로드 진행 중 테이블 상태도 참조하십시오.	<p>상당한 양의 데이터(예를 들어, 20000 이상의 레코드)를 갖는 로드 입력 파일 staffdata.del이 있는 경우,</p> <pre>update db cfg for sample using logretain recovery; backup db sample; connect to sample; create table newstaff like staff; load from staffdata.del of del insert into newstaff copy no; connect reset;</pre> <p>로드 조치가 실행 중인 동안 다른 세션에서 다음 스크립트를 실행하십시오.</p> <pre>connect to sample; list tablespaces; connect reset;</pre> <p>USERSPACE1에 대해 리턴되는 정보는 테이블 스페이스가 로드 진행 중(및 백업 보류) 상태임을 표시합니다.</p>
일반	0x0	테이블 스페이스 상태가 다른 (비정상) 테이블 스페이스 상태가 아닌 경우, 테이블 스페이스는 일반 상태입니다. 일반 상태는 테이블 스페이스가 작성된 후 초기 상태입니다.	<pre>connect to sample; create tablespace ts1 managed by database using (file '/home/melnyk/melnyk/NODE0000/SQL00001 /tsc1' 1024); list tablespaces show detail;</pre>
오프라인 및 액세스 불가능	0x4000	해당 컨테이너 중 하나 이상에 문제점이 있는 경우 테이블 스페이스는 이 상태에 있습니다. 컨테이너가 의도하지 않게 이름이 바뀌거나 이동 또는 손상되었을 수 있습니다. 문제점이 수정되고, 테이블 스페이스와 연관되는 컨테이너에 다시 액세스할 수 있으면, 데이터베이스에서 모든 응용프로그램의 연결을 끊은 후 다시 데이터베이스에 연결하여 이 비정상 상태를 제거할 수 있습니다. 또는 SWITCH ONLINE 절을 지정하는 ALTER TABLESPACE 문을 발행하여 데이터베이스에서 다른 응용프로그램의 연결을 끊지 않고 테이블 스페이스에서 오프라인 및 액세스 불가능 상태를 제거할 수도 있습니다.	<pre>connect to sample; create tablespace ts1 managed by database using (file '/home/melnyk/melnyk/NODE0000/SQL00001 /tsc1' 1024); alter tablespace ts1 add (file '/home/melnyk/melnyk /NODE0000/SQL00001/tsc2' 1024); export to st_data.del of del select * from staff; create table stafftemp like staff in ts1; import from st_data.del of del insert into stafftemp; connect reset;</pre> <p>테이블 스페이스 컨테이너 이름을 tsc1에서 tsc3으로 바꾼 후 STAFFTEMP 테이블을 쿼리하십시오.</p> <pre>connect to sample; select * from stafftemp;</pre> <p>쿼리는 SQL0290N(테이블 스페이스 액세스가 허용되지 않음)을 리턴하고, LIST TABLESPACES 명령은 TS1에 대해 0x4000(오프라인 및 액세스 불가능) 상태 값을 리턴합니다. 테이블 스페이스 컨테이너 이름을 tsc3에서 다시 tsc1로 바꾸십시오. 이제 쿼리가 제대로 실행됩니다.</p>

표 18. 지원되는 테이블 스페이스 상태 (계속)

상태	16진수 상태 값	설명	예:
Quiesce 상태 독점	0x4	테이블 스페이스 Quiesce 함수를 호출하는 응용프로그램이 테이블 스페이스에 대해 독점 (읽기 또는 쓰기) 액세스를 가지고 있는 경우 테이블 스페이스는 이 상태에 있습니다. QUIESCE TABLESPACES FOR TABLE 명령을 발행하여 테이블 스페이스가 명시적으로 Quiesce 상태 독점 상태가 되도록 할 수 있습니다.	<p>Quiesce 상태 독점으로 설정하기 전에 테이블 스페이스 상태가 일반 상태인지 확인하십시오.</p> <pre>connect to sample; quiesce tablespaces for table staff reset; quiesce tablespaces for table staff exclusive; connect reset;</pre> <p>다른 세션에서 다음 스크립트를 실행하십시오.</p> <pre>connect to sample; select * from staff where id=60; update staff set salary=50000 where id=60; list tablespaces; connect reset;</pre> <p>USERSPACE1에 대해 리턴되는 정보는 테이블 스페이스가 Quiesce 상태 독점 상태임을 표시합니다.</p>
Quiesce 상태 공유	0x1	테이블 스페이스 Quiesce 함수를 호출하는 응용프로그램과 동시 응용프로그램이 테이블 스페이스에 대해 읽기(쓰기는 아님) 액세스를 가지고 있는 경우 테이블 스페이스는 이 상태에 있습니다. QUIESCE TABLESPACES FOR TABLE 명령을 발행하여 테이블 스페이스가 명시적으로 Quiesce 상태 공유 상태가 되도록 할 수 있습니다.	<p>Quiesce 상태 공유로 설정하기 전에 테이블 스페이스 상태가 일반 상태인지 확인하십시오.</p> <pre>connect to sample; quiesce tablespaces for table staff reset; quiesce tablespaces for table staff share; connect reset;</pre> <p>다른 세션에서 다음 스크립트를 실행하십시오.</p> <pre>connect to sample; select * from staff where id=40; update staff set salary=50000 where id=40; list tablespaces; connect reset;</pre> <p>USERSPACE1에 대해 리턴되는 정보는 테이블 스페이스가 Quiesce 상태 공유 상태임을 표시합니다.</p>
Quiesce 상태 갱신	0x2	테이블 스페이스 Quiesce 함수를 호출하는 응용프로그램이 테이블 스페이스에 대해 독점 쓰기 액세스를 가지고 있는 경우 테이블 스페이스는 이 상태에 있습니다. QUIESCE TABLESPACES FOR TABLE 명령을 발행하여 테이블 스페이스가 명시적으로 Quiesce 상태 갱신 상태가 되도록 할 수 있습니다.	<p>Quiesce 상태 갱신으로 설정하기 전에 테이블 스페이스 상태가 일반 상태인지 확인하십시오.</p> <pre>connect to sample; quiesce tablespaces for table staff reset; quiesce tablespaces for table staff intent to update; connect reset;</pre> <p>다른 세션에서 다음 스크립트를 실행하십시오.</p> <pre>connect to sample; select * from staff where id=50; update staff set salary=50000 where id=50; list tablespaces; connect reset;</pre> <p>USERSPACE1에 대해 리턴되는 정보는 테이블 스페이스가 Quiesce 상태 갱신 상태임을 표시합니다.</p>

표 18. 지원되는 테이블 스페이스 상태 (계속)

상태	16진수 상태 값	설명	예:
Reorg 진행 중	0x400	Reorg 작업 중에만 적용되는 임시 상태입니다.	<p>REORG TABLE 명령을 발행합니다.</p> <pre>connect to sample; reorg table staff; connect reset;</pre> <p>Reorg 작업이 실행 중인 동안 다른 세션에서 다음 스크립트를 실행하십시오.</p> <pre>connect to sample;</pre> <p>1.</p> <pre>list tablespaces show detail;</pre> <p>또는</p> <p>2.</p> <pre>get snapshot for tablespaces on sample; connect reset;</pre> <p>USERSPACE1에 대해 리턴되는 정보는 테이블 스페이스가 Reorg 진행 중 상태임을 표시합니다.</p> <p>주: SAMPLE 데이터베이스와 관련된 테이블 재구성 작업은 짧은 시간 안에 완료되므로, 결국 이 접근 방식으로 Reorg 진행 중 상태를 조사하는 것은 어려울 수 있습니다.</p>
리스트어 보류	0x100	경로 재지정된 리스트어 조作的 첫 번째 파트 후(즉, SET TABLESPACE CONTAINERS 명령이 발행되기 전) 데이터베이스의 테이블 스페이스는 이 상태가 됩니다. 테이블 스페이스(또는 전체 데이터베이스)는 테이블 스페이스를 사용하기 전에 리스트어해야 합니다. 리스트어 조작이 완료될 때까지( <b>restore_pending</b> 정보용 데이터베이스 구성 매개변수 값이 NO로 설정될 때) 데이터베이스에 연결할 수 없습니다.	'스톰리지를 정의할 수 있음' 상태에 있는, 경로 재지정된 리스트어 조作的 첫 번째 파트가 완료될 때, 모든 테이블 스페이스는 리스트어 보류 상태가 됩니다.

표 18. 지원되는 테이블 스페이스 상태 (계속)

상태	16진수 상태 값	설명	예:
리스토어 진행 중	0x2000	리스토어 조작 중에만 적용되는 임시 상태입니다.	<pre>update db cfg for sample using logretain recovery; backup db sample; backup db sample tablespace (userspace1);</pre> <p>이 백업 이미지의 시간소인은 다음과 같습니다.</p> <p>20040611174124</p> <pre>restore db sample tablespace (userspace1) online taken at 20040611174124;</pre> <p>리스토어 조작이 실행 중인 동안 다른 세션에서 다음 스크립트를 실행하십시오.</p> <pre>connect to sample;</pre> <ol style="list-style-type: none"> <li>list tablespaces show detail;</li> </ol> <p>또는</p> <ol style="list-style-type: none"> <li>get snapshot for tablespaces on sample;           connect reset;</li> </ol> <p>USERSPACE1에 대해 리턴되는 정보는 테이블 스페이스가 리스토어 진행 중 상태임을 표시합니다.</p>
롤 포워드 보류	0x80	테이블 스페이스는 복구 가능한 데이터베이스에 대한 리스토어 조작 후에 이 상태가 됩니다. 테이블 스페이스(또는 전체 데이터베이스)는 테이블 스페이스를 사용하기 전에 롤 포워드해야 합니다. 데이터베이스는 <b>logretain</b> 데이터베이스 구성 매개변수를 RECOVERY로 설정하거나, <b>userexit</b> 데이터베이스 구성 매개변수를 YES로 설정한 경우에 복구할 수 있습니다. 롤 포워드 작업이 완료될 때까지( <b>rollfwd_pending</b> 정보용 데이터베이스 구성 매개변수 값이 NO로 설정될 때) 데이터베이스를 활성화거나 데이터베이스에 연결할 수 없습니다.	리스토어 진행 중 상태의 온라인 테이블 스페이스 리스토어 조작이 완료되면, 테이블 스페이스 USERSPACE1은 롤 포워드 보류 상태가 됩니다.

표 18. 지원되는 테이블 스페이스 상태 (계속)

상태	16진수 상태 값	설명	예:
롤 포워드 진행 중	0x40	롤 포워드 작업 중에만 적용되는 임시 상태입니다.	<p>상당한 양의 데이터(예를 들어, 20000 이상의 레코드)를 갖는 로드 입력 파일 staffdata.del이 있는 경우,</p> <pre>update db cfg for sample using logretain recovery; backup db sample; connect to sample; create tablespace ts1 managed by database using (file '/home/melnyk/melnyk/NODE0000/SQL00001 /ts1c1' 1024); create table newstaff like staff in ts1; connect reset; backup db sample tablespace (ts1) online;</pre> <p>이 백업 이미지의 시간소인은 다음과 같습니다.</p> <p>20040630000715</p> <pre>connect to sample; load from staffdata.del of del insert into newstaff copy yes to /home/melnyk/backups; connect reset; restore db sample tablespace (ts1) online taken at 20040630000715; rollforward db sample to end of logs and stop tablespace (ts1) online;</pre> <p>롤 포워드 작업이 실행 중인 동안 다른 세션에서 다음 스크립트를 실행하십시오.</p> <pre>connect to sample;</pre> <p>1.</p> <pre>list tablespaces show detail;</pre> <p>또는</p> <p>2.</p> <pre>get snapshot for tablespaces on sample; connect reset;</pre> <p>TS1에 대해 리턴되는 정보는 테이블 스페이스가 롤 포워드 진행 중 상태임을 표시합니다.</p>
스토리지를 정의할 수 있음	0x2000000	경로 재지정된 리스토어 조作的 첫 번째 파트 후(즉, SET TABLESPACE CONTAINERS 명령이 발행되기 전) 데이터베이스의 테이블 스페이스는 이 상태가 됩니다. 원하면 컨테이너를 재정의할 수 있습니다.	<pre>backup db sample;</pre> <p>이 백업 이미지의 시간소인이 20040613204955인 것을 가정합니다.</p> <pre>restore db sample taken at 20040613204955 redirect; list tablespaces;</pre> <p>LIST TABLESPACES 명령에 의해 리턴되는 정보는 모든 테이블 스페이스가 '스토리지를 정의할 수 있음' 및 '리스토어 보류' 상태에 있음을 표시합니다.</p>



표 18. 지원되는 테이블 스페이스 상태 (계속)

상태	16진수 상태 값	설명	예:
스토리지를 정의해야 함	0x1000	테이블 스페이스 컨테이너 설정 단계가 생략되거나 테이블 스페이스 컨테이너 설정 단계 중에 지정된 컨테이너를 획득할 수 없는 경우 새 데이터베이스 경로 재지정된 리스토어 조작 동안 데이터베이스의 테이블 스페이스는 이 상태가 됩니다. 예를 들어 유효하지 않은 경로 이름을 지정한 경우 나중의 경우가 발생할 수 있으며, 그렇지 않으면 디스크 스페이스가 충분하지 않습니다.	<pre>backup db sample;</pre> <p>이 백업 이미지의 시간소인이 20040613204955인 것을 가정합니다.</p> <pre>restore db sample taken at 20040613204955 into mydb redirect; set tablespace containers for 2 using (path 'ts2c1'); list tablespaces;</pre> <p>LIST TABLESPACES 명령에 의해 리턴되는 정보는 테이블 스페이스 SYSCATSPACE 및 테이블 스페이스 TEMPSPACE1이 스토리지를 정의해야 함, 스토리지를 정의할 수 있음 및 리스토어 보류 상태에 있음을 표시합니다. 스토리지를 정의해야 함 상태가 스토리지를 정의할 수 있음 상태보다 우선합니다.</p>
테이블 스페이스 작성 진행 중	0x40000000	테이블 스페이스 작성 작업 중에만 적용되는 임시 상태입니다.	<pre>connect to sample; create tablespace ts1 managed by database using (file '/home/melnyk/melnyk/NODE0000/SQL00001/tsc1' 1024); create tablespace ts2 managed by database using (file '/home/melnyk/melnyk/NODE0000/SQL00001/tsc2' 1024); create tablespace ts3 managed by database using (file '/home/melnyk/melnyk/NODE0000/SQL00001/tsc3' 1024);</pre> <p>테이블 스페이스 작성 작업이 실행 중인 동안 다른 세션에서 다음 스크립트를 실행하십시오.</p> <pre>connect to sample;</pre> <ol style="list-style-type: none"> <li> <pre>list tablespaces show detail;</pre> <p>또는</p> </li> <li> <pre>get snapshot for tablespaces on sample; connect reset;</pre> </li> </ol> <p>TS1, TS2 및 TS3에서 리턴되는 정보는 테이블 스페이스가 테이블 스페이스 작성 진행 중 상태에 있음을 나타냅니다.</p>

표 18. 지원되는 테이블 스페이스 상태 (계속)

상태	16진수 상태 값	설명	예:
테이블 스페이스 삭제 진행 중	0x20000000	테이블 스페이스 삭제 작업 중에만 적용되는 임시 상태입니다.	<pre>connect to sample; create tablespace ts1 managed by database using (file '/home/melnyk/melnyk/NODE0000/SQL00001 /tsc1' 1024); create tablespace ts2 managed by database using (file '/home/melnyk/melnyk/NODE0000/SQL00001 /tsc2' 1024); create tablespace ts3 managed by database using (file '/home/melnyk/melnyk/NODE0000/SQL00001 /tsc3' 1024); drop tablespace ts1; drop tablespace ts2; drop tablespace ts3;</pre> <p>테이블 스페이스 삭제 작업이 실행 중인 동안 다른 세션에서 다음 스크립트를 실행하십시오.</p> <pre>connect to sample;</pre> <ol style="list-style-type: none"> <li> <pre>list tablespaces show detail;</pre> <p>또는</p> </li> <li> <pre>get snapshot for tablespaces on sample; connect reset;</pre> </li> </ol> <p>TS1, TS2 및 TS3에서 리턴되는 정보는 테이블 스페이스가 테이블 스페이스 삭제 진행 중 상태에 있음을 나타냅니다.</p>

테이블 스페이스 상태에 대한 추가 정보는 [관련 링크](#) 섹션을 참조하십시오.

## 제 84 장 LIST UTILITIES

인스턴스에 대한 활성 유틸리티 목록을 표준 출력에 표시합니다. 각 유틸리티의 설명에는 시작 시간, 설명, 조정 우선순위(적용 가능한 경우)와 진행 모니터링 정보(적용 가능한 경우)와 같은 속성이 포함될 수 있습니다.

### 범위

이 명령은 모든 데이터베이스 파티션에 대한 정보를 리턴합니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*
- *sysmon*

### 필수 연결

인스턴스

### 명령 구문

```

>> LIST UTILITIES _____ <<
      | SHOW DETAIL |
  
```

### 명령 매개변수

#### SHOW DETAIL

진행 모니터링을 지원하는 유틸리티에 대한 자세한 진행 정보를 표시합니다.

#### 예:

some\_table 테이블에 대한 RUNSTATS 호출:

LIST UTILITIES

```

ID              = 1
유형            = RUNSTATS
데이터베이스 이름 = PROD
설명            = krrose.some_table
시작 시간       = 12/19/2003 11:54:45.773215
우선순위        = 10
  
```

오프라인 데이터베이스 백업의 성능 모니터링:

LIST UTILITIES SHOW DETAIL

ID	= 2
유형	= BACKUP
데이터베이스 이름	= SAMPLE
설명	= offline db
시작 시간	= 10/30/2003 12:55:31.786115
우선순위	= 0
진행 모니터링:	
단계 번호[현재]	= 1
설명	=
작업 메트릭	= BYTES
총 작업 단위	= 20232453
완료된 작업 단위	= 230637
시작 시간	= 10/30/2003 12:55:31.786115

## 사용 시 참고사항

실행 중인 유틸리티의 상태를 모니터링하려면 이 명령을 사용하십시오. 예를 들어, 이 유틸리티를 사용하여 온라인 백업 진행을 모니터링할 수 있습니다. 다른 예로는, 실행 중인 유틸리티를 판별하기 위해 이 명령을 사용하여 성능 문제점을 조사할 수 있습니다. 유틸리티로 인해 성능이 저하되는 경우 유틸리티를 조절할 것을 선택할 수 있습니다(유틸리티가 조절을 지원하는 경우). LIST UTILITIES 명령의 ID는 SET UTIL\_IMPACT\_PRIORITY 명령에서 사용되는 동일한 ID입니다.

---

## 제 85 장 LOAD

DB2 테이블에 데이터를 로드합니다. 서버에 있는 데이터는 파일, 테이프 또는 Named Pipe 양식이 될 수 있습니다. 리모트로 연결된 클라이언트에 있는 데이터는 완전한 파일 또는 Named Pipe 양식이 될 수 있습니다. 또한 데이터는 사용자 정의 커서에서 로드되거나 사용자가 작성한 스크립트나 응용프로그램을 사용하여 로드될 수 있습니다. 테이블의 COMPRESS 속성이 YES로 설정된 경우, 테이블의 XML 스토리지 오브젝트의 데이터를 포함하여 이미 사전이 테이블에 존재하는 모든 데이터 및 데이터베이스 파티션에서 로드된 데이터가 압축됩니다.

420 페이지의 『로드 유틸리티의 파일 유형 수정자』로의 빠른 링크.

### 제한사항

로드 유틸리티는 계층 구조 레벨에서 데이터 로드를 지원하지 않습니다. 로드 유틸리티는 범위로 클러스터된 테이블과 호환 가능하지 않습니다.

### 범위

이 명령은 단일 요청으로 다중 데이터베이스 파티션에 대해 실행할 수 있습니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *dataaccess*
- 데이터베이스의 LOAD 권한 및
  - 로드 유틸리티가 INSERT 모드, TERMINATE 모드 (이전 로드 삽입 작업을 종료하기 위해) 또는 RESTART 모드(이전 로드 삽입 작업을 재시작하기 위해)에서 호출될 때 테이블에 대한 INSERT 특권
  - 로드 유틸리티가 REPLACE 모드, TERMINATE 모드(이전 로드 삽입 작업을 종료하기 위해) 또는 RESTART 모드(이전 로드 삽입 작업을 재시작하기 위해)에서 호출될 때 테이블에 대한 INSERT 및 DELETE 특권
  - 이 테이블이 로드 작업의 일부로 사용되는 경우 예외 테이블에 대한 INSERT 특권.
- 보호 설정된 컬럼이 있는 테이블에 데이터를 로드하려면 세션 권한 부여 ID에 테이블의 모든 보호 컬럼에 대해 쓰기 액세스를 허용하는 LBAC 증명서가 있어야 합니다. 그렇지 않으면 로드에 실패하고 오류(SQLSTATE 5U014)가 리턴됩니다.
- 보호 설정된 행이 있는 테이블에 데이터를 로드하려면 세션 권한 부여 ID는 다음 기준에 부합되는 보안 레이블을 보유해야 합니다.

- 테이블을 보호하는 보안 규정에 포함됨
- 세션 권한 부여 ID에 쓰기 액세스 또는 모든 액세스 권한이 부여되었습니다.

세션 권한 부여 ID에 이러한 보안 레이블이 없으면 로드 실패하고 오류(SQLSTATE 5U014)가 리턴됩니다. 이 보안 레벨은 세션 권한 부여 ID의 LBAC 증명서가 데이터의 해당 행을 보호하는 보안 레벨에 쓰기를 허용하지 않는 경우 로드된 행을 보호하는 데 사용됩니다. 그러나 테이블을 보호하는 보안 규정이 CREATE SECURITY POLICY 명령문의 RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL 옵션을 사용하여 작성되는 경우에는 발생하지 않습니다. 이러한 경우, 로드 실패하고 오류(SQLSTATE 42519)가 리턴됩니다.

- REPLACE 옵션이 지정되어 있으면 세션 권한 부여 ID에 테이블을 삭제할 수 있는 권한이 있어야 합니다.
- LOCK WITH FORCE 옵션이 지정된 경우, SYSADM 권한이 필요합니다.

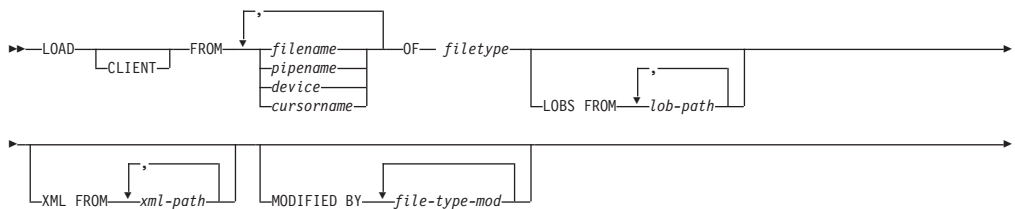
모든 로드 프로세스(및 일반적으로 모든 DB2 서버 프로세스)는 인스턴스 소유자가 소유하고 이들 모든 프로세스는 필요한 파일에 액세스하기 위해 인스턴스 소유자의 식별을 사용하므로 인스턴스 소유자는 입력 데이터 파일에 대한 읽기 액세스가 있어야 합니다. 이러한 입력 데이터는 명령 호출자와 상관없이 인스턴스 소유자가 읽을 수 있어야 합니다.

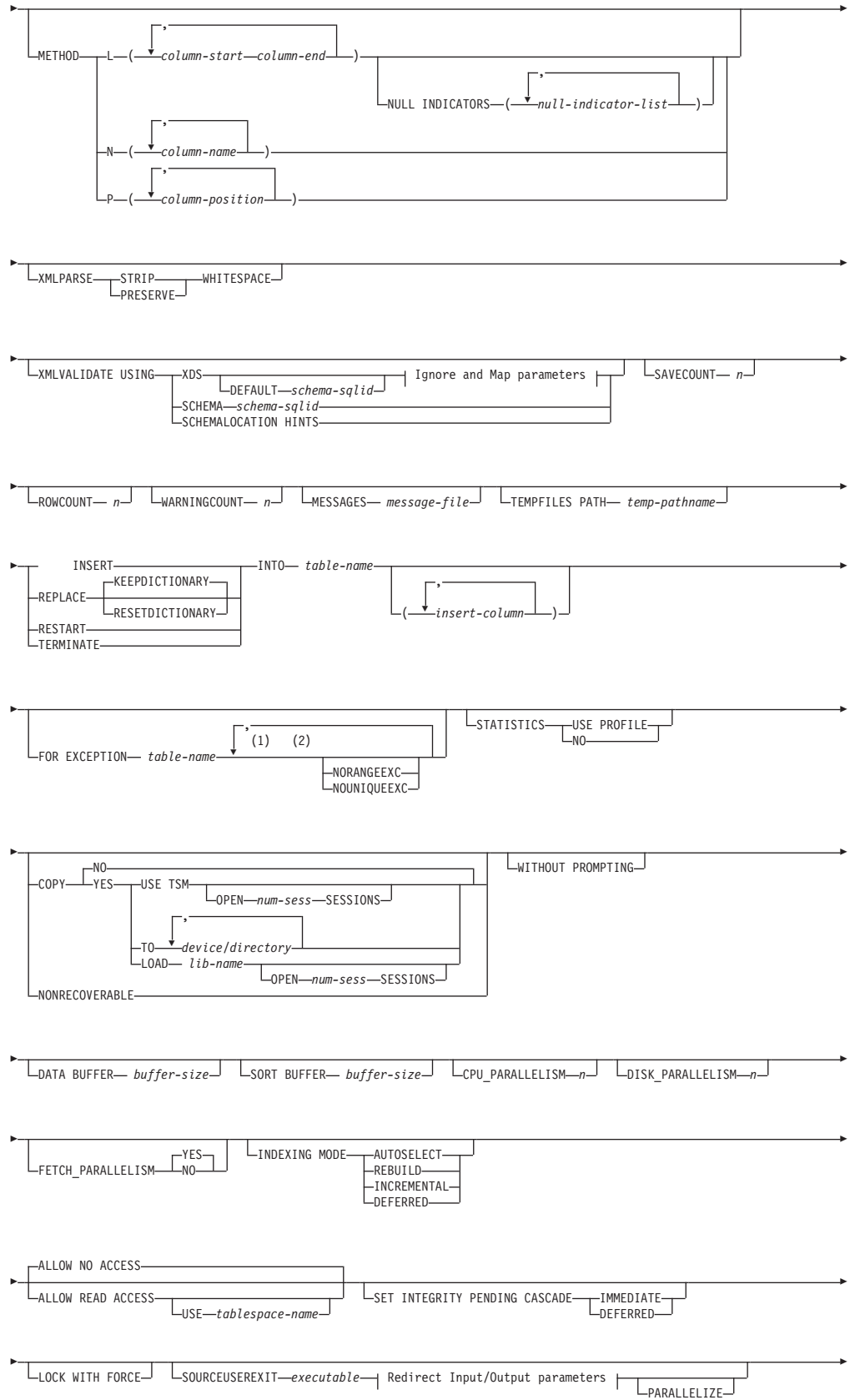
### 필수 연결

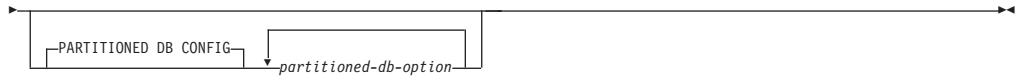
데이터베이스 내재된 연결이 사용 가능한 경우 디폴트 데이터베이스에 연결됩니다.

인스턴스 명시적 접속은 필요하지 않습니다. 데이터베이스에 대한 연결이 설정되면 로컬 인스턴스에 대한 내재적 접속이 시도됩니다.

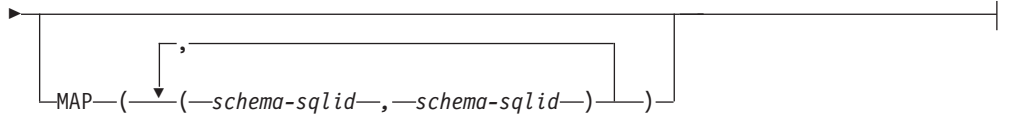
### 명령 구문



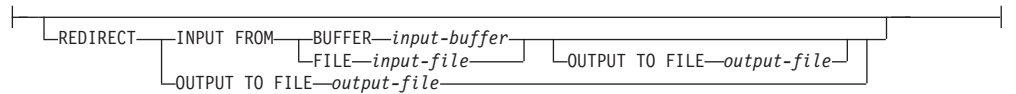




**Ignore and Map parameters:**



**Redirect Input/Output parameters:**



주:

- 1 이 키워드는 임의의 순서로 표시될 수 있습니다.
- 2 이들 각 키워드는 한 번만 표시될 수 있습니다.

**명령 매개변수**

**CLIENT**

로드될 데이터가 리모트로 연결된 클라이언트에 위치하도록 지정합니다. 이 옵션은 리모트 클라이언트에서 로드 조작을 호출하지 않는 경우 무시됩니다. 이 옵션은 CURSOR 파일 유형과 함께 지정된 경우 무시됩니다.

주:

1. dumpfile 및 lobsinfile 수정자 및 XML FROM 옵션은 CLIENT 키워드가 지정된 경우라도 서버의 파일을 참조합니다.
2. 리모트 로드 조작 중에는 코드 페이지 변환이 수행되지 않습니다. 데이터의 코드 페이지가 서버의 코드 페이지와 다른 경우, 데이터 코드 페이지는 codepage 수정자를 사용하여 지정해야 합니다.

다음 예에서는 리모트로 연결된 클라이언트에 있는 데이터 파일(/u/user/data.del)이 서버 데이터베이스로 로드됩니다.

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

**FROM filename | pipename | device | cursorname**

로드 중인 데이터를 포함하는 SQL문을 참조하는 파일, 파이프, 디바이스 또는



커서를 지정합니다. 입력 소스가 파일, 파이프 또는 디바이스인 경우 CLIENT 옵션이 지정되어 있지 않았으면 데이터베이스가 있는 데이터베이스 파티션에 입력 소스가 있어야 합니다.

이름이 여러 개 지정되면 순서대로 처리됩니다. 마지막으로 지정된 항목이 테이프 디바이스이면 사용자가 다른 테이프를 입력하도록 프롬프트됩니다. 유효한 응답은 다음과 같습니다.

- c** 계속. 경고 메시지를 생성한 디바이스를 사용하여 계속합니다(예를 들어, 새로운 테이프가 마운트된 경우).
- d** 디바이스 종료. 경고 메시지를 생성한 디바이스를 사용하여 중지합니다(예를 들어, 더 이상의 테이프가 없는 경우).
- t** 종료. 모든 디바이스를 종료합니다.

주:

1. 완전한 파일 이름을 사용할 것을 권장합니다. 서버가 리모트이면 완전한 파일 이름을 사용해야 합니다. 데이터베이스가 호출자와 동일한 데이터베이스 파티션에 있으면 상대 경로를 사용할 수 있습니다.
2. *ADMIN\_CMD* 프로시저를 사용하는 *EXPORT* 명령을 사용하여 파일로 데이터를 익스포트하는 경우, 데이터 파일은 분리 사용자 ID 소유가 됩니다. 일반적으로 이 파일은 인스턴스 소유자가 액세스할 수 없습니다. CLP 또는 *ADMIN\_CMD* 프로시저에서 *LOAD*를 실행시키려면 데이터 파일이 인스턴스 소유자 ID에 의해 액세스 가능해야 하며 따라서 데이터 파일에 대한 읽기 액세스 권한이 인스턴스 소유자에게 부여되어야 합니다.
3. 파일이 실제로는 분리되어 있지만 논리적으로는 한 개인 경우, 다중 IXF 파일로부터의 데이터 로드도 지원됩니다. 파일이 논리적 및 물리적으로 분리된 경우에는 지원되지 않습니다. (다중의 실제 파일도 *EXPORT* 명령을 한번 호출하여 모두 작성된 경우 논리적으로는 한 개로 간주됩니다.)
4. 클라이언트 머신에 있는 데이터를 로드 중인 경우, 데이터는 완전한 파일 또는 Named Pipe 양식이어야 합니다.
5. 파티션된 데이터베이스 환경에서, XML 데이터를 파일에서 테이블로 로딩하는 경우에는 로드를 실행 중인 모든 데이터베이스 파티션에서 XML 데이터 파일에 읽기 액세스가 가능해야 합니다.

#### OF filetype

다음과 같은 데이터 형식을 지정합니다.

- ASC(컬럼 식별자가 없는 ASCII 형식)
- DEL(컬럼 식별자가 있는 ASCII 형식)
- IXF(Integration Exchange Format, PC 버전)는 DB2에 의해 독점 사용되는 2진 형식입니다.

- CURSOR(SELECT 또는 VALUES문에 대해 선언된 커서).

주: 분산 데이터베이스 환경에서, CURSOR 파일 유형을 사용하여 XML 데이터를 테이블로 로드하는 경우에는 PARTITION\_ONLY 및 LOAD\_ONLY 모드가 지원되지 않습니다.

#### **LOBS FROM** *lob-path*

로드할 LOB 값이 포함된 데이터 파일의 경로. 경로는 슬래시(/)로 끝나야 합니다. CLIENT 옵션이 지정되어 있으면 경로가 완전해야 합니다. LOB 데이터 파일의 이름은 기본 데이터 파일(ASC, DEL 또는 IXF)에 저장되는데 LOB 컬럼으로 로드될 컬럼에 저장됩니다. 지정할 수 있는 최대 경로 수는 999입니다. 이는 내재적으로 LOBSINFILE 동작을 활성화합니다.

이 옵션은 CURSOR 파일 유형과 함께 지정되면 무시됩니다.

#### **MODIFIED BY** *file-type-mod*

파일 유형 수정자 옵션을 지정합니다. 420 페이지의 『로드 유틸리티의 파일 유형 수정자』을 참조하십시오.

#### **METHOD**

**L** 데이터를 로드할 시작 및 끝 컬럼을 지정합니다. 컬럼 번호는 데이터 행이 시작되는 바이트 오프셋입니다. 컬럼 번호는 1부터 시작됩니다. 이 메소드는 ASC 파일에서만 사용될 수 있으며 해당 파일 유형에 유일한 유효 메소드입니다.

#### **NULL INDICATORS** *null-indicator-list*

이 옵션은 METHOD L 매개변수가 지정된 경우(입력 파일이 ASC 파일)에만 사용할 수 있습니다. 널(NULL) 표시기 목록은 각 널(NULL) 표시기 필드의 컬럼 번호를 지정하는 쉼표로 구분된 양의 정수 목록입니다. 컬럼 번호는 데이터 행이 시작되는 널(NULL) 표시기 필드의 바이트 오프셋입니다. METHOD L 매개변수에 정의된 각 데이터 필드에 대해 한 개의 항목이 널(NULL) 표시기 목록에 있어야 합니다. 컬럼 번호 0은 해당 데이터 필드에 데이터가 항상 들어 있음을 나타냅니다.

널(NULL) 표시기의 Y 값은 컬럼 데이터를 널(NULL)로 지정합니다. 널(NULL) 표시기 컬럼에서 Y 이외의 모든 문자는 컬럼 데이터가 널(NULL)이 아니며 METHOD L 옵션이 지정하는 컬럼 데이터가 로드됨을 지정합니다.

널(NULL) 표시기 문자는 MODIFIED BY 옵션을 사용하여 변경할 수 있습니다.

**N** 로드할 데이터 파일의 컬럼 이름을 지정합니다. 이 컬럼 이름의 대소문

자는 시스템 카탈로그의 해당 이름의 대소문자와 일치해야 합니다. 널 (NULL) 입력이 가능하지 않은 각 테이블 컬럼에는 METHOD N 목록에 해당 항목이 있어야 합니다. 예를 들면, 주어진 데이터 필드 F1, F2, F3, F4, F5 및 F6과 테이블 컬럼 C1 INT, C2 INT NOT NULL, C3 INT NOT NULL 및 C4 INT, method N (F2, F1, F4, F3)은 유효한 요청인 반면, method N (F2, F1)은 유효하지 않습니다. 이 메소드는 파일 유형이 IXF 또는 CURSOR인 경우에만 사용될 수 있습니다.

**P** 로드할 입력 데이터 필드의 필드 번호(1부터 번호를 지정)를 지정합니다. 널(NULL) 입력이 가능하지 않은 각 테이블 컬럼에는 METHOD P 목록에 해당 항목이 있어야 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)은 유효한 요청인 반면, method P (2, 1)은 유효하지 않습니다. 이 메소드는 파일 유형이 IXF, DEL 또는 CURSOR인 경우에만 사용할 수 있으며 DEL 파일 유형에 유일한 유효 메소드입니다.

#### **XML FROM *xml-path***

XML 파일이 들어 있는 하나 이상의 경로를 지정합니다.XDS는 주 데이터 파일(ASC, DEL 또는 IXF)에 포함되며 XML 컬럼으로 로드될 컬럼에 있습니다.

#### **XMLPARSE**

XML 문서가 구문 분석되는 방법을 지정합니다. 이 옵션을 지정하지 않을 경우, XML 문서에 대한 구문 분석 동작은 CURRENT XMLPARSE OPTION 특수 레지스터의 값 으로 판별됩니다.

#### **STRIP WHITESPACE**

XML 문서가 구문 분석될 때 공백을 제거하도록 지정합니다.

#### **PRESERVE WHITESPACE**

XML 문서가 구문 분석될 때 공백을 제거하지 않도록 지정합니다.

#### **XMLVALIDATE**

XML 문서가 스키마에 대해 유효성이 확인되도록 지정합니다(해당되는 경우).

#### **USING XDS**

기본 데이터 파일의 XDS(XML Data Specifier)에 의해 식별된 XML 스키마에 대해 XML 문서의 유효성을 확인합니다. USING XDS와 함께 XMLVALIDATE 옵션을 호출한 경우, 유효성 확인을 수행하는 데 사용된 스키마는 디폴트로 XDS의 SCH 속성에 의해 판별됩니다. SCH 속성이 XDS에 존재하지 않을 경우, DEFAULT 절로 디폴트 스키마를 지정하지 않으면 스키마 유효성 확인이 발생하지 않습니다.

DEFAULT, IGNORE 및 MAP 절은 스키마 판별 동작을 수정하는 데 사용될 수 있습니다. 이들 세 개의 선택적 절은 XDS의 권장 스펙에 직접적으로 적용되며 서로에게는 적용되지 않습니다. 예를 들어, 한 스키마가 DEFAULT 절에서 지정되어 선택되었으면 이 스키마는 IGNORE 절에 지정되어도 무시되지 않습니다. 마찬가지로 한 스키마가 MAP 절에서 첫 번째 파트 쌍으로 지정되어 선택되면 다른 MAP 절 쌍의 두 번째 파트에 지정되어도 다시 맵핑되지 않습니다.

#### **USING SCHEMA** *schema-sqlid*

XML 문서가 지정된 SQL ID가 있는 XML 스키마에 대해 유효성이 확인됩니다. 이 경우 모든 XML 컬럼에 대해 XDS(XML Data Specifier)의 SCH 속성이 무시됩니다.

#### **USING SCHEMALOCATION HINTS**

XML 문서가 소스 XML 문서의 XML 스키마 위치 힌트에 의해 식별된 스키마에 대해 유효성이 확인됩니다. XML 문서에서 schemaLocation 속성을 찾을 수 없으면 유효성 확인이 발생하지 않습니다. USING SCHEMALOCATION HINTS 절을 지정하면 모든 XML 컬럼에 대해 XDS(XML Data Specifier)의 SCH 속성이 무시됩니다.

아래의 XMLVALIDATE 옵션 예를 참조하십시오.

#### **IGNORE** *schema-sqlid*

이 옵션은 USING XDS 매개변수가 지정된 경우에만 사용할 수 있습니다. SCH 속성에 의해 식별될 경우 IGNORE 절은 무시할 하나 이상의 스키마 목록을 지정합니다. 로드된 XML 문서에 대한 XDS(XML Data Specifier)에 SCH 속성이 존재하고 SCH 속성에 의해 식별된 스키마가 IGNORE할 스키마 목록에 포함된 경우, 이 로드된 XML 문서에 대해서는 스키마 유효성 확인이 발생하지 않습니다.

주:

IGNORE 절에 스키마가 지정되어 있으면 이 스키마는 또한 MAP 절에 있는 스키마 쌍의 왼쪽에 존재할 수 없습니다.

IGNORE 절은 XDS에만 적용됩니다. MAP 절에 의해 맵핑된 스키마는 IGNORE 절에 의해 지정된 경우 계속 무시되지 않습니다.

#### **DEFAULT** *schema-sqlid*

이 옵션은 USING XDS 매개변수가 지정된 경우에만 사용할 수 있습니다. DEFAULT 절을 통해 지정된 스키마는 로드된 XML 문서의 XDS(XML Data Specifier)가 XML 스키마를 식별하는 SCH 속성을 포함하지 않을 때 유효성 확인에 사용할 스키마를 식별합니다.

DEFAULT 절은 IGNORE 및 MAP 절보다 우선순위를 갖습니다. XDS가 DEFAULT 절을 충족시키면 IGNORE 및 MAP 스펙은 무시됩니다.

#### **MAP** *schema-sqlid*

이 옵션은 USING XDS 매개변수가 지정된 경우에만 사용할 수 있습니다. 로드된 각 XML 문서에 대한 XDS(XML Data Specifier)의 SCH 속성이 지정하는 스키마 대신 사용할 대체 스키마를 지정하려면 이 MAP 절을 사용하십시오. MAP 절은 하나 이상의 스키마 쌍 목록을 지정하며, 여기서 각 쌍은 한 스키마 대 다른 스키마의 매핑을 나타냅니다. 쌍에서 첫 번째 스키마는 XDS에 있는 SCH 속성에 의해 참조되는 스키마를 나타냅니다. 쌍에서 두 번째 스키마는 스키마 유효성 확인을 수행하는 데 사용되어야 하는 스키마를 나타냅니다.

MAP 절에 있는 스키마 쌍의 왼쪽에 있는 스키마는 IGNORE 절에 지정될 수 없습니다.

스키마 쌍 매핑이 적용된 후, 최종 결과가 됩니다. 매핑 조작은 전이되지 않으므로 선택된 스키마는 다른 스키마 상 매핑에 계속 적용되지 않습니다.

스키마가 두 번 이상 매핑될 수 없다는 것은 쌍의 왼쪽에 두 번 이상 나타날 수 없다는 것을 의미합니다.

#### **SAVECOUNT** *n*

로드 유틸리티가 *N*개의 행 뒤에 매번 일관성 지점을 설정하도록 지정합니다. 이 값은 쪽 수로 변환되며 Extent 크기의 간격으로 반올림됩니다. 메시지는 각각의 일관성 지점에서 발행되므로 LOAD QUERY를 사용하여 로드 조작을 모니터링하는 경우 이 옵션을 선택해야 합니다. *N* 값이 충분히 높지 않으면 각 일관성 지점에서 수행된 활동의 동기화가 성능에 영향을 줍니다.

디폴트값은 0으로, 필요하지 않으면 일관성 지점을 설정하지 않음을 의미합니다.

CURSOR 파일 유형과 함께 지정되거나 XML 컬럼을 포함하는 테이블 로드 시 이 옵션은 무시됩니다.

#### **ROWCOUNT** *n*

로드될 파일에 있는 *N*개의 실제 레코드 수를 지정합니다. 사용자가 첫 번째 *N* 개 행만 로드할 수 있도록 합니다.

#### **WARNINGCOUNT** *n*

*N*번의 경고 후에 로드 조작을 중지합니다. 경고가 예상되지는 않지만 올바른 파일 및 테이블이 사용되고 있는지에 대한 검증을 원할 경우 이 매개변수를 설정하십시오. 로드 파일이나 목표 테이블이 잘못 지정된 경우, 로드 유틸리티가 로드를 시도하는 각 행에 대해 경고를 생성하므로 로드 실패하게 됩니다. *N* 이 0이거나 이 옵션을 지정하지 않으면 발행된 경고 수에 관계없이 로드 조작을 계속합니다. 경고의 임계값에 도달하여 로드 조작이 중지된 경우, 다른 로드

조작이 RESTART 모드로 시작될 수 있습니다. 로드 조작은 마지막 일관성 지점에서 자동으로 계속됩니다. 또는 다른 로드 조작을 REPLACE 모드로 초기화하여 입력 파일의 처음부터 시작하게 할 수 있습니다.

#### **MESSAGES** *message-file*

로드 조작 중 발생하는 경고 및 오류 메시지의 목적지를 지정합니다. 메시지 파일이 지정되어 있지 않으면 메시지가 표준 출력에 작성됩니다. 파일에 대한 전체 경로가 지정되어 있지 않으면 로드 유틸리티는 현재 디렉토리 및 디폴트 드라이브를 목적으로 사용합니다. 이미 존재하는 파일의 이름이 지정되면 유틸리티는 정보를 추가합니다.

일반적으로 메시지 파일은 로드 조작이 종료될 때 메시지로 채워지므로 조작의 진행을 모니터링하는 데는 적합하지 않습니다.

#### **TEMPFILES PATH** *temp-pathname*

로드 조작 중에 임시 파일을 작성할 때 사용되는 경로의 이름을 지정하며 서버 데이터베이스 파티션에 따라 완전한 이름이어야 합니다.

임시 파일은 파일 시스템 스페이스를 차지합니다. 때때로 이 스페이스는 대량으로 요구됩니다. 다음은 모든 임시 파일에 할당되어야 하는 파일 시스템 스페이스의 예상 크기입니다.

- 로드 유틸리티가 생성하는 각각의 메시지 당 136바이트
- 데이터 파일에 Long 필드나 LOB가 포함된 경우 15KB의 오버헤드. 이 수량은 INSERT 옵션이 지정되어 있고 테이블에 대량의 Long 필드나 LOB 데이터가 있는 경우 급격히 증가할 수 있습니다.

#### **INSERT**

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

#### **REPLACE**

로드 유틸리티가 실행될 수 있는 네 가지 모드 중 하나. 테이블에서 기존 데이터를 모두 삭제하고 로드된 데이터를 삽입합니다. 테이블 정의 및 인덱스 정의는 변경되지 않습니다. 계층 구조 간에 데이터를 이동하는 경우에 이 옵션을 사용하면 개별 부속 테이블이 아닌 전체 계층 구조에 대한 데이터만 바뀔 수 있습니다.

#### **KEEPDICTIONARY**

기존 압축 사전은 LOAD REPLACE 조작에서 보존됩니다. 테이블 COMPRESS 속성이 YES로 제공되면, 새로 교체된 데이터는 로드의 호출 이전에 존재하는 사전을 사용하여 압축됩니다. 테이블에 이미 존재하는 사전이 없는 경우, 테이블 COMPRESS 속성이 YES이면 테이블로 교체 중인 데이터를 사용하여 새 사전이 빌드됩니다. 이 경우 압축 사전을 빌드하는 데 필요한 데이터의 양은 ADC의 규정에 따릅니다.

다. 이 데이터는 압축되지 않은 채로 테이블에 채워집니다. 사전이 테이블에 삽입되면, 로드되는 남아 있는 데이터는 이 사전과 함께 압축됩니다. 이것은 디폴트 매개변수입니다. 요약 정보는 아래 테이블 1을 참조하십시오.

표 19. LOAD REPLACE KEEPDICTIONARY

압축	테이블 행 데이터 사전이 존재합니다.	XML 스토리지 오브젝트 사전이 존재합니다. 1	압축 사전	데이터 압축
YES	YES	YES	테이블 행 데이터 및 XML 사전을 보존합니다.	로드되는 데이터는 압축됩니다.
YES	YES	NO	테이블 행 데이터 사전을 보존하고 새 XML 사전을 빌드합니다.	로드되는 테이블 행 데이터는 압축됩니다. XML 사전이 빌드된 후에, 로드되는 남아 있는 XML 데이터는 압축됩니다.
YES	NO	YES	테이블 행 데이터 사전을 빌드하고 새 XML 사전을 보존합니다.	테이블 행 데이터 사전이 빌드된 후에, 로드되는 남아 있는 테이블 행 데이터는 압축됩니다. 로드되는 XML 데이터는 압축됩니다.
YES	NO	NO	새 테이블 행 데이터 및 XML 사전을 빌드합니다.	사전이 빌드된 후에, 로드되는 남아 있는 데이터는 압축됩니다.
NO	YES	YES	테이블 행 데이터 및 XML 사전을 보존합니다.	로드되는 데이터는 압축되지 않습니다.
NO	YES	NO	테이블 행 데이터 사전을 보존합니다.	로드되는 데이터는 압축되지 않습니다.
NO	NO	YES	테이블 행 사전에 영향을 미치지 않습니다. XML 사전을 보존합니다.	로드되는 데이터는 압축되지 않습니다.
NO	NO	NO	영향을 미치지 않습니다.	로드되는 데이터는 압축되지 않습니다.

주:

1. XML 컬럼이 DB2 버전 9.7 이상에서 테이블에 추가된 경우 또는 테이블이 온라인 테이블 이동을 사용하여 이주된 경우에만 테이블의 XML 스토리지 오브젝트의 압축 사전이 작성될 수 있습니다.

### RESETDICTIONARY

이 지시문은 LOAD REPLACE 처리 시 테이블 COMPRESS 속성이 YES인 경우 테이블 데이터 오브젝트의 새 사전을 빌드하도록 합니다. COMPRESS 속성이 NO이고 사전이 이미 테이블에 있으면 제거되며 새 사전은 테이블로 삽입되지 않습니다. 압축 사전은 오직 하나의 사용자 레코드로 빌드될 수 있습니다. 로드된 데이터 세트 크기가 0이고 기존에 존재하는 사전이 있는 경우, 이 사전은 보존되지 않습니다. 이 지시문으로 사전을 빌드하도록 요구되는 데이터의 양은 ADC의 규정에 따르지 않습니다. 요약 정보는 아래 테이블 2를 참조하십시오.

표 20. LOAD REPLACE RESETDICTIONARY

압축	테이블 행 데이터 사전이 존재합니다.	XML 스토리지 오브젝트 사전이 존재합니다. <sup>1</sup>	압축 사전	데이터 압축
YES	YES	YES	새 사전을 빌드합니다 <sup>2</sup> . DATA CAPTURE CHANGES 옵션이 CREATE TABLE 또는 ALTER TABLE문에서 사용 가능하면, 현재 테이블 행 데이터 사전이 보존됩니다(실행기록 압축 사전이라고도 함).	사전이 빌드된 후에, 로드되는 남아 있는 데이터는 압축됩니다.
YES	YES	NO	새 사전을 빌드합니다 <sup>2</sup> . DATA CAPTURE CHANGES 옵션이 CREATE TABLE 또는 ALTER TABLE문에서 사용 가능하면, 현재 테이블 행 데이터 사전이 보존됩니다(실행기록 압축 사전이라고도 함).	사전이 빌드된 후에, 로드되는 남아 있는 데이터는 압축됩니다.
YES	NO	YES	새 사전을 빌드합니다.	사전이 빌드된 후에, 로드되는 남아 있는 데이터는 압축됩니다.
YES	NO	NO	새 사전을 빌드합니다.	사전이 빌드된 후에, 로드되는 남아 있는 데이터는 압축됩니다.
NO	YES	YES	사전을 제거합니다.	로드되는 데이터는 압축되지 않습니다.
NO	YES	NO	테이블 행 데이터 사전을 제거합니다.	로드되는 데이터는 압축되지 않습니다.
NO	NO	YES	XML 스토리지 오브젝트 사전을 제거합니다.	로드되는 데이터는 압축되지 않습니다.
NO	NO	NO	영향을 미치지 않습니다.	모든 테이블 데이터는 압축되지 않습니다.

**주:**

1. XML 컬럼이 DB2 버전 9.7 이상에서 테이블에 추가된 경우 또는 테이블이 온라인 테이블 이동을 사용하여 이주된 경우에만 테이블의 XML 스토리지 오브젝트의 압축 사전이 작성될 수 있습니다.
2. 사전이 존재하고 압축 속성이 사용 가능하지만 테이블 파티션에 로드할 레코드가 없는 경우, 새 사전은 빌드될 수 없으며 RESETDICTIONARY 조작은 기존의 사전을 보존하지 않습니다.

**TERMINATE**

로드 유틸리티가 실행될 수 있는 네 가지 모드 중 하나. 일관성 지점이 패스된 경우라도 이전에 인터럽트된 로드 조작을 종료하고 조작이 시작된 특정 시점으로 롤백합니다. 조작에 관계된 테이블 스페이스의 상태가 정상으로 리턴되고 모



든 테이블 오브젝트는 일관성을 갖게 됩니다(인덱스 오브젝트가 유효하지 않은 것으로 표시되는 경우, 인덱스 재빌드가 다음 액세스에서 자동으로 발생). 종료되는 로드 조작이 LOAD REPLACE이면, 테이블은 LOAD TERMINATE 조작 이후에 빈 테이블로 절단됩니다. 종료되는 로드 조작이 LOAD INSERT이면, LOAD TERMINATE 조작 이후에 테이블은 원래의 모든 레코드를 보유합니다. 사전 관리 요약 정보는 아래 테이블 3을 참조하십시오.

LOAD TERMINATE 옵션은 테이블 스페이스에서 백업 보류 상태를 제거하지 않습니다.

## RESTART

로드 유틸리티가 실행될 수 있는 네 가지 모드 중 하나. 이전에 인터럽트된 로드 조작을 재시작합니다. 로드 조작은 로드, 빌드 또는 삭제 단계의 마지막 일관성 지점에서 자동으로 계속됩니다. 사전 관리 요약 정보는 아래 테이블 4를 참조하십시오.

## INTO *table-name*

데이터가 로드되어야 할 데이터베이스 테이블을 지정합니다. 이 테이블은 시스템 테이블, 선언된 임시 테이블 또는 작성된 임시 테이블일 수 없습니다. 별명 또는 완전하거나 규정에 맞지 않는 테이블 이름을 지정할 수 있습니다. 규정된 테이블 이름의 양식은 `schema.tablename`입니다. 규정에 맞지 않는 테이블 이름이 지정되면 테이블은 CURRENT SCHEMA로 규정됩니다.

## *insert-column*

데이터가 삽입되어야 할 테이블 컬럼을 지정합니다.

로드 유틸리티는 하나 이상의 스페이스를 포함하는 이름을 가진 컬럼의 구문을 분석할 수 없습니다. 예를 들면, 다음과 같습니다.

```
db2 load from delfile1 of del 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)
```

Int 4 컬럼 때문에 실패합니다. 솔루션은 이 컬럼 이름을 큰따옴표 안에 두는 것입니다.

```
db2 load from delfile1 of del 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)
```

## FOR EXCEPTION *table-name*

오류 행이 복사될 예외 테이블을 지정합니다. 고유 인덱스 또는 기본 키를 위반하는 모든 행이 복사됩니다. 규정에 맞지 않는 테이블 이름이 지정되면 테이블은 CURRENT SCHEMA로 규정됩니다.

예외 테이블에 작성된 정보는 덤프 파일에는 작성되지 않습니다. 파티션된 데이터베이스 환경에서는 예외 테이블이 로드 중인 테이블이 정의된 데이터베이스 파티션에 대해 정의되어야 합니다. 그렇지 않으면, 덤프 파일에는 유효하지 않거나 구문 오류 때문에 로드될 수 없는 행이 포함됩니다.

XML 데이터 로드 시에는 다음과 같은 경우 FOR EXCEPTION절을 사용하여 로드 예외 테이블을 지정할 수 없습니다.

- 레이블 기반 액세스 제어(LBAC)를 사용하는 경우.
- 데이터 파티션 테이블로 데이터를 로드하는 경우.

#### **NORANGEEXC**

행이 범위 위반으로 거부되는 경우에는 예외 테이블에 삽입되지 않는다는 것을 표시합니다.

#### **NOUNIQUEEXC**

행이 고유 제한조건 위반으로 거부되는 경우에는 예외 테이블에 삽입되지 않는다는 것을 표시합니다.

#### **STATISTICS USE PROFILE**

이 테이블에 대해 정의된 프로파일에 따라 로드 중 통계를 수집하도록 로드를 지시합니다. 이 프로파일은 로드가 실행되기 전에 작성되어야 합니다. 이 프로파일은 RUNSTATS 명령으로 작성됩니다. 프로파일이 존재하지 않고 프로파일에 따라 통계를 수집하도록 로드에서 지시된 경우 경고가 리턴되며 통계가 수집되지 않습니다.

#### **STATISTICS NO**

통계가 수집되지 않고 카탈로그의 통계가 변경되지 않도록 지정합니다. 이는 디폴트값입니다.

#### **COPY NO**

포워드 복구가 사용 가능한 경우(즉, *logretain* 또는 *userexit*이 설정된 경우), 테이블이 있는 테이블 스페이스가 백업 보류 상태에 있도록 지정합니다. COPY NO 옵션을 사용하면 테이블 스페이스 상태를 로드 진행 중 테이블 스페이스 상태로 둡니다. 이 상태는 로드가 완료되거나 중단되면 소멸되는 일시적인 상태입니다. 테이블 스페이스의 모든 테이블의 데이터는 테이블 스페이스 백업 또는 전체 데이터베이스 백업이 완료되어야 갱신하거나 삭제할 수 있습니다. 그러나 SELECT문을 사용하여 테이블의 데이터에 액세스할 수 있습니다.

복구 가능한 데이터베이스에 대해 COPY NO 옵션이 지정된 LOAD는 테이블 스페이스를 백업 보류 상태로 둡니다. 예를 들어, COPY NO 및 INDEXING MODE DEFERRED 옵션과 함께 LOAD를 수행하면 인덱스를 새로 고쳐야 합니다. 테이블에 대한 특정 쿼리는 인덱스 스캔을 해야 하며 인덱스를 새로 고쳐야 성공할 수 있습니다. 백업 보류 상태에 있는 테이블 스페이스에 위치한 인덱스는 새로 고칠 수 없습니다. 이러한 경우, 백업을 수행해야 테이블에 대한 액세스가 허용됩니다. 인덱스 새로 고침은 쿼리로 인덱스를 액세스할 때 데이터베이스에 의해 자동으로 완료됩니다. COPY NO, COPY YES 또는

NONRECOVERABLE 중 하나가 지정되지 않고 데이터베이스가 복구 가능한 경우(**logretain** 또는 **logarchmeth1**이 사용 가능함), COPY NO가 디폴트입니다.

#### **COPY YES**

로드된 데이터의 사본이 저장되도록 지정합니다. 포워드 복구가 사용 불가능한 경우 이 옵션은 유효하지 않습니다.

#### **USE TSM**

TSM(Tivoli Storage Manager)을 사용하여 사본이 저장되도록 지정합니다.

#### **OPEN *num-sess* SESSIONS**

TSM 또는 벤더 제품과 함께 사용되는 입출력 세션 수. 디폴트값은 1입니다.

#### **TO *device/directory***

이미지 복사가 작성될 디바이스 또는 디렉토리를 지정합니다.

#### **LOAD *lib-name***

사용될 벤더 백업 및 리스토어 I/O 기능이 들어 있는 공유 라이브러리 이름 (Windows 운영 체제의 DLL). 여기에는 전체 경로가 포함될 수 있습니다. 전체 경로가 지정되지 않으면 디폴트값은 User Exit 프로그램이 있는 경로입니다.

#### **NONRECOVERABLE**

로드 트랜잭션이 복구 불가능하도록 표시하여 후속적인 롤 포워드 조치로 복구할 수 없도록 지정합니다. 롤 포워드 유틸리티가 트랜잭션을 건너뛰고 데이터가 로드 중인 테이블을 "유효하지 않음"으로 표시합니다. 또한 이 유틸리티는 해당 테이블에 대한 후속 트랜잭션을 무시합니다. 롤 포워드 조치가 완료된 후 이러한 테이블은 복구 불가능한 로드 조작 완료 후 커밋 지점 다음에 수행되는 백업(전체 또는 테이블 스페이스)에서 삭제되거나 리스토어될 수 있습니다.

이 옵션을 사용하면 로드 조작 후에 테이블 스페이스가 백업 보류 상태가 되지 않으며 로드 조작 중에 로드된 데이터의 사본을 작성할 필요가 없습니다. COPY NO, COPY YES 또는 NONRECOVERABLE 중 하나가 지정되지 않고 데이터베이스가 복구 가능하지 않은 경우(**logretain** 또는 **logarchmeth1**이 사용 가능하지 않음), NONRECOVERABLE이 디폴트입니다.

#### **WITHOUT PROMPTING**

데이터 파일의 목록에 로드될 모든 파일이 포함되고 나열된 디바이스 또는 디렉토리가 전체 로드 조작에 충분하도록 지정합니다. 연속 입력 파일을 찾을 수 없거나 목표 복사가 로드 조작이 완료되기 전에 채워지면 로드 조작에 실패하고 테이블이 로드 보류 상태가 됩니다.

이 옵션이 지정되지 않고 테이프 디바이스가 이미지 복사를 위한 테이프의 끝에 도달하거나 나열된 마지막 항목이 테이프 디바이스인 경우, 사용자에게 해당 디바이스에 새 테이프를 장착하도록 프롬프트합니다.

#### **DATA BUFFER** *buffer-size*

유틸리티에서 데이터 전송에 필요한 버퍼 스페이스로 사용할 4KB 페이지의 수를 지정합니다(병렬 처리 수준과 무관함). 지정된 값이 알고리즘의 최소보다 작으면, 최소 필요 자원이 사용되고 경고는 리턴되지 않습니다.

이 메모리는 유틸리티 힙에서 바로 할당되며 크기는 *util\_heap\_sz* 데이터베이스 구성 매개변수로 수정할 수 있습니다.

값이 지정되지 않으면 런타임시 유틸리티에 의해 적절한 디폴트값이 계산됩니다. 디폴트값은 테이블의 일부 등록 정보 뿐 아니라 로더의 인스턴스화 시간에 유틸리티 힙에서 사용 가능한 여유 공간의 백분율을 기초로 합니다.

#### **SORT BUFFER** *buffer-size*

이 옵션은 로드 조작 중 SORTHEAP 데이터베이스 구성 매개변수를 겹쳐쓰는 값을 지정합니다. 이 옵션은 인덱스와 함께 테이블을 로드하는 경우와 INDEXING MODE 매개변수가 DEFERRED로 지정되지 않은 경우에만 의미가 있습니다. 지정된 값은 SORTHEAP의 값을 초과할 수 없습니다. 이 매개변수는 SORTHEAP의 값을 변경하지 않고 많은 인덱스가 있는 테이블을 로드할 때 사용되는 정렬 메모리를 조절하는 데 유용하며 일반 쿼리 처리에도 영향을 줍니다.

#### **CPU\_PARALLELISM** *n*

테이블 오브젝트를 빌드할 때 레코드를 구문 분석, 변환 및 포맷팅하기 위해 로드 유틸리티가 작성하는 프로세스 또는 스레드의 수를 지정합니다. 이 매개변수는 데이터베이스 파티션당 실행하는 프로세스 수를 이용하도록 설계되었습니다. 소스 데이터에 있는 기록 순서가 보존되므로 사전에 정렬된 데이터를 로드할 때 특히 유용합니다. 이 매개변수의 값이 0이거나 지정되지 않으면 로드 유틸리티는 적절한 디폴트값(일반적으로 사용 가능한 CPU 수에 기초함)을 사용합니다.

주:

1. 이 매개변수는 LOB 또는 LONG VARCHAR 필드 중 하나가 있는 테이블과 함께 사용되면 시스템 CPU 수나 사용자가 지정한 값과 관계없이 값은 1이 됩니다.
2. SAVECOUNT 매개변수에 값을 작게 지정하면 로더는 데이터 및 테이블 메타데이터를 둘 다 비우기 위해 훨씬 더 많은 입출력 조작을 수행하게 됩니다. CPU\_PARALLELISM이 1보다 크면 플러시 조작은 비동기로 수행되며 로더가 CPU를 이용할 수 있도록 합니다. CPU\_PARALLELISM이 1로 설정되면 로더는 일관성 지점 동안 입출력을 대기합니다.

CPU\_PARALLELISM이 2로 설정되고 SAVECOUNT가 10 000으로 설정된 로드 조작용 CPU가 하나만 있어도 CPU\_PARALLELISM이 1로 설정된 동일한 조작용보다 빠르게 완료됩니다.

#### **DISK\_PARALLELISM *n***

로드 유틸리티가 데이터를 테이블 스페이스 컨테이너에 작성하기 위해 작성하는 프로세스나 스레드의 수를 지정합니다. 값이 지정되지 않으면 유틸리티가 테이블 스페이스 컨테이너의 수 및 테이블의 등록 정보를 기반으로 하여 적절한 디폴트값을 선택합니다.

#### **FETCH\_PARALLELISM YES | NO**

DATABASE 키워드로 선언된 커서에서 로드를 수행하거나 API sqlu\_remotefetch\_entry 미디어 항목을 사용하고 이 옵션이 YES로 설정되어 있는 경우, 로드 유틸리티는 가능한 리모트 데이터 소스로부터 페치(fetch)를 병렬 처리하려고 합니다. NO로 설정되면 병렬 페치(fetch)는 수행되지 않습니다. 디폴트값은 YES입니다. 자세한 정보는 *CURSOR* 파일 유형을 사용하여 데이터 이동을 참조하십시오.

#### **INDEXING MODE**

로드 유틸리티가 인덱스를 재빌드할 것인지 또는 인덱스를 점차적으로 확장할 것인지를 지정합니다. 가능한 값은 다음과 같습니다.

##### **AUTOSELECT**

로드 유틸리티는 자동으로 REBUILD나 INCREMENTAL 모드를 결정합니다. 이 결정은 로드 중인 데이터 양과 인덱스 트리의 용량을 기반으로 합니다. 인덱스 트리의 용량에 관한 정보는 인덱스 오브젝트에 저장되어 있습니다. 이 정보를 채우기 위해 RUNSTATS는 필요하지 않습니다. 디폴트 인덱스 모드는 AUTOSELECT입니다.

##### **REBUILD**

모든 인덱스가 재빌드됩니다. 이전 및 추가된 테이블 데이터에 대해 모든 인덱스 키 파트를 정렬하려면 유틸리티에 충분한 자원이 있어야 합니다.

##### **INCREMENTAL**

인덱스는 새로운 데이터와 함께 확장됩니다. 이 방법은 인덱스 여유 공간을 사용합니다. 삽입된 레코드에 대한 인덱스 키를 추가하려면 충분한 정렬 스페이스만 있으면 됩니다. 이 메소드는 인덱스 오브젝트가 유효하고 로드 조작용이 시작될 때 액세스할 수 있는 경우에만 지원됩니다(예를 들어, DEFERRED 모드가 지정된 로드 조작용 직후에는 유효하지 않음). 이 모드가 지정되었지만 인덱스 상태 때문에 지원되지 않는 경우, 경고가 리턴되며 로드 조작용은 계속 REBUILD 모드입니다. 마찬가지로 로드 빌드 단계에서 로드 재시작 조작용이 시작된 경우, INCREMENTAL 모드는 지원되지 않습니다.

다음 조건에 모두 해당될 때 증분 인덱싱은 지원되지 않습니다.

- LOAD COPY 옵션이 지정되었습니다(USEREXIT 또는 LOGRETAIN 옵션을 사용하여 *logarchmeth1*).
- 테이블이 DMS 테이블 스페이스에 있습니다.
- 인덱스 오브젝트가 로드 중인 테이블에 속한 다른 테이블 오브젝트가 공유하는 테이블에 있습니다.

이 제한사항을 통과하려면 인덱스를 별도의 테이블 스페이스에 넣는 것이 바람직합니다.

### DEFERRED

이 모드를 지정하면 로드 유틸리티가 인덱스 작성을 시도하지 않습니다. 인덱스는 새로 고칠 필요가 있는 것으로 표시됩니다. 로드 조작과 관련이 없는 이러한 인덱스에 처음 액세스할 때 강제로 재빌드될 수 있습니다. 그렇지 않으면 인덱스는 데이터베이스가 재시작될 때 재빌드될 수 있습니다. 이러한 방법을 수행하려면 가장 큰 인덱스의 모든 키 부분에 대해 충분한 정렬 스페이스가 있어야 합니다. 인덱스 구성에 연속적으로 사용되는 총 시간은 REBUILD 모드에 필요한 시간보다 깁니다. 따라서 지연된 인덱싱으로 다중 로드 조작을 수행할 때 로드하지 않는 액세스에서 먼저 인덱스를 재빌드하게 하는 것보다는 시퀀스에서 마지막 로드 조작에서 인덱스 재빌드를 수행하게 하는 것이 바람직합니다(성능 관점에서).

지연된 인덱싱은 비고유 인덱스가 있는 테이블에만 지원되므로 로드 단계 중 삽입된 중복 키는 로드 조작 후 지속되지 않습니다.

### ALLOW NO ACCESS

로드가 로드 중 독점 액세스를 위해 목표 테이블을 잠급니다. 로드 중 테이블 상태는 로드 진행 중으로 설정됩니다. ALLOW NO ACCESS는 디폴트 동작입니다. LOAD REPLACE에는 이 옵션만 유효합니다.

테이블에 제한조건이 있을 때 테이블 상태는 로드 진행 중과 함께 무결성 설정 오류로 설정됩니다. 테이블을 무결성 설정 오류 상태에서 해제하려면 SET INTEGRITY문을 사용해야 합니다.

### ALLOW READ ACCESS

로드가 목표 테이블을 공유 모드로 잠급니다. 테이블 상태는 로드 진행 중 및 읽기 액세스 모두로 설정됩니다. 관독기는 테이블이 로드되는 동안 데이터의 비델타 부분에 액세스할 수 있습니다. 다시 말해, 테이블 관독기는 로드가 시작되기 전에 있던 데이터에 액세스할 수 있지만 로드되고 있는 데이터는 로드가 완료될 때까지 사용할 수 없습니다. ALLOW READ ACCESS 로드의 LOAD TERMINATE 또는 LOAD RESTART는 이 옵션을 사용할 수 있지만 ALLOW NO ACCESS 로드의 LOAD TERMINATE 또는 LOAD

RESTART는 이 옵션을 사용할 수 없습니다. 더우기 목표 테이블의 인덱스가 재빌드가 필요한 것으로 표시된 경우 이 옵션은 유효하지 않습니다.

테이블에 제한조건이 있으면 테이블 상태는 로드 진행 중, 읽기 액세스뿐만 아니라 무결성 설정 보류로 설정됩니다. 로드가 종료되면 테이블 상태 중에서 로드 진행 중 상태는 제거되지만 무결성 설정 보류 및 읽기 액세스 상태는 남아 있습니다. 테이블을 무결성 설정 보류 상태에서 해제하려면 SET INTEGRITY 문을 사용해야 합니다. 테이블의 상태가 무결성 설정 보류 및 읽기 액세스인 경우, 판독기는 데이터의 비델타 부분에는 여전히 액세스할 수 있으나 SET INTEGRITY문이 완료될 때까지 데이터의 새 (델타) 부분에는 액세스할 수 없습니다. 사용자는 SET INTEGRITY문을 발행하지 않고 동일한 테이블에서 다중 로드를 수행할 수 있습니다. 그러나 SET INTEGRITY문이 발행될 때까지 원본(확인된) 데이터만 볼 수 있습니다.

ALLOW READ ACCESS는 다음 수정자도 지원합니다.

#### **USE *tablespace-name***

인덱스가 재빌드되는 경우, 인덱스의 웨도우 사본이 테이블 스페이스 *tablespace-name*에 빌드되고 INDEX COPY PHASE 중 로드 맨 끝에서 원본 테이블 스페이스로 복사됩니다. 이 옵션에는 시스템 임시 테이블 스페이스만 사용할 수 있습니다. 테이블 스페이스를 지정하지 않을 경우 음영 인덱스는 인덱스 오브젝트와 동일한 테이블 스페이스에 작성됩니다. 음영 사본이 인덱스 오브젝트와 동일한 테이블 스페이스에 작성된 경우, 이전 인덱스 오브젝트에 대한 음영 인덱스 오브젝트의 복사는 순간적입니다. 음영 사본이 인덱스 오브젝트와 다른 테이블 스페이스에 있을 경우 실제 복사가 수행됩니다. 상당한 I/O 및 시간이 포함될 수 있습니다. INDEX COPY PHASE 중 로드 맨 끝에서 테이블이 오프라인 상태에 있는 동안 복사가 발생합니다.

이 옵션을 사용하지 않으면 음영 인덱스는 원본과 동일한 테이블 스페이스에서 빌드됩니다. 원본 인덱스와 음영 인덱스가 둘 다 디폴트로 동시에 동일한 테이블 스페이스에 있으므로 한 테이블 스페이스에 이들 인덱스를 모두 보유하기 위한 스페이스가 충분하지 않을 수 있습니다. 이 옵션을 사용하면 인덱스에 충분한 테이블 스페이스를 보유할 수 있습니다.

사용자가 INDEXING MODE REBUILD 또는 INDEXING MODE AUTOSELECT를 지정하지 않을 경우 이 옵션은 무시됩니다. INDEXING MODE AUTOSELECT가 선택되고 로드가 인덱스를 점차적으로 갱신하도록 선택할 경우에도 이 옵션이 무시됩니다.

#### **SET INTEGRITY PENDING CASCADE**

LOAD로 인해 테이블이 무결성 설정 보류 상태에 놓일 경우, 사용자는 SET INTEGRITY PENDING CASCADE 옵션을 사용하여 로드된 테이블의 무결

성 설정 보류 상태가 모든 하위(하위 외부 키 테이블, 하위 즉시 구체화된 쿼리 테이블 및 하위 즉시 스테이징 테이블 포함)에 즉시 연쇄되는지 여부를 지정할 수 있습니다.

### **IMMEDIATE**

무결성 설정 보류 상태가 하위 외부 키 테이블, 하위 즉시 구체화된 쿼리 테이블 및 하위 즉시 스테이징 테이블로 즉시 확장됨을 표시합니다. LOAD INSERT 조작의 경우, IMMEDIATE 옵션을 지정해도 무결성 설정 보류 상태는 하위 외부 키 테이블로 확장되지 않습니다.

나중에 로드된 테이블에 제한조건 위반이 있는지 여부를 점검할 때(SET INTEGRITY문의 IMMEDIATE CHECKED 옵션을 사용하여) 무결성 설정 보류 읽기 액세스에 있던 하위 외부 키 테이블은 무결성 설정 보류 권한 없음 상태가 됩니다.

### **DEFERRED**

로드된 테이블만이 무결성 설정 보류 상태에 놓임을 표시합니다. 하위 외부 키 테이블, 하위 즉시 구체화된 쿼리 테이블 및 하위 즉시 스테이징 테이블의 상태는 변경되지 않습니다.

하위 외부 키 테이블에 제한조건 위반이 있는지 여부를 점검할 때(SET INTEGRITY문의 IMMEDIATE CHECKED 옵션을 사용하여) 하위 외부 키 테이블은 나중에 내재적으로 무결성 설정 보류 상태에 놓일 수 있습니다. 하위 즉시 구체화된 쿼리 테이블 및 하위 즉시 스테이징 테이블은 해당 기본 테이블 중 하나에 무결성 위반이 있는지 여부를 점검할 때 내재적으로 무결성 설정 보류 상태가 됩니다. 하위 테이블이 무결성 설정 보류 상태가 되었음을 나타내는 경고(SQLSTATE 01586)가 발행됩니다. 이들 하위 테이블이 언제 무결성 설정 보류 상태가 되는지에 대해서는 SQL 참조서에서 SET INTEGRITY문의 참고 절을 참조하십시오.

SET INTEGRITY PENDING CASCADE 옵션을 지정하지 않은 경우,

- 로드된 테이블만이 무결성 설정 보류 상태에 놓입니다. 하위 외부 키 테이블, 하위 즉시 구체화된 쿼리 테이블 및 하위 즉시 스테이징 테이블의 상태는 계속 변경되지 않으며 나중에 로드된 테이블에 제한조건 위반이 있는지 점검할 때 내재적으로 무결성 설정 보류 상태가 될 수 있습니다.

LOAD 결과 목표 테이블이 무결성 설정 보류 상태가 되지 않을 경우 SET INTEGRITY PENDING CASCADE 옵션은 무시됩니다.

### **LOCK WITH FORCE**

이 유틸리티는 로드 처리에서 테이블 잠금을 포함하는 다양한 잠금을 획득합니다. 대기라기 보다는 시간종료라고 할 수 있으며, 잠금을 획득할 때 로드는 이 옵션을 통해 목표 테이블에 충돌 잠금을 보유하고 있는 기타 응용프로그램을 강



제로 해제할 수 있습니다. 시스템 카탈로그 테이블에서 충돌 잠금을 보유하고 있는 응용프로그램은 로드 유틸리티로 강제로 해제할 수 없습니다. 강제된 응용프로그램은 롤백되고 로드 유틸리티가 필요로 하는 잠금을 릴리스합니다. 그런 다음 로드 유틸리티는 계속 진행될 수 있습니다. 이 옵션을 사용하려면 FORCE APPLICATIONS 명령(SYSADM 또는 SYSCTRL)과 동일한 권한이 필요합니다.

ALLOW NO ACCESS 로드는 로드 조작이 시작될 때 충돌 잠금을 보유하고 있는 응용프로그램을 강제 실행할 수 있습니다. 로드가 시작될 때 유틸리티는 테이블을 쿼리하거나 수정하려고 시도 중인 응용프로그램을 강제 실행할 수 있습니다.

ALLOW READ ACCESS 로드는 로드 조작이 시작되거나 종료될 때 충돌 잠금을 보유하고 있는 응용프로그램을 강제 실행할 수 있습니다. 로드가 시작될 때 유틸리티는 테이블 수정하려고 시도 중인 응용프로그램을 강제 실행할 수 있습니다. 로드 조작이 종료될 때 로드 유틸리티는 테이블을 쿼리하거나 수정하려고 시도 중인 응용프로그램을 강제 실행할 수 있습니다.

#### **SOURCEUSEREXIT** *executable*

유틸리티에 데이터를 입력하기 위해 호출될 실행 파일 이름을 지정합니다.

#### **REDIRECT**

##### **INPUT FROM**

###### **BUFFER** *input-buffer*

*input-buffer*에 지정된 바이트 스트림을 지정된 실행 파일을 실행하는 프로세스의 STDIN 파일 디스크립터로 패스합니다.

###### **FILE** *input-file*

이 클라이언트측 파일의 내용을 지정된 실행 파일을 실행하는 프로세스의 STDIN 파일 디스크립터로 패스합니다.

##### **OUTPUT TO**

###### **FILE** *output-file*

STDOUT 및 STDERR 파일 디스크립터를 지정된 서버측 파일로 캡처합니다.

#### **PARALLELIZE**

여러 User Exit 프로세스를 동시 호출하여 로드 유틸리티로 들어가는 데이터의 처리량을 증가시킵니다. 이 옵션은 다중 파티션 데이터베이스 환경에서만 적용할 수 있으므로 단일 파티션 데이터베이스 환경에서는 무시됩니다.

자세한 정보는 사용자 정의된 응용프로그램을 사용하여 데이터 이동을 참조하십시오.

### **PARTITIONED DB CONFIG** *partitioned-db-option*

다중 데이터베이스 파티션에 분산된 테이블 로드를 실행할 수 있도록 합니다. PARTITIONED DB CONFIG 매개변수를 사용하면 파티션된 데이터베이스에만 해당되는 구성 옵션을 지정할 수 있습니다. *partitioned-db-option* 값은 다음과 같이 지정할 수 있습니다.

```
PART_FILE_LOCATION x
OUTPUT_DBPARTNUMS x
PARTITIONING_DBPARTNUMS x
MODE x
MAX_NUM_PART_AGENTS x
ISOLATE_PART_ERRS x
STATUS_INTERVAL x
PORT_RANGE x
CHECK_TRUNCATION
MAP_FILE_INPUT x
MAP_FILE_OUTPUT x
TRACE x
NEWLINE
DISTFILE x
OMIT_HEADER
RUN_STAT_DBPARTNUM x
```

이들 옵션의 자세한 설명은 파티션된 데이터베이스 환경을 위한 로드 구성 옵션을 참조하십시오.

### **RESTARTCOUNT**

예약됨.

### **USING** *directory*

예약됨.

**예:**

**예 1**

TABLE1에는 다음과 같이 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 다음과 같이 7개의 요소가 있습니다.

- ELE1 위치 01 - 20

- ELE2 위치 21 - 22
- ELE3 위치 23 - 23
- ELE4 위치 24 - 27
- ELE5 위치 28 - 31
- ELE6 위치 32 - 32
- ELE7 위치 33 - 40

데이터 레코드:

```

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 있는 striptblanks 스펙은 VARCHAR(예를 들어, 각각 1, 2, 3행에서 길이가 11, 17, 19바이트인 COL1) 컬럼에 있는 공백을 강제로 절단합니다.
2. MODIFIED BY 매개변수에 있는 reclen=40 스펙은 각 입력 레코드 끝에 줄바꾸기 문자가 없으며 각 레코드의 길이가 40바이트임을 나타냅니다. 마지막 8바이트는 테이블을 로드하는 데 사용되지 않습니다.
3. 입력 파일에 COL4가 제공되어 있지 않으므로 이 컬럼은 TABLE1에 해당 디폴트 값으로 삽입됩니다(NOT NULL WITH DEFAULT로 정의됨).
4. 위치 23과 32는 TABLE1의 COL2 및 COL3이 주어진 행에 대해 널(NULL)로 로드될 것인지를 나타내는 데 사용됩니다. 주어진 레코드의 컬럼 널(NULL) 표시기 위치에 Y가 들어 있으면 이 컬럼은 널(NULL)이 됩니다. N이 들어 있으면 입력 레코드의 컬럼 데이터 위치에 있는 데이터 값(L(.....)에 정의됨)을 해당 행의 컬럼 데이터 소스로 사용합니다. 이 예에서 1행에는 널(NULL) 컬럼이 없고, 2행에서는 COL2가 널(NULL)이고 3행에서는 COL3이 널(NULL)입니다.
5. 이 예에서는 COL1과 COL5에 대해 NULL INDICATORS가 0(영)으로 지정되어 데이터가 널(NULL) 입력이 가능하지 않음을 나타냅니다.
6. 주어진 컬럼에 대한 NULL INDICATOR는 입력 레코드에서 어디서나 사용될 수 있지만 해당 위치가 지정되어야 하며 Y 또는 N값이 제공되어야 합니다.

예 2(파일에서 LOB 로드)

TABLE1에는 다음과 같이 3개의 컬럼이 있습니다.

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

ASCFILE1에는 다음과 같이 3개의 요소가 있습니다.

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

다음 파일은 /u/user1 또는 /u/user1/bin 중 하나에 있습니다.

- ASCFILE2에 LOB 데이터가 있음
- ASCFILE3에 LOB 데이터가 있음
- ASCFILE4에 LOB 데이터가 있음
- ASCFILE5에 LOB 데이터가 있음
- ASCFILE6에 LOB 데이터가 있음
- ASCFILE7에 LOB 데이터가 있음

ASCFILE1에 있는 데이터 레코드:

```
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 매개변수에 있는 lobsinfile 스펙은 파일에서 모든 LOB 데이터가 로드될 것을 로더에 알려줍니다.
2. MODIFIED BY 매개변수에 있는 reclen=22 스펙은 각 입력 레코드 끝에 줄 바꿈 문자가 없으며 각 레코드의 길이가 22바이트 임을 나타냅니다.
3. LOB 데이터는 6개의 파일(ASCFILE2 - ASCFILE7)에 들어 있습니다. 각 파일에는 특정 행에 대한 LOB 컬럼을 로드하는 데 사용될 데이터가 들어 있습니다. LOB와 기타 데이터 간의 관계는 ASCFILE1에 지정되어 있습니다. 이 파일의 첫 번째 레코드는 로더에게 1행의 COL1에 REC1을 넣도록 지시합니다. ASCFILE2의 내용은 1행의 LOB1을 로드하는 데 사용되고, ASCFILE3의 내용은 1행의 LOB2

를 로드하는 데 사용됩니다. 마찬가지로 ASCFILE4와 ASCFILE5는 2행의 LOB1과 LOB2를 로드하는 데 사용되고, ASCFILE6과 ASCFILE7은 3행의 LOB를 로드하는 데 사용됩니다.

4. LOBS FROM 매개변수에는 로더가 요구할 때 이름 지정된 LOB 파일을 검색하는 두 개의 경로가 들어 있습니다.
5. lobsinfile 수정자를 사용하지 않고 ASCFILE1(컬럼 식별자가 없는 ASCII 파일)에서 직접 LOB를 로드하려면 다음 규칙을 따라야 합니다.
  - LOB를 포함하는 레코드의 총 길이는 32KB를 초과할 수 없습니다.
  - 입력 레코드의 LOB 필드는 고정 길이여야 하며, 필요한 대로 공백으로 채워진 LOB 데이터가 있어야 합니다.
  - striptblanks 수정자를 지정하여 데이터베이스에 LOB가 삽입될 때 LOB를 채운 뒤 공백이 제거될 수 있도록 해야 합니다.

### 예 3(덤프 파일 사용)

FRIENDS 테이블은 다음과 같이 정의됩니다.

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

이 테이블에 다음 데이터 레코드를 로드할 경우,

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

첫 번째 INT가 널(NULL)이고 컬럼 정의가 NOT NULL을 지정하므로 두 번째 행은 거부됩니다. DEL 형식과 일치하지 않는 초기 문자가 들어 있는 컬럼은 오류를 생성하므로 해당 레코드는 거부됩니다. 이러한 레코드는 덤프 파일에 쓰여집니다.

컬럼에서 컬럼 분리문자 밖에 나타나는 DEL 데이터는 무시되지만 경고를 생성합니다. 예를 들면, 다음과 같습니다.

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

유틸리티는 테이블의 세 번째 컬럼에 "sam"을 로드하고 "sdf" 문자는 경고로 플래그됩니다. 이 레코드는 거부되지 않습니다. 다른 예의 경우를 보십시오.

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

유틸리티는 22,34,"bob"를 로드하고 컬럼 1에서 22 다음에 오는 일부 데이터가 무시되었다는 경고를 생성합니다. 이 레코드는 거부되지 않습니다.

### 예 4(ID 컬럼이 있는 테이블 로드)

TABLE1에는 다음과 같이 4개의 컬럼이 있습니다.

- C1 VARCHAR(30)

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

TABLE2는 TABLE1과 동일한데 다른 점은 C2가 GENERATED ALWAYS 식별 컬럼인 점입니다.

DATAFILE1의 데이터 레코드(DEL 형식):

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

DATAFILE2의 데이터 레코드(DEL 형식):

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

주:

1. 다음 명령은 DATAFILE1에 1행과 2행에 대한 식별 값이 제공되지 않았기 때문에 이들 행에 대한 식별 값을 생성합니다. 그러나 3행과 4행에는 각각 사용자 제공 ID 값 100과 101이 지정됩니다.

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

2. DATAFILE1을 TABLE1로 로드하여 식별 값이 모든 행에 대해 생성되도록 하려면 다음 명령 중 하나를 발행하십시오.

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

3. DATAFILE2를 TABLE1로 로드하여 식별 값이 각 행에 대해 생성되도록 하려면 다음 명령 중 하나를 발행하십시오.

```
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 로드하여 3행과 4행에 100과 101의 식별 값이 지정되도록 하려면 다음 명령을 발행하십시오.

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

이 경우, 1행과 2행은 거부됩니다. 이는 유틸리티가 사용자 제공 값을 사용하기 위해 시스템 생성 식별 값을 겹쳐쓰도록 지시되었기 때문입니다. 그러나 사용자 제공 값이 없을 경우 식별 컬럼이 내재적으로 널(NULL)이 아니므로 이 행은 거부됩니다.

5. 식별 관련 파일 유형 수정자를 사용하지 않고 DATAFILE1을 TABLE2로 로드할 경우 1행과 2행은 로드되지만 3행과 4행은 거부됩니다. 이는 이들 행이 자체의 널 (NULL)이 아닌 값을 제공하며 식별 컬럼이 GENERATED ALWAYS이기 때문입니다.

**예 5(행 변경 시간소인 컬럼이 있는 테이블 로딩)**

TABLE1에는 다음과 같이 4개의 컬럼이 있습니다.

- C1 VARCHAR(30)
- C2 ROW CHANGE TIMESTAMP GENERATED BY DEFAULT
- C3 DECIMAL(7,2)
- C4 CHAR(1)

TABLE2는 TABLE1과 동일한데 다른 점은 C2가 ENERATED ALWAYS 컬럼인 점입니다.

DATAFILE1의 데이터 레코드(DEL 형식):

```
"Liszt"  
"Hummel",,187.43, H  
"Grieg", 2006-05-23-15.55.53.209971, 66.34, G  
"Satie", 2006-05-22-19.34.14.947681, 818.23, I
```

DATAFILE2의 데이터 레코드(DEL 형식):

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

주:

1. 다음 명령은 DATAFILE1에 1행과 2행에 대한 ROW CHANGE TIMESTAMP 값이 제공되지 않았기 때문에 이들 행에 대한 ROW CHANGE TIMESTAMP 값을 생성합니다. 그러나 3과 4행에는 각각 2006-05-23-15.55.53.209971 및 2006-05-22-19.34.14.947681의 사용자 제공 ROW CHANGE TIMESTAMP 값이 지정됩니다.

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

2. DATAFILE1을 TABLE1로 로드하여 ROW CHANGE TIMESTAMP 값이 모든 행에 대해 생성되도록 하려면 다음 명령 중 하나를 발행하십시오.

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

```
db2 load from datafile1.del of del modified by rowchangetimestampignore  
replace into table1
```

3. DATAFILE2를 TABLE1로 로드하여 ROW CHANGE TIMESTAMP 값이 각 행에 대해 생성되도록 하려면 다음 명령 중 하나를 발행하십시오.

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

```
db2 load from datafile2.del of del modified by rowchangetimestampmissing  
replace into table1
```

4. DATAFILE1을 TABLE2로 로드하여 3행과 4행에 ROW CHANGE TIMESTAMP 값 2006-05-23-15.55.53.209971 및 2006-05-22-19.34.14.947681이 지정되게 하려면, 다음 명령을 발행하십시오.

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

이 경우, 1행과 2행은 거부됩니다. 이는 유틸리티가 사용자 제공 값을 사용하기 위해 시스템 생성 ROW CHANGE TIMESTAMP 값을 겹쳐쓰도록 지시되었기 때문입니다. 그러나 사용자 제공 값이 없을 경우 행 변경 시간소인 컬럼이 내재적으로 널(NULL)이 아니므로 이 행은 거부됩니다.

5. ROW CHANGE 관련 파일 유형 수정자를 사용하지 않고 DATAFILE1을 TABLE2로 로드할 경우 1행과 2행은 로드되지만 3행과 4행은 거부됩니다. 이는 이들 행 자체의 널(NULL)이 아닌 값을 제공하며 시간소인 컬럼이 GENERATED ALWAYS이기 때문입니다.

#### 예 6(CURSORS 파일 유형을 사용하여 로드)

ABC.TABLE1 테이블에는 다음과 같이 3개의 컬럼이 있습니다.

```
ONE INT  
TWO CHAR(10)  
THREE DATE
```

테이블 ABC.TABLE2에는 세 개의 컬럼이 있습니다.

```
ONE VARCHAR  
TWO INT  
THREE DATE
```

다음 명령을 실행하면 ABC.TABLE1의 모든 데이터가 ABC.TABLE2로 로드됩니다.

```
db2 declare mycurs cursor for select two,one,three from abc.table1  
db2 load from mycurs of cursor insert into abc.table2
```

ABC.TABLE1이 ABC.TABLE2가 있는 데이터베이스와 다른 데이터베이스에 있는 경우, 로드시 DECLARE CURSOR 명령의 DATABASE, USER 및 USING 옵션을 사용할 수 있습니다. 예를 들어, ABC.TABLE1이 DB1 데이터베이스에 있고 DB1의 사용자 ID 및 암호가 각각 user1 및 pwd1인 경우, 다음 명령을 실행하면 ABC.TABLE1의 모든 데이터가 ABC.TABLE2로 로드됩니다.

```
db2 declare mycurs cursor database DB1 user user1 using pwd1  
for select two,one,three from abc.table1  
db2 load from mycurs of cursor insert into abc.table2
```



## XML 문서에서 데이터 로드 예

### XML 데이터 로딩

#### 예 1

테이블에 삽입되는 문서를 설명하기 위해 사용자가 XDS 필드로 데이터 파일을 구성했습니다. 다음과 같이 표시됩니다.

```
1, "<XDS FIL=""file1.xml"" />"
2, "<XDS FIL='file2.xml' OFF='23' LEN='45' />"
```

첫 번째 행의 경우, XML 문서는 file1.xml 파일로 식별됩니다. 문자 분리문자는 큰따옴표이며 큰따옴표는 XDS 안에 존재하므로, XDS 안에 포함된 큰따옴표는 이중이 됩니다. 두 번째 행의 경우, XML 문서는 file2.xml 파일로 식별되며, 바이트 오프셋 23에서 시작하며 45바이트 길이입니다.

#### 예 2:

사용자는 XML 컬럼의 구문 분석 또는 유효성 확인 옵션 없이 로드 명령을 발행하며, 데이터는 정상적으로 로드됩니다.

```
LOAD FROM data.del of DEL INSERT INTO mytable
```

### 커서에서 XML 데이터 로딩

커서에서 데이터 로딩은 일반 관계형 컬럼 유형과 동일합니다. 두 개의 테이블 T1 및 T2가 있으며, 각각 C1이라는 단일 XML 컬럼으로 구성됩니다. T1에서 T2로 로드하기 위해 사용자는 우선 커서를 선언합니다.

```
DECLARE X1 CURSOR FOR SELECT C1 FROM T1;
```

다음으로 사용자는 커서 유형을 사용하여 LOAD를 발행할 수 있습니다.

```
LOAD FROM X1 of CURSOR INSERT INTO T2
```

XML 특정 LOAD 옵션을 커서 유형에 적용하는 것은 파일에서 로드하는 것과 같습니다.

## XMLVALIDATE절 사용의 예

### XMLVALIDATE USING XDS

#### 예 1

```
USING XDS DEFAULT schema-sqlid
```

사용자는 XDS에 표시된 스키마에 따라 유효성을 확인하려고 합니다. SCH 속성이 없는 XDS 값이 있는 경우, 이 문서는 이 절에서 사용되는 스키마에 대해 유효성 확인됩니다.

```
XMLVALIDATE USING XDS DEFAULT S1.SCHEMA_B
```

#### 예 2:

사용자는 XDS에 언급된 특정 스키마를 유효성 확인하지만 무시합니다.

```
XMLVALIDATE USING XDS IGNORE S1.SCHEMA_C
```

#### 예 3:

사용자는 XDS의 일부 스키마를 유효성 확인하지만 다시 맵핑합니다.

```
XMLVALIDATE USING XDS MAP( (S1.SCHEMA_A, S2.SCHEMA_B ), (S3.SCHEMA_C, S5.SCHEMA_E) )
```

위의 XDS의 경우, S1.SCHEMA\_A의 SCH 속성을 갖는 문서는 S2.SCHEMA\_B에 대해 유효성 확인됩니다. 또한 S3.SCHEMA\_C의 SCH 속성을 갖는 문서는 S5.SCHEMA\_E에 대해 유효성 확인됩니다.

#### 예 4

사용자는 DEFAULT, IGNORE 및 MAP 옵션의 조합을 사용하려고 합니다.

```
XMLVALIDATE USING XDS
  DEFAULT S8.SCHEMA_H
  IGNORE (S9.SCHEMA_I, S10.SCHEMA_J)
  MAP ((S1.SCHEMA_A, S2.SCHEMA_B), (S3.SCHEMA_C, S5.SCHEMA_E),
      (S6.SCHEMA_F, S3.SCHEMA_C), (S4.SCHEMA_D, S7.SCHEMA_G))
```

다음 XDS를 포함하는 XML 컬럼의 경우:

```
<XDS FIL='xmlfile.001.xml' />
```

xmlfile.001.xml 파일의 문서를 유효성 확인하는 데 SQL ID "S8.SCHEMA\_H"를 사용하는 XML 스키마를 사용합니다. 이는 "S8.SCHEMA\_H"가 디폴트 스키마로 사용되도록 지정되었기 때문입니다.

다음 XDS를 포함하는 XML 컬럼의 경우:

```
<XDS FIL='xmlfile.002.xml' OFF='10' LEN='500' SCH='S10.SCHEMA_J' />
```

xmlfile.002.xml 파일의 문서에 대해 스키마 유효성 확인이 발생하지 않습니다. 이는 XDS가 "S10.SCHEMA\_J"를 스키마로 사용하도록 지정하지만 이 스키마가 IGNORE 절에 포함되어 있기 때문입니다. 문서 내용은 파일의 바이트 오프셋 10(11번째 바이트를 의미함)에서 찾을 수 있으며 길이는 500바이트입니다.

다음 XDS를 포함하는 XML 컬럼의 경우:

```
<XDS FIL='xmlfile.003.xml' SCH='S6.SCHEMA_F' />
```

xmlfile.003.xml 파일의 문서를 유효성 확인하는 데 SQL ID "S3.SCHEMA\_C"를 사용하는 XML 스키마를 사용합니다. 이는 MAP절이 "S6.SCHEMA\_F" 스키마가 "S3.SCHEMA\_C" 스키마로 맵핑되도록 지정하기 때문입니다. 추가 맵핑이 일어나지 않으므로 이 경우 "S3.SCHEMA\_C" 스키마 대 "S5.SCHEMA\_E" 스키마의 맵핑이 적용되지 않는 점을 참고하십시오.

다음 XDS를 포함하는 XML 컬럼의 경우:

```
<XDS FIL='xmlfile.004.xml' SCH='S11.SCHEMA_K' />
```

xmlfile.004.xml 파일의 문서를 유효성 확인하는 데 SQL ID "S11.SCHEMA\_K"를 사용하는 XML 스키마를 사용합니다. 이 경우 DEFAULT, IGNORE 또는 MAP 권장 스펙 중 아무 것도 적용되지 않음을 참고하십시오.

## **XMLVALIDATE USING SCHEMA**

사용자는 단일 SCHEMA에 따라 모든 XML 문서의 유효성을 확인하려고 합니다. 다음 XMLVALIDATE절의 경우:

```
XMLVALIDATE USING SCHEMA S2.SCHEMA_B
```

다음 XDS를 포함하는 XML 컬럼의 경우:

```
<XDS FIL='xmlfile.001.xml' />
```

SQL ID "S2.SCHEMA\_B"를 사용하는 XML 스키마를 사용하여 xmlfile.001.xml 파일의 문서를 유효성 확인합니다.

다음 XDS를 포함하는 XML 컬럼의 경우:

```
<XDS FIL='xmlfile.002.xml' SCH='S1.SCHEMA_A' />
```

SQL ID "S2.SCHEMA\_B"를 사용하는 XML 스키마를 사용하여 xmlfile.002.xml 파일의 문서를 유효성 확인합니다. USING SCHEMA절에서 지정하는 스키마를 사용하여 유효성 확인을 수행하므로 SCH 속성이 무시되는 점을 참고하십시오.

## **XMLVALIDATE USING SCHEMALOCATION HINTS**

사용자는 문서 자체 내에 있는 스키마 정보의 유효성을 확인하려고 합니다. 다음 XMLVALIDATE 절의 경우:

```
XMLVALIDATE USING SCHEMALOCATION HINTS
```

다음 XDS를 포함하는 XML 컬럼의 경우:

```
<XDS FIL='xmlfile.001.xml' />
```

사용된 XML 스키마는 문서 내용의 schemaLocation 속성에 의해 판별되며 스키마가 존재하지 않으면 유효성 확인이 일어나지 않습니다.

다음 XDS를 포함하는 XML 컬럼의 경우:

```
<XDS FILE='xmlfile.002.xml' SCH='S1.SCHEMA_A' />
```

사용된 XML 스키마는 문서 내용의 schemaLocation 속성에 의해 판별되며 스키마가 존재하지 않으면 유효성 확인이 일어나지 않습니다. SCHEMALOCATION HINTS를 사용하여 유효성 확인을 수행하므로 SCH 속성이 무시됨을 참고하십시오.

## 사용 시 참고사항

- 데이터는 입력 파일에 나타나는 시퀀스로 로드됩니다. 특정 시퀀스를 원할 경우 로드를 시도하기 전에 데이터를 정렬해야 합니다. 소스 데이터 순서의 보존이 필요하지 않은 경우, 로드 유틸리티의 파일 유형 수정자 섹션에서 아래 설명된 ANYORDER 파일 유형 수정자 사용을 고려하십시오.
- 로드 유틸리티는 기존 정의에 따라 인덱스를 빌드합니다. 고유 키에 대한 중복을 처리하기 위해 예외 테이블을 사용합니다. 이 유틸리티는 참조 무결성을 강제하거나 제한조건 점검을 수행하거나 로드되는 테이블에 종속된 구체화된 쿼리 테이블을 갱신하지 않습니다. 참조 또는 점검 제한조건을 포함하는 테이블은 무결성 설정 보류 상태에 놓입니다. 로드되고 있는 테이블에 종속되고 REFRESH IMMEDIATE를 사용하여 정의된 요약 테이블도 무결성 설정 보류 상태에 놓입니다. 이들 테이블을 무결성 설정 보류 상태에서 해제하려면 SET INTEGRITY문을 발행하십시오. 복제된 구체화된 쿼리 테이블에서는 로드 조작을 수행할 수 없습니다.
- 테이블에 클러스터링 인덱스가 존재할 경우, 로드하기 전에 클러스터링 인덱스에 대해 데이터를 정렬해야 합니다. 그러나 다차원적으로 클러스터된(MDC) 테이블에 로드할 경우에는 미리 데이터를 정렬할 필요가 없습니다.
- 보호 설정된 테이블에 로드할 때 예외 테이블을 지정하면, 유효하지 않은 보안 레이블로 보호 설정된 행은 이 테이블로 보내집니다. 따라서 예외 테이블에 액세스할 수 있는 사용자는 일반적으로 액세스 권한이 없는 데이터에 액세스할 수 있습니다. 보안 강화를 위해 누구에게 예외 테이블 액세스 권한을 부여할 것인지 유의하고, 행을 수리한 후 로드되고 있는 테이블에 복사한 후에는 곧바로 각 행을 삭제하고 예외 테이블 사용을 완료하면 곧바로 예외 테이블을 삭제하십시오.
- 내부 형식으로 된 보안 레이블에는 줄 바꾸기 문자가 포함될 수 있습니다. DEL 파일 형식을 사용하여 파일을 로드할 경우, 이들 줄 바꾸기 문자는 분리문자로 오인될 수 있습니다. 이러한 문제점이 발생하면 LOAD 명령에 delprioritychar 파일 유형 수정자를 지정하여 분리문자에 대해 이전 디폴트 우선순위를 사용하십시오.
- DECLARE CURSOR 명령 중에 DATABASE 키워드가 지정된 CURSOR 파일 유형을 사용하여 로드 조작을 수행할 경우, 현재(로드 중에) 연결된 데이터베이스에 대해 인증할 때 사용한 사용자 ID 및 암호를 사용하여 소스 데이터베이스 (DECLARE CURSOR 명령의 DATABASE 옵션에 의해 지정됨)에 대해서도 인증합니다. 로드 중

인 데이터베이스에 대한 연결용으로 지정된 사용자 ID 및 암호가 없을 경우 DECLARE CURSOR 명령 중에 소스 데이터베이스에 대한 사용자 ID 및 암호를 지정해야 합니다.

- 개별적 파트가 Windows 시스템에서 AIX 시스템으로 복사되는 다중 파트 PC/IXF 파일 로딩이 지원 됩니다. 모든 파일의 이름이 LOAD 명령에서 지정되어야 합니다. 예를 들어, LOAD FROM DATA.IXF, DATA.002 OF IXF INSERT INTO TABLE1. 논리적으로 분할된 PC/IXF 파일에서 Windows 운영 체제로의 로딩은 지원되지 않습니다.
- 실패한 LOAD 재시작 시, 인덱스의 REBUILD 모드를 사용하도록 BUILD 단계를 강제하는 기존 동작을 따릅니다.
- 데이터베이스 중간에서는 XML 문서를 로드할 수 없으며 이 경우 오류 메시지 SQL1407N이 리턴됩니다.

### LOAD TERMINATE 및 LOAD RESTART 사전 관리의 요약

다음 도표는 TERMINATE 지시문 아래에서 LOAD 처리의 압축 사전 관리 동작을 요약합니다.

표 21. LOAD TERMINATE 사전 관리

테이블 COMPRESS 속성	LOAD 이전에 테이블 행 데이터 사전이 존재합니까?	XML 스토리지 오브젝트 사전이 LOAD 이전에 존재합니다. <sup>1</sup>	TERMINATE: LOAD REPLACE KEEPDICTIONARY 또는 LOAD INSERT	TERMINATE: LOAD REPLACE RESETDICTIONARY
YES	YES	YES	기존 사전을 보존합니다.	어느 쪽 사전도 보존하지 않습니다. <sup>2</sup>
YES	YES	NO	기존 사전을 보존합니다.	어떤 것도 보존하지 않습니다. <sup>2</sup>
YES	NO	YES	기존 사전을 보존합니다.	어떤 것도 보존하지 않습니다.
YES	NO	NO	어떤 것도 보존하지 않습니다.	어떤 것도 보존하지 않습니다.
NO	YES	YES	기존 사전을 보존합니다.	어떤 것도 보존하지 않습니다.
NO	YES	NO	기존 사전을 보존합니다.	어떤 것도 보존하지 않습니다.
NO	NO	YES	기존 사전을 보존합니다.	어떤 것도 보존하지 않습니다.
NO	NO	NO	아무 것도 수행하지 않습니다.	아무 것도 수행하지 않습니다.

주:

1. XML 컬럼이 DB2 버전 9.7 이상에서 테이블에 추가된 경우 또는 테이블이 온라인 테이블 이동을 사용하여 이주된 경우에만 테이블의 XML 스토리지 오브젝트의 압축 사전이 작성될 수 있습니다.
2. 테이블에서 데이터 캡처가 사용 가능한 특수한 경우, 테이블 행 데이터 사전이 보존됩니다.

LOAD RESTART는 최종 일관성 지점까지 테이블을 절단합니다. 압축 사전은 LOAD RESTART 처리의 파트로서, 최종 LOAD 일관성 지점이 지정된 시간에 테이블에 존재합니다. 이 경우, LOAD RESTART는 새 사전을 작성하지 않습니다. 가능한 조건에 대한 요약은 아래 테이블 4를 참조하십시오.

표 22. LOAD RESTART 사전 관리

테이블 COMPRESS 속 성	LOAD 일관성 지점 이전 테이블 행 데이 터 사전이 존재합니까? <sup>1</sup>	최종 로드 이전에 XML 스토리지 오브젝트 사전 이 존재했습니까? <sup>2</sup>	RESTART: LOAD REPLACE KEEPDICTIONARY 또는 LOAD INSERT	RESTART: LOAD REPLACE RESETDICTIONARY
YES	YES	YES	기존 사전을 보존합니다.	기존 사전을 보존합니다.
YES	YES	NO	기존의 테이블 행 데이터 사전을 빌드하고 ADC에 따라 XML 사전을 빌드합니다.	기존의 테이블 행 데이터 사전을 빌드하고 XML 사전을 빌드합니다.
YES	NO	YES	ADC에 따른 테이블 행 사전을 빌드합니다. 기존 XML 사전을 보존합니다.	테이블 행 데이터 사전을 빌드합니다. 기존 XML 사전을 보존합니다.
YES	NO	NO	ADC에 따른 XML 사전과 테이블 행 데이터를 빌드합니다.	테이블 행 데이터 및 XML 사전을 빌드합니다.
NO	YES	YES	기존 사전을 보존합니다.	기존 사전을 제거합니다.
NO	YES	NO	기존 테이블 행 데이터 사전을 보존합니다.	기존 테이블 행 데이터 사전을 제거합니다.
NO	NO	YES	기존 XML 사전을 보존합니다.	기존 XML 사전을 제거합니다.
NO	NO	NO	아무 것도 수행하지 않습니다.	아무 것도 수행하지 않습니다.

주:

1. XML 데이터 로드 시, 조작 시작부터 로드 단계 재시작 중 실패하는 로드 조작의 경우 SAVECOUNT 옵션이 무시됩니다.
2. XML 컬럼이 DB2 버전 9.7 이상에서 테이블에 추가된 경우 또는 테이블이 온라인 테이블 이동을 사용하여 이주된 경우에만 테이블의 XML 스토리지 오브젝트의 압축 사전이 작성될 수 있습니다.

### 로드 유틸리티의 파일 유형 수정자

표 23. 로드 유틸리티의 유효한 파일 유형 수정자: 모든 파일 형식

수정자	설명
anyorder	이 수정자는 <code>cpu_parallelism</code> 매개변수와 결합하여 사용됩니다. SMP 시스템에서 중요한 추가 성능 이점이 발생하며, 소스 데이터 순서의 보존이 필요하지 않음을 지정합니다. <code>cpu_parallelism</code> 의 값이 1이면, 이 옵션은 무시됩니다. 일관성 지점 이후의 응급 복구는 데이터가 시퀀스로 로드되도록 요구하므로 <code>SAVECOUNT &gt; 0</code> 인 경우 이 옵션은 지원되지 않습니다.

표 23. 로드 유틸리티의 유효한 파일 유형 수정자: 모든 파일 형식 (계속)

수정자	설명
generatedignore	이 수정자는 생성된 모든 컬럼의 데이터가 데이터 파일에 표시되지만 무시해야 함을 로드 유틸리티에 알립니다. 이로 인해 생성된 모든 컬럼의 값이 유틸리티에 의해 생성됩니다. 이 수정자는 generatedmissing 또는 generatedoverride 수정자 중 하나와 함께 사용될 수 없습니다.
generatedmissing	이 수정자가 지정된 경우, 유틸리티는 입력 데이터 파일에는 생성된 컬럼의 데이터를 포함하지 않음(NULL 값도 비포함)을 가정합니다. 이것은 유틸리티에 의해 모든 컬럼의 값을 생성하는 결과를 가져옵니다. 이 수정자는 generatedignore 또는 generatedoverride 수정자 중 하나와 함께 사용될 수 없습니다.
generatedoverride	<p>이 수정자는 테이블에서 생성되는 모든 컬럼의 사용자 제공 데이터를 승인하도록 로드 유틸리티에 지시합니다(이러한 유형의 컬럼에 대한 일반 규칙과 반대). 다른 데이터베이스 시스템에서 데이터를 이주하거나, ROLLFORWARD DATABASE 명령에서 RECOVER DROPPED TABLE 옵션을 사용하여 복구된 데이터에서 테이블을 로드할 때 유용합니다. 이 수정자가 사용될 때, 데이터가 없는 행이나 널(null) 값을 허용하지 않는 생성된 컬럼의 NULL 데이터는 거부됩니다(SQL3116W). 해당 수정자 사용 시, 테이블이 무결성 설정 보류 상태에 놓입니다. 사용자 제공 값을 검증하지 않고 무결성 설정 보류 상태에서부터 테이블을 얻으려면, 로드 조작 이후에 다음 명령을 발행하십시오.</p> <pre>SET INTEGRITY FOR &lt; table-name &gt; GENERATED COLUMN IMMEDIATE UNCHECKED</pre> <p>무결성 설정 보류 상태에서부터 테이블을 얻고 사용자 제공 값의 검증을 강제하려면, 로드 조작 이후에 다음 명령을 발행하십시오.</p> <pre>SET INTEGRITY FOR &lt; table-name &gt; IMMEDIATE CHECKED.</pre> <p>이 수정자가 지정되고 임의의 파티셔닝 키, 차원 키 또는 분산 키에 생성된 컬럼이 있으면, LOAD 명령은 자동으로 수정자를 generatedignore로 변환하고 로드를 진행합니다. 이것은 생성된 모든 컬럼 값을 재생성하는 효과를 갖습니다.</p> <p>이 수정자는 generatedmissing 또는 generatedignore 수정자 중 하나와 함께 사용될 수 없습니다.</p>
identityignore	이 수정자는 ID 컬럼의 데이터가 데이터 파일에 표시되지만 무시해야 함을 로드 유틸리티에 알립니다. 이로 인해 유틸리티에 의해 모든 ID 값이 생성됩니다. GENERATED ALWAYS 및 GENERATED BY DEFAULT ID 컬럼 둘 다의 동작이 동일합니다. GENERATED ALWAYS 컬럼의 경우, 어떤 행도 거부되지 않음을 의미합니다. 이 수정자는 identitymissing 또는 identityoverride 수정자 중 하나와 함께 사용될 수 없습니다.
identitymissing	이 수정자가 지정된 경우, 유틸리티는 입력 데이터 파일에는 ID 컬럼의 데이터를 포함하지 않음(NULL 값도 비포함)을 가정하므로 각 행의 값을 생성합니다. GENERATED ALWAYS 및 GENERATED BY DEFAULT ID 컬럼 둘 다의 동작이 동일합니다. 이 수정자는 identityignore 또는 identityoverride 수정자 중 하나와 함께 사용될 수 없습니다.

표 23. 로드 유틸리티의 유효한 파일 유형 수정자: 모든 파일 형식 (계속)

수정자	설명
identityoverride	<p>GENERATED ALWAYS로 정의된 ID 컬럼이 로드되는 테이블에 있을 때에만 이 수정자를 사용해야 합니다. 해당 컬럼에 대해 명시적, 널(NULL)이 아닌 데이터를 승인하도록 유틸리티에 지시합니다(이러한 유형의 ID 컬럼에 대한 일반 규칙과 반대). 테이블이 GENERATED ALWAYS로 정의되어야 할 때 다른 데이터베이스 시스템에서 데이터를 이주하거나, ROLLFORWARD DATABASE 명령에서 DROPPED TABLE RECOVERY 옵션을 사용하여 복구된 데이터에서 테이블을 로드할 때 유용합니다. 이 수정자가 사용될 때, 데이터가 없는 행이나 ID 컬럼의 NULL 데이터는 거부됩니다(SQL3116W). 이 수정자는 identitymissing 또는 identityignore 수정자 중 하나와 함께 사용될 수 없습니다. 이 옵션 사용 시 로드 유틸리티는 테이블의 ID 컬럼에서 값의 고유성을 검증하거나 유지하려고 시도하지 않습니다.</p>
indexfreespace=x	<p>x는 0 - 99의 정수입니다. 로드로 인덱스를 재빌드할 때 여유 공간으로 남게 되는 각 인덱스 페이지의 백분율입니다. INDEXING MODE INCREMENTAL을 사용하여 로드할 때 이 옵션은 무시됩니다. 페이지의 첫 번째 항목은 제한없이 추가되며, 여유 공간 비율 임계값을 유지하도록 추후 항목이 추가됩니다. 디폴트값은 CREATE INDEX 시간에 사용됩니다.</p> <p>이 값은 CREATE INDEX문에 지정된 PCTFREE 값보다 우선합니다. indexfreespace 옵션은 인덱스 단말 페이지에만 영향을 미칩니다.</p>
lobsinfile	<p>lob-path는 LOB 데이터를 포함하는 파일에 대한 경로를 지정합니다. ASC, DEL 또는 IXF 로드 입력 파일은 LOB 컬럼에 LOB 데이터가 있는 파일의 이름을 포함합니다.</p> <p>이 옵션은 CURSOR 파일 유형과 함께 지원되지 않습니다.</p> <p>LOBS FROM 절은 『lobsinfile』 수정자가 사용될 때 LOB 파일이 위치하는 곳을 지정합니다. LOBS FROM 절은 내재적으로 LOBSINFILE 동작을 활성화합니다. LOBS FROM 절은 데이터 импорт 중 LOB 파일을 검색하기 위해 경로 목록을 LOAD 유틸리티로 전달합니다.</p> <p>각 경로는 LLS(Lob Location Specifier)에 의해 포인트되는 최소한 하나의 LOB를 포함하는 최소한 하나의 파일을 데이터 파일에 포함합니다. LLS는 LOB 파일 경로에 저장된 파일에서 LOB 위치의 문자열 표시입니다. LLS의 형식은 filename.ext.nnn.mmm이며, 여기서 filename.ext는 LOB를 포함하는 파일의 이름이며, nnn은 파일 내에서 LOB의 오프셋을 바이트로 나타낸 것이며, mmm은 LOB의 길이를 바이트로 나타낸 것입니다. 예를 들어, 문자열 db2exp.001.123.456/가 데이터 파일에 저장되는 경우, LOB는 db2exp.001 파일에서 오프셋 123에 위치하며 456바이트 길이입니다.</p> <p>널(NULL) LOB를 표시하려면, 크기를 -1로 입력하십시오. 크기가 0으로 지정되면, 길이가 0인 LOB로 처리됩니다. 길이가 -1인 널(NULL) LOBS의 경우, 오프셋 및 파일 이름은 무시됩니다. 예를 들어, 널(NULL) LOB의 LLS는 db2exp.001.7.-1/입니다.</p>
noheader	<p>헤더 검증 코드를 건너뛰니다(단일 파티션 데이터베이스 파티션 그룹에 있는 테이블로 조각을 로드하는 경우만 적용 가능).</p> <p>디폴트 MPP 로드(모드 PARTITION_AND_LOAD)가 단일 파티션 데이터베이스 파티션 그룹에 있는 테이블에 대해 사용되는 경우, 파일에 헤더가 없습니다. 그러므로 헤더 수정자는 필요하지 않습니다. LOAD_ONLY 모드가 사용되는 경우, 파일에 헤더가 있습니다. 헤더가 없는 수정자를 사용할 필요가 있는 유일한 환경은 헤더가 없는 파일을 사용하여 LOAD_ONLY를 수행하려고 하는 경우입니다.</p>
norowwarnings	<p>거부된 행에 대한 모든 경고를 제외시킵니다.</p>



표 23. 로드 유틸리티의 유효한 파일 유형 수정자: 모든 파일 형식 (계속)

수정자	설명
pagefreespace=x	x는 0 - 100의 정수입니다. 여유 공간으로 남게 되는 각 데이터 페이지의 백분율입니다. 최소 행 크기 때문에 지정된 값이 유효하지 않은 경우, (예를 들어, 길이가 최소한 3000바이트이고, x 값이 50인 행), 새 페이지에 행이 위치합니다. 100 값이 지정된 경우, 각 행은 새 페이지에 있습니다. 테이블의 PCTFREE 값은 페이지 당 지정된 여유 공간의 크기를 결정합니다. 로드 조작의 pagefreespace 값이나 테이블의 PCTFREE 값이 설정되지 않은 경우, 유틸리티는 각 페이지에 가능한 많은 공간을 채웁니다. pagefreespace로 설정된 값은 테이블에 지정된 PCTFREE 값을 겹쳐줍니다.
rowchangetimestampignore	이 수정자는 행 변경 시간소인 컬럼의 데이터가 데이터 파일에 표시되지만 무시해야 함을 로드 유틸리티에 알립니다. 이로 인해 유틸리티에 의해 모든 ROW CHANGE TIMESTAMP가 생성됩니다. GENERATED ALWAYS 및 GENERATED BY DEFAULT 컬럼 둘 다의 동작이 동일합니다. GENERATED ALWAYS 컬럼의 경우, 어떤 행도 거부되지 않음을 의미합니다. 이 수정자는 rowchangetimestampmissing 또는 rowchangetimestampoverride 수정자 중 하나와 함께 사용될 수 없습니다.
rowchangetimestampmissing	이 수정자가 지정된 경우, 유틸리티는 입력 데이터 파일에는 행 변경 시간소인 컬럼의 데이터를 포함하지 않음(NULL 값도 비포함)을 가정하므로 각 행의 값을 생성합니다. GENERATED ALWAYS 및 GENERATED BY DEFAULT 컬럼 둘 다의 동작이 동일합니다. 이 수정자는 rowchangetimestampignore 또는 rowchangetimestampoverride 수정자 중 하나와 함께 사용될 수 없습니다.
rowchangetimestampoverride	GENERATED ALWAYS로 정의된 행 변경 시간소인 컬럼이 로드되는 테이블에 있을 때에만 이 수정자를 사용해야 합니다. 해당 컬럼에 대해 명시적, 널(NULL)이 아닌 데이터를 승인하도록 유틸리티에 지시합니다(이러한 유형의 시간소인 컬럼에 대한 일반 규칙 과 반대). 테이블이 GENERATED ALWAYS로 정의되어야 할 때 다른 데이터베이스 시스템에서 데이터를 이주하거나, ROLLFORWARD DATABASE 명령에서 DROPPED TABLE RECOVERY 옵션을 사용하여 복구된 데이터에서 테이블을 로드할 때 유용합니다. 이 수정자가 사용될 때, 데이터가 없는 행이나 ROW CHANGE TIMESTAMP 컬럼의 NULL 데이터는 거부됩니다(SQL3116W). 이 수정자는 rowchangetimestampmissing 또는 rowchangetimestampignore 수정자 중 하나와 함께 사용될 수 없습니다. 이 옵션 사용 시 로드 유틸리티는 테이블의 행 시간소인 컬럼에서 값의 고유성을 검증하거나 유지하려고 시도하지 않습니다.
seclabelchar	입력 소스 파일의 보안 레이블이 다폴트로 인코딩된 숫자 형식이 아니라 보안 레이블 값의 문자열 형식임을 표시합니다. LOAD는 각 보안 레이블을 로드된 대로의 내부 형식으로 변환합니다. 문자열이 적절한 형식으로 되어 있지 않은 경우 행은 로드되지 않으며 경고 (SQLSTATE 01H53, SQLCODE SQL3242W)가 리턴됩니다. 문자열이 테이블을 보호하는 보안 규정 파트인 유효한 보안 레이블을 나타내지 않는 경우, 행은 로드되지 않으며 경고 (SQLSTATE 01H53, SQLCODE SQL3243W)가 리턴됩니다.  seclabelname 수정자가 지정된 경우 이 수정자를 지정할 수 없으며, 그렇지 않으면 로드 실패하며 오류(SQLCODE SQL3525N)가 리턴됩니다.  테이블이 단일 DB2SECURITYLABEL 컬럼으로 구성된 경우, 데이터 파일은 다음과 같습니다.  "CONFIDENTIAL:ALPHA:G2" "CONFIDENTIAL;SIGMA:G2" "TOP SECRET:ALPHA:G2"  이 데이터를 로드하거나 импорт하려면, SECLABELCHAR 파일 유형 수정자가 사용되어야 합니다.  LOAD FROM input.del OF DEL MODIFIED BY SECLABELCHAR INSERT INTO t1

표 23. 로드 유틸리티의 유효한 파일 유형 수정자: 모든 파일 형식 (계속)

수정자	설명
seclabelname	<p>입력 소스 파일의 보안 레이블이 디폴트로 인코딩된 숫자 형식이 아니라 이름으로 표시됨을 나타냅니다. LOAD는 이름이 있는 경우 적절한 보안 레이블로 변환합니다. 테이블을 보호하는 보안 규정에 대해 표시된 이름이 있는 보안 레이블이 없는 경우, 행은 로드되지 않으며 경고(SQLSTATE 01H53, SQLCODE SQL3244W)가 리턴됩니다.</p> <p>seclabelchar 수정자가 지정된 경우 이 수정자를 지정할 수 없으며, 그렇지 않으면 로드에 실패하며 오류(SQLCODE SQL3525N)가 리턴됩니다.</p> <p>테이블이 단일 DB2SECURITYLABEL 컬럼으로 구성된 경우, 데이터 파일은 다음과 유사한 보안 레이블 이름으로 구성됩니다.</p> <pre>"LABEL1" "LABEL1" "LABEL2"</pre> <p>이 데이터를 로드하거나 импорт하려면, SECLABELNAME 파일 유형 수정자가 사용되어야 합니다.</p> <pre>LOAD FROM input.del OF DEL MODIFIED BY SECLABELNAME INSERT INTO t1</pre> <p>주: 파일 유형이 ASC인 경우, 보안 레이블의 이름 다음에 오는 모든 스페이스는 이름의 일부로 해석됩니다. 이를 피하려면 stripblanks 파일 유형 수정자를 사용하여 스페이스가 제거되었는지 확인하십시오.</p>
totalfreespace=x	<p>x는 0 이상의 정수입니다. 이 값은 여유 공간으로 테이블의 끝에 추가되는 테이블에서 총 페이지의 백분율로 해석됩니다. 예를 들어, x가 20이며 데이터가 로드된 이후에 테이블에 100개의 데이터 페이지가 있는 경우, 20개의 추가적인 빈 페이지가 추가됩니다. 테이블의 총 데이터 페이지 수는 120입니다. 데이터 페이지 총 수는 테이블에서 인덱스 페이지의 수를 계산에 넣지 않습니다. 이 옵션은 인덱스 오브젝트에 영향을 미치지 않습니다. 이 옵션이 지정된 두 개의 로드가 완료되면, 두 번째 로드는 첫 번째 로드에서 끝에 추가한 여유 스페이스를 재사용하지 않습니다.</p>
usedefaults	<p>목표 테이블 컬럼의 소스 컬럼이 지정되었지만 하나 이상의 행 인스턴스에 대한 데이터를 포함하지 않은 경우, 디폴트값이 로드됩니다. 누락된 데이터의 예:</p> <ul style="list-style-type: none"> <li>• DEL 파일: 임의 수의 스페이스(" ,")로 구분되는 2개의 인접 컬럼 분리문자나 2개의 인접 컬럼 분리문자(",,")가 컬럼 값으로 지정됩니다.</li> <li>• DEL/ASC/WSF 파일: 행의 컬럼이 충분하지 않거나 행의 길이가 원래 스펙만큼 충분하지 않습니다. ASC 파일의 경우, NULL 컬럼 값은 명시적으로 누락된 것으로 간주되지 않으며 디폴트가 NULL 컬럼 값을 대신하지 않습니다. 숫자, 날짜, 시간 및 /시간소인 컬럼의 모든 공백 문자로 NULL 컬럼 값을 표시하거나, 컬럼이 NULL임을 표시하기 위해 모든 유형의 컬럼에 널(NULL) 표시기를 사용함으로써 NULL 컬럼 값을 표시합니다.</li> </ul> <p>이 옵션이 없는 경우, 소스 컬럼이 행 인스턴스의 데이터를 포함하지 않으면 다음 중 하나가 발생합니다.</p> <ul style="list-style-type: none"> <li>• DEL/ASC/WSF 파일: 컬럼에 널(NULL) 입력 가능한 경우 NULL이 로드됩니다. 컬럼이 널(NULL) 입력 가능하지 않으면 유틸리티가 행을 거부합니다.</li> </ul>

표 24. 로드 유틸리티의 유효한 파일 유형 수정자: ASCII 파일 형식(ASC/DEL)

수정자	설명
codepage=x	<p>x는 ASCII 문자열입니다. 해당 값은 입력 데이터 세트에서 데이터의 코드 페이지로서 해석됩니다. 로드 조작 중 이 코드 페이지에서 데이터베이스 코드 페이지로 문자 데이터(및 문자로 지정된 숫자 데이터)를 변환합니다.</p> <p>다음 규칙이 적용됩니다.</p> <ul style="list-style-type: none"> <li>• 순수 DBCS(그래픽), 혼합 DBCS 및 EUC의 경우, 분리문자는 x00 - x3F의 범위로 제한됩니다.</li> <li>• EBCDIC 코드 페이지에서 지정된 DEL 데이터의 경우, 분리문자는 shift-in 및 shift-out DBCS 문자와 일치하지 않을 수 있습니다.</li> <li>• nullindchar은 x20 - x7F 코드 포인트의 표준 ASCII 세트에 포함된 기호를 지정해야 합니다. 이것은 ASCII 기호 및 코드 포인트를 나타냅니다. 코드 포인트가 달라도 EBCDIC 데이터는 해당 기호를 사용할 수 있습니다.</li> </ul> <p>이 옵션은 CURSOR 파일 유형과 함께 지원되지 않습니다.</p>
dateformat="x"	<p>x는 소스 파일에서 날짜의 형식입니다.<sup>1</sup> 유효한 날짜 요소는 다음과 같습니다.</p> <p>YYYY - Year(0000 - 9999 사이의 4자리 숫자)  M - Month(1 - 12 사이의 1 또는 2자리 숫자)  MM - Month(1 - 12 사이의 2자리 숫자,  M과 상호 독립적)  D - Day(1 - 31 사이의 1 또는 2자리 숫자)  DD - Day(1 - 31 사이의 2자리 숫자,  D와 상호 독립적)  DDD - Day of the year(001 - 366 사이의 3자리 숫자,  다른 day 또는 month 요소와 상호 독립적)</p> <p>지정되지 않은 각 요소에 대해 디폴트값 1이 지정됩니다. 날짜 형식의 예:</p> <p>"D-M-YYYY"  "MM.DD.YYYY"  "YYYYDDD"</p>
dumpfile = x	<p>x는 거부된 행이 쓰여진 예외 파일의 완전한 이름(서버 데이터베이스 파티션에 따른)입니다. 레코드당 최대 32KB의 데이터가 쓰여집니다. 다음은 덤프 파일을 지정하는 방법을 표시하는 예입니다.</p> <pre>db2 load from data of del modified by dumpfile = /u/user/filename insert into table_name</pre> <p>인스턴스 소유자가 파일을 작성하고 소유합니다. 디폴트 파일 권한을 겹쳐쓰려면, dumpfileaccessall 파일 유형 수정자를 사용하십시오.</p> <p>주:</p> <ol style="list-style-type: none"> <li>1. 파티션된 데이터베이스 환경에서, 경로는 로딩 데이터베이스 파티션에 로컬이어야 하므로 현재 실행 중인 로드 조작은 동일한 파일에 쓰지 않습니다.</li> <li>2. 파일의 콘텐츠는 비동기 버퍼 지정 모드로 디스크에 쓰여집니다. 실패하거나 인터럽트된 로드 조작의 경우, 디스크에 커밋된 레코드 수는 확실히 알 수 없으며 LOAD RESTART 이후에 일관성을 보장할 수 없습니다. 파일에서는 단일 패스로 시작하고 완료하는 로드 조작에 대해서만 완료될 것으로 가정합니다.</li> <li>3. 지정된 파일이 이미 존재하는 경우, 재작성되지 않지만 추가됩니다.</li> </ol>

표 24. 로드 유틸리티의 유효한 파일 유형 수정자: ASCII 파일 형식(ASC/DEL) (계속)

수정자	설명
dumpfileaccessall	<p>덤프 파일 작성 시 'OTHERS'에 대한 읽기 액세스를 권한 부여합니다.</p> <p>이 파일 유형 수정자는 다음 경우에만 유효합니다.</p> <ol style="list-style-type: none"> <li>1. dumpfile 파일 유형 수정자와 결합하여 사용됩니다.</li> <li>2. 사용자는 로드 목표 테이블에 대한 SELECT 특권을 갖습니다.</li> <li>3. UNIX 운영 체제에 있는 DB2 서버 데이터베이스 파티션에서 발행됩니다.</li> </ol> <p>지정된 파일이 이미 존재하는 경우, 권한을 변경하지 못합니다.</p>
fastparse	<p>주의와 함께 사용합니다. 사용자 제공 컬럼 값에 대한 구문 검사를 줄이고, 성능을 향상시킵니다. 테이블은 구조적으로 옳은 것으로 보장되지만(유틸리티는 충분한 데이터 검사를 수행하여 분할 위반 또는 트랩을 방지함), 데이터 일관성의 유효성은 확인되지 않습니다. 데이터가 일관되고 올바름을 확인하는 경우에만 이 옵션을 사용하십시오. 예를 들어, 사용자 제공 데이터가 유효하지 않은 시간소인 컬럼 값 :1&gt;0-00-20-07.11.12.000000을 포함하는 경우, FASTPARSE가 지정되면 이 값이 테이블에 삽입되며 FASTPARSE가 지정되지 않으면 거부됩니다.</p>
implieddecimal	<p>내포된 소수점의 위치는 컬럼 정의로 판별되며, 값의 끝으로 가정되지 않습니다. 예를 들어, 12345 값은 12345.00이 아닌 123.45로 DECIMAL(8,2) 컬럼에 로드됩니다.</p> <p>이 수정자는 packeddecimal 수정자와 함께 사용될 수 없습니다.</p>
timeformat="x"	<p>x는 소스 파일에서 시간의 형식입니다.<sup>1</sup> 유효한 시간 요소는 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>H - Hour(12시간 시스템의 경우 0 - 12 범위의 0 - 12 범위의 2자리 숫자, 24시간 시스템의 경우 0 - 24)</li> <li>HH - Hour(12시간 시스템의 경우 0 - 12 범위의 2자리 숫자, 24시간 시스템의 경우 0 - 24. H와 상호 독점적)</li> <li>M - Minute(0 - 59 사이의 1 또는 2자리 숫자)</li> <li>MM - Minute(0 - 59 사이의 2자리 숫자, M과 상호 독점적)</li> <li>S - Second(0 - 59 사이의 1 또는 2자리 숫자)</li> <li>SS - Second(0 - 59 사이의 2자리 숫자, S와 상호 독점적)</li> <li>SSSSS - Second of the day after midnight(00000 - 86399 사이의 5자리 숫자, 다른 시간 요소와 상호 독점적)</li> <li>TT - Meridian indicator(AM 또는 PM)</li> </ul> <p>지정되지 않은 각 요소에 대해 디폴트값 0이 지정됩니다. 시간 형식의 예:</p> <pre>"HH:MM:SS" "HH.MM TT" "SSSSS"</pre>

표 24. 로드 유틸리티의 유효한 파일 유형 수정자: ASCII 파일 형식(ASC/DEL) (계속)

수정자	설명
timestampformat="x"	<p>x는 소스 파일에서 시간소인의 형식입니다.<sup>1</sup> 유효한 시간소인 요소는 다음과 같습니다.</p> <p>YYYY - Year(0000 - 9999 사이의 4자리 숫자)  M - Month(1 - 12 사이의 1 또는 2자리 숫자)  MM - Month(01 - 12 사이의 2자리 숫자,  M 및 MM은 상호 독점적)  MMM - Month(월 이름으로 사용되는 3자리 문자로 대소문자를 구분하지 않으며,  M 및 MM은 상호 독점적)  D - Day(1 - 31 사이의 1 또는 2자리 숫자)  DD - Day(1 - 31 사이의 2자리 숫자, D와는 상호 독점적)  DDD - Day of the year(001 - 366 사이의 3자리 숫자,  다른 day 또는 month 요소와 상호 독점적)  H - Hour(12시간 시스템의 경우 0 - 12 범위의 1 또는 2자리 숫자,  24시간 시스템의 경우 0 - 24)  HH - Hour(12시간 시스템의 경우 0 - 12 범위의 2자리 숫자,  24시간 시스템의 경우 0 - 24.  H와는 상호 독점적)  M - Minute(0 - 59 사이의 1 또는 2자리 숫자)  MM - Minute(0 - 59 사이의 2자리 숫자,  M, minute와 상호 독점적)  S - Second(0 - 59 사이의 1 또는 2자리 숫자)  SS - Second(0 - 59 사이의 2자리 숫자,  S와 상호 독점적)  SSSSS - Second of the day after midnight(  00000 - 86399 사이의 5자리 숫자, 다른 시간 요소와  상호 독점적)  U(1 - 12회)  - Fractional seconds(U 어커런스 수는 각 자리가 0 - 9 사이의  자리 수를 나타냄)  TT - Meridian indicator(AM 또는 PM)</p>

표 24. 로드 유틸리티의 유효한 파일 유형 수정자: ASCII 파일 형식(ASC/DEL) (계속)

수정자	설명
timestampformat="x"(계속)	<p>디폴트값 1이 미지정된 YYYY, M, MM, D, DD 또는 DDD 요소에 지정됩니다. 디폴트값 'Jan'이 미지정된 MMM 요소에 지정됩니다. 미지정된 다른 모든 요소에 디폴트값 0이 지정됩니다. 다음은 시간소인 형식의 예입니다.</p> <p style="text-align: center;">"YYYY/MM/DD HH:MM:SS.UUUUUU"</p> <p>MMM 요소의 올바른 값은 다음을 포함합니다. 'jan', 'feb', 'mar', 'apr', 'may', 'jun', 'jul', 'aug', 'sep', 'oct', 'nov' 및 'dec'. 이들 값은 대소문자가 구분되지 않습니다.</p> <p>TIMESTAMPFORMAT 수정자가 지정되지 않으면, 로드 유틸리티는 두 개의 가능한 형식 중 하나를 사용하여 시간소인 필드를 형식화합니다.</p> <p>YYYY-MM-DD-HH.MM.SS YYYY-MM-DD HH:MM:SS</p> <p>로드 유틸리티는 DD와 HH 간의 구분자를 확인하여 형식을 선택합니다. 대시 '-'인 경우, 로드 유틸리티는 일반 대시와 점 형식(YYYY-MM-DD-HH.MM.SS)을 사용합니다. 공백인 경우, 로드 유틸리티는 HH, MM 및 SS를 구분하기 위해 콜론 ':'을 예상합니다.</p> <p>어느 쪽 형식이든, 마이크로초 필드(UUUUUU)를 포함하는 경우, 로드 유틸리티는 점 '.'을 구분자로 예상합니다. YYYY-MM-DD-HH.MM.SS.UUUUUU 또는 YYYY-MM-DD HH:MM:SS.UUUUUU를 승인할 수 있습니다.</p> <p>다음 예는 사용자 정의 날짜 및 시간 형식을 포함하는 데이터를 schedule이라는 테이블로 로드하는 방법을 설명합니다.</p> <pre>db2 load from delfile2 of del modified by timestampformat="yyyy.mm.dd hh:mm tt" insert into schedule</pre>
usegraphiccodepage	<p>usegraphiccodepage가 제공되면, 그래픽 또는 2바이트 문자 대형 오브젝트(DBCLOB) 데이터 필드로 로드되는 데이터는 그래픽 코드 페이지에 있는 것으로 가정합니다. 나머지 데이터는 문자 코드 페이지에 있다고 가정합니다. 그래픽 코드 페이지는 문자 코드 페이지와 연관됩니다. LOAD는 codepage 수정자가 지정된 경우 이를 통해 문자 코드 페이지를 판별하고 codepage 수정자가 지정되지 않은 경우 데이터베이스의 코드 페이지를 통해 문자 코드 페이지를 판별합니다.</p> <p>복구 중인 테이블에 그래픽 데이터가 있는 경우에만 삭제(drop) 테이블 복구로 생성되는 구분된 데이터 파일과 결합하여 이 수정자를 사용해야 합니다.</p> <p><b>제한사항</b></p> <p>이들 파일이 오직 하나의 코드 페이지에 인코딩된 데이터를 포함하면, usegraphiccodepage 수정자는 EXPORT 유틸리티로 작성된 DEL 파일과 함께 지정되지 않아야 합니다. usegraphiccodepage 수정자는 파일의 2바이트 문자 대형 오브젝트(DBCLOB)에서 무시됩니다.</p>

표 24. 로드 유틸리티의 유효한 파일 유형 수정자: ASCII 파일 형식(ASC/DEL) (계속)

수정자	설명
xmlchar	<p>XML 문서가 문자 코드 페이지에서 인코딩됨을 지정합니다.</p> <p>이 옵션은 지정된 문자 코드 페이지에서 인코딩되었지만 인코딩 선언을 포함하지 않는 XML 문서 처리에 유용합니다.</p> <p>각 문서에서 선언 태그가 존재하고 인코딩 속성을 포함하는 경우 인코딩은 문자 코드 페이지와 일치해야 하며, 그렇지 않으면 문서를 포함하는 행이 거부됩니다. 문자 코드 페이지는 codepage 파일 유형 수정자에서 지정한 값이거나, 지정되지 않은 경우 응용프로그램 코드 페이지입니다. 디폴트로 문서가 유니코드로 인코딩되거나 인코딩 속성을 가진 선언 태그를 포함합니다.</p>
xmlgraphic	<p>XML 문서가 지정된 그래픽 코드 페이지에서 인코딩됨을 지정합니다.</p> <p>이 옵션은 특정 그래픽 코드 페이지에서 인코딩되었지만 인코딩 선언을 포함하지 않는 XML 문서 처리에 유용합니다.</p> <p>각 문서에서 선언 태그가 존재하고 인코딩 속성을 포함하는 경우 인코딩은 그래픽 코드 페이지와 일치해야 하며, 그렇지 않으면 문서를 포함하는 행이 거부됩니다. 그래픽 코드 페이지는 codepage 파일 유형 수정자에서 지정한 값의 그래픽 구성요소이거나, 지정되지 않은 경우 응용프로그램 코드 페이지의 그래픽 구성요소입니다. 디폴트로 문서가 유니코드로 인코딩되거나 인코딩 속성을 가진 선언 태그를 포함합니다.</p>

표 25. 로드 유틸리티의 유효한 파일 유형 수정자: ASCII 파일 형식(구분되지 않은 ASCII)

수정자	설명
binarynumerics	<p>숫자(10진수는 아님) 데이터는 문자 표시가 아닌 2진 양식이어야 합니다. 손실이 큰 변환은 피합니다.</p> <p>이 옵션은 reclen 옵션으로 지정된 고정 길이 레코드를 사용하며 위치상 ASC에서만 지원됩니다.</p> <p>다음 규칙이 적용됩니다.</p> <ul style="list-style-type: none"> <li>• 데이터 유형 간의 변환은 수행되지 않으며, BIGINT, INTEGER 및 SMALLINT 예외가 있습니다.</li> <li>• 데이터 길이는 목표 컬럼 정의와 일치해야 합니다.</li> <li>• FLOAT는 IEEE 부동 소수점 형식이어야 합니다.</li> <li>• 로드 소스 파일의 2진 데이터는 로드 조작이 실행되는 플랫폼에 관계 없이 빅 엔디안(big-endian)으로 가정됩니다.</li> </ul> <p>이 수정자에 의해 영향을 받는 컬럼의 데이터에 NULL 값을 표시할 수 없습니다. 이 수정자가 사용될 때 공백(보통 NULL로 해석)은 2진 값으로 해석됩니다.</p>
nochecklengths	<p>nochecklengths가 지정되면, 목표 테이블 컬럼의 크기를 초과하는 컬럼 정의가 소스 데이터에 있는 경우에도 각 행을 로드하려고 시도합니다. 코드 페이지 변환으로 소스 데이터가 축소되는 경우 그러한 행은 정상적으로 로드될 수 있습니다. 예를 들어, 소스에서 4바이트 EUC 데이터는 목표에서 2바이트 DBCS 데이터로 축소되어 반 정도의 스페이스가 필요합니다. 불일치 컬럼 정의에도 불구하고 소스 데이터가 모든 경우에 맞는 경우 특히 이 옵션이 유용합니다.</p>

표 25. 로드 유틸리티의 유효한 파일 유형 수정자: ASCII 파일 형식(구분되지 않은 ASCII) (계속)

수정자	설명
nullindchar=x	<p>x는 단일 문자입니다. 널(NULL) 값을 나타내는 문자를 x로 변경합니다. x의 디폴트값은 Y입니다. <sup>2</sup></p> <p>문자가 영문자인 경우를 제외하고 EBCDIC 데이터 파일의 경우 이 수정자의 대소문자를 구분합니다. 예를 들어, 널(NULL) 표시기 문자가 N 문자가 되도록 지정되는 경우, n은 널(NULL) 표시기로 인식됩니다.</p>
packeddecimal	<p>binarynumerics 수정자가 DECIMAL 필드 유형을 포함하지 않으므로 압축 10진수 데이터를 직접 로드합니다.</p> <p>이 옵션은 reclen 옵션으로 지정된 고정 길이 레코드를 사용하며 위치상 ASC에서만 지원됩니다.</p> <p>지원되는 부호 니블의 값은 다음과 같습니다.</p> <p style="margin-left: 40px;">+ = 0xC 0xA 0xE 0xF - = 0xD 0xB</p> <p>이 수정자에 의해 영향을 받는 컬럼의 데이터에 NULL 값을 표시할 수 없습니다. 이 수정자가 사용될 때 공백(보통 NULL로 해석)은 2진 값으로 해석됩니다.</p> <p>서버 플랫폼에 관계 없이, 로드 소스 파일의 2진 데이터의 바이트 순서는 빅 엔디안(big-endian)으로 가정되므로, Windows 운영 체제에서 이 수정자를 사용하면 바이트 순서는 바뀌지 않아야 합니다.</p> <p>이 수정자는 implieddecimal 수정자와 함께 사용될 수 없습니다.</p>
reclen=x	<p>x는 최대값이 32 767인 정수입니다. 각 행에 대해 x 문자가 읽혀지며 행의 끝을 표시하기 위한 줄 바꾸기 문자는 사용되지 않습니다.</p>
striptblanks	<p>데이터를 변수 길이 필드로 로드할 때 뒤 공백을 절단합니다. 이 옵션이 지정되지 않으면, 공백이 보존됩니다.</p> <p>이 옵션은 striptnulls와 함께 지정될 수 없습니다. 이들은 상호 독점 옵션입니다. 이 옵션은 이전 호환성을 위해서만 지원되는 사용 안하는 t 옵션을 교체합니다.</p>
striptnulls	<p>데이터를 변수 길이 필드로 로드할 때 뒤 NULL 값(0x00 문자)을 절단합니다. 이 옵션이 지정되지 않으면, NULL 값이 보존됩니다.</p> <p>이 옵션은 striptblanks와 함께 지정될 수 없습니다. 이들은 상호 독점 옵션입니다. 이 옵션은 이전 호환성을 위해서만 지원되는 사용 안하는 padwithzero 옵션을 교체합니다.</p>
zoneddecimal	<p>BINARYNUMERICS 수정자는 DECIMAL 필드 유형을 포함하지 않으므로 존 10진수 (zoned DECIMAL) 데이터를 로드합니다. 이 옵션은 RECLEN 옵션으로 지정된 고정 길이 레코드를 사용하며 위치상 ASC에서만 지원됩니다.</p> <p>반 바이트 기호 값은 다음 중 하나일 수 있습니다.</p> <p style="margin-left: 40px;">+ = 0xC 0xA 0xE 0xF - = 0xD 0xB</p> <p>지원되는 숫자 값은 0x0 - 0x9입니다.</p> <p>지원되는 존 값은 0x3 및 0xF입니다.</p>



표 26. 로드 유틸리티의 유효한 파일 유형 수정자: DEL 파일 형식(컬럼 식별자가 있는 ASCII)

수정자	설명
chardelx	<p>x는 단일 문자열 분리문자입니다. 디폴트값은 큰따옴표(")입니다. 큰따옴표 대신 지정된 문자를 사용하여 문자열을 묶습니다. <sup>23</sup> 명시적으로 큰따옴표(")를 문자열 분리문자로 지정하려는 경우 다음과 같이 지정해야 합니다.</p> <pre>modified by chardel""</pre> <p>다음과 같이 작은따옴표(')를 문자열 분리문자로 지정할 수도 있습니다.</p> <pre>modified by chardel''</pre>
coldelx	<p>x는 단일 문자 컬럼 분리문자입니다. 디폴트값은 쉼표(,)입니다. 지정된 문자가 쉼표 대신 사용되어 컬럼 끝을 나타냅니다.<sup>23</sup></p>
decplusblank	<p>플러스 부호 문자. 플러스 부호(+) 대신 공백을 양의 10진수 값 앞에 붙입니다. 디폴트 조치는 플러스 부호를 양의 10진수 값 앞에 붙이는 것입니다.</p>
decptx	<p>x는 소수점 문자로 마침표를 대신하는 단일 문자입니다. 디폴트값은 마침표(.)입니다. 마침표 대신 지정된 문자가 소수점 문자로 사용됩니다.<sup>23</sup></p>
delprioritychar	<p>분리문자의 현재 디폴트 우선순위: 레코드 구분 문자, 문자 분리문자, 컬럼 분리문자. 이 수정자는 분리문자 우선순위를 문자 분리문자, 레코드 구분 문자, 컬럼 분리문자로 되돌림으로써 이전 우선순위에 따른 기존 응용프로그램을 보호합니다. 구분:</p> <pre>db2 load ... modified by delprioritychar ...</pre> <p>예를 들어, 다음과 같은 DEL 데이터 파일이 있습니다.</p> <pre>"Smith, Joshua",4000,34.98&lt;row delimiter&gt; "Vincent,&lt;row delimiter&gt;, is a manager", ... ... 4005,44.37&lt;row delimiter&gt;</pre> <p>delprioritychar 수정자를 지정하고, 이 데이터 파일에는 두 개의 행만이 있습니다. 두 번째 &lt;행 분리문자&gt;는 두 번째 행의 첫 번째 데이터 컬럼의 파트로 해석되지만, 첫 번째 및 세 번째 &lt;행 분리문자&gt;는 실제 레코드 구분 문자로 해석됩니다. 이 수정자가 지정되지 않은 경우, 이 데이터 파일에는 세 개의 행이 있으며, 각 행은 &lt;행 분리문자&gt;로 구분됩니다.</p>
keepblanks	<p>유형 CHAR, VARCHAR, LONG VARCHAR 또는 CLOB의 각 필드에 앞뒤 공백을 둡니다. 이 옵션이 없으면, 문자 분리문자 내에 없는 앞 공백과 뒤 공백은 모두 제거되며 공백 필드의 테이블로 NULL이 삽입됩니다.</p> <p>다음 예는 데이터 파일에서 앞과 뒤의 모든 공백을 보존하면서 TABLE1이라는 테이블로 데이터를 로드하는 방법을 설명합니다.</p> <pre>db2 load from delfile3 of del modified by keepblanks insert into table1</pre>
nochardel	<p>로드 유틸리티는 컬럼 분리문자 간의 모든 바이트를 컬럼 데이터의 파트로 가정합니다. 문자 분리문자는 컬럼 데이터의 파트로 구분 분석됩니다. DB2를 사용하여 데이터를 익스포트한 경우(익스포트 시 nochardel을 지정한 경우를 제외하고) 이 옵션을 지정하지 말아야 합니다. 이 옵션은 문자 분리문자가 없는 벤더 데이터 파일을 지원하기 위해 제공됩니다. 부적절한 사용은 데이터 유실이나 손상을 초래할 수 있습니다.</p> <p>이 옵션은 chardelx, delprioritychar 또는 nodoubledel과 함께 지정될 수 없습니다. 이들은 상호 독점 옵션입니다.</p>
nodoubledel	<p>2바이트 분리문자를 인식하지 않습니다.</p>

표 27. 로드 유틸리티의 유효한 파일 유형 수정자: IXF 파일 형식

수정자	설명
forcein	코드 페이지 불일치와 관계없이 데이터를 승인하고 코드 페이지 간에 변환하지 않도록 유틸리티에 지시합니다.  데이터의 고정 길이 대상 필드가 충분히 크지 검증하도록 고정 길이 대상 필드를 점검합니다. nochecklengths가 지정된 경우, 검사가 수행되지 않으며 각 행을 로드하려고 시도합니다.
nochecklengths	nochecklengths가 지정되면, 목표 테이블 컬럼의 크기를 초과하는 컬럼 정의가 소스 데이터에 있는 경우에도 각 행을 로드하려고 시도합니다. 코드 페이지 변환으로 소스 데이터가 축소되는 경우 그러한 행은 정상적으로 로드될 수 있습니다. 예를 들어, 소스에서 4바이트 EUC 데이터는 목표에서 2바이트 DBCS 데이터로 축소되어 반 정도의 스페이스가 필요합니다. 불일치 컬럼 정의에도 불구하고 소스 데이터가 모든 경우에 맞는 경우 특히 이 옵션이 유용합니다.

주:

1. 날짜 출력 문자열을 둘러싼 큰따옴표는 필수입니다. 필드 구분자는 a - z, A - Z 및 0 - 9를 포함할 수 없습니다. 필드 구분자는 DEL 파일 형식의 필드 분리문자나 문자 분리문자와 같지 않아야 합니다. 요소의 시작 및 종료 위치가 명확한 경우 필드 구분자가 선택적입니다. D, H, M 또는 S와 같은 요소가 사용되는 경우(수정자에 따라) 항목의 변수 길이 때문에 모호함이 있을 수 있습니다.

시간소인 형식의 경우, 문자 M을 사용하는 month 및 minute 디스크립터 간의 모호함을 피하려면 주의해야 합니다. month 필드는 다른 날짜 필드와 인접해야 합니다. minute 필드는 다른 시간 필드와 인접해야 합니다. 다음은 앰비규어스 시간소인 형식입니다.

```
"M"(month 또는 minute일 수 있음)
"M:M"(month 및 minute 구분 가능?)
"M:YYYY:M"(둘 다 month로 해석됨.)
"S:M:YYYY"(둘 다 시간 값 및 날짜 값에 인접)
```

앰비규어스 경우, 유틸리티는 오류 메시지를 발행하며, 조작에 실패합니다.

다음은 명확한 시간소인 형식입니다.

```
"M:YYYY" (Month)
"S:M" (Minute)
"M:YYYY:S:M" (Month...Minute)
"M:H:YYYY:M:D" (Minute...Month)
```

큰따옴표 및 백슬래시와 같은 일부 문자는 Escape 문자(예: #)가 앞에 와야 합니다.

2. chardel, coldel 또는 decpt 파일 유형에 제공되는 문자 값은 소스 데이터의 코드 페이지에 지정되어야 합니다.

문자 코드 포인트(문자 기호 대신)는 구문 xJJ 또는 0xJJ를 사용하여 지정될 수 있으며, 여기서 JJ는 코드 포인트의 16진수를 나타냅니다. 예를 들어, 컬럼 분리문자로 # 문자를 지정하려면, 다음 중 하나를 사용하십시오.

```
... modified by coldel# ...
... modified by coldel0x23 ...
... modified by coldelX23 ...
```

3. 데이터 이동을 위한 분리문자 고려사항은 분리문자를 겹쳐쓰기할 때 사용할 수 있는 문자에 적용하는 제한사항을 나열합니다.
4. MODIFIED BY 옵션과 함께 지원되지 않는 파일 유형을 사용하려고 시도하는 경우 로드 유틸리티는 경고를 발행하지 않습니다. 이런 경우, 로드 조작에 실패하며 오류 코드가 리턴됩니다.
5. 내재적으로 숨겨진 행 변경 시간소인 컬럼을 포함하는 테이블로 импорт할 때, 내재적으로 숨겨진 컬럼의 등록 정보는 무시됩니다. 그러므로, 컬럼의 데이터가 импорт되는 데이터에 표시되지 않으며 명시적 컬럼 목록이 표시되지 않으면 rowchangetimestampmissing 파일 유형 수정자가 импорт 명령에 지정되어야 합니다.

표 28. codepage 및 usegraphiccodepage 사용 시 LOAD 동작

codepage=N	usegraphiccodepage	LOAD 동작
Absent	Absent	CLIENT 옵션이 지정된 경우에도 파일의 모든 데이터는 응용프로그램 코드 페이지가 아닌 데이터베이스 코드 페이지에 있는 것으로 가정됩니다.
Present	Absent	파일의 모든 데이터는 코드 페이지 N에 있는 것으로 가정됩니다.  경고: N이 1바이트 코드 페이지이면 데이터베이스로 로드될 때 그래픽 데이터가 손상됩니다.
Absent	Present	CLIENT 옵션이 지정된 경우에도 파일의 문자 데이터는 데이터베이스 코드 페이지에 있는 것으로 가정됩니다. CLIENT 옵션이 지정된 경우에도 그래픽 데이터는 데이터베이스 그래픽 데이터의 코드 페이지에 있는 것으로 가정됩니다.  데이터베이스 코드 페이지가 1바이트이면, 모든 데이터는 데이터베이스 코드 페이지에 있는 것으로 가정됩니다.  경고: 1바이트 데이터베이스로 로드될 때 그래픽 데이터가 손상됩니다.
Present	Present	문자 데이터는 코드 페이지 N에 있는 것으로 가정됩니다. 그래픽 데이터는 N의 그래픽 코드 페이지에 있는 것으로 가정됩니다.  N이 1바이트 또는 2바이트 코드 페이지인 경우, 모든 데이터는 코드 페이지 N에 있는 것으로 가정됩니다.  경고: N이 1바이트 코드 페이지이면 데이터베이스로 로드될 때 그래픽 데이터가 손상됩니다.



## 제 86 장 LOAD QUERY

처리 중에 로드 조작의 상태를 점검하고 테이블 상태를 리턴합니다. 로드가 처리 중이 아닌 경우 테이블 상태만 리턴됩니다. 이 명령을 성공적으로 호출하려면 동일한 데이터 베이스에 대한 연결 및 개별 CLP 세션도 필요합니다. 로컬 또는 리모트 사용자가 사용할 수 있습니다.

### 권한 부여

없음

### 필수 연결

데이터베이스

### 명령 구문

```
▶▶—LOAD QUERY—TABLE—table-name—┬—TO—local-message-file—┬—NOSUMMARY—  
└──────────────────────────────────┘└──────────SUMMARYONLY—┘  
  
└—SHOWDELTA—┘▶▶
```

### 명령 매개변수

#### NOSUMMARY

로드 요약 정보(읽은 행 수, 생략한 행 수, 로드한 행, 거부된 행, 삭제된 행, 커미트된 행 및 경고 수)가 보고되지 않도록 지정합니다.

#### SHOWDELTA

새 정보만(LOAD QUERY 명령의 마지막 호출 이후 발생한 로드 이벤트에 관한) 보고되도록 지정합니다.

#### SUMMARYONLY

로드 요약 정보만 보고되도록 지정합니다.

#### TABLE *table-name*

데이터가 현재 로드되고 있는 테이블의 이름을 지정합니다. 규정에 맞지 않는 테이블 이름이 지정되면 테이블은 CURRENT SCHEMA로 규정됩니다.

#### TO *local-message-file*

로드 조작 중 발생하는 경고 및 오류 메시지의 목적지를 지정합니다. 이 파일은 LOAD 명령에 대해 지정된 *message-file*일 수 없습니다. 파일이 이미 존재하는 경우 로드 유틸리티가 생성한 모든 메시지가 파일에 추가됩니다.

## 예:

많은 양의 데이터를 BILLYBOB 데이터베이스의 STAFF 테이블에 로드하는 사용자가 로드 조작의 상태를 점검하려고 합니다. 사용자는 다음을 지정할 수 있습니다.

```
db2 connect to billybob
db2 load query table staff to /u/mydir/staff.tempmsg
```

출력 파일 /u/mydir/staff.tempmsg는 다음과 비슷할 수 있습니다.

```
SQL3501W 데이터베이스에 대한 포워드 복구가 사용 불가능하므로
backup pending state since forward recovery is disabled for the database.
```

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

```
SQL3500W The utility is beginning the "LOAD" phase at time "03-21-2002
11:31:16.597045".
```

```
SQL3519W Begin Load Consistency Point. 입력 레코드 계수 = "0".
```

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

```
SQL3519W Begin Load Consistency Point. 입력 레코드 계수 = "104416".
```

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

```
SQL3519W Begin Load Consistency Point. 입력 레코드 계수 = "205757".
```

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

```
SQL3519W Begin Load Consistency Point. 입력 레코드 계수 = "307098".
```

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

```
SQL3519W Begin Load Consistency Point. 입력 레코드 계수 = "408439".
```

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

```
SQL3532I The Load utility is currently in the "LOAD" phase.
```

```
Number of rows read           = 453376
Number of rows skipped        = 0
Number of rows loaded         = 453376
Number of rows rejected       = 0
Number of rows deleted        = 0
Number of rows committed     = 408439
Number of warnings            = 0
```

```
Tablestate:
로드 진행 중
```

## 사용 시 참고사항

잠금 외에 로드 유틸리티는 테이블 상태를 사용하여 테이블에 대한 액세스를 제어합니다. LOAD QUERY 명령을 사용하여 테이블 상태를 판별할 수 있습니다. LOAD

QUERY는 현재 로드되지 않고 있는 테이블에서 사용할 수 있습니다. 파티션된 테이블의 경우 보고되는 상태는 대응하는 가시적 데이터 파티션 상태 중에서 가장 제한적입니다. 예를 들어 단일 데이터 파티션은 읽기 액세스 전용 상태이고 다른 모든 데이터 파티션은 일반 상태에 있는 경우 로드 쿼리 조작용 읽기 액세스 전용 상태를 리턴합니다. 로드 조작용 데이터 파티션의 서브세트를 테이블의 나머지와 다른 상태에 두지 않습니다. LOAD QUERY에 의해 설명되는 테이블 상태는 다음과 같습니다.

**일반** 다른(비정상) 테이블 상태 중 하나에 있지 않은 경우 테이블은 일반 상태에 있습니다. 일반 상태는 테이블이 작성된 후 테이블의 초기 상태입니다.

#### 무결성 설정 보류

테이블은 아직 검증되지 않은 제한조건을 갖습니다. 테이블을 무결성 설정 보류 상태에서 벗어나도록 하려면 SET INTEGRITY문을 사용하십시오. 로드 유틸리티는 제한조건을 갖는 테이블에서 로드 조작용 시작할 때 테이블을 무결성 설정 보류 상태에 둡니다.

#### 로드 진행 중

이것은 로드 조작용 중에만 적용되는 임시 상태입니다. 로드 조작용 실패했거나 인터럽트된 로드 진행 중 상태에서 테이블을 벗어나도록 하는 데 대한 정보는 [관련 링크](#) 섹션에서 로드 조작용 후 보류 상태에 대한 섹션을 참조하십시오. 로드 진행 중 테이블 스페이스 상태도 참조하십시오.

#### 로드 보류

로드 조작용이 이 테이블에서 사용 중이었지만 데이터를 커밋할 수 있기 전에 중단되었습니다. LOAD TERMINATE, LOAD RESTART 또는 LOAD REPLACE 명령을 발행하여 테이블을 이 상태에서 벗어나게 하십시오.

#### 읽기 액세스 전용

테이블은 ALLOW READ ACCESS 옵션이 지정된 경우 로드 조작용 중에 이 상태에 있습니다. 읽기 액세스 전용은 다른 응용프로그램 및 유틸리티가 로드 조작용 전에 존재한 데이터에 대해 읽기 액세스할 수 있는 임시 상태입니다.

#### Reorg 보류

REORG 명령은 ALTER TABLE문이 테이블에서 실행되기를 권장합니다. 테이블에 다시 액세스할 수 있으려면 클래식 REORG를 수행해야 합니다.

#### 사용 불가능

테이블이 사용 불가능합니다. 테이블은 백업에서만 삭제 또는 리스토어할 수 있습니다. 복구 불가능한 로드 조작용 통한 롤 포워드는 테이블을 사용 불가능 상태에 둡니다.

#### 로드 재시작 불가능

테이블은 로드 재시작 조작용 허용하지 않는 부분적으로 로드된 상태에 있습니다. 이 테이블은 로드 보류 상태에도 있습니다. LOAD TERMINATE 또는 LOAD REPLACE 명령을 발행하여 테이블을 로드 재시작 불가능 상태에서 벗

어나게 하십시오. 테이블은 성공적으로 재시작 또는 종료되지 않은 실패한 로드 조작 후 롤 포워드 조작이 수행될 때 또는 테이블이 로드 진행 중 또는 로드 보류 상태에 있는 중에 작성된 온라인 백업으로부터 리스토어 조작이 수행될 때 로드 재시작 불가능 상태에 들어갑니다. 어느 경우이나 로드 재시작 조작에 필요한 정보를 얻을 수 없으며 로드 재시작 불가능 상태는 로드 재시작 조작이 발생하지 못하게 합니다.

### Unknown

LOAD QUERY 명령이 테이블 상태를 판별할 수 없습니다.

현재 IBM DB2 데이터베이스 제품이 지원하는 최소 25개 테이블 또는 테이블 스페이스가 있습니다. 이들 상태는 특정 상황에서 데이터에 대한 액세스를 제어하거나 특정 사용자 조치를 유추하거나 필요한 경우 데이터베이스의 무결성을 보호하는 데 사용됩니다. 대부분은 로드 유틸리티 또는 백업 및 리스토어 유틸리티 같은 DB2 유틸리티 중 하나의 조작과 관련된 이벤트의 결과입니다.

중속 테이블 스페이스가 로드 조작 전에 더 이상 Quiesce되지 않는 경우(Quiesce는 지속적 잠금임) 로드 진행 중 테이블 스페이스 상태는 로드 조작 중에 중속 테이블의 백업을 금지합니다. 로드 진행 중 테이블 스페이스 상태는 로드 진행 중 테이블 상태와 다릅니다. 모든 로드 조작은 로드 진행 중 테이블 상태를 사용하지만 COPY NO 옵션이 지정된 로드 조작(복구 가능한 데이터베이스에 대한)도 로드 진행 중 테이블 스페이스 상태를 사용합니다.

다음 표는 지원되는 각 테이블 상태를 설명합니다. 또한 데이터베이스를 관리하는 중에 발생할 수 있는 상태를 해석하고 응답하는 방법을 정확하게 보여주는 작업 예를 제공합니다. 예는 AIX에서 실행된 명령 스크립트에서 제공됩니다. 사용자가 직접 복사, 붙여넣기 및 실행할 수 있습니다. UNIX가 아닌 시스템에서 DB2 데이터베이스 제품을 실행 중인 경우 모든 경로가 시스템에 맞는 형식인지 확인하십시오. 대부분의 예는 DB2 데이터베이스 제품과 함께 제공되는 SAMPLE 데이터베이스의 테이블을 기초로 합니다. 몇 가지 예는 SAMPLE 데이터베이스의 파트가 아닌 시나리오가 필요하지만, SAMPLE 데이터베이스에 대한 연결을 시작점으로 사용할 수 있습니다.



표 29. 지원되는 테이블 상태

상태	예:
로드 보류	<p>상당한 양의 데이터(예를 들어, 20000개 이상의 레코드)를 갖는 로드 입력 파일 staffdata.del이 제공될 때, NEWSTAFF라는 새 테이블인 로드 조작의 대상 테이블이 들어있는 작은 테이블 스페이스를 작성하십시오.</p> <pre>connect to sample; create tablespace ts1 managed by database using (file '/home/melnk/melnk/NODE0000 /SQL00001/ts1c1' 256); create table newstaff like staff in ts1; load from staffdata.del of del insert into newstaff; load query table newstaff; load from staffdata.del of del terminate into newstaff; load query table newstaff; connect reset;</pre> <p>LOAD QUERY 명령이 리턴하는 정보는 NEWSTAFF 테이블이 로드 보류 상태에 있음을 표시합니다. 로드가 작업을 종료한 후 테이블은 일반 상태에 있습니다.</p>
로드 진행 중	<p>상당한 양의 데이터(예를 들어, 20000 이상의 레코드)를 갖는 로드 입력 파일 staffdata.del이 있는 경우,</p> <pre>connect to sample; create table newstaff like staff; load from staffdata.del of del insert into newstaff;</pre> <p>로드 조작이 실행 중인 동안 다른 세션에서 다음 스크립트를 실행하십시오.</p> <pre>connect to sample; load query table newstaff; connect reset;</pre> <p>LOAD QUERY 명령이 리턴하는 정보는 NEWSTAFF 테이블이 로드 진행 중 상태에 있음을 나타냅니다.</p>
일반	<pre>connect to sample; create table newstaff like staff; load query table newstaff;</pre> <p>LOAD QUERY 명령이 리턴하는 정보는 NEWSTAFF 테이블이 일반 상태에 있음을 표시합니다.</p>

표 29. 지원되는 테이블 상태 (계속)

상태	예:
로드 재시작 불가능	<p>상당한 양의 데이터(예를 들어, 20000 이상의 레코드)를 갖는 로드 입력 파일 staffdata.del이 있는 경우,</p> <pre> update db cfg for sample using logretain recovery; backup db sample; connect to sample; create tablespace ts1 managed by database using (file '/home/melnk/melnk/NODE0000 /SQL00001/ts1c1' 256); create table newstaff like staff in ts1; connect reset; backup db sample; </pre> <p>이 백업 이미지의 시간소인은 20040629205935입니다.</p> <pre> connect to sample; load from staffdata.del of del insert into newstaff copy yes to /home/melnk/backups; connect reset; restore db sample taken at 20040629205935; rollforward db sample to end of logs and stop; connect to sample; load query table newstaff; connect reset; </pre> <p>LOAD QUERY 명령이 리턴하는 정보는 NEWSTAFF 테이블이 로드 재시작 불가능 및 로드 보류 상태에 있음을 표시합니다.</p> <pre> connect to sample; load from staffdata.del of del terminate into newstaff copy yes to /home/melnk/backups; load query table newstaff; connect reset; </pre> <p>LOAD QUERY 명령이 리턴하는 정보는 NEWSTAFF 테이블이 이제 일반 상태에 있음을 표시합니다.</p>
읽기 액세스 전용	<p>상당한 양의 데이터(예를 들어, 20000 이상의 레코드)를 갖는 로드 입력 파일 staffdata.del이 있는 경우,</p> <pre> connect to sample; export to st_data.del of del select * from staff; create table newstaff like staff; import from st_data.del of del insert into newstaff; load from staffdata.del of del insert into newstaff allow read access; </pre> <p>로드 조작이 실행 중인 동안 다른 세션에서 다음 스크립트를 실행하십시오.</p> <pre> connect to sample; load query table newstaff; select * from newstaff; connect reset; </pre> <p>LOAD QUERY 명령이 리턴하는 정보는 NEWSTAFF 테이블이 읽기 액세스 전용 및 로드 진행 중 상태에 있음을 표시합니다. 쿼리는 STAFF 테이블의 익스포트된 콘텐츠인 로드 조작 전에 NEWSTAFF 테이블에 존재했던 데이터만 리턴합니다.</p>
무결성 설정 보류	<p>다음 콘텐츠를 갖는 로드 입력 파일 staff_data.del을 가정합니다.</p> <pre> 11,"Melnk",20,"Sales",10,70000,15000: </pre> <pre> connect to sample; alter table staff add constraint max_salary check (100000 - salary &gt; 0); load from staff_data.del of del insert into staff; load query table staff; </pre> <p>LOAD QUERY 명령이 리턴하는 정보는 STAFF 테이블이 무결성 설정 보류 상태에 있음을 표시합니다.</p>

표 29. 지원되는 테이블 상태 (계속)

상태	예:
사용 불가능	<p>다음 콘텐츠를 갖는 로드 입력 파일 staff_data.del을 가정합니다.</p> <pre> 11,"Melnyk",20,"Sales",10,70000,15000: update db cfg for sample using logretain recovery; backup db sample;  이 백업 이미지의 시간소인은 20040629182012입니다.  connect to sample; load from staff_data.del of del insert into staff nonrecoverable; connect reset; restore db sample taken at 20040629182012; rollforward db sample to end of logs and stop; connect to sample; load query table staff; connect reset;                     </pre> <p>LOAD QUERY 명령이 리턴하는 정보는 STAFF 테이블이 사용 불가능 상태에 있음을 표시합니다.</p>

테이블 상태에 대한 추가 정보는 [관련 링크](#) 섹션을 참조하십시오.

LIST UTILITIES 명령을 사용하여 로드 조작의 진행을 모니터링할 수도 있습니다.



---

## 제 87 장 UPGRADE DATABASE

이전 버전의 DB2 데이터베이스를 인스턴스에 의해 실행하는 릴리스에 해당하는 형식으로 변환합니다.

사용자 데이터베이스의 업그레이드 준비가 되었는지 확인하려면 인스턴스를 업그레이드하기 전에 `db2ckupgrade` 명령을 발행해야 합니다. `db2iupgrade` 명령은 내재적으로 `db2ckupgrade`를 호출합니다. 업그레이드 이전, 그리고 DB2 데이터베이스 제품을 Windows 운영 체제에 설치하기 전에 모든 데이터베이스를 백업하십시오.

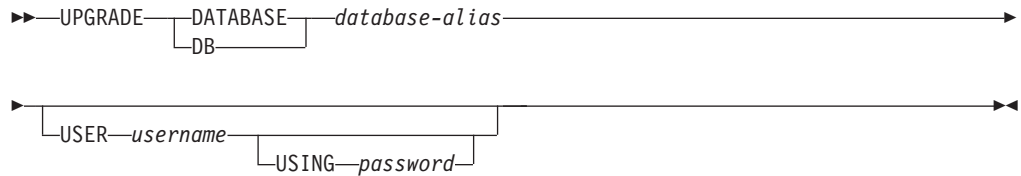
### 권한 부여

*sysadm*

### 필수 연결

이 명령은 데이터베이스 연결을 설정합니다.

### 명령 구문



### 명령 매개변수

#### **DATABASE** *database-alias*

현재 설치된 데이터베이스 관리 프로그램 버전으로 업그레이드할 데이터베이스의 별명을 지정합니다.

#### **USER** *username*

데이터베이스를 업그레이드할 사용자 이름을 식별합니다.

#### **USING** *password*

사용자 이름을 인증하는 데 사용되는 암호. 암호가 생략되었지만 사용자 이름을 지정한 경우, 사용자가 이를 입력하도록 프롬프트가 표시됩니다.

### 예:

다음 예는 데이터베이스 별명 `sales` 아래 카탈로그된 데이터베이스를 업그레이드합니다.

```
db2 UPGRADE DATABASE sales
```

## 사용 시 참고사항

이 명령은 데이터베이스를 새 버전으로 업그레이드만 하며, 업그레이드된 데이터베이스를 이전 버전으로 변환하는 데 사용될 수 없습니다.

업그레이드하기 전에 데이터베이스를 카탈로그해야 합니다.

업그레이드하는 동안 오류가 발생하면, 제시된 사용자 응답을 시도하기 전에 `TERMINATE` 명령을 발행하는 것이 필요합니다. 예를 들어, 업그레이드하는 동안 로그가 가득 참 오류가 발생하면(SQL1704: 데이터베이스 업그레이드 실패. 이유 코드 "3"), 데이터베이스 구성 매개변수 `LOGPRIMARY` 및 `LOGFILSIZ`의 값을 늘리기 전에 `TERMINATE` 명령을 발행하는 것이 필요합니다. 데이터베이스가 이미 재워치된 이후 업그레이드 실패 시("로그가 가득 참" 오류가 리턴되는 경우와 유사), CLP는 해당 데이터베이스 디렉토리 캐시를 새로 고쳐야 합니다.

## 제 88 장 PING

클라이언트와 연결된 데이터베이스 서버 사이의 기본 연결성의 네트워크 응답 시간을 테스트합니다.

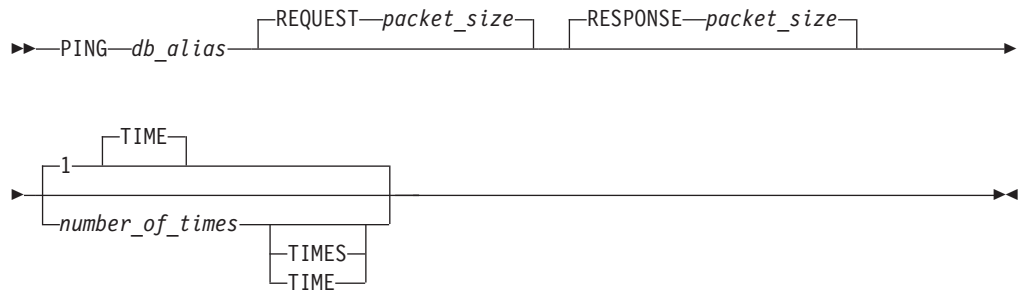
### 권한 부여

없음

### 필수 연결

데이터베이스

### 명령 구문



### 명령 매개변수

#### *db\_alias*

Ping이 전송될 DRDA 서버의 데이터베이스에 대한 데이터베이스 별명을 지정합니다. 이 매개변수는 필수이지만 현재 사용되지 않습니다. 나중에 사용하기 위해 예약됩니다. 모든 유효한 데이터베이스 별명 이름을 지정할 수 있습니다.

#### **REQUEST** *packet\_size*

서버로 전송될 패키지의 크기를 바이트 단위로 지정합니다. 크기는 0 - 32767 범위에 있어야 합니다. 디폴트는 10바이트입니다. 이 옵션은 Linux, UNIX 및 Windows용 DB2 데이터베이스 버전 8 이상 또는 z/OS용 DB2 Universal Database 버전 8을 실행 중인 서버에서만 유효합니다.

#### **RESPONSE** *packet\_size*

클라이언트로 다시 전송될 패키지의 크기를 바이트 단위로 지정합니다. 크기는 0 - 32767 범위에 있어야 합니다. 디폴트는 10바이트입니다. 이 옵션은 Linux, UNIX 및 Windows용 DB2 데이터베이스 버전 8 또는 z/OS용 DB2 UDB 버전 8 이상을 실행 중인 서버에서만 유효합니다.

### *number\_of\_times*

이 테스트에 대한 반복 횟수를 지정합니다. 값은 1 - 32767 범위에 있어야 합니다. 디폴트값은 1입니다. 각 반복에 대해 하나의 시간 제어가 리턴됩니다.

### 예:

#### 예 1

호스트 데이터베이스 hostdb로의 연결에 대한 네트워크 응답 시간을 한 번 테스트하려면 다음을 수행하십시오.

```
db2 ping hostdb 1
또는
db2 ping hostdb
```

명령은 다음과 비슷한 출력을 표시합니다.

경과 시간: 7221마이크로 초

#### 예 2:

호스트 데이터베이스 hostdb로의 연결에 대한 네트워크 응답 시간을 5번 테스트하려면 다음을 수행하십시오.

```
db2 ping hostdb 5
또는
db2 ping hostdb 5 times
```

명령은 다음과 비슷한 출력을 표시합니다.

경과 시간: 8412마이크로 초  
경과 시간: 11876마이크로 초  
경과 시간: 7789마이크로 초  
경과 시간: 10124마이크로 초  
경과 시간: 10988마이크로 초

#### 예 3:

100바이트 요청 패킷 및 200바이트 응답 패킷을 사용하여 호스트 데이터베이스 hostdb로의 연결에 대한 네트워크 응답 시간을 테스트하려면 다음을 수행하십시오.

```
db2 ping hostdb request 100 response 200
또는
db2 ping hostdb request 100 response 200 1 time
```

## 사용 시 참고사항

이 명령을 호출하기 전에 데이터베이스 연결이 존재해야 하며, 그렇지 않으면 오류가 발생합니다.



리턴되는 경과 시간은 IBM Data Server Client와 DB2 서버 사이의 연결에 대한 것입니다.

DB2 호스트 데이터베이스 서버에 연결된 DB2 Connect 버전 8을 통해 DB2 Universal Database 버전 7 클라이언트에서 사용될 때 이 명령은 작동하지 않습니다.



---

## 제 89 장 PRECOMPILE

Embedded SQL문을 포함하는 응용프로그램 소스 파일을 처리합니다. SQL문의 호스트 언어 호출을 포함하는 수정된 소스 파일이 생성되며, 디폴트로 패키지가 데이터베이스에 작성됩니다.

### 범위

이 명령은 db2nodes.cfg에서 모든 데이터베이스 파티션에서 발행될 수 있습니다. 파티션된 데이터베이스 환경에서, 이 명령은 db2nodes.cfg 파일에 정의된 모든 데이터베이스 파티션 서버에서 발행될 수 있습니다. 카탈로그 데이터베이스 파티션에서 데이터베이스 카탈로그를 갱신합니다. 모든 데이터베이스 파티션에 갱신사항이 적용됩니다.

### 권한 부여

다음 권한 부여 중 하나여야 합니다.

- *dbadm* 권한
- SQLERROR CHECK 또는 EXPLAIN ONLY가 지정된 경우, EXPLAIN 또는 SQLADM 권한 중 하나면 충분합니다.
- 패키지가 없는 경우, BINDADD 권한과 다음 권한이 필요합니다.
  - 패키지의 스키마 이름이 없는 경우, 데이터베이스에 대한 IMPLICIT\_SCHEMA 권한
  - 패키지의 스키마 이름이 없는 경우, 스키마에 대한 CREATEIN 특권
- 패키지가 존재하는 경우, 다음 특권 중 하나여야 합니다.
  - 스키마에 대한 ALTERIN 특권
  - 패키지에 대한 BIND 특권

또한, 사용자는 응용프로그램에서 정적 SQL문을 컴파일하는 데 필요한 모든 특권이 필요합니다. 그룹에 부여된 특권은 정적 명령문의 권한 부여 검사에는 사용되지 않습니다.

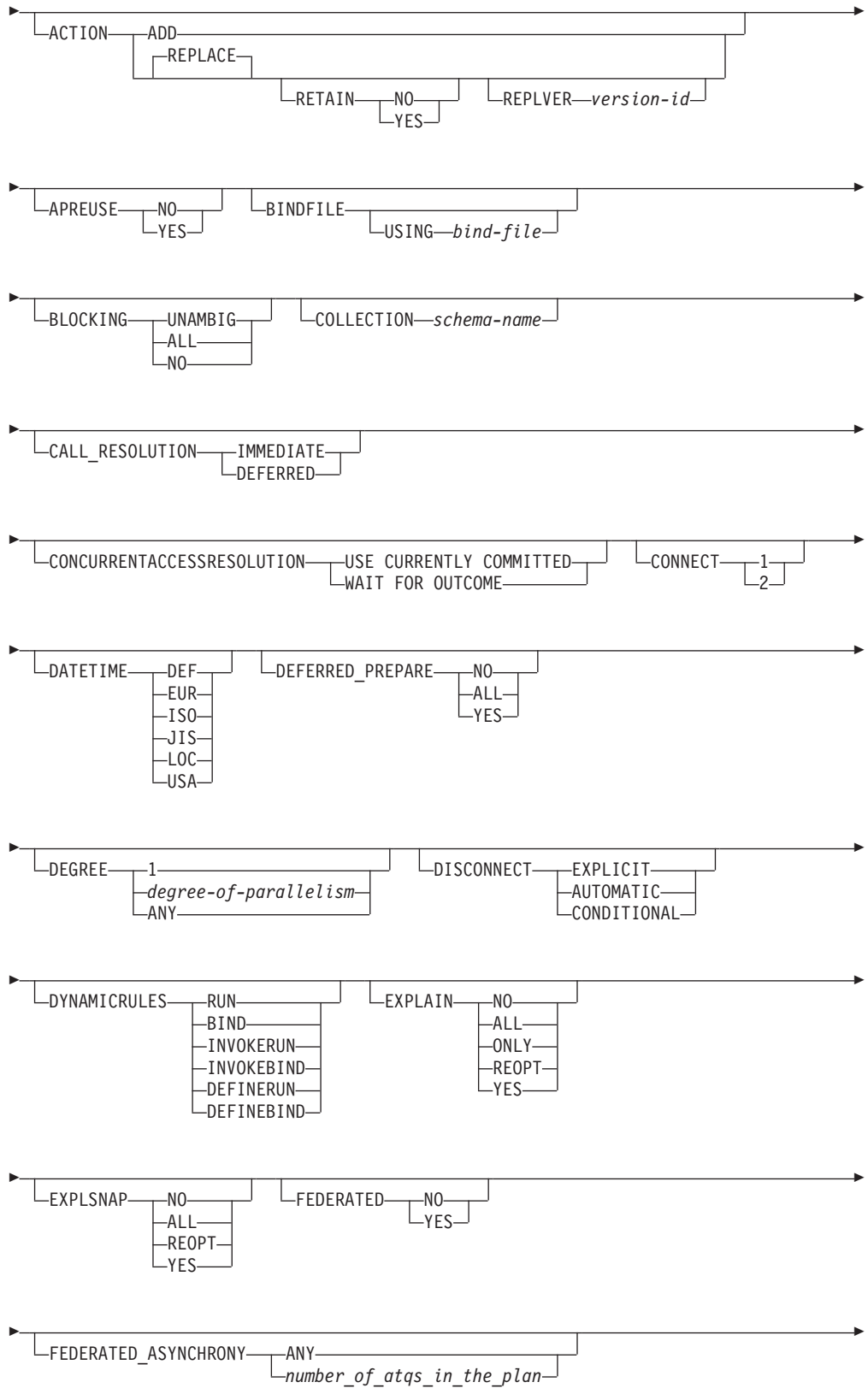
### 필수 연결

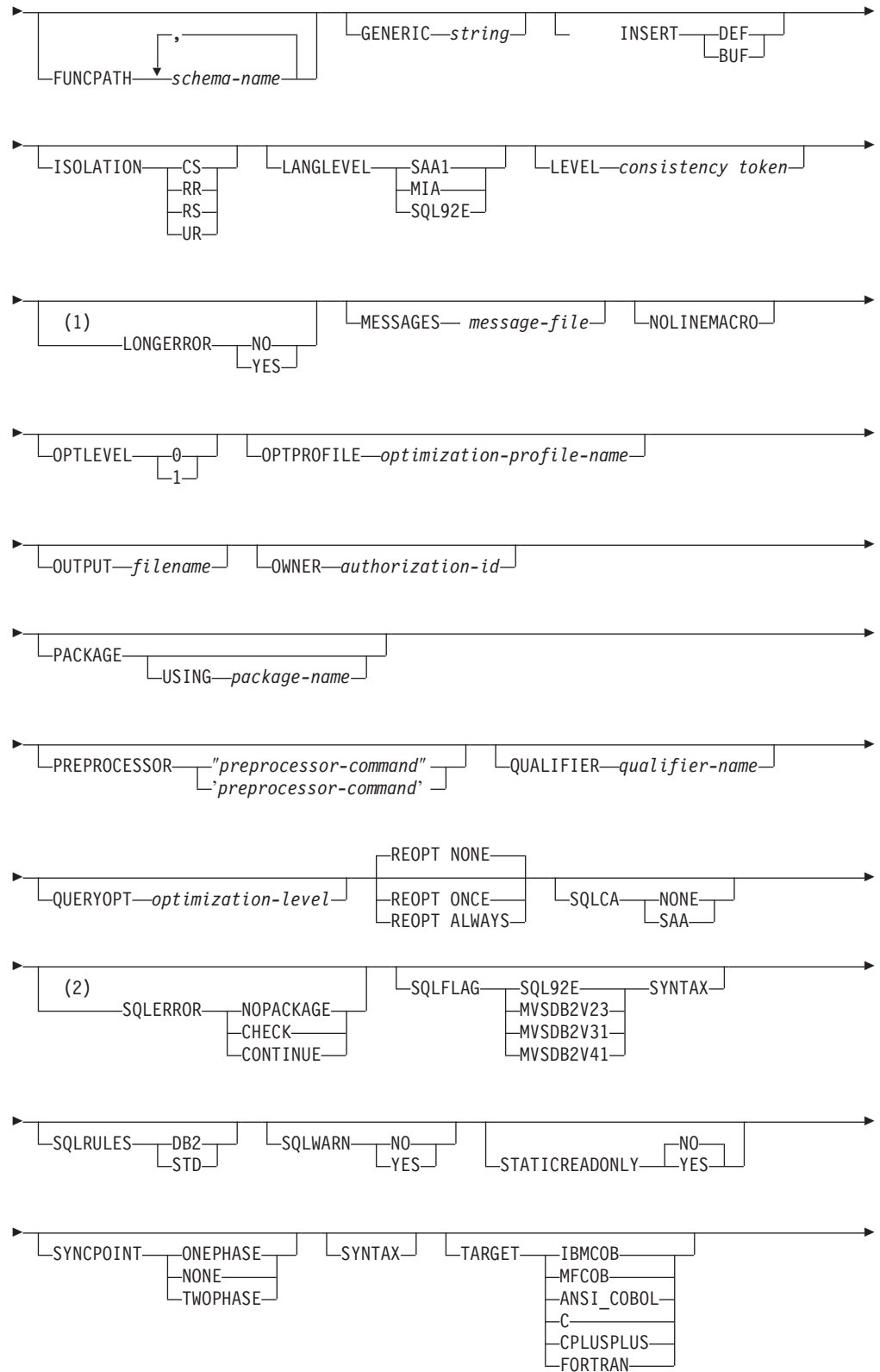
데이터베이스. 내재된 연결이 사용 가능한 경우 디폴트 데이터베이스에 연결됩니다.

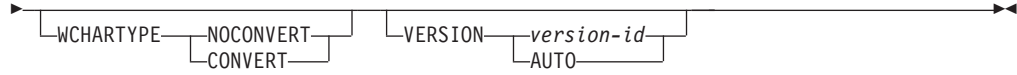
### 명령 구문

**Linux, UNIX 및 Windows용 DB2 데이터베이스의 경우**

▶ `PRECOMPILE filename`  
   └───┬───┘  
      PREP



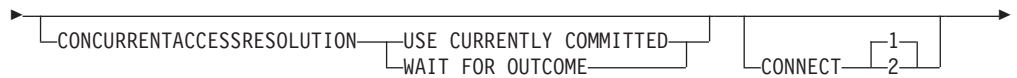
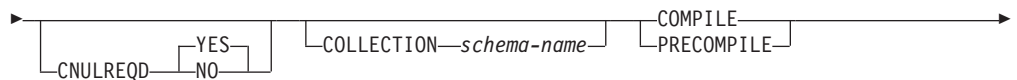
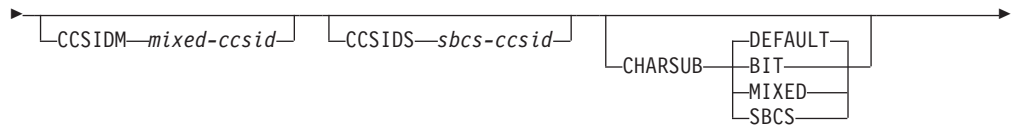
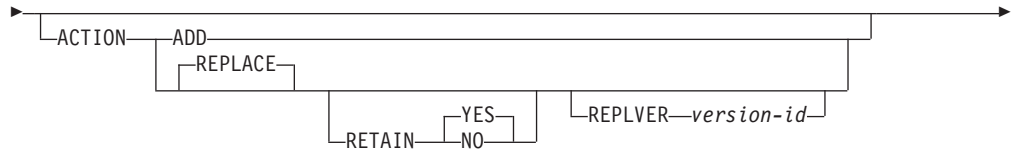
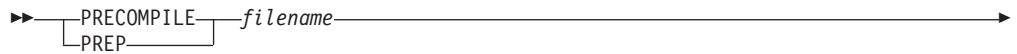


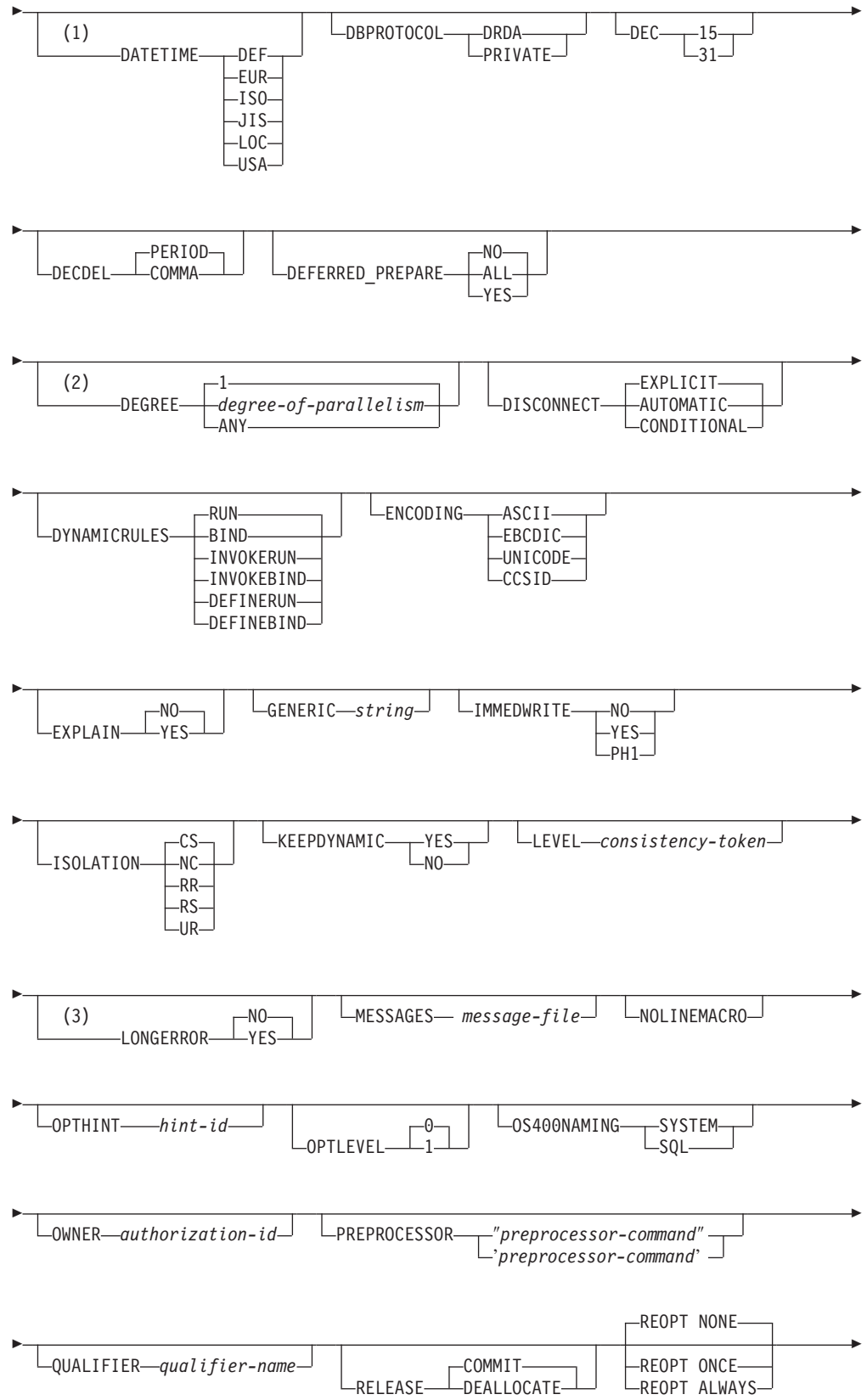


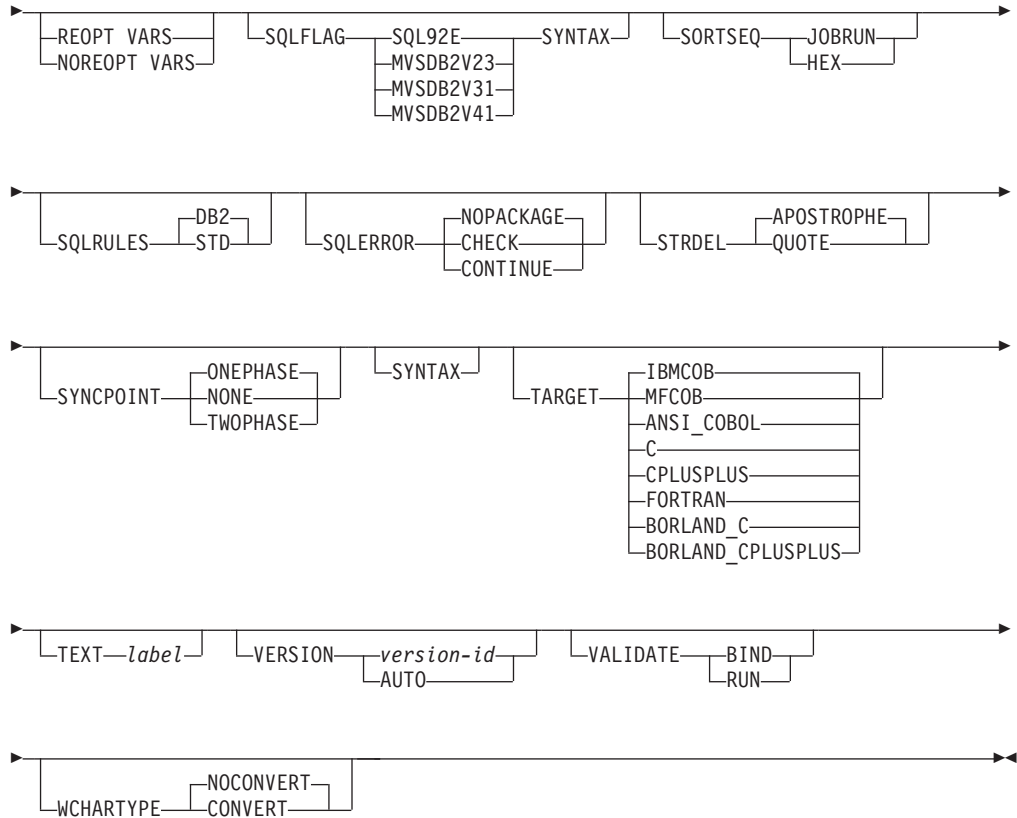
주:

- 1 NO는 32비트 시스템 및 64비트 NT 시스템에서 디폴트이며, 여기서 long 호스트 변수는 INTEGER 컬럼에서 선언하는 데 사용될 수 있습니다. YES는 64비트 UNIX 시스템에서 디폴트입니다.
- 2 SYNTAX는 SQLERROR(CHECK)의 동의어입니다.

**Linux, Windows 및 UNIX 이외의 서버에 있는 DB2 데이터베이스의 경우**







주:

- 1 서버가 DATETIME DEF 옵션을 지원하지 않는 경우, DATETIME ISO로 맵핑됩니다.
- 2 DEGREE 옵션은 DRDA 레벨 2 응용프로그램 서버에서만 지원됩니다.
- 3 NO는 32비트 시스템 및 64비트 NT 시스템에서 디폴트이며, 여기서 long 호스트 변수는 INTEGER 컬럼에서 선언하는 데 사용될 수 있습니다. YES는 64비트 UNIX 시스템에서 디폴트입니다.

## 명령 매개변수

*filename*

프리컴파일될 소스 파일을 지정합니다. 확장자는 다음과 같습니다.

- .sqc는 C 응용프로그램용으로 지정되어야 합니다(.c 파일 생성).
- .sqx(Windows 운영 체제) 또는 .sqC(UNIX 및 Linux 운영 체제)가 C++ 응용프로그램용으로 지정되어야 합니다(Windows 운영 체제에서 .cxx 파일을 또는 UNIX 및 Linux 운영 체제에서 .C 파일을 생성함).
- .sqb가 COBOL 응용프로그램용으로 지정되어야 합니다(.cb1 파일 생성).



- .sqf가 FORTRAN 응용프로그램용으로 지정되어야 합니다(Windows 운영 체제에서 .for 파일을 또는 UNIX 및 Linux 운영 체제에서 .f 파일을 생성함).

UNIX 및 Linux 운영 체제에서 Embedded SQL을 포함하는 C++ 응용프로그램의 선호하는 확장자는 sqC이지만, 대소문자를 구분하지 않는 시스템을 위한 sqx 규칙은 UNIX 및 Linux 운영 체제에서 허용됩니다.

## **ACTION**

패키지가 추가 또는 교체될 수 있는지 여부를 표시합니다.

**ADD** 이름 지정된 패키지가 없으며, 새 패키지가 작성됨을 표시합니다. 패키지가 이미 존재하는 경우, 실행이 중지하며 진단 오류 메시지가 리턴됩니다.

## **REPLACE**

기존 패키지는 동일한 패키지 이름 및 작성자가 있는 새 패키지로 교체됨을 표시합니다. 이것은 ACTION 옵션의 디폴트값입니다.

## **RETAIN**

패키지가 교체될 때 EXECUTE 권한이 보존되는지 여부를 표시합니다. 패키지 변경의 소유권이 변경되면, 새 소유자는 BIND 및 EXECUTE 권한을 이전 패키지 소유자에게 부여합니다.

**NO** 패키지가 교체될 때 EXECUTE 권한을 유지하지 않습니다. 이 값은 DB2에서 지원되지 않습니다.

**YES** 패키지가 교체될 때 EXECUTE 권한을 유지합니다. 이것은 디폴트값입니다.

## **REPLVER *version-id***

패키지의 특정 버전을 교체합니다. 버전 ID는 어떤 패키지 버전이 교체되는지 지정합니다. 지정된 버전이 없으면, 오류가 리턴됩니다. REPLACE의 REPLVER 옵션이 지정되지 않고 프리컴파일 중인 패키지의 이름과 버전이 일치하는 패키지가 이미 존재하는 경우 해당 패키지는 교체되며, 그렇지 않은 경우 새 패키지가 추가됩니다.

## **APREUSE**

정적 SQL 액세스 플랜이 재사용되는지 여부를 지정합니다. 이 옵션이 사용 가능하면, 쿼리 컴파일러는 바인드 동안 그리고 추후 내재된 리바인드와 명시적 리바인드 동안 기존 패키지에서 명령문의 액세스 플랜을 재사용하려고 시도합니다.

**YES** 쿼리 컴파일러가 패키지에서 명령문의 액세스 플랜을 재사용하려고 시도합니다. 기존 패키지가 있는 경우, 쿼리 컴파일러는 새 바인드 파일에서 명령문과 일치하는 모든 명령문의 액세스 플랜을 재사용하도록 시

도합니다. 일치하는 명령문의 경우, 명령문 텍스트가 동일해야 하며 기존 패키지의 명령문에 대한 섹션 번호는 새 패키지의 명령문에 대한 섹션 번호와 일치해야 합니다.

**NO** 쿼리 컴파일러가 패키지에서 명령문의 액세스 플랜을 재사용하려고 시도하지 않습니다. 디폴트 설정입니다.

## **BINDFILE**

바인드 파일이 작성됨. `package` 옵션이 지정되지 않으면 패키지가 작성되지 않습니다. 바인드 파일이 요청되지만, 다음 예에서와 같이 어떤 패키지도 작성되지 않습니다.

```
db2 prep sample.sqc bindfile
```

오브젝트 존재 및 인증 `SQLCODE`는 오류 대신 경고로 취급됩니다. 이는 프리컴파일에 사용되는 데이터베이스에 응용프로그램의 정적 SQL문에서 참조되는 모든 오브젝트가 없는 경우에도, 정상적으로 바인드 파일을 작성할 수 있습니다. 필수 오브젝트가 작성되고 나면, 바인드 파일은 정상적으로 바인드되며 패키지를 작성합니다.

### **USING** *bind-file*

프리컴파일러에서 생성될 바인드 파일의 이름. 파일 이름은 `.bnd`의 확장자를 가져야 합니다. 파일 이름을 입력하지 않은 경우, 프리컴파일러는 프로그램의 이름을 사용하고(*filename* 매개변수로 입력됨), `.bnd` 확장자를 추가합니다. 경로를 제공하지 않은 경우, 현재 디렉토리에 바인드 파일이 작성됩니다.

## **BLOCKING**

커서에 대해 행 블로킹 유형을 지정합니다. LOB 컬럼 데이터 유형에 대한 참조를 포함하는 행 데이터의 블로킹은 Database Partitioning Feature(DPF)가 사용 가능한 환경에서도 지원됩니다.

**ALL** FOR READ ONLY절로 지정된 커서나 FOR UPDATE로 지정되지 않은 커서의 경우, 블로킹이 발생합니다.

앰비규어스 커서는 읽기 전용입니다.

**NO** 임의 커서에 대한 블로킹은 발생하지 않습니다.

읽기 전용 커서 및 앰비규어스 커서의 정의는 `DECLARE CURSOR`문을 참조하십시오.

앰비규어스 커서는 갱신 가능합니다.

### **UNAMBIG**

FOR READ ONLY절로 지정된 커서의 경우, 블로킹이 발생합니다.

앰비규어스가 아니고 읽기 전용인 FOR READ ONLY 또는 FOR UPDATE절로 선언되지 않은 커서는 차단됩니다. 앰비규어스 커서는 차단되지 않습니다.

앰비규어스 커서는 갱신 가능합니다.

### **CALL\_RESOLUTION**

이 옵션이 설정되면, CALL\_RESOLUTION DEFERRED 옵션은 CALL문이 사용되지 않는 sqlproc() API의 호출로서 실행됨을 표시합니다. 이 옵션이 설정되지 않거나 IMMEDIATE가 설정되면, CALL문은 정상 SQL문으로 실행됩니다. CALL\_RESOLUTION IMMEDIATE가 있는 CALL문에 대해서 프로시저를 해결하는 데 프리컴파일러가 실패하면 SQL0204가 발행됩니다.

### **CCSIDG** *double-ccsid*

CREATE 및 ALTER TABLE SQL문의 문자 컬럼 정의에서(특정 CCSID절 없이) 2바이트 문자에 사용되는 코드화된 문자 세트 ID(CCSID)를 지정하는 정수. 이 옵션은 Linux, UNIX 및 Windows용 DB2 데이터베이스에서 지원되지 않습니다. 이 옵션이 지정되지 않은 경우 DRDA 서버는 시스템 정의 디폴트값을 사용합니다.

### **CCSIDM** *mixed-ccsid*

CREATE 및 ALTER TABLE SQL문의 문자 컬럼 정의에서(특정 CCSID절 없이) 혼합 바이트 문자에 사용되는 코드화된 문자 세트 ID(CCSID)를 지정하는 정수. 이 옵션은 Linux, UNIX 및 Windows용 DB2 데이터베이스에서 지원되지 않습니다. 이 옵션이 지정되지 않은 경우 DRDA 서버는 시스템 정의 디폴트값을 사용합니다.

### **CCSIDS** *sbc-ccsid*

CREATE 및 ALTER TABLE SQL문의 문자 컬럼 정의에서(특정 CCSID절 없이) 1바이트 문자에 사용되는 코드화된 문자 세트 ID(CCSID)를 지정하는 정수. 이 옵션은 Linux, UNIX 및 Windows용 DB2 데이터베이스에서 지원되지 않습니다. 이 옵션이 지정되지 않은 경우 DRDA 서버는 시스템 정의 디폴트값을 사용합니다.

### **CHARSUB**

CREATE 및 ALTER TABLE SQL문에서 컬럼 정의에 사용되는 디폴트 문자 부속 유형을 지정합니다. 이 DRDA 프리컴파일/바인드 옵션은 DB2에서 지원되지 않습니다.

**BIT** 명시적 부속 유형이 지정되지 않은 모든 새 문자 컬럼에서 FOR BIT DATA SQL 문자 부속 유형을 사용합니다.

### **DEFAULT**

명시적 부속 유형이 지정되지 않은 모든 새 문자 컬럼에서 목표 시스템 정의된 디폴트를 사용합니다.

## MIXED

명시적 부속 유형이 지정되지 않은 모든 새 문자 컬럼에서 FOR MIXED DATA SQL 문자 부속 유형을 사용합니다.

**SBCS** 명시적 부속 유형이 지정되지 않은 모든 새 문자 컬럼에서 FOR SBCS DATA SQL 문자 부속 유형을 사용합니다.

## CNULREQD

이 옵션은 DRDA에서 지원되지 않는 LANGLEVEL 프리컴파일 옵션과 관련됩니다. 바인드 파일이 C 또는 C++ 응용프로그램에서 작성된 경우에만 유효합니다. 이 DRDA 바인드 옵션은 DB2에서 지원되지 않습니다.

**NO** C 문자열 호스트 변수에서 널(NULL) 종료자에 대해 LANGLEVEL SAA1 프리컴파일 옵션에 기준하여 응용프로그램이 코딩되었습니다.

**YES** C 문자열 호스트 변수에서 널(NULL) 종료자에 대해 LANGLEVEL MIA 프리컴파일 옵션에 기준하여 응용프로그램이 코딩되었습니다.

## COLLECTION *schema-name*

패키지에 대해서 128바이트 컬렉션 ID를 지정합니다. 지정되지 않은 경우, 패키지 처리하는 사용자의 권한 부여 ID가 사용됩니다.

## CONCURRENTACCESSRESOLUTION

패키지에서 명령문에 사용할 동시 액세스 분석을 지정합니다.

## USE CURRENTLY COMMITTED

데이터베이스 관리 프로그램이 갱신 또는 삭제 중인 프로세스에 있을 때 데이터베이스 관리 프로그램이 적용 가능한 스캔을 위해 현재 커밋된 데이터 버전을 사용할 수 있음을 지정합니다. 삽입 중인 프로세스의 행은 건너뛸 수 있습니다. 유효한 분리 레벨이 커서 안정성 또는 읽기 안정성(읽기 안정성의 경우 커밋되지 않은 삽입만 건너뛸)이며 그렇지 않은 경우 무시되는 경우 이 절이 적용됩니다. 적용 가능한 스캔은 읽기 전용이 아닌 명령문과 읽기 전용 명령문의 부분인 읽기 전용 스캔을 포함합니다. 레지스트리 변수 **DB2\_EVALUNCOMMITTED**, **DB2\_SKIPDELETED** 및 **DB2\_SKIPINSERTED**의 설정은 더 이상 적용되지 않습니다.

## WAIT FOR OUTCOME

갱신 중인 프로세스에서 데이터 발생시 커밋 또는 롤백을 대기하기 위해 커서 안정성 및 상위 스캔을 지정합니다. 삽입 또는 삭제 중인 프로세스의 행은 건너뛸 수 없습니다. 레지스트리 변수 **DB2\_EVALUNCOMMITTED**, **DB2\_SKIPDELETED** 및 **DB2\_SKIPINSERTED**의 설정은 더 이상 적용되지 않습니다.

## CONNECT

1 CONNECT문이 유형 1 CONNECT로 처리됨을 지정합니다.

2 CONNECT문이 유형 2 CONNECT로 처리됨을 지정합니다.

### **DATETIME**

사용되는 날짜 및 시간 형식을 지정합니다.

**DEF** 데이터베이스의 지역 코드와 연관된 날짜 및 시간 형식을 사용합니다.

**EUR** 유럽 날짜 및 시간 형식의 IBM 표준을 사용합니다.

**ISO** 국제 표준 조직의 날짜 및 시간 형식을 사용합니다.

**JIS** 일본 산업 표준의 날짜 및 시간 형식을 사용합니다.

**LOC** 데이터베이스의 지역 코드와 연관된 로컬 양식의 날짜 및 시간 형식을 사용합니다.

**USA** 미국 날짜 및 시간 형식의 IBM 표준을 사용합니다.

### **DBPROTOCOL**

세 파트 이름 명령문으로 식별되는 리모트 사이트에 연결시 사용할 프로토콜을 지정합니다. OS/390 전용 DB2에 의해 지원됩니다. 지원되는 옵션 값 목록은 OS/390용 DB2 문서를 참조하십시오.

**DEC** 10진수 산술 연산에서 사용되는 최대 정밀도를 지정합니다. 이 DRDA 프리컴 파일/바인드 옵션은 DB2에서 지원되지 않습니다. 이 옵션이 지정되지 않은 경우 DRDA 서버는 시스템 정의 디폴트값을 사용합니다.

**15** 15자리 정밀도가 10진수 산술 연산에서 사용됩니다.

**31** 31자리 정밀도가 10진수 산술 연산에서 사용됩니다.

### **DECDEL**

마침표(.) 또는 쉼표(,)가 10진수 및 부동 소수점 리터럴에서 소수점 표시기로 사용되는지 여부를 지정합니다. 이 DRDA 프리컴파일/바인드 옵션은 DB2에서 지원되지 않습니다. 이 옵션이 지정되지 않은 경우 DRDA 서버는 시스템 정의 디폴트값을 사용합니다.

### **COMMA**

소수점 표시기로서 쉼표(,)를 사용합니다.

### **PERIOD**

소수점 표시기로서 마침표(.)를 사용합니다.

### **DEFERRED\_PREPARE**

DB2 일반 서버 데이터베이스 또는 DRDA 데이터베이스에 액세스할 때 성능 향상을 제공합니다. 이 옵션은 인터프로세스(inter-process) 또는 네트워크 플로우를 최소화하기 위해 연관된 OPEN, DESCRIBE 또는 EXECUTE문 플로우로 SQL PREPARE문을 조합합니다.

**NO** PREPARE문은 발행된 시간에 실행됩니다.

**YES** 해당 OPEN, DESCRIBE 또는 EXECUTE문이 발행될 때까지 PREPARE문의 실행이 지연됩니다.

PREPARE문이 즉시 리턴되는 SQLDA를 요구하는 INTO절을 사용하는 경우 지연되지 않습니다. 그러나 임의의 매개변수 표시문자를 사용하지 않는 커서에 대해 PREPARE INTO문이 발행되면, PREPARE가 실행될 때 커서를 pre-OPENing함으로써 처리가 최적화됩니다.

**ALL** PREPARE INTO문이 지연되는 경우를 제외하고 YES와 같습니다. PREPARE문이 SQLDA를 리턴하기 위해 INTO절을 사용하는 경우, OPEN, DESCRIBE 또는 EXECUTE문이 발행되어 리턴될 때까지 응용프로그램은 이 SQLDA의 콘텐츠를 참조하지 않아야 합니다.

## DEGREE

SMP 시스템에서 정적 SQL문의 실행을 위한 병렬 처리 수준을 지정합니다. 이 옵션은 CREATE INDEX 병렬 처리에 영향을 미치지 않습니다.

**1** 명령문의 실행은 병렬 처리를 사용하지 않습니다.

### *degree-of-parallelism*

명령문이 실행될 수 있는 병렬 처리 수준의 등급을 지정합니다(2 - 32767의 값).

**ANY** 데이터베이스 관리 프로그램에 의해 판별된 등급을 사용하여 명령문 실행이 병렬 처리를 포함할 수 있는지 지정합니다.

## DISCONNECT

### AUTOMATIC

커미트 시 모든 데이터베이스 연결이 끊어지도록 지정합니다.

### CONDITIONAL

RELEASE로 표시되거나 열린 WITH HOLD 커서가 없는 데이터베이스 연결은 커미트 시 연결이 끊기도록 지정합니다.

### EXPLICIT

RELEASE문으로 릴리스에 대해 명시적으로 표시된 데이터베이스 연결만 커미트 시 연결이 끊기도록 지정합니다.

## DYNAMICRULES

권한 부여 ID 및 규정되지 않은 오브젝트 참조의 내재된 자격에 대해 사용되는 값의 초기 설정에 대해 런타임에 어떤 규칙이 동적 SQL에 적용되는지 정의합니다.

**RUN** 패키지를 실행하는 사용자의 권한 부여 ID가 동적 SQL문의 권한 부여 검사에 사용됨을 지정합니다. 또한, 권한 부여 ID는 동적 SQL문 내 규정되지 않은 오브젝트 참조의 내재된 자격의 디폴트 패키지 규정자로서 권한 부여 ID가 사용됩니다. 이것은 디폴트값입니다.

**BIND** 권한 부여 및 자격에 대해 정적 SQL에 적용하는 모든 규칙이 런타임에 사용됨을 지정합니다. 즉, 패키지 소유자의 권한 부여 ID는 동적 SQL문의 권한 부여 검사에 사용되며 디폴트 패키지 규정자가 동적 SQL문 내 규정되지 않은 오브젝트 참조의 내재된 자격에는 디폴트 패키지 규정자가 사용됩니다.

#### **DEFINERUN**

패키지가 루틴 컨텍스트 내에서 사용되면, 루틴 정의자의 권한 부여 ID는 권한 부여 검사와 루틴 내 동적 SQL문 내에서 규정되지 않은 오브젝트 참조의 내재된 자격에 사용됩니다.

패키지가 독립형 응용프로그램으로 사용되면, 동적 SQL문은 패키지가 DYNAMICRULES RUN으로 바인드된 것처럼 처리됩니다.

#### **DEFINEBIND**

패키지가 루틴 컨텍스트 내에서 사용되면, 루틴 정의자의 권한 부여 ID는 권한 부여 검사와 루틴 내 동적 SQL문 내에서 규정되지 않은 오브젝트 참조의 내재된 자격에 사용됩니다.

패키지가 독립형 응용프로그램으로 사용되면, 동적 SQL문은 패키지가 DYNAMICRULES BIND로 바인드된 것처럼 처리됩니다.

#### **INVOKERUN**

패키지가 루틴 컨텍스트 내에서 사용되면, 루틴이 호출되었을 때 유효한 현재 명령문 권한 부여 ID가 동적 SQL문의 권한 부여 검사와 해당 루틴 내 동적 SQL문 내에서 규정되지 않은 오브젝트 참조의 내재된 자격에 사용됩니다.

패키지가 독립형 응용프로그램으로 사용되면, 동적 SQL문은 패키지가 DYNAMICRULES RUN으로 바인드된 것처럼 처리됩니다.

#### **INVOKEBIND**

패키지가 루틴 컨텍스트 내에서 사용되면, 루틴이 호출되었을 때 유효한 현재 명령문 권한 부여 ID가 동적 SQL문의 권한 부여 검사와 해당 루틴 내 동적 SQL문 내에서 규정되지 않은 오브젝트 참조의 내재된 자격에 사용됩니다.

패키지가 독립형 응용프로그램으로 사용되면, 동적 SQL문은 패키지가 DYNAMICRULES BIND로 바인드된 것처럼 처리됩니다.

동적 SQL문은 바인드 동작을 나타내는 패키지에서 패키지 소유자의 권한 부여 ID를 사용하므로, 패키지의 사용자가 받아서는 안되는 권한을 패키지 바인더에 부여하지 않아야 합니다. 마찬가지로, 동적 명령문은 루틴 정의자의 권한 부여 ID를 사용하므로 정의 동작을 나타내는 루틴을 정의할 때, 루틴의 정의자에게 패키지 사용자가 받아서는 안되는 권한을 부여해서는 안됩니다.

다음과 같은 동적 Prepared SQL문은 DYNAMICRULES RUN: GRANT, REVOKE, ALTER, CREATE, DROP, COMMENT ON, RENAME, SET INTEGRITY 및 SET EVENT MONITOR STATE로 바인드되지 않은 패키지 내에서 사용될 수 없습니다.

## ENCODING

플랜 또는 패키지에서 정적 명령문의 모든 호스트 변수에 대한 인코딩을 지정합니다. OS/390 전용 DB2에 의해 지원됩니다. 지원되는 옵션 값 목록은 OS/390용 DB2 문서를 참조하십시오.

## EXPLAIN

패키지의 각 SQL문에 선택된 액세스 플랜에 대한 Explain 테이블에 정보를 저장합니다. DRDA는 이 옵션의 ALL 값을 지원하지 않습니다.

**NO** Explain 정보는 캡처되지 않습니다.

**YES** Explain 테이블에는 정적 명령문의 경우 prep/바인드 시간에 그리고 증분식 바인드 명령문의 경우 런타임에 선택된 액세스 플랜에 대한 정보가 채워집니다.

패키지가 루틴에 사용되며 패키지가 증분식 바인드 명령문을 포함하는 경우, 루틴은 MODIFIES SQL DATA로 정의되어야 합니다. 그렇지 않으면, 패키지의 증분식 바인드 명령문이 런타임 오류(SQLSTATE 42985)를 유발합니다.

## REOPT

재최적화가 가능한 각 증분식 바인드 SQL문의 Explain 정보는 런타임 시 Explain 테이블에 저장됩니다. 또한, CURRENT EXPLAIN MODE 특수 레지스터가 NO로 설정되어도 런타임에 재최적화가 가능한 동적 SQL문에 대한 Explain 정보가 수집됩니다.

패키지가 루틴에 사용되면 루틴이 MODIFIES SQL DATA로 정의되어야 하며, 그렇지 않으면 패키지의 증분식 바인드 및 동적 명령문이 런타임 오류(SQLSTATE 42985)를 유발합니다.

## ONLY

ONLY 옵션은 특권 없이 Explain문을 실행하게 합니다. Explain 테이블이 채워지지만 지속적 패키지는 작성되지 않습니다. 바인드 프로세스 동안 동일한 이름과 버전을 가진 기존 패키지가 있는 경우, ACTION REPLACE를 지정해도 기존 패키지는 삭제되거나 교체되지 않습니다. Explain 테이블을 채우는 동안 오류가 발생하는 경우, 오류를 리턴한 명령문이나 다음에 오는 명령문에 대한 explain 정보는 추가되지 않습니다.

**ALL** 각 적격 정적 SQL문의 Explain 정보는 prep/바인드 시간에 Explain 테이블에 위치됩니다. 각 적격 증분식 바인드 SQL문의 Explain 정보



는 런타임에 Explain 테이블에 위치됩니다. 또한, CURRENT EXPLAIN MODE 특수 레지스터가 NO로 설정되는 경우에도 런타임에 적격 동적 SQL문에 대한 Explain 정보가 수집됩니다.

패키지가 루틴에 사용되면 루틴이 MODIFIES SQL DATA로 정의되어야 하며, 그렇지 않으면 패키지의 증분식 바인드 및 동적 명령문이 런타임 오류(SQLSTATE 42985)를 유발합니다.

### EXPLSNAP

Explain 스냅샷 정보를 Explain 테이블에 저장합니다. 이 DB2 프리컴파일/바인드 옵션은 DRDA에서 지원되지 않습니다.

**NO** Explain 스냅샷은 캡처되지 않습니다.

**YES** 각 적격 정적 SQL문의 Explain 정보는 정적 명령문의 경우 prep/바인드 시간에 그리고 증분식 바인드 명령문의 경우 런타임에 Explain 테이블에 위치됩니다.

패키지가 루틴에 사용되고 패키지가 증분식 바인드 명령문을 포함하는 경우, 루틴이 MODIFIES SQL DATA로 정의되어야 하며, 그렇지 않으면 패키지의 증분식 바인드 명령문이 런타임 오류(SQLSTATE 42985)를 유발합니다.

### REOPT

재최적화가 가능한 각 증분식 바인드 SQL문의 Explain 스냅샷 정보는 런타임에 Explain 테이블에 위치됩니다. 또한, CURRENT EXPLAIN SNAPSHOT 특수 레지스터가 NO로 설정되는 경우에도 런타임에 재최적화가 가능한 동적 SQL문에 대한 Explain 스냅샷 정보가 수집됩니다.

패키지가 루틴에 사용되면 루틴이 MODIFIES SQL DATA로 정의되어야 하며, 그렇지 않으면 패키지의 증분식 바인드 및 동적 명령문이 런타임 오류(SQLSTATE 42985)를 유발합니다.

**ALL** 각 적격 정적 SQL문의 Explain 스냅샷 정보는 prep/바인드 시간에 Explain 테이블에 위치됩니다. 각 적격 증분식 바인드 SQL문의 Explain 스냅샷 정보는 런타임에 Explain 테이블에 위치됩니다. 또한, CURRENT EXPLAIN SNAPSHOT 특수 레지스터가 NO로 설정되는 경우에도 런타임에 적격 동적 SQL문에 대한 Explain 스냅샷 정보가 수집됩니다.

패키지가 루틴에 사용되면 루틴이 MODIFIES SQL DATA로 정의되어야 하며, 그렇지 않으면 패키지의 증분식 바인드 및 동적 명령문이 런타임 오류(SQLSTATE 42985)를 유발합니다.

### FEDERATED

패키지의 정적 SQL문이 별칭 또는 페더레이티드 뷰를 참조하는지 여부를 지정

합니다. 이 옵션이 지정되지 않고 패키지의 정적 SQL문이 별칭이나 페더레이티드 뷰를 참조하는 경우, 경고가 리턴되며 패키지가 작성됩니다.

이 옵션은 DRDA 서버에서 지원되지 않습니다.

**NO** 별칭이나 페더레이티드 뷰는 패키지의 정적 SQL문에서 참조되지 않습니다. 이 패키지의 준비 또는 바인드 단계 동안 정적 SQL문에서 별칭이나 페더레이티드 뷰를 직면하는 경우, 오류가 리턴되며 패키지는 작성되지 않습니다.

**YES** 별칭이나 페더레이티드 뷰는 패키지의 정적 SQL문에서 참조될 수 있습니다. 패키지의 준비 또는 바인드 동안 정적 SQL문에서 별칭이나 페더레이티드 뷰를 직면하지 않는 경우, 오류나 경고가 리턴되지 않으며 패키지가 작성됩니다.

### **FEDERATED\_ASYNCRONY**

Embedded SQL을 사용하는 프로그램의 액세스 플랜에서 페더레이티드 서버가 지원하는 비동기 테이블 큐(ATQ)의 최대 수를 지정합니다.

**ANY** 옵티마이저는 액세스 플랜의 ATQ 수를 판별합니다. 옵티마이저는 플랜에서 모든 적격 SHIP이나 리모트 푸시다운 운영자에게 ATQ를 지정합니다. DB2\_MAX\_ASYNC\_REQUESTS\_PER\_QUERY 서버 옵션에 지정된 값은 비동기 요청 수를 제한합니다.

*number\_of\_atqs\_in\_the\_plan*

플랜에서의 ATQ 수. 0 - 32767 범위의 수를 지정합니다.

### **FUNCPATH**

사용자 정의 구별 유형에서 사용되는 함수 경로와 정적 SQL의 함수를 지정합니다. 이 옵션이 지정되지 않으면, 디폴트 함수 경로는 "SYSIBM","SYSFUN",USER이며 여기서 USER는 USER 특수 레지스터의 값입니다. 이 DB2 프리컴파일/바인드 옵션은 DRDA에서 지원되지 않습니다.

*schema-name*

응용프로그램 서버(AS)에 존재하는 스키마를 식별하는 SQL ID로, 일반 또는 구분된 유형이 있습니다. 프리컴파일 또는 바인드 시간에는 스키마가 존재하는지 확인되지 않습니다. 동일한 스키마는 함수 경로에 두 번 이상 나타날 수 없습니다. 함수 경로에 스키마 이름 SYSPUBLIC을 지정할 수 없습니다. 지정될 수 있는 스키마 수는 결과 함수 경로의 길이로 제한되며, 2048바이트를 초과할 수 없습니다. 스키마 SYSIBM은 명시적으로 지정될 필요가 없으며, 함수 경로에 포함되지 않는 경우 내재적으로 최초 스키마로 간주됩니다.

## INSERT

DB2 Enterprise Server Edition 서버에 대해 프리컴파일되거나 바인드되는 프로그램이 성능을 향상시키기 위해 해당 데이터 삽입을 버퍼 지정하도록 요청하게 합니다.

**BUF** 응용프로그램에서의 삽입이 버퍼 지정되도록 지정합니다.

**DEF** 응용프로그램에서의 삽입이 버퍼 지정되지 않도록 지정합니다.

## GENERIC *string*

목표 데이터베이스에 정의된 새 옵션의 바인딩을 지원하지만, DRDA에 의해서 는 지원되지 않습니다. BIND 또는 PRECOMPILED에 정의된 바인드 옵션을 패스하려면 이 옵션을 사용하지 마십시오. 이 옵션은 동적 SQL 성능을 크게 개선시킬 수 있습니다. 구문은 다음과 같습니다.

```
generic "option1 value1 option2 value2 ..."
```

각 옵션 및 값은 하나 이상의 공백으로 구분되어야 합니다. 예를 들어, 목표 DRDA 데이터베이스가 DB2 Universal Database, 버전 8이면,

```
generic "explsnap all queryopt 3 federated yes"
```

을 사용하여 각 EXPLSNAP, QUERYOPT 및 FEDERATED 옵션을 바인드 할 수 있습니다.

문자열의 최대 길이는 32768바이트입니다.

## IMMEDWRITE

그룹 버퍼 풀(GBP) 종속 페이지세트나 데이터베이스 파티션에 작성되는 갱신 에 즉시 쓰기가 수행되는지 여부를 표시합니다. OS/390 전용 DB2에 의해 지원됩니다. 지원되는 옵션 값 목록은 OS/390용 DB2 문서를 참조하십시오.

## ISOLATION

이 패키지에 바인드되는 프로그램이 다른 실행 프로그램의 효과로부터 얼마나 분리되는지 정도를 판별합니다.

**CS** 분리 레벨로 커서 안정성을 지정합니다.

**NC** 커미트가 없습니다. 커미트 제어가 사용되지 않음을 지정합니다. 이 분리 레벨은 DB2에서 지원되지 않습니다.

**RR** 반복 읽기를 분리 레벨로 지정합니다.

**RS** 읽기 안정성을 분리 레벨로 지정합니다. 패키지의 SQL문 실행이 행 읽기를 위한 다른 응용프로그램 프로세스에서 분리되며 응용프로그램에 의해 변경되도록 읽기 안정성이 보장합니다.

**UR** 커미트되지 않은 읽기를 분리 레벨로 지정합니다.

## LANGLEVEL

응용프로그램에서 정적 및 동적 SQL 둘 다의 시맨틱 및 구문 모두에 적용하는 SQL 규칙을 지정합니다. 이 옵션은 DRDA 서버에서 지원되지 않습니다.

**MIA** 다음과 같은 ISO/ANS SQL92 규칙을 선택합니다.

- 오류 SQLCODE 또는 SQLSTATE 검사를 지원하려면, SQLCA가 응용프로그램 코드에 선언되어야 합니다.
- C 널(null)로 끝나는 문자열은 공백으로 채워지며 절단이 발생하는 경우에도 항상 널(null)로 끝나는 문자를 포함합니다.
- FOR UPDATE 절은 위치된 UPDATE에서 갱신되는 모든 컬럼에 선택적입니다.
- 오브젝트 테이블의 컬럼이 검색 조건이나 지정 절의 오른쪽에서 참조되는 경우 검색된 UPDATE 또는 DELETE는 UPDATE 또는 DELETE문의 오브젝트 테이블에서 SELECT 특권을 요구합니다.
- 인덱스를 사용하여 해결될 수 있는 컬럼 함수(예: MIN 또는 MAX)는 NULL 값을 점검하며 널(NULL)이 있는 경우 경고 SQLSTATE 01003을 리턴합니다.
- 중복 고유 제한조건이 CREATE 또는 ALTER TABLE문에 포함되면 오류가 리턴됩니다.
- 특권이 부여되지 않거나 권한 부여자에게 오브젝트에 대한 특권이 없는 경우 오류가 리턴됩니다(그렇지 않으면 경고가 리턴됨).

**SAA1** 다음과 같이 일반 IBM DB2 규칙을 선택합니다.

- 오류 SQLCODE 또는 SQLSTATE 검사를 지원하려면, SQLCA가 응용프로그램 코드에 선언되어야 합니다.
- C 널(null)로 끝나는 문자열은 절단이 일어나는 경우 널(NULL) 문자로 종료하지 않습니다.
- FOR UPDATE 절은 위치된 UPDATE에서 갱신되는 모든 컬럼에 필요합니다.
- 명령문의 fullselect에서 오브젝트 테이블을 참조하는 경우를 제외하고 검색된 UPDATE 또는 DELETE는 UPDATE 또는 DELETE문의 오브젝트 테이블에서 SELECT 특권을 요구하지 않습니다.
- 인덱스를 사용하여 해결될 수 있는 컬럼 함수(예: MIN 또는 MAX)는 NULL 값을 점검하지 않으며 경고 SQLSTATE 01003이 리턴되지 않습니다.
- 경고가 리턴되며 중복 고유 제한조건은 무시됩니다.
- 특권이 부여되지 않으면 오류가 리턴됩니다.

## SQL92E

다음과 같이 ISO/ANS SQL92 규칙을 정의합니다.

- SQLCODE 또는 SQLSTATE 값의 검사를 지원하기 위해, 이 이름에 의한 변수는 호스트 변수 선언 섹션에 선언될 수 있습니다(어느 것도 선언되지 않으면, 프리컴파일 동안 SQLCODE가 가정됨).
- C 널(null)로 끝나는 문자열은 공백으로 채워지며 절단이 발생하는 경우에도 항상 널(null)로 끝나는 문자를 포함합니다.
- FOR UPDATE 절은 위치된 UPDATE에서 갱신되는 모든 컬럼에 선택적입니다.
- 오브젝트 테이블의 컬럼이 검색 조건이나 지정 절의 오른쪽에서 참조되는 경우 검색된 UPDATE 또는 DELETE는 UPDATE 또는 DELETE문의 오브젝트 테이블에서 SELECT 특권을 요구합니다.
- 인덱스를 사용하여 해결될 수 있는 컬럼 함수(예: MIN 또는 MAX)는 NULL 값을 점검하며 널(NULL)이 있는 경우 경고 SQLSTATE 01003을 리턴합니다.
- 중복 고유 제한조건이 CREATE 또는 ALTER TABLE문에 포함되면 오류가 리턴됩니다.
- 특권이 부여되지 않거나 권한 부여자에게 오브젝트에 대한 특권이 없는 경우 오류가 리턴됩니다(그렇지 않으면 경고가 리턴됨).

## KEEPDYNAMIC

동적 SQL문이 커밋 지점 이후에 보존되는지 여부를 지정합니다. OS/390 전용 DB2에 의해 지원됩니다. 지원되는 옵션 값 목록은 OS/390용 DB2 문서를 참조하십시오.

## LEVEL *consistency-token*

일관성 토큰을 사용하여 모듈 레벨을 정의합니다. 일관성 토큰은 길이가 최대 8문자인 영숫자 값입니다. RDB 패키지 일관성 토큰은 리퀘스터의 응용프로그램과 관계형 데이터베이스 패키지가 동기화되었는지 검증합니다. 이 옵션은 일반 용도로는 권장되지 않습니다.

## LONGERROR

long 호스트 변수 선언이 오류로 취급되는지 여부를 표시합니다. 이식성을 위해, sqlint32는 프리컴파일된 C 및 C++ 코드에서 INTEGER 컬럼의 선언으로 사용될 수 있습니다.

**NO** long 호스트 변수 선언의 사용을 위한 오류는 생성하지 않습니다. 이것은 32비트 시스템 및 64비트 NT 시스템의 디폴트이며, 여기서 long 호스트 변수는 INTEGER 컬럼의 선언으로서 사용될 수 있습니다. 64비트 UNIX 플랫폼에서 이 옵션을 사용하면 long 호스트 변수가 BIGINT 컬럼의 선언으로 사용되게 합니다.

**YES** long 호스트 변수 선언 사용 시 오류를 생성합니다. 이것은 64비트 UNIX 시 스템의 디폴트입니다.

#### **MESSAGES** *message-file*

경고, 오류 및 완료 상태 메시지의 대상을 지정합니다. 바인드가 정상적인지 여부의 메시지 파일이 작성됩니다. 메시지 파일 이름이 지정되지 않으면 메시지가 표준 출력에 작성됩니다. 파일에 대한 전체 경로가 지정되지 않으면, 전체 경로가 사용됩니다. 기존 파일의 이름이 지정되지 않으면, 파일의 콘텐츠가 겹쳐쓰입니다.

#### **NOLINEMACRO**

출력 .c 파일에서 #line 매크로의 생성을 제외시킵니다. 프로파일, 상호 참조 유틸리티 및 디버거와 같은 소스 라인 정보를 요구하는 개발 도구를 이용해서 파일이 사용되면 유용합니다. 이 프리컴파일 옵션은 C/C++ 프로그래밍 언어 전용으로 사용됩니다.

#### **OPTHINT**

쿼리 최적화 힌트가 정적 SQL에 사용되는지 여부를 제어합니다. OS/390 전용 DB2에 의해 지원됩니다. 지원되는 옵션 값 목록은 OS/390용 DB2 문서를 참조하십시오.

#### **OPTLEVEL**

호스트 변수가 SQL문에서 사용될 때 C/C++ 프리컴파일러가 내부 SQLDA의 초기화를 최적화할지 여부를 표시합니다. 단일 SQL문(예: FETCH)이 tight 루프 내에서 사용되면 그러한 최적화는 성능을 향상시킬 수 있습니다.

- 0** SQLDA 초기화를 최적화하지 않도록 프리컴파일러에 지시합니다.
- 1** SQLDA 초기화를 최적화하도록 프리컴파일러에 지시합니다. 응용프로그램이 사용하는 경우 이 값을 지정하지 말아야 합니다.

- 다음 예에서와 같은 포인터 호스트 변수:

```
exec sql begin declare section;  
char (*name)[20];  
short *id;  
exec sql end declare section;
```

- SQL문에서의 직접 C++ 데이터 구성원.

#### **OPTPROFILE** *optimization-profile-name*

패키지에서 모든 정적문에 사용되는 기존 최적화 프로파일의 이름을 지정합니다. 옵션의 디폴트값은 빈 문자열입니다. CURRENT OPTIMIZATION PROFILE 특수 레지스터가 널(NULL)인 DML문의 동적 준비를 위한 디폴트값으로 해당 값을 적용합니다. 지정된 이름이 규정되지 않은 경우, QUALIFIER 바인드 옵션에 의해 내재적으로 자격이 부여된 SQL ID입니다.

BIND 명령은 최적화 파일을 처리하지 않지만, 이름이 구문적으로 유효한지만 확인합니다. 그러므로 최적화 프로파일 없거나 유효하지 않은 경우, DML문

이 해당 최적화 프로파일을 사용하여 최적화될 때까지 이유 코드가 13인 SQL0437W 경고는 발생하지 않습니다.

#### **OUTPUT** *filename*

컴파일러에 의해 생성된 수정된 소스 파일의 디폴트 이름을 겹쳐씁니다. 경로를 포함할 수 있습니다.

#### **OS400NAMING**

DB2 for System i 데이터 액세스 시 이름 지정 옵션이 사용됨을 지정합니다. DB2 for System i에 의해서만 지원됩니다. 지원되는 옵션 값 목록은 System i용 DB2의 문서를 참조하십시오.

구분자로 사용되는 슬래시 때문에, 유틸리티가 OS400NAMING SYSTEM 옵션으로 바인드되거나 프리컴파일된 경우에도 DB2 유틸리티는 System i 시스템 이름 지정 규칙을 사용하는 특정 SQL문에 대해 실행 시 구문 오류를 보고할 수 있습니다. 예를 들어, System i 시스템 이름 지정 규칙이 사용된 경우, OS400NAMING SYSTEM 옵션을 사용하여 프리컴파일 또는 바인드되었던지에 관계없이 명령행 처리기는 SQL CALL문에 대해 구문 오류를 보고합니다.

#### **OWNER** *authorization-id*

패키지 소유자의 128바이트 권한 부여 ID를 지정합니다. 소유자는 패키지에 포함된 SQL문을 실행하는 데 필요한 특권을 가져야 합니다. DBADM 권한을 가진 사용자만이 사용자 ID 이외의 권한 부여 ID를 지정할 수 있습니다. 디폴트 값은 프리컴파일/바인드 프로세스의 기본 권한 부여 ID입니다. SYSIBM, SYSCAT 및 SYSSTAT는 이 옵션에 대해 유효한 값이 아닙니다. *authorization-id*는 오직 사용자일 수 있습니다(역할 또는 그룹일 수 없음).

#### **PACKAGE**

패키지를 작성합니다. PACKAGE, BINDFILE 또는 SYNTAX 중 어느 것도 지정되지 않으면, 패키지가 디폴트로 데이터베이스에 작성됩니다.

#### **USING** *package-name*

프리컴파일러에 의해 생성되는 패키지의 이름. 이름이 입력되지 않으면, 응용프로그램 소스 파일의 이름(확장자 빼고 대문자로)이 사용됩니다. 최대 길이는 128바이트입니다.

#### **PREPROCESSOR** "*preprocessor-command*"

Embedded SQL문을 처리하기 전에 프리컴파일러에 의해 실행될 수 있는 선행 처리기 명령을 지정합니다. 선행 처리기 명령 문자열(최대 길이 1024바이트)은 큰따옴표나 작은따옴표로 묶여야 합니다.

이 옵션은 선언 절 내에서 매크로의 사용을 가능하게 합니다. 유효한 선행 처리기 명령은 소스 파일을 지정하지 않고 선행 처리기를 호출하기 위해 명령행에서 발행될 수 있는 명령입니다. 예를 들면, 다음과 같습니다.

```
x1c -P -DMYMACRO=0
```

### **QUALIFIER** *qualifier-name*

패키지에 포함된 규정되지 않은 오브젝트의 128바이트 내재된 규정자를 제공합니다. 소유자가 명시적으로 지정되었는지 여부와 관계없이, 디폴트는 소유자의 권한 부여 ID입니다.

### **QUERYOPT** *optimization-level*

패키지에 포함된 모든 정적 SQL문의 원하는 최적화 레벨을 표시합니다. 디폴트값은 5입니다. SET CURRENT QUERY OPTIMIZATION문은 사용 가능한 최적화 레벨의 전체 범위를 기술합니다. 이 DB2 프리컴파일/바인드 옵션은 DRDA에서 지원되지 않습니다.

### **RELEASE**

자원이 각 커밋 지점에서 릴리스되었는지 여부와 응용프로그램이 종료하는 시기를 표시합니다. 이 DRDA 프리컴파일/바인드 옵션은 DB2에서 지원되지 않습니다.

### **COMMIT**

각 커밋 지점에서 자원을 릴리스합니다. 동적 SQL문에 사용됩니다.

### **DEALLOCATE**

응용프로그램 종료 시에만 자원을 릴리스합니다.

### **REOPT**

호스트 변수, 매개변수 표시문자, 전역 변수 및 특수 레지스터의 값을 사용하여 DB2가 액세스 경로를 최적화할지 여부를 지정합니다. 가능한 값은 다음과 같습니다.

#### **NONE**

호스트 변수, 매개변수 표시문자, 전역 변수 또는 특수 레지스터를 포함하는 제공된 SQL문의 액세스 경로는 이들 변수의 실제 값을 사용하여 최적화되지 않습니다. 이들 변수의 디폴트 추정치가 대신 사용되며, 이 플랜은 캐시되어 추후에 사용됩니다. 이 설정이 디폴트 동작입니다.

#### **ONCE**

제공된 SQL문의 액세스 경로는 쿼리가 처음 실행될 때 호스트 변수, 매개변수 표시문자, 전역 변수 또는 특수 레지스터의 실제 값을 사용하여 최적화됩니다. 이 플랜은 캐시되어 추후에 사용됩니다.

#### **ALWAYS**

제공된 SQL문의 액세스 경로는 항상 각 실행 시간에 알려진 호스트 변수, 매개변수 표시문자, 전역 변수 또는 특수 레지스터의 값을 사용하여 컴파일되고 재최적화됩니다.

### **REOPT | NOREOPT VARS**

이 옵션은 REOPT ALWAYS 및 REOPT NONE에 의해 교체되었지만, 이전 릴리스와의 호환성을 위해 여전히 지원됩니다. 호스트 변수, 매개변수 표시



문자, 전역 변수 및 특수 레지스터의 값을 사용하여 런타임에 DB2가 액세스 경로를 판별할지 여부를 지정합니다. OS/390 전용 DB2에 의해 지원됩니다. 지원되는 옵션 값 목록은 OS/390용 DB2 문서를 참조하십시오.

### **SQLCA**

FORTRAN 응용프로그램 전용입니다. 이 옵션은 다른 언어와 함께 사용되면 무시됩니다.

### **NONE**

수정된 소스 코드가 SAA 정의와 일관되지 않음을 지정합니다.

**SAA** 수정된 소스 코드가 SAA 정의와 일관됨을 지정합니다.

### **SQLERROR**

오류 발생 시 패키지나 바인드 파일을 작성할지 여부를 표시합니다.

### **CHECK**

목표 시스템이 바인드 중인 SQL문의 모든 구문과 시맨틱 점검을 수행하도록 지정합니다. 패키지는 이 프로세스의 파트로 작성되지 않습니다. 바인드 중, 동일한 이름과 버전을 가진 기존 패키지가 있으면, ACTION REPLACE가 지정된 경우에도 기존 패키지는 삭제되거나 교체되지 않습니다.

### **CONTINUE**

SQL문 바인드 시 오류가 발생하는 경우에도 패키지를 작성합니다. VALIDATE RUN이 지정된 경우 권한 부여 또는 기존 이유로 바인드에 실패한 명령문은 실행 시간에 증분식으로 바인드될 수 있습니다. 런타임에 실행하면 오류(SQLCODE -525, SQLSTATE 51015)를 생성합니다.

### **NOPACKAGE**

오류 발생 시 패키지나 바인드 파일이 작성되지 않습니다.

### **SQLFLAG**

이 옵션으로 지정된 SQL 언어 구문에서 편차를 식별하고 보고합니다.

SQLFLAG 이외에 BINDFILE 또는 PACKAGE 옵션이 지정된 경우에만 바인드 파일이나 패키지가 작성됩니다.

다음 옵션 중 하나가 지정된 경우에만 로컬 구문 검사가 수행됩니다.

- BINDFILE
- PACKAGE
- SQLERROR CHECK
- SYNTAX

SQLFLAG가 지정되지 않으면, 플래거 함수가 호출되지 않으며, 바인드 파일이나 패키지가 영향을 받지 않습니다.

## SQL92E SYNTAX

SQL문은 ANSI 또는 ISO SQL92 항목 레벨 SQL 언어 형식과 구문에 대해 점검되며 데이터베이스 카탈로그에 대한 액세스를 요구하는 구문 규칙의 예외가 있습니다. 프리컴파일러 목록에 편차가 보고됩니다.

## MVSDB2V23 SYNTAX

MVS DB2 버전 2.3 SQL 언어 구문에 대해 SQL문이 점검됩니다. 프리컴파일러 목록에 구문의 편차가 보고됩니다.

## MVSDB2V31 SYNTAX

MVS DB2 버전 3.1 SQL 언어 구문에 대해 SQL문이 점검됩니다. 프리컴파일러 목록에 구문의 편차가 보고됩니다.

## MVSDB2V41 SYNTAX

MVS DB2 버전 4.1 SQL 언어 구문에 대해 SQL문이 점검됩니다. 프리컴파일러 목록에 구문의 편차가 보고됩니다.

## SORTSEQ

System i 시스템에서 사용할 정렬 시퀀스 테이블을 지정합니다. DB2 for System i에 의해서만 지원됩니다. 지원되는 옵션 값 목록은 System i용 DB2의 문서를 참조하십시오.

## SQLRULES

다음을 지정합니다.

- DB2 규칙 또는 ISO/ANS SQL92에 근거한 표준(STD) 규칙에 따라 유형 2 CONNECT가 처리되는지 여부.
- 응용프로그램이 LOB 컬럼의 형식을 결과 세트에 지정하는 방식.

## DB2

- SQL CONNECT문이 현재 연결을 설정된(유효) 다른 연결로 스위치하게 합니다.
- 이 디폴트 설정은 최초 페치 요청 동안에만 LOB 값이나 LOB 로케이터가 검색되는지 여부를 응용프로그램이 지정하게 합니다. 후속 페치 요청에서는 동일한 LOB 컬럼 형식을 사용해야 합니다.

## STD

- SQL CONNECT문이 새 연결만을 설정하게 합니다. 유효 연결로 전환하려면 SQL SET CONNECTION문이 사용되어야 합니다.
- 각 페치 요청을 사용하여 LOB 값과 LOB 로케이터 검색 간을 응용프로그램이 변경할 수 있습니다. BLOCKING 바인드 옵션 설정과 관계없이, 하나 이상의 LOB 컬럼이 있는 커서는 차단될 수 없음을 의미합니다.

## SQLWARN

동적 SQL문(PREPARE 또는 EXECUTE IMMEDIATE를 통해)의 컴파일, 또는 설명 처리(PREPARE...INTO 또는 DESCRIBE)에서 경고가 리턴되는지 여부를 표시합니다.

**NO** 경고는 SQL 컴파일러에서 리턴되지 않습니다.

**YES** 경고는 SQL 컴파일러에서 리턴됩니다.

SQLCODE +238은 예외입니다. **sqlwarn** 옵션 값과 무관하게 리턴됩니다.

## STATICREADONLY

정적 커서가 READ ONLY로 처리되는지 여부를 판별합니다. 이 DB2 프리컴파일/바인드 옵션은 DRDA에서 지원되지 않습니다.

**NO** 모든 정적 커서는 제공된 명령문 텍스트와 LANGLEVEL 프리컴파일 옵션 설정이 정상적으로 생성될 때 속성에서 수행됩니다. 이것은 디폴트값입니다.

**YES** FOR UPDATE 또는 FOR READ ONLY절을 포함하고 있지 않은 모든 정적 커서는 READ ONLY로 간주됩니다.

## STRDEL

아포스트로피(') 또는 큰따옴표(")가 SQL문 내에서 문자열 분리문자로 사용될지 여부를 지정합니다. 이 DRDA 프리컴파일/바인드 옵션은 DB2에서 지원되지 않습니다. 이 옵션이 지정되지 않은 경우 DRDA 서버는 시스템 정의 디폴트값을 사용합니다.

## APOSTROPHE

문자열 분리문자로 아포스트로피(')를 사용합니다.

## QUOTE

문자열 분리문자로 큰따옴표(")를 사용합니다.

## SYNCPOINT

커밋 또는 롤백이 여러 데이터베이스 연결 사이에 조정되는 방식을 지정합니다. 이 명령 매개변수는 무시되며 이전 버전과의 호환성을 위해서만 여기에 포함됩니다.

## NONE

트랜잭션 관리 프로그램(TM)이 2단계 커밋을 수행하기 위해 사용되며, 단일 갱신자, 다중 판독기를 실시하지 않음을 지정합니다. 참여하는 각 데이터베이스로 COMMIT를 보냅니다. 커밋이 실패하는 경우 응용프로그램이 복구에 책임이 있습니다.

## ONEPHASE

2단계 커밋을 수행하는 데 TM이 사용되지 않음을 지정합니다. 다중

데이터베이스 트랜잭션에서 각 데이터베이스에 의해 수행되는 작업을 커밋하기 위해 1단계 커미트가 사용됩니다.

### **TWOPHASE**

이 프로토콜을 지원하는 데이터베이스 간에 2단계 커미트를 조정하는데 TM이 필요함을 지정합니다.

### **SYNTAX**

프리컴파일 동안 패키지나 바인드 파일의 작성을 제외시킵니다. 이 옵션은 기존 패키지나 바인드 파일을 변경하거나 수정하지 않고 소스 파일의 유효성을 점검하기 위해 사용됩니다. SYNTAX는 SQLERROR CHECK의 동의어입니다.

SYNTAX가 PACKAGE 옵션과 함께 사용되면, PACKAGE는 무시됩니다.

### **TARGET**

현재 플랫폼에서 지원되는 컴파일러 중 하나로 조정되는 수정된 코드를 프리컴파일러가 생성하도록 지시합니다.

### **IBMCOB**

AIX에서, AIX 컴파일러용 IBM COBOL 세트의 코드가 생성됩니다.

### **MFCOB**

Micro Focus COBOL 컴파일러용 코드가 생성됩니다. 모든 TARGET 값이 Linux, UNIX 및 Windows 운영 체제에서 COBOL 프리컴파일러로 지정되지 않으면 이것이 디폴트입니다.

### **ANSI\_COBOL**

ANS X3.23-1985 표준과 호환 가능한 코드가 생성됩니다.

**C** 현재 플랫폼에서 DB2로 지원되는 C 컴파일러와 호환 가능한 코드.

### **CPLUSPLUS**

현재 플랫폼에서 DB2로 지원되는 C++ 컴파일러와 호환 가능한 코드.

### **FORTRAN**

현재 플랫폼에서 DB2로 지원되는 FORTRAN 컴파일러와 호환 가능한 코드.

### **TEXT** *label*

패키지의 설명. 최대 길이는 255문자입니다. 디폴트값은 공백입니다. 이 DRDA 프리컴파일/바인드 옵션은 DB2에서 지원되지 않습니다.

### **TRANSFORM GROUP**

사용자 정의 구조화된 유형 값을 호스트 프로그램과 교환하기 위해 동적 SQL 문에 사용되는 변환 그룹의 이름을 지정합니다. 이 변환 그룹은 동적 SQL문이나 매개변수와 결과를 외부 함수 또는 메소드와 교환하기 위해 사용되지 않습니다. 이 옵션은 DRDA 서버에서 지원되지 않습니다.

*groupname*

길이가 최대 128바이트인 SQL ID. 그룹 이름은 규정자 접두부를 포함할 수 없으며 데이터베이스 사용을 위해 예약되어 있으므로 접두부 SYS로 시작할 수 없습니다. 호스트 변수와 상호작용하는 정적 SQL문에서, 구조화된 유형의 값 교환을 위해 사용되는 변환 그룹의 이름은 다음과 같습니다.

- 있는 경우, TRANSFORM GROUP 바인드 옵션에 있는 그룹 이름
- 있는 경우, 원래 프리컴파일 시간에 지정된 대로 TRANSFORM GROUP prep 옵션에 있는 그룹 이름
- 그룹 이름이 DB2\_PROGRAM인 유형에 대한 변환이 존재하는 경우, DB2\_PROGRAM 그룹
- 위의 조건 중 하나도 존재하지 않는 경우 변환 그룹 없음이 사용됩니다.

다음 오류는 정적 SQL문의 바인드 동안 가능합니다.

- SQLCODE yyy, SQLSTATE xxxxx: 변환이 필요하지만, 정적 변환 그룹이 선택되지 않았습니다.
- SQLCODE yyy, SQLSTATE xxxxx: 선택된 변환 그룹이 교환될 필요가 있는 데이터 유형에 대해 필요한 변환(입력 변수의 경우 TO SQL, 출력 변수의 경우 FROM SQL)을 포함하지 않습니다.
- SQLCODE yyy, SQLSTATE xxxxx: FROM SQL 변환의 결과 유형이 출력 변수의 유형과 호환 가능하지 않거나, TO SQL 변환의 매개변수 유형이 입력 변수의 유형과 호환 가능하지 않습니다.

이 오류 메시지에서, yyyyy는 SQL 오류 코드로 교체되며 xxxxx는 SQL 상태 코드로 교체됩니다.

## VALIDATE

데이터베이스 관리 프로그램가 권한 부여 오류와 오브젝트를 찾을 수 없음 오류를 점검하는 시기를 판별합니다. 패키지 소유자 권한 부여 ID가 유효성 점검에 사용됩니다.

**BIND** 유효성 확인은 프리컴파일/바인드 시간에 수행됩니다. 모든 오브젝트가 존재하지 않거나, 모든 권한이 보유되지 않으면, 오류 메시지가 생성됩니다. SQLERROR CONTINUE가 지정되면, 오류 메시지와 관계없이 패키지/바인드 파일이 생성되지만 오류에 있는 명령문은 실행할 수 없습니다.

**RUN** 유효성 확인이 바인드 시간에 시도됩니다. 모든 오브젝트가 존재하며 모든 권한이 보유되지만, 실행 시간에 더 이상의 점검은 수행되지 않습니다.

모든 오브젝트가 존재하지 않거나 프리컴파일/바인드 시간에 모든 권한이 보유되지 않으면, 경고 메시지가 생성되며, `SQLERROR CONTINUE` 옵션 설정과 관계없이 패키지가 정상적으로 바인드됩니다. 그러나 프리컴파일/바인드 프로세스 동안 해당 점검에 실패한 SQL문의 존재 점검과 권한 점검은 실행 시간에 재실행될 수 있습니다.

## VERSION

패키지의 버전 ID를 정의합니다. 이 옵션이 지정되지 않으면, 패키지 버전은 ""(비어 있는 문자열)입니다.

### *version-id*

최대 64문자의 영숫자 값(\$, #, @, \_, - 또는 .)인 버전 ID를 지정합니다.

## AUTO

버전 ID는 일관성 토큰에서 생성됩니다. 일관성 토큰이 시간소인이면 (LEVEL 옵션이 지정되지 않은 경우), 시간소인은 ISO 문자 형식으로 변환되며 버전 ID로 사용됩니다.

## WCHARTYPE

그래픽 데이터의 형식을 지정합니다.

## CONVERT

`wchar_t` 기본 유형을 사용하여 선언된 호스트 변수는 `wchar_t` 형식으로 된 데이터를 포함하는 것으로 취급됩니다. 이 형식은 데이터베이스 (DBCS 형식)에 저장된 그래픽 데이터의 형식과 바로 호환 가능하지 않으므로, `wchar_t` 호스트 변수의 입력 데이터는 ANSI C 함수 `wcstombs()`를 사용하여 응용프로그램을 위해 DBCS 형식으로 내재적으로 변환됩니다. 유사하게, 출력 DBCS 데이터는 호스트 변수에 저장되기 전에 `mbstowcs()`를 사용하여 `wchar_t` 형식으로 내재적으로 변환됩니다.

## NOCONVERT

`wchar_t` 기본 유형을 사용하여 선언된 호스트 변수는 DBCS 형식으로 된 데이터를 포함하는 것으로 취급됩니다. 이것은 그래픽 데이터의 데이터베이스 내에서 사용되는 형식이지만, C 언어에서 구현되는 원시 (native) `wchar_t` 형식과는 다릅니다. `NOCONVERT` 사용은 응용프로그램과 데이터베이스 간의 그래픽 데이터 변환을 수행하지 않음을 의미하며 효율성을 개선시킬 수 있습니다. 그러나 응용프로그램은 `wchar_t` 형식의 데이터가 데이터베이스 관리 프로그램로 전달되지 않도록 해야 합니다. 이 옵션이 사용되면, `wchar_t` 호스트 변수는 C 와이드 문자열 함수로 조작되지 않아야 하며, 와이드 문자 리터럴(L-literals)로 초기화되지 않아야 합니다.

## 사용 시 참고사항

SQL문과 동일한 호스트 언어를 포함하는 수정된 소스 파일이 생성됩니다. 디폴트로, 연결된 데이터베이스에 패키지가 작성됩니다. 패키지의 이름은 최대 8문자로 파일 이름(확장자 빼고 대문자로)과 같습니다. 패키지 이름의 최대 길이가 128바이트이지만, PACKAGE USING 옵션이 지정된 경우를 제외하고 파일 이름의 최초 8문자만이 DB2의 이전 버전과 호환성을 유지하도록 사용됩니다.

데이터베이스 연결 이후에, PREP는 시작된 트랜잭션 아래에서 실행합니다. 그런 다음 PREP는 COMMIT 또는 ROLLBACK을 발행하여 현재 트랜잭션을 종료하고 다른 트랜잭션을 시작합니다.

아직 존재하지 않는 스키마 이름으로 패키지를 작성하면 해당 스키마가 내재적으로 작성됩니다. 스키마 소유자는 SYSIBM입니다. 스키마에 대한 CREATEIN 특권은 PUBLIC에 권한 부여됩니다.

프리컴파일 동안, 패키지가 작성되고 EXPLSNAP가 지정된 경우를 제외하고 Explain 스냅샷을 얻지 못합니다. 패키지를 작성하는 사용자의 Explain 테이블로 스냅샷을 넣습니다. 유사하게, Explain 테이블 정보는 EXPLAIN이 지정될 때만 캡처되며, 패키지가 작성됩니다.

치명적 오류 또는 100개의 오류를 초과하는 경우 프리컴파일이 중지합니다. 치명적 오류가 발생한 경우, 유틸리티가 프리컴파일을 중지하며, 모든 파일을 닫으며, 패키지를 버립니다.

패키지가 바인드 동작을 표시하면, 다음이 참이 됩니다.

1. BIND 옵션 OWNER의 내재된 또는 명시적 값이 동적 SQL문의 권한 부여 검사에 사용됩니다.
2. BIND 옵션 QUALIFIER의 내재된 또는 명시적 값이 동적 SQL문 내에서 규정되지 않은 자격의 내재된 규정자로서 사용됩니다.
3. 특수 레지스터 CURRENT SCHEMA의 값은 자격에 영향을 미치지 않습니다.

단일 연결 시 다중 패키지가 참조될 경우 해당 패키지에서 준비한 모든 동적 SQL문은 사용되는 특정 패키지 및 환경에 대해 DYNAMICRULES 옵션에서 지정한 동작을 수행합니다.

SQL문이 오류에 있으며 PRECOMPILE 옵션 SQLERROR CONTINUE가 지정된 경우, 명령문은 유효하지 않은 것으로 표시되며 다른 PRECOMPILE이 SQL문의 상태를 변경하기 위해 발행되어야 합니다. 내재된 리바인드 및 명시적 리바인드는 VALIDATE RUN으로 바인드된 패키지에서 유효하지 않은 명령문의 상태를 변경하지 않습니다. 오

브젝트 존재 여부나 리바인드 중 권한 문제점 존재 여부에 따라 내재된 리바인드 및 명시적 리바인드에 걸쳐 명령문은 정적에서 증분식 바인드로 또는 증분식 바인드에서 정적 바인드로 변경할 수 있습니다.

REOPT ONCE 또는 REOPT ALWAYS로 패키지를 바인딩하면 정적 및 동적 명령문 컴파일 및 성능을 변경할 수 있습니다.

Embedded SQL 프로그램의 경우, FEDERATED\_ASYNCRONY 프리컴파일 옵션은 FEDERATED\_ASYNC 구성 매개변수를 사용하여 바인드된 패키지에서 명시적으로 지정된 정적 명령문이 아닙니다. FEDERATED\_ASYNCRONY 옵션이 명시적으로 지정되면, 해당 값은 패키지를 바인딩하는 데 사용되며 특수 레지스터의 초기 값입니다. 그렇지 않으면, 데이터베이스 관리 프로그램 구성 매개변수의 값이 특수 레지스터의 초기 값으로 사용됩니다. FEDERATED\_ASYNCRONY 프리컴파일 옵션은 명시적으로 설정될 때에만 동적 SQL에 영향을 미칩니다.



---

## 제 90 장 PRUNE HISTORY/LOGFILE

복구 실행기록 파일에서 항목을 삭제하거나 의 사용 중인 로그 파일 경로에서 로그 파일을 삭제하는 데 사용됩니다. 복구 실행기록 파일이 과도하게 크고 보유 기간이 긴 경우 이 파일에서 항목을 삭제할 필요가 있을 수 있습니다.

### 권한 부여

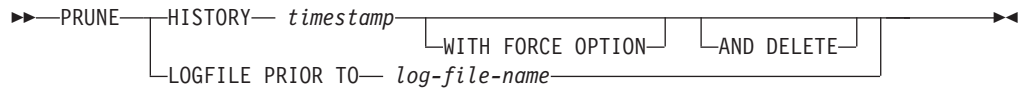
다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*

### 필수 연결

데이터베이스

### 명령 구문



### 명령 매개변수

#### **HISTORY** *timestamp*

삭제될 복구 실행기록 파일의 항목 범위를 식별합니다. 전체 시간소인 (yyyymmddhhmmss 형식) 또는 맨 앞의 접두부(최소한 yyyy)를 지정할 수 있습니다. 제공된 시간소인 이하의 시간소인을 갖는 모든 항목은 복구 실행기록 파일에서 삭제됩니다.

#### **WITH FORCE OPTION**

가장 최근의 리스토어 세트의 일부 항목이 파일에서 삭제되었더라도 지정된 시간소인에 따라 항목이 프론되도록 지정합니다. 리스토어 세트는 해당 백업 이미지의 모든 리스토어를 포함하는 가장 최근의 전체 데이터베이스 백업입니다. 이 매개변수를 지정하지 않으면 정방향 백업 이미지의 모든 항목을 실행기록에 유지보수합니다.

#### **AND DELETE**

실행기록 파일 항목이 제거될 때 연관된 로그 아카이브가 실제로 삭제되도록(위치 정보에 따라) 지정합니다. 이 옵션은 로그 아카이브가 더 이상 필요하지 않

을 때 아카이브 스토리지 스페이스가 복구되는지 확인하는 데 특히 유용합니다. User Exit 프로그램을 통해 로그를 아카이브 중인 경우, 이 옵션을 사용하여 로그를 삭제할 수 없습니다.

**auto\_del\_rec\_obj** 데이터베이스 구성 매개변수를 ON으로 설정하면, AND DELETE 매개변수로 PRUNE HISTORY를 호출 시 실행기록 파일 항목이 프룬(prune)된 경우 실제로 백업 이미지를 삭제하고 복사 이미지를 로드합니다.

#### LOGFILE PRIOR TO *log-file-name*

로그 파일 이름의 문자열을 지정합니다(예: S0000100.LOG). 지정된 로그 파일 이전의 모든 로그 파일(지정된 로그 파일은 제외함)을 삭제합니다. **logretain** 데이터베이스 구성 매개변수는 RECOVERY 또는 CAPTURE로 설정해야 합니다.

#### 예:

복구 실행기록 파일에서 1994년 12월 및 그 이전에 수행된 전체 데이터베이스 백업 및 모든 리스토어, 로드, 테이블 스페이스 백업에 대한 항목을 제거하려면 다음을 입력하십시오.

```
db2 prune history 199412
```

199412는 19941201000000으로 해석됩니다.

#### 사용 시 참고사항

WITH FORCE OPTION을 사용할 경우, 데이터베이스의 자동 리스토어에 필요한 항목을 삭제할 수 있습니다. 직접 리스토어는 계속 올라로 작동합니다. 이 명령을 사용하면 db2ckrst 유틸리티가 필수 백업 이미지의 전체 체인을 정확히 분석하지 못하게 할 수 있습니다. WITH FORCE OPTION을 사용하지 않고 PRUNE HISTORY 명령을 사용하면 필요한 항목이 삭제되지 않게 합니다.

상태가 DB2HISTORY\_STATUS\_DO\_NOT\_DELETE인 항목은 프룬되지 않습니다. WITH FORCE OPTION이 사용되면, DB2HISTORY\_STATUS\_DO\_NOT\_DELETE로 표시된 오브젝트는 여전히 프룬되거나 삭제됩니다. UPDATE HISTORY 명령, UPDATE\_HISTORY가 있는 ADMIN\_CMD 또는 db2HistoryUpdate API를 사용하여 복구 실행기록 파일 항목 상태를 DB2HISTORY\_STATUS\_DO\_NOT\_DELETE로 설정할 수 있습니다. DB2HISTORY\_STATUS\_DO\_NOT\_DELETE 상태를 사용하여 키 복구 실행기록 파일 항목이 프룬되지 않게 하고 연관된 복구 오브젝트가 삭제되지 않게 할 수 있습니다.

PRUNE HISTORY 명령을 사용하여 스냅샷 백업 데이터베이스 실행기록 파일 항목을 프룬할 수 있지만 AND DELETE 매개변수를 사용하여 관련된 실제 복구 오브젝트를 삭제할 수 없습니다. 스냅샷 백업 오브젝트를 삭제하는 유일한 방법은 db2acsutil 명령을 사용하는 것입니다.

---

## 제 91 장 PUT ROUTINE

데이터베이스에 루틴을 정의하기 위해 지정된 루틴 SQL 아카이브(SAR) 파일을 사용합니다.

### 권한 부여

*dbadm.* 이 권한은 역할을 통해 상속되지 않고 사용자에게 직접 부여되어야 합니다.

### 필수 연결

데이터베이스. 내재된 연결이 사용 가능한 경우 디폴트 데이터베이스에 연결됩니다.

### 명령 구문

```
▶▶—PUT ROUTINE—FROM—file-name—————▶▶
      |
      |—OWNER—new-owner—————|
      |                           |
      |                           |—USE REGISTERS—|
```

### 명령 매개변수

#### FROM *file-name*

루틴 SQL 아카이브(SAR)가 저장되는 파일의 이름을 지정합니다.

#### OWNER *new-owner*

루틴의 권한 부여 검사에 사용될 새 권한 부여 이름을 지정합니다. 새 소유자에게는 루틴을 정의하는 데 필요한 특권이 필요합니다. OWNER 절이 지정되지 않으면 원래 루틴에 정의된 권한 부여 이름이 사용됩니다.

#### USE REGISTERS

CURRENT SCHEMA 및 CURRENT PATH 특수 레지스터를 사용하여 루틴을 정의함을 나타냅니다. 이 절이 지정되지 않은 경우 디폴트 스키마 및 SQL 경로에 대한 설정은 루틴 정의 시 사용되는 설정입니다. CURRENT SCHEMA는 루틴 정의에서 규정되지 않은 오브젝트 이름의 스키마 이름(루틴의 이름 포함)으로 사용되며 CURRENT PATH는 루틴 정의에서 규정되지 않은 루틴과 데이터 유형을 해석하는 데 사용됩니다.

### 예:

```
PUT ROUTINE FROM procs/procl.sar;
```

### 사용 시 참고사항

단 하나의 프로시저만이 주어진 스키마 아래에 동시에 설치될 수 있습니다.

GET ROUTINE 또는 PUT ROUTINE 조작(또는 해당 프로시저)을 실행할 수 없는 경우 실패 원인에 대한 정보를 제공하는 진단 텍스트와 함께 항상 오류를 리턴합니다 (SQLSTATE 38000). 예를 들어, GET ROUTINE에 제공된 프로시저 이름이 SQL 프로시저를 식별하지 않는 경우, 진단 "-204, 42704" 텍스트가 리턴됩니다. 여기서, "-204" 및 "42704"는 각각 문제점의 원인을 식별하는 SQLCODE 및 SQLSTATE입니다. 이 예의 SQLCODE와 SQLSTATE는 GET ROUTINE 명령에 제공된 프로시저 이름이 정의되지 않음을 나타냅니다.

---

## 제 92 장 QUERY CLIENT

응용프로그램 프로세스에 대한 현재 연결 설정을 리턴합니다.

### 권한 부여

없음

### 필수 연결

없음

### 명령 구문

▶▶—QUERY CLIENT—▶▶

### 명령 매개변수

없음

### 예:

다음은 QUERY CLIENT의 샘플 출력입니다.

응용프로그램 프로세스의 현재 연결 설정값:

```
CONNECT = 1
DISCONNECT = EXPLICIT
MAX_NETBIOS_CONNECTIONS = 1
SQLRULES = DB2
SYNCPOINT = ONEPHASE
CONNECT_DBPARTITIONNUM = CATALOG_DBPARTITIONNUM
ATTACH_DBPARTITIONNUM = -1
```

CONNECT\_DBPARTITIONNUM 및 ATTACH\_DBPARTITIONNUM이 SET CLIENT 명령을 사용하여 설정되지 않는 경우 이들 매개변수는 환경 변수 DB2NODE의 값과 동일한 값을 갖습니다. CONNECT\_DBPARTITIONNUM 또는 ATTACH\_DBPARTITIONNUM 매개변수의 표시된 값이 -1인 경우, 매개변수가 설정되지 않았습니다. 즉, 환경 변수 DB2NODE가 설정되지 않았거나 매개변수가 이전에 발행된 SET CLIENT 명령에서 지정되지 않았습니다.

### 사용 시 참고사항

응용프로그램 프로세스에 대한 연결 설정은 실행 중에 언제든지 쿼리할 수 있습니다.



---

## 제 93 장 QUIESCE

강제로 모든 사용자가 지정된 인스턴스 및 데이터베이스를 해제하고 이를 Quiesce 모드로 넣습니다. 데이터베이스 인스턴스 또는 데이터베이스가 Quiesce 모드에 있는 동안 이에서 관리 태스크를 수행할 수 있습니다. 관리 태스크가 완료된 후, UNQUIESCE 명령을 사용하여 인스턴스 및 데이터베이스를 활성화하고 다른 사용자가 데이터베이스에 연결할 수 있도록 하되 데이터베이스를 종료하고 다른 데이터베이스를 수행할 수 없도록 하십시오.

이 모드에서는 이 제한된 모드에서 권한이 있는 사용자만 인스턴스/데이터베이스에 접속하거나 연결할 수 있습니다. *sysadm*, *sysmaint* 및 *sysctrl* 권한이 있는 사용자는 Quiesce 상태에 있는 인스턴스에 항상 액세스할 수 있으며 *sysadm* 및 *dbadm* 권한이 있는 사용자는 Quiesce 상태에 있는 데이터베이스에 항상 액세스할 수 있습니다.

### 범위

QUIESCE DATABASE의 결과는 데이터베이스에 있는 모든 오브젝트가 Quiesce 모드가 되는 것으로 나타냅니다. 허용된 사용자/그룹 및 *sysadm*, *sysmaint*, *dbadm* 또는 *sysctrl*만 데이터베이스 또는 해당 오브젝트에 액세스할 수 있습니다.

QUIESCE INSTANCE *instance-name*은 인스턴스 *instance-name*에 있는 인스턴스 및 데이터베이스가 Quiesce 모드가 되는 것을 의미합니다. 이 인스턴스는 *sysadm*, *sysmaint*, *sysctrl* 및 허용된 사용자/그룹만 액세스할 수 있습니다.

인스턴스가 Quiesce 모드일 경우 이 인스턴스의 데이터베이스는 Quiesce 모드가 될 수 없습니다.

### 권한 부여

다음 중 하나가 필요합니다.

데이터베이스 레벨 Quiesce의 경우:

- *sysadm*
- *dbadm*

인스턴스 레벨 Quiesce의 경우:

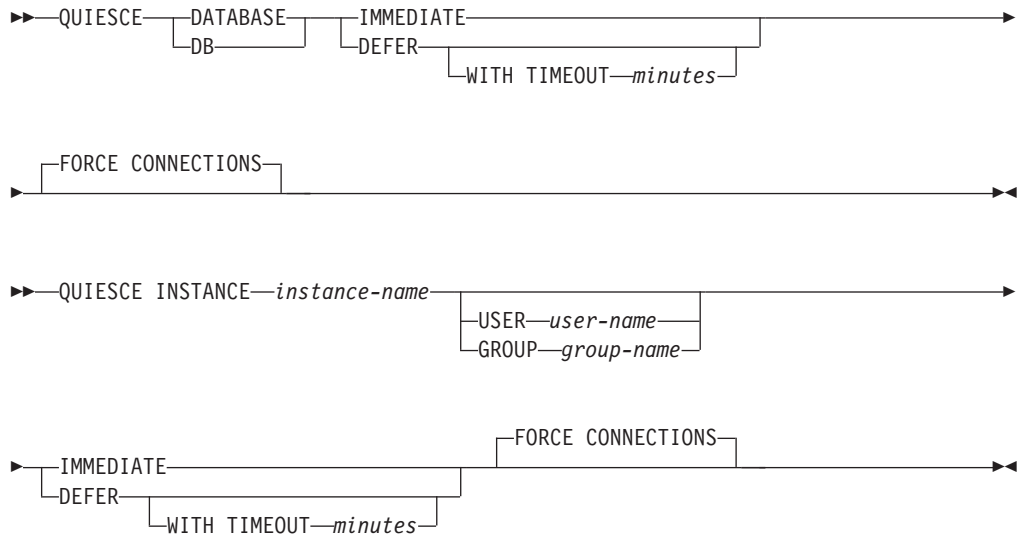
- *sysadm*
- *sysctrl*

## 필수 연결

데이터베이스

(인스턴스 Quiesce에는 데이터베이스 연결이 필요하지 않습니다.)

## 명령 구문



## 명령 매개변수

### DEFER

응용프로그램이 현재 작업 단위를 커밋할 때까지 대기하십시오.

### WITH TIMEOUT *minutes*

응용프로그램이 현재 작업 단위를 커밋할 때까지 대기할 시간을 분 단위로 지정합니다. 단일 파티션 데이터베이스 환경에서 값을 지정하지 않을 경우 디폴트값은 10분입니다. 파티션된 데이터베이스 환경에서는 **start\_stop\_time** 데이터베이스 관리 프로그램 구성 매개변수에 의해 지정된 값을 사용합니다.

### IMMEDIATE

트랜잭션이 커밋될 때까지 대기하지 말고 즉시 트랜잭션을 롤백하십시오.

### FORCE CONNECTIONS

연결을 강제로 해제하십시오.

### DATABASE

데이터베이스를 Quiesce하십시오. 데이터베이스의 모든 오브젝트가 Quiesce 모드에 넣어집니다. *sysadm*, *sysmaint* 및 *sysctrl* 권한이 있는 사용자 및 지정된 그룹의 지정된 사용자만 데이터베이스 또는 해당 오브젝트에 액세스할 수 있습니다.



### **INSTANCE** *instance-name*

인스턴스 *instance-name* 및 이 인스턴스의 데이터베이스가 Quiesce 모드에 넣어집니다. *sysadm*, *sysmaint* 및 *sysctrl* 권한이 있는 사용자 및 지정된 그룹의 지정된 사용자만 이 인스턴스에 액세스할 수 있습니다.

### **USER** *user-name*

Quiesce 상태인 인스턴스에 액세스할 수 있는 사용자 이름을 지정합니다.

### **GROUP** *group-name*

Quiesce 상태인 인스턴스에 액세스할 수 있는 그룹 이름을 지정합니다.

## **예:**

다음 예에서 디폴트 작동은 강제 연결을 수행하는 것이므로 명시적으로 지정할 나타낼 필요가 없으며 이 예에서 제거할 수 있습니다.

```
db2 quiesce instance crankarm user frank immediate force connections
```

다음 예는 데이터베이스에 연결된 모든 사용자를 강제로 해제합니다.

```
db2 quiesce db immediate
```

- 첫 번째 예는 사용자 frank가 계속 데이터베이스를 사용할 수 있도록 하면서 crankarm 인스턴스를 Quiesce합니다.

두 번째 예는 접속된 데이터베이스를 Quiesce하고 *sysadm*, *sysmaint*, *sysctrl* 또는 *dbadm* 권한 중 하나가 있는 사용자를 제외한 모든 사용자가 데이터베이스에 액세스하지 못하게 합니다.

- FORCE CONNECTIONS 옵션이 제공된 경우 이 명령은 강제로 모든 사용자가 데이터베이스 또는 인스턴스를 해제하도록 합니다. FORCE CONNECTIONS는 디폴트 동작이며, 호환성을 위해 명령에 이 매개변수를 사용할 수 있습니다.
- 이 명령은 FORCE CONNECTIONS와 동기화되며 FORCE CONNECTIONS가 완료된 후에만 완료됩니다.

## **사용 시 참고사항**

- QUIESCE INSTANCE 후, *sysadm*, *sysmaint* 또는 *sysctrl* 권한이 있는 사용자 또는 명령에 매개변수로서 제공된 사용자 이름 및 그룹 이름만 인스턴스에 연결할 수 있습니다.
- QUIESCE DATABASE 후, *sysadm*, *sysmaint*, *sysctrl* 또는 *dbadm* 권한과, GRANT/REVOKE 특권이 있는 사용자가 연결될 수 있는 사용자를 지정할 수 있습니다. 이 정보는 데이터베이스 카탈로그 테이블에 영구적으로 저장됩니다.

예를 들면, 다음과 같습니다.

```
grant quiesce_connect on database to <username/groupname>  
revoke quiesce_connect on database from <username/groupname>
```

---

## 제 94 장 QUIESCE TABLESPACES FOR TABLE

테이블에 대한 테이블 스페이스를 Quiesce합니다. 유효한 Quiesce 모드는 공유, 갱신 가능 및 독점으로 세 가지입니다. Quiesce 함수에서 다음 세 가지 상태가 나타날 수 있습니다.

- Quiesced: SHARE
- Quiesced: UPDATE
- Quiesced: EXCLUSIVE

### 범위

단일 파티션 환경에서 이 명령은 로드 조작의 지속 기간 동안 독점 모드에서 로드 조작에 관련된 모든 테이블 스페이스를 Quiesce합니다. 파티션된 데이터베이스 환경에서 이 명령은 데이터베이스 파티션에서 로컬로 작동합니다. 로드 조작이 수행되는 데이터베이스 파티션에 속한 테이블 스페이스의 해당 부분만 Quiesce합니다. 파티션된 테이블의 경우, 테이블과 연관된 SYSDATAPARTITIONS.TBSPACEID 및 SYSDATAPARTITIONS.LONG\_TBSPACEID에 나열되고, 정상 상태이며 접속되거나 접속이 끊어진 모든 테이블 스페이스(예를 들어, 각각 SYSDATAPARTITIONS.STATUS가 '', 'A' 또는 'D'인)가 Quiesce됩니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*
- *load*

### 필수 연결

데이터베이스

### 명령 구문

```
▶▶ QUIESCE TABLESPACES FOR TABLE tablename [ schema.tablename ] [ SHARE | INTENT TO UPDATE | EXCLUSIVE | RESET ] ▶▶
```

## 명령 매개변수

### TABLE

#### *tablename*

규정되지 않은 테이블 이름을 지정합니다. 테이블은 시스템 카탈로그 테이블이 될 수 없습니다.

#### *schema.tablename*

규정된 테이블 이름을 지정합니다. *schema*를 제공하지 않으면 CURRENT SCHEMA를 사용합니다. 테이블은 시스템 카탈로그 테이블이 될 수 없습니다.

### SHARE

Quiesce가 공유 모드가 되도록 지정합니다.

"Quiesce 상태 공유" 요청을 작성하면 트랜잭션은 테이블 스페이스에 대해 부분 공유 잠금을 요청하고 테이블에 대해서는 공유 잠금을 요청합니다. 트랜잭션이 잠금을 획득할 때 테이블 스페이스의 상태는 QUIESCED SHARE로 변경됩니다. 이 상태는 다른 사용자가 보유하는 상태와 충돌하지 않는 경우에만 Quiescer에게 권한 부여됩니다. 테이블 스페이스의 상태는 Quiescer의 데이터베이스 에이전트 ID 및 권한 부여 ID와 함께 테이블 스페이스 테이블에 기록되므로 상태는 지속됩니다. 테이블에 대한 테이블 스페이스가 QUIESCED SHARE 상태인 동안 테이블을 변경할 수 없습니다. 테이블 및 테이블 스페이스에 대한 다른 공유 모드 요청을 사용할 수 있습니다. 트랜잭션이 커밋하거나 롤백할 때 잠금은 릴리스되지만, 테이블에 대한 테이블 스페이스는 상태가 명시적으로 재설정될 때까지 QUIESCED SHARE 상태로 남아 있습니다.

### INTENT TO UPDATE

Quiesce가 갱신 가능 모드가 되도록 지정합니다.

"Quiesce 갱신 가능" 요청을 작성하면 테이블 스페이스가 의도를 가진 독점(IX) 모드로 잠겨지고 테이블은 갱신(U) 모드로 잠겨집니다. 테이블 스페이스의 상태는 테이블 스페이스 테이블에 기록됩니다.

### EXCLUSIVE

Quiesce가 독점 모드가 되도록 지정합니다.

"Quiesce 독점" 요청을 작성하면 트랜잭션은 테이블 스페이스에 대해 강한 배타적 잠금을 요청하고 테이블에 대해서는 강한 배타적 잠금을 요청합니다. 트랜잭션이 잠금을 획득할 때, 테이블 스페이스의 상태는 QUIESCED EXCLUSIVE로 변경됩니다. 테이블 스페이스의 상태는 Quiescer의 데이터베이스 에이전트 ID 및 권한 부여 ID와 함께 테이블 스페이스 테이블에 기록됩니다. 테이블 스페이스가 강한 독점 모드로 보유되어 있으므로 이 테이블 스페이스에 대한 다른 액세스는 허용되지 않습니다. Quiesce 함수(Quiescer)를 호출하는 사용자는 테이블 및 테이블 스페이스에 대해 독점 액세스를 가집니다.

## RESET

테이블 스페이스의 상태가 정상으로 재설정되도록 지정합니다. Quiesce 요청을 발행한 연결이 계속 사용 중이면 Quiesce 상태를 재설정할 수 없습니다.

예 :

```
db2 quiesce tablespaces for table staff share
```

```
db2 quiesce tablespaces for table boss.org intent to update
```

## 사용 시 참고사항

선언된 임시 테이블에 대해서는 이 명령이 지원되지 않습니다.

Quiesce는 지속적 잠금입니다. 이 잠금의 이점은 트랜잭션 실패, 연결 실패 및 심지어 시스템 실패(예: 전원 실패 또는 재부팅) 가운데서도 지속된다는 점입니다.

Quiesce는 연결에 소유됩니다. 연결이 유실될 경우 Quiesce는 남아 있지만 소유자가 없으므로 팬텀 Quiesce라고 합니다. 예를 들어, 삭제 단계 동안 정전으로 인해 로드 조작이 인터럽트될 경우, 로드된 테이블에 대한 테이블 스페이스는 삭제 보류, Quiesce 독점 상태로 남게 됩니다. 데이터베이스가 재시작될 때 이 Quiesce는 알 수 없는(또는 팬텀) Quiesce가 됩니다. 팬텀 Quiesce를 제거하려면 Quiesce 모드가 설정될 때 사용된 것과 동일한 사용자 ID로 연결되어야 합니다.

팬텀 Quiesce를 제거하려면 다음을 수행하십시오.

1. Quiesce 모드가 설정될 때 사용된 것과 동일한 사용자 ID로 데이터베이스에 연결하십시오.
2. LIST TABLESPACES 명령을 사용하여 Quiesce된 테이블 스페이스를 판별하십시오.
3. 현재 Quiesce 상태를 사용하여 해당 테이블 스페이스를 다시 Quiesce하십시오. 예를 들면, 다음과 같습니다.

```
db2 quiesce tablespaces for table mytable exclusive
```

완료되면 새 연결이 Quiesce를 소유하게 되며 로드 조작을 재시작할 수 있습니다.

지정된 시간에 한 테이블 스페이스에서 Quiescer를 5개까지 사용하도록 제한합니다.

Quiescer는 테이블 스페이스의 상태를 덜 제한적인 상태에서 더 제한적인 상태(예: S에서 U로 또는 U에서 X로)로 업그레이드할 수 있습니다. 사용자가 이미 보유되어 있는 상태보다 더 낮은 상태를 요청하면 원래 상태가 리턴됩니다. 상태는 강등되지 않습니다.



---

## 제 95 장 QUIT

명령행 처리기 대화식 입력 모드를 종료하고 운영 체제 명령 프롬프트로 돌아갑니다. 일괄처리 파일이 명령행 처리기에 명령을 입력하는 데 사용 중인 경우 명령은 QUIT, TERMINATE 또는 EOF가 발생할 때까지 처리됩니다.

### 권한 부여

없음

### 필수 연결

없음

### 명령 구문

▶—QUIT—▶

### 명령 매개변수

없음

### 사용 시 참고사항

QUIT는 명령행 처리기 백엔드 프로세스를 종료하거나 데이터베이스 연결을 중단하지 않습니다. CONNECT RESET이 연결을 중단하지만 백엔드 프로세스를 종료하지 않습니다. TERMINATE 명령은 둘 다 수행합니다.





## 제 96 장 REBIND

바인드 파일을 사용하지 않고도 데이터베이스에 저장된 패키지를 재작성할 수 있습니다.

### 권한 부여

다음 중 하나가 필요합니다.

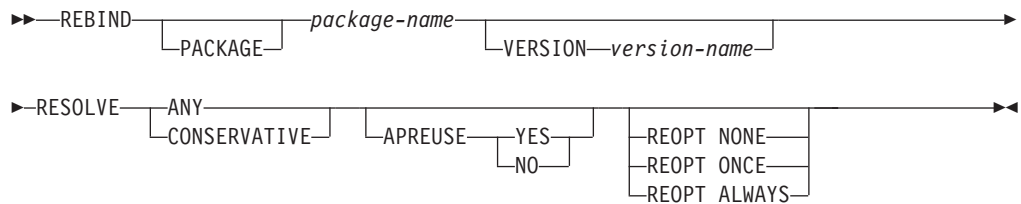
- *dbadm* 권한
- 스키마에서 ALTERIN 특권
- 패키지의 BIND 특권

패키지의 가장 최신 바인더의 ID인 SYSCAT.PACKAGES 시스템 카탈로그 테이블의 BOUNDBY 컬럼에 로그인된 권한 부여 ID가 리바인드의 바인더 권한 부여 ID와, 패키지의 테이블 참조를 위한 디폴트 스키마용으로 사용됩니다. 이 디폴트 규정자는 리바인드 요청을 실행 중인 사용자의 권한 부여 ID와는 다를 수 있습니다. REBIND는 패키지 작성 시에 지정한 동일한 바인드 옵션을 사용합니다.

### 필수 연결

데이터베이스. 데이터베이스 연결이 존재하지 않고 내재된 연결이 사용 가능한 경우 디폴트 데이터베이스에 연결됩니다.

### 명령 구문



### 명령 매개변수

#### PACKAGE *package-name*

리바인드될 패키지를 지정하는 규정되거나 규정되지 않은 이름.

#### VERSION *version-name*

리바인드될 패키지의 특정 버전. 버전이 지정되지 않으면, ""(비어 있는 문자열)이 됩니다.

#### RESOLVE

패키지의 리바인드 수행 시 제한적 바인딩 시맨틱이 사용되는지 여부를 지정합니다. 이는 분석을 위해 SQL 경로를 사용하는 새 오브젝트가 패키지에서 정적

DML 문에 대한 분석 중에 고려되는지 여부에 영향을 줍니다. 이 옵션은 DRDA에서 지원되지 않습니다. 가능한 값은 다음과 같습니다.

**ANY** 오브젝트 분석을 위해 SQL 경로를 사용하는 오브젝트에 대한 참조 분석에 SQL 경로에 있는 가능한 모든 일치이 고려됩니다. 제한적 바인딩 시맨틱은 사용되지 않습니다. 이는 디폴트값입니다.

#### **CONSERVATIVE**

오브젝트 분석을 위해 SQL 경로를 사용하는 오브젝트에 대한 참조 분석 시 SQL 경로에서 마지막 명시적 바인딩 시간소인 이전에 정의된 오브젝트만 고려됩니다. 제한적 바인딩 시맨틱이 사용됩니다. 이 옵션은 작동 불능 패키지에 대해 지원되지 않습니다.

#### **APREUSE**

정적 SQL 액세스 플랜이 재사용되는지 여부를 지정합니다. 이 옵션을 사용할 수 있는 경우 쿼리 컴파일러는 리바인드 중 및 이후 내재된 리바인드와 명시적 리바인드 중에 기존 패키지에서 정적 SQL문의 액세스 플랜을 재사용하려고 시도합니다. 디폴트는 BIND 또는 REBIND 명령이나 ALTER PACKAGE 문의 이전 호출 중에 사용된 값입니다. 값을 판별하려면 SYSCAT.PACKAGES에서 패키지에 대한 APREUSE 컬럼을 쿼리하십시오.

**YES** 쿼리 컴파일러가 패키지에서 명령문의 액세스 플랜을 재사용하려고 시도합니다.

**NO** 쿼리 컴파일러가 패키지에서 명령문의 액세스 플랜을 재사용하려고 시도하지 않습니다.

#### **REOPT**

호스트 변수, 매개변수 표시문자, 전역 변수 및 특수 레지스터의 값을 사용하여 DB2가 액세스 경로를 최적화할지 여부를 지정합니다.

#### **NONE**

호스트 변수, 매개변수 표시문자, 전역 변수 또는 특수 레지스터를 포함하는 제공된 SQL문의 액세스 경로는 이들 변수의 실제 값을 사용하여 최적화되지 않습니다. 이들 변수의 디폴트 추정치가 대신 사용되며, 이 플랜은 캐시되어 차후에 사용됩니다. 이 설정이 디폴트 동작입니다.

#### **ONCE**

제공된 SQL문의 액세스 경로는 쿼리가 처음 실행될 때 호스트 변수, 매개변수 표시문자, 전역 변수 또는 특수 레지스터의 실제 값을 사용하여 최적화됩니다. 이 플랜은 캐시되어 차후에 사용됩니다.

#### **ALWAYS**

제공된 SQL문의 액세스 경로는 항상 실행 시간마다 알려지는 호스트 변수, 매개변수 표시문자, 전역 변수 또는 특수 레지스터의 값을 사용하여 컴파일되고 재최적화됩니다.

## 사용 시 참고사항

REBIND는 리바인드가 성공한 후에 자동으로 트랜잭션을 커밋하지 않습니다. 사용자가 내재적으로 트랜잭션을 커밋해야 합니다. 그러면, 사용자가 특정 통계를 갱신하고 변경사항을 확인하기 위해 패키지를 리바인드 할 때 "what if" 분석을 사용할 수 있습니다. 작업 단위(UOW)에서 다중 리바인드를 허가합니다.

REBIND 명령은 자동 커밋이 사용 가능한 경우에 트랜잭션을 커밋합니다.

이 명령의 기능은 다음과 같습니다.

- 패키지 재작성을 위한 빠른 방법을 제공합니다. 이로서 사용자는 원래 바인드 파일이 없어도 시스템에서 변경사항을 사용할 수 있습니다. 예를 들어, 특정 SQL문이 새로 작성된 인덱스를 사용할 수 있으면 REBIND 명령을 패키지 재작성에 사용할 수 있습니다. RUNSTATS가 실행된 후에 새 통계를 사용하여 패키지를 재작성할 경우에도 REBIND를 사용할 수 있습니다.
- 작동 불능 패키지 재작성 방법을 제공합니다. 작동 불능 패키지는 바인드 유틸리티 또는 리바인드 유틸리티를 호출하여 명시적으로 리바인드해야 합니다. 패키지가 종속된 함수 인스턴스가 삭제된 경우에는 패키지가 작동 불능으로 표시됩니다 (SYSCAT.PACKAGES 시스템 카탈로그의 VALID 컬럼이 X로 설정됨).
- 유효하지 않은 패키지 리바인드를 사용자가 제어할 수 있도록 합니다. 유효하지 않은 패키지는 실행될 때 데이터베이스 관리 프로그램이 자동으로(또는 내재적으로) 리바인드합니다. 그러면, 유효하지 않은 패키지에 대한 첫 번째 SQL 요청 실행이 상당히 지연될 수 있습니다. 초기 지연을 방지하고 내재적인 리바인드가 실패하게 되는 예기치 않은 SQL 오류 메시지가 표시되지 않도록 하려면 시스템이 자동으로 유효하지 않은 패키지를 리바인드하는 것보다 명시적으로 유효하지 않은 패키지를 리바인드하는 것이 나을 수도 있습니다. 예를 들어 다음 데이터베이스 업그레이드에서 데이터베이스에 저장된 모든 패키지는 UPGRADE DATABASE 명령으로 무효화됩니다. 많은 패키지가 관련되어 있는 경우 한 번에 유효하지 않은 패키지 전체를 명시적으로 리바인드하는 것이 나을 수도 있습니다. BIND, REBIND 또는 db2rbind 도구를 사용하여 이 명시적 리바인드를 수행할 수 있습니다.

패키지의 여러 버전(패키지 이름과 작성자가 동일한 많은 버전)이 존재하는 경우 한 번에 하나의 버전만 리바인드됩니다. VERSION 옵션에 지정하지 않은 경우 패키지 버전 디폴트는 ""입니다. 일치하는 이름의 패키지가 하나만 있는 경우에도, 해당 버전이 지정된 버전과 일치하거나 디폴트가 아니면 리바인드되지 않습니다.

명시적으로 패키지를 리바인드하기 위해 BIND 또는 REBIND를 사용할 것인지 선택하는 것은 상황에 따라 다릅니다. REBIND를 사용하는 것이 BIND 사용 시보다 성능이 훨씬 좋기 때문에 반드시 BIND를 사용하도록 지정된 경우가 아닐 때는 REBIND를 사용하는 것이 좋습니다. 그러나 다음과 같은 경우에는 BIND를 사용해야 합니다.

- 프로그램이 수정된 경우(예를 들어 SQL문이 추가 또는 삭제되었거나 패키지가 프로그램 실행 파일에 일치하지 않는 경우).
- 리바인드 중에 임의의 바인드 옵션을 수정하려는 경우. REBIND는 바인드 옵션을 모두 지원하지 않습니다. 예를 들어 사용자가 바인드 프로세스의 일부로 부여되는 패키지의 특권을 가지려고 하면 BIND를 사용해야 합니다. GRANT 옵션이 있기 때문입니다.
- 현재 패키지가 데이터베이스에 없는 경우.
- 모든 바인드 오류를 발견해야 하는 경우. REBIND는 발견한 첫 번째 오류만 리턴하지만 BIND 명령은 바인드 중에 발생하는 처음 100개의 오류를 리턴합니다.

REBIND는 DB2 Connect에서 지원됩니다.

REBIND를 다른 사용자가 사용 중인 패키지에서 실행하면 리바인드 중에 배타적 잠금이 SYSCAT.PACKAGES 시스템 카탈로그 테이블의 패키지 레코드에서 유지되기 때문에 다른 사용자의 논리적 작업 단위가 종료될 때까지 리바인드가 수행되지 않습니다.

REBIND가 실행되면 데이터베이스 관리 프로그램은 SYSCAT.PACKAGES 시스템 카탈로그 테이블에 저장된 SQL문에서 패키지를 재작성합니다.

REBIND에 오류가 발생하면 처리가 중지되고 오류 메시지가 리턴됩니다.

REBIND는 YES 또는 ALL(패키지에 대한 SYSCAT.PACKAGES 카탈로그 테이블 항목의 EXPLAIN\_SNAPSHOT 컬럼에 표시됨)로 설정된 EXPLSNAP 바인드 옵션이나 YES 또는 ALL(패키지에 대한 SYSCAT.PACKAGES 카탈로그 테이블 항목의 EXPLAIN\_MODE 컬럼에 표시됨)로 설정된 EXPLAIN 바인드 옵션을 사용하여 작성된 패키지를 다시 explain합니다. 사용된 Explain 테이블은 원래 바인더의 테이블이 아니라 REBIND 리퀘스터의 테이블입니다.

SQL문에 오류가 있는 것으로 발견되고 BIND 옵션 SQLERROR CONTINUE가 지정된 경우, 문제점이 정정되었어도 명령문은 유효하지 않은 것으로 표시됩니다. REBIND는 유효하지 않은 명령문의 상태를 변경하지 않습니다. VALIDATE RUN으로 바인드된 패키지에서, 오브젝트 존재 여부나 REBIND 중 권한 문제점 존재 여부에 따라 REBIND에 걸쳐 명령문은 정적에서 증분식 바인드로 또는 증분식 바인드에서 정적 바인드로 변경할 수 있습니다.

REOPT ONCE | REOPT ALWAYS로 패키지를 리바인딩하면 정적 및 동적 명령문 컴파일 및 성능을 변경할 수 있습니다.

REOPT를 지정하지 않은 경우 REBIND는 프리컴파일 또는 바인드 시 사용된 기존 REOPT 값을 보존합니다.

## 제 97 장 RECOVER DATABASE

데이터베이스를 특정 시점 또는 로그 끝으로 리스토어하고 롤 포워드합니다.

### 범위

파티션된 데이터베이스 환경의 경우 이 명령은 카탈로그 파티션에서만 호출할 수 있습니다. 특정 시점에서의 데이터베이스 복구 조작은 `db2nodes.cfg` 파일에 나열된 모든 데이터베이스 파티션에 영향을 줍니다. 로그 끝으로의 데이터베이스 복구 조작은 지정된 데이터베이스 파티션에 영향을 줍니다. 파티션이 지정되지 않으면 `db2nodes.cfg` 파일에 나열된 모든 데이터베이스 파티션에 영향을 줍니다.

### 권한 부여

기존 데이터베이스를 복구하려면 다음 중 하나를 수행하십시오.

- `sysadm`
- `sysctrl`
- `sysmaint`

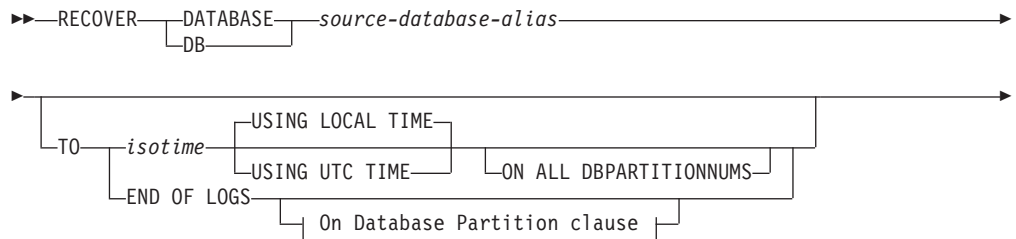
새 데이터베이스로 복구하려면 다음 중 하나를 수행하십시오.

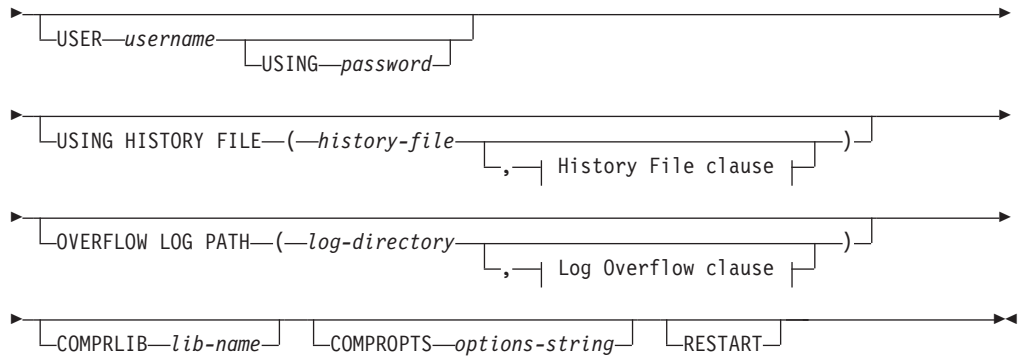
- `sysadm`
- `sysctrl`

### 필수 연결

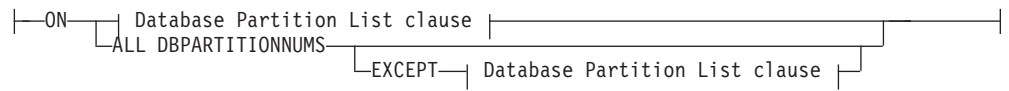
기존 데이터베이스를 복구하려면 데이터베이스 연결이 필요합니다. 이 명령은 지정된 데이터베이스에 대한 연결을 자동으로 설정하며 복구 조작 완료 시 연결을 해제합니다. 새 데이터베이스로 복구하려면 인스턴스 첨부과 데이터베이스 연결이 필요합니다. 인스턴스 첨부은 데이터베이스를 작성하는 데 필요합니다.

### 명령 구문

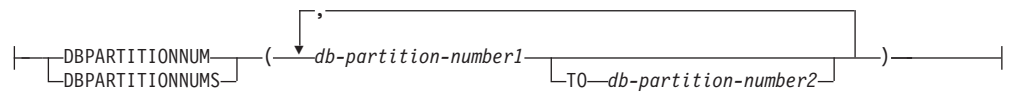




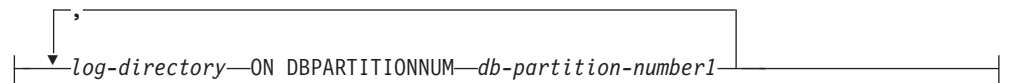
**On Database Partition clause:**



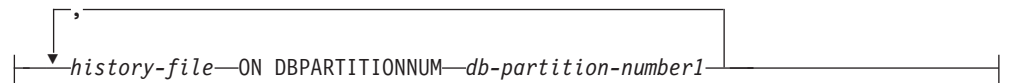
**Database Partition List clause:**



**Log Overflow clause:**



**History File clause:**



**명령 매개변수**

**DATABASE** *database-alias*

복구될 데이터베이스의 별명.

**USER** *username*

데이터베이스가 복구될 사용자 이름

**USING** *password*

사용자 이름을 인증하는 데 사용되는 암호. 암호를 생략하면 사용자가 이를 입력하도록 프롬프트가 표시됩니다.

**TO**

### *isotime*

모든 커밋된 트랜잭션이 복구되는 특정 시점(정확히 해당 시점에 커밋되는 트랜잭션과 이전에 커밋된 모든 트랜잭션 포함).

이 값은 날짜 및 시간 조합을 식별하는 일곱 파트로 구성된 문자열인 시간소인으로서 지정됩니다. 해당 형식은 `yyyy-mm-dd-hh.mm.ss.nnnnnn`(년, 월, 일, 시간, 분, 초, 마이크로초)입니다.. 백업 이미지의 시간소인은 백업 조작이 시작된 로컬 시간을 기반으로 합니다. `CURRENT TIMEZONE` 특수 레지스터는 응용프로그램 서버(AS)에서 UTC와 로컬 시간의 차이를 지정합니다. 이 차이는 시간 지속 기간(10진수로서, 처음 두 자릿수는 시간, 다음 두 자릿수는 분, 마지막 두 자릿수는 초를 나타냄)으로 나타냅니다. 로컬 시간에서 `CURRENT TIMEZONE`을 빼면 로컬 시간이 UTC로 변환됩니다.

### **USING LOCAL TIME**

복구할 특정 시점을 지정합니다. 사용자는 이 옵션을 사용하여 UTC 시간이 아닌 서버의 로컬 시간인 특정 시점으로 복구할 수 있습니다. 디폴트 옵션입니다.

주:

1. 사용자가 복구할 로컬 시간을 지정하면 사용자로 리턴된 모든 메시지 또한 로컬 시간으로 나타납니다. 서버, 파티션된 데이터베이스 환경 및 카탈로그 데이터베이스 파티션의 모든 시간이 변환됩니다.
2. 시간소인 문자열은 서버의 UTC로 변환되므로 시간은 클라이언트가 아닌 서버의 시간대에 로컬입니다. 클라이언트가 특정 시간대에 있고 서버가 다른 시간대에 있는 경우 서버의 로컬 시간을 사용해야 합니다. 이 시간은 클라이언트에 로컬인 제어 센터의 로컬 시간 옵션과 다릅니다.
3. 시간소인 문자열이 서버타임으로 인한 시계의 시간 변화와 유사한 경우 중지 시간이 시계 변환 이전 또는 이후인지 여부를 알고 올바르게 지정해야 합니다.

### **USING UTC TIME**

복구할 특정 시점을 지정합니다.

### **END OF LOGS**

데이터베이스 구성 매개변수 `logpath`에 나열된 모든 온라인 아카이브 로그 파일의 커밋된 모든 트랜잭션이 적용되는 것으로 지정합니다.

### **ON ALL DBPARTITIONNUMS**

`db2nodes.cfg` 파일에 지정된 모든 데이터베이스 파티션에 대해 트랜잭션이 롤포워드되는 것으로 지정합니다. 데이터베이스 파티션 절이 지정되지 않는 경우 디폴트값입니다.

## EXCEPT

데이터베이스 파티션 목록에 지정된 파티션을 제외한 `db2nodes.cfg` 파일에 지정된 모든 데이터베이스 파티션에 대해 트랜잭션이 롤 포워드되는 것으로 지정합니다.

## ON DBPARTITIONNUM | ON DBPARTITIONNUMS

데이터베이스 파티션 세트에 대해 데이터베이스를 롤 포워드합니다.

*db-partition-number1*

데이터베이스 파티션 목록의 데이터베이스 파티션 번호를 지정합니다.

**TO** *db-partition-number2*

*db-partition-number2*를 포함하는 *db-partition-number1*의 모든 데이터베이스 파티션이 데이터베이스 파티션 목록에 포함되도록 두 번째 데이터베이스 파티션 번호를 지정합니다.

## USING HISTORY FILE *history-file*

### *history-file* ON DBPARTITIONNUM

파티션된 데이터베이스 환경에서 다른 실행기록 파일을 허용합니다.

## OVERFLOW LOG PATH *log-directory*

복구 중에 아카이브 로그를 검색할 대체 로그 경로를 지정합니다. **logpath** 데이터베이스 구성 매개변수로 지정된 위치와 다른 위치로 로그 파일이 이동된 경우 이 매개변수를 사용하십시오. 파티션된 데이터베이스 환경에서는 모든 데이터베이스 파티션의 완전한 디폴트 오버플로우 로그 경로입니다. 단일 파티션 데이터베이스에 상대 오버플로우 로그 경로를 지정할 수 있습니다.

OVERFLOW LOG PATH 명령 매개변수는 데이터베이스 구성 매개변수 **overflowlogpath**의 값(있는 경우)을 겹쳐줍니다.

## COMPRLIB *lib-name*

압축 해제를 수행하는 데 사용될 라이브러리 이름을 표시합니다. 이 이름은 서버에 있는 파일을 나타내는 완전한 경로이어야 합니다. 이 매개변수를 지정하지 않으면 DB2가 이미지에 저장된 라이브러리 사용을 시도합니다. 백업이 압축되지 않으면 이 매개변수 값이 무시됩니다. 지정된 라이브러리를 로드할 수 없으면 리스토어 조작이 실패합니다.

## COMPROPTS *options-string*

압축 해제 라이브러리의 초기화 루틴으로 전달되는 2진 데이터 블록을 설명합니다. DB2 데이터베이스 시스템은 이 문자열을 클라이언트에서 서버로 직접 전달하므로 모든 바이트 리버설 또는 코드 페이지 변환 문제는 압축 해제 라이브러리가 직접 처리합니다. 데이터 블록의 첫 번째 문자가 『@』인 경우 DB2 데이터베이스 시스템은 나머지 데이터를 서버에 있는 파일의 이름으로 해석합니다.



다. 그런 다음 DB2 데이터베이스 시스템은 *string*의 콘텐츠를 이 파일의 콘텐츠로 바꾸고 대신 초기화 루틴으로 새 값을 전달합니다. 문자열의 최대 길이는 1024바이트입니다.

## RESTART

RESTART 키워드는 이전 RECOVER 조작이 인터럽트되었거나 완료되지 않은 경우 사용할 수 있습니다. V9.1부터는 가능한 경우 후속 RECOVER 명령이 이전 RECOVER를 대신합니다. RESTART 키워드를 사용하면 RECOVER가 새 리스토어로 시작한 다음 지정된 PIT로 롤 포워드됩니다.

## *log-directory* ON DBPARTITIONNUM

파티션된 데이터베이스 환경에서는 다른 로그 경로가 특정 데이터베이스 파티션의 디폴트 오버플로우 로그 경로를 겹쳐줄 수 있습니다.

## 예:

현재 복구 중인 데이터베이스가 존재하고 실행기록 파일의 최신 버전이 *dftdbpath*에 있는 단일 파티션 데이터베이스 환경의 경우:

1. 최신 백업 이미지를 사용하고 모든 디폴트값을 사용하여 로그 끝으로 롤 포워드하려면 다음을 수행하십시오.

```
RECOVER DB SAMPLE
```

2. 데이터베이스를 PIT로 복구하려면 다음을 실행하십시오. 사용 가능한 최신 이미지는 리스토어되며 PIT에 도달할 때까지 로그가 적용됩니다.

```
RECOVER DB SAMPLE TO 2001-12-31-04.00.00
```

3. 저장된 실행기록 파일 버전을 사용하여 데이터베이스를 복구하려면 다음을 실행하십시오. 예를 들어, 사용자가 현재 실행기록 파일에 더 이상 포함되지 않는 너무 오래된 PIT로 복구해야 하는 경우 사용자는 이 기간부터 실행기록 파일의 버전을 제공해야 합니다. 사용자가 이 기간부터 실행기록 파일을 저장한 경우 이 버전을 사용하여 복구를 수행할 수 있습니다.

```
RECOVER DB SAMPLE TO 1999-12-31-04.00.00  
USING HISTORY FILE (/home/user/old1999files/db2rhist.asc)
```

복구할 데이터베이스가 존재하지 않는 단일 파티션 데이터베이스 환경의 경우 USING HISTORY FILE 절을 사용하여 실행기록 파일을 지정해야 합니다.

1. 실행기록 파일의 백업을 작성하지 않아 사용 가능한 유일한 버전이 백업 이미지의 사본인 경우 권장사항은 RESTORE와 ROLLFORWARD를 차례로 실행하는 것입니다. 그러나 RECOVER를 사용하려면 먼저 이미지에서 특정 위치(예를 들어, /home/user/oldfiles/db2rhist.asc)로 실행기록 파일을 추출한 다음 이 명령을 실행해야 합니다. 이 버전의 실행기록 파일에는 롤 포워드에 필요한 로그 파일에 대한 정보가 들어 있지 않으므로 이 실행기록 파일은 RECOVER에는 유용하지 않습니다.

```
RECOVER DB SAMPLE TO END OF LOGS
USING HISTORY FILE (/home/user/fromimage/db2rhist.asc)
```

2. 실행기록의 백업 사본을 주기적으로 또는 자주 작성한 경우 USING HISTORY FILE 절을 사용하여 실행기록 파일의 이 버전을 지정해야 합니다. 파일이 /home/user/myfiles/db2rhist.asc인 경우 다음 명령을 실행하십시오.

```
RECOVER DB SAMPLE TO PIT
USING HISTORY FILE (/home/user/myfiles/db2rhist.asc)
```

이 경우 실행기록 파일의 임의 사본을 사용할 수 있습니다. 해당 사본은 최신 사본이 아니어도 되며 요청된 특정 시점(PIT) 이전에 수행된 백업을 포함하면 됩니다.

모든 데이터베이스 파티션에 데이터베이스가 존재하는 파티션된 데이터베이스 환경의 경우 모든 데이터베이스 파티션의 **dftdbpath**에서 최신 실행기록 파일을 사용할 수 있습니다.

1. 데이터베이스를 모든 노드의 PIT로 복구합니다. DB2는 리스토어 작업을 시작하기 전에 모든 노드에서 PIT에 도달할 수 있는지 확인합니다.

```
RECOVER DB SAMPLE TO 2001-12-31-04.00.00
```

2. 데이터베이스를 모든 노드의 이 PIT로 복구합니다. DB2는 리스토어 작업을 시작하기 전에 모든 노드에서 PIT에 도달할 수 있는지 확인합니다. 각 노드의 RECOVER 작업은 단일 파티션 RECOVER와 같습니다.

```
RECOVER DB SAMPLE TO END OF LOGS
```

3. 실행기록 파일의 최신 버전은 **dftdbpath**에 있지만 여러 특정 실행기록 파일을 사용할 수 있습니다. 다르게 지정되지 않는 한 각 데이터베이스 파티션은 /home/user/oldfiles/db2rhist.asc에 있는 로컬 실행기록 파일을 사용합니다. 단, 노드 2와 4의 경우 노드 2는 /home/user/node2files/db2rhist.asc를 사용하고 노드 4는 /home/user/node4files/db2rhist.asc를 사용합니다.

```
RECOVER DB SAMPLE TO 1999-12-31-04.00.00
USING HISTORY FILE (/home/user/oldfiles/db2rhist.asc,
/home/user/node2files/db2rhist.asc ON DBPARTITIONNUM 2,
/home/user/node4files/db2rhist.asc ON DBPARTITIONNUM 4)
```

4. 모든 노드 대신 노드 서브세트를 복구할 수 있지만 이 경우 PIT RECOVER를 수행할 수 없으며 EOL로 복구를 완료해야 합니다.

```
RECOVER DB SAMPLE TO END OF LOGS ON DBPARTITIONNUMS(2 TO 4, 7, 9)
```

데이터베이스가 존재하지 않는 파티션된 데이터베이스 환경의 경우:

1. 실행기록 파일의 백업을 작성하지 않아 사용 가능한 유일한 버전이 백업 이미지의 사본인 경우 권장사항은 RESTORE와 ROLLFORWARD를 차례로 실행하는 것입니다. 그러나 RECOVER를 사용하려면 먼저 이미지에서 특정 위치(예를 들어, /home/user/oldfiles/db2rhist.asc)로 실행기록 파일을 추출한 다음 이 명령

을 실행해야 합니다. 이 버전의 실행기록 파일에는 롤 포워드에 필요한 로그 파일에 대한 정보가 들어 있지 않으므로 이 실행기록 파일은 복구에는 유용하지 않습니다.

```
RECOVER DB SAMPLE TO PIT
USING HISTORY FILE (/home/user/fromimage/db2rhist.asc)
```

2. 실행기록의 백업 사본을 주기적으로 또는 자주 작성한 경우 USING HISTORY FILE 절을 사용하여 실행기록 파일의 이 버전을 지정해야 합니다. 이 파일이 /home/user/myfiles/db2rhist.asc인 경우 다음 명령을 실행할 수 있습니다.

```
RECOVER DB SAMPLE TO END OF LOGS
USING HISTORY FILE (/home/user/myfiles/db2rhist.asc)
```

## 사용 시 참고사항

- 데이터베이스를 복구하려면 테이프 디바이스를 사용하여 로드를 복구해야 합니다. 다른 테이프에 대한 프롬프트가 표시되는 경우 사용자는 다음 중 하나를 사용하여 응답할 수 있습니다.
  - c** 계속. 경고 메시지를 생성한 디바이스를 사용하여 계속합니다(예를 들어, 새로운 테이프가 마운트된 경우).
  - d** 디바이스 종료. 경고 메시지를 생성한 디바이스를 사용하여 중지합니다(예를 들어, 더 이상의 테이프가 없는 경우).
  - t** 종료. 모든 디바이스를 종료합니다.
- 복구 조작의 리스토어 부분에서 실패하는 경우 RECOVER DATABASE 명령을 재실행할 수 있습니다. 리스토어 조작에 성공했지만 롤 포워드 조작 중에 오류가 발생한 경우 ROLLFORWARD DATABASE 명령을 실행할 수 있습니다. 전체 복구 조작을 재실행하지 않아도 되고 시간 또한 소요되기 때문입니다.
- 파티션된 데이터베이스 환경에서 복구 조작의 리스토어 부분에서 오류가 발생하는 경우 단일 데이터베이스 파티션만의 오류일 수 있습니다. 모든 데이터베이스 파티션에서 데이터베이스를 리스토어하는 RECOVER DATABASE 명령을 재실행하는 것보다 실패한 데이터베이스 파티션에 대해 RESTORE DATABASE 명령을 실행한 다음 ROLLFORWARD DATABASE 명령을 실행하는 것이 효율적입니다.



---

## 제 98 장 REDISTRIBUTE DATABASE PARTITION GROUP

데이터베이스 파티션 그룹에 있는 데이터베이스 파티션 간에 데이터를 재분배합니다. 특정 시스템 요구사항에 맞게 데이터의 목표 분산이 일정(디폴트)하거나 사용자 지정될 수 있습니다.

REDISTRIBUTE DATABASE PARTITION GROUP 명령은 데이터베이스 파티션 그룹에 있는 모든 파티션 사이에 데이터를 재분배합니다. 이는 데이터베이스 파티션 그룹에 있는 모든 오브젝트에 영향을 주며 하나의 오브젝트로만 제한될 수 없습니다.

이 명령은 카탈로그 데이터베이스 파티션에서만 발행할 수 있습니다. 어떤 데이터베이스 파티션이 각 데이터베이스에 대한 카탈로그 데이터베이스 파티션인지 판별하려면 LIST DATABASE DIRECTORY 명령을 사용하십시오.

### 범위

이 명령은 데이터베이스 파티션 그룹에 있는 모든 데이터베이스 파티션에 영향을 줍니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

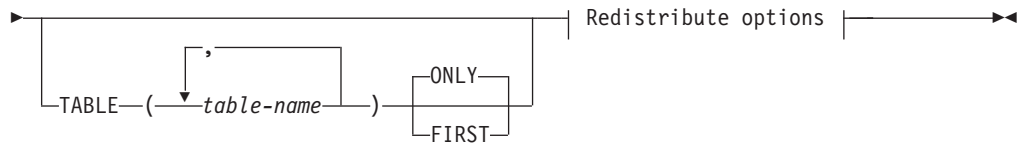
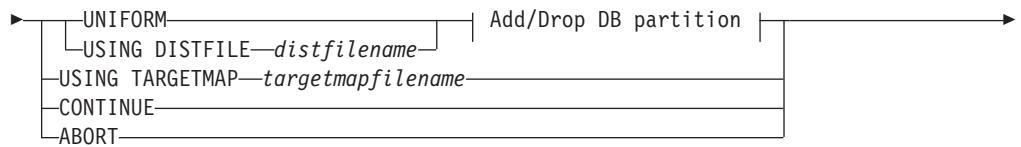
- SYSADM
- SYSCTRL
- DBADM

또한 다음과 같은 권한 그룹 중 하나도 필요합니다.

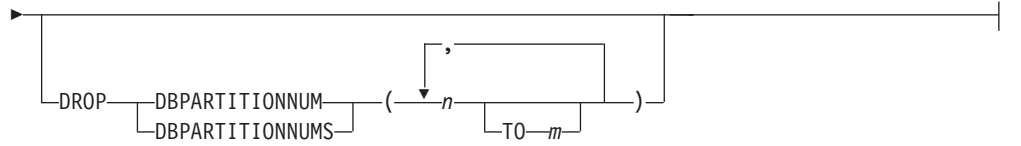
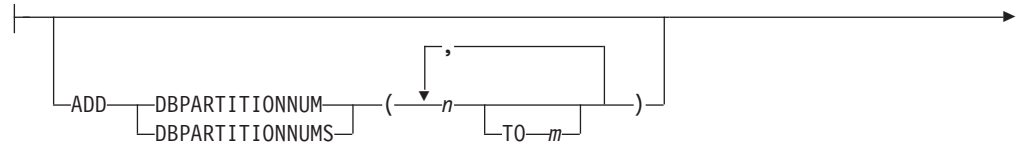
- 재분배 중인 데이터베이스 파티션 그룹의 모든 테이블에 대한 DELETE, INSERT, SELECT 특권
- DATAACCESS 권한

### 명령 구문

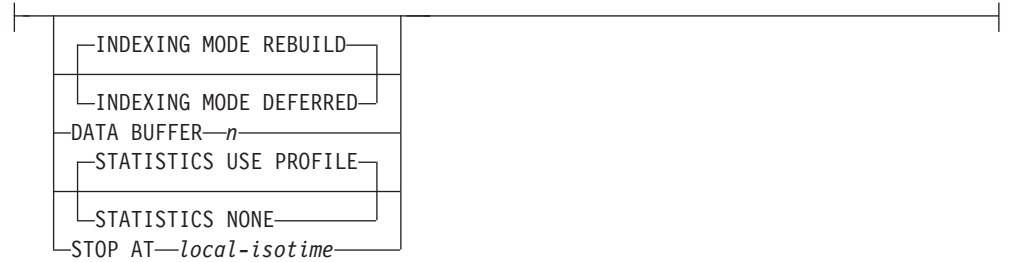
```
▶▶—REDISTRIBUTE DATABASE PARTITION GROUP—db-partition-group————▶▶  
▶ [NOT ROLLFORWARD RECOVERABLE]————▶▶
```



**Add/Drop DB partition:**



**Redistribute options:**



**명령 매개변수**

**DATABASE PARTITION GROUP *db-partition-group***

데이터베이스 파티션 그룹 이름. 이 한 부분으로 된 이름은 SYSCAT.DBPARTITIONGROUPS 카탈로그 테이블에 설명된 데이터베이스 파티션 그룹을 식별합니다. 데이터베이스 파티션 그룹은 현재 재분배될 수 없습니다.

주: IBMCATGROUP 및 IBMTEMPGROUP 데이터베이스 파티션 그룹에 있는 테이블을 재분배할 수 없습니다.

**NOT ROLLFORWARD RECOVERABLE**

이 옵션 사용 시, REDISTRIBUTE DATABASE PARTITION GROUP 명령은 롤 포워드 복구 가능하지 않습니다.

- 내부 삽입 및 삭제 조작 대신 대량으로 데이터가 이동됩니다. 이렇게 하면 테이블을 스캔하고 액세스하는 시간을 줄여 더 나은 성능을 가져옵니다.
- 각 삽입 및 삭제 조작에는 로그 레코드가 더 이상 필요하지 않습니다. 이는 데이터 재분배 시 사용자 시스템에 사용 중인 대량의 로그 스페이스 및 로그 아카이브 스페이스 관리가 필요하지 않음을 의미합니다. 과거에 대량의 사용 중인 로그 스페이스와 스토리지 요구사항으로 인해 단일 데이터 재분배 조작을 여러 개의 더 작은 재분배 태스크로 구분함으로써 단말 간 데이터 재분배 조작 완료에 더 많은 시간이 걸릴 수 있는 상황에서는 특히 유용합니다.
- REDISTRIBUTE DATABASE PARTITION GROUP 명령에 NOT ROLLFORWARD RECOVERABLE 옵션을 사용하는 경우 재분배 조작 시에 XML 컬럼을 포함하는 테이블에 대해 INDEXING MODE DEFERRED 옵션이 사용됩니다. 테이블에 XML 컬럼이 포함되지 않은 경우 재분배 조작은 명령 실행 시에 지정한 인덱스 모드를 사용합니다.

이 옵션이 사용되지 않으면, 모든 행 이동의 확장 로깅이 수행되어 인터럽트, 오류 또는 다른 비즈니스 필요에 따라 나중에 데이터베이스를 복구할 수 있습니다.

## UNIFORM

데이터가 해시 파티션에 균등하게 분산되도록 지정하지만(즉, 모든 해시 파티션이 동일한 행 수를 갖는 것으로 간주됨) 각 데이터베이스 파티션에 동일한 해시 파티션 수가 맵핑되지 않습니다. 재분배 후 데이터베이스 파티션 그룹의 모든 데이터베이스 파티션은 대략 동일한 수의 해시 파티션을 갖게 됩니다.

## USING DISTFILE *distfilename*

분산 키 값의 분산이 균등하지 않을 경우, 이 옵션을 사용하여 데이터베이스 파티션 그룹의 데이터베이스 파티션에서 균등한 데이터 재분배를 달성하십시오.

32 768 해시 파티션에서 현재 데이터 분산을 표시하려면 *distfilename*을 사용하십시오.

각 해시 파티션이 표시하는 데이터의 양을 표시하려면 행 계수, 바이트 볼륨 또는 기타 수치를 사용하십시오. 이 유틸리티는 파티션과 연관된 정수 값을 해당 파티션의 가중치로 읽습니다. *distfilename*이 지정될 때 유틸리티는 데이터베이스 파티션 그룹의 데이터베이스 파티션에서 데이터를 가능한 균등하게 재분배하는 데 사용하는 목표 분산 맵을 생성합니다. 재분배 후, 데이터베이스 파티션 그룹에 있는 각 데이터베이스 파티션의 가중치는 대략적으로 동일합니다(데이터베이스 파티션의 가중치는 해당 데이터베이스 파티션에 맵핑되는 모든 해시 파티션의 가중치 합계임).

예를 들어, 입력 분산 파일에는 다음과 같은 항목이 포함될 수 있습니다.

10223  
1345  
112000  
0  
100  
...

예에서, 해시 파티션 2의 가중치는 112000이며 파티션 3(가중치 0)에는 데이터가 전혀 맵핑되지 않습니다.

*distfilename*은 32 768 양의 정수값을 문자 형식으로 포함해야 합니다. 이들 값의 합계는 4 294 967 295 이하가 되어야 합니다.

*distfilename*의 경로를 지정하지 않은 경우 현재 디렉토리를 사용합니다.

#### **USING TARGETMAP** *targetmapfilename*

*targetmapfilename*에 지정된 파일은 목표 분산 맵으로 사용됩니다. 데이터 재분배는 이 파일에 따라 수행됩니다. 경로를 지정하지 않으면 현재 디렉토리를 사용합니다.

목표 맵에 포함된 데이터베이스 파티션이 데이터베이스 파티션 그룹에 있지 않으면 오류가 리턴됩니다. REDISTRIBUTE DATABASE PARTITION GROUP 명령을 실행하기 전에 ALTER DATABASE PARTITION GROUP ADD DBPARTITIONNUM문을 발행하십시오.

목표 맵에서 제외된 데이터베이스 파티션이 데이터베이스 파티션 그룹에 있는 경우, 이 데이터베이스 파티션은 파티션되는 데 포함되지 않습니다. 이러한 데이터베이스 파티션은 REDISTRIBUTE DATABASE PARTITION GROUP 명령 실행 후에 ALTER DATABASE PARTITION GROUP DROP DBPARTITIONNUM문을 사용하여 삭제될 수 있습니다.

#### **CONTINUE**

이전에 실패하거나 중지된 REDISTRIBUTE DATABASE PARTITION GROUP 조작을 계속 수행합니다. 아무 것도 발생하지 않으면 오류가 리턴됩니다.

#### **ABORT**

이전에 실패하거나 중지된 REDISTRIBUTE DATABASE PARTITION GROUP 조작을 중단합니다. 아무 것도 발생하지 않으면 오류가 리턴됩니다.

#### **ADD**

**DBPARTITIONNUM** *n*

**TO** *m*

*n* 또는 *n TO m*은 데이터베이스 파티션 그룹에 추가된 데이터베이스 파티션 번호 목록을 지정합니다. 지정된 파티션은 데이터베이스 파티션 그룹에 정의되어 있지 않아야 합니다(SQLSTATE 42728). 이것은 ADD



DBPARTITIONNUM 절이 지정된 ALTER DATABASE PARTITION GROUP문을 실행하는 것과 같습니다.

#### **DBPARTITIONNUMS *n***

**TO *m***

*n* 또는 *n TO m*은 데이터베이스 파티션 그룹에 추가된 데이터베이스 파티션 번호 목록을 지정합니다. 지정된 파티션은 데이터베이스 파티션 그룹에 정의되어 있지 않아야 합니다(SQLSTATE 42728). 이것은 ADD DBPARTITIONNUM 절이 지정된 ALTER DATABASE PARTITION GROUP문을 실행하는 것과 같습니다.

주: 이 옵션을 사용하여 데이터베이스 파티션을 추가하면, 테이블 스페이스의 컨테이너는 데이터베이스 파티션 그룹에서 가장 번호가 낮은 기존 파티션의 해당 테이블 스페이스의 컨테이너에 기준합니다. 이로 인해 컨테이너 간의 이름지정 충돌이 발생하는 경우, 이는 새 파티션이 기존 컨테이너와 동일한 실제 머신에 있을 경우 발생할 수 있으며, 이때 이 옵션을 사용하지 말아야 합니다. 대신 ALTER DATABASE PARTITION GROUP문은 REDISTRIBUTE DATABASE PARTITION GROUP 명령을 발행하기 전에 WITHOUT TABLESPACES 옵션과 함께 사용되어야 합니다. 그러면 수동으로 적절한 이름을 지정하여 테이블 스페이스 컨테이너를 작성할 수 있습니다.

### **DROP**

#### **DBPARTITIONNUM *n***

**TO *m***

*n* 또는 *n TO m*은 데이터베이스 파티션 그룹에서 삭제되는 데이터베이스 파티션 번호 목록을 지정합니다. 지정된 파티션은 데이터베이스 파티션 그룹에 정의되어 있어야 합니다(SQLSTATE 42729). 이것은 DROP DBPARTITIONNUM 절이 지정된 ALTER DATABASE PARTITION GROUP문을 실행하는 것과 같습니다.

#### **DBPARTITIONNUMS *n***

**TO *m***

*n* 또는 *n TO m*은 데이터베이스 파티션 그룹에서 삭제되는 데이터베이스 파티션 번호 목록을 지정합니다. 지정된 파티션은 데이터베이스 파티션 그룹에 정의되어 있어야 합니다(SQLSTATE 42729). 이것은 DROP DBPARTITIONNUM 절이 지정된 ALTER DATABASE PARTITION GROUP문을 실행하는 것과 같습니다.

## TABLE *tablename*

재분배 처리를 위한 테이블 순서를 지정합니다.

### ONLY

테이블 순서 다음에 **ONLY** 키워드(디폴트)가 오면, 지정된 테이블만이 재분배됩니다. 남아 있는 테이블은 나중에 **REDISTRIBUTE CONTINUE** 명령으로 처리될 수 있습니다. 이는 디폴트값입니다.

### FIRST

테이블 순서 다음에 **FIRST** 키워드가 표시되면, 지정된 테이블이 표시된 순서로 재분배되고 데이터베이스 파티션 그룹에 남아있는 테이블은 무작위 순서로 재분배됩니다.

## INDEXING MODE

**NOT ROLLFORWARD RECOVERABLE** 옵션이 지정되면 이 매개변수는 재분배 중 인덱스가 유지되는 방식을 지정합니다. 유효한 값은 다음과 같습니다.

### REBUILD

인덱스는 스크래치에서 재빌드됩니다. 이 옵션을 사용하기 위해 인덱스를 유효하게 할 필요는 없습니다. 이 옵션의 사용 결과로, 인덱스 페이지가 디스크에서 함께 클러스터됩니다.

### DEFERRED

재분배는 인덱스를 유지하려고 하지 않습니다. 인덱스는 새로 고칠 필요가 있는 것으로 표시됩니다. 그러한 인덱스에 대한 최초 액세스는 재빌드를 강제하거나, 데이터베이스 재시작 시 인덱스를 재빌드할 수 있습니다.

주: 비MDC 테이블의 경우 테이블에 유효하지 않은 인덱스가 있으면, **INDEXING MODE DEFERRED**를 지정하지 않은 경우 **REDISTRIBUTE DATABASE PARTITION GROUP** 명령이 자동으로 인덱스를 재빌드합니다. MDC 테이블의 경우 **INDEXING MODE DEFERRED**를 지정하면, 유틸리티가 MDC 테이블을 처리하기 위해 복합 인덱스를 필요로 하므로 테이블 재분배가 시작되기 전에 유효하지 않은 복합 인덱스가 재빌드됩니다.

## DATA BUFFER *n*

유틸리티에서 데이터 전송에 필요한 버퍼 스페이스로 사용할 4KB 페이지의 수를 지정합니다. 지정된 값이 지원되는 최소값보다 작으면, 최소값이 사용되며 경고는 리턴하지 않습니다. 이 메모리는 유틸리티 힙에서 바로 할당되며 크기는 **util\_heap\_sz** 데이터베이스 구성 매개변수로 수정할 수 있습니다. 값이 지정되지 않으면 처음에 각 테이블 처리 시 런타임에 유틸리티에 의해 적절한 디폴트

값이 계산됩니다. 특히, 테이블의 시간 재분배 시 유틸리티 힙에서 사용 가능한 메모리의 50%를 사용하고 다양한 테이블 등록 정보를 고려하는 것이 디폴트입니다.

#### **STOP AT *local-isotime***

이 옵션이 지정되면, 각 테이블의 데이터 재분배를 시작하기 전에 *local-isotime* 을 현재 로컬 시간소인과 비교합니다. 지정된 *local-isotime*이 현재 로컬 시간 소인과 같거나 이전이면, 유틸리티는 경고 메시지로 중지합니다. 중지 시 진행 중인 테이블의 데이터 재분배 처리는 인터럽트 없이 완료됩니다. 테이블의 새 데이터 재분배 처리는 시작되지 않습니다. **CONTINUE** 옵션을 사용하여 처리 되지 않은 테이블을 재분배할 수 있습니다. 이 *local-isotime* 값은 조합된 날짜 및 시간을 식별하는 7파트 문자열 시간소인으로 지정됩니다. 형식은 로컬 시간 으로 표시된 *yyyy-mm-dd-hh.mm.ss.nnnnnn*(연도, 월, 일, 시, 분, 초, 마이크로 초)입니다.

#### **STATISTICS**

이 옵션은 통계 프로파일이 있는 테이블의 통계를 유틸리티가 수집해야 함을 지정합니다. 이 옵션 지정은 데이터 재분배 완료 이후 별도로 **RUNSTATS** 명령 을 발행하는 것보다 더 효과적입니다.

#### **USE PROFILE**

통계 프로파일이 있는 테이블에 대한 통계가 수집됩니다. 통계 프로파일 이 없는 테이블의 경우, 어떤 것도 완료되지 않습니다. 이는 디폴트 값입니다.

#### **NONE**

테이블의 통계는 수집되지 않습니다.

### **예: 재분배 단계**

노드 그룹에서 노드를 삭제하거나 추가할 수 있습니다. 다음은 노드 그룹에 새 노드를 추가하고 데이터를 재분배하기 위한 단계입니다. 추가된 데이터베이스 파티션은 분산 맵 에 없지만 데이터베이스 파티션 그룹에 있는 테이블 스페이스에 대한 컨테이너는 작성 되었습니다. 데이터베이스 파티션은 데이터베이스 파티션 그룹 재분배 조작이 완료되었 을 때 데이터베이스 파티션이 분산 맵에 추가됩니다.

1. 재분배해야 할 노드 그룹을 식별하십시오. 이 문서에서 재분배해야 하는 노드 그룹은 『sampleNodegrp』입니다.
2. 재분배하기 전에 사용하지 않도록 설정하거나 제거해야 하는 오브젝트를 식별하십시오.
  - a. MQT 복제: 이 유형의 MQT는 **REDISTRIBUTE** 유틸리티의 일부로 지원되지 않습니다. 재분배를 실행하기 전에 삭제하고 이후에 다시 작성해야 합니다.

```
SELECT tabschema, tablename
FROM syscat.tables
WHERE partition_mode = 'R'
```

- b. 테이블에 기록 이벤트 모니터: 재분배되는 데이터베이스 파티션 그룹에 있는 테이블을 포함하는 자동으로 활성화되는 모든 테이블에 기록 이벤트 모니터를 사용하지 않도록 설정해야 합니다.

```
SELECT distinct evmonname FROM syscat.eventtables E
JOIN syscat.tables T on T.tabname = E.tabname AND T.tabschema = E.tabschema
JOIN syscat.tablespace S on S.tbspace = T.tbspace AND S.ngname = 'sampleNodegrp'
```

- c. Explain 테이블: 단일 파티션 노드 그룹에서 Explain 테이블을 작성할 것을 권장합니다. 그러나 재분배가 필요한 노드 그룹에 정의된 경우, 재분배 이전에 삭제하고 재분배 완료 시 재정의할 것을 고려할 수 있습니다(현재까지 생성된 데이터를 유지보수하지 않아도 되는 경우).
- d. 테이블 액세스 모드 및 로드 상태: 재분배되는 노드 그룹의 모든 테이블이 전체 액세스 모드 상태이고 보류 중이거나 진행 중인 상태의 로드가 없어야 합니다.

```
SELECT distinct trim(T.creator) || '#' || trim(T.name)
AS name, T.access_mode, A.load_status
FROM sysibm.systables T, sysibm.sysnodegroups N, sysibmadm.admintabinfo A
WHERE T.pmap_id = N.pmap_id
AND A.tabschema = T.creator
AND A.tabname = T.name
AND N.name = 'sampleNodegrp'
AND (T.access_mode <> 'F' or A.load_status is not null)
```

- e. 통계 프로파일: 통계 프로파일이 테이블에 대해 정의된 경우 테이블 통계는 재분배 프로세스의 일부로 갱신 가능합니다. REDISTRIBUTE 유틸리티가 테이블의 통계를 갱신하는 경우 재분배를 위해 모든 데이터가 스캔되어 RUNSTATS에 대해 추가 데이터 스캔이 필요하지 않기 때문에 입출력이 감소합니다.

```
RUNSTATS on table schema.table
USE PROFILE runstats_profile
SET PROFILE ONLY
```

3. 데이터베이스 구성을 검토하십시오. **util\_heap\_sz**는 데이터베이스 파티션 간의 데이터 이동 처리에 매우 중요합니다. 재분배 지속기간 동안 가능한 많은 메모리를 **util\_heap\_sz**에 할당하십시오. 인덱스 재빌드가 재분배의 일부로 수행되는 경우 충분한 **sortheap**이 필요합니다. 재분배 성능을 향상시키기 위해서는 **util\_heap\_sz** 및 **sortheap**을 필요한 만큼 늘리십시오.
4. 새 데이터베이스 파티션에 사용할 데이터베이스 구성 설정을 검색하십시오. 데이터베이스 파티션을 추가할 때 디폴트 데이터베이스 구성이 사용됩니다. 결과적으로, 전체 웨어하우스 간의 균형을 조정하기 위해 REDISTRIBUTE 명령이 실행되기 전에 새 노드에서 데이터베이스 구성을 갱신하는 것이 중요합니다.

```
SELECT name,
CASE WHEN deferred_value_flags = 'AUTOMATIC'
THEN deferred_value_flags
ELSE substr(deferred_value,1,20)
END
AS deferred_value
FROM sysibmadm.dbcfg
```

```

WHERE dbpartitionnum = existing-node
AND deferred_value != ''
AND name NOT IN ('hadr_local_host','hadr_local_svc','hadr_peer_window',
'hadr_remote_host','hadr_remote_inst','hadr_remote_svc',
'hadr_syncmode','hadr_timeout','backup_pending','codepage',
'codeset','collate_info','country','database_consistent',
'database_level','hadr_db_role','log_retain_status',
'loghead','logpath','multipage_alloc','numsegs','pagesize',
'release','restore_pending','restrict_access',
'rollfwd_pending','territory','user_exit_status',
'number_compat','varchar2_compat','database_memory')

```

5. 최근 복구점을 확인하려면 재분배 프로세스를 시작하기 전에 데이터베이스(재분배 할 노드 그룹의 테이블 스페이스)를 백업하십시오.

6. db2nodes.cfg 파일을 갱신하고 새 데이터 BCU 데이터베이스 파티션 스펙을 추가하여 DB2에 새 데이터 BCU를 정의하고 ADD NODE WITHOUT TABLESPACES 명령을 사용하여 DB2에 새 데이터베이스 파티션을 정의하십시오.

```

db2start nodenum x export DB2NODE=x
db2 add node without tablespaces
db2stop nodenum x

```

주: 데이터 BCU에 대해 첫 번째 논리적 포트가 아니면, 연속 논리적 포트에 대한 위의 명령 시퀀스 전후에 첫 번째 논리적 포트 번호 시작 및 중지를 실행하십시오.

7. 새로 정의된 데이터베이스 파티션에 대해 시스템 임시 테이블 스페이스 컨테이너를 정의하십시오.

```

ALTER TABLESPACE tablespace_name ADD container_information ON dbpartitionnums (x to y)

```

8. 데이터 BCU에 걸쳐 있는 데이터베이스 파티션 그룹에 새 논리 데이터베이스 파티션을 추가하십시오.

```

ALTER DATABASE PARTITION GROUP partition_group_name
ADD dbpartitionnums (x to y)
WITHOUT TABLESPACES

```

9. 새로 정의된 데이터베이스 파티션에 대해 영구 데이터 테이블 스페이스 컨테이너를 정의하십시오.

```

ALTER TABLESPACE tablespace_name ADD container_information ON dbpartitionnums (x to y)

```

10. 4단계에서 검색한 데이터베이스 구성 설정을 새 데이터베이스 파티션에 적용하십시오(또는 구성 지원의 새 DB2 9.5 단일 뷰를 사용하여 모든 데이터베이스 파티션에 대해 단일 UPDATE DB CFG 명령을 실행하십시오).

11. 재분배할 데이터베이스 파티션 그룹의 정의를 캡처한 후 이 그룹에 존재하는 복제된 MQT를 삭제하십시오.

```

db2look -d dbname -e -z schema -t replicated_MQT_table_names -o repMQTs.clp

```

12. 재분배할 데이터베이스 파티션 그룹에 존재하는 테이블에 기록 이벤트 모니터를 사용하지 않도록 설정하십시오.

```

SET EVENT MONITOR monitor_name STATE 0

```

13. REDISTRIBUTE 유틸리티를 실행하여 모든 데이터베이스 파티션 사이에 일정하게 재분배하십시오. 다음은 단순한 재분배 명령을 보여줍니다.

```
REDISTRIBUTE DATABASE PARTITION GROUP sampleNodegrp NOT ROLLFORWARD RECOVERABLE uniform;
```

사용자는 또한 REDISTRIBUTE 명령의 입력으로 테이블 목록을 지정하여 테이블이 처리될 순서를 지시할 것을 고려해야 합니다. REDISTRIBUTE 유틸리티는 데이터를 이동시킵니다(압축 및 간략화). 선택적으로, 인덱스가 재빌드되고 통계 프로파일이 정의된 경우 통계가 갱신됩니다. 따라서 이전 명령 대신 다음 스크립트를 실행할 수 있습니다.

```
REDISTRIBUTE DATABASE PARTITION GROUP sampleNodegrp NOT ROLLFORWARD RECOVERABLE uniform TABLE (tab1, tab2,...) FIRST;
```

## NOT ROLLFORWARD RECOVERABLE 옵션 사용 결과

REDISTRIBUTE DATABASE PARTITION GROUP 명령이 발행되고 NOT ROLLFORWARD RECOVERABLE 옵션이 지정되면, 이동된 각 행의 로그 레코드 쓰기를 최소화하는 최소 로깅 전략이 사용됩니다. 모든 데이터 이동을 완전히 로그하는 접근 방식이 대형 시스템에 가능하며, 비현실적인 활성 영구 로그 스페이스를 요구하며 일반적으로 성능이 더 저하된 성능 특성을 가지므로 재분배 조작의 사용 편리성에는 이러한 유형의 로깅이 중요합니다. 그러나 이러한 최소 로깅 모델의 결과로서, REDISTRIBUTE DATABASE PARTITION GROUP 명령이 복구 가능한 롤 포워드가 아님을 인식하는 것이 중요합니다. 이것은 재분배 조작을 통해 데이터베이스를 롤 포워드하는 모든 조작이 UNAVAILABLE 상태로 남아 있는 재분배 조작으로 모든 테이블을 처리하는 결과를 가져옴을 의미합니다. 그러한 테이블은 삭제만 가능하며, 이들 테이블의 데이터를 복구할 방법이 없음을 의미합니다. 그 이유는 복구 가능한 데이터베이스의 경우, REDISTRIBUTE DATABASE PARTITION GROUP 유틸리티가 NOT ROLLFORWARD RECOVERABLE 옵션으로 발행되면 사용하는 모든 테이블 스페이스를 BACKUP PENDING 상태로 만들어 정상적인 재분배 조작의 끝에서 사용자가 강제로 모든 재분배 테이블 스페이스를 백업하게 하기 때문입니다. 재분배 조작 이후의 백업 시, 사용자는 재분배 조작 자체를 통해 롤 포워드하지 말아야 합니다.

재분배 유틸리티에서 롤 포워드 복구 가능성이 없다는 점은 간과할 수 없는 중요한 문제입니다. 재분배에 의해 처리되는 테이블 스페이스를 사용자가 백업하는 재분배 주기의 종료 지점을 포함하여 재분배 조작이 실행되는 동안 데이터베이스의 테이블 갱신을 허용한 경우(재분배되는 데이터베이스 파티션 그룹 외부의 테이블인 경우에도) 데이터베이스 컨테이너 손상과 같은 심각한 장애의 경우 해당 갱신사항이 손실될 수 있습니다. 그러한 갱신이 유실되는 이유는 재분배 조작이 복구 가능한 롤 포워드가 아니기 때문입니다. 재분배 조작 이전의 백업에서 데이터베이스를 리스토어하는 것이 필요한 경우, 위에서 설명한 대로 재분배를 통해 롤 포워드하지 않는 재분배 조작 중 수행된 갱신을 재생하기 위해 로그를 롤 포워드하는 것이 가능하지 않고, 재분배된 테이블을 사용 불가능한 상태로 둡니다. 그러므로 이 상황에서 수행할 수 있는 유일한 것은 롤 포

워드 없이 재분배하기 전에 수행한 백업에서 데이터베이스를 리스토어하는 것입니다. 그러면 재분배 조작을 다시 수행할 수 있습니다. 그러나 원래의 재분배 조작 중 발생한 모든 갱신은 유실됩니다.

이러한 사항은 매우 중요합니다. 재분배 조작 중 갱신이 유실되지 않음을 보장하려면, 다음 중 하나가 참이어야 합니다.

- 영향을 받는 테이블 스페이스를 백업 중인 명령 완료 이후 기간을 포함하여 `REDISTRIBUTE DATABASE PARTITION GROUP` 명령의 조작 중 갱신하는 것을 피합니다.
- 재분배 조작 중 적용되는 갱신은 반복 가능한 소스에서 오는데, 이는 언제든지 다시 적용될 수 있음을 의미합니다. 예를 들어, 갱신 소스가 파일에 저장되는 데이터이며 일괄처리 중 갱신이 적용되는 경우, 데이터베이스 리스토어를 필요로 하는 실패에서조차도 갱신이 유실되지 않는 이유는 언제든지 다시 갱신을 적용할 수 있기 때문입니다.

재분배 조작 중 데이터베이스 갱신을 허용하는 경우, 필요하다면 사용자는 그러한 갱신이 적절한지 아니면 데이터베이스 리스토어 이후에 갱신을 반복할 수 있는지 여부를 결정해야 합니다.

**주:** `REDISTRIBUTE DATABASE PARTITION GROUP` 명령을 조작하는 동안 모든 실패가 이러한 문제점을 유발하지는 않습니다. 사실상 대부분은 문제점을 유발하지 않습니다. `REDISTRIBUTE DATABASE PARTITION GROUP` 명령은 완전하게 재시작할 수 있으며 이는 작업 중간에 유틸리티가 실패하는 경우 **CONTINUE** 또는 **ABORT** 옵션을 사용하여 간편하게 계속하거나 중단할 수 있음을 나타냅니다. 위에서 언급하는 실패는 재분배 조작 이전에 수행된 백업에서 리스토어하는 것이 필요한 실패입니다.

## 사용 시 참고사항

- **NOT ROLLFORWARD RECOVERABLE** 옵션을 지정하고 데이터베이스가 복구 가능한 데이터베이스인 경우 유틸리티가 테이블 스페이스에 처음 액세스할 때 **BACKUP PENDING** 상태가 됩니다. 해당 테이블 스페이스의 모든 테이블은 테이블 스페이스가 백업될 때까지 읽기 전용으로 됩니다.
- 재분배 조작이 실행 중이면, 처리되는 각 테이블의 시작 및 종료 시간과 같은 정보와 재분배 조작에 대한 일반 정보가 있는 이벤트 로그 파일을 생성합니다. 이 이벤트 로그 파일은 작성됩니다.
  - 서브디렉토리와 파일 이름에 대해 `database-name.database-partition-group-name.timestamp.log` 형식을 사용하는 Linux 및 UNIX 시스템의 `homeinst/sqllib/redis` 디렉토리.
  - 서브디렉토리와 파일 이름에 대해 `database-name.database-partition-group-name.timestamp.log` 형식을 사용하는 Windows 운영 체제의

**DB2INSTPROF#instance#redist** 디렉토리(여기서 **DB2INSTPROF**는 **DB2INSTPROF** 레지스트리 변수 값입니다).

- 시간소인 값은 명령이 발행된 시간입니다.

재분배 이벤트 로그에 대한 자세한 정보는 『재분배 이벤트 로그 파일』 주제를 참조하십시오.

- 유틸리티 프로그램은 처리 중 간헐적인 COMMIT를 수행합니다.
- 재분배가 일어난 테이블에 종속성을 갖는 모든 패키지는 무효화됩니다. 데이터베이스 파티션 그룹 재분배 조작이 완료된 후 이러한 패키지를 명시적으로 리바인드하는 것이 바람직합니다. 명시적으로 리바인드하면 유효하지 않은 패키지에 대한 첫 번째 SQL 요청을 실행할 때 초기 지연이 발생하지 않습니다. 재분배 메시지 파일에는 재분배가 일어난 모든 테이블 목록이 들어 있습니다.
- 디폴트로 재분배 유틸리티는 통계 프로파일이 있는 테이블의 통계를 갱신합니다. 통계 프로파일이 없는 테이블의 경우, 재분배 조작이 완료된 후에 RUNSTATS 명령을 발행하거나 db2Runstats API를 호출하여 해당 테이블의 테이블 및 인덱스 통계를 별도로 갱신하는 것이 좋습니다.
- DATA CAPTURE CHANGES로 정의된 복제된 테이블이나 복제된 구체화된 쿼리 테이블을 포함하는 데이터베이스 파티션 그룹은 재분배할 수 없습니다.
- 데이터베이스 파티션 그룹에 기존의 선언된 임시 테이블이나 작성된 임시 테이블이 있는 사용자 임시 테이블 스페이스가 있으면 재분배를 수행할 수 없습니다.
- INDEXING MODE와 같은 옵션은 해당 옵션이 적용되지 않는 테이블에서 경고 없이 무시됩니다. 예를 들어, INDEXING MODE는 인덱스가 없는 테이블에서 무시됩니다.
- 재분배 조작을 시작하기 전에 로드 보류 상태의 테이블이 없는지 확인하십시오. LOAD QUERY 명령을 사용하여 테이블 상태를 점검할 수 있습니다.
- 데이터베이스 파티션 서버 추가 요청이 보류 중 또는 진행 중인 경우에는 REDISTRIBUTE DATABASE PARTITION GROUP 명령이 실패할 수 있습니다(SQLSTATE 55071). 새 데이터베이스 파티션 서버가 온라인으로 인스턴스에 추가되고 모든 응용프로그램이 새 데이터베이스 파티션 서버를 인지하는 것이 아니라면 이 명령도 실패합니다(SQLSTATE 55077).

## 호환성

DB2 버전 9.5 이전의 XML 레코드 형식을 사용하는 XML 컬럼을 포함하는 테이블은 재분배할 수 없습니다. 테이블을 새 형식으로 이주하려면 ADMIN\_MOVE\_TABLE 스토어드 프로시저를 사용하십시오.

버전 8 이전 버전과의 호환성:



- **NODEGROUP** 키워드는 **DATABASE PARTITION GROUP**으로 대체될 수 있습니다.



---

## 제 99 장 REFRESH LDAP

LDAP(Lightweight Directory Access Protocol)의 정보가 변경될 때 로컬 시스템의 캐시를 갱신된 정보로 새로 고칩니다.

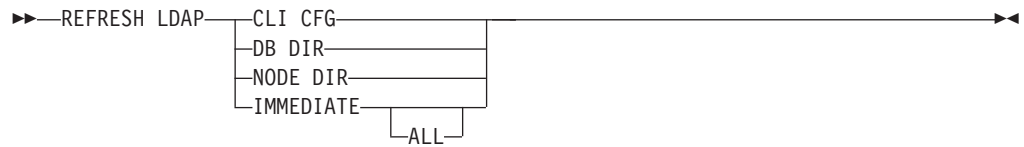
### 권한 부여

없음

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

#### CLI CFG

CLI 구성이 새로 고쳐지도록 지정합니다. 이 매개변수는 AIX 또는 Solaris 운영 체제에서 지원되지 않습니다.

#### DB DIR

데이터베이스 디렉토리가 새로 고쳐지도록 지정합니다.

#### NODE DIR

노드 디렉토리가 새로 고쳐지도록 지정합니다.

#### IMMEDIATE

로컬 데이터베이스 및 노드 디렉토리가 즉시 새로 고쳐지도록 지정합니다.

**ALL** LDAP 서버에 들어있는 모든 데이터베이스 및 노드 항목이 로컬 데이터베이스 및 노드 디렉토리에 추가되도록 지정합니다.

### 사용 시 참고사항

LDAP의 오브젝트가 새로 고침 중에 제거되는 경우 로컬 시스템의 대응하는 LDAP 항목도 제거됩니다. LDAP의 정보가 변경되는 경우 대응하는 LDAP 항목도 그에 따라서 수정됩니다. DB2CLI.INI 파일이 수동으로 갱신되는 경우 REFRESH LDAP CLI CFG 명령이 실행되어 현재 사용자에게 대한 캐시를 갱신해야 합니다.

REFRESH LDAP DB DIR 및 REFRESH LDAP NODE DIR 명령은 로컬 데이터베이스 또는 노드 디렉토리에서 발견되는 LDAP 데이터베이스 또는 노드 항목을 제거합니다. 데이터베이스 또는 노드 항목은 사용자가 데이터베이스에 연결하거나 LDAP에서 발견되는 인스턴스에 접속할 때 다시 로컬 데이터베이스 또는 노드 디렉토리에 추가되며, DB2LDAPCACHE는 설정되지 않거나 YES로 설정됩니다.

REFRESH LDAP IMMEDIATE 명령은 LDAP에서 발견되는 최신 정보를 사용하여 로컬 데이터베이스 및 노드 디렉토리의 항목을 갱신합니다. 이 갱신은 DB2LDAPCACHE가 사용 가능한지 여부와 상관없이 즉시 발생합니다. LDAP에서 시작한 데이터베이스 및 노드 항목만 갱신됩니다. 수동으로 추가된 항목은 변경되지 않고 그대로 있습니다.

REFRESH LDAP IMMEDIATE ALL 명령은 LDAP에서 발견되는 모든 정보로 로컬 데이터베이스 및 노드 디렉토리를 즉시 채웁니다. LDAP에서 발견되는 항목이 기존 로컬 항목과 일치하는 경우 명령은 해당 항목을 갱신합니다. 이 갱신은 로컬 항목이 LDAP에서 시작한 경우에만 발생합니다. 수동으로 추가된 항목은 변경되지 않고 그대로 있습니다. 이 갱신은 DB2LDAPCACHE의 사용 가능 여부와 상관없이 수행됩니다.

LDAP를 사용 안할 때 REFRESH LDAP IMMEDIATE 또는 REFRESH LDAP IMMEDIATE ALL을 수행하면 SQLCODE -3279(LDAP가 사용 불가능하기 때문에 시간은 성공적으로 완료되지 않음)가 생성됩니다.

## 제 100 장 REGISTER

네트워크 디렉토리 서버에 DB2 서버를 등록합니다.

### 권한 부여

없음

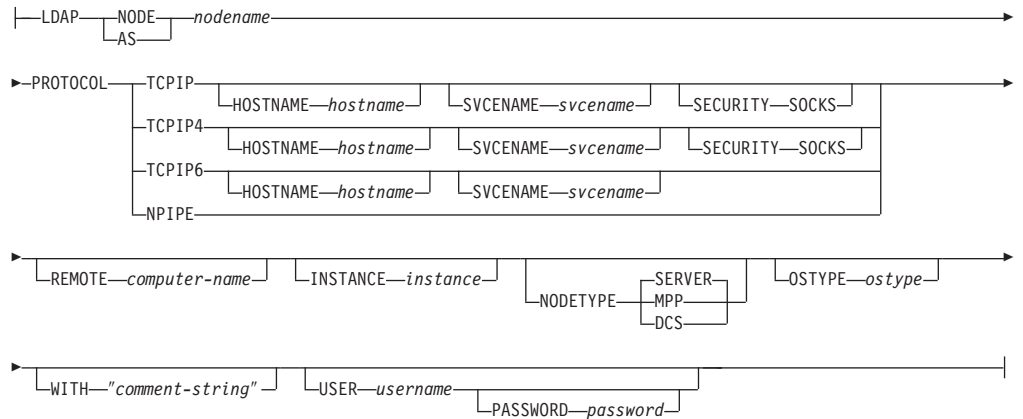
### 필수 연결

없음

### 명령 구문



### LDAP 경로:



### 명령 매개변수

**IN** DB2 서버를 등록할 네트워크 디렉토리 서버를 지정합니다. 올바른 값은 LDAP(Lightweight Directory Access Protocol) 디렉토리 서버의 경우 LDAP입니다.

### ADMIN

Administration Server 노드가 등록될 것임을 지정합니다.

### NODE | AS nodename

LDAP에서 DB2 서버를 표시할 단축 이름을 지정하십시오. 노드 항목은 이 노드 이름을 사용하여 LDAP에 카탈로그됩니다. 클라이언트는 이 노드 이름을 사

용하여 서버에 접속할 수 있습니다. 이 LDAP 노드 항목과 연관된 프로토콜은 PROTOCOL 매개변수를 통해 지정됩니다.

## PROTOCOL

LDAP 노드 항목과 연관된 프로토콜 유형을 지정합니다. 데이터베이스 서버가 둘 이상의 프로토콜 유형을 지원할 수 있으므로 이 값은 클라이언트 응용프로그램이 사용하는 프로토콜 유형을 지정합니다. DB2 서버는 프로토콜당 한 번 등록되어야 합니다. 올바른 값은 TCP/IP, TCP/IP4, TCP/IP6 및 NPIPE입니다. Windows Named Pipes를 사용하려면 NPIPE를 지정하십시오. NPIPE는 Windows 운영 체제에서만 지원됩니다.

## HOSTNAME *hostname*

TCP/IP 호스트 이름(또는 IP 주소)을 지정합니다. IP 주소는 TCP/IP 프로토콜을 사용할 때 IPv4 또는 IPv6 주소일 수 있습니다. TCP/IP4 프로토콜을 사용할 때 IP 주소는 IPv4 주소여야 합니다. TCP/IP6 프로토콜을 사용할 때는 IP 주소가 IPv6 주소여야 합니다.

## SVCENAME *svcename*

TCP/IP 서비스 이름 또는 포트 번호를 지정합니다.

## SECURITY SOCKS

TCP/IP SOCKS가 사용되도록 지정합니다. 이 매개변수는 IPv4에서만 지원됩니다. TCP/IP 프로토콜이 지정되면 사용되는 기본 프로토콜은 IPv4입니다.

## REMOTE *computer-name*

DB2 서버가 있는 머신의 컴퓨터 이름을 지정합니다. LDAP에 리모트 DB2 서버를 등록하는 경우에만 이 매개변수를 지정하십시오. 값은 LDAP에 서버 머신을 추가할 때 지정되는 값과 동일해야 합니다. Windows 운영 체제의 경우 이것은 컴퓨터 이름입니다. UNIX 기반 시스템의 경우 이것은 TCP/IP 호스트 이름입니다.

## INSTANCE *instance*

DB2 서버의 인스턴스 이름을 지정합니다. 인스턴스 이름은 리모트 인스턴스의 경우에(즉, REMOTE 매개변수에 대한 값이 지정되었을 때) 지정되어야 합니다.

## NODETYPE

데이터베이스 서버의 노드 유형을 지정합니다. 가능한 값은 다음과 같습니다.

### SERVER

DB2 Enterprise Server Edition의 SERVER 노드 유형을 지정하십시오. 이는 디폴트값입니다.

**MPP** DB2 Enterprise Server Edition - Extended(파티션된 데이터베이스) 서버에 대해 MPP 노드 유형을 지정하십시오.

DCS 호스트 데이터베이스 서버를 지정할 때 DCS 노드 유형을 지정하십시오.

#### **OSTYPE** *ostype*

서버 머신의 운영 체제 유형을 지정합니다. 올바른 값은 AIX, NT, HPUX, SUN, MVS, OS400, VM, VSE 및 LINUX입니다. 운영 체제 유형이 지정되지 않는 경우, 로컬 운영 체제 유형이 로컬 서버에 사용되며 리모트 서버의 경우 운영 체제 유형이 사용되지 않습니다.

#### **WITH** *"comment-string"*

DB2 서버를 설명합니다. 네트워크 디렉토리에 등록된 서버를 설명하는 데 도움이 되는 모든 주석을 입력할 수 있습니다. 최대 길이는 30문자입니다. 캐리지 리턴 또는 줄 바꾸기 문자는 허용되지 않습니다. 주석 텍스트는 큰따옴표로 묶어야 합니다.

### **사용 시 참고사항**

서버가 지원하는 각 프로토콜에 대해 DB2 서버를 한 번 등록하십시오.

서버를 디렉토리 서버에 발행하려면 각 DB2 서버 인스턴스에 대해 REGISTER 명령을 한 번 발행해야 합니다. 통신 매개변수가 재구성되거나 서버 네트워크 주소가 변경되는 경우 네트워크 디렉토리 서버의 DB2 서버를 갱신하십시오.

LDAP에서 DB2 서버를 갱신하려면 변경이 작성된 후 UPDATE LDAP NODE 명령을 사용하십시오.

DB2 서버를 로컬로 등록할 때 프로토콜 구성 매개변수가 지정되는 경우, 데이터베이스 관리 프로그램 구성 파일에서 지정된 값을 겹쳐씁니다.

REGISTER 명령이 LDAP에 로컬 DB2 인스턴스를 등록하는 데 사용되고 NODETYPE 및 OSTYPE 중 하나 또는 둘 다 지정되는 경우, 로컬 시스템에서 검색되는 값으로 바뀝니다. REGISTER 명령이 LDAP에서 리모트 DB2 인스턴스를 등록하는 데 사용되고 NODETYPE 및 OSTYPE 중 하나 또는 둘 다 지정되지 않는 경우, SERVER 및 Unknown의 디폴트값이 각각 사용됩니다.

REGISTER 명령이 LDAP에 리모트 DB2 서버를 등록하는 데 사용되는 경우, 리모트 서버의 컴퓨터 이름 및 인스턴스 이름이 리모트 서버의 통신 프로토콜과 함께 지정되어야 합니다.

호스트 데이터베이스 서버를 등록할 때 NODETYPE 매개변수에 대해 값 DCS를 지정해야 합니다.





---

## 제 101 장 REGISTER XMLSCHEMA

XML 스키마 저장소(XSR)에 XML 스키마를 등록합니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *dbadm*
- SQL 스키마가 없는 경우 IMPLICIT\_SCHEMA 데이터베이스 권한
- SQL 스키마가 있는 경우 CREATEIN 특권

### 필수 연결

데이터베이스

### 명령 구문

```
▶▶ REGISTER XMLSCHEMA schema-URI FROM content-URI
▶ [WITH properties-URI] [AS relational-identifier]
▶ [xml-document-subclause]
▶ [COMPLETE [WITH schema-properties-URI] [ENABLE DECOMPOSITION]]
```

#### xml-document-subclause:

```
| ( ( ADD document-URI FROM content-URI [WITH properties-URI] ) ) |
```

### 명령 매개변수

#### *schema-URI*

등록되는 XML 스키마의 URI(XML 인스턴스 문서에 참조된 대로)를 지정합니다.

#### FROM *content-URI*

XML 스키마 문서가 위치해 있는 URI를 지정합니다. 파일 스킴 URI로 지정된 로컬 파일만 지원됩니다.

**WITH** *properties-URI*

XML 스키마의 등록 정보 문서 URI를 지정합니다. 파일 스킴 URI로 지정된 로컬 파일만 지원됩니다.

**AS** *relational-identifier*

등록되는 XML 스키마를 참조하는 데 사용될 수 있는 이름을 지정합니다. 관계형 이름은 SQL 스키마와 XML 스키마 이름으로 구성되는 두 파트 SQL ID 로 지정될 수 있으며 형식은 SQLschema.name입니다. 스키마를 지정하지 않는 경우 디폴트 관계형 스키마(CURRENT SCHEMA 특수 레지스터에 정의된)가 사용됩니다. 이름을 제공하지 않으면 고유 값이 생성됩니다.

**COMPLETE**

추가할 XML 스키마 문서가 더 이상 없음을 표시합니다. 지정하는 경우, 스키마의 유효성이 확인되고 오류가 발견되지 않으면 사용 가능한 것으로 표시됩니다.

**WITH** *schema-properties-URI*

XML 스키마의 등록 정보 문서 URI를 지정합니다. 파일 스킴 URI로 지정된 로컬 파일만 지원됩니다.

**ENABLE DECOMPOSITION**

스키마가 XML 문서 분석에 사용됨을 지정합니다.

**ADD** *document-URI*

문서가 다른 XML 문서에서 참조되어 스키마에 추가될 XML 스키마 문서의 URI를 지정합니다.

**FROM** *content-URI*

XML 스키마 문서가 위치해 있는 URI를 지정합니다. 파일 스킴 URI로 지정된 로컬 파일만 지원됩니다.

**WITH** *properties-URI*

XML 스키마의 등록 정보 문서 URI를 지정합니다. 파일 스킴 URI로 지정된 로컬 파일만 지원됩니다.

**예:**

```
REGISTER XMLSCHEMA 'http://myPOschema/PO.xsd'
FROM 'file:///c:/TEMP/PO.xsd'
WITH 'file:///c:/TEMP/schemaProp.xml'
AS user1.POschema
```

**사용 시 참고사항**

- XML 스키마 문서를 참조하고 유효성 확인 및 어노테이션에 사용 가능하기 전에, 먼저 XSR에 등록해야 합니다. 이 명령은 기본 XML 스키마 문서를 등록하여 XML 스키마 등록 프로세스의 첫 번째 단계를 수행합니다. XML 스키마 등록의 최종 단계에서는 XML 스키마에 대해 COMPLETE XMLSCHEMA 명령이 성공적으로 실행

행되어야 합니다. 또는 포함할 다른 XML 스키마 문서가 없는 경우 COMPLETE 키워드와 함께 REGISTER XMLSCHEMA 명령을 실행하여 한 단계에서 등록을 완료하십시오.

- 데이터베이스에서 XML 스키마를 등록할 때 XML 스키마의 크기에 따라 더 큰 응용프로그램 힙(APPLHEAPSZ)이 필요할 수 있습니다. 권장 크기는 1024이지만 큰 스키마에는 추가 메모리가 필요합니다.



---

## 제 102 장 REGISTER XSROBJECT

데이터베이스 카탈로그에 XML 오브젝트를 등록합니다. 지원되는 오브젝트는 DTD와 외부 엔티티입니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *dbadm*
- SQL 스키마가 없는 경우 IMPLICIT\_SCHEMA 데이터베이스 권한
- SQL 스키마가 있는 경우 CREATEIN 특권

### 필수 연결

데이터베이스

### 명령 구문

```
▶▶ REGISTER XSROBJECT system-ID [PUBLIC public-ID] FROM content-URI
[AS relational-identifier] [DTD | EXTERNAL ENTITY]
```

### 명령 매개변수

*system-ID*

XML 오브젝트 선언에 지정된 시스템 ID를 지정합니다.

**PUBLIC** *public-ID*

XML 오브젝트 선언의 선택적 PUBLIC ID를 지정합니다.

**FROM** *content-URI*

XML 스키마 문서의 콘텐츠가 있는 URI를 지정합니다. 파일 스킴 URI로 지정된 로컬 파일만 지원됩니다.

**AS** *relational-identifier*

등록되는 XML 오브젝트를 참조하는 데 사용될 수 있는 이름을 지정합니다. 관계형 이름은 관계형 스키마와 이름이 마침표로 분리되는 두 파트로 구성된 SQL ID로 지정할 수 있습니다(예: "JOHNDOE.EMPLOYEEDTD"). 관계형 스키마가 지정되지 않으면 특수 레지스터 CURRENT SCHEMA에 정의된 디폴트 관계형 스키마가 사용됩니다. 이름이 지정되지 않은 경우 자동으로 이름이 생성됩니다.

**DTD** 등록되는 오브젝트가 데이터 유형 정의 문서(DTD)인 것으로 지정합니다.

## EXTERNAL ENTITY

등록되는 오브젝트가 외부 엔티티인 것으로 지정합니다.

### 예:

1. 외부 엔티티를 참조하는 이 샘플 XML 문서의 경우:

```
<?xml version="1.0" standalone="no" ?>
<!DOCTYPE copyright [
  <!ELEMENT copyright (#PCDATA)>
]
<copyright>c</copyright>
```

이 문서를 XML 컬럼에 삽입하려면 외부 엔티티가 등록되어야 합니다. 다음 명령은 엔티티 콘텐츠가 로컬로 C:\WTEMP에 저장되는 엔티티를 등록합니다.

```
REGISTER XSROBJECT 'http://www.xmlwriter.net/copyright.xml'
FROM 'c:\wtemp\copyright.xml' EXTERNAL ENTITY
```

2. DTD를 참조하는 이 XML 문서 조각의 경우:

```
<!--inform the XML processor
that an external DTD is referenced-->
<?xml version="1.0" standalone="no" ?>

<!--define the location of the
external DTD using a relative URL address-->
<!DOCTYPE document SYSTEM "http://www.xmlwriter.net/subjects.dtd">

<document>
  <title>Subjects available in Mechanical Engineering.</title>
  <subjectID>2.303</subjectID>
  <subjectname>Fluid Mechanics</subjectname>
  ...
```

이 문서를 XML 컬럼에 삽입하려면 DTD가 등록되어야 합니다. 다음 명령은 DTD 정의가 C:\WTEMP에 로컬로 저장되고 DTD와 연관될 관계형 ID가 "TEST.SUBJECTS"인 경우 DTD를 등록합니다.

```
REGISTER XSROBJECT 'http://www.xmlwriter.net/subjects.dtd'
FROM 'file:///c:/temp/subjects.dtd' AS TEST.SUBJECTS DTD
```

3. 공용 외부 엔티티를 참조하는 이 샘플 XML 문서의 경우:

```
<?xml version="1.0" standalone="no" ?>
<!DOCTYPE copyright [
  <!ELEMENT copyright (#PCDATA)>
]
<copyright>c</copyright>
```

이 문서를 XML 컬럼에 삽입하려면 공용 외부 엔티티가 등록되어야 합니다. 다음 명령은 엔티티 콘텐츠가 로컬로 C:\WTEMP에 저장되는 엔티티를 등록합니다.

```
REGISTER XSROBJECT 'http://www.w3.org/xmlspec/copyright.xml'
PUBLIC '-//W3C//TEXT copyright//EN' FROM 'file:///c:/temp/copyright.xml'
EXTERNAL ENTITY
```

---

## 제 103 장 REORG INDEXES/TABLE

### 인덱스 또는 테이블 재구성

인덱스 데이터를 조각나지 않은 실제로 연속된 페이지로 재빌드하여 테이블에 정의된 모든 인덱스를 재구성할 수 있습니다. 또는 범위 파티션 테이블에서 특정 인덱스를 재구성할 수 있는 옵션도 제공됩니다.

인덱스 절의 CLEANUP ONLY 옵션을 지정하면 인덱스를 재빌드하지 않고 정리를 수행합니다. 이 명령은 선언된 임시 테이블 또는 작성된 임시 테이블의 인덱스에 대해서만 사용할 수 없습니다(SQLSTATE 42995).

테이블 옵션은 행을 재구성하여 조각난 데이터를 제거하고 정보를 간략화함으로써 테이블을 재구성합니다.

### 범위

이 명령은 데이터베이스 파티션 그룹에 있는 모든 데이터베이스 파티션에 영향을 줍니다.

### 권한 부여

다음 중 하나가 필요합니다.

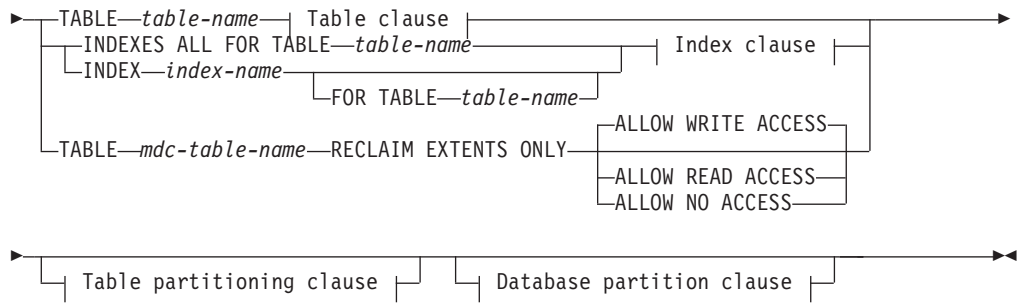
- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*
- *sqladm*
- 테이블에 대한 CONTROL 특권

### 필수 연결

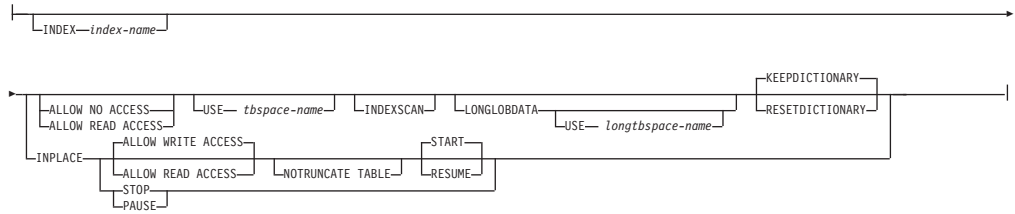
데이터베이스

### 명령 구문

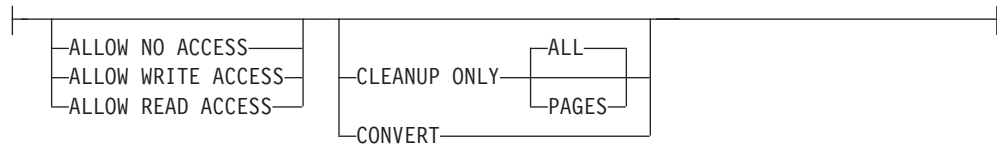
▶—REORG—————▶



**Table clause:**



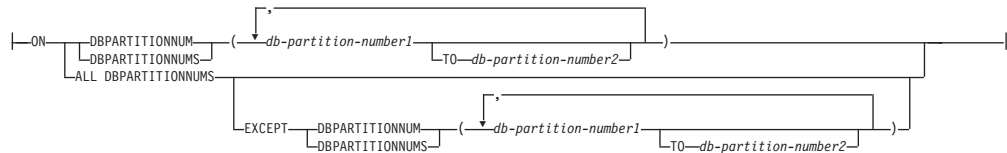
**Index clause:**



**테이블 파티션 절:**



**Database partition clause:**



**명령 매개변수**

**INDEXES ALL FOR TABLE *table-name***

인덱스가 재구성되는 테이블을 지정합니다. 테이블은 로컬 또는 리모트 데이터 베이스에 있을 수 있습니다.

**INDEX *index-name***

파티션된 테이블에 재구성될 개별 인덱스를 지정합니다. 개별 인덱스의 재구성은 파티션된 테이블의 파티션되지 않은 인덱스에만 지원됩니다. 블록 인덱스에 대해서는 이 매개변수가 지원되지 않습니다.



### FOR TABLE *table-name*

파티션된 테이블에서 재구성되는 개별 인덱스의 테이블 이름 위치를 지정합니다. 인덱스 이름이 데이터베이스에서 고유하다고 고려할 때 이 매개변수는 선택적입니다.

REORG INDEXES의 경우, ON DATA PARTITION 옵션이 지정되면 액세스 절은 이름 지정된 파티션에만 적용됩니다. 지정된 파티션의 인덱스가 재구성되는 동안 사용자는 나머지 테이블에서 읽고 쓸 수 있습니다. 이 상황은 디폴트 액세스 등급에도 적용됩니다.

#### ALLOW NO ACCESS

REORG INDEX 및 REORG INDEXES의 경우, 인덱스가 재구성되는 동안 다른 사용자가 테이블 또는 데이터 파티션에 액세스할 수 없도록 지정합니다.

#### ALLOW READ ACCESS

REORG INDEX 및 REORG INDEXES의 경우, 인덱스가 재구성되는 동안 다른 사용자가 테이블 또는 데이터 파티션에 읽기 전용 액세스할 수 있도록 지정합니다. CLEANUP ONLY 옵션이나 ON DATA PARTITION 옵션이 지정되지 않은 경우 파티션된 테이블의 REORG INDEXES에 대해서는 이 액세스 레벨이 지원되지 않습니다.

#### ALLOW WRITE ACCESS

REORG INDEX 및 REORG INDEXES의 경우, 인덱스가 재구성되는 동안 다른 사용자가 테이블 또는 데이터 파티션에서 읽고 쓸 수 있도록 지정합니다.

CLEANUP ONLY 옵션이 지정되지 않은 경우 다차원적으로 클러스터된(MDC) 테이블이나 확장 인덱스에 대해서는 이 액세스 레벨이 지원되지 않습니다. 또한, CLEANUP ONLY 옵션이나 ON DATA PARTITION 옵션이 지정되지 않은 경우 REORG INDEXES 명령이 사용되면 파티션된 테이블에 대해 이 액세스 레벨이 지원되지 않습니다.

ACCESS 모드가 지정되지 않으면, 다음 방법으로 테이블이나 데이터 파티션에 대해 선택됩니다.

표 30. 인덱스 절에 지정된 명령, 테이블 유형 및 추가 매개변수를 기본으로 선택된 디폴트 테이블 액세스

명령	테이블 유형	테이블 파티션 절	인덱스 절에 지정된 추가 매개변수	디폴트 액세스 모드
REORG INDEXES	파티션되지 않은 테이블	해당되지 않음	임의	ALLOW READ ACCESS
REORG INDEX	파티션된 테이블	해당되지 않음	임의	ALLOW READ ACCESS
REORG INDEXES	파티션된 테이블	없음	지정되지 않음	ALLOW NO ACCESS

표 30. 인덱스 절에 지정된 명령, 테이블 유형 및 추가 매개변수를 기본으로 선택된 디폴트 테이블 액세스 (계속)

명령	테이블 유형	테이블 파티션 절	인덱스 절에 지정된 추가 매개변수	디폴트 액세스 모드
REORG INDEXES	파티션된 테이블	ON DATA PARTITION	지정되지 않음	ALLOW READ ACCESS
REORG INDEXES	파티션된 테이블	ON DATA PARTITION 절이 있거나 없음	CLEANUP ONLY가 지정됨	ALLOW READ ACCESS

### CLEANUP ONLY

CLEANUP ONLY가 요청된 경우 전체 재구성 대신 정리가 수행됩니다. 인덱스는 재빌드되지 않으며 제거된 페이지는 이 테이블에만 정의된 인덱스를 통해 재사용할 수 있습니다.

CLEANUP ONLY PAGES 옵션은 커밋된 의사(pseudo) 공백 페이지를 검색하여 제거합니다. 커밋된 의사(pseudo) 공백 페이지는 페이지에 있는 모든 키가 삭제된 것으로 표시되고 삭제가 모두 커밋된 것으로 인식되는 페이지입니다. 인덱스에 있는 의사(pseudo) 공백 페이지 수는 RUNSTATS를 실행하여 SYSCAT.INDEXES의 NUM EMPTY LEAFS 컬럼을 보고 판별할 수 있습니다. PAGES 옵션은 NUM EMPTY LEAFS가 커밋된 것으로 판별되면 이를 정리합니다.

CLEANUP ONLY ALL 옵션은 의사(pseudo) 공백이 아닌 페이지에서 커밋된 의사 삭제된 키를 제거할 뿐 아니라 커밋된 의사 공백 페이지를 제거합니다. 이 옵션은 또한 인접 리프 페이지를 병합하는데 이렇게 할 경우 최소한 PCTFREE 여유 공간이 있는 병합된 리프 페이지로 나타납니다. 여기서 PCTFREE는 인덱스 작성시간에 인덱스에 대해 정의된 여유 공간 비율입니다. 디폴트 PCTFREE는 10퍼센트입니다. 두 개의 페이지가 병합될 수 있는 경우 페이지 중 하나가 제거됩니다. 의사(pseudo) 공백 페이지에 있는 경우를 제외하고 인덱스에 있는 의사(pseudo) 삭제된 키 수는 RUNSTATS를 실행한 후 SYSCAT.INDEXES에서 NUMRIDS DELETED를 삭제하여 판별할 수 있습니다. ALL 옵션은 NUMRIDS DELETED 및 NUM EMPTY LEAFS가 커밋된 것으로 판별되면 이를 정리합니다.

**ALL** 커밋된 의사(pseudo) 삭제된 키 및 커밋된 의사(pseudo) 공백 페이지를 제거하여 인덱스가 정리되도록 지정합니다.

### PAGES

커밋된 의사(pseudo) 공백 페이지가 인덱스 트리에서 제거되도록 지정합니다. 의사(pseudo) 공백이 아닌 페이지에 있는 의사(pseudo) 삭제된 키는 정리하지 않습니다. 의사(pseudo) 공백 리프 페이지만 점검하므로 대부분의 경우에 ALL 옵션을 사용하는 것보다 훨씬 더 신속합니다.

## CONVERT

유형 1 인덱스를 유형 2 인덱스로 변환합니다. 인덱스가 이미 유형 2 인 경우 이 옵션은 어떤 영향도 미치지 않습니다.

버전 9.7에서, 유형 1 인덱스는 중지되며 작성되는 모든 인덱스는 유형 2 인덱스입니다. 결과적으로, CONVERT 옵션은 사용되지 않습니다.

버전 8 이전에 작성된 모든 인덱스는 유형 1 인덱스입니다. 버전 9.7 이전에, 이미 유형 1 인덱스를 가지고 있는 테이블에서 인덱스를 작성할 경우를 제외하고는 버전 8 이후에 작성되는 모든 인덱스는 유형 2 인덱스입니다. 이 경우, 새로운 인덱스는 유형 1입니다. 작성되는 모든 인덱스는 유형 2이므로 이는 버전 9.7의 경우가 아닙니다.

인덱스가 재구성되는 동안 다른 트랜잭션이 테이블에 대해 읽기 전용 또는 읽기 쓰기 액세스할 수 있도록 하려면 ALLOW READ ACCESS 또는 ALLOW WRITE ACCESS 옵션을 사용하십시오. ALLOW READ ACCESS 및 ALLOW WRITE ACCESS 옵션을 사용하면 테이블에 액세스할 수 있는 반면, 재구성된 인덱스 사본을 사용할 수 있는 동안에는 테이블에 액세스할 수 없습니다.

## TABLE *mdc-table-name* RECLAIM EXTENTS ONLY

사용 중이지 않은 Extent를 재개하도록 재구성하기 위해 다차원적으로 클러스터된(MDC) 테이블을 지정합니다. *schema.table-name* 형식의 이름 또는 별명을 사용할 수 있습니다. *schema*는 테이블이 작성된 사용자 이름입니다. 스키마 이름을 생략하면 디폴트 스키마가 간주됩니다.

REORG TABLE RECLAIM EXTENTS ONLY의 경우, ON DATA PARTITION 옵션이 지정되면 액세스 절은 이름 지정된 파티션에만 적용됩니다. 지정된 파티션의 Extent가 재개되는 동안 사용자는 나머지 테이블에서 읽고 쓸 수 있습니다. 이 상황은 디폴트 액세스 등급에도 적용됩니다.

### ALLOW NO ACCESS

REORG TABLE RECLAIM EXTENTS ONLY의 경우, Extent가 재개되는 동안 다른 사용자는 해당 테이블을 액세스할 수 없습니다.

### ALLOW READ ACCESS

REORG TABLE RECLAIM EXTENTS ONLY의 경우, Extent가 재개되는 동안 다른 사용자는 테이블에 대한 읽기 전용 액세스 권한을 가질 수 있음을 지정합니다.

### ALLOW WRITE ACCESS

REORG TABLE RECLAIM EXTENTS ONLY의 경우, Extent가 재개되는 동안 다른 사용자는 테이블을 읽고 쓸 수 있음을 지정합니다.

## **TABLE** *table-name*

재구성할 테이블을 지정합니다. 테이블은 로컬 또는 리모트 데이터베이스에 있을 수 있습니다. *schema.table-name* 형식의 이름 또는 별명을 사용할 수 있습니다. *schema*는 테이블이 작성된 사용자 이름입니다. 스키마 이름을 생략하면 디폴트 스키마가 간주됩니다.

유형이 지정된 테이블의 경우, 지정된 테이블 명은 계층 구조의 루트 테이블 이름이어야 합니다.

다차원적으로 클러스터된(MDC) 테이블의 재구성에 대해서는 인덱스를 지정할 수 없습니다. MDC 테이블에는 테이블의 설정된 재구성을 사용할 수 없습니다.

## **INDEX** *index-name*

테이블을 재구성할 때 사용할 인덱스를 지정합니다. *schema.index-name* 형식으로 완전한 이름을 지정하지 않으면 디폴트 스키마가 간주됩니다. *schema*는 인덱스가 작성된 사용자 이름입니다. 데이터베이스 관리 프로그램은 인덱스를 사용하여 재구성 중인 테이블의 레코드를 실제로 다시 정렬합니다.

설정된 테이블 재구성의 경우, 테이블에 클러스터링 인덱스가 정의되어 있고 인덱스가 지정되어 있는 경우, 이는 클러스터링 인덱스여야 합니다. **Inplace** 옵션이 지정되어 있지 않으면 지정된 인덱스가 사용됩니다. 인덱스의 이름을 지정하지 않으면 레코드는 순서에 관계없이 재구성됩니다. 테이블에 클러스터링 인덱스가 정의되어 있지만 인덱스가 지정되어 있지 않으면 클러스터링 인덱스를 사용하여 테이블을 클러스터합니다. MDC 테이블을 재구성하고 있으면 인덱스를 지정할 수 없습니다.

## **ALLOW NO ACCESS**

테이블이 재구성되는 동안 다른 사용자가 테이블에 액세스할 수 없도록 지정합니다. Table partitioning 절을 사용하지 않고 파티션된 테이블을 재구성하는 경우 이 값이 디폴트입니다.

## **ALLOW READ ACCESS**

재구성 중 테이블에 대한 읽기 액세스만 허용합니다. 이 값은 파티션되지 않은 테이블이나 테이블 파티셔닝 절을 사용하는 파티션된 테이블의 디폴트입니다.

**ON DATA PARTITION** 옵션이 범위 파티션된 테이블의 **REORG TABLE** 용으로 지정되면, 액세스 절은 이름지정된 파티션에만 적용됩니다. 테이블의 모든 인덱스가 파티션되면, 파티션이 재구성되고 인덱스가 재빌드되는 동안 사용자는 나머지 테이블에서 읽고 쓸 수 있습니다. 그러나 파티션되지 않은 인덱스가 테이블에 정의되면, 전체 테이블의 액세스가 **ALLOW NO ACCESS**로 확대됩니다. 이 상황에서, 파티션되지 않은 인덱스는 지정된 파티션과 함께 재빌드됩니다.

## **INPLACE**

사용자 액세스를 허용하면서 테이블을 재구성합니다.

테이블 재구성은 유형 2 인덱스를 포함하지만 확장 인덱스는 없고 테이블의 XML 컬럼에 대해 정의된 인덱스가 없는 파티션되지 않은 테이블 및 비MDC 테이블에서만 허용됩니다. 위치 테이블에서 재구성은 최소한 3페이지 크기의 테이블에서만 수행될 수 있습니다.

설정된 테이블 재구성은 비동기적으로 발생하며 즉시 유효하지 않을 수 있습니다.

## **ALLOW READ ACCESS**

재구성 중 테이블에 대한 읽기 액세스만 허용합니다.

## **ALLOW WRITE ACCESS**

재구성 중 테이블에 대한 쓰기 액세스를 허용합니다. 이 설정이 디폴트 동작입니다.

## **NOTRUNCATE TABLE**

설정된 재구성 후 테이블을 절단하지 마십시오. 절단하는 동안 테이블은 S-잠금됩니다.

## **START**

설정된 REORG 처리를 시작합니다. 디폴트값이므로 이 키워드는 선택적입니다.

**STOP** 현재 위치에서 설정된 REORG 처리를 중지합니다.

## **PAUSE**

잠시 동안 설정된 REORG를 일시중단하거나 일시정지합니다.

## **RESUME**

이전이 일시정지된 설정된 재구성을 계속하거나 재개합니다. 온라인 재구성이 재개되고 재구성이 일시정지되었을 때와 동일한 옵션을 원할 경우, 재개되는 동안 이들 옵션을 다시 지정해야 합니다.

## **USE** *tblspace-name*

재구성 중인 테이블의 임시 사본을 저장할 시스템 임시 테이블 스페이스 이름을 지정합니다. 테이블 스페이스 이름을 제공하지 않으면 데이터베이스 관리 프로그램은 재구성 중인 테이블이 들어 있는 테이블 스페이스에 테이블의 작업 사본을 저장합니다.

8KB, 16KB 또는 32KB 테이블 오브젝트에서, 사용자가 지정하는 시스템 임시 테이블 스페이스의 페이지 크기가 테이블 데이터가 있는 테이블 스페이스의 페이지 크기와 일치하지 않을 경우 DB2 데이터베이스

스 제품은 올바른 크기의 LONG/LOB 오브젝트의 임시 테이블 스페이스를 찾으려고 시도합니다. 재구성을 성공하려면 이러한 테이블 스페이스가 존재해야 합니다.

동일한 페이지 크기의 임시 테이블 스페이스가 두 개 있으며 이들 중 하나를 USE절에 지정할 때 재구성 중인 테이블에 인덱스가 있으면 테이블 스페이스는 라운드로빈(round robin) 방식으로 사용됩니다. 두 개의 테이블 스페이스 temp space1 및 temp space2가 있으며 둘 다 동일한 페이지 크기이고 USE 옵션을 사용하는 REORG 명령에 temp space1을 지정한다고 하십시오. 처음 REORG를 수행하면 temp space1이 사용됩니다. 두 번째에는 temp space2가 사용됩니다. 세 번째에는 temp space1가 사용되고 계속 이와 같이 됩니다. 이렇게 하지 않으려면 임시 테이블 스페이스 중 하나를 삭제해야 합니다.

파티션된 테이블의 경우, 테이블 스페이스는 테이블에 있는 데이터 파티션의 재구성에 대한 임시 스토리지로 사용됩니다. 전체 파티션된 테이블을 재구성하면 한 번에 한 데이터 파티션이 재구성됩니다. 필요한 스페이스의 크기는 전체 테이블이 아니라 테이블에서 가장 큰 데이터 파티션 크기와 같습니다.

파티션된 테이블에 테이블 스페이스 이름을 제공하지 않으면 해당 데이터 파티션의 임시 스토리지에 각 데이터 파티션이 위치한 테이블 스페이스를 사용합니다. 각 데이터 파티션의 테이블 스페이스에는 데이터 베이스 파티션의 사본을 보유하기에 충분한 여유 공간이 있어야 합니다.

## INDEXSCAN

클러스터링 REORG의 경우 테이블 레코드를 다시 정렬하기 위해 인덱스 스캔을 사용합니다. 인덱스를 통해 테이블에 액세스하여 테이블 행을 재구성합니다. 디폴트 방법은 필요에 따라 임시 테이블 스페이스를 사용하여 테이블을 스캔한 후 그 결과를 정렬하여 테이블을 재구성하는 것입니다. 인덱스 키가 정렬 순서로 되었다더라도 스캔하고 정렬하는 것이 인덱스에서 먼저 행 ID를 읽고 행을 폐치하는 것보다 일반적으로 더 빠릅니다.

## LONGLOBDATA

Long 필드 및 LOB 데이터가 재구성됩니다.

테이블에 Long 또는 LOB 컬럼이 들어 있더라도 이 옵션은 필수적이지 않습니다. 디폴트값은 이들 오브젝트를 재구성하지 않는 것입니다. 이는 시간이 많이 들고 클러스터링을 향상시키지 않기 때문입니다. 그러나 XML 컬럼이 있는 테이블에서 LONGLOBDATA 옵션으로 재구성을 실행하면 미사용 스페이스를 재개하므로 XML 스토리지 오브젝트 크기를 감축합니다.

기존 LOB 데이터를 인라인된 LOB 데이터로 변환할 때 이 매개변수가 필요합니다.

**USE longtbspace-name**

이는 선택적 매개변수로서 Long 데이터를 재빌드할 때 사용될 임시 테이블 스페이스의 이름을 지정하는 데 사용될 수 있습니다. 테이블 오브젝트 또는 Long 오브젝트에 대해 임시 테이블 스페이스가 지정되지 않은 경우 오브젝트는 현재 있는 테이블 스페이스에서 구성됩니다. 테이블에 대해 임시 테이블 스페이스가 지정되어 있지만 이 매개변수가 지정되지 않은 경우, 페이지 크기가 서로 다르지 않으면 기본 재구성 데이터에 사용된 테이블 스페이스가 사용됩니다. 이러한 경우 DB2 데이터베이스 시스템은 Long 오브젝트를 작성할 적당한 페이지 크기의 임시 컨테이너를 선택하려고 시도합니다.

USE longtbspace-name이 지정된 경우, USE tbspace-name도 지정되어야 합니다. 그렇지 않으면 longtbspace-name 인수가 무시됩니다.

**KEEPDICTIONARY**

테이블에 대한 COMPRESS 속성이 YES 이고 테이블에 압축 사전이 들어 있으면 새 사전이 빌드되지 않습니다. 재구성 중 처리된 모든 행은 기존 사전을 사용하여 압축되기 쉽습니다. COMPRESS 속성이 YES이며 테이블의 압축 사전이 없는 경우, 테이블이 특정 크기(대략 1 - 2MB)이며 충분한 데이터가 이 테이블에 있는 경우 이 시나리오에서 사전만 작성됩니다(테이블은 압축됨). 대신 명시적으로 REORG RESETDICTIONARY를 언급하면, 최소한 1행이 테이블에 있는 한 사전이 빌드됩니다. 테이블에 대한 COMPRESS 속성이 NO이고 테이블에 압축 사전이 들어 있으면 REORG 처리는 이 사전을 보존하고 새로 재구성된 테이블의 모든 행은 압축되지 않은 형식이 됩니다. 기본 테이블 행에 저장되지 않은 LOB 데이터와 같은 일부 데이터는 압축할 수 없습니다.

LONGLOBDATA 옵션이 지정되지 않으면, 테이블 행 데이터만이 재구성됩니다. LONGLOBDATA 옵션이 지정되지 않으면 다음 표는 REORG 명령에서 KEEPDICTIONARY 구문의 동작을 설명합니다.

표 31. REORG KEEPDICTIONARY

압축	사전이 존재함	결과, 결론
Y	Y	사전 보존; 행 압축
Y	N	사전 빌드; 행 압축
N	Y	사전 보존; 모든 행 압축 해제
N	N	영향을 미치지 않음, 모든 행을 압축 해제합니다.

LONGLOBDATA 옵션이 지정되면 다음 표는 REORG 명령에서 KEEPDICTIONARY 구문의 동작을 설명합니다.

표 32. LONGLOBDATA 옵션이 지정될 때 REORG KEEPDICTIONARY

압축	테이블 행 데이터 사전이 존재합니다.	XML 스토리지 오브젝트 사전이 존재합니다. <sup>1</sup>	압축 사전	데이터 압축
Y	Y	Y	사전을 보존합니다.	기존 데이터가 압축됩니다. 새 데이터가 압축됩니다.
Y	Y	N	테이블 행 사전을 보존하며 XML 스토리지 오브젝트 사전을 작성합니다.	기존 데이터가 압축됩니다. 새 데이터가 압축됩니다.
Y	N	Y	테이블 행 사전을 작성하며 XML 사전을 보존합니다.	기존 데이터가 압축됩니다. 새 데이터가 압축됩니다.
Y	N	N	테이블 행 및 XML 사전을 작성합니다.	기존 데이터가 압축됩니다. 새 데이터가 압축됩니다.
N	Y	Y	테이블 행 및 XML 사전을 보존합니다.	테이블 데이터가 압축 해제됩니다. 새 데이터가 압축되지 않습니다.
N	Y	N	테이블 행 사전을 보존합니다.	테이블 데이터가 압축 해제됩니다. 새 데이터가 압축되지 않습니다.
N	N	Y	XML 사전을 보존합니다.	테이블 데이터가 압축 해제됩니다. 새 데이터가 압축되지 않습니다.
N	N	N	영향을 미치지 않습니다.	테이블 데이터가 압축 해제됩니다. 새 데이터가 압축되지 않습니다.

**메모:**

1. XML 컬럼이 DB2 V9.7 이상에서 테이블에 추가된 경우나, 테이블이 ONLINE\_TABLE\_MOVE 스토어드 프로시저를 사용하여 이주된 경우에만 XML 스토리지 오브젝트의 압축 사전이 작성될 수 있습니다.

모든 테이블을 다시 초기화하거나 절단할 경우(예: 바꾸기 조작) 테이블의 압축 속성이 NO이면 사전을 버립니다(사전이 있을 경우). 반대로 사전이 있으나 테이블의 압축 속성이 YES이면 절단 조작은 사전을 저장하고 버리지 않습니다. 복구 및 추후 지원을 목적으로 데이터 캡처 변경사항(즉, 복제)과 함께 사전 전체가 로그됩니다.

**RESETDICTIONARY**

테이블에 대한 COMPRESS 속성이 YES이면 새 행 압축 사전이 빌드됩니다. 재구성 중 처리된 모든 행은 이 새 사전을 사용하여 압축되기 쉽습니다. 이 사전은 이전 사전을 바꿉니다. 테이블에 대한 COMPRESS 속성이 NO이고 테이블에 기존 압축 사전이 들어 있으면 REORG 처리는 이 사전을 제거하고 새로 재구성된 테이블의 모든 행은 압축되지 않은



형식이 됩니다. 기본 테이블 행에 저장되지 않은 LOB 데이터와 같은 일부 데이터는 압축할 수 없습니다.

LONGLOBDATA 옵션이 지정되지 않으면, 테이블 행 데이터만이 재구성됩니다. LONGLOBDATA 옵션이 지정되지 않으면 다음 표에서는 REORG 명령에서 RESETDICTIONARY 구문의 동작을 설명합니다.

표 33. REORG RESETDICTIONARY

압축	사전이 존재함	결과, 결론
Y	Y	새 사전*을 빌드합니다. 행을 압축합니다. DATA CAPTURE CHANGES 옵션이 CREATE TABLE 또는 ALTER TABLE문에 지정되면, 현재 사전이 보존됩니다(실행기록 압축 사전으로 참조됨).
Y	N	새 사전을 빌드합니다. 행을 압축합니다.
N	Y	사전을 제거합니다. 모든 행을 압축 해제합니다. DATA CAPTURE NONE 옵션이 CREATE TABLE 또는 ALTER TABLE문에 지정되면, 지정된 테이블의 실행기록 압축 사전도 제거됩니다.
N	N	영향을 미치지 않음, 모든 행을 압축 해제합니다.

\* - 사전이 존재하고 압축 속성이 사용 가능하지만 현재 테이블에 데이터가 없는 경우 RESETDICTIONARY 조작용은 기존의 사전을 보존합니다. 내부 최소 레코드 길이보다 크기가 작은 행 및 압축해도 레코드 길이가 줄어들지 않는 행은 이 경우 "충분하지 않다"고 간주됩니다.

다음 표에서는 LONGLOBDATA 옵션 지정 시 REORG 명령에서 RESETDICTIONARY 구문의 동작에 대해 설명합니다.

표 34. LONGLOBDATA 옵션이 지정될 때 REORG RESETDICTIONARY

압축	테이블 행 데이터 사전이 존재합니다.	XML 스토리지 오브젝트 사전이 존재합니다. <sup>1</sup>	데이터 사전	데이터 압축
Y	Y	Y	사전을 빌드합니다. <sup>2 3</sup>	기존 데이터가 압축됩니다. 새 데이터가 압축됩니다.
Y	Y	N	새 테이블 행 사전을 빌드하고 새 XML 사전을 작성합니다. <sup>3</sup>	기존 데이터가 압축됩니다. 새 데이터가 압축됩니다.
Y	N	Y	테이블 행 데이터 사전을 작성하고 새 XML 사전을 빌드합니다.	기존 데이터가 압축됩니다. 새 데이터가 압축됩니다.
Y	N	N	사전을 작성합니다.	기존 데이터가 압축됩니다. 새 데이터가 압축됩니다.

표 34. LONGLOBDATA 옵션이 지정될 때 REORG RESETDICTIONARY (계속)

압축	테이블 행 데이터 사전이 존재합니다.	XML 스토리지 오브젝트 사전이 존재합니다. <sup>1</sup>	데이터 사전	데이터 압축
N	Y	Y	사전을 제거합니다. 기존 및 새 데이터는 압축되지 않습니다.	기존 테이블 데이터가 압축 해제됩니다. 새 데이터가 압축되지 않습니다.
N	Y	N	테이블 행 사전을 제거합니다. 모든 데이터가 압축 해제됩니다.	기존 테이블 데이터가 압축 해제됩니다. 새 데이터가 압축되지 않습니다.
N	N	Y	XML 스토리지 오브젝트 사전을 제거합니다.	기존 테이블 데이터가 압축 해제됩니다. 새 데이터가 압축되지 않습니다.
N	N	N	영향을 미치지 않습니다.	기존 테이블 데이터가 압축 해제됩니다. 새 데이터가 압축되지 않습니다.

**참고:**

1. XML 컬럼이 DB2 V9.7 이상에서 테이블에 추가된 경우 또는 테이블이 온라인 테이블 이동을 사용하여 이주된 경우에만 테이블의 XML 스토리지 오브젝트의 압축 사전을 작성할 수 있습니다.
2. 사전이 존재하고 압축 속성이 사용 가능하지만 현재 테이블에 데이터가 없는 경우 RESETDICTIONARY 조작은 기존의 사전을 보존합니다. 내부 최소 레코드 길이보다 크기가 작은 행 및 압축해도 레코드 길이가 줄어들지 않는 행은 이 경우 충분하지 않다고 간주합니다.
3. DATA CAPTURE CHANGES 옵션을 CREATE TABLE 또는 ALTER TABLE문에 지정하면 현재 데이터 사전이 보존됩니다(실행기록 압축 사전이라고 함).

**ALL DBPARTITIONNUMS**

db2nodes.cfg 파일에 지정된 모든 데이터베이스 파티션에서 조작을 수행하도록 지정합니다. 이는 노드 절이 지정되지 않는 경우 디폴트입니다.

**EXCEPT**

노드 목록에 지정된 파티션을 제외하고 db2nodes.cfg 파일에 지정된 모든 데이터베이스 파티션에서 조작을 수행하도록 지정합니다.

**ON DATA PARTITION *partition-name***

재구성을 위한 데이터 파티션을 지정합니다.

이 옵션은 RECLAIM EXTENTS ONLY에만 사용됩니다.

지정된 테이블의 데이터 파티션 이름이 없는 경우, REORG가 실패하며 SQL0204N이 리턴됩니다.

### ON DBPARTITIONNUM | ON DBPARTITIONNUMS

데이터베이스 파티션 세트에 대해 조작을 수행합니다.

#### db-partition-number1

데이터베이스 파티션 목록의 데이터베이스 파티션 번호를 지정합니다.

#### db-partition-number2

두 번째 데이터베이스 파티션 번호를 지정하여 *db-partition-number1*에서 최고 *db-partition-number2*까지의 모든 데이터베이스 파티션이 데이터베이스 파티션 목록에 포함되도록 합니다.

### 예:

임시 테이블 스페이스 mytemp1을 사용하고 스페이스를 이용하기 위해 테이블을 재구성하려면 다음 명령을 입력하십시오.

```
db2 reorg table homer.employee use mytemp1
```

노드 1, 2, 3, 4의 4-노드 시스템으로 구성된 파티션 그룹에서 테이블을 재구성하려면 다음 명령 중 하나를 입력할 수 있습니다.

```
db2 reorg table employee index empid on dbpartitionnum (1,3,4)
```

```
db2 reorg table homer.employee index homer.empid on all  
dbpartitionnums except dbpartitionnum (2)
```

다른 트랜잭션이 테이블을 읽고 갱신할 수 있도록 하면서 EMPLOYEE 테이블의 모든 인덱스에서 의사(pseudo) 삭제된 키 및 의사(pseudo) 공백 페이지를 정리하려면 다음을 입력하십시오.

```
db2 reorg indexes all for table homer.employee allow write  
access cleanup only
```

다른 트랜잭션이 테이블을 읽고 갱신할 수 있도록 하면서 EMPLOYEE 테이블의 모든 인덱스에서 의사(pseudo) 공백 페이지를 정리하려면 다음을 입력하십시오.

```
db2 reorg indexes all for table homer.employee allow write  
access cleanup only pages
```

시스템 임시 테이블 스페이스 TEMPSPACE1을 작업 영역으로 사용하여 EMPLOYEE 테이블을 재구성하려면 다음을 입력하십시오.

```
db2 reorg table homer.employee use temp space1
```

이전 예에서 명시적으로 지정된 디폴트 스키마 HOMER를 사용하는 EMPLOYEE 테이블의 설정된 REORG를 시작, 일시정지 및 재개하려면 다음 명령을 입력하십시오.

```

db2 reorg table employee index empid inplace start
db2 reorg table employee inplace pause
db2 reorg table homer.employee inplace allow read access
nottruncate table resume

```

REORG를 재개하는 명령에는 테이블을 공유 잠금하는 절단 단계를 건너뛰고 읽기 액세스 전용을 지정하기 위한 추가 키워드가 들어 있습니다.

## 사용 시 참고사항

제한사항:

- REORG 유틸리티는 별칭 사용을 지원하지 않습니다.
- REORG TABLE 명령은 선언된 임시 테이블 또는 작성된 임시 테이블에 대해서는 지원되지 않습니다.
- REORG TABLE 명령은 뷰에 대해서는 사용할 수 없습니다.
- 테이블의 재구성은 범위 클러스터 테이블과는 호환 가능하지 않은데, 이는 이 테이블의 범위 영역이 항상 클러스터된 상태로 남아 있기 때문입니다.
- REORG TABLE은 LOB 및 인덱스를 포함하여 해당 테이블이 있는 테이블 스페이스의 온라인 백업이 수행되는 동안 DMS 테이블 스페이스의 파티션된 테이블에서 사용할 수 없습니다.
- REORG TABLE은 인덱스 확장자에 기초한 인덱스를 사용할 수 없습니다.
- 테이블이 REORG 보류 상태에 있는 경우 이 테이블에서 inplace REORG를 사용할 수 없습니다.
- 파티션된 테이블의 경우,
  - 테이블은 SYSCAT.TABLES에서 읽기/쓰기의 ACCESS\_MODE를 가지고 있어야 합니다.
  - 재구성은 접속 또는 접속 해제 조작에 따라 제한된 상태에 있는 데이터 파티션을 건너뛸 수 있습니다. 테이블 파티셔닝 절이 지정된 경우, 해당 파티션이 전체 액세스 가능해야 합니다.
  - 테이블 재구성 동안 오류가 발생하면, 일부 인덱스나 인덱스 파티션이 유효하지 않게 남습니다. 재구성이 최초 데이터 파티션의 교체 단계에 도달하거나 패스한 경우 테이블의 파티션되지 않은 인덱스는 유효하지 않은 것으로 표시됩니다. 이미 교체 단계에 도달하거나 패스한 데이터 파티션의 인덱스 파티션은 유효하지 않은 것으로 표시됩니다. 인덱스는 테이블 또는 데이터 파티션에 대한 다음 액세스에서 재빌드됩니다.
  - ALLOW NONE 액세스 모드 사용 시 인덱스 재구성 중 오류가 발생하면, 테이블의 일부 인덱스가 유효하지 않게 남습니다. 테이블에서 파티션되지 않은 RID 인덱스의 경우, 실패 시점에서 재구성 중인 인덱스만이 유효하지 않게 남습니다. 파티션되지 않은 블록 인덱스가 있는 MDC 테이블의 경우, 오류 발생 시 하나 이상의 블록 인덱스가 유효하지 않게 남습니다. 파티션된 인덱스의 경우, 재구성

되는 데이터 파티션의 인덱스 오브젝트만이 유효하지 않게 남습니다. 유효하지 않은 것으로 표시된 인덱스는 테이블 또는 데이터 파티션에 대한 다음 액세스에서 재빌드됩니다.

테이블 재구성의 현재 진행 상태에 대한 정보는 데이터베이스 활동에 대한 실행기록 파일에 쓰여집니다. 이 실행기록 파일에는 각 재구성 이벤트에 대한 레코드가 들어 있습니다. 이 파일을 보려면 재구성 중인 테이블이 있는 데이터베이스에 대해 LIST HISTORY 명령을 실행하십시오.

또한 테이블 스냅샷을 사용하여 테이블 재구성의 진행 상태를 모니터링할 수 있습니다. 테이블 재구성 모니터링 데이터는 데이터베이스 모니터 테이블 전환 설정에 관계없이 기록됩니다.

오류가 발생할 경우 이 실행기록 파일에 SQLCA 덤프가 쓰여집니다. Inplace 테이블 재구성의 경우 상태는 PAUSED로 기록됩니다.

인덱스가 작성된 테이블이 여러 번 수정된 경우, 인덱스에 있는 데이터가 조각난 상태가 될 수 있습니다. 테이블이 인덱스에 관해 클러스터된 경우, 테이블과 인덱스는 클러스터 순서를 벗어날 수 있습니다. 이들 두 요인은 모두 인덱스를 사용하는 스캔 성능에 불리한 영향을 줄 수 있으며 인덱스 페이지 프리페치의 효과에 영향을 줄 수 있습니다. REORG INDEX 또는 REORG INDEXES를 사용하여 테이블에 있는 인덱스 중 하나 또는 전체를 재구성할 수 있습니다. 인덱스 재구성은 모든 단편화를 제거하고 실제 클러스터링을 리프 페이지로 리스토어합니다. REORGCHK를 사용하면 인덱스를 재구성할 필요가 있는지 판별하는 데 도움이 됩니다. 인덱스 재구성을 호출하기 전에 모든 데이터베이스 조작을 완료하고 모든 잠금을 해제해야 합니다. 이는 WITH HOLD로 열려진 모든 커서를 닫고 COMMIT를 발행하거나 ROLLBACK을 발행하여 수행할 수 있습니다.

클래식 테이블 재구성(오프라인 재구성)은 재구성의 마지막 단계 동안 인덱스를 재빌드합니다. 그러나 Inplace 테이블 재구성(온라인 재구성)은 인덱스를 재빌드하지 않습니다. Inplace 테이블 재구성 완료 후 REORG INDEXES 명령을 발행하는 것이 좋습니다. Inplace 테이블 재구성은 비동기이므로, REORG INDEXES 명령을 발행하기 전에 inplace 테이블 재구성 완료를 확인하는 것이 필요합니다. Inplace 테이블 재구성 완료 이전에 REORG INDEXES 명령을 발행하면, 재구성이 실패합니다(SQLCODE -2219).

테이블이 여러 번 수정되어 데이터가 분할되고 액세스 성능이 현저하게 느려지게 되면 REORG TABLE 명령의 후보가 됩니다. 또한 변경된 인라인 길이를 이용하기 위해 구조화된 유형 컬럼의 인라인 길이를 변경한 후에는 이 유틸리티를 호출해야 합니다. 테이블을 재구성할 필요가 있는지 여부를 판별하려면 REORGCHK를 사용하십시오. REORG TABLE을 호출하기 전에 반드시 모든 데이터베이스 조작을 완료하고 모든 잠금을 릴리스하십시오. 이는 WITH HOLD로 열려진 모든 커서를 닫고 COMMIT를 발행하거나 ROLLBACK을 발행하여 수행할 수 있습니다. 테이블을 재구성한 후

RUNSTATS를 사용하여 테이블 통계를 갱신하고 이 테이블을 사용하는 패키지를 리바인드하십시오. 재구성 유틸리티는 내재적으로 모든 커서를 닫습니다.

테이블 값 압축이 활성화되거나 비활성화되었기 때문에 테이블에 혼합 행 형식이 들어 있는 경우, 오프라인 테이블 재구성이 모든 기존 행을 목표 행 형식으로 변환할 수 있습니다.

테이블이 여러 데이터베이스 파티션에 분산되어 있으며 영향을 받는 데이터베이스 파티션 중 하나에서 테이블 재구성이 실패할 경우, 실패한 데이터베이스 파티션에서만 테이블 또는 인덱스 재구성이 롤백됩니다.

재구성이 실패할 경우 임시 파일이 삭제되어서는 안 됩니다. 데이터베이스 관리 프로그램이 데이터베이스를 복구할 때 이들 파일을 사용합니다.

인덱스 이름이 지정된 경우, 데이터베이스 관리 프로그램은 이 인덱스에 있는 순서에 따라 데이터를 재구성합니다. 성능을 최대화하려면 SQL 쿼리에서 자주 사용되는 인덱스를 지정하십시오. 인덱스 이름이 지정되지 않고 클러스터링 인덱스가 존재할 경우 데이터는 클러스터링 인덱스에 따라 정렬됩니다.

테이블의 PCTFREE 값은 페이지 당 지정된 여유 공간의 크기를 결정합니다. 이 값이 설정되어 있지 않으면 유틸리티는 각 페이지에서 가능한 많은 공간을 채웁니다.

테이블 재구성 후 테이블 스페이스 롤포워드 복구를 완료하려면 일반 및 대형 테이블 스페이스가 모두 롤포워드 복구에 사용할 수 있어야 합니다.

테이블에 COMPACT 옵션을 사용하지 않는 LOB 컬럼이 있는 경우, 테이블 재구성 후 LOB DATA 스토리지 오브젝트가 훨씬 더 커질 수 있습니다. 이는 행이 재구성된 순서와 사용된 테이블 스페이스의 유형(SMS 또는 DMS)의 결과가 될 수 있습니다.

XML 데이터의 인덱스는 REORG INDEXES/TABLE 명령으로 재작성될 수 있습니다. 자세한 내용은 『XML 데이터에서 인덱스 재작성』을 참조하십시오.

## 제 104 장 REORGCHK

테이블이나 인덱스, 또는 둘 다 재구성하거나 정리해야 하는지 여부를 판별하기 위해 데이터베이스에 대해 통계를 계산합니다.

### 범위

이 명령은 `db2nodes.cfg` 파일에서 모든 데이터베이스 파티션으로부터 발행될 수 있습니다. 카탈로그에서 테이블 및 인덱스 통계를 갱신하기 위해 사용할 수 있습니다.

### 권한 부여

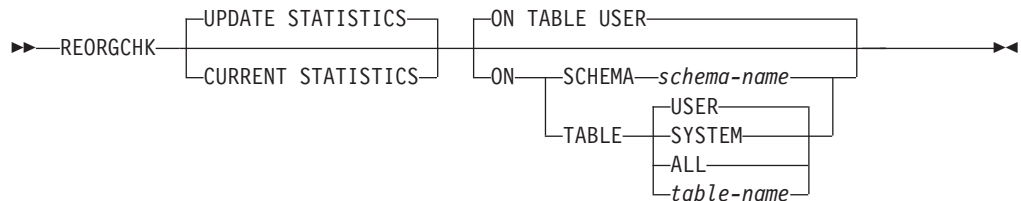
다음 중 하나가 필요합니다.

- `sysadm` 또는 `dbadm` 권한
- 테이블에 대한 CONTROL 특권

### 필수 연결

데이터베이스

### 명령 구문



### 명령 매개변수

#### UPDATE STATISTICS

RUNSTATS 루틴을 호출하여 테이블 및 인덱스 통계를 갱신한 후, 갱신된 통계를 사용하여 테이블 또는 인덱스 재구성이 필요한지 판별합니다.

REORGCHK가 발행된 데이터베이스 파티션에 테이블 분할 영역이 있는 경우, 유틸리티는 이 데이터베이스 파티션에서 실행됩니다. 이 데이터베이스 파티션에 테이블이 존재하지 않으면, 요청은 데이터베이스 파티션 그룹에서 테이블의 분할 영역을 소유하는 첫 번째 데이터베이스 파티션으로 보냅니다. RUNSTATS는 이 데이터베이스 파티션에서 실행됩니다.

#### CURRENT STATISTICS

현재 테이블 통계를 사용하여 테이블 재구성이 필요한지 판별합니다.

## **ON SCHEMA** *schema-name*

지정된 스키마에서 작성된 모든 테이블을 점검합니다.

## **ON TABLE**

**USER** 런타임 권한 부여 ID가 소유하는 테이블을 점검합니다.

## **SYSTEM**

시스템 테이블을 점검합니다.

**ALL** 모든 사용자 및 시스템 테이블을 점검합니다.

## *table-name*

점검할 테이블을 지정합니다. *schema.table-name* 양식의 완전한 이름이나 별명을 사용해야 합니다. *schema*는 테이블이 작성된 사용자 이름입니다. 지정된 테이블이 시스템 카탈로그 테이블인 경우 *schema*는 **SYSIBM**입니다. 유형이 지정된 테이블의 경우, 지정된 테이블명은 계층 구조의 루트 테이블 이름이어야 합니다.

## **예:**

**SAMPLE** 데이터베이스에 대해 다음 명령을 발행하십시오.

```
db2 reorgchk update statistics on table system
```

결과 출력에서, 테이블 통계(공식 1 - 3) 용어의 의미는 다음과 같습니다.

## **CARD**

(CARDINALITY) 기본 테이블의 행 수

**OV** (OVERFLOW) 오버플로우 행 수

**NP** (NPAGES) 데이터를 포함하는 페이지 수

**FP** (FPAGES) 총 페이지 수

## **ACTBLK**

다차원적으로 클러스터된(MDC) 테이블에 대한 활성 블록의 총 수. 이 필드는 **ORGANIZE BY** 절을 사용하여 정의된 테이블에만 적용 가능합니다. 데이터를 포함하는 테이블의 블록 수를 표시합니다.

## **TSIZE**

테이블 크기(바이트). 테이블의 행 수(CARD)와 평균 행 길이를 곱하여 계산됩니다. 평균 행 길이는 평균 컬럼 길이(SYSCOLUMNS의 AVGCOLLEN)에 행 오버헤드의 10바이트를 합하여 계산됩니다. Long 필드 및 LOB의 경우 대략적인 디스크립터 길이만 사용됩니다. 실제 Long 필드 또는 LOB 데이터는 **TSIZE**에서 계산되지 않습니다.

## **TABLEPAGESIZE**

테이블 데이터가 있는 테이블 스페이스의 페이지 크기



## NPARTITIONS

파티션된 테이블인 경우 파티션 수. 테이블이 파티션되지 않은 경우에는 1입니다.

**F1** 공식 1의 결과

**F2** 공식 2의 결과

**F3** 공식 3의 결과. 이 공식은 테이블에서 낭비된 스페이스 양을 표시합니다. 테이블의 페이지에 존재하는 데이터를 포함하는 페이지 수와 비어 있는 페이지 수로 측정됩니다. 다차원적으로 클러스터된(MDC) 테이블에서는 데이터를 포함하는 블록 수와 비어 있는 블록 수가 측정됩니다.

## REORG

이 컬럼에 표시되는 각 하이픈(-)은 계산된 결과가 해당 공식의 설정된 바운드 내에 있음을 표시하고, 각 별표(\*)는 계산된 결과가 해당 공식의 설정된 바운드를 초과했음을 표시합니다.

- 컬럼 왼쪽에 있는 - 또는 \*는 F1(공식 1)에 해당됩니다.
- 컬럼 중간에 있는 - 또는 \*는 F2(공식 2)에 해당됩니다.
- 컬럼 오른쪽에 있는 - 또는 \*는 F3(공식 3)에 해당됩니다.

계산 결과가 공식에서 설정된 바운드를 초과하는 경우 테이블을 재구성해야 합니다.

예를 들어, ---는 F1, F2 및 F3의 공식 결과가 공식의 설정된 바운드 내에 있으므로 테이블을 재구성하지 않아도 됨을 나타냅니다. \*-는 F2는 설정된 바운드 내에 있지만 F1 및 F3 결과는 테이블을 재구성해야 함을 나타냅니다. \*--는 F1이 바운드를 초과하는 유일한 공식임을 나타냅니다.

테이블 이름은 30자로 절단되고, 31번째 컬럼의 ">" 기호는 테이블 이름의 절단된 부분을 표시합니다. 테이블 이름의 접미부 『\*』는 MDC 테이블임을 표시합니다. 인덱스 이름의 『\*』 접미부는 MDC 차원 인덱스임을 표시합니다.

인덱스 통계(공식 4 - 8) 용어의 의미는 다음과 같습니다.

## INDCARD

(INDEX CARDINALITY) 인덱스의 인덱스 항목 수. 일부 인덱스의 경우 테이블 카디널리티(cardinality)와 다를 수 있습니다. 예를 들어 XML 컬럼에 대한 인덱스의 경우, 인덱스 카디널리티(cardinality)는 테이블 카디널리티보다 큼니다.

**LEAF** 총 인덱스 리프 페이지 수(NLEAF)

## ELEAF

수도(pseudo) 빈 인덱스 리프 페이지 수(NUM\_EMPTY\_LEAFS).

수도(pseudo) 빈 인덱스 리프 페이지는 모든 RID가 삭제 대상으로 표시되었지만 실제로 제거되지는 않은 페이지입니다.

#### **NDEL**

수도(pseudo) 삭제된 RID 수(NUMRIDS\_DELETED).

수도(pseudo) 삭제된 RID는 삭제 대상으로 표시된 RID입니다. 이 통계는 수도(pseudo) 비어 있지 않은 리프 페이지에 대한 수도 삭제된 RID를 보고합니다. 통계에는 모든 RID가 삭제 대상으로 표시된 리프 페이지에서 삭제 대상으로 표시된 RID가 포함되지 않습니다.

**KEYS** 삭제 대상으로 표시되지 않은 고유 인덱스 항목 수(FULLKEYCARD)

#### **LEAF\_RECSIZE**

리프 페이지에서 인덱스 항목의 레코드 크기. 오버헤드를 제외한 인덱스 항목의 평균 크기로, 인덱스에 참여하는 모든 컬럼의 평균 컬럼 길이에서 계산됩니다.

#### **NLEAF\_RECSIZE**

리프가 아닌 페이지에서 인덱스 항목의 레코드 크기. 오버헤드를 제외한 인덱스 항목의 평균 크기로, INCLUDE 컬럼을 제외하고 인덱스에 참여하는 모든 컬럼의 평균 컬럼 길이에서 계산됩니다.

#### **LEAF\_PAGE\_OVERHEAD**

내부 사용을 목적으로 인덱스 리프 페이지에서 예약된 스페이스

#### **NLEAF\_PAGE\_OVERHEAD**

내부 사용을 목적으로 인덱스 리프가 아닌 페이지에서 예약된 스페이스

#### **INDEXPAGESIZE**

인덱스가 있는 테이블 스페이스의 페이지 크기로, 인덱스 또는 테이블 작성 시 지정됩니다. 지정되지 않은 경우, INDEXPAGESIZE 값은 TABLEPAGESIZE 값과 같습니다.

**LVLS** 인덱스 레벨 번호(NLEVELS)

#### **PCTFREE**

여유 공간으로 남겨둘 각 인덱스 페이지의 백분율(인덱스를 정의할 때 지정되는 값)을 지정합니다. 값의 범위는 0 - 99입니다. 디폴트값은 10입니다.

#### **LEAF\_RECSIZE\_OVERHEAD**

리프 페이지의 인덱스 레코드 오버헤드. LARGE 테이블 스페이스에 있는 테이블의 인덱스에 대해 오버헤드는 파티션된 테이블의 경우 11이고 기타 테이블에 대해서는 9입니다. REGULAR 테이블 스페이스에 있는 테이블의 인덱스에 대해 이 값은 파티션된 테이블의 경우 9이고 기타 테이블의 경우 7입니다. 이

규칙에 대한 유일한 예외는 오버헤드가 항상 9인 XML 경로와 XML 영역 인덱스입니다. 이 정보는 쉽게 참조하기 위해 아래 테이블에서도 사용할 수 있습니다.

### NLEAF\_RECSIZE\_OVERHEAD

리프가 아닌 페이지의 인덱스 레코드 오버헤드. LARGE 테이블 스페이스에 있는 테이블의 인덱스에 대해 오버헤드는 파티션된 테이블의 경우 14이고 기타 테이블의 경우 12입니다. REGULAR 테이블 스페이스에 있는 테이블의 인덱스에 대해, 이 값은 파티션된 테이블의 경우 12이고 기타 테이블의 경우 10입니다. 이 규칙에 대한 유일한 예외는 오버헤드가 항상 12인 XML 경로와 XML 영역 인덱스입니다. 이 정보는 쉽게 참조하기 위해 아래 테이블에서도 사용할 수 있습니다.

### DUPKEYSIZE

인덱스 리프 페이지의 중복 키 크기. LARGE 테이블 스페이스에 있는 테이블의 인덱스에 대해 DUPKEYSIZE는 파티션된 테이블의 경우 9이고 기타 테이블에 대해서는 7입니다. REGULAR 테이블 스페이스에 있는 테이블의 인덱스에 대해, 이 값은 파티션된 테이블의 경우 7이고 기타 테이블의 경우 5입니다. 이 규칙에 대한 유일한 예외는 DUPKEYSIZE가 항상 7인 XML 경로와 XML 영역 인덱스입니다. 이 정보는 쉽게 참조하기 위해 아래 테이블에서도 사용할 수 있습니다.

표 35. LEAF\_RECSIZE\_OVERHEAD, NLEAF\_RECSIZE\_OVERHEAD 및 DUPKEYSIZE 값은 인덱스 유형, 테이블 파티셔닝 및 테이블 스페이스 유형의 함수임

변수	REGULAR 테이블 스페이스의 데이터			LARGE 테이블 스페이스의 데이터**		
	일반 테이블		파티션된 테이블	일반 테이블		파티션된 테이블
	XML 경로 또는 영역 인덱스	기타 모든 인덱스	모든 인덱스	XML 경로 또는 영역 인덱스	기타 모든 인덱스	모든 인덱스
LEAF_RECSIZE_OVERHEAD	9	7	9	9	9	11
NLEAF_RECSIZE_OVERHEAD	12	10	12	12	12	14
DUPKEYSIZE	7	5	7	7	7	9

\*\* 대형 테이블 스페이스에 있는 테이블의 인덱스에 대해, 인덱스는 대형 RID를 가지고 있는 것으로 간주됩니다. 이로서 테이블의 테이블 스페이스가 대형으로 변환되었지만 인덱스가 아직 재작성되거나 재구성되지 않은 경우 일부 공식이 정확하지 않은 결과를 제공할 수 있습니다.

**F4** 공식 4의 결과

**F5** 공식 5의 결과. +++는 결과가 999를 초과하여 유효하지 않음을 표시합니다. UPDATE STATISTICS 옵션과 함께 REORGCHK를 재실행하거나, RUNSTATS와 REORGCHK 명령을 차례대로 발행하십시오.

**F6** 공식 6의 결과. +++는 결과가 999를 초과하여 유효하지 않을 수 있음을 표시합니다. UPDATE STATISTICS 옵션과 함께 REORGCHK를 재실행하거나, RUNSTATS와 REORGCHK 명령을 차례대로 발행하십시오. 통계가 최신 상태이고 유효한 경우 재구성해야 합니다.

**F7** 공식 7의 결과

**F8** 공식 8의 결과

### REORG

이 컬럼에 표시되는 각 하이픈(-)은 계산된 결과가 해당 공식의 설정된 바운드 내에 있음을 표시하고, 각 별표(\*)는 계산된 결과가 해당 공식의 설정된 바운드를 초과했음을 표시합니다.

- 왼쪽 컬럼에 있는 - 또는 \*는 F4(공식 4)에 해당됩니다.
- 왼쪽 컬럼에서 두 번째에 있는 - 또는 \*는 F5(공식 5)에 해당됩니다.
- 중간 컬럼에 있는 - 또는 \*는 F6(공식 6)에 해당됩니다.
- 오른쪽에서 두 번째 컬럼에 있는 - 또는 \*는 F7(공식 7)에 해당됩니다.
- 오른쪽 컬럼에 있는 - 또는 \*는 F8(공식 8)에 해당됩니다.

인덱스 재구성 조언은 다음과 같습니다.

- 공식 1, 2 및 3에 대한 계산 결과가 공식에서 설정된 바운드를 초과하고 공식 4, 5 또는 6에 대한 계산 결과가 설정된 바운드를 초과하는 경우 인덱스 재구성이 권장됩니다.
- 공식 7에 대한 계산 결과가 설정된 바운드를 초과하지만 공식 1, 2, 3, 4, 5 및 6의 결과가 설정된 바운드 내에 있는 경우, 인덱스 재구성의 CLEANUP ONLY 옵션을 사용하여 인덱스를 정리할 것을 권장합니다.
- 설정된 바운드를 초과하는 유일한 계산 결과가 공식 8의 결과인 경우, 인덱스 재구성의 CLEANUP ONLY PAGES 옵션을 사용하여 인덱스의 수도(pseudo) 빈 페이지를 정리할 것을 권장합니다.

파티션된 테이블에서 공식(5 - 8)의 결과는 통계 수집 시기에 따라 혼동될 수 있습니다. 데이터 파티션이 접속 해제된 경우, 접속 해제된 파티션의 인덱스 키는 즉시 정리되지 않습니다. 대신, 정리는 지연되고 결국 백그라운드에서 비동기식으로 작동하는 인덱스 클리너에 의해 키가 제거됩니다(이를 비동기 인덱스 정리 또는 AIC라고 함). 인덱스에 정리를 보류하는 인덱스 키가 있어도, 볼 수 없고 더 이상 테이블의 일부가 아니므로 통계에서 키의 일부로 계산되지 않습니다. 따라서 비동기 인덱스 정리가 실행되기 전에 수집된 통계는 잘못된 것입니다. 비동기 인덱스 정리가 완료되기 전에 REORGCHK 명령이 발행될 경우 부정확한 통계에 따른 인덱스 재구성 또는 인덱스 정리에 대한 거짓 정보가 생성될 수 있습니다. 비동기 인덱스 정리가 실행된 후, 정리할 필요가 있는 접속 해제된 데이터 파티션에 여전히 속해 있는 인덱스 키는 모두 제거되므로 인덱스를 재구성할 필요가 없도록 합니다.

파티션된 테이블의 경우, 접속 해제된 데이터 파티션이 있는 상태에서 올바른 인덱스 통계를 생성하려면 비동기 인덱스 정리가 완료된 후 REORGCHK 명령을 발행하는 것이 바람직합니다. 테이블에 접속 해제된 데이터 파티션이 있는지 여부를 판별하려면 SYSDATAPARTITIONS 테이블의 상태 필드를 점검하여 I(인덱스 정리) 또는 D(중속 MQT와 접속 해제됨)를 찾아 볼 수 있습니다.

## 사용 시 참고사항

이 명령은 선언된 임시 테이블 또는 작성된 임시 테이블에 대해서는 통계 정보를 표시하지 않습니다.

이 유틸리티는 별칭 사용을 지원하지 않습니다.

새 서버 릴리스는 새 테이블 및 인덱스 기능을 도입할 수 있습니다. 이 새 기능은 REORGCHK 논리(즉, REORGCHK가 REORG 권장사항을 계산하는 방법)에 영향을 줄 수 있습니다. 서버와 동일한 레벨이 아니라 클라이언트에서 REORGCHK가 발행되는 경우, 서버와 동일한 레벨에서 클라이언트로부터 보고되는 것과 다른 결과를 보고할 수 있습니다. REORGCHK는 클라이언트 응용프로그램이므로, 서버와 동일한 레벨을 실행 중인 클라이언트에서 실행해야 합니다. 이와 같이 하면 가장 정확한 보고서가 생성됩니다. 서버 관리 작업의 경우, 일반적으로 동일한 레벨에서 서버와 클라이언트를 사용하십시오.

CURRENT STATISTICS 옵션을 지정하지 않으면, REORGCHK는 디폴트 옵션만 사용하여 모든 컬럼에 대해 통계를 수집합니다. 특히, 컬럼 그룹은 수집되지 않으며 LIKE 통계가 이전에 수집된 경우 REORGCHK에서 수집되지 않습니다. 수집되는 통계는 현재 카탈로그 테이블에 저장된 통계 종류에 따라 결정됩니다.

- 임의 인덱스의 카탈로그에 세부 인덱스 통계가 존재하는 경우, 모든 인덱스에 대한 테이블 통계 및 세부 인덱스 통계(샘플링 없이)가 수집됩니다.
- 세부 인덱스 통계가 발견되지 않으면, 테이블 통계와 일반 인덱스 통계가 모든 인덱스에 대해 수집됩니다.
- 분산 통계가 발견되면, 테이블에 대해 분산 통계가 수집됩니다. 분산 통계가 수집되는 경우, 자주 사용되는 값 수와 Quantile 수는 구성 매개변수 설정을 기초로 합니다.

REORGCHK는 성능이 저하되었거나 테이블 또는 해당 인덱스를 재구성하여 개선할 수 있는지 판별하기 위해 8개의 서로 다른 공식에서 확보되는 통계를 계산합니다. 테이블이 페이지 중 (NPARTITIONS \* 1 Extent 크기) 이하를 사용하는 경우, 각 공식을 기초로 테이블 재구성이 권장되지 않습니다. 특히,

- 파티션되지 않은 테이블의 경우(NPARTITIONS =1) 임계값은 다음과 같습니다.  
(FPAGES <= 1 extent size)
- 파티션된 테이블의 경우 다음과 같습니다.

(FPAGES <= NPARTITIONS \* 1 extent size)

- 다중 파티션된 데이터베이스의 경우, 테이블의 데이터베이스 파티션 그룹에 있는 데이터베이스 파티션 수가 고려된 후, 테이블 재구성을 권장하지 않기 위해 이 임계값이 다음으로 변경됩니다.

FPAGES <= 'number of database partitions in a database partition group of the table' \* NPARTITIONS \* 1 extent size

Long 필드 또는 LOB 데이터는 TSIZE 계산에서 고려되지 않습니다.

REORGCHK는 다음 공식을 사용하여 테이블의 행 및 크기의 실제 위치를 분석합니다.

- 공식 F1:

$$100 * \text{OVERFLOW} / \text{CARD} < 5$$

테이블에 있는 총 오버플로우 행 수는 총 행 수의 5퍼센트 미만이어야 합니다. 행이 갱신되고 새 행이 이전 행보다 많은 바이트를 포함할 때(VARCHAR 필드), 또는 컬럼이 기존 테이블에 추가될 때 오버플로우 행이 작성될 수 있습니다.

- 공식 F2:

일반 테이블의 경우:

$$100 * \text{TSIZE} / ((\text{FPAGES} - \text{NPARTITIONS}) * (\text{TABLEPAGESIZE} - 68)) > 70$$

바이트 단위의 테이블 크기(TSIZE)는 테이블에 할당된 총 스페이스의 70퍼센트보다 커야 합니다. (30% 미만의 여유 공간이 있어야 합니다.) 테이블에 할당된 총 스페이스는 테이블이 있는 테이블 스페이스의 페이지 크기(이 크기에서 68바이트의 오버헤드를 뺀)에 따라 결정됩니다. 데이터 오브젝트에서 할당된 마지막 페이지는 보통 채워지지 않으므로, 파티션마다 FPAGES에서 1을 뺍니다(FPAGES - NPARTITIONS와 같음).

MDC 테이블의 경우:

$$100 * \text{TSIZE} / ((\text{ACTBLK} - \text{FULLKEYCARD}) * \text{EXTENTSIZE} * (\text{TABLEPAGESIZE} - 68)) > 70$$

FULLKEYCARD는 MDC 테이블에 대한 복합 차원 인덱스의 카디널리티(cardinality)를 나타냅니다. Extentsize는 블록당 페이지 수입니다. 공식은 바이트 단위의 테이블 크기가 필요한 최소 블록 수를 뺀 후 테이블에 대해 남은 블록의 70퍼센트보다 크지 점검합니다.

- 공식 F3:

$$100 * \text{NPAGES} / \text{FPAGES} > 80$$

행을 포함하지 않는 페이지 수는 총 페이지 수의 20퍼센트 미만이어야 합니다. (페이지는 행이 삭제된 후 빈 상태가 될 수 있습니다.) 위에 명시된 것처럼, (FPAGES <= NPARTITIONS \* 1 extent size)의 경우 테이블 재구성은 권장되지 않습니다.

따라서 F3은 계산되지 않습니다. 파티션되지 않은 테이블의 경우 NPARTITIONS = 1 입니다. 다중 파티션된 데이터베이스에서, 이 조건은 FPAGES = 'number of database partitions in a database partition group of the table' \* NPARTITIONS \* 1 extent size로 변경됩니다.

MDC 테이블의 경우 공식은 다음과 같습니다.

$$100 * \text{activeblocks} / ( (\text{fpages} / \text{ExtentSize}) - 1 )$$

REORGCHK는 다음 공식을 사용하여 인덱스와, 테이블 데이터와의 인덱스 관계를 분석합니다.

• 공식 F4:

– 파티션되지 않은 테이블의 경우:

$$\text{CLUSTERRATIO or normalized CLUSTERFACTOR} > 80$$

전역 CLUSTERFACTOR 및 CLUSTERRATIO에는 인덱스 키 및 분산 키 사이의 상관관계가 고려됩니다. 인덱스의 클러스터 비율은 80퍼센트를 초과해야 합니다. 하나의 테이블에서 여러 개의 인덱스가 정의되는 경우 일부 인덱스는 클러스터 비율이 낮습니다. (인덱스 시퀀스는 테이블 시퀀스와 같지 않습니다.) 이는 피할 수 없습니다. 테이블을 재구성할 때 가장 중요한 인덱스를 지정하도록 하십시오. 클러스터 비율은 보통 많은 중복 키와 많은 항목을 포함하는 인덱스에 최적이지 않습니다.

– 파티션된 테이블의 경우,

$$\text{AVGPARTITION\_CLUSTERRATIO or normalized AVGPARTITION\_CLUSTERFACTOR} > 80$$

AVGPARTITION\_CLUSTERFACTOR 및 AVGPARTITION\_CLUSTERRATIO 값은 인덱스 키에 대해 데이터 파티션 내에서 데이터가 클러스터된 방법을 반영합니다. 파티션된 테이블은 각 데이터 파티션 내의 특정 인덱스 키에 대해 완전하게 클러스터될 수 있으며, CLUSTERFACTOR 및 CLUSTERRATIO에 대해 여전히 낮은 값을 가질 수 있습니다. 인덱스 키가 테이블 파티션 키의 접두부가 아니기 때문입니다. 가장 중요한 인덱스를 테이블 파티션 키의 접두부로 사용하여 테이블 및 인덱스를 설계하십시오. 또한, 옵티마이저는 전역 clusteredness 값을 사용하여 여러 데이터 파티션에 걸쳐 있는 쿼리에 대해 결정하므로, 클러스터링 재구성을 수행하고 키가 일치하지 않을 때 옵티마이저가 계속 클러스터링 인덱스를 선택하지 않도록 할 수 있습니다.

• 공식 F5:

– 단일 데이터베이스 파티션의 경우:

$$\begin{aligned} & 100 * ( \text{KEYS} * ( \text{LEAF\_RECSIZE} + \text{LEAF\_RECSIZE\_OVERHEAD} ) + ( \text{INDCARD} - \text{KEYS} ) * \text{DUPKEYSIZE} ) \\ & / ( ( \text{NLEAF} - \text{NUM\_EMPTY\_LEAFS} - 1 ) * ( \text{INDEXPAGESIZE} - \text{LEAF\_PAGE\_OVERHEAD} ) ) \\ & > \text{MIN}(50, (100 - \text{PCTFREE})) \end{aligned}$$

인덱스의 리프 레벨에서 사용 중인 스페이스는 최소 50 및 100 - PCTFREE 퍼센트보다 커야 합니다(NLEAF>1인 경우에만 점검함).

- 다중 파티션 데이터베이스 환경의 경우:

$$100 * ( KEYS * (LEAF\_RECSIZE + LEAF\_RECSIZE\_OVERHEAD) + (INDCARD - KEYS) * DUPKEYSIZE ) / ( (NLEAF - NUM\_EMPTY\_LEAFS - NPARTITIONS) * (INDEXPAGESIZE - LEAF\_PAGE\_OVERHEAD) ) > MIN(50, (100 - PCTFREE))$$

• 공식 F6:

$$(100 - PCTFREE) * ( FLOOR((100 - LEVEL2PCTFREE) / 100 * (INDEXPAGESIZE - NLEAF\_PAGE\_OVERHEAD) / (NLEAF\_RECSIZE + NLEAF\_RECSIZE\_OVERHEAD))) * (FLOOR((100 - MIN(10, LEVEL2PCTFREE)) / 100 * (INDEXPAGESIZE - NLEAF\_PAGE\_OVERHEAD) / (NLEAF\_RECSIZE + NLEAF\_RECSIZE\_OVERHEAD))) ** (NLEVELS - 3)) * (INDEXPAGESIZE - LEAF\_PAGE\_OVERHEAD) / (KEYS * (LEAF\_RECSIZE + LEAF\_RECSIZE\_OVERHEAD) + (INDCARD - KEYS) * DUPKEYSIZE) ) < 100$$

인덱스를 재작성한 결과 더 적은 레벨을 갖는 트리가 생성되는지 여부를 판별합니다. 이 공식은 현재 트리보다 레벨이 하나 적은 인덱스 트리의 스페이스 양과 필요한 스페이스 양 사이의 비율을 점검합니다. 레벨이 하나 적은 트리를 작성하고 계속 PCTFREE가 사용 가능하도록 둘 수 있으면, 재구성이 권장됩니다. 실제 인덱스 항목 수는 NLEVELS - 1 인덱스 트리가 처리할 수 있는 항목 수의 (100 - PCTFREE) 퍼센트보다 커야 합니다(NLEVELS>2의 경우에만 점검함). NLEVELS = 2의 경우, 인덱스를 재구성해야 하는지 여부를 판별하려면 다른 REORGCHK 공식에 의존해야 합니다.

간단한 양식으로, 공식 F6은 다음과 같이 다시 작성할 수 있습니다.

$$\frac{\text{Amount of space needed for an index if it was one level smaller}}{\text{Amount of space needed for all the entries in the index}} < 1$$

위의 왼쪽 부분이 > 1이면, 이는 기존 인덱스의 모든 인덱스 항목이 기존 인덱스보다 한 레벨 적은 인덱스에 맞출 수 있음을 의미합니다. 이 경우, 인덱스 reorg가 권장됩니다.

NLEVELS - 1 인덱스에 필요한 스페이스 양은 다음과 같이 계산됩니다.

$$\text{(The max number of leaf pages that a NLEVELS-1 index can have) * (Amount of space available to store index entries per leaf page)}$$

여기서,

$$\begin{aligned} &\text{The max number of leaf pages that a NLEVELS-1 index can have} = \\ &\text{(No. of entries a level 2 index page can have) *} \\ &\text{(No. of entries per page on levels greater than 2) **} \\ &\text{(No. of levels in the intended index - 2) =} \end{aligned}$$

$$\left\{ \frac{(100 - LEVEL2PCTFREE)}{100} \right\} * \left[ \frac{(\text{Page Size} - \text{Overhead})}{(\text{Avg. size of each nonleaf index entry})} \right] * (100 - \text{MIN}(10, LEVEL2PCTFREE))$$



$$\text{FLOOR}\left(\frac{\text{-----}}{100} * \frac{(\text{PageSize} - \text{Overhead})}{\text{-----}}\right) **$$

(Avg. size of each nonleaf index entry)

(NLEVELS-3) }

(100 - LEVEL2PCTFREE) is the percentage of used space on level 2 of the index.

Level 2 is the level immediately above the leaf level.

(100 - MIN(10, LEVEL2PCTFREE)) is the percentage of used space on all levels above the second level.

NLEVELS is the number of index levels in the existing index.

The amount of space available to store index entries per leaf page =  
 $((100 - \text{PCTFREE}) / 100 * (\text{INDEXPAGESIZE} - \text{LEAF\_PAGE\_OVERHEAD})) =$   
 ( Used space per page \* (PageSize - Overhead) )

The amount of space needed for all index entries:  
 $\text{KEYS} * (\text{LEAF\_RECSIZE} + \text{LEAF\_RECSIZE\_OVERHEAD}) +$   
 $(\text{INDCARD} - \text{KEYS}) * \text{DUPKEYSIZE}$

$(\text{KEYS} * (\text{LEAF\_RECSIZE} + \text{LEAF\_RECSIZE\_OVERHEAD}))$ 는 인덱스에서 각 키 값의 첫 번째 어커런스에 사용되는 스페이스를 나타내고,  $(\text{INDCARD} - \text{KEYS}) * \text{DUPKEYSIZE}$ 는 키 값의 연속(중복) 어커런스에 사용되는 스페이스를 나타냅니다.

- 공식 F7:

$$100 * (\text{NUMRIDS\_DELETED} / (\text{NUMRIDS\_DELETED} + \text{INDCARD})) < 20$$

수도(pseudo)가 아닌 빈 페이지의 수도 삭제된 RID 수는 20퍼센트 미만이어야 합니다.

- 공식 F8:

$$100 * (\text{NUM\_EMPTY\_LEAFS} / \text{NLEAF}) < 20$$

수도(pseudo) 빈 리프 페이지 수는 총 리프 페이지 수의 20퍼센트 미만이어야 합니다.

많은 테이블에 대해 통계를 실행하면 시간이 소요될 수 있습니다(특히, 테이블이 대형일 경우).

## 인덱스 압축에 대한 사용 시 참고사항

공식 F5는 키에 필요한 스페이스 양과 할당된 스페이스 양 사이의 비율을 판별합니다. 공식 F6은 인덱스를 재작성한 결과 더 적은 레벨을 갖는 트리가 생성되는지 여부를 판별합니다. 다음 공식은 현재 트리보다 레벨이 하나 적은 인덱스 트리의 스페이스 양과 필요한 스페이스 양 사이의 비율을 점검합니다. 이 공식은 모든 인덱스 항목에 필요한 스페이스 양에 따라 다릅니다. 두 공식 모두 모든 인덱스 항목에 필요한 스페이스 양을 사용합니다.

압축되지 않은 인덱스의 모든 인덱스 항목에 필요한 스페이스 양은 다음과 같습니다.

$$\text{KEYS} * (\text{LEAF\_RECSIZE} + \text{LEAF\_RECSIZE\_OVERHEAD}) + (\text{INDCARD} - \text{KEYS}) * \text{DUPKEYSIZE}$$

여기서 LEAF\_RECSIZE는 입력 키의 평균 크기이고 DUPKEYSIZE는 RID의 크기입니다.

압축된 인덱스에서, LEAF\_RECSIZE는 접두부 압축의 영향을 받습니다. DUPKEYSIZE는 인덱스에 대해 중복 키 크기를 측정하기 위한 신뢰할 수 있는 방법은 아닙니다. 압축된 인덱스에서 필요한 스페이스 양은 모든 압축되지 않은 인덱스 항목에 필요한 스페이스 양에 인덱스 압축 비율을 곱한 값입니다.

$$(\text{KEYS} * (\text{LEAF\_RECSIZE} + \text{LEAF\_RECSIZE\_OVERHEAD}) + (\text{INDCARD} - \text{KEYS}) * \text{DUPKEYSIZE}) * \text{COMPRESSION\_RATIO}$$

COMPRESSION\_RATIO는 인덱스에서 추정된 인덱스 압축 비율입니다.

COMPRESSION\_RATIO는 다음과 같이 계산됩니다.

$$(100 - \text{PCT\_PAGES\_SAVED}) / 100$$

PCT\_PAGES\_SAVED는 인덱스 압축으로 절약된 리프 페이지의 추정 백분율입니다. 이 값은 카탈로그에서 가져옵니다. 통계가 수집되지 않으면, PCT\_PAGES\_SAVED는 카탈로그에서 -1이고 COMPRESSION\_RATIO는 1입니다.

REORGCHK 명령과 REORGCHK\_IX\_STATS 프로시저 모두 PCT\_PAGES\_SAVED 값을 표시합니다.

---

## 제 105 장 RESET ADMIN CONFIGURATION

연결되어 있는 노드의 DB2 Administration Server(DAS) 구성 파일에 있는 항목을 재설정합니다. DAS는 DB2 서버의 리모트 관리를 가능하게 하는 특수 관리 도구입니다. 값은 노드 유형에 의해 재설정되는데, 항상 리모트 클라이언트를 갖는 서버입니다. DAS 매개변수의 목록은 UPDATE ADMIN CONFIGURATION 명령 설명을 참조하십시오.

### 범위

이 명령은 사용자가 접속되어 있는 시스템의 관리 노드에 있는 DAS 구성 파일을 재설정합니다.

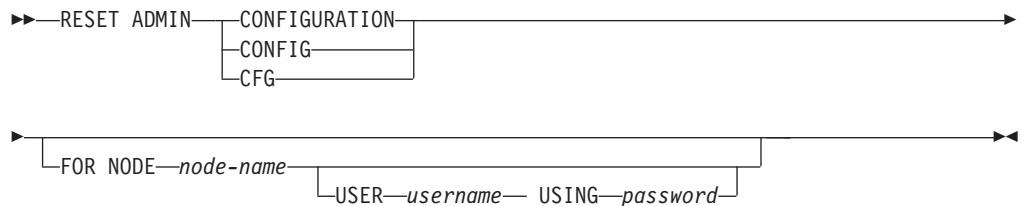
### 권한 부여

*dasadm*

### 필수 연결

파티션. 리모트 시스템의 DAS 구성을 재설정하려면 관리 노드 이름과 함께 FOR NODE 옵션을 사용하여 시스템을 지정하십시오.

### 명령 구문



### 명령 매개변수

#### FOR NODE *node-name*

DAS 구성 매개변수를 재설정할 관리 노드의 이름을 입력하십시오.

#### USER *username* USING *password*

리모트 시스템에 연결할 때 사용자 이름 및 암호가 필요한 경우 이 정보를 입력하십시오.

## 사용 시 참고사항

리모트 시스템에서 DAS 구성 매개변수를 재설정하려면 관리 노드 이름을 FOR NODE 옵션의 인수로 사용하여 시스템을 지정하고 해당 노드에 연결할 때 사용자 이름 및 암호 권한 부여가 필요한 경우 사용자 이름과 암호를 지정하십시오.

DAS 구성 매개변수의 목록을 보거나 인쇄하려면 GET ADMIN CONFIGURATION 명령을 사용하십시오. 관리 매개변수의 값을 변경하려면 UPDATE ADMIN CONFIGURATION 명령을 사용하십시오.

온라인으로 갱신할 수 있는 DAS 구성 매개변수에 대한 변경은 즉시 발생합니다. 기타 변경은 hdb2admin 명령으로 DAS를 재시작할 때 변경사항이 메모리에 로드된 후에만 적용됩니다.

오류가 발생하면 DAS 구성 파일이 변경되지 않습니다.

체크섬이 유효하지 않으면 DAS 구성 파일을 재설정할 수 없습니다. DAS 구성 파일을 수동으로 편집하며 적절한 명령을 사용하지 않을 경우 이러한 상황이 발생할 수 있습니다. 체크섬이 유효하지 않은 경우 DAS를 삭제 및 재작성하여 구성 파일을 재설정해야 합니다.

## 제 106 장 RESET ALERT CONFIGURATION

특정 오브젝트의 Health 표시기 설정값을 해당 오브젝트 유형의 현재 디폴트값으로 재 설정하거나 오브젝트 유형의 현재 디폴트 Health 표시기 설정값을 설치 디폴트값으로 재 설정합니다.

**중요사항:** Health Monitor가 버전 9.7에서 사용되지 않으므로 이 명령 또는 API는 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 *버전 9.7의 새로운 내용* 책에 있는 『Health Monitor는 사용되지 않음』 주제를 참조하십시오.

### 권한 부여

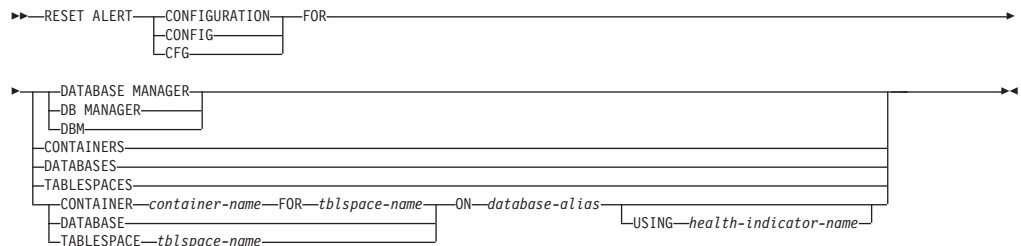
다음 중 하나가 필요합니다.

- *sysadm*
- *sysmaint*
- *sysctrl*

### 필수 연결

인스턴스. 명시적 접속은 필요하지 않습니다.

### 명령 구문



### 명령 매개변수

#### DATABASE MANAGER | DB MANAGER | DBM

데이터베이스 관리 프로그램에 대한 경보 설정값을 재설정합니다.

#### CONTAINERS

데이터베이스 관리 프로그램이 관리하는 모든 테이블 스페이스 컨테이너에 대한 경보 설정값을 설치 디폴트값으로 재설정합니다. 이들 설정값은 사용자 정의 설정값이 없는 모든 테이블 스페이스 컨테이너에 적용됩니다. 사용자 정의 설정은 `CONTAINER container-name FOR tbspace-name ON database-alias` 절을 사용하여 정의됩니다.

## DATABASES

데이터베이스 관리 프로그램이 관리하는 모든 데이터베이스에 대한 경보 설정값을 재설정합니다. 이들 설정값은 사용자 정의 설정값이 없는 모든 데이터베이스에 적용됩니다. 사용자 정의 설정값은 `DATABASE ON database-alias` 절을 사용하여 정의됩니다.

## TABLESPACES

데이터베이스 관리 프로그램이 관리하는 모든 테이블 스페이스에 대한 디폴트 경보 설정값을 설치 디폴트값으로 재설정합니다. 이들 설정값은 사용자 정의 설정값이 없는 모든 테이블 스페이스에 적용됩니다. 사용자 정의 설정은 `TABLESPACE tblspace-name ON database-alias` 절을 사용하여 정의됩니다.

## CONTAINER *container-name* FOR *tblspace-name* ON *database-alias*

`ON database-alias` 절을 사용하여 지정된 데이터베이스에서 `FOR tblspace-name` 절을 사용하여 지정된 테이블 스페이스의 경우 *container-name*이라는 테이블 스페이스 컨테이너의 경보 설정을 재설정합니다. 이 테이블 스페이스 컨테이너에 사용자 정의 설정값이 들어 있으면 이들 설정값을 제거하고 현재 테이블 스페이스 컨테이너 디폴트값을 사용합니다.

## DATABASE ON *database-alias*

`ON database-alias` 절을 사용하여 지정된 데이터베이스의 경보 설정을 재설정합니다. 이 데이터베이스가 사용자 정의 설정값을 가지고 있으면 이들 설정값을 제거하고 설치 디폴트값을 사용합니다.

## TABLESPACE *tblspace-name* ON *database-alias*

`ON database-alias` 절을 사용하여 지정된 데이터베이스에서 *tblspace-name*이라는 테이블 스페이스의 경보 설정을 재설정합니다. 이 테이블 스페이스가 사용자 정의 설정값을 가지고 있으면 이들 설정값을 제거하고 설치 디폴트값을 사용합니다.

## USING *health-indicator-name*

경보 구성이 재설정될 Health 표시기 세트를 지정합니다. Health 표시기 이름은 두 문자 오브젝트 ID 다음에 표시기가 측정하는 것을 설명하는 이름으로 구성됩니다. 예를 들면, 다음과 같습니다.

```
db.sort_privmem_util
```

이 옵션을 지정하지 않으면 지정된 오브젝트 또는 오브젝트 유형에 대한 모든 Health 표시기가 재설정됩니다.



사용자는 db2nodes.cfg에 정의된 대로 4개의 데이터베이스 파티션이 있는 파티션된 데이터베이스 환경을 갖습니다.

```
10 gilera 0
20 gilera 1
30 motobi 0
40 motobi 1
```

사용자가 인스턴스에 대한 SAMPLE 데이터베이스를 작성했습니다. SAMPLE의 카탈로그 파티션이 데이터베이스 파티션 번호 10에 있습니다. 사용자가 motobi 시스템에 로그인되었다고 가정하십시오.

다음 명령은 모든 데이터베이스 파티션에서 데이터베이스 SAMPLE의 모든 데이터베이스 구성 값을 재설정합니다.

```
db2 reset db cfg for sample
```

다음 명령은 데이터베이스 파티션 번호 30에서 데이터베이스 SAMPLE의 모든 데이터베이스 구성 값을 재설정합니다.

```
db2 reset db cfg for sample dbpartitionnum 30
```

또는

```
export DB2NODE=30
db2 reset db cfg for sample
```

## 사용 시 참고사항

데이터베이스 구성 매개변수 목록을 보거나 인쇄하려면 GET DATABASE CONFIGURATION 명령을 사용하십시오.

구성 가능 매개변수의 값을 변경하려면, UPDATE DATABASE CONFIGURATION 명령을 사용하십시오.

데이터베이스 구성 파일에 대한 변경사항은 메모리에 로드된 후에만 유효하게 됩니다. 이렇게 되려면 데이터베이스에서 모든 응용프로그램의 연결을 끊어야 합니다.

오류가 발생하면 데이터베이스 구성 파일이 변경되지 않습니다.

체크섬이 유효하지 않으면 데이터베이스 구성 파일을 재설정할 수 없습니다. 적절한 명령을 사용하지 않고 데이터베이스 구성 파일을 변경할 경우 이러한 일이 발생할 수 있습니다. 이러한 경우 데이터베이스를 리스토어하여 데이터베이스 구성 파일을 재설정해야 합니다.

RESET DATABASE CONFIGURATION 명령은 데이터베이스 구성 매개변수를 문서화된 디폴트 구성 값으로 재설정합니다. **Self\_tuning\_mem**은 파티션되지 않은 데이터베이스 환경에서는 ON으로 재설정되고 파티션된 데이터베이스 환경에서는 OFF로 재설정됩니다.



---

## 제 108 장 RESET DATABASE MANAGER CONFIGURATION

데이터베이스 관리 프로그램 구성 파일에 있는 매개변수를 시스템 디폴트값으로 재설정합니다. 값은 노드 유형별로 재설정됩니다.

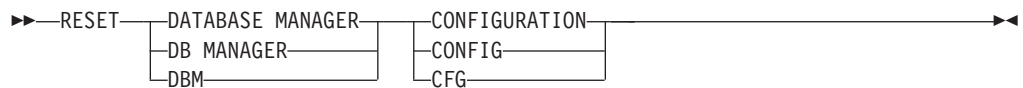
### 권한 부여

SYSADM

### 필수 연결

없음 또는 인스턴스. 로컬 데이터베이스 관리 프로그램 구성 작업을 수행하는 데는 인스턴스 접속이 필요하지 않지만 리모트 데이터베이스 관리 프로그램 구성 작업을 수행하는 데는 인스턴스 접속이 필요합니다. 리모트 인스턴스에 대한 데이터베이스 관리 프로그램 구성을 갱신하려면 먼저 이 인스턴스에 접속해야 합니다. 구성 매개변수를 온라인으로 갱신하는 경우에도 먼저 인스턴스에 접속해야 합니다.

### 명령 구문



### 명령 매개변수

없음

### 사용 시 참고사항

이 명령은 설치 프로그램에 의해 설정된 모든 매개변수를 재설정합니다. 이는 DB2를 재시작할 때 오류 메시지가 리턴되게 할 수 있습니다. 예를 들어, **svcename** 매개변수를 재설정할 경우 사용자는 DB2를 재시작하려고 할 때 SQL5043N 오류 메시지를 받습니다.

이 명령을 실행하기 전에 GET DATABASE MANAGER CONFIGURATION 명령의 출력을 파일에 저장하여 기존 설정값을 참조할 수 있도록 하십시오. UPDATE DATABASE MANAGER CONFIGURATION 명령을 사용하여 개별 설정값을 갱신할 수 있습니다.

설치 프로그램에 의해 설정된 **svcename** 매개변수를 사용자가 수정하는 것은 바람직하지 않습니다.

데이터베이스 관리 프로그램 구성 매개변수의 목록을 보거나 인쇄하려면 GET DATABASE MANAGER CONFIGURATION 명령을 사용하십시오. 구성 가능 매개변수의 값을 변경하려면, UPDATE DATABASE MANAGER CONFIGURATION 명령을 사용하십시오.

이들 매개변수에 대한 자세한 정보는 구성 매개변수의 요약 목록 및 개별 매개변수를 참조하십시오.

데이터베이스 관리 프로그램 구성 파일에 대한 일부 변경사항은 메모리로 로드된 후에만 유효하게 됩니다. 어떤 매개변수가 온라인으로 구성 가능하고 어떤 매개변수가 그렇지 않은지에 대한 자세한 정보는 구성 매개변수 요약을 참조하십시오. 즉시 재설정되지 않는 서버 구성 매개변수는 db2start 실행 중 재설정됩니다. 클라이언트 구성 매개변수의 경우, 매개변수는 다음 번에 응용프로그램을 재시작할 때 재설정됩니다. 클라이언트가 명령행 처리기일 경우 TERMINATE를 호출해야 합니다.

오류가 발생하면 데이터베이스 관리 프로그램 구성 파일이 변경되지 않습니다.

체크섬이 유효하지 않으면 데이터베이스 관리 프로그램 구성 파일을 재설정할 수 없습니다. 이는 구성 파일을 수동으로 편집하고 적절한 명령을 사용하지 않을 경우 발생할 수 있습니다. 체크섬이 유효하지 않으면, 인스턴스를 재작성해야 합니다.

## 제 109 장 RESET MONITOR

지정된 데이터베이스 또는 모든 활성 데이터베이스의 내부 데이터베이스 시스템 모니터 데이터 영역을 0으로 재설정합니다. 내부 데이터베이스 시스템 모니터 데이터 영역에는 데이터베이스에 연결된 모든 응용프로그램에 대한 데이터 영역과 데이터베이스 자체에 대한 데이터 영역이 포함됩니다.

### 권한 부여

다음 중 하나가 필요합니다.

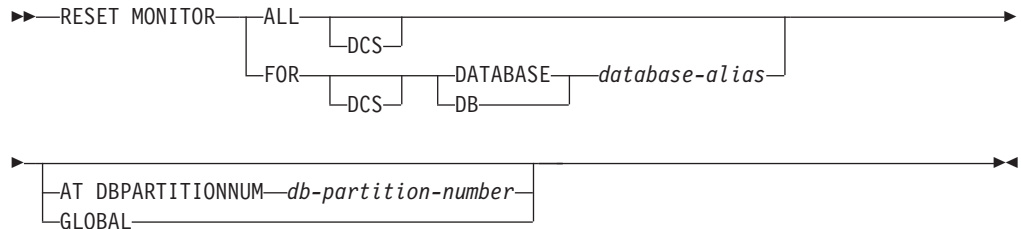
- *sysadm*
- *sysctrl*
- *sysmaint*
- *sysmon*

### 필수 연결

인스턴스. 인스턴스 첨부이 없는 경우 디폴트 인스턴스 첨부이 작성됩니다.

리모트 인스턴스(또는 다른 로컬 인스턴스)에 대한 모니터 스위치를 재설정하려면 먼저 해당 인스턴스에 접속해야 합니다.

### 명령 구문



### 명령 매개변수

**ALL** 이 옵션은 내부 카운터가 모든 데이터베이스에 대해 재설정되어야 함을 표시합니다.

**FOR DATABASE** *database-alias*

이 옵션에는 별명 *database-alias*를 갖는 데이터베이스만 내부 카운터가 재설정되어야 함을 표시합니다.

**DCS** 지정된 절에 따라서 이 키워드는 다음의 내부 카운터를 재설정합니다.

- 모든 DCS 데이터베이스

- 특정 DCS 데이터베이스.

#### **AT DBPARTITIONNUM** *db-partition-number*

모니터 스위치의 상태가 표시될 데이터베이스 파티션을 지정합니다.

#### **GLOBAL**

파티션된 데이터베이스 환경의 모든 데이터베이스 파티션에 대한 집계 결과를 리턴합니다.

### **사용 시 참고사항**

각 프로세스(접속)는 모니터 데이터의 개인용 뷰를 갖습니다. 한 사용자가 모니터 스위치를 재설정하거나 끄면 다른 사용자는 영향을 받지 않습니다. 모니터 스위치 구성 매개변수의 설정을 변경하여 모니터 스위치에 대한 전역 변경을 작성하십시오.

ALL이 지정되는 경우 일부 데이터베이스 관리 프로그램 정보도 재설정되어 리턴된 데이터의 일관성을 유지하고 일부 데이터베이스 파티션 레벨 카운터가 재설정됩니다.

### **호환성**

버전 8 이전 버전과의 호환성:

- DBPARTITIONNUM에 키워드 NODE를 대체할 수 있습니다.

---

## 제 110 장 RESTART DATABASE

비정상적으로 종료했고 불일치 상태에 남은 데이터베이스를 재시작합니다. RESTART DATABASE를 성공적으로 완료하면 사용자에게 CONNECT 특권이 있는 경우 응용 프로그램은 데이터베이스에 연결된 채로 있습니다.

### 범위

이 명령은 명령이 실행되는 노드에만 영향을 줍니다.

### 권한 부여

없음

### 필수 연결

이 명령은 데이터베이스 연결을 설정합니다.

### 명령 구문

```
➤ RESTART DATABASE database-alias
      DB
➤ USER username USING password
➤ DROP PENDING TABLESPACES (tablespace-name) WRITE RESUME
```

### 명령 매개변수

#### **DATABASE** *database-alias*

재시작할 데이터베이스를 식별합니다.

#### **USER** *username*

데이터베이스가 재시작될 사용자 이름을 식별합니다.

#### **USING** *password*

*username*을 인증하는 데 사용되는 암호입니다. 암호를 생략하면 사용자가 이를 입력하도록 프롬프트가 표시됩니다.

#### **DROP PENDING TABLESPACES** *tablespace-name*

테이블 스페이스 컨테이너 문제점에 직면하는 경우에도 데이터베이스 재시작 조 작이 성공적으로 완료되어야 함을 지정합니다.

재시작 프로세스 중에 지정된 테이블 스페이스에 대한 컨테이너에서 문제점이 발생하는 경우 대응하는 테이블 스페이스는 재시작 조작 후에 사용 가능하지 않습니다(삭제 보류 상태에 있음). 테이블 스페이스가 삭제 보류 상태에 있는 경우 유일하게 가능한 조치는 테이블 스페이스를 삭제(drop)하는 것입니다.

순환 로그의 경우 문제가 있는 테이블 스페이스는 재시작 실패를 유발합니다. 데이터베이스 재시작 조작이 컨테이너 문제점 때문에 실패하는 경우 문제가 있는 테이블 스페이스 이름의 목록은 관리 통지 로그에서 찾을 수 있습니다. 데이터베이스에 시스템 임시 테이블 스페이스가 하나만 있고 삭제 보류 상태에 있는 경우 데이터베이스 재시작 조작이 성공한 직후에 새 시스템 임시 테이블 스페이스를 작성해야 합니다.

## WRITE RESUME

입출력 쓰기가 일시중단된 동안 실패한 데이터베이스에서 데이터베이스 재시작을 강제로 실행할 수 있습니다. 응급 복구를 수행하기 전에 이 옵션은 데이터베이스의 모든 테이블 스페이스에서 SUSPEND\_WRITE 상태를 제거하여 입출력 쓰기를 재개합니다.

WRITE RESUME 옵션은 입출력 쓰기를 일시중단하는 데 사용된 연결이 현재 정지 중이고 모든 후속 연속적인 연결 시도도 중지 중인 경우에도 사용할 수 있습니다. 이 상황에서 사용할 때 RESTART DATABASE는 응급 복구를 수행하지 않고 데이터베이스에 대한 입출력 쓰기를 재개합니다. WRITE RESUME 옵션을 갖는 RESTART DATABASE는 사용자가 데이터베이스 손상 후에 사용할 때만 응급 복구를 수행합니다. WRITE RESUME 매개변수는 미러된 데이터베이스가 아니라 기본 데이터베이스에만 적용할 수 있습니다.

## 사용 시 참고사항

데이터베이스에 연결 시도 중에 데이터베이스를 재시작해야 함을 표시하는 오류 메시지가 리턴되는 경우 이 명령을 실행하십시오. 이 조치는 이 데이터베이스를 갖는 이전 세션이 비정상적으로(예를 들어 전원 장애로 인해) 종료한 경우에만 발생합니다.

파티션된 데이터베이스 시스템에서 인다우트(Indoubt) 트랜잭션을 해결하려면 아래 예에서와 같이 모든 노드에서 RESTART DATABASE 명령을 실행해야 합니다.

```
db2_all "db2 restart database database-alias"
```

데이터베이스가 MPP 시스템의 단일 노드에서만 재시작되는 경우 후속 데이터베이스 쿼리에서 데이터베이스를 재시작해야 함을 표시하는 메시지가 리턴될 수 있습니다. 이것은 쿼리가 의존하는 노드의 데이터베이스 파티션도 재시작되어야 하기 때문에 발생합니다. 모든 노드에서 데이터베이스를 재시작하면 문제점이 해결됩니다.

---

## 제 111 장 RESTORE DATABASE

RESTORE DATABASE 명령은 DB2 백업 유틸리티를 사용하여 백업한 손상된 데이터베이스를 다시 작성합니다. 리스토어된 데이터베이스는 백업 사본이 작성될 때의 상태와 동일한 상태가 됩니다. 이 유틸리티는 다른 이미지로 데이터베이스를 겹쳐쓸 수 있고 백업 사본을 새 데이터베이스로 리스토어할 수도 있습니다.

다양한 운영 체제 및 하드웨어 플랫폼 사이에 DB2 데이터베이스 시스템에서 지원되는 리스토어 조작에 관한 정보는 *Data Recovery and High Availability Guide and Reference*에서 『Backup and restore operations between different operating systems and hardware platforms』를 참조하십시오.

리스토어 유틸리티를 사용하여 DB2 Universal Database 버전 8, DB2 버전 9.1 또는 DB2 버전 9.5에서 백업한 백업 이미지를 DB2 버전 9.7에서 리스토어할 수도 있습니다. 데이터베이스 업그레이드가 필요한 경우 리스토어 조작 종료 시 자동으로 호출됩니다.

백업 작업 시 데이터베이스를 롤 포워드 복구에 사용할 수 있는 경우 리스토어 조작을 완료한 후 롤 포워드 유틸리티를 호출하여 데이터베이스를 이전 상태로 가져올 수 있습니다.

이 유틸리티는 또한 테이블 스페이스 레벨 백업을 리스토어할 수도 있습니다.

중분 이미지와 이전 캡처에서 차이점만 캡처하는 이미지(『델타 이미지』)는 운영 체제나 단어 크기(32비트 또는 64비트)가 다를 때 리스토어할 수 없습니다.

하나의 환경에서 다른 환경으로의 성공적인 리스토어 조작 후에는 중분이 아닌 백업을 가져올 때까지 중분 또는 델타 백업이 허용되지 않습니다. (이는 동일 환경에서의 리스토어 조작 다음의 제한사항이 아닙니다.)

하나의 환경에서 다른 환경으로의 리스토어 조작이 성공해도, 몇 가지 고려할 사항이 있습니다. 패키지는 사용 이전에 리바인드해야 합니다(BIND 명령, REBIND 명령 또는 db2rbind 유틸리티를 사용하여). SQL 프로시저는 삭제하고 다시 작성해야 합니다. 그리고 모든 외부 라이브러리는 새 플랫폼에서 재빌드해야 합니다. (동일한 환경으로 리스토어할 경우에는 이 사항을 고려하지 않아도 됩니다.)

기존 데이터베이스와 기존 컨테이너에 대해 실행되는 리스토어 조작은 동일한 컨테이너 및 테이블 스페이스 맵을 재사용합니다.

새 데이터베이스에 대해 실행되는 리스토어 조작용 모든 컨테이너를 다시 획득하고 최적화된 테이블 스페이스 맵을 재빌드합니다. 새 데이터베이스에 대해 실행되는 리스토어 조작용 모든 컨테이너를 다시 획득하고 최적화된 테이블 스페이스 맵을 재빌드합니다.

## 범위

이 명령은 명령이 실행되는 노드에만 영향을 미칩니다.

## 권한 부여

기존 데이터베이스를 리스토어하려면 다음 중 하나가 있어야 합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*

새 데이터베이스로 리스토어하려면 다음 중 하나가 있어야 합니다.

- *sysadm*
- *sysctrl*

## 필수 연결

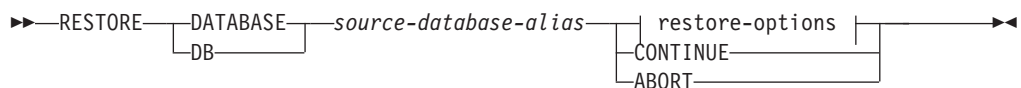
필수 연결은 리스토어 조치 유형에 따라 다릅니다.

- 기존 데이터베이스로 리스토어하려면 데이터베이스에 연결해야 합니다. 이 명령은 지정된 데이터베이스에 대한 독점 연결을 자동으로 설정합니다.
- 새 데이터베이스로 리스토어하려면 인스턴스 및 데이터베이스에 연결해야 합니다. 데이터베이스를 작성하려면 인스턴스에 접속해야 합니다.

현재 인스턴스와 다른 인스턴스에서 새 데이터베이스로 리스토어하려면, 먼저 새 데이터베이스가 상주할 인스턴스에 접속해야 합니다. 새 인스턴스는 로컬 또는 리모트 상태일 수 있습니다. 현재 인스턴스는 DB2INSTANCE 환경 변수 값으로 정의됩니다.

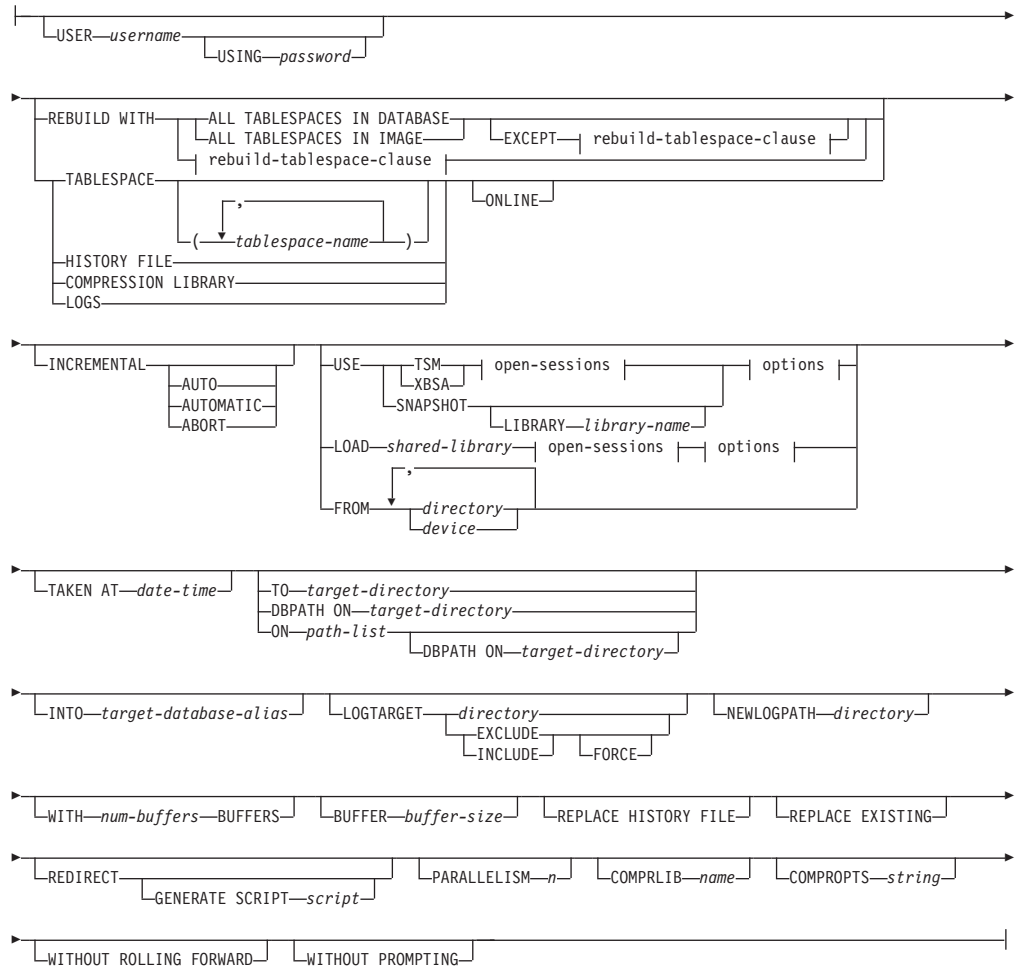
- 스냅샷 리스토어의 경우 인스턴스 및 데이터베이스에 연결해야 합니다.

## 명령 구문

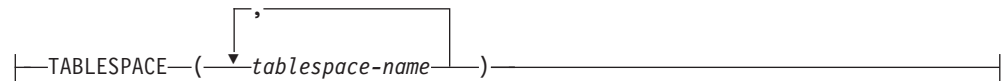


**restore-options:**

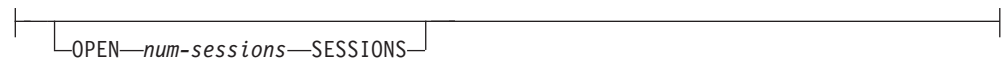




**rebuild-tablespace-clause:**



**open-sessions:**



**옵션:**



**명령 매개변수**

**DATABASE** *source-database-alias*

백업을 가져온 소스 데이터베이스의 별명.

## CONTINUE

컨테이너가 재정의되었으며, 경로 재지정된 리스토어 조작의 최종 단계를 수행해야 함을 지정합니다.

## ABORT

이 매개변수는

- 경로 재지정된 리스토어 조작을 중지합니다. 이는 하나 이상의 단계를 반복해야 하는 오류가 발생할 때 유용합니다. ABORT 옵션이 있는 RESTORE DATABASE를 발행하면, REDIRECT 옵션이 있는 RESTORE DATABASE를 포함하여 경로 재지정된 리스토어 조작의 각 단계를 반복해야 합니다.
- 완료 이전에 증분 리스토어 조작을 종료합니다.

## USER *username*

데이터베이스가 리스토어되는 사용자 이름을 식별합니다.

## USING *password*

사용자 이름을 인증하는 데 사용되는 암호. 암호를 생략하면 사용자가 이를 입력하도록 프롬프트가 표시됩니다.

## REBUILD WITH ALL TABLESPACES IN DATABASE

이미지가 리스토어될 때 데이터베이스에 알려진 모든 테이블 스페이스와 함께 데이터베이스를 리스토어합니다. 이 리스토어는 데이터베이스가 이미 존재하는 경우 겹쳐씹니다.

## REBUILD WITH ALL TABLESPACES IN DATABASE EXCEPT

*rebuild-tablespace-clause*

목록에 지정된 테이블 스페이스를 제외하고, 이미지가 리스토어될 때 데이터베이스에 알려진 모든 테이블 스페이스와 함께 데이터베이스를 리스토어합니다. 이 리스토어는 데이터베이스가 이미 존재하는 경우 겹쳐씹니다.

## REBUILD WITH ALL TABLESPACES IN IMAGE

리스토어 중인 이미지의 테이블 스페이스만 데이터베이스와 함께 리스토어합니다. 이 리스토어는 데이터베이스가 이미 존재하는 경우 겹쳐씹니다.

## REBUILD WITH ALL TABLESPACES IN IMAGE EXCEPT

*rebuild-tablespace-clause*

목록에 지정된 테이블 스페이스를 제외하고, 리스토어 중인 테이블 스페이스만 데이터베이스와 함께 리스토어합니다. 이 리스토어는 데이터베이스가 이미 존재하는 경우 겹쳐씹니다.

## REBUILD WITH *rebuild-tablespace-clause*

지정된 테이블 스페이스 목록만 데이터베이스와 함께 리스토어합니다. 이 리스토어는 데이터베이스가 이미 존재하는 경우 겹쳐씹니다.

**TABLESPACE** *tablespace-name*

리스토어될 테이블 스페이스를 지정하는 데 사용되는 이름 목록.

**ONLINE**

이 키워드는 테이블 스페이스 레벨 리스토어 작업을 수행하는 경우에만 적용되며, 백업 이미지가 온라인에서 리스토어될 수 있도록 허용하기 위해 지정합니다. 이는 백업 이미지가 리스토어되는 동안 다른 에이전트가 데이터베이스에 연결할 수 있으며, 지정된 테이블 스페이스를 리스토어하는 중에 다른 테이블 스페이스의 데이터를 사용할 수 있음을 의미합니다.

**HISTORY FILE**

이 키워드는 백업 이미지에서 실행기록 파일만 리스토어할 때 지정됩니다.

**COMPRESSION LIBRARY**

이 키워드는 백업 이미지에서 압축 라이브러리만 리스토어할 때 지정됩니다. 오브젝트가 백업 이미지에 있는 경우 데이터베이스 디렉토리로 리스토어됩니다. 오브젝트가 백업 이미지에 없는 경우 리스토어 작업이 실패합니다.

**LOGS**

이 키워드는 백업 이미지에 포함된 로그 파일 세트만 리스토어할 때 지정됩니다. 백업 이미지에 로그 파일이 없는 경우 리스토어 작업이 실패합니다. 이 옵션을 지정하면, LOGTARGET 옵션도 지정해야 합니다.

**INCREMENTAL**

추가 매개변수 없이, INCREMENTAL은 수동의 누적된 리스토어 작업을 지정합니다. 수동 리스토어 중, 사용자는 리스토어에 포함되는 이미지마다 수동으로 각각의 리스토어 명령을 발행해야 합니다. 마지막, 첫 번째, 두 번째, 세 번째, 그리고 마지막 이미지까지, 이 순서로 수행해야 합니다.

**INCREMENTAL AUTOMATIC/AUTO**

자동 누적 리스토어 작업을 지정합니다.

**INCREMENTAL ABORT**

진행 중인 수동 누적 리스토어 작업의 중단을 지정합니다.

**USE**

**TSM** Tivoli Storage Manager에서 관리되는 출력에서 데이터베이스가 리스토어됨을 지정합니다.

**XBSA** XBSA 인터페이스가 사용되도록 지정합니다. 백업 서비스 API(XBSA)는 백업 또는 아카이브용으로 데이터 스토리지 관리를 필요로 하는 응용프로그램 또는 기능에 대한 개방 API입니다.

**SNAPSHOT**

스냅샷 백업에서 데이터가 리스토어되도록 지정합니다.

다음 매개변수가 있는 SNAPSHOT 매개변수를 사용할 수 없습니다.

- INCREMENTAL
- TO
- ON
- DBPATH ON
- INTO
- NEWLOGPATH
- WITH *num-buffers* BUFFERS
- BUFFER
- REDIRECT
- REPLACE HISTORY FILE
- COMPRESSION LIBRARY
- PARALLELISM
- COMPRLIB
- OPEN *num-sessions* SESSIONS
- HISTORY FILE
- LOGS

또한 테이블 스페이스 목록을 수반하는 리스토어 조작(REBUILD WITH 옵션이 포함된)에서는 SNAPSHOT 매개변수를 사용할 수 없습니다.

스냅샷 백업 이미지에서 데이터를 리스토어할 때 디폴트 동작은 시간 소인이 제공되지 않는 경우 모든 컨테이너, 로컬 볼륨 디렉토리, 데이터베이스 경로(DBPATH), 1차 로그 및 가장 최신 스냅샷 백업의 미러 로그 경로를 포함하여 데이터베이스를 구성하는 모든 경로의 FULL DATABASE OFFLINE 리스토어입니다(INCLUDE LOGS는 EXCLUDE LOGS가 명시적으로 언급된 경우를 제외하고는 모든 스냅샷 백업의 디폴트임). 시간소인이 제공되는 경우에는 스냅샷 백업 이미지가 리스토어됩니다.

#### **LIBRARY** *library-name*

IBM Data Server로의 통합은 다음 스토리지 하드웨어에 대한 DB2 ACS API 드라이버입니다.

- IBM TotalStorage® SAN Volume Controller
- IBM Enterprise Storage Server® Model 800
- IBM System Storage™ DS6000™
- IBM System Storage DS8000®
- IBM System Storage N Series

- NetApp V-series
- NetApp FAS

다른 스토리지 하드웨어가 있으며 해당 스토리지 하드웨어의 DB2 ACS API 드라이버가 있는 경우, LIBRARY 매개변수를 사용하여 DB2 ACS API 드라이버를 지정할 수 있습니다. LIBRARY 매개변수의 값은 완전한 라이브러리 파일 이름입니다.

## OPTIONS

### *"options-string"*

리스트어 조작에 사용되는 옵션을 지정합니다. 문자열은 큰따옴표 없이 입력된 그대로 정확히 DB2 ACS API 드라이버로 전달됩니다. **VENDOROPT** 데이터베이스 구성 매개변수를 사용하여 스냅샷 리스트어 조작을 위한 벤더 특정 옵션을 지정할 수 없습니다. 대신 리스트어 유틸리티의 **OPTIONS** 매개변수를 사용해야 합니다.

### *@ file-name*

리스트어 조작에 사용될 옵션이 DB2 서버에 있는 파일에 들어 있음을 지정합니다. 이 문자열은 벤더 지원 라이브러리로 패스됩니다. 파일은 완전한 파일 이름이어야 합니다.

## **OPEN** *num-sessions* **SESSIONS**

TSM 또는 벤더 제품과 함께 사용되는 입출력 세션 수를 지정합니다.

## **FROM** *directory/device*

백업 이미지가 있는 디렉토리 또는 디바이스의 완전한 경로 이름. USE TSM, FROM 및 LOAD가 생략된 경우 디폴트 값은 클라이언트 컴퓨터의 현재 작업 디렉토리입니다. 이 목표 디렉토리 또는 디바이스는 목표 서버/인스턴스에 있어야 합니다.

몇 개의 항목이 지정되고 마지막 항목이 테이프 디바이스이면 사용자가 다른 테이프를 입력하도록 프롬프트됩니다. 유효한 응답은 다음과 같습니다.

- c** 계속. 경고 메시지를 생성한 디바이스를 사용하여 계속합니다(예를 들어, 새로운 테이프가 마운트된 경우 계속함).
- d** 디바이스 종료. 경고 메시지를 생성한 디바이스만 사용을 중지합니다(예를 들어, 더 이상 테이프가 없을 때 종료함).
- t** 종료. 유틸리티에서 요청된 일부 조치 수행에 실패한 후에는 리스트어 조작을 중단합니다.

## **LOAD** *shared-library*

사용될 벤더 백업 및 리스트어 I/O 기능이 들어 있는 공유 라이브러리 이름

(Windows 운영 체제의 DLL). 이름에는 전체 경로가 포함될 수 있습니다. 전체 경로가 제공되지 않을 경우 디폴트 값은 User Exit 프로그램이 있는 경로입니다.

**TAKEN AT<sup>®</sup>** *date-time*

데이터베이스 백업 이미지의 시간소인. 시간소인은 백업 작업이 완료된 후에 표시되며, 백업 이미지에 대한 경로 이름의 일부입니다. *yyyymmddhhmmss* 양식으로 지정됩니다. 부분 시간소인을 지정할 수도 있습니다. 예를 들어, 시간소인이 20021001010101 및 20021002010101인 두 개의 다른 백업 이미지가 존재하는 경우 20021002를 지정하면 시간소인이 20021002010101인 이미지가 사용됩니다. 이 매개변수의 값을 지정하지 않은 경우 소스 미디어에는 단 하나의 백업 이미지만 있어야 합니다.

**TO** *target-directory*

이 매개변수는 목표 데이터베이스 디렉토리를 언급합니다. 유틸리티가 기존 데이터베이스로 리스토어 중인 경우에는 이 매개변수는 무시됩니다. 지정하는 드라이브 및 디렉토리는 로컬이어야 합니다. 백업 이미지에 자동 스토리지가 사용 가능한 데이터베이스가 포함되면, 데이터베이스 디렉토리는 변경되고 데이터베이스에 연관된 스토리지 경로는 변경되지 않습니다.

**DBPATH ON** *target-directory*

이 매개변수는 목표 데이터베이스 디렉토리를 언급합니다. 유틸리티가 기존 데이터베이스로 리스토어 중인 경우에는 이 매개변수는 무시됩니다. 지정하는 드라이브 및 디렉토리는 로컬이어야 합니다. 백업 이미지에 자동 스토리지가 사용 가능하고 ON 매개변수가 지정되지 않은 데이터베이스가 포함되면, 이 매개변수는 TP 매개변수와 동의어로, 데이터베이스 디렉토리만 변경되고 데이터베이스에 연관된 스토리지 경로는 변경되지 않습니다.

**ON** *path-list*

이 매개변수는 자동 스토리지 데이터베이스와 연관된 스토리지 경로를 재정의합니다. 자동 스토리지에 대해 사용 가능하도록 설정되지 않은 데이터베이스에 이 매개변수를 사용하면 오류가 발생합니다(SQL20321N). 백업 이미지에 정의된 기존 스토리지 경로는 더 이상 사용되지 않으며 자동 스토리지 테이블 스페이스의 경로는 자동으로 새 경로로 재지정됩니다. 자동 스토리지 데이터베이스에 대해 이 매개변수를 지정하지 않으면, 스토리지 경로는 백업 이미지에 정의된 대로 유지됩니다.

하나 이상의 경로를 지정할 수 있으며 각각은 쉼표로 구분됩니다. 각 경로에는 절대 경로 이름이 있어야 하며 로컬에 있어야 합니다. 데이터베이스가 아직 디스크에 존재하지 않고 DBPATH ON 매개변수를 지정하지 않은 경우, 목표 데이터베이스 디렉토리로 첫 번째 경로가 사용됩니다.

다중 파티션 데이터베이스의 경우 ON *path-list* 옵션은 카탈로그 파티션에서만 지정할 수 있습니다. 카탈로그 파티션은 ON 옵션이 사용될 때 다른 파티션이

리스토어되기 전에 리스토어해야 합니다. 새 스토리지 경로를 사용하는 카탈로그 파티션의 리스토어에서는 카탈로그에 없는 모든 노드가 RESTORE\_PENDING 상태가 됩니다. 카탈로그에 없는 노드는 리스토어 명령에 ON절을 지정하지 않고 병렬로 리스토어될 수 있습니다.

일반적으로 멀티파티션 데이터베이스의 각 파티션에 대해 동일한 스토리지 경로를 사용해야 하며 RESTORE DATABASE 명령 실행 이전에 그 경로가 존재해야 합니다. 한 가지 예외는 데이터베이스 파티션 표현식을 스토리지 경로 내에서 사용하는 경우입니다. 이를 수행하면 데이터베이스 파티션 번호를 스토리지 경로에 반영하여 결과로 생성되는 경로 이름이 파티션마다 다릅니다.

인수 " \$N" ([blank]\$N)을 사용하여 데이터베이스 파티션 표현식을 나타낼 수 있습니다. 데이터베이스 파티션 표현식은 스토리지 경로의 어디에서든 사용할 수 있으며 데이터베이스 파티션 표현식을 여러 개 지정할 수도 있습니다. 데이터베이스 파티션 표현식은 스페이스 문자로 끝내십시오. 스페이스 다음에 오는 문자는 데이터베이스 파티션 표현식을 평가한 후 스토리지 경로에 추가됩니다. 스토리지 경로에서 데이터베이스 파티션 표현식 뒤에 스페이스 문자가 없으면, 나머지 문자열은 표현식의 일부로 간주됩니다. 인수는 다음 형식 중 한 가지 형식으로만 사용할 수 있습니다.

표 36. . 연산자는 왼쪽에서 오른쪽으로 계산됩니다. %는 모듈러스 연산자를 표시합니다. 예에서 데이터베이스 파티션 번호는 10으로 가정합니다.

구문	예 :	값
[blank]\$N	" \$N"	10
[blank]\$N+[number]	" \$N+100"	110
[blank]\$N%[number]	" \$N%5"	0
[blank]\$N+[number]%[number]	" \$N+1%5"	1
[blank]\$N%[number]+[number]	" \$N%4+2"	4

<sup>a</sup> %는 모듈러스입니다.

### INTO *target-database-alias*

목표 데이터베이스 별명. 목표 데이터베이스가 존재하지 않으면 작성됩니다.

데이터베이스 백업을 기존 데이터베이스로 리스토어할 때, 리스토어된 데이터베이스는 기존 데이터베이스의 별명 및 데이터베이스 이름을 상속합니다. 데이터베이스 백업을 기존에 없는 데이터베이스로 리스토어하는 경우에는 사용자가 지정하는 별명 및 데이터베이스 이름의 새 데이터베이스가 작성됩니다. 이 새 데이터베이스 이름은 리스토어하는 시스템에서 고유해야 합니다.

### LOGTARGET *directory*

비스냅샷 리스토어:

백업 이미지에서 로그 파일을 추출하는 목표 디렉토리로 사용될, 데이터베이스 서버에 있는 기존 디렉토리의 절대 경로 이름. 이 옵션을 지정하는 경우 백업

이미지에 포함된 모든 로그 파일이 목표 디렉토리로 추출됩니다. 이 옵션을 지정하지 않으면 백업 이미지에 포함된 로그 파일은 추출되지 않습니다. 백업 이미지에서 로그 파일만 추출하려면 LOGS 옵션을 지정하십시오.

스냅샷 리스토어:

#### **INCLUDE**

스냅샷 이미지에서 로그 디렉토리 볼륨을 리스토어합니다. 이 옵션을 지정하고 백업 이미지에 로그 디렉토리가 포함된 경우 해당 디렉토리가 리스토어됩니다. 디스크의 기존 로그 디렉토리 및 로그 파일은 백업 이미지의 로그 디렉토리와 충돌하지 않으면 그대로 유지됩니다. 디스크의 기존 로그 디렉토리가 백업 이미지의 로그 디렉토리와 충돌하는 경우에는 오류가 리턴됩니다.

#### **EXCLUDE**

로그 디렉토리 볼륨을 리스토어하지 않습니다. 이 옵션을 지정하면 로그 디렉토리가 백업 이미지에서 리스토어되지 않습니다. 디스크의 기존 로그 디렉토리 및 로그 파일은 백업 이미지의 로그 디렉토리와 충돌하지 않으면 그대로 유지됩니다. 이로 인해 데이터베이스에 속한 경로가 리스토어되고 로그 디렉토리가 내재적으로 리스토어되어 로그 디렉토리가 겹쳐지면 오류가 리턴됩니다.

#### **FORCE**

스냅샷 이미지를 리스토어할 때 현재 데이터베이스의 기존 로그 디렉토리가 겹쳐쓰여지고 교체되도록 허용합니다. 이 옵션을 사용하지 않으면, 스냅샷 이미지의 로그 디렉토리와 충돌하는 디스크의 기존 로그 디렉토리 및 로그 파일로 인해 리스토어가 실패합니다. 리스토어에서 기존 로그 디렉토리를 겹쳐쓰고 교체할 수 있음을 표시하려면 이 옵션을 사용하십시오.

주: 이 옵션을 사용할 때는 주의해야 하며 항상 복구에 필요할 수 있는 모든 로그를 백업하고 아카이브했는지 확인해야 합니다.

주: LOGTARGET이 지정된 비스냅샷 리스토어가 아닌 경우 디폴트 LOGTARGET 디렉토리는 LOGTARGET EXCLUDE입니다.

#### **NEWLOGPATH** *directory*

리스토어 조작 후 활성 로그 파일에 사용될 디렉토리의 절대 경로 이름. 이 매개변수에는 매개변수가 지정된 리스토어 조작에만 적용된다는 점을 제외하고, **newlogpath** 데이터베이스 구성 매개변수와 같은 기능이 있습니다. 이 매개변수는 백업 이미지의 로그 경로가 리스토어 조작 후 사용에 적합하지 않을 때 (예를 들어, 경로가 더 이상 유효하지 않거나 다른 데이터베이스에서 사용 중인 경우) 사용할 수 있습니다.



### **WITH *num-buffers* BUFFERS**

사용될 버퍼 수. DB2 데이터베이스 시스템은 사용자가 명시적으로 값을 입력하지 않는 경우를 제외하고는 자동으로 이 매개변수에 대한 최적 값을 선택합니다. 여러 개의 소스를 읽거나 PARALLELISM 값이 증가된 경우 성능을 개선하려면 더 많은 버퍼를 사용할 수 있습니다.

### **BUFFER *buffer-size***

리스트어 조작에 사용되는 버퍼의 크기(페이지 수). DB2 데이터베이스 시스템은 사용자가 명시적으로 값을 입력하지 않는 경우를 제외하고는 자동으로 이 매개변수에 대한 최적 값을 선택합니다. 이 매개변수의 최소값은 8페이지입니다.

리스트어 버퍼 크기는 백업 작업 중에 지정된 백업 버퍼 크기의 양의 배수입니다. 올바르지 않은 버퍼 크기를 지정하면 버퍼는 승인할 수 있는 가장 작은 크기로 할당됩니다.

### **REPLACE HISTORY FILE**

리스트어 조작이 디스크의 실행기록 파일을 백업 이미지의 실행기록 파일로 교체해야 함을 지정합니다.

### **REPLACE EXISTING**

목표 데이터베이스 별명과 같은 별명의 데이터베이스가 이미 있는 경우, 이 매개변수는 리스트어 유틸리티가 기존 데이터베이스를 리스트어된 데이터베이스로 교체하도록 지정합니다. 이는 리스트어 유틸리티를 호출하는 스크립트에 유용합니다. 명령행 처리기는 기존 데이터베이스의 삭제를 검증할 것을 사용자에게 프롬프트하지 않기 때문입니다. WITHOUT PROMPTING 매개변수를 지정하는 경우 REPLACE EXISTING를 지정할 필요가 없지만, 그러면 일반적으로 사용자 개입이 필요한 이벤트가 발생하는 경우에 작업이 실패합니다.

### **REDIRECT**

경로 재지정된 리스트어 조작을 지정합니다. 경로 재지정된 리스트어 조작을 완료하려면 이 명령 다음에 하나 이상의 SET TABLESPACE CONTAINERS 명령과, CONTINUE 옵션이 있는 RESTORE DATABASE 명령이 차례로 있어야 합니다. 경로 재지정된 단일 리스트어 조작과 연관된 모든 명령은 동일한 창 또는 CLP 세션에서 호출해야 합니다.

### **GENERATE SCRIPT *script***

지정된 파일 이름으로 리스트어 경로 재지정 스크립트를 작성합니다. 스크립트 이름은 상대적 또는 절대적일 수 있으며 스크립트는 클라이언트 측에서 생성됩니다. 클라이언트 측에서 파일을 작성할 수 없으면 오류 메시지(SQL9304N)가 리턴됩니다. 파일이 이미 존재하면 겹쳐씹니다. 추가 사용 정보는 아래의 예를 참조하십시오.

### **WITHOUT ROLLING FORWARD**

데이터베이스가 성공적으로 리스트어된 후에 롤 포워드 보류 상태가 되지 않음을 지정합니다.

리스트어 조작이 성공한 후에 데이터베이스가 롤 포워드 보류 상태가 되면 데이터베이스를 다시 사용하기 전에 ROLLFORWARD 명령을 호출해야 합니다. 온라인 백업 이미지에서 리스트어할 때 이 옵션을 지정한 경우 오류 SQL2537N 이 리턴됩니다.

백업 이미지가 복구 가능한 데이터베이스의 이미지인 경우 REBUILD 옵션과 함께 WITHOUT ROLLING FORWARD를 지정할 수 없습니다.

#### **PARALLELISM** *n*

리스트어 조작 중에 작성될 버퍼 조작자 수를 지정합니다. DB2 데이터베이스 시스템은 사용자가 명시적으로 값을 입력하지 않는 경우를 제외하고는 자동으로 이 매개변수에 대한 최적 값을 선택합니다.

#### **COMPRLIB** *name*

압축 해제를 수행하기 위해 사용할 라이브러리의 이름을 표시합니다(예: Windows의 경우 db2compr.dll, Linux/UNIX 시스템의 경우 libdb2compr.so). 이 이름은 서버에 있는 파일을 나타내는 완전한 경로이어야 합니다. 이 매개변수를 지정하지 않으면 DB2가 이미지에 저장된 라이브러리 사용을 시도합니다. 백업이 압축되지 않은 경우 이 매개변수 값은 무시됩니다. 지정된 라이브러리를 로드할 수 없으면 리스트어 조작이 실패합니다.

#### **COMPROPTS** *string*

압축 해제 라이브러리의 초기화 루틴으로 전달되는 2진 데이터 블록을 설명합니다. DB2 데이터베이스 시스템은 이 문자열을 클라이언트에서 서버로 직접 전달하므로 모든 바이트 리버설 또는 코드 페이지 변환 문제는 압축 해제 라이브러리가 직접 처리합니다. 데이터 블록의 첫 번째 문자가 『@』인 경우 DB2 데이터베이스 시스템은 나머지 데이터를 서버에 있는 파일의 이름으로 해석합니다. 그런 다음 DB2 데이터베이스 시스템은 *string*의 콘텐츠를 이 파일의 콘텐츠로 바꾸고 대신 초기화 루틴으로 새 값을 전달합니다. 문자열의 최대 길이는 1 024바이트입니다.

#### **WITHOUT PROMPTING**

리스트어 조작이 의도하지 않게 실행됨을 지정합니다. 보통 사용자가 개입해야 하는 조치는 오류 메시지를 표시합니다. 분리성 매체(예: 테이프 또는 디스켓)를 사용할 경우에는 이 옵션을 지정해도 디바이스 종료 시 사용자에게 프롬프트가 표시됩니다.

#### **예:**

1. 다음 예에서 데이터베이스 WSDb는 번호가 0부터 3까지인 모두 네 개의 데이터베이스 파티션에 정의됩니다. 경로 /dev3/backup은 모든 데이터베이스 파티션에서 액세스할 수 있습니다. 다음 오프라인 백업 이미지는 /dev3/backup에서 사용할 수 있습니다.

```
wsdb.0.db2inst1.NODE0000.CATN0000.20020331234149.001
wsdb.0.db2inst1.NODE0001.CATN0000.20020331234427.001
wsdb.0.db2inst1.NODE0002.CATN0000.20020331234828.001
wsdb.0.db2inst1.NODE0003.CATN0000.20020331235235.001
```

먼저 카탈로그 파티션을 리스토어한 후 /dev3/backup 디렉토리에서 WSDB 데이터베이스의 다른 모든 데이터베이스 파티션을 리스토어하려면, 데이터베이스 파티션 중 하나에서 다음 명령을 발행하십시오.

```
db2_all '<<+0< db2 RESTORE DATABASE wsdb FROM /dev3/backup
TAKEN AT 20020331234149
INTO wsdb REPLACE EXISTING'
db2_all '<<+1< db2 RESTORE DATABASE wsdb FROM /dev3/backup
TAKEN AT 20020331234427
INTO wsdb REPLACE EXISTING'
db2_all '<<+2< db2 RESTORE DATABASE wsdb FROM /dev3/backup
TAKEN AT 20020331234828
INTO wsdb REPLACE EXISTING'
db2_all '<<+3< db2 RESTORE DATABASE wsdb FROM /dev3/backup
TAKEN AT 20020331235235
INTO wsdb REPLACE EXISTING'
```

db2\_all 유틸리티는 지정된 데이터베이스 파티션마다 리스토어 명령을 발행합니다. db2\_all을 사용하여 리스토어를 수행할 때, 항상 REPLACE EXISTING 및/또는 WITHOUT PROMPTING을 지정해야 합니다. 그렇지 않으면 프롬프트가 표시되는 경우에 작업이 정지된 것처럼 보입니다. 이는 db2\_all이 사용자 프롬프팅을 지원하지 않기 때문입니다.

2. 다음은 별명이 MYDB인 데이터베이스에 대한 일반적인 경로 재지정된 리스토어 시나리오입니다.

a. REDIRECT 옵션과 함께 RESTORE DATABASE 명령을 발행하십시오.

```
restore db mydb replace existing redirect
```

1단계가 성공적으로 완료되면, 3단계 완료 이전에 다음을 발행하여 리스토어 작업을 중단할 수 있습니다.

```
restore db mydb abort
```

b. 컨테이너를 재정의해야 하는 테이블 스페이스마다 SET TABLESPACE CONTAINERS 명령을 발행하십시오. 예를 들면, 다음과 같습니다.

```
set tablespace containers for 5 using
(file 'f:Wts3con1' 20000, file 'f:Wts3con2' 20000)
```

리스토어된 데이터베이스의 컨테이너가 이 단계에서 지정된 컨테이너인지 검증하려면, LIST TABLESPACE CONTAINERS 명령을 발행하십시오.

c. 1 및 2단계를 성공적으로 완료한 후 다음을 발행하십시오.

```
restore db mydb continue
```

이는 경로 재지정된 리스토어 작업의 최종 단계입니다.

- d. 3단계가 실패하거나 리스토어 조적이 중단된 경우, 경로 재지정된 리스토어를 1단계부터 재시작할 수 있습니다.
3. 다음은 복구 가능한 데이터베이스의 주별 증분 백업 전략 샘플입니다. 주별 전체 데이터베이스 백업 조작과 일별 비누적(델타) 백업 조작 및 주중 누적(증분) 백업 조적이 포함됩니다.

```
(Sun) backup db mydb use tsm
(Mon) backup db mydb online incremental delta use tsm
(Tue) backup db mydb online incremental delta use tsm
(Wed) backup db mydb online incremental use tsm
(Thu) backup db mydb online incremental delta use tsm
(Fri) backup db mydb online incremental delta use tsm
(Sat) backup db mydb online incremental use tsm
```

금요일 아침에 작성된 이미지의 자동 데이터베이스 리스토어의 경우 다음을 발행하십시오.

```
restore db mydb incremental automatic taken at (Fri)
```

금요일 아침에 작성된 이미지의 수동 데이터베이스 리스토어의 경우 다음을 발행하십시오.

```
restore db mydb incremental taken at (Fri)
restore db mydb incremental taken at (Sun)
restore db mydb incremental taken at (Wed)
restore db mydb incremental taken at (Thu)
restore db mydb incremental taken at (Fri)
```

4. 리모트 사이트로의 전송을 위해 로그를 포함하는 백업 이미지를 생성하려면 다음을 발행하십시오.

```
backup db sample online to /dev3/backup include logs
```

백업 이미지를 리스토어하려면 LOGTARGET 경로를 제공하고 ROLLFORWARD 중에 이 경로를 지정하십시오.

```
restore db sample from /dev3/backup logtarget /dev3/logs
rollforward db sample to end of logs and stop overflow log path /dev3/logs
```

5. 로그를 포함하는 백업 이미지에서 로그 파일만 검색하려면 다음을 발행하십시오.

```
restore db sample logs from /dev3/backup logtarget /dev3/logs
```

6. USE TSM OPTIONS 키워드를 사용하여 리스토어 조작에 사용할 TSM 정보를 지정할 수 있습니다. Windows 플랫폼에서는 -fromowner 옵션을 생략하십시오.

- 구분된 문자열 지정:

```
restore db sample use TSM options "-fromnode=bar -fromowner=dmcinnis"
```

- 완전한 파일 지정:

```
restore db sample use TSM options @/u/dmcinnis/myoptions.txt
```

myoptions.txt 파일에는 다음 정보가 들어 있습니다. -fromnode=bar -fromowner=dmcinnis

7. 다음은 새 스토리지 경로로 사용하는 다중 파티션 자동 스토리지 사용 가능 데이터베이스의 단순 리스토어입니다. 데이터베이스는 원래 하나의 스토리지 경로 /myPath0으로 작성되었습니다.

- 카탈로그 파티션에서 restore db mydb on /myPath1,/myPath2를 발행하십시오.
- 모든 비카탈로그 파티션에서 restore db mydb를 발행하십시오.

8. 자동이 아닌 스토리지 데이터베이스에 대한 다음 명령의 스크립트 출력은

```
restore db sample from /home/jseifert/backups taken at 20050301100417 redirect
generate script SAMPLE_NODE0000.clp
```

다음과 유사합니다.

```
-- *****
-- ** automatically created redirect restore script
-- *****
UPDATE COMMAND OPTIONS USING S ON Z ON SAMPLE_NODE0000.out V ON;
SET CLIENT ATTACH_DBPARTITIONNUM 0;
SET CLIENT CONNECT_DBPARTITIONNUM 0;
-- *****
-- ** initialize redirected restore
-- *****
RESTORE DATABASE SAMPLE
-- USER '<username>'
-- USING '<password>'
FROM '/home/jseifert/backups'
TAKEN AT 20050301100417
-- DBPATH ON '<target-directory>'
INTO SAMPLE
-- NEWLOGPATH '/home/jseifert/jseifert/NODE0000/SQL00001/SQLLOGDIR/'
-- WITH <num-buff> BUFFERS
-- BUFFER <buffer-size>
-- REPLACE HISTORY FILE
-- REPLACE EXISTING
REDIRECT-- PARALLELISM <n>
-- WITHOUT ROLLING FORWARD
-- WITHOUT PROMPTING
;
-- *****
-- ** tablespace definition
-- *****
-- *****
-- ** Tablespace name                = SYSCATSPACE
-- ** Tablespace ID                  = 0
-- ** Tablespace Type                 = System managed space
-- ** Tablespace Content Type        = Any data
-- ** Tablespace Page size (bytes)   = 4096
-- ** Tablespace Extent size (pages) = 32
-- ** Using automatic storage        = No
-- ** Total number of pages          = 5572
-- *****
SET TABLESPACE CONTAINERS FOR 0
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
USING (
    PATH 'SQLT0000.0'
);
-- *****
-- ** Tablespace name                = TEMPSPACE1
-- ** Tablespace ID                  = 1
-- ** Tablespace Type                 = System managed space
-- ** Tablespace Content Type        = System Temporary data
-- ** Tablespace Page size (bytes)   = 4096
```

```

-- ** Tablespace Extent size (pages)          = 32
-- ** Using automatic storage                 = No
-- ** Total number of pages                  = 0
-- *****
SET TABLESPACE CONTAINERS FOR 1
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
USING (
  PATH 'SQLT0001.0'
);
-- *****
-- ** Tablespace name                         = USERSPACE1
-- ** Tablespace ID                           = 2
-- ** Tablespace Type                         = System managed space
-- ** Tablespace Content Type                 = Any data
-- ** Tablespace Page size (bytes)           = 4096
-- ** Tablespace Extent size (pages)         = 32
-- ** Using automatic storage                 = No
-- ** Total number of pages                  = 1
-- *****
SET TABLESPACE CONTAINERS FOR 2
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
USING (
  PATH 'SQLT0002.0'
);
-- *****
-- ** Tablespace name                         = DMS
-- ** Tablespace ID                           = 3
-- ** Tablespace Type                         = Database managed space
-- ** Tablespace Content Type                 = Any data
-- ** Tablespace Page size (bytes)           = 4096
-- ** Tablespace Extent size (pages)         = 32
-- ** Using automatic storage                 = No
-- ** Auto-resize enabled                     = No
-- ** Total number of pages                  = 2000
-- ** Number of usable pages                 = 1960
-- ** High water mark (pages)                = 96
-- *****
SET TABLESPACE CONTAINERS FOR 3
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
USING (
  FILE /tmp/dms1                               1000
, FILE /tmp/dms2                               1000
);
-- *****
-- ** Tablespace name                         = RAW
-- ** Tablespace ID                           = 4
-- ** Tablespace Type                         = Database managed space
-- ** Tablespace Content Type                 = Any data
-- ** Tablespace Page size (bytes)           = 4096
-- ** Tablespace Extent size (pages)         = 32
-- ** Using automatic storage                 = No
-- ** Auto-resize enabled                     = No
-- ** Total number of pages                  = 2000
-- ** Number of usable pages                 = 1960
-- ** High water mark (pages)                = 96
-- *****
SET TABLESPACE CONTAINERS FOR 4
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
USING (
  DEVICE '/dev/hdb1'                           1000
, DEVICE '/dev/hdb2'                           1000
);
-- *****
-- ** start redirect restore
-- *****
RESTORE DATABASE SAMPLE CONTINUE;

```

```
-- *****
-- ** end of file
-- *****
```

## 9. 자동 스토리지 데이터베이스에 대한 다음 명령의 스크립트 출력은

```
restore db test from /home/jseifert/backups taken at 20050304090733 redirect
generate script TEST_NODE0000.clp
```

다음과 유사합니다.

```
-- *****
-- ** automatically created redirect restore script
-- *****
UPDATE COMMAND OPTIONS USING S ON Z ON TEST_NODE0000.out V ON;
SET CLIENT ATTACH_DBPARTITIONNUM 0;
SET CLIENT CONNECT_DBPARTITIONNUM 0;
-- *****
-- ** initialize redirected restore
-- *****
RESTORE DATABASE TEST
-- USER '<username>'
-- USING '<password>'
FROM '/home/jseifert/backups'
TAKEN AT 20050304090733
ON '/home/jseifert'
-- DBPATH ON <target-directory>
INTO TEST
-- NEWLOGPATH '/home/jseifert/jseifert/NODE0000/SQL00002/SQLGDIR/'
-- WITH <num-buff> BUFFERS
-- BUFFER <buffer-size>
-- REPLACE HISTORY FILE
-- REPLACE EXISTING
REDIRECT-- PARALLELISM <n>
-- WITHOUT ROLLING FORWARD
-- WITHOUT PROMPTING
;
-- *****
-- ** tablespace definition
-- *****
-- ** Tablespace name = SYSCATSPACE
-- ** Tablespace ID = 0
-- ** Tablespace Type = Database managed space
-- ** Tablespace Content Type = Any data
-- ** Tablespace Page size (bytes) = 4096
-- ** Tablespace Extent size (pages) = 4
-- ** Using automatic storage = Yes
-- ** Auto-resize enabled = Yes
-- ** Total number of pages = 6144
-- ** Number of usable pages = 6140
-- ** High water mark (pages) = 5968
-- *****
-- ** Tablespace name = TEMPSPACE1
-- ** Tablespace ID = 1
-- ** Tablespace Type = System managed space
-- ** Tablespace Content Type = System Temporary data
-- ** Tablespace Page size (bytes) = 4096
-- ** Tablespace Extent size (pages) = 32
-- ** Using automatic storage = Yes
-- ** Total number of pages = 0
-- *****
-- ** Tablespace name = USERSPACE1
-- ** Tablespace ID = 2
-- ** Tablespace Type = Database managed space
```

```

-- ** Tablespace Content Type           = Any data
-- ** Tablespace Page size (bytes)      = 4096
-- ** Tablespace Extent size (pages)    = 32
-- ** Using automatic storage           = Yes
-- ** Auto-resize enabled                = Yes
-- ** Total number of pages              = 256
-- ** Number of usable pages             = 224
-- ** High water mark (pages)           = 96
-- *****
-- *****
-- ** Tablespace name                    = DMS
-- ** Tablespace ID                       = 3
-- ** Tablespace Type                     = Database managed space
-- ** Tablespace Content Type             = Any data
-- ** Tablespace Page size (bytes)        = 4096
-- ** Tablespace Extent size (pages)      = 32
-- ** Using automatic storage             = No
-- ** Auto-resize enabled                  = No
-- ** Total number of pages               = 2000
-- ** Number of usable pages              = 1960
-- ** High water mark (pages)            = 96
-- *****
SET TABLESPACE CONTAINERS FOR 3
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
USING (
  FILE   '/tmp/dms1'                      1000
, FILE   '/tmp/dms2'                      1000
);
-- *****
-- ** Tablespace name                    = RAW
-- ** Tablespace ID                       = 4
-- ** Tablespace Type                     = Database managed space
-- ** Tablespace Content Type             = Any data
-- ** Tablespace Page size (bytes)        = 4096
-- ** Tablespace Extent size (pages)      = 32
-- ** Using automatic storage             = No
-- ** Auto-resize enabled                  = No
-- ** Total number of pages               = 2000
-- ** Number of usable pages              = 1960
-- ** High water mark (pages)            = 96
-- *****
SET TABLESPACE CONTAINERS FOR 4
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
USING (
  DEVICE '/dev/hdb1'                      1000
, DEVICE '/dev/hdb2'                      1000
);
-- *****
-- ** start redirect restore
-- *****
RESTORE DATABASE TEST CONTINUE;
-- *****
-- ** end of file
-- *****

```

10. 다음은 SNAPSHOT 옵션을 사용하는 RESTORE DB 명령의 예입니다.

스냅샷 이미지에서 로그 디렉토리 볼륨을 리스토어하고 프롬프트하지 않습니다.

```
db2 restore db sample use snapshot LOGTARGET INCLUDE without prompting
```

로그 디렉토리 볼륨을 리스토어하지 않고 프롬프트도 하지 않습니다.

```
db2 restore db sample use snapshot LOGTARGET EXCLUDE without prompting
```



로그 디렉토리 볼륨을 리스토어하지 않고 프롬프트도 하지 않습니다. LOGTARGET 을 지정하지 않은 경우, 디폴트는 LOGTARGET EXCLUDE입니다.

```
db2 restore db sample use snapshot without prompting
```

충돌하는 로그 디렉토리를 포함하는 스냅샷 이미지를 리스토어할 때 프롬프트 없이 현재 데이터베이스의 기존 로그 디렉토리 위에 겹쳐쓰고 이 디렉토리가 교체되도록 허용합니다.

```
db2 restore db sample use snapshot LOGTARGET EXCLUDE FORCE without prompting
```

충돌하는 로그 디렉토리를 포함하는 스냅샷 이미지를 리스토어할 때 프롬프트 없이 현재 데이터베이스의 기존 로그 디렉토리 위에 겹쳐쓰고 이 디렉토리가 교체되도록 허용합니다.

```
db2 restore db sample use snapshot LOGTARGET INCLUDE FORCE without prompting
```

## 사용 시 참고사항

- db2 restore db <name> 양식의 RESTORE DATABASE 명령은 데이터베이스 이미지를 사용하여 전체 데이터베이스 리스토어를 수행하고 테이블 스페이스 이미지에서 발견된 테이블 스페이스의 테이블 스페이스 리스토어 작업을 수행합니다. db2 restore db <name> tablespace 양식의 RESTORE DATABASE 명령은 이미지에서 발견된 테이블 스페이스의 테이블 스페이스 리스토어를 수행합니다. 또한 이와 같은 명령과 함께 테이블 스페이스 목록이 제공되면, 명시적으로 나열된 테이블 스페이스가 리스토어됩니다.
- 온라인 백업의 리스토어 조작 다음에 롤 포워드 복구를 수행해야 합니다.
- 백업 이미지가 압축된 경우 DB2 데이터베이스 시스템은 이 상태를 발견하고 리스토어하기 전에 데이터를 자동으로 압축 해제합니다. 라이브러리가 db2Restore API에 지정된 경우 데이터 압축 해체에 사용됩니다. 그렇지 않으면, 백업 이미지에 라이브러리가 저장되어 있는지, 그리고 이미지가 사용되는 라이브러리가 존재하는지 확인합니다. 마지막으로, 백업 이미지에 저장된 라이브러리가 없는 경우에는 데이터를 압축 해제할 수 없으며 리스토어 조작이 실패합니다.
- 백업 이미지에서 압축 라이브러리를 리스토어 중인 경우(COMPRESSION LIBRARY 옵션을 지정하여 명시적으로, 또는 압축된 백업의 일반 리스토어를 수행하여 내재적으로) 리스토어 조작은 백업이 수행된 동일한 플랫폼 및 운영 체제에서 수행해야 합니다. DB2에서 일반적으로 두 시스템을 수반하는 플랫폼 간 리스토어 작업을 지원하는 경우에도, 백업이 수행된 플랫폼이 리스토어가 수행되는 플랫폼과 다른 경우 리스토어 조작이 실패합니다.
- 백업된 SMS 테이블 스페이스는 SMS 테이블 스페이스로만 리스토어할 수 있습니다. DMS 테이블 스페이스로, 또는 그 반대로 리스토어할 수 없습니다.
- 로그 파일을 포함하는 백업 이미지에서 로그 파일을 리스토어하려면 DB2 서버에 있는 완전하고 유효한 경로를 제공하여 LOGTARGET 옵션을 지정해야 합니다. 해당 조건이 충족되면 리스토어 유틸리티는 이미지의 로그 파일을 목표 경로에 기록합니다.

다. 로그를 포함하지 않는 백업 이미지 리스토어 중에 LOGTARGET을 지정한 경우, 리스토어 조작용 테이블 스페이스 데이터 리스토어 시도 전에 오류를 리턴합니다. 유효하지 않거나 읽기 전용의 LOGTARGET 경로를 지정한 경우에도 리스토어 조작용이 오류와 함께 실패합니다.

- RESTORE DATABASE 명령이 발행될 때 LOGTARGET 경로에 로그 파일이 있으면 경고 프롬프트가 사용자에게 리턴됩니다. WITHOUT PROMPTING을 지정한 경우에는 이 경고가 리턴되지 않습니다.
- LOGTARGET이 지정된 리스토어 조작용 중에 로그 파일을 추출할 수 없으면 리스토어 조작용이 실패하고 오류가 리턴됩니다. 백업 이미지에서 추출 중인 로그 파일이 LOGTARGET 경로에 있는 기존 파일과 이름이 같은 경우 리스토어 조작용이 실패하고 오류가 리턴됩니다. 리스토어 데이터베이스 유틸리티는 LOGTARGET 디렉토리의 기존 로그 파일을 겹쳐쓰지 않습니다.
- 백업 이미지에서는 저장된 로그 세트만 리스토어할 수 있습니다. 로그 파일만 리스토어할 것을 표시하려면 LOGTARGET 경로와 함께 LOGS 옵션을 지정하십시오. LOGTARGET 경로 없이 LOGS 옵션만 지정하면 오류가 발생합니다. 이 작업 모드에서 로그 파일을 리스토어하는 중에 문제점이 발생하면 리스토어 조작용이 즉시 종료되고 오류가 리턴됩니다.
- 자동 증분 리스토어 조작용 중에는 리스토어 조작용의 목표 이미지에 포함된 로그 파일만 백업 이미지에서 검색됩니다. 증분 리스토어 프로세스 중에 참조되는 중간 이미지에 포함된 로그 파일은 중간 백업 이미지에서 추출되지 않습니다. 수동 증분 리스토어 조작용 중에 LOGTARGET 경로는 발행할 최종 리스토어 명령에만 지정해야 합니다.
- 오프라인 전체 데이터베이스 백업과 오프라인 증분 데이터베이스 백업은 나중 데이터베이스 버전으로 리스토어할 수 있지만, 온라인 백업은 리스토어할 수 없습니다. 다중 파티션 데이터베이스의 경우, 카탈로그 파티션을 개별적으로 먼저 리스토어하고, 그 다음으로 나머지 데이터베이스 파티션을 리스토어합니다(병렬로 또는 순차적으로). 그러나 리스토어 조작용에서 수행된 내재된 데이터베이스 업그레이드는 실패할 수 있습니다. 다중 파티션 데이터베이스에서는 하나 이상의 데이터베이스 파티션에서 실패할 수 있습니다. 이와 같은 경우, 데이터베이스를 성공적으로 업그레이드하기 위해 카탈로그 파티션에서 발행된 단일 UPGRADE DATABASE 명령을 RESTORE DATABASE 명령 다음에 발행할 수 있습니다.

### 스냅샷 리스토어

일반(비스냅샷) 리스토어처럼, 스냅샷 백업 이미지를 리스토어할 때 디폴트 동작은 로그 디렉토리를 리스토어하지 않는 것입니다(LOGTARGET EXCLUDE).

DB2 관리 프로그램이 리스토어하려는 기타 경로에서 로그 디렉토리의 그룹 ID가 공유되는 것을 발견하면 오류가 리턴됩니다. 이 경우 LOGTARGET INCLUDE 또는 LOGTARGET INCLUDE FORCE를 지정해야 합니다. 로그 디렉토리가 리스토어의 일부여야 하기 때문입니다.

DB2 관리 프로그램은 백업 이미지의 경로 리스토어가 수행되기 전에 기존 로그 디렉토리를 모두 저장하려고 합니다(기본, 미러 및 오버플로우).

로그 디렉토리를 리스토어하려고 하는데 DB2 관리 프로그램이 디스크의 기존 로그 디렉토리가 백업 이미지의 로그 디렉토리와 충돌하는 것을 발견하면 DB2 관리 프로그램이 오류를 보고합니다. 이와 같은 경우 LOGTARGET INCLUDE FORCE를 지정하면 이 오류가 표시되지 않고 이미지의 로그 디렉토리가 리스토어되어 기존의 모든 사항은 삭제됩니다.

LOGTARGET EXCLUDE 옵션이 지정되고 로그 디렉토리 경로가 데이터베이스 디렉토리(즉, /NODExxxx/SQLxxxxx/SQLLOGDIR/)에 있는 특수한 상황도 있습니다. 이 경우 리스토어는 로그 디렉토리를 데이터베이스 경로로 겹쳐쓰고 포함된 모든 콘텐츠가 리스토어됩니다. DB2 관리 프로그램이 이 시나리오를 발견하고 로그 파일이 이 로그 디렉토리에 있는 경우 오류가 보고됩니다. LOGTARGET EXCLUDE FORCE를 지정하면, 이 오류가 표시되지 않고 백업 이미지의 해당 로그 디렉토리는 디스크의 충돌하는 로그 디렉토리를 겹쳐씁니다.



---

## 제 112 장 REWIND TAPE

백업 및 리스토어 작업을 위한 테이프를 스트리밍 테이프 디바이스로 되감습니다. 이 명령은 Windows 운영 체제에서만 지원됩니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*

### 필수 연결

없음

### 명령 구문

```
▶▶—REWIND TAPE—┐  
                   └—ON— device—┘
```

### 명령 매개변수

**ON** *device*

유효한 테이프 디바이스 이름을 지정합니다. 디폴트값은 `WW.WTAPE0`입니다.



---

## 제 113 장 ROLLFORWARD DATABASE

데이터베이스 로그 파일에 기록된 트랜잭션을 적용하여 데이터베이스를 복구합니다. 데이터베이스 또는 테이블 스페이스 백업 이미지가 리스토어된 후 또는 미디어 오류로 인해 데이터베이스에 의해 테이블 스페이스가 오프라인이 된 경우 호출됩니다. 롤 포워드 복구로 데이터베이스를 복구할 수 있으려면 데이터베이스가 복구 가능해야 합니다(즉, **logarchmeth1** 또는 **logarchmeth2** 데이터베이스 구성 매개변수가 OFF 이외의 값으로 설정되어야 함).

### 범위

파티션된 데이터베이스 환경에서 이 명령은 카탈로그 파티션에서만 호출할 수 있습니다. 지정된 특정 시점까지의 데이터베이스 또는 테이블 스페이스 롤 포워드 조작은 `db2nodes.cfg` 파일에 나열되는 모든 데이터베이스 파티션에 영향을 줍니다. 로그 끝까지 데이터베이스 또는 테이블 스페이스 롤 포워드 조작은 지정된 데이터베이스 파티션에 영향을 줍니다. 데이터베이스 파티션을 지정하지 않는 경우 `db2nodes.cfg` 파일에 나열되는 모든 데이터베이스 파티션에 영향을 줍니다. 롤 포워드 복구가 특정 파티션에서 필요없는 경우 해당 파티션은 무시됩니다.

파티션된 테이블의 경우 관련 테이블 스페이스도 동일한 특정 시점으로 롤 포워드해야 합니다. 이것은 테이블의 데이터 파티션이 들어 있는 테이블 스페이스에 적용됩니다. 단일 테이블 스페이스에 파티션된 테이블의 분할 영역이 포함되는 경우에도 로그 끝까지 롤 포워드할 수 있습니다.

이전 DB2 릴리스 버전에서 작성된 로그 파일을 통해 롤 포워드하는 것은 불가능합니다. 이것은 새 DB2 릴리스 버전으로 업그레이드할 때 중요한 고려사항입니다.

### 권한 부여

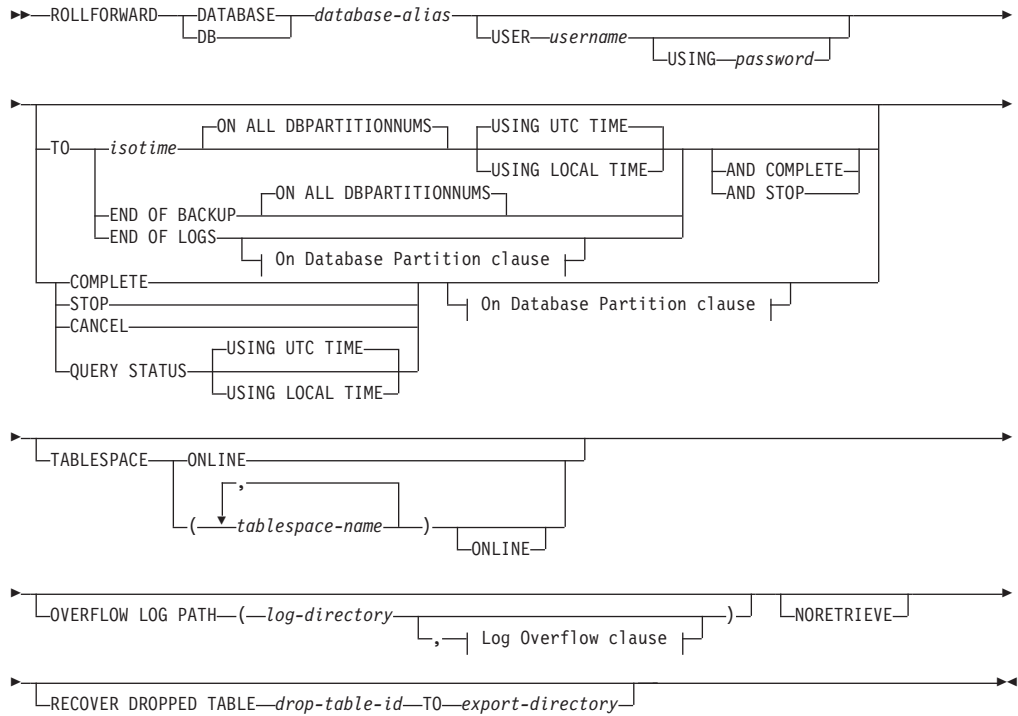
다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*

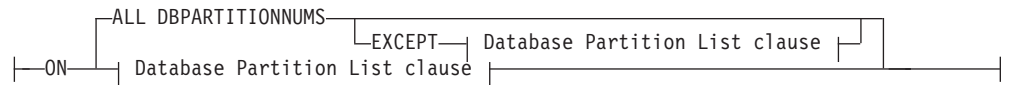
### 필수 연결

없음. 이 명령은 데이터베이스 연결을 설정합니다.

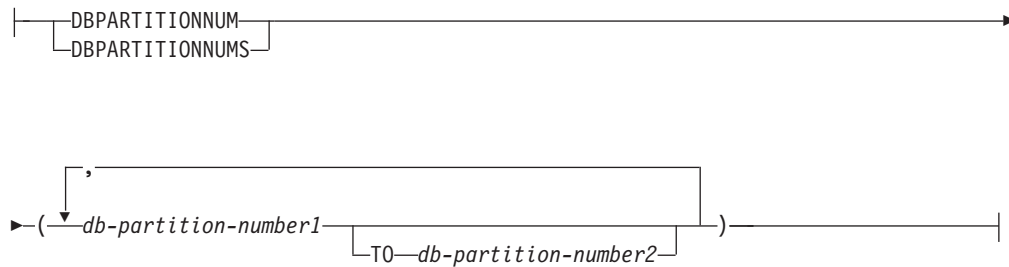
## 명령 구문



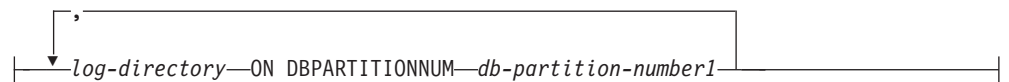
### On Database Partition clause:



### Database Partition List clause:



### Log Overflow clause:





## 명령 매개변수

### **DATABASE** *database-alias*

롤 포워드 복구할 데이터베이스의 별명.

### **USER** *username*

데이터베이스를 롤 포워드 복구할 사용자 이름

### **USING** *password*

사용자 이름을 인증하는 데 사용되는 암호. 암호를 생략하면 사용자가 이를 입력하도록 프롬프트가 표시됩니다.

### **TO**

#### *isotime*

커미트된 모든 트랜잭션이 롤 포워드될 특정 시점(해당 시간에 정확하게 커미트된 트랜잭션 및 이전에 커미트된 모든 트랜잭션 포함).

이 값은 조합된 날짜 및 시간을 식별하는 7자리 문자열인 시간소인으로 지정됩니다. 형식은 세계 표준시(UTC, 이전에는 GMT로 알려짐)로 표현되는 *yyyy-mm-dd-hh.mm.ss*(년, 월, 일, 시, 분, 초)입니다. UTC는 동일한 시간 소인이(예를 들어 일광 절약 시간과 연관된 시간 변경 때문에) 여러 로그와 연관되지 않도록 하는 데 도움이 됩니다. 백업 이미지의 시간소인은 백업 작업이 시작된 로컬 시간을 기초로 합니다. **CURRENT TIMEZONE** 특수 레지스터가 응용프로그램 서버(AS)에서 UTC와 로컬 시간 사이의 차이를 지정합니다. 차이는 시간 지속 기간(10진수로서, 처음 두 자릿수는 시간, 다음 두 자릿수는 분, 마지막 두 자릿수는 초임)으로 표현됩니다. 로컬 시간에서 **CURRENT TIMEZONE**을 빼면 로컬 시간이 UTC로 변환됩니다.

#### **USING UTC TIME**

UTC 시간으로 지정되는 특정 시점으로 롤 포워드할 수 있도록 허용합니다. 디폴트 옵션입니다.

#### **USING LOCAL TIME**

UTC 시간이 아니라 서버의 로컬 시간인 특정 시점으로 롤 포워드할 수 있도록 허용합니다.

#### 주:

1. 롤 포워드를 위한 로컬 시간을 지정하는 경우 사용자에게 리턴되는 모든 메시지도 로컬 시간입니다. 모든 시간은 카탈로그 데이터베이스 파티션의 서버와, 파티션된 데이터베이스 환경에서 변환됩니다.
2. 시간소인 문자열은 서버에서 UTC로 변환되므로, 시간은 클라이언트가 아니라 서버의 시간대에 로컬입니다. 클라이언트가 한 시간대

에 있고 서버가 다른 시간대에 있으면 서버의 로컬 시간을 사용해야 합니다. 이것은 클라이언트에 로컬인 제어 센터의 로컬 시간 옵션과 다릅니다.

3. 시간소인 문자열이 일광 절약 시간으로 인해 클럭의 시간 변경에 가까운 경우 중지 시간이 클럭 변경 이전 또는 이후인지 알고 올바르게 지정하는 것이 중요합니다.
4. USING LOCAL TIME 절을 지정할 수 없는 후속 ROLLFORWARD 명령은 이 옵션이 지정되는 경우 모든 메시지가 로컬 시간에 사용자에게 리턴되게 합니다.
5. USING LOCAL TIME 또는 USING UTC TIME(이전에는 GMT 시간으로 알려짐)을 올바르게 선택하는 것이 중요합니다. 지정되지 않는 경우 디폴트는 USING UTC TIME입니다. 선택할 때 실수하면 롤 포워드가 예상한 것과는 다른 특정 시점에 도달하고 해당 특정 시점 이후의 로그를 절단할 수 있습니다. 로컬 시간소인을 UTC 시간소인으로 잘못 취하면 필수 로그가 원하지 않게 전달될 수 있으며 잘못 취한 시간보다 늦은 시점에서의 추가 롤 포워드가 금지될 수 있습니다.

#### END OF LOGS

데이터베이스 구성 매개변수 **logpath**에 나열되는 모든 온라인 아카이브 로그 파일의 모든 커밋된 트랜잭션이 적용되도록 지정합니다.

#### END OF BACKUP

파티션된 데이터베이스의 모든 파티션이 최소 복구 시간으로 롤 포워드 되도록 지정합니다. 예에 대해서는 아래 예 섹션을 참조하십시오.

#### ALL DBPARTITIONNUMS | ON ALL DBPARTITIONNUMS

db2nodes.cfg 파일에서 지정되는 모든 데이터베이스 파티션에서 트랜잭션이 롤 포워드되도록 지정합니다. 이것은 데이터베이스 파티션 절이 지정되지 않는 경우 디폴트입니다.

#### EXCEPT

데이터베이스 파티션 목록에서 지정되는 것을 제외하고 db2nodes.cfg 파일에서 지정되는 모든 데이터베이스 파티션에서 트랜잭션이 롤 포워드되도록 지정합니다.

#### ON DBPARTITIONNUM | ON DBPARTITIONNUMS

데이터베이스 파티션 세트의 데이터베이스를 롤 포워드합니다.

*db-partition-number1*

데이터베이스 파티션 목록의 데이터베이스 파티션 번호를 지정합니다.

*db-partition-number2*

두 번째 데이터베이스 파티션 번호를 지정하여 *db-partition-number1*

에서 최고 *db-partition-number*2까지의 모든 데이터베이스 파티션이 데이터베이스 파티션 목록에 포함되도록 합니다.

### COMPLETE | STOP

로그 레코드의 롤 포워드를 중지하고, 불완전한 모든 트랜잭션을 롤백하고 데이터베이스의 롤 포워드 보류 상태를 꺼서 롤 포워드 복구 프로세스를 완료합니다. 이것은 롤 포워드될 데이터베이스 또는 테이블 스페이스에 대한 액세스를 허용합니다. 이들 키워드는 동등하므로, 둘 다 지정하지 말고 하나만 지정하십시오. 키워드 AND는 한 번에 다중 조작용의 지정을 허용합니다(예: `db2 rollforward db sample to end of logs and complete`). 테이블 스페이스를 특정 시점으로 롤 포워드할 때 테이블 스페이스는 백업 보류 상태에 들어 갑니다.

### CANCEL

롤 포워드 복구 조작용을 취소합니다. 그러면 포워드 복구가 시작된 모든 데이터베이스 파티션에 있는 하나 이상의 테이블 스페이스가 리스토어 보류 상태가 됩니다.

- 데이터베이스 롤 포워드 조작용이 진행 중이 아닌 경우(데이터베이스가 롤 포워드 보류 상태에 있음) 이 옵션을 사용하면 데이터베이스가 리스토어 보류 상태가 됩니다.
- 테이블 스페이스 롤 포워드 조작용이 진행 중이 아닌 경우(즉, 테이블 스페이스가 롤 포워드 보류 상태에 있음) 테이블 스페이스 목록을 지정해야 합니다. 목록의 모든 테이블 스페이스가 리스토어 보류 상태에 들어 갑니다.
- 테이블 스페이스 롤 포워드 조작용이 진행 중인 경우(즉, 적어도 하나의 테이블 스페이스가 롤 포워드 진행 상태에 있음), 롤 포워드 진행 상태에 있는 모든 테이블 스페이스가 리스토어 보류 상태가 됩니다. 테이블 스페이스 목록이 지정되는 경우 롤 포워드 진행 상태에 있는 모든 테이블 스페이스를 포함해야 합니다. 목록의 모든 테이블 스페이스가 리스토어 보류 상태에 들어 갑니다.
- 특정 시점으로 롤 포워드하는 경우, 전달되는 모든 테이블 스페이스 이름이 무시되고 롤 포워드 진행 상태에 있는 모든 테이블 스페이스는 리스토어 보류 상태가 됩니다.
- 테이블 스페이스 목록을 갖는 로그의 끝까지 롤 포워드하는 경우 나열된 테이블 스페이스만 리스토어 보류 상태가 됩니다.

이 옵션을 사용하여 이미 실행 중인 롤 포워드 조작용을 취소할 수 없습니다. 진행 중이지만 그때 실제로 실행 중이 아닌 롤 포워드 조작용을 취소하는 데만 사용할 수 있습니다. 다음 경우에 롤 포워드 조작용이 진행 중이지만 실행 중이 아닐 수 있습니다.

- 비정상적으로 종료했습니다.

- STOP 옵션을 지정하지 않았습니다.
- 오류 때문에 실패했습니다. 복구 불가능한 로드 조작을 통한 롤 포워드 같은 일부 오류는 테이블 스페이스를 리스토어 보류 상태로 만들 수 있습니다.

이 옵션은 주의해서, 일부 테이블 스페이스가 롤 포워드 보류 상태나 리스토어 보류 상태에 들어갔기 때문에 진행 중인 롤 포워드 조작을 완료할 수 없는 경우에만 사용하십시오. 의심이 가는 경우에는 LIST TABLESPACES 명령을 사용하여 롤 포워드 진행 상태 또는 롤 포워드 보류 상태에 있는 테이블 스페이스를 식별하십시오.

## QUERY STATUS

데이터베이스 관리 프로그램이 롤 포워드한 로그 파일, 필요한 다음 아카이브 파일 및 롤 포워드 처리가 시작한 이후 마지막 커밋된 트랜잭션의 시간소인(UTC 단위)을 나열합니다. 파티션된 데이터베이스 환경에서 이 상태 정보가 각 데이터베이스 파티션에 대해 리턴됩니다. 리턴된 정보에는 다음 필드가 들어 있습니다.

### 데이터베이스 파티션 번호

#### 롤 포워드 상태

상태는 데이터베이스 또는 테이블 스페이스 롤 포워드 보류, 데이터베이스 또는 테이블 스페이스 롤 포워드 진행, 데이터베이스 또는 테이블 스페이스 롤 포워드 중지 또는 보류 중이 아님일 수 있습니다.

#### 읽을 다음 로그 파일

다음 필수 로그 파일의 이름이 들어 있는 문자열입니다. 파티션된 데이터베이스 환경에서는 롤 포워드 유틸리티가 로그 파일이 누락되었거나 로그 정보 불일치가 발생했음을 표시하는 리턴 코드와 함께 실패하는 경우 이 정보를 사용하십시오.

#### 처리한 로그 파일

더 이상 복구에 필요하지 않으며 디렉토리에서 제거할 수 있는 처리된 로그 파일의 이름이 들어 있는 문자열입니다. 예를 들어 가장 오래된 언커밋 트랜잭션이 로그 파일  $x$ 에서 시작하는 경우, 사용하지 않는 로그 파일의 범위는  $x$ 를 포함하지 않으며 범위는  $x - 1$ 에서 끝납니다. 이 필드는 테이블 스페이스 롤 포워드 복구 조작의 경우 갱신되지 않습니다.

#### 최종 커밋 트랜잭션

“UTC” 또는 “Local”이 접미부로 붙는 ISO 형식(yyyy-mm-dd-hh.mm.ss)의 시간소인이 들어 있는 문자열입니다(USING LOCAL TIME 참조). 이 시간소인은 롤 포워드 복구 완료 후에 커밋된 마지

막 트랜잭션을 표시합니다. 시간소인은 데이터베이스에 적용됩니다. 테이블 스페이스 롤 포워드 복구의 경우 데이터베이스에 커밋된 마지막 트랜잭션의 시간소인입니다.

TO, STOP, COMPLETE 또는 CANCEL 절이 생략되는 경우 QUERY STATUS가 디폴트값입니다. TO, STOP 또는 COMPLETE가 지정된 경우, 명령이 성공적으로 완료되었으면 상태 정보가 표시됩니다. 개별적 테이블 스페이스가 지정되는 경우 무시됩니다. 상태 요청은 지정된 테이블 스페이스에만 적용되지 않습니다.

## TABLESPACE

이 키워드는 테이블 스페이스 레벨 롤 포워드 복구에 대해 지정됩니다.

### *tablespace-name*

특정 시점으로 테이블 스페이스 레벨 롤 포워드 복구의 경우에 필수입니다. 로그 끝까지 롤 포워드 복구에 대해 테이블 스페이스 서브세트를 지정할 수 있게 합니다. 파티션된 데이터베이스 환경에서 목록의 각 테이블 스페이스가 롤 포워드 중인 각 데이터베이스 파티션에 존재할 필요는 없습니다. 존재하는 경우 올바른 상태에 있어야 합니다.

파티션된 테이블의 경우 파티션된 테이블의 조각이 들어 있는 테이블 스페이스의 특정 시점 롤 포워드는 테이블이 있는 다른 모든 테이블 스페이스도 동일한 특정 시점으로 롤 포워드해야 합니다. 인덱스 파티션이 들어있는 테이블 스페이스가 파티션된 테이블의 조각 목록에 포함됩니다. 파티션된 테이블의 조각이 들어있는 단일 테이블 스페이스에 대한 로그 끝까지의 롤 포워드는 여전히 허용됩니다.

파티션된 테이블에 접속 또는 접속 해제된 데이터 파티션이 있는 경우 PIT 롤 포워드는 이들 데이터 파티션에 대한 모든 테이블 스페이스도 포함해야 합니다. 파티션된 테이블이 접속, 접속 해제 또는 삭제된 데이터 파티션을 갖는지 판별하려면 SYSDATAPARTITIONS 카탈로그 테이블의 상태 필드를 쿼리하십시오.

파티션된 테이블이 다중 테이블 스페이스에 상주할 수 있기 때문에 일반적으로 다중 테이블 스페이스를 롤 포워드해야 합니다. 삭제된 테이블 복구를 통해 복구되는 데이터는 ROLLFORWARD DATABASE 명령에 지정된 익스포트 디렉토리에 기록됩니다. 하나의 명령에서 모든 테이블 스페이스를 롤 포워드하거나 관련된 테이블 스페이스의 서브세트에 대해 반복된 롤 포워드 조작을 수행할 수 있습니다. ROLLFORWARD DATABASE 명령이 하나 또는 소수의 테이블 스페이스에 대해 완료되는 경우 해당 테이블 스페이스에 상주한 테이블의 모든 데이터가 복구됩니다. ROLLFORWARD DATABASE 명령이 테이블의 모든 데이터를 복구하기 위해 필요한 테이블 스페이스의 전체 세트를 지정하지 않은 경우 경고가 통지 로그에 기록됩니다. 테이블 스페이스의 서브세트 롤 포

워드를 허용하면 단일 익스포트 디렉토리에 적합할 수 있는 것보다 복구할 데이터가 더 많은 경우를 처리하기가 더 쉬워집니다.

### **ONLINE**

이 키워드는 테이블 스페이스 레벨 롤 포워드 복구를 온라인으로 수행할 수 있도록 지정합니다. 이는 롤 포워드 복구가 진행 중인 동안 다른 에이전트가 연결할 수 있음을 의미합니다.

### **OVERFLOW LOG PATH** *log-directory*

복구 중에 아카이브 로그를 검색할 대체 로그 경로를 지정합니다. 로그 파일이 **logpath** 데이터베이스 구성 매개변수에 의해 지정된 것과 다른 위치로 이동된 경우 이 매개변수를 사용하십시오. 파티션된 데이터베이스 환경에서 이것은 모든 데이터베이스 파티션에 대한 (완전한)디폴트 오버플로우 로그 경로입니다. 단일 파티션 데이터베이스의 경우 상대적 오버플로우 로그 경로를 지정할 수 있습니다. **OVERFLOW LOG PATH** 명령은 데이터베이스 구성 매개변수 **OVERFLOWLOGPATH**의 값(있는 경우)을 겹쳐씁니다.

### *log-directory* **ON DBPARTITIONNUM**

파티션된 데이터베이스 환경에서 다른 로그 경로가 특정 데이터베이스 파티션에 대한 디폴트 오버플로우 로그 경로를 겹쳐쓸 수 있게 합니다.

### **NORETRIEVE**

아카이브된 로그의 검색을 사용 안할 수 있게 하여 대기 머신에서 롤 포워드될 로그 파일을 제어할 수 있게 합니다. 이 옵션의 이점은 다음과 같습니다.

- 롤 포워드될 로그 파일을 제어하여 두 시스템 모두에 영향을 주지 않도록 대기 머신이 프로덕션 머신보다 X 시간 뒤가 되도록 할 수 있습니다.
- 대기 시스템이 아카이브에 액세스할 수 없는 경우(예를 들어, TSM이 아카이브인 경우 원래 머신만 파일을 검색할 수 있음)
- 프로덕션 시스템이 파일을 아카이브하는 동안 대기 시스템이 동일한 파일을 검색 중일 수 있으며, 불완전한 로그 파일을 가져올 수 있습니다. **Noretrieve**가 이 문제를 해결합니다.

### **RECOVER DROPPED TABLE** *drop-table-id*

롤 포워드 조작 동안 삭제된 테이블을 복구합니다. 출력 목록의 백업 ID 컬럼에서 **LIST HISTORY** 명령을 사용하여 테이블 ID를 알 수 있습니다. 파티션된 테이블의 경우 **drop-table-id**가 테이블을 전체적으로 식별하므로 해당 테이블의 모든 데이터 파티션이 하나의 롤 포워드 명령에서 복구될 수 있습니다.

### **TO** *export-directory*

테이블 데이터가 들어있는 파일이 기록될 디렉토리를 지정합니다. 디렉토리는 모든 데이터베이스 파티션에 액세스 가능해야 합니다.

예:

### 예 1

ROLLFORWARD DATABASE 명령은 각각 키워드 AND로 구분되는 다중 조작용 한 번에 지정할 수 있도록 합니다. 예를 들어 로그 끝까지 롤 포워드하고 완료하려면 개별 명령

```
db2 rollforward db sample to end of logs
db2 rollforward db sample complete
```

를 다음과 같이 조합할 수 있습니다.

```
db2 rollforward db sample to end of logs and complete
```

둘이 동등하지만 이런 조작용 2단계로 수행하는 것이 좋습니다. 조작용 중지하고 로그 롤 누락하기 전에 롤 포워드 조작용 예상대로 진행했는지 검증하는 것이 중요합니다. 특히 롤 포워드 복구 중에 잘못된 로그가 발견되고 잘못된 로그가 『로그 끝』을 의미하는 것으로 해석되는 경우에 중요합니다. 이 경우에는 해당 로그의 손상되지 않은 백업 사본을 사용하여 추가 로그를 통한 롤 포워드 조작용 계속할 수 있습니다. 그러나 롤 포워드 AND STOP 옵션이 사용되고 롤 포워드에서 오류가 발생하는 경우 오류가 사용자에게 리턴됩니다. 이 경우 강제로 롤 포워드가 중지하고 오류에도 불구하고 온라인이 되는(즉, 오류 전에 로그에 있는 해당 시점에 온라인이 되는) 유일한 방법은 ROLLFORWARD STOP 명령을 발행하는 것입니다.

### 예 2:

로그 끝까지 롤 포워드(두 테이블 스페이스가 리스토어되었음):

```
db2 rollforward db sample to end of logs
db2 rollforward db sample to end of logs and stop
```

이들 두 명령문은 동등합니다. 로그 끝까지 테이블 스페이스 롤 포워드 복구에는 AND STOP 또는 AND COMPLETE가 필요하지 않습니다. 테이블 스페이스 이름은 필수가 아닙니다. 지정되지 않는 경우 롤 포워드 복구가 필요한 모든 테이블 스페이스가 포함됩니다. 이들 테이블 스페이스의 서브세트만 롤 포워드될 경우 해당 이름을 지정해야 합니다.

### 예 3

3개의 테이블 스페이스가 리스토어된 후 하나를 로그 끝까지 롤 포워드하고 다른 둘은 특정 시점으로 롤 포워드하고 둘 다 온라인이 되게 하십시오.

```
db2 rollforward db sample to end of logs tablespace(TBS1) online

db2 rollforward db sample to 1998-04-03-14.21.56 and stop
tablespace(TBS2, TBS3) online
```

두 롤 포워드 조작을 동시에 실행할 수 없습니다. 두 번째 명령은 첫 번째 롤 포워드 조작이 완료된 후에만 호출할 수 있습니다.

#### 예 4

데이터베이스를 리스토어한 후 OVERFLOW LOG PATH를 사용하여 User Exit가 아카이브된 로그를 저장하는 디렉토리를 지정하여 특정 시점으로 롤 포워드하십시오.

```
db2 rollforward db sample to 1998-04-03-14.21.56 and stop
      overflow log path (/logs)
```

#### 예 5(파티션된 데이터베이스 환경)

0, 1 및 2의 세 데이터베이스 파티션이 있습니다. 테이블 스페이스 TBS1은 모든 데이터베이스 파티션에서 정의되고, 테이블 스페이스 TBS2는 데이터베이스 파티션 0과 2에서 정의됩니다. 데이터베이스 파티션 1의 데이터베이스를 리스토어하고 데이터베이스 파티션 0 및 2의 TBS1을 리스토어한 후, 데이터베이스 파티션 1에서 데이터베이스를 롤 포워드하십시오.

```
db2 rollforward db sample to end of logs and stop
```

이 명령은 경고 SQL1271(『데이터베이스가 복구되지만 하나 이상의 테이블 스페이스가 데이터베이스 파티션 0 및 2에서 오프라인입니다.』)을 리턴합니다.

```
db2 rollforward db sample to end of logs
```

이 명령은 데이터베이스 파티션 0과 2에서 TBS1을 롤 포워드합니다. 이 경우에 TABLESPACE(TBS1) 절은 선택적입니다.

#### 예 6(파티션된 데이터베이스 환경)

데이터베이스 파티션 0과 2만의 테이블 스페이스 TBS1을 리스토어한 후 데이터베이스 파티션 0과 2의 TBS1을 롤 포워드하십시오.

```
db2 rollforward db sample to end of logs
```

데이터베이스 파티션 1은 무시됩니다.

```
db2 rollforward db sample to end of logs tablespace(TBS1)
```

이 명령은 TBS1이 데이터베이스 파티션 1에서 롤 포워드 복구가 준비되지 않았기 때문에 실패합니다. SQL4906N을 보고합니다.

```
db2 rollforward db sample to end of logs on dbpartitionnums (0, 2)
      tablespace(TBS1)
```

이 명령은 성공적으로 완료됩니다.

```
db2 rollforward db sample to 1998-04-03-14.21.56 and stop
      tablespace(TBS1)
```



이 명령은 TBS1이 데이터베이스 파티션 1에서 롤 포워드 복구가 준비되지 않았기 때문에 실패합니다. 모든 조각이 함께 롤 포워드되어야 합니다. 테이블 스페이스가 특정 시점으로 롤 포워드될 때 데이터베이스 파티션 절은 승인되지 않습니다. 롤 포워드 조각은 테이블 스페이스가 있는 모든 데이터베이스 파티션에서 발생해야 합니다.

데이터베이스 파티션 1의 TBS1을 리스토어한 후,

```
db2 rollforward db sample to 1998-04-03-14.21.56 and stop
tablespace(TBS1)
```

이 명령은 성공적으로 완료됩니다.

#### 예 7(파티션된 데이터베이스 환경)

모든 데이터베이스 파티션의 테이블 스페이스를 리스토어한 후, 특정 시점 2로 롤 포워드하지만 AND STOP을 지정하지 마십시오. 롤 포워드 조치가 여전히 진행 중입니다. 취소하고 특정 시점 1로 롤 포워드하십시오.

```
db2 rollforward db sample to pit2 tablespace(TBS1)
db2 rollforward db sample cancel tablespace(TBS1)
```

**\*\* restore TBS1 on all database partitions \*\***

```
db2 rollforward db sample to pit1 tablespace(TBS1)
db2 rollforward db sample stop tablespace(TBS1)
```

#### 예 8(파티션된 데이터베이스 환경)

db2nodes.cfg 파일에 나열되는 8개 데이터베이스 파티션(3 - 10)에 있는 테이블 스페이스를 롤 포워드 복구하십시오.

```
db2 rollforward database dwtest to end of logs tablespace (tssprodt)
```

로그 끝(특정 시점이 아닌)까지의 이 조작은 성공적으로 완료됩니다. 테이블 스페이스가 있는 데이터베이스 파티션을 지정할 필요는 없습니다. 유틸리티는 db2nodes.cfg 파일이 디폴트입니다.

#### 예 9(파티션된 데이터베이스 환경)

단일 파티션 데이터베이스 파티션 그룹(데이터베이스 파티션 6의)에 있는 6개의 작은 테이블 스페이스를 롤 포워드 복구하십시오.

```
db2 rollforward database dwtest to end of logs on dbpartitionnum (6)
tablespace(tsstore, tssbuyer, tsstime, tsswhse, tsslscat, tssvendor)
```

로그 끝(특정 시점이 아닌)까지의 이 조작은 성공적으로 완료됩니다.

#### 예 10(파티션된 데이터베이스 환경)

TO END OF BACKUP 절을 ROLLFORWARD 명령과 함께 사용하여 파티션된 데이터베이스의 모든 파티션을 최소 복구 시간까지 롤 포워드할 수 있습니다. 최소 복구

시간은 데이터베이스가 일관성이 있을 때(데이터베이스 카탈로그에 나열된 오브젝트가 실제로 디스크에 존재하는 오브젝트와 일치할 때) 롤 포워드 중의 가장 빠른 특정 시점입니다. 데이터베이스를 롤 포워드할 올바른 특정 시점을 수동으로 판별하는 것은 특히 파티션된 데이터베이스의 경우에 어렵습니다. END OF BACKUP 옵션이 쉽게 만듭니다.

```
db2 rollforward db sample to end of backup and complete
```

## 사용 시 참고사항

온라인 백업 조작 중에 작성된 이미지에서 리스토어하는 경우, 롤 포워드 조작에 대해 지정된 특정 시점이 온라인 백업 조작이 완료한 시간보다 늦어야 합니다. 롤 포워드 조작이 이 시점을 통과하기 전에 중지되는 경우 데이터베이스는 롤 포워드 보류 상태에 남아 있습니다. 테이블 스페이스가 롤 포워드되는 프로세스에 있는 경우 롤 포워드 진행 상태에 남아 있습니다.

하나 이상의 테이블 스페이스가 특정 시점으로 롤 포워드되고 있는 경우 롤 포워드 조작은 이 테이블 스페이스 또는 해당 테이블에 대한 시스템 카탈로그에 대한 최종 갱신인 최소 복구 시간까지는 계속되어야 합니다. 테이블 스페이스에 대한 최소 복구 시간(세계 표준시 또는 UTC 단위)은 LIST TABLESPACES SHOW DETAIL 명령을 사용하여 검색할 수 있습니다.

데이터베이스 롤 포워드는 테이프 디바이스를 사용한 로드 복구가 필요할 수 있습니다. 다른 테이프를 프롬프트하는 경우 다음 중 하나로 응답할 수 있습니다.

- c** 계속. 경고 메시지를 생성한 디바이스를 계속 사용하십시오(예를 들어, 새 테이프가 마운트되었을 때).
- d** 디바이스 종료. 경고 메시지를 생성한 디바이스를 사용하여 중지합니다(예를 들어, 더 이상의 테이프가 없는 경우).
- t** 종료. 영향을 받은 모든 테이블 스페이스를 오프라인으로 만들지만 롤 포워드 처리를 계속합니다.

롤 포워드 유틸리티가 필요한 다음 로그를 찾을 수 없는 경우 로그 이름이 SQLCA에서 리턴되며 롤 포워드 복구가 중지됩니다. 추가 로그를 사용할 수 없는 경우 STOP 옵션을 사용하여 롤 포워드 복구를 종료하십시오. 불완전한 트랜잭션은 데이터베이스 또는 테이블 스페이스가 일관성 있는 상태에 남아 있음을 보장하기 위해 롤백됩니다.

**주:** 재분배 조작을 통한 롤 포워드는 로그 레코드가 데이터 재분배에 대해 기록되지 않으므로 데이터베이스 콘텐츠를 리스토어할 수 없습니다. 『REDISTRIBUTE DATABASE PARTITION GROUP 명령』을 참조하십시오.

## 호환성

버전 8 이전 버전과의 호환성:

- DBPARTITIONNUM에 키워드 NODE를 대체할 수 있습니다.
- DBPARTITIONNUMS에 키워드 NODES를 대체할 수 있습니다.
- 특정 시점 롤 포워드는 파티션된 테이블에 대한 V9.1 지원으로 인해 V9.1 이전 클라이언트에서는 지원되지 않습니다.



---

## 제 114 장 RUNCMD

CLP 대화식 모드 명령 실행기록에서 지정된 명령을 실행합니다.

### 범위

이 명령은 CLP 대화식 모드에서만 실행될 수 있습니다. 특히 CLP 명령 모드 또는 CLP 일괄처리 모드에서 실행할 수 없습니다.

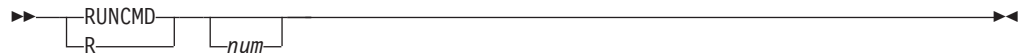
### 권한 부여

없음

### 필수 연결

필수 연결은 실행될 명령에 따라 다릅니다.

### 명령 구문



### 명령 매개변수

*num* *num*이 양수인 경우 명령 실행기록의 *num*에 해당하는 명령을 실행합니다. *num*이 음수인 경우 명령 실행기록의 최신 명령부터 역방향으로 카운팅하여 *num*에 해당하는 명령을 실행합니다. 0은 *num*에 유효한 값이 아닙니다. 이 매개변수가 지정되지 않으면 가장 최근에 실행된 명령을 실행합니다. 이는 *num*에 값 -1을 지정하는 것과 같습니다.

### 사용 시 참고사항

1. 일반적으로 HISTORY 명령을 실행하여 최근에 실행된 명령의 목록을 본 다음 RUNCMD를 실행하여 이 목록에서 명령을 실행합니다.
2. RUNCMD 명령은 명령 실행기록에 기록되지 않지만 RUNCMD 명령으로 실행된 명령은 명령 실행기록에 기록됩니다.



---

## 제 115 장 RUNSTATS

테이블 및/또는 연관된 인덱스 또는 통계적 뷰의 등록 정보에 대한 통계를 갱신하십시오. 이러한 등록 정보에는 레코드 수, 페이지 수 및 평균 레코드 길이가 포함됩니다. 옵티마이저는 데이터에 대한 액세스 경로를 판별할 때 이들 통계를 사용합니다.

테이블의 경우, 테이블을 재구성한 후 또는 테이블에 갱신사항이 많이 있을 때 이 유틸리티를 호출해야 합니다. 통계적 뷰의 경우, 기본이 되는 테이블에 대한 변경사항이 뷰에서 리턴된 행에 실질적으로 영향을 줄 때 이 유틸리티를 호출해야 합니다. 이 뷰는 ALTER VIEW 명령을 사용하여 쿼리 최적화에 사용될 수 있도록 미리 설정되어 되어 있어야 합니다.

### 범위

이 명령은 db2nodes.cfg 파일에서 모든 데이터베이스 파티션으로부터 발행될 수 있습니다. 카탈로그 데이터베이스 파티션에 있는 카탈로그를 갱신하는 데 사용될 수 있습니다.

테이블의 경우, 이 명령은 명령을 호출한 데이터베이스 파티션의 테이블에 대한 통계를 수집합니다. 해당 테이블이 이 데이터베이스 파티션에 존재하지 않으면 데이터베이스 파티션 그룹의 첫 번째 데이터베이스 파티션이 선택됩니다.

뷰의 경우, 이 명령은 모든 참여 중인 데이터베이스 파티션에 있는 테이블의 데이터를 사용하여 통계를 수집합니다.

### 권한 부여

테이블의 경우, 다음 중 하나이어야 합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*
- *sqladm*
- 테이블에 대한 CONTROL 특권
- LOAD 권한

해당 연결에 존재하는 선언된 임시 테이블에서 이 명령을 사용하는 데 명시적 특권이 필요하지는 않습니다.

통계 뷰의 경우, 다음 중 하나이어야 합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*
- *sqladm*
- 통계적 뷰에 대한 CONTROL 특권

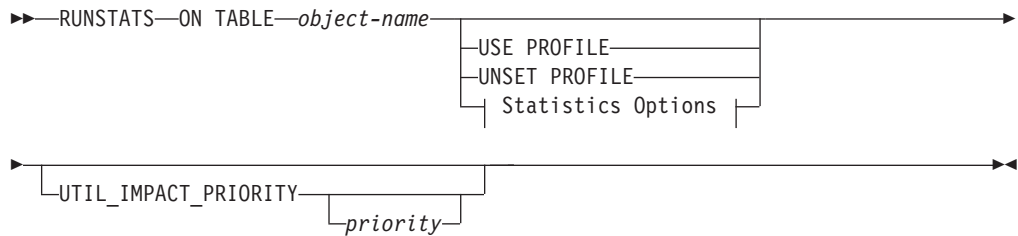
그 밖에, 통계적 뷰에서 행에 액세스하려면 적절한 특권이 있어야 합니다. 특히, 통계적 뷰 정의에 참조된 각 테이블, 통계적 뷰 또는 별명에 대해 사용자는 다음 특권 중 하나가 있어야 합니다.

- *dataaccess*
- CONTROL
- SELECT

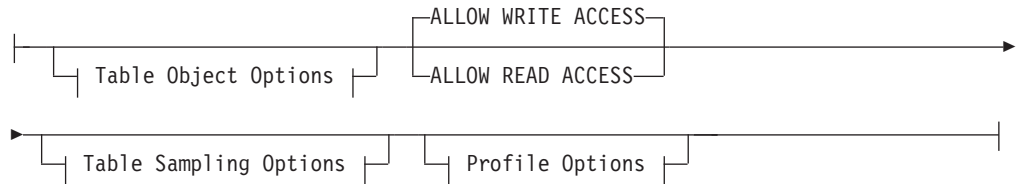
## 필수 연결

데이터베이스

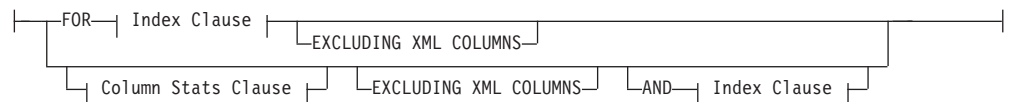
## 명령 구문



### Statistics Options:

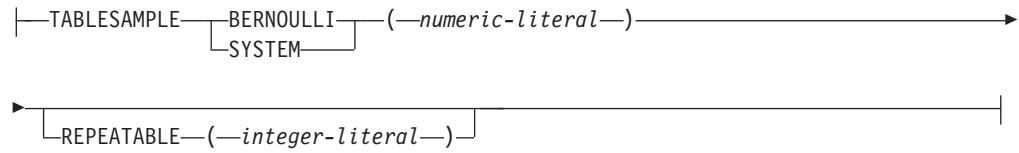


### Table Object Options:





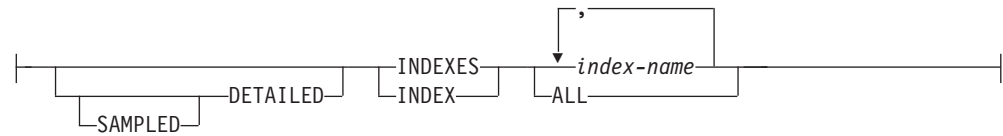
**Table Sampling Options:**



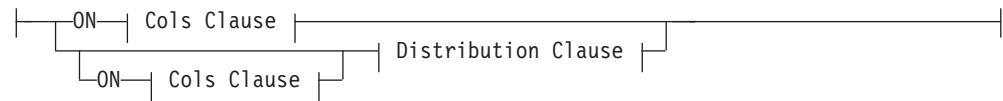
**Profile Options:**



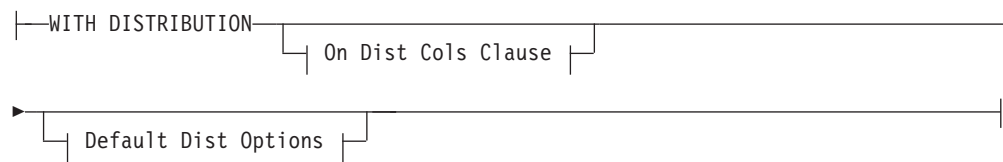
**Index Clause:**



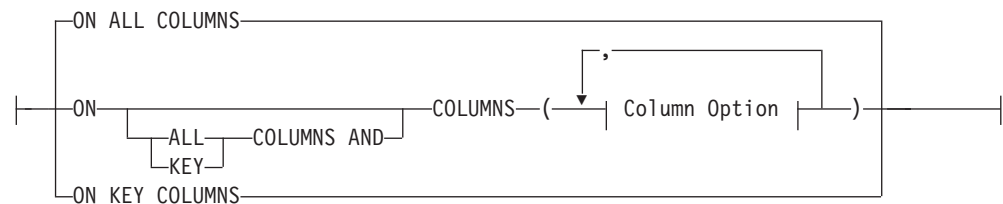
**Column Stats Clause:**



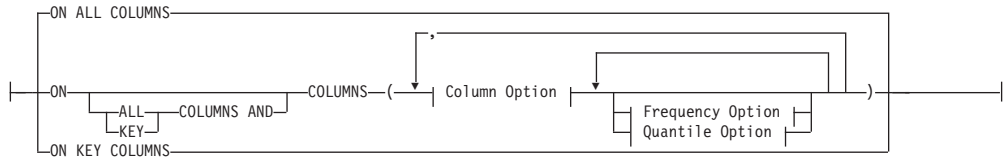
**Distribution Clause:**



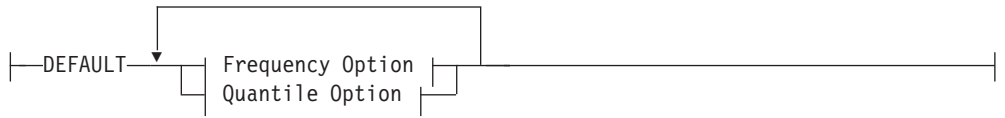
**On Col's Clause:**



**On Dist Col's Clause:**



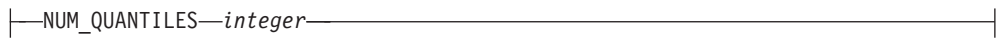
**Default Dist Option:**



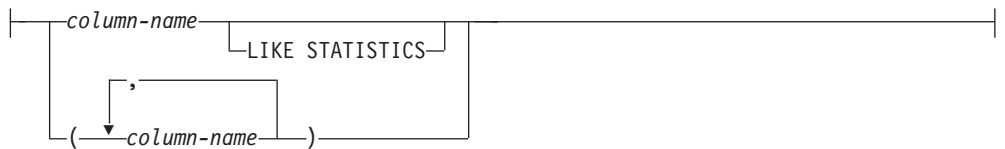
**Frequency Option:**



**Quantile Option:**



**Column Option:**



**명령 매개변수**

*object-name*

통계가 수집될 테이블 또는 통계 뷰를 식별합니다. 이는 계층 구조 테이블에서는 안 됩니다. 유형이 지정된 테이블의 경우, *object-name*는 테이블 계층 구조의 루트 테이블의 이름이어야 합니다. *schema.object-name* 양식의 완전한 이름 또는 별명을 사용해야 합니다. 스키마는 테이블이 작성된 사용자 이름입니다.

*index-name*

테이블에 정의된 기존 인덱스를 식별합니다. *schema.index-name* 양식의 완전한 이름을 사용해야 합니다. 뷰에는 이 옵션을 사용할 수 없습니다.

**USE PROFILE**

이 옵션은 RUNSTATS가 이전에 저장된 통계 프로파일을 사용하여 테이블 또는 통계 뷰에 대한 통계를 수집할 수 있도록 합니다. 통계 파일은 SET PROFILE 옵션을 사용하여 작성되고 UPDATE PROFILE 옵션을 사용하여 갱신됩니다.

## UNSET PROFILE

기존 통계 프로파일을 제거하려면 이 옵션을 지정하십시오. 예를 들면, 다음과 같습니다.

```
runstats on tablemyschema.mytable unset profile
```

## FOR INDEXES

인덱스 전용 통계를 수집하고 갱신합니다. 테이블에서 이전에 수집된 테이블 통계가 없는 경우 기본 테이블 통계가 수집되기도 합니다. 이들 기본 통계에는 분산 통계가 포함되지 않습니다. 뷰에는 이 옵션을 사용할 수 없습니다.

## AND INDEXES

테이블 및 인덱스 둘 다에 대한 통계를 수집하고 갱신합니다. 뷰에는 이 옵션을 사용할 수 없습니다.

## DETAILED

확장된 인덱스 통계를 계산합니다. 이는 비교적 대형 인덱스에 대해 수집되는 CLUSTERFACTOR 및 PAGE\_FETCH\_PAIRS 통계입니다. 뷰에는 이 옵션을 사용할 수 없습니다.

## SAMPLED

이 옵션은 DETAILED 옵션과 함께 사용될 때 RUNSTATS가 확장된 인덱스 통계를 컴파일할 때 CPU 샘플링 기술을 사용할 수 있도록 합니다. 이 옵션을 지정하지 않으면 인덱스의 모든 항목을 조사하여 확장된 인덱스 통계를 계산합니다. 뷰에는 이 옵션을 사용할 수 없습니다.

## ON ALL COLUMNS

적당한 모든 컬럼에서 통계를 수집하려면, ON ALL COLUMNS 절을 사용하십시오. 컬럼은 기본 통계 컬렉션(Cols 절에서)에 대해 또는 WITH DISTRIBUTION 절(Dist Cols 절에서)과 함께 지정될 수 있습니다. 해당 컬럼에만 해당되는 절이 지정되지 않은 경우 디폴트 옵션은 ON ALL COLUMNS 스펙입니다.

On Cols 절에 지정된 경우 WITH DISTRIBUTION 절에서 특정 컬럼을 선택한 경우를 제외하고는 모든 컬럼에 대해 기본 컬럼 통계만 수집됩니다. WITH DISTRIBUTION 절에 지정된 컬럼에 대해서는 또한 기본 및 분산 통계가 수집됩니다.

WITH DISTRIBUTION ON ALL COLUMNS을 지정하면 모든 적당한 컬럼에 대해 기본 통계 및 분산 통계가 수집됩니다. On Cols 절에 지정된 내용은 중복되므로 필요하지 않습니다.

## ON COLUMNS

이 절을 통해 사용자는 통계를 수집할 컬럼 목록을 지정할 수 있습니다. 컬럼

그룹을 지정하면 그룹에 대한 구별 값 수가 수집됩니다. 인덱스 통계를 수집하지 않고 테이블에서 RUNSTATS를 수행하고 통계가 수집될 컬럼의 서브세트를 지정하면 다음이 수행됩니다.

1. RUNSTATS 명령에 지정되지 않았지만 인덱스에서 첫 번째 컬럼인 컬럼에 대한 통계는 재설정되지 않습니다.
2. RUNSTATS 명령에 지정되지 않은 모든 다른 컬럼에 대한 통계는 재설정됩니다.

이 절은 On Cols 절 및 On Dist Cols 절에서 사용될 수 있습니다. 컬럼 그룹에 대한 분산 통계 수집은 현재 지원되지 않습니다.

컬럼 그룹에 XML 유형 컬럼이 지정된 경우, 그룹에 대한 구별 값을 수집하기 위해 XML 유형 컬럼을 무시합니다. 그러나 컬럼 그룹에 있는 XML 유형 컬럼을 위해 기본 XML 컬럼 통계는 수집됩니다.

### EXCLUDING XML COLUMNS

이 절은 통계 컬렉션에서 모든 XML 유형 컬럼을 생략할 수 있도록 합니다. 이 절은 XML 데이터를 포함하려면 더 많은 시스템 자원이 필요할 수 있으므로 비XML 컬럼에 대한 통계 컬렉션을 사용하도록 조정합니다. EXCLUDING XML COLUMNS절은 통계 컬렉션에 XML 컬럼을 지정하는 다른 절보다 우선순위를 갖습니다. 예를 들어, EXCLUDING XML COLUMNS절을 사용하며 ON COLUMNS절을 사용하여 XML 유형 컬럼을 지정하거나 ON ALL COLUMNS 절을 사용할 경우, 통계 컬렉션 중 모든 XML 유형 컬럼이 무시됩니다.

### ON KEY COLUMNS

특정 컬럼을 나열하는 대신 테이블에 정의된 모든 인덱스를 나열하는 컬럼에서 통계를 수집하도록 선택할 수 있습니다. 여기에서는 쿼리에서 중요한 컬럼이 테이블에서 인덱스를 작성하는 데 사용되는 컬럼이기도 하다고 간주됩니다. 테이블에 인덱스가 없는 경우 이는 공백 목록과 같으며 컬럼 통계는 수집되지 않습니다. 이 절은 on-cols-clause 또는 on-dist-cols-clause에서 사용할 수 있습니다. 두 절에 모두 지정될 경우 WITH DISTRIBUTION절이 기본 및 분산 통계 컬렉션을 지정하는 데 사용되므로 on-cols-clause에서 중복됩니다. XML 유형 컬럼은 정의이며 키 컬럼이 아니므로 ON KEY COLUMNS절에 의한 통계 컬렉션에 포함되지 않습니다. 뷰에는 이 옵션을 사용할 수 없습니다.

#### *column-name*

테이블 또는 통계 뷰에 있는 컬럼의 이름. 존재하지 않는 컬럼 또는 잘못된 입력된 컬럼 이름과 같이 부적당한 컬럼 이름을 통계 컬렉션에 지정하면 오류(-205)가 리턴됩니다. 두 개의 컬럼 목록을 하나는 분산을 포함하고 다른 하나는 분산을 포함하지 않고 지정할 수 있습니다. 컬럼이 WITH DISTRIBUTION절과 연관되지 않은 목록에 지정된 경우 기본 컬럼 통계만 수집됩니다. 컬럼이 두 목

록에 모두 나타날 경우 분산 통계가 수집됩니다(NUM\_FREQVALUES 및 NUM\_QUANTILES가 0으로 설정되지 않은 경우).

### **NUM\_FREQVALUES**

수집할 최대 빈도 수 값을 정의합니다. ON COLUMNS절에서 개별 컬럼에 대해 지정될 수 있습니다. 개별 컬럼에 대한 값이 지정되어 있지 않으면 DEFAULT절에 지정되어 있는 값에서 빈도 한계 값을 선택합니다. 여기에도 지정되어 있지 않으면 NUM\_FREQVALUES 데이터베이스 구성 매개변수에 설정된 값이 최대 빈도 수 값으로 수집됩니다.

### **NUM\_QUANTILES**

수집할 최대 분산 Quantile 수 값을 정의합니다. ON COLUMNS절에서 개별 컬럼에 대해 지정될 수 있습니다. 개별 컬럼에 대한 값이 지정되어 있지 않으면 DEFAULT절에 지정되어 있는 값에서 Quantile 한계 값을 선택합니다. 여기에도 지정되어 있지 않으면 NUM\_QUANTILES 데이터베이스 구성 매개변수에 설정된 값이 최대 Quantile 수 값으로 수집됩니다.

### **WITH DISTRIBUTION**

이 절은 컬럼에서 기본 통계와 분산 통계가 모두 수집되도록 지정합니다. ON COLUMNS 절을 지정하지 않으면 테이블 또는 통계 뷰의 모든 컬럼(CLOB 및 LONG VARCHAR과 같이 부적당한 컬럼은 제외)에서 분산 통계가 수집됩니다. ON COLUMNS절을 지정하면 제공된 컬럼 목록(통계 콜렉션에 부적당한 컬럼은 제외)에서만 분산 통계가 수집됩니다. 이 절을 지정하지 않으면 기본 통계만 수집됩니다.

컬럼 그룹에서 분산 통계 콜렉션은 현재 지원되지 않습니다. WITH DISTRIBUTION ON COLUMNS절에 컬럼 그룹이 지정되어 있을 때 분산 통계는 수집되지 않습니다.

### **DEFAULT**

NUM\_FREQVALUES 또는 NUM\_QUANTILES가 지정된 경우 이들 값은 해당 컬럼에 대해 수집될 최대 빈도 및 Quantile 수 통계를 판별하는 데 사용됩니다(ON COLUMNS절에서 개별 컬럼에 대해 지정되지 않은 경우). DEFAULT절이 지정되어 있지 않으면 해당 데이터베이스 구성 매개변수에 있는 값을 사용합니다.

### **LIKE STATISTICS**

이 옵션을 지정하면 추가 컬럼 통계가 수집됩니다. 이들 통계는 SYSSTAT.COLUMNS의 SUB\_COUNT 및 SUB\_DELIM\_LENGTH 통계입니다. 1바이트 문자 세트(SBCS), FOR BIT DATA 또는 UTF-8의 코드 페이지 속성이 있는 유형 CHAR 및 VARCHAR의 컬럼에 대한 통계가 수집됩니다. 쿼리 옵티마이저는 이들 통계를 사용하여 "column LIKE '%xyz'" 및 "column LIKE '%xyz%'" 유형의 술어에 대한 선택 빈도 추정을 향상시킵니다.

## ALLOW WRITE ACCESS

통계가 계산되는 동안 다른 사용자가 테이블에서 읽고 쓸 수 있도록 지정합니다. 통계 뷰의 경우 이는 뷰 정의에 참조된 기본 테이블입니다.

삽입, 갱신 또는 삭제가 동시에 대량으로 발생하는 테이블에는 ALLOW WRITE ACCESS 옵션을 사용하지 마십시오. RUNSTATS 명령은 먼저 테이블 통계를 수행한 다음 인덱스 통계를 수행합니다. 테이블 통계와 인덱스 통계를 수집하는 사이에 상태가 변경되면 불일치가 발생할 수 있습니다. 쿼리를 최적화하기 위해서는 최신 통계가 있어야 하지만 통계의 일치도 중요합니다. 따라서 삽입, 갱신 또는 삭제량이 최소일 때 통계를 수집해야 합니다.

## ALLOW READ ACCESS

통계가 계산되는 동안 다른 사용자가 테이블에 읽기 전용 액세스를 할 수 있도록 지정합니다. 통계 뷰의 경우 이는 뷰 정의에 참조된 기본 테이블입니다.

## TABLESAMPLE BERNOULLI

이 옵션은 RUNSTATS가 테이블 또는 통계 뷰에서 샘플 행에 대한 통계를 수집할 수 있도록 합니다. BERNOULLI 샘플링은 각 행을 개별적으로 검토하여 확률이 P/100인 행은 포함시키고 확률이 1-P/100인 행은 제외시킵니다(여기서 P는 numeric-literal의 값임). 따라서 numeric-literal의 값이 10(10퍼센트 샘플을 나타냄)으로 평가된 경우 확률이 0.1인 행은 포함되고 확률이 0.9인 행은 제외됩니다. 선택적 REPEATABLE절이 지정된 경우를 제외하고는 RUNSTATS를 실행할 때마다 보통 다른 샘플의 테이블이 생성됩니다. 테이블 스캔을 통해 모든 데이터 페이지가 검색되지만 통계 컬렉션에는 numeric-literal 매개변수를 통해 지정된 퍼센트의 행만 사용됩니다.

## TABLESAMPLE SYSTEM

이 옵션은 RUNSTATS가 테이블에서 샘플 데이터 페이지에 대한 통계를 수집할 수 있도록 합니다. SYSTEM 샘플링은 각 페이지를 개별적으로 검토하여 확률이 P/100인 페이지는 포함시키고 확률이 1-P/100인 페이지는 제외시킵니다(여기서 P는 numeric-literal의 값임). 선택적 REPEATABLE절이 지정된 경우를 제외하고는 RUNSTATS를 실행할 때마다 보통 다른 샘플의 테이블이 생성됩니다. 샘플 크기는 괄호 안의 numeric-literal 매개변수에 의해 제어되며 이는 테이블의 대략적인 백분을 P를 나타냅니다. numeric-literal 매개변수를 통해 지정된 퍼센트의 데이터 페이지만 검색되며 통계 컬렉션에 사용됩니다.

통계 뷰에서 SYSTEM 샘플링은 단일 기본 테이블에서 정의를 선택하는 뷰로 제한됩니다. 뷰가 다중 테이블을 포함하면, 다음의 경우 SYSTEM 샘플링도 가능합니다.

- 테이블 간에 정의된 참조 무결성 제한조건에 포함된 모든 기본 키와 외부 키에 대해 등호 술어를 사용하여 테이블이 조인됩니다.
- 어떤 검색 조건도 관계의 모든 상위 테이블 행을 필터하지 않습니다.

- 상위 테이블이 아닌 단일 하위 테이블은 모든 테이블 간에 식별될 수 있습니다.

통계 뷰가 이들 조건을 충족시키지 않는 경우, BERNOULLI 샘플링이 대신 사용되며 경고가 리턴됩니다(SQL2317W).

### **REPEATABLE** (*integer-literal*)

TABLESAMPLE절에 REPEATABLE절을 추가하면 RUNSTATS가 반복 실행할 때 동일한 샘플이 리턴됩니다. *integer-literal* 매개변수는 음수가 아닌 정수로서 샘플링에 사용되는 시드를 나타냅니다. 음수 시드를 패스하면 오류가 발생합니다(SQL1197N). 마지막 TABLESAMPLE REPEATABLE이 실행된 후 테이블 또는 통계 뷰에 대한 활동으로 테이블 또는 통계 뷰 데이터가 변경된 경우 샘플 세트는 반복 가능한 RUNSTATS 호출 사이에서 계속 다를 수 있습니다. 또한 일치된 결과를 얻기 위해서는 BERNOULLI 또는 SYSTEM 키워드가 지정한 대로 샘플을 확보하는 방법도 동일해야 합니다.

### *numeric-literal*

*numeric-literal* 매개변수는 가져올 샘플의 크기를 백분율 P로 지정합니다. 이 값은 100 이하의 양수이어야 하며 1과 0 사이의 숫자가 될 수 있습니다. 예를 들어, 값 0.01은 1/100퍼센트를 나타내고 평균적으로 10,000개의 행 중 하나의 행이 샘플링되는 것을 의미합니다. 값 0 또는 100은 TABLESAMPLE BERNOULLI 또는 TABLESAMPLE SYSTEM이 지정되었는지 여부에 관계 없이 DB2 데이터베이스 시스템에 의해 샘플링이 지정되지 않은 것처럼 처리됩니다. 100보다 크거나 0보다 작은 값은 DB2에 의해 오류(SQL1197N)로 처리됩니다.

### **SET PROFILE NONE**

이 RUNSTATS 호출에 대해 통계 프로파일이 설정되지 않음을 지정합니다.

### **SET PROFILE**

RUNSTATS가 시스템 카탈로그 테이블에서 특정 통계 프로파일을 생성하고 저장할 수 있도록 하며 RUNSTATS 명령 옵션을 실행하여 통계를 수집합니다.

### **SET PROFILE ONLY**

RUNSTATS이 RUNSTATS 명령 옵션을 실행하지 않고 시스템 카탈로그 테이블에 특정 프로파일을 생성 및 저장할 수 있게 합니다.

### **UPDATE PROFILE**

RUNSTATS가 시스템 카탈로그 테이블에서 기존 통계 프로파일을 수정할 수 있도록 하고 갱신된 통계 프로파일의 RUNSTATS 명령 옵션을 실행하여 통계를 수집합니다.

## UPDATE PROFILE ONLY

갱신된 통계 프로파일의 RUNSTATS 명령 옵션을 실행하지 않고 RUNSTATS가 시스템 카탈로그 테이블에서 기존 통계 프로파일을 수정할 수 있도록 합니다.

## UTIL\_IMPACT\_PRIORITY *priority*

RUNSTATS가 *priority*가 지정하는 레벨에서 조절되도록 지정합니다. *priority*는 1 - 100 범위의 수이며 100은 가장 높은 우선순위를 나타내고 1은 가장 낮은 우선순위를 나타냅니다. 이 우선순위는 유틸리티가 필요로 하는 조절 기능의 양입니다. 동일한 우선순위의 유틸리티는 모두 동일한 양으로 조절되지만 하위 우선순위의 유틸리티는 상위 우선순위의 유틸리티보다 더 많이 조절됩니다. *priority*가 지정되지 않으면 RUNSTATS는 디폴트 우선순위 50을 갖게 됩니다. UTIL\_IMPACT\_PRIORITY 키워드를 생략하면 조절 지원이 없이 RUNSTATS 유틸리티를 호출합니다. UTIL\_IMPACT\_PRIORITY 키워드를 지정하지만 **util\_impact\_lim** 구성 매개변수를 100으로 설정할 경우 유틸리티는 조절되지 않고 실행됩니다. 뷰에는 이 옵션을 사용할 수 없습니다.

파티션된 데이터베이스에서 RUNSTATS 명령을 테이블에서 사용하면 단일 데이터베이스 파티션에 대해서만 통계를 수집합니다. RUNSTATS 명령이 실행된 데이터베이스 파티션에 해당 테이블의 파티션이 있으면 명령은 이 데이터베이스 파티션에서 실행됩니다. 그렇지 않으면 명령은 해당 테이블이 파티션된 데이터베이스 파티션 그룹의 첫 번째 데이터베이스 파티션에서 실행됩니다.

## 예:

1. 분산 통계가 없는 모든 컬럼에 대해 해당 테이블에서만 통계를 수집하십시오.  

```
RUNSTATS ON TABLE db2user.employee
```
2. 분산 통계가 없는 empid 및 empname 컬럼에 대해 해당 테이블에서만 통계를 수집하십시오.  

```
RUNSTATS ON TABLE db2user.employee  
WITH DISTRIBUTION ON COLUMNS (empid, empname)
```
3. 구성 설정에서 NUM\_QUANTILES를 선택하면서 테이블의 지정된 빈도 수 한계를 사용하여 분산 통계가 있는 모든 컬럼에 대해 해당 테이블에서만 통계를 수집하십시오.  

```
RUNSTATS ON TABLE db2user.employee WITH DISTRIBUTION DEFAULT  
NUM_FREQVALUES 50
```
4. 인덱스 세트에서 통계를 수집하십시오.  

```
RUNSTATS ON TABLE db2user.employee for indexes  
db2user.emp11, db2user.emp12
```
5. 모든 인덱스에서 기본 통계를 수집하십시오.  

```
RUNSTATS ON TABLE db2user.employee FOR INDEXES ALL
```



6. 자세한 인덱스 통계 컬렉션에 대한 샘플링을 사용하여 테이블 및 모든 인덱스에서 기본 통계를 수집하십시오.

```
RUNSTATS ON TABLE db2user.employee AND SAMPLED DETAILED INDEXES ALL
```

7. 컬럼 empid, empname, empdept 및 두 개의 인덱스 Xempid, Xempname에서 분산 통계와 함께 테이블에서 통계를 수집하십시오. empdept에 대해 분산 통계 한계가 개별적으로 설정되며 다른 두 컬럼은 일반 디폴트값을 사용합니다.

```
RUNSTATS ON TABLE db2user.employee  
  WITH DISTRIBUTION ON COLUMNS (empid, empname, empdept NUM_FREQVALUES  
    50 NUM_QUANTILES 100)  
  DEFAULT NUM_FREQVALUES 5 NUM_QUANTILES 10  
  AND INDEXES db2user.Xempid, db2user.Xempname
```

8. 인덱스에 사용된 모든 컬럼과 모든 인덱스에서 통계를 수집하십시오.

```
RUNSTATS ON TABLE db2user.employee ON KEY COLUMNS AND INDEXES ALL
```

9. 한 컬럼을 제외하고 분산 통계가 없는 모든 컬럼 및 모든 인덱스에서 통계를 수집하십시오. T1에 c1, c2, ..., c8이 들어 있다고 간주하십시오.

```
RUNSTATS ON TABLE db2user.T1  
  WITH DISTRIBUTION ON COLUMNS (c1, c2, c3 NUM_FREQVALUES 20  
    NUM_QUANTILES 40, c4, c5, c6, c7, c8)  
  DEFAULT NUM_FREQVALUES 0, NUM_QUANTILES 0 AND INDEXES ALL
```

```
RUNSTATS ON TABLE db2user.T1  
  WITH DISTRIBUTION ON COLUMNS (c3 NUM_FREQVALUES 20 NUM_QUANTILES 40)  
  AND INDEXES ALL
```

10. 개별 컬럼 c1 및 c5에 대해 테이블 T1에서 및 컬럼 조합 (c2, c3) 및 (c2, c4)에서 통계를 수집하십시오. 다중 컬럼 카디널리티는 쿼리 옵티마이저가 데이터가 상관된 컬럼에서 술어에 대한 필터 인수를 평가할 때 매우 유용합니다.

```
RUNSTATS ON TABLE db2user.T1 ON COLUMNS (c1, (c2, c3),  
  (c2, c4), c5)
```

11. 개별 컬럼 c1 및 c2에 대해 테이블 T1에서 통계를 수집하십시오. 컬럼 c1에 대해서는 LIKE 술어 통계도 수집하십시오.

```
RUNSTATS ON TABLE db2user.T1 ON COLUMNS (c1 LIKE STATISTICS, c2)
```

12. 구성 설정에서 NUM\_QUANTILES를 선택하면서 테이블의 지정된 빈도 수 한계를 사용하여 분산 통계가 있는 모든 컬럼에 대해 해당 테이블에서만 통계를 수집하려면 통계 프로파일을 등록하십시오. 이 명령은 또한 지정된 대로 통계를 갱신합니다.

```
RUNSTATS ON TABLE db2user.employee WITH DISTRIBUTION DEFAULT  
  NUM_FREQVALUES 50 SET PROFILE
```

13. 구성 설정에서 NUM\_QUANTILES를 선택하면서 테이블의 지정된 빈도 수 한계를 사용하여 분산 통계가 있는 모든 컬럼에 대해 해당 테이블에서만 통계를 수집하려면 통계 프로파일을 등록하십시오. 통계는 수집되지 않습니다.

```
RUNSTATS ON TABLE db2user.employee WITH DISTRIBUTION  
  DEFAULT NUM_FREQVALUES 50 SET PROFILE ONLY
```

14. NUM\_FREQVALUES 값을 50에서 30으로 변경하여 이전에 등록된 통계 프로파일을 수정하십시오. 이 명령은 또한 지정된 대로 통계를 갱신합니다.

```
RUNSTATS ON TABLE db2user.employee WITH DISTRIBUTION
      DEFAULT NUM_FREQVALUES 30 UPDATE PROFILE
```

15. NUM\_FREQVALUES 값을 50에서 30으로 변경하여 이전에 등록된 통계 프로파일을 수정하십시오. 통계는 수집되지 않습니다.

```
RUNSTATS ON TABLE db2user.employee WITH DISTRIBUTION
      DEFAULT NUM_FREQVALUES 30 UPDATE PROFILE ONLY
```

16. 컬럼 emp1\_address 및 컬럼 그룹(emp1\_title, emp1\_salary) 옵션을 추가하여 이전에 등록된 통계 프로파일을 수정하십시오. 이 명령은 또한 지정된 대로 통계를 갱신합니다.

```
RUNSTATS ON TABLE db2user.employee
      ON COLUMNS (emp1_address, (emp1_title, emp1_salary))
      UPDATE PROFILE
```

17. 컬럼 emp1\_address 및 컬럼 그룹(emp1\_title, emp1\_salary) 옵션을 추가하여 이전에 등록된 통계 프로파일을 수정하십시오. 통계는 수집되지 않습니다.

```
RUNSTATS ON TABLE db2user.employee
      ON COLUMNS (emp1_address, (emp1_title, emp1_salary))
      UPDATE PROFILE ONLY
```

18. 해당 테이블에 대한 통계 프로파일에 기록된 옵션을 사용하여 테이블에서 통계를 수집하십시오.

```
RUNSTATS ON TABLE db2user.employee USE PROFILE
```

19. 테이블의 카탈로그에 저장된 이전에 등록된 통계 프로파일에 해당하는 RUNSTATS 명령 옵션을 쿼리하십시오.

```
SELECT STATISTICS_PROFILE FROM SYSIBM.SYSTABLES WHERE NAME =
      'EMPLOYEE'
```

20. 행의 30퍼센트에서 분산 통계를 포함하는 통계를 수집하십시오.

```
RUNSTATS ON TABLE db2user.employee WITH DISTRIBUTION
      TABLESAMPLE BERNOULLI(30)
```

21. 통계를 수집할 샘플 세트를 제어하고 동일한 샘플 세트를 반복 사용할 수 있게 하려면 다음과 같이 할 수 있습니다.

```
RUNSTATS ON TABLE db2user.employee WITH DISTRIBUTION
      TABLESAMPLE BERNOULLI(30) REPEATABLE(4196)
```

위와 같은 명령문을 발행하면 중간에 데이터가 변경되지 않는 한 동일한 통계 세트가 생성됩니다.

22. 데이터 페이지의 1.5퍼센트에서 테이블 통계와 인덱스 통계를 수집하십시오. 테이블 데이터 페이지만 샘플링되고 인덱스 페이지는 샘플링되지 않습니다. 이 예에서는 테이블 통계 컬렉션에 1.5퍼센트의 테이블 데이터 페이지가 사용되고 인덱스 통계에는 모든 인덱스 페이지가 사용됩니다.

```
RUNSTATS ON TABLE db2user.employee AND INDEXES ALL TABLESAMPLE SYSTEM(1.5)
```

23. 모든 컬럼에서 분산 통계를 제외하고 통계 뷰에 대한 통계를 수집하십시오.

```
RUNSTATS ON TABLE salesdb.product_sales_view
```

24. 컬럼 category, type 및 product\_key에서 분산 통계를 포함하여 통계 뷰에 대한 통계를 수집하십시오. category 컬럼에 대해 분산 통계 한계가 설정되며 나머지 컬럼은 일반 디폴트값을 사용합니다.

```
RUNSTATS ON TABLE salesdb.product_sales_view  
WITH DISTRIBUTION ON COLUMNS (category NUM_FREQVALUES 100 NUM_QUANTILES 100,  
type, product_key) DEFAULT NUM_FREQVALUES 50 NUM_QUANTILES 50
```

25. 행 레벨 샘플링을 사용하여 10퍼센트의 행에서 분산 통계를 포함하는 통계를 수집하십시오.

```
RUNSTATS ON TABLE db2user.daily_sales  
WITH DISTRIBUTION TABLESAMPLE BERNOULLI (10)
```

26. 데이터 페이지 레벨 샘플링을 사용하여 2.5퍼센트의 행에서 분산 통계를 포함하는 통계를 수집하십시오. 또한 동일한 샘플 세트를 반복 사용하도록 지정하십시오. 이 명령을 완료하려면 DB2 데이터베이스 시스템이 데이터 페이지 샘플링을 하나 이상의 테이블로 이동시킬 수 있도록 쿼리를 작성해야 합니다. 그렇지 않으면 오류 (SQL 20288N)가 발생합니다.

```
RUNSTATS ON TABLE db2user.daily_sales  
WITH DISTRIBUTION TABLESAMPLE SYSTEM (2.5)
```

27. 지정된 대로 분산 통계가 있는 뷰 및 모든 컬럼에서 통계를 수집하려면 통계 프로파일을 등록하십시오.

```
RUNSTATS ON TABLE salesdb.product_sales_view  
WITH DISTRIBUTION DEFAULT NUM_FREQVALUES 50 NUM_QUANTILES 50  
SET PROFILE
```

28. 이전에 등록된 통계 프로파일을 수정합니다. 이 명령은 또한 지정된 대로 통계를 갱신합니다.

```
RUNSTATS ON TABLE salesdb.product_sales_view  
WITH DISTRIBUTION DEFAULT NUM_FREQVALUES 25 NUM_QUANTILES 25  
UPDATE PROFILE
```

## 사용 시 참고사항

1. 파티션된 테이블에 접속이 해제된 파티션이 있을 때, 정리할 필요가 있는 접속 해제된 데이터 파티션에 계속 속해 있는 인덱스 키는 통계에서 키로 계수되지 않습니다. 이들 키는 표시되지 않으며 더 이상 테이블에 포함되어 있지 않으므로 계수되지 않습니다. 결국 비동기 인덱스 정리에 의해 인덱스에서 제거됩니다. 따라서 비동기 인덱스 정리가 실행되기 전에 수집된 통계는 잘못된 것입니다. 비동기 인덱스 정리가 완료되기 전에 RUNSTATS 명령이 발행될 경우 부정확한 통계에 따른 인덱스 재구성 또는 인덱스 정리에 대한 거짓 알람을 생성할 수 있습니다. 비동기 인덱스 정리가 실행된 후, 정리할 필요가 있는 접속 해제된 데이터 파티션에 여전히 속해 있는 인덱스 키는 모두 제거되므로 인덱스를 재구성할 필요가 없도록 합니다.

파티션된 테이블의 경우, 접속 해제된 데이터 파티션이 있는 상태에서 올바른 인덱스 통계를 생성하려면 비동기 인덱스 정리가 완료된 후 RUNSTATS 명령을 발행하는 것이 바람직합니다. 테이블에 접속 해제된 데이터 파티션이 있는지 여부를 판별하려면 SYSDATAPARTITIONS 테이블의 상태 필드를 점검하여 I(인덱스 정리) 또는 D(중속 MQT와 접속 해제됨)를 찾아 볼 수 있습니다.

RUNSTATS 명령은 파티션된 인덱스의 모든 인덱스 파티션의 통계를 수집합니다. 파티션된 인덱스의 SYSTAT.INDEXES 뷰의 통계는 FIRSTKEYCARD, FIRST2KEYCARD, FIRST3KEYCARD, FIRST4KEYCARD 및 FULLKEYCARD 통계를 제외한 인덱스 파티션을 표시합니다. 이 통계는 카디널리티(cardinality) 개략에서 사용되므로, 전체 인덱스에 대한 것이며 인덱스 파티션에 대한 것은 아닙니다. 분산된 인덱스의 분산 통계(자주 사용되는 값 및 quantiles)는 수집되지 않지만, RUNSTATS가 테이블에서 실행되는 경우 수집됩니다. 파티션된 인덱스의 선행 컬럼에 대한 통계는 파티션되지 않은 인덱스의 선행 컬럼에 대한 통계만큼 정확하지 않을 수 있습니다.

## 2. RUNSTATS 명령을 실행하는 것이 좋습니다.

- 상당한 부분이 수정된 테이블(예를 들어, 여러 번 갱신되었거나 상당한 데이터량이 삽입 또는 삭제된 경우 또는 LOAD 중 통계 옵션을 사용하지 않고 LOAD가 수행된 경우)
- 재구성된 테이블(REORG, REDISTRIBUTE DATABASE PARTITION GROUP을 사용하여).
- 행이 압축된 테이블
- 새 인덱스가 작성된 경우
- 성능이 중요한 응용프로그램을 바인드하기 전
- 프리페치량이 변경된 경우
- 기본 테이블이 실질적으로 수정되어 뷰에서 리턴된 행이 변경된 통계 뷰
- STATISTICS 옵션을 사용하여 LOAD가 실행된 후 RUNSTATS 유틸리티를 사용하여 XML 컬럼에서 통계를 수집하십시오. LOAD가 STATISTICS 옵션을 사용하여 실행될 때라도 LOAD 중 XML 컬럼에 대한 통계는 결코 수집되지 않습니다. 단지 XML 컬럼에 대한 통계를 수집하기 위해 RUNSTATS를 사용할 때 LOAD 또는 이전에 RUNSTATS 유틸리티를 실행하여 수집한 비XML 컬럼에 대한 기존 통계는 보유됩니다. 일부 XML 컬럼에 대한 통계가 이전에 수집된 경우, 현재 명령으로 해당 XML 컬럼에 통계가 수집되지 않으면 XML 컬럼에 대해 이전에 수집된 통계는 삭제되며 현재 명령으로 해당 XML 컬럼에 대한 통계가 수집되면 이전에 수집된 통계를 대체합니다.

## 3. 선택된 옵션은 특정 테이블 및 응용프로그램에 따라 달라야 합니다. 일반적으로 다음과 같습니다.

- 테이블이 중요한 쿼리에서 매우 중요한 테이블이고 비교적 작은 편이거나 너무 많이 변경되지 않으며 시스템 자체에서 활동이 너무 많지 않은 경우, 가능한 자세한 통계를 수집하는 데 노력을 기울일 만한 가치가 있을 수 있습니다.
  - 통계를 수집할 시간이 제한된 경우 테이블이 비교적 크거나 테이블이 자주 갱신되는 경우 술어에 사용된 컬럼 세트에 제한된 RUNSTATS를 실행하는 것이 유용할 수 있습니다. 이 같이 사용자는 RUNSTATS 명령을 좀 더 자주 실행할 수 있게 됩니다.
  - 통계 수집 시간이 매우 제한적이고 테이블 기반으로 테이블에 대해 RUNSTATS 명령을 조정하기 위한 노력이 주요 관건이 될 경우 "KEY" 컬럼에 대해서만 통계를 수집하는 것을 고려하십시오. 인덱스가 테이블에 중요한 컬럼 세트를 포함하고 있으며 필시 술어에 표시될 것으로 간주됩니다.
  - 통계 수집 시간이 매우 제한적이고 테이블 통계가 수집될 경우, TABLESAMPLE 옵션을 사용하여 테이블 데이터의 서브세트에 대한 통계를 수집하는 것을 고려하십시오.
  - 테이블에 많은 인덱스가 있으며 인덱스에 대한 DETAILED(확장) 정보가 액세스 플랜을 개선할 수 있는 경우, SAMPLED 옵션을 사용하여 통계를 수집하는 데 걸리는 시간을 줄여볼 것을 고려하십시오.
  - "column = constant" 유형의 특정 컬럼 및 술어에 편차가 있는 경우, 해당 컬럼에 더 큰 NUM\_FREQVALUES 값을 지정하는 것이 유용할 수 있습니다.
  - 등호 술어에 사용되고 분산 값이 편차가 있을 수 있는 모든 컬럼에 대해 분산 통계를 수집하십시오.
  - 범위 술어(예: "column >= constant", "column BETWEEN constant1 AND constant2") 또는 "column LIKE '%xyz'" 유형을 갖는 컬럼의 경우, 더 큰 NUM\_QUANTILES 값을 지정하는 것이 유용할 수 있습니다.
  - 스토리지 스페이스가 문제가 있고 통계를 수집하는 데 너무 많은 시간을 제공할 수 없는 경우 술어에 사용되지 않는 컬럼에 대해 높은 NUM\_FREQVALUES 또는 NUM\_QUANTILES 값을 지정하지 마십시오.
  - 인덱스 통계가 요청되었으며 인덱스가 있는 테이블에서 통계가 실행된 적이 없는 경우, 테이블 및 인덱스에 대한 통계가 계산됩니다.
  - 테이블의 XML 컬럼에 대한 통계가 필요하지 않은 경우 EXCLUDING XML COLUMNS 옵션을 사용하여 모든 XML 컬럼을 제외시킬 수 있습니다. 이 옵션은 통계 컬렉션에 XML 컬럼을 지정하는 다른 모든 절보다 우선순위를 갖습니다.
4. 명령 실행 이후에 다음을 참고하십시오.
- COMMIT를 발행하여 잠금을 릴리스해야 합니다.
  - 새 액세스 플랜이 생성될 수 있게 하려면 목표 테이블을 참조하는 패키지를 리바인드해야 합니다.

- 테이블의 부분에서 명령을 실행하면 명령이 마지막 발행된 후 테이블에 대한 활동의 결과로 불일치한 결과가 나타날 수 있습니다. 이 경우 경고 메시지가 리턴됩니다. 테이블에서만 RUNSTATS를 실행하면 테이블 및 인덱스 레벨 통계가 불일치할 수 있습니다. 예를 들어, 테이블에서 인덱스 레벨 통계를 수집한 후 테이블에서 상당한 수의 행을 삭제할 수 있습니다. 그런 다음 테이블에 대해서만 RUNSTATS를 발행하면 테이블 카디널리티가 FIRSTKEYCARD 보다 적을 수 있으며 불일치하게 됩니다. 마찬가지로 인덱스를 작성시 새 인덱스에서 통계를 수집하면 테이블 레벨 통계가 불일치할 수 있습니다.
5. 테이블 통계가 요청될 경우 RUNSTATS 명령은 이전에 수집된 분산 통계를 삭제합니다. 예를 들어, RUNSTATS ON TABLE 또는 RUNSTATS ON TABLE ... AND INDEXES ALL은 이전에 수집된 분산 통계가 삭제되게 합니다. 명령이 인덱스에 대해서만 실행될 경우 이전에 수집된 분산 통계는 보유됩니다. 예를 들어, RUNSTATS ON TABLE ... FOR INDEXES ALL은 이전에 수집된 분산 통계가 보유되게 합니다. RUNSTATS 명령이 XML 컬럼에 대해서만 실행될 경우 이전에 수집된 기본 컬럼 통계 및 분산 통계는 보유됩니다. 일부 XML 컬럼에 대한 통계가 이전에 수집된 경우, 현재 명령으로 해당 XML 컬럼에 통계가 수집되지 않으면 XML 컬럼에 대해 이전에 수집된 통계는 삭제되며 현재 명령으로 해당 XML 컬럼에 대한 통계가 수집되면 이전에 수집된 통계를 대체합니다.
  6. 범위 클러스터 테이블의 경우, 이 테이블의 범위 순서화 등록 정보를 나타내는 특수 시스템 생성 인덱스가 카탈로그 테이블에 들어 있습니다. 이 유형의 테이블에서 통계가 수집될 때 테이블이 통계 컬렉션의 일부로 포함될 경우, 시스템 생성 인덱스에 대해서도 통계가 수집됩니다. 이 통계는 인덱스를 기본 데이터 테이블만큼 많은 페이지가 있는 두 레벨 인덱스로 나타내고 인덱스 순서에 따라 기본 데이터를 완벽하게 클러스터되게 하여 범위 찾아보기의 빠른 액세스를 반영합니다.
  7. 명령 구문의 Dist Cols 절에서 빈도 옵션 및 Quantile 옵션 매개변수는 현재 컬럼 GROUPS에서 지원되지 않습니다. 이들 옵션은 단일 컬럼에만 지원됩니다.
  8. DMS 모드에서 작업할 때 계산될 수 없는 세 가지 프리페치 통계가 있습니다. 인덱스 카탈로그에서 인덱스 통계를 보면 다음 통계의 경우 -1값이 표시됩니다.
    - AVERAGE\_SEQUENCE\_FETCH\_PAGES
    - AVERAGE\_SEQUENCE\_FETCH\_GAP
    - AVERAGE\_RANDOM\_FETCH\_PAGES
  9. TABLESAMPLE을 통한 RUNSTATS 샘플링은 테이블 데이터 페이지에서만 발생하며 인덱스 페이지에서는 발생하지 않습니다. 샘플링과 함께 인덱스 통계가 요청되면 통계 수집을 위해 모든 인덱스 페이지가 스캔됩니다. TABLESAMPLE은 테이블 통계의 컬렉션에서만 적용됩니다. 그러나 SAMPLED DETAILED 옵션을 통해 보다 효율적인 자세한 인덱스 통계 컬렉션을 사용할 수 있습니다. 이는 TABLESAMPLE에서 사용된 것과 다른 샘플링 방법으로 자세한 인덱스 통계 세트에만 적용됩니다.

10. 프로파일 설정 또는 프로파일 갱신 옵션을 사용하여 RUNSTATS 명령에 지정된 테이블 또는 통계 뷰에 대한 통계 프로파일을 설정하거나 갱신할 수 있습니다. 통계 프로파일은 SYSIBM.SYSTABLES 시스템 카탈로그 테이블의 STATISTICS\_PROFILE 컬럼에서 RUNSTATS 명령을 나타내는 표시 가능한 문자열 형식으로 저장됩니다.
11. XML 유형 컬럼에 대한 통계 컬렉션은 다음과 같은 두 개의 DB2 데이터베이스 시스템 레지스트리 값으로 제어됩니다. DB2\_XML\_RUNSTATS\_PATHID\_K 및 DB2\_XML\_RUNSTATS\_PATHVALUE\_K. 이들 두 매개변수는 수집할 빈도 수 값을 지정한다는 점에서 NUM\_FREQVALUES 매개변수와 유사합니다. 이들 매개변수를 설정하지 않으면 디폴트값 200이 두 매개변수에 사용됩니다.
12. RUNSTATS는 SYSTABLES에 대해서는 IX 테이블 잠금을 획득하고 RUNSTATS가 시작될 때 stats가 수집되는 테이블의 행에 대해서는 U 잠금을 획득합니다. U 잠금이 설정된 행을 포함하는 SYSTABLES에서 여전히 읽기 조작을 수행할 수 있습니다. U 잠금이 설정된 행에 대해 발생하지 않았으면 쓰기 조작도 가능합니다. 그러나 다른 판독기나 기록기는 RUNSTATS' IX 잠금 때문에 SYSTABLES에서 S 잠금을 획득할 수 없습니다.
13. 구조화된 유형이 있는 컬럼에 대한 통계는 수집되지 않습니다. 통계가 지정되면, 해당 데이터 유형이 있는 컬럼은 무시됩니다.
14. LOB 또는 LONG 데이터 유형이 있는 컬럼에 대한 AVGCOLLEN 및 NUMNULLS만 수집됩니다.
15. 컬럼이 데이터베이스 메모리 또는 임시 테이블에 저장될 때 AVGCOLLEN은 바이트 단위로 평균 스페이스를 표시합니다. LOB 데이터가 데이터 페이지에서 인라인될 때를 제외하고 이 값은 LOB 또는 LONG 데이터 유형의 데이터 디스크립터의 길이를 표시합니다.

주: 디스크에서 컬럼을 저장하기 위해 필요한 평균 스페이스는 이 통계로 표시되는 값과 다를 수 있습니다.





---

## 제 116 장 SET CLIENT

백엔드 프로세스에 대한 연결 설정을 지정합니다.

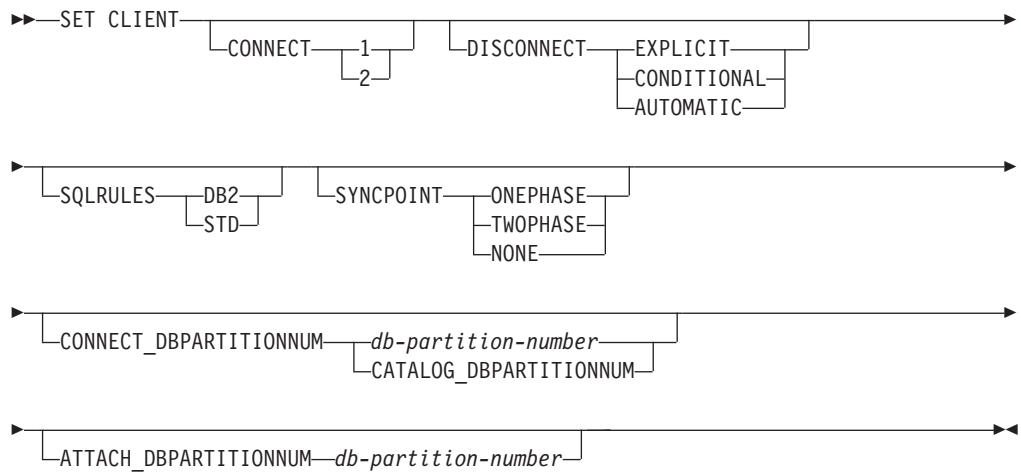
### 권한 부여

없음

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

#### CONNECT

- 1 CONNECT문이 유형 1 CONNECT로서 처리되도록 지정합니다.
- 2 CONNECT문이 유형 2 CONNECT로서 처리되도록 지정합니다.

#### DISCONNECT

##### EXPLICIT

RELEASE문에 의한 릴리스 대상으로 명시적으로 표시된 데이터베이스 연결만 커밋 시 연결이 끊어지도록 지정합니다.

##### CONDITIONAL

RELEASE 표시되었거나 열린 WITH HOLD 커서가 없는 데이터베이스 연결만 커밋 시 연결이 끊어지도록 지정합니다.

## **AUTOMATIC**

모든 데이터베이스 연결이 커밋 시 연결이 끊어지도록 지정합니다.

## **SQLRULES**

**DB2** 유형 2 CONNECT가 DB2 규칙에 따라서 처리되도록 지정합니다.

**STD** 유형 2 CONNECT가 ISO/ANS SQL92를 기초로 표준(STD) 규칙에 따라서 처리되도록 지정합니다.

## **SYNCPOINT**

커밋 또는 롤백이 여러 데이터베이스 연결 사이에 조정되는 방식을 지정합니다. 이 명령 매개변수는 무시되며 이전 버전과의 호환성을 위해서만 여기에 포함됩니다.

## **ONEPHASE**

트랜잭션 관리 프로그램(TM)이 2단계 커밋을 수행하는 데 사용되지 않도록 지정합니다. 1단계 커밋이 다중 데이터베이스 트랜잭션에서 각 데이터베이스에 의해 수행된 작업을 커밋하는 데 사용됩니다.

## **TWOPHASE**

TM이 이 프로토콜을 지원하는 데이터베이스 사이의 2단계 커밋을 조정하는 데 필요하다고 지정합니다.

## **NONE**

TM이 2단계 커밋을 수행하는 데 사용되지 않으며, 단일 갱신자, 다중 관독기를 강제하지 않도록 지정합니다. COMMIT를 참여하는 각 데이터베이스로 보냅니다. 커밋 중 하나가 실패하는 경우 응용프로그램이 복구를 수행해야 합니다.

## **CONNECT\_DBPARTITIONNUM(파티션된 데이터베이스 환경만 해당)**

### *db-partition-number*

연결을 작성할 데이터베이스 파티션을 지정합니다. 0과 999 사이의 값입니다. 환경 변수 DB2NODE의 값을 겹쳐씹니다.

## **CATALOG\_DBPARTITIONNUM**

이 값을 지정하면 해당 데이터베이스 파티션의 ID를 미리 알지 않고 클라이언트가 데이터베이스의 카탈로그 데이터베이스 파티션에 연결할 수 있습니다.

## **ATTACH\_DBPARTITIONNUM *db-partition-number*(파티션된 데이터베이스 환경만 해당)**

접속을 작성할 데이터베이스 파티션을 지정합니다. 0과 999 사이의 값입니다. 환경 변수 DB2NODE의 값을 겹쳐씹니다.

예를 들어 데이터베이스 파티션 1, 2 및 3이 정의되는 경우 클라이언트는 이들 데이터베이스 파티션 중 하나에 액세스할 수 있어야 합니다. 데이터베이스가 들

어있는 데이터베이스 파티션 1만 카탈로그되었고 이 매개변수가 3으로 설정된 경우, 다음 접속 시도는 데이터베이스 파티션 1에서의 초기 접속 후 데이터베이스 파티션 3에서의 접속입니다.

## 예:

특정 값을 설정하려면 다음을 수행하십시오.

```
db2 set client connect 2 disconnect automatic sqlrules std
syncpoint twophase
```

SQLRULES를 다시 DB2로 변경하지만 다른 설정을 유지하려면 다음을 수행하십시오.

```
db2 set client sqlrules db2
```

TERMINATE 명령이 발행된 후 연결 설정은 디폴트값으로 되돌아갑니다.

## 사용 시 참고사항

하나 이상의 연결이 활성 상태인 경우 SET CLIENT를 발행할 수 없습니다.

SET CLIENT가 성공하는 경우 후속 작업 단위(UOW)의 연결은 지정된 연결 설정을 사용합니다. SET CLIENT가 실패하면 백엔드 프로세스의 연결 설정은 변경되지 않습니다.

파티션된 데이터베이스 환경(DPF)에서, 연결 설정은 트러스트된 연결을 얻는 데 영향을 줄 수 있습니다. 예를 들어 CONNECT\_DBPARTITIONNUM 옵션이 노드로 설정되어 해당 노드에서 연결 설정이 중간 노드(홉 노드) 통과를 필요로 하는 경우, 중간 노드의 IP 주소 및 홉 노드와 트러스트된 연결로 표시될 수 있는지 여부를 판별하기 위해 이 연결을 평가할 때 고려되는 연결 노드 사이에 통신하는 데 사용되는 통신 프로토콜입니다. 즉 고려되는 연결이 시작된 원래 노드가 아닙니다. 그보다는, 고려되는 홉 노드입니다.

## 호환성

버전 8 이전 버전과의 호환성:

- CONNECT\_DBPARTITIONNUM에 키워드 CONNECT\_NODE를 대체할 수 있습니다.
- CATALOG\_DBPARTITIONNUM에 키워드 CATALOG\_NODE를 대체할 수 있습니다.
- ATTACH\_DBPARTITIONNUM에 키워드 ATTACH\_NODE를 대체할 수 있습니다.



---

## 제 117 장 SET RUNTIME DEGREE

지정된 활성 응용프로그램에 대한 SQL문의 파티션 내 병렬 처리의 최대 런타임 등급을 설정합니다.

### 범위

이 명령은 \$HOME/sql1lib/db2nodes.cfg 파일에 나열된 모든 데이터베이스 파티션에 영향을 줍니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*

### 필수 연결

인스턴스. 리모트 서버에서 파티션 내 병렬 처리의 최대 런타임 등급을 변경하려면 먼저 해당 서버에 접속해야 합니다. 접속이 존재하지 않는 경우 SET RUNTIME DEGREE 명령은 실패합니다.

### 명령 구문

```
▶▶ SET RUNTIME DEGREE FOR ALL TO degree
    (application-handle)
```

### 명령 매개변수

#### FOR

**ALL** 지정된 등급이 모든 응용프로그램에 적용됩니다.

*application-handle*

새 등급이 적용되는 에이전트를 지정합니다. LIST APPLICATIONS 명령을 사용하여 값을 나열하십시오.

#### TO *degree*

파티션 내 병렬 처리의 최대 런타임 등급입니다.

예:

다음 예는 41408 및 55458의 *application-handle* 값을 갖는 두 사용자에게 대한 병렬 처리 수준 최대 런타임 등급을 4로 설정합니다.

```
db2 SET RUNTIME DEGREE FOR ( 41408, 55458 ) TO 4
```

## 사용 시 참고사항

이 명령은 활성 응용프로그램에 대한 최대 병렬 처리 수준을 수정하는 메커니즘을 제공합니다. SQL문 컴파일 시에 결정된 값을 대체하는 데 사용할 수 있습니다.

파티션 내 병렬 처리의 런타임 등급은 명령문이 실행될 때 사용될 최대 병렬 작업 수를 지정합니다. SQL문에 대한 파티션 내 병렬 처리 수준은 **CURRENT DEGREE** 특수 레지스터 또는 **DEGREE** 바인드 옵션을 사용하여 명령문 컴파일 시에 지정할 수 있습니다. 활성 응용프로그램에 대한 파티션 내 병렬 처리의 최대 런타임 등급은 **SET RUNTIME DEGREE** 명령을 사용하여 지정할 수 있습니다. *max\_querydegree* 데이터베이스 관리 프로그램 구성 매개변수가 데이터베이스 관리 프로그램의 이 인스턴스에 대해 실행하는 SQL문에 대한 최대 런타임 등급을 지정합니다.

실제 런타임 등급은 다음 중에서 가장 낮은 등급입니다.

- **max\_querydegree** 구성 매개변수
- 응용프로그램 런타임 등급
- SQL문 컴파일 등급.

---

## 제 118 장 SET SERVEROUTPUT 명령

DBMS\_OUTPUT 메시지 버퍼로부터의 출력 경로가 표준 출력으로 재지정되는지 여부를 지정합니다.

### 권한 부여

DBMS\_OUTPUT 모듈에 대한 EXECUTE 특권

### 필수 연결

데이터베이스

### 명령 구문

▶▶ SET SERVEROUTPUT  OFF  ON ◀◀

### 명령 매개변수

#### ON

메시지 버퍼의 메시지 경로가 표준 출력으로 재지정됨을 지정합니다.

#### OFF

메시지 버퍼의 메시지 경로가 표준 출력으로 재지정되지 않음을 지정합니다.

### 예:

DBMS\_OUTPUT 메시지 버퍼의 메시지 경로를 표준 출력으로 재지정하려면 SET SERVEROUTPUT ON을 지정하십시오. 다음 예에서, PUT 프로시저는 부분 행을 DBMS\_OUTPUT 메시지 버퍼에 추가합니다. proc1이 실행될 때, SET SERVEROUTPUT ON이 지정되어 있으므로 DBMS\_OUTPUT 메시지 버퍼에 저장된 텍스트가 표시됩니다.

```
SET SERVEROUTPUT ON@

DROP PROCEDURE proc1@

CREATE PROCEDURE proc1()
  BEGIN
    CALL DBMS_OUTPUT.PUT( 'p1 = ' || p1 );
    CALL DBMS_OUTPUT.PUT( 'p2 = ' || p2 );
    CALL DBMS_OUTPUT.NEW_LINE;
  END@

CALL proc1( 10, 'Peter' )@

SET SERVEROUTPUT OFF@
```

이 예의 결과는 다음 출력과 같습니다.

```
SET SERVEROUTPUT ON
DB20000I The SET SERVEROUTPUT command completed successfully.
```

```
DROP PROCEDURE PROC1
DB20000I The SQL command completed successfully.
```

```
CREATE PROCEDURE proc1()
  BEGIN
    CALL DBMS_OUTPUT.PUT( 'p1 = ' || p1 );
    CALL DBMS_OUTPUT.PUT( 'p2 = ' || p2 );
    CALL DBMS_OUTPUT.NEW_LINE;
  END@
DB20000I The SQL command completed successfully.
```

```
CALL proc1( 10, 'Peter' )@
```

```
Return Status = 0
```

```
p1 = 10
p2 = Peter
```

```
SET SERVEROUTPUT OFF
DB20000I The SET SERVEROUTPUT command completed successfully.
```

## 사용 시 참고사항

메시지는 PUT, PUT\_LINE 및 NEW\_LINE 프로시저에 의해 DBMS\_OUTPUT 메시지 버퍼에 추가됩니다.

명령 SET SERVEROUTPUT ON이 실행될 때, 명령은 기본 버퍼 크기가 2000바이트인 DBMS\_OUTPUT.ENABLE 프로시저를 호출하고 클라이언트 응용프로그램에서 내부 플래그를 설정합니다. 이 플래그가 사용 가능한 경우, 클라이언트 응용프로그램은 각 SELECT 또는 CALL 문을 실행한 후 GET\_LINES 프로시저를 호출하고 메시지를 메시지 버퍼에서 표준 출력으로 경로 재지정합니다. DBMS\_OUTPUT 버퍼 크기를 늘리려면 SET SERVER OUTPUT ON 실행 후 더 큰 버퍼 크기로

DBMS\_OUTPUT.ENABLE 프로시저를 호출하십시오(예: CALL DBMS\_OUTPUT.ENABLE( 50000 );).

SET SERVEROUTPUT OFF 명령이 실행될 때 이 명령은 DBMS\_OUTPUT.DISABLE 프로시저를 호출합니다. 메시지 버퍼에 있는 메시지는 버려지고 PUT, PUT\_LINE 및 NEW\_LINE 프로시저에 대한 호출은 무시됩니다. DBMS\_OUTPUT.GET\_LINES 프로시저는 각각의 SELECT 또는 CALL 문 이후에 호출되지 않습니다.



## 제 119 장 SET TABLESPACE CONTAINERS

경로 재지정된 리스토어는 리스토어된 데이터베이스에 대한 테이블 스페이스 컨테이너 세트가 백업 수행 시 원래 데이터베이스의 컨테이너 세트와 다른 리스토어입니다. 이 명령은 리스토어되는 데이터베이스에 대한 테이블 스페이스 컨테이너의 추가, 변경 또는 제거를 허가합니다. 예를 들어, 하나 이상의 컨테이너가 어떤 이유로 액세스할 수 없게 되는 경우, 다른 컨테이너로 경로가 재지정되지 않으면 리스토어가 실패합니다.

이 명령을 사용하여 기존의 일반 또는 대형 데이터베이스 관리 테이블 스페이스가 자동 스토리지를 사용하도록 변환할 수 있습니다. 또한 데이터베이스가 사용 가능한 스토리지 경로에서 기존의 자동 스토리지 테이블 스페이스를 더 균일하게 다시 스트라이프하기 위해 사용할 수도 있습니다.

### 권한 부여

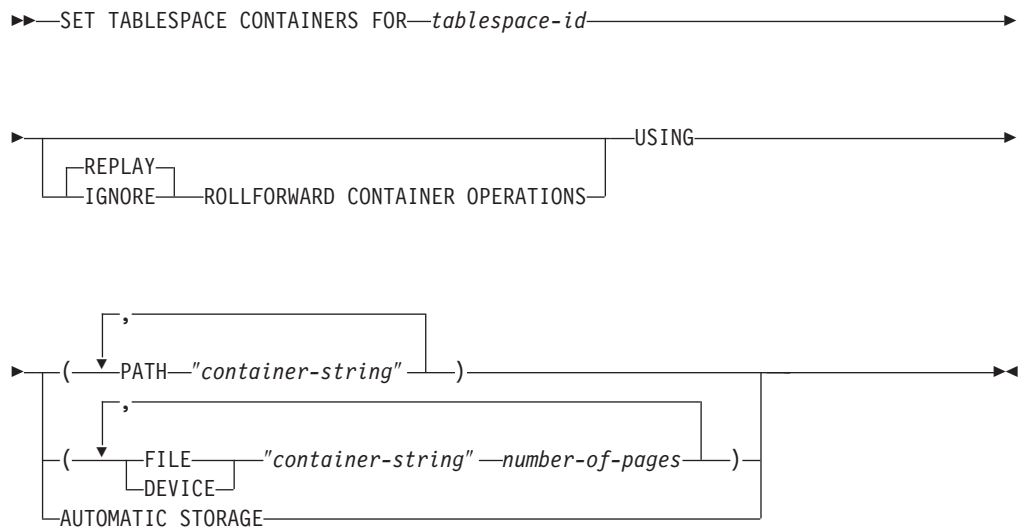
다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*

### 필수 연결

데이터베이스

### 명령 구문



## 명령 매개변수

### **FOR** *tablespace-id*

리스토어하는 데이터베이스에서 사용하는 테이블 스페이스를 고유하게 나타내는 정수.

### **REPLAY ROLLFORWARD CONTAINER OPERATIONS**

데이터베이스가 백업된 이후에 해당 테이블 스페이스에 대해 발행된 ALTER TABLESPACE 조작이 데이터베이스의 연속 롤 포워드 중에 재실행됩니다.

### **IGNORE ROLLFORWARD CONTAINER OPERATIONS**

롤 포워드를 수행할 때 로그에서의 ALTER TABLESPACE 조작이 무시됨을 지정합니다.

### **USING PATH** *"container-string"*

SMS 테이블 스페이스의 경우, 테이블 스페이스에 속하면서 테이블 스페이스의 데이터가 저장될 하나 이상의 컨테이너를 식별합니다. 절대 또는 상대적 디렉토리 이름입니다. 디렉토리 이름이 절대적이 아니면 데이터베이스 디렉토리에 대해 상대적입니다. 문자열 길이는 240바이트를 초과할 수 없습니다.

### **USING FILE | DEVICE** *"container-string" number-of-pages*

DMS 테이블 스페이스의 경우, 테이블 스페이스에 속하면서 테이블 스페이스의 데이터가 저장될 하나 이상의 컨테이너를 식별합니다. 컨테이너의 유형(FILE 또는 DEVICE) 및 크기가 지정됩니다. 파일 및 디바이스 컨테이너의 혼합을 지정할 수 있습니다. 문자열 길이는 254바이트를 초과할 수 없습니다.

파일 컨테이너의 경우, 문자열은 절대 또는 상대적 파일 이름이어야 합니다. 이름이 절대적이 아니면 데이터베이스 디렉토리에 상대적입니다.

디바이스 컨테이너의 경우, 문자열은 디바이스 이름이어야 합니다. 디바이스는 이미 존재하고 있어야 합니다.

### **USING AUTOMATIC STORAGE**

테이블 스페이스가 자동 스토리지를 사용하도록 변환되어야 하고 데이터베이스가 사용 가능한 스페이스 경로에 새 컨테이너를 작성함을 지정합니다. 테이블 스페이스 경로가 자동 스토리지로 재지정되면, 어떤 컨테이너 조작도 테이블 스페이스에 적용될 수 없습니다.

이 옵션은 자동 스토리지에서 이미 관리되는 테이블 스페이스의 컨테이너를 재정의하여 기존 스토리지 경로 사이에 더 나은 스트라이핑을 제공하기 위해 사용할 수도 있습니다.

주: 테이블 스페이스는 리스토어되는 동안 오프라인 상태가 됩니다.

이 옵션은 시스템 관리 테이블 스페이스에 대해 지원되지 않습니다.

예:

RESTORE DATABASE의 예를 참조하십시오.

## 사용 시 참고사항

데이터베이스 또는 하나 이상의 테이블 스페이스 백업은 백업하는 테이블 스페이스가 사용 중인 모든 테이블 스페이스 컨테이너 레코드를 보존합니다. 리스토어 중, 백업에 나열된 모든 컨테이너를 점검하여 현재 존재하며 액세스 가능한지 확인합니다. 어떤 이유로 하나 이상의 컨테이너에 액세스할 수 없는 경우 리스토어가 실패합니다. 이와 같은 경우에 리스토어를 허용하기 위해, 리스토어 중에 테이블 스페이스 컨테이너의 경로 재지정이 지원됩니다. 이러한 지원에는 테이블 스페이스 컨테이너 추가, 변경 또는 제거가 포함됩니다. 이는 사용자가 해당 컨테이너를 추가, 변경 또는 제거할 수 있도록 허용하는 명령입니다.

USING AUTOMATIC STORAGE 옵션과 함께 지정된 경우 IGNORE/REPLAY ROLLFORWARD CONTAINER OPERATIONS 옵션은 무시됩니다.

SET TABLESPACE CONTAINERS 명령의 USING AUTOMATIC STORAGE 옵션을 사용하는 다중 파티션 환경에서의 테이블 스페이스의 경로 재지정된 리스토어만 리스토어되는 파티션에서 테이블 스페이스를 자동 스토리지로 변환합니다. 다른 데이터베이스 파티션의 컨테이너는 재정의하지 않습니다.

주: 다른 데이터베이스 파티션에서 컨테이너를 재정의하지 않으면, 테이블 스페이스 정의가 각 파티션에서 다릅니다. 나중에 데이터베이스 파티션을 추가할 때 LIKE DBPARTITIONNUM 옵션과 함께 ADD DBPARTITIONNUM 명령을 사용하십시오. 이 옵션에서 선택된 데이터베이스 파티션에 따라, 새 데이터베이스 파티션은 자동 스토리지로 정의된 테이블 스페이스나 자동 스토리지 없이 정의된 테이블 스페이스를 갖습니다. 테이블 스페이스 정의에서의 불일치와 새 데이터베이스 파티션이 추가될 때마다 정의 사이에 결정해야 하는 필요성 모두를 제거하려면, 테이블 스페이스 정의가 모든 데이터베이스 파티션에서 동일한지 확인하십시오. 예를 들어, 모든 데이터베이스 파티션이 SET TABLESPACE CONTAINERS 명령의 USING AUTOMATIC STORAGE 옵션을 사용하여 경로 재지정된 리스토어를 따른 경우, 테이블 스페이스는 모든 데이터베이스 파티션에서 자동 스토리지로 변환됩니다. 나중에 다른 데이터베이스 파티션을 추가하면 다른 데이터베이스 파티션에서 발견된 것과 같은 테이블 스페이스 정의를 갖습니다.



---

## 제 120 장 SET TAPE POSITION

백업 및 리스토어 작업을 위한 테이프의 위치를 스트리밍 테이프 디바이스로 설정하십시오. 이 명령은 Windows 운영 체제에서만 지원됩니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*

### 필수 연결

없음

### 명령 구문

```
▶▶—SET TAPE POSITION—┬—ON— device—┬—TO— position—▶▶
```

### 명령 매개변수

#### **ON** *device*

유효한 테이프 디바이스 이름을 지정합니다. 디폴트값은 **WWW.WTAPE0**입니다.

#### **TO** *position*

테이프가 위치 지정될 표시를 지정합니다. Windows용 DB2는 모든 백업 이미지에 다음에 테이프 표시를 씁니다. 값 1은 첫 번째 위치를 지정하고 2는 두 번째 위치를 지정하며 계속 이와 같이 지정합니다. 테이프가 테이프 표시 1에 위치 지정된 경우 예를 들어 아카이브 2가 리스토어되도록 위치 지정됩니다.



---

## 제 121 장 SET UTIL\_IMPACT\_PRIORITY

실행하는 유틸리티에 대한 영향 설정을 변경합니다. 이 명령을 사용하여 다음을 수행할 수 있습니다.

- 조절되지 않는 모드에서 호출된 유틸리티 조절
- 조절된 유틸리티 조절 취소(조절 기능 사용 안함)
- 조절된 유틸리티의 우선순위 재지정(여러 개의 동시 조절된 유틸리티를 실행하는 경우에 유용함)

### 범위

#### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*
- *sysmon*

#### 필수 연결

인스턴스 로컬 머신에 두 개 이상의 파티션이 있는 경우 올바른 파티션에 접속해야 합니다. 예를 들어, 두 개의 파티션이 있고 LIST UTILITIES 명령 결과 다음 출력이 생성되었다고 가정합니다.

```
ID = 2
유형 = BACKUP
데이터베이스 이름 = IWZ
파티션 번호 = 1
설명 = online db
시작 시간 = 07/19/2007 17:32:09.622395
상태 = 실행 중
호출 유형 = 사용자
조절 기능:
우선순위 = 조절되지 않음
진행 모니터링:
예상된 백분율 완료 = 10
전체 작업 = 97867649689바이트
완료 작업 = 10124388481바이트
```

ID가 2인 유틸리티에 대해 SET UTIL\_IMPACT\_PRIORITY 명령을 발행하려면 파티션 1에 대해 인스턴스 접속이 작성되어야 합니다. 이와 같이 하려면 환경에서 DB2NODE=1을 설정한 후 인스턴스 접속 명령을 발행하십시오.

## 명령 구문

```
►►—SET UTIL_IMPACT_PRIORITY FOR—utility-id—TO—priority—◀◀
```

## 명령 매개변수

*utility-id*

영향 설정을 갱신할 유틸리티의 ID. 실행하는 유틸리티의 ID는 LIST UTILITIES 명령으로 얻을 수 있습니다.

**TO** *priority*

유틸리티 실행과 연관되는 영향에 대한 인스턴스 레벨 한계를 지정합니다. 100 값은 가장 높은 우선순위를 나타내고 1은 가장 낮은 우선순위를 나타냅니다. *priority*를 0으로 설정하면 조절 유틸리티는 강제로 계속 조절되지 않습니다. *priority*를 0이 아닌 값으로 설정하면 조절되지 않는 유틸리티가 강제로 계속 조절 모드에 있게 됩니다.

## 예:

다음 예는 ID가 2인 유틸리티를 조절하지 않습니다.

```
SET UTIL_IMPACT_PRIORITY FOR 2 TO 0
```

다음 예는 ID가 3인 유틸리티를 우선순위 10으로 조절합니다. 변경 이전에 우선순위가 0인 경우 이전에 조절 취소된 유틸리티는 이제 조절됩니다. 유틸리티가 이전에 조절된 경우(우선순위가 0보다 큰 값으로 설정됨), 유틸리티 우선순위가 재지정되었습니다.

```
SET UTIL_IMPACT_PRIORITY FOR 3 TO 10
```

## UTIL\_IMPACT\_LIM 및 UTIL\_IMPACT\_PRIORITY 설정 사이의 관계

데이터베이스 관리 프로그램 구성 매개변수 **util\_impact\_lim**은 조절 유틸리티가 머신의 전체 워크로드에 대해 가질 수 있는 영향에 대한 한계를 설정합니다. 0 - 99는 조절된 백분율이고 100은 조절되지 않습니다.

SET UTIL\_IMPACT\_PRIORITY 명령은 특정 유틸리티가 **util\_impact\_lim** 구성 매개변수에 정의된 대로 조절 유틸리티에 사용 가능한 자원에 대해 갖는 우선순위를 설정합니다(0 = 조절 취소).

예와 같이 백업 유틸리티를 사용할 때 **util\_impact\_lim=10**이면 모든 유틸리티는 조절 알고리즘에서 판단된 총 워크로드에 대해 10% 평균 영향을 초과할 수 없습니다. 예와 같이 두 개의 조절 유틸리티를 사용하는 경우:

- util\_impact\_priority 70의 백업
- util\_impact\_priority 50의 실행 상태



결합된 두 유틸리티 모두 총 워크로드에 대해 10% 평균 영향을 초과할 수 없으며 우선순위가 더 높은 유틸리티는 더 많은 사용 가능한 워크로드 자원을 가져옵니다. 백업 및 실행 상태 조작 둘 다에 대해, 해당 유틸리티의 명령행 안에서 영향 우선순위를 선언할 수도 있습니다. SET UTIL\_IMPACT\_PRIORITY 명령을 발행하지 않는 경우 유틸리티는 조절 취소 상태에서 실행됩니다(**util\_impact\_lim** 설정에 관계없이).

실행 중인 유틸리티에 대한 현재 우선순위 설정을 보려면 LIST UTILITIES 명령을 사용하면 됩니다.

## 사용 시 참고사항

조절을 수행하려면 **util\_impact\_lim** 구성 매개변수를 설정하여 영향 규정을 정의해야 합니다.



---

## 제 122 장 SET WORKLOAD 명령

데이터베이스 연결이 지정되는 워크로드를 지정합니다. 이 명령은 데이터베이스에 연결하기 전에 발행하거나, 연결이 설정된 후 현재 연결을 다시 지정하기 위해 사용할 수 있습니다. 연결이 설정된 경우, 워크로드 재지정은 다음 작업 단위(UOW) 시작 시 수행됩니다.

### 권한 부여

없음, 그러나 사용시 참고사항 참조

### 필수 연결

없음

### 명령 구문

```
▶▶—SET WORKLOAD TO AUTOMATIC  
SYSDEFAULTADMWORKLOAD▶▶
```

### 명령 매개변수

#### AUTOMATIC

서버가 자동으로 수행하는 워크로드 평가에서 선택된 워크로드에 데이터베이스 연결이 지정됨을 지정합니다.

#### SYSDEFAULTADMWORKLOAD

*accessctrl*, *dataaccess*, *wlmadm*, *secadm* 또는 *dbadm* 권한이 있는 사용자는 일반 워크로드 평가를 생략할 수 있도록 허용하여, 데이터베이스 연결이 SYSDEFAULTADMWORKLOAD에 지정됨을 지정합니다.

#### 예:

연결을 SYSDEFAULTADMWORKLOAD에 지정하려는 경우:

```
SET WORKLOAD TO SYSDEFAULTADMWORKLOAD
```

서버가 수행하는 워크로드 평가에서 선택되는 워크로드를 사용하도록 워크로드 지정을 재설정하려는 경우:

```
SET WORKLOAD TO AUTOMATIC
```

## 사용 시 참고사항

데이터베이스 연결의 세션 권한 부여 ID가 *accessctrl*, *dataaccess*, *wlmadm*, *secadm* 또는 *dbadm* 권한을 가지고 있지 않은 경우, 연결은 SYSDEFAULTADMWORKLOAD에 지정될 수 없으며 SQL0552N 오류가 리턴됩니다. SET WORKLOAD TO SYSDEFAULTADMWORKLOAD 명령이 데이터베이스에 연결하기 전에 발행된 경우, 데이터베이스 연결이 설정된 후에 첫 번째 작업 단위(UOW) 시작 시 SQL0552N 오류가 리턴됩니다. 데이터베이스 연결이 설정된 경우에 명령이 발행되면, 워크로드 재지정이 발생하기로 되어 있는 경우 다음 작업 단위(UOW) 시작 시 SQL0552N 오류가 리턴됩니다.

---

## 제 123 장 SET WRITE

SET WRITE 명령을 사용하면 사용자가 입출력 쓰기를 일시중단하거나 데이터베이스에 대한 입출력 쓰기를 다시 시작할 수 있습니다. 이 명령의 일반적인 용도는 미러된 데이터베이스를 분할하는 것입니다. 이러한 유형의 미러링은 디스크 스토리지 시스템을 통해 수행됩니다.

이 새 상태, SUSPEND\_WRITE는 스냅샷 모니터에서 볼 수 있습니다. 명령이 성공적으로 실행되려면 모든 테이블 스페이스가 NORMAL 상태여야 합니다. 상태가 NORMAL이 아닌 테이블 스페이스가 있으면 명령이 실패합니다.

### 범위

이 명령은 명령이 실행되는 데이터베이스 파티션에만 영향을 미칩니다.

### 권한 부여

이 명령은 명령이 실행되는 노드에만 영향을 줍니다. 이 명령의 권한을 부여하려면 실행자에게 다음 특권 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*

### 필요한 연결

데이터베이스

### 명령 구문

▶▶ SET WRITE {SUSPEND | RESUME} FOR {DATABASE | DB} ▶▶

### 명령 매개변수

#### SUSPEND

입출력 쓰기를 일시중단하면 모든 테이블 스페이스가 새 SUSPEND\_WRITE 상태가 됩니다. 로그에 쓰기 또한 이 명령으로 일시중단됩니다. 모든 데이터베이스 조작용 온라인 백업 및 리스토어와 별도로 데이터베이스 쓰기가 일시중단되더라도 정상적으로 수행되어야 합니다. 그러나 버퍼 풀 또는 로그 버퍼에서 로그로 더티 페이지를 비우려고 시도하는 동안 일부 조작용 대기할 수 있습니다. 이러한 조작용 데이터베이스 쓰기가 다시 시작되면 정상적으로 다시 시작됩니다.

## RESUME

입출력 쓰기를 다시 시작하면 모든 테이블 스페이스에서 SUSPEND\_WRITE 상태가 제거되며 테이블 스페이스를 갱신할 수 있습니다.

## 사용 시 참고사항

입출력 쓰기는 일시중단된 동일한 연결에서 다시 시작되어야 합니다. 이 연결을 사용하여 입출력 쓰기를 다시 시작할 수 있도록 하려는 경우 데이터베이스 쓰기가 다시 시작될 때까지 이 연결에서 조작을 수행하지 않아도 됩니다. 그러나 버퍼 풀 또는 로그 버퍼에서 로그에 더티 페이지를 비워야 하는 경우 입출력 쓰기가 다시 시작될 때까지 일부 조작이 대기할 수 있습니다. 또한 버퍼 풀에서 디스크로 더티 페이지를 비워야 하는 경우 연속 연결 시도가 정지될 수 있습니다. 연속 연결은 데이터베이스 입출력이 다시 시작되면 완료됩니다. 연결 시도가 정지되고 입출력을 일시중단하기 위해 사용한 연결에서 입출력을 다시 시작할 수 없는 경우 RESTART DATABASE 명령과 WRITE RESUME 옵션을 함께 실행해야 합니다. 이러한 경우 RESTART DATABASE 명령을 사용하면 응급 복구를 수행하지 않고 입출력 쓰기를 다시 시작합니다. RESTART DATABASE 명령과 WRITE RESUME 옵션을 함께 사용하면 데이터베이스 손상 후 사용하는 경우 응급 복구만 수행합니다.

---

## 제 124 장 START DATABASE MANAGER

단일 데이터베이스 파티션이나 다중 파티션된 데이터베이스 환경에 정의된 모든 데이터베이스 파티션에서 현재 데이터베이스 관리 프로그램 인스턴스 백그라운드 프로세스를 시작합니다.

### 범위

다중 파티션된 데이터베이스 환경에서, 이 명령은 DBPARTITIONNUM 매개변수를 사용하지 않는 한 db2nodes.cfg 파일에 나열된 모든 데이터베이스 파티션에 영향을 미칩니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*

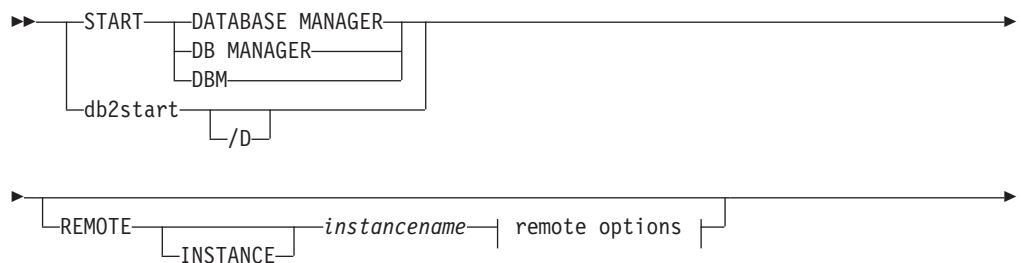
ADD DBPARTITIONNUM 시작 옵션에는 *sysadm* 또는 *sysctrl* 권한이 필요합니다.

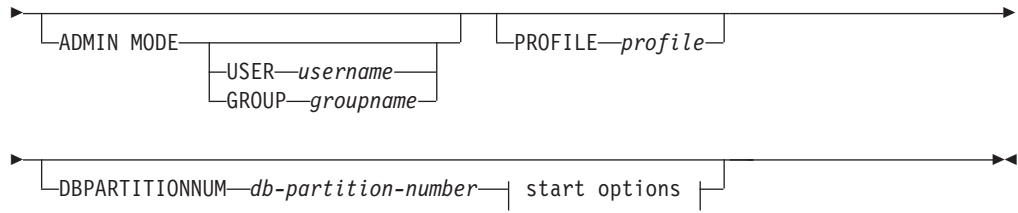
서비스를 시작하려면 Windows 운영 체제 요구사항을 충족해야 합니다. 확장 보안을 사용 안하면, 관리자, 서버 운영자 또는 권한이 있는 사용자 그룹의 구성원이어야 합니다. 확장 보안이 사용 가능하면, 데이터베이스를 시작하기 위해 관리자 그룹이나 DB2ADMNS 그룹 중 하나의 구성원이어야 합니다.

### 필수 연결

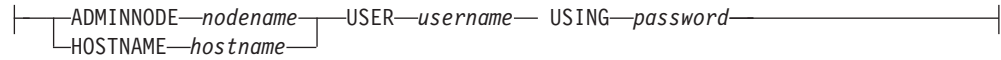
없음

### 명령 구문

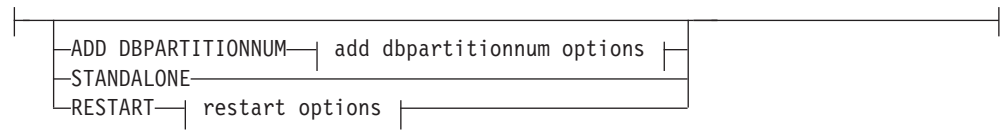




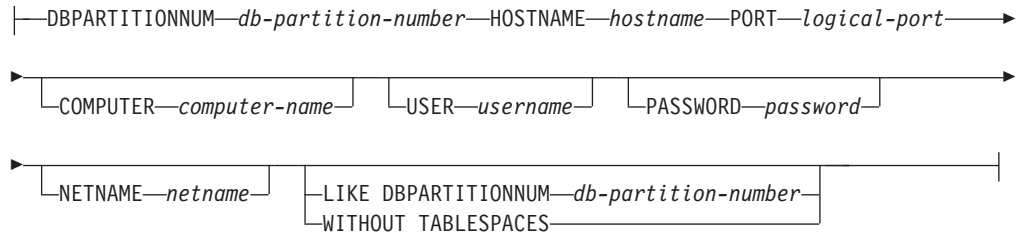
**remote options:**



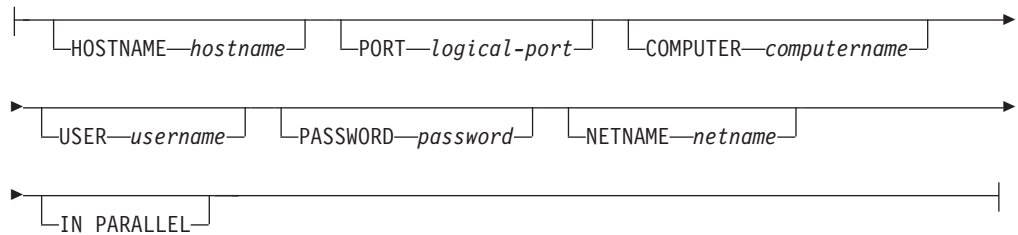
**start options:**



**add dbpartitionnum options:**



**restart options:**



**명령 매개변수**

**REMOTE [INSTANCE] instancename**

시작할 리모트 인스턴스의 이름을 지정합니다.

**ADMINNODE nodename**

REMOTE 또는 REMOTE INSTANCE와 함께, 관리 노드의 이름을 지정합니다.



**HOSTNAME** *hostname*

REMOTE 또는 REMOTE INSTANCE와 함께, 호스트 노드의 이름을 지정합니다.

**USER** *username*

REMOTE 또는 REMOTE INSTANCE와 함께, 사용자의 이름을 지정합니다.

**USING** *password*

REMOTE 또는 REMOTE INSTANCE 및 USER와 함께, 사용자의 암호를 지정합니다.

**ADMIN MODE**

관리 목적으로 Quiesce 모드에서 인스턴스를 시작합니다. 인스턴스가 아직 『가동』되지 않아서 연결 OFF를 강제로 실행하지 않아도 되는 경우를 제외하고 QUIESCE INSTANCE 명령과 같습니다.

**USER** *username*

ADMIN MODE와 함께, 사용자의 이름을 지정합니다.

**GROUP** *groupname*

ADMIN MODE와 함께, 그룹의 이름을 지정합니다.

다음 모든 매개변수는 ESE(Enterprise Server Edition) 환경에서만 유효합니다.

**PROFILE** *profile*

DB2 환경을 정의하기 위해 각 데이터베이스 파티션에서 실행될 프로파일 파일의 이름을 지정합니다. 이 파일은 데이터베이스 파티션이 시작되기 전에 실행됩니다. 프로파일 파일은 인스턴스 소유자의 sqllib 디렉토리에 상주해야 합니다. 프로파일 파일의 환경 변수 모두 사용자 세션에 정의할 필요는 없습니다.

**DBPARTITIONNUM** *db-partition-number*

시작할 데이터베이스 파티션을 지정합니다. 다른 옵션을 지정하지 않은 경우, 일반 시작이 이 데이터베이스 파티션에서 수행됩니다.

유효한 값은 0 - 999(0과 999 포함)입니다. ADD DBPARTITIONNUM을 지정하지 않은 경우, 값은 이미 인스턴스 소유자의 db2nodes.cfg 파일에 있어야 합니다. 데이터베이스 파티션 번호가 지정되지 않은 경우 구성 파일에 정의된 모든 데이터베이스 파티션이 시작됩니다.

**ADD DBPARTITIONNUM**

새 데이터베이스 파티션 서버가 *hostname* 및 *logical-port* 값을 가지고 있는 인스턴스 소유자의 db2nodes.cfg 파일에 추가됨을 지정합니다.

*hostname* 및 *logical-port*의 조합이 고유한지 확인하십시오.

추가되는 데이터베이스 서버 파티션에 대해 기존의 모든 데이터베이스를 작성하기 위해 데이터베이스 파티션 서버 추가 유틸리티가 내부적으로 실행됩니다. 새 데이터베이스 파티션 서버는 자동으로 db2nodes.cfg 파일에 추가됩니다.

주: 카탈로그 해제된 데이터베이스는 새 데이터베이스 파티션을 추가할 때 인식되지 않습니다. 카탈로그 해제된 데이터베이스는 새 데이터베이스 파티션에 표시되지 않습니다. 새 데이터베이스 파티션에서 데이터베이스에 연결하려고 하면 오류 메시지 SQL1013N이 리턴됩니다.

두 개 이상의 활성 데이터베이스 파티션 서버를 가지고 있는 환경에서 ADD 요청이 작성된 경우, ADD 처리 완료 시 새 데이터베이스 파티션 서버가 환경에 표시됩니다.

하나의 데이터베이스 파티션 서버가 있고 그 서버가 활성 상태인 환경에서 ADD 요청이 작성된 경우, ADD 처리 완료 후 새 데이터베이스 파티션 서버는 비활성 상태가 됩니다. 인스턴스는 새 데이터베이스 파티션 서버가 파티션된 데이터베이스 환경에 참여하기 전에 db2stop 및 db2start를 사용하여 다시 시작해야 합니다. 하나의 데이터베이스 파티션 서버가 있고 그 서버가 비활성 상태인 환경에서 ADD 요청이 작성된 경우, ADD 처리 완료 후 새 데이터베이스 파티션 서버(또는 두 개 이상이 추가된 경우에는 여러 서버)는 활성 상태가 됩니다. 원래의 데이터베이스 파티션 서버만 시작해야 합니다.

새로 추가되는 데이터베이스 파티션은 다음과 같이 ADD 처리 중에 구성됩니다.

1. 다중 파티션 환경에서, 새 데이터베이스 파티션은 비카탈로그 데이터베이스 파티션의 데이터베이스 구성 매개변수 값을 사용하여 구성됩니다.
2. 단일 파티션 환경에서, 새 데이터베이스 파티션은 카탈로그 파티션의 데이터베이스 구성 매개변수 값을 사용하여 구성됩니다.
3. 데이터베이스 구성 매개변수 값을 새 데이터베이스 파티션으로 복사할 때 문제가 발생하는 경우, 새 데이터베이스 파티션은 디폴트 데이터베이스 구성 매개변수 값을 사용하여 구성됩니다.

**DBPARTITIONNUM** *db-partition-number*

작성될 새 데이터베이스 파티션에 대한 데이터베이스 파티션 번호 값을 지정합니다.

**HOSTNAME** *hostname*

ADD DBPARTITIONNUM과 함께, db2nodes.cfg 파일에 추가될 호스트 이름을 지정합니다.

**PORT** *logical-port*

ADD DBPARTITIONNUM과 함께, db2nodes.cfg 파일에 추가될 논리 포트를 지정합니다. 유효한 값은 0 - 999입니다.

**COMPUTER *computername***

새 데이터베이스 파티션이 작성되는 머신에 대한 컴퓨터 이름. 이 매개 변수는 Windows에서 필수이지만 다른 운영 체제에서는 무시됩니다.

**USER *username***

새 데이터베이스 파티션의 어카운트에 대한 사용자 이름. 이 매개 변수는 Windows에서 필수이지만 다른 운영 체제에서는 무시됩니다.

**PASSWORD *password***

새 데이터베이스 파티션의 어카운트에 대한 암호. 이 매개 변수는 Windows에서 필수이지만 다른 운영 체제에서는 무시됩니다.

**NETNAME *netname***

db2nodes.cfg 파일에 추가될 *netname*을 지정합니다. 지정하지 않은 경우 이 매개 변수의 디폴트는 *hostname*에 대해 지정된 값입니다.

**IN PARALLEL**

병렬 실행을 위해 RESTART DATABASE 명령을 실행합니다.

**LIKE DBPARTITIONNUM *db-partition-number***

시스템 임시 테이블 스페이스에 대한 컨테이너가 인스턴스의 각 데이터베이스에 대해 지정된 *db-partition-number*의 컨테이너와 같도록 지정합니다. 지정된 데이터베이스 파티션은 이미 db2nodes.cfg 파일에 있는 데이터베이스 파티션이어야 합니다. 자동 스토리지를 사용하도록 정의된 시스템 임시 테이블 스페이스(즉, CREATE TABLESPACE 문의 MANAGED BY AUTOMATIC STORAGE절로 작성되었거나 MANAGED BY절이 전혀 지정되지 않은 시스템 임시 테이블 스페이스)의 경우, 컨테이너는 지정된 파티션의 컨테이너와 반드시 일치할 필요는 없습니다. 대신, 컨테이너는 데이터베이스와 연관되는 스토리지 경로를 기반으로 하는 데이터베이스 관리 프로그램에 의해 자동으로 지정됩니다. 이로서 두 파티션에서 동일한 컨테이너가 사용될 수도, 그렇지 않을 수도 있습니다.

**WITHOUT TABLESPACES**

시스템 임시 테이블 스페이스에 대한 컨테이너가 어떤 데이터베이스에 대해서도 작성되지 않음을 지정합니다. ALTER TABLESPACE 문을 사용하여 데이터베이스를 사용하기 전에 각 데이터베이스에 시스템 임시 테이블 스페이스 컨테이너를 추가해야 합니다. 자동 스토리지를 사용하도록 정의된 시스템 임시 테이블 스페이스(즉, CREATE TABLESPACE 문의 MANAGED BY AUTOMATIC STORAGE절로 작성되었거나 MANAGED BY절이 전혀 지정되지 않은 시스템 임시 테이블 스페이스)에 대해서는 이 옵션이 무시됩니다. 이 테이블 스페이스의 경우 컨테이너 작성을 지연할 방법이 없습니다. 컨테이너는 데

데이터베이스에 연결된 스토리지 경로를 기반으로 하는 데이터베이스 관리 프로그램에 의해 자동으로 할당됩니다.

## STANDALONE

데이터베이스 파티션이 독립형 모드에서 시작됨을 지정합니다. FCM은 다른 데이터베이스 파티션과의 연결 설정을 시도하지 않습니다. 이 옵션은 데이터베이스 파티션을 추가할 때 사용됩니다.

## RESTART

실패 후에 데이터베이스 관리 프로그램을 시작합니다. 다른 데이터베이스 파티션은 계속 작동하며 이 데이터베이스 파티션은 다른 데이터베이스 파티션에 연결하려고 합니다. *hostname*과 *logical-port* 매개변수를 지정하지 않은 경우, 데이터베이스 관리 프로그램은 *db2nodes.cfg*에 지정된 *hostname* 및 *logical-port* 값을 사용하여 다시 시작됩니다. 두 매개변수 중 하나를 지정한 경우에는 연결 설정 시 새 값을 다른 데이터베이스 파티션에 보냅니다. *db2nodes.cfg* 파일은 이 정보로 갱신됩니다.

## HOSTNAME *hostname*

RESTART 매개변수와 함께 HOSTNAME 옵션을 사용하여 데이터베이스 파티션 구성 파일 *db2nodes.cfg*에 지정된 것과 다른 머신에서 데이터베이스 파티션을 재시작할 수 있습니다.

### 제한사항:

DB2 고가용성 기능을 사용할 때, RESTART 매개변수와 함께 HOSTNAME 옵션을 사용하여 다른 머신에서 데이터베이스 파티션을 재시작하면 안됩니다. 클러스터에 있는 하나의 머신에서 다른 머신으로 이동하거나 재시작하려면 DB2 고가용성 인스턴스 구성 유틸리티(*db2haicu*)를 사용하십시오.

## PORT *logical-port*

RESTART와 함께, 데이터베이스 파티션 구성 파일에 있는 논리 포트 번호를 대체하기 위해 사용할 논리 포트 번호를 지정합니다. 지정하지 않은 경우 이 매개변수의 디폴트는 *db2nodes.cfg* 파일에서 *num* 값에 해당되는 *logical-port* 값입니다. 유효한 값은 0 - 999입니다.

## COMPUTER *computername*

새 데이터베이스 파티션이 작성되는 머신에 대한 컴퓨터 이름. 이 매개변수는 Windows에서 필수이지만 다른 운영 체제에서는 무시됩니다.

## USER *username*

새 데이터베이스 파티션의 어카운트에 대한 사용자 이름. 이 매개변수는 Windows에서 필수이지만 다른 운영 체제에서는 무시됩니다.

### **PASSWORD** *password*

새 데이터베이스 파티션의 어카운트에 대한 암호. 이 매개변수는 Windows에서 필수이지만 다른 운영 체제에서는 무시됩니다.

### **NETNAME** *netname*

db2nodes.cfg 파일에 지정된 값을 대체할 *netname*을 지정합니다. 지정하지 않은 경우 이 매개변수의 디폴트는 db2nodes.cfg 파일에서 *db-partition-number* 값에 해당되는 *netname* 값입니다.

## 예:

다음은 데이터베이스 파티션 10, 20 및 30이 있는 3 데이터베이스 파티션 시스템에서 발행된 db2start의 샘플 출력입니다.

```
04-07-1997 10:33:05 10 0 SQL1063N DB2START 처리가 완료되었습니다.  
04-07-1997 10:33:07 20 0 SQL1063N DB2START 처리가 완료되었습니다.  
04-07-1997 10:33:07 30 0 SQL1063N DB2START 처리가 완료되었습니다.  
SQL1063N DB2START 처리가 완료되었습니다.
```

## 사용 시 참고사항

Microsoft Windows Vista 이상의 버전에서는 전체 관리자 특권을 갖고 실행 중인 DB2 명령 창에서 이 명령을 실행해야 합니다.

클라이언트 노드에서 이 명령을 발행할 필요는 없습니다. 이전 클라이언트와의 호환성을 위해 제공되지만 데이터베이스 관리 프로그램에는 효과가 없습니다.

데이터베이스 관리 프로그램 인스턴스는 일단 시작한 후에는 사용 중이었던 모든 응용 프로그램이 종료한 경우에도 사용자가 중지할 때까지 실행됩니다.

데이터베이스 관리 프로그램이 시작되는 경우 성공적인 완료 메시지가 표준 출력 디바이스로 전송됩니다. 오류가 발생하면 처리가 중지되고 오류 메시지가 표준 출력 디바이스로 전송됩니다. 파티션된 데이터베이스 환경에서 메시지는 START DATABASE MANAGER 명령을 발행한 데이터베이스 파티션에서 리턴됩니다.

파티션된 데이터베이스 환경에서 매개변수를 지정하지 않은 경우 데이터베이스 관리 프로그램은 데이터베이스 파티션 구성 파일에 지정된 매개변수를 사용하여 모든 병렬 노드에서 시작됩니다.

START DATABASE MANAGER 명령이 진행 중이면, 데이터베이스에 요청을 발행하기 전에 적용 가능한 데이터베이스 파티션이 시작되었는지 확인하십시오.

db2cshrc 파일은 지원되지 않으므로 환경 정의에 사용할 수 없습니다.

Quiesce 상태에서 인스턴스를 시작할 수 있습니다. 다음 선택사항 중 하나를 사용하여 수행할 수 있습니다.

```
db2start admin mode
```

또는

```
db2start admin mode user username
```

또는

```
db2start admin mode group groupname
```

새 데이터베이스 파티션 서버를 추가할 때, START DATABASE MANAGER는 인스턴스의 각 데이터베이스가 자동 스토리지에 대해 사용 가능한지 여부를 판별해야 합니다. 이는 각 데이터베이스의 카탈로그 파티션과 통신하여 판별합니다. 자동 스토리지가 사용 가능한 경우 스토리지 경로 정의는 통신 중에 검색됩니다. 이와 유사하게 시스템 임시 테이블 스페이스가 데이터베이스 파티션과 함께 작성되는 경우 START DATABASE MANAGER는 해당 서버에 있는 데이터베이스 파티션의 테이블 스페이스 정의를 검색하기 위해 다른 데이터베이스 파티션 서버와 통신해야 합니다. **start\_stop\_time** 데이터베이스 관리 프로그램 구성 매개변수를 사용하여 다른 데이터베이스 파티션 서버가 자동 스토리지 및 테이블 스페이스 정의에 응답해야 하는 시간을 분 단위로 지정합니다. 이 시간이 초과되면 명령은 실패합니다. 이와 같은 상황이 발생하면 **start\_stop\_time** 값을 증가시키고 명령을 다시 실행하십시오.

다음 명령, 명령문 또는 조작 중 하나가 진행 중 상태에 있는 경우 새 데이터베이스 파티션 서버를 추가할 수 없습니다. 그렇지 않으면 SQL6074N이 리턴됩니다.

- QUIESCE INSTANCE
- UNQUIESCE INSTANCE
- STOP DB2 (db2stop)
- STOP DATABASE MANAGER DBPARTITIONNUM
- START DB2 (db2start)
- START DATABASE MANAGER DBPARTITIONNUM
- 재시작 옵션이 있는 START DATABASE MANAGER
- CREATE DATABASE
- DROP DATABASE
- QUIESCE DATABASE
- UNQUIESCE DATABASE
- ACTIVATE DATABASE
- DEACTIVATE DATABASE
- 데이터베이스 오브젝트에 대한 A Z 잠금
- 모든 데이터베이스 파티션 서버에서 데이터베이스 백업
- 데이터베이스 리스토어
- 테이블 스페이스의 ALTER, ALTER 또는 DROP

- 자동 스토리지 경로의 갱신

UNIX 플랫폼에서는 START DATABASE MANAGER 명령이 SIGINT 신호를 지원합니다. CTRL+C를 누르면 신호가 발행됩니다. 이 신호가 발생하면, 진행 중인 모든 시작이 인터럽트되며 인터럽트된 각 데이터베이스 파티션에서 \$HOME/sql1lib/log/db2start로 메시지(SQL1044N)가 리턴됩니다. *timestamp.log* 오류 로그 파일로 메시지 SQL6037N이 리턴됩니다. 이미 시작된 데이터베이스 파티션은 영향을 받지 않습니다. 시작 중인 데이터베이스 파티션에서 CTRL+C를 누른 경우, 다시 시작하기 전에 해당 데이터베이스 파티션에 대해 db2stop을 발행해야 합니다.

Windows 운영 체제에서, db2start 명령과 NET START 명령은 통신 서브시스템이 시작하지 못한 경우에 경고를 리턴하지 않습니다. Windows 환경의 데이터베이스 관리 프로그램은 서비스로 구현되고 서비스가 시작되면 오류를 리턴하지 않습니다. db2start 실행 중에 발생하였을 수 있는 오류에 대해 db2diag 로그나 이벤트 로그를 조사하도록 하십시오.

## 호환성

버전 8 이전 버전과의 호환성:

- LIKE DBPARTITIONNUM에 키워드 LIKE NODE를 대체할 수 있습니다.
- ADD DBPARTITIONNUM에 키워드 ADDNODE를 대체할 수 있습니다.
- DBPARTITIONNUM에 키워드 NODENUM을 대체할 수 있습니다.





---

## 제 125 장 START HADR

데이터베이스에 대한 HADR 조작을 시작합니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*

### 필수 연결

인스턴스. 이 명령은 데이터베이스 연결이 없는 경우 연결을 설정하고 명령이 완료되면 데이터베이스 연결을 닫습니다.

### 명령 구문

```
▶▶ START HADR ON DATABASE database-alias
                   DB
▶ USER user-name AS PRIMARY BY FORCE
                   USING password STANDBY
```

### 명령 매개변수

**DATABASE** *database-alias*

HADR 조작이 시작될 데이터베이스를 식별합니다.

**USER** *user-name*

HADR 조작이 시작될 사용자 이름을 식별합니다.

**USING** *password*

*user-name*을 인증하는 데 사용되는 암호

**AS PRIMARY**

HADR 기본 조작이 데이터베이스에서 시작되는 것으로 지정합니다.

**BY FORCE**

HADR 기본 데이터베이스가 대기 데이터베이스가 연결될 때까지 대기하지 않는 것으로 지정합니다. **BY FORCE**를 시작한 후 나중에 대기 데이터베이스를 사용할 수 있게 될 때마다 기본 데이터베이스가 대기 데이터베이스의 유효한 연결을 계속 승인합니다. **BY FORCE**가 사용되면 데이터베이스 구성 매개변수 **AUTORESTART**의 값에 관계 없

이 필요에 따라 응급 복구를 수행합니다. 기본 데이터베이스를 시작하는 다른 방법(예를 들어, 강제 실행되지 않는 START HADR 명령, ACTIVATE DATABASE 명령 또는 클라이언트 연결)은 **AUTORESTART** 설정을 준수합니다.

**주의:** START HADR 명령을 AS PRIMARY BY FORCE 옵션과 함께 사용하는 경우 주의를 기울여야 합니다. 대기 데이터베이스가 기본 데이터베이스로 변경되고 START HADR 명령과 AS PRIMARY BY FORCE 옵션을 함께 실행하여 원래 기본 데이터베이스가 재시작되는 경우 두 데이터베이스 사본이 모두 독립적으로 기본 데이터베이스로서 실행됩니다. 이를 때때로 분할뇌(split brain) 또는 이중 기본(dual primary)이라고 합니다. 이러한 경우 각 기본 데이터베이스는 연결을 승인하고 트랜잭션을 수행할 수 있으며 다른 사람이 작성한 갱신사항을 수신하거나 재생하지 않습니다. 결과적으로 두 데이터베이스 사본은 서로 불일치하게 됩니다.

### AS STANDBY

HADR 대기 조작이 데이터베이스에서 시작되는 것으로 지정합니다. 대기 데이터베이스는 연결이 설정될 때까지 또는 기본 데이터베이스가 연결 시도를 명시적으로 거부할 때까지 HADR 기본 데이터베이스에 대한 연결을 시도합니다. (HADR 구성 매개변수가 잘못 설정된 경우 기본 데이터베이스가 연결을 거부하거나, 데이터베이스 사본이 일치하지 않는 경우 연결을 계속 재시도하는 두 조건이 모두 적절하지 않습니다.)

### 사용 시 참고사항

다음 표는 다양한 조건의 데이터베이스 동작을 나타냅니다.

데이터베이스 상태	AS PRIMARY 옵션을 사용하는 START HADR 명령 동작	AS STANDBY 옵션을 사용하는 START HADR 명령 동작
비활성 표준 데이터베이스	HADR 기본 데이터베이스로 활성화됩니다.	데이터베이스가 롤 포워드 보류 모드(리스토어 또는 분할 미러의 결과일 수 있음) 또는 롤 포워드 진행 중 모드인 경우 대기 데이터베이스로서 시작됩니다. 그렇지 않으면 오류가 발생합니다.
활성 표준 데이터베이스	데이터베이스가 HADR 기본 역할을 시작합니다.	오류 메시지가 리턴됨.
비활성 기본 데이터베이스	HADR 기본 데이터베이스로 활성화됩니다.	장애 복구 후 실패한 기본 데이터베이스를 HADR 쌍으로 새 대기 데이터베이스로서 다시 통합합니다. 몇 가지 제한 사항이 적용됩니다.
활성 기본 데이터베이스	경고 메시지가 발행됨.	오류 메시지가 리턴됨.
비활성 대기 데이터베이스	오류 메시지가 리턴됨.	데이터베이스를 대기 데이터베이스로서 시작합니다.

데이터베이스 상태	AS PRIMARY 옵션을 사용하는 START HADR 명령 동작	AS STANDBY 옵션을 사용하는 START HADR 명령 동작
활성 대기 데이터베이스	오류 메시지가 리턴됨.	경고 메시지가 발행됨.

START HADR 명령을 실행하면 해당 오류 코드(SQL1767N, SQL1769N 또는 SQL1770N)가 이유 코드 98과 함께 생성됩니다. 이유 코드는 명령이 실행된 서버에서 HADR에 대해 설치된 라이선스가 없음을 나타냅니다. 문제점을 정정하려면 db2licm을 사용하여 유효한 HADR 라이선스를 설치하거나 유효한 HADR 라이선스가 분산의 일부로 포함되는 서버의 버전을 설치하십시오.



---

## 제 126 장 STOP DATABASE MANAGER

현재 데이터베이스 관리 프로그램 인스턴스를 중지합니다. 명시적으로 중지되지 않으면 데이터베이스 관리 프로그램은 계속 활성 상태입니다. 응용프로그램이 데이터베이스에 연결되는 경우 이 명령은 데이터베이스 관리 프로그램 인스턴스를 중지하지 않습니다. 데이터베이스 연결이 없지만 인스턴스 첨부이 있는 경우 인스턴스 첨부을 강제하고 데이터베이스 관리 프로그램을 중지합니다. 이 명령은 또한 데이터베이스 관리 프로그램을 중지하기 전에 모든 미완료 데이터베이스 활성을 비활성화합니다.

파티션된 데이터베이스 환경에서 이 명령은 데이터베이스 파티션 또는 모든 데이터베이스 파티션의 현재 데이터베이스 관리 프로그램 인스턴스를 중지합니다. 모든 데이터베이스 파티션의 데이터베이스 관리 프로그램을 중지할 때 `db2nodes.cfg` 구성 파일을 사용하여 각 데이터베이스 파티션에 관한 정보를 얻습니다.

이 명령을 사용하여 `db2nodes.cfg` 파일에서 데이터베이스 파티션을 삭제(drop)할 수도 있습니다(파티션된 데이터베이스 환경만 해당).

이 명령은 클라이언트에서 유효하지 않습니다.

### 범위

디폴트로 및 파티션된 데이터베이스 환경에서 이 명령은 `db2nodes.cfg` 파일에 나열되는 모든 데이터베이스 파티션에 영향을 줍니다.

### 권한 부여

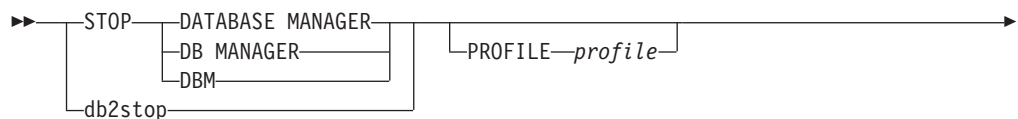
다음 중 하나가 필요합니다.

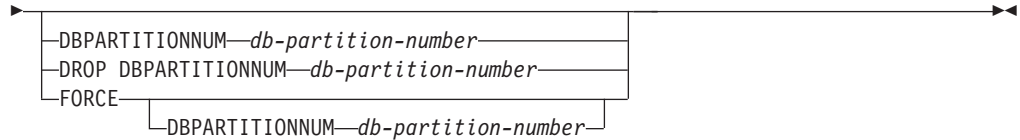
- `sysadm`
- `sysctrl`
- `sysmaint`

### 필수 연결

없음

### 명령 구문





## 명령 매개변수

### **PROFILE** *profile*

파티션된 데이터베이스 환경만 해당합니다. 시작된 데이터베이스 파티션에 대한 DB2 환경을 정의하기 위해 시작 시에 실행된 프로파일 파일의 이름을 지정합니다. START DATABASE MANAGER 명령에 대한 프로파일이 지정된 경우 동일한 프로파일을 여기에 지정해야 합니다. 프로파일 파일은 인스턴스 소유자의 sqllib 디렉토리에 상주해야 합니다.

### **DBPARTITIONNUM** *db-partition-number*

파티션된 데이터베이스 환경만 해당합니다. 중지될 데이터베이스 파티션을 지정합니다.

올바른 값은 0부터 999까지이며, db2nodes.cfg 파일에 있어야 합니다. 데이터베이스 파티션 번호가 지정되지 않은 경우 구성 파일에서 정의되는 모든 데이터베이스 파티션이 중지됩니다.

### **DROP DBPARTITIONNUM** *db-partition-number*

파티션된 데이터베이스 환경만 해당합니다. db2nodes.cfg 파일에서 삭제될 데이터베이스 파티션을 지정합니다.

이 매개변수를 사용하기 전에 DROP DBPARTITIONNUM VERIFY 명령을 실행하여 이 데이터베이스 파티션에 사용자 데이터가 없는지 확인하십시오.

이 옵션을 지정할 때 db2nodes.cfg 파일의 모든 데이터베이스 파티션이 중지됩니다.

### **FORCE**

각 데이터베이스 파티션에서 데이터베이스 관리 프로그램을 중지할 때 FORCE APPLICATION ALL을 사용하도록 지정합니다.

### **DBPARTITIONNUM** *db-partition-number*

파티션된 데이터베이스 환경만 해당합니다. 해당 데이터베이스 파티션의 모든 응용프로그램이 강제로 중지된 후 중지될 데이터베이스 파티션을 지정합니다. FORCE 옵션이 이 매개변수 없이 사용되는 경우 모든 데이터베이스 파티션의 모든 응용프로그램은 모든 데이터베이스 파티션이 중지되기 전에 강제 실행됩니다.

예:

다음은 데이터베이스 파티션 10, 20 및 30을 갖는 3 파티션 시스템에서 발행된 db2stop의 샘플 출력입니다.

```
04-07-1997 10:32:53 10 0 SQL1064N DB2STOP 처리가 완료되었습니다.
04-07-1997 10:32:54 20 0 SQL1064N DB2STOP processing was successful.
04-07-1997 10:32:55 30 0 SQL1064N DB2STOP processing was successful.
SQL1064N DB2STOP processing was successful.
```

## 사용 시 참고사항

Microsoft Windows Vista 이상의 버전에서는 전체 관리자 특권을 갖고 실행 중인 DB2 명령 창에서 이 명령을 실행해야 합니다.

클라이언트 노드에서 이 명령을 발행할 필요는 없습니다. 이전 클라이언트와의 호환성을 위해 제공되지만 데이터베이스 관리 프로그램에는 효과가 없습니다.

데이터베이스 관리 프로그램 인스턴스는 일단 시작한 후에는 사용 중이었던 모든 응용 프로그램이 종료한 경우에도 사용자가 중지할 때까지 실행됩니다.

데이터베이스 관리 프로그램이 중지되는 경우 완료 메시지가 표준 출력 디바이스로 전송됩니다. 오류가 발생하면 처리가 중지되고 오류 메시지가 표준 출력 디바이스로 전송됩니다.

응용프로그램이 아직 데이터베이스에 연결되어 있기 때문에 데이터베이스 관리 프로그램을 중지할 수 없는 경우, FORCE APPLICATION 명령을 사용하여 먼저 모든 사용자의 연결을 끊거나 FORCE 옵션과 함께 STOP DATABASE MANAGER 명령을 다시 실행하십시오.

다음 정보는 파티션된 데이터베이스 환경에만 적용됩니다.

- 매개변수를 지정하지 않으면 구성 파일에서 나열되는 각 데이터베이스 파티션에서 데이터베이스 관리 프로그램이 중지됩니다. 관리 통지 로그에 기타 데이터베이스 파티션이 종료되었음을 표시하는 메시지가 들어있을 수 있습니다.
- 이전 STOP DATABASE MANAGER 명령이 발행된 이후에 파티션된 데이터베이스 환경에 추가된 모든 데이터베이스 파티션은 db2nodes.cfg 파일에서 갱신됩니다.
- UNIX 플랫폼에서, **start\_stop\_time** 데이터베이스 관리 프로그램 구성 매개변수에 대해 지정되는 값에 도달하는 경우, 모든 진행 중 중지가 인터럽트되며 각 인터럽트된 데이터베이스 파티션에서 \$HOME/sqllib/log/db2stop.timestamp.log 오류 로그 파일로 메시지 SQL6037N이 리턴됩니다. 이미 중지된 데이터베이스 파티션은 영향을 받지 않습니다.
- db2cshrc 파일은 지원되지 않으며 PROFILE 매개변수의 값으로 지정할 수 없습니다.

**경고:** UNIX kill 명령은 제어된 종료 및 정리 처리 없이 데이터베이스 관리 프로그램 프로세스를 갑자기 종료하기 때문에 데이터베이스 관리 프로그램을 종료하는 데 사용해서는 안됩니다.



---

## 제 127 장 STOP HADR

데이터베이스에 대한 HADR 조작을 중지합니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*

### 필수 연결

인스턴스. 이 명령은 데이터베이스 연결이 없는 경우 연결을 설정하고 명령이 완료되면 데이터베이스 연결을 닫습니다.

### 명령 구문

```
▶▶ STOP HADR ON DATABASE database-alias
                   |
                   | DB
                   |
▶
USER user-name
   |
   | USING password
   |
▶▶
```

### 명령 매개변수

**DATABASE** *database-alias*

HADR 조작이 중지될 데이터베이스를 식별합니다.

**USER** *user-name*

HADR 조작이 중지될 사용자 이름을 식별합니다.

**USING** *password*

*user-name*을 인증하는 데 사용되는 암호

### 사용 시 참고사항

다음 표는 다양한 조건의 데이터베이스 동작을 나타냅니다.

데이터베이스 상태	STOP HADR 명령 동작
비활성 표준 데이터베이스	오류 메시지가 리턴됨.
활성 표준 데이터베이스	오류 메시지가 리턴됨.

데이터베이스 상태	STOP HADR 명령 동작
비활성 기본 데이터베이스	데이터베이스 역할이 표준으로 변경됩니다. 데이터베이스 구성 매개 변수 <code>hadr_db_role</code> 이 STANDARD로 갱신됩니다. 데이터베이스는 오프라인 상태를 유지합니다. 다음에 재시작할 때 표준 역할을 입력합니다.
활성 기본 데이터베이스	HADR 대기 데이터베이스에 대한 제공 로그를 중지하고 HADR 기본 데이터베이스에서 모든 HADR EDU를 종료합니다. 데이터베이스 역할은 표준으로 변경되고 데이터베이스는 온라인 상태를 유지합니다. 데이터베이스는 명시적인 START HADR 명령과 AS PRIMARY 옵션이 함께 실행될 때까지 표준 역할을 유지합니다. 열린 세션과 트랜잭션은 STOP HADR 명령의 영향을 받지 않습니다. 데이터베이스가 온라인 상태인 경우 STOP HADR 및 START HADR 명령을 반복적으로 실행할 수 있습니다. 이러한 명령은 동적으로 적용됩니다.
비활성 대기 데이터베이스	데이터베이스 역할이 표준으로 변경됩니다. 데이터베이스 구성 매개 변수 <code>hadr_db_role</code> 이 STANDARD로 갱신됩니다. 데이터베이스는 오프라인 상태를 유지합니다. 데이터베이스는 롤 포워드 보류 모드가 됩니다.
활성 대기 데이터베이스	오류 메시지가 리턴됨: 대기 데이터베이스를 표준 데이터베이스로 변환하기 전에 비활성화합니다.

STOP HADR 명령을 실행하면 해당 오류 코드(SQL1767N, SQL1769N 또는 SQL1770N)가 이유 코드 98과 함께 생성됩니다. 이유 코드는 명령이 실행된 서버에서 HADR에 대해 설치된 라이선스가 없음을 나타냅니다. 문제점을 정정하려면 db2licm을 사용하여 유효한 HADR 라이선스를 설치하거나 유효한 HADR 라이선스가 분산의 일부로 포함되는 서버의 버전을 설치하십시오.

---

## 제 128 장 TAKEOVER HADR

HADR 대기 데이터베이스가 HADR 쌍에 대한 새 HADR 기본 데이터베이스로서 인계되도록 지시합니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*

### 필수 연결

인스턴스. 이 명령은 데이터베이스 연결이 없는 경우 연결을 설정하고 명령이 완료되면 데이터베이스 연결을 단습니다.

### 명령 구문

```
▶▶ TAKEOVER HADR ON DATABASE database-alias
                        DB
USER user-name
                        USING password
                        BY FORCE
                        PEER WINDOW ONLY
```

### 명령 매개변수

#### **DATABASE** *database-alias*

HADR 기본 데이터베이스로서 인계되어야 하는 현재 HADR 대기 데이터베이스를 식별합니다.

#### **USER** *user-name*

인계 조작이 시작될 사용자 이름을 식별합니다.

#### **USING** *password*

*user-name*을 인증하는 데 사용되는 암호

#### **BY FORCE**

데이터베이스가 원래 HADR 기본 데이터베이스가 종료되었음을 확인하도록 대기하지 않는 것으로 지정합니다. 이 옵션은 HADR 쌍이 피어 상태가 아닌 경우 필요합니다.

#### **PEER WINDOW ONLY**

이 옵션이 지정되면 명령이 성공하고 피어 창 기간이 끝나기 전에 기

본 데이터베이스가 삭제되는 경우 커밋된 트랜잭션이 손실되지 않습니다(데이터베이스 구성 매개변수 **HADR\_PEER\_WINDOW**를 0이 아닌 값으로 설정). 피어 창이 만기되기 전에 기본 데이터베이스가 삭제되지 않으면 분할뇌(*split brain*) 결과가 발생합니다. HADR 쌍이 피어 또는 연결 끊김 피어 상태가 아닐 때(피어 창이 만기되지 않음) **TAKEOVER BY FORCE PEER WINDOW ONLY** 명령이 실행되면 오류가 리턴됩니다.

주: 기본 데이터베이스 클럭과 대기 데이터베이스 클럭이 서로 5초 이내에 동기화되지 않으면 **PEER WINDOW ONLY** 옵션을 사용한 인계 조작이 잘못 작동할 수 있습니다. 즉, 조작이 실패해야 할 때 성공하거나 성공해야 할 때 실패할 수 있습니다. 클럭이 동일한 소스로 동기화되도록 하려면 시간 동기화 서비스(예를 들어, NTP)를 사용해야 합니다.

### 사용 시 참고사항

다음 표는 가능한 각 상태 및 옵션 조합에 대해 사용 중인 대기 데이터베이스에서 실행되는 **TAKEOVER HADR** 명령의 조작을 나타냅니다. 이 명령이 비활성 대기 데이터베이스에서 실행되면 오류 메시지가 리턴됩니다.

대기 상태	<b>BY FORCE</b> 옵션 사용	인계 동작
연결이 끊어진 피어	없음	NOT ALLOWED (SQL1770N)
연결이 끊어진 피어	예, <b>BY FORCE</b>	ALLOWED - NO ASSURANCE OF DATA CONSISTENCY 주: "no transaction loss" 인계는 <b>PEER WINDOW ONLY</b> 옵션 없이 <b>TAKEOVER BY FORCE</b> 명령을 사용해서도 가능합니다. 즉, 필요한 조건이 충족되는 경우 무조건 장애 복구가 가능합니다. 이러한 장애 복구는 기본 데이터베이스가 실패할 때 유효한 피어 창이 만기되고 한참 후에도 실행될 수 있습니다.

대기 상태	BY FORCE 옵션 사용	인계 동작
연결이 끊어진 피어	예, BY FORCE PEER WINDOW ONLY	<p>ALLOWED - GREATER DEGREE OF DATA CONSISTENCY</p> <p>다음과 같이 데이터 손실이 계속 발생할 수 있는 경우가 있습니다.</p> <ul style="list-style-type: none"> <li>• 피어 창이 만기된 후에도 기본 데이터베이스가 계속 활성 상태를 유지하는 경우 또한 기본 데이터베이스가 대기 데이터베이스에 계속 연결되지 않는 경우, 기본 데이터베이스가 연결이 끊어진 피어 상태를 종료하고 처리 트랜잭션을 독립적으로 다시 시작할 수 있습니다.</li> <li>• NEARSYNC 모드에서 기본 데이터베이스로부터의 트랜잭션 로그 수신을 확인한 후 그러나 해당 트랜잭션 로그 정보를 디스크에 쓰기 전에 대기 데이터베이스가 실패하는 경우, 로그 수신 버퍼의 해당 트랜잭션 로그 정보가 손실될 수 있습니다.</li> </ul>
로컬 캐치업 또는 리모트 캐치업	없음	오류 메시지가 리턴됨
로컬 캐치업 또는 리모트 캐치업	예	오류 메시지가 리턴됨
피어	없음	<p>기본 데이터베이스 및 대기 데이터베이스 전환 역할.</p> <p>인계 중에 실패가 발생하지 않은 경우 데이터가 손실되지 않습니다. 그러나 인계 중에 실패가 발생하면 데이터가 손실되었을 수 있으며 기본 및 대기 역할이 변경 또는 변경되지 않았을 수 있습니다. 다음은 기본 및 대기 역할이 전환되는 인계 중 실패 처리를 위한 지침입니다.</p> <ol style="list-style-type: none"> <li>1. 인계 조작 중에 실패가 발생하는 경우 HADR 데이터베이스의 역할이 변경 또는 변경되지 않았을 수 있습니다. 가능한 경우 두 데이터베이스가 모두 온라인 상태여야 합니다. 스냅샷 모니터를 사용하거나 데이터베이스 구성 매개변수 <code>hadr_db_role</code>의 값을 검사하여 사용 가능한 하나 이상의 데이터베이스의 HADR 역할을 검사합니다.</li> <li>2. 필요한 새 기본이 계속 대기 역할이고 인계가 계속 필요한 경우 TAKEOVER HADR 명령을 다시 실행합니다(BY FORCE 옵션에 대한 다음 지침 참조).</li> <li>3. 두 데이터베이스를 모두 대기 역할로 종료할 수 있습니다. 이러한 경우 기본 노드가 되어야 하는 노드에서 TAKEOVER HADR 명령과 BY FORCE 옵션을 함께 실행할 수 있습니다. 이러한 경우 두 대기가 일반 HADR 기본-대기 연결을 설정할 수 없으므로 BY FORCE 옵션이 필요합니다.</li> </ol>

대기 상태	BY FORCE 옵션 사용	인계 동작
피어	예	이전 기본은 유효하지 않은 것으로 표시되어 추가 로그를 작성하거나 추가 트랜잭션을 커밋할 수 없습니다.  다음에 로그 쓰기를 시도하면 데이터베이스가 종료됩니다. 그러나 이전 기본 세션이 읽기 전용 쿼리만 실행하는 경우에는 이전 기본이 무한정 실행될 수 있습니다. 기존 클라이언트 연결은 읽기 전용 조작을 수행하고 새 클라이언트 연결이 승인되는 한 계속 열려 있습니다.  이중-기본 상황을 피하려면 TAKEOVER HADR 명령과 BY FORCE 옵션을 실행하기 전에 먼저 이전 기본에 대한 트랜잭션 처리를 중지해야 합니다.
리모트 캐치업 보류	없음	오류 메시지가 리턴됨.
리모트 캐치업 보류	예	대기가 기본이 됩니다.

TAKEOVER HADR 명령을 실행하면 해당 오류 코드(SQL1767N, SQL1769N 또는 SQL1770N)가 이유 코드 98과 함께 생성됩니다. 이유 코드는 명령이 실행된 서버에서 HADR에 대해 설치된 라이선스가 없음을 나타냅니다. 문제점을 정정하려면 db2licm을 사용하여 유효한 HADR 라이선스를 설치하거나 유효한 HADR 라이선스가 분산의 일부로 포함되는 서버의 버전을 설치하십시오.

TAKEOVER BY FORCE PEER WINDOW ONLY 명령을 실행하고 명령이 성공하면(기본 데이터베이스가 대기 데이터베이스와의 연결이 끊어졌을 때 명령을 호출했지만 피어 창에 계속 있는 경우) 아직 대기 데이터베이스에 복사되지 않은 기본 데이터베이스에 대한 트랜잭션 정보가 있습니다.

---

## 제 129 장 TERMINATE

명령행 처리기의 백엔드 프로세스를 명시적으로 종료합니다.

### 권한 부여

없음

### 필수 연결

없음

### 명령 구문

▶—TERMINATE—▶

### 명령 매개변수

없음

### 사용 시 참고사항

응용프로그램이 데이터베이스에 연결되거나 프로세스가 작업 단위(UOW)의 중간에 있는 경우 TERMINATE는 데이터베이스 연결이 유실되게 합니다. 그런 다음 내부 커미트가 수행됩니다.

TERMINATE 및 CONNECT RESET가 둘 다 데이터베이스에 대한 연결을 중단하는 경우에도 TERMINATE는 백엔드 프로세스 종료를 가져옵니다.

db2stop 명령을 종료하기 전에 TERMINATE를 발행하는 것이 좋습니다. 그러면 백엔드 프로세스가 더 이상 사용할 수 없는 데이터베이스 관리 프로그램 인스턴스에 대한 접속을 유지하지 못하게 합니다.

DB2NODE 환경 변수가 세션에서 갱신될 때도 MPP 시스템의 백엔드 프로세스가 종료되어야 합니다. 이 환경 변수는 MPP 다중 논리 노드 구성에서 코디네이터 데이터베이스 파티션 번호를 지정하는 데 사용됩니다.





---

## 제 130 장 UNCATALOG DATABASE

데이터베이스 디렉토리에서 데이터베이스 항목을 삭제합니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*

### 필수 연결

없음. 디렉토리 조작은 로컬 디렉토리에만 영향을 줍니다.

### 명령 구문

```
▶▶ UNCATALOG DATABASE database-alias ◀◀
```

### 명령 매개변수

**DATABASE** *database-alias*

카탈로그 해제할 데이터베이스의 별명을 지정합니다.

### 사용 시 참고사항

로컬 데이터베이스 디렉토리의 항목만 카탈로그 해제할 수 있습니다. 시스템 데이터베이스 디렉토리의 항목은 DROP DATABASE 명령을 사용하여 삭제할 수 있습니다.

인스턴스의 데이터베이스를 재카탈로그하려면 UNCATALOG DATABASE 및 CATALOG DATABASE 명령을 사용하십시오. 노드에서 카탈로그된 데이터베이스를 나열하려면 LIST DATABASE DIRECTORY 명령을 사용하십시오.

이전 서버와 통신할 때 사용되는 데이터베이스의 인증 유형은 먼저 데이터베이스를 카탈로그 해제한 후 다른 유형으로 다시 카탈로그하여 변경할 수 있습니다.

디렉토리 캐싱이 사용 가능한 경우 데이터베이스, 노드 및 DCS 디렉토리 파일이 메모리에 캐시됩니다. GET DATABASE MANAGER CONFIGURATION 명령의 구성 매개변수 **dir\_cache**에 대한 정보를 참조하십시오. 응용프로그램의 디렉토리 캐시는 첫 번째 디렉토리 찾아보기 중에 작성됩니다. 캐시는 응용프로그램이 디렉토리 파일 중 하나를 수정할 때만 새로 고쳐지기 때문에, 다른 응용프로그램이 작성한 디렉토리 변경은 응용프로그램이 재시작될 때까지 적용되지 않을 수 있습니다.

CLP의 디렉토리 캐시를 새로 고치려면 `TERMINATE` 명령을 사용하십시오. 데이터베이스 관리 프로그램의 공유 캐시를 새로 고치려면 데이터베이스 관리 프로그램을 중지한 후(`db2stop`) 재시작하십시오(`db2start`). 다른 응용프로그램에 대한 디렉토리 캐시를 새로 고치려면 해당 응용프로그램을 중지한 후 재시작하십시오.

주: 시스템에 데이터베이스 파티션을 추가할 때 인스턴스의 모든 기존 데이터베이스가 새 데이터베이스 파티션으로 확장됩니다. 그러나 카탈로그 해제된 모든 데이터베이스는 새 데이터베이스 파티션을 추가할 때 인식되지 않습니다. 카탈로그 해제된 데이터베이스는 새 데이터베이스 파티션에 존재하지 않습니다. 새 데이터베이스 파티션의 데이터베이스에 연결 시도는 오류 메시지 `SQL1013N`을 리턴합니다.

---

## 제 131 장 UNCATALOG DCS DATABASE

데이터베이스 연결 서비스(DCS) 디렉토리에서 항목을 삭제합니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*

### 필수 연결

없음. 디렉토리 조작성 로컬 디렉토리에만 영향을 줍니다.

### 명령 구문

```
▶▶ UNCATALOG DCS DATABASE database-alias ▶▶
```

### 명령 매개변수

**DATABASE** *database-alias*

카탈로그 해제할 DCS 데이터베이스의 별명을 지정합니다.

### 사용 시 참고사항

DCS 데이터베이스는 리모트 데이터베이스로서 시스템 데이터베이스 디렉토리에 카탈로그되며 UNCATALOG DATABASE 명령을 사용하여 카탈로그 해제될 수 있습니다.

DCS 디렉토리에 데이터베이스를 재카탈로그하려면 UNCATALOG DCS DATABASE 및 CATALOG DCS DATABASE 명령을 사용하십시오. 노드에서 카탈로그되는 DCS 데이터베이스를 나열하려면 LIST DCS DIRECTORY 명령을 사용하십시오.

디렉토리 캐싱이 사용 가능한 경우 데이터베이스, 노드 및 DCS 디렉토리 파일이 메모리에 캐시됩니다. GET DATABASE MANAGER CONFIGURATION 명령의 출력에 있는 구성 매개변수 **dir\_cache**에 대해 제공되는 정보를 참조하십시오. 응용프로그램의 디렉토리 캐시는 첫 번째 디렉토리 찾아보기 중에 작성됩니다. 캐시는 응용프로그램이 디렉토리 파일 중 하나를 수정할 때만 새로 고쳐지므로, 다른 응용프로그램이 작성한 디렉토리 변경은 응용프로그램이 재시작될 때까지 적용되지 않을 수 있습니다.

CLP의 디렉토리 캐시를 새로 고치려면 TERMINATE 명령을 사용하십시오. DB2의 공유 캐시를 새로 고치려면 데이터베이스를 중지한 후(db2stop) 재시작하십시오(db2start).

다른 응용프로그램에 대한 디렉토리 캐시를 새로 고치려면 해당 응용프로그램을 중지한 후 재시작하십시오.

---

## 제 132 장 UNCATALOG LDAP DATABASE

LDAP(Lightweight Directory Access Protocol)에서 데이터베이스를 등록 해제하는 데 사용됩니다.

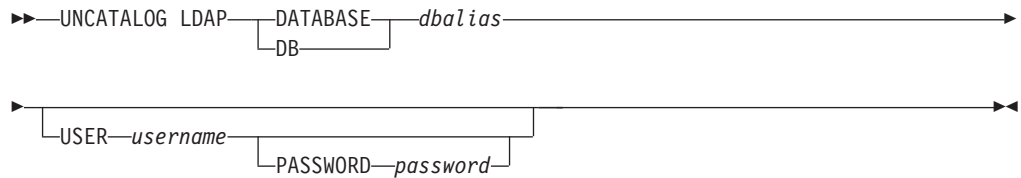
### 권한 부여

없음

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

#### **DATABASE** *dbalias*

카탈로그 해제할 LDAP 데이터베이스의 별명을 지정합니다.

#### **USER** *username*

사용자의 LDAP 식별 이름(DN)을 지정합니다. LDAP 사용자 DN은 LDAP 디렉토리에서 오브젝트를 삭제하기 위한 충분한 권한이 있어야 합니다. 사용자의 LDAP DN이 지정되지 않은 경우 현재 로그인 사용자의 증명서가 사용됩니다.

#### **PASSWORD** *password*

어카운트 암호

### 사용 시 참고사항

데이터베이스가 삭제될 때 데이터베이스 오브젝트가 LDAP에서 제거됩니다. 데이터베이스는 또한 데이터베이스를 관리하는 데이터베이스 서버가 LDAP에서 등록 해제될 때 자동으로 LDAP에서 등록 해제됩니다. 그러나 다음 경우에는 수동으로 LDAP에서 데이터베이스를 카탈로그 해제해야 할 수 있습니다.

- 데이터베이스 서버가 LDAP를 지원하지 않습니다. 관리자는 데이터베이스가 삭제된 후 LDAP에서 각 데이터베이스를 수동으로 카탈로그 해제해야 합니다.

- DROP DATABASE 중에 데이터베이스 오브젝트를 LDAP에서 제거할 수 없습니다(LDAP에 액세스할 수 없기 때문에). 이 경우 데이터베이스는 여전히 로컬 시스템에서 제거되지만 LDAP의 기존 항목은 삭제되지 않습니다.

---

## 제 133 장 UNCATALOG LDAP NODE

LDAP(Lightweight Directory Access Protocol)에서 노드 항목을 카탈로그 해제합니다.

**권한 부여**

없음

**필수 연결**

없음

**명령 구문**

```
▶▶ UNCATALOG LDAP NODE nodename [ USER username [ PASSWORD password ] ] ▶▶
```

**명령 매개변수**

**NODE** *nodename*

카탈로그 해제할 노드의 이름을 지정합니다.

**USER** *username*

사용자의 LDAP 식별 이름(DN)을 지정합니다. LDAP 사용자 DN은 LDAP 디렉토리에서 오브젝트를 삭제하기 위한 충분한 권한이 있어야 합니다. 사용자의 LDAP DN이 지정되지 않은 경우 현재 로그인 사용자의 증명서가 사용됩니다.

**PASSWORD** *password*

어카운트 암호

**사용 시 참고사항**

LDAP 노드는 DB2 서버가 LDAP에서 등록 해제될 때 자동으로 카탈로그 해제됩니다.





---

## 제 134 장 UNCATALOG NODE

노드 디렉토리에서 항목을 삭제합니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*

### 필수 연결

없음. 디렉토리 조작성 로컬 디렉토리에만 영향을 줍니다.

### 명령 구문

►►—UNCATALOG NODE—*nodename*—◄◄

### 명령 매개변수

**NODE** *nodename*

카탈로그 해제될 노드 항목을 지정합니다.

### 사용 시 참고사항

UNCATALOG NODE는 모든 유형의 노드에서 실행될 수 있지만, 리모트 인스턴스 또는 다른 로컬 인스턴스에 대한 첨부이 있는 경우에도 로컬 디렉토리만 영향을 받습니다.

디렉토리 캐싱이 사용 가능한 경우 데이터베이스, 노드 및 DCS 디렉토리 파일이 메모리에 캐시됩니다. 디렉토리 캐싱이 사용 가능한지 확인하려면 GET DATABASE MANAGER CONFIGURATION 명령의 출력에 있는 dir\_cache 디렉토리 캐시 지원 구성 매개변수에 대한 값을 점검하십시오. 응용프로그램의 디렉토리 캐시는 첫 번째 디렉토리 찾아보기 중에 작성됩니다. 캐시는 응용프로그램이 디렉토리 파일 중 하나를 수정할 때만 새로 고쳐지므로, 다른 응용프로그램이 작성한 디렉토리 변경은 응용프로그램이 재시작될 때까지 적용되지 않을 수 있습니다.

CLP의 디렉토리 캐시를 새로 고치려면 TERMINATE 명령을 사용하십시오. 데이터베이스 관리 프로그램의 공유 캐시를 새로 고치려면 데이터베이스 관리 프로그램을 중지한 후(db2stop) 재시작하십시오(db2start). 다른 응용프로그램에 대한 디렉토리 캐시를 새로 고치려면 해당 응용프로그램을 중지한 후 재시작하십시오.



---

## 제 135 장 UNCATALOG ODBC DATA SOURCE

사용자 또는 시스템 ODBC 데이터 소스를 카탈로그 해제합니다.

ODBC(Open Database Connectivity) 용어에서 데이터 소스는 특정 데이터베이스에 대한 사용자 정의 이름입니다. 해당 이름이 ODBC API를 통해 데이터베이스에 액세스하는 데 사용됩니다. Windows에서는 사용자 또는 시스템 데이터 소스를 카탈로그 해제할 수 있습니다. 사용자 데이터 소스는 이를 카탈로그한 사용자만 볼 수 있는 반면, 시스템 데이터 소스는 다른 모든 사용자가 보고 사용할 수 있습니다.

이 명령은 Windows에서만 사용할 수 있습니다.

### 권한 부여

없음

### 필수 연결

없음

### 명령 구문

```
►► UNCATALOG { USER | SYSTEM } ODBC DATA SOURCE data-source-name ►►
```

### 명령 매개변수

**USER** 사용자 데이터 소스를 카탈로그 해제합니다. 이것은 키워드를 지정하지 않는 경우 디폴트입니다.

#### **SYSTEM**

시스템 데이터 소스를 카탈로그 해제합니다.

#### **ODBC DATA SOURCE** *data-source-name*

카탈로그 해제될 데이터 소스의 이름을 지정합니다. 최대 길이는 32문자입니다.

### 사용 시 참고사항

Microsoft Windows Vista 이상 버전에서는 전체 관리자 특권을 갖고 실행 중인 DB2 명령 창에서 UNCATALOG SYSTEM ODBC DATA SOURCE 명령을 실행해야 합니다.



---

## 제 136 장 UNQUIESCE

유지보수 또는 기타 이유로 Quiesce 상태에 있는 인스턴스 또는 데이터베이스로 사용자 액세스를 리스토어합니다. UNQUIESCE는 종료 및 데이터베이스 재시작을 요청하지 않고 사용자 액세스를 리스토어합니다.

특정하게 지정된 경우를 제외하고는 *sysadm*, *sysmaint* 또는 *sysctrl* 권한이 있는 사용자만 Quiesce 상태에 있는 데이터베이스에 액세스할 수 있습니다. 따라서 UNQUIESCE된 데이터베이스로 일반 액세스를 리스토어하려면 UNQUIESCE가 필요합니다.

### 범위

UNQUIESCE DB는 Quiesce 상태의 데이터베이스에 있는 모든 오브젝트로 사용자 액세스를 리스토어합니다.

UNQUIESCE INSTANCE *instance-name*는 인스턴스 *instance-name*에 있는 데이터베이스 및 이 인스턴스로 사용자 액세스를 리스토어합니다.

인스턴스를 중지하고 이 인스턴스 및 해당되는 모든 데이터베이스를 Unquiesce하려면 `db2stop` 명령을 발행하십시오. DB2를 중지한 후 재시작하면 모든 인스턴스 및 데이터베이스가 Unquiesce됩니다.

### 권한 부여

다음 중 하나가 필요합니다.

데이터베이스 레벨 Unquiesce를 수행하려는 경우:

- *sysadm*
- *dbadm*

인스턴스 레벨 Unquiesce를 수행하려는 경우:

- *sysadm*
- *sysctrl*

### 명령 구문

```
▶▶ UNQUIESCE — DB —————▶▶  
      |  
      └─ INSTANCE — instance-name ─┘
```

## 필수 연결

데이터베이스

(인스턴스 Unquiesce에는 데이터베이스 연결이 필요하지 않습니다.)

## 명령 매개변수

**DB** 데이터베이스를 Unquiesce하십시오. 데이터베이스에 있는 모든 오브젝트로 사용자 액세스가 리스토어됩니다.

**INSTANCE** *instance-name*

인스턴스 *instance-name* 및 이 인스턴스의 데이터베이스로 액세스가 리스토어됩니다.

**예:**

데이터베이스 **Unquiesce**

```
db2 unquiesce db
```

이 명령은 이전에 Quiesce되었던 데이터베이스를 Unquiesce합니다.

---

## 제 137 장 UPDATE ADMIN CONFIGURATION

DB2 Administration Server(DAS) 구성 파일에서 지정된 항목을 수정합니다. DAS는 DB2 서버의 리모트 관리를 가능하게 하는 특수 관리 도구입니다.

DAS를 설치할 때 구성 파일의 빈 사본이 각 실제 데이터베이스 파티션에 저장됩니다. 각 사본에 항목을 작성해야 합니다. 다음에 DAS를 시작할 때 사용될 다음 DAS 구성 매개변수를 지정할 수 있습니다.

- DB2 서버 시스템의 이름 - **db2system**
- DAS 관리 권한 그룹 이름 - **dasadm\_group**
- 스케줄러 모드 - **sched\_enable**
- 도구 카탈로그 데이터베이스 인스턴스 - **toolscat\_inst**
- 도구 카탈로그 데이터베이스 - **toolscat\_db**
- 도구 카탈로그 데이터베이스 스키마 - **toolscat\_schema**
- 만기 태스크 실행 - **exec\_exp\_task**
- 스케줄러 사용자 ID - **sched\_userid**
- 인증 유형 DAS - **authentication**

다음 DAS 구성 매개변수는 원래 및 DAS가 온라인인 동안 나중에 변경할 수 있습니다.

- DAS 발견 모드 - **discover**
- SMTP 서버 - **smtp\_server**
- JDK(Java Development Kit) 설치 경로 DAS - **jdk\_path**
- 문의처 목록 위치 - **contact\_host**
- DAS 코드 페이지 - **das\_codepage**
- DAS 지역 - **das\_territory**

이들 매개변수에 대한 자세한 정보는 개별 매개변수 설명을 참조하십시오.

### 범위

각 관리 노드에 대한 매개변수 설정을 지정 또는 변경하려면 해당 노드에서 이 명령을 발행하십시오.

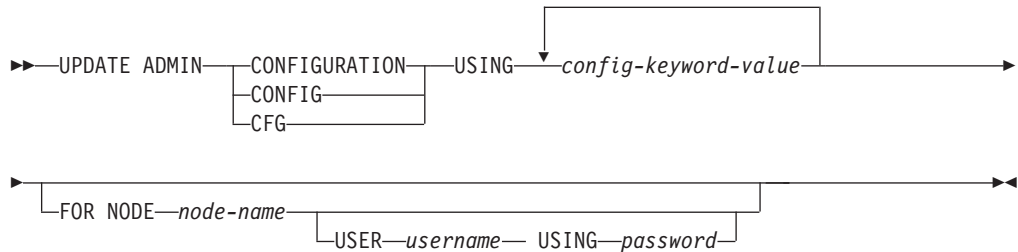
### 권한 부여

*dasadm*

## 필수 연결

노드 리모트 시스템의 DAS 구성을 갱신하려면 관리자 노드 이름과 함께 FOR NODE 옵션을 사용하십시오.

## 명령 구문



## 명령 매개변수

**USING** *config-keyword-value*

갱신할 관리 구성 매개변수를 지정합니다.

**FOR NODE**

DAS 구성 매개변수를 갱신할 관리 노드의 이름을 입력하십시오.

**USER** *username* **USING** *password*

관리 노드에 연결할 때 사용자 이름 및 암호가 필요한 경우 이 정보를 입력하십시오.

## 사용 시 참고사항

DAS 구성 매개변수의 목록을 보거나 인쇄하려면 GET ADMIN CONFIGURATION을 사용하십시오. DAS 구성 매개변수를 권장 DAS 디폴트로 재설정하려면 RESET ADMIN CONFIGURATION을 사용하십시오.

구성 매개변수가 적용되는 시기는 표준 구성 매개변수를 변경하는지 여부 또는 온라인 재설정할 수 있는 매개변수에 따라 다릅니다. 표준 구성 매개변수 값은 db2admin 명령을 실행할 때 재설정됩니다.

오류가 발생하면 DAS 구성 파일이 변경되지 않습니다.

UPDATE ADMIN CONFIGURATION을 사용하여 DAS 구성을 갱신하려면 DAS에 동일한 설치 레벨에 있는 인스턴스에서 명령행 처리기를 사용해야 합니다.

체크섬이 유효하지 않으면 DAS 구성 파일을 갱신할 수 없습니다. 적합한 명령을 사용하지 않고 DAS 구성 파일을 수동으로 변경하는 경우 이 오류가 발생할 수 있습니다. 이 경우 DAS를 삭제하고 재작성하여 구성 파일을 재설정해야 합니다.



## 제 138 장 UPDATE ALERT CONFIGURATION

Health 표시기의 경보 구성 설정값을 갱신합니다.

**중요사항:** Health Monitor가 버전 9.7에서 사용되지 않으므로 이 명령 또는 API는 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 버전 9.7의 새로운 내용 책에 있는 『Health Monitor는 사용되지 않음』 주제를 참조하십시오.

### 권한 부여

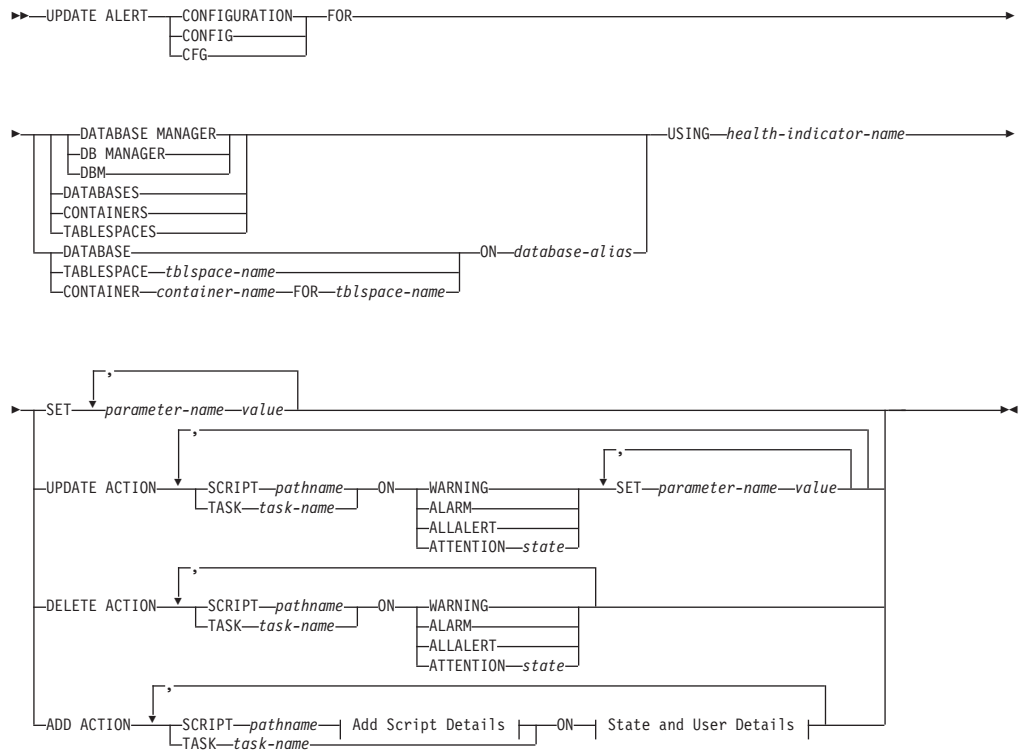
다음 중 하나가 필요합니다.

- *sysadm*
- *sysmaint*
- *sysctrl*

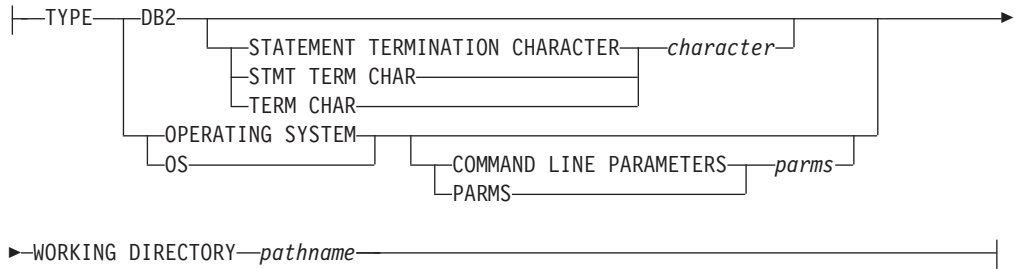
### 필요한 연결

인스턴스. 명시적 접속은 필요하지 않습니다.

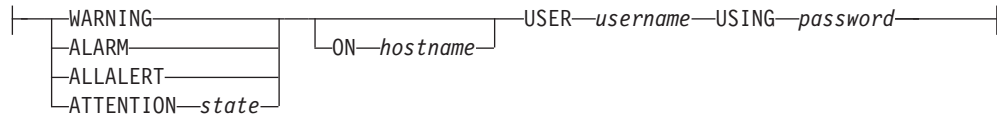
### 명령 구문



### Add Script Details:



### State and User Details:



## 명령 매개변수

### DATABASE MANAGER

데이터베이스 관리 프로그램에 대한 경보 설정값을 갱신합니다.

### DATABASES

데이터베이스 관리 프로그램이 관리하는 모든 데이터베이스에 대한 경보 설정값을 갱신합니다. 이들 설정값은 사용자 정의 설정값이 없는 모든 데이터베이스에 적용됩니다. 사용자 정의 설정값은 `DATABASE ON database-alias` 절을 사용하여 정의됩니다.

### CONTAINERS

데이터베이스 관리 프로그램이 관리하는 모든 테이블 스페이스 컨테이너에 대한 경보 설정값을 갱신합니다. 이들 설정값은 사용자 정의 설정값이 없는 모든 테이블 스페이스 컨테이너에 적용됩니다. 사용자 정의 설정은 `CONTAINER container-name ON database-alias` 절을 사용하여 정의됩니다.

### TABLESPACES

데이터베이스 관리 프로그램이 관리하는 모든 테이블 스페이스에 대한 경보 설정값을 갱신합니다. 이들 설정값은 사용자 정의 설정값이 없는 모든 테이블 스페이스에 적용됩니다. 사용자 정의 설정은 `TABLESPACE tblspace-name ON database-alias` 절을 사용하여 정의됩니다.

### DATABASE ON database-alias

`ON database-alias` 절을 사용하여 지정된 데이터베이스의 경보 설정을 갱신합니다. 이 데이터베이스가 사용자 정의 설정을 가지고 있을 경우, 이 설정은 `DATABASES` 매개변수를 사용하여 지정된 해당 인스턴스의 모든 데이터베이스에 대한 설정보다 우선합니다.

**CONTAINER** *container-name* **FOR** *tblspace-name* **ON** *database-alias*

**ON** *database-alias* 절을 사용하여 지정된 데이터베이스에서 **FOR** *tblspace-name* 절을 사용하여 지정된 테이블 스페이스에 대해 *container-name*이라고 하는 테이블 스페이스 컨테이너의 경보 설정을 갱신합니다. 이 테이블 스페이스 컨테이너에 사용자 정의 설정이 있는 경우, 이 설정은 **CONTAINERS** 매개변수를 사용하여 지정된 데이터베이스의 모든 테이블 스페이스 컨테이너에 대한 설정보다 우선합니다.

**TABLESPACE** *tblspace-name* **ON** *database-alias*

**ON** *database-alias* 절을 사용하여 지정된 데이터베이스에서 *name*이라고 하는 테이블 스페이스의 경보 설정을 갱신합니다. 이 테이블 스페이스에 사용자 정의 설정이 있는 경우, 이 설정은 **TABLESPACES** 매개변수를 사용하여 지정된 데이터베이스의 모든 테이블 스페이스에 대한 설정보다 우선합니다.

**USING** *health-indicator-name*

경보 구성이 갱신될 Health 표시기 세트를 지정합니다. Health 표시기 이름은 두 문자 오브젝트 ID와 그 뒤에 표시기가 측정하는 것을 설명하는 이름으로 구성됩니다. 예를 들면, 다음과 같습니다.

```
db.sort_privmem_util
```

**SET** *parameter-name value*

지정된 값에 대한 Health 표시기의 경보 구성 요소(*parameter-name*)를 갱신합니다. *parameter-name*은 다음 중 하나여야 합니다.

- **ALARM:** *value*는 Health 표시기 단위입니다.
- **WARNING:** *value*는 Health 표시기 단위입니다.
- **SENSITIVITY:** *value*는 초 단위입니다.
- **ACTIONSENABLED:** *value*는 YES 또는 NO가 될 수 있습니다.
- **THRESHOLDSCHECKED:** *value*는 YES 또는 NO가 될 수 있습니다.

**UPDATE ACTION SCRIPT** *pathname* **ON** [**WARNING** | **ALARM** | **ALLALERT** | **ATTENTION** *state*]

절대 경로 이름이 *pathname*인 사전 정의된 스크립트의 스크립트 속성이 다음 절에 따라 갱신됨을 지정합니다.

**SET** *parameter-name value*

스크립트 속성 *parameter-name*을 지정된 값으로 갱신합니다. *parameter-name*은 다음 중 하나여야 합니다.

- **SCRIPTTYPE**  
OS 또는 DB2가 유효한 유형입니다.
- **WORKINGDIR**
- **TERMCHAR**

- **CMDLINEPARMS**

운영 체제 스크립트에 대해 사용자가 지정하는 명령 매개변수는 디폴트 제공 매개변수 앞에 놓입니다. 운영 체제 스크립트로 보내는 매개변수는 다음과 같습니다.

- 사용자가 제공한 매개변수 목록
- Health 표시기 단축 이름
- 완전한 오브젝트 이름
- Health 표시기 값
- 경보 상태

- **USERID**

- **PASSWORD**

- **SYSTEM**

**UPDATE ACTION TASK** *task-name* **ON** [WARNING | ALARM | ALLALERT | ATTENTION *state*]

이름이 *name*인 태스크의 속성이 다음 절에 따라 갱신됨을 지정합니다.

**SET** *parameter-name value*

태스크 속성 *parameter-name*을 지정된 값으로 갱신합니다. *parameter-name*은 다음 중 하나여야 합니다.

- **USERID**

- **PASSWORD**

- **SYSTEM**

**DELETE ACTION SCRIPT** *pathname* **ON** [WARNING | ALARM | ALLALERT | ATTENTION *state*]

경보 조치 스크립트 목록에서 절대 경로 이름이 *pathname*인 조치 스크립트를 제거합니다.

**DELETE ACTION TASK** *task-name* **ON** [WARNING | ALARM | ALLALERT | ATTENTION *state*]

경보 조치 태스크 목록에서 이름이 *name*인 조치 태스크를 제거합니다.

**ADD ACTION SCRIPT** *pathname* **ON** [WARNING | ALARM | ALLALERT | ATTENTION *state*]

절대 경로 이름이 *pathname*인 새 조치 스크립트가 추가됨을 지정합니다. 속성은 다음과 같이 제공됩니다.

**TYPE** 조치 스크립트는 DB2 명령 스크립트나 운영 체제 스크립트여야 합니다.

- **DB2**

- **OPERATING SYSTEM**

DB2 명령 스크립트인 경우, 다음 절을 사용하여 명령문을 종료하기 위한 스크립트에서 사용되는 문자 *character*를 선택적으로 지정할 수 있습니다.

STATEMENT TERMINATION CHARACTER ;

운영 체제 스크립트인 경우, COMMAND LINE PARAMETERS *parms*절을 사용하여, 호출 시 스크립트에 전달되는 명령 매개변수 *parms*를 선택적으로 지정할 수 있습니다.

#### **WORKING DIRECTORY** *pathname*

스크립트가 실행될 디렉토리의 절대 경로 이름 *pathname*을 지정합니다.

#### **USER** *username* **USING** *password*

스크립트가 실행될 사용자 어카운트 *username* 및 연관된 암호 *password*를 지정합니다.

#### **ADD ACTION TASK** *name* **ON** [WARNING | ALARM | ALLALERT | ATTENTION *state*]

*name*이라고 하는 새 태스크가 지정된 조건에서 실행되도록 추가됨을 지정합니다.

#### **ON** [WARNING | ALARM | ALLALERT | ATTENTION *state*]

조치 또는 태스크가 실행될 조건을 지정합니다. 임계값 기반 Health 표시기(HI)의 경우, 이는 WARNING 또는 ALARM입니다. 상태 기반 HI의 경우, 이는 각 상태 기반 HI에 대해 문서화된 대로 숫자 상태가 됩니다(예를 들어, ts.ts\_op\_status Health 표시기의 경우 테이블 스페이스 상태에 대해 tablespace\_state 모니터 요소를 참조).

#### **ATTENTION** *state*

일부 데이터베이스 Health 표시기 상태에 대한 유효한 숫자 값이 ADD ACTION SCRIPT CLP 명령 옵션의 예로 아래에 제공되어 있습니다.

- 0 - 활성. 일반(ACTIVE)
- 1 - Quiesce 보류(QUIESCE\_PEND)
- 2 - Quiesce 상태(QUIESCED)
- 3 - 롤 포워드(ROLLFWD)

추가적인 Health 표시기가 헤더 파일 sqlmon.h 및 sqlutil.h에 정의되어 있습니다.

## 사용 시 참고사항

ADD ACTION 옵션의 경우, 제공된 *username* 및 *password*는 SQL문 텍스트가 캡처되는 다양한 위치에서 노출될 수 있습니다.

- 네트워크(사용자 이름/암호가 암호화되지 않은 연결을 통해 전달됨)

- db2diag 로그 파일
- 추적 파일
- 덤프 파일
- 스냅샷 모니터(동적 SQL 스냅샷)
- 시스템 모니터 스냅샷
- 여러 이벤트 모니터(명령문, 교착 상태)
- Query Patroller
- Explain 테이블
- db2pd 출력(그 중에서도 특히, 패키지 캐시 및 잠금 시간종료 메커니즘)
- DB2 감사 레코드

---

## 제 139 장 UPDATE ALTERNATE SERVER FOR DATABASE

시스템 데이터베이스 디렉토리에서 데이터베이스 별명의 대체 서버를 갱신합니다.

### 범위

이 명령은 명령이 실행되는 데이터베이스 파티션에만 영향을 미칩니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*

### 필수 연결

없음

### 명령 구문

```
►—UPDATE ALTERNATE SERVER FOR—DATABASE—database-alias—USING—►  
                                          └─DB─┘  
►—HOSTNAME—hostname—PORT—port-number—►
```

### 명령 매개변수

**DATABASE** *database-alias*

대체 서버가 갱신될 데이터베이스의 별명을 지정합니다.

**HOSTNAME** *hostname*

데이터베이스의 대체 서버가 있는 노드의 IP 주소 또는 완전한 호스트 이름을 지정합니다.

**PORT** *port-number*

데이터베이스 관리 프로그램 인스턴스 대체 서버의 포트 번호를 지정합니다.

### 예:

다음 예에서는 호스트 이름 *montero*와 포트 20396을 사용하여 **SAMPLE** 데이터베이스의 대체 서버를 갱신합니다.

```
db2 update alternate server for database sample using hostname montero port 20396
```

다음 두 예에서는 **SAMPLE** 데이터베이스의 대체 서버를 재설정합니다.

```
db2 update alternate server for database sample using hostname NULL port NULL
```

또는

```
db2 update alternate server for database sample using hostname "" port NULL
```

## 사용 시 참고사항

- 이 명령은 시스템 데이터베이스 디렉토리에만 적용됩니다.
- 이 명령은 서버 인스턴스에만 사용해야 합니다. 이 명령이 클라이언트 인스턴스에서 실행되면 명령이 무시되고 SQL1889W 메시지가 리턴됩니다.
- 명령이 실행되는 머신에서 LDAP(Lightweight Directory Access Protocol) 지원을 사용하는 경우 데이터베이스의 대체 서버가 LDAP 디렉토리에 자동으로 등록됩니다.



---

## 제 140 장 UPDATE ALTERNATE SERVER FOR LDAP DATABASE

LDAP(Lightweight Directory Access Protocol)에서 데이터베이스의 대체 서버를 갱신합니다.

### 권한 부여

LDAP 서버에 대한 읽기/쓰기 액세스.

### 필수 연결

없음

### 명령 구문

```
►► UPDATE ALTERNATE SERVER FOR LDAP DATABASE database-alias
      DB
► USING NODE node
      GWNODE gwnode USER username
      PASSWORD password
```

### 명령 매개변수

#### **DATABASE** *database-alias*

갱신될 데이터베이스의 별명을 지정합니다.

#### **NODE** *node*

데이터베이스의 대체 서버가 있는 노드 이름을 지정합니다.

#### **GWNODE** *gwnode*

데이터베이스의 대체 게이트웨이가 있는 노드 이름을 지정합니다.

#### **USER** *username*

사용자의 LDAP 식별 이름(DN)을 지정합니다. LDAP 사용자 DN은 LDAP 디렉토리에서 오브젝트를 작성할 수 있는 충분한 권한이 있어야 합니다. 사용자의 LDAP DN이 지정되지 않으면 현재 로그인 사용자의 증명서가 사용됩니다.

db2ldcfg를 사용하여 사용자의 LDAP DN과 암호가 지정된 경우 사용자 이름과 암호를 여기서 지정하지 않아도 됩니다.

#### **PASSWORD** *password*

어카운트 암호

db2ldcfg를 사용하여 사용자의 LDAP DN과 암호가 지정된 경우 사용자 이름과 암호를 여기서 지정하지 않아도 됩니다.

---

## 제 141 장 UPDATE CLI CONFIGURATION

db2cli.ini 파일의 지정된 섹션의 콘텐츠를 갱신합니다.

db2cli.ini 파일은 DB2 콜 레벨 인터페이스(CLI) 구성 파일로 사용됩니다. DB2 CLI 및 이를 사용하는 응용프로그램의 동작을 수정하는 데 사용할 수 있는 다양한 키워드 및 값이 들어 있습니다. 이 파일은 섹션으로 나뉘어지며, 각 섹션은 데이터베이스 별명 이름에 대응합니다.

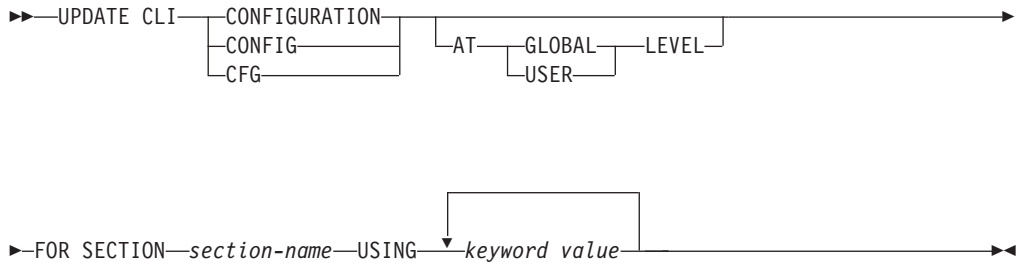
### 권한 부여

없음

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

#### FOR SECTION *section-name*

키워드가 갱신될 섹션의 이름입니다. 지정된 섹션이 존재하지 않는 경우 새 섹션이 작성됩니다.

#### AT GLOBAL LEVEL

CLI 구성 매개변수가 전역 레벨에서 갱신되도록 지정합니다. 이 매개변수는 LDAP 지원이 사용될 때만 적용할 수 있습니다.

#### AT USER LEVEL

CLI 구성 매개변수가 사용자 레벨에서 갱신되도록 지정합니다. LDAP 지원이 사용 가능한 경우 이 설정은 동일한 LDAP 사용자 ID로 다른 머신에 로그인할 때 일관성이 있습니다. LDAP 지원이 사용 불가능한 경우 이 설정은 동일한 운영 체제 사용자 ID로 동일한 머신에 로그인할 때만 일관성이 있습니다.

**USING** *keyword value*

갱신될 CLI/ODBC 매개변수를 지정합니다.

## 사용 시 참고사항

이 명령에서 지정되는 섹션 이름과 키워드는 대소문자를 구분하지 않습니다. 그러나 키워드 값은 대소문자를 구분합니다.

키워드 값이 작은따옴표나 임베드된 공백을 포함하는 문자열인 경우 전체 문자열을 큰따옴표로 구분해야 합니다. 예를 들면, 다음과 같습니다.

```
db2 update cli cfg for section tstcli1x
using TableType "'TABLE','VIEW','SYSTEM TABLE'"
```

AT USER LEVEL 키워드가 지정될 때, 지정된 섹션에 대한 CLI 구성 매개변수는 현재 사용자에게 대해서만 갱신됩니다. 그렇지 않으면 로컬 시스템의 모든 사용자에게 대해 갱신됩니다. 사용자 레벨에서의 CLI 구성은 LDAP 디렉토리에 유지되며 로컬 시스템에 캐시됩니다. CLI 구성을 읽을 때 DB2는 항상 캐시에서 읽습니다. 캐시는 다음 상황에서 새로 고쳐집니다.

- 사용자가 CLI 구성을 갱신합니다.
- 사용자가 명시적으로 REFRESH LDAP 명령을 사용하여 CLI 구성을 강제로 새로 고칩니다.

LDAP 환경에서 사용자는 LDAP 디렉토리에 카탈로그된 데이터베이스에 대한 디폴트 CLI 설정 세트를 구성할 수 있습니다. LDAP 카탈로그 데이터베이스가 구성 지원 프로그램(CA) 또는 ODBC 구성 유틸리티를 사용하여 데이터 소스 이름(DSN)으로 추가될 때 모든 디폴트 CLI 설정은 LDAP 디렉토리에 존재하는 경우 로컬 시스템의 해당 DSN에 대해 구성됩니다. CLI 매개변수를 디폴트 설정으로 구성하려면 AT GLOBAL LEVEL 절을 지정해야 합니다.

## 제 142 장 UPDATE COMMAND OPTIONS

대화식 세션 중에 또는 일괄처리 입력 파일에서 하나 이상의 명령 옵션을 설정합니다. 대화식 세션 또는 일괄처리 입력 파일이 종료할 때 설정은 시스템 디폴트값으로 되돌아갑니다(또는 시스템 디폴트값이 DB2OPTIONS에서 겹쳐 씁니다).

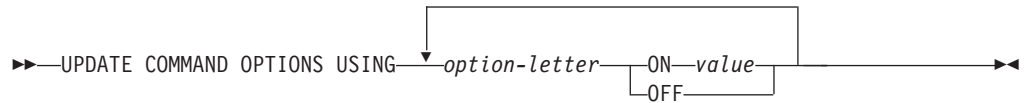
### 권한 부여

없음

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

#### USING *option-letter*

다음 옵션 문자를 설정할 수 있습니다.

- a** SQLCA 표시
- c** SQL문 자동 커밋
- d** XML 데이터의 XML 선언 표시
- e** SQLCODE/SQLSTATE 표시
- i** 적합한 들여쓰기를 사용하여 XQuery 결과 표시
- l** 실행기록 파일에 명령 로그
- m** INSERT, DELETE, UPDATE 또는 MERGE문의 영향을 받는 행 수 표시
- n** 개행 문자 제거
- o** 표준 출력에 표시
- p** DB2 대화식 프롬프트 표시
- q** 작은따옴표나 큰따옴표로 구분된 문자열에서 공백 및 줄 바꾸기 보존.
- r** 출력 보고서를 파일에 저장

- s 명령 오류 시 실행 중지
- v 현재 명령 에코
- w SQL문 경고 메시지 표시
- z 모든 출력을 파일로 경로 재지정.

**ON value**

e, l, r 및 z 옵션은 켜지는 경우 값이 필요합니다. e 옵션의 경우 *value*는 SQLCODE를 표시하는 c 또는 SQLSTATE를 표시하는 s일 수 있습니다. l, r 및 z 옵션의 경우 *value*는 실행기록 파일 또는 보고서 파일에 사용되는 이름을 표시합니다. 다른 옵션은 값을 승인하지 않습니다.

**사용 시 참고사항**

이들 설정은 시스템 디폴트값, DB2OPTIONS의 설정 및 명령행 옵션 플래그를 사용하여 지정된 옵션을 겹쳐씁니다.

파일 입력 옵션(-f) 및 명령문 종료 옵션(-t)은 이 명령을 사용하여 갱신할 수 없습니다.

현재 옵션 설정을 보려면 LIST COMMAND OPTIONS 명령을 사용하십시오.

---

## 제 143 장 UPDATE CONTACT

로컬 시스템에 정의되어 있는 문의처의 속성을 갱신합니다. 문의처는 스케줄러 및 Health Monitor가 메시지를 보내는 사용자입니다. 문의처를 작성하려면 ADD CONTACT 명령을 사용하십시오. DAS(Database Administration Server) **contact\_host** 구성 매개 변수의 설정은 목록이 로컬인지 또는 전역인지를 판별합니다.

### 권한 부여

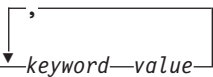
없음

### 필수 연결

없음. 로컬 실행 전용: 이 명령은 리모트 연결에 사용할 수 없습니다.

### 명령 구문

```
►► UPDATE CONTACT name USING keyword value ◀◀
```



### 명령 매개변수

**UPDATE CONTACT** *name*

갱신될 문의처 이름.

**USING** *keyword value*

갱신될 문의처 매개변수(*keyword*) 및 이 매개변수가 설정될 값(*value*)을 지정합니다. 유효한 키워드 세트는 다음과 같습니다.

#### **ADDRESS**

SMTP 서버가 통지를 보내기 위해 사용하는 전자 우편 주소.

**TYPE** 주소가 전자 우편 주소인지 또는 호출기인지를 나타냅니다.

#### **MAXPAGELEN**

호출기가 승인할 수 있는 최대 문자 수.

#### **DESCRIPTION**

문의처의 텍스트 설명. 최대 128 문자를 사용합니다.





---

## 제 144 장 UPDATE CONTACTGROUP

로컬 시스템에 정의되어 있는 문의처 그룹의 속성을 갱신합니다. 문의처 그룹은 스케줄러 및 Health Monitor가 통지해야 하는 사용자 목록입니다. DAS(Database Administration Server) **contact\_host** 구성 매개변수의 설정은 목록이 로컬인지 또는 전역인지를 판별합니다.

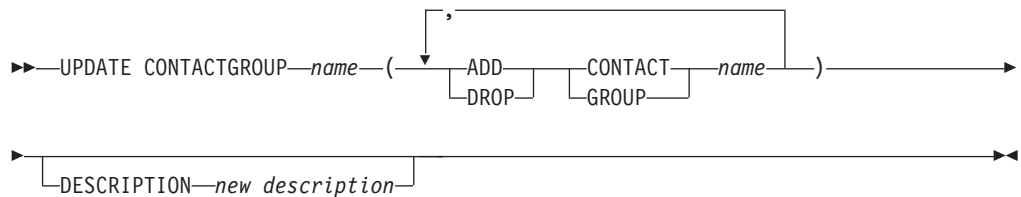
### 권한 부여

없음

### 필요한 연결

없음

### 명령 구문



### 명령 매개변수

#### CONTACTGROUP *name*

갱신될 문의처 그룹 이름

#### ADD CONTACT *name*

그룹에 추가될 새 문의처 이름을 지정합니다. 문의처를 그룹에 추가한 후 ADD CONTACT 명령을 사용하여 정의할 수 있습니다.

#### DROP CONTACT *name*

그룹에서 삭제될 그룹의 문의처 이름을 지정합니다.

#### ADD GROUP *name*

그룹에 추가될 새 문의처 그룹의 이름을 지정합니다.

#### DROP GROUP *name*

그룹에서 삭제될 문의처 그룹 이름을 지정합니다.

#### DESCRIPTION 새 설명

선택사항. 문의처 그룹의 새 텍스트 설명.



## 제 145 장 UPDATE DATABASE CONFIGURATION

특정 데이터베이스 구성 파일에 있는 개별 항목을 수정합니다.

데이터베이스 구성 파일은 해당 데이터베이스가 작성된 모든 데이터베이스 파티션에 있습니다.

### 범위

DBPARTITIONNUM이 하나의 데이터베이스 파티션만을 갱신하도록 지정된 경우를 제외하고, 이 명령은 디폴트로 모든 데이터베이스 파티션을 갱신합니다.

### 권한 부여

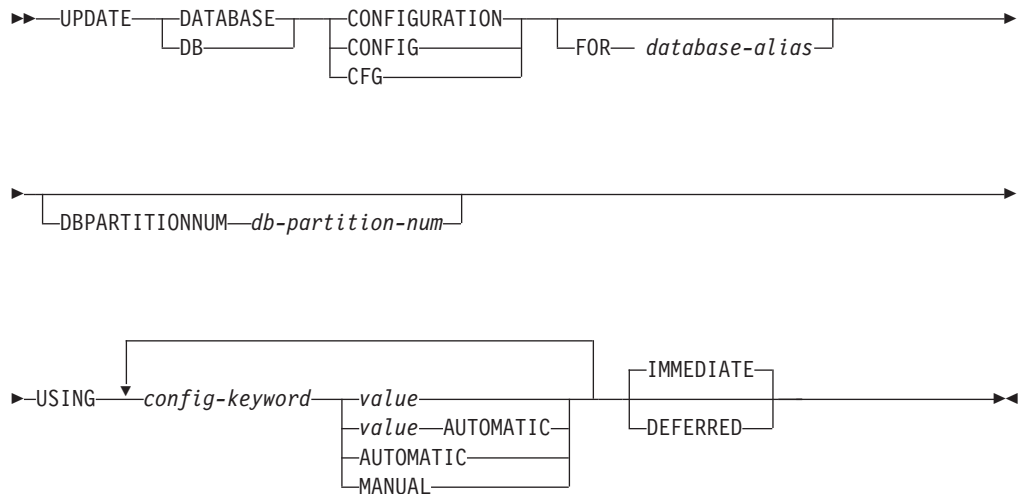
다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*

### 필수 연결

인스턴스. 명시적 접속은 필요하지 않지만 데이터베이스를 사용 중일 때는 데이터베이스 연결을 권장합니다. 데이터베이스가 리모트로 나열된 경우, 이 명령의 지속 기간 동안 리모트 노드에 대한 인스턴스 접속이 설정됩니다. 매개변수를 온라인으로 변경하려면 데이터베이스에 연결되어야 합니다.

### 명령 구문



## 명령 매개변수

### AUTOMATIC

일부 구성 매개변수는 AUTOMATIC으로 설정되어 DB2가 자동으로 이 매개변수를 조정하여 현재 자원 요구사항을 반영할 수 있습니다. AUTOMATIC 키워드를 지원하는 구성 매개변수 목록에 대해서는 구성 매개변수 요약을 참조하십시오. AUTOMATIC 키워드를 따라 값이 지정된 경우, 자동 계산에 영향을 미칠 수 있습니다. 이 작동에 대한 특정 정보는 구성 매개변수의 문서를 참조하십시오.

### DEFERRED

구성 파일에서만 변경사항을 작성하여 다음 번에 데이터베이스를 다시 활성화할 때 이 변경사항이 적용되도록 하십시오.

### FOR *database-alias*

구성을 갱신할 데이터베이스의 별명을 지정합니다. 데이터베이스 연결이 이미 설정되었을 때는 데이터베이스 별명을 지정할 필요가 없습니다. 동일한 데이터베이스 인스턴스를 사용하는 다른 데이터베이스의 구성 파일을 갱신할 수 있습니다. 예를 들어, db11 데이터베이스에만 연결되어 있는 경우 `update db config for alias db22 using .... immediate:`

- db22에 사용 중인 연결이 없는 경우, 구성 파일만 갱신하면 되기 때문에 갱신이 완료됩니다. 새 연결(데이터베이스를 활성화함)은 메모리 내의 새로운 변경사항을 표시합니다.
- db22에 다른 응용프로그램에서 사용 중인 연결이 있으면 갱신이 디스크에서는 작동하나 메모리에서는 작동하지 않습니다. 데이터베이스를 재시작하라는 경고가 수신됩니다.

### DBPARTITIONNUM *db-partition-num*

데이터베이스 구성 갱신이 특정 데이터베이스 파티션에 적용되는 경우, 이 매개변수를 사용할 수 있습니다. 이 매개변수가 제공되지 않으면, 모든 데이터베이스 파티션에서 갱신이 유효합니다.

### IMMEDIATE

데이터베이스가 실행되는 동안 변경사항을 즉시 작성합니다. IMMEDIATE는 디폴트 조치이지만 적용하려면 데이터베이스 연결이 필요합니다.

이것은 CLPPlus 인터페이스에서 조작 시 디폴트 절입니다. CLPPlus 프로세서 사용 시 IMMEDIATE를 호출할 필요가 없습니다.

### MANUAL

구성 매개변수의 자동 성능 조정을 사용하지 않습니다. 매개변수는 현재 내부 값으로 설정되며 더 이상 자동으로 갱신되지 않습니다.

**USING** *config-keyword value*

*config-keyword*는 갱신될 데이터베이스 구성 매개변수를 지정합니다. *value*는 매개변수에 지정될 값을 지정합니다.

**예:**

**DPF(다중 파티션) 인스턴스에서 데이터베이스 구성 갱신**

이 예는 SAMPLE이라는 데이터베이스에서 데이터베이스 구성 매개변수 **MAXAPPLS**를 10에서 50으로 갱신하는 방법을 설명합니다.

사용자는 db2nodes.cfg에 정의된 대로 4개의 파티션이 있는 DPF 인스턴스를 갖습니다.

```
10 gilera 0
20 gilera 1
30 motobi 0
40 motobi 1
```

사용자가 인스턴스에 대한 SAMPLE 데이터베이스를 작성했습니다. SAMPLE의 카탈로그 파티션이 dbpartitionnum 10에 있습니다. 사용자가 motobi 시스템에 로그인되었다고 가정하십시오.

DPF 인스턴스의 디폴트 동작은 모든 데이터베이스 파티션에서 데이터베이스 구성을 갱신하므로, 사용자가 발행한 다음 명령은 모든 데이터베이스 파티션에서 **MAXAPPLS**의 동일한 결과 값을 가져옵니다.

```
db2 update db cfg for sample using maxappls 50
```

dbpartitionnum 30에서 **MAXAPPLS**만을 갱신하기 위해, 다음 명령을 발행할 수 있습니다.

```
db2 update db cfg for sample dbpartitionnum 30 using maxappls 50
```

또는

```
export DB2NODE=30
db2 update db cfg for sample using maxappls 50
```

## 사용 시 참고사항

데이터베이스 구성 매개변수 목록을 보거나 인쇄하려면 **GET DATABASE CONFIGURATION** 명령을 사용하십시오.

모든 데이터베이스 구성 매개변수를 권장 디폴트값으로 재설정하려면 **RESET DATABASE CONFIGURATION** 명령을 사용하십시오.

데이터베이스 구성 매개변수를 변경하려면 UPDATE DATABASE CONFIGURATION 명령을 사용하십시오. 예를 들어, ZELLMART라는 데이터베이스가 들어 있는 단일 파티션 데이터베이스 환경에서 로깅 모드를 『아카이브 로깅』으로 변경하려면 다음 명령을 사용하십시오.

```
db2 update db cfg for zellmart using logretain recovery
```

**logretain** 구성 매개변수가 변경되었는지 확인하려면 다음 명령을 사용하십시오.

```
db2 get db cfg for zellmart
```

예를 들어, 『zellmart』라는 데이터베이스가 들어 있는 다중 파티션 데이터베이스 환경에서 모든 파티션에 대해 로깅 모드를 『아카이브 로깅』으로 변경하려면(디폴트로 레지스트리 변수 DB2\_UPDDBCFG\_SINGLE\_DBPARTITION이 NULL 또는 FALSE로 설정됨) 다음 명령을 사용하십시오.

```
db2 update db cfg for zellmart using logretain recovery
```

**logretain** 구성 매개변수가 모든 데이터베이스 파티션에서 변경되었는지 확인하려면 다음 명령을 사용하십시오.

```
db2_all ";db2 get db cfg for zellmart"
```

위와 동일한 예를 사용하여, 하나의 특정 파티션(30)만을 로깅 모드로 갱신하려면 다음을 사용하십시오.

```
db2 update db cfg for zellmart dbpartitionnum 30 using logretain recovery
```

선택적으로 db2\_all을 사용하지 않고 모든 파티션에서 데이터를 가져오도록 SYSIBMADM.DBCFG 뷰를 보강할 수 있습니다.

UNIX 운영 체제에서 작업 중이며 『grep』 명령을 사용할 수 있는 경우, 다음 명령을 사용하여 **logretain** 값만 볼 수 있습니다.

```
db2_all ";db2 get db cfg for zellmart | grep -i logretain"
```

DB2 구성 매개변수 및 각 데이터베이스 노드 유형에 사용할 수 있는 값에 대한 자세한 정보는 개별 구성 매개변수 설명을 참조하십시오. 이들 매개변수의 값은 구성된 각 데이터베이스 노드 유형(서버, 클라이언트 또는 리모트 클라이언트가 있는 서버)에 따라 다릅니다.

일부 매개변수는 갱신할 수 있습니다.

데이터베이스 구성 파일에 대한 일부 변경사항은 메모리로 로드된 후에만 유효하게 됩니다. 이렇게 되려면 데이터베이스에서 모든 응용프로그램의 연결을 끊어야 합니다. 어떤 매개변수가 온라인으로 구성 가능하고 어떤 매개변수가 그렇지 않은지에 대한 자세한 정보는 구성 매개변수의 요약 목록을 참조하십시오.

예를 들어 SALES 데이터베이스에 대한 **sortheap** 데이터베이스 구성 매개변수를 온라인으로 변경하려면 다음 명령을 입력하십시오.

```
db2 connect to sales
db2 update db cfg using sortheap 1000
db2 connect reset
```

오류가 발생하면 데이터베이스 구성 파일이 변경되지 않습니다. 체크섬이 유효하지 않으면 데이터베이스 구성 파일을 갱신할 수 없습니다. 적절한 명령을 사용하지 않고 데이터베이스 구성 파일을 변경할 경우 이러한 일이 발생할 수 있습니다. 이러한 경우 데이터베이스를 리스토어하여 데이터베이스 구성 파일을 재설정해야 합니다.





## 제 146 장 UPDATE DATABASE MANAGER CONFIGURATION

현재 연결된 데이터베이스가 있는 인스턴스의 데이터베이스 관리 프로그램 구성 파일.

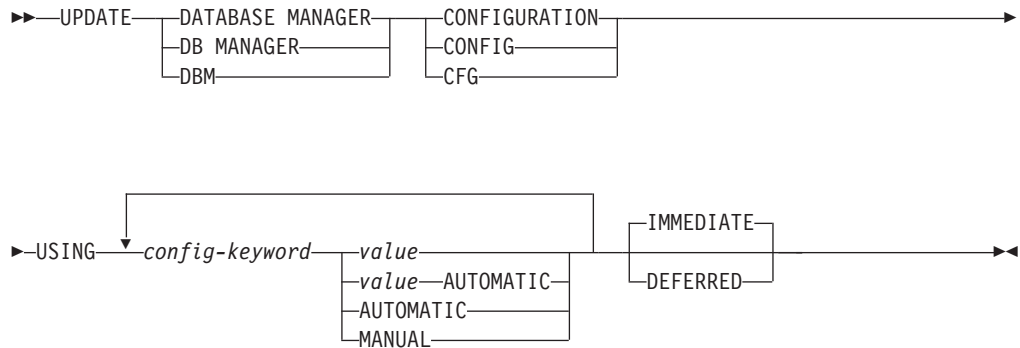
### 권한 부여

*sysadm*

### 필수 연결

없음 또는 인스턴스 로컬 DBM 구성 작업을 수행하는 데는 인스턴스 접속이 필요하지 않지만 리모트 DBM 구성 작업을 수행하는 데는 인스턴스 접속이 필요합니다. 리모트 인스턴스에 대한 데이터베이스 관리 프로그램 구성을 갱신하려면 먼저 이 인스턴스에 접속해야 합니다. 구성 매개변수를 온라인으로 갱신하는 경우에도 먼저 인스턴스에 접속해야 합니다.

### 명령 구문



### 명령 매개변수

#### AUTOMATIC

일부 구성 매개변수는 AUTOMATIC으로 설정되어 DB2가 자동으로 이 매개변수를 조정하여 현재 자원 요구사항을 반영할 수 있습니다. AUTOMATIC 키워드를 지원하는 구성 매개변수 목록에 대해서는 구성 매개변수 요약을 참조하십시오. AUTOMATIC 키워드를 따라 값이 지정된 경우, 자동 계산에 영향을 미칠 수 있습니다. 이 작동에 대한 특정 정보는 구성 매개변수의 문서를 참조하십시오.

주:

#### DEFERRED

구성 파일에서만 변경사항을 작성하여 인스턴스가 재시작될 때 이 변경사항이 적용되도록 하십시오.

이것은 CLPPlus 인터페이스에서 조작 시 디폴트 절입니다. CLPPlus 프로세서 사용 시 DEFERRED를 호출할 필요가 없습니다.

### IMMEDIATE

인스턴스가 실행되는 동안 변경사항을 즉시 동적으로 작성하십시오. IMMEDIATE는 디폴트이지만 적용되려면 인스턴스 접속이 필요합니다.

### MANUAL

구성 매개변수의 자동 성능 조정을 사용하지 않습니다. 매개변수는 현재 내부 값으로 설정되며 더 이상 자동으로 갱신되지 않습니다.

### USING *config-keyword value*

갱신될 데이터베이스 관리 프로그램 구성 매개변수를 지정합니다. 구성 매개변수 목록에 대해서는 구성 매개변수 요약을 참조하십시오. *value*는 매개변수에 지정될 값을 지정합니다.

## 사용 시 참고사항

데이터베이스 관리 프로그램 구성 매개변수의 목록을 보거나 인쇄하려면 GET DATABASE MANAGER CONFIGURATION 명령을 사용하십시오. 데이터베이스 관리 프로그램 구성 매개변수를 권장 데이터베이스 관리 프로그램 디폴트값으로 재설정하려면 RESET DATABASE MANAGER CONFIGURATION 명령을 사용하십시오. 구성된 각 데이터베이스 노드 유형에 적절한 데이터베이스 관리 프로그램 구성 매개변수 및 이들 매개변수의 값에 대한 자세한 정보는 개별 구성 매개변수 설명을 참조하십시오.

일부 매개변수는 갱신할 수 있습니다.

데이터베이스 관리 프로그램 구성 파일에 대한 일부 변경사항은 메모리로 로드된 후에만 적용됩니다. 어떤 매개변수가 온라인으로 구성 가능하고 어떤 매개변수가 그렇지 않은지에 대한 자세한 정보는 구성 매개변수 요약을 참조하십시오. 즉시 재설정되지 않는 서버 구성 매개변수는 db2start 실행 중 재설정됩니다. 클라이언트 구성 매개변수의 경우, 매개변수는 다음 번에 응용프로그램을 재시작할 때 재설정됩니다. 클라이언트가 명령행 처리기일 경우 TERMINATE를 호출해야 합니다.

예를 들어 데이터베이스 관리 프로그램의 eastern 인스턴스에 대한 DIAGLEVEL 데이터베이스 관리 프로그램 구성 매개변수를 온라인으로 변경하려면 다음 명령을 입력하십시오.

```
db2 attach to eastern
db2 update dbm cfg using DIAGLEVEL 1
db2 detach
```

오류가 발생하면 데이터베이스 관리 프로그램 구성 파일이 변경되지 않습니다.

체크섬이 유효하지 않으면 데이터베이스 관리 프로그램 구성 파일을 갱신할 수 없습니다. 데이터베이스 관리 프로그램 구성 파일을 편집하며 적절한 명령을 사용하지 않을 경우 이러한 일이 발생할 수 있습니다. 체크섬이 유효하지 않으면 데이터베이스 관리 프로그램을 재설치하여 데이터베이스 관리 프로그램 구성 파일을 재설정해야 합니다.

현재 인스턴스에 대한 **SVCENAME** 또는 **TPNAME** 데이터베이스 관리 프로그램 구성 매개변수를 갱신할 때 LDAP 지원이 사용 가능하거나 이 인스턴스에 대해 LDAP 서버가 등록되어 있으면 LDAP 서버가 이 새 값으로 갱신됩니다.



---

## 제 147 장 UPDATE HEALTH NOTIFICATION CONTACT LIST

인스턴스에 의해 발행된 Health 경보에 대한 통지의 문의처 목록을 갱신합니다.

### 권한 부여

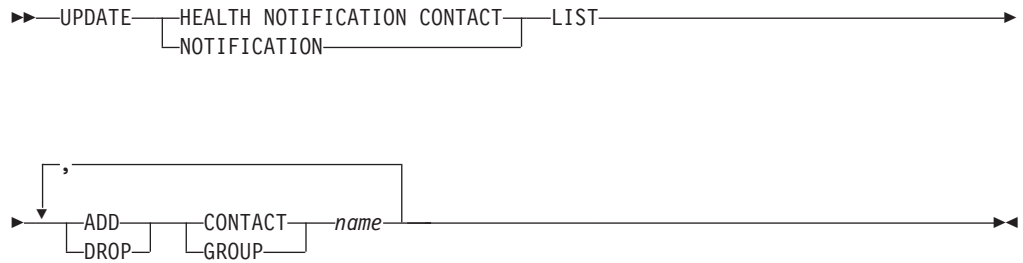
다음 중 하나가 필요합니다.

- sysadm
- sysctrl
- sysmaint

### 필요한 연결

인스턴스. 명시적 접속은 필요하지 않습니다.

### 명령 구문



### 명령 매개변수

#### **ADD GROUP** *name*

인스턴스의 Health를 통지하는 새 문의처 그룹을 추가합니다.

#### **ADD CONTACT** *name*

인스턴스의 Health를 통지할 새 문의처를 추가합니다.

#### **DROP GROUP** *name*

인스턴스의 Health를 통지할 문의처 목록에서 문의처 그룹을 제거합니다.

#### **DROP CONTACT** *name*

인스턴스의 Health를 통지할 문의처 목록에서 문의처를 제거합니다.



---

## 제 148 장 UPDATE HISTORY

실행기록 파일 항목에서 위치, 디바이스 유형, 주석 또는 상태를 갱신합니다.

### 권한 부여

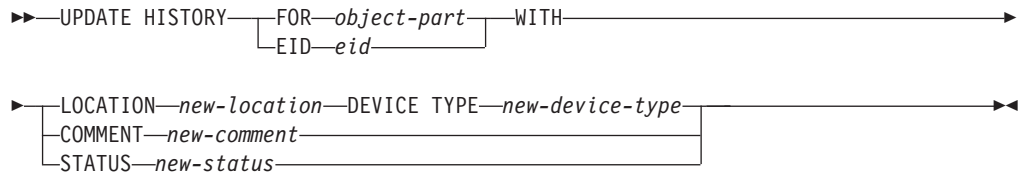
다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*

### 필수 연결

데이터베이스

### 명령 구문



### 명령 매개변수

#### FOR *object-part*

갱신될 실행기록 항목의 ID를 지정합니다. 001부터 999까지의 선택적 시퀀스 번호가 있는 시간소인입니다. 항목 상태를 갱신하는 데는 이 매개변수를 사용할 수 없습니다. 항목 상태를 갱신하려면 대신 EID를 지정하십시오.

#### EID *eid*

실행기록 항목 ID를 지정합니다.

#### LOCATION *new-location*

백업 이미지의 새 실제 위치를 지정합니다. 이 매개변수의 해석은 디바이스 유형에 따라 다릅니다.

#### DEVICE TYPE *new-device-type*

백업 이미지를 저장할 새 디바이스 유형을 지정합니다. 유효한 디바이스 유형은 다음과 같습니다.

**D**    디스크

- K** 디스켓
- T** 테이프
- A** Tivoli Storage Manager
- F** 스냅샷 백업
- U** User exit
- P** 파이프
- N** 널(NULL) 디바이스
- X** XBSA
- Q** SQL문
- O** 기타

**COMMENT** *new-comment*

항목을 설명할 새 주석을 지정합니다.

**STATUS** *new-status*

항목에 대한 새 상태를 지정합니다. 백업 항목만 상태를 갱신할 수 있습니다. 가능한 값은 다음과 같습니다.

- A** 사용 중. 백업 이미지가 사용 중인 로그 체인에 있습니다. 대부분의 항목이 사용 중입니다.
- I** 비활동. 현재 로그 시퀀스(현재 로그 체인이라고도 함)에 해당하지 않는 백업 이미지는 비활성으로 플래그됩니다.
- E** 만기됨. NUM\_DB\_BACKUPS를 초과한 활성 이미지가 있으므로 더 이상 필요하지 않은 백업 이미지는 만기된 것으로 플래그됩니다.
- D** 삭제됨. 더 이상 복구에 사용할 수 없는 백업 이미지는 삭제된 것으로 표시되어야 합니다.
- X** 삭제되지 않음. DB2HISTORY\_STATUS\_DO\_NOT\_DELETE로 표시된 복구 실행기록 파일 항목은 PRUNE HISTORY 명령을 호출하여 프른되지 않으며, PRUNE HISTORY로 ADMIN\_CMD 프로시저를 실행하며, db2Prune API나 자동화된 복구 실행기록 파일 프른을 호출합니다. DB2HISTORY\_STATUS\_DO\_NOT\_DELETE 상태를 사용하여 키 복구 파일 항목이 프른되지 않게 하고 연관된 복구 오브젝트가 삭제되지 않게 할 수 있습니다. 로그 파일, 백업 이미지 및 로드 복사 이미지만 DB2HISTORY\_STATUS\_DO\_NOT\_DELETE로 표시될 수 있습니다.



예 :

1997년 4월 13일 오전 10시에 수행된 전체 데이터베이스 백업에 대한 실행기록 파일을 갱신하려면 다음을 입력하십시오.

```
db2 update history for 19970413100000001 with
location /backup/dbbackup.1 device type d
```

## 사용 시 참고사항

데이터베이스 실행기록 파일의 1차 목적은 정보를 기록하는 것이지만 실행기록에 있는 데이터는 자동 리스토어 조작에서 직접 사용됩니다. AUTOMATIC 옵션을 지정하여 리스토어하는 동안 리스토어 유틸리티는 백업 이미지의 실행기록과 해당 위치를 참조 및 사용하여 자동 리스토어 요청을 이행합니다. 자동 리스토어 기능을 사용하며 백업 이미지가 작성된 이후 그 위치가 변경된 경우, 현재 위치를 반영하도록 해당 이미지에 대한 데이터베이스 실행기록 레코드를 갱신할 것을 권장합니다. 데이터베이스 실행기록에 있는 백업 이미지 위치를 갱신하지 않으면 자동 리스토어로 백업 이미지를 찾을 수 없으므로 직접 리스토어 명령을 사용할 수 있습니다.



## 제 149 장 UPDATE LDAP NODE

LDAP(Lightweight Directory Access Protocol)에서 DB2 서버를 표시하는 노드 항목과 연관된 프로토콜 정보를 갱신합니다.

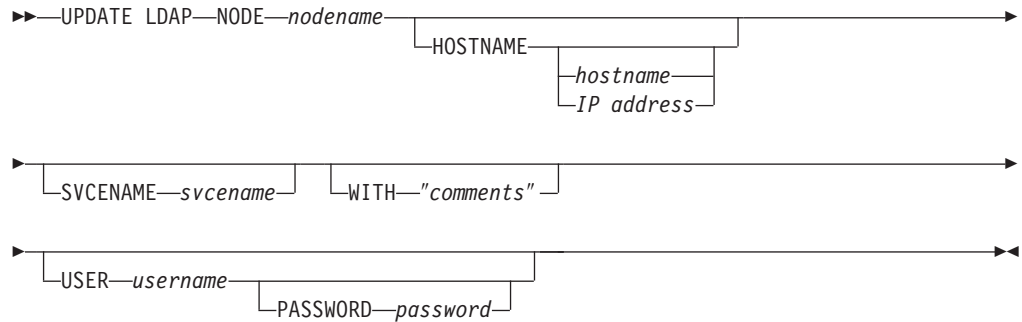
### 권한 부여

없음

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

#### **NODE** *nodename*

리모트 DB2 서버를 갱신할 때 노드 이름을 지정합니다. 노드 이름은 LDAP에 DB2 서버를 등록할 때 지정되는 값입니다.

#### **HOSTNAME** *hostname* | *IP address*

TCP/IP 호스트 이름 또는 IP 주소를 지정합니다.

- TCPIP 노드인 경우 호스트 이름은 IPv4 또는 IPv6 주소로 해석됩니다.
- TCPIP4 노드인 경우 호스트 이름은 IPv4 주소로만 해석됩니다.
- TCPIP6 노드인 경우 호스트 이름은 IPv6 주소로만 해석됩니다.

#### **SVCENAME** *svcename*

TCP/IP 서비스 이름 또는 포트 번호를 지정합니다.

#### **WITH** *"comments"*

DB2 서버를 설명합니다. 네트워크 디렉토리에 등록된 서버를 설명하는 데도

음이 되는 모든 주석을 입력할 수 있습니다. 최대 길이는 30문자입니다. 캐리지 리턴 또는 줄 바꾸기 문자는 허용되지 않습니다. 주석 텍스트는 큰따옴표로 묶어야 합니다.

**USER** *username*

사용자의 LDAP 식별 이름(DN)을 지정합니다. LDAP 사용자 DN은 LDAP 디렉토리에 오브젝트를 작성 및 갱신하기 위한 충분한 권한이 있어야 합니다. 사용자의 LDAP DN이 지정되지 않은 경우 현재 로그인 사용자의 증명서가 사용됩니다.

**PASSWORD** *password*

어카운트 암호

---

## 제 150 장 UPDATE MONITOR SWITCHES

하나 이상의 데이터베이스 모니터 기록 스위치를 켜거나 끕니다. 데이터베이스 관리 프로그램이 시작할 때 6개 스위치의 설정은 `dft_mon` 데이터베이스 관리 프로그램 구성 매개변수에 의해 판별됩니다.

데이터베이스 모니터는 언제나 기본 정보 세트를 기록합니다. 이 기본 정보보다 많은 것이 필요한 사용자는 적합한 스위치를 켤 수 있지만, 그로 인해 시스템 성능이 저하됩니다. `GET SNAPSHOT` 명령의 출력에서 사용 가능한 정보의 양은 켜있는 스위치(있는 경우)를 반영합니다.

### 권한 부여

다음 중 하나가 필요합니다.

- `sysadm`
- `sysctrl`
- `sysmaint`
- `sysmon`

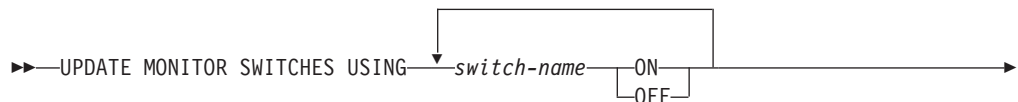
### 필수 연결

인스턴스 또는 데이터베이스:

- 인스턴스에 대한 접속이나 데이터베이스로의 연결이 없는 경우 기본 인스턴스 접속이 작성됩니다.
- 인스턴스에 대한 접속과 데이터베이스로의 연결이 둘 다 있는 경우 해당 인스턴스 접속이 작성됩니다.

리모트 인스턴스(또는 다른 로컬 인스턴스)에 대한 모니터 스위치를 갱신하려면 먼저 해당 인스턴스에 접속해야 합니다.

### 명령 구문



AT DBPARTITIONNUM *db-partition-number*  
GLOBAL

## 명령 매개변수

**USING** *switch-name*

다음 스위치 이름을 사용할 수 있습니다.

**BUFFERPOOL**

버퍼 풀 활동 정보

**LOCK**

잠금 정보

**SORT** 정렬 정보

**STATEMENT**

SQL문 정보

**TABLE**

테이블 활동 정보

**TIMESTAMP**

모니터링 시간소인 정보

**UOW** 작업 단위(UOW) 정보.

**AT DBPARTITIONNUM** *db-partition-number*

모니터 스위치의 상태가 표시될 데이터베이스 파티션을 지정합니다.

**GLOBAL**

파티션된 데이터베이스 환경의 모든 데이터베이스 파티션에 대한 집계 결과를 리턴합니다.

## 사용 시 참고사항

정보는 스위치가 켜진 후에만 데이터베이스 관리 프로그램에 의해 수집됩니다. 스위치는 db2stop이 발행되거나 UPDATE MONITOR SWITCHES 명령을 발행한 응용프로그램이 종료할 때까지 설정된 채로 있습니다. 특정 스위치와 관련된 정보를 지우려면 스위치를 해제한 후 설정하십시오.

한 응용프로그램에서 스위치를 갱신해도 다른 응용프로그램에는 영향을 주지 않습니다.

스위치 설정을 보려면 GET MONITOR SWITCHES 명령을 사용하십시오.

## 호환성

버전 8 이전 버전과의 호환성:

- DBPARTITIONNUM에 키워드 NODE를 대체할 수 있습니다.





---

## 제 151 장 UPDATE XMLSCHEMA

XML 스키마 저장소(XSR)의 다른 스키마로 특정 XML 스키마를 갱신합니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *dbadm*
- 카탈로그 뷰 SYSCAT.XSROBJECTS 및 SYSCAT.XSROBJECTCOMPONENTS의 SELECT 특권 및 다음 특권 세트 중 하나.
  - 갱신될 XML 스키마에 대한 ALTERIN 특권과 새 XML 스키마에 대한 DROPIN 특권(DROP NEW SCHEMA 옵션이 지정되지 않은 경우).
  - *xmlschema1*로 지정된 XML 스키마의 OWNER.

### 필수 연결

데이터베이스

### 명령 구문

```
▶▶ UPDATE XMLSCHEMA xmlschema1 WITH xmlschema2 [DROP NEW SCHEMA] ▶▶
```

### 명령 매개변수

**UPDATE XMLSCHEMA** *xmlschema1*

갱신될 원래 XML 스키마의 SQL ID를 지정합니다.

**WITH** *xmlschema2*

원래 XML 스키마를 갱신하는 데 사용된 새 XML 스키마의 SQL ID를 지정합니다.

**DROP NEW SCHEMA**

새 XML 스키마가 원래 XML 스키마를 갱신하는 데 사용된 후 삭제되어야 함을 나타냅니다.

**예 :**

```
UPDATE XMLSCHEMA JOHNDOE.OLDPROD  
WITH JOHNDOE.NEWPROD  
DROP NEW SCHEMA
```

XML 스키마 JOHNDOE.OLDPROD의 콘텐츠는 JOHNDOE.NEWPROD의 콘텐츠로 갱신되고 XML 스키마 JOHNDOE.NEWPROD가 삭제됩니다.

## 사용 시 참고사항

- 원래 XML 스키마와 새 XML 스키마는 호환되어야 합니다. 호환성 요구사항에 대한 자세한 내용은 『XML 스키마 전개를 위한 호환성 요구사항』을 참조하십시오.
- XML 스키마를 갱신하려면 원래 스키마와 새 스키마가 모두 XML 스키마 저장소(XSR)에 등록되어야 합니다.

---

## 제 5 부 시스템 명령어



---

## 제 152 장 dasauto - DB2 Administration Server 자동 시작

DB2 Administration Server 자동 시작을 사용 가능 또는 불가능하게 합니다.

**중요사항:** 이 명령은 제어 센터와 관련된 구성요소가 버전 9.7에서 사용되지 않기 때문에 사용되지 않으며 이후 릴리스에서 제거됩니다. 자세한 정보는 *버전 9.7의 새로운 내용* 책에 있는 『제어 센터 도구 및 DB2 Administration Server(DAS)는 사용되지 않음』 주제를 참조하십시오.

이 명령은 Linux 및 UNIX 시스템에서만 사용할 수 있습니다. 이 명령은 DB2DIR/das/adm 디렉토리에 있습니다. 여기서 DB2DIR은 DB2 데이터베이스 제품의 현재 버전이 설치된 위치입니다.

### 권한 부여

*dasadm*

### 필수 연결

없음

### 명령 구문

```
dasauto [-h] [-?] [-on] [-off]
```

### 명령 매개변수

#### **-h | -?**

도움말 정보를 표시합니다. 이 옵션이 지정되면 다른 모든 옵션은 무시되고 도움말 정보만 표시됩니다.

**-on** DB2 Administration Server 자동 시작을 사용 가능하게 합니다. 다음에 시스템이 재시작되면 DB2 Administration Server가 자동으로 시작됩니다.

**-off** DB2 Administration Server 자동 시작을 사용 불가능하게 합니다. 다음에 시스템이 재시작되면 DB2 Administration Server가 자동으로 시작되지 않습니다.



---

## 제 153 장 dascrt - DB2 Administration Server 작성

DAS(DB2 Administration Server)는 제어 센터 및 구성 지원 프로그램과 같은 DB2 도구에 대한 지원 서비스를 제공합니다. 시스템에 DAS가 없는 경우 이 명령을 사용하여 수동으로 생성할 수 있습니다.

**중요사항:** 이 명령은 제어 센터와 관련된 구성요소가 버전 9.7에서 사용되지 않기 때문에 사용되지 않으며 이후 릴리스에서 제거됩니다. 자세한 정보는 *버전 9.7의 새로운 내용* 책에 있는 『제어 센터 도구 및 DB2 Administration Server(DAS)는 사용되지 않음』 주제를 참조하십시오.

dascrt 명령은 DB2DIR/instance 디렉토리에 있습니다. 여기서 DB2DIR은 DB2 데이터베이스 제품의 현재 버전이 설치된 위치입니다.

이 명령은 Linux 및 UNIX 기반 시스템에서만 사용할 수 있습니다. Windows 운영 체제의 경우 동일한 용도로 db2admin create 명령을 사용할 수 있습니다.

### 권한 부여

Root 권한

### 필수 연결

없음

### 명령 구문

```
▶▶ dascrt -u DASuser [-d] ▶▶
```

### 명령 매개변수

**-u** *DASuser*

*DASuser*는 DAS가 작성될 사용자 ID입니다. DAS는 /home/*DASuser*/das 디렉토리에 작성됩니다.

다음과 같은 제한사항이 적용됩니다.

- 기존 ID를 사용하여 DB2 DAS를 작성하는 경우 ID가 잡기지 않고 만기된 암호가 없는지 확인하십시오.

**-d** DB2 서비스에 사용할 디버그 모드를 입력합니다.

## 사용 시 참고사항

- AIX 6.1(또는 그 이상)에서 시스템 워크로드 파티션(WPAR) 전역 환경의 공유 DB2 사본으로부터 이 명령을 실행할 때 이 명령은 루트 사용자로서 실행되어야 합니다.



---

## 제 154 장 dasdrop - DB2 Administration Server 제거

Linux 및 UNIX 기반 시스템에서만 DAS(DB2 Administration Server)를 제거합니다. Administration Server는 제어 센터 및 구성 지원 프로그램과 같은 DB2 도구에 대한 지원 서비스를 제공합니다. Windows 운영 체제의 경우 동일한 용도로 db2admin drop 명령을 사용할 수 있습니다.

**중요사항:** 이 명령은 제어 센터와 관련된 구성요소가 버전 9.7에서 사용되지 않기 때문에 사용되지 않으며 이후 릴리스에서 제거됩니다. 자세한 정보는 *버전 9.7의 새로운 내용 책*에 있는 『제어 센터 도구 및 DB2 Administration Server(DAS)는 사용되지 않음』 주제를 참조하십시오.

### 권한 부여

Root 권한

### 필수 연결

없음

### 명령 구문

```
▶▶—dasdrop —————▶▶  
└─d─┘
```

### 명령 매개변수

**-d** DB2 서비스에 사용할 디버그 모드를 입력합니다.

### 사용 시 참고사항

- dasdrop 명령은 DB2DIR/instance 디렉토리에 있습니다. 여기서 DB2DIR은 DB2 데이터베이스 제품의 현재 버전이 설치된 위치입니다.
- AIX 6.1(또는 그 이상)에서 시스템 워크로드 파티션(WPAR) 전역 환경의 공유 DB2 사본으로부터 이 명령을 실행할 때 이 명령은 루트 사용자로서 실행되어야 합니다.



---

## 제 155 장 dasmigr - DB2 Administration Server 이주

시스템의 DAS(DB2 Administration Server)를 DB2 데이터베이스 시스템의 이전 버전(DB2 데이터베이스 시스템의 현재 버전으로의 이주 지원)에서 DB2 데이터베이스 시스템으로 dasmigr이 실행된 경로와 관련된 DB2 데이터베이스 레벨로 이주합니다.

**중요사항:** 이 명령은 제어 센터와 관련된 구성요소가 버전 9.7에서 사용되지 않기 때문에 사용되지 않으며 이후 릴리스에서 제거됩니다. 자세한 정보는 *버전 9.7의 새로운 내용* 책에 있는 『제어 센터 도구 및 DB2 Administration Server(DAS)는 사용되지 않음』 주제를 참조하십시오.

하나의 DB2 데이터베이스 시스템 설치 위치에서 동일한 DB2 데이터베이스 시스템 버전 내 다른 위치로 DAS를 이동하려면 dasupdt 명령을 사용해야 합니다. DB2 데이터베이스 시스템의 이전 버전의 DAS를 DB2 데이터베이스 시스템의 현재 버전의 인스턴스를 관리하는 데 사용할 수 없습니다.

Linux 및 UNIX 시스템의 경우 이 유틸리티는 DB2DIR/instance 디렉토리에 있습니다. Windows 운영 체제의 경우에는 DB2DIR\bin 디렉토리에 있습니다. DB2DIR은 DB2 데이터베이스 시스템의 현재 버전이 설치된 설치 위치를 나타냅니다.

### 권한 부여

Root 액세스(UNIX 운영 체제) 또는 로컬 관리자 권한(Windows 운영 체제)

### 필수 연결

없음

### 명령 구문

#### Linux 및 UNIX 시스템용

```
▶▶ dasmigr [-d] _____▶▶
```

#### Windows 운영 체제용

```
▶▶ dasmigr [-h] [-p path override] _____▶▶
```

### 명령 매개변수

Linux 및 UNIX 운영 체제용

**-d** DB2 서비스에 사용할 디버그 모드를 입력합니다.

Windows 운영 체제용

**-h** 사용법 정보를 표시합니다.

**-p** *path override*

DAS 프로파일도 이동해야 함을 나타냅니다. *path override*는 디폴트 DAS 프로파일 경로 대신 사용될 사용자 지정 경로입니다.

**예:**

Linux 및 UNIX 시스템용

```
DB2DIR/instance/dasmigr
```

Windows 운영 체제의 경우

```
DB2DIR\bin\dasmigr
```

## 사용 시 참고사항

- AIX 6.1(또는 그 이상)에서 시스템 워크로드 파티션(WPAR) 전역 환경의 공유 DB2 사본으로부터 이 명령을 실행할 때 이 명령은 루트 사용자로서 실행되어야 합니다.

---

## 제 156 장 dasupdt - DAS 갱신

Linux 및 UNIX 시스템의 경우 관련 DB2 데이터베이스 시스템 설치가 갱신되면 이 명령이 DAS(DB2 Administration Server)를 갱신합니다.

**중요사항:** 이 명령은 제어 센터와 관련된 구성요소가 버전 9.7에서 사용되지 않기 때문에 사용되지 않으며 이후 릴리스에서 제거됩니다. 자세한 정보는 *버전 9.7의 새로운 내용* 책에 있는 『제어 센터 도구 및 DB2 Administration Server(DAS)는 사용되지 않음』 주제를 참조하십시오.

이 유틸리티는 DB2DIR/instance 디렉토리에 있습니다. 여기서 DB2DIR은 DB2 데이터베이스 제품의 현재 버전이 설치된 위치입니다. 두 위치가 모두 DB2 데이터베이스 시스템의 동일한 버전에 있는 경우 이 유틸리티를 사용하여 특정 설치 위치에서 다른 설치 위치로 DAS를 이동할 수도 있습니다.

시스템의 DAS가 installFixPack으로 갱신된 DB2 설치 경로와 관련되는 경우 FixPack을 설치하면 dasupdt 명령이 자동으로 실행됩니다.

Windows 운영 체제의 경우 이 명령은 특정 DB2 사본의 DAS를 동일한 DB2 데이터베이스 버전 내 다른 DAS로 갱신합니다. 이전 버전에서 DAS를 이주하려면 dasmgr 명령을 사용합니다. dasupdt를 사용하는 경우 DAS는 dasupdt 명령이 실행된 DB2 사본으로 갱신됩니다. 이 유틸리티는 DB2IPATH\bin 디렉토리에 있습니다. 여기서 DB2IPATH는 DB2 데이터베이스 제품의 현재 버전이 설치된 위치입니다.

### 권한 부여

Root 액세스(Linux 및 UNIX 시스템) 또는 로컬 관리자 권한(Windows 운영 체제)

### 필수 연결

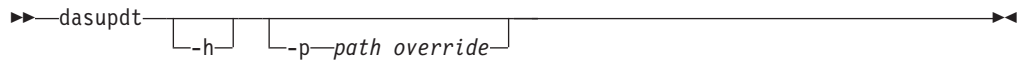
없음

### 명령 구문

Linux 및 UNIX 시스템용

```
►► dasupdt [-d] [-D] [-h] [-?]
```

Windows 운영 체제용



## 명령 매개변수

### Linux 및 UNIX 시스템용

- d 문제점 분석에 사용되는 디버그 모드를 설정합니다.
- D 특정 경로의 상위 코드 레벨에서 다른 경로에 설치된 하위 코드 레벨로 DAS를 이동합니다.

#### -h | -?

사용법 정보를 표시합니다.

### Windows 운영 체제용

-h 사용법 정보를 표시합니다.

#### -p *path override*

DAS 프로파일도 이동해야 함을 나타냅니다. *path override*는 디폴트 DAS 프로파일 경로 대신 사용될 사용자 지정 경로입니다.

### 예:

DAS가 특정 DB2 설치 경로에서 실행되고 하위 레벨의 다른 설치 경로로 DAS를 이동하려는 경우(그러나 두 설치 경로는 DB2 데이터베이스 시스템의 동일한 버전에 있음) 하위 레벨의 설치 경로에서 다음 명령을 실행합니다.

```
dasupdt -D
```

### 사용 시 참고사항

- AIX 6.1(또는 그 이상)에서 시스템 워크로드 파티션(WPAR) 전역 환경의 공유 DB2 사본으로부터 이 명령을 실행할 때 이 명령은 루트 사용자로서 실행되어야 합니다.

## 제 157 장 db2\_deinstall - DB2 제품, 기능 또는 언어 설치 제거

설치된 모든 DB2 제품, 기능 또는 언어를 설치 제거합니다.

- db2\_deinstall이 특정 DB2 설치 경로에서 실행되는 경우 동일한 경로에서 모두 또는 특정 기능이나 언어를 설치 제거할 수 있습니다.
- db2\_deinstall이 DB2 미디어에서 실행되는 경우 -b 옵션을 사용하여 경로를 지정해야 합니다. 그런 다음 해당 설치 경로에서 모두 또는 특정 기능이나 언어를 설치 제거할 수 있습니다.

이 옵션은 Linux 및 UNIX 시스템에서만 사용할 수 있습니다.

db2\_deinstall 명령은 DB2DIR/install에 있습니다. 여기서 DB2DIR은 DB2 데이터베이스 제품의 현재 버전이 설치된 위치입니다. db2\_deinstall 명령은 DB2 미디어에서만 사용할 수 있습니다. db2\_deinstall 명령은 설치 경로와 관련된 DB2 제품을 설치 제거하는 데만 사용할 수 있습니다.

### 권한 부여

루트 설치에는 루트 권한이 필요합니다. 비루트 설치의 경우 비루트 설치를 소유하는 사용자 ID로 로그온해야 합니다.

### 필요한 연결

없음

### 명령 구문

```
db2_deinstall -F feature-name [-a] [-r response-file] [-f sqllib]
[-b installpath] [-l log-file] [-t trace-file] [-h] [-?]
```

### 명령 매개변수

**-F feature-name**

하나의 기능 제거를 지정합니다. 여러 기능 설치 제거를 표시하려면 이 매개변수를 여러 번 지정합니다(예: -F feature1 -F feature2).

한 가지 경우를 제외하고는 -a와 함께 사용할 수 없습니다. 제거될 기능이 IBM TSAMP(Tivoli System Automation for Multiplatforms)이고 루트 권한이 있는 경우 -F TSAMP와 -a를 조합으로 사용하여 TSAMP와 DB2를 함께 제거할 수 있습니다.

한 가지 경우를 제외하고 -r과 함께 사용할 수 있습니다. 제거될 기능이 IBM TSAMP(Tivoli System Automation for Multiplatforms)인 경우 -F TSAMP와 -r을 조합으로 사용할 수 없습니다.

DB2 설치 제거 프로그램은 일부 DB2 기능을 제거한 후 관련 DB2 인스턴스를 자동으로 갱신합니다. 로그 파일에 보고된 대로 인스턴스 갱신에 실패하는 경우 db2iupdt(루트 인스턴스) 또는 db2nrupdt(루트 서버가 아닌 인스턴스) 명령으로 관련 DB2 인스턴스를 수동으로 갱신해야 합니다.

#### **-f sqllib**

이 옵션은 루트 서버가 아닌 서버 설치에만 유효합니다. -a와 함께 사용되면 인스턴스 맨 위 디렉토리와 모든 하위 항목이 제거됩니다.

**-a** 현재 위치에서 설치된 모든 DB2 제품을 제거합니다. 한 가지 경우를 제외하고는 -F와 함께 사용할 수 없습니다. 제거될 기능이 IBM TSAMP(Tivoli System Automation for Multiplatforms)이고 루트 권한이 있는 경우 -F TSAMP와 -a를 조합으로 사용하여 TSAMP와 DB2를 함께 제거할 수 있습니다.

-r 매개변수와 함께 사용할 수 없습니다.

루트 서버가 아닌 서버 설치의 경우 -a와 -f sqllib를 함께 사용하면 루트 서버가 아닌 인스턴스가 제거되고 \$HOME/sqllib 디렉토리도 제거됩니다.

#### **-r response-file**

응답 파일에 지정된 내용을 기반으로 제품, 기능 또는 언어의 설치 제거를 수행합니다(예: db2\_deinstall -r db2un.rsp).

-a 매개변수와 결합할 수 없습니다.

한 가지 경우를 제외하고 -F 매개변수와 함께 사용할 수 있습니다. 제거될 기능이 IBM TSAMP(Tivoli System Automation for Multiplatforms)인 경우 -F TSAMP와 -r을 조합으로 사용할 수 없습니다.

-r 및 -F 매개변수가 모두 지정되는 경우 -F 매개변수에 지정된 DB2 기능이 응답 파일의 REMOVE\_COMP 키워드를 겹쳐줍니다.

DB2 설치 제거 프로그램은 일부 DB2 기능을 제거한 후 관련 DB2 인스턴스를 자동으로 갱신합니다. 로그 파일에 보고된 대로 인스턴스 갱신에 실패하는 경우 db2iupdt(루트 인스턴스) 또는 db2nrupdt(루트 서버가 아닌 인스턴스) 명령으로 관련 DB2 인스턴스를 수동으로 갱신해야 합니다.

**-b** 이 옵션은 DB2 미디어에서 명령이 실행되는 경우 유효합니다. DB2 제품이 설치되었으며 설치 제거될 절대 경로를 지정합니다. 이 옵션이 지정되지 않으면 명령에서 경로에 대한 프롬프트가 표시됩니다.

#### **-l log-file**

로그 파일을 지정합니다. 루트 설치의 경우 디폴트 로그 파일은 /tmp/db2\_deinstall.log\$\$입니다. 여기서 \$\$는 프로세스 ID를 나타냅니다.



비루트 설치의 경우 디폴트 로그 파일은 /tmp/db2\_deinstall\_userID.log 입니다. 여기서 *userID*는 비루트 설치를 소유하는 사용자 ID를 나타냅니다. 제거될 기능이 IBM TSAMP(Tivoli System Automation for Multiplatforms)인 경우 SA MP의 설치 로그 파일은 DB2 로그 파일과 동일한 디렉토리에 있습니다.

**-t trace-file**

디버그 모드를 활성화합니다. 디버그 정보는 *trace-file*로 지정된 파일 이름에 작성됩니다.

**-h | -?**

도움말 정보를 표시합니다.

**예:**

- 특정 위치(DB2DIR)에 설치된 DB2 데이터베이스 제품을 모두 설치 제거하려면 DB2DIR/install 디렉토리에 있는 db2\_deinstall 명령을 실행하십시오.

```
DB2DIR/install/db2_deinstall -a
```

### 사용 시 참고사항

- db2\_deinstall -a -f sqllib를 실행하면 \$HOME/sqllib 디렉토리가 제거됩니다. 명령을 실행하기 전에 이 디렉토리에서 저장해야 할 모든 파일을 백업해야 합니다.
- 설치된 DB2 텍스트 검색이 있고 db2\_deinstall -f를 실행하는 경우, 텍스트 검색이 DB2 사본과 관련된 인스턴스에서 실행 중이면, 먼저 텍스트 검색 인스턴스 서비스를 중지해야 함을 표시하는 오류 메시지를 수신하게 됩니다. 텍스트 검색 인스턴스 서비스를 중지한 다음 명령을 재실행하십시오.



---

## 제 158 장 db2\_install - DB2 제품 설치

DB2 제품의 모든 기능을 지정된 경로에 설치합니다. 이 명령은 Linux 및 UNIX 시스템에서만 사용할 수 있습니다.

### 권한 부여

루트 설치에는 루트 권한이 필요합니다. 비루트 설치의 경우 비루트 설치를 소유하는 사용자 ID로 로그인해야 합니다.

### 필요한 연결

없음

### 명령 구문

```
db2_install -b install-path -f NOTSAMP -f nobackup  
-f ignoreType1 -p productShortName -c NLPACK_location  
-n -m -L language -l log-file -t trace-file  
-h  
-?
```

### 명령 매개변수

#### -b *install-path*

DB2 제품이 설치될 경로를 지정합니다. *install-path*는 전체 경로 이름이어야 하며 최대 경로는 128자로 제한됩니다. 이 매개변수는 *-n* 매개변수가 지정된 경우 필수 매개변수입니다.

*-b* 옵션은 DB2 제품의 루트 서버가 아닌 서버 설치에는 필요하지 않지만 *-n* 옵션을 사용하는 경우 루트 설치에는 계속 필요합니다. 루트 서버가 아닌 서버 설치에서 *-b*를 사용하는 경우 경로가 사용자의 *\$HOME/sqllib*로 유효해야 합니다. 루트 및 루트 서버가 아닌 서버 설치 모두 절대 설치 경로의 길이는 128 바이트로 제한됩니다.

#### -f NOTSAMP

SA MP가 설치 또는 갱신되지 않아야 함을 지정합니다.

#### -f *nobackup*

루트 서버가 아닌 서버 업그레이드에만 적용됩니다. 구성요소가 갱신될 때

db2\_install이 설치 파일을 백업하지 않도록 강제 실행합니다. 파일을 백업하지 않는 것으로 선택하면 설치 디렉토리의 공간 요구사항이 감소합니다. 그러나 파일을 백업하지 않는 것으로 선택하는 경우 오류 발생 시 DB2 설치 프로그램이 롤백 조작을 수행할 수 없음을 의미하기도 합니다. 이러한 경우 파일을 수동으로 제거하고 제품을 재설치해야 합니다.

**-f ignoreType1**

루트 서버가 아닌 서버 업그레이드에만 적용됩니다. 데이터베이스 상태를 검사할 때 db2\_install이 유형 1 인덱스를 무시하도록 강제 실행합니다.

**-p productShortName**

설치될 DB2 제품을 지정합니다. 이 매개변수는 대소문자를 구분하지 않으며 -n 매개변수가 지정된 경우 필수 매개변수입니다. 제품 단축 이름 (*productShortName*)은 미디어의 db2/*plat* 서브디렉토리에 있는 ComponentList.htm 파일(제품 전체 이름 아래)에 있습니다. 여기서 *plat*은 설치하는 플랫폼 이름입니다. 한 번에 하나의 제품만 설치할 수 있습니다.

**-c NLPACK\_location**

관련된 DB2 자국어 팩(NLPACK)의 절대 경로 위치를 지정합니다. 이 매개변수는 -n이 지정된 경우 필수 매개변수입니다. 다음 조건이 모두 충족되는 경우 DB2 NLPACK 위치가 명시적으로 제공되어야 합니다.

- -n 옵션이 지정되었습니다.
- 설치 시 자국어(영어 이외) 지원이 필요합니다.
- DB2 NLPACK이 DB2 제품 이미지에도 DB2 제품 이미지와 동일한 서브디렉토리에 없습니다.

**-n** 비대화형 모드를 지정합니다. 이 옵션이 지정되면 -b, -p 및/또는 -c 또한 지정해야 합니다.

**-m** 이 옵션은 비루트 설치에만 적용됩니다. 비루트 사본의 업그레이드를 지정합니다. 업그레이드 시 현재 경로의 모든 기존 DB2 제품이 제거됩니다. 업그레이드하면 지정된 제품만 설치됩니다. 업그레이드 후 다른 DB2 제품을 별도로 설치해야 합니다.

**-L language**

자국어 지원(NLS)을 지정합니다. DB2 제품의 영어 이외 버전을 설치할 수 있습니다. 그러나 이 명령은 자국어 팩 CD가 아닌 제품 CD에서 실행해야 합니다. 영어는 디폴트로 항상 설치되므로 영어는 지정하지 않아도 됩니다. 여러 언어가 필요한 경우 이 매개변수는 필수 매개변수입니다. 여러 언어를 표시하려면 이 매개변수를 여러 번 지정합니다. 예를 들어, 프랑스어와 독일어를 모두 설치하려면 -L FR -L DE를 지정합니다. 이 매개변수는 대소문자를 구분하지 않습니다.

### **-l log-file**

로그 파일을 지정합니다. 루트 설치의 경우 디폴트 로그 파일은 /tmp/db2\_install.log\$\$입니다. 여기서 \$\$는 프로세스 ID를 나타냅니다. 비루트 설치의 경우 디폴트 로그 파일은 /tmp/db2\_install\_userID.log입니다. 여기서 userID는 비루트 설치를 소유하는 사용자 ID를 나타냅니다. IBM Tivoli SA MP(System Automation for Multiplatforms)가 db2\_install 명령으로 설치 또는 갱신되는 경우 해당 로그 파일은 DB2 로그 파일과 동일한 디렉토리에 저장됩니다.

### **-t trace-file**

디버그 모드를 활성화합니다. 디버그 정보는 trace-file로 지정된 파일 이름에 작성됩니다.

### **-h | -?**

사용법 정보를 표시합니다.

### **예:**

- /mnt/cdrom에서 이미지를 설치하고 필요한 모든 입력에 대한 프롬프트를 표시하거나 /mnt/cdrom의 이미지에서 DB2 Enterprise Server Edition를 설치하려면 다음을 실행하십시오.

```
cd /mnt/cdrom
./db2_install
```

- DB2 Enterprise Server Edition를 /mnt/cdrom의 이미지에서 비대화식 영어로 /db2/newlevel에 설치하려면 다음을 실행하십시오.

```
cd /mnt/cdrom
./db2_install -p ese -b /db2/newlevel -n
```

## **사용 시 참고사항**

루트 서버가 아닌 DB2 설치의 디폴트 로그 및 추적 파일 이름에는 사용자 ID가 포함되며 DB2의 루트 설치에 따른 파일을 겹쳐쓰지 않습니다.

루트 서버가 아닌 서버 설치를 위한 디폴트 파일 이름은 다음과 같습니다(userID는 루트 서버가 아닌 서버 설치를 수행하는 루트 서버가 아닌 서버 사용자의 사용자 이름임).

- /tmp/db2\_install\_userID.log
- /tmp/db2\_install\_userID.err
- /tmp/db2\_install\_userID.trc

루트 서버가 아닌 서버 설치가 완료되면 로그 파일이 \$DB2DIR/install/logs 디렉토리에 복사됩니다. 이 디렉토리는 루트 설치의 경우와 동일한 디렉토리입니다.

추가 정보는 -l logfile 명령 매개변수를 참조하십시오.



---

## 제 159 장 db2\_local\_ps - Linux/UNIX의 DB2 프로세스 상태

Linux 및 UNIX 시스템의 경우 인스턴스에서 실행되는 모든 DB2 프로세스는 db2\_local\_ps 명령을 사용하여 표시될 수 있습니다.

### 권한 부여

없음

### 필수 연결

없음

### 명령 구문

▶▶—db2\_local\_ps—————▶▶

### 명령 매개변수

#### db2\_local\_ps

인스턴스에서 실행되는 모든 DB2 프로세스를 출력합니다.

#### 예:

```
[db2inst1@bower1 ~]$ db2_local_ps
Node 0
  UID      PID      PPID     C    STIME   TTY      TIME CMD
db2inst1  3254    3253     0    14:04 pts/1    00:00:00 db2sysc 0
  root     3255    3254     0    14:04 pts/1    00:00:00 db2ckpwd 0
  root     3256    3254     0    14:04 pts/1    00:00:00 db2ckpwd 0
  root     3257    3254     0    14:04 pts/1    00:00:00 db2ckpwd 0
  root     3266    3254     0    14:04 pts/1    00:00:00 db2gds 0
db2inst1  3267    3254     0    14:04 pts/1    00:00:00 db2licc 0
db2inst1  3268    3254     0    14:04 pts/1    00:00:00 db2ipccm 0
db2inst1  3269    3254     0    14:04 pts/1    00:00:00 db2tcpcm 0
db2inst1  3271    3254     0    14:04 pts/1    00:00:00 db2resync 0
db2inst1  3273    3254     0    14:04 pts/1    00:00:00 db2acd ,0,0,0,1,0,0,897b50,...
db2inst1  3297    3266     0    14:04 pts/1    00:00:00 db2loggr (SAMPLE) 0
db2inst1  3299    3266     0    14:04 pts/1    00:00:00 db2loggw (SAMPLE) 0
db2inst1  3300    3266     0    14:04 pts/1    00:00:00 db2lfr (SAMPLE) 0
db2inst1  3301    3266     0    14:04 pts/1    00:00:00 db2dlock (SAMPLE) 0
db2inst1  3303    3266     0    14:04 pts/1    00:00:00 db2pclnr 0
db2inst1  3313    3266     0    14:05 pts/1    00:00:00 db2pfchr 0
db2inst1  3314    3266     0    14:05 pts/1    00:00:00 db2pfchr 0
db2inst1  3315    3266     0    14:05 pts/1    00:00:00 db2pfchr 0
db2inst1  3316    3266     0    14:05 pts/1    00:00:00 db2stmm (SAMPLE) 0
db2inst1  3317    3266     0    14:05 pts/1    00:00:00 db2taskd (TOOLSDB) 0
db2inst1  3318    3266     0    14:05 pts/1    00:00:00 db2taskd (SAMPLE) 0
db2inst1  3319    3266     0    14:05 pts/1    00:00:00 db2stmm (TOOLSDB) 0
db2inst1  3320    3266     0    14:05 pts/1    00:00:00 db2evmgi (DB2DETAILDEADLOCK) 0
```

```

db2inst1      3321      3266      0      14:05      pts/1 00:00:00 db2evmgi (DB2DETAILDEADLOCK) 0
db2inst1      3341      3266      0      14:05      pts/1 00:00:00 db2loggr (TOOLSDB) 0
db2inst1      3343      3266      0      14:05      pts/1 00:00:00 db2loggw (TOOLSDB) 0
db2inst1      3344      3266      0      14:05      pts/1 00:00:00 db21fr (TOOLSDB) 0
db2inst1      3345      3266      0      14:05      pts/1 00:00:00 db2dlock (TOOLSDB) 0
db2inst1      3346      3266      0      14:05      pts/1 00:00:00 db2pclnr 0
db2inst1      3347      3266      0      14:05      pts/1 00:00:00 db2pfchr 0
db2inst1      3348      3266      0      14:05      pts/1 00:00:00 db2pfchr 0
db2inst1      3349      3266      0      14:05      pts/1 00:00:00 db2pfchr 0
db2inst1      3270      3268      2      14:04      pts/1 00:00:01 db2agent (TOOLSDB) 0
db2inst1      3285      3268      0      14:04      pts/1 00:00:00 db2agent (SAMPLE) 0

```

Node 1 ...

## 사용 시 참고사항

인스턴스가 중지되면 프로세스가 표시되지 않습니다. 프로세스가 나열되지 않은 경우 db2start 명령을 실행하십시오.



---

## 제 160 장 db2addicons - DB2 도구 명령의 주메뉴 항목 작성

DB2 도구의 주메뉴 항목을 작성합니다.

Linux 운영 체제에서 db2addicons 명령은 현재 사용자의 DB2 도구에 대한 주메뉴 항목을 작성합니다. db2addicons 명령을 수동으로 실행하여 DB2 도구의 주메뉴 항목이 작성됩니다. DB2 인스턴스 소유자의 경우, 메뉴 항목은 DB2 인스턴스가 작성, 갱신 또는 업그레이드되었을 때 인스턴스 유틸리티에 의하여 자동으로 작성됩니다. 주메뉴 항목이 다른 사용자의 데스크탑에 필요한 경우, db2addicons 명령은 해당 특정 사용자로 실행될 수 있으나, 명령을 실행하기 전에 먼저 해당 사용자의 환경에서 인스턴스 환경이 설정되어야 합니다.

### 권한 부여

없음

### 명령 구문

```
▶▶—db2addicons—┐└─h┘
```

### 명령 매개변수

**-h**      사용법 정보를 표시합니다.



## 제 161 장 db2admin - DB2 Administration Server

이 유틸리티는 DAS(DB2 Administration Server)를 관리하는 데 사용됩니다. 매개변수를 지정하지 않고 DAS가 존재하는 경우 이 명령은 DAS의 이름을 리턴합니다.

**중요사항:** 이 명령은 제어 센터와 관련된 구성요소가 버전 9.7에서 사용되지 않기 때문에 사용되지 않으며 이후 릴리스에서 제거됩니다. 자세한 정보는 *버전 9.7의 새로운 내용* 책에 있는 『제어 센터 도구 및 DB2 Administration Server(DAS)는 사용되지 않음』 주제를 참조하십시오.

Linux 및 UNIX 기반 시스템에서 db2admin 명령의 실행할 수 있는 파일은 DASHOME/das/bin 디렉토리에서 찾을 수 있으며, DASHOME은 DAS 사용자의 홈 디렉토리입니다. Windows 운영 체제에서 db2admin 실행 파일은 DB2PATH\bin 디렉토리에 있으며 DB2PATH는 DB2 사본이 설치된 위치입니다.

### 권한 부여

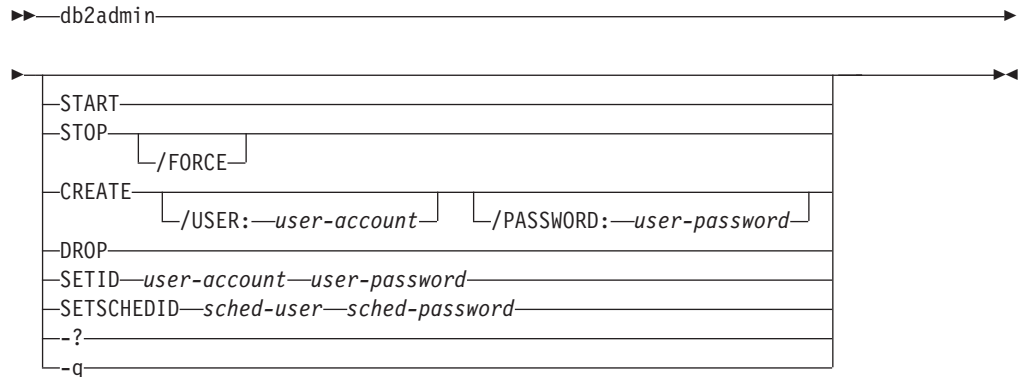
UNIX 운영 체제에서 *dasadm*이지만 64비트 인스턴스와 연관되지는 않습니다.

Windows 운영 체제의 로컬 관리자.

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

#### START

DAS를 시작합니다.

## **STOP /FORCE**

DAS를 중지합니다. force 옵션은 요청을 서비스하는 프로세스 중에 있는지 여부와 상관없이 DAS를 강제로 중지하는 데 사용됩니다.

## **CREATE /USER: user-account /PASSWORD: user-password**

DAS를 작성합니다. 사용자 이름과 암호를 지정하는 경우 DAS가 이 사용자 어카운트와 연관됩니다. 지정된 값이 유효하지 않은 경우 유틸리티는 인증 오류를 리턴합니다. 지정된 사용자 어카운트는 유효한 SQL ID여야 하며 보안 데이터베이스에 존재해야 합니다. 모든 DAS 기능에 액세스할 수 있도록 사용자 어카운트를 지정할 것을 권장합니다. UNIX 운영 체제에서 DAS를 작성하려면 dasprt 명령을 사용하십시오.

## **DROP**

DAS를 삭제합니다. UNIX 운영 체제에서 DAS를 삭제(drop)하려면 dasdrop 명령을 사용해야 합니다.

## **SETID user-account/user-password**

DAS와 연관된 사용자 어카운트를 설정 또는 수정합니다.

## **SETSCHEDID sched-user/sched-password**

스케줄러가 도구 카탈로그 데이터베이스에 연결하는 데 사용하는 로그인 어카운트를 설정합니다. 스케줄러가 사용 가능하고 도구 카탈로그 데이터베이스가 DAS에 리모트인 경우에만 필요합니다. 스케줄러에 대한 자세한 정보는 [관리 안내서](#)를 참조하십시오.

- ? 도움말 정보를 표시합니다. 이 옵션이 지정되면 다른 모든 옵션은 무시되고 도움말 정보만 표시됩니다.
- q 정숙 모드에서 db2admin 명령을 실행합니다. 명령이 실행될 때 메시지가 표시되지 않습니다. 이 옵션은 다른 모든 명령 옵션과 조합할 수 있습니다.

---

## 제 162 장 db2adutl - TSM에서 DB2 오브젝트 관리

사용자가 TSM(Tivoli Storage Manager)을 사용하여 저장되는 백업 이미지, 로그 및 로드 사본 이미지를 쿼리, 추출, 검증 및 삭제할 수 있게 합니다. 또한 사용자가 TSM 서버의 오브젝트에 대한 액세스 권한을 부여 및 취소할 수 있습니다.

UNIX 운영 체제에서 이 유틸리티는 sqllib/adsm 디렉토리에 있습니다. Windows 운영 체제에서는 sqllib#bin에 있습니다.

### 권한 부여

없음

### 필수 연결

없음

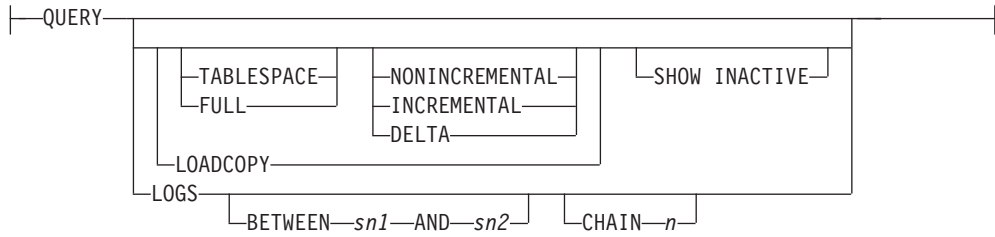
### 명령 구문

```
▶▶ db2adutl [db2-object-options] [access-control-options]
```

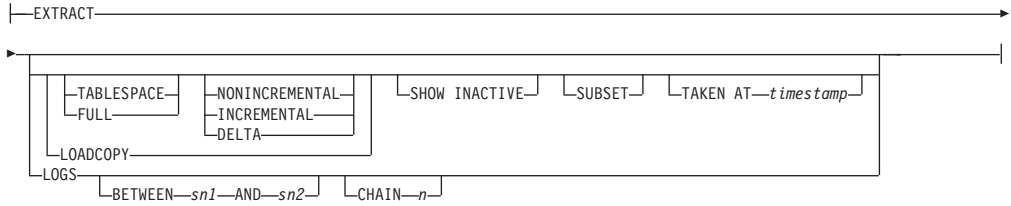
#### db2-object-options:

```
[QUERY-options] [EXTRACT-options] [DELETE-options] [VERIFY-options] [COMPRLIB—decompression-library]
[COMPROPTS—decompression-options] [VERBOSE]
[DATABASE—database_name] [DBPARTITIONNUM—db-partition-number]
[DB]
[PASSWORD—password] [NODENAME—node_name] [OWNER—owner]
[WITHOUT PROMPTING]
```

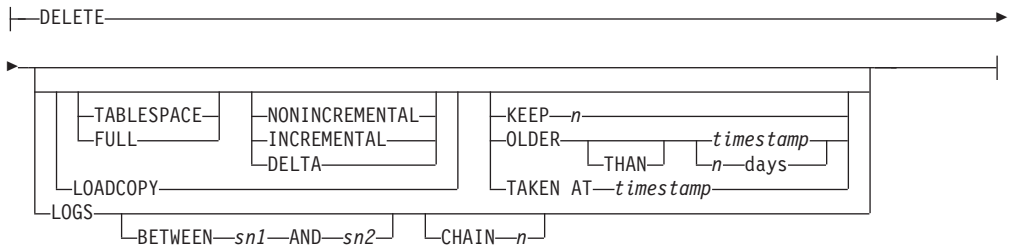
#### QUERY-options:



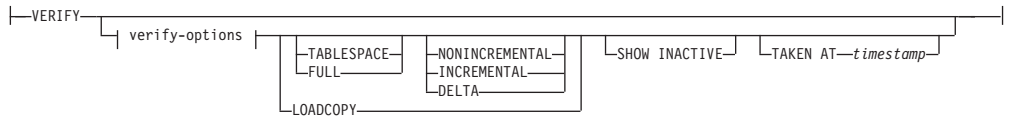
**EXTRACT-options:**



**DELETE-options:**



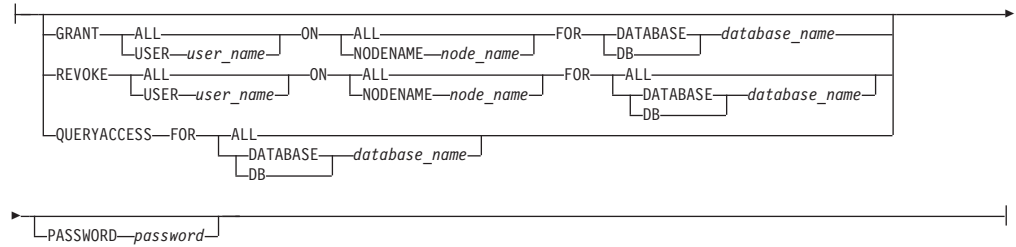
**VERIFY-options:**



**verify-options:**



## access-control-options:



## 명령 매개변수

### QUERY

TSM 서버에서 DB2 오브젝트를 쿼리합니다.

### EXTRACT

TSM 서버에서 로컬 시스템의 현재 디렉토리로 DB2 오브젝트를 복사합니다.

### DELETE

TSM 서버의 백업 오브젝트를 비활성화하거나 로그 아카이브를 삭제합니다.

### VERIFY

서버에 있는 백업 사본에 대해 일관성 검사를 수행합니다. 이 매개변수는 전체 백업 이미지가 네트워크를 통해 전송되게 합니다.

**ALL** 사용 가능한 모든 정보를 표시합니다.

### CHECK

체크비트 및 체크섬 결과를 표시합니다.

**DMS** DMS 테이블 스페이스 데이터 페이지의 헤더에 있는 정보를 표시합니다.

### HEADER

미디어 헤더 정보를 표시합니다.

### HEADERONLY

HEADER와 동일한 정보를 표시하지만 이미지의 시작부터 4K 미디어 헤더 정보만 읽습니다. 이미지 유효성을 확인하지 않습니다.

**LFH** 로그 파일 헤더(LFH) 데이터를 표시합니다.

### OBJECT

오브젝트 헤더의 세부사항 정보를 표시합니다.

### PAGECOUNT

이미지에서 발견한 각 오브젝트 유형의 페이지 수를 표시합니다.

**SGF** 이미지의 자동 스토리지 경로를 표시합니다.

### **SGFONLY**

이미지의 자동 스토리지 경로를 표시하지만 이미지 유효성을 확인하지는 않습니다.

### **TABLESPACES**

이미지의 테이블 스페이스에 대한 테이블 스페이스 세부사항을 컨테이너 정보와 함께 표시합니다.

### **TABLESPACESONLY**

TABLESPACES와 동일한 정보를 표시하지만 이미지 유효성을 확인하지 않습니다.

### **TABLESPACE**

테이블 스페이스 백업 이미지만 포함합니다.

**FULL** 전체 데이터베이스 백업 이미지만 포함합니다.

### **NONINCREMENTAL**

비증분식 백업 이미지만 포함합니다.

### **INCREMENTAL**

증분식 백업 이미지만 포함합니다.

### **DELTA**

증분 델타 백업 이미지만 포함합니다.

### **LOADCOPY**

로드 사본 이미지만 포함합니다.

### **LOGS**

로그 아카이브 이미지만 포함합니다.

### **BETWEEN *sn1* AND *sn2***

로그 시퀀스 번호 1과 로그 시퀀스 번호 2 사이의 로그가 사용되도록 지정합니다.

### **CHAIN *n***

사용될 로그의 체인 ID를 지정합니다.

### **SHOW INACTIVE**

비활성화된 백업 오브젝트를 포함합니다.

### **SUBSET**

이미지에서 파일로 페이지를 추출합니다. 페이지를 추출하려면 입력 및 출력 파일이 필요합니다. 디폴트 입력 파일은 `extractPage.in`입니다. `DB2LISTFILE` 환경 변수를 전체 경로로 설정하여 디폴트 입력 파일을 겹쳐쓸 수 있습니다. 입력 파일 형식은 다음과 같습니다.

SMS 테이블 스페이스의 경우:



S <tbSPID> <objID> <objType> <startPage> <numPages>

주:

1. <startPage>는 오브젝트와 관련되는 오브젝트 페이지 번호입니다.

DMS 테이블 스페이스의 경우:

D <tbSPID> <objType> <startPage> <numPages>

주:

1. <objType>은 DMS 로드 사본 이미지를 검증하는 경우에만 필요합니다.
2. <startPage>는 폴과 관련되는 오브젝트 페이지 번호입니다.

로그 파일의 경우:

L <log num> <startPos> <numPages>

기타 데이터의 경우(예: 초기 데이터):

O <objType> <startPos> <numBytes>

디폴트 출력 파일은 extractPage.out입니다. DB2EXTRACTFILE 환경 변수를 전체 경로로 설정하여 디폴트 출력 파일 이름을 겹쳐쓸 수 있습니다.

#### **TAKEN AT** *timestamp*

시간소인으로 백업 이미지를 지정합니다.

#### **KEEP** *n*

시간소인으로 가장 최근 *n*을 제외한 지정된 유형의 모든 오브젝트를 비활성화합니다.

#### **OLDER THAN** *timestamp* 또는 *n days*

*timestamp* 또는 *n*일보다 빠른 시간소인을 갖는 오브젝트가 비활성화되도록 지정합니다.

#### **COMPRLIB** *decompression-library*

압축 해제를 수행하는 데 사용될 라이브러리 이름을 표시합니다. 이 이름은 서버에 있는 파일을 나타내는 완전한 경로이어야 합니다. 이 매개변수를 지정하지 않으면 DB2는 이미지에 저장된 라이브러리를 사용하려고 합니다. 백업이 압축되지 않은 경우 이 매개변수 값은 무시됩니다. 지정된 라이브러리를 로드할 수 없는 경우 작업에 실패합니다.

#### **COMPROPTS** *decompression-options*

압축 해제 라이브러리의 초기화 루틴으로 전달될 2진 데이터 블록을 설명합니다. DB2는 클라이언트에서 서버로 직접 이 문자열을 전달하므로 모든 바이트 리버설 또는 코드 페이지 변환 문제는 압축 해제 라이브러리에서 처리되어야 합니다. 데이터 블록의 첫 번째 문자가 '@'일 경우, DB2는 이 데이터의 나머지 부분을 서버에 있는 파일 이름으로 해석합니다. 그런 다음 DB2는 데이터 블록

의 콘텐츠를 이 파일의 콘텐츠로 바꾼 후 대신 이 새 값을 초기화 루틴으로 전달합니다. 이 문자열의 최대 길이는 1024바이트입니다.

**DATABASE** *database\_name*

지정된 데이터베이스 이름과 연관된 오브젝트만 고려합니다.

**DBPARTITIONNUM** *db-partition-number*

지정된 데이터베이스 파티션 번호에 의해 작성되는 오브젝트만 고려합니다.

**PASSWORD** *password*

필요한 경우 이 노드에 대한 TSM 클라이언트 암호를 지정합니다. 데이터베이스가 지정되고 암호를 제공하지 않는 경우 *tsm\_password* 데이터베이스 구성 매개변수에 대해 지정된 값이 TSM으로 전달됩니다. 그렇지 않으면 암호가 사용되지 않습니다.

**NODENAME** *node\_name*

특정 TSM 노드 이름과 연관된 이미지만 고려합니다.

**OWNER** *owner*

지정된 소유자가 작성한 오브젝트만 고려합니다.

**WITHOUT PROMPTING**

오브젝트가 삭제되기 전에 사용자에게 검증이 프롬프트되지 않습니다.

**VERBOSE**

추가 파일 정보를 표시합니다.

**GRANT ALL | USER** *user\_name*

모든 사용자 또는 지정된 사용자에게 현재 TSM 노드의 TSM 파일에 대한 액세스 권한을 추가합니다. 사용자에게 액세스를 부여하면 지정된 데이터베이스와 관련된 모든 현재 및 미래 파일에 대한 액세스를 부여합니다.

**REVOKE ALL | USER** *user\_name*

모든 사용자 또는 지정된 사용자에서 현재 TSM 노드의 TSM 파일에 대한 액세스 권한을 제거합니다.

**QUERYACCESS**

현재 액세스 목록을 검색합니다. 사용자 및 TSM 노드 목록이 표시됩니다.

**ON ALL | NODENAME** *node\_name*

액세스 권한이 변경될 TSM 노드를 지정합니다.

**FOR ALL | DATABASE** *database\_name*

고려할 데이터베이스를 지정합니다.

**예:**

1. 다음은 db2 backup database rawsampl use tsm 명령의 출력 샘플입니다.

백업이 완료되었습니다. 이 백업 이미지에 대한 시간소인은 20031209184503입니다.

다음은 백업 작업 후에 발행되는 db2adutl query 명령의 출력 샘플입니다.

```
Query for database RAWSAMPL

Retrieving FULL DATABASE BACKUP information.
  1 Time: 20031209184403, Oldest log: S0000050.LOG, Sessions: 1

Retrieving INCREMENTAL DATABASE BACKUP information.
  No INCREMENTAL DATABASE BACKUP images found for RAWSAMPL

Retrieving DELTA DATABASE BACKUP information.
  No DELTA DATABASE BACKUP images found for RAWSAMPL

Retrieving TABLESPACE BACKUP information.
  No TABLESPACE BACKUP images found for RAWSAMPL

Retrieving INCREMENTAL TABLESPACE BACKUP information.
  No INCREMENTAL TABLESPACE BACKUP images found for RAWSAMPL

Retrieving DELTA TABLESPACE BACKUP information.
  No DELTA TABLESPACE BACKUP images found for RAWSAMPL

Retrieving LOCAL COPY information.
  No LOCAL COPY images found for RAWSAMPL

Retrieving log archive information.
  Log file: S0000050.LOG, Chain Num: 0, DB Partition Number: 0,
  Taken at 2003-12-09-18.46.13
  Log file: S0000051.LOG, Chain Num: 0, DB Partition Number: 0,
  Taken at 2003-12-09-18.46.43
  Log file: S0000052.LOG, Chain Num: 0, DB Partition Number: 0,
  Taken at 2003-12-09-18.47.12
  Log file: S0000053.LOG, Chain Num: 0, DB Partition Number: 0,
  Taken at 2003-12-09-18.50.14
  Log file: S0000054.LOG, Chain Num: 0, DB Partition Number: 0,
  Taken at 2003-12-09-18.50.56
  Log file: S0000055.LOG, Chain Num: 0, DB Partition Number: 0,
  Taken at 2003-12-09-18.52.39
```

2. 다음은 db2adutl delete full taken at 20031209184503 db rawsampl 명령의 출력 샘플입니다.

```
Query for database RAWSAMPL

Retrieving FULL DATABASE BACKUP information.
  Taken at: 20031209184503 DB Partition Number: 0 Sessions: 1

Do you want to delete this file (Y/N)? y

Are you sure (Y/N)? y

Retrieving INCREMENTAL DATABASE BACKUP information.
  No INCREMENTAL DATABASE BACKUP images found for RAWSAMPL

Retrieving DELTA DATABASE BACKUP information.
  No DELTA DATABASE BACKUP images found for RAWSAMPL
```

다음은 전체 백업 이미지를 삭제한 작업 후에 발행되는 db2adutl query 명령의 출력 샘플입니다. 백업 이미지의 시간소인을 주의하십시오.

```
Query for database RAWSAMPL

Retrieving FULL DATABASE BACKUP information.
```

1 Time: 20031209184403, Oldest log: S0000050.LOG, Sessions: 1

Retrieving INCREMENTAL DATABASE BACKUP information.  
No INCREMENTAL DATABASE BACKUP images found for RAWSAMPL

Retrieving DELTA DATABASE BACKUP information.  
No DELTA DATABASE BACKUP images found for RAWSAMPL

Retrieving TABLESPACE BACKUP information.  
No TABLESPACE BACKUP images found for RAWSAMPL

Retrieving INCREMENTAL TABLESPACE BACKUP information.  
No INCREMENTAL TABLESPACE BACKUP images found for RAWSAMPL

Retrieving DELTA TABLESPACE BACKUP information.  
No DELTA TABLESPACE BACKUP images found for RAWSAMPL

Retrieving LOCAL COPY information.  
No LOCAL COPY images found for RAWSAMPL

Retrieving log archive information.  
Log file: S0000050.LOG, Chain Num: 0, DB Partition Number: 0,  
Taken at 2003-12-09-18.46.13  
Log file: S0000051.LOG, Chain Num: 0, DB Partition Number: 0,  
Taken at 2003-12-09-18.46.43  
Log file: S0000052.LOG, Chain Num: 0, DB Partition Number: 0,  
Taken at 2003-12-09-18.47.12  
Log file: S0000053.LOG, Chain Num: 0, DB Partition Number: 0,  
Taken at 2003-12-09-18.50.14  
Log file: S0000054.LOG, Chain Num: 0, DB Partition Number: 0,  
Taken at 2003-12-09-18.50.56  
Log file: S0000055.LOG, Chain Num: 0, DB Partition Number: 0,  
Taken at 2003-12-09-18.52.39

3. 다음은 db2adutl queryaccess for all 명령의 출력 샘플입니다.

Node	User	Database Name	type
bar2	jchisan	sample	B
<all>	<all>	test	B

Access Types: B - Backup images L - Logs A - both

## 사용 시 참고사항

아래의 각 그룹의 한 매개변수를 사용하여 작업에 포함되는 백업 이미지 유형을 제한할 수 있습니다.

### 세분화도:

- FULL - 데이터베이스 백업 이미지만 포함합니다.
- TABLESPACE - 테이블 스페이스 백업 이미지만 포함합니다.

### 누적성:

- NONINCREMENTAL - 비증분식 백업 이미지만 포함합니다.
- INCREMENTAL - 증분식 백업 이미지만 포함합니다.

- DELTA - 증분 델타 백업 이미지만 포함합니다.

## 호환성

버전 8 이전 버전과의 호환성:

- DBPARTITIONNUM에 키워드 NODE를 대체할 수 있습니다.



## 제 163 장 db2advise - DB2 디자인 어드바이저

DB2 디자인 어드바이저는 구체화된 쿼리 테이블(MQT) 및 인덱스 작성, 테이블 재파티셔닝, 다차원적으로 클러스터된(MDC) 테이블로의 변환 및 사용되지 않는 오브젝트 삭제 시 사용자에게 조언합니다.

권장사항은 사용자가 제공하는 하나 이상의 SQL문을 기초로 합니다. 관련된 SQL문 그룹을 워크로드라고 합니다. 사용자는 워크로드에서 각 명령문의 중요성에 순위를 지정하고 워크로드의 각 명령문이 실행되는 빈도를 지정할 수 있습니다. 디자인 어드바이저는 권장 오브젝트를 작성하기 위한 CREATE INDEX, CREATE SUMMARY TABLE(MQT) 및 CREATE TABLE 문을 포함하는 DDL CLP 스크립트를 출력합니다.

이 명령이 실행될 때 구조화된 유형 컬럼은 고려되지 않습니다.

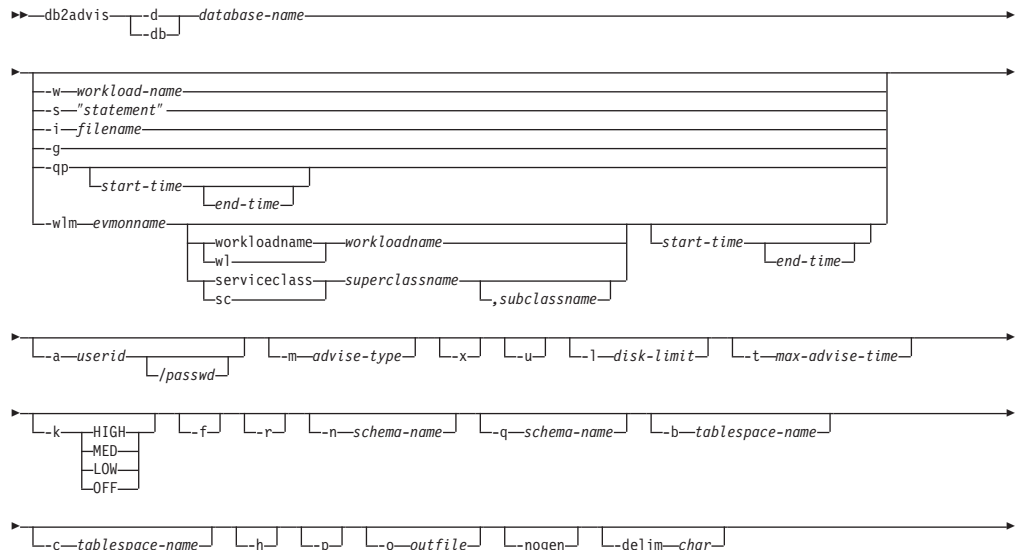
### 권한 부여

데이터베이스에 대한 읽기 액세스. Explain 테이블에 대한 읽기 및 쓰기 액세스. 구체화된 쿼리 테이블(MQT)이 사용되는 경우, CREATE TABLE 권한과 MQT에 대한 읽기 및 쓰기 액세스를 가지고 있어야 합니다.

### 필수 연결

없음. 이 명령은 데이터베이스 연결을 설정합니다.

### 명령 구문



## 명령 매개변수

### -a *userid/passwd*

데이터베이스에 연결하는 데 사용되는 이름 및 암호. 암호를 지정하는 경우 슬래시(/)가 포함되어야 합니다. **-x** 옵션이 지정되는 경우, 암호는 지정하면 안 됩니다.

### -b *tablespace-name*

새 MQT가 작성될 테이블 스페이스의 이름을 지정합니다. 지정하지 않는 경우, 어드바이저는 존재하는 테이블 스페이스 세트에서 테이블 스페이스를 선택합니다.

### -c *tablespace-name*

시뮬레이션 카탈로그 테이블을 작성할 테이블 스페이스의 이름을 지정합니다(테이블 스페이스는 어떤 유형도 가능하며, 예를 들어, 파일 이름이나 디렉토리를 사용함). 이 테이블 스페이스는 카탈로그 데이터베이스 카탈로그에서만 작성해야 합니다. 디폴트는 USERSPACE1입니다.

사용자는 디폴트 USERSPACE1을 사용하는 대신 시뮬레이션에 사용되는 테이블 스페이스를 작성하는 것이 좋습니다. 또한 ALTER TABLESPACE DROPPED TABLE RECOVERY OFF 문을 이 테이블 스페이스에서 실행하여 db2advis 유틸리티의 성능을 개선해야 합니다. 유틸리티가 완료될 때, 테이블 스페이스에 대해 실행기록을 다시 설정하십시오. 파티션된 데이터베이스 환경에서 이 옵션은 필수입니다. USERSPACE1은 보통 모든 파티션 그룹 사이에 작성되기 때문입니다.

### -d *database-name*

연결이 설정될 데이터베이스의 이름을 지정합니다.

### -delim *char*

위크로드 파일 입력에서 명령문 분리문자 *char*을 표시합니다. 디폴트값은 ';'입니다.

**-f** 이전에 존재하는 시뮬레이트된 카탈로그 테이블을 삭제합니다.

**-g** 동적 SQL 스냅샷에서의 SQL문 검색을 지정합니다. **-p** 명령 매개변수와 함께 조합하면, SQL문이 ADVISE\_WORKLOAD 테이블에서 보존됩니다. 이 옵션은 **-i**, **-s**, **-qp** 또는 **-w** 옵션과 함께 지정할 수 없습니다.

**-h** 도움말 정보를 표시합니다. 이 옵션이 지정되면 다른 모든 옵션은 무시되고 도움말 정보만 표시됩니다.

### -i *filename*

하나 이상의 SQL문이 포함된 입력 파일의 이름을 지정합니다. 디폴트는 표준



입력입니다. 각 라인의 시작 부분에 두 개의 하이픈을 사용하여 주석 텍스트를 식별하십시오(즉, -- comment). 명령문은 세미콜론으로 구분해야 합니다.

워크로드의 각 명령문이 실행될 빈도는 입력 파일에 다음 라인을 삽입하여 변경할 수 있습니다.

```
--#SET FREQUENCY x
```

빈도는 파일에서 얼마든지 갱신할 수 있습니다. 이 옵션은 **-g**, **-s**, **-qp** 또는 **-w** 옵션과 함께 지정할 수 없습니다.

**-k** 워크로드가 압축되는 정도를 지정합니다. 압축은 전체 워크로드가 고려될 때 어드바이저가 제공할 수 있는 것과 유사한 결과를 달성하면서 어드바이저가 어드바이저 실행의 복잡도를 줄일 수 있도록 하기 위해 수행됩니다. **HIGH**는 어드바이저가 작은 워크로드 서브세트에 집중할 것을 표시합니다. **MED**는 어드바이저가 중간 규모의 워크로드 서브세트에 집중할 것을 표시합니다. **LOW**는 어드바이저가 대형 워크로드 서브세트에 집중할 것을 표시합니다. **OFF**는 압축이 발생하지 않고 모든 쿼리가 고려됨을 표시합니다. 디폴트는 **MED**입니다.

#### **-l disk-limit**

기존 스키마에서 권장되는 모든 인덱스 및 구체화된 뷰에 대해 사용 가능한 MB 수를 지정합니다. 가능한 최대 크기를 사용하려면 **-1**을 지정하십시오. 디폴트값은 총 데이터베이스 크기의 20%입니다.

#### **-m advise-type**

어드바이저가 리턴할 권장사항의 유형을 지정합니다. **I**, **M**, **C** 및 **P**(대문자나 소문자)의 조합을 지정할 수 있습니다. 예를 들어, db2advise -m PC는 파티셔닝 및 **MDC** 테이블을 권장합니다. 파티션된 데이터베이스 환경에서 **-m P** 또는 **-m M**을 사용하는 경우, advise\_partition 테이블은 최종 파티션 권장사항으로 채워집니다. 가능한 값 선택사항은 다음과 같습니다.

**I** 새 인덱스를 권장합니다. 이는 디폴트값입니다.

**M** 새 구체화된 쿼리 테이블(MQT)과 MQT의 인덱스를 권장합니다. 파티션된 데이터베이스 환경에서는 MQT에 대한 파티셔닝도 권장됩니다.

**C** 표준 테이블을 다차원적으로 클러스터된(MDC) 테이블로 변환하거나, 테이블에서 클러스터링 인덱스를 작성할 권장사항.

**P** 기존 테이블의 재파티셔닝을 권장합니다.

#### **-mdcpcinflation percent**

MDC 권장사항에서 테이블 디스크 크기가 증가할 수 있는 최대 백분율을 지정합니다. 예를 들어, 테이블이 MDC 테이블로 변환될 때 원래 크기의 1+percent/100배까지 증가될 수 있음을 표시합니다. percent는 부동 소수점 숫자이며 디폴트값은 10입니다.

**-n** *schema-name*

시물레이션 카탈로그 테이블의 규정 이름과, 새 인덱스 및 MQT의 규정자를 지정합니다. 디폴트 스키마 이름이 SYSTOOLS인 카탈로그 시물레이션 테이블의 경우를 제외하고, 디폴트 스키마 이름은 호출자의 사용자 ID입니다. 디폴트는 새 인덱스가 인덱스 기본 스키마 이름을 상속하는 것입니다.

**-nogen**

생성된 컬럼이 다차원 클러스터링(MDC) 권장사항에 포함되지 않도록 지정합니다.

**-o** *outfile*

*outfile*에서 권장 오브젝트를 작성하기 위한 스크립트를 저장합니다.

**-p** Explain 테이블에서 도구를 실행하는 동안 생성된 플랜을 보존합니다. **-p** 명령 매개변수는 **-qp** 및 **-g**에 대한 워크로드가 ADVISE\_WORKLOAD 테이블에 저장되도록 하며 Explain 테이블에서 최종 권장사항을 사용하는 워크로드 쿼리 플랜을 저장합니다.

**-q** *schema-name*

워크로드에서 규정되지 않은 이름의 규정 이름을 지정합니다. db2advis가 실행될 때 CURRENT SCHEMA에 사용할 스키마 이름으로 사용됩니다. 디폴트 스키마 이름은 명령을 실행하는 사용자의 사용자 ID입니다.

**-qp** 워크로드가 DB2 Query Patroller에서 제공됨을 지정합니다. *start-time* 및 *end-time* 옵션은 DB2QP.TRACK\_QUERY\_INFO 테이블의 time\_completed 필드에 대해 점검하기 위해 사용되는 시간소인입니다. *start-time* 및 *end-time* 시간소인을 제공하지 않는 경우, 모든 행에는 테이블의 completion\_status 컬럼에 『D』(완료)가 제공됩니다. *start-time*만 제공하는 경우, 리턴되는 행은 TIME\_COMPLETED가 *start-time* 값 이상인 행입니다. 또한, *end-time* 값을 제공하는 경우에도, 리턴되는 행은 *end-time* 값 이하인 TIME\_COMPLETED에 의해 제한됩니다. 이 옵션은 **-w**, **-wlm**, **-s**, **-i** 또는 **-g** 옵션과 함께 사용할 수 없습니다.

*start-time*

시작 시간소인을 지정합니다.

*end-time*

종료 시간소인을 지정합니다. 이 매개변수는 선택적 매개변수입니다.

**-r** 가상 MQT 및 파티셔닝 선택에 세부 통계를 사용해야 함을 지정합니다. 이 옵션을 지정하지 않는 경우 디폴트는 MQT의 옵티마이저 통계를 사용하는 것입니다. 자세한 통계가 한층 정확할 수는 있지만, 파생 시간이 중요하므로 db2advis 실행 시간이 더 커집니다. **-r** 명령 매개변수는 샘플링을 사용하여 MQT 및 파

티셔닝에 대한 관련 통계를 확보합니다. MQT의 경우, 샘플 쿼리가 실패하거나 행을 리턴하지 않으면 옵티마이저 추정이 사용됩니다.

**-s** *"statement"*

평가될 단일 SQL문의 텍스트와, 디자인 어드바이저에서 인덱스가 제시되도록 지정합니다. 명령문은 큰따옴표로 묶어야 합니다. 이 옵션은 **-g**, **-i**, **-qp** 또는 **-w** 옵션과 함께 지정할 수 없습니다.

**-t** *max-advise-time*

조작을 완료할 수 있는 최대 허용 가능 시간(분)을 지정합니다. 이 옵션에 값을 지정하지 않는 경우 조작은 완료될 때까지 계속됩니다. 무제한 시간을 지정하려면 0 값을 입력하십시오. 디폴트값은 0입니다.

**-tables** *table-predicate-clause*

모든 기존 테이블의 서브세트만 고려해야 함을 표시합니다. *table-predicate-clause*는 SYSCAT.TABLES에 대한 쿼리의 WHERE절에 사용할 수 있는 술어여야 합니다. db2advise에 의해 고려되는 테이블은 이 쿼리의 테이블과 워크로드에 있는 테이블의 교차가 됩니다.

이 명령 매개변수는 새 MQT에 대한 권장사항에는 적용되지 않습니다.

**-u** 어드바이저가 지연된 MQT의 권장사항을 고려할 것을 지정합니다. 증분 MQT는 권장되지 않습니다. 이 옵션을 지정하는 경우, DDL CLP 스크립트의 주석은 중간 MQT로 변환될 수 있는 MQT를 표시합니다. 파티션된 데이터베이스 환경에서 중간 MQT가 권장되는 경우, 디폴트 분산 키는 MQT에 대한 내재된 고유 키입니다.

**-w** *workload-name*

평가될 워크로드의 이름과, 디자인 어드바이저에서 인덱스가 제시되도록 지정합니다. 이 이름은 ADVISE\_WORKLOAD 테이블에서 사용됩니다. 이 옵션은 **-g**, **-i**, **-qp** 또는 **-s** 옵션과 함께 지정할 수 없습니다.

**-wlm** *evmonname*

이벤트 이름 *evmonname*에 대한 SYSCAT.EVENTTABLES에서 ACTIVITY 및 ACTIVITYSTMT 논리 데이터 그룹에 해당되는 테이블 이름을 가져올 것을 지정하고, PARTIAL\_RECORD = 0(완료된 트랜잭션)의 레코드에 대한 ACTIVATE\_TIMESTAMP, ACTIVITY\_ID 및 ACTIVITY\_SECONDARY\_ID에서 테이블 이름을 함께 조인합니다. 선택적 *start-time* 및 *end-time* 시간소인을 추가하여 *start-time*에, 또는 그 이후에, 선택적으로 *end-time*이나 그 이전에 명령문을 가져올 수 있습니다. *start-time* 및 *end-time*은 ACTIVITY 테이블의 TIME\_COMPLETED 컬럼에 관련됩니다.

**workloadname | wl** *workloadname*

SYSCAT.WORKLOADS에서 검색되는 *workloadname*을 지정합니다.

ACTIVITY 이벤트 모니터 테이블은 워크로드 ID의 SYSCAT.WORKLOADS와 조인되어 이 명령문을 확보합니다.

**serviceclass | sc superclassname**

SYSCAT.SERVICECLASSES에서 제공되는 서비스 클래스 정보를 지정합니다. 서브클래스가 제공되지 않으면, 서비스 서브클래스(기본적으로 SYSCAT.SERVICECLASSES의 PARENTSERVICECLASS)에 대한 모든 명령문이 검색됩니다. ACTIVITY 이벤트 모니터 테이블은 서비스 클래스 ID의 SYSCAT.SERVICECLASSES와 조인되어 이 명령문을 확보합니다.

*,subclassname*

*superclassname*이 지정된 경우 쉼표로 구분하여 *subclassname*을 지정합니다. 이 매개변수는 선택적 매개변수입니다.

*start-time*

시작 시간소인을 지정합니다.

*end-time*

종료 시간소인을 지정합니다. 이 매개변수는 선택적 매개변수입니다.

-x 터미널 또는 사용자 입력을 통해 암호를 읽도록 지정합니다.

**예:**

1. 다음 예에서, 유틸리티는 데이터베이스 PROTOTYPE에 연결하고, 솔루션에 대한 제한조건 없이 테이블 ADDRESSES에 대한 인덱스를 권장합니다.

```
db2advise -d prototype -s "select * from addresses a
where a.zip in ('93213', '98567', '93412')
and (company like 'IBM%' or company like '%otus')"
```

2. 다음 예에서, 유틸리티는 데이터베이스 PROTOTYPE에 연결하고, ADVISE\_WORKLOAD 테이블에서 쿼리에 53MB를 초과하지 않을 인덱스를 권장합니다. 워크로드 이름은 "production"과 같습니다. 솔루션을 찾는 데 허용 가능한 최대 시간은 20분입니다.

```
db2advise -d prototype -w production -l 53 -t 20
```

3. 다음 예에서, 입력 파일 db2advise.in에는 SQL문과 각 명령문이 실행될 빈도 스펙이 포함됩니다.

```
--#SET FREQUENCY 100
SELECT COUNT(*) FROM EMPLOYEE;
SELECT * FROM EMPLOYEE WHERE LASTNAME='HAAS';
--#SET FREQUENCY 1
SELECT AVG(BONUS), AVG(SALARY) FROM EMPLOYEE
GROUP BY WORKDEPT ORDER BY WORKDEPT;
```

유틸리티는 데이터베이스 SAMPLE에 연결하고, 입력 파일에서 쿼리에 참조되는 각 테이블에 대한 인덱스를 권장합니다. 솔루션을 찾는 데 허용 가능한 최대 시간은 5 분입니다.

```
db2adviz -d sample -f db2adviz.in -t 5
```

- 다음 예에서, MQT는 테이블 스페이스 SPACE1에서 작성되고 시뮬레이션 테이블 스페이스는 SPACE2입니다. 워크로드에서 규정되지 않은 이름에 대한 규정 이름은 SCHEMA1이고, 새 MQT가 권장될 스키마 이름은 SCHEMA2입니다. 사용되는 워크로드 압축은 HIGH이고 디스크 스페이스는 무제한입니다. MQT에 샘플 통계가 사용됩니다. 다음 명령을 발행하면 MQT가 권장되고, 파티션된 데이터베이스 환경에서는 인덱스 및 파티셔닝도 권장됩니다.

```
db2adviz -d prototype -w production -l -1 -m M -b space1 -c space2 -k  
HIGH -q schema1 -n schema2 -r
```

권장되는 MQT와 MQT 및 기본 테이블의 인덱스, 파티셔닝 및 MDC도 가져오려면 다음과 같이 **-m** 옵션에 대해 IMCP 값을 지정하여 명령을 발행하십시오.

```
db2adviz -d prototype -w production -l -1 -m IMCP -b space1 -c space2 -k  
HIGH -q schema1 -n schema2 -r
```

- 다음 예에서, 유틸리티는 데이터베이스 SAMPLE에 연결하고, MDC 후보가 원래 크기의 30.5%까지 증가할 수 있는 EMPLOYEE 및 DEPT에 대한 테이블의 MDC를 권장합니다.

```
db2adviz -d sample -type C -disklimit 100  
-tables "TABNAME IN ('EMPLOYEE','DEPT')" -mdcpcinflation 30.5
```

## 사용 시 참고사항

이 기능은 DDL CLP 스크립트를 실행하기 전에 설정해야 하므로, 데이터베이스 파티셔닝, 다차원 클러스터링(MDC) 및 클러스터된 인덱스 권장사항은 리턴되는 DDL CLP 스크립트에서 주석 처리됩니다. 테이블을 권장되는 DDL로 변환하는 것은 사용자에게 달려 있습니다. 이에 대한 한 가지 예로, ALTER TABLE 스토어드 프로시저를 사용하지만 RENAME문이 제한되는 것과 같은 방식으로 연관되는 제한사항이 있습니다.

버전 9.7부터, 디자인 어드바이저는 파티션된 인덱스를 권장하지 않습니다. 모든 인덱스는 NOT PARTITIONED 절과 함께 권장됩니다. 이 권장사항에서, 응용프로그램 시나리오와 파티션된 인덱스가 제공할 수 있는 이득에 따라 인덱스를 작성하기 위해 PARTITIONED(디폴트) 또는 NOT PARTITIONED 중 어느 것을 사용할 것인지 사용자가 선택합니다.

동적 SQL문의 경우, 명령문이 실행되는 빈도는 다음과 같이 모니터에서 확보할 수 있습니다.

- 다음 명령을 발행하십시오.

```
db2 reset monitor for database database-alias
```

해당 시간 간격 동안 기다리십시오.

2. 다음 명령을 발행하십시오.

```
db2advis -g other-options
```

**-p** 매개변수를 **-g** 매개변수와 함께 사용하는 경우, 확보되는 동적 SQL문은 시간소인을 포함하는 생성된 워크로드 이름의 ADVISE\_WORKLOAD 테이블에 위치됩니다.

워크로드에서 각 SQL문에 대한 디폴트 빈도는 1이고, 디폴트 중요성 역시 1입니다. generate\_unique() 함수는 해당 SQL문의 한층 의미있는 설명이 되도록 사용자가 갱신할 수 있는 명령문의 고유 ID를 지정합니다.

db2diag 로그 파일에서 db2advis 오류 정보를 찾을 수도 있습니다.

어드바이저가 실행을 시작할 때, ADVISE\_INSTANCE 테이블은 어드바이저를 식별하는 행을 포함합니다. 기본 어드바이저 행은 어드바이저가 실행을 시작한 시기를 보여주는 START\_TIME으로 식별됩니다. 이 행의 STATUS는 『STARTED』입니다.

db2advis 명령을 발행한 결과 "Cannot insert into DB2ADVIS\_INSTANCE" 오류가 발생하면, db2advis.bnd를 바인드하고 **-l** 옵션과 함께 db2advis 명령을 실행해야 합니다. 바인드 조작은 다음 명령을 발행하여 수행할 수 있습니다.

```
db2 bind db2advis.bnd blocking all grant public
```

어드바이저가 완료될 때, ADVISE\_INSTANCE 테이블에서 적절한 START\_TIME과 연관되는 행을 점검할 수 있습니다. STATUS가 『COMPLETED』이면 어드바이저는 성공적으로 실행된 것입니다. STATUS가 계속 『STARTED』이고 실행 중인 db2advis 프로세스가 없는 경우, 어드바이저가 너무 빨리 종료된 것입니다. STATUS에 『EX』가 있으면, 어드바이저가 어떻게 실패했는지 판별하기 위한 『SQLCODE』도 표시됩니다.

**-l disk-limit** 옵션을 지정하지 않는 경우, GET\_DBSIZE\_INFO 스토어드 프로시저를 사용하여 최대 데이터베이스 크기를 판별하려면 하나 이상의 SYSADM, SYSCTRL, SYSMOINT 또는 SYSMON 권한을 가지고 있어야 합니다.

**-tables** 매개변수의 *table-predicate-clause*는 SYSCAT.TABLES를 쿼리하고 어드바이저가 고려할 테이블을 판별하는 데 사용됩니다. 기본 테이블 또는 기존 MQT만 고려할 수 있지만, 별명 및 논리 뷰를 *table-predicate-clause*에서 사용하여 기본 테이블 이름 또는 MQT 목록을 리턴할 수 있습니다. 예를 들어, 'TV'로 시작하는 뷰가 있는 테이블 서브세트를 지정하려면 `-tables "(tabname, tabschema) in (SELECT bname, bschema FROM SYSCAT.TABDEP WHERE TABNAME LIKE 'TV%')"`를 지정하십시오.

버전 9.7 현재, 쿼리 옵티마이저는 비용 모델에서 키 값 및 RID를 압축 해제하는 비용과 입출력 절약 비용을 측정합니다. 이와 같이, 인덱스 어드바이저는 압축된 인덱스 크기를 계산할 수 있습니다.

---

## 제 164 장 db2audit - 감사 기능 관리자 도구

DB2는 알 수 없거나 데이터에 대한 예상하지 못한 액세스 발견을 지원하기 위한 감사 기능을 제공합니다. DB2 감사 기능은 일련의 사전 정의된 데이터베이스 이벤트에 대한 감사 추적 유지보수를 생성하고 허용합니다. 이 유틸리티에서 생성되는 레코드는 감사 로그 파일에 보존됩니다. 이들 레코드의 분석은 시스템 오용을 식별하는 사용 패턴을 보여줄 수 있습니다. 식별된 후에는 그러한 시스템 오용을 줄이거나 제거하기 위한 조치를 취할 수 있습니다. 감사 기능은 인스턴스 또는 데이터베이스를 기반으로 별도의 로그에 있는 모든 활동을 독립적으로 기록하여 인스턴스 및 데이터베이스 레벨 모두에서 작동합니다.

DB2는 인스턴스 및 개별 데이터베이스 레벨 모두에서 독립적으로 감사하는 기능을 제공합니다. db2audit 도구는 인스턴스 레벨에서 감사를 구성하고 그런 감사 정보가 수집될 때 감사를 제어하는 데 사용됩니다. AUDIT SQL문은 개별 데이터베이스에 대한 감사 요구사항을 구성 및 제어하는 데 사용됩니다. db2audit 도구를 사용하면 인스턴스 및 데이터베이스 감사 로그를 둘 다 아카이브하고 어느 한 유형의 아카이브된 로그에서 추출할 수 있습니다. 추가 정보는 아래 관련 링크 섹션을 참조하십시오.

파티션된 데이터베이스 환경에서 많은 감사 이벤트는 사용자가 연결된 데이터베이스 파티션(코드네이터 파티션) 또는 카탈로그 파티션(동일한 데이터베이스 파티션이 아닌 경우)에서 발생합니다. 즉 이는 둘 이상의 데이터베이스 파티션이 감사 레코드를 생성할 수 있다는 것을 의미합니다. 각 감사 레코드의 파트에는 코드네이터 파티션 및 시작 데이터베이스 파티션 ID에 대한 정보가 들어 있습니다.

인스턴스 감사 로그(db2audit.instance.log.node\_number[.timestamp])는 인스턴스의 security/auditdata 서브디렉토리에 있으며, 감사 구성 파일(db2audit.cfg)은 인스턴스의 security 서브디렉토리에 있습니다. 데이터베이스 감사 로그 이름은 db2audit.db.dbname.log.node\_number[.timestamp]입니다. 인스턴스를 작성할 때, 가능한 경우 운영 체제가 이들 파일에 대한 읽기/쓰기 권한을 설정합니다. 디폴트로 권한은 인스턴스 소유자에 대해서만 읽기/쓰기입니다. 이들 권한은 변경하지 않는 것이 좋습니다.

감사 기능의 권한 부여된 사용자가 db2audit를 사용하여 감사 기능의 다음 조치를 제어할 수 있습니다.

- DB2 인스턴스에서 감사 가능 이벤트 기록을 시작합니다. 여기에는 데이터베이스 레벨 활동이 포함되지 않습니다.
- DB2 인스턴스에서 감사 가능 이벤트 기록을 중지합니다.
- 인스턴스 레벨에서만 감사 기능의 동작을 구성합니다.
- 인스턴스 레벨에서만 기록될 감사 가능 이벤트의 범주를 선택합니다.

- 인스턴스에 대한 현재 감사 구성의 설명을 요청합니다.
- 인스턴스에서 모든 보류 감사 레코드를 플래시(비우기)하고 감사 로그에 씁니다.
- 인스턴스 또는 인스턴스 아래의 데이터베이스에 대한 현재 감사 로그의 감사 레코드를 아카이브합니다.
- 감사 레코드를 형식화하고 플랫폼 파일 또는 ASCII 구분 파일에 복사하여 아카이브된 로그에서 감사 레코드를 추출합니다. 추출은 로그 레코드 분석 준비 과정에서 수행됩니다.

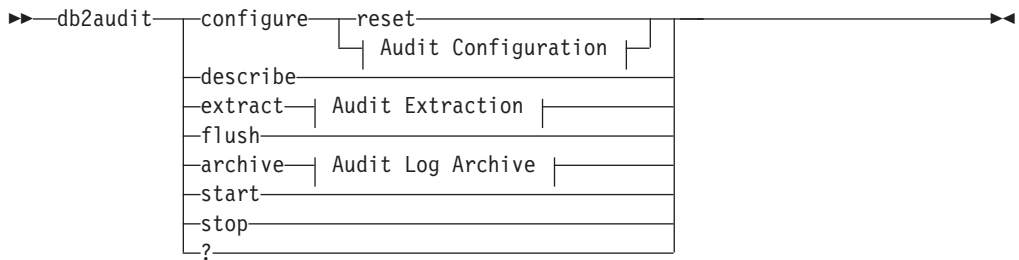
## 권한 부여

*sysadm*

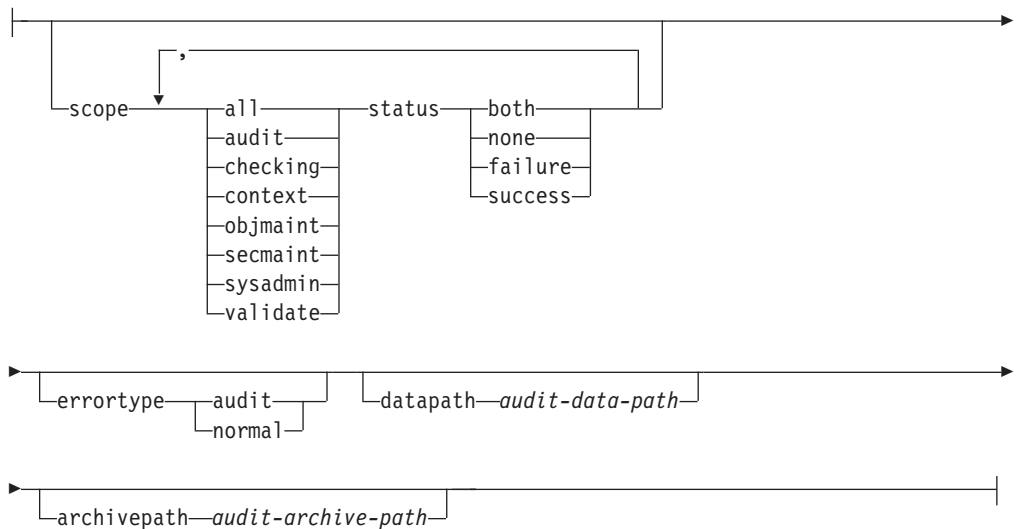
## 필요한 연결

없음

## 명령 구문

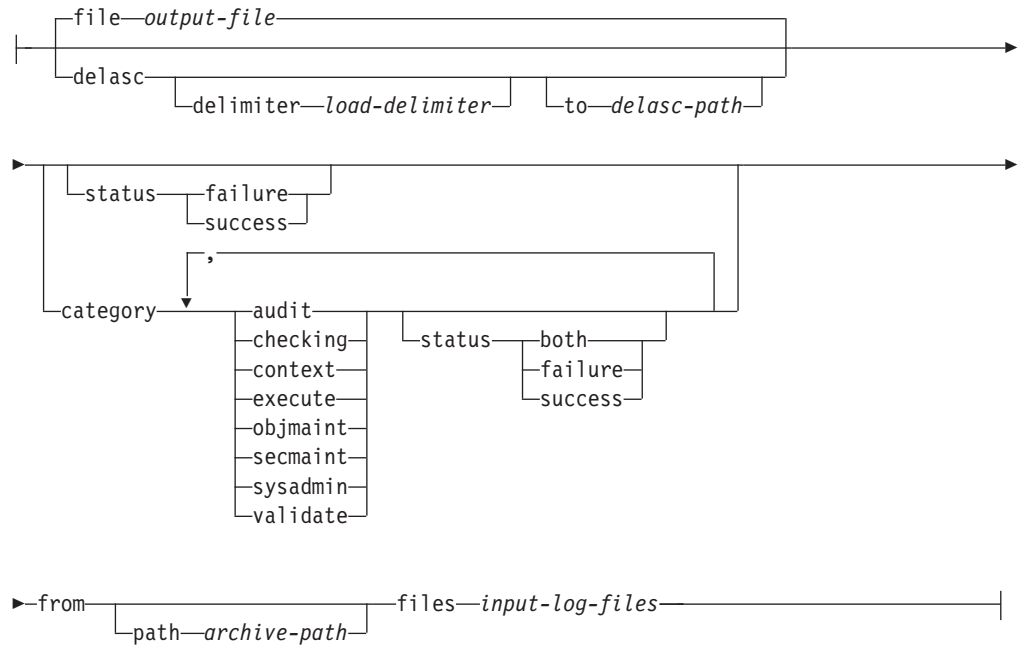


## 감사 구성:

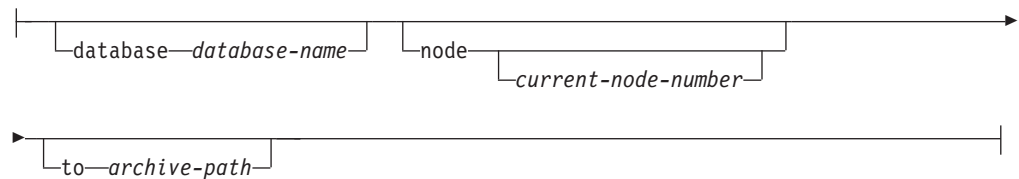




감사 추출:



감사 로그 아카이브:



## 명령 매개변수

### configure

이 매개변수는 인스턴스의 security 서브디렉토리에 있는 db2audit.cfg 구성 파일의 수정을 허용합니다. 이 파일에 대한 갱신은 인스턴스가 중지되었을 때도 발생할 수 있습니다. 인스턴스가 사용 중일 때 발생하는 갱신은 DB2 인스턴스가 수행 중인 감사에 동적으로 영향을 줍니다. 구성 파일의 구성 조치는 감사 기능이 시작되었고 감사 가능 이벤트의 **audit** 범주가 감사되고 있는 경우 감사 레코드가 작성되도록 합니다. 데이터 경로 및 아카이브 경로를 제외한 모든 구성 옵션은 인스턴스 레벨 감사 이벤트에만 적용되며 데이터베이스 레벨 감사 이벤트에는 적용되지 않습니다. 경로 옵션은 인스턴스 및 인스턴스의 모든 데이터베이스에 적용됩니다.

다음은 구성 파일에 대해 가능한 조치입니다.

**reset** 이 조치는 구성 파일이 초기 구성으로 되돌아가게 합니다(여기서 scope는 **context**를 제외한 모든 범주이고, 각 범주의 status는 failure

이며 `errortype`은 `normal`, 인스턴스 레벨 이벤트의 감사는 `off`임). 이 조치는 원본이 유실되거나 손상된 경우 새 구성 파일을 작성합니다. 감사 데이터 경로 및 아카이브 경로는 공백입니다. 이 옵션은 감사 규정 또는 데이터베이스 레벨에서 해당 규정의 사용을 재설정하지 않습니다.

**scope** 이 조치는 감사할 범주 및 해당되는 각 범주의 상태를 지정합니다.

**status** 이 조치는 성공 또는 실패한 이벤트만 또는 성공 및 실패 이벤트 모두 로그될지 여부를 지정합니다. `status`는 다음 옵션을 갖습니다.

**both** 성공 및 실패 이벤트가 감사됩니다.

**none** 이 범주의 이벤트는 감사하지 않습니다.

**failure**

실패 이벤트만 감사합니다.

**success**

성공 이벤트만 감사합니다.

`configure`문에 지정된 범주만 수정됩니다. 다른 모든 범주는 상태가 보존됩니다.

주:

- 디폴트 `scope`는 `context`를 제외한 모든 범주이며 레코드가 빨리 생성되게 할 수 있습니다. 모드(동기 또는 비동기)와 결합하여 범주를 선택하면 성능이 매우 저하되고 디스크 요구사항이 크게 늘어날 수 있습니다. 로그되는 이벤트의 수와 유형을 가능한 많이 제한하는 것이 바람직하며, 그렇지 않으면 감사 로그 크기가 신속하게 늘어납니다. 이 조치는 또한 감사에 대한 특정 축점을 허용하고 로그의 증가를 줄입니다.
- `context` 이벤트는 작업 상태가 알려지기 전에 발생합니다. 그러므로 그런 이벤트는 `status`가 `none`이 아닌 한 이 매개변수와 연관된 값과 상관 없이 로그됩니다.
- 동일한 범주가 반복되거나 `all` 키워드와 함께 범주도 지정하는 경우 구문 오류가 리턴됩니다.

**errortype**

이 조치는 감사 오류가 사용자에게 리턴되는지 아니면 무시되는지 여부를 지정합니다. 이 매개변수의 값은 다음과 같습니다.

**audit** 감사 기능에서 발생하는 오류를 포함한 모든 오류는 DB2 데이터베이스에 의해 관리되며 모든 음수 `SQLCODE`가 다시 호출자에게 보고됩니다.

## normal

db2audit가 생성하는 모든 오류가 무시되고 수행되고 있는 작업과 연관된 오류에 대한 SQLCODE만 응용프로그램으로 리턴됩니다.

## datapath *audit-data-path*

이것은 DB2가 생성하는 감사 로그가 기록될 디렉토리입니다. 디폴트는 `sqllib/security/auditdata`(Windows의 `<instance path>\instance\security\auditdata`)입니다. 이 매개변수는 데이터베이스 레벨 감사를 포함하여 인스턴스 내의 모든 감사에 영향을 줍니다. 상대 경로가 아니라 완전한 경로여야 합니다. 인스턴스 소유자가 이 디렉토리에 대한 쓰기 권한이 있어야 합니다. Windows에서는 로컬 인스턴스 명령(예: `db2start`, `db2audit` 및 `db2 update dbm cfg`)도 감사해야 하는 경우 해당 명령을 발행하는 사용자도 이 디렉토리에 대한 쓰기 권한이 있어야 합니다. DPF 시스템에서 이 디렉토리가 NFS 공유 디렉토리일 필요는 없지만 불가능한 것은 아닙니다. 비공유 디렉토리는 각 노드가 고유 디스크에 기록되기 때문에 성능이 증가됩니다. 경로의 최대 길이는 971바이트(UNIX/Linux) 및 208바이트(Windows 시스템)입니다.

경로가 ""로 제공되는 경우 경로는 디폴트로 갱신됩니다. `db2audit describe`는 설정된 경로가 없음을 표시하며 디폴트 경로가 사용됩니다. 셸이 인용 부호를 해석하지 못하게 하려면 일반적으로 인용 부호를 이스케이프해야 함을 주의하십시오. 예:

```
db2audit configure datapath #"#"
```

데이터 경로가 존재해야 합니다. DPF 환경에서 각 노드에서 동일한 데이터 경로가 사용됩니다. 데이터베이스 파티션 표현식이 데이터 경로 이름의 파트로 사용되지 않으면 특정 노드에 대한 고유한 데이터 경로 세트를 지정할 방법이 없습니다. 이를 수행하면 결과 경로 이름이 각 데이터베이스 파티션에서 다르도록 노드 번호가 스토리지 경로에 반영될 수 있습니다. 데이터베이스 파티션 표현식에 대한 정보는 자동 스토리지 데이터베이스를 참조하십시오.

## archivepath *audit-archive-path*

이것은 아카이브 및 추출 옵션에 대한 디폴트 디렉토리입니다. DPF 환경에서는 이 디렉토리가 모든 노드가 액세스할 수 있는 NFS 공유 디렉토리인 것이 바람직합니다. 디폴트는 `sqllib/security/auditdata`(Windows에서는 `sqllib\instance\security\auditdata`)입니다. 상대 경로가 아니라 완전한 경로여야 합니다. 인스턴스 소유자가 이 디렉토리에 대한 쓰기 권한이 있어야 합니다. 경로의 최대 길이는 971바이트(UNIX/Linux) 및 208바이트(Windows 시스템)입니다.

아카이브 경로가 존재해야 하며, 데이터베이스 파티션 표현식은 아카이브 경로에 허용되지 않습니다.

### describe

이 매개변수는 표준 출력에 현재 인스턴스 레벨 감사 구성 정보 및 상태를 표시합니다.

다음 항목이 표시됩니다.

- 감사가 활성화인지 여부.
- 각 범주의 상태.
- 오류 시 SQLCA가 리턴되는지 여부의 양식으로 오류 유형.
- 데이터 및 아카이브 경로.

다음은 describe 출력의 예입니다.

DB2 AUDIT SETTINGS:

```
Audit active: "FALSE "  
Log audit events: "SUCCESS"  
Log checking events: "FAILURE"  
Log object maintenance events: "BOTH"  
Log security maintenance events: "BOTH "  
Log system administrator events: "NONE"  
Log validate events: "FAILURE"  
Log context events: "NONE"  
Return SQLCA on audit error: "TRUE "  
Audit Data Path: "/auditdata"  
Audit Archive Path: "/auditarchive"
```

AUD0000I Operation succeeded.

### extract

이 매개변수는 감사 로그에서 표시된 대상으로 감사 레코드의 이동을 허용합니다. 감사 로그는 데이터베이스 코드 페이지에 작성됩니다. 추출이 실행될 때 모든 필드가 현재 응용프로그램 코드 페이지로 변환됩니다.

다음은 추출할 때 사용할 수 있는 옵션입니다.

#### **file** *output-file*

추출된 감사 레코드는 *output-file*에 배치됩니다. 디렉토리가 지정되지 않은 경우 *output-file*은 현재 작업 디렉토리에 기록됩니다. 해당 파일이 이미 존재하면 출력이 추가됩니다. 파일 이름을 지정하지 않으면 레코드는 감사 구성 파일에서 지정된 아카이브 경로의 db2audit.out 파일에 기록됩니다.

#### **delasc**

추출된 감사 레코드는 DB2 데이터베이스 관계형 테이블에 로드하기에 적합한 컬럼 식별자가 있는 ASCII 형식으로 배치됩니다. 출력은 각 범주에

대해 하나씩, 별도 파일에 배치됩니다. 또한 `auditlobs` 파일도 작성되어 감사 데이터에 포함되는 모든 LOB를 보유합니다. 파일 이름은 다음과 같습니다.

- `audit.del`
- `checking.del`
- `objmaint.del`
- `secmaint.del`
- `sysadmin.del`
- `validate.del`
- `context.del`
- `execute.del`
- `auditlobs`

파일이 이미 존재하면 출력이 파일에 추가됩니다. **context** 또는 **execute** 범주가 추출되는 경우 `auditlobs` 파일이 작성됩니다. `auditlobs` 파일에서 LOB를 참조하기 위해 LOB 위치 지정자가 `.del` 파일에 포함됩니다.

#### **delimiter** *load-delimiter*

감사 로그에서 추출할 때 디폴트 감사 문자열 분리문자(큰따옴표(""))를 겹쳐쓸 수 있습니다. 감사 레코드를 보유할 테이블에 로드하기 위한 준비 과정에서 사용하려는 새 분리문자를 `delimiter` 뒤에 사용합니다. 새 로드 분리문자는 단일 문자(예: `!`) 또는 16진 숫자를 표시하는 4 문자 열(예: `0xff`)일 수 있습니다.

#### **to** *delasc-path*

구분된 파일이 기록되는 경로를 지정할 수 있습니다. 지정되지 않는 경우 파일은 감사 구성 파일에서 지정되는 감사 아카이브 경로 옵션으로 표시되는 디렉토리에 기록됩니다.

#### **category**

감사 이벤트의 지정된 범주에 대한 감사 레코드가 추출됩니다. 지정하지 않으면 모든 범주를 추출할 수 있습니다.

#### **status**

지정된 상태에 대한 감사 레코드가 추출됩니다. 지정하지 않으면 모든 레코드를 추출할 수 있습니다.

#### **path**

아카이브된 감사 로그의 위치로의 경로입니다. 이것을 지정하지 않으면 감사 구성의 아카이브 경로가 사용됩니다. 파일 이름에 완전한 경로가 포함되는 경우 해당 경로는 사용되지 않습니다.

## files

추출될 감사 로그 파일의 목록입니다. 이것은 단일 파일 또는 파일 목록일 수 있습니다. 이들 파일은 추출 중에 변경되지 않습니다. 파일 이름이 아직 완전하지 않은 경우 파일 이름은 path와 결합되어 완전한 파일 이름을 확보합니다. 목록에 표준 셸 와일드 카드가 포함되어 다중 파일을 지정할 수 있습니다.

**flush** 이 매개변수는 모든 보류 감사 레코드가 감사 로그에 기록되게 합니다. 또한 감사 기능이 오류 상태에 있는 경우 감사 상태가 "unable to log"에서 "ready to log" 상태로 재설정됩니다.

## archive

이 매개변수는 개별 데이터베이스 또는 인스턴스에 대한 현재 감사 로그를 아카이브 및 추후 추출을 위한 새 위치로 이동합니다. 현재 시간소인이 파일 이름에 추가됩니다. 현재 감사 로그에 기록되고 있는 모든 레코드는 전체 레코드가 분할되지 않도록 하기 위해 로그가 아카이브되기 전에 완료됩니다. 아카이브가 진행 중인 동안 작성되는 모든 레코드는 아카이브가 완료된 후 아카이브된 로그가 아니라 현재 감사 로그에 기록됩니다.

다음은 아카이브할 때 사용할 수 있는 옵션입니다.

### database *database-name*

감사 로그를 아카이브하려는 데이터베이스의 이름입니다. 데이터베이스 이름을 제공하지 않으면 인스턴스 레벨 감사 로그가 아카이브됩니다.

## node

아카이브 명령이 현재 노드에 대해서만 실행되며 `node_number` 모니터 요소가 어느 것이 현재 노드인지 표시함을 표시합니다. 이것은 DPF 시스템에서만 필수입니다.

### *current-node-number*

현재 실행 중인 노드에 관해 db2audit 실행 파일에 알립니다. 이 매개변수는 DB2NODE 환경 변수에 현재 노드가 없는 경우 필수입니다.

### to *archive-path*

아카이브된 감사 로그가 작성될 디렉토리입니다. 디렉토리가 존재해야 하며 인스턴스 소유자는 이 디렉토리에 대한 작성 권한이 있어야 합니다. 제공되지 않는 경우 감사 구성의 아카이브 경로가 사용됩니다.

작성되는 파일 이름의 형식은 다음과 같습니다.

- 인스턴스 로그의 경우  
`db2audit.instance.log.node_number[.YYYYMMDDHHMMSS]`
- 데이터베이스 로그의 경우  
`db2audit.db.dbname.log.node_number[.YYYYMMDDHHMMSS]`

여기서 YYYY는 년, MM은 월, DD는 일, HH는 시, MM은 분, SS는 초입니다. 시간은 로컬 시간입니다. 인스턴스 감사 로그의 경우 데이터베이스 이름 분할 영역이 표시되지 않습니다. 비DPF 환경의 노드 번호는 0입니다. 파일이 이미 존재하는 경우 추가가 수행됩니다.

시간소인은 100% 정밀도를 갖고 로그의 마지막 레코드를 반영하지는 않습니다. 시간소인은 아카이브 명령이 실행된 시간을 표시합니다. 현재 로그 파일에 기록되고 있는 항목은 이동될 수 있기 전에 완료되어야 하며, 이들 항목은 파일 이름에 주어진 시간소인보다 더 늦은 시간소인을 가질 수 있습니다.

node 옵션이 지정되지 않은 경우 모든 노드의 감사 로그가 아카이브됩니다. 이 경우 데이터베이스 서버가 시작되어야 합니다. 데이터베이스 서버가 시작되지 않은 경우 아카이브가 각 노드에서 실행되어야 하며, archive가 실행될 노드 (AUD0029)를 표시하기 위해 node 옵션을 지정해야 합니다.

archive 옵션은 결과 및 아카이브가 실행된 각 노드의 파일 이름을 출력합니다.

**start** 이 매개변수는 감사 기능이 인스턴스에 대한 db2audit.cfg 파일의 콘텐츠만을 기반으로 감사 이벤트를 시작하게 합니다. 파티션된 DB2 데이터베이스 인스턴스에서, 감사는 이 절이 지정되는 모든 데이터베이스 파티션의 인스턴스 및 클라이언트 레벨 활동에 대해 시작됩니다. 이벤트의 **audit** 범주가 감사에 대해 지정된 경우 감사 기능이 시작될 때 감사 레코드가 로그됩니다. 이것은 데이터베이스 레벨 감사에 영향을 주지 않으며, 이는 AUDIT DDL문을 통해 제어됩니다.

**stop** 이 매개변수는 감사 기능이 인스턴스에 대한 감사 이벤트만 중지하게 합니다. 파티션된 DB2 데이터베이스 인스턴스에서, 감사는 이 절이 지정되는 모든 데이터베이스 파티션의 인스턴스 및 클라이언트 레벨 활동에 대해 중지됩니다. 이벤트의 **audit** 범주가 감사에 대해 지정된 경우 감사 기능이 중지될 때 감사 레코드가 로그됩니다. 이것은 데이터베이스 레벨 감사에 영향을 주지 않으며, 이는 AUDIT DDL문을 통해 제어됩니다.

? 이 매개변수는 db2audit 명령에 대한 도움말 정보를 표시합니다.

### 예:

이것은 DPF 환경에서 컬럼 식별자가 있는 ASCII 파일을 아카이브 및 추출하는 방법의 전형적인 예입니다. Windows remove(rm) 명령은 이전 컬럼 식별자가 있는 ASCII 파일을 삭제합니다.

```
rm /auditdelasc/*.del  
db2audit flush  
db2audit archive database mydb to /auditarchive
```

(파일은 다음 단계에서 사용되도록 표시됩니다.)

db2audit extract delasc to /auditdelasc from files /auditarchive  
/db2audit.db.mydb.log.\*.20070514102856

.del 파일을 DB2 테이블에 로드하십시오.

## 사용 시 참고사항

- 데이터베이스 레벨 감사는 AUDIT문으로 제어됩니다. 추가 정보는 아래 [관련 링크](#) 섹션을 참조하십시오.
- 인스턴스 레벨 감사 기능은 명시적으로 중지 및 시작되어야 합니다. 시작할 때 감사 기능은 기존의 감사 구성 정보를 사용합니다. 감사 기능은 DB2 데이터베이스 서버와 무관하므로 인스턴스가 중지되는 경우에도 계속 활성 상태입니다. 사실, 인스턴스가 중지될 때 감사 로그에 감사 레코드가 생성될 수 있습니다.
- 감사 기능을 사용하기 전에 db2audit start 명령을 발행하여 감사 기능이 켜졌는지 확인하십시오.
- 생성될 수 있는 여러 가지 범주의 감사 레코드가 있습니다. 감사에 사용할 수 있는 이벤트 범주의 설명에서(아래), 각 범주의 이름 뒤에 범주 유형을 식별하는 데 사용되는 1단어 키워드가 있음을 유의해야 합니다. 감사에 사용할 수 있는 이벤트의 범주는 다음과 같습니다.
  - 감사(**audit**). 감사 설정이 변경될 때 또는 감사 로그가 액세스될 때 레코드를 생성합니다.
  - 권한 부여 검사(**checking**). DB2 데이터베이스 오브젝트 또는 기능을 액세스 또는 조작하려는 시도의 권한 부여 검사 중에 레코드를 생성합니다.
  - 오브젝트 유지보수(**objmaint**). 데이터 오브젝트를 작성 또는 삭제할 때 레코드를 생성합니다.
  - 보안 유지보수(**secmaint**). 오브젝트 또는 데이터베이스 특권 또는 DBADM 권한을 부여하거나 취소할 때 레코드를 생성합니다. 데이터베이스 관리 프로그램 구성 매개변수 SYSADM\_GROUP, SYSCTRL\_GROUP 또는 SYSMANT\_GROUP이 수정될 때도 레코드가 생성됩니다.
  - 시스템 관리(**sysadmin**). SYSADM, SYSMANT 또는 SYSCTRL 권한이 필요한 작업이 수행될 때 레코드를 생성합니다.
  - 사용자 유효성 확인(**validate**). 사용자를 인증하거나 시스템 보안 정보를 검색할 때 레코드를 생성합니다.
  - 조작 컨텍스트(**context**). 인스턴스 작업이 수행될 때 조작 컨텍스트를 표시하는 레코드를 생성합니다. 이 범주는 감사 로그 파일의 더 좋은 해석을 허용합니다. 로그의 이벤트 상관자 필드와 함께 사용될 때 이벤트 그룹을 다시 단일 데이터베이스 조작과 연관시킬 수 있습니다.
  - 실패, 성공, 둘 다 감사하거나 둘 다 감사하지 않을 수 있습니다.

추가 정보는 아래 [관련 링크](#) 섹션을 참조하십시오.



- 인스턴스에 대한 모든 작업은 여러 레코드를 생성할 수 있습니다. 생성되고 감사 로그로 이동되는 레코드의 실제 수는 감사 기능 구성에 의해 지정되는 대로 기록될 이벤트의 범주 수에 따라 다릅니다. 또한 성공, 실패 또는 둘 다 감사되는지 여부에 따라 다릅니다. 이 때문에 감사할 이벤트를 선택할 수 있는 것이 중요합니다.
- 감사 로그를 제거하거나 보려면 정기적으로 archive를 실행한 후 아카이브된 파일에 대해 extract를 실행하여 유용한 정보를 저장하십시오. 그런 다음 표준 파일 시스템 삭제 명령으로 감사 로그를 삭제할 수 있습니다.



## 제 165 장 db2batch - 벤치마크 도구

플랫 파일이나 표준 입력에서 SQL문 및 XQuery 명령문을 읽고, 명령문을 동적으로 준비하고 설명하며 응답 세트를 리턴합니다.

이 도구는 단일 파티션 데이터베이스 및 다중 파티션 데이터베이스에서 작업할 수 있습니다.

도구의 선택적 매개변수를 통해 응답 세트에서 페치될 행 수, 출력 파일이나 표준 출력으로 보낼 페치된 행 수 및 리턴될 성능 정보 레벨을 제어할 수 있습니다.

출력 디폴트는 표준 출력을 사용하는 것입니다. 결과 요약에 대한 출력 파일 이름을 지정할 수 있습니다.

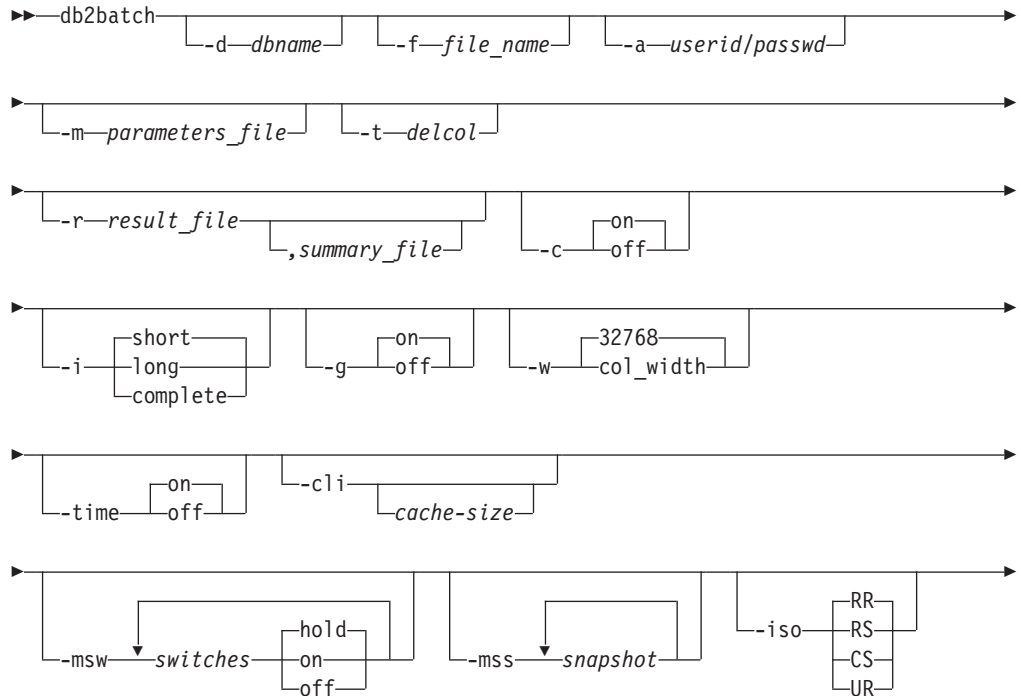
### 권한 부여

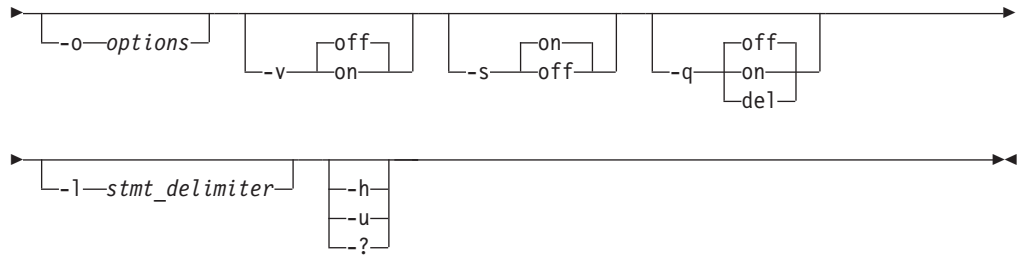
읽을 SQL문 또는 XQuery 명령문에서 필요한 것과 동일한 권한 레벨.

### 필수 연결

없음. 이 명령은 데이터베이스 연결을 설정합니다.

### 명령 구문





## 명령 매개변수

### -d dbname

SQL문 및 XQuery 명령문이 적용될 데이터베이스에 대한 별명 이름입니다. 이 옵션을 지정하지 않으면 DB2DBDFT 환경 변수 값이 사용됩니다.

### -f file\_name

SQL문 및 XQuery 명령문이 들어있는 입력 파일의 이름입니다. 디폴트는 표준 입력입니다.

주석 텍스트 앞에 두 개의 하이픈을 추가하여 주석 텍스트를 식별하십시오(예: `--comment`). 라인의 끝까지 두 하이픈 뒤에 오는 모든 텍스트는 주석으로 취급됩니다. 작은따옴표나 큰따옴표로 구분되는 문자열에 두 개의 인접한 하이픈이 포함될 수 있으며, 이들은 주석이 아니라 문자열 상수로 취급됩니다. 출력에 주석을 포함하려면 다음과 같이 표시하십시오. `--#COMMENT comment`.

블록은 하나로 취급되는 SQL문 및 XQuery 명령문의 그룹입니다. 디폴트로 한 번에 하나씩이 아니라 블록에 있는 모든 명령문에 대한 정보가 한 번에 수집됩니다. 쿼리 블록의 시작은 다음과 같이 식별하십시오. `--#BGBLK`. 쿼리 블록의 끝은 다음과 같이 식별하십시오. `--#EOBLK`. 쿼리 블록은 블록을 정의할 때 반복 계수를 지정하여 반복 루프에 포함될 수 있습니다. 예: `--#BGBLK repeat_count`. 블록의 명령문은 루프의 첫 번째 반복에서만 준비됩니다.

`#PARAM` 지시문이나 매개변수 파일을 사용하여 블록의 주어진 명령문 및 주어진 반복에 대한 매개변수 값을 지정할 수 있습니다. 자세한 내용은 아래 `-m` 옵션을 참조하십시오.

다음과 같이 하나 이상의 제어 옵션을 지정하십시오. `--#SET control option value`. 유효한 제어 옵션은 다음과 같습니다.

### ROWS\_FETCH

응답 세트에서 페치될 행 수입니다. 유효 값은 -1부터  $n$ 까지입니다. 디폴트값은 -1(모든 행이 페치됨)입니다.

### ROWS\_OUT

출력으로 전송될 페치된 행 수입니다. 유효 값은 -1부터  $n$ 까지입니다. 디폴트값은 -1(모든 페치된 행이 출력으로 전송됨)입니다.

## **PERF\_DETAIL** *perf\_detail*

리턴될 성능 정보 레벨을 지정합니다. 가능한 값은 다음과 같습니다.

- 0** 시간 제어 정보 또는 모니터링 스냅샷을 리턴하지 않습니다.
- 1** 경과 시간만 리턴합니다.
- 2** 경과 시간 및 응용프로그램의 스냅샷을 리턴합니다.
- 3** 경과 시간과, 데이터베이스 관리 프로그램, 데이터베이스 및 응용프로그램의 스냅샷을 리턴합니다.
- 4** 데이터베이스 관리 프로그램, 데이터베이스, 응용프로그램 및 명령문의 스냅샷을 리턴합니다(후자는 자동 커미트가 OFF이고 명령문 블록이 아닌 단일 명령문이 처리되고 있는 경우에만 리턴됨). 스냅샷에는 해시 조인 정보가 포함되지 않습니다.
- 5** 데이터베이스 관리 프로그램, 데이터베이스, 응용프로그램 및 명령문의 스냅샷을 리턴합니다(후자는 자동 커미트가 OFF이고 명령문 블록이 아닌 단일 명령문이 처리되고 있는 경우에만 리턴됨). 또한 버퍼 풀, 테이블 스페이스 및 FCM의 스냅샷을 리턴합니다(FCM 스냅샷은 다중 데이터베이스 파티션 환경에서만 사용 가능함). 스냅샷에는 해시 조인 정보가 포함되지 않습니다.

디폴트값은 1입니다. >1인 값은 DB2 버전 2 및 DB2 데이터베이스 서버에서만 유효하며 현재 호스트 머신에서는 지원되지 않습니다.

## **ERROR\_STOP**

중요하지 않은 오류가 발생할 때 db2batch가 실행을 중지할지 여부를 지정합니다. 가능한 값은 다음과 같습니다.

- no** 중요하지 않은 오류가 발생할 때 계속 실행합니다. 디폴트 옵션입니다.
- yes** 중요하지 않은 오류가 발생할 때 실행을 중지합니다.

## **DELIMITER**

1문자 또는 2문자의 명령문의 끝 분리문자입니다. 디폴트값은 세미콜론(; )입니다.

## **SLEEP**

유휴할 시간(초)입니다. 유효 값은 1부터 *n*까지입니다.

## **PAUSE**

사용자가 계속하도록 프롬프트합니다.

## SNAPSHOT *snapshot*

작성할 모니터링 스냅샷을 지정합니다. 작성할 수 있는 스냅샷은 `-mss` 옵션을 참조하십시오.

## TIMESTAMP

시간소인을 생성합니다.

## TIMING

시간 제어 정보를 인쇄합니다. 가능한 값은 다음과 같습니다.

**ON** 시간 제어 정보가 인쇄됩니다. 이는 디폴트값입니다.

**OFF** 시간 제어 정보가 인쇄되지 않습니다.

## **-a** *userid/passwd*

데이터베이스에 연결하는 데 사용되는 사용자 ID와 암호를 지정합니다. 슬래시 (/)가 포함되어야 합니다.

## **-m** *parameters\_file*

명령문을 실행하기 전에 SQL문 매개변수 표시문자에 바인드할 매개변수 값을 갖는 입력 파일을 지정합니다. 디폴트는 매개변수를 바인드하지 않는 것입니다.

매개변수 파일이 사용되는 경우 각 라인이 블록의 주어진 명령문 및 주어진 반복에 대한 매개변수 값을 지정합니다. 대신 #PARAM 지시문이 사용되는 경우, 각 명령문의 각 매개변수에 대해 복수의 값과 짝수 매개변수 범위가 미리 지정되고 블록의 각 반복에서 무작위 값이 각 매개변수에 대해 지정된 세트에서 선택됩니다. #PARAM 지시문과 매개변수 파일은 혼합할 수 없습니다.

매개변수 값 형식:

-36.6	'DB2'	X'0AB2'	G'...'	NULL	12
'batch'	x'32ef'	N'...'	null		
+1.345E-6	'db2 batch'	X'afd4'	g'...'	Null	

각 매개변수는 SQL 상수처럼 정의되며, 공백을 사용하여 다른 매개변수와 구분됩니다. 구분되지 않는 텍스트는 숫자를 표시하고, 일반 구분된(') 텍스트는 1바이트 문자열을 표시하며, 작은따옴표(')로 묶은 'x' 또는 'X' 접두부 텍스트는 16진 숫자의 쌍으로 인코딩되는 실행 파일 문자열을 표시하며, 작은따옴표(')로 묶은 'g', 'G', 'n' 또는 'N' 접두부 텍스트는 2바이트 문자로 구성된 그래픽 문자열을 표시하며, 'NULL'(대소문자 구분 없음)은 널(NULL) 값을 표시합니다. XML 데이터를 지정하려면 구분(') 텍스트를 사용하십시오(예: '<last>Brown</last>').

매개변수 입력 파일 형식:

라인 X는 입력 파일에서 실행되는 X번째 SQL문을 제공할 매개변수 세트를 나열합니다. 명령문 블록이 반복되지 않는 경우 이것은 입력 파일에 나열되는 X

번째 명령문에 해당합니다. 공백 라인은 대응하는 SQL문에 대한 매개변수 없음을 표시합니다. 매개변수 수 및 유형이 SQL문에서 예상되는 매개변수 수 및 유형과 일치해야 합니다.

매개변수 지시문 형식:

```
--#PARAM [single | start:end | start:step:end] [...]
```

각 매개변수 지시문은 쿼리의 각 실행에 대해 하나의 무작위 값이 선택되는 매개변수 값 세트를 지정합니다. 세트는 단일 매개변수 값과 매개변수 값 범위 둘 다를 구성됩니다. 매개변수 값 범위는 공백을 선택적 분리 문자로 사용하여 두 유효한 매개변수 값 사이에 콜론(:)을 배치하여 지정됩니다. 세 번째 매개변수 값이 시작 및 종료 값 사이에 위치하여 디폴트를 대체하는 단계 크기로 사용될 수 있습니다. 각 매개변수 범위는 'start', 'start+step', 'start+2\*step', ... 'start+n\*step'의 단일 값을 지정하는 것과 동등합니다. 여기서  $n$ 은 'start+n\*step'  $\geq$  'end'이지만 'start+(n+1)\*step' > 'end'이도록 선택됩니다. 매개변수 지시문을 사용하여 모든 유형의 매개변수(NULL 포함)에 대한 값 세트를 지정할 수 있지만, 숫자 매개변수 값(정수 및 10진수)에서만 범위가 지원됩니다.

#### **-t delcol**

단일 문자 컬럼 분리문자를 지정합니다. -t TAB(탭 컬럼 분리문자) 또는 -t SPACE(스페이스 컬럼 분리문자)를 지정하십시오. 디폴트로, -q on 옵션이 설정되면 스페이스가 사용되고, -q del 옵션이 설정되면 쉼표가 사용됩니다.

#### **-r result\_file**

쿼리 결과를 포함할 출력 파일입니다. 선택적 *summary\_file*이 지정되면 요약 테이블이 포함됩니다. 디폴트는 표준 출력입니다.

**-c** 각 명령문의 결과인 변경을 자동으로 커밋합니다. 디폴트는 ON입니다.

**-i** 경과 시간 간격을 측정할 것을 지정합니다. 가능한 값은 다음과 같습니다.

**short** 각 명령문을 실행할 경과 시간을 측정합니다. 이는 디폴트값입니다.

**long** 명령문 사이의 오버헤드를 포함하여 각 명령문을 실행할 경과 시간을 측정합니다.

#### **complete**

준비, 실행 및 폐치 시간이 개별적으로 보고되는 각 명령문을 실행할 경과 시간을 측정합니다.

**-g** 블록 또는 명령문이 시간 제어를 보고할지 여부를 지정합니다. 가능한 값은 다음과 같습니다.

**on** 전체 블록에 대한 스냅샷이 작성되며 요약 테이블에서 블록 시간 제어만 보고됩니다. 이는 디폴트값입니다.

**off** 스냅샷이 작성되고 블록에서 실행되는 각 명령문에 대해 요약 테이블 시간 제어가 보고됩니다.

- w 결과 세트의 최대 컬럼 너비를 지정하며, 허용 가능한 범위는 0부터 2G까지입니다. 데이터를 절단할 수 없는 경우가 아니면 데이터는 표시될 때 이 너비로 절단됩니다. 이 설정을 늘려서 경고 CLI0002W를 제거하고 보다 정확한 페치 시간을 얻을 수 있습니다. 디폴트 최대 너비는 32768 컬럼입니다.
- time 시간 제어 정보 보고 여부를 지정합니다. 가능한 값은 다음과 같습니다.
  - on 시간 제어가 보고됩니다. 이는 디폴트값입니다.
  - off 시간 제어가 보고되지 않습니다.
- cli 이전에 db2batch 명령에 대한 디폴트 모드인 임베디드(embedded) 동적 SQL 모드가 더 이상 지원되지 않습니다. 이 명령은 CLI 모드에서만 실행됩니다. -cli 옵션은 이전 버전과의 호환성을 위해 존재합니다. 이 옵션을(선택적 cache-size 인수를 포함하여) 지정해도 오류가 유발되지 않지만, 내부적으로 무시됩니다.
- msw *switch*

지정된 각 모니터 스위치의 상태를 설정합니다. uow, statement, table, bufferpool, lock, sort 및 timestamp 중 하나를 지정할 수 있습니다. 특수 스위치 all은 위의 모든 스위치를 설정합니다. 사용자가 지정하는 각 스위치에 대해 다음 중 하나를 선택해야 합니다.

  - hold 스위치 상태가 변경되지 않습니다. 이는 디폴트값입니다.
  - on 스위치가 켜집니다.
  - off 스위치가 꺼집니다.
- mss *snapshot*

각 명령문이나 블록이 실행된 후 -g 옵션에 따라서 작성되어야 하는 모니터링 스냅샷을 지정합니다. 인쇄하기 전에 모든 스냅샷의 정보를 하나의 큰 테이블로 결합하여 둘 이상의 스냅샷을 한 번에 작성할 수 있습니다. 가능한 스냅샷은 applinfo\_all, dbase\_applinfo, dcs\_applinfo\_all, db2, dbase, dbase\_all, dcs\_dbase, dcs\_dbase\_all, dbase\_remote, dbase\_remote\_all, agent\_id, dbase\_appls, appl\_all, dcs\_appl\_all, dcs\_appl\_handle, dcs\_dbase\_appls, dbase\_appls\_remote, appl\_remote\_all, dbase\_tables, appl\_locks\_agent\_id, dbase\_locks, dbase\_tablespace, bufferpools\_all, dbase\_bufferpools 및 dynamic\_sql입니다.

특수 스냅샷 all은 위의 모든 스냅샷을 작성합니다. APPL ID를 포함하는 모든 스냅샷은 동등한 에이전트 ID(응용프로그램 핸들)를 위해 지원되지 않습니다. 디폴트로 모니터링 스냅샷은 작성되지 않습니다.
- iso 데이터가 처리되고 있는 동안 데이터가 잠기고 다른 프로세스로부터 분리되는 방법을 판별하는 분리 레벨을 지정합니다. 디폴트로 db2batch는 RR 분리 레벨을 사용합니다.



db2cli.ini 파일의 TxnIsolation 구성 키워드는 db2batch에 영향을 주지 않습니다. RR 이외의 분리 레벨과 함께 이 명령을 실행하려면 -iso 매개변수를 지정해야 합니다.

**RR** 반복 읽기(ODBC 직렬화 가능). 이는 디폴트값입니다.

**RS** 읽기 안정성(ODBC 반복 읽기)

**CS** 커서 안정성(ODBC RC(Read Committed)).

**UR** 언커미트 읽기(ODBC Read Uncommitted).

#### **-o options**

제어 옵션. 유효한 옵션은 다음과 같습니다.

##### **f rows\_fetch**

응답 세트에서 페치될 행 수입니다. 유효 값은 -1부터 *n*까지입니다. 디폴트값은 -1(모든 행이 페치됨)입니다.

##### **r rows\_out**

출력으로 전송될 페치된 행 수입니다. 유효 값은 -1부터 *n*까지입니다. 디폴트값은 -1(모든 페치된 행이 출력으로 전송됨)입니다.

##### **p perf\_detail**

리턴될 성능 정보 레벨을 지정합니다. 가능한 값은 다음과 같습니다.

- 0** 시간 제어 정보 또는 모니터링 스냅샷을 리턴하지 않습니다.
- 1** 경과 시간만 리턴합니다.
- 2** 경과 시간 및 응용프로그램의 스냅샷을 리턴합니다.
- 3** 경과 시간과, 데이터베이스 관리 프로그램, 데이터베이스 및 응용프로그램의 스냅샷을 리턴합니다.
- 4** 데이터베이스 관리 프로그램, 데이터베이스, 응용프로그램 및 명령문의 스냅샷을 리턴합니다(후자는 자동 커미트가 OFF이고 명령문 블록이 아닌 단일 명령문이 처리되고 있는 경우에만 리턴됨).
- 5** 데이터베이스 관리 프로그램, 데이터베이스, 응용프로그램 및 명령문의 스냅샷을 리턴합니다(후자는 자동 커미트가 OFF이고 명령문 블록이 아닌 단일 명령문이 처리되고 있는 경우에만 리턴됨). 또한 버퍼 풀, 테이블 스페이스 및 FCM의 스냅샷을 리턴합니다(FCM 스냅샷은 다중 데이터베이스 파티션 환경에서만 사용 가능함).

디폴트값은 1입니다. >1인 값은 DB2 버전 2 및 DB2 데이터베이스 서버에서만 유효하며 현재 호스트 머신에서는 지원되지 않습니다.

**o** *query\_optimization\_class*

쿼리 최적화 클래스를 설정합니다. 유효 값은 0, 1, 2, 3, 5, 7 또는 9입니다. 디폴트는 현재 최적화 클래스를 사용하는 -1입니다.

**e** *explain\_mode*

db2batch가 실행되는 EXPLAIN 모드를 설정합니다. Explain 테이블이 이 옵션을 사용하기 전에 작성되어야 합니다. 가능한 값은 다음과 같습니다.

**no** 쿼리만 실행합니다(디폴트).

**explain**

Explain 테이블만 채웁니다. 이 옵션은 Explain 테이블을 채우고 Explain 스냅샷이 작성되도록 합니다.

**yes** Explain 테이블을 채우고 쿼리를 실행합니다. 이 옵션은 Explain 테이블을 채우고 Explain 스냅샷이 작성되도록 합니다.

**s** *error\_stop*

중요하지 않은 오류가 발생할 때 db2batch가 실행을 중지할지 여부를 지정합니다. 가능한 값은 다음과 같습니다.

**no** 중요하지 않은 오류가 발생할 때 계속 실행합니다. 디폴트 옵션입니다.

**yes** 중요하지 않은 오류가 발생할 때 실행을 중지합니다.

**-v** 상세 모드. 쿼리 처리 중에 표준 오류로 정보를 보냅니다. 디폴트값은 OFF입니다.

**-s** 요약 테이블. 산술 및 기하 평균, 폐치된 행 수 및 출력된 행 수를 갖는 경과 시간을 포함하여, 각 쿼리 또는 쿼리 블록에 대한 요약 테이블을 제공합니다.

**-q** 쿼리 출력. 가능한 값은 다음과 같습니다.

**off** 쿼리 결과 및 모든 연관된 정보를 출력합니다. 이는 디폴트값입니다.

**on** 컬럼 식별자 없는 형식으로 쿼리 결과만 출력합니다.

**del** 구분 형식으로 쿼리 결과만 출력합니다.

**-l** *stmt\_delimiter*

종료 문자(명령문 분리문자)를 지정합니다. 분리문자는 1 또는 2문자일 수 있습니다. 디폴트는 세미콜론(';')입니다.

**-h | -u | -?**

도움말 정보를 표시합니다. 이 옵션을 지정하면 다른 모든 옵션은 무시되며 도움말 정보만 표시됩니다.

예:

1. 다음은 db2batch -d crystal -f update.sql 명령의 출력 샘플입니다.

```
* 시간소인: Thu Feb 02 2006 10:06:13 EST
-----

* SQL문 번호 1:

create table demo (c1 bigint, c2 double, c3 varchar(8));

* 경과 시간:      0.101091초
-----

* SQL문 번호 2:

insert into demo values (-9223372036854775808, -0.0000000000000005, 'demo');

* 경과 시간:      0.002926초
-----

* SQL문 번호 3:

insert into demo values (9223372036854775807, 0.0000000000000005, 'demodemo');

* 경과 시간:      0.005676 seconds
-----

* SQL문 번호 4:

select * from demo;

C1                C2                C3
-----
-9223372036854775808 -5.000000000000000E-015 demo
9223372036854775807 +5.000000000000000E-015 demodemo

* 2행 페치(fetch)됨, 2행 출력됨.

* 경과 시간:      0.001104초
-----

* SQL문 번호 5:

drop table demo;

* 경과 시간:      0.176135초

* 요약 테이블:

유형   번호   반복   총시간(s)   최소 시간(s)   최대 시간(s)
-----
Statement 1       1       0.101091     0.101091     0.101091
Statement 2       1       0.002926     0.002926     0.002926
Statement 3       1       0.005676     0.005676     0.005676
Statement 4       1       0.001104     0.001104     0.001104
Statement 5       1       0.176135     0.176135     0.176135

산술 평균   기하 평균   페치(fetch)된 행   출력된 행
-----
0.101091   0.101091     0                   0
0.002926   0.002926     0                   0
0.005676   0.005676     0                   0
0.001104   0.001104     2                   2
0.176135   0.176135     0                   0

* 총 항목:      5
* 총 시간:      0.286932초
```

\* 최소 시간: 0.001104초  
\* 최대 시간: 0.176135초  
\* 산술 평균 시간: 0.057386초  
\* 기하 평균 시간: 0.012670초

-----  
\* 시간소인: Thu Feb 02 2006 10:06:13 EST

## 사용 시 참고사항

- 모든 SQL문은 `--#SET DELIMITER` 명령으로 설정되는 분리문자(디폴트 `';`)로 종료되어야 합니다. 이 분리문자는 1 또는 2문자일 수 있습니다.
- SQL문 길이는 사용 가능한 메모리 및 사용되는 인터페이스에 의해서만 제한됩니다. 명령문을 여러 라인으로 분할할 수 있지만, 한 라인에 여러 명령문은 허용되지 않습니다.
- 입력 파일 라인 길이는 사용 가능 메모리에 의해서만 제한됩니다.
- `c`는 자동으로 `CONNECT` 및 `CONNECT RESET`문을 발행합니다.
- `PAUSE` 및 `SLEEP`은 `-i` 시간 제어 옵션에 대해 `long`이 지정될 때 시간 제어됩니다.
- `EXPLAIN` 옵션을 사용할 수 있으려면 먼저 `Explain` 테이블을 작성해야 합니다.
- 모든 명령행 옵션 및 입력 파일 명령문은 `db2batch`에 관해 대소문자를 구분하지 않습니다.
- `db2batch`는 데이터 유형 `INTEGER`, `CHAR`, `VARCHAR`, `LONG VARCHAR`, `FLOAT`, `SMALLINT`, `BIGINT`, `DECIMAL`, `DATE`, `TIME`, `TIMESTAMP`, `CLOB`, `GRAPHIC`, `VARGRAPHIC`, `LONGVARGRAPHIC`, `DBCLOB`, `BLOB` 및 `XML`을 지원합니다.
- `--#SET PERF_DETAIL perf_detail`(또는 `-o p perf_detail`)은 모니터링 출력을 얻는 빠른 방법을 제공합니다. 성능 세부사항 레벨이 > 1인 경우, 모든 모니터 스위치는 내부적으로 `db2batch`에 의해 켜집니다. 더 정확한 모니터링 출력 제어가 필요한 경우 `-msw` 및 `-mss`(또는 `--#SET SNAPSHOT`) 옵션을 사용하십시오.

## 제 166 장 db2acsutil - DB2 스냅샷 백업 오브젝트 관리 명령

db2acsutil을 사용하면 다음과 같은 세 가지 방법으로 DB2 스냅샷 백업 오브젝트를 관리할 수 있습니다.

1. 데이터베이스를 리스토어하기 위해 사용할 수 있는 DB2 스냅샷 백업 나열
2. BACKUP 명령, db2Backup API 또는 BACKUP DATABASE 매개변수와 ADMIN\_CMD 스토어드 프로시저를 함께 사용하여 생성된 DB2 스냅샷 백업 삭제
3. DB2 스냅샷 백업 상태 모니터

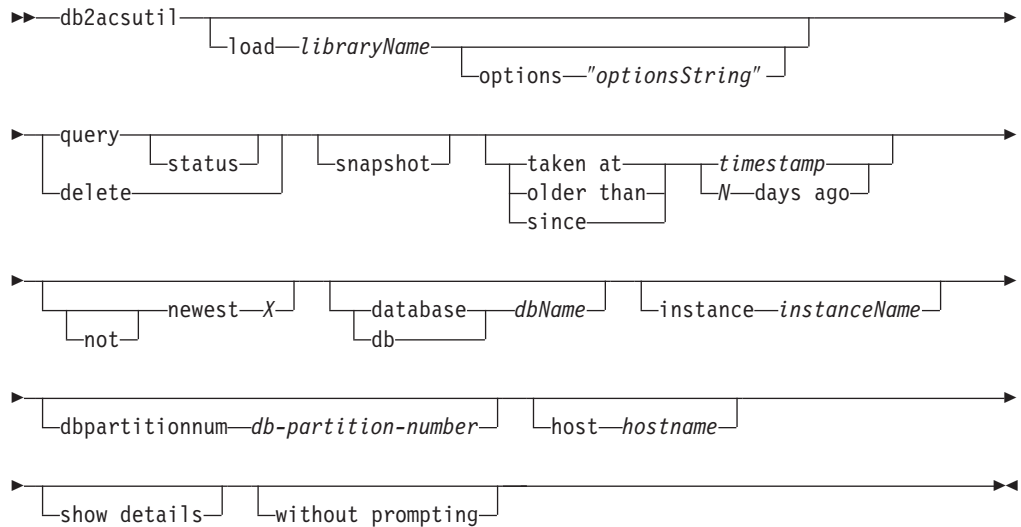
### 권한 부여

없음

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

#### load libraryName

DB2 스냅샷 백업에서 사용되는 벤더 고속 복사 기술이 들어 있는 공유 라이브러리의 이름. 이 매개변수는 전체 경로를 포함할 수 있습니다. 전체 경로가 지정되지 않는 경우 디폴트 경로는 BACKUP DB 및 RESTORE DB 명령을 사용하는 경우와 같은 라이브러리입니다(~/sqllib/acs).

**options** *"optionsString"*

이 유틸리티에 사용될 옵션을 지정합니다. 문자열은 큰따옴표 없이 입력된 시점과 동일한 상태로 벤더 지원 라이브러리로 전달됩니다.

**query**

ACS 저장소를 쿼리하고 알려진 오브젝트 테이블을 리턴합니다.

**status**

ACS 저장소를 쿼리하고 알려진 오브젝트의 테이블과 오브젝트의 현재 상태를 리턴합니다.

**delete**

DB2 스냅샷 오브젝트를 삭제하고 ACS 저장소에서 삭제된 해당 레코드를 제거합니다.

**snapshot**

스냅샷 오브젝트로만 리턴 또는 실행된 레코드를 필터링합니다.

**taken at | older than | since**

이러한 옵션은 유틸리티 결과를 지정된 시간 범위로 필터링합니다.

**시간소인**

YYYYMMDDhhmmss 양식의 시간소인.

***N* days ago**

경과 일 수. 여기서 *N*은 현재 날짜 이전의 일 수입니다.

**[not] newest *X***

최신(시간소인 기준) *X*개 레코드만 고려하도록 유틸리티 결과를 필터링합니다. NOT 키워드가 지정되면 최신 *X*개를 제외한 모든 레코드를 고려합니다.

**database | db *dbName***

지정된 데이터베이스 이름과 연관된 오브젝트만 고려합니다.

**instance *instanceName***

관리하는 DB2 스냅샷 백업 오브젝트와 연관된 데이터베이스 관리 프로그램 인스턴스의 이름.

**dbpartitionnum *db-partition-number***

지정된 데이터베이스 파티션 번호로 작성된 오브젝트만 고려합니다.

**host *hostname***

지정된 *hostname*으로 작성된 오브젝트만 고려합니다. 예를 들어, 일반적으로 DB2 서버의 TCP/IP 호스트 이름입니다.

**show details**

ACS 저장소의 세부 오브젝트 정보를 표시합니다. 이 옵션을 사용하는 경우 라인당 간략한 단일 레코드가 있는 테이블 대신 각 ACS 오브젝트마다 상세 스태자가 생성됩니다.

### without prompting

유틸리티가 자동으로 실행되고 일반적으로 사용자 개입이 필요한 모든 조치는 오류 메시지를 리턴하는 것으로 지정합니다.

### 예:

사용 중인 백그라운드 사본이 있는 스냅샷 백업의 샘플 출력.

```
db2acsutil query status db f01 instance db2inst1 dbpartitionnum 0
```

```

Instance  Database  Part Image Time      Status
=====  =====  ==  =====  =====
keon14    F01        0   20070719120848  Remotely mountable + Background_monitor
                                                pending (16 / 1024 MB)

```

완료된 백그라운드 사본이 있는 스냅샷 백업의 샘플 출력.

```
db2acsutil query status db f01 instance db2inst1 dbpartitionnum 0 show details
```

```

Instance : keon14
Database : F01
Partition : 0
Image timestamp : 20070719120848
Host : machine1
Owner :
DB2 Version : 9.5.0
Creation time : Thu Jul 19 12:08:50 2007
First active log (chain:file) : 0:0
Metadata bytes : 6196
Progress state : Successful
Usability state : Remotely mountable + Repetitively restorable + Swap restorable
                  + Physical protection + Full copy
Bytes completed : 0
Bytes total : 0

```

## 사용 시 참고사항

db2acsutil 사용은 BACKUP 명령, db2Backup API 또는 BACKUP DATABASE 매개변수와 ADMIN\_CMD 스토어드 프로시저를 함께 사용하여 작성된 DB2 스냅샷 백업을 삭제할 수 있는 유일한 방법입니다. 자동 복구 오브젝트 삭제 또는 PRUNE HISTORY 명령과 AND DELETE 매개변수를 함께 사용하여 DB2 스냅샷 백업을 삭제할 수 없습니다. 파일러/스토리지 시스템을 통해 수동으로 백업을 삭제할 수도 없습니다.

DB2 스냅샷 백업의 사용 편리성 상태는 DB2 스냅샷으로 수행 가능한 작업을 나타냅니다. 테이블 1에는 가능한 DB2 스냅샷 백업 사용 편리성 상태가 설명되어 있습니다.

표 37. DB2 스냅샷 백업에 대해 리턴되는 사용 편리성 상태

사용 편리성 상태	설명
LOCALLY_MOUNTABLE	로컬 머신에서 백업한 데이터를 마운트할 수 있습니다.
REMOTEY_MOUNTABLE	리모트 머신에서 백업한 데이터를 마운트할 수 있습니다.

표 37. DB2 스냅샷 백업에 대해 리턴되는 사용 편리성 상태 (계속)

사용 편리성 상태	설명
REPETITIVELY_RESTOREABLE	DB2 스냅샷 백업 이미지를 여러 번 사용하여 백업된 데이터를 리스토어할 수 있습니다.
DESTRUCTIVELY_RESTOREABLE	DB2 스냅샷 백업 이미지를 사용하여 한 번 백업된 데이터를 리스토어할 수 있습니다. 백업된 데이터가 리스토어되면 이 DB2 스냅샷 이미지(다른 이미지도 가능)가 파괴됩니다.
SWAP_RESTOREABLE	볼륨에 직접 액세스할 수는 있지만 RESTORE DB 명령을 실행할 수 없으며 백업된 데이터를 소스 볼륨에 직접 복사할 수 없습니다.
PHYSICAL_PROTECTION	소스 볼륨의 실제 실패로부터 스냅샷을 보호합니다.
FULL_COPY	데이터의 전체 사본이 작성되었습니다. DB2 스냅샷 백업 이미지를 사용하여 백업된 데이터를 리스토어할 수 있습니다.
DELETED	삭제할 백업이 표시되었음을 나타냅니다. DELETED 백업과 관련된 스냅샷 스토리지는 백그라운드에서 실행되는 유지보수 프로세스를 통해 취소됩니다. 이 작업이 완료되면 ACS 저장소에서 백업이 제거됩니다.
FORCED_MOUNT	AIX JFS 파일 시스템을 마운트하여 파일 시스템 일관성 검증을 대기합니다.
BACKGROUND_MONITOR_PENDING	ACS 백그라운드 진행 모니터가 상태를 모니터링합니다.
TAPE_BACKUP_PENDING	오프로드된 테이프 백업을 대기합니다.
TAPE_BACKUP_IN_PROGRESS	오프로드된 테이프 백업이 현재 진행 중입니다.
TAPE_BACKUP_COMPLETE	오프로드된 테이프 백업이 완료되었습니다.



---

## 제 167 장 db2bfd - 바인드 파일 설명 도구

바인드 파일의 콘텐츠를 표시합니다. 바인드 파일의 SQL문을 조사하고 검증할 뿐 아니라 바인드 파일을 작성하는 데 사용된 프리컴파일 옵션을 표시하는 데 사용할 수 있는 이 유틸리티는 응용프로그램의 바인드 파일과 관련된 문제점 판별에서 유용할 수 있습니다.

### 권한 부여

없음

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

- h 도움말 정보를 표시합니다. 이 옵션을 지정하면 다른 모든 옵션은 무시되며 도움말 정보만 표시됩니다.
- b 바인드 파일 헤더를 표시합니다.
- s SQL문을 표시합니다.
- v 호스트 변수 선언을 표시합니다.

*filespec*

컨텐츠를 표시할 바인드 파일의 이름입니다.



## 제 168 장 db2ca - 구성 지원 프로그램 시작

구성 지원 프로그램을 시작합니다. 구성 지원 프로그램은 DB2 데이터베이스 구성(예를 들어, 데이터베이스 관리 프로그램 구성, DB2 레지스트리, 노드 디렉토리, 데이터베이스 디렉토리 및 DCS 디렉토리)을 관리하는 데 사용되는 그래픽 인터페이스입니다.

**중요사항:** 이 명령은 제어 센터와 관련된 구성요소가 버전 9.7에서 사용되지 않기 때문에 사용되지 않으며 이후 릴리스에서 제거됩니다. 자세한 정보는 *버전 9.7의 새로운 내용* 책에 있는 『제어 센터 도구 및 DB2 Administration Server(DAS)는 사용되지 않음』 주제를 참조하십시오.

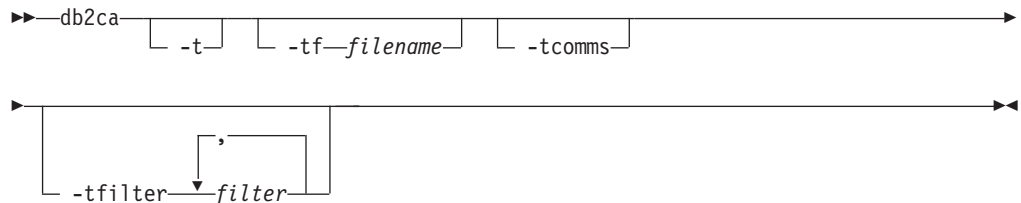
### 권한 부여

*sysadm*

### 필요한 연결

없음

### 명령 구문



### 명령 매개변수

**-t** GUI 추적을 활성화하고 콘솔 창으로 출력을 보냅니다. Windows 운영 체제의 경우에는 `db2ca` 명령에 콘솔 창이 없습니다. 따라서 Windows 운영 체제에서는 이 옵션이 영향을 주지 않습니다.

#### **-tf filename**

GUI 추적을 활성화하고 추적 결과를 지정된 파일에 저장합니다. 출력 파일은 `<DB2 install path>\sqllib\tools`(Windows 운영 체제)와 `/home/<userid>/sqllib/tools`(Linux 및 UNIX 시스템)에 저장됩니다.

#### **-tcomms**

통신 이벤트로 추적을 제한합니다.

#### **-tfilter filter**

지정된 하나 이상의 필터가 포함된 항목으로 추적을 제한합니다.



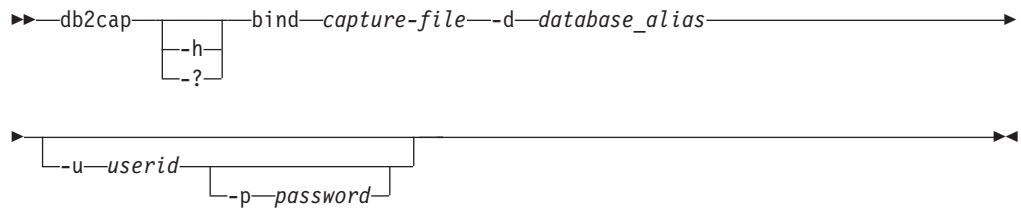
## 제 169 장 db2cap - CLI/ODBC 정적 패키지 바인딩 도구

하나 이상의 정적 패키지를 생성하기 위해 캡처 파일을 바인드합니다. 캡처 파일은 CLI/ODBC/JDBC/.NET 응용프로그램의 정적 프로파일링 세션 중에 생성되며, 응용프로그램 실행 중에 캡처된 SQL문이 포함됩니다. 이 유틸리티는 CLI/ODBC/JDBC/.NET 드라이버가 응용프로그램에 대한 정적 SQL을 실행하기 위해 사용할 수 있도록 캡처 파일을 처리합니다.

### 권한 부여

- 캡처 파일에 기록되는 SQL문이 참조하는 모든 데이터베이스 오브젝트에 대한 액세스 특권.
- db2cap 명령을 호출하는 데 사용되는 연결 ID와는 다른 경우 OWNER 및 QUALIFIER 같은 바인드 옵션을 설정할 충분한 권한.
- 패키지가 처음으로 바인드되는 경우 BINDADD 권한. 그렇지 않으면 BIND 권한이 필요합니다.

### 명령 구문



### 명령 매개변수

#### -h | -?

명령 구문에 대한 도움말 텍스트를 표시합니다.

#### bind capture-file

캡처 파일의 명령문을 바인드하고 하나 이상의 패키지를 작성합니다. 이 캡처 파일은 .NET용 pureQueryXML 파일이라고도 합니다.

#### -d database\_alias

하나 이상의 패키지를 포함할 데이터베이스에 대한 데이터베이스 별명을 지정합니다.

#### -u userid

데이터 소스에 연결하는 데 사용될 사용자 ID를 지정합니다. 사용자 ID가 지정되지 않은 경우 트러스트된 권한 부여 ID는 시스템에서 얻습니다.

**-p password**

데이터 소스에 연결하는 데 사용될 암호를 지정합니다.

## 사용 시 참고사항

이 명령은 UNIX 플랫폼에서 소문자로 입력해야 하지만, Windows 운영 체제에서 소문자 또는 대문자로 입력할 수 있습니다. .NET 응용프로그램에 대한 정적 패키지 바인딩은 Windows 운영 체제에서만 지원됩니다.

이 유틸리티는 캡처 파일에서 찾을 수 있는 많은 사용자 지정 바인드 옵션을 지원합니다. 바인드 옵션을 변경하려면 텍스트 편집기에서 캡처 파일을 여십시오.

SQLERROR(CONTINUE) 및 VALIDATE(RUN) 바인드 옵션을 사용하여 패키지를 작성할 수 있습니다.

이 유틸리티를 사용하여 패키지를 작성할 때 정적 프로파일링을 사용하지 않아야 합니다.

작성되는 패키지 수는 캡처 파일에 기록되는 SQL문에 대해 사용되는 분리 레벨에 따라 다릅니다. 패키지 이름은 캡처 파일에서 패키지 키워드의 처음 7자까지 및 다음의 단일 문자 접미부 중 하나로 구성됩니다.

- 0 - 언커미트 읽기(UR)
- 1 - 커서 안정성(CS)
- 2 - 읽기 안정성(RS)
- 3 - 반복 읽기(RR)
- 4 - 커미트 안함(NC)

패키지에 관한 특정 정보를 알기 위해 사용자는 다음을 수행할 수 있습니다.

- 캡처 파일에서 발견되는 COLLECTION 및 PACKAGE 키워드를 사용하여 적합한 SYSIBM 카탈로그 테이블을 쿼리합니다.
- 캡처 파일을 봅니다.

---

## 제 170 장 db2cat - 시스템 카탈로그 분석

팩형 디스크립터의 콘텐츠를 분석합니다. 데이터베이스 이름과 기타 규정 정보를 지정하면 이 명령이 시스템 카탈로그에서 정보를 쿼리하고 결과를 형식화합니다. 이 명령은 서버에서 실행되어야 합니다.

### 권한 부여

없음

### 필요한 연결

없음

### 명령 구문

```
db2cat [-d dbname] [-h] [-l] [-n name] [-o outfile]
        [-p descriptor] [-vi versionID] [-s schema] [-t] [-z]
        [-v] [-x] [-cb]
```

### 명령 매개변수

#### **-d** *dbname*

*dbname*은 명령이 시스템 카탈로그를 쿼리할 데이터베이스의 이름입니다.

**-h** 사용법 정보를 표시합니다.

**-l** 오브젝트 이름의 대소문자를 구분합니다.

#### **-n** *name*

오브젝트의 이름을 지정합니다.

#### **-o** *outfile*

출력 파일의 이름을 지정합니다.

#### **-p** *descriptor*

*descriptor*가 다음 중 하나인 경우 표시할 팩형 디스크립터(pd)의 이름을 지정합니다.

**check** 테이블 점검 제한조건 팩형 디스크립터를 표시합니다.

**rel** 참조 무결성 제한조건 팩형 디스크립터를 표시합니다.

**table** 테이블 팩형 디스크립터를 표시합니다. 테이블에 대해 하나 이상이 존재하는 경우 인라인 길이가 포함됩니다.

**summary**

요약 테이블 팩형 디스크립터를 표시합니다.

**trig** 테이블 트리거 팩형 디스크립터를 표시합니다.

**view** 뷰 팩형 디스크립터를 표시합니다.

**remote**

리모트 비관계형 데이터 소스 팩형 디스크립터를 표시합니다.

**ast** 구체화된 쿼리 테이블 팩형 디스크립터를 표시합니다.

**routine**

루틴 팩형 디스크립터를 표시합니다.

**sysplan**

패키지 팩형 디스크립터를 표시합니다.

**datatype**

구조화된 유형 팩형 디스크립터를 표시합니다.

**sequence**

시퀀스 팩형 디스크립터를 표시합니다.

**esri** 키 변환 스테드 및 인덱스 확장자 팩형 디스크립터를 표시합니다.

**event** 이벤트 모니터 팩형 디스크립터를 표시합니다.

**server** 서버 팩형 디스크립터를 표시합니다.

**auth** 이 오브젝트에 대해 이 권한 받은 사용자가 보유하는 특권을 표시합니다.

**-vi** *versionID*

패키지 팩형 디스크립터의 버전 ID를 지정합니다. **-vi**는 **-p sysplan**이 지정된 경우에만 유효합니다. *versionID*가 생략되는 경우 디폴트값은 비어 있는 문자열입니다.

**-s** *schema*

오브젝트 스키마의 이름을 지정합니다.

**-t** 터미널 출력을 표시합니다.

**-z** 키 입력 프롬프트를 사용하지 않습니다.

**-v** 팩형 디스크립터의 유효성을 확인합니다. 이 매개변수는 테이블 팩형 디스크립터에만 유효합니다.

**-x** 카탈로그에서 테이블 스페이스 Extent 크기의 유효성을 확인합니다(테이블 이름은 필요하지 않음).



- cb SYSCAT.BUFFERPOOLNODES에서 상위 항목이 삭제된 오브젝트 행을 지웁니다 (테이블 이름이 필요하지 않음).

## 사용 시 참고사항

- 테이블 이름과 테이블 스키마는 LIKE 술어 양식으로 제공될 수 있습니다. 이 양식에서는 하나의 호출로 여러 소스를 선택하기 위한 패턴 일치 문자로 퍼센트 기호(%)와 밑줄(\_)을 사용할 수 있습니다.
- 제공되지 않거나 불완전하게 지정된 모든 필드에 대해 프롬프트가 나타납니다(-h 및 -l 옵션 제외).
- -o가 파일 이름 없이 지정되고 -t가 지정되지 않으면 파일 이름에 대한 프롬프트가 표시됩니다(디폴트 이름은 db2cat.out).
- -o와 -t가 모두 지정되지 않으면 파일 이름에 대한 프롬프트가 표시됩니다(디폴트값은 터미널 출력).
- -o와 -t가 모두 지정되는 경우 출력 방향은 터미널입니다.



## 제 171 장 db2cc - 제어 센터 시작

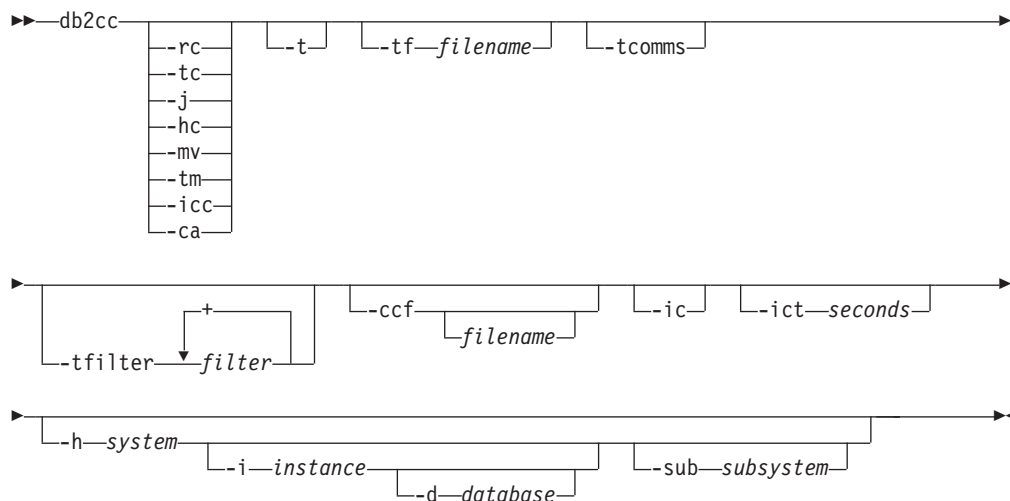
제어 센터를 시작합니다. 제어 센터는 데이터베이스 오브젝트(예: 데이터베이스, 테이블 및 패키지) 및 서로 간의 관계를 관리하는 데 사용되는 그래픽 인터페이스입니다.

**중요사항:** 이 명령은 제어 센터와 관련된 구성요소가 버전 9.7에서 사용되지 않기 때문에 사용되지 않으며 이후 릴리스에서 제거됩니다. 자세한 정보는 *버전 9.7의 새로운 내용* 책에 있는 『제어 센터 도구 및 DB2 Administration Server(DAS)는 사용되지 않음』 주제를 참조하십시오.

### 권한 부여

*sysadm*

### 명령 구문



### 명령 매개변수

- rc** 복제 센터를 엽니다.
- hc** Health Center를 엽니다.
- tc** 태스크 센터를 엽니다.
- j** 저널을 엽니다.
- mv** Memory Visualizer를 엽니다.
- tm** 인다우트(Indoubt) 트랜잭션 식별 프로그램을 엽니다.
- icc** Information Catalog Manager를 엽니다.
- ca** 구성 지원 프로그램을 엽니다.

- t 초기화 코드에 대한 제어 센터 추적을 켭니다. Windows 운영 체제에서 db2cc 명령은 콘솔 창이 없습니다. 그러므로 이 옵션은 Windows 운영 체제에 영향을 미치지 않습니다.
- tf 초기화 코드에 대한 제어 센터 추적을 켜고 추적 결과를 지정된 파일에 저장합니다. 출력 파일은 <DB2 install path>#sqllib#tools(Windows) 및 /home/<userid>/sqllib/tools(UNIX 운영 체제)에 저장됩니다.
- tcomms  
추적을 통신 이벤트로 제한합니다.
- tfilter *filter*  
추적을 지정된 필터가 있는 항목으로 제한합니다.
- ccf *filename*  
명령 편집기를 엽니다. 파일 이름이 지정되는 경우 이 파일의 콘텐츠가 명령 편집기의 스크립트 페이지에 로드됩니다. 파일 이름을 지정할 때 파일에 대한 절대 경로를 제공해야 합니다.
- ic 정보 센터를 엽니다.
- ict *seconds*  
유휴 연결 타이머. 지정된 시간(초) 후에 제어 센터가 유지하는 풀의 모든 유휴 연결을 닫습니다. 디폴트 타이머는 30분입니다.
- h *system*  
시스템의 컨텍스트에서 제어 센터를 엽니다.
- i *instance*  
인스턴스의 컨텍스트에서 제어 센터를 엽니다.
- d *database*  
데이터베이스의 컨텍스트에서 제어 센터를 엽니다.
- sub *subsystem*  
서브시스템의 컨텍스트에서 제어 센터를 엽니다.

## 제 172 장 db2cfexp - 연결성 구성 익스포트 도구

연결성 구성 정보를 익스포트 프로파일에 익스포트하며, 이는 나중에 비슷한 인스턴스 유형의 다른 DB2 데이터베이스 워크스테이션 인스턴스에서 임포트될 수 있습니다(즉, 클라이언트 인스턴스에서 클라이언트 인스턴스로). 결과 프로파일에는 현재 DB2 데이터베이스 인스턴스와 연관된 구성 정보만 들어 있습니다. 이 프로파일을 클라이언트 구성 프로파일 또는 인스턴스의 구성 프로파일로 참조할 수 있습니다.

이 유틸리티는 연결성 구성 정보를 구성 프로파일로 알려진 파일로 익스포트합니다. 지정된 익스포트 옵션의 요구사항을 충족하기 위해 필요한 모든 구성 정보를 패키징하는 비대화형 유틸리티입니다. 익스포트할 수 있는 항목은 다음과 같습니다.

- 데이터베이스 정보(DCS 및 ODBC 정보 포함)
- 노드 정보
- 프로토콜 정보
- 데이터베이스 관리 프로그램 구성 설정
- 레지스트리 설정
- 일반 ODBC/CLI 설정.

이 유틸리티는 특히 DB2 구성 지원 프로그램이 설치되지 않은 워크스테이션과, 여러 개의 비슷한 리모트 DB2 클라이언트가 설치, 구성 및 유지보수되는(예를 들어, 클라이언트 구성의 템플릿 복제 또는 작성) 상황에서 연결성 구성 정보를 익스포트할 때 유용합니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*

### 명령 구문

```
▶▶ db2cfexp filename [TEMPLATE | BACKUP | MAINTAIN] ▶▶
```

### 명령 매개변수

*filename*

목표 익스포트 파일의 완전한 이름을 지정합니다. 이 파일을 구성 프로파일이라고 합니다.

## TEMPLATE

동일한 인스턴스 유형의 다른 인스턴스에 대한 템플릿으로 사용되는 구성 프로파일을 작성합니다(즉, 클라이언트 인스턴스에서 클라이언트 인스턴스로). 프로파일에는 다음에 관한 정보가 포함되어 있습니다.

- 관련 ODBC 및 DCS 정보를 포함한 모든 데이터베이스
- 익스포트된 데이터베이스와 연관된 모든 노드
- 일반 ODBC/CLI 설정
- 데이터베이스 관리 프로그램 구성에서의 일반 클라이언트 설정
- DB2 레지스트리의 일반 클라이언트 설정

## BACKUP

로컬 백업 목적을 위한 DB2 데이터베이스 인스턴스의 구성 프로파일을 작성합니다. 이 프로파일에는 이 로컬 인스턴스에만 관련된 특정 특성의 정보를 포함하여 모든 인스턴스 구성 정보가 들어 있습니다. 프로파일에는 다음에 관한 정보가 포함되어 있습니다.

- 관련 ODBC 및 DCS 정보를 포함한 모든 데이터베이스
- 익스포트된 데이터베이스와 연관된 모든 노드
- 일반 ODBC/CLI 설정
- 데이터베이스 관리 프로그램 구성의 모든 설정
- DB2 레지스트리의 모든 설정
- 모든 프로토콜 정보.

## MAINTAIN

다른 인스턴스 유지보수 또는 갱신에 대한 데이터베이스 및 노드 관련 정보만 들어있는 구성 프로파일을 작성합니다.

## 주:

db2cfexp 명령은 클라이언트에서 파일 데이터 소스 위치 정보를 익스포트하지 않습니다.

디폴트 위치를 사용하는 경우 조치가 필요하지 않습니다. 클라이언트에서 디폴트 위치를 변경하는 경우 연결성 정보를 익스포트할 때 이 위치를 수동으로 복사해야 합니다.

한 클라이언트에서 다른 클라이언트로 파일 데이터 소스 위치를 복사하려면 다음을 수행하십시오.

1. 연결성 정보를 익스포트하려는 클라이언트에서 %DB2PATH%\TOOLS 디렉토리를 찾으십시오.
2. CA.properties 파일을 복사하십시오.

3. 연결성 정보를 임포트하려는 클라이언트에서 %DB2PATH%\TOOLS 디렉토리를 찾으십시오.
  4. 기존 CA.properties 파일을 원래 클라이언트에서 작성한 사본으로 겹쳐쓰십시오.
- 한 클라이언트에서 다른 클라이언트로 파일 데이터 소스 위치를 중복했습니다.





---

## 제 173 장 db2cfimp - 연결성 구성 импорт 도구

구성 프로파일로 알려진 파일에서 연결성 구성 정보를 импорт합니다. 구성 프로파일에서 발견되는 모든 정보를 импорт하려고 시도하는 비대화형 유틸리티입니다.

구성 프로파일은 다음과 같은 연결성 항목을 포함할 수 있습니다.

- 데이터베이스 정보(DB2 Connect 및 ODBC 정보 포함)
- 노드 정보
- 프로토콜 정보
- 데이터베이스 관리 프로그램 구성 설정
- DB2 데이터베이스 레지스트리 설정
- 일반 ODBC/CLI 설정.

이 유틸리티를 사용하면 이전에 구성된 다른 비슷한 인스턴스(즉, 클라이언트 인스턴스 대 클라이언트 인스턴스)에서 연결성 정보를 중복할 수 있습니다. 특히 DB2 구성 지원 프로그램(CA)이 설치되지 않은 워크스테이션과, 여러 개의 비슷한 리모트 DB2 클라이언트가 설치, 구성 및 유지보수되는(예를 들어, 클라이언트 구성의 템플릿 복제 또는 작성) 상황에서 유용합니다. 인스턴스를 복제할 때, импорт된 프로파일은 항상 하나의 DB2 데이터베이스 인스턴스만에 관한 구성 정보가 들어있는 클라이언트 구성 프로파일이어야 합니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*

주:

- 루트 ID는 도구를 실행하는 데 사용해서는 안됩니다.
- 유효한 ID가 도구를 실행하는 데 사용되는 경우 ID는 импорт될 구성 프로파일에 대한 올바른 권한을 가져야 합니다.

### 명령 구문

▶▶—db2cfimp—*filename*—————▶▶

## 명령 매개변수

*filename*

임포트될 구성 프로파일의 완전한 이름을 지정합니다. 유효한 임포트 구성 프로파일은 구성 지원 프로그램, 제어 센터 또는 db2cfexp를 사용하여 모든 DB2 데이터베이스 또는 DB2 Connect 제품에 의해 작성되는 프로파일입니다.

## 제 174 장 db2chglibpath - 임베디드(embedded) 런타임 라이브러리 검색 경로 수정

실행 파일 또는 공유 라이브러리 파일 내 임베디드(embedded) 런타임 라이브러리 검색 경로 값을 수정합니다. 기존 값이 더 이상 유효하지 않은 경우 임베디드(embedded) 런타임 라이브러리 검색 경로 값을 새 사용자 지정 값으로 바꾸는 데 사용될 수 있습니다.

db2chglibpath 명령은 LIBPATH(AIX), SHLIB\_PATH(HPPA, HPIPF) 및 LD\_LIBRARY\_PATH(AIX, SUN, HPPA64, HPIPF 및 Linux)와 같은 운영 체제 라이브러리 검색 경로 환경 변수를 사용하기 위한 요구사항을 대체하는 데 사용할 수 있습니다. 이 명령은 Linux 및 UNIX 운영 체제에서만 지원됩니다. 이 명령은 DB2DIR/bin 디렉토리에 있습니다. 여기서 DB2DIR은 DB2 데이터베이스 설치 위치입니다.

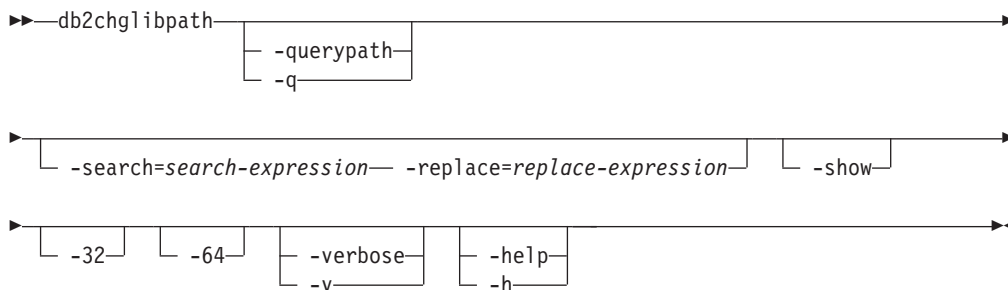
### 전제조건

- 모니터링 공유 라이브러리 또는 실행할 수 있는 파일에는 읽기 및 쓰기 액세스가 필요합니다.
- 실행 파일에는 시작할 임베디드(embedded) 라이브러리 경로가 필요하므로 임베디드(embedded) 경로를 실행 파일에 이미 있는 경로보다 큰 경로로 변경할 수 없습니다.
- 임베디드(embedded) 런타임 라이브러리 검색 경로 값을 교체하는 사용자 지정 값의 길이는 기존 값보다 커서는 안 됩니다.
- 이 명령은 공유 라이브러리 또는 실행할 수 있는 파일의 2진 코드를 직접 수정하므로 이 명령을 사용하기 전에 반드시 파일의 백업을 작성해야 합니다.

### 필요한 연결

없음

### 명령 구문



## 명령 매개변수

### **-querypath**

실행 파일에서 임베디드(embedded) 라이브러리 경로를 변경하지 않고 쿼리가 수행되어야 하는 것으로 지정합니다.

### **-search=***search-expression*

검색할 표현식을 지정합니다.

### **-replace=***replace-expression*

*search-expression*이 교체될 표현식을 지정합니다.

**-show** 변경사항을 실제로 파일에 작성하지 않고 검색 및 교체 조작이 수행되는 것으로 지정합니다.

**-32** 실행 파일 유형이 32비트인 경우 조작을 수행합니다.

**-64** 실행 파일 유형이 64비트인 경우 조작을 수행합니다.

### **-verbose**

수행되는 조작에 대한 정보를 표시합니다.

**-help** 사용법 정보를 표시합니다.

## 예:

- myexecutable 실행 파일에 있는 임베디드(embedded) 런타임 라이브러리 검색 경로 값을 /usr/opt/db2\_08\_01/lib에서 /u/usr1/sql/lib/lib32로 변경하려면 다음을 실행하십시오.

```
db2chglibpath -search=/usr/opt/db2_08_01/lib -replace=/u/usr1/sql/lib/lib32  
/mypath/myexecutable
```

새 값의 길이는 원래 값의 길이와 같습니다.

## 사용 시 참고사항

- 이 명령은 응용프로그램 및 루틴 업그레이드를 위해 다른 방법을 사용할 수 없거나 방법이 실패하는 경우 DB2 데이터베이스 응용프로그램 실행 파일과 DB2 외부 루틴 공유 라이브러리 파일을 갱신하는 데만 사용됩니다. 응용프로그램 및 일반 업그레이드에 대한 주제는 관련 링크를 참조하십시오.
- 이 명령은 DB2 서비스 계약에 동의한 경우에만 지원됩니다. 이 명령은 있는 그대로 제공되므로 IBM은 의도하지 않거나 악의적인 사용에 대해 책임을 지지 않습니다.
- 이 명령은 수정 전에 공유 라이브러리 또는 실행할 수 있는 파일의 백업을 작성하지 않습니다. 이 명령을 실행하기 전에 반드시 파일의 백업 사본을 작성해야 합니다.

## 제 175 장 db2chgpath - 임베디드(Embedded) 런타임 경로 변경

Linux 및 UNIX 시스템의 DB2 설치 프로그램에서 관련 DB2 라이브러리 및 실행할 수 있는 파일에서 임베디드(embedded) 런타임 경로를 갱신하는 데 사용됩니다. 이 명령은 DB2 설치 중에 명령 관련 오류가 발생한 경우 IBM DB2 Support의 지시에 따라 재실행될 수 있습니다.

주: Red Hat Enterprise Linux 버전 5(RHEL5)에 DB2를 설치한 후 SELinux(Security-Enhanced Linux)를 사용할 수 있게 되는 경우 현재 릴리스의 각 DB2 설치마다 이 명령을 수동으로 실행하여 DB2가 올바르게 작동하도록 해야 합니다. 추가 정보는 아래 사용 시 참고사항 섹션을 참조하십시오.

### 권한 부여

루트 설치에는 루트 권한이 필요합니다. 비루트 설치의 경우 비루트 설치를 소유하는 사용자 ID로 로그인해야 합니다.

### 필요한 연결

없음

### 명령 구문

```
db2chgpath [-d] [-f file-name]
```

### 명령 매개변수

**-d** 디버그 모드를 활성화합니다. 이 옵션은 DB2 Support의 지시가 있는 경우에만 사용됩니다.

**-f file-name**

런타임 경로를 갱신할 특정 파일 이름을 지정합니다. *file-name*은 현재 DB2 제품 설치 위치의 기준과 관련된 경로 이름을 가져야 합니다.

### 예:

- DB2 제품 설치 경로의 모든 파일을 점검하고 런타임 경로 갱신을 수행하려면 다음을 실행하십시오.

```
<DB2 installation path>/install/db2chgpath
```

- <DB2 installation path>/lib64 디렉토리에 있는 libdb2.a라는 특정 파일에 대한 경로를 갱신하려면 다음을 실행하십시오.

```
<DB2 installation path>/install/db2chgpath -f lib64/libdb2.a
```

## 사용 시 참고사항

RHEL5 시스템의 경우 SELinux가 설치 제거되거나 사용하지 않을 때 사용자가 DB2 제품을 설치하고 SELinux를 사용하려면 다음 단계를 수행합니다.

- 필요한 경우 SELinux rpms를 설치합니다.
- /etc/sysconfig/selinux를 변경합니다. 상태를 "permissive" 또는 "enforcing"으로 설정합니다.
- 머신을 재부트하여 모든 파일에 SELinux 레이블을 적용합니다.
- db2chgpath를 실행하여 텍스트 위치가 수정된 DB2 공유 라이브러리를 로드할 수 있는 SELinux 속성을 설정하십시오(textrel\_shlib\_t).

## 제 176 장 db2ckbkp - 백업 점검

이 유틸리티를 사용하면 백업 이미지의 무결성을 테스트하고 이미지를 리스토어할 수 있는지 여부를 판별할 수 있습니다. 또한 백업 헤더에 저장된 메타데이터를 표시할 수도 있습니다.

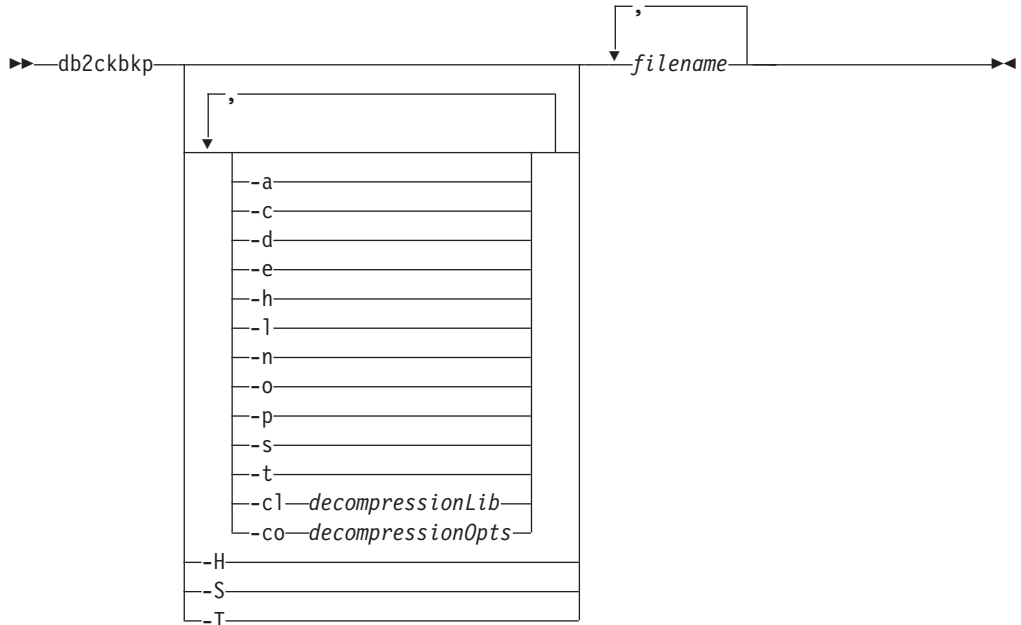
### 권한 부여

누구나 유틸리티에 액세스할 수 있지만, 사용자가 이미지 백업에 대해 이 유틸리티를 실행하려면 이미지 백업에 대한 읽기 권한이 있어야 합니다.

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

**-a** 사용 가능한 모든 정보를 표시합니다.

**-c** 체크비트 및 체크섬 결과를 표시합니다.

**-cl** *decompressionLib*

압축 해제를 수행하는 데 사용될 라이브러리 이름을 표시합니다. 이 이름은 서버에 있는 파일을 나타내는 완전한 경로이어야 합니다. 이 매개변수를 지정하

지 않으면 DB2는 이미지에 저장된 라이브러리를 사용하려고 합니다. 백업이 압축되지 않은 경우 이 매개변수 값은 무시됩니다. 지정된 라이브러리를 로드할 수 없는 경우 작업에 실패합니다.

**-co** *decompressionOpts*

압축 해제 라이브러리의 초기화 루틴으로 전달될 2진 데이터 블록을 설명합니다. DB2는 클라이언트에서 서버로 직접 이 문자열을 전달하므로 모든 바이트 리버설 또는 코드 페이지 변환 문제는 압축 해제 라이브러리에서 처리되어야 합니다. 데이터 블록의 첫 번째 문자가 '@'일 경우, DB2는 이 데이터의 나머지 부분을 서버에 있는 파일 이름으로 해석합니다. 그런 다음 DB2는 *string*의 콘텐츠를 이 파일의 콘텐츠로 바꾼 후 대신 이 새 값을 초기화 루틴으로 전달합니다. *string*의 최대 길이는 1024바이트입니다.

**-d** DMS 테이블 스페이스 데이터 페이지의 헤더에 있는 정보를 표시합니다.

**-e** 이미지에서 파일로 페이지를 추출합니다. 페이지를 추출하려면 입력 및 출력 파일이 필요합니다. 디폴트 입력 파일은 `extractPage.in`입니다. `DB2LISTFILE` 환경 변수를 전체 경로로 설정하여 디폴트 입력 파일을 겹쳐쓸 수 있습니다. 입력 파일 형식은 다음과 같습니다.

SMS 테이블 스페이스의 경우:

S <tblspID> <objID> <objType> <startPage> <numPages>

주:

1. <startPage>는 오브젝트와 관련되는 오브젝트 페이지 번호입니다.

DMS 테이블 스페이스의 경우:

D <tblspID> <objType> <startPage> <numPages>

주:

1. <objType>은 DMS 로드 사본 이미지를 검증하는 경우에만 필요합니다.
2. <startPage>는 풀과 관련되는 오브젝트 페이지 번호입니다.

로그 파일의 경우:

L <log num> <startPos> <numPages>

기타 데이터의 경우(예: 초기 데이터):

O <objType> <startPos> <numBytes>

디폴트 출력 파일은 `extractPage.out`입니다. `DB2EXTRACTFILE` 환경 변수를 전체 경로로 설정하여 디폴트 출력 파일 이름을 겹쳐쓸 수 있습니다.

**-h** 리스토어 유틸리티가 예상하는 이미지의 경로와 이름을 포함한 미디어 헤더 정보를 표시합니다.



- H -h와 동일한 정보를 표시하지만 이미지의 시작부터 4K 미디어 헤더 정보만 읽습니다. 이미지 유효성을 확인하지 않습니다. 이 옵션은 다른 옵션과 조합하여 사용할 수 없습니다.
- l 로그 파일 헤더(LFH) 및 미리 로그 파일 헤더(MFH) 데이터를 표시합니다.
- n 테이프 마운트를 프롬프트합니다. 디바이스당 하나의 테이프를 가정합니다.
- o 오브젝트 헤더의 세부사항 정보를 표시합니다.
- p 오브젝트 유형의 페이지 수를 표시합니다. 이 옵션은 백업이 DMS 테이블 스페이스 데이터에 대해 수행된 경우 다른 모든 오브젝트 유형에 대한 페이지 수를 표시하지 않습니다. 모든 페이지의 총계만 SQLUDMSTABLESPACEDATA로 표시합니다. SQLUDMSLOBDATA 및 SQLUDMSLONGDATA에 대한 오브젝트 유형은 DMS 테이블 스페이스의 경우 0입니다.
- s 이미지의 자동 스토리지 경로를 표시합니다.
- S 동일한 정보를 -s로 표시하지만 이미지 유효성을 확인하지 않습니다. 이 옵션은 다른 옵션과 조합하여 사용할 수 없습니다.
- t 이미지의 테이블 스페이스에 대한 테이블 스페이스 세부사항을 컨테이너 정보와 함께 표시합니다.
- T 동일한 정보를 -t로 표시하지만 이미지 유효성을 확인하지 않습니다. 이 옵션은 다른 옵션과 조합하여 사용할 수 없습니다.

*filename*

백업 이미지 파일의 이름입니다. 한 번에 하나 이상의 파일을 점검할 수 있습니다.

주:

1. 완전 백업이 다중 오브젝트로 구성되는 경우 유효성 확인은 db2ckbcp가 모든 오브젝트를 동시에 유효성 확인하는 데 사용하는 경우에만 성공합니다.
2. 이미지의 다중 파트를 검사할 때 첫 번째 백업 이미지 오브젝트(.001)가 첫 번째로 지정되어야 합니다.

예:

예 1(UNIX 플랫폼에서)

```
db2ckbcp SAMPLE.0.krodger.NODE0000.CATN0000.19990817150714.001
SAMPLE.0.krodger.NODE0000.CATN0000.19990817150714.002
SAMPLE.0.krodger.NODE0000.CATN0000.19990817150714.003
```

```
[1] Buffers processed: ##
[2] Buffers processed: ##
[3] Buffers processed: ##
Image Verification Complete - successful.
```

예 2:

```
db2ckbkp -h SAMPLE2.0.krodger.NODE0000.CATN0000.19990818122909.001

=====
MEDIA HEADER REACHED:
=====
Server Database Name          -- SAMPLE2
Server Database Alias        -- SAMPLE2
Client Database Alias        -- SAMPLE2
Timestamp                     -- 19990818122909
Database Partition Number    -- 0
Instance                      -- krodger
Sequence Number              -- 1
Release ID                   -- 900
Database Seed                 -- 65E0B395
DB Comment's Codepage (Volume) -- 0
DB Comment (Volume)          --
DB Comment's Codepage (System) -- 0
DB Comment (System)          --
Authentication Value         -- 255
Backup Mode                   -- 0
Include Logs                  -- 0
Compression                   -- 0
Backup Type                   -- 0
Backup Gran.                  -- 0
Status Flags                  -- 11
System Cats inc               -- 1
Catalog Database Partition No. -- 0
DB Codeset                    -- IS08859-1
DB Territory                  --
LogID                         -- 1074717952
LogPath                       -- /home/krodger/krodger/NODE0000/
                               SQL00001/SQLLOGDIR
Backup Buffer Size             -- 4194304
Number of Sessions            -- 1
Platform                      -- 0

The proper image file name would be:
SAMPLE2.0.krodger.NODE0000.CATN0000.19990818122909.001

[1] Buffers processed: ####
Image Verification Complete - successful.
```

## 사용 시 참고사항

1. 백업 이미지가 다중 세션을 사용하여 작성된 경우 db2ckbkp는 모든 파일을 동시에 조사할 수 있습니다. 사용자가 시퀀스 번호 001을 갖는 세션이 지정된 첫 번째 파일인지 확인해야 합니다.
2. 이 유틸리티는 또한(가변 블록 크기를 사용하여 작성된 이미지를 제외하고) 테이블에 저장되는 백업 이미지를 검증할 수 있습니다. 이것은 테이프를 리스��어 조작에 대해 준비한 후 유틸리티를 호출하고 테이프 디바이스 이름을 지정하여 수행됩니다. 예를 들어 UNIX 기반 시스템에서는

```
db2ckbkp -h /dev/rmt0
```

및 Windows에서는

```
db2ckbkp -d \\.\tape1
```

3. 이미지가 테이프 디바이스에 있는 경우 테이프 디바이스 경로를 지정하십시오. 옵션 -n이 제공되지 않은 경우 마운트되었는지 확인하도록 프롬프트됩니다. 다중 테이프

가 있는 경우 첫 번째 테이프가 제공되는 첫 번째 디바이스 경로에 마운트되어야 합니다. (이는 헤더에서 순서 001을 갖는 테이프입니다.)

테이프 디바이스가 발견될 때의 디폴트는 사용자에게 테이프를 마운트하도록 프롬프트하는 것입니다. 사용자는 프롬프트에서 선택할 수 있습니다. 다음은 프롬프트 및 옵션입니다. (디바이스 I가 지정되는 곳은 디바이스 경로 /dev/rmt0임)

```
Please mount the source media on device /dev/rmt0.  
Continue(c), terminate only this device(d), or abort this tool(t)?  
(c/d/t)
```

사용자에게 지정되는 각 디바이스에 대해 그리고 디바이스가 테이프 끝에 도달할 때 프롬프트됩니다.



## 제 177 장 db2ckupgrade - 데이터베이스 업그레이드 점검

데이터베이스를 이주할 수 있는지 검증합니다.

### 범위

파티션된 데이터베이스 환경에서는 각 데이터베이스 파티션에서 db2ckupgrade 명령을 실행하십시오. 이 명령은 명령이 발행되는 데이터베이스 파티션에만 영향을 줍니다.

### 권한 부여

SYSADM

### 필수 연결

없음

### 명령 구문

```
db2ckupgrade database [-e] [-l filename] [-not1] [-u userid] [-p password]
```

### 명령 매개변수

#### 데이터베이스

스캔할 로컬 데이터베이스의 별명 이름을 지정합니다.

**-e** 모든 로컬 카탈로그된 데이터베이스를 스캔하도록 지정합니다.

#### **-l filename**

스캔된 데이터베이스에 대해 생성된 오류 및 경고의 목록을 유지할 로그 파일을 지정합니다.

**-not1** 유형 1 인덱스에 대한 점검을 사용하지 않습니다. 이 옵션이 지정되지 않으면 db2ckupgrade 명령은 유형 1 인덱스를 점검하고 식별된 각 유형 1 인덱스에 대한 **ALLOW WRITE ACCESS** 및 **CONVERT** 절이 있는 REORG INDEXES ALL 명령이 포함된 로그 파일에 대해 표시된 동일한 디렉토리에 type1\_index\_database-name.db2 스크립트 파일을 생성합니다.

#### **-p password**

시스템 관리자 사용자 ID의 암호를 지정합니다.

**-u** *userid*

시스템 관리자의 사용자 ID를 지정합니다.

## 사용 시 참고사항

db2ckupgrade 명령을 실행하려면 다음을 수행하십시오.

- Linux 및 UNIX 운영 체제에서 업그레이드하려는 새 DB2 사본을 설치하십시오. 그런 다음, *DB2DIR/bin* 디렉토리에서 db2ckupgrade 명령을 실행하십시오. 여기서 *DB2DIR*은 DB2 사본이 설치된 위치입니다.
- Windows 운영 체제에서, 업그레이드할 DB2 데이터베이스 제품 CD를 삽입하십시오. 그런 다음 CD의 *db2\Windows\Utilities* 디렉토리에서 db2ckupgrade 명령을 실행하십시오.

db2ckupgrade 명령은 리모트 데이터베이스로 카탈로그되는 데이터베이스에 대해 실행하지 못합니다.

이 명령은 다음의 모든 조건이 참인지 검증합니다.

- 카탈로그된 데이터베이스가 실제로 존재합니다.
- 데이터베이스가 불일치 상태에 있지 않습니다.
- 데이터베이스가 백업 보류 상태에 있지 않습니다.
- 데이터베이스가 리스토어 보류 상태에 있지 않습니다.
- 데이터베이스가 롤 포워드 보류 상태에 있지 않습니다.
- 테이블 스페이스가 정상 상태에 있습니다.
- 데이터베이스에 이름 ARRAY, BINARY, CURSOR, DECFLOAT, ROW, VARBINARY 또는 XML을 갖는 사용자 정의 유형(UDT)이 없습니다.
- 데이터베이스에 시스템 정의 DATALINK 데이터 유형이 없습니다.
- 데이터베이스에 이름 SYSPUBLIC을 갖는 스키마가 없습니다.
- 데이터베이스의 시스템 카탈로그 테이블에 데이터베이스 업그레이드가 실패하도록 하는 상위 항목이 삭제된 오브젝트 행이 없습니다.
- HADR 기본 데이터베이스로서 사용 가능한 데이터베이스가 성공적인 연결을 허용합니다.
- HADR 데이터베이스 역할이 대기가 아닙니다.
- SYSCATSPACE가 DMS 테이블 스페이스이고 AUTORESIZE가 사용 가능하지 않은 경우 SYSCATSPACE는 최소한 총 페이지의 50% 사용 가능한 페이지를 갖습니다.
- 데이터베이스가 XML Extender에 대해 사용 가능하지 않습니다.

업그레이드 프로세스에서 성공하려면 로컬 데이터베이스가 이들 점검을 모두 통과해야 합니다. db2iupgrade 명령은 **-not1** 매개변수와 함께 db2ckupgrade.log를 로그 파일

로 지정하여 db2ckupgrade 명령을 호출합니다. db2ckupgrade 명령이 위에 나열된 조건 중 하나가 참이 아님을 발견하는 경우 db2iupgrade는 실패하며 DBI1205E 오류 코드를 리턴합니다. 사용자는 인스턴스를 업그레이드하기 전에 이들 오류를 해결해야 합니다.

db2ckupgrade 명령이 **-l** 매개변수로 지정된 로그 파일에 다음 조건 중 하나에 대한 경고 메시지를 기록합니다.

- 컬럼 이름, 루틴 매개변수 이름 또는 변수 이름이 NULL로 호출됩니다.
- 유형 1 인덱스가 데이터베이스에 존재합니다.
- 외부 루틴이 NOT FENCED 및 NOT THREADSAFE로 선언되거나 UNIX 및 Linux 운영 체제의 사용자 정의 랩퍼가 NOT FENCED로 선언됩니다.
- 워크로드 연결 속성에 별표(\*)가 있습니다.
- 데이터베이스를 DB2 WebSphere® MQ 함수에 사용 가능합니다.

Windows 운영 체제에 설치하는 중에, 기존 제품으로 설치 창에서 업그레이드 조치와 함께 DB2 사본을 선택하고 인스턴스에 카탈로그된 로컬 데이터베이스가 있는 경우, 메시지 상자가 DB2 데이터베이스 제품 CD에서 db2ckupgrade 명령을 실행해야 한다고 경고합니다. 그러면 다음 조치 중 하나를 선택할 수 있습니다.

- 메시지를 무시하고 설치 프로세스를 계속하십시오.
- db2ckupgrade 명령을 실행하십시오. 이 명령이 성공적으로 실행하는 경우 설치 프로세스를 계속하십시오. 오류를 발견하는 경우 설치 프로세스를 종료하고, 모든 오류를 정정한 후 설치 프로세스를 다시 실행하십시오.
- 설치 프로세스를 종료하십시오.

**-not1** 매개변수가 생략되면 db2ckupgrade 명령이 db2IdentifyType1 명령을 호출하여 유형 1 인덱스를 식별하고 특정 데이터베이스에 유형 1 인덱스를 유형 2 인덱스로 변환하도록 스크립트를 생성합니다. db2IdentifyType1 명령은 처리를 완료하는데 시간이 오래 걸릴 수 있습니다. db2IdentifyType1 명령의 실행 시간은 데이터베이스의 테이블 수 및 데이터베이스 파티션 수에 비례합니다. 다음 성능 고려사항을 고려하십시오.

- 데이터베이스에 테이블 수가 많거나 데이터베이스 파티션 수가 많은 경우 또는 양 쪽의 수가 모두 많은 경우 모든 테이블을 처리하기 전에 우선 **-s** 또는 **-t** 매개변수를 사용하여 특정 스키마 또는 테이블에 db2IdentifyType1 명령을 실행하십시오. 그런 다음, **-not1** 매개변수와 함께 db2ckupgrade 명령을 실행하십시오.
- 버전 9.1 또는 버전 9.5의 파티션된 데이터베이스 환경에서 모든 유형 1 인덱스를 발견하려면 하나의 데이터베이스 파티션 특히 성능이 더 빠른 카탈로그 파티션의 db2ckupgrade 명령을 실행하십시오. 그런 다음, 모든 후속 파티션에서 **-not1** 매개변수와 함께 db2ckupgrade 명령을 실행하십시오.

데이터베이스를 업그레이드할 준비가 되었는지 검증하려면 *DB2 버전 9.7로 업그레이드*의 『데이터베이스 업그레이드 준비 검증』을 참조하십시오.



## 제 178 장 db2ckrst - 증분 리스토어 이미지 순서 확인

데이터베이스 실행기록을 쿼리하고 증분 리스토어에 필요한 백업 이미지에 대한 시간소인 목록을 생성합니다. 수동 증분 리스토어를 위한 단순 리스토어 구문 또한 생성됩니다.

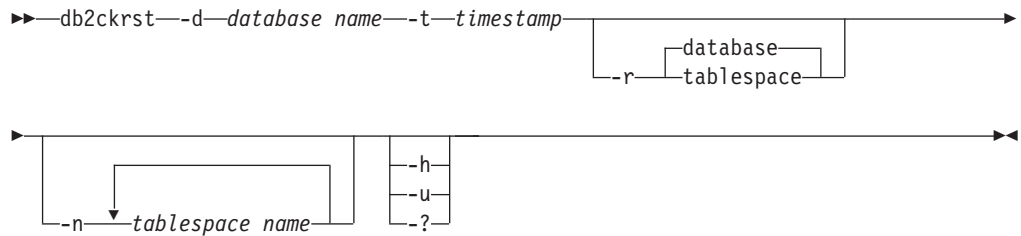
### 권한 부여

없음

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

#### **-d** *database name*

리스토어될 데이터베이스의 별명 이름을 지정합니다.

#### **-t** *timestamp*

점차적으로 리스토어될 백업 이미지의 시간소인을 지정합니다.

**-r** 실행될 리스토어 유형을 지정합니다. 디폴트값은 데이터베이스입니다. `tablespace` 가 선택되고 테이블 스페이스 이름이 지정되지 않으면 유틸리티가 지정된 이미지의 실행기록 항목을 확인하고 나열된 테이블 스페이스 이름을 사용하여 리스토어를 수행합니다.

#### **-n** *tablespace name*

리스토어될 하나 이상의 테이블 스페이스의 이름을 지정합니다. 데이터베이스 리스토어 유형이 선택되고 테이블 스페이스 이름의 목록이 지정되면 유틸리티가 지정된 테이블 스페이스 이름을 사용하여 테이블 스페이스 리스토어로서 계속 됩니다.

## **-h | -u | -?**

도움말 정보를 표시합니다. 이 옵션이 지정되면 다른 모든 옵션은 무시되고 도  
움말 정보만 표시됩니다.

## **예:**

```
db2ckrst -d mr -t 20001015193455 -r database
db2ckrst -d mr -t 20001015193455 -r tablespace
db2ckrst -d mr -t 20001015193455 -r tablespace -n tbspl tbsp2

> db2 backup db mr

백업이 완료되었습니다. 이 백업 이미지에 대한 시간소인은 20001016001426입니다.

> db2 backup db mr incremental

백업이 완료되었습니다. 이 백업 이미지에 대한 시간소인은 20001016001445입니다.

> db2ckrst -d mr -t 20001016001445

Suggested restore order of images using timestamp 20001016001445 for
database mr.
=====
db2 restore db mr incremental taken at 20001016001445
db2 restore db mr incremental taken at 20001016001426
db2 restore db mr incremental taken at 20001016001445
=====

> db2ckrst -d mr -t 20001016001445 -r tablespace -n userspace1
Suggested restore order of images using timestamp 20001016001445 for
database mr.
=====
db2 restore db mr tablespace ( USERSPACE1 ) incremental taken at
20001016001445
db2 restore db mr tablespace ( USERSPACE1 ) incremental taken at
20001016001426
db2 restore db mr tablespace ( USERSPACE1 ) incremental taken at
20001016001445
=====
```

## **사용 시 참고사항**

db2ckrst 유틸리티는 데이터베이스 재빌드를 위해 확장되지 않습니다. 실행기록 파일의 제한조건으로 인해, 여러 이미지에서 여러 테이블 스페이스를 리스토어해야 하는 경우 유틸리티는 올바른 목록을 제공할 수 없습니다.

이 유틸리티를 사용하려면 데이터베이스 실행기록이 존재해야 합니다. 데이터베이스 실행기록이 존재하지 않는 경우 이 유틸리티를 사용하기 전에 RESTORE 명령에 HISTORY FILE 옵션을 지정하십시오.

PRUNE HISTORY 명령의 FORCE 옵션을 사용하면 데이터베이스의 자동 증분 리스토어에 필요한 항목을 삭제할 수 있습니다. 직접 리스토어는 계속 올바르게 작동합니다. 이 명령을 사용하면 db2ckrst 유틸리티가 필수 백업 이미지의 전체 체인을 정확하게 분석하지 못하게 할 수 있습니다. PRUNE HISTORY 명령의 디폴트 조작은 필요한 항목이 삭제되지 않게 합니다. PRUNE HISTORY 명령의 FORCE 옵션은 사용하지 않는 것이 좋습니다.

이 유틸리티는 백업 레코드 보존을 위한 대체 방법으로 사용해서는 안됩니다.



---

## 제 179 장 db2cli - DB2 대화식 CLI

CLI에서 디자인 및 프로토타입 작성을 위해 대화식 콜 레벨 인터페이스(CLI) 환경을 시작합니다. 데이터베이스 인스턴스 소유자의 홈 디렉토리의 `sqllib/samples/cli/` 서브디렉토리에 있습니다. ODBC 및 CLI용 IBM Data Server Driver의 경우, `db2cli` 실행 파일은 `clidriver/bin` 디렉토리에 있습니다.

### 권한 부여

없음

### 필수 연결

없음

### 명령 구문

▶▶—db2cli—◀◀

### 명령 매개변수

없음

### 사용 시 참고사항

DB2 대화식 CLI는 CLI 함수 호출을 디자인, 프로토타입 작성 및 테스트하는 데 사용할 수 있는 명령 세트로 구성됩니다. 사용하려는 사람의 편의를 위해 제공되는 프로그래머의 테스트 도구이며, IBM은 성능에 대해 보증하지 않습니다. DB2 대화식 CLI는 일반 사용자를 위한 것이 아니므로, 방대한 오류 점검 기능이 없습니다.

두 유형의 명령이 지원됩니다.

#### CLI 명령

IBM CLI가 지원하는 각 함수 호출에 대응하는(또한 동일한 이름을 갖는) 명령

#### 지원 명령

동등한 CLI 기능이 없는 명령.

명령은 대화식으로 또는 파일 안에서 발행할 수 있습니다. 비슷하게 명령 출력을 터미널에 표시하거나 파일에 기록할 수 있습니다. CLI 명령 드라이버의 유용한 기능은 세션 중에 입력되는 모든 명령을 캡처하고 이를 파일에 기록한 후 나중에 다시 실행할 수 있는 명령 스크립트를 작성하는 기능입니다.



---

## 제 180 장 db2cmd - DB2 명령 창 열기

CLP 사용 DB2 창을 열고, DB2 명령행 환경을 초기화합니다. 이 명령을 발행하는 것은 DB2 명령 창 아이콘을 누르는 것과 동일합니다.

이 명령은 Windows 운영 체제에서만 사용할 수 있습니다.

### 권한 부여

없음

### 필수 연결

없음

### 명령 구문

▶ db2cmd [option-flag] [command] ▶

### 명령 매개변수

**-c | /c** 새 DB2 명령 창에서 -c 옵션 뒤에 오는 명령을 실행한 후 종료합니다. 예를 들어 db2cmd -c dir은 dir 명령이 새 DB2 명령 창에서 호출되도록 한 후 DB2 명령 창이 닫힙니다.

**-w | /w**

-w 옵션 뒤에 오는 명령을 새 DB2 명령 창에서 실행하고, 프로세스를 종료하기 전에 새 DB2 명령 창이 닫히기를 기다립니다. 예를 들어 db2cmd /w dir은 dir 명령을 호출하며, 프로세스는 새 DB2 명령 창이 닫힐 때까지 종료되지 않습니다.

**-i | /i** 동일한 DB2 명령 창을 공유하고 파일 핸들을 상속하는 동안 -i 옵션 뒤에 있는 명령을 실행합니다. 예를 들어 db2cmd -i dir은 동일한 DB2 명령 창에서 dir 명령을 실행합니다.

**-t | /t** 지정된 명령을 이 새 창의 제목으로 사용하여 -t 옵션 뒤에 오는 명령을 새 DB2 CLP 창에서 실행합니다.

## 사용 시 참고사항

CLP 사용 DB2 창을 표시할 때 DB21061E("명령행 환경이 초기화되지 않음.")가 리턴되는 경우 운영 체제가 환경 스페이스를 모두 사용해 버릴 수 있습니다. SHELL 환경 설정 매개변수에 대해 config.sys 파일을 점검하고, 그에 따라서 값을 늘리십시오. 예를 들면, 다음과 같습니다.

```
SHELL=C:\COMMAND.COM C:\# /P /E:32768
```



---

## 제 181 장 db2cptsa - DB2 HA 스크립트 설치 또는 갱신 명령

이 유틸리티는 UNIX 및 Linux 시스템의 /usr/sbin/rsct/sapolicies/db2에 DB2 HA(High Availability) 스크립트를 설치하거나 갱신합니다. DB2 HA 기능으로 IBM Tivoli SA MP(System Automation for Multiplatforms)를 사용하려면 이러한 DB2 HA 스크립트가 필요합니다.

### 권한 부여

이 명령은 root 권한으로 실행해야 합니다.

### 필수 연결

없음

### 명령 구문

```
db2cptsa [-c] [-f] [-r] [-h | -?]
```

### 명령 매개변수

- c DB2 HA 스크립트가 /usr/sbin/rsct/sapolicies/db2에 있고 올바른 레벨인지 검증합니다.
- f /usr/sbin/rsct/sapolicies/db2에서 DB2 HA 스크립트 재설치를 강제 실행합니다. 이 인수가 없는 상태에서 이미 설치된 DB2 HA 스크립트가 설치되는 스크립트의 버전보다 높거나 같은 경우 설치된 스크립트를 겹쳐쓰지 않습니다.
- r /usr/sbin/rsct/sapolicies/db2 디렉토리를 제거합니다. 이 디렉토리에는 SA MP용 DB2 HA 스크립트가 있습니다. 이러한 스크립트와 이 디렉토리는 SA MP가 설치되지 않은 경우에만 제거됩니다.
- h | -? 도움말 정보를 표시합니다.

### 사용 시 참고사항

디폴트로, 이 유틸리티는 DB2 HA 스크립트를 /usr/sbin/rsct/sapolicies/db2에 설치합니다(스크립트가 이 위치에 설치되지 않았거나 이미 설치된 스크립트 버전이 설치될 스크립트 버전보다 이전 버전인 경우). 이 유틸리티는 SA MP가 이미 설치된 경우에만 DB2 HA 스크립트를 설치 또는 갱신합니다.

이 명령은 DB2 설치 미디어의 `db2/plat/tsamp` 디렉토리에 있습니다. 여기서 `plat`은 다음을 나타냅니다.

- `aix`(AIX 5L™용 DB2)
- `linux`(32비트 AMD 및 Intel® 시스템(x86)의 Linux용 DB2)
- `linuxamd64`(AMD64 및 Intel EM64T 시스템(x64)의 Linux용 DB2)
- `linuxppc`(POWER®(System i 및 pSeries®) 시스템의 Linux용 DB2)
- `linux390`(System z9® 및 zSeries®의 Linux용 DB2)

이 명령은 `DB2DIR/install/tsamp` 디렉토리에 있습니다. 여기서 `DB2DIR`은 UNIX 및 Linux 시스템용 DB2 데이터베이스 제품의 설치 경로입니다.

---

## 제 182 장 db2dart - 데이터베이스 분석 및 보고 도구

아키텍처가 올바른지 데이터베이스를 조사하고 직면한 오류를 보고합니다.

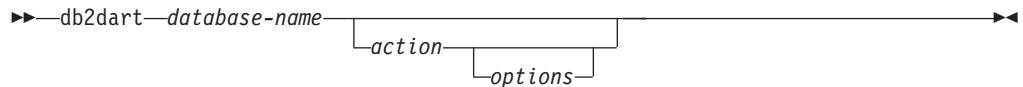
### 권한 부여

*sysadm*

### 필수 연결

없음. db2dart는 데이터베이스에 연결된 사용자가 없는 상태에서 실행해야 합니다.

### 명령 구문



### 명령 매개변수

#### 검사 조치

**/DB** 전체 데이터베이스를 검사합니다. 디폴트 옵션입니다.

**/T** 단일 테이블을 검사합니다. 두 개의 입력 값인 테이블 스페이스 ID와, 테이블 오브젝트 ID 또는 테이블 이름이 필요합니다.

**/TSF** 테이블 스페이스 파일 및 컨테이너만 검사합니다.

**/TSC** 테이블 스페이스의 구문은 검사하지만 해당 테이블은 검사하지 않습니다. 하나의 입력 값인 테이블 스페이스 ID가 필요합니다.

**/TS** 단일 테이블 스페이스와 해당 테이블을 검사합니다. 하나의 입력 값인 테이블 스페이스 ID가 필요합니다.

#### **/ATSC**

모든 테이블 스페이스의 구문을 검사하지만 해당 테이블은 검사하지 않습니다.

#### 데이터 형식화 조치

**/DD** 형식화된 테이블 데이터를 덤프합니다. 제시된 경우, 인라인 LOB 데이터도 표시됩니다. 5개의 입력 값인 테이블 오브젝트 ID나 테이블 이름, 테이블 스페이스 ID, 시작 페이지 번호, 페이지 수 및 상세 선택사항이 필요합니다.

**/DI** 형식화된 인덱스 데이터를 덤프합니다. 5개의 입력 값인 테이블 오브젝트 ID나 테이블 이름, 테이블 스페이스 ID, 시작 페이지 번호, 페이지 수 및 상세 선택사항이 필요합니다.

- 파티션된 테이블의 파티션되지 않은 인덱스에 대해, /DI 조치는 /OI 및 /TSI 옵션에 대한 처음 두 입력으로 SYSCAT.INDEXES의 INDEX\_OBJECTID 및 TBSPACEID를 사용합니다. 테이블 이름(/TN) 옵션은 이 조치에 대해 지원되지 않습니다.
- 파티션된 테이블의 파티션된 인덱스에 대해, /DI 조치는 SYSCAT.DATAPARTITION의 PARTITIONOBJECTID 및 TBSPACEID를 사용합니다. 테이블 이름(/TN) 옵션은 이 조치에 대해 지원되지 않습니다.

**/DM** 형식화된 블록 맵 데이터를 덤프합니다. 5개의 입력 값인 테이블 오브젝트 ID 나 테이블 이름, 테이블 스페이스 ID, 시작 페이지 번호, 페이지 수 및 상세 선택사항이 필요합니다. 데이터는 비어 있는 다차원적으로 클러스터된(MDC) 테이블 블록을 재개하기 위해 재구성 다음에 테이블 스페이스에 사용하도록 블록이 재개되었는지 여부를 표시합니다.

- /DP** 16진 형식으로 페이지를 덤프합니다.
- DMS 테이블 스페이스의 영구 오브젝트에 대해, /DP 조치에는 테이블 스페이스 ID, 시작 페이지 번호 및 페이지 수로 구성되는 세 개의 입력 값이 필요합니다.
  - SMS 테이블 스페이스의 영구 오브젝트에 대해, /DP 조치에는 테이블 스페이스 ID, 오브젝트 ID, 시작 페이지 번호, 페이지 수 및 오브젝트 유형으로 구성되는 5개의 입력 값이 필요합니다.

**/DTSF** 형식화된 테이블 스페이스 파일 정보를 덤프합니다.

**/DEMP** DMS 테이블에 대한 형식화된 Extent Map 페이지(EMP) 정보를 덤프합니다. 두 개의 입력 값인 테이블 스페이스 ID와, 테이블 오브젝트 ID 또는 테이블 이름이 필요합니다.

**/DDEL** 컬럼 식별자가 있는 ASCII 형식으로, 형식화된 테이블 데이터를 덤프합니다. 네 개의 입력 값인 테이블 오브젝트 ID나 테이블 이름, 테이블 스페이스 ID, 시작 페이지 번호, 페이지 수가 필요합니다.

**/DHWM** 상위 워터 마크(water mark) 정보를 덤프합니다. 하나의 입력 값인 테이블 스페이스 ID가 필요합니다.

**/DXA** ASCII 형식으로, 형식화된 XML 컬럼 데이터를 덤프합니다. 5개의 입력 값인 테이블 오브젝트 ID나 테이블 이름, 테이블 스페이스 ID, 시작 페이지 번호, 페이지 수 및 상세 선택사항이 필요합니다.

**/DXH** HEX 형식으로, 형식화된 XML 컬럼 데이터를 덤프합니다. 5개의 입력 값인

테이블 오브젝트 ID나 테이블 이름, 테이블 스페이스 ID, 시작 페이지 번호, 페이지 수 및 상세 선택사항이 필요합니다.

#### **/LHWM**

상위 워터 마크(water mark)를 낮추는 방법을 제안합니다. 두 개의 입력 값인 테이블 스페이스 ID 및 페이지 수(원하는 상위 워터 마크(water mark))가 필요합니다.

#### 수리 조치

**/ETS** 가능한 경우, 4KB 테이블 스페이스(DMS만)에서 테이블 한계를 확장합니다. 하나의 입력 값인 테이블 스페이스 ID가 필요합니다.

**/MI** 인덱스가 유효하지 않은 것으로 표시합니다. 이 매개변수를 지정하는 경우 데이터베이스는 오프라인이어야 합니다. 두 개의 입력 값인 테이블 스페이스 ID 및 인덱스 오브젝트 ID가 필요합니다. 파티션된 인덱스의 경우, 이 값은 SYSCAT.INDEXPARTITIONS의 INDPARTITIONOBJECTID 및 INDPARTITIONTBSPACEID에서 확보할 수 있습니다.

#### **/RHWM**

비어 있는 SMP Extent를 통해 상위 워터 마크(water mark)를 낮춥니다. 이 매개변수를 지정하는 경우 데이터베이스는 오프라인이어야 합니다. 하나의 입력 값인 테이블 스페이스 ID가 필요합니다.

#### 상태 변경 조치

#### **/CHST**

데이터베이스의 상태를 변경합니다. 이 매개변수를 지정하는 경우 데이터베이스는 오프라인이어야 합니다. 하나의 입력 값인 데이터베이스 백업 보류 상태가 필요합니다.

#### 도움말

**/H** 도움말 정보를 표시합니다.

#### 입력 값 옵션

#### **/OI** *object-id*

오브젝트 ID를 지정합니다.

#### **/TN** *table-name*

테이블의 이름

#### **/TSI** *tablespace-id*

테이블 스페이스 ID를 지정합니다.

**/ROW** *sum*

점검해야 하는 Long 필드 디스크립터, LOB 디스크립터 및 제어 정보를 식별합니다. 하나의 옵션만 지정하거나 값을 추가하여 두 개 이상의 옵션을 지정할 수 있습니다.

- 1       행에서 제어 정보를 점검합니다.
- 2       Long 필드 및 LOB 디스크립터를 점검합니다.

**/RPT** *path*

보고서 출력 파일의 선택적 경로.

**/RPTN** *file-name*

보고서 출력 파일의 선택적 이름.

**/PS** *number*

시작하는 페이지 번호를 지정합니다. 페이지 번호 뒤에는 폴 상대적임을 의미하는 p가 붙어야 합니다, /PS 0 /NP 0을 지정하면 지정된 오브젝트의 모든 페이지가 덤프됩니다.

**/NP** *number*

페이지 수를 지정합니다. /PS 0 /NP 0을 지정하면 지정된 오브젝트의 모든 페이지가 덤프됩니다.

**/V** *option*

상세 옵션을 구현해야 하는지 여부를 지정합니다. 가능한 값은 다음과 같습니다.

- Y       상세 옵션을 구현해야 함을 지정합니다.
- N       상세 옵션을 구현하지 않음을 지정합니다.

**/SCR** *option*

화면 출력 유형을 지정합니다(있는 경우). 가능한 값은 다음과 같습니다.

- Y       일반 화면 출력이 생성됩니다.
- M       최소화한 화면 출력이 생성됩니다.
- N       화면 출력이 생성되지 않습니다.

**/RPTF** *option*

보고서 파일 출력 유형을 지정합니다(있는 경우). 가능한 값은 다음과 같습니다.

- Y       일반 보고서 파일 출력이 생성됩니다.
- E       오류 정보만 보고서 파일로 생성됩니다.
- N       보고서 파일 출력이 생성되지 않습니다.

### **/ERR option**

DART.INF에서 생성할 로그의 유형을 지정합니다(있는 경우). 가능한 값은 다음과 같습니다.

**Y** DART.INF 파일에서 일반 로그를 생성합니다.

**N** 출력을 최소화하여 DART.INF 파일을 로그합니다.

**E** DART.INF 파일 및 화면 출력을 최소화합니다. 오류 정보만 보고서 파일로 보냅니다.

### **/WHAT DBBP option**

데이터베이스 백업 오류 상태를 지정합니다. 가능한 값은 다음과 같습니다.

**OFF** 해제 상태.

**ON** 설정 상태.

### **/QCK option**

빠른 옵션. /DB, /T 및 /TS 조치에만 적용됩니다. DAT 오브젝트의 페이지 0만 검사하고 부분적으로 인덱스 오브젝트를 검사합니다(BMP, LOB, LF 오브젝트는 검사하지 않으며 DAT 또는 INX 오브젝트를 전체적으로 횡단하지 않음).

### **/TYP option**

오브젝트의 유형을 지정합니다. 가능한 값은 다음과 같습니다.

**DAT** 오브젝트 유형은 DAT입니다.

**INX** 오브젝트 유형은 INDEX입니다.

**BKM** 오브젝트 유형은 BMP입니다.

## **사용 시 참고사항**

1. db2dart 명령을 호출할 때 하나의 조치만 지정할 수 있습니다. 조치에서 지원하는 옵션 수는 변할 수 있습니다.
2. db2dart 명령을 호출할 때 모든 필수 입력 값을 지정하지 않으면 값을 묻는 프롬프트가 표시됩니다. /DDEL 조치의 경우 명령행에서 옵션을 지정할 수 없으므로 db2dart에서 프롬프트될 때 입력해야 합니다.
3. 조치 외에 /ROW, /RPT, /RPTN, /SCR, /RPTF, /ERR 및 /WHAT DBBP 옵션 모두를 호출할 수 있습니다. 어떤 조치에도 필요하지 않습니다.
4. /DB, /T 및 /TS 옵션은 연관된 XML 스토리지 오브젝트를 포함하여, 지정된 오브젝트를 검사합니다. /DB 옵션은 데이터베이스에 있는 모든 XML 스토리지 오브젝트를 포함하고, /T 옵션은 지정된 테이블과 연관되는 XML 스토리지 오브젝트를 포함하며, /TS 옵션은 상위 오브젝트가 지정된 테이블 스페이스에 존재하는 모든 XML 스토리지 오브젝트를 검사합니다. 또한 /DEMP 옵션은 형식화된 EMP 정보(연관된 XML 스토리지 오브젝트에 대한 정보도 포함하여)를 덤프합니다.

5. 단일 테이블 스페이스에 대해 db2dart를 실행할 때, 종속 오브젝트가 있는 테이블 스페이스에 관계없이, 해당 테이블 스페이스에 있는 상위 테이블에 대한 모든 종속 오브젝트를 점검합니다. 지정된 테이블 스페이스의 외부에 있는 종속 오브젝트에 대해서는 Extent Map 페이지(EMP) 정보가 캡처되지 않습니다. EMP 정보는 상위 오브젝트가 지정된 테이블 스페이스가 아닌 다른 테이블 스페이스에 상주할 때도 지정된 테이블 스페이스에서 발견된 종속 오브젝트에 대해 캡처됩니다.
6. 파티션된 테이블의 경우, /DD, /DM, /DEMP, /DDEL, /DP, /DXA, /DXH 조치는 특정 파티션에 대한 테이블 오브젝트 ID(/OI) 및 테이블 스페이스 ID(/TSI)의 입력으로 syscat.datapartitions의 partitionobjectid 및 tbspaceid를 사용합니다. 테이블 이름 옵션(/TN)은 이 조치에 대해 지원되지 않습니다. /T 조치는 전체 테이블을 점검하기 위해 전역 테이블 스페이스 ID와 함께 사용할 때 테이블 이름 또는 전역 테이블 오브젝트 ID를 지원하고, 특정 파티션을 점검하기 위해 /OI 및 /TSI의 입력으로 syscat.datapartitions의 partitionobjectid 및 tbspaceid를 사용하는 것을 지원합니다.
7. 일반적으로, db2dart 요청은 데이터베이스가 오프라인일 때 실행됩니다. 그러나 /DHWM 및 /LHWM 조치의 경우 오프라인 데이터베이스는 엄밀히 필수는 아닙니다. 보고서는 데이터베이스가 오프라인이 아니어도 생성될 수 있지만, 결과에 대한 신뢰성은 최근에 발생한 쓰기/갱신 활동량에 따라 달라집니다(활동이 적으면 결과 신뢰성이 커짐).



---

## 제 183 장 db2daslevel - DAS 레벨 표시

시스템에서 DAS의 현재 레벨을 표시합니다. 이 명령은 디폴트로 콘솔에 표시됩니다.

**중요사항:** 이 명령은 제어 센터와 관련된 구성요소가 버전 9.7에서 사용되지 않기 때문에 사용되지 않으며 이후 릴리스에서 제거됩니다. 자세한 정보는 버전 9.7의 새로운 내용 책에 있는 『제어 센터 도구 및 DB2 Administration Server(DAS)는 사용되지 않음』 주제를 참조하십시오.

### 권한 부여

없음

### 필요한 연결

없음

### 명령 구문

▶▶—db2daslevel—▶▶

### 명령 매개변수

없음



## 제 184 장 db2dclgn - 선언 생성기

지정된 데이터베이스 테이블에 대한 선언을 생성하여 문서에서 선언을 찾을 필요성을 제거합니다. 생성된 선언은 필요한 대로 수정할 수 있습니다. 지원되는 호스트 언어는 C/C++, COBOL, JAVA 및 FORTRAN입니다.

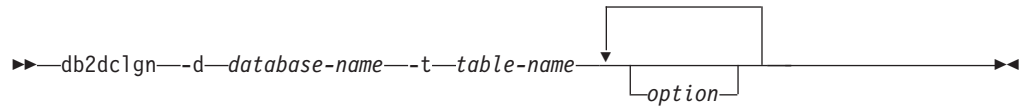
### 권한 부여

없음

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

#### **-d** *database-name*

연결이 설정될 데이터베이스의 이름을 지정합니다.

#### **-t** *table-name*

선언을 생성하기 위해 컬럼 정보를 검색할 테이블의 이름을 지정합니다.

*option* 다음 중 하나 이상:

#### **-a** *action*

선언이 추가 또는 교체될 것인지 여부를 지정합니다. 유효한 값은 **ADD** 및 **REPLACE**입니다. 디폴트값은 **ADD**입니다.

#### **-b** *lob-var-type*

LOB 컬럼에 대해 생성될 변수의 유형을 지정합니다. 가능한 값은 다음과 같습니다.

#### **LOB**(디폴트)

예를 들어 C에서, SQL TYPE은 CLOB(5K) x입니다.

#### **LOCATOR**

예를 들어 C에서, SQL TYPE은 CLOB\_LOCATOR x입니다.

**FILE** 예를 들어 C에서, SQL TYPE은 CLOB\_FILE x입니다.

- c 접두부(-n)가 지정될 때 컬럼 이름이 필드 이름에서 접미부로 사용될지 여부를 지정합니다. 접두부가 지정되지 않는 경우 이 옵션은 무시됩니다. 디폴트 동작은 컬럼 이름을 접미부로 사용하지 않고 대신 1에서 시작하는 컬럼 번호를 사용하는 것입니다.
- i 표시기 변수가 생성될지 여부를 지정합니다. 호스트 구조가 C 및 COBOL에서 지원되므로, 컬럼 수와 동일한 크기의 표시기 테이블이 생성되는 반면, JAVA 및 FORTRAN의 경우에는 각 컬럼에 대해 개별 표시기 변수가 생성됩니다. 표시기 테이블 및 변수의 이름은 각각 테이블 이름 및 컬럼 이름과 동일하고 "IND-"(COBOL) 또는 "ind\_"(기타 언어)가 접두부로 붙습니다. 디폴트 동작은 표시기 변수를 생성하지 않는 것입니다.

**-l language**

선언이 생성될 호스트 언어를 지정합니다. 유효 값은 C, COBOL, JAVA 및 FORTRAN입니다. 디폴트 동작은 C++에 대해서도 유효한 C 선언을 생성하는 것입니다.

**-n name**

필드 이름의 각각에 대한 접두부를 지정합니다. -c 옵션이 사용되는 경우 접두부를 지정해야 합니다. 지정되지 않으면 컬럼 이름이 필드 이름으로 사용됩니다.

**-o output-file**

선언에 대한 출력 파일의 이름을 지정합니다. 디폴트 동작은 테이블 이름을 기본 파일 이름으로 사용하고 생성된 호스트 언어를 반영하는 확장자를 사용하는 것입니다.

C의 경우 .h  
 COBOL의 경우 .cbl  
 JAVA의 경우 .java  
 FORTRAN(UNIX)의 경우 .f  
 FORTRAN(INTEL)의 경우 .for

**-p password**

데이터베이스에 연결하는 데 사용될 암호를 지정합니다. 사용자 ID가 지정되면 지정해야 합니다. 디폴트 동작은 연결을 설정할 때 암호를 제공하지 않는 것입니다.

**-r remarks**

컬럼 설명이 사용 가능한 경우 필드의 보다 자세한 설명을 제공하기 위해 선언에서 설명으로 사용될 것인지 여부를 지정합니다.

**-s structure-name**

선언에 모든 필드를 그룹화하기 위해 생성될 구조 이름을 지정합니다. 디폴트 동작은 규정되지 않은 테이블 이름을 사용하는 것입니다.

**-u** *userid*

데이터베이스에 연결하는 데 사용될 사용자 ID를 지정합니다. 암호가 지정되면 지정해야 합니다. 디폴트 동작은 연결을 설정할 때 사용자 ID를 제공하지 않는 것입니다.

**-v** 유틸리티의 상태(예: 연결 상태)가 표시될지 여부를 지정합니다. 디폴트 동작은 오류 메시지만 표시하는 것입니다.

**-w** *DBCS-var-type*

**sqldbchar** 또는 **wchar\_t**가 C에서 GRAPHIC/VARGRAPHIC/DBCLOB 컬럼에 사용될지 여부를 지정합니다.

**-y** *DBCS-symbol*

**G** 또는 **N**이 COBOL에서 DBCS 기호로 사용될지 여부를 지정합니다.

**-z** *encoding*

특정 서버에 따라서 코드화 규칙의 인코딩을 지정합니다. 인코딩은 **LUW** 또는 **OS390**일 수 있습니다. **OS390**이 지정되면 생성되는 파일은 OS390에 의해 생성되는 파일과 동일하게 보입니다.

**예:**

```
db2dclgn -d sample -t emp_resume -l cobol -a replace
```



## 제 185 장 db2diag - db2diag 로그 분석 도구

이 유틸리티는 단일 및 회전하는 db2diag 로그 파일 모두를 필터하고 형식화하기 위한 도구입니다. db2diag는 데이터베이스 관리 프로그램 **DIAGSIZE** 구성 매개변수가 설정된 경우 회전하는 db2 diag 로그에서 읽습니다. 그렇지 않으면, 디폴트로 디폴트 db2diag.log 파일에서 읽습니다.

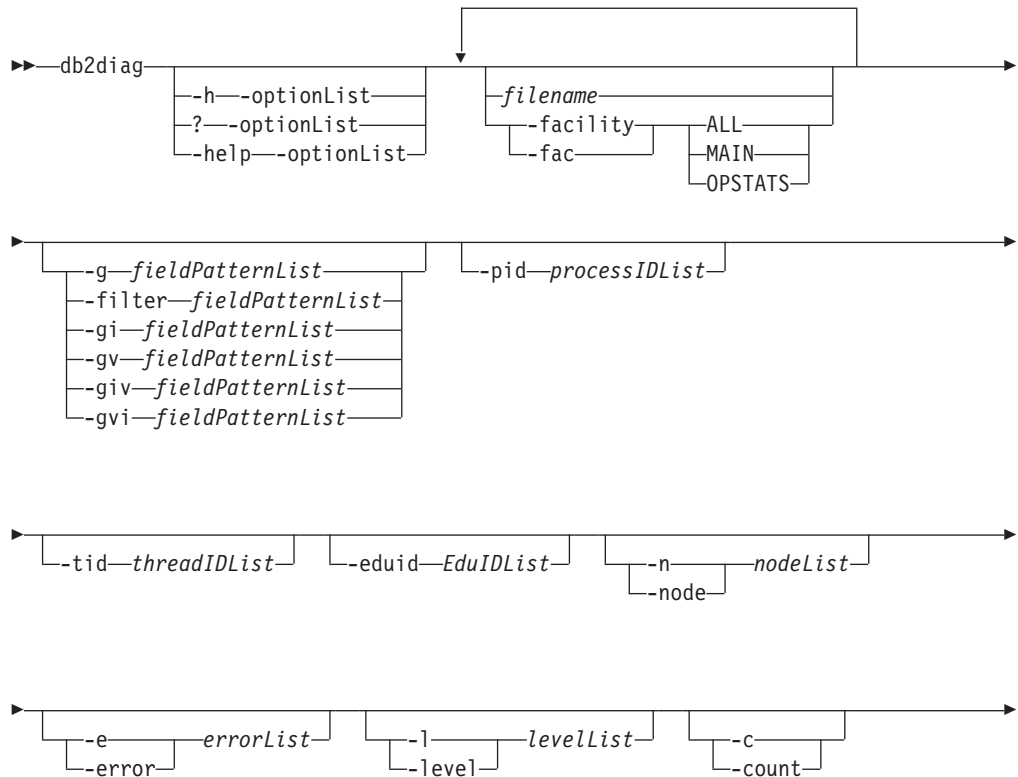
### 권한 부여

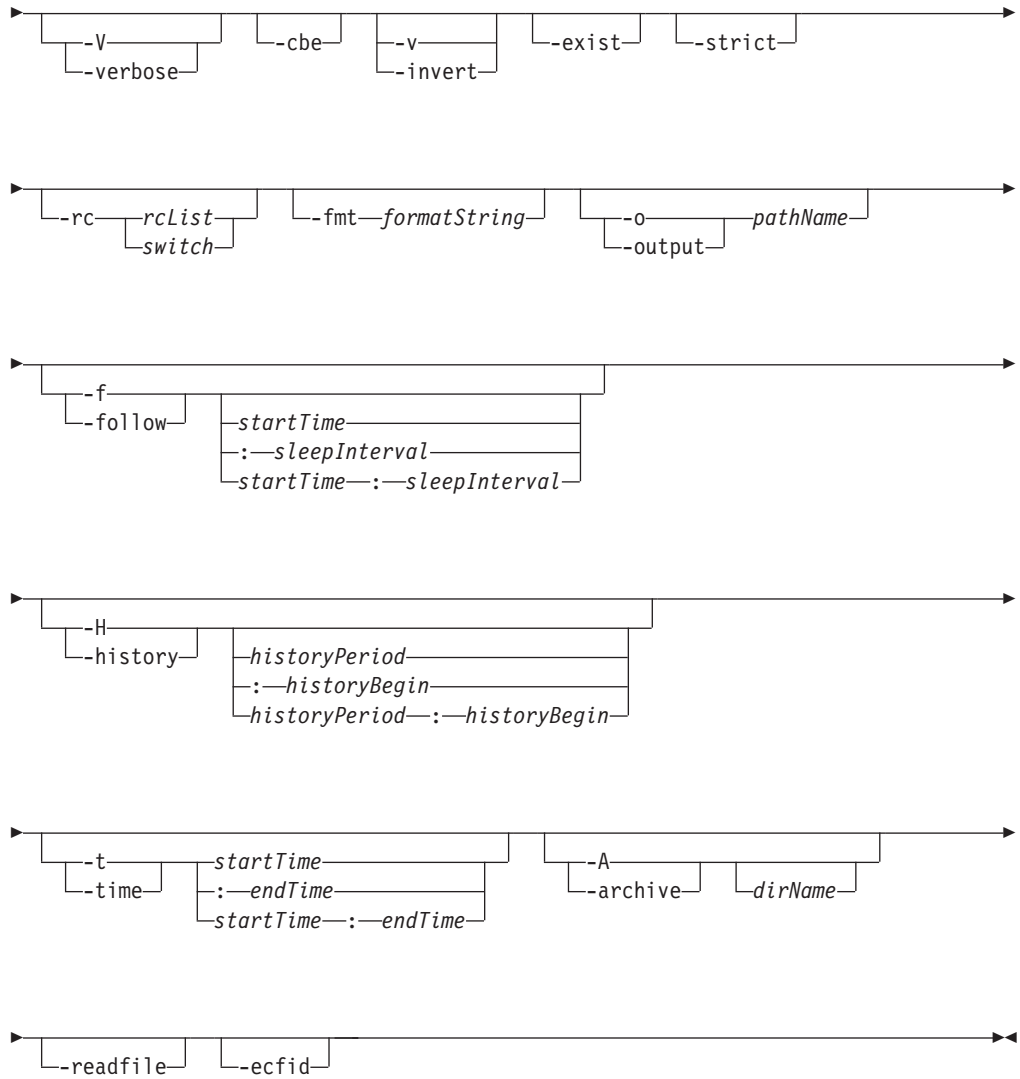
없음

### 필수 연결

없음

### 명령 구문





## 명령 매개변수

### *filename*

처리될 DB2 진단 로그의 경로 이름을 공백으로 구분하여 하나 이상 지정합니다. 파일 이름을 생략하면 현재 디렉토리의 db2diag 로그 파일이 처리됩니다. 파일을 찾을 수 없으면 **diagpath** 구성 매개변수로 설정된 디렉토리가 검색됩니다.

### **-facility | -fac**

해당 기능에서 파일을 읽습니다. 기능은 레코드의 논리적 그룹입니다. 예를 들어, 모든 옵티마이저 통계 레코드는 OPTSTATS 기능으로 그룹화됩니다. 출력은 디폴트로 텍스트 형식이 됩니다. 유효한 기능 옵션은 다음과 같습니다.

**ALL** 모든 기능의 레코드를 리턴합니다.



## MAIN

DB2 일반 진단 로그(예: db2diag 로그 파일) 및 회전 이벤트 로그의 레코드를 리턴합니다.

## OPSTATS

옵티마이저 통계에 관련된 레코드를 리턴합니다.

### -h | -help | ?

도움말 정보를 표시합니다. 이 옵션이 지정되면 다른 모든 옵션은 무시되고 도움말 정보만 표시됩니다. 쉘표로 구분되는 하나 이상의 명령 매개변수를 포함하는 옵션 목록(*optionList*)을 생략하면, 사용 가능한 모든 옵션 목록이 간단한 설명과 함께 표시됩니다. *optionList*에 지정된 옵션마다 세부사항 정보와 사용 예가 표시됩니다. *optionList* 인수 대신 다음 스위치 중 하나를 사용하여 도구에 대한 추가 정보 및 사용법을 표시하도록 도움말 출력을 수정할 수 있습니다.

**brief** 모든 옵션에 대한 도움말 정보를 예 없이 표시합니다.

**예** 도구 사용 시 지원하기 위한 몇 가지의 일반 예를 표시합니다.

### tutorial

고급 기능을 설명하는 예를 표시합니다.

**notes** 사용 시 참고사항 및 제한사항을 표시합니다.

**all** 각 옵션의 사용 예를 포함하여, 모든 옵션에 대한 전체 정보를 표시합니다.

### -fmt *formatString*

*%field*, *{field}*, *@field* 또는 *@{field}* 양식의 레코드 필드를 포함하는 형식 문자열(*formatString*)을 사용하여 db2diag 출력을 형식화합니다. *{field}* 및 *@{field}*는 필드 이름과, 필드 이름 뒤에 있을 수 있는 영숫자(또는 임의의 다른 허용 문자)를 구분하기 위해 사용됩니다. 모든 필드 이름에서는 대소문자가 구분되지 않습니다. 필드 이름은 확실하게 필드 이름을 인식하는 데 필요한 몇 개의 첫 문자로 축약될 수 있습니다. 또한 이름이 긴 필드에 별명을 사용할 수 있습니다. 필드 이름 이전의 접두부 *%* 또는 *@*는 필드가 비어 있는 경우 필드 앞에 있는 텍스트가 표시되는지(*%*), 그렇지 않은지(*@*) 여부를 지정합니다.

다음 필드는 현재 사용 가능합니다.

### timestamp | ts

시간소인. 이 필드는 해당되는 구성 필드(*%tsyear*, *%tsmonth*, *%tsday*, *%tshour*, *%tsmin*(분), *%tssec*(초), *%tsmsec*(UNIX 운영 체제의 경우 마이크로초, Windows 운영 체제의 경우 밀리초))로 나눌 수 있습니다.

**timezone | tz**

UTC(Universal Coordinated Time)와의 분 수 차이. 예를 들어, -300은 동부 표준시(Eastern Time)입니다.

**recordid | recid**

레코드의 고유한 영숫자 ID(예: I11455A696).

**audience**

로그된 메시지의 대상. 'E'는 외부 사용자(IBM 고객, 서비스 분석가 및 개발자)를 표시합니다. 'I'는 내부 사용자(서비스 분석가 및 개발자)를 표시합니다. 'D'는 개발자의 디버깅 정보를 표시합니다.

**level** 메시지의 심각도 레벨: Info, Warning, Error, Severe 또는 Event.

**source**

로그된 오류가 발생한 위치: Origin, OS, Received 또는 Sent.

**instance | inst**

인스턴스 이름.

**node** 데이터베이스 파티션 서버 번호.

**database | db**

데이터베이스 이름

**pid** 프로세스 ID.

**tid** 스레드 ID.

**eduid** EDU ID.

**eduname**

EDU 이름.

**process**

프로세스 ID와 연관된 이름으로, 큰따옴표 안에 있습니다(예: "db2sysc.exe").

**product**

제품 이름(예: DB2 COMMON).

**component**

구성요소 이름.

**funcname**

함수 이름.

**probe** 프로브 번호.

함수 전체 함수 설명: %prod, %comp, %funcname, probe:%probe.

**appid** 응용프로그램 ID. 이 값은 appl\_id 모니터 요소 데이터와 같습니다. 이 값을 해석하는 방법에 대한 세부사항 정보는 『appl\_id - 응용프로그램 ID 모니터 요소』를 참조하십시오.

**coordnode**

코디네이터 파티션.

**coordindex**

코디네이터 인덱스.

**apphdl**

응용프로그램 핸들: %coordnode - %coordindex.

**message | msg**

오류 메시지.

**calledprod**

오류를 리턴한 함수의 제품 이름.

**calledcomp**

오류를 리턴한 함수의 구성요소 이름.

**calledfunc**

오류를 리턴한 함수의 이름.

**called** 오류를 리턴한 함수의 전체 설명: %calledprod, %calledcomp, %calledfunc.

**rcval** 리턴 코드 값(32바이트).

**rcdesc** 오류 설명.

**retcode | rc**

호출된 함수에서 리턴되는 리턴 코드: %rcval %rcdesc.

**errno** 시스템 오류 번호.

**errname**

시스템 특정 오류 이름.

**oserror**

시스템 호출에 의해 리턴되는 운영 체제 오류: %errno %errname.

**callstack**

호출 스택.

**datadesc**

데이터 설명.

**dataobject**

데이터 오브젝트.

**data** 메시지의 전체 데이터 섹션: %datadesc %dataobject.

**argdesc**

인수 설명.

**argobject**

인수 오브젝트.

**arg** 오류를 리턴한 함수 호출의 인수: %argdesc %argobject.

이벤트 설명:

**impact**

사용자 영향(이벤트의 경우에만).

**startevent**

시작 이벤트 설명(\*).

**stopevent**

중지 이벤트 설명(\*).

**changeevent**

변경 이벤트 설명(\*).

**init** 초기화 이벤트 설명(\*).

**fini** 완료 이벤트 설명(\*).

**startup**

시동 이벤트 설명(\*).

**terminate**

종료 이벤트 설명(\*).

**bringdown**

Bringdown 이벤트 설명(\*).

**interrupt**

인터럽트 이벤트 설명(\*).

**associate**

연관 이벤트 설명(\*).

**disassociate**

연관 해제 이벤트 설명(\*).

**changeCFG**

구성 변경 이벤트 설명(\*).

**transfer**

전송 이벤트 설명(\*).

**dispatch**

디스패치 이벤트 설명(\*).

**switch** 전환 이벤트 설명(\*).

**report** 보고 이벤트 설명(\*).

**get** 가져오기 이벤트 설명(\*).

**free** 해제 이벤트 설명(\*).

**open** 열기 이벤트 설명(\*).

**close** 닫기 이벤트 설명(\*).

**work** 작업 이벤트 설명(\*).

**wait** 대기 이벤트 설명(\*).

**available**

사용 가능 이벤트 설명(\*).

**connect**

연결 이벤트 설명(\*).

**disconnect**

연결 끊기 이벤트 설명(\*).

**accept** 승인 이벤트 설명(\*).

**recv** 수신 이벤트 설명(\*).

**send** 보내기 이벤트 설명(\*).

**create** 작성 이벤트 설명(\*).

**destroy**

삭제(Destroy) 이벤트 설명(\*).

**request**

요청 이벤트 설명(\*).

**reply** 응답 이벤트 설명(\*).

**dependency**

종속성 이벤트 설명(\*).

**write** 쓰기 이벤트 설명(\*).

**read** 읽기 이벤트 설명(\*).

**reset** 재설정 이벤트 설명(\*).

**collect** 수집 이벤트 설명(\*).

**add** 추가 이벤트 설명(\*).

**alter** 변경(Alter) 이벤트 설명(\*).

**drop** 삭제(Drop) 이벤트 설명(\*).

**invalidate**

무효화 이벤트 설명(\*).

**권한 부여**

권한 부여 이벤트 설명(\*).

**revoke**

권한 취소 이벤트 설명(\*).

(\* ) 각 이벤트 필드에 다음과 같은 하위 필드가 있습니다.

**{event}type**

이벤트 유형(START, STOP, READ, WRITE, GET).

**{event}desc**

이벤트 설명(이벤트 정보가 있는 헤더).

**{event}state**

이벤트 상태(success, failure, start, stop, in progress, idle) 또는 이벤트 진행(%).

**{event}attr**

이벤트 속성(business level, cached, sync, async, internal, external, logical, physical, auto, manual, temporary, permanent).

**{event}objid**

고유한 오브젝트 ID(TABLE, CFG, DBM).

**{event}objname**

이벤트 오브젝트 이름(예: "schema.tablename").

**{event}objdata**

오브젝트 데이터(오브젝트가 문자열 또는 단순 정수 유형이 아닌 경우(예: 데이터 구조 또는 일부 복합 유형)에 사용됨).

**{event}qtype**

이벤트 규정자 유형(FROM, TO, ON, FOR, AT, BY, CONTEXT).

**{event}qname**

이벤트 규정자 이름/값(예: FOR "DB ABC").

### **{event}qhdr**

이벤트 규정자 데이터 헤더(포함된 파일 유형, 텍스트 설명 및 데이터 크기). `{event}qdata` 필드와 함께 사용됩니다.

### **{event}qdata**

이벤트 규정자 데이터(규정자가 문자열 또는 단순 정수 유형이 아닌 경우(예: 일부 데이터 구조 또는 복합 유형)에 사용됨).

위에서, 키워드 **{event}**는 특정 이벤트의 이벤트 유형(예: start, stop, change, read, write)으로 대체해야 합니다.

항상 필드 이름 앞에 텍스트를 표시하려면(예를 들어 필수 필드의 경우) % 필드 접두부를 사용해야 합니다. 필드에 데이터가 포함되는 경우 필드 이름 앞에 텍스트를 표시하려면 @ 필드 접두부를 사용해야 합니다. 해당되는 텍스트 설명과 함께 필수 및 선택적 필드의 조합이 허용됩니다.

`wn`, `wr`, `wf`, `wv` 및 `wt` 특수 문자가 형식 문자열에서 인식됩니다.

다른 필드와는 반대로, 데이터 및 인수 필드는 몇 개의 섹션을 포함할 수 있습니다. 특정 섹션을 출력하려면 필드 이름 다음에 `[n]`을 추가하십시오. 여기서 `n`은 섹션 번호( $1 \leq n \leq 64$ )입니다. 예를 들어, 첫 번째 데이터 오브젝트와 두 번째 데이터 설명 섹션을 출력하려면 `{dataobj}[1]` 및 `{datadesc}[2]`를 사용하십시오. `[n]`을 사용하지 않는 경우, 로그된 모든 섹션은 로그 메시지에 표시되는 대로 사전 형식화된 로그 데이터를 사용하여 출력됩니다. 따라서 각 데이터 필드, 인수 필드 또는 섹션 이전에 적용 가능한 텍스트 설명과 구분하는 줄 바꾸기를 추가하지 않아도 됩니다.

### **-filter** *fieldPatternList* | **-g** *fieldPatternList*

*fieldPatternList*는 `fieldName operator searchPattern` 형식의 필드 패턴 쌍 목록(쉼표로 구분)입니다.

연산자는 다음 중 하나가 될 수 있습니다.

- =** 전체 단어를 형성하는, 일치사항을 포함하는 레코드만 선택합니다(단어 검색).
- :=** 검색 패턴이 대형 표현식의 일부가 될 수 있는, 일치사항을 포함하는 레코드를 선택합니다.
- !=** 일치하지 않는 라인만 선택합니다(역단어 일치).
- !:=** 검색 패턴이 대형 표현식의 일부가 될 수 있는 일치하지 않는 행만 선택합니다.
- ^=** 필드 값이 지정된 검색 패턴으로 시작하는 레코드를 선택합니다.
- !^=** 필드 값이 지정된 검색 패턴으로 시작하지 않는 레코드를 선택합니다.

이 옵션에 대해 % 및 @ 접두부가 사용되지 않는 것을 제외하고, -fmt 옵션에 대해 설명된 대로 사용할 수 있습니다.

**-gi** *fieldPatternList*

-g와 같지만, 대소문자를 구분하지 않습니다.

**-gv** *fieldPatternList*

지정된 패턴과 일치하지 않는 메시지를 검색합니다.

**-gvi** | **-giv** *fieldPatternList*

-gv와 같지만, 대소문자를 구분하지 않습니다.

**-pid** *processIDList*

프로세스 ID가 나열된 로그 메시지만 표시합니다.

**-tid** *threadIDList*

스레드 ID가 나열된 로그 메시지만 표시합니다.

**-eduid** *EduIDList*

쉽표로 구분되는 하나 이상의 숫자 값을 포함하는 EDU ID 목록에서 지정된 EDU ID를 가지고 있는 모든 레코드를 찾습니다.

**-n** | **-node** *nodeList*

데이터베이스 파티션 번호가 나열된 로그 메시지만 표시합니다.

**-e** | **-error** *errorList*

오류 번호가 나열된 로그 메시지만 표시합니다.

**-l** | **-level** *levelList*

쉽표로 구분되는 하나 이상의 텍스트 값(Info, Warning, Error, Severe, Critical 및 Event)을 포함하는 심각도 레벨 목록에서 지정된 심각도 레벨의 모든 레코드를 찾습니다.

**-c** | **-count**

발견된 레코드 수를 표시합니다.

**-v** | **-invert**

지정된 패턴과 일치하지 않는 모든 레코드를 선택하기 위해 패턴 일치를 역으로 합니다.

**-strict** 라인마다 하나의 필드 값을 사용하여 레코드를 표시합니다. 비어 있는 모든 필드는 건너뛩니다. 구문 분석을 단순화하기 위해 스크립트에 사용할 수 있습니다.

**-V** | **-verbose**

비어 있는 필드를 포함하여 모든 필드를 출력합니다.

**-exist** 검색 요청 시 레코드의 필드가 처리되는 방법을 정의합니다. 이 옵션이 지정된 경우, 필드는 처리 순서로 되어 있어야 합니다.



**-cbe** CBE(Common Base Event) 정규 상황 데이터.

**-o | -output** *pathName*

완전한 *pathName*으로 지정된 파일로 출력을 저장합니다.

**-f | -follow**

출력 파일이 일반 파일인 경우, 입력 파일의 마지막 레코드가 처리된 후 도구가 종료하지 않음을 지정합니다. 대신, 특정 시간 간격(*sleepInterval*) 동안 유ힴ 상태가 된 후, 사용 가능하게 되면 입력 파일에서 추가 레코드를 읽어서 처리하려고 합니다.

follow 모드는 회전 db2diag 로그 파일도 처리합니다. 예를 들어, 명령은 사용 중인 최신 회전 진단 로그 파일(db2diag.23.log)을 읽은 후 db2diag.23.log 로그 파일이 크기 한계에 도달하면 다음으로 작성된 회전 로그 파일(db2diag.24.log)을 읽습니다.

이 옵션은 다른 프로세스에 의해 파일에 기록되는 레코드를 모니터할 때 사용할 수 있습니다. *startTime* 옵션을 지정하여 이후로 로그된 모든 레코드를 표시할 수 있습니다. *startTime* 옵션은 YYYY-MM-DD-hh.mm.ss.nnnnnn 형식을 사용하여 지정합니다.

YYYY 연도를 지정합니다.

MM 1년 중 월을 지정합니다(01 - 12).

DD 한 달 중 일을 지정합니다(01 - 31).

hh 하루 중 시를 지정합니다(00 - 23).

mm 1시간 중 분을 지정합니다(00 - 59).

ss 1분 중 초를 지정합니다(00 - 59).

nnnnnn

UNIX 운영 체제에서는 마이크로초를, Windows 운영 체제에서는 밀리초를 지정합니다.

연도 필드 다음에 오는 일부 또는 모든 필드는 생략할 수 있습니다. 생략하면 디폴트 값이 사용됩니다. 월 및 일의 디폴트값은 1이고, 다른 모든 필드의 디폴트값은 0입니다.

레코드 시간소인에 대한 완전 일치가 진단 로그 파일에 없는 경우, 지정된 시간 이전의 가장 가까운 시간이 사용됩니다.

*sleepInterval* 옵션은 유ힴ 간격(초)을 지정합니다. 가장 작은 시간 단위가 필요하면 부동 소수점 값으로 지정할 수 있습니다. 디폴트값은 2초입니다.

**-H | -history**

지정된 시간 간격 동안 로그된 메시지의 실행기록을 표시합니다. 이 옵션은 다음 옵션과 함께 지정할 수 있습니다.

### *historyPeriod*

가장 최근에 로그된 레코드에서 시작하여, *historyPeriod*에 지정된 지속기간 동안 로그된 메시지가 표시됨을 지정합니다. *historyPeriod* 옵션은 Number *timeUnit* 형식을 사용하여 지정됩니다. 여기서 Number는 시간 단위 수이고 *timeUnit*는 시간 단위 유형(M(월), d(일), h(시), m(분), s(초))을 표시합니다. Number의 디폴트값은 30이고, *timeUnit*의 디폴트값은 m입니다.

### *historyPeriod:historyBegin*

*historyBegin*에 지정된 시간에서 시작하여, *historyPeriod*에 지정된 지속기간 동안 로그된 메시지가 표시됨을 지정합니다.

형식은 *YYYY-MM-DD-hh.mm.ss.nnnnnn*입니다.

*YYYY* 연도를 지정합니다.

*MM* 1년 중 월을 지정합니다(01 - 12).

*DD* 한 달 중 일을 지정합니다(01 - 31).

*hh* 하루 중 시를 지정합니다(00 - 23).

*mm* 1시간 중 분을 지정합니다(00 - 59).

*ss* 1분 중 초를 지정합니다(00 - 59).

*nnnnnn*

마이크로초(UNIX 운영 체제의 경우)나 밀리초(Windows 운영 체제)를 지정합니다.

### **-t | -time**

시간소인 값을 지정합니다. 이 옵션은 다음 옵션 중 하나 또는 둘 다와 함께 지정할 수 있습니다.

#### *startTime*

*startTime* 이후에 로그된 모든 메시지를 표시합니다.

#### *:endTime*

*endTime* 이전에 로그된 모든 메시지를 표시합니다.

*startTime* 및 *endTime* 사이에 로그된 메시지를 표시하려면 **-t *startTime:endTime*** 을 지정하십시오.

형식은 *YYYY-MM-DD-hh.mm.ss.nnnnnn*입니다.

*YYYY* 연도를 지정합니다.

*MM* 1년 중 월을 지정합니다(01 - 12).

*DD* 한 달 중 일을 지정합니다(01 - 31).

*hh* 하루 중 시를 지정합니다(00 - 23).

*mm* 1시간 중 분을 지정합니다(00 - 59).

*ss* 1분 중 초를 지정합니다(00 - 59).

*nnnnnnn*

마이크로초(UNIX 운영 체제의 경우)나 밀리초(Windows 운영 체제)를 지정합니다.

연도 필드 다음에 오는 일부 또는 모든 필드는 생략할 수 있습니다. 생략하면 디폴트 값이 사용됩니다. 월 및 일의 디폴트값은 1이고, 다른 모든 필드의 디폴트값은 0입니다.

레코드 시간소인에 대한 완전 일치가 진단 로그 파일에 없는 경우, 지정된 시간소인에 가장 가까운 시간이 사용됩니다.

#### **-A | -archive** *dirName*

단일 및 회전 로그 파일 둘 다를 아카이브합니다. 이 옵션을 지정하면 다른 옵션은 모두 무시됩니다. 하나 이상의 파일 이름을 지정하는 경우 각 파일은 개별적으로 처리됩니다. YYYY-MM-DD-hh.mm.ss 형식의 시간소인이 파일 이름에 추가됩니다.

아카이브되는 파일 및 디렉토리의 이름을 지정할 수 있습니다. 디렉토리를 지정하지 않으면 파일은 파일이 있는 디렉토리에서 아카이브되고 디렉토리 이름은 파일 이름에서 추출됩니다.

디렉토리는 지정하고 파일 이름은 지정하지 않는 경우 현재 디렉토리에서 db2diag 로그 파일을 검색합니다. 발견되면, 지정된 디렉토리에서 파일이 아카이브됩니다. 파일을 찾을 수 없으면 **diagpath** 구성 매개변수로 지정된 디렉토리에서 db2diag 로그 파일을 검색합니다. 발견되면 지정된 디렉토리에서 아카이브됩니다.

파일 또는 디렉토리를 지정하지 않는 경우 현재 디렉토리에서 db2diag 로그 파일을 검색합니다. 발견되면 현재 디렉토리에서 아카이브됩니다. 파일을 찾을 수 없으면 **diagpath** 구성 매개변수로 지정된 디렉토리에서 db2diag 로그 파일을 검색합니다. 발견되면 **diagpath** 구성 매개변수에서 지정된 디렉토리에서 아카이브됩니다.

#### **-readfile**

터미널 입력을 무시하고 강제로 진단 로그 파일에서 읽습니다. 이 옵션은 db2diag가 터미널이 아닌 파일에서 읽도록 하기 위해 스크립트에서 사용할 수 있습니다. 특히, stdin이 사용되지 않거나 자동화된 도구가 사용되는 상황에서 사용됩니다.

#### **-rc** *rcList | switch*

특정의 ZRC 또는 ECF 16진수나 음수 10진수 리턴 코드의 공백으로 구분되

는 목록(rcList)에 대한 DB2 내부 오류 리턴 코드 설명을 표시합니다. 다음 스위치 중 하나를 지정하여 ZRC 또는 ECF 리턴 코드의 전체 목록을 표시할 수 있습니다.

**zrc** DB2 ZRC 리턴 코드의 간단한 설명을 표시합니다.

**ecf** DB2 ECF 리턴 코드의 간단한 설명을 표시합니다.

**html** DB2 ZRC 리턴 코드의 간단한 설명을 HTML 형식으로 표시합니다.

이 옵션을 지정하면 다른 옵션은 모두 무시되고 출력은 화면으로 보냅니다.

**-ecfid** *ecfid*

숫자 *ecfId*에서 추출된 함수 정보를 표시합니다. 이 옵션을 지정하면 다른 옵션은 모두 무시됩니다.

**예:**

중요한 모든 오류 메시지를 표시하려면 다음을 입력하십시오.

```
db2diag -level critical
```

또는

```
db2diag -g 'level=Critical'
```

프로세스 ID(PID)가 52356인 프로세스에 의해 노드 1, 2 또는 3에서 생성되는 심각한 모든 오류 메시지를 표시하려면 다음을 입력하십시오.

```
db2diag -g level=Severe,pid=952356 -n 1,2,3
```

데이터베이스 SAMPLE 및 인스턴스 aabrashk를 포함하는 모든 메시지를 표시하려면 다음을 입력하십시오.

```
db2diag -g db=SAMPLE,instance=aabrashk
```

데이터베이스 필드를 포함하는 모든 심각한 오류 메시지를 표시하려면 다음을 입력하십시오.

```
db2diag -g db:= -gi level=severe
```

DB2 ZRC 리턴 코드 0x87040055와, 응용프로그램 ID

G916625D.NA8C.068149162729를 포함하는 모든 오류 메시지를 표시하려면 다음을 입력하십시오.

```
db2diag -g msg:=0x87040055 -l Error | db2diag -gi appid^=G916625D.NA
```

LOADID 데이터를 포함하지 않는 모든 메시지를 표시하려면 다음을 입력하십시오.

```
db2diag -gv data:=LOADID
```

응용프로그램 ID 필드에 LOCAL 패턴을 포함하지 않는 로그된 레코드만 표시하려면 다음을 입력하십시오.

```
db2diag -gi appid!:=local or db2diag -g appid!:=LOCAL
```

일치하지 않는 모든 레코드가 표시됩니다. 응용프로그램 ID 필드가 있는 메시지만 출력하려면 다음을 입력하십시오.

```
db2diag -gvi appid:=local -exist
```

시간소인이 2003-03-03-12.16.26.230520인 메시지와 이후에 로그된 모든 메시지를 표시하려면 다음을 입력하십시오.

```
db2diag -time 2003-03-03-12.16.26.230520
```

마지막 3일 동안 심각한 오류를 표시하려면 다음을 입력하십시오.

```
db2diag -gi "level=severe" -H 3d
```

함수 필드에 대해 pdLog 패턴과 일치하지 않는 모든 로그 메시지를 표시하려면 다음을 입력하십시오.

```
db2diag -g 'funcname!=pdLog' or db2diag -gv 'funcn=pdLog'
```

"base sys에서 시작하는 구성요소 이름을 포함하는 모든 심각한 오류 메시지를 표시하려면 다음을 입력하십시오.

```
db2diag -l severe | db2diag -g "comp^=base sys"
```

db2diag.log 파일의 증가를 보려면 db2diag -f db2diag.log를 입력하십시오. 그러면 현재 디렉토리에 있는 db2diag.log 파일에 기록된 모든 레코드가 표시됩니다. 레코드는 파일에 추가되는 대로 표시됩니다. Ctrl-C를 누를 때까지 계속 표시됩니다.

db2diag.log의 컨텍스트를 /home/user/Logs 디렉토리에 위치한 db2diag\_123.log 파일에 기록하려면 다음을 입력하십시오.

```
db2diag -o /home/user/Logs/db2diag_123.log
```

디폴트 설정을 사용하여 Perl 스크립트에서 db2diag를 호출하려면 다음을 입력하십시오.

```
system("db2diag -readfile");
```

그러면 db2diag가 **DIAGPATH** 구성 매개변수에 지정된 디렉토리에서 db2diag.log/db2diag.\*.log 파일(데이터베이스 관리 프로그램 **DIAGSIZE** 구성 매개변수가 설정된 경우 회전 로그)을 처리하도록 강요합니다.

터미널 입력을 무시하고 지정된 디렉토리에서 db2diag.log1 파일을 읽으려면 다음을 입력하십시오.

```
system("db2diag -readfile /u/usr/sql/lib/db2dump/db2diag.log1");
```

ecfId = 0x1C30000E에 해당되는 함수 정보를 표시하려면 다음을 입력하십시오.

```
db2diag -ecfid 0x1C30000E
```

이는 다음과 같습니다.

```
db2diag -ecfid 472907790
```

이 명령은 함수 이름, 구성요소 및 제품 이름을 표시합니다.

eduid = 123을 포함하는 로그된 레코드만 표시하려면 다음을 입력하십시오.

```
db2diag -eduid 123
```

eduid = 123 또는 eduid = 5678을 포함하는 레코드만 표시하려면 다음을 입력하십시오.

```
db2diag -eduid "123,5678"
```

eduid = 15 스레드에서 생성되는 모든 심각한 오류 메시지를 표시하려면 다음을 입력하십시오.

```
db2diag -g "level=Severe, eduid=15"
```

또는 다음과 같습니다.

```
db2diag -g level=Severe | db2diag -eduid 15
```

## 사용 시 참고사항

- 각 옵션은 한 번만 표시될 수 있습니다. 어떤 순서로도 지정할 수 있으며 선택적 매개변수를 가질 수 있습니다. 짧은 옵션은 함께 포함할 수 없습니다. 예를 들어, `-le`가 아닌 `-l -e`를 사용하십시오.
- 디폴트로 `db2diag`는 현재 디렉토리에서 `db2diag` 로그 파일을 찾습니다. 파일을 찾을 수 없으면 `diagpath` 구성 매개변수로 설정된 디렉토리가 그 다음으로 검색됩니다. `db2diag` 로그 파일을 찾을 수 없는 경우 `db2diag`는 오류를 리턴하고 종료합니다.
- 필터링 및 형식화 옵션을 단일 명령행에서 조합하여 파이프를 사용하는 복잡한 검색을 수행할 수 있습니다. 형식화 옵션 `-fmt`, `-strict`, `-cbe` 및 `-verbose`는 표준 필드(사용자가 생략하거나 정의한 필드가 아님)가 있는 원래의 로그된 메시지만 필터되도록 모든 필터링이 수행된 후에만 사용해야 합니다. 파이프를 사용할 때 `-l`를 사용할 필요는 없습니다.
- 파이프가 사용되고 하나 이상의 파일 이름을 명령행에 지정한 경우, `db2diag` 입력은 `-l` 지정 여부에 따라 다르게 처리됩니다. `-l`를 생략하면 지정된 파일에서 입력을 가져옵니다. 반대로, `-l` 옵션을 지정하면 파일 이름은 무시되고(명령행에 있어도) 터미널의 입력이 사용됩니다. 파이프를 사용하고 파일 이름을 지정하지 않을 경우, `db2diag` 입력은 명령행에 `-l`를 지정하거나 지정하지 않고 정확히 동일한 방법으로 처리됩니다.

- -exist 옵션은 패턴과 일치하지 않는 모든 레코드가 적절한 필드를 포함하는지 여부에 관계없이 출력될 때 역일치 검색에 대한 디폴트 db2diag 동작보다 우선합니다. -exist 옵션을 지정한 경우 요청된 필드를 포함하는 레코드만 처리되어 출력됩니다.
- -fmt(형식) 옵션을 지정하지 않은 경우, 모든 메시지(필터되지 않은)는 진단 로그 파일에 기록된 대로 정확히 출력됩니다. 출력 레코드 형식은 -strict,-cbe 및 -verbose 옵션을 사용하여 변경할 수 있습니다.
- -fmt 옵션은 -strict,-cbe 및 -verbose 옵션보다 우선합니다.
- 일부 제한사항은 -cbe 옵션이 지정되고 db2diag 로그 파일이 원래 컴퓨터에서 네트워크를 통해 전송된 경우에 적용됩니다. db2diag 도구는 DB2에 대한 정보와 컴퓨터 호스트 이름을 로컬로 수집합니다. 이는 DB2 버전과 로컬 시스템의 소스 또는 리포터 componentID 위치 필드가 원래 컴퓨터에서 사용한 해당 값과 다를 수 있음을 의미합니다.
- 스크립트에서 db2diag를 사용할 때 -readfile 옵션을 사용하도록 하십시오. 그러면 터미널 입력을 무시하고 파일에서 읽습니다.
- 보통, 종료 상태는 일치사항이 발견된 경우 0이고 발견되지 않으면 1입니다. 입력 데이터 및 패턴에 구문 오류가 있거나, 입력 파일에 액세스할 수 없거나, 다른 오류가 발견되는 경우 종료 상태는 2입니다.
- DB2 텍스트 검색에서 발생하는 심각한 오류는 db2diag 로그 파일에 로그된 것으로 발견될 수 있습니다.
- 이 도구를 사용하여 회전 DB2DIAG 로그 파일을 읽고 필터하면(diagsize 데이터베이스 구성 매개변수가 0이 아닌 경우) 모든 회전 진단 로그 파일이 일련의 최대 10개 파일까지 읽고 필터됩니다.





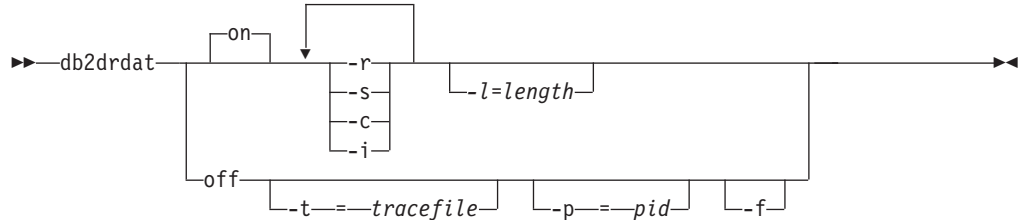
## 제 186 장 db2drdat - DRDA 추적

사용자가 DRDA 응용프로그램 리퀘스터(AR)와 DB2 DRDA 응용프로그램 서버(AS) 사이에 교환되는 DRDA 데이터 스트림을 캡처할 수 있게 합니다. 이 도구가 문제점 판별에 가장 자주 사용되지만, 응용프로그램을 실행하기 위해 필요한 보내기 및 수신 수를 판별하여 클라이언트/서버 환경에서 성능 조정에 사용할 수도 있습니다.

### 권한 부여

없음

### 명령 구문



### 명령 매개변수

- on** AS 추적 이벤트(지정된 것이 없으면 모두)를 켭니다.
- off** AS 추적 이벤트를 끕니다.
- r** DRDA AR에서 수신되는 DRDA 요청을 추적합니다.
- s** DRDA AR로 전송된 DRDA 응답을 추적합니다.
- c** 호스트 시스템의 DRDA 서버에서 수신되는 SQLCA를 추적합니다. 이것은 널(*null*)이 아닌 SQLCA의 형식화되고 읽기 쉬운 버전입니다.
- i** 추적 정보에 시간소인을 포함합니다.
- l** 추적 정보를 저장하는 데 사용되는 버퍼의 크기를 지정합니다.
- p** 이 프로세스에 대한 이벤트만 추적합니다. **-p**가 지정되지 않는 경우 서버에서 수신 DRDA 연결을 갖는 모든 에이전트가 추적됩니다. 추적할 *pid*는 LIST APPLICATIONS 명령에 의해 리턴되는 에이전트 필드에서 찾을 수 있습니다.
- t** 추적 대상을 지정합니다. 전체 경로 없이 파일 이름이 지정되는 경우 누락된 정보는 현재 경로에서 옵니다. *tracefile*이 지정되지 않는 경우 메시지는 현재 디렉토리의 db2drdat.dmp로 방향지정됩니다.
- f** 통신 버퍼를 형식화합니다.

## 사용 시 참고사항

db2drdat가 사용 중인 경우 db2trc 명령을 발행하지 마십시오.

db2drdat는 다음 정보를 *tracefile*에 기록합니다.

1. -r
  - DRDA 요청의 유형
  - 수신 버퍼
2. -s
  - DRDA 응답/오브젝트의 유형
  - 보내기 버퍼

명령은 종료 코드를 리턴합니다. 영(0) 값은 명령이 성공적으로 완료했음을 표시하고, 0이 아닌 값은 명령이 실패했음을 표시합니다. db2drdat가 이미 존재하는 파일로 출력을 보내는 경우, 파일의 권한이 해당 파일을 지우도록 허용하는 경우에는 이전 파일이 지워집니다. 그 경우에 운영 체제가 오류를 리턴합니다.

---

## 제 187 장 db2drvmp - DB2 데이터베이스 드라이브 맵

MSCS(Microsoft Cluster Server)의 데이터베이스 드라이브를 맵핑합니다. 이 명령은 Windows 플랫폼에서만 사용할 수 있습니다.

### 권한 부여

Windows 레지스트리 및 클러스터 레지스트리에 대한 읽기/쓰기 액세스.

### 필수 연결

인스턴스. 인스턴스 첨부이 없는 경우 응용프로그램이 디폴트 인스턴스 첨부을 작성합니다.

### 명령 구문

```
▶▶ db2drvmp {add | drop | query | reconcile} dbpartition_number from_drive to_drive ▶▶
```

### 명령 매개변수

**add** 새 데이터베이스 드라이브 맵을 지정합니다.

**drop** 기존 데이터베이스 드라이브 맵을 제거합니다.

**쿼리** 데이터베이스 맵을 쿼리합니다.

#### reconcile

레지스트리 콘텐츠가 손상되거나 실수로 삭제된 경우 레지스트리에 데이터베이스 드라이브 맵핑을 재적용합니다.

#### dbpartition\_number

데이터베이스 파티션 번호. 이 매개변수는 추가 및 삭제(drop) 조작에 필요합니다. 이 매개변수가 조정 조작에 지정되지 않으면 db2drvmp가 모든 데이터베이스 파티션에 대한 맵핑을 조정합니다.

#### from\_drive

맵핑할 드라이브 문자. 이 매개변수는 추가 및 삭제(drop) 조작에 필요합니다. 이 매개변수가 조정 조작에 지정되지 않으면 db2drvmp가 모든 드라이브에 대한 맵핑을 조정합니다.

#### to\_drive

맵핑할 드라이브 이름. 이 매개변수는 추가 조작에 필요합니다. 다른 조작에는 적용되지 않습니다.

**예:**

NODE0에 대해 F:에서 E:로의 데이터베이스 드라이브 맵핑을 설정하려면 다음 명령을 실행하십시오.

```
db2drvmp add 0 F E
```

NODE1에 대해 E:에서 F:로의 데이터베이스 드라이브 맵핑을 설정하려면 다음 명령을 실행하십시오.

```
db2drvmp add 1 E F
```

## 사용 시 참고사항

1. 데이터베이스 드라이브 맵핑은 테이블 스페이스, 컨테이너 또는 다른 데이터베이스 스토리지 오브젝트에 적용되지 않습니다.
2. 데이터베이스 드라이브 맵핑에 대한 변경 또는 설정은 즉시 적용되지 않습니다. 데이터베이스 드라이브 맵핑을 활성화하려면 Microsoft Cluster Administrator 도구를 사용하여 DB2 자원을 오프라인으로 설정한 다음 다시 온라인으로 설정하십시오.
3. DB2MCS.CFG 파일에서 TARGET\_DRVMAP\_DISK 키워드를 사용하면 드라이브 맵핑이 자동으로 수행됩니다.

---

## 제 188 장 db2empfa - 다중페이지 파일 할당 사용

데이터베이스에 대한 다중페이지 파일 할당을 사용할 수 있게 합니다. SMS 테이블 스페이스에 다중페이지 파일 할당이 사용 가능하면 디스크 스페이스는 한 번에 한 페이지가 아니라 한 번에 하나의 Extent가 할당됩니다.

### 범위

이 명령은 명령이 실행되는 데이터베이스 파티션에만 영향을 미칩니다.

### 권한 부여

*sysadm*

### 필수 연결

없음. 이 명령은 데이터베이스 연결을 설정합니다.

### 명령 구문

▶▶—db2empfa—*database-alias*—————▶▶

### 명령 매개변수

*database-alias*

다중페이지 파일 할당이 사용 가능해야 하는 데이터베이스의 별명을 지정합니다.

### 사용 시 참고사항

이 유틸리티는 다음과 같습니다.

- 독점 모드에서 데이터베이스 파티션(적용 가능한 경우)에 연결합니다.
- 모든 SMS 테이블 스페이스에서 하나의 Extent보다 더 큰 모든 데이터 및 인덱스 파일에서 마지막 Extent까지 채울 빈 페이지를 할당합니다.
- 데이터베이스 구성 매개변수 **multipage\_alloc** 값을 YES로 변경합니다.
- 연결을 끊습니다.

db2empfa가 독점 모드로 데이터베이스 파티션에 연결하므로, 카탈로그 데이터베이스 파티션 또는 다른 모든 데이터베이스 파티션에서 동시에 실행될 수 없습니다.



---

## 제 189 장 db2envar.bat - 현재 명령 창의 환경 설정

db2envar.bat가 실행된 DB2 사본의 현재 명령 창 환경을 설정합니다. 명령행에서 다른 DB2 사본 간을 전환하려는 경우 유용합니다. 이 명령은 Windows 운영 체제에서만 사용할 수 있습니다.

### 권한 부여

없음

### 필요한 연결

없음

### 명령 구문

▶—db2envar.bat—▶

### 명령 매개변수

없음

### 사용 시 참고사항

머신에 여러 DB2 사본이 있는 경우 실행될 db2envar.bat를 표시하기 위해 전체 경로를 사용해야 합니다. 예를 들어, e:\sql1lib에 설치된 DB2 사본의 환경을 설정하려면 e:\sql1lib\bin\db2envar.bat를 실행해야 합니다.





---

## 제 190 장 db2eva - 이벤트 분석기

사용자가 테이블로 보낸 데이터가 있는 DB2 이벤트 모니터에서 생성된 성능 데이터를 추적할 수 있도록 하는 이벤트 분석기를 시작합니다.

**중요사항:** 이 명령은 제어 센터와 관련된 구성요소가 버전 9.7에서 사용되지 않기 때문에 사용되지 않으며 이후 릴리스에서 제거됩니다. 자세한 정보는 *버전 9.7의 새로운 내용* 책에 있는 『제어 센터 도구 및 DB2 Administration Server(DAS)는 사용되지 않음』 주제를 참조하십시오.

### 권한 부여

다음 특권 및 권한 전체가 필요합니다.

- CONNECT 권한(또는 내재적으로 CONNECT 권한을 포함하는 권한)
- 다음 카탈로그 테이블에 대한 SELECT 특권(또는 카탈로그 테이블에서 내재적으로 SELECT 특권을 포함하는 권한):
  - SYSIBM.EVENTMONITORS
  - SYSIBM.EVENTS
  - SYSIBM.EVENTTABLE

### 필수 연결

데이터베이스 연결

### 명령 구문

```
▶▶ db2eva [-db database-alias] [-evm evmon-name] ▶▶
```

### 명령 매개변수

db2eva 매개변수는 선택사항입니다. 매개변수를 지정하지 않을 경우, 데이터베이스 및 이벤트 모니터 이름을 묻는 프롬프트를 표시하기 위해 이벤트 분석기 열기 대화 상자가 표시됩니다.

#### **-db** *database-alias*

이벤트 모니터에 대해 정의된 데이터베이스의 이름을 지정합니다.

#### **-evm** *evmon-name*

해당 추적을 분석할 이벤트 모니터의 이름을 지정합니다.

## 사용 시 참고사항

필요한 액세스 권한이 없으면, 사용자는 이벤트 모니터 데이터를 검색할 수 없습니다.

이벤트 모니터 추적을 검색하는 방법은 두 가지입니다.

1. 사용자는 명령행에서 db2eva를 입력할 수 있고 사용자가 확인을 눌러 이벤트 분석기 대화 상자를 열기 전에 드롭다운 목록에서 데이터베이스 및 이벤트 모니터 이름을 선택할 수 있는 이벤트 분석기 열기 대화 상자가 열립니다.
2. 사용자는 명령행에서 -db 및 -evm 매개변수를 지정할 수 있고 이벤트 분석기 대화 상자는 지정된 데이터베이스에 대해 열립니다.

이벤트 분석기는 데이터베이스에 연결하고, SYSIBM.SYSEVENTTABLES에서 대상 선택을 발행하여 이벤트 모니터 테이블을 가져옵니다. 그런 다음 필요한 데이터가 검색된 후에 연결이 해제됩니다.

이벤트 분석기를 사용하여 활성 이벤트 모니터가 생성하는 데이터를 분석할 수 있습니다. 그러나, 이벤트 분석기가 호출된 이후에 캡처된 이벤트 모니터는 표시되지 않을 수 있습니다. 이벤트 분석기를 호출하기 전에 이벤트 모니터를 꺼서 데이터가 적절하게 표시되는지 확인하십시오.

---

## 제 191 장 db2evmon - 이벤트 모니터 제작 도구

이벤트 모니터 파일 및 Named Pipe 출력을 형식화하고 이를 표준 출력에 씁니다.

### 권한 부여

데이터베이스에 연결하는 경우를 제외하고는 없음(-db -evm). 연결하려면 다음 권한 전체가 필요합니다.

- CONNECT 권한(또는 내재적으로 CONNECT를 포함하는 권한)
- 다음 카탈로그 테이블에 대한 SELECT 특권(또는 카탈로그 테이블에서 내재적으로 SELECT를 포함하는 권한):
  - SYSIBM.SYSTABLES
  - SYSIBM.SYSEVENTMONITORS

이벤트 모니터가 db2detaildeadlock인 경우 다음 권한이나 특권 중 하나가 추가적으로 필요합니다.

- *sysmon*
- *sysmaint*
- *sysctrl*
- *sysadm*
- SNAPSHOT\_DATABASE 테이블 함수에 대한 EXECUTE 특권
- 데이터 액세스

### 필수 연결

없음

### 명령 구문

```
▶▶ db2evmon [ -db database-alias -evm event-monitor-name ] [ -path event-monitor-target ] ▶▶
```

### 명령 매개변수

**-db** *database-alias*

데이터를 표시할 데이터베이스를 지정합니다. 이 매개변수는 대소문자가 구분됩니다.

**-evm** *event-monitor-name*

이벤트 모니터의 한 부분 이름입니다. 일반 또는 구분된 SQL ID입니다. 이 매개변수는 대소문자가 구분됩니다.

**-path** *event-monitor-target*

이벤트 모니터 추적 파일이 있는 디렉토리를 지정합니다.

## 사용 시 참고사항

db2evmon은 명령이 데이터베이스에 연결되거나 경로 옵션을 지정하고 발행되는지 여부와 상관 없이 동일한 출력을 생성합니다.

- db2evmon이 **-db** 및 **-evm** 옵션과 함께 발행될 때 인스턴스가 아직 시작되지 않은 경우 명령이 인스턴스를 시작합니다.
- db2evmon이 **-path** 옵션과 함께 발행될 때 인스턴스가 아직 시작되지 않은 경우 명령은 인스턴스를 시작하지 않습니다. 인스턴스를 명시적으로 시작해야 합니다.

데이터가 파일에 기록되고 있는 경우 도구는 표준 출력을 사용하여 표시하도록 파일을 형식화합니다. 이 경우 모니터가 먼저 켜지고 파일의 모든 이벤트 데이터가 도구에 의해 표시됩니다. 도구가 실행된 후 파일에 기록된 모든 데이터를 보려면 db2evmon을 다시 실행하십시오.

데이터가 파이프에 기록되고 있는 경우 도구는 이벤트가 발생할 때 표준 출력을 사용하여 표시하도록 출력을 형식화합니다. 이 경우에 도구는 모니터가 켜지기 전에 시작됩니다.

---

## 제 192 장 db2evtb1 - 이벤트 모니터 목표 테이블 정의 생성

SQL 테이블에 작성되는 이벤트 모니터를 정의할 때 사용할 수 있는 샘플 CREATE EVENT MONITOR SQL문을 생성합니다.

### 권한 부여

없음

### 필수 연결

없음

### 명령 구문

```
▶--db2evtb1 [ -schema schema-name ] [ -partitioned ]
▶--evm event-monitor-name [ event type ]
```

### 명령 매개변수

**-schema** *schema-name*

스키마 이름 지정되지 않으면 테이블 이름이 규정되지 않습니다.

**-partitioned**

지정되는 경우 파티션된 데이터베이스 환경에만 적용 가능한 요소 또한 생성됩니다.

**-evm** *event-monitor-name*

이벤트 모니터의 이름.

*event type*

CREATE EVENT MONITOR 문에서 사용 가능한 이벤트 유형(예: DATABASE, TABLES, UNIT OF WORK)

예:

```
db2evtb1 -schema smith -evm foo database, tables, tablespaces, bufferpools
```

### 사용 시 참고사항

출력은 표준 출력에 작성됩니다.

WRITE TO TABLE 이벤트 모니터 정의는 db2evtbl 도구를 사용하는 경우 더 직접적인 방법입니다. 예를 들어, 다음 단계를 수행하여 이벤트 모니터를 정의 및 활성화할 수 있습니다.

1. db2evtbl을 사용하여 CREATE EVENT MONITOR문을 생성합니다.
2. 불필요한 컬럼을 제거하여 SQL문을 편집합니다.
3. CLP를 사용하여 SQL문을 처리합니다(CREATE EVENT MONITOR문이 실행되는 경우 목표 테이블이 작성됨).
4. SET EVENT MONITOR STATE를 실행하여 새 이벤트 모니터를 활성화합니다.

교착 상태 이벤트 모니터를 제외한 모든 이벤트를 플러시할 수 있으므로, 이벤트당 둘 이상의 레코드를 작성하는 경우 FLUSH EVENT MONITOR문을 사용하지 않는 사용자는 목표 테이블에서 evmon\_flushes 요소를 제외할 수 있습니다.

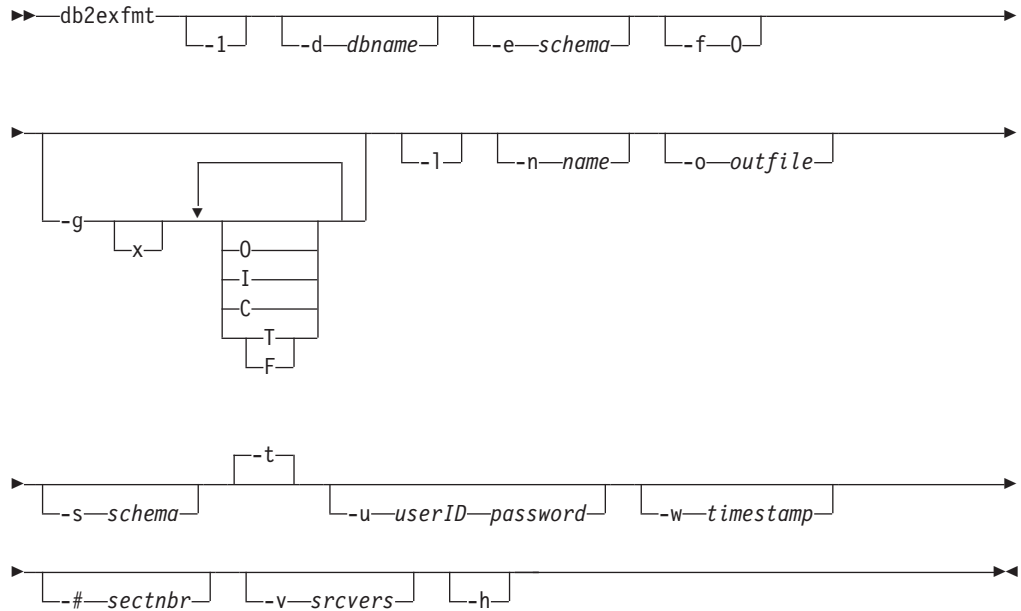
## 제 193 장 db2exfmt - Explain 테이블 형식

db2exfmt 도구를 사용하여 EXPLAIN 테이블의 콘텐츠를 형식화합니다. 이 도구는 인스턴스 sqllib 디렉토리의 misc 서브디렉토리에 있습니다. 이 도구는 스냅샷이 사용 가능한 경우 EXPLAIN 스냅샷의 통계를 사용합니다.

### 권한 부여

이 도구를 사용하려면 형식화되는 Explain 테이블에 대한 읽기 액세스가 필요합니다.

### 명령 구문



### 명령 매개변수

#### db2exfmt

옵션이 지정되지 않으면 명령은 대화식 모드에 들어가고 항목을 작성하도록 프롬프트됩니다.

**-l** 디폴트 -e % -n % -s % -v % -w -l -# 0을 사용합니다.

Explain 스키마가 제공되지 않는 경우 환경 변수 \$USER 또는 \$USERNAME의 콘텐츠가 디폴트로 사용됩니다. 이 변수가 없는 경우 사용자에게 Explain 스키마가 프롬프트됩니다.

**-d dbname**

패키지가 들어있는 데이터베이스의 이름.

**-e** *schema*

Explain 테이블 SQL 스키마.

**-f** 포매팅 플래그입니다. 이 릴리스에서 유일하게 지원되는 값은 O(연산자 요약)입니다.

**-g** 그래프 플랜.

**x** 옵션을 끕니다(디폴트는 켜는 것입니다).

-g만 지정되는 경우 모든 테이블에 대한 형식화 정보 뒤에 그래프가 생성됩니다. 그렇지 않으면 다음 유효 값의 모든 조합을 지정할 수 있습니다.

**O** 그래프만 생성합니다. 테이블 콘텐츠를 형식화하지 않습니다.

**T** 그래프의 각 연산자 아래에 전체 비용을 포함합니다.

**F** 그래프에 첫 번째 조 비용을 포함합니다.

**I** 그래프의 각 연산자 아래에 입출력 비용을 포함합니다.

**C** 그래프의 각 연산자의 예측한 출력 카디널리티(cardinality)(조 수)를 포함합니다.

상호 독점적인 F 및 T를 제외한 이들 옵션의 모든 조합이 허용됩니다.

**-l** 패키지 이름을 처리할 때 문자 크기를 고려합니다.

**-n** *name*

EXPLAIN 요청의 소스 이름(SOURCE\_NAME).

**-s** *schema*

EXPLAIN 요청 소스의 SQL 스키마 또는 규정자(SOURCE\_SCHEMA).

**-o** *outfile*

출력 파일 이름.

**-t** 출력을 터미널로 보냅니다.

**-u** *userID password*

데이터베이스에 연결할 때 제공되는 사용자 ID 및 암호를 사용하십시오.

사용자 ID와 암호는 이름 지정 규칙에 따라 유효해야 하며 데이터베이스가 인식해야 합니다.

**-w** *timestamp*

Explain 시간소인입니다. 최신 EXPLAIN 요청을 확보하려면 -1을 지정하십시오.

**-#** *sectnbr*

소스의 섹션 번호입니다. 모든 섹션을 요청하려면 0을 지정하십시오.



**-v** *srcvers*

Explain 요청 소스의 소스 버전(디폴트 %)

**-h** 도움말 정보를 표시합니다. 이 옵션을 지정하면 다른 모든 옵션은 무시되며 도움말 정보만 표시됩니다.

## 사용 시 참고사항

-h 및 -l 옵션의 경우를 제외하면 제공되지 않거나 불완전하게 지정되는 모든 매개변수 값에 대해 사용자에게 프롬프트합니다.

Explain 테이블 SQL 스키마가 제공되지 않는 경우, 환경 변수 `USER`의 값이 디폴트로 사용됩니다. 이 변수를 찾을 수 없는 경우 사용자에게 Explain 테이블 SQL 스키마를 프롬프트합니다.

소스 이름, 소스 SQL 스키마 및 Explain 시간소인은 LIKE 술어 양식으로 제공될 수 있으며, 이는 퍼센트 기호(%) 및 밑줄(\_)이 한 번의 호출로 다중 소스를 선택하기 위한 패턴 일치 문자로 사용될 수 있게 합니다. 최신 Explain문의 경우 Explain 시간은 -1로 지정할 수 있습니다.

-o가 파일 이름 없이 지정되고 -t가 지정되지 않는 경우 사용자에게 파일 이름에 대해 프롬프트됩니다(디폴트 이름은 `db2exfmt.out`). -o 및 -t가 지정되지 않는 경우 사용자에게 파일 이름에 대해 프롬프트됩니다(디폴트 옵션은 터미널 출력임). -o와 -t 둘 다 지정되면 출력은 터미널로 보냅니다.

`db2exfmt` 명령은 스냅샷이 사용 가능한 경우 EXPLAIN 스냅샷의 통계를 사용합니다. 그렇지 않으면 `db2exfmt`는 EXPLAIN\_OBJECT 테이블에 저장된 통계를 표시하고 시스템 카탈로그에서 직접 검색되는 일부 통계도 표시합니다.

다음은 EXPLAIN 스냅샷 예입니다.

```
db2 explain plan with snapshot for query
db2exfmt
```

또는

```
db2 set current explain mode yes
db2 set current explain snapshot yes
run the query
db2exfmt
```



---

## 제 194 장 db2exmig - Explain 테이블 이주

Explain 테이블을 이주합니다. db2exmig 명령을 발행 중이거나 데이터베이스에 연결하기 위해 사용되는 사용자 ID에 속하는 Explain 테이블이 이주됩니다. Explain 테이블 이주 도구는 기존 Explain 테이블의 이름을 바꾸고 EXPLAIN.DDL을 사용하여 새 테이블 세트를 작성한 후 기존 Explain 테이블의 콘텐츠를 새 테이블에 복사합니다. 마지막으로, 기존 Explain 테이블을 삭제합니다. db2exmig 명령은 Explain 테이블에서 사용자 추가 컬럼을 보존합니다.

### 권한 부여

db2exmig 응용프로그램 패키지가 바인드된 경우 필요한 권한은 다음 권한 중 하나입니다.

- *dbadm* 권한
- db2exmig 응용프로그램 패키지에 대한 EXECUTE 권한과 다음 시스템 카탈로그에 대한 SELECT 특권 또는 CONTROL 특권:
  - SYSCAT.COLUMNS
  - SYSCAT.TABLES
  - SYSCAT.REFERENCES

db2exmig 응용프로그램 패키지가 바인드되지 않은 경우 필요한 권한은 *dbadm* 권한입니다.

### 필요한 연결

없음

### 명령 구문

```
►► db2exmig -d dbname -e explain_schema -u userID password ◀◀
```

### 명령 매개변수

**-d** *dbname*

데이터베이스 이름을 지정합니다.

**-e** *explain\_schema*

이주할 Explain 테이블의 스키마 이름을 지정합니다.

**-u** *userID password*

현재 사용자의 ID 및 암호를 지정합니다.

## 사용 시 참고사항

db2bfd -b db2exmig.bnd 명령을 사용하여 db2exmig 응용프로그램 패키지 이름을 판별할 수 있습니다. db2exmig.bnd 파일은 sqllib/bnd 폴더에 위치합니다.

## 제 195 장 db2expln - SQL 및 XQuery Explain

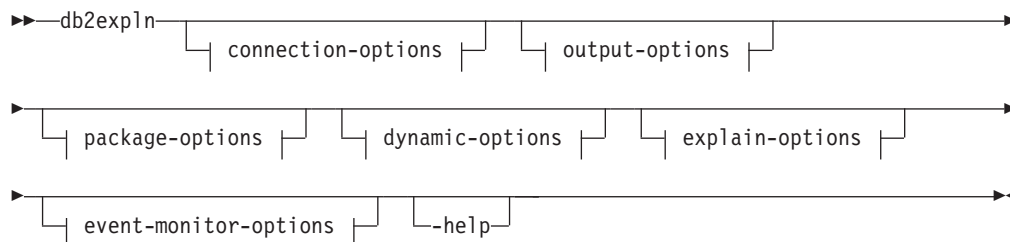
db2expln 도구는 SQL 및 XQuery 문에 대해 선택된 액세스 플랜을 설명합니다. Explain 데이터가 캡처되지 않은 경우에 선택된 액세스 플랜의 빠른 설명을 보기 위해 사용할 수 있습니다. 정적 SQL 및 XQuery 문의 경우, db2expln은 시스템 카탈로그 테이블에 저장된 패키지를 조사합니다. 동적 SQL 및 XQuery 문의 경우, db2expln은 쿼리 캐시의 섹션을 조사합니다.

### 권한 부여

dbadm이나 다음 권한 또는 특권 중 하나이어야 합니다.

- 정적 명령문의 경우, 카탈로그 테이블에 대한 SELECT 특권
- 동적 명령문의 경우, 카탈로그 테이블에 대한 SELECT 특권과 다음 권한 또는 특권 중 하나이어야 합니다.
  - 명령문을 컴파일하기에 충분한 특권
  - EXPLAIN 권한
  - SQLADM 권한

### 명령 구문



#### connection-options:

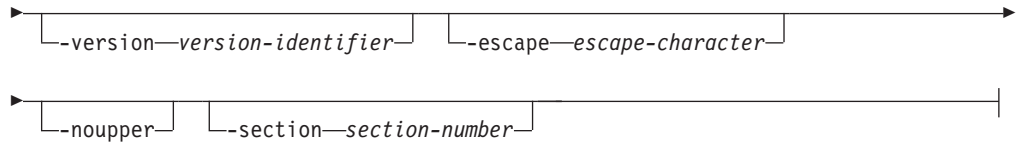
`--database database-name` `--user user-id password`

#### output-options:

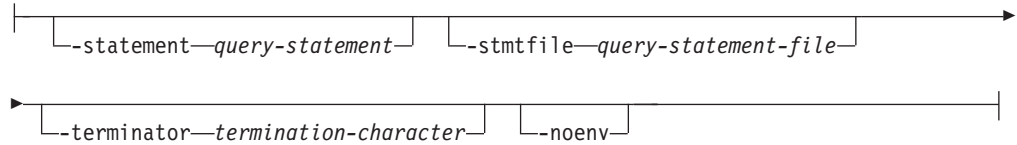
`-output output-file` `-terminal`

#### package-options:

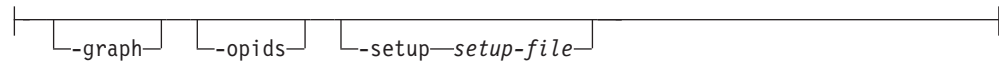
`-schema schema-name` `--package package-name`



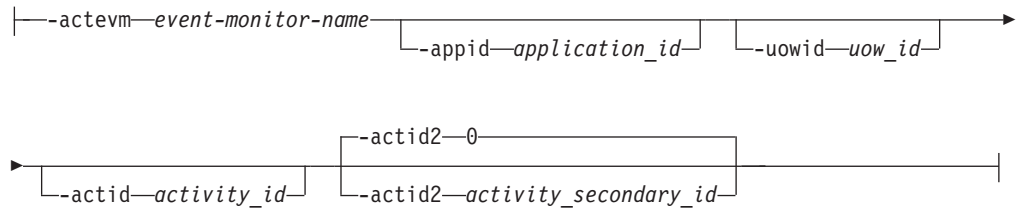
**dynamic-options:**



**explain-options:**



**event-monitor-options:**



**명령 매개변수**

옵션은 어떤 순서로도 지정할 수 있습니다.

**connection-options:**

이 옵션은 연결할 데이터베이스와 연결에 필요한 옵션을 지정합니다. 연결 옵션은 `-help` 옵션을 지정하는 경우를 제외하고 필수입니다.

**-database database-name**

Explain 대상 패키지를 포함하는 데이터베이스의 이름.

이전 버전과의 호환성을 위해, `-database` 대신 `-d`를 사용할 수 있습니다.

**-user user-id password**

데이터베이스 연결을 설정할 때 사용할 권한 부여 ID 및 암호. `user-id` 및 `password` 모두 DB2 이름 지정 규칙에 따라 유효해야 하고 데이터베이스에서 인식되어야 합니다.

이전 버전과의 호환성을 위해, `-user` 대신 `-u`를 사용할 수 있습니다.

### **output-options:**

이 옵션은 db2expln 출력을 보낼 위치를 지정합니다. -help 옵션을 지정하는 경우를 제외하고 하나 이상의 출력 옵션을 지정해야 합니다. 두 옵션 모두를 지정하면, 출력이 파일과 터미널로 전송됩니다.

#### **-output** *output-file*

db2expln 출력은 사용자가 지정하는 파일에 기록됩니다.

이전 버전과의 호환성을 위해, -output 대신 -o를 사용할 수 있습니다.

#### **-terminal**

db2expln 출력을 터미널로 보냅니다.

이전 버전과의 호환성을 위해, -terminal 대신 -t를 사용할 수 있습니다.

### **package-options:**

이 옵션은 Explain 대상이 되는 하나 이상의 패키지 및 섹션을 지정합니다. 패키지 및 섹션의 정적 쿼리만 Explain 대상이 됩니다.

LIKE 술어에서와 같이, 패턴 일치 문자(퍼센트 기호(%) 및 밑줄(\_))를 사용하여 *schema-name*, *package-name* 및 *version-identifier*를 지정합니다.

#### **-schema** *schema-name*

Explain 대상 패키지의 SQL 스키마.

이전 버전과의 호환성을 위해, -schema 대신 -s를 사용할 수 있습니다.

#### **-package** *package-name*

Explain 대상 패키지의 이름.

이전 버전과의 호환성을 위해, -package 대신 -p를 사용할 수 있습니다.

#### **-version** *version-identifier*

Explain 대상 패키지의 버전 ID. 디폴트 버전은 비어 있는 문자열입니다.

#### **-escape** *escape-character*

*schema-name*, *package-name* 및 *version-identifier*에서 패턴 일치의 Escape 문자로 사용할 문자(*escape-character*).

예를 들어, 패키지 TESTID.CALC%를 Explain할 db2expln 명령은 다음과 같습니다.

```
db2expln -schema TESTID -package CALC% ....
```

그러나 이 명령은 CALC로 시작하는 다른 플랜도 Explain합니다. TESTID.CALC% 패키지만 Explain하려면 Escape 문자를 사용해야 합니다. 느낌표(!)를 Escape 문자로 지정하는 경우, db2expln -schema TESTID -escape ! -package CALC!% ... 로 명령을 변경할 수 있습니다. 그러면 ! 문자가

Escape 문자로 사용되므로 !%는 % 문자로 해석되고 "모든 것 일치" 패턴으로 해석되지 않습니다. 디폴트 Escape 문자는 없습니다.

이전 버전과의 호환성을 위해, -escape 대신 -e를 사용할 수 있습니다.

문제점을 예방하려면 db2expln Escape 문자로 운영 체제 Escape 문자를 지정하지 마십시오.

#### **-noupper**

*schema-name*, *package-name* 및 *version-identifier*가 일치하는 패키지를 검색하기 전에 대문자로 변환되지 않아야 함을 지정합니다.

디폴트로, 이러한 변수는 패키지 검색 이전에 대문자로 변환됩니다. 이 옵션은 값이 입력된 그대로 사용되어야 함을 표시합니다.

이전 버전과의 호환성을 위해, -noupper 대신 -l(숫자 1이 아니고 L의 소문자 임)을 사용할 수 있습니다.

#### **-section *section-number***

선택된 패키지 내에서 Explain할 섹션 번호.

각 패키지에서 모든 섹션을 Explain하려면 숫자 영(0)을 사용하십시오. 이 설정이 디폴트 동작입니다. 이 옵션을 지정하지 않거나 *schema-name*, *package-name* 또는 *version-identifier*에 패턴 일치 문자가 포함되는 경우, 모든 섹션이 표시됩니다.

섹션 번호를 찾으려면 시스템 카탈로그 뷰 SYSCAT.STATEMENTS를 쿼리하십시오. 시스템 카탈로그 뷰의 설명은 SQL 참조서를 참조하십시오.

이전 버전과의 호환성을 위해, -section 대신 -s를 사용할 수 있습니다.

#### **dynamic-options:**

이 옵션은 Explain 대상이 되는 하나 이상의 동적 쿼리 명령문을 지정합니다.

#### **-statement *query-statement***

동적으로 준비하고 Explain할 SQL 또는 XQuery 쿼리 명령문. 두 개 이상의 명령문을 Explain하려면, -stmtfile 옵션을 사용하여 Explain할 쿼리 명령문을 포함하는 파일을 제공하거나, -terminator 옵션을 사용하여 -statement 옵션에서 명령문을 구분하기 위해 사용할 수 있는 종료 문자를 정의하십시오.

#### **-stmtfile *query-statement-file***

동적으로 준비하고 Explain할 하나 이상의 쿼리 명령문을 포함하는 파일. 디폴트로, 파일의 각 라인은 구별 쿼리 명령문으로 가정합니다. 명령문이 여러 라인에 걸쳐 있어야 하는 경우, -terminator 옵션을 사용하여 쿼리 명령문의 끝을 표시하는 문자를 지정하십시오.

#### **-terminator *termination-character***

동적 쿼리 명령문의 끝을 표시하는 문자. 디폴트로, -statement 옵션은 단일 쿼리



리 명령문을 제공하고 -stmtfile에 있는 파일의 각 라인은 별도의 쿼리 명령문으로 처리됩니다. 사용자가 지정하는 종료 문자를 사용하여 -statement로 여러 쿼리 명령문을 제공하거나 -stmtfile 파일에서 명령문이 여러 라인에 걸쳐 있도록 할 수 있습니다.

#### **-noenv**

컴파일 환경을 변경하는 동적 명령문이 Explain 후에 실행되지 않아야 함을 지정합니다.

디폴트로, db2expln은 Explain 후에 다음 명령문 중에서 하나를 실행합니다.

```
SET CURRENT DEFAULT TRANSFORM GROUP
SET CURRENT DEGREE
SET CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION
SET CURRENT QUERY OPTIMIZATION
SET CURRENT REFRESH AGE
SET PATH
SET SCHEMA
```

이 명령문은 db2expln이 처리하는 연속 동적 쿼리 명령문에 대해 선택된 플랜을 변경할 수 있도록 합니다.

-noenv를 지정하는 경우, 이 명령문은 Explain되지만 실행되지 않습니다.

동적 쿼리를 Explain하려면 -statement 또는 -stmtfile을 지정해야 합니다. 두 옵션 모두 단일 db2expln 호출에서 지정할 수 있습니다.

#### **explain-options:**

이 옵션은 Explain 대상 플랜에 제공되는 추가 정보를 판별합니다.

#### **-graph**

옵티마이저 플랜 그래프를 표시합니다. 각 섹션이 조사되고, 원래의 옵티마이저 플랜 그래프는 Visual Explain에서 표시되는 대로 생성됩니다.

생성된 그래프는 Visual Explain 그래프와 정확히 일치하지 않을 수 있습니다. 옵티마이저 그래프는 섹션 플랜에 포함된 정보를 기초로 일부 값을 표시할 수 있습니다.

이전 버전과의 호환성을 위해, -graph 대신 -g를 지정할 수 있습니다.

**-opids** Explain 대상 플랜의 운영자 ID 번호를 표시합니다.

운영자 ID 번호를 사용하여 db2expln의 출력을 Explain 기능의 출력과 일치시킬 수 있습니다. 모든 운영자에게 ID 번호가 있는 것은 아니며 Explain 기능 출력에 표시되는 일부 ID 번호는 db2expln 출력에 표시되지 않습니다.

이전 버전과의 호환성을 위해, -opids 대신 -i를 지정할 수 있습니다.

**-help** db2expln에 대한 도움말 텍스트를 표시합니다. 이 옵션을 지정하는 경우 패키지는 Explain되지 않습니다.

대부분의 명령행은 db2exsrv 스토어드 프로시저에서 처리됩니다. 사용 가능한 모든 옵션에 대한 도움말을 가져오려면 -help와 함께 **connection-options**를 제공해야 합니다. 예를 들어 다음을 사용하십시오.

```
db2expln -help -database SAMPLE
```

이전 버전과의 호환성을 위해, -? 대신 -h를 지정할 수 있습니다.

**-setup** *setup-file*

재컴파일에 필요한 동적 명령문이나 정적 명령문(예: 선언된 임시 테이블을 참조하는 정적 명령문)에 대한 환경을 설정하는 데 필요한 하나 이상의 명령문을 포함하는 파일. 파일의 각 명령문이 실행되고 오류 또는 경고가 보고됩니다. 파일의 명령문은 Explain되지 않습니다.

**event-monitor-options:**

이 옵션은 Explain 대상이 되는 활동 이벤트 모니터의 하나 이상의 섹션 환경을 지정합니다.

**-actevm** *event-monitor-name*

activitystmt 논리 그룹에 Explain 대상이 되는 섹션 환경(**section\_env** 모니터 요소에 있는)이 있는 활동 이벤트 모니터의 이름을 지정합니다.

**-appid** *application-id*

Explain 대상이 되는 섹션 환경의 활동을 발행한 응용프로그램을 고유하게 식별하는 응용프로그램 ID(**appl\_id** 모니터 요소)를 지정합니다. -appid가 지정되면 -actevm이 지정되어야 합니다.

**-uowid** *uow-id*

섹션 환경이 Explain 대상인 작업 단위(UOW) ID(**uow\_id** 모니터 요소)를 지정합니다. 작업 단위(UOW)는 주어진 응용프로그램 내에서만 고유합니다. -uowid가 지정되면 -actevm이 지정되어야 합니다.

**-actid** *activity-id*

섹션 환경이 Explain 대상인 활동 ID(**activity\_id** 모니터 요소)를 지정합니다. 활동 ID는 주어진 작업 단위(UOW) 내에서만 고유합니다. -actid가 지정되면 -actevm이 지정되어야 합니다.

**-actid2** *activity-secondary-id*

섹션 환경이 Explain 대상인 활동 보조 ID(**activity\_secondary\_id** 모니터 요소)를 지정합니다. 지정하지 않을 경우 디폴트는 0입니다. -actid2가 지정되면 -actevm이 지정되어야 합니다.

## 사용 시 참고사항

-help 옵션을 지정하지 않으면, package-options 또는 dynamic-options를 지정해야 합니다. 단일 db2expln 호출에서 패키지와 동적 SQL 모두 Explain할 수 있습니다.

위의 옵션 플래그 중 일부는 사용자 운영 체제에 특수한 의미를 가지고 있을 수도 있으므로, db2expln 명령행에서 올바르게 해석되지 않을 수도 있습니다. 그러나 운영 체제 Escape 문자를 앞에 붙여서 이러한 문자를 입력할 수 있습니다. 자세한 정보는 운영 체제 문서를 참조하십시오. db2expln Escape 문자로 운영 체제 Escape 문자를 지정하지 않도록 하십시오.

db2expln에서 생성되는 도움말 및 상태 메시지는 표준 출력에 기록됩니다. Explain 도구에서 생성되는 모든 프롬프트 및 기타 상태 메시지는 표준 오류에 기록됩니다. Explain 텍스트는 표준 출력에 기록되거나 선택된 출력에 따라 파일에 기록됩니다.

db2expln에서 다음 메시지가 리턴될 수 있습니다.

- No packages found for database package pattern: "<creator>".<package> with version "<version>"

이 메시지는 데이터베이스에서 지정된 패턴과 일치하는 패키지가 발견되지 않은 경우에 출력에 표시됩니다.

- Bind messages can be found in db2expln.msg

이 메시지는 db2expln.bnd 바인드에 실패한 경우 출력에 표시됩니다. 직면한 문제점에 대한 추가 정보는 현재 디렉토리의 db2expln.msg 파일에서 찾을 수 있습니다.

- Section number overridden to 0 (all sections) for potential multiple packages.

이 메시지는 db2expln에서 여러 개의 패키지가 발견될 수 있는 경우에 출력에 표시됩니다. 이 조치는 패턴 일치 문자 중 하나가 패키지 또는 작성자 입력 인수에 사용되는 경우에 취합니다.

- Bind messages for <bind file> can be found in <message file>

이 메시지는 지정된 바인드 파일의 바인드에 실패한 경우 표시됩니다. 직면한 문제점에 대한 추가 정보는 데이터베이스 서버에서 지정된 메시지 파일에서 찾을 수 있습니다.

- No static sections qualify from package.

이 메시지는 지정된 패키지만 동적 쿼리 명령문을 포함하는 경우(즉, 정적 섹션이 없는 경우) 출력에 표시됩니다.

- Package "<creator>".<package>", "<version>", is not valid. Rebind the package and then rerun db2expln.

이 메시지는 지정된 패키지가 현재 유효하지 않은 경우 출력에 표시됩니다. 플랜에 대해 BIND 또는 REBIND 명령을 다시 발행하여 데이터베이스에서 유효한 패키지를 다시 작성한 후 db2expln을 다시 실행하십시오.

다음 명령문은 Explain 대상이 되지 않습니다.

- BEGIN/END COMPOUND
- BEGIN/END DECLARE SECTION
- CLOSE cursor
- COMMIT 및 ROLLBACK
- CONNECT
- DESCRIBE
- Dynamic DECLARE CURSOR
- EXECUTE
- EXECUTE IMMEDIATE
- FETCH
- INCLUDE
- OPEN cursor
- PREPARE
- SQL 제어 명령문
- WHENEVER

복합 SQL문 내의 각 부속 명령문에는 자체의 고유 섹션이 있을 수 있습니다. 이 섹션은 db2expln에 의해 Explain할 수 있습니다.

주: db2expln 명령은 XQuery문을 제외하지 않습니다.

## 예:

한 번의 db2expln 호출로 여러 플랜을 Explain하려면, -package, -schema 및 -version 옵션을 사용하고 LIKE 패턴의 패키지 및 작성자에 대한 문자열 상수를 지정하십시오. 즉, 밑줄(\_)을 사용하여 단일 문자를 나타내고, 퍼센트 기호(%)를 사용하여 0개 이상의 문자 발생을 나타냅니다.

SAMPLE 데이터베이스에 있는 모든 패키지의 모든 섹션을 Explain하고 결과는 my.exp 파일에 기록되도록 하려면 다음 명령을 사용하십시오.

```
db2expln -database SAMPLE -schema % -package % -output my.exp
```

다른 예로, 사용자가 "statements.db2"라고 하는 CLP 스크립트 파일을 가지고 있고 파일의 명령문을 Explain하려고 합니다. 파일은 다음 명령문을 포함하고 있습니다.

```
SET PATH=SYSIBM, SYSPFUN, DEPT01, DEPT93@  
SELECT EMPNO, TITLE(JOBID) FROM EMPLOYEE@
```

이 명령문을 Explain하려면 다음 명령을 입력하십시오.

```
db2expln -database DEPTDATA -stmtfile statements.db2 -terminator @ -terminal
```

다음 명령문을 Explain합니다.

```
SELECT e.lastname, e.job, d.deptname, d.location, p.projname
FROM employee AS e, department AS d, project AS p
WHERE e.workdept = d.deptno AND e.workdept = p.deptno
```

다음 명령은

```
db2expln -database SAMPLE
-statement "SELECT e.lastname, e.job,
d.deptname, d.location, p.projname
FROM employee AS e, department AS d, project AS p
WHERE e.workdept = d.deptno AND e.workdept = p.deptno"
-terminal
```

다음은 리턴합니다.

```
DB2 Enterprise Server Edition n.n, nnnn-xxx (c) Copyright IBM Corp. 1991, yyyy
Licensed Material - Program Property of IBM
IBM DB2 Database SQL and XQUERY Explain Tool
```

```
***** DYNAMIC *****
```

```
===== STATEMENT =====
```

```
Isolation Level          = Cursor Stability
Blocking                  = Block Unambiguous Cursors
Query Optimization Class = 5

Partition Parallel       = No
Intra-Partition Parallel = No

SQL Path                  = "SYSIBM", "SYSFUN", "SYSPROC", "SYSIBMADM",
                          "SDINIRO"
```

Statement:

```
SELECT e.lastname, e.job, d.deptname, d.location, p.projname
FROM employee AS e, department AS d, project AS p
WHERE e.workdept =d.deptno AND e.workdept =p.deptno
```

Section Code Page = 1208

Estimated Cost = 22.802252

Estimated Cardinality = 105.000000

Access Table Name = SDINIRO.PROJECT ID = 2,10

```
| #Columns = 2
| Skip Inserted Rows
| Avoid Locking Committed Data
| Currently Committed for Cursor Stability
| Relation Scan
| | Prefetch: Eligible
| Lock Intents
| | Table: Intent Share
| | Row : Next Key Share
| Sargable Predicate(s)
| | Process Build Table for Hash Join
```

Hash Join

```
| Estimated Build Size: 4000
| Estimated Probe Size: 4000
| Access Table Name = SDINIRO.DEPARTMENT ID = 2,6
| | #Columns = 3
| | Skip Inserted Rows
```

```

| | Avoid Locking Committed Data
| | Currently Committed for Cursor Stability
| | Relation Scan
| | | Prefetch: Eligible
| | Lock Intents
| | | Table: Intent Share
| | | Row : Next Key Share
| | Sargable Predicate(s)
| | | Process Probe Table for Hash Join
Hash Join
| Estimated Build Size: 4000
| Estimated Probe Size: 4000
| Access Table Name = SDINIRO.EMPLOYEE ID = 2,7
| | #Columns = 3
| | Skip Inserted Rows
| | Avoid Locking Committed Data
| | Currently Committed for Cursor Stability
| | Relation Scan
| | | Prefetch: Eligible
| | Lock Intents
| | | Table: Intent Share
| | | Row : Next Key Share
| | Sargable Predicate(s)
| | | Process Probe Table for Hash Join
Return Data to Application
| | #Columns = 5

End of section

```

## 제 196 장 db2extsec - DB2 오브젝트에 대한 권한 설정

갱신된 DB2 데이터베이스 시스템 설치에서 DB2 오브젝트(예를 들어, 파일, 디렉토리, 네트워크 공유, 레지스트리 키 및 서비스)에 대한 권한을 설정합니다.

DB2 버전 9 FixPack 2부터는 확장 보안을 위해 도메인 그룹을 사용할 수 있습니다.

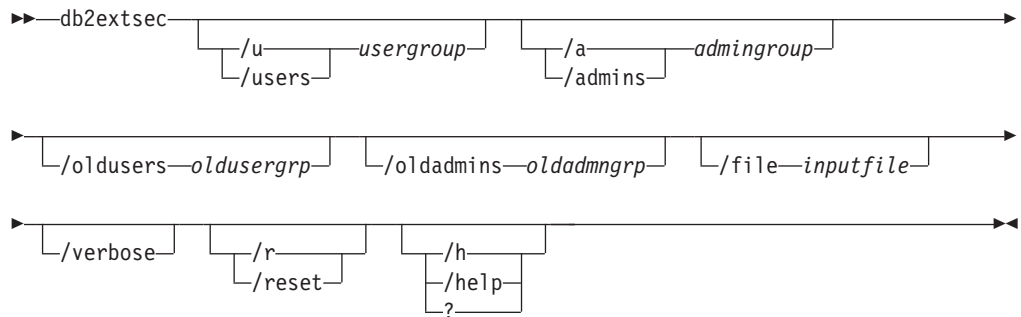
### 권한 부여

*sysadm*

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

#### `/u` | `/users` *usergroup*

추가될 사용자 그룹의 이름을 지정합니다. 이 옵션을 지정하지 않으면 디폴트 DB2 사용자 그룹(DB2USERS)이 사용됩니다. *usergroup*은 로컬 그룹 또는 도메인 그룹입니다. 로컬 그룹을 지정하려면 머신 이름을 사용하거나 사용하지 않고 그룹 이름을 지정할 수 있습니다(예를 들어, DB2USERS 또는 MYWKSTN\DB2USERS). 도메인 그룹을 지정하려면 *usergroup*을 DOMAIN\GROUP 양식으로 지정합니다 (예를 들어, MYDOMAIN\DB2USERS).

#### `/a` | `/admins` *admingroup*

추가될 관리 그룹의 이름을 지정합니다. 이 옵션을 지정하지 않으면 디폴트 DB2 관리 그룹(DB2ADMNS)이 사용됩니다. *admingroup*은 로컬 그룹 또는 도메인 그룹입니다. 로컬 그룹을 지정하려면 머신 이름을 사용하거나 사용하지 않고 그룹 이름을 지정할 수 있습니다(예를 들어, DB2ADMNS 또는

MYWKSTN\DB2ADMNS). 도메인 그룹을 지정하려면 *admin*group을 DOMAIN\GROUP 양식으로 지정합니다(예를 들어, MYDOMAIN\DB2ADMNS).

**메모:** 확장 보안 그룹 이름을 사용하거나 디폴트 위치(예를 들어, 설치 디렉토리 또는 데이터베이스 디렉토리) 외부에서 작성된 파일 또는 디렉토리 오브젝트가 있는 경우 /oldusers, /oldadmins 및 /file 매개변수가 필요합니다. db2extsec 명령은 알려진 DB2 파일에 대한 권한만 변경할 수 있습니다. 사용자가 확장 보안으로 개인용 DB2 파일을 작성한 경우 사용자는 이러한 파일의 위치를 제공해야 하므로 db2extsec 명령이 새 확장 보안 그룹 이름으로 이러한 파일에 대한 권한을 변경할 수 있습니다. 파일 위치는 /file 옵션을 사용하여 *inputfile*에 제공됩니다.

**/oldusers** *oldusergrp*

변경될 이전 DB2 사용자 그룹 이름

**/oldadmins** *oldadmngrp*

변경될 이전 DB2 관리자 그룹 이름

**/file** *inputfile*

권한을 갱신해야 하는 추가 파일/디렉토리를 나열하는 파일.

**/verbose**

추가 정보를 출력합니다.

**/r | /reset**

이전에 db2extsec를 실행하여 작성된 변경사항을 변환해야 하는 것으로 지정합니다. 이 옵션을 지정하면 다음 옵션은 모두 무시됩니다. 이 옵션은 다른 db2extsec 명령이 실행된 후 다른 DB2 명령이 실행되지 않은 경우에만 실행됩니다.

**/h | /help | ?**

명령 도움말 정보를 표시합니다.

**예:**

확장 보안을 사용하고 mydom\db2users 및 mydom\db2admns 도메인 그룹을 사용하여 DB2 오브젝트를 보호하려면 다음을 실행하십시오.

```
db2extsec /u mydom\db2users /a mydom\db2admns
```

확장 보안을 이전 설정으로 재설정하려면 다음을 실행하십시오(위의 /reset 옵션 참조).

```
db2extsec /reset
```

위와 같이 확장 보안을 사용할 뿐만 아니라 db2admns 및 db2users 로컬 그룹의 c:\mylist.lst에 나열된 파일/디렉토리에 대한 보안 그룹을 mydom\db2admns 및 mydom\db2users 도메인 그룹으로 변경하려면 다음을 실행하십시오.



```
db2extsec /users mydom#db2users /admins mydom#db2admns /oldadmins db2admns  
/oldusers db2users /file c:#mylist.lst
```

주: 입력 파일 형식은 다음과 같습니다.

```
* This is a comment
```

```
D:#MYBACKUPDIR
```

```
D:#MYEXPORTDIR
```

```
D:#MYMISCFILE#myfile.dat
```

```
* This is another comment
```

```
E:#MYOTHERBACKUPDIR
```

```
E:#MYOTHEREXPORTDIR
```

```
* These are more comments
```



---

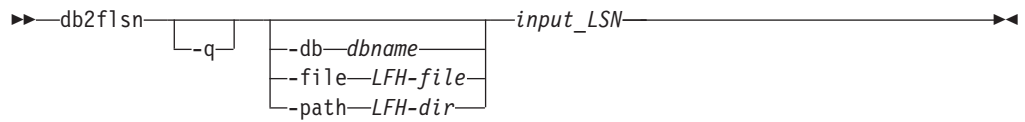
## 제 197 장 db2flsn - 로그 시퀀스 번호 찾기

지정된 로그 시퀀스 번호(LSN)로 식별되는 로그 레코드가 들어있는 파일의 이름을 리턴합니다.

### 권한 부여

없음

### 명령 구문



### 명령 매개변수

**-q** 로그 파일 이름만 인쇄되도록 지정합니다. 오류 또는 경고 메시지는 인쇄되지 않으며, 상태는 리턴 코드를 통해서만 판별할 수 있습니다. 유효한 오류 코드는 다음과 같습니다.

- -100 유효하지 않은 입력
- -101 LFH 파일을 열 수 없음
- -102 LFH 파일을 읽지 못했음
- -103 유효하지 않은 LFH
- -104 데이터베이스 복구 불가능
- -105 LSN이 너무 큼
- -106 유효하지 않은 데이터베이스
- -500 논리적 오류

기타 유효한 리턴 코드는 다음과 같습니다.

- 0 실행 성공
- 99 경고: 결과가 마지막 알려진 로그 파일 크기에 기초합니다.

#### **-db dbname**

조사하려는 데이터베이스 이름을 지정합니다.

#### **-file LFH-file**

파일 이름을 포함한 LFH 파일의 전체 경로를 지정합니다.

### **-path** LFH-dir

LFH 파일 SQLOGCTL.LFH.1 및 해당 미러 사본 SQLOGCTL.LFH.2가 있는 디렉토리의 전체 경로를 지정합니다.

### *input\_LSN*

선행 영(0)을 갖는 내부(6 또는 8바이트) 16진 값을 표시하는 12 또는 16문자 열.

### 예:

```
db2flsn 000000BF0030
  Given LSN is contained in log page 2 in log file S0000002.LOG

db2flsn -q 000000BF0030
  S0000002.LOG

db2flsn 000000BE0030
  Given LSN is contained in log page 2 in log file S0000001.LOG

db2flsn -q 000000BE0030
  S0000001.LOG

db2flsn -db flsntest 0000000000FA0000
  Warning: the result is based on the last known log file size (6
  4K pages starting from log extent 10). The input_LSN might be before
  the database becomes recoverable.

  Given LSN is contained in log page 2 in log file S0000002.LOG

db2flsn -q -db flsntest 0000000000FA0000
  S0000002.LOG

db2flsn -file C:\DB2\NODE0000\SQL00001\SQLOGCTL.LFH.1 0000000000FA4368
  Given LSN is contained in log page 6 in log file S0000002.LOG

db2flsn -path C:\DB2\NODE0000\SQL00001 0000000000FA4368
  Given LSN is contained in log page 6 in log file S0000002.LOG
```

### 사용 시 참고사항

- -db, -file 및 -path가 지정되지 않으면 도구는 LFH 파일 SQLOGCTL.LFH.1 및 미러 사본 SQLOGCTL.LFH.2가 현재 디렉토리에 있다고 가정합니다.
- -file이 지정되면 제공된 LFH 파일만 사용됩니다.
- -file이 지정되지 않으면 두 LFH 파일 SQLOGCTL.LFH.1 및 미러 사본 SQLOGCTL.LFH.2에서 사용 가능한 최근 로그 제어 레코드가 사용됩니다. 두 파일 중 하나가 누락 또는 손상된 경우 다른 파일이 사용됩니다. 두 파일이 모두 누락 또는 손상된 경우 db2flsn은 실패합니다.
- 도구는 **logfilsiz** 데이터베이스 구성 매개변수를 사용합니다. DB2는 이 매개변수에 대한 세 개의 최근 값을 기록하며, 첫 번째 로그 파일이 각 **logfilsiz** 값을 사용하여 작성됩니다. 이것은 도구가 **logfilsiz**가 변할 때 올바르게 작업할 수 있게 합니다. 지정된 LSN이 **logfilsiz**의 가장 빨리 기록된 값을 앞서는 경우 도구는 이 값을 사용하고 경고를 리턴합니다. DB2 Universal Database 버전 5.2 이전의 데이터베이스 관

리 프로그램에서 도구를 사용할 수 있습니다. 이 경우 올바른 결과(**logfilsiz** 값이 변경되지 않는 경우에 얻는)에서도 경고가 리턴됩니다.

- 이 도구는 복구 가능한 데이터베이스에서만 사용할 수 있습니다. 데이터베이스는 **logarchmeth1** 또는 **logarchmeth2** 구성 매개변수가 OFF 이외의 값으로 설정되어 구성되는 경우 복구 가능합니다.



## 제 198 장 db2fm - DB2 결합 모니터

DB2 결합 모니터 디먼을 제어합니다. db2fm을 사용하여 결합 모니터를 구성할 수 있습니다.

이 명령은 UNIX 운영 체제에서만 사용할 수 있습니다.

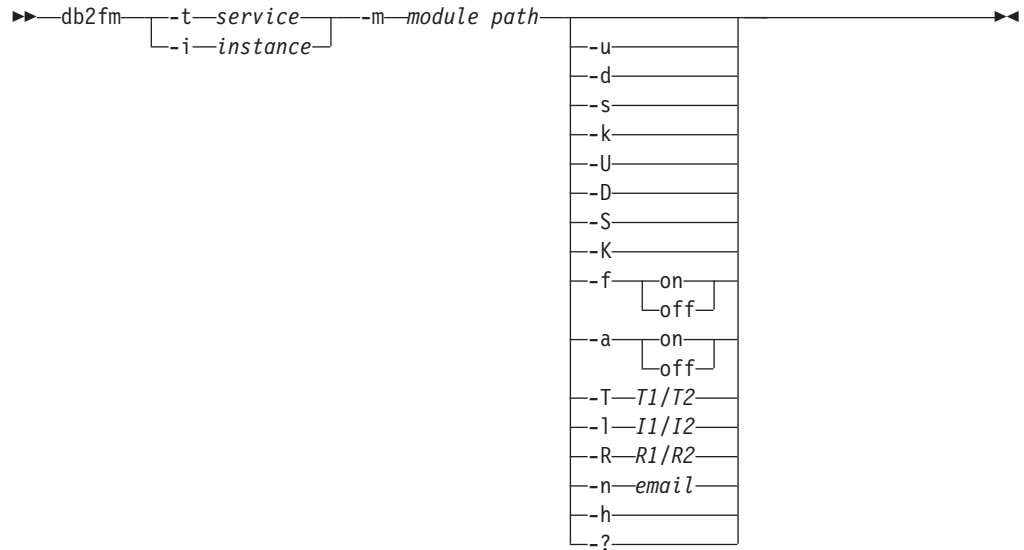
### 권한 부여

명령을 실행하는 인스턴스에 대한 권한 부여.

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

**-m** *module-path*

모니터할 제품의 결합 모니터 공유 라이브러리 전체 경로를 정의합니다. 디폴트 값은 \$INSTANCEHOME/sql1lib/lib/libdb2gcf입니다.

**-t** *service*

서비스의 고유 텍스트 디스크립터를 제공합니다.

**-i** *instance*

서비스 인스턴스를 정의합니다.

- u 서비스를 시작합니다.
- U 결합 모니터 디면을 시작합니다.
- d 인스턴스를 종료합니다.
- D 결합 모니터 디면을 종료합니다.
- k 서비스를 제거합니다.
- K 결합 모니터 디면을 제거합니다.
- s 서비스 상태를 리턴합니다.
- S 결합 모니터 디면의 상태를 리턴합니다. 서비스 또는 결합 모니터의 상태는 다음 중 하나입니다.
  - 올바르게 설치되지 않음,
  - 올바르게 설치되었지만 활성화되지 않음,
  - 활성화되었지만 사용 불가능(유지보수),
  - 사용 가능 또는
  - 알 수 없음

**-f on | off**

결합 모니터를 설정 또는 해제합니다. 이 옵션이 해제로 설정되면 결합 모니터 디면이 시작되지 않거나 실행 중인 디면이 종료됩니다.

**-a on | off**

결합 모니터링을 설정 또는 해제합니다. 이 옵션이 해제로 설정되면 결합 모니터가 적극적으로 모니터링하지 않습니다. 즉, 서비스가 중단되어도 재시작을 시도하지 않습니다.

**-T T1/T2**

시작 및 중지 시간종료를 겹쳐줍니다.

예를 들면, 다음과 같습니다.

- -T 15/10은 두 개 시간종료를 각각 갱신합니다.
- -T 15는 시작 시간종료를 15초로 갱신합니다.
- -T /10은 중지 시간종료를 10초로 갱신합니다.

**-I I1/I2**

상태 간격과 시간종료를 각각 설정합니다.

**-R R1/R2**

중단 전 상태 방법 및 조치에 대한 재시도 수를 설정합니다.

**-n email**

이벤트 통지를 위한 전자 우편 주소를 설정합니다.



**-h | -?**

명령 사용법 도움말을 표시합니다.



---

## 제 199 장 db2fmcu - DB2 결합 모니터 제어기 명령

DB2 결합 모니터는 손상 후 자동으로 인스턴스를 시작하는 DB2 기능입니다. 이 기능은 또한 머신이 재부트되면 인스턴스를 자동으로 재시작할 수 있습니다. DB2 결합 모니터는 Linux 및 UNIX 시스템에서 DB2 결합 모니터 제어기 명령을 사용하여 구성할 수 있습니다. 이 명령은 시스템의 inittab 파일에 액세스하므로 root로서 실행되어야 합니다.

### 권한 부여

Root

### 필수 연결

없음

### 명령 구문

```
▶▶ db2fmcu -u -p db2fmc_path -d -f inittab ▶▶
```

### 명령 매개변수

#### **-u -p** *db2fmc\_path*

이 옵션은 시스템 시작 시 결합 모니터 제어기(FMC)를 포함하도록 inittab 파일을 재구성합니다. 여기서 *db2fmc\_path*는 FMC 디먼(db2fmc) 오브젝트의 전체 경로(예를 들어, /opt/IBM/db2/bin/db2fmc)입니다.

**-d** 이 옵션은 시스템 시작 시 FMC가 시작되지 않도록 inittab 파일 구성을 변경합니다.

#### **-f** *inittab*

이 옵션은 inittab 파일 경로를 지정합니다.

### 예:

시스템 시작 시 inittab 파일을 재구성하여 결합 모니터 제어기를 시작하려면 다음 명령을 실행하십시오.

```
db2fmcu -u -p /opt/IBM/db2/bin/db2fmc
```

시스템 시작 시 결합 모니터 제어기가 시작되지 않도록 하려면 다음 명령을 실행하십시오.

```
db2fmcu -d
```

## 사용 시 참고사항

/etc/inittab을 수동으로 변경한 경우 SIGHUP을 프로세스 1로 보내 /etc/inittab을 즉시 다시 스캔하도록 요청해야 합니다. 그렇지 않은 경우 다음에 다시 스캔하기 전에 시간이 소요될 수 있습니다. db2fmcu를 통해 /etc/inittab를 갱신하는 경우 db2fmcu가 이미 신호를 보냈으므로 다시 신호를 보내지 않아도 됩니다.

---

## 제 200 장 db2fodc - DB2 FODC(First Occurrence Data Collection) 명령

db2fodc 유틸리티는 문제점 판별 상황에서 도움을 주기 위해 DB2 인스턴스에 대한 증상 기반 데이터를 캡처합니다. 가능한 정지사항, 심각한 성능 문제점 및 다양한 오류 유형에 대한 정보를 수집하기 위한 것입니다.

### 목적

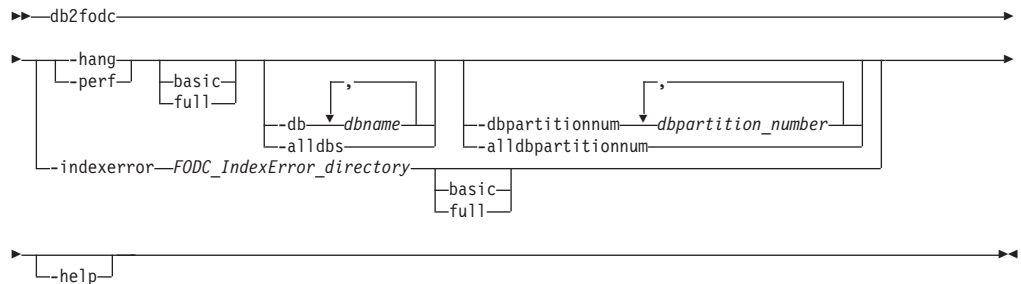
이 도구는 기능을 교체하고 다른 PD 도구의 기능을 통합합니다. db2fodc 명령은 자동 FODC를 트리거할 수 없는 문제점(예: 정지사항 또는 심각한 성능 문제점)에 대한 수동 FODC(First Occurrence Data Collection)에 사용할 수 있습니다. 인덱스 오류에 대한 데이터를 수집하는 경우에도 사용할 수 있습니다. db2fodc 도구는 FODC 패키지에 포함될 데이터를 캡처하고, 현재 DIAGPATH에서 작성된 FODC\_symptom 디렉토리 내에 이 데이터를 위치시킵니다. 여기서 symptom은 문제점 증상입니다.

### 권한 부여

다음 중 하나가 필요합니다.

- Linux 및 UNIX 시스템의 경우 *sysadm* 권한 레벨. 또한 인스턴스 소유자여야 합니다.
- Windows 운영 체제의 경우 *sysadm* 권한 레벨.

### 명령 구문



### 명령 매개변수

#### -hang

가능한 정지 상황이나 심각한 성능 문제점에 관련된 FODC 데이터를 수집합니다. FODC 패키지는 이 옵션을 실행한 결과로 준비됩니다. 이 매개변수를 사용하는 경우, 인스턴스는 사용할 수 없는 것으로 간주되므로 재시작해야 합니다. 데이터 콜렉션은 가능한 많은 정보를 수집할 수 있도록 빠르게 수행됩니다.

## 기본

기본 수집 모드가 사용자 상호 작용 없이 실행됩니다.

## full

전체 수집 모드가 사용자 상호 작용 없이 실행됩니다. 이 옵션을 사용하려면 기본 수집 모드에서 실행하는 것보다 상당히 많은 자원과 시간이 필요합니다.

## -perf

성능 문제점에 관련된 데이터를 수집합니다. 이 옵션은 인스턴스가 계속 사용 가능하고 재시작하지 않아도 되는 경우에 사용해야 합니다. 이 옵션은 -hang 매개변수보다 적게 시스템에 영향을 줍니다. FODC 패키지는 이 옵션을 실행한 결과로 준비됩니다.

## 기본

기본 수집 모드가 사용자 상호 작용 없이 실행됩니다.

## full

전체 수집 모드가 사용자 상호 작용 없이 실행됩니다. 이 옵션에는 db2trc 명령과 추가 스냅샷이 포함됩니다. 이 옵션을 사용하려면 기본 수집 모드에서 실행하는 것보다 상당히 많은 자원이 필요합니다.

## -hang 및 -perf의 서브옵션

### -alldbs

모든 활성 데이터베이스에 관련된 FODC 데이터를 수집합니다. 이 옵션은 디폴트로 활성화됩니다.

### -db *dbname*

지정된 데이터베이스에 관련된 FODC 데이터를 수집합니다. 예를 들어, 다음과 같습니다.

```
db2fodc -hang -db sample,dbsample
```

### -alldbpartitionnum

명령이 인스턴스에서 활성 데이터베이스 파티션 서버 모두에 대해 실행됨을 지정합니다. db2fodc는 db2fodc가 실행되는 동일한 실제 머신의 데이터베이스 파티션 서버에서 제공되는 정보를 보고합니다.

### -dbpartitionnum *dbpartition\_number*

지정된 모든 데이터베이스 파티션 번호에 관련된 FODC 데이터를 수집합니다.

## -indexerror *FODC\_IndexError\_directory*

인덱스 오류에 관련된 데이터를 수집합니다. *FODC\_IndexError\_directory*가 필요하면 db2cos\_indexerror\_short(.bat) 스크립트 및/또는 db2cos\_indexerror\_long(.bat) 스크립트를 포함해야 합니다.

## 기본

기본 수집 모드가 실행됩니다. 동일한 *FODC\_IndexError\_directory*에 db2dart 보고서가 존재하지 않도록 확인하십시오.

## full

전체 수집 모드가 실행됩니다. 동일한 *FODC\_IndexError\_directory*에 db2dart 보고서가 존재하지 않도록 확인하십시오. db2cos\_indexerror\_long(.bat) 스크립트에 db2dart /t 명령이 포함된 경우, 전체 모드를 사용하려면 db2dart /t 명령이 실행될 데이터베이스가 오프라인이어야 합니다.

## -help

사용법 및 도움말 정보를 표시합니다. 이 옵션이 지정되면 다른 모든 옵션은 무시되고 도움말 정보만 표시됩니다.

## 예:

정지 가능성이 있는 동안 데이터베이스 관리 프로그램을 중지하지 않고 데이터를 수집하려면 다음을 입력하십시오.

```
db2fodc -hang -alldb
```

디폴트 DB2FODC 레지스트리 변수 및 매개변수가 사용됩니다. 새 디렉토리 *FODC\_hang\_<timestamp>*가 현재 진단 경로 아래에 작성됩니다(이미 존재하면 오류가 생성됨). db2cos\_hang 스크립트를 사용하여 수동 FODC 데이터를 하나 이상의 파일로 수집합니다. 이 파일은 *FODC\_hang\_<timestamp>* 디렉토리에 놓입니다.

특정 데이터베이스에서 데이터를 수집하려면 다음을 입력하십시오.

```
db2fodc -db SAMPLE -hang
```

데이터 컬렉션은 데이터베이스 SAMPLE로 제한됩니다. 새 디렉토리 *FODC\_hang\_<timestamp>*가 자동으로 현재 진단 경로 아래에 작성됩니다. 시간소인은 db2fodc가 호출된 시간입니다. db2cos\_hang 스크립트를 실행하여 *FODC\_hang\_<timestamp>* 디렉토리에 저장되는 FODC 패키지로 수동 FODC 데이터를 수집합니다.

전체 수집 스크립트를 사용하여 특정 데이터베이스에서 성능 문제가 있는 동안 데이터를 수집하려면 다음을 입력하십시오.

```
db2fodc -db SAMPLE -perf full
```

데이터 컬렉션은 데이터베이스 SAMPLE로 제한됩니다. 새 디렉토리 *FODC\_perf\_<timestamp>*가 현재 진단 경로 아래에 작성됩니다. db2cos\_perf 스크립트를 실행하여 수동 FODC 데이터를 하나 이상의 파일로 수집합니다. 이 파일은 *FODC\_perf\_<timestamp>* 디렉토리에 놓입니다.

데이터베이스 관리 프로그램을 중지하지 않고 디폴트(기본) 모드에서 인덱스 오류에 대한 데이터를 수집하려면 다음을 입력하십시오.

```
db2fodc -indexerror FODC_IndexError_directory
```

db2cos\_indexerror\_short(.bat) 스크립트를 실행하여 수동 FODC 데이터를 하나 이상의 파일로 수집합니다. 파일은 디렉토리 *FODC\_IndexError\_directory*에 놓입니다.

## 사용 시 참고사항

db2fodc -hang 및 db2fodc -perf는 여러 개의 실제 노드가 있는 다중 파티션된 환경에서 실행할 수 있습니다. 이 환경에서

```
rah ";db2fodc -hang <full | basic> -alldbpartitionum other_options"
```

또는

```
rah ";db2fodc -perf <full | basic> -alldbpartitionum other_options"
```

위의 명령은 모든 DB2 DPF 노드에서 잠재된 정지 또는 심각한 성능 문제가 있는 동안 단일 호출로 db2fodc를 호출하기 위해 사용해야 합니다. 사용자 상호 작용이 필요하지 않은 수집 모드를 설정하려면 서버옵션 full 또는 basic을 사용하십시오. 옵션 -alldbpartitionum 및 -dbpartitionnum은 논리적 파티션 번호(다르지 않은 실제 머신)에 대해서만 작동합니다. 디폴트로, 현재 파티션 번호의 정보만 수집됩니다.

db2fodc -hang 및 db2fodc -perf는 *FODC\_symptom* 디렉토리에 위치하는 *db2fodc\_symptom.log* 로그 파일을 사용합니다. 여기서 *symptom*은 hang 또는 perf입니다. 이 파일 내에서, db2fodc는 FODC 서브디렉토리에 있는 FODC 패키지를 설명하는 메타데이터와 상태 정보도 저장합니다. 이 파일은 FODC 유형 정보, 데이터 콜렉션 시작 및/또는 종료 시간소인, FODC 패키지의 분석에 유용한 기타 정보가 포함됩니다.

db2fodc -hang 및 db2fodc -perf에 의한 데이터 콜렉션 동작은 매개변수를 통해 제어되므로, 실행되는 스크립트를 변경하여 사용자 정의할 수 있습니다. UNIX 시스템에서 데이터 콜렉션을 사용자 정의하려면, /bin/db2cos\_symptom에 있는 스크립트를 /adm/db2cos\_symptom으로 복사하십시오. 여기서 *symptom*은 hang 또는 perf입니다. 이 새 디렉토리에서 원하는 대로 스크립트를 수정하십시오. Windows 시스템에서는 단지 디폴트 스크립트 #bin#db2cos\_symptom.bat를 수정하면 됩니다. UNIX 시스템에서는 db2fodc가 우선 /adm/db2cos\_symptom에서 스크립트를 실행하려고 시도하고, 스크립트를 찾지 못하면 /bin/db2cos\_symptom에서 원래 스크립트를 실행합니다. Windows 시스템에서는 항상 스크립트 #bin#db2cos\_symptom.bat가 실행됩니다.

수집된 데이터는 디폴트 진단 경로의 기존 또는 새 서브디렉토리 *FODC\_symptom\_timestamp*에 저장되는 몇 개의 파일에 기록됩니다. 여기서 *symptom*



은 문제점 증상이고 *timestamp*는 자동 또는 수동 FODC 호출 시간입니다. db2diag 로그 파일 진단 메시지는 이 특정의 자동 또는 수동 FODC에 사용되는 디렉토리 이름에 대해 통지하기 위해 로그됩니다.

db2fodc -indexerror는 여러 개의 실제 또는 논리 노드가 있는 다중 파티션된 환경에서 실행할 수도 있습니다. 이 환경에서

```
db2_all "<<+node#< db2fodc -indexerror FODC_IndexError_directory <basic | full>"
```

위의 명령을 사용하여 db2fodc를 호출하여 특정 DB2 DPF 파티션 번호에 있는 인덱스 오류에 대한 정보를 수집해야 합니다. node#를 특정 DPF 노드의 번호로 바꾸십시오. 이 번호는 디렉토리 이름 <FODC\_IndexError\_timestamp\_PID\_EDUID\_Node#>에서 마지막 숫자입니다. 절대 경로와 함께 *FODC\_IndexError\_directory*를 지정해야 합니다. 서브옵션 full 또는 basic을 사용하여 수집 모드를 설정할 수 있습니다. 절대 경로는 db2\_all 명령과 함께 db2fodc -indexerror를 사용하는 경우에만 필요합니다.

db2fodc -indexerror은 명령 진행을 출력하고 메시지(있는 경우)를 db2diag 로그 파일에 로그합니다.

db2fodc -indexerror에는 입력으로 db2cos\_indexerror\_short(.bat) 스크립트 및 또는 db2cos\_indexerror\_long(.bat) 스크립트를 포함하는 *FODC\_IndexError\_directory*가 필요합니다.

db2cos\_indexerror\_short(.bat) 및 db2cos\_indexerror\_long(.bat) 스크립트는 자동 인덱스 오류 FODC 프로세스 중에 작성된 *FODC\_IndexError\_directory*에 위치됩니다. 스크립트는 여러 개의 db2dart 명령을 포함합니다. 스크립트를 실행하면, 스크립트가 위치한 동일한 *FODC\_IndexError\_directory*에서, 생성된 db2dart 보고서를 찾을 수 있습니다. 수동 db2fodc -indexerror 명령을 발행하면 새 디렉토리가 작성되지 않습니다. db2fodc -indexerror은 자동 인덱스 오류 FODC 프로세스에 의해 작성된 동일한 *FODC\_IndexError\_directory*에서 새 db2dart 보고서를 생성합니다.

*FODC\_IndexError\_directory* 이름을 바꾸거나 이동하지 마십시오. 스크립트의 db2dart 명령은 보고서를 올바르게 생성하기 위해 이 디렉토리를 필요로 합니다.

db2fodc -indexerror을 수동으로 실행해야 하는 경우, 기존 db2dart 보고서를 찾기 위해 *FODC\_IndexError\_directory*를 점검하십시오. 보고서의 확장자는 .rpt 및 .rpthex입니다. 보고서가 있으면 이 보고서의 이름을 바꾸거나 보고서를

*FODC\_IndexError\_directory* 아래의 서브디렉토리로 이동한 후 수동으로 db2fodc -indexerror을 실행해야 합니다. 그러면 수집하기 위한 db2support 도구에 대한 기존 보고서가 보존되어 db2fodc -indexerror이 새 db2dart 보고서를 작성할 수 있게 됩니다.

### FODC 패키지 준비

db2fodc 명령 실행이 완료되면, db2support 도구를 실행하여 결과 진단 파일을 수집하고 IBM Support에 제출할 FODC 패키지를 준비해야 합니다.

디폴트로, db2support는 진단 데이터 디렉토리 경로에서 발견되는 모든 F0DC\_XXX\_XXX 디렉토리를 수집합니다. 이는 진단 정보를 위한 IBM Support의 추가 요청을 피하기 위해 수행됩니다.

### db2fodc 진단 데이터 콜렉션

**db2fodc -hang**은 다음 정보를 수집합니다.

db2fodc -hang은 다음 정보를 수집합니다.

- 기본 운영 체제 정보. 문제점의 원인은 OS 레벨, 패치 등이 될 수 있습니다.
- 기본 DB2 구성 정보.
- 운영 체제 모니터 정보 vmstat, netstat, iostat 등
  - 최소한 두 번의 반복: 저장된 시간소인 사용
- 부분 호출 스택: 맨 위 CPU 에이전트의 DB2 스택 추적.
- 운영 체제 추적: AIX에서의 추적.
- db2pd에서 수집된 진단 정보.
- DB2 추적.
- 전체 DB2 호출 스택.
- DB2 구성 정보의 두 번째 라운드.
  - 두 번째 DB2 추적 콜렉션 포함.
- 스냅샷 정보: db2 get snapshot for 데이터베이스, 응용프로그램, 테이블 등
  - 정보는 여러 개의 논리 노드가 있는 경우 노드마다 수집됩니다.

**db2fodc -perf**는 다음 정보를 수집하여 시스템을 모니터링합니다.

- 스냅샷
- 스택 추적
- 가상 메모리(Vmstat)
- 입출력(I/O) 정보(Iostat)
- 추적
- 경우에 따라 일부 다른 정보. 세부사항은 스크립트를 참조하십시오.

**db2fodc -indexerror**은 다음 정보를 수집합니다.

- 기본 모드
  - db2cos\_indexerror\_short(.bat) 스크립트가 실행됩니다. 추가적인 세부사항은 스크립트를 참조하십시오.
  - 적용 가능한 db2dart 명령이 스크립트에 존재하는 경우, db2dart /DD 및/또는 db2dart /DI 데이터 형식화 조치는 100개로 제한된 페이지 수를 사용하여 실행됩니다.

- 전체 모드
  - db2cos\_indexerror\_short(.bat) 및 db2cos\_indexerror\_long(.bat) 스크립트가 실행됩니다. 추가적인 세부사항은 스크립트를 참조하십시오.
  - 적용 가능한 db2dart 명령이 스크립트 db2cos\_indexerror\_short(.bat)에 존재하는 경우, db2dart /DD 및/또는 db2dart /DI 데이터 형식화 조치는 100개로 제한된 페이지 수를 사용하여 실행됩니다.
  - 적용 가능한 db2dart 명령이 스크립트 db2cos\_indexerror\_long(.bat)에 존재하는 경우, db2dart /DD 및/또는 db2dart /DI 데이터 형식화 조치는 페이지 수 제한 없이 실행됩니다.
  - 적용 가능한 db2dart 명령이 db2cos\_indexerror\_long(.bat) 스크립트에 존재하는 경우, db2dart /T 명령이 실행됩니다. 이 명령을 사용하려면 데이터베이스가 오프라인이어야 합니다.



---

## 제 201 장 db2fs - 첫 단계

사용자가 DB2에 대한 학습 및 사용을 시작하기 위해 필요한 기능에 대한 링크가 들어 있는 첫 단계 인터페이스를 시작합니다.

UNIX 운영 체제에서 db2fs는 sql1lib/bin 디렉토리에 있습니다. Windows 운영 체제에서 db2fs.exe는 DB2PATH\bin 디렉토리에 있습니다.

db2fs 명령을 발행하려면 다음 브라우저 중 하나가 설치되어야 합니다.

- Internet Explorer 6.0 이상
- Mozilla 1.7 이상
- Firefox 2.0 이상

### 권한 부여

*sysadm*

### 명령 구문

UNIX 운영 체제의 경우

```
▶▶ db2fs -h -b browser ▶▶
```

Windows 운영 체제의 경우

```
▶▶ db2fs ▶▶
```

### 명령 매개변수

UNIX 운영 체제의 경우

**-h** 명령 사용법 정보를 표시합니다.

**-b browser**

사용될 브라우저를 지정합니다. 지정되지 않는 경우 db2fs는 PATH에 지정된 디렉토리에서 브라우저를 검색합니다.

Windows 운영 체제의 경우

없음



## 제 202 장 db2gcf - DB2 인스턴스 제어

고가용성(HA) 클러스터에서와 같이, 보통 자동화된 스크립트를 통해 DB2 인스턴스를 시작, 중지 또는 모니터링합니다.

UNIX 운영 체제에서, 이 명령은 INSTHOME/sqlllib/bin에 위치합니다. 여기서 INSTHOME은 인스턴스 소유자의 홈 디렉토리입니다. Windows 시스템에서, 이 명령은 sqlllib\bin 서브디렉토리에 위치합니다.

### 권한 부여

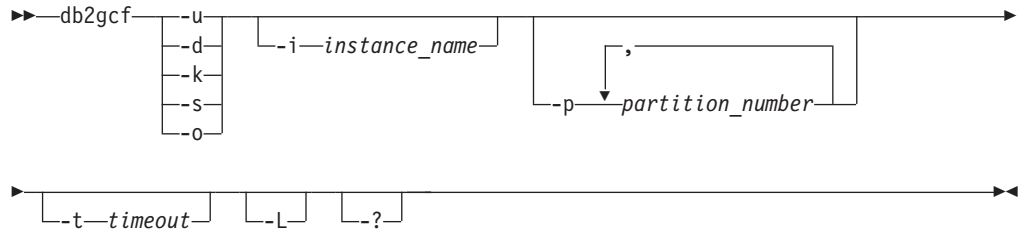
다음 중 하나가 필요합니다.

- 인스턴스 소유자
- Linux 및 UNIX 시스템에서의 루트 액세스나, Windows 운영 체제에서의 관리자 권한

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

- u 현재 데이터베이스 파티션 서버(노드)에서 지정된 인스턴스에 대해 지정된 데이터베이스 파티션을 시작합니다.
- d 지정된 인스턴스에 대해 지정된 데이터베이스 파티션을 중지합니다.
- k 지정된 인스턴스와 연관되는 모든 프로세스를 제거합니다.
- s 지정된 데이터베이스 파티션 및 지정된 인스턴스의 상태를 리턴합니다. 가능한 상태는 다음과 같습니다.
  - 사용 가능: 지정된 인스턴스에 대해 지정된 데이터베이스 파티션이 처리에 사용 가능합니다.
  - 작동 가능: 인스턴스가 설치되어 있지만 현재 사용 가능하지 않습니다.

- 작동 불가능: 인스턴스를 사용 가능 상태로 가져올 수 없습니다.
- o 가능한 조치마다 디폴트 시간종료를 리턴합니다. -t 매개변수에 값을 지정하여 모든 디폴트를 대체할 수 있습니다.
- i *instance\_name*  
조치를 수행할 대상 인스턴스 이름. 인스턴스 이름이 지정되지 않으면 DB2INSTANCE 값이 사용됩니다. 인스턴스 이름이 지정되지 않고 DB2INSTANCE가 설정되지 않은 경우, 다음 오류가 리턴됩니다.  
db2gcf Error: Neither DB2INSTANCE is set nor instance passed.
- p *partition\_number*  
파티션된 데이터베이스 환경에서, 로컬 노드에서 조치를 수행할 데이터베이스 파티션 번호를 지정합니다(리모트 파티션은 이 명령으로 모니터링하지 않음). 공백 없이, 쉼표로 구분하여 파티션 번호를 지정하십시오. 값을 지정하지 않은 경우 디폴트는 0입니다. 이 값은 단일 파티션 데이터베이스 환경에서 무시됩니다.
- t *timeout*  
시간종료(초). db2gcf 명령은 지정된 시간 내에 처리가 완료되지 않은 경우 리턴하지 못합니다. 가능한 조치마다 디폴트 시간종료가 있습니다. -t 매개변수에 값을 지정하여 이 모든 디폴트를 대체할 수 있습니다.
- L 오류 로깅을 사용합니다. 인스턴스 특정 정보는 인스턴스 로그 디렉토리의 db2diag 로그 파일에 로깅됩니다. 인스턴스 이외의 특정 정보는 시스템 로그 파일에 로깅됩니다.
- ? 도움말 정보를 표시합니다. 이 옵션이 지정되면 다른 모든 옵션은 무시되고 도움말 정보만 표시됩니다.

**예:**

1. 다음 예는 파티션 0에서 stevera 인스턴스를 시작합니다.

```
db2gcf -u -p 0 -i stevera
```

다음 출력이 리턴됩니다.

```
Instance : stevera
DB2 Start : Success
Partition 0 : Success
```

2. 다음 예는 파티션 0에서 인스턴스 stevera의 상태를 리턴합니다.

```
db2gcf -s -p 0 -i stevera
```

다음 출력이 리턴됩니다.

```
Instance : stevera
DB2 State
Partition 0 : Available
```

3. 다음 예는 파티션 0에서 stevera 인스턴스를 중지합니다.



```
db2gcf -d -p 0 -i stevera
```

다음 출력이 리턴됩니다.

```
Instance : stevera  
DB2 Stop : Success  
Partition 0 : Success
```

## 사용 시 참고사항

함께 사용하는 경우, -k 및 -p 매개변수는 지정된 파티션에서 모든 프로세스를 제거할 수 없습니다. 오히려 인스턴스의 모든 프로세스(모든 파티션)가 제거됩니다.

## 리턴 코드

다음은 이 명령의 모든 리턴 코드 목록입니다.

```
db2gcf Return Values :  
0 : db2 service(start,stop,kill) success or db2gcf -s status Available  
1 : db2 service(start,stop) failed or db2gcf -s status Not Available  
2 : db2gcf has been called with wrong number of parameters  
3 : gcfmodule failed to execute the requested service
```



## 제 203 장 db2gov - DB2 조정자

데이터베이스에 대해 실행되는 응용프로그램의 동작을 모니터하고 변경합니다. 디폴트로 모든 데이터베이스 파티션에서 디먼이 시작되지만, 프론트엔드 유틸리티를 사용하여 특정 데이터베이스 파티션에서 단일 디먼을 시작할 수 있습니다.

**중요사항:** 새 워크로드 관리 기능이 DB2 버전 9.5에 도입되었으므로 DB2 조정자는 버전 9.7에서 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 *버전 9.7의 새로운 내용* 책에 있는 『DB2 조정자 및 Query Patroller는 사용되지 않음』 주제를 참조하십시오.

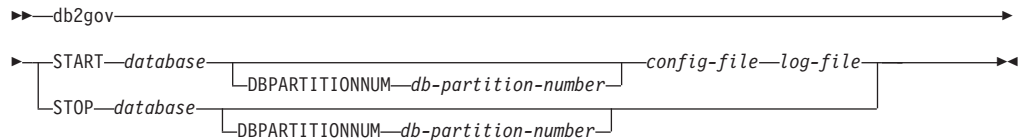
### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*

db2nodes.cfg 파일이 정의된 인스턴스를 갖는 환경에서는 db2\_all 명령을 호출하기 위한 권한 부여가 필요합니다. db2nodes.cfg 파일이 정의된 환경에는 파티션된 데이터베이스 환경 및 db2nodes.cfg에 데이터베이스 파티션이 정의된 단일 파티션 데이터베이스 환경이 포함됩니다.

### 명령 구문



### 명령 매개변수

#### START database

지정된 데이터베이스를 모니터할 조정자(governor) 디먼을 시작합니다. 데이터베이스 이름 또는 데이터베이스 별명을 지정할 수 있습니다. 지정된 이름은 조정자 구성 파일에서 지정된 이름과 동일해야 합니다. 모니터링하는 각 데이터베이스에 대해 하나의 디먼이 실행됩니다. 파티션된 데이터베이스 환경에서는 각 데이터베이스 파티션에 대해 하나의 디먼이 실행됩니다. 조정자가 둘 이상의 데이터베이스에 대해 실행 중인 경우 해당 데이터베이스 서버에서 둘 이상의 디먼이 실행 중입니다.

### **DBPARTITIONNUM** *db-partition-number*

조정자 디먼을 시작 또는 중지할 데이터베이스 파티션을 지정합니다. 지정된 숫자는 데이터베이스 파티션 구성 파일에서 지정된 번호와 동일해야 합니다.

#### *config-file*

데이터베이스를 모니터링할 때 사용할 구성 파일을 지정합니다. 구성 파일의 디폴트 위치는 `sqllib` 디렉토리입니다. 지정된 파일이 그곳에 없는 경우 프론트엔드는 지정된 이름이 파일의 전체 이름이라고 가정합니다.

#### *log-file*

조정자가 로그 레코드를 쓰는 파일의 기본 이름을 지정합니다. 로그 파일은 `sqllib` 디렉토리의 `log` 서브디렉토리에 저장됩니다. 조정자가 실행 중인 데이터베이스 파티션 수가 자동으로 로그 파일 이름에 추가됩니다. 예: `mylog.0`, `mylog.1`, `mylog.2`.

### **STOP** *database*

지정된 데이터베이스를 모니터링 중인 조정자 디먼을 중지합니다. 파티션된 데이터베이스 환경에서는 프론트엔드 유틸리티가 데이터베이스 파티션 구성 파일 `db2nodes.cfg`를 읽어서 모든 데이터베이스 파티션의 조정자를 중지합니다.

## 사용 시 참고사항

조정자 구성 파일의 `[action]` 절에서 `nice nnn` 매개변수를 설정하여 응용프로그램에 대해 작업 중인 에이전트의 상대적 우선순위를 올리거나 내릴 수 있습니다. 추가 정보는 *Tuning Database Performance* 안내서의 『조정자 규칙 요소』를 참조하십시오.

주: AIX 5.3 이상에서는 인스턴스 소유자가 `CAP_NUMA_ATTACH` 기능이 있어서 응용프로그램에 대해 작업 중인 에이전트의 상대적 우선순위를 올릴 수 있어야 합니다. 이 기능을 부여하려면 `root`로 로그인하여 다음 명령을 실행하십시오.

```
chuser capabilities=CAP_NUMA_ATTACH,CAP_PROPAGATE
```

## 호환성

버전 8 이전 버전과의 호환성:

- `DBPARTITIONNUM`에 키워드 `NODENUM`을 대체할 수 있습니다.

## 제 204 장 db2govlg - DB2 조정자 로그 쿼리

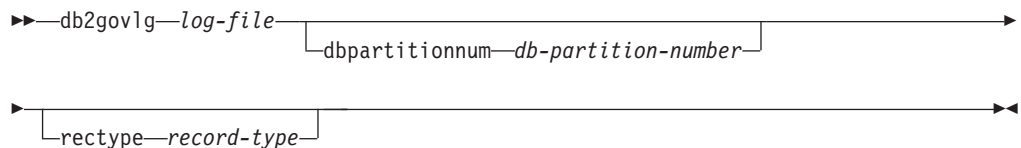
조정자(governor) 로그 파일에서 지정된 유형의 레코드를 추출합니다. DB2 조정자는 데이터베이스에 대해 실행되는 응용프로그램의 동작을 모니터하고 변경합니다.

중요사항: 새 워크로드 관리 기능이 DB2 버전 9.5에 도입되었으므로 DB2 조정자는 버전 9.7에서 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 버전 9.7의 새로운 내용 책에 있는 『DB2 조정자 및 Query Patroller는 사용되지 않음』 주제를 참조하십시오.

### 권한 부여

없음

### 명령 구문



### 명령 매개변수

*log-file*

쿼리할 하나 이상의 로그 파일의 기본 이름.

**dbpartitionnum** *db-partition-number*

조정자가 실행 중인 데이터베이스 파티션 수.

**rectype** *record-type*

쿼리할 레코드의 유형입니다. 유효한 레코드 유형은 다음과 같습니다.

- START
- FORCE
- NICE
- ERROR
- WARNING
- READCFG
- STOP
- ACCOUNT

## 호환성

버전 8 이전 버전과의 호환성:

- dbpartitionnum에 키워드 nodenum을 대체할 수 있습니다.

## 제 205 장 db2gmap - 분산 맵 가져오기

데이터베이스가 이미 설정되어 있고 해당 데이터베이스 파티션 그룹이 정의된 경우, db2gmap은 카탈로그 파티션 데이터베이스 서버에서 데이터베이스 파티션 그룹 또는 데이터베이스 테이블의 분산 맵을 가져옵니다.

### 권한 부여

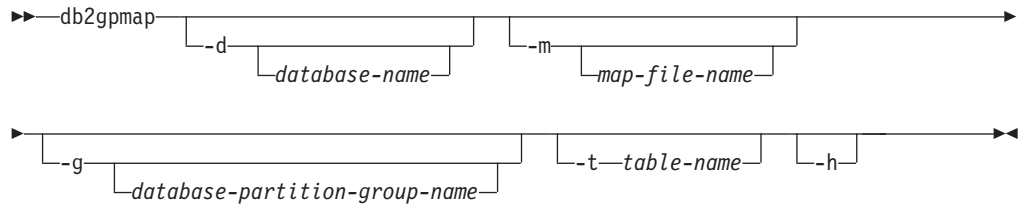
다음 두 가지 모두:

- 시스템 카탈로그 테이블에 대한 읽기 액세스
- db2gmap.bnd에 대한 BIND 및 EXECUTE 패키지 특권

### 필수 연결

db2gmap을 사용하기 전에 데이터베이스 관리 프로그램을 시작하고 db2gmap.bnd가 데이터베이스에 바인드되어야 합니다. 아직 바인드되지 않은 경우 db2gmap이 파일 바인드를 시도합니다.

### 명령 구문



### 명령 매개변수

#### **-d** *database-name*

분산 맵을 생성할 데이터베이스의 이름을 지정합니다. 데이터베이스 이름이 지정되지 않으면 DB2DBDFT 환경 변수 값이 사용됩니다. DB2DBDFT가 설정되지 않는 경우 디폴트값은 SAMPLE 데이터베이스입니다.

#### **-m** *map-file-name*

분산 맵이 저장될 완전한 파일 이름을 지정합니다. 디폴트값은 db2split.map입니다.

#### **-g** *database-partition-group-name*

분산 맵을 생성할 데이터베이스 파티션 그룹의 이름을 지정합니다. 디폴트값은 IBMDEFAULTGROUP입니다.

**-t** *table-name*

테이블의 이름

**-h** 사용법 정보를 표시합니다.

**예:**

다음 예는 SAMPLE 데이터베이스에 있는 ZURBIE.SALES 테이블의 분산 맵을 C:\wpmaps\zurbie\_sales.map 파일로 추출합니다.

```
db2gpm -d SAMPLE -m C:\wpmaps\zurbie_sales.map -t ZURBIE.SALES
```



---

## 제 206 장 db2hc - Health Center 시작

Health Center를 시작합니다. Health Center는 데이터베이스 시스템의 전체 상태를 보는 데 사용되는 그래픽 인터페이스입니다. Health Center를 사용하면 Health 표시기의 정보에 대한 세부사항과 권장사항을 보고 경보 해결을 위한 권장 조치를 수행할 수 있습니다.

**중요사항:** 이 명령은 제어 센터와 관련된 구성요소가 버전 9.7에서 사용되지 않기 때문에 사용되지 않으며 이후 릴리스에서 제거됩니다. 자세한 정보는 *버전 9.7의 새로운 내용* 책에 있는 『제어 센터 도구 및 DB2 Administration Server(DAS)는 사용되지 않음』 주제를 참조하십시오.

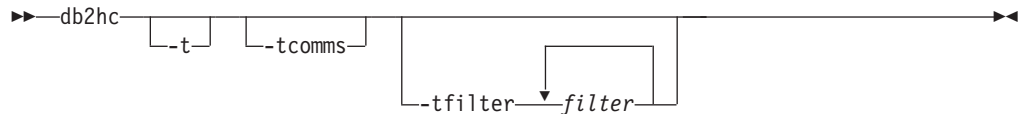
### 권한 부여

정보를 보기 위해 특수 권한이 필요하지는 않습니다. 조치를 수행하려면 적합한 권한이 필요합니다.

### 필요한 연결

인스턴스

### 명령 구문



### 명령 매개변수

**-t** 초기화 코드에 대해 NavTrace를 활성화합니다. 이 옵션은 DB2 Support의 지시가 있는 경우에만 사용해야 합니다.

#### **-tcomms**

통신 이벤트로 추적을 제한합니다. 이 옵션은 DB2 Support의 지시가 있는 경우에만 사용해야 합니다.

#### **-tfilter filter**

지정된 하나 이상의 필터가 포함된 항목으로 추적을 제한합니다. 이 옵션은 DB2 Support의 지시가 있는 경우에만 사용해야 합니다.



---

## 제 207 장 db2iauto - 인스턴스 자동 시작

시스템이 재시작될 때마다 인스턴스 자동 시작을 사용하거나 사용하지 않습니다. 이 명령은 Linux 및 UNIX 시스템에서만 사용할 수 있습니다.

### 권한 부여

다음 중 하나가 필요합니다.

- Root 권한
- *sysadm*

### 필수 연결

없음

### 명령 구문

```
▶▶ db2iauto [-on|-off] instance-name ▶▶▶▶
```

### 명령 매개변수

- on 지정된 인스턴스에 자동 시작을 사용합니다.
- off 지정된 인스턴스에 자동 시작을 사용하지 않습니다.

*instance-name*

인스턴스의 로그인 이름.



---

## 제 208 장 db2iclus - Microsoft Cluster Server

사용자가 MSCS(Microsoft Cluster Server) 환경에서 DAS(DB2 Administration Server)와 인스턴스를 추가, 삭제(drop), 이주 및 이주 해제할 수 있습니다. 이 명령은 Windows 플랫폼에서만 사용할 수 있습니다.

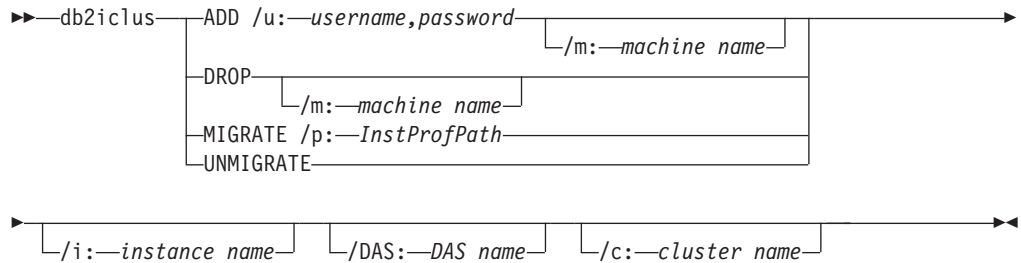
### 권한 부여

태스크가 수행될 머신에서는 로컬 관리자 권한이 필요합니다. 인스턴스에 리모트 머신을 추가하거나 인스턴스에서 리모트 머신을 제거하는 경우 해당 머신에서 로컬 관리자 권한이 필요합니다.

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

**ADD** DB2 MSCS 인스턴스에 MSCS 노드를 추가합니다.

#### **DROP**

DB2 MSCS 인스턴스에서 MSCS 노드를 제거합니다.

#### **MIGRATE**

비MSCS 인스턴스를 MSCS 인스턴스로 이주합니다.

#### **UNMIGRATE**

MSCS 이주를 실행 취소합니다.

#### **/DAS:DAS name**

DAS 이름을 지정합니다. 이 옵션은 DB2 Administration Server에 대해 클러스터 조작을 수행할 때 필요합니다.

#### **/c:cluster name**

디폴트/현재 클러스터와 다른 MSCS 클러스터 이름을 지정합니다.

**/p:instance profile path**

인스턴스 프로파일 경로를 지정합니다. 이 경로가 클러스터 디스크에 있어야 MSCS 클러스터의 머신에서 DB2가 활성화될 때 액세스할 수 있습니다. 이 옵션은 비MSCS 인스턴스를 MSCS 인스턴스로 이주할 때 필요합니다.

**/u:username,password**

DB2 서비스의 어카운트 이름과 암호를 지정합니다. 이 옵션은 DB2 MSCS 파티션된 데이터베이스 인스턴스에 다른 MSCS 노드를 추가할 때 필요합니다.

**/m:machine name**

MSCS 노드 추가 및 제거를 위한 리모트 컴퓨터 이름을 지정합니다.

**/i:instance name**

디폴트/현재 인스턴스와 다른 인스턴스 이름을 지정합니다.

**예:**

이 예에서는 db2iclus 명령을 사용하여 DB2 인스턴스가 핫 대기 구성(WA26 머신과 WA27 머신으로 구성)에서 실행되도록 수동으로 구성하는 방법을 나타냅니다.

1. 시작하려면 두 머신에 모두 MSCS와 DB2 Enterprise Server Edition가 설치되어 있어야 합니다.

2. 다음과 같이 WA26 머신에 새 DB2 인스턴스를 작성하십시오.

```
db2icrt DB2
```

3. Windows Services 대화 상자에서 인스턴스가 수동으로 시작되도록 구성되었는지 확인하십시오.

4. DB2 인스턴스가 실행 중인 경우 DB2STOP 명령을 사용하여 중지하십시오.

5. 다음과 같이 WA26에서 DB2 자원 유형을 설치하십시오.

```
c:>db2wolfi i  
ok
```

db2wolfi 명령이 "오류 : 183"을 리턴하는 경우 이미 설치되어 있는 것입니다. 확인하려면 자원 유형을 삭제하고 다시 추가할 수 있습니다. 또한 자원 유형이 없으면 클러스터 관리자에 나타나지 않습니다.

```
c:>db2wolfi u  
ok  
c:>db2wolfi i  
ok
```

6. WA26에서 db2iclus 명령을 사용하여 DB2 인스턴스를 클러스터된 인스턴스로 변환하십시오.

```
c:\>db2iclus migrate /i:db2 /c:mycluster /m:wa26 /p:p:\db2profs
```

DBI1912I DB2 클러스터 명령이 성공적이었습니다.

설명: 사용자 요청이 처리되었습니다.

사용자 응답: 조치가 필요하지 않습니다.

p:#db2profs 디렉토리는 클러스터된 드라이브에 이미 존재해야 합니다. 이 드라이브는 현재 WA26 머신도 소유하고 있어야 합니다.

7. 다음과 같이 WA26에서 db2iclus 명령을 사용하여 DB2 클러스터 목록에 다른 머신을 추가하십시오.

```
c:#>db2iclus add /i:db2 /c:mycluster /m:wa27
```

```
DBI1912I DB2 클러스터 명령이 성공적이었습니다.  
설명: 사용자 요청이 처리되었습니다.  
사용자 응답: 조치가 필요하지 않습니다.
```

이 명령은 클러스터 내 각 연속 머신에 대해 실행되어야 합니다.

8. 클러스터 관리자에서 새 "DB2 Group" 그룹을 작성하십시오.
9. 클러스터 관리자에서 실제 디스크 자원 디스크 O와 디스크 P를 DB2 Group으로 이동하십시오.
10. 클러스터 관리자에서 공용 네트워크에 있는 "IP 주소" 유형의 새 "mcs5" 자원 유형을 작성하십시오. 이 자원은 DB2 Group에도 속해야 합니다. 이 주소는 사용량이 많은 IP 주소이므로 이 주소와 네트워크의 머신이 일치해서는 안 됩니다. IP 주소 자원 유형을 온라인으로 설정하고 리모트 머신에서 해당 주소를 Ping할 수 있는지 확인하십시오.
11. 클러스터 관리자에서 DB2 Group에 속하는 "DB2" 유형의 새 자원을 작성하십시오. 이 자원의 이름은 인스턴스 이름과 정확하게 일치해야 하므로 이 경우 DB2입니다. 클러스터 관리자가 DB2 자원과 연관된 종속성에 대한 프롬프트를 표시하는 경우 디스크 O, 디스크 P 및 mcs5에 종속되는지 확인하십시오.
12. 필요한 경우 클러스터 관리자를 통해 DB2\_FALLBACK 프로파일 변수를 사용하여 DB2 Group이 폴백(fallback)하도록 구성하십시오.
13. 모든 데이터를 디스크 O 및 디스크 P에 배치하여 모든 데이터베이스를 작성 또는 리스토어하십시오.
14. 장애 복구 구성을 테스트하십시오.

## 사용 시 참고사항

MSCS 장애 복구 환경에서 실행될 인스턴스를 이주하려면 먼저 현재 머신의 인스턴스를 이주한 다음 db2iclus와 ADD 옵션을 함께 사용하여 인스턴스에 다른 MSCS 노드를 추가해야 합니다.

MSCS 인스턴스를 일반 인스턴스로 다시 되돌리려면 먼저 db2iclus와 DROP 옵션을 함께 사용하여 인스턴스에서 다른 모든 MSCS 노드를 제거해야 합니다. 다음으로 현재 머신의 인스턴스에 대한 이주를 실행 취소해야 합니다.





## 제 209 장 db2icrt - 인스턴스 작성

DB2 인스턴스를 작성합니다.

Linux 및 UNIX 운영 체제에서 이 유틸리티는 *DB2DIR/instance* 디렉토리에 있으며, *DB2DIR*은 DB2 데이터베이스 시스템의 현재 버전이 설치된 설치 위치입니다. Windows 운영 체제에서 이 유틸리티는 **DB2PATH\bin** 디렉토리에 위치하며, **DB2PATH**는 DB2 사본이 설치된 위치입니다.

db2icrt 명령은 인스턴스 소유자의 홈 디렉토리에 DB2 인스턴스를 작성합니다.

주: 이 명령은 Linux 및 UNIX 운영 체제의 DB2 데이터베이스 제품의 루트가 아닌 설치에 사용할 수 없습니다.

### 권한 부여

Linux 및 UNIX 운영 체제의 루트 액세스 또는 Windows 운영 체제의 로컬 관리자 권한.

### 명령 구문

Linux 및 UNIX 운영 체제의 경우

```
db2icrt [-h] [-d] [-a AuthType] [-p PortName]
        [-s InstType] [-u FencedID] InstName
```

Windows 운영 체제의 경우

```
db2icrt InstName [-s InstType] [-u UserName, Password]
                 [-p InstProfPath] [-h HostName] [-r PortRange]
                 [-j "TEXT_SEARCH ,servicename ,portnumber"] [-?]
```

### 명령 매개변수

Linux 및 UNIX 운영 체제의 경우

**-h | -?**

사용법 정보를 표시합니다.

**-d** 디버그 모드를 켭니다. DB2 Support가 지시할 때만 이 옵션을 사용하십시오.

**-a AuthType**

인스턴스의 인증 유형(SERVER, CLIENT 또는 SERVER\_ENCRYPT)을 지정합니다. 디폴트는 SERVER입니다.

**-p PortName**

인스턴스가 사용하는 포트 이름 또는 번호를 지정합니다. 이 옵션은 클라이언트 인스턴스에 적용되지 않습니다.

**-s InstType**

작성할 인스턴스의 유형을 지정합니다. db2icrt를 실행 중인 설치된 제품과 연관된 디폴트가 아닌 인스턴스를 작성 중일 때만 **-s** 옵션을 사용하십시오. 가능한 값은 다음과 같습니다.

**client** 클라이언트의 인스턴스를 작성하는 데 사용됩니다. 이것은 IBM Data Server Client, IBM Data Server Runtime Client 및 DB2 Connect Personal Edition에 대한 디폴트 인스턴스 유형입니다.

**standalone**

로컬 클라이언트가 있는 데이터베이스 서버에 대한 인스턴스를 작성하는 데 사용됩니다. DB2 Personal Edition의 디폴트 인스턴스 유형입니다.

**ese** DPF 지원과 함께 로컬 및 리모트 클라이언트가 있는 데이터베이스 서버에 대한 인스턴스를 작성하는 데 사용됩니다. 이것은 DB2 Enterprise Server Edition의 디폴트 인스턴스 유형입니다.

**wse** 로컬 및 리모트 클라이언트가 있는 데이터베이스 서버에 대한 인스턴스를 작성하는 데 사용됩니다. 이것은 DB2 Workgroup Server Edition, DB2 Express 또는 Express-C Edition 및 DB2 Connect Enterprise Edition의 디폴트 인스턴스 유형입니다.

DB2 제품은 디폴트 인스턴스 유형 및 디폴트보다 낮은 인스턴스 유형을 지원합니다. 예를 들어 DB2 Enterprise Server Edition은 ese, wse, standalone 및 client의 인스턴스 유형을 지원합니다.

**-u Fenced ID**

분리 사용자 정의 기능 및 분리(fenced) 스토어드 프로시저가 실행할 사용자 ID의 이름을 지정합니다. 클라이언트 인스턴스를 작성 중이 아닌 경우 **-u** 옵션이 필수입니다.

### *InstName*

운영 체제에서 기존 사용자의 이름이기도 한 인스턴스의 이름을 지정합니다. 이것은 db2icrt 명령의 마지막 인수여야 합니다.

### **Windows** 운영 체제의 경우

#### *InstName*

인스턴스 이름을 지정합니다.

#### **-s** *InstType*

작성할 인스턴스의 유형을 지정합니다. 현재는 4가지 유형의 DB2 인스턴스 유형이 있습니다. 가능한 값은 다음과 같습니다.

**client** 클라이언트의 인스턴스를 작성하는 데 사용됩니다. 이것은 IBM Data Server Client, IBM Data Server Runtime Client 및 DB2 Connect Personal Edition에 대한 디폴트 인스턴스 유형입니다.

#### **standalone**

로컬 클라이언트가 있는 데이터베이스 서버에 대한 인스턴스를 작성하는 데 사용됩니다. DB2 Personal Edition의 디폴트 인스턴스 유형입니다.

**ese** DPF 지원과 함께 로컬 및 리모트 클라이언트가 있는 데이터베이스 서버에 대한 인스턴스를 작성하는 데 사용됩니다.

*-s ese -u Username, Password*

옵션은 ESE 인스턴스 유형 및 DPF 인스턴스를 작성하기 위해 db2icrt와 함께 사용되어야 합니다.

**wse** 로컬 및 리모트 클라이언트가 있는 데이터베이스 서버에 대한 인스턴스를 작성하는 데 사용됩니다. 이것은 DB2 Workgroup Server Edition, DB2 Express 또는 Express-C Edition 및 DB2 Connect Enterprise Edition의 디폴트 인스턴스 유형입니다.

DB2 제품은 디폴트 인스턴스 유형 및 디폴트보다 낮은 인스턴스 유형을 지원합니다. 예를 들어 DB2 Enterprise Server Edition은 ese, wse, standalone 및 client의 인스턴스 유형을 지원합니다.

#### **-u** *Username, Password*

DB2 서비스의 어카운트 이름 및 암호를 지정합니다. 이 옵션은 파티션된 데이터베이스 인스턴스를 작성할 때 필수입니다.

#### **-p** *InstProfPath*

인스턴스 프로파일 경로를 지정합니다.

#### **-h** *HostName*

현재 머신에 대해 둘 이상이 있는 경우 디폴트 TCP/IP 호스트 이름을 겹쳐씀

니다. TCP/IP 호스트 이름은 디폴트 데이터베이스 파티션(데이터베이스 파티션 0)을 작성할 때 사용됩니다. 이 옵션은 파티션된 데이터베이스 인스턴스에 대해서만 유효합니다.

**-r PortRange**

MPP 모드에서 실행할 때 파티션된 데이터베이스 인스턴스가 사용할 TCP/IP 포트의 범위를 지정합니다. 예: `-r 50000,50007`. 이 옵션이 지정되면 로컬 시스템의 서비스 파일이 다음 항목으로 갱신됩니다.

DB2_InstName	baseport/tcp
DB2_InstName_END	endport/tcp

**/j "TEXT\_SEARCH"**

서비스 이름 및 TCP/IP 포트 번호에 대해 생성된 디폴트값을 사용하여 DB2 텍스트 검색 서버를 구성합니다. 인스턴스 유형이 클라이언트이면 이 매개변수를 사용할 수 없습니다.

**/j "TEXT\_SEARCH, servicename"**

제공되는 서비스 이름 및 자동으로 생성된 포트 번호를 사용하여 DB2 텍스트 검색 서버를 구성합니다. 서비스 파일에서 서비스 이름에 포트 번호가 지정된 경우 지정된 포트 번호를 사용합니다.

**/j "TEXT\_SEARCH, servicename, portnumber"**

제공된 서비스 이름 및 포트 번호를 사용하여 DB2 텍스트 검색 서버를 구성합니다.

**/j "TEXT\_SEARCH, portnumber"**

디폴트 서비스 이름 및 제공된 포트 번호를 사용하여 DB2 텍스트 검색 서버를 구성합니다. 유효한 포트 번호는 1024 - 65535 범위에 있어야 합니다.

**-?** 사용법 정보를 표시합니다.

**예:**

- AIX 머신에서 사용자 ID db2inst1에 대한 인스턴스를 작성하려면 다음 명령을 실행하십시오.

클라이언트 머신에서:

```
DB2DIR/instance/db2icrt db2inst1
```

서버 머신에서:

```
DB2DIR/instance/db2icrt -u db2fenc1 db2inst1
```

여기서 db2fenc1은 분리 사용자 정의 기능 및 분리(fenced) 스토어드 프로시저가 실행할 사용자 ID입니다.

## 사용 시 참고사항

- Linux 및 UNIX 운영 체제에서는 db2icrt에 의해 instance home/sqlllib/db2tss/config 폴더가 작성됩니다. sqlllib 디렉토리 외부의 영역에 대한 기호 링크를 사용하는 것이 좋습니다.
- -s 옵션은 시스템의 전체 기능을 사용하지 않는 인스턴스를 작성하려는 상황을 위한 것입니다. 예를 들어 UNIX 운영 체제에서 ESE(Enterprise Server Edition)를 사용 중이지만 파티션 기능을 원하지 않는 경우, 옵션 -s WSE를 사용하여 WSE(Workgroup Server Edition) 인스턴스를 작성할 수 있습니다.
- Microsoft Cluster Server를 지원하는 DB2 인스턴스를 작성하려면 먼저 인스턴스를 작성한 후 db2mscs 명령을 사용하여 MSCS 인스턴스에서 실행하도록 이주하십시오.
- Linux 및 UNIX 운영 체제에 대하여 하나의 인스턴스만 사용자 이름 아래에서 작성될 수 있습니다. 이미 인스턴스가 있는 사용자 이름에서 인스턴스를 작성하려는 경우, 새 인스턴스를 작성하기 전에 기존 인스턴스를 삭제(drop)해야 합니다.
- DB2 인스턴스를 작성할 때 다음 제한사항을 고려하십시오.
  - 기존 ID가 DB2 인스턴스를 작성하는 데 사용되는 경우 ID가 잠기지 않고 암호가 만기되지 않았는지 확인하십시오.
- db2isetup 명령을 사용하여 지원되는 모든 Linux 및 UNIX 운영 체제에서 그래픽 인터페이스를 사용하여 DB2 인스턴스를 작성 및 갱신할 수도 있습니다.
- Linux 및 UNIX 시스템에서, login 명령 대신 su 명령을 사용하여 루트 사용자가 되는 경우 - 옵션과 함께 su 명령을 발행하여 프로세스 환경이 login 명령을 사용하여 시스템에 로그인한 것처럼 설정되도록 표시해야 합니다.
- Linux 및 UNIX 운영 체제에서 루트 사용자에게 대한 DB2 인스턴스 환경을 소스해서는 안됩니다. DB2 인스턴스 환경을 소스했을 때 db2icrt 명령을 실행하는 것은 지원되지 않습니다.
- AIX 6.1(또는 그 이상)에서 시스템 워크로드 파티션(WPAR) 전역 환경의 공유 DB2 사본으로부터 이 명령을 실행할 때 이 명령은 루트 사용자로서 실행되어야 합니다.
- Windows 운영 체제에서, IBM Tivoli Monitoring for Databases: DB2 Agent가 설치되고 DB2 인스턴스가 작성되는 경우 다음이 성립하면 Monitoring Agent for DB2 인스턴스도 작성됩니다.
  - DB2 인스턴스 유형이 standalone, wse 또는 ese입니다.
  - 디폴트 DB2 사본에 ITM 에이전트 구성요소가 설치되었습니다.
  - DB2 인스턴스가 버전 9.5(또는 그 이상)에 있습니다.
  - 기존 ITM for Databases 제품이 없습니다.

또한 Monitoring Agent for DB2 인스턴스 작성 후에 Monitoring Agent for DB2 인스턴스 파일, NT 서비스 및 레지스트리 항목도 작성됩니다.



---

## 제 210 장 db2idrop - 인스턴스 제거

db2icrt에 의해 작성된 DB2 인스턴스를 제거합니다. db2idrop을 발행 중인 동일한 DB2 사본에 대해 db2ilist에 의해 나열되는 인스턴스만 삭제할 수 있습니다.

Linux 및 UNIX 운영 체제에서 이 유틸리티는 *DB2DIR/instance* 디렉토리에 있으며, *DB2DIR*은 DB2 데이터베이스 시스템의 현재 버전이 설치된 설치 위치입니다. Windows 운영 체제에서 이 유틸리티는 **DB2PATH**\bin 디렉토리에 위치하며, **DB2PATH**는 DB2 사본이 설치된 위치입니다.

주: Linux 및 UNIX 운영 체제의 루트가 아닌 사용자가 설치한 DB2 인스턴스는 이 명령을 사용하여 삭제할 수 없습니다. 유일한 옵션은 루트가 아닌 DB2 사본을 설치 제거하는 것입니다. 자세한 내용은 아래 사용 시 참고사항을 참조하십시오.

### 권한 부여

Linux 및 UNIX 운영 체제에서 Root 액세스 또는 Windows 운영 체제의 로컬 관리자.

### 명령 구문

Linux 및 UNIX 운영 체제의 경우

```
▶▶ db2idrop—InstName [-d] [-f] [-h] [-?]
```

Windows 운영 체제의 경우

```
▶▶ db2idrop—InstName [-f] [-h]
```

### 명령 매개변수

Linux 및 UNIX 운영 체제의 경우

*InstName*

인스턴스 이름을 지정합니다.

- d DB2 Service가 사용하도록 디버그 모드를 입력합니다.
- f 이 매개변수는 사용되지 않습니다. 응용프로그램 강제 중단 플래그를 지정합니다. 이 플래그가 지정되면 해당 인스턴스를 사용하는 모든 응용프로그램이 강제로 종료됩니다.

**-h | -?**

사용법 정보를 표시합니다.

## Windows 운영 체제의 경우

*InstName*

인스턴스 이름을 지정합니다.

**-f** 응용프로그램 강제 중단 플래그를 지정합니다. 이 플래그가 지정되면 해당 인스턴스를 사용하는 모든 응용프로그램이 강제로 종료됩니다.

**-h** 사용법 정보를 표시합니다.

## 예:

- Linux 또는 UNIX 운영 체제에서 다음 명령을 발행하여 db2inst1을 작성한 경우,

```
/opt/IBM/db2/copy1/instance/db2icrt -u db2fenc1 db2inst1
```

db2inst1을 삭제하려면 다음 명령을 실행해야 합니다.

```
/opt/IBM/db2/copy1/instance/db2idrop db2inst1
```

## 사용 시 참고사항

- 인스턴스가 삭제되기 전에 DB2 데이터베이스 관리 프로그램이 중지되었고 인스턴스에 액세스하는 DB2 데이터베이스 응용프로그램이 연결이 끊어지고 종료되는지 확인하십시오. 인스턴스와 연관된 DB2 데이터베이스를 백업할 수 있으며, 구성 데이터는 필요한 경우 나중에 참조하기 위해 저장할 수 있습니다.
- db2idrop 명령은 데이터베이스를 제거하지 않습니다. 데이터베이스가 더 이상 필요 없는 경우 먼저 데이터베이스를 제거하십시오. 데이터베이스가 제거되지 않는 경우 데이터베이스는 항상 동일한 릴리스의 다른 DB2 사본 아래에 카탈로그될 수 있으며 계속 사용될 수 있습니다.
- DB2 텍스트 검색 구성을 저장하기 원하며 인스턴스 데이터베이스를 재사용할 계획인 경우 db2idrop 명령을 발행하기 전에 config 디렉토리(UNIX에서는 *instance\_home/sqllib/db2tss/config* 및 Windows에서는 *instance\_profile\_path\instance\_name\db2tss\config*) 또는 config 디렉토리를 저장하는 추가 단계를 수행해야 합니다. 새 인스턴스가 작성된 후 config 디렉토리를 리스토어할 수 있습니다. 그러나 config 디렉토리 리스토어는 작성되는 새 인스턴스가 동일한 릴리스 및 FixPack 레벨인 경우에만 적용할 수 있습니다.
- Linux 및 UNIX 운영 체제에서 루트가 아닌 사용자가 설치한 인스턴스는 삭제할 수 없습니다. 이 DB2 인스턴스를 제거하기 위해 사용자가 사용할 수 있는 유일한 옵션은 db2\_deinstall -a를 실행하여 DB2의 루트가 아닌 사본을 설치 제거하는 것입니다.



- Linux 및 UNIX 운영 체제에서, login 명령 대신 su 명령을 사용하여 루트 사용자가 되는 경우 - 옵션과 함께 su 명령을 발행하여 프로세스 환경이 login 명령을 사용하여 시스템에 로그인한 것처럼 설정되도록 표시해야 합니다.
- Linux 및 UNIX 운영 체제에서 루트 사용자에게 대한 DB2 인스턴스 환경을 소스해서는 안됩니다. DB2 인스턴스 환경을 소스했을 때 db2idrop 명령을 실행하는 것은 지원되지 않습니다.
- AIX 6.1(또는 그 이상)에서 시스템 워크로드 파티션(WPAR) 전역 환경의 공유 DB2 사본으로부터 이 명령을 실행할 때 이 명령은 루트 사용자로서 실행되어야 합니다.
- Windows 운영 체제에서, IBM Tivoli Monitoring for Databases: DB2 Agent가 설치되고 DB2 인스턴스가 삭제되는 경우 Monitoring Agent for DB2가 삭제됩니다. Monitoring Agent for DB2 인스턴스에 대해 작성된 모든 파일, 서비스 및 레지스트리 항목도 삭제됩니다.



## 제 211 장 db2IdentifyType1 - 유형 1 인덱스 식별 도구

지정된 데이터베이스에 대해 유형 1 인덱스를 유형 2 인덱스로 변환하기 위해 사용할 수 있는 **ALLOW WRITE ACCESS** 및 **CONVERT** 절이 있는 적절한 **REORG INDEXES ALL** 명령을 식별하여 출력 파일로 생성합니다. 이 명령은 버전 9.7로 아직 업그레이드되지 않은 데이터베이스에 실행됩니다.

데이터베이스에 유형 1 인덱스가 없는 경우에는 이 명령을 실행하지 않아도 됩니다. 디폴트로 버전 8(또는 이상)에 작성된 모든 새 인덱스는 유형 2 인덱스이나 유형 1 인덱스가 이미 있는 테이블에 인덱스를 작성하거나 **DB2\_INDEX\_TYPE2** 레지스트리 변수가 해제로 설정된 경우는 예외입니다. DB2 UDB 버전 7(또는 이하)에 작성된 데이터베이스 및 아직 변환되지 않은 데이터베이스에도 유형 1 인덱스가 있을 수 있습니다.

유형 1 인덱스가 있는 테이블로의 첫 번째 액세스에 대한 성능 영향을 최소화하려면 db2IdentifyType1을 실행하고 이 인덱스를 변환한 후 데이터베이스를 버전 9.7로 업그레이드하십시오. 데이터베이스 업그레이드 중에 일부 테이블에 계속 유형 1 인덱스가 있는 경우 해당 인덱스는 유효하지 않은 것으로 표시됩니다. 업그레이드 후에는 해당 인덱스가 첫 번째 테이블 액세스에서 또는 데이터베이스를 재시작할 때(indexrec 구성 매개변수가 **RESTART**로 설정된 경우) 유형 2 인덱스로 변환됩니다. 인덱스가 재빌드될 때까지 테이블에 액세스할 수 없습니다. 기타 중요 성능 참고사항 및 파티션된 데이터베이스 환경에서 이 명령을 실행할 경우 미치는 영향에 대한 자세한 정보는 사용법 설명 섹션을 참고하십시오.

이 명령은 인덱스 변환을 수행하지 않습니다. 출력 파일을 생성한 후, 그 파일을 검토하고 필요에 따라 해당 콘텐츠를 편집한 후 다음 명령을 사용하여 출력 파일을 실행할 수 있습니다.

```
db2 -tvf filename
```

다음 디렉토리에서 db2IdentifyType1 명령을 실행하십시오.

- Linux 및 UNIX 운영 체제: DB2DIR/bin 디렉토리에서 실행하십시오(DB2DIR은 DB2 사본이 설치된 위치임).
- Windows 운영 체제: 제품 CD의 db2#Windows#Utilities 디렉토리에서 실행하십시오.

### 권한 부여

버전 9.7로 업그레이드되지 않은 데이터베이스의 경우, 다음 권한 부여 레벨 중 하나가 필요합니다.

- SYSADM

- DBADM

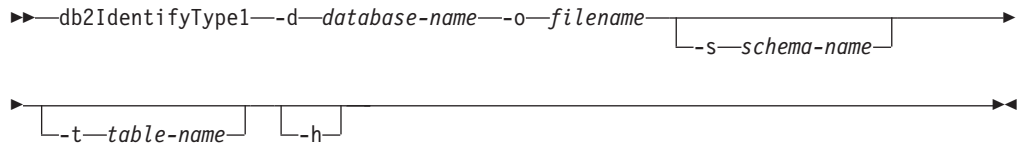
버전 9.7로 업그레이드된 데이터베이스의 경우, 다음 권한 부여 레벨이 필요합니다.

- DBADM

## 필수 연결

데이터베이스. 이 연결은 지정된 데이터베이스에 대한 연결을 자동으로 설정합니다.

## 명령 구문



## 명령 매개변수

### -d database-name

쿼리할 데이터베이스의 이름을 지정합니다.

### -o filename

REORG INDEX 명령을 쓸 파일의 경로 및 이름을 지정합니다. db2IdentifyType1.err 로그 파일은 동일한 경로에서 작성됩니다. db2IdentifyType1.err 로그 파일에는 문제점 해결 정보가 포함되며 db2IdentifyType1 명령이 실패하는 경우에만 작성됩니다. 전체 또는 상대 경로 이름을 사용하여 파일 이름을 규정할 수 있습니다. 이 이름은 246자를 초과할 수 없습니다. 동일한 이름을 사용하는 파일이 이미 존재하는 경우 해당 파일을 겹쳐씁니다.

### -s schema-name

쿼리할 테이블의 스키마(작성자 사용자 ID)를 지정합니다. 스키마를 지정하지 않으면 모든 스키마가 있는 테이블을 쿼리합니다. 한 번에 여러 스키마 이름을 지정할 수 없습니다. 스키마 이름 ID는 대문자를 사용하여 시스템 카탈로그 테이블에 표시되는 대로 지정해야 합니다. 구분된 스키마 이름 ID는 큰따옴표로 묶어야 합니다.

### -t table-name

쿼리할 테이블의 이름을 지정합니다. 테이블 이름을 지정하지 않으면 모든 테이블을 쿼리합니다. 한 번에 여러 테이블 이름을 지정할 수 없습니다. 테이블 이름 ID는 대문자를 사용하여 시스템 카탈로그 테이블에 표시되는 대로 지정해야 합니다. 구분된 테이블 이름 ID는 큰따옴표로 묶어야 합니다.

**-h** 도움말 정보를 표시합니다. 이 옵션을 지정하면 다음 옵션은 모두 무시됩니다.

## 사용 시 참고사항

데이터베이스에 유형 1 인덱스가 없거나, 데이터베이스에 유형 1 인덱스가 있는지 확실한 경우 이 도구를 사용하지 않아도 됩니다. 유형 1 인덱스가 많은 경우 특정 테이블을 식별하기 위해 이 명령을 사용하기 보다는 데이터베이스의 모든 테이블 또는 유형 1 인덱스가 있는 특정 테이블에 대해 CONVERT 옵션을 사용하여 REORG INDEXES를 실행하도록 선택할 수 있습니다. 인덱스가 이미 유형 2인 경우에는 CONVERT 옵션을 사용하여 REORG INDEXES를 실행할 수 없습니다.

이 명령은 DB2 UDB 버전 8 데이터베이스에서 유형 1 인덱스가 있는 유형이 지정된 테이블을 발견하지 않습니다. 버전 8에서 업그레이드하는 경우 『유형 1 인덱스를 유형 2 인덱스로 변환』 주제를 참조하여 유형이 지정된 테이블에 유형 1 인덱스가 있는지 여부를 판별하고 해당 인덱스를 변환하십시오.

테이블 수가 많은 데이터베이스의 경우 db2IdentifyType1 명령은 처리를 완료하는 데 시간이 오래 걸리므로 특정 서브세트만 한 번에 스캔하려면 -s 또는 -t 옵션을 사용하십시오.

파티션된 데이터베이스 환경의 성능 및 사용 고려사항: 파티션된 데이터베이스 환경에서 테이블이 상주하는 모든 데이터베이스 파티션에 유형 1 인덱스가 존재하지 않을 수 있습니다. 이것은 인덱스가 데이터베이스 파티션의 한 서브세트에서만 유형 2 인덱스로 변환된 경우 발생할 수 있습니다. 또한 모든 데이터베이스 파티션이 포함되지 않은 데이터베이스 파티션 그룹에 테이블이 상주할 수도 있습니다. 데이터베이스 버전 레벨에 따라 다음과 같이 하나 이상의 파티션에서 명령을 실행해야 합니다.

- 버전 9.1 또는 버전 9.5의 파티션된 데이터베이스 환경에서 모든 유형 1 인덱스를 검색하려면 이 인덱스가 모든 데이터베이스 파티션에 존재하는지 여부에 관계없이 db2IdentifyType1 명령을 하나의 데이터베이스 파티션에서만 실행해야 합니다. 최고의 성능을 위해서 카탈로그 데이터베이스 파티션에서 이 명령을 실행하십시오. 데이터베이스에 파티션이 많은 경우 데이터를 각 파티션에서 수집해야 하므로 이 명령을 완료하는데 오랜 시간이 걸릴 수 있습니다.
- 버전 8의 파티션된 데이터베이스 환경에서 db2IdentifyType1 명령을 사용하면 이 명령이 실행되는 파티션에 존재하는 유형 1 인덱스만 발견합니다. 모든 데이터베이스 파티션에 테이블이 존재하지 않거나 인덱스가 특정 파티션에서만 유형 2로 변환되었을 가능성이 있는 경우, 모든 데이터베이스 파티션에 이 명령을 실행하여 모든 유형 1 인덱스를 발견해야 합니다.



---

## 제 212 장 db2ilist - 인스턴스 목록

db2ilist 명령을 실행 중인 동일한 DB2 사본 위치에서 db2icrt 명령을 사용하여 작성되는 모든 인스턴스를 나열합니다.

Linux 및 UNIX 운영 체제에서, 이 유틸리티는 *DB2DIR/instance* 디렉토리에 있으며 *DB2DIR*은 DB2 사본이 설치된 인스턴스 디렉토리입니다. Windows 운영 체제에서 이 유틸리티는 *DB2PATH\bin* 디렉토리에 위치하며 *DB2PATH*는 DB2 데이터베이스 시스템의 현재 버전이 설치된 설치 위치입니다.

### 권한 부여

없음

### 명령 구문

▶▶—db2ilist—┐  
└─h─┘

### 명령 매개변수

**-h**     사용법 정보를 표시합니다.

### 사용 시 참고사항

- Linux 및 UNIX 운영 체제에서, login 명령 대신 su 명령을 사용하여 루트 사용자가 되는 경우 - 옵션과 함께 su 명령을 발행하여 프로세스 환경이 login 명령을 사용하여 시스템에 로그인한 것처럼 설정되도록 표시해야 합니다.
- Linux 및 UNIX 운영 체제에서 루트 사용자에게 대한 DB2 인스턴스 환경을 소스해서는 안됩니다. DB2 인스턴스 환경을 소스했을 때 db2ilist 명령을 실행하는 것은 지원되지 않습니다.
- AIX 6.1(또는 그 이상)에서 시스템 워크로드 파티션(WPAR) 전역 환경의 공유 DB2 사본으로부터 이 명령을 실행할 때 이 명령은 루트 사용자로서 실행되어야 합니다.





---

## 제 213 장 db2iupgrade - 인스턴스 업그레이드

db2iupgrade 명령은 이전 릴리스의 DB2 사본에서 현재 릴리스의 DB2 사본으로 인스턴스를 업그레이드합니다. db2iupgrade 명령을 실행 중인 DB2 사본이 업그레이드하려는 DB2 사본에서의 인스턴스 업그레이드를 지원해야 합니다.

Linux 및 UNIX 시스템에서 이 명령은 DB2DIR/instance 디렉토리에 있으며, 여기서 DB2DIR은 DB2 데이터베이스 시스템의 새 릴리스가 설치되는 설치 위치를 표시합니다. 이 명령은 루트가 아닌 설치에 대한 인스턴스 업그레이드를 지원하지 않습니다.

Windows 운영 체제에서 이 명령은 DB2PATH\bin 디렉토리에 위치하는데, DB2PATH는 DB2 사본이 설치된 위치입니다. 현재 위치에서 다른 위치로 인스턴스 프로파일을 이동하려면 /p 옵션을 사용하고 인스턴스 프로파일 경로를 지정하십시오. 그렇지 않으면 인스턴스 프로파일은 업그레이드 후 원래 위치에 남아 있습니다.

db2iupgrade 명령은 **-not1** 매개변수와 함께 db2ckupgrade.log를 로그 파일로 지정하여 db2ckupgrade 명령을 호출합니다. 인스턴스를 업그레이드하기 전에 로컬 데이터베이스의 업그레이드 준비 여부를 검증합니다. **-not1** 매개변수는 유형 1 인텍스 점검을 사용하지 않습니다. 로그 파일은 Linux 및 UNIX 운영 체제의 경우 인스턴스 홈 디렉토리, Windows 운영 체제의 경우 현재 디렉토리에 작성됩니다. db2ckupgrade 명령이 오류를 리턴하는 경우 인스턴스 업그레이드가 계속되지 않습니다.

파티션된 데이터베이스 환경에서 db2iupgrade 명령은 db2iupgrade 명령을 발행하는 데이터베이스 파티션에서 db2ckupgrade 명령을 실행합니다. 인스턴스를 업그레이드하기 전에 모든 데이터베이스 파티션을 업그레이드할 수 있는지 검증하십시오.

### 권한 부여

Linux 및 UNIX 시스템에서 루트 액세스 또는 Windows 운영 체제에서 로컬 관리자.

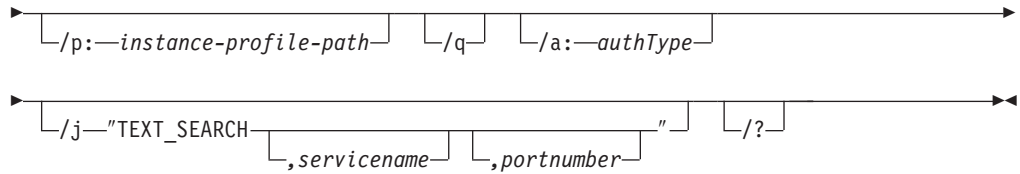
### 명령 구문

#### Linux 및 UNIX 시스템의 경우

```
▶▶ db2iupgrade [-d] [-k] [-a AuthType] [-u FencedID] InstName ▶▶
```

#### Windows 운영 체제의 경우

```
▶▶ db2iupgrade InstName /u: -username, password ▶▶
```



## 명령 매개변수

### Linux 및 UNIX 시스템의 경우

- d 디버그 모드를 켭니다. DB2 Support가 지시할 때만 이 옵션을 사용하십시오.
- k db2iupgrade 명령을 실행 중인 DB2 사본에서 지원되는 경우 사전 업그레이드 인스턴스 유형을 보존합니다. 이 매개변수를 지정하지 않으면 인스턴스 유형이 지원되는 디폴트 인스턴스 유형으로 업그레이드됩니다.

#### -a AuthType

인스턴스의 인증 유형(SERVER, CLIENT 또는 SERVER\_ENCRYPT)을 지정합니다. 디폴트는 SERVER입니다.

#### -u FencedID

분리 사용자 정의 기능 및 분리(fenced) 스토어드 프로시저가 실행할 사용자 ID의 이름을 지정합니다. 이 옵션은 DB2 클라이언트 인스턴스를 DB2 서버 인스턴스로 업그레이드할 때 필요합니다.

#### InstName

인스턴스 이름을 지정합니다.

### Windows 운영 체제의 경우

#### InstName

인스턴스 이름을 지정합니다.

#### /u:username,password

DB2 서비스의 어카운트 이름 및 암호를 지정합니다. 이 옵션은 파티션된 인스턴스를 업그레이드할 때 필수입니다.

#### /p:instance-profile-path

업그레이드된 인스턴스에 대한 새 인스턴스 프로파일 경로를 지정합니다.

#### /q 정숙 모드에서 db2iupgrade 명령을 발행합니다.

#### /a:authType

인스턴스의 인증 유형(SERVER, CLIENT 또는 SERVER\_ENCRYPT)을 지정합니다.

### **/j "TEXT\_SEARCH"**

서비스 이름 및 TCP/IP 포트 번호에 대해 생성된 디폴트값을 사용하여 DB2 텍스트 검색 서버를 구성합니다. 인스턴스 유형이 클라이언트이면 이 매개변수를 사용할 수 없습니다.

### **/j "TEXT\_SEARCH, servicename"**

제공되는 서비스 이름 및 자동으로 생성된 포트 번호를 사용하여 DB2 텍스트 검색 서버를 구성합니다. 서비스 파일에서 서비스 이름에 포트 번호가 지정된 경우 지정된 포트 번호를 사용합니다.

### **/j "TEXT\_SEARCH, servicename, portnumber"**

제공된 서비스 이름 및 포트 번호를 사용하여 DB2 텍스트 검색 서버를 구성합니다.

### **/j "TEXT\_SEARCH, portnumber"**

디폴트 서비스 이름 및 제공된 포트 번호를 사용하여 DB2 텍스트 검색 서버를 구성합니다. 유효한 포트 번호는 1024 - 65535 범위에 있어야 합니다.

**/?** db2iupgrade 명령의 사용법 정보를 표시합니다.

## **사용 시 참고사항**

### **Linux 및 UNIX 시스템의 경우**

- db2iupgrade 명령은 업그레이드 중인 버전에서 /usr/lib 및 /usr/include에 존재하는 모든 기호 링크를 제거합니다. 운영 체제의 라이브러리 환경 변수를 사용하여 찾는 대신 /usr/lib에서 직접 libdb2를 로드하는 응용프로그램이 있는 경우, db2iupgrade 명령을 실행한 후 응용프로그램이 제대로 실행하지 못할 수 있습니다.
- db2iupgrade 명령을 사용하여 DB2 데이터베이스 시스템의 이전 버전에서 현재 버전으로 DB2 인스턴스를 업그레이드하는 경우, 이전 DB2 데이터베이스 설치 경로에서 정의되는 DB2 전역 프로파일 변수가 새 설치 경로로 업그레이드되지 않습니다. 업그레이드될 인스턴스에 특정한 DB2 인스턴스 프로파일 변수는 인스턴스가 업그레이드된 후 이월됩니다.
- 로그인 명령 대신 su 명령을 사용하여 루트 사용자가 되는 경우, - 옵션과 함께 su 명령을 발행하여 프로세스 환경이 로그인 명령을 사용하여 시스템에 로그인한 경우처럼 설정되도록 해야 합니다.
- 루트 사용자에게 대한 DB2 인스턴스 환경을 소스해서는 안됩니다. DB2 인스턴스 환경을 소스했을 때 db2iupgrade 명령을 실행하는 것은 지원되지 않습니다.
- UNIX 및 Linux 운영 체제에서, IBM Tivoli Monitoring for Databases: DB2 Agent가 설치되고 DB2 인스턴스가 업그레이드되는 경우, 다음이 성립하면 Monitoring Agent for DB2 인스턴스도 작성됩니다.

- DB2 인스턴스 유형이 standalone, wse 또는 ese입니다.
- DB2 인스턴스가 버전 9.5(또는 그 이상)에 있습니다.

또한 인스턴스를 갱신하려는 DB2 사본에 대해 ITMA가 이미 설치되어 있어야 합니다. DB2DIR/itma 디렉토리에 위치하며, DB2DIR은 DB2 제품이 설치된 디렉토리를 표시합니다.

- AIX 6.1(또는 그 이상)에서 시스템 워크로드 파티션(WPAR) 전역 환경의 공유 DB2 사본으로부터 이 명령을 실행할 때 이 명령은 루트 사용자로서 실행되어야 합니다.

#### **Windows 운영 체제의 경우**

- Windows 운영 체제에서, IBM Tivoli Monitoring for Databases: DB2 Agent가 설치되고 DB2 사본 인스턴스가 갱신되는 경우 다음이 성립하면 Monitoring Agent for DB2 인스턴스도 작성됩니다.
  - DB2 인스턴스 유형이 standalone, wse 또는 ese입니다.
  - 디폴트 DB2 사본에 ITM 에이전트 구성요소가 설치되었습니다.
  - DB2 인스턴스가 버전 9.5(또는 그 이상)에 있습니다.
  - 기존 ITM for Databases 제품이 없습니다.

또한 Monitoring Agent for DB2 인스턴스 작성 후에 Monitoring Agent for DB2 인스턴스 파일, NT 서비스 및 레지스트리 항목도 작성됩니다.

## 제 214 장 db2inidb - 미러된 데이터베이스 초기화

분할 미러 환경에서 미러된 데이터베이스를 초기화합니다. 미러된 데이터베이스는 기본 데이터베이스의 클론으로서 초기화하거나 롤 포워드 보류 상태로 두거나 기본 데이터베이스를 리스토어하기 위한 백업 이미지로서 사용할 수 있습니다. 이 명령은 분할 미러 데이터베이스에 대해서만 실행할 수 있으며 분할 미러를 사용하려면 이 명령을 실행해야 합니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*

### 필수 연결

없음

### 명령 구문

```
▶▶ db2inidb database_alias AS {SNAPSHOT | STANDBY | MIRROR} RELOCATE USING configFile ▶▶
```

### 명령 매개변수

*database\_alias*

초기화될 데이터베이스의 별명을 지정합니다.

#### SNAPSHOT

미러된 데이터베이스가 기본 데이터베이스의 클론으로 초기화됨을 지정합니다.

#### STANDBY

데이터베이스가 롤 포워드 보류 상태가 됨을 지정합니다. 기본 데이터베이스의 새 로그는 폐치되며 대기 데이터베이스에 적용될 수 있습니다. 기본 데이터베이스 작동이 중단되면 대신 대기 데이터베이스를 사용할 수 있습니다.

#### MIRROR

미러된 데이터베이스가 기본 데이터베이스를 리스토어하는 데 사용할 수 있는 백업 이미지로서 사용되는 것으로 지정합니다.

#### RELOCATE USING *configFile*

데이터베이스 파일 위치가 데이터베이스를 스냅샷, 대기 또는 미러로서 초기화

하기 전에 지정된 *configFile*에 나열된 정보를 기반으로 다시 지정되는 것으로 지정합니다. *configFile* 형식은 1157 페이지의 제 245 장 『db2relocatedb - 데이터베이스 재배치』에서 설명합니다.

## 사용 시 참고사항

db2inidb *database\_alias* as mirror 명령을 실행하기 전에 db2 connect to *database-alias* 조사를 실행하지 마십시오. 초기화하기 전에 분할된 미러 데이터베이스에 대한 연결을 시도하면 롤 포워드 복구 중에 필요한 로그 파일이 지워집니다. 연결 시 데이터베이스를 일시중단할 때의 상태로 데이터를 다시 설정합니다. 데이터베이스가 일시중단된 시점과 일치하는 것으로 표시되면 DB2 데이터베이스는 응급 복구가 필요하지 않은 것으로 판단하여 나중에 사용할 수 있도록 로그를 비웁니다. 로그를 비운 경우 롤 포워드를 시도하면 SQL4970N 오류 메시지가 리턴됩니다.

파티션된 데이터베이스 환경의 경우 데이터베이스 파티션의 분할 미러를 사용하려면 모든 데이터베이스 파티션에서 db2inidb를 실행해야 합니다. db2inidb는 db2\_all 명령을 사용하여 모든 데이터베이스 파티션에서 동시에 실행할 수 있습니다.

그러나 RELOCATE USING 옵션을 사용하는 경우에는 db2\_all 명령을 사용하여 모든 파티션에서 동시에 db2inidb를 실행할 수 있습니다. 각 파티션마다 변경될 데이터베이스 파티션의 NODENUM 값을 포함하는 개별 구성 파일을 제공해야 합니다. 예를 들어, 데이터베이스 이름이 변경되는 경우 모든 데이터베이스 파티션이 영향을 받으며 각 데이터베이스 파티션의 개별 구성 파일과 함께 db2relocatedb 명령을 실행해야 합니다. 단일 데이터베이스 파티션에 속하는 컨테이너를 이동하는 경우 db2relocatedb 명령은 해당 데이터베이스 파티션에서 한 번만 실행해야 합니다.

RELOCATE USING *configFile* 매개변수가 지정되고 데이터베이스 위치가 변경되면 지정된 *configFile*이 데이터베이스 디렉토리에 복사되며 이름이 db2path.cfg로 바뀝니다. 이 파일은 후속 응급 복구 또는 롤 포워드 복구 중에 로그 파일이 처리됨에 따라 컨테이너 경로의 이름을 바꾸는 데 사용됩니다.

클론 데이터베이스를 초기화하는 경우, 지정된 *configFile*은 응급 복구가 완료된 후 데이터베이스 디렉토리에서 자동으로 제거됩니다.

대기 데이터베이스 또는 미러된 데이터베이스를 초기화하는 경우, 지정된 *configFile*은 롤 포워드 복구가 완료 또는 취소된 후 데이터베이스 디렉토리에서 자동으로 제거됩니다. db2inidb가 실행된 후 db2path.cfg 파일에 새 컨테이너 경로를 추가할 수 있습니다. 이는 원래 데이터베이스에서 CREATE 또는 ALTER TABLESPACE 조작이 완료되고 대기 데이터베이스에 다른 경로를 사용해야 하는 경우 필요합니다.

## 제 215 장 db2inspf - 검사 결과 형식화

이 유틸리티는 INSPECT CHECK 결과 데이터를 ASCII 형식으로 형식화합니다. 이 유틸리티를 사용하면 검사 세부사항을 볼 수 있습니다. db2inspf 유틸리티를 사용하면 테이블 또는 테이블 스페이스를 형식화할 수 있으며 오류, 경고 및 요약은 개별적으로 또는 조합으로 지정할 수 있습니다.

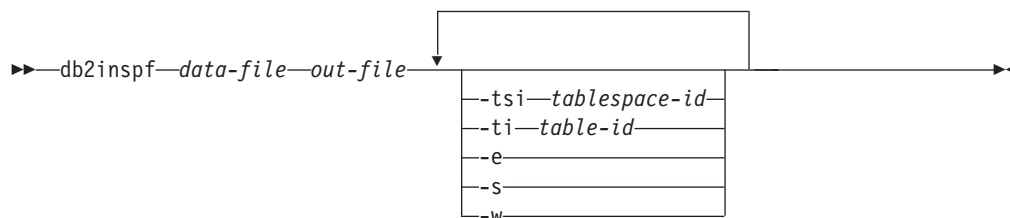
### 권한 부여

이 유틸리티는 누구나 액세스할 수 있지만 이 유틸리티를 실행하려면 사용자에게 결과 파일에 대한 읽기 권한이 있어야 합니다.

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

#### *data-file*

형식화할 형식화되지 않은 검사 결과 파일.

#### *out-file*

형식화된 출력의 출력 파일.

#### **-tsi** *tablespace-id*

테이블 스페이스 ID. 이 테이블 스페이스의 테이블에 대해서만 형식화합니다.

#### **-ti** *table-id*

테이블 ID. 이 ID를 사용하는 테이블에 대해서만 형식화하며 테이블 스페이스 ID 또한 제공되어야 합니다.

**-e** 오류만 형식화합니다.

**-s** 요약만.

**-w** 경고만.

**예:**

데이터 파일 tbschk.log에서 모든 오류, 경고 및 요약을 형식화하려면 다음을 실행하십시오.

```
db2inspf tbschk.log tbschk_esw.txt -e -s -w
```



## 제 216 장 db2iprune - 설치 이미지 크기 줄이기 명령

db2iprune 명령은 설치 전에 DB2 제품 설치 이미지의 크기를 줄일 수 있습니다.

이 도구는 DB2의 대규모 전개와 응용프로그램에 DB2를 임베딩하는 데 유용합니다. 입력 파일, 또는 제거할 수 있는 제품, 구성요소 및 언어의 전체 목록을 포함하는 .prn 파일을 사용하면 설치 이미지에서 제거할 대상을 지정할 수 있습니다. db2iprune 명령은 입력 파일을 호출하고 해당 구성요소 및 언어와 연관된 파일을 제거합니다. 결과는 일반 DB2 설치 메소드를 사용하여 설치할 수 있는 작은 새 DB2 설치 이미지입니다.

모든 제품을 프룬(prune)할 수는 없습니다. 최소한 하나의 제품이 결과 이미지의 일부여야 합니다.

### 권한 부여

없음

### 필수 연결

없음

### 명령 구문

```
db2iprune -r input_file_path -p root_directory_path  
-o destination_directory_path -c  
-t trace_file  
-l log_filename -h  
-?
```

### 명령 매개변수

#### -r input\_file\_path

사용될 입력 파일의 전체 경로를 지정합니다. 입력 파일 또는 .prn 파일에는 제거할 수 있는 구성요소의 전체 목록이 포함되며 설치 이미지에서 제거할 제품, 구성요소 및 언어를 표시하는 데 사용됩니다.

#### -p root\_directory\_path

(Windows 운영 체제에만 해당). 소스 설치 이미지의 루트 디렉토리 전체 경로를 지정합니다. 이 디렉토리에는 setup이 포함되며 DB2 설치 DVD의 루트 디렉토리입니다.

**-o** *destination\_directory\_path*

DB2 프론 이미지가 복사될 전체 경로를 지정합니다. 이 디렉토리에 대한 쓰기 액세스 권한이 있는지 확인하십시오.

**-c** 소스 설치 이미지를 직접 프룬(prune)하려는 것으로 지정합니다. 소스 설치 이미지 디렉토리가 쓰기 가능한지 확인합니다.

**-l** 오류 로깅을 사용합니다. Linux 및 UNIX 운영 체제에서 -l 옵션이 지정되지 않는 경우 디폴트 로그 파일 이름은 tmpdir/db2iprune\_username.log입니다. Windows 운영 체제의 경우 로그 파일 db2iprune.log는 대상 디렉토리에 기록됩니다.

**-t** *trace\_file*

(Linux 및 UNIX 운영 체제에만 해당). 디버그 모드를 활성화합니다. 디버그 정보는 지정된 파일 이름에 작성됩니다.

**-? | -h**

사용법 정보를 표시합니다.

**예:**

Windows 운영 체제에서, 입력 파일이 c:\db2client.prn에 있고 DB2 setup.exe 파일이 d:\#에 있으며 프룬(prune)된 IBM Data Server Client 이미지를 e:\#compact\_client 디렉토리에 복사하려는 경우 IBM Data Server Client 이미지를 프룬(prune)하려면 명령 프롬프트에서 다음 명령을 입력합니다.

```
db2iprune.exe -r c:\db2client.prn -p d:\# -o e:\#compact_client
```

Linux 및 UNIX 운영 체제에서, 입력 파일이 /tmp/db2client.prn에 있고 DB2 소스 파일이 /mnt/cdrom에 있으며 프룬(prune)된 IBM Data Server Client 이미지를 /compact\_client 디렉토리에 복사하려는 경우 IBM Data Server Client 이미지를 프룬(prune)하려면 명령 프롬프트에서 다음 명령을 입력합니다.

```
db2iprune -r /tmp/db2client.prn /mnt/cdrom -o /compact_client
```

**사용 시 참고사항**

db2iprune 명령과 샘플 입력 파일은 설치 DVD에 제공됩니다.

**Windows 운영 체제의 경우**

```
dvd_drive:\db2\Windows\utilities\db2iprune
```

**Linux 및 UNIX 운영 체제의 경우**

```
product_image/db2/platform/utilities/db2iprune
```

---

## 제 217 장 db2isetaup - 인스턴스 작성 인터페이스 시작

인스턴스를 작성하고 기존 인스턴스에 새 기능을 구성하는 그래픽 도구인 DB2 인스턴스 설정 마법사를 시작합니다.

### 권한 부여

루트 설치의 경우 명령이 실행된 시스템에서 루트 권한이 필요합니다. 루트 서버가 아닌 설치의 경우 루트 서버가 아닌 설치를 소유하는 사용자 ID로 로그인해야 합니다.

### 필수 연결

없음

### 명령 구문

```
db2isetaup [-i language-code] [-l logfile] [-t tracefile]
            [-r response_file] [-? | -h]
```

### 명령 매개변수

#### -i language-code

설치를 실행할 선호 언어의 2자 코드. 이 매개변수가 지정되지 않는 경우 디폴트값은 현재 사용자의 로케일입니다.

#### -l logfile

지정된 파일 이름에 로그를 작성합니다. 루트 설치의 경우 경로 및 파일 이름의 디폴트값은 /tmp/db2isetaup.log입니다. 루트 서버가 아닌 설치의 경우 디폴트 로그 파일은 /tmp/db2isetaup\_userID.log입니다. 여기서 userID는 루트 서버가 아닌 설치를 소유하는 사용자 ID를 나타냅니다.

#### -t tracefile

tracefile로 지정된 추적 파일의 전체 경로 및 이름.

#### -r response\_file

사용할 응답 파일의 전체 경로 및 파일 이름. 응답 파일에는 키워드 FILE이 포함되어야 합니다.

#### -? | -h

사용법 정보를 출력합니다.

## 사용 시 참고사항

1. 이 인스턴스 설정 마법사는 DB2 설정 마법사가 제공하는 기능의 서브세트를 제공합니다. DB2 설정 마법사는 설치 미디어에서 실행되며 DB2 구성요소를 설치하고 DAS 작성/구성과 같은 시스템 설정 태스크를 수행하며 인스턴스를 설정하는 데 사용됩니다. DB2 인스턴스 설정 마법사는 인스턴스 설정과 관련된 기능만 제공합니다.
2. 이 명령에 실행할 수 있는 파일은 DB2DIR/instance 디렉토리에 있습니다. 해당 파일은 최소 설치가 아닌 일반 설치에서 사용할 수 있습니다.
3. db2isetup은 지원되는 모든 Linux 및 UNIX 시스템에서 실행됩니다.
4. AIX 6.1(또는 그 이상)에서 시스템 워크로드 파티션(WPAR) 전역 환경의 공유 DB2 사본으로부터 이 명령을 실행할 때 이 명령은 루트 사용자로서 실행되어야 합니다.

## 제 218 장 db2iupdt - 인스턴스 갱신

이 명령은 새 DB2 데이터베이스 제품이나 기능이 설치된 DB2 사본에서 실행하거나, 인스턴스와 연관된 DB2 사본과 동일한 버전의 DB2 사본에서 실행하거나 인스턴스 유형을 상위 레벨의 인스턴스 레벨로 업그레이드하도록 인스턴스를 갱신합니다.

db2iupdt 명령은 동일하거나 다른 DB2 사본과 연관된 동일한 버전의 인스턴스에 대해 발행할 수 있습니다. 모든 경우에 db2iupdt 명령을 발행한 곳과 동일한 DB2 사본에 위치하는 코드에 대해 실행하도록 인스턴스를 갱신합니다. 다음 경우에 이 명령을 발행해야 합니다.

- FixPack을 설치하고 자동 인스턴스 갱신에 실패하는 경우.
- DB2 인스턴스에 연관되는 DB2 사본에 새 DB2 데이터베이스 제품이나 기능을 설치하는 경우.
- 동일한 버전의 DB2 데이터베이스 제품의 한 DB2 사본에서 다른 DB2 사본으로 DB2 인스턴스를 갱신하려는 경우.

FixPack이 Linux 및 UNIX 운영 체제에 설치된 후 db2iupdt 명령이 자동으로 실행됩니다.

db2iupdt를 사용하여 인스턴스를 갱신하려면 먼저 인스턴스에 대해 실행 중인 모든 프로세스를 중지해야 합니다.

### 권한 부여

UNIX 및 Linux 운영 체제에서 Root 액세스 및 Windows 운영 체제에서 로컬 관리자.

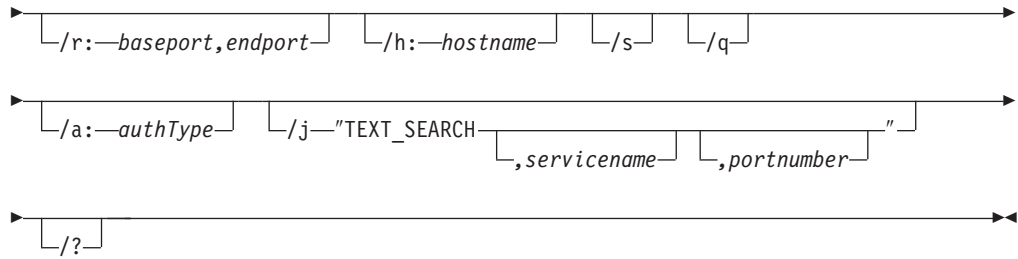
### 명령 구문

UNIX 및 Linux 운영 체제의 경우

```
▶▶ db2iupdt [-h] [-d] [-k] [-D] [-s] [-a AuthType]
             [-u FencedID] [-e InstName]
```

Windows 운영 체제의 경우

```
▶▶ db2iupdt -InstName /u:—username,password /p:—instance-profile-path
```



## 명령 매개변수

### UNIX 및 Linux 운영 체제의 경우

#### **-h | -?**

사용법 정보를 표시합니다.

#### **-d** 디버그 모드를 켭니다.

#### **-k** 갱신 중에 현재 인스턴스 유형을 유지합니다.

#### **-D** 한 경로의 상위 코드 레벨에서 다른 경로에 설치된 낮은 코드 레벨로 인스턴스를 이동합니다.

#### **-s** 기존 SPM 로그 디렉토리를 무시합니다.

#### **-a AuthType**

인스턴스의 인증 유형(SERVER, SERVER\_ENCRYPT 또는 CLIENT)을 지정합니다. 디폴트는 SERVER입니다.

#### **-u Fenced ID**

분리 사용자 정의 기능 및 분리(fenced) 스토어드 프로시저가 실행할 사용자 ID의 이름을 지정합니다. 이 옵션은 클라이언트 인스턴스에서 비클라이언트 인스턴스 유형으로 인스턴스를 변환할 때만 필요합니다. 현재 인스턴스 유형을 판별하려면 GET DBM CFG 명령의 출력에서 노드 유형 매개변수를 참조하십시오. 인스턴스가 이미 비클라이언트 인스턴스이거나 인스턴스가 클라이언트 인스턴스이고 클라이언트 인스턴스로서 유지되는 경우(예를 들어, **-k** 옵션을 사용하여) **-u** 옵션은 필요하지 않습니다. **-u** 옵션은 기존 인스턴스에 대한 분리 사용자를 변경할 수 없습니다.

#### *InstName*

인스턴스 이름을 지정합니다.

#### **-e** 모든 인스턴스를 갱신합니다.

### Windows 운영 체제의 경우

#### *InstName*

인스턴스 이름을 지정합니다.

**/u:username,password**

DB2 서비스의 어카운트 이름 및 암호를 지정합니다.

**/p:instance-profile-path**

갱신된 인스턴스에 대한 새 인스턴스 프로파일 경로를 지정합니다.

**/r:baseport,endport**

MPP 모드에서 실행할 때 파티션된 데이터베이스 인스턴스가 사용할 TCP/IP 포트의 범위를 지정합니다. 이 옵션이 지정되면 로컬 시스템의 서비스 파일이 다음 항목으로 갱신됩니다.

```
DB2_InstName      baseport/tcp
DB2_InstName_END  endport/tcp
```

**/h:hostname**

현재 머신에 대해 둘 이상의 TCP/IP 호스트 이름이 있는 경우 디폴트 TCP/IP 호스트 이름을 겹쳐줍니다.

**/s** 인스턴스를 파티션된 인스턴스로 갱신합니다.

**/q** 정숙 모드에서 db2iupdt 명령을 발행합니다.

**/a:authType**

인스턴스의 *authType*, 인증 유형(SERVER, CLIENT 또는 SERVER\_ENCRYPT)을 지정합니다.

**/j "TEXT\_SEARCH"**

서비스 이름 및 TCP/IP 포트 번호에 대해 생성된 디폴트값을 사용하여 DB2 텍스트 검색 서버를 구성합니다. 인스턴스 유형이 클라이언트이면 이 매개변수를 사용할 수 없습니다.

**/j "TEXT\_SEARCH, servicename"**

제공되는 서비스 이름 및 자동으로 생성된 포트 번호를 사용하여 DB2 텍스트 검색 서버를 구성합니다. 서비스 파일에서 서비스 이름에 포트 번호가 지정된 경우 지정된 포트 번호를 사용합니다.

**/j "TEXT\_SEARCH, servicename, portnumber"**

제공된 서비스 이름 및 포트 번호를 사용하여 DB2 텍스트 검색 서버를 구성합니다.

**/j "TEXT\_SEARCH, portnumber"**

디폴트 서비스 이름 및 제공된 포트 번호를 사용하여 DB2 텍스트 검색 서버를 구성합니다. 유효한 포트 번호는 1024 - 65535 범위에 있어야 합니다.

**/?** db2iupdt 명령의 사용법 정보를 표시합니다.

## 예(UNIX 및 Linux)

- 인스턴스 db2inst2가 DB2DIR1에 설치된 DB2 데이터베이스 제품의 DB2 사본에 연관됩니다. DB2DIR1에 설치된 것과 동일한 버전의 DB2 데이터베이스 제품에 대한 DB2DIR2에 동일한 컴퓨터의 DB2 데이터베이스 제품의 다른 DB2 사본이 있습니다. DB2DIR1에 설치된 DB2 사본에서 DB2DIR2에 설치된 DB2 사본으로 실행할 인스턴스를 갱신하려면 다음 명령을 발행하십시오.

```
DB2DIR2/instance/db2iupdt db2inst2
```

DB2DIR2에 설치된 DB2 사본이 DB2DIR1에 설치된 DB2 사본보다 낮은 경우 다음을 발행하십시오.

```
DB2DIR2/instance/db2iupdt -D db2inst2
```

## 사용 시 참고사항

지원되는 모든 운영 체제의 경우

- db2iupdt 명령을 사용하여 DB2 인스턴스를 동일한 버전의 한 DB2 데이터베이스 제품의 한 DB2 사본에서 다른 DB2 사본으로 갱신하는 경우, 이전 DB2 사본 설치에서 정의된 DB2 전역 프로파일 변수는 새 설치 위치로 갱신되지 않습니다. 인스턴스에 특정한 DB2 인스턴스 프로파일 변수가 인스턴스가 갱신된 후 이월됩니다.

UNIX 및 Linux 운영 체제의 경우

- db2iupdt 명령은 DB2DIR/instance 디렉토리에 위치하며, DB2DIR은 DB2 데이터베이스 제품의 현재 버전이 설치된 위치입니다.
- 루트가 아닌 인스턴스를 갱신하려는 경우 db2nrupdt 루트가 아닌 사용자가 설치한 인스턴스 갱신 명령을 참조하십시오. db2iupdt는 루트가 아닌 인스턴스의 갱신을 지원하지 않습니다.
- login 명령 대신 su 명령을 사용하여 루트 사용자가 되는 경우 su 명령을 - 옵션과 함께 발행하여 프로세스 환경이 login 명령을 사용하여 시스템에 로그인한 것처럼 설정되도록 표시해야 합니다.
- 루트 사용자에게 대한 DB2 인스턴스 환경을 소스해서는 안됩니다. DB2 인스턴스 환경을 소스했을 때 db2iupdt 명령을 실행하는 것은 지원되지 않습니다.
- UNIX 및 Linux 운영 체제에서, IBM Tivoli Monitoring for Databases: DB2 Agent가 설치되고 DB2 인스턴스가 갱신되는 경우, 다음이 성립하면 Monitoring Agent for DB2 인스턴스도 작성됩니다.
  - DB2 인스턴스 유형이 standalone, wse 또는 ese입니다.
  - DB2 인스턴스가 버전 9.5(또는 그 이상)에 있습니다.



또한 인스턴스를 갱신하려는 DB2 사본에 대해 ITMA가 이미 설치되어 있어야 합니다. DB2DIR/itma 디렉토리에 위치하며, DB2DIR은 DB2 제품이 설치된 디렉토리를 표시합니다.

- AIX 6.1(또는 그 이상)에서 시스템 워크로드 파티션(WPAR) 전역 환경의 공유 DB2 사본으로부터 이 명령을 실행할 때 이 명령은 루트 사용자로서 실행되어야 합니다.

#### Windows 운영 체제의 경우

- db2iupdt 명령은 *DB2PATH*\bin 디렉토리에 위치하며, *DB2PATH*는 DB2 데이터베이스 제품의 현재 버전이 설치된 위치입니다.
- 인스턴스는 db2iupdt 명령을 발행하는 DB2 사본으로 갱신됩니다. 그러나 현재 위치에서 다른 위치로 인스턴스 프로파일을 이동하려면 */p* 옵션을 사용하고 인스턴스 프로파일 경로를 지정하십시오. 그렇지 않으면 인스턴스 프로파일은 인스턴스 갱신 후 원래 위치에 남아 있습니다. 대신 db2iupgrade 명령을 사용하여 이전 릴리스에서 현재 릴리스로 업그레이드하십시오.
- Windows 운영 체제에서, IBM Tivoli Monitoring for Databases: DB2 Agent가 설치되고 DB2 사본 인스턴스가 갱신되는 경우 다음이 성립하면 Monitoring Agent for DB2 인스턴스도 작성됩니다.
  - DB2 인스턴스 유형이 standalone, wse 또는 ese입니다.
  - 디폴트 DB2 사본에 ITM 에이전트 구성요소가 설치되었습니다.
  - DB2 인스턴스가 버전 9.5(또는 그 이상)에 있습니다.
  - 기존 ITM for Databases 제품이 없습니다.

또한 Monitoring Agent for DB2 인스턴스 작성 후에 Monitoring Agent for DB2 인스턴스 파일, NT 서비스 및 레지스트리 항목도 작성됩니다.



---

## 제 219 장 db2jdbcbind - DB2 JDBC 패키지 바인더

이 유틸리티는 JDBC 패키지를 DB2 데이터베이스에 바인드하거나 리바인드하기 위해 사용됩니다. DB2 버전 8 데이터베이스에는 이미 JDBC 패키지가 사전에 설치되어 있으므로, 이 명령은 보통 이전 서버에 대해서만 필요합니다. JDBC 및 CLI는 동일한 패키지를 공유합니다. CLI 패키지가 이미 데이터베이스에 바인드된 경우 이 유틸리티를 실행할 필요가 없습니다. 그 반대도 마찬가지입니다.

### 권한 부여

다음 중 하나가 필요합니다.

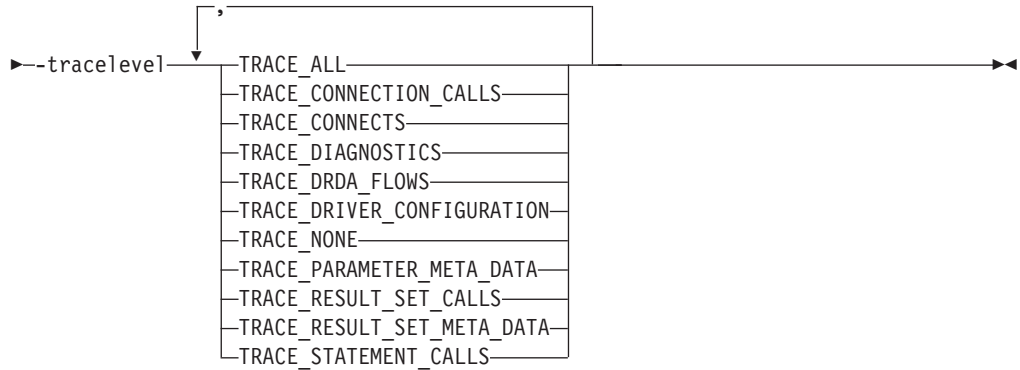
- *dbadm*
- 패키지가 없는 경우 BINDADD 특권, 그리고 다음 중 하나:
  - 패키지의 스키마 이름이 존재하지 않는 경우 데이터베이스에 대한 IMPLICIT\_SCHEMA 권한
  - 패키지의 스키마 이름이 존재하는 경우 스키마에 대한 CREATEIN 특권
- 패키지가 존재하는 경우 스키마에 대한 ALTERIN 특권
- 패키지가 존재하는 경우 패키지에 대한 BIND 특권

### 필수 연결

이 명령은 데이터베이스 연결을 설정합니다.

### 명령 구문

```
▶▶ db2jdbcbind [help] -url jdbc:db2://servername:portnumber/dbname →
▶ --user username --password password [collection collection ID] →
▶ [-size number of packages] →
```



## 명령 매개변수

**-help** 도움말 정보를 표시합니다. 다른 모든 옵션은 무시됩니다.

**-url jdbc:db2://servername:portnumber/dbname**

데이터베이스 연결을 설정하기 위한 JDBC URL을 지정합니다. DB2 JDBC 유형 4 드라이버가 연결 설정에 사용됩니다.

**-user username**

데이터베이스에 연결할 때 사용되는 이름을 지정합니다.

**-password password**

사용자 이름의 암호를 지정합니다.

**-collection collection ID**

패키지에 사용할 콜렉션 ID(CURRENT PACKAGESET). 디폴트는 NULLID입니다. 설정된 패키지의 여러 인스턴스를 작성하려면 이 옵션을 사용하십시오. 이 옵션은 연결 또는 데이터 소스 등록 정보인 currentPackageSet와 함께 사용해야 합니다.

**-size number of packages**

DB2 트랜잭션 분리 레벨 및 유지 가능성 설정마다 바인드할 내부 패키지 수. 디폴트는 3입니다. 네 개의 DB2 분리 레벨과 두 개의 커서 유지 가능성 설정이 있으므로, 이 옵션에 지정된 대로 바인드된 동적 패키지 수만큼  $4 \times 2 = 8$  배가 됩니다. 또한 단일 정적 패키지는 항상 내부용으로 바인드됩니다.

**-tracelevel**

문제점 해결을 위해서만 필요한 추적 레벨을 식별합니다.

---

## 제 220 장 db2ldcfg - LDAP 환경 구성

IBM LDAP 클라이언트를 사용하여 LDAP 환경의 현재 로그인 사용자에게 대한 LDAP(Lightweight Directory Access Protocol) 사용자 식별 이름(DN) 및 암호를 구성합니다.

### 권한 부여

없음

### 필수 연결

없음

### 명령 구문

```
▶▶ db2ldcfg -u userDN -w password
```

### 명령 매개변수

#### **-u** *userDN*

LDAP 디렉토리에 액세스할 때 사용될 LDAP 사용자의 식별 이름을 지정합니다. 아래 예에 표시되는 것처럼 식별 이름은 사용자 ID(예: jdoe), 도메인 및 소속 이름과 접미부(예: com 또는 org)의 여러 파트를 갖습니다.

#### **-w** *password*

암호를 지정합니다.

**-r** 머신 환경에서 사용자의 DN 및 암호를 제거합니다.

예:

```
db2ldcfg -u "uid=jdoe,dc=mydomain,dc=myorg,dc=com" -w password
```

### 사용 시 참고사항

IBM LDAP 클라이언트를 사용하는 LDAP 환경에서, 현재 로그인 사용자에게 대한 디폴트 LDAP 사용자의 DN 및 암호를 구성할 수 있습니다. 구성된 후에는 LDAP 사용자의 DN 및 암호가 사용자 환경에서 사용되고 DB2가 LDAP 디렉토리에 액세스할 때 마다 사용됩니다. 이것은 LDAP 명령이나 API를 발행할 때 LDAP 사용자의 DN 및 암호를 지정할 필요성을 제거합니다. 그러나 LDAP 사용자의 DN 및 암호가 지정되는 경우, 명령이나 API가 발행될 때 디폴트 설정이 겹쳐집니다.

이 명령은 IBM LDAP 클라이언트를 사용할 때만 실행할 수 있습니다. Microsoft LDAP 클라이언트에서 현재 로그인 사용자의 증명서가 사용됩니다.

---

## 제 221 장 db2level - DB2 서비스 레벨 표시

설치된 DB2 제품의 현재 버전 및 서비스 레벨을 표시합니다. 이 명령은 디폴트로 콘솔에 표시됩니다.

### 권한 부여

없음

### 필요한 연결

없음

### 명령 구문

```
▶▶—db2level—————▶▶
```

### 명령 매개변수

없음

### 예:

Windows 제품의 경우 db2level 명령은 DB2 사본 이름을 표시합니다. 예를 들면, 다음과 같습니다.

```
DB21085I 인스턴스 "DB2"은(는) "32"비트와 레벨 ID "01010107"의 DB2 코드 릴리스
"SQL09010"을(를) 사용합니다.
레벨 ID "01010107".
정보용 토큰은 "DB2 v9.1.0.189", "n060119" 및 FixPack "0"입니다.
제품이 DB2 사본 이름 "db2build"(으)로 "c:\SQLLIB"에 설치되었습니다.
```

Linux 및 UNIX 기반 운영 체제의 경우 db2level 명령은 DB2 사본 이름을 표시하지 않습니다. 예를 들면, 다음과 같습니다.

```
DB21085I 인스턴스 "wqzhuang"은(는) "64"비트와 레벨 ID "01010107"의 DB2 코드 릴리스
"SQL09010"을(를) 사용합니다.
정보용 토큰은 "DB2 v9.1.0.0", "n060124" 및 FixPack "0"입니다.
제품이 "/home/wqzhuang/sql1ib"에 설치되었습니다.
```

### 사용 시 참고사항

이 명령의 출력 명령에는 릴리스, 레벨 및 다양한 정보 토큰이 포함됩니다.





## 제 222 장 db2licm - 라이선스 관리 도구

제어 센터가 없는 경우에 기본 라이선스 기능을 수행합니다. 로컬 시스템에 설치된 라이선스 및 규정을 추가, 제거, 나열 및 수정합니다.

주: 프로세서 값 단위(PVU) 라이선스 구조에서, 각 프로세서 코어에 특정한 값 단위 수가 지정됩니다. 소프트웨어 프로그램이 전개되는 각 프로세서 코어에 대한 프로세서 값 단위의 총 수를 획득해야 합니다. IBM은 계속해서 칩의 각 프로세서 코어가 될 프로세서를 정의합니다. 예를 들어 이중 코어 칩에는 두 개의 프로세서 코어가 있습니다.

각 소프트웨어 프로그램은 가치 단위당 고유한 가격을 갖습니다. 개별 소프트웨어 프로그램을 전개하는 전체 비용을 판별하려면 가치 단위당 프로그램 가격에 필요한 프로세서 값 단위의 총 수를 곱합니다.

### 권한 부여

Windows 운영 체제에서,

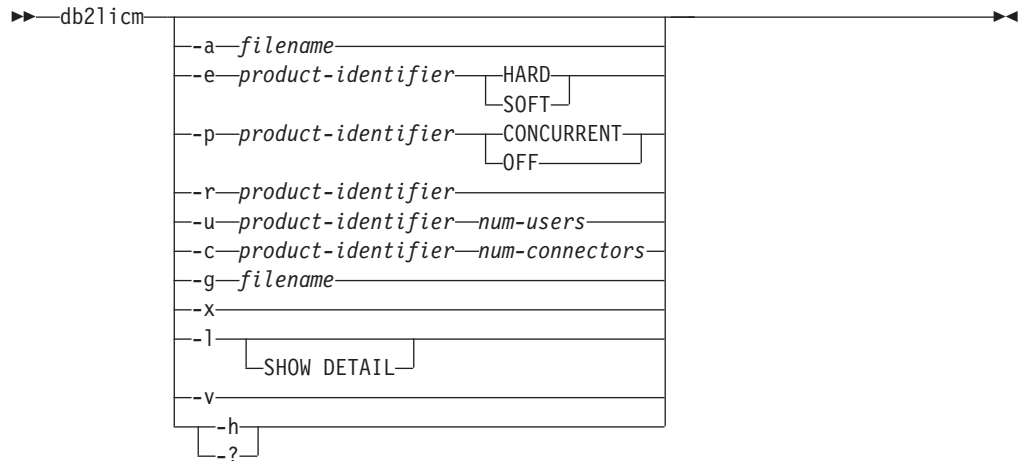
- **-a**, **-r** 또는 **-x** 명령 매개변수를 사용하려면 로컬 관리자 또는 권한이 있는 사용자 그룹에 속해야 합니다.
- **-c**, **-e**, **-p**, **-r** 또는 **-u** 명령 매개변수를 사용하려면 SYSADM 권한이 필요합니다.

UNIX 및 Linux 운영 체제에서는 권한이 필요하지 않습니다.

### 필수 연결

없음

### 명령 구문



## 명령 매개변수

### -a *filename*

제품에 대한 라이선스를 추가합니다. 유효한 라이선스 정보가 들어있는 파일 이름을 지정하십시오. 이 이름은 사용 허가된 제품 CD에서 또는 IBM 담당자나 권한 부여된 딜러에 문의하여 알 수 있습니다.

### -c *product-identifier num-connectors*

구입한 커넥터 사용권 수를 갱신합니다. 제품 ID 및 커넥터 사용권 수를 지정하십시오.

### -e *product-identifier*

시스템의 강제 규정을 갱신합니다. 가능한 값은 다음과 같습니다.

#### **HARD**

사용 허가가 없는 요청이 허용되지 않도록 지정합니다.

**SOFT** 사용 허가가 없는 요청이 로그되지만 제한되지는 않음을 지정합니다.

### -g *filename*

준수 보고서를 생성합니다. 출력이 저장될 파일 이름을 지정하십시오.

### -h | -?

도움말 정보를 표시합니다. 이 옵션이 지정되면 다른 모든 옵션은 무시되고 도움말 정보만 표시됩니다.

-l 제품 ID를 포함하여 사용 가능한 라이선스 정보와 함께 모든 제품을 나열합니다.

#### **SHOW DETAIL**

사용 허가된 기능(있는 경우)에 관한 상세한 정보를 보려면 지정하십시오.

### -p *product-identifier*

시스템에서 사용할 라이선스 규정 유형을 갱신합니다.

#### **CONCURRENT**

동시 사용자 규정의 경우에 지정하십시오.

**OFF** 모든 규정을 끄려면 지정하십시오.

### -r *product-identifier*

제품에 대한 라이선스를 제거합니다. 특정 제품에 대한 제품 ID를 가져오려면 -l 옵션과 함께 명령을 호출하십시오.

### -u *product-identifier num-users*

고객이 구입한 사용자 라이선스 수를 갱신합니다. 제품 ID 및 사용자 수를 지정하십시오.

-v 버전 정보를 표시합니다.

-x      라이선스 준수 보고서를 위한 라이선스 준수 정보를 재설정합니다.

예:

```
db2licm -a db2ese.lic
db2licm -p db2consv concurrent
db2licm -r db2ese
db2licm -u db2wse 10
db2licm -e db2ese SOFT
```

제품 ID를 포함하여 사용 가능한 라이선스 정보와 함께 모든 제품을 나열하는 출력 예  
:

```
C:\Program Files\IBM\SQLLIB\BIN>db2licm -l
제품 이름:                   "DB2 Enterprise Server Edition"
라이선스 유형:               "시험"
만기 날짜:                   "08/31/2009"
제품 ID:                     "db2ese"
버전 정보:                   "9.7"
```



---

## 제 223 장 db2listvolumes - 모든 디스크 볼륨의 GUID 표시

Windows 운영 체제에 지정된 모든 디스크 볼륨의 GUID를 표시합니다. 이 명령은 도구가 실행된 디렉토리에서 두 개 파일을 작성합니다. volumes.xml 파일에는 XML 사용 브라우저에서 쉽게 볼 수 있도록 XML로 인코딩된 각 디스크 볼륨에 대한 정보가 들어 있습니다. 두 번째 파일인 tablespace.ddl에는 테이블 스페이스 컨테이너를 지정하는 데 필요한 구문이 들어 있습니다. 테이블 스페이스 정의에 필요한 나머지 정보를 채우려면 이 파일을 갱신해야 합니다. db2listvolumes 명령에는 명령행 인수가 필요하지 않습니다. 이 명령은 Windows 운영 체제에서만 사용할 수 있습니다.

### 권한 부여

Administrator

### 필요한 연결

없음

### 명령 구문

▶▶—db2listvolumes—▶▶

### 명령 매개변수

없음



---

## 제 224 장 db2logsforrfwd - 롤 포워드 복구에 필요한 로그 나열

DB2TSCHG.HIS 파일 구문을 분석합니다. 이 유틸리티를 사용하면 사용자가 테이블 스페이스 롤 포워드 조작에 필요한 로그 파일을 찾을 수 있습니다. 이 유틸리티는 sqllib/bin에 있습니다.

### 권한 부여

없음

### 필수 연결

없음

### 명령 구문

▶▶—db2logsforrfwd—*path* -all▶▶

### 명령 매개변수

*path* DB2TSCHG.HIS 파일의 전체 경로 및 이름.

**-all** 자세한 정보를 표시합니다.

### 예:

```
db2logsForRfwd /home/ofer/ofer/NODE0000/S0000001/DB2TSCHG.HIS
```

```
db2logsForRfwd DB2TSCHG.HIS -all
```





---

## 제 225 장 db2look - DB2 통계 및 DDL 추출 도구

필요한 데이터 정의 언어(DDL)문을 추출하여 테스트 데이터베이스에서 프로덕션 데이터베이스의 데이터베이스 오브젝트를 다시 생성합니다. db2look 명령은 오브젝트 유형별로 DDL문을 생성합니다.

이 도구는 테스트 데이터베이스의 오브젝트에 대해 통계를 복제하는 데 사용되는 필수 UPDATE문을 생성할 수 있습니다. 테스트 데이터베이스의 레지스트리 변수 및 쿼리 옵티마이저 관련 구성 매개변수가 프로덕션 데이터베이스의 변수 및 매개변수와 일치하도록 UPDATE DATABASE CONFIGURATION 및 UPDATE DATABASE MANAGER CONFIGURATION 명령과 db2set 명령을 생성하기 위해 사용할 수도 있습니다.

종종 테스트 시스템에 프로덕션 시스템의 데이터 서브셋이 포함되도록 하는 것이 편리합니다. 그러나 이와 같은 테스트 시스템에 대해 선택된 액세스 플랜은 프로덕션 시스템에 대해 선택되는 액세스 플랜과 반드시 같을 필요는 없습니다. 테스트 시스템의 구성 매개변수 및 카탈로그 통계를 갱신하여 프로덕션 시스템의 것과 일치하도록 해야 합니다. 이 도구를 사용하면 액세스 플랜이 프로덕션 시스템에서 사용되는 것과 유사한 테스트 데이터베이스를 작성할 수 있습니다.

db2look 명령이 생성하는 DDL문을 점검해야 합니다. 원래 SQL 오브젝트의 모든 특성을 정확히 재생성하지 못할 수도 있기 때문입니다. 파티션된 데이터베이스 환경의 테이블 스페이스에 대해서는 일부 데이터베이스 파티션이 활성 상태가 아닌 경우 DDL이 완전하지 않을 수 있습니다. ACTIVATE 명령을 사용하여 모든 데이터베이스 파티션이 활성 상태인지 확인하십시오.

### 권한 부여

시스템 카탈로그 테이블에 대한 SELECT 특권

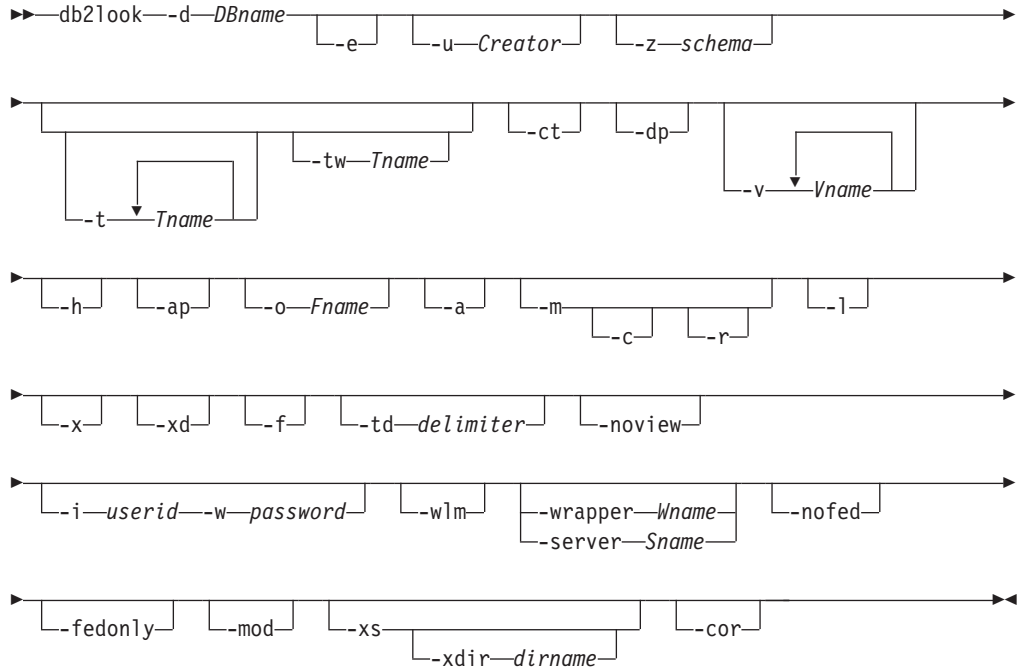
일부 경우(예: 테이블 스페이스 컨테이너 DDL을 생성하는 경우)에는 다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*
- *sysmon*
- *dbadm*

## 필수 연결

없음

## 명령 구문



## 명령 매개변수

### -d DBname

쿼리할 프로덕션 데이터베이스의 별명 이름. *DBname*은 Linux, UNIX 및 Windows용 DB2 데이터베이스 또는 z/OS용 DB2 버전 9.1(z/OS용 DB2) 데이터베이스의 이름이 될 수 있습니다. *DBname*이 z/OS용 DB2 데이터베이스인 경우, db2look 유틸리티는 OS/390 및 z/OS 오브젝트에 대한 DDL 및 UPDATE 통계 명령문을 추출합니다. 이 DDL 및 UPDATE 통계 명령문은 Linux, UNIX 및 Windows용 DB2 데이터베이스 데이터베이스에 적용 가능한 명령문으로 z/OS용 DB2 데이터베이스에는 적용되지 않습니다. 이는 OS/390 및 z/OS 오브젝트를 추출하고 Linux, UNIX 및 Windows용 DB2 데이터베이스 데이터베이스에서 이 오브젝트를 재작성하려고 하는 사용자에게 유용합니다.

*DBname*이 z/OS용 DB2 데이터베이스이면, db2look 명령의 출력은 다음으로 제한됩니다.

- 테이블, 인덱스, 뷰 및 사용자 정의 구별 유형에 대한 DDL 생성
- 테이블, 컬럼, 컬럼 분포 및 인덱스에 대한 UPDATE 통계 명령문 생성

- e 데이터베이스 오브젝트에 대한 DDL문을 추출합니다. 다음 데이터베이스 오브젝트에 대한 DDL은 -e 옵션을 사용할 때 추출됩니다.
  - 감사 규정
  - 스키마
  - 테이블(해당 테이블과, 파티션된 테이블에 대한 파티션 레벨 INDEX IN절에 대해 하나 이상이 존재하는 경우 인라인 길이 포함)
  - 뷰
  - 구체화된 쿼리 테이블(MQT)
  - 별명
  - 인덱스(파티션된 테이블의 파티션된 인덱스 포함)
  - 트리거
  - 시퀀스
  - 사용자 정의 구별 유형
  - 기본 키, 참조 무결성 및 점검 제한조건
  - 사용자 정의 구조화 유형
  - 사용자 정의 함수(UDF)
  - 사용자 정의 메소드
  - 사용자 정의 변환
  - 랩퍼
  - 서버
  - 사용자 맵핑
  - 별칭
  - 유형 맵핑
  - 함수 템플릿
  - 함수 맵핑
  - 인덱스 스펙
  - 스토어드 프로시저
  - 역할
  - 트러스트된 컨텍스트
  - 전역 변수
  - 보안 레이블 구성요소
  - 보안 규정
  - 보안 레이블

db2look 명령이 생성하는 DDL을 사용하여 사용자 정의 함수(UDF)를 다시 작성할 수 있습니다. 그러나 특정의 사용자 정의 함수가 참조하는 사용자 소스 코드(예: EXTERNAL NAME절)가 사용 가능해야 사용자 정의 함수를 사용할 수 있습니다.

**-u** *Creator*

작성자 ID. 이 작성자 ID를 가지고 있는 오브젝트로 출력을 제한합니다. 옵션 -a를 지정한 경우 이 매개변수는 무시됩니다. 출력에는 작동 불능 오브젝트가 포함되지 않습니다. 작동 불능 오브젝트를 표시하려면 -a 옵션을 사용하십시오.

**-z** *schema*

스키마 이름. 이 스키마 이름을 가지고 있는 오브젝트로 출력을 제한합니다. 출력에는 작동 불능 오브젝트가 포함되지 않습니다. 작동 불능 오브젝트를 표시하려면 -a 옵션을 사용하십시오. 이 매개변수가 지정되지 않으면 모든 스키마 이름의 오브젝트가 추출됩니다. -a 옵션을 지정하면 이 매개변수는 무시됩니다. 이 옵션은 페더레이티드 DDL에 대해 무시됩니다.

**-t** *Tname1 Tname2 ... TnameN*

테이블 이름 목록. 테이블 목록의 특정 테이블로 출력을 제한합니다. 최대 테이블 수는 30입니다. 테이블 이름은 공백으로 구분됩니다. 대소문자 구분 이름과 2바이트 문자 세트(DBCS) 이름은 백슬래시와 큰따옴표 분리문자 안에 묶어야 합니다(예: # " MyTabLe #"). 다중 단어 테이블 이름의 경우, 따옴표 안에 분리문자를 놓아서(예: #"My Table#") 명령행 처리기가 쌍을 단어별로 평가하지 않도록 해야 합니다. 다중 단어 테이블 이름을 백슬래시와 큰따옴표 분리문자로 묶지 않으면(예: "My Table"), 모든 단어가 대문자로 변환되고 db2look 명령은 대문자 테이블을 찾습니다(예: "MY TABLE"). -i와 함께 -t를 사용하는 경우, 이 조합은 DB2 버전 9.5에서 파티션된 테이블을 지원하지 않습니다.

**-tw** *Tname*

*Tname*에 지정된 패턴 기준과 일치하는 테이블 이름에 대한 DDL을 생성합니다. 리턴된 모든 테이블의 모든 종속 오브젝트에 대한 DDL도 생성합니다. *Tname*은 단일 값만 가능합니다. *Tname*의 밑줄 문자(\_)는 단일 문자를 나타냅니다. 퍼센트 부호(%)는 0개 이상의 문자로 된 문자열을 나타냅니다. *Tname*의 다른 문자는 단지 문자 자체를 나타냅니다. -tw를 지정한 경우 -t 옵션이 무시됩니다.

**-ct**

오브젝트 작성 시간별로 DDL을 생성합니다. 오브젝트 작성 시간별로 DDL을 생성하면 일부 오브젝트 DDL이 올바른 종속성 순서로 표시되지 않을 수도 있습니다. db2look 명령은 -ct 옵션도 지정하는 경우 -e, -a, -u, -z, -t, -tw, -v, -l, -noview, -wlm 옵션만 지원합니다.

**-dp**

CREATE문 이전에 DROP문을 생성합니다. 삭제(drop)된 오브젝트에 의존하는 오브젝트가 있는 경우 DROP문이 작동하지 않을 수 있습니다. 예를 들어, 삭제된 스키마에 의존하는 테이블이 있으면 스키마 삭제에 실패하고, 사용자 정

의 유형/함수에 의존하는 다른 유형, 함수, 트리거 또는 테이블이 있는 경우 사용자 정의 유형/함수 삭제에 실패합니다. 유형이 지정된 테이블의 경우, 루트 테이블에 대해서만 DROP TABLE HIERARCHY 문이 생성됩니다. DROP 문은 인덱스, 기본 및 외부 키, 제한조건에 대해 생성되지 않습니다. 테이블이 삭제될 때 항상 삭제되기 때문입니다. 테이블에 RESTRICT ON DROP 속성이 있으면 테이블을 삭제할 수 없습니다.

**-v** *Vname1 Vname2 ... VnameN*

지정된 뷰에 대한 DDL을 생성합니다. 최대 뷰 수는 30입니다. -t 옵션을 지정한 경우 -v 옵션이 무시됩니다. 대소문자 구분, DBCS 및 다중 단어 테이블 이름에 관한 규칙은 뷰 이름에도 적용됩니다.

**-h** 도움말 정보를 표시합니다. 이 옵션이 지정되면 다른 모든 옵션은 무시되고 도움말 정보만 표시됩니다.

**-ap** 감사 규정을 다른 데이터베이스 오브젝트와 연관시키는데 필요한 AUDIT USING 문을 생성합니다.

**-o** *Fname*

출력을 *filename.sql*에 기록합니다. 이 옵션을 지정하지 않으면 출력은 표준 출력에 기록됩니다. 확장자와 함께 파일 이름을 지정하면 출력은 해당 파일에 기록됩니다.

**-a** 이 옵션을 지정하면 출력은 특정 작성자 ID에 의해 작성된 오브젝트로 제한되지 않습니다. 작동 불능 오브젝트를 포함하여, 모든 사용자가 작성한 모든 오브젝트가 고려됩니다. 예를 들어, -e 옵션과 함께 이 옵션을 지정하면 데이터베이스에 있는 모든 오브젝트에 대해 DDL문이 추출됩니다. -m 옵션과 함께 이 옵션을 지정하면 데이터베이스에 있는 모든 사용자 작성 테이블 및 인덱스에 대해 UPDATE 통계 명령문이 추출됩니다. -u나 -a 모두 지정하지 않으면 환경 변수 USER가 사용됩니다. UNIX 운영 체제에서는 이 변수를 명시적으로 설정하지 않아도 됩니다. 그러나 Windows 시스템에는 USER 환경 변수의 디폴트값이 없습니다. SYSTEM 변수에서 사용자 변수를 설정하거나 세션에 대해 set USER=*username*을 발행해야 합니다.

**-m** 테이블, 통계 뷰, 컬럼 및 인덱스에 대해 통계를 복제하는 데 사용되는 필수 UPDATE문을 생성합니다.

**-c** -m 옵션과 함께 이 옵션을 지정하는 경우, db2look 명령은 COMMIT, CONNECT 및 CONNECT RESET 문을 생성하지 않습니다. 디폴트 조치는 이 명령문을 생성하는 것입니다.

**-r** -m 옵션과 함께 이 옵션을 지정하는 경우, db2look 명령은 RUNSTATS 명령을 생성하지 않습니다. 디폴트 조치는 RUNSTATS 명령을 생성하는 것입니다.

주: 다른 데이터베이스에 대해 가상 갱신 모드(-m 옵션)에서 db2look 명령을 사용하여 작성된 명령 프로세서 스크립트를 실행하려는 경우(예를 들어, 테스트 데이터베이스의 카탈로그 통계가 프로덕션 데이터베이스의 통계와 일치하도록 하기 위해) 두 데이터베이스 모두 동일한 코드 세트 및 지역을 사용해야 합니다.

**-l** 이 옵션을 지정하는 경우, db2look 명령은 사용자 정의 테이블 스페이스, 데이터베이스 파티션 그룹 및 버퍼 풀에 대해 DDL을 생성합니다. 다음 데이터베이스 오브젝트에 대한 DDL은 -l 옵션을 사용할 때 추출됩니다.

- 사용자 정의 테이블 스페이스
- 사용자 정의 데이터베이스 파티션 그룹
- 사용자 정의 버퍼 풀

**-x** 이 옵션을 지정하는 경우, db2look 명령은 권한 부여 DDL(예: GRANT문)을 생성합니다.

지원되는 권한은 다음과 같습니다.

- 테이블: ALTER, SELECT, INSERT, DELETE, UPDATE, INDEX, REFERENCE, CONTROL
- 뷰: SELECT, INSERT, DELETE, UPDATE, CONTROL
- 인덱스: CONTROL
- 스키마: CREATEIN, DROPIN, ALTERIN
- 데이터베이스: ACCESSCTRL, BINDADD, CONNECT, CREATETAB, CREATE\_EXTERNAL\_ROUTINE, CREATE\_NOT\_FENCED\_ROUTINE, DATAACCESS, DBADM, EXPLAIN, IMPLICIT\_SCHEMA, LOAD, QUIESCE\_CONNECT, SECADM, SQLADM, WLMADM
- 사용자 정의 함수(UDF): EXECUTE
- 사용자 정의 메소드: EXECUTE
- 스토어드 프로시저: EXECUTE
- 패키지: CONTROL, BIND, EXECUTE
- 컬럼: UPDATE, REFERENCES
- 테이블 스페이스: USE
- 시퀀스: USAGE, ALTER
- 워크로드: USAGE
- 전역 변수
- 역할
- 보안 레이블
- 면제

**-xd** 이 옵션을 지정하는 경우, db2look 명령은 오브젝트 작성 시 SYSIBM에 의해 권한이 부여된 오브젝트에 대한 권한 부여 DDL을 포함하여 모든 권한 부여 DDL을 생성합니다.

**-f** 쿼리 옵티마이저에 영향을 주는 구성 매개변수와 레지스트리 변수를 추출하려면 이 옵션을 사용하십시오.

**-td delimiter**

db2look 명령으로 생성되는 SQL문의 명령문 분리문자를 지정합니다. 이 옵션을 지정하지 않는 경우 디폴트는 세미콜론(;)입니다. -e 옵션이 지정된 경우 이 옵션을 사용할 것을 권장합니다. 이 경우, 추출된 오브젝트에는 트리거 또는 SQL 루틴이 포함될 수 있습니다.

**-noview**

이 옵션을 지정하면 CREATE VIEW DDL문이 추출되지 않습니다.

**-i userid**

리모트 데이터베이스에 대해 작업할 때 이 옵션을 사용하십시오.

**-w password**

-i 옵션과 함께 사용하는 경우, 사용자는 이 매개변수를 사용하여 리모트 시스템에 있는 데이터베이스에 대해 db2look 명령을 실행할 수 있습니다. 리모트 시스템에 로그인하기 위해 db2look에서 사용자 ID 및 암호가 사용됩니다. 리모트 데이터베이스에 대해 작업하는 경우, 리모트 데이터베이스의 버전은 로컬 데이터베이스와 같아야 합니다. db2look 명령에는 하위 레벨 또는 상위 레벨 지원이 없습니다.

**-wlm** 이 옵션은 다음에 대해 CREATE문과 ALTER문을 생성하기 위해 제공할 수 있는 WLM 특정 DDL 출력을 생성합니다.

- 막대 그래프
- WLM 이벤트 모니터
- 서비스 클래스
- 워크로드
- 임계값
- 작업 클래스 세트
- 작업 조치 세트

**-wrapper Wname**

해당 래퍼에 적용되는 페더레이티드 오브젝트에 대한 DDL문을 생성합니다. 생성될 수 있는 페더레이티드 DDL문은 CREATE WRAPPER, CREATE SERVER, CREATE USER MAPPING, CREATE NICKNAME, CREATE TYPE MAPPING, CREATE FUNCTION ... AS TEMPLATE, CREATE FUNCTION MAPPING, CREATE INDEX SPECIFICATION, GRANT(별

명, 서버, 인덱스에 특권 부여)입니다. 하나의 랩퍼 이름만 지원됩니다. 하나 미만 또는 두 개 이상의 이름을 지정하면 오류가 리턴됩니다. 이 옵션은 관계형이 아닌 데이터 소스를 지원하지 않습니다.

**-server** *Sname*

해당 서버에 적용되는 페더레이티드 오브젝트에 대한 DDL문을 생성합니다. 생성될 수 있는 페더레이티드 DDL문은 CREATE WRAPPER, CREATE SERVER, CREATE USER MAPPING, CREATE NICKNAME, CREATE TYPE MAPPING, CREATE FUNCTION ... AS TEMPLATE, CREATE FUNCTION MAPPING, CREATE INDEX SPECIFICATION, GRANT(별명, 서버, 인덱스에 특권 부여)입니다. 하나의 서버 이름만 지원됩니다. 하나 미만 또는 두 개 이상의 이름을 지정하면 오류가 리턴됩니다. 이 옵션은 관계형이 아닌 데이터 소스를 지원하지 않습니다.

**-nofed** 페더레이티드 DDL문이 생성되지 않음을 지정합니다. 이 옵션을 지정하면 **-wrapper** 및 **-server** 옵션은 무시됩니다.

**-fedonly**

페더레이티드 DDL문만 생성됨을 지정합니다.

**-mod** 각 모듈 및 각 모듈에 정의된 모든 오브젝트에 대해 DDL을 생성합니다.

**-xs** 목표 데이터베이스에서 XML 스키마와 DTD를 등록하는 데 필요한 모든 파일을 익스포트하고 파일 등록에 적절한 명령을 생성합니다. 익스포트되는 XSR 오브젝트 세트는 **-u**, **-z** 및 **-a** 옵션으로 제어됩니다.

**-xdir** *dirname*

익스포트된 XML 관련 파일을 제공된 경로에 위치시킵니다. 이 옵션이 지정되지 않으면, 모든 XML 관련 파일이 현재 디렉토리로 익스포트됩니다.

**-cor** CREATE OR REPLACE절로 DDL문을 생성하며 이 경우 해당 절에 처음부터 해당 문이 포함되어 있는지 여부는 상관이 없습니다.

**예:**

- 데이터베이스 DEPARTMENT에서 사용자 walid가 작성한 오브젝트에 대한 DDL문을 생성합니다. db2look 출력은 db2look.sql 파일로 보냅니다.

```
db2look -d department -u walid -e -o db2look.sql
```

- 데이터베이스 DEPARTMENT에서 사용자 walid가 작성한, 스키마 이름이 ianhe 인 오브젝트에 대한 DDL문을 생성합니다. db2look 출력은 db2look.sql 파일로 보냅니다.

```
db2look -d department -u walid -z ianhe -e -o db2look.sql
```

- 데이터베이스 DEPARTMENT에서 사용자 walid가 작성한 데이터베이스 오브젝트에 대한 통계를 복제하기 위한 UPDATE문을 생성합니다. 출력은 db2look.sql 파일로 보냅니다.



```
db2look -d department -u walid -m -o db2look.sql
```

- 사용자 walid가 작성한 오브젝트에 대한 DDL문과 동일한 사용자가 작성한 데이터베이스 오브젝트에 대한 통계를 복제하기 위한 UPDATE문을 생성합니다. db2look 출력은 db2look.sql 파일로 보냅니다.

```
db2look -d department -u walid -e -m -o db2look.sql
```

- 데이터베이스 DEPARTMENT에서 모든 사용자가 작성한 오브젝트에 대한 DDL문을 생성합니다. db2look 출력은 db2look.sql 파일로 보냅니다.

```
db2look -d department -a -e -o db2look.sql
```

- 모든 사용자 정의 데이터베이스 파티션 그룹, 버퍼 풀 및 테이블 스페이스에 대한 DDL문을 생성합니다. db2look 출력은 db2look.sql 파일로 보냅니다.

```
db2look -d department -l -o db2look.sql
```

- 옵티마이저 관련 데이터베이스 및 데이터베이스 관리 프로그램 구성 매개변수에 대한 UPDATE문과, 옵티마이저 관련 레지스트리 변수에 대한 db2set문을 데이터베이스 DEPARTMENT에서 생성합니다. db2look 출력은 db2look.sql 파일로 보냅니다.

```
db2look -d department -f -o db2look.sql
```

- 데이터베이스 DEPARTMENT에서 모든 오브젝트에 대한 DDL, 데이터베이스 DEPARTMENT에서 모든 테이블 및 인덱스에 대한 통계를 복제하기 위한 UPDATE문, GRANT 권한 부여 명령문, 옵티마이저 관련 데이터베이스 및 데이터베이스 관리 프로그램 구성 매개변수에 대한 UPDATE문, 옵티마이저 관련 레지스트리 변수에 대한 db2set문, 그리고 데이터베이스 DEPARTMENT에서 모든 사용자 정의 데이터베이스 파티션 그룹, 버퍼 풀 및 테이블 스페이스에 대한 DDL을 생성합니다. 출력은 db2look.sql 파일로 보냅니다.

```
db2look -d department -a -e -m -l -x -f -o db2look.sql
```

- 원래 작성자가 작성한 오브젝트를 포함하여, 데이터베이스 DEPARTMENT에서 모든 오브젝트에 대한 모든 권한 부여 DDL문을 생성합니다. (이 경우, 권한 부여는 오브젝트 작성 시 SYSIBM에 의해 부여되었습니다.) db2look 출력은 db2look.sql 파일로 보냅니다.

```
db2look -d department -xd -o db2look.sql
```

- 데이터베이스 DEPARTMENT에서 모든 사용자가 작성한 오브젝트에 대한 DDL문을 생성합니다. db2look 출력은 db2look.sql 파일로 보냅니다.

```
db2look -d department -a -e -td % -o db2look.sql
```

출력은 CLP에서 읽을 수 있습니다.

```
db2 -td% -f db2look.sql
```

- CREATE VIEW 문을 제외하고, 데이터베이스 DEPARTMENT에서 오브젝트에 대한 DDL문을 생성합니다. db2look 출력은 db2look.sql 파일로 보냅니다.

```
db2look -d department -e -noview -o db2look.sql
```

- 데이터베이스 DEPARTMENT에서 지정된 테이블에 관련된 오브젝트에 대한 DDL 문을 생성합니다. db2look 출력은 db2look.sql 파일로 보냅니다.

```
db2look -d department -e -t tab1 # "My Table#" -o db2look.sql
```

- 페더레이티드 데이터베이스 FEDDEPART에서 모든 오브젝트(페더레이티드 및 비페더레이티드)에 대한 DDL문을 생성합니다. 페더레이티드 DDL문의 경우, 지정된 랩퍼 FEDWRAP에 적용되는 명령문만 생성됩니다. db2look 출력은 표준 출력으로 보냅니다.

```
db2look -d feddepart -e -wrapper fedwrap
```

- 비페더레이티드 DDL문만 포함하는 스크립트 파일을 생성합니다. 다음 시스템 명령은 페더레이티드 데이터베이스(FEDDEPART)에 대해 실행될 수 있으며 아직은 페더레이티드 데이터베이스가 아닌 데이터베이스에 대해 실행될 때 발견되는 것과 같은 출력만 생성합니다. db2look 출력은 out.sql 파일로 보냅니다.

```
db2look -d feddepart -e -nofed -o out
```

- 데이터베이스 DEPARTMENT에서 스키마 이름이 valid인 오브젝트에 대한 DDL 문을 생성합니다. 포함된 XML 스키마와 DTD를 등록하는 데 필요한 파일이 현재 디렉토리로 익스포트됩니다. db2look 출력은 db2look.sql 파일로 보냅니다.

```
db2look -d department -z valid -e -xs -o db2look.sql
```

- 데이터베이스 DEPARTMENT에서 모든 사용자가 작성한 오브젝트에 대한 DDL문을 생성합니다. 포함된 XML 스키마와 DTD를 등록하는 데 필요한 파일이 /home/ofer/ofer/ 디렉토리로 익스포트됩니다. db2look 출력은 표준 출력으로 보냅니다.

```
db2look -d department -a -e -xs -xdir /home/ofer/ofer/
```

- 데이터베이스 DEPARTMENT에서 독립적으로 WLM 특정 DDL을 생성합니다.

```
db2look -d department -wlm
```

데이터베이스 DEPARTMENT에서 모든 오브젝트에 대한 DDL을 생성합니다.

```
db2look -d department -wlm -e -l
```

## 사용 시 참고사항

Windows 운영 체제에서는 DB2 명령 창에서 db2look 명령을 실행해야 합니다.

기존 옵션 중 몇 개의 옵션은 페더레이티드 환경을 지원합니다. 다음 db2look 명령행 옵션은 페더레이티드 환경에서 사용됩니다.

- -ap

사용된 경우, AUDIT USING 문이 생성됩니다.

- -e

사용된 경우, 페더레이티드 DDL문이 생성됩니다.

- -x

사용된 경우, 페더레이티드 오브젝트에 특권을 부여하기 위한 GRANT문이 생성됩니다.

- -xd

사용된 경우, 페더레이티드 오브젝트에 시스템에서 부여한 특권을 추가하기 위한 페더레이티드 DDL문이 생성됩니다.

- -f

사용된 경우, 데이터베이스 관리 프로그램 구성에서 페더레이티드 관련 정보가 추출됩니다.

- -m

사용된 경우, 별칭에 대한 통계가 추출됩니다.

- -wlm

사용된 경우, WLM 특정 DDL이 출력됩니다.

페더레이티드 DDL문을 작성하려면 데이터베이스 관리 프로그램 구성에서 페더레이티드 시스템을 사용할 수 있는 기능을 사용 가능하도록 설정해야 합니다. db2look 명령이 스크립트 파일을 생성하면, **federated** 구성 매개변수를 YES로 설정한 후 스크립트를 실행해야 합니다.

CREATE USER MAPPING 문의 리모트 암호를 추가하도록 출력 스크립트를 수정해야 합니다.

DB2 계열 인스턴스를 데이터 소스로 정의하기 위해 사용되는 CREATE SERVER 문에 AUTHORIZATION 및 PASSWORD를 추가하여 db2look 명령 출력 스크립트를 수정해야 합니다.

-tw 옵션은 다음과 같이 사용합니다.

- DEPARTMENT 데이터베이스에서 이름이 abc로 시작하는 테이블과 연관되는 오브젝트에 대한 DDL문을 생성하고 출력을 db2look.sql 파일로 보내려면 다음 명령을 사용하십시오.

```
db2look -d department -e -tw abc% -o db2look.sql
```

- DEPARTMENT 데이터베이스에서 이름의 두 번째 문자가 d인 테이블과 연관되는 오브젝트에 대한 DDL문을 생성하고 출력을 db2look.sql 파일로 보내려면 다음 명령을 사용하십시오.

```
db2look -d department -e -tw _d% -o db2look.sql
```

- db2look 명령은 *Tname* 인수에 지정된 패턴과 일치하는 테이블 이름을 평가할 때 LIKE 술어를 사용합니다. LIKE 술어가 사용되므로, \_ 문자나 % 문자가 테이블 이

름의 일부인 경우 백슬래시(\) Escape 문자를 \_ 또는 % 바로 앞에 사용해야 합니다. 이 경우, \_과 %는 *Tname*에서 와일드 카드 문자로 사용할 수 없습니다. 예를 들어, DEPARTMENT 데이터베이스에서 이름의 첫 번째 및 마지막 위치에 퍼센트 부호없는 테이블과 연관되는 오브젝트에 대한 DDL문을 생성하려면 다음 명령을 사용하십시오.

```
db2look -d department -e -tw string%string
```

- 대소문자 구분, DBCS 및 다중 단어 테이블 및 뷰 이름은 백슬래시와 큰따옴표로 묶어야 합니다. 예를 들면, 다음과 같습니다.

```
\#"My Table#"
```

멀티바이트 문자 세트(MBCS) 또는 2바이트 문자 세트(DBCS) 이름이 백슬래시와 큰따옴표 분리문자로 묶이지 않고, 소문자와 같은 바이트를 포함하는 경우, 대문자로 변환되므로 db2look 명령은 변환된 이름의 데이터베이스 오브젝트를 찾습니다. 결과적으로, DDL문은 추출되지 않습니다.

- -tw 옵션은 -x 옵션(GRANT 특권을 생성하는 경우), -m 옵션(테이블 및 컬럼 통계를 리턴하는 경우) 및 -l 옵션(사용자 정의 테이블 스페이스, 데이터베이스 파티션 그룹 및 버퍼 풀에 대한 DDL을 생성하는 경우)과 함께 사용할 수 있습니다. -t 옵션을 -tw 옵션과 함께 지정하면, -t 옵션(및 연관된 *Tname* 인수)은 무시됩니다.
- -tw 옵션은 페더레이티드 데이터 소스나, z/OS용 DB2, i용 DB2 또는 VSE & VM용 DB2 서버에 있는 테이블(및 해당되는 연관 오브젝트)에 대한 DDL을 생성하는 데 사용할 수 없습니다.
- -tw 옵션은 CLP를 통해서만 지원됩니다.

데이터베이스 파티셔닝 기능을 사용하여 시스템에서 DDL을 요청하는 경우, 비활성 데이터베이스 파티션에 존재하는 테이블 스페이스에 대한 DDL 대신 경고 메시지가 표시 됩니다. 모든 테이블 스페이스에 대해 적절한 DDL이 생성되도록 하려면 모든 데이터베이스 파티션을 활성화해야 합니다.

배열 유형의 보안 레이블 구성요소에 대한 DDL을 추출할 때, 추출된 DDL은 내부 표현(즉, 해당 배열에서의 요소 인코딩)이, db2look 추출이 수행된 데이터베이스 내의 해당되는 보안 레이블 구성요소의 내부 표현과 일치하는 보안 레이블 구성요소를 생성할 수 없습니다. 이는 배열 유형의 보안 레이블 구성요소가 변경되고 하나 이상의 요소가 추가된 경우에 발생할 수 있습니다. 이와 같은 경우, 하나의 테이블에서 추출되어 다른 테이블(db2look 출력에서 작성된)로 이동된 데이터에는 해당되는 보안 레이블 값이 수반되지 않으므로 새 테이블 보호가 손상될 수 있습니다.

## 관련 정보

별칭 컬럼 및 인덱스 이름

이주용 응용프로그램 변경

## 제 226 장 db2ls - 설치된 DB2 제품 및 기능 나열

DB2 HTML 문서를 포함하여 Linux 및 UNIX 시스템에 설치된 DB2 제품과 기능을 나열합니다. 시스템에 DB2 제품의 여러 사본을 설치하는 기능을 사용하고 선택한 경로에 DB2 제품과 기능을 설치할 수 있는 경우 db2ls 명령을 사용하여 다음을 나열할 수 있습니다.

- 시스템에서 DB2 제품이 설치된 위치와 DB2 제품 레벨을 나열합니다.
- 특정 설치 경로의 모든 또는 특정 DB2 제품과 기능을 나열합니다.

db2ls 명령은 설치 미디어와 시스템의 DB2 설치 사본에 모두 있을 수 있습니다. db2ls 명령은 둘 중 임의의 위치에서 실행할 수 있습니다. db2ls 명령은 모든 제품의 설치 미디어에서 실행할 수 있습니다(IBM Data Server Driver Package 제외).

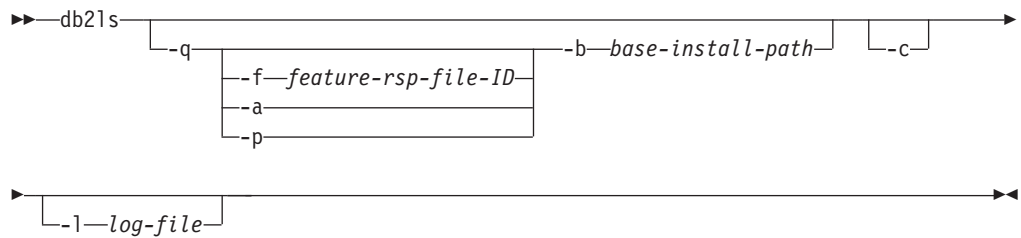
### 권한 부여

없음

### 필요한 연결

없음

### 명령 구문



### 명령 매개변수

**-q** 쿼리로 설치된 DB2 제품과 기능을 나열하는 것으로 지정합니다. **-a** 매개변수도 지정되지 않은 경우 디폴트로, 표시 가능한 구성요소(기능)만 표시됩니다.

#### **-f** feature-rsp-file-ID

설치된 특정 기능을 쿼리합니다. 기능이 설치되지 않은 경우 프로그램의 리턴 코드는 0이 아니며 설치된 경우에는 리턴 코드가 0입니다.

**-a** 숨겨진 모든 구성요소와 표시 가능 기능을 나열합니다. db2ls 명령은 디폴트로 표시 가능 기능만 나열합니다.

**-p** 제품만 나열합니다. 기능을 나열하지 않고 고객이 설치한 제품의 간략한 목록을 제공합니다.

**-b** *base-install-path*

/usr/local/bin의 글로벌 db2ls 명령을 사용하는 경우 쿼리하는 디렉토리를 지정해야 합니다. 글로벌 db2ls 명령은 해당 설치 경로에서 db2ls를 호출하고 나머지 매개변수로 전달합니다.

**-c** 컬럼 기반이 아닌 콜론으로 구분되는 항목의 목록으로 출력을 인쇄합니다. 따라서 이 정보를 프로그램에 사용할 수 있습니다. 첫 번째 출력 라인은 각 항목을 설명하는 콜론으로 구분된 토큰 목록입니다. 이 첫 번째 라인은 해시 문자("#")로 시작되어 프로그램에서 쉽게 무시할 수 있습니다.

**-l** *log-file*

디버깅 목적으로 사용할 로그 파일을 추적합니다.

**예:**

• 특정 경로에 설치된 DB2 데이터베이스 기능을 쿼리하려면 다음을 실행하십시오.

```
db2ls -q -b /opt/ibm/ese/v9
```

• 특정 경로에 설치된 모든 DB2 데이터베이스 기능을 보려면 다음을 실행하십시오.

```
db2ls -q -a -b /opt/ibm/ese/v9
```

• 특정 DB2 데이터베이스 기능이 설치되었는지 여부를 확인하려면 다음을 실행하십시오.

```
db2ls -q -b /opt/ibm/ese/v9 -f <feature>
```

**사용 시 참고사항**

• db2ls 명령은 Windows 운영 체제에서 사용할 수 없습니다.

• 루트에 /usr/local/bin의 쓰기 권한이 있거나 /usr/local/bin을 작성할 수 있는 경우, 시스템에 설치된 DB2 버전 9 이상의 최초 설치를 위한 DB2DIR/install/db2ls를 가리키는 기호 링크 /usr/local/bin/db2ls가 작성됩니다. DB2의 여러 사본이 설치된 경우 루트는 시스템에 설치된 DB2의 최상위 버전과 레벨을 가리키는 링크를 갱신합니다.

루트 서버가 아닌 서버 설치는 /usr/local/bin/db2ls를 작성 또는 변경하지 않습니다. 이러한 경우 db2ls를 실행하려면 다음 두 가지 중 하나를 수행해야 합니다.

– 사용자 경로에 *inst\_home/sql/lib/install*을 추가합니다. 그런 다음 db2ls를 루트 서버가 아닌 사용자로서 실행할 수 있습니다.

– 명령의 정확한 경로(예를 들어, *inst\_home/sql/lib/install/db2ls*)로 전달합니다.

• db2ls 명령은 버전 9 이상의 DB2 제품을 쿼리할 수 있는 유일한 방법입니다. DB2 제품은 Linux 또는 UNIX 운영 체제 원시(native) 유틸리티(예를 들어, pkgadd, rpm,

SMIT 또는 swinstall)를 사용하여 쿼리할 수 없습니다. DB2 설치를 통한 인터페이스 및 쿼리에 사용하는 원시(native) 설치 유틸리티가 포함된 기존 스크립트를 변경해야 합니다.

- DB2 설치의 루트 대 루트 서버가 아닌 방법과 명령을 실행하는 사용자에 따라 다른 기능 목록을 얻을 수 있습니다.

-q 옵션을 사용하지 않는 경우:

- 루트 서버가 아닌 서버 설치 인스턴스 사용자를 제외한 모든 사용자의 경우 이 명령은 루트 사용자가 설치한 모든 사본을 표시합니다.
- 루트 서버가 아닌 서버 설치 인스턴스 사용자의 경우 이 명령은 루트 사용자가 설치한 모든 DB2 사본과 루트 서버가 아닌 서버 설치 사용자가 소유하는 루트 서버가 아닌 사본을 표시합니다.

-q 옵션을 사용하는 경우:

- 사용자 A가 사용자 B에게 DB2가 설치되었는지 여부를 알리기 위해 사용자 A는 db2ls -q -b \$userBHomeDir/sqlllib를 실행할 수 있습니다. 사용자 A에게 액세스 권한이 있으면 사용자 B가 설치한 DB2 기능이 표시되며, 그렇지 않으면 액세스 권한이 거부되었음을 나타내는 오류 메시지가 리턴됩니다.
  - -b 옵션 없이 db2ls -q를 실행하면 db2ls가 속하는 설치 경로에 설치된 기능이 표시됩니다.
- 디렉토리가 읽기 전용인 경우 db2ls 명령을 /usr/local/bin 디렉토리에서 링크할 수 없습니다. 시스템 워크로드 파티션(WPAR)에서 실행하는 경우 설치 이미지 루트 디렉토리에 있는 db2ls 명령을 사용하여 설치된 사본의 목록을 쿼리할 수 있습니다.





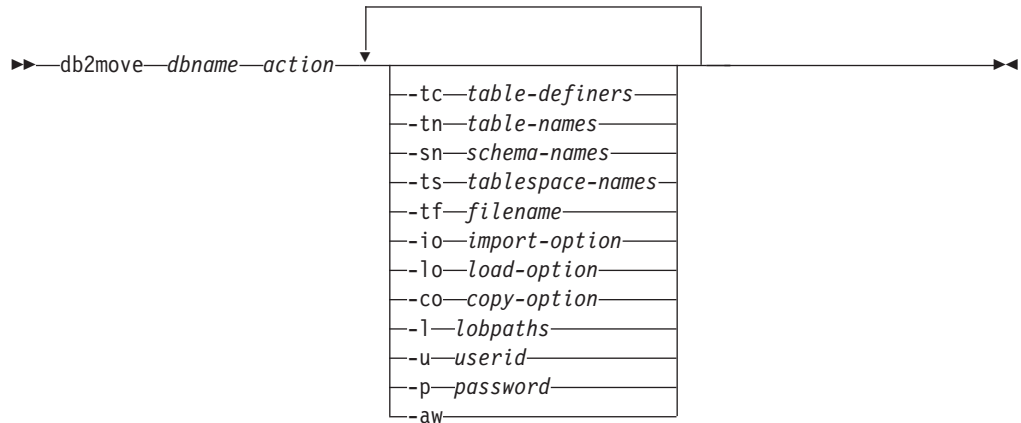
## 제 227 장 db2move - 데이터베이스 이동 도구

이 도구는 EXPORT/IMPORT/LOAD 모드에서 사용할 때 워크스테이션에 있는 DB2 데이터베이스 사이에 많은 수의 테이블 이동을 용이하게 합니다. 도구는 시스템 카탈로그 테이블에서 특정 데이터베이스를 쿼리하고 모든 사용자 테이블의 목록을 컴파일합니다. 그런 다음 이들 테이블을 PC/IXF 형식으로 익스포트합니다. PC/IXF 파일은 동일한 시스템의 다른 로컬 DB2 데이터베이스에 임포트 또는 로드할 수 있거나, 다른 워크스테이션 플랫폼으로 변환하고 해당 플랫폼의 DB2 데이터베이스에 임포트 또는 로드할 수 있습니다. 이 도구를 사용할 때 구조화된 유형을 갖는 테이블은 이동되지 않습니다. COPY 모드에서 사용할 때 이 도구는 스키마 중복을 용이하게 합니다.

### 권한 부여

이 도구는 사용자가 요청한 조치에 따라서 DB2 export, import 및 load API를 호출합니다. 그러므로 요청하는 사용자 ID가 해당 API에서 필요한 올바른 권한 부여를 가져야 하며, 그렇지 않으면 요청은 실패합니다.

### 명령 구문



### 명령 매개변수

*dbname*

데이터베이스 이름

*action* 다음 중 하나여야 합니다.

#### EXPORT

options의 필터링 기준을 만족하는 모든 테이블을 익스포트합니다. options가 지정되지 않으면 모든 테이블을 익스포트합니다. 내부 스테이징 정보는 db2move.lst 파일에 저장됩니다.

## IMPORT

내부 스테이징 파일 db2move.lst에 나열되는 모든 테이블을 임포트합니다. IMPORT 특정 조치에 대해서는 -io 옵션을 사용하십시오.

## LOAD

내부 스테이징 파일 db2move.lst에 나열되는 모든 테이블을 로드합니다. LOAD 특정 조치에 대해서는 -lo 명령을 사용하십시오.

## COPY

스키마를 목표 데이터베이스에 중복합니다. -sn 옵션을 사용하여 하나 이상의 스키마를 지정하십시오. COPY 특정 옵션에 대해서는 -co 옵션을 참조하십시오. LOAD\_ONLY 모드에서 테이블을 필터링하려면 -tn 또는 -tf 옵션을 사용하십시오.

각 조치 중에 생성되는 파일 목록은 아래를 참조하십시오.

### **-tc** *table-definers*

디폴트는 모든 정의자입니다.

이것은 EXPORT 조치 전용입니다. 지정되는 경우 이 옵션과 함께 나열되는 정의자가 작성한 테이블만 익스포트됩니다. 지정되지 않는 경우 디폴트는 모든 정의자를 사용하는 것입니다. 다중 정의자를 지정할 때 정의자를 쉼표로 분리해야 하며 정의자 ID 사이에 공백은 허용되지 않습니다. 이 옵션을 -tn *table-names* 옵션과 함께 사용하여 익스포트할 테이블을 선택할 수 있습니다. 별표(\*)를 문자열의 어디에나 배치될 수 있는 와일드 카드로 사용할 수 있습니다.

### **-tn** *table-names*

디폴트는 모든 사용자 테이블입니다.

이것은 EXPORT 또는 COPY 조치 전용입니다.

EXPORT 옵션과 함께 지정하면 지정된 문자열에 있는 이름과 이름이 일치하는 테이블만 익스포트됩니다. 지정되지 않는 경우 디폴트는 모든 사용자 테이블을 사용하는 것입니다. 다중 테이블 이름을 지정할 때 이름을 쉼표로 분리해야 하며 테이블 이름 사이에 공백은 허용되지 않습니다. 테이블 이름은 규정되지 않은 채로 나열되어야 하며 -sn 옵션을 사용하여 스키마를 필터링해야 합니다.

익스포트의 경우 별표(\*)를 문자열의 어디에나 배치될 수 있는 와일드 카드로 사용할 수 있습니다.

COPY 조치와 함께 지정되는 경우 -co "MODE" LOAD\_ONLY *copy-option* 도 지정해야 하며, 지정된 테이블만 목표 데이터베이스에서 다시 채워집니다. 테이블 이름은 "schema"."table" 형식으로 스키마 규정자와 함께 나열되어야 합니다.

### **-sn** *schema-names*

EXPORT의 디폴트는 모든 스키마입니다(COPY의 경우는 아님).

지정되는 경우 스키마 이름이 일치하는 테이블만 익스포트 또는 복사됩니다. 다중 스키마 이름이 지정되는 경우 쉼표로 구분해야 하며 스키마 이름 사이에 공백은 허용되지 않습니다. 8문자 미만의 스키마 이름은 8문자까지 채워집니다.

익스포트의 경우, 별표 와일드 카드 문자(\*)가 스키마 이름에서 사용되는 경우 퍼센트 기호(%)로 변경되며 테이블 이름(퍼센트 기호를 갖는)이 WHERE절의 LIKE 술어에서 사용됩니다. 지정되지 않는 경우 디폴트는 모든 스키마를 사용하는 것입니다. -tn 또는 -tc 옵션과 함께 사용되는 경우 db2move는 스키마가 지정된 스키마 이름과 일치하고 정의자가 지정된 정의자와 일치하는 테이블에만 작용합니다. 별표를 사용할 때 스키마 이름 fred에 -sn fr\*d 대신 -sn fr\*d\*가 지정되어야 합니다.

### **-ts** *tablespace-names*

디폴트는 모든 테이블 스페이스입니다.

이것은 EXPORT 조치 전용입니다. 이 옵션이 지정되면 지정된 테이블 스페이스에 있는 테이블만 익스포트됩니다. 별표 와일드 카드 문자(\*)가 테이블 스페이스 이름에서 사용되는 경우 퍼센트 기호(%)로 변경되며 테이블 이름(퍼센트 기호를 갖는)이 WHERE절의 LIKE 술어에서 사용됩니다. -ts 옵션이 지정되지 않는 경우 디폴트는 모든 테이블 스페이스를 사용하는 것입니다. 다중 테이블 스페이스 이름이 지정되는 경우 쉼표로 구분해야 하며 테이블 스페이스 이름 사이에 공백은 허용되지 않습니다. 8문자 미만의 테이블 스페이스 이름은 8문자까지 채워집니다. 예를 들어 테이블 스페이스 이름 mytb는 별표를 사용할 때 -sn my\*b 대신 -ts my\*b\*로 지정되어야 합니다.

### **-tf** *filename*

EXPORT 옵션과 함께 지정하면 이름이 지정된 파일에 있는 이름과 일치하는 테이블만 익스포트됩니다. 지정되지 않는 경우 디폴트는 모든 사용자 테이블을 사용하는 것입니다. 테이블은 라인당 하나씩 나열되어야 하며 각 테이블은 완전해야 합니다. 문자열에서 와일드 카드 문자는 허용되지 않습니다. 다음은 파일 콘텐츠의 예입니다.

```
"SCHEMA1"."TABLE NAME1"  
"SCHEMA NAME77"."TABLE155"
```

COPY 조치와 함께 지정되는 경우 -co "MODE" LOAD\_ONLY *copy-option* 도 지정해야 하며, 파일에서 지정된 테이블만 목표 데이터베이스에서 다시 채워집니다. 테이블 이름은 "schema"."table" 형식으로 스키마 규정자와 함께 나열되어야 합니다.

**-io import-option**

디폴트는 REPLACE\_CREATE입니다. 임포트 작성 기능의 제한사항은 『IMPORT 명령 옵션 CREATE 및 REPLACE\_CREATE는 사용되지 않음』을 참조하십시오.

유효한 옵션은 INSERT, INSERT\_UPDATE, REPLACE, CREATE 및 REPLACE\_CREATE입니다.

**-lo load-option**

디폴트는 INSERT입니다.

유효한 옵션은 INSERT와 REPLACE입니다.

**-co db2move 조치가 COPY일 때, 다음 -co 후속 옵션을 사용할 수 있습니다.**

**“TARGET\_DB db name [USER userid USING password]”**

사용자가 목표 데이터베이스의 이름 및 사용자/암호를 지정할 수 있습니다. (소스 데이터베이스 사용자/암호는 기존 -p 및 -u 옵션을 사용하여 지정할 수 있습니다.) USER/USING절은 선택적입니다. USER가 사용자 ID를 지정하는 경우 USING절 뒤에서 암호가 제공되거나, 지정되지 않는 경우 db2move가 암호 정보를 위해 프롬프트합니다. 프롬프트 이유는 아래에서 설명하는 보안 이유 때문입니다. TARGET\_DB는 COPY 조치에 대한 필수 옵션입니다. TARGET\_DB는 소스 데이터베이스와 같을 수 없습니다. ADMIN\_COPY\_SCHEMA 프로시저를 동일한 데이터베이스에서 스키마 복사에 사용할 수 있습니다. COPY 조치는 최소한 하나의 스키마(-sn) 또는 한 테이블(-tn 또는 -tf) 입력이 필요합니다.

다중 db2move 명령을 실행하여 한 데이터베이스에서 다른 데이터베이스로 스키마를 복사하면 교착 상태가 발생합니다. 한 번에 하나의 db2move 명령만 발행해야 합니다. 복사 처리 중에 소스 스키마의 테이블 변경은 목표 스키마의 데이터가 복사 후에 동일하지 않음을 의미할 수 있습니다.

**“MODE”**

**DDL\_AND\_LOAD**

소스 스키마로부터 지원되는 모든 오브젝트를 작성하고 테이블을 소스 테이블 데이터로 채웁니다. 디폴트 옵션입니다.

**DDL\_ONLY**

소스 스키마로부터 지원되는 모든 오브젝트를 작성하지만 테이블을 다시 채우지 않습니다.

**LOAD\_ONLY**

소스 데이터베이스에서 목표 데이터베이스로 모든 지정된 테이블을 로드합니다. 테이블은 이미 목표에 존재해야 합니다.

LOAD\_ONLY 모드는 -tn 또는 -tf 옵션을 사용하여 최소 하나의 테이블을 입력해야 합니다.

이것은 COPY 조치에서만 사용되는 선택적 옵션입니다.

#### “SCHEMA\_MAP”

사용자가 목표에 복사할 때 스키마 이름을 바꿀 수 있습니다. 소스-목표 스키마 맵핑 목록을 쉼표로 구분하고 대괄호로 묶어서 제공합니다. 예: schema\_map ((s1, t1), (s2, t2)). 이것은 스키마 s1의 오브젝트가 목표의 스키마 t1에 복사되며 스키마 s2의 오브젝트는 목표의 스키마 t2에 복사됨을 의미합니다. 디폴트이며 권장되는 목표 스키마 이름은 소스 스키마 이름입니다. 이유는 db2move는 오브젝트 본문 내의 모든 자격이 부여된 오브젝트에 대한 스키마를 수정하려고 하지 않기 때문입니다. 그러므로 다른 목표 스키마 이름을 사용하면 오브젝트 본문 안에 자격이 부여된 오브젝트가 있는 경우 문제가 발생할 수 있습니다.

예를 들어, 다음과 같습니다.

```
create view F00.v1 as 'select c1 from F00.t1'
```

이 경우 BAR로 스키마 FOO를 복사하면 다음과 같이 v1이 다시 생성됩니다.

```
create view BAR.v1 as 'select c1 from F00.t1'
```

이것은 스키마 FOO가 목표 데이터베이스에 존재하지 않으므로 실패하거나 FOO가 BAR과 다름으로 인해 예기치 않은 결과를 생성할 수 있습니다. 소스와 동일한 스키마 이름을 유지하면 이 문제를 피할 수 있습니다. 스키마 사이에 교차 종속성이 있는 경우 모든 상호 종속된 스키마가 복사되어야 하며, 그렇지 않으면 교차 종속성을 갖는 오브젝트를 복사할 때 오류가 발생할 수 있습니다.

예를 들어, 다음과 같습니다.

```
create view F00.v1 as 'select c1 from BAR.t1'
```

이 경우 v1의 복사는 BAR도 복사되지 않는 경우 실패하거나, 목표의 BAR이 소스의 BAR과 다른 경우 예기치 않은 결과를 생성합니다. db2move는 교차 스키마 종속성을 발견하려고 시도하지 않습니다.

이것은 COPY 조치에서만 사용되는 선택적 옵션입니다.

#### “NONRECOVERABLE”

이 옵션을 사용하면 사용자가 COPY-NO로 수행되는 로드의 디폴트 동작을 겹쳐쓸 수 있습니다. 디폴트 동작을 사용하면 사용자는 로드된 각 테이블 스페이스의 백업을 작성해야 합니다. 이 NONRECOVERABLE 키워드를 지정하면 사용자는 테이블 스페이스의 백업을 즉시 작성하지 않아도 됩니다. 그러나 새로 작성되는 테이블이 제대로 복구 가능하도록

록 가능한 빨리 백업을 작성하는 것이 매우 바람직합니다. 이것은 COPY 조치에 사용 가능한 선택적 옵션입니다.

#### “OWNER”

사용자가 성공적인 COPY 후에 목표 스키마에 작성된 각 새 오브젝트의 소유자를 변경할 수 있습니다. 목표 오브젝트의 디폴트 소유자는 연결된 사용자입니다. 이 옵션이 지정되면 소유권이 새 소유자에게 이전됩니다. 이것은 COPY 조치에 사용 가능한 선택적 옵션입니다.

#### “TABLESPACE\_MAP”

사용자는 복사 중에 소스 시스템의 테이블 스페이스 대신 사용될 테이블 스페이스 이름 매핑을 지정할 수 있습니다. 이것은 대괄호로 묶은 테이블 스페이스 매핑의 배열입니다. 예: `tablespace_map ((TS1, TS2),(TS3, TS4))`. 이것은 테이블 스페이스 TS1의 모든 오브젝트가 목표 데이터베이스의 테이블 스페이스 TS2에 복사되며 테이블 스페이스 TS3의 오브젝트는 목표의 테이블 스페이스 TS4에 복사됨을 의미합니다. ((T1, T2),(T2, T3))의 경우, 소스 데이터베이스의 T1에 있는 모든 오브젝트는 목표 데이터베이스의 T2에서 재작성되며 소스 데이터베이스의 T2에서 발견되는 모든 오브젝트는 목표 데이터베이스의 T3에 재작성됩니다. 디폴트는 소스에서와 동일한 테이블 스페이스 이름을 사용하는 것이며, 이 경우 이 테이블 스페이스에 대한 입력 매핑은 필요하지 않습니다. 지정된 테이블 스페이스가 존재하지 않는 경우 해당 테이블 스페이스를 사용하는 오브젝트의 복사는 실패하며 오류 파일에 로그됩니다.

사용자는 또한 `SYS_ANY` 키워드를 사용하여 목표 테이블 스페이스가 디폴트 테이블 스페이스 선택 알고리즘을 사용하여 선택되어야 함을 표시하는 옵션도 있습니다. 이 경우 `db2move`는 목표로 사용할 수 있는 모든 사용 가능한 테이블 스페이스를 선택할 수 있습니다. `SYS_ANY` 키워드를 모든 테이블 스페이스에 사용할 수 있습니다(예: `tablespace_map SYS_ANY`). 또한 사용자는 일부 테이블 스페이스에 대해 특정 매핑을 지정하고 나머지에 대해서는 디폴트 테이블 스페이스 선택 알고리즘을 지정할 수 있습니다. 예: `tablespace_map ((TS1, TS2),(TS3, TS4), SYS_ANY)`. 이것은 테이블 스페이스 TS1이 TS2로 매핑되고, TS3은 TS4에 매핑되지만 나머지 테이블 스페이스는 디폴트 테이블 스페이스 목표를 사용함을 표시합니다. "SYS"로 시작하는 테이블 스페이스를 가질 수 없으므로 `SYS_ANY` 키워드가 사용되고 있습니다.

이것은 COPY 조치에 사용 가능한 선택적 옵션입니다.

#### **-l lobpaths**

IMPORT 및 EXPORT의 경우 이 옵션이 지정되면 XML 경로에도 사용됩니다. 디폴트는 현재 디렉토리입니다.

이 옵션은 LOB 또는 XML 파일이 작성되거나(EXPORT의 파트로서) 검색되는(IMPORT 또는 LOAD의 파트로서) 절대 경로 이름을 지정합니다. 다중 경로를 지정할 때 각 경로는 쉼표로 분리되어야 하며 경로 사이에 공백은 허용되지 않습니다. 다중 경로가 지정되는 경우 EXPORT는 경로를 라운드 로빈 방식으로 사용합니다. 하나의 LOB 문서를 첫 번째 경로에 쓰고, 하나를 두 번째 경로에 쓰고, 이런 식으로 마지막까지 쓴 후 다시 첫 번째 경로로 돌아옵니다. XML 문서의 경우도 동일합니다. 파일이 첫 번째 경로에 없는 경우(IMPORT 또는 LOAD 중에) 두 번째 경로가 사용되는 방식입니다.

#### **-u userid**

디폴트는 로그인한 사용자 ID입니다.

사용자 ID 및 암호가 둘 다 선택적입니다. 그러나 하나가 지정되면 다른 것도 지정해야 합니다. 명령이 리모트 서버에 연결된 클라이언트에서 실행되는 경우 사용자 ID와 암호를 지정해야 합니다.

#### **-p password**

디폴트는 로그인된 암호입니다. 사용자 ID 및 암호가 둘 다 선택적입니다. 그러나 하나가 지정되면 다른 것도 지정해야 합니다. -p 옵션이 지정되지만 암호가 제공되지 않는 경우 db2move는 암호를 위해 프롬프트합니다. 이것은 보안상의 이유 때문입니다. 명령행을 통해 암호를 입력하면 보안 문제가 발생합니다. 예를 들어 ps -ef 명령은 암호를 표시합니다. 그러나 db2move가 스크립트를 통해 호출되면 암호를 제공해야 합니다. 명령이 리모트 서버에 연결된 클라이언트에서 실행되는 경우 사용자 ID와 암호를 지정해야 합니다.

**-aw** 경고를 허용합니다. -aw가 지정되지 않으면 익스포트 중에 경고가 발생하는 테이블은(테이블의 .ixf 파일 및 .msg 파일은 여전히 생성되지만) db2move.lst 파일에 포함되지 않습니다. 일부 시나리오(예: 데이터 절단)에서는 사용자가 그런 테이블이 db2move.lst 파일에 포함되기 원할 수 있습니다. 이 옵션을 지정하면 익스포트 중에 경고를 수신하는 테이블이 .lst 파일에 포함될 수 있습니다.

#### **예:**

- SAMPLE 데이터베이스의 모든 테이블을(모든 옵션에 대해 디폴트값을 사용하여) 익스포트하려면 다음 명령을 발행하십시오.

```
db2move sample export
```

- userid1 또는 사용자 ID LIKE us%rid2가 작성하고 이름이 tname1이거나 테이블 이름 LIKE %tname2를 갖는 모든 테이블을 익스포트하려면 다음 명령을 발행하십시오.

```
db2move sample export -tc userid1,us*rid2 -tn tname1,*tname2
```

- SAMPLE 데이터베이스의 모든 테이블을 임포트하려면(LOB 경로 D:\#LOBPATH1 및 C:\#LOBPATH2가 LOB 파일에 대해 검색되며, 이 예는 Windows 운영 체제에만 적용할 수 있음), 다음 명령을 발행하십시오.

```
db2move sample import -l D:\#LOBPATH1,C:\#LOBPATH2
```

- SAMPLE 데이터베이스의 모든 테이블을 로드하려면(/home/userid/lobpath 서버 디렉토리 및 tmp 서브디렉토리에서 LOB 파일을 검색하며, 이 예는 Linux 및 UNIX 시스템에만 적용할 수 있음) 다음 명령을 발행하십시오.

```
db2move sample load -l /home/userid/lobpath,/tmp
```

- REPLACE 모드에서 지정된 사용자 ID와 암호를 사용하여 SAMPLE 데이터베이스의 모든 테이블을 임포트하려면 다음 명령을 발행하십시오.

```
db2move sample import -io replace -u userid -p password
```

- 소스 데이터베이스 dbsrc에서 목표 데이터베이스 dbtgt로 schema1 스키마를 중복하려면 다음 명령을 발행하십시오.

```
db2move dbsrc COPY -sn schema1 -co TARGET_DB dbtgt USER myuser1 USING mypass1
```

- schema1 스키마를 소스 데이터베이스 dbsrc에서 목표 데이터베이스 dbtgt로 중복하고, 목표에서 스키마 이름을 newschema1로 변경하고 소스 테이블 스페이스 ts1을 목표의 ts2에 맵핑하려면 다음 명령을 발행하십시오.

```
db2move dbsrc COPY -sn schema1 -co TARGET_DB dbtgt USER myuser1 USING mypass1  
SCHEMA_MAP ((schema1,newschema1)) TABLESPACE_MAP ((ts1,ts2), SYS_ANY)
```

## 사용 시 참고사항

- db2move IMPORT/LOAD가 뒤따르는 db2move EXPORT는 테이블 데이터의 이동을 용이하게 합니다. 이들 테이블이 종속될 수 있는 오브젝트(예: 사용자 정의 유형 또는 사용자 정의 함수(UDF))뿐 아니라 테이블과 연관된 다른 모든 데이터베이스 오브젝트(예: 별명, 뷰 또는 트리거)를 수동으로 이동해야 합니다.
- CREATE 또는 REPLACE\_CREATE 옵션을 갖는 IMPORT 조치를 사용하여 목표 데이터베이스에 테이블을 작성하는 경우(두 옵션은 모두 사용되지 않으며 추후 릴리스에서 제거될 수 있음), 『임포트한 테이블 재작성』에서 설명하는 제한사항이 적용됩니다. REPLACE\_CREATE 옵션이 사용될 때 db2move 임포트 단계 중에 예기치 않은 오류가 발생하는 경우, 적합한 tabnnn.msg 메시지 파일을 조사하고 오류가 테이블 작성에 대한 제한사항의 결과일 수 있는지 고려하십시오.
- GENERATED ALWAYS ID 컬럼이 들어있는 테이블은 db2move를 사용하여 임포트하거나 로드할 수 없습니다. 그러나 이들 테이블을 수동으로 임포트 또는 로드할 수 있습니다. 자세한 정보는 『ID 컬럼 로드 고려사항』 또는 『ID 컬럼 임포트 고려사항』을 참조하십시오.
- db2move가 export, import 또는 load API를 호출할 때 **FileTypeMod** 매개변수는 lobsinfile로 설정됩니다. 즉, LOB 데이터는 모든 테이블에 대해 PC/IXF 파일과 별개의 파일에 보존됩니다.



- LOAD 명령은 데이터베이스 및 데이터 파일이 있는 머신에서 로컬로 실행되어야 합니다.
- db2move LOAD를 사용하고 데이터베이스에 대해 logretain이 사용될 때(데이터베이스가 복구 가능함),
  - NONRECOVERABLE 옵션이 지정되지 않으면 db2move는 디폴트 COPY NO 옵션을 사용하여 db2Load API를 호출하는 경우, 로드된 테이블이 있는 테이블 스페이스는 유틸리티 완료 시에 백업 보류 상태에 있습니다(테이블 스페이스를 백업 보류 상태에서 벗어나게 하려면 전체 데이터베이스 또는 테이블 스페이스 백업이 필요함).
  - NONRECOVERABLE 옵션이 지정되면 테이블 스페이스가 백업 보류 상태에 들어가지 않지만, 나중에 롤 포워드 복구가 수행되는 경우 테이블은 액세스 불가능으로 표시되며 삭제해야 합니다. 로드 복구 가능성 옵션에 대한 자세한 정보는 『로드 성능 개선을 위한 옵션』을 참조하십시오.
- IMPORT 또는 LOAD 옵션을 갖는 db2move 명령의 성능은 디폴트 버퍼 풀 IBMDEFAULTBP를 변경하고 구성 매개변수 **sortheap**, **util\_heap\_sz**, **logfilsiz** 및 **logprimary**를 갱신하여 개선될 수 있습니다.

**EXPORT를 사용할 때 필요한/생성되는 파일:**

- 입력: 없음
- 출력:

**EXPORT.out**

EXPORT 조치의 요약 결과.

**db2move.lst**

원래 테이블 이름, 대응하는 PC/IXF 파일 이름(tabnnn.ixf) 및 메시지 파일 이름(tabnnn.msg)의 목록입니다. 이 목록, 익스포트된 PC/IXF 파일 및 LOB 파일(tabnnnc.yyy)이 db2move IMPORT 또는 LOAD 조치의 입력으로 사용됩니다.

**tabnnn.ixf**

특정 테이블의 익스포트된 PC/IXF 파일.

**tabnnn.msg**

대응하는 테이블의 익스포트 메시지 파일.

**tabnnnc.yyy**

특정 테이블의 익스포트된 LOB 파일.

『nnn』은 테이블 번호입니다. 『c』는 영문자의 한 문자입니다. 『yyy』는 001 - 999 범위의 숫자입니다.

이들 파일은 익스포트되는 테이블에 LOB 데이터가 있는 경우에만 작성됩니다. 작성되는 경우 이들 LOB 파일은 『lobpath』 디렉토리에 위치합니다. LOB 파일에 대해 총 26,000개의 가능한 이름이 있습니다.

**system.msg**

파일 또는 디렉토리 작성 또는 삭제 명령에 대한 시스템 메시지가 들어있는 메시지 파일입니다. 이것은 조치가 EXPORT이고 LOB 경로가 지정되는 경우에만 사용됩니다.

**IMPORT를 사용할 때 필요한/생성되는 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 요약 결과.

**tabnnn.msg**

대응하는 테이블의 임포트 메시지 파일입니다.

**LOAD를 사용할 때 필요한/생성되는 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 요약 결과.

**tabnnn.msg**

대응하는 테이블의 LOAD 메시지 파일.

**COPY를 사용할 때 필요한/생성되는 파일:**

- 입력: 없음
- 출력:

#### **COPYSCHEMA.msg**

COPY 작업 중에 생성되는 메시지가 들어있는 출력 파일.

#### **COPYSCHEMA.err**

목표 데이터베이스에서 재작성될 수 없는 각 오브젝트에 대한 DDL문을 포함하여 COPY 작업 중에 발생하는 각 오류에 대한 오류 메시지가 들어있는 출력 파일.

#### **LOADTABLE.msg**

로드 유틸리티(목표 데이터베이스에서 데이터를 다시 채우는 데 사용되는)의 각 호출에 의해 생성되는 메시지가 들어있는 출력 파일.

#### **LOADTABLE.err**

로드 중에 실패했거나 목표 데이터베이스에서 채워져야 하는 테이블의 이름이 들어있는 출력 파일입니다. 자세한 내용은 『실패한 스키마 복사 조작 재시작』 주제를 참조하십시오.

이들 파일은 시간소인이 추가되며 한 번의 실행에서 생성되는 모든 파일은 동일한 시간소인을 갖습니다.



---

## 제 228 장 db2mqdsn - MQ 리스너

비동기 MQListener를 호출하여 WebSphere MQ 메시지 큐 세트를 모니터링하고 해당 메시지 큐에 도달하는 메시지를 구성된 DB2 스토어드 프로시저로 전달합니다. 연관된 관리 및 구성 태스크를 수행할 수도 있습니다. MQListener 구성 정보는 DB2 데이터베이스에 저장되며 디폴트값을 포함하여 이름 지정된 구성 세트로 구성됩니다. 각 구성은 태스크 세트로 구성됩니다. MQListener 태스크는 메시지를 검색하는 메시지 큐와 메시지가 전달될 스토어드 프로시저로 정의됩니다. 메시지 큐 설명에는 메시지 큐와 해당 큐 관리 프로그램의 이름이 포함되어야 합니다(디폴트값이 아닌 경우). 스토어드 프로시저에 대한 정보에는 프로시저가 정의된 데이터베이스, 데이터베이스에 액세스하기 위해 사용하는 사용자 이름과 암호 및 프로시저 이름과 스키마가 포함되어야 합니다.

Linux 및 UNIX 운영 체제의 경우 이 유틸리티는 DB2DIR/instance 디렉토리에 있습니다. 여기서 DB2DIR은 DB2 데이터베이스 제품의 현재 버전이 설치된 위치입니다.

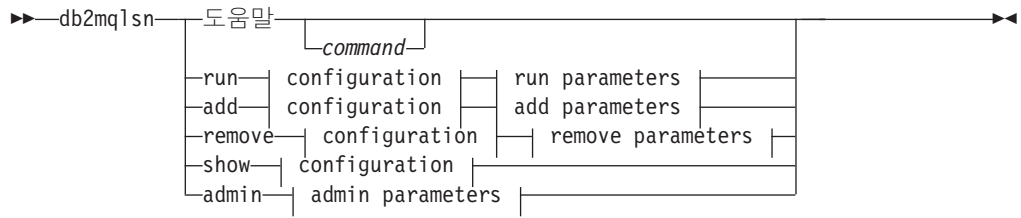
Windows 운영 체제의 경우 이 유틸리티는 DB2PATH\sqlib\bin 디렉토리에 있습니다. 여기서 DB2PATH는 DB2 데이터베이스 제품의 현재 버전이 설치된 위치입니다.

WebSphere MQ 오브젝트 액세스 제어에 대한 자세한 정보는 *WebSphere MQ 시스템 관리 안내서(SC34-6068-00)*를 참조하십시오.

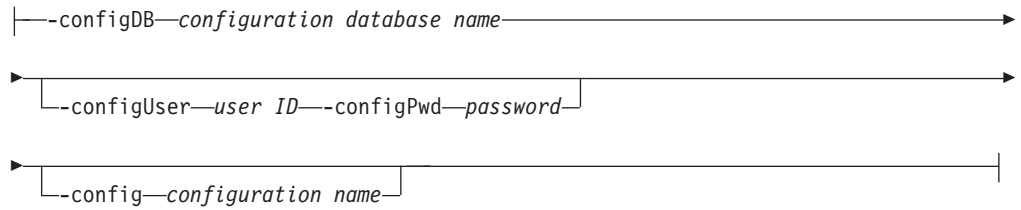
### 권한 부여

- db2mqdsn admin을 제외한 모든 옵션은 configDB 데이터베이스의 MQListener 구성에 액세스합니다. 연결은 configUser로서 작성되거나, 사용자가 지정되지 않으면 내재된 연결이 시도됩니다. 해당 이름으로 연결이 작성된 사용자에게는 mqlConf패키지에 대한 EXECUTE 특권이 필요합니다.
- db2mqdsn run 및 db2mqdsn admin 옵션으로 MQ 오브젝트에 액세스하려면 프로그램을 실행하는 사용자가 해당 MQ 오브젝트를 열 수 있어야 합니다.
- db2mqdsn run 옵션을 성공적으로 실행하려면 태스크를 작성한 db2mqdsn add 옵션에 지정된 dbUser에게 지정된 스토어드 프로시저에 대한 EXECUTE 특권이 있어야 하며 dbName 데이터베이스의 mqlRun 패키지에 대한 EXECUTE 특권이 있어야 합니다.

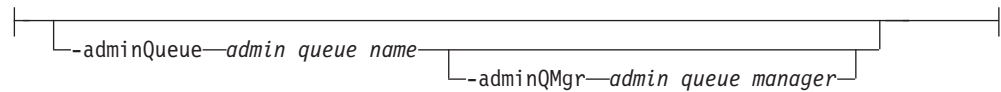
## 명령 구문



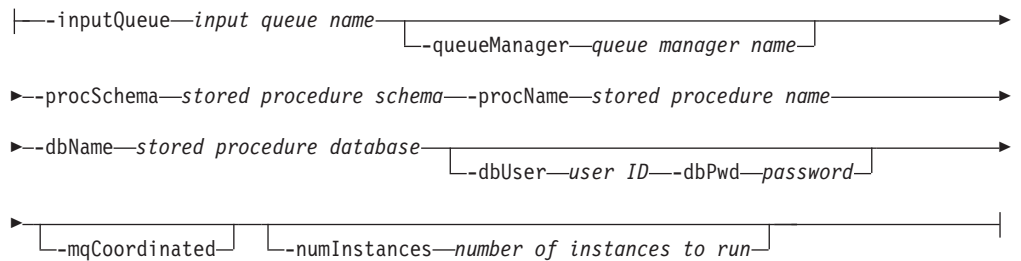
### configuration:



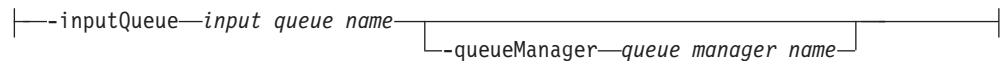
### run parameters:



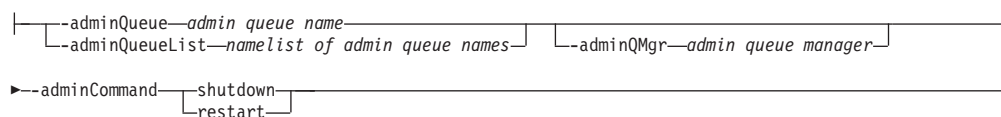
### add parameters:



### remove parameters:



### admin parameters:



## 명령 매개변수

### **help** *command*

특정 명령에 대한 세부사항 정보를 제공합니다. 명령 이름을 지정하지 않으면 일반 도움말 메시지가 표시됩니다.

### **-configDB** *configuration database*

구성 정보가 포함된 데이터베이스의 이름.

### **-configUser** *user ID* **-configPwd** *password*

구성 데이터베이스에 액세스하기 위한 권한 부여 정보.

### **-config** *configuration name*

개별 태스크를 구성으로 그룹화할 수 있습니다. 이를 통해 태스크 그룹을 함께 실행할 수 있습니다. 구성 이름을 지정하지 않으면 유틸리티가 디폴트 구성을 실행합니다.

## run

### **-adminQueue** *admin queue name* **-adminQMGr** *admin queue manager*

MQListener가 관리 명령을 대기(listen)하는 큐입니다. 큐 관리 프로그램을 지정하지 않으면 유틸리티가 구성된 디폴트 큐 관리 프로그램을 사용합니다. adminQueue를 지정하지 않으면 응용프로그램이 메시지 큐를 통해 관리 명령(예: shutdown 또는 restart)을 수신하지 않습니다.

## add

### **-inputQueue** *input queue name* **-queueManager** *queue manager name*

MQListener가 이 태스크의 메시지를 대기(listen)하는 큐입니다. 큐 관리 프로그램을 지정하지 않으면 유틸리티가 WebSphere MQ에 구성된 디폴트 큐 관리 프로그램을 사용합니다.

### **-procSchema** *stored procedure schema* **-procName** *stored procedure name*

MQListener가 도착한 메시지를 전달하는 스토어드 프로시저.

### **-dbName** *stored procedure database*

MQListener가 스토어드 프로시저로 메시지를 전달합니다. 스토어드 프로시저가 정의되는 데이터베이스입니다.

### **-dbUser** *user ID* **-dbPwd** *password*

대신 스토어드 프로시저가 호출되는 사용자.

### **-mqCoordinated**

WebSphere MQ 메시지 큐 읽기와 쓰기가 DB2 스토어드 프로시저 호출과 함께 트랜잭션으로 통합되어야 함을 나타냅니다. 전체 트랜잭션은 WebSphere MQ 코디네이터로 조정됩니다. 큐 관리 프로그램 또한 이러한 방식으로 트랜잭션을 조정하도록 구성되어야 합니다. 자세한 정보

는 WebSphere MQ 문서를 참조하십시오. 디폴트로, 메시지 큐 조작성은 스토어드 프로시저가 호출되는 트랜잭션의 일부가 아닙니다.

**-numInstances** *number of instances to run*

이 구성에서 실행할 이 태스크의 중복 인스턴스 수. 값을 지정하지 않으면 하나의 인스턴스만 실행됩니다.

## remove

**-inputQueue** *input queue name* **-queueManager** *queue manager name*

구성에서 제거될 태스크를 정의하는 큐 및 큐 관리 프로그램입니다. 입력 큐와 큐 관리 프로그램의 조합은 구성에서 고유합니다.

## admin

**-adminQueue** *admin queue name* **-adminQueueList** *namelist of admin queue names* **-adminQMGr** *admin queue manager*

관리 명령을 보낼 큐 이름의 이름 목록 또는 큐. 큐 관리 프로그램을 지정하지 않으면 유틸리티가 WebSphere MQ에 구성된 디폴트 큐 관리 프로그램을 사용합니다.

**-adminCommand** *admin command*

명령을 제출합니다. 명령은 shutdown 또는 restart입니다. shutdown을 실행하면 리스너가 현재 메시지 처리를 완료할 때 실행 중인 MQListener가 종료됩니다. restart는 종료를 수행한 다음 구성을 다시 읽고 재시작됩니다.

## 예:

```
db2mq1sn show -configDB sampleDB -config nightlies
```

```
db2mq1sn add -configDB sampleDB -config nightlies -inputQueue app3  
-procSchema imauser -procName proc3 -dbName aDB -dbUser imauser -dbPwd aSecret
```

```
db2mq1sn run -configDB -config nightlies
```



---

## 제 229 장 db2mscs - Windows 장애 복구 유틸리티 설정

MSCS(Microsoft Cluster Server)를 사용하여 Windows에서 DB2 장애 복구에 대한 인프라스트럭처를 작성합니다. 이 유틸리티를 사용하면 단일 파티션 및 파티션된 데이터베이스 환경 모두에서 장애 복구를 사용할 수 있습니다.

### 권한 부여

사용자는 MSCS 클러스터의 각 시스템의 Administrators 그룹에 속하는 도메인 사용자 어카운트에 로그인되어야 합니다.

### 명령 구문

```
db2mscs --f:--input_file --u:--instance_name
```

### 명령 매개변수

#### **-f:***input\_file*

MSCS 유틸리티가 사용할 입력 파일을 지정합니다. 이 매개변수가 지정되면 db2mscs 유틸리티는 파일 이름을 입력 파일로 사용하며, 이 매개변수가 지정되지 않으면 db2mscs 유틸리티는 현재 디렉토리에 있는 DB2MSCS.CFG 파일을 사용하려고 합니다.

#### **-u:***instance\_name*

이 옵션을 사용하면 db2mscs 조작을 실행 취소하고 인스턴스를 다시 *instance\_name*으로 지정되는 비MSCS 인스턴스로 되돌릴 수 있습니다.

### 사용 시 참고사항

db2mscs 유틸리티는 비MSCS 인스턴스를 MSCS 인스턴스로 변환하는 데 사용되는 독립형 명령행 유틸리티입니다. 유틸리티는 모든 MSCS 그룹, 자원 및 자원 종속성을 작성합니다. 또한 Windows 레지스트리에 저장된 모든 DB2 정보를 레지스트리의 클러스터 분할 영역에 복사하고 인스턴스 디렉토리를 공유 클러스터 디스크로 이동합니다. db2mscs 유틸리티는 클러스터가 설정될 방법을 지정하는 사용자가 제공하는 구성 파일을 입력으로 사용합니다. DB2MSCS.CFG 파일은 db2mscs 유틸리티가 읽는 매개변수가 들어있는 ASCII 텍스트 파일입니다. PARAMETER\_KEYWORD=parameter\_value 형식을 사용하여 별도의 라인에 각 입력 매개변수를 지정합니다. 예를 들면, 다음과 같습니다.

```
CLUSTER_NAME=FINANCE
GROUP_NAME=DB2 Group
IP_ADDRESS=9.21.22.89
```

두 예제 구성 파일은 DB2 install 디렉토리의 CFG 서브디렉토리에서 찾을 수 있습니다. 먼저 DB2MSCS.EE는 단일 파티션 데이터베이스 환경의 예입니다. 두 번째, DB2MSCS.EEE는 파티션된 데이터베이스 환경의 예입니다.

db2mscs 명령을 발행하기 전에 제어 센터가 사용 중이 아닌지 확인하십시오. 제어 센터가 사용 중이면 db2mscs 유틸리티는 오류를 갖고 종료합니다.

DB2MSCS.CFG 파일의 매개변수는 다음과 같습니다.

#### **DB2\_INSTANCE**

DB2 인스턴스의 이름입니다. 이 매개변수는 전역 범위를 갖고 DB2MSCS.CFG 파일에서 한 번만 지정되어야 합니다.

#### **DAS\_INSTANCE**

DB2 Admin Server 인스턴스의 이름입니다. MSCS 환경에서 실행하도록 DB2 Admin Server를 이주하려면 이 매개변수를 지정하십시오. 이 매개변수는 전역 범위를 갖고 DB2MSCS.CFG 파일에서 한 번만 지정되어야 합니다.

#### **CLUSTER\_NAME**

MSCS 클러스터의 이름입니다. 이 라인 뒤에 지정되는 모든 자원은 다른 CLUSTER\_NAME 매개변수가 지정될 때까지 이 클러스터에 작성됩니다.

#### **DB2\_LOGON\_USERNAME**

DB2 서비스에 대한 도메인 어카운트의 사용자 이름입니다(*domain#user*로 지정됨). 이 매개변수는 전역 범위를 갖고 DB2MSCS.CFG 파일에서 한 번만 지정되어야 합니다.

#### **DB2\_LOGON\_PASSWORD**

DB2 서비스에 대한 도메인 어카운트의 암호입니다. 이 매개변수는 전역 범위를 갖고 DB2MSCS.CFG 파일에서 한 번만 지정되어야 합니다.

#### **GROUP\_NAME**

MSCS 그룹의 이름입니다. 이 매개변수가 지정되면 새 MSCS 그룹이 존재하지 않는 경우 작성됩니다. 그룹이 이미 존재하는 경우 목표 그룹으로 사용됩니다. 이 매개변수 뒤에 지정되는 모든 MSCS 자원은 다른 GROUP\_NAME 매개변수가 지정될 때까지 이 그룹에 작성되거나 이 그룹으로 이동됩니다. 각 그룹에 대해 이 매개변수를 한 번 지정하십시오. MSCS 그룹은 MSCS 클러스터에 작성할 수 있습니다.

#### **DB2\_NODE**

현재 MSCS 그룹에 포함될 데이터베이스 파티션 서버(또는 데이터베이스 파티션)의 데이터베이스 파티션 번호입니다. 다중 논리적 데이터베이스 파티션이 동일한 머신에 존재하는 경우 각 데이터베이스 파티션은 별도의 DB2\_NODE 매개변수가 필요합니다. GROUP\_NAME 매개변수 뒤에 이 매개변수를 지정하여

DB2 자원이 올바른 MSCS 그룹에 작성되게 하십시오. 이 매개변수는 다중 파티션된 데이터베이스 환경에 필수입니다.

#### **IP\_NAME**

IP 주소 자원의 이름입니다. IP\_NAME의 값은 임의이지만, 클러스터에서 고유해야 합니다. 이 매개변수가 지정되면 IP 주소 유형의 MSCS 자원이 작성됩니다. 이 매개변수는 리모트 TCP/IP 연결에 필수입니다. 단일 파티션 데이터베이스 환경에서 이 매개변수는 선택적입니다. 권장 이름은 IP 주소에 대응하는 호스트 이름입니다.

#### **IP\_ADDRESS**

선행 IP\_NAME 매개변수로 지정되는 IP 자원의 TCP/IP 주소입니다. 이 매개변수는 IP\_NAME 매개변수가 지정되는 경우 필수입니다. 이것은 네트워크의 어떤 머신도 사용하지 않는 새 IP 주소입니다.

#### **IP\_SUBNET**

선행 IP\_NAME 매개변수로 지정되는 IP 자원의 TCP/IP 서브넷 마스크입니다. 이 매개변수는 IP\_NAME 매개변수가 지정되는 경우 필수입니다.

#### **IP\_NETWORK**

선행하는 IP 주소 자원이 속하는 MSCS 네트워크의 이름입니다. 이 매개변수는 선택적입니다. 지정되지 않는 경우 시스템이 발견하는 첫 번째 MSCS 네트워크가 사용됩니다. MSCS 네트워크의 이름은 클러스터 관리자의 네트워크 분기 아래에 표시되는 대로 정확하게 입력해야 합니다. 이전의 4 IP 키워드가 IP 주소 자원을 작성하는 데 사용됩니다.

#### **NETNAME\_NAME**

네트워크 이름 자원의 이름입니다. 네트워크 이름 자원을 작성하려면 이 매개변수를 지정하십시오. 이 매개변수는 단일 파티션 데이터베이스 환경의 경우 선택적입니다. 파티션된 데이터베이스 환경에서 DB2 인스턴스 디렉토리가 있는 인스턴스 소유 머신에 대해 이 매개변수를 지정해야 합니다.

#### **NETNAME\_VALUE**

네트워크 이름 자원에 대한 값입니다. NETNAME\_NAME 매개변수가 지정되는 경우 이 매개변수를 지정해야 합니다.

#### **NETNAME\_DEPENDENCY**

네트워크 이름 자원이 종속되는 IP 자원에 대한 이름입니다. 각 네트워크 이름 자원은 IP 주소 자원에 대한 종속성을 가져야 합니다. 이 매개변수는 선택적입니다. 지정되지 않는 경우 네트워크 이름 자원은 그룹의 첫 번째 IP 자원에 대한 종속성을 갖습니다.

#### **SERVICE\_DISPLAY\_NAME**

일반 서비스 자원의 표시 이름입니다. 일반 서비스 자원을 작성하려는 경우 이 매개변수를 지정하십시오.

### **SERVICE\_NAME**

일반 서비스 자원의 서비스 이름입니다. SERVICE\_DISPLAY\_NAME 매개변수가 지정되는 경우 이 매개변수를 지정해야 합니다.

### **SERVICE\_STARTUP**

일반 자원 서비스에 대한 선택적 시작 매개변수.

### **DISK\_NAME**

현재 그룹으로 이동될 실제 디스크 자원의 이름입니다. 필요한 만큼 많은 디스크 자원을 지정하십시오. 디스크 자원은 이미 존재해야 합니다. db2mcs 유틸리티가 장애 복구 지원을 위해 DB2 인스턴스를 구성할 때, 인스턴스 디렉토리가 그룹의 첫 번째 MSCS 디스크에 복사됩니다. 인스턴스 디렉토리에 대해 다른 MSCS 디스크를 지정하려면 INSTPROF\_DISK 매개변수를 사용하십시오. 사용되는 디스크 이름은 클러스터 관리자에 표시되는 대로 정확하게 입력해야 합니다.

### **INSTPROF\_DISK**

DB2 인스턴스 디렉토리를 포함할 MSCS 디스크를 지정하는 선택적 매개변수입니다. 이 매개변수가 지정되지 않는 경우 db2mcs 유틸리티는 동일한 그룹에 속하는 첫 번째 디스크를 사용합니다.

### **INSTPROF\_PATH**

인스턴스 디렉토리가 복사될 정확한 경로를 지정하는 선택적 매개변수입니다. 이 매개변수는 IPSHADisks, ServerRAID Netfinity® 디스크 자원(예: INSTPROF\_PATH=p:\db2profs)을 사용할 때 반드시 지정되어야 합니다. 둘 다 지정되면 INSTPROF\_PATH가 INSTPROF\_DISK에 우선권을 갖습니다.

### **TARGET\_DRVMAP\_DISK**

다중 파티션 데이터베이스 환경에 대한 데이터베이스 드라이브 맵핑에 대한 목표 MSCS 디스크를 지정하는 선택적 매개변수입니다. 이 매개변수는 데이터베이스 작성 명령이 지정하는 드라이브에서 맵핑하여 데이터베이스가 작성될 디스크를 지정합니다. 이 매개변수가 지정되지 않는 경우 데이터베이스 드라이브 맵핑은 db2drvmp 유틸리티를 사용하여 수동으로 등록되어야 합니다.

### **DB2\_FALLBACK**

DB2 자원이 오프라인이 될 때 응용프로그램이 강제로 해제되는지 여부를 제어하는 선택적 매개변수입니다. 지정되지 않는 경우 DB2\_FALLBACK의 설정은 YES입니다. 응용프로그램이 강제로 해제되지 않기를 원하는 경우 DB2\_FALLBACK을 NO로 설정하십시오.

---

## 제 230 장 db2mtrk - 메모리 추적 프로그램

메모리 상태의 전체 보고서(예: 데이터베이스, 에이전트 및 응용프로그램)를 제공합니다. 이 명령은 다음 메모리 풀 할당 정보를 출력합니다.

- 현재 크기
- 최대 크기(하드 한계)
- 가장 큰 크기(상위 워터 마크(water mark))
- 유형(메모리가 사용될 함수를 표시하는 ID)
- 풀을 할당한 에이전트(풀이 개인용인 경우에만)
- 응용프로그램

스냅샷 모니터에서 동일한 정보를 사용할 수도 있습니다.

### 범위

파티션된 데이터베이스 환경의 경우 이 명령은 db2nodes.cfg 파일에 정의된 데이터베이스 파티션에서 호출할 수 있습니다. 인스턴스 레벨 정보가 리턴되는 경우를 제외하고, 명령은 해당 데이터베이스 파티션에 대해서만 정보를 리턴합니다. 이 명령은 리모트 서버에 대한 정보는 리턴하지 않습니다.

### 권한 부여

다음 중 하나가 필요합니다.

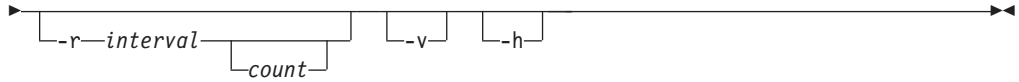
- *sysadm*
- *sysctrl*
- *sysmaint*
- *sysmon*

### 필요한 연결

인스턴스. 인스턴스 첨부이 없는 경우 응용프로그램이 디폴트 인스턴스 첨부을 작성합니다.

### 명령 구문

```
db2mtrk [-i] [-d] [-p] [--m] [--a] [-w]
```



## 명령 매개변수

- i** 인스턴스 레벨 메모리를 표시합니다.
- d** 데이터베이스 레벨 메모리를 표시합니다.
- a** 응용프로그램 메모리 사용량을 표시합니다.
- p** 사용되지 않습니다. 전용 메모리를 표시합니다.  
응용프로그램 메모리 사용량을 표시하기 위해 **-a** 매개변수로 교체되었습니다.
- m** 각 풀의 최대값을 표시합니다.
- w** 각 풀의 상위 워터 마크(water mark) 값을 표시합니다.
- r** 반복 모드

### *interval*

메모리 추적 프로그램에 대한 연속 호출 사이에 기다릴 초 수(반복 모드에서).

*count* 반복할 횟수.

- v** 상세 출력.
- h** 도움말 화면을 표시합니다. **-h**를 지정하는 경우 도움말 화면만 표시됩니다. 다른 정보는 표시되지 않습니다.

## 예:

다음 호출은 데이터베이스 및 인스턴스 일반 값을 리턴하고 10초마다 반복합니다.

```
db2mtrk -i -d -v -r 10
```

다음 출력 샘플을 고려하십시오.

명령 `db2mtrk -i -d`는 다음 출력을 표시합니다.

Tracking Memory on: 2006/01/17 at 15:24:38

Memory for instance

monh	other
576.0K	8.0M

Memory for database: AJSTORM

utilh	pckcacheh	catcacheh	bph (1)	bph (S32K)	bph (S16K)	bph (S8K)
64.0K	640.0K	128.0K	34.2M	576.0K	320.0K	192.0K
bph (S4K)	shsorth	lockh	dbh	apph (13)	appshrh	
128.0K	64.0K	9.6M	4.8M	64.0K	256.0K	

Memory for database: CMGARCIA

```

utilh      pckcacheh  catcacheh  bph (1)  bph (S32K)  bph (S16K)  bph (S8K)
64.0K      640.0K     128.0K     34.2M    576.0K      320.0K      192.0K

bph (S4K)  shsorth    lockh      dbh       apph (26)   appshrh
128.0K     64.0K     9.6M      4.8M     64.0K      256.0K

```

명령 db2mtrk -a -i -d는 다음 출력을 표시합니다.

Tracking Memory on: 2007/01/15 at 11:30:38

Memory for instance

```

other      monh      fcmbp
11.5M     64.0K    640.0K

```

Memory for database: SAMPLE

```

utilh      pckcacheh  other      catcacheh  bph (1)  bph (S32K)  bph (S16K)
64.0K      1.0M      576.0K     448.0K     1.3M     832.0K      576.0K

bph (S8K)  bph (S4K)  shsorth    lockh      dbh       apph (12)  apph (11)
448.0K     384.0K    192.0K     320.0K     10.4M    64.0K      64.0K

apph (10)  apph (9)   apph (8)
64.0K     64.0K     64.0K

```

Application Memory for database: SAMPLE

```

appshrh
256.0K

```

Memory for application 11

```

apph      other
64.0K     64.0K

```

Memory for application 10

```

apph      other
64.0K     64.0K

```

Memory for application 9

```

apph      other
64.0K     64.0K

```

Memory for application 8

```

apph      other
64.0K     448.0K

```

명령 db2mtrk -a -v -i -d는 다음 출력을 표시합니다.

Tracking Memory on: 2007/01/15 at 11:22:56

Memory for instance

```

Other Memory is of size 12058624 bytes
Database Monitor Heap is of size 65536 bytes
FCMBP Heap is of size 655360 bytes
Total: 12779520 bytes

```

Memory for database: SAMPLE

```

Backup/Restore/Util Heap is of size 65536 bytes

```

Package Cache is of size 1048576 bytes  
Other Memory is of size 589824 bytes  
Catalog Cache Heap is of size 458752 bytes  
Buffer Pool Heap (1) is of size 1376256 bytes  
Buffer Pool Heap (System 32k buffer pool) is of size 851968 bytes  
Buffer Pool Heap (System 16k buffer pool) is of size 589824 bytes  
Buffer Pool Heap (System 8k buffer pool) is of size 458752 bytes  
Buffer Pool Heap (System 4k buffer pool) is of size 393216 bytes  
Shared Sort Heap is of size 196608 bytes  
Lock Manager Heap is of size 327680 bytes  
Database Heap is of size 10944512 bytes  
Application Heap (12) is of size 65536 bytes  
Application Heap (11) is of size 65536 bytes  
Application Heap (10) is of size 65536 bytes  
Application Heap (9) is of size 65536 bytes  
Application Heap (8) is of size 65536 bytes  
Applications Shared Heap is of size 524288 bytes  
Total: 18153472 bytes

Application Memory for database: SAMPLE

Applications Shared Heap is of size 524288 bytes  
Total: 524288 bytes

Memory for application 11

Application Heap is of size 65536 bytes  
Other Memory is of size 65536 bytes  
Total: 131072 bytes

Memory for application 10

Application Heap is of size 65536 bytes  
Other Memory is of size 65536 bytes  
Total: 131072 bytes

Memory for application 9

Application Heap is of size 65536 bytes  
Other Memory is of size 65536 bytes  
Total: 131072 bytes

Memory for application 8

Application Heap is of size 65536 bytes  
Other Memory is of size 458752 bytes  
Total: 524288 bytes

Total: 1441792 bytes

## 사용 시 참고사항

주:

1. 플래그를 지정하지 않은 경우 사용량이 리턴됩니다.
2. -d, -h, -i, -p 또는 -a 플래그 중 하나를 지정해야 합니다.



3. -p 매개변수를 지정한 경우, 자세한 전용 메모리 사용량 정보가 에이전트 ID별로 그룹화되어 리턴됩니다.
4. -p 매개변수를 지정한 경우, 자세한 응용프로그램 메모리 사용량 정보가 응용프로그램 ID별로 그룹화되어 리턴됩니다.
5. 보고된 "기타 메모리"는 데이터베이스 관리 시스템 작동의 오버헤드와 연관되는 메모리입니다.
6. 일부 경우(예: 패키지 캐시)에, 표시되는 최대 크기는 구성 매개변수에 지정된 값보다 크게 됩니다. 이와 같은 경우, 구성 매개변수에 지정된 값은 '소프트 한계'로 사용되며, 풀의 실제 메모리 사용량은 구성된 크기를 초과하여 커질 수 있습니다.
7. 버퍼 풀 힙의 경우, 괄호에 지정된 숫자는 버퍼 풀 ID이거나, 이 버퍼 풀이 시스템 버퍼 풀 중 하나임을 표시합니다.
8. 응용프로그램 힙의 경우 괄호에 지정된 숫자는 응용프로그램 ID입니다.
9. 일부 힙에 대해 메모리 추적 프로그램이 보고하는 최대 크기는 머신의 실제 메모리 양입니다. 이 힙을 바운드되지 않은 힙이라고 하며 무제한 최대 크기로 선언됩니다. 힙이 선언될 때 최대 활동 시간에 필요할 메모리 양이 명확하지 않기 때문입니다. 이러한 힙이 머신의 실제 메모리에 의해 엄격하게 바운드되지 않아도, 적절한 근사치이므로 최대 크기로 보고됩니다.



---

## 제 231 장 db2nchg - 데이터베이스 파티션 서버 구성 변경

데이터베이스 파티션 서버 구성을 수정합니다. 여기에는 한 머신에서 다른 머신으로 데이터베이스 파티션 서버(노드) 이동, 머신의 TCP/IP 호스트 이름 변경 및 데이터베이스 파티션 서버(노드)에 대한 다른 논리적 포트 번호 또는 다른 네트워크 이름 선택이 포함됩니다. 이 명령은 데이터베이스 파티션 서버가 중지되는 경우에만 사용할 수 있습니다.

이 명령은 Windows 운영 체제에서만 사용 가능합니다.

### 권한 부여

로컬 관리자

### 명령 구문

```
db2nchg -/n:—dbpartitionnum —————→
                                     /i:—instance_name
-----→
/u:—username,password — /p:—logical_port — /h:—host_name
-----→
/m:—machine_name — /g:—network_name -----→
```

### 명령 매개변수

#### **/n:dbpartitionnum**

변경될 데이터베이스 파티션 서버의 구성의 데이터베이스 파티션 번호를 지정합니다.

#### **/i:instance\_name**

이 데이터베이스 파티션 서버가 참여하는 인스턴스를 지정합니다. 매개변수가 지정되지 않는 경우 디폴트는 현재 인스턴스입니다.

#### **/u:username,password**

사용자 이름 및 암호를 지정합니다. 매개변수가 지정되지 않는 경우 기존 사용자 이름과 암호가 적용됩니다.

#### **/p:logical\_port**

데이터베이스 파티션 서버에 대한 논리적 포트를 지정합니다. 데이터베이스 파티션 서버를 다른 머신으로 이동하려면 이 매개변수를 지정해야 합니다. 매개변수가 지정되지 않는 경우 논리적 포트 번호는 변경되지 않습니다.

**/h:host\_name**

FCM이 내부 통신에 사용하는 TCP/IP 호스트 이름을 지정합니다. 이 매개변수가 지정되지 않는 경우 호스트 이름은 동일하게 남습니다.

**/m:machine\_name**

데이터베이스 파티션 서버가 상주할 머신을 지정합니다. 데이터베이스 파티션 서버는 인스턴스에 기존 데이터베이스가 없는 경우에만 이동될 수 있습니다.

**/g:network\_name**

데이터베이스 파티션 서버에 대한 네트워크 이름을 변경합니다. 이 매개변수를 사용하여 머신에 다중 IP 주소가 있을 때 데이터베이스 파티션 서버에 특정 IP 주소를 적용할 수 있습니다. 네트워크 이름 또는 IP 주소를 입력할 수 있습니다.

**예:**

인스턴스 TESTMPP에 참여하는 데이터베이스 파티션 2에 지정된 논리적 포트를 논리적 포트 3으로 변경하려면 다음 명령을 입력하십시오.

```
db2nchg /n:2 /i:TESTMPP /p:3
```

---

## 제 232 장 db2ncrt - 인스턴스에 데이터베이스 파티션 서버 추가

인스턴스에 데이터베이스 파티션 서버(노드)를 추가합니다.

이 명령은 Windows 운영 체제에서만 사용할 수 있습니다.

### 범위

데이터베이스 파티션 서버가 인스턴스가 이미 존재하는 컴퓨터에 추가되면 데이터베이스 파티션 서버는 논리적 데이터베이스 파티션 서버로서 컴퓨터에 추가됩니다. 데이터베이스 파티션 서버가 인스턴스가 존재하지 않는 컴퓨터에 추가되는 경우 인스턴스가 추가되고 컴퓨터는 새 실제 데이터베이스 파티션 서버가 됩니다. 인스턴스에 데이터베이스가 있는 경우 이 명령을 사용하지 않아야 합니다. 대신 START DATABASE MANAGER 명령이 ADD DBPARTITIONNUM 옵션과 함께 발행되어야 합니다. 이것은 데이터베이스가 새 데이터베이스 파티션 서버에 올바르게 추가되도록 보장합니다. 또한 데이터베이스가 작성된 인스턴스에 데이터베이스 파티션 서버를 추가할 수 있습니다. 파일을 변경하면 파티션된 데이터베이스 환경에서 불일치를 유발할 수 있으므로 db2nodes.cfg 파일은 편집해서는 안 됩니다.

### 권한 부여

새 데이터베이스 파티션 서버가 추가되는 컴퓨터의 로컬 관리자 권한.

### 명령 구문

```
►► db2ncrt -/n:—dbpartitionnum—/u:—username,password—►►
    [ /i:—instance_name ] [ /m:—machine_name ] [ /p:—logical_port ]
    [ /h:—host_name ] [ /g:—network_name ] [ /o:—instance_owning_machine ]
```

### 명령 매개변수

#### */n:dbpartitionnum*

데이터베이스 파티션 서버를 식별하는 고유한 데이터베이스 파티션 번호입니다. 입력되는 번호는 1 - 999 범위에 있을 수 있습니다.

#### */u:username,password*

DB2에 대한 로그인 어카운트 이름 및 암호를 지정합니다.

#### */i:instance\_name*

인스턴스 이름을 지정합니다. 매개변수가 지정되지 않는 경우 디폴트는 현재 인스턴스입니다.

### */m:machine\_name*

데이터베이스 파티션 서버가 있는 Windows 워크스테이션의 컴퓨터 이름을 지정합니다. 데이터베이스 파티션 서버가 리모트 컴퓨터에 추가되는 경우 이 매개 변수는 필수입니다.

### */p:logical\_port*

데이터베이스 파티션 서버에 사용되는 논리적 포트 번호를 지정합니다. 이 매개 변수를 지정하지 않으면 지정되는 논리적 포트 번호는 0입니다. 논리적 데이터베이스 파티션 서버를 작성할 때 이 매개 변수를 지정해야 하며 사용 중인 아닌 논리적 포트 번호를 선택해야 합니다. 다음 제한사항을 주의하십시오.

- 모든 컴퓨터는 논리적 포트가 0인 데이터베이스 파티션 서버를 가져야 합니다.
- 포트 번호는 `x:\winnt\system32\drivers\etc\` 디렉토리에서 FCM 통신에 대해 보존되는 포트 범위를 초과하지 않아야 합니다. 예를 들어 4 포트의 범위가 현재 인스턴스에 대해 보존되는 경우 최대 포트 번호는 3입니다. 포트 0은 디폴트 논리적 데이터베이스 파티션 서버에 사용됩니다.

### */h:host\_name*

FCM이 내부 통신에 사용하는 TCP/IP 호스트 이름을 지정합니다. 데이터베이스 파티션 서버가 리모트 컴퓨터에 추가될 때 이 매개 변수는 필수입니다.

### */g:network\_name*

데이터베이스 파티션 서버의 네트워크 이름을 지정합니다. 매개 변수가 지정되지 않는 경우 시스템에서 발견되는 첫 번째 IP 주소가 사용됩니다. 이 매개 변수를 사용하여 컴퓨터에 다중 IP 주소가 있을 때 데이터베이스 파티션 서버에 특정 IP 주소를 적용할 수 있습니다. 네트워크 이름 또는 IP 주소를 입력할 수 있습니다.

### */o:instance\_owning\_machine*

인스턴스 소유 컴퓨터의 컴퓨터 이름을 지정합니다. 디폴트는 로컬 컴퓨터입니다. 이 매개 변수는 인스턴스 소유 컴퓨터가 아닌 모든 컴퓨터에서 `db2ncrt` 명령이 호출될 때 필수입니다.

## 예:

새 데이터베이스 파티션 서버가 데이터베이스 파티션 2로 알려지고 논리적 포트 1을 사용하는 인스턴스 소유 컴퓨터 SHAYER의 TESTMPP 인스턴스에 새 데이터베이스 파티션 서버를 추가하려면 다음 명령을 입력하십시오.

```
db2ncrt /n:2 /u:QBPAULZ\paulz,g1reeky /i:TESTMPP /m:TEST /p:1 /o:SHAYER /h:TEST
```

---

## 제 233 장 db2ndrop - 인스턴스에서 데이터베이스 파티션 서버 삭제

데이터베이스가 없는 인스턴스에서 데이터베이스 파티션 서버(노드)를 삭제(drop)합니다. 데이터베이스 파티션 서버가 삭제되는 경우 데이터베이스 파티션 번호가 새 데이터베이스 파티션 서버에 재사용될 수 있습니다. 이 명령은 데이터베이스 파티션 서버가 중지되는 경우에만 사용할 수 있습니다.

이 명령은 Windows 운영 체제에서만 사용 가능합니다.

### 권한 부여

데이터베이스 파티션 서버가 삭제될 머신의 로컬 관리자 권한.

### 명령 구문

```
db2ndrop /n:dbpartitionnum [/i:instance_name]
```

### 명령 매개변수

*/n:dbpartitionnum*

데이터베이스 파티션 서버를 식별하는 고유한 데이터베이스 파티션 번호입니다.

*/i:instance\_name*

인스턴스 이름을 지정합니다. 매개변수가 지정되지 않는 경우 디폴트는 현재 인스턴스입니다.

### 예:

```
db2ndrop /n:2 /i=KMASCI
```

### 사용 시 참고사항

인스턴스 소유 데이터베이스 파티션 서버(*dbpartitionnum* 0)가 인스턴스에서 삭제되는 경우 인스턴스는 사용 불가능하게 됩니다. 인스턴스를 삭제하려면 *db2idrop* 명령을 사용하십시오.

이 인스턴스에 데이터베이스가 있는 경우 이 명령을 사용해서는 안됩니다. 대신 *db2stop drop nodenum* 명령을 사용해야 합니다. 이것은 데이터베이스 파티션 서버가 파티션 데이터베이스 환경에서 올바르게 제거되도록 합니다. 또한 데이터베이스가 존재하는 인스턴스에서 데이터베이스 파티션 서버를 삭제할 수도 있습니다. 파일을 변경하면 파티션된 데이터베이스 환경에서 불일치를 유발할 수 있으므로 *db2nodes.cfg* 파일은 편집해서는 안됩니다.

다중 논리적 데이터베이스 파티션 서버를 실행 중인 머신에서 논리적 포트 0에 지정된 데이터베이스 파티션 서버를 삭제하려면, 다른 논리적 포트에 지정된 다른 모든 데이터베이스 파티션 서버가 먼저 삭제되어야 합니다. 각 데이터베이스 파티션 서버는 논리적 포트 0에 지정된 데이터베이스 파티션 서버가 있어야 합니다.



---

## 제 234 장 db2nrcfg - 루트 서버가 아닌 설치 구성 도구 명령

DB2 루트 서버가 아닌 설치에 사용되는 구성 도구입니다.

### 권한 부여

루트 서버가 아닌 설치를 소유하는 루트 서버가 아닌 ID.

### 필요한 연결

없음

### 명령 구문

```
db2nrcfg -a AuthType -d -p PortName -s InstType -h | -?
```

### 명령 매개변수

#### -a *AuthType*

인스턴스의 인증 유형(SERVER, CLIENT 또는 SERVER\_ENCRYPT)을 설정합니다.

-d 디버그 모드를 활성화합니다.

#### -p *PortName*

이 인스턴스가 사용할 포트 이름 또는 포트 번호를 설정합니다.

#### -s *InstType*

작성될 인스턴스 유형(wse, ese 또는 client)을 설정합니다.

#### -h | -?

도움말 정보를 표시합니다.

### 사용 시 참고사항

이 명령은 루트 서버가 아닌 설치 중에 DB2 설치 프로그램이 자동으로 실행합니다.

db2icrt, db2iupdt 및 db2iupgrade 명령은 루트 설치에 사용되며 루트 서버가 아닌 설치에서는 사용할 수 없습니다.



---

## 제 235 장 db2rfe - 루트 서버가 아닌 설치에 루트 기능 사용 명령

이 명령은 구성 파일에 따라 DB2를 루트 서버가 아닌 서버에 설치할 때 지원되는 루트 기능을 사용합니다. DB2 루트 서버가 아닌 인스턴스는 db2rfe 명령이 실행되기 전에 중지해야 합니다.

### 권한 부여

루트 특권이 있는 사용자.

### 필요한 연결

없음

### 명령 구문

```
►► db2rfe -f db2rfe_config_file -h  
-?
```

### 명령 매개변수

**-f** *db2rfe\_config\_file*

루트 기능을 사용하는 데 사용되는 구성 파일을 지정합니다.

**-h | -?**

도움말 정보를 표시합니다.

### 사용 시 참고사항

구성 파일의 각 루트 기능은 별도 섹션에 있습니다. 각 섹션에는 시작 및 끝 표시, 사용될 섹션을 설명하는 주석 및 루트 기능을 사용하는 명령이 포함됩니다. 샘플 구성 파일 db2rfe.cfg는 \$DB2DIR/instance 디렉토리에 설치됩니다.

샘플 구성 파일은 다음과 유사합니다(이 예에서 루트 서버가 아닌 설치 소유자는 db2inst3임).

```
** =====  
**  
** Sample configuration file for db2rfe of IBM DB2  
** -----  
**  
** To select features and settings to configure, uncomment the corresponding  
** keywords and specify values for those keywords.  
**  
** Comments are made by placing either an asterisk (*) or a number sign (#) at  
** the start of a line  
**  
** =====
```

```

INSTANCENAME=db2inst3
** This is required keyword.

** -----
** Set hard/soft data ulimit to unlimited, and hard/soft nofile ulimit to 65536.
**
** Note: This is for AIX only. On other platforms, refer to system documentation
** to set it manually.
** -----
** Valid value is NO and YES. Change to YES if you need to set the ulimit.

SET_ULIMIT=NO

** -----
** Enable DB2 High Availability (HA) feature
** -----
** Valid value is NO and YES. Change to YES if you need to enable this feature.

ENABLE_HA=NO

** -----
** ENABLE DB2 Authentication on the server using local operating system security.
** -----
** Valid value is NO and YES. Change to YES if you need to enable this feature.

ENABLE_OS_AUTHENTICATION=NO

** -----
** Reserve DB2 remote connection service entry
** -----
** Valid value is NO and YES. Change to YES if you need to enable this feature.

RESERVE_REMOTE_CONNECTION=NO

*SVCENAME=db2c_db2inst3
** char(14)

*SVCEPORT=48000
** Valid value: 1024 - 65535

** -----
** Reserve DB2 text search service entry
** -----
** Valid value is NO and YES. Change to YES if you need to enable this feature.

RESERVE_TEXT_SEARCH_CONNECTION=NO

*SVCENAME_TEXT_SEARCH=db2j_db2inst3
** char(14)

*SVCEPORT_TEXT_SEARCH=55000
** Valid value: 1024 - 65535

```

---

## 제 236 장 db2nrupdt - 루트 서버가 아닌 설치 인스턴스 갱신 명령

DB2의 루트 서버가 아닌 설치로 작성된 인스턴스에 사용되는 갱신 도구입니다.

### 권한 부여

루트 서버가 아닌 설치를 소유하는 루트 서버가 아닌 ID.

### 필요한 연결

없음

### 명령 구문

```
▶▶ db2nrupdt [-a AuthType] [-d] [-k] [-h | -?] ▶▶▶
```

### 명령 매개변수

#### -a *AuthType*

인스턴스의 인증 유형(SERVER, CLIENT 또는 SERVER\_ENCRYPT)을 설정합니다.

-d 디버그 모드를 활성화합니다.

#### -k

갱신 중에 현재 인스턴스 유형을 유지합니다.

#### -h | -?

도움말 정보를 표시합니다.

### 사용 시 참고사항

db2icrt, db2iupdt 및 db2iupgrade 명령은 루트 설치에 사용되며 루트 서버가 아닌 설치에서는 사용할 수 없습니다.



---

## 제 237 장 db2nrupgrade - 루트 서버가 아닌 인스턴스 업그레이드 명령

이 명령은 Linux 및 UNIX 시스템에서만 사용할 수 있습니다.

db2nrupgrade 명령은 DB2 데이터베이스 시스템의 이전 버전에서 db2nrupgrade 명령을 실행하는 DB2 사본의 현재 버전으로 루트 서버가 아닌 인스턴스를 업그레이드합니다.

이 명령은 DB2DIR/instance 디렉토리에 있습니다. 여기서 DB2DIR은 DB2 데이터베이스 시스템의 새 릴리스가 설치된 설치 위치를 나타냅니다. 이 명령은 루트 설치에 대한 인스턴스 업그레이드를 지원하지 않습니다.

### 권한 부여

루트 서버가 아닌 설치 사본을 소유하는 루트 서버가 아닌 ID.

### 명령 구문

```
▶▶ db2nrupgrade -d -a AuthType -b backup_dir ▶▶
```

### 명령 매개변수

#### Linux 및 UNIX 시스템용

**-d** 디버그 모드를 활성화합니다. 이 옵션은 DB2 Support의 지시가 있는 경우에만 사용합니다.

**-a AuthType**

인스턴스의 인증 유형(SERVER, CLIENT 또는 SERVER\_ENCRYPT)을 지정합니다. 디폴트값은 SERVER입니다.

**-b backup\_dir**

이 매개변수는 필수 매개변수입니다. 이전 DB2 버전의 구성 파일이 저장되는 디렉토리를 지정합니다.

### 사용 시 참고사항

- 이 명령은 사본을 업그레이드할 때 자동으로 실행됩니다. 사본 업그레이드에 실패하지 않으면 이 명령을 수동으로 실행하지 않아도 됩니다.





---

## 제 238 장 db2osconf - 커널 매개변수 값 유틸리티

시스템 크기를 기반으로 커널 매개변수 값을 권장합니다. 권장값은 가장 합리적인 워크로드를 수용할 수 있을 정도로 지정된 시스템에 충분합니다. 이 명령은 현재 64비트 인스턴스 HP-UX의 DB2와 Solaris 운영 체제에만 사용할 수 있습니다.

### 권한 부여

- DB2 for HP-UX에서는 권한 부여가 필요하지 않습니다. db2osconf 유틸리티의 권한대로 변경하려면 루트 액세스 권한이 필요합니다.
- DB2 for the Solaris 운영 체제의 경우에는 루트 액세스 권한이 있거나 sys 그룹의 구성원이어야 합니다.

### 명령 구문

현재 지원되는 옵션의 목록을 가져오려면 db2osconf -h를 입력하십시오.

```
db2osconf -h
Usage:
-c                # Client only
-f                # Compare to current
-h                # Help screen
-l                # List current
-m <mem in GB>   # Specify memory in GB
-n <num CPUs>    # Specify number of CPUs
-p <perf level>  # Msg Q performance level (0-3)
-s <scale factor> # Scale factor (1-3)
-t <threads>     # Number of threads
```

### 명령 매개변수

- c -c 스위치는 클라이언트 설치 전용입니다. 이 옵션은 DB2 for the Solaris 운영 체제에서만 사용할 수 있습니다.
- f 현재 커널 매개변수와 db2osconf 유틸리티가 권장하는 값을 비교하는 데 사용됩니다. -f 옵션은 db2osconf 명령으로 다른 옵션이 입력되지 않는 경우 디폴트값입니다. Solaris 운영 체제의 경우 다른 커널 매개변수만 표시됩니다. 현재 커널 매개변수는 활성 커널에서 직접 가져오므로 Solaris 시스템 지정 파일인 /etc/system의 매개변수와 일치하지 않을 수 있습니다. 활성 커널의 커널 매개변수가 /etc/system에 나열된 매개변수와 다른 경우 재부트하지 않고 /etc/system 파일이 변경되었거나 파일에 구문 오류가 있을 수 있습니다. HP-UX의 경우 -f 옵션은 권장 매개변수 목록과 권장 매개변수 값 변경사항을 리턴합니다.

\*\*\*\*\* Please Change the Following in the Given Order \*\*\*\*\*

WARNING [<parameter name>] should be set to <value>

- l 현재 커널 매개변수를 나열합니다.
- m 실제 메모리 크기(GB)를 겹쳐줍니다. 일반적으로 db2osconf 유틸리티는 실제 메모리의 크기를 자동으로 판별합니다. 이 옵션은 DB2 for the Solaris 운영 체제에서만 사용할 수 있습니다.
- n 시스템의 CPU 수를 겹쳐줍니다. 일반적으로 db2osconf 유틸리티는 CPU 수를 자동으로 판별합니다. 이 옵션은 DB2 for the Solaris 운영 체제에서만 사용할 수 있습니다.
- p SYSV 메시지 큐의 성능 레벨을 설정합니다. 디폴트값은 0이고 가장 높은 설정값은 3입니다. 이 값을 더 높게 설정하면 메시지 큐 기능의 성능을 향상시킬 수 있지만 보다 많은 메모리를 사용하게 됩니다.
- s 스케일 인수를 설정합니다. 디폴트 스케일 인수는 1이며 거의 모든 워크로드에 충분해야 합니다. 스케일 인수 1이 충분하지 않은 경우 시스템이 워크로드를 처리하기에 너무 적기 때문일 수 있습니다. 스케일 인수는 커널 매개변수 권장사항을 현재 시스템 크기에 비례하여 큰 시스템의 커널 매개변수로 설정합니다. 예를 들어, 스케일 인수 2.5는 현재 시스템 크기의 2.5배인 시스템의 권장 커널 매개변수입니다.
- t semsys:seminfo\_semume 및 shmsys:shminfo\_shmseg 커널 매개변수 값에 대한 권장사항을 제공합니다. 이 옵션은 DB2 for the Solaris 운영 체제에서만 사용할 수 있습니다. 연결 수가 많은 멀티 스레드 프로그램의 경우 이러한 커널 매개변수는 디폴트값보다 크게 설정해야 합니다. 커널 매개변수는 필요한 멀티 스레드 프로그램이 로컬 응용프로그램인 경우에만 재설정해야 합니다.

**semsys:seminfo\_semume**

특정 프로세스에서 사용할 수 있는 세마포어 실행 취소 구조의 한계

**shmsys:shminfo\_shmseg**

특정 프로세스에서 작성할 수 있는 공유 메모리 세그먼트 수에 대한 한계.

이러한 매개변수는 /etc/system 파일에서 설정됩니다. 다음은 값을 설정하는 방법으로서, db2osconf 유틸리티가 값을 권장하기 위해 사용하는 방법입니다. DB2는 각 로컬 연결마다 하나의 세마포어와 하나의 공유 메모리 세그먼트를 사용하여 통신합니다. 멀티 스레드 응용프로그램이 로컬 응용프로그램이고 DB2에 대한 연결 수가 X개인 경우 해당 응용프로그램(프로세스)에는 DB2와 통신하기 위한 X개의 공유 메모리 세그먼트와 X개의 세마포어 실행 취소 구조가 필요합니다. 따라서 두 개 커널 매개변수의 값은  $X + 10$ 으로 설정되어야 합니다(플러스 10은 안전 여백을 제공함).

-l 또는 -f 스위치를 사용하지 않는 경우 db2osconf 유틸리티는 /etc/system 파일의 구문을 사용하여 커널 매개변수를 표시합니다. 사용자 오류를 방지하기 위해 출력을 잘라내어 /etc/system 파일에 직접 붙여넣을 수 있습니다.

커널 매개변수는 CPU 수와 시스템의 실제 메모리 양을 모두 기반으로 하여 권장됩니다. 둘 중 하나가 너무 낮은 경우 권장사항은 낮은 값을 기반으로 합니다.

## 예:

다음은 db2osconf 유틸리티를 실행하고 -t 스위치가 500개 스레드에 대해 설정하여 생성된 샘플 출력입니다. 결과는 머신에 따라 다르므로 수신 결과는 환경에 따라 다릅니다.

```
db2osconf -t 500

set msgsys:msginfo_msgmax = 65535
set msgsys:msginfo_msgmnb = 65535
set msgsys:msginfo_msgssz = 32
set msgsys:msginfo_msgseg = 32767
set msgsys:msginfo_msgmap = 2562
set msgsys:msginfo_msgmni = 2560
set msgsys:msginfo_msgtql = 2560
set semsys:seminfo_semmap = 3074
set semsys:seminfo_semmni = 3072
set semsys:seminfo_semmns = 6452
set semsys:seminfo_semmnu = 3072
set semsys:seminfo_semume = 600
set shmsys:shminfo_shmmax = 2134020096
set shmsys:shminfo_shmmni = 3072
set shmsys:shminfo_shmseg = 600
```

```
Total kernel space for IPC:
0.35MB (shm) + 1.77MB (sem) + 1.34MB (msg) == 3.46MB (total)
```

set semsys:seminfo\_semume 및 set shmsys:shminfo\_shmseg에 대한 권장값은 db2osconf -t 500을 실행하여 제공된 추가 값입니다.

## 사용 시 참고사항

특정 DB2 워크로드를 기반으로 커널 매개변수를 권장할 수 있지만 이러한 정밀도 레벨은 도움이 되지 않습니다. 커널 매개변수 값이 실제로 필요한 값과 너무 유사하고 추후 워크로드가 변경되는 경우 프로세스간 통신(IPC) 자원 부족으로 인해 DB2에 문제점이 발생할 수 있습니다. IPC 자원이 부족하면 DB2에 대해 계획되지 않은 작동 중단이 발생할 수 있으며 커널 매개변수를 늘리기 위해 재부트가 필요합니다. 커널 매개변수를 올바르게 높게 설정함으로써 추후 매개변수 변경 요구를 줄이거나 제거해야 합니다. 커널 매개변수 권장사항이 소비하는 메모리의 양은 시스템 크기에 비해 매우 작습니다. 예를 들어, RAM이 4GB이고 CPU가 네 개인 시스템의 경우 권장되는 커널 매개변수의 메모리 양은 4.67MB 또는 0.11%입니다. 해당 이점을 고려할 때 커널 매개변수에 사용되는 이러한 메모리의 일부를 승인해야 합니다.

Solaris 운영 체제에는 두 가지 버전의 db2osconf 유틸리티(64비트 커널용과 32비트 커널용)가 있습니다. 유틸리티는 다음 특수 디바이스에 액세스하므로(액세스는 읽기 전용) root로서 실행되거나 sys 그룹과 함께 실행되어야 합니다.

```
crw-r----- 1 root    sys      13,  1 Jul 19 18:06 /dev/kmem
crw-rw-rw-   1 root    sys      72,  0 Feb 19 1999 /dev/ksyms
crw-r----- 1 root    sys      13,  0 Feb 19 1999 /dev/mem
```

## 제 239 장 db2pd - DB2 데이터베이스 모니터 및 문제점 해결

DB2 데이터베이스 시스템 메모리 세트에서 정보를 검색합니다.

### 권한 부여

다음 권한 레벨 중 하나가 필요합니다.

- SYSADM 권한 레벨.
- SYSCTRL 권한 레벨.
- SYSMOINT 권한 레벨.
- SYSMON 권한 레벨.

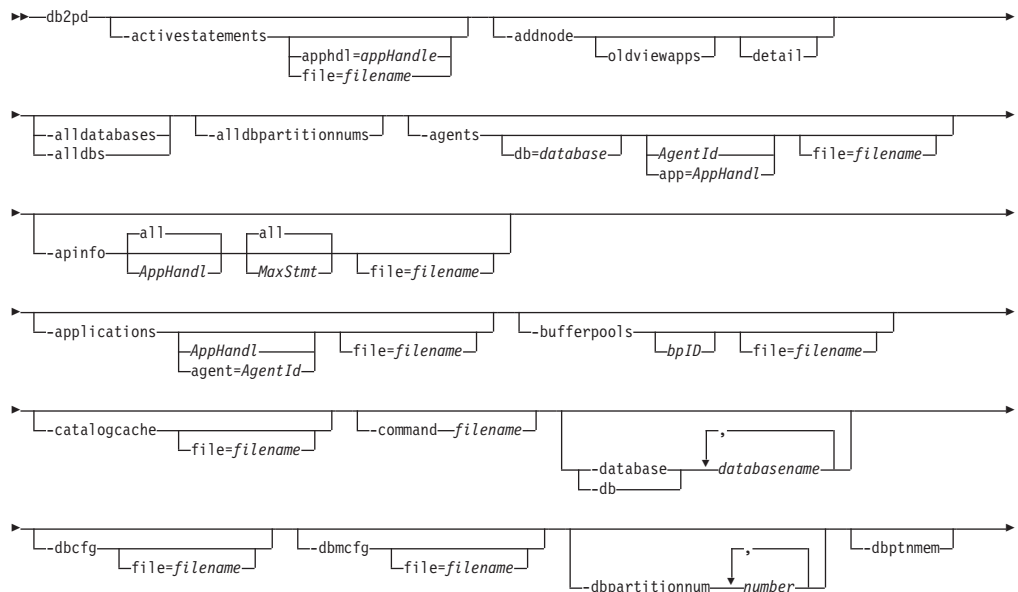
SYSMON 권한 레벨이 부여된 경우 다음 옵션을 사용할 수 없습니다.

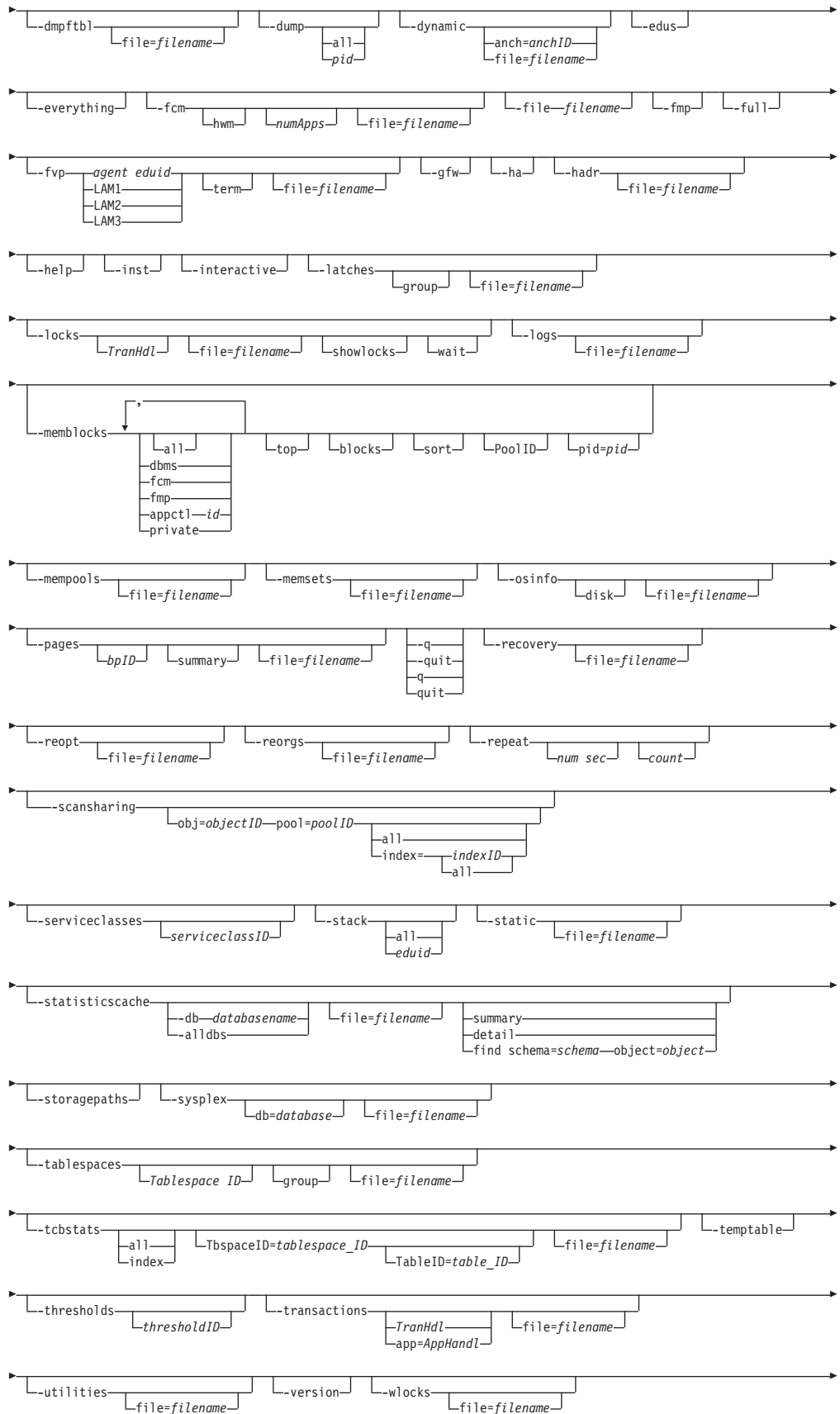
- **dump**
- **memblocks**
- **stack**

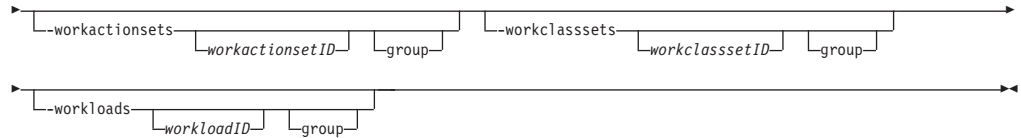
### 필수 연결

최소한의 연결 요구사항은 없습니다. 그러나 데이터베이스 범위 옵션을 지정한 경우 해당 데이터베이스가 활성화되어야 명령이 요청된 정보를 리턴할 수 있습니다.

### 명령 구문







## 명령 매개변수

### -activestatements

활성 명령문 목록에 대한 정보를 리턴합니다.

#### apphdl=appHandle

응용프로그램 핸들을 지정한 경우 특정 응용프로그램에 대한 정보가 리턴됩니다.

#### file=filename

**-activestatements** 출력을 지정된 파일로 보냅니다.

### -addnode

데이터베이스 파티션 서버 추가 조작에 대한 진행 정보를 리턴합니다. 이 매개변수는 추가하는 데이터베이스 파티션 서버에 대해 발행된 경우에만 정보를 리턴합니다. 진행 정보는 재시작할 때까지 새 데이터베이스 파티션 서버에 대해 지속됩니다. 기존 데이터베이스 파티션 서버에 대해 발행한 경우 이 매개변수는 정보를 리턴하지 않습니다.

db2pd -addnode 명령의 샘플 출력을 참조하십시오.

### -alldatabases | -alldbbs

명령이 모든 데이터베이스의 모든 메모리 세트에 접속됨을 지정합니다.

### -alldbpartitionnums

명령이 인스턴스에서 활성 데이터베이스 파티션 서버 모두에 대해 실행됨을 지정합니다. db2pd는 db2pd가 실행되는 동일한 실제 머신의 데이터베이스 파티션 서버에서만 정보를 보고합니다.

### -agents

에이전트에 대한 정보를 리턴합니다.

에이전트 ID를 지정한 경우 에이전트에 대한 정보가 리턴됩니다. 응용프로그램 ID를 지정한 경우에는 응용프로그램에 대한 작업을 수행 중인 모든 에이전트에 대한 정보가 리턴됩니다. 출력 범위를 지정할 데이터베이스를 선택한 경우, **-inst** 매개변수와 함께 이 명령 매개변수를 지정하십시오. 에이전트 사용 시 참고사항을 참조하십시오.

### -apinfo

적용 가능한 경우, 현재 작업 단위(UOW)의 동적 SQL문 실행을 포함하여 응용프로그램에 대한 자세한 정보를 표시합니다.

### *AppHandl*

응용프로그램 핸들을 지정한 경우 특정 응용프로그램에 대한 정보가 리턴됩니다. 디폴트는 해당 파티션에서 실행 중인 모든 응용프로그램에 대한 정보를 표시하는 것입니다.

### *MaxStmt*

최대 명령문 수를 지정한 경우 최근 SQL문에 대한 정보(지정된 최대 수와 같은)가 리턴됩니다. 디폴트는 실행된 모든 SQL문에 대한 정보를 표시하는 것입니다.

### **file=filename**

**-apinfo** 출력을 지정된 파일로 보냅니다.

db2pd -apinfo 명령의 샘플 출력을 참조하십시오.

주: 응용프로그램에 대한 SQL문 텍스트를 포함하여 작업 단위(UOW)의 지난 실행기록을 캡처하려면 명령문 실행기록 절을 사용하여 교착 상태 이벤트 모니터를 활성화하십시오. 예를 들어, 다음 명령문 중 하나를 사용하십시오.

```
create event monitor testit for deadlocks with details history write to file path global  
create event monitor testit for deadlocks with details history write to table
```

CREATE EVENT MONITOR 문에는 데이터를 기록할 테이블 스페이스 및 테이블의 이름을 지정할 수 있는 기능과 같은 추가 옵션이 있습니다. 세부사항은 CREATE EVENT MONITOR 문 설명을 참조하십시오. 명령문 실행기록 기능이 있는 이벤트 모니터는 모든 응용프로그램에 영향을 주고 DB2 데이터베이스 관리 프로그램에 의한 모니터 힙 사용량을 늘립니다.

**-apinfo** 사용 시 참고사항을 참조하십시오.

### **-applications**

응용프로그램에 대한 정보를 리턴합니다.

응용프로그램 ID를 지정한 경우 해당 응용프로그램에 대한 정보가 리턴됩니다.

에이전트 ID를 지정한 경우에는 응용프로그램 대신 작업 중인 에이전트에 대한 정보가 리턴됩니다. **-applications** 사용 시 참고사항을 참조하십시오.

### **-bufferpools**

버퍼 풀에 대한 정보를 리턴합니다. 버퍼 풀 ID를 지정한 경우 버퍼 풀에 대한 정보가 리턴됩니다. **-bufferpools** 사용 시 참고사항을 참조하십시오.

### **-catalogcache**

통계의 인메모리 버전을 유지보수하는 카탈로그 캐시에 대한 정보를 리턴합니다. 예를 들면, 다음과 같습니다.

```
db2pd -catalogcache -db sample
```

db2pd -catalogcache 명령의 샘플 출력을 참조하십시오.



리턴된 정보에 대한 정의는 **-catalogcache**에 있습니다. SYSTABLES에 대한 출력은 동일 테이블에 대해 여러 개의 항목을 가질 수 있습니다(위의 출력에서 DEPT 참조). 여러 항목은 동일 테이블에 대한 통계의 다양한 버전에 해당됩니다. 사용 잠금 이름은 동일한 오브젝트의 항목 사이에 고유하며 소프트 무효화 항목은 'S'로 표시됩니다. **-catalogcache** 사용 시 참고사항을 참조하십시오.

**-command filename**

파일에 지정된 db2pd 명령 옵션을 읽고 실행할 것을 지정합니다.

**-database | -db databasename**

명령이 지정된 데이터베이스의 데이터베이스 메모리 세트에 접속됨을 지정합니다.

**-dbcfg** 데이터베이스 구성 매개변수의 설정을 리턴합니다. **-dbcfg** 사용 시 참고사항을 참조하십시오.

**-dbmcfg**

데이터베이스 관리 프로그램 구성 매개변수의 설정을 리턴합니다.

출력 범위를 지정할 데이터베이스를 선택한 경우, **-inst** 명령 매개변수와 함께 이 옵션을 지정하십시오. **-dbmcfg** 사용 시 참고사항을 참조하십시오.

**-dbpartitionnum number**

지정된 데이터베이스 파티션 서버에 대해 명령을 실행할 것을 지정합니다.

**-dbptnmem**

데이터베이스 파티션 메모리 통계를 나열합니다.

**-dmpftbl**

파일 테이블 콘텐츠를 덤프합니다.

**-dump**

**diagpath** 디렉토리에서 스택 추적 및 2진 덤프 파일을 생성합니다. UNIX 운영 체제에서만 사용 가능합니다.

- 현재 데이터베이스 파티션의 모든 에이전트에 대한 스택 추적 및 2진 덤프 파일을 생성하려면 **all** 명령 매개변수와 함께 지정하십시오.
- 특정 에이전트에 대한 스택 추적 및 2진 덤프 파일을 생성하려면 **pid** 옵션과 함께 지정하십시오.

**-dynamic**

동적 SQL 실행에 대한 정보를 리턴합니다. **-dynamic** 사용 시 참고사항을 참조하십시오.

**anch=anchID**

앵커 ID를 지정한 경우 특정 동적 SQL에 대한 정보가 리턴됩니다.

**file=filename**

**-dynamic** 출력을 지정된 파일로 보냅니다.

**-edus** 인스턴스의 모든 EDU를 나열합니다. 지속된 트랩의 경우, 이 옵션을 지정하면 EDU Name을 출력하여 EDU가 일시중단됨을 표시합니다.

db2pd -edus 명령의 샘플 출력을 참조하십시오.

**-everything**

서버에 대해 로컬인 모든 데이터베이스 파티션 서버의 모든 데이터베이스에 대해 모든 옵션을 실행합니다.

**-fcm** FCM(Fast Communication Manager)에 대한 정보를 리턴합니다.

- 출력 범위를 지정할 데이터베이스를 선택한 경우, **-inst** 매개변수와 함께 이 매개변수를 지정하십시오.
- DB2 인스턴스 시작 이후에 응용프로그램별로 FCM 버퍼 및 채널의 상위 워터 마크(water mark) 소비를 검색하려면 **hwm** 매개변수와 함께 이 매개변수를 지정하십시오. 응용프로그램의 상위 워터 마크(water mark) 소비 값은 이미 데이터베이스와의 연결이 끊어진 경우에도 보유됩니다.
- 현재 및 HWM 소비 통계에서 db2pd 명령이 보고하는 최대 응용프로그램 수를 제한하려면 **numApps** 옵션과 함께 이 매개변수를 지정하십시오.

**-fcm** 사용 시 참고사항을 참조하십시오.

**-file filename**

지정된 파일에 출력을 기록할 것을 지정합니다.

**-fmp** 분리(fenced) 루틴이 실행되는 프로세스에 대한 정보를 리턴합니다. **-fmp** 사용 시 참고사항을 참조하십시오.

**-full** 모든 출력이 해당되는 최대 길이로 확장됨을 지정합니다. 지정하지 않은 경우, 출력은 표시 스페이스를 절약하기 위해 절단됩니다.

**-fvp** 분리(fenced) 벤더 프로세스 정보를 표시하고 응답하지 않는 상황에서 분리(fenced) 벤더 프로세스의 종료를 허용합니다. 이는 벤더 미디어 디바이스가 사용되는 백업, 리스토어, 프룬(prune) 실행기록, 로드, 로드 사본(롤 포워드) 및 로그 관리 프로그램에 적용됩니다.

주: **-database database** 명령 매개변수는 올바른 메모리 세트에 연결하여 정보를 수집하기 위해 이 매개변수와 함께 사용해야 합니다.

*agent eduid*

백업, 리스토어, 프룬(prune) 실행기록, 로드 또는 로드 사본(롤 포워드) 에이전트의 DB2 EDU ID에 대한 분리(fenced) 벤더 프로세스 정보를 표시합니다.

**LAM1**

**logarchmeth1**에 대한 분리(fenced) 벤더 프로세스 정보를 표시합니다.

## LAM2

**logarchmeth2**에 대한 분리(fenced) 벤더 프로세스 정보를 표시합니다.

## LAM3

현재 로그 아카이브 메소드 구성 매개변수가 **VENDOR**로 설정되지 않아서, 이전 벤더 아카이브 메소드에서 로그를 검색하기 위해 **ROLLFORWARD DATABASE** 동안 분리(fenced) 벤더 프로세스를 임시로 작성해야 하는 특수한 경우의 분리(fenced) 벤더 프로세스 정보를 표시합니다.

**term** 분리(fenced) 벤더 프로세스 정보 표시 외에도, 이 옵션은 지정된 분리(fenced) 벤더 프로세스를 종료하기도 합니다.

주: 이 옵션은 Windows 운영 체제에 대해 영향을 주지 않습니다.

**-gfw** 현재 활성 상태이지만 어떤 이유로 비활성화되었던 이벤트 모니터 목록을 리턴합니다. 또한 이벤트 모니터가 각각의 급속 기록기 EDU에 대한 데이터를 기록하는 목표에 대한 정보 및 통계도 리턴합니다.

**-ha** 고가용성 통계를 보고합니다.

**-hadr** 고가용성 재해 복구(HADR) 정보를 보고합니다. 보고된 각 요소의 설명은 데이터베이스 모니터링 안내서 및 참조서의 고가용성 재해 복구 섹션에서 찾을 수 있습니다.

### **-h | -help**

온라인 도움말 정보를 표시합니다.

**-inst** 모든 인스턴스 범위 정보를 리턴합니다.

### **-interactive**

db2pd 명령을 실행할 때 **DB2PDOPT** 환경 변수에 대해 지정된 값을 겹쳐쓸 것을 지정합니다.

### **-latches**

모든 래치 보유자와 모든 래치 대기자를 보고합니다.

**group** 뒤에 대기자 목록이 있는 보유자 목록을 인쇄합니다.

**file=filename**

**-latches** 출력을 *filename*으로 보냅니다.

**-locks** 잠금에 대한 정보를 리턴합니다.

특정 트랜잭션이 보유하는 잠금에 대한 정보를 얻으려면 트랜잭션 핸들을 지정하십시오.

잠금 이름에 대한 자세한 정보를 리턴하려면 **showlocks** 명령 매개변수와 함께 지정하십시오. 파티션된 테이블 및 개별 데이터 파티션의 행 및 블록 잠금의 경우, **showlocks**는 잠금 정보가 있는 행의 일부로 데이터 파티션 ID를 표시합니다.

대기 상태에 있는 잠금과 잠금 소유자를 리턴하려면 **wait** 명령 매개변수를 지정하십시오.

**-locks** 사용 시 참고사항을 참조하십시오.

**-logs** 로그 파일에 대한 정보를 리턴합니다. **-logs** 사용 시 참고사항을 참조하십시오.

#### **-memblocks**

메모리 풀에 대한 정보를 리턴합니다.

**dbms** dbms 메모리 세트의 블록만 보고합니다.

**fcm** FCM(Fast Communication Manager) 메모리 세트의 블록만 보고합니다.

**fmp** 분리(fenced) 모드 프로시저 메모리 세트의 블록만 보고합니다.

**appctl id**  
응용프로그램 제어 세트의 블록만 보고합니다.

**all** 모든 메모리 세트의 블록을 보고합니다.

**top** 각 세트의 맨 위 메모리 사용자를 보고합니다.

**blocks** 각 세트의 메모리 블록을 보고합니다.

**sort** 각 세트에 있는 각 풀에 대한 정렬된 메모리 블록을 보고합니다.

#### **PoolID**

특정 풀의 메모리 블록을 보고합니다.

**pid=pid**  
특정 프로세스 ID의 메모리 블록을 보고합니다(UNIX 운영 체제의 경우에만).

#### **private**

전용 메모리 세트의 메모리 블록을 보고합니다(Windows 운영 체제의 경우에만).

**-memblocks** 사용 시 참고사항을 참조하십시오.

#### **-mempools**

메모리 풀에 대한 정보를 리턴합니다.

리턴되는 정보에 모든 인스턴스 범위 정보를 포함하려면 **-inst** 옵션과 함께 이 옵션을 지정하십시오. **-mempools** 사용 시 참고사항을 참조하십시오.

### **-memsets**

메모리 세트에 대한 정보를 리턴합니다.

리턴되는 정보에 모든 인스턴스 범위 정보를 포함하려면 **-inst** 명령 매개변수와 함께 이 명령 매개변수를 지정하십시오. **-memsets** 사용 시 참고사항을 참조하십시오.

### **-osinfo**

운영 체제 정보를 리턴합니다. 디스크 경로가 지정된 경우 디스크에 대한 정보가 인쇄됩니다. **-osinfo** 사용 시 참고사항을 참조하십시오.

### **-pages** 버퍼 풀 페이지에 대한 정보를 리턴합니다.

*bpID* 버퍼 풀 ID가 지정된 경우, 지정된 버퍼 풀의 페이지만 리턴됩니다.

#### **summary**

이 옵션을 지정하면 요약 정보 섹션만 표시됩니다.

**-pages** 사용 시 참고사항을 참조하십시오. `db2pd -pages` 명령의 샘플 출력을 참조하십시오.

### **-q | -quit | q | quit**

종료. `db2pd` 키워드만 발행된 경우 `db2pd`는 대화식 모드에서 실행됩니다. **quit** 명령을 사용하면 이 모드에서 다시 표준 명령 프롬프트로 나갑니다.

### **-recovery**

복구 활동에 대한 정보를 리턴합니다. **-recovery** 사용 시 참고사항을 참조하십시오.

**-reopt REOPT ONCE** 옵션을 사용하여 다시 최적화된, 캐시된 SQL문에 대한 정보를 리턴합니다. **-reopt** 사용 시 참고사항을 참조하십시오.

### **-reorgs**

테이블 및 데이터 파티션 재구성에 대한 정보 및 통계를 리턴합니다. **-reorgs** 사용 시 참고사항을 참조하십시오.

### **-repeat num sec count**

지정된 초 수 이후에 명령이 반복됨을 지정합니다. 초 수의 값을 지정하지 않은 경우, 명령은 5초마다 반복합니다. 또한 출력이 반복될 횟수를 지정할 수 있습니다. *count* 값을 지정하지 않는 경우, 명령은 인터럽트될 때까지 반복됩니다.

### **-scansharing**

지정된 데이터베이스에서 테이블 또는 블록 인덱스 스캔 공유를 가지고 있는 모든 테이블에 대한 스캔 공유 정보를 리턴합니다.

**obj=objectID pool=poolID**

지정된 테이블에 대한 스캔 공유 정보를 리턴합니다.

**all** 모든 테이블에 대한 스캔 공유 정보를 리턴합니다. 테이블마다,

테이블 또는 범위 스캔 공유 정보가 리턴됩니다. 또한 MDC 테이블의 경우, 블록 인덱스 스캔 공유 정보가 리턴됩니다.

**index=**

*indexID*

지정된 테이블에 대한 스캔 공유 정보와, 지정된 블록 인덱스에 대한 블록 인덱스 스캔 공유 정보를 리턴합니다.

**all** 지정된 테이블에 대한 테이블 스캔 공유 정보와, 모든 블록 인덱스에 대한 블록 인덱스 스캔 공유 정보를 리턴합니다.

db2pd -scansharing 명령의 샘플 출력을 참조하십시오.

**-scansharing** 사용 시 참고사항을 참조하십시오.

**-serviceclasses** *serviceclassID*

데이터베이스의 서비스 클래스에 대한 정보를 리턴합니다. *serviceclassID*는 하나의 특정 서비스 클래스에 대한 정보를 검색하기 위한 선택적 매개변수입니다. *serviceclassID*를 지정하지 않은 경우 모든 서비스 클래스에 대한 정보가 검색됩니다.

**-serviceclasses** 사용 시 참고사항을 참조하십시오. db2pd -serviceclasses 명령의 샘플 출력도 참조하십시오.

**-stack** 엔진 정지의 경우, 이벤트 스택 및 이벤트 실행기록(이벤트 플로우)을 사용하여 DB2 상태에 대한 정보를 가져올 수 있습니다. **diagpath** 디렉토리에서 이벤트 스택 및 실행기록 추적 파일을 생성합니다. 트랩 및 패닉 상태의 EDU를 제외하고, 다른 모든 EDU는 스택 추적 파일 *pid.tid.node.stack.txt*로 덤프합니다. UNIX 운영 체제에서 **-stack**을 사용하고, Windows 운영 체제에서 **-stack all**을 사용하십시오.

**all** 현재 데이터베이스 파티션의 모든 프로세스에 대한 스택 추적 파일을 생성하려면 이 옵션을 지정하십시오.

*eduid* 지정된 ID의 EDU로 출력을 제한합니다. 형식화된 이벤트 및 접속된 데이터는 db2dump 디렉토리에 있는 관련된 *pid.tid/EDUID.node.trap.txt* 트랩 파일로 덤프됩니다.

이벤트 스택은 다음 순서로 출력됩니다.

마지막 이벤트(이벤트 스택의 맨 위에 있음)

- 이벤트 유형 및 간단한 설명
- 고객 영향
- 오브젝트 ID

- ECF ID, 프로브
  - 맨 위 이벤트 헤더
  - 맨 위 이벤트 규정자(있는 경우)
  - 맨 위 이벤트 데이터(있는 경우)
- 첫 번째 이벤트(이벤트 스택의 맨 아래에 있음)
- 이벤트 유형 및 간단한 설명
  - 고객 영향
  - 오브젝트 ID
  - ECF ID, 프로브
  - 맨 아래 이벤트 헤더
  - 맨 아래 이벤트 규정자(있는 경우)
  - 맨 아래 이벤트 데이터(있는 경우)

위에서, ECF ID는 ECF 식별자(*product, component, function*으로 형식화될)이고 프로브는 코드 또는 고유 번호(함수에 대한) 라인입니다.

이벤트 플로우(기록된 이벤트 “실행기록”)는 다음 순서로 출력됩니다.

첫 번째 이벤트 레코드

- 이벤트 유형 및 간단한 설명
- 고객 영향
- 오브젝트 ID
- ECF ID, 프로브
- 이벤트 헤더
- 오브젝트 데이터(문자열 또는 정수가 아닌 경우)

마지막 이벤트 레코드

- 이벤트 유형 및 간단한 설명
- 고객 영향
- 오브젝트 ID
- ECF ID, 프로브
- 이벤트 헤더
- 오브젝트 데이터(문자열 또는 정수가 아닌 경우)

**-static** 정적 SQL 및 패키지의 실행에 대한 정보를 리턴합니다. **-static** 사용 시 참고 사항을 참조하십시오.

#### **-statisticscache**

데이터베이스 레벨에서 통계 캐시에 대한 정보를 리턴합니다.

### summary

통계 캐시를 요약합니다. 데이터베이스 `sample`에 대한 통계 캐시 요약을 덤프하려면 다음 명령을 발행하십시오.

```
db2pd -db sample -statisticscache summary
```

**detail** 실시간 통계 모음으로 수집된 최신 통계가 있는 모든 테이블의 통계 캐시에 저장된 자세한 통계 정보를 덤프하려면 이 옵션을 지정하십시오. 모든 데이터베이스의 통계 캐시에 저장된 자세한 통계 정보를 덤프하려면 다음 명령을 발행하십시오.

```
db2pd -statisticscache detail -alldbs
```

### find schema=*schema* object=*object*

스키마 이름의 스키마와 테이블 이름의 오브젝트가 있는 특정 테이블에 대한 자세한 통계 정보를 덤프하려면 이 옵션을 지정하십시오. 데이터베이스 `sample`의 `USER1.T1` 테이블에 대한 자세한 통계 정보를 덤프하려면 다음 명령을 발행하십시오.

```
db2pd -db sample -statisticscache find schema=USER1 object=T1
```

**-statisticscache** 사용 시 참고사항을 참조하십시오.

### -storagepaths

데이터베이스에 대해 정의된 자동 스토리지 경로에 대한 정보를 리턴합니다.

**-storagepaths** 사용 시 참고사항을 참조하십시오. `db2pd -storagepaths` 명령의 샘플 출력을 참조하십시오.

### -sysplex

**db** 매개변수로 표시되는 데이터베이스 별명과 연관된 서버 목록에 대한 정보를 리턴합니다. **-database** 명령 매개변수를 지정하지 않는 경우, 모든 데이터베이스에 대한 정보가 리턴됩니다.

출력 범위를 지정할 데이터베이스를 선택한 경우, **-inst** 명령 매개변수와 함께 이 명령 매개변수를 지정하십시오.

**-sysplex** 사용 시 참고사항을 참조하십시오.

### -tablespaces

테이블 스페이스에 대한 정보를 리턴합니다.

테이블 스페이스와 함께 그룹화된 테이블 스페이스의 컨테이너에 대한 정보를 표시하려면 **group** 명령 매개변수와 함께 지정하십시오.

특정 테이블 스페이스 및 해당 컨테이너에 대한 정보를 표시하려면 *Tablespace ID* 명령 매개변수와 함께 지정하십시오.

**-tablespaces** 사용 시 참고사항을 참조하십시오. `db2pd -tablespaces` 명령의 샘플 출력도 참조하십시오.



### **-tcbstats**

테이블 및 인덱스에 대한 정보를 리턴합니다. 테이블의 총 갱신 수, UDI 및 실시간 통계 UDI 카운터(RTSUDI)도 리턴됩니다.

#### **TbpaceID=tablespace\_ID**

특정 테이블 스페이스에 대한 정보를 표시하려면 이 옵션을 지정하십시오.

#### **TableID=table\_ID**

특정 테이블에 대한 정보를 표시하려면 이 옵션을 지정하십시오. **TbpaceID** 옵션은 **TableID** 옵션을 사용할 때 필요합니다.

임시 테이블 압축의 경우, **-tcbstats** 출력에는 출력의 TCB 테이블 통계 섹션에 있는 두 개의 새 컬럼이 포함됩니다.

1. StoredBytes: 이 컬럼은 db2pd -temptable 출력의 “저장된 총 임시 바이트 수”에 해당됩니다.
2. BytesSaved: 이 컬럼은 db2pd -temptable 출력의 “저장된 총 바이트 수” 값에 해당됩니다.

**-tcbstats** 사용 시 참고사항을 참조하십시오.

### **-temptable**

디폴트로, 임시 테이블에 대한 다음 정보를 리턴합니다.

- Number of Temp Tables 데이터베이스 관리 프로그램이 시작되었거나 마지막 카운트 재설정 이후에 작성 및 삭제된 총 임시 테이블 수.
- Comp Eligible Temp 데이터베이스 관리 프로그램이 *query type*, *minimum row size* 및 *minimum expected table size* 등록 정보를 기초로 압축에 적합한 것으로 판별한 임시 테이블 수.
- Compressed Temps 실제로 압축된 총 임시 테이블 수. 이는 테이블에 충분한 데이터가 있어서 임시 테이블에 대해 압축 사전을 작성할 수 있음을 의미합니다.
- Total Stored Temp Bytes 디스크에 저장된 임시 테이블에 대한 실제 행 데이터의 총계. 이 수치는 압축 및 압축되지 않은 임시 테이블에서 제공될 수 있습니다.
- Total Bytes Saved 행을 압축하여 저장한 총 바이트 수
- Total Compressed Rows 압축을 사용하여 최소한 1바이트를 저장한 행 수의 누적 계수
- Total Temp Table Rows 압축 여부에 관계없이 모든 임시 테이블에 삽입된 총 행 수. 압축된 임시 테이블에 삽입되는 모든 행을 압축할 필요는 없습니다.

**-temptable** 사용 시 참고사항을 참조하십시오. `db2pd -temptable` 명령의 샘플 출력을 참조하십시오.

**-thresholds** *thresholdID*

임계값에 대한 정보를 리턴합니다. *thresholdID*는 선택사항이지만 임계값 ID를 지정하면 특정 임계값에 대한 정보가 리턴됩니다. *thresholdID*를 지정하지 않은 경우 모든 사용 가능한 임계값과 사용하지 않는 임계값에 대한 정보가 검색됩니다.

**-thresholds** 사용 시 참고사항을 참조하십시오. `db2pd -thresholds` 명령의 샘플 출력을 참조하십시오.

**-transactions**

활성 트랜잭션에 대한 정보를 리턴합니다. 트랜잭션 핸들을 지정한 경우, 해당 트랜잭션 핸들에 대한 정보가 리턴됩니다. 응용프로그램 핸들을 지정한 경우, 트랜잭션의 응용프로그램 핸들에 대한 정보가 리턴됩니다. **-transactions** 사용 시 참고사항을 참조하십시오.

**-utilities**

유틸리티 정보를 보고합니다. 보고된 각 요소의 설명은 데이터베이스 모니터링 안내서 및 참조서의 유틸리티 섹션에서 찾을 수 있습니다.

**-v | -version**

설치된 DB2 제품의 현재 버전 및 서비스 레벨을 표시합니다.

**-wlocks**

대기 중인 각 잠금에 대한 소유자 및 대기자 정보를 표시합니다. `db2pd -wlocks` 명령의 샘플 출력에서, G 값의 잠금 상태(Sts)는 잠금 소유자를 지정하고, W 값의 Sts는 해당 잠금의 대기자를 지정합니다.

**file=filename**

**-wlocks** 출력을 지정된 파일로 보냅니다.

**-wlocks** 사용 시 참고사항을 참조하십시오.

**-workactionsets** *workactionsetID*

사용 가능한 모든 작업 조치 세트와, 이 조치에서 사용 가능한 모든 작업 조치에 대한 정보를 리턴합니다.

**group** 작업 조치 세트 ID별로 그룹화된 동일한 정보를 리턴합니다.

**-workactionsets** 사용 시 참고사항을 참조하십시오.

**-workclasssets** *workclasssetID*

사용 가능한 작업 조치 세트에 참조된 모든 작업 클래스 세트와, 작업 클래스 세트의 모든 작업 클래스에 대한 정보를 리턴합니다.

**group** 작업 클래스 세트 ID별로 그룹화된 동일한 정보를 리턴합니다.

**-workclasssets** 사용 시 참고사항을 참조하십시오.

**-workloads** *workloadID*

명령이 실행될 때 메모리에 있는 워크로드 정의, 사용자 특권 보유자 및 로컬 파티션 워크로드 통계 목록을 리턴합니다.

**group** 워크로드 ID별로 그룹화된 동일한 정보를 리턴합니다.

db2pd -workloads 명령의 샘플 출력을 참조하십시오.

**-workloads** 사용 시 참고사항을 참조하십시오.

**예:**

클라이언트 요청에 서비스를 제공하는 에이전트에 대한 정보를 얻으려면 다음 방법으로 명령행에서 db2pd 명령을 사용하십시오.

```
db2pd -agents
```

클라이언트 요청에 서비스를 제공하는 에이전트에 대한 정보를 얻으려면 다음 방법으로 명령행에서 db2pd 명령을 사용하십시오. 이 경우, db2pd 명령을 호출하기 전에 **-agents** 매개변수로 **DB2PDOPT** 환경 변수를 설정합니다. 명령은 실행될 때 환경 변수에 설정된 정보를 사용합니다.

```
export DB2PDOPT="-agents"  
db2pd
```

클라이언트 요청에 서비스를 제공하는 에이전트에 대한 정보를 얻으려면 다음 방법으로 명령행에서 db2pd 명령을 사용하십시오. 이 경우, db2pd 명령을 호출하기 전에 file.out 파일에서 **-agents** 매개변수가 설정됩니다. **-command** 매개변수를 사용하면 명령이 실행될 때 file.out 파일에 있는 정보를 사용합니다.

```
echo "-agents" > file.out  
db2pd -command file.out
```

모든 데이터베이스와 인스턴스 범위 정보를 얻으려면 다음 방법으로 명령행에서 db2pd 명령을 사용하십시오.

```
db2pd -inst -alldbs
```

분리(fenced) 벤더 프로세스 상태 정보를 얻으려면 다음 방법으로 명령행에서 db2pd -fvp 명령을 사용하십시오.

**로그 관리 프로그램의 경우:**

- 데이터베이스 SAMPLE에서 **logarchmeth1**이 TSM으로 설정되어 있습니다. 언제든지 다음을 발행하십시오.

```
db2pd -db sample -fvp lam1
```

결과 출력은 다음과 같습니다.

-----  
Fenced Vendor Process State Information:  
-----

Log Manager:

-----  
LOGARCHMETH1 available.

Vendor EDU is available and running.  
startTime: 1155581841 20060814145721  
function: sqluvint

이 출력은 분리(fenced) 벤더 프로세스가 2006년 8월 14일 14:57 이후에 벤더 함수 sqluvint에서 실행 중임을 알려줍니다. 이제, 함수를 너무 오래 실행했다고 느끼거나, 이 프로세스가 TSM 자원을 기다리면서 정지되었다고 판별한 경우, 다음을 발행하여 분리(fenced) 벤더 프로세스를 종료할 수 있습니다.

```
db2pd -db sample -fvp lam1 term
```

결과 출력은 다음과 같습니다.

-----  
Fenced Vendor Process State Information:  
-----

Log Manager:

-----  
LOGARCHMETH1 available.

Vendor EDU is available and running.  
startTime: 1155581841 20060814145721  
function: sqluvint  
This fenced vendor process has been sent a signal to terminate.

이 출력은 위와 동일한 정보를 표시하지만, 종료 요청이 전송되었음을 알립니다. 잠시 기다린 후 요청이 적용되었다는 점에 유의해야 합니다.

- 분리(fenced) 벤더 프로세스가 실행 중이지만 벤더 코드에서 실행하지 않는 경우, 일반 표시 요청에 대해 다음이 표시됩니다.

-----  
Fenced Vendor Process State Information:  
-----

Log Manager:

-----  
LOGARCHMETH1 available.

Vendor EDU is available and running.  
No vendor code being run.

**백업의 경우:**

주: 아래에 설명된 명령에 대한 대안으로 FORCE APPLICATION 명령을 사용할 수 있는 점에 유의해야 합니다.

- 데이터베이스 SAMPLE이 두 개의 세션을 사용하여 TSM으로 백업됩니다. 백업 에이전트 EDU ID를 찾아야 합니다. 이 ID는 db2pd -edus 또는 DB2 진단 로그를 통해 찾을 수 있습니다. 찾고 나면 다음을 발행할 수 있습니다.

```
db2pd -db sample -fvp 149
```

결과 출력은 다음과 같습니다.

```
-----
Fenced Vendor Process State Information:
-----
```

```
Backup:
-----
```

```
Media Controller(s):
-----
```

```
    EDU ID: 504
mediaSession: 1
mediaSeqNum: 0
Vendor EDU is available and running.
  startTime: 1155583315 20060814152155
  function: sqluvint
```

```
    EDU ID: 505
mediaSession: 2
mediaSeqNum: 0
Vendor EDU is available and running.
No vendor code being run.
```

이 출력은 DB2 Media Controller 0(EDU ID: 504)이 벤더 코드에 있는 반면, DB2 Media Controller 1(EDU ID: 505)에 분리(fenced) 벤더 프로세스가 있지만 벤더 코드를 실행하고 있지 않음을 알려줍니다. 이제, 함수를 너무 오래 실행했다고 느끼거나, 이 프로세스가 TSM 자원을 기다리면서 정지되었다고 판별한 경우, 다음을 발행하여 분리(fenced) 벤더 프로세스를 종료할 수 있습니다.

```
db2pd -db sample -fvp 149 term
```

결과 출력은 다음과 같습니다.

```
-----
Fenced Vendor Process State Information:
-----
```

```
Backup:
-----
```

```
Media Controller(s):
-----
```

```
    EDU ID: 504
mediaSession: 1
mediaSeqNum: 0
Vendor EDU is available and running.
  startTime: 1155583315 20060814152155
  function: sqluvint
This fenced vendor process has been sent a signal to terminate.
```

```
    EDU ID: 505
```

```
mediaSession: 2
  mediaSeqNum: 0
Vendor EDU is available and running.
No vendor code being run.
This fenced vendor process has been sent a signal to terminate.
```

이 출력은 위와 동일한 정보를 표시하지만, 두 분리(fenced) 벤더 프로세스 모두에 종료 요청이 전송되었고 잠시 후 종료됨을 알립니다.

## 사용 시 참고사항

다음 섹션은 다양한 db2pd 매개변수로 생성되는 출력을 설명합니다.

- **-agents**
- **-apinfo**
- **-applications**
- **-bufferpools**
- **-catalogcache**
- **-dbcfg**
- **-dbmcfg**
- **-dynamic**
- **-fcm**
- **-fmp**
- **-locks**
- **-logs**
- **-memblocks**
- **-mempools**
- **-memsets**
- **-osinfo**
- **-pages**
- **-recovery**
- **-reopt**
- **-reorgs**
- **-scansharing**
- **-serviceclasses**
- **-static**
- **-statisticscache**
- **-storagepaths**
- **-sysplex**
- **-tablespaces**
- **-tcbstats**
- **-temptable**
- **-thresholds**

- -transactions
- -wlocks
- -workactionsets
- -workclasssets
- -workloads

## **-agents** 매개변수

-agents 매개변수의 경우 다음 정보가 리턴됩니다.

### **AppHandl**

노드 및 인덱스를 포함하는 응용프로그램 핸들.

### **AgentPid**

에이전트 프로세스의 프로세스 ID.

### **Priority**

에이전트의 우선순위.

**Type** 에이전트의 유형.

**상태** 에이전트의 상태.

### **ClientPid**

클라이언트 프로세스의 프로세스 ID.

### **Userid**

에이전트를 실행 중인 사용자 ID.

### **ClientNm**

클라이언트 프로세스의 이름.

### **Rowsread**

에이전트가 읽은 행 수.

### **Rowswrtn**

에이전트가 기록한 행 수.

### **LkTmOt**

에이전트에 대한 잠금 시간종료 설정.

## **-apinfo** 매개변수

-apinfo 매개변수의 경우 다음 정보가 리턴됩니다.

### **AppHandl**

노드 및 인덱스를 포함하는 응용프로그램 핸들.

### **Application PID**

응용프로그램의 프로세스 ID.

**Application Node Name**

응용프로그램 노드의 이름.

**IP Address**

데이터베이스 연결이 설정된 IP 주소.

**Connection Start Time**

응용프로그램 연결이 시작된 시간소인.

**Client User ID**

클라이언트 사용자 ID.

**System Auth ID**

연결의 시스템 권한 부여 ID.

**Coordinator EDU ID**

응용프로그램에 대한 코디네이터 에이전트의 EDU ID.

**Coordinator Partition**

응용프로그램에 대한 코디네이터 에이전트의 파티션 번호.

**Number of Agents**

응용프로그램 대신 작동 중인 에이전트 수.

**Locks timeout value**

응용프로그램에 대한 잠금 시간종료 값.

**Locks Escalation**

잠금 에스컬레이션 플래그는 응용프로그램이 사용한 잠금이 에스컬레이트되는지 여부를 표시합니다.

**Workload ID**

워크로드 ID.

**Workload Occurrence ID**

워크로드 어커런스 ID.

**Trusted Context**

연결이 내재된 트러스트 연결이거나 명시적 트러스트 연결인 경우 연결과 연관되는 트러스트된 컨텍스트의 이름.

**Connection Trust Type**

연결 트러스트 유형. non-trusted, implicit trusted 또는 explicit trusted 연결 중 하나입니다.

**Role Inherited**

트러스트된 연결을 통해 상속된 역할(있는 경우).

**Application Status**

응용프로그램의 상태.



**Application Name**

응용프로그램의 이름.

**응용프로그램 ID**

응용프로그램 ID. 이 값은 **appl\_id** 모니터 요소 데이터와 같습니다. 이 값을 해석하는 방법에 대한 세부사항 정보는 『appl\_id - 응용프로그램 ID 모니터 요소』를 참조하십시오.

**UOW-ID**

응용프로그램의 현재 UOW ID.

**Activity ID**

UOW 내의 활동 ID.

**Package Schema**

패키지 스키마.

**Package Name**

패키지 이름.

**Package Version**

패키지 버전.

**Section Number**

SQL문의 섹션 번호.

**SQL Type**

SQL 유형: dynamic 또는 static.

**Isolation**

응용프로그램에 대해 설정된 분리 모드.

**Statement Type**

명령문 조작의 유형(예: DML, DDL).

**명령문**

SQL문.

**ClientUserID**

트랜잭션의 클라이언트 사용자 ID로 tpmon\_client\_userid(TP 모니터 클라이언트 사용자 ID 모니터 요소)와 동일.

**ClientWrkstnName**

트랜잭션의 클라이언트 워크스테이션 이름으로 tpmon\_client\_wkstn(TP 모니터 클라이언트 워크스테이션 이름 모니터 요소)과 동일.

**ClientAppName**

트랜잭션을 구동하는 클라이언트 응용프로그램 이름으로 tpmon\_client\_app(TP 모니터 클라이언트 응용프로그램 모니터 요소)와 동일.

### **ClientAcctng**

트랜잭션을 구동하는 클라이언트의 어카운팅 문자열로 tpmon\_acc\_str(TP 모니터 클라이언트 어카운팅 문자열 모니터 요소)과 동일.

db2pd -apinfo 명령의 샘플 출력을 참조하십시오.

### **-applications 매개변수**

**-applications** 매개변수의 경우 다음 정보가 리턴됩니다.

#### **AppHandl**

노드 및 인덱스를 포함하는 응용프로그램 핸들.

#### **NumAgents**

응용프로그램 대신 작동 중인 에이전트 수.

#### **CoorPid**

응용프로그램에 대한 코디네이터 에이전트의 프로세스 ID.

#### **Status**

응용프로그램의 상태.

**Appid** 응용프로그램 ID. 이 값은 **appl\_id** 모니터 요소 데이터와 같습니다. 이 값을 해석하는 방법에 대한 세부사항 정보는 **appl\_id** 모니터 요소에 대한 문서를 참조하십시오.

#### **ClientIPAddress**

데이터베이스 연결이 설정된 IP 주소.

#### **EncryptionLvl**

연결에 사용되는 데이터 스트림 암호화. NONE, LOW 또는 HIGH 중 하나입니다. NONE은 데이터 스트림 암호화가 사용되지 않음을 의미합니다. LOW는 데이터베이스 서버 **authentication** 유형이 DATA\_ENCRYPT로 설정됨을 의미합니다. HIGH는 SSL이 사용됨을 의미합니다.

#### **SystemAuthID**

연결의 시스템 권한 부여 ID.

#### **ConnTrustType**

연결 트러스트 유형. non-trusted, implicit trusted 또는 explicit trusted 연결 중 하나입니다.

#### **TrustedContext**

연결이 내재된 트러스트 연결이거나 명시적 트러스트 연결인 경우 연결과 연관되는 트러스트된 컨텍스트의 이름.

#### **RoleInherited**

트러스트된 연결을 통해 상속된 역할(있는 경우).

## **-bufferpools 매개변수**

**-bufferpools** 매개변수의 경우 다음 정보가 리턴됩니다.

### **First Active Pool ID**

첫 번째 활성 버퍼 풀의 ID.

### **Max Bufferpool ID**

모든 활성 버퍼 풀의 최대 ID.

### **Max Bufferpool ID on Disk**

디스크에 정의된 모든 버퍼 풀의 최대 ID.

### **Num Bufferpools**

사용 가능한 버퍼 풀 수.

**ID** 버퍼 풀의 ID.

**이름** 버퍼 풀의 이름.

### **PageSz**

버퍼 풀 페이지의 크기.

### **PA-NumPgs**

버퍼 풀의 페이지 영역에 있는 페이지 수.

### **BA-NumPgs**

버퍼 풀의 블록 영역에 있는 페이지 수. 버퍼 풀이 블록 기반 입출력에 대해 사용 가능하지 않으면 이 값은 0입니다.

### **BlkSize**

버퍼 풀의 블록 영역에서 블록의 블록 크기. 버퍼 풀이 블록 기반 입출력에 대해 사용 가능하지 않으면 이 값은 0입니다.

### **NumTbsp**

버퍼 풀을 사용하는 테이블 스페이스 수.

### **PgsLeft**

크기가 감소되는 경우 버퍼 풀에서 제거할 남겨진 페이지 수.

### **CurrentSz**

페이지에서 버퍼 풀의 현재 크기.

### **PostAlter**

버퍼 풀이 재시작될 때 페이지에서 버퍼 풀의 크기.

### **SuspdTSCt**

현재 입출력이 일시중단된 버퍼 풀에 맵핑된 테이블 스페이스 수. 모든 버퍼 풀에 대해 0이 리턴되는 경우, 데이터베이스 입출력이 일시중단되지 않습니다.

**DatLRds**

버퍼 풀 데이터 논리적 읽기. 일반 및 대형 테이블 스페이스에 대한 버퍼 풀 (논리)에서 요청된 데이터 페이지 수를 표시합니다.

**DatPRds**

버퍼 풀 데이터 실제 읽기. 일반 및 대형 테이블 스페이스에 대한 테이블 스페이스 컨테이너(실제)에서 읽은 데이터 페이지 수를 표시합니다.

**HitRatio**

$1 - \text{DatPRds} / \text{DatLRds}$  공식을 사용한 버퍼 풀의 데이터 페이지 사용 비율.

**TmpDatLRds**

버퍼 풀 임시 데이터 논리적 읽기. 임시 테이블 스페이스에 대한 버퍼 풀(논리)에서 요청된 데이터 페이지 수를 표시합니다.

**TmpDatPRds**

버퍼 풀 임시 데이터 실제 읽기. 임시 테이블 스페이스에 대한 테이블 스페이스 컨테이너(실제)에서 읽은 데이터 페이지 수를 표시합니다.

**HitRatio**

$1 - \text{TmpDatPRds} / \text{TmpDatLRds}$  공식을 사용한 버퍼 풀의 임시 데이터 페이지 사용 비율.

**IdxLRds**

버퍼 풀 인덱스 논리적 읽기. 일반 및 대형 테이블 스페이스에 대한 버퍼 풀 (논리)에서 요청된 인덱스 페이지 수를 표시합니다.

**IdxPRds**

버퍼 풀 인덱스 실제 읽기. 일반 및 대형 테이블 스페이스에 대한 테이블 스페이스 컨테이너(실제)에서 읽은 인덱스 페이지 수를 표시합니다.

**HitRatio**

$1 - \text{IdxPRds} / \text{IdxLRds}$  공식을 사용한 버퍼 풀의 인덱스 페이지 사용 비율.

**TmpIdxLRds**

버퍼 풀 임시 인덱스 논리적 읽기. 임시 테이블 스페이스에 대한 버퍼 풀(논리)에서 요청된 인덱스 페이지 수를 표시합니다.

**TmpIdxPRds**

버퍼 풀 임시 인덱스 실제 읽기. 임시 테이블 스페이스에 대한 테이블 스페이스 컨테이너(실제)에서 읽은 인덱스 페이지 수를 표시합니다.

**HitRatio**

$1 - \text{TmpIdxPRds} / \text{TmpIdxLRds}$  공식을 사용한 버퍼 풀의 임시 인덱스 페이지 사용 비율.

**DataWrts**

버퍼 풀 데이터 쓰기. 버퍼 풀 데이터 페이지가 실제로 디스크에 기록된 횟수를 표시합니다.

**IdxWrts**

버퍼 풀 인덱스 쓰기. 버퍼 풀 인덱스 페이지가 실제로 디스크에 기록된 횟수를 표시합니다.

**DirRds**

데이터베이스로부터 직접 읽기. 버퍼 풀을 사용하지 않는 읽기 조작 수.

**DirRdReqs**

직접 읽기 요청. 하나 이상의 데이터 섹터 직접 읽기를 수행하기 위한 요청 수.

**DirRdTime**

직접 읽기 시간. 직접 읽기를 수행하기 위해 필요한 경과 시간(밀리초).

**DirWrts**

데이터베이스에 대한 직접 쓰기. 버퍼 풀을 사용하지 않는 쓰기 조작 수.

**DirWrtReqs**

직접 쓰기 요청. 하나 이상의 데이터 섹터 직접 쓰기를 수행하기 위한 요청 수.

**DirWrtTime**

직접 쓰기 시간. 직접 쓰기를 수행하기 위해 필요한 경과 시간(밀리초).

**AsDatRds**

버퍼 풀 비동기 데이터 읽기. 모든 유형의 테이블 스페이스에 대해 비동기 EDU(engine dispatchable unit)별로 테이블 스페이스 컨테이너(실제)에서 읽은 데이터 페이지 수를 표시합니다.

**AsDatRdReq**

버퍼 풀 비동기 읽기 요청. 비동기 읽기 요청 수.

**AsIdxRds**

버퍼 풀 비동기 인덱스 읽기. 모든 유형의 테이블 스페이스에 대해 비동기 EDU(engine dispatchable unit)별로 테이블 스페이스 컨테이너(실제)에서 읽은 인덱스 페이지 수를 표시합니다.

**AsIdxRdReq**

버퍼 풀 비동기 인덱스 읽기 요청. 인덱스 페이지에 대한 비동기 읽기 요청 수.

**AsRdTime**

버퍼 풀 비동기 읽기 시간. 모든 유형의 테이블 스페이스에 대해 비동기 EDU(engine dispatchable unit)별로 테이블 스페이스 컨테이너(실제)에서 데이터 및 인덱스 페이지 읽기에 소비한 총 시간량을 표시합니다. 이 값은 마이크로초 단위로 제공됩니다.

**AsDatWrts**

버퍼 풀 비동기 데이터 쓰기. 비동기 페이지 클리너 또는 프리페처에 의해 버퍼 풀 데이터 페이지가 실제로 디스크에 기록된 횟수. 프리페처는 디스크에 더티 페이지를 기록하여 프리페치되는 페이지에 대해 스페이스를 만들 수 있습니다.

**AsIdxWrts**

버퍼 풀 비동기 인덱스 쓰기. 비동기 페이지 클리너 또는 프리페처에 의해 버퍼 풀 인덱스 페이지가 실제로 디스크에 기록된 횟수. 프리페처는 디스크에 더티 페이지를 기록하여 프리페치되는 페이지에 대해 스페이스를 만들 수 있습니다.

**AsWrtTime**

버퍼 풀 비동기 쓰기 시간. 데이터베이스 관리 프로그램 페이지 클리너가 버퍼 풀에서 디스크로 데이터 또는 인덱스 페이지를 기록하는 데 소비한 총 경과 시간.

**TotRdTime**

총 버퍼 풀 실제 읽기 시간. 모든 유형의 테이블 스페이스에 대해 테이블 스페이스 컨테이너(실제)에서 데이터 및 인덱스 페이지 읽기에 소비한 총 시간량을 표시합니다. 이 값은 마이크로초 단위로 제공됩니다.

**TotWrtTime**

총 버퍼 풀 실제 쓰기 시간. 버퍼 풀에서 디스크로 데이터 또는 인덱스 페이지를 실제로 기록하는 데 소비한 총 시간량을 제공합니다. 경과 시간은 마이크로초로 제공됩니다.

**VectIORds**

벡터 입출력에 의한 총 페이지 읽기 수. 벡터 입출력이 버퍼 풀의 페이지 영역으로 읽은 총 페이지 수.

**VectIOReq**

벡터 입출력 요청 수. 벡터 입출력 요청 수를 표시합니다. 즉, DB2 데이터베이스 제품이 버퍼 풀의 페이지 영역으로 페이지를 순차적으로 프리페치하는 횟수입니다.

**BlockIORds**

블록 입출력에 의한 총 페이지 읽기 수. 블록 입출력이 버퍼 풀의 페이지 영역으로 읽은 총 페이지 수.

**BlockIOReq**

블록 입출력 요청 수. 블록 입출력 요청 수를 표시합니다. 즉, DB2 데이터베이스 제품이 버퍼 풀의 블록 영역으로 페이지를 순차적으로 프리페치하는 횟수입니다.

**PhyPgMaps**

실제 페이지 맵 수. 실제 페이지 맵 수를 표시합니다.

**FilesClose**

닫힌 데이터베이스 파일 수. 닫힌 총 데이터베이스 파일 수입니다.

**NoVictAvl**

버퍼 풀 희생(victim) 없음 버퍼. 에이전트가 사용 가능한 사전 선택 희생(victim) 버퍼를 가지고 있지 않았던 횟수.

**UnRdPFetch**

읽지 않은 프리페치 페이지. 프리페처가 읽은, 사용한 적이 없는 페이지 수를 표시합니다.

**-catalogcache 매개변수**

-catalogcache 매개변수의 경우 다음 정보가 리턴됩니다.

카탈로그 캐시:

**Configured Size**

catalogcache\_sz 데이터베이스 구성 매개변수에 의해 지정된 바이트 수.

**Current Size**

카탈로그 캐시에서 사용된 현재 바이트 수.

**Maximum Size**

캐시에 사용 가능한 최대 메모리 양(최대 데이터베이스 전역 메모리까지).

**High Water Mark**

처리 중 접근한 가장 큰 실제 크기.

**SYSTABLES:**

스키마

테이블의 스키마 규정자.

이름 테이블의 이름

Type 테이블의 유형.

TableID

테이블 ID

TbpaceID

테이블이 있는 테이블 스페이스의 ID.

LastRefID

테이블을 참조한 마지막 프로세스 ID.

CatalogCache LoadingLock

캐시 항목에 대한 카탈로그 캐시 로딩 잠금의 이름.

**CatalogCache UsageLock**

캐시 항목에 대한 사용 잠금의 이름.

**Sts** 항목의 상태. 가능한 값은 다음과 같습니다.

- V(유효함).
- I(유효하지 않음).
- S(소프트 무효화. 카탈로그 캐시 항목은 실시간 통계 콜렉션에 의해 통계가 갱신된 경우에 소프트웨어 무효화 상태가 됩니다. 이 카탈로그 캐시 항목은 여전히 데이터베이스 에이전트가 사용할 수 있지만 새 카탈로그 캐시 요청에 사용하기에는 유효하지 않습니다. 소프트웨어 무효화 항목이 더 이상 사용 중이 아니면 제거됩니다. 새 카탈로그 캐시 요청은 유효한 항목을 사용합니다.)

**SYSRTNS:****RoutineID**

루틴 ID.

스키마

루틴의 스키마 규정자.

이름 루틴의 이름.

**LastRefID**

루틴을 참조한 마지막 프로세스 ID.

**CatalogCache LoadingLock**

캐시 항목에 대한 카탈로그 캐시 로딩 잠금의 이름.

**CatalogCache UsageLock**

캐시 항목에 대한 사용 잠금의 이름.

**Sts** 항목의 상태. 가능한 값은 다음과 같습니다.

- V(유효함).
- I(유효하지 않음).

**SYSRTNS\_PROCSCHEMAS:****RtnName**

루틴의 이름.

**ParmCount**

루틴의 매개변수 수.

**LastRefID**

PROCSCHEMAS 항목을 참조한 마지막 프로세스 ID.

**CatalogCache LoadingLock**

캐시 항목에 대한 카탈로그 캐시 로딩 잠금의 이름.



**CatalogCache UsageLock**

캐시 항목에 대한 사용 잠금의 이름.

**Sts** 항목의 상태. 가능한 값은 다음과 같습니다.

- V(유효함).
- I(유효하지 않음).

**SYSDATATYPES:**

**TypID** 유형 ID.

**LastRefID**

유형을 참조한 마지막 프로세스 ID.

**CatalogCache LoadingLock**

캐시 항목에 대한 카탈로그 캐시 로딩 잠금의 이름.

**CatalogCache UsageLock**

캐시 항목에 대한 사용 잠금의 이름.

**Sts** 항목의 상태. 가능한 값은 다음과 같습니다.

- V(유효함).
- I(유효하지 않음).

**SYSCODEPROPERTIES:****LastRefID**

SYSCODEPROPERTIES 항목을 참조한 마지막 프로세스 ID.

**CatalogCache LoadingLock**

캐시 항목에 대한 카탈로그 캐시 로딩 잠금의 이름.

**CatalogCache UsageLock**

캐시 항목에 대한 사용 잠금의 이름.

**Sts** 항목의 상태. 가능한 값은 다음과 같습니다.

- V(유효함).
- I(유효하지 않음).

**SYSNODEGROUPS:****PMapID**

분산 맵 ID.

**RBaID**

데이터 분산에 사용된 분산 맵의 ID.

**CatalogCache LoadingLock**

캐시 항목에 대한 카탈로그 캐시 로딩 잠금의 이름.

**CatalogCache UsageLock**

캐시 항목에 대한 사용 잠금의 이름.

- Sts** 항목의 상태. 가능한 값은 다음과 같습니다.
- V(유효함).
  - I(유효하지 않음).

**SYSDBAUTH:**

**AuthID**

권한 부여 ID(*authid*).

**AuthType**

권한 부여 유형.

**LastRefID**

캐시 항목을 참조할 마지막 프로세스 ID.

**CatalogCache LoadingLock**

캐시 항목에 대한 카탈로그 캐시 로딩 잠금의 이름.

**SYSRTNAUTH:**

**AuthID**

권한 부여 ID(*authid*).

**AuthType**

권한 부여 유형.

**스키마**

루틴의 스키마 규정자.

**RoutineName**

루틴의 이름.

**RtnType**

루틴의 유형.

**CatalogCache LoadingLock**

캐시 항목에 대한 카탈로그 캐시 로딩 잠금의 이름.

**SYSROLEAUTH:**

**AuthID**

권한 부여 ID(*authid*).

**AuthType**

권한 부여 유형.

**Roleid**

권한 부여 ID가 역할인 경우 역할 ID.

**LastRefID**

캐시 항목을 참조할 마지막 프로세스 ID.

### **CatalogCache LoadingLock**

캐시 항목에 대한 카탈로그 캐시 로딩 잠금의 이름.

db2pd -catalogcache 명령의 샘플 출력을 참조하십시오.

### **-dbcfg parameter**

-dbcfg 매개변수의 경우, 데이터베이스 구성 매개변수의 현재 값이 리턴됩니다.

### **-dbmcfg 매개변수**

-dbmcfg 매개변수의 경우, 데이터베이스 관리 프로그램 구성 매개변수의 현재 값이 리턴됩니다.

### **-dynamic 매개변수**

-dynamic 매개변수의 경우 다음 정보가 리턴됩니다.

동적 캐시:

#### **Current Memory Used**

패키지 캐시에 사용되는 바이트 수.

#### **Total Heap Size**

패키지 캐시에 대해 내부적으로 구성된 바이트 수.

#### **Cache Overflow flag state**

패키지 캐시가 오버플로우 상태인지 여부를 표시하기 위한 플래그.

#### **Number of references**

패키지 캐시의 동적 분할 영역이 참조된 횟수.

#### **Number of Statement Inserts**

패키지 캐시로의 명령문 삽입 수.

#### **Number of Statement Deletes**

패키지 캐시로부터의 명령문 삭제 수.

#### **Number of Variation Inserts**

패키지 캐시로의 변형 삽입 수.

#### **Number of statements**

패키지 캐시의 명령문 수.

동적 SQL문:

#### **AnchID**

해시 앵커(anchor) ID

#### **StmtID**

명령문 ID.

**NumEnv**

명령문에 속하는 환경 수.

**NumVar**

명령문에 속하는 변형 수.

**NumRef**

명령문이 참조된 횟수.

**NumExe**

명령문이 실행된 횟수.

**Text** SQL문의 텍스트.

**동적 SQL 환경:****AnchID**

해시 앵커(anchor) ID

**StmtID**

명령문 ID.

**EnvID** 환경 ID.

**Iso** 환경의 분리 레벨.

**QOpt** 환경의 쿼리 최적화 레벨.

**Blk** 환경의 블로킹 인수.

**동적 SQL 변형:****AnchID**

해시 앵커(anchor) ID

**StmtID**

해당 변형의 명령문 ID

**EnvID** 해당 변형의 환경 ID

**VarID** 변형 ID

**NumRef**

해당 변형이 참조된 횟수.

**Typ** 변형 섹션에 대한 내부 명령문 유형 값.

**Lockname**

변형 잠금 이름.

**-fcm 매개변수**

-fcm 매개변수의 경우 다음 정보가 리턴됩니다.

**FCM 사용 통계:**

**Total Buffers**

모든 사용 가능한 버퍼 및 사용 중 버퍼를 포함한 총 버퍼 수.

**Free Buffers**

사용 가능한 버퍼 수.

**Buffers LWM**

가장 낮은 사용 가능한 버퍼 수.

**Total Channels**

모든 사용 가능한 채널 및 사용 중 채널을 포함한 총 채널 수.

**Free Channels**

사용 가능한 채널 수.

**Channels LWM**

가장 낮은 사용 가능한 채널 수.

**Total Sessions**

모든 사용 가능한 세션 및 사용 중 세션을 포함한 총 세션 수.

**Free Sessions**

사용 가능한 세션 수.

**Sessions LWM**

가장 낮은 사용 가능한 세션 수.

**Partition**

데이터베이스 파티션 서버 번호.

**Bufs Sent**

db2pd 명령이 실행 중인 데이터베이스 파티션 서버에서 출력에 식별된 데이터베이스 파티션 서버로 보낸 총 FCM 버퍼 수.

**Bufs Recv**

db2pd 명령이 실행 중인 데이터베이스 파티션 서버가 출력에 식별된 데이터베이스 파티션 서버에서 수신한 총 FCM 버퍼 수.

**Status**

db2pd 명령이 실행 중인 데이터베이스 파티션 서버와 출력에 식별된 데이터베이스 파티션 서버 사이의 논리적 연결 상태. 가능한 값은 다음과 같습니다.

- Inactive: 데이터베이스 파티션 서버가 DPF(Database Partitioning Feature) 구성에 정의되어 있지만 현재 비활성 상태입니다(예를 들어, 사용자가 파티션을 중지했음).
- Active: 데이터베이스 파티션 서버가 활성 상태입니다.
- Undefined: 데이터베이스 파티션 서버가 DPF(Database Partitioning Feature) 구성에 정의되어 있지 않습니다. 이는 오류를 표시합니다.

- Unknown: 데이터베이스 파티션 서버가 알 수 없는 상태에 있습니다. 이는 오류를 표시합니다.

#### 버퍼 현재 소비

##### **AppHandl**

노드 및 인덱스를 포함하는 응용프로그램 핸들.

##### **TimeStamp**

응용프로그램 핸들 사용에 대한 고유 ID.

##### **Buffers In-use**

응용프로그램이 현재 사용 중인 버퍼 수.

#### 채널 현재 소비

##### **AppHandl**

노드 및 인덱스를 포함하는 응용프로그램 핸들.

##### **TimeStamp**

응용프로그램 핸들 사용에 대한 고유 ID.

##### **Channels In-use**

응용프로그램이 현재 사용 중인 채널 수.

#### 버퍼 소비 HWM

##### **AppHandl**

노드 및 인덱스를 포함하는 응용프로그램 핸들.

##### **TimeStamp**

응용프로그램 핸들 사용에 대한 고유 ID.

##### **Buffers Used**

인스턴스 시작 이후에 응용프로그램에 사용된 상위 워터 마크(water mark) 버퍼 수.

#### 채널 소비 HWM

##### **AppHandl**

노드 및 인덱스를 포함하는 응용프로그램 핸들.

##### **TimeStamp**

응용프로그램 핸들 사용에 대한 고유 ID.

##### **Channels Used**

인스턴스 시작 이후에 응용프로그램에 사용된 상위 워터 마크(water mark) 채널 수.

## -fmp 매개변수

-fmp 매개변수의 경우 다음 정보가 리턴됩니다.

- Pool Size: FMP 풀의 현재 FMP 프로세스 수
- Max Pool Size: FMP 풀의 최대 FMP 프로세스 수.
- Keep FMP: **keepfenced** 데이터베이스 관리 프로그램 구성 매개변수의 값.
- Initialized: FMP가 초기화됩니다. 가능한 값은 Yes 및 No입니다.
- Trusted Path: 트러스트된 프로시저의 경로
- Fenced User: 분리(fenced) 사용자 ID

### FMP 프로세스:

- FmpPid: FMP 프로세스의 프로세스 ID.
- Bit: 비트 모드. 값은 32비트 또는 64비트입니다.
- Flags: FMP 프로세스에 대한 상태 플래그. 가능한 값은 다음과 같습니다.
  - 0x00000000 - JVM 초기화됨
  - 0x00000002 - 스레드됨
  - 0x00000004 - 페더레이티드 랩퍼 실행에 사용됨
  - 0x00000008 - Health Monitor에 사용됨
  - 0x00000010 - 종료하도록 표시되며 새 태스크를 승인하지 않음
  - 0x00000020 - db2sysc가 정리하도록 표시됨
  - 0x00000040 - 에이전트 정리에 대해 표시됨
  - 0x00000100 - 프로세스에 대한 모든 ipc가 제거되었음
  - 0x00000200 - .NET 런타임이 초기화됨
  - 0x00000400 - 디버깅에 대해 JVM이 초기화됨
  - 0x00000800 - 종료 플래그
- ActiveTh: FMP 프로세스에서 실행 중인 활성 스레드 수
- PooledTh: FMP 프로세스가 보유하는 풀된 스레드 수
- Active: FMP 프로세스의 활성 상태. 값은 Yes 및 No입니다.

### 활성 스레드:

- FmpPid: 활성 스레드를 소유하는 FMP 프로세스 ID
- EduPid: 스레드가 작동 중인 EDU 프로세스 ID
- ThreadId: 활성 스레드 ID

### 풀된 스레드:

- FmpPid: 풀된 스레드를 소유하는 FMP 프로세스 ID
- ThreadId: 풀된 스레드 ID

## **-locks 매개변수**

**-locks** 매개변수의 경우 다음 정보가 리턴됩니다.

### **TranHdl**

잠금을 요청하는 트랜잭션 핸들.

### **Lockname**

잠금의 이름.

**Type** 잠금의 유형. 가능한 값은 다음과 같습니다.

- Row
- Pool
- Partition
- Table
- AlterTab
- ObjectTab
- OnlBackup
- DMS Seq
- Internal P
- Internal V
- Key Value
- No Lock
- Block Lock
- LOG Release
- LF Release
- LFM File
- LOB/LF 4K
- APM Seq
- Tbsp Load
- Table Part
- DJ UserMap
- DF NickNm
- CatCache
- OnlReorg
- Buf Pool

**Mode** 잠금 모드. 가능한 값은 다음과 같습니다.

- IS
- IX
- S
- SIX
- X



- IN
- Z
- U
- NS
- NW

**Sts** 잠금 상태. 가능한 값은 다음과 같습니다.

- G(권한 부여됨)
- C(변환 중)
- W(대기 중)

**Owner** 잠금을 소유하는 트랜잭션 핸들.

**Dur** 잠금의 지속기간.

#### **HoldCount**

잠금에 제출된 보류 수. 보류가 있는 잠금은 트랜잭션이 커밋될 때 해제되지 않습니다.

**Att** 잠금의 속성. 가능한 값은 다음과 같습니다.

- 0x01 사용 가능성 대기.
- 0x02 에스컬레이션에 의해 획득됨.
- 0x04 블록 "내의" RR 잠금.
- 0x08 잠금 삽입.
- 0x10 RR 스캔에 의한 잠금.
- 0x20 행 잠금 갱신/삭제.
- 0x40 새 잠금 요청 허용.
- 0x80 새 잠금 요청자.

#### **ReleaseFlg**

잠금 릴리스 플래그. 가능한 값은 다음과 같습니다.

- 0x80000000 SQL 컴파일러에 의한 잠금.
- 0x40000000 고유하지 않은, 추적되지 않은 잠금.

**rrIID** RR 잠금(0x10 속성 이상)을 획득한 인덱스의 IID입니다. 가능한 값은 다음과 같습니다.

- 0 단일, 특정 인덱스(또는 RR 잠금)와 관련이 없습니다.
- <>0 특정 인덱스 IID는 잠금을 획득하는 데 사용됩니다.

### **-logs 매개변수**

**-logs** 매개변수의 경우 다음 정보가 리턴됩니다.

#### **Current Log Number**

현재 활성화 로그의 번호.

**Pages Written**

현재 로그에 기록되는 현재 페이지.

**Cur 커미트 디스크 로그 읽기**

디스크(대 로그 버퍼)에서 로그 읽기로 검색되어 현재 커미트된 행 버전의 횟수.

**Cur 커미트 전체 로그 읽기**

로그(로그 버퍼 및 디스크)에서 검색되어 현재 커미트된 행 버전의 총 횟수.

**Method 1 Archive Status**

최근 로그 아카이브 시도의 결과. 가능한 값은 Success 또는 Failure입니다.

**Method 1 Next Log to Archive**

아카이브할 다음 로그 파일.

**Method 1 First Failed**

아카이브되지 않은 첫 번째 로그 파일.

**Method 2 Archive Status**

최근 로그 아카이브 시도의 결과. 가능한 값은 Success 또는 Failure입니다.

**Method 2 Next Log to Archive**

아카이브할 다음 로그 파일.

**Method 2 First Failed**

아카이브되지 않은 첫 번째 로그 파일.

**StartLSN**

시작 로그 시퀀스 번호.

상태 0x00000020은 로그가 아카이브되었음을 표시합니다.

**Size** 로그 Extent 크기(페이지 수).

**Pages** 로그의 페이지 수.

**Filename**

로그의 파일 이름.

**-memblocks 매개변수**

**-memblocks** 매개변수의 경우 세 개의 출력 섹션이 있습니다. 메모리 세트에 대한 개별 블록, 메모리 풀별로 그룹화된 정렬된 총계 및 메모리 세트에 대해 정렬된 총계입니다.

메모리 블록:

**PoolID**

메모리 블록을 소유하는 메모리 풀 ID.

**PoolName**

메모리 블록을 소유하는 메모리 풀 이름.

**BlockAge**

메모리 블록의 블록 유효 기간. 블록이 할당될 때 지정되는 증가하는 카운터입니다.

**Size** 메모리 블록의 크기(바이트).

**I** 할당 유형. 값 1은 블록이 개별적으로 사용 가능하도록 해제됨을 의미하고 0은 풀에 대해 해제됨을 의미합니다.

**LOC** 메모리 블록을 할당한 코드 라인.

**File** 블록이 할당된 파일 이름 해시 값.

각 메모리 풀에 대해 보고되는 정렬된 총계:

**PoolID**

메모리 블록을 소유하는 메모리 풀 ID.

**PoolName**

메모리 블록을 소유하는 메모리 풀 이름.

**TotalSize**

코드 및 파일의 동일 라인에서 할당된 총 블록 크기(바이트).

**TotalCount**

코드 및 파일의 동일 라인에서 할당된 블록 수.

**LOC** 메모리 블록을 할당한 코드 라인.

**File** 블록이 할당된 파일 이름 해시 값.

각 메모리 세트에 대해 보고되는 정렬된 총계:

**PoolID**

메모리 블록을 소유하는 메모리 풀 ID.

**PoolName**

메모리 블록을 소유하는 메모리 풀 이름.

**TotalSize**

코드 및 파일의 동일 라인에서 할당된 총 블록 크기(바이트).

**%Bytes**

코드 및 파일의 동일 라인에서 할당된 바이트 백분율.

**TotalCount**

코드 및 파일의 동일 라인에서 할당된 블록 수.

**%Count**

코드 및 파일의 동일 라인에서 할당된 계수 백분율.

**LOC** 메모리 블록을 할당한 코드 라인.

**File** 블록이 할당된 파일 이름 해시 값.

**-mempools 매개변수**

**-mempools** 매개변수의 경우, 다음 정보가 리턴됩니다(모든 크기는 바이트 단위로 지정됨).

**MemSet**

메모리 풀을 소유하는 메모리 세트.

**PoolName**

버퍼 풀의 이름.

**Id** 메모리 풀 ID.

**Overhead**

풀 구조에 필요한 내부 오버헤드.

**LogSz** 현재 총 풀 메모리 요청 수.

**LogUpBnd**

현재 논리적 크기 상한.

**LogHWM**

논리적 크기 상위 워터 마크(water mark).

**PhySz** 논리적 크기에 필요한 실제 메모리.

**PhyUpBnd**

현재 실제 크기 상한.

**PhyHWM**

처리 중 접근한 가장 큰 실제 크기.

**Bnd** 내부 바운딩 전략.

**BlkCnt**

메모리 풀의 현재 할당 블록 수.

**CfgParm**

보고하는 풀의 크기를 선언하는 구성 매개변수.

**-memsets 매개변수**

**-memsets** 매개변수의 경우 다음 정보가 리턴됩니다.

이름 메모리 세트의 이름.

**Address**

메모리 세트의 주소.

**Id** 메모리 세트 ID.

**Size(Kb)**

메모리 세트의 크기(KB).

**Key** 메모리 세트 키(UNIX 운영 체제의 경우).

**DBP** 메모리 세트를 소유하는 데이터베이스 파티션 서버.

**Type** 메모리 세트의 유형.

**Unrsv(Kb)**

특정 풀에 대해 메모리가 예약되지 않습니다. 필요하다면 세트의 임의 풀이 이 메모리를 사용할 수 있습니다.

**Used(Kb)**

현재 메모리 풀에 할당된 메모리.

**Cmt(Kb)**

DB2 데이터베이스가 커미트한 모든 메모리로, 실제 RAM이나 페이징 스페이스, 또는 둘 다를 점유합니다.

**HWM(Kb)**

메모리 풀에 할당된 최대 메모리.

**Uncmt(Kb)**

현재 사용하고 있지 않고, DB2 데이터베이스에 의해 언커미트되도록 표시된 메모리. 운영 체제에 따라, 이 메모리는 실제 RAM이나 페이징 스페이스, 또는 둘 다를 점유할 수 있습니다.

**-osinfo 매개변수**

-osinfo 매개변수의 경우 다음 정보가 리턴됩니다.

**CPU 정보: (Windows, AIX, HP-UX, Solaris 및 Linux 운영 체제의 경우)**

**TotalCPU**

총 CPU 수.

**OnlineCPU**

CPU 온라인 수.

**ConfigCPU**

구성된 CPU 수.

**Speed(MHz)**

CPU 속도(MHz).

### **HMTDegree**

하드웨어 멀티스레드를 지원하는 시스템은 운영 체제에 존재하는 것으로 표시될 프로세스 수를 표시하는 값을 리턴합니다. nonHMT 시스템에서 이 값은 항상 1입니다. HMT 시스템에서, TOTAL은 논리적 CPU 수를 반영합니다. 실제 CPU 수를 가져오려면, 총계를 THREADING DEGREE로 나누십시오.

### **Timebase**

시간대 레지스터 증분 빈도(Hz). Linux PPC에서만 지원됩니다.

**실제 메모리 및 스왑(MB): (Windows, AIX, HP-UX, Solaris 및 Linux 운영 체제의 경우)**

### **TotalMemTotal**

메모리 크기(MB).

### **FreeMem**

여유 메모리의 양(MB).

### **AvailMem**

제품에 사용 가능한 메모리의 양(MB).

### **TotalSwap**

스왑 영역 총량(MB).

### **FreeSwap**

여유 스왑 영역의 양(MB).

**가상 메모리(MB)(Windows, AIX, HP-UX 및 Solaris 운영 체제의 경우)**

**Total** 시스템의 가상 메모리 총량(MB).

### **Reserved**

예약된 가상 메모리 양(MB).

### **Available**

사용 가능한 가상 메모리 양(MB).

**Free** 여유 가상 메모리 양(MB).

**운영 체제 정보(Windows, AIX, HP-UX, Solaris 및 Linux 운영 체제의 경우)**

### **OSName**

운영 체제 소프트웨어의 이름.

### **NodeName**

시스템의 이름.

### **Version**

운영 체제의 버전.

**Machine**

머신 하드웨어 식별.

메시지 큐 정보(AIX, HP-UX 및 Linux 운영 체제의 경우)

**MsgSeg**

시스템 전반의 총 SysV 메시지 세그먼트 수.

**MsgMax**

시스템 전반의 메시지 최대 크기.

**MsgMap**

시스템 전반의 메시지 맵 항목 수.

**MsgMni**

시스템 전반의 시스템에 대한 메시지 큐 ID 수.

**MsgTql**

시스템 전반의 메시지 헤더 수.

**MsgMnb**

메시지 큐의 최대 바이트 수.

**MsgSsz**

메시지 세그먼트 크기.

공유 메모리 정보(AIX, HP-UX 및 Linux 운영 체제의 경우)

**ShmMax**

시스템 전반의 공유 메모리 세그먼트 최대 크기(바이트).

**ShmMin**

시스템 전반의 공유 메모리 세그먼트 최소 크기(바이트).

**ShmIds**

시스템 전반의 공유 메모리 ID 수.

**ShmSeg**

프로세스 전반의 프로세스당 최대 공유 메모리 세그먼트 수.

세마포어 정보: (AIX, HP-UX 및 Linux 운영 체제의 경우)

**SemMap**

시스템 전반의 세마포어 맵 항목 수.

**SemMni**

시스템 전반의 최대 세마포어 ID 수.

**SemMns**

시스템 전반의 시스템 최대 세마포어 수.

**SemMnu**

시스템 전반의 시스템 최대 실행 취소 구조 수.

**SemMsl**

시스템 전반의 ID당 최대 세마포어 수.

**SemOpm**

시스템 전반의 semop 호출당 최대 조작 수.

**SemUme**

프로세스 전반의 프로세스당 최대 실행 취소 구조 수.

**SemUsz**

시스템 전반의 실행 취소 구조 크기. semume에서 파생됩니다.

**SemVmx**

시스템 전반의 세마포어 최대값.

**SemAem**

시스템 전반의 최대 종료 시 조정 값.

**CPU 로드 정보(Windows, AIX, HP-UX, Solaris 및 Linux 운영 체제의 경우)****shortPeriod**

이전 1분 동안의 실행 가능한 프로세스 수

**mediumPeriod**

이전 5분 동안의 실행 가능한 프로세스 수

**longPeriod**

이전 15분 동안의 실행 가능한 프로세스 수

**디스크 정보****BkSz (bytes)**

파일 시스템 블록 크기(바이트)

**Total (bytes)**

디바이스의 총 바이트 수(바이트)

**Free (bytes)**

디바이스의 사용 가능한 바이트 수(바이트)

**Inodes**

총 inode 수

**FSID** 파일 시스템 ID**DeviceType**

디바이스 유형



**FSName**

파일 시스템 이름

**MountPoint**

파일 시스템의 마운트 지점

**-pages 매개변수**

-pages 매개변수의 경우 다음 정보가 페이지마다 리턴됩니다.

**BPID** 페이지를 포함하는 버퍼 풀 ID

**TbSpaceID**

페이지를 포함하는 테이블 스페이스 ID

**TbSpacePgNum**

테이블 스페이스 내의 논리 페이지 번호(DMS만)

**ObjID** 페이지를 포함하는 오브젝트 ID

**ObjPgNum**

오브젝트 내의 논리 페이지 번호

**ObjClass**

페이지에 포함된 오브젝트의 클래스. 가능한 값은 Perm, Temp, Reorg, Shadow 및 EMP입니다.

**ObjType**

페이지에 포함된 오브젝트의 유형. 가능한 값은 Data, Index, LongField, XMLData, SMP, LOB, LOBA 및 MDC\_BMP입니다.

**Dirty** 페이지가 더티한지 여부를 표시합니다. 가능한 값은 Y 및 N입니다. 페이지 출력의 요약 정보 섹션에서, 값은 더티 페이지 수를 표시합니다.

**Permanent**

페이지 출력의 요약 정보 섹션에서, 값은 PERMANENT 페이지 수를 표시합니다.

**Temporary**

페이지 출력의 요약 정보 섹션에서, 값은 TEMPORARY 페이지 수를 표시합니다.

**Prefetched**

페이지가 프리페치되었는지 여부를 표시합니다. 가능한 값은 Y 및 N입니다.

db2pd -pages 명령의 샘플 출력을 참조하십시오.

**-recovery 매개변수**

-recovery 매개변수의 경우 다음 정보가 리턴됩니다.

**Recovery Status**

내부 복구 상태

**Current Log**

복구 조작에 사용되는 현재 로그

**Current LSN**

현재 로그 시퀀스 번호

**Job Type**

수행 중인 복구의 유형. 가능한 값은 다음과 같습니다.

- 5: 응급 복구
- 6: 데이터베이스 또는 테이블 스페이스에 대한 롤 포워드 복구

**Job ID**

작업 ID.

**Job Start Time**

복구 조작이 시작된 시간

**Job Description**

복구 활동의 설명. 가능한 값은 다음과 같습니다.

- Tablespace Rollforward Recovery
- Database Rollforward Recovery
- Crash Recovery

**Invoker Type**

복구 조작이 호출된 방법. 가능한 값은 다음과 같습니다.

- User
- DB2

**Total Phases**

복구 조작을 완료하기 위해 필요한 단계 수

**Current phase**

복구 조작의 현재 단계

**Phase** 복구 조작에서 현재 단계의 번호

**Forward phase**

롤 포워드 복구의 첫 번째 단계. 이 단계를 REDO 단계라고도 합니다.

**Backward phase**

롤 포워드 복구의 두 번째 단계. 이 단계는 UNDO 단계라고도 합니다.

**Metric**

작업 단위(UOW). 가능한 값은 다음과 같습니다.

- 1: 바이트 수

- 2: Extent 수
- 3: 행 수
- 4: 페이지 수
- 5: 인덱스 수

#### **TotWkUnits**

복구 조작의 해당 단계에 대해 수행될 총 작업 단위(UOW) 수.

#### **TotCompUnits**

완료된 총 UOW 수

#### **-reopt 매개변수**

-reopt 매개변수의 경우 다음 정보가 리턴됩니다.

#### **Dynamic SQL Statements**

-dynamic을 참조하십시오.

#### **Dynamic SQL Environments**

-dynamic을 참조하십시오.

#### **Dynamic SQL Variations**

-dynamic을 참조하십시오.

#### **Reopt Values**

제공된 SQL문을 다시 최적화하기 위해 사용되는 변수에 대한 정보를 표시합니다. 사용되지 않은 변수에 대한 정보는 리턴되지 않습니다. 가능한 값은 다음과 같습니다.

#### **AnchID**

해시 앵커(anchor) ID

#### **StmtID**

해당 변형의 명령문 ID

**EnvID** 해당 변형의 환경 ID

**VarID** 변형 ID

#### **OrderNum**

SQL문의 재최적화에 사용된 변수의 서수

#### **SQLZType**

변수 유형

**CodPg** 변수 코드 페이지

**NulID** 값이 널(null) 종료되는지 여부를 표시하는 플래그

**Len** 변수값의 길이(바이트)

**Data** 변수에 사용되는 값

## **-reorgs** 매개변수

**-reorgs** 매개변수의 경우 다음 정보가 리턴됩니다.

테이블 **Reorg** 정보:

### **Address**

16진수 값

### **TabSpaceID**

테이블 스페이스 ID

### **TableID**

테이블 ID

### **PartID**

데이터 파티션 ID. 데이터 파티션마다 재구성 정보를 표시하는 하나의 행이 리턴됩니다.

### **MasterTbs**

파티션된 테이블의 경우, 파티션된 테이블이 속하는 논리 테이블 스페이스 ID입니다. 파티션되지 않은 테이블의 경우, 이 값은 TbspaceID에 해당됩니다.

### **MasterTab**

파티션된 테이블의 경우, 파티션된 테이블의 논리 테이블 ID입니다. 파티션되지 않은 테이블의 경우, 이 값은 TableID에 해당됩니다.

### **TableName**

테이블의 이름

**Type** 재구성 유형. 가능한 값은 다음과 같습니다.

- Online
- Offline

### **IndexID**

테이블을 재구성하기 위해 사용되는 인덱스의 ID

### **TempSpaceID**

테이블이 재구성되는 테이블 스페이스

테이블 **Reorg** 상태:

### **Address**

16진수 값

### **TableName**

테이블의 이름

**Start** 테이블 재구성이 시작된 시간

**End** 테이블 재구성이 종료된 시간

**PhaseStart**

테이블 재구성 단계의 시작 시간

**MaxPhase**

재구성 중에 발생할 최대 재구성 단계 수. 이 값은 오프라인 테이블 재구성에만 적용됩니다.

**Phase** 테이블 재구성의 단계. 이 값은 오프라인 테이블 재구성에만 적용됩니다. 가능한 값은 다음과 같습니다.

- 정렬
- Build
- Replace
- InxRecreat

**CurCount**

완료된 테이블 재구성 정도를 표시하는 진행 단위. 이 값이 표시하는 진행 정도는, 테이블 재구성에 필요한 총 작업량을 표시하는 MaxCount의 값에 상대적입니다.

**MaxCount**

테이블 재구성에 필요한 총 작업량을 표시하는 값. 이 값은 CurCount와 함께 사용하여 테이블 재구성의 진행을 판별할 수 있습니다.

**Status**

온라인 테이블 재구성의 상태. 이 값은 오프라인 테이블 재구성에 적용되지 않습니다. 가능한 값은 다음과 같습니다.

- Started
- Paused
- Stopped
- Done
- Truncat

**Completion**

테이블 재구성의 성공 표시기. 가능한 값은 다음과 같습니다.

- 0: 테이블 재구성이 완료되었습니다.
- -1: 테이블 재구성이 실패했습니다.

```
db2pd -db wsdw -reorgs file=reorg_pd.out
```

Table Reorg Information:

Address	TbspaceID	TableID	PartID	MasterTbs	MasterTab	TableName	Reclaim	0	0
Type	IndexID	TempSpaceID							
0x00002B214C24D5A8	2	4	n/a	n/a	n/a	T1			

Table Reorg Stats:

Address	TableName	Start	End			
PhaseStart	MaxPhase	Phase	CurCount	MaxCount	Status	Completion
0x00002B214C24D5A8	T1	10/21/2008 13:24:05	10/21/2008 13:24:05	10/21/2008 13:24:05	10/21/2008 13:24:05	3
Release	0	0	Done	0		

## -scansharing 매개변수

-scansharing 매개변수의 경우 표제에 특정한 다음 필드가 리턴됩니다.

### 개별적 공유 스캔

- 에이전트 ID
- 응용프로그램 ID
- ScanMode(prewrap 또는 postwrap)
- IsScanWrappable
- 스캔 속도
- 조절되는 데 소비된 시간
- 그룹 내의 페이지에서 스캔의 상대적 위치(블록 인덱스 스캔의 경우). 페이지에서 스캔의 절대 위치(테이블 및 범위 스캔의 경우)
- 예측된 속도 범주(SLOW 또는 FAST)
- 처리할 남아 있는 페이지(테이블 및 범위 스캔의 경우 정확함). 블록 인덱스 스캔의 경우에는 옵티마이저 예측이 대신 리턴됩니다.

db2pd -scansharing 명령의 샘플 출력을 참조하십시오.

### 공유 설정

- 테이블 스페이스 ID
- 테이블 ID
- 스캔 오브젝트(테이블 스캔 또는 블록 인덱스 ID의 경우 0)
- 그룹 수
- 공유 세트 풋프린트(페이지 수)
- 페이지 수 단위의 테이블 크기(파티션되지 않은 테이블에 대한 테이블 스캔 및 블록 인덱스 스캔과, 파티션된 테이블에 대한 범위 스캔의 경우. 파티션된 테이블에 대한 블록 인덱스 스캔의 경우 값은 unknown임)
- 급속 스캔 속도(FAST 스캔이 진행되는 속도)
- 느린 스캔 속도(SLOW 스캔이 진행되는 속도)

### 공유 그룹

- 그룹의 스캔 수
- 그룹 풋프린트(페이지 수)

## **-serviceclasses 매개변수**

**-serviceclasses** 매개변수의 경우 표제에 특정한 다음 필드가 리턴됩니다.

서비스 클래스 필드:

- Service Class Name: 서비스 클래스의 이름
- Service Class ID: 서비스 클래스의 시스템 생성 ID
- Service Class Type: 서비스 클래스의 유형: superclass 또는 subclass
- Service Class State (Effective and Catalog): 서비스 클래스의 상태: enabled 또는 disabled
- Effective Agent Priority 및 Catalog Agent Priority: SYSCAT.SERVICECLASSES에 기록된 우선순위에 맵핑되는 서비스 클래스에 대한 유효 에이전트 우선순위 설정
- Effective Prefetch Priority 및 Catalog Prefetch Priority: SYSCAT.SERVICECLASSES에 기록된 우선순위에 맵핑되는 서비스 클래스에 대한 유효 프리페치 우선순위 설정
- Effective Bufferpool Priority 및 Catalog Bufferpool Priority: SYSCAT.SERVICECLASSES에 기록된 우선순위에 맵핑되는 서비스 클래스에 대한 유효 버퍼 풀 우선순위 설정
- Effective Outbound Correlator 및 Catalog Outbound Correlator: SYSCAT.SERVICECLASSES에 기록된 상관자에 맵핑되는 서비스 클래스에 대한 유효 아웃바운드 상관자 설정
- Last Statistics Reset Time: 서비스 클래스에 대한 마지막 통계 재설정 시간소인.

서비스 수퍼 클래스 필드:

- Default Subclass ID: 디폴트 서브클래스의 서비스 클래스 ID
- Work Action Set ID: 서비스 수퍼 클래스와 연관되는 작업 조치 세트의 ID
- Collect Request Metrics: 서비스 클래스에 대한 COLLECT REQUEST METRICS 옵션의 설정
- Num Connections: 서비스 수퍼 클래스의 현재 코디네이터 및 리모트 연결 수
- Num Coordinator Connections: 서비스 수퍼 클래스의 현재 코디네이터 연결 수
- Coordinator Connections HWM: 마지막 통계 재설정 이후 코디네이터 연결에 대한 상위 워터 마크(water mark)
- Associated Workload Occurrences (WLO): 서비스 수퍼 클래스의 현재 워크로드 어커런스 목록

### 서비스 서브클래스 필드:

- Parent Superclass ID: 상위 슈퍼 클래스의 서비스 클래스 ID
- Collect Activity Opt: 서비스 서브클래스에 대한 COLLECT ACTIVITY DATA 옵션의 설정
- Collect Aggr Activity Opt: 서비스 서브클래스에 대한 COLLECT AGGREGATE ACTIVITY 옵션의 설정
- Collect Aggr Request Opt: 서비스 서브클래스에 대한 COLLECT AGGREGATE REQUEST 옵션의 설정
- Act Lifetime Histogram Template ID: 활동 라이프타임 막대 그래프 템플릿의 ID
- Act Queue Time Histogram Template ID: 활동 큐 시간 막대 그래프 템플릿의 ID
- Act Execute Time Histogram Template ID: 활동 실행 시간 막대 그래프 템플릿의 ID
- Act Estimated Cost Histogram Template ID: 활동 계산 비용 막대 그래프 템플릿의 ID
- Act Interarrival Time Histogram Template ID: 활동 도달 간 시간 막대 그래프 템플릿의 ID
- Request Execute Time Histogram Template ID: 요청 실행 시간 막대 그래프 템플릿의 ID
- Access Count: 서비스 서브클래스의 현재 활동 수
- Activities HWM: 마지막 통계 재설정 이후 활동에 대한 상위 워터 마크 (water mark)로, 해당 서브클래스를 통해 시스템에 입력된 활동과 REMAP ACTIVITY 임계값 조치에 의해 사용자가 해당 서브클래스에 재맵핑하는 활동 모두 계산됩니다.
- Activities Completed: 마지막 통계 재설정 이후에 완료된 총 활동 수. 활동이 완료되기 전에 REMAP ACTIVITY 조치로 다른 서브클래스에 활동을 다시 맵핑하는 경우, 이 활동은 완료되는 서브클래스 총계에만 계산됩니다.
- Activities Rejected: 마지막 통계 재설정 이후에 거부된 총 활동 수.
- Activities Aborted: 마지막 통계 재설정 이후에 중단된 총 활동 수. 활동이 중단되기 전에 REMAP ACTIVITY 조치로 다른 서브클래스에 활동을 다시 맵핑하는 경우, 이 활동은 중단되는 서브클래스 총계에만 계산됩니다.
- Associated Agents: 서비스 서브클래스에서 현재 작동 중인 에이전트의 목록



- Associated Non-agent threads: 서비스 서브클래스에서 현재 작동 중인 에이전트가 아닌 엔티티의 목록.

db2pd -serviceclasses 명령의 샘플 출력을 참조하십시오.

## **-static** 매개변수

**-static** 매개변수의 경우 다음 정보가 리턴됩니다.

정적 캐시:

### **Current Memory Used**

패키지 캐시에 사용되는 바이트 수.

### **Total Heap Size**

패키지 캐시에 대해 내부적으로 구성된 바이트 수.

### **Cache Overflow flag state**

패키지 캐시가 오버플로우 상태인지 여부를 표시하기 위한 플래그.

### **Number of References**

패키지 캐시에서 패키지에 대한 참조 수.

### **Number of Package Inserts**

패키지 캐시로의 패키지 삽입 수.

### **Number of Section Inserts**

패키지 캐시로의 정적 섹션 삽입 수.

패키지:

스키마

패키지의 규정자.

**PkgName**

패키지의 이름.

**Version**

패키지의 버전 ID.

**UniqueID**

패키지와 연관되는 일관성 토큰.

**NumSec**

로드된 섹션 수.

**UseCount**

캐시된 패키지의 사용 계수.

**NumRef**

캐시된 패키지가 참조된 횟수.

**Iso** 패키지의 분리 레벨.

**QOpt** 패키지의 쿼리 최적화.

**Blk** 패키지의 블로킹 인수.

**Lockname**

패키지의 잠금 이름.

**섹션:**

**스키마**

섹션이 속하는 패키지의 규정자.

**PkgName**

섹션이 속하는 패키지 이름.

**UniqueID**

섹션이 속하는 패키지와 연관되는 일관성 토큰.

**SecNo** 섹션 번호.

**NumRef**

캐시된 섹션이 참조된 횟수.

**UseCount**

캐시된 섹션의 사용 계수.

**StmtType**

캐시된 섹션에 대한 내부 명령문 유형 값.

**커서** 커서 이름(적용 가능한 경우).

**W-Hld** 커서가 WITH HOLD 커서인지 여부를 표시합니다.

**-statisticscache 매개변수**

**-statisticscache** 매개변수의 경우, 다음 정보가 리턴됩니다.

**Current Size**

통계 캐시에서 사용된 현재 바이트 수.

**Address**

통계 캐시에서 항목의 주소.

**스키마**

테이블의 스키마 규정자.

**이름** 테이블의 이름

**LastRefID**

테이블을 참조한 마지막 프로세스 ID.

**LastStatsTime**

테이블에 대한 최근 통계 컬렉션 시간.

**Sts** 항목의 상태. 가능한 값은 다음과 같습니다.

- V(유효함).
- I(유효하지 않음).

**-statisticscache** 명령 매개변수를 사용할 경우 리턴되는 정보에 대한 추가 세부사항은 [문제점 해결 및 데이터베이스 성능 조정의 『카탈로그 통계 테이블』](#)을 참조하십시오.

**-storagepaths** 매개변수

**-storagepaths** 매개변수의 경우 다음 정보가 리턴됩니다.

**Number of Storage Paths**

데이터베이스에 대해 정의된 자동 스토리지 경로 수.

**PathName**

데이터베이스에 대해 정의된 자동 스토리지 경로의 이름. 경로에 데이터베이스 파티션 표현식이 포함되는 경우, 확장 경로 다음에 괄호 안에 포함됩니다.

**PathID**

스토리지 경로 ID.

**PathState**

스토리지 경로의 현재 상태: NotInUse, InUse 또는 DropPending

db2pd -storagepaths 명령의 샘플 출력을 참조하십시오.

**-sysplex** 매개변수

**-sysplex** 매개변수의 경우, 다음 정보가 리턴됩니다.

**Alias** 데이터베이스 별명.

**Location Name**

데이터베이스 서버의 고유 이름.

**Count** 서버 목록에서 발견된 항목 수.

**IP Address**

서버의 IP 주소.

**Port** 서버에서 사용되는 IP 포트.

**Priority**

표준화된 워크로드 관리 프로그램(WLM)의 중요도

**Connections**

해당 서버에 대한 사용 중인 연결 수.

## Status

연결의 상태. 가능한 값은 다음과 같습니다.

- 0: Healthy.
- 1: Unhealthy. 서버가 목록에 있지만 연결을 설정할 수 없습니다. 연결 설정 시 이 항목은 현재 고려되지 않습니다.
- 2: Unhealthy. 서버가 이전에 사용 불가능했지만, 연결 설정 시 고려됩니다.

**PRDID** 마지막 연결 시 서버의 제품 ID.

## -tablespaces 매개변수

**-tablespaces** 매개변수의 경우, 출력은 네 개의 세그먼트로 구성됩니다.

테이블 스페이스 구성:

**Id** 테이블 스페이스 ID.

**Type** 테이블 스페이스의 유형. 가능한 값은 다음과 같습니다.

- SMS
- DMS

## Content

컨텐츠의 유형. 가능한 값은 다음과 같습니다.

- Regular
- Large
- SysTmp
- UsrTmp

## PageSz

테이블 스페이스에 사용되는 페이지 크기.

## ExtentSz

Extent 크기(페이지 수).

**Auto** 프리페치 크기가 AUTOMATIC으로 설정되는지 여부를 표시합니다. 가능한 값은 다음과 같습니다.

- YES
- NO

## Prefetch

각 범위 프리페치 요청에 대해 테이블 스페이스에서 읽은 페이지 수.

**BufID** 해당 테이블 스페이스가 맵핑되는 버퍼 풀의 ID.

## BufIDDisk

해당 테이블 스페이스가 다음 시작에서 맵핑되는 버퍼 풀의 ID.

**FSC** 파일 시스템 캐싱으로, CREATE TABLESPACE 또는 ALTER

TABLESPACE를 수행할 때 사용자가 버퍼 지정 입출력을 지정했는지 여부를 표시합니다. 가능한 값은 다음과 같습니다.

- YES
- NO

#### **NumCntrs**

테이블 스페이스가 소유하는 컨테이너 수.

#### **MaxStripe**

현재 테이블 스페이스에 정의된 최대 스트라이프 세트(DMS 테이블 스페이스에만 적용 가능).

#### **LastConsecPg**

마지막 연속 오브젝트 테이블 Extent.

이름 테이블 스페이스의 이름.

테이블 스페이스 통계:

**Id** 테이블 스페이스 ID.

#### **TotalPages**

DMS 테이블 스페이스의 경우, 테이블 스페이스 컨테이너 각각의 전체 크기 합 (컨테이너의 총 페이지 수 필드에 보고됨).

SMS 테이블 스페이스의 경우, 이 값은 테이블 스페이스가 소유하는 파일 시스템의 페이지 수를 반영합니다.

#### **UsablePgs**

DMS 테이블 스페이스의 경우, 테이블 스페이스 컨테이너 각각의 네트 크기 합 (컨테이너의 사용 가능한 페이지 수 필드에 보고됨).

SMS 테이블 스페이스의 경우, 이 값은 테이블 스페이스가 소유하는 파일 시스템의 페이지 수를 반영합니다.

#### **UsedPgs**

DMS 테이블 스페이스의 경우, 현재 테이블 스페이스에서 사용 중인 총 페이지 수.

SMS 테이블 스페이스의 경우, 이 값은 테이블 스페이스가 소유하는 파일 시스템의 페이지 수를 반영합니다.

#### **PndFreePgs**

사용할 수 없지만 현재 미해결 상태인 모든 트랜잭션이 커밋되는 경우 사용 가능하게 될 페이지 수.

#### **FreePgs**

DMS 테이블 스페이스의 경우, 테이블 스페이스에서 사용 가능한 페이지 수.

SMS 테이블 스페이스의 경우 이 값은 항상 0입니다.

**HWM** 테이블 스페이스에서 할당된 최상위 페이지.

상태

- 0x00000000 - NORMAL
- 0x00000001 - QUIESCED: SHARE
- 0x00000002 - QUIESCED: UPDATE
- 0x00000004 - QUIESCED: EXCLUSIVE
- 0x00000008 - LOAD PENDING
- 0x00000010 - DELETE PENDING
- 0x00000020 - BACKUP PENDING
- 0x00000040 - ROLLFORWARD IN PROGRESS
- 0x00000080 - ROLLFORWARD PENDING
- 0x00000100 - RESTORE PENDING
- 0x00000200 - DISABLE PENDING
- 0x00000400 - REORG IN PROGRESS
- 0x00000800 - BACKUP IN PROGRESS
- 0x00001000 - STORAGE MUST BE DEFINED
- 0x00002000 - RESTORE IN PROGRESS
- 0x00004000 - OFFLINE
- 0x00008000 - DROP PENDING
- 0x00010000 - WRITE SUSPENDED
- 0x00020000 - LOAD IN PROGRESS
- 0x00040000 - STORAGE MAY BE DEFINED
- 0x00080000 - STORAGE DEFINITION IS IN FINAL STATE
- 0x00100000 - STORAGE DEFINITION CHANGED PRIOR TO ROLLFORWARD
- 0x00200000 - DMS REBALANCER IS ACTIVE
- 0x00400000 - DELETION IN PROGRESS
- 0x00800000 - CREATION IN PROGRESS

**MinRecTime**

테이블 스페이스에 대한 최소 복구 시간.

**NQuiescers**

quiescer 수.

**PathsDropped**

자동 스토리지 테이블 스페이스의 경우, 하나 이상의 컨테이너가 삭제된 스토리지 경로에 있는지 여부를 지정합니다. 가능한 값은 다음과 같습니다.

- YES
- NO

테이블 스페이스 자동 크기 재조정 통계:

- Id** 테이블 스페이스 ID.
- AS** 테이블 스페이스가 자동 스토리지를 사용 중인지 여부를 표시합니다. 가능한 값은 다음과 같습니다.
- YES
  - NO
- AR** 테이블 스페이스의 크기가 자동으로 조정되는지 여부를 표시합니다. 가능한 값은 다음과 같습니다.
- YES
  - NO

**InitSize**

자동 스토리지 테이블 스페이스의 경우, 이 매개변수의 값은 테이블 스페이스의 초기 크기(바이트)입니다.

**IncSize**

자동으로 크기가 조정되는 테이블 스페이스의 경우, IIP 필드의 값이 No이면 이 매개변수의 값은 테이블 스페이스가 가득 차고 스페이스에 대한 요청이 작성될 때 테이블 스페이스가 자동으로 증가하는(데이터베이스 파티션당) 크기(바이트)입니다. IIP 필드의 값이 Yes이면 이 매개변수의 값은 백분율입니다.

- IIP** 자동으로 크기가 조정되는 테이블 스페이스의 경우, 이 매개변수의 값은 IncSize 필드의 증분값이 퍼센트인지 여부를 표시합니다. 가능한 값은 다음과 같습니다.
- YES
  - NO

**MaxSize**

자동으로 크기가 조정되는 테이블 스페이스의 경우, 이 매개변수의 값은 테이블 스페이스가 자동으로 증가하는(데이터베이스 파티션당) 최대 크기(바이트)를 지정합니다. NONE 값은 최대 크기가 없음을 표시합니다.

**LastResize**

마지막으로 성공한 자동 크기 조정 조작의 시간소인.

- LRF** 마지막 크기 조정 실패는 마지막 자동 크기 조정 조작의 성공 여부를 표시합니다. 가능한 값은 다음과 같습니다.
- YES
  - NO

**테이블 스페이스 컨테이너:**

**TspId** 컨테이너를 소유하는 테이블 스페이스의 ID.

**ContainNum**

테이블 스페이스에서 컨테이너에 지정된 번호.

**Type** 컨테이너의 유형. 가능한 값은 다음과 같습니다.

- Path
- Disk
- File
- Striped Disk
- Striped File

**TotalPgs**

컨테이너의 페이지 수.

**UsablePgs**

컨테이너에서 사용 가능한 페이지 수.

**StripeSet**

컨테이너가 있는 스트라이프 세트(DMS 테이블 스페이스에만 적용 가능).

**Container**

컨테이너의 이름.

**PathID**

자동 스토리지 테이블 스페이스의 경우, 컨테이너가 있는 스토리지 경로의 ID.

db2pd -tablespaces 명령의 샘플 출력을 참조하십시오.

**-tcbstats 매개변수**

-tcbstats 매개변수의 경우 다음 정보가 리턴됩니다.

**TCB 테이블 정보:**

**TbpaceID**

테이블 스페이스 ID

**TableID**

테이블 ID

**PartID**

파티션된 테이블의 경우 데이터 파티션 ID. 파티션되지 않은 테이블의 경우 'n/a'를 표시합니다.

**MasterTbs**

파티션된 테이블의 경우, 파티션된 테이블이 속하는 논리 테이블 스페이스 ID입니다. 파티션되지 않은 테이블의 경우, 이 값은 TbpaceID에 해당됩니다.

**MasterTab**

파티션된 테이블의 경우, 파티션된 테이블의 논리 테이블 ID입니다. 파티션되지 않은 테이블의 경우, 이 값은 TableID에 해당됩니다.



**TableName**

테이블의 이름.

**SchemaNm**

테이블 이름을 규정하는 스키마

**ObjClass**

오브젝트 클래스. 가능한 값은 다음과 같습니다.

- Perm(영구).
- Temp(임시).

**DataSize**

데이터 오브젝트의 페이지 수.

**LfSize**

Long 필드 오브젝트의 페이지 수.

**LobSize**

대형 오브젝트(LOB)의 페이지 수.

**XMLSize**

XML 오브젝트의 페이지 수.

**TCB 테이블 통계:****TableName**

테이블의 이름

**SchemaNm**

테이블 이름을 규정하는 스키마

**Scans** 테이블에 대해 수행된 스캔 수.

**UDI** 백그라운드 통계 컬렉션 프로세스를 통하거나 수동으로 RUNSTATS 명령을 사용하여 테이블 통계가 갱신된 마지막 시간 이후에 테이블에 대해 수행된 갱신, 삭제 및 삽입 조작 수.

**RTSUDI**

실시간 통계 수집, 백그라운드 통계 컬렉션 프로세스 또는 수동 RUNSTATS에 의해 테이블 통계가 갱신된 마지막 시간 이후에 테이블에 대해 수행된 갱신, 삭제 및 삽입 조작 수.

**PgReorgs**

재구성이 수행된 페이지 수.

**NoChgUpdts**

테이블에서 컬럼을 변경하지 않은 갱신 수.

**Reads** 모니터링에 대해 테이블 스위치가 켜진 테이블에서 읽은 행 수.

**FscrUpdates**

여유 공간 제어 레코드에 대한 갱신 수.

**Inserts**

테이블에 대해 수행된 삽입 조작 수.

**Updates**

테이블에 대해 수행된 갱신 조작 수.

**Deletes**

테이블에 대해 수행된 삭제 조작 수.

**0vF1Reads**

모니터링에 대해 테이블 스위치가 켜진 테이블에서 읽은 오버플로우 수.

**0vF1Crtes**

작성된 새 오버플로우 수.

**RowsComp**

압축된 총 행 수.

**RowsUNcomp**

압축되지 않은 총 행 수.

**CCLogReads**

테이블에서 검색되어 현재 커밋된 행 버전의 횟수.

**StoredBytes**

이 컬럼은 **db2pd -temptable** 출력의 “저장된 총 임시 바이트 수”에 해당됩니다.

**BytesSaved**

이 컬럼은 **db2pd -temptable** 출력의 “저장된 총 바이트 수”에 해당됩니다.

참고 다음 데이터는 **-tcbstats** 매개변수와 함께 **-all** 또는 **-index** 옵션을 지정한 경우에만 표시됩니다.

**TCB 인덱스 정보:****InxTbSpace**

인덱스가 있는 테이블 스페이스

**ObjectID**

인덱스의 오브젝트 ID

**PartID**

파티션된 테이블의 경우 데이터 파티션 ID. 파티션되지 않은 테이블의 경우 N/A가 표시됩니다.

**TbSpaceID**

테이블 스페이스 ID

**TableID**

테이블 ID

**MasterTbs**

파티션된 테이블의 경우, 파티션된 테이블이 속하는 논리 테이블 스페이스 ID입니다. 파티션되지 않은 테이블의 경우, 이 값은 TbSpaceID에 해당됩니다.

**MasterTab**

파티션된 테이블의 경우, 파티션된 테이블의 논리 테이블 ID입니다. 파티션되지 않은 테이블의 경우, 이 값은 TableID에 해당됩니다.

**TableName**

테이블의 이름

**SchemaNm**

테이블 이름을 규정하는 스키마

**IID** 인덱스 ID**IndexObjSize**

인덱스 오브젝트의 페이지 수

**TCB 인덱스 통계:****TableName**

테이블의 이름

**IID** 인덱스 ID**PartID**

파티션된 테이블의 경우 데이터 파티션 ID. 파티션되지 않은 테이블의 경우 N/A가 표시됩니다.

**EmpPgDel**

삭제된 비어 있는 리프 노드 수

**RootSplits**

인덱스 트리 용량이 증가하도록 한 키 삽입 또는 갱신 조작 수

**BndrySplits**

최하위 또는 최상위 키로의 삽입 조작을 야기하는 경계 리프 분할 수

**PseuEmptPg**

유사(pseudo) 빈 상태가 되는 것으로 표시되는 리프 노드 수

**EmpPgMkdUsd**

재사용된 유사(pseudo) 빈 상태의 페이지 수

**Scans** 인덱스에 대한 스캔 수

**IxOnlyScns**

인덱스만 스캔 수(인덱스만에 대한 액세스에서 충족된 스캔의 결과)

**KeyUpdates**

키에 대한 갱신 수

**InclUpdats**

포함된 컬럼 갱신 수

**NonBndSpts**

경계 없는 리프 분할 수

**PgAllocs**

할당된 페이지 수

**Merges**

인덱스 페이지에 대해 수행된 병합 수

**PseuDels**

유사(pseudo) 삭제된 것으로 표시되는 키 수

**DelClean**

삭제된 유사(pseudo) 삭제 상태의 키 수

**IntNodSp1**

중간 레벨 분할 수

## **-temptable** 매개변수

모든 임시 테이블 사이에 누적 압축 비율을 계산하기 위해 다음 공식을 사용할 수 있습니다.

$$\% \text{ 압축} = \frac{\text{저장된 총 바이트 수}}{\text{저장된 총 바이트 수} + \text{저장된 총 임시 바이트 수}}$$

주:

- Eligible 용어는 압축 기준을 충족하는 임시 테이블을 표시합니다.
- Compressed 용어는 마지막으로 압축되도록 삽입된 충분한 데이터를 가지고 있는 임시 테이블을 표시합니다.

```
hotel26:/home/billyp> db2pd -db billdb -temptable
db_state1 db_state2 db_state3 db_state4 use_count dbname dbpath
-----
000000001 000000000 000000000 000000000 000000009 BILL /home/billyp/billyp/NODE0000/SQL00011/
```

```
System Temp Table Stats:
Number of Temp Tables : 0
Comp Eligible Temps : 0
Compressed Temps : 0
Total Temp Bytes : 0
Total Bytes Saved : 0
Total Compressed Rows : 0
```

Total Temp Table Rows: : 0

User Temp Table Stats:

Number of Temp Tables : 0  
Comp Eligible Temps : 0  
Compressed Temps : 0  
Total Stored Temp Bytes : 0  
Total Bytes Saved : 0  
Total Compressed Rows : 0  
Total Temp Table Rows : 0

숨겨진 reset 옵션을 사용하여 모든 카운터를 0으로 재설정할 수 있습니다.

```
hotel26:/home/billyp> db2pd -db bill -temptable -reset  
카운터가 0으로 재설정됩니다.
```

db2pd -temptable 명령의 샘플 출력을 참조하십시오.

## **-thresholds** 매개변수

**-thresholds** 매개변수의 경우 다음 정보가 리턴됩니다.

- Threshold Name: 임계값 이름
- Threshold ID: 임계값 ID
- Domain: 임계값 도메인
- Domain ID: 임계값 도메인 ID
- Predicate ID: 임계값 술어 ID
- Maximum Value: 임계값 최대값
- Enforcement: 임계값 시행 범위
- Queuing: 임계값은 큐잉 임계값임
- Queue Size: 임계값 큐 크기 설정
- Collect Flags: 임계값에 대한 COLLECT ACTIVITY DATA 옵션의 설정
- Partition Flags: COLLECT ACTIVITY 옵션 설정이 적용되는 파티션
- Execute Flags: 임계값 조치 설정
- Enabled: 임계값의 상태(enabled 또는 disabled)
- Check Interval (seconds): 임계값 조건 점검에 대한 빈도 설정
- Remap Target Serv. Subclass: 임계값 재맵핑 조치에 대한 목표 서비스 서브클래스 설정
- Log Violation Evmon Record: THRESHOLD VIOLATIONS 이벤트 모니터 로그 설정

임계값이 큐잉 임계값인 경우, 큐 섹션은 다음 정보도 표시합니다.

- Queue information for threshold: 임계값 이름
- Max Concurrency: 최대 동시 설정

- Concurrency: 실제 동시 값
- Max Queue Size: 최대 임계값 큐 크기 설정
- Agents Currently Queued: 카탈로그 노드에서, 임계값 큐에서 대기 중인 모든 에이전트의 목록(에이전트가 큐에 대기되는 경우에만 표시됨).

db2pd -thresholds 명령의 샘플 출력을 참조하십시오.

## **-transactions** 매개변수

**-transactions** 매개변수의 경우, 다음 정보가 리턴됩니다.

### **AppHandl**

트랜잭션의 응용프로그램 핸들.

### **TranHdl**

트랜잭션의 트랜잭션 핸들.

**Locks** 트랜잭션이 보유하는 잠금 수.

상태 트랜잭션 상태.

**Tflag** 트랜잭션 플래그. 가능한 값은 다음과 같습니다.

- 0x00000002. 이 값은 2단계 커밋 응용프로그램의 코디네이터 노드에만 기록되며, 이는 모든 종속 노드가 "커밋 준비" 요청을 보냈음을 표시합니다.
- 0x00000020. 트랜잭션은 캡처 소스 테이블(데이터 복제에만 사용되는)을 변경해야 합니다.
- 0x00000040. 응급 복구에서 트랜잭션이 준비 상태에 놓일 것으로 간주합니다.
- 0x00010000. 이 값은 파티션된 데이터베이스 환경의 코디네이터 파티션에만 기록되며, 이는 코디네이터 파티션이 2단계 커밋 트랜잭션의 모든 종속 파티션으로부터 커밋 요청을 수신하지 않았음을 표시합니다.
- 0x00040000. 트랜잭션의 롤백이 보류 상태입니다.
- 0x01000000. 트랜잭션이, 코디네이터 파티션이 아닌 데이터베이스 파티션 서버에서 갱신을 발생했습니다.
- 0x04000000. 느슨하게 결합된 XA 트랜잭션이 지원됩니다.
- 0x08000000. 여러 개의 분기가 해당 트랜잭션과 연관되고 느슨하게 결합된 XA 프로토콜을 사용 중입니다.
- 0x10000000. 데이터 정의 언어(DDL)문이 발행되었습니다. 이는 트랜잭션에 참여하는 분기에서 느슨하게 결합된 XA 프로토콜을 사용할 수 없음을 표시합니다.

### **Tflag2**

트랜잭션 플래그 2. 가능한 값은 다음과 같습니다.

- 0x00000004. 트랜잭션이 **num\_log\_span** 데이터베이스 구성 매개변수에 지정된 한계를 초과했습니다.
- 0x00000008. DB2 유틸리티 실행으로 트랜잭션이 발생했습니다.
- 0x00000020. 트랜잭션은 우선순위가 더 높은 응용프로그램에 해당 잠금을 양도합니다(이 값은 일반적으로 DB2 데이터베이스 시스템이 자동으로 자체 조정 및 자체 관리를 위해 시작하는 작업에 대해 발생함).
- 0x00000040. 트랜잭션은 우선순위가 더 높은 응용프로그램에 해당되는 행 레벨 잠금을 양도하지 않습니다(이 값은 일반적으로 DB2 데이터베이스 시스템이 자동으로 자체 조정 및 자체 관리를 위해 시작하는 작업에 대해 발생함).

#### **Firstlsn**

트랜잭션의 첫 번째 LSN.

#### **Lastlsn**

트랜잭션의 마지막 LSN.

#### **LogSpace**

트랜잭션에 대해 예약된 로그 스페이스의 양.

#### **SpaceReserved**

사용된 스페이스 및 모든 보상 레코드를 포함하여, 트랜잭션에 대해 예약된 총 로그 스페이스.

**TID** 트랜잭션 ID.

#### **AxRegCnt**

전역 트랜잭션에 대해 등록된 응용프로그램 수. 로컬 트랜잭션의 경우 값은 1입니다.

**GXID** 전역 트랜잭션 ID. 로컬 트랜잭션의 경우 값은 0입니다.

#### **ClientUserID**

트랜잭션의 클라이언트 사용자 ID로 tpmon\_client\_userid(TP 모니터 클라이언트 사용자 ID 모니터 요소)와 동일.

#### **ClientWrkstnName**

트랜잭션의 클라이언트 워크스테이션 이름으로 tpmon\_client\_wkstn(TP 모니터 클라이언트 워크스테이션 이름 모니터 요소)과 동일.

#### **ClientAppName**

트랜잭션을 구동하는 클라이언트 응용프로그램 이름으로 tpmon\_client\_app(TP 모니터 클라이언트 응용프로그램 모니터 요소)와 동일.

#### **ClientAcctng**

트랜잭션을 구동하는 클라이언트의 어카운팅 문자열로 tpmon\_acc\_str(TP 모니터 클라이언트 어카운팅 문자열 모니터 요소)과 동일.

## **-wlocks** 매개변수

**-wlocks** 매개변수의 경우, 다음 정보가 리턴됩니다.

### **AppHandl**

노드 및 인덱스를 포함하는 응용프로그램 핸들.

### **TranHdl**

잠금을 요청하는 트랜잭션 핸들.

### **LockName**

잠금의 이름.

**Type** 잠금의 유형.

**Mode** 잠금 모드. 가능한 값은 다음과 같습니다.

- IS
- IX
- S
- SIX
- X
- IN
- Z
- U
- NS
- NW

**Conv** 잠금 대기 종료 후 잠금이 변환될 잠금 모드.

**Sts** 잠금 상태. 가능한 값은 다음과 같습니다.

- G(권한 부여됨)
- C(변환 중)
- W(대기 중)

### **CoorEDU**

응용프로그램에 대한 코디네이터 에이전트의 EDU ID.

### **AppName**

응용프로그램의 이름.

### **AuthID**

권한 부여 ID.

**AppID** 응용프로그램 ID. 이 값은 **appl\_id** 모니터 요소 데이터와 같습니다.

db2pd -wlock 명령의 샘플 출력을 참조하십시오.



## **-workactionsets 매개변수**

**-workactionsets** 매개변수의 경우, 다음 정보가 리턴됩니다.

- 주소
- 작업 조치 세트 ID
- 작업 조치 세트 이름
- 연관된 작업 클래스 세트 ID
- 작업 조치 세트가 연관된 오브젝트의 유형(데이터베이스 또는 서비스 클래스)
- 작업 조치 세트가 연관되는 오브젝트(서비스 클래스 또는 데이터베이스)의 ID
- 작업 조치 세트 내의 모든 작업 조치:
  - 주소
  - 조치 ID
  - 조치 유형
  - 참조 오브젝트 ID(조치 유형에 따라 임계값 ID 또는 서비스 클래스 ID 또는 널 (NULL))

## **-workclasssets 매개변수**

**-workclasssets** 매개변수의 경우, 다음 정보가 리턴됩니다.

- 주소
- 작업 클래스 ID
- 참조 카운터(작업 클래스 세트를 참조하는 여러 작업 조치 세트 수)
- 작업 클래스 세트 내의 모든 작업 클래스(해당 평가 순서대로 표시됨)
  - 주소
  - 클래스 ID
  - 클래스 이름
  - 클래스 유형
  - 스키마 이름
  - from 값
  - to 값
  - 범위 단위

## **-workloads 매개변수**

**-workloads** 매개변수의 경우 표제에 특정한 다음 필드가 리턴됩니다.

워크로드 정의

- 워크로드 ID 및 이름

- 워크로드 어커런스에 대한 데이터베이스 액세스 권한
- 동시 워크로드 어커런스 수
- 워크로드 임계값
- 연관된 서비스 클래스
- 통계 콜렉션 설정
- 막대 그래프 템플릿 ID

#### 사용 특권 보유자

- 워크로드 ID
- 보유자의 유형
- 권한 부여 ID

#### 로컬 파티션 워크로드 통계

- 워크로드 ID 및 이름
- 워크로드 어커런스 통계
- 마지막 통계 재설정 이후의 시간
- 활동 통계

db2pd -workloads 명령의 샘플 출력을 참조하십시오.

#### 샘플 출력

##### -addnode

다음은 db2pd -addnode 명령의 출력 샘플입니다.

```
-----
Summary of add partition processing done for partition[50]
-----
00:Creating database partitions                : True
01:Database partitions are created             : True
08:Collecting storage information              : True
09:Storage information is collected            : True
11:FCM Send & Receive daemons are blocked    : True
12:FCM Send & Receive daemons are reactivated : True
13:db2start processing is complete            : True

Conflicting states or activities for add partition for partition[50]

-----
[14] Messages found for partition [50]
-----
[Fri Oct 24 16:16:27 2008]:Addnode agent:Got automatic storage details
[Fri Oct 24 16:16:28 2008]:Addnode agent:Skeleton database is created
[Fri Oct 24 16:16:28 2008]:Addnode agent:Scanning for db alias=[PE      ] name=[PE      ]
[Fri Oct 24 16:16:28 2008]:Addnode agent:Found db alias=[PE      ] name=[PE      ]
[Fri Oct 24 16:16:28 2008]:Addnode agent:Instance directory already exists
[Fri Oct 24 16:16:28 2008]:Addnode agent:Node directory already exists
[Fri Oct 24 16:16:28 2008]:Addnode agent:Node directory is created
[Fri Oct 24 16:16:29 2008]:Addnode agent:Getting automatic storage details
[Fri Oct 24 16:16:29 2008]:Addnode agent:Got automatic storage details
[Fri Oct 24 16:16:30 2008]:Addnode agent:Skeleton database is created
```

```
[Fri Oct 24 16:16:30 2008]:Addnode agent:Database activation is not required
[Fri Oct 24 16:16:30 2008]:Addnode agent:Database activation is complete
[Fri Oct 24 16:16:30 2008]:Addnode agent:Online mode processing is complete
[Fri Oct 24 16:16:30 2008]:db2start is complete
```

### oldviewapps

데이터베이스 파티션 서버 추가 조작이 발생하기 전에 응용프로그램이 인스턴스의 데이터베이스 파티션 서버(노드) 수를 확인하는 정보를 리턴합니다.

다음은 db2pd -addnode oldviewsapps 명령의 출력 샘플입니다.

```
-----
Summary of add partition processing done for partition[0]
-----
```

```
-----
Conflicting states or activities for add partition for partition[0]
-----
```

```
-----
Applications with old view of instance for partition [0]
-----
```

```
App.Handle(00000000,00000072) view has [3] nodes, instance has [4] nodes
App.Handle(00000000,00000065) view has [3] nodes, instance has [4] nodes
App.Handle(00000000,00000071) view has [3] nodes, instance has [4] nodes
App.Handle(00000000,00000005) view has [3] nodes, instance has [4] nodes
App.Handle(00000000,00000051) view has [3] nodes, instance has [4] nodes
App.Handle(00000000,00000070) view has [3] nodes, instance has [4] nodes
App.Handle(00000000,00000069) view has [3] nodes, instance has [4] nodes
App.Handle(00000000,00000068) view has [3] nodes, instance has [4] nodes
App.Handle(00000001,00000058) view has [3] nodes, instance has [4] nodes
App.Handle(00000000,00000067) view has [3] nodes, instance has [4] nodes
App.Handle(00000000,00000073) view has [3] nodes, instance has [4] nodes
```

**detail** oldviewapps 옵션과 함께 사용한 경우, 최근에 추가된 데이터베이스 파티션 서버가 없는 인스턴스의 뷰가 있는 응용프로그램의 정보를 리턴합니다. **oldviewapps** 옵션과 함께 사용한 경우, 최근에 추가된 데이터베이스 파티션 서버가 없는 인스턴스의 뷰가 있는 응용프로그램에 대한 정보도 리턴합니다.

다음은 db2pd -addnode detail 명령의 출력 샘플입니다.

```
-----
Add partition processing with detail for partition[50]
-----
```

```
00:Creating database partitions : True
01:Database partitions are created : True
02:Dropping database entries : False
03:Dropping db entries are completed : False
04:Activating databases explicitly : False
05:Database explicit activation is completed : False
06:Updating database configuration : False
07:Database configuration is updated : False
08:Collecting storage information : True
09:Storage information is collected : True
10:Add partition operation is complete : False
11:FCM Send & Receive daemons are blocked : True
12:FCM Send & Receive daemons are reactivated : True
13:db2start processing is complete : True
```

```
-----
Conflicting states or activities for add partition for partition[50]
-----
```

```
restricted :False
```

```

db2start          :False
db2stop           :False
instance quiesced :False
database quiesced :False
quiesce instance :False
unquiesce instance :False
quiesce db        :False
unquiesce db      :False
activate db       :False
deactivate db     :False
exclusive use of db :False
create db         :False
drop db          :False
create tablespace :False
alter tablespace  :False
drop tablespace   :False
add partition     :False
backup database   :False
restore database  :False
snapshot restore  :False

```

[14] Messages found for partition [50]

```

-----
[Fri Oct 24 16:16:27 2008]:Addnode agent:Got automatic storage details
[Fri Oct 24 16:16:28 2008]:Addnode agent:Skeleton database is created
[Fri Oct 24 16:16:28 2008]:Addnode agent:Scanning for db alias=[PE      ] name=[PE      ]
[Fri Oct 24 16:16:28 2008]:Addnode agent:Found db alias=[PE      ] name=[PE      ]
[Fri Oct 24 16:16:28 2008]:Addnode agent:Instance directory already exists
[Fri Oct 24 16:16:28 2008]:Addnode agent:Node directory already exists
[Fri Oct 24 16:16:28 2008]:Addnode agent:Node directory is created
[Fri Oct 24 16:16:29 2008]:Addnode agent:Getting automatic storage details
[Fri Oct 24 16:16:29 2008]:Addnode agent:Got automatic storage details
[Fri Oct 24 16:16:30 2008]:Addnode agent:Skeleton database is created
[Fri Oct 24 16:16:30 2008]:Addnode agent:Database activation is not required
[Fri Oct 24 16:16:30 2008]:Addnode agent:Database activation is complete
[Fri Oct 24 16:16:30 2008]:Addnode agent:Online mode processing is complete
[Fri Oct 24 16:16:30 2008]:db2start is complete

```

Total [00] Conflicting application handles for partition [50]

-----

다음 예와 같이 충돌하는 조작이 표시됩니다.

Total [01] Conflicting application handles for partition [20]

-----

Agents for app\_handle 00000000 00000052 : Activity occurrence:[1] time(s) ActivityName:[exclusive use of db]

다음은 db2pd -addnode oldviewapps detail 명령의 출력 샘플입니다.

-----

Add partition processing with detail for partition[0]

```

-----
00:Creating database partitions          : False
01:Database partitions are created       : False
02:Dropping database entries             : False
03:Dropping db entries are completed     : False
04:Activating databases explicitly       : False
05:Database explicit activation is completed : False
06:Updating database configuration       : False
07:Database configuration is updated     : False
08:Collecting storage information        : False
09:Storage information is collected      : False
10:Add partition operation is complete   : False
11:FCM Send & Receive daemons are blocked : False
12:FCM Send & Receive daemons are reactivated : False
13:db2start processing is complete      : False

```

Conflicting states or activities for add partition for partition[0]

```
-----  
restricted                :False  
db2start                  :False  
db2stop                   :False  
instance quiesced         :False  
database quiesced         :False  
quiesce instance         :False  
unquiesce instance       :False  
quiesce db                :False  
unquiesce db             :False  
activate db               :False  
deactivate db            :False  
exclusive use of db      :False  
create db                 :False  
drop db                   :False  
create tablespace        :False  
alter tablespace         :False  
drop tablespace          :False  
add partition             :False  
backup database          :False  
restore database         :False  
snapshot restore         :False  
create/alter nodegroup   :False  
drop nodegroup           :False  
add storage               :False  
redistribute              :False
```

Total [00] Conflicting application handles for partition [0]

-----  
Applications with old view of instance for partition [0]

```
-----  
App.Handle(00000000,00000072) view has [3] nodes, instance has[4] nodes  
[Viewnodes:0:1:2:]  
App.Handle(00000000,00000065) view has [3] nodes, instance has[4] nodes  
[Viewnodes:0:1:2:]  
App.Handle(00000000,00000071) view has [3] nodes, instance has[4] nodes  
[Viewnodes:0:1:2:]  
App.Handle(00000000,00000005) view has [3] nodes, instance has[4] nodes  
[Viewnodes:0:1:2:]  
App.Handle(00000000,00000051) view has [3] nodes, instance has[4] nodes  
[Viewnodes:0:1:2:]  
App.Handle(00000000,00000070) view has [3] nodes, instance has[4] nodes  
[Viewnodes:0:1:2:]  
App.Handle(00000000,00000069) view has [3] nodes, instance has[4] nodes  
[Viewnodes:0:1:2:]  
App.Handle(00000000,00000068) view has [3] nodes, instance has[4] nodes  
[Viewnodes:0:1:2:]  
App.Handle(00000001,00000058) view has [3] nodes, instance has[4] nodes  
[Viewnodes:0:1:2:]  
App.Handle(00000000,00000067) view has [3] nodes, instance has[4] nodes  
[Viewnodes:0:1:2:]  
App.Handle(00000000,00000073) view has [3] nodes, instance has[4] nodes  
[Viewnodes:0:1:2:]
```

### -apinfo

다음은 db2pd -apinfo 명령의 출력 샘플입니다.

```
db2pd -apinfo 12 -db mydb10
```

```
Database Partition 0 -- Database MYDB10 -- Active -- Up 0 days 00:03:28
```

Application:

```
Address : 0x0780000000D76EE0  
AppHandl [nod-index] : 12 [000-00012]  
Application PID : 1384708
```

```

Application Node Name : boson
IP Address: n/a
Connection Start Time : (1195265036)Fri Nov 16 21:03:56 2007
Client User ID : venus
System Auth ID : VENUS
Coordinator EDU ID : 1801
Coordinator Partition : 0
Number of Agents : 1
Locks timeout value : 4294967294 seconds
Locks Escalation : No
Workload ID : 1
Workload Occurrence ID : 1
Trusted Context : n/a
Connection Trust Type : non trusted
Role Inherited : n/a
Application Status : Lock-wait
Application Name : db2bp
Application ID : *LOCAL.venus.071117020356
ClientUserID : n/a
ClientWrkstnName : n/a
ClientApplName : n/a
ClientAccntng : n/a

```

```

List of active statements :
*UOW-ID : 8
Activity ID : 2
Package Schema : NULLID
Package Name : SQLC2G13
Package Version :
Section Number : 201
SQL Type : Dynamic
Isolation : CS
Statement Type : DML, Select (blockable)
Statement : select * from t2

```

```

List of inactive statements of current UOW :
UOW-ID : 8
Activity ID : 1
Package Schema : NULLID
Package Name : SQLC2G13
Package Version :
Section Number : 203
SQL Type : Dynamic
Isolation : CS
Statement Type : DML, Insert/Update/Delete
Statement : insert into t1 values 1

```

### -catalogcache

다음은 db2pd -catalogcache 명령의 SYSTABLES 출력 샘플입니다.

Database Partition 0 -- Database SAMPLE -- Active -- Up 0 days 00:05:34

```

Catalog Cache:
Configured Size      1064960
Current Size         78272
Maximum Size         4294901760
High Water Mark      131072

```

```

SYSTABLES:
Address      Schema  Name      Type  TableID  TbspaceID  LastRefID  CatalogCacheLoadingLock  CatalogCacheUsageLock  Sts
0x07800000232FF820  SYSIBM  SYSTABLES  T     5         0          19288214  0001000007800000232FF82043  0000000500001804232FF82043  V
0x07800000232FD360  SYSCAT  TABLES    V     0         0          19288214  0001000007800000232FD36043  000000050000CC907232FD36043  V
0x07800000232FFB60  KEON014  EMPLOYEE   0     0         0          19288214  0001000007800000232FFB6043  000000050013AE07232FFB6043  I
0x07800000232FC500  SYSTOOLS  POLICY     0     0         0          19288214  0001000007800000232FC50043  0000000000000000000000000000  I
0x07800000232FCF40  KEON014  DEPT       T     4         2          19288214  0001000007800000232FCF4043  000000050013AE06232FCF0343  V
0x07800000238FCF40  KEON014  DEPT       T     4         2          19288214  0001000007800000238FCF4043  000000050013AE06238FCF0143  S
0x07800000234433A0  KEON014  SALARY     0     0         0          19288214  0001000007800000234433A043  000000050013AF00234433A043  I

```

-edus 다음은 db2pd -edus 명령의 출력 샘플입니다.

Database Partition 0 -- Active -- Up 0 days 01:14:05

List of all EDUs for database partition 0

db2sysc PID: 18485  
db2wdog PID: 18483  
db2acd PID: 18504

EDU ID	TID	Kernel TID	EDU Name	USR	SYS
24	47155322546496	12108	db2pfchr (TESTDB)	0.010000	0.000000
23	47155326740800	12107	db2pclnr (TESTDB)	0.000000	0.000000
22	47155330935104	12106	db2pclnr (TESTDB)	0.000000	0.000000
21	47155335129408	12105	db2pclnr (TESTDB)	0.000000	0.000000
20	47155339323712	12104	db2dlock (TESTDB)	0.000000	0.000000
19	47155343518016	12103	db2lfr (TESTDB)	0.000000	0.000000
18	47155347712320	12102	db2loggw (TESTDB)	0.000000	0.000000
17	47155351906624	12101	db2loggr (TESTDB)	0.080000	0.000000
16	47155356100928	27704	db2agent (TESTDB) (suspended)	0.930000	0.140000
15	47155360295232	18502	db2resync	0.080000	0.000000
14	47155364489536	18500	db2ipccm	0.030000	0.000000
13	47155368683840	18499	db2licc	0.000000	0.000000
12	47155372878144	18498	db2thcln	0.000000	0.000000
11	47155377072448	18497	db2alarm	0.000000	0.000000
1	47155117025600	18493	db2sysc	3.340000	0.070000

**-pages**

다음은 summary 매개변수를 지정하지 않은 db2pd -pages 명령의 출력 샘플입니다.

venus@baryon:/home/venus =>db2pd -pages -db pdtest

Database Partition 0 -- Database PDTEST -- Active -- Up 0 days 00:01:28

Bufferpool Pages:  
First Active Pool ID 1  
Max Bufferpool ID 1  
Max Bufferpool ID on Disk 1  
Num Bufferpools 5

Pages for all bufferpools:

Address	BPID	TbspaceID	TbspacePgNum	ObjID	ObjPgNum	ObjClass	ObjType	Dirty	Prefetched
0x0000002AC22ABAC0	1	0	92	10	0	EMP	Data	N	N
0x0000002AC22ABB80	1	0	2503	10	11	Perm	Index	N	N
0x0000002AC22ABC40	1	0	2501	10	9	Perm	Index	Y	N
0x0000002AC22ABD00	1	0	2494	10	2	Perm	Index	N	N
0x0000002AC22ABDC0	1	0	3437	5	17	Perm	Data	N	N
0x0000002AC22ABE80	1	0	2504	10	12	Perm	Index	Y	N
0x0000002AC22ABF40	1	0	2505	10	13	Perm	Index	N	N
0x0000002AC22AC000	1	0	2506	10	14	Perm	Index	N	N
0x0000002AC22AC0C0	1	0	28	5	0	EMP	LOB	N	N
0x0000002AC22AC180	1	0	2509	10	17	Perm	Index	N	N
0x0000002AC22AC240	1	0	2495	10	3	Perm	Index	Y	N
0x0000002AC22AC300	1	0	2498	10	6	Perm	Index	Y	N
0x0000002AC22AC3C0	1	2	128	4	0	Perm	Data	Y	N
0x0000002AC22AC480	1	0	2499	10	7	Perm	Index	N	N
0x0000002AC22AC540	1	0	99	10	3	Perm	Data	Y	N
0x0000002AC22AC600	1	0	96	10	0	Perm	Data	Y	N
0x0000002AC22AC6C0	1	0	110	5	2	Perm	Index	N	N
0x0000002AC22AC780	1	0	2500	10	8	Perm	Index	N	N
0x0000002AC22AC840	1	0	2740	5	16	Perm	Index	N	N
0x0000002AC22AC900	1	0	2507	10	15	Perm	Index	Y	N

Total number of pages: 20

Summary info for all bufferpools:

BPID	TbspaceID	ObjID	Total	Dirty	Permanent	Temporary	Data	Index	LongField	XMLData	SMP	LOB	LOBA	BMP
1	0	5	4	0	3	0	1	2	0	0	0	1	0	0
1	0	10	15	7	14	0	3	12	0	0	0	0	0	0
1	2	4	1	1	1	0	1	0	0	0	0	0	0	0

Total number of pages: 20

다음은 summary 매개변수를 지정한 db2pd -pages 명령의 출력 샘플입니다.

venus@baryon:/home/venus =>db2pd -pages summary -db pdtest

Database Partition 0 -- Database PDTEST -- Active -- Up 0 days 00:02:07

Bufferpool Pages:  
First Active Pool ID 1  
Max Bufferpool ID 1  
Max Bufferpool ID on Disk 1  
Num Bufferpools 5

Total number of pages: 20

Summary info for all bufferpools:

BPID	TbSpaceID	ObjID	Total	Dirty	Permanent	Temporary	Data	Index	LongField	XMLData	SMP	LOB	LOBA	BMP
1	0	5	4	0	3	0	1	2	0	0	0	1	0	0
1	0	10	15	7	14	0	3	12	0	0	0	0	0	0
1	2	4	1	1	1	0	1	0	0	0	0	0	0	0

Total number of pages: 20

## -scansharing

다음은 **-scansharing** 매개변수를 사용한 출력의 예입니다. 출력은 두 개의 공유 세트를 나타냅니다. 테이블 스캔 세트에는 두 개의 그룹이 있고 블록 인덱스 스캔 세트에는 하나의 그룹이 있습니다.

Database Partition 0 -- Database TPCD -- Active -- Up 0 days 00:00:45

Scan Sets:

TbSpaceID	TableID	ScanObject	NumGroups	Footprint	TableSize	FastScanRate	SlowScanRate
2	3	0	2	11520	22752	2486	1000

Group Information:

FootPrint	NumScannersInGroup
8288	3

Scans In Group :

AgentID	AppID	Mode	Wrappable	Fast/Slow	Speed	ThrottleTime	Absolute Location	Remaining Pages
9768	1173	0	0	1	2486	0	32	22751
11332	1165	0	0	1	2486	0	5056	17727
15466	1155	0	0	1	2486	0	8288	14495

Group Information:

FootPrint	NumScannersInGroup
3232	2

Scans In Group :

AgentID	AppID	Mode	Wrappable	Fast/Slow	Speed	ThrottleTime	Absolute Location	Remaining Pages
15209	1150	0	0	1	2486	0	14080	8703
12103	1148	0	0	1	2486	0	17280	5503

Scan Sets:

TbSpaceID	TableID	ScanObject	NumGroups	Footprint	TableSize	FastScanRate	SlowScanRate
2	3	1	1	9056	22752	1000	1000

Group Information:

FootPrint	NumScannersInGroup
9056	3

Scans In Group :

AgentID	AppID	Mode	Wrappable	Fast/Slow	Speed	ThrottleTime	Relative Location	Estimated Remaining Pages
6170	1209	0	0	1	1000	0	896	13535
13645	1215	0	0	1	1000	0	3552	10879
4371	1204	0	0	1	1000	0	9920	4511

## -serviceclasses

다음은 하나의 서비스 수퍼 클래스 및 해당되는 서브클래스에 대한 서비스 클래스 정보 출력의 샘플입니다.

샘플 서비스 수퍼 클래스 출력:

```

Service Class Name      = SYSDEFAULTSYSTEMCLASS
Service Class ID       = 1
Service Class Type     = Service Superclass
Default Subclass ID   = 11
Effective Service Class State = Enabled
Catalog Service Class State = Enabled
Effective Agent Priority = 0
Catalog Agent Priority = Default
Effective Prefetch Priority = Medium
Catalog Prefetch Priority = Default
Effective Bufferpool Priority = Low
Catalog Bufferpool Priority = Default
Effective Outbound Correlator = None
Catalog Outbound Correlator = None
Work Action Set ID    = N/A
Collect Activity Opt  = None
Collect Request Metrics = Base

Num Connections      = 5

```



Last Statistics Reset Time = 12/16/2008 15:27:42.000000  
 Num Coordinator Connections = 5  
 Coordinator Connections HWM = 5

Associated Workload Occurrences (WLO):

AppHandl	[nod-index]	WL ID	WLO ID	UOW ID	WLO State
10	[000-00010]	0	0	1	UOWWAIT
11	[000-00011]	0	0	1	UOWWAIT
12	[000-00012]	0	0	1	UOWWAIT
13	[000-00013]	0	0	1	UOWWAIT
14	[000-00014]	0	0	1	UOWWAIT

샘플 서비스 서브클래스 출력:

```

Service Class Name      = SYSDEFAULTSUBCLASS
Service Class ID       = 11
Service Class Type     = Service Subclass
Parent Superclass ID   = 1
Effective Service Class State = Enabled
Catalog Service Class State = Enabled
Effective Agent Priority = 0
Catalog Agent Priority = Default
Effective Prefetch Priority = Medium
Catalog Prefetch Priority = Default
Effective Bufferpool Priority = Low
Catalog Bufferpool Priority = Default
Effective Outbound Correlator = None
Catalog Outbound Correlator = None
Collect Activity Opt   = None
Collect Request Metrics = None
Collect Aggr Activity Opt = None
Collect Aggr Request Opt = None
Act Lifetime Histogram Template ID = 1
Act Queue Time Histogram Template ID = 1
Act Execute Time Histogram Template ID = 1
Act Estimated Cost Histogram Template ID = 1
Act Interarrival Time Histogram Template ID = 1
Request Execute Time Histogram Template ID = 1

Access Count          = 0
Last Stats Reset Time = 12/16/2008 15:27:42.000000
Activities HWM        = 0
Activities Completed   = 0
Activities Rejected    = 0
Activities Aborted     = 0
  
```

Associated Agents:

EDU ID	AppHandl	[nod-index]	WL ID	WLO ID	UOW ID	Activity ID
26	10	[000-00010]	0	0	0	0
29	11	[000-00011]	0	0	0	0
28	12	[000-00012]	0	0	0	0
27	13	[000-00013]	0	0	0	0
30	14	[000-00014]	0	0	0	0

Associated Non-agent threads:

PID	TID	Thread Name
6834	2948590480	db2loggr
6834	2947541904	db2loggw
6834	2946493328	db2lfr
6834	2945444752	db2dlock
6834	2944396176	db2pc1nr
6834	2943347600	db2pfchr
6834	2942299024	db2pfchr
6834	2941250448	db2pfchr

**-storagepaths**

다음은 **-storagepaths** 매개변수를 사용한 출력의 예입니다.

```
Database Storage Paths:
Number of Storage Paths      3
```

```
Address          PathID PathState PathName
0x07000000400101C0 0      InUse     /dataPath1
0x0700000040010540 1      DropPending /dataPath2
0x07000000400108C0 2      NotInUse   /PathWithDPE_0 (/PathWithDPE_ $N)
```

## -tablespaces

다음은 자동 스토리지 데이터베이스에 적용 가능한 정보(예: PathsDropped 및 PathID)를 표시하는 db2pd -tablespaces 명령의 출력 샘플입니다.

Tablespace Configuration:

...

Tablespace Statistics:

Address	Id	...	State	MinRecTime	NQuiescers	PathsDropped
0x070000004108AB40	0	...	0x00000000 0	0	0	Yes
0x070000004108B520	1	...	0x00000000 0	0	0	Yes
0x0700000041078100	2	...	0x00000000 0	0	0	Yes

Tablespace Autoresize Statistics:

...

Containers:

Address	TspId	...	PathID	StripeSet	Container
0x070000004108B240	0	...	0	0	/dataPath1/inst/NODE0000/TESTDB/T0000000/C0000000.CAT
0x070000004108B398	0	...	1	0	/dataPath2/inst/NODE0000/TESTDB/T0000000/C0000001.CAT
0x070000004108BBC0	1	...	0	0	/dataPath1/inst/NODE0000/TESTDB/T0000001/C0000000.TMP
0x070000004108BD18	1	...	1	0	/dataPath2/inst/NODE0000/TESTDB/T0000001/C0000001.TMP
0x07000000410787A0	2	...	0	0	/dataPath1/inst/NODE0000/TESTDB/T0000002/C0000000.LRG
0x07000000410788F8	2	...	1	0	/dataPath2/inst/NODE0000/TESTDB/T0000002/C0000001.LRG

인스턴스가 시작된 이후 DMS 테이블 스페이스에 대한 최대 HWM을 표시하기 위해 새 'Max HWM' 컬럼이 db2pd -tablespaces 출력에 추가됩니다. 출력의 'HWM' 컬럼은 현재 HWM으로, 임시 DMS 테이블 스페이스의 경우 사용되는 디스크 스페이스 양의 특정 시점 값을 표시합니다. SMS 테이블 스페이스의 경우, HWM 및 Max HWM에는 값이 없습니다.

쿼리가 발행된 후, db2pd를 사용하여 마지막 트랜잭션에서 사용된 임시 테이블에 대한 인메모리 정보를 사용할 수 있습니다. 아래의 예에서 새 컬럼은 굵은체로 표시되어 있습니다. Max HWM의 값은 항상 HWM 이상이어야 합니다.

```
hotel26:/home/billyp> db2pd -db bill -tablespaces
```

```
Database Partition 0 -- Database BILL -- Active -- Up 0 days 00:02:15
```

Tablespace Configuration:

Address	Id	Type	Content	PageSz	ExtentSz	Auto	Prefetch	BufID	BufIDDisk	FSC	NumCntrs	MaxStripe	LastConsecPg	Name
0x00002B9DCA582720	0	DMS	Regular	4096	4	Yes	4	1	1	Off 1	0	0	3	SYSCATSPACE
0x00002B9DCA583560	1	DMS	UsrTmp	4096	2	Yes	2	1	1	Off 1	0	0	1	DMSUSRTMP
0x00002B9DCA5863E0	2	DMS	Large	4096	32	Yes	32	1	1	Off 1	0	0	31	USERSPACE1
0x00002B9DCA587220	3	DMS	SysTmp	4096	2	Yes	2	1	1	Off 1	0	0	1	DMSYSTMP
0x00002B9DCA58A0A0	4	DMS	Large	4096	4	Yes	4	1	1	Off 1	0	0	3	SYSTOOLSPACE

Tablespace Statistics:

Address	Id	TotalPgs	UsablePgs	UsedPgs	PndFreePgs	FreePgs	HWM	Max HWM	State	MinRecTime	NQuiescers
0x00002B9DCA582720	0	12544	12540	12308	0	232	12308	<b>12308</b>	0x00000000	0	0
0x00002B9DCA583560	1	20000	19998	3266	0	16732	3266	<b>3266</b>	0x00000000	0	0
0x00002B9DCA5863E0	2	7168	7136	3232	0	3904	7072	<b>7072</b>	0x00000000	0	0
0x00002B9DCA587220	3	20000	19998	1700	0	18298	1700	<b>2000</b>	0x00000000	0	0
0x00002B9DCA58A0A0	4	256	252	144	0	108	144	<b>200</b>	0x00000000	0	0

### -temptable

버퍼 풀 읽기 및 쓰기 양을 조사하여 임시 테이블 압축의 효과를 판별하기 위해 시스템 모니터 요소도 사용할 수 있습니다. 다음은 db2pd -temptable 명령의 출력 샘플입니다.

```
hotel26:/home/billyp> db2pd -db billdb -temptable
db_state1 db_state2 db_state3 db_state4 use_count dbname      dbpath
-----
000000001 000000000 000000000 000000000 000000009 BILL      /home/billyp/billyp/NODE0000/SQL00011/
```

#### System Temp Table Stats:

```
Number of Temp Tables   : 0
Comp Eligible Temps    : 0
Compressed Temps       : 0

Total Stored Temp Bytes : 0
Total Bytes Saved      : 0
Total Compressed Rows  : 0
Total Temp Table Rows  : 0
```

#### User Temp Table Stats:

```
Number of Temp Tables   : 0
Comp Eligible Temps    : 0
Compressed Temps       : 0

Total Stored Temp Bytes : 0
Total Bytes Saved      : 0
Total Compressed Rows  : 0
Total Temp Table Rows  : 0
```

시스템 임시 테이블과 사용자 임시 테이블에 대해 동일한 정보가 저장됩니다. 그러나 위의 모든 카운터는 누적되고, 임시 테이블이 삭제될 때 갱신됩니다. 이와 같이, 카운터는 실행기록 정보만 표시합니다.

### -thresholds

다음은 데이터베이스 임계값 및 해당 큐에 대한 임계값 정보 출력의 샘플입니다.

샘플 임계값 출력:

```
Threshold Name          = MAXDBACTIVITIES
Threshold ID            = 6
Domain                  = 10
Domain ID               = 10
Predicate ID            = 90
Maximum Value           = 2
Enforcement              = D
Queueing                 = Y
Queue Size              = 0
Collect Flags           = V
Partition Flags         = C
Execute Flags           = C
Enabled                  = Y
Check Interval (seconds) = -1
Remap Target Serv. Subclass = 0
Log Violation Evmon Record = Y
```

샘플 데이터베이스 임계값 큐 출력:

### Database Threshold Tickets:

임계값에 대한 티켓 정보: 임계값 ID 10이 있는 TH1

활동 ID	UOW ID	분류	AppHandl	[nod-index]
1	6	READ_DML	51	[000-00051]

Queue information for threshold: MAXDBACTIVITIES

Max Concurrency = 2  
Concurrency = 2  
Max Queue Size = 0

Agents Currently Queued:

EDU ID	AppHandl	[nod-index]	Agent	Type	Activity ID	UOW ID
36	14994	[000-14994]	1	4	1	

### -wlocks

다음은 db2pd -wlocks 명령의 출력 샘플입니다.

```
db2pd -wlocks -db mydb2
```

Database Partition 0 -- Database MYDB2 -- Active -- Up 0 days 00:02:17

Locks being waited on :

AppHandl	[nod-index]	TranHdl	Lockname	Type	Mode	Conv	Sts	CoorEDU	AppName	AuthID	AppID
13	[000-00013]	7	0002000B000000000340000452	Row	..X	G	352614	db2bp	VENUS		*LOCAL.venus.071117030309
15	[000-00015]	9	0002000B000000000340000452	Row	.NS	W	1176046	db2bp	VENUS		*LOCAL.venus.071117030358
12	[000-00012]	2	0002000B000000000340000452	Row	.NS	W	1052748	db2bp	VENUS		*LOCAL.venus.071117030231
12	[000-00012]	2	0002000400000000080001652	Row	..X	G	1052748	db2bp	VENUS		*LOCAL.venus.071117030231
14	[000-00014]	8	0002000400000000080001652	Row	.NS	W	634900	db2bp	VENUS		*LOCAL.venus.071117030340

### -workloads

다음은 디폴트 워크로드 SYSDEFAULTUSERWORKLOAD 및 SYSDEFAULTADMWORKLOAD에 대한 출력의 샘플입니다.

Database Partition 0 -- Database SB -- Active -- Up 0 days 00:00:57

Workload Definitions:

Address = 0x00002B3E772ACB40  
WorkloadID = 1  
WorkloadName = SYSDEFAULTUSERWORKLOAD  
DBAccess = ALLOW  
ConcWLOThresID = 0  
ConcWLOThresName = ^H  
MaxConcWLOs = 9223372036854775806  
WLOActsThresName = ^H  
WLOActsThresID = 0  
MaxWLOActs = 9223372036854775806  
ServiceClassID = 13  
Collect Activity Opt = None  
Collect Lock Timeout = Without History  
Collect Deadlock = Without History  
Collect Lock Wait = None  
Collect Aggr Activity Opt = None  
Collect Activity Metrics = Base  
Collect Unit of Work Data = None  
Act Lifetime Histogram Template ID = 1  
Act Queue Time Histogram Template ID = 1  
Act Execute Time Histogram Template ID = 1  
Act Estimated Cost Histogram Template ID = 1  
Act Interarrival Time Histogram Template ID = 1  
Address = 0x00002B3E772ACD50

```

WorkloadID = 2
WorkloadName = SYSDEFAULTADMWORKLOAD
DBAccess = ALLOW
ConcWLOThresID = 0
ConcWLOThresName = ^H
MaxConcWLOs = 9223372036854775806
WLOActsThresName = ^H
WLOActsThresID = 0
MaxWLOActs = 9223372036854775806
ServiceClassID = 13
Collect Activity Opt = None
Collect Lock Timeout = Without History
Collect Deadlock = Without History
Collect Lock Wait = None
Collect Aggr Activity Opt = None
Collect Activity Metrics = Base
Collect Unit of Work Data = None
Act Lifetime Histogram Template ID = 1
Act Queue Time Histogram Template ID = 1
Act Execute Time Histogram Template ID = 1
Act Estimated Cost Histogram Template ID = 1
Act Interarrival Time Histogram Template ID = 1

```

Usage Privilege Holders:

Address	WorkloadID	Type	AuthID
0x00002B3E772BCD60	1	GROUP	PUBLIC

Local Partition Workload Statistics:

```

Address = 0x00002B3E772DA0C0
WorkloadID = 1
WorkloadName = SYSDEFAULTUSERWORKLOAD
NumWLO = 0
LastResetTime = 10/07/2008 16:34:43.000000
WLO HWM = 0
WLOActHWM = 0
WLOCompleted = 0
ActCompleted = 0
ActAborted = 0
ActRejected = 0

```

```

Address = 0x00002B3E7730A0C0
WorkloadID = 2
WorkloadName = SYSDEFAULTADMWORKLOAD
NumWLO = 0
LastResetTime = 10/07/2008 16:34:43.000000
WLO HWM = 0
WLOActHWM = 0
WLOCompleted = 0
ActCompleted = 0
ActAborted = 0
ActRejected = 0

```



---

## 제 240 장 db2pdcfg - 문제점 판별 동작을 위한 DB2 데이터베이스 구성

DB2 데이터베이스 메모리 세트의 플래그가 문제점 판별 동작을 위해 데이터베이스 시스템 동작에 영향을 미치도록 설정합니다.

### 권한 부여

다음 권한 레벨 중 하나가 필요합니다.

- SYSADM 권한 레벨.
- SYSCTRL 권한 레벨.
- SYSMANT 권한 레벨.
- SYSMON 권한 레벨.

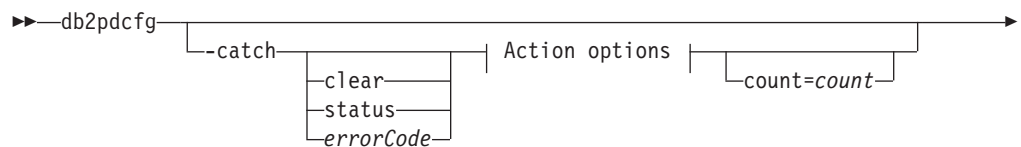
SYSMON 권한 레벨이 부여된 경우 다음 옵션을 사용할 수 없습니다.

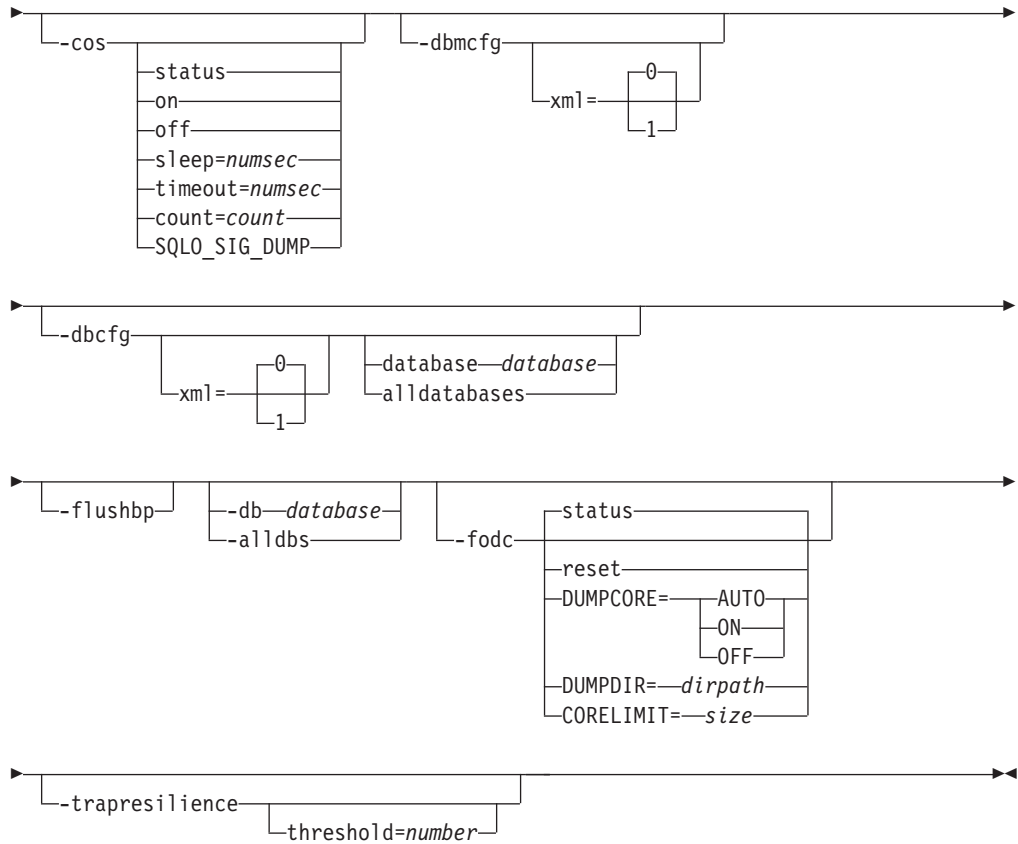
- **catch**
- **cos**
- **dbcfg**
- **dbmcfg**
- **fodc**
- **trapresilience**

### 필수 연결

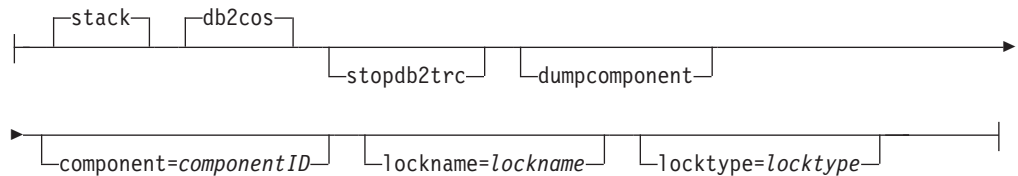
최소한의 연결 요구사항은 없습니다. 그러나 데이터베이스 범위 옵션을 지정한 경우 해당 데이터베이스가 활성화되어야 명령이 요청된 정보를 리턴할 수 있습니다.

### 명령 구문





**Action options:**



**명령 매개변수**

**-catch** 데이터베이스 관리 프로그램이 오류 또는 경고를 캐치하도록 지시합니다.

**clear** 설정된 캐치 플래그를 지웁니다.

**status** 설정된 캐치 플래그를 표시합니다.

*errorCode*

설정된 특정 플래그를 캐치합니다.

가능한 *errorCode* 옵션은 다음과 같습니다.

- *sqlCode*[,*reasonCode*] / *sqlCode*=*sqlCode*[,*reasonCode*]
- ZRC(16진수 또는 정수)
- ZRC #define(예: SQLP\_LTIMEOUT)



- ECF(16진수 또는 정수)
- “deadlock” 또는 “locktimeout”

**stack** db2diag 로그 파일에서 스택 추적을 생성합니다. 디폴트.

**db2cos**

bin 디렉토리에서 발견된 db2cos 콜아웃 스크립트를 실행합니다. 디폴트.

**stopdb2trc**

db2trc 명령을 중지합니다.

**dumpcomponent**

구성요소 플래그를 덤프합니다.

**component=componentID**

구성요소 ID.

**lockname=lockname**

특정 잠금을 캐치하기 위한 잠금 이름  
(lockname=000200030000001F0000000052).

**locktype=locktype**

특정 잠금을 캐치하기 위한 잠금 유형(locktype=R 또는 locktype=52).

**count=count**

데이터베이스 관리 프로그램이 데이터베이스 관리 프로그램 트랩 동안 db2cos를 실행하는 횟수. 디폴트는 255입니다.

**-cos** 데이터베이스 관리 프로그램 트랩 시 db2cos 콜아웃 스크립트를 호출하는 방법을 데이터베이스 관리 프로그램에 지시합니다.

**status** 상태를 인쇄합니다.

**off** 데이터베이스 관리 프로그램 트랩 동안 db2cos에 대한 데이터베이스 관리 프로그램 호출을 해제합니다.

**on** 데이터베이스 관리 프로그램 트랩 동안 db2cos에 대한 데이터베이스 관리 프로그램 호출을 설정합니다.

**sleep=numsec**

db2cos에서 생성되는 출력 파일의 크기를 점검하는 사이에 유희할 시간. 디폴트는 3초입니다.

**timeout=numsec**

db2cos 스크립트가 완료된 것으로 간주하기 전에 기다릴 시간. 디폴트는 30초입니다.

**count=count**

데이터베이스 관리 프로그램 트랩 동안 db2cos를 실행할 횟수. 디폴트는 255입니다.

### **SQLO\_SIG\_DUMP**

SQLO\_SIG\_DUMP 신호가 수신될 때 db2cos 실행을 사용합니다.

### **-dbmcfg**

DBM 구성 예약 비트맵을 설정합니다. 이 옵션은 IBM DB2 Service에서 확보할 수 있는 암호 보호입니다.

**xml=0 | 1**

값 0(디폴트) 또는 1(인스턴스에 xml 데이터가 있음).

**-dbcfg** 데이터베이스 구성 예약 비트맵을 설정합니다. 이 옵션은 IBM DB2 Service에서 확보할 수 있는 암호 보호입니다.

**xml=0 | 1**

값 0(디폴트) 또는 1(데이터베이스에 xml 데이터가 있음).

**-fodc** DB2 데이터베이스 메모리 세트에서 플래그를 설정합니다. 이 플래그는 FODC(First Occurrence Data Collection)가 수반되는 문제점 판별 상황에서 데이터베이스 시스템 동작에 영향을 줍니다.

지원되는 -fodc 옵션과 해당되는 가능한 값 및 디폴트는 다음과 같습니다.

**reset** 모든 FODC 옵션을 디폴트로 리스토어합니다.

**status** 모든 FODC 옵션의 상태를 표시합니다. 이는 디폴트 옵션입니다. 즉, FODC 상태는 db2pdcfg가 매개변수 없이 호출될 때 표시됩니다.

### **DUMPCORE=**

UNIX 및 Linux 플랫폼에서만 코어 파일 생성이 가능하거나 불가능하도록 설정합니다.

#### **AUTO**

트랩이 지속되고 인스턴스가 종료되면 코어 파일이 생성됩니다.

**ON** 코어 파일 생성이 가능하며 **DB2RESILIENCE** 레지스트리 변수 설정을 대체할 수 있습니다.

**OFF** 코어 파일 생성이 불가능합니다.

### **DUMPPDIR=dirpath**

코어 파일 또는 공유 메모리 덤프가 작성될 디렉토리의 절대 경로 이름을 지정합니다. 이 옵션은 코어 파일 및 공유 메모리 덤프뿐만 아니라 FODC 패키지 외부에 저장해야 하는 다른 대형 2진 덤프에 사용할 수 있습니다. 디폴트는 DIAGPATH 디렉토리이거나, DIAGPATH가 정의되지 않은 경우 디폴트 진단 디렉토리입니다.

## CORELIMIT=size

작성된 코어 파일의 최대 크기. 이 값은 현재 코어 파일 크기 한계 설정보다 우선합니다. 코어 파일은 상당히 클 수 있으므로 사용 가능한 파일 시스템 스페이스를 고려해야 합니다. 크기는 DB2 구성과 문제점 발생 시 프로세스의 상태에 따라 다릅니다. CORELIMIT를 설정하지 않은 경우, DB2는 코어 파일 크기를 현재 ulimit 설정과 같은 값으로 설정합니다. 단, unlimited의 코어 ulimit 설정을 대체하기 위해 8GB 값이 사용되는 AIX는 예외입니다.

db2pdcfg를 사용하여 CORELIMIT를 변경하는 경우, 일반적인 UNIX 액세스 사용 권한이 있어야 하며, 일부 경우에는 CORELIMIT가 ulimit 설정을 초과하지 못할 수도 있습니다. DB2FODC 레지스트리 변수를 사용하여 db2start에서 값을 변경하거나 DB2 제품을 시작하기 전에 큰 ulimit 설정을 사용하십시오.

AIX에서, "unlimited"의 사용자 코어 한계는 DB2 서버 프로세스에 대해서만 8GB 값으로 대체됩니다. 이는 DPF 또는 Intra-parallel-enabled 환경에서 대형 FCM(Fast Communication Manager) 메모리 영역의 덤프를 방지하기 위한 것입니다. AIX에서, 이 메모리 영역의 대형 분할 영역(간혹 문제점 진단에서 필요한)도 사전 할당되거나 커밋되지 않습니다(시스템 메모리의 도움을 받지 않음). 이 사전 할당은 FCM에 대한 동적 메모리 요구사항을 지원하지만, 이 메모리를 코어 파일로 덤프하면 해당 메모리의 중간 커밋이 발생하고 많은 경우에 불필요한 대형 코어 파일이 생성됩니다. 8GB보다 큰 코어 덤프가 필요한 경우, 사용자 코어 한계는 unlimited와 다른 큰 값(예: RAM 크기)으로 설정해야 합니다. 또는 CORELIMIT를 충분히 큰 값으로 설정할 수 있습니다. 사용자 코어 한계나 CORELIMIT에서의 변경사항은 다음 DB2 인스턴스를 재사용할 때까지 적용되지 않습니다.

### -flushbp

이 옵션은 버퍼 풀에서 디스크로 더티 페이지를 플러시합니다. **-db** 키워드와 뒤에 오는 활성 데이터베이스 이름을 다음과 같이 지정할 수 있습니다.

```
db2pdcfg -flushbp -db <dbname>
```

모든 활성 데이터베이스에 대한 모든 버퍼 풀에서 더티 페이지를 플러시하려면 다음과 같이 **-alldbs** 키워드를 지정하십시오.

```
db2pdcfg -flushbp -alldbs
```

### -trapresilience

이 옵션은 문제점 판별을 위한 트랩 회복 매개변수를 표시하거나 수정합니다.

다음은 이 옵션이 지정되는 경우의 샘플 출력입니다.

```
DB2 trap resilience is enabled.  
Current threshold setting : 0 (threshold disabled)  
Number of traps sustained : 0
```

**threshold=number**

디폴트값: 0(임계값 사용 안함)

최소값: 0

최대값: 4294967295

threshold= 옵션 다음에 숫자를 지정하면 인스턴스 수명 동안 지속될 트랩의 상한이 설정됩니다. 이 임계값을 제외하면 인스턴스는 다음 트랩의 지속 여부에 관계없이 중지됩니다.

다음은 이 옵션이 지정되는 경우의 샘플 출력입니다.

```
db2pdcfg -trapresilience threshold=1
```

```
DB2 trap resilience threshold is set to 1
```

## 사용 시 참고사항

db2pdcfg는 FODC 옵션을 동적으로 변경(온라인)하기 위한 메소드입니다.

db2pdcfg는 DB2 데이터베이스 메모리에서 플래그를 설정하므로, db2pdcfg 도구에 대해 수행된 변경사항은 인스턴스가 가동되는 동안에만 활성화됩니다. 변경사항이 영구적이 되도록 하려면 DB2FODC 레지스트리 변수를 사용하십시오.

-fodc 옵션에서는 일부 설정이 variable=value 형식으로 지정됩니다. 단일 명령행에 여러 개의 옵션을 지정할 수 있습니다.

```
db2pdcfg -fodc DUMPCORE=ON -fodc CORELIMIT=8GB
```

또는 공백을 사용하여 몇 개의 설정을 단일 명령행 문자열로 병합할 수도 있습니다.

```
db2pdcfg -fodc DUMPCORE=ON CORELIMIT=8GB
```

옵션 없이 db2pdcfg 명령을 실행하면 트랩 회복(굵게 강조 표시된)에 대해 비형식화 요약 출력이 제공됩니다.

```
quark:/home/quark/marvin> db2pdcfg  
Current PD Control Block Settings:
```

```
All error catch flag settings cleared.
```

```
db2cos is enabled for engine traps.
```

```
PD Bitmap:      0x1000  
Sleep Time:     3  
Timeout:        30
```

```
Current bitmap value: 0x0
```

```
Instance is not in a sleep state
```

```
DB2 trap resilience is enabled.
```

```
Current threshold setting : 0 (threshold disabled)
```

**Number of traps sustained : 0**  
FODC (First Occurrence Data Capture) options:  
Dump directory for large objects (DUMPDIR)= /home/quark/marvin/sql1lib/db2dump/  
Dump Core files (DUMPCORE)= ON



---

## 제 241 장 db2perf - 데이터베이스 성능 값 재설정

하나 이상의 데이터베이스에 대한 성능 값을 재설정합니다. Windows 운영 체제에서 성능 모니터와 함께 사용됩니다.

### 권한 부여

로컬 관리자

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

**-d** DCS 데이터베이스에 대한 성능 값이 재설정되도록 지정합니다.

*dbalias*

성능 값이 재설정되어야 하는 데이터베이스를 지정합니다. 데이터베이스가 지정되지 않은 경우 모든 활성 데이터베이스에 대한 성능 값이 재설정됩니다.

### 예:

다음 예는 모든 활성 DB2 데이터베이스에 대한 성능 값을 재설정합니다.

```
db2perf
```

다음 예는 특정 DB2 데이터베이스에 대한 성능 값을 재설정합니다.

```
db2perf dbalias1 dbalias2
```

다음 예는 모든 활성 DB2 DCS 데이터베이스에 대한 성능 값을 재설정합니다.

```
db2perf -d
```

다음 예는 특정 DB2 DCS 데이터베이스에 대한 성능 값을 재설정합니다.

```
db2perf -d dbalias1 dbalias2
```

## 사용 시 참고사항

응용프로그램이 DB2 모니터 API를 호출할 때 리턴되는 정보는 일반적으로 DB2 서버가 시작된 이후의 누적 값입니다. 그러나 종종 성능 값을 재설정하고 테스트를 실행하고 값을 다시 재설정 후 테스트를 다시 실행하는 것이 유용합니다.

프로그램이 관련 DB2 서버 인스턴스(즉, db2perf를 실행하는 세션에서 db2instance에 보유한 인스턴스)에 대한 데이터베이스 성능 정보에 현재 액세스하는 모든 프로그램에 대한 값을 재설정합니다. db2perf를 호출하면 명령이 실행될 때 DB2 성능 정보에 리모트로 액세스하는 누구나 보는 값도 재설정합니다.

db2ResetMonitor API는 응용프로그램이 특정 데이터베이스에 대해 전역이 아니라 로컬로 표시되는 값을 재설정할 수 있게 합니다.



---

## 제 242 장 db2perfi - 성능 카운터 등록 유틸리티

Windows 운영 체제에 DB2 성능 카운터를 추가합니다. Windows 성능 모니터가 DB2 및 DB2 Connect 성능 정보에 액세스할 수 있게 하려면 이 작업을 수행해야 합니다.

### 권한 부여

로컬 관리자

### 필수 연결

없음

### 명령 구문

```
▶▶ db2perfi -i _____▶▶
```

### 명령 매개변수

- i DB2 성능 카운터를 등록합니다.
- u DB2 성능 카운터를 등록 해제합니다.

### 사용 시 참고사항

db2perfi -i 명령은 다음을 수행합니다.

1. DB2 카운터 오브젝트의 이름과 설명을 Windows 레지스트리에 추가합니다.
2. 다음과 같이 Windows 레지스트리의 서비스 키에 레지스트리 키를 작성하십시오.

```
HKEY_LOCAL_MACHINE
  #System
    #CurrentControlSet
      #Services
        #DB2_NT_Performance
          #Performance
            Library=Name of the DB2 performance support DLL
            Open=Open function name, called when the DLL is
              first loaded
            Collect=Collect function name, called to request
              performance information
            Close=Close function name, called when the DLL is
              unloaded
```



---

## 제 243 장 db2perfr - 성능 모니터 등록 도구

Windows 운영 체제에서 성능 모니터와 함께 사용됩니다. db2perfr 명령은 성능 카운터를 액세스할 때 DB2에 관리자 사용자 이름 및 암호를 등록하는 데 사용됩니다. 이것은 리모트 성능 모니터 요청이 자기 자신을 DB2 데이터베이스 관리 프로그램에 올바르게 식별하고 관련 DB2 성능 정보에 액세스될 수 있게 합니다. 또한 성능 로그 기능을 사용하여 카운터 정보를 파일에 로그하려는 경우 관리자 사용자 이름 및 암호를 등록해야 합니다.

### 권한 부여

로컬 관리자

### 필수 연결

없음

### 명령 구문

```
db2perfr -r username password
```

### 명령 매개변수

- r 사용자 이름 및 암호를 등록합니다.
- u 사용자 이름 및 암호를 등록 해제합니다.

### 사용 시 참고사항

- 사용자 이름 및 암호 조합이 DB2에 등록된 후에는 성능 모니터의 로컬 인스턴스도 해당 사용자 이름 및 암호를 사용하여 명시적으로 로그인합니다. 이것은 DB2에 등록된 사용자 이름 정보가 일치하지 않는 경우 성능 모니터의 로컬 세션은 DB2 성능 정보를 표시하지 않음을 의미합니다.
- 사용자 이름 및 암호 조합은 Windows 보안 데이터베이스에 저장된 사용자 이름 및 암호 값과 일치하도록 유지되어야 합니다. 사용자 이름 또는 암호가 Windows 보안 데이터베이스에서 변경되는 경우 리모트 성능 모니터링에 사용되는 사용자 이름과 암호 조합이 재설정되어야 합니다.
- 디폴트 Windows 성능 모니터 사용자 이름인 SYSTEM은 DB2 예약어이며 사용할 수 없습니다.



---

## 제 244 장 db2rbind - 모든 패키지 리바인드

데이터베이스에서 패키지를 리바인드합니다.

### 권한 부여

*dbadm*

### 필수 연결

없음

### 명령 구문

```
db2rbind database -l logfile [all] [-u userid -p 암호]
[-r [conservative|any]]
```

### 명령 매개변수

*database*

패키지의 유효성을 다시 확인할 데이터베이스의 별명 이름을 지정합니다.

**-l logfile**

패키지 유효성 다시 확인 프로시저의 원인이 되는 오류 기록을 위해 사용할 (선택적) 경로와 (필수) 파일 이름을 지정합니다.

**all**

모든 유효한 패키지와 유효하지 않은 패키지의 리바인드를 수행함을 지정합니다. 이 옵션을 지정하지 않으면 데이터베이스의 모든 패키지가 조사되지만 유효하지 않은 것으로 표시된 패키지만 리바인드되어, 응용프로그램 실행 중에는 내재적으로 리바인드되지 않습니다.

**-u userid**

사용자 ID. 이 매개변수는 암호를 지정한 경우에 지정해야 합니다.

**-p password**

암호. 이 매개변수는 사용자 ID를 지정한 경우에 지정해야 합니다.

**-r**

분석. 패키지의 리바인드 수행 시 제한적 바인딩 시맨틱이 사용되는지 여부를 지정합니다. 이는 분석을 위해 SQL 경로를 사용하는 새 오브젝트가 패키지에서 정적 DML 문에 대한 분석 중에 고려되는지 여부에 영향을 줍니다. 이 옵션은 DRDA에서 지원되지 않습니다. 가능한 값은 다음과 같습니다.

### **conservative**

오브젝트 분석을 위해 SQL 경로를 사용하는 오브젝트에 대한 참조 분석 시 SQL 경로에서 마지막 명시적 바인드 시간소인 이전에 정의된 오브젝트만 고려됩니다. 제한적 바인딩 시맨틱이 사용됩니다. 이것은 디폴트값입니다. 이 옵션은 작동 불능 패키지에 대해 지원되지 않습니다.

### **any**

오브젝트 분석을 위해 SQL 경로를 사용하는 오브젝트에 대한 참조 분석에 SQL 경로에 있는 가능한 모든 일치이 고려됩니다. 제한적 바인딩 시맨틱은 사용되지 않습니다.

## **사용 시 참고사항**

- 이 명령은 리바인드 API(sqlarbind)를 사용하여 데이터베이스에 있는 모든 패키지의 유효성 다시 확인을 시도합니다.
- db2rbind는 반드시 사용하지 않아도 됩니다.
- 유효하지 않은 패키지의 경우, 패키지가 처음 사용될 때 내재적으로 패키지 유효성 다시 확인이 발생하도록 선택할 수 있습니다. REBIND 또는 BIND 명령을 사용하여 선택적으로 패키지의 유효성을 다시 확인하도록 선택할 수 있습니다.
- 패키지 중 하나의 리바인드에서 교착 상태 또는 잠금 시간종료가 발생하는 경우 모든 패키지의 리바인드가 롤백됩니다.

---

## 제 245 장 db2relocatedb - 데이터베이스 재배치

이 명령은 사용자가 제공하는 구성 파일에 지정된 대로 데이터베이스의 이름을 바꾸거나 데이터베이스 또는 데이터베이스의 일부(예: 컨테이너 및 로그 디렉토리)를 재배치합니다. 이 도구는 DB2 인스턴스와 데이터베이스 자원 파일에 대해 필요한 변경사항을 작성합니다.

### 권한 부여

없음

### 명령 구문

```
►►—db2relocatedb—-f—configFilename—►►
```

### 명령 매개변수

#### -f configFilename

데이터베이스 재배치에 필요한 구성 정보를 포함하는 파일의 이름을 지정합니다. 상대적 또는 절대적 파일 이름이 될 수 있습니다. 구성 파일의 형식은 다음과 같습니다.

```
DB_NAME=oldName,newName
DB_PATH=oldPath,newPath
INSTANCE=oldInst,newInst
NODENUM=nodeNumber
LOG_DIR=oldDirPath,newDirPath
CONT_PATH=oldContPath1,newContPath1
CONT_PATH=oldContPath2,newContPath2
...
STORAGE_PATH=oldStoragePath1,newStoragePath1
STORAGE_PATH=oldStoragePath2,newStoragePath2
...
```

여기서:

#### **DB\_NAME**

재배치할 데이터베이스의 이름을 지정합니다. 데이터베이스 이름이 변경되는 경우 이전 이름과 새 이름 모두 지정해야 합니다. 필수 필드입니다.

#### **DB\_PATH**

재배치할 데이터베이스의 원래 경로를 지정합니다. 데이터베이스 경로가 변경되는 경우, 이전 경로와 새 경로 모두 지정해야 합니다. 필수 필드입니다.

## INSTANCE

데이터베이스가 있는 인스턴스를 지정합니다. 데이터베이스가 새 인스턴스로 이동하는 경우, 이전 인스턴스와 새 인스턴스 모두 지정해야 합니다. 필수 필드입니다.

## NODENUM

데이터베이스 노드의 노드 번호가 변경됨을 지정합니다. 디폴트는 0입니다.

## LOG\_DIR

로그 경로의 해당 위치에서 변경을 지정합니다. 로그 경로가 변경되는 경우, 이전 경로와 새 경로 모두 지정해야 합니다. 로그 경로가 데이터베이스 경로 아래에 있는 경우(경로가 자동으로 갱신되는 경우) 이 스펙은 선택적입니다.

## CONT\_PATH

테이블 스페이스 컨테이너의 해당 위치에서 변경을 지정합니다. 이전 및 새 컨테이너 경로 모두 지정해야 합니다. 여러 개의 컨테이너 경로 변경사항이 작성되는 경우 여러 개의

CONT\_PATH 라인을 제공할 수 있습니다. 컨테이너 경로가 데이터베이스 경로 아래에 있는 경우(경로가 자동으로 갱신되는 경우) 이 스펙은 선택적입니다. 동일한 이전 경로가 공통적인 새 경로로 교체되는 두 개 이상의 컨테이너를 변경하는 경우, 단일 CONT\_PATH 항목을 사용할 수 있습니다. 이와 같은 경우, 별표(\*)는 이전 경로와 새 경로 모두에서 와일드 카드로 사용할 수 있습니다.

## STORAGE\_PATH

자동 스토리지가 사용 가능한 데이터베이스에만 적용 가능합니다. 데이터베이스에 대한 스토리지 경로 중 하나의 위치에서 변경을 지정합니다. 이전 스토리지 경로와 새 스토리지 경로 모두 지정해야 합니다. 몇 개의 스토리지 경로 변경사항이 작성되는 경우 여러 개의 STORAGE\_PATH 라인을 제공할 수 있습니다.

공백 라인이나 주석 문자(#)로 시작하는 라인은 무시됩니다.

예:

예 1

/home/db2inst1 경로에 있는 db2inst1 인스턴스에서 데이터베이스 TESTDB의 이름을 PRODDB로 변경하려면 다음 구성 파일을 작성하십시오.

```
DB_NAME=TESTDB,PRODDB
DB_PATH=/home/db2inst1
INSTANCE=db2inst1
NODENUM=0
```



구성 파일을 relocate.cfg로 저장하고 다음 명령을 사용하여 데이터베이스 파일을 변경하십시오.

```
db2relocatedb -f relocate.cfg
```

### 예 2:

데이터베이스 DATAB1을 경로 /dbpath의 인스턴스 jsmith에서 인스턴스 prodinst로 이동하려면 다음을 수행하십시오.

1. 디렉토리 /dbpath/jsmith의 파일을 /dbpath/prodinst로 이동하십시오.
2. db2relocatedb 명령과 함께 다음 구성 파일을 사용하여 데이터베이스 파일 변경사항을 작성하십시오.

```
DB_NAME=DATAB1
DB_PATH=/dbpath
INSTANCE=jsmith,prodinst
NODENUM=0
```

### 예 3

데이터베이스 PRODDB는 경로 /databases/PRODDB의 inst1 인스턴스에 있습니다. 두 테이블 스페이스 컨테이너의 위치는 다음과 같이 변경해야 합니다.

- SMS 컨테이너 /data/SMS1은 /DATA/NewSMS1로 이동해야 합니다.
- DMS 컨테이너 /data/DMS1은 /DATA/DMS1로 이동해야 합니다.

실제 디렉토리 및 파일이 새 위치로 이동된 후, db2relocatedb 명령과 함께 다음 구성 파일을 사용하여 새 위치를 인식하도록 데이터베이스 파일 변경사항을 작성할 수 있습니다.

```
DB_NAME=PRODDB
DB_PATH=/databases/PRODDB
INSTANCE=inst1
NODENUM=0
CONT_PATH=/data/SMS1,/DATA/NewSMS1
CONT_PATH=/data/DMS1,/DATA/DMS1
```

### 예 4

데이터베이스 TESTDB는 인스턴스 db2inst1에 있으며 경로 /databases/TESTDB에서 작성되었습니다. 테이블 스페이스는 다음 컨테이너를 사용하여 작성되었습니다.

```
TS1
TS2_Cont0
TS2_Cont1
/databases/TESTDB/TS3_Cont0
/databases/TESTDB/TS4/Cont0
/Data/TS5_Cont0
/dev/rTS5_Cont1
```

TESTDB는 새 시스템으로 이동됩니다. 새 시스템의 인스턴스는 newinst가 되고 데이터베이스의 위치는 /DB2가 됩니다.

데이터베이스를 이동할 때 /databases/TESTDB/db2inst1 디렉토리에 존재하는 모든 파일은 /DB2/newinst 디렉토리로 이동해야 합니다. 이는 처음 5개의 컨테이너가 이동의 일부로 재배치됨을 의미합니다. (처음 세 개는 데이터베이스 디렉토리에 상대적이고 다음 두 개는 데이터베이스 경로에 상대적입니다.) 이 컨테이너는 데이터베이스 디렉토리나 데이터베이스 경로 내에 위치하므로, 구성 파일에 나열하지 않아도 됩니다. 나머지 두 개의 컨테이너가 새 시스템의 다른 위치로 이동되는 경우, 그 컨테이너는 구성 파일에 나열해야 합니다.

실제 디렉토리 및 파일이 새 위치로 이동된 후, db2relocatedb와 함께 다음 구성 파일을 사용하여 새 위치를 인식하도록 데이터베이스 파일 변경사항을 작성할 수 있습니다.

```
DB_NAME=TESTDB
DB_PATH=/databases/TESTDB,/DB2
INSTANCE=db2inst1,newinst
NODENUM=0
CONT_PATH=/Data/TS5_Cont0,/DB2/TESTDB/TS5_Cont0
CONT_PATH=/dev/rTS5_Cont1,/dev/rTESTDB_TS5_Cont1
```

## 예 5

데이터베이스 TESTDB는 데이터베이스 파티션 서버 10 및 20에 두 개의 데이터베이스 파티션을 가지고 있습니다. 인스턴스는 servinst이고 데이터베이스 경로는 두 데이터베이스 파티션 서버 모두 /home/servinst입니다. 데이터베이스의 이름은 SERVDB로 변경되고 데이터베이스 경로는 두 데이터베이스 파티션 서버 모두에서 /databases로 변경됩니다. 또한 로그 디렉토리가 데이터베이스 파티션 서버 20에서, /testdb\_logdir로부터 /servdb\_logdir로 변경됩니다.

변경사항은 두 데이터베이스 파티션 모두에 대해 작성되므로, 각 데이터베이스 파티션에 대해 구성 파일을 작성하고 해당되는 구성 파일을 사용하여 각 데이터베이스 파티션 서버에서 db2relocatedb를 실행해야 합니다.

데이터베이스 파티션 서버 10에서, 다음 구성 파일이 사용됩니다.

```
DB_NAME=TESTDB,SERVDB
DB_PATH=/home/servinst,/databases
INSTANCE=servinst
NODE_NUM=10
```

데이터베이스 파티션 서버 20에서, 다음 구성 파일이 사용됩니다.

```
DB_NAME=TESTDB,SERVDB
DB_PATH=/home/servinst,/databases
INSTANCE=servinst
NODE_NUM=20
LOG_DIR=/testdb_logdir,/servdb_logdir
```

## 예 6

데이터베이스 MAINDB는 경로 /home/maininst의 maininst 인스턴스에 있습니다. 네 개의 테이블 스페이스 컨테이너의 위치는 다음과 같이 변경해야 합니다.

```
/maininst_files/allconts/C0 needs to be moved to /MAINDB/C0
/maininst_files/allconts/C1 needs to be moved to /MAINDB/C1
/maininst_files/allconts/C2 needs to be moved to /MAINDB/C2
/maininst_files/allconts/C3 needs to be moved to /MAINDB/C3
```

실제 디렉토리 및 파일이 새 위치로 이동된 후, db2relocatedb 명령과 함께 다음 구성 파일을 사용하여 새 위치를 인식하도록 데이터베이스 파일 변경사항을 작성할 수 있습니다.

모든 컨테이너에 유사한 변경이 작성됩니다. 즉, /maininst\_files/allconts/는 와일드 카드 문자가 있는 단일 항목을 사용할 수 있도록 /MAINDB/로 교체됩니다.

```
DB_NAME=MAINDB
DB_PATH=/home/maininst
INSTANCE=maininst
NODE_NUM=0
CONT_PATH=/maininst_files/allconts/*, /MAINDB/*
```

## 사용 시 참고사항

데이터베이스가 속하는 인스턴스가 변경되는 경우, 이 명령을 실행하기 전에 다음을 수행하여 인스턴스 및 데이터베이스 지원 파일의 변경사항이 작성되는지 확인해야 합니다.

- 데이터베이스가 다른 인스턴스로 이동되는 경우 새 인스턴스를 작성하십시오. 새 인스턴스는 데이터베이스가 현재 상주하는 인스턴스와 동일한 릴리스 레벨에 있어야 합니다.
- 복사하는 데이터베이스에 속하는 파일 및 디바이스를 새 인스턴스가 있는 시스템으로 복사하십시오. 필요에 따라 경로 이름을 변경해야 합니다. 그러나 데이터베이스 파일이 이동되는 디렉토리에 이미 데이터베이스가 있는 경우, 실수로 기존 sqldbdir 파일 위에 겹쳐써서 기존 데이터베이스에 대한 참조를 제거할 수 있습니다. 이 시나리오에서, db2relocatedb 유틸리티는 사용할 수 없습니다. db2relocatedb 대신, 경로 재지정된 리스토어 조작을 사용할 수 있습니다.
- 인스턴스 소유자가 소유하도록 복사된 파일/디바이스의 사용 권한을 변경하십시오.

ALTER TABLESPACE MANAGED BY AUTOMATIC STORAGE 문을 사용하여 자동 스토리지를 사용하도록 변환된 테이블 스페이스에 대한 기존의 사용자 작성 컨테이너를 이동하는 경우에는 db2relocatedb 명령을 사용할 수 없습니다.

인스턴스가 변경되는 경우 도구는 새 인스턴스 소유자가 실행해야 합니다.

파티션된 데이터베이스 환경에서 이 도구는 변경해야 하는 모든 데이터베이스 파티션에 대해 실행해야 합니다. 데이터베이스 파티션마다 변경될 데이터베이스 파티션의 NODENUM

값을 포함하는 개별 구성 파일을 제공해야 합니다. 예를 들어, 데이터베이스 이름이 변경되는 경우 모든 데이터베이스 파티션이 영향을 받으며 각 데이터베이스 파티션의 개별 구성 파일과 함께 `db2relocatedb` 명령을 실행해야 합니다. 단일 데이터베이스 파티션에 속하는 컨테이너를 이동하는 경우 `db2relocatedb` 명령은 해당 데이터베이스 파티션에서 한 번만 실행해야 합니다.

로드 진행 중이거나 `LOAD RESTART` 또는 `LOAD TERMINATE` 명령이 완료되기를 기다리고 있는 데이터베이스를 재배치하는 경우에는 `db2relocatedb` 명령을 사용할 수 없습니다.

**제한사항:** 파티션된 데이터베이스 환경에서, 노드가 동일한 디바이스에 있는 두 개 이상의 논리적 파티션 중 하나인 경우 전체 노드를 재배치할 수 없습니다.

---

## 제 246 장 db2rfpen - 롤 포워드 보류 상태 재설정

데이터베이스를 롤 포워드 보류 상태로 둡니다. 고가용성 재해 복구(HADR)를 사용하는 경우 데이터베이스가 표준 데이터베이스로 재설정됩니다.

### 권한 부여

없음

### 필수 연결

없음

### 명령 구문

```
▶▶ db2rfpen -ON database_alias -file log_control_file -path log_control_files_dir ▶▶
```

### 명령 매개변수

#### *database\_alias*

롤 포워드 보류 상태가 될 데이터베이스의 이름을 지정합니다. 고가용성 재해 복구(HADR)를 사용하는 경우 데이터베이스가 표준 데이터베이스로 재설정됩니다.

#### **-file** *log\_control\_file*

로그 제어 파일 경로와 파일 이름을 지정합니다.

#### **-path** *log\_control\_files\_dir*

로그 제어 파일, SQLOGCTL.LFH.1과 해당 미리 사본, SQLOGCTL.LFH.2가 있는 디렉토리의 전체 경로를 지정합니다.

### 사용 시 참고사항

-file 매개변수를 사용하면 지정된 로그 제어 파일(예를 들어, SQLOGCTL.LFH.1 또는 SQLOGCTL.LFH.2)만 갱신됩니다. 이 경우 두 파일이 동기화되지 않습니다. 이러한 이유로 인해 가능한 경우 -file 옵션 대신 *database\_alias* 또는 -path 옵션을 사용하는 것이 좋습니다.



---

## 제 247 장 db2rmicons - 주메뉴 명령에서 DB2 도구 제거

DB2 도구의 주메뉴 항목을 제거합니다.

Linux 운영 체제에서 db2rmicons 명령은 현재 사용자의 DB2 도구에 대한 주메뉴 항목을 제거합니다. DB2 도구의 주메뉴 항목은 db2rmicons 명령을 실행하여 수동으로 제거되거나 특정 DB2 명령이 실행될 때 자동으로 제거됩니다(예: db2\_deinstall 또는 db2idrop). 루트 서버가 아닌 설치의 경우, db2\_deinstall 명령은 루트 서버가 아닌 설치와 관련된 DB2 인스턴스의 항목을 제거합니다.

### 권한 부여

없음

### 명령 구문

```
▶▶ db2rmicons -h
```

### 명령 매개변수

**-h**     사용법 정보를 표시합니다.



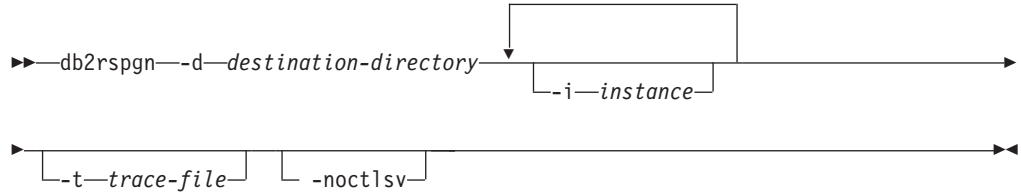


---

## 제 248 장 db2rspgn - 응답 파일 생성 프로그램

현재 사본의 응답 파일 및 인스턴스 구성 프로파일을 생성합니다. 이러한 생성된 파일은 다른 머신에서 정확한 설정을 재작성하는 데 사용됩니다.

### 명령 구문



### 명령 매개변수

#### **-d** *destination-directory*

생성된 파일의 출력 디렉토리 전체 경로를 지정합니다. 지정된 출력 디렉토리가 기존 디렉토리인 경우 해당 디렉토리는 비어 있고 쓰기 가능한 디렉토리여야 합니다. 지정된 출력 디렉토리가 없는 경우 쓰기 가능한 위치에 새 디렉토리가 작성됩니다. 이 매개변수는 필수 매개변수입니다.

#### **-i** *instance*

지정된 인스턴스 구성을 생성하고 생성된 응답 파일 및 인스턴스 구성 프로파일에 이 정보를 저장합니다. 이 매개변수는 선택적 매개변수입니다. 디폴트로 모든 인스턴스가 선택됩니다. 여러 인스턴스를 지정하려면 이 매개변수를 여러 번 지정합니다(예: -i db2inst1 -i db2inst3).

#### **-t** *trace-file*

Linux 및 UNIX 운영 체제용. 디버그 모드를 활성화합니다. 디버그 정보는 trace-file로 지정된 파일 이름에 작성됩니다.

#### **-noctlsv**

Windows 운영 체제 전용. 제어 서버 인스턴스에 대해 인스턴스 구성 프로파일 파일이 생성되지 않음을 나타냅니다. 이 매개변수는 선택적 매개변수입니다.



## 제 249 장 db2sampl - 샘플 데이터베이스 작성

SAMPLE이라고 하는 샘플 데이터베이스를 작성합니다.

이 데이터베이스는 처음 작성될 때 자동으로 구성되지 않습니다. 사용자는 나중에 SAMPLE 데이터베이스에 대해 AUTOCONFIGURE 명령을 발행할 수 있습니다.

### 권한 부여

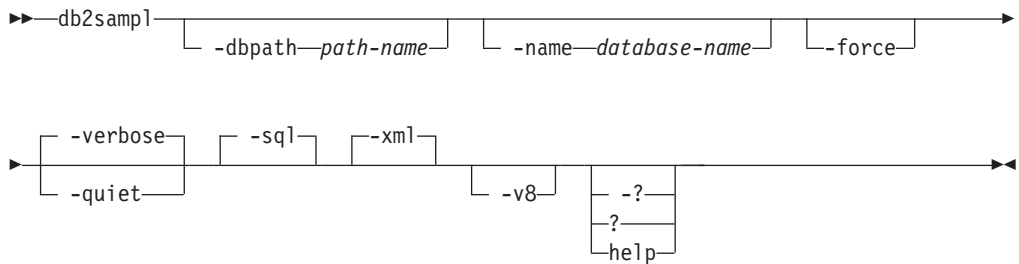
다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*

### 필요한 연결

없음

### 명령 구문



### 명령 매개변수

#### **-dbpath** *path-name*

데이터베이스를 작성할 경로를 지정합니다. Windows 운영 체제에서, 데이터베이스를 작성할 드라이브의 이름을 지정합니다. *path-name*의 최대 길이는 175 자입니다. 디폴트로, *path-name*은 데이터베이스 관리 프로그램 구성 파일에 지정된 디폴트 경로입니다(`dftdbpath` 매개변수).

#### **-name** *database-name*

샘플 데이터베이스의 이름을 지정합니다. 데이터베이스 이름은 데이터베이스에 대한 이름 지정 규칙을 준수해야 합니다. 디폴트로, *database-name*은 SAMPLE입니다.

**-force** 샘플 데이터베이스에 지정된 것과 같은 이름으로 인스턴스에서 기존 데이터베이스를 강제로 삭제하고 재작성합니다.

**-verbose**

상태 메시지를 표준 출력에 인쇄합니다.

**-quiet** 표준 출력에 상태 메시지를 인쇄하는 것을 제외합니다.

**-sql** 테이블, 트리거, 함수, 프로시저를 작성하고 테이블을 데이터로 채웁니다.

**-xml** 데이터 유형 XML의 컬럼으로 테이블을 작성하고, XML 컬럼에서 인덱스를 작성하며, XML 스키마를 등록하고 이 테이블을 XML 문서 값을 포함하는 데이터로 채웁니다.

이 옵션은 XML이 지원되는 경우에만 지원됩니다. XML이 지원되지 않으면 이 옵션은 무시됩니다.

**-v8** DB2 Universal Database 버전 8 샘플 데이터베이스, 데이터베이스 오브젝트 및 데이터를 작성합니다. 버전 8 샘플 데이터베이스는 데이터베이스 관리 프로그램 구성 파일에 지정된 디폴트 경로(dftdbpath 매개변수)에서 작성되는 SAMPLE이라고 하는 비유니코드 데이터베이스입니다.

**-? | ? | help**

db2sampl 명령 구문 도움말을 리턴합니다.

**db2sampl**의 디폴트 동작

db2sampl 명령이 선택적 인수 없이 발행되는 경우, 환경이 파티션되는지 여부에 따라 달리 작동합니다.

파티션되지 않는 데이터베이스 환경에서:

- 디폴트 데이터베이스 경로에 유니코드(UTF-8) 코드 세트 및 UCA400\_NO 조합과 C(POSIX) 지역이 있는 SAMPLE 데이터베이스를 작성합니다.
- 테이블, 인덱스, 제한조건, 트리거, 함수, 프로시저, 다차원의 데이터베이스 테이블 및 구체화된 쿼리 테이블을 포함하는 관계형 데이터베이스 오브젝트를 작성합니다.
- 관계형 테이블을 데이터로 채웁니다.
- XML 데이터 유형 컬럼으로 테이블을 작성합니다.
- XML 데이터를 통한 인덱스를 작성합니다.
- XML 스키마 문서를 포함하는 XML 스키마 저장소를 작성합니다.
- 모든 데이터베이스 오브젝트 이름은 CURRENT\_SCHEMA 특수 레지스터의 값으로 규정됩니다.

파티션된 데이터베이스 환경에서:

- 디폴트 데이터베이스 경로에 유니코드(UTF-8) 코드 세트 및 UCA400\_NO 조합과 C(POSIX) 지역이 있는 SAMPLE 데이터베이스를 작성합니다.
- 테이블, 인덱스, 제한조건, 트리거, 함수, 프로시저, 다차원의 데이터베이스 테이블 및 구체화된 쿼리 테이블을 포함하는 관계형 데이터베이스 오브젝트를 작성합니다.

- 테이블을 데이터로 채웁니다.
- 모든 데이터베이스 오브젝트 이름은 CURRENT\_SCHEMA 특수 레지스터의 값으로 규정됩니다.

## 사용 시 참고사항

- db2sampl 명령은 DB2 데이터베이스 서버가 설치된 컴퓨터에서만 발행할 수 있습니다. 리모트 IBM Data Server Client에서는 발행할 수 없습니다.
- 샘플 데이터베이스는 데이터베이스 관리 프로그램 구성 매개변수 authentication 에 지정된 인스턴스 인증 유형으로 작성됩니다.

## 예:

- 디폴트 특성을 가지고 있는 샘플 데이터베이스를 작성하려면 다음을 발행하십시오.  
db2sampl
- Windows 운영 체제에서, 디폴트 스키마에 SQL 데이터베이스 오브젝트만 포함하는 E: 드라이브에서 *mysample* 샘플 데이터베이스를 작성하고 상태 메시지를 보려면 다음을 발행하십시오.  
db2sampl -dbpath E -name mysample -sql -force -verbose
- DB2 버전 8 샘플 데이터베이스를 작성하려면 다음을 발행하십시오.  
db2sampl -v8



---

## 제 250 장 db2schex - Active Directory 스키마 확장 명령

Windows Server 2003 이상에서 LDAP(Lightweight Directory Access Protocol) 디렉토리 서버 기능을 사용하려는 경우 새 DB2 오브젝트 클래스와 속성 정의를 포함하도록 Microsoft Active Directory 스키마를 확장해야 합니다. 새 오브젝트 클래스와 속성을 디렉토리 스키마에 추가하는 프로세스를 스키마 확장이라고 합니다.

이미 DB2 V9.5 버전이 설치된 머신에서 db2schex DB2 스키마 확장 명령을 실행하여 Active Directory에서 스키마를 확장해야 합니다. 또는 DB2 V9.1 또는 DB2 V8.2 설치 이미지/CD에서 db2schex.exe를 실행합니다.

db2schex 명령은 제품 DVD에 포함됩니다. DVD에서 이 명령의 위치 경로는 x:\db2\windows\utilities입니다. 여기서 x:는 DVD 드라이브를 지정합니다.

### 권한 부여

Active Directory 스키마를 갱신하려면 스키마 관리자 그룹의 구성원이거나 스키마 갱신 권한을 위임받아야 합니다.

### 필수 연결

대상 도메인의 Windows 도메인 제어기 서버에 액세스합니다.

### 명령 구문

```
db2schex [-b bindDN] [-w password] [-k] [-u] [-x filename]
```

### 명령 매개변수

#### -b bindDN

사용자 식별 이름을 지정합니다.

#### -w password

바인드 암호를 지정합니다.

#### -k

오류를 무시하고 설치 제거를 계속 하도록 강제 실행합니다.

#### -u

스키마를 설치 제거합니다.

#### -x filename

이 매개변수를 지정하면 유틸리티가 수행한 Active Directory 스키마에 대한 변경사항이 파일에 작성됩니다.

예:

DB2 스키마를 설치하려면 다음 명령을 실행하십시오.

```
db2schex
```

DB2 스키마를 설치하고 바인드 DN과 암호를 지정하려면 다음 명령을 실행하십시오.

```
db2schex -b "cn=A_Name,dc=toronto1,dc=ibm,dc=com" -w password
```

또는

```
db2schex -b Administrator -w password
```

DB2 스키마를 설치 제거하려면 다음 명령을 실행하십시오.

```
db2schex -u
```

DB2 스키마를 설치 제거하고 오류를 무시하려면 다음 명령을 실행하십시오.

```
db2schex -u -k
```

## 사용 시 참고사항

*bindDN*과 *password*가 지정되지 않으면 *db2schex*가 현재 로그인한 사용자로서 바인드됩니다.

*bindDN* 매개변수는 Windows 사용자 이름으로 지정할 수 있습니다.

DB2 스키마 확장 명령은 다음 태스크를 수행합니다.

- 스키마 마스터인 서버 발견
- 스키마 마스터인 도메인 제어기에 바인드
- 사용자에게 스키마에 클래스와 속성을 추가할 수 있는 권한이 있는지 확인
- 스키마 마스터가 쓰기 가능한지(즉, 레지스트리의 안전 인터록이 제거되는지) 확인
- 모든 새 속성 작성
- 모든 새 오브젝트 클래스 작성
- 오류 발견 및 오류 발생 시 프로그램이 스키마의 변경사항 롤백.



## 제 251 장 db2set - DB2 프로파일 레지스트리

DB2 프로파일 변수를 표시, 설정 또는 제거합니다. DB2 Administration Server를 통해 DB2 프로파일 레지스트리에 저장된 DB2 환경 변수의 로컬 및 리모트 관리를 지원하는 외부 환경 레지스트리 명령.

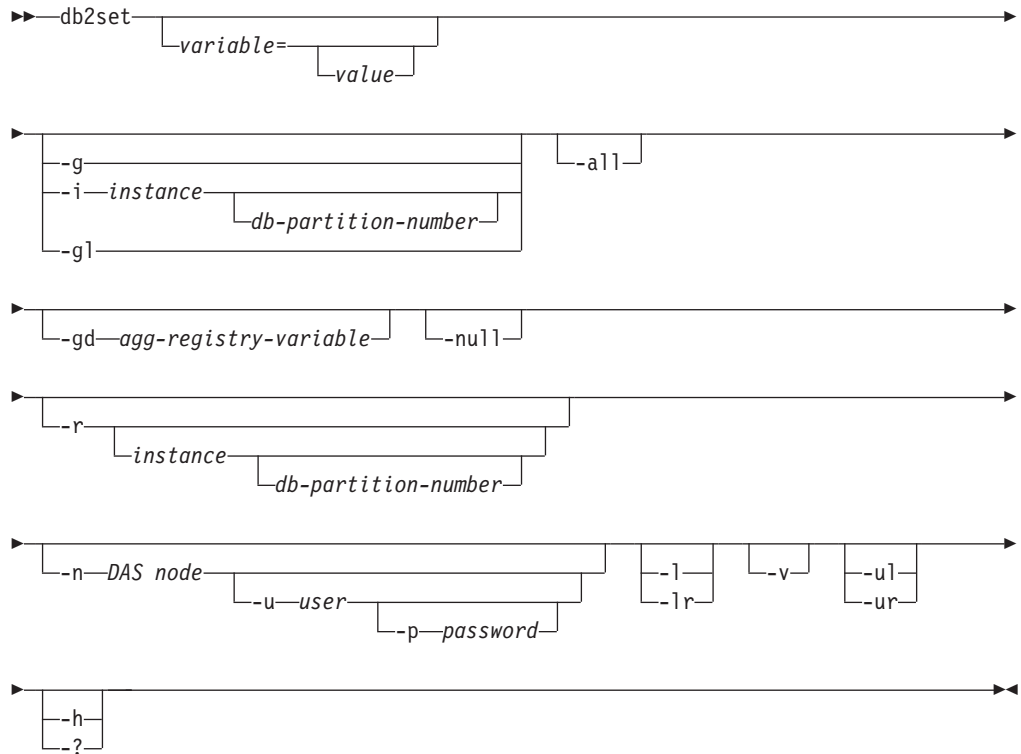
### 권한 부여

*sysadm*

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

*variable=*

지정된 변수의 값을 표시합니다.

값 지정된 변수를 입력된 값으로 설정합니다. 변수를 삭제하려면 등호(=) 뒤에 지정된 변수에 대한 값을 지정하지 마십시오. 설정 변경은 인스턴스가 재시작된 후에 적용됩니다.

**-g** 특정 DB2 사본에 관련된 모든 인스턴스에 대한 전역 프로파일 레지스트리 변수에 액세스합니다.

**-i** 현재 또는 디폴트 대신 사용할 인스턴스 프로파일을 지정합니다.

*db-partition-number*

db2nodes.cfg 파일에 나열된 숫자를 지정합니다.

**-gl** LDAP에 저장된 전역 프로파일 변수에 액세스합니다. 이 옵션은 레지스트리 변수 DB2\_ENABLE\_LDAP가 YES로 설정된 경우에만 적용됩니다.

**-all** 다음에 정의된 로컬 환경 변수의 모든 어커런스를 표시합니다.

- 환경 - [e]로 표시됨
- 노드 레벨 레지스트리 - [n]으로 표시됨
- 인스턴스 레벨 레지스트리 - [i]로 표시됨
- 전역 레벨 레지스트리 - [g]로 표시됨.

**-gd** *agg-registry-variable*

집계 레지스트리 변수의 그룹 정의를 표시합니다. 추가 정보는 *Data Servers, Databases, and Database Objects Guide*의 『집계 레지스트리 변수』를 참조하십시오.

**-null** 지정된 레지스트리 레벨의 변수 값을 NULL로 설정합니다. 그러면 검색 순서로 정의되는 것처럼 다음 레지스트리 레벨에서 값을 찾을 필요가 없습니다.

**-r** *instance*

주어진 인스턴스에 대한 프로파일 레지스트리를 재설정합니다. 인스턴스를 지정하지 않고 인스턴스 접속이 존재하는 경우 현재 인스턴스의 프로파일을 재설정합니다. 인스턴스를 지정하지 않고 접속이 존재하지 않는 경우 DB2INSTANCE 환경 변수로 지정되는 인스턴스에 대한 프로파일을 재설정합니다.

**-n** *DAS node*

리모트 DB2 Administration Server 노드 이름을 지정합니다.

**-u** *user*

Administration Server 접속에 사용할 사용자 ID를 지정합니다.

**-p** *password*

Administration Server 접속에 사용할 암호를 지정합니다.

**-l** 현재 DB2 제품 설치의 모든 인스턴스 프로파일을 나열합니다.

**-lr** 지원되는 모든 레지스트리 변수를 나열합니다.

**-v** 상세 모드를 지정합니다.

- ul 사용자 프로파일 변수에 액세스합니다. 이 매개변수는 Windows 운영 체제에서만 지원됩니다.
- ur 사용자 프로파일 변수를 새로 고칩니다. 이 매개변수는 Windows 운영 체제에서만 지원됩니다.
- h | -? 도움말 정보를 표시합니다. 이 옵션을 지정하면 다른 모든 옵션은 무시되며 도움말 정보만 표시됩니다.

## 예:

- 특정 설치와 관련된 모든 정의된 프로파일(DB2 인스턴스)을 표시하십시오.  
db2set -l
- 모든 지원되는 레지스트리 변수를 표시하십시오.  
db2set -lr
- 특정 설치에 관련된 모든 인스턴스가 볼 수 있는 모든 정의된 전역 변수를 표시하십시오.  
db2set -g
- 현재 인스턴스에 대해 정의된 모든 변수를 표시하십시오.  
db2set
- 현재 인스턴스에 대해 정의된 모든 값을 표시하십시오.  
db2set -all
- 현재 인스턴스에 대한 DB2COMM의 모든 정의된 값을 표시하십시오.  
db2set -all DB2COMM
- 노드 3의 INST 인스턴스에 대해 정의된 모든 변수를 재설정하십시오.  
db2set -r -i INST 3
- 사용자 ID MYID와 암호 MYPASSWD를 사용하여 DAS 노드를 통해 리모트 인스턴스 RMTINST의 DB2CHKPTR 변수를 설정 해제하십시오.  
db2set -i RMTINST -n RMTDAS -u MYID -p MYPASSWD DB2CHKPTR=
- 특정 설치와 관련된 모든 인스턴스에 대해 전역으로 TCP/IP가 되도록 DB2COMM 변수를 설정하십시오.  
db2set -g DB2COMM=TCPIP
- MYINST 인스턴스에 대해 TCP/IP가 되도록 DB2COMM 변수를 설정하십시오.  
db2set -i MYINST DB2COMM=TCPIP
- 주어진 인스턴스 레벨에서 DB2COMM 변수를 널(NULL)로 설정하십시오.  
db2set -null DB2COMM

## 사용 시 참고사항

변수 이름을 지정하지 않는 경우 모든 정의된 변수의 값이 표시됩니다. 변수 이름이 지정되는 경우 해당 변수의 값만 표시됩니다. 변수의 모든 정의된 값을 표시하려면 *variable -all*을 지정하십시오. 모든 레지스트리의 모든 정의된 변수를 표시하려면 *-all*을 지정하십시오.

변수의 값을 수정하려면 *variable=*를 지정한 후 새 값을 지정하십시오. 변수 값을 NULL로 설정하려면 *variable -null*을 지정하십시오. 설정 변경은 인스턴스가 재시작된 후에 적용됩니다.

변수를 삭제하려면 *variable=*를 지정한 후 값을 지정하지 마십시오.

명령이 DB2의 루트가 아닌 설치에 대해 동일하게 작동하지만, DAS 노드 이름을 지정하는 것처럼 모든 매개변수가 사용 가능하지는 않습니다.

---

## 제 252 장 db2setup - DB2 설치

DB2 제품을 설치합니다. 이 명령은 Linux 및 UNIX 시스템에서만 사용할 수 있습니다. Windows 운영 체제용 명령은 setup입니다.

이 유틸리티는 DB2 설치 미디어에 있습니다. 이 유틸리티는 DB2 설정 마법사를 시작하여 설치를 정의하고 DB2 제품을 설치합니다. -r 옵션으로 호출되는 경우 추가 입력 없이 응답 파일에서 설치 구성 정보를 가져와 설치를 수행합니다.

### 권한 부여

Linux 및 UNIX 시스템의 경우 루트 설치 시 루트 권한이 필요합니다. 비루트 설치의 경우 비루트 설치를 소유하는 사용자 ID로 로그인해야 합니다.

### 명령 구문

```
db2setup [-i language] [-l log_file] [-t trace_file]
         [-r response_file] [-f nobackup] [--? -h]
```

### 명령 매개변수

#### -i language

설치를 수행하는 언어의 2자 언어 코드.

#### -l log\_file

지정된 파일 이름에 로그를 작성합니다. 루트 설치의 경우 디폴트 로그 파일은 /tmp/db2setup.log입니다. 비루트 설치의 경우 디폴트 로그 파일은 /tmp/db2setup\_userID.log입니다. 여기서 userID는 비루트 설치를 소유하는 사용자 ID를 나타냅니다. IBM Tivoli SA MP(System Automation for Multiplatforms)를 db2setup을 사용하여 설치하는 경우 SA MP의 설치 로그 파일은 DB2 로그 파일과 동일한 디렉토리에 있습니다.

#### -t trace\_file

설치 추적 정보로 파일을 생성합니다.

#### -r response\_file

사용할 응답 파일의 전체 경로 및 파일 이름.

#### -f nobackup

루트 서버가 아닌 업그레이드에만 적용됩니다. 구성요소가 갱신될 때 db2setup이 설치 파일을 백업하지 않도록 강제 실행합니다. 파일을 백업하지 않는 것으

로 선택하면 설치 디렉토리의 공간 요구사항이 감소합니다. 그러나 파일을 백업하지 않는 것으로 선택하는 경우 오류 발생 시 DB2 설치 프로그램이 롤백 작업을 수행할 수 없음을 의미하기도 합니다. 이러한 경우 파일을 수동으로 제거하고 제품을 재설치해야 합니다.

**-? | -h**

사용법 정보를 생성합니다.

## 사용 시 참고사항

올바른 권한이 있는 ID로 로그인하거나 su와 "-" 플래그를 함께 사용하여(su -) 올바른 권한이 있는 ID로 로그인한 것처럼 프로세스 환경을 설정해야 합니다. 프로세스 환경이 올바른 권한이 있는 ID로 설정되지 않아도 오류 없이 설치 프로세스가 완료되지만 DB2 사본을 실행하면 오류가 발생합니다.

## 제 253 장 db2sqljbind - SQLJ 프로파일 바인더

db2sqljbind는 이전에 db2sqljcustomize 명령을 사용하여 사용자 정의한 직렬화 프로파일에서 사용하도록 DB2 패키지를 바인드합니다.

IBM Data Server Driver for JDBC and SQLJ를 사용하여 실행되는 응용프로그램에는 패키지가 필요하지만 플랜은 필요하지 않습니다. db2sqljcustomize -automaticbind 옵션이 YES로 지정되거나 디폴트값이 YES인 경우, db2sqljcustomize는 -url 매개변수에서 지정한 데이터 소스에 패키지를 바인드합니다. 그러나 -automaticbind가 NO인 경우, db2sqljcustomize 실행 시 바인드에 실패하는 경우 또는 여러 위치에서 동일한 직렬화 프로파일에서 사용할 동일한 패키지를 작성하려는 경우에는 db2sqljbind 명령을 사용하여 패키지를 바인드할 수 있습니다.

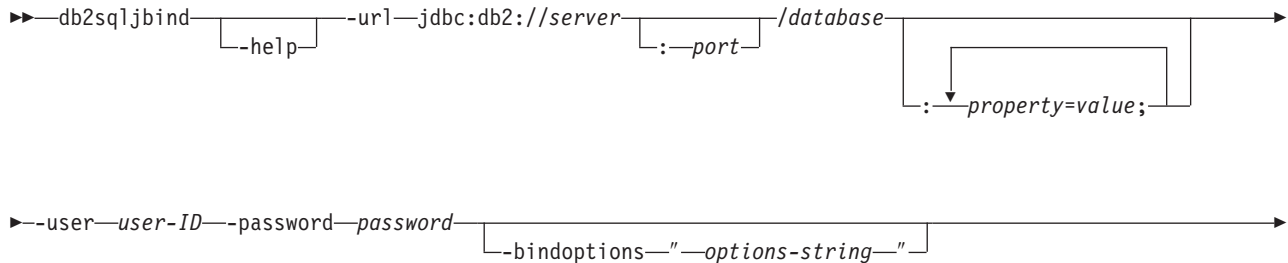
### 권한 부여

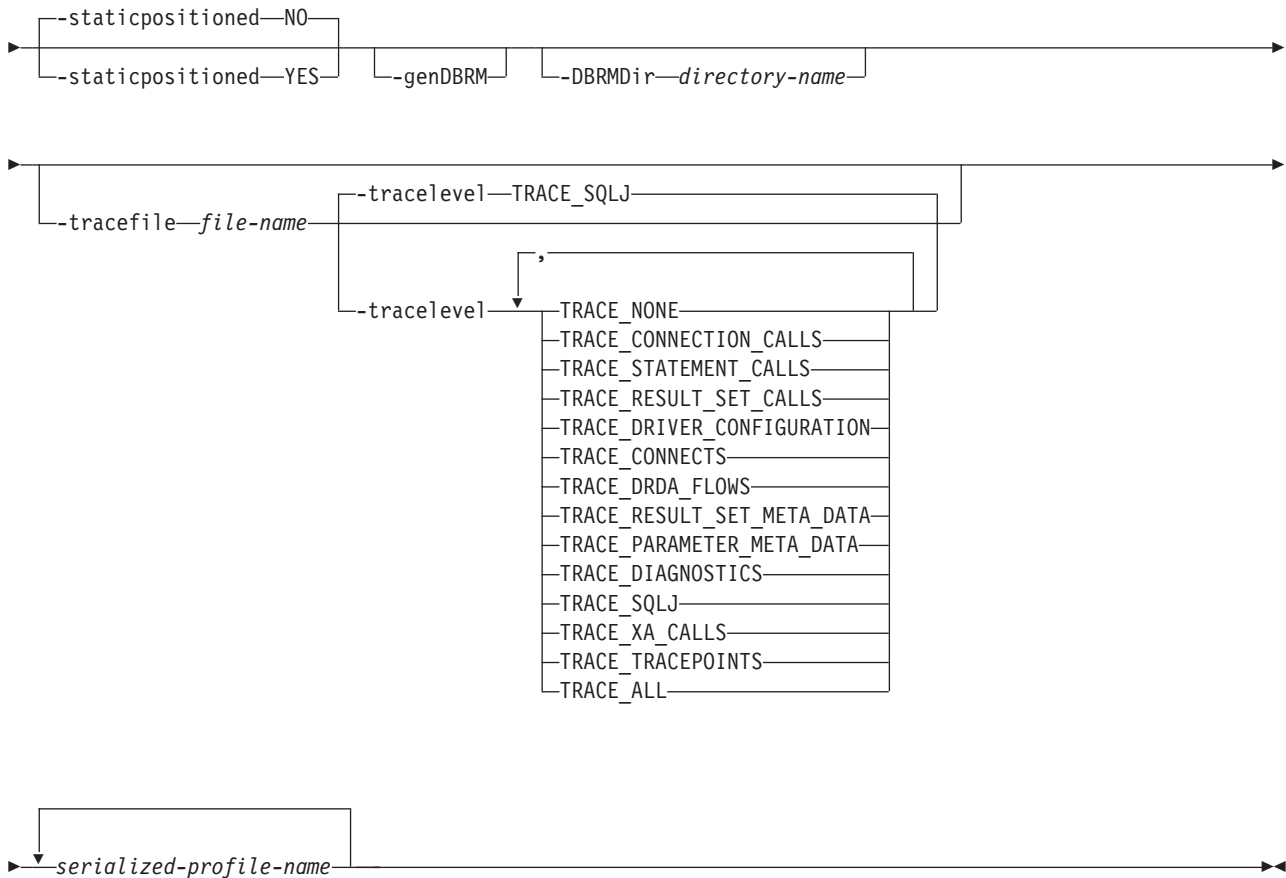
프로세스의 특권 세트에는 다음 권한 중 하나가 포함되어야 합니다.

- DBADM 권한
- 패키지가 없는 경우 BINDADD 특권 및 다음 특권 중 하나가 포함되어야 합니다.
  - CREATEIN 특권
  - 패키지의 스키마 이름이 존재하지 않는 경우 데이터베이스에 대한 IMPLICIT\_SCHEMA 권한
- 패키지가 있는 경우 다음 특권 중 하나가 포함되어야 합니다.
  - 스키마에서 ALTERIN 특권
  - 패키지에 대한 BIND 특권

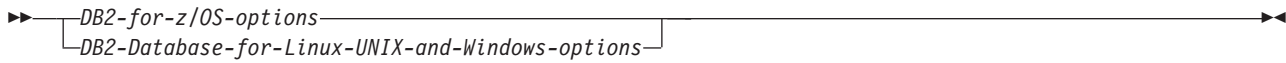
응용프로그램에서 정적 SQL을 컴파일하는 데 필요한 모든 특권도 사용자에게 있어야 합니다. 그룹에 부여된 특권은 정적 명령문의 권한 부여 검사에 사용되지 않습니다.

### 명령 구문

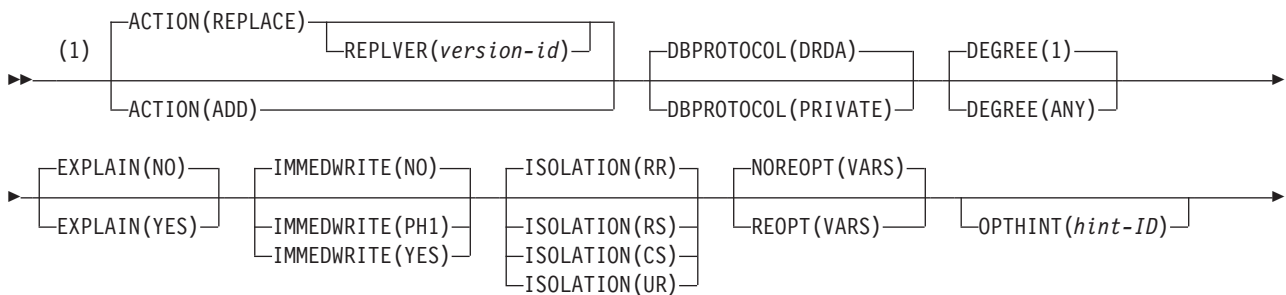




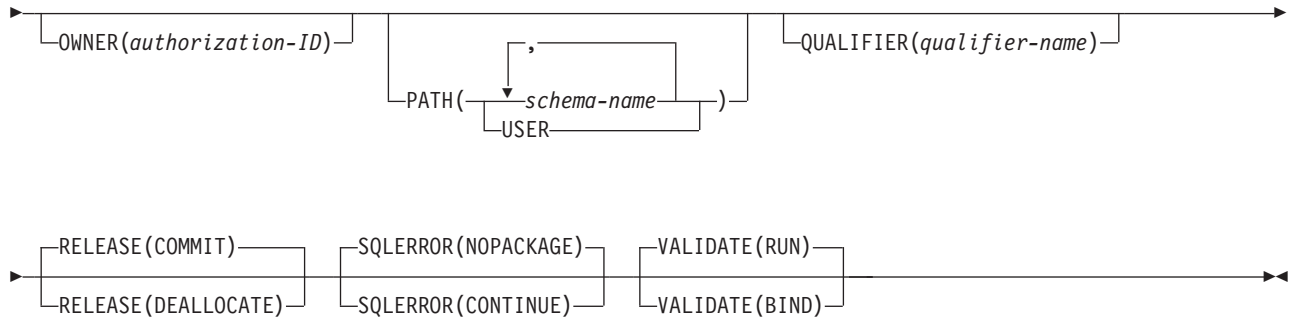
**options-string:**



**DB2 옵션:**



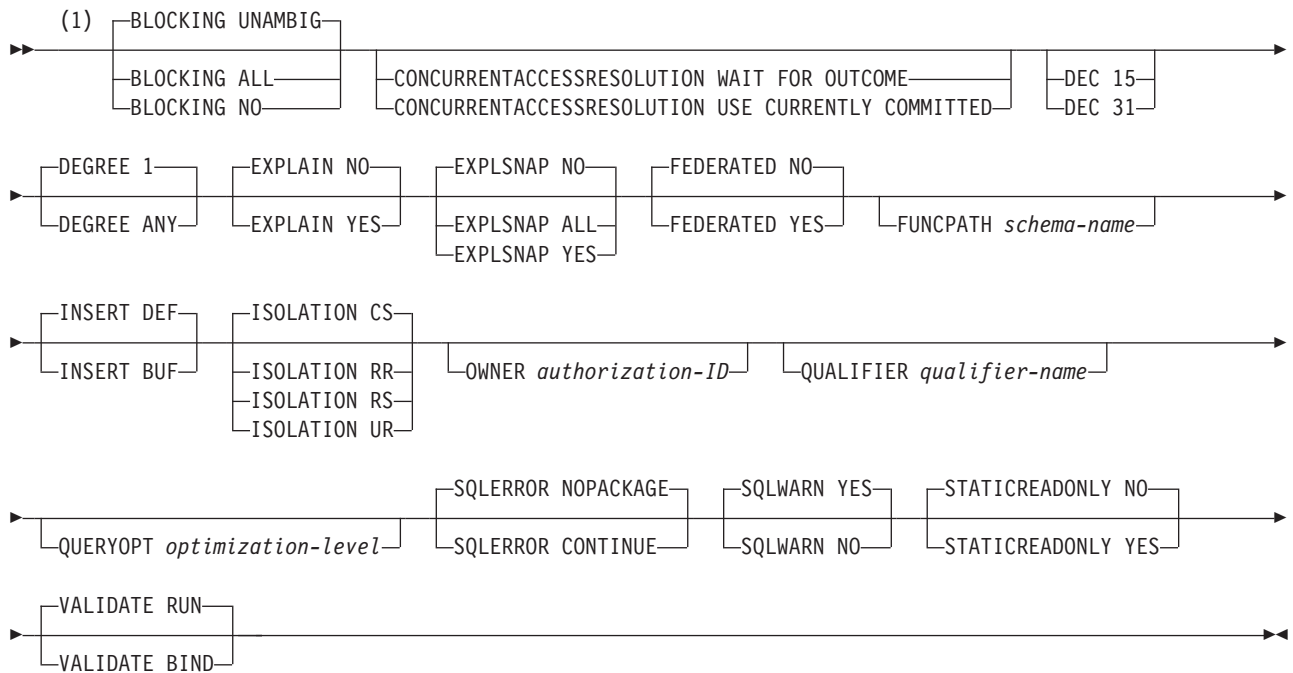




주:

- 1 순서에 관계 없이 다음 옵션을 지정할 수 있습니다.

### Linux, UNIX 및 Windows용 DB2 데이터베이스 옵션



주:

- 1 순서에 관계 없이 다음 옵션을 지정할 수 있습니다.

### 명령 매개변수

#### -help

db2sqljbind가 지원하는 각 옵션에 대해 설명하도록 지정합니다. -help로 지정되는 기타 옵션은 무시됩니다.

#### -url

프로파일을 사용자 정의할 데이터 소스의 URL을 지정합니다. -automaticbind 또

는 `-onlinecheck` 옵션이 YES로 지정되거나 디폴트가 YES인 경우 이 URL이 나타내는 데이터 소스에 연결됩니다. `-url` 값의 변수 파트는 다음과 같습니다.

#### 서버

데이터베이스 서버가 상주하는 운영 체제의 도메인 이름 또는 IP 주소입니다.

#### port

데이터베이스 서버에 지정된 TCP/IP 서버 포트 번호입니다. 디폴트값은 446입니다.

#### database

프로파일을 사용자 정의할 데이터베이스 서버의 이름입니다.

DB2 서버에 연결되는 경우 `database`는 설치 중에 정의된 DB2 위치 이름입니다. 이 값의 모든 문자는 대문자여야 합니다. 서버에서 다음 SQL문을 실행하여 위치 이름을 판별할 수 있습니다.

```
SELECT CURRENT SERVER FROM SYSIBM.SYSDUMMY1;
```

Linux, UNIX 및 Windows용 DB2 데이터베이스 서버에 연결되는 경우 `database`는 설치 중에 정의된 데이터베이스 이름입니다.

IBM Cloudscape® 서버에 연결되는 경우 `database`는 데이터베이스가 포함된 파일의 완전한 이름입니다. 이 이름을 큰따옴표(")로 묶어야 합니다. 예를 들어, 다음과 같습니다.

```
"c:/databases/testdb"
```

```
property=value;
```

JDBC 연결에 대한 등록 정보입니다.

#### **-user** *user-ID*

패키지 바인딩을 위해 데이터 소스에 연결하는 데 사용할 사용자 ID를 지정합니다.

#### **-password** *password*

패키지 바인딩을 위해 데이터 소스에 연결하는 데 사용할 암호를 지정합니다.

#### **-bindoptions** *options-string*

스페이스로 구분되는 옵션 목록을 지정합니다. 이들 옵션에는 동일한 이름을 가진 DB2 프리컴파일 및 바인드 옵션과 같은 기능이 있습니다. DB2 시스템에서 실행하기 위해 프로그램을 준비 중인 경우 DB2 옵션을 지정하십시오. Linux, UNIX 및 Windows용 DB2 데이터베이스 시스템에서 실행하기 위해 프로그램을 준비 중인 경우 Linux, UNIX 및 Windows용 DB2 데이터베이스 옵션을 지정하십시오.

#### **바인드 옵션에 대한 참고사항:**

- 다음 조건이 참인 경우에만 `VERSION`을 지정하십시오.
  - Linux, UNIX 및 Windows용 DB2 데이터베이스 시스템에서 패키지를 바인드 중인 경우 시스템 버전은 8 이상이어야 합니다.

- 연관된 패키지와 새 VERSION 값을 바인드하기 전에 먼저 프로그램에서 변환기를 재실행해야 합니다.

- **STATICREADONLY**의 값은 **STATICREADONLY**를 지원하는 서버에서는 YES이고 기타 서버에서는 NO입니다. **STATICREADONLY YES**로 지정하면 DB2에서 앰비규어스 커서를 읽기 전용 커서인 것처럼 처리합니다. 반복자(iterator) 선언 오류를 해결하려면 **STATICREADONLY**를 명시적으로 NO로 지정하거나 반복자(iterator)를 명확하게 선언해야 합니다. 예를 들어, 반복자(iterator)를 명확하게 갱신할 수 있게 하려면 `sqlj.runtime.ForUpdate`를 구현하도록 반복자(iterator)를 선언하십시오. 반복자(iterator)를 읽기 전용으로 설정하려면 반복자(iterator)를 사용하는 **SELECT**문에 **FOR READ ONLY** 절을 포함시키십시오.

**중요:** 패키지를 바인드 중인 데이터 소스에 적합한 프로그램 준비 옵션만 지정하십시오. IBM Data Server Driver for JDBC and SQLJ의 일부 값 및 디폴트값은 DB2의 값 및 디폴트값과 다릅니다.

#### **-staticpositioned NO|YES**

반복자(iterator)를 사용하는 위치 지정된 **UPDATE**문과 동일한 소스 파일에 반복자가 선언된 경우, 위치 지정된 **UPDATE**가 정적으로 바인드된 명령문으로 실행되는지 여부를 지정합니다. 디폴트값은 NO입니다. NO는 위치 지정된 **UPDATE**가 동적으로 준비된 명령문으로 실행됨을 의미합니다. 이 값은 직렬화 프로파일에 적용되는 이전 `db2sqljcustomize` 호출의 **-staticpositioned** 값과 동일해야 합니다.

#### **-genDBRM**

`db2sqljbind`가 직렬화 프로파일에서 **DBRM(Database Request Module)**을 생성하고 `db2sqljbind`가 리모트 바인드 조작을 수행하지 않도록 지정합니다.

**-genDBRM**은 DB2 데이터베이스 서버에서 실행할 프로그램에만 적용됩니다.

#### **-DBRMDir directory-name**

**-genDBRM**이 지정된 경우, **-DBRMDir**은 `db2sqljbind`가 생성된 **DBRM** 파일을 저장하는 로컬 디렉토리를 지정합니다. 디폴트는 현재 디렉토리입니다.

**-DBRMdir**은 DB2 데이터베이스 서버에서 실행할 프로그램에만 적용됩니다.

#### **-tracefile file-name**

추적을 사용 가능하게 하고 추적 정보에 관한 출력 파일을 식별합니다. IBM Software Support의 지시문에 따라서만 이 옵션을 지정해야 합니다.

#### **-tracelevel**

**-tracefile**이 지정된 경우, `db2sqljcustomize`를 실행하는 동안 추적할 내용을 표시합니다. 디폴트값은 **TRACE\_SQLJ**입니다. IBM Software Support의 지시문에 따라서만 이 옵션을 지정해야 합니다.

#### **serialized-profile-name**

패키지가 바인드되는 하나 이상의 직렬화 프로파일 이름을 지정합니다. 직렬화된 프로파일 이름의 양식은 다음과 같습니다.

*program-name\_SJProfileIDNumber.ser*

*program-name*은 SQLJ 소스 프로그램의 이름으로 .sqlj 확장자를 사용하지 않습니다. *n*은 0과 *m-1* 사이의 정수로 여기서 *m*은 SQLJ 변환기가 SQLJ 소스 프로그램에서 생성한 직렬화된 프로파일의 수입니다.

하나 이상의 직렬화 프로파일 이름을 지정하여 여러 직렬화 프로파일에서 단일 DB2 패키지를 바인드하는 경우, db2sqljcustomize 실행 시 동일한 직렬화 프로파일 이름이 동일한 순서로 지정되어 있어야 합니다.

### 예:

```
db2sqljbind -user richler -password mordecai  
-url jdbc:db2://server:50000/sample -bindoptions "EXPLAIN YES"  
pgmname_SJProfile0.ser
```

## 사용법 참고사항

*db2sqljbind*를 통해 생성된 패키지 이름: db2sqljbind를 사용하여 작성한 패키지 이름은 db2sqljcustomize 실행 시 *-rootpkgname* 또는 *-singlepkgname* 매개변수를 사용하여 지정한 이름입니다. *-rootpkgname* 또는 *-singlepkgname*을 지정하지 않은 경우, 프로파일 이름의 첫 7바이트가 패키지 이름이 되고 분리 레벨 문자가 추가됩니다.

*db2sqljbind*의 **DYNAMICRULES** 값: DYNAMICRULES 바인드 옵션은 DB2 패키지의 런타임 속성 번호를 판별합니다. 이러한 속성 중 2개는 권한을 확인하는 데 사용되는 권한 부여 ID 및 규정되지 않은 오브젝트에 사용되는 규정자입니다. SQLJ 프로그램에서 동적으로 실행되는 위치 지정된 UPDATE 및 DELETE문에 올바른 권한이 부여되도록 하기 위해 db2sqljbind가 항상 DB2 패키지와 DYNAMICRULES(BIND) 옵션을 바인드합니다. 이 옵션을 수정할 수 없습니다. DYNAMICRULES(BIND) 옵션을 지정하면 SET CURRENT SQLID문 및 SET CURRENT SCHEMA문이 SQLJ 프로그램에 아무 영향도 주지 않습니다. 이들 명령문은 BIND가 아니라 DYNAMICRULES 값을 사용하여 바인드된 동적 명령문에만 영향을 주기 때문입니다.

DYNAMICRULES(BIND)를 지정하면 동적 SQL문의 규정되지 않은 테이블, 뷰, 인덱스 및 별명 이름이 바인드 옵션 QUALIFIER 값을 통해 내재적으로 규정됩니다. QUALIFIER를 지정하지 않은 경우 DB2에서는 패키지 소유자의 권한 부여 ID를 내재된 규정자로 사용합니다. 이러한 동작이 사용자 프로그램에 적합하지 않은 경우 다음 기술 중 하나를 사용하여 올바른 규정자를 설정할 수 있습니다.

- 위치 지정된 UPDATE 및 DELETE문을 정적으로 실행되도록 강제 실행합니다. 위치 지정된 UPDATE 또는 DELETE문의 커서(반복자)가 위치 지정된 UPDATE 또는 DELETE문과 동일한 패키지에 있는 경우에 db2sqljcustomize 또는 db2sqljbind의 *-staticpositioned YES* 옵션을 사용하여 이를 수행할 수 있습니다.
- 위치 지정된 UPDATE 및 DELETE문에서 DB2 테이블 이름을 완전히 규정합니다.

## 제 254 장 db2sqljcustomize - SQLJ 프로파일 사용자 정의 프로그램

db2sqljcustomize는 Embedded SQL문이 포함된 SQLJ 프로파일을 처리합니다.

디폴트로, db2sqljcustomize는 분리 레벨마다 하나씩 4개의 DB2 패키지를 생성합니다. db2sqljcustomize는 런타임 시 사용하기 위해 DB2에 맞는 정보로 프로파일 기능을 보강합니다.

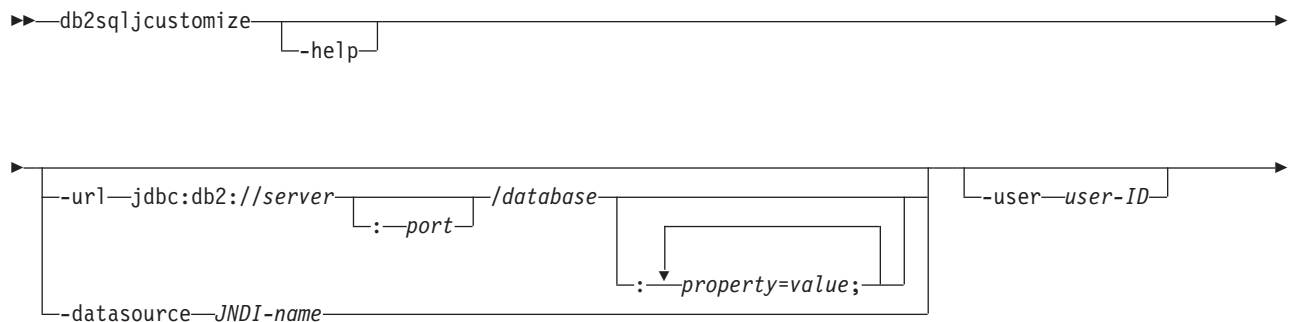
### 권한 부여

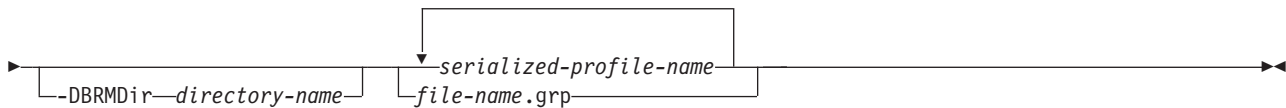
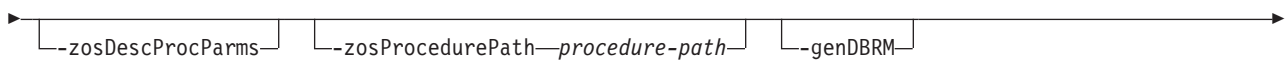
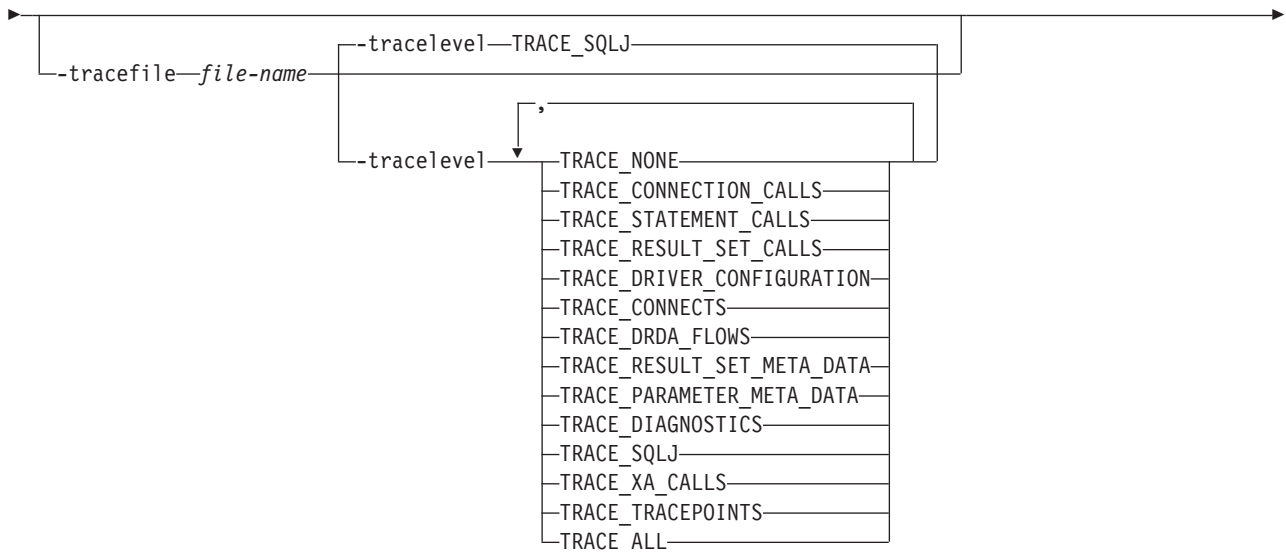
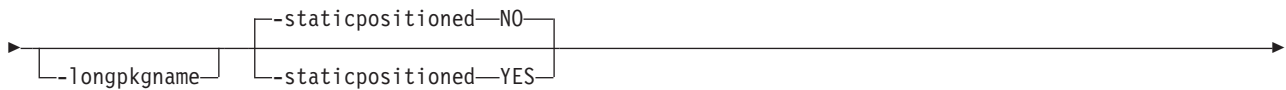
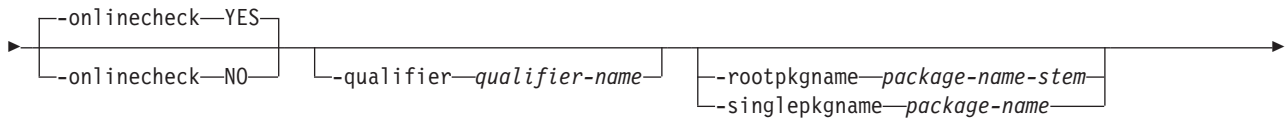
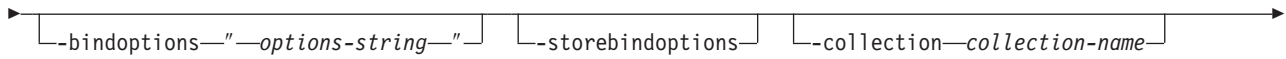
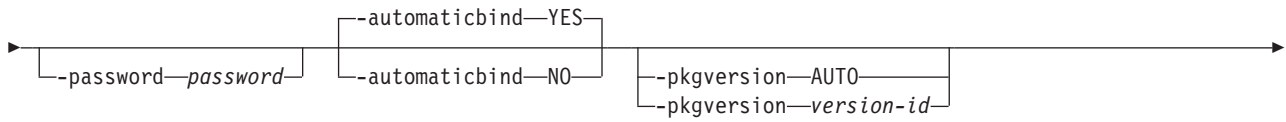
프로세스의 특권 세트에는 다음 권한 중 하나가 포함되어야 합니다.

- DBADM 권한
- 패키지가 없는 경우 BINDADD 특권 및 다음 특권 중 하나가 포함되어야 합니다.
  - CREATEIN 특권
  - 패키지의 스키마 이름이 존재하지 않는 경우 데이터베이스에 대한 IMPLICIT\_SCHEMA 권한
- 패키지가 있는 경우 다음 특권 중 하나가 포함되어야 합니다.
  - 스키마에서 ALTERIN 특권
  - 패키지에 대한 BIND 특권

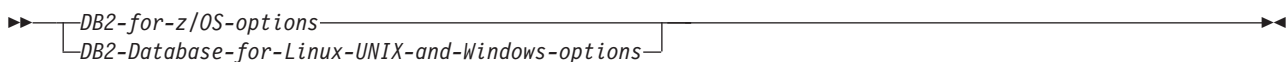
응용프로그램에서 정적 SQL을 컴파일하는 데 필요한 모든 특권도 사용자에게 있어야 합니다. 그룹에 부여된 특권은 정적 명령문의 권한 부여 검사에 사용되지 않습니다.

### 명령 구문

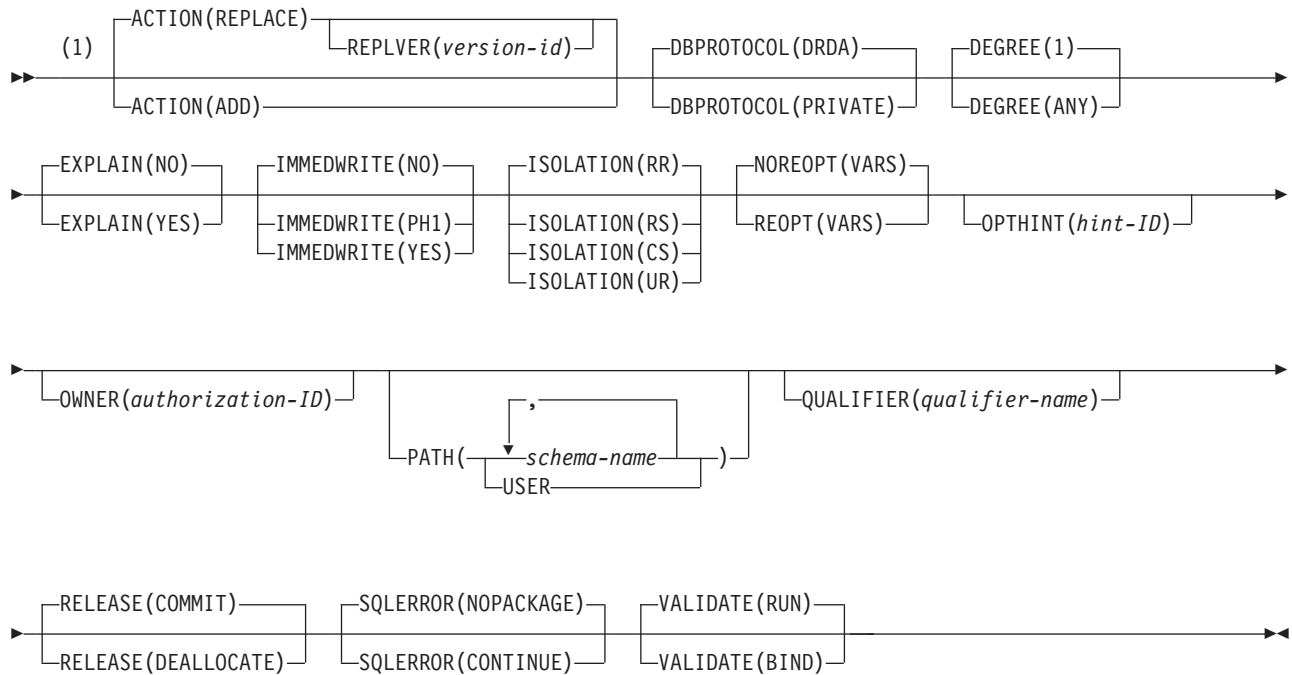




**options-string:**



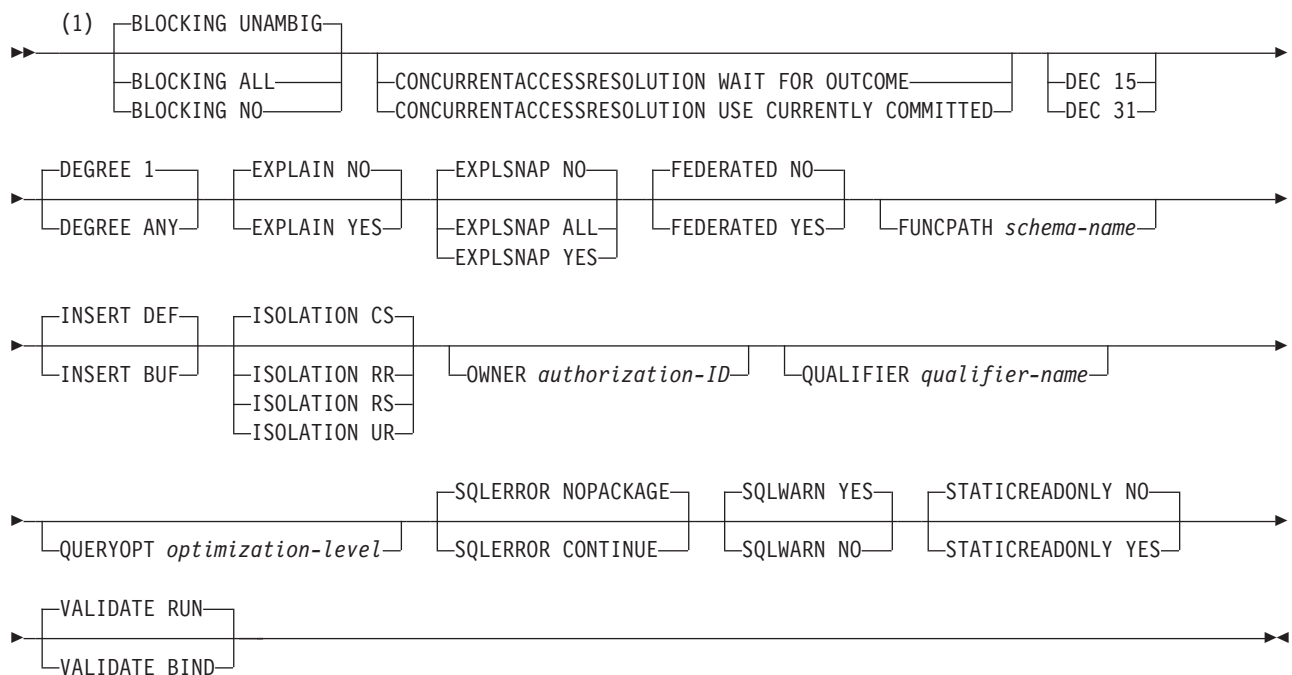
## DB2 옵션:



주:

- 1 순서에 관계 없이 다음 옵션을 지정할 수 있습니다.

## Linux, UNIX 및 Windows용 DB2 데이터베이스 옵션



주:

1 순서에 관계 없이 다음 옵션을 지정할 수 있습니다.

## 명령 매개변수

### -help

SQLJ 사용자 정의 프로그램이 지원하는 각 옵션에 대해 설명하도록 지정합니다. -help로 지정되는 기타 옵션은 무시됩니다.

### -url

프로파일을 사용자 정의할 데이터 소스의 URL을 지정합니다. -automaticbind 또는 -onlinecheck 옵션이 YES로 지정되거나 디폴트가 YES인 경우 이 URL이 나타내는 데이터 소스에 연결됩니다. -url 값의 변수 파트는 다음과 같습니다.

#### 서버

DB2 서브시스템이 상주하는 z/OS 시스템의 도메인 이름 또는 IP 주소입니다.

#### port

DB2 서브시스템에 지정된 TCP/IP 서버 포트 번호입니다. 디폴트값은 446입니다.

### -url

프로파일을 사용자 정의할 데이터 소스의 URL을 지정합니다. -automaticbind 또는 -onlinecheck 옵션이 YES로 지정되거나 디폴트가 YES인 경우 이 URL이 나타내는 데이터 소스에 연결됩니다. -url 값의 변수 파트는 다음과 같습니다.

#### 서버

데이터베이스 서버가 상주하는 운영 체제의 도메인 이름 또는 IP 주소입니다.

#### port

데이터베이스 서버에 지정된 TCP/IP 서버 포트 번호입니다. 디폴트값은 446입니다.

### database

프로파일을 사용자 정의할 데이터베이스 서버의 이름입니다.

DB2 서버에 연결되는 경우 *database*는 설치 중에 정의된 DB2 위치 이름입니다. 이 값의 모든 문자는 대문자여야 합니다. 서버에서 다음 SQL문을 실행하여 위치 이름을 판별할 수 있습니다.

```
SELECT CURRENT SERVER FROM SYSIBM.SYSDUMMY1;
```

Linux, UNIX 및 Windows용 DB2 데이터베이스 서버에 연결되는 경우 *database*는 설치 중에 정의된 데이터베이스 이름입니다.



IBM Cloudscape 서버에 연결되는 경우 *database*는 데이터베이스가 포함된 파일의 완전한 이름입니다. 이 이름을 큰따옴표(")로 묶어야 합니다. 예를 들어, 다음과 같습니다.

```
"c:/databases/testdb"
```

```
property=value;
```

JDBC 연결에 대한 등록 정보입니다.

```
property=value;
```

JDBC 연결에 대한 등록 정보입니다.

#### **-datasource** *JNDI-name*

JNDI로 등록된 DataSource 오브젝트의 논리적 이름을 지정합니다. DataSource 오브젝트는 프로파일을 사용자 정의할 데이터 소스를 나타냅니다. -automaticbind 또는 -onlinecheck 옵션이 YES로 지정되거나 디폴트로 YES인 경우 데이터 소스에 연결됩니다. -url을 지정하는 대신 -datasource를 지정할 수 있습니다. DataSource 오브젝트는 IBM Data Server Driver for JDBC and SQLJ 유형 4 연결성을 사용하는 연결을 나타내야 합니다.

#### **-user** *user-ID*

온라인 검사 또는 패키지 바인딩을 위해 데이터 소스에 연결하는 데 사용할 사용자 ID를 지정합니다. -url을 지정하는 경우에는 -user를 지정해야 합니다. -datasource를 지정하는 경우 반드시 -user를 지정해야 하며 *JNDI-name*이 나타내는 DataSource 오브젝트에는 사용자 ID가 포함되어 있지 않습니다.

#### **-password** *password*

온라인 검사 또는 패키지 바인딩을 위해 데이터 소스에 연결하는 데 사용할 암호를 지정합니다. -url을 지정하는 경우에는 -password를 지정해야 합니다. -datasource를 지정하는 경우 반드시 -password를 지정해야 하며 *JNDI-name*이 나타내는 DataSource 오브젝트에는 암호가 포함되어 있지 않습니다.

#### **-automaticbind** YES|NO

사용자 정의 프로그램이 -url 매개변수로 지정된 데이터 소스에서 DB2 패키지를 바인드하는지 여부를 지정합니다.

디폴트값은 YES입니다.

패키지 수 및 해당 패키지의 분리 레벨은 -rootpkgname과 -singlepkgname 옵션을 통해 제어됩니다.

바인드 작업을 수행하기 전에 다음 조건이 충족되어야 합니다.

- TCP/IP 및 DRDA가 목표 데이터 소스에 설치되어 있어야 합니다.
- 유효한 -url, -username 및 -password 값이 지정되어야 합니다.
- -username 값에 목표 데이터 소스에서 패키지를 바인드할 권한이 있어야 합니다.

### **-pkgversion** *AUTO|version-ID*

사용자 정의 중인 직렬화된 프로파일에 사용하기 위해 서버에서 패키지를 바인드할 때 사용할 패키지 버전을 지정합니다. db2sqljcustomize는 버전 ID를 직렬화된 프로파일 및 DB2 패키지에 저장합니다. 런타임 버전 검증은 버전 이름이 아니라 일관성 토큰에 따라 수행됩니다. 일관성 토큰을 기본으로 하는 버전 이름을 자동으로 생성하려면 -pkgversion AUTO를 지정하십시오.

디폴트값은 버전이 없는 것입니다.

### **-bindoptions** *options-string*

스페이스로 구분되는 옵션 목록을 지정합니다. 이들 옵션에는 동일한 이름을 가진 DB2 프리컴파일 및 바인드 옵션과 같은 기능이 있습니다. DB2 시스템에서 실행하기 위해 프로그램을 준비 중인 경우 DB2 옵션을 지정하십시오. Linux, UNIX 및 Windows용 DB2 데이터베이스 시스템에서 실행하기 위해 프로그램을 준비 중인 경우 Linux, UNIX 및 Windows용 DB2 데이터베이스 옵션을 지정하십시오.

#### **바인드 옵션에 대한 참고사항:**

- -singlepkgname 옵션도 지정하는 경우에만 ISOLATION을 지정하십시오.
- STATICREADONLY의 값은 STATICREADONLY를 지원하는 서버에서는 YES이고 기타 서버에서는 NO입니다. STATICREADONLY YES로 지정하면 DB2에서 앰비규어스 커서를 읽기 전용 커서인 것처럼 처리합니다. 반복자(iterator) 선언 오류를 해결하려면 STATICREADONLY를 명시적으로 NO로 지정하거나 반복자(iterator)를 명확하게 선언해야 합니다. 예를 들어, 반복자(iterator)를 명확하게 갱신할 수 있게 하려면 sqlj.runtime.ForUpdate를 구현하도록 반복자(iterator)를 선언하십시오. 반복자(iterator)를 읽기 전용으로 설정하려면 반복자(iterator)를 사용하는 SELECT문에 FOR READ ONLY 절을 포함시키십시오.

**중요:** 패키지를 바인드 중인 데이터 소스에 적합한 프로그램 준비 옵션만 지정하십시오. IBM Data Server Driver for JDBC and SQLJ의 일부 값 및 디폴트값은 DB2의 값 및 디폴트값과 다릅니다.

### **-storebindoptions**

-bindoptions 및 -staticpositioned 매개변수의 값이 직렬화된 프로파일에 저장되는지를 지정합니다. -bindoptions 또는 -staticpositioned 매개변수를 지정하지 않고 db2sqljbind를 호출하는 경우 직렬화된 프로파일에 저장된 값이 바인드 조작 중에 사용됩니다. db2sqljcustomize를 한 번 호출하는 데 직렬화된 프로파일을 여러 개 지정하면 각각의 직렬화된 프로파일에 매개변수 값이 저장됩니다. 저장된 값은 db2sqljprint 유틸리티에서 출력에 표시됩니다.

### **-collection** *collection-name*

db2sqljcustomize가 바인드하는 패키지의 규정자입니다. db2sqljcustomize는

사용자 정의된 직렬화 프로파일에 이 값을 저장하며 연관된 패키지가 바인드될 때 이 값이 사용됩니다. 이 매개변수를 지정하지 않으면 db2sqljcustomize는 NULLID의 컬렉션 ID를 사용합니다.

**-onlinecheck YES|NO**

SQLJ 프로그램에서 데이터 유형 온라인 검사를 수행할지 여부를 지정합니다. -url 또는 -datasource 옵션은 온라인 검사에 사용할 데이터 소스를 판별합니다. -url 또는 -datasource 매개변수가 지정된 경우 디폴트값은 YES입니다. 그렇지 않으면 디폴트값이 NO입니다.

**-qualifier *qualifier-name***

온라인 검사 중에 SQLJ 프로그램에서 규정되지 않은 오브젝트에 사용할 규정자를 지정합니다. 패키지가 바인드되면 이 값이 규정자로 사용되지 않습니다.

**-rootpkgname|singlepkgname**

프로그램과 연관된 패키지의 이름을 지정합니다. -automaticbind가 NO이면 db2sqljbind 실행 시 이들 패키지 이름이 사용됩니다. 매개변수의 의미는 다음과 같습니다.

**-rootpkgname *package-name-stem***

사용자 정의 프로그램이 4개의 DB2 분리 레벨마다 하나씩 4개의 패키지를 작성하도록 지정합니다. 4개의 패키지 이름은 다음과 같습니다.

*package-name-stem1*

분리 레벨 UR의 경우

*package-name-stem2*

분리 레벨 CS의 경우

*package-name-stem3*

분리 레벨 RS의 경우

*package-name-stem4*

분리 레벨 RR의 경우

-longpkgname이 지정되지 않은 경우에는 *package-name-stem*이 7바이트 이하의 영숫자 문자열이어야 합니다.

-longpkgname이 지정된 경우에는 *package-name-stem*이 127바이트 이하의 영숫자 문자열이어야 합니다.

**-singlepkgname *package-name***

사용자 정의 프로그램이 이름이 *package-name*인 하나의 패키지를 작성하도록 지정합니다. 이 옵션을 지정하면 하나의 분리 레벨에서만 프로그램을 실행할 수 있습니다. -bindoptions options string에서 ISOLATION 옵션을 지정하여 패키지의 분리 레벨을 지정합니다.

-longpkgname이 지정되지 않은 경우에는 *package-name*이 8바이트 이하의 영숫자 문자열이어야 합니다.

-longpkgname이 지정된 경우에는 *package-name*이 128바이트 이하의 영숫자 문자열이어야 합니다.

-singlepkgname 옵션 사용은 권장되지 않습니다.

**권장사항:** 목표 데이터 소스가 DB2인 경우, *package-name-stem* 또는 *package-name* 값에 대문자를 사용하십시오. 특정 CCSID 값으로 정의된 DB2 시스템에서는 패키지 이름 또는 콜렉션 이름에 소문자를 사용할 수 없습니다.

-rootpkgname 또는 -singlepkgname을 지정하지 않는 경우, db2sqljcustomize는 직렬화된 프로파일 이름을 기본으로 하는 4개의 패키지 이름을 생성합니다. 직렬화된 프로파일 이름의 양식은 다음과 같습니다.

*program-name\_SJProfileIDNumber.ser*

생성되는 4개의 패키지 이름 양식은 다음과 같습니다.

*Bytes-from-program-nameIDNumberPkgIsolation*

표 38에서는 생성된 패키지 이름의 파트 및 각 파트의 바이트 수를 표시합니다.

패키지 이름의 최대 길이는 *maxlen*입니다. -longpkgname이 지정되지 않은 경우 *maxlen*은 8입니다. -longpkgname이 지정된 경우 *maxlen*은 128입니다.

표 38. db2sqljcustomize를 통해 생성되는 패키지 이름 파트

패키지 이름 파트	바이트 수	값
<i>Bytes-from-program-name</i>	$m = \min(\text{Length}(\text{프로그램 이름}), \text{대문자로 된 프로그램 이름의 첫 } m\text{바이트 } maxlen-1-\text{Length}(IDNumber))$	
<i>IDNumber</i>	$\text{Length}(IDNumber)$	<i>IDNumber</i>
<i>PkgIsolation</i>	1	1, 2, 3 또는 4. 이 값은 패키지의 트랜잭션 분리 레벨을 나타냅니다. 표 39의 내용을 참조하십시오.

표 39에서는 db2sqljcustomize를 통해 생성되는 패키지 이름의 *PkgIsolation* 부분의 값을 표시합니다.

표 39. PkgIsolation 값 및 연관된 분리 레벨

<i>PkgNumber</i> 값	패키지의 분리 레벨
1	커밋되지 않은 읽기(UR)
2	커서 안정성(CS)
3	읽기 안정성(RS)
4	반복 읽기(RR)

예: 프로파일 이름이 ThisIsMyProg\_SJProfile111.ser이라고 가정합니다. db2sqljcustomize 옵션 -longpkgname이 지정되지 않았습니까. 따라서 *Bytes-from-*

*program-name*은 ThisIsMyProg의 첫 4바이트(대문자로 변환됨) 또는 THIS입니다. *IDNumber*는 111입니다. 4개의 패키지 이름은 다음과 같습니다.

```
THIS1111  
THIS1112  
THIS1113  
THIS1114
```

예: 프로파일 이름이 ThisIsMyProg\_SJProfile111.ser이라고 가정합니다. *db2sqljcustomize* 옵션 *-longpkgname*이 지정되었습니다. 따라서 *Bytes-from-program-name*은 ThisIsMyProg(대문자로 변환됨) 또는 THISISMYPROG입니다. *IDNumber*는 111입니다. 4개의 패키지 이름은 다음과 같습니다.

```
THISISMYPROG1111  
THISISMYPROG1112  
THISISMYPROG1113  
THISISMYPROG1114
```

예: 프로파일 이름이 A\_SJProfile0.ser이라고 가정합니다. *Bytes-from-program-name*은 A입니다. *IDNumber*는 0입니다. 따라서 4개의 패키지 이름은 다음과 같습니다.

```
A01  
A02  
A03  
A04
```

*db2sqljcustomize*가 패키지 이름을 작성하도록 하는 것은 권장되지 않습니다. 생성된 패키지 이름이 기존 패키지 이름과 동일한 경우에는 *db2sqljcustomize*가 기존 패키지를 겹쳐씁니다. 패키지 이름이 고유하도록 하려면 *-rootpkgname*을 지정하십시오.

### **-longpkgname**

*db2sqljcustomize*가 생성하는 DB2 패키지의 이름이 최대 128바이트가 될 수 있도록 지정합니다. 긴 패키지 이름을 지원하는 서버에서 패키지를 바인드 중인 경우에만 이 옵션을 사용하십시오. *-singlepkgname* 또는 *-rootpkgname*을 지정하는 경우 다음 조건에서 *-longpkgname*도 지정해야 합니다.

- *-singlepkgname*의 인수 길이가 8바이트보다 깁니다.
- *-rootpkgname*의 인수 길이가 7바이트보다 깁니다.

### **-staticpositioned NO|YES**

반복자(iterator)를 사용하는 위치 지정된 UPDATE문과 동일한 소스 파일에 반복자가 선언된 경우, 위치 지정된 UPDATE가 정적으로 바인드된 명령문으로 실행되는지 여부를 지정합니다. 디폴트값은 NO입니다. NO는 위치 지정된 UPDATE가 동적으로 준비된 명령문으로 실행됨을 의미합니다.

### **-zosDescProcParms**

DB2가 스토어드 프로시저 매개변수에 대해 DESCRIBE 연산을 수행하도록 지정합니다.

-zosDescProcParms는 DB2 데이터베이스 서버에서 실행할 프로그램에만 적용됩니다.

DESCRIBE 정보가 사용 가능한 경우, SQLJ에는 INOUT 및 OUT 매개변수의 길이 및 정밀도에 대한 정보가 있어 해당 매개변수에 필요한 크기의 메모리만 할당합니다. DESCRIBE 정보의 사용 가능성은 문자 INOUT 매개변수, LOB OUT 매개변수 및 10진수 OUT 매개변수의 스토리지 사용에 가장 큰 영향을 미칩니다.

-zosDescProcParms가 지정되면 DB2 데이터베이스 서버가 -zosProcedurePath의 디폴트값 또는 지정된 값을 사용하여 DESCRIBE 정보가 요청되는 스토어드 프로시저의 규정되지 않은 이름을 분석합니다.

#### **-zosProcedurePath** *procedure-path*

SQLJ 프로그램을 온라인 검사하는 동안 규정되지 않은 스토어드 프로시저 이름을 분석하기 위해 DB2에서 사용하는 스키마 이름 목록을 지정합니다.

-zosProcedurePath는 DB2 데이터베이스 서버에서 실행할 프로그램에만 적용됩니다.

목록은 문자열 값으로 큰따옴표로 묶인 스키마 이름 목록이 쉼표로 구분되어 있습니다. DB2 데이터베이스 서버는 규정되지 않은 스토어드 프로시저 이름을 분석하도록 해당 목록을 SQL 경로에 삽입합니다. SQL 경로는 다음과 같습니다.

SYSIBM, SYSFUN, SYSPROC, *procedure-path*, *qualifier-name*, *user-ID*

*qualifier-name*은 -*qualifier* 매개변수 값이고 *user-ID*는 -*user* 매개변수 값입니다.

DB2 데이터베이스 서버는 해당 데이터베이스 서버에 존재하는 스토어드 프로시저 이름과 일치하는 이름을 찾을 때까지 왼쪽에서 오른쪽으로 SQL 경로에서 스키마 이름을 찾습니다. DB2 데이터베이스 서버가 일치하는 이름을 찾으면 DB2 카탈로그에서 해당 스토어드 프로시저의 매개변수에 대한 정보를 가져옵니다. DB2 데이터베이스 서버가 일치하는 이름을 찾을 수 없는 경우에는 SQLJ가 DB2 카탈로그 정보 없이 매개변수 데이터를 설정합니다.

-zosProcedurePath가 지정되지 않은 경우 DB2 데이터베이스 서버는 다음 SQL 경로를 사용합니다.

SYSIBM, SYSFUN, SYSPROC, *qualifier-name*, *user-ID*

-*qualifier* 매개변수가 지정되지 않은 경우 SQL 경로에는 *qualifier-name*이 포함되지 않습니다.

#### **-genDBRM**

db2sqljcustomize가 DBRM(Database Request Module)을 생성하도록 지정합니다. 이 DBRM을 사용하여 DB2 플랜 및 패키지를 작성할 수 있습니다.

-genDBRM은 DB2 데이터베이스 서버에서 실행할 프로그램에만 적용됩니다.

-genDBRM 및 -automaticbind NO가 지정된 경우, db2sqljcustomize가 DBRM을 작성하지만 이를 DB2 패키지에 바인드하지 않습니다. -genDBRM 및 -automaticbind YES가 지정된 경우, db2sqljcustomize가 DBRM을 작성하고 이를 DB2 패키지에 바인드합니다.

각 DB2 분리 레벨마다 하나의 DBRM이 작성됩니다. 생성된 DBRM 파일의 이름 지정 규칙은 패키지의 이름 지정 규칙과 같습니다. 예를 들어, -rootpkgname SQLJSA0가 지정되고 -genDBRM도 지정된 경우 4개의 DBRM 파일 이름은 다음과 같습니다.

- SQLJSA01
- SQLJSA02
- SQLJSA03
- SQLJSA04

**-DBRMDir** *directory-name*

-genDBRM이 지정된 경우, -DBRMDir은 db2sqljcustomize가 생성된 DBRM 파일을 저장하는 로컬 디렉토리를 지정합니다. 디폴트는 현재 디렉토리입니다.

-DBRMDir은 DB2 데이터베이스 서버에서 실행할 프로그램에만 적용됩니다.

**-tracefile** *file-name*

추적을 사용 가능하게 하고 추적 정보에 관한 출력 파일을 식별합니다. IBM Software Support의 지시문에 따라서만 이 옵션을 지정해야 합니다.

**-tracelevel**

-tracefile이 지정된 경우, db2sqljcustomize를 실행하는 동안 추적할 내용을 표시합니다. 디폴트값은 TRACE\_SQLJ입니다. IBM Software Support의 지시문에 따라서만 이 옵션을 지정해야 합니다.

**serialized-profile-name|file-name.grp**

사용자 정의할 하나 이상의 직렬화된 프로파일 이름을 지정합니다. 지정된 직렬화된 프로파일은 CLASSPATH 환경 변수로 이름이 지정된 디렉토리에 있어야 합니다. 직렬화된 프로파일 이름의 양식은 다음과 같습니다.

*program-name\_SJProfileIDNumber.ser*

.ser 확장자 사용 여부에 관계 없이 직렬화된 프로파일 이름을 지정할 수 있습니다.

*program-name*은 SQLJ 소스 프로그램의 이름으로 .sqlj 확장자를 사용하지 않습니다. *n*은 0과 *m-1* 사이의 정수로 여기서 *m*은 SQLJ 변환기가 SQLJ 소스 프로그램에서 생성한 직렬화된 프로파일의 수입니다.

다음 방법 중 하나로 직렬화된 프로파일 이름을 지정할 수 있습니다.

- db2sqljcustomize 명령으로 이름을 나열하십시오. 여러 개의 직렬화된 프로파일 이름은 스페이스로 구분해야 합니다.

- 이름이 *file-name.grp*인 파일에서 각 라인에 하나씩 직렬화된 프로파일 이름을 지정하고 `db2sqljcustomize` 명령으로 *file-name.grp*를 지정하십시오.

직렬화된 프로파일 이름을 둘 이상 지정하고 디폴트값인 `-automaticbind YES`를 지정하거나 사용하는 경우, `db2sqljcustomize`는 해당 프로파일에서 단일 DB2 패키지를 바인드합니다. `db2sqljcustomize`를 사용하여 다중 직렬화된 프로파일에서 단일 DB2 패키지를 작성하는 경우, `-rootpkgname` 또는 `-singlepkgname` 옵션도 지정해야 합니다.

직렬화된 프로파일 이름을 둘 이상 지정하고 `-automaticbind NO`를 지정하는 경우, `db2sqljbind` 실행 시 직렬화된 프로파일을 단일 DB2 패키지에 바인드하려면 `db2sqljcustomize`와 `db2sqljbind`에서 동일한 직렬화된 프로파일 이름을 동일한 순서로 지정해야 합니다.

## 출력

`db2sqljcustomize`가 실행되면 사용자 정의 직렬화된 프로파일을 작성합니다. 자동 바인드 값이 YES인 경우에는 DB2 패키지도 작성합니다.

## 예:

```
db2sqljcustomize -user richler -password mordecai
                 -url jdbc:db2:/server:50000/sample -collection duddy
                 -bindoptions "EXPLAIN YES" pgmname_SJProfile0.ser
```

## 사용법 참고사항

**항상 온라인 검사 수행 권장:** 직렬화된 프로파일을 사용자 정의하는 경우에는 온라인 검사를 사용해야 합니다. 온라인 검사는 DB2 호스트 변수의 데이터 유형 및 길이에 대한 정보를 판별하며 특히 다음 항목에서 중요합니다.

- `java.lang.String` 호스트 변수 및 CHAR 컬럼이 포함된 술어

기타 호스트 언어의 문자 변수와 달리 Java 문자열 호스트 변수는 길이 속성을 사용하여 선언되지 않습니다. 문자 호스트 변수가 포함된 쿼리를 최적화하려면 DB2에 호스트 변수 길이가 필요합니다. 예를 들어, 쿼리에 문자열 호스트 변수를 CHAR 컬럼과 비교하는 술어가 있고 CHAR 컬럼에 인덱스가 정의되어 있다고 가정합니다. DB2에서 호스트 변수의 길이를 판별할 수 없는 경우, 인덱스 스캔 대신 테이블 스페이스 스캔을 수행합니다. 온라인 검사를 수행하면 해당 문자 컬럼의 길이를 제공함으로써 이러한 문제점이 발생하지 않습니다.

- `java.lang.String` 호스트 변수 및 GRAPHIC 컬럼이 포함된 술어

온라인 검사를 수행하지 않으면 DB2에서 문자열 호스트 변수를 GRAPHIC 컬럼과 비교하는 술어가 발견되는 경우 바인드 오류(SQLCODE -134)를 발행합니다.

- 리모트 서버의 SQLJ SELECT문의 결과 테이블에 있는 컬럼 이름:



온라인 검사를 수행하지 않으면 드라이버가 리포트 SELECT의 결과 테이블에서 사용할 컬럼 이름을 판별할 수 없습니다.

**여러 개의 직렬화된 프로파일을 함께 사용자 정의:** 여러 개의 직렬화된 프로파일을 함께 사용자 정의하여 단일 DB2 패키지를 작성할 수 있습니다. 이를 수행하고, -staticpositioned YES를 지정하면 UPDATE 또는 DELETE문이 커서 선언과 다른 소스 파일에 있어도 패키지 초반에 선언된 커서를 참조하는 위치 지정된 UPDATE 또는 DELETE문이 정적으로 실행됩니다. 프로그램이 여러 소스 파일로 구성되어 있는 경우에 -staticpositioned YES의 동작을 구현하려면 db2sqljcustomize 명령으로 프로파일의 순서를 지정하여 패키지에서 위치 지정된 UPDATE 또는 DELETE문에 앞서 커서를 선언하도록 해야 합니다. 이를 수행하려면, 반복자(iterator)에 결과 테이블을 지정하는 SELECT문이 포함된 프로파일을 해당 반복자를 참조하는 위치 지정된 UPDATE 또는 DELETE문이 포함된 프로파일 이전에 나열하십시오.

**다른 데이터 소스에서 사용자 정의된 임의의 데이터 소스에서 사용자 정의된 직렬화 프로파일 사용:** db2sqljcustomize를 실행하여 임의의 데이터 소스에서 SQLJ 프로그램에서 사용할 사용자 정의된 직렬화 프로파일을 생성한 다음 다른 데이터 소스에서 해당 프로파일을 사용할 수 있습니다. db2sqljcustomize를 한 번 실행하여 작성한 사용자 정의된 직렬화 프로파일에 대해 db2sqljbind를 여러 번 실행하여 이를 수행합니다. 이들 데이터 소스에서 프로그램을 실행하는 경우 프로그램에서 액세스하는 DB2 오브젝트는 모든 데이터 소스에서 동일해야 합니다. 예를 들어, 모든 데이터 소스의 테이블에는 동일한 인코딩 스킴 및 같은 데이터 유형을 가진 동일한 컬럼이 있어야 합니다.

**-collection 매개변수 사용:** db2sqljcustomize는 DB2 컬렉션 이름을 생성된 각 사용자 정의된 직렬화 프로파일에 저장합니다. SQLJ 프로그램이 실행되면 드라이버가 사용자 정의된 직렬화 프로파일에 저장된 컬렉션 이름을 사용하여 실행할 패키지를 검색합니다. -collection 매개변수의 값으로 사용자 정의된 직렬화 프로파일에 저장된 이름을 판별합니다. 직렬화 프로파일에는 하나의 컬렉션 ID만 저장할 수 있습니다. 그러나 -bindoptions 매개변수에 COLLECTION 옵션을 지정하여 다중 패키지 컬렉션에 동일한 직렬화 프로파일을 바인드할 수 있습니다. 직렬화 프로파일에 지정된 컬렉션이 아닌 다른 컬렉션에 있는 패키지를 실행하려면 프로그램에 SET CURRENT PACKAGESET 문을 포함시키십시오.

**VERSION 매개변수 사용:** VERSION 매개변수를 사용하여 동일한 SQLJ 프로그램에서 사용할 둘 이상의 패키지 버전을 같은 컬렉션에 바인드할 수 있습니다. SQLJ 소스 프로그램을 변경했으며 이전 버전 및 새 버전의 프로그램을 실행하려는 경우 이와 같이 수행합니다.

두 버전의 패키지를 유지하려면 다음 단계를 수행하십시오.

1. 소스 프로그램의 코드를 변경하십시오.

2. 새 직렬화 프로파일을 작성하도록 소스 프로그램을 변환하십시오. 원래 직렬화 프로파일을 겹쳐쓰지 않도록 하십시오.
3. db2sqljcustomize를 실행하여 직렬화 프로파일을 사용자 정의하고 원래 패키지와 동일한 콜렉션에서 동일한 패키지 이름을 사용하여 DB2 패키지를 작성하십시오. 새 패키지를 바인드할 때 원래 패키지 작성 시 사용한 것과 같은 -rootpkgname 및 -collection의 값을 사용하여 이를 수행하십시오. -bindoptions 매개변수에 VERSION 옵션을 지정하여 새 사용자 정의 직렬화 프로파일 및 새 패키지에 버전 ID를 저장 하십시오.

이 단계를 수행할 때 반드시 VERSION 옵션을 지정해야 합니다. 그렇지 않으면 원래 패키지를 겹쳐쓰게 됩니다.

이전 버전의 프로그램을 실행하면 DB2에서 이전 버전의 패키지를 로드합니다. 새 버전의 프로그램을 실행하면 DB2에서 새 버전의 패키지를 로드합니다.

**DB2에서 패키지 및 플랜 바인딩:** db2sqljcustomize -genDBRM 매개변수를 사용하여 로컬 시스템에 DBRM을 작성할 수 있습니다. 그런 다음 작성한 DBRM을 DB2 시스템에 전송하여 해당 시스템에서 이를 패키지 또는 플랜에 바인드할 수 있습니다. 이 기술을 사용하려면 DBRM 파일을 z/OS 시스템 및 레코드 형식이 FB이고 레코드 길이가 80인 파티션된 데이터 세트에 2진 파일로 전송해야 합니다. 패키지 또는 플랜을 바인드하는 경우 다음 바인드 옵션 값을 지정해야 합니다.

#### **ENCODING(EBCDIC)**

DB2의 IBM Data Server Driver for JDBC and SQLJ에서는 패키지 및 플랜에 적합한 EBCDIC 인코딩이 필요합니다.

#### **DYNAMICRULES(BIND)**

이 옵션은 SQLJ에서 동적 SQL 사용 시 권한 부여 규칙에 일관성이 있도록 합니다. SQLJ에서는 다중 SQLJ 프로그램이 관련된 위치 지정된 UPDATE 또는 DELETE 조작성에 동적 SQL을 사용합니다.

#### **DBPROTOCOL(DRDA)**

개인용 프로토콜은 더 이상 사용되지 않으므로 모든 응용프로그램에서 DBPROTOCOL(DRDA)을 사용해야 합니다. 그러나 세 파트로 구성된 리모트 테이블 이름을 사용하는 SQLJ 응용프로그램에서는 DBPROTOCOL(DRDA)을 사용해야 합니다. 그렇지 않으면 해당 응용프로그램이 실패합니다.

---

## 제 255 장 db2sqljprint - SQLJ 프로파일 프린터

db2sqljprint는 사용자 정의된 프로파일 버전의 콘텐츠를 일반 텍스트로 인쇄합니다.

### 권한 부여

None

### 명령 구문

▶▶—db2sqljprint—*profilename*—————▶▶

### 명령 매개변수

*profilename*

SQLJ 프로파일 파일의 상대 또는 절대 이름을 지정합니다. SQLJ 파일이 Java 소스로 변환되면 파일에 포함된 SQL 조작에 대한 정보가 프로파일이라고 하는 SQLJ에서 생성된 리소스 파일에 저장됩니다. 원래 입력 파일 이름 뒤에 표시되는 접미부 \_SJProfileN(여기서 N은 정수)으로 프로파일을 식별합니다. 프로파일의 확장자는 .ser입니다. .ser 확장자 사용 여부에 관계 없이 프로파일 이름을 지정할 수 있습니다.

### 예:

```
db2sqljprint pgmname_SJProfile0.ser
```



---

## 제 256 장 db2start - DB2 시작

단일 데이터베이스 파티션이나 파티션된 데이터베이스 환경에 정의된 모든 데이터베이스 파티션에서 현재 데이터베이스 관리 프로그램 인스턴스 백그라운드 프로세스를 시작합니다. 데이터베이스에 연결하거나, 응용프로그램을 프리컴파일하거나, 패키지를 데이터베이스에 바인딩하기 전에 서버에서 DB2를 시작하십시오. db2start는 시스템 명령이나 CLP 명령으로 실행될 수 있습니다.

db2start 명령은 Windows 서비스로서 DB2 제품을 시작합니다. Windows에서 DB2 제품 설치시 db2start 호출 시 /D 스위치를 지정하여 여전히 프로세스로서 실행될 수 있습니다. 또한 DB2 제품 설치시 제어판이나 NET START 명령을 사용하여 서비스로서 시작될 수 있습니다.

db2start는 Windows 서비스를 시작하므로, 서비스 시작을 위한 Windows 요구사항을 충족시켜야 합니다. 확장 보안을 사용 안하면, 관리자, 서버 운영자 또는 권한이 있는 사용자 그룹의 구성원이어야 합니다. 확장 보안이 사용 가능하면, 데이터베이스를 시작하기 위해 관리자 그룹이나 DB2ADMNS 그룹 중 하나의 구성원이어야 합니다.

다중 파티션 데이터베이스의 db2start 조작이 start\_stop\_time 데이터베이스 관리 프로그램 구성 매개변수로 지정된 값 내에서 완료되지 않은 경우, 시간종료된 데이터베이스 파티션에는 시작된 데이터베이스 관리 프로그램 인스턴스 백그라운드 프로세스가 없습니다(데이터베이스 파티션과 연관된 모든 자원이 제거됨). start\_stop\_timeout의 낮은 값을 갖는 여러 데이터베이스 파티션이 있는 환경에서 이러한 동작이 발생합니다. 이 동작을 해결하려면, start\_stop\_time 데이터베이스 관리 프로그램 구성 매개변수의 값을 증가시키십시오.

Linux 및 UNIX 운영 체제에서의 루트 설치 DB2 사본의 경우, db2start 명령은 인스턴스 소유자 ID의 영구 설정 ulimit를 변경하지 않고 데이터베이스 관리 프로그램에서 필요로 하는 ulimit 값을 설정합니다.

루트 서버가 아닌 설치의 경우, 'data'의 ulimit를 'unlimited'로 설정하고 'nofiles'를 'unlimited' 또는 시스템에서 허용하는 최대값으로 설정해야 합니다.



---

## 제 257 장 db2stat - Windows의 DB2 프로세스 상태

Windows 시스템의 경우 인스턴스에서 실행되는 모든 DB2 프로세스는 db2stat 명령을 사용하여 표시될 수 있습니다.

### 권한 부여

없음

### 필수 연결

없음

### 명령 구문

▶—db2stat—◀

### 명령 매개변수

#### db2stat

인스턴스에서 실행되는 모든 DB2 프로세스를 출력합니다.

#### 예:

```
C:\Program Files\IBM\SQLLIB\BIN>db2stat
```

#### Environment Strings

```
--> DB2CLP=DB20FADE
--> DB2INSTANCE=DB2
--> DB2PATH=C:\Program Files\IBM\SQLLIB
```

#### DB2 Processes

DB2DASRRM	1960	x7A8
DB2MGMTSVC	2012	x7DC
DB2RCMD	1212	x4BC
DB2DASSTM	2044	x7FC
DB2SYSTRAY	724	x2D4
DB2	3100	xC1C
DB2BP	3180	xC6C
DB2SYSCS	1592	x638
DB2FMP	3468	xD8C
DB2STAT	1748	x6D4

### 사용 시 참고사항

Windows의 경우 주의할 점은 DB2는 프로세스 기반이 아닌 스레드 기반이므로 모든 인스턴스 EDU에 하나의 프로세스(DB2SYSCS)만 나타난다는 것입니다. Windows에서 리턴되는 정보의 등급은 Linux/UNIX 시스템과 다르지만 때때로 실행 인스턴스의

프로세스 ID를 아는 것이 유용한 경우도 있습니다. 예를 들어, Windows Task Manager 유틸리티를 사용하여 지정된 프로세스 ID의 CPU 및 메모리 사용을 관별할 수 있습니다.



---

## 제 258 장 db2stop - Stop DB2

현재 데이터베이스 관리 프로그램 인스턴스를 중지합니다. db2stop은 시스템 명령어나 CLP 명령으로 실행될 수 있습니다.

다중 파티션 데이터베이스의 db2stop 조작이 start\_stop\_time 데이터베이스 관리 프로그램 구성 매개변수로 지정된 값 내에서 완료되지 않은 경우, 시간종료된 데이터베이스 파티션은 내부적으로 제거됩니다(데이터베이스 파티션과 연관된 모든 자원이 제거됨). start\_stop\_time의 낮은 값을 갖는 여러 데이터베이스 파티션이 있는 환경에서는 이러한 동작이 발생합니다. 이 동작을 해결하려면, start\_stop\_time의 값을 증가시키십시오.



## 제 259 장 db2support - 문제점 분석 및 환경 콜렉션 도구

클라이언트 또는 서버 머신에 대한 환경 데이터를 수집하고 시스템 데이터가 들어 있는 파일을 압축된 파일 아카이브에 배치합니다.

이 도구는 또한 사용자와의 대화식 질문 및 응답 프로세스를 통해 문제점의 본질에 대한 기본 데이터를 수집합니다.

### 권한 부여

완료된 출력의 경우 인스턴스 소유자가 이 유틸리티를 호출해야 합니다. 시스템에 대한 특권이 보다 제한적인 사용자도 이 도구를 실행할 수 있지만 일부 데이터 콜렉션 조치를 수행하면 보고와 출력이 감소합니다.

### 필수 연결

없음

### 명령 구문

```
►► db2support output path
    [-a] [-r] [-cd current degree]
    [-cl collect level] [-co] [-cs current schema]
    [-d database name] [-c] [-u userid] [-p password] [-f]
    [-fp function path] [-g] [-h] [-il isolation level] [-l]
    [-m] [-n] [-ol optimization level] [-op optimization profile]
    [-ot optimization tables] [-q] [-ra refresh age] [-ro] [-s]
    [-se embedded SQL file] [-sf SQL file] [-st SQL statement]
    [-td termination character delimiter] [-v] [-x]
```

주:

1. 옵티마이저 모드에서 이 도구를 호출하기 위한 별도 옵션은 없습니다.
2. db2support 도구는 -st, -sf 또는 -se 옵션이 지정된 경우에만 잘못된 쿼리 관련 정보를 수집합니다. 최적화 중에 오류 또는 트랩이 있는 경우, 잘못된 쿼리를 설명하지 않고 -cl 0(수집 레벨 0)을 사용하여 모든 카탈로그 테이블과 db2look 테이블 정의를 수집해야 합니다. 옵티마이저 문제점을 해결하려면 여기서 언급된 네 가지 옵션 중 하나를 지정해야 합니다.
3. 특수 레지스터가 명령문 실행 중 디폴트값 이외의 값으로 설정된 경우, 올바른 문제점 분석을 위해서는 동일한 값이 매개변수로서 db2support 도구로 전달되어야 합니다.

## 명령 매개변수

### *output path*

아카이브 라이브러리가 작성될 경로를 지정합니다. 이 디렉토리는 아카이브에 포함하기 위해 사용자 작성 파일이 배치되어야 하는 디렉토리입니다.

### **-a | -all\_core**

모든 코어 파일이 캡처되는 것으로 지정합니다.

### **-r | -recent\_core**

최신 코어 파일이 캡처되는 것으로 지정합니다. -a 옵션이 지정되면 이 옵션이 무시됩니다.

### **-c | -connect**

지정된 데이터베이스 연결을 시도해야 하는 것으로 지정합니다.

### **-cd | -curdegree**

사용할 현재 등급 특수 레지스터의 값을 지정합니다. 디폴트값은 **dft\_degree** 데이터베이스 구성 매개변수의 값입니다.

### **-cl | -collect**

리턴될 성능 정보의 레벨 값을 지정합니다. 가능한 값은 다음과 같습니다.

- 0 = 카탈로그(db2look, dbcfg, dbmcfg, db2set)만 수집
- 1 = 0과 exfmt 수집
- 2 = 1과 .db2service 수집(디폴트)
- 3 = 2와 db2batch 수집

**-co** 데이터베이스 내 모든 테이블의 카탈로그를 수집합니다. 디폴트값은 문제점이 있는 쿼리에서 사용되는 테이블에 대한 카탈로그 정보만 수집하는 것입니다.

### **-cs | -curschema**

명령문에서 규정되지 않은 테이블 이름을 규정하기 위해 사용할 현재 스키마의 값을 지정합니다. 디폴트값은 현재 세션 사용자의 권한 부여 ID입니다.

### **-d database\_name | -database database\_name**

데이터가 수집되는 데이터베이스의 이름을 지정합니다.

**-f | -flow**

사용자가 Press <Enter> key to continue를 요청하는 경우 일시정지를 무시합니다. 이 옵션은 스크립트 또는 일부 기타 자동 프로시저(자동 실행이 필요한 경우)를 통해 db2support 도구를 실행하거나 호출할 때 유용합니다.

**-fp | -funcpath**

규정되지 않은 사용자 정의 함수와 유형을 해석하기 위해 사용될 함수 경로 특수 레지스터의 값을 지정합니다. 디폴트값은 『SYSIBM』, 『SYSFUN』, 『SYSPROC』, X입니다. 여기서 X는 큰따옴표로 구분된 USER 특수 레지스터의 값입니다.

**-g | -get\_dump**

코어 파일을 제외한 덤프 디렉토리의 모든 파일이 캡처되는 것으로 지정합니다.

**-h | -help**

도움말 정보를 표시합니다. 이 옵션이 지정되면 다른 모든 옵션은 무시되고 도움말 정보만 표시됩니다.

**-il | -isolation**

데이터에 액세스하는 동안 데이터를 잠그고 다른 프로세스와 분리시키는 방법을 판별하기 위해 사용할 분리 레벨을 지정합니다. 디폴트로, CURRENT ISOLATION 특수 레지스터는 공백으로 설정됩니다.

**-l | -logs**

활성 로그가 캡처되는 것으로 지정합니다.

**-m | -html**

모든 시스템 출력이 HTML 형식 파일로 덤프되는 것으로 지정합니다. 디폴트로, 모든 시스템 관련 정보는 일반 텍스트 파일로 덤프됩니다(이 매개변수를 사용하지 않는 경우).

**-n | -number**

현재 문제점의 문제점 관리 보고서(PMR) 번호 또는 ID를 지정합니다.

**-ol | -optlevel**

사용할 최적화 레벨 특수 레지스터의 값을 지정합니다. 디폴트값은 *dft\_queryopt* 데이터베이스 구성 매개변수의 값입니다.

**-op | -optprofile**

사용할 최적화 프로파일 특수 레지스터의 값을 지정합니다. 명령문이 바운드를 때 유효한 최적화 프로파일이 있는 경우에만 필요합니다. 디폴트값은 『』(비어 있는 문자열)입니다.

**-ot | -opttables**

동적 SQL 쿼리 처리를 최적화할 때 고려할 수 있는 테이블 유형을 식별하는데 사용되는 『CURRENT MAINTAINED TABLE TYPES FOR

OPTIMIZATION』 특수 레지스터의 값을 지정합니다. CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION의 초기값은 『SYSTEM』입니다.

**-p** *password* | **-password** *password*  
사용자 ID의 암호를 지정합니다.

**-q** | **-question\_response**  
대화식 문제점 분석 모드를 사용하는 것으로 지정합니다.

**-ra** | **-refreshage**  
갱신 유효 기간 특수 레지스터의 값을 지정합니다. 명령문에 테이블을 참조하는 구체화된 쿼리 테이블(MQT)이 있는 경우에만 적용됩니다. CURRENT REFRESH AGE의 디폴트값은 0입니다.

**-ro** | **-reopt**  
쿼리 설명 시 REOPT ONCE와 함께 EXPLAIN을 사용해야 하는지 여부를 지정합니다. 디폴트값은 REOPT ONCE 옵션을 무시하는 것입니다.

**-s** | **-system\_detail**  
자세한 하드웨어 및 운영 체제 정보를 수집하는 것으로 지정합니다.

**-se** *embedded SQL file* | **-sqlembed** *embedded SQL file*  
데이터를 수집하는 SQL문이 포함된 Embedded SQL 파일의 경로를 지정합니다.

**-sf** *SQL file* | **-sqlfile** *SQL file*  
데이터를 수집하는 SQL 문이 포함된 파일 경로를 지정합니다.

**-st** *SQL statement* | **-sqlstmt** *SQL statement*  
데이터를 수집하는 SQL문을 지정합니다.

**-td** | **-delimiter**  
명령문 종료 문자를 지정합니다. 이 명령 매개변수는 db2 명령의 -td 옵션과 동일한 방식으로 실행됩니다. 디폴트값은 세미콜론입니다.

**-u** *userid* | **-user** *userid*  
데이터베이스에 연결할 사용자 ID를 지정합니다.

**-v** | **-verbose**  
이 도구가 실행되는 동안 상세 출력을 사용함을 지정합니다.

**-x** | **-xml\_generate**  
대화식 문제점 분석 모드(-q 모드)에서 사용되는 전체 의사 결정 트리 논리가 포함된 XML 문서가 생성되는 것으로 지정합니다.

예:

db2support 도구는 다음 방법 중 하나로 옵티마이저 모드에서 호출됩니다.

- 명령행의 SQL 문으로서.

```
db2support <output_directory> -d <database name> -st <sql_statement>
```

db2support 도구는 『bad\_query.sql』 파일에 쿼리를 복사하여 옵티마이저 디렉토리에 쿼리를 저장합니다.

- 파일에 저장된 SQL 문으로서.

```
db2support <output_directory> -d <database name> -sf <sql_file>
```

쿼리가 포함된 파일은 도구를 사용하여 옵티마이저 디렉토리에 복사됩니다.

- 쿼리에 문제점이 있는 임베디드(embedded) 정적 SQL 문이 포함된 파일로서.

```
db2support <output_directory> -d <database name> -se <embedded_sql_file>
```

쿼리가 포함된 파일은 도구를 사용하여 옵티마이저 디렉토리에 복사됩니다. 파일은 현재 디렉토리에 없어도 되지만 호출 사용자 ID로 읽을 수 있어야 합니다.

- 다른 레벨의 성능 정보를 리턴하는 동안.

```
db2support <output_directory> -d <database name> -collect 0
```

db2support 도구는 요청된 세부사항 레벨을 기반으로 다양한 레벨의 성능 정보를 수집합니다. 0에서 3까지의 값은 증가하는 세부사항을 수집합니다. 레벨 0을 사용하는 경우 프로덕션 데이터베이스의 데이터베이스 오브젝트를 재생하는 데 사용할 수 있는 카탈로그 정보와 테이블 정의가 수집됩니다.

디폴트로 설정된 옵티마이저 관련 특수 레지스터를 사용하여 느린 쿼리를 진단하기 위해 정보를 수집하려면 다음을 사용하십시오.

```
db2support . -d sample -st "SELECT * FROM EMPLOYEE"
```

이 예에서는 모든 데이터가 db2support.zip 파일에 리턴됩니다. 진단 정보는 현재 디렉토리와 서브디렉토리에서 작성됩니다.(가 출력 경로로 지정됨). 시스템 정보와 진단 파일 또한 수집됩니다.

이전 예에 표시된 것과 동일한 정보와 옵티마이저 관련 특수 레지스터의 사용자 정의 값을 수집하려면 다음을 사용하십시오.

```
db2support . -d sample -st "SELECT * FROM EMPLOYEE" -cs db2usr -cd 3  
-ol 5 -ra ANY -fp MYSCHEMA -op MYPROFSHEMA.MYPROFILE -ot ALL -il CS
```

이 예에서는 특수 레지스터를 설정합니다. 즉, 현재 스키마를 db2usr로, 현재 등급을 3으로, 최적화 수준을 5로, 새로 고침 유효 기간을 ANY로, 함수 경로를 MYSCHEMA 스키마로, 최적화 프로파일을 MYPROFSHEMA.MYPROFILE로, 현재 유지보수되는 테이블 유형을 ALL로, 분리 레벨을 CS로 설정합니다. 이러한 값은 db2support가 지정된 데이터베이스에 대해 설정하고 전체 환경에는 영향을 주지 않는 연결에 대해서만 설정됨

니다. 진단을 정정하는 경우 쿼리 실행 시 사용된 것과 동일한 특수 레지스트리 변수를 제공해야 합니다.

## 사용 시 참고사항

비즈니스 데이터의 보안을 보호하기 위해 이 도구는 테이블 데이터, 스키마(DDL) 또는 로그를 수집하지 않습니다. 일부 옵션은 스키마와 데이터의 일부 측면(예를 들어, 아카이브 로그)을 포함하도록 허용합니다. 데이터베이스 스키마 또는 데이터를 노출시키는 옵션은 주의깊게 사용해야 합니다. 이 도구가 호출되면 데이터가 어느 정도 민감하게 처리되는지 나타내는 메시지가 표시됩니다.

db2support 도구에서 수집된 데이터는 도구가 실행되는 머신의 데이터입니다. 클라이언트 서버 환경의 경우, 데이터베이스 관련 정보는 데이터베이스가 인스턴스 첨부 또는 데이터베이스 연결을 통해 있는 머신의 정보입니다. 예를 들어, 진단 디렉토리 (DIAGPATH)의 운영 체제 또는 하드웨어 정보(-s 옵션)와 파일은 db2support 도구가 실행되는 로컬 머신의 정보와 파일입니다. 버퍼 풀 정보, 데이터베이스 구성 및 테이블스페이스 정보와 같은 데이터는 데이터베이스가 실제로 있는 머신의 데이터입니다.

db2support 옵티마이저가 승인하는 쿼리 유형에는 다음과 같은 제한사항이 있습니다.

- 다중 쿼리는 지원되지 않습니다. 파일에 여러 쿼리를 배치하는 경우 도구는 각 쿼리에 필요한 모든 오브젝트를 수집합니다. 그러나 마지막 쿼리만 설명됩니다. 이는 임베디드(embedded) 정적 SQL문이 있는 파일에도 해당됩니다.
- 도구는 고객 응용프로그램을 실행하지 않습니다. 그러나 설명한 세 가지 방법 중 하나를 사용하여 잘못되거나 느린 특정 쿼리를 평가하는 경우 db2support를 실행할 때 동시에 응용프로그램을 실행할 수 있습니다.
- 스토어드 프로시저는 지원되지 않습니다.

db2support는 동적 SQL에 대한 Explain 데이터를 수집하지 않습니다.



---

## 제 260 장 db2swtch - 디폴트 DB2 사본 및 데이터베이스 클라이언트 인터페이스 사본 전환

디폴트 DB2 사본과 디폴트 데이터베이스 클라이언트 인터페이스 사본을 모두 전환합니다. 디폴트 DB2 사본은 특정 DB2 사본을 대상으로 하지 않는 응용프로그램이 사용하는 사본입니다. db2swtch를 실행하면 디폴트 DB2 및 IBM Database Client Interface Selection 마법사가 시작됩니다. 이 마법사를 따르면 새 디폴트 DB2 사본을 설정하고 디폴트 데이터베이스 클라이언트 인터페이스 사본을 설정할 수 있습니다. 이 명령은 Windows 운영 체제에서만 사용할 수 있습니다.

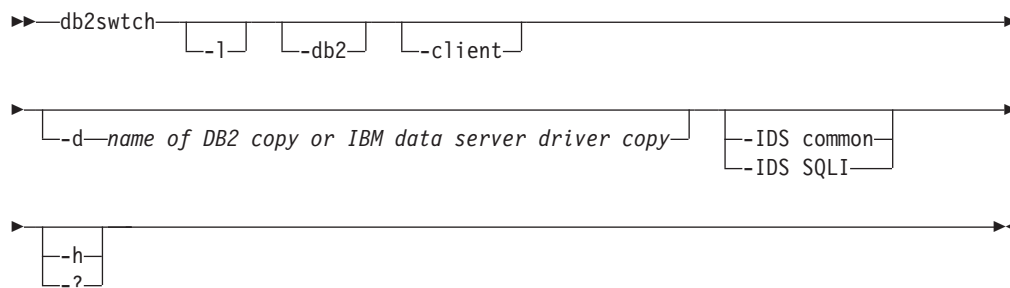
### 권한 부여

*sysadm*

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

인수 없음

유틸리티를 그래픽 모드로 시작합니다.

**-i** 시스템의 DB2 사본과 IBM Data Server Driver 사본의 목록을 표시합니다.

**-db2 -d** *name of DB2 copy*

디폴트 DB2 사본을 지정된 이름으로 전환합니다.

db2swtch -db2 -d *name of DB2 copy*

**-client -d** *name of DB2 copy or IBM data server driver copy*

디폴트 클라이언트 인터페이스 사본을 지정된 이름으로 전환합니다.

db2swtch -client -d *name of DB2 copy or IBM data server driver copy*

**-d** *name of DB2 copy*

디폴트 DB2 사본과 클라이언트 인터페이스 사본을 모두 지정된 이름으로 전환합니다.

`db2swtch -d name of DB2 copy`

**-IDS**

**일반** machine.config의 IDS .NET Data Provider 참조 경로를 공통 IDS .NET Data Provider로 재지정합니다.

**SQLI** machine.config의 IDS .NET Data Provider 참조 경로를 SQLI IDS .NET Data Provider로 재지정합니다.

**-h | -?**

도움말 정보를 표시합니다.

---

## 제 261 장 db2sync - DB2 동기 프로그램 시작

Satellite의 초기 구성 및 구성에 대한 변경을 용이하게 합니다. 또한 이 명령을 사용하여 동기화의 진행을 시작, 중지 및 모니터링하고 제어 서버로 Satellite의 구성 정보(예: 통신 매개변수)를 업로드할 수 있습니다.

### 권한 부여

없음

### 필수 연결

없음

### 명령 구문

```
▶▶ db2sync [options]
           --t
           --s application_version
           --g
```

### 명령 매개변수

- t 관리자가 Satellite에 대한 응용프로그램 버전 또는 동기화 증명서를 변경할 수 있는 그래픽 사용자 인터페이스를 표시합니다.
- s *application\_version*  
Satellite에 응용프로그램 버전을 설정합니다.
- g 현재 Satellite에 설정된 응용프로그램 버전을 표시합니다.



---

## 제 262 장 db2sysstray - DB2 시스템 트레이 시작

DB2 시스템 트레이 도구를 시작합니다. Windows 운영 체제의 DB2 데이터베이스 서비스 상태를 모니터링하는 Windows 운영 체제 통지 아이콘입니다. db2sysstray는 서비스가 시작 및 중지되는 시점을 시각적으로 보여주며 서비스 시작 및 중지 기능을 제공합니다. 또한 DB2 제어 센터의 시작점을 제공합니다.

db2sysstray 아이콘에는 시작됨과 중지됨 모드가 있습니다. 모니터링한 인스턴스가 중지되면 아이콘에 빨간색 사각형의 오버레이가 포함됩니다. 인스턴스가 시작되면 빨간색 사각형이 사라집니다.

파티션된 데이터베이스 환경의 경우 db2sysstray 아이콘은 모든 파티션이 시작되어야 시작됨 모드가 됩니다. 하나 이상의 파티션이 중지되면 db2sysstray 아이콘이 중지됨 모드가 됩니다.

여러 DB2 사본이 단일 Windows 운영 체제에 설치되는 경우 db2sysstray는 설치된 각 DB2 사본의 DB2 인스턴스를 모니터링할 수 있습니다. 디폴트가 아닌 DB2 사본을 모니터링하려면 모니터링할 DB2 사본의 SQLLIB/bin에서 db2sysstray.exe 응용프로그램을 실행할 수 있습니다.

단일 DB2 인스턴스 또는 여러 인스턴스를 동시에 모니터링할 수 있습니다. 여러 db2sysstray 프로세스를 사용하여 여러 인스턴스를 모니터링할 수 있습니다. db2sysstray로 모니터링한 각 인스턴스마다 개별 아이콘이 시스템 트레이에 나타납니다. 각 아이콘 위에 마우스를 이동시키면 모니터링할 DB2 사본의 이름과 해당 db2sysstray 아이콘이 모니터링한 DB2 인스턴스 이름이 차례로 표시됩니다.

db2sysstray 아이콘은 db2sysstray 명령을 실행하여 DB2 명령 창에서 수동으로 또는 Windows 운영 체제가 시작될 때 자동으로 시작될 수 있습니다. db2sysstray는 DB2 데이터베이스를 설치할 때 자동으로 시작되도록 구성됩니다. 그러나 시스템이 시작될 때 db2sysstray가 자동으로 시작되도록 구성하더라도 DB2 서비스 또한 시작되는 것을 의미하지는 않습니다. 단지 DB2 데이터베이스 상태 모니터링이 자동으로 시작됨을 의미합니다.

db2sysstray 프로세스를 실행하여 모니터링되는 인스턴스에 대해 db2idrop 명령을 실행하면 db2sysstray 응용프로그램이 해당 레지스트리 항목을 강제로 제거하고 종료됩니다.

db2sysstray는 Windows 플랫폼에서만 사용할 수 있습니다.

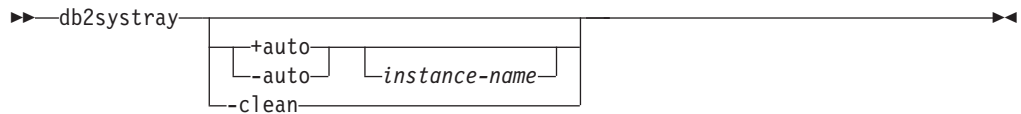
## 권한 부여

db2systray를 시작하는 데 필요한 특수 권한은 없습니다. 조치를 수행하려면 적합한 권한이 필요합니다.

## 필수 연결

없음

## 명령 구문



## 명령 매개변수

**+auto** Windows 운영 체제가 시작될 때 지정된 인스턴스에 대해 db2systray를 자동으로 시작합니다. db2systray는 또한 Launch Tool at Startup db2systray 메뉴 옵션을 사용하여 자동으로 시작되도록 구성될 수 있습니다.

**-auto** Windows 운영 체제가 시작될 때 지정된 인스턴스에 대해 db2systray가 자동으로 시작되지 않도록 사용 불가능하게 합니다.

### *instance-name*

모니터할 DB2 인스턴스의 이름. 인스턴스 이름이 지정되지 않으면 db2systray가 디폴트 로컬 DB2 인스턴스를 모니터링합니다. 인스턴스가 존재하지 않거나 지정된 인스턴스를 찾을 수 없는 경우 db2systray가 자동으로 종료됩니다.

**-clean** db2systray가 모니터링하는 모든 DB2 인스턴스의 레지스트리 항목을 제거하고 실행 중인 모든 db2systray.exe 프로세스를 중지합니다.

## 예:

1. C:\SQLLIB\bin> db2systray

DB2INSTANCE 환경 변수로 지정한 디폴트 DB2 인스턴스에 대해 db2systray를 시작합니다.

2. C:\SQLLIB\bin#> db2systray DB2INST1

DB2INST1 인스턴스에 대해 db2systray를 시작합니다.

3. C:\SQLLIB\bin#> db2systray +auto

디폴트 DB2 인스턴스에 대해 db2systray를 시작하고, Windows 운영 체제가 시작될 때 이 인스턴스 모니터링을 자동으로 시작하도록 db2systray를 구성합니다.

4. C:\SQLLIB\bin#> db2systray +auto DB2INST1

DB2INST1 인스턴스에 대해 db2sysstray를 시작하고, Windows 운영 체제가 시작될 때 이 인스턴스 모니터링을 자동으로 시작하도록 db2sysstray를 구성합니다.

5. C:\SQLLIB\bin#> db2sysstray -auto

DB2INSTANCE 환경 변수로 정의된 디폴트 인스턴스에 대한 자동 시작 옵션을 사용 불가능하게 합니다.

6. C:\SQLLIB\bin#> db2sysstray -auto DB2INST1

DB2INST1 인스턴스에 대한 자동 시작 옵션을 사용 불가능하게 합니다.

7. C:\SQLLIB\bin#> db2sysstray -clean

db2sysstray로 작성한 모든 레지스트리 항목을 제거하고 실행 중인 db2sysstray.exe 프로세스를 모두 중지합니다. 설치된 다른 DB2 사본에 대해 db2sysstray.exe 프로세스가 실행되는 경우 사본이 제거되지 않습니다. 제거할 각 DB2 사본에 대해 SQLLIB/bin에서 db2sysstray -clean을 실행해야 합니다.





## 제 263 장 db2tapemgr - 테이프의 로그 파일 관리

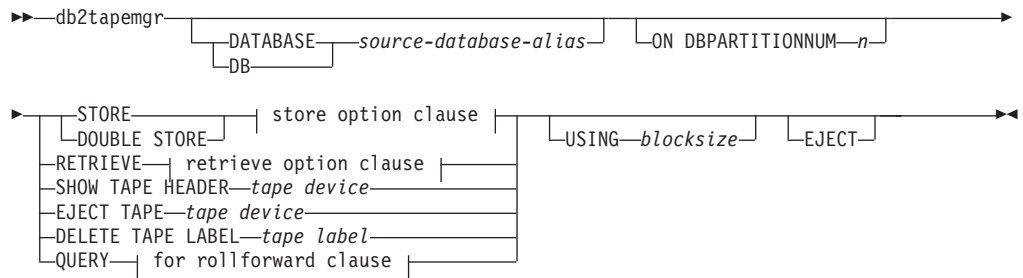
테이프에 대한 DB2 로그 파일의 저장 및 검색을 허용합니다. 테이프상의 위치는 실행 기록 파일에 저장됩니다.

### 권한 부여

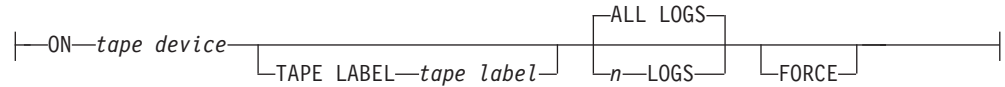
다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*

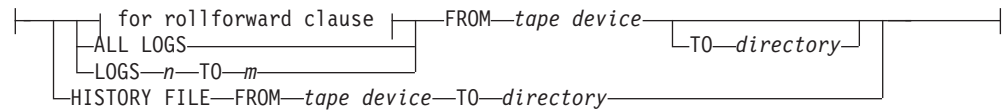
### 명령 구문



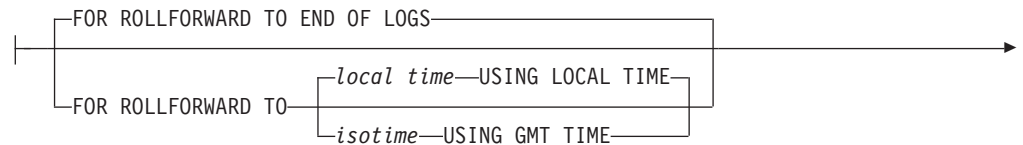
#### store option clause:



#### retrieve option clause:



#### for rollforward clause:



## 명령 매개변수

### **DATABASE** *source-database-alias*

데이터베이스 이름을 지정합니다. 값이 지정되지 않으면 DB2DBDFT가 사용됩니다. 값이 지정되지 않고 DB2DBDFT가 설정되지 않으면 조작이 실패합니다.

### **ON DBPARTITIONNUM**

실행할 데이터베이스 파티션 번호를 지정합니다. 값이 지정되지 않으면 DB2NODE가 사용됩니다.

### **STORE ON** *tape device*

로그 파일을 테이프에 저장하고 삭제합니다.

### **DOUBLE STORE ON** *tape device*

한 번만 저장된 모든 로그 파일과 저장되지 않은 로그 파일을 저장합니다. 테이프에 두 번 저장된 로그 파일만 삭제하고 나머지는 디스크에 보존됩니다.

### **TAPE LABEL**

테이프에 붙일 레이블을 지정합니다. *tape label*이 지정되지 않으면 *database-alias!timestamp* 형식으로 자동 생성됩니다(최대 22자, 데이터베이스 별명은 최대 8자, 시간소인(초)은 최대 14자).

### **ALL LOGS** 또는 *n LOGS*

모든 로그 또는 지정된 수의 로그에 명령이 적용되는 것으로 지정합니다.

### **FORCE**

테이프가 만기되지 않으면 겹쳐쓰는 것으로 지정합니다.

### **USING** *blocksize*

테이프 액세스를 위한 블록 크기를 지정합니다. 디폴트 크기는 5120이며 512의 배수여야 합니다. 최소값은 512입니다.

### **EJECT**

조작 완료 후 테이프를 꺼내는 것으로 지정합니다.

### **RETRIEVE FOR ROLLFORWARD TO**

유틸리티가 지정된 롤 포워드에 필요한 모든 로그에 대한 프롬프트를 대화식으로 표시하고 테이프에서 검색하는 것으로 지정합니다. 디렉토리가 지정되지 않으면 **overflowlogpath** 구성 매개변수로 지정한 경로가 사용됩니다. 디렉토리가 지정되지 않고 **overflowlogpath**가 설정되지 않으면 조작이 실패합니다.

### **END OF LOGS**

로그 끝까지의 로그 파일이 검색되는 것으로 지정합니다.

*isotime* **USING GMT TIME**

지정된 시간까지의 로그 파일이 검색되는 것으로 지정합니다.

*local time* **USING LOCAL TIME**

지정된 시간까지의 로그 파일이 검색되는 것으로 지정합니다.

**USING HISTORY FILE** *history file*

사용될 대체 실행기록 파일을 지정합니다.

**FROM** *tape device*

로그 파일을 검색할 테이프 디바이스를 지정합니다.

**TO** *directory*

검색된 로그 파일을 복사할 디렉토리를 지정합니다.

**RETRIEVE ALL LOGS** 또는 **LOGS** *n TO m*

명령이 테이프에서 지정된 수의 로그 또는 모든 로그에 적용되는 것으로 지정합니다.

**FROM** *tape device*

로그 파일을 검색할 테이프 디바이스를 지정합니다.

**TO** *directory*

검색된 로그 파일을 복사할 디렉토리를 지정합니다.

**RETRIEVE HISTORY FILE**

실행기록 파일을 검색합니다.

**FROM** *tape device*

로그 파일을 검색할 테이프 디바이스를 지정합니다.

**TO** *directory*

검색된 로그 파일을 복사할 디렉토리를 지정합니다.

**SHOW TAPE HEADER** *tape device*

테이프 헤더 파일 DB2TAPEMGR.HEADER의 콘텐츠를 나타냅니다.

**EJECT TAPE** *tape device*

테이프를 꺼냅니다.

**DELETE TAPE LABEL** *tape label*

실행기록 파일에서 지정된 테이프 레이블을 참조하는 모든 위치를 삭제합니다.

**QUERY FOR ROLLFORWARD TO**

롤 포워드에서 필요한 로그 파일의 위치를 표시합니다.

**END OF LOGS**

*isotime* **USING GMT TIME**

조작이 지정된 시간까지 로그를 쿼리해야 하는 것으로 지정합니다.

*local time* **USING LOCAL TIME**

조작이 지정된 시간까지 로그를 쿼리해야 하는 것으로 지정합니다.

**USING HISTORY FILE** *history file*

사용될 대체 실행기록 파일을 지정합니다.

---

## 제 264 장 db2tbst - 테이블 스페이스 상태 확보

16진 테이블 스페이스 상태 값을 승인하고 상태를 리턴합니다. 상태 값은 LIST TABLESPACES 출력의 일부입니다.

권한 부여

없음

필수 연결

없음

명령 구문

▶—db2tbst—*tablespace-state*—————▶

명령 매개변수

*tablespace-state*

16진 테이블 스페이스 상태 값.

예:

db2tbst 0x0000 요청은 다음 출력을 생성합니다.

State = Normal



---

## 제 265 장 db2tdbmgr - 도구 카탈로그 데이터베이스 이주 명령

db2tdbmgr 명령은 도구 카탈로그 데이터베이스에서 UPGRADE DATABASE 명령을 실행한 후 특정 도구 카탈로그 데이터베이스 오브젝트를 이주합니다.

### 권한 부여

*sysadm*

### 필수 연결

이 명령은 데이터베이스 연결을 설정합니다.

### 명령 구문

```
▶▶ db2tdbmgr -d db_name -s schema_name -u user_name  
-p password
```

### 명령 매개변수

**-d** *db\_name*

도구 카탈로그 데이터베이스 이름.

**-s** *schema\_name*

도구 카탈로그 스키마 이름.

**-u** *user\_name*

도구 카탈로그 데이터베이스를 연결하는 데 사용되는 사용자 이름

**-p** *password*

도구 카탈로그 데이터베이스에 연결하는 데 사용되는 암호

### 예:

다음 예에서는 toolsdb 데이터베이스 별명과 systools 스키마로 도구 카탈로그 테이블을 이주합니다.

```
db2tdbmgr -d toolsdb -s systools -u db2inst1 -p *****
```

### 사용 시 참고사항

이 명령은 도구 카탈로그 테이블을 최신 버전으로만 이주하므로 이주된 도구 카탈로그 테이블을 이전 버전으로 변환하는 데 사용할 수 없습니다.

데이터베이스는 이주 전에 카탈로그되어야 합니다. 대부분의 경우 이주 오류 메시지에 오류 위치가 명확하게 나타납니다. 오류 메시지에 테이블 또는 컬럼 이름과 같은 오브젝트에 대한 문제가 나타나는 경우 보고된 오브젝트가 손상되었거나 이주를 위해 제출된 데이터베이스 이름에서 누락되었을 수 있습니다.



## 제 266 장 db2trc - 추적

db2trc는 DB2 인스턴스 또는 DAS(DB2 Administration Server)의 추적 기능을 제어합니다. 추적 기능은 작업에 관한 정보를 기록하고 이 정보를 읽을 수 있는 양식으로 형식화합니다. 추적 기능을 사용하면(디폴트로 OFF임) 시스템 성능에 영향을 줄 수 있습니다. 결과적으로, DB2 기술 지원부 담당자가 지시할 때만 추적 기능을 사용하십시오. 그렇지 않으면 충분한 정보가 기록된 후 추적을 끄십시오.

DB2 추적은 특히 되풀이 발생하고 재생 가능한 문제를 분석할 때 유용한데, 문제점 판별의 지원 담당자 작업을 수월하게 합니다.

DB2를 사용할 때, 때때로 "추적을 시작하고 IBM Support에 문의하거나", "추적을 끄고 추적 레코드를 조사하거나", 또는 "문제점 판별, SQLCODE, SQLCA 콘텐츠(가능한 경우) 및 추적 파일(가능한 경우) 정보를 준비하고 기술 지원부 담당자에게 문의"할 것을 지시하는 오류 메시지를 볼 수 있습니다. 또는 IBM Support에 문제점을 보고할 때 사용자에게 추적을 수행하여 환경에 대한 상세한 정보를 캡처할 것을 요청할 수 있습니다.

### 권한 부여

UNIX 운영 체제에서 DB2 인스턴스를 추적하려면 다음 권한 중 하나가 있어야 합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*

UNIX 운영 체제에서 DB2 Administration Server를 추적하려면 다음 권한 중 하나가 있어야 합니다.

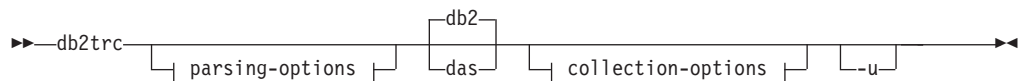
- *dasadm*

Windows 운영 체제에서는 권한이 필요하지 않습니다.

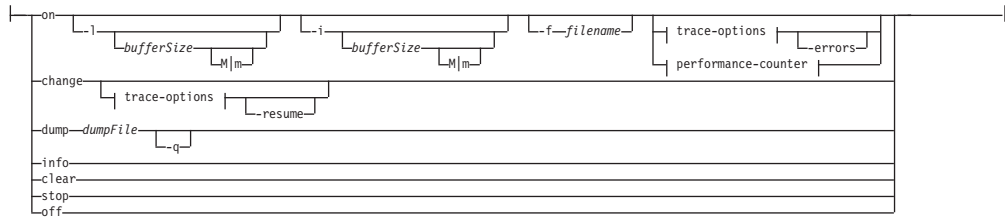
### 필수 연결

없음

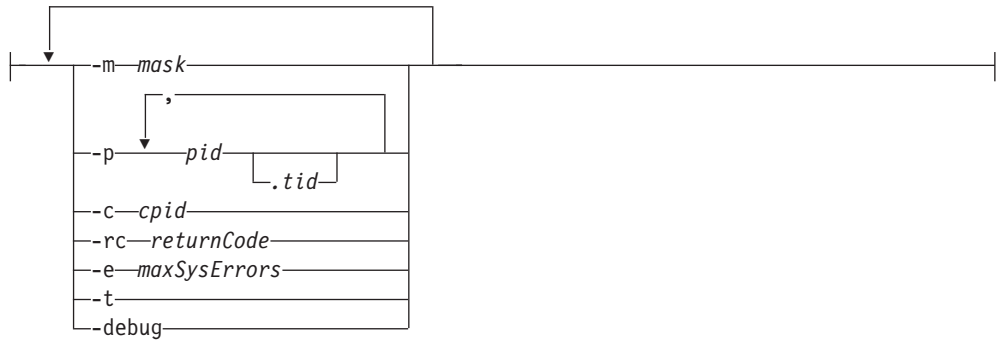
### 명령 구문



콜렉션 옵션:



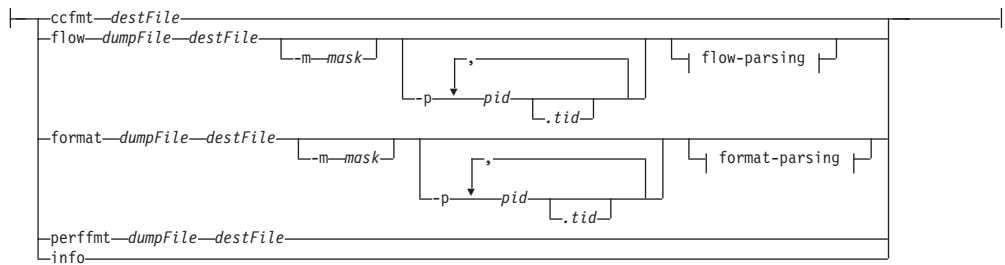
추적 옵션:



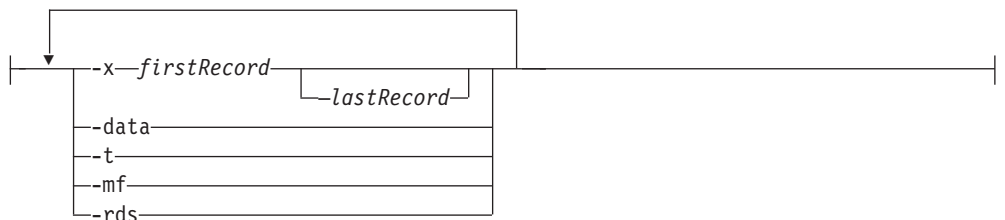
성능 카운터:



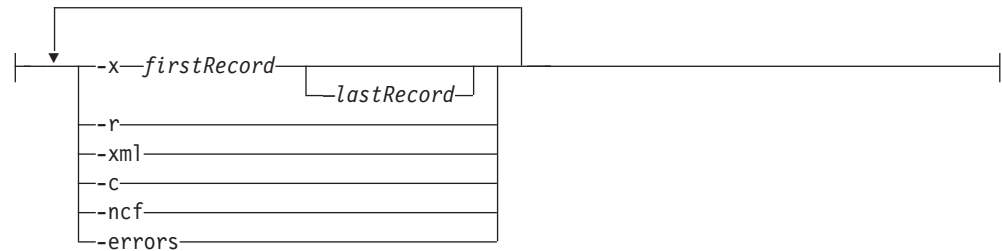
구문 분석 옵션:



플로우 구문 분석:



형식 구문 분석:



## 명령 매개변수

- db2** 모든 추적 작업이 DB2 인스턴스에서 수행되도록 지정합니다. 이는 디폴트값입니다.
- das** 모든 추적 작업이 DB2 Administration Server 인스턴스에서 수행되도록 지정합니다.
- on** 추적 기능을 시작하려면 이 매개변수를 사용하십시오. 매개변수 목록에 대해서는 아래 공유 추적 옵션 섹션을 참조하십시오.

### -l [bufferSize]

이 옵션은 추적 버퍼의 크기 및 동작을 지정합니다. -l은 최종 추적 레코드가 보유되도록 지정합니다(즉, 버퍼가 가득 찰 때 첫 번째 레코드가 겹쳐져짐). 버퍼 크기는 바이트 또는 MB 단위로 지정할 수 있습니다. 버퍼 크기를 MB 단위로 지정하려면 버퍼 크기에 문자 M | m을 추가하십시오. 예를 들어 4MB 버퍼를 갖고 db2trc를 시작하려면 다음을 수행하십시오.

```
db2trc on -l 4m
```

디폴트 및 최대 추적 버퍼 크기는 플랫폼에 따라 다릅니다. 최소 버퍼 크기는 1MB입니다. 버퍼 크기는 2의 멱급수여야 합니다.

### -i [bufferSize]

이 옵션은 추적 버퍼의 크기 및 동작을 지정합니다. -i는 초기 추적 레코드가 보유되도록 지정합니다(즉, 버퍼가 한 번 가득 차면 추가 레코드가 기록되지 않음). 버퍼 크기는 바이트 또는 MB 단위로 지정할 수 있습니다. 버퍼 크기를 MB 단위로 지정하려면 버퍼 크기에 문자 M | m을 추가하십시오.

### -f filename

파일로 추적할 때 완전한 파일 이름을 지정해야 하며 -f 옵션과 함께 -l 또는 -i를 사용하는 경우 bufferSize 값이 디스크의 파일 크기를 제한합니다. -l은 최종 추적 레코드를 보존하며 파일 안에서 래핑하도록 허용됩니다. -i는 초기 추적 레코드를 보존하고 파일 크기 한계에 도달

할 때 추적을 중지합니다. 파일 크기를 MB로 지정하려면 `-i` 및/또는 `-i buffersize`에 대해 지정된 값 뒤에 문자 `M` | `m`, `GB`의 경우 문자 `G` | `g`를 추가하십시오.

#### **-errors**

기능 종료 시 오류 및 0이 아닌 리턴 코드만 추적합니다. 이 옵션은 `-debug` 또는 `-perfcount` 옵션과 동시에 지정할 수 없습니다.

#### **change**

이 콜렉션 옵션을 사용하면 적용되는 추적 옵션을 변경할 수 있습니다. 매개변수 목록에 대해서는 아래 공유 추적 옵션 섹션을 참조하십시오.

#### **-resume**

이 옵션으로 일시중단된 프로세스의 실행을 다시 시작할 수 있습니다. `-suspend`가 사용 가능하지 않은 경우 다시 시작할 수 없습니다.

#### **공유 추적 옵션**

`on`과 `change` 사이에 공유되는 일반 추적 옵션.

#### **-m mask**

수집 또는 형식화된 데이터 양을 줄입니다. 추적 마스크는 다음 형식을 갖습니다. `types.products.components.functions.categories`

마스크에 대한 값은 IBM Support가 제공합니다.

마스크는 5개 파트(추적 레코드 유형, 제품, 구성요소, 기능 및 기능 범주)로 구성됩니다. 각 파트는 쉼표로 구분된 목록, 하이픈으로 구분된 범위 또는 단일 항목으로 구성될 수 있습니다. 별표(\*)를 사용하면 모든 것과 일치할 수 있습니다. 필드 값은 이름 또는 대응하는 번호로 지정할 수 있습니다. 전체 형식의 제품, 구성요소 또는 기능 파트의 이름을 지정하는 마스크의 짧은 형식을 사용할 수 있습니다. 마스크를 `"*.*.*.*"`로 지정하는 것은 마스크를 지정하지 않는 것과 같습니다.

예: `-m *.*.SQL0,SQL.*.entry,exit`

#### **-p pid [.tid]**

지정된 프로세스 ID(`pid`) 및 스레드 ID(`tid`)에 대해서만 추적 기능을 사용 가능하게 합니다. 마침표(.)는 `tid`가 지정되는 경우 포함되어야 합니다. 최대 5개의 `pid.tid` 조합이 지원됩니다.

예를 들어 프로세스 10, 20 및 30에 대해 추적을 사용하려면 구문은 다음과 같습니다.

```
db2trc on -p 10,20,30
```

프로세스 100의 스레드 33 및 프로세스 200의 스레드 66에 대해서만 추적을 사용하려면 구문은 다음과 같습니다.

```
db2trc on -p 100.33,200.66
```

**-c cpid**

이 보조 프로세스만 추적 또는 형식화합니다.

**-rc returnCode**

*returnCode*를 시스템 오류로 취급합니다. *returnCode*는 부호 표시 정수로 지정해야 합니다.

**-e maxSysErrors**

*maxSysErrors* 시스템 오류가 발생한 후 추적을 중지합니다.

**-t** 시간소인을 포함합니다.

**-debug**

이것은 IBM Support의 디버깅을 위해 사용되는 내부 옵션입니다. 사용은 권장하지 않습니다.

**info** 다음은 이 매개변수로 나열되는 환경 정보의 예입니다.

```
D:\Program Files\IBM\SQLLIB\BIN>db2trc info
Marker                : @TRACE@
Trace version         :      7.0
Platform              :      NT
Build level           : s060629
maxBufferSize         : 2097152 bytes (2 MB)
auxBufferSize         : 6291456 bytes (6 MB)
allocationCount       : 1
DB2TRCD pid          : 2384
DB2TRCD64 pid         : 0
Trace destination     : <shared memory buffer>
debug                 : disabled
debug runtime passno  : 0
numSuspended         : 0

Buffer size           : 2097152 bytes (2 MB)
Allow buffer to wrap  : yes
Mask                  : *.*.*.*.*
Timestamps            : enabled
PID.TID mask         : all
Fixed data mask #1    : all
Fixed data mask #2    : all
Max system errors     : infinite
Treat this rc as sys err: none
```

**dump dumpFile**

버퍼에 저장된 2진 형식 추적 정보를 파일로 덤프합니다. 다음 명령은 해당 정보를 현재 디렉토리의 db2trc.dmp라는 파일에 넣습니다.

```
db2trc dump db2trc.dmp
```

이 매개변수와 함께 덤프 파일 이름을 지정하십시오. 경로를 명시적으로 지정하지 않으면 2진 형식 덤프 파일은 현재 디렉토리에 저장됩니다.

**-q** 정숙 모드.

**ccfmt** *destFile*

코드 범위 추적을 덤프하고 형식화합니다. 덤프에 대한 대상 파일 이름을 지정하십시오.

**flow** *dumpFile destFile*

추적이 실행 파일로 덤프된 후 읽을 수 있는 텍스트 파일로 형식화하십시오. flow 옵션을 사용하여 프로세스 또는 스레드에 의해 정렬된 레코드를 형식화하십시오. 덤프 파일의 이름 및 생성될 대상 파일의 이름을 지정하십시오. 예를 들면, 다음과 같습니다.

```
db2trc flow db2trc.dmp db2trc.flw
```

**-x** *firstRecord* [**—***lastRecord*]

*firstRecord*부터 *lastRecord*까지의 레코드 번호만 표시합니다.

**-data** 순서에 모든 추적 레코드 데이터를 포함합니다.

**-t** 사용 가능한 경우 시간소인을(sec:nsec 형식으로) 포함합니다.

**-mf** 각 고유 순서에 대해 개별 대상 파일을 생성합니다.

**-rds** 사용 가능한 경우 RDS 연산자 정보를 포함합니다.

**format** *dumpFile destFile*

추적이 실행 파일로 덤프된 후 읽을 수 있는 텍스트 파일로 형식화하십시오. format 옵션을 사용하여 레코드를 연대기순으로 형식화하십시오.

**-x** *firstRecord* [**—***lastRecord*]

*firstRecord*부터 *lastRecord*까지의 레코드 번호만 표시합니다.

**-r** 역방향 순서로 출력합니다.

**-xml** xml 구문 분석 가능 형식으로 데이터를 출력합니다.

**-c** 통신 버퍼를 형식화합니다.

**-ncf** 구성요소 사용자 정의 포매팅을 사용하지 않습니다.

**-errors**

기능 종료 시 오류 및 0이 아닌 리턴 코드만 추적합니다.

**perffmt** *dumpFile destFile*

성능 추적 포맷터는 성능 카운터가 들어있는 덤프 파일을 읽을 수 있는 텍스트로 형식화하는 구문 분석 옵션입니다.

**clear** 특히 특정 데이터베이스에 연결하기 직전에 추적 버퍼의 콘텐츠를 지웁니다. 이 옵션을 사용하면 원하는 데이터베이스에 연결하기 전에 누적된 불필요한 정보의 버퍼를 지워서 수집된 정보의 양을 줄일 수 있습니다.

**stop** 이 콜렉션 옵션은 요구 시 추적을 중지합니다. 모든 프로세스가 추적을 일시중단하지만 추적 버퍼의 콘텐츠는 보존되므로 나중에 덤프할 수 있습니다. 이 조치는 추적 기능을 함께 사용 안하는 off 옵션과는 대조적입니다.

**off** 추적 기능을 사용하지 않습니다. 추적이 파일로 덤프된 후 다음을 입력하여 추적 기능을 사용 불가능하게 하십시오.

```
db2trc off
```

**-u** 대부분의 명령행 옵션에 관한 추가 정보를 제공합니다. 명령행 항목의 일반 양식은 아래의 사용 시 참고사항에 표시되어 있습니다. 다음은 DAS 인스턴스에 대한 덤프 명령의 자세한 정보를 얻는 예입니다.

```
db2trc das dump -u
```

## 사용 시 참고사항

추적을 수행하는 과정에서 db2trc 명령을 여러 번 발행해야 합니다. DB2 인스턴스가 중지되었을 때, 일반적인 순서는 먼저 추적을 켜는(on) 것인데, DB2 인스턴스가 시작된 후 지정된 데이터의 콜렉션 및 버퍼에 저장을 즉시 시작합니다. 그 다음에는 데이터베이스에 연결하기 전에 버퍼를 지우고, 2진 형식 데이터를 덤프 파일로 덤프하는 것입니다. 그 다음, 추적을 끄고(off) 마지막으로 덤프 파일을 쉽게 읽을 수 있는 텍스트 대상 파일로 형식화하는 것입니다. 다음은 dmp 파일에 기록된 추적 버퍼의 콘텐츠를 사용하여 SAMPLE 데이터베이스의 추적을 수행하기 위해 실행되는 명령의 예입니다.

```
db2trc on -i 8m -m " *.*.2.*.*" -t
db2start
db2trc clear
db2 connect to sample
db2trc dump dmp
db2trc off
```

db2trc 명령의 일반 구문은 아래에 표시됩니다. 명령 옵션은 콜렉션 및 구문 분석의 두 가지 스테이지로 그룹화할 수 있습니다.

- 콜렉션 옵션에는 추적 켜기 또는 끄기, 추적 버퍼 크기 지정, 추적 옵션 지정 또는 변경, 추적 덤프 및 추적 버퍼 지우기가 포함됩니다.
- 구문 분석 옵션에는 형식화된 추적 레코드를 연대기순으로, 프로세스별로 또는 스레드별로 정렬이 포함됩니다.

단계 #1 - 콜렉션

사용법: db2trc [facility] <command> [-u]

[facility]

db2 - DB2 인스턴스(디폴트)

das - DB2 Administration Server 인스턴스

<command>

change - 추적 옵션 변경

clear - 추적 버퍼 지우기

dump - 추적 덤프 파일 생성

info - 정보

off - 추적 기능 사용 안함

on - 추적 기능 사용

stop - 추적 중지

## STAGE #2 - PARSING

사용법: db2trc <command> [-u]

<command>

- ccfmt - 코드 범위 추적 덤프 및 형식화
- flow - 제어 순서 다이어그램 생성
- format - 포맷
- info - 정보
- perffmt - 성능 추적 포맷

자세한 정보는 위의 명령 중 어디에나 "-u" 옵션을 추가하십시오.

위의 Stage #2 - Parsing 섹션에서 ccfmt 명령은 "코드 범위 추적"을 덤프하고 형식화합니다. 코드 범위 추적은 함수 항목, 종료, 프로브 포인트 및 코드 경로의 계수를 보존하는 db2trc의 확장입니다. 어떤 기능이 많이 사용되고 있는지 또는 어떤 기능이 테스트 중에 사용되지 않고 있는지에 대한 통계를 수집하는 데 사용할 수 있습니다.

데이터베이스 서버를 추적할 때는 데이터베이스 관리 프로그램을 시작하기 전에 추적 기능을 켜는 것이 좋습니다. 이것은 UNIX 및 Linux 플랫폼에서 실행 중인 데이터베이스 관리 프로그램이 추적 변경을 즉시 인식하기 위해 가장 믿을 수 있는 메소드입니다.

추적을 켜고 DB2 텍스트 검색에 관한 정보를 수신하기 위해 cie(155)에 대한 구성요소 코드를 갖는 마스크를 사용할 수 있습니다.

```
db2trc on -m "*.*.155.*.*"
```



## 제 267 장 db2unins - DB2 데이터베이스 제품, 기능 또는 언어 설치 제거

하나 이상의 DB2 데이터베이스 제품, 기능 또는 언어를 설치 제거합니다. db2unins는 설치 미디어와 시스템의 DB2 설치 사본에도 있습니다. 설치 미디어에서 실행하는 경우 -f, -l, -t 및 -? 매개변수만 사용할 수 있습니다. DB2 설치 사본에서 실행하는 경우 -f를 제외한 모든 옵션을 사용할 수 있습니다.

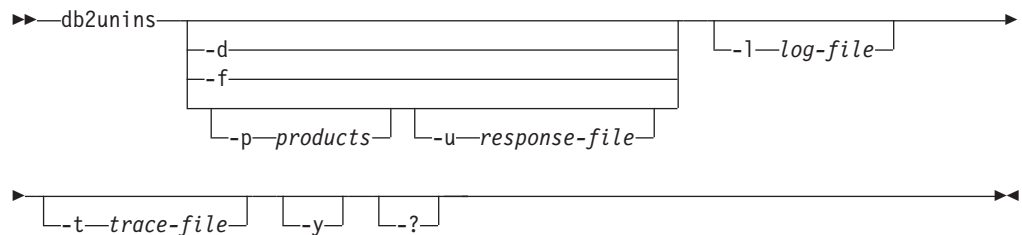
### 권한 부여

*sysadm*

### 필수 연결

없음

### 명령 구문



### 명령 매개변수

-, -d, -p 또는 -u 매개변수 없이 db2unins 명령을 실행하면 현재 설치 디렉토리에 있는 모든 DB2 데이터베이스 제품이 제거됩니다.

**-d** 시스템의 현재 DB2 사본에 설치된 제품을 표시합니다. 이 옵션은 DB2 데이터베이스 제품의 설치된 사본에서 실행된 경우에만 사용할 수 있습니다.

**-f** 시스템에 있는 모든 DB2 데이터베이스 제품의 강제 설치 제거를 수행합니다. db2unins -f 명령은 설치 미디어에서만 실행될 수 있습니다. db2unins -f 명령 실행에 성공하면 시스템이 재부트됩니다. 이 명령은 시스템에 버전 9 이전의 다른 DB2 제품이 설치되지 않은 경우에만 실행될 수 있습니다.

#### **-p products**

설치 제거될 제품을 지정합니다. 이 매개변수는 설치된 DB2 사본에서 실행하는 경우에만 사용할 수 있습니다. 여러 제품을 설치 제거하려면 제품 목록을 세미콜론으로 구분하고 큰따옴표로 묶습니다. -p 및 -u 매개변수가 모두 지정된 경우 -p에 지정된 제품은 응답 파일에 지정된 제품을 겹쳐줍니다. 예를 들어,

db2unins -p "ESE;QP" -u db2un.rsp는 db2un.rsp의 REMOVE\_PROD 키워드 값에 관계 없이 DB2 ESE 및 QP를 모두 설치 제거합니다.

**-u response-file**

응답 파일에 지정된 내용을 기반으로 제품, 기능 또는 언어의 설치 제거를 수행합니다(예: db2unins -u db2un.rsp). 이 매개변수는 설치된 DB2 사본에서 실행하는 경우에만 사용할 수 있습니다. -p 및 -u 매개변수가 모두 지정된 경우 -p 매개변수에 지정된 DB2 제품은 응답 파일의 REMOVE\_PROD 키워드를 겹쳐씹니다.

클러스터 환경에서는 응답 파일을 사용하여 DB2 제품을 설치 제거하기 전에 먼저 장애 복구 인프라스트럭처를 작성하기 위해 db2mscs 명령을 처음 실행한 서버에서 db2mscs 명령과 -u 옵션을 함께 실행해야 합니다. 자세한 내용은 db2mscs 명령을 참조하십시오.

**-l log-file**

로그 파일의 위치를 지정합니다. 디폴트 로그 파일 위치는 My Documents\DB2LOG\db2un\_<timestamp>.log입니다.

**-t trace-file**

추적 기능을 활성화합니다. 추적 파일은 db2unins 명령으로 문제점을 디버깅하는 데 사용됩니다.

**-y** 설치 제거 프로세스에서 확인이 수행되지 않는지 확인합니다.

**-?** db2unins 명령의 도움말을 표시합니다.

## 사용 시 참고사항

db2unins -f를 사용하여 시스템의 모든 DB2 데이터베이스 제품을 수동으로 제거하려는 경우 시스템의 최상위 DB2 제품 버전과 같은 버전에서 유틸리티를 사용해야 합니다. 예를 들어, 두 개의 사본이 설치되어 있는 경우( DB2 V9.1 DB2COPY1과 DB2 V9.5 DB2COPY2) db2unins -f를 실행하여 두 DB2 버전을 모두 DB2 V9.5 제품 이미지에서 제거합니다. DB2 V9.1 제품 이미지에서 db2unins -f를 실행하는 경우 머신을 완전히 정리하지 않습니다.

---

## 제 268 장 db2untag - 컨테이너 태그 릴리스

테이블 스페이스 컨테이너의 DB2 태그를 제거합니다. 이 태그는 DB2가 둘 이상의 테이블 스페이스에서 컨테이너를 재사용하지 못하게 하는 데 사용됩니다. 컨테이너가 연관되는 데이터베이스를 식별하는 컨테이너 태그에 관한 정보를 표시합니다. 삭제된 이후 데이터베이스가 마지막 사용한 컨테이너를 릴리스해야 할 때 유용합니다. 태그가 남아 있으면 DB2는 나중에 해당 자원을 사용할 수 없습니다.

**경고:** 이 도구는 정보가 알려진 시스템 관리자만 사용해야 합니다.

### 권한 부여

사용자는 데이터베이스를 작성한 ID가 소유하는 테이블 스페이스에 대한 컨테이너에 대한 읽기/쓰기 액세스가 필요합니다.

### 필수 연결

없음

### 명령 구문

```
▶▶—db2untag—-f—filename—————▶▶
```

### 명령 매개변수

**-f** *filename*

DB2 태그를 제거할 테이블 스페이스 컨테이너의 완전한 이름을 지정합니다.

### 사용 시 참고사항

SQLCODE -294(컨테이너 사용 중 오류)가 가끔 데이터베이스 작성 또는 테이블 스페이스 작성 또는 변경 작업에서 리턴되며, 이는 대개 컨테이너가 다른 테이블 스페이스에 의해 이미 사용 중일 때 운영 체제 자원 이름의 지정 오류를 표시합니다. 한 번에 하나의 테이블 스페이스만 컨테이너를 사용할 수 있습니다.

컨테이너를 마지막으로 사용한 데이터베이스가 삭제되었음을 확인하는 시스템 또는 데이터베이스 관리자는 컨테이너의 태그가 제거되지 않은 경우 db2untag 도구를 사용할 수 있습니다. 컨테이너를 릴리스해야 하는 경우 다음 중 하나를 수행하십시오.

- SMS 컨테이너의 경우 적합한 삭제 명령을 사용하여 디렉토리 및 해당 콘텐츠를 제거하십시오.

- DMS 원시 컨테이너의 경우 파일 또는 디바이스를 삭제하거나 db2untag를 사용하여 컨테이너 태그를 제거하십시오. 그렇지 않으면 도구는 그러한 DMS 컨테이너를 수정되지 않은 채로 둡니다.

---

## 제 269 장 db2updserv - 제품 갱신사항 표시

DB2 제품에 사용 가능한 제품 갱신사항 및 향상/개선된 기능을 표시합니다. Windows에서는 이 명령의 출력이 웹 페이지로 이동하고 UNIX에서는 Java 응용프로그램으로 이동합니다.

### 권한 부여

없음

### 필요한 연결

인터넷 연결 필요.

### 명령 구문

▶▶—db2updserv—————▶▶

### 명령 매개변수

없음



---

## 제 270 장 db2val - DB2 사본 유효성 확인 도구 명령

설치 파일, 인스턴스 설정 및 로컬 데이터베이스 연결의 상태를 확인하여 DB2 사본의 기본 기능을 검증합니다.

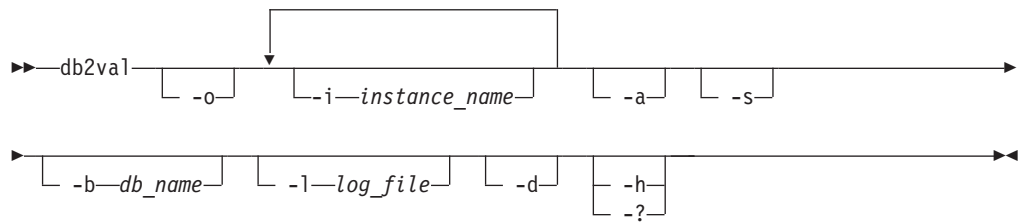
### 권한 부여

SYSADM

### 필요한 연결

없음

### 명령 구문



### 명령 매개변수

**-o** 설치 파일만 유효성을 확인하는 것으로 지정합니다. 인스턴스, 데이터베이스 및 확장 보안에 대한 유효성 확인은 수행되지 않습니다. 이 매개변수가 지정되면 `-i`, `-a`, `-b` 및 `-s` 매개변수가 무시됩니다.

#### **-i instance\_name**

유효성을 확인할 인스턴스의 이름을 지정합니다. 여러 인스턴스의 유효성을 확인하는 것으로 지정하려면 이 매개변수를 여러 번 지정합니다(예: `-i inst1 -i inst2`). Windows 운영 체제의 경우 이 매개변수가 지정되지 않으면 현재 인스턴스가 디폴트값으로 사용됩니다. Linux 및 UNIX 운영 체제의 경우 이 매개변수는 DB2 사본의 루트 설치에서만 루트 사용자가 사용할 수 있습니다.

**-a** DB2 사본의 모든 인스턴스의 유효성을 확인합니다. Linux 및 UNIX 운영 체제의 경우 이 매개변수는 DB2 사본의 루트 설치에서만 루트 사용자가 사용할 수 있습니다. 이 매개변수는 `-i` 매개변수를 겹쳐씹니다.

#### **-b <db\_name>**

지정된 데이터베이스에 대한 연결 및 데이터베이스 작성의 유효성을 확인합니다. DB2 클라이언트 인스턴스에 대해서는 이 매개변수가 무시됩니다.

- d Linux 및 UNIX 운영 체제에서만 유효합니다. 이 매개변수는 DB2 Support의 지시가 있는 경우에만 사용합니다. 디버그 모드를 활성화합니다.
- s DPF 환경의 일부인 지정된 인스턴스의 DB2 데이터베이스 관리 프로그램을 시작합니다.

**-l <log\_file>**

지정된 파일 이름에 로그를 작성합니다. -l 매개변수가 지정되지 않는 경우 Linux 및 UNIX 운영 체제의 디폴트 로그 경로는 /tmp/db2va1xx.log이고 Windows 운영 체제에서는 My Documents\DB2LOG\db2va1xx.log입니다. 여기서 xx는 생성된 값입니다.

**/? | -h**

db2va1 명령의 사용법 정보를 표시합니다.

**예:**

TEST1 인스턴스와 DATA1 데이터베이스의 유효성을 확인하려면 다음 명령을 실행하십시오.

```
db2va1 -i TEST1 -b DATA1
```

DB2 사본의 모든 인스턴스 유효성을 확인하려면 다음 명령을 실행하십시오.

```
db2va1 -a
```

DB2 설치 파일의 유효성만 확인하려면 다음 명령을 실행하십시오.

```
db2va1 -o
```



---

## 제 271 장 db2xdbmig - XSR 오브젝트 이주 명령

분석할 수 있는 모든 XML 스키마 저장소(XSR) 오브젝트를 명령을 실행하는 DB2 사본의 현재 버전 및 서비스 레벨로 이주합니다.

이 명령은 *DB2DIR/bin* 디렉토리에 있습니다. 여기서 *DB2DIR*은 DB2 데이터베이스 시스템의 현재 버전이 설치된 설치 위치를 나타냅니다.

### 권한 부여

데이터베이스의 모든 XSR 오브젝트에 대한 CREATE, ALTER 및 DROP 특권.

### 명령 구문

▶▶—db2xdbmig—*database-alias*—————▶▶

### 명령 매개변수

*database-alias*

XSR 오브젝트가 들어 있는 데이터베이스의 별명을 지정합니다.

### 사용 시 참고사항

- db2xdbmig 명령은 분석 가능 XML 스키마에만 영향을 줍니다.
- DB2 버전 9.1 GA 또는 FixPack 1 사본에서 DB2 버전 9.7로 이주하는 경우 UPGRADE DATABASE 명령은 내재적으로 db2xdbmig 명령을 실행합니다. DB2 버전 9.7에서는 이 명령을 실행하지 않아도 됩니다.



---

## 제 272 장 db2xpvt - 트랩 파일 형식화

DB2 데이터베이스 2진 트랩 파일을 사용자가 읽을 수 있는 ASCII 파일로 형식화합니다. 트랩 파일(\*.TRP)은 인스턴스 디렉토리(DB2INSTPROF)(디폴트값) 또는 진단 데이터 디렉토리 경로(DIAGPATH 데이터베이스 관리 프로그램 구성 매개변수가 설정된 경우)에 있습니다. 이 파일은 SQLLIB/BIN 디렉토리 아래에 있습니다. db2xpvt 명령은 트랩 파일을 형식화하기 위해 DB2 기호 파일(.PDB)을 사용합니다.

### 권한 부여

DIAGPATH 디렉토리에 대한 액세스 권한이 필요합니다.

### 명령 구문

```
▶▶ db2xpvt /p path /n infile outfile ▶▶
```

Diagram illustrating the command syntax: db2xpvt followed by a path specification ( /p path /n ) and two file names ( infile outfile ).

### 명령 매개변수

#### */p path*

실행 파일과 PDB 파일이 있는 하나 이상의 위치를 가리키는 세미콜론(;)으로 구분된 경로.

*/v* 버전 정보를 표시합니다.

*/n* 라인 번호 정보에 관계 없이 데이터를 형식화합니다.

*infile* 입력 파일을 지정합니다.

*outfile* 출력 파일을 지정합니다.

### 예:

DB30882416.TRP 트랩 파일이 DIAGPATH에서 생성된 경우 다음과 같이 형식화할 수 있습니다.

```
db2xpvt DB30882416.TRP DB30882416.FMT
```



---

## 제 273 장 disable\_MQFunctions - WebSphere MQ 함수 비활성화 명령

지정된 데이터베이스에 대해 DB2 WebSphere MQ 함수를 사용 불가능하게 합니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *dbadm*
- 함수의 내재적 또는 명시적 스키마 이름이 존재하지 않는 경우 데이터베이스에 대한 IMPLICIT\_SCHEMA
- 스키마 이름, DB2MQ 또는 DB2MQ1C가 존재하는 경우 스키마에 대한 CREATEIN 특권

### 명령 구문

```
►—disable_MQFunctions—-n—database—-u—userid—-p—password—►  
  
└─┬─  
  │  
  └─┬─ all  
      ├── 0pc  
      └── 1pc
```

### 명령 매개변수

**-n** *database*

데이터베이스 이름을 지정합니다.

**-u** *userid*

데이터베이스에 연결하는 데 사용되는 사용자 ID를 지정합니다.

**-p** *password*

사용자 ID의 암호를 지정합니다.

**-v** 선택사항. 트랜잭션 및 비트랜잭션 사용자 정의 함수 지원에 사용됩니다. 해당 값은 all, 0pc 또는 1pc입니다. 0pc를 지정하는 경우 사용 불가능은 db2mq 스키마에서 삭제됩니다. 1pc를 지정하는 경우 사용 불가능은 db2mq1c 스키마에서 삭제됩니다. all을 지정하는 경우 사용 불가능은 두 스키마(db2mq 및 db2mq1c)에서 모두 삭제됩니다. 이 옵션을 지정하지 않는 경우 사용 불가능 디폴트값은 all 옵션입니다.

예 :

다음 예에서는 SAMPLE 데이터베이스에 DB2MQ 및 DB2MQ1C 함수를 사용할 수 없습니다.

```
disable_MQFunctions -n sample -u user1 -p password1
```

---

## 제 274 장 doce\_deinstall - DB2 정보 센터 설치 제거

doce\_deinstall 도구와 동일한 설치 경로에 있는 DB2 정보 센터를 설치 제거합니다. 이 명령은 Linux 운영 체제에서만 사용할 수 있습니다.

doce\_deinstall 명령은 DB2DIR/install에 있습니다. 여기서 DB2DIR은 DB2 정보 센터의 현재 버전이 설치된 위치입니다. doce\_deinstall 명령은 DOCE DVD에서도 사용할 수 있습니다. DVD에서 실행되는 경우 doce\_deinstall 명령에는 -b 옵션이 필요합니다.

### 권한 부여

Root

### 필요한 연결

없음

### 명령 구문

```
doce_deinstall -a [ -b installpath ] [ -l log-file ] [ -t trace-file ] [ -r response_file ] [ -h ] [ -? ]
```

### 명령 매개변수

- a 현재 위치에서 정보 센터를 제거합니다.
- b 이 옵션은 DB2 미디어에서 명령이 실행되는 경우 유효합니다. DB2 제품이 설치되었으며 설치 제거될 절대 경로를 지정합니다. 이 옵션이 지정되지 않으면 명령에서 경로를 위해 프롬프트가 표시됩니다.
- l *log-file*  
로그 파일을 지정합니다. 디폴트 로그 파일은 /tmp/doce\_deinstall.log\$\$입니다. 여기서 \$\$는 프로세스 ID입니다.
- t *trace-file*  
디버그 모드를 활성화합니다. 디버그 정보는 *trace-file*로 지정된 파일 이름에 작성됩니다.
- r *response\_file*  
응답 파일을 사용하여 정보 센터를 제거합니다(예: doce\_deinstall -r db2un.rsp).  
-a 매개변수와 결합할 수 없습니다.

**-h | -?**

사용법 정보를 표시합니다.

**예:**

- /opt/ibm/db2/doce에 설치된 DB2 정보 센터를 설치 제거하려면 다음을 실행하십시오.

```
cd /opt/ibm/db2/doce
doce_deinstall -a
```



---

## 제 275 장 doce\_install - DB2 정보 센터 설치

DB2 정보 센터를 설치합니다. 경로가 지정되지 않으면 DB2 정보 센터가 디폴트로 /opt/ibm/db2ic/V9.7에 설치됩니다. 이 명령은 Linux 운영 체제에만 적용됩니다.

### 권한 부여

Root

### 필요한 연결

없음

### 명령 구문

```
doce_install -b install-path -p product -c image-location -n -L language -l log-file -t trace-file -h -?
```

### 명령 매개변수

#### -b *install-path*

DB2 정보 센터가 설치될 경로를 지정합니다. *install-path*는 전체 경로 이름이어야 하며 최대 경로는 128자로 제한됩니다. 디폴트 설치 경로는 /opt/ibm/db2ic/V9.7입니다. 이 매개변수는 -n 매개변수가 지정된 경우 필수 매개변수입니다.

#### -p *productID*

DB2 정보 센터의 *productID*를 지정합니다. *productID*에는 DB2 접두부가 필요하지 않습니다. 이 매개변수는 -n 매개변수가 지정된 경우 필수 매개변수입니다.

#### -c *image-location*

제품 이미지 위치를 지정합니다. 여러 이미지 위치를 표시하려면 이 매개변수를 여러 번 지정합니다(예: -c CD1 -c CD2). 이 매개변수는 -n 매개변수가 지정되고 설치 시 여러 개의 CD가 필요하며 자동 발견을 위한 이미지가 설정되지 않은 경우에만 필수 매개변수입니다. 그렇지 않으면 필요할 때 다음 CD의 위치를 입력하라는 프롬프트가 표시됩니다.

#### -n 비대화형 모드를 지정합니다.

**-L language**

자국어 지원(NLS)을 지정합니다. 디폴트값은 영어입니다. 동시에 여러 언어를 설치하려면 이 매개변수를 여러 번 지정하면 됩니다. 예를 들어, 영어와 독일어를 모두 설치하려면 -L EN -L DE를 지정합니다.

**-l log-file**

로그 파일을 지정합니다. 디폴트 로그 파일은 /tmp/doce\_install.log\$\$입니다. 여기서 \$\$는 프로세스 ID입니다.

**-t trace-file**

디버그 모드를 활성화합니다. 디버그 정보는 *trace-file*로 지정된 파일 이름에 작성됩니다.

**-h | -?**

사용법 정보를 표시합니다.

**예:**

- /mnt/cdrom에서 이미지를 설치하고 필요한 모든 입력에 대한 프롬프트를 표시하려면 다음을 실행하십시오.

```
cd /mnt/cdrom
./doce_install
```

- DB2 정보 센터를 /mnt/cdrom의 이미지에서 비대화식 영어로 /db2/v9.7에 설치하려면 다음을 실행하십시오.

```
cd /mnt/cdrom
./doce_install -p doce -b /db2/v9.7 -n
```

---

## 제 276 장 enable\_MQFunctions - WebSphere MQ 함수 사용 명령

지정된 데이터베이스에 DB2 WebSphere MQ 함수를 사용 가능하게 하고 DB2 WebSphere MQ 함수가 올바르게 실행될 수 있는지 확인합니다. WebSphere MQ 및 WebSphere MQ AMI가 설치 및 구성되지 않으면 명령이 실패합니다.

### 권한 부여

다음 중 하나가 필요합니다.

- 데이터베이스의 CREATE\_EXTERNAL\_ROUTINE 권한은 다음 중 최소한 하나를 포함해야 합니다.
  - 함수의 스키마 이름이 기존 스키마를 참조하지 않는 경우, 데이터베이스에 대한 IMPLICIT\_SCHEMA 권한
  - 스키마 이름 DB2MQ 또는 DB2MQ1C가 존재하는 경우, 스키마에 대한 CREATEIN 특권

• *dbadm*

### 명령 구문

```
enable_MQFunctions -n database -u userid -p password  
-q queuemanager -force -xml xmlSize -c clobSize  
-novalidate -v all 0pc 1pc
```

### 명령 매개변수

**-n database**

사용할 데이터베이스의 이름을 지정합니다.

**-u userid**

데이터베이스에 연결할 사용자 ID를 지정합니다.

**-p password**

사용자 ID의 암호를 지정합니다.

**-q queuemanager**

선택사항. 트랜잭션 MQ 사용자 정의 함수를 지원하는 큐 관리 프로그램 이름.

이름을 지정하지 않는 경우 디폴트 큐 관리 프로그램인

DB2MQ\_DEFAULT\_MQM입니다. 이 옵션을 사용하는 경우 함수는 -novalidate 매개변수를 사용하는 것으로 가정합니다.

**-echo** 선택사항. UDF 또는 진단 정보를 작성하는 데 사용되는 자세한 SQL을 인쇄합니다.

**-force** 선택사항. 이 옵션을 사용하면 유틸리티 프로그램이 기존 MQ UDF를 무시할 수 있습니다. 즉, 프로그램이 MQ UDF를 재작성하기 전에 기존 기능을 삭제(drop)합니다. 이 옵션을 사용하지 않으면 MQ UDF가 이미 존재하는 것을 확인한 후 명령이 처리되지 않습니다.

**-novalidate**

선택사항. DB2 MQSeries® 함수의 유효성을 확인하지 않는 것으로 지정합니다.

**-xml xmlSize**

선택사항. 0pc 함수의 XML 버전 정의에 사용됩니다. "-v 1pc" 옵션을 지정하면 이 옵션은 효과가 없습니다.

xmlSize는 XML 데이터 길이를 지정합니다. 최소 길이는 1바이트입니다. 최대 길이는 100M입니다. 디폴트는 1M입니다. 길이는 n(바이트 수), nK(KB 단위의 길이) 또는 nM(MB 단위의 길이)으로 지정할 수 있습니다.

**-c clobSize**

선택사항. CLOB 데이터 길이를 지정합니다. 최소 길이는 1바이트이며 이는 디폴트입니다. 최대 길이는 100M입니다. 길이는 n(바이트 수), nK(KB 단위의 길이) 또는 nM(MB 단위의 길이)으로 지정할 수 있습니다.

**-v** 선택사항. 트랜잭션 및 비트랜잭션 사용자 정의 함수 지원에 사용됩니다. 해당 값은 all, 0pc 또는 1pc입니다. 0pc를 지정하는 경우 사용 시 db2mq 스키마를 작성합니다. 1pc를 지정하는 경우 사용 시 db2mq1c 스키마를 작성합니다. all을 지정하는 경우 사용 시 사용자 정의 함수(UDF)의 모든 스키마(db2mq 및 db2mq1c)를 작성합니다. 이 옵션을 지정하지 않는 경우 사용 시 디폴트 옵션은 all입니다.

**예:**

다음 예에서는 트랜잭션 및 비트랜잭션 사용자 정의 함수(UDF)를 사용합니다. 사용자는 SAMPLE 데이터베이스에 연결됩니다.

```
enable_MQFunctions -n sample -u user1 -p password1
```

다음 예에서는 사용자가 SAMPLE 데이터베이스에 연결됩니다. 예에서는 DB2MQ1C 스키마로 DB2MQ1C 함수를 작성합니다.

```
enable_MQFunctions -n sample -u user1 -p password1 -v 1pc
```

## 사용 시 참고사항

DB2 MQ 사용자 정의 함수(UDF)는 이 명령이 자동으로 작성하는 DB2MQ 또는 DB2MQ1C 스키마에서 실행됩니다. 이 명령을 실행하기 전에 다음을 수행하십시오.

- WebSphere MQ 및 WebSphere AMI(Application Messaging Interface)가 설치되고 WebSphere MQ 버전이 5.1 이상인지 확인하십시오.
- \$AMT\_DATA\_PATH 환경 변수가 정의되었는지 확인하십시오.
- 트랜잭션 MQ UDF를 사용하려는 경우 페더레이티드 조작에 대해 데이터베이스가 구성되었는지 확인하십시오. 이 작업은 다음 명령으로 수행하십시오.

```
update dbm cfg using federated yes
```

- 디렉토리를 DB2PATH의 서브디렉토리로 변경하십시오.

UNIX:

- db2set을 사용하여 DB2ENVLIST에 AMT\_DATA\_PATH를 추가하십시오.
- UDF 실행과 연관된 사용자 어카운트가 mqm 그룹의 구성원인지 확인하십시오.
- 이 명령을 호출하는 사용자가 mqm 그룹의 구성원인지 확인하십시오.

주: AIX 4.2는 MQSeries 5.2가 지원하지 않습니다.



## 제 277 장 installFixPack - 설치된 DB2 제품 갱신

모든 UNIX 및 Linux 플랫폼에서 지정된 위치에 설치된 DB2 제품을 이미지와 동일한 레벨로 갱신합니다. 복수 사본 DB2 제품이 설치된 경우 installFixPack 명령은 지정된 경로에 따라 한 번에 하나의 사본을 갱신합니다. 이 명령은 이미지에서 맨 위 디렉토리에 있습니다.

설치 경로와 관련된 모든 인스턴스(또한 DPF의 경우 모든 노드)의 데이터베이스 관리 프로그램(DBM)이 중지되면 FixPack 설치가 진행되고 모든 DB2 라이브러리가 언로드됩니다. 모든 사전조건이 충족되면 installFixPack이 설치 경로와 관련된 해당 인스턴스 및 DAS를 갱신합니다. 추가 수동 갱신은 필요하지 않습니다. 모든 UNIX 및 Linux 플랫폼의 경우, 데이터베이스가 다시 연결되거나 응용프로그램이 재시작될 때 dxlink 명령이 자동으로 시작됩니다.

예를 들어, 일부 DBM만 중지되거나 DB2 라이브러리가 로드 상태를 유지하는 경우, FixPack 설치를 계속하기 위해 다른 강제실행 옵션을 지정할 수도 있습니다. installFixPack은 계속 실행되지만 인스턴스와 DAS를 수동으로 갱신하고 응용프로그램을 다시 시작해야 합니다.

DPF 인스턴스의 경우 모든 노드에 FixPack을 설치합니다. 이 경우 인스턴스 갱신은 노드를 소유하는 인스턴스에만 필요합니다. 갱신 후에도 인스턴스가 계속 실행되도록 하려면 적어도 노드를 소유하는 인스턴스의 모든 노드에 모든 제품과 기능을 설치하는 것이 좋습니다.

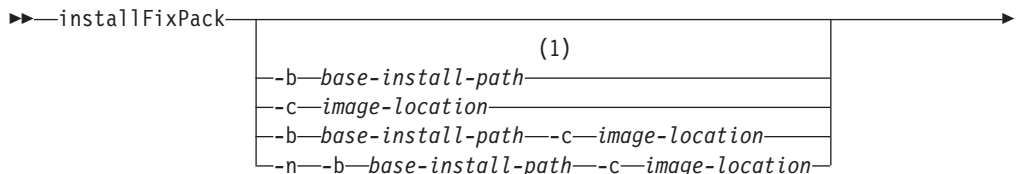
### 권한 부여

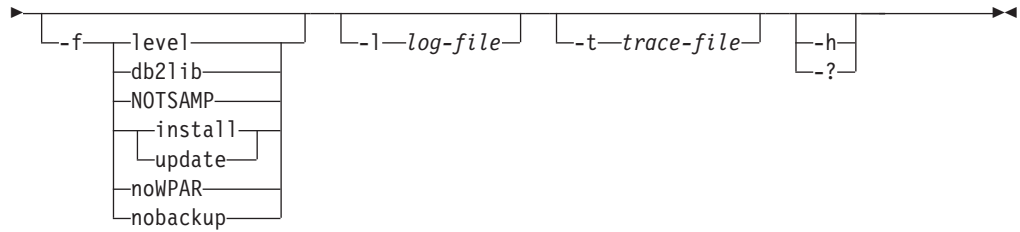
루트 설치에는 root 권한이 필요합니다. 비루트 설치의 경우 비루트 설치를 소유하는 사용자 ID로 로그인해야 합니다.

### 필요한 연결

없음

### 명령 구문





주:

- 1 이 옵션을 생략하면 오류 메시지 정지 없이 필요한 정보에 대한 프롬프트가 표시됩니다.

## 명령 매개변수

**-n** 비대화형 모드를 지정합니다. 이 옵션이 지정되면 **-b** 및/또는 **-c**도 지정해야 합니다. 이 모드를 사용하면 응용프로그램이 명령행에서 자동 모드로 필요한 모든 정보를 제공할 수 있습니다.

### **-b** *base-install-path*

DB2 제품이 설치될 경로를 지정합니다. **-n**이 지정된 경우 필수입니다. 경로 길이는 128자로 제한되며 전체 경로 이름입니다.

**-b** 옵션은 DB2의 비루트 설치에는 필요하지 않지만 루트 설치에는 필수 옵션입니다. **-b**가 루트 서버가 아닌 설치에서 사용되는 경우 *base-install-path* 값은 사용자의 HOME/sqllib 디렉토리여야 합니다. 그렇지 않으면 경로가 유효하지 않은 것으로 간주됩니다. **-b**가 루트 서버가 아닌 설치에서 제공되지 않으면 DB2 설치 프로그램이 사용자의 HOME/sqllib를 설치 경로로 사용하여 계속합니다. 그러나 **-b**가 사용되고 제공된 설치 경로에 루트 사용자가 설치한 DB2 사본이 있는 경우 해당 경로는 유효하지 않은 것으로 간주됩니다. DB2 사본은 설치한 사용자만 갱신할 수 있기 때문입니다.

### **-c** *NLPACK\_location*

관련된 DB2 자국어 팩(NLPACK)의 위치를 지정합니다. 이 매개변수는 **-n**이 지정된 경우 필수 매개변수입니다. 다음 조건이 모두 충족되는 경우 DB2 NLPACK 위치가 명시적으로 제공되어야 합니다.

- **-n** 옵션이 지정되었습니다.
- 설치 시 자국어(영어 이외) 지원이 필요합니다.
- DB2 NLPACK이 DB2 DVD에도 설치될 DB2 제품과 동일한 서브디렉토리에도 없습니다.

**-f** 강제 실행 옵션. 인수가 없는 **-f**는 지원되지 않습니다. 아래 강제 인수와 조합될 수 있습니다(예를 들어, **-f level -f db2lib**).



**-f level**

하위 레벨 또는 동일한 레벨의 FixPack 설치를 강제 실행합니다. FixPack 이미지가 설치된 DB2 제품보다 상위 레벨인 경우에는 이 옵션이 무시됩니다.

**-f db2lib**

DB2 라이브러리 로딩 시 검사를 생략하도록 installFixPack을 강제 실행합니다. 설치 후 인스턴스 또는 응용프로그램이 올바르게 실행되도록 하려면 모든 관련 인스턴스(관련 DPF 인스턴스의 모든 노드 포함)에 대해 DBM을 중지해야 하며 설치 경로 관련 모든 DB2 라이브러리를 언로드해야 합니다.

**-f NOTSAMP**

SA MP를 갱신해서는 안되는 것으로 지정합니다(루트 설치에만 해당).

**-f install**

DB2 로딩 시 올바르게 중지되거나 중지되지 않은 인스턴스 및 DAS에 대한 모든 검사를 생략하도록 installFixPack을 강제 실행합니다. 설치 후 인스턴스 또는 응용프로그램이 올바르게 실행되도록 하려면 모든 관련 인스턴스(관련 DPF 인스턴스의 모든 노드 포함)에 대해 DBM을 중지해야 하며 설치 경로 관련 모든 DB2 라이브러리를 언로드해야 합니다. 이 옵션이 지정되면 인스턴스/DAS가 갱신되지 않습니다. 사용자는 설치 후 인스턴스/DAS 갱신을 수동으로 수행해야 합니다. 또한 옵션 갱신과 설치는 상호 배타적이므로 동일한 설치에 지정할 수 없습니다.

**-f update**

DB2 로딩 시 올바르게 중지되거나 중지되지 않은 인스턴스 및 DAS에 대한 모든 검사를 생략하도록 installFixPack을 강제 실행합니다. 설치 후 인스턴스 또는 응용프로그램이 올바르게 실행되도록 하려면 모든 관련 인스턴스(관련 DPF 인스턴스의 모든 노드 포함)에 대해 DBM을 중지해야 하며 설치 경로 관련 모든 DB2 라이브러리를 언로드해야 합니다. 이 옵션이 지정되면 인스턴스/DAS가 갱신됩니다. 또한 옵션 갱신과 설치의 상호 배타적이므로 동일한 설치에 지정할 수 없습니다.

**-f noWPAR**

글로벌 환경에서 AIX 6.1 이상에만 적용됩니다. 글로벌 환경에서 갱신되는 DB2 사본을 공유하는 AIX 시스템 워크로드 파티션(WPAR)에 대해 아무런 검사 또는 조치를 수행하지 않도록 installFixPack을 강제 실행합니다. -f noWPAR이 지정되면 이 DB2 사본을 공유하는 각 시스템 WPAR에서 인스턴스와 DAS를 수동으로 갱신해야 합니다.

**-f nobackup**

구성요소가 갱신될 때 installFixPack이 설치 파일을 백업하지 않도록 강제 실행합니다. 파일을 백업하지 않는 것으로 선택하면 설치 디렉토리의 공간 요구 사항이 감소합니다. 그러나 파일을 백업하지 않는 것으로 선택하는 경우 오류

발생 시 DB2 설치 프로그램이 롤백 조작을 수행할 수 없음을 의미하기도 합니다. 이러한 경우 파일을 수동으로 제거하고 FixPack을 재설치해야 합니다.

#### **-l** *log-file*

로그 파일을 지정합니다. 루트 설치의 경우 디폴트 로그 파일은 /tmp/installFixPack.log\$\$입니다. 여기서 \$\$는 프로세스 ID를 나타냅니다. 비루트 설치의 경우 디폴트 로그 파일은 /tmp/installFixPack\_userID.log입니다. 여기서 *userID*는 비루트 설치를 소유하는 사용자 ID를 나타냅니다. IBM Tivoli SA MP(System Automation for Multiplatforms)가 installFixPack 명령으로 설치 또는 갱신되는 경우 해당 로그 파일은 DB2 로그 파일과 동일한 디렉토리에 저장됩니다.

#### **-t** *trace-file*

디버그 모드를 활성화합니다. 디버그 정보는 지정된 파일 이름에 작성됩니다.

#### **-h | -?**

도움말 정보를 표시합니다.

### 사용 시 참고사항

- 설치된 DB2 텍스트 검색이 있고 텍스트 검색이 DB2 사본과 관련된 인스턴스에서 실행 중인 경우 먼저 텍스트 검색 인스턴스 서비스를 중지해야 함을 표시하는 오류 메시지를 수신하게 됩니다. 텍스트 검색 인스턴스 서비스를 중지한 다음 명령을 재실행하십시오.

#### 예:

- DB2 Enterprise Server Edition German이 /opt/ibm/db2/COPY1에 설치된 경우 GA에서 FixPack 1로의 대화식 갱신을 수행하려면 FixPack 1 이미지에서 다음을 실행하십시오.

```
./installFixPack -b /opt/ibm/db2/COPY1
```

- DB2 Enterprise Server Edition German이 /opt/ibm/db2/COPY1에 설치된 경우 GA에서 FixPack 1로의 자동 갱신을 수행하려면 FixPack 1 이미지에서 다음을 실행하십시오.

```
./installFixPack -b /opt/ibm/db2/COPY1 -c full_path_to_NLPACK_image -n
```

- 어떠한 이유로 설치된 DB2 제품 파일이 손상되는 경우 설치 제거한 후 다시 설치하여 설치를 새로 고치는 대신 다음을 실행하십시오.

```
./installFixPack -f level -b full_path_where_DB2_product_installed
```

- 설치 디렉토리의 공간 요구사항을 줄이려면 다음을 실행하십시오.

```
./installFixPack -f nobackup -b full_path_where_DB2_product_installed
```

---

## 제 278 장 setup - DB2 설치

DB2 제품을 설치합니다. 이 명령은 Windows 운영 체제에서만 사용할 수 있습니다. UNIX 운영 체제용 명령은 db2setup입니다.

이 유틸리티는 DB2 설치 미디어에 있습니다. 이 유틸리티는 DB2 설정 마법사를 시작하여 설치를 정의하고 DB2 제품을 설치합니다. -u 옵션으로 호출되는 경우 추가 입력 없이 응답 파일에서 설치 구성 정보를 가져와 설치를 수행합니다.

IBM Data Server Runtime Client를 Windows에 설치하는 경우의 설정 옵션은 DB2 제품을 설치하는 경우와 다릅니다. 적합한 옵션은 『IBM Data Server Runtime Client 설치 명령행 옵션(Windows)』을 참조하십시오.

### 명령 구문

```
▶▶ setup [-c] [-f] [-i language] [-l log-file] [-m]
          [-p install-directory] [-t trace-file] [-u response-file]
          [-n DB2-copy-name] [-o] [--?] [-h]
```

### 명령 매개변수

주:

DB2 setup 설치 명령은 / 또는 - 전환 기호를 교대로 사용할 수 있습니다.

- c 설치 시작 직후 setup.exe가 존재하는지 확인합니다. 이 옵션을 선택하면 setup.exe의 종료 코드를 모니터링할 때 설치 리턴 코드를 사용할 수 없습니다.
- f 설치 전에 DB2 프로세스를 강제로 중지합니다.
- i language  
설치를 수행하는 언어의 2자 언어 코드를 지정합니다.
- l log-file  
사용할 로그 파일의 전체 경로 및 파일 이름.
- m -u 옵션과 함께 사용되어 설치 중 진행 대화 상자를 나타냅니다. 그러나 입력 프롬프트는 표시되지 않습니다.

- p** *install-directory*  
제품의 설치 경로를 변경합니다. 이 옵션을 지정하면 응답 파일에 지정된 설치 경로를 겹쳐씁니다.
- t** *trace-file*  
설치 추적 정보로 파일을 생성합니다.
- u** *response-file*  
사용할 응답 파일의 전체 경로 및 파일 이름을 지정합니다.
- n** *DB2-copy-name*  
사용하기 위해 설치할 DB2 사본 이름을 지정합니다. 이 옵션을 지정하면 응답 파일에 지정된 사본 이름을 겹쳐씁니다.
- o** 항상 생성된 디폴트 사본 이름으로 새 사본 설치를 수행합니다. 이 옵션은 Windows에 IBM Data Server Driver Package을 설치하는 경우에만 사용할 수 있습니다.
- ? | -h**  
사용법 정보를 생성합니다.

---

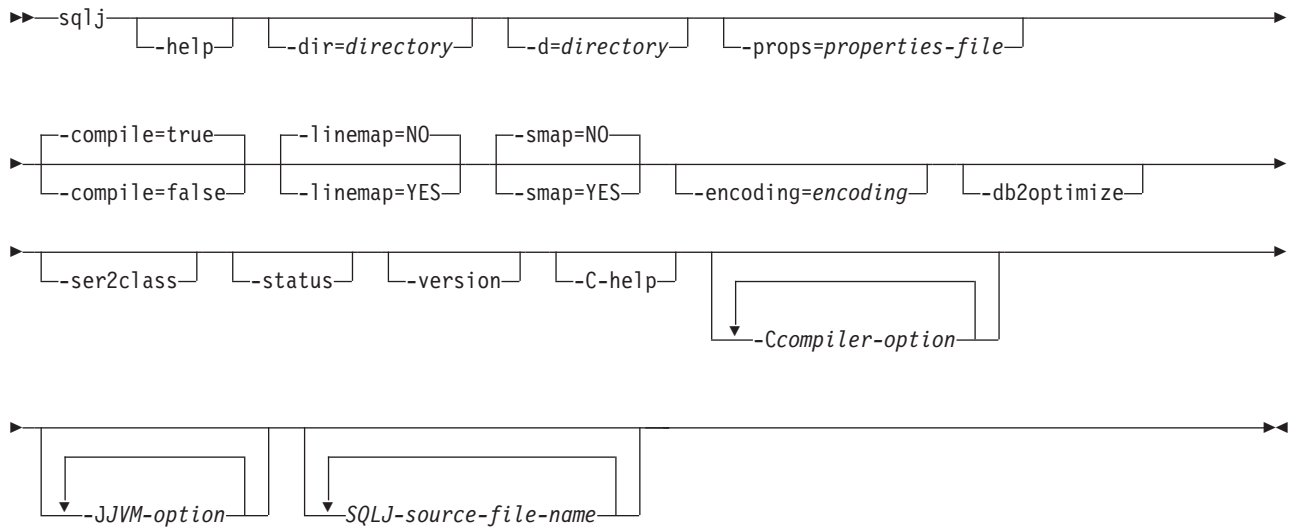
## 제 279 장 sqlj - SQLJ 변환기

sqlj 명령은 SQLJ 소스 파일을 Java 소스 파일 및 0개 이상의 SQLJ 직렬화 프로파일로 변환합니다. 디폴트로, sqlj 명령은 Java 소스 파일을 컴파일하기도 합니다.

### 권한 부여

None

### 명령 구문



### 명령 매개변수

#### -help

SQLJ 변환기가 지원하는 각 옵션에 대해 설명하도록 지정합니다. -help로 지정되는 기타 옵션은 무시됩니다.

#### -dir=directory

SQLJ가 변환기를 통해 생성된 .java 파일 및 컴파일러를 통해 생성된 .class 파일을 저장하는 디렉토리 이름을 지정합니다. 디폴트 디렉토리는 SQLJ 소스 파일이 들어 있는 디렉토리입니다.

변환기는 생성된 파일을 디렉토리에 저장할 때 SQLJ 소스 파일의 디렉토리 구조를 사용합니다. 예를 들어, 변환기를 통해 다음 2개의 파일을 처리하려 한다고 가정합니다.

- file1.sqlj(Java 패키지에 없음)
- file2.sqlj(Java 패키지 sqlj.test에 있음)

또한 변환기 호출 시 `-dir=/src` 매개변수를 지정한다고 가정합니다. 변환기는 `file1.sqlj`의 Java 소스 파일을 `/src` 디렉토리에 저장하고 `file2.sqlj`의 Java 소스 파일을 `/src/sqlj/test` 디렉토리에 저장합니다.

**-d=directory**

SQLJ가 변환기 및 컴파일러를 통해 생성된 2진 파일을 저장하는 디렉토리 이름을 지정합니다. 이 파일에는 컴파일러를 통해 생성된 `.ser` 파일, `name_SJProfileKeys.class` 파일 및 `.class` 파일이 포함됩니다.

디폴트 디렉토리는 SQLJ 소스 파일이 들어 있는 디렉토리입니다.

변환기는 생성된 파일을 디렉토리에 저장할 때 SQLJ 소스 파일의 디렉토리 구조를 사용합니다. 예를 들어, 변환기를 통해 다음 2개의 파일을 처리하려 한다고 가정합니다.

- `file1.sqlj`(Java 패키지에 없음)
- `file2.sqlj`(Java 패키지 `sqlj.test`에 있음)

또한 변환기 호출 시 `-d=/src` 매개변수를 지정한다고 가정합니다. 변환기는 `file1.sqlj`의 직렬화 프로파일을 `/src` 디렉토리에 저장하고 `file2.sqlj`의 직렬화 프로파일을 `/src/sqlj/test` 디렉토리에 저장합니다.

**-compile=truelfalse**

SQLJ 변환기가 생성된 Java 소스를 바이트코드로 컴파일하는지 여부를 지정합니다.

**true**

변환기가 생성된 Java 소스 코드를 컴파일합니다. 이는 디폴트입니다.

**false**

변환기가 생성된 Java 소스 코드를 컴파일하지 않습니다.

**-linemap=nolyes**

Java 예외의 라인 번호가 SQLJ 소스 파일(.sqlj 파일)의 라인 번호 또는 SQLJ 변환기를 통해 생성된 Java 소스 파일(.java 파일)의 라인 번호와 일치하는지를 지정합니다.

**no** Java 예외의 라인 번호가 Java 소스 파일의 라인 번호와 일치합니다. 이는 디폴트입니다.

**yes**

Java 예외의 라인 번호가 SQLJ 소스 파일의 라인 번호와 일치합니다.

**-smap=nolyes**

SQLJ 변환기가 각 SQLJ 소스 파일에 적합한 소스 맵(SMAP) 파일을 생성하는지 여부를 지정합니다. SMAP 파일은 몇몇 Java 언어 디버그 도구에서 사용됩니다. 이 파일은 SQLJ 소스 파일의 라인을 SQLJ 변환기를 통해 생성된 Java 소스 파일의

라인에 맵핑합니다. 파일은 유니코드 UTF-8 인코딩 스킴으로 되어 있습니다. 다음 웹 사이트에 있는 Original JSR(Java Specification Request) 45에서 해당 형식에 대해 설명합니다.

<http://www.jcp.org>

**no** SMAP 파일을 생성하지 않습니다. 이는 디폴트입니다.

**yes**

SMAP 파일을 생성합니다. SMAP 파일 이름은 *SQLJ-source-file-name.java.smap*입니다. SQLJ 변환기는 SMAP 파일을 생성된 Java 소스 파일과 동일한 디렉토리에 저장합니다.

**-encoding=encoding-name**

소스 파일의 인코딩을 지정합니다. 예로는 JIS 또는 EUC가 있습니다. 이 옵션을 지정하지 않으면 운영 체제의 디폴트 변환기가 사용됩니다.

**-db2optimize**

SQLJ 변환기가 DB2에 적합하게 최적화된 연결 컨텍스트 클래스의 코드를 생성하도록 지정합니다. **-db2optimize**는 사용자 정의 컨텍스트의 코드를 최적화하며 디폴트 컨텍스트를 최적화하지 않습니다.

**-db2optimize** 옵션을 사용하여 SQLJ 변환기를 실행할 때 응용프로그램에서 JDBC 3.0 이전 기능을 사용하는 경우, 생성된 Java 응용프로그램을 컴파일할 수 있도록 IBM Data Server Driver for JDBC and SQLJ 파일 *db2jcc.jar*이 CLASSPATH에 있어야 합니다. 사용자의 응용프로그램에서 JDBC 4.0 이전 기능을 사용하는 경우에는 생성된 Java 응용프로그램을 컴파일할 수 있도록 IBM Data Server Driver for JDBC and SQLJ 파일 *db2jcc4.jar*이 CLASSPATH에 있어야 합니다.

**-ser2class**

SQLJ 변환기가 *.ser* 파일을 *.class* 파일로 변환하도록 지정합니다.

**-status**

SQLJ 변환기가 실행되는 동안 상태 메시지를 표시하도록 지정합니다.

**-version**

SQLJ 변환기가 IBM Data Server Driver for JDBC and SQLJ 버전을 표시하도록 지정합니다. 정보 양식은 다음과 같습니다.

IBM SQLJ *xxxx.xxxx.xx*

**-C-help**

SQLJ 변환기가 Java 컴파일러에 대한 도움말 정보를 표시하도록 지정합니다.

**-Ccompiler-option**

대시(-)로 시작하는 유효한 Java 컴파일러 옵션을 지정합니다. **-C**와 컴파일러 옵션 사이에 스페이스를 포함시키지 마십시오. 여러 컴파일러 옵션을 지정해야 하는 경우 각 컴파일러 옵션 앞에 **-C**를 표시하십시오. 예를 들어, 다음과 같습니다.

-C-g -C-verbose

모든 옵션은 Java 컴파일러에 전달되며 다음 옵션을 이외에는 SQLJ 변환기에서 사용되지 않습니다.

#### **-classpath**

SQLJ 변환기 및 Java 컴파일러에서 사용할 사용자 클래스 경로를 지정합니다. 이 값은 CLASSPATH 환경 변수를 겹쳐씁니다.

#### **-sourcepath**

SQLJ 변환기 및 Java 컴파일러가 클래스 또는 인터페이스 정의를 검색하는 소스 코드 경로를 지정합니다. SQLJ 변환기는 디렉토리에서만 .sqlj 및 .java 파일을 검색하고 JAR이나 zip 파일에서는 검색하지 않습니다.

#### **-JVM-option**

sqlj 명령이 실행되는 JVM(Java Virtual Machine)에 전달할 옵션을 지정합니다. 옵션은 대시(-)로 시작하는 유효한 JVM 옵션이어야 합니다. -J와 JVM 옵션 사이에 스페이스를 포함시키지 마십시오. 여러 JVM 옵션을 지정해야 하는 경우 각 컴파일러 옵션 앞에 -J를 표시하십시오. 예를 들어, 다음과 같습니다.

-J-Xmx128m -J-Xmine2M

#### **SQLJ-source-file-name**

변환할 SQLJ 소스 파일 목록을 지정합니다. 이것은 필수 매개변수입니다. 모든 SQLJ 소스 파일 이름에는 .sqlj 확장자가 있어야 합니다.

## **출력**

각 소스 파일 *program-name.sqlj*에서 SQLJ 변환기는 다음 파일을 생성합니다.

- 생성되는 소스 프로그램

생성되는 소스 파일 이름은 *program-name.java*로 지정됩니다.

- SQLJ executable 절에서 사용되는 각 연결 컨텍스트 클래스의 직렬화 프로파일 파일

직렬화 프로파일 이름의 양식은 다음과 같습니다.

*program-name\_SJProfileIDNumber.ser*

- SQLJ 변환기가 Java 컴파일러를 호출하는 경우, 컴파일러가 생성하는 클래스 파일

## **예:**

```
sqlj -encoding=UTF8 -C-0 MyApp.sqlj
```



---

## 제 6 부 DB2 텍스트 검색 명령



---

## 제 280 장 db2ts ALTER INDEX

이 명령은 인덱스의 갱신 특성을 변경합니다.

이 명령을 실행하려면 명령행에서 db2ts를 접두부로 추가해야 합니다.

### 권한 부여

명령문의 권한 부여 ID는 적어도 다음과 같은 특권 중 하나를 가지고 있어야 합니다.

- 텍스트 인덱스가 정의되는 테이블에 대한 CONTROL 특권
- DBADM 권한

### 전제조건

인스턴스 소유자는 DATAACCESS 권한이 있는 DBADM을 보유해야 합니다. SYSADM은 더 이상 V9.7에서 SECADM 및 DBADM 특권을 보유하지 않습니다. SECADM은 ALTER INDEX 명령을 실행하기 전에 인스턴스 소유자에게 DATAACCESS 권한이 있는 DBADM을 명시적으로 권한 부여해야 합니다.

### 필수 연결

데이터베이스

### 명령 구문

```
▶▶ ALTER INDEX index-name FOR TEXT | update characteristics |—————▶  
▶ | connection options |—————▶▶
```

#### update characteristics:

```
|—————▶  
| UPDATE FREQUENCY | NONE |—————▶  
| | update frequency |  
▶ | incremental update characteristics |—————▶
```

#### update frequency:

```
|—D—(—*—) |—H—(—*—) |—M—(—integer3—) |—————▶  
| | integer1 | | integer2 |  
| | | |
```

## incremental update characteristics:

```
|-----|  
| UPDATE MINIMUM—minchanges-----|
```

## connection options:

```
|-----|  
| CONNECT TO—database-name-----|  
| USER—username—USING—password-----|
```

## 명령 매개변수

### ALTER INDEX *index-name*

CREATE INDEX 명령에 지정된 인덱스의 스키마와 이름. 데이터베이스에서 텍스트 검색 인덱스를 고유하게 식별합니다.

### UPDATE FREQUENCY

인덱스 갱신 빈도를 지정합니다. 인덱스는 변경 수가 UPDATE MINIMUM에 설정된 값 이상인 경우 갱신됩니다. 갱신 빈도 NONE은 인덱스가 더 이상 갱신되지 않음을 나타냅니다. 이 옵션은 변경되지 않는 데이터가 있는 테이블의 텍스트 컬럼에 유용합니다. 또한 사용자가 UPDATE INDEX 명령을 사용하여 인덱스를 수동으로 갱신하려는 경우에도 유용합니다. 자동 갱신은 START FOR TEXT 명령이 실행되고 DB2 텍스트 검색 인스턴스 서비스가 실행 중인 경우에만 수행될 수 있습니다.

디폴트 빈도 값은 SYSIBMTS.TSDEFAULTS 뷰 (DEFAULTNAME='UPDATEFREQUENCY')에서 가져옵니다.

### NONE

텍스트 인덱스에 자동 갱신이 적용되지 않습니다. 추가 인덱스 갱신은 수동으로 시작해야 합니다.

### D 인덱스가 갱신된 요일.

\* 모든 요일.

*integer1*

일요일에서 토요일까지의 특정 요일(0 - 6)

### H 인덱스가 갱신된 지정된 날짜의 시간.

\* 모든 시간.

*integer2*

자정에서 오후 11시까지 하루 중 특정 시간(0 - 23)

### M 인덱스가 갱신된 지정된 시간의 분.

*integer3*

시간의 맨 위(0) 또는 해당 시간 이후 5분 간격으로 증가하는 시간(0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50 또는 55)

UPDATE FREQUENCY 옵션을 지정하지 않으면 빈도 설정이 변경되지 않습니다.

#### **UPDATE MINIMUM** *minchanges*

인덱스가 점차적으로 갱신되기 전에 발생해야 하는 텍스트 문서에 대한 최소 변경 횟수를 지정합니다. 동일한 텍스트 문서를 여러 번 변경하는 경우 개별 변경사항으로 간주됩니다. UPDATE MINIMUM 옵션을 지정하지 않으면 설정이 변경되지 않습니다.

#### **CONNECT TO** *database-name*

이 절은 연결이 설정될 데이터베이스를 지정합니다. 데이터베이스는 로컬 시스템에 있어야 합니다. 이 절이 지정된 경우 DB2DBDFT 환경 변수에 우선합니다. 다음의 조건에 모두 해당되는 경우 이 절을 생략할 수 있습니다.

- DB2DBDFT 환경 변수가 유효한 데이터베이스 이름으로 설정됩니다.
- 명령을 실행하는 사용자에게 데이터베이스 서버에 연결하기 위해 필요한 권한이 있습니다.

#### **USER** *username* **USING** *password*

이 절은 연결을 설정하는 데 사용될 사용자 이름과 암호를 지정합니다.

## 사용 시 참고사항

DB2 데이터베이스 오브젝트 및 쿼리에 적용되는 모든 한계와 이름 지정 규칙은 DB2 텍스트 검색 기능과 쿼리에도 적용됩니다. DB2 텍스트 검색 관련 ID는 DB2 이름 지정 규칙을 준수해야 합니다. 또한 몇 가지 추가 제한사항도 있습니다. 예를 들어, 이러한 ID의 양식은 다음과 같습니다.

[A-Za-z][A-Za-z0-9@#\$\_]\*

또는

"[A-Za-z ][A-Za-z0-9@#\$\_]\*"

충돌 가능성이 있는 경우에는 텍스트 검색 인덱스에서 동시에 여러 명령을 실행할 수 없습니다. 충돌 명령이 실행 중일 때 이 명령이 실행되는 경우 오류가 발생하여 명령이 실패하며 이후 명령 실행을 다시 시도할 수 있습니다. 충돌 명령의 예는 다음과 같습니다.

- ALTER INDEX
- CLEAR EVENTS FOR INDEX
- DROP INDEX
- UPDATE INDEX

- DISABLE DATABASE FOR TEXT
- STOP FOR TEXT

데이터베이스 변경사항: DB2 텍스트 검색 카탈로그 정보를 갱신합니다.

---

## 제 281 장 db2ts CLEANUP FOR TEXT

이 명령은 인스턴스 내에서 사용하지 않는 DB2 텍스트 검색 컬렉션을 제거합니다. 컬렉션은 다음의 경우에 사용하지 않게 됩니다.

- 텍스트 검색 인덱스가 있는 데이터베이스는 데이터베이스에 대해 DB2 텍스트 검색이 사용되지 않도록 설정되기 전에 삭제됩니다.
- 테이블은 연관된 텍스트 검색 인덱스가 삭제되기 전에 삭제됩니다.

주: 텍스트 검색 컬렉션은 텍스트 검색 인덱스의 기본적인 표현을 참조합니다. 텍스트 검색 컬렉션과 텍스트 검색 인덱스 사이에는 일대일 관계가 있습니다. 텍스트 검색 인덱스에 대해 명령이 작동하는 반면 도구는 텍스트 검색 컬렉션에 대해 작동합니다. 텍스트 검색 인덱스의 텍스트 검색 컬렉션을 판별하려면

SYSIBMTS.TSCOLLECTIONNAMES 카탈로그 테이블을 쿼리하십시오. 추가 정보는 『DB2 텍스트 검색에 대한 관리 도구』를 참조하십시오.

이 명령을 실행하려면 명령행에서 db2ts를 접두부로 추가해야 합니다.

### 권한 부여

명령을 실행하려면 사용자가 DBADM 및 DATAACCESS 권한을 가지고 있는 DB2 인스턴스 소유자여야 합니다.

### 필수 연결

이 명령은 DB2 데이터베이스 서버에서 실행되어야 합니다.

### 명령 구문

▶▶—CLEANUP FOR TEXT—▶▶

### 명령 매개변수

없음





---

## 제 282 장 db2ts CLEAR COMMAND LOCKS

특정 텍스트 검색 인덱스 또는 데이터베이스의 모든 텍스트 검색 인덱스에 대한 모든 명령 잠금을 제거합니다. 명령 잠금은 텍스트 검색 인덱스 명령 시작 시 작성되며 완료 시 삭제됩니다. 이렇게 하면 다른 명령 간의 바람직하지 않은 충돌을 방지할 수 있습니다.

정리는 더 이상 사용하지 않는 프로세스와 연관된 모든 잠금에 대해 자동으로 수행됩니다. 이는 새 검색 요청 시 텍스트 검색 인덱스에 액세스할 수 있도록 하기 위해 수행됩니다.

이 명령은 자주 사용되지는 않지만 예기치 않은 시스템 동작으로 인해 잠금이 남아 있고 명시적으로 제거해야 하는 경우 사용합니다.

이 명령을 실행하려면 명령행에서 db2ts를 접두부로 추가해야 합니다.

### 권한 부여

인덱스 이름이 지정되지 않은 경우 데이터베이스 연결의 *username*에는 DBADM 권한이 있어야 합니다. 특정 인덱스에 대한 명령 잠금을 지우려면 데이터베이스 연결의 *username*에 텍스트 검색 인덱스가 작성된 테이블에 대한 CONTROL 특권이 있어야 합니다.

### 전제조건

인스턴스 소유자는 DATAACCESS 권한이 있는 DBADM을 보유해야 합니다. SYSADM은 더 이상 V9.7에서 SECADM 및 DBADM 특권을 보유하지 않습니다. SECADM은 CLEAR COMMAND 명령을 실행하기 전에 인스턴스 소유자에게 DATAACCESS 권한이 있는 DBADM을 명시적으로 권한 부여해야 합니다.

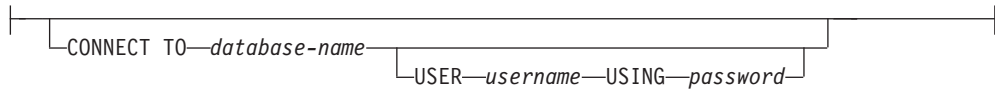
### 필수 연결

데이터베이스

### 명령 구문

```
▶▶ CLEAR COMMAND LOCKS ───────────────────┬── FOR TEXT ───────────────────▶
      └── FOR INDEX index-name ───────────┘
▶ ─── connection options ───────────────────▶▶
```

## connection options:



## 명령 매개변수

### FOR INDEX *index-name*

CREATE INDEX 명령에 지정된 인덱스의 이름.

### CONNECT TO *database-name*

이 절은 연결이 설정될 데이터베이스를 지정합니다. 데이터베이스는 로컬 시스템에 있어야 합니다. 이 절이 지정된 경우 DB2DBDFT 환경 변수에 우선합니다. 다음의 조건에 모두 해당되는 경우 이 절을 생략할 수 있습니다.

- DB2DBDFT 환경 변수가 유효한 데이터베이스 이름으로 설정됩니다.
- 명령을 실행하는 사용자에게 데이터베이스 서버에 연결하기 위해 필요한 권한이 있습니다.

### USER *username* USING *password*

이 절은 연결을 설정하는 데 사용될 권한 부여 이름과 암호를 지정합니다.

## 사용 시 참고사항

이 명령은 명령을 소유하는 프로세스가 종료된 경우 호출합니다. 이 경우 명령(잠금으로 표시)이 완료되지 않고 인덱스가 실행되지 않을 수 있습니다. 적절한 조치를 수행해야 합니다. 예를 들어, DROP INDEX 명령을 실행하는 프로세스가 갑자기 종료됩니다. 이 경우 일부 인덱스 데이터가 삭제되었지만 모든 카탈로그 및 컬렉션 정보가 삭제되지는 않았습니다. 명령 잠금은 그대로 유지됩니다. DROP INDEX 명령 잠금을 지운 후 DROP INDEX 명령을 재실행할 수 있습니다. 또는 CREATE INDEX 명령을 실행하는 프로세스가 갑자기 종료됩니다. 이 경우 일부 인덱스 카탈로그 및 컬렉션 정보를 작성했지만 모든 정보를 삭제하지는 않습니다. 명령 잠금은 그대로 유지됩니다. CREATE INDEX 명령 잠금을 지우면 DROP INDEX 및 CREATE INDEX 명령을 실행할 수 있습니다.

이 명령이 실행되면 DB2 텍스트 검색 보기 SYSIBMTS.TSLOCKS의 콘텐츠가 갱신됩니다.

---

## 제 283 장 db2ts CLEAR EVENTS FOR INDEX

이 명령은 관리에 사용되는 인덱스의 이벤트 테이블에서 인덱싱 이벤트를 삭제합니다. 이 테이블의 이름은 EVENTVIEWNAME 컬럼의 SYSIBMTS.TSINDEXES 뷰에 있습니다.

모든 인덱스 갱신 조작은 이벤트 테이블에서 정보 및 경우에 따라 오류 항목을 생성합니다. 자동 갱신의 경우 이 테이블을 정기적으로 검사해야 합니다. 문서 특정 오류는 문서 콘텐츠를 변경하여 정정해야 합니다. 오류를 정정한 후 이벤트를 지울 수 있으며 너무 많은 공간을 소비하지 않도록 지워야 합니다.

이 명령을 실행하려면 명령행에서 db2ts를 접두부로 추가해야 합니다.

### 권한 부여

명령문의 권한 부여 ID는 적어도 다음과 같은 특권 중 하나를 가지고 있어야 합니다.

- 인덱스가 정의되는 테이블에 대한 CONTROL 특권
- DBADM 권한

### 전제조건

인스턴스 소유자는 DATAACCESS 권한이 있는 DBADM을 보유해야 합니다. SYSADM은 더 이상 V9.7에서 SECADM 및 DBADM 특권을 보유하지 않습니다. SECADM은 CLEAR EVENTS 명령을 실행하기 전에 인스턴스 소유자에게 DATAACCESS 권한이 있는 DBADM을 명시적으로 권한 부여해야 합니다.

### 필수 연결

데이터베이스

### 명령 구문

```
▶▶—CLEAR EVENTS FOR INDEX—index-name—FOR TEXT—| connection options |————▶▶
```

**connection options:**

```
|—————|  
|—CONNECT TO—database-name—|  
|—USER—username—USING—password—|
```

## 명령 매개변수

*index-name*

CREATE INDEX 명령에 지정된 인덱스의 이름. 인덱스 이름은 DB2 인덱스의 이름 지정 제한사항을 준수해야 합니다.

**CONNECT TO** *database-name*

이 절은 연결이 설정될 데이터베이스를 지정합니다. 데이터베이스는 로컬 시스템에 있어야 합니다. 이 절이 지정된 경우 DB2DBDFT 환경 변수에 우선합니다. 다음의 조건에 모두 해당되는 경우 이 절을 생략할 수 있습니다.

- DB2DBDFT 환경 변수가 유효한 데이터베이스 이름으로 설정됩니다.
- 명령을 실행하는 사용자에게 데이터베이스 서버에 연결하기 위해 필요한 권한이 있습니다.

**USER** *username* **USING** *password*

이 절은 연결을 설정하는 데 사용될 권한 부여 이름과 암호를 지정합니다.

## 사용 시 참고사항

DB2 데이터베이스 오브젝트 및 쿼리에 적용되는 모든 한계와 이름 지정 규칙은 DB2 텍스트 검색 기능과 쿼리에도 적용됩니다. DB2 텍스트 검색 관련 ID는 DB2 이름 지정 규칙을 준수해야 합니다. 또한 몇 가지 추가 제한사항도 있습니다. 예를 들어, 이러한 ID의 양식은 다음과 같습니다.

```
[A-Za-z][A-Za-z0-9@#$_]*
```

또는

```
"[A-Za-z ][A-Za-z0-9@#$_ ]*"
```

정기적인 갱신이 스케줄되는 경우(CREATE INDEX 또는 ALTER INDEX 명령의 UPDATE FREQUENCY 옵션 참조) 이벤트 테이블을 정기적으로 점검해야 합니다. 텍스트 검색 인덱스의 DB2 텍스트 검색 이벤트 테이블을 정리하려면 이벤트 이유를 점검하고 오류 소스를 제거한 후 CLEAR EVENTS FOR INDEX 명령을 사용하십시오.

이벤트 테이블에서 참조되는 모든 행을 변경해야 합니다. 사용자 테이블의 행을 변경함으로써 이전에 오류가 있던 문서를 다시 인덱싱하여 다음 UPDATE INDEX 시도를 수행할 수 있습니다.

충돌 가능성이 있는 경우에는 텍스트 검색 인덱스에서 동시에 여러 명령을 실행할 수 없습니다. 충돌 명령이 실행 중일 때 이 명령이 실행되는 경우 오류가 발생하여 명령이 실패하며 이후 명령 실행을 다시 시도할 수 있습니다. 충돌 명령의 예는 다음과 같습니다.

- CLEAR EVENTS FOR INDEX

- UPDATE INDEX
- ALTER INDEX
- DROP INDEX
- DISABLE DATABASE FOR TEXT
- STOP FOR TEXT

데이터베이스 변경사항: 이벤트 테이블이 지워집니다.



---

## 제 284 장 db2ts CREATE INDEX

텍스트 검색 기능을 사용하여 컬럼 데이터를 검색할 수 있는 텍스트 컬럼의 텍스트 검색 인덱스를 작성합니다.

인덱스에는 인덱스에 정의된 갱신 빈도에 따라 사용자가 텍스트 검색 UPDATE INDEX 명령을 명시적으로 실행하거나 텍스트 검색 서비스로 내재적으로 실행할 때까지 데이터가 포함되지 않습니다.

이 명령을 실행하려면 명령행에서 db2ts를 접두부로 추가해야 합니다.

### 권한 부여

명령문의 권한 부여 ID는 적어도 다음과 같은 특권 중 하나를 가지고 있어야 합니다.

다음 중 하나

- 인덱스가 정의되는 테이블에 대한 CONTROL 특권
- 인덱스가 정의되는 테이블에 대한 INDEX 특권

다음 중 하나이어야 합니다.

- 인덱스의 내재적 또는 명시적 스키마 이름이 존재하지 않을 경우, 데이터베이스에 대한 IMPLICIT\_SCHEMA 권한
  - 인덱스의 스키마 이름이 기존의 스키마를 참조할 경우, 스키마에 대한 CREATEIN 특권
- DBADM 권한

### 전제조건

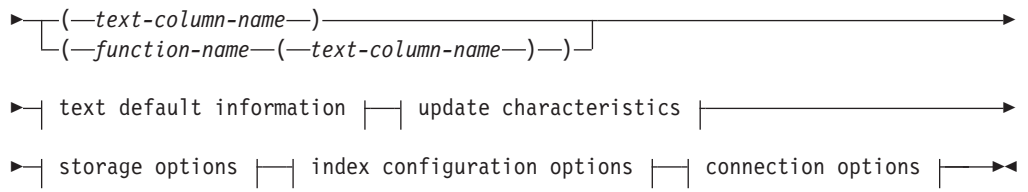
인스턴스 소유자는 DATAACCESS 권한이 있는 DBADM을 보유해야 합니다. SYSADM은 더 이상 V9.7에서 SECADM 및 DBADM 특권을 보유하지 않습니다. SECADM은 CREATE INDEX 명령을 실행하기 전에 인스턴스 소유자에게 DATAACCESS 권한이 있는 DBADM을 명시적으로 권한 부여해야 합니다.

### 필수 연결

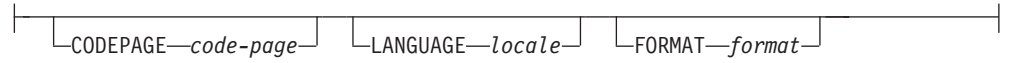
데이터베이스

### 명령 구문

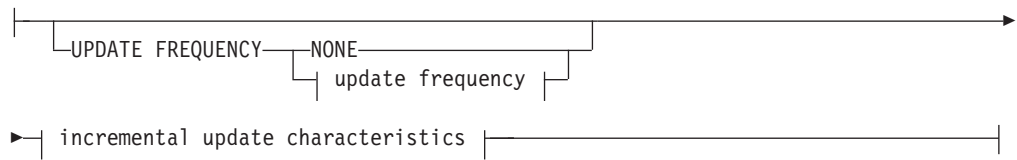
►►—CREATE INDEX—*index-name*—FOR TEXT—ON—*table-name*—►►



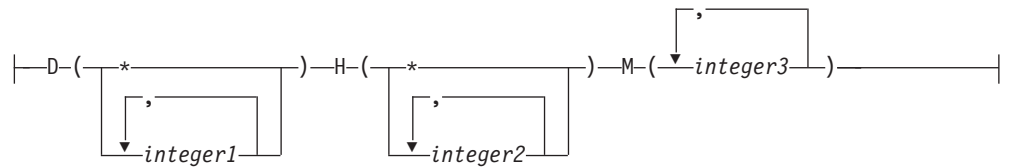
**text default information:**



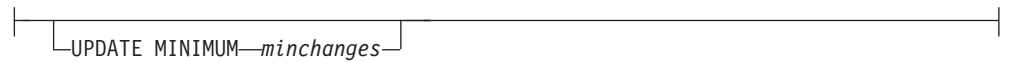
**update characteristics:**



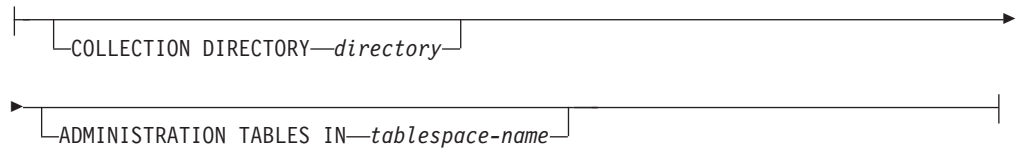
**update frequency:**



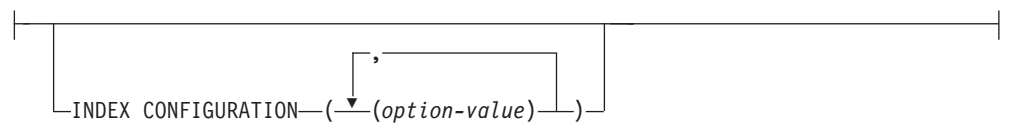
**incremental update characteristics:**



**storage options:**

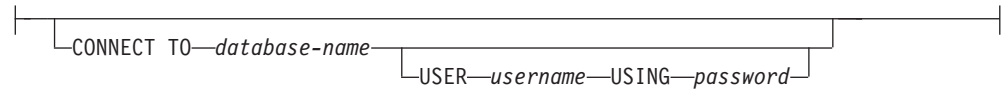


**index configuration options:**





## connection options:



## 명령 매개변수

### CREATE INDEX *index-name*

작성될 인덱스의 이름을 지정합니다. 인덱스에는 선택적으로 스키마가 규정될 수 있으며 이름은 데이터베이스에서 텍스트 검색 인덱스를 고유하게 식별합니다. 인덱스 이름은 DB2 인덱스의 이름 지정 제한사항을 준수해야 합니다.

### ON *table-name*

텍스트 컬럼을 포함하는 테이블 이름. 다음 테이블에서는 텍스트 검색 인덱스를 작성할 수 없습니다.

- 범위 파티션 테이블
- 페더레이티드 테이블
- 구체화된 쿼리 테이블
- 뷰

### *text-column-name*

인덱스화될 컬럼의 컬럼 이름. 컬럼의 올바른 데이터 유형은 CHAR, VARCHAR, LONG VARCHAR, CLOB, DBCLOB, BLOB, GRAPHIC, VARGRAPHIC, LONG VARGRAPHIC 또는 XML입니다. 컬럼의 데이터 유형이 이러한 유형이 아닌 경우 *function-schema.function-name*으로 지정된 변환 함수를 사용하여 컬럼 유형을 유효한 유형 중 하나로 변환하십시오. 또는 인덱스화될 텍스트 문서에 액세스하는 사용자 정의 외부 함수를 지정할 수 있습니다. 하나의 컬럼에 단일 텍스트 검색 인덱스만 작성할 수 있습니다.

### *function-name(text-column-name)*

텍스트 검색에 지원되는 유형이 아닌 컬럼에서 텍스트 문서에 액세스하며 DB2 이름 지정 규칙을 준수하는 외부 스칼라 함수의 스키마 규정 이름을 지정합니다. 해당 값의 데이터 유형 변환을 수행하고 텍스트 검색에 지원되는 데이터 유형 중 하나로 값을 리턴합니다. 해당 태스크는 컬럼 자료형 변환을 수행하는 것입니다. 이 함수는 하나의 매개변수만 사용하고 하나의 값만 리턴해야 합니다.

### CODEPAGE *code-page*

텍스트 문서를 인덱싱할 때 사용될 DB2 코드 페이지(CODEPAGE)를 지정합니다. 디폴트값은 SYSIBMTS.TSDEFAULTS의 값으로 지정됩니다. 여기서 DEFAULTNAME은 'CODEPAGE'(데이터베이스 코드 페이지)입니다. 이 인수는 2진 데이터 유형에만 적용됩니다. 즉, 변환 함수의 컬럼 유형 또는 리턴 유형은 BLOB 또는 FOR BIT DATA 문자 유형이어야 합니다.

## LANGUAGE *locale*

인덱싱 중에 문서의 언어 특정 처리를 위해 DB2 텍스트 검색에서 사용될 언어를 지정합니다. 로케일을 지정하지 않으면 데이터베이스 지역을 사용하여 LANGUAGE의 디폴트 설정을 판별합니다. 문서를 자동으로 스캔하여 로케일을 판별하려면 *locale*을 AUTO로 지정합니다.

## FORMAT *format*

컬럼에서 텍스트 문서의 형식을 지정합니다. 지원되는 형식은 TEXT, XML 및 HTML 등입니다. DB2 텍스트 검색에서는 문서를 인덱싱할 때 이 정보가 필요합니다. 형식이 지정되지 않으면 디폴트값이 사용됩니다. 디폴트값은 SYSIBMTS.TSDEFAULTS 뷰(DEFAULTNAME='FORMAT')에 있습니다. XML 데이터 유형 컬럼의 경우 DEFAULTNAME 값에 관계없이 디폴트 형식 'XML'이 사용됩니다.

## UPDATE FREQUENCY

인덱스 갱신 빈도를 지정합니다. 인덱스는 변경 수가 UPDATE MINIMUM에 설정된 값 이상인 경우 갱신됩니다. 갱신 빈도 NONE은 인덱스가 더 이상 갱신되지 않음을 나타냅니다. 이 옵션은 변경되지 않는 데이터가 있는 테이블의 텍스트 컬럼에 유용합니다. 또한 사용자가 UPDATE INDEX 명령을 사용하여 인덱스를 수동으로 갱신하려는 경우에도 유용합니다. 자동 갱신은 START FOR TEXT 명령이 실행되고 DB2 텍스트 검색 인스턴스 서비스가 실행 중인 경우에만 수행될 수 있습니다.

디폴트 빈도 값은 SYSIBMTS.TSDEFAULTS 뷰 (DEFAULTNAME='UPDATEFREQUENCY')에서 가져옵니다.

### NONE

더 이상 인덱스가 갱신되지 않습니다. 갱신은 수동으로 시작해야 합니다.

### D 인덱스가 갱신된 요일.

\* 모든 요일.

*integer1*

일요일에서 토요일까지의 특정 요일(0 - 6)

### H 인덱스가 갱신된 지정된 날짜의 시간.

\* 모든 시간.

*integer2*

자정에서 오후 11시까지 하루 중 특정 시간(0 - 23)

### M 인덱스가 갱신된 지정된 시간의 분.

*integer3*

시간의 맨 위(0) 또는 해당 시간 이후 5분 간격으로 증가하는 시간(0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50 또는 55)

## UPDATE MINIMUM *minchanges*

UPDATE FREQUENCY에 지정된 시간에 인덱스가 점진적으로 갱신되기 전에 텍스트 문서에 대한 최소 변경사항 수를 지정합니다. 양수 값만 허용됩니다. 디폴트 값은 SYSIBMTS.TSDEFAULTS 뷰(DEFAULTNAME='UPDATEFREQUENCY')에서 가져옵니다.

주: 이 값은 UPDATE INDEX 명령에서 무시됩니다(USING UPDATE MINIMUM 옵션이 사용되지 않는 경우). 작은 값으로도 테이블 컬럼과 텍스트 검색 인덱스 사이의 일관성이 증가합니다. 그러나 성능 오버헤드 또한 증가합니다.

## COLLECTION DIRECTORY *directory*

텍스트 검색 인덱스가 저장되는 디렉토리. 디폴트로, 콜렉션 데이터는 *DBPATH/NODExxxx/SQLxxxx/db2collections/index identifier/data*에 있습니다. 절대 경로를 지정해야 합니다. 절대 경로 이름의 최대 길이는 215자입니다.

## ADMINISTRATION TABLES IN *tablespace-name*

인덱스에 대해 작성되는 관리 테이블의 기존 일반 테이블 스페이스 이름을 지정합니다. 지정되지 않는 경우 인덱스가 작성되는 기본 테이블의 테이블 스페이스가 사용됩니다.

## INDEX CONFIGURATION (*option-value*)

추가 인덱스 관련 값을 옵션 값 문자열 쌍으로 지정합니다. 이러한 값은 작은따옴표로 묶어야 합니다.

주: 문자열 값 내부의 작은따옴표 문자는 두 개의 연속 작은따옴표로 나타내야 합니다. 지원되는 값은 다음과 같습니다.

표 40. 옵션 값 스펙

옵션	허용되는 값(디폴트값)	의미
COMMENT	512바이트보다 짧은 문자열 값.	DB2 텍스트 검색 카탈로그 뷰 TSINDEXES의 REMARKS 컬럼에 문자열 주석 값을 추가합니다. 또한 문자열 주석 값을 콜렉션의 설명으로서 추가합니다.

예: INDEX CONFIGURATION (COMMENT 'Index on User''s Guide column')

## CONNECT TO *database-name*

이 절은 연결이 설정될 데이터베이스를 지정합니다. 데이터베이스는 로컬 시스템에 있어야 합니다. 이 절이 지정된 경우 DB2DBDFT 환경 변수에 우선합니다. 다음의 조건에 모두 해당되는 경우 이 절을 생략할 수 있습니다.

- DB2DBDFT 환경 변수가 유효한 데이터베이스 이름으로 설정됩니다.
- 명령을 실행하는 사용자에게 데이터베이스 서버에 연결하기 위해 필요한 권한이 있습니다.

## USER *username* USING *password*

이 절은 연결을 설정하는 데 사용될 권한 부여 이름과 암호를 지정합니다.

## 사용 시 참고사항

DB2 데이터베이스 오브젝트 및 쿼리에 적용되는 모든 한계와 이름 지정 규칙은 DB2 텍스트 검색 기능과 쿼리에도 적용됩니다. DB2 텍스트 관련 ID는 DB2 이름 지정 규칙을 준수해야 합니다. 또한 몇 가지 추가 제한사항도 있습니다. 예를 들어, 이러한 ID의 양식은 다음과 같습니다.

```
[A-Za-z][A-Za-z0-9@#$_]*
```

또는

```
"[A-Za-z ][A-Za-z0-9@#$_ ]*"
```

CREATE INDEX 명령 실행에 성공하는 경우:

- DB2 텍스트 검색 서버 데이터가 갱신됩니다. 다음 예와 같이 이름이 *instance\_database-name\_index-identifier\_number*인 컬렉션이 작성됩니다.

```
tigertail_MYTSDB_TS250517_0000
```

컬렉션 이름은 SYSIBMTS.TSCOLLECTIONNAMES 뷰(COLLECTIONNAME 컬럼)에서 검색할 수 있습니다.

- DB2 텍스트 검색 카탈로그 정보가 갱신됩니다. 적합한 DB2 인덱스와 함께 지정된 테이블 스페이스에 인덱스 스테이징 테이블이 작성됩니다. 또한 지정된 테이블 스페이스에 인덱스 이벤트 테이블이 작성됩니다.
- 새로 작성된 텍스트 검색 인덱스는 자동으로 채워지지 않습니다. UPDATE INDEX 명령은 채워질 텍스트 검색 인덱스에 대해 수동 또는 자동으로 실행되어야 합니다 (UPDATE FREQUENCY 옵션 스펙을 통해 인덱스에 대해 정의된 갱신 스케줄의 결과로서).
- DB2 데이터베이스 서버의 텍스트 검색 인덱스 데이터 파일은 갱신됩니다. 인스턴스의 각 인덱스마다 스케줄된 갱신 정보가 기록됩니다.

사용 제한사항:

- 테이블에 대한 기본 키가 정의되어야 합니다. DB2 텍스트 검색에서는 유형 제한 없이 다중 컬럼 DB2 기본 키를 사용할 수 있습니다. 기본 키 컬럼의 수는 DB2에서 허용되는 기본 키 컬럼의 수보다 작은 두 개 컬럼으로 제한됩니다.
- DB2 텍스트 검색 인덱스가 있는 테이블의 모든 기본 키 컬럼의 총 길이는 DB2에서 허용되는 최대 총 기본 키 길이보다 작은 15바이트로 제한됩니다. DB2 CREATE INDEX문 제한사항을 참조하십시오.

충돌 가능성이 있는 경우에는 텍스트 검색 인덱스에서 동시에 여러 명령을 실행할 수 없습니다. 충돌 명령이 실행 중일 때 이 명령이 실행되는 경우 오류가 발생하여 명령이 실패하며 이후 명령 실행을 다시 시도할 수 있습니다. 충돌 명령의 예는 다음과 같습니다.

- DROP INDEX
- UPDATE INDEX
- CLEAR EVENTS FOR INDEX
- ALTER INDEX
- DISABLE DATABASE FOR TEXT
- STOP FOR TEXT



---

## 제 285 장 db2ts DISABLE DATABASE FOR TEXT

이 명령은 ENABLE DATABASE FOR TEXT 명령으로 수행된 변경사항(예: 텍스트 검색 관련 테이블 및 뷰 삭제)을 약간 되돌립니다.

이 명령이 실행되면 다음과 같은 상황이 발생합니다.

- 데이터베이스에 대해 DB2 텍스트 검색 기능이 사용되지 않습니다.
- 다음과 같은 테이블 및 뷰가 삭제됩니다.
  - SYSIBMTS.TSDEFAULTS
  - SYSIBMTS.TSLOCKS
  - SYSIBMTS.TSINDEXES
  - SYSIBMTS.TSCONFIGURATION
  - SYSIBMTS.TSCOLLECTIONNAMES

테이블은 데이터베이스의 디폴트 테이블 스페이스(IBMDEFAULTGROUP)에서 제거됩니다.

- FORCE 옵션을 지정한 경우 모든 텍스트 인덱스 정보가 데이터베이스에서 제거되고 연관된 모든 콜렉션이 삭제됩니다. 또한 나머지 갱신 스케줄 정보를 제거하기 위해 텍스트 서비스가 갱신됩니다. 『db2ts DROP INDEX 명령』을 참조하십시오.

이 명령을 실행하려면 명령행에서 db2ts를 접두부로 추가해야 합니다.

### 권한 부여

명령문의 권한 부여 ID가 보유하는 특권에는 DBADM 권한이 포함되어야 합니다.

### 전제조건

인스턴스 소유자는 DATAACCESS 권한이 있는 DBADM을 보유해야 합니다. SYSADM은 더 이상 V9.7에서 SECADM 및 DBADM 특권을 보유하지 않습니다. SECADM은 DISABLE DATABASE 명령을 실행하기 전에 인스턴스 소유자에게 DATAACCESS 권한이 있는 DBADM을 명시적으로 권한 부여해야 합니다.

### 필수 연결

데이터베이스

## 명령 구문

▶▶—DISABLE DATABASE FOR TEXT—FORCE—| connection options —▶▶

### connection options:

|—CONNECT TO—*database-name*—|  
|—USER—*username*—USING—*password*—|

## 명령 매개변수

### FORCE

모든 텍스트 검색 인덱스가 데이터베이스에서 강제로 삭제되도록 지정합니다.

이 옵션을 지정하지 않고 해당 데이터베이스에 대해 텍스트 검색 인덱스가 정의된 경우 명령은 실패합니다.

이 옵션을 지정하고 DB2 텍스트 검색 서비스를 시작하지 않은 경우(db2ts START FOR TEXT 명령을 실행하지 않은 경우), 텍스트 검색 인덱스(컬렉션)가 삭제되지 않습니다.

### CONNECT TO *database-name*

이 절은 연결이 설정될 데이터베이스를 지정합니다. 데이터베이스는 로컬 시스템에 있어야 합니다. 이 절이 지정된 경우 DB2DBDFT 환경 변수에 우선합니다. 다음에 모두 해당되는 경우 이 절을 생략할 수 있습니다.

- DB2DBDFT 환경 변수가 유효한 데이터베이스 이름으로 설정됩니다.
- 명령을 실행하는 사용자에게 데이터베이스 서버에 연결하기 위해 필요한 권한이 있습니다.

### USER *username* USING *password*

이 절은 연결을 설정하는 데 사용될 권한 부여 이름과 암호를 지정합니다.

## 사용 시 참고사항

이 명령은 데이터베이스의 DB2 Net Search Extender 사용 가능 상태에는 영향을 주지 않습니다. ENABLE FOR TEXT 명령으로 작성된 DB2 텍스트 검색 카탈로그 테이블 및 뷰를 삭제합니다.

텍스트 검색 인덱스 정의가 있는 DB2 데이터베이스를 삭제하기 전에, 이 명령을 실행하여 텍스트 인덱스 및 컬렉션이 성공적으로 제거되었는지 확인하십시오.

일부 인덱스가 FORCE 옵션 사용으로 삭제될 수 없는 경우, 컬렉션 이름이 db2diag 로그 파일에 기록됩니다. 텍스트 검색 인덱스 명령 DISABLE DATABASE FOR TEXT가 CLP 명령 DROP DATABASE 이전에 실행되지 않은 경우, 텍스트 검색 인



텍스 서비스도 CLEANUP FOR TEXT 명령을 사용하여 제거해야 합니다. 인덱스 삭제에 대한 자세한 정보는 DROP INDEX 명령을 참조하고, 텍스트 검색 컬렉션 및 텍스트 검색 인덱스와의 해당 관계에 대한 자세한 정보는 CLEANUP FOR TEXT 명령을 참조하십시오.

주: 사용자는 사용되지 않는 컬렉션(즉, 텍스트 검색 서버에 정의된 상태로 남아 있지만 DB2에서 사용되지 않는 컬렉션)이 발생하도록 사용하면 안됩니다. 다음은 사용되지 않는 컬렉션이 발생하는 경우입니다.

- DISABLE DATABASE FOR TEXT 명령을 실행하지 않고 DROP DATABASE CLP 명령 또는 DROP TABLE 문을 실행하는 경우
- FORCE 옵션을 사용하여 DISABLE DATABASE FOR TEXT 명령을 실행하는 경우
- 기타 오류 조건. CLEANUP FOR TEXT 명령은 일부 시나리오에서 사용될 수 있습니다.

충돌 가능성이 있는 경우에는 텍스트 검색 인덱스에서 동시에 여러 명령을 실행할 수 없습니다. 충돌 명령이 실행 중일 때 이 명령이 실행되는 경우 오류가 발생하여 명령이 실패하며 이후 명령 실행을 다시 시도할 수 있습니다. 충돌 명령의 예는 다음과 같습니다.

- DROP INDEX
- UPDATE INDEX
- CLEAR EVENTS FOR INDEX
- ALTER INDEX
- DISABLE DATABASE FOR TEXT
- STOP FOR TEXT



---

## 제 286 장 db2ts DROP INDEX

기존 텍스트 검색 인덱스를 삭제(drop)합니다.

이 명령을 실행하려면 명령행에서 db2ts를 접두부로 추가해야 합니다.

### 권한 부여

명령문의 권한 부여 ID는 적어도 다음과 같은 특권 중 하나를 가지고 있어야 합니다.

- 인덱스가 정의되는 테이블에 대한 CONTROL 특권
- DBADM 권한

### 전제조건

인스턴스 소유자는 DATAACCESS 권한이 있는 DBADM을 보유해야 합니다. SYSADM은 더 이상 V9.7에서 SECADM 및 DBADM 특권을 보유하지 않습니다. SECADM은 DROP INDEX 명령을 실행하기 전에 인스턴스 소유자에게 DATAACCESS 권한이 있는 DBADM을 명시적으로 권한 부여해야 합니다.

### 필수 연결

데이터베이스

### 명령 구문

►► DROP INDEX *index-name* FOR TEXT | connection options |◀◀

**connection options:**

| CONNECT TO *database-name* | USER *username* USING *password* |

### 명령 매개변수

#### **DROP INDEX** *index-name*

CREATE INDEX 명령에 지정된 인덱스의 스키마와 이름. 데이터베이스에서 텍스트 검색 인덱스를 고유하게 식별합니다.

#### **CONNECT TO** *database-name*

이 절은 연결이 설정될 데이터베이스를 지정합니다. 데이터베이스는 로컬 시스템에 있어야 합니다. 이 절이 지정된 경우 DB2DBDFT 환경 변수에 우선합니다. 다음의 조건에 모두 해당되는 경우 이 절을 생략할 수 있습니다.

- DB2DBDFT 환경 변수가 유효한 데이터베이스 이름으로 설정됩니다.
- 명령을 실행하는 사용자에게 데이터베이스 서버에 연결하기 위해 필요한 권한이 있습니다.

**USER** *username* **USING** *password*

이 절은 연결을 설정하는 데 사용될 권한 부여 이름과 암호를 지정합니다.

## 사용 시 참고사항

충돌 가능성이 있는 경우에는 텍스트 검색 인덱스에서 동시에 여러 명령을 실행할 수 없습니다. 충돌 명령이 실행 중일 때 이 명령이 실행되는 경우 오류가 발생하여 명령이 실패하며 이후 명령 실행을 다시 시도할 수 있습니다. 충돌 명령의 예는 다음과 같습니다.

- DROP INDEX
- UPDATE INDEX
- CLEAR EVENTS FOR INDEX
- ALTER INDEX
- DISABLE DATABASE FOR TEXT
- STOP FOR TEXT

DB2에서 사용자 테이블을 삭제해도 텍스트 검색 인덱스는 삭제되지 않습니다. 이 인덱스는 테이블 삭제 전후에(올바른 시점은 테이블 삭제 전) 수동으로 삭제해야 합니다. 텍스트 검색 인덱스가 삭제되면 해당 텍스트 컬럼에서 더 이상 텍스트를 검색할 수 없습니다. 동일한 텍스트 컬럼에서 새 텍스트 검색을 작성하려면 먼저 데이터베이스 연결을 끊은 다음 새 텍스트 검색 인덱스를 작성하기 전에 다시 연결해야 합니다.

데이터베이스 변경사항:

- DB2 텍스트 검색 카탈로그 정보를 갱신하십시오.
- 인덱스 스테이징/이벤트 테이블을 삭제하십시오.
- 사용자 텍스트 테이블에서 트리거를 삭제하십시오.
- DB2 텍스트 검색 인덱스 정의와 연관된 컬렉션이 파괴됩니다.

---

## 제 287 장 db2ts ENABLE DATABASE FOR TEXT

현재 데이터베이스에 DB2 텍스트 검색을 사용 가능하게 합니다. 데이터베이스 내 테이블의 컬럼에 대한 텍스트 검색 인덱스를 작성하려면 이 명령이 성공적으로 실행되어야 합니다.

이 명령을 실행하려면 명령행에서 db2ts를 접두부로 추가해야 합니다.

### 권한 부여

사용자는 ENABLE DATABASE 명령을 실행하기 위한 DBADM 특권을 가지고 있어야 합니다.

### 전제조건

인스턴스 소유자는 DATAACCESS 권한이 있는 DBADM을 보유해야 합니다. SYSADM은 더 이상 V9.7에서 SECADM 및 DBADM 특권을 보유하지 않습니다. SECADM은 ENABLE DATABASE 명령을 실행하기 전에 인스턴스 소유자에게 DATAACCESS 권한이 있는 DBADM을 명시적으로 권한 부여해야 합니다.

### 필수 연결

데이터베이스

### 명령 구문

▶▶—ENABLE DATABASE FOR TEXT—| connection options |▶▶▶▶

**connection options:**

|  
|—CONNECT TO—*database-name*—|  
|—USER—*username*—USING—*password*—|

### 명령 매개변수

**CONNECT TO** *database-name*

이 절은 연결이 설정될 데이터베이스를 지정합니다. 데이터베이스는 로컬 시스템에 있어야 합니다. 이 절이 지정된 경우 DB2DBDFT 환경 변수에 우선합니다. 다음에 모두 해당되는 경우 이 절을 생략할 수 있습니다.

- DB2DBDFT 환경 변수가 유효한 데이터베이스 이름으로 설정됩니다.
- 명령을 실행하는 사용자에게 데이터베이스 서버에 연결하기 위해 필요한 권한이 있습니다.

**USER** *username* **USING** *password*

이 절은 연결을 설정하는 데 사용될 권한 부여 이름과 암호를 지정합니다.

## 사용 시 참고사항

이 명령이 성공적으로 실행되면 다음과 같은 기능을 수행합니다.

- 데이터베이스에 대해 DB2 텍스트 검색 기능을 활성화합니다.
- SYSIBMTS.TSDEFAULTS 뷰에서 DB2 텍스트 검색 데이터베이스 구성 디폴트값을 설정합니다.
- SYSIBMTS 스키마에 다음 DB2 텍스트 검색 관리 뷰를 작성합니다.
  - SYSIBMTS.TSDEFAULTS
  - SYSIBMTS.TSLOCKS
  - SYSIBMTS.TSINDEXES
  - SYSIBMTS.TSCONFIGURATION
  - SYSIBMTS.TSCOLLECTIONNAMES

테이블은 데이터베이스의 디폴트 테이블 스페이스(IBMDEFAULTGROUP)에 작성됩니다.

DB2 서버의 파일 시스템 변경사항: 없음.

---

## 제 288 장 db2ts HELP

HELP는 사용 가능한 DB2 텍스트 검색 명령 또는 개별 명령의 구문의 목록을 표시합니다.

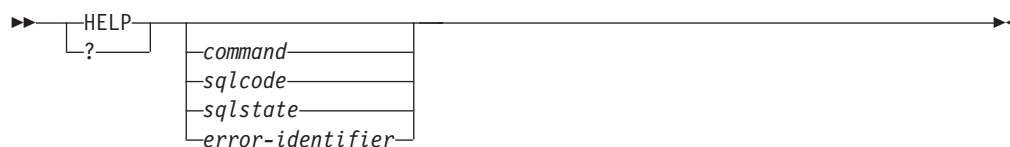
db2ts HELP 명령을 사용하여 특정 오류 메시지에 대한 도움말을 가져올 수도 있습니다.

이 명령을 실행하려면 명령행에서 db2ts를 접두부로 추가해야 합니다.

### 권한 부여

없음

### 명령 구문



### 명령 매개변수

#### HELP | ?

명령 또는 이유 코드에 대한 도움말 정보를 제공합니다.

#### *command*

DB2 텍스트 검색 명령을 식별하는 첫 번째 키워드:

- ENABLE
- DISABLE
- CREATE
- DROP
- ALTER
- UPDATE
- CLEAR(CLEAR COMMAND LOCKS 및 CLEAR EVENTS FOR INDEX에 모두 해당)
- CLEANUP
- START
- STOP

### *sqlcode*

db2ts 명령(관리 스토어드 프로시저 내부 또는 외부) 또는 텍스트 검색 쿼리로 리턴된 메시지의 SQLCODE.

### *sqlstate*

명령, 관리 스토어드 프로시저 또는 텍스트 검색 쿼리로 리턴된 Sqlstate.

### *error-identifier*

ID는 오류 메시지에 임베드(embed)된 *text-search-error-msg*의 일부입니다. 이 ID는 'CIE'로 시작되며 해당 양식은 CIE#####입니다. 여기서 #####은 숫자입니다. 이 ID는 텍스트 검색 중 오류에 대해 리턴되는 특정 오류를 나타냅니다. 텍스트 검색 명령 완료 시 정보 메시지 또는 텍스트 검색 관리 프로시저 완료 시 인쇄되는 메시지에 리턴될 수도 있습니다. ID가 'CIE'로 시작되지 않는 경우 db2ts help는 *error-identifier*에 대한 정보를 제공할 수 없습니다. 예를 들어, db2ts는 IQQR0012E와 같은 *error-identifier*가 있는 메시지에 대한 도움말을 제공할 수 없습니다.

## 사용 시 참고사항

UNIX 셸을 사용하는 경우 다음 예와 같이 큰따옴표를 사용하여 db2ts에 인수를 제공해야 합니다.

```
db2ts "? CIE00323"
```

인용 부호가 없으면 셸이 작업 디렉토리의 콘텐츠와 와일드카드를 비교하여 예기치 않은 결과가 발생할 수 있습니다.

db2ts 명령의 첫 번째 키워드가 지정되면 식별된 명령의 구문이 표시됩니다. 동일한 첫 번째 키워드를 공유하는 두 개 db2ts 명령(CLEAR COMMAND LOCKS 및 CLEAR EVENTS FOR INDEX)의 경우 두 명령 모두의 구문은 db2ts help clear가 실행될 때 표시되지만, 두 명령을 구분하는 두 번째 키워드(예를 들어, db2ts help clear events)를 추가하여 각 명령을 구체적으로 표시할 수 있습니다. ? 또는 HELP 다음에 매개변수가 지정되지 않으면 db2ts가 사용 가능한 모든 db2ts 명령을 나열합니다.

*sqlcode*, *sqlstate* 또는 CIE *error-identifier*를 지정하면 해당 코드, 상태 또는 오류 ID에 대한 정보가 리턴됩니다. 예를 들어, 다음과 같습니다.

```
db2ts help SQL20423
```

또는

```
db2ts ? 38H10
```

또는

```
db2ts ? CIE00323
```



주: 다음 텍스트 검색 명령의 경우 이 시점에서는 나열된 명령 도움말 옵션이 유효하지 않습니다.

**db2ts DISABLE DATABASE FOR TEXT**

[SERVER]

**db2ts ENABLE DATABASE FOR TEXT**

[SERVER]

**db2ts UPDATE INDEX**

[REORGANIZE] 및 [PARSE ONLY]



---

## 제 289 장 db2ts START FOR TEXT

이 명령의 기능은 다음과 같습니다.

- 다른 DB2 텍스트 검색 관리 명령을 지원하는 DB2 텍스트 검색 인스턴스 서비스와 SQL 쿼리에서 텍스트 검색 인덱스를 참조하는 기능을 시작합니다.
- DB2 데이터베이스 서버에서 실행되는 호스트 머신에서 서비스(UNIX 시스템의 디먼)를 시작합니다. 이러한 서비스는 DB2 데이터베이스의 텍스트 검색 인덱스 갱신 사항 스케줄링과 텍스트 검색 엔진 처리를 담당합니다. 인스턴스 서비스는 DB2 인스턴스 소유자의 권한 부여 이름으로 시작됩니다. 인스턴스 서비스가 이미 실행 중인 경우에는 명령이 영향을 주지 않습니다.

이 명령을 실행하려면 명령행에서 db2ts를 접두부로 추가해야 합니다.

### 권한 부여

인스턴스 소유자는 현재 DB2 인스턴스에 대한 DATAACCESS 권한이 있는 DBADM 을 보유해야 합니다.

### 필수 연결

이 명령은 DB2 데이터베이스 서버에서 실행되어야 합니다.

### 명령 구문

▶▶—START FOR TEXT—◀◀

### 명령 매개변수

없음



---

## 제 290 장 db2ts STOP FOR TEXT

이 명령은 DB2 텍스트 검색 인스턴스 서비스를 중지합니다.

이 명령을 실행하려면 명령행에서 db2ts를 접두부로 추가해야 합니다.

### 권한 부여

인스턴스 소유자는 현재 DB2 인스턴스에 대한 DATAACCESS 권한이 있는 DBADM을 보유해야 합니다.

### 필수 연결

이 명령은 DB2 데이터베이스 서버에서 실행되어야 합니다.

### 명령 구문

▶—STOP FOR TEXT—▶

### 명령 매개변수

없음

### 사용 시 참고사항

충돌 가능성이 있는 경우에는 텍스트 검색 인덱스에서 동시에 여러 명령을 실행할 수 없습니다. 충돌 명령이 실행 중일 때 이 명령이 실행되는 경우 오류가 발생하여 명령이 실패하며 이후 명령 실행을 다시 시도할 수 있습니다. 현재 실행 중인 명령의 실행이 인터럽트되지 않도록 하려면, 중지 명령을 실행하기 전에 인덱스 갱신과 같은 다른 관리 명령이 계속 사용되고 있지 않도록 확인하십시오. 충돌 명령의 예는 다음과 같습니다.

- DROP INDEX
- UPDATE INDEX
- CLEAR EVENTS FOR INDEX
- ALTER INDEX
- DISABLE DATABASE FOR TEXT
- STOP FOR TEXT



---

## 제 291 장 db2ts UPDATE INDEX

이 명령은 색인이 연관될 텍스트 컬럼의 현재 콘텐츠를 반영하여 텍스트 검색 인덱스 (DB2 텍스트 검색의 컬렉션)를 갱신합니다. 갱신이 수행되는 동안에는 검색할 수 없습니다. 갱신이 완료될 때까지 부분적으로 갱신된 인덱스에 대해 검색이 수행됩니다.

이 명령을 실행하려면 명령행에서 db2ts를 접두부로 추가해야 합니다.

### 권한 부여

명령문의 권한 부여 ID는 적어도 다음과 같은 특권 중 하나를 가지고 있어야 합니다.

- 텍스트 인덱스가 정의되는 테이블에 대한 CONTROL 특권
- DATAACCESS 권한

### 전제조건

인스턴스 소유자는 DATAACCESS 권한이 있는 DBADM을 보유해야 합니다. SYSADM은 더 이상 V9.7에서 SECADM 및 DBADM 특권을 보유하지 않습니다. SECADM은 UPDATE INDEX 명령을 실행하기 전에 인스턴스 소유자에게 DATAACCESS 권한이 있는 DBADM을 명시적으로 권한 부여해야 합니다.

### 필수 연결

데이터베이스

### 명령 구문

```
▶▶ UPDATE INDEX index-name FOR TEXT USING UPDATE MINIMUM
▶ | connection options |
```

**connection options:**

```
| CONNECT TO database-name USER username USING password |
```

### 명령 매개변수

**UPDATE INDEX** *index-name*

갱신될 텍스트 검색 인덱스의 이름을 지정합니다. 인덱스 이름은 DB2 인덱스의 이름 지정 제한사항을 준수해야 합니다.

## USING UPDATE MINIMUM

인덱스를 작성하기 위해 사용되는 CREATE INDEX 명령에 지정된 UPDATE MINIMUM *minchange* 설정을 사용해야 하는 것으로 지정하며 지정된 최소 횟수의 변경이 발생하면 증분 갱신을 시작합니다. 디폴트로, 무조건적으로 갱신이 시작됩니다.

## CONNECT TO *database-name*

이 절은 연결이 설정될 데이터베이스를 지정합니다. 데이터베이스는 로컬 시스템에 있어야 합니다. 이 절이 지정된 경우 DB2DBDFT 환경 변수에 우선합니다. 다음의 조건에 모두 해당되는 경우 이 절을 생략할 수 있습니다.

- DB2DBDFT 환경 변수가 유효한 데이터베이스 이름으로 설정됩니다.
- 명령을 실행하는 사용자에게 데이터베이스 서버에 연결하기 위해 필요한 권한이 있습니다.

## USER *username* USING *password*

이 절은 연결을 설정하는 데 사용될 권한 부여 이름과 암호를 지정합니다.

## 사용 시 참고사항

DB2 데이터베이스 오브젝트 및 쿼리에 적용되는 모든 한계와 이름 지정 규칙은 DB2 텍스트 검색 기능과 쿼리에도 적용됩니다. DB2 스트 검색 관련 ID는 DB2 이름 지정 규칙을 준수해야 합니다. 또한 몇 가지 추가 제한사항도 있습니다. 예를 들어, 이러한 ID의 양식은 다음과 같습니다.

```
[A-Za-z][A-Za-z0-9@#$_]*
```

또는

```
"[A-Za-z ][A-Za-z0-9@#$_]*"
```

이 명령은 모든 인덱스 갱신 처리가 완료될 때까지 리턴되지 않습니다. 지속기간은 인덱스화될 문서의 수와 이미 인덱스된 문서의 수에 따라 다릅니다. 인덱스의 컬렉션 이름은 SYSIBMTS.TSCOLLECTIONNAMES 뷰(COLLECTIONNAME 컬럼)에서 검색할 수 있습니다.

충돌 가능성이 있는 경우에는 텍스트 검색 인덱스에서 동시에 여러 명령을 실행할 수 없습니다. 충돌 명령이 실행 중일 때 이 명령이 실행되는 경우 오류가 발생하여 명령이 실패하며 이후 명령 실행을 다시 시도할 수 있습니다. 충돌 명령의 예는 다음과 같습니다.

- UPDATE INDEX
- CLEAR EVENTS FOR INDEX
- ALTER INDEX



- DROP INDEX
- DISABLE DATABASE FOR TEXT
- STOP FOR TEXT

주: 개별 문서 오류의 경우 문서를 정정해야 합니다. 오류 문서의 기본 키는 인덱스의 이벤트 테이블에서 검색할 수 있습니다. 다음 UPDATE INDEX 명령은 사용자 테이블의 해당 행이 수정된 경우 이 문서를 재처리합니다.

데이터베이스 변경사항:

- 이벤트 테이블에 행을 삽입합니다(DB2 텍스트 검색의 구문 분석기 오류 정보 포함).
- 증분 갱신의 경우 인덱스 스테이징 테이블에서 삭제합니다.
- 첫 번째 갱신 이전에 사용자 텍스트 테이블에서 트리거를 작성합니다.
- 컬렉션이 갱신됩니다.
- 새 문서 또는 변경된 문서의 구문을 분석하고 인덱스화됩니다.
- 삭제된 문서를 인덱스에서 버립니다.



---

## 제 7 부 부록



---

## 부록 A. 이름 지정 규칙

---

### 이름 지정 규칙

다음 규칙이 데이터베이스 및 테이블 같은 데이터베이스 관리 프로그램 오브젝트를 이름 지정할 때 적용됩니다.

- 데이터베이스 관리 프로그램 오브젝트의 이름을 표시하는 문자열은 a - z, A - Z, 0 - 9, @, # 및 \$를 포함할 수 있습니다.
- 별도로 언급되지 않는 경우, 이름을 소문자로 입력할 수 있습니다. 그러나 데이터베이스 관리 프로그램은 이름이 대문자인 것처럼 처리합니다.

이 규칙에 대한 예외는 통신 프로토콜로서 더 이상 지원되지 않는 SNA(Systems Network Architecture)에서 이름을 표시하는 문자열입니다. 논리 장치 이름(partner\_lu 및 local\_lu) 같은 많은 값은 대소문자를 구분합니다. 해당 항목에 대응하는 SNA 정의에서 나타나는 그대로 이름을 정확하게 입력해야 합니다.

- 데이터베이스 이름 또는 데이터베이스 별명은 위에서 설명한 세트의 1 - 8개의 문자, 숫자 또는 키보드 문자가 들어있는 고유한 문자열입니다.

데이터베이스는 한 필드에서는 별명, 다른 필드에서는 원래 이름으로 시스템 및 로컬 데이터베이스 디렉토리에서 카탈로그됩니다. 대부분의 기능에 대해 데이터베이스 관리 프로그램은 데이터베이스 디렉토리의 별명 필드에 입력되는 이름을 사용합니다. (예외는 디렉토리 경로를 지정해야 하는 CHANGE DATABASE COMMENT 및 CREATE DATABASE입니다.)

- 테이블 또는 뷰의 이름 또는 별명 이름은 길이가 1 - 128바이트인 고유한 문자열인 SQL ID입니다. 컬럼 이름의 길이는 1 - 128바이트일 수 있습니다.

완전한 테이블 이름은 *schema.tablename*으로 구성됩니다. 스키마는 테이블이 작성된 고유한 사용자 ID입니다. 선언된 임시 테이블에 대한 스키마 이름은 SESSION 이어야 합니다.

- 노드 디렉토리에서 카탈로그되어야 하는 리모트 노드에 대한 로컬 별명은 길이가 8 문자를 초과할 수 없습니다.
- 문자열의 첫 번째 문자는 영문자, @, # 또는 \$여야 합니다. 숫자 또는 문자 시퀀스 SYS, DBM 또는 IBM일 수 없습니다.

사용자 ID 및 인증 ID의 이름을 지정할 때 다음 규칙이 적용됩니다.

- 데이터베이스 관리 프로그램 오브젝트의 이름을 표시하는 문자열은 a - z, A - Z, 0 - 9, @, # 및 \$를 포함할 수 있습니다.

- 사용자 ID 및 그룹은 보안 플러그인에서 지원될 때 `_`, `!`, `%`, `(`, `)`, `{`, `}`, `-`, `.` 중 하나를 포함할 수도 있습니다.
- 다음 문자 중 하나를 포함하는 사용자 ID 및 그룹은 명령행 처리기를 통해 입력할 때 따옴표로 구분되어야 합니다. `!`, `%`, `(`, `)`, `{`, `}`, `-`, `..`, `^`,
- 문자열의 첫 번째 문자는 영문자, `@`, `#` 또는 `$`여야 합니다. 숫자 또는 문자 시퀀스 `SYS`, `DBM` 또는 `IBM`일 수 없습니다.
- 인증 ID는 길이가 128바이트를 초과할 수 없습니다.
- 그룹 ID는 길이가 128바이트를 초과할 수 없습니다.

---

## 부록 B. 파일 유형 수정자 및 분리문자

---

### 익스포트, 임포트 및 로드 유틸리티의 파일 유형 수정자

아래 링크를 통해 명령 주제 맨 위로 이동할 수 있습니다. 해당 위치에서 각 파일 유형 수정자 섹션에 대한 빠른 링크를 찾을 수 있습니다.

193 페이지의 제 43 장 『EXPORT』

283 페이지의 제 65 장 『IMPORT』

387 페이지의 제 85 장 『LOAD』

---

### 데이터 이동 시 분리문자 고려사항

컬럼 식별자가 있는 ASCII(DEL) 파일을 이동하는 경우 구분 문자 인식 문제점 때문에 이동할 데이터가 우연히 변경되지 않도록 해야 합니다. 이러한 오류를 방지하기 위해 DB2에서는 여러 제한사항을 시행하고 여러 파일 유형 수정자를 제공합니다.

#### 분리문자 제한사항

선택한 구분 문자를 이동할 데이터로 처리하지 않도록 하는 여러 제한사항이 있습니다. 먼저 분리문자는 상호 배타적입니다. 두 번째로 분리문자는 줄 바꾸기 문자, 캐리지 리턴 또는 공백일 수 없습니다. 또한 디폴트 소수점(.)은 문자열 분리문자일 수 없습니다. 마지막으로 DBCS 환경에서 파이프(|) 문자 분리문자는 지원되지 않습니다.

다음 문자는 ASCII 계열 코드 페이지 및 EBCDIC 계열 코드 페이지에서 서로 다르게 지정됩니다.

- 시프트 인(Shift-In) 및 시프트 아웃(Shift-Out) 문자는 EBCDIC MBCS 데이터 파일에서 분리문자로 사용할 수 없습니다.
- MBCS, EUC 또는 DBCS 코드 페이지의 분리문자는 0x40보다 클 수 없습니다. 단, EBCDIC MBCS 데이터의 디폴트 소수점(0x4b)은 제외됩니다.
- ASCII 코드 페이지 또는 EBCDIC MBCS 코드 페이지에서 데이터 파일의 디폴트 분리문자는 다음과 같습니다.
  - 문자열 분리문자: "(0x22, 큰따옴표)
  - 컬럼 분리문자: ,(0x2c, 쉼표)
- EBCDIC SBCS 코드 페이지의 데이터 파일에 대한 디폴트 분리문자는 다음과 같습니다.
  - 문자열 분리문자: "(0x7F, 큰따옴표 기호)

- 컬럼 분리문자: ,(0x6B, 쉼표)
- ASCII 데이터 파일의 디폴트 소수점은 0x2e(마침표)입니다.
- EBCDIC 데이터 파일의 디폴트 소수점은 0x4B(마침표)입니다.
- 서버의 코드 페이지가 클라이언트의 코드 페이지와 다른 경우 디폴트가 아닌 분리문자의 16진 표현을 지정해야 합니다. 예를 들어, 다음과 같습니다.

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

## 데이터 이동 중 분리문자에 대한 문제

### 2바이트 분리문자

디폴트로 DEL 파일의 문자 기반 필드에서 필드에 있는 문자 분리문자의 인스턴스는 2바이트 분리문자로 표시합니다. 예를 들어 문자 분리문자가 큰따옴표라고 가정하고 텍스트 "I am 6" tall."을 익스포트하면 DEL 파일의 출력 텍스트는 "I am 6"" tall."을 읽습니다. 반대로 DEL 파일의 입력 텍스트가 "What a ""nice"" day!"를 읽으면 텍스트는 What a "nice" day!

로 임포트됩니다. **nodoubledel**

2바이트 분리문자 동작은 **nodoubledel** 파일 유형 수정자를 지정하여 임포트, 익스포트 및 로드 유틸리티에서 사용하지 않도록 할 수 있습니다. 그러나 구문 분석 오류를 피하려면 2바이트 분리문자 동작이 있어야 합니다. 익스포트에서 **nodoubledel**을 사용하면 문자 필드에 있을 때 문자 분리문자가 이중으로 처리되지 않습니다. 임포트 및 로드에서 **nodoubledel**을 사용하면 2바이트 분리문자가 문자 분리문자의 리터럴 인스턴스로 해석되지 않습니다.

### **nocharde1**

익스포트에서 **nocharde1** 파일 유형 수정자를 사용하면 문자 필드가 문자 분리문자로 묶이지 않습니다. 임포트 및 로드에서 **nocharde1**을 사용하면 문자 분리문자가 특수 문자로 처리되지 않으며 실제 데이터로 해석됩니다.

### **charde1**

디폴트 분리문자 및 데이터를 혼동하지 않도록 기타 파일 유형 수정자를 사용할 수 있습니다. **charde1** 파일 유형 수정자는 단일 문자, x를 문자열 분리문자로 지정하여 큰따옴표(디폴트) 대신 사용합니다.

### **colde1**

마찬가지로 디폴트 쉼표를 컬럼 분리문자로 사용하지 않으려면 **colde1**을 사용하여 단일 문자, x를 열 데이터 분리문자로 지정합니다.

### **delprioritychar**

DEL 파일 이동과 관련된 또 다른 문제로 분리문자의 올바른 선행 순서를 유지해야 합



니다. 분리문자의 디폴트 우선순위는 행, 문자, 컬럼입니다. 그러나 일부 응용프로그램은 우선순위가 문자, 행, 컬럼이기도 합니다. 예를 들어 디폴트 우선순위를 사용하는 다음 DEL 데이터 파일이 있습니다.

```
"Vincent <row delimiter> is a manager",<row delimiter>
```

여기서 Vincent 및 is a manager와 같은 두 개 행을 포함한다고 해석됩니다. <row delimiter>가 문자 분리문자(")보다 우선되기 때문입니다. delprioritychar을 사용하면 문자 분리문자(")가 행 분리문자(<row delimiter>)보다 우선됩니다. 즉, 동일한 DEL 파일은 Vincent is a manager와 같은 하나의 행을 포함하는 것으로 해석됩니다.



---

## 부록 C. DB2 기술 정보 개요

DB2 기술 정보는 다음 도구 및 메소드를 통해 사용할 수 있습니다.

- DB2 정보 센터
  - 주제 항목(태스크, 개념 및 참조 항목)
  - DB2 도구에 대한 도움말
  - 샘플 프로그램
  - 자습서
- DB2 서적
  - PDF 파일(다운로드)
  - PDF 파일(DB2 PDF DVD)
  - 인쇄된 서적
- 명령행 도움말
  - 명령 도움말
  - 메시지 도움말

주: DB2 정보 센터의 주제는 PDF 또는 하드카피 서적보다 더 자주 갱신됩니다. 최신 정보를 보려면 사용 가능한 문서 갱신사항을 설치하거나 [ibm.com](http://ibm.com)에서 DB2 정보 센터를 참조하십시오.

[ibm.com](http://www.ibm.com)에서 추가 DB2 기술 정보(예: 기술 노트, 백서 및 IBM Redbooks® 서적)를 온라인으로 액세스할 수 있습니다. 다음은 DB2 정보 관리 라이브러리 소프트웨어 사이트의 주소입니다. <http://www.ibm.com/software/data/sw-library/>

### 문서 피드백

DB2 문서에 대한 피드백을 환영합니다. DB2 문서를 향상시키는 방법에 대해서 제안 사항이 있는 경우 [db2docs@ca.ibm.com](mailto:db2docs@ca.ibm.com)으로 전자 우편을 보내십시오. DB2 문서 팀에서는 고객의 모든 피드백을 읽지만 직접 응답할 수는 없습니다. 고객의 문제를 더 잘 이해할 수 있도록 가능한 한 구체적인 예를 제공하십시오. 특정 주제 또는 도움말 파일에 대한 피드백을 보내실 경우, 제목 및 URL을 알려주십시오.

DB2 고객 지원에 문의할 때는 이 전자 우편 주소를 사용하지 마십시오. 문서에서 해결할 수 없는 DB2 기술 문제점이 있는 경우, 해당 지역의 IBM 서비스 센터에 도움을 요청하십시오.

## DB2 기술 라이브러리(하드카피 또는 PDF 형식)

다음 표는 IBM Publications Center([www.ibm.com/shop/publications/order](http://www.ibm.com/shop/publications/order))에서 사용할 수 있는 DB2 라이브러리에 대한 설명입니다. PDF 형식의 영문 DB2 버전 9.7 매뉴얼 및 번역된 버전은 [www.ibm.com/support/docview.wss?rs=71&uid=swg2700947](http://www.ibm.com/support/docview.wss?rs=71&uid=swg2700947)에서 다운로드할 수 있습니다.

표에 인쇄할 수 있는 책으로 설명된 경우라도, 사용 국가 또는 지역에 따라 해당 책을 사용할 수 없을 수도 있습니다.

매뉴얼이 갱신될 때마다 문서 번호가 증가합니다. 다음 사항을 참조하여 읽고 있는 매뉴얼이 최신 버전인지 확인하십시오.

주: DB2 정보 센터는 PDF 또는 하드카피 서적보다 자주 갱신됩니다.

표 41. DB2 기술 정보

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
관리 API 참조서	SA30-3958-00	예	2009년 8월
관리 루틴 및 뷰	SA30-3955-00	아니오	2009년 8월
<i>Call Level Interface Guide and Reference, Volume 1</i>	SC27-2437-00	예	2009년 8월
<i>Call Level Interface Guide and Reference, Volume 2</i>	SC27-2438-00	예	2009년 8월
명령어 참조서	SA30-3959-00	예	2009년 8월
데이터 이동 유틸리티 안내서 및 참조서	SA30-3969-00	예	2009년 8월
데이터 복구 및 고가용성 안내서 및 참조서	SA30-3970-00	예	2009년 8월
데이터베이스 관리 개념 및 구성 참조서	SA30-3951-00	예	2009년 8월
데이터베이스 모니터링 안내서 및 참조서	SA30-3953-00	예	2009년 8월
데이터베이스 보안 안내서	SA30-3971-00	예	2009년 8월
<i>DB2 Text Search Guide</i>	SC27-2459-00	예	2009년 8월
<i>Developing ADO.NET and OLE DB Applications</i>	SC27-2444-00	예	2009년 8월
<i>Developing Embedded SQL Applications</i>	SC27-2445-00	예	2009년 8월
<i>Developing Java Applications</i>	SC27-2446-00	예	2009년 8월
<i>Developing Perl, PHP, Python, and Ruby on Rails Applications</i>	SC27-2447-00	아니오	2009년 8월

표 41. DB2 기술 정보 (계속)

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
<i>Developing User-defined Routines (SQL and External)</i>	SC27-2448-00	예	2009년 8월
<i>Getting Started with Database Application Development</i>	GI11-9410-00	예	2009년 8월
Linux 및 Windows에서 DB2 설치 및 관리 시작하기	GA30-3960-00	예	2009년 8월
자국어 안내서	SA30-3972-00	예	2009년 8월
DB2 Server 설치	GA30-3962-00	예	2009년 8월
IBM Data Server Client 설치	GA30-3963-00	아니오	2009년 8월
<i>Message Reference Volume 1</i>	SC27-2450-00	아니오	2009년 8월
<i>Message Reference Volume 2</i>	SC27-2451-00	아니오	2009년 8월
<i>Net Search Extender Administration and User's Guide</i>	SC27-2469-00	아니오	2009년 8월
파티셔닝 및 클러스터링 안내서	SA30-3973-00	예	2009년 8월
<i>pureXML Guide</i>	SC27-2465-00	예	2009년 8월
Query Patroller 관리 및 사용자 안내서	SA30-3974-00	아니오	2009년 8월
<i>Spatial Extender and Geodetic Data Management Feature User's Guide and Reference</i>	SC27-2468-00	아니오	2009년 8월
<i>SQL Procedural Languages: Application Enablement and Support</i>	SC27-2470-00	예	2009년 8월
SQL 참조서, 볼륨 1	SA30-3956-00	예	2009년 8월
SQL 참조서, 볼륨 2	SA30-3957-00	예	2009년 8월
문제점 해결 및 데이터베이스 성능 조정	SA30-3952-00	예	2009년 8월
DB2 버전 9.7로 업그레이드	SA30-3961-00	예	2009년 8월
Visual Explain 자습서	SA30-3968-00	아니오	2009년 8월
DB2 버전 9.7의 새로운 내용	SA30-3967-00	예	2009년 8월
<i>Workload Manager Guide and Reference</i>	SC27-2464-00	예	2009년 8월

표 41. DB2 기술 정보 (계속)

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
<i>XQuery Reference</i>	SC27-2466-00	아니오	2009년 8월

표 42. DB2 Connect 특정 기술 정보

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
<i>DB2 Connect Personal Edition 설치 및 구성</i>	SA30-3965-00	예	2009년 8월
<i>DB2 Connect Server 설치 및 구성</i>	SA30-3966-00	예	2009년 8월
<i>DB2 Connect 사용자 안내서</i>	SA30-3964-00	예	2009년 8월

표 43. Information Integration 기술 정보

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
<i>Information Integration: Administration Guide for Federated Systems</i>	SC19-1020-02	예	2009년 8월
<i>Information Integration: ASNCLP Program Reference for Replication and Event Publishing</i>	SC19-1018-04	예	2009년 8월
<i>Information Integration: Configuration Guide for Federated Data Sources</i>	SC19-1034-02	아니오	2009년 8월
<i>Information Integration: SQL Replication Guide and Reference</i>	SC19-1030-02	예	2009년 8월
<i>Information Integration: Introduction to Replication and Event Publishing</i>	GC19-1028-02	예	2009년 8월

## 인쇄된 DB2 서적 주문

인쇄된 DB2 서적이 필요한 경우, 대부분 온라인으로 구매할 수 있으나 모든 국가 또는 지역에서 가능한 것은 아닙니다. 언제든지 해당 지역의 IBM 담당자로부터 인쇄된 DB2 서적을 주문할 수 있습니다. *DB2 PDF* 문서 DVD의 일부 소프트웨어 서적은 인쇄할 수 없다는 점에 유의하십시오. 예를 들어, *DB2 메시지 참조서*의 볼륨은 인쇄된 서적으로 사용할 수 없습니다.

DB2 PDF 문서 DVD에서 사용할 수 있는 다수의 DB2 서적의 인쇄된 버전은 IBM에서 유료로 주문할 수 있습니다. 주문하는 위치에 따라 IBM Publications Center에서 온라인으로 서적을 주문할 수도 있습니다. 해당 국가 또는 지역에서 온라인 주문이

불가능하면, 언제든지 해당 지역의 IBM 담당자로부터 인쇄된 DB2 서적을 주문할 수 있습니다. DB2 PDF 문서 DVD의 모든 서적을 인쇄할 수는 없다는 점에 유의하십시오.

주: 가장 최신의 완전한 DB2 문서는 <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7>의 DB2 정보 센터에서 유지보수됩니다.

인쇄된 DB2 서적을 주문하려면 다음을 수행하십시오.

- 해당 국가 또는 지역에서 인쇄된 DB2 서적을 온라인으로 주문할 수 있는지 여부를 확인하려면 <http://www.ibm.com/shop/publications/order>의 IBM Publications Center를 확인하십시오. 서적 주문 정보를 액세스하려면 국가/지역/언어를 선택한 다음 해당 위치에서 주문 지시사항을 따르십시오.
- 해당 지역의 IBM 담당자로부터 인쇄된 DB2 서적을 주문하려면 다음을 수행하십시오.
  1. 다음 웹 사이트 중 하나에서 해당 지역 담당자에 대한 문의처 정보를 찾으십시오.
    - [www.ibm.com/planetwide](http://www.ibm.com/planetwide)에 있는 IBM 전세계 문의처 디렉토리
    - <http://www.ibm.com/shop/publications/order>의 IBM Publications 웹 사이트. 사용 지역의 해당 서적 홈 페이지에 액세스하려면 해당 국가, 지역 또는 언어를 선택해야 합니다. 이 페이지에서 "이 제품의 정보" 링크를 수행하십시오.
  2. 전화로 주문할 경우, 주문할 DB2 서적을 지정하십시오.
  3. 담당자에게 주문하려는 서적의 제목 및 문서 번호를 제공하십시오. 서적의 제목 및 문서 번호는 1322 페이지의 『DB2 기술 라이브러리(하드카피 또는 PDF 형식)』를 참조하십시오.

---

## 명령행 처리기에서 SQL 상태 도움말 표시

DB2 제품은 SQL문의 결과로 나타나는 상태에 대한 SQLSTATE 값을 리턴합니다. SQLSTATE 도움말은 SQL 상태 및 SQL 상태 클래스 코드의 의미를 설명합니다.

SQL 상태 도움말을 시작하려면 명령행 처리기를 열고 다음을 입력하십시오.

```
? sqlstate or ? class code
```

여기서, *sqlstate*는 유효한 5자리 숫자로 된 SQL 상태이고 *class code*는 SQL 상태의 처음 2자리 숫자를 나타냅니다.

예를 들어, ? 08003은 08003 SQL 상태에 대한 도움말을 표시하고, ? 08은 08 클래스 코드에 대한 도움말을 표시합니다.

---

## DB2 정보 센터의 다른 버전에 액세스

DB2 버전 9.7 주제에 대한 DB2 정보 센터 URL은 <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/>입니다.

DB2 버전 9.5 주제에 대한 DB2 정보 센터 URL은 <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/>입니다.

DB2 버전 9 주제에 대한 DB2 정보 센터 URL은 <http://publib.boulder.ibm.com/infocenter/db2luw/v9/>입니다.

DB2 버전 8 주제에 대한 버전 8 정보 센터 URL은 <http://publib.boulder.ibm.com/infocenter/db2luw/v8/>입니다.

---

## DB2 정보 센터에서 원하는 언어로 항목 표시

DB2 정보 센터는 브라우저 환경 설정에 지정된 언어로 주제 항목을 표시합니다. 주제가 원하는 언어로 변환되지 않은 경우, DB2 정보 센터는 해당 주제 항목을 영어로 표시합니다.

- Internet Explorer 브라우저에서 원하는 언어로 항목을 표시하려면 다음을 수행하십시오.

1. Internet Explorer에서 도구 → 인터넷 옵션 → 언어 단추를 누르십시오. 언어 환경 설정 창이 열립니다.
2. 원하는 언어가 언어 목록의 첫 번째 항목으로 지정되었는지 확인하십시오.
  - 목록에 새 언어를 추가하려면 추가... 단추를 누르십시오.

주: 언어를 추가하더라도 원하는 언어로 항목을 표시하는 데 필요한 글꼴이 컴퓨터에 설치되지 않습니다.

- 언어를 목록 맨위로 이동하려면, 언어를 선택한 후 해당 언어가 언어 목록의 첫 번째 항목이 될 때까지 위로 이동 단추를 누르십시오.
3. 브라우저 캐시를 지운 후 페이지를 새로 고치면 원하는 언어로 DB2 정보 센터가 표시됩니다.

- Firefox 또는 Mozilla 브라우저에서 원하는 언어로 주제 항목을 표시하려면 다음을 수행하십시오.

1. 도구 → 설정 → 내용 대화 상자의 언어 섹션에서 단추를 선택하십시오. 환경 설정 창에 언어 패널이 표시됩니다.
2. 원하는 언어가 언어 목록의 첫 번째 항목으로 지정되었는지 확인하십시오.
  - 목록에 새 언어를 추가하려면 언어 선택 창에서 원하는 언어를 선택한 다음 추가... 단추를 누르십시오.



- 언어를 목록 맨위로 이동하려면, 언어를 선택한 후 해당 언어가 언어 목록의 첫 번째 항목이 될 때까지 위로 이동 단추를 누르십시오.
3. 브라우저 캐시를 지운 후 페이지를 새로 고치면 원하는 언어로 DB2 정보 센터가 표시됩니다.

일부 브라우저 및 운영 체제 조합에서는 운영 체제의 국가별 설정을 선택한 로케일 및 언어로 변경해야 합니다.

## 컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 갱신

로컬로 설치된 DB2 정보 센터는 주기적으로 갱신해야 합니다.

### 시작하기 전에

DB2 버전 9.7 정보 센터는 미리 설치된 상태여야 합니다. 자세한 내용은 *DB2 Server* 설치의 『DB2 설치 마법사를 사용하여 DB2 정보 센터 설치』 주제를 참조하십시오. 정보 센터 설치에 적용되는 모든 전제조건 및 제한사항은 정보 센터 갱신에도 적용됩니다.

### 이 태스크에 대한 정보

기존의 DB2 정보 센터는 자동 또는 수동으로 갱신할 수 있습니다.

- 자동 갱신 - 기존 정보 센터 기능 및 언어를 갱신합니다. 자동 갱신의 또 다른 이점으로는 갱신 동안 정보 센터를 사용할 수 없는 시간이 매우 짧다는 점입니다. 또한 자동 갱신은 주기적으로 실행되는 기타 일괄처리 작업의 일부로 실행되도록 설정할 수도 있습니다.
- 수동 갱신 - 갱신 프로세스 중에 기능이나 언어를 추가하려는 경우 사용하십시오. 예를 들어, 로컬 정보 센터는 기본적으로 영어와 프랑스로 설치되어 있으며, 수동 갱신을 통해 기존 정보 센터의 기능 및 언어 갱신뿐만 아니라 독일어도 설치할 수 있습니다. 단, 수동 갱신을 수행하려면 정보 센터를 중지한 다음 갱신하고 재시작해야 합니다. 정보 센터는 갱신 프로세스 동안에는 사용할 수 없습니다.

### 프로시저

이 주제는 자동 갱신 프로세스에 대한 설명입니다. 수동 갱신에 대한 지시사항은 『컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 수동 갱신』 주제를 참조하십시오.

컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터를 자동으로 갱신하려면 다음을 수행하십시오.

1. Linux 운영 체제의 경우
  - a. 정보 센터가 설치된 경로를 찾아가십시오. DB2 정보 센터는 `/opt/ibm/db2ic/V9.7` 디렉토리에 디폴트로 설치됩니다.

- b. 설치 디렉토리에서 doc/bin 디렉토리로 이동하십시오.
- c. 다음과 같이 ic-update 스크립트를 실행하십시오.

```
ic-update
```

2. Windows 운영 체제의 경우

- a. 명령 창을 여십시오.
- b. 정보 센터가 설치된 경로를 찾아가십시오. DB2 정보 센터는 <Program Files>\IBM\DB2 Information Center\Version 9.7 디렉토리에 디폴트로 설치됩니다. 여기서 <Program Files>는 프로그램 파일 디렉토리의 위치를 나타냅니다.
- c. 설치 디렉토리에서 doc\bin 디렉토리로 이동하십시오.
- d. 다음과 같이 ic-update.bat 파일을 실행하십시오.

```
ic-update.bat
```

결과

DB2 정보 센터가 자동으로 재시작됩니다. 갱신사항이 사용 가능한 경우, 정보 센터에는 새로 갱신된 주제가 표시됩니다. 정보 센터 갱신을 사용할 수 없는 경우, 메시지가 로그에 추가됩니다. 로그 파일은 doc\weclipse\configuration 디렉토리에 있습니다. 이 로그 파일 이름은 임의로 생성된 번호입니다. 예: 1239053440785.log

## 컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 수동 갱신

DB2 정보 센터를 로컬로 설치한 경우, IBM으로부터 문서 갱신사항을 받아 설치할 수 있습니다.

로컬로 설치된 DB2 정보 센터를 수동으로 갱신하려면 다음을 수행하십시오.

1. 컴퓨터에서 DB2 정보 센터를 중지한 후 독립형 모드에서 다시 시작하십시오. 독립형 모드에서 정보 센터를 실행하면 사용자의 네트워크와 연결된 다른 사용자는 정보 센터에 액세스할 수 없으므로 갱신사항을 적용할 수 있습니다. DB2 정보 센터의 워크스테이션 버전은 항상 독립형 모드에서 실행됩니다.
2. 사용 가능한 갱신사항을 확인하려면 갱신 기능을 사용하십시오. 설치해야 할 갱신사항이 있는 경우, 갱신 기능을 사용하여 이를 가져온 후 설치할 수 있습니다.

주: 인터넷에 연결되지 않은 머신에 DB2 정보 센터 갱신사항을 설치해야 할 경우, 인터넷에 연결되고 DB2 정보 센터가 설치된 머신을 사용하여 갱신 사이트를 로컬 파일 시스템으로 미리하십시오. 네트워크 상에 문서 갱신사항을 설치하려는 사용자가 많을 경우에는 갱신 사이트를 로컬로 미리링하거나 갱신 사이트의 프록시를 작성하여 갱신을 수행하면 각 개인에게 필요한 시간을 줄일 수 있습니다.

갱신 패키지가 사용 가능하면 갱신 기능을 사용하여 패키지를 가져오십시오. 그러나 갱신 기능은 독립형 모드에서만 사용할 수 있습니다.

3. 독립형 정보 센터를 중지한 후 컴퓨터에서 DB2 정보 센터를 재시작하십시오.

주: Windows 2008, Windows Vista 이상의 경우 이 절 다음에 나오는 명령은 관리자 권한으로 실행해야 합니다. 전체 관리자 권한으로 명령 프롬프트 또는 그래픽 도구를 열려면 단축 아이콘을 마우스 오른쪽 단추로 누른 후 관리자 권한으로 실행을 선택하십시오.

컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터를 갱신하려면 다음을 수행하십시오.

1. DB2 정보 센터를 중지하십시오.

- Windows의 경우, 시작 → 제어판 → 관리 도구 → 서비스를 누르십시오. 그런 다음 **DB2 Information Center** 서비스를 마우스 오른쪽 단추로 누른 후 중지를 선택하십시오.
- Linux의 경우, 다음 명령을 입력하십시오.  
`/etc/init.d/db2icdv97 stop`

2. 독립형 모드에서 정보 센터를 시작하십시오.

- Windows의 경우:
  - a. 명령 창을 여십시오.
  - b. 정보 센터가 설치된 경로를 찾아가십시오. DB2 정보 센터는 <Program Files>#IBM#DB2 Information Center#Version 9.7 디렉토리에 디폴트로 설치됩니다. 여기서 <Program Files>는 프로그램 파일 디렉토리의 위치를 나타냅니다.
  - c. 설치 디렉토리에서 doc#bin 디렉토리로 이동하십시오.
  - d. 다음과 같이 help\_start.bat 파일을 실행하십시오.  
`help_start.bat`
- Linux의 경우:
  - a. 정보 센터가 설치된 경로를 찾아가십시오. DB2 정보 센터는 /opt/ibm/db2ic/V9.7 디렉토리에 디폴트로 설치됩니다.
  - b. 설치 디렉토리에서 doc/bin 디렉토리로 이동하십시오.
  - c. 다음과 같이 help\_start 스크립트를 실행하십시오.  
`help_start`

시스템의 기본 웹 브라우저가 열리고 독립형 정보 센터가 표시됩니다.

3. 갱신 단추(🔄)를 누르십시오. (JavaScript™가 브라우저에서 사용 가능해야 합니다.) 정보 센터의 오른쪽 패널에서 갱신사항 찾기를 누르십시오. 기존 문서의 갱신사항 목록이 표시됩니다.
4. 설치 프로세스를 시작하려면 설치할 선택란을 체크한 후 갱신사항 설치를 누르십시오.
5. 설치 프로세스가 완료되면 완료를 누르십시오.

## 6. 독립형 정보 센터를 중지하십시오.

- Windows의 경우, 설치 디렉토리의 doc\win 디렉토리로 이동한 후 다음과 같이 help\_end.bat 파일을 실행하십시오.

```
help_end.bat
```

주: help\_end 일괄처리 파일에는 help\_start 일괄처리 파일로 시작된 프로세스를 안전하게 중지하는 데 필요한 명령이 포함되어 있습니다. help\_start.bat 를 중지할 때 Ctrl+C 또는 다른 메소드를 사용하지 마십시오.

- Linux의 경우, 설치 디렉토리의 doc/bin 디렉토리로 이동한 후 다음과 같이 help\_end 스크립트를 실행하십시오.

```
help_end
```

주: help\_end 스크립트에는 help\_start 스크립트로 시작된 프로세스를 안전하게 중지하는 데 필요한 명령이 포함되어 있습니다. help\_start 스크립트를 중지할 때 다른 메소드를 사용하지 마십시오.

## 7. DB2 정보 센터를 재시작하십시오.

- Windows의 경우, 시작 → 제어판 → 관리 도구 → 서비스를 누르십시오. 그런 다음 **DB2 Information Center** 서비스를 마우스 오른쪽 단추로 누른 후 시작을 선택하십시오.
- Linux의 경우, 다음 명령을 입력하십시오.

```
/etc/init.d/db2icdv97 start
```

갱신된 DB2 정보 센터에는 새로 갱신된 주제가 표시됩니다.

---

## DB2 자습서

DB2 자습서는 DB2 제품의 여러가지 측면을 학습하는 데 유용합니다. 각 레슨은 단계별 지시사항을 제공합니다.

### 시작하기 전에

정보 센터(<http://publib.boulder.ibm.com/infocenter/db2help/>)에서 XHTML 버전의 자습서를 볼 수 있습니다.

일부 레슨에서는 샘플 데이터나 코드를 사용합니다. 특정 태스크에 필요한 전제조건 설명은 자습서를 참조하십시오.

### DB2 자습서

자습서를 보려면 제목을 누르십시오.

### 『pureXML®』(pureXML Guide)

DB2 데이터베이스를 설정하여 XML 데이터를 저장하고 원시 XML 데이터 스토어로 기본 조작을 수행할 수 있습니다.

### Visual Explain 자습서의 『Visual Explain』

더 나은 성능을 위해 Visual Explain을 사용하여 SQL문을 분석, 최적화 및 조정할 수 있습니다.

---

## DB2 문제점 해결 정보

DB2 데이터베이스 제품 사용 시 발생하는 광범위한 문제점을 판별하고 해결하는 데 도움이 되는 정보를 사용할 수 있습니다.

### DB2 문서

문제점 해결 정보는 *DB2 문제점 해결 안내서* 또는 *DB2 정보 센터*의 데이터베이스 기본 절을 참조하십시오. DB2 진단 도구 및 유틸리티를 사용하여 문제점을 찾아내고 식별하는 방법, 가장 일반적인 문제점에 대한 솔루션 및 DB2 데이터베이스 제품에서 발생할 수 있는 문제점을 해결하는 방법 등에 관한 정보가 있습니다.

### DB2 기술 지원 웹 사이트

문제점이 발생한 경우 해당 원인 및 솔루션을 찾으려면 DB2 기술 지원 웹 사이트를 참조하십시오. 기술 지원 사이트에는 최신 DB2 서적, 기술 노트, APAR(Authorized Program Analysis Report 또는 버그 수정), FixPack 및 기타 자원에 대한 링크가 있습니다. 이러한 기술 자료를 검색하여 문제에 대해 사용 가능한 솔루션을 찾을 수 있습니다.

다음은 DB2 기술 지원 웹 사이트의 주소입니다. [http://www.ibm.com/software/data/db2/support/db2\\_9/](http://www.ibm.com/software/data/db2/support/db2_9/)

---

## 이용약관

다음 조건에 따라 이 책을 사용할 수 있습니다.

**개인적 사용:** 모든 소유권 사항을 표시하는 경우에 한하여 귀하는 이 책을 개인적, 비상업적 용도로 복제할 수 있습니다. IBM의 명시적인 동의 없이는 이 책 또는 그 일부를 배포 또는 전시하거나 2차적 저작물을 만들 수 없습니다.

**상업적 사용:** 모든 소유권 사항을 표시하는 경우에 한하여 귀하는 이 책을 귀하 기업 집단 내에서만 복제, 배포 및 전시할 수 있습니다. 귀하는 IBM의 명시적 동의 없이 이 책의 2차적 저작물을 만들거나 이 책 또는 그 일부를 복제, 배포 또는 전시할 수 없습니다.

본 허가에서 명시적으로 부여된 경우를 제외하고, 이 책이나 이 책에 포함된 정보, 데이터, 소프트웨어 또는 기타 지적 재산권에 대한 어떠한 허가나 라이선스 또는 권한도 명시적 또는 묵시적으로 부여되지 않습니다.

IBM은 이 책의 사용이 IBM의 이익을 해친다고 판단되거나 위에서 언급된 지시사항이 준수되지 않는다고 판단하는 경우 언제든지 이 사이트에서 부여한 허가를 철회할 수 있습니다.

귀하는 미국 수출법 및 관련 규정을 포함하여 모든 적용 가능한 법률 및 규정을 철저히 준수하는 경우에만 본 정보를 다운로드, 송신 또는 재송신할 수 있습니다.

IBM은 이 책의 내용에 대해 어떠한 보증도 제공하지 않습니다. 타인의 권리 침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 (단 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증 없이 현 상태대로 제공합니다.

---

## 부록 D. 주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다. 비IBM 제품에 대한 정보는 이 책을 처음 발행할 때의 정보에 기초하고 있으며 변경될 수 있습니다.

IBM은 다른 국가에서 이 책에 기술된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산을 침해하지 않는 한, 기능상으로 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수도 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 라이선스까지 부여하는 것은 아닙니다. 라이선스에 대한 의문사항은 다음으로 문의하십시오.

135-700

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

전화번호: 080-023-8080

2바이트 문자 세트(DBCS) 정보에 관한 라이선스 문의는 한국 IBM 고객만족센터에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan, Ltd.

3-2-12, Roppongi, Minato-ku, Tokyo 106-8711 Japan

다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다. IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 (단, 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증없이 이 책을 『현상 태대로』 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책 사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 변경된 사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통지 없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 언급되는 비IBM의 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

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

(i) 독자적으로 작성된 프로그램과 다른 프로그램(본 프로그램 포함) 간의 정보 교환 및  
(ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 라이선스 사용자는 다음 주소로 문의하십시오.

135-700

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠. 주식회사

고객만족센터

이러한 정보는 해당 조건(예를 들면, 사용료 지불 등) 하에서 사용될 수 있습니다.

이 정보에 기술된 라이선스가 부여된 프로그램 및 프로그램에 대해 사용 가능한 모든 라이선스가 부여된 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이선스 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

본 문서에 포함된 모든 성능 데이터는 제한된 환경에서 산출된 것입니다. 따라서 다른 운영 환경에서 얻어진 결과는 상당히 다를 수 있습니다. 일부 측정치는 개발 레벨 시스템에서 작성되었을 수 있으며, 따라서 이러한 측정치가 일반적으로 사용되고 있는 시스템에서도 동일하게 나타날 것이라고는 보증할 수 없습니다. 또한 일부 성능은 추정을 통해 추측되었을 수도 있으므로 실제 결과는 다를 수 있습니다. 이 책의 사용자는 해당 데이터를 본인의 특정 환경에서 검증해야 합니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 다른 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 제품들을 테스트하지 않았으므로, 비IBM 제품과 관련된 성능의 정확성, 호환성 또는 기타 청구에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

IBM이 제시하는 방향 또는 의도에 관한 모든 언급은 특별한 통지 없이 변경될 수 있습니다.



이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이들 예제에는 개념을 가능한 완벽하게 설명하기 위하여 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 기업의 이름 및 주소와 유사하더라도 이는 전적으로 우연입니다.

#### 저작권 라이선스:

이 정보에는 여러 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 응용프로그램이 들어 있습니다. 귀하는 이러한 샘플 프로그램의 작성 기준이 되는 운영 플랫폼의 응용프로그램 프로그래밍 인터페이스(API)에 부합하는 응용프로그램을 개발, 사용, 판매 또는 배포할 목적으로 IBM에 추가 비용을 지불하지 않고 이들 샘플 프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다. 이러한 샘플 프로그램은 모든 조건하에서 완전히 테스트된 것은 아닙니다. 따라서 IBM은 이러한 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 진술하지 않습니다. 샘플 프로그램은 어떠한 보증없이 "있는 그대로" 제공됩니다. IBM은 샘플 프로그램의 사용으로 인해 발생하는 모든 손해에 대해 책임을 지지 않습니다.

이러한 샘플 프로그램 또는 파생 제품의 각 사본이나 일부에는 반드시 다음과 같은 저작권 표시가 포함되어야 합니다.

© (귀하의 회사명) (연도). 이 코드의 일부는 IBM Corp.의 샘플 프로그램에서 파생됩니다. © Copyright IBM Corp. *\_enter 연도\_*. All rights reserved.

#### 상표

IBM, IBM 로고 및 [ibm.com](http://ibm.com)<sup>®</sup>은 여러 국가에 등록된 International Business Machines Corp.의 상표 또는 등록상표입니다. 기타 제품 및 서비스 이름은 IBM 또는 기타 회사의 상표입니다. 현재 IBM 상표 목록은 웹 "저작권 및 상표 정보"([www.ibm.com/legal/kr/copytrade.shtml](http://www.ibm.com/legal/kr/copytrade.shtml))에 있습니다.

다음 용어는 기타 회사의 상표 또는 등록상표입니다.

- Linux는 미국 또는 기타 국가에서 사용되는 Linus Torvalds의 등록상표입니다.
- Java 및 모든 Java 기반 상표는 미국 또는 기타 국가에서 사용되는 Sun Microsystems, Inc.의 상표입니다.
- UNIX는 미국 또는 기타 국가에서 사용되는 The Open Group의 등록상표입니다.
- Intel, Intel 로고, Intel Inside<sup>®</sup>, Intel Inside 로고, Intel<sup>®</sup> Centrino<sup>®</sup>, Intel Centrino 로고, Celeron<sup>®</sup>, Intel<sup>®</sup> Xeon<sup>®</sup>, Intel SpeedStep<sup>®</sup>, Itanium<sup>®</sup> 및 Pentium<sup>®</sup>은 미국 또는 기타 국가에서 사용되는 Intel Corporation의 상표 또는 등록상표입니다.
- Microsoft, Windows, Windows NT<sup>®</sup> 및 Windows 로고는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

기타 회사, 제품 및 서비스 이름은 해당 회사의 상표 또는 서비스표입니다.



# 색인

## [가]

감사 기능 관리자 도구 명령 781  
강조표시 규칙 xii  
갱신  
  DAS 갱신 명령 747  
갱신사항  
  DB2 정보 센터 1327, 1328  
검사 결과 형식화 명령 973  
검색 경로 변경 명령 827  
결함 모니터 제어기 유틸리티 921  
관리 구성  
  네트워크 매개변수 값 693  
  디폴트로 재설정 561  
  샘플 209  
  파일 209  
구문 1273, 1277, 1279, 1281, 1285, 1293, 1297, 1299, 1301, 1305, 1307, 1309  
명령행 처리기 SQL문에 대한 명령문 29  
설명 xiii  
구성  
  관리  
    디폴트로 재설정 561  
    샘플 209  
  데이터베이스  
    갱신 713  
    디폴트로 재설정 565  
    샘플 227  
  데이터베이스 관리 프로그램  
    샘플 233  
  CLI  
    샘플 217  
구성 지원 프로그램(CA)  
  시작 809  
규정자 프리컴파일/바인드 옵션 77, 449  
그룹 그룹 프리컴파일/바인드 옵션 77, 449

## [나]

내재적 연결 3  
널(NULL) 값  
  SQL  
    명령행 처리기 표시 3

노드  
  SOCKS  
    CATALOG TCPIP/TCPIP4/TCPIP6 NODE 명령 119  
노드 디렉토리  
  항목 삭제 687

## [다]

다중페이지 파일 할당 883  
대소문자 구분성  
  명령 3  
  이름 지정 규칙에서 1315  
대화식 CLI 명령 843  
데이터  
  무결성  
    분리 레벨 125  
  분할화  
    테이블 재구성을 사용하여 제거 533  
  재분배  
    REDISTRIBUTE DATABASE PARTITION GROUP 명령  
      507  
데이터 이동  
  데이터베이스 간 이동 283  
  분리문자 제한사항 1317  
데이터베이스  
  간접 디렉토리 항목 335  
  롤 포워드 복구  
    ROLLFORWARD DATABASE 명령 597  
  리모트 디렉토리 항목 335  
  리스트어 573  
  모니터링 569  
백업  
  실행기록 파일 479  
복구  
  ROLLFORWARD DATABASE 명령 597  
삭제  
  데이터베이스 연결 서비스(DCS) 681  
  시스템 데이터베이스 디렉토리 679  
  DROP DATABASE 명령 183  
업그레이딩  
  UPGRADE DATABASE 명령 443  
작성  
  CREATE DATABASE문 129  
재구성 549

데이터베이스 (계속)

재빌드

RESTORE DATABASE 명령 573

재시작 571

정보 콜렉션 263

카탈로그 해제

데이터베이스 연결 서비스(DCS) 681

시스템 데이터베이스 디렉토리 679

카탈로그

CATALOG DATABASE 명령 99

테이블에 데이터 로드 387

테이블에서 파일로 익스포트

EXPORT 명령 193

통계 613

파일에서 테이블로 임포트

IMPORT 명령 283

홈 디렉토리 항목 335

데이터베이스 관리 프로그램

모니터 스위치

GET DATABASE MANAGER MONITOR SWITCHES 명령 239

GET MONITOR SWITCHES 명령 253

시작 653

중지 667

통계 263

데이터베이스 관리 프로그램 구성 파일

샘플 파일 233

GET DATABASE MANAGER CONFIGURATION 명령으로 값 검색 233

데이터베이스 구성 파일

값 검색 227

값 재설정 565

갱신 713

샘플 227

데이터베이스 디렉토리

샘플 콘텐츠 335

설명 335

주석 변경 123

데이터베이스 분석 및 보고 도구 명령

설명 849

데이터베이스 성능 값 재설정 명령 1149

데이터베이스 시스템 모니터

기록 스위치 갱신 731

GET DATABASE MANAGER MONITOR SWITCHES 명령 239

GET MONITOR SWITCHES 명령 253

GET SNAPSHOT 명령 263

RESET MONITOR 명령 569

데이터베이스 시스템 모니터 (계속)

UPDATE MONITOR SWITCHES 명령 731

데이터베이스 업그레이드 점검 명령 835

데이터베이스 연결 서비스(DCS) 디렉토리

항목 제거 681

데이터베이스 이동 도구 명령 1015

데이터베이스 재배치 명령 1157

데이터베이스 파티션 서버 구성 변경 명령 1041

도구 카탈로그 데이터베이스 이주

db2tdbmgr 명령 1229

도움말

메시지 27

명령 25

언어 구성 1326

SQL문 1325

디렉토리

노드

항목 삭제 687

데이터베이스 연결 서비스(DCS)

항목 삭제 681

로컬 데이터베이스

주석 변경 123

시스템 데이터베이스

데이터베이스 카탈로그 해제(API) 679

주석 변경 123

항목 삭제 679

디자인 어드바이저

db2advis 명령 773

디폴트 구성 재설정

데이터베이스 565

DB2 Administration Server(DAS) 561

디폴트 DB2 사본 및 데이터베이스 클라이언트 인터페이스 사본 전환 명령 1215

# [ 라 ]

라이선스

라이선스 관리 도구 명령 991

라인 연속 문자

명령행 처리기(CLP) 3

레벨 프리컴파일 옵션 449

로그

롤 포워드 중 나열 597

로그 시퀀스 번호 찾기 명령 913

로드

데이터베이스 테이블에 파일 387

파일 유형 수정자 387

로드 유틸리티  
 임시 파일  
 LOAD 명령 387  
 로컬 데이터베이스 디렉토리  
 주식 변경 123  
 롤 포워드 보류 상태 재설정 명령 1163  
 롤 포워드 복구에 필요한 로그 나열 명령 997  
 루트 서버가 아닌 서버에 설치된 DB2 명령  
 구성  
 db2nrcfg 1047  
 루트 기능 사용  
 db2rfe 1049  
 인스턴스 갱신  
 db2nrupdt 1051  
 루트 서버가 아닌 인스턴스 업그레이드 명령 1053  
 리스트어  
 DB2 데이터베이스의 이전 버전 573  
 리턴 코드  
 명령행 처리기(CLP) 23  
 릴리스 프리컴파일/바인드 옵션 77, 449

## [ 마 ]

메모리 추적 프로그램 명령 1035  
 메시지  
 도움말 액세스 11  
 프리컴파일/바인드 옵션 77, 449  
 메시지 도움말  
 호출 27  
 명령  
 구문 도움말 37  
 도움말 호출 25  
 설정 1265  
 출력 경로 재지정 3  
 ACTIVATE DATABASE 43  
 ADD CONTACT 45  
 ADD CONTACTGROUP 47  
 ADD DBPARTITIONNUM 49  
 ARCHIVE LOG 55  
 ATTACH 59  
 AUTOCONFIGURE 63  
 BACKUP DATABASE 67  
 BIND 77  
 CATALOG DATABASE 99  
 CATALOG DCS DATABASE 103  
 CATALOG LDAP DATABASE 107  
 CATALOG LDAP NODE 111  
 CATALOG LOCAL NODE 113

명령 (계속)  
 CATALOG NAMED PIPE NODE 115  
 CATALOG ODBC DATA SOURCE 117  
 CATALOG TCP/IP NODE 119  
 CHANGE DATABASE COMMENT 123  
 CHANGE ISOLATION LEVEL 125  
 CREATE DATABASE 129  
 CREATE TOOLS CATALOG 153  
 dasauto 739  
 dasCRT 741  
 dasdrop 743  
 dasmigr 745  
 dasupdt 747  
 db2 11  
 DB2 JDBC 패키지 바인더 유틸리티 985  
 db2acsutil 803  
 db2addicons 759  
 db2admin 761  
 db2adutl 763  
 db2advis 773  
 db2audit 781  
 db2batch 793  
 db2bfd 807  
 db2cap 811  
 db2cc 817  
 db2cfexp 819  
 db2cfimp 823  
 db2chglbpath 825  
 db2chgpath 827  
 db2ckbkp 829  
 db2ckrst 839  
 db2ckupgrade 835  
 db2cli 843  
 db2cmd 845  
 db2cptsa 847  
 db2dart 849  
 db2daslevel 855  
 db2dclgn 857  
 db2diag 861  
 db2drdat 879  
 db2drvmp 881  
 db2empfa 883  
 db2envar.bat 885  
 db2eva 887  
 db2evmon 889  
 db2evtbl 891  
 db2exmig 897  
 db2expln 899

## 명령 (계속)

db2extsec 909  
 db2flsn 913  
 db2fm 917  
 db2fmcu 921  
 db2fode 923  
 db2fs 931  
 db2gcf 933  
 db2gov 937  
 db2govlg 939  
 db2gpmmap 941  
 db2hc 943  
 db2iauto 945  
 db2iclus 947  
 db2icrt 951  
 db2IdentifyType1 961  
 db2idrop 957  
 db2ilist 965  
 db2inidb 971  
 db2inspf 973  
 db2iprune 975  
 db2isetup 977  
 db2iupdt 979  
 db2iupgrade 967  
 db2jdbcbind 985  
 db2ldcfg 987  
 db2level 989  
 db2licm 991  
 db2listvolumes 995  
 db2logsforrwd 997  
 db2look 999  
 db2ls 1011  
 db2move 1015  
 db2mqsln 1027  
 db2mscs 1031  
 db2mtrk 1035  
 db2nchg 1041  
 db2ncrt 1043  
 db2ndrop 1045  
 db2nrcfg 1047  
 db2nrupdt 1051  
 db2nrupgrade 1053  
 db2osconf 1055  
 db2pd 1059  
 db2pdcfg 1141  
 db2perfc 1149  
 db2perfi 1151  
 db2perfr 1153

## 명령 (계속)

db2rbind 1155  
 db2relocatedb 1157  
 db2rfe 1049  
 db2rmicons 1165  
 db2sampl 1169  
 db2schex 1173  
 db2set 1175  
 db2setup 1179  
 db2sqljbind 1181  
 db2sqljprint 1201  
 db2start 1203  
 db2stat 1205  
 db2stop 1207  
 db2support 1209  
 db2swtch 1215  
 db2sync 1217  
 db2systray 1219  
 db2tapemgr 1223  
 db2tbst 1227  
 db2tdbmgr 1229  
 db2trc 1231  
 db2ts ALTER INDEX 1273  
 db2ts CLEANUP FOR TEXT 1277  
 db2ts CLEAR COMMAND LOCKS 1279  
 db2ts CLEAR EVENTS FOR INDEX 1281  
 db2ts CREATE INDEX 1285  
 db2ts DISABLE DATABASE FOR TEXT 1293  
 db2ts DROP INDEX 1297  
 db2ts ENABLE DATABASE FOR TEXT 1299  
 db2ts HELP 1301  
 db2ts START FOR TEXT 1305  
 db2ts STOP FOR TEXT 1307  
 db2ts UPDATE INDEX 1309  
 db2unins 1239  
 db2untag 1241  
 db2updserv 1243  
 db2xdbmig 1247  
 db2xpvt 1249  
 db2\_deinstall 749  
 db2\_install 753  
 db2\_local\_ps 757  
 DEACTIVATE DATABASE 157  
 DECOMPOSE XML DOCUMENT 159, 161  
 DEREGISTER 163  
 DESCRIBE 165  
 DETACH 177  
 disable\_MQFunctions 1251

## 명령 (계속)

doce\_deinstall 1253  
doce\_install 1255  
DROP CONTACT 179  
DROP CONTACTGROUP 181  
DROP DATABASE 183  
DROP DBPARTITIONNUM VERIFY 185  
DROP TOOLS CATALOG 187  
ECHO 189  
EDIT 191  
enable\_MQFunctions 1257  
EXPORT 193  
FORCE APPLICATION 207  
GET ADMIN CONFIGURATION 209  
GET ALERT CONFIGURATION 211  
GET CLI CONFIGURATION 217  
GET CONNECTION STATE 219  
GET CONTACTGROUP 221  
GET CONTACTGROUPS 223  
GET CONTACTS 225  
GET DATABASE CONFIGURATION 227  
GET DATABASE MANAGER CONFIGURATION 233  
GET DATABASE MANAGER MONITOR SWITCHES 239  
GET DESCRIPTION FOR HEALTH INDICATOR 243  
GET HEALTH NOTIFICATION CONTACT LIST 245  
GET HEALTH SNAPSHOT 247  
GET INSTANCE 251  
GET MONITOR SWITCHES 253  
GET RECOMMENDATIONS 257  
GET ROUTINE 261  
GET SNAPSHOT 263  
HELP 279  
HISTORY 281  
IMPORT 283  
INITIALIZE TAPE 315  
INSPECT 317  
installFixPack 1261  
LIST ACTIVE DATABASES 327  
LIST APPLICATIONS 329  
LIST COMMAND OPTIONS 333  
LIST DATABASE DIRECTORY 335  
LIST DATABASE PARTITION GROUPS 339  
LIST DBPARTITIONNUMS 343  
LIST DCS APPLICATIONS 345  
LIST DCS DIRECTORY 347  
LIST DRDA INDOUBT TRANSACTIONS 349  
LIST HISTORY 351  
LIST INDOUBT TRANSACTIONS 355

## 명령 (계속)

LIST NODE DIRECTORY 359  
LIST ODBC DATA SOURCES 363  
LIST PACKAGES/TABLES 365  
LIST TABLESPACE CONTAINERS 369  
LIST TABLESPACES 371  
LIST UTILITIES 385  
LOAD 387  
LOAD QUERY 435  
Microsoft Cluster Server 947  
MQ 리스너 1027  
PING 445  
PRECOMPILE 449  
PRUNE HISTORY/LOGFILE 479  
PUT ROUTINE 481  
QUERY CLIENT 483  
QUIESCE 485  
QUIESCE TABLESPACES FOR TABLE 489  
QUIT 493  
REBIND 495  
RECOVER DATABASE 499  
REDISTRIBUTE DATABASE PARTITION GROUP 507  
REFRESH LDAP 521  
REGISTER 523  
REORG INDEXES/TABLE 533  
REORGCHK 549  
RESET ADMIN CONFIGURATION 561  
RESET ALERT CONFIGURATION 563  
RESET DATABASE CONFIGURATION 565  
RESET DATABASE MANAGER CONFIGURATION 567  
RESET MONITOR 569  
RESTART DATABASE 571  
RESTORE DATABASE 573  
REWIND TAPE 595  
ROLLFORWARD DATABASE 597  
RUNCMD 611  
RUNSTATS 613  
SET CLIENT 631  
SET RUNTIME DEGREE 635  
SET SERVEROUTPUT 637  
SET TABLESPACE CONTAINERS 639  
SET TAPE POSITION 643  
SET UTIL\_IMPACT\_PRIORITY 645  
SET WORKLOAD 649  
SET WRITE 651  
sqlj 1267  
SQLJ 변환기 1267  
SQLJ 프로파일 바인더 1181

## 명령 (계속)

START DATABASE MANAGER 653  
START HADR 663  
STOP DATABASE MANAGER 667  
STOP HADR 671  
TAKEOVER HADR 673  
TERMINATE 677  
UNCATALOG DATABASE 679  
UNCATALOG DCS DATABASE 681  
UNCATALOG LDAP DATABASE 683  
UNCATALOG LDAP NODE 685  
UNCATALOG NODE 687  
UNCATALOG ODBC DATA SOURCE 689  
UNQUIESCE 691  
UPDATE ADMIN CONFIGURATION 693  
UPDATE ALERT CONFIGURATION 695  
UPDATE ALTERNATE SERVER FOR DATABASE 701  
UPDATE ALTERNATE SERVER FOR LDAP  
DATABASE 703  
UPDATE CLI CONFIGURATION 705  
UPDATE COMMAND OPTIONS 707  
UPDATE CONTACT 709  
UPDATE CONTACTGROUP 711  
UPDATE DATABASE CONFIGURATION 713  
UPDATE DATABASE MANAGER CONFIGURATION 719  
UPDATE HEALTH NOTIFICATION CONTACT LIST 723  
UPDATE HISTORY FILE 725  
UPDATE LDAP NODE 729  
UPDATE MONITOR SWITCHES 731  
UPDATE XMLSCHEMA 735  
UPGRADE DATABASE 443

## 명령 구문

CLP 명령 11

## 명령행 처리기(CLP)

도움말 액세스 11

라인 연속 문자 3

리턴 코드 23

## 명령

구문 11

## 백엔드 프로세스

종료 677

사용 3

셸 명령 11

옵션 13

종료 11, 493, 677

호출 명령 11

SQL문 29

## 모니터링

데이터베이스 239, 253

db2pd 명령 1059

모든 디스크 볼륨의 GUID 표시 명령 995

모든 패키지 리바인드 명령 1155

목표 프리컴파일 옵션 449

## 문서

개요 1321

이용약관 1331

인쇄됨 1322

PDF 1322

## 문제점 판별

사용 가능 정보 1331

자습서 1331

진단 도구

db2fodc 923

db2support 1209

문제점 판별 동작을 위한 DB2 데이터베이스 구성 명령 1141

## 문제점 해결

온라인 정보 1331

자습서 1331

db2pd 명령 1059

미러된 데이터베이스 초기화 명령 971

## [ 바 ]

바인드 파일 설명 도구 명령 807

## 바인딩

내재적으로 작성된 스키마 77, 449

오류 129

## 반복 읽기

변경 125

## 백업

오브젝트 803

백업 점검 명령 829

## 버전

프리컴파일 옵션 449

벤치마크 도구 명령 793

## 복구

데이터베이스 573

롤 포워드 사용 597

롤 포워드 없이 573

복합 파일 유형 수정자 283

## 분리 레벨

CHANGE ISOLATION LEVEL 명령 125

분리 프리컴파일/바인드 옵션 77, 449

## 분리문자

데이터 이동 시 제한사항 1317



분리문자 (계속)  
 수정 1317  
 분산 맵 가져오기 명령 941  
 블로킹 프리킵파일/바인드 옵션 77, 449  
 비정상 종료  
 재시작 명령 571

## [사]

삼입 프리킵파일/바인드 옵션 77, 449  
 샘플 데이터베이스 작성 명령 1169  
 서적  
 인쇄됨  
 주문 1324  
 선언 생성기 명령 857  
 설정 명령 1265  
 설치 이미지  
 이미지 크기 줄이기 975  
 설치된 DB2 제품 갱신 명령 1261  
 설치된 DB2 제품 및 기능 나열 명령 1011  
 성능  
 인덱스  
 REORGCHK 명령 549  
 테이블  
 재구성 533  
 REORGCHK 명령 549  
 Windows  
 성능 모니터 등록 도구 명령 1153  
 성능 카운터 등록 유틸리티 명령 1151  
 소유자 프리킵파일/바인드 옵션 77, 449  
 수정자  
 파일 유형  
 EXPORT 명령 193  
 IMPORT 명령 283  
 LOAD 명령 387  
 스냅샷 백업 오브젝트 관리 803  
 스키마  
 새 데이터베이스 129  
 스토리지  
 실제 533  
 시스템 데이터베이스 디렉토리  
 주석 변경 123  
 카탈로그 해제 679  
 항목 삭제 679  
 시스템 카탈로그  
 분석 813

시작  
 DB2  
 db2start 명령 1203  
 실행 파일 1231

## [아]

암호  
 변경  
 ATTACH 명령 59  
 CONNECT문 29  
 액세스 경로  
 최적화 613  
 어드바이저  
 디자인 어드바이저 773  
 db2advis 773  
 언커밋 읽기(UR)  
 변경 125  
 연결 프리킵파일 옵션 449  
 연결성 구성 익스포트 도구 명령 819  
 연결성 구성 임포트 도구 명령 823  
 연속 문자  
 명령행 처리기(CLP) 3  
 오류 메시지  
 데이터베이스 구성 파일 227  
 리모트 데이터베이스 삭제  
 DROP DATABASE 명령 183  
 체크섬  
 데이터베이스 관리 프로그램 구성 파일 561  
 데이터베이스 구성 파일 565, 713  
 온라인 검사  
 사용자 정의 중 필요함 1187  
 제한 1187  
 최적화를 위한 온라인 검사 1187  
 워크로드 관리  
 SET WORKLOAD 명령 649  
 워크스테이션  
 리모트  
 노드 카탈로그 해제 687  
 데이터베이스 카탈로그 해제 679  
 데이터베이스 카탈로그 99  
 유형 1 인덱스 식별 도구 명령 961  
 유효성 확인 프리킵파일/바인드 옵션  
 BIND 명령 77  
 PRECOMPILE 명령 449  
 응답 파일  
 생성 프로그램  
 db2rspgn 1167

- 이 책의 구성 xi
- 이 책의 사용자 xi
- 이름 지정 규칙
  - 데이터베이스 관리 프로그램 오브젝트 1315
- 이미지 975
- 이벤트 모니터 명령 889
- 이벤트 모니터 목표 테이블 정의 생성 명령 891
- 이벤트 분석기 명령 887
- 이용약관
  - 서적 사용 1331
- 익스포트
  - 데이터
    - 파일 유형 수정자 193
    - EXPORT 명령 193
- 인다우트(Indoubt) 트랜잭션 필드 355
- 인덱스
  - 통계
    - RUNSTATS 명령 613
    - REORGCHK 명령 549
- 인스턴스 갱신 명령 979
- 인스턴스 목록 명령 965
- 인스턴스 업그레이드 명령 967
- 인스턴스 자동 시작 명령 945
- 인스턴스 작성 명령 951
- 인스턴스 작성 인터페이스 시작 명령 977
- 인스턴스 제거 명령 957
- 인스턴스에 데이터베이스 파티션 서버 추가 명령 1043
- 인스턴스에서 데이터베이스 파티션 서버 삭제 명령 1045
- 일반 프리컴파일/바인드 옵션 77, 449
- 읽기 안정성
  - 변경 125
- 임베디드(embedded) 런타임 라이브러리 검색 경로
  - 변경 825
- 입시 파일
  - LOAD 명령 387
- 임포트
  - 데이터 283

## [ 자 ]

- 자습서
  - 문제점 판별 1331
  - 문제점 해결 1331
  - Visual Explain 1330
- 잠금
  - 디폴트로 최대값 재설정 565
- 제어 센터
  - 시작 817

- 제어 센터 시작 명령 817
- 제품 갱신사항 표시 명령 1243
- 조치 프리컴파일/바인드 옵션 77, 449
- 존 10진수(zoned DECIMAL) 파일 유형 수정자 387
- 종료
  - 명령행 처리기 백엔드 프로세스 677
  - 비정상 571
  - 일반 667
- 종료 코드
  - CLP 23
- 주의사항 1333
- 중지
  - DB2
    - db2stop 명령 1207
- 중분 리스토어 이미지 순서 확인 명령 839

## [ 차 ]

- 첫 단계 931
- 최적화
  - REORG INDEXES/TABLE 명령 533
- 추적
  - 활성화 1231
  - 추적 명령 1231
  - 추적을 파일에 덤프
    - db2trc 명령 1231
- 출력 프리컴파일 옵션 449

## [ 카 ]

- 카 탈로그
  - 호스트 데이터베이스 103
- 카탈로그 해제
  - 데이터베이스 항목 679
  - 시스템 데이터베이스 디렉토리 679
  - 호스트 DCS 데이터베이스 항목 681
- 카탈로그
  - 데이터베이스 99
- 커널 매개변수 값 유틸리티 명령 1055
- 커밋 없음(NC) 125
- 커서 안정성
  - 변경 125
- 컨테이너 태그 릴리스 명령 1241
- 코드 페이지
  - EXPORT 명령 193
  - IMPORT 명령 283
- 코드 페이지 파일 유형 수정자 387

콜 레벨 인터페이스(CLI)

구성 217

컬렉션 프리컴파일/바인드 옵션 77, 449

## [ 타 ]

테이블

재구성

필요성 판별 549

REORG INDEXES/TABLE 명령 533

통계

설명 613

파일 로드 387

파일 임포트 283

파일로 익스포트 193

테이블 스페이스 상태 확보 명령 1227

테이프 백업 67

테이프의 로그 파일 관리 명령 1223

텍스트 검색 1273, 1279, 1281, 1285, 1293, 1297, 1299, 1301, 1305, 1307, 1309

명령

ALTER INDEX 1273

CLEANUP FOR TEXT 1277

CLEAR COMMAND LOCKS 1279

CLEAR EVENTS FOR INDEX 1281

CREATE INDEX 1285

DISABLE DATABASE FOR TEXT 1293

DROP INDEX 1297

ENABLE DATABASE FOR TEXT 1299

HELP 1301

START FOR TEXT 1305

STOP FOR TEXT 1307

UPDATE INDEX 1309

텍스트 프리컴파일/바인드 옵션 77, 449

통계

데이터베이스 613

데이터베이스 관리 프로그램 263

인덱스 재구성 549

REORGCHK 549

트랩 파일 형식화 명령 1249

트루 타입 글꼴

명령행 처리기에 대한 요구사항 3

특권

데이터베이스

작성 시 권한 부여됨 129

## [ 파 ]

파일 유형 수정자

EXPORT 유틸리티 193

exportimportload 1317

IMPORT 명령 283

LOAD 명령 387

파일 형식

테이블로 파일 임포트 283

파일로 테이블 익스포트 193

패키지

재작성 495

프리컴파일 옵션 449

팬텀 Quiesce 489

페더레이티드 프리컴파일/바인드 옵션 77, 449

## [ 하 ]

현재 DAS 레벨 표시 명령 855

호스트 시스템

데이터베이스 카탈로그 103

DB2 Connect에 의해 지원되는 연결

CATALOG DCS DATABASE 명령 103

DCS 카탈로그 항목 제거 681

호출 27

명령 도움말 25

환경 변수

DB2OPTIONS 13

## A

ACTIVATE DATABASE 명령

구문 설명 43

Active Directory

스키마 확장

명령 1173

ADD CONTACT 명령

구문 설명 45

ADD CONTACTGROUP 명령

구문 설명 47

ADD DBPARTITIONNUM 명령 49

ADD XMLSCHEMA DOCUMENT 명령

구문 53

Administration Server

구성 209

삭제 761

작성 761

anyorder 파일 유형 수정자 387

APPC(Advanced Program-to-Program Communication)

노드

카탈로그 해제 687

ARCHIVE LOG 명령 55

ASC импорт 파일 유형 283

ATTACH 명령 59

AUTOCONFIGURE 명령 63

## B

BACKUP DATABASE 명령 67

binarynumerics 파일 유형 수정자 387

BIND 명령

구문 77

bindfile 프리컴파일 옵션 449

## C

CALL문

CLP를 통한 실행 29

CATALOG DATABASE 명령

구문 99

CATALOG DCS DATABASE 명령 103

CATALOG LDAP DATABASE 명령 107

CATALOG LDAP NODE 명령 111

CATALOG LOCAL NODE 명령 113

CATALOG NAMED PIPE NODE 명령 115

CATALOG ODBC DATA SOURCE 명령 117

CATALOG TCP/IP NODE 명령 119

CCSIDG 프리컴파일/바인드 옵션 77, 449

CCSIDM 프리컴파일/바인드 옵션 77, 449

CCSIDS 프리컴파일/바인드 옵션 77, 449

CHANGE DATABASE COMMENT 명령 123

CHANGE ISOLATION LEVEL 명령 125

chardel 파일 유형 수정자

익스포트 193

임포트 283

load 387

charsub 프리컴파일/바인드 옵션 77, 449

CLIPKG 프리컴파일/바인드 옵션 77

CLI/ODBC 정적 패키지 바인딩 도구 명령 811

CLOSE문

CLP를 통한 실행 29

cnulreqd 프리컴파일/바인드 옵션 77, 449

coldel 파일 유형 수정자

익스포트

EXPORT 명령 193

coldel 파일 유형 수정자 (계속)

임포트

IMPORT 명령 283

load

LOAD 명령 387

COMPLETE XMLSCHEMA 명령

구문 127

CONNECT문

CLP를 통한 실행 29

CREATE DATABASE문

설명 129

CREATE DATABASE문의 RESTRICTIVE 절 129

CREATE TOOLS CATALOG 명령 153

## D

DAS 자동 시작 명령 739

dasauto 명령 739

dasprt 명령 741

DAS(DB2 Administration Server)

관리 761

명령

설명 761

작성 741

삭제 761

작성

db2admin 명령 761

dasdrop 명령 743

dasmigr 명령

설명 745

dasupdt 명령 747

dateformat 파일 유형 수정자

IMPORT 명령 283

LOAD 명령 387

DB2 Administration Server 이주 명령 745

DB2 Administration Server 제거 명령 743

DB2 Administration Server(DAS)

구성 209

DB2 Connect

지원되는 연결 103

DB2 HA 스크립트 설치 또는 갱신 847

DB2 결함 모니터 명령 917

DB2 데이터베이스 드라이브 맵 명령 881

DB2 데이터베이스 제품 설치 제거 명령 1239

DB2 도구 명령의 주메뉴 항목 작성 759

DB2 도구 명령의 주메뉴 항목 제거 1165

DB2 동기 프로그램 시작 명령 1217

db2 명령 11

DB2 명령 창 명령을 엽니다. 845

DB2 사본  
  유효성 확인 도구 1245

DB2 사본 유효성 확인 도구 1245

DB2 서비스 레벨 표시 명령 989

DB2 서적 주문 1324

DB2 설치 명령 975, 1179, 1265

DB2 시스템 트레이 시작 명령 1219

DB2 시작 명령 1203

DB2 오브젝트에 대한 권한 설정 명령 909

DB2 인스턴스 제어 명령 933

DB2 정보 센터  
  갱신 1327, 1328  
  다른 언어로 보기 1326  
  버전 1326  
  언어 1326

DB2 정보 센터 설치 명령 1255

DB2 정보 센터 설치 제거 명령 1253

DB2 제품 설치 명령 753

DB2 제품, 기능 또는 언어 설치 제거 명령 749

DB2 조정자  
  로그 쿼리 명령 939  
  명령 937

DB2 중지 명령 1207

DB2 통계 및 DDL 추출 도구 명령 999

DB2 프로파일 레지스트리 명령 1175

db2acsutil 명령 803

db2addicons 명령  
  설명 759

db2admin 명령 761

db2adutl 명령  
  설명 763

db2advis 명령  
  설명 773

db2audit 명령  
  설명 781

db2batch 명령  
  설명 793

db2bfd 명령  
  설명 807

db2ca 명령 809

db2cap 명령 811

db2cat 명령 813

db2cc 명령 817

db2cfexp 명령 819

db2cfimp 명령 823

db2chglbpath 명령 825

db2chgpath 명령 827

db2ckbcp 명령 829

db2ckrst 명령 839

db2ckupgrade 명령  
  설명 835

db2cli 명령 843

db2cmd 명령 845

db2cptsa 명령 847

db2dart 명령  
  설명 849

db2daslevel 명령 855

db2dc1gn 명령  
  설명 857

db2diag 로그 분석 도구 명령 861

db2diag 명령  
  설명 861

db2drdat 명령  
  설명 879

db2drvmp 명령 881

db2empfa 명령 883

db2envar.bat 명령 885

db2eva 명령 887

db2evmon 명령 889

db2evtbl 명령 891

db2exfmt 명령  
  설명 893

db2exmig 명령  
  설명 897

db2expln 명령  
  설명 899

db2extsec 명령 909

db2flsn 명령 913

db2fm 명령  
  설명 917

db2fmcu 명령 921

db2fodc 명령 923

db2fs 명령 931

db2gcf 명령 933

db2gov 명령  
  설명 937

db2govlg 명령 939

db2gpmap 명령 941

db2hc 명령 943

db2iauto 명령 945

db2iclus 명령 947

db2icrt 명령  
  설명 951

db2IdentifyType1 명령 961

db2idrop 명령  
 설명 957

db2ilist 명령  
 설명 965

db2inidb 명령  
 설명 971

db2inspf 명령  
 설명 973

db2iprune 명령 975

db2isetup 명령 977

db2iupdt 명령  
 설명 979

db2iupgrade 명령  
 설명 967

db2jdbcbind 명령 985

db2ldcfg 명령  
 설명 987

db2level 명령  
 설명 989

db2licm 명령  
 설명 991

db2listvolumes 명령 995

db2logsforrwd 명령 997

db2look 명령  
 설명 999

db2ls 명령  
 설명 1011

db2move 명령  
 설명 1015

db2mqslsn 명령 1027

db2mscs 명령 1031

db2mtrk 명령  
 설명 1035

db2nchg 명령  
 설명 1041

db2ncrt 명령  
 설명 1043

db2ndrop 명령  
 설명 1045

db2nrcfg 명령 1047

db2nrupdt 명령 1051

db2nrupgrade 명령  
 설명 1053

DB2OPTIONS 환경 변수  
 CLP 옵션 설정 13

db2osconf 명령  
 설명 1055

db2pd 명령  
 설명 1059

db2pdcfg 명령  
 설명 1141

db2perfc 명령  
 설명 1149

db2perfi 명령  
 설명 1151

db2perfr 명령  
 설명 1153

db2rbind 명령  
 설명 1155

db2relocatedb 명령  
 설명 1157

db2rfe 명령  
 설명 1049

db2rfpen 명령 1163

db2rmicons 명령  
 설명 1165

db2rspgn 응답 파일 생성 프로그램 1167

db2saml 명령 1169

db2schex 명령 1173

db2set 명령  
 설명 1175

db2setup 명령  
 설명 1179

db2sqljbind 명령 1181

db2sqljcustomize 명령 1187

db2sqljprint 명령  
 description 1201

db2start 명령  
 설명 653, 1203

db2stat 명령 1205

db2stop 명령  
 설명 667, 1207

db2swtch 명령 1215

db2sync 명령 1217

db2systray 명령 1219

db2tapemgr 명령  
 설명 1223

db2tbst 명령 1227

db2tdbmgr 명령  
 설명 1229

db2trc 명령  
 설명 1231

db2ts 명령  
 ALTER INDEX 1273  
 CLEANUP FOR TEXT 1277

db2ts 명령 (계속)

- CLEAR COMMAND LOCKS 1279
- CLEAR EVENTS FOR INDEX 1281
- CREATE INDEX 1285
- DISABLE DATABASE FOR TEXT 1293
- DROP INDEX 1297
- ENABLE DATABASE FOR TEXT 1299
- HELP 1301
- START FOR TEXT 1305
- STOP FOR TEXT 1307
- UPDATE INDEX 1309

db2unins 명령 1239

db2untag 명령 1241

db2updserv 명령 1243

db2val 1245

db2xdbmig 명령 1247

db2xpirt 명령 1249

db2\_deinstall 명령

- 설명 749

db2\_install 명령

- 설명 753

db2\_local\_ps 명령 757

DCLGEN 명령 857

DEACTIVATE DATABASE 명령 157

DECLARE CURSOR문

- CLP에서 실행 29

DECOMPOSE XML DOCUMENT 명령

- 설명 159, 161

decplusblank 파일 유형 수정자

- EXPORT 명령 193
- IMPORT 명령 283
- LOAD 명령 387

decppt 파일 유형 수정자

- EXPORT 명령 193
- IMPORT 명령 283
- LOAD 명령 387

delprioritychar 파일 유형 수정자

- IMPORT 명령 283
- LOAD 명령 387

DEREGISTER 명령 163

DESCRIBE 명령

- 설명
- ADMIN\_CMD 프로시저 없는 명령 165

DETACH 명령

- 설명 177

disable\_MQFunctions 명령 1251

doce\_deinstall 명령

- 설명 1253

doce\_install 명령

- 설명 1255

DRDA 추적 명령 879

DROP CONTACT 명령

- 설명 179

DROP CONTACTGROUP 명령

- 설명 181

DROP DATABASE 명령

- 설명 183

DROP DBPARTITIONNUM VERIFY 명령 185

DROP TOOLS CATALOG 명령 187

dumpfile 파일 유형 수정자 387

DYNAMICRULES 프리컴파일/바인드 옵션

- BIND 명령 77
- PRECOMPILE 명령 449

## E

ECHO 명령 189

EDIT 명령 191

enable\_MQFunctions 명령 1257

Explain 테이블

- 컨텐츠 형식화 893

Explain 테이블 이주 명령 897

EXPORT 명령

- 설명
- ADMIN\_CMD 프로시저 없는 명령 193

## F

fastparse 파일 유형 수정자 387

federated\_asynchrony 프리컴파일/바인드 옵션 77, 449

FETCH문

- CLP를 통한 실행 29

FODC(first occurrence data capture)

- db2pdcfg 명령의 옵션 1141

FODC(First Occurrence Data Collection) 923

FORCE APPLICATION 명령 207

forcein 파일 유형 수정자 283, 387

funcpath 프리컴파일/바인드 옵션 77, 449

## G

generatedignore 파일 유형 수정자 283, 387

generatedmissing 파일 유형 수정자 283, 387

generatedoverride 파일 유형 수정자 387

GET ADMIN CONFIGURATION 명령 209

GET ALERT CONFIGURATION 명령 211

GET CLI CONFIGURATION 명령 217  
 GET CONNECTION STATE 명령 219  
 GET CONTACTGROUP 명령 221  
 GET CONTACTGROUPS 명령 223  
 GET CONTACTS 명령 225  
 GET DATABASE CONFIGURATION 명령 227  
 GET DATABASE MANAGER CONFIGURATION 명령 233  
 GET DATABASE MANAGER MONITOR SWITCHES 명령  
 239  
 GET DESCRIPTION FOR HEALTH INDICATOR 명령 243  
 GET HEALTH NOTIFICATION CONTACT LIST 명령 245  
 GET HEALTH SNAPSHOT 명령 247  
 GET INSTANCE 명령 251  
 GET MONITOR SWITCHES 명령 253  
 GET RECOMMENDATIONS  
 명령 257  
 GET ROUTINE 명령 261  
 GET SNAPSHOT 명령 263  
 UPDATE MONITOR SWITCHES에 대한 효과 731  
 grant 바인드 옵션 77  
 grantgroup 바인드 옵션 77  
 grantuser 바인드 옵션 77

## H

Health Center 시작 명령 943  
 HELP 명령  
 Linux, UNIX 및 Windows용 DB2 데이터베이스 279  
 HISTORY 명령 281

## I

identityignore 283  
 파일 유형 수정자 387  
 identitymissing  
 파일 유형 수정자 283, 387  
 identityoverride  
 파일 유형 수정자 387  
 implieddecimal 파일 유형 수정자 283, 387  
 IMPORT 명령 283  
 indexfreespace 파일 유형 수정자 387  
 indexixf 파일 유형 수정자 283  
 indexschema 파일 유형 수정자 283  
 INITIALIZE TAPE 명령 315  
 INSPECT 명령 317  
 installFixPack 명령 1261  
 IPX/SPX 노드  
 카탈로그 해제 687

## J

JDBC(Java Database Connectivity)  
 패키지 바인더 유틸리티 명령 985

## K

keepblanks 파일 유형 수정자  
 로드  
 LOAD 명령 387  
 IMPORT 명령 283

## L

LANGLEVEL 프리컴파일 옵션  
 SQL92E 449  
 LDAP 환경 구성 명령 987  
 Linux/UNIX의 DB2 프로세스 상태 명령 757  
 LIST ACTIVE DATABASES 명령 327  
 LIST APPLICATIONS 명령 329  
 LIST COMMAND OPTIONS 명령 333  
 LIST DATABASE DIRECTORY 명령 335  
 LIST DATABASE PARTITION GROUPS 명령 339  
 LIST DBPARTITIONNUMS 명령 343  
 LIST DCS APPLICATIONS 명령 345  
 LIST DCS DIRECTORY 명령 347  
 LIST DRDA INDOUBT TRANSACTIONS 명령 349  
 LIST HISTORY 명령 351  
 LIST INDOUBT TRANSACTIONS 명령 355  
 LIST NODE DIRECTORY 명령 359  
 LIST ODBC DATA SOURCES 명령 363  
 LIST PACKAGES 명령 365  
 LIST PACKAGES/TABLES 명령 365  
 LIST TABLES 명령 365  
 LIST TABLESPACE CONTAINERS 명령 369  
 LIST TABLESPACES 명령 371  
 LIST UTILITIES 명령 385  
 LOAD QUERY 명령 435  
 LOAD 명령  
 개요 387  
 lobsinfile 파일 유형 수정자  
 로드 387  
 익스포트 193  
 임포트 283  
 longerror 프리컴파일 옵션 449



## M

Microsoft Cluster Server 명령 947  
MQ 리스너 명령 1027

## N

NetBIOS

노드

    카탈로그 해제 687

nochecklengths 파일 유형 수정자

    로드 387

    임포트 283

nodefaults 파일 유형 수정자

    임포트 283

nodoubledel 파일 유형 수정자

    로드 387

    익스포트 193

    임포트 283

noeofchar 파일 유형 수정자

    로드 387

    임포트 283

noheader 파일 유형 수정자

    로드 387

NOLINEMACRO 프리컴파일 옵션 449

norowwarnings 파일 유형 수정자

    LOAD 명령 387

notypeid 파일 유형 수정자

    IMPORT 명령 283

NULL 문자열 3

nullindchar 파일 유형 수정자

    IMPORT 명령 387

    LOAD 명령 283

## O

OPEN문

    CLP를 통한 실행 29

optlevel 프리컴파일 옵션 449

## P

packeddecimal 파일 유형 수정자 387

pagefreespace 파일 유형 수정자 387

PING 명령

    설명 445

PRECOMPILE 명령

    설명 449

PREP 명령 449

PRUNE HISTORY/LOGFILE 명령

    ADMIN\_CMD 프로시저 없는 명령 479

PUT ROUTINE 명령 481

## Q

QUERY CLIENT 명령 483

queryopt 프리컴파일/바인드 옵션

    BIND 명령 77

    PRECOMPILE 명령 449

Quiesce

    팬텀 489

QUIESCE TABLESPACES FOR TABLE 명령 489

QUIESCE 명령 485

QUIT 명령 493

## R

REBIND 명령 495

reclen 파일 유형 수정자 283

    로드 387

RECOVER DATABASE 명령 499

REDISTRIBUTE DATABASE PARTITION GROUP 명령 507

REFRESH LDAP 명령 521

REGISTER XMLSCHEMA 명령

    구문 527

REGISTER XSROBJECT 명령

    구문 531

REGISTER 명령 523

REORG TABLE 명령 533

REORGCHK 명령 549

RESET ADMIN CONFIGURATION 명령 561

RESET ALERT CONFIGURATION 명령 563

RESET DATABASE CONFIGURATION 명령 565

RESET DATABASE MANAGER CONFIGURATION 명령 567

RESET MONITOR 명령 569

RESTART DATABASE 명령 571

RESTORE DATABASE 명령 573

REWIND TAPE 명령 595

ROLLFORWARD DATABASE 명령 597

RUNCMD 명령 611

RUNSTATS 명령

    구문 613

## S

### SELECT문

CLP를 통한 실행 29

EXPORT 명령에서 193

SET CLIENT 명령 631

SET RUNTIME DEGREE 명령 635

SET SERVEROUTPUT 명령 637

SET TABLESPACE CONTAINERS 명령 639

SET TAPE POSITION 명령 643

SET UTIL\_IMPACT\_PRIORITY 명령 645

SET WORKLOAD 명령 649

SET WRITE 명령 651

### SIGNALRM 신호

데이터베이스 관리 프로그램 시작 653

### SIGINT 신호

데이터베이스 관리 프로그램 시작 653

### SOCKS

node

매개변수 119

SQL 및 XQuery Explain 명령 899

sqlca 프리컴파일 옵션 449

sqlerror 프리컴파일/바인드 옵션 77, 449

sqlflag 프리컴파일 옵션 449

sqlj 명령 1267

SQLJ 변환기 명령 1267

SQLJ 프로파일 바인더 명령 1181

SQLJ 프로파일 프린터 명령 1201

sqlrules 프리컴파일 옵션 449

sqlwarn 프리컴파일/바인드 옵션 77, 449

### SQL문

도움말 액세스 11

도움말 표시 1325

명령행 사용 29

START DATABASE MANAGER 명령 653

START HADR 명령 663

STOP DATABASE MANAGER 명령 667

STOP HADR 명령 671

strdel 프리컴파일/바인드 옵션 77, 449

striptblanks 파일 유형 수정자 283, 387

striptnulls 파일 유형 수정자 283, 387

subtableconvert 파일 유형 수정자 387

syncpoint 프리컴파일 옵션 449

## T

TAKEOVER HADR 명령 673

## TCP/IP

노드

카탈로그 해제 687

TERMINATE 명령 677

timeformat 파일 유형 수정자 283, 387

timestampformat 파일 유형 수정자

IMPORT 명령 283

LOAD 명령 387

totalfreespace 파일 유형 수정자 387

## TSM

아카이브된 이미지 763

## U

UNCATALOG DATABASE 명령 679

UNCATALOG DCS DATABASE 명령 681

UNCATALOG LDAP DATABASE 명령 683

UNCATALOG LDAP NODE 명령 685

UNCATALOG NODE 명령 687

UNCATALOG ODBC DATA SOURCE 명령 689

UNQUIESCE 명령 691

UPDATE ADMIN CONFIGURATION 명령 693

UPDATE ALERT CONFIGURATION 명령 695

UPDATE ALTERNATE SERVER FOR DATABASE 명령 701

UPDATE ALTERNATE SERVER FOR LDAP DATABASE 명령  
703

UPDATE CLI CONFIGURATION 명령 705

UPDATE COMMAND OPTIONS 명령 707

UPDATE CONTACT 명령 709

UPDATE CONTACTGROUP 명령 711

UPDATE DATABASE CONFIGURATION 명령 713

UPDATE DATABASE MANAGER CONFIGURATION 명령  
719

UPDATE HEALTH NOTIFICATION CONTACT LIST 명령 723

UPDATE HISTORY FILE 명령 725

UPDATE LDAP NODE 명령 729

UPDATE MONITOR SWITCHES 명령 731

UPDATE XMLSCHEMA 명령

구문 735

UPGRADE DATABASE 명령 443

usedefaults 파일 유형 수정자 283, 387

## V

Visual Explain

자습서 1330

## W

WCHARTYPE 프리킵과일터 옵션

설명 449

Windows 장애 복구 유틸리티 설정 명령 1031

Windows의 DB2 프로세스 상태 명령 1205

## X

XBSA(Backup Services APIs) 옵션

BACKUP DATABASE 명령 67

XBSA(X/Open Backup Services API)

인터페이스 67

XML 스키마

저장소

ADD XMLSCHEMA DOCUMENT 명령 53

COMPLETE XMLSCHEMA 명령 127

REGISTER XMLSCHEMA 명령 527

REGISTER XSROBJECT 명령 531

UPDATE XMLSCHEMA 명령 735

XSR 오브젝트 이주 명령 1247

## Z

z/OS용 DB2

바인딩 플랜 및 패키지 1187







SA30-3959-00



Spine information:

Linux, UNIX 및 Windows용 IBM DB2 9.7

명령어 참조서

