

**IBM DB2 10.1**  
**for Linux, UNIX, and Windows**

**コマンド・リファレンス**  
**2013 年 1 月更新版**

**IBM**



**IBM DB2 10.1**  
**for Linux, UNIX, and Windows**

**コマンド・リファレンス**  
**2013 年 1 月更新版**



**ご注意**

本書および本書で紹介する製品をご使用になる前に、1445 ページの『付録 D. 特記事項』に記載されている情報をお読みください。

本書には、IBM の専有情報が含まれています。その情報は、使用許諾条件に基づき提供され、著作権により保護されています。本書に記載される情報には、いかなる製品の保証も含まれていません。また、本書で提供されるいかなる記述も、製品保証として解釈すべきではありません。

IBM 資料は、オンラインでご注文いただくことも、ご自分の国または地域の IBM 担当員を通してお求めいただくこともできます。

- オンラインで資料を注文するには、IBM Publications Center (<http://www.ibm.com/shop/publications/order>) をご利用ください。
- ご自分の国または地域の IBM 担当員を見つけるには、IBM Directory of Worldwide Contacts (<http://www.ibm.com/planetwide/>) をお調べください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： SC27-3868-01  
IBM DB2 10.1  
for Linux, UNIX, and Windows  
Command Reference  
Updated January, 2013

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第1刷 2012.12

© Copyright IBM Corporation 1993, 2013.

# 目次

本書について . . . . .	ix
本書の対象読者 . . . . .	ix
本書の構成 . . . . .	ix
強調表記規則 . . . . .	x

構文図の見方 . . . . .	xi
------------------	----

## 第 1 章 コマンド行プロセッサ (CLP) . . . 1

コマンド行プロセッサのフィーチャー . . . . .	2
db2 - コマンド行プロセッサの呼び出し . . . . .	9
コマンド行プロセッサのオプション . . . . .	11
コマンド行プロセッサの戻りコード . . . . .	21
コマンド行プロセッサからのコマンド・ヘルプの 呼び出し . . . . .	22
コマンド行プロセッサからのメッセージ・ヘルプ の呼び出し . . . . .	22

## 第 2 章 コマンド行 SQL ステートメン トおよび XQuery ステートメントの使用 . 23

## 第 3 章 Command Line Processor Plus (CLPPlus) . . . . . 31

既存の IBM Data Server Client における CLPPlus のインストール . . . . .	31
CLPPlus の開始 . . . . .	32
CLPPLUS コマンド . . . . .	34
CLPPlus コンソール・タイプ . . . . .	36
ウィンドウ・モードと UTF-8 文字サポート . . . . .	38
CLPPlus ウィンドウ・モードのフォントの設定 . . . . .	38
CLPPlus ウィンドウ・モードのフォントの色の設 定 . . . . .	39
CLPPlus ウィンドウ・モードの背景色の設定 . . . . .	39
CLPPlus での DSN 別名 . . . . .	40
CLPPlus での Kerberos 認証 . . . . .	42
CLPPlus での SERVER_ENCRYPT 認証 . . . . .	43
CLPPlus での LDAP サポート . . . . .	45
CLPPlus でのスクリプト・ファイルの実行 . . . . .	48
CLPPlus コマンドのスキップと割り込み . . . . .	50
CLPPlus でのコメント . . . . .	51
CLPPlus のエスケープ文字 . . . . .	52
CLPPlus のバインド変数 . . . . .	54
CLPPlus のシェルおよび環境変数 . . . . .	58
CLPPlus でサポートされる DB2 コマンド . . . . .	59
CLPPlus での CREATE DATABASE . . . . .	61
CLPPlus の制約事項 . . . . .	62
CLPPlus トラブルシューティングのヒント . . . . .	63
CLPPlus のトレースおよびレコードのロギング . . . . .	64

## 第 4 章 コマンド構文ヘルプの読み方 . . 67

## 第 5 章 CLP コマンド . . . . . 71

ACTIVATE DATABASE . . . . .	72
ADD CONTACT . . . . .	74
ADD CONTACTGROUP . . . . .	75
ADD DBPARTITIONNUM . . . . .	76
ADD XMLSCHEMA DOCUMENT . . . . .	79
ARCHIVE LOG . . . . .	81
ATTACH . . . . .	84
AUTOCONFIGURE . . . . .	86
BACKUP DATABASE . . . . .	89
BIND . . . . .	100
CATALOG DATABASE . . . . .	122
CATALOG DCS DATABASE . . . . .	126
CATALOG LDAP DATABASE . . . . .	129
CATALOG LDAP NODE . . . . .	132
CATALOG LOCAL NODE . . . . .	133
CATALOG NAMED PIPE NODE . . . . .	135
CATALOG ODBC DATA SOURCE . . . . .	137
CATALOG TCPIP/TCPIP4/TCPIP6 NODE . . . . .	139
CHANGE DATABASE COMMENT . . . . .	143
CHANGE ISOLATION LEVEL . . . . .	145
COMPLETE XMLSCHEMA . . . . .	147
CREATE DATABASE . . . . .	148
CREATE TOOLS CATALOG . . . . .	168
DEACTIVATE DATABASE . . . . .	171
DECOMPOSE XML DOCUMENT . . . . .	173
DECOMPOSE XML DOCUMENTS . . . . .	175
DEREGISTER . . . . .	179
DESCRIBE . . . . .	180
DETACH . . . . .	190
DROP CONTACT . . . . .	191
DROP CONTACTGROUP . . . . .	192
DROP DATABASE . . . . .	193
DROP DBPARTITIONNUM VERIFY . . . . .	195
DROP TOOLS CATALOG . . . . .	196
ECHO . . . . .	198
EDIT . . . . .	199
EXPORT . . . . .	200
FORCE APPLICATION . . . . .	213
GET ADMIN CONFIGURATION . . . . .	215
GET ALERT CONFIGURATION . . . . .	217
GET CLI CONFIGURATION . . . . .	223
GET CONNECTION STATE . . . . .	225
GET CONTACTGROUP . . . . .	226
GET CONTACTGROUPS . . . . .	227
GET CONTACTS . . . . .	228
GET DATABASE CONFIGURATION . . . . .	229
GET DATABASE MANAGER CONFIGURATION . . . . .	242
GET DATABASE MANAGER MONITOR SWITCHES . . . . .	252
GET DESCRIPTION FOR HEALTH INDICATOR . . . . .	255

GET HEALTH NOTIFICATION CONTACT LIST	256	RESTART DATABASE	605
GET HEALTH SNAPSHOT	257	RESTORE DATABASE	608
GET INSTANCE	261	REWIND TAPE	633
GET MONITOR SWITCHES	262	ROLLFORWARD DATABASE	634
GET RECOMMENDATIONS FOR HEALTH INDICATOR	265	RUNCMD	647
GET ROUTINE	269	RUNSTATS	648
GET SNAPSHOT	271	SET CLIENT	665
HELP	288	SET RUNTIME DEGREE	668
HISTORY	289	SET SERVEROUTPUT	670
IMPORT	290	SET TABLESPACE CONTAINERS	672
INGEST	323	SET TAPE POSITION	675
INITIALIZE TAPE	373	SET UTIL_IMPACT_PRIORITY	676
INSPECT	374	SET WORKLOAD	679
LIST ACTIVE DATABASES	382	SET WRITE	680
LIST APPLICATIONS	384	START DATABASE MANAGER	683
LIST COMMAND OPTIONS	387	START HADR	693
LIST DATABASE DIRECTORY	388	STOP DATABASE MANAGER	696
LIST DATABASE PARTITION GROUPS	392	STOP HADR	701
LIST DBPARTITIONNUMS	394	TAKEOVER HADR	703
LIST DCS APPLICATIONS	395	TERMINATE	708
LIST DCS DIRECTORY	397	UNCATALOG DATABASE	709
LIST DRDA INDOUBT TRANSACTIONS	399	UNCATALOG DCS DATABASE	711
LIST HISTORY	401	UNCATALOG LDAP DATABASE	712
LIST INDOUBT TRANSACTIONS	406	UNCATALOG LDAP NODE	713
LIST NODE DIRECTORY	409	UNCATALOG NODE	714
LIST ODBC DATA SOURCES	412	UNCATALOG ODBC DATA SOURCE	715
LIST PACKAGES/TABLES	413	UNQUIESCE	716
LIST TABLESPACE CONTAINERS	416	UPDATE ADMIN CONFIGURATION	718
LIST TABLESPACES	418	UPDATE ALERT CONFIGURATION	720
LIST UTILITIES	423	UPDATE ALTERNATE SERVER FOR DATABASE	725
LOAD	425	UPDATE ALTERNATE SERVER FOR LDAP DATABASE	727
LOAD QUERY	478	UPDATE CLI CONFIGURATION	728
MIGRATE DATABASE	484	UPDATE COMMAND OPTIONS	730
PING	485	UPDATE CONTACT	732
PRECOMPILE	487	UPDATE CONTACTGROUP	733
PRUNE HISTORY/LOGFILE	517	UPDATE DATABASE CONFIGURATION	734
PUT ROUTINE	520	UPDATE DATABASE MANAGER CONFIGURATION	738
QUERY CLIENT	522	UPDATE HEALTH NOTIFICATION CONTACT LIST	740
QUIESCE	523	UPDATE HISTORY	741
QUIESCE TABLESPACES FOR TABLE	527	UPDATE LDAP NODE	744
QUIT	530	UPDATE MONITOR SWITCHES	746
REBIND	531	UPDATE XMLSCHEMA	749
RECOVER DATABASE	536	UPGRADE DATABASE	751
REDISTRIBUTE DATABASE PARTITION GROUP	544		
REFRESH LDAP	553		
REGISTER	555		
REGISTER XMLSCHEMA	558		
REGISTER XSROBJECT	560		
REORG INDEXES/TABLE	562		
REORGCHK	582		
RESET ADMIN CONFIGURATION	595		
RESET ALERT CONFIGURATION	597		
RESET DATABASE CONFIGURATION	599		
RESET DATABASE MANAGER CONFIGURATION	601		
RESET MONITOR	603		
		<b>第 6 章 CLPPlus コマンド</b>	<b>753</b>
		..	754
		!	755
		/.	756
		@	757
		@@	758
		ACCEPT	759
		APPEND	762
		BREAK	763

BTITLE . . . . .	764	db2adutl - TSM 内の DB2 オブジェクトの管理 . . . . .	867
CHANGE . . . . .	766	db2advis - DB2 設計アドバイザー . . . . .	877
CLEAR . . . . .	768	db2audit - 監査機能管理者用ツール . . . . .	884
COLUMN . . . . .	770	db2batch - ベンチマーク・ツール . . . . .	895
COMPUTE . . . . .	775	db2bfd - バインド・ファイル記述ツール . . . . .	906
CONNECT . . . . .	777	db2caem - アクティビティ・イベント・モニター・データ・ツールのキャプチャー . . . . .	907
COPY . . . . .	780	db2cap - CLI/ODBC 静的パッケージ・バインディング・ツール . . . . .	911
DEFINE . . . . .	783	db2cat - システム・カタログ分析 . . . . .	913
DEL . . . . .	784	db2cfexp - 接続構成エクスポート・ツール . . . . .	916
DESCRIBE . . . . .	786	db2cfimp - 接続構成インポート・ツール . . . . .	918
DISCONNECT . . . . .	791	db2chglbpath - 組み込みランタイム・ライブラリー検索パスの変更 . . . . .	919
EDIT . . . . .	792	db2chgpath - 組み込みランタイム・パスの変更 . . . . .	922
EXECUTE . . . . .	793	db2ckbkp - バックアップの検査 . . . . .	924
EXIT . . . . .	796	db2cklog - アーカイブ・ログ・ファイルの妥当性の検査 . . . . .	929
EXPLAIN PLAN . . . . .	798	db2ckrst - 増分リストア・イメージ順序の検査 . . . . .	932
GET . . . . .	799	db2ckupgrade - アップグレードのためのデータベースの検査 . . . . .	934
HELP . . . . .	800	db2cli - DB2 対話機能 CLI . . . . .	937
HOST . . . . .	801	db2cmd - DB2 コマンド・ウィンドウのオープン . . . . .	954
IMPORT CLPPlus コマンド . . . . .	802	db2cptsa - DB2 HA スクリプトのインストールまたは更新 . . . . .	955
INPUT . . . . .	804	db2dart - データベース分析およびレポート・ツール . . . . .	957
LIST . . . . .	805	db2daslevel - DAS レベルの表示 . . . . .	966
PAUSE . . . . .	807	db2dclgn - 宣言生成プログラム . . . . .	967
PRINT . . . . .	808	db2diag - db2diag ログ分析ツール . . . . .	970
PROMPT . . . . .	809	db2drdat - DRDA トレース . . . . .	989
QUIT . . . . .	810	db2drvmp - DB2 データベース・ドライブのマップ . . . . .	991
REMARK . . . . .	811	db2empfa - 複数ページ・ファイル割り振りの有効化 . . . . .	993
REORGCHK . . . . .	812	db2envar.bat - 現行コマンド・ウィンドウの環境の設定 . . . . .	994
REPFOOTER . . . . .	814	db2evmon - イベント・モニター生産性向上ツール . . . . .	995
REPHEADER . . . . .	816	db2evtbl - イベント・モニターのターゲット表定義の生成 . . . . .	997
RUN . . . . .	818	db2exfmt - Explain 表フォーマット . . . . .	999
SAVE . . . . .	819	db2exmig - Explain 表のマイグレーション . . . . .	1003
SET . . . . .	820	db2expln - SQL および XQuery Explain . . . . .	1004
SPOOL . . . . .	830	db2extsec - DB2 オブジェクトの権限の設定 . . . . .	1014
SHOW . . . . .	831	db2flsn - ログ・シーケンス番号の検出 . . . . .	1017
START . . . . .	832	db2fm - DB2 障害モニター . . . . .	1022
TTITLE . . . . .	833	db2fmcu - DB2 障害モニター・コントローラー . . . . .	1025
UNDEFINE . . . . .	835	db2fodc - DB2 First Occurrence Data Collection . . . . .	1027
WHENEVER OSERROR . . . . .	836	db2fopt - 照会オプティマイザー・パラメーターの指定 . . . . .	1039
WHENEVER SQLERROR . . . . .	838	db2fs - ファースト・ステップ . . . . .	1042
<b>第 7 章 システム・コマンド . . . . . 841</b>			
dasauto - DB2 Administration Server 自動始動 . . . . .	842	db2gcf - DB2 インスタンスの制御 . . . . .	1043
dasprt - DB2 Administration Server の作成 . . . . .	843	db2gov - DB2 管理プログラム . . . . .	1046
dasdrop - DB2 Administration Server の除去 . . . . .	845	db2govlg - DB2 管理プログラム・ログ照会 . . . . .	1049
dasmigr - DB2 Administration Server のマイグレーション . . . . .	846	db2gpmap - 分散マップの取得 . . . . .	1050
dasuptd - DAS の更新 . . . . .	848	db2iauto - インスタンスの自動開始 . . . . .	1051
db2_deinstall - DB2 データベース製品、フィーチャー、または言語のアンインストール . . . . .	850	db2iclus - Microsoft Cluster Server . . . . .	1052
db2_install - DB2 データベース製品のインストール . . . . .	854	db2icrt - インスタンスの作成 . . . . .	1055
db2_local_ps - Linux/UNIX の DB2 プロセス状況 . . . . .	858		
db2acsutil - DB2 スナップショット・バックアップ・オブジェクトの管理 . . . . .	860		
db2addicons - DB2 ツールのメインメニュー項目の作成 . . . . .	864		
db2admin - DB2 Administration Server . . . . .	865		



db2idrop - インスタンスの除去 . . . . .	1065	db2rfpen - ロールフォワード・ペンディング状態 にリセット . . . . .	1286
db2IdentifyType1 - タイプ 1 索引識別ツール . . .	1068	db2rmicons - メインメニューからの DB2 ツール の除去 . . . . .	1287
db2iilist - インスタンスのリスト . . . . .	1071	db2rspgn - 応答ファイル生成プログラム . . . . .	1288
db2inidb - ミラーリングされたデータベースの初 期化 . . . . .	1072	db2sampl - サンプル・データベースの作成 . . . . .	1289
db2inspf - 検査結果のフォーマット . . . . .	1075	db2schex - Active Directory スキーマ拡張 . . . . .	1292
db2iprune - インストール・イメージ・サイズの削 減 . . . . .	1076	db2set - DB2 プロファイル・レジストリー . . . . .	1294
db2isetup - インスタンス作成インターフェースの 開始 . . . . .	1078	db2setup - DB2 データベース製品のインストール db2snapcore - Linux および Solaris 用 DB2 snapcore . . . . .	1299 1301
db2iupdt - インスタンスの更新 . . . . .	1080	db2start - Start DB2 . . . . .	1303
db2iupgrade - インスタンスのアップグレード・ プログラム . . . . .	1090	db2stat - Windows の DB2 プロセス状況 . . . . .	1304
db2jdbcbind - DB2 JDBC パッケージ・バインド・ プログラム . . . . .	1094	db2stop - DB2 の停止 . . . . .	1305
db2ldcfg - LDAP 環境の構成 . . . . .	1096	db2support - 問題分析および環境収集ツール . . . . .	1306
db2level - DB2 サービス・レベルの表示 . . . . .	1098	db2swtch - デフォルト DB2 コピーとデータベー ス・クライアント・インターフェース・コピーの 切り替え . . . . .	1326
db2licm - ライセンス管理ツール . . . . .	1099	db2sync - DB2 シンクロナイザーの開始 . . . . .	1328
db2listvolumes - すべてのディスク・ボリュームの GUID の表示 . . . . .	1102	db2sysstray - DB2 システム・トレイの開始 . . . . .	1329
db2logsForRfwd - ロールフォワード・リカバリー に必要なログのリスト . . . . .	1103	db2tapemgr - テープ上のログ・ファイルの管理 . . . . .	1332
db2look - DB2 統計および DDL 抽出ツール . . .	1104	db2tbst - 表スペース状態の獲得 . . . . .	1336
db2ls - インストール済みの DB2 製品およびフィー チャーのリスト . . . . .	1118	db2tdbmgr - ツール・カタログ・データベースのマ イグレーション . . . . .	1337
db2move - データベース移動ツール . . . . .	1121	db2top - DB2 モニター・ツール . . . . .	1338
db2mqdsn - MQListener . . . . .	1132	db2trc - トレース . . . . .	1343
db2mscs - Windows フェイルオーバー・ユーティ リティーのセットアップ . . . . .	1136	db2trccoff - db2trc のトレース OFF オプション . . . . .	1360
db2mtrk - メモリー・トラッカー . . . . .	1141	db2trcon - db2trc のトレース ON オプション . . . . .	1361
db2nchg - データベース・パーティション・サーバ ー構成の変更 . . . . .	1146	db2unins - DB2 データベース製品、フィーチャ ー、または言語のアンインストール . . . . .	1363
db2nprt - インスタンスへのデータベース・パーテ ィション・サーバーの追加 . . . . .	1148	db2untag - コンテナ・タグの解放 . . . . .	1366
db2ndrop - インスタンスからのデータベース・パー ティション・サーバーのドロップ . . . . .	1150	db2updsvr - 製品の更新の表示 . . . . .	1367
db2nrcfg - 非 root インストール構成ツール . . .	1152	db2val - DB2 コピー妥当性検査ツール . . . . .	1368
db2nrupdt - 非 root インストールを行ったインス タンスの更新 . . . . .	1153	db2xdbmig - XSR オブジェクトのマイグレーショ ン . . . . .	1370
db2nrupgrad - 非 root インスタンスのアップグレ ード . . . . .	1154	db2xpert - トラップ・ファイルのフォーマット . . . . .	1371
db2osconf - カーネル・パラメーター値のためのユ ーティリティー . . . . .	1156	disable_MQFunctions - WebSphere MQ 関数の無効 化 . . . . .	1372
db2pd - DB2 データベースのモニターおよびトラ ブルシューティング . . . . .	1159	doce_deinstall - DB2 インフォメーション・センタ ーのアンインストール . . . . .	1373
db2pdcfg - 問題判別動作の DB2 データベース の構成 . . . . .	1261	doce_install - DB2 インフォメーション・センタ ーのインストール . . . . .	1375
db2perfc - データベース・パフォーマンス値のリ セット . . . . .	1269	enable_MQFunctions - WebSphere MQ 関数の有効 化 . . . . .	1377
db2perfi - パフォーマンス・カウンター登録ユーテ ィリティー . . . . .	1271	installDSDriver - IBM Data Server Driver コンポー ネントの抽出 . . . . .	1380
db2perfr - パフォーマンス・モニター登録ツール .	1272	installFixPack - インストール済み DB2 データベ ース製品の更新 . . . . .	1382
db2prereqcheck - インストール前提条件の検査 .	1273	setup - DB2 データベース製品のインストール . . . . .	1388
db2rbind - すべてのパッケージの再バインド . . .	1275		
db2relocatedb - データベースの再配置 . . . . .	1277		
db2rfe - 非 root インストール用 root フィーチャ ー使用可能化 . . . . .	1284		

## 第 8 章 DB2 Text Search のコマンド 1391

db2ts ALTER INDEX . . . . .	1392
db2ts CLEANUP FOR TEXT . . . . .	1397
db2ts CLEAR COMMAND LOCKS . . . . .	1398
db2ts CLEAR EVENTS FOR TEXT . . . . .	1400
db2ts CREATE INDEX . . . . .	1402



db2ts DISABLE DATABASE FOR TEXT . . . . .	1410
db2ts DROP INDEX . . . . .	1412
db2ts ENABLE DATABASE FOR TEXT . . . . .	1414
db2ts HELP. . . . .	1416
db2ts START FOR TEXT . . . . .	1418
db2ts STOP FOR TEXT . . . . .	1420
db2ts UPDATE INDEX . . . . .	1422

<b>付録 A. 命名規則 . . . . .</b>	<b>1425</b>
命名規則 . . . . .	1426

<b>付録 B. ファイル・タイプ修飾子および区切り文字 . . . . .</b>	<b>1429</b>
エクスポート・ユーティリティ、インポート・ユーティリティ、ロード・ユーティリティで使用するファイル・タイプ修飾子 . . . . .	1430
データ移動時の区切り文字に関する考慮事項 . . . . .	1431

<b>付録 C. DB2 技術情報の概説 . . . . .</b>	<b>1433</b>
------------------------------------	-------------

DB2 テクニカル・ライブラリー (ハードコピーまたは PDF 形式) . . . . .	1434
コマンド行プロセッサから SQL 状態ヘルプを表示する . . . . .	1436
異なるバージョンの DB2 インフォメーション・センターへのアクセス . . . . .	1437
コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの更新 . . . . .	1437
コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの手動更新 . . . . .	1439
DB2 チュートリアル . . . . .	1441
DB2 トラブルシューティング情報 . . . . .	1442
ご利用条件 . . . . .	1442

<b>付録 D. 特記事項 . . . . .</b>	<b>1445</b>
-----------------------------	-------------

<b>索引 . . . . .</b>	<b>1449</b>
---------------------	-------------



---

## 本書について

本書には、データベース管理機能を実行するためのシステム・コマンドと IBM® DB2® コマンド行プロセッサ (CLP) の使用に関する情報が記載されています。

---

## 本書の対象読者

本書は、読者がデータベース管理について理解していること、また構造化照会言語 (SQL) についての知識があることを前提としています。

---

## 本書の構成

本書には、CLP、システム、および DB2 Text Search の各コマンドを使用するために必要な参照情報が記載されています。

本書の各章で説明する主なサブジェクト・エリアは、以下のとおりです。

### コマンド行プロセッサ (CLP) の使用

- 第 1 章『コマンド行プロセッサ (CLP)』では、コマンド行プロセッサを起動して使用する方法を説明し、CLP オプションについて説明しています。
- 第 2 章『コマンド行 SQL ステートメントと XQuery ステートメントの使用』には、コマンド行から SQL ステートメントを使用する方法についての情報があります。

### Command Line Processor Plus (CLPPlus) の使用

- 第 3 章『Command Line Processor Plus (CLPPlus)』では、Command Line Processor Plus インターフェースをインストールして起動し、使用方法を説明しています。

### コマンド使用法ヘルプ

- 第 4 章『コマンド構文ヘルプの読み方』では、コマンド・ヘルプ画面の呼び出し方を説明し、採用されているコマンド・ヘルプ画面構文規則を解説します。

### CLP コマンド

- 第 5 章『CLP コマンド』では、すべてのデータベース・マネージャー・コマンドをアルファベット順にリストし、説明しています。

### CLPPlus コマンド

- 第 6 章『CLPPlus コマンド』では、CLPPlus インターフェースで使用できるすべてのコマンドをアルファベット順にリストし、説明しています。

### システム・コマンド

- 第 7 章『システム・コマンド』では、オペレーティング・システムのコマンド・プロンプトまたはシェル・スクリプトから実行できる、データベース・マネージャーにアクセスするためのすべてのコマンドをアルファベット順にリストし、説明しています。

### DB2 Text Search のコマンド

- 第 8 章『DB2 Text Search のコマンド』では、**db2ts** を接頭部としてオペレーティング・システムのコマンド・プロンプトに入力できるすべての Text Search コマンドをアルファベット順にリストし、説明しています。

### 付録

- 付録 A『命名規則』では、データベースや表などのオブジェクトの命名規則について説明しています。
- 付録 B『ファイル・タイプ修飾子および区切り文字』では、ロード、インポート、およびエクスポート・ユーティリティのファイル・タイプ修飾子について説明しています。さらに、データの移動のための区切り文字の考慮事項についても説明しています。

---

## 強調表記規則

本書では、以下の強調表記規則を使用します。

太字	コマンド、キーワード、および名前がシステムによって事前定義されている他の項目を表します。大文字で表示されているコマンドは CLP コマンドで、小文字で表示されているコマンドはシステム・コマンドです。
イタリック	次のいずれかの項目を示します。 <ul style="list-style-type: none"><li>ユーザーが指定する必要がある名前または値 (変数)</li><li>一般的な強調</li><li>新しい用語の紹介</li><li>他の情報源の参照</li></ul>
モノスペース	次のいずれかの項目を示します。 <ul style="list-style-type: none"><li>ファイルおよびディレクトリー</li><li>コマンド・プロンプトまたはウィンドウでユーザーが入力する必要がある情報</li><li>特定のデータ値の例</li><li>システムで表示されるものと類似したテキストの例</li><li>システム・メッセージの例</li><li>プログラミング・コードのサンプル</li></ul>

---

## 構文図の見方

このトピックでは、SQL 構文ダイアグラムの構造について説明します。

構文図は、左から右、上から下に、線に沿って読みます。

記号  $\blacktriangleright$ — は、構文図の始まりを示します。

記号 — $\blacktriangleright$  は、構文が次の行に続くことを示します。

記号  $\blacktriangleleft$ — は、構文が前の行から続いていることを示します。

記号 — $\blacktriangleleft$  は、構文図の終わりを示します。

構文フラグメントは、記号 |— で始まり、記号 —| で終わります。

必須項目は、横線 (メインパス) 上に示されます。



オプション項目は、メインパスの下に示されます。



オプション項目をメインパスの上に示すこともありますが、それは構文図を見やすくするためであり、実行には関係しません。

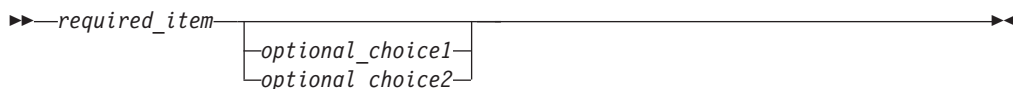


複数の項目からの選択が可能な場合、それらの項目を縦に並べて (スタックに) 示しています。

項目から 1 つを選択しなければならない場合、スタックの項目の 1 つはメインパス上に示されます。

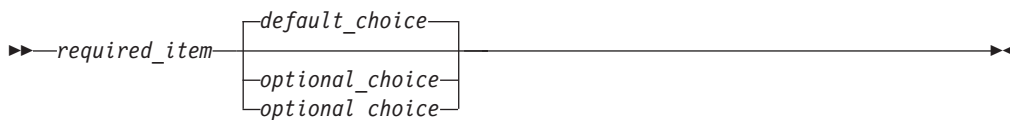


項目から 1 つをオプションで選択できる場合、スタック全体がメインパスよりも下に示されます。



## 構文図の見方

項目の 1 つがデフォルト値の場合、その項目はメインパスより上に示され、残りの選択項目はメインパスよりも下に示されます。



メインパスの上に、左へ戻る矢印がある場合には、項目を繰り返して指定できることを示しています。このような場合、繰り返す項目相互の間は、1 つ以上の空白で区切らなければなりません。



繰り返しの矢印にコンマが示されている場合は、繰り返し項目をコンマで区切らなければなりません。



スタックの上部の反復の矢印の記号は、そのスタックの中から複数の項目を選択できること、または 1 つの選択項目を繰り返して選択できることを示します。

キーワードは英大文字で示してあります (例: FROM)。示されているとおりに入力することが必要です。変数は英小文字で示してあります (例: column-name)。このような変数は、構文にユーザーが指定する名前や値を示しています。

句読点、括弧、算術演算子、その他の記号が示されている場合には、それらを構文の一部として入力する必要があります。

1 つの変数が、構文を構成する大きいフラグメントを表すことがあります。例えば次の図で、変数 `parameter-block` は、**parameter-block** というラベルの構文フラグメント全体を表します。



### parameter-block:



黒丸 (●) ではさまれて隣接しているセグメントは、任意の順序で指定することができます。



上記の図は、item2 と item3 をどのような順序で指定しても構わないことを示しています。以下はいずれも有効です。

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



## 構文図の見方

---

## 第 1 章 コマンド行プロセッサ (CLP)

### コマンド行プロセッサのフィーチャー

このセクションでは、コマンド行プロセッサのフィーチャーに関する情報を記載しています。

コマンド行プロセッサは以下の順序で動作します。

- CLP コマンド (大文字または小文字) をコマンド・プロンプトに入力します。
- ENTER キーを押すと、コマンドがコマンド・シェルに送られます。
- 出力の宛先が標準出力装置に自動的に指定されます。
- パイピングとリダイレクトがサポートされます。
- 処理の成功または失敗が、ユーザーに通知されます。
- コマンドの実行の後に、制御がオペレーティング・システム・コマンド・プロンプトに戻され、ユーザーは続けて次のコマンドを入力することができるようになります。
- ファイル入力オプションを指定して CLP を呼び出すと、自動的に CLIENT APPLNAME 特殊レジスターが CLP *filename* に設定されます。

以下のいずれかで、コマンド行プロセッサを開始できます。

- **db2** コマンドを入力します。または、
- Linux オペレーティング・システムの場合、「メイン・メニュー (Main Menu)」をクリックして、「**IBM DB2**」 > 「**コマンド行プロセッサ**」を選択します。

特定の CLP コマンドと SQL ステートメントを実行するには、サーバー・インスタンスが稼働中であり、データベース接続が存在する必要があります。以下のいずれかのアクションを実行して、データベースに接続します。

- 以下の SQL ステートメントを発行します。

```
db2 connect to database
```
- DB2 レジストリー変数 **DB2DBDFT** に定義されているデフォルト・データベースに暗黙接続を確立します。

コマンドの文字数がコマンド・プロンプトに許可されている文字制限を超える場合、円記号 (¥) を行継続文字として使用することができます。コマンド行プロセッサは行継続文字を検出すると、次の行を読み取って、その行にある文字を前の行の文字に連結します。その代わりに、**-t** オプションを使用して別の行終了文字を設定することもできます。

コマンド行プロセッサは、NULL と呼ばれるストリングを NULL ストリングとして識別します。以前に何らかの値に設定したフィールドを後で NULL に設定できません。例えば、

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

この場合、**tm\_database** フィールドが NULL 値に設定されます。この操作は大文字小文字を区別します。小文字の `null` は NULL ストリングとしては解釈されず、`null` を含むストリングとして解釈されます。

## コマンド行プロセッサのカスタマイズ

**DB2\_CLPPROMPT** レジストリー変数を使用することによって、対話式入力プロンプトをカスタマイズすることができます。このレジストリー変数には最大長 100 の任意のテキスト・ストリングを設定することができ、トークン %i、%ia、%d、%da および %n を含めることができます。実行時にこれらのトークンが特定の値に置き換えられます。

表 1. DB2\_CLPPROMPT トークンおよびランタイムの値

DB2_CLPPROMPT トークン	実行時の値
%ia	現行インスタンス接続の許可 ID
%i	現在アタッチされているインスタンスのローカル別名。インスタンス接続が存在していない場合は、 <b>DB2INSTANCE</b> レジストリー変数の値。Windows プラットフォーム上でのみ、 <b>DB2INSTANCE</b> レジストリー変数が設定されていない場合、 <b>DB2INSTDEF</b> レジストリー変数の値。
%da	現行データベース接続の許可 ID
%d	現在接続されているデータベースのローカル別名。データベース接続が存在していない場合は、 <b>DB2DBDFT</b> レジストリー変数の値。
%n	改行

- 実行時にどのトークンにも関連する値がない場合は、そのトークンが空のストリングに置き換えられます。
- 対話式入力プロンプトは常に許可 ID、データベース名、およびインスタンス名を大文字で提示するので、プロンプトに表示される接続および接続情報と一貫したものになります。
- DB2\_CLPPROMPT** レジストリー変数が CLP 対話モード内で変更された場合、CLP 対話モードが閉じて再オープンされるまで、**DB2\_CLPPROMPT** の新しい値は有効になりません。

**DB2\_CLPHISTSIZE** レジストリー変数を使用すると、コマンド履歴に保管できるコマンドの数を指定できます。**HISTORY** コマンドを使用すると、CLP 対話式モード・セッション内で実行したコマンド履歴の内容にアクセスできます。

また、**DB2\_CLP\_EDITOR** レジストリー変数を使用して、**EDIT** コマンドの発行時にオープンされるエディターを指定できます。CLP 対話式セッションで **EDIT** コマンドを使用すると、ユーザー指定されたコマンドがプリロードされたエディターが開くので、編集してから実行できます。

### 例

**DB2\_CLPPROMPT** が (%ia%i, %da%d) と定義されている場合、入力プロンプトは以下の値を持ちます。

- インスタンス接続なし、およびデータベース接続なし。**DB2INSTANCE** は DB2 に設定される。**DB2DBDFT** は設定されない。  
(@DB2, @)
- (Windows) インスタンス接続なし、およびデータベース接続なし。**DB2INSTANCE** と **DB2DBDFT** は設定されない。**DB2INSTDEF** は DB2 に設定される。

## コマンド行プロセッサのフィーチャー

(@DB2, @)

- インスタンス接続なし、およびデータベース接続なし。 **DB2INSTANCE** は DB2 に設定される。**DB2DBDFT** は "SAMPLE" に設定される。

(@DB2, @SAMPLE)

- 許可 ID "keon14" を使用してインスタンス "DB2" にインスタンス接続。**DB2INSTANCE** は DB2 に設定される。**DB2DBDFT** は "SAMPLE" に設定される。

(KEON14@DB2, @SAMPLE)

- 許可 ID "horton7" を使用してデータベース "sample" にデータベース接続。**DB2INSTANCE** は DB2 に設定される。**DB2DBDFT** は SAMPLE に設定される。

(@DB2, HORTON7@SAMPLE)

- 許可 ID "keon14" を使用してインスタンス "DB2" にインスタンス接続。許可 ID "horton7" を使用してデータベース "sample" にデータベース接続。**DB2INSTANCE** は DB2 に設定される。**DB2DBDFT** は設定されない。

(KEON14@DB2, HORTON7@SAMPLE)

## コマンド・ファイルでのコマンド行プロセッサの使用

データベース・マネージャーへの CLP 要求をシェル・スクリプトのコマンド・ファイルに組み込むことができます。次の例では、シェル・スクリプト・コマンド・ファイルに CREATE TABLE ステートメントを入力する方法を示しています。

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

コマンドおよびコマンド・ファイルの詳細については、適切なオペレーティング・システムの解説書を参照してください。

## コマンド行プロセッサの設計

コマンド行プロセッサは 2 つのプロセス、つまりユーザー・インターフェースとして機能するフロントエンド・プロセス (DB2 コマンド) と、データベース接続を維持するバックエンド・プロセス (**db2bp**) とで構成されています。

### データベース接続の保守

**db2** が呼び出されるたびに、新しいフロントエンド・プロセスが開始されます。バックエンド・プロセスは、最初の **db2** 呼び出しのときに開始し、**TERMINATE** によって明示的に終了します。親が同じであるフロントエンド・プロセスはすべて単一のバックエンド・プロセスにより保守されているので、単一のデータベース接続を共有していることとなります。

例えば以下のように、同じオペレーティング・システムのコマンド・プロンプトから複数の **db2** 呼び出しを行うと、1 つのバックエンド・プロセスを共有する複数のフロントエンド・プロセスが個別に開始します。そしてこのバックエンド・プロセスがデータベース接続を保留します。

- db2 'connect to sample',
- db2 'select \* from org',
- . test01 (test01 は DB2 コマンドを含むシェル・スクリプト)
- db2 -tf myfile.clp

以下のように、同じオペレーティング・システム・プロンプトから複数の呼び出しを行うと、それぞれの呼び出しの親プロセスが異なるために個別のバックエンド・プロセスが必要になり、結果として複数のデータベース接続が個別に開始してしまいます。

- test01
- . test01 &
- test01 &
- sh test01

### フロントエンド・プロセスとバックエンド・プロセスとの間の通信

フロントエンド・プロセスとバックエンド・プロセスとの間の通信は、要求キュー、入力キュー、および出力キューの 3 つのメッセージ・キューを介して行います。

### 環境変数

以下の環境変数によって、2 つのプロセス間での通信が構成されます。

表 2. 環境変数

変数	最小	最大	デフォルト
DB2BQTIME	1 秒	5294967295	1 秒
DB2BQTRY	試行 0 回	5294967295	試行 60 回
DB2RQTIME	1 秒	5294967295	5 秒
DB2IQTIME	1 秒	5294967295	5 秒

#### DB2BQTIME

コマンド行プロセッサを呼び出すと、フロントエンド・プロセスは、バックエンド・プロセスが既にアクティブになっているかどうかを調べます。アクティブになっている場合、フロントエンド・プロセスは接続を再び確立します。アクティブになっていない場合、フロントエンド・プロセスがバックエンド・プロセスをアクティブ化します。フロントエンド・プロセスは、次に **DB2BQTIME** 変数に指定された期間、活動を停止し、その後もう一度検査します。フロントエンド・プロセスは、**DB2BQTRY** 変数に指定した回数だけ検査を継続し、それでもバックエンド・プロセスがアクティブになっていない場合は、時間切れとなりエラー・メッセージを戻します。

#### DB2BQTRY

**DB2BQTIME** 変数と連携しており、バックエンド・プロセスがアクティブになっているかどうかの判別をフロントエンド・プロセスが試行する回数を指定します。

**DB2BQTIME** および **DB2BQTRY** の値は、ピーク時に照会時間を最適化するために増やすことができます。

#### DB2RQTIME

一度バックエンド・プロセスが開始すると、このプロセスはフロントエンドからの要求があるまで要求キューで待機します。さらに、コマンド・プロンプトから要求が開始されてから次の要求が開始されるまでの間も、要求キューで待機します。

**DB2RQTIME** 変数は、バックエンド・プロセスがフロントエンド・プロセスからの要求を待機する長さを指定します。この時間が経過すると、要求キューに要求がない場合、バックエンド・プロセスはフロントエンド・プロセスの親がまだ存在しているかを調べ、存在していないことが分かるとバックエンド・プロセスは終了します。親が存在している場合は、要求キューで待機を継続します。

### DB2IQTIME

バックエンド・プロセスがフロントエンド・プロセスから要求を受信すると、バックエンド・プロセスはフロントエンド・プロセスに肯定応答を送り、入力キューを介して入力を受け取る準備ができていないことを知らせます。そして、バックエンド・プロセスは入力キューで待機します。さらに、バッチ・ファイル (**-f** オプションで指定) が実行している間、またユーザーが対話式モードの間も、バックエンド・プロセスは入力キューで待機します。

**DB2IQTIME** 変数は、フロントエンド・プロセスの入力キューがコマンドを渡すまで、バックエンド・プロセスが待機する時間を指定します。指定した時間が経過すると、バックエンド・プロセスは、フロントエンド・プロセスがアクティブになっているかどうかを調べ、フロントエンド・プロセスが既に存在していない場合は、要求キューに戻って待機します。フロントエンド・プロセスが存在している場合は、バックエンド・プロセスはフロントエンド・プロセスからの入力を待機します。

これらの環境変数の値を表示するには、**LIST COMMAND OPTIONS** を使用します。

バックエンド環境変数は、バックエンド・プロセスが開始されるときにフロントエンド・プロセスから値の集合を継承します。ただし、フロントエンド環境変数が変更された場合は、バックエンド・プロセスはその変更を継承することはありません。変更した値を継承するには、まずバックエンド・プロセスを終了してから、再始動 (**db2** コマンドを実行する) しなければなりません。

バックエンド・プロセスを終了しなければならない場合の例を、次のシナリオで示します。

1. ユーザー A がログオンし、いくつか **CLP** コマンドを出した後、**TERMINATE** を発行せずにログオフします。
2. ユーザー B が同じウィンドウを使用してログオンします。
3. ユーザー B が特定の **CLP** コマンドを発行すると、コマンドはメッセージ **DB21016** (システム・エラー) を出して失敗します。

ユーザー B のフロントエンド・プロセス (コマンドが出されたオペレーティング・システム・ウィンドウ) の親が依然としてアクティブであるため、ユーザー A が開始したバックエンド・プロセスは、ユーザー B が **CLP** を使用して開始した時点でもまだアクティブになっています。バックエンド・プロセスは、ユーザー B が発行した新規コマンドを処理しようとしませんが、ユーザー B のフロントエンド・プロセスには、バックエンド・プロセスのメッセージ・キューを使用するだけの十分な権限がありません。そのためにはバックエンド・プロセスを作成したユーザー A の権限が必要だか



らです。CLP セッションを **TERMINATE** コマンドを使用して終了してから、ユーザーは同じオペレーティング・システム・ウィンドウを使用して、新規 CLP セッションを開始しなければなりません。これによって新規ユーザーごとに新しいバックエンド・プロセスが作成されることになり、権限の問題は回避され、新規ユーザーのバックエンド・プロセス内で環境変数 (**DB2INSTANCE** など) の値を訂正して設定できます。

### CLP 使用上の注意

コマンドはコマンド・プロンプトから大文字または小文字で入力できます。ただし、DB2 が大文字小文字を区別するパラメーターには、大文字小文字を正確に入力しなければなりません。例えば、**CHANGE DATABASE COMMENT** の **WITH** 節にある *comment-string* は、大文字小文字を区別するパラメーターです。

区切り付き ID は、SQL ステートメント内で使用できます。

特殊文字またはメタ文字 (例えば、\$ & \* ( ) ; < > ? ¥ ' " など) は、CLP コマンド内で使用できます。これらの文字が CLP 対話式モード以外または CLP バッチ入力モード以外で使用されている場合、これらの文字はオペレーティング・システム・シェルによって解釈されます。シェルが特殊な処置を行わない場合は、引用符またはエスケープ文字が必要です。

例えば、AIX® Korn シェル環境でコマンド

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

を実行すると、"select <the names of all files> from org where division" と解釈されます。この結果の SQL 構文エラーは、ファイル Eastern にリダイレクトされます。次の構文であれば、正しい出力が生成されます。

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

特殊文字は、プラットフォームごとに異なります。AIX Korn シェルでは、前の例はエスケープ文字 (¥)、例えば ¥\*、¥>、または ¥' などを使って書き直すことができます。

ほとんどのオペレーティング・システム環境では、入力と出力をリダイレクトすることが可能です。例えば、SAMPLE データベースに接続している場合、次に示す要求は STAFF 表を照会し、出力を mydata ディレクトリーにある staflist.txt という名前のファイルに送ります。

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

出力のリダイレクトがサポートされていない環境では、CLP オプションを使用することができます。例えば、上記の要求は以下のように書き直すことができます。

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

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

コマンド行プロセッサはプログラミング言語ではありません。例えば、ホスト変数はサポートしません。そのため、次のステートメント、

```
db2 connect to :HostVar in share mode
```

は、:HostVar が有効なデータベース名ではないため、構文上、正しくありません。

## コマンド行プロセッサのフィーチャー

コマンド行プロセッサは、SQL NULL 値をハイフン (-) で表します。列が数値である場合、ハイフンは列の一番右に置きます。列が数値ではない場合、ハイフンは列の一番左に置きます。

DB2 コマンド行プロセッサ・ウィンドウから 1 バイト (SBCS) 言語の国別文字を正しく表示するには、True Type フォントを選択する必要があります。例えば、Windows 環境では、コマンド・ウィンドウのプロパティ・ノートブックを開いて、Lucinda Console などのフォントを選択します。

コマンド行プロセッサは、ファイル・パス名中の各国語サポート (NLS) 文字をサポートしていません。このことは、特に **IMPORT**、**EXPORT**、および **REGISTER XMLSCHEMA** などのコマンドに影響があります。これらのコマンドでは、問題になるファイル・パス名が最も頻繁に使用されるからです。

コマンド行プロセッサ・コマンド出力の他のコマンド行プロセッサ・コマンドへのパイピングはサポートされています。例えば、**db2 -x <SQL\_statement> | db2 +p -tv** のようになります。このサポートには、パイプのバッファ・サイズ以外の制限はありません。パイプのバッファ・サイズは構成可能ではありません。最初のコマンドでパイプのバッファ・サイズを超えた場合、コマンド行プロセッサはハングするか、パイプのバッファに出力を書き込もうとする際に障害が起こります。2 番目のコマンドがコマンド行プロセッサ・コマンドではない場合、例えば UNIX シェル・コマンドの場合、パイプのバッファ・サイズ制限が原因でハングしたり障害が起きたりすることはありません。

## db2 - コマンド行プロセッサの呼び出し

**db2** コマンドはコマンド行プロセッサ (CLP) を開始します。CLP は、データベース・ユーティリティ、SQL ステートメントおよびオンライン・ヘルプを実行するために使用します。

これにはさまざまなコマンド・オプションがあり、以下のモードで開始することができます。

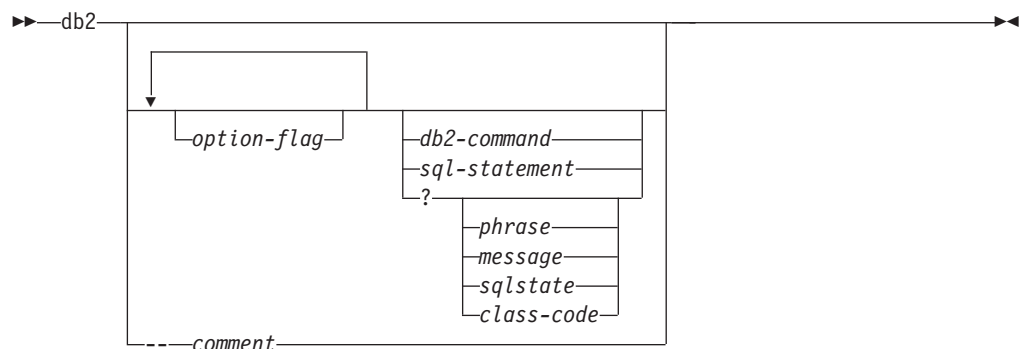
- 対話式入力モード。特徴は `db2 =>` 入力プロンプトです。
- コマンド・モード。各コマンドの前に **db2** を付ける必要があります。
- バッチ・モード。 **-f** ファイル入力オプションを使用します。

Windows オペレーティング・システムで **db2cmd** を実行すると、CLP 対応の DB2 ウィンドウが開き、DB2 コマンド行環境が初期化されます。このコマンドを実行することは、「**DB2 コマンド・ウィンドウ**」アイコンをクリックすることと同じです。

**QUIT** はコマンド行プロセッサを停止します。**TERMINATE** もコマンド行プロセッサを停止しますが、関連するバックエンド・プロセスを除去し、使用されていたメモリーをすべて解放します。**TERMINATE** の発行を、すべての **STOP DATABASE MANAGER (db2stop)** コマンドより優先させることをお勧めします。また、データベース構成パラメーターの変更を有効にするために、それらの変更の後に **TERMINATE** の発行が必要になることもあります。CLP を終了する前に、既存の接続をリセットしなければなりません。

シェル・コマンド (!) を使うと、Linux および UNIX オペレーティング・システム、および Windows オペレーティング・システム上で、オペレーティング・システムのコマンドを対話式またはバッチ・モードで実行できるようになります (例えば、UNIX の場合 `!ls`、Windows オペレーティング・システムの場合 `!dir`)。

### コマンド構文



#### option-flag

CLP オプション・フラグを指定します。

#### db2-command

DB2 コマンドを指定します。

#### sql-statement

SQL ステートメントを指定します。

## db2 - コマンド行プロセッサの呼び出し

? CLP 一般ヘルプを要求します。

? **phrase**

指定したコマンドまたはトピックに関連のあるヘルプ・テキストを要求します。データベース・マネージャーは、要求した情報を見つけられない場合、一般ヘルプ画面を表示します。

? **options** は、CLP オプションの説明と、現在の設定値を要求します。 ? **help** は、オンライン・ヘルプ構文図を読み取るための情報を要求します。

? **message**

有効な SQLCODE (例えば ? sql10007n) により指定されているメッセージのヘルプを要求します。

? **sqlstate**

有効な SQLSTATE により指定されているメッセージのヘルプを要求します。

? **class-code**

有効なクラス・コードによって指定されているメッセージのヘルプを要求します。

-- **comment**

コマンド行プロセッサは、コメント文字 -- で始まる入力をコメントとして扱います。

各パラメーターについて、疑問符 (?) の後にスペースを入れて変数名から分ける必要があります。

## コマンド行プロセッサのオプション

CLP コマンド・オプションは、コマンド行プロセッサ **DB2OPTIONS** 環境変数 (大文字) を設定するか、またはコマンド行フラグを使用することにより指定できます。

ユーザーは、**DB2OPTIONS** を使用して、セッション全体に適用されるオプションを設定することができます。

オプション・フラグの現在の設定および **DB2OPTIONS** の値を表示するには、**LIST COMMAND OPTIONS** を使用します。オプションの設定を対話式入力モードまたはコマンド・ファイルから変更するには、**UPDATE COMMAND OPTIONS** を使用します。

コマンド行プロセッサは、以下の手順でオプションを設定します。

1. デフォルト・オプションを設定します。
2. **DB2OPTIONS** を読み取って、デフォルトを一時変更します。
3. コマンド行を読み取って、**DB2OPTIONS** を一時変更します。
4. **UPDATE COMMAND OPTIONS** からの入力を、最終的な対話式一時変更として受け入れられます。

表 3 に CLP オプション・フラグを要約してあります。これらのオプションは、どのような順序や組み合わせでも指定することができます。オプションをオンにするには、対応するオプション文字の前に負符号 (-) を付けます。オプションをオフにする場合、オプション文字の接頭部に負符号を付けオプション文字の後にも別の負符号を付けるか、またはオプション文字の接頭部に正符号 (+) を付けてください。例えば、**-c** で自動コミット・オプションはオンになり、**-c-** または **+c** でオフになります。これらのオプション文字では大文字と小文字は区別されず、**-a** と **-A** は同じものと見なされます。

表 3. CLP コマンド・オプション

オプション・フラグ	説明	デフォルト設定
<b>-a</b>	コマンド行プロセッサは <b>SQLCA</b> データを表示します。	OFF
<b>-b</b>	このオプションは、SQL ステートメントの実行に必要なすべての欠落パッケージや無効パッケージを自動作成するようコマンド行プロセッサに通知します。	ON
<b>-c</b>	コマンド行プロセッサは SQL ステートメントを自動的にコミットします。	ON
<b>-d</b>	このオプションは、コマンド行プロセッサが XML データの XML 宣言を検索して表示するよう指示します。	OFF
<b>-e{cls}</b>	コマンド行プロセッサは <b>SQLCODE</b> または <b>SQLSTATE</b> を表示します。この 2 つのオプションは相互に排他的です。	OFF
<b>-f filename</b>	コマンド行プロセッサは、コマンド入力を標準入力ではなくファイルから読み取ります。	OFF

## コマンド行プロセッサのオプション

表 3. CLP コマンド・オプション (続き)

オプション・フラグ	説明	デフォルト設定
-i	このオプションは、コマンド行プロセッサが、適切な字下げを使って XML データを「整形印刷」するよう指示します。このオプションは、XQuery ステートメントの結果セットにのみ影響を与えます。	OFF
-l <i>filename</i>	コマンド行プロセッサはコマンドを履歴ファイルに記録します。	OFF
-m	このオプションは、ステートメント INSERT、DELETE、UPDATE、および MERGE の影響を受ける行数を表示するようにコマンド行プロセッサに指示します。	OFF
-n	単一の区切りトークン内の改行文字を除去します。このオプションを指定しない場合、改行文字はスペースで置換されます。このオプションは、-t オプションと共に使用する必要があります。	OFF
-o	コマンド行プロセッサは、出力データおよびメッセージを標準出力に表示します。	ON
-p	コマンド行プロセッサは、対話型入力モードのときにコマンド行プロセッサ・プロンプトを表示します。	ON
-q	このオプションは、コマンド行プロセッサが、単一または二重引用符で区切られているストリング内の空白文字および改行文字を保持するよう指示します。オプション q が ON のとき、オプション n は無視されます。	OFF
-r <i>filename</i>	コマンド行プロセッサは、コマンドが生成したレポートをファイルに書き込みます。	OFF
-s	コマンド行プロセッサは、バッチ・ファイルまたは対話式モードでコマンドを実行中にエラーが発生した場合、実行を停止します。	OFF
-t	コマンド行プロセッサはセミコロン (;) をステートメント終了文字として使用します。	OFF
-tdx または -tdxx	このオプションは、コマンド行プロセッサが x または xx をステートメント終了文字 (長さは 1 文字または 2 文字) として定義し、使用するよう指示します。	OFF
-v	コマンド行プロセッサはコマンド・テキストを標準出力にエコーさせます。	OFF
-w	このオプションは、FETCH および SELECT の警告メッセージを表示するようにコマンド行プロセッサに指示します。	ON
-x	コマンド行プロセッサは列名などのヘッダーなしでデータを戻します。このフラグは、すべてのコマンドには影響を与えません。これは、SQL ステートメント、および SQL ステートメントに基づくいくつかのコマンド ( <b>LIST TABLES</b> など) に適用されます。	OFF
-z <i>filename</i>	コマンド行プロセッサはすべての出力をファイルにリダイレクトします。これは -r オプションと似ていますが、出力の他にメッセージやエラー・コードも含まれます。	OFF

## AIX コマンド

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

を実行すると、このセッションのデフォルト設定は以下のようになります。

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

以下に、これらのオプションの詳細を説明します。

### SQLCA データ表示オプション (-a):

DB2 コマンドまたは SQL ステートメントを実行した後で、SQLCA データを標準出力に表示します。SQLCA データが、エラーまたは完了メッセージの代わりに表示されます。

このコマンド・オプションのデフォルト設定は OFF です (+a または -a-)。

-o および -r オプションは -a オプションに影響します。詳細については、オプションの説明を参照してください。

### 自動バインド・オプション (-b):

DB2 V10.1 フィックスパック 2 以降、このオプションは、SQL ステートメントを実行するために必要な、すべての欠落パッケージや無効パッケージを作成します。OFF (+b) を設定すると、サーバーからコマンド行プロセッサにパッケージが見つからないというエラーやタイム・スタンプに矛盾があるというエラーがスローされても、コマンド行プロセッサはパッケージの再バインドを試みません。

このコマンド・オプションのデフォルト設定は ON です。

この新規コマンド行オプションも、DB2OPTIONS 環境変数を使用して設定できます。

#### ユースケース・シナリオ:

```
# explicitly dropping db2c1pcs.bnd from a database
db2 drop package NULLID.SQLC2J24
...
# turning off autobind ("+b") results in SQL0805N as db2c1pcs.bnd is dropped now
db2 +b "create table a (c1 int)"

# bind needed file explicitly now
db2 bind db2c1pcs.bnd

# now same SQL will be successful as binding is done explicitly
db2 +b "create table a (c1 int)"

# again explicitly dropping db2c1pcs.bnd from a database
db2 drop package NULLID.SQLC2J24
...
# keeping default autobind ("-b") results autobinding of missing packages and SQL execution is
success now

db2 -b "create table a (c1 int)"
-OR-
db2 "create table a (c1 int)"
```

### 自動コミット・オプション (-c):

このオプションは、各コマンドまたはステートメントを独立して処理するかどうかを指定します。ON (-c) に設定する場合、各コマンドまたはステートメントは自動的にコミットされるかまたはロールバックされます。コマンド



## コマンド行プロセッサのオプション

またはステートメントが正常に処理された場合、それ自体、およびそれ以前に自動コミット OFF (+c または -c-) で発行され、正常に発行されたコマンドおよびステートメントがすべてコミットされます。しかし、コマンドまたはステートメントが失敗した場合、それ自体、およびそれ以前に自動コミット OFF で発行され、正常に発行されたコマンドおよびステートメントはすべてロールバックされます。OFF (+c または -c-) に設定した場合、COMMIT または ROLLBACK ステートメントを明示的に発行しなければなりません。そうしないと、次に自動コミット ON (-c) でコマンドを発行したときに、COMMIT または ROLLBACK のどちらかのアクションが起こります。

このコマンド・オプションのデフォルト設定は ON です。

自動コミット・オプションは、コマンド行プロセッサの他のオプションには影響しません。

**例:** 以下のシナリオを考えてみましょう。

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

ステップ 4 の SQL ステートメントは、表 A に C2 という名前の列がないため失敗します。このステートメントは自動コミット ON (デフォルト) で発行されたため、ステップ 4 のステートメントだけでなく、ステップ 3 のステートメントもロールバックされます。これは、ステップ 3 が自動コミット OFF で発行されたためです。コマンド

```
db2 list tables
```

を実行しても、空のリストが戻されます。

### XML 宣言オプション (-d):

-d オプションにより、コマンド行プロセッサが、XML データの XML 宣言を検索して表示するかどうかが決まります。

ON (-d) に設定した場合、XML 宣言は検索されて表示されます。OFF (+d または -d-) に設定した場合、XML 宣言は検索および表示されません。このコマンド・オプションのデフォルト設定は OFF です。

XML 宣言オプションは、他のコマンド行プロセッサ・オプションには影響しません。

### SQLCODE/SQLSTATE 表示オプション (-e):

-e{c|s} オプションを指定すると、コマンド行プロセッサは SQLCODE (-ec) または SQLSTATE (-es) を標準出力に表示します。オプション -ec および -es は、CLP 対話式モードでは無効です。

このコマンド・オプションのデフォルト設定は OFF です (+e または -e-)。

-o および -r オプションは、-e オプションに影響します。詳細については、オプションの説明を参照してください。

SQLCODE/SQLSTATE 表示オプションは他のコマンド行プロセッサ・オプションには影響しません。

**例:** AIX 上で実行しているコマンド行プロセッサから SQLCODE を検索するには、次のように入力します。

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

### 入力ファイルからの読み取りオプション (-f):

**-f filename** オプションを指定すると、コマンド行プロセッサは標準入力ではなく指定したファイルから入力を読み取るようになります。 *filename* は絶対ファイル名または相対ファイル名で、ファイルまでのディレクトリー・パスを含むことができます。ディレクトリー・パスを指定していない場合、現行ディレクトリーが使用されます。

ファイル入力オプションを指定して CLP を呼び出すと、自動的に CLIENT APPLNAME 特殊レジスターが CLP *filename* に設定されます。

他のオプションと一緒にオプション **-f** を指定する場合には、オプション **-f** は最後に指定する必要があります。以下に例を示します。

```
db2 -tvf filename
```

**db2 -tvf filename** コマンドを使用して CLP スクリプト・ファイルを実行すると、CLIENT APPLNAME 特殊レジスターが CLP *filename* に設定されます。次に実行するコマンドによって、特殊レジスター CLIENT APPLNAME および CLIENT ACCTNG は、**db2 -tvf filename** コマンドが発行される前の古い値にリセットされます。次に実行するコマンドが **db2 terminate** であつたり、*filename* 内の最後のコマンドが **TERMINATE** であつたりすると、特殊レジスターはリセットされません。これは、どのバッチ・ジョブが現在実行中であるかをモニターし、CLP ワークロードを区別するのに有用です。

このオプションは、対話式モード内では変更できません。

このコマンド・オプションのデフォルト設定は OFF です (+f または -f-)。

コマンド処理は **QUIT** または **TERMINATE** が出されるまで、またはファイルの終わりまで続行されます。

このオプションとデータベース・コマンドの両方を指定すると、コマンド行プロセッサはコマンドをまったく処理せずにエラー・メッセージを戻します。

コメント文字 **--** で始まる入力ファイル行は、コマンド行プロセッサによりコメントとして処理されます。行の最初の非空白文字はコメント文字でなければなりません。

(= で開始する入力ファイル行は、コメント・ブロックの開始として処理されます。=) で終了する行は、コメント・ブロックの終了を知らせます。

(= で開始して =) で終了する入力行のブロックは、連続したコメントとしてコマンド行プロセッサによって処理されます。(= の前および =) の後には、スペースを置くことが許可されています。コメントをネストさせることも可能で、ステートメント内にネストさせて使用することもできます。コマンド終了文字 (;) は、=) の後では使用できません。

**-f filename** オプションを指定すると、**-p** オプションは無視されます。

入力ファイルからの読み取りオプションは、他のコマンド行プロセッサ・オプションには影響しません。

## コマンド行プロセッサのオプション

デフォルトの終了文字は、改行文字のいずれか (-t オプションで別に指定されていない限り) またはファイルの終わりであることに注意してください。

### 整形印刷オプション (-i):

-i オプションは、適切な字下げを使って XML データを「整形印刷」するようにコマンド行プロセッサに指示します。このオプションは、XQuery ステートメントの結果セットにのみ影響を与えます。

このコマンド・オプションのデフォルト設定は OFF です (+i または -i-)。

整形印刷オプションは、他のコマンド行プロセッサ・オプションには影響しません。

### 履歴ファイルへのコマンドのログ・オプション (-l):

-l *filename* オプションを指定すると、コマンド行プロセッサは指定したファイルにコマンドのログを記録します。この履歴ファイルには、実行したコマンドとその完了状況が記録されます。*filename* は絶対ファイル名または相対ファイル名で、ファイルまでのディレクトリー・パスを含むことができます。ディレクトリー・パスを指定していない場合、現行ディレクトリーが使用されます。指定したファイルまたはデフォルトのファイルが既に存在している場合、そのファイルに新しいログ項目が追加されます。

他のオプションと一緒にオプション -l を指定する場合には、オプション -l は最後に指定する必要があります。以下に例を示します。

```
db2 -tv1 filename
```

このコマンド・オプションのデフォルト設定は OFF です (+l または -l-)。

履歴ファイルへのコマンドのログ・オプションは、他のコマンド行プロセッサ・オプションには影響しません。

### 影響を受ける行数の表示オプション (-m):

-m オプションは、ステートメント INSERT、DELETE、UPDATE、または MERGE の影響を受ける行数を表示するかどうかをコマンド行プロセッサに指示します。

ON (-m) に設定した場合、ステートメント INSERT、DELETE、UPDATE、または MERGE の影響を受ける行数が表示されます。OFF (+m または -m-) に設定した場合、影響を受ける行数は表示されません。その他のステートメントでは、このオプションは無視されます。このコマンド・オプションのデフォルト設定は OFF です。

-o および -r オプションは -m オプションに影響します。詳細については、オプションの説明を参照してください。

### 改行文字除去オプション (-n):

単一の区切りトークン内の改行文字を除去します。このオプションを指定しない場合、改行文字はスペースで置換されます。このオプションは、対話式モード内では変更できません。

このコマンド・オプションのデフォルト設定は OFF です (+n または -n-)。

このオプションは、-t オプションと共に使用する必要があります。詳細については、オプションの説明を参照してください。

**出力表示オプション (-o):**

-o オプションを指定すると、コマンド行プロセッサは出力データおよびメッセージを標準出力に送信します。

このコマンド・オプションのデフォルト設定は ON です。

対話式モード開始情報は、このオプションによって影響を受けることはありません。出力データは、ユーザー指定コマンドからの実行結果のレポート出力、および SQLCA データ (要求した場合) で構成されます。

以下のオプションは、+o オプションの影響を受けることがあります。

- -r *filename*: 対話式開始情報は保存されません。
- -e: +o を指定しても、SQLCODE または SQLSTATE が標準出力に表示されます。
- -a: +o を指定しても、影響を受けません。-a、+o および -r*filename* を指定すると、SQLCA 情報がファイルに書き込まれます。

-o と -e オプションの両方を指定すると、データおよび SQLCODE または SQLSTATE のどちらかが画面に表示されます。

-o と -v オプションの両方を指定すると、データが表示され、発行した各コマンドのテキストが画面にエコーされます。

出力表示オプションは、他のコマンド行プロセッサ・オプションには影響しません。

**DB2 対話式プロンプト表示オプション (-p):**

-p オプションを指定すると、ユーザーが対話式モードになっているときに、コマンド行プロセッサはコマンド行プロセッサ・プロンプトを表示します。

このコマンド・オプションのデフォルト設定は ON です。

コマンドがコマンド行プロセッサにパイプ接続されているときには、プロンプトをオフにすると便利です。例えば、CLP コマンドを含むファイルを、次のコマンドを実行することによって実行できます。

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

-p オプションは、-f *filename* オプションを指定すると無視されます。

DB2 対話式プロンプト表示オプションは、他のコマンド行プロセッサ・オプションには影響しません。

**空白文字および改行文字の保持オプション (-q):**

-q オプションにより、コマンド行プロセッサは、単一または二重引用符で区切られているストリング内の空白文字および改行文字を保持します。

このコマンド・オプションのデフォルト設定は OFF です (+q または -q-)。

オプション -q が ON の場合、オプション -n は無視されます。

**報告ファイルへの保存オプション (-r):**

-r *filename* オプションは、コマンドによって生成される出力データをすべて指定したファイルに書き込まれるようにします。これは、キャプチャーしないと画面がスクロールして見えなくなってしまうレポートのキャプチャーに便利です。メッセージまたはエラー・コードはファイルに書き込まれません。*filename* は絶対ファイル名または相対ファイル名で、ファイルまでの

## コマンド行プロセッサのオプション

ディレクトリー・パスを含むことができます。ディレクトリー・パスを指定していない場合、現行ディレクトリーが使用されます。新しいレポート項目がファイルに追加されます。

このコマンド・オプションのデフォルト設定は OFF です (+r または -r-)。

-a オプションを指定すると、SQLCA データがファイルに書き込まれます。

-r オプションは -e オプションには影響しません。 -e オプションを指定すると、SQLCODE または SQLSTATE がファイルではなく標準出力に書き込まれます。

-r *filename* を **DB2OPTIONS** に設定すると、ユーザーはコマンド行から +r (または -r-) オプションを設定して、特定のコマンド呼び出しの出力データがファイルに書き込まれないようにすることができます。

報告ファイルへの保存オプションは、他のコマンド行プロセッサ・オプションには影響しません。

### コマンド・エラー時の実行の停止オプション (-s):

コマンドが対話式モードでまたは入力ファイルから発行され、構文またはコマンド・エラーが起きた場合、-s オプションを指定してあると、コマンド行プロセッサが実行を停止して、標準出力にエラー・メッセージが書き込まれます。

このコマンド・オプションのデフォルト設定は OFF です (+s または -s-)。この設定により、コマンド行プロセッサはエラー・メッセージを表示し、残りのコマンドの実行を続け、システム・エラーが発生したとき (戻りコード 8) に限って実行を停止します。

次の表に、この動作を要約します。

表 4. CLP 戻りコードおよびコマンド実行

戻りコード	-s オプション設定	+s オプション設定
0 (成功)	実行の継続	実行の継続
1 (0 行が選択された)	実行の継続	実行の継続
2 (警告)	実行の継続	実行の継続
4 (DB2 または SQL エラー)	実行の停止	実行の継続
8 (システム・エラー)	実行の停止	実行の停止

### ステートメント終了文字のオプション (-t と -tdx または -tdxx):

-t オプションを指定すると、コマンド行プロセッサはステートメント終了文字としてセミコロン (;) を使用するようになり、円記号 (¥) 行継続文字は無効になります。このオプションは、対話式モード内では変更できません。

このコマンド・オプションのデフォルト設定は OFF です (+t または -t-)。

注: XQuery ステートメントを発行するために CLP を使用する場合、セミコロン以外の終了文字を使用するのが最善の選択です。名前空間宣言もセミコロンで終了するため、こうすることによって、名前空間宣言を使用するステートメントまたは照会が誤って認識されないようにします。



長さが 1 または 2 文字の終了文字を定義するには、`-td` の後に選択した文字を指定します。例えば、`-td%` とすると `%` がステートメント終了文字として設定されます。またはその代わりに、入力ファイル中で `--#SET TERMINATOR` ディレクティブを使用してステートメント終了文字を設定することもできます。例:

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

または

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

`file2.txt` には、以下のステートメントがファイル中の最初のステートメントとして含まれます。

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

このコマンド・オプションのデフォルト設定は OFF です。

終了文字かどうかを調べられるのが各入力行の最後の 1 つまたは 2 つの非空白文字だけであるため、コマンド行の複数のステートメントを連結するために終了文字を使用することはできません。

ステートメント終了文字オプションは、他のコマンド行プロセッサ・オプションには影響しません。

### 冗長出力オプション (-v):

`-v` オプションを指定すると、コマンド行プロセッサはコマンドからの出力またはメッセージを表示する前に、ユーザーが入力したコマンド・テキストを (標準出力に) エコーします。ECHO にはこのオプションは無効です。

このコマンド・オプションのデフォルト設定は OFF です (+v または -v-)。

+o (または -o-) が指定された場合、`-v` オプションは無効になります。

冗長出力オプションは、他のコマンド行プロセッサ・オプションには影響しません。

### 警告メッセージ表示オプション (-w):

`-w` オプションにより、コマンド行プロセッサは、照会 (FETCH または SELECT ステートメント) 中に生じることのある警告メッセージを表示するかどうかを決定します。警告は照会実行のさまざまな段階で生じることがあるので、メッセージはデータが戻されるよりも前、その最中、またはその後に表示されることがあります。戻されるデータに警告メッセージが含まれないようにするには、このフラグを使用できます。

このコマンド・オプションのデフォルト設定は ON です。

### 列見出しの印刷抑制オプション (-x):

`-x` オプションは、列名などのヘッダーなしでデータを戻すようコマンド行プロセッサに通知します。このフラグは、すべてのコマンドには影響を与えません。これは、SQL ステートメント、および SQL ステートメントに基づくいくつかのコマンド (LIST TABLES など) に適用されます。

このコマンド・オプションのデフォルト設定は OFF です。

### すべての出力のファイルへの保存オプション (-z):

`-z filename` オプションは、コマンドによって生成される出力データをすべ

## コマンド行プロセッサのオプション

て指定したファイルに書き込まれるようにします。これは、キャプチャーしないと画面がスクロールして見えなくなってしまうレポートのキャプチャーに便利です。これは、`-r` オプションに似ていますが、メッセージ、エラー・コード、および情報出力もファイルに書き込まれる点が異なります。`filename` は絶対ファイル名または相対ファイル名で、ファイルまでのディレクトリー・パスを含むことができます。ディレクトリー・パスを指定していない場合、現行ディレクトリーが使用されます。新しいレポート項目がファイルに追加されます。

このコマンド・オプションのデフォルト設定は OFF です (`+z` または `-z-`)。

`-a` オプションを指定すると、SQLCA データがファイルに書き込まれます。

`-z` オプションは `-e` オプションには影響しません。 `-e` オプションを指定すると、SQLCODE または SQLSTATE がファイルではなく標準出力に書き込まれます。

`-z filename` を **DB2OPTIONS** に設定すると、ユーザーはコマンド行から `+z` (または `-z-`) オプションを設定して、特定のコマンド呼び出しの出力データがファイルに書き込まれないようにすることができます。

すべての出力のファイルへの保存オプションは、他のコマンド行プロセッサ・オプションには影響しません。



## コマンド行プロセッサの戻りコード

このセクションでは、コマンド行プロセッサの戻りコードに関する情報を記載しています。

コマンド行プロセッサはコマンドの処理または SQL ステートメントの処理を終了すると、戻り (または終了) コードを戻します。これらのコードは、このコマンド行から CLP 関数を実行しているユーザーには意識されませんが、これらの関数をシェル・スクリプトから実行した場合は、コードを検索することができます。

例えば、次の B シェル・スクリプトは、GET DATABASE MANAGER CONFIGURATION コマンドを実行してから、CLP 戻りコードを検査します。

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

戻りコードは以下の値のいずれかになります。

### コード 説明

- 0 DB2 コマンドまたは SQL ステートメントが正常に実行されました。
- 1 SELECT または FETCH ステートメントが行を戻しませんでした。
- 2 DB2 コマンドまたは SQL ステートメントからの警告です。
- 4 DB2 コマンドまたは SQL ステートメント・エラーです。
- 8 コマンド行プロセッサのシステム・エラーです。

ユーザーが対話式モードでステートメントを実行しているとき、または入力を (-f オプションを使用して) ファイルから読み込んでいるときは、コマンド行プロセッサは戻りコードを戻しません。

戻りコードを使用できるのは、ユーザーが対話式モードを終了した後、または入力ファイルの処理が終了したときだけです。この場合、戻りコードは、その時点までに実行された個々のコマンドまたはステートメントから戻された個別のコードの論理和になります。

例えば、ユーザーが対話式モードでコマンドを発行した結果が戻りコード 0、1、および 2 になった場合、戻りコード 3 はユーザーが対話式モードを終了した後で戻されます。個々のコード 0、1、および 2 は戻されません。戻りコード 3 によって、対話式モード処理中に、1 つ以上のコマンドが 1 を戻し、1 つ以上のコマンドが 2 を戻したことがユーザーに通知されます。

戻りコード 4 になるのは、DB2 コマンドまたは SQL ステートメントが負の SQLCODE を戻したときです。戻りコード 8 になるのは、コマンド行プロセッサでシステム・エラーが起きたときだけです。

コマンドが入力ファイルからまたは対話式モードで発行され、かつコマンド行プロセッサでシステム・エラーが起きた場合 (戻りコード 8)、コマンドの実行は直ちに停止されます。1 つ以上の DB2 コマンドまたは SQL ステートメントがエラーで終了した場合 (戻りコード 4)、-s (コマンド・エラー時の実行の停止) オプション

## コマンド行プロセッサの戻りコード

ンが設定されていると、コマンドの実行が停止します。このオプションが設定されていないならば、コマンドの実行は継続します。

---

## コマンド行プロセッサからのコマンド・ヘルプの呼び出し

コマンド・ヘルプは、コマンド行プロセッサのコマンドの構文を説明します。

### 手順

コマンド・ヘルプを呼び出すには、コマンド行プロセッサを開いて、次のように入力します。

```
? command
```

ここで、*command* はキーワードまたはコマンド全体を表します。

例えば、? catalog はすべての **CATALOG** コマンドのヘルプを表示し、? catalog database は **CATALOG DATABASE** コマンドのみのヘルプを表示します。

---

## コマンド行プロセッサからのメッセージ・ヘルプの呼び出し

メッセージ・ヘルプはメッセージの原因を説明し、エラーに応答して取るべきアクションを示します。

### 手順

メッセージ・ヘルプを呼び出すには、コマンド行プロセッサを開いて、次のように入力します。

```
? XXXnnnnn
```

ここで、*XXXnnnnn* は有効なメッセージ ID を表します。

例えば、? SQL30081 は、SQL30081 メッセージのヘルプを表示します。

## 第 2 章 コマンド行 SQL ステートメントおよび XQuery ステートメントの使用

この節では、コマンド行から SQL (構造化照会言語) ステートメントを使用する方法について説明します。

これらのステートメントは、オペレーティング・システムのコマンド・プロンプトから直接実行できるもので、コマンドをアプリケーション・プログラムに書き込む場合とほぼ同じようにして、データベースの表や索引やビューに格納されている情報を定義したり処理したりするのに使用できます。情報については追加や削除や更新ができます。レポートは表の内容から生成できます。

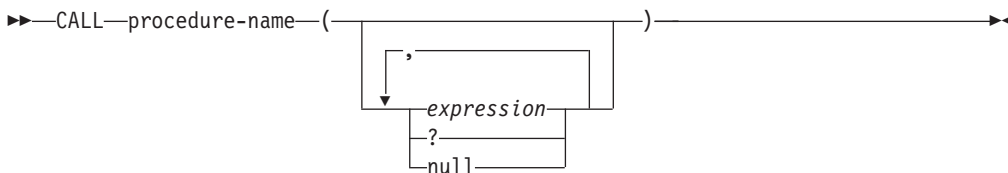
コマンド行から SQL ステートメントを使用できます。SQL を介して CLP コマンドを実行するストアード・プロシージャ (SYSPROC.ADMIN\_CMD()) を使用できます。このストアード・プロシージャの使用方法についての詳細は、「SQL 管理ルーチン」を参照してください。

CLP で XQuery ステートメントを発行するには、ステートメントに XQUERY キーワードの接頭部を付けてください。

注: XQuery ステートメントを発行するために CLP を使用する場合、セミコロン (; オプション) 以外の終了文字を使用するのが最善の選択です。名前空間宣言もセミコロンで終了するため、こうすることによって、名前空間宣言を使用するステートメントまたは照会が誤って認識されないようにします。

コマンド行プロセッサによって実行できる SQL ステートメントはすべて、27 ページの表 5 の CLP 列にリストされています。SQL ステートメントの構文は、コマンド行から実行できるものであれ、ソース・プログラムに組み込まれているものであれ、すべてが「SQL リファレンス」に記述されています。多くの場合、組み込み SQL ステートメントと CLP SQL ステートメントの構文は同じです。ただし、ホスト変数、パラメーター・マーカー、記述子名、およびステートメント名は、組み込み SQL にのみ適用できます。CALL、CLOSE、CONNECT、DECLARE CURSOR、FETCH、および OPEN の構文は、組み込み型のものとは CLP によって実行されるものとは異なります。これらのステートメントの CLP 構文を以下のセクションに示します。

### CALL



### CLOSE

## コマンド行 SQL ステートメントおよび XQuery ステートメントの使用

▶▶ CLOSE *cursor-name* ▶▶

### CONNECT

▶▶ CONNECT

TO <i>server-name</i>	lock-block	authorization
RESET	(1)	
authorization		

#### authorization:

USER *authorization-name*

USING <i>password</i>	NEW <i>password</i> CONFIRM <i>password</i>
CHANGE PASSWORD	

#### lock-block:

IN SHARE MODE	
IN EXCLUSIVE MODE	ON SINGLE NODE

#### 注:

- この形式は、暗黙接続が有効である場合にのみ有効です。

### DECLARE CURSOR

▶▶ DECLARE *cursor-name* CURSOR

WITH HOLD
-----------

▶▶ DATABASE *dbname*

USER <i>user</i> USING <i>password</i>
--

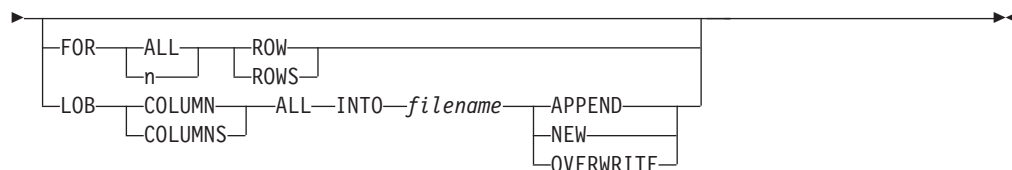
▶▶ FOR

<i>select-statement</i>	<i>xquery-statement</i>
-------------------------	-------------------------

### FETCH

▶▶ FETCH

FROM	<i>cursor-name</i>
------	--------------------



## OPEN

►► OPEN *cursor-name* ◀◀

### 注:

- CALL の発行時には、以下のことが必要です。
  - プロシージャの各 IN または INOUT パラメーターごとに式を使用することが必要です。INOUT パラメーターの場合、式は単一のリテラル値でなければなりません。INOUT XML パラメーターは NULL (NULL 可能である場合)、または XMLPARSE(DOCUMENT *string*) という形式でなければなりません。XMLPARSE の引数内の *string* はストリング・リテラルでなければならず、CURRENT IMPLICIT XMLPARSE OPTION 特殊レジスターに制限されます。式にすることはできません。
  - プロシージャの各 OUT パラメーターごとに、疑問符 (?) を使用することが必要です。
  - ストアード・プロシージャはカタログされている必要があります。カタログされていないプロシージャが呼び出されると、SQL0440N エラー・メッセージが戻ります。

以下の CLP スクリプトは、XML 列 *C1* を含む表の作成後に PROC4 と呼ばれるプロシージャを作成します。これは、IN (*PARM1*)、INOUT (*PARM2*)、および OUT (*PARM3*) の 3 つの XML パラメーターを使用し、結果セットを XML データで戻します。

```

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
  
```

コマンド行プロセッサからプロシージャ PROC4 を呼び出すには、以下の CALL ステートメントを発行します。

## コマンド行 SQL ステートメントおよび XQuery ステートメントの使用

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

- CONNECT の CLP バージョンを使用すると、ユーザーは、次のパラメーターを使用してパスワードを変更することができます。

### **NEW password**

ユーザー名に割り当てられる新規パスワードを指定します。パスワードの長さは、最大で 18 文字です。パスワードが変更されるシステムは、ユーザー認証がセットアップされた方法によって異なります。

### **CONFIRM password**

新規パスワードと同一のストリング。このパラメーターは、入力エラーを検出するために使用されます。

### **CHANGE PASSWORD**

このオプションが指定されていると、ユーザーにプロンプトが出され、現在のパスワード、新規パスワード、および新規パスワードの確認を要求します。入力時にパスワードは表示されません。

- DECLARE CURSOR ステートメント内の DATABASE 節は、それ以降のカーソルからのロード操作でカーソルが使用される場合にのみ適用されます。
- XQuery ステートメントとともに DECLARE CURSOR ステートメントを使用するには、ユーザーが XQuery ステートメントの接頭部としてキーワード XQUERY を明示する必要があります。
- FETCH がコマンド行プロセッサから出されると、10 進数および浮動小数点、各地域の 10 進数区切り文字と共に表示されます。米国、カナダ、英国の場合はピリオド (.)、他のほとんどの国地域の場合はコンマ (,) です。ただし、INSERT、UPDATE、CALL、およびその他の SQL ステートメントをコマンド行プロセッサから出して表を更新する場合は、たとえコンマを使用する国や地域であっても、10 進数区切り文字としてピリオドを使用しなければなりません。
- FETCH がコマンド行プロセッサから出されると、通常 NULL 値はハイフン (-) で表示されます。DFT\_SQLMATHWARN YES で構成されたデータベースでは、算術計算エラーとなる式は NULL 値として処理されます。そのような算術計算エラー NULL 値は、プラス (+) で表示されます。

例えば、表 t1 を次のように作成します。

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

ステートメント `select i1/i2 from t1` は、次の結果を生成します。

```
1
---
1
+
-
3 records selected
```

- 新しい LOB オプションが FETCH に追加されています。LOB 節を指定する場合は、次の行だけがフェッチされます。
  - LOB 列を含む表を照会するためにコマンド行プロセッサを介して SELECT が発行された場合、出力ではすべての列が 8 KB に切り捨てられます。

## コマンド行 SQL ステートメントおよび XQuery ステートメントの使用

- それぞれの LOB 列値は *filename.xxx* という名前のファイルにフェッチされます。この場合、*filename* は LOB 節に指定され、*xxx* は 001 から 999 までのファイル拡張子です。(001 は対応する DECLARE CURSOR ステートメントの選択リストの最初の LOB 列で、002 は、2 番目の LOB 列、以降 999 は 999 番目の列になります。) ファイルにフェッチできる LOB 列の最大数は 999 です。
  - データを含むファイルの名前は LOB 列に表示されます。
8. コマンド行プロセッサは、BLOB 列を 16 進表記で表示します。
  9. 適切な変換関数を使用できない場合には、構造化タイプ列への参照を含む SQL ステートメントを発行することはできません。
  10. SQL ステートメント、および SQL ステートメント・コンポーネントを含む CLP コマンドに関する CLP の制限 (64K) は、除去されました。
  11. SELECT、CALL、または XQuery を介して取得される XML データは、出力で 4000 バイトに切り捨てられます。

CLP を介する SQL ステートメントを使用してデータベースを照会する場合、CLP がデータを表示する仕方を変更するには、CLP バインド・ファイルを照会するデータベースに対して再バインドします。例えば、日時を ISO 形式で表示したい場合、次のようにできます。

1. CLP バインド・ファイルの名前を含むテキスト・ファイルを作成する。このファイルは、1 回の BIND コマンドで複数のファイルをバインドする場合のリスト・ファイルとして使用します。この例では、とりあえずこのファイルの名前を *clp.lst* としておきます。このファイルの中身は次のようになっています。

```
db2c1pcs.bnd +
db2c1prp.bnd +
db2c1pur.bnd +
db2c1prs.bnd +
db2c1pns.bnd
```

2. データベースに接続する。
3. 次のコマンドを発行する。

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

表 5. SQL ステートメント (DB2)

SQL ステートメント	動的 <sup>1</sup>	コマンド行 プロセッサ ー (CLP)	コール・レベル・ インターフェース <sup>3</sup> (CLI)	SQL プロシ ージャー
ALLOCATE CURSOR				X
割り当てステートメント				X
ASSOCIATE LOCATORS				X
ALTER { BUFFERPOOL, DATABASE PARTITION GROUP, NICKNAME, <sup>9</sup> 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
CASE ステートメント				X

## コマンド行 SQL ステートメントおよび XQuery ステートメントの使用

表 5. SQL ステートメント (DB2) (続き)

SQL ステートメント	動的 <sup>1</sup>	コマンド行 プロセッサ (CLP)	コール・レベル・ インターフェース <sup>3</sup> (CLI)	SQL プロシ ージャー
CLOSE		X	SQLCloseCursor(), SQLFreeStmt()	X
COMMENT ON	X	X	X	X
COMMIT	X	X	SQLEndTran(), SQLTransact()	X
コンパウンド SQL (組み込み)			X <sup>4</sup>	
コンパウンド・ステートメント				X
CONNECT (タイプ 1)		X	SQLBrowseConnect(), SQLConnect(), SQLDriverConnect()	
CONNECT (タイプ 2)		X	SQLBrowseConnect(), SQLConnect(), SQLDriverConnect()	
CREATE { ALIAS, BUFFERPOOL, DATABASE PARTITION GROUP, DISTINCT TYPE, EVENT MONITOR, FUNCTION, FUNCTION MAPPING <sup>9</sup> , GLOBAL TEMPORARY TABLE, INDEX, INDEX EXTENSION, METHOD, NICKNAME, <sup>9</sup> 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	
FOR ステートメント				X
FREE LOCATOR			X <sup>4</sup>	X
GET DIAGNOSTICS				X
GOTO ステートメント				X



## コマンド行 SQL ステートメントおよび XQuery ステートメントの使用

表 5. SQL ステートメント (DB2) (続き)

SQL ステートメント	動的 <sup>1</sup>	コマンド行 プロセッサ ー (CLP)	コール・レベル・ インターフェース <sup>3</sup> (CLI)	SQL プロシ ージャー
GRANT	X	X	X	X
IF ステートメント				X
INCLUDE <sup>2</sup>				
INSERT	X	X	X	X
ITERATE				X
LEAVE ステートメント				X
LOCK TABLE	X	X	X	X
LOOP ステートメント				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	
REPEAT ステートメント				X
RESIGNAL ステートメント				X
RETURN ステートメント				X
REVOKE	X	X	X	
ROLLBACK	X	X	SQLEndTran(), SQLTransact()	X
SAVEPOINT	X	X	X	X
select-statement	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

## コマンド行 SQL ステートメントおよび XQuery ステートメントの使用

表 5. SQL ステートメント (DB2) (続き)

SQL ステートメント	動的 <sup>1</sup>	コマンド行 プロセッサ ー (CLP)	コール・レベル・ インターフェース <sup>3</sup> (CLI)	SQL プロシ ージャー
SET SERVER OPTION <sup>9</sup>	X	X	X	X
SET transition-variable <sup>5</sup>	X	X	X	X
SIGNAL ステートメント				X
SIGNAL SQLSTATE <sup>5</sup>	X	X	X	
UPDATE	X	X	X	X
VALUES INTO				X
WHENEVER <sup>2</sup>				
WHILE ステートメント				X

### 注:

- このリストのすべてのステートメントは静的 SQL としてコーディングできますが、動的 SQL としてコーディングできるのは X になっているステートメントだけです。
- このステートメントは実行できません。
- X は、該当するステートメントが SQLExecDirect() または SQLPrepare() と SQLExecute() のどちらによっても実行できるという意味です。同等の CLI 機能がある場合は、機能名がリストされています。
- このステートメントは動的ではないものの、CLI によって SQLExecDirect() または SQLPrepare() と SQLExecute() のどちらかを呼び出すときにステートメントは指定されます。
- CREATE TRIGGER ステートメント内だけで使用できます。
- SQL DESCRIBE ステートメントで出力の記述ができますが、CLI を使用すると、入力記述も可能となります (SQLDescribeParam() 関数を使用する場合)。
- SQL FETCH ステートメントで 1 つの行を一度に 1 方向にフェッチすることができますが、CLI の SQLExtendedFetch() および SQLFetchScroll() を使用すると、配列の形でフェッチすることができます。さらに、どの方向でも、また結果セットのどの位置でもフェッチができます。
- DESCRIBE SQL** ステートメントの構文は、CLP DESCRIBE コマンドの構文と異なります。
- ステートメントは、フェデレーテッド・データベース・サーバーでのみサポートされます。
- SQL プロシージャーは、索引、表、ビューには CREATE および DROP ステートメントしか発行できません。

---

## 第 3 章 Command Line Processor Plus (CLPPlus)

Command line processor plus (CLPPlus) は、データベースに接続したり、ステートメント、スクリプト、およびコマンドを定義、編集、および実行したりするために使用できるコマンド行ユーザー・インターフェースを提供します。

CLPPlus は、コマンド行プロセッサ (CLP) が提供する機能を補います。CLPPlus には以下のフィーチャーが含まれます。

- データベースへの接続を確立するためのサポート (データベース・ユーザー ID およびパスワードを提供する場合)。
- 編集および実行用に、スクリプト、スクリプトの一部、SQL ステートメント、SQL PL ステートメント、または PL/SQL ステートメントを保管するために使用できるバッファー。バッファー内のテキストをリスト、プリント、編集することができます。また、バッファー内のテキストをバッチ・スクリプトとして実行できます。
- 包括的なプロセッサ・コマンドのセット。これを使用して、バッファーに保管できる変数およびストリングを定義できます。
- データベースおよびデータベース・オブジェクトに関する情報を取得するコマンドのセット。
- バッファーまたはバッファー出力をファイルに保管する機能。
- スクリプトおよび照会の出力をフォーマット設定するための複数のオプション。
- システム定義ルーチンの実行のサポート。
- オペレーティング・システム・コマンドの実行のサポート。
- 実行されたコマンド、ステートメント、またはスクリプトの出力を記録するオプション。

CLPPlus は、SERVER、SERVER\_ENCRYPT、および KERBEROS の各認証のみをサポートします。

---

### 既存の IBM Data Server Client における CLPPlus のインストール

デフォルトでは、CLPPlus はすべての IBM Data Server Client にインストールされるわけではありません。IBM Data Server Client インストールに CLPPlus が含まれていない場合、後でインストールと構成を行えます。

#### 始める前に

Java™ バージョン 1.5.0 以降の SDK または JRE がインストールされていることを確認してください。すべての IBM Data Server Client (IBM Data Server Runtime Client および IBM Data Server Driver Package 以外) では、IBM Data Server Client インストール・プロセスによって Java バージョン 5 の SDK が自動的にインストールされます。

### 手順

既存の IBM Data Server Client で CLPPlus をインストールおよび構成するには、以下のようにします。

1. **JAVA\_HOME** 環境変数が、ご使用のシステムの Java インストール・ディレクトリに設定されていることを確認します。
2. **PATH** 環境変数設定に、ご使用のシステムの *Java installation directory/bin* ディレクトリが含まれていることを確認します。
3. `clpplus.jar`、`jline-0.9.93.jar`、および `db2jcc.jar` の各ファイルを **CLASSPATH** 環境変数設定に追加します。CLPPlus がこうしたファイルを検出できないと、これらのファイルに関するエラー・メッセージが表示され、始動しません。

---

## CLPPlus の開始

CLPPlus コマンドを実行するには、まず CLPPlus を開始する必要があります。

### 始める前に

- 既存の IBM データ・サーバー・クライアントには、CLPPlus がインストールおよびセットアップされていない可能性があります。CLPPlus は、他のすべての DB2 V9.7 または V10.1 製品と共に使用できるようにセットアップされます。
- **CLPPLUS** コマンドのトピックをお読みください。

### 手順

CLPPlus を開始するには、次のようにします。

- Windows、Linux、または UNIX オペレーティング・システム上で、**CLPPLUS** コマンドを使用します。
  1. コマンド行プロセッサ (CLP) または DB2 コマンド・ウィンドウで、パスワードを指定せずに **CLPPLUS** コマンドを実行します。例えば、ユーザー ID `db2admin` とポート番号 `50000` を使用し、CLPPlus を開始してローカル・ホスト上の `SAMPLE` という名前のデータベースへの接続を試みるには、次のコマンドを発行します。

```
clpplus db2admin@localhost:50000/sample
```

また、`login.sql` というファイルで接続情報を指定することもできます。変数を定義して、CLPPlus を開始するために使用するコマンドでその変数を指定できます。例えば、`login.sql` に `define connStr = localhost:50000/sample` が含まれている場合、コマンド `clpplus db2admin@connStr` はこの手順の最初の例と同等です。

2. パスワードを入力します。ステップ 1 のコマンドを例にとると、`db2admin` に対応する有効なパスワードが入力された場合は、次のように表示されます。

```
C:¥DB2¥9.7¥db2 > clpplus db2admin@localhost:50000/sample
Connected to SAMPLE v 9.7 AS db2admin
SQL>
```

- Windows オペレーティング・システムの場合は、メニュー・オプションを使用します。

1. 「スタート」 > 「IBM DB2」 > 「CLPPlus」をクリックします。

2. ユーザー ID および接続情報を指定します。
- Linux オペレーティング・システムの場合は、メニュー・オプションを使用します。
    1. 「メインメニュー」 > 「IBM DB2」 > 「Command Line Processor Plus」をクリックします。
    2. ユーザー ID および接続情報を指定します。

### タスクの結果

CLPPlus プロンプト (SQL>) が使用可能になり、指定したデータベースに接続されます。

### 次のタスク

これで、CLPPlus コマンドと関連フィーチャーを使用できるようになりました。CLPPlus プロンプトでコマンドを指定します。

CLPPlus セッションを終了するには、**EXIT** または **QUIT** CLPPlus コマンドを発行します。

## CLPPLUS コマンド

CLPPlus (command line processor plus) を開始します。 CLPPlus を開始した後は、CLPPlus コマンドの発行、データベースへの接続、SQL ステートメントとデータベース・コマンドの定義および実行、SQL ステートメントとコマンドを含んだスクリプトの実行が可能です。

### 呼び出し

このコマンドはオペレーティング・システムのコマンド・プロンプトから実行する必要があります。

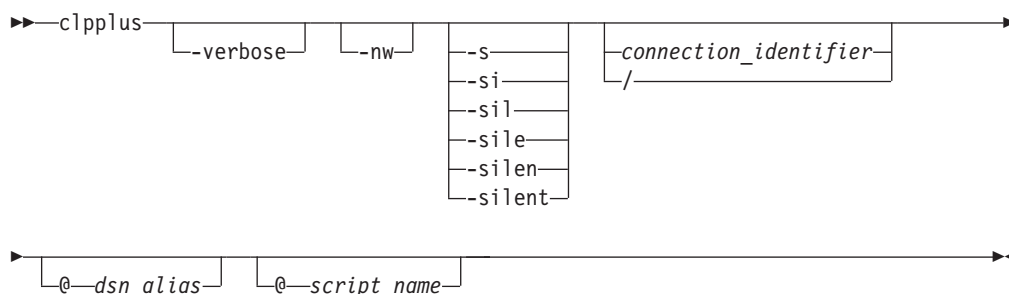
### 許可

なし

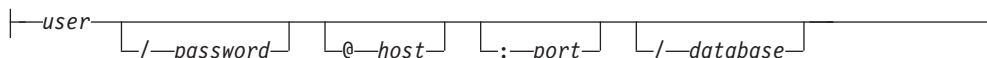
### 必要な接続

なし

### コマンド構文



#### connection\_identifier:



### コマンド・パラメーター

#### -verbose

verbose をオンに設定します。 verbose がオンに設定されている場合、すべての CLPPlus メッセージがコンソールに出力されます。

#### -nw

CLPPlus セッションが現在のコマンド行ウィンドウで開始されるように指定します。

#### -s | -si | -sil | -sile | -silen | -silent

現行の CLPPlus セッションでの、バージョン情報、著作権情報、プロンプト・メッセージ、コマンド・エコーおよび接続情報の表示を抑制します。サイレント・モード中、デフォルトでは、ECHO は OFF に設定されます。ECHO を ON に設定しようとするいかなる試みも無視されます。

/ データベースへの接続に、現在のオペレーティング・システムのログイン・ユーザー ID を使用するように指定します。

*user*

データベースへ接続するユーザー ID を指定します。

*password*

ユーザー ID に対応するパスワードを指定します。

*hostname*

データベースが稼働するコンピューター名を指定します。例えば、ascender という名前のコンピューターの場合は、@ascender と指定します。

*port*

データベース・サーバーがインストールされたコンピューター上の接続受信ポート番号を指定します。デフォルトは 50000 です。

*database*

接続先のデータベース名を指定します。デフォルトは SAMPLE です。

*dsn\_alias*

別名 *dsn\_alias* の DSN の db2dsdriver.cfg ファイルから、データベース接続情報を取得するように指定します。指定した *dsn\_alias* が db2dsdriver.cfg ファイルにないと、*dsn\_alias* というストリングをデータベース名として使用し、他のすべての接続パラメーターは対話的に取得します。

*script-name*

スクリプト・ファイルの名前を指定します。ファイルが現行作業ディレクトリにない場合は、ファイルのロケーションの絶対パスも含める必要があります。スクリプト・ファイルには、CLPPlus 開始後に自動的に実行されてデータベース接続を確立する SQL ステートメントを含めることができます。

## 例

以下の例は、CLPPLUS コマンドで CLPPlus プロセッサーを開始し、オプションでデータベース接続を行う方法を示しています。

次のコマンドは、CLPPlus をクライアント・モードで開始します。

```
clpplus
```

データベース接続は試みられません。このコマンドの発行後、データベースに接続するには、CONNECT CLPPlus コマンドを使用します。

次のコマンドは CLPPlus を開始し、ローカル・ホスト上の SAMPLE という名前のデータベースへの接続を試みます。ユーザー ID db2admin、ポート番号 50000 を使用します。

```
clpplus db2admin@localhost:50000/sample
```

次のコマンドは CLPPlus を開始し、ユーザー ID db2admin に対応するパスワードを求めるプロンプトを出します。パスワードが有効であれば、CLPPlus は SAMPLE (デフォルトのデータベース名) という名前のデータベースへの接続を試みます。

```
clpplus db2admin
```

## CLPPLUS コマンド

次のコマンドは CLPPlus を開始し、SAMPLE という名前のデータベースへの接続を試みます。ユーザー ID db2admin、パスワード mypassw0rd を使用します。ユーザー ID とパスワードが有効であれば、データベース接続が確立されます。パスワードを指定することの欠点は、パスワードが画面に表示されることです。

```
clpplus db2admin/mypassw0rd
```

次のコマンドは CLPPlus を開始し、リモート・マシン ascender 上の SAMPLE という名前のデータベースへの接続を試みます。ポート 50000、ユーザー ID db2admin、パスワード mypassw0rd を使用します。これらの値が有効であれば、データベース接続が確立されます。

```
clpplus db2admin/mypassw0rd@ascender:50000/sample
```

次のコマンドは CLPPlus を開始し、ローカル・コンピューター上の DB という名前のデータベースへの接続を試みます。ユーザー ID db2admin、パスワード mypassw0rd、およびデフォルト・ポート番号 (50000) を使用します。

```
clpplus db2admin/mypassw0rd@localhost/db
```

以下のコマンドは、CLPPlus を開始し、最初に db2dsdriver.cfg ファイルを検索して、データベースへの接続を試みます。このファイルが見つかると、ホスト、ポート、データベースの各値に関して default\_dsn が読み取られます。接続の試行時には、現在のログオン ID が使用されます。db2dsdriver.cfg ファイルが見つからない場合は、必須パラメーターすべてが対話的に要求されます。

```
clpplus /
```

以下のコマンドは、CLPPlus を開始し、最初に db2dsdriver.cfg ファイルを検索して、データベースへの接続を試みます。このファイルが見つかると、ホスト、ポート、データベースの各値に関して default\_dsn が読み取られます。接続の試行時には、db2admin ID が使用されます。db2dsdriver.cfg ファイルが見つからない場合は、必須パラメーターすべてが対話的に要求されます。

```
clpplus db2admin
```

以下のコマンドは、CLPPlus を開始し、db2dsdriver.cfg ファイルの data\_dsn 別名からパラメーターを取り出して、データベースへの接続を試行します。ユーザー ID が指定されていないので、現在のログオン・ユーザー ID が使用されます。読み取れないパラメーターに関しては、対話的に要求されます。

```
clpplus /@data_dsn
```

以下のコマンドは、CLPPlus を開始し、db2dsdriver.cfg ファイルの data\_dsn 別名からパラメーターを取り出して、データベースへの接続を試行します。接続には、db2admin ユーザー ID が使用されます。読み取れないパラメーターに関しては、対話的に要求されます。

```
clpplus db2admin@data_dsn
```

---

## CLPPlus コンソール・タイプ

V9.7 フィックスパック 2 の時点で、CLPPlus には、従来のコマンド行コンソールに新しいウィンドウ・モードが追加されました。



## ウィンドウ・モード

V9.7 フィックスパック 2 で導入された新しいコンソールは、ウィンドウ・スタイルのコンソールです。 **CLPPLUS** コマンドを使って **CLPPlus** が開始されると、デフォルトでウィンドウ・モードのコンソールが作成されます。この新しいコンソール・ウィンドウには、改善されたコマンド編集機能が備えられています。そのため、対話モードで **CLPPlus** を使用する際には、このウィンドウ・モードのコンソールの選択が望ましい可能性があります。

## 非ウィンドウ・モード

**CLPPlus** には、非ウィンドウ・モードまたは従来のコマンド行コンソールを使用するというオプションも残されています。 **CLPPLUS** コマンドを使って **CLPPlus** が開始され、**-nw** オプションが指定される場合、現在のコマンド行環境が使用されます。つまり、新しいウィンドウは作成されません。スクリプト・ファイルを呼び出しており、作成されたコンソール・ウィンドウを必要としない場合には、この選択が望ましい可能性があります。

## ウィンドウ・モードと UTF-8 文字サポート

V9.7 フィックスパック 3 以降、新しいウィンドウ・モード・コンソールは UTF-8 文字のみをサポートします。

新しいウィンドウ・モード・コンソールが導入されるまでは、CLPPlus はファイル・システムまたはデータベースから、コンソール内の UTF-8 文字を正確に読み書きできませんでした。画面上に表示される UTF-8 文字は不揃いだったり、不正確だったりしました。キーボードから UTF-8 文字を入力できませんでした。

新しいウィンドウ・モード・コンソールは、コンソールの外観を保ちながら、UTF-8 文字に関するこうした問題すべてに対応しています。CLPPlus はファイル・システムとデータベースから UTF-8 文字を読み書きできるようになり、キーボードやユーザー・インターフェースから問題なく対話できます。また、新しいウィンドウ・モード・コンソールでは、列の位置合わせに関する問題も解決されています。よって、表出力が正しく表示されるようになりました。

### 注:

このサポートが有効なのはウィンドウ・モードに限定され、このモードは、V9.7 フィックスパック 2 以降のデフォルト・コンソールとなっています。ウィンドウなしモードの場合、UTF-8 文字のサポートは依然として限定的です。ウィンドウなしモードの場合、キーボードから入力できない UTF-8 文字もあります。UTF-8 文字が結果セットに含まれていると、結果セット列の位置合わせに狂いが生じる可能性があります。

## CLPPlus ウィンドウ・モードのフォントの設定

バージョン 10.1 フィックスパック 2 以降、CLPPlus ウィンドウ・モードでさまざまなフォントを設定できるようになりました。設定にはフォントの名前、スタイル、およびサイズが含まれます。

### このタスクについて

#### 制約事項

このタスクは、バージョン 10.1 フィックスパック 2 レベル以降の CLPPlus クライアントに当てはまります。

#### 手順

CLPPlus ウィンドウ・モードのフォントの名前、スタイル、およびサイズを設定するには、次のようにします。

1. **CLPPLUS** CLPPlus コマンドを使用して、CLPPlus を開始します。デフォルトでは、CLPPlus はウィンドウ・モードで開始されます。このコマンドについて詳しくは、『関連参照』を参照してください。
2. **SET CLPPlus** コマンドで **FONT** キーワードを使用して、**SET FONT name, style, size** のようにします。SET コマンドについて詳しくは、『関連参照』を参照してください。

### 例

この例では、CLPPlus ウィンドウ・モードで表示されるフォントを、serif、太字のタイプセット、32 ポイントのサイズに設定します。

```
SET FONT serif,1,32
```

この例では、CLPPlus ウィンドウ・モードで表示されるフォントを、SansSerif、プレーンのタイプセット、48 ポイントのサイズに設定します。

```
SET FONT "SansSerif", 0, 48
```

この例では、CLPPlus ウィンドウ・モードで表示されるフォントを、lucinda console、太字のタイプセット、35 ポイントのサイズに設定します。

```
SET FONT "lucida console",1, 35
```

## CLPPlus ウィンドウ・モードのフォントの色の設定

バージョン 10.1 フィックスパック 2 以降、CLPPlus ウィンドウ・モードでさまざまなフォントの色を設定できるようになりました。

### このタスクについて

制約事項

このタスクは、バージョン 10.1 フィックスパック 2 レベル以降の CLPPlus クライアントに当てはまります。

### 手順

CLPPlus ウィンドウ・モードのフォントの色を設定するには、次のようにします。

1. **CLPPLUS** CLPPlus コマンドを使用して、CLPPlus を開始します。デフォルトでは、CLPPlus はウィンドウ・モードで開始されます。このコマンドについては、『[関連参照](#)』を参照してください。
2. **SET CLPPlus** コマンドで **COLOR** キーワードを使用して、**SET COLOR color|<r,g,b>** のようにします。SET コマンドについては、『[関連参照](#)』を参照してください。

### 例

この例では、CLPPlus ウィンドウ・モードで表示されるフォントの色を赤に設定します。

```
SET COLOR RED
```

この例では、CLPPlus ウィンドウ・モードで表示されるフォントの色を白に設定します。

```
SET COLOR 255,255,255
```

## CLPPlus ウィンドウ・モードの背景色の設定

バージョン 10.1 フィックスパック 2 以降、CLPPlus ウィンドウ・モードでさまざまな背景色を設定できるようになりました。

### このタスクについて

#### 制約事項

このタスクは、バージョン 10.1 フィックスパック 2 レベル以降の CLPPlus クライアントに当てはまります。

#### 手順

CLPPlus ウィンドウ・モードの背景色を設定するには、次のようにします。

1. **CLPPLUS** CLPPlus コマンドを使用して、CLPPlus を開始します。デフォルトでは、CLPPlus はウィンドウ・モードで開始されます。このコマンドについて詳しくは、『関連参照』を参照してください。
2. **SET CLPPlus** コマンドで **BGCOLOR** キーワードを使用して、**SET BGCOLOR color|<r,g,b>** のようにします。SET コマンドについて詳しくは、『関連参照』を参照してください。

#### 例

この例では、CLPPlus ウィンドウ・モードで表示される背景色をシアンに設定します。

```
SET BGCOLOR cyan
```

この例では、CLPPlus ウィンドウ・モードで表示される背景色を白に設定します。

```
SET BGCOLOR 255,255,255
```

---

## CLPPlus での DSN 別名

CLPPlus は db2dsdriver.cfg 構成ファイルで定義されている DSN 別名への接続をサポートしています。これがサポートされる前は、CLPPlus インターフェースでは対話式の接続だけが許可されていました。

ds ドライバーの構成ファイル db2dsdriver.cfg は、DSN 別名とそのプロパティのリストが含まれている XML ファイルです。これを使用して、接続の詳細を 1 箇所に保管します。接続が試行されるたびにすべての接続の詳細を対話式に要求する代わりに、CLPPlus はその情報を使用してデータ・ソースに自動的に接続します。

db2dsdriver.cfg ファイルを指すように **DB2DSDRIVER\_CFG\_PATH** 環境変数を設定することができます。DB2DSDRIVER\_CFG\_PATH 環境変数について詳しくは、『関連参照』にリストされている『その他の変数』トピックを参照してください。

DB2DSDRIVER\_CFG\_PATH が設定されていない場合は、CLPPlus がデフォルト・ディレクトリーのロケーションで db2dsdriver.cfg ファイルを検索します。db2dsdriver.cfg ファイルのデフォルト・ロケーションについて詳しくは、『関連概念』を参照してください。

構成ファイルが見つかり、読み取ることができる場合、CLPPlus はそれ以降の接続試行でそれを使います。

接続を試行するユーザーにデータベース名が求められます。そのデータベース名は DSN 別名として扱われ、それを求めて構成ファイル内を検索されます。その DSN

別名が見つかる場合、接続属性が読み取られてパスワードが要求され、接続を完了します。DSN 別名が見つからない場合、ホスト名、ポート番号、ユーザー名、およびパスワードが対話式に要求され、元のデータベース名と一緒に、集められたすべての情報を使って、接続が試行されます。

V9.7 フィックスパック 6 の時点では、データベース接続中に ExtendedIndicators プロパティを有効にするか無効にするかを指定できます。デフォルトでは、このプロパティは無効です。詳しくは、関連する解説書を参照してください。

## 例

以下の db2dsdriver.cfg ファイルの内容を検討してください。

```
<configuration>
<dsncollection>
  <dsn alias="S" name="SAMPLE" host="9.121.221.159" port="50000">
    <parameter name="ExtendedIndicators" value="0"/>
  </dsn>
</dsncollection>
<databases>
  <database name="SAMPLE" host="9.121.221.159" port="50000">
    <parameter name="UserID" value="john"/>
  </database>
</databases>
</configuration>
```

この例は、db2dsdriver.cfg ファイルの内容を使って確立される接続を示しています。

まず、ユーザーは `DB2DSDRIVER_CFG_PATH` を設定します。

```
C:\>set DB2DSDRIVER_CFG_PATH="C:\john\clpplus\db2dsdriver.cfg"
```

次に、ユーザーは CLPPlus セッションを開始し、DSN 別名「S」への接続を試行します。

```
C:\>clpplus
CLPPlus: Version 1.1
Copyright (c) 2009, IBM CORPORATION. All rights reserved.
```

```
SQL> connect
```

```
Enter DATABASE NAME [SAMPLE]: S
Enter ID [john] :
Enter Password: *****
```

```
Database Connection Information
-----
Hostname = 9.121.221.159
Database server = DB2/NT SQL09071
SQL authorization ID = john
Local database alias = S
Port = 50000
```

```
SQL>
```

次の例は、入力されたデータベース名が db2dsdriver.cfg ファイル中に別名として見つからないときの接続を示しています。

```
SQL> connect
```

```
Enter DATABASE NAME [SAMPLE]:
Enter HOSTNAME [localhost]:
```

## CLPPlus での DSN 別名

```
Enter PORT [50000]:
Enter ID : john
Enter Password: *****

Database Connection Information
-----
Hostname = 9.121.221.159
Database server = DB2/NT SQL09071
SQL authorization ID = john
Local database alias = SAMPLE
Port = 50000

SQL>
```

「sample」は構成ファイルで DSN 別名として見つからないため、残りの値は CLPPlus によって対話式に要求され、その後、接続が試行されます。

---

## CLPPlus での Kerberos 認証

V9.7 フィックスパック 2 以降、CLPPlus は Kerberos 認証を使用した、db2dsdriver.cfg 構成ファイルで定義されている DSN 別名への接続をサポートするようになりました。これがサポートされる前は、CLPPlus インターフェースで対話式の接続だけが許可されていました。

db2dsdriver.cfg 構成ファイルは、DSN 別名とそのプロパティのリストを含む XML ファイルです。当該の DSN 別名の **Authentication** プロパティが読み取られる場合はそれが読み取られ、値が *kerberos* の場合は Kerberos 認証メカニズムが試行されます。CLPPlus での DSN 別名への接続について詳しくは、40 ページの『CLPPlus での DSN 別名』を参照してください。

CLPPlus は単独では Kerberos TGT チケットを要求しません。ユーザーが他のアプリケーションまたはツールを使って既に取得したチケットを使用します。

### 例

以下の db2dsdriver.cfg ファイルの内容をよく確認してください。

```
<configuration>
  <dsncollection>
    <dsn alias="S" name="SAMPLE" host="9.121.221.159" port="50000">
    </dsn>
  </dsncollection>
  <databases>
    <database name="SAMPLE" host="9.121.221.159" port="50000">
      <parameter name="UserID" value="john"/>
    </database>
  </databases>
  <parameters>
    <parameter name="Authentication" value="KERBEROS"/>
  </parameters>
</configuration>
```

この例は、**Authentication** パラメーター値の読み取りを含む db2dsdriver.cfg ファイルの内容を使用して確立される接続を示しています。

ユーザーは CLPPlus セッションを開始し、DSN 別名「S」への接続を試行します。

```
C:\>clpplus
CLPPlus: Version 1.1
Copyright (c) 2009, IBM CORPORATION. All rights reserved.
```

```

SQL> connect

Enter DATABASE NAME [SAMPLE]: S
Enter ID [john] :
Enter Password: *****

Database Connection Information
-----
Hostname = 9.121.221.159
Database server = DB2/NT  SQL09071
SQL authorization ID = john
Local database alias = S
Port = 50000

SQL>

```

---

## CLPPlus での SERVER\_ENCRYPT 認証

V9.7 フィックスパック 5 以降、CLPPlus は SERVER\_ENCRYPT 認証を使用した、db2dsdriver.cfg 構成ファイルで定義されている DSN 別名への接続をサポートするようになりました。

db2dsdriver.cfg 構成ファイルは、DSN 別名とそのプロパティのリストを含む XML ファイルです。当該の DSN 別名の **Authentication** プロパティが読み取られる場合はそれが読み取られ、値が **SERVER\_ENCRYPT** の場合は SERVER\_ENCRYPT 認証メカニズムが使用されます。CLPPlus での DSN 別名への接続について詳しくは、関連する解説書を参照してください。

### 例

この例は、**Authentication** パラメーター値 **SERVER\_ENCRYPT** を含む db2dsdriver.cfg ファイルの内容による接続を示しています。

このサンプル db2dsdriver.cfg ファイルの内容をよく確認してください。

```

<configuration>
  <dsncollection>
    <dsn alias="S" name="SAMPLE" host="9.121.221.159" port="50000">
      <parameter name="Authentication" value="SERVER_ENCRYPT"/>
    </dsn>
  </dsncollection>
  <databases>
    <database name="SAMPLE" host="9.121.221.159" port="50000">
      <parameter name="UserID" value="john"/>
    </database>
  </databases>
</configuration>

```

ユーザーは CLPPlus セッションを開始し、DSN 別名「S」に接続します。

```

C:¥>clpplus -nw
CLPPlus: Version 1.5
Copyright (c) 2009, 2011, IBM CORPORATION. All rights reserved.

SQL> connect
Enter DATABASE NAME [SAMPLE]: S
Enter ID [john]:
Enter password: *****

```

```

Database Connection Information :
-----

```

## CLPPlus での SERVER\_ENCRYPT 認証

```
Hostname = 9.121.221.159  
Database server = DB2/NT SQL09075  
SQL authorization ID = john  
Local database alias = S  
Port = 50000
```



## CLPPlus での LDAP サポート

V9.7 フィックスパック 5 以降、CLPPlus 接続は構成済み LDAP ディレクトリー・サーバーでの DSN 別名検索をサポートするようになりました。

### 説明

V9.7 フィックスパック 5 より前では、CLPPlus は接続の試行の途中で、提供される db2dsdriver.cfg ファイルで DSN 別名を検索していました。db2dsdriver.cfg ファイルで一致が検出されると、ホスト名やポート番号といった情報がこのファイルから取得され、接続の試行で使用されました。DSN 別名が db2dsdriver.cfg ファイルで見つからない場合、DSN 別名は対話式の接続試行中にデータベース名として扱われました。この接続試行では、ホスト名やポート番号などの情報をユーザーが提供する必要がありました。

V9.7 フィックスパック 5 以降、CLPPlus では DSN 別名の参照先を構成済みの LDAP ディレクトリー・サーバーにまで拡張しました。ユーザーが指定した DSN 別名が db2dsdriver.cfg ファイルで見つからない場合、CLPPlus はユーザーが構成ファイルで指定した URL を使用して LDAP ディレクトリー・サーバーへの接続を試みます。CLPPlus が構成済み LDAP ディレクトリー・サーバーへの接続に成功すると、このサーバー上で DSN 別名が参照されます。LDAP ディレクトリー・サーバーで一致が検出されると、ホスト名、ポート番号、ユーザー ID、およびパスワードといった必須の接続詳細を使用して CLPPlus 接続が確立されます。LDAP ディレクトリー・サーバーで一致が検出されない場合、または LDAP ディレクトリー・サーバーへの接続に失敗した場合は、以前に行われていたように、対話式の接続で DSN 別名がデータベース名として扱われます。

この新しいサポートを使用可能にするには、db2dsdriver.cfg ファイルで **<ldapsrver>** タグを使用して、LDAP ディレクトリー・サーバーの URL を指定します。db2dsdriver.cfg ファイルでは、LDAP サーバー項目が 1 つのみ許可されます。**<ldapsrver>** タグ内の **UserID** フィールドおよび **Password** フィールドの指定は任意です。ユーザー ID およびパスワードの情報は実行時に入力することができます。ユーザー ID およびパスワードの情報は、CLPPlus セッションの間キャッシュに入れられます。

db2dsdriver.cfg ファイルで **UserID** パラメーターを「\*anonymous」に設定すると、LDAP サーバーへの匿名接続が試行され、ユーザー ID およびパスワードの情報は渡されません。パスワードの入力を求めるプロンプトは出されません。また、db2dsdriver.cfg ファイルで **Password** パラメーターを設定している場合、そのパラメーターは無視されます。

### 重要:

**UserID** パラメーターを「\*anonymous」に設定する場合、匿名接続をサポートするように LDAP サーバーを構成する必要があります。

### 例

以下のサンプルの db2dsdriver.cfg の内容をよく確認してください。

```
<configuration>
<dsncollection>
  <dsn alias="alias1" name="name1" host="server1.net1.com" port="50001"/>
```

## CLPPlus での LDAP サポート

```
</dsncollection>

<databases>

  <database name="name1" host="server1.net1.com" port="50001">
    <parameter name="CurrentSchema" value="OWNER1"/>
    <wlb>
      <parameter name="enableWLB" value="true"/>
      <parameter name="maxTransports" value="50"/>
    </wlb>
    <acr>
      <parameter name="enableACR" value="true"/>
    </acr>
  </database>

</databases>

<ldapsrv>
  <parameter name="EnableLDAP" value="YES"/>
  <parameter name="LDAPServerHost" value="ipv6lab7.torolab.ibm.com"/>
  <parameter name="LDAPServerPort" value="389"/>
  <parameter name="UserID" value="root"/>
  <parameter name="Password" value="itdsv23"/>
  <parameter name="BaseDN" value="O=IBM"/>
  <parameter name="Authentication" value="simple"/>
</ldapsrv>
</configuration>
```

サンプル db2dsdriver.cfg を使用した次の例に示すように、別名 DBLDAP1 への接続が試行されます。このファイルでは別名が見つかりません。ただし、LDAP ディレクトリー・サーバー ipv6lab7.torolab.ibm がこのファイルで指定されています。CLPPlus はこのサーバーに正常に接続します。LDAP ディレクトリー・サーバーで DBLDAP1 の検出に成功し、ホスト、ポート、およびデータベースの情報が取得されます。CLPPlus の接続が試行されます。

```
SQL> connect
Enter DATABASE NAME [SAMPLE]: DBLDAP1
Enter ID : db2admin
Enter password: *****
```

```
Database Connection Information :
-----
Hostname = winguest.torolab.ibm.com
Database server = DB2/NT SQL09075
SQL authorization ID = db2admin
Local database alias = DBLDAP1
Port = 50000
```

次の例の状況は、前述の例と同じですが、以下の出力では CLPPlus を **VERBOSE** モードで実行した結果が示されます。

```
SQL> connect
DB250001I: CLPPlus has successfully read the configuration file named
'C:\Documents and Settings\All Users\Application data\IBM\DB2\COPY1\cfg\db2dsdriver.cfg'.

Enter DATABASE NAME [SAMPLE]: DBLDAP1

DB250014I: DSN alias 'DBLDAP1' is not found in the configuration file named
'C:\Documents and Settings\All Users\Application Data\IBM\DB2\COPY1\cfg\db2dsdriver.cfg'.

DB250015I: CLPPlus successfully established a connection with LDAP directory
server 'ipv6lab7.torolab.ibm.com:389'

Enter ID : db2admin
Enter password: *****

Database Connection Information :
```

```
-----
Hostname = winguest.torolab.ibm.com
Database server = DB2/NT SQL09075
SQL authorization ID = db2admin
Local database alias = DBLDAP1
Port = 50000
```

サンプルの db2dsdriver.cfg を再び使用した次の例では、別名 DBLDAP2 への接続が試行されます。CLPPlus は **VERBOSE** モードで実行されています。DBLDAP2 は、db2dsdriver.cfg ファイルおよび指定の LDAP ディレクトリー・サーバーのいずれでも見つかりません。続けて、対話式の CLPPlus 接続が試行されます。

```
SQL> connect
DB250001I: CLPPlus has successfully read the configuration file named
'C:\Documents and Settings\All Users\Application data\IBM\DB2\DB2COPY1\cfg\db2dsdriver.cfg'.

Enter DATABASE NAME [SAMPLE]: DBLDAP2

DB250014I: DSN alias 'DBLDAP2' is not found in the configuration file named
'C:\Documents and Settings\All Users\Application Data\IBM\DB2\DB2COPY1\cfg\db2dsdriver.cfg'.

DB250015I: CLPPlus successfully established a connection with LDAP directory server 'ipv6lab7.torolab.ibm.com:389'

DB250016E: DSN alias 'DBLDAP2' was not found in LDAP directory server 'ipv6lab7.torolab.ibm.com:389'.
'DBLDAP2' is used as the database name in the subsequent interactive CLPPlus connect attempt.

Enter HOSTNAME [localhost]: 9.128.34.89
Enter PORT [50000]: 50003
Enter ID: db2admin
Enter password:*****

Database Connection Information :
-----
Hostname = 9.128.34.89
Database server = DB2/NT SQL09075
SQL authorization ID = db2admin
Local database alias = DBLDAP2
Port = 50003
```

次の例の状況も、前述の例と同じですが、ここでは LDAP ディレクトリー・サーバーへの接続を確立できません。CLPPlus は **VERBOSE** モードで実行されています。

```
SQL> connect
DB250001I: CLPPlus has successfully read the configuration file named
'C:\Documents and Settings\All Users\Application data\IBM\DB2\DB2COPY1\cfg\db2dsdriver.cfg'.

Enter DATABASE NAME [SAMPLE]: DBLDAP2

DB250014I: DSN alias 'DBLDAP2' is not found in the configuration file named
'C:\Documents and Settings\All Users\Application Data\IBM\DB2\DB2COPY1\cfg\db2dsdriver.cfg'.

DB250017E: CLPPlus failed to establish a connection with LDAP
directory server 'ipv6lab7.torolab.ibm.com:389'. 'DBLDAP2' is
used as the database name in an interactive connect attempt.

Enter HOSTNAME [localhost]: 9.128.34.89
Enter PORT [50000]: 50003
Enter ID: db2admin
Enter password:*****

Database Connection Information :
-----
Hostname = 9.128.34.89
Database server = DB2/NT SQL09075
SQL authorization ID = db2admin
Local database alias = DBLDAP2
Port = 50003
```

次の変更された db2dsdriver.cfg ファイルの内容をよく確認してください。このファイルには、LDAP ディレクトリー・サーバー構成の **UserID** パラメーターおよび **Password** パラメーターが含まれていません。LDAP ディレクトリー・サーバーは、ipv6lab7.torolab.ibm と指定されています。

## CLPPlus での LDAP サポート

```
<configuration>
  <dsncollection>
    <dsn alias="alias1" name="name1" host="server1.net1.com" port="50001"/>
  </dsncollection>

  <databases>

    <database name="name1" host="server1.net1.com" port="50001">
      <parameter name="CurrentSchema" value="OWNER1"/>
      <wlb>
        <parameter name="enableWLB" value="true"/>
        <parameter name="maxTransports" value="50"/>
      </wlb>
      <acr>
        <parameter name="enableACR" value="true"/>
      </acr>
    </database>

  </databases>

  <ldapservers>
    <parameter name="EnableLDAP" value="YES"/>
    <parameter name="LDAPServerHost" value="ipv6lab7.torolab.ibm.com"/>
    <parameter name="LDAPServerPort" value="389"/>
    <parameter name="BaseDN" value="O=IBM"/>
    <parameter name="Authentication" value="simple"/>
  </ldapservers>
</configuration>
```

更新された db2dsdriver.cfg ファイルを使用して、別名 SAMPLE32 への接続が試行されます。このファイルでは別名が見つかりません。次の例に示すように、LDAP ディレクトリー・サーバーに対するユーザー ID とパスワードが対話式に入力されると、CLPPlus は ipv6lab7.torolab.ibm LDAP ディレクトリー・サーバーに接続します。LDAP ディレクトリー・サーバーで SAMPLE32 の検出に成功し、ホスト、ポート、およびデータベースの情報が取得されます。CLPPlus の接続が試行されます。この例では、CLPPlus は **VERBOSE** モードで実行されていません。

```
C:\Documents and Settings>clpplus /@SAMPLE32
```

```
CLPPlus: Version 1.4
Copyright (c) 2009, 2011, IBM CORPORATION. All rights reserved.
```

```
Connecting to LDAP server '9.234.67.89:389'.
Enter LDAP server user ID: root
Enter LDAP server password: *****
```

```
Enter password: *****
```

```
Database Connection Information :
```

```
-----
Hostname = 9.128.32.149
Database server = DB2/NT SQL09075
SQL authorization ID = db2admin
Local database alias = SAMPLE32
Port = 50002
```

---

## CLPPlus でのスクリプト・ファイルの実行

CLPPlus では、さまざまな方法でスクリプト・ファイルを実行できます。データベース・コマンドと SQL コマンドを含んだスクリプト・ファイルの名前を、**CLPLUS** コマンドのトークンとして指定できます。 **START** CLPPlus コマンドを使用してスクリプト・ファイルを実行できます。 **GET** CLPPlus コマンドを使用してスクリプトの

内容を CLPPlus SQL バッファにコピーした後、**RUN CLPPlus** コマンドを発行することによってスクリプトを実行することもできます。

## このタスクについて

このタスクでは、**CLPPLUS** コマンドでスクリプトを実行する方法を示します。他の方法については、関連リンクを参照してください。

## 手順

**CLPPLUS** コマンドでスクリプトを実行するには、以下のようにします。

スクリプト名を指定して **CLPPLUS** コマンドを実行します。例えば、`dept_query.sql` という名前の次のようなスクリプト・ファイルとします。

```
SET PAGESIZE 9999
SET ECHO ON
SELECT * FROM DEPT;
EXIT
```

ユーザー名 **db2user** とパスワード **passwd** を使用して、ポート 50000 でデフォルト **SAMPLE** データベースに対して `dept_query.sql` スクリプトを実行するには、次のコマンドを発行します。

```
c1pp1us db2user/passw0rd @dept_query
```

`dept_query.sql` スクリプト・ファイルは、ユーザーがデータベースに接続されてから実行されます。スクリプトが実行されると、コマンドの **SET PAGESIZE** と **SET ECHO ON**、およびステートメント **SELECT \* FROM** が発行されます。

このスクリプトの出力は以下のとおりです。 **ECHO ON** は、スクリプト・ファイル内で発行されたステートメントを表示します。DEPT の値はページ限度の 9999 まで表示されます。

```
C:¥DB2¥9.7¥db2 > c1pp1us db2user/passw0rd @dept_query.sql
Connected to DB2 v 9.7 (localhost:5444/db2samp1) AS db2user
```

```
SQL>
SELECT * FROM dept;
```

DEPT	NODNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
4	OPERATIONS	BOSTON

```
SQL >
```

```
EXIT
```

---

### CLPPlus コマンドのスキップと割り込み

V9.7 フィックスパック 3 以降、CLPPlus では、コマンド実行やスクリプト実行をスキップしたり割り込んだりする操作ができます。

CLPPlus のこれまでのリリースでは、コマンドとスクリプトの実行に関するスキップや割り込みの操作に対するサポートはありませんでした。V9.7 フィックスパック 3 では、この要件に対応しています。

Ctrl+C キー・ストロークを使用して、CLPPlus を実行するコマンドやスクリプトに対して割り込み操作を行うことが可能です。この割り込み操作は、長時間実行されている照会やスクリプトが検出され、CLPPlus インターフェースに制御を戻す必要が生じる場合に役立ちます。

また、ENTER キーを続けて 2 回押すと、次の SQL> プロンプトにスキップできます。このスキップ操作は、間違ったコマンドを入力し、それをキャンセルする場合に役立ちます。間違ったコマンドはバッファー内に残るので、編集または検討用の任意の CLPPlus コマンドを使用して編集が可能です。

#### 例

以下の例は、コマンドのスキップを示しています。

```
SQL> select *
      2 from employee          <- first carriage return
      3                       <- 2 回目のキャリッジ・リターン
SQL>                          <- 次の SQL プロンプト
```

---

## CLPPlus でのコメント

V9.7 フィックスパック 3 以降、CLPPlus ではスクリプトとコマンドにコメントを含められるようになりました。

CLPPlus では、コメントは複数の行にまたがるのが可能です。1 行だけのコメントは # あるいは -- で始めます。複数行にまたがるコメントの場合、/\* と \*/ で囲みます。

### 例

以下の例では、単一行のコメントと複数行のコメントの両方が示されています。

```
SQL> # This is a single line comment
SQL>

SQL> -- This is also a single line comment
SQL>

SQL> /* This comment
spans
multiple lines. */
SQL>
```

## CLPPlus のエスケープ文字

V9.7 フィックスパック 4 以降、CLPPlus コマンドおよび照会の中で、エスケープ文字を使用できるようになりました。

### 説明

CLPPlus 中では、アンパーサンド (&) 文字を使用して、SQL ステートメントの変数を置換できます。エスケープ文字を使用して、入力値中のアンパーサンド文字 (例えば、"AT&M") をエスケープして、置換がなされないようにすることができます。

エスケープ文字を使用して、文字 "\$" および "%" をエスケープすることもできます。これらの文字は、CLPPlus 中でシェルおよび環境変数の参照に使用されるものです。

エスケープ文字は、**SET** コマンドで定義できます。デフォルトのエスケープ文字は「¥」です。**SET** コマンドについて詳しくは、関連する解説書を参照してください。

### 例

1. この例は、デフォルトのエスケープ文字を使用して "&M" が置換変数として扱われないようにする方法を示しています。

```
SQL> set escape ON
SQL> insert into testtab values('AT¥&M');
DB250000I: The command completed successfully.
```

```
SQL> select * from testtab;
TEXT
-----
AT&M
```

2. この例は、ユーザー定義のエスケープ文字を使用して "&G" が置換変数として扱われないようにする方法を示しています。

```
SQL> set escape ^
SQL> insert into testtab values('L^&G');
DB250000I: The command completed successfully.
```

```
SQL> select * from testtab;
TEXT
-----
AT&M
L&G
```

3. この例は、エスケープ文字が使用されていない時の動作を示しています。 "&V" は置換変数として扱われ、ユーザーは "Planet" の入力値を提供する必要があります。

```
SQL> set escape OFF
SQL> insert into testtab values('Smarter &V');
Enter a value for variable V: Planet
```

```
Original statement: insert into testtab values('Smarter &V')
New statement with substitutions: insert into testtab values('Smarter Planet')
DB250000I: The command completed successfully.
```

```
SQL> select * from testtab;
TEXT
```



- ```

-----
AT&M
L&G
Smarter Planet

```
4. この例は、エスケープ文字が使用されていない時の動作を示しています。 "&V" は置換変数として扱われ、ユーザーは "Gene" の入力値を提供する必要があります。
- ```

SQL> set escape OFF
SQL> insert into testtab values('Blue %&V');
Enter a value for variable V: Gene

Original statement: insert into testtab values('Blue \&V')
New statement with substitutions: insert into testtab values('Blue \Gene')
DB250000I: The command completed successfully.

SQL> select * from testtab;
TEXT
-----
AT&M
L&G
Smarter Planet
Blue %Gene

```
5. この例は、エスケープ文字が使用されている時の動作を示しています。 "\$100" は値として扱われ、シェルまたは環境変数としては扱われません。
- ```

SQL> set escape ON
SQL> insert into testsub values('%$100');
DB250000I: The command completed successfully.

SQL> select * from testsub;
TEXT
-----
$100

```
6. この例は、エスケープ文字が使用されている時の動作を示しています。 "86%" は値として扱われ、シェルまたは環境変数としては扱われません。
- ```

SQL> set escape ON
SQL> insert into testsub values('86\%');
DB250000I: The command completed successfully.

SQL> select * from testsub;
TEXT
-----
$100
86%

```

## CLPPlus のバインド変数

バインド変数はリテラル値の代わりに使用されます。SQL ステートメントを複数回発行する場合、バインド変数を使用してリテラル値の数を減らすことができます。

### 許可

特別な許可は必要ありません。

### 宣言

バインド変数は、次の構文を使用して宣言できます。

```
▶▶—VARIABLE—name—datatype;—▶▶
```

#### *name*

バインド変数の名前を指定します。

#### *datatype*

バインド変数に関連付けられるデータ・タイプを指定します。これは、BOOLEAN、CHARACTER、DATE、DECIMAL、DOUBLE、FLOAT、INTEGER、REAL、SMALLINT、VARCHAR のいずれかです。

V9.7 フィックスパック 2 の時点で、REFCURSOR もサポートされています。REFCURSOR は、プロシージャ、関数、および PL/SQL の無名ブロックの中で、タイプ **CURSOR** の **OUT** パラメーターの値を受け取るために使用されます。

V9.7 Fix Pack 3 以降、NUMBER、NUMBER(p[,s]) および VARCHAR2 もサポートされています。NUMBER および NUMBER(p[,s]) は暗黙的に DECIMAL データ・タイプにマッピングされます。VARCHAR2 は暗黙的に VARCHAR データ・タイプにマッピングされます。

V9.7 フィックスパック 4 以降、DB2 for Linux, UNIX, and Windows サーバーでは、CLPPlus で BOOLEAN、ROW、および ARRAY データ・タイプをストアード・プロシージャのパラメーターとして使用できるようになりました。ストアード・プロシージャは、**CALL** または **EXEC CLPPlus** ステートメントで実行できます。

### 有効範囲

バインド変数は、ユーザーの CLPPlus セッション中は存続します。CLPPlus セッションが開始すれば、バインド変数を宣言してそのセッション中使用することができます。CLPPlus セッションが終了すると、すべてのバインド変数はクリアされます。

### 制約事項

SQL ステートメントまたは PL/SQL の無名ブロックで使用する場合、バインド変数は一回のみ出現可能です。バインド変数を複数回使用すると、データベース・サーバーからエラーが戻されます。

DB2 for z/OS<sup>®</sup> および Informix<sup>®</sup> Dynamic Server データ・サーバーでは、バインド変数の使用に以下の制限が課せられます。

- バインド変数は **EXEC** コマンドを使用して初期化することはできません。

```
Exec :var_name:='john' /* this is not supported */
```

- バインド変数は **begin-end** ブロックを使用して初期化することはできません。

```
begin
:var_name:='john'; /* this is not supported in a begin-end block */
end;
```

- DB2 for z/OS および Informix Dynamic Server データ・サーバーでは PL/SQL がサポートされていないので、バインド変数は PL/SQL の本文中ではサポートされていません。

- CURSOR** 型の変数はサポートされていません。

```
SQL> CREATE PROCEDURE getEmployeeData( ID INT, OUT NAME char(10),
      OUT DOB Date, OUT SAL DECIMAL(7,2))
      LET NAME='dummy';
      LET DOB='10/10/2010';
      LET SAL=0;
      SELECT empname, empdob, salary INTO name, dob, sal FROM emp WHERE empid = ID;
END PROCEDURE;
```

```
/
DB250000I: The command completed successfully.
```

```
SQL> define var_id=1001 /* usage of substitution variable */
SQL> Variable name varchar(10)
DB250000I: The command completed successfully.
SQL> Variable dob date
DB250000I: The command completed successfully.
SQL> Variable salary double
DB250000I: The command completed successfully.
```

```
Call getEmployeeData(&var_id, :name, :dob, :salary)
DB250000I: The command completed successfully.
SQL> Print name
'JOHN'
SQL> Print dob
'26/04/1982'
SQL> Print salary
10000.50
```

- 精度の値および位取りの値は、**NUMBER** データ・タイプおよび **DECIMAL** データ・タイプのバインド変数を作成する際に指定できます。精度のサポートには制限があります。割り当てられる 10 進値または数値は、変数の定義で指定された精度に合わせて変更されません。詳しくは、例 13 を参照してください。

これらの制限は、**EXECUTE CLPPlus** コマンドにも適用されます。

## 例

以下の例は、バインド変数の定義、初期化、および使用の方法を示すものです。

- 名前が **ID** および **LNAME** で、タイプが **VARCHAR** のバインド変数:

```
VARIABLE ID VARCHAR
VARIABLE LNAME VARCHAR
```

- 名前が **ID** で、PL/SQL ブロック内で初期化されるバインド変数:

```
BEGIN
  :ID := '000020';
END;
/
```

- PL/SQL ブロック内で使用されるバインド変数 **ID** および **LNAME**:

## CLPPlus のバインド変数

```
BEGIN
  SELECT lastname INTO :LNAME FROM employee
  WHERE empno = :ID;
END;
/
```

4. 単一の PL/SQL ステートメントがバインド変数 **ID** を初期化します。  
EXECUTE :ID := '000022';
5. 変数 **ID** を置換変数 *a* に基づいて初期化します (*a* は **DEFINE** CLPPlus コマンドにより定義されています)。

```
EXECUTE :ID := &a;
```

6. **ID** バインド変数を **SELECT** ステートメント内で使用します。  
SELECT lastname FROM employee WHERE empno = :ID;
7. バインド変数 **ID** および **LNAME** を **UPDATE** ステートメント内で使用します。

```
UPDATE employee SET lastname = :LNAME WHERE empno = :ID;
```

8. **salary** バインド変数を number データ・タイプで定義します。  
variable salary number  
exec :salary := 1000.00
9. **bonus** バインド変数を number(p[,s]) データ・タイプで定義します。

```
variable bonus number(6)  
exec :bonus:= 999.999
```

10. **comm** バインド変数を number(p[,s]) データ・タイプで定義します。

```
variable bonus comm(4,2)  
exec :comm:= 10.455
```

```
SQL> print comm  
10.45
```

11. **name** バインド変数を varchar2 データ・タイプで定義します。

```
variable name varchar2  
exec :name:='MICHAEL'
```

12. この例では、プロシージャーの実行中に、バインド変数を入出力引数として置換することを示します。example\_proc.db2 という名前のファイルに、次のステートメントが含まれていると仮定します。

```
CREATE PROCEDURE DEPT_MEDIAN  
(IN deptNumber SMALLINT, OUT medianSalary DOUBLE)
```

バインド変数が入出力引数として置換されました。バインド変数を次のように定義します。

```
variable in_var integer  
variable out_var double
```

プロシージャーを実行し、変数をパラメーターとして置換します。

```
call dept_median(:in_var, :out_var)
```

オプションで、**out\_var** バインド変数である出力引数の内容を出します。

```
print out_var
```

13. この例では、バインド変数 **var1** を示します。この変数は、定義内の精度を反映しません。

```
variable var1 number(4,2)  
DB250000I: The command completed successfully.
```

精度が 6 桁で、位取りが 3 桁の値を割り当てます。

```
exec :var1 := 333.333
/
DB250000I: The command completed successfully.
```

**var1** の内容を出力します。

```
print var1
333.33
```

位取りは 2 で正しいですが、精度は定義された 4 ではありません。このシナリオは、現在の機能上の制約です。

## CLPPlus のシェルおよび環境変数

V9.7 フィックスパック 4 以降、CLPPlus コマンドおよび照会の中で、シェルおよび環境変数を使用できるようになりました。

### 説明

このサポートにより、シェルまたは環境変数値を、CLPPlus コマンドまたは照会中で使用できるようになりました。この使用法は、シェルまたはコマンド・ウィンドウでこれらの変数がアクセスされる方法と同じです。

シェルと環境変数置換は、CLPPlus コマンド **SET ENVVARSUBST** を使用して使用可能または使用不可にできます。このコマンドについて詳しくは、『関連参照』を参照してください。

接頭部 "\$" が付いたいかなる変数も、シェルまたは環境変数として扱われます。この文字は、異なるオペレーティング・システムで使用されるスクリプトに、オペレーティング・システムからの独立性をもたらしています。Microsoft プラットフォームの環境変数は、変数名の前後を "%" で囲む (例えば、%VAR%) ことによってもアクセスできます。

CLPPlus は、シェルまたは環境変数に文字 a-z、A-Z、0-9、および "\_" が含まれている場合、それを理解し解決することができます。他の文字がある場合、そこで変数名が終わったものとして扱われます。次の例は、この命名規則を示しています。

```
$clplusplus <==== "clplusplus" followed by "&"
$clplusplus[version] <==== "clplusplus" followed by "["
$clplusplus version <==== "clplusplus" followed by " "(space)
$clplusplus#version <==== "clplusplus" followed by "#"
$clplusplus-version <==== "clplusplus" followed by "-"
$clplusplus(version) <==== "clplusplus" followed by "("
```

CLPPlus が変数を使用しようとしていて、かつ変数が定義されている場合、その変数の値が使用されます。CLPPlus が変数を使用しようとしておりかつ変数が定義されていない場合、空ストリング値で置換されて情報メッセージがログに記録されます。情報メッセージは、CLPPlus が冗長モードで作動している際に表示されます。

CLPPlus 中で、シェルおよび環境変数に、次の方式でアクセスできます。

- CLPPlus の開始中に、シェルまたは環境変数を引数としてスクリプトに渡します。これらの渡された変数は、&" 文字とその位置 (番号) でアクセスされる、位置パラメーターに変換されます。
- シェルまたは環境変数は、CLPPlus 中で "\$" 文字に変数名を続けたもので直接アクセスできます。この方式は、スクリプトを実行するオペレーティング・システムにかかわらず使用できます。

### 例

1. 次の例は、CLPPlus の開始中にシェルまたは環境変数をどのように使用できるのかを示しています。

UNIX および Linux プラットフォームの場合:

```
Export TABLE_NAME=employee
Export SCRIPT_PATH=/home/user

clppplus -nw @$SCRIPT_PATH/script.sql $TABLE_NAME
```

Windows プラットフォームの場合:

```
Set TABLE_NAME=employee
Set SCRIPT_PATH=c:%testfiles

clppplus -nw @%SCRIPT_PATH%\script.sql %TABLE_NAME%
```

ここで、script.sql として、次のものが含まれるものとします。

```
Select * from &1
```

そして、&1 は、変数 TABLE\_NAME の値に解決されます。

2. 次の例は、CLPPlus 中でシェルまたは環境変数をどのように直接使用できるのかを示しています。

UNIX および Linux プラットフォームの場合:

```
select * from $TABLE_NAME;
insert into test values ($NAME, $ID, $DESG);
```

Windows プラットフォームの場合:

```
select * from %TABLE_NAME%;
insert into test values (%NAME%, %ID%, %DESG%);
```

---

## CLPPlus でサポートされる DB2 コマンド

CLPPlus は、データベースおよびデータベース・マネージャーの管理、チューニング、および保守のための DB2 コマンドのサブセットをサポートします。

以下の DB2 コマンドが、CLPPlus インターフェースでサポートされています。

- GET DATABASE CONFIGURATION
- GET DATABASE MANAGER CONFIGURATION
- UPDATE DATABASE CONFIGURATION
- UPDATE DATABASE MANAGER CONFIGURATION
- RESET DATABASE CONFIGURATION
- RESET DATABASE MANAGER CONFIGURATION

V9.7 フィックスパック 3 以降、次の DB2 コマンドの CLPPlus サポートが追加されました。

- LIST PACKAGES
- IMPORT
- EXPORT
- LOAD

注: IMPORT、EXPORT および LOAD コマンドには、処理対象のファイルがサーバー上に存在する必要があるという制約があります。

## CLPPlus でサポートされる DB2 コマンド

V9.7 フィックスパック 4 以降、次の DB2 コマンドの CLPPlus サポートが追加されました。

- **REORG**、DB2 for Linux, UNIX, and Windows および DB2 for z/OS に接続する場合にサポートされます。
- **RUNSTATS**、DB2 for Linux, UNIX, and Windows および DB2 for z/OS に接続する場合にサポートされます。
- **REORGCHK**、詳しくは、関連の解説書を参照してください。

V9.7 フィックスパック 5 以降、次の DB2 コマンドの CLPPlus サポートが追加されました。

- **CREATE DATABASE** の制限付きサポート、IBM Informix への接続時にもサポートされます。制約事項について詳しくは、関連の解説書を参照してください。
- **DROP DATABASE**、IBM Informix への接続時にもサポートされます。

注: DB2 for Linux, UNIX, and Windows では、リモート・データベース・マネージャーのインスタンスに接続されている場合、**CREATE DATABASE** コマンドと **DROP DATABASE** コマンドは失敗します。ローカル・データベース・マネージャーのインスタンスに接続していなければなりません。



## CLPPlus での CREATE DATABASE

V9.7 フィックスパック 5 以降、**CREATE DATABASE** コマンドの制限付きサポートが、DB2 for Linux, UNIX, and Windows および IBM Informix の CLPPlus に追加されました。

### 制約事項

DB2 for Linux, UNIX, and Windows では、リモート・データベース・マネージャーのインスタンスに接続されている場合、**CREATE DATABASE** コマンドは失敗します。ローカル・データベース・マネージャーのインスタンスに接続していなければなりません。CLPPlus インターフェースにおいて、**CREATE DATABASE** コマンドで有効なパラメーターは以下のものだけです。

- 必須の *database-name* 変数
- オプションの **CODESET** パラメーター
- オプションの **TERRITORY** パラメーター
- V9.7 フィックスパック 6 の時点では、オプションの **PAGESIZE** パラメーター

**注:** バージョン 10.1 フィックスパック 2 以降では、DB2 for Linux, UNIX, and Windows データベースを作成するときに **pagesize** パラメーターを指定しない場合、新しいデータベースのデフォルトのページ・サイズは 32K になります。この変更の前は、デフォルトのページ・サイズは 4K に設定されていました。

IBM Informix では、CLPPlus インターフェースの **CREATE DATABASE** に、特に **sysmaster** データベースに対する接続信用証明情報が必要です。接続が存在しない場合、接続信用証明情報を要求するプロンプトが出されます。IBM Informix での **CREATE DATABASE** コマンドについて詳しくは、[http://publib.boulder.ibm.com/infocenter/idshelp/v115/index.jsp?topic=/com.ibm.sqls.doc/ids\\_sq\\_0368.htm](http://publib.boulder.ibm.com/infocenter/idshelp/v115/index.jsp?topic=/com.ibm.sqls.doc/ids_sq_0368.htm) を参照してください。

### 例

次のコマンドは、DB2 for Linux, UNIX, and Windows に testdb という名前のデータベースを作成します。

```
create database testdb;
```

次のコマンドは、DB2 for Linux, UNIX, and Windows にページ・サイズが 4K にされた testdb という名前のデータベースを作成します。

```
create database testdb pagesize 4K;
```

次のコマンドは、DB2 for Linux, UNIX, and Windows に testdb という名前のデータベースを作成します。コード・ページ・セットは UTF-8 として定義され、テリトリーは US として定義されます。

```
create db testdb using codeset UTF-8 territory US;
```

次のコマンドは、IBM Informix に udttest という名前のデータベースを作成します。以前の接続が存在しないため、コマンドから情報を要求するプロンプトが出されます。

```
SQL> create database udttest;
```

```
Enter DATABASE NAME [sysmaster]:
Enter HOSTNAME [localhost]: 9.130.34.100
```

## CLPPlus での CREATE DATABASE

```
Enter PORT [50000]: 9089
Enter ID: informix
Enter password: *****
```

```
DB250000I: The command completed successfully.
```

次の例では、最初のコマンドにより IBM Informix データベースに接続されます。2 番目のコマンドは、udttest という名前のデータベースを作成します。

```
SQL> connect Informix/informix123@9.130.34.100:9089/stores
```

```
Database Connection Information :
```

```
-----
Hostname = 9.130.34.100
Database server = IDS/NT32 IFX11700
SQL authorization ID = informix
Local database alias = stores
Port = 9089
```

```
SQL> create database udttest with log mode ansi ;
DB250000I: The command completed successfully.
```

---

## CLPPlus の制約事項

CLPPlus には、特定の接続、コマンド、およびステートメントに関する制約事項があります。

CLPPlus は、以下の DB2 データベース製品とのデータベース接続を確立することができます。ただし、CLPPlus バージョンは少なくとも V9.7 フィックスパック 6 でなければなりません。

- IBM DB2 for IBM i
- DB2 Express-C

CLPPlus は、以下の DB2 データベース製品とのデータベース接続を確立することができます。ただし、制約事項があります。

- DB2 for Linux, UNIX, and Windows バージョン 9.8 フィックスパック 1 以降。接続をサポートするには、事前にフィックスパック 1 を V9.8 に適用しておく必要があります。

CLPPlus には、PL/SQL サポートに関して以下の制約事項があります。

- PL/SQL 関数およびトリガーを、パーティション・データベース環境で作成することはできません。
- NCLOB データ・タイプは、データベースが Unicode データベースとして定義されていない場合には PL/SQL ステートメントまたは PL/SQL コンテキストで使用することはできません。Unicode データベースの場合、NCLOB データ・タイプは DB2 DBCLOB データ・タイプにマップされます。
- XMLTYPE データ・タイプはサポートされていません。
- TYPE 宣言は、関数、プロシージャ、トリガー、または匿名ブロックではサポートされていません。
- FOR EACH STATEMENT オプションは、PL/SQL トリガーではサポートされていません。

表 6. データ・サーバー別の CLPPlus の制限。

CLPPlus フィーチャー	DB2 for Linux, UNIX, and Windows	DB2 for z/OS	IBM Informix
変数 REFCURSOR	はい	いいえ	いいえ
SERVEROUTPUT	はい	いいえ	いいえ
EXECUTE	はい	いいえ	いいえ
LIST PACKAGES	はい	はい	いいえ
SHOW ERRORS	はい	いいえ	いいえ
UPDATE/GET/RESET DB CFG	はい	いいえ	いいえ
UPDATE/GET/RESET DBM CFG	はい	いいえ	いいえ
EXPORT	はい	いいえ	いいえ
IMPORT	はい	いいえ	いいえ
LOAD	はい	いいえ	いいえ

## CLPPlus トラブルシューティングのヒント

CLPPlus 問題とエラーをトラブルシューティングする際に、以下の情報を利用してください。

表 7. CLPPlus の開始: 問題と解決策

課題	解説および解決策
<p>以下のメッセージが CLPPlus を開始しようとする则表示されます。</p> <p>CLPPlus requires Java 1.5 or higher to execute. Please ensure Java is in your PATH.</p> <p>CLPPlus を開始できません。</p>	<p>Java 1.5 以降がインストールされていることを確認してください。CLPPlus インストール・プロセスで Java 製品がインストールされない場合、CLPPlus は Java 製品を <b>JAVA_HOME</b> および <b>PATH</b> 環境変数で探します。</p> <p>Java JRE または SDK バージョン 1.5 以降をダウンロードおよびインストールしてください。<b>JAVA_HOME</b> が Java インストール・ディレクトリーを指すように設定します。Java bin ディレクトリーを <b>PATH</b> 環境変数に追加します。</p>
<p>以下のメッセージが CLPPlus を開始しようとする则表示されます。</p> <p>Could not find db2jcc.jar. Please ensure that your installation completed successfully. If the problem persists, please locate and add db2jcc.jar to your CLASSPATH.</p> <p>CLPPlus を開始できません。</p>	<p>CLPPlus では、Java Universal Driver が db2jcc.jar ファイルに必要です。CLPPlus が db2jcc.jar ファイルを <b>CLASSPATH</b> または <i>installation directory/java</i> ディレクトリーで検出できないと、始動が失敗します。</p> <p>インストールが正常に行われたことを確認してください。db2jcc.jar ファイルへの絶対パスを <b>CLASSPATH</b> 環境変数設定に追加します。</p>

表 7. CLPPlus の開始: 問題と解決策 (続き)

課題	解説および解決策
<p>以下のメッセージが CLPPlus を開始しようとすると表示されます。</p> <p>Could not find clpplus.jar. Please ensure that your installation completed successfully. If the problem persists, please locate and add clpplus.jar to your CLASSPATH.</p> <p>CLPPlus を開始できません。</p>	<p>CLPPlus では、製品に付属している clpplus.jar ファイルが必要です。CLPPlus が clpplus.jar を <b>CLASSPATH</b> またはインストール・ディレクトリーで検出できないと、始動が失敗します。</p> <p>インストールが正常に行われたことを確認してください。db2jcc.jar ファイルへの絶対パスを <b>CLASSPATH</b> 環境変数設定に追加します。</p>
<p>以下のメッセージが CLPPlus を開始しようとすると表示されます。</p> <p>Could not find jline-0.9.93.jar. Please ensure that your installation completed successfully. If the problem persists, please locate and add jline-0.9.93.jar to your CLASSPATH.</p> <p>CLPPlus を開始できません。</p>	<p>CLPPlus では、製品に付属している jline-0.9.93.jar ファイルが必要です。CLPPlus が <b>CLASSPATH</b> またはインストール・ディレクトリーで jline-0.9.93.jar を見つけられないとき、CLPPlus は開始に失敗します。</p> <p>インストールが正常に行われたことを確認してください。jline-0.9.93.jar ファイルへの絶対パスを <b>CLASSPATH</b> 環境変数設定に追加します。</p>

## CLPPlus のトレースおよびレコードのロギング

CLPPlus は、ファイルへのトレースおよびログの記録用のメカニズムを提供します。CLPPlus は、CLPPlus クライアント・レイヤーおよび JDBC ドライバー・レイヤーからのロギングまたはトレースをサポートします。

IBM Data Server Driver for JDBC and SQLJ と IBM Data Server Driver for ODBC and CLI は、包括的なトレース機能を提供します。これらの機能は CLPPlus に拡張されました。トレース機能は、アプリケーションが特定のドライバー (CLPPlus クライアント・レイヤーまたは JDBC ドライバー・レイヤー) にアクセスするたびに、**SET** コマンドを使用してテキスト・ログ・ファイルを生成します。これらのログ・ファイルは、CLPPlus クライアントおよび JDBC に関する以下のような詳細情報を提供します。

- アプリケーションによって呼び出された関数
- 関数の内容 (受け渡された入出力パラメーターを含む)
- 関数からの戻りコードおよび生成されたエラー (または警告メッセージ)。

CLPPlus トレース機能を構成するには、CLPPlus コマンド・プロンプトから **SET** コマンドを発行します。クライアント・レイヤーまたはドライバー・レイヤーのトレースを使用可能にするには、**LOGMODE** パラメーターを以下のように設定します。

```
CLPPlus> SET LOGMODE logmode-value
```

ここで、*logmode-value* は、トレースを実行するかどうか、およびどのレイヤーが対象になるかを示します。デフォルト値は **NONE** であり、トレースが実行されないことを示します。他の有効な値は、**CLPPLUS** (クライアント・レイヤーをトレースす

る)、JCC (JDBC レイヤーをトレースする)、および BOTH (クライアント・レイヤーと JDBC レイヤーの両方をトレースする) です。

より詳細な JDBC トレースを実行するには、以下のように *logmode-value* を JCC または BOTH に設定し、**JCCLOGMODE** パラメーターを指定します。

```
SET LOGMODE JCC  
SET JCCLOGMODE jcclogmode_value
```

ここで、*jcclogmode\_value* はトレースおよびログの対象となるフィーチャーを示します。有効な *jcclogmode\_value* 設定について詳しくは、820 ページの『SET』を参照してください。



---

## 第 4 章 コマンド構文ヘルプの読み方

時折、コマンドに有効なオプションを忘れてしまうことがあるかもしれません。ここに説明されている構文規則を使用する、便利なコマンド・ヘルプ画面出力を呼び出すことができます。

コマンド行プロセッサ (CLP) のコマンドはすべて、コマンド・キーワードの前に疑問符 (?) を付けることによって、CLP プロンプトからヘルプ画面を呼び出すことができます。多くのシステム・コマンドでは、コマンド・キーワードの後に **help** パラメーターを付けて発行すると、サマリーのヘルプ画面を表示できます。

### ヘルプの呼び出し

#### CLP コマンド

CLP コマンド・ヘルプ画面を表示するには、db2 対話式モードのプロンプト (db2 =>) でコマンド・キーワードの前に疑問符を付けます。**BACKUP DATABASE** コマンドの例を次に示します。

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

または、「db2」対話式モードでない場合は、それぞれのコマンド・ヘルプ画面呼び出しの前に db2 を付けます。**BACKUP DATABASE** コマンドの例を次に示します。

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

#### システム・コマンド

システム・コマンドの大部分では、システム・コマンド・キーワードの後にヘルプ・オプションを入力すると、コマンド・ヘルプ画面を表示できます。多くのシステム・コマンドは一般的なヘルプ・オプションを使用しますが、一部のシステム・コマンドでは、異なる *help* オプションまたは追加の *help* オプション (あるいはその両方) を使用することがあります。忘れてしまったコマンドの *help* オプションを最初からすぐに検索する必要はありません。まず、次の最も一般的なオプションを試してみます。通常は、正常にコマンド・ヘルプ画面を呼び出せます。

#### ヘルプ・オプション

- -h
- -?
- -help
- コマンド・キーワードの後に何も入力しない。

注: コマンド・キーワードの後に何も入力しないと、場合によっては、オプションが必要ないコマンドで実際にコマンドが実行されることがあります。

### ヘルプ画面の構文規則

[ ] オプション・パラメーターを囲みます

{ } 必須パラメーターを囲みます

## コマンド構文ヘルプの読み方

- | 2 つ以上の項目を区切り、そのうち 1 つだけを選択できることを示します
  - ... 反復可能なパラメーターを示します
  - ( ) 反復可能なパラメーターの区切り文字 (常に使用するとは限りません)
- コマンド・キーワードは大文字で表示します
- ユーザーによる指定および適切な入力が必要な変数は、小文字で示します

### コマンド・ヘルプ画面の出力例

次に示すのは、**UPDATE MONITOR SWITCHES** コマンドの CLP コマンド・ヘルプ画面です。

```
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
```

次に示すのは **db2look** コマンドのシステム・コマンド・ヘルプ画面ですが、この場合は、指定された **-h** ヘルプ・オプションで呼び出されたものではありません。

```
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: Database Name: This must be specified

-e: Extract DDL file needed to duplicate database

-xs: Export XSR objects and generate a script containing DDL statements

-xdir: Path name: the directory in which XSR objects will be placed

-u: Creator ID: If -u and -a are both not specified then \$USER will be used

-z: Schema name: If -z and -a are both specified then -z will be ignored

-t: Generate statistics for the specified tables

-tw: Generate DDLs for tables whose names match the pattern criteria (wildcard characters) of the table name

-ap: Generate AUDIT USING Statements

-wlm: Generate WLM specific DDL Statements

-h: More detailed help message

-o: Redirects the output to the given file name

-a: Generate statistics for all creators

-m: Run the db2look utility in mimic mode

-c: Do not generate COMMIT statements for mimic

-r: Do not generate RUNSTATS statements for mimic

-l: Generate Database Layout: Database partition groups, Bufferpools and Tablespace

-x: Generate Authorization statements DDL excluding the original definer of the object

-xd: Generate Authorization statements DDL including the original definer of the object

-f: Extract configuration parameters and environment variables

-td: Specifies x to be statement delimiter (default is semicolon(;))

-i: User ID to log on to the server where the database resides

-w: Password to log on to the server where the database resides

-noview: Do not generate CREATE VIEW ddl statements

-wrapper: Generates DDLs for federated objects that apply to this wrapper

-server: Generates DDLs for federated objects that apply to this server

-FEDONLY: Only created Federated DDL Statements



- nofed: Do not generate Federated DDL
- fd: Generates db2fopt statements for opt\_buffpage and opt\_sortheap along with other  
cfg and env parameters.
- v: Generate DDL for view only, this option is ignored when -t is specified
- dp: Generate DROP statement before CREATE statement
- ct: Generate DDL Statements by object creation time

注: 一般に、システム・コマンド・ヘルプ画面は CLP コマンド・ヘルプ画面よりも詳細な情報を提供する傾向があります。

## コマンド入力の例

**UPDATE MONITOR SWITCHES** コマンド・ヘルプ画面を例として取り上げます。

```
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
```

次のコマンド入力は有効です。

```
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
```

しかし、次のコマンド入力は無効です。

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

## 注意

オンライン・インフォメーション・センターで検索せずにコマンド・ヘルプ画面の構文規則を入手するには、CLP プロンプトで次のコマンドを発行します。

```
db2 => ? help
```

または、システム・コマンド・プロンプトで、次の照会を入力します。

```
=> db2 ? help
```



---

## 第 5 章 CLP コマンド

## ACTIVATE DATABASE

指定されたデータベースをアクティブ化し、必要なデータベースのサービスをすべて開始します。こうして、データベースが接続可能になり、任意のアプリケーションが使用できるようになります。

### 有効範囲

このコマンドは、インスタンス内のすべてのメンバーで、ターゲット・データベースをアクティブ化します。アクティブ化中に 1 つ以上のメンバーがエラーを検出すると、警告が戻されます。コマンドが正常に処理されたすべてのメンバーで、データベースは活動状態を維持します。

データベースは暗黙的または明示的にアクティブ化されます。最初のユーザー接続によって主導されるデータベースのアクティブ化は、暗黙的なアクティブ化として知られています。ACTIVATE DATABASE コマンドの発行に基づくデータベースのアクティブ化は、明示的なアクティブ化として知られています。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMANT

### 必要な接続

なし

### コマンド構文

```

▶▶ ACTIVATE { DATABASE | DB } database-alias
▶
  USER username USING password

```

### コマンド・パラメーター

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

開始するデータベースの別名を指定します。

**USER** *username*

データベースを開始するユーザーを指定します。

**USING** *password*

ユーザー名のパスワードを指定します。

### 使用上の注意

データベース管理者は **ACTIVATE DATABASE** コマンドを使用して、選択したデータベースを開始させることができます。こうすると、アプリケーションがデータベース

の初期設定で時間を浪費してしまうことを避けられます。 **ACTIVATE DATABASE** コマンドは、データベースを明示的にアクティブ化する唯一の方法です。

**ACTIVATE DATABASE** コマンドによって初期化されたデータベースは、**DEACTIVATE DATABASE** または **db2stop** コマンドを使用することによってシャットダウンできません。

**ACTIVATE DATABASE** コマンドを実行するアプリケーションは、どのデータベースへのアクティブ・データベース接続も持つことができません。

データベースが **CONNECT** ステートメント (または暗黙的な接続) の発行によって開始され、その後同じデータベースに対して **ACTIVATE DATABASE** コマンドを発行した場合、そのデータベースをシャットダウンするには **DEACTIVATE DATABASE** コマンドを使用しなければなりません。 **ACTIVATE DATABASE** コマンドを使用しないでデータベースを始動した場合、最後のアプリケーションが切断した時にデータベースはシャットダウンされます。ただし、DB2 pureScale® 環境では、あるメンバーのアクティブ・データベースは、**DEACTIVATE DATABASE** コマンドを発行することによってのみ、そのメンバーで非アクティブ化することができます。

DB2 pureScale 環境では、**ACTIVATE DATABASE** コマンドと **CONNECT** ステートメントのどちらも、データベースを再始動するために使用することはできません。この場合、データベースは **autorestart** データベース構成パラメーターの値に応じて、自動的に再始動します。ターゲット・データベースが不整合状態にあるときのデータベース接続の動作について、**autorestart** 構成パラメーターを参照してください。

## ADD CONTACT

システムでローカルに定義できる連絡先リストかまたはグローバル・リストで定義できる連絡先リストに、連絡先を追加します。連絡先とは、スケジューラーおよびヘルス・モニターなどのプロセスが、メッセージを送信する先のユーザーです。

Database Administration Server (DAS) の **contact\_host** 構成パラメーターの設定により、リストがローカルかグローバルかが決まります。

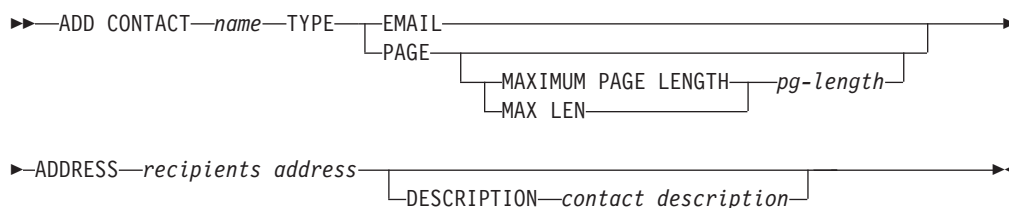
### 許可

なし

### 必要な接続

なし。ローカル実行のみ: このコマンドはリモート接続では使用できません。

### コマンド構文



### コマンド・パラメーター

#### ADD CONTACT *name*

追加される連絡先の名前。デフォルトでは、DB2 Administration Server 構成パラメーター **contact\_host** が別のシステムを指示していない限り、連絡先はローカル・システムに追加されます。

**TYPE** 連絡の方法。次のうちいずれかです。

**EMAIL** この連絡先には、(ADDRESS) に E メールで通知します。

**PAGE** この連絡先には、ADDRESS にページを送信することで通知します。

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

ページング・サービスにメッセージ長に関する制限がある場合、その制限を文字数で指定します。

通知システムは、SMTP プロトコルを使用して、2DB2 Administration Server 構成パラメーター **smtp\_server** が指定するメール・サーバーに通知を送信します。E メールを送信したり、ページャーを呼び出したりするのは、SMTP サーバーの役割です。

#### **ADDRESS** *recipients-address*

宛先の SMTP メールボックス・アドレス。例えば、joe@somewhere.org などです。 **smtp\_server** DAS 構成パラメーターは、SMTP サーバーの名前に設定することが必要です。

#### **DESCRIPTION** *contact description*

連絡先のテキスト記述。長さは、最大 128 文字です。

## ADD CONTACTGROUP

ローカル・システムで定義されたグループのリストに、新しい連絡先グループを追加します。連絡先グループとは、スケジューラーおよびヘルス・モニターなどのモニター・プロセスが、メッセージを送信する先のユーザーおよびグループのリストです。

Database Administration Server (DAS) の **contact\_host** 構成パラメーターの設定により、リストがローカルかグローバルかが決まります。

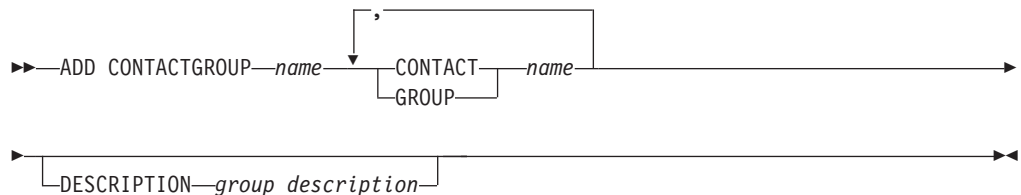
### 許可

なし

### 必要な接続

なし。ローカル実行のみ: このコマンドはリモート接続では使用できません。

### コマンド構文



### コマンド・パラメーター

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

新しい連絡先グループの名前。システム上のグループの集合の中で固有なものでなければなりません。

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

グループのメンバーである連絡先の名前。グループに追加された後、**ADD CONTACT** コマンドを使用して連絡先を定義できます。

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

このグループがメンバーである連絡先グループの名前。

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

オプション。連絡先グループのテキスト記述。

## ADD DBPARTITIONNUM

データベース・パーティションをデータベース・パーティション・サーバーに追加します。

### 有効範囲

このコマンドは、それが実行されるデータベース・パーティション・サーバーにのみ影響を与えます。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL

### 必要な接続

なし

### コマンド構文

```

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

### コマンド・パラメーター

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

新しい SYSTEM TEMPORARY 表スペース用のコンテナが、*db-partition-number* で指定されたデータベース・パーティション・サーバーのデータベースのコンテナと同一であることを指定します。指定するデータベース・パーティション・サーバーは、あらかじめ `db2nodes.cfg` ファイルで定義されていなければなりません。

自動ストレージを使用するよう定義された SYSTEM TEMPORARY 表スペース (これは CREATE TABLESPACE ステートメントの MANAGED BY AUTOMATIC STORAGE 節を使って作成されたか、あるいは MANAGED BY CLAUSE がまったく指定されなかった SYSTEM TEMPORARY 表スペースを示します) の場合、これらのコンテナは指定されたパーティションのコンテナと必ずしも一致しません。その代わりに、コンテナは、表スペースのストレージ・グループに関連付けられたストレージ・パスに基づいてデータベース・マネージャーによって自動的に割り当てられます。その結果、これら 2 つのパーティションで使用されるコンテナと同じになるとは限りません。

#### WITHOUT TABLESPACES

SYSTEM TEMPORARY 表スペースのコンテナが追加するデータベース・パーティションに対して作成されないことを指定します。データベースを使用する前に、ALTER TABLESPACE ステートメントを使用して、SYSTEM TEMPORARY 表スペース・コンテナを各データベース・パーティションに追加しなければなりません。



オプションを指定しない場合、SYSTEM TEMPORARY 表スペース用のコンテナは各データベースのカタログ・パーティション上のコンテナと同じになります。カタログ・パーティションは、パーティション・データベース環境内の各データベースごとに別々のデータベース・パーティションにすることができます。自動ストレージを使用するよう定義された SYSTEM TEMPORARY 表スペース (これは CREATE TABLESPACE ステートメントの MANAGED BY AUTOMATIC STORAGE 節を使って作成されたか、あるいは MANAGED BY CLAUSE がまったく指定されなかった SYSTEM TEMPORARY 表スペースを示します) の場合、このオプションは無視されます。このような表スペースに関しては、コンテナ作成を先に延ばすことはできません。コンテナは、データベース・マネージャーにより、データベースに関連するストレージ・パスを基に自動的に割り当てられます。

## 使用上の注意

このコマンドを使用できるのは、データベース・パーティション・サーバーに 1 つのデータベースがあり、そのデータベースがパーティションの追加操作時にカタログされていない環境に追加される場合に限りです。この状況では、データベースはカタログされないため、パーティションの追加操作ではデータベースを認識しません。また、新規データベース・パーティション・サーバー上にデータベースのデータベース・パーティションを作成しません。新規データベース・パーティション・サーバー上でデータベース・パーティションに接続しようとする、エラーが発生します。データベースを最初にカタログしてから、**ADD DBPARTITIONNUM** コマンドを使用して、新規データベース・パーティション・サーバー上にデータベースのデータベース・パーティションを作成する必要があります。

このコマンドは、複数のデータベースがあり、そのデータベースの少なくとも 1 つがパーティションの追加操作時にカタログされていない環境では使用してはなりません。この状況では、**CREATE DATABASE** コマンドの **AT DBPARTITIONNUM** パラメーターを使用して、パーティション追加操作時にカタログされなかったデータベースごとにデータベース・パーティションを作成します。アンカタログされたデータベースをそれぞれ最初にカタログしてから、**CREATE DATABASE** コマンドを使用して、新規データベース・パーティション・サーバー上にデータベースのデータベース・パーティションを作成する必要があります。

新規データベース・パーティションを追加する前に、コンテナを作成するだけの十分なストレージがあることを確認してください。

データベース・パーティション・サーバーの追加操作は、インスタンス中に存在する各データベースに、空のデータベース・パーティションを作成します。新規データベース・パーティションの構成パラメーターは、デフォルト値に設定されます。

**注:** 新規データベース・パーティションを追加する際に、アンカタログされたデータベースは認識されません。アンカタログされたデータベースは、新規データベース・パーティションには存在しません。新規データベース・パーティション上でデータベースに接続しようとする、エラー・メッセージ **SQL1013N** が戻されます。

データベース・パーティション・サーバーをローカルで作成中にデータベース・パーティション・サーバーの追加操作が失敗すると、この操作は終結処理フェーズに入り、既に作成されているすべてのデータベースをローカルにドロップします。こ

## ADD DBPARTITIONNUM

これは、追加中のデータベース・パーティション・サーバーからのみデータベース・パーティションが削除されることを意味しています。その他のデータベース・パーティション・サーバー上に存在しているデータベース・パーティションは影響を受けません。この終結処理のフェーズが失敗すると、終結処理は停止し、エラーが戻されます。

データベース・パーティション・サーバーの新規データベース・パーティションには、ALTER DATABASE PARTITION GROUP ステートメントを使用してデータベース・パーティション・グループにデータベース・パーティションが追加されるまで、ユーザー・データを入れることができません。

データベース作成操作またはデータベース・ドロップ操作が進行中の場合、このコマンドは失敗します。コマンドは、競合する操作がいったん完了してから、再発行できます。

データベースの自動ストレージが有効になっているかどうかを判別するために、**ADD DBPARTITIONNUM** はインスタンス内の各データベースごとに、カタログ・パーティションと通信を行う必要があります。自動ストレージが有効になっていれば、その通信の中でストレージ・グループ定義が取得されます。同様に、データベース・パーティションに **SYSTEM TEMPORARY** 表スペースを作成する場合は、別のデータベース・パーティション・サーバーに常駐するデータベース・パーティションの表スペース定義を入手するために、**ADD DBPARTITIONNUM** はそのサーバーと通信しなければならないかもしれません。 **start\_stop\_time** データベース・マネージャー構成パラメーターを使用して、時間 (分) を指定します。他のデータベース・パーティション・サーバーはこの時間内で自動ストレージおよび表スペース定義の応答を行う必要があります。この時間を超えると、このコマンドは失敗します。そのような場合は、**start\_stop\_time** の値を増やして、コマンドを再発行してください。

**ADD DBPARTITIONNUM** コマンドが発行されるときに、トランザクションに整合しないストレージ・グループ・エントリーがあってはなりません。

## ADD XMLSCHEMA DOCUMENT

登録が完了する前に、1 つ以上の XML スキーマ文書を、既存の未完了の XML スキーマに追加します。

### 許可

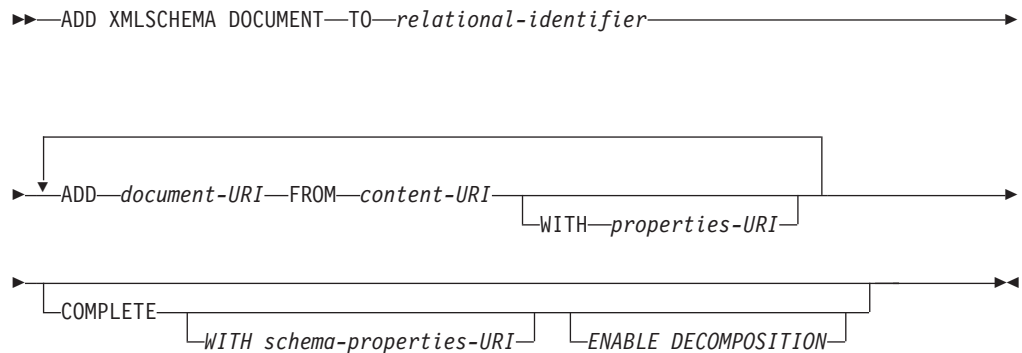
以下の権限が必要です。

- ユーザー ID は、カタログ・ビュー SYSCAT.XSROBJECTS で記録されたとおりに XSR オブジェクトの所有者でなければなりません。

### 必要な接続

データベース

### コマンド構文



### 説明

#### TO *relational-identifier*

追加のスキーマ文書が追加される、登録済みであっても未完了の XML スキーマのリレーショナル名を指定します。

#### ADD *document-URI*

このスキーマに追加される XML スキーマ文書の Uniform Resource Identifier (URI) を指定します。この文書は別の XML 文書から参照されることがあるからです。

#### FROM *content-URI*

XML スキーマ文書が置かれている URI を指定します。ファイル・スキーム URI だけがサポートされています。

#### WITH *properties-URI*

XML スキーマのプロパティ文書の URI を指定します。ファイル・スキーム URI だけがサポートされています。

#### COMPLETE

これ以上の XML スキーマ文書は追加されないことを示します。これが指定される場合、スキーマの妥当性検査が行われ、エラーが見つからなければ使用できるものとしてマークされます。

## ADD XMLSCHEMA DOCUMENT

### WITH *schema-properties-URI*

XML スキーマのプロパティ文書の URI を指定します。ファイル・スキーム URI だけがサポートされています。

### ENABLE DECOMPOSITION

このスキーマが XML 文書の分解のために使用されることを指定します。

### 例

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

## ARCHIVE LOG

リカバリー可能データベースのアクティブ・ログ・ファイルをクローズし、切り捨てます。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMANT
- DBADM

### 必要な接続

なし。このコマンドは、コマンドの持続期間の間、データベース接続を確立します。

### コマンド構文

```

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

```

#### On Database Partition Number Clause:

```

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

```

#### Database Partition Number List Clause:

```

| DBPARTITIONNUM |
| DBPARTITIONNUMS |
▶ ( db-partition-number TO db-partition-number )

```

### コマンド・パラメーター

#### DATABASE *database-alias*

アーカイブするアクティブ・ログを持つデータベースの別名を指定します。

**USER** *username*

接続を試みるユーザー名を指定します。

**USING** *password*

ユーザー名を認証するためのパスワードを指定します。

**ON ALL DBPARTITIONNUMS**

コマンドを `db2nodes.cfg` ファイルにあるすべてのデータベース・パーティションで発行することを指定します。データベース・パーティション番号節が指定されていない場合、これがデフォルトです。

**EXCEPT**

コマンドを、データベース・パーティション番号リストに指定されたデータベース・パーティションを除く、`db2nodes.cfg` ファイルにあるすべてのデータベース・パーティションで発行することを指定します。

**ON DBPARTITIONNUM | ON DBPARTITIONNUMS**

指定されたデータベースのログをデータベース・パーティションのセットでアーカイブすることを指定します。

*db-partition-number*

データベース・パーティション番号リスト内のデータベース・パーティション番号を指定します。

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

ログをアーカイブするデータベース・パーティションの範囲を指定するときに使用されます。指定された最初のデータベース・パーティション番号から 2 番目のデータベース・パーティション番号までのすべてのデータベース・パーティションがデータベース・パーティション・リストに含まれます。

### 使用上の注意

このコマンドは、ある時点までのログ・データの完全なセットを収集するために使用できます。次に、そのログ・データを使用してスタンバイ・データベースを更新することができます。

**ARCHIVE LOG** コマンドが発行されるときまでのログ・データが、ログ・ファイルの中にある場合、このログ・ファイルは切り捨てられ、ロギングは次のファイルで続行されます。

このコマンドは、起動側アプリケーションまたはシェルに、指定されたデータベースへのデータベース接続がないときにしか実行できません。これにより、コミットされていないトランザクションでユーザーがコマンドを実行するのを防ぎます。実際に、**ARCHIVE LOG** コマンドは、ユーザーの不完全なトランザクションをコミットしません。起動側アプリケーションまたはシェルに、指定されたデータベースへのデータベース接続が既に存在している場合は、コマンドは終了してエラーを戻します。このコマンドを実行したときに、指定されたデータベースで進行中のトランザクションが別のアプリケーションにあった場合には、コマンドがログ・バッファをディスクにフラッシュするため、パフォーマンスがやや低下する可能性があります。ログ・レコードをバッファに書き込む別のトランザクションは、フラッシュが完了するまで待機しなければなりません。

パーティション・データベース環境で使用する場合は、データベース・パーティション番号節を使用してデータベース・パーティションのサブセットを指定できます。データベース・パーティション番号節が指定されていない場合、このコマンドのデフォルトの動作は、クローズしてすべてのデータベース・パーティションのアクティブ・ログをアーカイブすることです。

このコマンドを使用すると、アクティブ・ログ・ファイルの切り捨てのために、アーカイブ・ログ・スペースの部分を使い果たします。アクティブ・ログ・スペースは、切り捨てられたログが非アクティブになると前のサイズを再開します。このコマンドを頻繁に使用すると、トランザクションで使用できるアクティブ・ログ・スペースの量が劇的に削減できます。使用可能な十分のログ・スペースがない場合には、このコマンドは `SQL0964C` で失敗します。

**ARCHIVE LOG** コマンドは非同期です。**ARCHIVE LOG** コマンドを発行すると、ログがクローズされてアーカイブの実行が可能になります。ログが直ちにアーカイブされるわけではありません。したがって、このコマンドを実行依頼してからログがアーカイブされるまで遅延が生じることがあります。この遅延は、**ARCHIVE LOG** コマンドを発行した直後にデータベースを非活動化した場合に特に顕著です。ログのアーカイブは `db2logmgr` プロセスによって行われます。次回のデータベース・アクティブ化までログがアーカイブされないことがあります。

このコマンドは、DB2 pureScale 環境で **ARCHIVE LOG** コマンドの発行または `db2ArchiveLog` API の呼び出しを行うと、データベースがアクセスできるすべてのメンバーに配布されます。メンバーが整合状態である場合 (オンライン/オフラインには無関係)、**ARCHIVE LOG** は、そのメンバーをスキップします。その最後のログ・ファイルは、このメンバーが最後にクリーンにシャットダウンしたときに既に切り捨てられたはずだからです。オフラインのメンバーが整合していない場合には、**ARCHIVE LOG** は、切り捨ての前に `MCR` が完了するのを待ちます。

## ATTACH

インスタンス・レベルのコマンド (例えば **CREATE DATABASE** および **FORCE APPLICATION**) を実行するインスタンスを、アプリケーションが指定できるようにします。このインスタンスは、現在のインスタンス、同じワークステーション上の別のインスタンス、またはリモート・ワークステーションのインスタンスのいずれかになります。

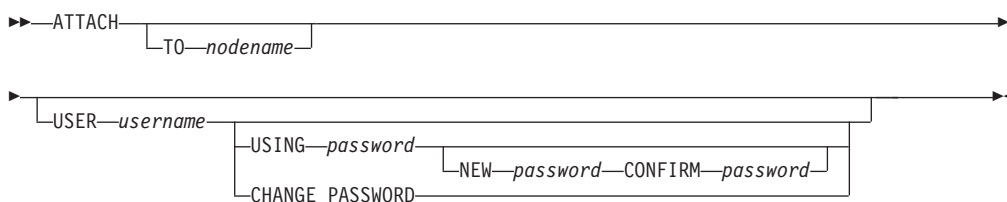
### 許可

なし

### 必要な接続

なし。このコマンドは、インスタンス接続を確立します。

### コマンド構文



### コマンド・パラメーター

#### TO *nodename*

ユーザーがアタッチすることを希望するインスタンスの別名。このインスタンスには、ローカル・ノード・ディレクトリーに一致する項目がなければなりません。この唯一の例外はローカル・インスタンス (**DB2INSTANCE** 環境変数で指定されている) です。これは、アタッチのオブジェクトとして指定できますが、ノード・ディレクトリーのノード名として使用することはできません。

#### USER *username*

認証 ID を指定します。Windows オペレーティング・システム上の DB2 データベース・インスタンスにアタッチする場合は、Microsoft Security Account Manager (SAM) と互換性のあるフォーマットでユーザー名を指定することができます。修飾子は、フラット・スタイル (最大長 15 文字) の名前であればなりません。例えば、*domainname¥username* のようにします。

#### USING *password*

ユーザー名のパスワードを指定します。ユーザー名は指定されているが、パスワードは指定されていない場合、現在のパスワードを要求するプロンプトがユーザーに出されます。入力時にパスワードは表示されません。

#### NEW *password*

ユーザー名に割り当てられる新規パスワードを指定します。パスワードが変更されるシステムは、ユーザー認証がセットアップされた方法によって異なります。DB2 データベース・システムは、AIX、Linux、および Windows



オペレーティング・システムでパスワードを変更するためのサポートを提供します。また、独自に作成したプラグインに対して最大 255 文字までサポートします。パスワードについて詳しくは、『パスワードの規則』を参照してください。

#### **CONFIRM password**

新規パスワードと同一のストリング。このパラメーターは、入力エラーを検出するために使用されます。

#### **CHANGE PASSWORD**

このオプションが指定されていると、ユーザーにプロンプトが出され、現在のパスワード、新規パスワード、および新規パスワードの確認を要求します。入力時にパスワードは表示されません。

### **例**

2 つのリモート・ノードをカタログにします。

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

node1 にアタッチし、すべてのユーザーを強制終了し、その後デタッチします。

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

node2 にアタッチして、どのアプリケーションが接続しているかを確認します。

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

コマンドがエージェント ID 1、2 および 3 を戻した後で 1 および 3 を強制終了し、その後デタッチします。

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

現在のインスタンスにアタッチして (必ずしも必要ではありません。暗黙的にアタッチされます)、すべてのユーザーを強制終了し、その後デタッチします (AIX のみ)。

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

### **使用上の注意**

コマンドから *nodename* を省略すると、現在のアタッチ状態についての情報が戻されます。

**ATTACH** が実行されなかった場合、インスタンス・レベルのコマンドは、**DB2INSTANCE** 環境変数によって指定した現在のインスタンスに対して実行されません。

## AUTOCONFIGURE

バッファ・プール・サイズ、データベース構成パラメーター、およびデータベース・マネージャー構成パラメーターの初期値を計算して表示します。オプションで、これらの報告値を適用するように指定できます。

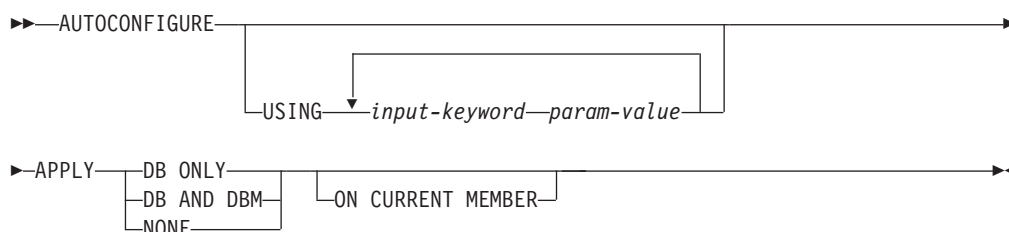
### 許可

SYSADM

### 必要な接続

データベース

### コマンド構文



### コマンド・パラメーター

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

表 8. 有効な入力キーワードおよびパラメーター値

キーワード	有効値	デフォルト値	説明
<b>mem_percent</b>	1 から 100	25	データベースに割り当てられているインスタンス・メモリーのパーセンテージ。ただし、 <b>CREATE DATABASE</b> コマンドによって構成アドバイザーが呼び出され、 <b>mem_percent</b> の値を指定しない場合、インスタンスとシステムのメモリー使用量に基づいてパーセンテージが計算され、最大でインスタンス・メモリーの 25% になります。
<b>workload_type</b>	simple、 mixed、 complex	mixed	単純 (simple) ワークロードは入出力集中の傾向があり大部分がトランザクションであるのに対し、複雑 (complex) ワークロードは CPU 集中の傾向があり大部分が照会です。
<b>num_stmts</b>	1 から 1,000,000	10	作業単位ごとのステートメント数
<b>tpm</b>	1 から 200,000	60	1 分ごとのトランザクション

表 8. 有効な入力キーワードおよびパラメーター値 (続き)

キーワード	有効値	デフォルト値	説明
<b>admin_priority</b>	performance、 recovery、 both	both	より良いパフォーマンス (分あたりのより多いトランザクション数) またはより良いリカバリー時間のための最適化
<b>is_populated</b>	yes、no	yes	データベースにデータを移植するかどうか
<b>num_local_apps</b>	0 から 5,000	0	接続されたローカル・アプリケーションの数
<b>num_remote_apps</b>	0 から 5,000	10	接続されたりリモート・アプリケーションの数
<b>isolation</b>	RR、RS、 CS、UR	RR	このデータベースに接続するアプリケーションの最大分離レベル (反復可能読み取り (RR)、読み取り固定 (RS)、カーソル固定 (CS)、非コミット読み取り (UR))。これは、他の構成パラメーターの値を決定するためにのみ使用されます。アプリケーションを特定の分離レベルに制限するよう設定されるものではなく、デフォルト値を使用するのが安全です。
<b>bp_resizeable</b>	yes、no	yes	バッファ・プールのサイズが変更可能かどうか。

**APPLY**

**DB ONLY**

現行のデータベース・マネージャーの構成に基づいて、データベース構成およびバッファ・プール設定の推奨値を表示します。データベース構成およびバッファ・プール設定に対して推奨される変更を適用します。

**DB AND DBM**

データベース・マネージャー構成、データベース構成、およびバッファ・プール設定に対して推奨される変更を、表示および適用します。

**NONE** 推奨される変更を表示しますが、適用はしません。

**ON CURRENT MEMBER**

デフォルトでは、パーティション・データベース環境または DB2 pureScale 環境において、構成アドバイザーによりすべてのメンバー上のデータベース構成が更新されます。**ON CURRENT MEMBER** オプションを指定すると、構成アドバイザーは接続から判別した現在のメンバーに対してメンバー・レベルの構成パラメーターを設定します。一方、グローバル・レベルの構成パラメーターはグローバル・レベルでのみ機能するように構成することができ、設定するとすべてのメンバーに影響を与えます。

バッファ・プールの変更事項は常にシステム・カタログに適用されます。したがってすべてのメンバーが影響を受けます。**ON CURRENT MEMBER** オプションは、バッファ・プールの推奨事項では無視されます。

## 使用上の注意

- このコマンドは、現在接続されているデータベースのための推奨構成を作成し、このデータベースがインスタンス上で唯一のアクティブ・データベースであると仮定します。セルフチューニング・メモリー・マネージャーを使用可能にしておらず、インスタンス上に複数のアクティブ・データベースがある場合、データベース・メモリー分散を反映する **mem\_percent** 値を指定してください。例えば、インスタンス・メモリーの 80% を使用しリソースを平等に共有する、2 つのアクティブ・データベースがインスタンスにある場合、40% (80% を 2 データベースで割る) を **mem\_percent** 値に指定します。
- 同じコンピューター上に複数のインスタンスがあり、セルフチューニング・メモリー・マネージャーが使用可能になっていない場合は、各インスタンスで **instance\_memory** に固定値を設定するか、データベース・メモリーの配分を反映する **mem\_percent** 値を指定してください。例えば、すべてのアクティブ・データベースがコンピューター・メモリーの 80% を使用し、それぞれ 1 つのデータベースを持つインスタンスが 4 つある場合は、20% (80% を 4 データベースで割る) を **mem\_percent** 値に指定します。
- **AUTOCONFIGURE** コマンドによって構成アドバイザーを明示的に呼び出す場合、**DB2\_ENABLE\_AUTOCONFIG\_DEFAULT** レジストリー変数の設定値は無視されます。
- データベースに対して **AUTOCONFIGURE** コマンドを実行すると、セルフチューニング・メモリー・マネージャーを有効にすることが推奨されます。ただし、**sheapthres** がゼロではないインスタンス内で、あるデータベースに対して **AUTOCONFIGURE** コマンドを実行した場合、ソート・メモリー・チューニング (**sortheap**) は自動的には有効になりません。ソート・メモリー・チューニング (**sortheap**) を有効にするには、**UPDATE DATABASE MANAGER CONFIGURATION** コマンドを使用して、**sheapthres** をゼロに設定する必要があります。 **sheapthres** の値を変更すると、これまでの既存データベース内のソート・メモリーの使用に影響を与える可能性があることに注意してください。

## 互換性

以前のバージョンとの互換性:

- **DB2\_ENFORCE\_MEMBER\_SYNTAX** レジストリー変数が ON に設定されている場合を除き、**NODE** および **DBPARTITIONNUM** を **MEMBER** の代わりに指定できます。

## BACKUP DATABASE

データベースまたは表スペースのバックアップ・コピーを作成します。

異なるさまざまなオペレーティング・システムおよびハードウェア・プラットフォームの間で DB2 データベース・システムによってサポートされるバックアップ操作の詳細は、『異なるオペレーティング・システムおよびハードウェア・プラットフォーム間のバックアップおよびリストア操作』を参照してください。

### 有効範囲

パーティション・データベース環境で、データベース・パーティションを指定しない場合、このコマンドはコマンドが実行されたデータベース・パーティションにのみ作用します。

パーティション・バックアップを実行するためのオプションが指定された場合、コマンドを呼び出すことができるのは、カタログ・データベース・パーティションに対してだけです。すべてのデータベース・パーティション・サーバーをバックアップするためのオプションが指定されているなら、それは `db2nodes.cfg` ファイルの中にリストされているすべてのデータベース・パーティション・サーバーに影響を与えます。そうでない場合は、コマンドで指定されたデータベース・パーティション・サーバーに作用します。

### 許可

以下の権限のいずれか。

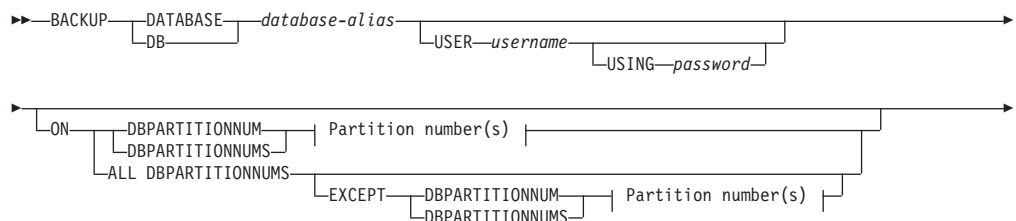
- SYSADM
- SYSCTRL
- SYSMANT

### 必要な接続

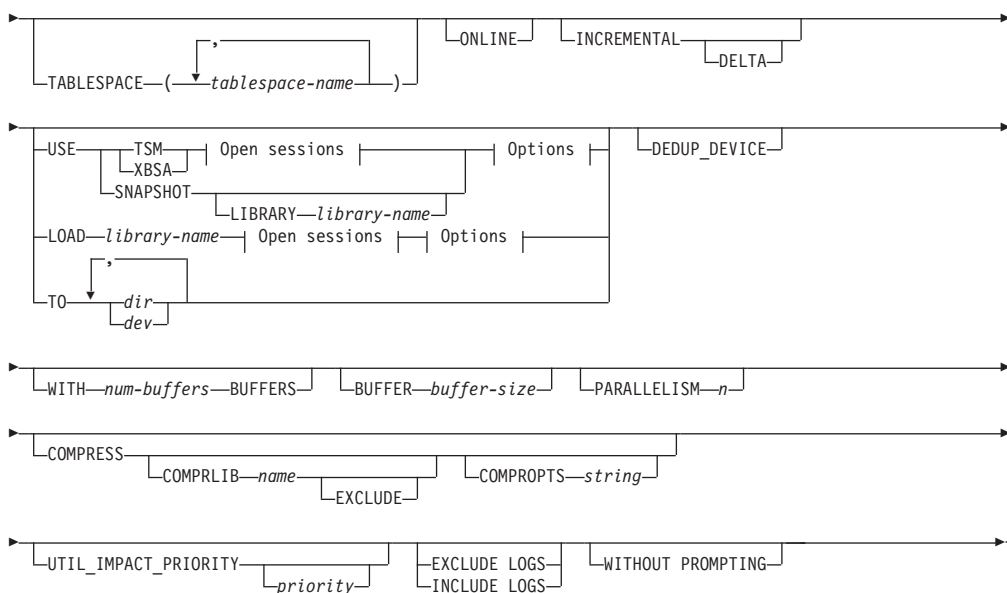
データベース。このコマンドは、指定されたデータベースへの排他接続を自動的に確立します。

**注:** 指定したデータベースへの接続が既に存在している場合、その接続は終了して、バックアップ操作のために専用の接続が新規に確立されます。接続は、バックアップ操作の完了時に終了します。

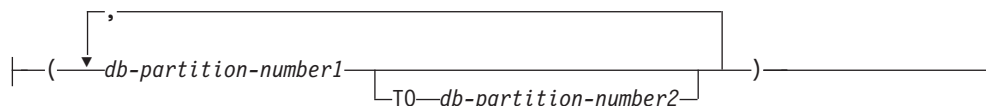
### コマンド構文



## BACKUP DATABASE



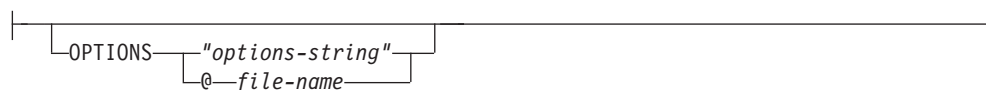
### Partition number(s):



### Open sessions:



### Options:



## コマンド・パラメーター

### DATABASE | DB *database-alias*

バックアップを取るデータベースの別名を指定します。

### USER *username*

データベースのバックアップを取るユーザー名を識別します。

### USING *password*

ユーザー名を認証するために使用するパスワード。パスワードを省略すると、ユーザーに入力を求めるプロンプトが出ます。

**ON** データベース・パーティションのセットに対してデータベースをバックアップします。

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

データベース・パーティション・リスト内のデータベース・パーティション番号を指定します。

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

データベース・パーティション番号の範囲を指定します。  
*db-partition-number1* から *db-partition-number2* までの間のすべてのパーティションがデータベース・パーティション・リストに含まれます。

**ALL DBPARTITIONNUMS**

*db2nodes.cfg* ファイルで指定されたすべてのパーティションでデータベースをバックアップすることを指定します。

**EXCEPT** *db2nodes.cfg* ファイルで指定されたパーティションのうち、データベース・パーティション・リストで指定されたパーティションを除くすべてのパーティションでデータベースをバックアップすることを指定します。

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

データベース・パーティション・リスト内のデータベース・パーティション番号を指定します。

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

*db-partition-number2*

データベース・パーティション番号の範囲を指定します。  
*db-partition-number1* から *db-partition-number2* までの間のすべてのパーティションがデータベース・パーティション・リストに含まれます。

**TABLESPACE** *tablespace-name*

バックアップを取る表スペースを指定するときに使用する名前のリスト。

**ONLINE**

オンライン・バックアップを指定します。デフォルトはオフライン・バックアップです。オンライン・バックアップは、**logarchmeth1** を有効にして構成されたデータベースにのみ使用できます。オンライン・バックアップでは、SMS 表スペース内の表が処理される際にそのようなすべての表に対して DB2 が IN (Intent None) ロックを取得します。S (共有ロック) は、オンライン・バックアップ中に SMS 表スペースで LOB データに対して保持されることはなくなりました。

**INCREMENTAL**

累積 (増分) バックアップ・イメージを指定します。増分バックアップ・イメージとは、正常に実行されたフルバックアップ操作のうち最新のものが実行されて以来変更された、すべてのデータベース・データのコピーです。

**DELTA** 非累積 (差分) バックアップ・イメージを指定します。差分バックアップ・イメージとは、正常に実行された任意のタイプのバックアップ操作のうち最新のものが実行されて以来変更された、すべてのデータベース・データのコピーです。

**USE**

- TSM** バックアップで使用するターゲット・デバイスとして Tivoli® Storage Manager (TSM) を指定します。
- XBSA** XBSA インターフェースを使用するように指定します。バックアップ・サービス API (XBSA) は、バックアップやアーカイブの目的でデータ・ストレージ管理を必要とするアプリケーションまたは機能のための、オープン・アプリケーション・プログラミング・インターフェースです。

### SNAPSHOT

スナップショット・バックアップを取ることを指定します。

**SNAPSHOT** パラメーターを以下のいずれかのパラメーターと一緒に使用することはできません。

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

スナップショット・バックアップのデフォルトの動作は、データベースを構成するすべてのパスの完全データベース・オフライン・バックアップです。つまり、すべてのコンテナ、ローカル・ボリューム・ディレクトリー、データベース・パス (**DBPATH**)、1 次ログとミラー・ログのパスが含まれます (**EXCLUDE LOGS** が明示的に記述されない限り、**INCLUDE LOGS** がすべてのスナップショット・バックアップのデフォルトになります)。

### **LIBRARY** *library-name*

IBM DB2 Server には、以下のストレージ・ハードウェアのための DB2 ACS API ドライバーが組み込まれています。

- IBM TotalStorage SAN ボリューム・コントローラー
- IBM Enterprise Storage Server® Model 800
- IBM Storwize® V7000
- IBM System Storage® DS6000™
- IBM System Storage DS8000®
- IBM System Storage N Series
- IBM XIV®

他のストレージ・ハードウェアを使用していて、そのストレージ・ハードウェア用の DB2 ACS API ドライバーがある場合、**LIBRARY** パラメーターを使用してその DB2 ACS API ドライバーを指定できます。

**LIBRARY** パラメーターの値は、完全修飾ライブラリー・ファイル名です。



## OPTIONS

*"options-string"*

バックアップ操作で使用するオプションを指定します。ストリングは、二重引用符なしで、入力されたとおりに渡されます。

*@ file-name*

バックアップ操作で使用するオプションが、DB2 サーバー上のファイルに含まれていることを指定します。このストリングは、ベンダー・サポートのライブラリーに渡されます。ファイル名は完全修飾ファイル名でなければなりません。

**VENDOROPT** データベース構成パラメーターを使用して、スナップショット・バックアップ操作のベンダー固有オプションを指定することはできません。これには、バックアップ・ユーティリティーの **OPTIONS** パラメーターを使用する必要があります。

**OPEN num-sessions SESSIONS**

DB2 製品と TSM 製品または他のバックアップ・ベンダー製品との間に作成する入出力セッションの数。このパラメーターは、テープ、ディスク、または他のローカル装置にバックアップする場合には効果はありません。オンライン・バックアップで **INCLUDE LOGS** パラメーターを指定した場合、最初のセッションがクローズされた後に、**OPEN num-sessions SESSIONS** パラメーターに対して追加のセッションが 1 つ作成されます。Single System View (SSV) オンライン・バックアップを作成する場合、各ノード・バックアップで、最初のセッションがクローズされた後に **OPEN num-sessions SESSIONS** パラメーターに対して追加のセッションが 1 つ作成されます。TSM オプションでこのパラメーターを使用すると、履歴ファイルに作成される項目の数が、作成されるセッションの数と等しくなります。

**TO dir | dev**

ディレクトリーまたはテープ装置名のリストです。ディレクトリーが常駐する絶対パスを指定しなければなりません。**USE TSM**、**TO**、および **LOAD** を省略すると、バックアップ・イメージのデフォルト・ターゲット・ディレクトリーは、クライアント・コンピューターの現行作業ディレクトリーになります。このターゲット・ディレクトリーまたは装置は、データベース・サーバー上に存在している必要があります。

パーティション・データベースでは、すべてのデータベース・パーティションにターゲット・ディレクトリーまたは装置が存在している必要があります。オプションで共有パスにすることができます。ディレクトリー名や装置名は、データベース・パーティション式を使用して指定できます。データベース・パーティション式について詳しくは、『自動ストレージ・データベース』を参照してください。

このパラメーターは、バックアップ・イメージが複数の宛先ディレクトリーや装置にわたる場合に、それらを指定するために繰り返すことができます。宛先が複数指定されている場合 (例えば、宛先 1、宛先 2、および宛先 3)、宛先 1 が最初にオープンされます。メディア・ヘッダーおよび特殊ファイル (構成ファイル、表スペース表、および履歴ファイルを含む) は、宛先 1 にあります。他の残りの宛先は、オープンされており、これらはバックアップ操作のときに並列で使用されます。Windows オペレーティング・システ

ムの場合、汎用テープ装置はサポートされていないので、テープ装置のタイプごとに固有のデバイス・ドライバーが必要です。

テープ装置やフロッピー・ディスクを使用することにより、メッセージやユーザー入力のプロンプトを生成できます。有効な応答オプションは、次のとおりです。

- c** 続行。警告メッセージを生成した装置の使用を続けます (例えば、新しいテープをマウントしたときなど)。
- d** 装置の終了。警告メッセージの原因となった装置の使用だけを停止します (例えば、これ以上テープがない場合など)。
- t** 終了。バックアップ操作を打ち切ります。

テープ・システムでバックアップ・イメージを固有に参照する機能をサポートしていない場合は、同じテープに同じデータベースの複数のバックアップ・コピーは作成しないことをお勧めします。

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

使用するバックアップおよびリストア I/O ベンダー関数を含む共有ライブラリー (Windows オペレーティング・システムでは DLL) の名前。絶対パスで指定することができます。絶対パスを指定していない場合、デフォルトはユーザー出口プログラムが常駐しているパスになります。

### **DEDUP\_DEVICE**

データ非重複化をサポートするターゲット・ストレージ・デバイス用にバックアップ・イメージのフォーマットを最適化します。

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

使用するバッファの数です。指定するバッファの数が足りなくて正常にバックアップを作成できない場合、このパラメーターでは、バックアップを完了するために必要な最小値が自動的に選択されます。複数の場所にバックアップする場合は、パフォーマンスを改善するため、より大きな数値をバッファ数に指定することもできます。 **COMPRESS** パラメーターを指定する場合は、パフォーマンス改善に役立てるため、**PARALLELISM** パラメーターに指定する各表スペース用の追加のバッファを追加できます。

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

4 KB ページごとの単位で表した、バックアップ・イメージを作成する際に使用するバッファのサイズ。値を明示的に指定しない場合、DB2 はこのパラメーターの最適値を自動的に選択します。このパラメーターの最小値は 8 ページです。

さまざまなブロック・サイズのテープを使用する場合は、磁気テープ装置がサポートする範囲内にバッファ・サイズを削減してください。この範囲内でないと、バックアップ操作は正常に実行されることもありますが、作成されたイメージはリカバリー不能になることがあります。

Linux のほとんどのバージョンでは、SCSI テープ装置でバックアップ操作を行うときに、DB2 に用意されたデフォルトのバッファ・サイズを使用すると、エラー SQL2025N、理由コード 75 が表示されます。Linux 内部 SCSI バッファがオーバーフローするのを防ぐには、以下の公式を使用してください。

```
bufferpages <= ST_MAX_BUFFERS * ST_BUFFER_BLOCKS / 4
```

*bufferpages* は **BUFFER** パラメーターと共に使用する値であり、**ST\_MAX\_BUFFERS** と **ST\_BUFFER\_BLOCKS** は `drivers/scsi` ディレクトリー中の Linux カーネルで定義されています。

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

バックアップ・ユーティリティーによって同時に読み取り可能な表スペースの数を決定します。値を明示的に指定しない場合、DB2 はこのパラメーターの最適値を自動的に選択します。

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

バックアップを、指定した優先順位によりスロットル・モードで実行することを指定します。スロットル・モードでは、バックアップ操作によるパフォーマンスの影響を調整できます。優先順位 (*priority*) は 1 から 100 までの範囲の任意の数であり、1 が優先順位最低、100 が優先順位最高を意味します。優先順位の値なしで **UTIL\_IMPACT\_PRIORITY** キーワードが指定された場合は、デフォルトの優先順位 50 でバックアップが実行されます。

**UTIL\_IMPACT\_PRIORITY** を指定しない場合、バックアップは非スロットル・モードで実行されます。バックアップをスロットル・モードで実行するためには、**util\_impact\_lim** 構成パラメーターを設定することによって影響ポリシーが定義されていなければなりません。

#### **COMPRESS**

バックアップを圧縮することを指定します。

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

圧縮を実行するために使用するライブラリーの名前を指定します (例えば、Windows の場合は `db2compr.dll`、Linux および UNIX オペレーティング・システムの場合は `libdb2compr.so`)。この名前は、サーバー上の 1 個のファイルを参照する完全修飾パスでなければなりません。このパラメーターを指定しない場合、デフォルトの DB2 圧縮ライブラリーが使用されます。指定されたライブラリーをロードできない場合、バックアップは失敗します。

#### **EXCLUDE**

圧縮ライブラリーをバックアップ・イメージに格納しないことを指定します。

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

バイナリー・データのうち、圧縮ライブラリーの初期設定ルーチンに渡すブロックを記述します。DB2 はこのストリングをクライアントからサーバーに直接渡すため、バイト反転やコード・ページ変換の問題がある場合は圧縮ライブラリーで処理する必要があります。データ・ブロックの最初の文字が「@」である場合、DB2 は、データの残りの部分をサーバー上に存在するファイルの名前と解釈します。その場合 DB2 は、*string* の内容をそのファイルの内容で置き換え、そのようにして得られる新しい値を初期設定ルーチンに渡します。*string* の最大長は 1024 バイトです。

#### **EXCLUDE LOGS**

バックアップ・イメージにログ・ファイルをまったく含めないことを指定します。オフライン・バックアップ操作の実行の場合、このオプションが指定

されていてもいなくても、ログは除外されます (ただし、スナップショット・バックアップは例外です)。ログは、以下のバックアップ・シナリオにおいて、デフォルトで除外されます。

- 単一パーティション・データベースのオフライン・バックアップ。
- 単一システム・ビュー・バックアップを使用しない場合は、マルチ・パーティション・データベースのオンラインまたはオフラインのバックアップ。

スナップショット・バックアップで **EXCLUDE LOGS** を指定すると、バックアップ中のログ・ファイルへの書き込みが可能になります。これらのログ・ファイルは、デフォルトでスナップショット・バックアップに含まれますが、リカバリーには使用できません。このバックアップをリストアした場合、バックアップからログ・ファイルを抽出してはなりません。バックアップが取られたときにログ・パスがデフォルトに設定されていた場合は、ログ・ファイルをリストアから除外することはできず、バックアップのリストア後にこれらのログ・ファイルを手動で削除する必要があります。ログ・パスがデフォルトではなかった場合、**RESTORE DATABASE** コマンドに **LOGTARGET EXCLUDE** オプションを指定して使用することで、ログ・ファイルをリストア時に除外することができます。

### INCLUDE LOGS

ログ・ファイルのうち、特定の整合ポイント・イン・タイムまでこのイメージをリストアおよびロールフォワードするために必要な範囲をバックアップ・イメージに含めることを指定します。オフライン・バックアップの場合、このオプションは無効です。ただし、スナップショット・バックアップの場合は例外です。**INCLUDE LOGS** は、各データベース・パーティションが独立してバックアップされる (つまり非シングル・システム・ビュー・バックアップ) 複数パーティション・オンライン・バックアップの場合を除き、すべてのオンライン・バックアップ操作において常にデフォルト・オプションです。

バックアップで必要とされるいずれかのログ・ファイルが既にバックアップされていて、ログ・パスにはもうない場合、オーバーフロー・ログ・パスが既に設定されているのであれば、DB2 データベース・マネージャーはそのパスからバックアップ用にログ・ファイルを取り出します。その他の場合には、データベース・マネージャーは現在のログ・パスまたはミラー・ログ・パスから、バックアップ用のログ・ファイルを取り出します。これらのログ・ファイルは、バックアップ完了後にログ・パスから削除されます。

### WITHOUT PROMPTING

バックアップは、管理されることなく実行されるため、通常はユーザーの介入を必要とするアクションでエラー・メッセージが戻されるように指定されます。

#### 注:

1. バックアップ・コマンドで、パーティション・データベース内のどのパーティションのバックアップを作成するのかを指定した場合は、暗黙的に **WITHOUT PROMPTING** でバックアップ操作が実行されます。
2. パーティション・データベース環境では、**BACKUP DATABASE** コマンドで指定されたオプションは、各パーティションに個別に適用されます。例

例えば、バックアップ操作が **USE TSM OPEN 3 SESSIONS** に指定されている場合、DB2 は各パーティションで 3 つの TSM セッションを開きます。

## 例

1. 以下の例で、データベース WSDB は 0 から 3 までの番号が付けられた 4 つのデータベース・パーティションすべてに定義されています。パス `/dev3/backup` はすべてのデータベース・パーティションからアクセスできます。データベース・パーティション 0 はカタログ・パーティションです。すべての WSDB データベース・パーティションの `/dev3/backup` へのオフライン・バックアップを実行する場合は、データベース・パーティション 0 から以下のコマンドを発行します。

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

バックアップは、すべてのパーティションで同時に実行されます。4 つのデータベース・パーティションのバックアップ・イメージがすべて `/dev3/backup` ディレクトリーに格納されます。この格納先としては、複数のパーティションからアクセスできる共有ディレクトリーを使用することも、各パーティションから個別にアクセスできるローカル・マウント・ディレクトリーを使用することも、その両方を組み合わせて使用することもできます。

2. 以下の例で、データベース SAMPLE は TSM サーバーに 2 つの並行 TSM クライアント・セッションを使用してバックアップされます。この環境に最適なバッファ・サイズが DB2 によって計算されます。

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

3. 次の例では、データベース payroll の表スペース (syscatspace、userspace1) の表スペース・レベル・バックアップがテープに対して実行されます。

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

4. バックアップ操作で使用する TSM 情報を指定するには、**USE TSM OPTIONS** キーワードを使用します。次の例は、**USE TSM OPTIONS** キーワードを使用して、完全修飾ファイル名を指定する方法を示すものです。

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

ファイル `myoptions.txt` には、`-fromnode=bar -fromowner=dmcinnis` という文字列が含まれています。

5. 以下は、リカバリー可能データベース用の週次の増分バックアップ・ストラテジーのサンプルです。週 1 回の全データベース・バックアップ操作、1 日 1 回の非累積 (差分) バックアップ操作、および週 2 回の累積 (増分) バックアップ操作が含まれています。

```
(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
```



## BACKUP DATABASE

6. 次の例では、データベース SAMPLE のバックアップ操作のために、同一のターゲット・ディレクトリーを 3 回指定しています。ターゲット・ファイル・システムが複数の物理ディスクで構成されている場合には、この方法を使用することができるでしょう。

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

データは 3 つのターゲット・ディレクトリーに並行してバックアップされ、それら 3 つのバックアップ・イメージは拡張子 .001、.002、および .003 が付けられて生成されます。

7. 以下の例で、データベース WSDB は 0 から 3 までの番号が付けられた 4 つのデータベース・パーティションすべてに定義されています。データベース・パーティション 0 はカタログ・パーティションです。データベース・パーティション 1 および 2 で表スペース USERSPACE1 のオンライン・バックアップを実行し、そのバックアップ・イメージを TSM サーバーに保管する場合は、パーティション 0 から以下のコマンドを発行します。

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

8. 生成されたサンプル出力は、各パーティションから戻された sqlcode を示しています。

**例 1** 全パーティションが成功 (sqlcode >= 0)

```
$ db2 backup db foo on all dbpartitionnums tablespace(T1)
Part Result
----
0000 DB20000I The BACKUP DATABASE command completed successfully.
0001 SQL2430W The database backup succeeded, but the following
           table spaces do not exist on this database partition: "T1".

Backup successful. The timestamp for this backup image is :
                               20040908010203
```

**例 2** 1 つ以上のパーティションが失敗 (sqlcode < 0)

```
$ db2 backup db foo on all dbpartitionnums to /backups
Part Result
----
0000 SQL2001N The utility was interrupted. The output data may be
           incomplete.
0001 SQL2419N The target disk "/backups" has become full.

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

9. 以下のバックアップでは、作成されるイメージ内にログ・ディレクトリーが含まれます。

```
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. 以下のバックアップでは、作成されるイメージ内にログ・ディレクトリーが含まれません。

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

## 使用上の注意

- バックアップ内のデータは、データベース・サーバーによって保護されるわけではありません。バックアップに LBAC で保護されたデータが含まれる場合は特に、バックアップを適切に保護しておく必要があります。
- テープへのバックアップの場合、現在、可変ブロック・サイズの使用はサポートされていません。そのオプションを使用する必要がある場合は、リカバリーが正常に実行されるように十分にテストしたプロシージャが使用できるようになっていることを確認し、また可変ブロック・サイズを指定して作成されたバックアップ・イメージを使用してください。
- 可変ブロック・サイズを使用する場合、使用している磁気テープ装置の最大限度以下のバックアップ・バッファ・サイズを指定する必要があります。パフォーマンスを最適化するには、使用している装置のブロック・サイズの最大限度と等しい値をバッファ・サイズとして使用しなければなりません。
- スナップショット・バックアップは、ファイラー・システムまたはストレージ・システムで障害が発生した場合に備えて、通常のディスク・バックアップで補完する必要があります。
- 定期的にデータベースをバックアップしていくと、非常に大きなデータベース・バックアップ・イメージ、多くのデータベース・ログ、およびロード・コピー・イメージが累積する場合があります、これらすべてが大量のディスク・スペースを占めることがあります。これらのリカバリー・オブジェクトの管理方法については、『リカバリー・オブジェクトの管理』を参照してください。
- プロキシ・ノードをサポートする TSM 環境では、**OPTIONS** パラメーターを使用してバックアップ操作を使用可能にすることができます。詳しくは、『Tivoli Storage Manager クライアントの構成』のトピックを参照してください。
- **DB2\_BCKP\_PAGE\_VALIDATION** レジストリー変数を使用して、バックアップ中の DMS および AS ページの妥当性検査を有効にすることができます。
- **DB2\_BCKP\_INCLUDE\_LOGS\_WARNING** レジストリー変数を使用すると、一部のオンライン・バックアップは必要なログすべてが正常に組み込まれていなくても成功できるということを指定できます。

## BIND

バインド・ユーティリティーを呼び出し、プリコンパイラーによって生成されたバインド・ファイルに保管された SQL ステートメントを作成します。また、データベースに保管されるパッケージを作成します。

### 有効範囲

このコマンドは、db2nodes.cfg 中のどのデータベース・パーティションからでも発行できます。実行すると、カタログ・データベース・パーティションのデータベース・カタログが更新されます。その影響はすべてのデータベース・パーティションから見えます。

### 許可

以下のいずれかの権限が必要です。

- DBADM 権限
- EXPLAIN ONLY が指定されている場合、EXPLAIN 権限または暗黙的に EXPLAIN を含む権限があれば問題ありません。
- パッケージが存在しない場合は、BINDADD 権限および以下のものがが必要です。
  - パッケージのスキーマ名が存在しない場合は、データベースに対する IMPLICIT\_SCHEMA 権限。
  - パッケージのスキーマ名が存在する場合は、スキーマに対する CREATEIN 特権。
- パッケージが存在する場合は、以下のいずれかの特権です。
  - スキーマに対する ALTERIN 特権
  - パッケージに対する BIND 特権

さらに、EXPLAIN または EXPLSNAP 節を使用して Explain 情報をキャプチャーする場合は、以下のいずれかの権限が必要です。

- Explain 表に対する INSERT 特権
- DATAACCESS 権限

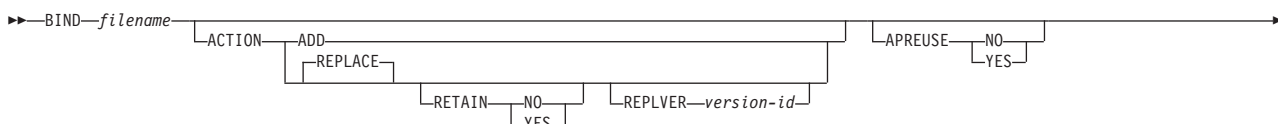
アプリケーション内の静的 SQL ステートメントをコンパイルするために必要な特権もすべて必要です。グループに認可された特権が、静的ステートメントの許可の検査に使用されることはありません。

### 必要な接続

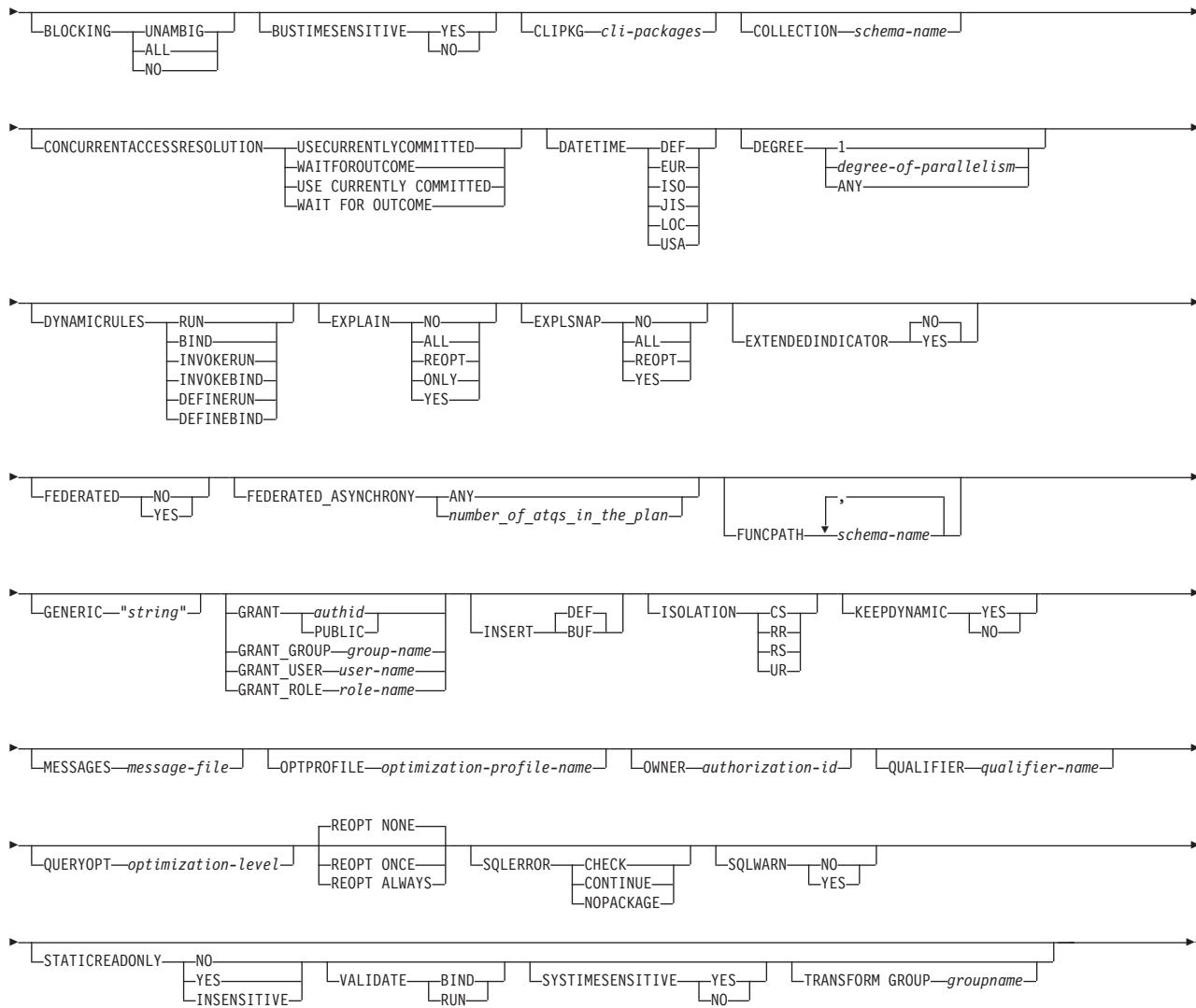
データベース。暗黙接続が可能な場合には、デフォルト・データベースへの接続が確立されます。

### コマンド構文

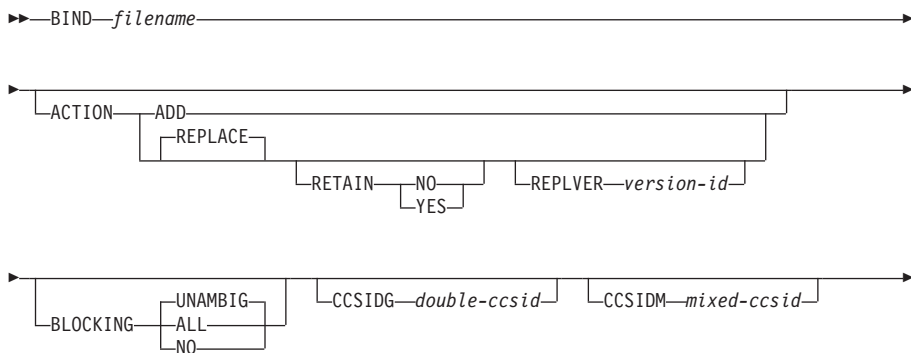
#### DB2 for Linux, Windows and UNIX の場合



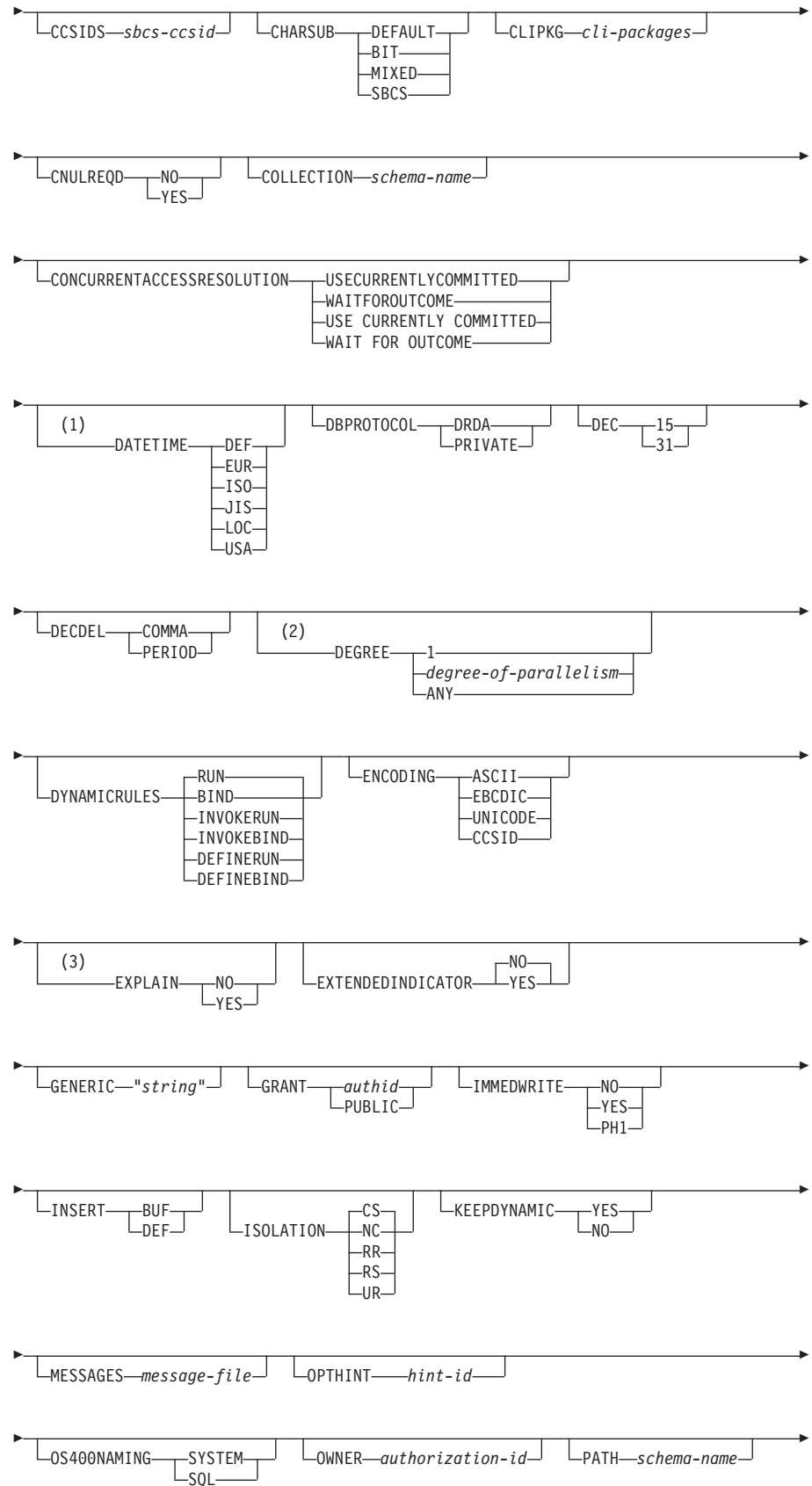


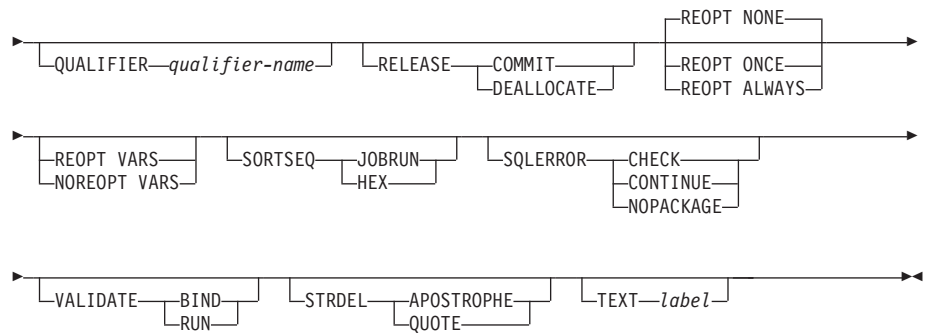


Linux、Windows、UNIX 以外のサーバー上の DB2 の場合



# BIND





注:

- 1 サーバーが DATETIME DEF オプションをサポートしない場合、それは DATETIME ISO にマップされます。
- 2 DEGREE オプションは DRDA® レベル 2 のアプリケーション・サーバーでしかサポートされていません。
- 3 DRDA は、EXPLAIN オプションが値 YES または NO を持つように定義します。サーバーが EXPLAIN YES オプションをサポートしない場合、この値は EXPLAIN ALL にマップされます。

コマンド・パラメーター

*filename*

アプリケーション・プログラムをプリコンパイルしたときに生成されたバインド・ファイル、または複数のバインド・ファイル名を含むリスト・ファイルの名前を指定します。バインド・ファイルの拡張子は .bnd です。また、絶対パス名も指定できます。

リスト・ファイルを指定した場合、その名前の先頭文字は @ 文字でなければなりません。リスト・ファイルには、数行のバインド・ファイル名を含めることができます。同一行にリストするバインド・ファイルはプラス (+) 文字で区切る必要がありますが、各行の先頭ファイルの前や最後のファイルの後に + を挿入することはできません。例えば、次のようになります。

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

これは、次のバインド・ファイルを含むリスト・ファイルです。

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

**ACTION**

パッケージを追加または置換できるかどうかを示します。

**ADD** 名前付きパッケージが存在せず、新規パッケージを作成することを指示します。既にパッケージがある場合は、実行停止状態となり、診断エラー・メッセージが戻されます。

**REPLACE**

既存のパッケージを、パッケージ名および作成者が同じ新規パッケージと置き換えることを指示します。これは **ACTION** オプションのデフォルト値です。

**RETAIN**

パッケージを置き換えたときに **BIND** および **EXECUTE** 権限が保持されるかどうかを指示します。パッケージの所有権を変更した場合、新規所有者は前のパッケージ所有者に **BIND** 権限と **EXECUTE** 権限を付与します。

**NO** パッケージを置き換えたとき、**BIND** および **EXECUTE** 権限を保持しません。この値は **DB2** ではサポートされていません。

**YES** パッケージを置き換えたとき、**BIND** および **EXECUTE** 権限を保持します。これはデフォルト値です。

**REPLVER** *version-id*

特定のバージョンのパッケージを置き換えます。バージョン **ID** は、どのバージョンのパッケージを置き換えるのかを指定するものです。指定されたバージョンが存在しない場合には、エラーが戻されます。**REPLACE** の **REPLVER** オプションが指定されておらず、バインドされるパッケージのパッケージ名、作成者、およびバージョンと一致するパッケージが既に存在する場合、そのパッケージは置換されます。存在しなければ、新規のパッケージが追加されます。

**APREUSE**

静的 **SQL** アクセス・プランが再使用されるかどうかを指定します。このオプションが有効である場合、照会コンパイラーは、バインド時および将来の暗黙および明示の再バインド時に、既存のパッケージ内のステートメントにアクセス・プランを再使用しようとしています。

**YES** 照会コンパイラーは、パッケージ内のステートメントにアクセス・プランを再使用しようとしています。既存のパッケージがある場合、照会コンパイラーは、新規バインド・ファイル内のステートメントと合致できるすべてのステートメントに対して、アクセス・プランを再使用しようとしています。ステートメントが合致するには、ステートメント・テキストは同一でなければならない、既存のパッケージ内のステートメントのセクション番号は、新規パッケージ内のステートメントに対するセクション番号の内容と一致しなければなりません。

**NO** 照会コンパイラーは、パッケージ内のステートメントにアクセス・プランを再使用しようとしません。これはデフォルト設定です。

**BLOCKING**

カーソルの行ブロッキングのタイプを指定します。パーティション・データベース環境では、**LOB** 列データ・タイプ参照を含んだ行データのブロッキングもサポートされます。

**ALL** **FOR READ ONLY** 節で指定されているカーソルまたは **FOR UPDATE** として指定されていないカーソルの場合、ブロック化が発生します。

未確定カーソルは、読み取り専用として扱われます。

**NO** どのカーソルについてもブロック化は発生しません。  
読み取り専用カーソルおよび未確定カーソルの定義については、*DECLARE CURSOR* ステートメントを参照してください。  
あいまいなカーソルは、更新可能として扱われます。

**UNAMBIG**

FOR READ ONLY 節で指定されているカーソルの場合、ブロック化が発生します。

FOR READ ONLY または FOR UPDATE 節で宣言されていない、未確定でない読み取り専用のカーソルはブロック化されます。未確定カーソルはブロック化されません。

あいまいなカーソルは、更新可能として扱われます。

**BUSTIMESENSITIVE**

静的および動的 SQL ステートメントでのアプリケーション期間のテンポラル表への参照が、CURRENT TEMPORAL BUSINESS\_TIME 特殊レジスタの値の影響を受けるかどうかを示します。

**YES** アプリケーション期間のテンポラル表への参照は、CURRENT TEMPORAL BUSINESS\_TIME 特殊レジスタの値の影響を受けます。この値はデフォルト値です。

**NO** アプリケーション期間のテンポラル表への参照は、CURRENT TEMPORAL BUSINESS\_TIME 特殊レジスタの値の影響を受けません。

**CCSIDG** *double-ccsid*

CREATE および ALTER TABLE SQL ステートメントの文字カラム定義で、2 バイト文字用のコード化文字セット ID (CCSID) (特定の CCSID 節は使用しない) を指定する整数。このオプションは、DB2 for Linux, UNIX, and Windows 用のサーバーではサポートされません。このオプションを指定しないと、DRDA サーバーは、組み込みのデフォルト値を使用します。

**CCSIDM** *mixed-ccsid*

CREATE および ALTER TABLE SQL ステートメントの文字カラム定義で、混合バイト文字用のコード化文字セット ID (CCSID) (特定の CCSID 節は使用しない) を指定する整数。このオプションは、DB2 for Linux, UNIX, and Windows 用のサーバーではサポートされません。このオプションを指定しないと、DRDA サーバーは、組み込みのデフォルト値を使用します。

**CCSIDS** *sbc-ccsid*

CREATE および ALTER TABLE SQL ステートメントの文字カラム定義で、1 バイト文字用のコード化文字セット ID (CCSID) (特定の CCSID 節は使用しない) を指定する整数。なお、DB2 for Linux, UNIX, and Windows はこのオプションをサポートしません。このオプションを指定しないと、DRDA サーバーは、組み込みのデフォルト値を使用します。

**CHARSUB**

CREATE および ALTER TABLE SQL ステートメントの列定義に使用す

る、デフォルトの文字サブタイプを指定します。このプリコンパイル/バインド・オプションは、DB2 for Linux, UNIX, and Windows 用のサーバーではサポートされません。

**BIT** 明示的にサブタイプを指定しなかった場合、すべての新規文字カラムに FOR BIT DATA SQL 文字サブタイプが使用されます。

**DEFAULT**

明示的にサブタイプを指定しなかった場合、すべての新規文字カラムにターゲットの組み込みデフォルト・サブタイプが使用されます。

**MIXED**

明示的にサブタイプを指定しなかった場合、すべての新規文字カラムに FOR MIXED DATA SQL 文字サブタイプが使用されます。

**SBCS** 明示的にサブタイプを指定しなかった場合、すべての新規文字カラムに FOR SBCS DATA SQL 文字サブタイプが使用されます。

**CLIPKG** *cli-packages*

3 から 30 の整数で、CLI バインド・ファイルをデータベースに対してバインドするときに作成される、CLI ラージ・パッケージの数を指定します。

**CNULREQD**

このオプションは、**LANGLEVEL** プリコンパイル・オプションと関連します。これは、C または C++ アプリケーションで作成されたバインド・ファイルの場合のみ有効です。このバインド・オプションは、DB2 for Linux, UNIX, and Windows 用のサーバーではサポートされません。

**NO** C ストリング・ホスト変数中の NULL 終止符に関して、**LANGLEVEL SAA1** プリコンパイル・オプションに基づいてアプリケーションがコード化された場合です。

**YES** C ストリング・ホスト変数中の NULL 終止符に関して、**LANGLEVEL MIA** プリコンパイル・オプションに基づいてアプリケーションがコード化された場合です。

**COLLECTION** *schema-name*

パッケージ用の 128 バイトの収集 ID を指定します。これを指定しなかった場合、パッケージを処理する際には、ユーザーの許可 ID が使用されます。

**CONCURRENTACCESSRESOLUTION**

パッケージ内のステートメントに使用する並行アクセス解決を指定します。

**USE CURRENTLY COMMITTED**

データベース・マネージャーが更新または削除の処理中である場合に、該当するスキャンに現在コミット済みバージョンのデータを使用できることを指定します。挿入処理中の行はスキップできます。この節は、有効な分離レベルがカーソル固定または読み取り固定である場合に適用され (読み取り固定の場合、それはコミットされていない挿入だけをスキップします)、そうでない場合は無視されません。該当するスキャンには、非読み取り専用ステートメントに加え、読み取り専用ステートメントの一部にできる読み取り専用スキャンが含まれます。レジストリー変数 **DB2\_EVALUNCOMMITTED**、

**DB2\_SKIPDELETED**、および **DB2\_SKIPINSERTED** の設定は、Currently Committed を使用するスキャンには適用されません。ただし、これらのレジストリー変数の設定は、Currently Committed を使用しないスキャンには、今までどおり適用されます。

#### WAIT FOR OUTCOME

カーソル固定およびさらに上位のスキャンで、更新処理中または削除処理中のデータが検出されたときにはコミットまたはロールバックを待機するように指定します。挿入処理中の行はスキップされません。レジストリー変数 **DB2\_EVALUNCOMMITTED**、**DB2\_SKIPDELETED**、および **DB2\_SKIPINSERTED** の設定は、適用されなくなりました。

#### DATETIME

使用する日時形式を指定します。

- DEF** データベースのテリトリー・コードと対応する日時形式を使用します。
- EUR** IBM 欧州規格の日時形式を使用します。
- ISO** 国際標準化機構規格の日時形式を使用します。
- JIS** 日本工業規格の日時形式を使用します。
- LOC** データベースのテリトリー・コードと対応する地域別日時形式を使用します。
- USA** IBM 米国規格の日時形式を使用します。

#### DBPROTOCOL

3 部構成の名前のステートメントによって識別されるリモート・サイトに接続するときに使用するプロトコルを指定します。サポートしているのは DB2 for OS/390® だけです。サポートされているオプション値のリストについては、DB2 for OS/390 の資料を参照してください。

- DEC** 10 進算術演算に使用する最大精度を指定します。このプリコンパイル/バインド・オプションは、DB2 for Linux, UNIX, and Windows 用のサーバーではサポートされません。このオプションを指定しないと、DRDA サーバーは、組み込みのデフォルト値を使用します。
- 15** 10 進算術演算に 15 桁精度が使用されます。
- 31** 10 進算術演算に 31 桁精度が使用されます。

#### DECDEL

10 進数および浮動小数点リテラル中で小数点標識としてピリオド (.) またはコンマ (,) のどちらかを指定します。このプリコンパイル/バインド・オプションは、DB2 for Linux, UNIX, and Windows 用のサーバーではサポートされません。このオプションを指定しないと、DRDA サーバーは、組み込みのデフォルト値を使用します。

#### COMMA

小数点標識としてコンマ (,) を使用します。

#### PERIOD

小数点標識としてピリオド (.) を使用します。



**DEGREE**

SMP システムで静的 SQL ステートメントを実行するための並列処理の度合いを指定します。このオプションは、CREATE INDEX 並列処理には影響を与えません。

**1** ステートメントの実行に並列処理を使用しません。

*degree-of-parallelism*

ステートメントを実行する際の並列処理の度合いを指定します。値の範囲は 2 から 32 767 です。

**ANY** ステートメントの実行時にデータベース・マネージャーで判別した程度で並列処理を行うよう指定します。

**DYNAMICRULES**

許可 ID に使用される値の初期設定、および非修飾オブジェクト参照の暗黙的な修飾の、実行時の動的 SQL に適用される規則を定義します。

**RUN** パッケージを実行するユーザーの許可 ID が動的 SQL ステートメントの許可検査に使用されるように指定します。許可 ID は、動的 SQL ステートメント内の非修飾オブジェクト参照を暗黙的に修飾するためのデフォルトのパッケージ修飾子としても使用されます。これはデフォルト値です。

**BIND** 許可および修飾の静的 SQL に適用されるすべての規則が、実行時に使用されるように指定します。つまり、パッケージ所有者の許可 ID が動的 SQL の許可検査に使用され、デフォルトのパッケージ修飾子が動的 SQL ステートメント内の非修飾オブジェクト参照の暗黙的な修飾に使用されます。

**DEFINERUN**

パッケージがルーチン・コンテキスト内で使用される場合、ルーチン定義者の許可 ID が許可検査およびルーチン内の動的 SQL ステートメント内の非修飾オブジェクト参照の暗黙的な修飾に使用されます。

パッケージがスタンドアロン・アプリケーションとして使用される場合、動的 SQL ステートメントはパッケージが **DYNAMICRULES RUN** を指定してバインドされたかのように処理されます。

**DEFINEBIND**

パッケージがルーチン・コンテキスト内で使用される場合、ルーチン定義者の許可 ID が許可検査およびルーチン内の動的 SQL ステートメント内の非修飾オブジェクト参照の暗黙的な修飾に使用されます。

パッケージがスタンドアロン・アプリケーションとして使用される場合、動的 SQL ステートメントはパッケージが **DYNAMICRULES BIND** を指定してバインドされたかのように処理されます。

**INVOKERUN**

パッケージがルーチン・コンテキスト内で使用される場合、ルーチン起動時に有効だった現行のステートメント許可 ID が、動的 SQL



ステートメントの許可検査およびそのルーチン内の動的 SQL ステートメント内の非修飾オブジェクト参照の暗黙的な修飾に使用されます。

パッケージがスタンドアロン・アプリケーションとして使用される場合、動的 SQL ステートメントはパッケージが **DYNAMICRULES RUN** を指定してバインドされたかのように処理されます。

### INVOKEBIND

パッケージがルーチン・コンテキスト内で使用される場合、ルーチン起動時に有効だった現行のステートメント許可 ID が、動的 SQL ステートメントの許可検査およびそのルーチン内の動的 SQL ステートメント内の非修飾オブジェクト参照の暗黙的な修飾に使用されます。

パッケージがスタンドアロン・アプリケーションとして使用される場合、動的 SQL ステートメントはパッケージが **DYNAMICRULES BIND** を指定してバインドされたかのように処理されます。

動的 SQL ステートメントは、バインド動作を公開しているパッケージ内のパッケージ所有者の許可 ID を使用します。したがって、パッケージのユーザーが受け取るべきでない権限を、パッケージのバインド・プログラムに付与してはなりません。同様に、定義動作を公開するルーチンを定義するとき、パッケージのユーザーが受け取るべきでない権限を、ルーチンの定義者に付与してはなりません。動的ステートメントがルーチンの定義者の許可 ID を使用するためです。

次の動的な準備済み SQL ステートメントは、**DYNAMICRULES RUN** にバインドされなかったパッケージ内では使用できません。

GRANT、REVOKE、ALTER、CREATE、DROP、COMMENT ON、RENAME、SET INTEGRITY、および SET EVENT MONITOR STATE です。

### ENCODING

プランまたはパッケージ内の静的ステートメント内にあるすべてのホスト変数のエンコード方式を指定します。サポートしているのは DB2 for OS/390 だけです。サポートされているオプション値のリストについては、DB2 for OS/390 の資料を参照してください。

### EXPLAIN

各 SQL ステートメント用に選択したアクセス・プランに関する Explain 表の情報を、パッケージに保管します。

**NO** Explain 情報はキャプチャーされません。

**YES** Explain 表には、静的ステートメントの場合は prep/bind 時に、追加バインド・ステートメントの場合は実行時に、選択されたアクセス・プランについての情報が取り込まれます。

パッケージがルーチンに使用されるもので、パッケージに追加バインド・ステートメントが含まれる場合、そのルーチンは **MODIFIES SQL DATA** として定義されなければなりません。これが行われられない場合、パッケージ内の追加バインド・ステートメントはランタイム・エラーを生じます (SQLSTATE 42985)。

**REOPT**

再最適化可能な追加バインド SQL ステートメントの Explain 情報が、実行時に各 Explain 表に入れます。さらに、CURRENT EXPLAIN MODE レジスターが NO に設定されていても、Explain 情報は実行時に再最適化可能な動的 SQL ステートメント用に集められます。

パッケージがルーチンに使用される場合、そのルーチンは MODIFIES SQL DATA として定義されなければなりません。そうしなければ、パッケージ内の追加バインドおよび動的ステートメントはランタイム・エラーを生じます (SQLSTATE 42985)。

**ONLY ONLY** オプションにより、実行する特権がなくてもステートメントの EXPLAIN ができます。Explain 表にはデータが入りますが、永続的パッケージは作成されません。バインド処理中に、名前とバージョンが同じ既存パッケージを検出した場合、その既存パッケージはドロップも置換 (**ACTION REPLACE** を指定した場合) もされません。Explain 表の移植中にエラーが発生した場合、Explain 情報は、エラーを戻したステートメント、およびそれに続くどのステートメントに対しても追加されません。

**ALL** 適格な静的 SQL ステートメントの Explain 情報が、prep/bind 時に各 Explain 表に入れます。適格な追加バインド SQL ステートメントの Explain 情報が、実行時に各 Explain 表に入れます。さらに、CURRENT EXPLAIN MODE レジスターが NO に設定されている場合でも、適格な動的 SQL ステートメントの Explain 情報が実行時に収集されます。

パッケージがルーチンに使用される場合、そのルーチンは MODIFIES SQL DATA として定義されなければなりません。そうしなければ、パッケージ内の追加バインドおよび動的ステートメントはランタイム・エラーを生じます (SQLSTATE 42985)。

**EXPLSNAP**

Explain 表に Explain スナップショットを保管します。

**NO** Explain スナップショットはキャプチャーされません。

**YES** Explain 表には、静的ステートメントの場合は prep/bind 時に、追加バインド・ステートメントの場合は実行時に、適格な各静的 SQL ステートメントの Explain スナップショットが入れます。

パッケージがルーチンに使用されるもので、パッケージに追加バインド・ステートメントが含まれる場合、そのルーチンは MODIFIES SQL DATA として定義されなければなりません。これが行われない場合、パッケージ内の追加バインド・ステートメントはランタイム・エラーを生じます (SQLSTATE 42985)。

**REOPT**

再最適化可能な追加バインド SQL ステートメントの Explain スナップショット情報が、実行時に各 Explain 表に入れます。さらに、CURRENT EXPLAIN SNAPSHOT レジスターが NO に設定されていても、Explain スナップショット情報は実行時に再最適化可能な動的 SQL ステートメント用に集められます。

パッケージがルーチンに使用される場合、そのルーチンは `MODIFIES SQL DATA` として定義されなければなりません。そうしなければ、パッケージ内の追加バインドおよび動的ステートメントはランタイム・エラーを生じます (SQLSTATE 42985)。

**ALL** 適格な各静的 SQL ステートメントの Explain スナップショットが、`prep/bind` 時に Explain 表内に入れられます。適格な追加バインド SQL ステートメントの Explain スナップショット情報が、実行時に各 Explain 表に入れられます。さらに、`CURRENT EXPLAIN SNAPSHOT` レジスターが `NO` に設定されていても、Explain スナップショット情報は実行時に適格な動的 SQL ステートメント用に集められます。

パッケージがルーチンに使用される場合、そのルーチンは `MODIFIES SQL DATA` として定義されなければなりません。そうしなければ、パッケージ内の追加バインドおよび動的ステートメントはランタイム・エラーを生じます (SQLSTATE 42985)。

#### EXTENDEDINDICATOR

関連のあるプランまたはパッケージの実行時に、拡張された標識変数値を認識できるようにします。

**NO** 拡張された標識変数値が認識されない。標識変数は、通常標識変数です。負の標識変数値は `NULL` であることを暗黙に示し、正またはゼロの値は `NULL` 以外であることを暗黙に示します。これがデフォルトの条件です。

**YES** 拡張された標識変数値が認識される。認識されない標識変数値を使用するか、サポートされない場所でデフォルトまたは未割り当ての標識変数ベースの値を使用すると、バインドされたステートメントの実行時に、`DB2` データベース・マネージャーはエラー・メッセージを生成します。

#### FEDERATED

パッケージ内の静的 SQL ステートメントがニックネームまたはフェデレーテッド・ビューを参照するかどうかを指定します。このオプションが指定されず、パッケージ内の静的 SQL ステートメントがニックネームまたはフェデレーテッド・ビューを参照する場合は、警告が返され、パッケージは作成されません。

**NO** ニックネームまたはフェデレーテッド・ビューは、パッケージ内の静的 SQL ステートメントで参照されません。ニックネームまたはフェデレーテッド・ビューがこのパッケージの準備またはバインド・フェーズ中に静的 SQL ステートメントで見つかった場合、エラーが返され、パッケージは作成されません。

**YES** ニックネームまたはフェデレーテッド・ビューは、パッケージ内の静的 SQL ステートメントで参照が可能です。ニックネームまたはフェデレーテッド・ビューがこのパッケージの準備またはバインド中に静的 SQL ステートメントで見つからなかった場合、エラーまたは警告は返されず、パッケージは作成されます。

**FEDERATED\_ASYNCRONY**

フェデレーテッド・サーバーが組み込み SQL を使用するプログラムのアクセス・プランでサポートする非同期表キュー (ATQ) の最大数を指定します。

**ANY** オプティマイザーはアクセス・プラン用の ATQ の数を判別します。オプティマイザーは ATQ をプラン内のすべての適格な SHIP またはリモート・プッシュダウン演算子に割り当てます。

**DB2\_MAX\_ASYNC\_REQUESTS\_PER\_QUERY** サーバー・オプションに指定される値は、非同期要求の数を制限します。

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

プラン内の ATQ の数。数値を 0 から 32767 の範囲で指定します。

**FUNCPATH**

静的 SQL で、ユーザー定義の特殊タイプおよび機能を解析する際に使用する関数パスを指定します。このオプションを指定しなかった場合、デフォルトの関数パスは "SYSIBM"、"SYSFUN"、または USER になります。ここで USER は USER 特殊レジスターの値です。

*schema-name*

SQL ID (通常または区切り)。これは、アプリケーション・サーバーに存在するスキーマを識別します。スキーマが存在する場合、プリコンパイル時やバインド時に妥当性検査は行われません。同一スキーマは、関数パス内に一度しか存在できません。スキーマ名 SYSPUBLIC は、関数パスには指定できません。指定できるスキーマ数は、処理結果の関数パスの長さによって限定され、2048 バイトを超えることはできません。スキーマ SYSIBM は、明示的に指定する必要がありません。関数パス内に含まれていなければ、最初のスキーマに暗黙的に想定されます。

**GENERIC "string"**

BIND PACKAGE または REBIND PACKAGE のオプションとして、ターゲット・サーバーではサポートされていても DB2 for Linux, UNIX, and Windows ではサポートされていない新しいオプションのバインドをサポートします。 **BIND** または **PRECOMPILE** で定義されている BIND オプションを渡すようにするには、このオプションを使用しないでください。構文は次のとおりです。

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

各オプションと値は、1 つ以上のブランク・スペースで区切らなければなりません。例えば、以下のようにして、OPT1、OPT2、および OPT3 オプションをそれぞれバインドできます。

```
generic "opt1 value1 opt2 value2 opt3 value3"
```

ストリングの最大長は 4096 バイトです。各オプション名の最大長はストリングで 255 バイトです。

リモートの **BIND** コマンドで BIND GENERIC 節を使用すると、クライアントをアップグレードする必要なく新しい DB2 サーバーのバインド・オプションを送信できます。クライアントは、このオプションの構文を検査せず

に **BIND** コマンドを DB2 サーバーに渡します。DB2 サーバーは、**BIND** コマンドの構文を検査して処理します。

## GRANT

**注:** **GRANT**、**GRANT\_GROUP**、**GRANT\_USER**、および **GRANT\_ROLE** のうち複数のオプションが指定された場合には、指定された最後のオプションのみが実行されます。

**authid** 指定したユーザー名、ロール名、またはグループ ID に EXECUTE 特権と BIND 特権を付与します。SQL GRANT ステートメントで付与対象の **authid** のタイプを示す **USER**、**GROUP**、**ROLE** のいずれも指定されなかった場合は、GRANT ステートメントとその規則を使用してそのタイプが判別されます。規則については、**GRANT (Role)** ステートメント (**GRANT (Role) statement**) を参照してください。

## PUBLIC

**PUBLIC** に EXECUTE 特権と BIND 特権を付与します。

## GRANT\_GROUP *group-name*

指定したグループ名に EXECUTE 特権と BIND 特権を付与します。

## GRANT\_USER *user-name*

指定したユーザー名に EXECUTE 特権と BIND 特権を付与します。

## GRANT\_ROLE *role-name*

指定したロール名に EXECUTE 特権と BIND 特権を付与します。

## INSERT

DB2 Enterprise Server Edition サーバーヘプリコンパイルまたはバインドされているプログラムが、パフォーマンス向上のために挿入データをバッファリングすることを要求できるようにします。

**BUF** アプリケーションからの挿入データをバッファリングすることを指定します。

**DEF** アプリケーションからの挿入データをバッファリングしないことを指定します。

## ISOLATION

このパッケージにバインドされるプログラムを、他の実行プログラムの影響からどの程度分離できるかを指定します。

**CS** カーソル固定を分離レベルとして指定します。

**NC** コミットなし。コミットメント制御が使用されないということ指定します。なお、DB2 for Linux, UNIX, and Windows はこの分離レベルをサポートしません。

**RR** 反復可能読み取りを分離レベルとして指定します。

**RS** 読み取り固定を分離レベルとして指定します。読み取り固定は、パッケージ内での SQL ステートメントの実行を、他のアプリケーションが読み取りおよび変更を行った行に対する処理から分離させます。

**UR** 非コミット読み取りを分離レベルとして指定します。



**IMMEDWRITE**

グループ・バッファ・プールに從属するページセットまたはデータベース・パーティションに対する更新について、即時書き込みを行うかどうかを示します。サポートしているのは DB2 for OS/390 だけです。サポートされているオプション値のリストについては、DB2 for OS/390 の資料を参照してください。

**KEEPDYNAMIC**

このパラメーターは、コミット・ポイントの後で動的 SQL ステートメントを保持するかどうかを指定します。

DB2 for z/OS でサポートされているオプション値のリストについては、DB2 for z/OS インフォメーション・センターの資料を参照してください。

DB2 for Linux, UNIX, and Windows バージョン 9.8 フィックスパック 2 以降では、パッケージに対する **KEEPDYNAMIC** バインド・オプションの値を変更することができます。その際、新たなバインド操作は必要ないので、次のバインド操作が行われるまで、不必要な再コンパイルを避けることができます。**KEEPDYNAMIC** バインド・オプションは、動的 SQL ステートメントが COMMIT または ROLLBACK の後にも保持されるかどうかを指定することにより、準備済みステートメントに関連付けられたステートメント・テキストとセクションが SQL コンテキストにとどまる期間を制御します。

**YES**

準備済みステートメントに関連付けられたステートメント・テキストとセクションが SQL コンテキストで無期限に保持されるように指示します。動的 SQL ステートメントは複数のトランザクション間で保持されます。**KEEPDYNAMIC YES** を使ってバインド済みのすべてのパッケージは、デフォルトで、既存のパッケージ・キャッシュの動作との互換性を保ちます。

## いいえ

各作業単位の終わりに、準備済みステートメントに関連付けられたステートメント・テキストとセクションが SQL コンテキストで除去されるように指示します。**KEEPDYNAMIC NO** を使ってバインド済みのパッケージ内に準備された非アクティブの動的 SQL ステートメントは、COMMIT または ROLLBACK 操作の間に SQL コンテキストから除去されます。ステートメントは、新しいトランザクションの中で再び準備される必要があります。クライアント、ドライバー、またはアプリケーションは、再使用を望むすべての動的 SQL ステートメントを、新しい作業単位で再び準備する必要があります。パッケージがリモート・アプリケーションによって実行される場合には、準備済みステートメントの実行可能バージョンは、トランザクション終了時に、アプリケーション SQL コンテキストとの関連がなくなります。

アクティブな動的 SQL ステートメントは、それらが非アクティブになる次の COMMIT または ROLLBACK 操作まで、保持される必要があります。動的 SQL ステートメントがトランザクション境界でアクティブであることのできる、次のようなシチュエーションがあります。

- WITH HOLD オプションを使って宣言されたカーソルが、コミット・ポイントで開いている。
- 動的 SQL ステートメントが COMMIT または ROLLBACK 操作を実行している。
- COMMIT または ROLLBACK 操作を実行するストアド・プロシージャまたはユーザー定義関数を、動的 SQL ステートメントが呼び出す。

#### MESSAGES *message-file*

警告メッセージ、エラー・メッセージ、および完了状況メッセージの宛先を指定します。メッセージ・ファイルは、バインドが正常であるかどうかによって作成されます。メッセージ・ファイル名を指定しなかった場合、メッセージは標準出力に書き込まれます。ファイルへの完全パスを指定しなかった場合は、現行ディレクトリーが使用されます。なお、既存ファイルの名前を指定すると、そのファイルの内容は上書きされます。

#### OPTHINT

照会最適化ヒントを静的 SQL に使用するかどうかを制御します。サポートしているのは DB2 for OS/390 だけです。サポートされているオプション値のリストについては、DB2 for OS/390 の資料を参照してください。

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

パッケージ内のすべての静的ステートメントに使用される既存の最適化プロファイルの名前を指定します。このオプションのデフォルト値は、空ストリングです。この値は CURRENT OPTIMIZATION PROFILE 特殊レジスターが NULL である DML ステートメントの動的準備のデフォルトとしても適用されます。指定された名前が修飾されていない場合、それは SQL 識別子であり、QUALIFIER バインド・オプションによって暗黙的に修飾されます。

**BIND** コマンドは最適化ファイルを処理せず、名前が構造的に有効であることだけを確認します。したがって、最適化プロファイルが存在しないか無効である場合、最適化プロファイルを使用して DML ステートメントが最適化されるまで、理由コード 13 の SQL0437W 警告は発生しません。

#### OS400NAMING

DB2 for System i<sup>®</sup> データにアクセスする際に使用する命名オプションを指定します。サポートしているのは DB2 for System i だけです。サポートされているオプション値のリストについては、DB2 for System i の資料を参照してください。

区切り記号としてスラッシュが使用されているため、DB2 ユーティリティーが **OS400NAMING** SYSTEM オプションを指定してプリコンパイルまたはバインドされていても、このユーティリティーは、System i システムの命名規則を使用する特定の SQL ステートメントに関して実行時に構文エラーを報告することがあります。例えば、**OS400NAMING** SYSTEM オプションを指定してプリコンパイルまたはバインドされているかどうかには関係なく、System i システムの命名規則が使用されている場合、コマンド行プロセッサは SQL CALL ステートメントに関して構文エラーを報告します。

#### OWNER *authorization-id*

パッケージ所有者の 128 バイトの許可 ID を指定します。その所有者には、パッケージに含まれる SQL ステートメントを実行するための特権が必

要です。DBADM 権限を持つユーザーのみが、ユーザー ID 以外の許可 ID を指定できます。デフォルト値は、プリコンパイル/バインド処理の呼び出し側の許可 ID です。SYSIBM、SYSCAT、および SYSSTAT はこのオプションには無効な値です。authorization-id はユーザーでなければなりません。OWNER オプションを使用してロールまたはグループを指定することはできません。

**PATH** 静的 SQL で、ユーザー定義の特殊タイプおよび機能を解析する際に使用する関数パスを指定します。このオプションを指定しなかった場合、デフォルトの関数パスは "SYSIBM"、"SYSFUN"、または USER になります。ここで USER は USER 特殊レジスターの値です。

*schema-name*

SQL ID (通常または区切り)。これは、アプリケーション・サーバーに存在するスキーマを識別します。スキーマが存在する場合、プリコンパイル時やバインド時に妥当性検査は行われません。

**QUALIFIER** *qualifier-name*

パッケージに含まれる非修飾オブジェクトの 128 バイトの暗黙修飾子を指定します。OWNER が明示的に指定されているかどうかにかかわらず、その所有者の許可 ID がデフォルト ID になります。

**QUERYOPT** *optimization-level*

パッケージに含まれるすべての静的 SQL ステートメントに必要な最適化レベルを指示します。デフォルト値は 5 です。SET CURRENT QUERY OPTIMIZATION ステートメントが、使用できる最適化レベルの範囲の詳細を記述します。

**RELEASE**

リソースを、各 COMMIT ポイントで解放するか、アプリケーションの終了時に解放するかどうかを指示します。なお、DB2 for Linux, UNIX, and Windows はこの DRDA プリコンパイル/BIND オプションをサポートしません。DB2 z/OS バージョン 10 以降のサーバーでは、デフォルト値は DEALLOCATE です。

**COMMIT**

各コミット点でリソースを解放します。これは、動的 SQL ステートメントに使用されます。

**DEALLOCATE**

アプリケーションの終了時にだけリソースを解放します。

**SORTSEQ**

System i で使用するソート・シーケンス表を指定します。サポートしているのは DB2 for System i だけです。サポートされているオプション値のリストについては、DB2 for System i の資料を参照してください。

**SQLERROR**

エラーを検出した場合に、パッケージまたはバインド・ファイルを作成するかどうかを指示します。

**CHECK**

ターゲット・システムが、バインドされている SQL ステートメントのすべての構文、およびセマンティックの検査を行うことを指定



します。この処理の一部としてパッケージが作成されることはありません。バインド中に、名前とバージョンが同じ既存パッケージを検出した場合、その既存パッケージはドロップも置換 (**ACTION REPLACE** を指定した場合) もされません。

#### CONTINUE

SQL ステートメントのバインド時にエラーが発生しても、パッケージを作成します。許可または存在などの理由でバインドに失敗したこれらのステートメントは、**VALIDATE RUN** も指定されている場合は、実行時に増分でバインドすることができます。実行時にこれらのステートメントを実行しようとする、エラー (SQLCODE -525, SQLSTATE 51015) が生成されます。

#### NOPACKAGE

エラーを検出した場合、パッケージもバインド・ファイルも作成しません。

#### REOPT

DB2 がホスト変数、パラメーター・マーカ、グローバル変数、および特殊レジスターの値を使用して実行時にアクセス・パスを判別するようにするかどうかを指定します。有効な値は以下のとおりです。

**NONE** ホスト変数、パラメーター・マーカ、グローバル変数、または特殊レジスターを含む SQL ステートメントのアクセス・パスは、実際の値によって最適化されません。これらの変数のデフォルトの推定値が使用され、このプランがキャッシュされて使用されます。これはデフォルト値です。

**ONCE** 最初に照会が実行されるときに、ホスト変数、パラメーター・マーカ、グローバル変数、または特殊レジスターの実際の値によって、SQL ステートメントのアクセス・パスが最適化されます。このプランがキャッシュされて使用されます。

#### ALWAYS

照会が実行されるたびに、ホスト変数、パラメーター・マーカ、グローバル変数、または特殊レジスターの既知の値によって、SQL ステートメントのアクセス・パスが必ずコンパイルおよび再最適化されます。

#### REOPT | NOREOPT VARS

これらのオプションは、**REOPT ALWAYS** および **REOPT NONE** によって置き換えられましたが、前のものとの互換性のために引き続きサポートされています。DB2 がホスト変数、グローバル変数、パラメーター・マーカ、および特殊レジスターの値を使用して実行時にアクセス・パスを判別するようにするかどうかを指定します。サポートしているのは DB2 for OS/390 だけです。サポートされているオプション値のリストについては、DB2 for OS/390 の資料を参照してください。

#### SQLWARN

動的 SQL ステートメントのコンパイル時 (**PREPARE** または **EXECUTE IMMEDIATE** を通して)、または記述処理時 (**PREPARE...INTO** または **DESCRIBE** を通して) に警告を戻すかどうかを指示します。

**NO** SQL コンパイラから警告を戻しません。

**YES** SQL コンパイラーから警告を戻します。

SQLCODE +236、+237、および +238 は例外です。それらは、**SQLWARN** オプションの値が何であろうと戻されます。

**STATICREADONLY**

静的カーソルを **READ ONLY** または **INSENSITIVE** として扱うかどうかを指定します。

**NO** すべての静的カーソルは、ステートメント・テキストと **LANGLEVEL** プリコンパイル・オプションの設定に基づいて通常生成される場合と同じ属性を取ります。これはデフォルト値です。

**YES** **FOR UPDATE** または **FOR READ ONLY** 節を含まない静的カーソルは、**READ ONLY** と見なされます。

**INSENSITIVE**

**FOR UPDATE** 節を含まない静的カーソルは、**READ ONLY** または **INSENSITIVE** と見なされます。

**STRDEL**

SQL ステートメントで使用するストリング区切り文字として、アポストロフィ (') または二重引用符 (") のどちらを使用するか指定します。このプリコンパイル/バインド・オプションは、DB2 for Linux, UNIX, and Windows 用のサーバーではサポートされません。このオプションを指定しないと、DRDA サーバーは、組み込みのデフォルト値を使用します。

**APOSTROPHE**

ストリング区切り文字として、アポストロフィ (') を使用します。

**QUOTE**

ストリング区切り文字として、二重引用符 (") を使用します。

**SYSTIMESENSITIVE**

静的および動的 SQL ステートメントでのシステム期間テンポラル表への参照が、**CURRENT TEMPORAL SYSTEM\_TIME** 特殊レジスターの値の影響を受けるかどうかを示します。

**YES** システム期間テンポラル表への参照は、**CURRENT TEMPORAL SYSTEM\_TIME** 特殊レジスターの値の影響を受けます。この値はデフォルト値です。

**NO** システム期間テンポラル表への参照は、**CURRENT TEMPORAL SYSTEM\_TIME** 特殊レジスターの値の影響を受けません。

**TEXT** *label*

パッケージの記述。最大長は 255 文字です。また、デフォルト値はブランクです。このプリコンパイル/バインド・オプションは、DB2 for Linux, UNIX, and Windows 用のサーバーではサポートされません。

**TRANSFORM GROUP**

静的 SQL ステートメントが、ユーザー定義の構造化タイプの値をホスト・プログラムと交換するために使用する、変換グループ名を指定します。この変換グループは、動的 SQL ステートメントには使用されません。また、パラメーターの交換や外部関数またはメソッドの結果にも使用されません。

*groupname*

SQL ID。長さは最大で 18 バイトです。グループ名には、修飾子接頭部を含めることはできません。また、接頭部 **SYS** はデータベースで使用するために予約されているので、その接頭部は使用できません。ホスト変数とやりとりする静的 SQL ステートメントでは、構造化タイプの値の交換に使用する変換グループの名前は以下のようになります。

- **TRANSFORM GROUP BIND** オプション内のグループ名 (もしあれば)
- **TRANSFORM GROUP** 準備オプション内のグループ名。最初のプリコンパイル時に指定したとおりのもの (もしあれば)
- **DB2\_PROGRAM** グループ。グループ名が **DB2\_PROGRAM** の、特定のタイプに対する変換がある場合。
- 上にリストされた条件のいずれもない場合には、変換グループは使用されません。

静的 SQL ステートメントのバインド時には、以下のエラーが発生する可能性があります。

- **SQLCODE yyyyy, SQLSTATE xxxxx**: 変換が必要ですが、静的変換グループが選択されていません。
- **SQLCODE yyyyy, SQLSTATE xxxxx**: 選択された変換グループには、交換するデータ・タイプに必要な変換が含まれていません (入力変数用の **TO SQL**、出力変数用の **FROM SQL**)。
- **SQLCODE yyyyy, SQLSTATE xxxxx**: **FROM SQL** 変換の結果タイプは、出力変数のタイプと互換性がありません。または、**TO SQL** 変換のパラメーター・タイプは、入力変数のタイプと互換性がありません。

これらのエラー・メッセージで、*yyyyy* は SQL エラー・コードによって置き換えられ、*xxxxx* は SQL 状態コードによって置き換えられます。

**VALIDATE**

データベース・マネージャーが、許可エラーとエラー未検出のオブジェクトをいつ検査するかを判別します。この妥当性検査には、パッケージ所有者の許可 ID が使用されます。

**BIND** プリコンパイル/バインド時に妥当性検査が実行されます。オブジェクトが 1 つもない場合、または権限がまったく保持されていない場合、エラー・メッセージが作成されます。 **SQLERROR CONTINUE** を指定した場合、エラー・メッセージにかかわらずパッケージ/バインド・ファイルは作成されますが、エラーのあるステートメントは実行できません。

**RUN** バインド時に妥当性検査が行われます。すべてのオブジェクトが存在し、全権限が保持されていれば、それ以上実行しても検査は行われません。

プリコンパイル/バインド時にオブジェクトが 1 つもない場合、または権限がまったく保持されていない場合、 **SQLERROR CONTINUE** オプションの設定とは無関係に警告メッセージが作成されて、パッケ

ージは正常にバインドされます。ただし、プリコンパイル/バインド処理時に SQL ステートメントの権限検査と存在検査に障害が生じた場合、実行時に再実行される可能性があります。

## 例

以下は、myapp.bnd (myapp.sqc プログラムのプリコンパイル時に生成されるバインド・ファイル) と接続が確立しているデータベースをバインドする例です。

```
db2 bind myapp.bnd
```

バインド処理で生じたすべてのメッセージは、標準出力に送信されます。

## 使用上の注意

**REOPT** オプションに **ONCE** または **ALWAYS** 値を指定してパッケージをバインディングすると、静的および動的ステートメントのコンパイルおよびパフォーマンスが変化することがあります。

バインド処理は、アプリケーション・プログラムのソース・ファイルのプリコンパイル処理の一部として実行することもできますし、別のプロセスとして後から実行することもできます。バインドを別のプロセスとして実行するときは **BIND** を使用します。

パッケージを作成するのに使われた名前はバインド・ファイルに保管されます。その名前は、その名前が生成されたソース・ファイルの名前を基にして付けられます(既存のパスや拡張子は取り除かれます)。例えば、myapp.sql というプリコンパイル・ソース・ファイルは、myapp.bnd というデフォルト・バインド・ファイルとデフォルト・パッケージ名の **MYAPP** を生成します。ただし、**BINDFILE** および **PACKAGE** オプションを使用すれば、プリコンパイル時にバインド・ファイル名とパッケージ名を指定変更することができます。

既に存在しないスキーマ名にパッケージをバインドすると、そのスキーマを暗黙に作成することになります。スキーマの所有者は **SYSIBM** になります。スキーマに対する **CREATEIN** 特権が **PUBLIC** に付与されます。

**BIND** は、開始されたトランザクションのもとで実行されます。バインドの実行後、**BIND** は **COMMIT** または **ROLLBACK** を出して、現行トランザクションを終了させ、別のトランザクションを開始します。

致命的エラーまたは 100 を超えるエラーが生じた場合、バインドは停止します。致命的エラーが生じた場合、ユーティリティは処理を停止させ、全ファイルのクローズを試み、そのパッケージを廃棄します。

パッケージがバインド動作を公開するとき、以下のとおりとなります。

1. **BIND** オプション **OWNER** の暗黙的または明示的な値は、動的 SQL ステートメントの許可検査に使用されます。
2. **BIND** オプション **QUALIFIER** の暗黙的または明示的な値は、動的 SQL ステートメント内の非修飾オブジェクトを修飾するための暗黙的修飾子として使用されません。
3. 特殊レジスター **CURRENT SCHEMA** の値は、修飾には影響しません。

単一の接続で複数のパッケージが参照される場合、これらのパッケージによって準備されたすべての動的 SQL ステートメントはその特定のパッケージおよびそれらが使用される環境について **DYNAMICRULES** オプションで指定された動作をします。

**SQL0020W** メッセージに表示されるパラメーターは正しくエラーと示され、メッセージが示すとおりは無視されます。

SQL ステートメントがエラーであることが検出され、**BIND** オプション **SQLERROR CONTINUE** が指定されている場合、このステートメントは無効とマークされます。この SQL ステートメントの状態を変えるためには、さらに別の **BIND** を発行する必要があります。暗黙的および明示的な再バインドでは、無効なステートメントの状態は変わりません。**VALIDATE RUN** でバインドされたパッケージの場合、ステートメントは、再バインド時にオブジェクトが存在するかどうか、または権限の問題があるかどうかに応じて、暗黙的再バインドと明示的再バインドの両方で、静的バインドから追加バインドに変更されたり、追加バインドから静的バインドに変更されたりします。

パッケージのバインドに使用される許可 ID (**OWNER** バインド・オプションの値) に付与されているロールの特権、または **PUBLIC** への特権が、パッケージのバインド時に考慮されます。パッケージのバインドに使用される許可 ID がメンバーであるグループから取得したロールは使用されません。

組み込み SQL プログラムでは、**BIND** オプションが明示的に指定されていない場合、パッケージ内の静的ステートメントは **federated\_async** 構成パラメーターを使用してバインドされます。**FEDERATED\_ASYNCHRONY BIND** オプションが明示的に指定されている場合、値はパッケージのバインドに使用され、特殊レジスターの初期値ともなります。指定されていない場合、データベース・マネージャー構成パラメーターの値が特殊レジスターの初期値として使用されます。**FEDERATED\_ASYNCHRONY BIND** オプションは、明示的に設定されている場合に限り、動的 SQL に影響を与えます。

**FEDERATED\_ASYNCHRONY BIND** オプションの値は、**SYSCAT.PACKAGES** カタログ表の **FEDERATED\_ASYNCHRONY** 列に記録されます。この **BIND** オプションが明示的に指定されていない場合は、**federated\_async** 構成パラメーターの値が使用され、カタログの **FEDERATED\_ASYNCHRONY** 列には値 **-2** が示されます。

パッケージのバインド時に **BIND** オプション **FEDERATED\_ASYNCHRONY** が明示的に指定されていないと、そのパッケージが暗黙的または明示的に再バインドされる場合、パッケージの再バインドには **federated\_async** 構成パラメーターの現行値が使用されます。

## CATALOG DATABASE

システム・データベース・ディレクトリーに、データベースのロケーション情報を保管します。データベースは、ローカル・ワークステーションかリモート・データベース・パーティション・サーバーのどちらかに位置付けることができます。

### 有効範囲

パーティション・データベース環境では、ローカル・データベースをシステム・データベース・ディレクトリーにカタログするときに、データベースが常駐するサーバー上のデータベース・パーティションでこのコマンドを発行する必要があります。

### 許可

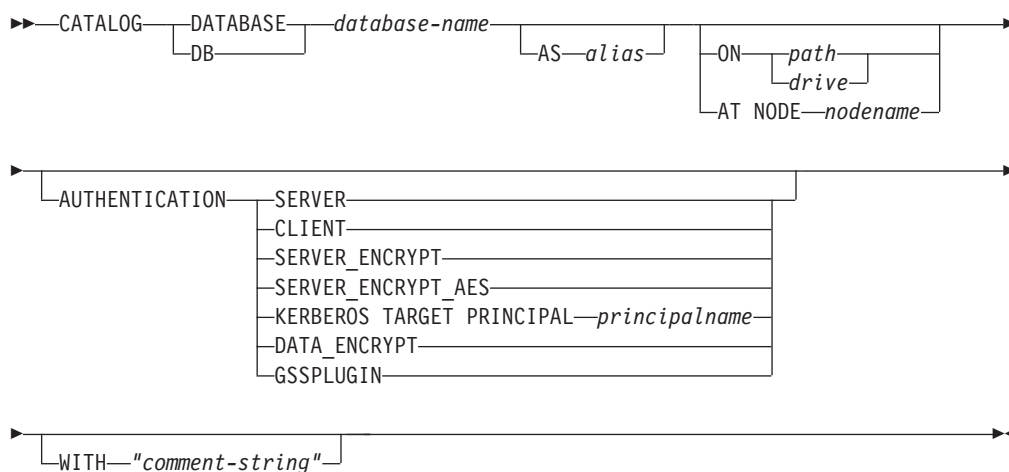
以下の権限のいずれか。

- SYSADM
- SYSCTRL

### 必要な接続

なし。ディレクトリー操作は、ローカル・ディレクトリーだけに影響します。

### コマンド構文



### コマンド・パラメーター

#### DATABASE *database-name*

カタログするデータベースの名前を指定します。

#### AS *alias*

カタログするデータベースの代替名として、別名を指定します。これを指定しなかった場合、データベース・マネージャーは *database-name* を別名として使用します。

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

カタログするデータベースが存在するパスを指定します。Windows オペレーティング・システムでは、上記の指定法に代えて、カタログするデータバ



ースが存在するドライブ名を指定してもかまいません (特定のパス上ではなく、ドライブ上に作成されたものの場合)。

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

カタログするデータベースが常駐するデータベース・パーティション・サーバーの名前を指定します。この名前は、ノード・ディレクトリー内の項目名と一致させてください。指定したノード名がノード・ディレクトリーに存在しない場合、警告が戻されますが、データベースはシステム・データベース・ディレクトリーにカタログされます。カタログしたデータベースへの接続が必要なときは、ノード名をノード・ディレクトリーにカタログしなければなりません。

#### **AUTHENTICATION**

リモート・データベースの認証値は保管されますが (**LIST DATABASE DIRECTORY** コマンドからの出力に現れる)、ローカル・データベースの認証値は保管されません。

認証タイプを指定すると、パフォーマンスが向上する場合があります。

**SERVER** 認証が、ターゲット・データベースを含むデータベース・パーティション・サーバーで行われるということを指定します。

**CLIENT** 認証が、アプリケーションを呼び出すデータベース・パーティション・サーバーで行われるということを指定します。

#### **SERVER\_ENCRYPT**

認証が、ターゲット・データベースを含むデータベース・パーティション・サーバーで行われるということ、およびユーザー ID とパスワードが送信元で暗号化されることを指定します。送信元でカタログされる認証タイプによって指定されるとおり、ユーザー ID とパスワードはターゲットで暗号化解除されます。

#### **KERBEROS**

認証が、kerberos セキュリティー・メカニズムを使用するということを指定します。

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

ターゲット・サーバー用の完全修飾 Kerberos プリンシパル名。つまり、name/instance@REALM という形式の、DB2 インスタンス所有者の完全修飾 Kerberos プリンシパル名。Windows 2000、Windows XP、および Windows Server 2003 では、DB2 サーバー・サービスのログオン・アカウントで *userid@DOMAIN*、*userid@xxx.xxx.xxx.com*、または *domain#userid* という形式です。

#### **DATA\_ENCRYPT**

認証が、ターゲット・データベースを含むデータベース・パーティション・サーバーで行われ、接続がデータ暗号化を使用しなければならないことを指定します。

#### **GSSPLUGIN**

外部 GSS API ベースのプラグイン・セキュリティ機構を使って認証が行われることを指定します。

### SERVER\_ENCRYPT\_AES

認証が、ターゲット・データベースを含むデータベース・パーティション・サーバーで行われるということ、および Advanced Encryption Standard (AES) 暗号化アルゴリズムによってユーザー ID とパスワードが送信元で暗号化されてターゲットで暗号化解除されることを指定します。

### WITH "comment-string"

システム・データベース・ディレクトリー内のデータベースまたはデータベース項目について記述します。コメント・ストリングの最大長は 30 文字です。復帰文字や改行文字は許可されません。コメント・テキストは必ず二重引用符で囲んでください。

### 例

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

### 使用上の注意

ローカル・ノードやリモート・データベース・パーティション・サーバーにあるデータベースをカタログする場合、以前にアンカタログしたデータベースを再カタログする場合、または 1 つのデータベースに対して複数の別名を保持する場合 (データベースのロケーションにかかわらず)、 **CATALOG DATABASE** を使用してください。

データベースを作成したとき、DB2 は自動的にそれらをカタログします。ローカル・データベース・ディレクトリーにデータベースの項目、システム・データベース・ディレクトリーに別の項目をカタログします。リモート・クライアント (または、同じマシンの別のインスタンスから実行しているクライアント) からデータベースを作成した場合、クライアント・インスタンスでは、システム・データベース・ディレクトリーにも項目が作成されます。

パス名もデータベース・パーティション・サーバー名も指定しなかった場合、データベースはローカルに、データベースのロケーションはデータベース・マネージャー構成パラメーターの **dftdbpath** に指定したものに想定されます。

データベース・マネージャー・インスタンスと同じデータベース・パーティション・サーバーのデータベースは、間接 項目としてカタログされます。その他のデータベース・パーティション・サーバーのデータベースは、リモート 入力としてカタログされます。

システム・データベース・ディレクトリーが存在しない場合、**CATALOG DATABASE** は自動的にそれを作成します。システム・データベース・ディレクトリーは、使用しているデータベース・マネージャー・インスタンスを含むパスに保管され、データベース外部で保守されます。

システム・データベース・ディレクトリーの内容をリストする場合は、 **LIST DATABASE DIRECTORY** コマンドを使用してください。ローカル・データベース・ディレクトリーの内容をリストする場合は、 **LIST DATABASE DIRECTORY ON path** を使用します。 *path* はデータベースが作成された場所です。



ディレクトリーのキャッシュが有効な場合、データベース、ノード、および DCS ディレクトリー・ファイルはメモリーにキャッシュされます。ディレクトリー・キャッシングが有効であるかどうかを確認する場合は、**GET DATABASE MANAGER CONFIGURATION** コマンドの出力内の *dir\_cache* ディレクトリー・キャッシュ・サポート構成パラメーターの値を検査してください。アプリケーションのディレクトリー・キャッシュは、最初のディレクトリー検索時に作成されます。キャッシュはアプリケーションがディレクトリー・ファイルのいずれかを修正したときにのみ最新にされるため、他のアプリケーションが行ったディレクトリーの変更は、アプリケーションを再始動するまで有効にならないことがあります。

CLP のディレクトリー・キャッシュを最新表示するには、**TERMINATE** コマンドを使用します。データベース・マネージャーの共有キャッシュを最新表示するには、データベース・マネージャーを停止させてから (**db2stop**)、再始動させます (**db2start**)。別のアプリケーション用のディレクトリー・キャッシュを最新にするには、そのアプリケーションを停止させてから再始動させてください。

## CATALOG DCS DATABASE

リモート・ホストまたは System i データベースについての情報を、データベース接続サービス (DCS)ディレクトリーに保管します。

このようなデータベースには、DB2 Connect™ などのアプリケーション・リクエスター (AR) を介してアクセスします。システム・データベース・ディレクトリー内のデータベース名と一致するデータベース名が DCS ディレクトリー項目にある場合、指定した AR を呼び出して、データベースが存在するリモート・サーバーに SQL 要求を転送します。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL

### 必要な接続

なし

### コマンド構文

```

▶▶ CATALOG DCS DATABASE database-name AS target-database-name
   DB
▶ AR library-name PARMS "parameter-string"
▶ WITH "comment-string"

```

### コマンド・パラメーター

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

カタログするターゲット・データベースの別名を指定します。この名前は、リモート・データベース・パーティション・サーバーに関連したデータベース・ディレクトリー内の項目名と一致させてください。

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

カタログするターゲット・ホストまたは System i データベースの名前を指定します。

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

DCS ディレクトリーにリストされたリモート・データベースにアクセスするとき、ロードおよび使用されるアプリケーション・リクエスター・ライブラリーの名前を指定します。

DB2 Connect AR を使用する場合は、ライブラリー名を指定しないでください。デフォルト値で DB2 Connect を呼び出します。

DB2 Connect を使用しない場合は、AR のライブラリー名を指定します。そして、そのライブラリーをデータベース・マネージャー・ライブラリーと同じパスに置いてください。Windows オペレーティング・システムでは、そ

のパスは `drive:%sqllib%bin` です。Linux および UNIXオペレーティング・システムでは、パスはインスタンス所有者の `$HOME/sqllib/lib` になります。

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

呼び出したときに AR にパスされるパラメーター・ストリングを指定します。パラメーター・ストリングは、二重引用符で囲んでください。

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

DCS ディレクトリー項目について記述します。このディレクトリーにカタログするデータベースについて、任意の注釈を入力できます。最大長は 30 文字です。復帰文字や改行文字は許可されません。コメント・テキストは必ず二重引用符で囲んでください。

## 例

次は、DB1 データベース (DB2 for z/OS データベース) に関する情報を DCS ディレクトリーにカタログする例です。

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

## 使用上の注意

DB2 Connect プログラムは、次のような DRDA アプリケーション・サーバーへの接続を提供します。

- System/370 および System/390<sup>®</sup> アーキテクチャーのホスト・コンピューター上の DB2 for OS/390 and z/OS データベース。
- System/370 および System/390 アーキテクチャーのホスト・コンピューター上の DB2 for VM and VSE データベース。
- Application System/400<sup>®</sup> (System i) および System i コンピューター上の System i データベース。

データベース接続サービスが存在しない場合、データベース・マネージャーは自動的にそれを作成します。このディレクトリーは、使用しているデータベース・マネージャー・インスタンスを含むパスに保管されます。また、データベースの外側で保持されます。

データベースは、システム・データベース・ディレクトリーにリモート・データベースとしてもカタログしなければなりません。

DCS ディレクトリーの内容をリストする場合は、**LIST DCS DIRECTORY** コマンドを使用してください。

ディレクトリーのキャッシュが有効な場合、データベース、ノード、および DCS ディレクトリー・ファイルはメモリーにキャッシュされます。アプリケーションのディレクトリー・キャッシュは、最初のディレクトリー検索時に作成されます。キャッシュはアプリケーションがディレクトリー・ファイルのいずれかを修正したときにのみ最新にされるため、他のアプリケーションが行ったディレクトリーの変更は、アプリケーションを再始動するまで有効にならないことがあります。

## CATALOG DCS DATABASE

CLP のディレクトリー・キャッシュを最新表示するには、**TERMINATE** コマンドを使用します。DB2 の共有キャッシュを最新表示するには、データベース・マネージャー を停止 (**db2stop**) してから、再始動 (**db2start**) してください。別のアプリケーション用のディレクトリー・キャッシュを最新にするには、そのアプリケーションを停止させてから再始動させてください。

## CATALOG LDAP DATABASE

Lightweight Directory Access Protocol (LDAP) でデータベースを登録します。

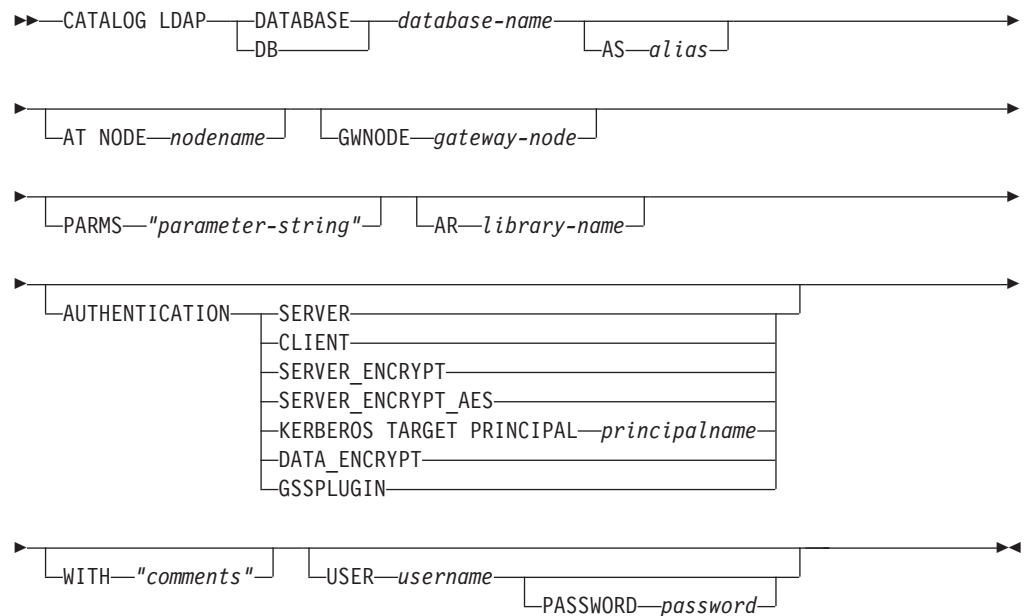
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

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

カタログするデータベースの名前を指定します。

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

カタログするデータベースの代替名として、別名を指定します。別名が指定されない場合は、データベース名が別名として使用されます。

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

データベースが常駐するデータベース・サーバーに LDAP ノード名を指定します。このパラメーターは、リモート・サーバーでデータベースを登録する場合に指定する必要があります。

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

ゲートウェイ・サーバーに LDAP ノード名を指定します。

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

DCS データベースへのアクセス時にアプリケーション・リクエスター (AR) に渡される、パラメーター・ストリングを指定します。変更パスワード *sym\_dest\_name* をパラメーター・ストリングで指定しないでください。

LDAP で DB2 サーバーを登録する場合、キーワード **CHGPWDLU** を使用して変更パスワード LU 名を指定してください。

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

DCS ディレクトリーにリストされたリモート・データベースにアクセスするとき、ロードおよび使用されるアプリケーション・リクエスター・ライブラリーの名前を指定します。

DB2 Connect AR を使用する場合は、ライブラリー名を指定しないでください。デフォルト値で DB2 Connect を呼び出します。

DB2 Connect を使用しない場合は、AR のライブラリー名を指定します。そして、そのライブラリーをデータベース・マネージャー・ライブラリーと同じパスに置いてください。Windows オペレーティング・システムでは、そのパスは `drive:¥sqllib¥dll` です。UNIX オペレーティング・システムでは、パスはインスタンス所有者の `$HOME/sqllib/lib` になります。

### **AUTHENTICATION**

認証レベルを指定します。有効な値は以下のとおりです。

**SERVER** 認証が、ターゲット・データベースを含むノードで行われるということを指定します。

**CLIENT** 認証が、アプリケーションの呼び出し元であるノードで行われるということを指定します。

### **SERVER\_ENCRYPT**

認証が、ターゲット・データベースを含むデータベース・パーティション・サーバーで行われるということ、およびユーザー ID とパスワードが送信元で暗号化されることを指定します。送信元でカタログされる認証タイプによって指定されるとおり、ユーザー ID とパスワードはターゲットで暗号化解除されます。

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

認証が、ターゲット・データベースを含むデータベース・パーティション・サーバーで行われるということ、および Advanced Encryption Standard (AES) 暗号化アルゴリズムによってユーザー ID とパスワードが送信元で暗号化されてターゲットで暗号化解除されることを指定します。

### **KERBEROS**

認証が、kerberos セキュリティー・メカニズムを使用するということを指定します。

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

ターゲット・サーバー用の完全修飾 Kerberos プリンシパル名。つまり、DB2 サーバー・サービスのログオン・アカウントで、`userid@xxx.xxx.xxx.com` または `domain¥userid` の形式です。

### **DATA\_ENCRYPT**

認証が、ターゲット・データベースを含むノードで行われ、接続がデータ暗号化を使用しなければならないことを指定します。

**GSSPLUGIN**

外部 GSS API ベースのプラグイン・セキュリティー機構を使って認証が行われることを指定します。

**WITH "comments"**

DB2 サーバーについて記述します。ネットワーク・ディレクトリーに登録されているサーバーについての記述を補足する、任意のコメントを入力できます。最大長は 30 文字です。復帰文字や改行文字は許可されません。コメント・テキストは必ず二重引用符で囲んでください。

**USER username**

ユーザーの LDAP 識別名 (DN) を指定します。LDAP ユーザー DN には、LDAP ディレクトリーでオブジェクトを作成するための十分な権限が必要です。ユーザーの LDAP DN が指定されない場合、現行ログオン・ユーザーの資格情報が使用されます。ユーザーの LDAP DN およびパスワードが **db2ldcfg** を使用して指定されている場合、ユーザー名とパスワードをここで指定する必要はありません。

**PASSWORD password**

アカウント・パスワードを示します。ユーザーの LDAP DN およびパスワードが **db2ldcfg** を使用して指定されている場合、ユーザー名とパスワードをここで指定する必要はありません。

**使用上の注意**

ノード名が指定されないと、DB2 は現行のマシン上で DB2 サーバーを表す、LDAP の最初のノードを使用します。

次の場合、LDAP でデータベースを手動で登録 (カタログ) する必要があるかもしれません。

- データベース・サーバーが LDAP をサポートしない場合。管理者は、LDAP をサポートするクライアントが、各クライアント・マシン上でローカルにデータベースをカタログしなくてもデータベースにアクセスできるように、LDAP で各データベース手動で登録する必要があります。
- アプリケーションが、データベースに接続するために異なる名前を使用する必要がある場合。この場合、管理者は異なる別名を使用してデータベースにカタログできます。
- データベースは、ホストまたは System i データベース・サーバーに常駐します。この場合、管理者は LDAP でデータベースを登録し、**GWNODE** パラメーターを介してゲートウェイ・ノードを指定することができます。
- CREATE DATABASE IN LDAP の実行中、データベース名が既に LDAP に存在する場合。データベースは依然としてローカル・マシン上で作成されています (ローカル・アプリケーションからアクセス可能) が、LDAP にある既存の項目には、新規データベースの内容は反映されません。この場合、管理者は次のことを行えます。
  - LDAP の既存のデータベース項目を除去し、手動で新しいデータベースを LDAP に登録する。
  - 異なる別名を使って LDAP で新しいデータベースを登録する。

## CATALOG LDAP NODE

Lightweight Directory Access Protocol (LDAP) で新しいノード項目をカタログします。

### 許可

なし

### 必要な接続

なし

### コマンド構文

```

▶▶ CATALOG LDAP NODE nodename AS nodealias
└──┬── USER username ───┬── PASSWORD password
    └──────────────────────────┘
  
```

### コマンド・パラメーター

**NODE** *nodename*

DB2 サーバーの LDAP ノード名を指定します。

**AS** *nodealias*

LDAP ノード項目に新しい別名を指定します。

**USER** *username*

ユーザーの LDAP 識別名 (DN) を指定します。LDAP ユーザー DN には、LDAP ディレクトリーでオブジェクトを作成するための十分な権限が必要です。ユーザーの LDAP DN が指定されない場合、現行ログオン・ユーザーの資格情報が使用されます。

**PASSWORD** *password*

アカウント・パスワードを示します。

### 使用上の注意

**CATALOG LDAP NODE** コマンドは、DB2 サーバーを表すノードに、異なる別名を指定するのに使用されます。



## CATALOG LOCAL NODE

同一のマシンに常駐するインスタンスのローカル別名を作成します。同じワークステーション上にユーザーのクライアントからアクセスされる複数のインスタンスがある場合、ローカル・ノードをカタログする必要があります。ローカル・ノードにアクセスするために、プロセス間通信 (IPC) が使用されます。

### 許可

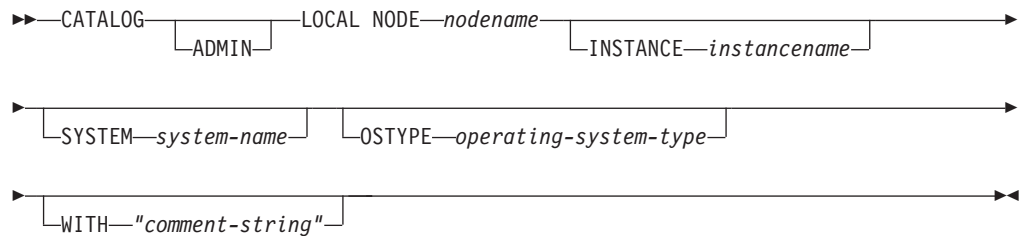
以下の権限のいずれか。

- SYSADM
- SYSCTRL

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### ADMIN

カタログするローカル Administration Server ノードを指定します。

#### INSTANCE *instancename*

アクセスするローカル・インスタンスの名前。

#### SYSTEM *system-name*

サーバー・マシンを識別するために使用する DB2 システム名を指定します。

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

サーバー・マシンのオペレーティング・システムのタイプを指定します。有効な値は、AIX、WIN、HPUX、SUN、OS390、OS400、VM、VSE、SNI、SCO、LINUX、および DYNIX です。

### 例

ワークステーション A には、inst1 および inst2 の 2 つのサーバー・インスタンスがあります。単一の CLP セッションから両方のインスタンスにデータベースを作成するには、以下の一連のコマンドを発行します (DB2INSTANCE 環境変数が inst1 に設定されているものと想定)。

1. inst1 にローカル・データベースを作成します。

```
db2 create database mydb1
```

## CATALOG LOCAL NODE

2. このワークステーションに別のサーバー・インスタンスをカタログします。

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

3. mynode2 にデータベースを作成します。

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

### 使用上の注意

ディレクトリーのキャッシュが有効な場合、データベース、ノード、および DCS ディレクトリー・ファイルはメモリーにキャッシュされます。アプリケーションのディレクトリー・キャッシュは、最初のディレクトリー検索時に作成されます。キャッシュはアプリケーションがディレクトリー・ファイルのいずれかを修正したときにのみ最新にされるため、他のアプリケーションが行ったディレクトリーの変更は、アプリケーションを再始動するまで有効にならないことがあります。

CLP のディレクトリー・キャッシュを最新表示するには、**TERMINATE** を使用します。DB2 の共有キャッシュを最新表示するには、データベース・マネージャーを停止させてから (**db2stop**)、再始動させます (**db2start**)。別のアプリケーション用のディレクトリー・キャッシュを最新にするには、そのアプリケーションを停止させてから再始動させてください。

## CATALOG NAMED PIPE NODE

ノード・ディレクトリーに Named PIPE ノード項目を追加します。リモート・ノードにアクセスするときに、この Named PIPE を使用します。

このコマンドは Windows のみで使用できます。

### 許可

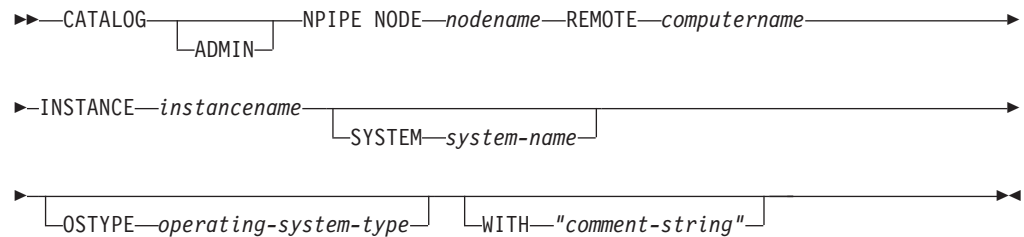
以下の権限のいずれか。

- SYSADM
- SYSCTRL

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### ADMIN

NPIPE Administration Server ノードをカタログすることを指定します。

#### REMOTE *computername*

ターゲット・データベースが存在するノードのコンピューター名。最大長は 15 文字です。

#### INSTANCE *instancename*

ターゲット・データベースが常駐するサーバー・インスタンスの名前です。リモート・ノードと通信する際に使用する、リモート Named PIPE の名前と同じにしてください。

#### SYSTEM *system-name*

サーバー・マシンを識別するために使用する DB2 システム名を指定します。

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

サーバー・マシンのオペレーティング・システムのタイプを指定します。有効な値は、AIX、WIN、HPUX、SUN、OS390、OS400、VM、VSE、SNI、SCO、および LINUX です。

### 例

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

### 使用上の注意

データベース・マネージャーは、最初のノードがカタログされたとき (つまり、**CATALOG...NODE** コマンドが最初に発行されたとき) にノード・ディレクトリーを作成します。Windows クライアントでは、そのクライアントをインストールしたインスタンス・サブディレクトリーに、ノード・ディレクトリーを保管して維持します。また、AIX クライアントでは、DB2 インストール・ディレクトリーにノード・ディレクトリーを作成します。

ローカル・ノード・ディレクトリーの内容をリストする場合は、**LIST NODE DIRECTORY** コマンドを使用してください。

ディレクトリーをキャッシュできる場合 (**GET DATABASE MANAGER CONFIGURATION** コマンドの構成パラメーター **dir\_cache** を参照)、データベース、ノード、および DCS のディレクトリー・ファイルはメモリーにキャッシュされます。アプリケーションのディレクトリー・キャッシュは、最初のディレクトリー検索時に作成されます。キャッシュはアプリケーションがディレクトリー・ファイルのいずれかを修正したときのみ最新にされるため、他のアプリケーションが行ったディレクトリーの変更は、アプリケーションを再始動するまで有効にならないことがあります。

CLP のディレクトリー・キャッシュを最新表示するには、**TERMINATE** コマンドを使用します。DB2 の共有キャッシュを最新表示するには、データベース・マネージャーを停止させてから (**db2stop**)、再始動させます (**db2start**)。別のアプリケーション用のディレクトリー・キャッシュを最新にするには、そのアプリケーションを停止させてから再始動させてください。

## CATALOG ODBC DATA SOURCE

ユーザーまたはシステム ODBC データ・ソースをカタログします。

ODBC (Open Database Connectivity) でのデータ・ソース という語は、特定のデータベースまたはファイル・システムのユーザー定義名のことです。この名前は、ODBC API を介してデータベースまたはファイル・システムにアクセスするときに使用されます。ユーザー・データ・ソースまたはシステム・データ・ソースのどちらであってもカタログできます。ユーザー・データ・ソースはそれをカタログしたユーザーにのみ可視になりますが、システム・データ・ソースは他のすべてのユーザーから可視であり使用できます。

このコマンドは Windows プラットフォームでのみ使用できます。

### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

**USER** ユーザー・データ・ソースをカタログします。キーワードを指定しない場合、これがデフォルトです。

#### SYSTEM

システム・データ・ソースをカタログします。

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

カタログするデータ・ソースの名前を指定します。データ・ソースの名前とデータベースの名前は同じでなければなりません。したがって、データ・ソースの名前はデータベース名の最大長に制限されます。

#### ALL DATA SOURCES

すべてのローカル・データベース別名を ODBC データ・ソース (DSN) としてカタログするよう指定します。

### 使用上の注意

Microsoft Windows Vista 以降のバージョンでは、完全な管理者特権で実行する DB2 コマンド・ウィンドウから **CATALOG SYSTEM ODBC DATA SOURCE** コマンドを実行する必要があります。

**ALL DATA SOURCES** パラメーターを指定しても、dbalias パラメーターがローカル・データベース・ディレクトリー内のデータベースの別名に一致する値に設定された、既存の ODBC DSN が更新されるわけではありません。

## CATALOG ODBC DATA SOURCE

Windows 環境では、DB2 のアップグレード後に ODBC DSN を復元する必要があります。ODBC DSN 設定をインポートおよびエクスポートするには、**db2cfimp** コマンドまたは **db2cfexp** コマンドを使用するか、あるいは DSN を手動で追加します。

### 例 1

「MyProdDatabase」という名前の既存の ODBC DSN があると想定します。dbalias パラメーターは「PRODDB」に設定されています。ローカル・ディレクトリーに「PRODDB」という別名のデータベースもがあると想定します。**CATALOG ODBC DATA SOURCE myproddb** コマンドまたは **CATALOG ODBC ALL DATA SOURCES** コマンドを実行しても、「MyProdDatabase」DSN は変更されません。DSN がデータベース別名と一致しないためです。その代わりに、「PRODDB」の ODBC DSN 項目が作成され、dbalias が「PRODDB」に設定されます。データベース別名と同じ名前の既存の ODBC DSN があると、既存の ODBC DSN の dbalias パラメーターは更新されて、データベースの別名になります。関連する CLI パラメーターと値はすべて未変更のままになります。

### 例 2

dbalias パラメーターが「salesdb」に設定されている、「MYDB」という名前の既存の DSN があると想定します。ローカル・ディレクトリー名に「MYDB」という名前のデータベースがある場合、**CATALOG ODBC DATA SOURCE mydb** コマンドまたは **CATALOG ODBC ALL DATA SOURCES** コマンドを実行すると、DSN の dbalias パラメーターは「MYDB」に変更されます。

## CATALOG TCPIP/TCPIP4/TCPIP6 NODE

ノード・ディレクトリーに、伝送制御プロトコル/インターネット・プロトコル (TCP/IP) データベース・パーティション・サーバー項目を追加します。リモート・データベース・パーティション・サーバーにアクセスするには、TCP/IP 通信プロトコルを使用します。 **CATALOG TCPIP/TCPIP4/TCPIP6 NODE** コマンドは、クライアントで実行されます。

注: DB2 for Linux, UNIX, and Windows バージョン 10.1 リリース以降、 AIX 5.3 オペレーティング・システムはサポートされません。 DB2 for Linux, UNIX, and Windows バージョン 9.7 が、 AIX 5.3 オペレーティング・システムをサポートしている最後のリリースです。 AIX 6.1 オペレーティング・システムが、サポートされる最小レベルです。

### 許可

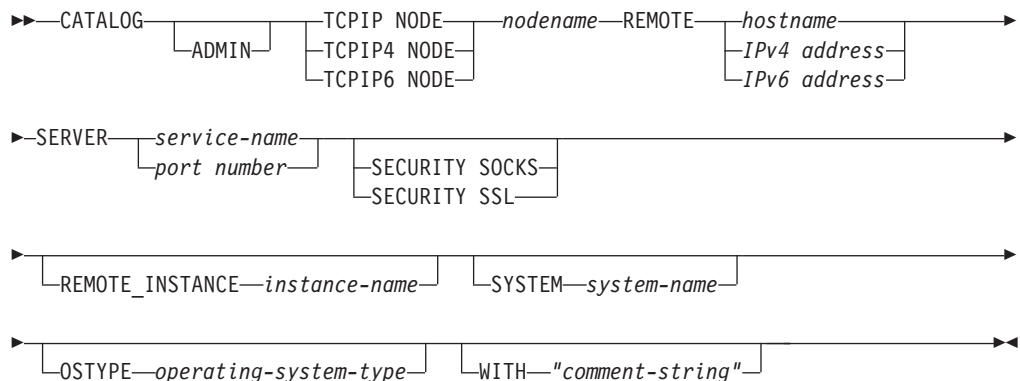
以下の権限のいずれか。

- SYSADM
- SYSCTRL

### 必要な接続

なし。ディレクトリー操作は、ローカル・ディレクトリーだけに影響します。

### コマンド構文



### コマンド・パラメーター

**ADMIN** TCP/IP Administration Server ノードをカタログすることを指定します。

**SECURITY SOCKS** パラメーターを指定する場合、このパラメーターは指定できません。

**TCPIP NODE** *nodename*

TCPIP、TCPIP4、または TCPIP6 データベース・パーティション・サーバーのノード名は、カタログするデータベースが入っているマシンに設定できるローカル・ニックネームを表します。 IPv4 IP アドレスを指定する場合は **TCPIP4** のみ指定し、IPv6 IP アドレスを指定する場合は **TCPIP6** のみ指定してください。

## CATALOG TCPIP/TCPIP4/TCPIP6 NODE

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

ターゲット・データベースが存在するノードのホスト名または IP アドレス。 *IP address* としては、IPv4 のアドレスも IPv6 のアドレスも使用できます。ホスト名は、TCP/IP ネットワークで認識されるデータベース・パーティション・サーバーの名前です。ホスト名の最大長は 255 文字です。

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

サーバー・データベース・マネージャー・インスタンスのサービス名またはポート番号を指定します。最大長は 14 文字です。このパラメーターには、大文字と小文字の区別があります。

サービス名を指定する場合、クライアントの *services* ファイルは、そのサービス名をポート番号にマップするのに使用されます。サービス名は、サーバーのデータベース・マネージャー構成ファイルで指定します。サーバーの *services* ファイルは、このサービス名をポート番号にマップするのに使用されます。クライアントとサーバーのポート番号は一致していなければなりません。

サーバーのデータベース・マネージャー構成ファイルでサービス名の代わりにポート番号を指定することもできますが、お勧めできません。ポート番号を指定した場合、ローカルの *services* ファイルに、サービス名を指定する必要はありません。

ADMIN ノードの場合、このパラメーターは指定できません。一方、ADMIN 以外のノードの場合、これは必須です。ADMIN ノードでの値は、常に 523 です。

### SECURITY SOCKS

ノードが SOCKS 対応であることを指定します。このパラメーターは、IPv4 でのみサポートされます。 **CATALOG TCPIP NODE** が使用されており、かつ **SECURITY SOCKS** が指定された場合、DB2 データベース製品では接続確立のために IPv4 が使用されます。 **ADMIN** パラメーターを指定する場合、このパラメーターは指定できません。

以下の環境変数は必須で、SOCKS を有効にするよう設定しなければなりません。

#### SOCKS\_NS

SOCKS サーバーのホスト・アドレスを解決するためのドメイン・ネーム・サーバーです。これはホスト名または IPv4 アドレスでなければなりません。

#### SOCKS\_SERVER

SOCKS サーバーの完全修飾ホスト名または IPv4 アドレス。  
SOCKS 化した IBM Data Server Client が完全修飾ホスト名を解決できない場合、IPv4 アドレスが入力されたと想定します。

以下の条件の少なくとも 1 つが真である必要があります。

- SOCKS サーバーは、ドメイン・ネーム・サーバー経由で到達可能。
- *hosts* ファイルの中に、ホスト名のリストがある。このファイルのロケーションは、TCP/IP ドキュメンテーションに説明されています。
- IPv4 アドレスが指定されている。



このコマンドを **db2start** の後に出す場合、このコマンドを有効にするためには **TERMINATE** コマンドを出す必要があります。

**SECURITY SSL**

ノードが SSL 対応であることを指定します。**SECURITY SSL** 節は、**ADMIN** パラメーターも指定する場合には指定できません。

**REMOTE\_INSTANCE** *instance-name*

データベースが存在し、アタッチまたは接続を確立するサーバー・インスタンスの名前を指定します。

**SYSTEM** *system-name*

サーバー・マシンを識別するために使用する DB2 システム名を指定します。これは、物理マシン、サーバー・システム、またはワークステーションの名前です。

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

サーバー・マシンのオペレーティング・システムのタイプを指定します。有効な値は、AIX、WIN、HPUX、SUN、OS390、OS400、VM、VSE、および LINUX です。

**WITH** *comment-string*

データベース・ディレクトリー内のデータベース項目について記述します。そのデータベースについての記述を補足する、任意の注釈を入力することができます。最大長は 30 文字です。復帰文字や改行文字は許可されません。注釈テキストは、単一引用符または二重引用符で囲む必要があります。

**例**

**CATALOG TCPIP NODE** コマンドを使用してホスト名を指定するには、次のようにコマンドを発行します。

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

**CATALOG TCPIP4 NODE** コマンドを使用して IPv4 アドレスを指定するには、次のようにコマンドを発行します。

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

この例では IPv4 アドレスを指定しています。**CATALOG TCPIP4 NODE** コマンドでは、IPv6 アドレスを指定しないようにしてください。そのようにしてもカタログで障害が発生するわけではありませんが、カタログにおいて無効なアドレスが指定されたため、それ以降のアタッチまたは接続は失敗します。

**CATALOG TCPIP6 NODE** コマンドを使用して IPv6 アドレスを指定するには、次のようにコマンドを発行します。

```
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"
```

この例では、**SERVER** について IPv6 アドレスとポート番号を指定しています。

**CATALOG TCPIP4 NODE** コマンドでは、IPv6 アドレスを指定しないようにしてくださ

い。そのようにしてもカタログで障害が発生するわけではありませんが、カタログにおいて無効なアドレスが指定されたため、それ以降のアタッチまたは接続は失敗します。

以下の例では、SSL 接続にノードをカタログします (サーバー・ホスト名は *hostname* で、*ssl\_port* はこのデータベース・サーバーが SSL プロトコルを使用するリモート・クライアント・ノードからの通信を待機するポート番号です)。

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

### 使用上の注意

データベース・マネージャーは、最初のノードがカタログされたとき (つまり、**CATALOG...NODE** コマンドが最初に発行されたとき) にノード・ディレクトリーを作成します。Windows クライアントでは、そのクライアントをインストールしたインスタンス・サブディレクトリーに、ノード・ディレクトリーを保管して維持します。また、AIX クライアントでは、DB2 インストール・ディレクトリーにノード・ディレクトリーを作成します。

ローカル・ノード・ディレクトリーの内容をリストする場合は、**LIST NODE DIRECTORY** コマンドを使用してください。

ディレクトリーのキャッシュが有効な場合、データベース、ノード、および DCS ディレクトリー・ファイルはメモリーにキャッシュされます。アプリケーションのディレクトリー・キャッシュは、最初のディレクトリー検索時に作成されます。キャッシュはアプリケーションがディレクトリー・ファイルのいずれかを修正したときにのみ最新にされるため、他のアプリケーションが行ったディレクトリーの変更は、アプリケーションを再始動するまで有効にならないことがあります。

CLP のディレクトリー・キャッシュを最新表示するには、**TERMINATE** コマンドを使用します。DB2 の共有キャッシュを最新表示するには、データベース・マネージャーを停止させてから (**db2stop**)、再始動させます (**db2start**)。別のアプリケーション用のディレクトリー・キャッシュを最新にするには、そのアプリケーションを停止させてから再始動させてください。

DB2 データベース・マネージャーに IPv6 を *listen* させるには、最初にオペレーティング・システムとサーバーを IPv6 用に構成する必要があります。システム管理者に相談して、この構成が IPv6 TCPIP ノードのカタログの前に行われるようにしてください。AIX 6.1 でこれを行う方法については、IPv4 構成の IPv6 へのアップグレードを参照してください。

## CHANGE DATABASE COMMENT

システム・データベース・ディレクトリーまたはローカル・データベース・ディレクトリー内の、データベースの注釈を変更します。現行のコメント関連テキストは、新規のコメント・テキストと置き換えることができます。

### 有効範囲

このコマンドは、それが実行されたデータベース・パーティションに対してだけ影響を与えます。

### 許可

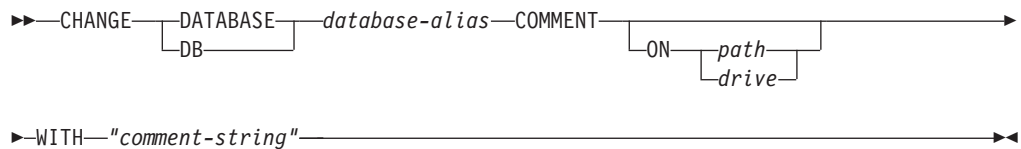
以下の権限のいずれか。

- SYSADM
- SYSCTRL

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### DATABASE *database-alias*

注釈を変更するデータベースの別名を指定します。システム・データベース・ディレクトリー内の注釈を変更するには、そのデータベースの別名を指定します。また、ローカル・データベース・ディレクトリー内の注釈を変更するには、そのデータベースが常駐するパスを指定し (*path* パラメーターで)、データベース名 (別名ではない) を入力します。

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

データベースが存在するパスを指定して、ローカル・データベース・ディレクトリー内の注釈を変更します。パスを指定しなかった場合、システム・データベース・ディレクトリー内の項目のデータベース注釈が変更されます。Windows オペレーティング・システムでは、上記の指定法に代えて、データベースが存在するドライブ名を指定してもかまいません (特定のパス上ではなく、ドライブ上に作成されたものの場合)。

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

システム・データベース・ディレクトリーまたはローカル・データベース・ディレクトリー内の項目について記述します。カタログしたデータベースについての記述を補足する、あらゆるコメントを入力することができます。コメント・ストリングの最大長は 30 文字です。復帰文字や改行文字は許可されません。コメント・テキストは必ず二重引用符で囲んでください。

## CHANGE DATABASE COMMENT

### 例

以下は、SAMPLE データベースのシステム・データベース・ディレクトリー注釈テキストを、"Test 2 - Holding" から "Test 2 - Add employee inf rows" に変更する例です。

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

### 使用上の注意

既存のコメント・テキストは、新規のテキストに置き換えられます。情報を追加する場合、既存のコメント・テキストに続けて新規テキストを入力してください。

データベース別名と関連する項目のコメントだけが修正されます。データベース名が同じでも、別名が異なるその他の項目には影響しません。

パスを指定する場合、データベース別名をローカル・データベース・ディレクトリーにカタログしてください。また、パスを指定しない場合は、データベース別名をシステム・データベース・ディレクトリーにカタログしてください。

## CHANGE ISOLATION LEVEL

データベースのアクセス中に、DB2 がデータを他のプロセスから分離する方法を変更します。

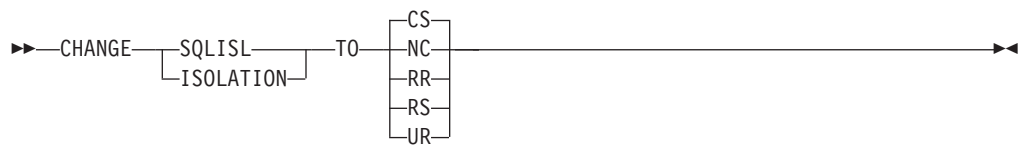
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### TO

- CS** カーソル固定を分離レベルとして指定します。
- NC** コミットを分離レベルとして指定しません。DB2 ではサポートされていません。
- RR** 反復可能読み取りを分離レベルとして指定します。
- RS** 読み取り固定を分離レベルとして指定します。
- UR** 非コミット読み取りを分離レベルとして指定します。

### 使用上の注意

DB2 は、分離レベルを使用して、データベース中でデータの整合性を維持します。分離レベルは、並行して実行される他のアプリケーション処理によって加えられる変更から、アプリケーション処理が分離 (シールド) される程度を定義します。

選択された分離レベルがデータベースでサポートされていない場合、接続時に、サポートされているレベルまで自動的にエスカレートされます。

タイプ 1 の接続でデータベースへ接続中に、分離レベルを変更することは許可されていません。バックエンド処理は、分離レベルを変更する前に以下のようにして終了しなければなりません。

```

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

タイプ 2 の接続を使用した変更は許可されますが、変更は同じコマンド行プロセッサのバックエンド処理からのすべての接続に適用されるので、注意が必要です。ユーザーは、どの分離レベルが、接続されたどのデータベースに適用するか、記憶している責任があるということを前提とします。

## CHANGE ISOLATION LEVEL

次の例では、SAMPLE データベースの作成に続いて、ユーザーが DB2 対話式モードに入っています。

```
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
```

c1 はこの分離レベルに対応した PREPARE 状態になっていないため、SQL0514N エラーが発生します。

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

c1 はこのデータベースに対応した PREPARE 状態になっていないため、SQL0514N エラーが発生します。

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

カーソル c1 は既に宣言されて OPEN しているため、DB21029E エラーが発生します。

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

この場合は、元のデータベース (SAMPLE) が元の分離レベル (CS) で使用されたので、うまくいきます。

## COMPLETE XMLSCHEMA

XML スキーマを XML スキーマ・リポジトリ (XSR) に登録するプロセスを完了します。

### 許可

- ユーザー ID は、カタログ・ビュー SYSCAT.XSROBJECTS で記録されたとおりに XSR オブジェクトの所有者でなければなりません。

### 必要な接続

データベース

### コマンド構文

```

▶▶ COMPLETE XMLSCHEMA relational-identifier
    └── WITH schema-properties-URI ───┘
└── ENABLE DECOMPOSITION ───┘▶▶
  
```

### 説明

#### *relational-identifier*

以前に **REGISTER XMLSCHEMA** コマンドで登録された XML スキーマのリレーショナル名を指定します。リレーショナル名は 2 つの部分の SQL ID として指定することができます。これは、SQL スキーマと XML スキーマ名から成り、*SQLschema.name* というフォーマットを持ちます。スキーマが指定されない場合、**CURRENT SCHEMA** 特殊レジスターで定義されたとおりに、デフォルト SQL スキーマが使用されます。

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

XML スキーマのプロパティ文書の Uniform Resource Identifier (URI) を指定します。ファイル・スキーム URI で指定されたローカル・ファイルだけがサポートされます。スキーマ・プロパティ文書は、XML スキーマ登録の完了段階でのみ指定できます。

#### **ENABLE DECOMPOSITION**

スキーマを XML インスタンス文書の分解に使用できることを示します。

### 例

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

### 使用上の注意

XML スキーマ登録プロセスが完了するまで、XML スキーマを参照したり、妥当性検査またはアノテーションに使用することはできません。このコマンドは、

**REGISTER XMLSCHEMA** コマンドで開始された XML スキーマの XML スキーマ登録プロセスを完了します。

---

### CREATE DATABASE

オプションでユーザー定義の照合シーケンスを持つ新規データベースを初期設定し、3つの初期表スペースやシステム表を作成し、さらにはリカバリー・ログ・ファイルを割り当てます。新規のデータベースの初期化時には、**AUTOCONFIGURE** コマンドがデフォルトで発行されます。

注: インスタンスおよびデータベース・ディレクトリーが DB2 データベース・マネージャーにより作成される時、権限は正確であるため変更すべきではありません。

**CREATE DATABASE** コマンドを発行すると、構成アドバイザーも自動的に実行されます。つまり、システム・リソースに合わせてデータベース構成パラメーターが自動的に調整されるということです。さらに、自動 **Runstats** が有効になります。データベースの作成時に構成アドバイザーが稼働しないようにするには、**DB2\_ENABLE\_AUTOCONFIG\_DEFAULT** レジストリー変数を参照してください。自動 **Runstats** を無効にするには、**auto\_runstats** データベース構成パラメーターを参照してください。

単一パーティション・データベースの場合は、適応セルフチューニング・メモリーもデフォルトで有効になっています。デフォルトで適応セルフチューニング・メモリーが無効になるようにするには、**self\_tuning\_mem** データベース構成パラメーターを参照してください。複数パーティション・データベースの場合、適応セルフチューニング・メモリーはデフォルトで無効になっています。

コード・セットを **CREATE DATABASE** コマンドに指定しない場合、使用可能な照合は、SYSTEM、IDENTITY\_16BIT、*language-aware-collation*、および *locale-sensistive-collation* です (SQLCODE -1083)。データベースのデフォルト・コード・セットは UTF-8 です。あるデータベースのために特定のコード・セットおよびテリトリーが必要な場合、**CREATE DATABASE** コマンドにおいて、必要なコード・セットおよびテリトリーを指定する必要があります。

このコマンドはクライアントでは無効です。

#### 有効範囲

パーティション・データベース環境では、このコマンドは、*db2nodes.cfg* ファイルにリストされているすべてのデータベース・パーティションに影響を与えます。

このコマンドを発行したデータベース・パーティションは、新規データベースのカタログ・データベース・パーティションになります。

#### 許可

以下の権限のいずれかが必要です。

- SYSADM
- SYSCTRL

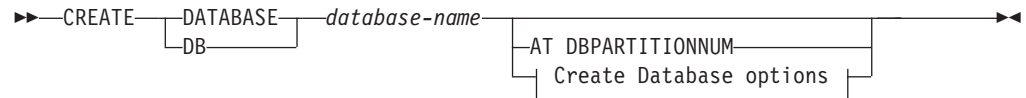
#### 必要な接続

インスタンス。別の (リモート) データベース・パーティション・サーバーでデータベースを作成するには、まずそのサーバーにアタッチする必要があります。このコ

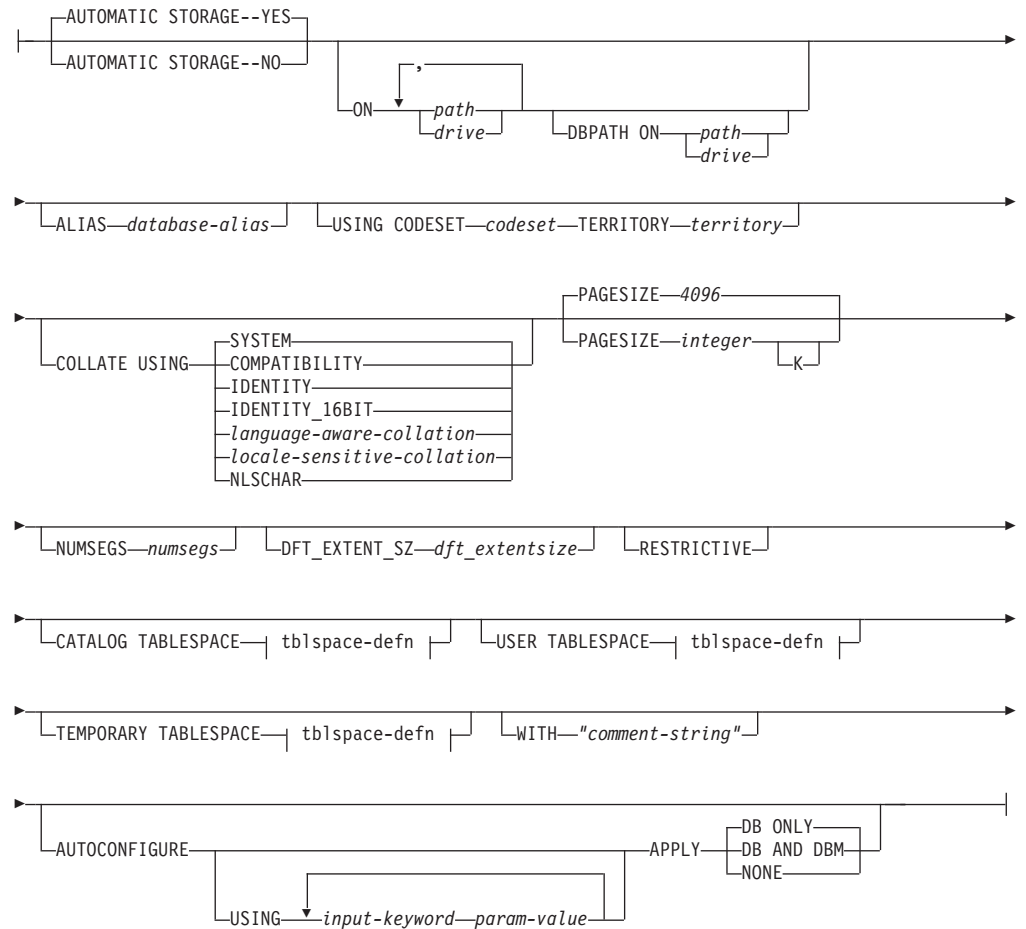


マンドの処理中、データベース接続が一時的に確立されます。

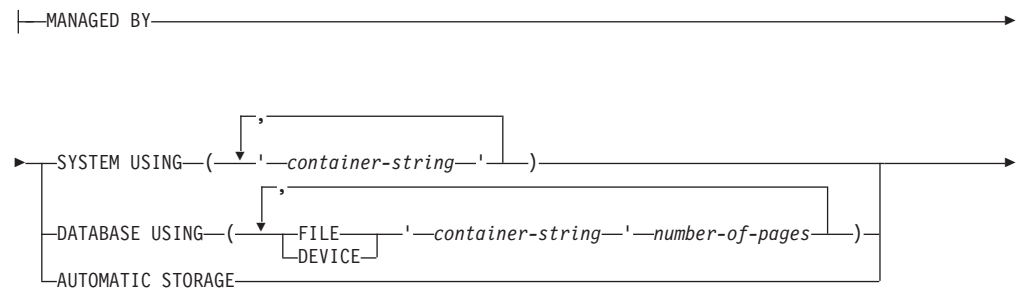
コマンド構文



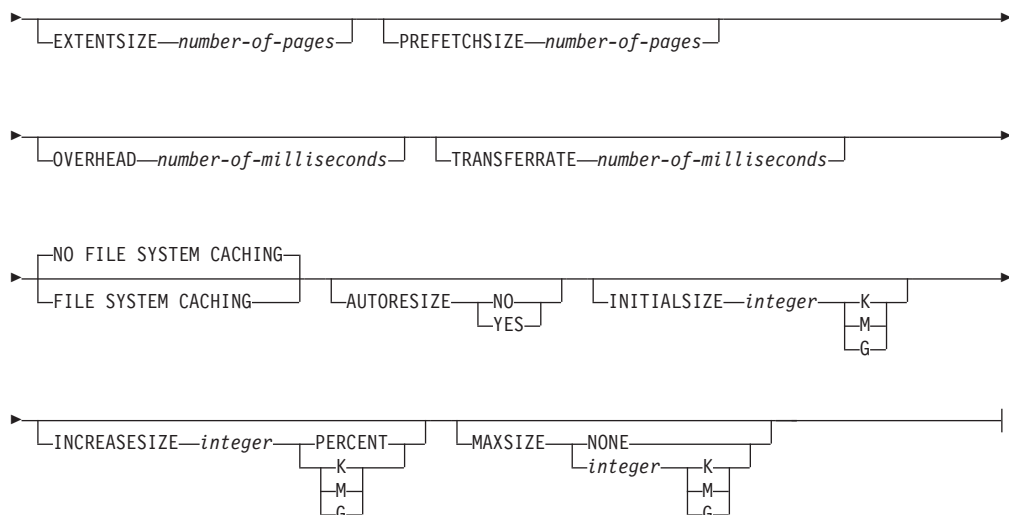
Create Database options:



tblspace-defn:



## CREATE DATABASE



### 注:

1. 指定されるコード・セットおよびテリトリー値の組み合わせは有効なものではありません。
2. コード・セットとテリトリーのどんな組み合わせでも照合順序が有効になるとは限りません。
3. **CREATE DATABASE** で指定した表スペース定義は、データベースを作成しているすべてのデータベース・パーティションに適用されます。定義をデータベース・パーティションごとに個別に指定することはできません。表スペース定義を特定のデータベース・パーティションごとに異なるものにして作成する場合、**CREATE TABLESPACE** ステートメントを使用しなければなりません。

表スペースにコンテナを定義する場合、**\$N** を使用できます。**\$N** は、コンテナが実際に作成されるときにデータベース・パーティション番号で置き換えられます。これは、ユーザーが複数の論理パーティション・データベースでコンテナを指定する場合に必要です。

4. **AUTOCONFIGURE** パラメーターには **SYSADM** 権限が必要です。

## コマンド・パラメーター

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

新しいデータベースに割り当てられる名前。ローカル・データベース・ディレクトリーまたはシステム・データベース・ディレクトリーの他のデータベースから、そのデータベースを区別する、固有の名前でなければなりません。名前は、データベースの命名規則に適合していることが必要です。特に、名前にはスペース文字を含めてはなりません。

### **AT DBPARTITIONNUM**

データベースが、コマンドを発行したデータベース・パーティションでのみ作成されることを指定します。新規のデータベースを作成するときには、このパラメーターを指定しないでください。これは損傷したためにドロップしたデータベース・パーティションを再作成するために使用できます。**AT DBPARTITIONNUM** パラメーターを指定した **CREATE DATABASE** コマンドを使用

した後、このデータベース・パーティションにあるデータベースはリストア・ペンディング状態になります。このデータベース・パーティション・サーバーのデータベースを即時にリストアしなければなりません。このパラメーターは、通常の使用を意図したものではありません。例えば、あるデータベース・パーティション・サーバーのデータベース・パーティションが損傷して再作成が必要な場合に、**RESTORE DATABASE** コマンドとともに使用する必要があります。このパラメーターの使用が不適切であると、システム内に不整合が生じることもありえますので、ご使用の際には十分注意してください。

損傷したためにドロップされたデータベース・パーティションを再作成する目的でこのパラメーターを使用した場合、このデータベース・パーティションのデータベースはリストア・ペンディング状態になります。データベース・パーティションを再作成した後、このデータベース・パーティション上のデータベースをただちにリストアする必要があります。

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

データベースの自動ストレージを明示的に無効または有効にすることを指定します。デフォルト値は、**YES** です。**AUTOMATIC STORAGE** 節を指定しなくても、自動ストレージは、デフォルトで暗黙で有効になります。

**NO** データベースの自動ストレージは有効ではありません。このパラメーターは、DB2 pureScale 環境では指定できません。

**YES** データベースの自動ストレージが有効です。デフォルトのストレージ・グループ **IBMSTOGROUP** が、**SYSSTOGROUPS** カタログ表に作成されます。ストレージ・グループを変更するには、**ALTER STOGROUP** ステートメントを使用します。

**重要:** このパラメーターは推奨されなくなりました。今後のリリースで除去される可能性があります。除去された場合は、**AUTOMATIC STORAGE YES** が唯一のオプションとなります。

#### **ON path または drive**

このパラメーターの意味は、**AUTOMATIC STORAGE** パラメーターの値に応じて異なります。

- **AUTOMATIC STORAGE NO** を指定した場合、データベースの自動ストレージは無効になります。この場合、**ON** パラメーターにはパスを 1 つしか含めることができず、そのパスは、データベースを作成するパスを示します。パスを指定しないと、データベースはデータベース・マネージャー構成ファイル (**dftdbpath** パラメーター) に指定されているデフォルトのデータベース・パスに作成されます。
- このオプションを指定しなかった場合、データベースの自動ストレージはデフォルトで有効になります。この場合、複数のパスをコマンドで区切ってリストすることができます。これらは、デフォルト・ストレージ・グループ **IBMSTOGROUP** に定義されたストレージ・パスとして参照され、自動ストレージ表スペース用の表スペース・コンテナを保持するために使用されます。複数パーティション・データベースの場合、すべてのパーティションで同じストレージ・パスが使用されます。

## CREATE DATABASE

**DBPATH ON** パラメーターは、データベースを作成するパスを指定します。**DBPATH ON** パラメーターが指定されていない場合、データベースは **ON** パラメーターのリストの最初のパスに作成されます。**ON** パラメーターでパスが指定されていない場合、データベース・マネージャー構成ファイルで指定されているデフォルトのデータベース・パス (**dftdbpath** パラメーター) にデータベースが作成されます。また、これはデフォルトのストレージ・グループに関連付けられた単一ストレージ・パスの場所としても使用されます。

データベース・パスは、階層ディレクトリー構造が作成される場所です。この構造は、データベースの操作に必要な以下のファイルを持します。

- バッファ・プール情報
- 表スペース情報
- ストレージ・パス情報
- データベース構成情報
- バックアップ、リストア、表のロード、表の再編成、表スペースの変更、および他のデータベース変更に関する履歴ファイル情報。
- アクティブ・ログに関する情報があるログ制御ファイル

**DBPATH ON** パラメーターは、これらのファイルと情報を、データベース・データが保持されているストレージ・パスとは別のディレクトリー内に置くために使用できます。自動ストレージが有効な場合は、**DBPATH ON** パラメーターを使用して、データベース情報をデータベース・データと分けて保持することが提案されています。

パスの最大長は 175 文字です。

パーティション・データベース環境では、データベースを NFS マウント・ディレクトリーに作成しないようにしてください。パスを指定しない場合、**dftdbpath** データベース・マネージャー構成パラメーターが NFS マウント・パスに設定されていないことを確認してください (例えば、UNIX オペレーティング・システムの場合、パラメーターがインスタンス所有者の \$HOME ディレクトリーを指定しないようにします)。パーティション・データベース環境では、このコマンドに相対パスを指定することはできません。また、**ON** パラメーターに指定するすべてのパスは、すべてのデータベース・パーティションに存在していなければなりません。

そのようなデータベース・パスまたはストレージ・パスは、存在していなければならず、各データベース・パーティション上でアクセス可能でなければなりません。

### **DBPATH ON** *path or drive*

自動ストレージが有効な場合に、**DBPATH ON** パラメーターは、データベース作成先のパスを指定します。自動ストレージが有効で **DBPATH ON** パラメーターが指定されていない場合、データベースは **ON** パラメーターのリストの最初のパスに作成されます。

データベース・パスの最大長は 215 文字、ストレージ・パスの最大長は 175 文字です。

**ALIAS** *database-alias*

システム・データベース・ディレクトリーのデータベースに付けられる別名。別名が付けられないと、指定されたデータベース名が使用されます。

**USING CODESET** *codeset*

このデータベースに入るデータに使用するコード・セットを指定します。データベースを作成した後は、指定のコード・セットを変更できません。

**TERRITORY** *territory*

このデータベースに入るデータに使用するテリトリー ID またはロケール ID を指定します。データベースを作成した後は、指定のテリトリーを変更できません。指定されるコード・セットおよびテリトリーまたはロケール値の組み合わせは有効なものでなければなりません。

**COLLATE USING**

データベースに使用する照合順序のタイプを識別します。一度データベースが作成されてしまうと、照合順序を変更することはできません。

Unicode データベースでは、カタログ表およびビューが、**COLLATE USING** 節に指定された照合とは関係なく、常に **IDENTITY** 照合によって作成されます。非 Unicode データベースでは、カタログ表およびビューがデータベース照合によって作成されます。

**COMPATIBILITY**

DB2 バージョン 2 の照合順序です。一部の照合表が拡張されています。このパラメーターは、これらの表の前のバージョンを使用するように指定します。

**IDENTITY**

ストリングがバイト単位で比較される一致照合順序。

**IDENTITY\_16BIT**

Unicode Technical Report #26 で指定された、CESU-8 (Compatibility Encoding Scheme for UTF-16: 8-Bit) 照合順序。この仕様は Unicode Consortium Web サイト ([www.unicode.org](http://www.unicode.org)) で入手できます。このパラメーターは、Unicode データベースを作成する場合に限り指定できます。

*language-aware-collation*

このパラメーターは、Unicode データベースにのみ使用できます。データベース照合順序は、非 Unicode データベースの **SYSTEM** 照合に基づきます。このストリングの形式は **SYSTEM\_codepage\_territory** でなければなりません。提供されるストリングが無効の場合、データベースの作成は失敗します (SQLCODE -204; object not found)。詳細について、およびシステム・ベースの照合の命名については、『Unicode データの言語対応型の照合』を参照してください。

**注:** **CREATE DATABASE** コマンドをバージョン 9.0 サーバーに対して実行する場合、このパラメーターは使用できません。デフォルトでは、そうしたサーバー上の Unicode データベースは **SYSTEM** 照合を使用して作成されます。

## CREATE DATABASE

### *locale-sensitive-collation*

このパラメーターは、Unicode データベースにのみ使用できます。詳細については、およびロケールを区別する UCA ベースの照合の命名については、『Unicode 照合アルゴリズムに基づく照合』を参照してください。指定された照合名が無効の場合、**CREATE DATABASE** コマンドの実行は失敗します (SQLCODE -204)。

### **NLSCHAR**

特定のコード・セット/テリトリー用の固有な照合規則を使用する組み込みの照合シーケンス。

このパラメーターは、タイ語コード・ページ (CP874) でのみ使用できます。このパラメーターを非タイ語環境で指定すると、コマンドは失敗し、エラー SQL1083N と理由コード 4 が返されます。

**SYSTEM** これは、データベースを作成するときのデフォルト・パラメーターです。Unicode 以外のデータベースの場合、照合シーケンスはデータベース・テリトリーをベースにします。Unicode データベースの場合、このパラメーターはクライアントのコード・セットおよびテリトリーをベースとする言語認識照合にマップされます。適切な言語認識照合が使用できない場合、IDENTITY 照合が使用されます。

### **PAGESIZE** *integer*

データベース作成時のデフォルト・バッファー・プールのページ・サイズ、および初期表スペース (SYSCATSPACE、TEMPSPACE1、USERSPACE1) を指定します。また、今後のすべての CREATE BUFFERPOOL および CREATE TABLESPACE ステートメントのデフォルト・ページ・サイズも表します。接尾部 K を持たない integer の有効値は、4 096、8 192、16 384 または 32 768 です。接尾部 K を持つ integer の有効値は、4、8、16、または 32 です。整数と接尾部 K の間には、スペースが少なくとも 1 つ必要です。デフォルトのページ・サイズは 4 096 バイト (4 K) です。

### **NUMSEGS** *numsegs*

すべてのデフォルトの SMS 表スペース用のデータベース表ファイルを保管するために作成および使用されるディレクトリー数 (表スペース・コンテナ) を指定します。このパラメーターは、自動ストレージ表スペース、DMS 表スペース、作成特性が明示的に指定された SMS 表スペース (データベース作成時に作成される)、またはデータベース作成後に明示的に作成された SMS 表スペースには影響を与えません。

### **DFT\_EXTENT\_SZ** *dft\_extentsize*

データベース内の表スペースのデフォルト・エクステント・サイズを指定します。

### **RESTRICTIVE**

**RESTRICTIVE** パラメーターを使用すると、**restrict\_access** データベース構成パラメーターが YES に設定されるため、特権や権限が自動的に PUBLIC に付与されることがなくなります。**RESTRICTIVE** パラメーターを使用しない場合は、**restrict\_access** データベース構成パラメーターが NO に設定されるため、特権が自動的に PUBLIC に付与されます。特権については、『データベースの作成時に付与されるデフォルト特権』を参照してください。



**CATALOG TABLESPACE** *tblspace-defn*

カタログ表を保持する表スペース SYSCATSPACE の定義を指定します。これを指定せず、データベースの自動ストレージが有効でない場合には、**NUMSEGS** 個のディレクトリーをコンテナーとして持ち、エクステント・サイズが **DFT\_EXTENTSIZE** である SYSCATSPACE がシステム管理スペース (SMS) 表スペースとして作成されます。例えば、**NUMSEGS** に 5 を指定した場合、以下のコンテナーが作成されます。

```
/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
```

これを指定せず、データベースの自動ストレージが有効の場合、SYSCATSPACE は、自動ストレージ表スペースとして作成され、定義済みのストレージ・パスにコンテナーが作成されます。この表スペースのエクステント・サイズは 4 です。**AUTORESIZE**、**INITIALSIZE**、**INCREASESIZE**、および **MAXSIZE** の適切な値が自動的に設定されます。

表スペース定義フィールドについての詳細は、『CREATE TABLESPACE ステートメント』を参照してください。

パーティション・データベース環境の場合、カタログ表スペースはカタログ・データベース・パーティション (**CREATE DATABASE** を発行するデータベース・パーティション) でのみ作成されます。

**USER TABLESPACE** *tblspace-defn*

初期ユーザー表スペース USERSPACE1 の定義を指定します。これを指定せず、データベースの自動ストレージが有効でない場合、USERSPACE1 は、**NUMSEGS** 個のディレクトリーをコンテナーとして持ち、エクステント・サイズが **DFT\_EXTENTSIZE** である SMS 表スペースとして作成されます。例えば、**NUMSEGS** に 5 を指定した場合、以下のコンテナーが作成されます。

```
/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
```

これを指定せず、データベースの自動ストレージが有効の場合には、自動ストレージ表スペースとして USERSPACE1 が作成され、定義済みのストレージ・パスにコンテナーが作成されます。この表スペースのエクステント・サイズは **DFT\_EXTENTSIZE** です。**AUTORESIZE**、**INITIALSIZE**、**INCREASESIZE**、および **MAXSIZE** の適切な値が自動的に設定されます。

表スペース定義フィールドについての詳細は、『CREATE TABLESPACE ステートメント』を参照してください。

**TEMPORARY TABLESPACE** *tblspace-defn*

初期 SYSTEM TEMPORARY 表スペース、TEMPSPACE1 の定義を指定します。これを指定せず、データベースの自動ストレージが有効でない場合、TEMPSPACE1 は、**NUMSEGS** 個のディレクトリーをコンテナーとして持ち、エクステント・サイズが **DFT\_EXTENTSIZE** である SMS 表スペースとして作成されます。例えば、**NUMSEGS** に 5 を指定した場合、以下のコンテナーが作成されます。

## CREATE DATABASE

```
/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
```

これを指定せず、データベースの自動ストレージが有効の場合には、自動ストレージ表スペースとして **TEMPSPACE1** が作成され、定義済みのストレージ・パスにコンテナが作成されます。この表スペースのエクステント・サイズは **DFT\_EXTENTSIZE** です。

表スペース定義フィールドについての詳細は、『CREATE TABLESPACE ステートメント』を参照してください。

### *tblspace-defn*

次のコマンド・パラメーターを使用して、様々な表スペース定義を指定できます。DB2 pureScale 環境では、**MANAGED BY AUTOMATIC STORAGE** のみ使用できます。

#### **MANAGED BY**

##### **SYSTEM USING** *container-string*

表スペースが **SMS** 表スペースになることを指定します。表スペースのタイプが指定されていない場合、デフォルトの動作として **REGULAR** 表スペースを作成します。

**重要: USER TABLESPACE** 指定で **MANAGED BY SYSTEM** は非推奨になっており、今後のリリースで除去される可能性があります。代わりに、**MANAGED BY AUTOMATIC STORAGE** を使用してください。

**SMS** 表スペースに関して、表スペースに属し、表スペース・データの保管先となる、1 つ以上のコンテナを識別します。 *container-string* の長さは、240 バイトを超えてはなりません。

各 *container-string* は、絶対ディレクトリー名または相対ディレクトリー名にすることができます。

ディレクトリー名が絶対でない場合は、データベース・ディレクトリーからの相対ディレクトリーになります。ディレクトリー名は、データベース・ディレクトリーに物理的に関連付けられていないストレージのパス名の別名 (UNIX オペレーティング・システムではシンボリック・リンク) にすることもできます。例えば、*dbdir/work/c1* は個別のファイル・システムに対するシンボリック・リンクになることができます。

ディレクトリー名のコンポーネントで存在しないものがあれば、データベース・マネージャーによって作成されます。表スペースがドロップされると、データベース・マネージャーによって作成されたすべてのコンポーネントが削除されます。 *container-string* で識別されるディレクトリーが存在する場合、それにファイルまたはサブディレクトリーを含めてはなりません (SQLSTATE 428B2)。



*container-string* の形式は、オペレーティング・システムによって異なります。Windows オペレーティング・システムの場合、絶対ディレクトリー・パス名はドライブ名とコロン(:) で始まり、UNIX オペレーティング・システムの場合、絶対パス名はスラッシュ(/) で始まります。相対パス名はどのプラットフォームでも、オペレーティング・システムに依存する文字では始まりません。

リモート・リソース (LAN リダイレクト・ドライブまたは NFS マウント・ファイル・システムなど) は現在、Network Appliance Filers、IBM iSCSI、IBM Network Attached Storage、Network Appliance iSCSI、NEC iStorage S2100、S2200、または S4100、あるいは NEC Storage NS Series を Windows DB2 サーバーとともに使用する場合にはのみサポートされます。NEC Storage NS Series は、無停電電源装置 (UPS) とともに使用する場合にはのみサポートされます。(スタンバイではなく) 連続 UPS が推奨されます。AIX 上の NFS マウント・ファイル・システムは、**-o nointr** パラメーターを使用して割り込み不可モードでマウントする必要があります。

#### DATABASE USING

表スペースが DMS 表スペースになることを指定します。表スペースのタイプが指定されていない場合、デフォルトの動作として LARGE 表スペースを作成します。

**重要: USER TABLESPACE** 指定で **MANAGED BY DATABASE** は非推奨になっており、今後のリリースで除去される可能性があります。代わりに、**MANAGED BY AUTOMATIC STORAGE** を使用してください。

DMS 表スペースの場合、表スペースに属し、表スペース・データの保管先となる、1 つ以上のコンテナを識別します。コンテナのタイプ (**FILE** または **DEVICE**) とそのサイズ (**PAGESIZE** ページの数) を指定します。 **FILE** コンテナおよび **DEVICE** コンテナを混合して指定することができます。 *container-string* の長さは、254 バイトを超えてはなりません。

リモート・リソース (LAN リダイレクト・ドライブまたは NFS マウント・ファイル・システムなど) は現在、Network Appliance Filers、IBM iSCSI、IBM Network Attached Storage、Network Appliance iSCSI、NEC iStorage S2100、S2200、または S4100、あるいは NEC Storage NS Series を Windows DB2 サーバーとともに使用する場合にはのみサポートされます。NEC Storage NS Series は、無停電電源装置 (UPS) とともに使用する場合にはのみサポートされます。(スタンバイではなく) 連続 UPS が推奨されます。

すべてのコンテナはすべてのデータベース間で固有でなければなりません。コンテナは 1 つの表スペースにのみ属することができます。コンテナのサイズは異なることがあ

ります。しかし、すべてのコンテナーが同じサイズの場合に最適のパフォーマンスが実現します。 *container-string* の正しい形式は、オペレーティング・システムによって異なります。

### **FILE** *container-string number-of-pages*

FILE コンテナーの場合、*container-string* は、絶対ファイル名または相対ファイル名でなければなりません。ファイル名が絶対ファイル名でない場合、データベース・ディレクトリーに対して相対的です。ディレクトリー名のコンポーネントで存在しないものがある場合は、データベース・マネージャーによって作成されます。ファイルが存在しない場合は作成され、データベース・マネージャーによって指定されたサイズに初期設定されます。表スペースがドロップされると、データベース・マネージャーによって作成されたすべてのコンポーネントが削除されます。

注: ファイルが存在する場合は上書きされ、ファイルが指定されたサイズより小さい場合は拡張されます。ファイルが指定されたサイズより大きい場合には切り捨てられません。

### **DEVICE** *container-string number-of-pages*

DEVICE コンテナーの場合、*container-string* は装置名でなければなりません。また、装置が既に存在していなければなりません。

## **AUTOMATIC STORAGE**

表スペースが自動ストレージ表スペースになることを指定します。ストレージ・グループが定義されていないと、エラーが返されます (SQLSTATE 55060)。

自動ストレージ表スペースは、TEMPORARY 表スペースである場合はシステム管理スペース (SMS) 表スペースとして作成され、永続表スペースである場合はデータベース管理スペース (DMS) 表スペースとして作成されます。DMS 表スペースのタイプが指定されていない場合、デフォルトの動作として LARGE 表スペースが作成されます。自動ストレージ表スペースを使用すると、データベース・マネージャーはデータベースに関連付けられたストレージ・パスに基づいて、表スペースに割り当てられるコンテナーを判別します。

### **EXTENTSIZE** *number-of-pages*

次のコンテナーに移る前にコンテナーに書き込まれる

**PAGESIZE** ページの数を指定します。このエクステント・サイズ値は、後に K (K バイトの場合)、または M (M バイトの場合) を付けた整数値として指定することもできます。このように指定した場合、バイト数をページ・サイズで割った値を下限に丸めたものを使用してエクステント・サイズの値

が決定されます。データベース・マネージャーは、データが保管されると、コンテナについてこの処理を繰り返し実行します。

デフォルト値は **dft\_extent\_sz** データベース構成パラメーターによって提供されます。その有効範囲は 2 から 256 ページです。

#### **PREFETCHSIZE** *number-of-pages*

データのプリフェッチの実行中に、表スペースから読み取られる **PAGESIZE** ページの数を指定します。このプリフェッチ・サイズ値は、後に K (K バイトの場合)、M (M バイトの場合)、または G (G バイトの場合) を付けた整数値としても指定できます。このように指定した場合、バイト数をページ・サイズで割った値を下限に丸めたものを使用してプリフェッチ・サイズのページ数の値が決定されます。

#### **OVERHEAD** *number-of-milliseconds*

入出力コントローラー使用量、ディスク・シーク、および待ち時間を指定するミリ秒単位の数値。この値を使用して、照会最適化時の入出力のコストを判別します。

*number-of-milliseconds* の値は数値リテラル (整数、10 進数、または浮動小数点数) です。この値がすべてのコンテナに同じでなければ、数値は表スペースに属するすべてのコンテナの平均になるはずです。

バージョン 9 以上で作成されたデータベースの場合、デフォルトの入出力コントローラーの使用量、ディスク・シーク、および待ち時間は 7.5 ミリ秒です。DB2 の前のバージョンからバージョン 9 以上にアップグレードされたデータベースの場合、デフォルトは 12.67 ミリ秒です。

#### **TRANSFERRATE** *number-of-milliseconds*

1 ページをメモリーに読み込むための時間を指定します。この値を使用して、照会最適化時の入出力のコストを判別します。 *number-of-milliseconds* の値は数値リテラル (整数、10 進数、または浮動小数点数) です。この値がすべてのコンテナに同じでなければ、数値は表スペースに属するすべてのコンテナの平均になるはずです。

バージョン 9 以上で作成されたデータベースの場合、1 ページをメモリーに読み込むためのデフォルト時間は 0.06 ミリ秒です。DB2 の前のバージョンからバージョン 9 以上にアップグレードされたデータベースの場合、デフォルトは 0.18 ミリ秒です。

#### **NO FILE SYSTEM CACHING**

すべての入出力操作がファイル・システム・レベルのキャッシュを迂回することを指定します。詳しくは、『ファイル・システム・キャッシングを使用しない表スペース』を参照してください。これは、ほとんどの構成におけるデフォルト・パラメーターです。詳細については、『ファイル・システム・キャッシング構成』を参照してください。

### FILE SYSTEM CACHING

ターゲット表スペースでのすべての入出力操作がファイル・システム・レベルでキャッシュに入れられることを指定します。詳しくは、『ファイル・システム・キャッシングを使用しない表スペース』を参照してください。これは、一部の構成におけるデフォルト・パラメーターです。詳細については、『ファイル・システム・キャッシング構成』を参照してください。

### AUTORESIZE

DMS 表スペースまたは自動ストレージ表スペースの自動サイズ変更機能が有効かどうかを指定します。自動サイズ変更可能表スペースは、いっぱいになると、サイズを自動的に大きくします。デフォルトは、DMS 表スペースの場合は NO、自動ストレージ表スペースの場合は YES です。

**NO** DMS 表スペースまたは自動ストレージ表スペースの自動サイズ変更機能が無効であることを指定します。

**YES** DMS 表スペースまたは自動ストレージ表スペースの自動サイズ変更機能が有効であることを指定します。

### INITIALSIZE *integer*

自動ストレージ表スペースの初期サイズをデータベース・パーティションごとに指定します。このパラメーターは、自動ストレージ表スペースに対してのみ有効です。整数値の後に K (キロバイト)、M (メガバイト)、または G (ギガバイト) を指定する必要があります。使用される実際の値は指定されたものより多少小さい場合があることに注意してください。これは、データベース・マネージャーが表スペース内のコンテナ間で整合したサイズを維持しようとするためです。さらに自動サイズ変更可能な表スペースで、初期サイズの大きさが不足しており、新規表スペースに追加しなければならないメタデータを入れることができない場合、データベース・マネージャーは十分なスペースになるまで **INCREASESIZE** の値によって表スペースの拡張を続けます。 **INITIALSIZE** 節が指定されていない場合、データベース・マネージャーが適切な値を判別します。 *integer* の値は、少なくとも 48 K でなければなりません。

**K** K (キロバイト)。

**M** M (メガバイト)。

**G** G (ギガバイト)。

### INCREASESIZE *integer*

自動サイズ変更が有効な表スペースで、表スペースがいっぱいでスペース要求が出された場合に表スペース・サイズが自

動変更されるときサイズの増加単位 (データベース・パーティションごと) を指定します。整数値の後に次のいずれかを続ける必要があります。

- PERCENT。スペースの要求がなされた時点の表スペース・サイズのパーセンテージとして量を指定します。PERCENT を指定する場合、整数値は 0 と 100 の間でなければなりません (SQLSTATE 42615)。
- K (K バイト)、M (M バイト)、または G (G バイト)。バイト単位で量を指定します。

使用される実際の値は指定されたものより多少増減する場合があります。これは、データベース・マネージャーが表スペース内のコンテナ間で整合した増加量を維持しようとするためです。表スペースが自動サイズ変更可能であっても、**INCREASESIZE** 節が指定されていない場合、データベース・マネージャーが適切な値を判別します。

#### PERCENT

0 から 100 までのパーセント。

**K** K (キロバイト)。

**M** M (メガバイト)。

**G** G (ギガバイト)。

#### MAXSIZE

自動サイズ変更が有効な表スペースで、自動的に増加可能な最大サイズを指定します。表スペースが自動サイズ変更可能であっても、**MAXSIZE** 節が指定されていない場合、デフォルトは NONE です。

**NONE** 表スペースがファイル・システム容量または最大表スペース・サイズまで増加できることを指定します。

*integer* **DMS** 表スペースまたは自動ストレージ表スペースが自動的に増加できるサイズのハード・リミットを、データベース・パーティションごとに指定します。整数値の後に K (キロバイト)、M (メガバイト)、または G (ギガバイト) を指定する必要があります。使用される実際の値は指定されたものより多少小さい場合があります。これは、データベース・マネージャーが表スペース内のコンテナ間で整合した増加量を維持しようとするためです。

**K** K (キロバイト)。

**M** M (メガバイト)。

**G** G (ギガバイト)。

#### WITH *comment-string*

データベース・ディレクトリー内のデータベース項目について記述します。

## CREATE DATABASE

そのデータベースについての記述を補足する、任意の注釈を入力することができます。最大長は 30 文字です。復帰文字や改行文字は許可されません。注釈テキストは、単一引用符または二重引用符で囲む必要があります。

### AUTOCONFIGURE

ユーザー入力に基づいて、バッファ・プール・サイズ、データベース構成、およびデータベース・マネージャー構成の推奨設定値を計算します (オプションでこれを適用することもできます)。**CREATE DATABASE** コマンドを発行すると、デフォルトで構成アドバイザーが実行されます。

**AUTOCONFIGURE** パラメーターが必要なのは、推奨内容を調整する場合のみです。

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

表 9. 有効な入力キーワードおよびパラメーター値

キーワード	有効値	デフォルト値	説明
mem_percent	1 から 100	25	データベースに割り当てられているインスタンス・メモリーのパーセンテージ。ただし、 <b>CREATE DATABASE</b> コマンドによって構成アドバイザーが呼び出され、 <b>mem_percent</b> の値を指定しない場合、インスタンスとシステムのメモリー使用量に基づいてパーセンテージが計算され、最大でインスタンス・メモリーの 25% になります。
workload_type	simple、mixed、complex	mixed	単純 (simple) ワークロードは入出力集中の傾向があり大部分がトランザクションであるのに対し、複雑 (complex) ワークロードは CPU 集中の傾向があり大部分が照会です。
num_stmts	1 から 1,000,000	25	作業単位ごとのステートメント数
tpm	1 から 200,000	60	1 分ごとのトランザクション

表 9. 有効な入力キーワードおよびパラメーター値 (続き)

キーワード	有効値	デフォルト値	説明
admin_priority	performance、 recovery、 both	both	より良いパフォーマンス (分あたりのより多いトランザクション数) またはより良いリカバリー時間のための最適化
num_local_apps	0 から 5,000	0	接続されたローカル・アプリケーションの数
num_remote_apps	0 から 5,000	100	接続されたリモート・アプリケーションの数
isolation	RR、RS、 CS、 UR	RR	このデータベースに接続するアプリケーションの分離レベル (反復可能読み取り (RR)、読み取り固定 (RS)、カーソル固定 (CS)、非コミット読み取り (UR))
bp_resizeable	yes、 no	yes	バッファ・プールのサイズが変更可能かどうか。

**APPLY****DB ONLY**

現行のデータベース・マネージャーの構成に基づいて、データベース構成およびバッファ・プール設定の推奨値を表示します。データベース構成およびバッファ・プール設定に対して推奨される変更を適用します。

**DB AND DBM**

データベース・マネージャー構成、データベース構成、およびバッファ・プール設定に対して推奨される変更を、表示および適用します。

**NONE** 構成アドバイザー (デフォルトでは有効) を無効にします。

- **CREATE DATABASE** コマンドに **AUTOCONFIGURE** キーワードを指定した場合、**DB2\_ENABLE\_AUTOCONFIG\_DEFAULT** 変数値は検討の対象になりません。適応セルフチューニング・メモリーおよび自動 Runstats が有効になり、構成アドバイザーは、データベース構成パラメーターおよびデータベース・マネージャー構成パラメーターを、**APPLY DB** または **APPLY DBM** パラメーターによって示されているように、調整します。
- データベースに対する **CREATE DATABASE** コマンドに **AUTOCONFIGURE** パラメーターを指定すると、セルフチューニング・メモリー・マネージャーを有効にすることが勧められます。ただし、**sheapthres** がゼロではないインスタンス内で、あるデータベースに対して **AUTOCONFIGURE** コマンドを



## CREATE DATABASE

実行した場合、ソート・メモリー・チューニング (**sortheap**) は自動的に有効になりません。ソート・メモリー・チューニング (**sortheap**) を有効にするには、**UPDATE DATABASE MANAGER CONFIGURATION** コマンドを使用して、**sheapthres** をゼロに設定する必要があります。**sheapthres** の値を変更すると、これまでの既存データベース内のソート・メモリーの使用に影響を与える可能性があることに注意してください。

### 例

**CREATE DATABASE** コマンドの例を以下にいくつか示します。

例 1:

```
CREATE DATABASE TESTDB3
```

データベース・マネージャー構成パラメーター **dftdbpath** の値であるドライブに、データベース **TESTDB3** が作成されます。デフォルトで、ストレージ・グループ **IBMSTOGROUP** がパス **dftdbpath** に作成されます。

例 2:

```
CREATE DATABASE TESTDB7 ON C:,D:
```

データベース **TESTDB7** がドライブ **C:** (ストレージ・パス・リストの最初のドライブ) に作成されます。ストレージ・グループ **IBMSTOGROUP** のストレージ・パスは、**C:** および **D:** となります。

例 3:

```
CREATE DATABASE TESTDB15  
ON C:,D: DBPATH ON E:
```

(**DBPATH** として明示的にリストされた) ドライブ **E:** にデータベース **TESTDB15** が作成されます。ストレージ・グループ **IBMSTOGROUP** のストレージ・パスは、**C:** および **D:** となります。

### 使用上の注意

**CREATE DATABASE** コマンド:

- 指定されたサブディレクトリーにデータベースを作成します。パーティション・データベース環境では、**db2nodes.cfg** にリストされたすべてのデータベース・パーティションにデータベースを作成し、各データベース・パーティションの指定されたサブディレクトリーの下に **\$DB2INSTANCE/NODExxxx** ディレクトリーを作成します。単一パーティション・データベース環境では、指定されたサブディレクトリーの下に **\$DB2INSTANCE/NODE0000** ディレクトリーを作成します。
- システム・カタログ表およびリカバリー・ログを作成します。
- 次のデータベース・ディレクトリーにデータベースをカタログします。
  - *path* で示されたパス上のサーバーのローカル・データベース・ディレクトリー。ただし、パスが指定されていない場合には、**dftdbpath** パラメーターによってデータベース管理システム構成ファイルで定義されたデフォルトのデータベース・パス上のサーバーのローカル・データベース・ディレクトリー。ローカル・データベース・ディレクトリーは、データベースが入っている各ファイル・システムに常駐しています。



- アタッチしたインスタンスのサーバーのシステム・データベース・ディレクトリー。結果のディレクトリー項目には、データベース名とデータベース別名が入ることになります。

コマンドがリモート・クライアントから発行された場合、クライアントのシステム・データベース・ディレクトリーもデータベース名と別名で更新されません。

システムまたはローカル・データベース・ディレクトリーがどちらも存在しない場合に作成します。指定されていれば、注釈およびコード・セット値は両方のディレクトリーに入れられます。

**注:** データベース構成パラメーター **newlogpath** が設定されていない場合、ログ・ファイル構成パラメーター **logpath** のデフォルトの場所は、**DBPATH ON** パラメーターによって示されたパスです。自動ストレージが有効な場合は、**DBPATH ON** パラメーターを使用して、データベース情報をデータベース・データと分けて保持することが提案されています。

- 指定されたコード・セット、テリトリー、および照合順序を保管します。照合順序が固有な重みで構成される場合、またはそれが識別順序である場合、データベース構成ファイルにフラグが設定されます。
- **SYSIBM** を所有者として、**SYSCAT**、**SYSFUN**、**SYSIBM**、および **SYSSTAT** というスキーマを作成します。このコマンドを発行したデータベース・パーティション・サーバーは、新規データベースのカatalog・データベース・パーティションになります。2つのデータベース・パーティション・グループ (**IBMDEFAULTGROUP** および **IBMCATGROUP**) が自動的に作成されます。
- 以前に定義されたデータベース管理バインド・ファイルをデータベースにバインドします (このリストは、ユーティリティーのバインド・ファイル・リスト **db2ubind.lst** にあります)。これらのファイルの1つ以上が正常にバインドされない場合、**CREATE DATABASE** は **SQLCA** に警告を返し、失敗したバインドについての情報を提供します。バインドが失敗した場合、ユーザーは修正処置を行った後、失敗したファイルを手動でバインドできます。データベースはどのような場合にでも作成されます。**RESTRICTIVE** パラメーターを選択しない場合、**PUBLIC** に付与された **CREATEIN** 特権を使ってバインドを実行すると、**NULLID** というスキーマが暗黙的に作成されます。

ユーティリティー・バインド・ファイル・リストには、前のバージョンのサーバーに対してバインドできない2つのバインド・ファイルが含まれています。

- **db2ugtpi.bnd** は、DB2 バージョン 2 サーバーに対してバインドすることができません。
- **db2dropv.bnd** は、DB2 パラレル・エディション バージョン 1 サーバーに対してバインドすることができません。

**db2ubind.lst** が最新レベルでないサーバーに対してバインドされていると、これらの2つのファイルに関する警告が戻されますが、これは無視することができます。

- **SYSCATSPACE**、**TEMPSPACE1**、および **USERSPACE1** 表スペースを作成します。**SYSCATSPACE** 表スペースはカatalog・データベース・パーティションでのみ作成されます。

## CREATE DATABASE

- データベースの作成時に付与される特権については、『データベースの作成時に付与されるデフォルト特権』を参照してください。

自動ストレージとは、コンテナ定義を明示的に指定せずに表スペースを作成できる、ストレージ・グループに関連した複数のストレージ・パスからなる集合です(詳しくは、『CREATE TABLESPACE ステートメント』を参照してください)。自動ストレージはデフォルトで有効になっていますが、データベースの作成時にそのデータベースの自動ストレージを明示的に無効にすることができます。データベースの作成時に自動ストレージを無効にするには、**AUTOMATIC STORAGE NO** パラメータを指定します。

特定のデータベース・パーティション用の自動ストレージ・パスのフリー・スペースを計算する際、データベース・マネージャは、ストレージ・パス内に以下のディレクトリまたはマウント・ポイントが存在するかどうかを検査し、最初に見つかったものを使用します。そのとき、ストレージ・パスの下の地点にファイル・システムをマウントできるので、データベース・マネージャは、表スペース・コンテナに使用できるフリー・スペースの実際の量が、ストレージ・パス・ディレクトリ自体に関連付けられている量と同じになるとは限らないことを認識します。

1. *storage\_path/instance\_name/NODE####/database\_name*
2. *storage\_path/instance\_name/NODE####*
3. *storage\_path/instance\_name*
4. *storage\_path/*

詳細は次のとおりです。

- *storage\_path* は、データベースに関連付けられたストレージ・パスです。
- *instance\_name* は、データベースが置かれているインスタンスです。
- *NODE####* は、データベース・パーティション番号 (例えば、NODE0000 または NODE0001) に対応します。
- *database\_name* は、データベースの名前です。

1 つの物理マシン上に 2 つの論理データベース・パーティションが存在し、1 つのストレージ・パス */db2data* を使ってデータベースを作成しようとしている例を考えてみます。どちらのデータベース・パーティションもこのストレージ・パスを使用しますが、ユーザーは、各パーティションごとにデータをそれぞれ独自のファイル・システムに隔離する予定であるとします。この場合、各パーティションごとに別々のファイル・システムを作成し、*/db2data/instance/NODE####* にマウントすることができます。データベース・マネージャは、ストレージ・パス上にコンテナを作成し、フリー・スペースを判別するときに、*/db2data* のフリー・スペース情報を取り出すのではなく、それに対応する */db2data/instance/NODE####* ディレクトリのもを取り出すことを承知しています。

一般的に、複数パーティション・データベースでは、どのパーティションにも同じストレージ・パスを使用する必要があり、それらはすべて、**CREATE DATABASE** コマンドの実行前に存在している必要があります。その例外の 1 つとして、ストレージ・パス内でデータベース・パーティション式を使用する場合があります。その使用によって、処理結果のパス名が各パーティションごとに異なるように、データベース・パーティション番号をストレージ・パスにおいて反映することができます。

パーティション・データベース環境では、データベース・マネージャーは、すべてのデータベース・パーティションで、指定されたパスまたはデフォルトのパスの下にサブディレクトリー \$DB2INSTANCE/NODExxxx を作成します。xxxx は db2nodes.cfg ファイルで定義されたデータベース・パーティション番号です (つまり、データベース・パーティション 0 が NODE0000 になる)。サブディレクトリー SQL00001 から SQLnnnnn は、このパスに常駐します。これにより、異なるデータベース・パーティションに関連したデータベース・オブジェクトが異なるディレクトリーに (指定したパスまたはデフォルト・パスの下のサブディレクトリー \$DB2INSTANCE が、すべてのデータベース・パーティションで共通だとしても) 保管されることが保証されます。

LDAP (Lightweight Directory Access Protocol) サポートが現行のマシン上で有効である場合、データベースは自動的に LDAP ディレクトリーに登録されます。同じ名前のデータベース・オブジェクトが既に LDAP ディレクトリーに存在している場合でも、データベースはローカル・マシンに作成されますが、名前の競合があることを示す警告メッセージが戻されます。この場合、ユーザーは **CATALOG LDAP DATABASE** コマンドを使用して、LDAP データベース項目を手動でカタログすることができません。

**CREATE DATABASE** は、アプリケーションが既にデータベースに接続されている場合、失敗します。

データベースが作成される際に、詳細デッドロック・イベント・モニターが作成されます。他のモニターと同様に、このイベント・モニターに関連する追加の処理使用量がいくつかあります。デッドロック・イベント・モニターは、**DROP EVENT MONITOR** コマンドを実行することによってドロップできます。

**CATALOG DATABASE** を使用して、新しいデータベースに異なる別名を定義してください。

指定されるコード・セットおよびテリトリ値の組み合わせは有効なものでなければなりません。サポートされる組み合わせのリストについては、『サポートされているテリトリ・コードおよびコード・ページ』を参照してください。

Windows オペレーティング・システムでドライブではなくデータベース・パスを指定するには、DB2 レジストリー変数を **DB2\_CREATE\_DB\_ON\_PATH=YES** と設定する必要があります。

UCA400\_NO、UCA400\_LSK、または UCA400\_LTH の代わりに language-aware-collation または locale-sensitive-collation を指定した **COLLATE USING** 節を使用してください。

**重要:** Unicode Standard バージョン 4.0.0 の Unicode Collation Algorithm に基づく照合は、バージョン 10.1 で非推奨になっており、将来のリリースで除去される可能性があります。詳しくは、「DB2 バージョン 10.1 の新機能」の『Unicode Standard バージョン 4.0.0 の Unicode Collation Algorithm に基づく照合が推奨されなくなった』を参照してください。

## CREATE TOOLS CATALOG

新規または既存のデータベースで DB2 ツール・カタログ表を作成します。ツール・カタログには、使用可能な管理タスクについての情報が格納されます。

データベースはローカルでなければなりません。

このコマンドは、オプションですべてのアプリケーションを強制クローズし、新しい表スペースがツール・カタログに作成されると、データベース・マネージャーを停止して再開します。また、DB2 Administration Server (DAS) 構成を更新し、スケジューラーをアクティブ化します。

このコマンドは、IBM Data Server Client では無効です。

### 有効範囲

このコマンドを発行したノードは、新規データベースのカタログ・ノードになります。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL

また、ユーザーには DB2 Administration Server 構成パラメーターを更新するための、DASADM 権限も必要です。

### 必要な接続

このコマンドの処理中、データベース接続が一時的に確立されます。このコマンドは、新しい表スペースが作成されると、オプションでデータベース・マネージャーを停止して再開します。

### コマンド構文

```

▶▶ CREATE TOOLS CATALOG catalog-name
▶ CREATE NEW DATABASE database-name
  USE EXISTING TABLESPACE tablespace-name IN DATABASE database-name
▶ [FORCE] [KEEP INACTIVE]

```

### コマンド・パラメーター

#### CATALOG *catalog-name*

DB2 ツール・カタログを固有に識別するのに使用される名前。カタログ表はこのスキーマ名で作成されます。

#### NEW DATABASE *database-name*

新しいデータベースに割り当てられる名前。ローカル・データベース・ディレクトリーまたはシステム・データベース・ディレクトリーの他のデータベ

ースから、そのデータベースを区別する、固有名でなければなりません。名前は、データベースの命名規則に適合していることが必要です。

#### EXISTING DATABASE *database-name*

ツール・カタログのホストになる既存のデータベースの名前。ローカル・データベースでなければなりません。

#### EXISTING TABLESPACE *tablespace-name*

DB2 ツール・カタログ表を作成するのに使用される、既存の 32K ページの表スペースを指定するのに使う名前。表を正常に作成するためには、32K ページ・サイズの TEMPORARY 表スペースも必要です。

#### FORCE

新しい表スペースでツール・カタログを作成する場合は、データベース・マネージャーを再開することが必要です。この場合、アプリケーションが接続してはなりません。 **FORCE** オプションを使って、データベースに接続しているアプリケーションを確実になくします。アプリケーションが接続していると、既存の表スペースを指定しない限り、ツール・カタログの作成は失敗します。

#### KEEP INACTIVE

このオプションは、DB2 Administration Server 構成パラメーターを更新したり、スケジューラーを有効にしたりすることはありません。

#### 例

```
db2 create tools catalog cc create new database toolsdb
```

```
db2 create tools catalog catalog1 use existing database toolsdb force
```

```
db2 create tools catalog catalog1 use existing tablespace user32Ksp
in database toolsdb
```

```
db2 create tools catalog toolscat use existing database toolsdb keep inactive
```

#### 使用上の注意

- ツール・カタログ表には 32K ページの表スペースが 2 つ (正規の表スペースと、TEMPORARY 表スペース) 必要です。さらに、既存の表スペースを指定しない限り、その表スペースには新しい 32 K のバッファ・プールが作成されます。これにはデータベース・マネージャーの再開が必要です。データベース・マネージャーを再開する必要がある場合、すべての既存のアプリケーションは強制クローズしなければなりません。新しい表スペースは、デフォルトのデータベース・ディレクトリー・パスのそれぞれで、1 つのコンテナを伴って作成されます。
- このコマンドの実行前に、この名前のアクティブ・カタログが存在する場合、そのカタログは非アクティブ化され、新しいカタログがアクティブ・カタログになります。
- 同じデータベースに複数の DB2 ツール・カタログが作成されることがあり、それらはカタログ名によって固有に識別されます。
- **jdk\_path** 構成パラメーターは、DB2 Administration Server (DAS) 構成で、最小サポート・レベルの Java 用 SDK に設定する必要があります。
- DAS 構成パラメーターを更新するには、DB2 Administration Server の DASADM 権限が必要です。

## CREATE TOOLS CATALOG

- **KEEP INACTIVE** オプションを指定しない限り、このコマンドは DB2 ツールのカタログ・データベース構成に関連するローカル DAS 構成パラメーターを更新し、スケジューラーをローカル DAS サーバーで有効にします。
- 32 ビットおよび 64 ビット・インスタンスの両方をサポートするいずれかのプラットフォーム (AIX、HP-UX、Solaris) 上の 64 ビット・インスタンスに対してツール・カタログを作成する場合には、**jdk\_64\_path** 構成パラメーターを設定することが必要です。
- パーティション・データベース環境では、カタログ・パーティションに 32 KB REGULAR 表スペースが存在しなければなりません。そうでない場合は、表スペースが指定されたコマンド (例えば次に示すコマンド) は失敗します。

```
db2 create tools catalog1 use existing tablespace user32Ksp
in database toolsdb
```

## DEACTIVATE DATABASE

指定されたデータベースを非アクティブ化し、必要なデータベース・サービスをすべて停止します。

### 有効範囲

このコマンドは、インスタンス内のすべてのメンバー、または指定されたメンバーで、ターゲット・データベースを非アクティブ化します。非アクティブ化中に 1 つ以上のメンバーがエラーを検出すると、警告が戻されます。それらのメンバーでは、データベースはアクティブのままになります。

### 許可

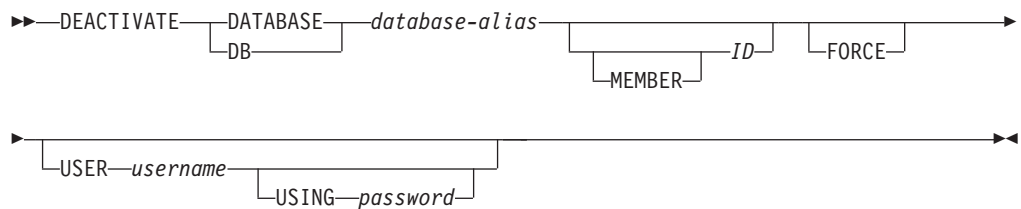
以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMAINT

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

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

停止するデータベースの別名を指定します。

#### MEMBER

データベースを非アクティブ化する対象となる 1 つ以上のメンバーを指定します。

**ID** データベースを非アクティブ化する対象となる 1 つ以上のメンバーの数値 ID を指定します。 **MEMBER** パラメーター・キーワードを指定せずに数値 ID を使用することができます。

#### FORCE

未確定トランザクションが存在しても、データベースを強制的に非アクティブ化します。

**注:** **FORCE** オプションを使用せずに、未確定トランザクションが存在するデータベースを非アクティブ化しようとする、エラーが戻されます。



## DEACTIVATE DATABASE

**USER** *username*

データベースを停止するユーザーを指定します。

**USING** *password*

ユーザー名のパスワードを指定します。

### 使用上の注意

- **DEACTIVATE DATABASE** コマンドを実行するアプリケーションは、どのデータベースへもアクティブ・データベース接続を持つことができません。
- **ACTIVATE DATABASE** コマンドによって初期化されたデータベースは、**DEACTIVATE DATABASE** または **db2stop** コマンドを発行することによってシャットダウンできます。データベースが **ACTIVATE DATABASE** コマンドによって初期化された場合、そのデータベースから最後のアプリケーションが切断されてもデータベースはシャットダウンされないため、**DEACTIVATE DATABASE** コマンドを使用する必要があります。
- DB2 pureScale 環境では、あるメンバーのアクティブ・データベースは、**DEACTIVATE DATABASE** または **db2stop** コマンドを発行することによってのみ、そのメンバーで非アクティブ化することができます。

### 互換性

以前のバージョンとの互換性:

- **DB2\_ENFORCE\_MEMBER\_SYNTAX** レジストリー変数が ON に設定されている場合を除き、**DBPARTITIONNUM** を **MEMBER** の代わりに使用できます。



## DECOMPOSE XML DOCUMENT

ストアド・プロシージャを呼び出し、登録済みの分解可能 XML スキーマを使用して、単一の XML 文書を分解します。

### 許可

以下のグループの特権または権限のいずれかが必要です。

- 以下のいずれかの権限が必要です。
  - アノテーション付きスキーマ文書の集合で参照されるすべてのターゲット表における CONTROL 特権
  - DATAACCESS 権限
- 以下の特権すべて:
  - ターゲット表に対する INSERT 特権 (アクション・ファイルで指定される操作に必要)
  - db2-xdb:expression または db2-xdb:condition アノテーションによって参照されるすべての表における SELECT、INSERT、UPDATE、または DELETE 特権 (必要に応じて)

**VALIDATE** オプションを指定すると、XML スキーマにおける USAGE 特権も必要になります。

### 必要な接続

データベース

### コマンド構文

```

▶▶—DECOMPOSE XML DOCUMENT—xml-document-name—XMLSCHEMA—xml-schema-name————▶▶
▶└──VALIDATE──┘────────────────────────────────────────────────────────────────────────────────▶▶

```

### コマンド・パラメーター

**DECOMPOSE XML DOCUMENT** *xml-document-name*

*xml-document-name* は、分解される入力 XML 文書のファイル・パスおよびファイル名です。

**XMLSCHEMA** *xml-schema-name*

*xml-schema-name* は、文書の分解に使用される、XML スキーマ・リポジトリに登録された既存の XML スキーマの名前です。 *xml-schema-name* は修飾 SQL ID で、オプションの SQL スキーマ名の後にピリオドと XML スキーマ名が続く形で構成されます。 SQL スキーマ名が指定されない場合、DB2 特殊レジスター CURRENT SCHEMA の値であると想定されます。

**VALIDATE**

このパラメーターは、入力 XML 文書が最初に妥当性検査され、その文書が有効な場合に限り、分解されることを示します。 **VALIDATE** が指定されない場合、入力 XML 文書は分解前に妥当性検査されません。

## DECOMPOSE XML DOCUMENT

### 例

以下の例は、XML 文書 ./gb/document1.xml が、登録済み XML スキーマ DB2INST1.GENBANKSCHEMA を使用して妥当性検査され、分解されることを指定します。

```
DECOMPOSE XML DOCUMENT ./gb/document1.xml
XMLSCHEMA DB2INST1.GENBANKSCHEMA
VALIDATE
```

以下の例は、XML 文書 ./gb/document2.xml が、妥当性検査されずに、登録済み XML スキーマ DB2INST2."GENBANK SCHEMA1" を使用して分解されることを指定します。このとき、DB2 特殊レジスター CURRENT SCHEMA の値が DB2INST2 に設定されていることを想定しています。

```
DECOMPOSE XML DOCUMENT ./gb/document2.xml
XMLSCHEMA "GENBANK SCHEMA1"
```

## DECOMPOSE XML DOCUMENTS

データベース列に格納されている XML 文書を分解します。XML 文書からのデータは、XML スキーマで指定されたアノテーションに基づいてリレーショナル表の列に格納されます。

DECOMPOSE XML DOCUMENTS コマンドは

XDB\_DECOMP\_XML\_FROM\_QUERY ストアド・プロシージャを呼び出し、登録済みの分解可能 XML スキーマを使用してバイナリーまたは XML 列の 1 つ以上の XML 文書を分解します。

### 許可

以下のグループの特権のいずれかが必要です。

- 以下の特権すべて:
  - アノテーション付きスキーマで参照されるすべてのターゲット表における INSERT 特権
  - 入力文書を保持する列が含まれる表、別名、またはビューにおける SELECT 特権
  - db2-xdb:expression または db2-xdb:condition アノテーションによって参照されるすべての表における SELECT、INSERT、UPDATE、または DELETE 特権 (必要に応じて)
- 以下のいずれかの権限が必要です。
  - アノテーション付きスキーマ文書の集合で参照されるすべての表、および入力文書を保持する列が含まれる表、別名、またはビューにおける CONTROL 特権
  - DATAACCESS 権限

**VALIDATE** オプションを指定すると、XML スキーマにおける USAGE 特権も必要になります。

### 必要な接続

データベース

### コマンド構文

```

▶▶—DECOMPOSE XML DOCUMENTS IN—select-statement—XMLSCHEMA—xml-schema-name—▶▶
┌—VALIDATE—┐ ┌—ALLOW NO ACCESS—┐ ┌—ALLOW ACCESS—┐ ┌—COMMITCOUNT—integer—┐
└──────────┘ └──────────────────┘ └────────────────┘ └────────────────────────┘
┌—CONTINUE_ON_ERROR—┐ ┌—MESSAGES—message-file—┐
└──────────────────┘ └──────────────────────────┘

```

### コマンド・パラメーター

#### DECOMPOSE XML DOCUMENTS IN *select-statement*

*select-statement* は SQL SELECT ステートメントの規則に準拠しており、2 つの列が含まれる結果セットを戻す必要があります。最初の列は、文書 ID です。各文書 ID は、分解する XML 文書を一意的に識別します。この列は文字タイプであるか、文字タイプにキャスト可能でなければなりません。2 番目の列には、分解する XML 文書が含まれます。文書列としてサポートされるタイプは、XML、BLOB、VARCHAR FOR BIT DATA、および LONG VARCHAR FOR BIT DATA です。XML 文書を含んだ列は、基礎となる基本表の列に解決されなければなりません。この列を生成された列にすることはできません。

例えば、以下の SELECT ステートメントの DOCID 列には SALESDOC 列に格納されている XML 文書のユニーク ID が含まれています。

```
SELECT DOCID, SALESDOC FROM SALESTAB
```

#### XMLSCHEMA *xml-schema-name*

*xml-schema-name* は、文書の分解に使用される、XML スキーマ・リポジトリに登録された既存の XML スキーマの名前です。*xml-schema-name* は修飾 SQL ID で、オプションの SQL スキーマ名の後にピリオドと XML スキーマ名が続く形で構成されます。SQL スキーマ名が指定されない場合、DB2 特殊レジスター CURRENT SCHEMA の値であると想定されます。

#### VALIDATE

*xml-schema-name* に対して各入力 XML 文書が妥当性検査され、その文書が無効な場合には分解することを指定します。**VALIDATE** を指定しないと、分解前に入力 XML 文書は妥当性検査されません。

**VALIDATE** を指定しない場合、このコマンドを呼び出す前にユーザーの責任で文書を妥当性検査する必要があります。例えば、XML 文書を列に挿入する際に XMLVALIDATE を使用するか、文書の挿入前に XML プロセッサを使用します。入力 XML 文書が無効な場合に **VALIDATE** を指定しないと、分解結果は未定義になります。XML 妥当性検査については、このトピックの末尾の関連参照を参照してください。

#### ALLOW

XML スキーマ *xml-schema-name* で指定されたターゲット表へのアクセスが分解中に許可されるかどうかを指定します。ALLOW NO ACCESS はデフォルト値です。

#### ALLOW ACCESS

ALLOW ACCESS を指定すると、ターゲット表でロックを取得する場合、DECOMPOSE 操作は待機状態になり、タイムアウトになる可能性があります。

#### ALLOW NO ACCESS

ALLOW NO ACCESS を指定するか、デフォルト値として使用すると、DECOMPOSE 操作によって、XML スキーマで指定されたマッピングを持つすべての表で排他ロック (X) が取得されます。各文書の分解時に必ずしもすべてのターゲット表が関与する訳ではありませんが、長期作業単位ではデッドロックの可能性が低い状態ですべてのターゲット表がロックされます。

**COMMITCOUNT** *integer*

*integer* 個の文書の分解が正常に行われるたびに、COMMIT を実行するように指定します。値 0、またはこのオプションを指定しない場合には、DECOMPOSE 操作では COMMIT が実行されないことを意味します。

**CONTINUE\_ON\_ERROR**

文書固有のエラーが発生した場合には、DECOMPOSE 操作を次の文書で続行することを指定します。正常に分解されなかった文書によって行われたデータベースへの変更は、次の文書に移動する前に取り消されます。

CONTINUE\_ON\_ERROR を指定しないと、DECOMPOSE 操作は、分解を正常に行えない最初の文書で停止します。

DECOMPOSE 操作は致命的エラーの場合、および文書固有のエラーではない場合には、CONTINUE\_ON\_ERROR オプションが指定されている場合であっても続行しません。

**MESSAGES** *message-file*

DECOMPOSE 操作は UTF-8 エンコード XML 文書を生成します。この文書は正常に分解されなかった入力 XML 文書と、その失敗の理由をリスト表示します。分解エラーが含まれる文書が生成されるのは、正常に分解できなかった XML 文書が少なくとも 1 つある場合に限られます。サーバーのロケールに応じて、メッセージが変換されます。*message-file* は、分解情報が含まれる XML 文書が入ったファイルです。*message-file* が指定されている場合、CLP コマンドが呼び出されたシステムにそのファイルが作成されます。ファイルへの完全パスを指定しなかった場合は、現行ディレクトリーに作成されます。

このオプションを指定しない場合、分解情報は標準出力に書き込まれます。

XML 文書の分解に関する情報は XML 文書として表示され、オプションとしてパラメーター **MESSAGES** によって指定した *message-file* に送ることもできます。*message-file* 内の XML 文書の形式は、次のとおりです。

```
<?xml version='1.0' xmlns:xdb="http://www.ibm.com/xmlns/prod/db2/xdb1"?>
<xdb:errorReport>
  <xdb:document>
    <xdb:documentId>sssss</xdb:documentId>
    <xdb:errorMsg>qqqq</xdb:errorMsg>
  </xdb:document>
  <xdb:document>
    .
    .
    .
  </xdb:document>
  .
  .
  .
</xdb:errorReport>
```

documentId 値 *sssss* は、*select-statement* によって指定された最初の列からの値です。この値は、正常に分解されなかった XML 文書を識別します。errorMsg 値 *qqqq* は、文書を分解しようとした際に生じたエラーです。

**例**

分解する XML 文書を関連表 (例: ABC.SALESTAB) に挿入できます。すべての文書は ABC.SALES として登録された XML スキーマに対応しており、このスキーマは分解情報によってアノテーションが付けられ、分解で使用できます。分解が挿入される列名が SALESDOC で、対応する ID が DOCID に挿入されるとすると、DECOMPOSE XML DOCUMENTS コマンドを以下のように呼び出します。

## DECOMPOSE XML DOCUMENTS

```
DECOMPOSE XML DOCUMENTS IN 'SELECT DOCID, SALESDOC FROM SALESTAB'  
XMLSCHEMA ABC.SALES  
MESSAGES /home/myid/errors/errorreport.xml
```

## DEREGISTER

DB2 サーバーの登録をネットワーク・ディレクトリー・サーバーから取り消します。

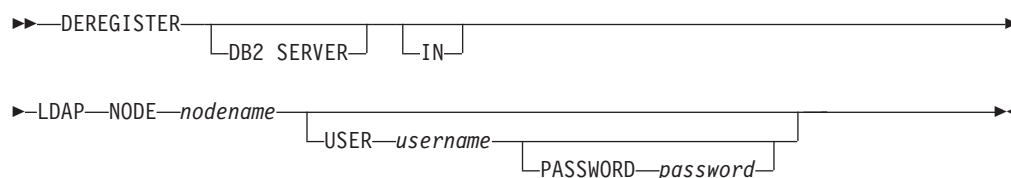
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

**IN** DB2 サーバーを登録解除するネットワーク・ディレクトリー・サーバーを指定します。有効な値は、LDAP (Lightweight Directory Access Protocol) ディレクトリー・サーバーの場合、LDAP です。

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

ユーザーの LDAP 識別名 (DN) を指定します。LDAP ユーザー DN には、LDAP ディレクトリーからオブジェクトを削除するための十分な権限が必要です。ユーザー名は、LDAP での登録解除時には任意指定です。ユーザーの LDAP DN が指定されない場合、現行ログオン・ユーザーの資格情報が使用されます。

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

アカウント・パスワードを示します。

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

ノード名は、DB2 サーバーが LDAP で登録されるときに指定される値です。

### 使用上の注意

このコマンドは、LDAP 環境にあるリモート・マシンにしか発行できません。リモート・マシンに発行される場合、リモート・サーバーのノード名を指定する必要があります。

DB2 サーバーは、インスタンスがドロップされるときに自動的に登録解除されません。

## DESCRIBE

**DESCRIBE** コマンドは、表またはビューの列、索引、およびデータ・パーティションに関するメタデータを表示します。このコマンドは、SELECT、CALL、または XQuery ステートメントの出力に関するメタデータも表示できます。

**DESCRIBE** コマンドを使用して、以下の項目のいずれかに関する情報を表示します。

- SELECT、CALL、または XQuery ステートメントの出力
- 表またはビューの列
- 表またはビューの索引
- 表またはビューのデータ・パーティション

### 許可

必要な許可は、**DESCRIBE** コマンドを使用して表示する情報のタイプによって異なります。

- SYSTOOLSTMPSPACE 表スペースが存在する場合、以下の表に示されている権限の 1 つが必要になります。

情報を表示するオブジェクト	必要な特権または権限
SELECT ステートメントまたは XQuery ステートメントの出力	SELECT ステートメント内で参照されている表またはビューごとに、以下のいずれかの特権または権限: <ul style="list-style-type: none"> <li>• SELECT 特権</li> <li>• DATAACCESS 権限</li> <li>• DBADM 権限</li> <li>• SQLADM 権限</li> <li>• EXPLAIN 権限</li> </ul>
CALL ステートメントの出力	以下の特権または権限のいずれか。 <ul style="list-style-type: none"> <li>• DATAACCESS 権限</li> <li>• ストアード・プロシージャでの EXECUTE 特権</li> </ul>



情報を表示するオブジェクト	必要な特権または権限
表またはビューの列	<p>SYSCAT.COLUMNS システム・カタログ表に対して、以下のいずれかの特権または権限:</p> <ul style="list-style-type: none"> <li>• SELECT 特権</li> <li>• ACCESSCTRL 権限</li> <li>• DATAACCESS 権限</li> <li>• DBADM 権限</li> <li>• SECADM 権限</li> <li>• SQLADM 権限</li> </ul> <p><b>SHOW DETAIL</b> パラメーターを使用する場合、SYSCAT.DATAPARTITIONEXPRESSION システム・カタログ表に対しても、これらの特権または権限のいずれかが必要になります。</p> <p>PUBLIC には宣言済み一時表に対するすべての特権が付与されているので、このコマンドを使用して、接続内に存在するすべての宣言済み一時表に関する情報を表示できます。</p>
表またはビューの索引	<p>SYSCAT.INDEXES システム・カタログ表に対して、以下のいずれかの特権または権限:</p> <ul style="list-style-type: none"> <li>• SELECT 特権</li> <li>• ACCESSCTRL 権限</li> <li>• DATAACCESS 権限</li> <li>• DBADM 権限</li> <li>• SECADM 権限</li> <li>• SQLADM 権限</li> </ul> <p><b>SHOW DETAIL</b> パラメーターを使用する場合、GET_INDEX_COLNAMES() UDF に対する EXECUTE 特権も必要です。</p> <p>PUBLIC には宣言済み一時表に対するすべての特権が付与されているので、このコマンドを使用して、接続内に存在するすべての宣言済み一時表に関する情報を表示できます。</p>

## DESCRIBE

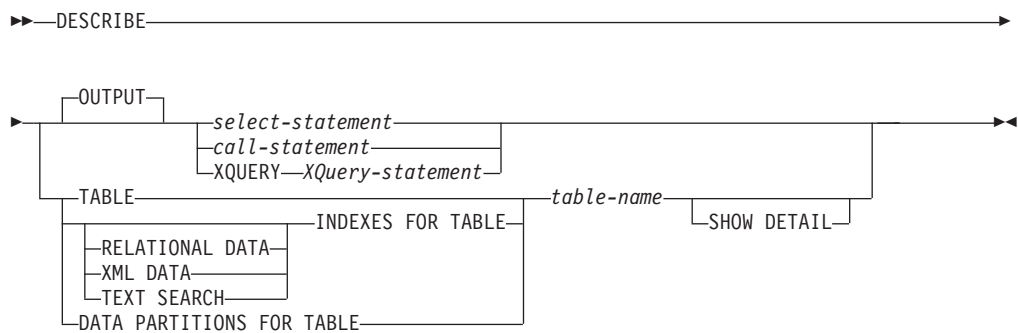
情報を表示するオブジェクト	必要な特権または権限
表またはビューのデータ・パーティション	<p>SYSCAT.DATAPARTITIONS システム・カタログ表に対して、以下のいずれかの特権または権限:</p> <ul style="list-style-type: none"> <li>• SELECT 特権</li> <li>• ACCESSCTRL 権限</li> <li>• DATAACCESS 権限</li> <li>• DBADM 権限</li> <li>• SECADM 権限</li> <li>• SQLADM 権限</li> </ul> <p>PUBLIC には宣言済み一時表に対するすべての特権が付与されているので、このコマンドを使用して、接続内に存在するすべての宣言済み一時表に関する情報を表示できます。</p>

- SYSTOOLSTMPSPACE 表スペースが存在しない場合は、上にリストされた権限のいずれかに加えて SYSADM または SYSCTRL 権限も必要になります。

### 必要な接続

データベース。暗黙接続が可能な場合には、デフォルト・データベースへの接続が確立されます。

### コマンド構文



### コマンド・パラメーター

**OUTPUT** ステートメントの出力を記述するよう指示します。このキーワードはオプションです。

*select-statement* | *call-statement* | **XQUERY** *XQuery-statement*

情報が必要なステートメントを識別します。ステートメントは CLP によって自動的に準備されます。XQuery ステートメントであることを示すには、ステートメントの先頭にキーワード **XQUERY** を入れます。DESCRIBE OUTPUT ステートメントで暗黙的な隠し列についての情報が戻されるのは、記述対象の照会の最終的な結果表の SELECT リスト内にその列が明示的に指定されている場合のみです。

**TABLE** *table-name*

記述する表またはビューを指定します。 *schema.table-name* 形式の完全修飾名を使用しなければなりません。実際の表の代わりに表の別名を使用することはできません。暗黙的に非表示になっている列に関する情報が戻されますが、どの列が暗黙的に非表示にされているかを示すには **SHOW DETAIL** を使用する必要があります。

**DESCRIBE TABLE** コマンドは、各列に関する以下の情報をリストします。

- 列名
- タイプ・スキーマ
- タイプ名
- 長さ
- スケール
- NULL 値 (yes/no)

**INDEXES FOR TABLE** *table-name*

索引を記述する必要がある表またはビューを指定します。

*schema.table-name* という形式の完全修飾名を使用するか、*table-name* のみを指定してデフォルトのスキーマが自動的に使用されるようにできます。実際の表の代わりに表の別名を使用することはできません。

**DESCRIBE INDEXES FOR TABLE** コマンドは、その表またはビューの各索引に関する以下の情報をリストします。

- 索引スキーマ
- 索引名
- ユニーク規則
- 列の数
- 索引タイプ

**DESCRIBE INDEXES FOR TABLE** コマンドが **SHOW DETAIL** オプションを指定して使用される場合、索引名が 18 バイトより長いと、索引名は切り捨てられます。索引タイプ・オプションが指定されていない場合、すべての索引タイプ (リレーショナル・データ索引、XML データに対する索引、および Text Search 索引) の情報がリストされます。出力には以下の追加情報が含まれます。

- リレーショナル・データ索引の索引 ID、XML パス索引、XML 領域索引、または XML データに対する索引
- XML データに対する索引のデータ・タイプ
- XML データに対する索引のハッシュ
- XML データに対する索引の最大 VARCHAR 長
- XML データに対する索引に指定された XML パターン
- テキスト検索索引のコード・ページ
- テキスト検索索引の言語
- テキスト検索索引に指定された形式
- テキスト検索索引の更新最小数
- テキスト検索索引の更新頻度

## DESCRIBE

- テキスト検索索引のコレクション・ディレクトリー
- 列名
- BUSINESS\_TIME WITHOUT OVERLAPS 節の指定の有無

特定の索引タイプについてだけの情報をリストする索引タイプを指定します。複数の索引タイプの指定はサポートされていません。

### RELATIONAL DATA

RELATIONAL DATA 索引タイプ・オプションが **SHOW DETAIL** オプションなしで指定されている場合は、以下の情報だけがリストされます。

- 索引スキーマ
- 索引名
- ユニーク規則
- 列の数

**SHOW DETAIL** を指定した場合、列名情報もリストされます。

### XML DATA

XML DATA 索引タイプ・オプションが **SHOW DETAIL** オプションなしで指定されている場合は、以下の情報だけがリストされます。

- 索引スキーマ
- 索引名
- ユニーク規則
- 列の数
- 索引タイプ

**SHOW DETAIL** を指定した場合、以下の XML データに対する索引の情報もリストされます。

- 索引 ID
- データ・タイプ
- ハッシュ
- 最大 varchar 長
- XML パターン
- 列名

### TEXT SEARCH

TEXT SEARCH 索引タイプ・オプションが **SHOW DETAIL** オプションなしで指定されている場合は、以下の情報だけがリストされます。

- 索引スキーマ
- 索引名

**SHOW DETAIL** を指定した場合、以下のテキスト検索索引情報もリストされます。

- 列名
- コード・ページ
- 言語
- フォーマット

- 更新最小数
- 更新頻度
- コレクション・ディレクトリー

**TEXT SEARCH** オプションが指定され、テキスト検索オプションがインストールされていないか正しく構成されていない場合、エラー (SQLSTATE 42724) が戻されます。

列にリストされる情報については、DB2 Text Search の情報を参照してください。

#### DATA PARTITIONS FOR TABLE *table-name*

データ・パーティションを記述する必要がある表またはビューを指定します。表に含まれるデータ・パーティションごとに表示される情報には、パーティション ID とパーティション・インターバルが含まれます。結果は、パーティション ID の順になっています。 *schema.table-name* 形式の完全修飾名を使用しなければなりません。実際の表の代わりに表の別名を使用することはできません。 *schema* はユーザー名で、その下に表またはビューが作成されます。

**DESCRIBE DATA PARTITIONS FOR TABLE** コマンドの場合、出力に以下の追加情報を内容とする 2 番目の表を含めることを指定します。

- データ・パーティション順序 ID
- SQL でのデータ・パーティション式

#### SHOW DETAIL

**DESCRIBE TABLE** コマンドの場合、

- CHARACTER、VARCHAR または LONG VARCHAR 列のいずれかが FOR BIT DATA として定義されたかどうか
- 列番号
- 分散キー・シーケンス
- コード・ページ
- 非表示属性
- デフォルト
- 表パーティションのタイプ (範囲によってパーティション化されている表の場合、元の出力の後にこの出力が表示される)
- パーティション・キー列 (範囲によってパーティション化されている表の場合、元の出力の後にこの出力が表示される)
- 索引に使用される表スペースの ID
- 表に定義された期間 (テンポラル表の場合、元の出力の後にこの出力が表示される)
- 表でバージョン管理が有効かどうか (テンポラル表の場合、元の出力の後にこの出力が表示される)

#### 例

##### SELECT ステートメントの出力の記述

次に示すのは、SELECT ステートメントを記述する方法の一例です。

## DESCRIBE

```
db2 describe output select * from staff
Column Information
```

Number of columns: 7

Data Type	Length	Column Name	Name Length
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

### CALL ステートメントの出力の記述

次のステートメントでストアド・プロシージャが作成されたとします。

```
CREATE PROCEDURE GIVE_BONUS (IN EMPNO INTEGER,
                             IN DEPTNO INTEGER,
                             OUT CHEQUE INTEGER,
                             INOUT BONUS DEC(6,0))
...
```

次の例は、CALL ステートメントの出力を記述する方法を示しています。

```
db2 describe output call give_bonus(123456, 987, ?, 15000.)
Column Information
Number of Columns: 2
```

Data Type	Length	Column Name	Name Length
497 INTEGER	4	CHEQUE	6
485 DECIMAL	6, 0	BONUS	5

プロシージャで配列タイプに 1 つ以上のパラメーターがある場合、**DESCRIBE** コマンドの出力には追加列が 1 つあります。これは、配列パラメーターの最大カーディナリティを示します。空の値は、パラメーターが配列でないことを示します。

ステートメントで作成された配列タイプおよびプロシージャを指定すると、次のようになります。

```
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)
...
```

次の例は、配列パラメーターを指定した CALL ステートメントの出力を記述する方法を示しています。前の例とのフォーマットの違いは、Max cardinality 列だけです。

```
db2 describe output call discontinue_prod(ARRAY[12, 34, 26], '04/13/2006',?)
Column Information
Number of Columns: 2
```

SQL type	Type length	Column name	Name length	Max cardinality
497 INTEGER	4	NUM_PENDING_ORDERS	17	
497 INTEGER	10	CUSTOMER_LIST	13	1000

## XQuery ステートメントの出力の記述

XML データ・タイプの列 INFO を含む表 CUSTOMER に対する XQuery ステートメントの記述例を以下に示します。

```
db2 describe xquery for $cust in db2-fn:xmlcolumn("CUSTOMER.INFO") return $cust
```

Column Information

Number of Columns: 1

SQL type	Type length	Column name	Name length
998 XML	0	1	1

キーワード **XQUERY** が指定されていない場合、SQL0104N が戻されます。

```
db2 describe for $cust in db2-fn:xmlcolumn("CUSTOMER.INFO") return $cust
SQL0104N An unexpected token "for" was found following "DESCRIBE". Expected
tokens may include: "OUTPUT". SQLSTATE=42601
```

**XQUERY** オプションがサポートされていない旧バージョンのサーバーに対して **DESCRIBE XQUERY** コマンドが発行された場合、旧バージョンのサーバーによってその機能がサポートされていないことを示すメッセージ DB21108E が戻されます。

## 表の記述

次に示すのは、表を記述する方法の一例です。

```
db2 describe table user1.department
```

Table: USER1.DEPARTMENT

Column name	Data Type schema	Data Type name	Column length	Scale	Nulls
AREA	SYSIBM	SMALLINT	2		0 No
DEPT	SYSIBM	CHARACTER	3		0 No
DEPTNAME	SYSIBM	CHARACTER	20		0 Yes

次に示すのは、詳細を含めて表索引を記述する方法の一例です。この例の場合のように表がパーティション化されている場合、付加的な詳細が既存の出力の後に表示されます。非パーティション表の場合、付加的な表ヘッダーは表示されません。

```
db2 describe table user1.employee show detail
```

Column name	Data Type schema	Column number	Data Type name	Column length	Hidden
FIRST	SYSIBM	0	CHARACTER	10	No
LAST	SYSIBM	1	CHARACTER	10	No
EMPLOYEEENUM	SYSTEM	2	CHARACTER	10	Implicitly

Table is partitioned by range (ordered on the following column/s):

```
-----
LAST
FIRST
EMPLOYEEENUM
```

## 表索引の記述

次に示すのは、表索引を記述する方法の一例です。このコマンドは、2 つのリレーショナル・データ索引、6 つの XML データ索引、2 つのテキスト検索索引、およびシステム索引をリストします。

## DESCRIBE

db2 describe indexes for table user1.department

Index schema	Index name	Unique rule	Number of columns	Index type
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

以下のコマンドは、表 USER1.DEPARTMENT のリレーショナル・データ索引をリストします。

db2 describe relational data indexes for table user1.department

Index schema	Index name	Unique rule	Number of columns
SYSIBM	SQL070531145253650	P	1
USER1	RELIDX1	D	1
USER1	RELIDX2	D	2

以下のコマンドは、表 USER1.DEPARTMENT の XML データに対する索引をリストします。

db2 describe xml data indexes for table user1.department

Index schema	Index name	Unique rule	Number of columns	Index type
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

以下のコマンドは、表 USER1.DEPARTMENT のテキスト検索索引情報をリストします。

db2 describe text search indexes for table user1.department

Index schema	Index name
USER1	TXTIDX1
USER1	TXTIDX2



以下のコマンドは、パーティション表 myDpartT のパーティションおよび非パーティション索引の両方に関する情報をリストします。

```
db2 describe indexes for table myDPartT
```

Index schema	Index name	Unique rule	Number of columns	Index Partitioning
NEWTON	IDXNDP	D	1	N
NEWTON	IDXDP	D	1	P

### データ・パーティションの記述

次に示すのは、データ・パーティションを記述する方法の一例です。

```
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

詳細を含めてデータ・パーティションを記述した場合、前の例と同じ出力に加えて、パーティション ID およびデータ・パーティションのデータの入っている表スペースと、索引の入っている表スペースの ID を示す付加的な表が戻されます。

```
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

---

### DETACH

論理 DBMS インスタンス接続を除去し、この層を使用した論理接続がほかがない場合、物理通信接続を終了します。

#### 許可

なし

#### 必要な接続

なし。既存のインスタンス接続を除去します。

#### コマンド構文

▶▶—DETACH—◀◀

#### コマンド・パラメーター

なし

---

## DROP CONTACT

ローカル・システムで定義された連絡先のリストから、連絡先を除去します。連絡先とは、スケジューラーおよびヘルス・モニターがメッセージを送信する先のユーザーです。Database Administration Server (DAS) の **contact\_host** 構成パラメータの設定により、リストがローカルかグローバルかが決まります。

### 許可

なし

### 必要な接続

なし。ローカル実行のみ: このコマンドはリモート接続では使用できません。

### コマンド構文

```
▶▶—DROP CONTACT—name—————▶▶
```

### コマンド・パラメーター

**CONTACT** *name*

ローカル・システムからドロップされる連絡先の名前。

### DROP CONTACTGROUP

ローカル・システムで定義された連絡先のリストから、連絡先グループを除去します。連絡先グループには、スケジューラーおよびヘルス・モニターがメッセージを送信する先のユーザーのリストが入っています。 Database Administration Server (DAS) の **contact\_host** 構成パラメーターの設定により、リストがローカルかグローバルかが決まります。

#### 許可

なし

#### 必要な接続

なし

#### コマンド構文

▶▶—DROP CONTACTGROUP—*name*—————▶▶

#### コマンド・パラメーター

**CONTACTGROUP** *name*

ローカル・システムからドロップされる連絡先グループの名前。

## DROP DATABASE

データベースの内容とそのデータベースのすべてのログ・ファイルを削除し、データベースをアンカタログし、さらにデータベースのサブディレクトリーを削除します。

### 有効範囲

デフォルトでは、このコマンドは `db2nodes.cfg` ファイル内にリストされているデータベース・パーティションすべてに影響を与えます。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL

### 必要な接続

インスタンス。明示的なアタッチは必要ありません。データベースがリモートとして示されている場合、リモート・データベース・パーティション・サーバーへのインスタンス・アタッチはコマンドの持続期間の間、ずっと確立されたままになります。

### コマンド構文

```

▶▶ DROP DATABASE database-alias
      DB
      AT DBPARTITIONNUM
  
```

### コマンド・パラメーター

#### DATABASE *database-alias*

ドロップするデータベースの別名を指定します。データベースはシステム・データベース・ディレクトリー内にカタログされている必要があります。

#### AT DBPARTITIONNUM

**DROP DATABASE** コマンドを発行したデータベース・パーティションだけで、データベースが削除されることを指定します。このパラメーターは、IBM InfoSphere® Warehouse に付属のユーティリティーがパーティション・データベース環境で使用できるものであり、汎用を目的としたものではありません。このパラメーターの使用が不適切であると、システム内に不整合が生じることもありえますので、ご使用の際には十分注意してください。

### 例

次の例は、データベース別名 `SAMPLE` で参照されるデータベースを削除します。

```
db2 drop database sample
```

### 使用上の注意

**DROP DATABASE** はすべてのユーザー・データとログ・ファイル、およびデータベースのバックアップおよびリストア履歴を削除します。リストア操作後のロールフォ

## DROP DATABASE

ワード・リカバリーにログ・ファイルが必要である場合、またはデータベースのリストアにバックアップ履歴が必要である場合、このコマンドを実行する前にそれらのファイルを保存しておく必要があります。

データベースは使用中であってはなりません。データベースをドロップする前に、すべてのユーザーをデータベースから切断しなければなりません。

ドロップするためには、データベースがシステム・データベース・ディレクトリーにカタログされている必要があります。指定されたデータベース別名だけがシステム・データベース・ディレクトリーから除去されます。同じデータベースに対して他の別名が存在する場合、その項目はそのままです。ドロップしようとするデータベースがローカル・データベース・ディレクトリーの最後の項目である場合、ローカル・データベース・ディレクトリーは自動的に削除されます。

**DROP DATABASE** がリモート・クライアント (または同一マシンの別のインスタンス) から出される場合、指定された別名はクライアントのシステム・データベース・ディレクトリーからドロップされます。それに対応するデータベース名は、サーバーのシステム・データベース・ディレクトリーから除去されます。

## DROP DBPARTITIONNUM VERIFY

任意のデータベースのデータベース・パーティション・グループ中にデータベース・パーティションが存在するかどうか、およびそのデータベース・パーティションでイベント・モニターが定義されているかどうかを検査します。このコマンドは、データベース・パーティションをパーティション・データベース環境からドロップする前に使用してください。

### 有効範囲

このコマンドは、それが発行されたデータベース・パーティションに対してだけ影響を与えます。

### 許可

SYSADM

### コマンド構文

▶—DROP DBPARTITIONNUM VERIFY—▶

### コマンド・パラメーター

なし

### 使用上の注意

データベース・パーティションが使用中ではないことを示すメッセージが戻された場合、**STOP DATABASE MANAGER** コマンドに **DROP DBPARTITIONNUM** を指定して使用し、`db2nodes.cfg` ファイルからそのデータベース・パーティションの項目をドロップしてください。これでそのデータベース・パーティションはデータベース・システムからドロップされます。

そのデータベース・パーティションが使用中であることを示すメッセージが戻された場合、以下のアクションをとる必要があります。

1. そのデータベース・パーティションにデータがある場合、**REDISTRIBUTE DATABASE PARTITION GROUP** を使用してデータを再配分し、データベース・パーティションからデータを除去します。データベースの任意のデータベース・パーティション・グループからデータベース・パーティションをドロップするには、**REDISTRIBUTE DATABASE PARTITION GROUP** コマンドか **ALTER DATABASE PARTITION GROUP** ステートメントで、**DROP DBPARTITIONNUM** オプションを使用します。このアクションを、データベース・パーティション・グループ中にデータベース・パーティションを含むデータベースごとに行う必要があります。
2. データベース・パーティションで定義されているイベント・モニターをすべてドロップします。
3. **DROP DBPARTITIONNUM VERIFY** を実行し、データベースが使用中ではなくなったことを確認してください。

## DROP TOOLS CATALOG

指定されたデータベースの指定されたカタログで、DB2 ツール・カタログ表をドロップします。

このコマンドは、IBM Data Server Client では無効です。

**警告:** アクティブ・ツール・カタログをドロップすると、タスクのスケジュールができなくなり、スケジュール済みのタスクも実行されません。スケジューラーをアクティブ化するには、前のツール・カタログをアクティブ化するか、新しいツール・カタログを作成することが必要です。

### 有効範囲

このコマンドはデータベースに影響します。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL

また、ユーザーには DB2 Administration Server (DAS) 構成パラメーターを更新するための、DASADM 権限も必要です。

### 必要な接続

このコマンドの処理中、データベース接続が一時的に確立されます。

### コマンド構文

```
►►—DROP TOOLS CATALOG—catalog-name—IN DATABASE—database-name—┐FORCE┘—►►
```

### コマンド・パラメーター

**CATALOG** *catalog-name*

DB2 ツール・カタログを固有に識別するのに使用される名前。カタログ表はこのスキーマからドロップされます。

**DATABASE** *database-name*

カタログ表を含むローカル・データベースに接続するのに使用される名前。

**FORCE**

**force** オプションは、DB2 Administration Server のスケジューラーを強制的に停止させるのに使用されます。このオプションを指定しないと、スケジューラーが停止できない場合、ツール・カタログはドロップされません。

### 例

```
db2 drop tools catalog cc in database toolsdb
```

```
db2 drop tools catalog in database toolsdb force
```



**使用上の注意**

- **jdk\_path** 構成パラメーターは、DB2 Administration Server (DAS) 構成で、最小サポート・レベルの Java 用 SDK に設定する必要があります。
- このコマンドは、ローカル DAS でスケジューラーを無効にし、DB2 ツール・カタログ・データベース構成に関連する DAS 構成パラメーターをリセットします。

---

## ECHO

ユーザーが文字ストリングを標準出力に書き込めるようにします。

### 許可

なし

### 必要な接続

なし

### コマンド構文

▶▶ ECHO character-string ▶▶

### コマンド・パラメーター

*character-string*  
任意の文字ストリング。

### 使用上の注意

標準入力として入力ファイルが使用される場合、または注釈がコマンド・シェルによる解釈を受けずに出力される場合、**ECHO** コマンドは文字ストリングを標準出力に直接出力します。

**ECHO** を出す度に 1 行が出力されます。

**ECHO** コマンドは verbose (**-v**) オプションの影響を受けません。

## EDIT

編集用に指定したコマンドによってユーザー指定のエディターを起動します。ユーザーが編集作業を終えると、エディターの内容が保存され、エディターが終了し、ユーザーが CLP 対話モードでこのコマンドを実行できるようになります。

### 有効範囲

このコマンドは、CLP 対話モードでのみ実行できます。特に CLP コマンド・モードや CLP バッチ・モードでは実行できません。

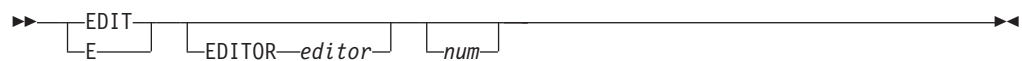
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### EDITOR

編集用に指定したエディターを起動します。このパラメーターを指定しない場合は、使用するエディターが以下の順序に基づいて決定されます。

1. **DB2\_CLP\_EDITOR** レジストリー変数に指定されているエディター
2. **VISUAL** 環境変数に指定されているエディター
3. **EDITOR** 環境変数に指定されているエディター
4. Windows オペレーティング・システムではメモ帳、UNIX オペレーティング・システムでは vi エディター

*num* *num* が正の場合は、*num* に対応するコマンドによってエディターを起動します。*num* が負の場合は、コマンド履歴の最新のコマンドから逆算する形で *num* に対応するコマンドによってエディターを起動します。ゼロは、*num* の有効な値ではありません。このパラメーターを指定しない場合は、実行した最新のコマンドによってエディターを起動します。(この動作は、*num* に -1 を指定した場合と同じです。)

### 使用上の注意

1. 指定するエディターは、オペレーティング・システムの **PATH** に入っている有効なエディターでなければなりません。
2. 最近実行した編集用コマンドのリストを表示するには、**HISTORY** コマンドを実行します。
3. **EDIT** コマンドは、どんな場合でもコマンド履歴に記録されません。ただし、**EDIT** コマンドによって編集したコマンドを実行する場合、そのコマンドはコマンド履歴に記録されます。

## EXPORT

データベースから、いくつかある外部ファイル形式のいずれかにデータをエクスポートします。ユーザーは、SQL SELECT ステートメントによって、または型付き表の階層情報によってエクスポートするデータを指定します。

206 ページの『エクスポート・ユーティリティー用のファイル・タイプ修飾子』へのクイック・リンク。

### 許可

以下の権限のいずれか。

- DATAACCESS 権限
- 関係するそれぞれの表またはビューに対する CONTROL または SELECT 特権

### 必要な接続

データベース。暗黙接続が可能な場合には、デフォルト・データベースへの接続が確立されます。Linux、UNIX、または Windows クライアントから Linux、UNIX、または Windows データベース・サーバーへのユーティリティー・アクセスは、DB2 Connect ゲートウェイまたはループバック環境を経由してではなく、エンジンを使用した直接接続でなければなりません。

### コマンド構文



**traversal-order-list:****コマンド・パラメーター****TO filename**

データのエクスポート先のファイルの名前を指定します。このファイルへの完全パスが指定されていない場合、エクスポート・ユーティリティーは現行のディレクトリーおよびデフォルトのドライブを宛先として使用します。

既に存在するファイルの名前を指定した場合、エクスポート・ユーティリティーはファイルの内容を上書きします。つまり情報は追加されません。

**OF filetype**

次のような出力ファイルのデータ・フォーマットを指定します。

- DEL (区切り文字付き ASCII フォーマット)。さまざまなデータベース・マネージャーやファイル・マネージャー・プログラムで使用します。
- IXF (統合交換フォーマット、PC バージョン) は、プロプラエタリー・バイナリー・フォーマットです。

**LOBS TO lob-path**

LOB ファイルが保管される、ディレクトリーへの 1 つ以上のパスを指定します。LOB パスごとに少なくとも 1 つのファイルが存在し、各ファイルには少なくとも 1 つの LOB が入ります。指定できるパスの最大数は 999 です。これによって、LOBSINFILE 動作が暗黙的にアクティブ化されます。

**LOBFILE filename**

LOB ファイルに 1 つ以上の基本ファイル名を指定します。最初の名前の名前空間がいっぱいになると、2 番目の名前が使用され、以下 3 番目、4 番目と続きます。これによって、LOBSINFILE 動作が暗黙的にアクティブ化されます。

エクスポート操作中に LOB ファイルを作成するときに、まずこのリストから現行パス (*lob-path* で指定されたパス) に現行のベース名を追加してから、最初に 3 桁のシーケンス番号、次に 3 文字の ID *lob* を追加して、ファイル名が構成されます。例えば、現行 LOB パスがディレクトリー `/u/foo/lob/path/` で、現行 LOB ファイル名が `bar` の場合、LOB ファイルは、`/u/foo/lob/path/bar.001.lob`、`/u/foo/lob/path/bar.002.lob` (以下 003、004 と続く) などのように作成されます。LOB ファイル名の 3 桁のシーケンス番号で 999 が使用されてしまうとこの番号は 4 桁に増え、また、9999 が使用されてしまうと 4 桁が 5 桁に増えます。以下同様にシーケンス番号の桁数が増えていきます。

**XML TO xml-path**

XML ファイルが保管されるディレクトリーを指す 1 つ以上のパスを指定します。XML パスごとに少なくとも 1 つのファイルが存在し、各ファイルには少なくとも 1 つの XQuery データ・モデル (XDM) インスタンスが含まれることとなります。複数のパスが指定された場合、XDM インスタンスはそれらのパスに均等に分散されます。

**XMLFILE** *filename*

XML ファイルのための 1 つ以上の基本ファイル名を指定します。最初の名前の名前空間がいっぱいになると、2 番目の名前が使用され、以下 3 番目、4 番目と続きます。

エクスポート操作中に XML ファイルを作成するときに、まずこのリストから現行パス (*xml-path* で指定されたパス) に現行のベース名を追加し、それに 3 桁のシーケンス番号を追加し、さらに 3 文字の ID *xml* を追加したファイル名が構成されます。例えば、現行 XML パスがディレクトリー `/u/foo/xml/path/` で、現行 XML ファイル名が `bar` の場合、XML ファイルは、`/u/foo/xml/path/bar.001.xml`、`/u/foo/xml/path/bar.002.xml` などのように作成されます。

**MODIFIED BY** *filetype-mod*

ファイル・タイプ修飾子オプションを指定します。206 ページの『エクスポート・ユーティリティー用のファイル・タイプ修飾子』を参照してください。

**XMLSAVESCHEMA**

すべての XML 列について XML スキーマ情報を保存することを指定します。エクスポートされた各 XML 文書のうち、挿入時に XML スキーマに関する妥当性検査が実行されたものについては、そのスキーマの完全修飾 SQL ID が、対応する XML Data Specifier (XDS) 内に SCH 属性として格納されます。エクスポートされた文書に対して XML スキーマに関する妥当性検査が実行されていない場合、またはそのスキーマ・オブジェクトがデータベースにもやば存在しない場合は、対応する XDS に SCH 属性は含まれません。

SQL ID のスキーマと名前の各部分は、XML スキーマに対応する SYSCAT.XSROBJECTS カタログ表の行の "OBJECTSCHEMA" および "OBJECTNAME" の値として格納されます。

**XMLSAVESCHEMA** オプションには、整形 XML 文書を生成しない XQuery シーケンスとの互換性がありません。

**METHOD N** *column-name*

出力ファイルで使用される 1 つ以上の列名を指定します。このパラメーターが指定されない場合、表の列名が使用されます。このパラメーターは IXF ファイルでのみ有効ですが、階層データをエクスポートするときは無効です。

**MESSAGES** *message-file*

エクスポート操作中に生じ得る警告およびエラー・メッセージの宛先を指定します。宛先ファイルが既に存在している場合、エクスポート・ユーティリティーは情報を追加します。 *message-file* を省略すると、メッセージは標準出力に書き込まれます。

*select-statement*

エクスポートされるデータを戻す SELECT または XQUERY ステートメントを指定します。このステートメントによってエラーが発生する場合、メッセージ・ファイル (または標準出力) にメッセージが書き込まれます。エラー・コードが SQL0012W、SQL0347W、SQL0360W、SQL0437W、または

SQL1824W である場合、エクスポート操作は続行します。これ以外のエラー・コードの場合、操作は停止します。

SELECT ステートメントが `SELECT * FROM tablename` という形式で、表に暗黙的な非表示列が含まれている場合には、エクスポート操作に非表示列のデータを含めるかどうかを明示的に指定する必要があります。非表示列のデータを含めるかどうかを示すには、以下のいずれかの方法を使用します。

- 非表示列のファイル・タイプ修飾子のいずれかを使用します。非表示列のデータをエクスポートに含める場合には `implicitlyhiddeninclude` を指定し、エクスポートに含めない場合には `implicitlyhiddenmissing` を指定します。

```
db2 export to t.del of del modified by implicitlyhiddeninclude
select * from t
```

- 暗黙的な非表示列が含まれる表をデータ移動ユーティリティーが検出した場合のデフォルト動作を設定するには、`DB2_DMU_DEFAULT` レジストリー変数をクライアント・サイドで使用します。

```
db2set DB2_DMU_DEFAULT=IMPLICITLYHIDDENINCLUDE
db2 export to t.del of del select * from t
```

#### **HIERARCHY STARTING** *sub-table-name*

デフォルトの走査順序 (ASC、または DEL ファイルの OUTER 順序、または PC/IXF データ・ファイルに保管されている順序) を使用して、*sub-table-name* から始まる副階層をエクスポートします。

#### **HIERARCHY** *traversal-order-list*

指定した走査順序を使用して副階層をエクスポートします。すべての副表は、PRE-ORDER 方式でリストされていなければなりません。最初の副表名が、SELECT ステートメントのターゲット表名として使用されます。

## 例

次に示すのは、SAMPLE データベースにある STAFF 表から、ファイル `myfile.ixf` に情報をエクスポートする方法の一例です。これは、IXF フォーマットで出力されます。コマンドを発行する前に、SAMPLE データベースと接続していなければなりません。データベース接続が DB2 Connect を介して確立されていない場合、索引定義 (もしあれば) は出力ファイルに保管されます。

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

次に示すのは、SAMPLE データベースの STAFF 表から部門 20 の従業員に関する情報を、エクスポートする方法の一例です。これは IXF フォーマットで出力され、`awards.ixf` ファイルに入ります。コマンドを発行する前に、SAMPLE データベースと接続していなければなりません。また、表の中の実際の列名は、'department' ではなく 'dept' であることにも注意してください。

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

次の例は LOB を DEL ファイルにエクスポートする方法を示しています。

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

次の例は LOB を DEL ファイルにエクスポートする方法を示しています。ここでは、最初のディレクトリーにファイルを入れることができない場合のために 2 番目のディレクトリーを指定しています。

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

次の例はデータを DEL ファイルにエクスポートする方法を示しています。ここでは、単一引用符をストリング区切り文字として使用し、セミコロンを列の区切り文字として使用し、コンマを小数点として使用します。データを再びデータベースにインポートする場合、これと同じ規則を使用する必要があります。

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

### 使用上の注意

- エクスポート操作を開始する前に、すべての表操作が完了し、すべてのロックが解除されていることを確認してください。これは、WITH HOLD でオープンされたすべてのカーソルをクローズした後で COMMIT を発行するか、または ROLLBACK を発行することによって行うことができます。
- SELECT ステートメントでは表の別名を使用できます。
- メッセージ・ファイルに置かれたメッセージには、メッセージ検索サービスから戻される情報が含まれています。各メッセージは新しい行から始まります。
- PC/IXF インポートは、データベース間でデータを移動する場合に使用します。行区切り文字を含む文字データが区切り文字付き ASCII (DEL) ファイルにエクスポートされ、テキスト転送プログラムによって処理される場合、行区切り文字を含むフィールドは長さが変わることがあります。
- ソースとターゲットのデータベースが両方とも同じクライアントからアクセス可能である場合、ファイルのコピーというステップは必要ありません。
- DB2 Connect を使用して、DB2 for OS/390、DB2 for VM and VSE、および DB2 for OS/400® などの DRDA サーバーから表をエクスポートすることができます。PC/IXF エクスポートだけがサポートされています。
- IXF フォーマットにエクスポートする際に、ID が IXF フォーマットでサポートされる最大サイズを超えている場合、エクスポートは成功しますが、生成されるデータ・ファイルは CREATE モードを使用する後続のインポート操作では使用できません。SQL27984W が戻されます。
- Windows でディスクットにエクスポートしているときに 1 枚のディスクットの容量を超えるデータを持つ表がある場合、別のディスクットを挿入するようプロンプトが出され、複数パーツ PC/IXF ファイル (マルチボリューム PC/IXF ファイル、または論理的に分割された PC/IXF ファイルとして知られる) が生成されて別々のディスクットに保管されます。最後のファイルを除く各ファイルには、そのファイルが論理的に分割されていること、および次のファイルを検索する場所を示すための DB2 CONTINUATION RECORD (または略して "AC" Record) が書き込まれます。その後、ファイルがインポート・ユーティリティーおよびロード・ユーティリティーによって読み取られるように、ファイルを AIX システムに転送できます。エクスポート・ユーティリティーは、AIX システムから呼び出される場合、複数部分からなる PC/IXF ファイルを作成しません。詳しい使用方法については、IMPORT コマンドまたは LOAD コマンドを参照してください。



- エクスポート・ユーティリティーは、提供される SELECT ステートメントが、SELECT \* FROM tablename という形式である場合、IXF ファイルの表の NOT NULL WITH DEFAULT 属性を保管します。
- 型付き表をエクスポートする場合、副選択ステートメントは、ターゲット表名と WHERE 節を指定することによってのみ表現することができます。階層をエクスポートする場合、全選択と select-statement は指定できません。
- IXF 以外のファイル形式の場合は、階層の全探索の方法、およびエクスポートする副表とが DB2 に知らされるよう、全探索順序リストを指定することをお勧めします。このリストを指定しない場合、階層内のすべての表がエクスポートされ、デフォルトの順序は OUTER 順序になります。OUTER 関数によって指定されるデフォルトの順序を使うこともできます。
- インポート操作の間も同じ走査順序を使用します。ロード・ユーティリティーでは、階層または副階層のロードはサポートされていません。
- 保護行のある表からデータをエクスポートする場合は、セッション許可 ID の保持する LBAC 信用証明情報のために、エクスポートされる行が制限されることがあります。セッション許可 ID に読み取りアクセスがない行はエクスポートされません。エラーも警告も出ません。
- セッション許可 ID の保持する LBAC 信用証明情報のために、エクスポートに含まれている 1 つ以上の保護列からの読み取りが許可されない場合、エクスポートは失敗し、エラー (SQLSTATE 42512) が戻されます。
- export や db2move などのデータ移動ユーティリティーを実行する際に、照会コンパイラーが、基礎照会を基本表よりも MQT に対して実行する方が効率が良いと判別することがあります。そのような場合、照会は据え置きリフレッシュの MQT に対して実行され、ユーティリティーの結果は基礎表内のデータを正確に表していない可能性があります。
- エクスポートするパッケージは DATETIME ISO 形式を使用してバインドされるため、ストリング表記へのキャスト時にすべての日付/時間/タイム・スタンプの値は ISO 形式に変換されます。CLP パッケージは DATETIME LOC 形式 (ロケール固有形式) を使用してバインドされるため、CLP DATETIME 形式が ISO と異なる場合に CLP とエクスポート間で矛盾した振る舞いが見られる場合があります。例えば、以下の SELECT ステートメントは、以下の期待される結果を戻しません。

```
db2 select col2 from tab1 where char(col2)='05/10/2005';
COL2
-----
05/10/2005
05/10/2005
05/10/2005
3 record(s) selected.
```

しかし、同じ select 節を使用した export コマンドでは、期待される結果が戻されません。

```
db2 export to test.del of del select col2 from test
where char(col2)='05/10/2005';
Number of rows exported: 0
```

ここで、LOCALE 日付形式を ISO 形式に置き換えると、以下のように、期待される結果になります。

## EXPORT

```
db2 export to test.del of del select col2 from test
where char(col2)='2005-05-10';
Number of rows exported: 3
```

### エクスポート・ユーティリティー用のファイル・タイプ修飾子

表 10. エクスポート・ユーティリティーで有効なファイル・タイプ修飾子: すべてのファイル・フォーマット

修飾子	説明
lobsinfile	<p><i>lob-path</i> には、LOB データの入ったファイルへのパスを指定します。</p> <p>各パスには、データ・ファイル内で LOB ロケーション指定子 (LLS) によって示される 1 つ以上の LOB の入った、少なくとも 1 つのファイルが組み込まれます。LLS は、LOB ファイル・パスに保管されるファイル内の LOB のロケーションのストリング表現です。LLS の形式は、<i>filename.ext.nnn.mmm/</i> です。<i>filename.ext</i> は LOB を収めたファイルの名前、<i>nnn</i> はファイル内の LOB のオフセット (バイト単位)、<i>mmm</i> は LOB の長さ (バイト単位) を表します。例えば、ストリング <i>db2exp.001.123.456/</i> がデータ・ファイルに保管される場合、LOB はファイル <i>db2exp.001</i> のオフセット 123 に位置し、456 バイト長です。</p> <p><b>EXPORT</b> の使用時に <b>lobsinfile</b> 修飾子を指定した場合、LOB データは <b>LOBS TO</b> 節に指定されたロケーションに置かれます。指定しない場合、LOB データはデータ・ファイル・ディレクトリーに送られます。<b>LOBS TO</b> 節は、LOB ファイルが保管されるディレクトリーへのパスを 1 つ以上指定します。LOB パスごとに少なくとも 1 つのファイルが存在し、各ファイルには少なくとも 1 つの LOB が入ります。<b>LOBS TO</b> または <b>LOBFILE</b> オプションによって、<b>LOBSINFILE</b> 動作が暗黙的にアクティブ化されます。</p> <p>NULL LOB を指定するには、サイズに -1 と入力します。サイズを 0 と指定すると、長さが 0 の LOB として扱われます。長さが -1 の NULL LOB の場合、オフセットとファイル名は無視されます。例えば、NULL LOB の LLS は、<i>db2exp.001.7.-1/</i> のようになります。</p>
implicitlyhiddeninclude	<p>この修飾子は <b>SELECT *</b> 照会と一緒に使用し、<b>SELECT *</b> 照会の結果に含まれない暗黙的な非表示列のデータもエクスポートすることを指定します。この修飾子は、<b>implicitlyhiddenmissing</b> 修飾子と共に使用することはできません。</p> <p>この修飾子を使用して、照会に <b>SELECT *</b> が含まれていないと、エラーが戻りません (SQLCODE SQL3526N)。</p>
implicitlyhiddenmissing	<p>この修飾子は <b>SELECT *</b> 照会と一緒に使用し、暗黙的な非表示列のデータをエクスポートしないことを指定します。この修飾子は、<b>implicitlyhiddeninclude</b> 修飾子と共に使用することはできません。</p> <p>この修飾子を使用して、照会に <b>SELECT *</b> が含まれていないと、エラーが戻りません (SQLCODE SQL3526N)。</p>
xmlinsepfiles	<p>各 XQuery データ・モデル (XDM) インスタンスが別のファイルに書き込まれます。デフォルトでは、同じファイルの中で複数の値が連結されます。</p>
lobsinsepfiles	<p>各 LOB 値が別のファイルに書き込まれます。デフォルトでは、同じファイルの中で複数の値が連結されます。</p>
xmlnodeclaration	<p>XDM インスタンスが XML 宣言タグなしで書き込まれます。デフォルトでは、XDM インスタンスのエクスポート時に、エンコード属性を指定した XML 宣言タグが先頭に組み込まれます。</p>

表 10. エクスポート・ユーティリティーで有効なファイル・タイプ修飾子: すべてのファイル・フォーマット (続き)

修飾子	説明
xmlchar	XDM インスタンスが文字コード・ページで書き込まれます。文字コード・ページは <code>codepage</code> ファイル・タイプ修飾子で指定されている値であるか、または指定がない場合はアプリケーションのコード・ページであることに注意してください。デフォルトでは、XDM インスタンスは、Unicode で書き込まれます。
xmlgraphic	<b>EXPORT</b> コマンドで <code>xmlgraphic</code> 修飾子を指定すると、アプリケーション・コード・ページまたは <code>codepage</code> ファイル・タイプ修飾子にかかわりなく、エクスポート XML 文書は、UTF-16 コード・ページでエンコードされます。

表 11. エクスポート・ユーティリティーの有効なファイル・タイプ修飾子: `DEL` (区切り付き ASCII) ファイル形式

修飾子	説明
chardelx	<p><code>x</code> は単一文字の文字区切り文字です。デフォルト値は、二重引用符 (") です。文字ストリングを囲む二重引用符の代わりに指定の文字を使用します。<sup>2</sup> 文字ストリング区切りとして二重引用符を明示的に指定する場合は、以下のように指定します。</p> <pre>modified by chardel""</pre> <p>以下のように、文字ストリング区切りとして単一引用符 (') を指定することもできます。</p> <pre>modified by chardel''</pre>
codepage=x	<p><code>x</code> は、ASCII 文字ストリングです。この値は、出力データ・セットに含まれているデータのコード・ページとして解釈されます。エクスポート操作の実行中に、文字データは、アプリケーション・コード・ページからこのコード・ページに変換されます。</p> <p>DBCS のみ (GRAPHIC)、混合 DBCS、および EUC の場合、区切り文字の範囲は <code>x00</code> から <code>x3F</code> に制限されます。<code>codepage</code> 修飾子を <code>lobsinfile</code> 修飾子と一緒に使用することはできません。</p>
coldelx	<p><code>x</code> は単一文字の列区切り文字です。デフォルト値はコンマ (,) です。列の終わりを示すコンマの代わりに指定の文字を使用します。<sup>2</sup></p> <p>以下の例では <code>coldel;</code> を指定しているため、エクスポート・ユーティリティーは、セミコロン文字 (;) をエクスポート・データの列区切りとして使用します。</p> <pre>db2 "export to temp of del modified by coldel; select * from staff where dept = 20"</pre>
decplusblank	正符号文字。正の 10 進値の接頭部として、正符号 (+) ではなく空白・スペースを使用します。デフォルトのアクションでは、正の 10 進数の前に正符号 (+) が付けられます。
decptx	<code>x</code> は、小数点文字としてピリオドの代わりに使用される単一の置換文字です。デフォルト値はピリオド (.) です。小数点文字として、ピリオドの代わりに指定の文字を使用します。 <sup>2</sup>



表 11. エクスポート・ユーティリティの有効なファイル・タイプ修飾子: DEL (区切り付き ASCII) ファイル形式 (続き)

修飾子	説明
timestampformat="x"	<p>x は、ソース・ファイルのタイム・スタンプの形式です。<sup>4</sup> 有効なタイム・スタンプ・エレメントは、以下のとおりです。</p> <p><b>YYYY</b> 年 (0000 から 9999 の範囲の 4 桁の数)</p> <p><b>M</b> 月 (1 から 12 の範囲の 1 桁または 2 桁の数)</p> <p><b>MM</b> 月 (01 から 12 の 2 桁の数。M および MMM とは相互に排他的)</p> <p><b>MMM</b> 月 (大文字小文字を区別しない月名の 3 文字の省略形。M と MM とは相互に排他的)</p> <p><b>D</b> 日 (1 から 31 の範囲の 1 桁または 2 桁の数)</p> <p><b>DD</b> 日 (01 から 31 の範囲の 2 桁の数。D とは相互に排他的)</p> <p><b>DDD</b> 元日から数えた日数 (001 から 366 の範囲の 3 桁の数。他の日または月のエレメントとは相互に排他的)</p> <p><b>H</b> 時 (12 時間制の場合は 0 から 12、24 時間制では 0 から 24 の範囲の 1 桁または 2 桁の数。)</p> <p><b>HH</b> 時 (12 時間制の場合は 00 から 12、24 時間制では 0 から 24 の範囲の 2 桁の数。H と相互に排他的)</p> <p><b>M</b> 分 (0 から 59 の範囲の 1 桁または 2 桁の数)</p> <p><b>MM</b> 分 (00 から 59 の範囲の 2 桁の数。M (分) とは相互に排他的)</p> <p><b>S</b> 秒 (0 から 59 の範囲の 1 桁または 2 桁の数)</p> <p><b>SS</b> 秒 (00 から 59 の範囲の 2 桁の数。S とは相互に排他的)</p> <p><b>SSSSS</b> 夜中の 12 時から数えた秒数 (00000 から 86400 の範囲の 5 桁の数。他の時刻エレメントとは相互に排他的)</p> <p><b>U (1 から 12 時)</b>  <small>小数秒 (U のオカレンス数は、各桁を 0 から 9 の範囲として、桁数を表します)</small></p> <p><b>TT</b> 午前/午後の指定子 (AM または PM)</p> <p>タイム・スタンプ・フォーマットの例を以下に示します。</p> <p style="text-align: center;">"YYYY/MM/DD HH:MM:SS.UUUUUU"</p> <p>MMM エレメントは、以下の値を生成します。  「Jan」、「Feb」、「Mar」、「Apr」、「May」、「Jun」、「Jul」、「Aug」、  「Sep」、「Oct」、「Nov」、および「Dec」。「Jan」は 1 月と等しく、「Dec」は 12 月と等しいです。</p> <p>以下の例は、「schedule」という表から、ユーザー定義のタイム・スタンプ・フォーマットを示すデータをエクスポートする方法を示しています。</p> <pre>db2 export to delfile2 of del modified by timestampformat="yyyy.mm.dd hh:mm tt" select * from schedule</pre>

表 12. エクスポート・ユーティリティで有効なファイル・タイプ修飾子: IXF ファイル・フォーマット

修飾子	説明
codepage=x	<p>x は、ASCII 文字ストリングです。この値は、出力データ・セットに含まれているデータのコード・ページとして解釈されます。エクスポート操作中に、文字データをこのコード・ページからアプリケーション・コード・ページに変換します。</p> <p>DBCS のみ (GRAPHIC)、混合 DBCS、および EUC の場合、区切り文字の範囲は x00 から x3F に制限されます。</p>

## 注:

- サポートされていないファイル・タイプを **MODIFIED BY** オプションで使用しようとしても、エクスポート・ユーティリティは警告を出しません。サポートされていないファイル・タイプを使おうとすると、エクスポート操作は失敗し、エラー・コードが戻されます。
- データ移動のための区切り文字の制約事項に、区切り文字の指定変更として使用できる文字に適用される制限のリストが示されています。
- 通常、エクスポート・ユーティリティは、以下のような形式で書き込みを行います。

- 日付データ: *YYYYMMDD* の形式
- 文字 (日付) データ: *YYYY-MM-DD* の形式
- 時刻データ: *HH.MM.SS* の形式
- タイム・スタンプ・データ: *YYYY-MM-DD-HH.MM.SS.aaaaaa* の形式

エクスポート操作のために **SELECT** ステートメントで指定される日時列に組み込まれたデータも、これらの形式になります。

- タイム・スタンプ・フォーマットの場合、月の記述子と分の記述子のどちらも文字 **M** を使用するため、区別があいまいにならないように注意する必要があります。月のフィールドは、他の日付フィールドと隣接していなければなりません。分フィールドは、他の時刻フィールドに隣接していなければなりません。あいまいなタイム・スタンプ形式の例を以下に示します。

"M" (月または分のどちらにもとれる)  
 "M:M" (月と分の区別がつかない)  
 "M:YYYY:M" (両方とも月と解釈される)  
 "S:M:YYYY" (時刻値と日付値の両方に隣接している)

あいまいな場合、ユーティリティはエラー・メッセージを報告し、操作は失敗します。

以下に、明確なタイム・スタンプ・フォーマットを示します。

"M:YYYY" (M (月))  
 "S:M" (M (分))  
 "M:YYYY:S:M" (M (月)...M (分))  
 "M:H:YYYY:M:D" (M (分)...M (月))

- XMLFILE** 節または **XML TO** 節のどちらも指定されていない場合でも、すべての XDM インスタンスは、メイン・データ・ファイルとは別個の XML ファイルに書き込まれます。デフォルトで、XML ファイルはエクスポートされるデータ・ファイルのパスに書き込まれます。XML ファイルのデフォルトのベース名は、エクスポートされるデータ・ファイル名に拡張子 ".xml" を追加したものとなります。

6. XMLNODEDECLARATION ファイル・タイプ修飾子が指定されていない場合は、すべての XDM インスタンスはエンコード属性を含む XML 宣言を先頭に付けて書き込まれます。
7. XMLCHAR または XMLGRAPHIC ファイル・タイプ修飾子が指定されていない場合は、デフォルトで、すべての XDM インスタンスは Unicode で書き込まれます。
8. XML データおよび LOB データのデフォルト・パスは、メイン・データ・ファイルのパスです。デフォルトの XML ファイルのベース名は、メイン・データ・ファイルです。デフォルトの LOB ファイルのベース名は、メイン・データ・ファイルです。例えば、メイン・データ・ファイルが

```
/mypath/myfile.del
```

の場合、XML データおよび LOB データのデフォルト・パスは

```
/mypath"
```

となり、デフォルトの XML ファイルのベース名は

```
myfile.del
```

となり、デフォルトの LOB ファイルのベース名は

```
myfile.del
```

LOB ファイルを生成するには、LOBSINFILE ファイル・タイプ修飾子が指定されている必要があります。

9. エクスポート・ユーティリティは、数値 ID を各 LOB ファイルまたは XML ファイルに付加します。この ID は、0 が埋め込まれた 3 桁のシーケンス値で、
- ```
.001
```

から始まります。999 番目の LOB ファイルまたは XML ファイルの後には、ID への 0 の埋め込みは行われません (例えば、1000 番目の LOG ファイルまたは XML ファイルの拡張子は

```
.1000
```

となります)。数値 ID の後に、データ・タイプを表す次のどちらかの 3 文字のタイプ ID が続きます。

```
.lob
```

または

```
.xml
```

例えば、生成される LOB ファイルの名前は

```
myfile.del.001.lob
```

の形式となり、生成される XML ファイルの名前は

```
myfile.del.001.xml
```

の形式となります。

10. XQuery を指定すると、エクスポート・ユーティリティは、整形形式でない文書である XDM インスタンスをエクスポートできます。ただし、XML 列に含め

## EXPORT

ることができるのは完全な文書だけなので、これらのエクスポートされた文書を XML 列に直接インポートまたはロードすることはできません。



## FORCE APPLICATION

システムからローカルまたはリモートのユーザーやアプリケーションを強制終了し、サーバー上での保守を可能にします。

**重要:** 割り込みが許されない操作 (例えば、**RESTORE DATABASE**) を強制終了する場合は、その操作を再び実行して正常終了しなければデータベースは使用可能になりません。

### 有効範囲

このコマンドは、`$HOME/sql1lib/db2nodes.cfg` ファイルにリストされているすべてのデータベース・パーティションに影響を与えます。

パーティション・データベース環境では、このコマンドを発行するのは、強制終了するアプリケーションのコーディネーター・データベース・パーティションからでなくてもかまいません。パーティション・データベース環境ではどのデータベース・パーティション・サーバーからでも発行できます。

### 許可

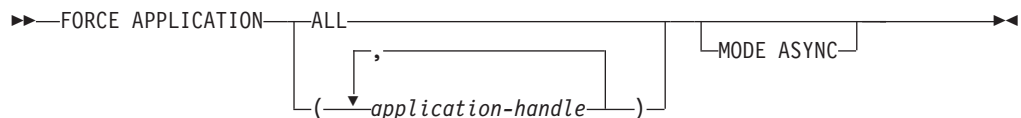
以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYMAINT

### 必要な接続

インスタンス。リモート・サーバーからユーザーを強制終了する場合、最初にそのサーバーにアタッチする必要があります。アタッチが存在しない場合、このコマンドはローカルで実行されます。

### コマンド構文



### コマンド・パラメーター

#### FORCE APPLICATION

**ALL** すべてのアプリケーションがデータベースから切断されます。

*application-handle*

エージェントの終了を指定します。 **LIST APPLICATIONS** コマンドを使用して値をリストします。

#### MODE ASYNC

コマンドは、指定したすべてのユーザーが終了するのを待たずに戻ってきます。コマンドは、機能を正常に発行するか、またはエラー (無効な構文などの) を発見するとすぐに戻ります。

## FORCE APPLICATION

現在サポートしているモードはこのモードだけです。

### 例

次の例は、*application-handle* の値が 41408 と 55458 の 2 つのユーザーをデータベースから強制的に切断します。

```
db2 "force application ( 41408, 55458 )"
```

### 使用上の注意

データベース・マネージャーは、**db2start** を必要とせずに、後続のデータベース・マネージャー操作を処理できるようにするため、アクティブなままになっています。

データベースの整合性を確保するため、終了できるのは、アイドル中のユーザー、または割り込み可能なデータベース操作を実行中のユーザーだけです。

以下のタイプのユーザーおよびアプリケーションは、強制終了できません。

- データベースを作成しているユーザー
- システム・アプリケーション

これらのタイプのユーザーおよびアプリケーションを正常に強制終了するには、データベースの非活動化とインスタンスの再始動のいずれかまたは両方を行う必要があります。

**FORCE APPLICATION** が出された後も、データベースはまだ接続要求を受諾します。すべてのユーザーを完全に強制終了するためには、追加の **FORCE** が必要になる場合があります。

## GET ADMIN CONFIGURATION

システムの管理ノードにある、個々の DB2 Administration Server (DAS) 構成パラメータの値を戻します。DAS は、DB2 サーバーのリモート管理を有効にする特別な管理ツールです。

DAS 構成パラメータのリストは、**UPDATE ADMIN CONFIGURATION** コマンドの説明を参照してください。

**重要:** DB2 Administration Server (DAS) は、バージョン 9.7 で非推奨となり、将来のリリースで除去される可能性があります。DAS は、DB2 pureScale環境ではサポートされていません。リモート管理のためには、Secure Shell プロトコルを使用するソフトウェア・プログラムを使用してください。詳しくは、『DB2 Administration Server (DAS) が推奨されなくなった』( ) を参照してください。

### 有効範囲

このコマンドは、アタッチするシステム、または **FOR NODE** オプションで指定するシステムの管理ノードにある、DAS 構成パラメータに関する情報を戻します。

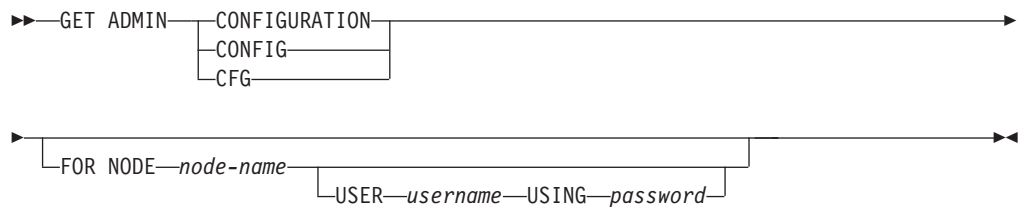
### 許可

なし

### 必要な接続

ノード。リモート・システムの DAS 構成を表示する場合は、まずそのシステムに接続するか、**FOR NODE** オプションを使用してシステムの管理ノードを指定します。

### コマンド構文



### コマンド・パラメーター

**FOR NODE** *node-name*

DAS 構成パラメータを表示する管理ノードの名前を入力します。

**USER** *username* **USING** *password*

ノードへの接続にユーザー名とパスワードが必要な場合は、この情報を入力します。

### 例

以下に示すのは、**GET ADMIN CONFIGURATION** の出力例です。

Admin Server Configuration

Authentication Type DAS (AUTHENTICATION) = SERVER\_ENCRYPT

## GET ADMIN CONFIGURATION

```
DAS Administration Authority Group Name (DASADM_GROUP) = ADMINISTRATORS
DAS Discovery Mode (DISCOVER) = SEARCH
Name of the DB2 Server System (DB2SYSTEM) = swalkty
Java Development Kit Installation Path DAS (JDK_PATH) = e:%sqllib%java%jdk
DAS Code Page (DAS_CODEPAGE) = 0
DAS Territory (DAS_TERRITORY) = 0
Location of Contact List (CONTACT_HOST) = hostA.ibm.ca
Execute Expired Tasks (EXEC_EXP_TASK) = NO
Scheduler Mode (SCHED_ENABLE) = ON
SMTP Server (SMTP_SERVER) = smtp1.ibm.ca
Tools Catalog Database (TOOLSCAT_DB) = CCMD
Tools Catalog Database Instance (TOOLSCAT_INST) = DB2
Tools Catalog Database Schema (TOOLSCAT_SCHEMA) = TOOLSCAT
Scheduler User ID = db2admin
```

### 使用上の注意

エラーが生じた場合には、戻された情報は無効になります。構成ファイルが無効な場合には、エラー・メッセージが戻されます。そのような場合には、DAS を再インストールしてリカバリーする必要があります。

DAS 出荷時のデフォルト値に構成パラメーターを設定するには、**RESET ADMIN CONFIGURATION** コマンドを使用してください。

## GET ALERT CONFIGURATION

特定のインスタンスに関するヘルス・インディケーターのアラート構成設定を取得します。

**重要:** バージョン 9.7 でヘルス・モニターが非推奨になったため、このコマンドおよび API は推奨されておらず、将来のリリースで除去される可能性があります。これはDB2 pureScale環境ではサポートされていません。詳しくは、『ヘルス・モニターが推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055045.html>) を参照してください。

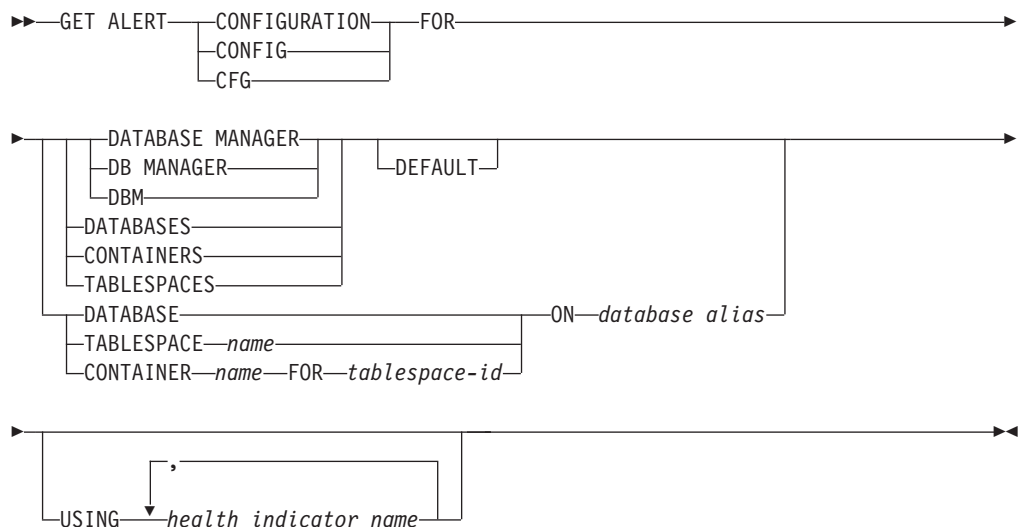
### 許可

なし

### 必要な接続

インスタンス。明示的なアタッチは必要ありません。

### コマンド構文



### コマンド・パラメーター

#### DATABASE MANAGER

データベース・マネージャーのアラート設定を検索します。

#### DATABASES

データベース・マネージャーによって管理されるすべてのデータベースのアラート設定を検索します。これは、カスタム設定を持たないすべてのデータベースに適用される設定です。カスタム設定は、**DATABASE ON database alias** 節を使って定義されます。

#### CONTAINERS

データベース・マネージャーによって管理されるすべての表スペース・コンテナのアラート設定を検索します。これは、カスタム設定を持たないすべ

## GET ALERT CONFIGURATION

ての表スペース・コンテナに適用される設定です。カスタム設定は、**CONTAINER name ON database alias** 節を使って定義されます。

### TABLESPACES

データベース・マネージャーによって管理されるすべての表スペースのアラート設定を検索します。これは、カスタム設定を持たないすべての表スペースに適用される設定です。カスタム設定は、**TABLESPACE name ON database alias** 節を使って定義されます。

### DEFAULT

インストール・デフォルトを検索するように指定します。

### DATABASE ON database alias

**ON database alias** 節を使って指定したデータベースのアラート設定を検索します。このデータベースがカスタム設定を持たない場合、インスタンスの全データベースの設定が戻されます。これは、**DATABASES** パラメーターと同じ結果です。

### CONTAINER name FOR tablespace-id ON database alias

**ON database alias** 節を使って指定したデータベース上で、**FOR tablespace-id** 節を使って指定した表スペースの、*name* という名前の表スペース・コンテナのアラート設定を検索します。この表スペース・コンテナがカスタム設定を持たない場合、データベースの全表スペース・コンテナの設定が戻されます。これは、**CONTAINERS** パラメーターと同じ結果です。

### TABLESPACE name ON database alias

**ON database alias** 節を使って指定したデータベース上で、*name* という名前の表スペースのアラート設定を検索します。この表スペースがカスタム設定を持たない場合、データベースの全表スペースの設定が戻されます。これは、**TABLESPACES** パラメーターと同じ結果です。

### USING health indicator name

アラート構成情報が戻されるヘルス・インディケーターのセットを指定します。ヘルス・インディケーター名は 2 文字のオブジェクト ID で構成され、その後にインディケーターの測定対象を説明する名前が続きます。例えば、**db.sort\_privmem\_util** のようになります。これはオプションの節で、これを使用しない場合は、指定したオブジェクトまたはオブジェクト・タイプのすべてのヘルス・インディケーターが戻されます。

## 例

以下のセクションに示すのは、データベース・マネージャー情報を要求した結果として表示される一般的な出力です。

DB2 GET ALERT CFG FOR DBM

```
Alert Configuration
Indicator Name           = db2.db2_op_status
Default                  = Yes
Type                    = State-based
Sensitivity              = 0
Formula                  = db2.db2_status;
Actions                  = Disabled
Threshold or State checking = Enabled

Indicator Name           = db2.sort_privmem_util
Default                  = Yes
```

## GET ALERT CONFIGURATION

```
Type = Threshold-based
Warning = 90
Alarm = 100
Unit = %
Sensitivity = 0
Formula = ((db2.sort_heap_allocated/sheapthres)
          *100);
Actions = Disabled
Threshold or State checking = Enabled

Indicator Name = db2.mon_heap_util
Default = Yes
Type = Threshold-based
Warning = 85
Alarm = 95
Unit = %
Sensitivity = 0
Formula = ((db2.mon_heap_cur_size/
          db2.mon_heap_max_size)*100);
Actions = Disabled
Threshold or State checking = Enabled
```

以下のセクションに示すのは、構成情報を要求した結果として表示される一般的な出力です。

### DB2 GET ALERT CFG FOR DATABASES

```
Alert Configuration
Indicator Name = db.db_op_status
Default = Yes
Type = State-based
Sensitivity = 0
Formula = db.db_status;
Actions = Disabled
Threshold or State checking = Enabled

Indicator Name = db.sort_shrmem_util
Default = Yes
Type = Threshold-based
Warning = 70
Alarm = 85
Unit = %
Sensitivity = 0
Formula = ((db.sort_shrheap_allocated/sheapthres_shr)
          *100);
Actions = Disabled
Threshold or State checking = Enabled

Indicator Name = db.spilled_sorts
Default = Yes
Type = Threshold-based
Warning = 30
Alarm = 50
Unit = %
Sensitivity = 0
Formula = ((delta(db.sort_overflows,10))/
          (delta(db.total_sorts,10)+1)*100);
Actions = Disabled
Threshold or State checking = Enabled

Indicator Name = db.max_sort_shrmem_util
Default = Yes
Type = Threshold-based
Warning = 60
Alarm = 30
Unit = %
Sensitivity = 0
```

## GET ALERT CONFIGURATION

```
Formula = ((db.max_shr_sort_mem/
           sheapthres_shr)*100);
Actions = Disabled
Threshold or State checking = Enabled

Indicator Name = db.log_util
Default = Yes
Type = Threshold-based
Warning = 75
Alarm = 85
Unit = %
Sensitivity = 0
Formula = (db.total_log_used/
           (db.total_log_used+db.total_log_available)
           )*100;
Actions = Disabled
Threshold or State checking = Enabled

Indicator Name = db.log_fs_util
Default = Yes
Type = Threshold-based
Warning = 75
Alarm = 85
Unit = %
Sensitivity = 0
Formula = ((os.fs_used/os.fs_total)*100);
Actions = Disabled
Threshold or State checking = Enabled

Indicator Name = db.deadlock_rate
Default = Yes
Type = Threshold-based
Warning = 5
Alarm = 10
Unit = Deadlocks per hour
Sensitivity = 0
Formula = delta(db.deadlocks);
Actions = Disabled
Threshold or State checking = Enabled

Indicator Name = db.locklist_util
Default = Yes
Type = Threshold-based
Warning = 75
Alarm = 85
Unit = %
Sensitivity = 0
Formula = (db.lock_list_in_use/(locklist*4096))
           *100;
Actions = Disabled
Threshold or State checking = Enabled

Indicator Name = db.lock_escal_rate
Default = Yes
Type = Threshold-based
Warning = 5
Alarm = 10
Unit = Lock escalations per hour
Sensitivity = 0
Formula = delta(db.lock_escal);
Actions = Disabled
Threshold or State checking = Enabled

Indicator Name = db.apps_waiting_locks
Default = Yes
Type = Threshold-based
Warning = 50
```



## GET ALERT CONFIGURATION

```

Alarm = 70
Unit = %
Sensitivity = 0
Formula = (db.locks_waiting/db.appls_cur_cons)*100;

Actions = Disabled
Threshold or State checking = Enabled

Indicator Name = db.pkgcache_hitratio
Default = Yes
Type = Threshold-based
Warning = 80
Alarm = 70
Unit = %
Sensitivity = 0
Formula = (1-
           (db.pkg_cache_inserts/db.pkg_cache_lookups)
           )*100;

Actions = Disabled
Threshold or State checking = Disabled

Indicator Name = db.catcache_hitratio
Default = Yes
Type = Threshold-based
Warning = 80
Alarm = 70
Unit = %
Sensitivity = 0
Formula = (1-
           (db.cat_cache_inserts/db.cat_cache_lookups)
           )*100;

Actions = Disabled
Threshold or State checking = Disabled

Indicator Name = db.shrworkspace_hitratio
Default = Yes
Type = Threshold-based
Warning = 80
Alarm = 70
Unit = %
Sensitivity = 0
Formula = ((1-
           (db.shr_workspace_section_inserts/
            db.shr_workspace_section_lookups))
           *100);

Actions = Disabled
Threshold or State checking = Disabled

Indicator Name = db.db_heap_util
Default = Yes
Type = Threshold-based
Warning = 85
Alarm = 95
Unit = %
Sensitivity = 0
Formula = ((db.db_heap_cur_size/
            db.db_heap_max_size)*100);

Actions = Disabled
Threshold or State checking = Enabled

Indicator Name = db.tb_reorg_req
Default = Yes
Type = Collection state-based
Sensitivity = 0
Actions = Disabled
Threshold or State checking = Disabled

```

## GET ALERT CONFIGURATION

|                             |                                                                                   |
|-----------------------------|-----------------------------------------------------------------------------------|
| Indicator Name              | = db.hadr_op_status                                                               |
| Default                     | = Yes                                                                             |
| Type                        | = State-based                                                                     |
| Sensitivity                 | = 0                                                                               |
| Formula                     | = db.hadr_connect_status;                                                         |
| Actions                     | = Disabled                                                                        |
| Threshold or State checking | = Enabled                                                                         |
| Indicator Name              | = db.hadr_delay                                                                   |
| Default                     | = Yes                                                                             |
| Type                        | = Threshold-based                                                                 |
| Warning                     | = 10                                                                              |
| Alarm                       | = 15                                                                              |
| Unit                        | = Minutes                                                                         |
| Sensitivity                 | = 0                                                                               |
| Formula                     | = (db.hadr_log_gap*var.refresh_rate/60)<br>DIV(delta(db.hadr_secondary_log_pos)); |
| Actions                     | = Disabled                                                                        |
| Threshold or State checking | = Enabled                                                                         |
| Indicator Name              | = db.db_backup_req                                                                |
| Default                     | = Yes                                                                             |
| Type                        | = State-based                                                                     |
| Sensitivity                 | = 0                                                                               |
| Actions                     | = Disabled                                                                        |
| Threshold or State checking | = Disabled                                                                        |
| Indicator Name              | = db.fed_nicknames_op_status                                                      |
| Default                     | = Yes                                                                             |
| Type                        | = Collection state-based                                                          |
| Sensitivity                 | = 0                                                                               |
| Actions                     | = Disabled                                                                        |
| Threshold or State checking | = Disabled                                                                        |
| Indicator Name              | = db.fed_servers_op_status                                                        |
| Default                     | = Yes                                                                             |
| Type                        | = Collection state-based                                                          |
| Sensitivity                 | = 0                                                                               |
| Actions                     | = Disabled                                                                        |
| Threshold or State checking | = Disabled                                                                        |
| Indicator Name              | = db.tb_runstats_req                                                              |
| Default                     | = Yes                                                                             |
| Type                        | = Collection state-based                                                          |
| Sensitivity                 | = 0                                                                               |
| Actions                     | = Disabled                                                                        |
| Threshold or State checking | = Disabled                                                                        |

## GET CLI CONFIGURATION

db2cli.ini ファイルの内容をリスト表示します。ファイル全体または指定したセクションをリスト表示することができます。

db2cli.ini ファイルは、DB2 コール・レベル・インターフェース (CLI) 構成ファイルとして使用されます。このファイルには、CLI およびそれを使用するアプリケーションの動作を変更するために使用できるさまざまなキーワードと値が含まれます。このファイルは複数のセクションに分かれており、それぞれのセクションはデータベース別名に対応します。

### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### AT GLOBAL LEVEL

LDAP ディレクトリーのデフォルト CLI 構成パラメーターを表示します。このパラメーターは Windows オペレーティング・システム上だけで有効です。

#### FOR SECTION *section-name*

キーワードがリスト表示されるセクションの名前。指定しない場合、すべてのセクションがリスト表示されます。

### 例

以下の出力例は、2 つのセクションがある db2cli.ini ファイルの内容を表しています。

```

[tstcli1x]
uid=userid
pwd=password
autocommit=0
TableType="'TABLE','VIEW','SYSTEM TABLE'"

[tstcli2x]
SchemaList="'OWNER1','OWNER2',CURRENT SQLID"
  
```

### 使用上の注意

このコマンドで指定するセクション名では、大/小文字の区別がありません。例えば、db2cli.ini ファイルのセクション名 (大括弧で区切られる) が小文字であり、コマンドで指定したセクション名が大文字であっても、正しいセクションがリスト表示されます。

PWD (パスワード) キーワードの値がリスト表示されることはありません。代わりに、5 つのアスタリスク (\*\*\*\*\*) がリスト表示されます。

LDAP (Lightweight Directory Access Protocol) が有効な場合、CLI 構成パラメータを、マシン・レベルに加えてユーザー・レベルでも設定できます。ユーザー・レベルの CLI 構成は、LDAP ディレクトリーに保持されます。指定されたセクションがユーザー・レベルで存在する場合、ユーザー・レベルでのそのセクションの CLI 構成が戻されます。そうでない場合、マシン・レベルの CLI 構成が戻されます。

ユーザー・レベルの CLI 構成は、LDAP ディレクトリーに保持され、ローカル・マシンでキャッシュされます。CLI 構成をユーザー・レベルで読み取る場合、DB2 は常にキャッシュから読み取ります。キャッシュは、次のときに更新されます。

- ユーザーが CLI 構成を更新するとき。
- ユーザーが **REFRESH LDAP** コマンドを使用して、明示的に CLI 構成の最新表示を強制するとき。

LDAP 環境では、ユーザーは LDAP ディレクトリーにカタログされたデータベースに対して、デフォルト CLI 設定値のセットを構成することができます。LDAP カタログ・データベースが、データ・ソース名 (DSN) として、CLI/ODBC 構成ユーティリティを使用して追加されると、デフォルトの CLI 設定が LDAP ディレクトリーにある場合には、それらはローカル・マシン上のその DSN 用に構成されます。デフォルトの CLI 設定を表示するには、**AT GLOBAL LEVEL** 節を指定する必要があります。

## GET CONNECTION STATE

接続状態を表示します。

以下の状態のいずれかが表示されます。

- 接続可能で接続済み
- 接続可能で未接続
- 接続不可能で接続済み
- 暗黙接続可能 (暗黙接続が使用できる場合に限る)

このコマンドは、以下の事柄に関する情報も戻します。

- データベース接続モード (SHARE または EXCLUSIVE)
- 接続が存在する場合の接続先のデータベースの別名および名前
- 接続が TCP/IP を使用している場合、接続のホスト名およびサービス名

### 許可

なし

### 必要な接続

なし

### コマンド構文

▶▶ GET CONNECTION STATE ◀◀

### コマンド・パラメーター

なし

### 例

以下に示すのは、**GET CONNECTION STATE** の出力例です。

```
Database Connection State

Connection state      = Connectable and Connected
Connection mode       = SHARE
Local database alias  = SAMPLE
Database name         = SAMPLE
Hostname              = montero
Service name          = 29384
```

### 使用上の注意

このコマンドは、タイプ 2 接続には適用されません。

### GET CONTACTGROUP

ローカル・システムで定義される、1つの連絡先グループに含まれる連絡先を取得します。連絡先とは、スケジューラーおよびヘルス・モニターがメッセージを送信する先のユーザーです。

**ADD CONTACTGROUP** コマンドを使用すると、名前付きの連絡先のグループを作成できます。

#### 許可

なし

#### 必要な接続

なし。ローカル実行のみ: このコマンドはリモート接続では使用できません。

#### コマンド構文

▶▶ GET CONTACTGROUP *name* ◀◀

#### コマンド・パラメーター

**CONTACTGROUP** *name*  
連絡先を検索するグループの名前。

#### 例

```
GET CONTACTGROUP support
Description
-----
Foo Widgets broadloom support unit
```

| Name    | Type          |
|---------|---------------|
| -----   | -----         |
| joe     | contact       |
| support | contact group |
| joline  | contact       |

## GET CONTACTGROUPS

システムにローカルに定義されるか、またはグローバル・リストに定義される、連絡先グループのリストを示します。連絡先グループは、スケジューラーおよびヘルス・モニターなどのモニター・プロセスが、メッセージを送信する先のアドレスのリストです。

Database Administration Server (DAS) の **contact\_host** 構成パラメーターの設定により、リストがローカルかグローバルかが決まります。 **ADD CONTACTGROUP** コマンドを使用すると、名前付きの連絡先のグループを作成できます。

### 許可

なし

### 必要な接続

なし

### コマンド構文

▶▶ GET CONTACTGROUPS ◀◀

### コマンド・パラメーター

なし

### 例

次の例は、コマンド **GET CONTACTGROUPS** を発行した出力結果です。

| Name    | Description                          |
|---------|--------------------------------------|
| support | Foo Widgets broadloom support unit   |
| service | Foo Widgets service and support unit |

---

### GET CONTACTS

ローカル・システムで定義された連絡先のリストを取得します。連絡先とは、スケジューラーおよびヘルス・モニターなどのモニター・プロセスが、通知やメッセージを送信する先のユーザーです。

連絡先を作成するには、**ADD CONTACT** コマンドを使用します。

#### 許可

なし

#### 必要な接続

なし

#### コマンド構文

▶▶ GET CONTACTS ◀◀

#### コマンド・パラメーター

なし

#### 例

GET CONTACTS

| Name   | Type   | Address                  | Max Page Length | Description  |
|--------|--------|--------------------------|-----------------|--------------|
| joe    | e-mail | joe@somewhere.com        | -               | -            |
| joline | e-mail | joline@somewhereelse.com | -               | -            |
| john   | page   | john@relay.org           | 50              | Support 24x7 |



## GET DATABASE CONFIGURATION

特定のデータベース構成ファイル内にある個々の項目の値を取得します。

### 有効範囲

このコマンドは、それが実行されたデータベース・パーティションに関する情報だけを返します。

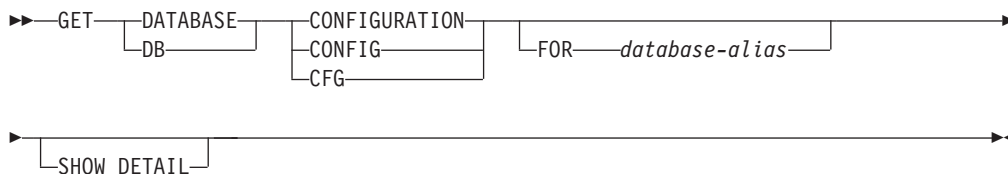
### 許可

なし

### 必要な接続

インスタンス。**SHOW DETAIL** 節を使用するとき、明示的なアタッチは必要ありませんが、データベースへの接続は必要です。データベースがリモートとして示されている場合、リモート・ノードへのインスタンス・アタッチはコマンドの持続期間の間、ずっと確立されたままになります。

### コマンド構文



### コマンド・パラメーター

#### FOR *database-alias*

構成を表示したいデータベースの別名を指定します。データベースへの接続が既に存在する場合、別名を指定する必要はありません。

#### SHOW DETAIL

データベース構成パラメーターの現行値、および次回データベースをアクティブ化する際のパラメーター値についての詳細情報を表示します。このオプションによって、構成パラメーターを動的に変更した結果を見ることができます。

これは、CLPPlus インターフェースで操作する場合のデフォルトの節です。**SHOW DETAIL** は、CLPPlus プロセッサの使用時に呼び出す必要はありません。

**SHOW DETAIL** オプションが指定されていない場合、このコマンドはディスクに保存されている値のみを返しますが、この値はメモリー内の現行値と異なる可能性があります。

### 例

#### 注:

1. プラットフォームが異なると、プラットフォーム固有のパラメーターを反映して、出力の結果に微妙な違いが生じる場合があります。

## GET DATABASE CONFIGURATION

2. キーワードが括弧で囲まれているパラメーターは、**UPDATE DATABASE CONFIGURATION** コマンドによって変更できます。
3. キーワードを持たないフィールドは、データベース・マネージャーが保守します。そのようなフィールドを更新することはできません。

**GET DATABASE CONFIGURATION** コマンドを UNIX オペレーティング・システムの DB2 pureScale 環境外で実行した場合の出力例を以下に示します。

```
Database Configuration for Database sample

Database configuration release level          = 0x0f00
Database release level                      = 0x0f00

Database territory                         = US
Database code page                         = 1208
Database code set                          = UTF-8
Database country/region code              = 1
Database collating sequence               = IDENTITY
Alternate collating sequence              (ALT_COLLATE) =
Number compatibility                       = OFF
Varchar2 compatibility                   = OFF
Date compatibility                        = OFF
Database page size                        = 4096

Dynamic SQL Query management              (DYN_QUERY_MGMT) = DISABLE

Statement concentrator                    (STMT_CONC) = OFF

Discovery support for this database        (DISCOVER_DB) = ENABLE

Restrict access                           = NO
Default query optimization class          (DFT_QUERYOPT) = 5
Degree of parallelism                     (DFT_DEGREE) = 1
Continue upon arithmetic exceptions      (DFT_SQLMATHWARN) = NO
Default refresh age                       (DFT_REFRESH_AGE) = 0
Default maintained table types for opt    (DFT_MTTB_TYPES) = SYSTEM
Number of frequent values retained        (NUM_FREQVALUES) = 10
Number of quantiles retained              (NUM_QUANTILES) = 20

Decimal floating point rounding mode      (DECFLT_ROUNDING) = ROUND_HALF_EVEN

Backup pending                            = NO

All committed transactions have been written to disk = YES
Rollforward pending                       = NO
Restore pending                           = NO

Multi-page file allocation enabled        = YES

Log retain for recovery status            = NO
User exit for logging status              = NO

Self tuning memory                        (SELF_TUNING_MEM) = OFF
Size of database shared memory (4KB)     (DATABASE_MEMORY) = AUTOMATIC(40944)
Database memory threshold                 (DB_MEM_THRESH) = 10
Max storage for lock list (4KB)          (LOCKLIST) = 4096
Percent. of lock lists per application    (MAXLOCKS) = 10
Package cache size (4KB)                 (PCKCACHESZ) = (MAXAPPLS*8)
Sort heap thres for shared sorts (4KB)   (SHEAPTHRES_SHR) = 5000
Sort list heap (4KB)                     (SORTHEAP) = 256

Database heap (4KB)                      (DBHEAP) = AUTOMATIC(1200)
Catalog cache size (4KB)                 (CATALOGCACHE_SZ) = (MAXAPPLS*5)
Log buffer size (4KB)                    (LOGBUFSZ) = 256
Utilities heap size (4KB)                 (UTIL_HEAP_SZ) = 5000
Buffer pool size (pages)                 (BUFFPAGE) = 1000
SQL statement heap (4KB)                  (STMTHEAP) = AUTOMATIC(8192)
Default application heap (4KB)            (APPLHEAPSZ) = AUTOMATIC(256)
Application Memory Size (4KB)             (APPL_MEMORY) = AUTOMATIC(40000)
Statistics heap size (4KB)                (STAT_HEAP_SZ) = AUTOMATIC(4384)

Interval for checking deadlock (ms)       (DLCHKTIME) = 10000
Lock timeout (sec)                       (LOCKTIMEOUT) = -1

Changed pages threshold                   (CHNGPGS_THRESH) = 60
Number of asynchronous page cleaners      (NUM_IOCLEANERS) = AUTOMATIC(1)
Number of I/O servers                     (NUM_IOSERVERS) = AUTOMATIC(3)
Index sort flag                           (INDEXSORT) = YES
Sequential detect flag                    (SEQDETECT) = YES
Default prefetch size (pages)             (DFT_PREFETCH_SZ) = AUTOMATIC

Track modified pages                      (TRACKMOD) = NO

Default number of containers              = 1
Default tablespace extentsize (pages)    (DFT_EXTENT_SZ) = 32
```

## GET DATABASE CONFIGURATION

```

Max number of active applications          (MAXAPPLS) = AUTOMATIC(40)
Average number of active applications     (AVG_APPLS) = AUTOMATIC(1)
Max DB files open per application         (MAXFILOP) = 61440

Log file size (4KB)                      (LOGFILSIZ) = 1000
Number of primary log files               (LOGPRIMARY) = 3
Number of secondary log files            (LOGSECOND) = 2
Changed path to log files                 (NEWLOGPATH) =
Path to log files                         = /home/db2inst1/db2inst1/SAMPLE/NODE0000/LOGSTREAM0000/
Overflow log path                         (OVERFLOWLOGPATH) =
Mirror log path                           (MIRRORLOGPATH) =
First active log file                     =
Block log on disk full                    (BLK_LOG_DSK_FUL) = NO
Block non logged operations               (BLOCKNONLOGGED) = NO
Percent max primary log space by transaction (MAX_LOG) = 0
Num. of active log files for 1 active UOW(NUM_LOG_SPAN) = 0

Group commit count                       (MINCOMMIT) = 1
Percent log file reclaimed before soft ckcpt (SOFTMAX) = 100

HADR database role                       = STANDARD
HADR local host name                     (HADR_LOCAL_HOST) =
HADR local service name                   (HADR_LOCAL_SVC) =
HADR remote host name                     (HADR_REMOTE_HOST) =
HADR remote service name                  (HADR_REMOTE_SVC) =
HADR instance name of remote server       (HADR_REMOTE_INST) =
HADR timeout value                        (HADR_TIMEOUT) = 120
HADR target list                          (HADR_TARGET_LIST) =
HADR log write synchronization mode       (HADR_SYNCMODE) = NEARSYNC
HADR peer window duration (seconds)       (HADR_PEER_WINDOW) = 0
HADR spool log data limit (4KB)           (HADR_SPOOL_LIMIT) = 0
HADR log replay delay (seconds)           (HADR_REPLAY_DELAY) = 0
First log archive method                  (LOGARCHMETH1) = OFF
Archive compression for logarchmeth1      (LOGARCHCOMPR1) = OFF
Options for logarchmeth1                  (LOGARCHOPT1) =
Second log archive method                  (LOGARCHMETH2) = OFF
Archive compression for logarchmeth2      (LOGARCHCOMPR2) = OFF
Options for logarchmeth2                  (LOGARCHOPT2) =
Failover log archive path                 (FAILARCHPATH) =
Number of log archive retries on error     (NUMARCHRETRY) = 5
Log archive retry Delay (secs)             (ARCHRETRYDELAY) = 20
Vendor options                            (VENDOROPT) =

Auto restart enabled                      (AUTORESTART) = ON
Index re-creation time and redo index build (INDEXREC) = SYSTEM (RESTART)
Log pages during index build              (LOGINDEXBUILD) = OFF
Default number of loadrec sessions        (DFT_LOADREC_SES) = 1
Number of database backups to retain      (NUM_DB_BACKUPS) = 12
Recovery history retention (days)        (REC_HIS_RETENTN) = 366
Auto deletion of recovery objects         (AUTO_DEL_REC_OBJ) = OFF

TSM management class                     (TSM_MGMTCLASS) =
TSM node name                             (TSM_NODENAME) =
TSM owner                                  (TSM_OWNER) =
TSM password                              (TSM_PASSWORD) =

Automatic maintenance                     (AUTO_MAINT) = ON
  Automatic database backup                (AUTO_DB_BACKUP) = OFF
  Automatic table maintenance              (AUTO_TBL_MAINT) = ON
  Automatic runstats                       (AUTO_RUNSTATS) = ON
    Automatic statement statistics         (AUTO_STMT_STATS) = ON
    Automatic stat view statistics         (AUTO_STATS_VIEWS) = OFF
  Automatic statistics profiling           (AUTO_STATS_PROF) = OFF
    Automatic profile updates              (AUTO_PROF_UPD) = OFF
  Automatic reorganization                 (AUTO_REORG) = OFF

Auto-Revalidation                         (AUTO_REVAL) = DEFERRED
Currently Committed                       (CUR_COMMIT) = ON
CHAR output with DECIMAL input            (DEC_TO_CHAR_FMT) = NEW
Enable XML Character operations            (ENABLE_XMLCHAR) = YES
WLM Collection Interval (minutes)         (WLM_COLLECT_INT) = 0
Monitor Collect Settings
Request metrics                           (MON_REQ_METRICS) = BASE
Activity metrics                          (MON_ACT_METRICS) = BASE
Object metrics                            (MON_OBJ_METRICS) = BASE
Unit of work events                       (MON_UOW_DATA) = NONE
  UOW events with package list             (MON_UOW_PKGLIST) = OFF
  UOW events with executable list         (MON_UOW_EXECLIST) = OFF
Lock timeout events                       (MON_LOCKTIMEOUT) = NONE
Deadlock events                           (MON_DEADLOCK) = WITHOUT_HIST
Lock wait events                          (MON_LOCKWAIT) = NONE
Lock wait event threshold                  (MON_LW_THRESH) = 5000000
Number of package list entries             (MON_PKGLIST_SZ) = 32
Lock event notification level              (MON_LCK_MSG_LVL) = 1

SMTP Server                               (SMTP_SERVER) =
SQL conditional compilation flags          (SQL_CCFLAGS) =
Section actuals setting                    (SECTION_ACTUALS) = NONE
Connect procedure                         (CONNECT_PROC) =
Adjust temporal SYSTEM_TIME period (SYSTIME_PERIOD_ADJ) = NO
Log DDL Statements                        (LOG_DDL_STMTS) = NO

```

## GET DATABASE CONFIGURATION

```
Log Application Information          (LOG_APPL_INFO) = NO
Default Data Capture on new Schemas (DFT_SCHEMAS_DCC) = NO
Database is in write suspend state  = NO
```

**GET DATABASE CONFIGURATION** コマンドを Linux オペレーティング・システムの DB2 pureScale 環境で実行した場合の出力例を以下に示します。

Database Configuration for Database sample

```
Database configuration release level      = 0x0e00
Database release level                   = 0x0e00

Database territory                       = US
Database code page                       = 1208
Database code set                       = UTF-8
Database country/region code            = 1
Database collating sequence              = IDENTITY
Alternate collating sequence              (ALT_COLLATE) =
Number compatibility                     = OFF
Varchar2 compatibility                  = OFF
Date compatibility                       = OFF
Database page size                       = 4096

Statement concentrator                   (STMT_CONC) = OFF

Discovery support for this database       (DISCOVER_DB) = ENABLE

Restrict access                          = NO
Default query optimization class         (DFT_QUERYOPT) = 5
Degree of parallelism                    (DFT_DEGREE) = 1
Continue upon arithmetic exceptions      (DFT_SQLMATHWARN) = NO
Default refresh age                      (DFT_REFRESH_AGE) = 0
Default maintained table types for opt   (DFT_MTTB_TYPES) = SYSTEM
Number of frequent values retained       (NUM_FREQVALUES) = 10
Number of quantiles retained             (NUM_QUANTILES) = 20

Decimal floating point rounding mode     (DECFLT_ROUNDING) = ROUND_HALF_EVEN

Backup pending                           = NO

All committed transactions have been written to disk = YES
Rollforward pending                      = NO
Restore pending                           = NO

Multi-page file allocation enabled       = YES

Log retain for recovery status            = NO
User exit for logging status              = NO

Self tuning memory                       (SELF_TUNING_MEM) = ON
Size of database shared memory (4KB)     (DATABASE_MEMORY) = AUTOMATIC(60032)
Database memory threshold                (DB_MEM_THRESH) = 10
Max storage for lock list (4KB)          (LOCKLIST) = AUTOMATIC(6575)
Percent. of lock lists per application   (MAXLOCKS) = AUTOMATIC(60)
Package cache size (4KB)                 (PCKCACHESZ) = AUTOMATIC(1533)
Sort heap thres for shared sorts (4KB)   (SHEAPTHRES_SHR) = AUTOMATIC(21919)
Sort list heap (4KB)                     (SORTHEAP) = AUTOMATIC(1095)

Database heap (4KB)                      (DBHEAP) = AUTOMATIC(2462)
Catalog cache size (4KB)                 (CATALOGCACHE_SZ) = 300
Log buffer size (4KB)                   (LOGBUFSZ) = 256
Utilities heap size (4KB)                (UTIL_HEAP_SZ) = 52763
Buffer pool size (pages)                 (BUFFPAGE) = 1000
SQL statement heap (4KB)                 (STMTHEAP) = AUTOMATIC(4096)
Default application heap (4KB)           (APPLHEAPSZ) = AUTOMATIC(256)
Application Memory Size (4KB)            (APPL_MEMORY) = AUTOMATIC(40000)
Statistics heap size (4KB)               (STAT_HEAP_SZ) = AUTOMATIC(4384)

Interval for checking deadlock (ms)      (DLCHKTIME) = 10000
Lock timeout (sec)                       (LOCKTIMEOUT) = -1

Changed pages threshold                   (CHNGPGS_THRESH) = 80
Number of asynchronous page cleaners     (NUM_IOCLEANERS) = AUTOMATIC(1)
Number of I/O servers                    (NUM_IOSERVERS) = AUTOMATIC(4)
Index sort flag                           (INDEXSORT) = YES
Sequential detect flag                    (SEQDETECT) = YES
Default prefetch size (pages)            (DFT_PREFETCH_SZ) = AUTOMATIC
```

## GET DATABASE CONFIGURATION

```

Track modified pages                (TRACKMOD) = NO

Default number of containers        = 1
Default tablespace extentsize (pages) (DFT_EXTENT_SZ) = 32

Max number of active applications    (MAXAPPLS) = AUTOMATIC(40)
Average number of active applications (AVG_APPLS) = AUTOMATIC(1)
Max DB files open per application    (MAXFILOP) = 61440

Log file size (4KB)                 (LOGFILSIZ) = 1024
Number of primary log files          (LOGPRIMARY) = 13
Number of secondary log files        (LOGSECOND) = 4
Changed path to log files            (NEWLOGPATH) =
Path to log files                    = /home/db2inst1/db2inst1/SAMPLE/NODE0000/LOGSTREAM0000/
Overflow log path                    (OVERFLOWLOGPATH) =
Mirror log path                      (MIRRORLOGPATH) =
First active log file                =
Block log on disk full               (BLK_LOG_DSK_FUL) = NO
Block non logged operations          (BLOCKNONLOGGED) = NO
Percent max primary log space by transaction (MAX_LOG) = 0
Num. of active log files for 1 active UOW(NUM_LOG_SPAN) = 0

Group commit count                  (MINCOMMIT) = 1
Percent log file reclaimed before soft ckcpt (SOFTMAX) = 520
HADR database role                  = STANDARD
HADR local host name                (HADR_LOCAL_HOST) =
HADR local service name             (HADR_LOCAL_SVC) =
HADR remote host name               (HADR_REMOTE_HOST) =
HADR remote service name           (HADR_REMOTE_SVC) =
HADR instance name of remote server (HADR_REMOTE_INST) =
HADR timeout value                  (HADR_TIMEOUT) = 120
HADR log write synchronization mode (HADR_SYNCMODE) = NEARSYNC
HADR peer window duration (seconds) (HADR_PEER_WINDOW) = 0
HADR spool log data limit (4KB)     (HADR_SPOOL_LIMIT) = 0
HADR log replay delay (seconds)     (HADR_REPLAY_DELAY) = 0
First log archive method            (LOGARCHMETH1) = OFF
Archive compression for logarchmeth1 (LOGARCHCOMPR1) = OFF
Options for logarchmeth1            (LOGARCHOPT1) =
Second log archive method           (LOGARCHMETH2) = OFF
Archive compression for logarchmeth2 (LOGARCHCOMPR2) = OFF
Options for logarchmeth2            (LOGARCHOPT2) =
Failover log archive path           (FAILARCHPATH) =
Number of log archive retries on error (NUMARCHRETRY) = 5
Log archive retry Delay (secs)      (ARCHRETRYDELAY) = 20
Vendor options                      (VENDOROPT) =

Auto restart enabled                (AUTORESTART) = ON
Index re-creation time and redo index build (INDEXREC) = SYSTEM (RESTART)
Log pages during index build        (LOGINDEXBUILD) = OFF
Default number of loadrec sessions  (DFT_LOADREC_SES) = 1
Number of database backups to retain (NUM_DB_BACKUPS) = 12
Recovery history retention (days)   (REC_HIS_RETENTN) = 366
Auto deletion of recovery objects   (AUTO_DEL_REC_OBJ) = OFF

TSM management class                (TSM_MGMTCLASS) =
TSM node name                       (TSM_NODENAME) =
TSM owner                           (TSM_OWNER) =
TSM password                        (TSM_PASSWORD) =

Automatic maintenance               (AUTO_MAINT) = ON
Automatic database backup           (AUTO_DB_BACKUP) = OFF
Automatic table maintenance         (AUTO_TBL_MAINT) = ON
Automatic runstats                  (AUTO_RUNSTATS) = ON
Automatic statement statistics      (AUTO_STMT_STATS) = ON
Automatic statistics profiling      (AUTO_STATS_PROF) = OFF
Automatic profile updates           (AUTO_PROF_UPD) = OFF
Automatic reorganization            (AUTO_REORG) = OFF

Auto-Revalidation                   (AUTO_REVAL) = DEFERRED
CF Resource Configuration:
CF database memory size (4KB)       (CF_DB_MEM_SZ) = AUTOMATIC(127488)
Group buffer pool size (4KB)       (CF_GBP_SZ) = AUTOMATIC(80896)
Global lock memory size (4KB)      (CF_LOCK_SZ) = AUTOMATIC(19200)
Shared communication area size (4KB) (CF_SCA_SZ) = AUTOMATIC(26368)

Catchup target for secondary CF (mins)(CF_CATCHUP_TRGT) = AUTOMATIC(15)

Currently Committed                 (CUR_COMMIT) = ON
CHAR output with DECIMAL input      (DEC_TO_CHAR_FMT) = NEW

```

## GET DATABASE CONFIGURATION

```

Enable XML Character operations      (ENABLE_XMLCHAR) = YES
WLM Collection Interval (minutes)   (WLM_COLLECT_INT) = 0
Monitor Collect Settings
Request metrics                     (MON_REQ_METRICS) = BASE
Activity metrics                    (MON_ACT_METRICS) = BASE
Object metrics                      (MON_OBJ_METRICS) = BASE
Unit of work events                 (MON_UOW_DATA) = NONE
Lock timeout events                 (MON_LOCKTIMEOUT) = NONE
Deadlock events                     (MON_DEADLOCK) = WITHOUT_HIST
Lock wait events                    (MON_LOCKWAIT) = NONE
Lock wait event threshold           (MON_LW_THRESH) = 5000000
Number of package list entries      (MON_PKGLIST_SZ) = 32
Lock event notification level       (MON_LCK_MSG_LVL) = 1

SMTP Server                         (SMTP_SERVER) =
SQL conditional compilation flags    (SQL_CCFLAGS) =
Section actuals setting             (SECTION_ACTUALS) = NONE

Database is in write suspend state  = NO

```

DB2 pureScale 環境で **SHOW DETAIL** オプションを指定した場合のコマンドの出力例を以下に示します。Delayed Value 列の値は、インスタンスを次回開始する際に適用される値です。

Database Configuration for Database sample

| Description                                          | Parameter         | Current Value       | Delayed Value     |
|------------------------------------------------------|-------------------|---------------------|-------------------|
| Database configuration release level                 |                   | = 0x0e00            |                   |
| Database release level                               |                   | = 0x0e00            |                   |
| Database territory                                   |                   | = US                |                   |
| Database code page                                   |                   | = 1208              |                   |
| Database code set                                    |                   | = UTF-8             |                   |
| Database country/region code                         |                   | = 1                 |                   |
| Database collating sequence                          |                   | = IDENTITY          | IDENTITY          |
| Alternate collating sequence                         | (ALT_COLLATE)     | =                   |                   |
| Number compatibility                                 |                   | = OFF               |                   |
| Varchar2 compatibility                               |                   | = OFF               |                   |
| Date compatibility                                   |                   | = OFF               |                   |
| Database page size                                   |                   | = 4096              | 4096              |
| Statement concentrator                               | (STMT_CONC)       | = OFF               | OFF               |
| Discovery support for this database                  | (DISCOVER_DB)     | = ENABLE            | ENABLE            |
| Restrict access                                      |                   | = NO                |                   |
| Default query optimization class                     | (DFT_QUERYOPT)    | = 5                 | 5                 |
| Degree of parallelism                                | (DFT_DEGREE)      | = 1                 | 1                 |
| Continue upon arithmetic exceptions                  | (DFT_SQLMATHWARN) | = NO                | NO                |
| Default refresh age                                  | (DFT_REFRESH_AGE) | = 0                 | 0                 |
| Default maintained table types for opt               | (DFT_MTTB_TYPES)  | = SYSTEM            | SYSTEM            |
| Number of frequent values retained                   | (NUM_FREQVALUES)  | = 10                | 10                |
| Number of quantiles retained                         | (NUM_QUANTILES)   | = 20                | 20                |
| Decimal floating point rounding mode                 | (DECFLT_ROUNDING) | = ROUND_HALF_EVEN   | ROUND_HALF_EVEN   |
| Backup pending                                       |                   | = NO                |                   |
| All committed transactions have been written to disk |                   | = YES               |                   |
| Rollforward pending                                  |                   | = NO                |                   |
| Restore pending                                      |                   | = NO                |                   |
| Multi-page file allocation enabled                   |                   | = YES               |                   |
| Log retain for recovery status                       |                   | = NO                |                   |
| User exit for logging status                         |                   | = NO                |                   |
| Self tuning memory                                   | (SELF_TUNING_MEM) | = ON (Active)       | ON                |
| Size of database shared memory (4KB)                 | (DATABASE_MEMORY) | = AUTOMATIC(345584) | AUTOMATIC(345584) |
| Database memory threshold                            | (DB_MEM_THRESH)   | = 10                | 10                |
| Max storage for lock list (4KB)                      | (LOCKLIST)        | = AUTOMATIC(6575)   | AUTOMATIC(6575)   |
| Percent. of lock lists per application               | (MAXLOCKS)        | = AUTOMATIC(60)     | AUTOMATIC(60)     |
| Package cache size (4KB)                             | (PCKCACHESZ)      | = AUTOMATIC(1533)   | AUTOMATIC(1533)   |
| Sort heap thres for shared sorts (4KB)               | (SHEAPTHRES_SHR)  | = AUTOMATIC(21919)  | AUTOMATIC(21919)  |
| Sort list heap (4KB)                                 | (SORTHEAP)        | = AUTOMATIC(1095)   | AUTOMATIC(1095)   |
| Database heap (4KB)                                  | (DBHEAP)          | = AUTOMATIC(2462)   | AUTOMATIC(2462)   |
| Catalog cache size (4KB)                             | (CATALOGCACHE_SZ) | = 300               | 300               |
| Log buffer size (4KB)                                | (LOGBUFSZ)        | = 256               | 256               |
| Utilities heap size (4KB)                            | (UTIL_HEAP_SZ)    | = 52763             | 52763             |
| Buffer pool size (pages)                             | (BUFFPAGE)        | = 1000              | 1000              |
| SQL statement heap (4KB)                             | (STMTHEAP)        | = AUTOMATIC(6402)   | AUTOMATIC(6402)   |
| Default application heap (4KB)                       | (APPLHEAPSZ)      | = AUTOMATIC(256)    | AUTOMATIC(256)    |
| Application Memory Size (4KB)                        | (APPL_MEMORY)     | = AUTOMATIC(40016)  | AUTOMATIC(40000)  |
| Statistics heap size (4KB)                           | (STAT_HEAP_SZ)    | = AUTOMATIC(4384)   | AUTOMATIC(4384)   |
| Interval for checking deadlock (ms)                  | (DLCHKTIME)       | = 10000             | 10000             |
| Lock timeout (sec)                                   | (LOCKTIMEOUT)     | = -1                | -1                |
| Changed pages threshold                              | (CHNGPGS_THRESH)  | = 80                | 80                |
| Number of asynchronous page cleaners                 | (NUM_IOCLEANERS)  | = AUTOMATIC(3)      | AUTOMATIC(3)      |
| Number of I/O servers                                | (NUM_IOSERVERS)   | = AUTOMATIC(6)      | AUTOMATIC(6)      |
| Index sort flag                                      | (INDEXSORT)       | = YES               | YES               |
| Sequential detect flag                               | (SEQDETECT)       | = YES               | YES               |
| Default prefetch size (pages)                        | (DFT_PREFETCH_SZ) | = AUTOMATIC         | AUTOMATIC         |

# GET DATABASE CONFIGURATION

```

Track modified pages                (TRACKMOD) = NO                NO
Default number of containers        = 1                                1
Default tablespace extentsize (pages) (DFT_EXTENT_SZ) = 32                32
Max number of active applications   (MAXAPPLS) = AUTOMATIC(40)     AUTOMATIC(40)
Average number of active applications (AVG_APPLS) = AUTOMATIC(1)     AUTOMATIC(1)
Max DB files open per application   (MAXFILOP) = 61440             61440

Log file size (4KB)                 (LOGFILSIZ) = 1024             1024
Number of primary log files         (LOGPRIMARY) = 13              13
Number of secondary log files       (LOGSECOND) = 4                4
Changed path to log files           (NEWLOGPATH) =                  /home/db2inst1/db2inst1/SAMPLE/NODE0000/LOGSTREAM0000/
Path to log files                    = /home/db2inst1/db2inst1/SAMPLE/NODE0000/LOGSTREAM0000/
Overflow log path                    (OVERFLOWLOGPATH) =
Mirror log path                      (MIRRORLOGPATH) =
First active log file                =
Block log on disk full              (BLK_LOG_DSK_FUL) = NO        NO
Block non logged operations          (BLOCKNONLOGGED) = NO        NO
Percent max primary log space by transaction (MAX_LOG) = 0                0
Num. of active log files for 1 active UOW(NUM_LOG_SPAN) = 0                0

Group commit count                  (MINCOMMIT) = 1                1
Percent log file reclaimed before soft ckcpt (SOFTMAX) = 520            520
HADR database role                  = STANDARD                      STANDARD
HADR local host name                (HADR_LOCAL_HOST) =
HADR local service name             (HADR_LOCAL_SVC) =
HADR remote host name               (HADR_REMOTE_HOST) =
HADR remote service name            (HADR_REMOTE_SVC) =
HADR instance name of remote server (HADR_REMOTE_INST) =
HADR timeout value                  (HADR_TIMEOUT) = 120          120
HADR log write synchronization mode (HADR_SYNCMODE) = NEARSYNC    NEARSYNC
HADR peer window duration (seconds) (HADR_PEER_WINDOW) = 0                0
HADR spool log data limit (4KB)     (HADR_SPOOL_LIMIT) = 0        0
HADR log replay delay (seconds)     (HADR_REPLAY_DELAY) = 0      0
First log archive method            (LOGARCHMETH1) = OFF          OFF
Archive compression for logarchmeth1 (LOGARCHCOMPR1) = OFF        OFF
Options for logarchmeth1            (LOGARCHOPT1) =
Second log archive method           (LOGARCHMETH2) = OFF          OFF
Archive compression for logarchmeth2 (LOGARCHCOMPR2) = OFF        OFF
Options for logarchmeth2            (LOGARCHOPT2) =
Failover log archive path           (FAILARCHPATH) =
Number of log archive retries on error (NUMARCHRETRY) = 5                5
Log archive retry Delay (secs)      (ARCHRETRYDELAY) = 20         20
Vendor options                      (VENDOROPT) =

Auto restart enabled                (AUTORESTART) = ON            ON
Index re-creation time and redo index build (INDEXREC) = SYSTEM (RESTART) SYSTEM (RESTART)
Log pages during index build        (LOGINDEXBUILD) = OFF        OFF
Default number of loadrec sessions  (DFT_LOADREC_SES) = 1         1
Number of database backups to retain (NUM_DB_BACKUPS) = 12        12
Recovery history retention (days)   (REC_HIS_RETENTN) = 366      366
Auto deletion of recovery objects    (AUTO_DEL_REC_OBJ) = OFF     OFF

TSM management class                (TSM_MGMTCLASS) =
TSM node name                      (TSM_NODENAME) =
TSM owner                          (TSM_OWNER) =
TSM password                        (TSM_PASSWORD) =

Automatic maintenance               (AUTO_MAINT) = ON            ON
Automatic database backup            (AUTO_DB_BACKUP) = OFF       OFF
Automatic table maintenance         (AUTO_TBL_MAINT) = ON        ON
Automatic runstats                  (AUTO_RUNSTATS) = ON        ON
Automatic statement statistics       (AUTO_STMT_STATS) = ON       ON
Automatic statistics profiling       (AUTO_STATS_PROF) = OFF     OFF
Automatic profile updates            (AUTO_PROF_UPD) = OFF       OFF
Automatic reorganization            (AUTO_REORG) = OFF          OFF

Auto-Revalidation                   (AUTO_REVAL) = DEFERRED      DEFERRED
CF Resource Configuration:
CF database memory size (4KB)       (CF_DB_MEM_SZ) = AUTOMATIC(114688) AUTOMATIC(114688)
Group buffer pool size (4KB)        (CF_GBP_SZ) = AUTOMATIC(72704)  AUTOMATIC(72704)
Global lock memory size (4KB)       (CF_LOCK_SZ) = AUTOMATIC(17408)  AUTOMATIC(17408)
Shared communication area size (4KB) (CF_SCA_SZ) = AUTOMATIC(23552)  AUTOMATIC(23552)

Catchup target for secondary CF (mins)(CF_CATCHUP_TRGT) = AUTOMATIC(15)  AUTOMATIC(15)

Currently Committed                  (CUR_COMMIT) = ON            ON
CHAR output with DECIMAL input      (DEC_TO_CHAR_FMT) = NEW      NEW
Enable XML Character operations      (ENABLE_XMLCHAR) = YES      YES
WLM Collection Interval (minutes)    (WLM_COLLECT_INT) = 0        0

Monitor Collect Settings
Request metrics                      (MON_REQ_METRICS) = BASE     BASE
Activity metrics                    (MON_ACT_METRICS) = BASE     BASE
Object metrics                      (MON_OBJ_METRICS) = BASE     BASE
Unit of work events                 (MON_UOW_DATA) = NONE       NONE
Lock timeout events                 (MON_LOCKTIMEOUT) = NONE     NONE
Deadlock events                    (MON_DEADLOCK) = WITHOUT_HIST WITHOUT_HIST
Lock wait events                    (MON_LOCKWAIT) = NONE       NONE
Lock wait event threshold           (MON_LW_THRESH) = 5000000    5000000

Number of package list entries       (MON_PGCLIST_SZ) = 32        32
Lock event notification level        (MON_LCK_MSG_LVL) = 1        1

SMTP Server                          (SMTP_SERVER) =
SQL conditional compilation flags    (SQL_CCFLAGS) =

Section actuals setting              (SECTION_ACTUALS) = NONE     NONE

Database is in write suspend state   = NO

```



## GET DATABASE CONFIGURATION

GET DATABASE CONFIGURATION コマンドを Windows オペレーティング・システムで実行した後の出力例を以下に示します。

```
Database Configuration for Database

Database configuration release level          = 0x0d00
Database release level                      = 0x0d00

Database territory                          = US
Database code page                          = 1208
Database code set                           = UTF-8
Database country/region code                = 1
Database collating sequence                 = IDENTITY
Alternate collating sequence                (ALT_COLLATE) =
Number compatibility                        = OFF
Varchar2 compatibility                     = OFF
Date compatibility                          = OFF

Database page size                          = 4096

Dynamic SQL Query management                (DYN_QUERY_MGMT) = DISABLE

Statement concentrator                      (STMT_CONC) = OFF

Discovery support for this database          (DISCOVER_DB) = ENABLE

Restrict access                             = NO
Default query optimization class            (DFT_QUERYOPT) = 5
Degree of parallelism                       (DFT_DEGREE) = 1
Continue upon arithmetic exceptions          (DFT_SQLMATHWARN) = NO
Default refresh age                         (DFT_REFRESH_AGE) = 0
Default maintained table types for opt      (DFT_MTTB_TYPES) = SYSTEM
Number of frequent values retained          (NUM_FREQVALUES) = 10
Number of quantiles retained                (NUM_QUANTILES) = 20

Decimal floating point rounding mode        (DECFLT_ROUNDING) = ROUND_HALF_EVEN

Backup pending                              = NO

All committed transactions have been written to disk = YES
Rollforward pending                         = NO
Restore pending                             = NO

Multi-page file allocation enabled           = YES

Log retain for recovery status               = NO
User exit for logging status                = NO

Self tuning memory                          (SELF_TUNING_MEM) = ON
Size of database shared memory (4KB)        (DATABASE_MEMORY) = AUTOMATIC(60464)
Database memory threshold                  (DB_MEM_THRESH) = 10
Max storage for lock list (4KB)             (LOCKLIST) = AUTOMATIC(6200)
Percent. of lock lists per application      (MAXLOCKS) = AUTOMATIC(60)
Package cache size (4KB)                   (PCKCACHESZ) = AUTOMATIC(1533)
Sort heap thres for shared sorts (4KB)      (SHEAPTHRES_SHR) = AUTOMATIC(6728)
Sort list heap (4KB)                       (SORTHEAP) = AUTOMATIC(336)

Database heap (4KB)                        (DBHEAP) = AUTOMATIC(2283)
Catalog cache size (4KB)                   (CATALOGCACHE_SZ) = 300
Log buffer size (4KB)                      (LOGBUFSZ) = 256
Utilities heap size (4KB)                  (UTIL_HEAP_SZ) = 5115
Buffer pool size (pages)                   (BUFFPAGE) = 1000
SQL statement heap (4KB)                   (STMTHEAP) = AUTOMATIC(4096)
Default application heap (4KB)             (APPLHEAPSZ) = AUTOMATIC(256)
Application Memory Size (4KB)              (APPL_MEMORY) = AUTOMATIC(40000)
Statistics heap size (4KB)                 (STAT_HEAP_SZ) = AUTOMATIC(4384)
```



## GET DATABASE CONFIGURATION

```

Interval for checking deadlock (ms)          (DLCHKTIME) = 10000
Lock timeout (sec)                          (LOCKTIMEOUT) = -1

Changed pages threshold                     (CHNGPGS_THRESH) = 80
Number of asynchronous page cleaners        (NUM_IOCLEANERS) = AUTOMATIC(3)
Number of I/O servers                       (NUM_IOSERVERS) = AUTOMATIC(3)
Index sort flag                             (INDEXSORT) = YES
Sequential detect flag                      (SEQDETECT) = YES
Default prefetch size (pages)              (DFT_PREFETCH_SZ) = AUTOMATIC

Track modified pages                        (TRACKMOD) = OFF

Default number of containers                = 1
Default tablespace extentsize (pages)      (DFT_EXTENT_SZ) = 32

Max number of active applications          (MAXAPPLS) = AUTOMATIC(40)
Average number of active applications      (AVG_APPLS) = AUTOMATIC(1)
Max DB files open per application          (MAXFILOP) = 61440

Log file size (4KB)                        (LOGFILSIZ) = 1024
Number of primary log files                 (LOGPRIMARY) = 13
Number of secondary log files              (LOGSECOND) = 4
Changed path to log files                  (NEWLOGPATH) =
Path to log files                          = D:\DB2\NODE0000\SQL00003\LOGSTREAM0000\
Overflow log path                           (OVERFLOWLOGPATH) =
Mirror log path                             (MIRRORLOGPATH) =
First active log file                      =
Block log on disk full                     (BLK_LOG_DSK_FUL) = NO
Block non logged operations                (BLOCKNONLOGGED) = NO
Percent max primary log space by transaction (MAX_LOG) = 0
Num. of active log files for 1 active UOW (NUM_LOG_SPAN) = 0

Group commit count                         (MINCOMMIT) = 1
Percent log file reclaimed before soft chckpt (SOFTMAX) = 520
Log retain for recovery enabled            (LOGRETAIN) = OFF
User exit for logging enabled              (USEREXIT) = OFF

HADR database role                         = STANDARD
HADR local host name                       (HADR_LOCAL_HOST) =
HADR local service name                    (HADR_LOCAL_SVC) =
HADR remote host name                      (HADR_REMOTE_HOST) =
HADR remote service name                   (HADR_REMOTE_SVC) =
HADR instance name of remote server        (HADR_REMOTE_INST) =
HADR timeout value                         (HADR_TIMEOUT) = 120
HADR log write synchronization mode        (HADR_SYNCMODE) = NEARSYNC
HADR peer window duration (seconds)        (HADR_PEER_WINDOW) = 0
HADR spool log data limit (4KB)            (HADR_SPOOL_LIMIT) = 0
HADR log replay delay (seconds)            (HADR_REPLAY_DELAY) = 0
First log archive method                   (LOGARCHMETH1) = OFF
Archive compression for logarchmeth1      (LOGARCHCOMPR1) = OFF
Options for logarchmeth1                   (LOGARCHOPT1) =
Second log archive method                  (LOGARCHMETH2) = OFF
Archive compression for logarchmeth2      (LOGARCHCOMPR2) = OFF
Options for logarchmeth2                   (LOGARCHOPT2) =
Failover log archive path                  (FAILARCHPATH) =
Number of log archive retries on error     (NUMARCHRETRY) = 5
Log archive retry Delay (secs)             (ARCHRETRYDELAY) = 20
Vendor options                             (VENDOROPT) =

Auto restart enabled                       (AUTORESTART) = ON
Index re-creation time and redo index build (INDEXREC) = SYSTEM (RESTART)
Log pages during index build               (LOGINDEXBUILD) = OFF
Default number of loadrec sessions         (DFT_LOADREC_SES) = 1
Number of database backups to retain      (NUM_DB_BACKUPS) = 12
Recovery history retention (days)         (REC_HIS_RETENTN) = 366
Auto deletion of recovery objects         (AUTO_DEL_REC_OBJ) = OFF

```

## GET DATABASE CONFIGURATION

```

TSM management class          (TSM_MGMTCLASS) =
TSM node name                 (TSM_NODENAME) =
TSM owner                     (TSM_OWNER) =
TSM password                  (TSM_PASSWORD) =

Automatic maintenance        (AUTO_MAINT) = ON
Automatic database backup    (AUTO_DB_BACKUP) = OFF
Automatic table maintenance  (AUTO_TBL_MAINT) = ON
Automatic runstats           (AUTO_RUNSTATS) = ON
Automatic statement statistics (AUTO_STMT_STATS) = ON
Automatic statistics profiling (AUTO_STATS_PROF) = OFF
Automatic profile updates    (AUTO_PROF_UPD) = OFF
Automatic reorganization    (AUTO_REORG) = OFF

Auto-Revalidation            (AUTO_REVAL) = DEFERRED
Currently Committed          (CUR_COMMIT) = ON
CHAR output with DECIMAL input (DEC_TO_CHAR_FMT) = NEW
Enable XML Character operations (ENABLE_XMLCHAR) = YES
WLM Collection Interval     (WLM_COLLECT_INT) = 0
Monitor Collect Settings

Request metrics              (MON_REQ_MATRICS) = BASE
Activity metrics             (MON_ACT_MATRICS) = BASE
Object metrics               (MON_OBJ_MATRICS) = BASE
Unit of work events         (MON_UOW_DATA) = NONE
Lock timeout events         (MON_LOCKTIMEOUT) = NONE
Deadlock events             (MON_DEADLOCK) = WITHOUT_HIST
Lock wait events            (MON_LOCKWAIT) = NONE
Lock wait event threshold   (MON_LW_THRESH) = 5000000

SMTP Server                  (SMTP_SERVER) =

```

**SHOW DETAIL** オプションを Windows オペレーティング・システムで指定した場合のコマンドの出力例の一部を以下に示します。 Delayed Value 列の値は、インスタンスを次回開始する際に適用される値です。

### Database Configuration for Database

| Description                            | Parameter         | Current Value | Delayed Value |
|----------------------------------------|-------------------|---------------|---------------|
| Database configuration release level   |                   | = 0x0d00      |               |
| Database release level                 |                   | = 0x0d00      |               |
| Database territory                     |                   | = US          |               |
| Database code page                     |                   | = 1208        |               |
| Database code set                      |                   | = utf-8       |               |
| Database country/region code           |                   | = 1           |               |
| Database collating sequence            |                   | = IDENTITY    | IDENTITY      |
| Alternate collating sequence           | (ALT_COLLATE)     | =             |               |
| Number compatibility                   |                   | = OFF         |               |
| Varchar2 compatibility                 |                   | = OFF         |               |
| Date compatibility                     |                   | = OFF         |               |
| Database page size                     |                   | = 4096        | 4096          |
| Dynamic SQL Query management           | (DYN_QUERY_MGMT)  | = DISABLE     | DISABLE       |
| Statement concentrator                 | (STMT_CONC)       | = OFF         | OFF           |
| Discovery support for this database    | (DISCOVER_DB)     | = ENABLE      | ENABLE        |
| Restrict access                        |                   | = NO          |               |
| Default query optimization class       | (DFT_QUERYOPT)    | = 5           | 5             |
| Degree of parallelism                  | (DFT_DEGREE)      | = 1           | 1             |
| Continue upon arithmetic exceptions    | (DFT_SQLMATHWARN) | = NO          | NO            |
| Default refresh age                    | (DFT_REFRESH_AGE) | = 0           | 0             |
| Default maintained table types for opt | (DFT_MTTB_TYPES)  | = SYSTEM      | SYSTEM        |
| Number of frequent values retained     | (NUM_FREQVALUES)  | = 10          | 10            |
| Number of quantiles retained           | (NUM_QUANTILES)   | = 20          | 20            |

## GET DATABASE CONFIGURATION

```

Decimal floating point rounding mode (DECFLT_ROUNDING) = ROUND_HALF_EVEN      ROUND_HALF_EVEN

Backup pending                                     = NO

All committed transactions have been written to disk = No
Rollforward pending                               = NO
Restore pending                                   = NO

Multi-page file allocation enabled                 = YES

Log retain for recovery status                     = NO
User exit for logging status                      = NO

Self tuning memory                               (SELF_TUNING_MEM) = OFF              OFF
Size of database shared memory (4KB) (DATABASE_MEMORY) = AUTOMATIC(282400)  AUTOMATIC(282400)
Database memory threshold                       (DB_MEM_THRESH)   = 10              10
Max storage for lock list (4KB)                 (LOCKLIST)        = 4096           4096
Percent. of lock lists per application          (MAXLOCKS)        = 10              10
Package cache size (4KB)                        (PCKCACHESZ)      = (MAXAPPLS*8)   (MAXAPPLS*8)
Sort heap thres for shared sorts (4KB) (SHEAPTHRES_SHR) = 5000            5000
Sort list heap (4KB)                            (SORTHEAP)        = 256             256

Database heap (4KB)                              (DBHEAP)          = AUTOMATIC(1200) AUTOMATIC(1200)
Catalog cache size (4KB) (CATALOGCACHE_SZ) = (MAXAPPLS*5)    (MAXAPPLS*5)
Log buffer size (4KB)                            (LOGBUFSZ)        = 256             256
Utilities heap size (4KB)                        (UTIL_HEAP_SZ)    = 5000            5000
Buffer pool size (pages)                        (BUFFPAGE)        = 200             200
SQL statement heap (4KB)                        (STMHEAP)         = AUTOMATIC(6402) AUTOMATIC(4096)
Default application heap (4KB)                  (APPLHEAPSZ)      = AUTOMATIC(256)  AUTOMATIC(256)
Application Memory Size (4KB)                   (APPL_MEMORY)     = AUTOMATIC(40016) AUTOMATIC(40000)
Statistics heap size (4KB)                      (STAT_HEAP_SZ)    = AUTOMATIC(4384) AUTOMATIC(4384)

Interval for checking deadlock (ms)              (DLCHKTIME)       = 10000           10000
Lock timeout (sec)                              (LOCKTIMEOUT)     = -1              -1

Changed pages threshold                         (CHNGPGS_THRESH) = 60              60
Number of asynchronous page cleaners             (NUM_IOCLEANERS)  = AUTOMATIC(3)    AUTOMATIC(3)
Number of I/O servers                          (NUM_IOSERVERS)   = AUTOMATIC(3)    AUTOMATIC(3)
Index sort flag                                (INDEXSORT)       = YES             YES
Sequential detect flag                         (SEQDETECT)       = YES             YES
Default prefetch size (pages)                  (DFT_PREFETCH_SZ) = AUTOMATIC       AUTOMATIC

Track modified pages                           (TRACKMOD)        = NO              NO

Default number of containers                    = 1                1
Default tablespace extentsize (pages) (DFT_EXTENT_SZ) = 32                32

Max number of active applications               (MAXAPPLS)        = AUTOMATIC(40)   AUTOMATIC(40)
Average number of active applications          (AVG_APPLS)       = AUTOMATIC(1)    AUTOMATIC(1)
Max DB files open per application              (MAXFILOP)        = 61440           61440

Log file size (4KB)                            (LOGFILSIZ)       = 1000            1000
Number of primary log files                    (LOGPRIMARY)      = 3                3
Number of secondary log files                  (LOGSECOND)       = 2                2
Changed path to log files                      (NEWLOGPATH)      =
Path to log files                              = D:¥DB2¥NODE0000  D:¥DB2¥NODE0000
  \SQL00001\LOGSTREAM0000\ \SQL00001\LOGSTREAM0000\

Overflow log path                              (OVERFLOWLOGPATH) =
Mirror log path                                (MIRRORLOGPATH)   =
First active log file                          =
Block log on disk full                        (BLK_LOG_DSK_FUL) = NO              NO
Percent max primary log space by transaction (MAX_LOG)         = 0                0
Num. of active log files for 1 active UOW(NUM_LOG_SPAN) = 0                0
Group commit count                             (MINCOMMIT)       = 1                1
Percent log file reclaimed before soft ckcpt (SOFTMAX) = 100              100
Log retain for recovery enabled                (LOGRETAIN)       = OFF              OFF
User exit for logging enabled                  (USEREXIT)        = OFF              OFF

```

## GET DATABASE CONFIGURATION

```

HADR database role                                = STANDARD          STANDARD
HADR local host name                            (HADR_LOCAL_HOST) =
HADR local service name                        (HADR_LOCAL_SVC) =
HADR remote host name                          (HADR_REMOTE_HOST) =
HADR remote service name                      (HADR_REMOTE_SVC) =
HADR instance name of remote server           (HADR_REMOTE_INST) =
HADR timeout value                             (HADR_TIMEOUT) = 120          120
HADR log write synchronization mode           (HADR_SYNCMODE) = NEARSYNC    NEARSYNC
HADR peer window duration (seconds)          (HADR_PEER_WINDOW) = 0          0
HADR spool log data limit (4KB)               (HADR_SPOOL_LIMIT) = 0          0
HADR log replay delay (seconds)               (HADR_REPLAY_DELAY) = 0          0

First log archive method                      (LOGARCHMETH1) = OFF          OFF
Archive compression for logarchmeth1         (LOGARCHCOMPR1) = OFF          OFF
Options for logarchmeth1                     (LOGARCHOPT1) =
Second log archive method                    (LOGARCHMETH2) = OFF          OFF
Archive compression for logarchmeth2         (LOGARCHCOMPR2) = OFF          OFF
Options for logarchmeth2                     (LOGARCHOPT2) =
Failover log archive path                    (FAILARCHPATH) =
Number of log archive retries on error        (NUMARCHRETRY) = 5            5
Log archive retry Delay (secs)                (ARCHRETRYDELAY) = 20         20
Vendor options                               (VENDOROPT) =
Auto restart enabled                          (AUTORESTART) = ON            ON
Index re-creation time and redo index build  (INDEXREC) = SYSTEM          SYSTEM (RESTART)
Log pages during index build                  (LOGINDEXBUILD) = OFF         OFF
Default number of loadrec sessions           (DFT_LOADREC_SES) = 1          1
Number of database backups to retain          (NUM_DB_BACKUPS) = 12         12
Recovery history retention (days)           (REC_HIS_RETENTN) = 366       366
Auto deletion of recovery objects            (AUTO_DEL_REC_OBJ) = OFF      OFF

TSM management class                         (TSM_MGMTCLASS) =
TSM node name                                (TSM_NODENAME) =
TSM owner                                    (TSM_OWNER) =
TSM password                                 (TSM_PASSWORD) =

Automatic maintenance                        (AUTO_MAINT) = ON            ON
  Automatic database backup                  (AUTO_DB_BACKUP) = OFF        OFF
  Automatic table maintenance                (AUTO_TBL_MAINT) = ON         ON
  Automatic runstats                        (AUTO_RUNSTATS) = ON          ON
  Automatic statement statistics             (AUTO_STMT_STATS) = ON        ON
  Automatic statistics profiling             (AUTO_STATS_PROF) = OFF       OFF
  Automatic profile updates                  (AUTO_PROF_UPD) = OFF         OFF
  Automatic reorganization                  (AUTO_REORG) = OFF            OFF
Auto-Revalidation                            (AUTO_REVAL) = DEFERRED      DEFERRED
Currently Committed                          (CUR_COMMIT) = ON             ON
CHAR output with DECIMAL input               (DEC_TO_CHAR_FMT) = NEW       NEW
Enable XML Character operations              (ENABLE_XMLCHAR) = YES        YES
WLM Collection Interval                      (WLM_COLLECT_INT) = 0         0

Monitor Collect Settings
Request metrics                              (MON_REQ_MATRICS) = BASE      BASE
Activity metrics                             (MON_ACT_MATRICS) = BASE      BASE
Object metrics                               (MON_OBJ_MATRICS) = BASE      BASE
Unit of work events                          (MON_UOW_DATA) = NONE         NONE
Lock timeout events                          (MON_LOCKTIMEOUT) = NONE      NONE
Deadlock events                             (MON_DEADLOCK) = WITHOUT_HIST WITHOUT_HIST
Lock wait events                             (MON_LOCKWAIT) = NONE         NONE
Lock wait event threshold                    (MON_LW_THRESH) = 0           5000000
SMTP Server                                  (SMTP_SERVER) =

```

### 使用上の注意

データベース構成ファイルが無効である場合には、データベースをバックアップ版からリストアしなければなりません。

## GET DATABASE CONFIGURATION

データベース構成パラメーターをデータベース・マネージャーのデフォルトに設定するには、**RESET DATABASE CONFIGURATION** コマンドを使用してください。

すべてのデータベース・パーティションから情報を取り出すには、**SYSIBMADM.DBCFG** 管理ビューを使用します。

**GET DATABASE CONFIGURATION** コマンドを発行することによって戻される構成パラメーター値は、ディスクに割り当てられた構成パラメーターの値とは少し異なる可能性があります。

## GET DATABASE MANAGER CONFIGURATION

データベース・マネージャー構成ファイル内の、個々の項目の値を取得します。

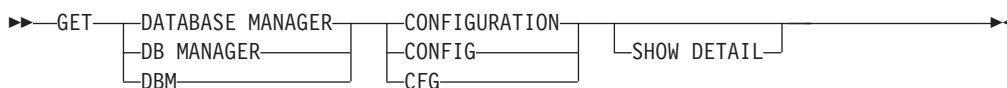
### 許可

なし

### 必要な接続

なし、またはインスタンス。インスタンス接続は、ローカル・データベース・マネージャーの構成操作を実行する場合には必要ありませんが、リモート・データベース・マネージャーの構成操作を実行する場合には必要です。リモート・インスタンスのデータベース・マネージャー構成を表示するには、まず最初にそのインスタンスとアタッチする必要があります。 **SHOW DETAIL** 節では、インスタンス接続が必要です。

### コマンド構文



### コマンド・パラメーター

#### SHOW DETAIL

データベース・マネージャー構成パラメーターの現行値、および次回データベース・マネージャーを始動する際のパラメーター値についての詳細情報を表示します。このオプションによって、構成パラメーターを動的に変更した結果を見ることができます。

これは、CLPPlus インターフェースで操作する場合のデフォルトの節です。**SHOW DETAIL** は、CLPPlus プロセッサの使用時に呼び出す必要はありません。

### 例

ノード・タイプとプラットフォームによって、どの構成パラメーターをリストするかが決まります。

**GET DATABASE MANAGER CONFIGURATION** コマンドを Linux オペレーティング・システムの DB2 pureScale インスタンスで実行した後の出力例を以下に示します。

```
Database Manager Configuration
```

```
Node type = Enterprise Server Edition with local and remote clients
```

```
Database manager configuration release level          = 0x0e00
```

```
CPU speed (millisec/instruction)                    (CPUSPEED) = 4.000000e-05
Communications bandwidth (MB/sec)                    (COMM_BANDWIDTH) = 1.000000e+02
```

```
Max number of concurrently active databases          (NUMDB) = 1
Federated Database System Support                   (FEDERATED) = NO
Transaction processor monitor name                   (TP_MON_NAME) =
```

```
Default charge-back account                          (DFT_ACCOUNT_STR) =
```

## GET DATABASE MANAGER CONFIGURATION

```

Java Development Kit installation path      (JDK_PATH) = /home/db2inst1/sql1lib/java/jdk64

Diagnostic error capture level             (DIAGLEVEL) = 3
Notify Level                              (NOTIFYLEVEL) = 3
Diagnostic data directory path             (DIAGPATH) = /home/hotel67/db2inst1/sql1lib/db2dump/ $m
Current member resolved DIAGPATH          = /home/hotel67/db2inst1/sql1lib/db2dump/DIAG0000
Alternate diagnostic data directory path (ALT_DIAGPATH) = /home/hotel67/db2inst1/sql1lib/db2altdump/ $m
Current member resolved ALT_DIAGPATH      = /home/hotel67/db2inst1/sql1lib/db2altdump/DIAG0000
Size of rotating db2diag & notify logs (MB) (DIAGSIZE) = 0

Default database monitor switches
  Buffer pool                             (DFT_MON_BUFPOOL) = OFF
  Lock                                    (DFT_MON_LOCK) = OFF
  Sort                                    (DFT_MON_SORT) = OFF
  Statement                               (DFT_MON_STMT) = OFF
  Table                                   (DFT_MON_TABLE) = OFF
  Timestamp                              (DFT_MON_TIMESTAMP) = ON
  Unit of work                            (DFT_MON_UOW) = OFF
Monitor health of instance and databases  (HEALTH_MON) = OFF

SYSADM group name                        (SYSADM_GROUP) = DB2ADMIN
SYSCTRL group name                      (SYSCTRL_GROUP) =
SYSMAINT group name                     (SYSMAINT_GROUP) =
SYSMON group name                       (SYSMON_GROUP) =

Client Userid-Password Plugin            (CLNT_PW_PLUGIN) =
Client Kerberos Plugin                   (CLNT_KRB_PLUGIN) =
Group Plugin                             (GROUP_PLUGIN) =
GSS Plugin for Local Authorization       (LOCAL_GSSPLUGIN) =
Server Plugin Mode                       (SRV_PLUGIN_MODE) = UNFENCED
Server List of GSS Plugins                (SRVCON_GSSPLUGIN_LIST) =
Server Userid-Password Plugin            (SRVCON_PW_PLUGIN) =
Server Connection Authentication          (SRVCON_AUTH) = NOT_SPECIFIED
Cluster manager                          = TSA

Database manager authentication           (AUTHENTICATION) = SERVER
Alternate authentication                  (ALTERNATE_AUTH_ENC) = NOT_SPECIFIED
Cataloging allowed without authority      (CATALOG_NOAUTH) = NO
Trust all clients                         (TRUST_ALLCLNTS) = YES
Trusted client authentication             (TRUST_CLNTAUTH) = CLIENT
Bypass federated authentication           (FED_NOAUTH) = NO

Default database path                    (DFTDBPATH) = /home/db2inst1

Database monitor heap size (4KB)          (MON_HEAP_SZ) = AUTOMATIC(90)
Java Virtual Machine heap size (4KB)     (JAVA_HEAP_SZ) = 2048
Audit buffer size (4KB)                   (AUDIT_BUF_SZ) = 0
Size of instance shared memory (4KB)     (INSTANCE_MEMORY) = AUTOMATIC(1705741)
Instance memory for restart light (%)    (RSTRT_LIGHT_MEM) = AUTOMATIC(10)
Backup buffer default size (4KB)          (BACKBUFSZ) = 1024
Restore buffer default size (4KB)         (RESTBUFSZ) = 1024

Agent stack size                          (AGENT_STACK_SZ) = 1024
Sort heap threshold (4KB)                 (SHEAPTHRES) = 0

Directory cache support                   (DIR_CACHE) = YES

Application support layer heap size (4KB) (ASLHEAPSZ) = 15
Max requester I/O block size (bytes)     (RQRIOBLK) = 32767
Query heap size (4KB)                     (QUERY_HEAP_SZ) = 1000

Workload impact by throttled utilities(UTIL_IMPACT_LIM) = 10

Priority of agents                        (AGENTPRI) = SYSTEM
Agent pool size                           (NUM_POOLAGENTS) = AUTOMATIC(100)
Initial number of agents in pool          (NUM_INITAGENTS) = 0
Max number of coordinating agents         (MAX_COORDAGENTS) = AUTOMATIC(200)
Max number of client connections         (MAX_CONNECTIONS) = AUTOMATIC(MAX_COORDAGENTS)

```



## GET DATABASE MANAGER CONFIGURATION

```

Keep fenced process                (KEEPFENCED) = YES
Number of pooled fenced processes  (FENCED_POOL) = AUTOMATIC(MAX_COORDAGENTS)
Initial number of fenced processes (NUM_INITFENCED) = 0

Index re-creation time and redo index build (INDEXREC) = RESTART

Transaction manager database name    (TM_DATABASE) = 1ST_CONN
Transaction resync interval (sec)    (RESYNC_INTERVAL) = 180

SPM name                            (SPM_NAME) =
SPM log size                         (SPM_LOG_FILE_SZ) = 256
SPM resync agent limit               (SPM_MAX_RESYNC) = 20
SPM log path                         (SPM_LOG_PATH) =

TCP/IP Service name                 (SVCENAME) =
Discovery mode                      (DISCOVER) = SEARCH
Discover server instance            (DISCOVER_INST) = ENABLE

SSL server keydb file               (SSL_SVR_KEYDB) =
SSL server stash file               (SSL_SVR_STASH) =
SSL server certificate label        (SSL_SVR_LABEL) =
SSL service name                   (SSL_SVCENAME) =
SSL cipher specs                   (SSL_CIPHERSPECS) =
SSL versions                        (SSL_VERSIONS) =
SSL client keydb file              (SSL_CLNT_KEYDB) =
SSL client stash file              (SSL_CLNT_STASH) =

Maximum query degree of parallelism (MAX_QUERYDEGREE) = ANY
Enable intra-partition parallelism  (INTRA_PARALLEL) = NO

Maximum Asynchronous TQs per query  (FEDERATED_ASYNC) = 0

No. of int. communication buffers(4KB) (FCM_NUM_BUFFERS) = AUTOMATIC(4096)
No. of int. communication channels    (FCM_NUM_CHANNELS) = AUTOMATIC(2048)
Node connection elapse time (sec)    (CONN_ELAPSE) = 3
Max number of node connection retries (MAX_CONNRETRIES) = 3
Max time difference between nodes (min) (MAX_TIME_DIFF) = 1

db2start/db2stop timeout (min)      (START_STOP_TIME) = 10

CF Server Configuration:
Memory size (4KB)                   (CF_MEM_SZ) = AUTOMATIC
Number of worker threads             (CF_NUM_WORKERS) = AUTOMATIC
Number of connections               (CF_NUM_CONNS) = AUTOMATIC
Diagnostic error capture level       (CF_DIAGLEVEL) = 2
Diagnostic data directory path       (CF_DIAGPATH) = /home/hotel67/db2inst1/sqllib/db2dump/ $m
Current member resolved CF_DIAGPATH = /home/hotel67/db2inst1/sqllib/db2dump/DIAG0000/

```

以下の出力例は、**SHOW DETAIL** オプションを指定したときに表示される情報を示しています。Delayed Value 列の値は、データベース・マネージャー・インスタンスを次回開始する際に有効になる値です。

```

Database Manager Configuration

Node type = Enterprise Server Edition with local and remote clients

Description                Parameter    Current Value    Delayed Value
-----
Database manager configuration release level    = 0x0e00

CPU speed (millicsec/instruction)             (CPUSPEED) = 4.000000e-05    4.000000e-05
Communications bandwidth (MB/sec)             (COMM_BANDWIDTH) = 1.000000e+02    1.000000e+02

Max number of concurrently active databases    (NUMDB) = 1    1
Federated Database System Support             (FEDERATED) = NO    NO
Transaction processor monitor name            (TP_MON_NAME) =

Default charge-back account                   (DFT_ACCOUNT_STR) =

Java Development Kit installation path         (JDK_PATH) = /home/db2inst1/sqllib/java/jdk64 /home/db2inst1/sqllib/java/jdk64

Diagnostic error capture level                 (DIAGLEVEL) = 3    3
Notify Level                                  (NOTIFYLEVEL) = 3    3
Diagnostic data directory path                 (DIAGPATH) = /home/hotel67/db2inst1/sqllib/db2dump/ $m

```



## GET DATABASE MANAGER CONFIGURATION

```

Current member resolved DIAGPATH                = /home/hotel67/db2inst1/sql1lib/db2dump/DIAG0000
Alternate diagnostic data directory path (ALT_DIAGPATH) = /home/hotel67/db2inst1/sql1lib/db2altdump/ $m
Current member resolved ALT_DIAGPATH            = /home/hotel67/db2inst1/sql1lib/db2altdump/DIAG0000
Size of rotating db2diag & notify logs (MB) (DIAGSIZE) = 0
0

Default database monitor switches
  Buffer pool (DFT_MON_BUFPOOL) = OFF OFF
  Lock (DFT_MON_LOCK) = OFF OFF
  Sort (DFT_MON_SORT) = OFF OFF
  Statement (DFT_MON_STMT) = OFF OFF
  Table (DFT_MON_TABLE) = OFF OFF
  Timestamp (DFT_MON_TIMESTAMP) = ON ON
  Unit of work (DFT_MON_UOW) = OFF OFF
Monitor health of instance and databases (HEALTH_MON) = OFF OFF

SYSADM group name (SYSADM_GROUP) = DB2ADMIN DB2ADMIN
SYSCTRL group name (SYSCTRL_GROUP) =
SYSMAINT group name (SYSMAINT_GROUP) =
SYSMON group name (SYSMON_GROUP) =

Client Userid-Password Plugin (CLNT_PW_PLUGIN) =
Client Kerberos Plugin (CLNT_KRB_PLUGIN) =
Group Plugin (GROUP_PLUGIN) =
GSS Plugin for Local Authorization (LOCAL_GSSPLUGIN) =
Server Plugin Mode (SRV_PLUGIN_MODE) = UNFENCED UNFENCED
Server List of GSS Plugins (SRVCON_GSSPLUGIN_LIST) =
Server Userid-Password Plugin (SRVCON_PW_PLUGIN) =
Server Connection Authentication (SRVCON_AUTH) = NOT_SPECIFIED NOT_SPECIFIED
Cluster manager = TSA TSA

Database manager authentication (AUTHENTICATION) = SERVER SERVER
Alternate authentication (ALTERNATE_AUTH_ENC) = NOT_SPECIFIED NOT_SPECIFIED
Cataloging allowed without authority (CATALOG_NOAUTH) = NO NO
Trust all clients (TRUST_ALLCLNTS) = YES YES
Trusted client authentication (TRUST_CLNTAUTH) = CLIENT CLIENT
Bypass federated authentication (FED_NOAUTH) = NO NO

Default database path (DFTDBPATH) = /home/db2inst1 /home/db2inst1

Database monitor heap size (4KB) (MON_HEAP_SZ) = AUTOMATIC(90) AUTOMATIC(90)
Java Virtual Machine heap size (4KB) (JAVA_HEAP_SZ) = 2048 2048
Audit buffer size (4KB) (AUDIT_BUF_SZ) = 0 0
Size of instance shared memory (4KB) (INSTANCE_MEMORY) = AUTOMATIC(1705741) AUTOMATIC(1705741)
Instance memory for restart light (%) (RSTRT_LIGHT_MEM) = AUTOMATIC(10) AUTOMATIC(10)
Agent stack size (AGENT_STACK_SZ) = 1024 1024
Sort heap threshold (4KB) (SHEAPTHRES) = 0 0

Directory cache support (DIR_CACHE) = YES YES

Application support layer heap size (4KB) (ASLHEAPSZ) = 15 15
Max requester I/O block size (bytes) (RQRIOBLK) = 32767 32767
Workload impact by throttled utilities (UTIL_IMPACT_LIM) = 10 10

Priority of agents (AGENTPRI) = SYSTEM SYSTEM
Agent pool size (NUM_POOLAGENTS) = AUTOMATIC(100) AUTOMATIC(100)
Initial number of agents in pool (NUM_INITAGENTS) = 0 0
Max number of coordinating agents (MAX_COORDAGENTS) = AUTOMATIC(200) AUTOMATIC(200)
Max number of client connections (MAX_CONNECTIONS) = AUTOMATIC(MAX_COORDAGENTS) AUTOMATIC(MAX_COORDAGENTS)

Keep fenced process (KEEPFENCED) = YES YES
Number of pooled fenced processes (FENCED_POOL) = AUTOMATIC(MAX_COORDAGENTS) AUTOMATIC(MAX_COORDAGENTS)
Initial number of fenced processes (NUM_INITFENCED) = 0 0

Index re-creation time and redo index build (INDEXREC) = RESTART RESTART

Transaction manager database name (TM_DATABASE) = 1ST_CONN 1ST_CONN
Transaction resync interval (sec) (RESYNC_INTERVAL) = 180 180

SPM name (SPM_NAME) =
SPM log size (SPM_LOG_FILE_SZ) = 256 256
SPM resync agent limit (SPM_MAX_RESYNC) = 20 20
SPM log path (SPM_LOG_PATH) =

TCP/IP Service name (SVCENAME) =
Discovery mode (DISCOVER) = SEARCH SEARCH
Discover server instance (DISCOVER_INST) = ENABLE ENABLE

SSL server keydb file (SSL_SVR_KEYDB) =
SSL server stash file (SSL_SVR_STASH) =
SSL server certificate label (SSL_SVR_LABEL) =
SSL service name (SSL_SVCENAME) =
SSL cipher specs (SSL_CIPHERSPECS) =
SSL versions (SSL_VERSIONS) =

```

## GET DATABASE MANAGER CONFIGURATION

```
SSL client keydb file          (SSL_CLNT_KEYDB) =
SSL client stash file         (SSL_CLNT_STASH) =

Maximum query degree of parallelism (MAX_QUERYDEGREE) = ANY
Enable intra-partition parallelism (INTRA_PARALLEL) = NO

Maximum Asynchronous TQs per query (FEDERATED_ASYNC) = 0

No. of int. communication buffers(4KB)(FCM_NUM_BUFFERS) = AUTOMATIC(4096)
No. of int. communication channels (FCM_NUM_CHANNELS) = AUTOMATIC(2048)
Node connection elapse time (sec) (CONN_ELAPSE) = 3
Max number of node connection retries (MAX_CONNRETRIES) = 3
Max time difference between nodes (min) (MAX_TIME_DIFF) = 1

db2start/db2stop timeout (min) (START_STOP_TIME) = 10

CF Server Configuration:
Memory size (4KB) (CF_MEM_SZ) = AUTOMATIC(131072)
Number of worker threads (CF_NUM_WORKERS) = AUTOMATIC(1)
Number of connections (CF_NUM_CONNS) = AUTOMATIC(19)
Diagnostic error capture level (CF_DIAGLEVEL) = 2
Diagnostic data directory path (CF_DIAGPATH) = /home/hotel67/db2inst1/sql1lib/ $m
Current member resolved CF_DIAGPATH = /home/hotel67/db2inst1/sql1lib/db2dump/DIAG0000/
```

**GET DATABASE MANAGER CONFIGURATION** コマンドを Windows オペレーティング・システムで実行した後の出力例を以下に示します。

### Database Manager Configuration

Node type = Enterprise Server Edition with local and remote clients

```
Database manager configuration release level          = 0x0c00

Maximum total of files open (MAXTOTFILOP) = 16000
CPU speed (millisec/instruction) (CPUSPEED) = 4.251098e-007
Communications bandwidth (MB/sec) (COMM_BANDWIDTH) = 1.000000e+002

Max number of concurrently active databases (NUMDB) = 8
Federated Database System Support (FEDERATED) = NO
Transaction processor monitor name (TP_MON_NAME) =

Default charge-back account (DFT_ACCOUNT_STR) =

Java Development Kit installation path (JDK_PATH) =

Diagnostic error capture level (DIAGLEVEL) = 3
Notify Level (NOTIFYLEVEL) = 3
Diagnostic data directory path (DIAGPATH) =

Default database monitor switches
Buffer pool (DFT_MON_BUFPOOL) = OFF
Lock (DFT_MON_LOCK) = OFF
Sort (DFT_MON_SORT) = OFF
Statement (DFT_MON_STMT) = OFF
Table (DFT_MON_TABLE) = OFF
Timestamp (DFT_MON_TIMESTAMP) = ON
Unit of work (DFT_MON_UOW) = OFF
Monitor health of instance and databases (HEALTH_MON) = ON

SYSADM group name (SYSADM_GROUP) =
SYSCTRL group name (SYSCTRL_GROUP) =
SYSMAINT group name (SYSMAINT_GROUP) =
SYSMON group name (SYSMON_GROUP) =

Client Userid-Password Plugin (CLNT_PW_PLUGIN) =
Client Kerberos Plugin (CLNT_KRB_PLUGIN) = IBMkrb5
Group Plugin (GROUP_PLUGIN) =
GSS Plugin for Local Authorization (LOCAL_GSSPLUGIN) =
Server Plugin Mode (SRV_PLUGIN_MODE) = UNFENCED
Server List of GSS Plugins (SRVCON_GSSPLUGIN_LIST) =
Server Userid-Password Plugin (SRVCON_PW_PLUGIN) =
Server Connection Authentication (SRVCON_AUTH) = NOT_SPECIFIED
Cluster manager (CLUSTER_MGR) =
```

## GET DATABASE MANAGER CONFIGURATION

```

Database manager authentication      (AUTHENTICATION) = SERVER
Cataloging allowed without authority (CATALOG_NOAUTH) = NO
Trust all clients                   (TRUST_ALLCLNTS) = YES
Trusted client authentication      (TRUST_CLNTAUTH) = CLIENT
Bypass federated authentication    (FED_NOAUTH) = NO

Default database path               (DFTDBPATH) = C:

Database monitor heap size (4KB)    (MON_HEAP_SZ) = AUTOMATIC
Java Virtual Machine heap size (4KB) (JAVA_HEAP_SZ) = 2048
Audit buffer size (4KB)            (AUDIT_BUF_SZ) = 0
Size of instance shared memory (4KB) (INSTANCE_MEMORY) = AUTOMATIC
Backup buffer default size (4KB)    (BACKBUFSZ) = 1024
Restore buffer default size (4KB)   (RESTBUFSZ) = 1024

Agent stack size                    (AGENT_STACK_SZ) = 16
Minimum committed private memory (4KB) (MIN_PRIV_MEM) = 32
Private memory threshold (4KB)      (PRIV_MEM_THRESH) = 20000

Sort heap threshold (4KB)           (SHEAPTHRES) = 0

Directory cache support             (DIR_CACHE) = YES

Application support layer heap size (4KB) (ASLHEAPSZ) = 15
Max requester I/O block size (bytes) (RQRIOBLK) = 32767
Query heap size (4KB)              (QUERY_HEAP_SZ) = 1000

Workload impact by throttled utilities (UTIL_IMPACT_LIM) = 10

Priority of agents                   (AGENTPRI) = SYSTEM
Agent pool size                     (NUM_POOLAGENTS) = AUTOMATIC
Initial number of agents in pool    (NUM_INITAGENTS) = 0
Max number of coordinating agents    (MAX_COORDAGENTS) = AUTOMATIC
Max number of client connections    (MAX_CONNECTIONS) = AUTOMATIC

Keep fenced process                 (KEEPFENCED) = YES
Number of pooled fenced processes   (FENCED_POOL) = AUTOMATIC
Initial number of fenced processes  (NUM_INITFENCED) = 0

Index re-creation time and redo index build (INDEXREC) = RESTART

Transaction manager database name    (TM_DATABASE) = 1ST_CONN
Transaction resync interval (sec)    (RESYNC_INTERVAL) = 180

SPM name                            (SPM_NAME) = KEON14
SPM log size                         (SPM_LOG_FILE_SZ) = 256
SPM resync agent limit               (SPM_MAX_RESYNC) = 20
SPM log path                         (SPM_LOG_PATH) =

NetBIOS Workstation name             (NNAME) =

TCP/IP Service name                  (SVCENAME) = db2c_DB2
Discovery mode                       (DISCOVER) = SEARCH
Discover server instance              (DISCOVER_INST) = ENABLE

Maximum query degree of parallelism  (MAX_QUERYDEGREE) = ANY
Enable intra-partition parallelism  (INTRA_PARALLEL) = NO

Maximum Asynchronous TQs per query  (FEDERATED_ASYNC) = 0

No. of int. communication buffers(4KB) (FCM_NUM_BUFFERS) = AUTOMATIC
No. of int. communication channels   (FCM_NUM_CHANNELS) = AUTOMATIC
Node connection elapse time (sec)    (CONN_ELAPSE) = 10
Max number of node connection retries (MAX_CONNRETRIES) = 5

```

## GET DATABASE MANAGER CONFIGURATION

Max time difference between nodes (min) (MAX\_TIME\_DIFF) = 60

db2start/db2stop timeout (min) (START\_STOP\_TIME) = 10

**SHOW DETAIL** オプションを Windows オペレーティング・システムで指定した場合に表示される情報の出力例を以下に示します。Delayed Value 列の値は、データベース・マネージャー・インスタンスを次回開始する際に有効になる値です。

```
db2 => get dbm cfg show detail
```

### Database Manager Configuration

Node type = Enterprise Server Edition with local and remote clients

| Description                                  | Parameter                 | Current Value | Delayed Value |
|----------------------------------------------|---------------------------|---------------|---------------|
| -----                                        |                           |               |               |
| Database manager configuration release level |                           | = 0x0c00      |               |
| Maximum total of files open                  | (MAXTOTFILOP) =           | 16000         | 16000         |
| CPU speed (millisec/instruction)             | (CPUSPEED) =              | 4.251098e-007 | 4.251098e-007 |
| Communications bandwidth (MB/sec)            | (COMM_BANDWIDTH) =        | 1.000000e+002 | 1.000000e+002 |
| Max number of concurrently active databases  | (NUMDB) =                 | 8             | 8             |
| Federated Database System Support            | (FEDERATED) =             | NO            | NO            |
| Transaction processor monitor name           | (TP_MON_NAME) =           |               |               |
| Default charge-back account                  | (DFT_ACCOUNT_STR) =       |               |               |
| Java Development Kit installation path       | (JDK_PATH) =              |               |               |
| Diagnostic error capture level               | (DIAGLEVEL) =             | 3             | 3             |
| Notify Level                                 | (NOTIFYLEVEL) =           | 3             | 3             |
| Diagnostic data directory path               | (DIAGPATH) =              |               |               |
| Default database monitor switches            |                           |               |               |
| Buffer pool                                  | (DFT_MON_BUFPOOL) =       | OFF           | OFF           |
| Lock                                         | (DFT_MON_LOCK) =          | OFF           | OFF           |
| Sort                                         | (DFT_MON_SORT) =          | OFF           | OFF           |
| Statement                                    | (DFT_MON_STMT) =          | OFF           | OFF           |
| Table                                        | (DFT_MON_TABLE) =         | OFF           | OFF           |
| Timestamp                                    | (DFT_MON_TIMESTAMP) =     | ON            | ON            |
| Unit of work                                 | (DFT_MON_UOW) =           | OFF           | OFF           |
| Monitor health of instance and databases     | (HEALTH_MON) =            | ON            | ON            |
| SYSADM group name                            | (SYSADM_GROUP) =          |               |               |
| SYSCTRL group name                           | (SYSCTRL_GROUP) =         |               |               |
| SYSMAINT group name                          | (SYSMAINT_GROUP) =        |               |               |
| SYSMON group name                            | (SYSMON_GROUP) =          |               |               |
| Client Userid-Password Plugin                | (CLNT_PW_PLUGIN) =        |               |               |
| Client Kerberos Plugin                       | (CLNT_KRB_PLUGIN) =       | IBMkrb5       | IBMkrb5       |
| Group Plugin                                 | (GROUP_PLUGIN) =          |               |               |
| GSS Plugin for Local Authorization           | (LOCAL_GSSPLUGIN) =       |               |               |
| Server Plugin Mode                           | (SRV_PLUGIN_MODE) =       | UNFENCED      | UNFENCED      |
| Server List of GSS Plugins                   | (SRVCON_GSSPLUGIN_LIST) = |               |               |
| Server Userid-Password Plugin                | (SRVCON_PW_PLUGIN) =      |               |               |
| Server Connection Authentication             | (SRVCON_AUTH) =           | NOT_SPECIFIED | NOT_SPECIFIED |
| Cluster manager                              | (CLUSTER_MGR) =           |               |               |

## GET DATABASE MANAGER CONFIGURATION

|                                             |                                                |                            |
|---------------------------------------------|------------------------------------------------|----------------------------|
| Database manager authentication             | (AUTHENTICATION) = SERVER                      | SERVER                     |
| Cataloging allowed without authority        | (CATALOG_NOAUTH) = NO                          | NO                         |
| Trust all clients                           | (TRUST_ALLCLNTS) = YES                         | YES                        |
| Trusted client authentication               | (TRUST_CLNTAUTH) = CLIENT                      | CLIENT                     |
| Bypass federated authentication             | (FED_NOAUTH) = NO                              | NO                         |
| Default database path                       | (DFTDBPATH) = C:                               | C:                         |
| Database monitor heap size (4KB)            | (MON_HEAP_SZ) = AUTOMATIC(66)                  | AUTOMATIC(66)              |
| Java Virtual Machine heap size (4KB)        | (JAVA_HEAP_SZ) = 2048                          | 2048                       |
| Audit buffer size (4KB)                     | (AUDIT_BUF_SZ) = 0                             | 0                          |
| Size of instance shared memory (4KB)        | (INSTANCE_MEMORY) = AUTOMATIC(73728)           | AUTOMATIC(73728)           |
| Backup buffer default size (4KB)            | (BACKBUFSZ) = 1024                             | 1024                       |
| Restore buffer default size (4KB)           | (RESTBUFSZ) = 1024                             | 1024                       |
| Agent stack size                            | (AGENT_STACK_SZ) = 16                          | 16                         |
| Sort heap threshold (4KB)                   | (SHEAPTHRES) = 0                               | 0                          |
| Directory cache support                     | (DIR_CACHE) = YES                              | YES                        |
| Application support layer heap size (4KB)   | (ASLHEAPSZ) = 15                               | 15                         |
| Max requester I/O block size (bytes)        | (RQRIOBLK) = 32767                             | 32767                      |
| Query heap size (4KB)                       | (QUERY_HEAP_SZ) = 1000                         | 1000                       |
| Workload impact by throttled utilities      | (UTIL_IMPACT_LIM) = 10                         | 10                         |
| Priority of agents                          | (AGENTPRI) = SYSTEM                            | SYSTEM                     |
| Agent pool size                             | (NUM_POOLAGENTS) = AUTOMATIC(100)              | AUTOMATIC(100)             |
| Initial number of agents in pool            | (NUM_INITAGENTS) = 0                           | 0                          |
| Max number of coordinating agents           | (MAX_COORDAGENTS) = AUTOMATIC(200)             | AUTOMATIC(200)             |
| Max number of client connections            | (MAX_CONNECTIONS) = AUTOMATIC(MAX_COORDAGENTS) | AUTOMATIC(MAX_COORDAGENTS) |
| Keep fenced process                         | (KEEPFENCED) = YES                             | YES                        |
| Number of pooled fenced processes           | (FENCED_POOL) = AUTOMATIC(MAX_COORDAGENTS)     | AUTOMATIC(MAX_COORDAGENTS) |
| Initial number of fenced processes          | (NUM_INITFENCED) = 0                           | 0                          |
| Index re-creation time and redo index build | (INDEXREC) = RESTART                           | RESTART                    |
| Transaction manager database name           | (TM_DATABASE) = 1ST_CONN                       | 1ST_CONN                   |
| Transaction resync interval (sec)           | (RESYNC_INTERVAL) = 180                        | 180                        |
| SPM name                                    | (SPM_NAME) = KEON14                            | KEON14                     |
| SPM log size                                | (SPM_LOG_FILE_SZ) = 256                        | 256                        |
| SPM resync agent limit                      | (SPM_MAX_RESYNC) = 20                          | 20                         |
| SPM log path                                | (SPM_LOG_PATH) =                               |                            |
| NetBIOS Workstation name                    | (NNAME) =                                      |                            |
| TCP/IP Service name                         | (SVCENAME) = db2c_DB2                          | db2c_DB2                   |
| Discovery mode                              | (DISCOVER) = SEARCH                            | SEARCH                     |
| Discover server instance                    | (DISCOVER_INST) = ENABLE                       | ENABLE                     |
| Maximum query degree of parallelism         | (MAX_QUERYDEGREE) = ANY                        | ANY                        |
| Enable intra-partition parallelism          | (INTRA_PARALLEL) = NO                          | NO                         |
| Maximum Asynchronous TQs per query          | (FEDERATED_ASYNC) = 0                          | 0                          |
| No. of int. communication buffers(4KB)      | (FCM_NUM_BUFFERS) = AUTOMATIC(4096)            | AUTOMATIC(4096)            |
| No. of int. communication channels          | (FCM_NUM_CHANNELS) = AUTOMATIC(2048)           | AUTOMATIC(2048)            |
| Node connection elapse time (sec)           | (CONN_ELAPSE) = 10                             | 10                         |
| Max number of node connection retries       | (MAX_CONNRETRIES) = 5                          | 5                          |
| Max time difference between nodes (min)     | (MAX_TIME_DIFF) = 60                           | 60                         |
| db2start/db2stop timeout (min)              | (START_STOP_TIME) = 10                         | 10                         |

### 使用上の注意

- リモート・インスタンスまたは別のローカル・インスタンスへのアタッチが存在する場合、それらのインスタンスにアタッチされたサーバーのデータベース・マネージャー構成パラメーターが返されます。そのようなインスタンスが存在しない場合には、ローカルのデータベース・マネージャー構成パラメーターが返されます。
- エラーが生じた場合には、返された情報は無効になります。構成ファイルが無効な場合には、エラー・メッセージが戻されます。ユーザーはインスタンスを除去し、それを再作成してリカバリーする必要があります。
- データベース・マネージャー出荷時のデフォルト値に構成パラメーターを設定するには、**RESET DATABASE MANAGER CONFIGURATION** コマンドを使用してください。
- **GET DATABASE MANAGER CONFIGURATION SHOW DETAIL** において **fcm\_num\_buffers** および **fcm\_num\_channels** について指定される **AUTOMATIC** の値は、インスタンス起動時の初期値であり、実行時に発生した可能性のある自動増減は反映されていません。
- 構成パラメーター **max\_connections**、**max\_coordagents**、および **num\_poolagents** は **AUTOMATIC** に設定されます。
- 構成パラメーター **maxagents** および **maxcagents** は使用すべきではありません。推奨されない関数を使用すると、次のようになります。
  - **CLP** および **db2CfgSet** API はこれらのパラメーターへの更新を許容しますが、更新は **DB2** には無視されます。
  - クライアントおよびサーバーが **DB2 v9.5** コード・ベース上にある場合、**CLP** はこれらのデータベース構成パラメーターを表示しなくなります。サーバーが **DB2 v9.5** の場合、それより前のバージョンのクライアントではこれらのパラメーターの出力の値として **0** が表示されます。クライアントが **DB2 v9.5** であるが、サーバーが **DB2 v9.5** より前である場合、パラメーターは割り当てられた値で表示されます。
  - **db2CfgGet** API は **SQLF\_KTN\_MAXAGENTS** および **SQLF\_KTN\_MAXCAGENTS** の要求を許容しますが、サーバーが **DB2 v9.5** の場合は **0** が戻されます。
  - **db2AutoConfig** API の動作は、API に渡される **db2VersionNumber** によって決まります。バージョンが **DB2 v9.5** 以上の場合、**maxagents** は戻されませんが、これより前のバージョンの場合は戻されます。
  - **AUTOCONFIGURE** **CLP** コマンドは、**maxagents** の値を、以前のバージョンのクライアントからの要求とともに表示します (現行値および推奨値は **0**)。現行バージョンのクライアント要求の場合、**maxagents** は適切な値で表示されます。
  - サーバーが **DB2 v9.5** 以上の場合、**AUTOCONFIGURE** **ADMIN\_CMD** は **maxagents** に関する情報を戻しません。
  - **ADMIN\_CMD** による **maxagents** または **maxcagents** への更新は正常に戻されますが、サーバーが **DB2 v9.5** 以上の場合、そうした更新はサーバーに対して無効です。
  - サーバーが **DB2 v9.5** 以上の場合、**DBMCFG** 管理ビューを使用したデータベース・マネージャー構成パラメーターの照会では **maxagents** または **maxcagents** の行は戻されません。

## GET DATABASE MANAGER CONFIGURATION

今後のリリースでは、これらの構成パラメーターは完全に除去される予定です。



---

## GET DATABASE MANAGER MONITOR SWITCHES

現在アクティブなデータベース・システム・モニター・スイッチの状況を表示します。モニター・スイッチは、データベース活動情報を収集するように、データベース・マネージャーに指示します。

データベース・システム・モニター・インターフェースを使用している各アプリケーションには、それ自体のモニター・スイッチの集合があります。モニター中のアプリケーションのいずれかで ON にすると、データベース・システム・モニター・スイッチは ON になります。データベース・システム・モニターが現在どんなデータを収集しているかを判別するために、このコマンドを使用します。データベース・マネージャー構成に設定されているデフォルトのモニター・スイッチで指定されているものよりも多くのデータが収集されている可能性もあります。それは、アプリケーションがそれ自体のビューの中に ON に設定されたスイッチを持っているためです。

### 有効範囲

このコマンドは、現在アタッチされているメンバーに対して呼び出され、デフォルトではそのメンバーについての情報のみを返します。現在アタッチされているメンバーは、デフォルトでは、コマンドが実行されるホストになります。ホストに対して複数のメンバーがある場合、現在アタッチされているメンバーは、そのホスト上の `db2nodes.cfg` ファイルにリストされている最初のメンバーです。

現在アタッチされているメンバーではないメンバーに対してコマンドを実行する場合、特定のメンバーに対してこのコマンドを実行するには **AT MEMBER** オプションを指定し、すべてのメンバーに対してこのコマンドを実行して集約された結果を受け取るには **GLOBAL** オプションを指定してください。

後続のコマンドを実行する対象の、現在アタッチされているメンバーを変更するには、**SET CLIENT** コマンドの **ATTACH\_MEMBER** パラメーターを使用します。このクライアント設定を有効にするには、アプリケーションを **DETACH** してから再び **ATTACH** する必要があります。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMOINT
- SYSMON

### 必要な接続

インスタンスまたはデータベース

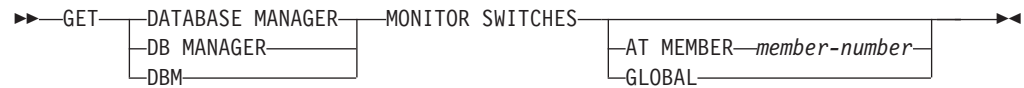
- インスタンスへのアタッチや、データベースへの接続がない場合、デフォルトのインスタンス接続が作成されます。
- インスタンスへのアタッチとデータベース接続の両方がある場合、インスタンス接続が使用されます。



## GET DATABASE MANAGER MONITOR SWITCHES

リモート・インスタンス、または別のローカル・インスタンスの設定値を表示するには、まず最初にそのインスタンスとアタッチすることが必要です。

### コマンド構文



### コマンド・パラメーター

#### AT MEMBER *member-number*

データベース・マネージャー・モニター・スイッチの状況を表示するメンバーを指定します。

#### GLOBAL

すべてのメンバーからデータベース・マネージャー・モニター・スイッチの状況に戻します。

### 例

以下に示すのは、**GET DATABASE MANAGER MONITOR SWITCHES** の出力例です。

DBM System Monitor Information Collected

```
Switch list for member 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
```

### 使用上の注意

記録スイッチ BUFFERPOOL、LOCK、SORT、STATEMENT、TABLE、および UOW は、デフォルトですべてオフになっています。いずれも **UPDATE MONITOR SWITCHES** コマンドを使用して、オンにすることができます。これらのスイッチのいずれかをオンにすると、このコマンドはそのスイッチがオンになった時点のタイム・スタンプも表示します。

記録スイッチ **TIMESTAMP** はデフォルトではオンですが、**UPDATE MONITOR SWITCHES** を使用してオフに切り替えることもできます。このスイッチがオンのとき、システムはタイム・スタンプ・モニター・エレメントについての情報を収集する際にタイム・スタンプ呼び出しを出します。これらのエレメントの例を以下に示します。

- **agent\_sys\_cpu\_time**
- **agent\_usr\_cpu\_time**
- **appl\_con\_time**
- **con\_elapsed\_time**
- **con\_response\_time**
- **conn\_complete\_time**

## GET DATABASE MANAGER MONITOR SWITCHES

- `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 スイッチがオフの場合、タイム・スタンプ・オペレーティング・システム呼び出しが出されてこれらのエレメントを判別することはありません。これらのエレメントにはゼロが含まれることとなります。CPU 使用率が 100% に近づくと、このスイッチのオフが重要になります。これが起きた場合、タイム・スタンプを出すために必要な CPU 時間は急激に増加します。

### 互換性

以前のバージョンとの互換性:

- **DB2\_ENFORCE\_MEMBER\_SYNTAX** レジストリー変数が ON に設定されている場合を除き、**DBPARTITIONNUM** または **NODE** を **MEMBER** の代わりに使用できます。

## GET DESCRIPTION FOR HEALTH INDICATOR

指定されたヘルス・インディケータの記述を取得します。ヘルス・インディケータは、データベース・システムの特定の状態、能力、または振る舞いの正常度を測定します。状態は、データベース・オブジェクトまたはリソースが通常通り操作しているかどうか定義します。

### 許可

なし

### 必要な接続

インスタンス。インスタンス接続が存在しない場合は、デフォルトのインスタンス接続が作成されます。

リモート・インスタンスのスナップショットを獲得するには、まず最初にそのインスタンスにアタッチすることが必要です。

### コマンド構文

► GET DESCRIPTION FOR HEALTH INDICATOR *shortname* ◄

### コマンド・パラメーター

**HEALTH INDICATOR** *shortname*

記述を検索したいヘルス・インディケータの名前。ヘルス・インディケータ名は、2、3 文字のオブジェクト ID に、インディケータが測定するものを説明する名前が続きます。以下に例を示します。

```
db.sort_privmem_util
```

### 例

以下に示すのは、**GET DESCRIPTION FOR HEALTH INDICATOR** コマンドの出力例です。

```
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.

---

### GET HEALTH NOTIFICATION CONTACT LIST

インスタンスのヘルスについて通知される連絡先および連絡先グループのリストを取得します。連絡先リストは、非常時ヘルス状況がインスタンスまたはそのデータベース・オブジェクトに示されたときに通知される、個人の電子メール・アドレスまたはページャー・インターネット・アドレスで構成されます。

#### 許可

なし

#### 必要な接続

インスタンス。明示的なアタッチは必要ありません。

#### コマンド構文

```
▶▶ GET HEALTH NOTIFICATION CONTACT LIST ◀◀
```

#### コマンド・パラメーター

なし

#### 例

**GET NOTIFICATION LIST** コマンドを実行した、出力結果です。

| Name      | Type          |
|-----------|---------------|
| Joe Brown | Contact       |
| Support   | Contact group |

## GET HEALTH SNAPSHOT

データベース・マネージャーとそのデータベースのヘルス状況情報を検索します。戻された情報は、コマンドを発行した時点でのヘルス状態のスナップショットを表しています。

**重要:** バージョン 9.7 でヘルス・モニターが非推奨になったため、このコマンドおよび API は推奨されておらず、将来のリリースで除去される可能性があります。これはDB2 pureScale環境ではサポートされていません。詳しくは、『ヘルス・モニターが推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055045.html>) を参照してください。

### 有効範囲

このコマンドは、現在アタッチされているメンバーに対して呼び出され、デフォルトではそのメンバーについての情報のみを返します。現在アタッチされているメンバーは、デフォルトでは、コマンドが実行されるホストになります。ホストに対して複数のメンバーがある場合、現在アタッチされているメンバーは、そのホスト上の `db2nodes.cfg` ファイルにリストされている最初のメンバーです。

現在アタッチされているメンバーではないメンバーに対してコマンドを実行する場合、特定のメンバーに対してこのコマンドを実行するには **AT MEMBER** オプションを指定し、すべてのメンバーに対してこのコマンドを実行して集約された結果を受け取るには **GLOBAL** オプションを指定してください。

後続のコマンドを実行する対象の、現在アタッチされているメンバーを変更するには、**SET CLIENT** コマンドの **ATTACH\_MEMBER** パラメーターを使用します。このクライアント設定を有効にするには、アプリケーションを **DETACH** してから再び **ATTACH** する必要があります。

### 許可

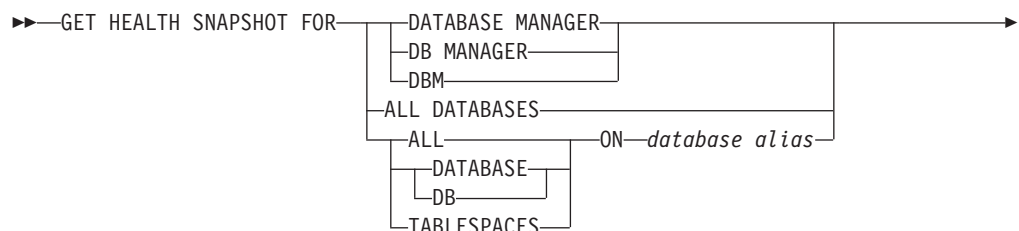
なし

### 必要な接続

インスタンス。インスタンス接続が存在しない場合は、デフォルトのインスタンス接続が作成されます。

リモート・インスタンスのスナップショットを獲得するには、まず最初にそのインスタンスにアタッチすることが必要です。

### コマンド構文





## コマンド・パラメーター

### DATABASE MANAGER

アクティブ・データベース・マネージャー・インスタンスの統計を提供します。

### ALL DATABASES

現行データベース・パーティション上のアクティブ・データベースすべてに関する稼働状態を提供します。

### ALL ON *database-alias*

指定したデータベースの表スペースおよびバッファー・プールに関するヘルス状態と情報を提供します。

### DATABASE ON *database-alias*

### TABLESPACES ON *database-alias*

特定のデータベースの表スペースに関する情報を提供します。

### AT MEMBER *member-number*

指定されたメンバーの結果を戻します。

### GLOBAL

すべてのメンバーのヘルス・モニター状況を戻します。

### SHOW DETAIL

各ヘルス・モニター・データ・エレメントの履歴データを {(Timestamp, Value, Formula)} という形式で出力に含めるように指定します。括弧で囲まれたパラメーター (Timestamp, Value, Formula) は、返される履歴レコードごとに繰り返されます。例えば、

```
(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))
```

ATTENTION または AUTOMATE FAILED の状態にあるすべてのコレクション・オブジェクトのコレクション・オブジェクト履歴が戻されます。

また、**SHOW DETAIL** オプションは、関連ヘルス・インディケーターの値とアラート状態を理解する上で役立つ追加のコンテキスト情報も提供します。例えば、表スペースのストレージ使用率ヘルス・インディケーターを使用して表スペースの使用率を判別する場合、表スペースの増大率も **SHOW DETAIL** によって提供されます。

### WITH FULL COLLECTION

コレクションの状態に基づくすべてのヘルス・インディケーターの完全なコレクション情報を戻すことを指定します。このオプションは、名前およびサイズ・フィルターの両方の基準を検査します。ユーザーが完全なコレクションを伴うヘルス・スナップショットを要求する場合、レポートは、ポリシー内の名前およびサイズの基準に適合するすべての表を示します。これを使用して、特定のリフレッシュ・サイクルでどの表が評価されるかを妥当性検査できます。このオプションを指定した場合に戻される出力は、

NORMAL、AUTOMATED、ATTENTION、AUTOMATE FAILED のいずれかの状態にあるコレクション・オブジェクトに関する出力になります。このオプションは、**SHOW DETAIL** オプションと一緒に指定できます。

このオプションを指定しない場合、自動再編成のために評価されて手操作による介入を必要とする（つまり、手動の再編成または自動操作が失敗した）表だけが、GET HEALTH SNAPSHOT によるレポートに含まれることになります。

## 例

以下のセクションに示すのは、データベース・マネージャー情報を要求した結果として表示される一般的な出力です。

```
D:>DB2 GET HEALTH SNAPSHOT FOR DBM
```

### Database Manager Health Snapshot

```
Node name                =
Node type                 = Enterprise Server Edition
                        with local and remote clients
Instance name            = DB2
Snapshot timestamp       = 02/17/2004 12:39:44.818949

Number of members in DB2 instance = 1
Start Database Manager timestamp = 02/17/2004 12:17:21.000119
Instance highest severity alert state = Normal
```

#### Health Indicators:

```
Indicator Name           = db2.db2_op_status
Value                    = 0
Evaluation timestamp     = 02/17/2004 12:37:23.393000
Alert state              = Normal
```

```
Indicator Name           = db2.sort_privmem_util
Value                    = 0
Unit                     = %
Evaluation timestamp     = 02/17/2004 12:37:23.393000
Alert state              = Normal
```

```
Indicator Name           = db2.mon_heap_util
Value                    = 6
Unit                     = %
Evaluation timestamp     = 02/17/2004 12:37:23.393000
Alert state              = Normal
```

## 使用上の注意

**GET HEALTH SNAPSHOT** コマンドが、データ・パーティション表のデータおよび索引を再編成するための推奨値を戻す場合、この推奨値は単に表レベルのものであり、表の何らかの個別のパーティションに特定されたものではありません。DB2 パーティション 9.7 フィックスパック 1 以降では、**REORG INDEXES/TABLE** コマンドまたは db2Reorg API を使用して、特定のデータ・パーティションのデータまたはパーティション索引を再編成できます。データ・パーティション表の特定のデータ・パーティションのみを再編成する必要があるかどうか判断するために、**REORGCHK** コマンドを使用して、データ・パーティション表のデータ・パーティションについての統計および推奨値を取得します。**ON DATA PARTITION** 節と共に **REORG TABLE** または **REORG INDEXES ALL** コマンドを使用して、特定のデータ・パーティションのデータまたはパーティション索引を再編成します。

## GET HEALTH SNAPSHOT

### 互換性

以前のバージョンとの互換性:

- **DB2\_ENFORCE\_MEMBER\_SYNTAX** レジストリー変数が ON に設定されている場合を除き、**DBPARTITIONNUM** を **MEMBER** の代わりに使用できます。



---

## GET INSTANCE

**DB2INSTANCE** 環境変数の値を戻します。

### 許可

なし

### 必要な接続

なし

### コマンド構文

▶▶—GET INSTANCE—▶▶

### コマンド・パラメーター

なし

### 例

以下に示すのは、**GET INSTANCE** の出力例です。

```
The current database manager instance is: smith
```

## GET MONITOR SWITCHES

現行セッションのデータベース・システム・モニター・スイッチの状況を表示します。モニター・スイッチは、データベース活動情報を収集するように、データベース・マネージャーに指示します。

データベース・システム・モニター・インターフェースを使用している各アプリケーションには、それ自体のモニター・スイッチの集合があります。このコマンドはそれらを表示します。データベース・マネージャー・レベル・スイッチを表示するには、**GET DBM MONITOR SWITCHES** コマンドを使用します。

### 有効範囲

このコマンドは、現在アタッチされているメンバーに対して呼び出され、デフォルトではそのメンバーについての情報のみを返します。現在アタッチされているメンバーは、デフォルトでは、コマンドが実行されるホストになります。ホストに対して複数のメンバーがある場合、現在アタッチされているメンバーは、そのホスト上の `db2nodes.cfg` ファイルにリストされている最初のメンバーです。

現在アタッチされているメンバーではないメンバーに対してコマンドを実行する場合、特定のメンバーに対してこのコマンドを実行するには **AT MEMBER** オプションを指定し、すべてのメンバーに対してこのコマンドを実行して集約された結果を受け取るには **GLOBAL** オプションを指定してください。

後続のコマンドを実行する対象の、現在アタッチされているメンバーを変更するには、**SET CLIENT** コマンドの **ATTACH\_MEMBER** パラメーターを使用します。このクライアント設定を有効にするには、アプリケーションを **DETACH** してから再び **ATTACH** する必要があります。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMOINT
- SYSMON

### 必要な接続

インスタンス。インスタンス接続が存在しない場合は、デフォルトのインスタンス接続が作成されます。

リモート・インスタンス、または別のローカル・インスタンスの設定値を表示するには、まず最初にそのインスタンスとアタッチすることが必要です。

### コマンド構文

```

▶▶ GET MONITOR SWITCHES [AT MEMBER member-number] [GLOBAL]

```

## コマンド・パラメーター

**AT MEMBER** *member-number*

モニター・スイッチの状況を表示するメンバーを指定します。

**GLOBAL**

すべてのメンバーからモニター・スイッチの状況を戻します。

## 例

以下に示すのは、**GET MONITOR SWITCHES** の出力例です。

Monitor Recording Switches

```
Switch list for member 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
```

## 使用上の注意

記録スイッチ **TIMESTAMP** はデフォルトではオンですが、**UPDATE MONITOR SWITCHES** を使用してオフに切り替えることもできます。このスイッチがオンのとき、システムはタイム・スタンプ・モニター・エレメントについての情報を収集する際にタイム・スタンプ呼び出しを出します。

記録スイッチ **TIMESTAMP** はデフォルトではオンですが、**UPDATE MONITOR SWITCHES** を使用してオフに切り替えることもできます。このスイッチがオフの場合、このコマンドはこのスイッチがオフになった時点のタイム・スタンプも表示します。このスイッチがオンのとき、システムはタイム・スタンプ・モニター・エレメントについての情報を収集する際にタイム・スタンプ呼び出しを出します。これらのエレメントの例を以下に示します。

- **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**

## GET MONITOR SWITCHES

- `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 スイッチがオフの場合、タイム・スタンプ・オペレーティング・システム呼び出しが出されてこれらのエレメントを判別することはありません。これらのエレメントにはゼロが含まれることとなります。CPU 使用率が 100% に近づくと、このスイッチのオフが重要になります。これが起きた場合、タイム・スタンプを出すために必要な CPU 時間は急激に増加します。

### 互換性

以前のバージョンとの互換性:

- **DB2\_ENFORCE\_MEMBER\_SYNTAX** レジストリー変数が ON に設定されている場合を除き、**DBPARTITIONNUM** または **NODE** を **MEMBER** の代わりに使用できます。

## GET RECOMMENDATIONS FOR HEALTH INDICATOR

指定されたヘルス・インディケーターがモニターするデータベース・システムの局所の正常度を改善するための、推奨事項の記述を戻します。特定のオブジェクトでアラート状態にあるヘルス・インディケーターの推奨事項を取得することもできれば、特定のヘルス・インディケーターの推奨事項の完全セットを照会することもできます。

**重要:** バージョン 9.7 でヘルス・モニターが非推奨になったため、このコマンドおよび API は推奨されておらず、将来のリリースで除去される可能性があります。これはDB2 pureScale環境ではサポートされていません。詳しくは、『ヘルス・モニターが推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055045.html>) を参照してください。

### 有効範囲

このコマンドは、現在アタッチされているメンバーに対して呼び出され、デフォルトではそのメンバーについての情報のみを返します。現在アタッチされているメンバーは、デフォルトでは、コマンドが実行されるホストになります。ホストに対して複数のメンバーがある場合、現在アタッチされているメンバーは、そのホスト上の db2nodes.cfg ファイルにリストされている最初のメンバーです。

現在アタッチされているメンバーではないメンバーに対してコマンドを実行する場合、特定のメンバーに対してこのコマンドを実行するには **AT MEMBER** オプションを指定し、すべてのメンバーに対してこのコマンドを実行して集約された結果を受け取るには **GLOBAL** オプションを指定してください。

後続のコマンドを実行する対象の、現在アタッチされているメンバーを変更するには、**SET CLIENT** コマンドの **ATTACH\_MEMBER** パラメーターを使用します。このクライアント設定を有効にするには、アプリケーションを **DETACH** してから再び **ATTACH** する必要があります。

パーティション・データベースまたは DB2 pureScale 環境では、このコマンドは、db2nodes.cfg ファイルに定義されているどのメンバーからでも呼び出すことができます。**GLOBAL** パラメーターを指定していない限り、このコマンドは、そのメンバー上でのみ機能します。

### 許可

なし

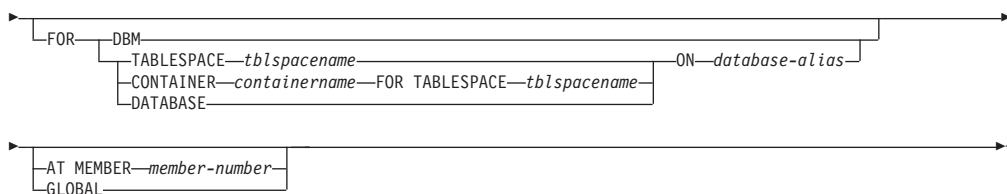
### 必要な接続

インスタンス。インスタンス接続が存在しない場合は、デフォルトのインスタンス接続が作成されます。リモート・インスタンスの推奨事項を検索するには、まず最初にそのインスタンスにアタッチする必要があります。

### コマンド構文

▶—GET RECOMMENDATIONS FOR HEALTH INDICATOR—*health-indicator-name*—▶

## GET RECOMMENDATIONS FOR HEALTH INDICATOR



### コマンド・パラメーター

#### HEALTH INDICATOR *health-indicator-name*

推奨事項を検索したいヘルス・インディケーターの名前。ヘルス・インディケーター名は、2、3 文字のオブジェクト ID に、インディケーターが測定するものを説明する名前が続きます。

**DBM** アラート状態に入ったデータベース・マネージャー・ヘルス・インディケーターの推奨事項を戻します。

#### **TABLESPACE** *tblspacename*

指定した表スペースおよびデータベースでアラート状態に入ったヘルス・インディケーターの推奨事項を戻します。

#### **CONTAINER** *containername*

指定した表スペースおよびデータベースの指定したコンテナでアラート状態に入ったヘルス・インディケーターの推奨事項を戻します。

#### **DATABASE**

指定したデータベースでアラート状態に入ったヘルス・インディケーターの推奨事項を戻します。

#### **ON** *database-alias*

データベースを指定します。

#### **AT MEMBER** *member-number*

ヘルス・インディケーターがアラート状態に入ったメンバー番号を指定します。メンバー番号を指定せず、**GLOBAL** も指定しない場合、このコマンドは、現在接続しているメンバーの情報を戻します。

#### **GLOBAL**

すべてのメンバーで、指定したヘルス・インディケーターの推奨事項を検索します。別々のメンバーに関する推奨事項が同じ場合は、それらの推奨事項が、影響を受けるメンバーに関するヘルス・インディケーターを解決するための 1 セットの推奨事項として戻されます。

### 例

```
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:

## GET RECOMMENDATIONS FOR HEALTH INDICATOR

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 required utilization rate. For example, if your required 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%).

Execute the following commands at the DB2 server:

```
CONNECT TO SAMPLE;  
UPDATE DB CFG USING DBHEAP 149333;  
CONNECT RESET;  
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 about 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.

### 使用上の注意

**GET RECOMMENDATIONS FOR HEALTH INDICATOR** コマンドは、以下の 2 つの方法で使用できます。

- すべての可能な推奨事項の情報リストを取得するために、ヘルス・インディケーターだけを指定する方法。オブジェクトを指定しない場合、このコマンドは、そのヘルス・インディケーターのアラートを解決するためのすべての推奨事項の完全リストを戻します。
- 特定のオブジェクトに関する特定のアラートを解決するために、オブジェクトを指定する方法。オブジェクト (データベースや表スペースなど) を指定した場合、戻される推奨事項は、指定されているオブジェクトに対するアラートに固有の情報になります。その場合、推奨事項はより具体的になり、アラートの解決に関するより多くの情報を含むようになります。指定したヘルス・インディケーターが、指定したオブジェクトでアラート状態になっていない場合、推奨事項は戻されません。

**GET RECOMMENDATIONS FOR HEALTH INDICATOR** コマンドが、データ・パーティション表のデータおよび索引を再編成するための推奨値を戻す場合、その推奨値は単に表レベルのものであり、表の何らかの個別のデータ・パーティションに特定されたものではありません。DB2 バージョン 9.7 フィックスパック 1 以降では、**REORG INDEXES/TABLE** コマンドまたは db2Reorg API を使用して、特定のデータ・パーティションのデータまたはパーティション索引を再編成できます。データ・パーティション表の特定のデータ・パーティションのみを再編成する必要があるかどうか判断するために、**REORGCHK** コマンドを使用して、データ・パーティション表のデータ・パーティションについての統計および推奨値を取得します。**ON DATA PARTITION** 節と共に **REORG TABLE** または **REORG INDEXES ALL** コマンドを使用し

## GET RECOMMENDATIONS FOR HEALTH INDICATOR

て、特定のデータ・パーティションのデータまたはパーティション索引を再編成します。



## GET ROUTINE

指定された SQL ルーチンのルーチン SQL アーカイブ (SAR) ファイルを検索します。

### 許可

SYSFUN.GET\_ROUTINE\_SAR プロシージャに対する EXECUTE 特権。

### 必要な接続

データベース。暗黙接続が可能な場合には、デフォルト・データベースへの接続が確立されます。

### コマンド構文

```

▶▶ GET ROUTINE INTO file_name FROM SPECIFIC PROCEDURE routine_name ▶▶
HIDE BODY

```

### コマンド・パラメーター

**INTO** *file\_name*

ルーチン SQL アーカイブ (SAR) が保管されているファイルの名前。

**FROM**

検索するルーチンの仕様の開始を示します。

**SPECIFIC**

指定されたルーチン名を特定の名前として与えます。

**PROCEDURE**

ルーチンは SQL プロシージャです。

*routine\_name*

プロシージャの名前。 **SPECIFIC** が指定された場合、これは特定の名前のプロシージャになります。名前がスキーマ名で修飾されていない場合には、CURRENT SCHEMA がルーチンのスキーマ名として使用されます。*routine-name* は、SQL プロシージャとして定義された既存のプロシージャでなければなりません。

**HIDE BODY**

カタログからルーチン・テキストが抽出されるときに、ルーチンの本体が空の本体に置き換えられるように指定します。

これは、テキストにのみ影響を与え、コンパイル済みコードには影響を与えません。

### 例

```
GET ROUTINE INTO procs/proc1.sar FROM PROCEDURE myapp1.proc1;
```

### 使用上の注意

**GET ROUTINE** または **PUT ROUTINE** 操作 (またはそれに対応するプロシージャ) が正常に実行できない場合、エラー (SQLSTATE 38000)、および失敗の原因に関する情報を示す診断テキストを毎回戻します。例えば、**GET ROUTINE** に指定されたプロシージャ名が SQL プロシージャを識別しない場合、"-204, 42704" という診断テキストが戻ります。"-204" は SQLCODE、"42704" は SQLSTATE で、それぞれ問題の原因を示します。この例の SQLCODE および SQLSTATE は、**GET ROUTINE** コマンドに指定されたプロシージャ名が未定義であることを示しています。

---

## GET SNAPSHOT

状況情報を収集して、ユーザー用に出力を形式設定します。戻された情報は、コマンドを発行した時点でのデータベース・マネージャー操作状況のスナップショットを表しています。

### 有効範囲

このコマンドは、現在アタッチされているメンバーに対して呼び出され、デフォルトではそのメンバーについての情報のみを返します。現在アタッチされているメンバーは、デフォルトでは、コマンドが実行されるホストになります。ホストに対して複数のメンバーがある場合、現在アタッチされているメンバーは、そのホスト上の `db2nodes.cfg` ファイルにリストされている最初のメンバーです。

現在アタッチされているメンバーではないメンバーに対してコマンドを実行する場合、特定のメンバーに対してこのコマンドを実行するには **AT MEMBER** オプションを指定し、すべてのメンバーに対してこのコマンドを実行して集約された結果を受け取るには **GLOBAL** オプションを指定してください。

後続のコマンドを実行する対象の、現在アタッチされているメンバーを変更するには、**SET CLIENT** コマンドの **ATTACH\_MEMBER** パラメーターを使用します。このクライアント設定を有効にするには、アプリケーションを **DETACH** してから再び **ATTACH** する必要があります。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMOINT
- SYSMON

### 必要な接続

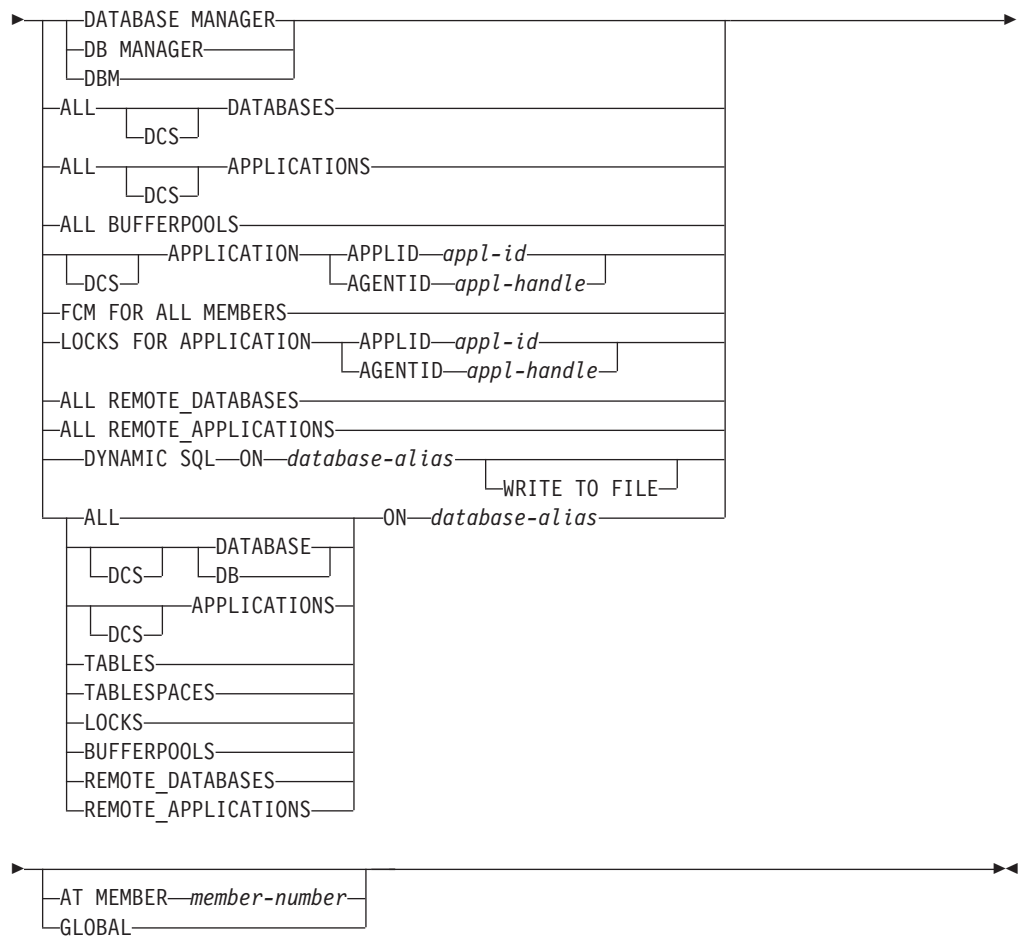
インスタンス。インスタンス接続が存在しない場合は、デフォルトのインスタンス接続が作成されます。

リモート・インスタンスのスナップショットを獲得するには、まず最初にそのインスタンスにアタッチすることが必要です。

### コマンド構文

▶▶ GET SNAPSHOT FOR →

## GET SNAPSHOT



統計を収集するために、モニター・スイッチをオンにする必要があります。

### コマンド・パラメーター

#### DATABASE MANAGER

アクティブ・データベース・マネージャー・インスタンスの統計を提供します。

#### ALL DATABASES

現行メンバー上のアクティブ・データベースすべてに関する一般統計を提供します。

#### ALL APPLICATIONS

現行メンバー上のデータベースに接続されたアクティブ・アプリケーションすべてに関する情報を提供します。

#### ALL BUFFERPOOLS

アクティブ・データベースすべてのバッファー・プール・アクティビティーに関する情報を提供します。

#### APPLICATION APPLID *appl-id*

指定された ID を持つアプリケーションの情報だけを提供します。特定のアプリケーション ID を取得するには、**LIST APPLICATIONS** コマンドを使用してください。

**APPLICATION AGENTID** *appl-handle*

指定されたアプリケーション・ハンドルを持つアプリケーションの情報だけを提供します。アプリケーション・ハンドルは 32 ビットの数字で、現在実行中のアプリケーションを固有に識別できるものです。特定のアプリケーション・ハンドルを取得するには、**LIST APPLICATIONS** コマンドを使用してください。

**FCM FOR ALL MEMBERS**

**GET SNAPSHOT** コマンドの発行対象のメンバーと、パーティション・データベース環境または DB2 pureScale 環境の他のメンバーとの間の高速コミュニケーション・マネージャー (FCM) 統計を提供します。

**LOCKS FOR APPLICATION APPLID** *appl-id*

アプリケーション ID によって識別される、指定したアプリケーションによって保留されているロックすべてに関する情報を提供します。

**LOCKS FOR APPLICATION AGENTID** *appl-handle*

アプリケーション・ハンドルによって識別される、指定したアプリケーションによって保留されているロックすべてに関する情報を提供します。

**ALL REMOTE\_DATABASES**

現行メンバー上のアクティブ・リモート・データベースすべてに関する一般統計を提供します。

**ALL REMOTE\_APPLICATIONS**

現行メンバーに接続されたアクティブ・リモート・アプリケーションすべてに関する情報を提供します。

**ALL ON** *database-alias*

指定したデータベースのアプリケーション、表、表スペース、バッファークール、およびロックすべてに関する一般統計および情報を提供します。

**DATABASE ON** *database-alias*

特定のデータベースの一般統計を提供します。

**APPLICATIONS ON** *database-alias*

特定のデータベースに接続されたアプリケーションすべてに関する情報を提供します。

**TABLES ON** *database-alias*

特定のデータベース内の表に関する情報を提供します。これには、TABLE 記録スイッチがオンになった後にアクセスのあった表だけが含まれます。

**TABLESPACES ON** *database-alias*

特定のデータベースの表スペースに関する情報を提供します。

**LOCKS ON** *database-alias*

特定のデータベースに接続された各アプリケーションが保留するロックすべてに関する情報を提供します。

**BUFFERPOOLS ON** *database-alias*

指定したデータベースのバッファークール活動に関する情報を提供します。

## GET SNAPSHOT

### REMOTE\_DATABASES ON *database-alias*

指定されたデータベースのアクティブ・リモート・データベースすべてに関する一般統計を提供します。

### REMOTE\_APPLICATIONS ON *database-alias*

指定されたデータベースのリモート・アプリケーションに関する情報を提供します。

### DYNAMIC SQL ON *database-alias*

データベースに対して SQL ステートメント・キャッシュの内容のスナップショットを戻します。

### WRITE TO FILE

スナップショットの結果が、サーバーでファイルに保管されるとともに、クライアントに戻されることを指定します。このコマンドは、データベース接続でのみ有効です。その後スナップショット・データは、表関数 `SYSFUN.SQLCACHE_SNAPSHOT` を介して、呼び出しが行われた同じ接続で照会することができます。

**DCS** 指定された節に従って、このキーワードは以下のものに関する統計を要求します。

- **DB2 Connect** ゲートウェイで現在実行されている特定の DCS アプリケーション
- すべての DCS アプリケーション
- 特定の DCS データベースに現在接続されているすべての DCS アプリケーション
- 特定の DCS データベース
- すべての DCS データベース

### AT MEMBER *member-number*

指定されたメンバーの結果を戻します。

**GLOBAL** すべてのメンバーのスナップショット結果を戻します。

## 例

- データベース・マネージャーに関するスナップショット情報を要求するには、次のコマンドを発行します。

```
get snapshot for database manager
```

以下は、上記のコマンドからリストされたサンプル出力です。

### Database Manager Snapshot

```
Node name                =
Node type                 = Enterprise Server Edition with local and remote clients
Instance name            = DB2
Number of members in DB2 instance = 1
Database manager status  = Active

Product name              = DB2 v9.5.0.535
Service level             = s070101 (NT32)

Private Sort heap allocated = 0
Private Sort heap high water mark = 0
Post threshold sorts      = Not Collected
Piped sorts requested      = 0
```

```

Piped sorts accepted = 0

Start Database Manager timestamp = 01/10/2007 15:18:36.241035
Last reset timestamp =
Snapshot timestamp = 01/10/2007 15:28:26.989789

Remote connections to db manager = 3
Remote connections executing in db manager = 0
Local connections = 1
Local connections executing in db manager = 0
Active local databases = 1

High water mark for agents registered = 0
Agents registered = 8
Idle agents = 0

Committed private Memory (Bytes) = 8912896

Switch list for member number 0
Buffer Pool Activity Information (BUFFERPOOL) = OFF
Lock Information (LOCK) = ON 01/10/2007 15:22:43.145437
Sorting Information (SORT) = OFF
SQL Statement Information (STATEMENT) = OFF
Table Activity Information (TABLE) = OFF
Take Timestamp Information (TIMESTAMP) = ON 01/10/2007 15:18:36.241035
Unit of Work Information (UOW) = OFF

Agents assigned from pool = 3
Agents created from empty pool = 11
Agents stolen from another application = 0
High water mark for coordinating agents = 9
Hash joins after heap threshold exceeded = 0
OLAP functions after heap threshold exceeded = 0

Total number of gateway connections = 0
Current number of gateway connections = 0
Gateway connections waiting for host reply = 0
Gateway connections waiting for client request = 0
Gateway connection pool agents stolen = 0

Node FCM information corresponds to = 1
Free FCM buffers = 13425
Total FCM buffers = 13425
Free FCM buffers low water mark = 13420
Maximum number of FCM buffers = 28640
Free FCM channels = 8036
Total FCM channels = 8055
Free FCM channels low water mark = 8036
Maximum number of FCM channels = 28640
Number of FCM nodes = 3

Node      Total Buffers      Total Buffers      Connection
Number    Sent                Received           Status
-----
          1                1                1            Active
          2                1                0            Active
         10                1                0            Active

Node FCM information corresponds to = 2
Free FCM buffers = 13425
Total FCM buffers = 13425
Free FCM buffers low water mark = 13420
Maximum number of FCM buffers = 28640
Free FCM channels = 8036
Total FCM channels = 8055
Free FCM channels low water mark = 8036
Maximum number of FCM channels = 28640

```

## GET SNAPSHOT

Number of FCM nodes = 3

| Node Number | Total Buffers Sent | Total Buffers Received | Connection Status |
|-------------|--------------------|------------------------|-------------------|
| 1           | 0                  | 1                      | Active            |
| 2           | 1                  | 1                      | Active            |
| 10          | 0                  | 0                      | Active            |

Node FCM information corresponds to = 10  
Free FCM buffers = 13425  
Total FCM buffers = 13425  
Free FCM buffers low water mark = 13420  
Maximum number of FCM buffers = 28640  
Free FCM channels = 8036  
Total FCM channels = 8055  
Free FCM channels low water mark = 8036  
Maximum number of FCM channels = 28640  
Number of FCM nodes = 3

| Node Number | Total Buffers Sent | Total Buffers Received | Connection Status |
|-------------|--------------------|------------------------|-------------------|
| 1           | 0                  | 1                      | Active            |
| 2           | 0                  | 0                      | Active            |
| 10          | 1                  | 1                      | Active            |

Memory usage for database manager:

Node number = 0  
Memory Pool Type = Other Memory  
Current size (bytes) = 11534336  
High water mark (bytes) = 11599872  
Configured size (bytes) = 34275328

Node number = 0  
Memory Pool Type = Database Monitor Heap  
Current size (bytes) = 65536  
High water mark (bytes) = 65536  
Configured size (bytes) = 327680

Node number = 0  
Memory Pool Type = FCMBP Heap  
Current size (bytes) = 655360  
High water mark (bytes) = 655360  
Configured size (bytes) = 851968

- エージェント ID 29 のアプリケーションに関するスナップショット情報を要求するには次のようにします。

```
get snapshot for application agentid 29
```

以下は、上記のコマンドからリストされたサンプル出力で、ロック・スイッチとステートメント・モニター・スイッチが ON になっていることを想定していません。

### Application Snapshot

```
Application handle = 29  
Application status = Lock-wait  
Status change time = Not Collected  
Application code page = 819  
Application country/region code = 1  
DUOW correlation token = *LOCAL.jwr.070222182152  
Application name = db2bp  
Application ID = *LOCAL.jwr.070222182152
```



## GET SNAPSHOT

```

Sequence number = 00001
TP Monitor client user ID =
TP Monitor client workstation name =
TP Monitor client application name =
TP Monitor client accounting string =

Connection request start timestamp = 02/22/2007 13:21:52.587168
Connect request completion timestamp = 02/22/2007 13:21:53.291779
Application idle time =
CONNECT Authorization ID = JWR
Client login ID = jwr
Configuration NNAME of client = gilera
Client database manager product ID = SQL09050
Process ID of client application = 843852
Platform of client application = AIX 64BIT
Communication protocol of client = Local Client

Inbound communication address = *LOCAL.jwr

Database name = SAMPLE
Database path = /home/jwr/jwr/NODE0000/SQL00001/
Client database alias = SAMPLE
Input database alias =
Last reset timestamp =
Snapshot timestamp = 02/22/2007 13:22:39.766300
Authorization level granted =
  User authority:
    DBADM authority
    CREATETAB authority
    BINDADD authority
    CONNECT authority
    CREATE_NOT_FENC authority
    LOAD authority
    IMPLICIT_SCHEMA authority
    CREATE_EXT_RT authority
    QUIESCE_CONN authority
  Group authority:
    SYSADM authority
    CREATETAB authority
    BINDADD authority
    CONNECT authority
    IMPLICIT_SCHEMA authority
Coordinator member number = 0
Current member number = 0
Coordinator agent process or thread ID = 1801
Current Workload ID = 1
Agents stolen = 0
Agents waiting on locks = 1
Maximum associated agents = 1
Priority at which application agents work = 0
Priority type = Dynamic

Lock timeout (seconds) = -1
Locks held by application = 4
Lock waits since connect = 1
Time application waited on locks (ms) = 20268
Deadlocks detected = 0
Lock escalations = 0
Exclusive lock escalations = 0
Number of Lock Timeouts since connected = 0
Total time UOW waited on locks (ms) = Not Collected

Total sorts = 0
Total sort time (ms) = Not Collected
Total sort overflows = 0

Buffer pool data logical reads = Not Collected

```

## GET SNAPSHOT

```
Buffer pool data physical reads           = Not Collected
Buffer pool temporary data logical reads  = Not Collected
Buffer pool temporary data physical reads = Not Collected
Buffer pool data writes                   = Not Collected
Buffer pool index logical reads           = Not Collected
Buffer pool index physical reads          = Not Collected
Buffer pool temporary index logical reads = Not Collected
Buffer pool temporary index physical reads = Not Collected
Buffer pool index writes                  = Not Collected
Buffer pool xda logical reads             = Not Collected
Buffer pool xda physical reads            = Not Collected
Buffer pool temporary xda logical reads   = Not Collected
Buffer pool temporary xda physical reads  = Not Collected
Buffer pool xda writes                    = Not Collected
Total buffer pool read time (milliseconds) = Not Collected
Total buffer pool write time (milliseconds) = Not Collected
Time waited for prefetch (ms)           = Not Collected
Unread prefetch pages                    = Not Collected
Direct reads                              = Not Collected
Direct writes                             = Not Collected
Direct read requests                      = Not Collected
Direct write requests                     = Not Collected
Direct reads elapsed time (ms)           = Not Collected
Direct write elapsed time (ms)           = Not Collected

Number of SQL requests since last commit = 3
Commit statements                         = 0
Rollback statements                       = 0
Dynamic SQL statements attempted         = 3
Static SQL statements attempted          = 0
Failed statement operations               = 0
Select SQL statements executed            = 1
Xquery statements executed                = 0
Update/Insert/Delete statements executed = 0
DDL statements executed                   = 0
Inactive stmt history memory usage (bytes) = 0
Internal automatic rebinds                = 0
Internal rows deleted                     = 0
Internal rows inserted                    = 0
Internal rows updated                     = 0
Internal commits                          = 1
Internal rollbacks                        = 0
Internal rollbacks due to deadlock        = 0
Binds/precompiles attempted              = 0
Rows deleted                              = 0
Rows inserted                             = 0
Rows updated                              = 0
Rows selected                             = 0
Rows read                                 = 95
Rows written                              = 0

UOW log space used (Bytes)                = Not Collected
Previous UOW completion timestamp         = Not Collected
Elapsed time of last completed uow (sec.ms) = Not Collected
UOW start timestamp                       = Not Collected
UOW stop timestamp                        = Not Collected
UOW completion status                     = Not Collected

Open remote cursors                       = 0
Open remote cursors with blocking         = 0
Rejected Block Remote Cursor requests     = 0
Accepted Block Remote Cursor requests     = 1
Open local cursors                        = 1
Open local cursors with blocking           = 1
Total User CPU Time used by agent (s)     = 0.019150
Total System CPU Time used by agent (s)   = 0.001795
Host execution elapsed time                = 0.012850
```

```

Package cache lookups           = 2
Package cache inserts          = 1
Application section lookups     = 3
Application section inserts    = 1
Catalog cache lookups         = 11
Catalog cache inserts         = 8
Catalog cache overflows       = 0
Catalog cache high water mark = 0

```

## Workspace Information

```

Shared high water mark         = 0
Total shared overflows        = 0
Total shared section inserts  = 0
Total shared section lookups  = 0
Private high water mark       = 0
Total private overflows      = 0
Total private section inserts = 0
Total private section lookups = 0

```

```

Most recent operation          = Fetch
Cursor name                    = SQLCUR201
Most recent operation start timestamp = 02/22/2007 13:22:19.497439
Most recent operation stop timestamp =
Agents associated with the application = 1
Number of hash joins          = 0
Number of hash loops         = 0
Number of hash join overflows = 0
Number of small hash join overflows = 0
Number of OLAP functions     = 0
Number of OLAP function overflows = 0

```

```

Statement type                 = Dynamic SQL Statement
Statement                     = Fetch
Section number                 = 201
Application creator            = NULLID
Package name                   = SQLC2G11
Consistency Token             = AAAANBX
Package Version ID            =
Cursor name                    = SQLCUR201
Statement member number       = 0
Statement start timestamp     = 02/22/2007 13:22:19.497439
Statement stop timestamp      =
Elapsed time of last completed stmt(sec.ms) = 0.000289
Total Statement user CPU time = 0.002172
Total Statement system CPU time = 0.001348
SQL compiler cost estimate in timerons = 14
SQL compiler cardinality estimate = 57
Degree of parallelism requested = 1
Number of agents working on statement = 1
Number of subagents created for statement = 1
Statement sorts               = 0
Total sort time               = 0
Sort overflows                = 0
Rows read                     = 0
Rows written                  = 0
Rows deleted                  = 0
Rows updated                  = 0
Rows inserted                 = 0
Rows fetched                  = 0
Buffer pool data logical reads = Not Collected
Buffer pool data physical reads = Not Collected
Buffer pool temporary data logical reads = Not Collected
Buffer pool temporary data physical reads = Not Collected
Buffer pool index logical reads = Not Collected

```

## GET SNAPSHOT

```
Buffer pool index logical reads          = Not Collected
Buffer pool temporary index logical reads = Not Collected
Buffer pool temporary index physical reads = Not Collected
Buffer pool xda logical reads            = Not Collected
Buffer pool xda physical reads           = Not Collected
Buffer pool temporary xda logical reads  = Not Collected
Buffer pool temporary xda physical reads = Not Collected
Blocking cursor                          = YES
Dynamic SQL statement text:
select * from org

Agent process/thread ID                   = 1801

Memory usage for application:

Memory Pool Type                         = Application Heap
Current size (bytes)                     = 65536
High water mark (bytes)                  = 65536
Configured size (bytes)                   = 1048576

Agent process/thread ID                   = 1801
Agent Lock timeout (seconds)              = -1
Memory usage for agent:

Memory Pool Type                         = Other Memory
Current size (bytes)                     = 589824
High water mark (bytes)                  = 786432
Configured size (bytes)                   = 34359738368

ID of agent holding lock                  = 34
Application ID holding lock               = *LOCAL.jwr.070222182158
Lock name                                 = 0x0002000E000000000000000000000054
Lock attributes                           = 0x00000000
Release flags                             = 0x00000001
Lock object type                          = Table
Lock mode                                 = Exclusive Lock (X)
Lock mode requested                       = Intention Share Lock (IS)
Name of tablespace holding lock           = USERSPACE1
Schema of table holding lock              = JWR
Name of table holding lock                = ORG
Data Partition Id of table holding lock   = 0
Lock wait start timestamp                 = 02/22/2007 13:22:19.497833
```

- すべてのデータベースに関するスナップショット情報を要求するには、以下のコマンドを発行します。

```
get snapshot for all databases
```

以下は、上記のコマンドからリストされたサンプル出力です。

### Database Snapshot

```
Database name                            = SAMPLE
Database path                            = C:\#DB2\#NODE0000\#SQL00001#
Input database alias                      =
Database status                           = Active
Catalog database partition number         = 0
Catalog network node name                 =
Operating system running at database server= NT
Location of the database                   = Local
First database connect timestamp          = 06/21/2007 14:46:49.771064
Last reset timestamp                      =
Last backup timestamp                     =
Snapshot timestamp                        = 06/21/2007 14:51:50.235993

Number of automatic storage paths         = 1
Automatic storage path                    = C:
```

```

Node number = 0

High water mark for connections = 6
Application connects = 4
Secondary connects total = 4
Applications connected currently = 1
Appls. executing in db manager currently = 0
Agents associated with applications = 5
Maximum agents associated with applications= 6
Maximum coordinating agents = 6

Number of Threshold Violations = 0
Locks held currently = 0
Lock waits = 0
Time database waited on locks (ms) = Not Collected
Lock list memory in use (Bytes) = 2256
Deadlocks detected = 0
Lock escalations = 0
Exclusive lock escalations = 0
Agents currently waiting on locks = 0
Lock Timeouts = 0
Number of indoubt transactions = 0

Total Private Sort heap allocated = 0
Total Shared Sort heap allocated = 0
Shared Sort heap high water mark = 0
Post threshold sorts (shared memory) = Not Collected
Total sorts = 0
Total sort time (ms) = Not Collected
Sort overflows = 0
Active sorts = 0

Buffer pool data logical reads = Not Collected
Buffer pool data physical reads = Not Collected
Buffer pool temporary data logical reads = Not Collected
Buffer pool temporary data physical reads = Not Collected
Asynchronous pool data page reads = Not Collected
Buffer pool data writes = Not Collected
Asynchronous pool data page writes = Not Collected
Buffer pool index logical reads = Not Collected
Buffer pool index physical reads = Not Collected
Buffer pool temporary index logical reads = Not Collected
Buffer pool temporary index physical reads = Not Collected
Asynchronous pool index page reads = Not Collected
Buffer pool index writes = Not Collected
Asynchronous pool index page writes = Not Collected
Buffer pool xda logical reads = Not Collected
Buffer pool xda physical reads = Not Collected
Buffer pool temporary xda logical reads = Not Collected
Buffer pool temporary xda physical reads = Not Collected
Buffer pool xda writes = Not Collected
Asynchronous pool xda page reads = Not Collected
Asynchronous pool xda page writes = Not Collected
Total buffer pool read time (milliseconds) = Not Collected
Total buffer pool write time (milliseconds)= Not Collected
Total elapsed asynchronous read time = Not Collected
Total elapsed asynchronous write time = Not Collected
Asynchronous data read requests = Not Collected
Asynchronous index read requests = Not Collected
Asynchronous xda read requests = Not Collected
No victim buffers available = Not Collected
LSN Gap cleaner triggers = Not Collected
Dirty page steal cleaner triggers = Not Collected
Dirty page threshold cleaner triggers = Not Collected
Time waited for prefetch (ms) = Not Collected
Unread prefetch pages = Not Collected
Direct reads = Not Collected

```

## GET SNAPSHOT

```
Direct writes = Not Collected
Direct read requests = Not Collected
Direct write requests = Not Collected
Direct reads elapsed time (ms) = Not Collected
Direct write elapsed time (ms) = Not Collected
Database files closed = Not Collected
Vectored IOs = Not Collected
Pages from vectored IOs = Not Collected
Block IOs = Not Collected
Pages from block IOs = Not Collected

Host execution elapsed time = Not Collected

Commit statements attempted = 0
Rollback statements attempted = 0
Dynamic statements attempted = 6
Static statements attempted = 3
Failed statement operations = 0
Select SQL statements executed = 0
Xquery statements executed = 0
Update/Insert/Delete statements executed = 0
DDL statements executed = 0
Inactive stmt history memory usage (bytes) = 0

Internal automatic rebinds = 0
Internal rows deleted = 0
Internal rows inserted = 0
Internal rows updated = 0
Internal commits = 6
Internal rollbacks = 0
Internal rollbacks due to deadlock = 0
Number of MDC table blocks pending cleanup = 0

Rows deleted = 0
Rows inserted = 0
Rows updated = 0
Rows selected = 0
Rows read = 98
Binds/precompiles attempted = 0

Log space available to the database (Bytes)= 20400000
Log space used by the database (Bytes) = 0
Maximum secondary log space used (Bytes) = 0
Maximum total log space used (Bytes) = 0
Secondary logs allocated currently = 0
Log pages read = 0
Log read time (sec.ns) = 0.000000004
Log pages written = 0
Log write time (sec.ns) = 0.000000004
Number write log IOs = 0
Number read log IOs = 0
Number partial page log IOs = 0
Number log buffer full = 0
Log data found in buffer = 0
Appl id holding the oldest transaction = 93
Log to be redone for recovery (Bytes) = 0
Log accounted for by dirty pages (Bytes) = 0

Node number = 0
File number of first active log = 0
File number of last active log = 2
File number of current active log = 0
File number of log being archived = Not applicable

Package cache lookups = 6
Package cache inserts = 0
Package cache overflows = 0
```

```

Package cache high water mark (Bytes) = 196608
Application section lookups           = 6
Application section inserts           = 0

Catalog cache lookups                 = 37
Catalog cache inserts                  = 10
Catalog cache overflows                = 0
Catalog cache high water mark         = 65536
Catalog cache statistics size         = 0

Workspace Information

Shared high water mark                 = 0
Corresponding shared overflows        = 0
Total shared section inserts          = 0
Total shared section lookups          = 0
Private high water mark                = 0
Corresponding private overflows       = 0
Total private section inserts         = 0
Total private section lookups         = 0

Number of hash joins                  = 0
Number of hash loops                  = 0
Number of hash join overflows         = 0
Number of small hash join overflows   = 0
Post threshold hash joins (shared memory) = 0
Active hash joins                     = 0

Number of OLAP functions               = 0
Number of OLAP function overflows     = 0
Active OLAP functions                 = 0

Statistic fabrications                 = Not Collected
Synchronous runstats                  = Not Collected
Asynchronous runstats                 = Not Collected
Total statistic fabrication time (milliseconds) = Not Collected
Total synchronous runstats time (milliseconds) = Not Collected

```

## Memory usage for database:

```

Node number = 0
Memory Pool Type = Backup/Restore/Util Heap
Current size (bytes) = 65536
High water mark (bytes) = 65536
Configured size (bytes) = 20512768

Node number = 0
Memory Pool Type = Package Cache Heap
Current size (bytes) = 196608
High water mark (bytes) = 196608
Configured size (bytes) = 402653184

Node number = 0
Memory Pool Type = Other Memory
Current size (bytes) = 131072
High water mark (bytes) = 131072
Configured size (bytes) = 20971520

Node number = 0
Memory Pool Type = Catalog Cache Heap
Current size (bytes) = 65536
High water mark (bytes) = 65536
Configured size (bytes) = 402653184

Node number = 0
Memory Pool Type = Buffer Pool Heap

```

## GET SNAPSHOT

```
Secondary ID = 1
Current size (bytes) = 2424832
High water mark (bytes) = 2424832
Configured size (bytes) = 402653184

Node number = 0
Memory Pool Type = Buffer Pool Heap
Secondary ID = System 32k buffer pool
Current size (bytes) = 851968
High water mark (bytes) = 851968
Configured size (bytes) = 402653184

Node number = 0
Memory Pool Type = Buffer Pool Heap
Secondary ID = System 16k buffer pool
Current size (bytes) = 589824
High water mark (bytes) = 589824
Configured size (bytes) = 402653184

Node number = 0
Memory Pool Type = Buffer Pool Heap
Secondary ID = System 8k buffer pool
Current size (bytes) = 458752
High water mark (bytes) = 458752
Configured size (bytes) = 402653184

Node number = 0
Memory Pool Type = Buffer Pool Heap
Secondary ID = System 4k buffer pool
Current size (bytes) = 393216
High water mark (bytes) = 393216
Configured size (bytes) = 402653184

Node number = 0
Memory Pool Type = Shared Sort Heap
Current size (bytes) = 0
High water mark (bytes) = 0
Configured size (bytes) = 20512768

Node number = 0
Memory Pool Type = Lock Manager Heap
Current size (bytes) = 327680
High water mark (bytes) = 327680
Configured size (bytes) = 393216

Node number = 0
Memory Pool Type = Database Heap
Current size (bytes) = 10551296
High water mark (bytes) = 10551296
Configured size (bytes) = 12582912

Node number = 0
Memory Pool Type = Application Heap
Secondary ID = 97
Current size (bytes) = 65536
High water mark (bytes) = 65536
Configured size (bytes) = 1048576

Node number = 0
Memory Pool Type = Application Heap
Secondary ID = 96
Current size (bytes) = 65536
High water mark (bytes) = 65536
Configured size (bytes) = 1048576

Node number = 0
Memory Pool Type = Application Heap
```



```

Secondary ID = 95
Current size (bytes) = 65536
High water mark (bytes) = 65536
Configured size (bytes) = 1048576

Node number = 0
Memory Pool Type = Application Heap
Secondary ID = 94
Current size (bytes) = 65536
High water mark (bytes) = 65536
Configured size (bytes) = 1048576

Node number = 0
Memory Pool Type = Application Heap
Secondary ID = 93
Current size (bytes) = 65536
High water mark (bytes) = 65536
Configured size (bytes) = 1048576

Node number = 0
Memory Pool Type = Applications Shared Heap
Current size (bytes) = 65536
High water mark (bytes) = 65536
Configured size (bytes) = 20512768

```

ユーザー権限は、そのユーザーに付与されているすべての権限およびロールを表しており、グループ権限は、そのグループに付与されているすべての権限とロールを表しています。

- アプリケーション・ハンドル 765 で SAMPLE データベースと接続されている特定のアプリケーションに関するスナップショット情報を要求するには、次のコマンドを発行します。

```
get snapshot for application agentid 765
```

- SAMPLE データベースに関する動的 SQL スナップショット情報を要求するには、次のコマンドを発行します。

```
get snapshot for dynamic sql on sample
```

- 高速コミュニケーション・マネージャー (FCM) 統計を要求するには、次のコマンドを発行します。

```
get snapshot for fcm for all dbpartitionnums
```

以下は、上記のコマンドからリストされたサンプル出力です。

#### FCM Snapshot

```

Node FCM information corresponds to = 1
Free FCM buffers = 10740
Total FCM buffers = 10740
Free FCM buffers low water mark = 10740
Maximum number of FCM buffers = 28640
Free FCM channels = 6265
Total FCM channels = 6265
Free FCM channels low water mark = 6265
Maximum number of FCM channels = 28640
Snapshot timestamp = 02/17/2010 15:54:57.094901
Number of FCM nodes = 3

```

| Node Number | Total Buffers Sent | Total Buffers Received | Connection Status |
|-------------|--------------------|------------------------|-------------------|
| -----       | -----              | -----                  | -----             |

|    |   |   |        |
|----|---|---|--------|
| 1  | 2 | 2 | Active |
| 2  | 1 | 1 | Active |
| 10 | 1 | 1 | Active |

### 使用上の注意

- データベースに対して書き込みサスペンドが ON になっている場合、書き込みサスペンドが OFF になるまで、そのデータベースに対してスナップショットを発行することはできません。書き込みサスペンドが ON になっているデータベースに対してスナップショットを発行すると、診断プローブが **db2diag** ログ・ファイルに書き込まれ、そのデータベースはスキップされます。
- リモート・インスタンス (または別のローカル・インスタンス) からスナップショットを獲得するには、まず最初にそのインスタンスとアタッチすることが必要です。別のインスタンスに存在するデータベースの別名が指定されている場合には、エラー・メッセージが戻されます。
- 統計を取得するには、データベース・システム・モニター・スイッチが ON であることが必要です。記録スイッチ **TIMESTAMP** が OFF に設定されている場合、タイム・スタンプに関連したエレメントは "Not Collected" を報告します。
- 以下の条件のいずれかが真の場合には、表情報を要求してもデータは返ってきません。
  - **TABLE** 記録スイッチがオフである。
  - スイッチをオンにして以来、アクセスのあった表がない。
  - 最後に **RESET MONITOR** コマンドを発行して以来、アクセスのあった表がない。

しかし、**REORG TABLE** が実行中であるか、またはこの期間中に実行された場合、一部の情報は戻されますが一部のフィールドは表示されません。パーティション表の場合は、再編成された各データ・パーティションに関する情報が戻されません。

- すべてのメンバーからのスナップショット情報 (すべてのメンバーに関する集約結果とは違う) を取得するには、スナップショット管理ビューを使用する必要があります。
- パーティション・データベース環境または **DB2 pureScale** 環境で、コマンドを **GLOBAL** オプションと共に指定すると、「High water mark for connections」パラメーターの値が戻されます。これは、すべてのメンバーでの個々の最高水準点の合計ではなく、すべてのメンバー間で最大の、接続の最高水準点を表します。例:
  - メンバー 1 には、現在 5 つのアプリケーションが接続されていて、接続の最高水準点は 5 です。
  - メンバー 2 には、現在 4 つのアプリケーションが接続されていて、接続の最高水準点は 6 です。

前の例では、「High water mark for connections」値は 6 で、「Applications connected currently」値は 9 です。

- ストレージ・グループを使用している場合、スナップショット・モニターはデフォルトのストレージ・グループのみの情報を報告します。すべてのデータベース・ストレージ・グループのストレージ・パスに関する情報を表示するには、**ADMIN\_GET\_STORAGE\_PATHS** 表関数を使用してください。

## 互換性

以前のバージョンとの互換性:

- **DB2\_ENFORCE\_MEMBER\_SYNTAX** レジストリー変数が ON に設定されている場合を除き、**DBPARTITIONNUM** または **NODE** を **MEMBER** の代わりに使用できます。
- **DB2\_ENFORCE\_MEMBER\_SYNTAX** レジストリー変数が ON に設定されている場合を除き、**DBPARTITIONNUMS** または **NODES** を **MEMBERS** の代わりに使用できます。
- バージョン 9.5 の新しいレジストリー変数 **DB2\_SYSTEM\_MONITOR\_SETTINGS** は、Linux 上での CPU 使用量のモニター動作に影響を与えます。Linux 上でシステムとユーザーの両方の CPU 使用時間を戻す CPU 使用量読み取り方式を使用する必要がある場合、以下のアクションのいずれかを実行してください。

RHEL4 および SLES9 の Linux の場合:

```
db2set DB2_SYSTEM_MONITOR_SETTINGS=DISABLE_CPU_USAGE:FALSE
```

RHEL5 および SLES10 の Linux の場合:

```
db2set DB2_SYSTEM_MONITOR_SETTINGS=OLD_CPU_USAGE:TRUE
```

## HELP

ユーザーはインフォメーション・センターからヘルプを呼び出すことができます。

このコマンドは UNIX オペレーティング・システムでは使用できません。

### 許可

なし

### 必要な接続

なし

### コマンド構文

▶▶—HELP—◀◀

### 例

以下の例は、**HELP** コマンドの使用方法を示しています。

- db2 help

このコマンドは、*DB2* インフォメーション・センターをオープンします。インフォメーション・センターでは、*DB2* に関する情報が、作業、解説書、ブックなどに分類されています。これは、パラメーターを指定しないで **db2ic** コマンドを呼び出すのと同じです。

### 使用上の注意

コマンド行プロセッサは、コマンドが成功したかどうかを知ることができないため、エラー状態を報告できません。

## HISTORY

CLP 対話モード・セッション内で実行したコマンドの履歴を表示します。

### 有効範囲

このコマンドは、CLP 対話モードでのみ実行できます。特に CLP コマンド・モードや CLP バッチ・モードでは実行できません。

### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### REVERSE | R

コマンド履歴を逆順で表示します。つまり、実行した最新のコマンドがリストの先頭に来ます。このパラメーターを指定しない場合は、コマンドのリストを発生順で表示します。つまり、実行した最新のコマンドがリストの末尾に来ます。

*num* 最新の *num* 個のコマンドだけを表示します。このパラメーターを指定しない場合は、最大 20 個のコマンドが表示されます。ただし、表示されるコマンドの数は、コマンド履歴に保管されているコマンドの数によっても制限されます。

### 使用上の注意

1. コマンド履歴に保管できるコマンドの最大数は、**DB2\_CLP\_HISTSZ** レジストリー変数の値によって指定します。このレジストリー変数は、1 から 500 までの任意の値に設定できます。このレジストリー変数を設定しない場合や、有効範囲外の値を設定した場合は、最大 20 個のコマンドがコマンド履歴に保管されません。
2. コマンド履歴には **HISTORY** コマンドが必ずリストされるので、表示されるコマンドの最大数は、ユーザー指定の最大数よりも必ず 1 個多くなります。
3. コマンド履歴は、CLP 対話モード・セッション間で持続しません。つまり、対話モード・セッションの終了時に保存されません。
4. 複数の CLP 対話モード・セッションを同時に実行している場合、それぞれのコマンド履歴は互いに独立した形で存在します。

---

## IMPORT

外部ファイルのデータを、サポートされているファイル・フォーマットで表、階層、ビュー、またはニックネームに挿入します。 **LOAD** はより高速な代替方法です。しかしロード・ユーティリティーでは、階層レベルのデータのロードはサポートされていません。

310 ページの『インポート・ユーティリティー用のファイル・タイプ修飾子』へのクイック・リンク。

### 許可

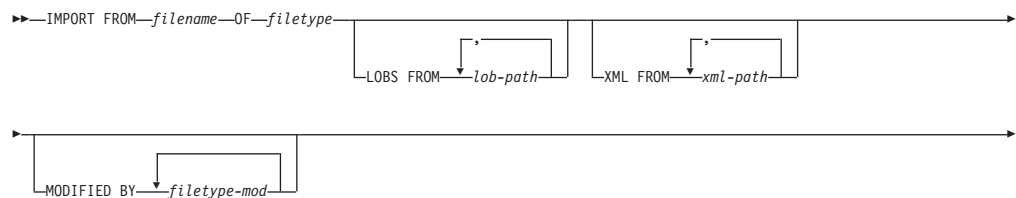
- **INSERT** オプションを使用して **IMPORT** する場合、以下の権限のいずれかが必要です。
  - DATAACCESS 権限
  - 関係するそれぞれの表、ビュー、またはニックネームに対する CONTROL 特権
  - 関係するそれぞれの表またはビューに対する INSERT および SELECT 特権
- **INSERT\_UPDATE** オプションを使用して既存の表に **IMPORT** するには、以下の権限のいずれかが必要です。
  - DATAACCESS 権限
  - 関係するそれぞれの表、ビュー、またはニックネームに対する CONTROL 特権
  - 関係するそれぞれの表またはビューに対する INSERT、SELECT、UPDATE、および DELETE 特権
- **REPLACE** または **REPLACE\_CREATE** オプションを使用して既存の表に **IMPORT** するには、以下の権限のいずれかが必要です。
  - DATAACCESS 権限
  - 表またはビューに対する CONTROL 特権
  - 表またはビューに対する INSERT、SELECT、および DELETE 特権
- **CREATE** または **REPLACE\_CREATE** オプションを使用して新規の表に **IMPORT** するには、以下の権限のいずれかが必要です。
  - DBADM 権限
  - データベースに対する CREATETAB 権限および表スペースに対する USE 特権に加えて、以下のいずれかが必要です。
    - データベースに対する IMPLICIT\_SCHEMA 権限 (表の暗黙的または明示的スキーマ名が存在しない場合)
    - スキーマに対する CREATEIN 特権 (表のスキーマ名が既存のスキーマを指す場合)
- **CREATE** または **REPLACE\_CREATE** オプションを使って、存在しない階層に **IMPORT** するには、以下の権限のいずれかが必要です。
  - DBADM 権限
  - データベースに対する CREATETAB 権限および表スペースに対する USE 特権と、以下のいずれかが必要です。

- データベースに対する `IMPLICIT_SCHEMA` 権限 (表のスキーマ名が存在しない場合)
- スキーマに対する `CREATEIN` 特権 (表のスキーマが存在する場合)
- 階層全体に対して `REPLACE_CREATE` オプションが使用されている場合は、階層内のすべての副表に対する `CONTROL` 特権
- **REPLACE** オプションを使用して既存の階層に **IMPORT** するには、以下の権限のどれかが必要です。
  - `DATAACCESS` 権限
  - 階層内のすべての副表に対する `CONTROL` 特権
- 保護列のある表にデータをインポートするには、セッション許可 ID に、その表内のすべての保護列への書き込みアクセスを許可する `LBAC` 信用証明情報が必要です。そうでない場合、インポートは失敗し、エラー (`SQLSTATE 42512`) が戻されます。
- 保護されている行のある表にデータをインポートするには、セッション許可 ID に、以下の基準を満たす `LBAC` 信用証明情報が必要です。
  - 表を保護しているセキュリティ・ポリシーの一部である
  - 書き込みアクセスに関して、セッション許可 ID に付与された挿入する行のラベル、ユーザーの `LBAC` 信用証明情報、セキュリティ・ポリシー定義、および `LBAC` 規則が行のラベルを決定します。
- **REPLACE** または **REPLACE\_CREATE** オプションが指定された場合、セッション許可 ID には、その表をドロップするための権限が付与されていなければなりません。
- ニックネームにデータをインポートするには、セッション許可 ID には指定したデータ・ソースにパススルー・モードでアクセスおよび使用する特権がなければなりません。
- 行アクセス制御がアクティブになっている表に対して **IMPORT REPLACE** を使用する場合は、その表をドロップするための権限が必要です。具体的には、表に対する `CONTROL` または `DBADM` のいずれかが必要になります。

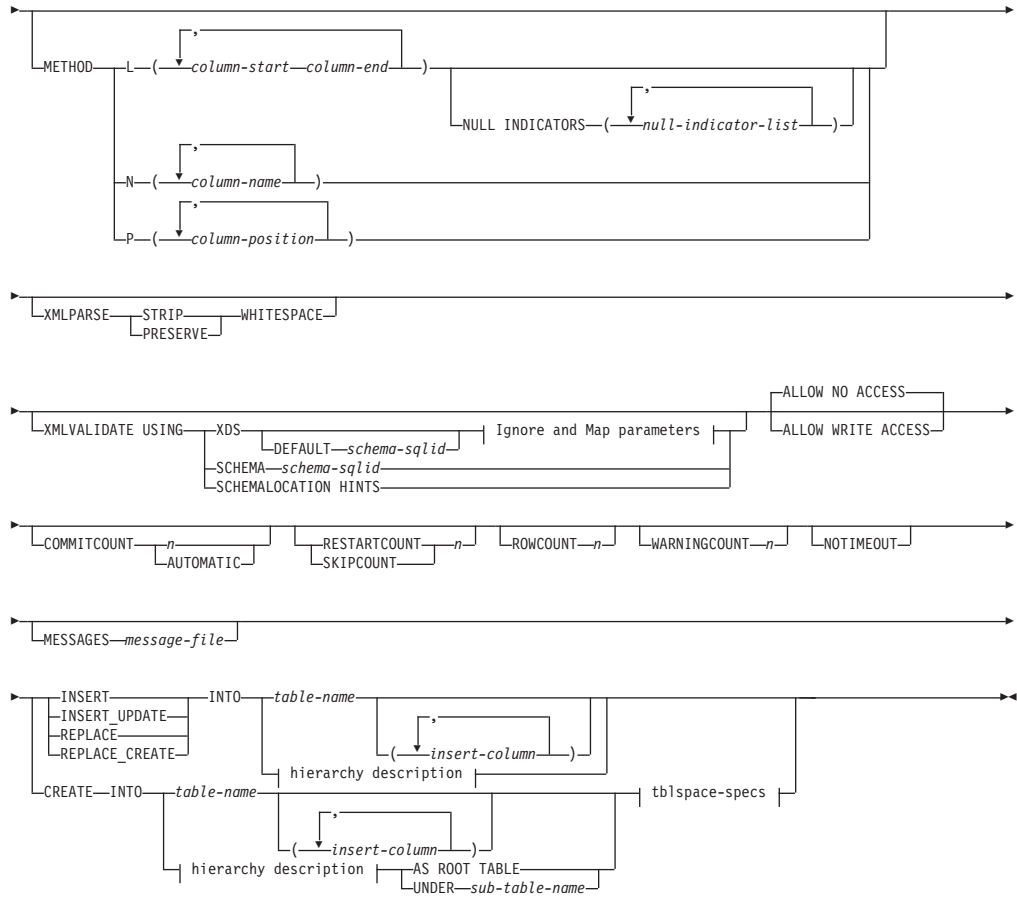
## 必要な接続

データベース。暗黙接続が可能な場合には、デフォルト・データベースへの接続が確立されます。Linux、UNIX、または Windows クライアントから Linux、UNIX、または Windows データベース・サーバーへのユーティリティ・アクセスは、`DB2 Connect` ゲートウェイまたはループバック環境を経由してではなく、エンジンを使用した直接接続でなければなりません。

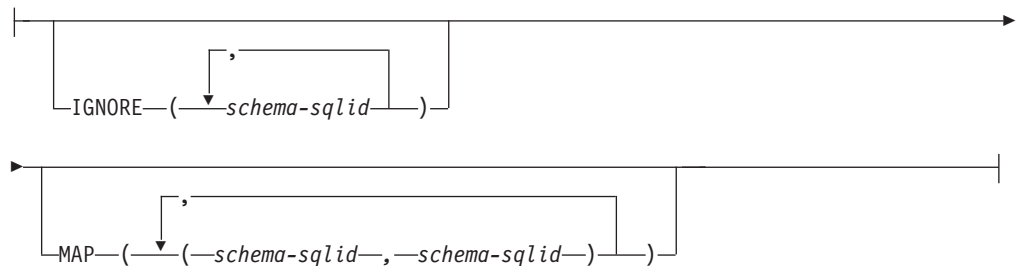
## コマンド構文



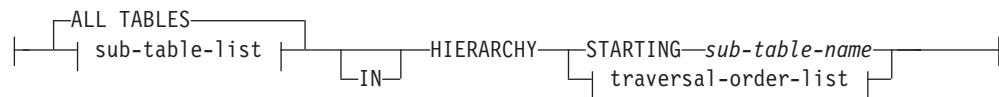
# IMPORT



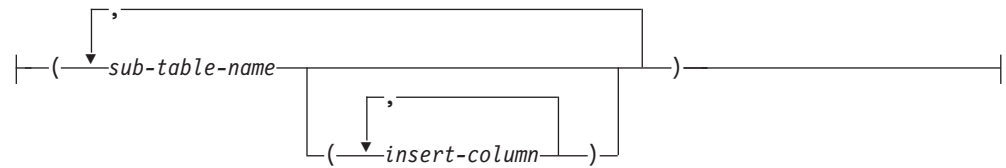
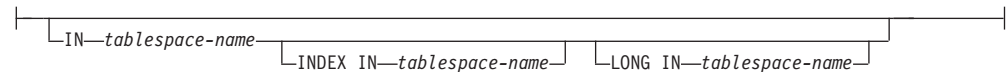
## Ignore and Map parameters:



## hierarchy description:





**sub-table-list:****traversal-order-list:****tblspace-specs:****コマンド・パラメーター****FROM *filename***

インポートするデータの入ったファイルを指定します。パスを省略すると、現行作業ディレクトリーが使用されます。

**OF *filetype***

入力ファイル内のデータのフォーマットを指定します。

- ASC (区切りなし ASCII フォーマット)
- DEL (区切り文字付き ASCII フォーマット)。さまざまなデータベース・マネージャーやファイル・マネージャー・プログラムで使用します。
- IXF (統合交換フォーマット、PC バージョン) は、DB2 によって排他的に使用されるバイナリー・フォーマットです。

**LOBS FROM *lob-path***

LOB ファイルを保管する 1 つ以上のパスを指定します。LOB データ・ファイルの名前は、メイン・データ・ファイル (ASC、DEL、または IXF) の、LOB 列にロードされる列内に保管されます。指定できるパスの最大数は 999 です。これによって、LOBSINFILE 動作が暗黙的にアクティブ化されます。

ニックネームにインポートするときには、このパラメーターは無効です。

**XML FROM *xml-path***

XML ファイルが含まれているパスを 1 つ以上指定します。

**MODIFIED BY *filetype-mod***

ファイル・タイプ修飾子オプションを指定します。310 ページの『インポート・ユーティリティー用のファイル・タイプ修飾子』を参照してください。

**METHOD**

- L** データのインポートを開始する列および終了する列の番号を指定します。列の番号は、データの行の先頭からのバイト単位のオフセットです。この番号は 1 から始まります。

注: このメソッドは、ASC ファイルの場合にのみ使用することができます、そのファイル・タイプに対してのみ有効なオプションです。

- N** インポートするデータ・ファイルの中の列の名前を指定します。これらの列名の大文字小文字の区別は、システム・カタログ内の対応する名前の大文字小文字の区別と一致しなければなりません。NULL 可能ではない各表の列には、**METHOD N** リスト内に対応する項目が必要です。例えば、データ・フィールドが F1、F2、F3、F4、F5、および F6 であり、表の列が C1 INT、C2 INT NOT NULL、C3 INT NOT NULL、および C4 INT の場合、method N (F2, F1, F4, F3) は有効な要求ですが、method N (F2, F1) は無効です。

注: この方式は、IXF ファイルの場合にのみ使用することができます。

- P** インポートする入力データ・フィールドのフィールド番号 (1 から始まる) を指定します。NULL 可能ではない各表の列には、**METHOD P** リスト内に対応する項目が必要です。例えば、データ・フィールドが F1、F2、F3、F4、F5、および F6 であり、表の列が C1 INT、C2 INT NOT NULL、C3 INT NOT NULL、および C4 INT の場合、method P (2, 1, 4, 3) は有効な要求ですが、method P (2, 1) は無効です。この方式を使用できるのは、ファイル・タイプ IXF または DEL の場合のみです。また、DEL ファイル・タイプに対しては、これが唯一有効な方式です。

method P によって指定したフィールドごとに、対応する列をアクション・ステートメント内に定義する必要があります。ただし、次の例に示すようにすべての列が対象になる場合、および最初の *x* 列がロードされる場合は必要ありません。

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

#### NULL INDICATORS *null-indicator-list*

このオプションは、**METHOD L** パラメーターを指定した場合にのみ使用できます。つまり、入力ファイルが ASC ファイルの場合です。NULL 標識リストは、コンマで区切られた正の整数のリストで、各 NULL 標識フィールドの列の番号を指定します。列の番号は、データの行の先頭からのバイト単位の、各 NULL 標識フィールドのオフセットです。NULL 標識リストには、**METHOD L** パラメーターで定義された各データ・フィールドに対する 1 つの項目がなければなりません。列の番号がゼロであることは、対応するデータ・フィールド内に必ずデータがあることを示します。

NULL 標識列中の *Y* の値は、その列データが NULL であることを指定します。NULL 標識列に *Y* 以外の文字を指定した場合は、列データが NULL ではなく、**METHOD L** オプションで指定された列データがインポートされることを指定することになります。

nullindchar ファイル・タイプ修飾子を指定した **MODIFIED BY** オプションを使用すれば、NULL 標識文字を変更することができます。

#### **XMLPARSE**

XML 文書の解析方法を指定します。このオプションが指定されていない場合、XML 文書の解析の動作は、CURRENT XMLPARSE OPTION 特殊レジスターの値によって決まります。

#### **STRIP WHITESPACE**

XML 文書の解析時に空白文字を除去することを指定します。

#### **PRESERVE WHITESPACE**

XML 文書の解析時に空白文字を除去しないことを指定します。

#### **XMLVALIDATE**

該当する場合に、XML 文書がスキーマに準拠しているかどうかの妥当性検査を実行することを指定します。

#### **USING XDS**

メイン・データ・ファイルの中で XML Data Specifier (XDS) によって指定されている XML スキーマに準拠しているかどうかについて、XML 文書の妥当性検査が実行されます。デフォルトでは、**USING XDS** 節によって **XMLVALIDATE** オプションが呼び出された場合、妥当性検査実行のために使用されるスキーマは、その XDS の SCH 属性によって決まります。XDS の中で SCH 属性が指定されていない場合、**DEFAULT** 節によってデフォルト・スキーマが指定されているのでない限り、スキーマ妥当性検査は実行されません。

**DEFAULT**、**IGNORE**、および **MAP** 節を使用することにより、スキーマ決定の動作を変更することができます。これら 3 つの節はオプションであり、相互に適用されるのではなく XDS の指定に直接適用されます。例えば、**DEFAULT** 節で指定されているためにあるスキーマが選択された場合、それが **IGNORE** 節で指定されていたとしても無視されることはありません。同じように、**MAP** 節のペアの最初の部分で指定されているためにあるスキーマが選択された場合、それが別の **MAP** 節のペアの 2 番目の部分で指定されていたとしても再びマップされることはありません。

#### **USING SCHEMA *schema-sqlid***

指定されている SQL ID の XML スキーマに準拠しているかどうかについて、XML 文書の妥当性検査が実行されます。この場合、すべての XML 列について XML Data Specifier (XDS) の SCH 属性は無視されます。

#### **USING SCHEMALOCATION HINTS**

ソース XML 文書の中で XML スキーマ・ロケーション・ヒントによって指定されているスキーマに準拠しているかどうかについて、XML 文書の妥当性検査が実行されます。その XML 文書の中に schemaLocation 属性が指定されていない場合、妥当性検査は実行されません。**USING SCHEMALOCATION HINTS** 節が指定されているなら、すべての XML 列について XML Data Specifier (XDS) の SCH 属性は無視されます。

以下のセクションに示す **XMLVALIDATE** オプションの例を参照してください。

#### **DEFAULT** *schema-sqlid*

このオプションは、**USING XDS** パラメーターを指定した場合にのみ使用できます。**DEFAULT** 節で指定されたスキーマは、インポート対象 XML 文書の XML Data Specifier (XDS) に XML スキーマを指定する SCH 属性が含まれていない場合に、妥当性検査のために使用するスキーマとなります。

**DEFAULT** 節は、**IGNORE** 節および **MAP** 節よりも優先されます。XDS が **DEFAULT** 節を満たすなら、**IGNORE** と **MAP** の指定は無視されます。

#### **IGNORE** *schema-sqlid*

このオプションは、**USING XDS** パラメーターを指定した場合にのみ使用できます。**IGNORE** 節は、SCH 属性によって指定されていても無視するスキーマとして、1 つ以上のスキーマのリストを指定します。インポートする XML 文書の XML Data Specifier の中に SCH 属性が存在し、その SCH 属性によって指定されるスキーマが無視するスキーマ・リストに含まれている場合には、インポートするその XML 文書についてスキーマ妥当性検査は実行されません。

あるスキーマが **IGNORE** 節の中で指定されている場合、**MAP** 節のスキーマ・ペアの左辺にそれを含めることはできません。

**IGNORE** 節は XDS にのみ適用されます。あるスキーマが **IGNORE** 節によって指定されていても、それが **MAP** 節によってマップされているなら、それ以降そのスキーマが無視されることはありません。

#### **MAP** *schema-sqlid*

このオプションは、**USING XDS** パラメーターを指定した場合にのみ使用できます。**MAP** 節は、インポートする各 XML 文書について XML Data Specifier (XDS) の SCH 属性によって指定されるスキーマの代わりに使用する代替スキーマを指定するのに使用します。**MAP** 節には、それぞれがあるスキーマから別のスキーマへのマッピングを表すスキーマ・ペアを 1 つ以上列挙したリストを指定します。ペア中の最初のスキーマは、XDS 内の SCH 属性によって参照されるスキーマを表します。ペア中の 2 番目のスキーマは、スキーマ妥当性検査の実行で使用する必要のあるスキーマを表します。

あるスキーマが **MAP** 節のスキーマ・ペアの左辺で指定されている場合、**IGNORE** 節でさらにそれを指定することはできません。

スキーマ・ペアのマッピングが適用されたなら、その結果は最終的なものです。マッピング操作は推移的ではないため、選択されたスキーマが、それ以降に別のスキーマ・ペアのマッピングに適用されることはありません。

スキーマを複数回マップすることはできません。つまり、複数のペアの左辺に指定することはできません。

#### **ALLOW NO ACCESS**

オフライン・モードでインポートを実行します。行の挿入の前には常に、ターゲット表に排他 (X) ロックがかけられます。これで、同時アプリケーションは表データにアクセスできなくなります。これがデフォルトのインポート動作です。

**ALLOW WRITE ACCESS**

オンライン・モードでインポートを実行します。最初の行の挿入時には、ターゲット表に意図的排他 (IX) ロックがかけられます。これで、表データへの同時の読み取りおよび書き出しアクセスが可能になります。オンライン・モードには、**REPLACE**、**CREATE**、または **REPLACE\_CREATE** インポート・オプションとの互換性はありません。オンライン・モードとバッファ挿入との連携はサポートされません。インポート操作によって挿入後のデータが定期的にコミットされるので、表ロックへのロック・エスカレーションが削減され、アクティブなログ・スペースが使い果たされることはなくなります。このようなコミットは、**COMMITCOUNT** オプションを使わなくても実行されます。各コミットごとに、インポートでは IX 表ロックが外されるので、コミットの完了後に再びロックの設定が試みられます。ニックネームにインポートするときにはこのパラメーターが必要で、有効な数値を使って **COMMITCOUNT** を指定する必要があります (AUTOMATIC は有効なオプションとは見なされません)。

**COMMITCOUNT *n* | AUTOMATIC**

*n* 個のレコードがインポートされるたびに **COMMIT** を実行します。数 *n* を指定すると、インポートでは *n* 個のレコードのインポートの後にそのつど **COMMIT** が実行されます。コンパウンド挿入を使用した場合、ユーザー指定のコミット頻度 *n* は、そのコンパウンド・カウント値に最も近い整数の倍数に切り上げられます。AUTOMATIC を指定すると、コミットの必要時期はインポート操作で内部的に判別されます。次の 2 つのうちのいずれかの理由で、このユーティリティーはコミットを行います。

- アクティブ・ログ・スペースを使いきらないようにするため。
- ロックが行レベルから表レベルにエスカレーションしないようにするため。

**ALLOW WRITE ACCESS** オプションを指定した場合に **COMMITCOUNT** オプションを指定しないと、インポート・ユーティリティーは、**COMMITCOUNT** AUTOMATIC が指定されたものとしてコミットを実行します。

アクティブ・ログ・スペースを使いきらないようにする点でのインポート操作の能力は、以下のように DB2 レジストリー変数

**DB2\_FORCE\_APP\_ON\_MAX\_LOG** の影響を受けます。

- **DB2\_FORCE\_APP\_ON\_MAX\_LOG** を FALSE にして、**COMMITCOUNT** AUTOMATIC コマンド・オプションを指定した場合、インポート・ユーティリティーは自動的にアクティブ・ログ・スペースを使い切らないようにすることができます。
- **DB2\_FORCE\_APP\_ON\_MAX\_LOG** を FALSE にして、**COMMITCOUNT *n*** コマンド・オプションを指定した場合、インポート・ユーティリティーはレコードを挿入または更新しようとして SQL0964C (トランザクション・ログがフル) を検出したら、ログ・フル状態が解決しようとしています。無条件のコミットを実行してから、レコードの挿入または更新を再試行します。これで問題が解決しない場合 (ログ・フルがデータベース上の他のアクティビティーに起因する場合など)、予期されるように **IMPORT** コマンドは失敗しますが、コミットされる行数は **COMMITCOUNT *n*** 値の倍数にならないこと



があります。インポート操作を再試行する際に既にコミットされた行を処理しないようにするには、**RESTARTCOUNT** または **SKIPCOUNT** コマンド・パラメーターを使用します。

- **DB2\_FORCE\_APP\_ON\_MAX\_LOG** が TRUE (これがデフォルトです) の場合、レコードの挿入または更新中に SQL0964C を検出すると、インポート操作は失敗します。これは、**COMMITCOUNT** AUTOMATIC と **COMMITCOUNT** *n* のどちらを指定したかに関係なく発生します。

アプリケーションは強制的にデータベースから切断され、現在の作業単位はロールバックされます。インポート操作を再試行する際に既にコミットされた行を処理しないようにするには、**RESTARTCOUNT** または **SKIPCOUNT** コマンド・パラメーターを使用します。

#### **RESTARTCOUNT** *n*

*n* + 1 の位置のレコードからインポート操作を開始するよう指定します。最初の *n* 個のレコードはスキップされます。このオプションは機能的には **SKIPCOUNT** と同等です。**RESTARTCOUNT** と **SKIPCOUNT** は相互に排他的です。

#### **SKIPCOUNT** *n*

*n* + 1 の位置のレコードからインポート操作を開始するよう指定します。最初の *n* 個のレコードはスキップされます。このオプションは機能的には **RESTARTCOUNT** と同等です。**SKIPCOUNT** と **RESTARTCOUNT** は相互に排他的です。

#### **ROWCOUNT** *n*

インポート (挿入または更新) するファイル内の物理レコードの数 *n* を指定します。ユーザーは、**SKIPCOUNT** または **RESTARTCOUNT** オプションで指示されたレコードから始めて、ファイルの *n* 行だけをインポートすることができます。**SKIPCOUNT** または **RESTARTCOUNT** オプションの指定がないと、最初の *n* 行がインポートされます。**SKIPCOUNT** *m* または **RESTARTCOUNT** *m* を指定すると、行 *m*+1 から *m*+*n* がインポートされます。コンパウンド挿入を使用した場合、ユーザー指定の **ROWCOUNT** *n* は、そのコンパウンド・カウント値に最も近い整数の倍数に切り上げられます。

#### **WARNINGCOUNT** *n*

*n* 個の警告後に、インポート操作を停止します。このパラメーターは、警告は予期されないが、正しいファイルと表が使用されていることを確認するのが望ましい場合に設定してください。インポート・ファイルまたはターゲット表が不適切に指定されると、インポート対象の各行ごとにインポート・ユーティリティーによって警告が生成され、このためにインポートが失敗する可能性があります。*n* をゼロにした場合や、このオプションを指定しない場合、発行された警告の回数に関係なくインポート操作は続行します。

#### **NOTIMEOUT**

インポート・ユーティリティーがロックの待機中にタイムアウトしないことを指定します。このオプションのほうが、**locktimeout** データベース構成パラメーターより優先されます。他のアプリケーションは影響を受けません。

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

インポート操作中に生じ得る警告およびエラー・メッセージの宛先を指定します。宛先ファイルが既に存在している場合、インポート・ユーティリティー

ーは情報を追加します。このファイルへの完全パスが指定されていない場合、このユーティリティーは現行のディレクトリーおよびデフォルトのドライブを宛先として使用します。 *message-file* を省略すると、メッセージは標準出力に書き込まれます。

**INSERT** 既存の表データを変更することなく、インポートされたデータを表に追加します。

#### **INSERT\_UPDATE**

インポートしたデータ行をターゲット表に追加するか、または主キーが一致するものがあればターゲット表の既存行を更新します。

#### **REPLACE**

データ・オブジェクトを切り捨てることによって表内の既存のデータすべてを削除してから、インポートしたデータを挿入します。表定義および索引定義は変更されません。表がない場合は、このオプションを使用できません。階層間でデータを移動する際にこのオプションを使用する場合は、階層全体に関係したデータだけが置き換えられます。副表は置き換えられません。

ニックネームにインポートするときには、このパラメーターは無効です。

このオプションでは、CREATE TABLE ステートメントの NOT LOGGED INITIALLY (NLI) 節、あるいは ALTER TABLE ステートメントの ACTIVE NOT LOGGED INITIALLY 節は考慮されません。

このオプションは、システム期間テンポラル表へのデータのインポートには使用できません。

NLI 節が呼び出される CREATE TABLE または ALTER TABLE ステートメントと同じランザクションの中で、**REPLACE** オプションの指定されたインポートが実行された場合、インポートにおいてその NLI 節は考慮されません。すべての挿入操作がログ対象となります。

#### **予備手段 1**

DELETE ステートメントを使用して表の内容を削除した後、  
INSERT ステートメントによりインポートを呼び出す

#### **予備手段 2**

表をドロップしてからそれを再作成した後、INSERT ステートメントによってインポートを呼び出す

この制限は、DB2 Universal Database™ バージョン 7 および DB2 バージョン 8 に適用されます。

#### **REPLACE\_CREATE**

**注:** **REPLACE\_CREATE** パラメーターは推奨されておらず、今後のリリースで除去される可能性があります。さらに詳しくは、『IMPORT コマンドの推奨されなくなったオプション CREATE および REPLACE\_CREATE』を参照してください。

表が既にある場合には、データ・オブジェクトを切り捨てることによって表内の既存のデータすべてを削除し、表定義や索引定義は変えることなく、インポートしたデータを挿入します。

表がまだない場合には、データベースのコード・ページで、表と索引の定義と行の内容を作成します。制約事項のリストは、『インポート済みの表の再作成』を参照してください。

このオプションは、IXF ファイルの場合にのみ使用することができます。階層間でデータを移動する際にこのオプションを使用する場合は、階層全体に関係したデータだけが置き換えられます。副表は置き換えられません。

ニックネームにインポートするときには、このパラメーターは無効です。

### **INTO** *table-name*

データのインポート先となるデータベース表を指定します。この表として、システム表、作成済み一時表、宣言済み一時表、またはサマリー表は指定できません。

以前のサーバーの場合を除き、**INSERT**、**INSERT\_UPDATE**、または **REPLACE** オプションには、完全修飾または非修飾の表名を使用しなければならないようなときでも、別名を使用することができます。修飾された表名は、*schema.tablename* の形式になります。 *schema* には、表作成時のユーザー名が入ります。

暗黙的な非表示列がデータベース表に含まれている場合には、インポート操作に非表示列のデータを含めるかどうかを指定する必要があります。非表示列のデータを含めるかどうかを示すには、以下のいずれかの方法を使用します。

- データの挿入先の列を明示的に指定する場合、*insert-column* を使用します。

```
db2 import from delfile1 of del
insert into table1 (c1, c2, c3,...)
```

- 非表示列のファイル・タイプ修飾子のいずれかを使用します。入力ファイルに非表示列のデータが含まれる場合には **implicitlyhiddeninclude** を指定し、入力ファイルに含まれない場合には **implicitlyhiddenmissing** を指定します。

```
db2 import from delfile1 of del modified by implicitlyhiddeninclude
insert into table1
```

- 暗黙的な非表示列が含まれる表をデータ移動ユーティリティーが検出した場合のデフォルト動作を設定するには、**DB2\_DMU\_DEFAULT** レジストリー変数をクライアント・サイドで使用します。

```
db2set DB2_DMU_DEFAULT=IMPLICITLYHIDDENINCLUDE
db2 import from delfile1 of del insert into table1
```

### *insert-column*

データの挿入先となる表またはビュー内の列名を指定します。

### **ALL TABLES**

階層専用の暗黙キーワード。階層をインポートする場合、走査順序で指定されるすべての表をインポートすることがデフォルトです。

### *sub-table-list*

型付き表で **INSERT** または **INSERT\_UPDATE** オプションを指定した場合、データのインポート先副表を指定するために副表名のリストが使われます。

### **HIERARCHY**

階層データをインポートするよう指定します。



**STARTING** *sub-table-name*

階層専用キーワード。 *sub-table-name* から始まるデフォルト順を要求します。 PC/IXF ファイルの場合、デフォルト順は入力ファイルに保管されている順です。 PC/IXF ファイル・フォーマットの場合、デフォルト順は有効な唯一の順序です。

*traversal-order-list*

型付き表で **INSERT**、**INSERT\_UPDATE**、または **REPLACE** オプションを指定した場合、インポートする階層内の副表の横断順序を指定するために副表名のリストを使います。

**CREATE**

**注:** **CREATE** パラメーターは推奨されておらず、今後のリリースで除去される可能性があります。さらに詳しくは、『IMPORT コマンドの推奨されなくなったオプション **CREATE** および **REPLACE\_CREATE**』を参照してください。

データベースのコード・ページで表の定義と行の内容を作成します。 DB2 の表、副表、または階層からエクスポートされたデータの場合、索引も作成されます。このオプションが階層に対するものである場合に、DB2 からデータがエクスポートされると、タイプ階層も作成されます。このオプションは、IXF ファイルの場合にのみ使用することができます。

ニックネームにインポートするときには、このパラメーターは無効です。

**注:** データが MVS™ ホスト・データベースからエクスポートされたもので、ページ・サイズで計算した長さが 254 より大きい LONGVAR フィールドを含んでいる場合、**CREATE** は行が長過ぎるために失敗します。制約事項のリストは、『インポート済みの表の再作成』を参照してください。この場合、その表は手動で作成します。そして、**IMPORT** に **INSERT** を指定して呼び出すか、または **LOAD** コマンドを使用してください。

**AS ROOT TABLE**

1 つ以上の副表を、独立した表階層として作成します。

**UNDER** *sub-table-name*

1 つ以上の副表を作成する場合に親表を指定します。

**IN** *tablespace-name*

表を作成する表スペースを指定します。表スペースは存在している必要があります、REGULAR 表スペースでなければなりません。他の表スペースを指定しない場合、すべての表パーツはこの表スペースに保管されます。この節を指定しない場合、表は許可 ID によって作成された表スペース中に作成されます。何も検出されない場合、その表はデフォルト表スペースの USERSPACE1 に入れられます。USERSPACE1 がドロップされていた場合、表作成は失敗します。

**INDEX IN** *tablespace-name*

表の索引を作成する表スペースを指定します。このオプションは、**IN** 節で指定される PRIMARY 表スペースが DMS 表スペースである場合のみ使用できます。指定した表スペースは存在している必要があります、かつ REGULAR または LARGE DMS 表スペースでなければなりません。

注: どの表スペースに索引を配置するかは、表を作成するときのみ指定できます。

## LONG IN *tablespace-name*

ロング列の値 (LONG VARCHAR、LONG VARGRAPHIC、LOB データ・タイプ、またはソース・タイプとしてこれらが指定されている特殊タイプ) を保管する表スペースを指定します。このオプションは、IN 節で指定した PRIMARY 表スペースが DMS 表スペースである場合のみ使用できます。指定した表スペースは存在している必要があり、LARGE DMS 表スペースでなければなりません。

## 例

### 例 1

次に示すのは myfile.ixf から STAFF 表に情報をインポートする方法の例です。

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

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

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

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

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

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

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

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

### 例 2 (ID 列がある表へのインポート)

TABLE1 には以下の 4 つの列があります。

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

TABLE2 は TABLE1 と同じですが、C2 が GENERATED ALWAYS ID 列である点が異なります。

DATAFILE1 内のデータ・レコード (DEL フォーマット):

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

DATAFILE2 内のデータ・レコード (DEL フォーマット):

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

以下のコマンドは、DATAFILE1 で行 1 および 2 への IDENTITY 値が入力されていないので、それらの行のための IDENTITY 値を生成します。ただし、行 3 にはユーザー提供の ID 値 100 が、行 4 にはユーザー提供の ID 値 101 がそれぞれ割り当てられます。

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

DATAFILE1 を TABLE1 にインポートしてすべての行に対する IDENTITY 値を生成するには、以下のコマンドのいずれかを発行します。

```
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
```

DATAFILE2 を TABLE1 にインポートして各行に対する IDENTITY 値を生成するには、以下のコマンドのいずれかを発行します。

```
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
```

DATAFILE1 を TABLE2 に、IDENTITY 関連のファイル・タイプ修飾子を使用せずにインポートした場合、行 1 と 2 は挿入されますが、行 3 と 4 はリジェクトされます。その理由は、それらが固有の非 NULL 値を提供し、IDENTITY 列が GENERATED ALWAYS であるからです。

## XMLVALIDATE 節の使用例

### 例 1 (XMLVALIDATE USING XDS)

以下の XMLVALIDATE 節の場合、

```
XMLVALIDATE USING XDS
  IGNORE (S1.SCHEMA_A)
  MAP ((S1.SCHEMA_A, S2.SCHEMA_B))
```

S1.SCHEMA\_A の IGNORE が S1.SCHEMA\_A から S2.SCHEMA\_B への MAP と矛盾しているため構文が無効であり、インポートは失敗します。

### 例 2 (XMLVALIDATE USING XDS)

以下の XMLVALIDATE 節の場合、

```
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))
```

XML 列に以下の XDS が含まれているとすると、

```
<XDS FIL='xmlfile.001.xml' />
```

## IMPORT

使用するデフォルト・スキーマとして "S8.SCHEMA\_H" が指定されているため、ファイル "xmlfile.001.xml" に含まれる文書は、SQL ID が "S8.SCHEMA\_H" である XML スキーマを使用して妥当性検査されます。

XML 列に以下の XDS が含まれているとすると、

```
<XDS FIL='xmlfile.002.xml' OFF='10' LEN='500' SCH='S10.SCHEMA_J' />
```

使用するスキーマとして XDS で指定されているのは "S10.SCHEMA\_J" ですが、このスキーマは **IGNORE** 節に含まれているため、ファイル "xmlfile.002.xml" に含まれる文書のスキーマ妥当性検査は実行されません。文書の内容はファイル中のバイト・オフセット 10 (11 番目のバイト) にあり、その長さは 500 バイトです。

XML 列に以下の XDS が含まれているとすると、

```
<XDS FIL='xmlfile.003.xml' SCH='S6.SCHEMA_F' />
```

ファイル "xmlfile.003.xml" に含まれる文書は、SQL ID が "S3.SCHEMA\_C" である XML スキーマを使用して妥当性検査されます。これは、**MAP** 節でスキーマ "S6.SCHEMA\_F" をスキーマ "S3.SCHEMA\_C" にマップすることが指定されているためです。それ以外のマッピングは実行されないため、この場合、スキーマ "S3.SCHEMA\_C" からスキーマ "S5.SCHEMA\_E" へのマッピングは適用されないことに注意してください。

XML 列に以下の XDS が含まれているとすると、

```
<XDS FIL='xmlfile.004.xml' SCH='S11.SCHEMA_K' />
```

ファイル "xmlfile.004.xml" に含まれる文書は、SQL ID が "S11.SCHEMA\_K" である XML スキーマを使用して妥当性検査されます。この場合、**DEFAULT**、**IGNORE**、または **MAP** の指定のいずれも適用されないことに注意してください。

### 例 3 (XMLVALIDATE USING XDS)

以下の **XMLVALIDATE** 節の場合、

```
XMLVALIDATE USING XDS
  DEFAULT S1.SCHEMA_A
  IGNORE (S1.SCHEMA_A)
```

XML 列に以下の XDS が含まれているとすると、

```
<XDS FIL='xmlfile.001.xml' />
```

使用するデフォルト・スキーマとして "S1.SCHEMA\_1" が指定されているため、ファイル "xmlfile.001.xml" に含まれる文書は、SQL ID が "S1.SCHEMA\_A" である XML スキーマを使用して妥当性検査されます。

XML 列に以下の XDS が含まれているとすると、

```
<XDS FIL='xmlfile.002.xml' SCH='S1.SCHEMA_A' />
```

使用するスキーマとして XDS で指定されているのは "S1.SCHEMA\_A" ですが、このスキーマは **IGNORE** 節に含まれているため、ファイル "xmlfile.002" に含まれる文書のスキーマ妥当性検査は実行されません。

### 例 4 (XMLVALIDATE USING XDS)

以下の **XMLVALIDATE** 節の場合、

```
XMLVALIDATE USING XDS
  DEFAULT S1.SCHEMA_A
  MAP ((S1.SCHEMA_A, S2.SCHEMA_B), (S2.SCHEMA_B, S1.SCHEMA_A))
```

XML 列に以下の XDS が含まれているとすると、

```
<XDS FIL='xmlfile.001.xml' />
```

使用するデフォルト・スキーマとして "S1.SCHEMA\_1" が指定されているため、ファイル "xmlfile.001.xml" に含まれる文書は、SQL ID が "S1.SCHEMA\_A" である XML スキーマを使用して妥当性検査されます。**DEFAULT** 節が適用されたため、それ以降 **MAP** 節は適用されないことに注意してください。したがって、この場合、スキーマ "S1.SCHEMA\_A" からスキーマ "S2.SCHEMA\_B" へのマッピングは適用されません。

XML 列に以下の XDS が含まれているとすると、

```
<XDS FIL='xmlfile.002.xml' SCH='S1.SCHEMA_A' />
```

ファイル "xmlfile.002.xml" に含まれる文書は、SQL ID が "S2.SCHEMA\_B" である XML スキーマを使用して妥当性検査されます。これは、**MAP** 節でスキーマ "S1.SCHEMA\_A" をスキーマ "S2.SCHEMA\_B" にマップすることが指定されているためです。それ以外のマッピングは実行されないため、この場合、スキーマ "S2.SCHEMA\_B" からスキーマ "S1.SCHEMA\_A" へのマッピングは適用されないことに注意してください。

XML 列に以下の XDS が含まれているとすると、

```
<XDS FIL='xmlfile.003.xml' SCH='S2.SCHEMA_B' />
```

ファイル "xmlfile.003.xml" に含まれる文書は、SQL ID が "S1.SCHEMA\_A" である XML スキーマを使用して妥当性検査されます。これは、**MAP** 節でスキーマ "S2.SCHEMA\_B" をスキーマ "S1.SCHEMA\_A" にマップすることが指定されているためです。それ以外のマッピングは実行されないため、この場合、スキーマ "S1.SCHEMA\_A" からスキーマ "S2.SCHEMA\_B" へのマッピングは適用されないことに注意してください。

#### 例 5 (XMLVALIDATE USING SCHEMA)

以下の **XMLVALIDATE** 節の場合、

```
XMLVALIDATE USING SCHEMA S2.SCHEMA_B
```

XML 列に以下の XDS が含まれているとすると、

```
<XDS FIL='xmlfile.001.xml' />
```

ファイル xmlfile.001.xml に含まれる文書は、SQL ID が "S2.SCHEMA\_B" の XML スキーマを使用して妥当性検査されます。

XML 列に以下の XDS が含まれているとすると、

```
<XDS FIL='xmlfile.002.xml' SCH='S1.SCHEMA_A' />
```

ファイル "xmlfile.002.xml" に含まれる文書は、SQL ID が "S2.SCHEMA\_B" の XML スキーマを使用して妥当性検査されます。妥当性検査は **USING SCHEMA** 節で指定されたスキーマを使用して実行されるため、SCH 属性は無視されることに注意してください。

#### 例 6 (XMLVALIDATE USING SCHEMALOCATION HINTS)

XML 列に以下の XDS が含まれているとすると、

```
<XDS FIL='xmlfile.001.xml' />
```

使用される XML スキーマは、文書の内容として含まれる schemaLocation 属性によって決まります。その属性が指定されていない場合、妥当性検査は実行されません。

XML 列に以下の XDS が含まれているとすると、

```
<XDS FIL='xmlfile.002.xml' SCH='S1.SCHEMA_A' />
```

使用される XML スキーマは、文書の内容として含まれる schemaLocation 属性によって決まります。その属性が指定されていない場合、妥当性検査は実行されません。妥当性検査は **SCHEMALOCATION HINTS** を使用して実行されるため、SCH 属性は無視されることに注意してください。

### 使用上の注意

インポート操作を開始する前に、すべての表操作が完了し、すべてのロックがペンディング解除になっていることを確認してください。これは、WITH HOLD でオープンされた、すべてのカーソルをクローズした後で COMMIT または ROLLBACK を発行することによって行われます。

インポート・ユーティリティーは、SQL INSERT ステートメントを使用してターゲット表に行を追加します。このユーティリティーは、入力ファイル中の各行のデータにつき 1 つずつ INSERT ステートメントを発行します。INSERT ステートメントが失敗した場合、以下の 2 通りの結果のいずれかになります。

- 後続の INSERT ステートメントが成功すると予測される場合には、警告メッセージがメッセージ・ファイルに書き込まれ、処理が継続されます。
- 後続の INSERT ステートメントが失敗すると予測され、データベースが損傷する可能性がある場合には、エラー・メッセージがメッセージ・ファイルに書き込まれ、処理が停止されます。

このユーティリティーは、**REPLACE** または **REPLACE\_CREATE** 操作時に以前の行が削除された後、自動 COMMIT を実行します。したがって、表オブジェクトが切り捨てられた後、システムに障害が起こったり、アプリケーションがデータベース・マネージャーに割り込んだりすると、元のデータがすべて失われてしまいます。これらのオプションを使用する前に、元のデータがもはや必要ないことを確認してください。

**CREATE**、**REPLACE**、または **REPLACE\_CREATE** 操作時にログが満杯になると、このユーティリティーは挿入されたレコードに対して自動 COMMIT を実行します。自動 COMMIT の後に、システムに障害が起こるか、またはアプリケーションがデータベース・マネージャーに割り込むと、部分的にデータの挿入された表はデータベース



内に残ります。 **REPLACE** または **REPLACE\_CREATE** オプションを使用してインポート操作全体をやり直すか、または正常にインポートされる行の数に設定した **RESTARTCOUNT** パラメーターを指定して **INSERT** を使用してください。

**IMPORT** コマンドからの更新は、常に **IMPORT** タスクの終わりにコミットされません。**IMPORT** コマンドの実行時に自動コミットを実行して、ロック・リストのサイズとアクティブ・ログ・スペースを減らすこともできます。**IMPORT** 処理中にアクティブ・ログがいっぱいになると、**IMPORT** コマンドはロールバックします。

- デフォルトでは、**INSERT** または **INSERT\_UPDATE** オプションについては自動コミットは実行されません。しかし、**COMMITCOUNT** パラメーターがゼロでない場合は実行されます。
- 以下のいずれかの条件が真であると、オフライン・インポートでは自動の **COMMIT** は実行されません。
  - ターゲットは表ではなくビューである。
  - コンパウンド挿入を使用している。
  - バッファ挿入を使用している。
- デフォルトでは、オンライン・インポートは自動コミットを実行して、アクティブ・ログ・スペースとロック・リストを両方とも解放します。自動コミットが実行されないのは、ゼロの **COMMITCOUNT** 値を指定した場合のみです。

インポート・ユーティリティーが **COMMIT** を実行するたびに、2つのメッセージがメッセージ・ファイルに書き込まれます。一方は、コミットされるレコードの数を示し、もう一方は、**COMMIT** の成功後に書き込まれます。障害の後にインポート操作を再開するときには、スキップするレコードの数 (最後の正常なコミットから判別される) を指定してください。

インポート・ユーティリティーでは、多少の非互換性問題がある入力データは受け入れられます (例えば、文字データは埋め込みまたは切り捨てを用いてインポートできます。数値データは異なる数値データ・タイプを用いてインポートできます)。しかし、大きな非互換性問題のあるデータは受け入れられません。

オブジェクト表に何らかの従属 (それ自体への従属は除く) がある場合は、そのオブジェクト表を **REPLACE** または **REPLACE\_CREATE** することはできません。また、オブジェクト・ビューの基本表に何らかの従属 (それ自体への従属を含む) がある場合は、そのオブジェクト・ビューを **REPLACE** または **REPLACE\_CREATE** することはできません。そのような表またはビューを置換するには、以下のとおりに行ってください。

1. その表が親となっているすべての外部キーをドロップします。
2. インポート・ユーティリティーを実行します。
3. 表を変更して、外部キーを再作成します。

外部キーの再作成中にエラーが発生する場合、参照整合性を保守するためにデータを変更してください。

参照制約および外部キー定義は、**PC/IXF** ファイルから表を再作成する場合は保存されません。(主キー定義は、データが前に **SELECT \*** を使ってエクスポートされた場合、保存されます。)

リモート・データベースへのインポートでは、サーバーに、入力データ・ファイルのコピー、出力メッセージ・ファイル、およびデータベースのサイズ拡大を見込んだ十分なディスク・スペースが必要とされます。

インポート操作がリモート・データベースに対して実行され、出力メッセージ・ファイルが非常に長くなった (60 KB を超過) 場合、クライアントのユーザーに戻されるメッセージ・ファイルで、インポート操作中にメッセージが欠落する可能性があります。メッセージ情報の最初の 30 KB と最後の 30 KB は、常に保持されます。

PC/IXF ファイルのリモート・データベースへのインポートは、PC/IXF ファイルがディスクにあるときよりも、ハード・ディスクにあるときの方がより速く行うことができます。

データベース表または階層が存在していないと、ASC または DEL ファイル形式のデータをインポートできません。ただし、表が存在していない場合でも、**IMPORT CREATE** または **IMPORT REPLACE\_CREATE** は、PC/IXF ファイルからデータをインポートする際に表を作成します。型付き表の場合、**IMPORT CREATE** はタイプ階層と表階層も作成することができます。

PC/IXF インポートは、データベース間でデータ (階層データも含む) を移動する場合に使用します。行区切り文字を含む文字データが区切り文字付き ASCII (DEL) ファイルにエクスポートされ、テキスト転送プログラムによって処理される場合、行区切り文字を含むフィールドは長さが変わることがあります。ソースとターゲットのデータベースが両方とも同じクライアントからアクセス可能である場合、ファイルのコピーというステップは必要ありません。

ASC および DEL ファイルのデータは、インポートを実行するクライアント・アプリケーションのコード・ページであると仮定されます。異なるコード・ページのデータをインポートする場合は、異なるコード・ページを使用することのできる PC/IXF ファイルをお勧めします。PC/IXF ファイルとインポート・ユーティリティーが同じコード・ページである場合は、通常 of アプリケーションの場合のように処理が行われます。それぞれのコード・ページが異なり、**FORCEIN** オプションが指定されている場合、インポート・ユーティリティーは、PC/IXF ファイルのデータのコード・ページと、インポートを実行中のアプリケーションのコード・ページが同じであると見なします。この処理は、それら 2 つのコード・ページ用の変換テーブルが存在する場合であっても行われます。それぞれのコード・ページが異なり、**FORCEIN** オプションが指定されておらず、変換テーブルが存在する場合、PC/IXF ファイルのすべてのデータは、そのファイルのコード・ページからアプリケーションのコード・ページに変換されます。それぞれのコード・ページが異なり、**FORCEIN** オプションが指定されておらず、変換テーブルが存在しない場合、インポート操作は失敗します。これが該当するのは、AIX オペレーティング・システムの DB2 クライアント上の PC/IXF ファイルの場合だけです。

8 KB ページ上の表オブジェクトの量が 1012 列の制限に近い場合、PC/IXF データ・ファイルをインポートすると、SQL ステートメントの最大サイズを超過するため、DB2 はエラーを戻します。この状態が発生する可能性があるのは、列が CHAR、VARCHAR、または CLOB タイプの場合だけです。DEL または ASC ファイルのインポートには、この制限事項は適用されません。PC/IXF ファイルを



使って新しい表を作成している場合、別の方法として、**db2look** を使って表を作成した DDL ステートメントをダンプしてから、そのステートメント CLP から発行する、という方法があります。

DB2 Connect を使用して、DB2 for OS/390、DB2 for VM and VSE、および DB2 for OS/400 などの DRDA サーバーにデータをインポートすることができます。サポートされているのは、PC/IXF インポート (**INSERT** オプション) だけです。

**RESTARTCOUNT** パラメーターもサポートされていますが、**COMMITCOUNT** パラメーターはサポートされていません。

型付き表で **CREATE** オプションを使用するときは、PC/IXF ファイルで定義されているすべての副表を作成してください。副表の定義は変更できません。型付き表で **CREATE** 以外のオプションを使用するときは、全探索順序リストによって全探索順序を指定できます。このため、全探索順序リストはエクスポート操作時に使用したものと一致する必要があります。PC/IXF ファイル形式の場合は、ターゲット副表の名前を指定して、ファイルに格納されている全探索順序を使用するだけです。

インポート・ユーティリティーは、以前 PC/IXF ファイルにエクスポートされた表をリカバリーする場合に使用できます。その表は、エクスポート時の状態に戻ります。

システム表、作成済み一時表、宣言済み一時表、またはサマリー表にデータをインポートすることはできません。

インポート・ユーティリティーを介してビューを作成することはできません。

それぞれのパーツが Windows システムから AIX システムにコピーされる、複数のパーツからなる PC/IXF ファイルのインポートがサポートされています。最初のファイルの名前だけ、**IMPORT** コマンドで指定する必要があります。例えば、**IMPORT FROM data.ixf OF IXF INSERT INTO TABLE1** のようにします。ファイル data.002 などが、data.ixf と同じディレクトリーで使用できます。

Windows オペレーティング・システムの場合は、以下のとおりです。

- 論理分割された PC/IXF ファイルのインポートはサポートされていません。
- 不正な形式の PC/IXF ファイルのインポートは、サポートされていません。

内部形式のセキュリティー・ラベルには、改行文字が含まれている可能性があります。DEL ファイル形式を使用してファイルをインポートする場合、それらの改行文字が間違っただけで区切りと解釈される可能性があります。この問題が発生する場合は、**IMPORT** コマンドで **delprioritychar** ファイル・タイプ修飾子を指定することによって、区切り文字に関して以前に使用されていた古いデフォルト優先順位を使用してください。

暗黙的な非表示列がデータベース表に含まれている場合には、インポート操作に非表示列のデータを含めるかどうかを指定する必要があります。

**IMPORT** ユーティリティーは、表内の列数とデータ・ファイル内のフィールド数の突き合わせを行いません。このユーティリティーは、データ・ファイル内に十分な量のデータがあるかどうかをチェックします。データ・ファイル内の行に十分なデータ列が含まれていない場合、その行は拒否されて警告メッセージが表示される

(表の対応するデータの無い列が NOT NULL として定義されている場合) か、警告メッセージは表示されずに正常に挿入されます (表の対応する列が NULL として定義されている場合)。一方、行に必要な数より多くの列が含まれている場合、必要な数の列が処理され、残りのデータ列は省略されます。警告メッセージは表示されません。

### フェデレーテッドに関する考慮事項

IMPORT コマンドで INSERT、UPDATE、または INSERT\_UPDATE コマンド・パラメータを使用するときには、関係するニックネームに対する CONTROL 特権があることを確認してください。インポート操作で使いたいニックネームが既に存在することを確認する必要があります。そのほかにも、IMPORT コマンド・パラメータのセクションに記載されているようないくつかの制約事項に注意する必要があります。

ODBC などの一部のデータ・ソースでは、ニックネームへのインポートはサポートされていません。

### インポート・ユーティリティー用のファイル・タイプ修飾子

表 13. インポート・ユーティリティーの有効なファイル・タイプ修飾子: すべてのファイル形式

修飾子	説明
compound= <i>x</i>	<p><i>x</i> は 1 から 100 の数字です。非アトミック・コンパウンド SQL を使用してデータを挿入します。毎回 <i>x</i> 個のステートメントが試行されます。</p> <p>この修飾子が指定され、トランザクション・ログに十分な大きさが無い場合、インポート操作は失敗します。トランザクション・ログは、COMMITCOUNT によって指定された行数か、または COMMITCOUNT が指定されていない場合はデータ・ファイルの行数を入れる十分な大きさが必要です。したがって、トランザクション・ログのオーバーフローを避けるために、COMMITCOUNT オプションを指定することをお勧めします。</p> <p>この修飾子は、INSERT_UPDATE モード、階層表、および修飾子 usedefaults、identitymissing、identityignore、generatedmissing、generatedignore とは互換性がありません。</p>
generatedignore	<p>この修飾子は、インポート・ユーティリティーに、すべての生成列のデータはデータ・ファイルに存在するが、それらを見捨てることを知らせます。この結果として、生成列のすべての値は、このユーティリティーによって生成されます。この修飾子は、generatedmissing 修飾子と共に使用することはできません。</p>
generatedmissing	<p>この修飾子が指定されている場合、ユーティリティーは、生成列のデータが入力データ・ファイルに入っていない (NULL も入っていない) ものと見なし、行ごとに値を生成します。この修飾子は、generatedignore 修飾子と共に使用することはできません。</p>
identityignore	<p>この修飾子は、インポート・ユーティリティーに、ID 列のデータはデータ・ファイルに存在するが、それらを見捨てることを知らせます。この結果として、すべて ID 値はこのユーティリティーによって生成されます。この動作は、GENERATED ALWAYS および GENERATED BY DEFAULT のどちらの ID 列の場合も同じです。つまり、GENERATED ALWAYS 列の場合には、リジェクトされる行はありません。この修飾子は、identitymissing 修飾子とともに使用することはできません。</p>

表 13. インポート・ユーティリティの有効なファイル・タイプ修飾子: すべてのファイル形式 (続き)

修飾子	説明
identitymissing	この修飾子を指定すると、ユーティリティは、ID 列のデータが入力データ・ファイルに入っていない (NULL も入っていない) ものと見なし、行ごとに値を生成します。この動作は、GENERATED ALWAYS および GENERATED BY DEFAULT のどちらの ID 列の場合も同じです。この修飾子は、identityignore 修飾子とともに使用することはできません。
implicitlyhiddeninclude	この修飾子が指定されている場合、ユーティリティは、入力データ・ファイルに暗黙非表示列へのデータが含まれると見なし、そのデータもインポートします。この修飾子は、implicitlyhiddenmissing 修飾子と共に使用することはできません。複数の修飾子が指定されている場合の優先順位については、注: セクションを参照してください。
implicitlyhiddenmissing	この修飾子を指定すると、ユーティリティは入力データ・ファイルに暗黙的な非表示列のデータが含まれないと見なし、ユーティリティがそれらの暗黙的な非表示列の値を生成します。この修飾子は、implicitlyhiddeninclude 修飾子と共に使用することはできません。複数の修飾子が指定されている場合の優先順位については、注: セクションを参照してください。
lobsinfile	<p><i>lob-path</i> には、LOB データの入ったファイルへのパスを指定します。</p> <p>各パスには、データ・ファイル内で LOB ロケーション指定子 (LLS) によって示される 1 つ以上の LOB の入った、少なくとも 1 つのファイルが組み込まれます。LLS は、LOB ファイル・パスに保管されるファイル内の LOB のロケーションのストリング表現です。LLS の形式は、<i>filename.ext.nnn.mmm/</i> です。<i>filename.ext</i> は LOB を収めたファイルの名前、<i>nnn</i> はファイル内の LOB のオフセット (バイト単位)、<i>mmm</i> は LOB の長さ (バイト単位) を表します。例えば、ストリング <i>db2exp.001.123.456/</i> がデータ・ファイルに保管される場合、LOB はファイル <i>db2exp.001</i> のオフセット 123 に位置し、456 バイト長です。</p> <p><b>LOBS FROM</b> 節は、「lobsinfile」修飾子が使用されているときの、LOB ファイルの場所を指定します。<b>LOBS FROM</b> 節によって、LOBSINFILE 動作が暗黙的にアクティブ化されます。<b>LOBS FROM</b> 節は、データのインポート中に、IMPORT ユーティリティに LOB ファイルを検索するためのパスのリストを送ります。</p> <p>NULL LOB を指定するには、サイズに -1 と入力します。サイズを 0 と指定すると、長さが 0 の LOB として扱われます。長さが -1 の NULL LOB の場合、オフセットとファイル名は無視されます。例えば、NULL LOB の LLS は、<i>db2exp.001.7.-1/</i> のようになります。</p>
no_type_id	1 つの副表にインポートする場合にのみ有効です。これを使う場合として典型的な例は、REGULAR 表からデータをエクスポートした後、この修飾子を使ってインポート操作を呼び出してそのデータを単一の副表に変換する場合です。
nodefaults	<p>ターゲット表の列に対応するソース列が明示的に指定されていない場合、その表列が NULL 不可能なら、デフォルト値はロードされません。このオプションを指定せず、あるターゲット表列のためのソース列が明示的に指定されていない場合、以下のいずれかになります。</p> <ul style="list-style-type: none"> <li>• 列にデフォルト値を指定できる場合、そのデフォルト値がロードされます。</li> <li>• 列が NULL 可能で、デフォルト値がその列に指定できない場合、NULL がロードされます。</li> <li>• 列が NULL 不可能で、デフォルト値がその列に指定できない場合、エラーが戻され、ユーティリティは処理を停止します。</li> </ul>
norowwarnings	リジェクトされた行についてのすべての警告を抑制します。

## IMPORT

表 13. インポート・ユーティリティの有効なファイル・タイプ修飾子: すべてのファイル形式 (続き)

修飾子	説明
<b>periodignore</b>	この修飾子は、インポート・ユーティリティに対して、期間列へのデータがデータ・ファイル内に存在しても無視するように通知します。この修飾子が指定された場合、すべての期間列の値はユーティリティによって生成されます。この修飾子は、 <b>periodmissing</b> 修飾子と共に使用することはできません。
<b>periodmissing</b>	この修飾子が指定されている場合、ユーティリティは、期間列のデータが入力データ・ファイルに入っていないものと見なします。この修飾子が指定された場合、すべての期間列の値はユーティリティによって生成されます。この修飾子は、 <b>periodignore</b> 修飾子と共に使用することはできません。
rowchangetimestampignore	この修飾子は、インポート・ユーティリティに、行変更タイム・スタンプの列のデータがデータ・ファイルに存在するが、それらを見捨てることを知らせます。この結果、すべての ROW CHANGE TIMESTAMP はユーティリティによって生成されます。この動作は、GENERATED ALWAYS 列でも GENERATED BY DEFAULT 列でも同じです。つまり、GENERATED ALWAYS 列の場合には、リジェクトされる行はありません。この修飾子は、rowchangetimestampmissing 修飾子とともに使用することはできません。
rowchangetimestampmissing	この修飾子を指定すると、ユーティリティは、行変更タイム・スタンプ列のデータが入力データ・ファイルに入っていない (NULL も入っていない) ものと見なし、行ごとに値を生成します。この動作は、GENERATED ALWAYS 列でも GENERATED BY DEFAULT 列でも同じです。この修飾子は、rowchangetimestampignore 修飾子と共に使用することはできません。
seclabelchar	<p>入力ソース・ファイル内のセキュリティ・ラベルが、デフォルトのエンコードされた数値形式ではなく、セキュリティ・ラベル値の文字列形式であることを示します。 <b>IMPORT</b> は、各セキュリティ・ラベルをロードする際に内部形式に変換します。文字列が適切な形式ではない場合、行はロードされず、警告 (SQLSTATE 01H53) が戻されます。文字列が表を保護するセキュリティ・ポリシーの一部である有効なセキュリティ・ラベルを表していない場合、行はロードされず、警告 (SQLSTATE 01H53、SQLCODE SQL3243W) が戻されます。</p> <p>seclabelname 修飾子が指定されている場合には、この修飾子は指定できません。指定すると、インポートは失敗して、エラー (SQLCODE SQL3525N) が戻されます。</p>
seclabelname	<p>入力ソース・ファイル内のセキュリティ・ラベルが、デフォルトのエンコードされた数値形式ではなく、名前によって指定されることを示します。 <b>IMPORT</b> はその名前を適切なセキュリティ・ラベル (存在する場合) に変換します。表を保護するセキュリティ・ポリシーで、指定された名前のセキュリティ・ラベルが存在しない場合、行はロードされず、警告 (SQLSTATE 01H53、SQLCODE SQL3244W) が戻されます。</p> <p>seclabelchar 修飾子が指定されている場合には、この修飾子は指定できません。指定すると、インポートは失敗して、エラー (SQLCODE SQL3525N) が戻されます。</p> <p><b>注:</b> ファイル・タイプが ASC の場合、セキュリティ・ラベルの名前の後にスペースがあれば、それも名前の一部として解釈されます。これを回避するには、striptblanks ファイル・タイプ修飾子を使用してスペースが除去されるようにします。</p>

表 13. インポート・ユーティリティの有効なファイル・タイプ修飾子: すべてのファイル形式 (続き)

修飾子	説明
<b>transactionidignore</b>	この修飾子は、インポート・ユーティリティに対して、TRANSACTION START ID 列へのデータがデータ・ファイル内に存在しても無視するように通知します。この修飾子が指定された場合、TRANSACTION START ID 列の値はユーティリティによって生成されます。この修飾子は、 <b>transactionidmissing</b> 修飾子と共に使用することはできません。
<b>transactionidmissing</b>	この修飾子が指定されている場合、ユーティリティは、TRANSACTION START ID 列のデータが入力データ・ファイルに入っていないものと見なします。この修飾子が指定された場合、TRANSACTION START ID 列の値はユーティリティによって生成されます。この修飾子は、 <b>transactionidignore</b> 修飾子と共に使用することはできません。
usedefaults	<p>ターゲット表の列のソース列が指定されているが、1 つ以上の行インスタンスのデータが入っていない場合は、デフォルト値がロードされます。欠落データの例を以下に示します。</p> <ul style="list-style-type: none"> <li>• DEL ファイルの場合: 列の値として、2 つの連続した列区切り (,) や、任意の数のスペースで分離した 2 つの連続する列区切り (, ) が指定されている。</li> <li>• DEL/ASC ファイルの場合: 列が不足している行、または元の指定には十分な長さでない行。</li> </ul> <p><b>注:</b> ASC ファイルの場合: NULL 列値は明示的に欠落しているとは見なされず、NULL 列値にはデフォルトが置換されません。数値、日付、時刻、タイム・スタンプの列では、全桁スペース文字で NULL 列値を表記します。また、どのタイプの列でも、NULL INDICATOR を使用すれば、その列が NULL であることを示せます。</p> <p>このオプションが指定されていない場合、行インスタンスのソース列にデータがないと、以下のいずれかの処理が行われます。</p> <ul style="list-style-type: none"> <li>• DEL/ASC ファイルの場合: 列が NULL 可能であれば、NULL がロードされます。列が NULL 可能でない場合、ユーティリティはその行をリジェクトします。</li> </ul>

表 14. インポート・ユーティリティで有効なファイル・タイプ修飾子: ASCII ファイル・フォーマット (ASC/DEL)

修飾子	説明
codepage=x	<p>x は、ASCII 文字ストリングです。この値は、入力データ・セット内のデータのコード・ページとして解釈されます。インポート操作中に、文字データをこのコード・ページからアプリケーション・コード・ページに変換します。</p> <p>以下の規則が適用されます。</p> <ul style="list-style-type: none"> <li>• 純 DBCS (GRAPHIC)、混合 DBCS、および EUC では、区切り文字は x00 から x3F の範囲に制限されます。</li> <li>• nullindchar では、標準の ASCII セットのコード・ポイント x20 から x7F の範囲に含まれているシンボルを指定する必要があります。これは、ASCII 記号およびコード・ポイントを示します。</li> </ul> <p><b>注:</b></p> <ol style="list-style-type: none"> <li>1. codepage 修飾子を lobsinfile 修飾子と一緒に使用することはできません。</li> <li>2. コード・ページがアプリケーションのコード・ページからデータベースのコード・ページに変換されているときにデータの拡張が発生する場合は、データは切り捨てられ、データの消失が発生する可能性があります。</li> </ol>



## IMPORT

表 14. インポート・ユーティリティーで有効なファイル・タイプ修飾子: ASCII ファイル・フォーマット (ASC/DEL) (続き)

修飾子	説明
dateformat="x"	<p>x はソース・ファイルの日付のフォーマットです。<sup>2</sup> 有効な日付要素は以下のとおりです。</p> <p><b>YYYY</b> 年 (0000 から 9999 の範囲の 4 桁の数)</p> <p><b>M</b> 月 (1 から 12 の範囲の 1 桁または 2 桁の数)</p> <p><b>MM</b> 月 (01 から 12 の 2 桁の数。M とは相互に排他的)</p> <p><b>D</b> 日 (1 から 31 の範囲の 1 桁または 2 桁の数)</p> <p><b>DD</b> 日 (01 から 31 の範囲の 2 桁の数。D とは相互に排他的)</p> <p><b>DDD</b> 元日から数えた日数 (001 から 366 の範囲の 3 桁の数。他の日または月の要素とは相互に排他的)</p> <p>デフォルト値の 1 が、指定されない各要素に割り当てられます。日付形式の例を以下に示します。</p> <p>"D-M-YYYY"  "MM.DD.YYYY"  "YYYYDDD"</p>
implieddecimal	<p>暗黙指定されている小数点の位置が列定義によって決定され、値の終わりにあるとは見なされなくなります。例えば、値 12345 は、12345.00 ではなく、123.45 として DECIMAL(8,2) 列にロードされます。</p>
timeformat="x"	<p>x はソース・ファイル内の時刻のフォーマットです。<sup>2</sup> 有効な時刻要素は以下のとおりです。</p> <p><b>H</b> 時 (12 時間制の場合は 0 から 12、24 時間制では 0 から 24 の範囲の 1 桁または 2 桁の数。)</p> <p><b>HH</b> 時 (12 時間制の場合は 00 から 12、24 時間制では 0 から 24 の範囲の 2 桁の数。H と相互に排他的)</p> <p><b>M</b> 分 (0 から 59 の範囲の 1 桁または 2 桁の数)</p> <p><b>MM</b> 分 (00 から 59 の範囲の 2 桁の数。M とは相互に排他的)</p> <p><b>S</b> 秒 (0 から 59 の範囲の 1 桁または 2 桁の数)</p> <p><b>SS</b> 秒 (00 から 59 の範囲の 2 桁の数。S とは相互に排他的)</p> <p><b>SSSSS</b> 夜中の 12 時から数えた秒数 (00000 から 86400 の範囲の 5 桁の数。他の時刻要素とは相互に排他的)</p> <p><b>TT</b> 午前/午後の指定子 (AM または PM)</p> <p>指定されない各要素には、デフォルト値の 0 が割り当てられます。時刻フォーマットの例を以下に示します。</p> <p>"HH:MM:SS"  "HH.MM TT"  "SSSSS"</p>

表 14. インポート・ユーティリティーで有効なファイル・タイプ修飾子: ASCII ファイル・フォーマット (ASC/DEL) (続き)

修飾子	説明
timestampformat="x"	<p>x はソース・ファイル内のタイム・スタンプのフォーマットです。<sup>2</sup> 有効なタイム・スタンプ・エレメントは以下のとおりです。</p> <p><b>YYYY</b> 年 (0000 から 9999 の範囲の 4 桁の数)</p> <p><b>M</b> 月 (1 から 12 の範囲の 1 桁または 2 桁の数)</p> <p><b>MM</b> 月 (01 から 12 の 2 桁の数。M および MMM とは相互に排他的)</p> <p><b>MMM</b> 月 (大文字小文字を区別しない月名の 3 文字の省略形。M と MM とは相互に排他的)</p> <p><b>D</b> 日 (1 から 31 の範囲の 1 桁または 2 桁の数)</p> <p><b>DD</b> 日 (01 から 31 の範囲の 2 桁の数。D とは相互に排他的)</p> <p><b>DDD</b> 元日から数えた日数 (001 から 366 の範囲の 3 桁の数。他の日または月のエレメントとは相互に排他的)</p> <p><b>H</b> 時 (12 時間制の場合は 0 から 12、24 時間制では 0 から 24 の範囲の 1 桁または 2 桁の数。)</p> <p><b>HH</b> 時 (12 時間制の場合は 00 から 12、24 時間制では 0 から 24 の範囲の 2 桁の数。H と相互に排他的)</p> <p><b>M</b> 分 (0 から 59 の範囲の 1 桁または 2 桁の数)</p> <p><b>MM</b> 分 (00 から 59 の範囲の 2 桁の数。M (分) とは相互に排他的)</p> <p><b>S</b> 秒 (0 から 59 の範囲の 1 桁または 2 桁の数)</p> <p><b>SS</b> 秒 (00 から 59 の範囲の 2 桁の数。S とは相互に排他的)</p> <p><b>SSSSS</b> 夜中の 12 時から数えた秒数 (00000 から 86400 の範囲の 5 桁の数。他の時刻エレメントとは相互に排他的)</p> <p><b>U (1 から 12 時)</b>  <small>小数秒 (U のオカレンス数は、各桁を 0 から 9 の範囲として、桁数を表します)</small></p> <p><b>TT</b> 午前/午後の指定子 (AM または PM)</p> <p>YYYY、M、MM、D、DD、または DDD エレメントが指定されていない場合、デフォルト値の 1 が割り当てられます。MMM エレメントが指定されていない場合、デフォルト値の「Jan」が割り当てられます。他のエレメントが指定されていない場合には、デフォルト値の 0 が割り当てられます。タイム・スタンプ・フォーマットの例を以下に示します。</p> <p style="text-align: center;">"YYYY/MM/DD HH:MM:SS.UUUUUU"</p> <p>MMM エレメントの有効な値は、「jan」、「feb」、「mar」、「apr」、「may」、「jun」、「jul」、「aug」、「sep」、「oct」、「nov」、および「dec」です。これらの値では、大/小文字は区別されません。</p> <p>次の例では、ユーザ一定義の日時形式を指示するデータを、schedule という表にインポートする方法を示します。</p> <pre>db2 import from delfile2 of del modified by timestampformat="yyyy.mm.dd hh:mm tt" insert into schedule</pre>

## IMPORT

表 14. インポート・ユーティリティーで有効なファイル・タイプ修飾子: ASCII ファイル・フォーマット (ASC/DEL) (続き)

修飾子	説明
usegraphiccodepage	<p>usegraphiccodepage が指定された場合、GRAPHIC または 2 バイト文字ラージ・オブジェクト (DBCLOB) データ・フィールドにインポートされるデータは、GRAPHIC コード・ページであると見なされます。データの残りは、文字コード・ページであると見なされます。GRAPHIC コード・ページは、文字コード・ページと関連付けられます。IMPORT は、codepage 修飾子 (指定されている場合)、または codepage 修飾子が指定されていない場合はアプリケーションのコード・ページを介して、文字コード・ページを決定します。</p> <p>この修飾子は、リカバリーされている表に GRAPHIC データがある場合のみ、ドロップ済み表のリカバリーによって生成された区切りデータ・ファイルとともに使用される必要があります。</p> <p><b>制約事項</b></p> <p><b>EXPORT</b> ユーティリティーで作成された DEL ファイルは、1 つのコード・ページのみでエンコードされたデータを含んでいるため、これらのファイルで usegraphiccodepage 修飾子を指定することはできません。usegraphiccodepage 修飾子はまた、ファイル内の 2 バイト文字ラージ・オブジェクト (DBCLOB) には無視されます。</p>
xmlchar	<p>XML 文書が文字コード・ページでエンコードされていることを示します。</p> <p>このオプションは、指定の文字コード・ページでエンコードされていてもエンコード宣言を含まない XML 文書进行处理するために役立ちます。</p> <p>各文書で、宣言タグが存在してエンコード属性が含まれる場合、そのエンコード方式は文字コード・ページと一致する必要があります。一致しない場合、その文書を含む行はリジェクトされます。文字コード・ページは codepage ファイル・タイプ修飾子で指定されている値であるか、または指定がない場合はアプリケーションのコード・ページであることに注意してください。デフォルトでは、文書は Unicode でエンコードされているか、またはエンコード属性のある宣言タグを含んでいます。</p>
xmlgraphic	<p>XML 文書が指定された GRAPHIC コード・ページでエンコードされていることを示します。</p> <p>このオプションは、特定の GRAPHIC コード・ページでエンコードされていてもエンコード宣言を含まない XML 文書进行处理するために役立ちます。</p> <p>各文書で、宣言タグが存在してエンコード属性が含まれる場合、そのエンコード方式は GRAPHIC コード・ページと一致する必要があります。一致しない場合、その文書を含む行はリジェクトされます。GRAPHIC コード・ページは codepage ファイル・タイプ修飾子で指定されている値のグラフィック・コンポーネントであるか、または指定がない場合はアプリケーションのコード・ページのグラフィック・コンポーネントであることに注意してください。デフォルトでは、文書は Unicode でエンコードされているか、またはエンコード属性のある宣言タグを含んでいます。</p> <p><b>注:</b> xmlgraphic 修飾子が <b>IMPORT</b> コマンドで指定されている場合、インポートされる XML 文書は、UTF-16 コード・ページでエンコードされていなければなりません。そうでない場合、XML 文書は構文解析エラーで拒否されるか、またはデータ破損のある状態で表にインポートされる場合があります。</p>



表 15. インポート・ユーティリティーで有効なファイル・タイプ修飾子: ASC (区切り文字で区切られていない ASCII) ファイル・フォーマット

修飾子	説明
nochecklengths	nochecklengths が指定されていると、ソース・データの列定義がターゲット表の列のサイズを超えるものであっても、各行のインポートが試行されます。このような行が正常にインポートされるのは、コード・ページ変換でソース・データが縮小する場合です。例えば、ソースにある 4 バイトの EUC データがターゲットで 2 バイトの DBCS データに縮小すれば、必要スペースは半分になります。このオプションが特に役立つのは、列の定義は不一致であるがソース・データが常に適合することが分かっている場合です。
nullindchar=x	$x$ は、単一文字です。NULL 値を示す文字を $x$ に変更します。 $x$ のデフォルト値は $\backslash$ です。 <sup>3</sup>  文字が 1 つの英字である場合を除いて、この修飾子は EBCDIC データ・ファイルで大文字小文字を区別します。例えば、NULL 標識文字を文字 $N$ に指定した場合、 $n$ も NULL 標識と認識されます。
reclen=x	$x$ は、32 767 以下の整数です。各行ごとに $x$ 個の文字が読み取られ、行の終わりを示すのに改行文字は使用されません。
striptblanks	データを可変長フィールドにロードする際に、後書きブランク・スペースを切り捨てます。このオプションを指定しない場合、ブランク・スペースはそのまま保持されます。  次の例の場合、インポート・ユーティリティーは、striptblanks によって後書きブランク・スペースを切り捨てます。  <pre>db2 import from myfile.asc of asc   modified by striptblanks   method 1 (1 10, 12 15) messages msgs.txt   insert into staff</pre> このオプションは、striptnulls と一緒に指定することはできません。これらは、相互に排他的なオプションです。このオプションは、廃止された t オプション (下位互換性のためだけにサポートされる) に代わるものです。
striptnulls	データを可変長フィールドにロードする際に、後書き NULL (0x00 文字) を切り捨てます。このオプションを指定しない場合、NULL はそのまま保持されます。  このオプションは、striptblanks と一緒に指定することはできません。これらは、相互に排他的なオプションです。このオプションは、廃止された padwithzero オプション (下位互換性のためだけにサポートされる) に代わるものです。

## IMPORT

表 16. インポート・ユーティリティーで有効なファイル・タイプ修飾子: DEL (区切り文字で区切られている ASCII ファイル・フォーマット)

修飾子	説明
chardelx	<p>x は単一文字のストリング区切り文字です。デフォルト値は二重引用符 (") です。指定した文字は、文字ストリングを囲むために、二重引用符の代わりに使用されます。<sup>34</sup> 文字ストリング区切り文字として明示的に二重引用符を指定する場合、次のように指定します。</p> <pre>modified by chardel"</pre> <p>単一引用符 (') も、文字ストリングの区切り文字として指定できます。以下の例では、chardel' が指定されており、インポート・ユーティリティーは検出するすべての単一引用符 (') を文字ストリングの区切り文字として解釈します。</p> <pre>db2 "import from myfile.del of del modified by chardel'" method p (1, 4) insert into staff (id, years)"</pre>
coldelx	<p>x は単一文字の列区切り文字です。デフォルト値はコンマ (,) です。指定した文字は、列の終わりを表すために、コンマの代わりに使用されます。<sup>34</sup></p> <p>以下の例では、coldel; が指定されており、インポート・ユーティリティーは検出するすべてのセミコロン (;) を列の区切り文字として解釈します。</p> <pre>db2 import from myfile.del of del modified by coldel; messages msgs.txt insert into staff</pre>
decplusblank	<p>正符号文字。正の 10 進値の接頭部として、正符号 (+) ではなくブランク・スペースを使用します。デフォルトのアクションでは、正の 10 進数の前に正符号 (+) が付けられます。</p>
decptx	<p>x は、小数点文字としてピリオドの代わりに使用される単一の置換文字です。デフォルト値はピリオド (.) です。指定した文字は、小数点文字としてピリオドの代わりに使用されます。<sup>34</sup></p> <p>以下の例では、decpt; が指定されており、インポート・ユーティリティーは検出するすべてのセミコロン (;) を小数点として解釈します。</p> <pre>db2 "import from myfile.del of del modified by chardel'" decpt; messages msgs.txt insert into staff"</pre>
delprioritychar	<p>区切り文字の現在のデフォルト優先順位は、(1) レコード区切り文字、(2) 区切り文字、(3) 列区切り文字です。この修飾子を使用すると、区切り文字の優先順位が (1) 区切り文字、(2) レコード区切り文字、(3) 列区切り文字に戻り、以前の優先順位に依存している既存のアプリケーションが保護されます。構文は以下のとおりです。</p> <pre>db2 import ... modified by delprioritychar ...</pre> <p>例えば、以下のような DEL データ・ファイルがあるとします。</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 修飾子を指定しているため、このデータ・ファイルは、2 行だけになります。2 番目の &lt;row delimiter&gt; は 2 番目の行の最初のデータ列の一部と解釈されますが、1 番目と 3 番目の &lt;row delimiter&gt; は実レコードの区切り文字と解釈されます。この修飾子が指定されていない場合、このデータ・ファイルでは 3 行になり、各行は &lt;row delimiter&gt; によって区切られます。</p>

表 16. インポート・ユーティリティで有効なファイル・タイプ修飾子: DEL (区切り文字で区切られている ASCII) ファイル・フォーマット (続き)

修飾子	説明
keepblanks	タイプが CHAR、VARCHAR、LONG VARCHAR、または CLOB の各フィールドの前後の空白を保持します。このオプションを指定しないと、区切り文字で囲まれていないすべての前後の空白は除去され、表のすべての空白・フィールドに NULL が挿入されます。
nochardel	インポート・ユーティリティは、列区切り文字の間にあるすべてのバイトを列のデータの一部であると見なします。文字区切り文字は、列データの一部として構文解析されます。データが DB2 を使用してエクスポートされている場合は、このオプションを指定しないでください (エクスポート時に nochardel が指定されない限り)。これは、区切り文字を持たないベンダー・データ・ファイルをサポートするために用意されています。不適切に使用すると、データが損失または破壊される場合があります。  このオプションを chardelex、delprioritychar または nodoubledel と一緒に指定することはできません。これらは、相互に排他的なオプションです。
nodoubledel	二重文字区切りの認識を抑制します。

表 17. インポート・ユーティリティで有効なファイル・タイプ修飾子: IXF ファイル・フォーマット

修飾子	説明
forcein	コード・ページが不一致でもデータを受け入れ、コード・ページ間の変換を抑制するようにユーティリティに指示します。  固定長ターゲット・フィールドに、そのデータが入るだけの十分な大きさがあるかどうかチェックされます。nochecklengths が指定されていると、チェックは実行されず、各行のインポートが試行されます。
indexixf	既存の表に現在定義されている索引をすべてドロップし、PC/IXF ファイルの索引定義に基づいて新しい索引を作成するようにユーティリティに指示します。このオプションを使用できるのは、表の内容を置換する場合だけです。ビューでは使用できません。また、insert-column が指定されている場合にも使用できません。
indexschema=schema	指定した schema を、索引作成時の索引名として使用します。schema を指定しなかった場合 (しかしキーワード indexschema は指定した 場合) には、接続ユーザー ID が使用されます。このキーワードを指定しない場合、IXF ファイルのスキーマが使用されます。
nochecklengths	nochecklengths が指定されていると、ソース・データの列定義がターゲット表の列のサイズを超えるものであっても、各行のインポートが試行されます。このような行が正常にインポートされるのは、コード・ページ変換でソース・データが縮小する場合です。例えば、ソースにある 4 バイトの EUC データがターゲットで 2 バイトの DBCS データに縮小すれば、必要スペースは半分になります。このオプションが特に役立つのは、列の定義は不一致であるがソース・データが常に適合することが分かっている場合です。
forcecreate	インポート操作中に SQL3311N が戻された後、欠落している可能性のある、または限られた情報で表が作成されることを指定します。

## IMPORT

表 18. `codepage` および `usegraphiccodepage` 使用時の `IMPORT` 動作

<code>codepage=N</code>	<code>usegraphiccodepage</code>	<code>IMPORT</code> 動作
なし	なし	ファイル内のすべてのデータは、アプリケーション・コード・ページであると見なされます。
あり	なし	ファイル内のすべてのデータは、コード・ページ <code>N</code> であると見なされます。  <b>警告:</b> <code>N</code> が 1 バイト・コード・ページの場合、 <code>GRAPHIC</code> データをデータベースにインポートすると、壊れます。
なし	あり	ファイル内の文字データは、アプリケーション・コード・ページであると見なされます。 <code>GRAPHIC</code> データは、アプリケーション <code>GRAPHIC</code> データのコード・ページであると見なされます。  アプリケーション・コード・ページが 1 バイトの場合は、すべてのデータはアプリケーション・コード・ページであると見なされます。  <b>警告:</b> アプリケーション・コード・ページが 1 バイトの場合、 <code>GRAPHIC</code> データは、データベースにたとえ <code>GRAPHIC</code> 列が収められていても、データベースにインポートされると壊れます。
あり	あり	文字データは、コード・ページ <code>N</code> であると見なされま す。 <code>GRAPHIC</code> データは、 <code>N</code> の <code>GRAPHIC</code> コード・ペ ージであると見なされます。  <code>N</code> が 1 バイトまたは 2 バイト・コード・ページの場合 は、すべてのデータは、コード・ページ <code>N</code> であると見な されます。  <b>警告:</b> <code>N</code> が 1 バイト・コード・ページの場合、 <code>GRAPHIC</code> データをデータベースにインポートすると、壊れます。

### 注:

1. サポートされていないファイル・タイプを **MODIFIED BY** オプションで使用しようとしても、インポート・ユーティリティは警告を出しません。この場合、インポート操作が失敗し、エラー・コードが戻されます。
2. 日付形式ストリングは必ず二重引用符で囲まなければなりません。フィールド区切り文字には、`a` から `z`、`A` から `Z`、および `0` から `9` を使用することはできません。フィールド区切り文字として、`DEL` ファイル・フォーマットの文字区切りまたはフィールド区切りと同じ文字を使用することはできません。エレメントの開始および終了位置が明らかな場合、フィールド区切り文字は任意指定です。あいまいさが生じうるのは、項目の長さが一定でない `D`、`H`、`M`、または `S` などのエレメントが使用されている場合です (修飾の仕方によって異なります)。

タイム・スタンプ・フォーマットの場合、月の記述子と分の記述子のどちらも文字 `M` を使用するため、区別があいまいにならないように注意する必要があります。月のフィールドは、他の日付フィールドと隣接していなければなりま

せん。分フィールドは、他の時刻フィールドに隣接していなければなりません。あいまいなタイム・スタンプ形式の例を以下に示します。

```
"M" (could be a month, or a minute)
"M:M" (Which is which?)
"M:YYYY:M" (Both are interpreted as month.)
"S:M:YYYY" (adjacent to both a time value and a date value)
```

あいまいな場合、ユーティリティーはエラー・メッセージを報告し、操作は失敗します。

以下に、明確なタイム・スタンプ・フォーマットを示します。

```
"M:YYYY" (Month)
"S:M" (Minute)
"M:YYYY:S:M" (Month...Minute)
"M:H:YYYY:M:D" (Minute...Month)
```

二重引用符や円記号などの文字の前には、エスケープ文字 (例えば、¥) を付けるなければなりません。

3. `chardel`、`coldel`、または `decpt` ファイル・タイプ修飾子に提供される文字値は、ソース・データのコード・ページで指定する必要があります。

文字コード・ポイント (文字記号ではない) は、`xJJ` または `0xJJ` という構文で指定することができます (`JJ` はコード・ポイントの 16 進表記)。例えば、列区切りとして `#` 文字を指定するには、以下のステートメントのいずれかを使用します。

```
... modified by coldel# ...
... modified by coldel0x23 ...
... modified by coldelX23 ...
```

4. データ移動のための区切り文字の制約事項に、区切り文字の指定変更として使用できる文字に適用される制限のリストが示されています。
5. 以下のファイル・タイプ修飾子は、ニックネームにインポートするときには使用できません。
  - `indexixf`
  - `indexschema`
  - `dldel filetype`
  - `nodefaults`
  - `usedefaults`
  - `no_type_id filetype`
  - `generatedignore`
  - `generatedmissing`
  - `identityignore`
  - `identitymissing`
  - `lobsinfile`
6. **CREATE** モードは、XML 列ではサポートされません。
7. すべての XML データは、メイン・データ・ファイルとは別の XML ファイル内に存在する必要があります。XML Data Specifier (XDS) (または NULL 値) が、メイン・データ・ファイル内の XML 列ごとに存在する必要があります。

## IMPORT

8. XMLCHAR または XMLGRAPHIC ファイル・タイプ修飾子が指定されていない場合は、XML 文書は Unicode 形式であるか、またはエンコード属性のある宣言タグを含むと想定されます。
9. 整形式でない文書が含まれている行はリジェクトされます。
10. **XMLVALIDATE** オプションが指定されている場合、対応するスキーマに対して正常に妥当性検査された文書は、挿入される際にスキーマ情報がアノテーションとして付加されます。対応するスキーマに対する妥当性検査が失敗した文書を含む行は、リジェクトされます。妥当性検査を正常に行うためには、インポートを起動するユーザーの保持する特権に、次の 1 つ以上が含まれている必要があります。
  - DBADM 権限
  - 妥当性検査に使用する XML スキーマに対する USAGE 特権
11. **ignore**、**include**、**missing**、**override** が接尾部に付された複数の修飾子が指定されると、それらの修飾子はリストされた順に適用されます。以下のステートメントの場合、入力データに ID 列ではない暗黙的な非表示列のデータが含まれます。一方、暗黙的に非表示かどうかに関係なく、ID 列のデータは含まれません。

```
db2 import from delfile1 of del modified by
    implicitlyhiddeninclude identitymissing insert into table1
```

しかし、以下のステートメントにおいてファイル・タイプ修飾子の順序を変更すると、入力データにすべての暗黙的な非表示列のデータ (非表示の ID 列も含む) が含まれることとなります。一方、暗黙的に非表示でない ID 列のデータは含まれません。

```
db2 import from delfile1 of del modified by
    identitymissing implicitlyhiddeninclude insert into table1
```

DB2\_DMU\_DEFAULT レジストリー変数が **IMPLICITLYHIDDENINCLUDE** に設定されている場合:

```
db2set DB2_DMU_DEFAULT=IMPLICITLYHIDDENINCLUDE
db2 import from delfile1 of del modified by identitymissing insert into table1
```

これは、以下と同じ意味になります。

```
db2 import from delfile1 of del modified by
    implicitlyhiddeninclude identitymissing insert into table1
```

## INGEST

入力ファイルまたは入力パイプからデータを DB2 表に INGEST します。

### 許可

データベースへの接続に使用する許可 ID の特権には、以下が含まれていなければなりません。

- 以下の権限の少なくとも 1 つ
  - DATAACCESS 権限
  - ターゲット表に対する CONTROL 特権
  - **INGEST** コマンドに INSERT ステートメントを指定する場合 (MERGE ステートメントの一部として指定する場合も含む) は、ターゲット表に対する SELECT 特権および INSERT 特権
  - **INGEST** コマンドに UPDATE ステートメントを指定する場合 (MERGE ステートメントの一部として指定する場合も含む) は、ターゲット表に対する SELECT 特権および UPDATE 特権
  - **INGEST** コマンドに DELETE ステートメントを指定する場合 (MERGE ステートメントの一部として指定する場合も含む) は、ターゲット表に対する SELECT 特権および DELETE 特権
  - **INGEST** コマンドに REPLACE 節を指定する場合は、ターゲット表に対する INSERT 特権、SELECT 特権、および DELETE 特権
- 次のカタログ・ビューに対する SELECT 特権。
  - SYSCAT.COLUMNS
  - SYSCAT.DATATYPES
  - SYSCAT.INDEXES
  - SYSCAT.INDEXCOLUSE
  - SYSCAT.SECURITYPOLICIES (ターゲット表にセキュリティー・ラベル列がある場合)
  - SYSCAT.TABDEP
  - SYSCAT.TABLES
  - SYSCAT.VIEWS

**注:** RESTRICTIVE 節を指定してデータベースを作成したのでなければ、ユーザーはデフォルトでこれらの特権を持っています。

- 次のプロシージャに対する EXECUTE 特権。
  - SYSPROC.DB\_PARTITIONS (V9.7 以前) または SYSPROC.DB\_MEMBERS (V9.8 以降)
  - SYSPROC.MON\_GET\_CONNECTION (V9.7 以降) または SYSIBMADM.APPLICATIONS (V9.5)
- ターゲット表にトリガーがある場合は、そのトリガーに指定されている操作の実行に必要な特権を、許可 ID が持っていなければなりません。
- 保護列のある表を挿入または更新するには、許可 ID に、その表内のすべての保護列への書き込みアクセスを許可する LBAC 信用証明情報が必要です。そうでない場合、コマンドは失敗し、エラーが戻されます。

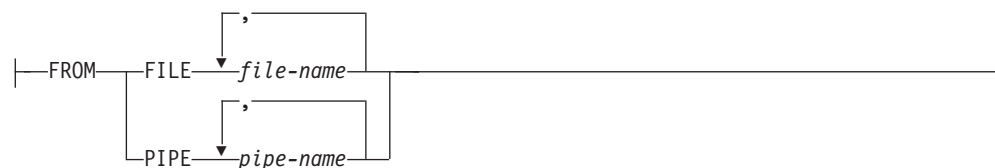






SQL-statement

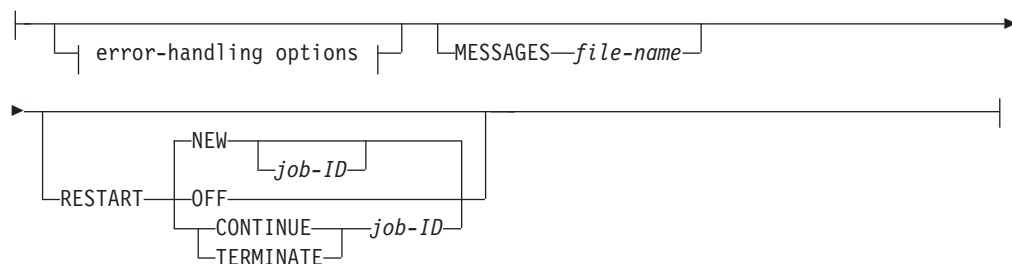
**from-clause:**



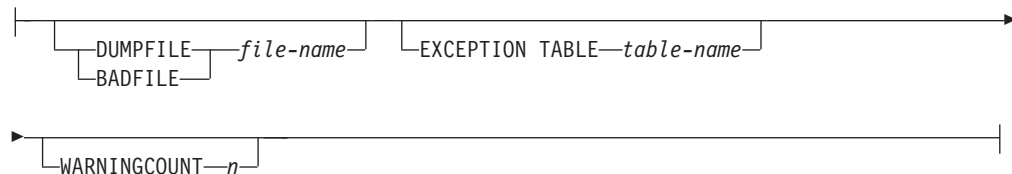
**format-clause:**

FORMAT format-definition

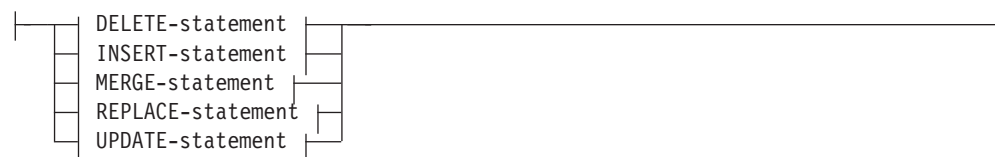
**options:**



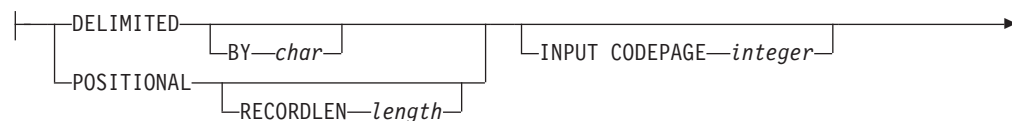
**error-handling options:**



**SQL-statement:**



**format-definition:**

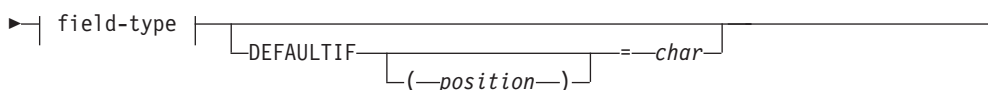
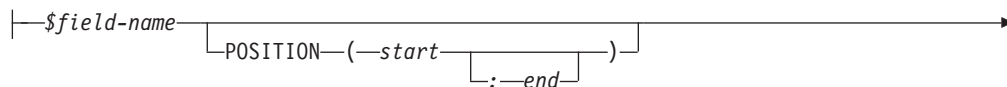




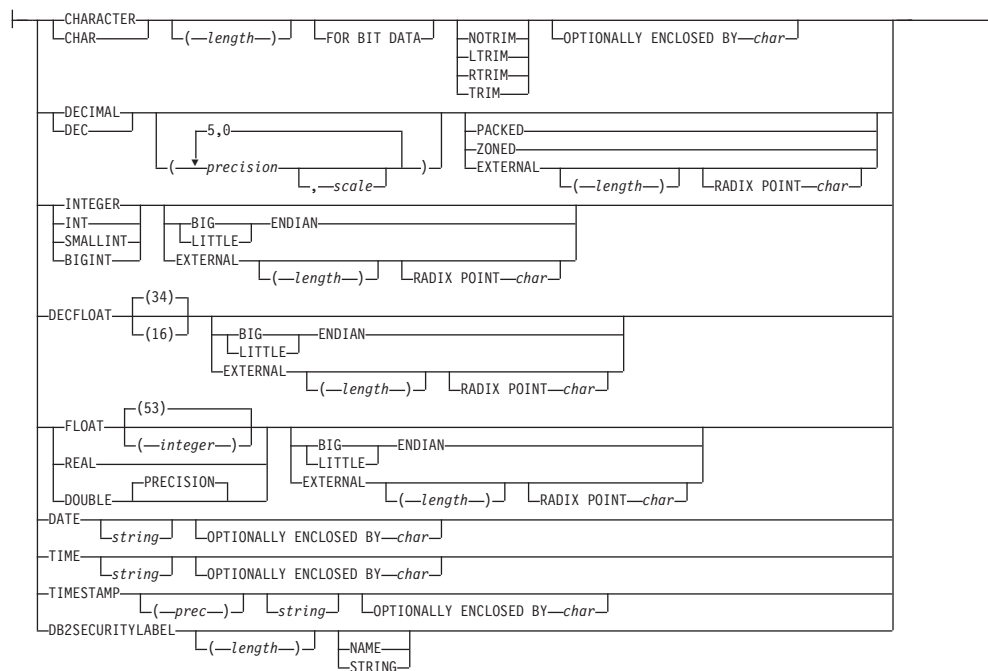
**field-list:**



**field-definition:**

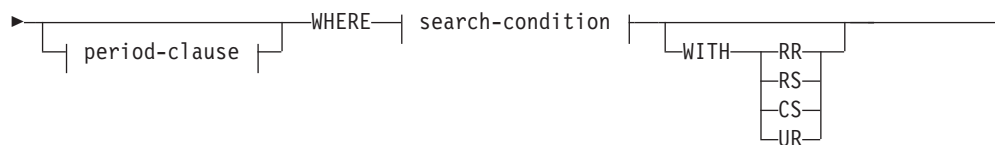


**field-type:**



**DELETE-statement:**

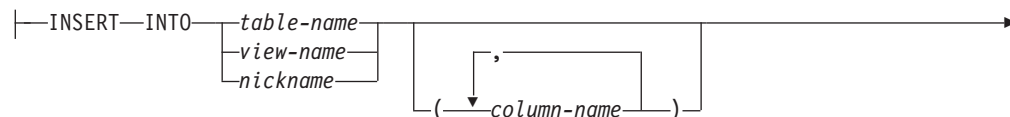




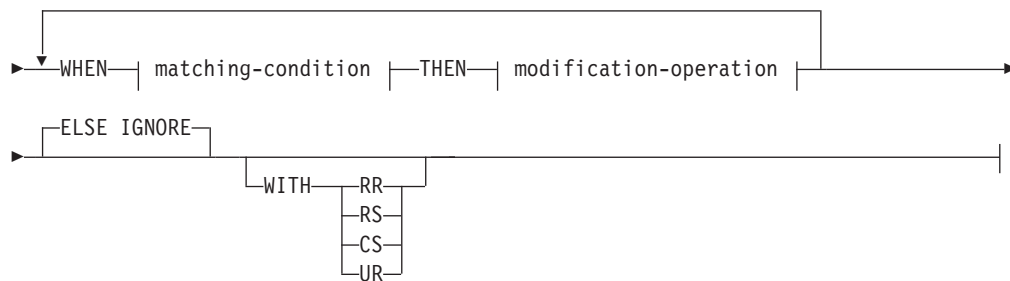
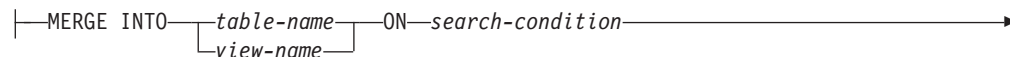
**period-clause:**



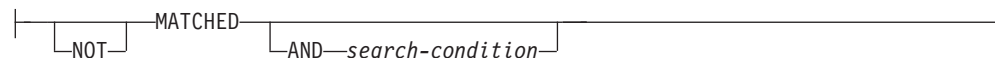
**INSERT-statement:**



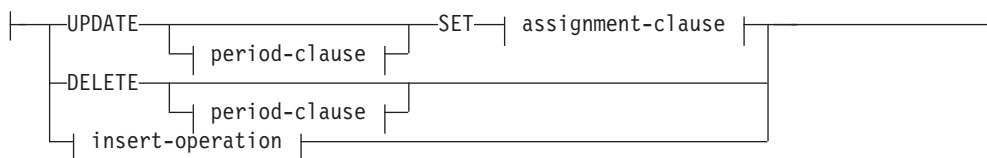
**MERGE-statement:**



**matching-condition:**



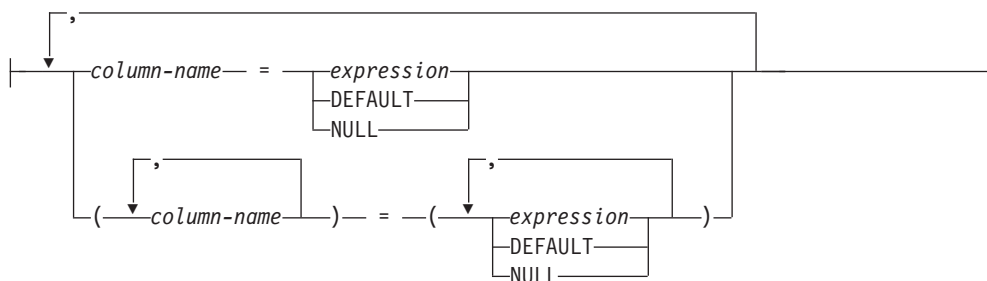
**modification-operation:**



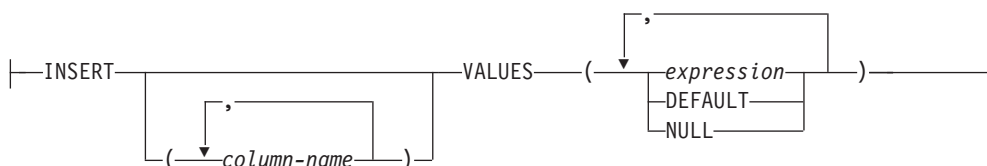
**period-clause:**



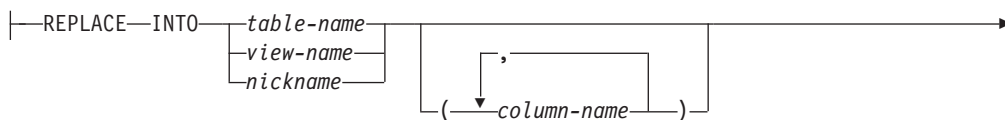
**assignment-clause:**



**insert-operation:**

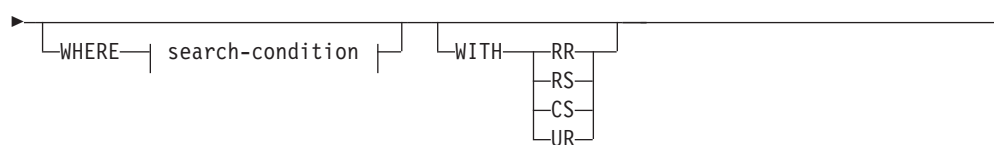


**REPLACE-statement:**

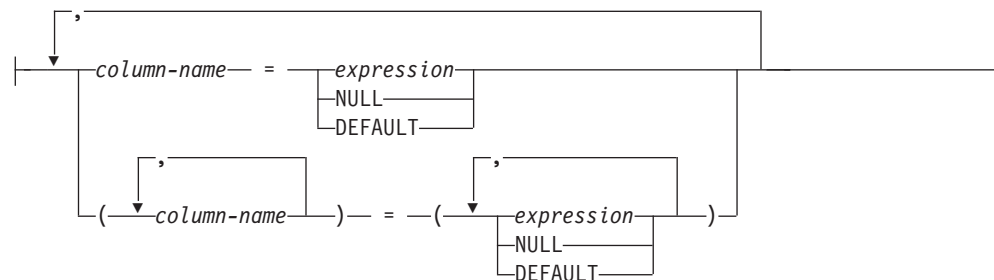


**UPDATE-statement:**

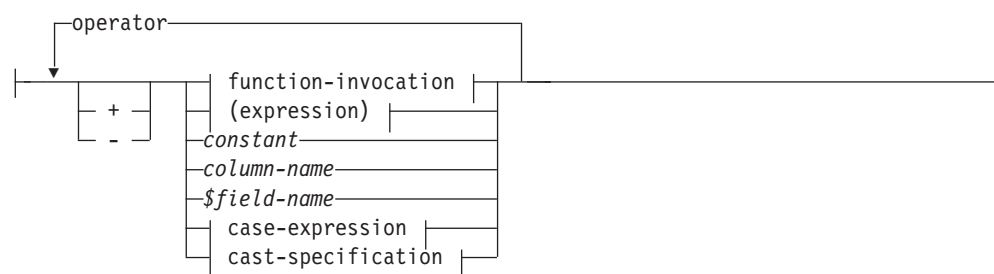




**assignment-clause:**



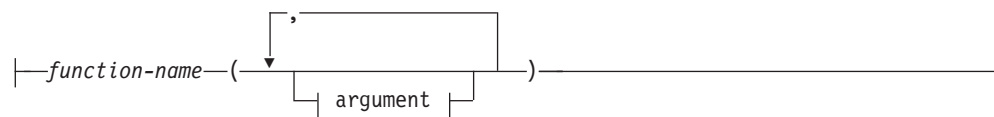
**expression:**



**operator:**



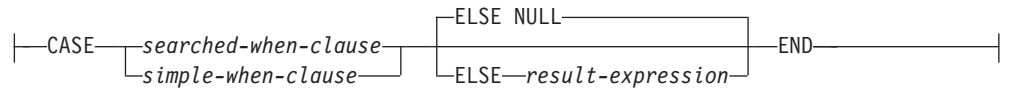
**function-invocation:**



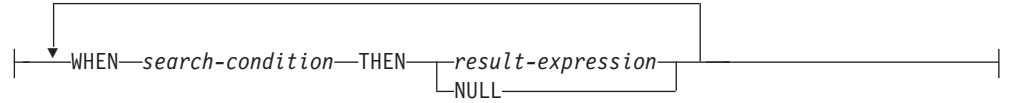
**argument:**



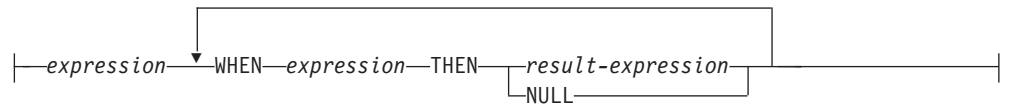
**case-expression:**



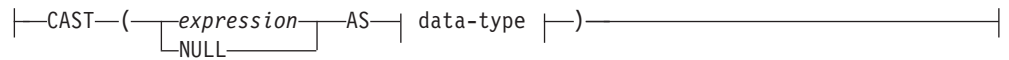
**searched-when-clause:**



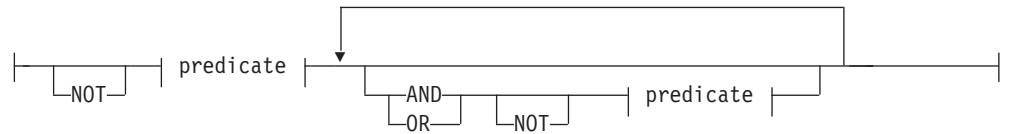
**simple-when-clause:**



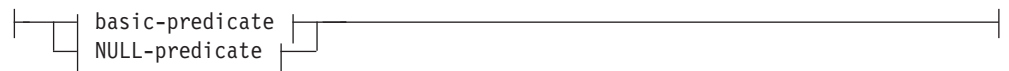
**cast-specification:**



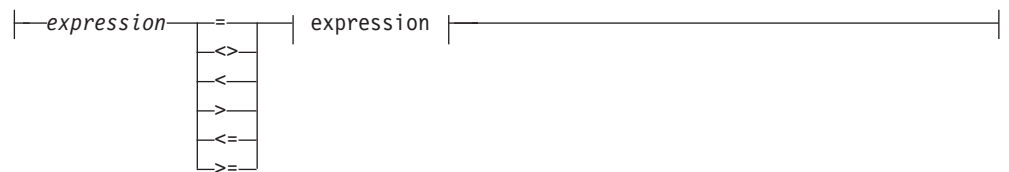
**search-condition:**



**predicate:**



**basic-predicate:**



**NULL-predicate:****コマンド・パラメーター****FROM**

入力データのソースを指定します。INGEST ユーティリティは、常に、入力データをクライアントから読み取ります。

**FILE *file-name***

指定されたファイルからデータを読み取るように指定します。ファイル名の構文については、357 ページの『ファイル名およびパイプ名』を参照してください。

**PIPE *pipe-name***

指定されたパイプからデータを読み取るように指定します。パイプ名の構文については、357 ページの『ファイル名およびパイプ名』を参照してください。

**FORMAT *format-definition***

データは、指定されたフォーマット定義で規定されたフォーマットに従います。フォーマット定義の構文は、前述の『format-definition』のセクションに示されています。

**DUMPFIL または BADFILE *file-name***

フォーマッターから拒否された行を、指定のファイルに書き込むように指定します。次のタイプのエラーの場合に、フォーマッターは行を拒否します。

- 無効な数値または範囲外の数値 (フィールド・タイプに基づく)
- 指定された書式と合わない日付、時刻、およびタイム・スタンプ
- フォーマッターによって検出されたその他のエラー

相対パスを指定する場合は、現行のディレクトリおよびデフォルト・ドライブが基準となります。Linux および UNIX プラットフォームでは、ファイルが存在しない場合、INGEST ユーティリティが、ユーザーの `umask` 設定に指定されている権限フラグを使用してファイルを作成します。

INGEST ユーティリティは、DB2 CLP バックエンド・プロセス (db2bp) の開始時 (通常は最初の接続時) に有効であった `umask` 設定を使用します。ファイルが既に存在する場合、INGEST ユーティリティは、そのファイルの最後にレコードを追加します。

INGEST ユーティリティは、入力データの表記と同じフォーマットを使用して、各レコードを書き込みます。つまり、エラーを修正した後に、ダンプ・ファイルを入力データとして使用して、INGEST コマンドを再実行することができます。ただし、INGEST ユーティリティは、複数の入力ファイルにおけるレコードの出現順序と同じ順序で、レコードを書き込むとは限りません。また、各入力ファイル内においても同様です。各入力ファイル内における出現順序と同じ順序で、レコードがダンプ・ファイルに入るにするには、`num_formatters` 構成パラメーターを 1 に設定します。このような場合には、次のような処理が行われます。

- 操作が INSERT または REPLACE である場合、さまざまな入力ファイルに由来するレコードは、やはり順序どおりには出現しない可能性があります。しかし、1 つの入力ファイルに由来するレコードは、すべてそのファイル内での順序と同じ順序で出現します。
- 操作が UPDATE、DELETE、または MERGE である場合、すべてのレコードは、すべての入力ファイルにおける順序と同じ順序となり、また、各入力ファイル内においても同じ順序となります。

このファイルのオープンまたは書き込みでエラーが発生した場合、INGEST ユーティリティーは、入力ソースおよび行番号を含むエラー・メッセージを出し、コマンドを終了させます。この場合、レコードはどこにも保存されません。(入力データがファイルからのものである場合、ユーザーはそのファイルから元の入力レコードを取得できます。)

ダンプ・ファイルが完結していることを保証できるのは、INGEST コマンドが 1 回の実行で正常に完了した場合のみです。INGEST コマンドが失敗し、再開させたコマンドが成功した場合、これら 2 回のコマンドで生成されたダンプ・ファイルの両方を合わせても、レコードが欠落していたり、重複レコードが含まれていたりする可能性があります。

このパラメーターを指定しない場合、INGEST ユーティリティーは、拒否された行ごとにエラー・メッセージを出しますが、その行をどこにも保存しません。

**注:** このパラメーターを指定しており、拒否された入力レコードに機密データが含まれている場合は、それらのレコードがダンプ・ファイル内に露出します。必要に応じて、そのファイルへの無許可アクセスを防止してください。

#### EXCEPTION TABLE *table-name*

INGEST ユーティリティーが挿入を試行したときに DB2 が特定の SQLSTATE で拒否した行を、指定の表に書き込むように指定します。DB2 は、以下のタイプのエラーで行を拒否する可能性があります。なお、これらのエラーの各々は、入力ファイル内の不良データを示しています。

- 文字データにおいて、右側が切り捨てられた。例えば、更新または挿入する値が、列に対して長すぎる文字列である、あるいは、日時値が小さすぎてホスト変数に割り当てられない、などが考えられます。
- 値が NULL であること、または標識パラメーターがないことが検出された。例えば、標識変数が指定されていないため、NULL 値をホスト変数に割り当てられない。
- 数値が範囲を超えている。
- 無効な日時フォーマットが検出された。つまり、無効な文字列表現または値が指定された。
- CAST 指定またはキャスト・スカラー関数の文字値が無効である。
- 文字がコード化文字セット内にはない。
- データ・パーティション・キーの値が無効である。
- 結果として得られた行が、行権限に適合しなかった。
- 挿入または更新の値は NULL であるが、列に NULL を含めることができない。



- FOREIGN KEY の挿入値または更新値が、親表の親キーのいずれの値とも異なる。
- ユニーク索引またはユニーク制約で定められている制約に対する違反が発生した。
- INSERT または UPDATE の結果の行が、チェック制約定義に合わない。
- その値は、表を保護しているセキュリティ・ポリシーの有効なセキュリティ・ラベルに変換できない。
- この許可 ID は、保護表に対して操作を実行することを許可されていない。
- セキュリティ・ラベル・コンポーネントにコンポーネント・エレメントが定義されていない。
- 指定されたセキュリティ・ラベル名が、指定されたセキュリティ・ポリシーで見つからない。
- ルーチンの引数のデータ・タイプ、長さ、または値が誤っている。

指定する表は、関連リンクのセクションのトピック『例外表』にある説明に従って定義する必要があります。そのセクションおよびメッセージ SQL3604N にリストされた制限も適用されますが、以下は例外です。

- ターゲット表に DB2SECURITYLABEL 列が含まれる場合、その DB2SECURITYLABEL 列に対応する例外表の列は、タイプが VARCHAR で、長さが 128 以上でなければなりません。(LONG VARCHAR タイプも許容されますが、推奨されていません。)

INGEST ユーティリティは、操作が INSERT または REPLACE である場合に限り、例外表を使用します。操作が UPDATE、MERGE、または DELETE であり、EXCEPTION TABLE パラメーターが指定されている場合、このユーティリティはエラーを発行します。

この表への挿入でエラーが発生した場合、INGEST ユーティリティは、入力ソースおよび行番号を含むエラー・メッセージを出し、処理を継続させます。この場合、レコードはどこにも保存されません。(入力データがファイルからのものである場合、ユーザーはそのファイルから元の入力レコードを取得できます。)

表名が完全修飾されていない場合、INGEST ユーティリティは、スキーマ名として CURRENT SCHEMA の値を使用します。EXCEPTION TABLE パラメーターを指定しない場合、INGEST ユーティリティは、拒否された行ごとにエラー・メッセージを出しますが、その行をどこにも保存しません。

#### WARNINGCOUNT *n*

*n* 個の警告およびエラー・メッセージが出たら、INGEST コマンドを停止するように指定します。このパラメーターの対象となるのは、特定のタイプのエラーのみです。詳しくは、エラーのカテゴリーを参照してください。その数はすべての入力ソースの合計です。トランザクションの途中でこの制限に達した場合、そのトランザクションはロールバックされます。ただし、制限に達する前にターゲット表に対してコミットされた行は、表に残されます。0 個の場合、INGEST コマンドは、警告およびエラーの数にかかわらず処理を継続します。範囲は、0 から 2,147,483,647 (32 ビット符号付き整数の最大値) までです。デフォルトは 0 です。

**MESSAGES** *file-name*

通知、警告、およびエラー・メッセージを受け取るファイルを指定します。

相対パスを指定する場合は、現行のディレクトリーおよびデフォルト・ドライブが基準となります。Linux および UNIX プラットフォームでは、ファイルが存在しない場合、INGEST ユーティリティーが、ユーザーの `umask` 設定に指定されている権限フラグを使用してファイルを作成します。

INGEST ユーティリティーは、DB2 CLP バックエンド・プロセス (db2bp) の開始時 (通常は最初の接続時) に有効であった `umask` 設定を使用します。ファイルが既に存在する場合、INGEST ユーティリティーは、そのファイルの最後に追加します。パラメーターが指定されていない場合、INGEST ユーティリティーは、メッセージを標準出力に書き込みます。

パラメーターが指定されていても、次の場合は、INGEST ユーティリティーはメッセージを標準出力に書き込みます。

- 構文エラー
- 入力ファイルが見つからない、または読み取り可能ではない
- ターゲット表または例外表が見つからない
- ダンプ・ファイルもメッセージ・ファイルも開けない
- 開始中に検出される他のエラー

また、最後に出力される要約メッセージ (読み取りや挿入などを行った行数と、警告およびエラーの総数を表示) は、常に標準出力に書き込まれます。

メッセージ・ファイルのオープン、またはそのファイルへの書き込みでエラーが発生した場合、INGEST ユーティリティーはエラー・メッセージを出してコマンドを終了します。

**INGEST** コマンドからのメッセージの例については、356 ページの『INGEST コマンドからのメッセージ』を参照してください。

**RESTART NEW** *job-ID*

**INGEST** コマンドが完了せずに失敗した場合に、後で **INGEST** コマンドに `RESTART CONTINUE` オプションを指定することによって、最後のコミット・ポイントから再開できるように指定します。*job-ID* は、最大 128 バイトのストリングであり、**INGEST** コマンドを一意的に識別します。この *job-ID* は、`RESTART` オプションが指定されていて、まだ完了していない、現行データベース内のすべての **INGEST** コマンドにおいて固有でなければなりません (このようなコマンドは、まだ実行中のコマンドであったり、完了せずに失敗したコマンドであったりする可能性があります)。**INGEST** コマンドが完了した場合、その *job-ID* は、今後の **INGEST** コマンドの `RESTART` パラメーターに再利用できます。*job-ID* を省略すると、INGEST ユーティリティーによって *job-ID* が生成されます。このオプションがデフォルトです。

このオプションを使用する前に、再開ログ表を作成しておく必要があります。詳しくは、下記の関連リンクのセクションにある『INGEST コマンドが失敗した場合の再開方法』を参照してください。

**RESTART OFF**

再開に関する情報を保存しないように指定します。**INGEST** コマンドが完了せずに失敗した場合、`RESTART CONTINUE` オプションを使用して再開することができません。このコマンドを再実行して完了させるには、ターゲット表

を **INGEST** コマンドの実行前の状態にリストアし、同じ入力データを使用して **INGEST** コマンドを再実行する必要があります。

#### **RESTART CONTINUE** *job-ID*

**RESTART NEW** オプションが指定されていて完了せずに失敗した、過去の **INGEST** コマンドを、**INGEST** ユーティリティーで再開するように指定します。このオプションに指定する *job-ID* は、その過去の **INGEST** コマンドに指定されていた *job-ID* と一致しなければなりません。この再開コマンドもまた再開可能です。再開コマンドが完了した場合、その *job-ID* は、今後の **INGEST** コマンドの **RESTART NEW** パラメーターに再利用できます。詳しくは、下記の関連リンクのセクションにある『**INGEST** コマンドが失敗した場合の再開方法』を参照してください。

#### **RESTART TERMINATE** *job-ID*

**RESTART NEW** オプションが指定されていて完了せずに失敗した、過去の **INGEST** コマンドの再開に関する情報を、**INGEST** ユーティリティーでクリーンアップするように指定します。このオプションに指定するストリングは、その過去の **INGEST** コマンドに指定されていたストリングと一致しなければなりません。

このオプションは、再開可能な **INGEST** コマンドが過去に失敗しているが、そのジョブを再開する予定がない場合に使用します。このオプションを指定すると、前に失敗した **INGEST** コマンドを、再開できなくなります。しかし、そのストリングは、今後の **INGEST** コマンドの **RESTART NEW** パラメーターに再利用できるようになります。

**注:** **LOAD** コマンドの **TERMINATE** パラメーターと異なり、元の **INGEST** コマンドが失敗する前にコミットしたデータは、ターゲット表内に残されます。詳しくは、下記の関連リンクのセクションにある『失敗した **INGEST** コマンドの終了』を参照してください。

#### **DELIMITED BY** *char*

各入力レコードが、指定の文字で区切られたテキスト・フィールドのシーケンスであることを指定します。区切り文字には、次の値を除く任意の 1 バイト文字を使用できます。

- ヌル文字 (X'00')
- ブランク (X'20')
- 復帰または改行 (X'0D' または X'0A')
- ドット (X'2E')、**RADIX POINT** 節の指定とは無関係
- 小数点 (**RADIX POINT** 節で指定されるもので、デフォルトはドット .)
- ストリング区切り文字 (**OPTIONALLY ENCLOSED BY** 節で指定されるもので、デフォルトは二重引用符 ")

入力データのコード・ページが **DBCS** または **EUC** コード・ページである場合、この文字は X'01' から X'3F' までの範囲内 (境界値を含む) になければなりません。垂直バー (I または X'7C') は含まれないことに注意してください。入力データのコード・ページが 1 バイト・コード・ページではない場合、この文字は X'01' から X'7F' までの範囲内 (境界値を含む) になければ

ばなりません。入力データのコード・ページが 1 バイト・コード・ページである場合、この文字は X'01' から X'FF' までの範囲内にすることができます。

デフォルトはコンマ , です。

### string

メタ変数 `string` には、次のような文字ストリング定数を指定します。

- ストリング区切り文字で始まりストリング区切り文字で終わる文字のシーケンス。この場合のストリング区切り文字はアポストロフィ (') です。
- 16 進定数: アポストロフィで囲んだ 16 進文字シーケンスの前に、X を付加したもの。
- Unicode ストリング定数: アポストロフィで囲んだ 16 進文字シーケンスの前に、U& を付加したもの (オプションで後に UESCAPE 節を付加することもできます)。
- グラフィック・ストリング定数: アポストロフィで囲んだ 2 バイト文字シーケンスの前に、G または N を付加したもの。
- 16 進グラフィック・ストリング定数: アポストロフィで囲んだ 16 進文字シーケンスの前に、GX を付加したもの。
- UCS-2 グラフィック・ストリング定数: アポストロフィで囲んだ 16 進文字シーケンスの前に、UX を付加したもの。

ストリング内に囲み文字がある場合は、そのストリング内で 2 回重ねて入力する必要があります。例えば、ストリングが "don't know" であり、囲み文字が単一引用符である場合は、'don''t know' と指定します。この例として、'This is a string'、X'303132'、U&'Unicode chars' を挙げます。

### char

メタ変数 `char` には、長さが 1 バイトの文字ストリング定数を指定します。例えば、'A' や X'30' とします。

## POSITIONAL

各入力レコードが、入力レコード内の位置によって特定されるテキスト・フィールドまたは 2 進数フィールド、またはその両方のフィールドのシーケンスであることを指定します。各フィールドの位置は、フィールド定義内の POSITION 節で指定します。

## RECORDLEN *length*

入力データが固定長レコードであり、各レコードが指定の長さ (バイト) であることを指定します。レコード内の復帰または改行文字 (X'0D' または X'0A') は、データの一部と見なされず。一部のエディターでは、これらの文字は表示に含まれないため、場合によって、エディターに表示されているレコード長に 1 または 2 を加算する必要が生じることに注意してください。入力にバイナリー・データが含まれている場合、このパラメーターは必須です。

範囲は 1 から 32,767 です。RECORDLEN に指定した長さが、各フィールド定義に指定したフィールド長の合計より短い場合、INGEST ユーティリテ

イーは、エラー・メッセージを出して終了します。RECORDLEN に指定した長さが、最長フィールドの終了位置を超えている場合、INGEST ユーティリティーは、最長終了位置のフィールドの最後より後ろのバイトを無視します。最後のレコードが短すぎる場合、INGEST ユーティリティーは、エラー・メッセージを出し、そのレコードを拒否します。

このパラメーターを省略すると、INGEST ユーティリティーは、プラットフォームにかかわらず、次の両方をレコード (行) 区切り文字として認識します。

- 改行 (X'0A')
- 改行 (X'0D0A') が後に続く 1 つ以上の復帰

#### **INPUT CODEPAGE** *integer*

入力データのコード・ページを指定します。INGEST ユーティリティーは、関連リンクのセクションの『サポートされているテリトリー・コードおよびコード・ページ』のトピックにリストされているコード・ページをサポートします。ただし、以下の値を除きます。

- 1200 (UTF-16)
- 「ホスト」のコード・ページとしてリストされているコード・ページ

デフォルトは DB2 アプリケーションのコード・ページです。

入力データが、このパラメーターに指定されたコード・ページのものでもなくとも、INGEST ユーティリティーは、エラーを出しません。ただし、INGEST ユーティリティーまたは DB2 によって入力データを別のコード・ページに変換する必要がある場合、一部の文字が、置換文字 (Unicode U+001A) または意図しない他の文字に変換される可能性があります。詳しくは、『INGEST ユーティリティーに関するコード・ページの考慮事項』のトピックを参照してください。

#### **IMPLICITLYHIDDENINCLUDE**

暗黙的な非表示列がデフォルトの列リストに含まれるように指定します。クライアント・サイドで DB2\_DMU\_DEFAULT レジストリー変数も指定されている場合には、このキーワードが優先されます。このキーワードは、表の名前の後に列リストが存在しない INSERT ステートメントに対してのみ許可されます。SQL ステートメントが INSERT ではない場合や表の名前の後に列リストがある場合には、エラーが返されます (SQLCODE -2918)。

#### **IMPLICITLYHIDDENMISSING**

暗黙的な非表示列がデフォルトの列リストから除外されるように指定します。クライアント・サイドで DB2\_DMU\_DEFAULT レジストリー変数も指定されている場合には、このキーワードが優先されます。このキーワードは、表の名前の後に列リストが存在しない INSERT ステートメントに対してのみ許可されます。SQL ステートメントが INSERT ではない場合や表の名前の後に列リストがある場合には、エラーが返されます (SQLCODE -2918)。

#### **field-list**

フィールド定義のリストを指定します。このリストを省略すると、デフォルトのフィールド定義リストの説明のように、デフォルト設定されます。



*\$field-name*

フィールド名の長さは、2 から 129 バイト (ドル記号を含む) であり、SQL 識別子と同じ規則に従います。また、`"My Field Name"` のように、引用符で区切られている名前の前にドル記号を指定することによって、引用符で区切られているフィールド名を指定できます

SQL ステートメントでドル記号で始まる表名や列名を指定する場合、表名または列名を区切り文字で区切られた ID として指定してください。例として、次の各識別子を、INGEST ユーティリティがどのように解釈するかを示します。

表 19. フィールド名 ID の意味

ID	意味
<code>\$name</code>	フィールド「 <code>\$NAME</code> 」
<code>\$NAME</code>	フィールド「 <code>\$NAME</code> 」
<code>"My Name"</code>	フィールド「 <code>\$My Name</code> 」
<code>"\$name"</code>	列「 <code>\$name</code> 」(小文字)
<code>"\$NAME"</code>	列「 <code>\$NAME</code> 」(大文字)
<code>"\$My Name"</code>	列「 <code>\$My Name</code> 」

**INGEST** コマンドを UNIX のシステム・コマンド・ラインに指定する場合は、フィールド名が環境変数として扱われないように、コマンドを単一引用符で囲む必要があります。詳しくは、354 ページの『システム・コマンド行の INGEST の使用方法 (UNIX)』を参照してください。

**POSITION** (*start:end*)

この節は、形式が **POSITIONAL** である場合に限り有効です。これは、レコード内のフィールドの開始位置および終了位置 (オプション) を指定します。単位はバイトです。レコードの最初のバイトが位置 1 です。終了位置を指定する場合は、開始位置より大か等しい値でなければなりません。

異なるフィールドに指定した位置が、オーバーラップしていても構いません。例えば、米国州の完全な名前が 1 から 10 までの位置にある場合、次の 2 つのフィールドを定義できます。

- `$state_full_name POSITION(1:10)`
- `$state_abbreviation POSITION(1:2)`

指定のフィールド位置の外にデータが存在していても、INGEST ユーティリティは無視します。例えば、フィールド定義リストに `POSITION(1:10)` および `POSITION(21:30)` の 2 つのフィールドを指定した場合、INGEST ユーティリティは、位置 11:20 のデータと、位置 30 より後からレコードの終わりまでのデータを無視します。

**POSITION** 節を省略すると、前のフィールドの次のバイトが、開始位置と見なされます。最初のフィールドについては、位置 1 が開始位置と見なされます。終了位置を省略した場合、360 ページの『フィールド長の規則およびデフォルト』の説明のように、ユーティリティは、デフォルトを使用するか、またはエラーを出します。

フィールド定義で長さが指定されていない場合、INGEST ユーティリティーは  $\text{end\_position} - \text{start\_position} + 1$  の長さを使用します。この長さは次の制限を受けます。

- タイプが SMALLINT、INTEGER、BIGINT、DECIMAL、REAL、FLOAT、および DECFLOAT であり、EXTERNAL 修飾子を持つフィールドでは、最大長は 50 です。
- タイプが DATE、TIME、および TIMESTAMP(p) のフィールドにフォーマット・ストリングが指定されている場合、そのフィールドの長さはフォーマット・ストリングに一致する最小値の長さ以上である必要があります。
- フォーマット・ストリングが指定されていない場合は、次のようになります。
  - DATE の場合、フィールド長は 8 以上 10 以下である必要があります。
  - TIME の場合、フィールド長は 4 以上 8 以下である必要があります。
  - TIMESTAMP の場合、フィールド長は 19 以上 32 以下である必要があります。

#### field-type

フィールドのデータ・タイプを指定します。

#### CHARACTER (*length*)

文字データ。コード・ページは INPUT CODEPAGE パラメーターで指定されます。デフォルトはアプリケーション・コード・ページです。

デフォルトの長さについては、360 ページの『フィールド長の規則およびデフォルト』を参照してください。

#### NOTRIM, LTRIM, RTRIM, TRIM

ストリング区切り文字で囲まれていない、フィールド内の先行空白 (LTRIM)、末尾空白 (RTRIM)、または先行空白と末尾空白の両方 (TRIM) を、ストリングに含めないように指定します。形式が DELIMITED の場合のデフォルトは TRIM です。形式が POSITIONAL の場合のデフォルトは NOTRIM です。このパラメーターをいずれかの文字フィールドに指定する場合は、他のすべての文字フィールドにも同じ値がなければなりません。

#### OPTIONALLY ENCLOSED BY *char*

この節は、形式が DELIMITED である場合に限り有効です。ストリングを、ここに指定した文字で囲むことができます。制約事項は、フィールド区切り文字の場合と同じです。また、この文字は、フィールド区切り文字または小数点と同じ文字にはできません。文字、日付、時刻、またはタイム・スタンプのいずれかのフィールドにこのパラメーターを指定する場合は、これらの他のすべてのフィールドでも同じ値でなければなりません。

ユーティリティーでは、以下のアルゴリズムを使用して、値の先頭が区切り文字であるかどうかを判別します。

- TRIM または LTRIM オプションが指定されている (またはデフォルトである) 場合、ユーティリティーは先行ブランクのトリミング後に、フィールド内の先頭文字を検査します。
- RTRIM または NOTRIM オプションが指定されている場合、ユーティリティーはフィールド内の先頭文字を検査します。(すなわち、RTRIM または NOTRIM が指定されている場合、ユーティリティーはストリング区切り文字を見つける際に先行ブランクを飛ばしません。)

ユーティリティーは、レコード区切り文字を除き、囲み文字の間にあるすべての文字を、ストリングの一部と見なします。ストリング内に囲み文字がある場合は、そのストリング内で 2 回重ねて入力する必要があります。例えば、ストリングが `don't know` であり、囲み文字が単一引用符である場合は、`'don't know'` と指定します。ストリングの末尾が囲み文字でない場合、ユーティリティーは開始区切り文字をストリングの一部と見なします。

ユーティリティーは、以下の状況では警告 `SQL3114W` を出します。

- TRIM または RTRIM が指定されていて (またはデフォルト)、終了区切り文字と次のフィールド区切り文字 (またはレコードの終わり) との間に、余分な非ブランク文字がある。
- NOTRIM または LTRIM が指定されていて、終了区切り文字と次のフィールド区切り文字 (またはレコードの終わり) との間に、余分な文字 (ブランクを含む) がある。

このパラメーターを指定しない場合、デフォルトは `OPTIONALLY ENCLOSED BY '''` です。その場合、値の先頭が二重引用符でないのであれば、ユーティリティーは、ストリングの周囲にある (フィールド区切り文字以外の) すべての文字をストリングの一部と見なします。

#### FOR BIT DATA

フィールドにビット・データが入っていることを示し、また、INGEST ユーティリティーで、そのデータを入力データ・コード・ページから INGEST ユーティリティー・アプリケーション・コード・ページに変換しないように指定します。このフィールドを使用して、FOR BIT DATA 指定の列に割り当てる場合は、フィールド定義にも FOR BIT DATA と指定することを強くお勧めします。そうしないと、INGEST ユーティリティーが時によって行うコード・ページ変換の最適化方法によって、列に割り当てられる値が予測不能になります。

#### INTEGER/SMALLINT/BIGINT EXTERNAL (*length*)

数値を表す最大 *length* バイトの文字ストリング。このストリングは、整数、10 進数、または浮動小数点定数にすることができます。このストリングに 10 進数または浮動小数点の値が指定されていると、INGEST ユーティリティーは、小数部分を切り捨てることによって、指定された整数タイプに変換します。ストリングが有効な数値ではない場合、または数値が範囲外



である場合、INGEST ユーティリティーは、エラー・メッセージを出し、そのレコードを拒否します。詳しくは、367 ページの『無効な数値データの処理』を参照してください。

数値フィールド・タイプに EXTERNAL を使用すると、フィールド値がバイナリーではなく ASCII 文字で指定されていることを示します。いずれかの整数フィールドに EXTERNAL を指定する場合は、他のすべての整数、10 進浮動小数点、または浮動小数点フィールドにも指定する必要があります。

デフォルトの長さについては、360 ページの『フィールド長の規則およびデフォルト』を参照してください。

#### **DECIMAL** (*precision, scale*) **EXTERNAL** (*length*)

10 進数を表す最大 *length* バイトの文字ストリング。このストリングは、整数、10 進数、または浮動小数点定数にすることができます。INGEST ユーティリティーは、指定された精度および位取りを使用して、ストリングを 10 進数値に変換します。ストリングに、フィールド・タイプに指定された位取りよりも大きい位取りの 10 進数値が指定されていると、INGEST ユーティリティーは、警告もエラーも出さずに、その値の小数部分を切り捨てます。フィールド・タイプに指定した位取りよりも小さい位取りの 10 進数値がストリングに指定されていると、INGEST ユーティリティーは、その値の小数部分にゼロを埋め込みます。ストリングが有効な数値ではない場合、または数値が範囲外である場合 (位取りの範囲ではない)、INGEST ユーティリティーは、エラー・メッセージを出し、そのレコードを拒否します。詳しくは、367 ページの『無効な数値データの処理』を参照してください。

精度には、1 から 31 までを指定できます。位取りには、0 から精度の値までを指定できます。デフォルトは (5,0) です。デフォルトの長さについて詳しくは、360 ページの『フィールド長の規則およびデフォルト』を参照してください。

いずれかの 10 進フィールドに EXTERNAL を指定する場合は、他のすべての 10 進フィールドにも指定する必要があります。

#### **DECFLOAT** (16|34) **EXTERNAL** (*length*)

10 進浮動小数点数を表す最大 *length* バイトの文字ストリング。このストリングは、整数、10 進数、浮動小数点定数、または 10 進浮動小数点特殊値にすることができます。INGEST ユーティリティーは、このストリングを、指定された精度の 10 進浮動小数点値に変換します。ストリングが有効な数値または 10 進浮動小数点特殊値ではない場合、あるいは数値が範囲外である場合、INGEST ユーティリティーは、エラー・メッセージを出し、そのレコードを拒否します。詳しくは、367 ページの『無効な数値データの処理』を参照してください。精度は、16 または 34 (デフォルト) のいずれかにすることができます。

デフォルトの長さについて詳しくは、360 ページの『フィールド長の規則およびデフォルト』を参照してください。

いずれかの 10 進浮動小数点フィールドに EXTERNAL を指定する場合は、他のすべての整数、10 進浮動小数点、または浮動小数点フィールドにも指定する必要があります。

**FLOAT**(*integer*) **EXTERNAL**(*length*)

浮動小数点数を表す最大 *length* バイトの文字ストリング。このストリングは、整数、10 進数、または浮動小数点定数にすることができます。ストリングが有効な数値ではない場合、または数値が範囲外である場合、INGEST ユーティリティーは、エラー・メッセージを出し、そのレコードを拒否します。詳しくは、367 ページの『無効な数値データの処理』を参照してください。

*integer* の値は、1 から 53 の範囲の整数でなければなりません。1 から 24 の値は単精度、25 から 53 の値は倍精度を示します。デフォルトは 53 です。また、以下を指定することもできます。

**REAL** 単精度浮動小数点数。

**DOUBLE、DOUBLE PRECISION、または FLOAT**

倍精度浮動小数点数。

いずれかの浮動小数点フィールドに **EXTERNAL** を指定する場合は、他のすべての整数、10 進浮動小数点、または浮動小数点フィールドにも指定する必要があります。

デフォルトの長さについては、360 ページの『フィールド長の規則およびデフォルト』を参照してください。

**RADIX POINT** *char*

小数点として使用する文字。制約事項は、フィールド区切り文字の場合と同じです。また、この文字は、フィールド区切り文字またはストリング区切り文字と同じ文字にはできません。いずれかの整数、10 進数、10 進浮動小数点、または浮動小数点フィールドに、このパラメーターを指定する場合は、他のすべての整数、10 進数、10 進浮動小数点、または浮動小数点フィールドにも同じ値がなければなりません。デフォルトはドット `.` です。

**INTEGER、SMALLINT、または BIGINT**

指定されたタイプのバイナリー整数。これらのタイプを指定できるのは、形式が **POSITIONAL** であり、**RECORDLEN** パラメーターを指定する場合のみです。また、次を指定することもできます。

**BIG | LITTLE ENDIAN**

整数が、ビッグ・エンディアン形式 (アドレスが低い方が最上位バイト) またはリトル・エンディアン形式 (アドレスが低い方が最下位バイト) のどちらであるかを指定します。このオプションは、整数、浮動小数点、および 10 進浮動小数点タイプに適用されますが、10 進タイプには適用されません。デフォルトは、INGEST ユーティリティーを実行中の現行ハードウェア・プラットフォームのデフォルト形式です。

**DECIMAL** (*precision, scale*) **PACKED | ZONED**

指定された精度および位取りの、パック形式またはゾーン形式の 10 進数。これらのタイプを指定できるのは、形式が **POSITIONAL** であり、**RECORDLEN** パラメーターを指定する場合のみです。精度には、1 から 31 までを指定できます。位取りには、0 から精度の値までを指定できます。デフォルトは (5,0) です。この **PACKED/ZONED** 設定は、他のすべての 10 進フィールドで同じでなければなりません。デフォルトは **PACKED** です。

**DECFLOAT(16|34)**

指定された精度のバイナリー 10 進浮動小数点数。精度は、16 または 34 (デフォルト) のいずれかにすることができます。このタイプを指定できるのは、形式が **POSITIONAL** であり、**RECORDLEN** パラメーターを指定する場合のみです。

**FLOAT(integer)**

指定されたサイズのバイナリーの単精度または倍精度浮動小数点数。(IEEE 浮動小数点数形式)。integer の値は、1 から 53 の範囲の整数でなければなりません。1 から 24 の値は単精度、25 から 53 の値は倍精度を示します。デフォルトは 53 です。これらのタイプを指定できるのは、形式が **POSITIONAL** であり、**RECORDLEN** パラメーターを指定する場合のみです。また、以下を指定することもできます。

**REAL** 単精度浮動小数点数。

**DOUBLE、DOUBLE PRECISION、または FLOAT**

倍精度浮動小数点数。

**DATE, TIME, TIMESTAMP**

文字列として指定された日付、時刻、またはタイム・スタンプ。データベースの **date\_compat** データベース構成パラメーターが **ON** に設定されている場合、**DATE** は **TIMESTAMP(0)** と等価です。

デフォルトの長さについては、360 ページの『フィールド長の規則およびデフォルト』を参照してください。

**精度** タイム・スタンプの精度を指定します。範囲は 0 から 12 です。デフォルトは 6 です。入力データに、フィールド・タイプに指定された精度よりも高い精度のタイム・スタンプが指定されていると、**INGEST** ユーティリティーは、警告もエラーも出さずに、そのタイム・スタンプの小数部分を切り捨てます (そのタイム・スタンプに、書式制御文字列で指定された書式が含まれている場合)。入力データに、フィールド・タイプに指定された精度よりも低い精度のタイム・スタンプが指定されていると、**INGEST** ユーティリティーは、そのタイム・スタンプの小数部分にゼロを埋め込みます。

**文字列**

日付、時刻、またはタイム・スタンプ値の書式を指定します。この書式制御文字列の要素は、現在 **IMPORT** および **LOAD** コマンドでサポートされている要素と同じです。詳しくは、「**IMPORT** コマンド」トピックの **dateformat**、**timeformat**、および **timestampformat** ファイル・タイプ修飾子の説明を参照してください。

形式が **POSITIONAL** であり、**POSITION** パラメーターに指定された長さよりも文字列が長い場合、**INGEST** ユーティリティーはエラーを出します。例えば、文字列が 'yyyy-mm-dd' (10 バイト) であるのに、位置が 1:6 である場合、**INGEST** ユーティリティーは、エラーを出します。

入力データ内の日付、時刻、またはタイム・スタンプの値は、この書式に一致していなければなりません。ただし、右側のサブフィー

ルドは省略することができます。例えば、タイム・スタンプの書式が「yyyy/mm/dd hh:mm:ss.uuu」である場合、次のタイム・スタンプは有効です。

```
2010/06/03 12:34
2010/06/03 12:34:56
2010/06/03 12:34:56.1
2010/06/03 12:34:56.12
2010/06/03 12:34:56.123
```

しかし、「2010/06/03 12:34:56.1234」は有効ではありません。サブフィールドを省略すると、「IMPORT コマンド」トピックの `dateformat`、`timeformat`、および `timestampformat` ファイル・タイプ修飾子の説明にあるように、これらはデフォルト設定されます。

サブフィールドの省略に関する上記の説明には、例外が 1 つあります。形式が `POSITIONAL` で、フィールドがレコードの最後のフィールドである場合に、レコードがフィールドの終了位置より手前で終了していると、ユーティリティは警告を出し、その値として `NULL` を使用します。

日付、時刻、タイム・スタンプの値が指定の書式ではない場合 (右側のサブフィールドの欠如は除く)、INGEST ユーティリティは、エラー・メッセージを出し、その行を拒否します。

また、以下の規則も適用されます。

- このパラメーターをいずれかの `DATE` フィールドに指定する場合は、他のすべての `DATE` フィールドについても同じ値を指定しなければなりません。データベースの `date_compat` データベース構成パラメーターが `ON` に設定されている場合 (この場合、フィールド・タイプ `DATE` は `TIMESTAMP(0)` と等価です) は、他のすべての `TIMESTAMP` フィールドについても同じ値を指定しなければなりません。
- これをいずれかの `TIME` フィールドに指定する場合は、他のすべての `TIME` フィールドにも同じ値を指定しなければなりません。
- これをいずれかの `TIMESTAMP` フィールドに指定する場合は、他のすべての `TIMESTAMP` フィールドにも同じ値を指定しなければなりません。データベースの `date_compat` データベース構成パラメーターが `ON` に設定されている場合 (この場合、フィールド・タイプ `DATE` は `TIMESTAMP(0)` と等価です) は、他のすべての `DATE` フィールドについても同じ値を指定しなければなりません。
- 

フォーマット・ストリングが指定されていない場合、345 ページの表 20 に示すように、INGEST ユーティリティは様々なフォーマットを識別します。

表 20. DATE、TIME、および TIMESTAMP のデフォルト値

フィールド・タイプ	デフォルトの書式
DATE	<ul style="list-style-type: none"> <li>• <b>date_compat</b> データベース構成パラメーターが OFF に設定されている場合は、以下のとおりです。 <ul style="list-style-type: none"> <li>- 'yyyymmdd' (DELIMITED 形式のみ)</li> <li>- 'yyyy-mm-dd'</li> <li>- 'mm/dd/yyyy'</li> <li>- 'dd.mm.yyyy'</li> </ul> </li> <li>注: 1 番目以外の形式では、'yyyy-m-d' のように月および日を 1 桁で指定することも可能です。</li> <li>• <b>date_compat</b> データベース構成パラメーターが ON に設定されている場合、TIMESTAMP(0) として同じデフォルトが使用されます。</li> </ul>
TIME	<ul style="list-style-type: none"> <li>• 'hh.mm.ss'</li> <li>• 'hh:mm tt'</li> <li>• 'hh:mm:ss'</li> </ul>
TIMESTAMP	<p>'yyyy-mm-dd-hh.mm.ss.aaaaaaaaaaaa' または 'yyyy-mm-dd hh:mm:ss.aaaaaaaaaaaa'</p> <ul style="list-style-type: none"> <li>• 'yyyy-mm-dd-hh.mm.ss.aaaaaaaaaaaa'</li> <li>• 'yyyy-mm-dd hh:mm:ss.aaaaaaaaaaaa'</li> <li>• 'yyyy-mm-dd'</li> <li>• 'mm/dd/yyyy'</li> <li>• 'dd.mm.yyyy'</li> <li>• 'yyyymmdd' (DELIMITED 形式のみ)</li> </ul> <p>注: 1 番目以外の形式では、'yyyy-m-d-hh.mm.ss.aaaaaaaaaaaa' のように月および日を 1 桁で指定することも可能です。</p>

**OPTIONALLY ENCLOSED BY char**

この節は、形式が DELIMITED である場合に限り有効です。ストリングを、ここに指定した文字で囲むことができます。制約事項は、フィールド区切り文字の場合と同じです。また、この文字は、フィールド区切り文字または小数点と同じ文字にはできません。文字、日付、時刻、またはタイム・スタンプのいずれかのフィールドにこのパラメーターを指定する場合は、これらの他のすべてのフィールドでも同じ値でなければなりません。

ユーティリティでは、以下のアルゴリズムを使用して、値の先頭が区切り文字であるかどうかを判別します。

- **TRIM** または **LTRIM** オプションが指定されている (またはデフォルトである) 場合、ユーティリティは先行ブランクのトリミング後に、フィールド内の先頭文字を検査します。

- **RTRIM** または **NOTRIM** オプションが指定されている場合、ユーティリティーはフィールド内の先頭文字を検査します。(すなわち、**RTRIM** または **NOTRIM** が指定されている場合、ユーティリティーはストリング区切り文字を見つける際に先行空白を飛ばしません。)

ユーティリティーは、レコード区切り文字を除き、囲み文字の間にあるすべての文字を、ストリングの一部と見なします。ストリング内に囲み文字がある場合は、そのストリング内で 2 回重ねて入力する必要があります。例えば、ストリングが `don't know` であり、囲み文字が単一引用符である場合は、`'don't know'` と指定します。ストリングの末尾が囲み文字でない場合、ユーティリティーは開始区切り文字をストリングの一部と見なします。

ユーティリティーは、以下の状況では警告 **SQL3114W** を出します。

- **TRIM** または **RTRIM** が指定されていて (またはデフォルト)、終了区切り文字と次のフィールド区切り文字 (またはレコードの終わり) との間に、余分な非空白文字がある。
- **NOTRIM** または **LTRIM** が指定されていて、終了区切り文字と次のフィールド区切り文字 (またはレコードの終わり) との間に、余分な文字 (空白を含む) がある。

このパラメーターを指定しない場合、デフォルトは **OPTIONALLY ENCLOSED BY ''** です。その場合、値の先頭が二重引用符でないのであれば、ユーティリティーは、ストリングの周囲にある (フィールド区切り文字以外の) すべての文字をストリングの一部と見なします。

#### **DB2SECURITYLABEL** (*length*)

DB2 セキュリティー・ラベルを指定します。NAME または **STRING** のいずれも指定されておらず、形式が **POSITIONAL** である場合、デフォルトの形式はエンコードされた数値形式です。それ以外の場合は、エラーが戻されます。

デフォルトの長さについては、360 ページの『フィールド長の規則およびデフォルト』を参照してください。

#### **NAME**

DB2 セキュリティー・ラベルは、その名前を使用して指定します。形式が **DELIMITED** の場合は、NAME または **STRING** のいずれかを指定する必要があります。このパラメーターをいずれかの **DB2SECURITYLABEL** フィールドに指定する場合は、他のすべての **DB2SECURITYLABEL** フィールドにも同じ値がなければなりません。表を保護するセキュリティ・ポリシーで、指定された名前のセキュリティ・ラベルが存在しない場合、行はロードされず、警告が戻されます。

#### **STRING**

DB2 セキュリティー・ラベルは、ストリング・フォーマットで指定します。形式が **DELIMITED** の場合は、NAME または **STRING** のいずれかを指定する必要があります。このパラメーターをいずれか



の DB2SECURITYLABEL フィールドに指定する場合は、他のすべての DB2SECURITYLABEL フィールドにも同じ値がなければなりません。ストリングが適切な形式ではない場合、行はロードされず、警告が戻されます。ストリングが表を保護するセキュリティー・ポリシーの一部である有効なセキュリティー・ラベルを表していない場合、行はロードされず、警告が戻されます。

デフォルトは、エンコードされた数値形式であり、形式が POSITIONAL である場合に限り有効です。

#### DEFAULTIF (*position*) = *char*

位置 (*position*) はオプションであり、形式が POSITIONAL の場合にのみ指定できます。位置が指定されている場合、ユーティリティーは、指定またはデフォルトのトリム・オプションを適用する前に、入力レコードから該当文字を読み取ります。この位置は、フィールド内でなくても構いません。位置が省略されている場合、ユーティリティーは、指定またはデフォルトのトリム・オプションを適用した後に、フィールドの先頭文字を読み取ります。いずれの場合も、入力レコードから読み取った文字が指定の文字と一致していると、ユーティリティーは、そのフィールドの値を、フィールドの割り当て先の表の列のデフォルト値であると見なします。

DEFAULTIF パラメーターには、以下の制限があります。

- 位置を省略する場合は、DEFAULTIF と等号記号との間に、少なくとも 1 つスペースを入れる必要があります。そうでない場合は、構文エラーが発生します。
- DEFAULTIF の文字には、次の値を除く任意の 1 バイト文字を使用できます。
  - ヌル文字 (X'00')
  - 復帰または改行 (X'0D' または X'0A')

入力データのコード・ページが DBCS または EUC コード・ページである場合、この文字は X'01' から X'3F' までの範囲内 (境界値を含む) になければなりません。入力データのコード・ページが 1 バイト・コード・ページではない場合、この文字は X'01' から X'7F' までの範囲内 (境界値を含む) になければなりません。入力データのコード・ページが 1 バイト・コード・ページである場合、この文字は X'01' から X'FF' までの範囲内にすることができます。

- フィールドを割り当てる列は、1 つのみでなければならず、また、フィールドを、列に割り当てる式の一部にすることはできません。例えば、以下の INGEST コマンドについて考えてみます。

```
INGEST FROM FILE ...
UPDATE my_table
  SET (c1, c2, c3, c4, c5) =
      ($field1, $field2, $field2, $field3, $field4+$field5)
  WHERE $field3 = 1;
```

\$field1 にのみ DEFAULTIF を指定できます。\$field2 は、複数列に割り当てられているため指定できません。\$field3 は、述部でも使用されているため指定できません。\$field4 および \$field5 は、式で使用されているため指定できません。

- 表の列のデフォルト値は、定数または NULL でなければなりません。特殊レジスターにすることはできません。

注: 列がヌル可能である (つまり列定義に NOT NULL が指定されていない) 場合は、WITH DEFAULT 節が列定義に指定されていなくても、その列は NULL のデフォルト値を持っていることになります。

- 表の列のタイプが、フィールド・タイプと異なる場合、INGEST ユーティリティーは、入力データをフィールド・タイプに変換するときと同じ規則を適用して、デフォルト値をフィールド・タイプに変換します。例えば、表の列に FLOAT WITH DEFAULT 3.7 と定義されていて、フィールドに INTEGER DEFAULTIF ... と定義されている場合、INGEST ユーティリティーはデフォルト値 3 を使用します。同様に、表の列のデフォルト値の長さがフィールド長より長い場合、ユーティリティーは、デフォルト値をフィールド長に切り捨てます。

これらの制約事項に違反している場合、INGEST ユーティリティーはエラーを出し、その INGEST コマンドは終了します。

## 例

### 基本的な INGEST の例

次の例では、区切り文字で区切られているテキスト・ファイルからデータを挿入します。

```
INGEST FROM FILE my_file.txt
  FORMAT DELIMITED
  INSERT INTO my_table;
```

次の例では、フィールドがコンマ (デフォルト) で区切られているテキスト・ファイルからデータを挿入します。ファイル内の各フィールドは、表の各列に対応しています。

```
INGEST FROM FILE my_file.txt
  FORMAT DELIMITED
  (
    $field1 INTEGER EXTERNAL,
    $field2 DATE 'mm/dd/yyyy',
    $field3 CHAR(32)
  )
  INSERT INTO my_table
  VALUES($field1, $field2, $field3);
```

### 区切り文字のオーバーライドの例

次の例では、前の例と類似したデータを挿入していますが、垂直バーを使用してフィールドが区切られています。

```
INGEST FROM FILE my_file.txt
  FORMAT DELIMITED by '|'
  (
    $field1 INTEGER EXTERNAL,
    $field2 DATE 'mm/dd/yyyy',
    $field3 CHAR(32)
  )
  INSERT INTO my_table
  VALUES($field1, $field2, $field3);
```



## フィールド定義および VALUES リストを省略する例

次の例では、表は以下のように定義されています。

```
CREATE TABLE my_table (
  c1 VARCHAR(32),
  c2 INTEGER GENERATED BY DEFAULT AS IDENTITY,
  c3 INTEGER GENERATED ALWAYS AS (c2 + 1),
);
```

ユーザーは次の **INGEST** コマンドを発行します。

```
INGEST FROM FILE my_file.txt
  FORMAT DELIMITED
  INSERT INTO mytable;
```

- デフォルトのフィールド定義リストは次のようになります。

```
(
  $C1 CHARACTER(32),
  $C2 INTEGER EXTERNAL,
  $C3 INTEGER EXTERNAL
)
```

- INSERT ステートメントのデフォルトの VALUES リストは、次のとおりです。

```
VALUES($C1, $C2, DEFAULT)
```

フィールド \$C3 に対応する列は GENERATED ALWAYS と定義されているため、3 番目の値が DEFAULT になっていることに注意してください。4 番目の値は、フィールドがないため省略されています。

## 余分なフィールドを使用して列値を計算する例

次の例は、区切り文字のオーバーライドの例と同じですが、最初の 2 つのフィールドのみが、表の最初の 2 列 (PROD\_ID および DESCRIPTION) に対応していて、表の 3 番目の列 (TOTAL\_PRICE) の値は、残りの 3 つのフィールドから計算しています。

```
INGEST FROM FILE my_file.txt
  FORMAT DELIMITED BY '|'
  (
    $prod_ID    CHAR(8),
    $description CHAR(32),
    $price      DECIMAL(5,2) EXTERNAL,
    $sales_tax  DECIMAL(4,2) EXTERNAL,
    $shipping   DECIMAL(3,2) EXTERNAL
  )
  INSERT INTO my_table(prod_ID, description, total_price)
  VALUES($prod_id, $description, $price + $sales_tax + $shipping);
```

## 充てん文字フィールドの例

次の例では、フィールドがコンマ (デフォルト) で区切られているテキスト・ファイルからデータを挿入します。ファイル内の各フィールドは、列 2 と 3 の間、および列 3 と 4 の間に余分なフィールドが存在している点を除けば、表の各列に対応しています。

```
INGEST FROM FILE my_file.txt
  FORMAT DELIMITED
  (
    $field1  INTEGER,
    $field2  CHAR(8),
    $filler1 CHAR,
```

```

$field3 CHAR(32),
$filler2 CHAR,
$field4 DATE
)
INSERT INTO my_table VALUES($field1, $field2, $field3, $field4);

```

### 形式修飾子の例

次の例では、区切り文字で区切られている、コード・ページ 850 のテキスト・ファイルからデータを挿入しています。日付フィールドは米国形式であり、文字フィールドは等号で囲まれています。

```

INGEST FROM FILE my_file.txt
FORMAT DELIMITED
INPUT CODEPAGE 850
(
$field1 INTEGER EXTERNAL,
$field2 DATE 'mm/dd/yyyy',
$field3 CHAR(32) OPTIONALLY ENCLOSED BY '='
)
INSERT INTO my_table
VALUES($field1, $field2, $field3);

```

### 定位置の例

次の例では、指定された位置に各フィールドが存在するファイルから、データを挿入しています。ファイル内の各フィールドは、表の各列に対応しています。

```

INGEST FROM FILE my_file.txt
FORMAT POSITIONAL
(
$field1 POSITION(1:8) INTEGER EXTERNAL,
$field2 POSITION(10:19) DATE 'yyyy-mm-dd',
$field3 POSITION(25:34) CHAR(10)
)
INSERT INTO my_table
VALUES($field1, $field2, $field3);

```

### DEFAULTIF の例

この例は、前の例に類似していますが、2番目のフィールドの先頭が空白である場合に、INGEST ユーティリティがデフォルト値を挿入する点が異なっています。

```

INGEST FROM FILE my_file.txt
FORMAT POSITIONAL
(
$field1 POSITION(1:8) INTEGER EXTERNAL,
$field2 POSITION(10:19) DATE 'yyyy-mm-dd' DEFAULTIF = ' ',
$field3 POSITION(25:34) CHAR(10)
)
INSERT INTO my_table
VALUES($field1, $field2, $field3);

```

この例は、前の例と同じですが、データ列の後の列に、デフォルト・インディケータがある点が異なっています。

```

INGEST FROM FILE my_file.txt
FORMAT POSITIONAL
(
$field1 POSITION(1:8) INTEGER EXTERNAL,
$field2 POSITION(10:19) DATE 'yyyy-mm-dd' DEFAULTIF(35) = ' ',

```

```

    $field3 POSITION(25:34) CHAR(10)
)
INSERT INTO my_table
VALUES($field1, $field2, $field3);

```

## 複数入力ソースの例

次の例では、区切り文字で区切られている 3 つのテキスト・ファイルからデータを挿入します。

```

INGEST FROM FILE my_file.txt, my_file2.txt, my_file3.txt
FORMAT DELIMITED
(
  $field1 INTEGER EXTERNAL,
  $field2 DATE 'mm/dd/yyyy',
  $field3 CHAR(32)
)
INSERT INTO my_table
VALUES($field1, $field2, $field3);

```

## パイプ例

この例では、パイプからデータを挿入します。

```

INGEST FROM PIPE my_pipe
FORMAT DELIMITED
(
  $field1 INTEGER EXTERNAL,
  $field2 DATE 'mm/dd/yyyy',
  $field3 CHAR(32)
)
INSERT INTO my_table
VALUES($field1, $field2, $field3);

```

## オプションの例

この例では、フィールドがコンマ (デフォルト) で区切られているテキスト・ファイルからデータを挿入します。ファイル内の各フィールドは、表の各列に対応しています。このコマンドは、(制約違反などによって) DB2 から拒否された行を表 EXCP\_TABLE に書き込み、他のエラーによって拒否された行を破棄し、メッセージをファイル messages.txt に書き込むように指定しています。

```

INGEST FROM FILE my_file.txt
FORMAT DELIMITED
(
  $field1 INTEGER EXTERNAL,
  $field2 DATE 'mm/dd/yyyy',
  $field3 CHAR(32)
)
EXCEPTION TABLE excp_table
  MESSAGES messages.txt
INSERT INTO my_table
VALUES($field1, $field2, $field3);

```

## 再開の例

この例では、INGEST ジョブ ID を指定して、INGEST コマンド (デフォルトで再開可能) を実行しています。

```

INGEST FROM FILE my_file.txt
FORMAT DELIMITED
(
  $field1 INTEGER EXTERNAL,

```

```

$field2 DATE 'mm/dd/yyyy',
$field3 CHAR(32)
)
RESTART NEW 'ingestcommand001'
INSERT INTO my_table
VALUES($field1, $field2, $field3);

```

このコマンドが完了せずに終了した場合は、次のコマンドを使用して再開することができます。

```

INGEST FROM FILE my_file.txt
FORMAT DELIMITED
(
$field1 INTEGER EXTERNAL,
$field2 DATE 'mm/dd/yyyy',
$field3 CHAR(32)
)
RESTART CONTINUE 'ingestcommand001'
INSERT INTO my_table
VALUES($field1, $field2, $field3);

```

### 再開を終了させる例

この例では、前述の『再開の例』と同じ **INGEST** コマンドを実行しています。

```

INGEST FROM FILE my_file.txt
FORMAT DELIMITED
(
$field1 INTEGER EXTERNAL,
$field2 DATE 'mm/dd/yyyy',
$field3 CHAR(32)
)
RESTART NEW 'ingestcommand001'
INSERT INTO my_table
VALUES($field1, $field2, $field3);

```

このコマンドが完了せずに終了した場合に、再開させる予定がなければ、次のコマンドを使用して、再開レコードをクリーンアップできます。

```

INGEST FROM FILE my_file.txt
FORMAT DELIMITED
(
$field1 INTEGER EXTERNAL,
$field2 DATE 'mm/dd/yyyy',
$field3 CHAR(32)
)
RESTART TERMINATE 'ingestcommand001'
INSERT INTO my_table
VALUES($field1, $field2, $field3);

```

このコマンドを実行すると、ジョブ ID 「ingestcommand001」を指定して **INGEST** コマンドを再開できなくなりますが、このストリングは、新しい **INGEST** コマンドの **RESTART NEW** パラメーターに再利用できるようになります。

### 列の再配列の例

この例では、フィールドがコンマで区切られているテキスト・ファイルからデータを挿入します。表には 3 つの列があり、入力データ内の各フィールドは、表の列とは逆の順序で並んでいます。

```

INGEST FROM FILE my_file.txt
FORMAT DELIMITED
(
$field1 INTEGER EXTERNAL,

```

```

$field2 DATE 'mm/dd/yyyy',
$field3 CHAR(32)
)
INSERT INTO my_table
VALUES($field3, $field2, $field1);

```

## 基本的な UPDATE、MERGE、および DELETE の例

次の例では、主キーが、入力ファイル内の対応するフィールドと一致する場合に、その表の行を更新します。

```

INGEST FROM FILE my_file.txt
FORMAT DELIMITED
(
  $key1 INTEGER EXTERNAL,
  $key2 INTEGER EXTERNAL,
  $data1 CHAR(8),
  $data2 CHAR(32),
  $data3 DECIMAL(5,2) EXTERNAL
)
UPDATE my_table
SET (data1, data2, data3) = ($data1, $data2, $data3)
WHERE (key1 = $key1) AND (key2 = $key2);

```

または

```

INGEST FROM FILE my_file.txt
FORMAT DELIMITED
(
  $key1 INTEGER EXTERNAL,
  $key2 INTEGER EXTERNAL,
  $data1 CHAR(8),
  $data2 CHAR(32),
  $data3 DECIMAL(5,2) EXTERNAL
)
UPDATE my_table
SET data1 = $data1, data2 = $data2, data3 = $data3
WHERE (key1 = $key1) AND (key2 = $key2);

```

この例では、入力ファイル内のデータを、ターゲット表にマージします。主キー・フィールドが表の行と一致する入力行については、その入力行を使用して表の行を更新します。他の入力行については、表に追加します。

```

INGEST FROM FILE my_file.txt
FORMAT DELIMITED
(
  $key1 INTEGER EXTERNAL,
  $key2 INTEGER EXTERNAL,
  $data1 CHAR(8),
  $data2 CHAR(32),
  $data3 DECIMAL(5,2) EXTERNAL
)
MERGE INTO my_table
ON (key1 = $key1) AND (key2 = $key2)
WHEN MATCHED THEN
  UPDATE SET (data1, data2, data3) = ($data1, $data2, $data3)
WHEN NOT MATCHED THEN
  INSERT VALUES($key1, $key2, $data1, $data2, $data3);

```

この例では、主キーが、入力ファイル内の対応する主キー・フィールドと一致する場合に、その表の行を削除します。

```

INGEST FROM FILE my_file.txt
FORMAT DELIMITED
(

```

```

$key1 INTEGER EXTERNAL,
$key2 INTEGER EXTERNAL
)
DELETE FROM my_table
WHERE (key1 = $key1) AND (key2 = $key2);

```

## 複合 SQL の例

列 KEY、DATA、および ACTION を持つ表が存在する次の例を考えてみましょう。次のコマンドは、主キー列 (KEY) が、入力ファイル内の対応するフィールドと一致していて、ACTION 列が「U」である場合に、その表の行の DATA 列を更新します。

```

INGEST FROM FILE my_file.txt
FORMAT DELIMITED
(
  $key_fld INTEGER EXTERNAL,
  $data_fld INTEGER EXTERNAL
)
UPDATE my_table
SET data = $data_fld
WHERE (key = $key_fld) AND (action = 'U');

```

次の例は、前の例と同じですが、キーが一致していて ACTION 列が「D」である場合に、その行を表から削除する点が異なります。

```

INGEST FROM FILE my_file.txt
FORMAT DELIMITED
(
  $key_fld INTEGER EXTERNAL,
  $data_fld INTEGER EXTERNAL
)
MERGE INTO my_table
ON (key1 = $key_fld)
WHEN MATCHED AND (action = 'U') THEN
  UPDATE SET data = $data_fld
WHEN MATCHED AND (action = 'D') THEN
  DELETE;

```

## 使用上の注意

### システム・コマンド行の INGEST の使用方法 (UNIX)

INGEST コマンドを、UNIX コマンド・シェルのシステム・コマンド行に指定する場合は、フィールド名を単一引用符で囲むか、または、フィールド名の先頭の \$ の前にエスケープ文字 (円記号) を入れる必要があります。そうしないと、コマンド・シェルがフィールド名を環境変数として解釈し、対応する環境変数の値に置換します。

例えば、環境変数 *\$field1* の値が abc であり、環境変数 *\$field2* には値がない場合に、次のコマンドを UNIX コマンド・シェルで実行したとします。

```
db2 INGEST ... FORMAT DELIMITED ($field1 INTEGER, $field2 CHAR) ...
```

DB2 CLP は、次のコマンドとして解釈しますが、このコマンドは有効ではありません。

```
INGEST ... FORMAT DELIMITED (abc INTEGER, CHAR) ...
```

これは、次のように単一引用符を使用することで回避できます。

```
db2 'INGEST ... FORMAT DELIMITED ($field1 INTEGER, $field2 CHAR) ...'
```

**INGEST** コマンド内に単一引用符を含める場合は、コマンド全体を二重引用符 (") で囲み、フィールド名先頭の \$ の前にエスケープ文字 (\\$) を使用することができます。以下に例を示します。

```
db2 "INGEST ... FORMAT DELIMITED BY X'7C' (\$field1 ...) ..."
```

上記の問題をすべて回避する最も簡単な方法は、コマンドをファイルに入力し、db2 コマンドに `-f` オプションを指定することです。

## コミット頻度

**INGEST** コマンドによる更新は、**INGEST** 操作の最後にコミットされます。**INGEST** コマンドは、`commit_period` および `commit_count` 構成パラメーターに基づいて、コミットを発行します。その結果、以下のオプションが **INGEST** コマンドに影響を与えることがなくなります。

- `CLP -c` または `+c` オプション。通常であれば、`CLP` が自動的にコミットするかどうかに影響を与えます。
- `CREATE TABLE` ステートメントの `NOT LOGGED INITIALLY` オプション

## エラーの区分

**INGEST** ユーティリティーによるエラー処理を明確にするという目的で、エラーを次のカテゴリーに分類できます。

### 1. 開始時のエラー

それらのエラーは、以下のとおりです。

- 構文エラー
- 入力ファイルが見つからない、または読み取り可能ではない
- ターゲット表または例外表が見つからない
- ダンプ・ファイルもメッセージ・ファイルも開けない
- 十分なメモリーがない
- 開始中に検出される他のエラー

ユーティリティーはこのようなエラーを検出すると、エラー・メッセージを出し、データを **INGEST** せずに終了します。

### 2. データのエラー

それらのエラーは、以下のとおりです。

- フォーマッターによって検出された入力データ内のエラー。以下に例を示します。
  - 無効な数値または範囲外の数値 (フィールド・タイプに基づく)
  - 指定された書式と合わない日付、時刻、およびタイム・スタンプ
  - フォーマッターによって検出される他のエラー
- **SQL** ステートメントのエラー。これについては、**INGEST** コマンドの `EXCEPTION TABLE` パラメーターの説明中にリストしています。最も一般的なものは以下のとおりです。
  - 長すぎる、範囲外である、または形式が誤っているデータ (フォーマッターでは検出されないもの)

- NOT NULL 制約、ユニーク制約、参照制約、チェック制約などのターゲット表の制約違反
- ターゲット表の範囲違反

ユーティリティーは、このようなエラーを検出すると、警告またはエラー・メッセージを出して処理を続行します。また、フォーマッター・エラーの場合は、ユーティリティーは該当レコードをダンプ・ファイルに書き込みます (指定されている場合)。SQL エラーの場合は、ユーティリティーは該当行を例外表に挿入します (指定されている場合)。これらのエラーが特定の回数発生した場合に、ユーティリティーを終了させるには、WARNINGCOUNT パラメーターを使用します。

### 3. リカバリー可能エラー

これは、ユーティリティーを再接続し (必要な場合)、しばらく待機してから再実行すれば解決する可能性があるというエラーです。このようなエラーをユーティリティーが検出した場合、**retry\_count** または **reconnect\_count** 構成パラメーターが 0 より大きければ、ユーティリティーはエラーからのリカバリーを試みます。**retry\_count** および **reconnect\_count** が 0 である場合、またはユーティリティーがリカバリーできない場合、ユーティリティーはこれらのエラーを強制終了エラー と見なします。

### 4. 強制終了エラー

これは、3 つのタイプとして上記にリストしたエラー以外のすべてのエラーです。ユーティリティーは、このようなエラーを検出すると、エラー・メッセージを出して終了します。コミット済みのデータは、ターゲット表内に残されます。最後のコミット・ポイントからコマンドを再開することができます。

## INGEST コマンドからのメッセージ

ユーティリティーが入力ソースから少なくとも 1 レコードを読み取っていた場合は、読み取った行数、挿入した行数、および拒否した行数の要約 (import および load ユーティリティーと類似したもの) および正常完了メッセージを出します。以下に例を示します。

```
INGEST FROM FILE my_table.del FORMAT DELIMITED
($field1 INTEGER_EXTERNAL, $field2 CHAR(32))
INSERT INTO my_table VALUES($field1, $field2)
Number of rows read      = 6
Number of rows inserted = 6
Number of rows rejected = 0
SQL2980I The ingest utility completed successfully at timestamp "11/02/2011 12:34:56.123456".
```

「Number of rows...」メッセージの意味は以下のとおりです。

#### Number of rows read

ユーティリティーが入力ソースから読み取ったレコードの数。

#### Number of rows inserted (updated, deleted, merged)

ターゲット表に対する SQL ステートメントの実行によって影響を受け、データベースにコミットされた行の数。メッセージには、SQL ステートメントに応じて「inserted」、「updated」、「deleted」、または「merged」と表示されます。

#### Number of rows rejected

(ユーティリティーまたは DB2 によって) 拒否された行の数。



警告またはエラーが 1 つしかなく、そのエラーが、処理できる入力レコードがありませんというようなエラーである場合、コマンドは、エラー・メッセージのみを出します。読み取った行数、拒否した行数などの要約は表示されません。以下に例を示します。

```
INGEST FROM FILE bad_file.del FORMAT DELIMITED
($field1 INTEGER EXTERNAL, $field2 CHAR)
INSERT INTO my_table VALUES($field1, $field2)
```

SQL2036N The path for the file or device "bad\_file.del" is not valid.

**INGEST** コマンドがレコードの読み取りを開始した後に、1 つ以上の警告またはエラー状態が検出された場合には、読み取った行数、スキップした行数などの要約、ならびに警告およびエラーの数の要約と一緒に、メッセージがコマンドから出力されます。重複する 2 つのキーを挿入しようとする以下の例について考えてみましょう。

```
INGEST FROM FILE my_table1.del, my_table2.del FORMAT DELIMITED
($field1 integer external, $field2 char(32) )
INSERT INTO my_table VALUES($field1, $field2)
```

SQL2905I The following error occurred issuing the SQL "INSERT" statement on table "ALVEYWD.MY\_TABLE" using data from line "6" of input file "my\_table1.del".

SQL0803N One or more values in the INSERT statement, UPDATE statement, or foreign key update caused by a DELETE statement are not valid because the primary key, unique constraint or unique index identified by "1" constrains table "ALVEYWD.MY\_TABLE" from having duplicate values for the index key. SQLSTATE=23505

SQL2905I The following error occurred issuing the SQL "INSERT" statement on table "ALVEYWD.MY\_TABLE" using data from line "9" of input file "my\_table2.del".

SQL0803N One or more values in the INSERT statement, UPDATE statement, or foreign key update caused by a DELETE statement are not valid because the primary key, unique constraint or unique index identified by "1" constrains table "ALVEYWD.MY\_TABLE" from having duplicate values for the index key. SQLSTATE=23505

```
Number of rows read      = 6
Number of rows inserted = 4
Number of rows rejected  = 2
Number of rows committed = 4
```

SQL2902I The ingest utility completed at timestamp "11/02/2011 12:34:56.123456".  
Number of errors: 2. Number of warnings: 0.

**MESSAGES** パラメーターを指定すると、このパラメーターの説明に明記されているものを除き、ほとんどのメッセージがファイルに書き込まれます。

## ファイル名およびパイプ名

メタ変数 *file-name* および *pipe-name* は、オペレーティング・システムの構文に従ってファイル名またはパイプ名を指定するものです。 **INGEST** コマンドに指定するファイル名またはパイプ名は、次の文字以外の文字が含まれている場合、単一引用符で囲む必要があります。

- 英数字
- 下線 ()
- ダッシュ (-)
- スラッシュ (/) または円記号 (¥)
- ドット (.)

引用符は、名前全体を囲む必要があります。例えば、ファイル名に等号 (=) が含まれている場合は、次のように指定します。

```
'dir1/dir2/my=file'
```

次のように指定するものではありません。

```
dir1/dir2/'my=file'
```

## デフォルトのフィールド定義リスト

次のすべてが当てはまる場合は、フィールド定義リストを省略できます。

- 区切り文字で区切られている形式。
- SQL ステートメントは INSERT
- VALUES 節が INSERT ステートメントから省略されている。

フィールド定義リストは、次のようにデフォルト設定されます。

- INSERT ステートメントの表名の後に列リストがある場合は、そのリスト内の各列について、対応する 1 つのフィールドがある。
- INSERT ステートメントで列リストを省略し、暗黙的な非表示列がない場合、その表の各列にはフィールドが 1 つずつある。
- INSERT ステートメントで列リストを省略し、暗黙的な非表示列がある場合、暗黙的な非表示列を含めるかどうかを明示的に指定する必要がある。  
DB2\_DMU\_DEFAULT レジストリー変数、または IMPLICITLYHIDDENINCLUDE キーワードか IMPLICITLYHIDDENMISSING キーワードを使用して、暗黙的な非表示列を含めるかどうかを指定します。
- 各フィールドの名前は、対応する表の列の名前に、ドル記号 \$ を接頭部として付加したものである。
- 各フィールドは、対応する表の列と同じデータ・タイプおよびデータ長 (または精度および位取り) である。数値フィールド (整数、10 進など) はデフォルトで EXTERNAL 形式となる。DB2SECURITYLABEL フィールドは、デフォルトで STRING 形式となる。

359 ページの表 21 に、フィールド、列、および値の各リストを省略したり指定したりする状況として考えられる全通りの組み合わせにおいて、それらのリストがどのようにデフォルト設定されるかを示します。

表 21. フィールド定義リストが指定または省略される場合に考えられる組み合わせ

フィールド定義リストの指定	列リストの指定	VALUES 節の指定	デフォルト
no	no	no	フィールドのリストおよび列のリストには、すべての列がデフォルト設定されます。値のリストには、フィールド・リストがデフォルト設定されます。 <b>注:</b> 表に暗黙的な非表示列が含まれる場合、DB2_DMU_DEFAULT レジストリー変数、または IMPLICITLYHIDDENINCLUDE キーワードか IMPLICITLYHIDDENMISSING キーワードを使用して、暗黙的な非表示列を含めるかどうかを明示的に指定する必要があります。
no	no	yes	不可
no	yes	no	フィールド・リストおよび値リストには、指定の列リストがデフォルト設定されます。
no	yes	yes	不可
yes	no	no	列のリストには、すべての列がデフォルト設定されます。  値のリストには、ユーザー指定のフィールド・リストがデフォルト設定されます。 <b>注:</b> 1. フィールドの数は、列の数と等しくなければなりません。それ以外の場合、エラーが発生します。 2. 表に暗黙的な非表示列が含まれる場合、DB2_DMU_DEFAULT レジストリー変数、または IMPLICITLYHIDDENINCLUDE キーワードか IMPLICITLYHIDDENMISSING キーワードを使用して、暗黙的な非表示列を含めるかどうかを明示的に指定する必要があります。

表 21. フィールド定義リストが指定または省略される場合に考えられる組み合わせ (続き)

フィールド定義リストの指定	列リストの指定	VALUES 節の指定	デフォルト
yes	no	yes	列のリストには、すべての列がデフォルト設定されます。 注: 表に暗黙的な非表示列が含まれる場合、DB2_DMU_DEFAULT レジストリー変数、または IMPLICITLYHIDDENINCLUDE キーワードか IMPLICITLYHIDDENMISSING キーワードを使用して、暗黙的な非表示列を含めるかどうかを明示的に指定する必要があります。
yes	yes	no	値のリストには、ユーザー指定のフィールド・リストがデフォルト設定されます。 注: フィールド数は、列数と等しくなければなりません。それ以外の場合、エラーが発生します。
yes	yes	yes	デフォルトなし

## フィールド長の規則およびデフォルト

フィールド・タイプに指定するフィールド長はバイト単位で、1 から 32 767 まで (両端を含む) でなければなりません。

以下の各表に、入力データのフォーマット、フィールド・タイプ、およびフィールド長と終了位置の指定の有無に関するすべての組み合わせにおけるフィールド長を示します。セルの値にある指定の長さ は、例えば「INTEGER EXTERNAL」の後に指定できる長さのように、明示的に指定されたフィールド長を意味します。一部の数値タイプで使用される精度を意味するものではありません。セルの値である *n/a* は、適用外または不可を意味します。

表 22. *DELIMITED* フォーマットのフィールド長

フィールド・タイプ	フィールド・タイプの長さを省略 (または日時フォーマット) <sup>1</sup>	フィールド・タイプに長さを指定 (または日時フォーマット) <sup>1</sup>
CHARACTER	255 <sup>2</sup>	指定の長さ
SMALLINT、INTEGER、BIGINT EXTERNAL	255 <sup>2</sup>	指定の長さ
DECIMAL(p,s) EXTERNAL ((p,s) のデフォルトは (5,0))	255 <sup>2</sup>	指定の長さ
DECFLOAT(16 または 34) EXTERNAL	255 <sup>2</sup>	指定の長さ
REAL EXTERNAL	255 <sup>2</sup>	指定の長さ

表 22. *DELIMITED* フォーマットのフィールド長 (続き)

フィールド・タイプ	フィールド・タイプの長さを省略 (または日時フォーマット) <sup>1</sup>	フィールド・タイプに長さを指定 (または日時フォーマット) <sup>1</sup>
FLOAT または DOUBLE EXTERNAL	255 <sup>2</sup>	指定の長さ
DATE	<ul style="list-style-type: none"> <li>• <b>date_compat</b> off: date の最大長デフォルト・フォーマットの長さ (10)</li> <li>• <b>date_compat</b> on: timestamp (32) の最大長デフォルト・フォーマットの長さ</li> </ul>	<ul style="list-style-type: none"> <li>• <b>date_compat</b> off: date の最大長デフォルト・フォーマットの長さ (10)</li> <li>• <b>date_compat</b> on: timestamp (32) の最大長デフォルト・フォーマットの長さ</li> </ul>
TIME	最大長デフォルト・フォーマットの長さ (8)	最大長デフォルト・フォーマットの長さ (8)
TIMESTAMP(p) (p のデフォルトは 6)	最大長デフォルト・フォーマットの長さ (32)	最大長デフォルト・フォーマットの長さ (32)
NAME 修飾子を指定した DB2SECURITYLABEL	255 <sup>2</sup>	指定の長さ
STRING 修飾子を指定した DB2SECURITYLABEL	255 <sup>2</sup>	指定の長さ

<sup>1</sup> 区切り文字で区切られたフォーマットでは、フィールド長とは、ユーティリティが長さ限度を超えたフィールド・データの切り捨てを行う場合の、限度となる長さを意味します。切り捨ては、トリム・オプションを適用した場合に行われます。切り捨てによって非ブランク文字が失われた場合、ユーティリティは警告を出します。

<sup>2</sup> 区切り文字で区切られたフォーマットでは、ASCII で指定した文字フィールドおよび数値フィールドは、長さを省略した場合、デフォルトの長さ 255 バイトとなります。より長くする必要がある場合は、フィールド・タイプに明示的に長さを指定してください。

表 23. *POSITIONAL* フォーマットのフィールド長

フィールド・タイプ	フィールド・タイプの長さを省略 (または日付/時刻フォーマット)、 <b>POSITION</b> の終了位置を省略	フィールド・タイプに長さを指定 (または日付/時刻フォーマット)、 <b>POSITION</b> の終了位置を省略	フィールド・タイプの長さを省略 (または日付/時刻フォーマット)、 <b>POSITION</b> に終了位置を指定	フィールド・タイプに長さを指定 (または日付/時刻フォーマット)、 <b>POSITION</b> に終了位置を指定
CHARACTER	n/a <sup>3</sup>	指定の長さ	(end - start + 1)	指定の長さとして (end - start + 1) のうち短い方 <sup>7</sup>
SMALLINT (バイナリー)	2	n/a <sup>4</sup>	(終了位置 - 開始位置 + 1) が 2、4、または 8 の場合は、これを使用します。それ以外の場合はバイナリー形式の長さを使用します。矛盾する場合は警告を出します。 <sup>5</sup>	n/a <sup>4</sup>

表 23. POSITIONAL フォーマットのフィールド長 (続き)

フィールド・タイプ	フィールド・タイプの長さを省略 (または日付/時刻フォーマット)、 POSITION の終了位置を省略	フィールド・タイプに長さを指定 (または日付/時刻フォーマット)、 POSITION の終了位置を省略	フィールド・タイプの長さを省略 (または日付/時刻フォーマット)、 POSITION に終了位置を指定	フィールド・タイプに長さを指定 (または日付/時刻フォーマット)、 POSITION に終了位置を指定
INTEGER (バイナリー)	4	n/a <sup>4</sup>	(end - start + 1) が 2、4、または 8 の場合は、これを使用します。それ以外の場合はバイナリー形式の長さを使用します。矛盾する場合は警告を出します。 <sup>5</sup>	n/a <sup>4</sup>
BIGINT (バイナリー)	8	n/a <sup>4</sup>	(end - start + 1) が 2、4、または 8 の場合は、これを使用します。それ以外の場合はバイナリー形式の長さを使用します。矛盾する場合は警告を出します。 <sup>5</sup>	n/a <sup>4</sup>
SMALLINT、INTEGER、BIGINT EXTERNAL	n/a <sup>3</sup>	指定の長さ。これは <= 50 でなければなりません	(end - start + 1) が <= 50 でなければなりません。	指定の長さ (end - start + 1) のうち短い方 <sup>7</sup>
DECIMAL(p,s) PACKED ((p,s) のデフォルトは (5,0))	(p+2)/2	n/a <sup>4</sup>	(p+2)/2 を使用します。終了位置と矛盾する場合は警告を出します。 <sup>5</sup>	n/a <sup>4</sup>
DECIMAL(p,s) ZONED ((p,s) のデフォルトは (5,0))	p	n/a <sup>4</sup>	精度を使用します。終了位置と矛盾する場合は警告を出します。 <sup>5</sup>	n/a <sup>4</sup>
DECIMAL(p,s) EXTERNAL ((p,s) のデフォルトは (5,0))	n/a <sup>3</sup>	指定の長さ	(end - start + 1)	指定の長さ (end - start + 1) のうち短い方 <sup>7</sup>
DECFLOAT(16) (バイナリー)	8	n/a <sup>4</sup>	8 を使用します。終了位置と矛盾する場合は警告を出します。 <sup>5</sup>	n/a <sup>4</sup>
DECFLOAT(34) (バイナリー)	16	n/a <sup>4</sup>	16 を使用します。終了位置と矛盾する場合は警告を出します。 <sup>5</sup>	n/a <sup>4</sup>
DECFLOAT(16) または 34) EXTERNAL	n/a <sup>3</sup>	指定の長さ	(end - start + 1)	指定の長さ (end - start + 1) のうち短い方 <sup>7</sup>

表 23. POSITIONAL フォーマットのフィールド長 (続き)

フィールド・タイプ	フィールド・タイプの長さを省略 (または日付/時刻フォーマット)、 POSITION の終了位置を省略	フィールド・タイプに長さを指定 (または日付/時刻フォーマット)、 POSITION の終了位置を省略	フィールド・タイプの長さを省略 (または日付/時刻フォーマット)、 POSITION に終了位置を指定	フィールド・タイプに長さを指定 (または日付/時刻フォーマット)、 POSITION に終了位置を指定
REAL (バイナリー)	4	n/a <sup>4</sup>	4 を使用します。終了位置と矛盾する場合は警告を出します。 <sup>5</sup>	n/a <sup>4</sup>
REAL EXTERNAL	n/a <sup>3</sup>	指定の長さ	(end - start + 1)	指定の長さ (end - start + 1) のうち短い方 <sup>7</sup>
FLOAT または DOUBLE (バイナリー)	8	n/a <sup>4</sup>	8 を使用します。終了位置と矛盾する場合は警告を出します。 <sup>5</sup>	n/a <sup>4</sup>
FLOAT または DOUBLE EXTERNAL	n/a <sup>3</sup>	指定の長さ	(end - start + 1)	指定の長さ (end - start + 1) のうち短い方 <sup>7</sup>
DATE	<ul style="list-style-type: none"> <li>• <b>date_compat</b> off: date の最大長デフォルト・フォーマットの長さ (10)</li> <li>• <b>date_compat</b> on: TIMESTAMP (0) の最大長デフォルト・フォーマットの長さ (19)</li> </ul>	フォーマット・ストリングの長さ。これは $\geq 1$ でなければなりません	<ul style="list-style-type: none"> <li>• <b>date_compat</b> off: (end - start + 1) が <math>\geq 8</math> かつ <math>\leq 10</math> でなければなりません。</li> <li>• <b>date_compat</b> on: (end - start + 1) が <math>\geq 19</math> かつ <math>\leq 32</math> でなければなりません。</li> </ul>	(end - start + 1) が、フォーマット・ストリングの長さ以上である場合は、これを使用します。そうでない場合は、(end - start + 1) が、フォーマット・ストリングと一致する最も短い値の長さ以上でなければなりません。 <sup>6</sup>
TIME	最大長デフォルト・フォーマットの長さ (8)	フォーマット・ストリングの長さ。これは $\geq 1$ でなければなりません	(end - start + 1) が $\geq 4$ かつ $\leq 8$ でなければなりません。	(end - start + 1) が、フォーマット・ストリングの長さ以上である場合は、これを使用します。そうでない場合は、(end - start + 1) が、フォーマット・ストリングと一致する最も短い値の長さ以上でなければなりません。 <sup>6</sup>

表 23. POSITIONAL フォーマットのフィールド長 (続き)

フィールド・タイプ	フィールド・タイプの長さを省略 (または日付/時刻フォーマット)、 POSITION の終了位置を省略	フィールド・タイプに長さを指定 (または日付/時刻フォーマット)、 POSITION の終了位置を省略	フィールド・タイプの長さを省略 (または日付/時刻フォーマット)、 POSITION に終了位置を指定	フィールド・タイプに長さを指定 (または日付/時刻フォーマット)、 POSITION に終了位置を指定
TIMESTAMP(p) (p のデフォルトは 6)	デフォルト・フォーマット長 (p = 0 の場合は 19。それ以外の場合は 20 + p。)	フォーマット・ストリングの長さ。これは $\geq 1$ でなければなりません	( $end - start + 1$ ) は $\geq 19$ かつ $\leq 32$ でなければなりません。	( $end - start + 1$ ) が、フォーマット・ストリングの長さ以上である場合は、これを使用します。そうでない場合は、( $end - start + 1$ ) が、フォーマット・ストリングと一致する最も短い値の長さ以上でなければなりません。 <sup>6</sup>
DB2SECURITYLABEL (バイナリー)	n/a <sup>3</sup>	指定の長さ	(終了位置 - 開始位置 + 1)	指定の長さとして ( $end - start + 1$ ) のうち短い方 <sup>7</sup>
NAME 修飾子を指定した DB2SECURITYLABEL	n/a <sup>3</sup>	指定の長さ	(終了位置 - 開始位置 + 1)	指定の長さとして ( $end - start + 1$ ) のうち短い方 <sup>7</sup>
STRING 修飾子を指定した DB2SECURITYLABEL	n/a <sup>3</sup>	指定の長さ	(終了位置 - 開始位置 + 1)	指定の長さとして ( $end - start + 1$ ) のうち短い方 <sup>7</sup>

<sup>3</sup> これらのケース (定位置フォーマットで、長さとして終了位置が省略されている) は許可されません。意図されたフィールド長をユーティリティが判別できず、また適切なデフォルト値も存在しないためです。

<sup>4</sup> これらのケース (定位置フォーマット、数値バイナリー・フィールド、長さの指定あり、終了位置は省略) は許可されません。バイナリー数値タイプの場合、長さはバイナリー・タイプによって暗黙指定されるため、バイト単位のフィールド長を明示的に構文に指定することはできません。

<sup>5</sup> 各種ユーティリティでは、指定された数値バイナリー形式と、POSITION パラメーターに指定された長さとの間に矛盾がある場合、365 ページの表 24 に示されているように処理します。

<sup>6</sup> 各種ユーティリティでは、日付/時刻フォーマットのストリング長と、POSITION パラメーターに指定された長さとの間に矛盾がある場合、365 ページの表 25 に示されているように処理します。

<sup>7</sup> これらのケース (長さおよび終了位置が指定されているが矛盾している) では、INGEST ユーティリティは警告を出し、指定された長さとして ( $end - start + 1$ ) の値のうち短い方を使用します。



表 24. 数値バイナリー形式と、指定された長さとの間の矛盾

フィールド・タイプ	ロード <sup>1</sup>	INGEST
SMALLINT、INTEGER、BIGINT	( $end - start + 1$ ) が 2、4、または 8 の場合は、警告もエラーも出さずに、この長さを使用します。それ以外の場合は警告を出し、NULL を使用するか、またはその行を拒否します。	( $end - start + 1$ ) が 2、4、または 8 の場合は、その長さを使用して警告を出します。それ以外の場合は、バイナリー形式の長さを使用して警告を出します。
DECIMAL	エラー SQL3123W を出してその行を拒否します。	バイナリー形式の長さを使用して警告を出します。
REAL、FLOAT、DECFLOAT	バイナリー形式の長さを使用します。警告もエラーも出しません。	バイナリー形式の長さを使用して警告を出します。
<sup>1</sup> インポートはバイナリー数値タイプ (MODIFIED BY binarynumerics) をサポートしません。		

表 25. 日付/時刻フォーマットのストリング長と、指定された長さとの間の矛盾

競合	インポート/ロード	INGEST
( $end - start + 1$ ) が、フォーマット・ストリングの長さ以上である。	$end - start + 1$ を使用します。警告もエラーも出しません。	$end - start + 1$ を使用します。警告もエラーも出しません。
( $end - start + 1$ ) がフォーマット・ストリングの長さより短い。	エラーを出します。	エラーを出します。

## INGEST ユーティリティーがフィールド長を決定する方法

フィールド長を決定するアルゴリズムは、次のとおりです。

1. 入力データ内に出現するフィールドの長さを調べます。
  - フォーマットが **DELIMITED** である場合、次のようにします。
    - 指定またはデフォルトのトリム・オプションを適用します (CHAR フィールドの場合、デフォルトは **TRIM** です。他のすべての非バイナリー・フィールド・タイプについては、INGEST ユーティリティーは、常に、先行ブランクおよび末尾ブランクをトリムします)。
    - **OPTIONALLY ENCLOSED BY** が指定されていない場合は、データ長は、フィールドに残ったバイト数となります。
    - **OPTIONALLY ENCLOSED BY** が指定されている場合、データ長は、(ストリング内の 2 回重ねて入力された区切り文字を 1 つの区切り文字に変換した後の) ストリング区切り文字に囲まれているバイト数となります。
  - フォーマットが **POSITIONAL** である場合、次のようにします。
    - 指定またはデフォルトのトリム・オプションを適用します (CHAR フィールドの場合、デフォルトは **NOTRIM** です。他のすべての非バイナリー・フィールド・タイプについては、INGEST ユーティリティーは、常に、先行ブランクおよび末尾ブランクをトリムします)。

2. ステップ 1 で決定された長さが 360 ページの『フィールド長の規則およびデフォルト』で説明したフィールド長より長い場合は、以下のようになります。
  - DATE、TIME、または TIMESTAMP 以外のフィールドの場合は、フィールド長になるようにデータを切り捨てます。非空白文字が切り捨てられた場合は、警告が出ます。
  - DATE、TIME、および TIMESTAMP フィールドの場合は、エラーを出しません。
3. SQL ステートメントで列にフィールド値を設定する場合に、値が長すぎたり範囲外にあると、DB2 はエラー・メッセージを出し、その行を拒否します。

注: これは、INSERT、UPDATE、または MERGE ステートメントの場合のみ、発生する可能性があります。

次の各表に、例をいくつか示します。これらの表では、波形記号 (~) が空白を意味しています。

表 26. *FORMAT DELIMITED BY 'I'*

フィールド定義	入力フィールド	最終的なフィールド値
CHAR TRIM	~ABC~	ABC
CHAR NOTRIM	~ABC~	~ABC~
CHAR OPTIONALLY ENCLOSED BY '''	~"ABC"~	ABC
CHAR OPTIONALLY ENCLOSED BY '''	~"~ABC~"~	~ABC~
CHAR NOTRIM OPTIONALLY ENCLOSED BY '''	"~ABC~"	~ABC~
CHAR NOTRIM OPTIONALLY ENCLOSED BY '''	~"ABC"~	~"ABC"~
CHAR(1) TRIM	~ABC~	A (非空白がトリミングされたため警告が出ます)
CHAR(1) NOTRIM	~ABC~	~ (非空白がトリミングされたため警告が出ます)
CHAR(1) OPTIONALLY ENCLOSED BY '''	~ABC~	A (非空白がトリミングされたため警告が出ます)
CHAR(6) TRIM	~ABC~	ABC
CHAR(6) NOTRIM	~ABC~	~ABC~
CHAR(6) OPTIONALLY ENCLOSED BY '''	~"ABC"~	ABC
CHAR(6) OPTIONALLY ENCLOSED BY '''	~"~ABC~"~	~ABC~
INTEGER EXTERNAL	~12345~	12345
INTEGER EXTERNAL(3)	~12345~	123 (非空白がトリミングされたため警告が出ます)
INTEGER EXTERNAL(9)	~12345~	12345

表 27. *FORMAT POSITIONAL*

フィールド定義	入力フィールド	最終的なフィールド値
POSITION(1:5) CHAR TRIM	~ABC~	ABC
POSITION(1:5) CHAR NOTRIM	~ABC~	~ABC~
POSITION(1:5) CHAR(3) TRIM	ABC	A (終了位置が長さとは矛盾しているため警告が出ます)
POSITION(1:5) CHAR(5) NOTRIM	~ABC~	~ABC~
POSITION(1:5) CHAR(5) TRIM	~ABC~	ABC
POSITION(1:5) CHAR(7) NOTRIM	~ABC~	~ABC~ (終了位置が長さとは矛盾しているため警告が出ます)
POSITION(1:7) INTEGER EXTERNAL	~12345~	12345
POSITION(1:7) INTEGER EXTERNAL(3)	123	123 (終了位置が長さとは矛盾しているため警告が出ます)
POSITION(1:7) INTEGER EXTERNAL(9)	~12345~	12345 (終了位置が長さとは矛盾しているため警告が出ます)

## 無効な数値データの処理

一般的に、数値データが文字ストリングとして指定された場合に、データを数値に変換できなかつたり、データがフィールド・タイプの範囲外にあったりすると、INGEST ユーティリティは、そのレコードを拒否してエラーを出します。これは、インポートおよびロード・ユーティリティと異なる点です。これらのユーティリティは、このようなエラーを以下のように処理します。

- すべての数値タイプにおいて、データを列のタイプに変換できない場合 (整数フィールド内に "ABC" がある場合など) は、以下のようにします。
  - 列が NULL 可能な場合、インポートおよびロード・ユーティリティは、その値を NULL 値に置き換えて警告を出します。
  - 列が NULL 可能でない場合、インポートおよびロード・ユーティリティは、そのレコードを拒否してエラーを出します。
- DECIMAL を除くすべての数値タイプにおいて、データが範囲外である場合は、以下のようにします。
  - 列が NULL 可能な場合、インポートおよびロード・ユーティリティは、その値を NULL 値に置き換えて警告を出します。
  - 列が NULL 可能でない場合、インポートおよびロード・ユーティリティは、そのレコードを拒否してエラーを出します。
- DECIMAL フィールドのデータが範囲外である場合、インポートおよびロード・ユーティリティは、そのレコードを拒否してエラーを出します。

## FORMAT 節に関するメモ

- 区切り文字の優先順位は、ロード・ユーティリティと同じく、レコード、文字、フィールドの順です。インポートおよびロード・ユーティリティでは、こ

の優先順位を文字、レコード、フィールドの順に変更する `delprioritychar` ファイル・タイプ修飾子もサポートされていますが、INGEST ユーティリティではサポートされていません。

- 修飾子をフィールドに指定する場合は、その修飾子を同様のタイプのフィールドすべてに指定する必要があります。例えば、10 進フィールドに `RADIX POINT` を指定する場合は、同じ `RADIX POINT` 文字を他のすべての `INTEGER`、`DECIMAL`、`DECLFOAT`、および `FLOAT` タイプのフィールドにも指定する必要があります。
- 文字フィールドの場合は、さまざまな `TRIM` オプションを使用して、先行空白、末尾空白、またはその両方を INGEST ユーティリティによって削除するかどうかを指定します。他のすべての非バイナリー・フィールド・タイプ (`INTEGER EXTERNAL` など) については、INGEST ユーティリティは、常に、先行空白および末尾空白をトリムします。
- 形式が `DELIMITED` である場合、または形式が `POSITIONAL` であって `RECORDLEN` パラメーターが省略されている場合
  - フォーマットに定義されているよりも少ない数のフィールドしかレコードに含まれていない場合、INGEST ユーティリティは、欠落しているフィールドを `NULL` と見なします。
  - レコードに余分なフィールドが含まれている場合、INGEST ユーティリティはそれらを見捨てます。
- 形式が `POSITIONAL` である場合
  - `RECORDLEN` パラメーターが指定されていて、最後のレコードが短すぎる場合、INGEST ユーティリティは、エラーを出し、そのレコードを拒否します。
  - 指定のフィールド位置の外にデータが存在していても、INGEST ユーティリティは無視します。例えば、`POSITION(1:10)` および `POSITION(21:30)` の 2 つのフィールドを指定した場合、INGEST ユーティリティは、位置 11:20 のデータと、位置 30 からレコードの終わりまでのデータを無視します。

## INGEST コマンドの SQL ステートメント

- INGEST ユーティリティは、`PERIOD` 節で次の値をサポートします。
  - 定数
  - 特殊レジスター

注: 以下の特殊レジスターのみが、INGEST コマンドの SQL ステートメントの実行に影響を与えます。

- `CURRENT SCHEMA`
- `CURRENT TEMPORAL SYSTEM_TIME`
- `CURRENT TEMPORAL BUSINESS_TIME`
- サポートされるオペランドが引数に指定されたスカラー関数 (ただし、ネストされた関数呼び出しおよびユーザー定義関数は使用できません)。
- キャストのオペランドが、算術演算子およびオペランドを使用する、サポートされるオペランド式である `CAST` 指定
- 算術演算子および算術オペランドを使用する式

また、フィールド名を指定することもできます (フィールド名は、ユーティリティによってパラメーター・マーカに変換されます)。

- テンポラル表への INGEST には、他にもいくつかの考慮事項があります。
  - 特殊レジスター CURRENT TEMPORAL SYSTEM\_TIME が設定されている場合、システム期間テンポラル表およびバイテンポラル表に INGEST することはできません。
  - 特殊レジスター CURRENT TEMPORAL BUSINESS\_TIME が設定されている場合であっても、ビジネス期間テンポラル表またはバイテンポラル表 (システム期間テンポラル表に関する他の制限は受けず) に INGEST することができます。
  - システム期間テンポラル表またはバイテンポラル表に REPLACE または DELETE 操作を使用することはできません。
  - INGEST ユーティリティーは、システム期間テンポラル表の SYSTEM\_TIME 期間に値を挿入できません。
- SQL ステートメント内のストリングは、アプリケーション・コード・ページのものとなされます。
- 先頭がドル記号 (\$) の列名は、フィールド名と区別するために、SQL 区切り ID (二重引用符で囲む) として指定する必要があります。
- ほとんどの場合、フィールド名をパラメーター・マーカーに変換し、フィールドの値をバインドした後、INGEST ユーティリティーは、これらのステートメントをそのまま DB2 に渡します。そのため、これらには同じ制約事項が適用されます。以下に例を示します。
  - フィールドは、そのフィールド・タイプとしてバインドされます。このため、指定されたタイプの値が許可されている場合にのみ使用できます。例えば、タイプ DB2SECURITYLABEL のフィールドを使用できるのは、セキュリティ・ラベルが許可されている場合のみです。
  - INSERT ステートメントの列リストを省略し、VALUES リストは指定する場合、その VALUES リストには、すべての列に対応する項目を含める必要があります。表に暗黙的な非表示列が含まれる場合、暗黙的な非表示列を含めるかどうかを明示的に指定する必要があります。DB2\_DMU\_DEFAULT レジストリー変数、または IMPLICITLYHIDDENINCLUDE キーワードか IMPLICITLYHIDDENMISSING キーワードを使用して、暗黙的な非表示列を含めるかどうかを指定します。
- 次の条件のいずれかが当てはまる場合、INGEST ユーティリティーはエラーを出し、その INGEST コマンドを終了させます。
  - コマンドに指定されている SQL ステートメントがフィールドを参照していない。
  - SQL ステートメントが DELETE または UPDATE であり、WHERE 節がないか、または WHERE 節がフィールドを参照していない。
  - SQL ステートメントが MERGE であり、ON 節がフィールドを参照していない。
- デフォルトの分離レベルはカーソル固定 (CS) です。これは、SQL ステートメント (INSERT 以外) に WITH 節を指定したり、環境変数 DB2CLIINIPATH で指定されているファイル (デフォルトは sqllib/cfg/db2cli.ini) 内に、CLI の IsolationLevel キーワードを設定したりしてオーバーライドすることができます。ただし、IsolationLevel キーワードを設定すると、すべての CLI アプリケーションに影響します。

## MERGE ステートメント

**INGEST** コマンドは、**MERGE** ステートメントを入力データ・レコードごとに発行し、各レコードを 1 行表 (SQL **MERGE** ステートメントの **USING table-reference** パラメーターと同等) として処理します。これは、時に同等の SQL **MERGE** ステートメントとは異なる結果を生成することがあります。以下の例について考慮します。

- 表 **SOURCE\_TABLE** には、次の 2 行が入っています。

```
(1, 'source data 1')
(2, 'source data 2')
```

- 入力ファイル **source\_table.del** には、**SOURCE\_TABLE** と同じ 2 つのレコードが入っています。

- 表 **TARGET\_TABLE** には、次の 1 行が入っています。

```
(NULL, 'target data 1')
```

SQL **MERGE** ステートメントは以下のとおりです。

```
MERGE INTO target_table
  USING (SELECT * FROM source_table) source_table
  ON target_table.c1 <> source_table.c1
  WHEN NOT MATCHED THEN
    INSERT VALUES(source_table.c1, source_table.c2);
```

**INGEST** コマンド (同等の **MERGE** ステートメントを使用するもの) は以下のとおりです。

```
INGEST FROM FILE source_table.del
  FORMAT DELIMITED
  (
    $c1 INTEGER EXTERNAL,
    $c2 CHAR(32)
  )
  MERGE INTO target_table
  ON c1 <> $c1
  WHEN NOT MATCHED THEN
    INSERT VALUES($c1, $c2);
```

この SQL **MERGE** ステートメントの実行後は、**TARGET\_TABLE** には以下の行が入ります。

```
(NULL, 'target data 1')
(1, 'source data 1')
(2, 'source data 2')
```

この **INGEST** コマンドの実行後は、**TARGET\_TABLE** には以下の行が入ります。

```
(NULL, 'target data 1')
(1, 'source data 1')
```

**INGEST** コマンドの場合に行数が少なくなる理由は、**INGEST** コマンドが最初の入力データ・レコードを処理した後、ターゲット表には上記の 2 行が入るからです。この時点以降、**ON** 節は一致する行を持つため、**WHEN NOT MATCHED** 節の **INSERT** ステートメントはもう処理されないこととなります。



この例は、MERGE ステートメントの一般的な使用方法ではありません。MERGE ステートメントのより一般的な形式として、突き合わせキーに基づいて行を更新または挿入する場合、SQL MERGE ステートメントと INGEST コマンドは、同じ結果を生成します。

### REPLACE ステートメント

REPLACE は、INGEST ユーティリティが挿入を開始する前に DELETE ステートメントを実行して表の全行を削除する点を除き、基本的には INSERT と同じです。表が大きい場合、DELETE に時間がかかり、かなりのログ・スペースが使用される可能性があります。

### データ・タイプ変換

ユーティリティが INGEST するフィールドごとに、以下の 2 つの場所でデータ変換が発生する可能性があります。

- 入力データが文字フォーマットである場合 (ASCII で指定された数値を含む)、ユーティリティが、データを文字からフィールド・タイプに変換します。
- フィールドのタイプが、対応する列のタイプと異なっている場合、DB2 が、フィールド値をフィールド・タイプから列のタイプに変換します。

どちらの場合も、INGEST ユーティリティおよび DB2 は、『代入と比較』のトピックで説明している規則を使用します。これは、時に、文字から列タイプに直接変換する場合とは異なる結果を生成することがあります。以下の例について考慮します。

- 入力ファイルに、有効な DECFLOAT(34) の数値ではあるが DECFLOAT(16) の範囲外である、ストリング '9.99E6143' が指定されています。
- フィールド・タイプは DECFLOAT(34) です。
- 列タイプは DECFLOAT(16) です。

この例では、INGEST ユーティリティが、フィールド値を文字から DECFLOAT(34) に変換します。次に、DB2 が、DECFLOAT(34) から DECFLOAT(16) に変換します。この結果、警告が生成され、DECFLOAT(16) 列の値が DECFLOAT の無限大になります。

### フィールドの SQL データ・タイプへのキャスト

INGEST コマンドに指定された SQL ステートメントを実行する前に、INGEST ユーティリティは、フィールド名をパラメーター・マーカーに変換し、その後、SQL ステートメントを動的に実行します。DB2 は、パラメーター・マーカーのデータ・タイプを判別するために、『型なし式のデータ・タイプの判別』のトピックの説明にある規則を使用します。動的 SQL と同様、フィールドのデータ・タイプを、使用されているコンテキストから判別できない場合は、エラーを受け取る可能性があります。その場合は、以下のアクションのいずれかを実行する必要があります。

- **DB2\_DEFERRED\_PREPARE\_SEMANTICS** レジストリー変数を YES にします。
- CAST 指定を使用して、フィールド名を SQL データ・タイプにキャストします。

例えば、以下の INGEST コマンドについて考えてみます。

## INGEST

```
INGEST FROM FILE my_file.txt
  FORMAT DELIMITED ( $date_fld DATE )
  INSERT INTO my_table(int_col) VALUES(day($date_fld));
```

**DB2\_DEFERRED\_PREPARE\_SEMANTICS** レジストリー変数を NO (デフォルト) に設定している場合は、INGEST ユーティリティーがこのステートメントを実行しようとする、エラーが返される可能性があります。これは、DB2 が、どのバージョンの DAY 関数を使用すべきか判別できないためです。(このフィールドを DATE として定義し、INGEST ユーティリティーが DATE としてバインドした場合であっても同じです)。

この問題は、以下のように、フィールド名を対応するフィールド・タイプにキャストすることで解決できます。

```
INSERT INTO my_table(int_col) VALUES( day(CAST($date_fld AS DATE)) );
```

場合によっては、レジストリー変数

**DB2\_DEFERRED\_PREPARE\_SEMANTICS=YES** を設定することにより、問題を解決することもできます。ただし、これは常に有効とは限りません。

追加の例として、以下の **INGEST** コマンドについて考えてみましょう。

```
INGEST FROM FILE my_file.txt
  FORMAT DELIMITED ( $field1 DECIMAL(5,2) )
  INSERT INTO my_table(dec_col) VALUES($field1 + 1);
```

\$field1 は整数に加算されているため、DB2 は、このフィールドにタイプ INTEGER を割り当てます。DB2 が DECIMAL(5,2) タイプを割り当てるようにするには、SQL ステートメントを以下のように変更する必要があります。

```
INSERT INTO my_table(dec_col) VALUES(CAST($field1 AS DECIMAL(5,2)) + 1);
```

または

```
INSERT INTO my_table(dec_col) VALUES($field1 + 001.00);
```



## INITIALIZE TAPE

ストリーミング磁気テープ装置へのバックアップおよびリストア操作のためにテープを初期化します。このコマンドは Windows オペレーティング・システムでのみサポートされています。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMANT

### 必要な接続

なし

### コマンド構文

```
▶▶ INITIALIZE TAPE [ON=device] [USING=blksize]
```

### コマンド・パラメーター

#### ON *device*

有効なテープ装置名を指定します。デフォルト値は ¥¥.¥TAPE0 です。

#### USING *blksize*

装置のブロック・サイズを指定します (バイト単位)。値が装置のブロック・サイズとしてサポートされている範囲内であれば、装置は指定されたそのブロック・サイズを使用するよう初期化されます。

**BACKUP DATABASE** コマンドおよび **RESTORE DATABASE** コマンドで指定されるバッファー・サイズは、ここで指定されるブロック・サイズで割り切れなければなりません。

このパラメーターに値を指定しなかった場合、装置はデフォルトのブロック・サイズを使用するよう初期化されます。値ゼロを指定した場合は、装置は可変長のブロック・サイズを使用するよう初期化されます。装置が可変長のブロック・モードをサポートしていない場合は、エラーが戻されます。

テープへのバックアップの場合、現在、可変ブロック・サイズの使用はサポートされていません。そのオプションを使用する必要がある場合は、リカバリーが正常に実行されるように十分にテストしたプロシージャが使用できるようになっていることを確認し、また可変ブロック・サイズを指定して作成されたバックアップ・イメージを使用してください。

可変ブロック・サイズを使用する場合、使用している磁気テープ装置の最大限度以下のバックアップ・バッファー・サイズを指定する必要があります。パフォーマンスを最適化するには、使用している装置のブロック・サイズの最大限度と等しい値をバッファー・サイズとして使用しなければなりません。

## INSPECT

データベースのページの整合性がとれているかどうかを調べることにより、データベースの構造上の整合性を検査します。 **INSPECT** コマンドでは、表オブジェクトの構造および表スペースの構造が有効かどうか調べられます。オブジェクト間妥当性検査により、オンラインで索引とデータの整合性検査を実行できます。

### 有効範囲

単一パーティション・データベース環境では、有効範囲はその単一のパーティションのみです。パーティション・データベース環境では、`db2nodes.cfg` に定義されている論理パーティションすべてのコレクションです。パーティション表の場合、**CHECK DATABASE** および **CHECK TABLESPACE** オプションの対象には、個々のデータ・パーティションと非パーティション索引が含まれます。**CHECK TABLE** オプションもまたパーティション表に使用できますが、その場合、単一のデータ・パーティションや索引ではなく、表内のすべてのデータ・パーティションと索引が検査されます。

### 許可

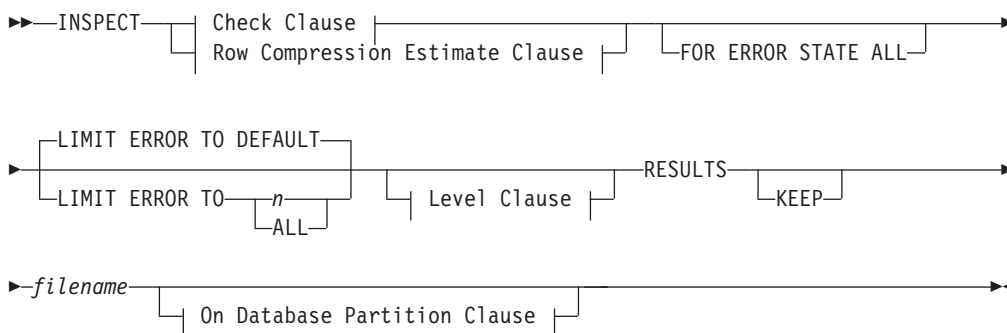
**INSPECT CHECK** の場合、以下の権限のどれかになります。

- SYSADM
- DBADM
- SYSCTRL
- SYSMANT
- 単一表の場合 CONTROL 特権。

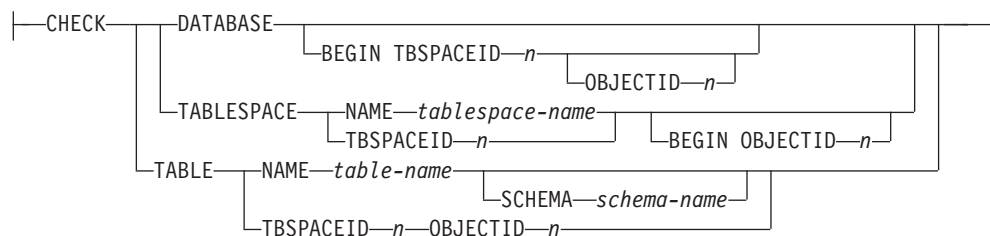
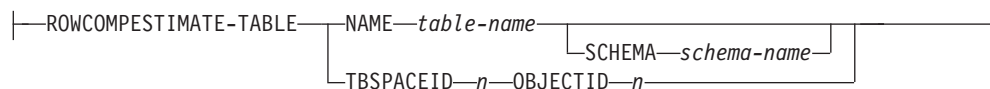
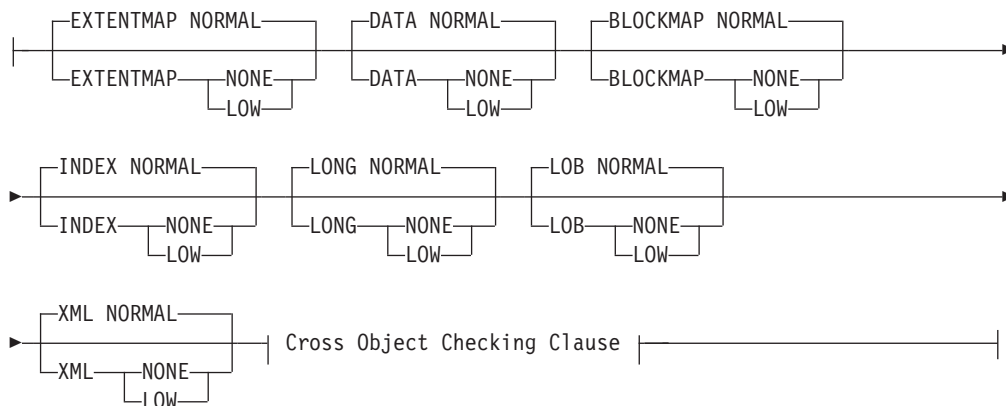
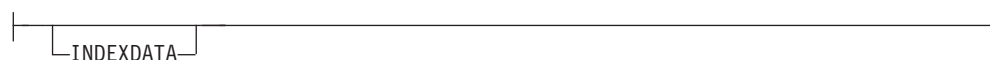
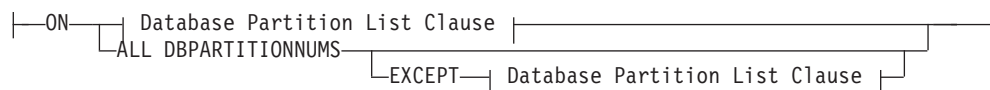
### 必要な接続

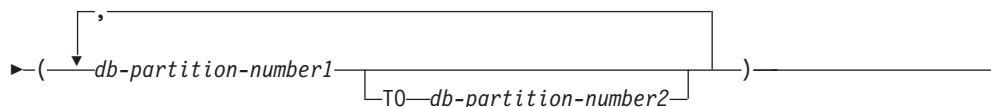
データベース

### コマンド構文



### Check Clause:

**Row Compression Estimate Clause:****Level Clause:****Cross Object Checking Clause:****On Database Partition Clause:****Database Partition List Clause:**



## コマンド・パラメーター

### CHECK

チェック処理を指定します。

### DATABASE

全データベースを指定します。

### BEGIN TBSpaceID *n*

指定された表スペース ID 番号を持つ表スペースから開始する処理を指定します。

### OBJECTID *n*

指定された表スペース ID 番号およびオブジェクト ID 番号を持つ表から開始する処理を指定します。

### TABLESPACE

#### NAME *tablespace-name*

指定された表スペース名を持つ単一の表スペースを指定します。

#### TBSpaceID *n*

指定した表スペース ID 番号を持つ単一表スペースを指定します。

#### BEGIN OBJECTID *n*

指定されたオブジェクト ID 番号の表から開始する処理を指定します。

### TABLE

#### NAME *table-name*

指定された表名を持つ表を指定します。

#### SCHEMA *schema-name*

単一の表操作に対して指定された表名のスキーマ名を指定します。

#### TBSpaceID *n* OBJECTID *n*

指定された表スペース ID 番号およびオブジェクト ID 番号を持つ表を指定します。

### ROWCOMPESTIMATE

表での行の圧縮の効果を見積もります。また、この操作の対象データベース・パーティションを指定することもできます。

この操作は、**KEEP** オプションが指定されているかどうかにかかわらず、**RESULTS** 出力ファイルを保持します。

このツールは表データのサンプル (標本) を取り、それに基づいてディクショナリーを作成することができます。その後、このディクショナリーを使用し、サンプルに含まれる記録に基づいて圧縮をテストすることができます。このテスト圧縮から得られるデータを使って、以下のような見積もりが可能になります。

- 圧縮により削減されるバイトのパーセンテージ

- 圧縮により削減されるページのパーセンテージ
- コンプレッション・ディクショナリーのサイズ
- エクスパンション・ディクショナリーのサイズ

この表に関して **COMPRESS YES** 属性が設定されている場合、この表のディクショナリーがまだ存在しなければ、**INSPECT** はこれらの圧縮見積もりの収集用に作成されたディクショナリーを挿入します。**INSPECT** は、表にアクセスする他のアプリケーションと並行して、ディクショナリーを挿入しようとしています。ディクショナリーを挿入するには、「表の排他的変更」ロックおよび「表の意図的排他」ロックが必要です。**INSPECT** は、行の圧縮をサポートする表にのみディクショナリーを挿入します。パーティション表の場合、各パーティションごとに個別のディクショナリーが作成され、挿入されます。

表の行データのサンプリングおよび表のコンプレッション・ディクショナリーのビルドの際、**INSPECT** コマンドがサポートするのは表オブジェクト内の表の行データのみです。表に **XML** 列が含まれる場合、その表の **XML** ストレージ・オブジェクト内にある **XML** データに関してはデータはサンプリングされず、コンプレッション・ディクショナリーはビルドされません。代わりに、表関数を使用します。

**ROWCOMPESTIMATE** オプションは、索引圧縮見積もりは提供しません。代わりに、表関数を使用します。

## RESULTS

結果出力ファイルを指定します。ファイルは診断データ・ディレクトリー・パスに書き込まれます。チェック処理によってエラーが検出されない場合、この結果出力ファイルは **INSPECT** 操作の終了時に消去されます。チェック処理によってエラーが検出される場合、この結果出力ファイルは **INSPECT** 操作の終了時に消去されません。

**KEEP** 結果出力ファイルを常に維持することを指定します。

*file-name*

結果出力ファイルの名前を指定します。このファイルは診断データ・ディレクトリー・パスに作成されなければなりません。

## ALL DBPARTITIONNUMS

**db2nodes.cfg** ファイルに指定されているすべてのデータベース・パーティションで操作が実行されることを指定します。データベース・パーティション節が指定されていない場合、これがデフォルトです。

## EXCEPT

データベース・パーティション・リストに指定されているものを除き、**db2nodes.cfg** ファイルに指定されているすべてのデータベース・パーティションで操作が実行されることを指定します。

## ON DBPARTITIONNUM | ON DBPARTITIONNUMS

データベース・パーティションのセットに対して操作を実行します。

*db-partition-number1*

データベース・パーティション・リスト内のデータベース・パーティション番号を指定します。

*db-partition-number2*

2 番目のデータベース・パーティション番号を指定し、  
*db-partition-number1* から *db-partition-number2* までのすべてのデータベース・パーティションがデータベース・パーティション・リストに含まれるようにします。

**FOR ERROR STATE ALL**

その内部状態が既にエラー状態を示している表オブジェクトに関してチェックを実行する場合、その状況だけを報告し、オブジェクトのスキャンは行いません。このオプションを指定すると、内部状態が既にエラー状態を示している場合でも、オブジェクトのスキャンを行います。

**INDEXDATA** オプションと共にこれを使用すると、索引またはデータ・オブジェクトがエラー状態にある限り、オンラインでの索引とデータの整合性検査は実行されません。

**LIMIT ERROR TO *n***

レポート作成の限界値となる、オブジェクトに関するエラー状態のページ数。オブジェクトのエラー・ページ数がこの限界に達すると、残りのオブジェクトのチェック処理は中止されます。

**INDEXDATA** オプションと共にこれを使用する場合、*n* は、オンラインでの索引とデータの整合性検査におけるレポート作成の限界値となるエラー数を示します。

**LIMIT ERROR TO DEFAULT**

1 つのオブジェクトについて、報告されるエラーを制限するデフォルトのページ数。この値は、オブジェクトのエクステント・サイズです。このパラメーターがデフォルトです。

**INDEXDATA** オプションと共に使用される場合、**DEFAULT** は、オンラインでの索引とデータの整合性検査におけるレポート作成の限界値となるデフォルトのエラー数を示します。

**LIMIT ERROR TO ALL**

報告されるエラー・ページ数の限界はありません。

**INDEXDATA** オプションと共に使用される場合、**ALL** は、オンラインでの索引とデータの整合性検査においてレポートされるエラー数に制限がないことを示します。

**EXTENTMAP****NORMAL**

エクステント・マップの処理レベルを通常に指定します。デフォルト。

**NONE** エクステント・マップの処理レベルをなしに指定します。

**LOW** エクステント・マップの処理レベルを低に指定します。

**DATA****NORMAL**

データ・オブジェクトの処理レベルを通常に指定します。デフォルト。

**NONE** データ・オブジェクトの処理レベルをなしに指定します。

**LOW** データ・オブジェクトの処理レベルを低に指定します。

## BLOCKMAP

### NORMAL

ブロック・マップ・オブジェクトの処理レベルを通常に指定します。デフォルト。

**NONE** ブロック・マップ・オブジェクトの処理レベルをなしに指定します。

**LOW** ブロック・マップ・オブジェクトの処理レベルを低に指定します。

## INDEX

### NORMAL

索引オブジェクトの処理レベルを通常に指定します。デフォルト。

**NONE** 索引オブジェクトの処理レベルをなしに指定します。

**LOW** 索引オブジェクトの処理レベルを低に指定します。

## LONG

### NORMAL

LONG オブジェクトの処理レベルを通常に指定します。デフォルト。

**NONE** LONG オブジェクトの処理レベルをなしに指定します。

**LOW** LONG オブジェクトの処理レベルを低に指定します。

## LOB

### NORMAL

LOB オブジェクトの処理レベルを通常に指定します。デフォルト。

**NONE** LOB オブジェクトの処理レベルをなしに指定します。

**LOW** LOB オブジェクトの処理レベルを低に指定します。

## XML

### NORMAL

XML 列オブジェクトの処理レベルを通常に指定します。デフォルト。XML オブジェクトのページが検査され、ほとんどの不整合が検出されます。実際の XML データは検査されません。

**NONE** XML 列オブジェクトの処理レベルを「なし」に指定します。XML オブジェクトはまったく検査されなくなります。

**LOW** XML 列オブジェクトの処理レベルを「低」に指定します。XML オブジェクトのページが検査され、いくつかの不整合が検出されます。実際の XML データは検査されません。

## INDEXDATA

索引とデータの整合性検査を実行するために指定します。デフォルトでは **INDEXDATA** 検査は実行されません。



**例**

- その時点で検査されているオブジェクトを含むすべてのオブジェクトに対して読み取り/書き込みアクセスできる索引とデータの整合性検査を実行するには、以下のコマンドを発行します。

```
inspect check table name fea3 indexdata results keep fea3high.out
```

- 現在検査されているオブジェクトを含む、すべてのオブジェクトに対して読み取り/書き込みアクセスできる索引とデータの整合性検査を実行するには、以下のコマンドを発行します。

```
INSPECT CHECK TABLE NAME car SCHEMA vps INDEXDATA RESULTS KEEP table1.out
```

- **EMPLOYEE** という名前の表にあるデータが圧縮される場合に、保存されるストレージ・スペース量を見積もるには、以下のコマンドを発行します。

```
INSPECT ROWCOMPESTIMATE TABLE NAME car SCHEMA vps RESULTS table2.out
```

**使用上の注意**

1. 表オブジェクトでの **CHECK** 操作に関して、処理レベルはオブジェクトに対して指定できます。デフォルトは **NORMAL** レベルです。オブジェクトに **NONE** を指定すると、そのオブジェクトは除外されます。**LOW** を指定すると、**NORMAL** で行われるチェックのサブセットのチェック操作を行います。
2. 表スペースまたは表を識別する **ID** 値を指定することにより、特定の表スペースまたは表から開始するように **CHECK DATABASE** オプションを指定できます。
3. 表を識別する **ID** 値を指定することにより、特定の表から開始するように **CHECK TABLESPACE** オプションを指定できます。
4. 表スペースの処理は、表スペース内にあるオブジェクトにのみ影響を与えます。ただし **INDEXDATA** オプションを使用する場合は例外です。**INDEXDATA** は、索引オブジェクトが表スペース内に存在する限り、索引とデータの整合性を検査します。つまり、
  - 検査対象として指定された、索引オブジェクトがある表スペースとは別の表スペースにデータ・オブジェクトが存在する場合でも、**INDEXDATA** 検査は依然として有効です。
  - パーティション表の場合、それぞれの索引は別個の表スペースに存在する可能性があります。索引/データ検査を実行するメリットがあるのは、指定された表スペース内に存在する索引だけです。1つの表に対するすべての索引を検査するには、**CHECK TABLE** オプションまたは **CHECK DATABASE** オプションを使用してください。
5. オンライン検査処理では、分離レベルを非コミット読み取りに指定してデータベース・オブジェクトにアクセスします。**COMMIT** 処理は、**INSPECT** 処理の際に行われます。**INSPECT** を呼び出す前に、**COMMIT** または **ROLLBACK** を発行して作業単位を終了することをお勧めします。
6. オンライン検査処理により、フォーマットされていない検査データが、指定された結果ファイルに書き出されます。ファイルは診断データ・ディレクトリー・パスに書き込まれます。チェック処理によってエラーが検出されない場合、この結果出力ファイルは **INSPECT** 操作の終了時に消去されます。チェック処理によってエラーが検出される場合、この結果出力ファイルは **INSPECT** 操作の終了時に消去されません。チェック処理が完了した後、検査の詳細を表示す



るには、ユーティリティ `db2inspf` を使って検査結果データをフォーマットする必要があります。結果ファイルには、データベース・パーティション番号のファイル拡張子が付きます。

7. パーティション・データベース環境では、各データベース・パーティションごとに、そのデータベース・パーティション番号と一致した拡張子を持つ独自の結果出力ファイルが生成されます。結果出力ファイルは、データベース・マネージャの診断データ・ディレクトリー・パスに出力されます。既に存在するファイル名を指定すると、操作は処理されません。ですから、ファイル名を指定する前にそのファイルを除去する必要があります。
8. 通常のオンライン検査処理では、分離レベルを非コミット読み取りに指定してデータベース・オブジェクトにアクセスします。コンプレッション・ディクショナリーを表に挿入するとき、書き込みロックの獲得が試みられます。ディクショナリー挿入ロックの詳細については、**ROWCOMPESTIMATE** オプションを参照してください。コミット処理は、検査処理時に行われます。検査操作を開始する前に、**COMMIT** または **ROLLBACK** を発行して作業単位を終了することをお勧めします。
9. **INDEXDATA** オプションは、索引とデータの間の論理的な不整合だけを調べます。したがって、**INDEXDATA** 検査を実行する前に、物理的な破損が生じないように、まず **INDEX** および **DATA** 検査を別々に実行するようお勧めします。
10. **INSPECT** コマンドに **INDEXDATA** パラメーターを指定すると、索引とデータの整合性検査を実行し、その時点で検査されているものを含むすべてのオブジェクト/表に対して読み取り/書き込みアクセスできます。**INSPECT INDEXDATA** オプションには以下の検査が含まれます。
  - 指定された索引項目に関するデータ行が存在するかどうか。
  - データ値検査に対するキー。

**INDEXDATA** オプションが指定される場合、

- デフォルトでは、明示的に指定されたレベル節オプションの値だけが使用されます。レベル節オプションが明示的に指定されていない場合、デフォルト・レベルは **NORMAL** から **NONE** に上書きされます。例えば、指定されているレベル節オプションが **INDEXDATA** だけである場合、デフォルトでは、索引とデータのチェックのみ実行されます。
11. **BLOCKMAP** オプションは、マルチディメンション・クラスタリング (MDC) 表または挿入時クラスタリング表 (ITC) の空きブロックを再利用するための再編成の後に、表スペースでブロックが再利用されたかどうかなどの情報を返します。

## LIST ACTIVE DATABASES

**GET SNAPSHOT FOR ALL DATABASES** コマンドによってリスト表示される情報のサブセットを表示します。アクティブなデータベースは、すべてのアプリケーションが接続して使用できます。

アクティブなデータベースごとに、このコマンドは以下の情報を表示します。

- データベース名
- データベースに現在接続しているアプリケーションの数
- データベース・パス

### 有効範囲

このコマンドは、現在アタッチされているメンバーに対して呼び出され、デフォルトではそのメンバーについての情報のみを返します。現在アタッチされているメンバーは、デフォルトでは、コマンドが実行されるホストになります。ホストに対して複数のメンバーがある場合、現在アタッチされているメンバーは、そのホスト上の `db2nodes.cfg` ファイルにリストされている最初のメンバーです。

現在アタッチされているメンバーではないメンバーに対してコマンドを実行する場合、特定のメンバーに対してこのコマンドを実行するには **AT MEMBER** オプションを指定し、すべてのメンバーに対してこのコマンドを実行して集約された結果を受け取るには **GLOBAL** オプションを指定してください。

後続のコマンドを実行する対象の、現在アタッチされているメンバーを変更するには、**SET CLIENT** コマンドの **ATTACH\_MEMBER** パラメーターを使用します。このクライアント設定を有効にするには、アプリケーションを **DETACH** してから再び **ATTACH** する必要があります。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMOINT
- SYSMON

### コマンド構文

```

▶▶—LIST ACTIVE DATABASES—┐
└─AT MEMBER—member-number─┘
└─GLOBAL───────────────────┘

```

### コマンド・パラメーター

**AT MEMBER** *member-number*

アクティブ・データベースのリストを表示するメンバーを指定します。

**GLOBAL**

パーティション・データベース環境または DB2 pureScale 環境におけるすべてのメンバーのアクティブ・データベースのリストを戻します。

**例**

以下に示すのは、**LIST ACTIVE DATABASES** コマンドの出力例です。

## Active Databases

```

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/

```

**互換性**

以前のバージョンとの互換性:

- **DB2\_ENFORCE\_MEMBER\_SYNTAX** レジストリー変数が ON に設定されている場合を除き、**DBPARTITIONNUM** または **NODE** を **MEMBER** の代わりに使用できます。

## LIST APPLICATIONS

アクティブなすべてのデータベース・アプリケーションに関して、アプリケーション・プログラム名、許可 ID (ユーザー名)、アプリケーション・ハンドル、アプリケーション ID、およびデータベース名を標準出力に出力します。このコマンドでは、オプションとしてアプリケーションのシーケンス番号、状況、状況変更時刻、およびデータベース・パスを表示することもできます。

### 有効範囲

このコマンドは、現在アタッチされているメンバーに対して呼び出され、デフォルトではそのメンバーについての情報のみを返します。現在アタッチされているメンバーは、デフォルトでは、コマンドが実行されるホストになります。ホストに対して複数のメンバーがある場合、現在アタッチされているメンバーは、そのホスト上の `db2nodes.cfg` ファイルにリストされている最初のメンバーです。

現在アタッチされているメンバーではないメンバーに対してコマンドを実行する場合、特定のメンバーに対してこのコマンドを実行するには **AT MEMBER** オプションを指定し、すべてのメンバーに対してこのコマンドを実行して集約された結果を受け取るには **GLOBAL** オプションを指定してください。

後続のコマンドを実行する対象の、現在アタッチされているメンバーを変更するには、**SET CLIENT** コマンドの **ATTACH\_MEMBER** パラメーターを使用します。このクライアント設定を有効にするには、アプリケーションを **DETACH** してから再び **ATTACH** する必要があります。

### 許可

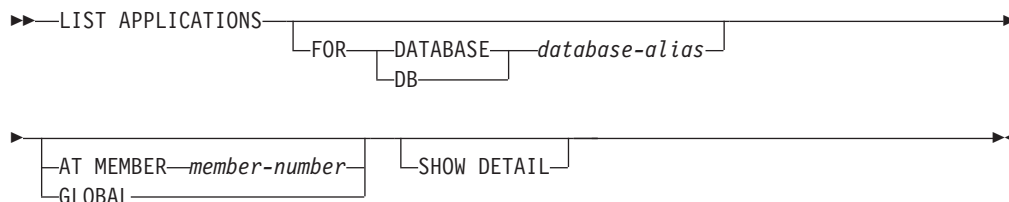
以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMOINT
- SYSMON

### 必要な接続

インスタンス。リモート・インスタンスのアプリケーションをリスト表示するには、まず最初にそのインスタンスにアタッチする必要があります。

### コマンド構文



## コマンド・パラメーター

### FOR DATABASE *database-alias*

指定したデータベースに接続された各アプリケーションの情報が表示されます。データベース名情報は表示されません。このオプションを指定しない場合、このコマンドは、ユーザーが現在アタッチしているメンバーにあるデータベースと接続している各アプリケーションごとに、情報を表示します。

デフォルトのアプリケーション情報には、以下のもので構成されます。

- 許可 ID
- アプリケーション名
- アプリケーション・ハンドル
- アプリケーション ID
- データベース名
- エージェントの数

### AT MEMBER *member-number*

アクティブ・アプリケーションを表示する対象となるメンバーを指定します。

**GLOBAL** パーティション・データベース環境または DB2 pureScale 環境におけるすべてのメンバーのアクティブ・アプリケーションのリストを戻します。

### SHOW DETAIL

追加の出力情報の一部として、以下が含まれます。

- CONNECT 許可 ID
- シーケンス番号
- 調整メンバー番号
- コーディネーター PID またはスレッド
- 状況
- 状況変更時刻
- ノード
- データベース・パス

このオプションを指定した場合には、出力をファイルにリダイレクトしておいて、エディターを使ってそのレポートを表示するのがよいでしょう。画面上に表示する場合には、出力行が折り返す場合もあります。

## 例

**SAMPLE** データベースに接続されているアプリケーションに関する詳細情報をリスト表示するには、以下のコマンドを発行します。

```
list applications for database sample show detail
```

## 使用上の注意

データベース管理者は、このコマンドの出力を問題判別の参考にすることができます。さらにこの情報は、データベース管理者がアプリケーションの中で、**GET SNAPSHOT** コマンドまたは **FORCE APPLICATION** コマンドを使用する場合に必要になります。

## LIST APPLICATIONS

リモート・インスタンス (または別のローカル・インスタンス) のアプリケーションをリスト表示するには、まず最初にそのインスタンスにアタッチする必要があります。アタッチが既に存在しているのに **FOR DATABASE** を指定し、かつ現行のアタッチとは異なるインスタンスにデータベースが存在している場合、このコマンドは失敗します。

**LIST APPLICATIONS** は、ユーザー・アプリケーションのみを表示するのに対し、**LIST APPLICATIONS SHOW DETAIL** は、システム・アプリケーションを含む全アプリケーションを表示します。イベント・モニターは、システム・アプリケーションの一例です。スナップショット出力において、通常、システム・アプリケーションは、「db2」で始まるアプリケーション名 (例えば db2stmm、db2taskd など) で表示されます。

### 互換性

以前のバージョンとの互換性:

- **DB2\_ENFORCE\_MEMBER\_SYNTAX** レジストリー変数が **ON** に設定されている場合を除き、**DBPARTITIONNUM** または **NODE** を **MEMBER** の代わりに使用できます。

## LIST COMMAND OPTIONS

環境変数 **DB2BQTIME**、**DB2DQTRY**、**DB2RQTIME**、**DB2IQTIME**、**DB2OPTIONS** の現行設定値をリストします。

### 許可

なし

### 必要な接続

なし

### コマンド構文

▶—LIST COMMAND OPTIONS—▶

### コマンド・パラメーター

なし

### 例

以下に示すのは、**LIST COMMAND OPTIONS** の出力例です。

Command Line Processor Option Settings

```
Backend process wait time (seconds)      (DB2BQTIME) = 1
No. of retries to connect to backend     (DB2BQTRY)  = 60
Request queue wait time (seconds)        (DB2RQTIME) = 5
Input queue wait time (seconds)          (DB2IQTIME) = 5
Command options                          (DB2OPTIONS) =
```

Option	Description	Current Setting
-a	Display SQLCA	OFF
-b	Auto-Bind	ON
-c	Auto-Commit	ON
-d	XML declarations	OFF
-e	Display SQLCODE/SQLSTATE	OFF
-f	Read from input file	OFF
-l	Log commands in history file	OFF
-n	Remove new line character	OFF
-o	Display output	ON
-p	Display interactive input prompt	ON
-r	Save output to report file	OFF
-s	Stop execution on command error	OFF
-t	Set statement termination character	OFF
-v	Echo current command	OFF
-w	Display FETCH/SELECT warning messages	ON
-z	Save all output to output file	OFF





```

Database 2 entry:
Database alias           = TC004000
Database name           = TC004000
Node name                = PRINODE
Database release level  = a.00
Comment                 =
Directory entry type    = LDAP
Catalog database partition number = -1
Gateway node name       = PRIGW
Alternate server node name =
Alternate server gateway node name = ALTGW

```

次に示すのは、ローカル・データベース・ディレクトリーの場合の出力例です。

```
Local Database Directory on /u/smith
```

```
Number of entries in the directory = 1
```

```

Database 1 entry:
Database alias           = SAMPLE
Database name           = SAMPLE
Database directory      = SQL00001
Database release level  = 8.00
Comment                 =
Directory entry type    = Home
Catalog database partition number = 0
Database partition number = 0

```

各フィールドは、次のとおりです。

#### Database alias

データベースの作成時またはカタログ時の *alias* パラメーターの値。データベースのカタログ時に別名が入力されなかった場合、データベース・マネージャーは、データベースのカタログ時の *database-name* パラメーターの値を使用します。

#### Database name

データベースのカタログ時の *database-name* パラメーターの値。通常、この名前はデータベース作成時点での名前です。

#### Local database directory

データベースが存在しているパス。このフィールドが表示されるのは、システム・データベース・ディレクトリーがスキャンされた場合だけです。

#### Database directory

データベースが置かれているディレクトリーの名前。このフィールドが表示されるのは、ローカル・データベース・ディレクトリーがスキャンされた場合だけです。

#### Node name

リモート・ノードの名前。この名前は、データベースおよびノードのカタログ時に *nodename* パラメーターに入力した値に対応します。

#### Database release level

データベースに対して実行可能なデータベース・マネージャーのリリース・レベル。

#### Comment

データベースをカタログした時点で入力された、データベースに関連する注釈。

### Directory entry type

データベースの存在ロケーション。

- Remote 項目には、別のノードにあるデータベースについて記述されません。
- Indirect 項目にはローカルのデータベースについて記述されます。システム・データベース・ディレクトリーと同じノードにあるデータベースは、ローカル・データベース・ディレクトリーに対するホーム (home) 項目を間接的に参照していると見なされるため、間接 (indirect) 項目と見なされます。
- Home 項目は、そのデータベース・ディレクトリーがローカル・データベース・ディレクトリーと同じパスにあることを示します。
- LDAP 項目は、データベース・ロケーション情報が LDAP サーバーに保管されることを示します。

システム・データベース・ディレクトリーにあるすべての項目は、リモート (remote) か間接 (indirect) です。システム・データベース・ディレクトリーにあるローカル・データベース・ディレクトリーの項目は、すべて間接 (indirect) 項目として表示されます。

### Authentication

クライアントでカタログされる認証タイプ。

### Principal name

完全修飾の Kerberos プリンシパル名を指定します。

### Catalog database partition number

どのノードをカタログ・データベース・パーティションにするかを指定します。これは、**CREATE DATABASE** コマンドを発行したデータベース・パーティションです。

### Database partition number

db2nodes.cfg の中で、このコマンドを発行したノードに対して割り当てられている番号を指定します。

### Alternate server hostname

データベースへの接続に通信障害がある場合に使用する代替サーバーのホスト名または IP アドレスを指定します。このフィールドは、システム・データベース・ディレクトリーに関してのみ表示されます。

### Alternate server port number

データベースへの接続に通信障害がある場合に使用する代替サーバーのポート番号を指定します。このフィールドは、システム・データベース・ディレクトリーに関してのみ表示されます。

### Alternate server node name

ディレクトリー項目タイプが LDAP の場合に、データベースへの接続に通信障害がある場合に使用する代替サーバーのノード名を指定します。

### Alternate server gateway node name

ディレクトリー項目タイプが LDAP の場合に、データベースへの接続に通信障害がある場合に使用する代替ゲートウェイのゲートウェイ・ノード名を指定します。

## 使用上の注意

**DB2LDAPCACHE** その他の変数設定にかかわらず、**LIST DATABASE DIRECTORY** コマンドまたは **LIST NODE DIRECTORY** コマンドを使用するときには、LDAP サーバーからローカル・データベースおよびノード項目のリストが読み取られます。

プロセスごとに最大 8 つのデータベース・ディレクトリー・スキャンをオープンすることができます。単一の DB2 セッション内で 9 つ以上の **LIST DATABASE DIRECTORY** コマンドを発行できないというこのバッチ・ファイルの制限を解決するためには、バッチ・ファイルをシェル・スクリプトに変換してください。"db2" 接頭部を使用すれば、コマンドごとに新しい DB2 セッションが生成されます。

## LIST DATABASE PARTITION GROUPS

現行データベースに関連付けられているすべてのデータベース・パーティション・グループのリストを表示します。

### 有効範囲

このコマンドは、\$HOME/sql1lib/db2nodes.cfg の中に指定されているどのデータベース・パーティションからでも発行できます。これらのどのデータベース・パーティションからでも同一の情報が戻されます。

### 許可

システム・カタログ SYSCAT.DBPARTITIONGROUPS および SYSCAT.DBPARTITIONGROUPDEF の場合には、以下のいずれかの権限が必要です。

- SYSADM
- SYSCTRL
- SYSMOINT
- SYSMON
- DBADM
- CONTROL 特権
- SELECT 特権

### 必要な接続

データベース

### コマンド構文

```
▶▶—LIST DATABASE PARTITION GROUPS—┐
└─SHOW DETAIL─┘
```

### コマンド・パラメーター

#### SHOW DETAIL

以下の情報を出力に含めることを指定します。

- 分散マップ ID
- データベース・パーティション番号
- 使用中フラグ

### 例

以下に示すのは、**LIST DATABASE PARTITION GROUPS** コマンドの出力例です。

```
DATABASE PARTITION GROUP NAME
-----
IBMCGROUP
IBMDEFAULTGROUP
```

2 record(s) selected.

## LIST DATABASE PARTITION GROUPS

以下に示すのは、**LIST DATABASE PARTITION GROUPS SHOW DETAIL** コマンドの出力例です。

DATABASE PARTITION GROUP NAME	PMAP_ID	DATABASE PARTITION NUMBER	IN_USE
IBMCATGROUP	0	0	Y
IBMDEFAULTGROUP	1	0	Y

2 record(s) selected.

各フィールドは、次のとおりです。

### DATABASE PARTITION GROUP NAME

データベース・パーティション・グループの名前。この名前は、データベース・パーティション・グループのデータベース・パーティションごとに繰り返されます。

### PMAP\_ID

分散マップの ID。この ID は、データベース・パーティション・グループのデータベース・パーティションごとに繰り返されます。

### DATABASE PARTITION NUMBER

データベース・パーティション番号。

### IN\_USE

次の 4 つの値のいずれか。

- Y** データベース・パーティション・グループで使用するデータベース・パーティションの名前。
- D** **REDISTRIBUTE DATABASE PARTITION GROUP** 操作の結果として、そのデータベース・パーティションはデータベース・パーティション・グループからドロップ中です。操作が完了すると、そのデータベース・パーティションは **LIST DATABASE PARTITION GROUPS** のレポートに含まれなくなります。
- A** そのデータベース・パーティションはデータベース・パーティション・グループには既に追加されていますが、分散マップにはまだ追加されていません。データベース・パーティション・グループ中の表スペース用のコンテナは、このデータベース・パーティションに追加されています。 **REDISTRIBUTE DATABASE PARTITION GROUP** 操作が正常に完了した場合、値は Y に変わります。
- T** そのデータベース・パーティションはデータベース・パーティション・グループには既に追加されていますが、分散マップにはまだ追加されていません。データベース・パーティション・グループ中の表スペース用のコンテナは、このデータベース・パーティションに追加されません。表スペース・コンテナはデータベース・パーティション・グループの表スペースごとに、新しいデータベース・パーティションに追加する必要があります。コンテナが正常に追加されると、値は A に変わります。

## LIST DBPARTITIONNUMS

現行データベースに関連付けられているすべてのデータベース・パーティションまたはメンバーのリストを表示します。

### 有効範囲

このコマンドは、\$HOME/sql1lib/db2nodes.cfg の中に指定されているどのデータベース・パーティションまたはメンバーからでも発行できます。これらのどのデータベース・パーティションまたはメンバーからでも同一の情報が戻されます。

### 許可

なし

### 必要な接続

データベース

### コマンド構文

▶▶—LIST DBPARTITIONNUMS—◀◀

### コマンド・パラメーター

なし

### 例

**例 1 - パーティション・データベース・インスタンス** : 以下に示すのは、**LIST DBPARTITIONNUMS** コマンドの出力例です。

```
DATABASE PARTITION NUMBER
-----
                                0
                                2
                                5
                                7
                                9
```

5 record(s) selected.

**例 2 - DB2 pureScale インスタンス** : DB2 pureScale 環境では、**LIST DBPARTITIONNUMS** は常に、単一パーティションを報告します。

```
DATABASE PARTITION NUMBER
-----
                                0
```

1 record(s) selected.

## LIST DCS APPLICATIONS

DB2 Connect Enterprise Edition を介してホスト・データベースに接続しているアプリケーションに関する情報を、標準出力に表示します。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMANT
- SYSMON

### 必要な接続

インスタンス。リモート・インスタンスの DCS アプリケーションをリスト表示するには、まず最初にそのインスタンスにアタッチする必要があります。

### コマンド構文

```

▶▶—LIST DCS APPLICATIONS—▶▶
                                ┌—SHOW DETAIL—┐
                                └—EXTENDED—┘
  
```

### コマンド・パラメーター

#### LIST DCS APPLICATIONS

デフォルトのアプリケーション情報の内容は、以下のとおりです。

- ホスト許可 ID (*username*)
- アプリケーション・プログラム名
- アプリケーション・ハンドル
- アウトバウンド・アプリケーション ID (*luwid*)

#### SHOW DETAIL

出力に以下の追加情報を含めることを指定します。

- クライアント・アプリケーション ID
- クライアント・シーケンス番号
- クライアント・データベース別名
- クライアント・ノード名 (*nname*)
- クライアント・リリース・レベル
- クライアント・コード・ページ
- アウトバウンド・シーケンス番号
- ホスト・データベース名
- ホスト・リリース・レベル

### EXTENDED

拡張されたレポートを生成します。このレポートには、**SHOW DETAIL** オプションを指定した場合に出力されるすべてのフィールドに加えて、以下の追加フィールドが含まれます。

- DCS アプリケーション状況
- 状況変更時刻
- クライアント・プラットフォーム
- クライアント・プロトコル
- クライアント・コード・ページ
- クライアント・アプリケーションのプロセス ID
- ホスト・コード化文字セット ID (CCSID)

注:

1. アプリケーション状況 (Application status) の値は、以下のうちのいずれかです。

#### **connect pending - outbound**

ホスト・データベースとの接続要求が発行され、その接続が確立されるのを DB2 Connect が待機している状態。

#### **waiting for request**

ホスト・データベースとの接続が既に確立され、クライアント・アプリケーションからの SQL ステートメントを DB2 Connect が待機している状態。

#### **waiting for reply**

SQL ステートメントがホスト・データベースに送られた状態。

2. 状況変更時刻 (Status change time) が表示されるのは、システム・モニターが処理中にその UOW スイッチがオンになっていた場合だけです。それ以外の場合は、Not Collected と表示されます。

### 使用上の注意

データベース管理者は、このコマンドを使用することによって、ゲートウェイへのクライアント・アプリケーション接続と、対応するゲートウェイからの ホスト接続を一致させることができます。

またデータベース管理者は、エージェント ID 情報を使うことによって、指定したアプリケーションを DB2 Connect サーバーから切り離すことができます。



## LIST DCS DIRECTORY

データベース接続サービス (DCS) ディレクトリーの内容をリスト表示します。

### 許可

なし

### 必要な接続

なし

### コマンド構文

▶—LIST DCS DIRECTORY—▶

### コマンド・パラメーター

なし

### 例

次に示すのは **LIST DCS DIRECTORY** の出力例です。

```
Database Connection Services (DCS) Directory
Number of entries in the directory = 1
DCS 1 entry:
Local database name           = DB2
Target database name         = DSN_DB_1
Application requestor name   =
DCS parameters               =
Comment                       = DB2/MVS Location name DSN_DB_1
DCS directory release level  = 0x0100
```

各フィールドは、次のとおりです。

#### Local database name

ターゲット・ホスト・データベースのローカル別名を指定します。これは、DCS ディレクトリーの中にホスト・データベースをカタログした時点で入力した *database-name* パラメーターに対応します。

#### Target database name

アクセスできるホスト・データベースの名前を指定します。これは、DCS ディレクトリーの中にホスト・データベースをカタログした時点で入力した *target-database-name* パラメーターに対応します。

#### Application requester name

アプリケーション・リクエスターまたはサーバー上に存在するプログラムの名前を指定します。

#### DCS parameters

アプリケーション・リクエスターで 사용되는接続およびオペレーティング環境パラメーターを含むストリング。ホスト・データベースをカタログした

## LIST DCS DIRECTORY

時点でのパラメーター・ストリングに対応します。ストリングは二重引用符で囲み、パラメーターはコンマで区切る必要があります。

### Comment

データベース項目の説明。

### DCS directory release level

データベースが作成された Distributed Database Connection Services<sup>®</sup> プログラムのバージョン番号を指定します。

### 使用上の注意

DCS ディレクトリーは、**CATALOG DCS DATABASE** コマンドを最初に呼び出した時点で作成されます。これは、DB2 のインストール先のパスまたはドライブにあり、DB2 Connect プログラムがインストールされている場合にワークステーションからアクセスできるホスト・データベースについての情報を提供します。ホスト・データベースとしては次のものが可能です。

- OS/390 および z/OS ホスト上の DB2 データベース
- System i ホスト上の DB2 データベース
- VSE &VM ホスト上の DB2 データベース

## LIST DRDA INDOUBT TRANSACTIONS

DRDA リクエスターと DRDA サーバーの間の未確定トランザクションのリストを表示します。DRDA コミット・プロトコルが使用されている場合は、DRDA 同期点管理プログラム相互間の未確定トランザクションが表示されます。

### 許可

なし

### 必要な接続

インスタンス

### コマンド構文

```

▶▶—LIST DRDA INDOUBT TRANSACTIONS—▶▶
    └──WITH PROMPTING──┘
  
```

### コマンド・パラメーター

#### WITH PROMPTING

未確定トランザクションを処理することを指定します。このパラメーターを指定すると、対話式ダイアログ・モードが開始され、未確定トランザクションのコミットまたはロールバックが可能になります。このパラメーターを指定しない場合、未確定トランザクションは標準出力装置に書き込まれ、対話式ダイアログ・モードは開始されません。

破棄 (forget) オプションはサポートされていません。未確定トランザクションをコミットまたはロールバックすると、そのトランザクションは自動的に破棄されます。

対話式ダイアログ・モードでは次のことが可能です。

- すべての未確定トランザクションのリスト表示 (1 を入力)
- 未確定トランザクション番号  $x$  のリスト表示 (1 の後に有効なトランザクション番号を入力)
- 終了 (q を入力)
- トランザクション番号  $x$  をコミット (c の後に有効なトランザクション番号を入力)
- トランザクション番号  $x$  をロールバック (r の後に有効なトランザクション番号を入力)

コマンド文字と引数の間は、ブランク・スペースで区切る必要があります。

トランザクションのコミットまたはロールバックを実行する前に、トランザクション・データが表示され、アクションを確認するように求められます。

### 使用上の注意

DRDA 未確定トランザクションが発生するのは、分散作業単位内のコーディネーターと参加者との間の通信が失われた場合です。分散作業単位では、ユーザーやアプ

## LIST DRDA INDOUBT TRANSACTIONS

リケーションが、単一の作業単位内で複数の場所にあるデータを読んだり更新したりできます。そのような作業には、2 フェーズ・コミットが必要となります。

第 1 のフェーズでは、すべての参加者に対してコミットの準備を要求します。第 2 のフェーズでは、トランザクションをコミットまたはロールバックします。第 1 フェーズ終了後にコーディネーターまたは参加者が使用できなくなると、分散トランザクションが未確定になります。

**LIST DRDA INDOUBT TRANSACTIONS** コマンドを実行するには、その前にアプリケーション・プロセスは、DB2 同期点管理プログラム (SPM) のインスタンスに接続する必要があります。CONNECT ステートメントの *dbalias* として、データベース・マネージャー構成パラメーター **spm\_name** を使います。

コミットの調整に SPM を使う TCP/IP 接続では、DRDA 2 フェーズ・コミット・プロトコルを使います。

## LIST HISTORY

データベース履歴レコードの中の項目のリストを表示します。データベース履歴レコードには、リカバリーと管理のさまざまなイベントの記録が含まれています。リカバリー・イベントには、データベース・レベルおよび表スペース・レベルのフル・バックアップ、増分バックアップ、リストア、およびロールフォワード操作が含まれます。さらにログ記録されるイベントには、表スペースの作成、変更、ドロップ、または名前変更、統計実行、表の再編成、表のドロップ、およびロードが含まれます。

**LIST HISTORY** コマンドは、それが発行されたデータベース・パーティションの履歴情報だけを返します。複数のパーティションに関して履歴をリストするには、**LIST HISTORY** コマンドをそれぞれ個々のデータベース・パーティションから発行するか、または **db2\_a11** 接頭部を使用して **LIST HISTORY** コマンドをすべてのデータベース・パーティションに対して実行します。

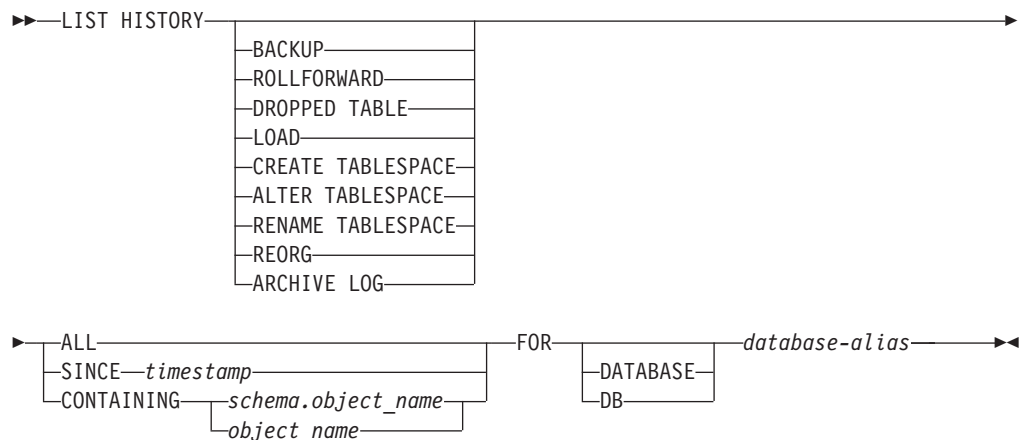
### 許可

なし

### 必要な接続

インスタンス。これに対してこのコマンドを実行するためには、いずれかのリモート・データベースにアタッチしていなければなりません。ローカル・データベースの場合、明示的なアタッチは必要ありません。

### コマンド構文



### コマンド・パラメーター

#### HISTORY

現在データベース履歴レコードの中に記録されているイベントのすべてのリストを表示します。

#### BACKUP

バックアップ操作およびリストア操作をリストします。

#### ROLLFORWARD

ロールフォワード操作をリストします。

**DROPPED TABLE**

ドロップした表レコードをリストします。ドロップした表レコードが作成されるのは、表がドロップされ、それを含む表スペースについて **DROPPED TABLE RECOVERY** オプションが有効になっている場合だけです。パーティション表の **CREATE TABLE** 構文を戻し、どの表スペースに入っている表データがドロップされたかを示します。

**LOAD** ロード操作をリストします。

**CREATE TABLESPACE**

表スペースの作成およびドロップ操作をリストします。

**RENAME TABLESPACE**

表スペースの名前変更操作をリストします。

**REORG**

再編成操作のリストを表示します。パーティション表の再編成された各データ・パーティションに関する情報が入っています。

**ALTER TABLESPACE**

表スペースの変更操作をリストします。

**ARCHIVE LOG**

アーカイブ・ログ操作と、それによってアーカイブされるログのリストを表示します。

**ALL** データベース履歴レコードのうち、指定したタイプのすべての項目のリストを表示します。

**SINCE** *timestamp*

完全なタイム・スタンプ (*yyyymmddhhmmss*の形式)、または先頭の接頭部 (最小で *yyyy*) を指定できます。指定したタイム・スタンプ以降のタイム・スタンプの項目のリストを表示します。

**CONTAINING** *schema.object\_name*

この修飾名は表を固有に識別します。

**CONTAINING** *object\_name*

この非修飾名は表スペースを固有に識別します。

**FOR DATABASE** *database-alias*

データベース・リカバリー履歴レコードをリストするデータベースを指定します。

**例**

以下の例は、**LIST HISTORY** コマンドのさまざまな使用法を示しています。

```
db2 list history since 19980201 for sample
db2 list history backup containing userspace1 for sample
db2 list history dropped table all for db sample
```

**例 1**

次の出力例は、2 つの項目を示しています。1 つはロード (L) 操作、もう 1 つはバックアップ (B) 操作です。

```
db2 list history all for SAMPLE
```

```
List History File for sample
```

Number of matching file entries = 2

```
Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log Backup ID
-----
L T 20100106133005001 R S S0000000.LOG S0000000.LOG
-----
"USERNAME"."T1" resides in 1 tablespace(s):

00001 USERSPACE1
-----
Comment: DB2
Start Time: 20100106133005
End Time: 20100106133006
Status: A
-----
EID: 3 Location: /home/hotel19/username/mydatafile.de1
-----
```

SQLCA Information

```
sqlcaid : SQLCA      sqlcabc: 136   sqlcode: 3107   sqlerrml: 0

sqlerrmc:
sqlerrp : SQLUVLD
sqlerrd : (1) -2146107283      (2) 0           (3) 0
          (4) 0                (5) 0           (6) 0
sqlwarn : (1) W      (2)      (3)      (4)      (5)      (6)
          (7)      (8)      (9)      (10)     (11)
sqlstate:
```

```
Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log Backup ID
-----
B D 20100106135509001 F D S0000000.LOG S0000000.LOG
-----
Contains 2 tablespace(s):

00001 SYSCATSPACE
00002 USERSPACE1
-----
Comment: DB2 BACKUP SAMPLE OFFLINE
Start Time: 20100106135509
End Time: 20100106135512
Status: A
-----
EID: 4 Location: /home/hotel19/username
```

**例 2** 次の出力例は、再編成再利用操作の 1 項目を示しています。

```
db2 -v "list history reorg all for wsdb"
```

```
Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log Backup ID
-----
G T 20080924101408      N      S0000000.LOG S0000000.LOG
-----
Table: "ZHMFENG"."T1"
-----
Comment: REORG RECLAIM
Start Time: 20080924101408
End Time: 20080924101409
Status: A
```

**例 3** **db2\_all** 接頭部を使用して LIST HISTORY コマンドをすべてのデータベース・パーティションで実行します。

```
db2_all "db2 list history since 20010601 for sample"
```

**例 4**

以下に示すのは、DB2 pureScale 環境における DB ヒストリー・レコードの例です。

```
db2 list history since 20091020163200 for database sample
```

```
Op Obj Timestamp+Sequence Type Dev Backup ID
-----
X D 20091020163218      1      D
```

## LIST HISTORY

```
-----  
-----  
Log Stream ID Earliest Log Current Log  
-----  
3 S0000023.LOG C0000000  
-----  
Comment:  
Start Time: 20091020163218  
End Time: 20091020163245  
Status: A  
-----  
EID: 28 Location: /notnfs/billings/arch_logs/billings/SAMPLE/NODE0000/LOGSTREAM0002/C0000000/S0000023.LOG  
-----  
Op Obj Timestamp+Sequence Type Dev Backup ID  
-----  
X D 20091020163219 1 D  
-----  
-----  
Log Stream ID Earliest Log Current Log  
-----  
0 S0000001.LOG C0000000  
-----  
Comment:  
Start Time: 20091020163219  
End Time: 20091020163257  
Status: A  
-----  
EID: 29 Location: /notnfs/billings/arch_logs/billings/SAMPLE/NODE0000/LOGSTREAM0000/C0000000/S0000001.LOG
```

### 例 5

以下に示すのは、DB2 pureScale 環境外における DB ヒストリー・レコードの例です。

db2 list history since 20091020155300 for database sample

```
Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log Backup ID  
-----  
X D 20091020155341 1 D S0000004.LOG C0000000  
-----  
-----  
Comment:  
Start Time: 20091020155341  
End Time: 20091020155345  
Status: A  
-----  
EID: 9 Location: /notnfs/billings/arch_logs/billings/SAMPLE/NODE0000/LOGSTREAM0000/C0000000/S0000004.LOG
```

### 使用上の注意

SYSIBMADM.DB\_HISTORY 管理ビューを使用して、すべてのデータベース・パーティションからデータを取り出すことができます。

DB2 pureScale インスタンスでは、データベースに対するすべての DB ヒストリー・レコードはグローバルです。いずれかのメンバーに接続された、リスト履歴または管理ビュー・インターフェースを使用して、DB ヒストリー・レコードをリトリブできます。

このコマンドによって生成されるレポートには、以下の記号が含まれます。

操作

- A - 表スペースの作成
- B - バックアップ
- C - ロード・コピー
- D - 表のドロップ
- F - ロールフォワード
- G - 再編成
- L - ロード
- N - 表スペースの名前変更
- O - 表スペースのドロップ
- Q - 静止
- R - リストア
- T - 表スペースの変更
- U - アンロード



X - アーカイブ・ログ

#### オブジェクト

D - データベース  
 I - 索引  
 P - 表スペース  
 T - 表  
 R - パーティション表

#### タイプ

表スペースの変更操作タイプ:

C - コンテナの追加  
 R - 再調整

アーカイブ・ログ操作タイプ:

F - フェイルオーバー・アーカイブ・パス  
 M - 2 次 (ミラー) ログ・パス  
 N - アーカイブ・ログ・コマンド  
 P - 1 次ログ・パス  
 1 - 1 次ログ・アーカイブ・メソッド  
 2 - 2 次ログ・アーカイブ・メソッド

バックアップおよびリストア操作タイプ:

D - デルタ・オフライン  
 E - デルタ・オンライン  
 F - オフライン  
 I - 増分オフライン  
 M - Merged  
 N - オンライン  
 O - 増分オンライン  
 R - 再ビルド

ロード操作タイプ:

I - 挿入  
 R - 置換

ロールフォワード操作タイプ:

E - ログの最後  
 P - ポイント・イン・タイム

静止操作タイプ:

S - 静止共有  
 U - 静止更新  
 X - 静止排他  
 Z - 静止リセット

履歴項目状況フラグ:

A - アクティブ  
 D - 削除  
 E - 有効期限切れ  
 I - 非アクティブ  
 N - コミットされていない  
 P - Pending delete  
 X - 削除禁止  
 a - 未完了アクティブ  
 i - 未完了非アクティブ

## LIST INDOUBT TRANSACTIONS

未確定トランザクションのリストを表示します。未確定トランザクションのコミット、ロールバック、または破棄を対話式で実行できます。

2 フェーズ・コミット・プロトコルは、以下のもので構成されます。

1. PREPARE フェーズ。このフェーズでは、リソース・マネージャーがログ・ページをディスクに書き込んでいるので、COMMIT または ROLLBACK プリミティブのどちらにも応答することができます。
2. COMMIT (または ROLLBACK) フェーズ。このフェーズでは、トランザクションの実際のコミットまたはロールバックが実行されます。

トランザクションを取り消すと、ヒューリスティックに完了した (ヒューリスティックにコミットあるいはロールバックされた) トランザクションに保持されていたリソースが解放されます。未確定トランザクションは、準備済みのトランザクションのうち、コミットまたはロールバックが実行されていないものです。

### 有効範囲

このコマンドは、このコマンドが実行されたノード上にある未確定トランザクションのリストを戻します。

### 許可

なし

### 必要な接続

データベース。暗黙接続が可能な場合には、デフォルト・データベースへの接続が確立されます。

### コマンド構文

```
▶▶—LIST INDOUBT TRANSACTIONS—┐
                                └─WITH PROMPTING─┘▶▶
```

### コマンド・パラメーター

#### WITH PROMPTING

未確定トランザクションを処理することを指定します。このパラメーターを指定すると、対話式ダイアログ・モードが開始され、未確定トランザクションのコミット、ロールバック、または破棄が可能になります。このパラメーターを指定しない場合、未確定トランザクションは標準出力装置に書き込まれ、対話式ダイアログ・モードは開始されません。

対話式ダイアログ・モードでは次のことが可能です。

- すべての未確定トランザクションのリスト表示 (1 を入力)
- 未確定トランザクション番号  $x$  のリスト表示 (1 の後に有効なトランザクション番号を入力)
- 終了 (q を入力)

- トランザクション番号  $x$  をコミット (c の後に有効なトランザクション番号を入力)
- トランザクション番号  $x$  をロールバック (r の後に有効なトランザクション番号を入力)
- トランザクション番号  $x$  を破棄 (f の後に有効なトランザクション番号を入力)

コマンド文字と引数の間は、ブランク・スペースで区切る必要があります。トランザクションのコミット、ロールバック、または破棄を実行する前に、トランザクション・データが表示され、アクションを確認するように求められます。

**LIST INDOUBT TRANSACTIONS** コマンドは、それぞれの未確定トランザクションでのデータベースの役割を示す、以下のタイプ 情報を戻します。

- TM** 未確定トランザクションは、データベースをトランザクション・マネージャー・データベースとして使用することを示します。
- RM** 未確定トランザクションは、データベースをリソース・マネージャーとして使用することを示します。つまり、それがトランザクションに参加する複数のデータベースの 1 つであっても、トランザクション・マネージャー・データベースではないことを示します。

## 使用上の注意

未確定トランザクションは、未確定状態のままになっているグローバル・トランザクションです。これは、2 フェーズ・コミット・プロトコルの第 1 フェーズ (つまり PREPARE フェーズ) を正常終了した後、トランザクション・マネージャー (TM)、または少なくとも 1 つのリソース・マネージャー (RM) のいずれかが使用できなくなった場合に発生します。RM がもう一度使用できるようになり、TM が RM からの未確定状況情報に関するログを統合できるようになるまで、RM はトランザクションの分岐をコミットするかそれともロールバックするかがわかりません。未確定トランザクションは MPP 環境でも発生する可能性があります。

現在接続されているデータベースに対して **LIST INDOUBT TRANSACTIONS** が出された場合、そのデータベースの未確定トランザクションに関する情報が戻されます。

コミットできるのは、状況が未確定 (i)、またはコミット肯定応答欠落 (m)、またはフェデレーテッド・コミット肯定応答欠落 (d) のトランザクションだけです。

ロールバックできるのは、状況が未確定 (i)、フェデレーテッド・ロールバック肯定応答欠落 (b)、または終了済み (e) のトランザクションだけです。

取り消しができるのは、状況がコミット済み (c)、ロールバック (r)、フェデレーテッド・コミット肯定応答欠落 (d)、またはフェデレーテッド・ロールバック肯定応答欠落 (b) のトランザクションだけです。

2 フェーズ・コミットのコミット・フェーズでは、コーディネーター・ノードがコミットの肯定応答を待機します。(ノード障害などの理由で) 応答しないノードが 1 つ以上ある場合、そのトランザクションはコミット肯定応答欠落状態になります。

## LIST INDOUBT TRANSACTIONS

未確定トランザクション情報は、コマンドが出された時点でしか有効ではありません。対話式ダイアログ・モードに入ってしまうと、外部の活動のためにトランザクション状況が変更されることがあります。その場合、該当する状態にない未確定トランザクションを処理しようとする、エラー・メッセージが表示されます。

このタイプのエラーが発生した場合、ユーザーは対話式ダイアログを終了 (q) しなければなりません。そして、表示される情報を最新表示にするために、 **LIST INDOUBT TRANSACTIONS WITH PROMPTING** コマンドを再発行する必要があります。

## LIST NODE DIRECTORY

ノード・ディレクトリーの内容をリスト表示します。

### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### ADMIN

Administration Server ノードを指定します。

#### SHOW DETAIL

以下の情報を出力に含めることを指定します。

- リモート・インスタンス名
- システム
- オペレーティング・システムのタイプ

### 例

次に示すのは **LIST NODE DIRECTORY** の出力例です。

```
Node Directory
```

```
Number of entries in the directory = 2
```

```
Node 1 entry:
```

```
Node name           = LANNODE
Comment             =
Directory entry type = LDAP
Protocol            = TCPIP
Hostname            = LAN.db2ntd3.torolab.ibm.com
Service name        = 50000
```

```
Node 2 entry:
```

```
Node name           = TLBA10ME
Comment             =
Directory entry type = LOCAL
Protocol            = TCPIP
Hostname            = tlba10me
Service name        = 447
```

次に示すのは **LIST ADMIN NODE DIRECTORY** の出力例です。

## LIST NODE DIRECTORY

Node Directory

Number of entries in the directory = 2

Node 1 entry:

Node name	= LOCALADM
Comment	=
Directory entry type	= LOCAL
Protocol	= TCPIP
Hostname	= jaguar
Service name	= 523

Node 2 entry:

Node name	= MYDB2DAS
Comment	=
Directory entry type	= LDAP
Protocol	= TCPIP
Hostname	= peng.torolab.ibm.com
Service name	= 523

共通フィールドは、次のとおりです。

### Node name

リモート・ノードの名前。これは、ノードのカタログ時に *nodename* パラメーターに入力された名前に対応します。

### Comment

ノードのカタログ時に入力された、ノードに関連する注釈。ノード・ディレクトリー内の注釈を変更するには、そのノードをアンカタログしてから、別の注釈を付けてもう一度カタログします。

### Directory entry type

LOCAL は、項目がローカル・ノードのディレクトリー・ファイルに見付かったことを意味します。LDAP は、項目が LDAP サーバーまたは LDAP キャッシュで見付かったことを意味します。

### Protocol

ノード用にカタログされた通信プロトコル。

特定のノード・タイプに関連したフィールドについては、該当する **CATALOG ... NODE** コマンドを参照してください。

## 使用上の注意

**DB2LDAPCACHE** その他の変数設定にかかわらず、**LIST DATABASE DIRECTORY** コマンドまたは **LIST NODE DIRECTORY** コマンドを使用するときには、LDAP サーバーからローカル・データベースおよびノード項目のリストが読み取られます。

ノード・ディレクトリーは、個々の **IBM Data Server Runtime Client** で作成および保守されます。これには、そのクライアントからアクセスできるデータベースを含む各リモート・ワークステーションごとに 1 つの項目が含まれています。DB2 クライアントは、データベース接続やインスタンス接続が要求されると、常にノード・ディレクトリー内の通信エンドポイント情報を使います。

データベース・マネージャーは、**CATALOG...NODE** コマンドを処理するたびに、ノード項目を作成してそれをノード・ディレクトリーに追加します。その項目は、ノードが使用する通信プロトコルによって異なります。

ノード・ディレクトリーには、次のようなタイプのノード用のエントリーを含めることができます。

- LDAP
- ローカル
- Named PIPE
- TCPIP
- TCPIP4
- TCPIP6

## LIST ODBC DATA SOURCES

使用できるユーザーまたはシステム ODBC データ・ソースのすべてのリストを表示します。

ODBC (Open Database Connectivity) でのデータ・ソース という語は、特定のデータベースのユーザー定義名のことです。この名前は、ODBC API を介してデータベースまたはファイル・システムにアクセスするときに使用されます。Windows では、ユーザー・データ・ソースまたはシステム・データ・ソースのどちらであってもカタログできます。ユーザー・データ・ソースはそれをカタログしたユーザーにのみ可視になりますが、システム・データ・ソースは他のすべてのユーザーから可視であり使用できます。

このコマンドは Windows のみで使用できます。

### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

**USER** ユーザー ODBC データ・ソースのみリスト表示します。キーワードを指定しない場合、これがデフォルトです。

### SYSTEM

システム ODBC データ・ソースのみリスト表示します。

### 例

以下に示すのは、**LIST ODBC DATA SOURCES** コマンドの出力例です。

User ODBC Data Sources

Data source name	Description
SAMPLE	IBM DB2 ODBC DRIVER



## LIST PACKAGES/TABLES

現行データベースに関連付けられているパッケージまたは表のリストを表示します。

### 許可

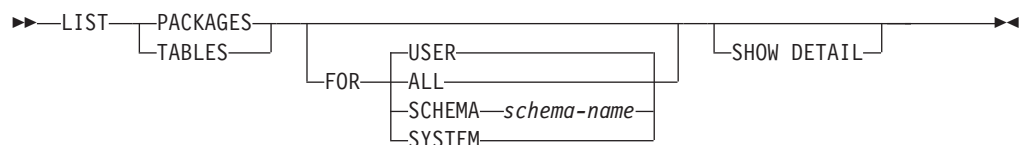
システム・カタログ SYSCAT.PACKAGES (**LIST PACKAGES**) と SYSCAT.TABLES (**LIST TABLES**) の場合には、以下のいずれかが必要です。

- SYSADM
- SYSCTRL
- SYSMOINT
- SYSMON
- DBADM
- CONTROL 特権
- SELECT 特権

### 必要な接続

データベース。暗黙接続が可能な場合には、デフォルト・データベースへの接続が確立されます。

### コマンド構文



### コマンド・パラメーター

**FOR** **FOR** 節を指定しないなら、**USER** のパッケージまたは表のリストが表示されます。

**ALL** データベース内のすべてのパッケージまたは表のリストが表示されます。

**SCHEMA** *schema-name*

指定されたスキーマのデータベース内のすべてのパッケージまたは表のリストのみ表示します。

**SYSTEM**

データベース内のすべてのシステム・パッケージまたは表のリストのみ表示します。

**USER** 現行ユーザーのデータベース内のすべてのユーザー・パッケージまたは表のリストを表示します。

### SHOW DETAIL

このオプションを **LIST TABLES** コマンドと共に指定した場合、表名とスキーマ名の全体が表示されます。このオプションを指定しなかった場合、表名は 30 文字で切り捨てられ、31 列目の ">" 記号が表名の切り捨て位置を

## LIST PACKAGES/TABLES

表します。スキーマ名は 14 文字で切り捨てられ、15 列目の ">" 記号がスキーマ名の切り捨て位置を表します。このオプションを **LIST PACKAGES** コマンドと共に指定した場合、パッケージ・スキーマの全体 (作成者)、バージョン、結合 ID、およびパッケージの `unique_id` (16 進数で示される整合性トークン) が表示されます。このオプションを指定しなかった場合、スキーマ名および結合 ID は 8 文字で切り捨てられ、9 列目の ">" 記号がスキーマまたは結合 ID の切り捨て位置を表します。バージョンは 10 文字で切り捨てられ、11 列目の ">" 記号がバージョンの切り捨て位置を表します。

### 例

次に示すのは **LIST PACKAGES** の出力例です。

Package	Schema	Version	Bound by	Total sections	Valid	Format	Isolation level	Blocking
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

次に示すのは **LIST TABLES** の出力例です。

Table/View	Schema	Type	Creation time
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

9 record(s) selected.

### 使用上の注意

**LIST PACKAGES** コマンドおよび **LIST TABLES** コマンドは、システム・カタログ表への簡単なインターフェースを提供します。

以下の **SELECT** ステートメントは、システム・カタログ表で検出した情報を戻します。このステートメントを拡張して、システム・カタログ表が提供するその他の情報を選択することもできます。

```
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
```

## LIST PACKAGES/TABLES

```
where tabschema = 'SYSCAT'  
order by tabschema, tablename;
```

```
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 tablename, tabschema, type, create_time  
from syscat.tables  
where tabschema = USER  
order by tabschema, tablename;
```

```
select pkgname, pkgschema, pkgversion, unique_id, boundby, total_sect,  
       valid, format, isolation, blocking  
from syscat.packages  
where pkgschema = USER  
order by pkgschema, pkgname, pkgversion;
```

## LIST TABLESPACE CONTAINERS

指定した表スペースのコンテナーのリストを表示します。

**重要:** このコマンドおよび API はバージョン 9.7 で非推奨となっており、将来のリリースで除去される可能性があります。代わりに、`MON_GET_TABLESPACE` および `MON_GET_CONTAINER` 表関数を使用できます。これらは、より多くの情報を返します。詳しくは、『LIST TABLESPACES コマンドおよび LIST TABLESPACE CONTAINERS コマンドが推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055001.html>) を参照してください。

表スペースのスナップショットには、**LIST TABLESPACE CONTAINERS** コマンドによって表示されるすべての情報が含まれます。

### 有効範囲

このコマンドは、それが実行されたノードに対してだけ情報を戻します。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMANT
- SYSMON
- DBADM

### 必要な接続

データベース

### コマンド構文

```
▶▶—LIST TABLESPACE CONTAINERS FOR—tablespace-id—┐
└—SHOW DETAIL—┘▶▶
```

### コマンド・パラメーター

**FOR** *tablespace-id*

現行データベースで使用する表スペースを表す固有の整数。現行データベースで使用するすべての表スペースのリストを表示するには、**LIST TABLESPACES** コマンドを使用します。

**SHOW DETAIL**

このオプションを指定しない場合、各コンテナーごとに以下の基本情報だけが表示されます。

- コンテナー ID
- 名前
- タイプ (ファイル、ディスク、またはパス)

## LIST TABLESPACE CONTAINERS

このオプションを指定した場合は、各コンテナーに関して下記の付加的な情報が表示されます。

- ページの合計数
- 使用できるページの数
- アクセス可能性 (yes または no)

### 例

次に示すのは **LIST TABLESPACE CONTAINERS FOR 0** の出力例です。

Tablespace Containers for Tablespace 0

```
Container ID      = 0
Name              = /home/smith/smith/NODE0000/SQL00001/SQLT0000.0
Type              = Path
```

次に示すのは、**LIST TABLESPACE CONTAINERS FOR 0 SHOW DETAIL** を指定した場合の出力例です。

Tablespace Containers for Tablespace 0

```
Container ID      = 0
Name              = /home/smith/smith/NODE0000/SQL00001/SQLT0000.0
Type              = Path
Total pages      = 895
Useable pages    = 895
Accessible       = Yes
```

---

### LIST TABLESPACES

現行データベースの表スペースとその情報のリストを表示します。

**重要:** このコマンドおよび API はバージョン 9.7 で非推奨となっており、将来のリリースで除去される可能性があります。代わりに、MON\_GET\_TABLESPACE および MON\_GET\_CONTAINER 表関数を使用できます。これらは、より多くの情報を返します。詳しくは、『LIST TABLESPACES コマンドおよび LIST TABLESPACE CONTAINERS コマンドが推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055001.html>) を参照してください。

このコマンドによって表示される情報は、表スペースのスナップショットでも使用できます。

#### 有効範囲

このコマンドは、それが実行されたデータベース・パーティションに関する情報だけを返します。

#### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMMAINT
- SYSMON
- DBADM
- LOAD authority

#### 必要な接続

データベース

#### コマンド構文

```
►►—LIST TABLESPACES—┐SHOW DETAIL┘◄◄
```

#### コマンド・パラメーター

##### SHOW DETAIL

このオプションを指定しない場合、各表スペースごとに以下の基本情報だけが表示されます。

- 表スペース ID
- 名前
- タイプ (システム管理スペースまたはデータベース管理スペース)
- 内容 (すべてのデータ、長形式または索引データ、または一時データ)

- 状態。現在の表スペースの状態を示す 16 進値。外から見える表スペースの状態は、特定の状態値の 16 進値を合計したものです。例えば、状態が "quiesced: EXCLUSIVE" かつ "Load pending" の場合、その値は  $0x0004 + 0x0008$ 、つまり  $0x000c$  となります。 **db2tbst** (表スペース状態の獲得) コマンドを使うと、特定の 16 進値と関連した表スペース状態を取得できます。以下は、`sqlutil.h` に示されているビット定義です。

0x0	Normal
0x1	Quiesced: SHARE
0x2	Quiesced: UPDATE
0x4	Quiesced: EXCLUSIVE
0x8	Load pending
0x10	Delete pending
0x20	Backup pending
0x40	Roll forward in progress
0x80	Roll forward pending
0x100	Restore pending
0x100	Recovery pending (not used)
0x200	Disable pending
0x400	Reorg in progress
0x800	Backup in progress
0x1000	Storage must be defined
0x2000	Restore in progress
0x4000	Offline and not accessible
0x8000	Drop pending
0x10000	Suspend Write
0x20000	Load in progress
0x2000000	Storage may be defined
0x4000000	StorDef is in 'final' state
0x8000000	StorDef was change before roll forward
0x10000000	DMS rebalance in progress
0x20000000	Table space deletion in progress
0x40000000	Table space creation in progress

注: DB2 LOAD は、表スペースを ロード・ペンディング状態や削除ペンディング状態に設定することはありません。

このオプションを指定した場合は、各表スペースに関して下記の付加的な情報が表示されます。

- ページの合計数
- 使用できるページの数
- 使用されたページの数
- 未使用ページの数
- 最高水準点 (ページ単位)
- ページ・サイズ (バイト単位)
- エクステンツ・サイズ (ページ単位)
- プリフェッチ・サイズ (ページ単位)
- コンテナの数
- 最小リカバリー時間 (表スペースをロールフォワード可能な最も早いポイント・イン・タイム。UTC 時間で表記されるタイム・スタンプで、ゼロでない場合にのみ表示されます)
- 状態変更表スペース ID (表スペース状態が "load pending" または "delete pending" の場合のみ表示)

## LIST TABLESPACES

- 状態変更オブジェクト ID (表スペース状態が "load pending" または "delete pending" の場合のみ表示)
- 静止者の数 (表スペース状態が "quiesced: SHARE"、"quiesced: UPDATE"、または "quiesced: EXCLUSIVE" の場合のみ表示)
- 各静止プログラムごとに表スペース ID とオブジェクト ID (静止プログラムの数が 0 より大きい場合のみ表示)

### 例

下記に示すのは、**LIST TABLESPACES SHOW DETAIL** の 2 つの出力例です。

```
Tablespaces for Current Database
Tablespace ID          = 0
Name                   = SYSCATSPACE
Type                   = Database managed space
Contents               = Any data
State                  = 0x0000
  Detailed explanation:
    Normal
Total pages            = 895
Useable pages          = 895
Used pages             = 895
Free pages             = Not applicable
High water mark (pages) = Not applicable
Page size (bytes)     = 4096
Extent size (pages)   = 32
Prefetch size (pages) = 32
Number of containers   = 1

Tablespace ID          = 1
Name                   = TEMPSPACE1
Type                   = System managed space
Contents               = Temporary data
State                  = 0x0000
  Detailed explanation:
    Normal
Total pages            = 1
Useable pages          = 1
Used pages             = 1
Free pages             = Not applicable
High water mark (pages) = Not applicable
Page size (bytes)     = 4096
Extent size (pages)   = 32
Prefetch size (pages) = 32
Number of containers   = 1

Tablespace ID          = 2
Name                   = USERSPACE1
Type                   = Database managed space
Contents               = Any data
State                  = 0x000c
  Detailed explanation:
    Quiesced: EXCLUSIVE
    Load pending
Total pages            = 337
Useable pages          = 337
Used pages             = 337
Free pages             = Not applicable
High water mark (pages) = Not applicable
Page size (bytes)     = 4096
Extent size (pages)   = 32
Prefetch size (pages) = 32
Number of containers   = 1
```



## LIST TABLESPACES

```

State change tablespace ID      = 2
State change object ID          = 3
Number of quiescers             = 1
  Quiescer 1:
    Tablespace ID                = 2
    Object ID                    = 3

```

DB21011I In a partitioned database server environment, only the table spaces on the current node are listed.

### Tablespaces for Current Database

```

Tablespace ID                    = 0
Name                              = SYSCATSPACE
Type                              = System managed space
Contents                          = Any data
State                              = 0x0000

```

Detailed explanation:

Normal

```

Total pages                      = 1200
Useable pages                    = 1200
Used pages                       = 1200
Free pages                       = Not applicable
High water mark (pages)         = Not applicable
Page size (bytes)               = 4096
Extent size (pages)              = 32
Prefetch size (pages)           = 32
Number of containers             = 1

```

```

Tablespace ID                    = 1
Name                              = TEMPSPACE1
Type                              = System managed space
Contents                          = Temporary data
State                              = 0x0000

```

Detailed explanation:

Normal

```

Total pages                      = 1
Useable pages                    = 1
Used pages                       = 1
Free pages                       = Not applicable
High water mark (pages)         = Not applicable
Page size (bytes)               = 4096
Extent size (pages)              = 32
Prefetch size (pages)           = 32
Number of containers             = 1

```

```

Tablespace ID                    = 2
Name                              = USERSPACE1
Type                              = System managed space
Contents                          = Any data
State                              = 0x0000

```

Detailed explanation:

Normal

```

Total pages                      = 1
Useable pages                    = 1
Used pages                       = 1
Free pages                       = Not applicable
High water mark (pages)         = Not applicable
Page size (bytes)               = 4096
Extent size (pages)              = 32
Prefetch size (pages)           = 32
Number of containers             = 1

```

```

Tablespace ID                    = 3
Name                              = DMS8K
Type                              = Database managed space
Contents                          = Any data
State                              = 0x0000

```

Detailed explanation:

## LIST TABLESPACES

```
      Normal
Total pages                = 2000
Useable pages              = 1952
Used pages                 = 96
Free pages                 = 1856
High water mark (pages)   = 96
Page size (bytes)         = 8192
Extent size (pages)       = 32
Prefetch size (pages)     = 32
Number of containers       = 2

Tablespace ID              = 4
Name                       = TEMP8K
Type                       = System managed space
Contents                   = Temporary data
State                      = 0x0000
  Detailed explanation:
    Normal
Total pages                = 1
Useable pages              = 1
Used pages                 = 1
Free pages                 = Not applicable
High water mark (pages)   = Not applicable
Page size (bytes)         = 8192
Extent size (pages)       = 32
Prefetch size (pages)     = 32
Number of containers       = 1
DB21011I In a partitioned database server environment, only the table spaces
on the current node are listed.
```

### 使用上の注意

パーティション・データベース環境では、このコマンドがデータベースのすべての表スペースを戻すわけではありません。すべての表スペースのリストを表示するには、SYSCAT.TABLESPACES を照会します。

**LIST TABLESPACES SHOW DETAIL** コマンドが発行されると、表スペース内の保留中空きエクステントすべての解放が試行されます。保留中のフリー・エクステントが正常に解放された場合には、レコードがログに記録されます。

表スペースのバランス調整操作中には、使用できるページ数には新しく追加されたコンテナのページ数が含まれますが、バランス調整完了までの間、それらの新しいページは、未使用ページ数に反映されません。表スペースのバランス調整が実行されていない場合、使用されたページの数と未使用ページ数を合計すると、使用できるページ数の値に等しくなります。

## LIST UTILITIES

インスタンス上のアクティブなユーティリティのリストを標準出力に表示します。各ユーティリティの記述には、開始時刻、説明、スロットルの優先順位 (該当する場合)、進捗モニター情報 (該当する場合) などの属性が含まれます。

### 有効範囲

このコマンドは、すべてのデータベース・パーティションの情報を戻します。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMANT
- SYSMON

### 必要な接続

インスタンス

### コマンド構文

►►—LIST UTILITIES — [SHOW DETAIL] —◄◄

### コマンド・パラメーター

#### SHOW DETAIL

進捗モニター機能をサポートしているユーティリティに関する詳細な進捗情報を表示します。

### 例

表 `some_table` に対する `RUNSTATS` 呼び出し:

```
LIST UTILITIES
ID                        = 1
Type                     = RUNSTATS
Database Name            = PROD
Description              = krrose.some_table
Start Time               = 12/19/2003 11:54:45.773215
Priority                  = 10
```

オフライン・データベース・バックアップのパフォーマンスのモニター:

```
LIST UTILITIES SHOW DETAIL
ID                        = 2
Type                     = BACKUP
Database Name            = SAMPLE
Description              = offline db
Start Time               = 10/30/2003 12:55:31.786115
Priority                  = 0
Progress Monitoring:
```

## LIST UTILITIES

```
Phase Number [CURRENT] = 1
Description              =
Work Metric              = BYTES
Total Work Units        = 20232453
Completed Work Units    = 230637
Start Time              = 10/30/2003 12:55:31.786115
```

### 使用上の注意

このコマンドは、実行中のユーティリティーの状況をモニターするために使用します。例えば、オンライン・バックアップの進捗状況をモニターするためにこのユーティリティーを使用できます。また、パフォーマンス上の問題を調査するときに、このコマンドを使用して、どのユーティリティーが実行中になっているかを確認することもできます。パフォーマンス低下の原因になっていると考えられるユーティリティーがスロットル機能をサポートしていれば、そのユーティリティーのスロットルを絞ることも可能です。 **LIST UTILITIES** コマンドで表示される ID は、**SET UTIL\_IMPACT\_PRIORITY** コマンドで使用する ID と同じです。

**LIST UTILITIES** コマンドを使用して、非同期索引クリーンアップによる索引の据え置きクリーンアップの進行をモニターできます。

DB2 バージョン 9.7 フィックスパック 1 以降では、**LIST UTILITIES** コマンドを使用して、非同期パーティション・デタッチ・タスクによるパーティション表からのデータ・パーティションのデタッチの完了までの進行をモニターできます。データ・パーティション表からのデータ・パーティションのデタッチは、**DETACH PARTITION** 節を指定した **ALTER TABLE** ステートメントを発行して開始します。

## LOAD

データを DB2 表にロードします。

サーバー上に保管するデータは、ファイル、テープ、または名前付きパイプの形式にすることができます。リモートに接続されたクライアント上に保管するデータは、完全修飾ファイルまたは名前付きパイプの形式にすることができます。また、データは、ユーザー定義カーソルから、あるいはユーザー作成のスクリプトまたはアプリケーションを使用してロードできます。表の COMPRESS 属性が YES に設定されている場合、ロードされるデータは、表内にディクショナリーが存在するデータおよびデータベース・パーティションごとに圧縮の対象となります。これには表の XML ストレージ・オブジェクト内のデータが含まれます。

460 ページの『ロード・ユーティリティー用のファイル・タイプ修飾子』へのクイック・リンク。

### 制約事項

ロード・ユーティリティーでは、階層レベルのデータのロードはサポートされていません。ロード・ユーティリティーには、範囲がクラスター化された表との互換性はありません。ロード・ユーティリティーでは、NOT LOGGED INITIALLY パラメーターを CREATE TABLE や ALTER TABLE ステートメントでサポートしていません。

### 有効範囲

このコマンドは、一度の要求で複数のデータベース・パーティションに対して発行できます。

### 許可

以下の権限のいずれか。

- DATAACCESS
- データベースに対する LOAD 権限と以下の特権
  - 表の INSERT 特権 (ロード・ユーティリティーが INSERT モード、TERMINATE モード、または RESTART モードで呼び出される場合 (TERMINATE モードは直前のロード挿入操作を終了するためのもので、RESTART モードは直前のロード挿入操作を再開するためのものです))
  - 表の INSERT および DELETE 特権 (ロード・ユーティリティーが REPLACE モード、TERMINATE モード、または RESTART モードで呼び出される場合 (TERMINATE モードは直前のロード置換操作を終了するためのもので、RESTART モードは直前のロード置換操作を再開するためのものです))
  - 例外表の INSERT 特権 (例外表をロード操作の一部として使用する場合)。
- 保護列のある表にデータをロードするには、その表内のすべての保護列への書き込みアクセスを許可する LBAC 資格情報を、セッション許可 ID が直接保持しているか、あるいはグループまたはロールを介して間接的に保持している必要があります。そうでない場合は、ロードが失敗してエラー (SQLSTATE 5U014) が戻されます。

## LOAD

- 保護された行を持つ表にデータをロードするには、セッション許可 ID が、以下の基準を満たすセキュリティ・ラベルを保持していなければなりません。
  - セキュリティ・ラベルは表を保護しているセキュリティ・ポリシーの一部である
  - セキュリティ・ラベルは、書き込みアクセスまたはすべてのアクセス用にセッション許可 ID に直接付与されているか、グループまたはロールを介して間接的に付与されている。

こうしたセキュリティ・ラベルをセッション許可 ID が保持していない場合は、ロードが失敗してエラー (SQLSTATE 5U014) が戻されます。セキュリティ・ラベルは、セッション許可 ID LBAC 信用証明情報が、データ内のロードされる行を保護するセキュリティ・ラベルにその許可 ID が書き込むことを許可しない場合に、その行を保護します。ただし、表を保護しているセキュリティ・ポリシーが CREATE SECURITY POLICY ステートメントの RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL オプションを使用して作成されている場合は、その状況にはなりません。その場合は、ロードが失敗してエラー (SQLSTATE 42519) が戻されます。

保護行がある表にデータをロードする場合、ターゲット表にはデータ・タイプ DB2SECURITYLABEL の列が 1 つあります。入力データ行にその列の値が含まれていない場合、その行はリジェクトされます。ただし、LOAD コマンドで `usedefaults` ファイル・タイプ修飾子が指定されている場合は例外です。その場合には、その表を保護するセキュリティ・ポリシーの書き込みアクセス用のセキュリティ・ラベルが使用されます。書き込みアクセス用のセキュリティ・ラベルがない場合、その行はリジェクトされ、次の行に処理が続行されます。

- REPLACE オプションを指定する場合、セッション許可 ID は表をドロップできる権限を持っていないければなりません。
- LOCK WITH FORCE オプションが指定される場合、SYSADM 権限が必要です。
- 行アクセス制御がアクティブになっている表に対して **LOAD REPLACE** を使用する場合は、その表をドロップするための権限が必要です。具体的には、表に対する CONTROL または DBADM のいずれかが必要になります。

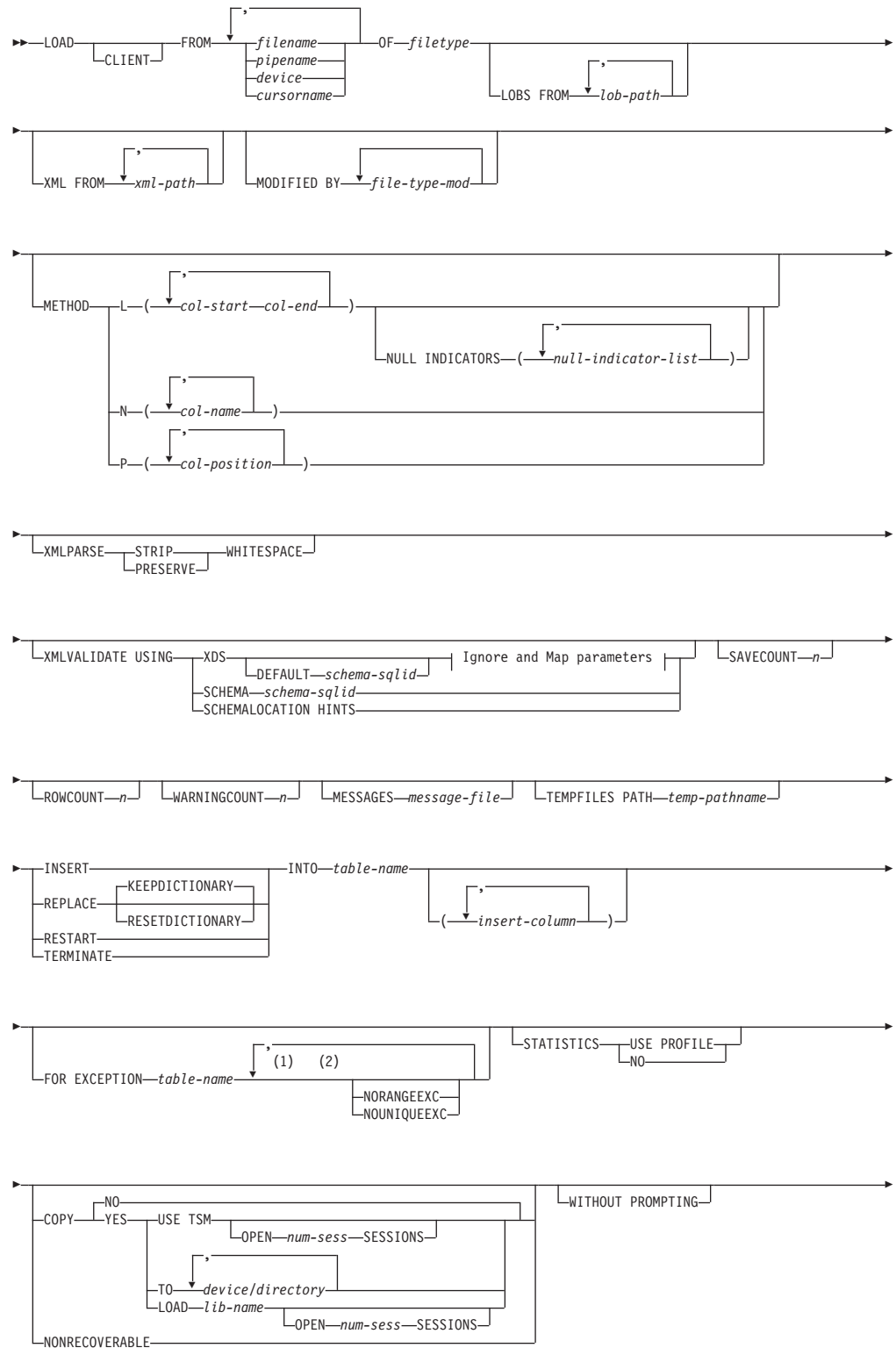
すべてのロード・プロセス (および一般にすべての DB2 サーバー・プロセス) はインスタンス所有者によって所有されており、それらのプロセスすべてにおいて、必要なファイルにアクセスするためにそのインスタンス所有者の ID を使用するため、インスタンス所有者には入力データ・ファイルに対する読み取りアクセス権が必要です。このコマンドをだれが呼び出すかには関係なく、それらの入力データ・ファイルをインスタンス所有者から読むことができなければなりません。

### 必要な接続

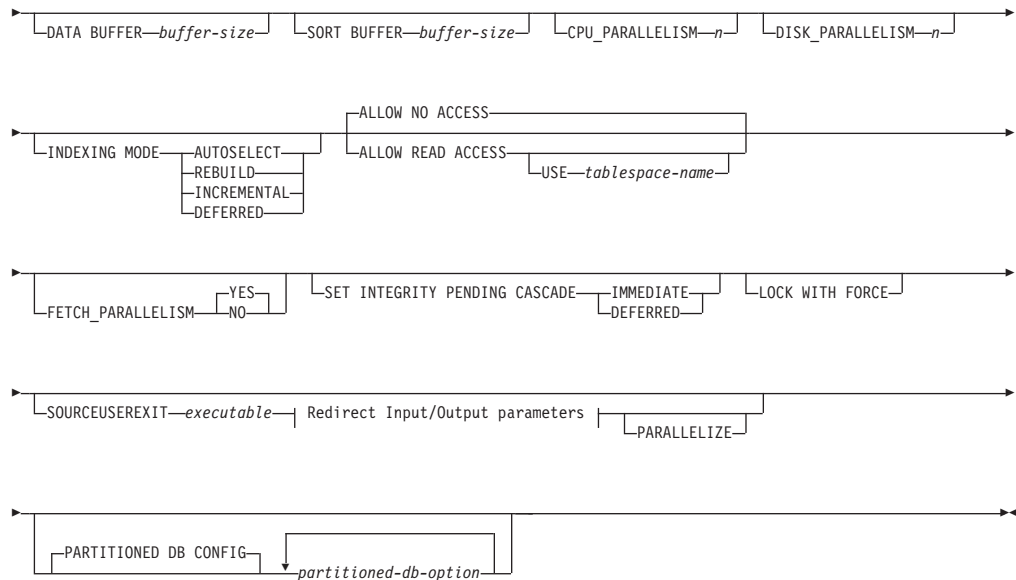
データベース。暗黙接続が可能な場合には、デフォルト・データベースへの接続が確立されます。

インスタンス。明示的なアタッチは必要ありません。データベースへの接続が確立されている場合には、ローカル・インスタンスへの暗黙的な接続が試みられます。

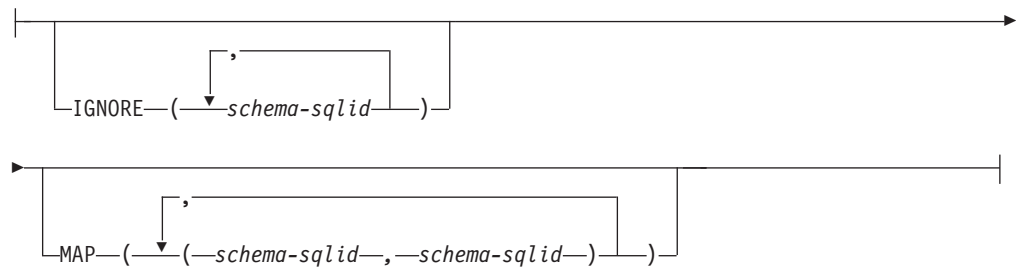
## コマンド構文



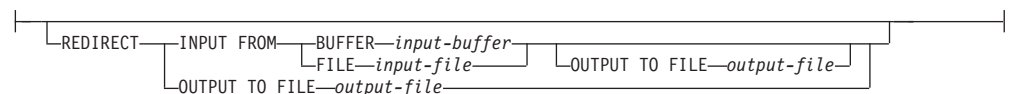
## LOAD



### Ignore and Map parameters:



### Redirect Input/Output parameters:



#### 注:

- 1 これらのキーワードの出現順序は任意です。
- 2 これらのキーワードは、それぞれ 1 回だけ出現できます。

### コマンド・パラメーター

**CLIENT** ロードするデータが、リモートに接続するクライアントにあることを指定します。ロード操作がリモート・クライアントから呼び出されない場合、このオプションは無視されます。 **CURSOR** ファイル・タイプと併せて指定された場合、このオプションは無視されます。

#### 注:



1. **dumpfile** および **lobsinfile** 修飾子および **XML FROM** オプションは、**CLIENT** キーワードが指定されている場合でも、サーバー上のファイルを参照します。
2. コード・ページ変換は、リモートのロード操作時には実行されません。データのコード・ページがサーバーのコード・ページとは異なる場合、**codepage** 修飾子を使用してデータのコード・ページを指定する必要があります。

以下の例では、リモートに接続されたクライアント上に存在するデータ・ファイル (/u/user/data.del) は、サーバー・データベース上の **MYTABLE** にロードされます。

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

**FROM** *filename* | *pipename* | *device* | *cursorname*

ロードされるデータを含んだ SQL ステートメントを参照するファイル、パイプ、装置、またはカーソルを指定します。入力ソースがファイル、パイプ、または装置の場合、**CLIENT** オプションが指定されていなければ、データベースが存在するデータベース・パーティションになければなりません。

複数の名前を指定すると、それらは順番に処理されます。最後に指定した項目がテープ装置の場合は、別のテープを使用するようユーザーに対してプロンプトが出ます。有効な応答オプションは、次のとおりです。

- c** 続行。警告メッセージを生成した装置の使用を続けます (例えば、新しいテープをマウントしたときなど)。
- d** 装置の終了。警告メッセージを生成した装置の使用を停止します (例えば、それ以上テープがない場合)。
- t** 終了。すべての装置を終了します。

注:

- 完全修飾ファイル名を使用します。リモート・サーバーの場合は、常に完全修飾ファイル名を使用する必要があります。呼び出し側と同じデータベース・パーティションにデータベースが存在する場合には、相対パスを使用することもできます。
- **ADMIN\_CMD** プロシージャを使用した **EXPORT** コマンドを使用してデータをファイルにエクスポートした場合、そのデータ・ファイルは **fenced** ユーザー ID によって所有されます。このファイルは通常、インスタンス所有者がアクセスすることはできません。LOAD を CLP または **ADMIN\_CMD** プロシージャから実行するには、インスタンス所有者 ID はデータ・ファイルにアクセスできなければならないので、データ・ファイルに対する読み取りアクセス権限をインスタンス所有者に付与する必要があります。
- ファイルが物理的には分割されてはいるが論理的には 1 つのファイルである場合には、複数の **IXF** ファイルからのデータのロードがサポートされています。ファイルが論理的にも物理的にも分割されている場合は、サポートされていません。論理的にも物理的にも分割されているファイルを複数指定した場合、最初のファイルより後のファイルはすべて無視される

## LOAD

ことに注意してください。(複数の物理ファイルがすべて一度の **EXPORT** コマンドの呼び出しで作成された場合、それらは論理的には 1 つであると見なされます。)

- クライアント・マシン上に存在するデータをロードする場合、そのデータは、完全修飾ファイルまたは名前付きパイプのいずれかの形式でなければなりません。
- XML データをファイルからパーティション・データベース環境にある表にロードする場合、XML データ・ファイルはロードが実行されるすべてのデータベース・パーティションから読み取りアクセス可能でなければなりません。

### **OF** *filetype*

データのフォーマットを指定します。

- ASC (区切りなし ASCII フォーマット)
- DEL (区切り付き ASCII フォーマット)
- IXF (統合交換フォーマット、PC バージョン) は、DB2 データベースによってのみ使用されるバイナリー・フォーマットです。
- CURSOR (SELECT または VALUES ステートメントに対して宣言されたカーソル)。

### 注:

- CURSOR ファイル・タイプを使用して XML データを分散データベース環境にある表にロードする場合、PARTITION\_ONLY および LOAD\_ONLY モードはサポートされません。
- DECLARE CURSOR ステートメントで DATABASE キーワードを指定した CURSOR ファイル・タイプを使用してロードを実行する場合、ロードは、データをフェッチするための別のアプリケーションを内部的に生成します。一方、DATABASE キーワードを指定しなかった場合、ロードは同じアプリケーション内でデータをフェッチします。この 2 つのケースの違いにより、ロック動作にも違いが生じます。特に、DATABASE キーワードを指定しており、現在接続しているデータベースと同じデータベースをそこで使用している場合 (また、ユーザー ID およびパスワードを指定している場合はそれらが現行接続と同じである場合)、ロックの問題 (データベース構成によって異なるが、ロック待機やロック・タイムアウトなど) が発生する可能性があります。これは、DATABASE キーワードを省略することで回避できます。

### **LOBS FROM** *lob-path*

ロードする LOB 値が収められているデータ・ファイルへのパス。パスの最後はスラッシュでなければなりません。 **CLIENT** オプションを指定した場合、パスは完全修飾しなければなりません。 LOB データ・ファイルの名前は、メイン・データ・ファイル (ASC、DEL、または IXF) の、LOB 列にロードされる列内に保管されます。指定できるパスの最大数は 999 です。これによって、**LOBSINFILE** 動作が暗黙的にアクティブ化されます。

CURSOR ファイル・タイプと併せて指定された場合、このオプションは無視されます。

**MODIFIED BY** *file-type-mod*

ファイル・タイプ修飾子オプションを指定します。 460 ページの『ロード・ユーティリティー用のファイル・タイプ修飾子』を参照してください。

**METHOD**

- L** データのロードを開始する列および終了する列の番号を指定します。列の番号は、データの行の先頭からのバイト単位のオフセットです。この番号は 1 から始まります。この方式は、ASC ファイルの場合にのみ使用することができ、そのファイル・タイプでは唯一の有効な方式です。

**NULL INDICATORS** *null-indicator-list*

このオプションは、**METHOD L** パラメーターを指定した場合だけ使用できます (つまり、入力ファイルが ASC ファイルの場合)。NULL 標識リストは、コンマで区切られた正の整数のリストで、各 NULL 標識フィールドの列の番号を指定します。列の番号は、データの行の先頭からのバイト単位の、各 NULL 標識フィールドのオフセットです。NULL 標識リストには、**METHOD L** パラメーターで定義された各データ・フィールドに対する 1 つの項目がなければなりません。列の番号がゼロであることは、対応するデータ・フィールド内に必ずデータがあることを示します。

NULL 標識列中の Y の値は、その列データが NULL であることを指定します。NULL 標識列に Y 以外の文字を指定した場合は、列データが NULL ではなく、**METHOD L** オプションで指定された列データがロードされることを指定することになります。

NULL 標識文字は、**MODIFIED BY** オプションを使用して変更できます。

- N** ロードするデータ・ファイルの中の列の名前を指定します。これらの列名の大文字小文字の区別は、システム・カタログ内の対応する名前の大文字小文字の区別と一致しなければなりません。NULL 可能ではない各表の列には、**METHOD N** リスト内に対応する項目が必要です。例えば、データ・フィールドが F1、F2、F3、F4、F5、および F6 であり、表の列が C1 INT、C2 INT NOT NULL、C3 INT NOT NULL、および C4 INT の場合、method N (F2, F1, F4, F3) は有効な要求ですが、method N (F2, F1) は無効です。この方式は、ファイル・タイプ IXF または CURSOR の場合にのみ使用することができます。
- P** ロードする入力データ・フィールドのフィールド番号 (1 から始まる) を指定します。NULL 可能ではない各表の列には、**METHOD P** リスト内に対応する項目が必要です。例えば、データ・フィールドが F1、F2、F3、F4、F5、および F6 であり、表の列が C1 INT、C2 INT NOT NULL、C3 INT NOT NULL、および C4 INT の場合、method P (2, 1, 4, 3) は有効な要求ですが、method P (2, 1) は無効です。この方式は、ファイル・タイプ IXF、DEL、または CURSOR の場合にのみ使用でき、DEL ファイル・タイプに対して有効な唯一の方式です。

method P によって指定したフィールドごとに、対応する列をアクション・ステートメント内に定義する必要があります。ただし、次の例に示すようにすべての列が対象になる場合、および最初の *x* 列がロードされる場合は必要ありません。

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

#### XML FROM *xml-path*

XML ファイルが含まれているパスを 1 つ以上指定します。XDS は、メイン・データ・ファイル (ASC、DEL、または IXF) の、XML 列にロードされる列内に入れられます。

#### XMLPARSE

XML 文書の解析方法を指定します。このオプションが指定されていない場合、XML 文書の解析の動作は、CURRENT XMLPARSE OPTION 特殊レジスターの値によって決まります。

#### STRIP WHITESPACE

XML 文書の解析時に空白文字を除去することを指定します。

#### PRESERVE WHITESPACE

XML 文書の解析時に空白文字を除去しないことを指定します。

#### XMLVALIDATE

該当する場合に、XML 文書がスキーマに準拠しているかどうかの妥当性検査を実行することを指定します。

#### USING XDS

メイン・データ・ファイルの中で XML Data Specifier (XDS) によって指定されている XML スキーマに準拠しているかどうかについて、XML 文書の妥当性検査が実行されます。デフォルトでは、**USING XDS** 節によって **XMLVALIDATE** オプションが呼び出された場合、妥当性検査実行のために使用されるスキーマは、その XDS の SCH 属性によって決まります。XDS の中で SCH 属性が指定されていない場合、**DEFAULT** 節によってデフォルト・スキーマが指定されているのでない限り、スキーマ妥当性検査は実行されません。

**DEFAULT**、**IGNORE**、および **MAP** 節を使用することにより、スキーマ決定の動作を変更することができます。これら 3 つの節はオプションであり、相互に適用されるのではなく XDS の指定に直接適用されます。例えば、**DEFAULT** 節で指定されているためにあるスキーマが選択された場合、それが **IGNORE** 節で指定されていたとしても無視されることはありません。同じように、**MAP** 節のペアの最初の部分で指定されているためにあるスキーマが選択された場合、それが別の **MAP** 節のペアの 2 番目の部分で指定されていたとしても再びマップされることはありません。

#### USING SCHEMA *schema-sqlid*

指定されている SQL ID の XML スキーマに準拠しているかどうかについて、XML 文書の妥当性検査が実行されます。この場合、すべての XML 列について XML Data Specifier (XDS) の SCH 属性は無視されます。

**USING SCHEMALOCATION HINTS**

ソース XML 文書の中で XML スキーマ・ロケーション・ヒントによって指定されているスキーマに準拠しているかどうかについて、XML 文書の妥当性検査が実行されます。その XML 文書の中に schemaLocation 属性が指定されていない場合、妥当性検査は実行されません。**USING SCHEMALOCATION HINTS** 節が指定されているなら、すべての XML 列について XML Data Specifier (XDS) の SCH 属性は無視されます。

以下のセクションに示す **XMLVALIDATE** オプションの例を参照してください。

**IGNORE** *schema-sqlid*

このオプションは、**USING XDS** パラメーターを指定した場合にのみ使用できます。**IGNORE** 節は、SCH 属性によって指定されていても無視するスキーマとして、1 つ以上のスキーマのリストを指定します。ロードする XML 文書の XML Data Specifier の中に SCH 属性が存在し、その SCH 属性によって指定されるスキーマが無視するスキーマ・リストに含まれている場合には、ロードするその XML 文書についてスキーマ妥当性検査は実行されません。

**注:**

あるスキーマが **IGNORE** 節の中で指定されている場合、**MAP** 節のスキーマ・ペアの左辺にそれを含めることはできません。

**IGNORE** 節は XDS にのみ適用されます。あるスキーマが **IGNORE** 節によって指定されていても、それが **MAP** 節によってマップされているなら、それ以降そのスキーマが無視されることはありません。

**DEFAULT** *schema-sqlid*

このオプションは、**USING XDS** パラメーターを指定した場合にのみ使用できます。**DEFAULT** 節で指定されたスキーマは、ロード対象 XML 文書の XML Data Specifier (XDS) に XML スキーマを指定する SCH 属性が含まれていない場合に、妥当性検査のために使用するスキーマとなります。

**DEFAULT** 節は、**IGNORE** 節および **MAP** 節よりも優先されます。XDS が **DEFAULT** 節を満たすなら、**IGNORE** と **MAP** の指定は無視されます。

**MAP** *schema-sqlid*

このオプションは、**USING XDS** パラメーターを指定した場合にのみ使用できます。**MAP** 節は、ロードする各 XML 文書について XML Data Specifier (XDS) の SCH 属性によって指定されるスキーマの代わりに使用する代替スキーマを指定するのに使用します。**MAP** 節には、それぞれがあるスキーマから別のスキーマへのマッピングを表すスキーマ・ペアを 1 つ以上列挙したリストを指定します。ペア中の最初のスキーマは、XDS 内の SCH 属性によって参照されるスキーマを表します。ペア中の 2 番目のスキーマは、スキーマ妥当性検査の実行で使用する必要のあるスキーマを表します。

あるスキーマが **MAP** 節のスキーマ・ペアの左辺で指定されている場合、**IGNORE** 節でさらにそれを指定することはできません。



スキーマ・ペアのマッピングが適用されたなら、その結果は最終的なものです。マッピング操作は推移的ではないため、選択されたスキーマが、それ以降に別のスキーマ・ペアのマッピングに適用されることはありません。

スキーマを複数回マップすることはできません。つまり、複数のペアの左辺に指定することはできません。

#### **SAVECOUNT** *n*

ロード・ユーティリティーが *n* 行ごとに整合点を取ることを指定します。この値はページ・カウントに変換され、エクステント・サイズのインターバルに切り上げられます。メッセージは整合点ごとに発行されるので、**LOAD QUERY** を使用してロード操作をモニターする場合には、このオプションを選択する必要があります。*n* の値が十分な大きさをでない場合、各整合点で実行される活動の同期化によってパフォーマンスに影響してしまいます。

デフォルト値はゼロですが、それは、必要がなければ整合点は確立されないことを意味します。

**CURSOR** ファイル・タイプと併せて指定された場合、または XML 列を含む表をロードする場合、このオプションは使用できません。

#### **ROWCOUNT** *n*

ロードするファイル内の物理レコードの数 *n* を指定します。ユーザーはファイル内の最初の *n* 個の行だけをロードできます。

#### **WARNINGCOUNT** *n*

*n* 個の警告後に、ロード操作を停止します。このパラメーターは、警告は予期されないが、正しいファイルと表が使用されていることを確認するのが望ましい場合に設定してください。ロード・ファイルまたはターゲット表が不適切に指定されると、ロード対象の行ごとにロード・ユーティリティーによって警告が生成され、このためにロードが失敗する可能性があります。*n* がゼロの場合、またはこのオプションが指定されていない場合、何度警告が出されてもロード操作は続行します。警告のしきい値に達したためにロード操作が停止された場合でも、あらためて **RESTART** モードでロード操作を開始できます。ロード操作は、最後の整合点から自動的に続行します。または、入力ファイルの先頭から **REPLACE** モードであらためてロード操作を開始できます。

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

ロード操作中に生じ得る警告およびエラー・メッセージの宛先を指定します。メッセージ・ファイルを指定しなかった場合、インスタンスがパーティション・データベース環境として構成されていない限り、メッセージは標準出力に書き込まれます。このファイルへの完全パスが指定されていない場合、ロード・ユーティリティーは現行のディレクトリーおよびデフォルトのドライブを宛先として使用します。既に存在するファイル名を指定すると、ロード時に情報が追加されます。

通常、メッセージ・ファイルには、ロード操作の終了時にメッセージが入られますが、それ自体は操作の進行状況のモニターには適していません。

#### **TEMPFILES PATH** *temp-pathname*

ロード操作時に一時ファイルを作成する場合に使用するパスの名前を指定します。これはサーバー・データベース・パーティションに従って完全に修飾しなければなりません。

一時ファイルは、ファイル・システムのスペースを使用します。場合によっては、このスペースが相当必要になります。以下のリストに示すのは、すべての一時ファイルにどの程度のファイル・システム・スペースを割り振るべきかの見積もりです。

- ロード・ユーティリティーが生成するメッセージごとに 136 バイト
- データ・ファイルに長フィールド・データまたは LOB が入っている場合は、15 KB のオーバーヘッド。 **INSERT** オプションを指定した場合で、表の中に多量の長フィールドまたは LOB データが既にある場合には、この数値はこれよりもかなり大きくなる場合があります。

**INSERT** ロード・ユーティリティーを実行できる 4 つのモードのうちの 1 つ。既存の表データを変更することなく、ロードされたデータを表に追加します。

#### REPLACE

ロード・ユーティリティーを実行できる 4 つのモードのうちの 1 つ。表から既存データをすべて削除し、ロードされたデータを挿入します。表定義および索引定義は変更されません。階層間でデータを移動する際にこのオプションを使用する場合は、階層全体に関係したデータだけが置き換えられます。副表は置き換えられません。

このオプションは、システム期間テンポラル表へのデータのロードには使用できません。

#### KEEPDICTIONARY

**LOAD REPLACE** 操作の後も、既存のコンプレッション・ディクショナリーを保持します。表の **COMPRESS** 属性が **YES** の場合、新規に置換されるデータは、ロードの呼び出し前に存在していたディクショナリーを使用して圧縮されるという影響を受けます。ディクショナリーが表の中に以前に存在していない場合、表の **COMPRESS** 属性が **YES** である限り、置換されて表に入れられるデータを使用して新規ディクショナリーが作成されます。この場合のコンプレッション・ディクショナリーを作成するために必要なデータ量は、ADC のポリシーの影響を受けます。このデータは圧縮解除された状態で表に取り込まれます。ディクショナリーが表にいったん挿入されると、ロード対象の残りのデータは、このディクショナリーを使用して圧縮されるという影響を受けます。これはデフォルト・パラメーターです。要約を表 1 に示します。

表 28. **LOAD REPLACE KEEPDICTIONARY**

圧縮	表の行データ・ディクショナリーが存在する	XML ストレージ・オブジェクト・ディクショナリーが存在する <sup>1</sup>	コンプレッション・ディクショナリー	データ圧縮
YES	YES	YES	表の行データ・ディクショナリーおよび XML ディクショナリーを保存します。	ロードされるデータは圧縮の対象になります。

## LOAD

表 28. LOAD REPLACE KEEPDICTIONARY (続き)

圧縮	表の行データ・ディクショナリーが存在する	XML ストレージ・オブジェクト・ディクショナリーが存在する <sup>1</sup>	コンプレッション・ディクショナリー	データ圧縮
YES	YES	NO	表の行データ・ディクショナリーを保存し、新規 XML ディクショナリーを作成します。	ロードされる表の行データは圧縮の対象になります。XML ディクショナリーが作成された後に、ロードされる残りの XML データは圧縮の対象になります。
YES	NO	YES	表の行データ・ディクショナリーを作成し、XML ディクショナリーを保存します。	表の行データ・ディクショナリーが作成された後に、ロードされる残りの表の行データは圧縮の対象になります。ロードされる XML データは圧縮の対象になります。
YES	NO	NO	新規の表の行データおよび XML ディクショナリーを作成します。	ディクショナリーが作成された後に、ロードされる残りのデータは圧縮の対象になります。
NO	YES	YES	表の行データ・ディクショナリーおよび XML ディクショナリーを保存します。	ロードされるデータは、圧縮されません。
NO	YES	NO	表の行データ・ディクショナリーを保存します。	ロードされるデータは、圧縮されません。
NO	NO	YES	表の行ディクショナリーには影響はありません。XML ディクショナリーを保存します。	ロードされるデータは、圧縮されません。
NO	NO	NO	影響なし。	ロードされるデータは、圧縮されません。

### 注:

1. コンプレッション・ディクショナリーは、XML 列が DB2 パージョン 9.7 以降の表に追加された場合または表が Online Table Move を使用してマイグレーションされた場合にのみ、表の XML ストレージ・オブジェクトに対して作成できます。
2. **LOAD REPLACE KEEPDICTIONARY** 操作が中断された場合、**LOAD RESTART** または **LOAD TERMINATE** のどちらかを実行すると、ロード・ユーティリティをリカバリーすることができます。既存の XML ストレージ・オブジェクト・ディクショナリーは、中断された **LOAD REPLACE KEEPDICTIONARY** 操作をリカバリーした後は、保持されていない可能性があります。**LOAD RESTART** を使用する場合、新しい XML ストレージ・オブジェクト・ディクショナリーが作成されます。



## RESETDICTIONARY

このディレクティブは、**LOAD REPLACE** 処理に、表の **COMPRESS** 属性が **YES** の場合には表データ・オブジェクトの新規ディクショナリーを作成するように指示します。 **COMPRESS** 属性が **NO** で、ディクショナリーがすでに表にある場合、それは除去され、新規ディクショナリーは表に挿入されません。コンプレッション・ディクショナリーは 1 つのユーザー・レコードのみを使用して作成できます。ロードするデータ・セットのサイズがゼロの場合は、既存のディクショナリーが存在していても、そのディクショナリーは保持されません。このディレクティブを使用してディクショナリーを作成するために必要なデータ量は、ADC のポリシーの影響を受けません。要約については、表 2 を参照してください。

表 29. LOAD REPLACE RESETDICTIONARY

圧縮	表の行データ・ディクショナリーが存在する	XML ストレージ・オブジェクト・ディクショナリーが存在する <sup>1</sup>	コンプレッション・ディクショナリー	データ圧縮
YES	YES	YES	新規ディクショナリーの作成 <sup>2</sup> 。DATA CAPTURE CHANGES オプションが CREATE TABLE または ALTER TABLE ステートメントで有効になっていると、現在の表の行データ・ディクショナリーが保持されます (これは履歴コンプレッション・ディクショナリーと呼ばれます)。	ディクショナリーが作成された後に、ロードされる残りのデータは圧縮の対象になります。
YES	YES	NO	新規ディクショナリーの作成 <sup>2</sup> 。DATA CAPTURE CHANGES オプションが CREATE TABLE または ALTER TABLE ステートメントで有効になっていると、現在の表の行データ・ディクショナリーが保持されます (これは履歴コンプレッション・ディクショナリーと呼ばれます)。	ディクショナリーが作成された後に、ロードされる残りのデータは圧縮の対象になります。
YES	NO	YES	新規ディクショナリーの作成。	ディクショナリーが作成された後に、ロードされる残りのデータは圧縮の対象になります。

## LOAD

表 29. LOAD REPLACE RESETDICTIONARY (続き)

圧縮	表の行データ・ディクショナリーが存在する	XML ストレージ・オブジェクト・ディクショナリーが存在する <sup>1</sup>	コンプレッション・ディクショナリー	データ圧縮
YES	NO	NO	新規ディクショナリーの作成。	ディクショナリーが作成された後に、ロードされる残りのデータは圧縮の対象になります。
NO	YES	YES	ディクショナリーを除去します。	ロードされるデータは、圧縮されません。
NO	YES	NO	表の行データ・ディクショナリーを除去します。	ロードされるデータは、圧縮されません。
NO	NO	YES	XML ストレージ・オブジェクト・ディクショナリーを除去します。	ロードされるデータは、圧縮されません。
NO	NO	NO	影響なし。	すべての表データが圧縮されるわけではありません。

### 注:

1. コンプレッション・ディクショナリーは、XML 列が DB2 パージョン 9.7 以降の表に追加された場合または表が Online Table Move を使用してマイグレーションされた場合にのみ、表の XML ストレージ・オブジェクトに対して作成できます。
2. ディクショナリーが存在していて圧縮属性が有効であるものの、表パーティションにロードするレコードがない場合、新規ディクショナリーを作成することはできず、**RESETDICTIONARY** 操作では既存のディクショナリーは保持されません。

### TERMINATE

ロード・ユーティリティを実行できる 4 つのモードのうちの 1 つ。以前に割り込みを受けたロード操作を終了し、ロード操作が開始された時点まで操作をロールバックします。途中で整合点があっても通過します。その操作に関係する表スペースの状態は通常に戻され、すべての表オブジェクトの整合性が保たれます (索引オブジェクトが無効とマークされる場合がありますが、そのような場合には、次のアクセス時に索引の再作成が自動的に行われます)。終了するロード操作が **LOAD REPLACE** の場合、その表は **LOAD TERMINATE** 操作完了後に空の表まで切り捨てられます。終了するロード操作が **LOAD INSERT** の場合、その表は **LOAD TERMINATE** 操作完了後も元のレコードをすべて保持します。ディクショナリー管理の要約を表 3 に示します。

**LOAD TERMINATE** オプションでは、表スペースの BACKUP PENDING 状態は解除されません。

### RESTART

ロード・ユーティリティを実行できる 4 つのモードのうちの 1 つ。以前に割り込みを受けたロード操作を再開します。ロード操作は、ロード、作

成、または削除フェーズの最後の整合点から自動的に続行されます。ディクショナリー管理の要約を表 4 に示します。

#### INTO *table-name*

データのロード先となるデータベース表を指定します。この表として、システム表、宣言済み一時表、または作成済み一時表は指定できません。別名、完全修飾、または非修飾の表名を指定できます。修飾された表名は、*schema.tablename* の形式になります。非修飾の表名を指定すると、その表は CURRENT SCHEMA で修飾されます。

暗黙的な非表示列がデータベース表に含まれる場合には、ロード操作に非表示列のデータを含めるかどうかを指定する必要があります。非表示列のデータを含めるかどうかを示すには、以下のいずれかの方法を使用します。

- データの挿入先の列を明示的に指定する場合、*insert-column* を使用します。

```
db2 load from delfile1 of del
insert into table1 (c1, c2, c3,...)
```

- 非表示列のファイル・タイプ修飾子のいずれかを使用します。入力ファイルに非表示列のデータが含まれる場合には **implicitlyhiddeninclude** を指定し、入力ファイルに含まれない場合には **implicitlyhiddenmissing** を指定します。

```
db2 load from delfile1 of del modified by implicitlyhiddeninclude
insert into table1
```

- 暗黙的な非表示列が含まれる表をデータ移動ユーティリティーが検出した場合のデフォルト動作を設定するには、DB2\_DMU\_DEFAULT レジストリー変数をサーバー・サイドで使用します。暗黙的な非表示列が含まれるとユーティリティーが想定するようにするには **IMPLICITLYHIDDENINCLUDE** を、暗黙的な非表示列は含まれないとユーティリティーが想定するようにするには **IMPLICITLYHIDDENMISSING** を指定します。

```
db2set DB2_DMU_DEFAULT=IMPLICITLYHIDDENINCLUDE
db2 load from delfile1 of del insert into table1
```

#### *insert-column*

データの挿入先となる表の列を指定します。

ロード・ユーティリティーは、1 つ以上のスペースを使った名前前の列を解析できません。例えば、次のようになります。

```
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)
```

は、Int 4 列があるためエラーになります。これは、次のようにして二重引用符で列名を囲むことによって解決できます。

```
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*

エラーが発生した行のコピー先となる例外表を指定します。ユニーク索引または主キー索引に違反した行がすべてコピーされます。非修飾の表名を指定すると、その表は CURRENT SCHEMA で修飾されます。

## LOAD

例外表に書き込まれる情報は、ダンプ・ファイルには書き込まれません。パーティション・データベース環境では、ロードする表を定義されたデータベース・パーティションの例外表を定義する必要があります。一方、ダンプ・ファイルには、無効であるか構文エラーであるためにロードできない行が入ります。

XML データのロード時に、**FOR EXCEPTION** 節を使用したロード例外表の指定は、以下の状況ではサポートされません。

- ラベル・ベースのアクセス制御 (LBAC) を使用する場合。
- パーティション表にデータをロードする場合。

### NORANGEEXC

範囲違反のためにリジェクトされた行は、例外表に挿入しないことを指定します。

### NOUNIQUEEXC

ユニーク制約に違反しているためにリジェクトされた行は、例外表に挿入しないことを指定します。

### STATISTICS USE PROFILE

この表で定義されているプロファイルに従ってロード中に統計を収集するようロード操作に指示します。そのプロファイルは、ロードの実行前に作成されていなければなりません。そのプロファイルは、**RUNSTATS** コマンドで作成します。プロファイルが存在しない場合に、プロファイルに従って統計を収集するようロード操作に指示すると、警告メッセージが戻されて統計は収集されません。

ロードの際に、分散統計はタイプ XML の列については収集されません。

### STATISTICS NO

統計データを収集せず、したがってカタログ内の統計データも変更しないことを指定します。これはデフォルトです。

### COPY NO

順方向リカバリーが有効である場合 (つまり、**logarchmeth1** または **logarchmeth2** に OFF 以外の値が設定されている場合)、表が存在している表スペースをバックアップ・ペンディング状態にするよう指定します。**COPY NO** オプションを使用する場合も、表スペース状態は **LOAD IN PROGRESS** になります。これは、一時的な状態であり、ロードが完了するか打ち切られると解除されます。表スペースのバックアップまたはデータベースのフルバックアップを実行しない限り、表スペースのどの表のデータも更新または削除できません。ただし、**SELECT** ステートメントを使用すれば、どの表のデータにもアクセス可能です。

リカバリー可能データベースでの **COPY NO** を指定した **LOAD** は、表スペースを **BACKUP PENDING** 状態のままにします。例えば、**COPY NO** を指定した **LOAD** および **INDEXING MODE DEFERRED** を実行すると、索引はリフレッシュが必要な状態になります。表での照会には、索引スキャンが必要なものがあり、索引がリフレッシュされるまで、成功しません。索引が、バックアップ・ペンディング状態にある表スペース内にある場合、その索引のリフレッシュはできません。この場合、表へのアクセスは、バックアップが行われるまで許可されません。索引リフレッシュは、索引が照会によってアクセスされたときに、データベースによって自動的に行われます。**COPY NO**、**COPY**

**YES、NONRECOVERABLE** のいずれも指定されておらず、データベースがリカバリー可能である場合 (**logarchmeth1** または **logarchmeth2** に OFF 以外の値が設定されている場合)、**COPY NO** がデフォルトです。

#### **COPY YES**

ロードするデータのコピーを保存することを指定します。フォワード・リカバリーが使用不可になっている場合、このオプションは無効です。

#### **USE TSM**

Tivoli Storage Manager (TSM) を使ってコピーを保管することを指定します。

#### **OPEN num-sess SESSIONS**

TSM またはベンダー製品とともに使用する入出力セッションの数です。デフォルト値は 1 です。

#### **TO device/directory**

コピー・イメージを作成する先の装置またはディレクトリーを指定します。

#### **LOAD lib-name**

使用するバックアップおよびリストア I/O ベンダー関数を含む共有ライブラリー (Windows オペレーティング・システムでは DLL) の名前。絶対パスで指定することができます。絶対パスを指定しない場合、デフォルトでユーザー出口プログラムの存在するパスになります。

#### **NONRECOVERABLE**

ロード・トランザクションがリカバリー不能としてマークされており、それ以降のロールフォワード・アクションによってそれをリカバリーさせることは不可能であることを指定します。ロールフォワード・ユーティリティーは、そのトランザクションをスキップし、データのロード先の表に「無効」としてマークします。さらに、ユーティリティーは、その表に対する後続のすべてのトランザクションを無視します。ロールフォワード操作が完了すると、そのような表は、ドロップするか、またはリカバリー不能なロード操作完了後のコミット・ポイントの後に取られたバックアップ (フルバックアップまたは表スペースのバックアップ) からリストアすることのみ可能です。

このオプションを使用すると、表スペースはロード操作後に **BACKUP PENDING** 状態になりません。また、ロード操作中にロードされたデータのコピーが作成される必要もなくなります。 **COPY NO、COPY YES、NONRECOVERABLE** のいずれも指定されておらず、データベースがリカバリー可能ではない場合 (**logarchmeth1** および **logarchmeth2** の両方に OFF が設定されている場合)、**NONRECOVERABLE** がデフォルトです。

#### **WITHOUT PROMPTING**

データ・ファイルのリストにロードするすべてのファイルを含め、しかもリストに入っている装置またはディレクトリーがロード操作全体で十分であるということを指定します。続きの入力ファイルが見つからなかったり、ロード操作が終了する前にコピー先がいっぱいになるとロード操作は失敗し、表は **LOAD PENDING** 状態のままになります。

このオプションを指定しない場合に、テープ装置がコピー・イメージ用のテープの終わりに達した場合、またはリスト中の最後の項目がテープ装置であった場合は、ユーザーに対してその装置に新しいテープを装着するよう求めるプロンプトが出されます。

#### DATA BUFFER *buffer-size*

ユーティリティ内でデータを転送するためのバッファ・スペースとして使用する 4 KB ページの数を指定します (並列処理の度合いには依存しません)。指定する値がアルゴリズム上の最小値より小さい場合、最小限必要なりソースが使用され、警告は戻されません。

このメモリーは、ユーティリティ・ヒープから直接に割り当てられ、そのサイズは **util\_heap\_sz** データベース構成パラメーターで修正可能です。バージョン 9.5 以降では、システムにさらに使用可能なメモリーがある場合、**LOAD** コマンドの DATA BUFFER オプションの値は、一時的に **util\_heap\_sz** を超えることができます。この場合、ユーティリティ・ヒープは、必要に応じて **database\_memory** 限度に達するまで動的に増加します。このメモリーは、ロード操作が完了すると解放されます。

値が指定されていない場合、実行時にユーティリティによって適切なデフォルトが計算されます。デフォルトは、ローダーのインスタンス生成時にユーティリティ・ヒープで使用できるフリー・スペースの割合と、表の一部の特性に基づいて決まります。

#### SORT BUFFER *buffer-size*

このオプションは、ロード操作時に **sortheap** データベース構成パラメーターをオーバーライドする値を指定します。これは、索引とともに表をロードする場合、また **INDEXING MODE** パラメーターが DEFERRED として指定されていない場合にのみ関係があります。指定された値は **sortheap** の値を超えることはできません。このパラメーターは、**sortheap** の値を変更せずに多くの索引を持つ表をロードする際に使用される、ソート・メモリーのスロットルで役に立ちます。これは、一般的な照会処理にも影響を与えます。

#### CPU\_PARALLELISM *n*

表オブジェクトの作成時に、レコードの解析、変換、およびフォーマット設定のためにロード・ユーティリティによって作成されるプロセスまたはスレッドの数を指定します。このパラメーターは、データベース・パーティションごとに実行するプロセスの数を活用するために設計されています。これは、事前にソートされたデータをロードする際に役立ちます (ソース・データのレコード順序が保持されるため)。このパラメーターの値が 0 の場合や、このパラメーターを指定しなかった場合、ロード・ユーティリティは、実行時に自動的に計算された適切なデフォルト値 (通常は使用できる CPU の数に基づく) を使用します。

#### 注:

1. LOB または LONG VARCHAR フィールドのどちらかの入った表でこのパラメーターを使用する場合、システムの CPU の数またはユーザーが指定した値には関係なく、値は 1 になります。
2. **SAVECOUNT** パラメーターに指定する値が小さいと、データと表のメタデータの両方をフラッシュするために、ローダーがさらに多くの入出力操作を実行することになります。 **CPU\_PARALLELISM** が 1 より大きな



ら、フラッシュ操作は非同期になり、ローダーは CPU を活用できません。 **CPU\_PARALLELISM** が 1 に設定されている場合、ローダーは整合点において入出力を待ちます。 **CPU\_PARALLELISM** を 2 に設定し、 **SAVECOUNT** を 10 000 に設定したロード操作は、 CPU が 1 つしかなくても、同じ操作で **CPU\_PARALLELISM** を 1 に設定した場合より速く完了します。

#### **DISK\_PARALLELISM *n***

表スペース・コンテナにデータを書き込むためにロード・ユーティリティーが作成するプロセスまたはスレッドの数を指定します。値を指定しない場合、ユーティリティーは表スペース・コンテナの数と表の特性に基づいて、自動的に計算された適切なデフォルトを選択します。

#### **INDEXING MODE**

ロード・ユーティリティーが索引を再作成するのか、それとも索引を増分で拡張するのかを指定します。有効な値は以下のとおりです。

##### **AUTOSELECT**

**REBUILD** モードと **INCREMENTAL** モードのいずれにするかを、ロード・ユーティリティーが自動的に決定します。決定は、ロードされるデータ量と索引ツリーの深さに基づいて行われます。索引ツリーの深さに関連する情報は索引オブジェクトに保管されています。この情報を設定するために、 **RUNSTATS** は不要です。 **AUTOSELECT** がデフォルトの索引付けモードです。

##### **REBUILD**

すべての索引が再作成されます。古い表データの索引キー部分も、追加される新しい表データの索引キー部分もすべてソートできるようにするため、ロード・ユーティリティーには十分なりソースが必要となります。

**LogIndexBuild** データベース構成パラメーターがオンになっている場合、トランザクション・ログには、各索引ページの作成後にそのページのイメージが含まれます。 **LogIndexBuild** データベース構成パラメーターがオフになっている場合、索引マネージャーがログに記録するのは、各ページの割り振りと初期化のみです (各ページの空ではない部分とは異なり、ページ当たり約 250 バイト)。

##### **INCREMENTAL**

索引に新しいデータが取り込まれて拡張します。このアプローチでは、索引のフリー・スペースが消費されます。このアプローチでは、新たに挿入されるレコードの索引キーを追加するためのソート・スペースだけがあれば十分です。この方式がサポートされるのは、索引オブジェクトが有効で、かつロード操作の開始時にアクセス可能な場合だけです (例えば、 **DEFERRED** モードが指定されたロード操作の直後では、この方式は無効です)。このモードを指定したものの、索引の状態などの理由でサポートされない場合は、警告が戻され、 **REBUILD** モードでロード操作が続行されます。同様に、ロード作成フェーズでロード再開操作を開始した場合も、 **INCREMENTAL** モードはサポートされません。

**LogIndexBuild** データベース構成パラメーターがオンになっている場合、DB2 ソフトウェアは、実行されたページ分割のほか、索引へ

の各キーの挿入に関するログ・レコードを生成します。このパラメーターがオフになっている場合 (HADR を使用していない場合に一般的)、索引マネージャーによって実行される索引ロギングの量は、ALLOW READ ACCESS オプションが指定されたかどうかに応じて異なります。ALLOW READ ACCESS オプションが指定されている場合、ページ分割に関するログを含んだログ・レコードが生成されます。ALLOW READ ACCESS オプションが指定されていない場合、索引マネージャーからのログ・レコードは生成されません。

#### DEFERRED

このモードが指定されている場合、ロード・ユーティリティーは索引の作成を試みません。リフレッシュが必要であることを示すマークが索引に付けられます。ロード操作とは関係のないこのような索引に最初にアクセスするときは、再作成が強制的に実行されたり、データベースの再始動時に索引が再作成されたりする場合があります。このアプローチでは、最も大きい索引のキー部分をすべて処理できるだけのソート・スペースが必要です。索引を作成するためにその後かかる合計時間は、REBUILD モードの場合よりも長くなります。したがって、この索引作成据え置きモードで複数のロード操作を実行する場合、最初の非ロード・アクセス時に索引を再作成できるようにしておくよりも、順序列内の最後のロード操作で索引の再作成を実行できるようにした方が (パフォーマンスの観点から) 賢明であるといえます。

据え置き索引作成がサポートされるのは、非ユニーク索引がある表だけです。そのため、ロード・フェーズで挿入される重複キーがロード操作後は永続的ではなくなります。

#### ALLOW NO ACCESS

ロードを使用すると、ロード中に、排他的アクセスのためにターゲット表がロックされます。ロード中、表の状態は LOAD IN PROGRESS に設定されます。ALLOW NO ACCESS はデフォルトの動作です。これは、LOAD REPLACE で唯一有効なオプションです。

表に制約があると、表の状態は、ロード中の他に、SET INTEGRITY ペンディングに設定されます。表の SET INTEGRITY ペンディング状態を解除するには、SET INTEGRITY ステートメントを使用する必要があります。

#### ALLOW READ ACCESS

ロードを使用すると、ターゲット表は共有モードでロックされます。表の状態は、LOAD IN PROGRESS および READ ACCESS の両方に設定されます。表のロード中、データの非デルタ部分にアクセスすることができます。つまり、表を読み取る側はロードの開始前に存在していたデータにはアクセスができ、ロード中のデータはロードが完了するまで利用できない、ということです。

**重要:** バージョン 10.1 フィックスパック 1 以降、ALLOW READ ACCESS パラメーターは非推奨となっており、将来のリリースで除去される可能性があります。詳しくは、「DB2 バージョン 10.1 の新機能」の『LOAD コマンドの ALLOW READ ACCESS パラメーターが非推奨になった』を参照してください。



**ALLOW READ ACCESS** ロードの **LOAD TERMINATE** または **LOAD RESTART** はこのパラメーターを使用できますが、**ALLOW NO ACCESS** ロードの **LOAD TERMINATE** または **LOAD RESTART** はこのパラメーターを使用できません。また、ターゲット表上の索引が要再作成のマークが付けられると、このオプションは無効になります。

表に制約があると、表の状態は、LOAD IN PROGRESS、および READ ACCESS の他に、SET INTEGRITY PENDING に設定されます。ロードの終了時に、表の状態 LOAD IN PROGRESS は解除されますが、SET INTEGRITY PENDING および READ ACCESS はそのまま残ります。表の SET INTEGRITY PENDING を解除するには、SET INTEGRITY ステートメントを使用する必要があります。表が SET INTEGRITY PENDING および READ ACCESS の状態にある間、データの非デルタ部分には引き続き読み取りアクセスできますが、データの新しい (デルタ) 部分には、SET INTEGRITY ステートメントが完了するまでアクセス不能のままになります。ユーザーは、SET INTEGRITY ステートメントを発行しないで、同じ表上で複数のロードを実行できます。ただし、元の (チェック済み) データは、SET INTEGRITY ステートメントが発行されるまで可視のままです。

**ALLOW READ ACCESS** は、以下の修飾子もサポートします。

#### USE *tablespace-name*

索引が再作成される場合、表スペース *tablespace-name* に索引のシャドー・コピーが作成され、ロード終了時の INDEX COPY PHASE で、元の表スペース上にコピーされます。SYSTEM TEMPORARY 表スペースのみ、このオプションを使用できます。指定されない場合、シャドー索引が、索引オブジェクトと同じ表スペース内に作成されます。シャドー・コピーが索引オブジェクトと同じ表スペース内に作成される場合、古い索引オブジェクトを介したシャドー索引オブジェクトのコピーは瞬時に終了します。シャドー・コピーが索引オブジェクトとは異なる表スペースにある場合、物理コピーが実行されます。これにはかなりの入出力および時間を要します。コピーは、表がオフラインの間、ロード終了時の INDEX COPY PHASE で行われます。

このオプションを使用しないと、シャドー索引は元の索引と同じ表スペースに作成されます。デフォルトでは、元の索引とシャドー索引の両方が同時に同じ表スペースに常駐するため、1つの表スペース内に両方の索引を保持するためのスペースが不足する場合があります。このオプションを使用すれば、索引用の十分な表スペースを保持できます。

ユーザーが **INDEXING MODE REBUILD** または **INDEXING MODE AUTOSELECT** を指定しない場合、このオプションは無視されます。このオプションは **INDEXING MODE AUTOSELECT** が選択され、ロードが索引を徐々に更新することを選択した場合にも無視されます。

#### FETCH\_PARALLELISM YES | NO

**DATABASE** キーワードを使用してカーソルが宣言されていてカーソルからのロードを実行するとき、または API の `sqlu_remotefetch_entry` メディア項目を使用するとき、このオプションが YES に設定されていると、ロード・ユーティリティは、リモート・データ・ソースからのフェッチの並列化を試

みます (可能な場合)。NO に設定されている場合、並列フェッチは行われません。デフォルト値は、YES です。詳しくは、『CURSOR ファイル・タイプを使用したデータの移動』を参照してください。

#### SET INTEGRITY PENDING CASCADE

**LOAD** によって表が SET INTEGRITY ペンディング状態になる場合、**SET INTEGRITY PENDING CASCADE** オプションを使用することによって、ユーザーはロードされる表の SET INTEGRITY ペンディング状態を即時にすべての下層 (下層外部キー表、下層即時マテリアライズ照会表、および下層即時ステージング表を含む) にカスケードするかどうか指定することができます。

#### IMMEDIATE

外部キー制約の SET INTEGRITY PENDING 状態が即時にすべての下層外部キー表、下層即時マテリアライズ照会表、および下層ステージング表に拡張されることを示します。**LOAD INSERT** 操作の場合、**IMMEDIATE** オプションが指定されている場合でも、SET INTEGRITY PENDING 状態は下層外部キー表に拡張されません。

後で (SET INTEGRITY ステートメントの IMMEDIATE CHECKED オプションを使用して) ロードされる表の制約違反をチェックする際、SET INTEGRITY PENDING READ ACCESS 状態だった下層外部キー表は、SET INTEGRITY PENDING NO ACCESS 状態になります。

#### DEFERRED

ロードされる表だけが、SET INTEGRITY PENDING 状態になることを示します。下層外部キー表、下層即時マテリアライズ照会表、および下層即時ステージング表は、未変更のままになります。

下層外部キー表は、(SET INTEGRITY ステートメントの IMMEDIATE CHECKED オプションを使用して) その親表の制約違反がチェックされる時、後で暗黙的に SET INTEGRITY PENDING 状態になる場合があります。下層即時マテリアライズ照会表および下層即時ステージング表は、その基礎表のいずれかの健全性違反がチェックされる際、暗黙的に SET INTEGRITY PENDING 状態になります。照会のアクセス先が、指定された表ではなく、SET INTEGRITY PENDING 状態にない適格なマテリアライズ照会表である場合は、SET INTEGRITY PENDING 状態の表の照会が成功することがあります。下層表が SET INTEGRITY ペンディング状態になったことを示す警告 (SQLSTATE 01586) が出されます。この下層表がいつ SET INTEGRITY PENDING 状態になるかについては、SQL リファレンスにある SET INTEGRITY ステートメントの「注」の項を参照してください。

**SET INTEGRITY PENDING CASCADE** オプションが指定されない場合、次のようになります。

- ロードされる表だけが、SET INTEGRITY PENDING 状態になります。下層外部キー表、下層即時マテリアライズ照会表、および下層即時ステージング表の状態は未変更のままになり、後にロードされた表の制約違反がチェックされる際に、暗黙的に SET INTEGRITY PENDING 状態になる場合があります。

LOAD によってターゲット表が SET INTEGRITY PENDING 状態にならない場合、SET INTEGRITY PENDING CASCADE オプションは無視されます。

#### LOCK WITH FORCE

ユーティリティーはロード・プロセス中に、表ロックなどの様々なロックを獲得します。ロックを獲得する際、このオプションを使用すると、ロードは待機することなく、またタイムアウトになることなく、ターゲット表に競合するロックを持つ他のアプリケーションを強制的にオフにします。システム・カタログ表に対する競合するロックを保持するアプリケーションは、ロード・ユーティリティーによって強制的にオフにされることはありません。強制されたアプリケーションは、ロールバックし、ロード・ユーティリティーが必要とするロックをリリースします。その後、ロード・ユーティリティーを続行できます。このオプションは、**FORCE APPLICATIONS** コマンドと同じ権限 (SYSADM または SYSCTRL) を必要とします。

**ALLOW NO ACCESS** ロードは、ロード操作の開始時に競合するロックを持つアプリケーションを強制的にロールバックさせる場合があります。ロードの開始時に、ユーティリティーは、表の照会または変更を試みているアプリケーションを強制的にロールバックさせる場合があります。

**ALLOW READ ACCESS** ロードは、ロード操作の開始時および終了時に競合するロックを持つアプリケーションを強制的にロールバックさせる場合があります。ロードの開始時に、ロード・ユーティリティーは、表の変更を試みているアプリケーションを強制的にロールバックさせる場合があります。ロード操作の終了時に、ロード・ユーティリティーは、表の照会または変更を試みているアプリケーションを強制的にロールバックさせる場合があります。

#### SOURCEUSEREXIT *executable*

このユーティリティーにデータを送るために呼び出される実行可能ファイル名を指定します。

#### REDIRECT

##### INPUT FROM

###### **BUFFER** *input-buffer*

*input-buffer* で指定されたバイトのストリームが、所定の実行可能ファイルを実行するプロセスの STDIN ファイル記述子に渡されます。

###### **FILE** *input-file*

このクライアント・サイドのファイルの内容が、所定の実行可能ファイルを実行するプロセスの STDIN ファイル記述子に渡されます。

##### OUTPUT TO

###### **FILE** *output-file*

STDOUT および STDERR ファイル記述子が、指定した完全修飾されたサーバー・サイドのファイルに取り込まれます。

#### PARALLELIZE

複数のユーザー出力プロセスを同時に呼び出すことによって、ロード・ユーティリティーへのデータ入力のスループットを高めめます。

## LOAD

このオプションは、複数パーティション・データベース環境でのみ適用でき、単一パーティション・データベース環境では無視されません。

詳細については、『カスタマイズしたアプリケーション (ユーザー出口) を使用したデータの移動』を参照してください。

### **PARTITIONED DB CONFIG** *partitioned-db-option*

複数のデータベース・パーティションに分散した表へのロードの実行を可能にします。 **PARTITIONED DB CONFIG** パラメーターを使用すると、パーティション・データベース固有の構成オプションを指定することができます。

*partitioned-db-option* の値は、以下のオプションのいずれかになります。

```
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
```

これらのオプションの詳細な説明は、『パーティション・データベース環境のロード構成オプション』で扱われています。

### **RESTARTCOUNT**

非推奨。

### **USING** *directory*

非推奨。

## 例

### 例 1

TABLE1 に以下の 5 つの列があるとします。

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

ASCFILE1 に以下の 7 つのエレメントがあるとします。

- ELE1、位置 01 から 20
- ELE2、位置 21 から 22
- ELE3、位置 23 から 23
- ELE4、位置 24 から 27

- ELE5、位置 28 から 31
- ELE6、位置 32 から 32
- ELE7、位置 33 から 40

データ・レコードは以下のとおりです。

```
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 WWN6789   Y
```

以下のコマンドは、ファイルから表をロードします。

```
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)
```

注:

1. **MODIFIED BY** パラメーターで **striptblanks** を指定すると、VARCHAR 列の中のブランクが切り捨てられるようになります (例えば行 1、2、および 3 の長さが 11、17、および 19 バイトである COL1)。
2. **MODIFIED BY** パラメーターで **reclen=40** を指定すると、各入力レコードの最後が改行文字でなく、各レコードが 40 バイト長であることを指定することになります。最後の 8 バイトは、表のロードには使用されません。
3. COL4 は入力ファイルにはないので、そのデフォルト値 (NOT NULL WITH DEFAULT と定義されている) を使用して TABLE1 に挿入されます。
4. 位置 23 と 32 は、特定の行で TABLE1 の COL2 と COL3 が NULL としてロードされるかどうかを指示するために使用されます。特定のレコードのうち列の NULL 標識位置が Y なら、その列は NULL になります。それが N の場合は、入力レコードのその列のデータ位置 (L(.....) で定義) にあるデータ値が、その行の列データのソースとして使用されます。この例では、行 1 のどの列も NULL ではなく、行 2 の COL2 は NULL であり、行 3 の COL3 は NULL です。
5. この例では、COL1 と COL5 の NULL INDICATORS は 0 (ゼロ) として指定されますが、それはそのデータが NULL 不可能であることを示しています。
6. 特定の列に対する NULL INDICATOR は入力レコードのどの位置でも可能ですが、その位置は必ず指定しなければならず、Y または N のいずれかの値が提供される必要があります。

## 例 2 (ファイルから LOB をロードする)

TABLE1 に次の 3 つの列があるとします。

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

ASCFILE1 には次の 3 つのエレメントがあるとします。

- ELE1、位置 01 から 04
- ELE2、位置 06 から 13

## LOAD

- ELE3、位置 15 から 22

次に示すファイルは、 /u/user1 または /u/user1/bin のどちらかにあります。

- ASCFILE2 - LOB データを持つ
- ASCFILE3 - LOB データを持つ
- ASCFILE4 - LOB データを持つ
- ASCFILE5 - LOB データを持つ
- ASCFILE6 - LOB データを持つ
- ASCFILE7 - LOB データを持つ

ASCFILE1 内のデータ・レコード

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

以下のコマンドは、ファイルから表をロードします。

```
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
```

注:

1. **MODIFIED BY** パラメーターの中で **lobsinfile** を指定すると、ファイルからすべての LOB データをロードすることをローダーに対して指定することになります。
2. **MODIFIED BY** パラメーターで **reclen=22** を指定すると、各入力レコードの最後が改行文字でなく、各レコードが 22 バイト長であることを指定することになります。
3. LOB データは、ASCFILE2 から ASCFILE7 までの 6 つのファイルに入っています。各ファイルには、特定の行の LOB 列をロードするのに使用されるデータが入れています。LOB と他のデータのリレーションシップは、ASCFILE1 に指定します。このファイルの最初のレコードは、REC1 を行 1 の COL1 にするようローダーに指示します。行 1 の LOB1 をロードするには ASCFILE2 の内容が使われ、ASCFILE3 の内容は行 1 の LOB2 をロードするのに使われます。同じように、行 2 の LOB1 および LOB2 をロードするには ASCFILE4 と ASCFILE5 が使われ、行 3 の LOB をロードするには ASCFILE6 と ASCFILE7 が使われます。
4. これらのファイルがローダーで必要になった場合には、名前の指定された LOB ファイルを探索するのに使われる 2 つのパスが、LOBS FROM パラメーターに入っています。
5. **lobsinfile** 修飾子を指定しないで ASCFILE1 (区切りなしの ASCII ファイル) から直接 LOB をロードする場合は、以下の規則を守ってください。
  - LOB を含めたレコードの全長は 32 KB 以下でなければなりません。
  - 入力レコード内の LOB フィールドは固定長でなければならず、必要なら LOB データにブランクを埋め込まなければなりません。



- LOB をデータベースに挿入する際に、LOB の埋め込みに使われる後続ブランクを除去できるよう、**striptblanks** 修飾子を指定する必要があります。

### 例 3 (ダンプ・ファイルの使用)

表 FRIENDS は、次のように定義されています。

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

この表に対して、以下に示すデータ・レコードのロードを試みたとします。

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

最初の INT が NULL で、列定義に NOT NULL が指定されているため、第 2 行はリジェクトされます。DEL フォーマットと互換でない開始文字の入った列は、エラーを生成し、レコードはリジェクトされます。そのようなレコードは、ダンプ・ファイルに書き込むことができます。

区切り文字の外側にある列の DEL データは無視されますが、警告が生成されません。以下に例を示します。

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

ユーティリティーは、表の第 3 列に "sam" をロードし、警告の中で文字 "sdf" にフラグが付けられます。このレコードはリジェクトされません。別の例を考えましょう。

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

ユーティリティーは 22,34,"bob" をロードし、列 1 の 22 の後にある一部のデータが無視されたという警告を生成します。このレコードはリジェクトされません。

### 例 4 (ID 列がある表へのロード)

TABLE1 には以下の 4 つの列があります。

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

TABLE2 は TABLE1 と同じですが、C2 が GENERATED ALWAYS ID 列である点が異なります。

DATAFILE1 のデータ・レコード (DEL フォーマット):

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

DATAFILE2 のデータ・レコード (DEL フォーマット):

## LOAD

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

### 注:

1. 以下のコマンドは、DATAFILE1 で行 1 および 2 への IDENTITY 値が入力されていないので、それらの行のための IDENTITY 値を生成します。ただし、行 3 にはユーザー提供の ID 値 100 が、行 4 にはユーザー提供の ID 値 101 がそれぞれ割り当てられます。

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

2. DATAFILE1 を TABLE1 にロードしてすべての行に対する ID 値を生成するには、以下のコマンドのいずれかを発行します。

```
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. DATAFILE2 を TABLE1 にロードして各行に対する ID 値を生成するには、以下のコマンドのいずれかを発行します。

```
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. ID 値 100 と 101 を行 3 および 4 に割り当てるために DATAFILE1 を TABLE2 にロードするには、以下のコマンドを発行します。

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

この場合、ユーティリティには、ユーザー提供の値を優先して、システム生成の ID 値に上書きするように指示しているため、行 1 および 2 はリジェクトされます。ユーザー提供の値が存在しない場合でも、ID 列が暗黙的に非 NULL であるため、この行はリジェクトする必要があります。

5. ID に関するファイル・タイプ修飾子を使用せずに DATAFILE1 を TABLE2 にロードすると、行 1 と 2 はロードされますが、行 3 と 4 はリジェクトされます。これは、行 3 と 4 では独自に非 NULL 値が提供されており、ID 列が GENERATED ALWAYS であるためです。

### 例 5 (ROW CHANGE TIMESTAMP 列がある表へのロード)

TABLE1 には以下の 4 つの列があります。

- C1 VARCHAR(30)
- C2 ROW CHANGE TIMESTAMP GENERATED BY DEFAULT
- C3 DECIMAL(7,2)
- C4 CHAR(1)

TABLE2 は TABLE1 と同じですが、C2 が GENERATED ALWAYS 列である点が異なります。

DATAFILE1 のデータ・レコード (DEL フォーマット):



```
"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
```

DATAFILE2 のデータ・レコード (DEL フォーマット):

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

注:

1. 以下のコマンドは、DATAFILE1 で行 1 および 2 への ROW CHANGE TIMESTAMP 値が入力されていないので、それらの行のための ROW CHANGE TIMESTAMP 値を生成します。ただし、行 3 および 4 は、ユーザー提供の ROW CHANGE TIMESTAMP 値 2006-05-23-15.55.53.209971 と 2006-05-22-19.34.14.947681 が割り当てられます。

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

2. DATAFILE1 を TABLE1 にロードしてすべての行に対する ROW CHANGE TIMESTAMP 値を生成するには、以下のコマンドのいずれかを発行します。

```
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. DATAFILE2 を TABLE1 にロードして各行に対する ROW CHANGE TIMESTAMP 値を生成するには、以下のコマンドのいずれかを発行します。

```
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. DATAFILE1 を TABLE2 にロードして、2006-05-23-15.55.53.209971 および 2006-05-22-19.34.14.947681 の ROW CHANGE TIMESTAMP 値を行 3 および 4 に割り当てるには、以下のコマンドを発行します。

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

この場合、ユーティリティには、ユーザー提供の値を優先して、システム生成の ROW CHANGE TIMESTAMP 値に上書きするように指示しているため、行 1 および 2 はリジェクトされます。ユーザー提供の値が存在しない場合でも、ROW CHANGE TIMESTAMP 列が暗黙的に非 NULL であるため、この行はリジェクトする必要があります。

5. ROW CHANGE に関するファイル・タイプ修飾子を使用せずに DATAFILE1 を TABLE2 にロードすると、行 1 と 2 はロードされますが、行 3 と 4 はリジェクトされます。これは、行 3 と 4 では独自に非 NULL 値が提供されており、ROW CHANGE TIMESTAMP 列が GENERATED ALWAYS であるためです。

#### 例 6 (CURSOR ファイル・タイプを使用したロード)

表 ABC.TABLE1 には次の 3 つの列があります。

## LOAD

```
ONE INT
TWO CHAR(10)
THREE DATE
```

表 ABC.TABLE2 には次の 3 つの列があります。

```
ONE VARCHAR
TWO INT
THREE DATE
```

以下のコマンドを実行すると、すべてのデータが ABC.TABLE1 から ABC.TABLE2 にロードされます。

```
db2 declare mycurs cursor for select two,one,three from abc.table1
db2 load from mycurs of cursor insert into abc.table2
```

ABC.TABLE1 が、ABC.TABLE2 のあるデータベースとは異なるデータベースにある場合、DECLARE CURSOR ステートメントの DATABASE、USER、および USING オプションを使用してロードを実行することができます。例えば、ABC.TABLE1 がデータベース DB1 にあり、DB1 のユーザー ID およびパスワードが user1 および pwd1 である場合、以下のコマンドを実行すると、すべてのデータが ABC.TABLE1 から ABC.TABLE2 にロードされます。

```
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
```

### 例：XML データのロード

ユーザーは、表に挿入される文書を記述するために、XDS フィールドを持つデータ・ファイルを構成しました。内容は以下のとおりです。

```
1, "<XDS FIL=""file1.xml"" />"
2, "<XDS FIL='file2.xml' OFF='23' LEN='45' />"
```

最初の行では、XML 文書は file1.xml というファイルで示されます。区切り文字は二重引用符であり、二重引用符は XDS の内側にあるため、XDS 内にある二重引用符は二重になります。2 番目の行では、XML 文書は file2.xml というファイルで示され、バイト・オフセット 23 で始まります。長さは 45 バイトです。

ユーザーは、XML 列の構文解析または妥当性検査オプションを指定しないでロード・コマンドを発行し、データは正常にロードされました。

```
LOAD
FROM data.del of DEL INSERT INTO mytable
```

### 例：カーソルからの XML データのロード

データをカーソルからロードすることは、正規のリレーショナル列タイプを使用する場合と同じです。ユーザーは 2 つの表、T1 および T2 を持っており、それぞれは C1 という単一の XML 列で構成されます。T1 から T2 にロードするには、ユーザーは最初に次のようにカーソルを宣言します。

```
DECLARE
X1 CURSOR FOR SELECT C1 FROM T1;
```

その後、ユーザーは次のようにカーソル・タイプを使用して **LOAD** を発行できます。

```
LOAD FROM X1 of
CURSOR INSERT INTO T2
```

カーソル・タイプに XML 固有の **LOAD** オプションを適用する操作は、ファイルからロードする場合と同じです。

## XMLVALIDATE 節の使用例

### XMLVALIDATE USING XDS

#### 例 1

```
USING XDS DEFAULT schema-sqlid
```

ユーザーは、XDS で示されたスキーマに従って妥当性検査を行いたいと思います。SCH 属性を指定しない XDS 値がある場合、それらの文書はこの節で使用されるスキーマによって妥当性検査されます。

```
XMLVALIDATE
USING XDS DEFAULT S1.SCHEMA_B
```

#### 例 2

ユーザーは妥当性検査を行いたいと思いますが、XDS で言及されている特定のスキーマは無視します。

```
XMLVALIDATE USING XDS IGNORE S1.SCHEMA_C
```

#### 例 3

ユーザーは妥当性検査を行いたいと思いますが、XDS 内のいくつかのスキーマを再マップします。

```
XMLVALIDATE USING XDS MAP( (S1.SCHEMA_A,
S2.SCHEMA_B ), (S3.SCHEMA_C, S5.SCHEMA_E) )
```

上述の XDS の場合、SCH 属性が S1.SCHEMA\_A である文書は、S2.SCHEMA\_B に対して妥当性検査されます。また、SCH 属性が S3.SCHEMA\_C である文書は、S5.SCHEMA\_E に対して妥当性検査されます。

#### 例 4

ユーザーは、**DEFAULT**、**IGNORE**、**MAP** の各オプションを組み合わせて使用します。

```
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))
```

XML 列に以下の XDS が含まれているとすると、

```
<XDS FIL='xmlfile.001.xml' />
```

使用するデフォルト・スキーマとして "S8.SCHEMA\_H" が指定されているため、ファイル xmlfile.001.xml に含まれる文書は、SQL ID が "S8.SCHEMA\_H" である XML スキーマを使用して妥当性検査されます。

XML 列に以下の XDS が含まれているとすると、

## LOAD

```
<XDS FIL='xmlfile.002.xml' OFF='10' LEN='500' SCH='S10.SCHEMA_J' />
```

使用するスキーマとして XDS で指定されているのは "S10.SCHEMA\_J" ですが、このスキーマは **IGNORE** 節に含まれているため、ファイル xmlfile.002.xml に含まれる文書のスキーマ妥当性検査は実行されません。文書の内容はファイル中のバイト・オフセット 10 (11 番目のバイト) にあり、その長さは 500 バイトです。

XML 列に以下の XDS が含まれているとすると、

```
<XDS FIL='xmlfile.003.xml' SCH='S6.SCHEMA_F' />
```

ファイル xmlfile.003.xml に含まれる文書は、SQL ID が "S3.SCHEMA\_C" である XML スキーマを使用して妥当性検査されます。これは、**MAP** 節でスキーマ "S6.SCHEMA\_F" をスキーマ "S3.SCHEMA\_C" にマップすることが指定されているためです。それ以外のマッピングは実行されないため、この場合、スキーマ "S3.SCHEMA\_C" からスキーマ "S5.SCHEMA\_E" へのマッピングは適用されないことに注意してください。

XML 列に以下の XDS が含まれているとすると、

```
<XDS FIL='xmlfile.004.xml' SCH='S11.SCHEMA_K' />
```

ファイル xmlfile.004.xml に含まれる文書は、SQL ID が "S11.SCHEMA\_" である XML スキーマを使用して妥当性検査されます。この場合、**DEFAULT**、**IGNORE**、または **MAP** の指定のいずれも適用されないことに注意してください。

### XMLVALIDATE USING SCHEMA

ユーザーは、単一の SCHEMA に従ってすべての XML 文書の妥当性検査を行いたいと思います。以下の **XMLVALIDATE** 節の場合、

```
XMLVALIDATE USING SCHEMA S2.SCHEMA_B
```

XML 列に以下の XDS が含まれているとすると、

```
<XDS FIL='xmlfile.001.xml' />
```

ファイル xmlfile.001.xml に含まれる文書は、SQL ID が "S2.SCHEMA\_B" の XML スキーマを使用して妥当性検査されます。

XML 列に以下の XDS が含まれているとすると、

```
<XDS FIL='xmlfile.002.xml' SCH='S1.SCHEMA_A' />
```

ファイル xmlfile.002.xml 内の文書は、"S2.SCHEMA\_B" という SQL ID の XML スキーマを使用して妥当性検査されます。妥当性検査は **USING SCHEMA** 節で指定されたスキーマを使用して実行されるため、SCH 属性は無視されることに注意してください。

### XMLVALIDATE USING SCHEMALOCATION HINTS

ユーザーは、文書そのものの中にあるスキーマ情報に従って妥当性検査を行いたいと思います。以下の **XMLVALIDATE** 節の場合、

```
XMLVALIDATE  
USING SCHEMALOCATION HINTS
```

XML 列に以下の XDS が含まれているとすると、

```
<XDS FIL='xmlfile.001.xml' />
```

使用される XML スキーマは、文書の内容として含まれる schemaLocation 属性によって決まります。その属性が指定されていない場合、妥当性検査は実行されません。

XML 列に以下の XDS が含まれているとすると、

```
<XDS FIL='xmlfile.002.xml' SCH='S1.SCHEMA_A' />
```

使用される XML スキーマは、文書の内容として含まれる schemaLocation 属性によって決まります。その属性が指定されていない場合、妥当性検査は実行されません。妥当性検査は **SCHEMALOCATION HINTS** を使用して実行されるため、SCH 属性は無視されることに注意してください。

## 使用上の注意

- データは、入力ファイル内に並んでいる順序でロードされます。特定の順序にしたい場合には、ロードが試行される前にデータをソートしてください。ソース・データの順序を保持する必要がなければ、**ANYORDER** ファイル・タイプ修飾子を使用することを考慮してください。この修飾子については、以下の『ロード・ユーティリティーのファイル・タイプ修飾子』セクションを参照してください。
- ロード・ユーティリティーは、既存の定義に基づいて索引を作成します。ユニーク・キーの重複を処理するのに、例外表が使用されます。ユーティリティーは、参照整合性を強制したり、制約検査を実行したり、ロードする表に從属するマテリアライズ照会表を更新したりすることはありません。参照制約またはチェック制約を含む表は、**SET INTEGRITY** ペンディング状態になります。 **REFRESH IMMEDIATE** として定義されているサマリー表、およびロードする表に依存するサマリー表もまた、**SET INTEGRITY** ペンディング状態になります。これらの表に関して、**SET INTEGRITY** ペンディング (Set Integrity Pending) 状態を解除するには、**SET INTEGRITY** ステートメントを発行してください。ロード操作は、複製されたマテリアライズ照会表では実行できません。
- クラスタリング索引が表に存在する場合、ロード前にクラスタリング索引でデータをソートしてください。ただし、データはマルチディメンション・クラスタリング (MDC) 表にロードする前にソートする必要はありません。
- 保護された表へのロード時に例外表を指定すると、無効なセキュリティ・ラベルで保護されている行がその表に送られます。そのため、例外表にアクセスできるユーザーは、通常はアクセス権限のないデータにアクセスできてしまう可能性があります。セキュリティ・レベルを上げるために、誰に例外表アクセス権限を付与するかに注意し、行が修復されてロードする表にコピーされたら直ちにそれぞれの行を削除するとともに、使い終わった例外表は直ちにドロップしてください。
- 内部形式のセキュリティ・ラベルには、改行文字が含まれている可能性があります。 **DEL** ファイル形式を使用するファイルをロードする場合、この改行文字が区切り文字と間違われることがあります。この問題が起きた場合は、**LOAD** コマンドで **delprioritychar** ファイル・タイプ修飾子を指定することによって、区切り文字に以前のデフォルト優先順位を使用してください。
- **DECLARE CURSOR** ステートメントの実行中に指定した **DATABASE** キーワードが **CURSOR** ファイル・タイプを使用してロードを実行する場合、現在接続されているデータベース (ロード用) の認証に使用されるユーザー ID およびパス

## LOAD

ワードが (DECLARE CURSOR ステートメントの DATABASE オプションによって指定された) ソース・データベースの認証に使用されます。ユーザー ID またはパスワードがロード・データベースの接続に指定されない場合、ソース・データベースのユーザー ID とパスワードは DECLARE CURSOR ステートメントの実行中に指定する必要があります。

- 個々のパートが Windows システムから AIX システムにコピーされる、複数パーツの PC/IXF ファイルのロードがサポートされます。すべてのファイルの名前は、LOAD コマンドで指定する必要があります。例えば、LOAD FROM DATA.IXF, DATA.002 OF IXF INSERT INTO TABLE1 となります。論理分割された PC/IXF ファイルから Windows オペレーティング・システムにロードすることはサポートされていません。
- 失敗した LOAD を再始動する場合、その動作は、BUILD フェーズで索引用に REBUILD モードの使用が強制されるという点で、既存の動作に従います。
- COPY YES オプションが使用されている場合、ロード・ユーティリティーはイメージ・ファイルの非常に大きなコピーを生成する場合があります。この動作は、ロードされる LOB/LF 列値ごとに LOAD コマンドがデータのバッファー全体をコピー・イメージに書き出すときに行われることになっています。バッファーは内部オブジェクトであり、そのサイズは複数の内部および外部要因によって決まります。一般的に、バッファー・サイズは 68 KB から数百 KB の間です。
- データベース間での XML 文書のロードはサポートされておらず、エラー・メッセージ SQL1407N が戻されます。
- LOAD ユーティリティーでは、fenced プロシージャを参照する列を含む表へのロードはサポートされていません。このような表に対して LOAD コマンドを発行すると、エラー・メッセージ SQL1376N を受け取ります。この制約事項に対処するには、ルーチンを unfenced に再定義するか、インポート・ユーティリティーを使用することができます。
- 暗黙的な非表示列がデータベース表に含まれる場合には、ロード操作に非表示列のデータを含めるかどうかを指定する必要があります。
- IMPORT ユーティリティーは、表内の列数とデータ・ファイル内のフィールド数の突き合わせを行いません。このユーティリティーは、データ・ファイル内に十分な量のデータがあるかどうかをチェックします。データ・ファイル内の行に十分なデータ列が含まれていない場合、その行は拒否されて警告メッセージが表示される (表の対応するデータのない列が NOT NULL として定義されている場合) か、警告メッセージは表示されずに正常に挿入されます (表の対応する列が NULL として定義されている場合)。一方、行に必要な数より多くの列が含まれている場合、必要な数の列が処理され、残りのデータ列は省略されます。警告メッセージは表示されません。

### LOAD TERMINATE および LOAD RESTART ディクショナリー管理の要約

次の図表は、TERMINATE ディレクティブの下での LOAD 処理に関するコンプレッション・ディクショナリー管理の動作を要約しています。



表 30. LOAD TERMINATE ディクショナリー管理

表の COMPRESS 属 性	LOAD 前に表の行デ ータ・ディクショ ナリーが存在した かどうか	LOAD 前に XML ス トレージ・オブジェ クト・ディクショナ リーが存在する <sup>1</sup>	TERMINATE: LOAD REPLACE KEEPDICTIONARY または LOAD INSERT	TERMINATE: LOAD REPLACE RESETDICTIONARY
YES	YES	YES	既存のディクショナリーを保持します。	どちらのディクショナリーも保持しません。 <sup>2</sup>
YES	YES	NO	既存のディクショナリーを保持します。	何も保持しません。 <sup>2</sup>
YES	NO	YES	既存のディクショナリーを保持します。	何も保持しません。
YES	NO	NO	何も保持しません。	何も保持しません。
NO	YES	YES	既存のディクショナリーを保持します。	何も保持しません。
NO	YES	NO	既存のディクショナリーを保持します。	何も保持しません。
NO	NO	YES	既存のディクショナリーを保持します。	何も保持しません。
NO	NO	NO	何も行われません。	何も行われません。

## 注:

1. コンプレッション・ディクショナリーは、XML 列が DB2 バージョン 9.7 以降の表に追加された場合または表が Online Table Move を使用してマイグレーションされた場合にのみ、表の XML ストレージ・オブジェクトに対して作成できます。
2. 表でデータ・キャプチャーが使用可能にされているという特殊な場合には、表の行データ・ディクショナリーが保持されます。

**LOAD RESTART** は、到達した最後の整合点まで表を切り捨てます。**LOAD RESTART** 処理の一部として、最後の **LOAD** 整合点が取られたときにコンプレッション・ディクショナリーが表にあった場合、そのディクショナリーが表に存在します。その場合、**LOAD RESTART** では新規ディクショナリーは作成されません。考えられる条件の要約については、表 4 を参照してください。

表 31. LOAD RESTART ディクショナリー管理

表の COMPRESS 属 性	LOAD 整合点の前に 表の行データ・デ ィクショナリーが 存在するか <sup>1</sup>	最後の LOAD 前に XML ストレージ・ オブジェクト・ディ クショナリーが存在 したか <sup>2</sup>	RESTART: LOAD REPLACE KEEPDICTIONARY または LOAD INSERT	RESTART: LOAD REPLACE RESETDICTIONARY
YES	YES	YES	既存のディクショナリーを保持します。	既存のディクショナリーを保持します。
YES	YES	NO	既存の表の行データ・ディクショナリーを保存し、ADC の制約を受ける XML ディクショナリーを作成します。	既存の表の行データ・ディクショナリーを保存し、XML ディクショナリーを作成します。

## LOAD

表 31. LOAD RESTART ディクショナリー管理 (続き)

表の COMPRESS 属 性	LOAD 整合点の前に 表の行データ・デ ィクショナリーが 存在するか <sup>1</sup>	最後の LOAD 前に XML ストレージ・ オブジェクト・デ ィクショナリーが存在 したか <sup>2</sup>	RESTART: LOAD REPLACE KEEPDICTIONARY または LOAD INSERT	RESTART: LOAD REPLACE RESETDICTIONARY
YES	NO	YES	ADC の制約を受ける表の行 データ・ディクショナリー を作成します。既存の XML ディクショナリーを保持し ます。	表の行データ・ディクシ ョナリーを作成します。 既存の XML ディクショ ナリーを保持します。
YES	NO	NO	ADC の制約を受ける表の行 データ・ディクショナリー および XML ディクショナ リーを作成します。	表の行データ・ディクシ ョナリーおよび XML デ ィクショナリーを作成し ます。
NO	YES	YES	既存のディクショナリーを 保持します。	既存のディクショナリー を除去します。
NO	YES	NO	既存の表の行データ・ディ クショナリーを保存しま す。	既存の表の行データ・デ ィクショナリーを除去し ます。
NO	NO	YES	既存の XML ディクショナ リーを保持します。	既存の XML ディクショ ナリーを除去します。
NO	NO	NO	何も行われません。	何も行われません。

注:

1. XML データのロード時には **SAVECOUNT** オプションは使用できません。ロード・フェーズ中に失敗したロード操作は、操作の最初から再開されます。
2. コンプレッション・ディクショナリーは、XML 列が DB2 バージョン 9.7 以降の表に追加された場合または表が Online Table Move を使用してマイグレーションされた場合にのみ、表の XML ストレージ・オブジェクトに対して作成できます。

### ロード・ユーティリティー用のファイル・タイプ修飾子

表 32. ロード・ユーティリティーで有効なファイル・タイプ修飾子: すべてのファイル・フォーマット

修飾子	説明
<b>anyorder</b>	この修飾子は、 <b>cpu_parallelism</b> パラメーターと共に使用されます。ソース・データの順序を保持する必要がないことを指定します。そのため、SMP システムでは、パフォーマンスがかなり向上します。 <b>cpu_parallelism</b> の値が 1 になっていると、このオプションは無視されます。このオプションは、 <b>SAVECOUNT</b> > 0 の場合はサポートされません。整合点の後のクラッシュ・リカバリーでは、順序のとおりデータを読み込む必要があるからです。
<b>generatedignore</b>	この修飾子を指定すると、ロード・ユーティリティーは、データ・ファイルに入っている、すべての生成済み列のデータを無視するようになります。この結果、すべての生成列の値はユーティリティーによって生成されます。この修飾子は、 <b>generatedmissing</b> または <b>generatedoverride</b> 修飾子とともに使用することはできません。



表 32. ロード・ユーティリティで有効なファイル・タイプ修飾子: すべてのファイル・フォーマット (続き)

修飾子	説明
<b>generatedmissing</b>	この修飾子が指定されている場合、ユーティリティは、生成列のデータが入力データ・ファイルに入っていない (NULL も入っていない) ものを見なします。この結果、すべての生成列の値はユーティリティによって生成されます。この修飾子は、 <b>generatedignore</b> または <b>generatedoverride</b> 修飾子とともに使用することはできません。
<b>generatedoverride</b>	<p>この修飾子は、(こうした列のタイプの通常の規則に反して) 表内のすべての生成列で、ユーザーのデータを受け入れるようにロード・ユーティリティに指示します。別のデータベース・システムからデータをマイグレーションする場合や、<b>ROLLFORWARD DATABASE</b> コマンドの <b>RECOVER DROPPED TABLE</b> オプションを使用してリカバリーしたデータから表をロードする場合は、この修飾子を使用すると便利です。この修飾子を使用した場合、NULL 不可の生成列でデータまたは NULL データの入っていない行はリジェクトされます (SQL3116W)。この修飾子を使用される場合、表は SET INTEGRITY PENDING 状態になります。ユーザー提供の値をチェックせずに表を SET INTEGRITY PENDING 状態から解放するには、ロード操作後に以下のコマンドを発行します。</p> <pre>SET INTEGRITY FOR <i>table-name</i> GENERATED COLUMN IMMEDIATE UNCHECKED</pre> <p>表の SET INTEGRITY PENDING 状態を解除し、ユーザー定義の値の検査を強制するには、ロード操作の後以下のコマンドを発行してください。</p> <pre>SET INTEGRITY FOR <i>table-name</i> IMMEDIATE CHECKED.</pre> <p>この修飾子が指定され、パーティション・キー、ディメンション・キー、または分散キーのいずれかに生成された列がある場合、<b>LOAD</b> コマンドが修飾子を <b>generatedignore</b> 自動的に変換し、ロードを進めます。この影響で、生成された列の値すべてが再生成されます。</p> <p>この修飾子は、<b>generatedmissing</b> または <b>generatedignore</b> 修飾子と共に使用することはできません。</p>
<b>identityignore</b>	この修飾子はロード・ユーティリティに対して、ID 列のデータがデータ・ファイル内に存在するが、それらのデータは無視するべきものであることを通知します。この結果として、すべて ID 値はこのユーティリティによって生成されます。この動作は、GENERATED ALWAYS および GENERATED BY DEFAULT のどちらの ID 列の場合も同じです。つまり、GENERATED ALWAYS 列の場合には、リジェクトされる行はありません。この修飾子は、 <b>identitymissing</b> または <b>identityoverride</b> 修飾子とともに使用することはできません。
<b>identitymissing</b>	この修飾子を指定すると、ユーティリティは、ID 列のデータが入力データ・ファイルに入っていない (NULL も入っていない) ものを見なし、行ごとに値を生成します。この動作は、GENERATED ALWAYS および GENERATED BY DEFAULT のどちらの ID 列の場合も同じです。この修飾子は、 <b>identityignore</b> または <b>identityoverride</b> 修飾子とともに使用することはできません。

## LOAD

表 32. ロード・ユーティリティーで有効なファイル・タイプ修飾子: すべてのファイル・フォーマット (続き)

修飾子	説明
<b>identityoverride</b>	この修飾子は、GENERATED ALWAYS として定義した ID 列が、ロードする表に存在している場合にのみ使用するべきです。この修飾子はユーティリティーに対し、そのような列に関して、明示的な非 NULL データを受け入れる (これらのタイプの ID 列に関する通常の規則に反する) ように指示します。表を GENERATED ALWAYS として定義しなければならない状況で別のデータベース・システムからデータをマイグレーションする場合や、 <b>ROLLFORWARD DATABASE</b> コマンドの <b>DROPPED TABLE RECOVERY</b> オプションを使用してリカバリーしたデータから表をロードする場合は、この修飾子を使用すると便利です。この修飾子を使用すると、ID 列にデータのない行や NULL データが入っている行は、リジェクトされます (SQL3116W)。この修飾子は、 <b>identitymissing</b> または <b>identityignore</b> 修飾子とともに使用することはできません。このオプションが使用されていると、ロード・ユーティリティーは、表の ID 列内の値の固有性の保守または検証を行いません。
<b>implicitlyhiddeninclude</b>	この修飾子が指定されている場合、ユーティリティーは、入力データ・ファイルに暗黙非表示列へのデータが含まれると見なし、そのデータもロードします。この修飾子は、 <b>implicitlyhiddenmissing</b> 修飾子と共に使用することはできません。複数の修飾子が指定されている場合の優先順位については、注: セクションを参照してください。
<b>implicitlyhiddenmissing</b>	この修飾子を指定すると、ユーティリティーは入力データ・ファイルに暗黙的な非表示列のデータが含まれないと見なし、ユーティリティーがそれらの暗黙的な非表示列の値を生成します。この修飾子は、 <b>implicitlyhiddeninclude</b> 修飾子と共に使用することはできません。複数の修飾子が指定されている場合の優先順位については、注: セクションを参照してください。
<b>indexfreespace=x</b>	<p><math>x</math> は、0 から 99 までの整数です。この値は、ロード操作で索引を再作成するときに各索引ページに残すフリー・スペースのパーセンテージとして解釈されます。<b>INDEXING MODE INCREMENTAL</b> を指定したロード操作では、このオプションが無視されます。ページの最初の項目は、制限なしで追加されます。それより後の項目は、フリー・スペースのパーセントしきい値内である場合に追加されます。デフォルト値は、CREATE INDEX の実行時に使用した値です。</p> <p>この値は、CREATE INDEX ステートメントに指定された PCTFREE 値よりも優先して使用されます。<b>indexfreespace</b> オプションの対象になるのは、索引のリーフ・ページだけです。</p>

表 32. ロード・ユーティリティーで有効なファイル・タイプ修飾子: すべてのファイル・フォーマット (続き)

修飾子	説明
<b>lobsinfile</b>	<p><i>lob-path</i> には、LOB データの入ったファイルへのパスを指定します。ASC、DEL、または IXF ロード入力ファイルには、LOB 列に LOB データが入っているファイルの名前が入っています。</p> <p>ファイル・タイプが CURSOR の場合、このオプションはサポートされていません。</p> <p><b>lobsinfile</b> 修飾子を使用するときには、LOB ファイルの配置場所を <b>LOBS FROM</b> 節で指定します。<b>LOBS FROM</b> 節を指定すると、<b>lobsinfile</b> の動作が暗黙的にアクティブになります。<b>LOAD</b> ユーティリティーは、データをロードするときに、LOB ファイルを検索するためのパスのリストを <b>LOBS FROM</b> 節から受け取ります。</p> <p>各パスには、データ・ファイル内で LOB ロケーション指定子 (LLS) によって示される 1 つ以上の LOB の入った、少なくとも 1 つのファイルが組み込まれます。LLS は、LOB ファイル・パスに保管されるファイル内の LOB のロケーションのストリング表現です。LLS の形式は、<i>filename.ext.nnn.mmm/</i> です。ここで、<i>filename.ext</i> は、LOB が含まれているファイルの名前、<i>nnn</i> は、そのファイルに入っている LOB のオフセット (バイト単位)、<i>mmm</i> は、その LOB の長さ (バイト単位) です。例えば、ストリング <i>db2exp.001.123.456/</i> がデータ・ファイルに保管される場合、LOB はファイル <i>db2exp.001</i> のオフセット 123 に位置し、456 バイト長です。</p> <p>NULL LOB を指定するには、サイズに -1 と入力します。サイズを 0 と指定すると、長さが 0 の LOB として扱われます。長さが -1 の NULL LOB の場合、オフセットとファイル名は無視されます。例えば、NULL LOB の LLS は、<i>db2exp.001.7.-1/</i> のようになります。</p>
<b>noheader</b>	<p>ヘッダー検査コードをスキップします (単一パーティション・データベースのパーティション・グループに存在する表へのロード操作にのみ適用します)。</p> <p>単一パーティションのデータベース・パーティション・グループに存在する表に対してデフォルトの MPP ロード (モード PARTITION_AND_LOAD) が使用される場合、ファイルにはヘッダーが組み込まれないと想定されます。したがって、<b>noheader</b> 修飾子を指定する必要はありません。LOAD_ONLY モードが使用される場合、ファイルにはヘッダーが付いていると想定されます。<b>noheader</b> 修飾子が必要になるのは、ヘッダーのないファイルを使用して LOAD_ONLY 操作を実行する場合に限られます。</p>
<b>norowwarnings</b>	リジェクトされた行についてのすべての警告を抑制します。
<b>pagefreespace=x</b>	<p><i>x</i> は、0 から 100 までの整数です。この値は、各データ・ページ内でフリー・スペースとして残される部分のパーセンテージとして解釈されます。最小の行サイズのために指定値が無効な場合 (例えば、最低でも 3,000 バイトの長さが必要な行で、<i>x</i> 値が 50 になっている場合など) は、行が新しいページに配置されます。値として 100 を指定すると、各行が新しいページに配置されます。表の PCTFREE 値は、ページごとに指定されたフリー・スペースの量を決定します。ロード操作の <b>pagefreespace</b> 値または表の PCTFREE 値が設定されていないと、ユーティリティーはそれぞれのページで可能なかぎり多くのスペースを満たします。</p> <p><b>pagefreespace</b> に設定されている値は、表で指定されている PCTFREE 値をオーバーライドします。</p>

## LOAD

表 32. ロード・ユーティリティで有効なファイル・タイプ修飾子: すべてのファイル・フォーマット (続き)

修飾子	説明
<b>periodignore</b>	この修飾子はロード・ユーティリティに対して、期間列のデータがデータ・ファイル内に存在するが、それらのデータは無視するべきものであることを通知します。この修飾子が指定された場合、すべての期間列の値はユーティリティによって生成されます。この修飾子は、 <b>periodmissing</b> 修飾子および <b>periodoverride</b> 修飾子のどちらとも、共に使用することはできません。
<b>periodmissing</b>	この修飾子が指定されている場合、ユーティリティは、期間列のデータが入力データ・ファイルに入っていないものと見なします。この修飾子が指定された場合、すべての期間列の値はユーティリティによって生成されます。この修飾子は、 <b>periodignore</b> 修飾子および <b>periodoverride</b> 修飾子のどちらとも、共に使用することはできません。
<b>periodoverride</b>	この修飾子は、システム期間テンポラル表の GENERATED ALWAYS AS ROW BEGIN 列および GENERATED ALWAYS AS ROW END 列で、ユーザーのデータを受け入れるようにロード・ユーティリティに指示します。この動作は、このようなタイプの列に対する通常の規則とは異なっています。この修飾子は履歴データを維持する場合、およびシステム期間テンポラル表にタイム・スタンプを含むデータをロードする場合に、役立てることができます。この修飾子を使用すると、データが入っていない行や、ROW BEGIN 列または ROW END 列に対する NULL データはリジェクトされます。
<b>rowchangetimestampignore</b>	この修飾子はロード・ユーティリティに対して、ROW CHANGE TIMESTAMP 列のデータがデータ・ファイル内に存在するが、それらのデータは無視するべきものであることを通知します。この結果、すべての ROW CHANGE TIMESTAMP 列がユーティリティによって生成されます。この動作は、GENERATED ALWAYS 列でも GENERATED BY DEFAULT 列でも同じです。つまり、GENERATED ALWAYS 列の場合には、リジェクトされる行はありません。この修飾子は、 <b>rowchangetimestampmissing</b> または <b>rowchangetimestampoverride</b> 修飾子とともに使用することはできません。
<b>rowchangetimestampmissing</b>	この修飾子を指定すると、ユーティリティは、行変更タイム・スタンプ列のデータが入力データ・ファイルに入っていない (NULL も入っていない) ものと見なし、行ごとに値を生成します。この動作は、GENERATED ALWAYS 列でも GENERATED BY DEFAULT 列でも同じです。この修飾子は、 <b>rowchangetimestampignore</b> または <b>rowchangetimestampoverride</b> 修飾子とともに使用することはできません。
<b>rowchangetimestampoverride</b>	この修飾子は、GENERATED ALWAYS として定義した ROW CHANGE TIMESTAMP 列が、ロードする表に存在している場合のみ使用するべきです。この修飾子はユーティリティに対し、そのような列に関して、明示的な非 NULL データを受け入れる (これらのタイプの ROW CHANGE TIMESTAMP 列に関する通常の規則に反する) ように指示します。表を GENERATED ALWAYS として定義しなければならない状況で別のデータベース・システムからデータをマイグレーションする場合や、 <b>ROLLFORWARD DATABASE</b> コマンドの <b>DROPPED TABLE RECOVERY</b> オプションを使用してリカバリーしたデータから表をロードする場合は、この修飾子を使用すると便利です。この修飾子を使用すると、データが入っていない行や ROW CHANGE TIMESTAMP 列に対する NULL データはすべてリジェクトされます (SQL3116W)。この修飾子は、 <b>rowchangetimestampmissing</b> または <b>rowchangetimestampignore</b> 修飾子とともに使用することはできません。このオプションが使用されていると、ロード・ユーティリティは、表の ROW CHANGE TIMESTAMP 列内の値の固有性の保守または検証を行いません。

表 32. ロード・ユーティリティーで有効なファイル・タイプ修飾子: すべてのファイル・フォーマット (続き)

修飾子	説明
<b>seclabelchar</b>	<p>入力ソース・ファイル内のセキュリティ・ラベルが、デフォルトのエンコードされた数値形式ではなく、セキュリティ・ラベル値の文字列形式であることを示します。 <b>LOAD</b> は、ロード時に各セキュリティ・ラベルを内部形式に変換します。文字列が適切な形式ではない場合、行はロードされず、警告 (SQLSTATE 01H53、SQLCODE SQL3242W) が戻されます。文字列が表を保護するセキュリティ・ポリシーの一部である有効なセキュリティ・ラベルを表していない場合、行はロードされず、警告 (SQLSTATE 01H53、SQLCODE SQL3243W) が戻されます。</p> <p><b>seclabelname</b> 修飾子を指定した場合は、この修飾子を指定できません。同時に指定すると、ロードは失敗し、エラー (SQLCODE SQL3525N) が戻されます。</p> <p>単一の DB2SECURITYLABEL 列で構成される表がある場合、データ・ファイルは例えば次のようになります。</p> <pre>"CONFIDENTIAL:ALPHA:G2" "CONFIDENTIAL;SIGMA:G2" "TOP SECRET:ALPHA:G2"</pre> <p>このデータのロードまたはインポートでは、以下のように <b>seclabelchar</b> ファイル・タイプ修飾子を使用する必要があります。</p> <pre>LOAD FROM input.del OF DEL MODIFIED BY SECLABELCHAR INSERT INTO t1</pre>
<b>seclabelname</b>	<p>入力ソース・ファイル内のセキュリティ・ラベルが、デフォルトのエンコードされた数値形式ではなく、名前によって指定されることを示します。 <b>LOAD</b> は、その名前に対応する適切なセキュリティ・ラベルがあれば、その名前をそのセキュリティ・ラベルに変換します。表を保護するセキュリティ・ポリシーで、指定された名前のセキュリティ・ラベルが存在しない場合、行はロードされず、警告 (SQLSTATE 01H53、SQLCODE SQL3244W) が戻されます。</p> <p><b>seclabelchar</b> 修飾子を指定した場合は、この修飾子を指定できません。同時に指定すると、ロードは失敗し、エラー (SQLCODE SQL3525N) が戻されます。</p> <p>単一の DB2SECURITYLABEL 列で構成される表がある場合、データ・ファイルは以下のようなセキュリティ・ラベル名で構成される可能性があります。</p> <pre>"LABEL1" "LABEL1" "LABEL2"</pre> <p>このデータのロードまたはインポートでは、以下のように <b>seclabelname</b> ファイル・タイプ修飾子を使用する必要があります。</p> <pre>LOAD FROM input.del OF DEL MODIFIED BY SECLABELNAME INSERT INTO t1</pre> <p><b>注:</b> ファイル・タイプが ASC の場合、セキュリティ・ラベルの名前の後にスペースがあれば、それも名前の一部として解釈されます。これを回避するには、<b>striptblanks</b> ファイル・タイプ修飾子を使用してスペースが除去されるようにします。</p>



## LOAD

表 32. ロード・ユーティリティーで有効なファイル・タイプ修飾子: すべてのファイル・フォーマット (続き)

修飾子	説明
<b>total freespace=x</b>	$x$ は、0 以上の整数です。この値は表内の合計ページのうち、表の終わりにフリー・スペースとして追加される部分のパーセンテージと解釈されます。例えば、 $x$ が 20 で、データのロード後に表に 100 個のデータ・ページがある場合は、20 個の空ページが追加されます。その表のデータ・ページの合計数は 120 になります。データ・ページの総数は、表の索引ページの数には影響を与えません。このオプションは、索引オブジェクトには影響を与えません。このオプションを指定して 2 つのロードが行われる場合、2 番目のロードは、最初のロードによって最後に付加された余分のスペースを再利用しません。
<b>transactionidignore</b>	この修飾子はロード・ユーティリティーに対して、TRANSACTION START ID 列のデータがデータ・ファイル内に存在するが、それらのデータは無視するべきものであることを通知します。この修飾子が指定された場合、TRANSACTION START ID 列の値はユーティリティーによって生成されます。この修飾子は、 <b>transactionidmissing</b> 修飾子および <b>transactionidoverride</b> 修飾子のどちらとも、共に使用することはできません。
<b>transactionidmissing</b>	この修飾子が指定されている場合、ユーティリティーは、TRANSACTION START ID 列のデータが入力データ・ファイルに入っていないものと見なします。この修飾子が指定された場合、TRANSACTION START ID 列の値はユーティリティーによって生成されます。この修飾子は、 <b>transactionidignore</b> 修飾子および <b>transactionidoverride</b> 修飾子のどちらとも、共に使用することはできません。
<b>transactionidoverride</b>	この修飾子は、システム期間テンポラル表の GENERATED ALWAYS AS TRANSACTION START ID 列で、ユーザーのデータを受け入れるようにロード・ユーティリティーに指示します。この動作は、このタイプの列に対する通常の規則とは異なっています。この修飾子を使用すると、データが入っていない行や、TRANSACTION START ID 列に対する NULL データはリジェクトされます。
<b>usedefaults</b>	<p>ターゲット表の列のソース列が指定されているが、1 つ以上の行インスタンスのデータが入っていない場合は、デフォルト値がロードされます。欠落データの例を以下に示します。</p> <ul style="list-style-type: none"> <li>• DEL ファイルの場合: 列の値として、2 つの連続した列区切り (,) や、任意の数のスペースで分離した 2 つの連続する列区切り (, ) が指定されている。</li> <li>• DEL/ASC ファイルの場合: 列が不足している行、または元の指定には十分な長さでない行。ASC ファイルの場合: NULL 列値は明示的に欠落していると思われず、NULL 列値にはデフォルトが置換されません。数値、日付、時刻、タイム・スタンプの列では、全桁スペース文字で NULL 列値を表記します。また、どのタイプの列でも、NULL INDICATOR を使用すれば、その列が NULL であることを示せます。</li> </ul> <p>このオプションが指定されていない場合、行インスタンスのソース列にデータがないと、以下のいずれかの処理が行われます。</p> <ul style="list-style-type: none"> <li>• DEL/ASC ファイルの場合: 列が NULL 可能であれば、NULL がロードされます。列が NULL 可能でない場合、ユーティリティーはその行をリジェクトします。</li> </ul>

表 33. ロード・ユーティリティーで有効なファイル・タイプ修飾子: ASCII ファイル・フォーマット (ASC/DEL)

修飾子	説明
<code>codepage=x</code>	<p><code>x</code> は、ASCII 文字ストリングです。この値は、入力データ・セット内のデータのコード・ページとして解釈されます。ロード操作時に、文字データ (および文字内で指定された数値データ) は、このコード・ページからデータベースのコード・ページへ変換されます。</p> <p>以下の規則が適用されます。</p> <ul style="list-style-type: none"> <li>• DBCS のみ (GRAPHIC)、混合 DBCS、および EUC の場合、区切り文字の範囲は <code>x00</code> から <code>x3F</code> に制限されます。</li> <li>• EBCDIC コード・ページで指定された DEL データの場合、区切り文字は DBCS のシフトイン文字およびシフトアウト文字と一致しない場合があります。</li> <li>• <code>nullindchar</code> では、標準の ASCII セットのコード・ポイント <code>x20</code> から <code>x7F</code> の範囲に含まれているシンボルを指定する必要があります。これは、ASCII 記号およびコード・ポイントを示します。EBCDIC データでは、コード・ポイントが異なるとしても、対応する記号を使用できます。</li> </ul> <p>ファイル・タイプが <code>CURSOR</code> の場合、このオプションはサポートされていません。</p>
<code>dateformat="x"</code>	<p><code>x</code> はソース・ファイルの日付のフォーマットです。<sup>1</sup> 有効な日付エレメントは次のとおりです。</p> <p>YYYY - 年 (0000 から 9999 の範囲の 4 桁の数)  M - 月 (1 から 12 の範囲の 1 桁または 2 桁の数)  MM - 月 (01 から 12 の 2 桁の数。  M とは相互に排他的)  D - 日 (1 から 31 の範囲の 1 桁または 2 桁の数)  DD - 日 (01 から 31 の範囲の 2 桁の数。  D とは相互に排他的)  DDD - 元日から数えた日数 (001 から 366 の範囲の 3 桁の数。  他の日または月エレメントとは相互に排他的)</p> <p>デフォルト値の 1 が、指定されない各エレメントに割り当てられます。日付形式の例を以下に示します。</p> <p>"D-M-YYYY"  "MM.DD.YYYY"  "YYYYDDD"</p>

## LOAD

表 33. ロード・ユーティリティーで有効なファイル・タイプ修飾子: ASCII ファイル・フォーマット (ASC/DEL) (続き)

修飾子	説明
<b>dumpfile = x</b>	<p>x は、リジェクトされた行を書き込む例外ファイルの (サーバー・データベース・パーティションによる) 完全修飾名です。1 レコードにつき、最大で 32 KB のデータが書き込まれます。以下に、ダンプ・ファイルの指定方法の例を示します。</p> <pre>db2 load from data of del   modified by dumpfile = /u/user/filename   insert into table_name</pre> <p>ファイルは、インスタンスの所有者によって作成されて所有されます。デフォルトのファイル権限をオーバーライドするには、<b>dumpfileaccessall</b> ファイル・タイプ修飾子を使用します。</p> <p><b>注:</b></p> <ol style="list-style-type: none"> <li>パーティション・データベース環境の場合、パスはロードを実行するデータベース・パーティションにローカルなものでなければなりません。それによって、並行して実行される複数のロード操作が同じファイルに書き込むことを防ぐことができます。</li> <li>ファイルの内容は、非同期バッファ・モードでディスクに書き込まれます。ロード操作が失敗した場合や割り込みが発生した場合は、ディスクにコミットされたレコードの数を確実に把握する方法がありません。<b>LOAD RESTART</b> 後の整合性も保証できません。ファイルが完全であるとされるのは、1 回のパスの中で開始して完了するロード操作の場合だけです。</li> <li>指定されたファイルが既に存在する場合は、再作成されずに切り捨てられます。</li> </ol>
<b>dumpfileaccessall</b>	<p>ダンプ・ファイルの作成時に、読み取りアクセスを OTHERS に付与します。</p> <p>このファイル・タイプ修飾子が有効なのは、以下の場合のみです。</p> <ol style="list-style-type: none"> <li><b>dumpfile</b> ファイル・タイプ修飾子と一緒に使用された場合。</li> <li>ロード・ターゲット表に対してユーザーが <b>SELECT</b> 特権をもっている場合。</li> <li>UNIX オペレーティング・システムに置かれている DB2 サーバー・データベース・パーティション上で発行された場合。</li> </ol> <p>指定されたファイルが既に存在する場合、その権限は変更されません。</p>
<b>fastparse</b>	<p>使用に際しては、注意が必要です。ユーザー指定の列値の構文検査が削減されるので、パフォーマンスは向上します。表は、体系的な正確さが確保されます (セグメント化違反またはトラップを防ぐための十分なデータ・チェックがユーティリティーで実行されます) が、データの一貫性の妥当性検査は行われません。データに一貫性があり、正確であることが確実な場合にのみ、このオプションを使用してください。例えば、ユーザー指定のデータに無効なタイム・スタンプ列値 :1&gt;0-00-20-07.11.12.000000 が含まれている場合でも、<b>fastparse</b> が指定されていれば、その値は表に挿入されてしまいますが、<b>fastparse</b> が指定されていなければ、その値はリジェクトされます。</p>
<b>implieddecimal</b>	<p>暗黙指定されている小数点の位置が列定義によって決定され、値の終わりにあるとは見なされなくなります。例えば、値 12345 は、12345.00 ではなく、123.45 として DECIMAL(8,2) 列にロードされます。</p> <p>この修飾子は、<b>packeddecimal</b> 修飾子と共に使用することはできません。</p>



表 33. ロード・ユーティリティーで有効なファイル・タイプ修飾子: ASCII ファイル・フォーマット (ASC/DEL) (続き)

修飾子	説明
<b>timeformat="x"</b>	<p>x はソース・ファイル内の時刻のフォーマットです。<sup>1</sup> 有効な時刻エレメントは以下のとおりです。</p> <p>H - 時 (12 時間制の場合は 0 から 12、24 時間制では 0 から 24 の範囲の 1 桁または 2 桁の数)</p> <p>HH - 時 (12 時間制の場合は 00 から 12、24 時間制では 00 から 24 の範囲の 2 桁の数; H と相互に排他的)</p> <p>M - 分 (0 から 59 の範囲の 1 桁または 2 桁の数)</p> <p>MM - 分 (00 から 59 の範囲の 2 桁の数。 M とは相互に排他的)</p> <p>S - 秒 (0 から 59 の範囲の 1 桁または 2 桁の数)</p> <p>SS - 秒 (00 から 59 の範囲の 2 桁の数。 S と相互に排他的)</p> <p>SSSSS - 夜中の 12 時から数えた秒数 (00000 から 86400 の範囲の 5 桁の数。 他の時刻エレメントとは相互に排他的)</p> <p>TT - 午前/午後の指定子 (AM または PM)</p> <p>指定されない各エレメントには、デフォルト値の 0 が割り当てられます。時刻フォーマットの例を以下に示します。</p> <p>"HH:MM:SS" "HH.MM TT" "SSSSS"</p>
<b>timestampformat="x"</b>	<p>x はソース・ファイルのタイム・スタンプのフォーマットです。<sup>1</sup> 有効なタイム・スタンプ・エレメントは以下のとおりです。</p> <p>YYYY - 年 (0000 から 9999 の範囲の 4 桁の数)</p> <p>M - 月 (1 から 12 の範囲の 1 桁または 2 桁の数)</p> <p>MM - 月 (01 から 12 の 2 桁の数。 M および MMM とは相互に排他的)</p> <p>MMM - 月 (大文字小文字を区別しない月名の 3 文字の省略形。 M と MM とは相互に排他的)</p> <p>D - 日 (1 から 31 の範囲の 1 桁または 2 桁の数)</p> <p>DD - 日 (01 から 31 の範囲の 2 桁の数。D とは相互に排他的)</p> <p>DDD - 元日から数えた日数 (001 から 366 の範囲の 3 桁の数。 他の日または月のエレメントとは相互に排他的)</p> <p>H - 時 (12 時間制の場合は 0 から 12、24 時間制では 0 から 24 の範囲の 1 桁または 2 桁の数。)</p> <p>HH - 時 (12 時間制の場合は 00 から 12、24 時間制では 0 から 24 の範囲の 2 桁の数。 H と相互に排他的)</p> <p>M - 分 (0 から 59 の範囲の 1 桁または 2 桁の数)</p> <p>MM - 分 (00 から 59 の範囲の 2 桁の数。 M (分) とは相互に排他的)</p> <p>S - 秒 (0 から 59 の範囲の 1 桁または 2 桁の数)</p> <p>SS - 秒 (00 から 59 の範囲の 2 桁の数。 S と相互に排他的)</p> <p>SSSSS - 夜中の 12 時から数えた秒数 (00000 から 86400 の範囲の 5 桁の数。 他の時刻エレメントとは相互に排他的)</p> <p>U (1 から 12 時) - 小数秒 (U のオカレンス数は、各桁を 0 から 9 の範囲として、桁数を表します)</p> <p>TT - 午前/午後の指定子 (AM または PM)</p>

## LOAD

表 33. ロード・ユーティリティーで有効なファイル・タイプ修飾子: ASCII ファイル・フォーマット (ASC/DEL) (続き)

修飾子	説明
<b>timestampformat="x"</b> (Continued)	<p>YYYY、M、MM、D、DD、または DDD エlementが指定されていない場合、デフォルト値の 1 が割り当てられます。MMM Elementが指定されていない場合、デフォルト値の「Jan」が割り当てられます。他のElementが指定されていない場合には、デフォルト値の 0 が割り当てられます。タイム・スタンプ・フォーマットの例を以下のセクションに示します。</p> <p style="text-align: center;">"YYYY/MM/DD HH:MM:SS.UUUUUU"</p> <p>MMM Elementの有効な値は、 「jan」、「feb」、「mar」、「apr」、「may」、「jun」、「jul」、「aug」、 「sep」、「oct」、「nov」、および「dec」です。これらの値では、大/小文字は区別されません。</p> <p><b>timestampformat</b> 修飾子を指定しなかった場合、ロード・ユーティリティーは、タイム・スタンプ・フィールドで以下の 2 つの有効な形式のいずれかを使用します。</p> <p>YYYY-MM-DD-HH.MM.SS YYYY-MM-DD HH:MM:SS</p> <p>ロード・ユーティリティーは、DD と HH の間の区切り記号を調べてフォーマットを選択します。ダッシュ「-」になっていれば、ロード・ユーティリティーは、通常のダッシュとドットの形式 (YYYY-MM-DD-HH.MM.SS) を使用します。区切り文字がブランク・スペースの場合、ロード・ユーティリティーはコロン「:」を使用して、HH、MM、および SS を区切ります。</p> <p>どちらのフォーマットでも、マイクロ秒フィールド (UUUUUU) が含まれる場合、ロード・ユーティリティーは区切り文字としてドット「.」を使用します。 YYYY-MM-DD-HH.MM.SS.UUUUUU も YYYY-MM-DD HH:MM:SS.UUUUUU も有効です。</p> <p>次の例では、ユーザー定義の日時形式を指示するデータを、schedule という表にロードする方法を示します。</p> <pre>db2 load from delfile2 of del       modified by timestampformat="yyyy.mm.dd hh:mm tt"       insert into schedule</pre>

表 33. ロード・ユーティリティーで有効なファイル・タイプ修飾子: ASCII ファイル・フォーマット (ASC/DEL) (続き)

修飾子	説明
<b>usegraphiccodepage</b>	<p><b>usegraphiccodepage</b> が指定された場合、GRAPHIC または 2 バイト文字ラージ・オブジェクト (DBCLOB) データ・フィールドにロードされるデータは、GRAPHIC コード・ページであると見なされます。データの残りは、文字コード・ページであると見なされます。GRAPHIC コード・ページは、文字コード・ページと関連付けられます。<b>LOAD</b> は、<b>codepage</b> 修飾子が指定されている場合はその修飾子、または <b>codepage</b> 修飾子が指定されていない場合は、データベースのコード・ページを通じて、文字コード・ページを決定します。</p> <p>この修飾子は、リカバリーされている表に GRAPHIC データがある場合のみ、ドロップ済み表のリカバリーによって生成された区切りデータ・ファイルとともに使用される必要があります。</p> <p><b>制約事項</b></p> <p><b>EXPORT</b> ユーティリティーで作成された DEL ファイルは、1 つのコード・ページのみでエンコードされたデータを含んでいるため、これらのファイルで <b>usegraphiccodepage</b> 修飾子を指定することはできません。 <b>usegraphiccodepage</b> 修飾子はまた、ファイル内の 2 バイト文字ラージ・オブジェクト (DBCLOB) には無視されます。</p>
<b>xmlchar</b>	<p>XML 文書が文字コード・ページでエンコードされていることを示します。</p> <p>このオプションは、指定の文字コード・ページでエンコードされていてもエンコード宣言を含まない XML 文書を処理するために役立ちます。</p> <p>各文書で、宣言タグが存在してエンコード属性が含まれる場合、そのエンコード方式は文字コード・ページと一致する必要があります。一致しない場合、その文書を含む行はリジェクトされます。文字コード・ページは <b>codepage</b> ファイル・タイプ修飾子で指定されている値であるか、または指定がない場合はアプリケーションのコード・ページであることに注意してください。デフォルトでは、文書は Unicode でエンコードされているか、またはエンコード属性のある宣言タグを含んでいません。</p>
<b>xmlgraphic</b>	<p>XML 文書が指定された GRAPHIC コード・ページでエンコードされていることを示します。</p> <p>このオプションは、特定の GRAPHIC コード・ページでエンコードされていてもエンコード宣言を含まない XML 文書を処理するために役立ちます。</p> <p>各文書で、宣言タグが存在してエンコード属性が含まれる場合、そのエンコード方式は GRAPHIC コード・ページと一致する必要があります。一致しない場合、その文書を含む行はリジェクトされます。GRAPHIC コード・ページは <b>codepage</b> ファイル・タイプ修飾子で指定されている値のグラフィック・コンポーネントであるか、または指定がない場合はアプリケーションのコード・ページのグラフィック・コンポーネントであることに注意してください。デフォルトでは、文書は Unicode でエンコードされているか、またはエンコード属性のある宣言タグを含んでいません。</p>

## LOAD

表 34. ロード・ユーティリティーで有効なファイル・タイプ修飾子: ASC ファイル・フォーマット (区切り文字で区切られていない ASCII)

修飾子	説明
<b>binarynumerics</b>	<p>数値データ (DECIMAL 以外) は、文字表記ではなく、バイナリー形式でなければなりません。これによって、コストの大きい変換操作を避けることができます。</p> <p>このオプションがサポートされるのは、定位置 ASC において、<b>reclen</b> オプションによって固定長レコードが指定されている場合だけです。</p> <p>以下の規則が適用されます。</p> <ul style="list-style-type: none"> <li>• BIGINT、INTEGER、および SMALLINT を除き、データ・タイプ間の変換は実行されません。</li> <li>• データ長は、それぞれのターゲット列定義と一致している必要があります。</li> <li>• FLOAT は、IEEE 浮動小数点フォーマットでなければなりません。</li> <li>• ロード・ソース・ファイル中のバイナリー・データは、ロード操作を実行するプラットフォームに関係なく、ビッグ・エンディアンであると見なされます。</li> </ul> <p>この修飾子の影響を受ける列のデータに NULL があってはなりません。この修飾子を使用すると、ブランク (通常は NULL と解釈される) は、バイナリー値であると解釈されます。</p>
<b>nochecklengths</b>	<p><b>nochecklengths</b> を指定した場合は、ソース・データの中にターゲット表の列のサイズを超える列定義がある場合であっても、各行のロードが試みられます。コード・ページ変換によってソース・データが縮小されれば、そのような行であったとしても正常にロードすることができます。例えば、ソースに 4 バイトの EUC データがある場合、それがターゲットで 2 バイトの DBCS データに縮小されれば、必要なスペースは半分で済みます。このオプションが特に役立つのは、列の定義は不一致であるがソース・データが常に適合することが分かっている場合です。</p>
<b>nullindchar=<i>x</i></b>	<p><i>x</i> は、単一文字です。NULL 値を示す文字を <i>x</i> に変更します。<i>x</i> のデフォルト値は Y です。<sup>2</sup></p> <p>文字が 1 つの英字である場合を除いて、この修飾子は EBCDIC データ・ファイルで大文字小文字を区別します。例えば、NULL 標識文字を文字 N に指定した場合、<i>n</i> も NULL 標識と認識されます。</p>

表 34. ロード・ユーティリティーで有効なファイル・タイプ修飾子: ASC ファイル・フォーマット (区切り文字で区切られていない ASCII) (続き)

修飾子	説明
<b>packeddecimal</b>	<p><b>binarynumerics</b> 修飾子は DECIMAL フィールド・タイプで構成されないため、パック 10 進数データを直接ロードします。</p> <p>このオプションがサポートされるのは、定位置 ASC において、<b>reclen</b> オプションによって固定長レコードが指定されている場合だけです。</p> <p>符号ニブル用にサポートされる値は以下のとおりです。</p> <pre>+ = 0xC 0xA 0xE 0xF - = 0xD 0xB</pre> <p>この修飾子の影響を受ける列のデータに NULL があってはなりません。この修飾子を使用すると、ブランク (通常は NULL と解釈される) は、バイナリー値であると解釈されます。</p> <p>サーバーのプラットフォームには関係なく、ロードのソース・ファイルに入っているバイナリー・データのバイト順はビッグ・エンディアンであることが前提となっています。つまり、この修飾子を Windows オペレーティング・システムで使用する場合も、バイト順を逆にしてはなりません。</p> <p>この修飾子は、<b>implieddecimal</b> 修飾子と共に使用することはできません。</p>
<b>reclen=x</b>	<p><math>x</math> は、最大値 32767 の整数です。各行では <math>x</math> 個の文字が読み取られ、行の終わりを示す改行文字は使用されません。</p>
<b>striptblanks</b>	<p>データを可変長フィールドにロードする際に、後書きブランク・スペースを切り捨てます。このオプションを指定しない場合、ブランク・スペースはそのまま保持されます。</p> <p>このオプションは、<b>striptnulls</b> と一緒に指定することはできません。これらは、相互に排他的なオプションです。このオプションは、廃止された <b>t</b> オプション (下位互換性のためだけにサポートされる) に代わるものです。</p>
<b>striptnulls</b>	<p>データを可変長フィールドにロードする際に、後書き NULL (0x00 文字) を切り捨てます。このオプションを指定しない場合、NULL はそのまま保持されます。</p> <p>このオプションは、<b>striptblanks</b> と一緒に指定することはできません。これらは、相互に排他的なオプションです。このオプションは、廃止された <b>padwithzero</b> オプション (下位互換性のためだけにサポートされる) に代わるものです。</p>
<b>zoneddecimal</b>	<p>ゾーン 10 進数データをロードします。DECIMAL フィールド・タイプは、<b>binarynumerics</b> 修飾子の対象に含まれていません。このオプションがサポートされるのは、定位置 ASC において、<b>reclen</b> オプションによって固定長レコードが指定されている場合だけです。</p> <p>ハーフバイト符号値は、以下の値のいずれかになります。</p> <pre>+ = 0xC 0xA 0xE 0xF 0x3 - = 0xD 0xB 0x7</pre> <p>数字としてサポートされている値は、0x0 から 0x9 です。</p> <p>ゾーンとしてサポートされている値は、0x3 および 0xF です。</p>

## LOAD

表 35. ロード・ユーティリティーで有効なファイル・タイプ修飾子: DEL ファイル・フォーマット (区切り文字で区切られている ASCII)

修飾子	説明
<b>charde1x</b>	<p><i>x</i> は単一文字のストリング区切り文字です。デフォルト値は、二重引用符 (") です。指定した文字は、文字ストリングを囲むために、二重引用符の代わりに使用されます。<sup>23</sup> 文字ストリング区切り文字として明示的に二重引用符 (") を指定したい場合、以下のように指定します。</p> <pre>modified by charde1""</pre> <p>以下のように、文字ストリング区切りとして単一引用符 (') を指定することもできます。</p> <pre>modified by charde1''</pre>
<b>colde1x</b>	<p><i>x</i> は単一文字の列区切り文字です。デフォルト値はコンマ (,) です。指定した文字は、列の終わりを表すために、コンマの代わりに使用されます。<sup>23</sup></p>
<b>decplusblank</b>	<p>正符号文字。正の 10 進値の接頭部として、正符号 (+) ではなくブランク・スペースを使用します。デフォルトのアクションでは、正の 10 進数の前に正符号 (+) が付けられます。</p>
<b>decptx</b>	<p><i>x</i> は、小数点文字としてピリオドの代わりに使用される単一文字です。デフォルト値はピリオド (.) です。指定した文字は、小数点文字としてピリオドの代わりに使用されます。<sup>23</sup></p>
<b>delprioritychar</b>	<p>区切り文字の現在のデフォルト優先順位は、(1) レコード区切り文字、(2) 区切り文字、(3) 列区切り文字です。この修飾子を使用すると、区切り文字の優先順位が (1) 区切り文字、(2) レコード区切り文字、(3) 列区切り文字に戻り、以前の優先順位に依存している既存のアプリケーションが保護されます。構文:</p> <pre>db2 load ... modified by delprioritychar ...</pre> <p>例えば、以下のような DEL データ・ファイルがあるとします。</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><b>delprioritychar</b> 修飾子を指定しているため、このデータ・ファイルは、2 行だけになります。2 番目の &lt;row delimiter&gt; は 2 番目の行の最初のデータ列の一部と解釈されますが、1 番目と 3 番目の &lt;row delimiter&gt; は実レコードの区切り文字と解釈されます。この修飾子が指定されていない場合、このデータ・ファイルでは 3 行になり、各行は &lt;row delimiter&gt; によって区切られます。</p>
<b>keepblanks</b>	<p>タイプが CHAR、VARCHAR、LONG VARCHAR、または CLOB の各フィールドの前後のブランクを保持します。このオプションを指定しないと、区切り文字で囲まれていないすべての前後のブランクは除去され、表のすべてのブランク・フィールドに NULL が挿入されます。</p> <p>以下の例では、データ・ファイルにある前後のブランクを保存しながら、TABLE1 という表にデータをロードする方法を示します。</p> <pre>db2 load from delfile3 of del modified by keepblanks insert into table1</pre>



表 35. ロード・ユーティリティで有効なファイル・タイプ修飾子: DEL ファイル・フォーマット (区切り文字で区切られている ASCII) (続き)

修飾子	説明
<b>nocharde1</b>	<p>ロード・ユーティリティは、列区切り文字と列区切り文字の間にあるすべてのバイトが列データの一部であると見なします。文字区切り文字は、列データの一部として構文解析されます。DB2 データベース・システムを使用してエクスポートしたデータについては、このオプションを指定しないでください (ただし、エクスポート時に <b>nocharde1</b> を指定していた場合は例外です)。これは、区切り文字を持たないベンダー・データ・ファイルをサポートするために用意されています。不適切に使用すると、データが損失または破壊される場合があります。</p> <p>このオプションを <b>chardelx</b>、<b>delprioritychar</b> または <b>nodoublede1</b> と一緒に指定することはできません。これらは、相互に排他的なオプションです。</p>
<b>nodoublede1</b>	二重文字区切りの認識を抑止します。

表 36. ロード・ユーティリティで有効なファイル・タイプ修飾子: IXF ファイル・フォーマット

修飾子	説明
<b>forcein</b>	<p>コード・ページが不一致でもデータを受け入れ、コード・ページ間の変換を抑止するようにユーティリティに指示します。</p> <p>固定長ターゲット・フィールドに、そのデータが入るだけの十分な大きさがあるかどうかチェックされます。<b>nochecklengths</b> を指定した場合、そのような検査は実行されず、各行のロードが試みられます。</p>
<b>nochecklengths</b>	<p><b>nochecklengths</b> を指定した場合は、ソース・データの中にターゲット表の列のサイズを超える列定義がある場合であっても、各行のロードが試みられます。コード・ページ変換によってソース・データが縮小されれば、そのような行であったとしても正常にロードすることができます。例えば、ソースに 4 バイトの EUC データがある場合、それがターゲットで 2 バイトの DBCS データに縮小されれば、必要なスペースは半分で済みます。このオプションが特に役立つのは、列の定義は不一致であるがソース・データが常に適合することが分かっている場合です。</p>

## 注:

1. 日付形式ストリングは必ず二重引用符で囲まなければなりません。フィールド区切り文字には、a から z、A から Z、および 0 から 9 の文字を使用することはできません。フィールド区切り文字として、DEL ファイル・フォーマットの文字区切りまたはフィールド区切りと同じ文字を使用することはできません。エレメントの開始および終了位置が明らかな場合、フィールド区切り文字は任意指定です。あいまいさが生じうるのは、項目の長さが一定でない D、H、M、または S などのエレメントが使用されている場合です (修飾の仕方によって異なります)。

タイム・スタンプ・フォーマットの場合、月の記述子と分の記述子のどちらも文字 M を使用するため、区別があいまいにならないように注意する必要があります。月のフィールドは、他の日付フィールドと隣接していなければなりません。分フィールドは、他の時刻フィールドに隣接していなければなりません。あいまいなタイム・スタンプ形式の例を以下に示します。

## LOAD

"M" (月または分のどちらにもとれる)  
"M:M" (月と分の区別がつかない)  
"M:YYYY:M" (両方とも月と解釈される)  
"S:M:YYYY" (時刻値と日付値の両方に隣接している)

あいまいな場合、ユーティリティーはエラー・メッセージを報告し、操作は失敗します。

以下に、明確なタイム・スタンプ・フォーマットを示します。

"M:YYYY" (M (月))  
"S:M" (M (分))  
"M:YYYY:S:M" (M (月)...M (分))  
"M:H:YYYY:M:D" (M (分)...M (月))

二重引用符や円記号などの文字の前には、エスケープ文字 (例えば、¥) を付けなければなりません。

2. **charde1**、**colde1**、または **decpt** ファイル・タイプ修飾子に提供される文字値は、ソース・データのコード・ページで指定する必要があります。

文字コード・ポイント (文字記号ではない) は、xJJ または 0xJJ という構文で指定することができます (JJ はコード・ポイントの 16 進表記)。例えば、列区切りとして # 文字を指定するには、以下のステートメントのいずれかを使用します。

```
... modified by colde1# ...  
... modified by colde10x23 ...  
... modified by colde1X23 ...
```

3. 『データ移動のための区切り文字の制約事項』に、区切り文字の指定変更として使用できる文字に適用される制限のリストが示されています。
4. サポートされていないファイル・タイプを **MODIFIED BY** オプションで使用しようとしても、ロード・ユーティリティーは警告を出しません。この場合、ロード操作が失敗し、エラー・コードが戻されます。
5. **ignore**、**include**、**missing**、**override** が接尾部に付された複数の修飾子が指定されると、それらの修飾子はリストされた順に適用されます。以下のステートメントの場合、入力データに ID 列ではない暗黙的な非表示列のデータが含まれません。一方、暗黙的に非表示かどうかに関係なく、ID 列のデータは含まれません。

```
db2 load from delfile1 of del modified by  
    implicitlyhiddeninclude identitymissing insert into table1
```

しかし、以下のステートメントにおいてファイル・タイプ修飾子の順序を変更すると、入力データにすべての暗黙的な非表示列のデータ (非表示の ID 列も含む) が含まれることとなります。一方、暗黙的に非表示でない ID 列のデータは含まれません。

```
db2 load from delfile1 of del modified by  
    identitymissing implicitlyhiddeninclude insert into table1
```



表 37. codepage および usegraphiccodepage 使用時の LOAD 動作

codepage=N	usegraphiccodepage	LOAD の動作
なし	なし	<b>CLIENT</b> オプションの指定があっても、ファイル内のすべてのデータは、アプリケーション・コード・ページではなく、データベース・コード・ページであると見なされます
あり	なし	ファイル内のすべてのデータは、コード・ページ N であると見なされます。  <b>警告:</b> N が 1 バイト・コード・ページの場合、GRAPHIC データをデータベースにロードすると壊れます。
なし	あり	<b>CLIENT</b> オプションの指定があっても、ファイル内の文字データは、データベース・コード・ページであると見なされます。 <b>CLIENT</b> オプションの指定があっても、 <b>GRAPHIC</b> データは、データベース GRAPHIC データのコード・ページであると見なされます。  データベース・コード・ページが 1 バイトの場合は、すべてのデータはデータベース・コード・ページであると見なされます。  <b>警告:</b> 1 バイト・データベースに GRAPHIC データをロードすると、壊れます。
あり	あり	文字データは、コード・ページ N であると見なされま す。 GRAPHIC データは、N の GRAPHIC コード・ペ ージであると見なされます。  N が 1 バイトまたは 2 バイト・コード・ページの場合 は、すべてのデータは、コード・ページ N であると見な されます。  <b>警告:</b> N が 1 バイト・コード・ページの場合、GRAPHIC データをデータベースにロードすると壊れます。

## LOAD QUERY

処理中にロード操作の状況を調べ、表の状態を戻します。ロードが行われていない場合は、表の状態だけが戻されます。

このコマンドを正常に呼び出すためには、同じデータベースへの接続と、別の CLP セッションも必要になります。このコマンドは、ローカル・ユーザーでもリモート・ユーザーでも使用できます。

### 許可

なし

### 必要な接続

データベース

### コマンド構文

```

▶▶—LOAD QUERY—TABLE—table-name—┬─TO—local-message-file—┬─NOSUMMARY—
└──────────────────────────────────┘└──────────────────┘
└──────────────────────────────────┘└──────────────────┘
└──────────────────────────────────┘
└──SHOWDELTA──┘

```

### コマンド・パラメーター

#### TABLE *table-name*

データが現在ロード中の表の名前を指定します。非修飾の表名を指定すると、その表は CURRENT SCHEMA で修飾されます。

#### TO *local-message-file*

ロード操作中に生じ得る警告およびエラー・メッセージの宛先を指定します。このファイルは、LOAD コマンド用に指定された *message-file* であってはなりません。ファイルが既に存在する場合、ロード・ユーティリティーが生成するメッセージはすべてそのファイルに追加されます。

#### NOSUMMARY

ロード・サマリー情報 (読み取られた行、スキップされた行、ロードされた行、リジェクトされた行、削除された行、コミットされた行、警告の数) のレポートを生成しないよう指定します。

#### SUMMARYONLY

ロード・サマリー情報のレポートだけを生成するよう指定します。

#### SHOWDELTA

新しい情報 (LOAD QUERY コマンドの最後の呼び出し以後に発生したロード・イベントに関する) だけをレポートするよう指定します。

**例**

大量のデータを BILLYBOB データベース内の STAFF 表にロードしている場合、ロード操作の状況をチェックすることが必要になるかもしれません。ユーザーは次のように指定することができます。

```
db2 connect to billybob
db2 load query table staff to /u/mydir/staff.tempmsg
```

出力ファイル /u/mydir/staff.tempmsg は、次の出力のようになります。

```
SQL3501W The table space(s) in which the table resides will not be placed in
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. Input record count = "0".
```

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

```
SQL3519W Begin Load Consistency Point. Input record count = "104416".
```

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

```
SQL3519W Begin Load Consistency Point. Input record count = "205757".
```

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

```
SQL3519W Begin Load Consistency Point. Input record count = "307098".
```

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

```
SQL3519W Begin Load Consistency Point. Input record count = "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:
  Load in Progress
```

**使用上の注意**

ロード・ユーティリティは、ロックに加えて、表状態を使用して、表へのアクセスを制御します。 **LOAD QUERY** コマンドを使用して、表の状態を判別することができます。また、現在ロードされていない表に対しても、 **LOAD QUERY** を使用することができます。パーティション表の場合、報告される状態は、対応する可視のデータ・パーティションの状態のうち、最も限定的なものです。例えば、ある 1 つのデータ・パーティションが Read Access Only 状態にあり、他のすべてのデータ・パーティションは Normal 状態にある場合、ロード照会操作からは Read Access Only

状態が戻されます。ロード操作によって、データ・パーティションのサブセットが、表の残りとは異なる状態のままになることはありません。 **LOAD QUERY** で記述される表の状態は次のとおりです。

**通常** 表が通常以外の状態 (異常な状態) にない場合はこの状態になります。通常状態は、表が作成された後に置かれる最初の状態です。

### **SET INTEGRITY** ペンディング

表には、未確認の制約があります。この表に関して、**SET INTEGRITY** ペンディング (Set Integrity Pending) 状態を解除するには、**SET INTEGRITY** ステートメントを使用してください。ロード・ユーティリティーは、制約のある表でロード操作を開始する際に、表を **SET INTEGRITY** ペンディング (Set Integrity Pending) 状態にします。

### **Load in Progress**

これは、一時的な状態であり、ロード操作時のみ有効です。

### ロード・ペンディング

この表ではロード操作がアクティブでしたが、データがコミットできるようになる前に打ち切られました。表をこの状態から解除するには、**LOAD TERMINATE**、**LOAD RESTART**、または **LOAD REPLACE** コマンドを発行してください。

### 読み取りアクセスのみ

**ALLOW READ ACCESS** オプションを指定した場合、表はロード操作時にこの状態になります。「読み取りアクセスのみ」は一時的な状態です。この状態になっていると、他のアプリケーションやユーティリティーは、ロード操作より前に存在していたデータに読み取りアクセスすることができます。

### **REORG** ペンディング

**REORG** コマンドの推奨対象となる **ALTER TABLE** ステートメントが、表に対して実行されました。この表をもう一度アクセス可能にするには、まずクラシック **REORG** を実行する必要があります。

### 使用不可

表は使用できません。表のドロップまたはバックアップからのリストアのどちらかしか行えません。リカバリー不能のロード操作からロールフォワードを実行すると、表は使用できない状態になります。

### ロード再始動不可

表は部分的にロードされた状態になっているので、ロードの再始動操作は行えません。さらにこの表はロード・ペンディング状態にもなります。 **LOAD TERMINATE** または **LOAD REPLACE** コマンドを使用すると、表はロード再始動不可状態から解除されます。表がロード再始動不可状態になるのは、正常に再始動も終了もしないで失敗に終わったロード操作の後でロールフォワード操作を実行した場合か、または表がロード進行中状態またはロード・ペンディング状態であった間にとられたオンライン・バックアップからリストア操作が実行された場合です。どちらの場合も、ロード再始動操作に必要な情報は信頼性に欠けるため、ロード再始動不可状態では、ロードの再始動操作はできません。

**不明** **LOAD QUERY** コマンドは、表の状態を判別できません。

IBM DB2 データベース製品によってサポートされる表または表スペースの状態は、現在少なくとも 25 種類あります。それらの状態は、特定の環境下でのデータへのアクセスを制御したり、必要に応じて特定のユーザー・アクションを引き出して、データベースの整合性を保護するために使用されます。そのほとんどは、DB2 データベース・ユーティリティーのいずれか（例えばロード・ユーティリティーやバックアップおよびリストア・ユーティリティー）の操作に関連したイベントの結果として発生します。

ロード操作の前に従属表スペースの静止状態（静止とは永続的なロックのこと）は解除されますが、表スペースのロード中状態のときは、ロード操作時に従属表のバックアップを行うことができません。表スペースのロード中状態は、表のロード中状態と異なります。すべてのロード操作で表のロード中状態が使用されますが、**COPY NO** オプションを指定した（リカバリー可能なデータベースに対する）ロード操作でも表スペースのロード中状態が使用されます。

次の表では、サポートされているそれぞれの表の状態について説明しています。また、実施例も示し、データベースの管理中に遭遇する可能性のある状態の解釈および対応の仕方を正確に示しています。これらの例は、AIX で実行されたコマンド・スクリプトから取られたものです。コピーして貼り付け、実行することができます。UNIX 以外のシステムで DB2 製品を実行している場合は、ご使用のシステムで正しい形式のパス名となるようにしてください。例のほとんどは、DB2 データベース製品に付属する SAMPLE データベースの表に基づいています。SAMPLE データベースの一部ではないシナリオを必要とする例も少数ながら存在しますが、開始点としては SAMPLE データベースへの接続を使用することができます。

表 38. サポートされる表の状態

状態	例
ロード・ペンディング	<p>大量のデータ（例えば 20000 以上のレコード）を持つロード入力ファイル staffdata.del があるとします。ロード操作のターゲット表（NEWSTAFF という名前の新しい表）を含む小さな表スペースを作成します。</p> <pre>connect to sample; create tablespace ts1 managed by database using (file '/home/melnyk/melnyk/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><b>LOAD QUERY</b> コマンドによって戻される情報は、NEWSTAFF 表がロード・ペンディング状態にあることを示します。ロード終了操作の後、表は通常状態になります。</p>

## LOAD QUERY

表 38. サポートされる表の状態 (続き)

状態	例
Load in Progress	<p>大量のデータ (例えば 20000 以上のレコード) を持つロード入力ファイル staffdata.del があるとします。</p> <pre>connect to sample; create table newstaff like staff; load from staffdata.del of del insert into newstaff;</pre> <p>ロード操作の実行中に、別のセッションから次のスクリプトを実行します。</p> <pre>connect to sample; load query table newstaff; connect reset;</pre> <p><b>LOAD QUERY</b> コマンドによって戻される情報は、NEWSTAFF 表がロード中状態にあることを示します。</p>
通常	<pre>connect to sample; create table newstaff like staff; load query table newstaff;</pre> <p><b>LOAD QUERY</b> コマンドによって戻される情報は、NEWSTAFF 表が通常状態にあることを示します。</p>
ロード再始動不可	<p>大量のデータ (例えば 20000 以上のレコード) を持つロード入力ファイル staffdata.del があるとします。</p> <pre>update db cfg for sample using logarchmeth1 logretain; backup db sample; connect to sample; create tablespace ts1 managed by database using (file '/home/melnyk/melnyk/NODE0000 /SQL00001/ts1c1' 256); create table newstaff like staff in ts1; connect reset; backup db sample;</pre> <p>このバックアップ・イメージのタイム・スタンプは 20040629205935 です。</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 taken at 20040629205935; rollforward db sample to end of logs and stop; connect to sample; load query table newstaff; connect reset;</pre> <p><b>LOAD QUERY</b> コマンドによって戻される情報は、NEWSTAFF 表がロード再始動不可およびロード・ペンディング状態にあることを示します。</p> <pre>connect to sample; load from staffdata.del of del terminate into newstaff copy yes to /home/melnyk/backups; load query table newstaff; connect reset;</pre> <p><b>LOAD QUERY</b> コマンドによって戻される情報は、NEWSTAFF 表が通常状態になったことを示します。</p>

表 38. サポートされる表の状態 (続き)

状態	例
読み取りアクセスのみ	<p>大量のデータ (例えば 20000 以上のレコード) を持つロード入力ファイル staffdata.del があるとします。</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>ロード操作の実行中に、別のセッションから次のスクリプトを実行します。</p> <pre>connect to sample; load query table newstaff; select * from newstaff; connect reset;</pre> <p><b>LOAD QUERY</b> コマンドによって戻される情報は、NEWSTAFF 表が「読み取りアクセスのみ」および「ロード中」状態にあることを示します。照会は、STAFF 表のエクスポートされた内容のみを戻します。これは、ロード操作の前に NEWSTAFF 表に存在していたデータです。</p>
SET INTEGRITY ベンディング	<p>次の内容を持つロード入力ファイル staff_data.del があるとします。</p> <pre>11,"Melnyk",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><b>LOAD QUERY</b> コマンドによって戻される情報は、STAFF 表が SET INTEGRITY ベンディング状態にあることを示します。</p>
使用不可	<p>次の内容を持つロード入力ファイル staff_data.del があるとします。</p> <pre>11,"Melnyk",20,"Sales",10,70000,15000:</pre> <pre>update db cfg for sample using logarchmeth1 logretain; backup db sample;</pre> <p>このバックアップ・イメージのタイム・スタンプは 20040629182012 です。</p> <pre>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><b>LOAD QUERY</b> コマンドによって戻される情報は、STAFF 表が使用不可状態にあることを示します。</p>

ロード操作の進行状況は、**LISTUTILITIES** コマンドを使ってモニターすることもできます。

## MIGRATE DATABASE

旧バージョンの DB2データベースを、このインスタンスで実行されているリリースに対応する形式に変換します。

このコマンドは推奨されておらず、今後のリリースでは廃止されます。代わりに **UPGRADE DATABASE** コマンドを使用する必要があります。

### 許可

SYSADM

### 必要な接続

このコマンドは、データベース接続を確立します。

### コマンド構文

```
▶▶ MIGRATE DATABASE database-alias
    DB
USER username
    USING password
```

### コマンド・パラメーター

**DATABASE** *database-alias*

現在インストールされているデータベース・マネージャーのバージョンにマイグレーションするデータベースの別名を指定します。

**USER** *username*

データベースをマイグレーションするのに使うユーザー名を指定します。

**USING** *password*

ユーザー名を認証するために使用するパスワード。ユーザー名を指定してパスワードを省略すると、ユーザーに入力を求めるプロンプトが出ます。

### 使用上の注意

**UPGRADE DATABASE** コマンドの資料を参照してください。



## PING

クライアントと接続済みデータベース・サーバー間の基礎接続のネットワーク応答時間をテストします。

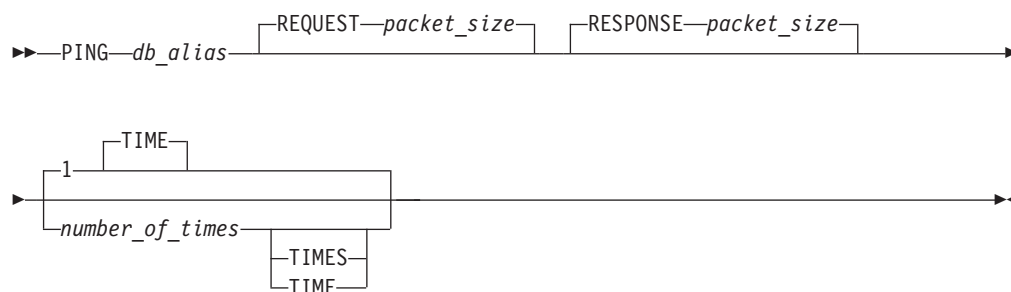
### 許可

なし

### 必要な接続

データベース

### コマンド構文



### コマンド・パラメーター

#### *db\_alias*

`ping` 送信先の DRDA サーバー上のデータベースのデータベース別名を指定します。このパラメーターは必須ですが現在使用されていません。将来の利用のために予約してあります。有効なデータベース別名を指定します。

#### **REQUEST** *packet\_size*

サーバーに送信するパケットのサイズをバイト単位で指定します。サイズは 0 から 32767 の範囲でなければなりません。デフォルトは 10 バイトです。このオプションは、DB2 for Linux, UNIX, and Windows バージョン 8 以降、または DB2 Universal Database for z/OS バージョン 8 以降を実行しているサーバーでのみ有効です。

#### **RESPONSE** *packet\_size*

クライアントに戻るパケットのサイズをバイト単位で指定します。サイズは 0 から 32767 の範囲でなければなりません。デフォルトは 10 バイトです。このオプションは、DB2 for Linux, UNIX, and Windows バージョン 8 以降、または DB2 UDB for z/OS バージョン 8 以降を実行しているサーバーでのみ有効です。

#### *number\_of\_times*

このテストの反復数を指定します。この値は 1 から 32767 の範囲でなければなりません。デフォルトは 1 です。1 個のタイミングが、反復ごとに戻ります。

## 例

### 例 1

ホスト・データベース `hostdb` への 1 回の接続に対するネットワーク応答時間をテストするには、次のように行います。

```
db2 ping hostdb 1
または
db2 ping hostdb
```

コマンドは、次の例と類似した出力を表示します。

```
Elapsed time: 7221 microseconds
```

### 例 2

ホスト・データベース `hostdb` への 5 回の接続に対するネットワーク応答時間をテストするには、次のように行います。

```
db2 ping hostdb 5
または
db2 ping hostdb 5 times
```

コマンドは、次の例と類似した出力を表示します。

```
Elapsed time: 8412 microseconds
Elapsed time: 11876 microseconds
Elapsed time: 7789 microseconds
Elapsed time: 10124 microseconds
Elapsed time: 10988 microseconds
```

### 例 3

100 バイトの `REQUEST` パケットおよび 200 バイトの `RESPONSE` パケットを使用して、ホスト・データベース `hostdb` への接続のネットワーク応答時間をテストするには、次のようにします。

```
db2 ping hostdb request 100 response 200
または
db2 ping hostdb request 100 response 200 1 time
```

## 使用上の注意

データベース接続は、このコマンドを呼び出す前に存在している必要があります。存在していない場合、エラーが起きます。

返される経過時間は、IBM Data Server Client と DB2 サーバーの間の接続に関するものです。

このコマンドは、DB2 Universal Database バージョン 7 クライアントから、DB2 Connect バージョン 8 経由で DB2 ホスト・データベース・サーバーへの接続に使用された場合、機能しません。

## PRECOMPILE

組み込み SQL ステートメントを含むアプリケーション・プログラム・ソース・ファイル処理します。SQL ステートメントに対するホスト言語呼び出しを含む変更後のソース・ファイルが作成されます。また、デフォルトとして、データベース内にパッケージが作成されます。

### 有効範囲

このコマンドは、db2nodes.cfg 中のどのデータベース・パーティションからでも発行できます。パーティション・データベース環境では、これは db2nodes.cfg ファイルで定義されたどのデータベース・パーティション・サーバーからでも出すことができます。実行すると、カタログ・データベース・パーティションのデータベース・カタログが更新されます。その影響はすべてのデータベース・パーティションから見えます。

### 許可

以下のいずれかの権限が必要です。

- DBADM 権限
- EXPLAIN ONLY が指定されている場合、EXPLAIN 権限または暗黙的に EXPLAIN を含む権限があれば問題ありません。
- パッケージが存在しない場合は、BINDADD 権限および以下のものがが必要です。
  - パッケージのスキーマ名が存在しない場合は、データベースに対する IMPLICIT\_SCHEMA 権限。
  - パッケージのスキーマ名が存在する場合は、スキーマに対する CREATEIN 特権。
- パッケージが存在する場合は、以下のいずれかの特権です。
  - スキーマに対する ALTERIN 特権
  - パッケージに対する BIND 特権

さらに、EXPLAIN または EXPLSNAP 節を使用して Explain 情報をキャプチャーする場合は、以下のいずれかの権限が必要です。

- Explain 表に対する INSERT 特権
- DATAACCESS 権限

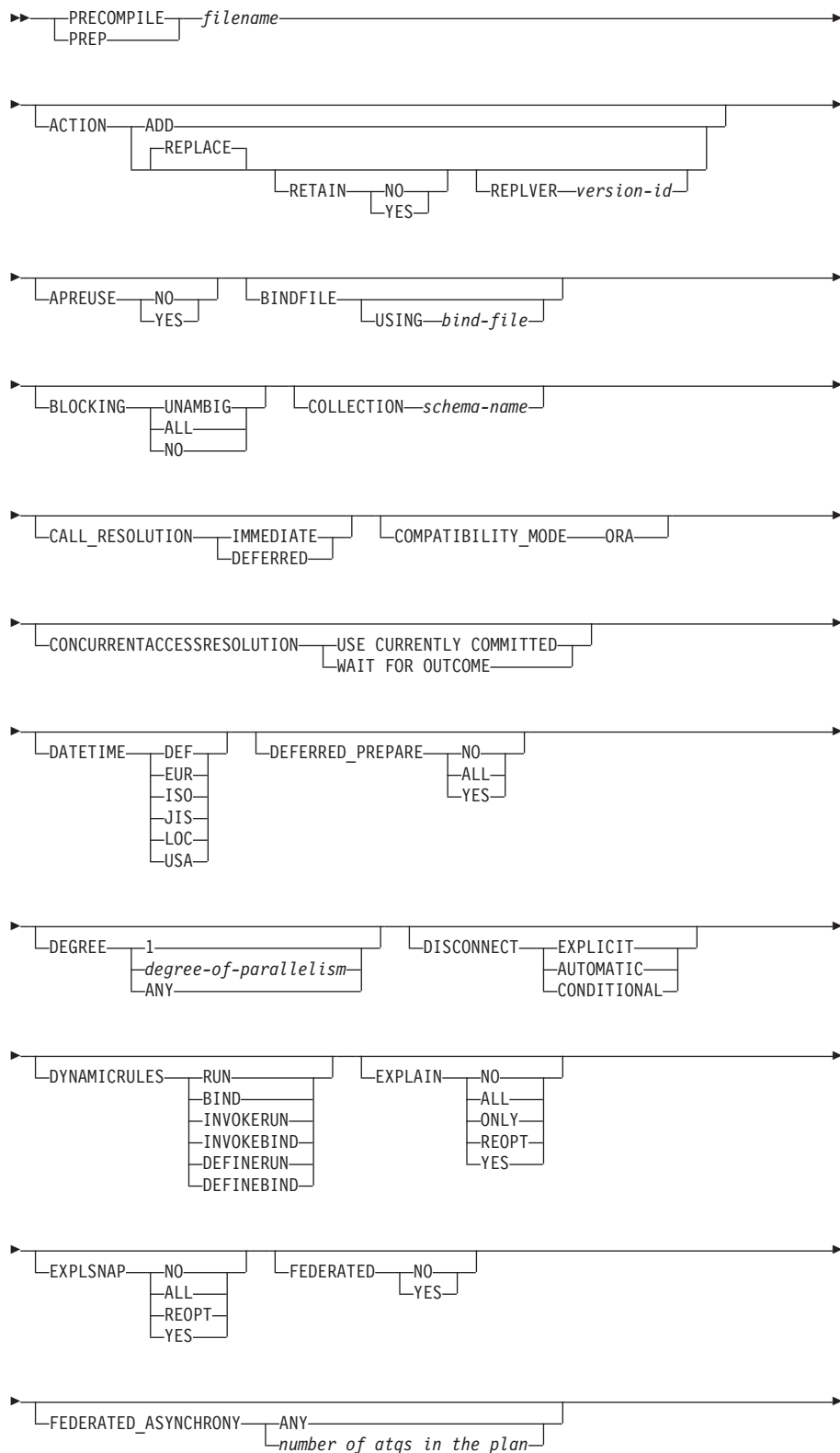
アプリケーション内の静的 SQL ステートメントをコンパイルするために必要な特権もすべて必要です。グループに認可された特権が、静的ステートメントの許可の検査に使用されることはありません。

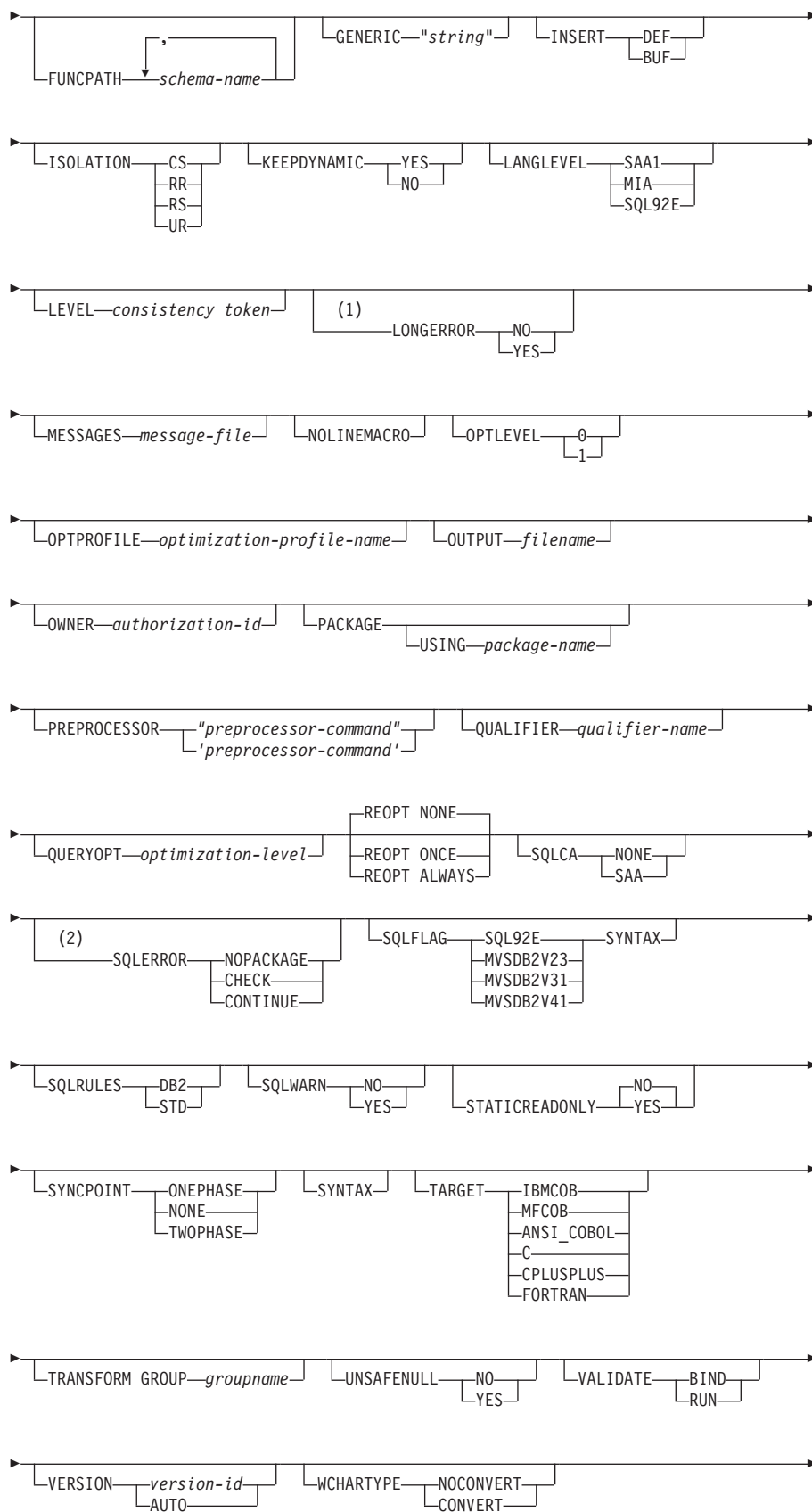
### 必要な接続

データベース。暗黙接続が可能な場合には、デフォルト・データベースへの接続が確立されます。

## コマンド構文

DB2 for Linux, UNIX, and Windows の場合





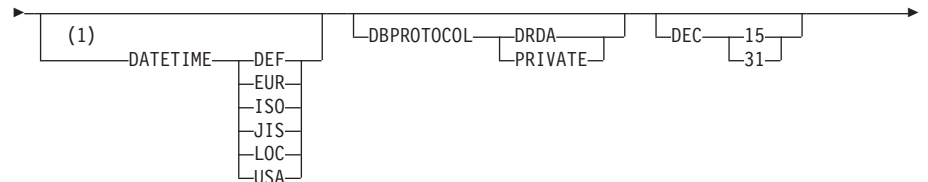
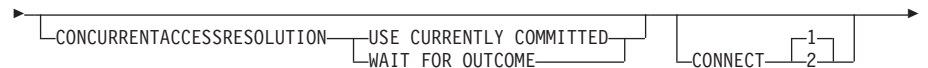
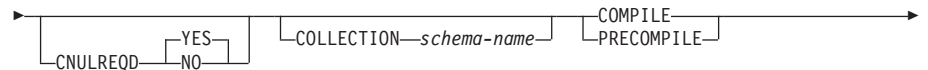
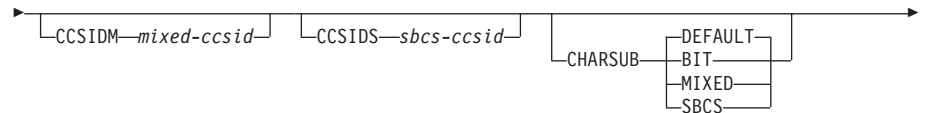
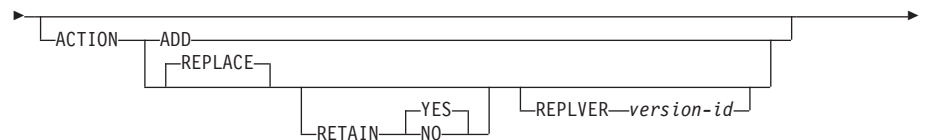
# PRECOMPILE

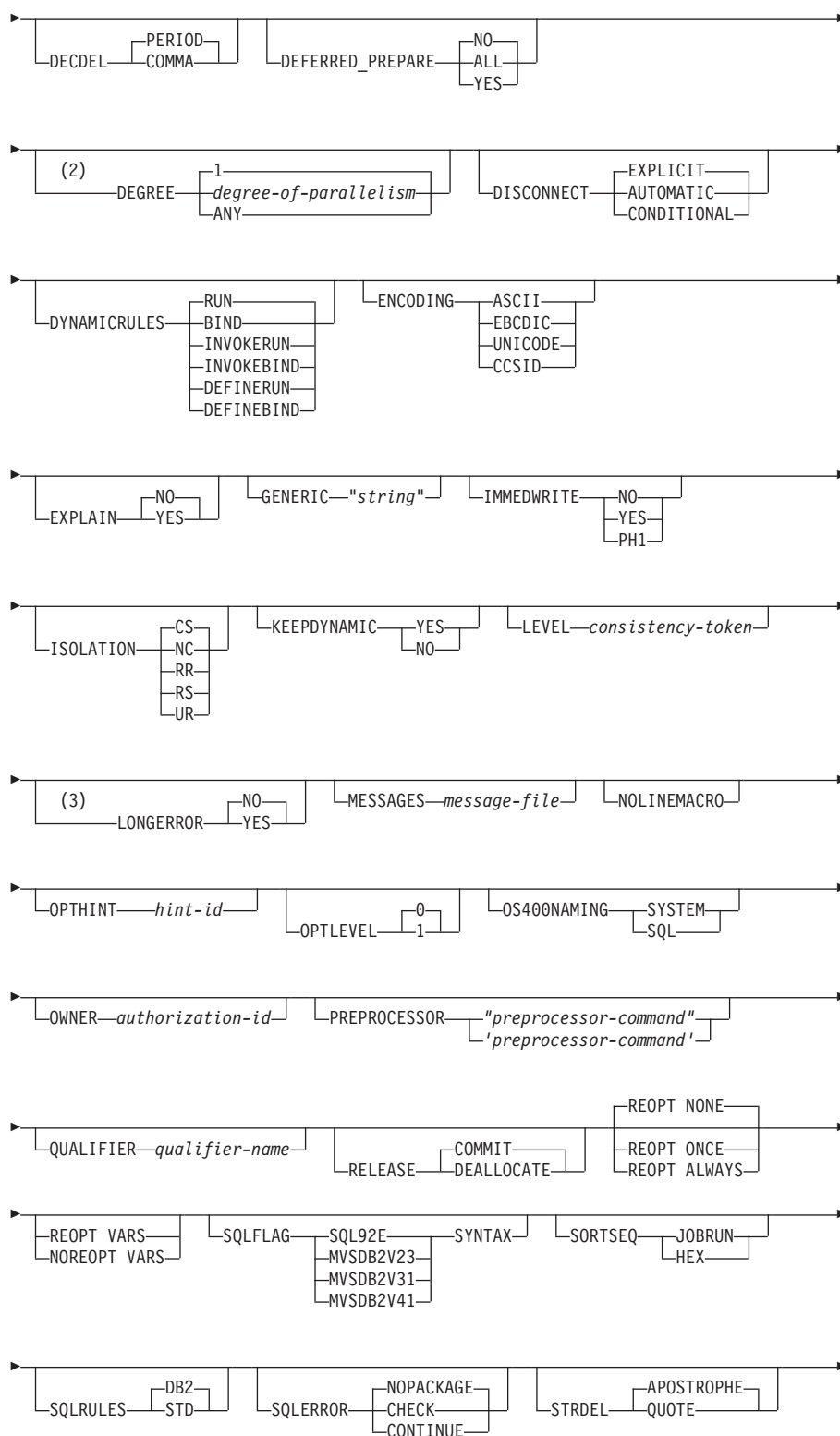


注:

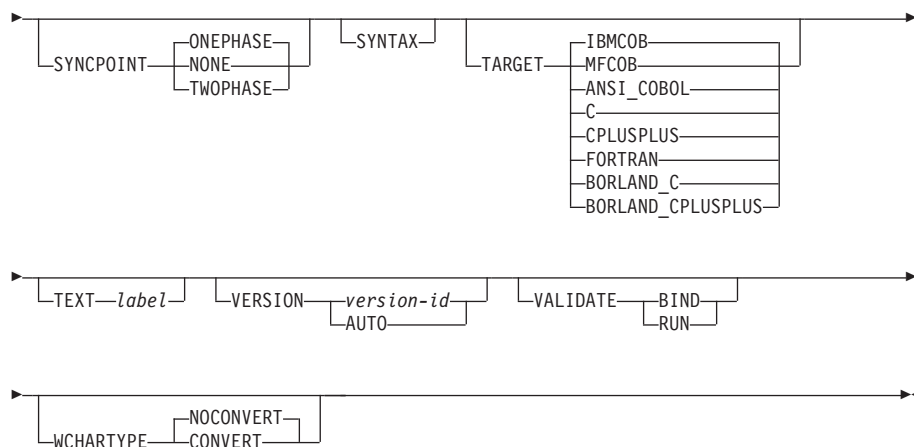
- 1 NO は、32 ビット・システムと 64 ビット Windows システムのデフォルトです。それらのシステムでは、長いホスト変数を INTEGER 列の宣言として使用できます。YES は、64 ビット UNIX システムのデフォルトです。
- 2 SYNTAX は SQLERROR(CHECK) の同義語です。

## Linux、Windows、UNIX 以外のサーバー上の DB2 データベースの場合





## PRECOMPILE



### 注:

- 1 サーバーが **DATETIME DEF** オプションをサポートしない場合、それは **DATETIME ISO** にマップされます。
- 2 **DEGREE** オプションは DRDA レベル 2 のアプリケーション・サーバーでしかサポートされていません。
- 3 **NO** は、32 ビット・システムと 64 ビット Windows システムのデフォルトです。それらのシステムでは、長いホスト変数を **INTEGER** 列の宣言として使用できます。**YES** は、64 ビット UNIX システムのデフォルトです。

## コマンド・パラメーター

### *filename*

プリコンパイルするソース・ファイルを指定します。拡張子の指定は、以下のようになります。

- C アプリケーションの場合、**.sqc** を指定します (**.c** ファイルが生成されます)。
- C++ アプリケーションの場合、**.sqx** (Windows オペレーティング・システム)、または **.sqC** (UNIX および Linux オペレーティング・システム) を指定します (Windows オペレーティング・システムの場合は **.cxx** ファイル、UNIX および Linux オペレーティング・システムの場合は **.C** ファイルが作成されます)。
- COBOL アプリケーションの場合、**.sqb** を指定します (**.cb1** ファイルが生成されます)。
- FORTRAN アプリケーションの場合、**.sqf** を指定します (Windows オペレーティング・システムの場合は **.for** ファイル、UNIX および Linux オペレーティング・システムの場合は **.f** ファイルが生成されます)。

UNIX および Linux オペレーティング・システムにおいて、組み込み SQL を含む C++ アプリケーションの場合に望ましい拡張子は **sqC** です。しかし、UNIX および Linux オペレーティング・システムでは、大文字小文字を区別しないシステムのための **sqx** 規則も通用します。

### ACTION

パッケージを追加または置換できるかどうかを示します。



**ADD** 名前付きパッケージが存在せず、新規パッケージを作成するということを指示します。既にパッケージがある場合は、実行停止状態となり、診断エラー・メッセージが戻されます。

#### REPLACE

既存のパッケージを、パッケージ名および作成者が同じ新規パッケージと置き換えることを指示します。これは **ACTION** オプションのデフォルト値です。

#### RETAIN

パッケージを置き換えたときに EXECUTE 権限が保持されるかどうかを指示します。パッケージの所有権を変更した場合、新規所有者は前のパッケージ所有者に BIND 権限と EXECUTE 権限を付与します。

**NO** パッケージを置き換えたとき、EXECUTE 権限を保持しません。この値は DB2 ではサポートされていません。

**YES** パッケージを置き換えたとき、EXECUTE 権限を保持します。これはデフォルト値です。

#### REPLVER *version-id*

特定のバージョンのパッケージを置き換えます。バージョン ID は、どのバージョンのパッケージを置き換えるのかを指定するものです。指定されたバージョンが存在しない場合には、エラーが戻されます。REPLACE の REPLVER オプションが指定されていない場合、プリコンパイルされるパッケージのパッケージ名、およびバージョンと一致するパッケージが既に存在すれば、そのパッケージは置換されます。存在しなければ、新規のパッケージが追加されます。

#### APREUSE

静的 SQL アクセス・プランが再使用されるかどうかを指定します。このオプションが有効である場合、照会コンパイラーは、バインド時および将来の暗黙および明示の再バインド時に、既存のパッケージ内のステートメントにアクセス・プランを再使用しようとします。

**YES** 照会コンパイラーは、パッケージ内のステートメントにアクセス・プランを再使用しようとします。既存のパッケージがある場合、照会コンパイラーは、新規バインド・ファイル内のステートメントと合致できるすべてのステートメントに対して、アクセス・プランを再使用しようとします。ステートメントが合致するには、ステートメント・テキストは同一でなければならない、既存のパッケージ内のステートメントのセクション番号は、新規パッケージ内のステートメントに対するセクション番号の内容と一致しなければなりません。

**NO** 照会コンパイラーは、パッケージ内のステートメントにアクセス・プランを再使用しようとしません。これはデフォルト設定です。

#### BINDFILE

バインド・ファイルが作成されます。package オプションも指定されてい

なければ、パッケージは作成されません。次に示す例のように、バインド・ファイルを要求したのにパッケージが作成されないなら、

```
db2 prep sample.sqc bindfile
```

オブジェクトの存在と許可 `SQLCODE` はエラーではなく警告として扱われます。それで、プリコンパイルに使用されるデータベースに、アプリケーション内の静的 `SQL` ステートメントで参照されているすべてのオブジェクトが入っているわけではない場合でも、バインド・ファイルは正常に作成されます。必要なオブジェクトが作成されたなら、バインド・ファイルは正常にバインドされ、パッケージが作成されます。

### USING *bind-file*

プリコンパイラーが生成するバインド・ファイルの名前。ファイル名には、`.bnd` 拡張子が付いていなければなりません。ファイル名を入力しないなら、プリコンパイラーは、`filename` パラメーターとして入力されているプログラムの名前を使用し、それに `.bnd` 拡張子を付けてファイル名とします。パスを指定しないと、バインド・ファイルは現行ディレクトリーに作成されます。

## BLOCKING

カーソルの行ブロッキングのタイプを指定します。パーティション・データベース環境では、LOB 列データ・タイプ参照を含んだ行データのブロッキングもサポートされます。

**ALL** FOR READ ONLY 節で指定されているカーソルまたは FOR UPDATE として指定されていないカーソルの場合、ブロック化が発生します。

未確定カーソルは、読み取り専用として扱われます。

**NO** どのカーソルについてもブロック化は発生しません。

読み取り専用カーソルおよび未確定カーソルの定義については、`DECLARE CURSOR` ステートメントを参照してください。

あいまいなカーソルは、更新可能として扱われます。

## UNAMBIG

FOR READ ONLY 節で指定されているカーソルの場合、ブロック化が発生します。

FOR READ ONLY または FOR UPDATE 節で宣言されていない、未確定でない読み取り専用のカーソルはブロック化されます。未確定カーソルはブロック化されません。

あいまいなカーソルは、更新可能として扱われます。

## CALL\_RESOLUTION

設定されている場合、`CALL_RESOLUTION DEFERRED` オプションは使用すべきでない `sqlproc()` API の起動として、`CALL` ステートメントが実行されることを示します。設定されていないか、または `IMMEDIATE` が設定されている場合、`CALL` ステートメントは通常の `SQL` ステートメントとして実行されます。`SQL0204` が出されるのは、`CALL_RESOLUTION IMMEDIATE` を指定した `CALL` ステートメントのプロシーチャーをプリコンパイラーが解決できなかった場合です。

**CCSIDG** *double-ccsid*

CREATE および ALTER TABLE SQL ステートメントの文字カラム定義で、2 バイト文字用のコード化文字セット ID (CCSID) (特定の CCSID 節は使用しない) を指定する整数。このオプションは、DB2 for Linux, UNIX, and Windows 用のサーバーではサポートされません。このオプションを指定しないと、DRDA サーバーは、組み込みのデフォルト値を使用します。

**CCSIDM** *mixed-ccsid*

CREATE および ALTER TABLE SQL ステートメントの文字カラム定義で、混合バイト文字用のコード化文字セット ID (CCSID) (特定の CCSID 節は使用しない) を指定する整数。このオプションは、DB2 for Linux, UNIX, and Windows 用のサーバーではサポートされません。このオプションを指定しないと、DRDA サーバーは、組み込みのデフォルト値を使用します。

**CCSIDS** *sbc-ccsid*

CREATE および ALTER TABLE SQL ステートメントの文字カラム定義で、1 バイト文字用のコード化文字セット ID (CCSID) (特定の CCSID 節は使用しない) を指定する整数。このオプションは、DB2 for Linux, UNIX, and Windows 用のサーバーではサポートされません。このオプションを指定しないと、DRDA サーバーは、組み込みのデフォルト値を使用します。

**CHARSUB**

CREATE および ALTER TABLE SQL ステートメントの列定義に使用する、デフォルトの文字サブタイプを指定します。このプリコンパイル/バインド・オプションは、DB2 for Linux, UNIX, and Windows 用のサーバーではサポートされません。

**BIT** 明示的にサブタイプを指定しなかった場合、すべての新規文字カラムに FOR BIT DATA SQL 文字サブタイプが使用されます。

**DEFAULT**

明示的にサブタイプを指定しなかった場合、すべての新規文字カラムにターゲットの組み込みデフォルト・サブタイプが使用されます。

**MIXED**

明示的にサブタイプを指定しなかった場合、すべての新規文字カラムに FOR MIXED DATA SQL 文字サブタイプが使用されます。

**SBCS** 明示的にサブタイプを指定しなかった場合、すべての新規文字カラムに FOR SBCS DATA SQL 文字サブタイプが使用されます。

**CNULREQD**

このオプションは、**LANGLEVEL** プリコンパイル・オプションと関連します。これは、C または C++ アプリケーションで作成されたバインド・ファイルの場合のみ有効です。このバインド・オプションは、DB2 for Linux, UNIX, and Windows 用のサーバーではサポートされません。

**NO** C ストリング・ホスト変数中の NULL 終止符に関して、**LANGLEVEL SAA1** プリコンパイル・オプションに基づいてアプリケーションがコード化された場合です。

## PRECOMPILE

**YES** C ストリング・ホスト変数中の NULL 終止符に関して、**LANGLEVEL** MIA プリコンパイル・オプションに基づいてアプリケーションがコード化された場合です。

### COLLECTION *schema-name*

パッケージ用の 128 バイトの収集 ID を指定します。これを指定しなかった場合、パッケージを処理する際には、ユーザーの許可 ID が使用されません。

### COMPATIBILITY\_MODE

**ORA** DB2 データベース・マネージャーは、他のデータベース・システムからの組み込み SQL C アプリケーションのマイグレーションを容易にするフィーチャーを提供します。これらの互換性フィーチャーは、**COMPATIBILITY\_MODE** を ORA に設定することによって使用可能にすることができます。例えば、以下のコマンドを使用すると、**tbse1.sqc** ファイルをコンパイルする際に互換フィーチャーを使用可能にすることができます。

```
$ db2 PRECOMPILE tbse1.sqc BINDFILE COMPATIBILITY_MODE ORA
```

### CONCURRENTACCESSRESOLUTION

パッケージ内のステートメントに使用する並行アクセス解決を指定します。

### USE CURRENTLY COMMITTED

データベース・マネージャーが更新または削除の処理中である場合に、該当するスキャンに現在コミット済みバージョンのデータを使用できることを指定します。挿入処理中の行はスキップできます。この節は、有効な分離レベルがカーソル固定または読み取り固定である場合に適用され (読み取り固定の場合、それはコミットされていない挿入だけをスキップします)、そうでない場合は無視されません。該当するスキャンには、非読み取り専用ステートメントに加え、読み取り専用ステートメントの一部にできる読み取り専用スキャンが含まれます。レジストリー変数 **DB2\_EVALUNCOMMITTED**、**DB2\_SKIPDELETED**、および **DB2\_SKIPINSERTED** の設定は、適用されなくなりました。

### WAIT FOR OUTCOME

カーソル固定およびさらに上位のスキャンで、更新処理中または削除処理中のデータが検出されたときにはコミットまたはロールバックを待機するように指定します。挿入処理中の行はスキップされません。レジストリー変数 **DB2\_EVALUNCOMMITTED**、**DB2\_SKIPDELETED**、および **DB2\_SKIPINSERTED** の設定は、適用されなくなりました。

### CONNECT

- 1 **CONNECT** ステートメントをタイプ 1 の **CONNECT** として処理するよう指定します。
- 2 **CONNECT** ステートメントをタイプ 2 の **CONNECT** として処理するよう指定します。

### DATETIME

使用する日時形式を指定します。

- DEF** データベースのテリトリー・コードと対応する日時形式を使用します。
- EUR** IBM 欧州規格の日時形式を使用します。
- ISO** 国際標準化機構規格の日時形式を使用します。
- JIS** 日本工業規格の日時形式を使用します。
- LOC** データベースのテリトリー・コードと対応する地域別日時形式を使用します。
- USA** IBM 米国規格の日時形式を使用します。

**DBPROTOCOL**

3 部構成の名前のステートメントによって識別されるリモート・サイトに接続するとき使用するプロトコルを指定します。サポートしているのは DB2 for OS/390 だけです。サポートされているオプション値のリストについては、DB2 for OS/390 の資料を参照してください。

- DEC** 10 進算術演算に使用する最大精度を指定します。このプリコンパイル・オプションおよびバインド・オプションは、DB2 for Linux, UNIX, and Windows のサーバーではサポートされません。このオプションを指定しないと、DRDA サーバーは、組み込みのデフォルト値を使用します。
- 15** 10 進算術演算に 15 桁精度が使用されます。
- 31** 10 進算術演算に 31 桁精度が使用されます。

**DECDEL**

10 進数および浮動小数点リテラル中で小数点標識としてピリオド (.) またはコンマ (,) のどちらかを指定します。このプリコンパイル/バインド・オプションは、DB2 for Linux, UNIX, and Windows 用のサーバーではサポートされません。このオプションを指定しないと、DRDA サーバーは、組み込みのデフォルト値を使用します。

**COMMA**

小数点標識としてコンマ (,) を使用します。

**PERIOD**

小数点標識としてピリオド (.) を使用します。

**DEFERRED\_PREPARE**

DB2 共通サーバー・データベースまたは DRDA データベースにアクセスする際のパフォーマンス機能を強化します。このオプションは、SQL PREPARE ステートメントを、それに関連した OPEN、DESCRIBE、または EXECUTE ステートメント・フローと結合して、プロセス間またはネットワーク・フローを最小にします。

- NO** PREPARE ステートメントは、それが発行された時点で実行されます。

- YES** PREPARE ステートメントの実行は、対応する OPEN、DESCRIBE、または EXECUTE ステートメントが発行されるまで据え置かれます。

PREPARE ステートメントは、INTO 節 (SQLDA がすぐに戻されることを必要とする) を使用する場合には据え置かれませんが、

し、パラメーター・マーカを使用しないカーソルについて  
PREPARE INTO ステートメントが発行された場合には、PREPARE  
の実行時にカーソルを事前オープンすることによって、処理が最適  
化されます。

**ALL** PREPARE INTO ステートメントも据え置かれること以外は、YES  
と同じです。PREPARE ステートメントが SQLDA を戻すために  
INTO 節を使用する場合、アプリケーションでは、  
OPEN、DESCRIBE、または EXECUTE ステートメントが発行さ  
れ、戻されるまで、この SQLDA の内容を参照してはなりません。

### DEGREE

SMP システムで静的 SQL ステートメントを実行するための並列処理の度  
合いを指定します。このオプションは、CREATE INDEX 並列処理には影響  
を与えません。

**1** ステートメントの実行に並列処理を使用しません。

#### *degree-of-parallelism*

ステートメントを実行する際の並列処理の度合いを指定します。値  
の範囲は 2 から 32 767 です。

**ANY** ステートメントの実行時にデータベース・マネージャーで判別した  
程度で並列処理を行うよう指定します。

### DISCONNECT

#### AUTOMATIC

コミット時にすべてのデータベース接続を切断するよう指定しま  
す。

#### CONDITIONAL

RELEASE をマークしたか、またはオープン状態の WITH HOLD  
カーソルをもたないデータベース接続を、コミット時に切断するよ  
う指定します。

#### EXPLICIT

RELEASE ステートメントで明示的に解放をマークしたデータベー  
ス接続だけを、コミット時に切断するよう指定します。

### DYNAMICRULES

許可 ID に使用される値の初期設定、および非修飾オブジェクト参照の暗黙  
的な修飾の、実行時の動的 SQL に適用される規則を定義します。

**RUN** パッケージを実行するユーザーの許可 ID が動的 SQL ステートメ  
ントの許可検査に使用されるように指定します。許可 ID は、動的  
SQL ステートメント内の非修飾オブジェクト参照を暗黙的に修飾す  
るためのデフォルトのパッケージ修飾子としても使用されます。こ  
れはデフォルト値です。

**BIND** 許可および修飾の静的 SQL に適用されるすべての規則が、実行時  
に使用されるように指定します。つまり、パッケージ所有者の許可  
ID が動的 SQL の許可検査に使用され、デフォルトのパッケージ修  
飾子が動的 SQL ステートメント内の非修飾オブジェクト参照の暗  
黙的な修飾に使用されます。



**DEFINERUN**

パッケージがルーチン・コンテキスト内で使用される場合、ルーチン定義者の許可 ID が許可検査およびルーチン内の動的 SQL ステートメント内の非修飾オブジェクト参照の暗黙的な修飾に使用されます。

パッケージがスタンドアロン・アプリケーションとして使用される場合、動的 SQL ステートメントはパッケージが **DYNAMICRULES RUN** を指定してバインドされたかのように処理されます。

**DEFINEBIND**

パッケージがルーチン・コンテキスト内で使用される場合、ルーチン定義者の許可 ID が許可検査およびルーチン内の動的 SQL ステートメント内の非修飾オブジェクト参照の暗黙的な修飾に使用されます。

パッケージがスタンドアロン・アプリケーションとして使用される場合、動的 SQL ステートメントはパッケージが **DYNAMICRULES BIND** を指定してバインドされたかのように処理されます。

**INVOKERUN**

パッケージがルーチン・コンテキスト内で使用される場合、ルーチン起動時に有効だった現行のステートメント許可 ID が、動的 SQL ステートメントの許可検査およびそのルーチン内の動的 SQL ステートメント内の非修飾オブジェクト参照の暗黙的な修飾に使用されます。

パッケージがスタンドアロン・アプリケーションとして使用される場合、動的 SQL ステートメントはパッケージが **DYNAMICRULES RUN** を指定してバインドされたかのように処理されます。

**INVOKEBIND**

パッケージがルーチン・コンテキスト内で使用される場合、ルーチン起動時に有効だった現行のステートメント許可 ID が、動的 SQL ステートメントの許可検査およびそのルーチン内の動的 SQL ステートメント内の非修飾オブジェクト参照の暗黙的な修飾に使用されます。

パッケージがスタンドアロン・アプリケーションとして使用される場合、動的 SQL ステートメントはパッケージが **DYNAMICRULES BIND** を指定してバインドされたかのように処理されます。

動的 SQL ステートメントは、バインド動作を公開しているパッケージ内のパッケージ所有者の許可 ID を使用します。したがって、パッケージのユーザーが受け取るべきでない権限を、パッケージのバインド・プログラムに付与してはなりません。同様に、定義動作を公開するルーチンを定義するとき、パッケージのユーザーが受け取るべきでない権限を、ルーチンの定義者に付与してはなりません。動的ステートメントがルーチンの定義者の許可 ID を使用するためです。

次の動的な準備済み SQL ステートメントは、**DYNAMICRULES RUN** にバインドされなかったパッケージ内では使用できません。

## PRECOMPILE

GRANT、REVOKE、ALTER、CREATE、DROP、COMMENT  
ON、RENAME、SET INTEGRITY、および SET EVENT MONITOR STATE  
です。

### ENCODING

プランまたはパッケージ内の静的ステートメント内にあるすべてのホスト変数のエンコード方式を指定します。サポートしているのは DB2 for OS/390 だけです。サポートされているオプション値のリストについては、DB2 for OS/390 の資料を参照してください。

### EXPLAIN

各 SQL ステートメント用に選択したアクセス・プランに関する Explain 表の情報を、パッケージに保管します。

**NO** Explain 情報はキャプチャーされません。

**YES** Explain 表には、静的ステートメントの場合は prep/bind 時に、追加バインド・ステートメントの場合は実行時に、選択されたアクセス・プランについての情報が取り込まれます。

パッケージがルーチンに使用されるもので、パッケージに追加バインド・ステートメントが含まれる場合、そのルーチンは MODIFIES SQL DATA として定義されなければなりません。これが行われない場合、パッケージ内の追加バインド・ステートメントはランタイム・エラーを生じます (SQLSTATE 42985)。

### REOPT

再最適化可能な追加バインド SQL ステートメントの Explain 情報が、実行時に各 Explain 表に入れられます。さらに、CURRENT EXPLAIN MODE 特殊レジスターが NO に設定されている場合でも、再最適化可能な動的 SQL ステートメントの Explain 情報が実行時に収集されます。

パッケージがルーチンに使用される場合、そのルーチンは MODIFIES SQL DATA として定義されなければなりません。そうしなければ、パッケージ内の追加バインドおよび動的ステートメントはランタイム・エラーを生じます (SQLSTATE 42985)。

**ONLY ONLY** オプションにより、実行する特権がなくてもステートメントの EXPLAIN ができます。Explain 表にはデータが入りますが、永続的パッケージは作成されません。バインド処理中に、名前とバージョンが同じ既存パッケージを検出した場合、その既存パッケージはドロップも置換 (**ACTION REPLACE** を指定した場合) もされません。Explain 表の移植中にエラーが発生した場合、Explain 情報は、エラーを戻したステートメント、およびそれに続くどのステートメントに対しても追加されません。

**ALL** 適格な静的 SQL ステートメントの Explain 情報が、prep/bind 時に各 Explain 表に入れられます。適格な追加バインド SQL ステートメントの Explain 情報が、実行時に各 Explain 表に入れられます。さらに、CURRENT EXPLAIN MODE 特殊レジスターが NO に設定されている場合でも、適格な動的 SQL ステートメントの Explain 情報が実行時に収集されます。



パッケージがルーチンに使用される場合、そのルーチンは **MODIFIES SQL DATA** として定義されなければなりません。そうしなければ、パッケージ内の追加バインドおよび動的ステートメントはランタイム・エラーを生じます (SQLSTATE 42985)。

### EXPLSNAP

**Explain** 表に **Explain** スナップショットを保管します。

**NO** **Explain** スナップショットはキャプチャーされません。

**YES** **Explain** 表には、静的ステートメントの場合は **prep/bind** 時に、追加バインド・ステートメントの場合は実行時に、適格な各静的 SQL ステートメントの **Explain** スナップショットが入れられます。

パッケージがルーチンに使用されるもので、パッケージに追加バインド・ステートメントが含まれる場合、そのルーチンは **MODIFIES SQL DATA** として定義されなければなりません。これが行われない場合、パッケージ内の追加バインド・ステートメントはランタイム・エラーを生じます (SQLSTATE 42985)。

### REOPT

再最適化可能な追加バインド SQL ステートメントの **Explain** スナップショット情報が、実行時に各 **Explain** 表に入れられます。さらに、**CURRENT EXPLAIN SNAPSHOT** 特殊レジスターが **NO** に設定されている場合でも、再最適化可能な動的 SQL ステートメントの **Explain** スナップショット情報が実行時に収集されます。

パッケージがルーチンに使用される場合、そのルーチンは **MODIFIES SQL DATA** として定義されなければなりません。そうしなければ、パッケージ内の追加バインドおよび動的ステートメントはランタイム・エラーを生じます (SQLSTATE 42985)。

**ALL** 適格な各静的 SQL ステートメントの **Explain** スナップショットが、準備またはバインド時に **Explain** 表に入れられます。適格な追加バインド SQL ステートメントの **Explain** スナップショット情報が、実行時に各 **Explain** 表に入れられます。さらに、**CURRENT EXPLAIN SNAPSHOT** 特殊レジスターが **NO** に設定されている場合でも、適格な動的 SQL ステートメントの **Explain** スナップショット情報が実行時に収集されます。

パッケージがルーチンに使用される場合、そのルーチンは **MODIFIES SQL DATA** として定義されなければなりません。これが行われない場合、パッケージ内の追加バインドおよび動的ステートメントはランタイム・エラーを生じます (SQLSTATE 42985)。

### FEDERATED

パッケージ内の静的 SQL ステートメントがニックネームまたはフェデレーテッド・ビューを参照するかどうかを指定します。このオプションが指定されず、パッケージ内の静的 SQL ステートメントがニックネームまたはフェデレーテッド・ビューを参照する場合は、警告が返され、パッケージは作成されません。

**NO** ニックネームまたはフェデレーテッド・ビューは、パッケージ内の静的 SQL ステートメントで参照されません。ニックネームまたは

フェデレーテッド・ビューがこのパッケージの準備またはバインド・フェーズ中に静的 SQL ステートメントで見つかった場合、エラーが返され、パッケージは作成されません。

- YES** ニックネームまたはフェデレーテッド・ビューは、パッケージ内の静的 SQL ステートメントで参照が可能です。ニックネームまたはフェデレーテッド・ビューがこのパッケージの準備またはバインド中に静的 SQL ステートメントで見つからなかった場合、エラーまたは警告は返されず、パッケージは作成されます。

#### FEDERATED\_ASYNCRONY

フェデレーテッド・サーバーが組み込み SQL を使用するプログラムのアクセス・プランでサポートする非同期表キュー (ATQ) の最大数を指定します。

- ANY** オプティマイザーはアクセス・プラン用の ATQ の数を判別します。オプティマイザーは ATQ をプラン内のすべての適格な SHIP またはリモート・プッシュダウン演算子に割り当てます。

**DB2\_MAX\_ASYNC\_REQUESTS\_PER\_QUERY** サーバー・オプションに指定される値は、非同期要求の数を制限します。

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

プラン内の ATQ の数。数値を 0 から 32767 の範囲で指定します。

#### FUNCPATH

静的 SQL で、ユーザー定義の特殊タイプおよび機能を解析する際に使用する関数パスを指定します。このオプションを指定しなかった場合、デフォルトの関数パスは "SYSIBM"、"SYSFUN"、または USER になります。ここで USER は USER 特殊レジスターの値です。

*schema-name*

SQL ID (通常または区切り)。これは、アプリケーション・サーバーに存在するスキーマを識別します。スキーマが存在する場合、プリコンパイル時やバインド時に妥当性検査は行われません。同一スキーマは、関数パス内に一度しか存在できません。スキーマ名 **SYSPUBLIC** は、関数パスには指定できません。指定できるスキーマ数は、処理結果の関数パスの長さによって限定され、2048 バイトを超えることはできません。スキーマ **SYSIBM** は、明示的に指定する必要がありません。関数パス内に含まれていなければ、最初のスキーマに暗黙的に想定されます。

#### INSERT

サーバーへプリコンパイルまたはバインドされているプログラムが、パフォーマンス向上のために挿入データをバッファリングすることを要求できるようにします。

- BUF** アプリケーションからの挿入データをバッファリングすることを指定します。

- DEF** アプリケーションからの挿入データをバッファリングしないことを指定します。

**GENERIC "string"**

ターゲット・データベースで定義される新規オプションのバインディングをサポートします。BIND または PRECOMPILE で定義されている BIND オプションを渡すようにするには、このオプションを使用しないでください。構文は次のとおりです。

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

各オプションと値は、1 つ以上の空白・スペースで区切らなければなりません。例えば、以下のようにして、OPT1、OPT2、および OPT3 オプションをそれぞれバインドできます。

```
generic "opt1 value1 opt2 value2 opt3 value3"
```

ストリングの最大長は 4096 バイトです。各オプション名の最大長はストリングで 255 バイトです。

**IMMEDWRITE**

グループ・バッファ・プールに依存するページセットまたはデータベース・パーティションに対する更新について、即時書き込みを行うかどうかを示します。サポートしているのは DB2 for OS/390 だけです。サポートされているオプション値のリストについては、DB2 for OS/390 の資料を参照してください。

**ISOLATION**

このパッケージにバインドされるプログラムを、他の実行プログラムの影響からどの程度分離できるかを指定します。

- CS** カーソル固定を分離レベルとして指定します。
- NC** コミットなし。コミットメント制御が使用されないということ指定します。なお、DB2 はこの分離レベルをサポートしません。
- RR** 反復可能読み取りを分離レベルとして指定します。
- RS** 読み取り固定を分離レベルとして指定します。読み取り固定は、パッケージ内での SQL ステートメントの実行を、他のアプリケーションが読み取りおよび変更を行った行に対する処理から分離させます。
- UR** 非コミット読み取りを分離レベルとして指定します。

**KEEPDYNAMIC**

このパラメーターは、トランザクション間で動的 SQL ステートメントを保持するかどうかを指定します。

DB2 for z/OS でサポートされるオプションの詳細については、DB2 for z/OS インフォメーション・センターの資料を参照してください。

DB2 for Linux, UNIX, and Windows バージョン 9.8 フィックスパック 2 以降、KEEPDYNAMIC YES オプションを指定してバインドされたパッケージ内に準備された動的 SQL ステートメントは、COMMIT または ROLLBACK 操作の後も SQL コンテキスト内に保持されます。これがデフォルトの動作です。

**YES**

準備済みステートメントに関連付けられたステートメント・テキストとセクションが SQL コンテキストで無期限に保持されるように指示します。動的 SQL ステートメントは複数のトランザクション間で保持されます。 **KEEPDYNAMIC YES** を使ってバインド済みのすべてのパッケージは、デフォルトで、既存のパッケージ・キャッシュの動作との互換性を保ちます。

### いいえ

各作業単位の終わりに、準備済みステートメントに関連付けられたステートメント・テキストとセクションが SQL コンテキストで除去されるように指示します。 **KEEPDYNAMIC NO** を使ってバインド済みのパッケージ内に準備された非アクティブの動的 SQL ステートメントは、**COMMIT** または **ROLLBACK** 操作の間に SQL コンテキストから除去されます。ステートメントは、新しいトランザクションの中で再び準備される必要があります。クライアント、ドライバ、またはアプリケーションは、再使用を望むすべての動的 SQL ステートメントを、新しい作業単位で再び準備する必要があります。

以下の環境のもとでは、動的 SQL ステートメントはトランザクションの終わりに達しても、アクティブなままでいることができます。

- **WITH HOLD** オプションを使って宣言されたカーソルが、コミット・ポイントで開いている。
- 動的 SQL ステートメントが **COMMIT** または **ROLLBACK** 操作を実行している。
- **COMMIT** または **ROLLBACK** 操作を実行するストアード・プロシージャまたはユーザー定義関数を、動的 SQL ステートメントが呼び出す。

### LANGLEVEL

アプリケーション内の静的および動的 SQL の構文およびセマンティクスに適用される SQL 規則を指定します。

**MIA** 以下のように ISO/ANS SQL92 規則を選択します。

- エラー **SQLCODE** または **SQLSTATE** の値の検査をサポートするには、アプリケーション・コードの中で **SQLCA** が宣言されていなければなりません。
- **C** の **NULL** 終了ストリングにはブランクが埋め込まれ、切り捨てが実行された場合でも常に **NULL** 終了文字が含まれます。
- **FOR UPDATE** 節は、定位置 **UPDATE** で更新されるすべての列において任意指定です。
- **UPDATE** または **DELETE** ステートメントの対象となる表の列が、検索条件の中や代入節の右辺で参照されているなら、検索 **UPDATE** または **DELETE** に、対象となる表に対する **SELECT** 特権が必要です。
- 索引を使って解決可能な列関数 (例えば **MIN** または **MAX**) は、**NULL** も検査し、**NULL** があれば警告 **SQLSTATE 01003** を戻します。

- CREATE または ALTER TABLE ステートメントの中に重複したユニーク制約が含まれているなら、エラーが戻されます。
- 特権が何も付与されていない場合、付与者にそのオブジェクトに対する権限がないなら、エラーが戻されます (その権限があるなら警告が戻されます)。

**SAA1** 以下のように共通 IBM DB2 規則を選択します。

- エラー SQLCODE または SQLSTATE の値の検査をサポートするには、アプリケーション・コードの中で SQLCA が宣言されていなければなりません。
- C の NULL 終了ストリング、切り捨てが実行された場合には、NULL 文字が最後に付けられません。
- FOR UPDATE 節は、定位置 UPDATE で更新されるすべての列において必須です。
- UPDATE または DELETE ステートメントの対象となる表が、そのステートメントの全選択から参照されるのであれば、検索 UPDATE または DELETE に、対象となる表に対する SELECT 特権は不要です。
- 索引を使って解決可能な列関数 (例えば MIN または MAX) は、NULL を検査せず、警告 SQLSTATE 01003 は戻されません。
- 警告が戻され、ユニーク制約が重複していても無視されます。
- 特権が付与されていないなら、エラーが戻されます。

### SQL92E

以下のように ISO/ANS SQL92 規則を定義します。

- SQLCODE または SQLSTATE の値の検査をサポートするには、その名前の変数をホスト変数の宣言セクションで宣言できます (どちらも宣言されていないなら、プリコンパイル中には SQLCODE が指定されているものとされます)。
- C の NULL 終了ストリングにはブランクが埋め込まれ、切り捨てが実行された場合でも常に NULL 終了文字が含まれます。
- FOR UPDATE 節は、定位置 UPDATE で更新されるすべての列において任意指定です。
- UPDATE または DELETE ステートメントの対象となる表の列が、検索条件の中や代入節の右辺で参照されているなら、検索 UPDATE または DELETE に、対象となる表に対する SELECT 特権が必要です。
- 索引を使って解決可能な列関数 (例えば MIN または MAX) は、NULL も検査し、NULL があれば警告 SQLSTATE 01003 を戻します。
- CREATE または ALTER TABLE ステートメントの中に重複したユニーク制約が含まれているなら、エラーが戻されます。
- 特権が何も付与されていない場合、付与者にそのオブジェクトに対する権限がないなら、エラーが戻されます (その権限があるなら警告が戻されます)。



**LEVEL** *consistency-token*

一貫性トークンを使用するモジュールのレベルを定義します。一貫性トークンとは、8 文字までの長さの任意の英数字値のことです。RDB パッケージの一貫性トークンは、リクエスターのアプリケーションとリレーショナル・データベース・パッケージが同期化されているかどうかを検証します。このオプションは、通常は使用しないでください。

**LONGERROR**

長いホスト変数宣言をエラーとして扱うかどうかを示します。移植性のために、sqlint32 は、プリコンパイル C および C++ コードで INTEGER 列の宣言として使用できます。

**NO** 長いホスト変数宣言の使用に対してエラーを生成しません。これが 32 ビット・システムと 64 ビット Windows システムのデフォルトです。それらのシステムでは、長いホスト変数を INTEGER 列の宣言として使用できます。このオプションを 64 ビット UNIX プラットフォームで使用すると、長いホスト変数を、BIGINT 列の宣言として使用することができます。

**YES** 長いホスト変数宣言の使用に対してエラーを生成します。これが 64 ビット UNIX システムのデフォルトです。

**MESSAGES** *message-file*

警告メッセージ、エラー・メッセージ、および完了状況メッセージの宛先を指定します。メッセージ・ファイルは、バインドが正常であるかどうかによって作成されます。メッセージ・ファイル名を指定しなかった場合、メッセージは標準出力に書き込まれます。ファイルへの完全パスを指定しなかった場合は、現行ディレクトリが使用されます。なお、既存ファイルの名前を指定すると、そのファイルの内容は上書きされます。

**NOLINEMACRO**

出力 .c ファイルへの #line マクロの生成を抑制します。これは、プロファイル、相互参照ユーティリティ、およびデバッガーのようなソース行情報を必要とする開発ツールでファイルを使用する場合に役に立ちます。このプリコンパイル・オプションは、C/C++ プログラム言語でのみ使用します。

**OPTHINT**

照会最適化ヒントを静的 SQL に使用するかどうかを制御します。サポートしているのは DB2 for OS/390 だけです。サポートされているオプション値のリストについては、DB2 for OS/390 の資料を参照してください。

**OPTLEVEL**

SQL ステートメントの中でホスト変数を使用されている場合に、内部 SQLDA の初期化を C/C++ プリコンパイラーで最適化するかどうかを指示します。そのように最適化すれば、密なループ内で単一の SQL ステートメント (FETCH など) を使う場合のパフォーマンスが向上します。

**0** SQLDA の初期化を最適化しないようプリコンパイラーに指示します。

**1** SQLDA の初期化を最適化するようプリコンパイラーに指示します。アプリケーションで以下のものを使っている場合は、この値を指定しないでください。

- 次の例に示すようなポインター・ホスト変数

```

exec sql begin declare section;
char (*name)[20];
short *id;
exec sql end declare section;

```

- 直接 SQL ステートメントに指定される C++ データ・メンバー

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

パッケージ内のすべての静的ステートメントに使用される既存の最適化プロファイルの名前を指定します。このオプションのデフォルト値は、空ストリングです。この値は CURRENT OPTIMIZATION PROFILE 特殊レジスターが NULL である DML ステートメントの動的準備のデフォルトとしても適用されます。指定された名前が修飾されていない場合、それは SQL 識別子であり、QUALIFIER バインド・オプションによって暗黙的に修飾されます。

**BIND** コマンドは最適化ファイル进行处理せず、名前が構造的に有効であることだけを確認します。したがって、最適化プロファイルが存在しないか無効である場合、最適化プロファイルを使用して DML ステートメントが最適化されるまで、理由コード 13 の SQL0437W 警告は発生しません。

#### OUTPUT *filename*

コンパイラーが生成した修正済みソース・ファイルのデフォルト名をオーバーライドします。この名前にはパスを含めることもできます。

#### OS400NAMING

DB2 for System i データにアクセスする際に使用する命名オプションを指定します。サポートしているのは DB2 for System i だけです。サポートされているオプション値のリストについては、DB2 for System i の資料を参照してください。

区切り記号としてスラッシュが使用されているため、DB2 ユーティリティーが **OS400NAMING** SYSTEM オプションを指定してプリコンパイルまたはバインドされていても、このユーティリティーは、System i システムの命名規則を使用する特定の SQL ステートメントに関して実行時に構文エラーを報告することがあります。例えば、**OS400NAMING** SYSTEM オプションを指定してプリコンパイルまたはバインドされているかどうかには関係なく、System i システムの命名規則が使用されている場合、コマンド行プロセッサは SQL CALL ステートメントに関して構文エラーを報告します。

#### OWNER *authorization-id*

パッケージ所有者の 128 バイトの許可 ID を指定します。その所有者には、パッケージに含まれる SQL ステートメントを実行するための特権が必要です。DBADM 権限を持つユーザーのみが、ユーザー ID 以外の許可 ID を指定できます。デフォルト値は、プリコンパイル/バインド処理の 1 次許可 ID です。SYSIBM、SYSCAT、および SYSSTAT はこのオプションには無効な値です。authorization-id はユーザーに限ります (ロールまたはグループにすることはできません)。

#### PACKAGE

パッケージを作成します。PACKAGE、BINDFILE、または SYNTAX のどれも指定されていない場合は、デフォルトで、データベースの中にパッケージが作成されます。

### **USING** *package-name*

プリコンパイラーが生成するパッケージの名前。名前を入力しないなら、アプリケーション・プログラムのソース・ファイルの名前(括弧を取り去って大文字に変換したもの)が使われます。最大長は 128 バイトです。

### **PREPROCESSOR** "*preprocessor-command*"

組み込み SQL ステートメントを処理する前に、プリコンパイラーが実行できるプリプロセッサ・コマンドを指定します。プリプロセッサ・コマンド・ストリング (最大長 1024 バイト) は、二重引用符または単一引用符で囲む必要があります。

このオプションは、宣言セクションでマクロを使用できるようにします。有効なプリプロセッサ・コマンドとは、コマンド行から発行でき、ソース・ファイルを指定しなくてもプリプロセッサを呼び出せるコマンドです。以下に例を示します。

```
xlc -P -DMYMACRO=0
```

### **QUALIFIER** *qualifier-name*

パッケージに含まれる非修飾オブジェクトの 128 バイトの暗黙修飾子を指定します。owner が明示的に指定されているかどうかにかかわらず、その所有者の許可 ID がデフォルト ID になります。

### **QUERYOPT** *optimization-level*

パッケージに含まれるすべての静的 SQL ステートメントに必要な最適化レベルを指示します。デフォルト値は 5 です。SET CURRENT QUERY OPTIMIZATION ステートメントが、使用できる最適化レベルの範囲の詳細を記述します。

### **RELEASE**

リソースを、各 COMMIT ポイントで解放するか、アプリケーションの終了時に解放するかを指示します。このプリコンパイル/バインド・オプションは、DB2 for Linux, UNIX, and Windows 用のサーバーではサポートされません。

### **COMMIT**

各コミット点でリソースを解放します。これは、動的 SQL ステートメントに使用されます。

### **DEALLOCATE**

アプリケーションの終了時にだけリソースを解放します。

### **REOPT**

DB2 がホスト変数、パラメーター・マーカ、グローバル変数、および特殊レジスターの値を使用して実行時にアクセス・パスを最適化するようにするかを指定します。有効な値は以下のとおりです。

**NONE** ホスト変数、パラメーター・マーカ、グローバル変数、または特殊レジスターを含む SQL ステートメントのアクセス・パスは、これらの変数の実際の値によって最適化されません。その代わりに、これらの変数のデフォルトの推定値が使用され、このプランがキャッシュされて使用されます。これがデフォルトの動作です。

**ONCE** 最初に照会が実行されるときに、ホスト変数、パラメーター・マー



カー、グローバル変数、または特殊レジスターの実際の値によって、SQL ステートメントのアクセス・パスが最適化されます。このプランがキャッシュされて使用されます。

#### ALWAYS

毎回の実行時に、ホスト変数、パラメーター・マーカ、グローバル変数、または特殊レジスターの既知の値によって、SQL ステートメントのアクセス・パスが必ずコンパイルおよび再最適化されます。

#### REOPT | NOREOPT VARS

これらのオプションは、**REOPT ALWAYS** および **REOPT NONE** によって置き換えられましたが、以前のリリースとの互換性のために引き続きサポートされています。DB2 がホスト変数、グローバル変数、パラメーター・マーカ、および特殊レジスターの値を使用して実行時にアクセス・パスを判断するようにするかどうかを指定します。サポートしているのは DB2 for OS/390 だけです。サポートされているオプション値のリストについては、DB2 for OS/390 の資料を参照してください。

#### SQLCA

FORTRAN アプリケーションでのみ使用します。その他の言語で使用しても、このオプションは無視されます。

**NONE** 修正されたソース・コードは SAA 定義と整合性がないことを指定します。

**SAA** 修正されたソース・コードは SAA 定義と整合性があることを指定します。

#### SQLERROR

エラーを検出した場合に、パッケージまたはバインド・ファイルを作成するかどうかを指示します。

#### CHECK

ターゲット・システムが、バインドされている SQL ステートメントのすべての構文、およびセマンティックの検査を行うことを指定します。この処理の一部としてパッケージが作成されることはありません。バインド中に、名前とバージョンが同じ既存パッケージを検出した場合、その既存パッケージはドロップも置換 (**ACTION REPLACE** を指定した場合) もされません。

#### CONTINUE

SQL ステートメントのバインド時にエラーが発生しても、パッケージを作成します。許可または存在などの理由でバインドに失敗したこれらのステートメントは、**VALIDATE RUN** も指定されている場合は、実行時に増分でバインドすることができます。実行時にこれらのステートメントを実行しようとする、エラー (SQLCODE -525, SQLSTATE 51015) が生成されます。

#### NOPACKAGE

エラーを検出した場合、パッケージもバインド・ファイルも作成しません。

### SQLFLAG

このオプションに指定された SQL 言語構文との相違点を識別して報告します。

バインド・ファイルまたはパッケージが作成されるのは、**SQLFLAG** オプションに加えて、**BINDFILE** または **PACKAGE** オプションが指定されている場合だけです。

ローカル構文検査が実行されるのは、次に示すオプションのうちの 1 つが指定された場合だけです。

- **BINDFILE**
- **PACKAGE**
- **SQLERROR CHECK**
- **SYNTAX**

**SQLFLAG** が指定されていない場合、**flagger** 関数は呼び出されず、バインド・ファイルやパッケージが影響を受けることはありません。

### SQL92E SYNTAX

SQL ステートメントは、データベース・カタログにアクセスするのに必要な構文規則を除いて、ANSI または ISO SQL92 の項目レベル SQL 言語の書式および構文と比較して調べられます。相違があれば、プリコンパイラ・リストに報告されます。

### MVSDB2V23 SYNTAX

SQL ステートメントは、MVS DB2 バージョン 2.3 SQL 言語構文と比較して調べられます。構文の相違があれば、プリコンパイラ・リストに報告されます。

### MVSDB2V31 SYNTAX

SQL ステートメントは、MVS DB2 バージョン 3.1 SQL 言語構文と比較して調べられます。構文の相違があれば、プリコンパイラ・リストに報告されます。

### MVSDB2V41 SYNTAX

SQL ステートメントは、MVS DB2 バージョン 4.1 SQL 言語構文と比較して調べられます。構文の相違があれば、プリコンパイラ・リストに報告されます。

### SORTSEQ

System i システムで使用するソート・シーケンス表を指定します。サポートしているのは DB2 for System i だけです。サポートされているオプション値のリストについては、DB2 for System i の資料を参照してください。

### SQLRULES

下記のことを指定します。

- タイプ 2 の **CONNECT** を DB2 規則に従って処理するか、それとも ISO/ANS SQL92 の標準 (STD) の規則に従って処理するか。
- 結果セット内の **LOB** 列の形式をアプリケーションが指定する方法。

### DB2

- SQL CONNECT ステートメントで、現在の接続と、確立されている (休止 状態の) 別の接続との間で切り換えることができるようにします。
- このデフォルト設定を使用する場合、アプリケーションは LOB 値と LOB ロケーターのいずれを取り出すかを最初のフェッチ要求時にのみ指定できます。その後のフェッチ要求では、同じ形式を LOB 列に使用する必要があります。

### STD

- SQL CONNECT ステートメントでは、新しい 接続を確立することしかできないようにします。 休止接続へ切り替えるには、SQL SET CONNECTION ステートメントを使います。
- アプリケーションは、それぞれのフェッチ要求で LOB 値を取り出すのか LOB ロケーターを取り出すのかを変更することができます。つまり、BLOCKING バインド・オプションの設定値にかかわらず、1 つ以上の LOB 列を持つカーソルをブロック化することはできません。

### SQLWARN

動的 SQL ステートメントのコンパイル時 (PREPARE または EXECUTE IMMEDIATE を通して)、または記述処理時 (PREPARE...INTO または DESCRIBE を通して) に警告を戻すかどうかを指示します。

**NO** SQL コンパイラーから警告を戻しません。

**YES** SQL コンパイラーから警告を戻します。

SQLCODE +238 は例外です。これは、**SQLWARN** オプションの値が何であろうと戻されます。

### STATICREADONLY

静的カーソルを READ ONLY として扱うかどうかを指定します。

**NO** すべての静的カーソルは、ステートメント・テキストと **LANGLEVEL** プリコンパイル・オプションの設定に基づいて通常生成される場合と同じ属性を取ります。これはデフォルト値です。

**YES** FOR UPDATE または FOR READ ONLY 節を含まない静的カーソルは、READ ONLY と見なされます。

### STRDEL

SQL ステートメントで使用するストリング区切り文字として、アポストロフィ (') または二重引用符 (") のどちらを使用するか指定します。このプリコンパイル/バインド・オプションは、DB2 for Linux, UNIX, and Windows 用のサーバーではサポートされません。このオプションを指定しないと、DRDA サーバーは、組み込みのデフォルト値を使用します。

### APOSTROPHE

ストリング区切り文字として、アポストロフィ (') を使用します。

### QUOTE

ストリング区切り文字として、二重引用符 (") を使用します。

### SYNCPOINT

複数のデータベース接続にまたがってコミットまたはロールバックを調整す

る仕方を指定します。このコマンド・パラメーターは無視されます。ここでは、後方互換性のためにこれを掲載しています。

**NONE** 2 フェーズ・コミットを実行するのにトランザクション管理機能(TM)を使用しないことを指定し、単一更新機構、多重読み取り機構を適用しません。コミットは、関連する各データベースに送られます。コミットが失敗したときのリカバリーは、アプリケーションが行います。

### ONEPHASE

2 フェーズ・コミットを実行するのに TM を使用しないことを指定します。複数のデータベース・トランザクションの各データベースが行う作業をコミットするときは、1 フェーズ・コミットが使用されます。

### TWOPHASE

このプロトコルをサポートする複数のデータベースにまたがって 2 フェーズ・コミットを調整するのに TM が必要であることを指定します。

### SYNTAX

プリコンパイル時に、パッケージまたはバインド・ファイルの作成を抑制します。このオプションを使用すれば、既存のパッケージまたはバインド・ファイルを修正したり変更したりしないで、ソース・ファイルの妥当性を検査できます。SYNTAX は SQLERROR CHECK の同義語です。

SYNTAX を PACKAGE オプションと一緒に使うと、PACKAGE は無視されます。

### TARGET

現行のプラットフォームでサポートされているコンパイラーの 1 つに合わせて調整した修正コードを生成するように、プリコンパイラーに指示します。

### IBMCOB

AIX では、IBM COBOL Set (AIX 版) のコンパイラーのためのコードが生成されます。

### MFCOB

Micro Focus COBOL コンパイラー用のコードが生成されます。Linux、UNIX、および Windows のすべてのオペレーティング・システムでは、TARGET 値が COBOL プリコンパイラーで指定されていない場合、これがデフォルトになります。

### ANSI\_COBOL

ANS X3.23-1985 標準規格と互換性のあるコードが生成されます。

**C** 現行プラットフォーム上の DB2 によりサポートされる C コンパイラーと互換性のあるコードが生成されます。

### CPLUSPLUS

現行プラットフォーム上の DB2 によりサポートされる C++ コンパイラーと互換性のあるコードが生成されます。

### FORTRAN

現行プラットフォーム上の DB2 によりサポートされる FORTRAN コンパイラーと互換性のあるコードが生成されます。

**TEXT** *label*

パッケージの記述。最大長は 255 文字です。また、デフォルト値はブランクです。このプリコンパイル/バインド・オプションは、DB2 for Linux, UNIX, and Windows 用のサーバーではサポートされません。

**TRANSFORM GROUP**

静的 SQL ステートメントが、ユーザー定義の構造化タイプの値をホスト・プログラムと交換するために使用する、変換グループ名を指定します。この変換グループは、動的 SQL ステートメントには使用されません。また、パラメーターの交換や外部関数またはメソッドの結果にも使用されません。

*groupname*

SQL ID。長さは最大で 128 バイトです。グループ名には、修飾子接頭部を含めることはできません。また、接頭部 SYS はデータベースで使用するために予約されているので、その接頭部は使用できません。ホスト変数とやりとりする静的 SQL ステートメントでは、構造化タイプの値の交換に使用する変換グループの名前は以下のようになります。

- **TRANSFORM GROUP BIND** オプション内のグループ名 (もしあれば)
- **TRANSFORM GROUP** 準備オプション内のグループ名。最初のプリコンパイル時に指定したとおりのもの (もしあれば)
- **DB2\_PROGRAM** グループ。グループ名が **DB2\_PROGRAM** の、特定のタイプに対する変換がある場合。
- 上にリストされた条件のいずれもない場合には、変換グループは使用されません。

静的 SQL ステートメントのバインド時には、以下のエラーが発生する可能性があります。

- **SQLCODE** *yyy*, **SQLSTATE** *xxxx*: 変換が必要ですが、静的変換グループが選択されていません。
- **SQLCODE** *yyy*, **SQLSTATE** *xxxx*: 選択された変換グループには、交換するデータ・タイプに必要な変換が含まれていません (入力変数用の **TO SQL**、出力変数用の **FROM SQL**)。
- **SQLCODE** *yyy*, **SQLSTATE** *xxxx*: **FROM SQL** 変換の結果タイプは、出力変数のタイプと互換性がありません。または、**TO SQL** 変換のパラメーター・タイプは、入力変数のタイプと互換性がありません。

これらのエラー・メッセージで、*yyyyy* は SQL エラー・コードによって置き換えられ、*xxxxx* は SQL 状態コードによって置き換えられます。

**UNSAFENULL**

DB2 V10.1 フィックスパック 2 以降、PRECOMPILE オプション

**UNSAFENULL YES** を **COMPATIBILITY\_MODEORA** と組み合わせて指定すると、DB2 アプリケーションへのマイグレーションの間に、未指定の標識変数エラー (アプリケーションでプログラムに NULL 標識を指定していない場合に NULL 値が存在すると生成される) を抑止する互換性が提供されます。

**NO** デフォルトでは、**UNSAFENULL** 機能は **NO** に設定されます。

## PRECOMPILE

データ・タイプが NULL を処理できる場合、アプリケーションは NULL 標識を指定しなければなりません。NULL 標識が使用されず、結果セットのフェッチ中に 1 つ以上の列データが NULL であることが検出されると、未指定の標識変数エラーがアプリケーションに返されます。

**YES** 元は他のデータベース・ベンダーからマイグレーションされたものである DB2 アプリケーションを実行している間に、未指定の標識変数エラー (アプリケーションでプログラムに NULL 標識を指定していない場合に NULL 値が存在すると生成される) を抑止する互換性を提供します。

### VALIDATE

データベース・マネージャーが、許可エラーとエラー未検出のオブジェクトをいつ検査するかを判別します。この妥当性検査には、パッケージ所有者の許可 ID が使用されます。

**BIND** プリコンパイル/バインド時に妥当性検査が実行されます。オブジェクトが 1 つもない場合、または権限がまったく保持されていない場合、エラー・メッセージが作成されます。 **SQLERROR CONTINUE** を指定した場合、エラー・メッセージにかかわらずパッケージ・ファイルやバインド・ファイルは作成されますが、エラーのあるステートメントは実行できません。

**RUN** バインド時に妥当性検査が行われます。すべてのオブジェクトが存在し、全権限が保持されていれば、それ以上実行しても検査は行われません。

プリコンパイル/バインド時にオブジェクトが 1 つもない場合、または権限がまったく保持されていない場合、**SQLERROR CONTINUE** オプションの設定とは無関係に警告メッセージが作成されて、パッケージは正常にバインドされます。ただし、プリコンパイル/バインド処理時に SQL ステートメントの権限検査と存在検査に障害が生じた場合、実行時に再実行される可能性があります。

### VERSION

パッケージのバージョン ID を定義します。このオプションが指定されていない場合、パッケージ・バージョンは "" (空ストリング) です。

*version-id*

任意の英数字値、\$、#、@、\_、-、または . で、長さは 64 文字以下のバージョン ID を指定します。

**AUTO** バージョン ID は、整合性トークンから生成されます。整合性トークンがタイム・スタンプの場合 (**LEVEL** オプションが指定されていなければそうなります)、そのタイム・スタンプは ISO 文字フォーマットに変換されて、バージョン ID として使用されます。

### WCHARTYPE

**GRAPHIC** データのフォーマットを指定します。

### CONVERT

`wchar_t` 基本タイプを使って宣言されたホスト変数には、`wchar_t` 形式のデータが入れられるものとして扱われます。この形式は、デ



データベースに格納される GRAPHIC データの形式 (DBCS 形式) と直接に互換性はないので、 `wchar_t` ホスト変数の入力データは、ANSI C 関数 `wcstombs()` を使って DBCS 形式に暗黙のうちに変換されます。同じように出力 DBCS データは、ホスト変数を保管する前に、`mbstowcs()` を使用して `wchar_t` 形式に暗黙のうちに変換されます。

### NOCONVERT

`wchar_t` 基本タイプを使って宣言されたホスト変数には、 DBCS 形式のデータが入られるものとして扱われます。これは、データベースの中で GRAPHIC データ用に使われる形式ですが、 C 言語で採用されている固有の `wchar_t` 形式とは違うものです。 **NOCONVERT** を使用する場合、 GRAPHIC データはアプリケーションとデータベースの間で変換されないことになり、それによって効率が改善されます。しかし、アプリケーション側では、データベース・マネージャーに `wchar_t` 形式のデータが渡されることがないようにしなければなりません。このオプションを使用する場合は、 `wchar_t` ホスト変数を C ワイド文字列関数で処理しないようにし、ワイド文字リテラル (*L-literals*) で初期化しないようにしなければなりません。

### WITH / WITHOUT TEMPORAL HISTORY

静的 SQL ステートメントまたは動的 SQL ステートメントでシステム期間テンポラル表のデータに対する変更を行ったときに、対応する履歴表を変更するかどうかを指定します。

**WITH** システム期間テンポラル表のデータを変更したときに、対応する履歴表を変更するよう指定します。

これはデフォルト・オプションです。

### WITHOUT

システム期間テンポラル表のデータを変更したときに、対応する履歴表を変更しないよう指定します。データベース・マネージャーは、行開始列、行終了列、およびトランザクション開始 ID 列が **GENERATED ALWAYS** として定義されている場合であっても、これらの列をオーバーライドする値を提供できます。

このオプションには、DBADM 権限が必要です。

## 使用上の注意

修正されたソース・ファイルが作成されますが、これには SQL ステートメントと同じホスト言語ステートメントが入っています。デフォルトでは、接続が既に確立されているデータベース内にパッケージが作成されます。パッケージの名前は、ファイル名と同じ (拡張子を取り去って大文字に変換したもの) であり、最大 8 文字までです。パッケージ名の最大長は 128 バイトですが、**PACKAGE USING** オプションが指定されていなければ、ファイル名の最初の 8 文字だけを使用して DB2 の前のバージョンとの互換性が保守されます。

データベースへの接続が終わると、開始されているトランザクションの下で **PREP** が実行されます。次に **PREP** は **COMMIT** または **ROLLBACK** を発行し、現行トランザクションを終了して別のトランザクションを開始します。

## PRECOMPILE

既に存在していないスキーマ名を指定してパッケージを作成すると、そのスキーマが暗黙のうちに作成されます。スキーマの所有者は **SYSIBM** になります。スキーマに対する **CREATEIN** 特権が **PUBLIC** に付与されます。

プリコンパイル時には、パッケージが作成されて **EXPLSNAP** が指定されているのでない限り、**Explain** スナップショットは取られません。スナップショットは、パッケージを作成するユーザーの **Explain** 表に入れられます。同じように、**Explain** 表情報が取得されるのは、**EXPLAIN** が指定されていて、パッケージが作成される場合だけです。

1 つでも致命的エラーが発生するか、または 100 を超えるエラーが発生すると、プリコンパイルは停止してしまいます。致命的エラーが発生すると、ユーティリティーはプリコンパイルを停止し、すべてのファイルをクローズしてからパッケージを廃棄します。

パッケージがバインド動作を公開するとき、以下のとおりとなります。

1. **BIND** オプション **OWNER** の暗黙的または明示的な値は、動的 **SQL** ステートメントの許可検査に使用されます。
2. **BIND** オプション **QUALIFIER** の暗黙的または明示的な値は、動的 **SQL** ステートメント内の非修飾オブジェクトを修飾するための暗黙的修飾子として使用されません。
3. 特殊レジスター **CURRENT SCHEMA** の値は、修飾には影響しません。

単一の接続で複数のパッケージが参照される場合、これらのパッケージによって準備されたすべての動的 **SQL** ステートメントはその特定のパッケージおよびそれらが使用される環境について **DYNAMICRULES** オプションで指定された動作をします。

**SQL** ステートメントがエラーであることが検出され、**PRECOMPILE** オプション **SQLERROR CONTINUE** が指定されている場合、このステートメントは無効とマークされます。この **SQL** ステートメントの状態を変えるには、さらに別の **PRECOMPILE** を発行する必要があります。暗黙的および明示的な再バインドでは、**VALIDATE RUN** でバインドされたパッケージ内の無効なステートメントの状態は変わりません。ステートメントは、再バインド時にオブジェクトが存在するかまたは権限の問題があるかどうかに応じて、暗黙的と明示的な再バインドとの両方で、静的バインドから追加バインドに変更したり、追加バインドを静的バインドに変更することができます。

**REOPT ONCE** または **REOPT ALWAYS** を指定してパッケージをバインドすると、静的ステートメントおよび動的ステートメントのコンパイル数とパフォーマンスが変化することがあります。

組み込み **SQL** プログラムでは、**FEDERATED\_ASYNC** プリコンパイル・オプションが明示的に指定されていない場合、パッケージ内の静的ステートメントは **federated\_async** 構成パラメーターを使用してバインドされます。

**FEDERATED\_ASYNC** オプションが明示的に指定されている場合、値はパッケージのバインドに使用され、特殊レジスターの初期値ともなります。指定されていない場合、データベース・マネージャー構成パラメーターの値が特殊レジスターの初期値として使用されます。**FEDERATED\_ASYNC** プリコンパイル・オプションは、明示的に設定されている場合に限り、動的 **SQL** に影響を与えます。



## PRUNE HISTORY/LOGFILE

リカバリー履歴ファイルから項目を削除したり、アクティブ・ログ・ファイル・パスからログ・ファイルを削除したりするのに使用します。リカバリー履歴ファイルからの項目の削除は、ファイルが非常に大きくなったり保持期間が長くなっている場合に必要になることがあります。

区画に分割された環境では、**PRUNE HISTORY** コマンドは、発行されたデータベース・パーティションに関してのみ実行されます。複数のパーティションに関して履歴を整理するには、**PRUNE HISTORY** コマンドをそれぞれ個々のデータベース・パーティションから発行するか、または `db2_all` 接頭部を使用して **PRUNE HISTORY** コマンドをすべてのデータベース・パーティションに対して実行します。

**重要:** **PRUNE LOGFILE** コマンドは非推奨になっており、将来のリリースで除去される可能性があります。代わりに **PRUNE HISTORY** コマンドを使用してください。

### 許可

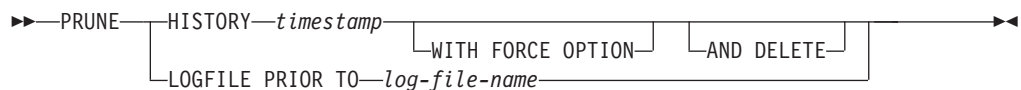
以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMANT
- DBADM

### 必要な接続

データベース

### コマンド構文



### コマンド・パラメーター

#### **HISTORY** *timestamp*

削除される、リカバリー履歴ファイルにある項目範囲を識別します。完全なタイム・スタンプ (書式 `yyyymmddhhmmss`)、または最初の接頭部 (最小値 `yyyy`) を指定できます。提供されているそのタイム・スタンプ以下のタイム・スタンプ付きのすべての項目は、リカバリー履歴ファイルから削除されます。最初の接頭部を指定した場合、指定されていないタイム・スタンプの構成要素は `yyyy0101000000` と解釈されます。

#### **WITH FORCE OPTION**

最新のリストア・セットのいくつかの項目がファイルから削除されるとしても、指定したタイム・スタンプに従って項目を整理することを指定します。リストア・セットは、バックアップ・イメージのすべてのリストアを含む、最新の全データベース・バックアップです。このパラメーターを指定しない場合、バックアップ・イメージ転送からのすべての項目は履歴の中で保守されます。

### AND DELETE

履歴ファイルの項目を削除する際に、関連するログ・アーカイブを (ロケーション情報に基づいて) 物理的に削除することを指定します。このオプションは、ログ・アーカイブが不要になった場合に、アーカイブ・ストレージ・スペースがリカバリーされるようにする上で、特に有用です。ユーザー出口プログラムによりログをアーカイブしている場合は、このオプションを使用してそれらのログを削除することはできません。

**auto\_del\_rec\_obj** データベース構成パラメーターを ON に設定している場合、**AND DELETE** パラメーターを指定して **PRUNE HISTORY** を呼び出すと、バックアップ・イメージも物理的に削除され、その履歴ファイル項目が整理される場合にはコピー・イメージがロードされます。

### LOGFILE PRIOR TO *log-file-name*

ログ・ファイル名を表すストリング (例: S0000100.LOG) を指定します。指定したログ・ファイルより前のすべてのログ・ファイルは削除されます。指定したログ・ファイルそのものは削除されません。 **logarchmeth1** データベース構成パラメーターは、**OFF** 以外の値に設定する必要があります。

注: この値は、DB2 pureScale 環境ではサポートされていません。

### 例

前に行われた、すべてのリストア、ロード、表スペース、バックアップ、および全部のデータベース・バックアップのための項目を除去するには、リカバリー履歴ファイルから 1994 12.1 を含んで、次のように入力してください。

```
db2 prune history 199412
```

199412 は 19941201000000 と解釈されます。

### 使用上の注意

**WITH FORCE OPTION** が使用されている場合、データベースの自動リストアに必要な項目を削除してしまう可能性があります。その場合でも手動リストアは正常に動作します。また、このコマンドを使用すると、**db2ckrst** ユーティリティーが、必要なバックアップ・イメージの完全なチェーンを正しく分析できなくなる可能性もあります。 **PRUNE HISTORY** コマンドを **WITH FORCE OPTION** なしで使用した場合、必要な項目が削除されることはありません。

状況が **DB2HISTORY\_STATUS\_DO\_NOT\_DELETE** の項目は整理されません。 **WITH FORCE OPTION** が使用されている場合、**DB2HISTORY\_STATUS\_DO\_NOT\_DELETE** というマークが付いたオブジェクトは、やはり整理または削除されます。 **UPDATE HISTORY** コマンド、**UPDATE\_HISTORY** を指定した **ADMIN\_CMD**、または **db2HistoryUpdate** API を使用して、リカバリー履歴ファイルの項目の状況を **DB2HISTORY\_STATUS\_DO\_NOT\_DELETE** に設定できます。 **DB2HISTORY\_STATUS\_DO\_NOT\_DELETE** 状況を使用して、キー・リカバリー履歴ファイルの項目が整理されないようにしたり、関連するリカバリー・オブジェクトが削除されないようにすることができます。

**PRUNE HISTORY** コマンドを使用してスナップショット・データベース・バックアップ履歴ファイルの項目を整理できますが、**AND DELETE** パラメーターを使用して関連

する物理リカバリー・オブジェクトを削除することはできません。スナップショット・バックアップ・オブジェクトを削除する唯一の方法は、**db2acsutil** コマンドを使用することです。

## PUT ROUTINE

指定されたルーチン SQL アーカイブ (SAR) ファイルを使用して、データベースにルーチンを定義します。

### 許可

DBADM。この権限は、直接ユーザーに付与しなければならず、ロールを介して継承することはできません。

### 必要な接続

データベース。暗黙接続が可能な場合には、デフォルト・データベースへの接続が確立されます。

### コマンド構文

```

▶▶ PUT ROUTINE FROM file-name
└──┬── OWNER new-owner
    └── USE REGISTERS
  
```

### コマンド・パラメーター

#### FROM *file-name*

ルーチン SQL アーカイブ (SAR) が保管されているファイルの名前。

#### OWNER *new-owner*

ルーチンの許可検査に使用する新しい許可名を指定します。新規所有者は、定義されるルーチンに必要な権限を持っていないければなりません。OWNER 節が指定されない場合は、元々ルーチンに定義されていた許可名が使用されます。

#### USE REGISTERS

CURRENT SCHEMA および CURRENT PATH 特殊レジスターをルーチンの定義に使用することを指示します。この節が指定されない場合、ルーチンが定義されるときに、デフォルト・スキーマと SQL パスの設定が使用されます。ルーチン定義 (ルーチンの名前を含む) の非修飾オブジェクト名のスキーマ名として CURRENT SCHEMA が使用され、ルーチン定義の非修飾ルーチンとデータ・タイプを解決するために CURRENT PATH が使用されます。

### 例

```
PUT ROUTINE FROM procs/procl.sar;
```

### 使用上の注意

指定のスキーマの下で、複数のプロシージャが並行してインストールされることはありません。

GET ROUTINE または PUT ROUTINE 操作 (またはそれに対応するプロシージャ) が正常に実行できない場合、エラー (SQLSTATE 38000)、および失敗の原因に関する情報を示す診断テキストを毎回戻します。例えば、GET ROUTINE に指定されたプロシージャ名が SQL プロシージャを識別しない場合、"-204, 42704" という診断テキストが戻ります。"-204" は SQLCODE、"42704" は SQLSTATE で、それぞれ問題の原因を示します。この例の SQLCODE および SQLSTATE は、GET ROUTINE

コマンドに指定されたプロシージャ名が未定義であることを示しています。

---

### QUERY CLIENT

アプリケーション・プロセスの現行の接続設定を戻します。

#### 許可

なし

#### 必要な接続

なし

#### コマンド構文

▶—QUERY CLIENT—▶

#### コマンド・パラメーター

なし

#### 例

以下に示すのは、**QUERY CLIENT** の出力例です。

The current connection settings of the application process are:

```
CONNECT      = 1
DISCONNECT   = EXPLICIT
MAX_NETBIOS_CONNECTIONS = 1
SQLRULES     = DB2
SYNCPOINT    = ONEPHASE
CONNECT_MEMBER = 0
ATTACH_MEMBER = -1
```

**CONNECT\_MEMBER** および **ATTACH\_MEMBER** が **SET CLIENT** コマンドを使用して設定されていない場合、これらのパラメーターの値は環境変数 **DB2NODE** のものと同じになります。 **CONNECT\_MEMBER** または **ATTACH\_MEMBER** パラメーターの表示値が **-1** である場合、パラメーターは設定されていません。つまり、環境変数 **DB2NODE** が設定されなかったか、あるいは以前に発行された **SET CLIENT** コマンドでパラメーターが指定されませんでした。

#### 使用上の注意

アプリケーション・プロセスの接続設定は、実行中にいつでも照会できます。

## QUIESCE

指定したインスタンスまたはすべてのメンバーにわたるデータベースのどちらかに関して、すべてのユーザーを強制的に切断して静止モードにします。

インスタンスまたはデータベースが静止モードにある間、それに対して管理タスクを実行できます。管理タスクの完了後、**UNQUIESCE** コマンドを使用してインスタンスまたはデータベースを活動化して、他のユーザーがデータベースに接続できるようにします。

このモードでは、この制限モード中に権限を持つユーザーだけがインスタンスまたはデータベースにアタッチまたは接続することができます。

**SYSADM**、**SYSMAINT**、および **SYSCTRL** 権限を持つユーザーは、インスタンスの静止中に常にそのインスタンスにアクセスでき、**SYSADM** および **DBADM** 権限を持つユーザーは、データベースの静止中に常にそのデータベースにアクセスできます。

### 有効範囲

**QUIESCE DATABASE** を使用すると、データベース内のすべてのオブジェクトは静止モードに入ります。許可が与えられたユーザー/グループ、および **SYSADM**、**SYSMAINT**、**DBADM**、または **SYSCTRL** だけが、データベースまたはそのオブジェクトにアクセスできます。

**QUIESCE INSTANCE** *instance-name* は、インスタンス *instance-name* 内のインスタンスおよびデータベースが静止モードに入ることを意味します。インスタンスにアクセスできるのは、**SYSADM**、**SYSMAINT**、および **SYSCTRL** と、許可が与えられているグループだけです。

インスタンスが静止モードにある場合、インスタンス内のデータベースを静止モードにすることはできません。

データベースが **SUSPEND\_WRITE** 状態の場合、そのデータベースを静止モードにはできません。

### 許可

以下の権限のいずれか。

データベース・レベルの静止の場合:

- **SYSADM**
- **DBADM**

インスタンス・レベルの静止の場合:

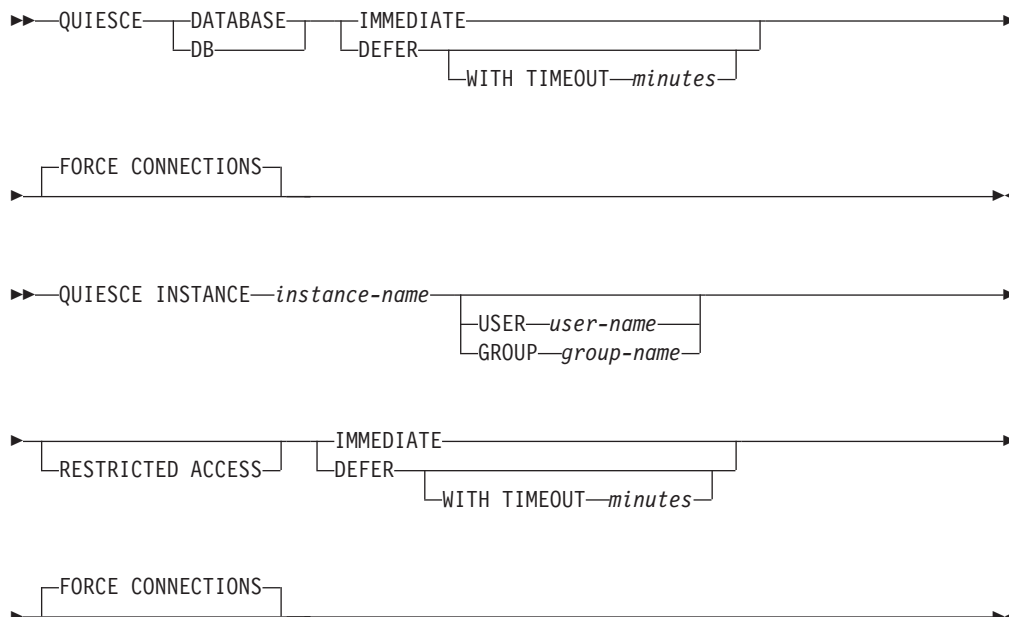
- **SYSADM**
- **SYSCTRL**

## 必要な接続

データベース

(インスタンスの静止ではデータベース接続は必須ではありません。)

## コマンド構文



## コマンド・パラメーター

**DEFER** QUIESCE の実行をアプリケーションが現行の作業単位をコミットするまで待ちます。

### WITH TIMEOUT *minutes*

アプリケーションが現在の作業単位をコミットするのを待機する時間を分単位で指定します。値を指定しない場合、単一パーティション・データベース環境では、デフォルト値が 10 分になります。パーティション・データベース環境では、データベース・マネージャ構成パラメーター **start\_stop\_time** によって指定された値が使用されます。

### IMMEDIATE

トランザクションがコミットされるのを待たず、即時にトランザクションをロールバックします。

### FORCE CONNECTIONS

接続を強制的にオフにします。

### DATABASE

データベースを静止します。データベース内のすべてのオブジェクトを静止モードにします。指定したグループのユーザーと、SYSADM、SYSMAINT、および SYSCTRL 権限を持つユーザーだけが、データベースまたはそのオブジェクトにアクセスすることができます。



**INSTANCE** *instance-name*

インスタンス *instance-name*、およびインスタンス内のデータベースを静止モードにします。インスタンスにアクセスできるのは、SYSADM、SYSMAINT、および SYSCTRL 権限を持つユーザーと、指定したグループのユーザーだけです。

静止インスタンス内のデータベースは、データベースへのすべての接続試行に対する許可検査を行うためにアクティブ化されます。こうする必要があるのは、接続しようとするユーザー ID が DBADM 権限を持っているかどうかを判別するためです。この権限はデータベース・カタログに保管されているので、ユーザー ID がこの権限を持っているかどうかを判別するためには、そのデータベースをアクティブ化することが必要になります。この許可検査が行われないようにするには、**RESTRICTED ACCESS** オプションを指定します。

**USER** *user-name*

インスタンスの静止中にそのインスタンスへのアクセスが許可されているユーザーの名前を指定します。

**GROUP** *group-name*

インスタンスの静止中にそのインスタンスへのアクセスが許可されているグループの名前を指定します。

**RESTRICTED ACCESS**

このオプションは、静止インスタンスのデータベースへのすべての接続試行に対する許可検査 (ユーザー ID が DBADM 権限を持っているかどうかを判別するための許可検査) が行われないようにする場合に指定します。この場合でも、インスタンス・レベルの許可検査は行われます。ユーザー ID の SYSADM、SYSCTRL、または SYSMAINT 権限の検査は、データベースのアクティブ化を必要としないからです。

**注:** このコマンド・オプションは、DB2 バージョン 9.7 フィックスパック 2 以降で使用できます。

**例**

以下に示すのは、接続の強制がデフォルトの動作であるため、明示的な説明は必要なく、この例から省くことができる例です。

```
db2 quiesce instance crankarm user frank immediate force connections
```

以下の例は、データベースに接続しているすべてのユーザーを、強制的に切断にします。

```
db2 quiesce db immediate
```

- 最初の例は、インスタンス crankarm を静止し、ユーザー frank は引き続きデータベースを使用できるようにします。

2 番目の例は、アタッチしているデータベースを静止し、以下のいずれかの権限をもつユーザー以外のユーザーは、すべてアクセスできないようにします。

SYSADM、SYSMAINT、SYSCTRL、または DBADM

- このコマンドとともに **FORCE CONNECTIONS** オプションを指定すると、データベースまたはインスタンスからすべてのユーザーを強制的に切断します。 **FORCE CONNECTIONS** はデフォルトの動作です。コマンドのパラメーターは、互換性の理由により許容されています。
- コマンドは **FORCE CONNECTIONS** と同期され、**FORCE CONNECTIONS** が完了しないと完了しません。

### 使用上の注意

- **QUIESCE INSTANCE** の後、SYSADM、SYSMAINT、または SYSCTRL 権限を持つユーザー、またはコマンドにパラメーターとして指定するユーザー名およびグループ名だけが、インスタンスに接続できます。
- **RESTRICTED ACCESS** オプションを指定すると、許可検査を行うために静止インスタンス内のデータベースがアクティブ化されることがなくなります。データベースに接続しようとしているユーザー ID が、そのデータベースに対する DBADM 権限または QUIESCE\_CONNECT 特権を持っていても、接続は許可されません。SYSADM、SYSCTRL、または SYSMAINT 権限を持つユーザー ID、およびこのコマンドで指定されたユーザーまたはグループのみが、データベースへの接続を許可されます。
- **RESTRICTED ACCESS** オプションは、静止インスタンス内のデータベースへの排他的接続が必要な場合に使用するようになっています。このようなケースとしては、オフライン・バックアップを行う場合や、他の保守アクティビティを実行する場合があります。
- **QUIESCE DATABASE** の後、SYSADM、SYSMAINT、SYSCTRL、または DBADM 権限、および GRANT または REVOKE 特権を持つユーザーは、接続可能なユーザーを指定できます。この情報は永続的にデータベース・カタログ表に保管されます。

以下に例を示します。

```
grant quiesce_connect on database to username/groupname
revoke quiesce_connect on database from username/groupname
```

- DB2 pureScale 環境では、データベースの静止およびインスタンスの再始動の後、データベースはすべてのメンバーにわたり静止状態のままになります。静止状態を解除するには、明示的な **UNQUIESCE DATABASE** コマンド呼び出しが必要です。

## QUIESCE TABLESPACES FOR TABLE

特定の表の表スペースを静止させます。有効な静止モードは、共有、更新意図、および排他の 3 つです。

静止機能の結果として生じる状態には、次の 3 つの状態があります。

- 静止: SHARE
- 静止: UPDATE
- 静止: EXCLUSIVE

### 有効範囲

単一パーティション環境では、ロード操作中に排他モードのロード操作を起動すると、このコマンドは表スペースをすべて静止します。パーティション・データベース環境では、このコマンドはデータベース・パーティションでローカルに活動します。このコマンドは、ロード操作を実行しているデータベース・パーティションに属する表スペースの部分のみを静止します。パーティション表の場合、表に関連付けられている SYSDATAPARTITIONS.TBSPACEID および SYSDATAPARTITIONS.LONG\_TBSPACEID の中のリストに含まれているすべての表スペースのうち、状況が「正常」、「アタッチ」、または「デタッチ」であるもの (例えば SYSDATAPARTITIONS.STATUS が '', 'A', または 'D') が静止されません。

### 許可

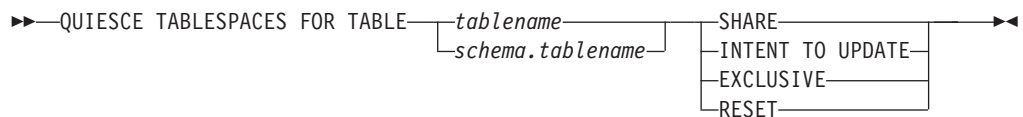
以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMANT
- DBADM
- LOAD

### 必要な接続

データベース

### コマンド構文



### コマンド・パラメーター

#### TABLE

*tablename*

非修飾表名を指定します。システム・カタログ表を指定することはできません。

## QUIESCE TABLESPACES FOR TABLE

*schema.tablename*

修飾表名を指定します。 *schema* が指定されない場合には、CURRENT SCHEMA が使用されます。システム・カタログ表を指定することはできません。

**SHARE** 静止が共有モードであることを指定します。

「静止モードでの共有」要求を行うと、トランザクションは、表スペースに対して意図的共有ロックを、および表に対して共有ロックを要求します。トランザクションがロックを獲得すると、表スペースの状態が QUIESCED SHARE に変更されます。この状態は、他のユーザーがこれと矛盾する状態を保持していない場合に限り、静止者に付与されます。表スペースの状態は、その状態が持続されるように、許可 ID およびそのユーザーのデータベース・エージェント ID とともに、表スペースにある表に記録されます。ある表の表スペースが QUIESCED SHARE 状態である間は、その表を変更できません。表および表スペースに要求するその他の共有モードは、認められません。トランザクションがコミットまたはロールバックされる際、ロックは解除されますが、その表の表スペースはその状態が明示的にリセットされるまで、QUIESCED SHARE 状態のまま残ります。

### INTENT TO UPDATE

静止モードが更新意図モードであることを指定します。

「静止モードでの更新意図」要求を行うと、表スペースは意図的排他 (IX) モードでロックされ、表は更新 (U) モードでロックされます。表スペースの状態は、表スペースの表に記録されます。

### EXCLUSIVE

静止が排他モードであることを指定します。

「静止モードでの排他」要求を行うと、トランザクションは、表スペースに対する特別な排他ロックと、表に対する特別な排他ロックを要求します。トランザクションがロックを獲得すると、表スペースの状態が QUIESCED EXCLUSIVE に変更されます。表スペースの状態は、静止者の許可 ID およびデータベース・エージェント ID とともに、表スペース表に記録されます。表スペースは、スーパー排他モードで保留されているため、表スペースへの他のアクセスは認められません。静止プログラム機能呼び出すユーザー (静止プログラム) は、その表と表スペースへの排他的アクセスを行うことができます。

**RESET** 表スペースの状態が、「正常」にリセットされることを指定します。静止要求を発行した接続がまだアクティブである場合、静止状態をリセットすることはできません。

静止者がリセットを発行すると、その静止者に関する静止モードのみリセットされます。静止者が複数存在する場合は、表スペースの状態が変化していないように見えます。

システム期間テンポラル表および関連した履歴表を操作している場合、リセット操作は、最初に静止モードを設定したときに使用したのと同じ表に対して実行する必要があります。

## 例

```
db2 quiesce tablespaces for table staff share
db2 quiesce tablespaces for table boss.org intent to update
```

## 使用上の注意

このコマンドは、DB2 pureScale 環境ではサポートされていません。

静止は持続ロックです。その利点は、それがトランザクション障害、接続障害、およびシステム障害 (電源障害や、リブートなど) が生じても持続することです。

静止は接続によって所有されます。接続が失われた場合、静止は残りますが、非所有の状態に移り、ファントム静止 と呼ばれます。例えば、ロード操作の削除フェーズ中に停電が発生して中断すると、ロード中の表の表スペースは静止排他状態のままになります。データベースの再始動時に、この静止は非所有 (ファントム) の状態になります。ファントム静止を取り除くには、静止モードが設定されたときに使用されたのと同じユーザー ID による接続が必要です。

ファントム静止を取り除くには、次のようにします。

1. 静止モードが設定されたときに使用されたのと同じユーザー ID でデータベースと接続する。
2. **LIST TABLESPACES** コマンドを使用して、静止させる表スペースを決定する。
3. 現行の静止状態を使用して、表スペースを再静止させる。以下に例を示します。

```
db2 quiesce tablespaces for table mytable exclusive
```

完了すると、新しい接続が静止を所有するようになり、ロード操作を再開できるようになります。

1 つの表スペースに対する静止者の限度は、常に 5 人です。

静止プログラムは表スペースの状態を、あまり制限的でない状態から、より制限的な状態へ (例えば、S から U へ、または U から X へ) 変更できます。しかし、ユーザーが既に保持している状態より低い状態を要求しても、元の状態が戻されません。つまり、状態がダウングレードされることはありません。

システム期間テンポラル表に対する静止操作を実行すると、そのシステム期間テンポラル表および履歴表に関連付けられたすべての表スペースが静止されます。履歴表に対する静止操作を実行すると、その履歴表に関連付けられたすべての表スペースと、関連付けられたシステム期間テンポラル表が静止されます。

---

## QUIT

コマンド行プロセッサの対話式入力モードを終了し、オペレーティング・システムのコマンド・プロンプトに戻ります。

バッチ・ファイルが、コマンド行プロセッサにコマンドを入力するのに使用されている場合には、**QUIT**、**TERMINATE**、またはファイルの終わりが検出されるまで、コマンドは処理されます。

### 許可

なし

### 必要な接続

なし

### コマンド構文

▶▶—QUIT—◀◀

### コマンド・パラメーター

なし

### 使用上の注意

**QUIT** はコマンド行プロセッサ・バックエンド・プロセスを終了することも、データベース接続を中断することもしません。**CONNECT RESET** は、接続を中断しますが、バックエンド・プロセスを終了しません。**TERMINATE** コマンドは両方を行います。

## REBIND

バインド・ファイルを用いずに、データベースに保管されているパッケージを再作成できるようにします。

### 許可

以下の権限のいずれか。

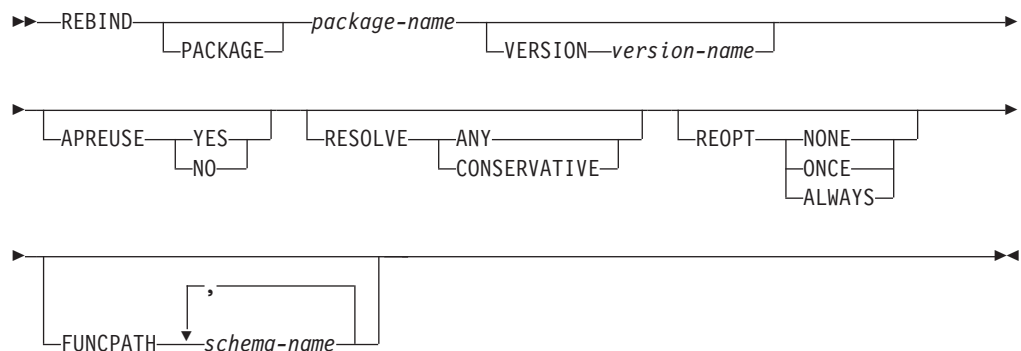
- DBADM 権限
- スキーマに対する ALTERIN 特権
- パッケージに対する BIND 特権

SYSCAT.PACKAGES システム・カタログ表の BOUNDBY 列に記録した許可 ID は、パッケージの最新のバインド・プログラムの ID であり、再バインド用のバインド・プログラム許可 ID として使用されます。また、パッケージの表参照のためのデフォルト・スキーマとしても使用されます。このデフォルト修飾子は、ユーザーが実行する再バインド要求の許可 ID と異なっていても問題ありません。REBIND は、パッケージの作成時に指定されたのと同じ BIND オプションを使用します。

### 必要な接続

データベース。データベース接続が存在しない場合で、暗黙の接続が有効な場合には、デフォルト・データベースへの接続が行われます。

### コマンド構文



### コマンド・パラメーター

#### PACKAGE *package-name*

再バインドされるパッケージを指定する修飾されている、または修飾されていない名前。

#### VERSION *version-name*

再バインドするパッケージのバージョン。バージョンが指定されない場合は、"" (空ストリング) と見なされます。

#### APREUSE

静的 SQL アクセス・プランが再使用されるかどうかを指定します。このオプションが有効である場合、照会コンパイラーは、再バインド時および将来の暗黙および明示の再バインド時に、既存のパッケージ内の静的 SQL ステ



## REBIND

ートメントにアクセス・プランを再使用しようとしています。デフォルトは、**BIND** コマンドまたは **REBIND** コマンド、あるいは **ALTER PACKAGE** ステートメントの以前の呼び出し時に使用された値です。この値を判別するには、**SYSCAT.PACKAGES** 内のパッケージの **APREUSE** 列を照会します。

**YES** 照会コンパイラーは、パッケージ内のステートメントにアクセス・プランを再使用しようとしています。

**NO** 照会コンパイラーは、パッケージ内のステートメントにアクセス・プランを再使用しようとしません。

## RESOLVE

パッケージの再バインドの実行に、従来のバインド・セマンティクスを使用するかどうかを指定します。これは、解決に **SQL** パスを使用する新しいオブジェクトが、パッケージ内の静的 **DML** ステートメントの解決時に考慮されるかどうかに影響します。このオプションは **DRDA** ではサポートされていません。有効な値は以下のとおりです。

**ANY** オブジェクト解決にその **SQL** パスを使用するオブジェクトに対する参照を解決するのに、**SQL** パスにあるすべての可能な組み合わせが考慮されます。従来のバインド・セマンティクスは使用されません。これはデフォルトです。

## CONSERVATIVE

オブジェクト解決に **SQL** パスを使用するオブジェクトに対する参照を解決するのに、最後の明示的バインドのタイム・スタンプより前に定義された **SQL** パスのオブジェクトのみが考慮されます。従来のバインド・セマンティクスを使用します。このオプションは、作動不能パッケージではサポートされていません。

## REOPT

**DB2** がホスト変数、パラメーター・マーカ、グローバル変数、および特殊レジスターの値を使用して実行時にアクセス・パスを最適化するようにするかどうかを指定します。

**NONE** ホスト変数、パラメーター・マーカ、グローバル変数、または特殊レジスターを含む **SQL** ステートメントのアクセス・パスは、これらの変数の実際の値によって最適化されません。その代わりに、これらの変数のデフォルトの推定値が使用され、このプランがキャッシュされて使用されます。これがデフォルトの動作です。

**ONCE** 最初に照会が実行されるときに、ホスト変数、パラメーター・マーカ、グローバル変数、または特殊レジスターの実際の値によって、**SQL** ステートメントのアクセス・パスが最適化されます。このプランがキャッシュされて使用されます。

## ALWAYS

毎回の実行時に、ホスト変数、パラメーター・マーカ、グローバル変数、または特殊レジスターの既知の値によって、**SQL** ステートメントのアクセス・パスが必ずコンパイルおよび再最適化されます。

## FUNCPATH

静的 **SQL** で、ユーザー定義の特殊タイプおよび機能を解析する際に使用する



る関数パスを指定します。デフォルトは、そのパッケージに対して前回 **BIND** コマンドまたは **REBIND** コマンドが呼び出されたときに使用された値です。この値を調べるには、**SYSCAT.PACKAGES** ビューで該当パッケージの **FUNC\_PATH** 列を照会してください。

*schema-name*

SQL ID (通常または区切り)。これは、アプリケーション・サーバーに存在するスキーマを識別します。スキーマが存在する場合、プリコンパイル時やバインド時に妥当性検査は行われません。同じスキーマを関数パス内で 2 回以上使用することはできません。

**SYSPUBLIC** スキーマ名は関数パスには指定できません。指定できるスキーマの数は、最終的な関数パスの長さ (2048 バイト以下) により制限されます。**SYSIBM** スキーマを指定する必要はありません。これを関数パスに含めない場合、**SYSIBM** スキーマが最初のスキーマであると仮定されます。

## 使用上の注意

再バインドが正常に行われても、**REBIND** がトランザクションを自動的にコミットすることはありません。ですから、ユーザー自身がトランザクションを明示的にコミットする必要があります。しかし、このことにより「what if」分析が可能になります。つまり、特定の統計を更新した後、変更した内容を見るためにパッケージの再バインドを試行できるようになります。さらに、1 作業単位内で複数の再バインドを実行することも可能になります。

**REBIND** コマンドは、自動コミットが有効な場合には、トランザクションをコミットします。

このコマンドは以下の事柄を行います。

- パッケージを短時間で再作成できます。これによりユーザーは、元のバインド・ファイルを必要とせずに、システムにおける変更を利用することができます。例えば、特定の SQL 言語が新しく作成された索引を利用できるような場合には、**REBIND** コマンドがパッケージを再作成するのに使用できます。**REBIND** によって、**RUNSTATS** の実行後にパッケージを再作成することもできます。その結果、新規の統計を利用できるようになります。
- 作動不能パッケージを再作成できます。作動不能パッケージは、バインド・ユーティリティーまたは再バインド・ユーティリティーのどちらかを呼び出すことにより、明示的に再バインドしなければなりません。パッケージが依存する機能インスタンスがドロップされると、パッケージは作動不能とマークされます (**SYSCAT.PACKAGES** システム・カタログの **VALID** 列は、X と設定されます。)
- 無効パッケージの再バインドに関する制御がユーザーに与えられます。無効パッケージは、実行される際にデータベース・マネージャーによって、自動的に (または暗黙的に) 再バインドされます。これは、その結果無効パッケージの最初の SQL 要求の実行を遅らせる可能性があります。暗黙の再バインドが失敗した場合に、初期遅延を無くし、戻される予期していない SQL エラー・メッセージを防ぐためには、システムが自動的に再バインドできるようにするのではなく、無効なパッケージを明示的に再バインドすることが必要とされます。例えば、データベース・アップグレードの後、データベースに格納されているすべてのパッケージが、**UPGRADE DATABASE** コマンドによって無効にされます。これは多数のパッ

## REBIND

ージを含んでいる場合には、一度に無効パッケージのすべてを明示的に再バインドする必要があります。この明示的な再バインドは、**BIND**、**REBIND**、または **db2rbind** ツールを使用して行うことができます。

パッケージに複数のバージョン (同じパッケージ名と作成者を持つ数多くのバージョン) が存在する場合は、一度に 1 つのバージョンしか再バインドできません。**VERSION** オプションでバージョンが指定されない場合、パッケージのバージョンはデフォルトで "" になります。同じ名前を持つパッケージが 1 つしか存在しない場合でも、そのバージョンが、指定されたバージョンまたはデフォルトのバージョンと一致しない限り、再バインドは行われません。

パッケージを明示的に再バインドするのに、**BIND** と **REBIND** のどちらを使用すべきかは、環境によって異なります。特に **BIND** を使用する理由がない限り、**REBIND** を使用するようにしてください。これは、**REBIND** の方が **BIND** よりもパフォーマンスの点で非常に優れているためです。ただし、以下の場合には必ず **BIND** を使用してください。

- プログラムに修正が加えられている場合 (例えば、SQL ステートメントが追加または削除された場合、またはパッケージがそのプログラムの実行可能モジュールと一致しない場合など)。
- 再バインドにおいて、**REBIND** コマンドでサポートされないバインド・オプションを変更する場合。**REBIND** コマンドは、すべてのバインド・オプションをサポートしているわけではありません。例えば、バインド処理の過程でパッケージに対する特権が付与されるようにする場合、**BIND** コマンドには **GRANT** オプションがあるため、このコマンドを使用する必要があります。
- パッケージが現在ではデータベース内に存在していない場合。
- すべての バインド・エラーを検出する必要がある場合。**REBIND** は、検出される最初のエラーのみ戻しますが、**BIND** コマンドはバインド中に発生する、最初の 100 のエラーを戻します。

**REBIND** は DB2 Connect によってサポートされています。

他のユーザーが使用中のパッケージで **REBIND** が実行された場合、他のユーザーの作業論理単位が終了するまで、再バインドは起こりません。これは、再バインド中に **SYSCAT.PACKAGES** システム・カタログ表中のパッケージのレコードで、排他ロックが掛けられるためです。

**REBIND** を実行すると、データベース・マネージャーは、**SYSCAT.STATEMENTS** システム・カタログ表に保管されている SQL ステートメントからパッケージを再作成します。

**REBIND** を実行してエラーが発生した場合、処理は停止し、エラー・メッセージが戻されます。

**REBIND** を実行するとパッケージが再び Explain されますが、その対象となるパッケージは、**EXPLSNAP** バインド・オプションを YES または ALL (パッケージの **SYSCAT.PACKAGES** カタログ表項目の **EXPLAIN\_SNAPSHOT** 列に示される) に設定して作成されたもの、および **EXPLAIN** バインド・オプションを YES または ALL (パッケージの **SYSCAT.PACKAGES** カタログ表項目の **EXPLAIN\_MODE** 列に示さ

れる) に設定して作成されたものです。使用される Explain 表は、**REBIND** を要求したユーザーのものであり、最初にバインドを実行したユーザーのものではありません。

SQL ステートメントに誤りがあることが検出され、**BIND** オプションの **SQLERROR CONTINUE** を指定していた場合、問題が修正されたとしても、そのステートメントには無効とマークされます。**REBIND** しても、ステートメントが無効の状態は変更できません。**VALIDATE RUN** でバインドされたパッケージの場合、ステートメントは、**REBIND** 実行時にオブジェクトが存在するかどうか、または権限の問題があるかどうかに応じて、**REBIND** 全体において静的バインドから追加バインドに変更されたり、追加バインドから静的バインドに変更されたりします。

**REOPT** に **ONCE** または **ALWAYS** を指定してパッケージを再バインドすると、静的ステートメントおよび動的ステートメントのコンパイル数とパフォーマンスが変化することがあります。

**REOPT** を指定しない場合、**REBIND** は **PRECOMPILE** 時または **BIND** 時に使用された既存の **REOPT** 値を保持します。

## RECOVER DATABASE

データベースを、特定のポイント・イン・タイムまで、またはログの終わりまでリストアおよびロールフォワードします。

### 有効範囲

パーティション・データベース環境では、このコマンドはカタログ・パーティションからのみ呼び出すことができます。ポイント・イン・タイムへのデータベース・リカバリー操作は、db2nodes.cfg ファイルにリストされているすべてのデータベース・パーティションに影響を与えます。ログの終わりへのデータベース・リカバリー操作は、指定されたデータベース・パーティションに影響を与えます。パーティションが指定されない場合、コマンドは、db2nodes.cfg ファイルにリストされているすべてのデータベース・パーティションに影響を与えます。

DB2 pureScale 環境では、**RECOVER DATABASE** コマンドをどのメンバーからでも発行できます。

### 許可

既存のデータベースにリカバリーするには、次のいずれかの権限が必要です。

- SYSADM
- SYSCTRL
- SYSMANT

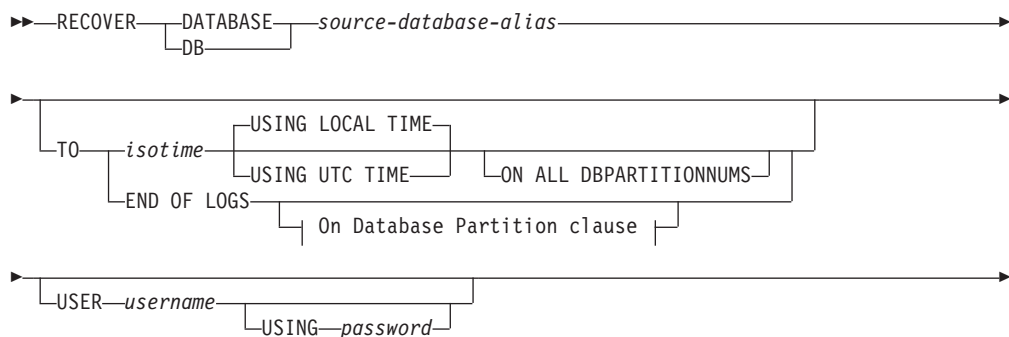
新規のデータベースにリカバリーするには、次のいずれかの権限が必要です。

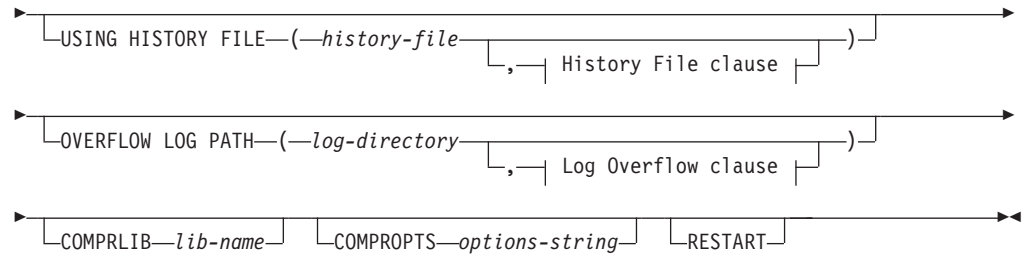
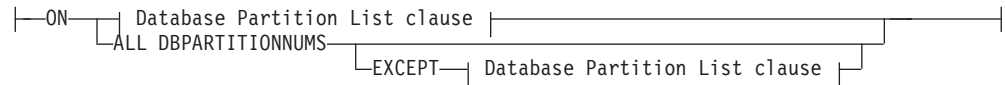
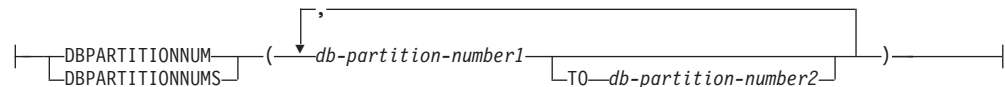
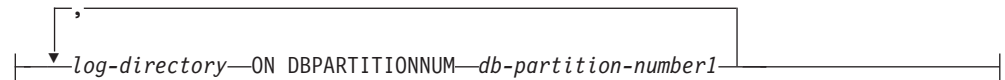
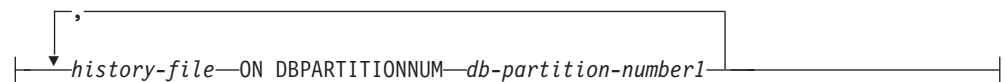
- SYSADM
- SYSCTRL

### 必要な接続

既存のデータベースをリカバリーするには、データベース接続が必要です。このコマンドを呼び出せば、指定したデータベースへの接続が自動的に確立され、リカバリー操作が終了すると接続が解放されます。新しいデータベースにリカバリーするには、インスタンス接続とデータベース接続が必要です。データベースを作成するには、インスタンス接続が必要です。

### コマンド構文



**On Database Partition clause:****Database Partition List clause:****Log Overflow clause:****History File clause:****コマンド・パラメーター****DATABASE** *database-alias*

リカバリーするデータベースの別名。

**TO**

*isotime* コミットされたすべてのトランザクション (その時点の前にコミットされたすべてのトランザクションのほかに、ちょうどその時点でコミットされたトランザクションを含む) がリカバリーされるポイント・イン・タイム。特定時点までのリカバリー操作で成功メッセージが返されるのは、それより大きいタイム・スタンプ値のトランザクションがログ・ファイルにある場合のみです。それより大きいタイム・スタンプのトランザクションがない場合であっても、**COMPLETE** オプションを指定したロールフォワード操作を実行できません。

この値は、タイム・スタンプとして指定されます。これは、日付と時刻の組み合わせを指定する 7 つの部分からなる文字ストリングで

す。形式は `yyyy-mm-dd-hh.mm.ss.nnnnnn` (年、月、日、時、分、秒、マイクロ秒) です。バックアップ・イメージのタイム・スタンプは、バックアップ操作が開始されたローカル時刻に基づいています。CURRENT TIMEZONE 特殊レジスターは、UTC とアプリケーション・サーバーのローカル時刻との差異を指定します。この差異は、時刻期間で表されます (最初の 2 文字が時間数を表し、次の 2 桁が分数を表し、最後の 2 桁が秒数を表す 10 進数です)。ローカル時刻から CURRENT TIMEZONE を減算すると、ローカル時刻を UTC に変換できます。

### USING LOCAL TIME

リカバリーに指定するポイント・イン・タイム。このオプションを指定することにより、ユーザーは UTC 時間ではなくサーバーの現地時間を使用して特定のポイント・イン・タイムにリカバリーできます。これはデフォルト・オプションです。

#### 注:

1. ユーザーがリカバリーのために現地時間を指定した場合、ユーザーに戻されるすべてのメッセージも現地時間で表示されます。すべての時刻はサーバー上で変換されます。パーティション・データベース環境では、カタログ・データベース・パーティションで変換されます。
2. タイム・スタンプ・ストリングは、サーバー上で UTC に変換されるため、この時刻はクライアントではなく、サーバーのタイム・ゾーンのローカル時刻になります。クライアントとサーバーのタイム・ゾーンが異なっている場合、サーバーのローカル時刻が使用されます。
3. タイム・スタンプ・ストリングが夏時間調整のための時間変更に接近している場合、停止時刻が時間変更の前か後かを判別して、それを適切に指定することが大切です。
4. データベースをリカバリーするときには有効なタイム・スタンプを指定することが重要です。有効なタイム・スタンプは、パーティション・データベース・システムで最新のバックアップが完了した時刻になります。
5. **RECOVER DATABASE** コマンドを複数回発行するとき、2 回目以降の各コマンドで指定するタイム・スタンプは、その直前のコマンドで指定したタイム・スタンプより大きくする必要があります。

### USING UTC TIME

リカバリーに指定するポイント・イン・タイム。

### END OF LOGS

**logpath** で指定されたパスにあるすべてのログに加えて、**logarchmeth1** および **logarchmeth2** データベース構成パラメーターで指定された場所からリトリブできる最大番号のログ・チェーンにあるログから、すべてのコミット済みトランザクションをロールフォワードするように指定します。

### ON ALL DBPARTITIONNUMS

`db2nodes.cfg` ファイルに指定されているすべてのデータベース・パーティ

ションでトランザクションがロールフォワードされることを指定します。データベース・パーティション節が指定されていない場合、これがデフォルトです。

**EXCEPT**

データベース・パーティション・リストに指定されているものを除き、`db2nodes.cfg` ファイルに指定されているすべてのデータベース・パーティションに対してトランザクションがロールフォワードされることを指定します。

**ON DBPARTITIONNUM | ON DBPARTITIONNUMS**

データベース・パーティションのセットに対してデータベースをロールフォワードします。

*db-partition-number1*

データベース・パーティション・リスト内のデータベース・パーティション番号を指定します。

**TO** *db-partition-number2*

2 番目のデータベース・パーティション番号を指定し、*db-partition-number1* から *db-partition-number2* までのすべてのデータベース・パーティションがデータベース・パーティション・リストに含まれるようにします。

**USER** *username*

データベースがリカバリーされる際のユーザー名。

**USING** *password*

ユーザー名を認証するために使用するパスワード。パスワードを省略すると、ユーザーに入力を求めるプロンプトが出ます。

**USING HISTORY FILE** *history-file*

*history-file* **ON DBPARTITIONNUM**

パーティション・データベース環境で、異なる履歴ファイルを可能にします。

**OVERFLOW LOG PATH** *log-directory*

リカバリー中に、アーカイブ・ログを検索する代替のログ・パスを指定します。**logpath** データベース構成パラメーターによって指定されている以外の場所にログ・ファイルが移動された場合には、このパラメーターを使用してください。パーティション・データベース環境では、これはすべてのデータベース・パーティションの (完全修飾) デフォルト・オーバーフロー・ログ・パスになります。単一パーティション・データベースには、相対オーバーフロー・ログ・パスを指定できます。

**OVERFLOW LOG PATH** コマンド・パラメーターは、データベース構成パラメーター **overflowlogpath** の値 (存在する場合) を上書きします。

*log-directory* **ON DBPARTITIONNUM**

パーティション・データベース環境では、これにより、特定のデータベース・パーティションのデフォルト・オーバーフロー・ログ・パスを別のログ・パスでオーバーライドできます。

**COMPRLIB** *lib-name*

解凍を実行するために使用するライブラリーの名前。この名前は、サーバー



## RECOVER DATABASE

上の 1 個のファイルを参照する完全修飾パスでなければなりません。このパラメーターを指定しない場合、DB2 はイメージ内に格納されているライブラリーの使用を試みます。バックアップが圧縮されていなかった場合、このパラメーターの値は無視されます。指定されたライブラリーをロードできない場合、リストア操作は失敗します。

### COMPROPTS *options-string*

バイナリー・データのうち、解凍ライブラリーの初期設定ルーチンに渡すブロックを記述します。DB2 データベース・システムはこのストリングをクライアントからサーバーに直接渡すため、バイト反転やコード・ページ変換の問題がある場合は、解凍ライブラリーで処理されます。データ・ブロックの最初の文字が「@」なら、データの残りの部分は、サーバー上に存在するファイルの名前を指定するものとして、DB2 データベース・システムは解釈します。その場合、DB2 データベース・システムは *string* の内容をこのファイルの内容で置き換え、新しい値を初期設定ルーチンに渡します。ストリングの最大長は 1 024 バイトです。

### RESTART

**RESTART** キーワードは、前のリカバリー操作が中断されたか、あるいは完了しなかった場合に使用できます。後続の **RECOVER DATABASE** コマンドは、可能であれば前のリカバリー操作を再開しようとします。**RESTART** キーワードを使用すると、フレッシュ・リストアから開始して指定の時点までロールフォワードするようにリカバリー操作に強制します。

### 例

以下の例は、単一パーティション・データベース環境または DB2 pureScale 環境に適用されます。すなわち、リカバリーするデータベースが現在存在しており、データベース・マネージャー構成ファイル (**dftdbpath** パラメーター) で指定されたデフォルトのデータベース・パスで、最新バージョンのヒストリー・ファイルが使用可能である状況です。

1. 最新のバックアップ・イメージを使用し、すべてデフォルト値を使用してログの終わりまでロールフォワードするには、

```
RECOVER DB SAMPLE
```

2. データベースをある時点 (時刻ポイント) までリカバリーするには、次のコマンドを発行します。指定された時点に達するまで、使用できる最新のイメージがリストアされ、ログが適用されます。

```
RECOVER DB SAMPLE TO 2001-12-31-04.00.00
```

3. 次のコマンドを発行します。履歴ファイルの保存バージョンを使用してデータベースをリカバリーするには、次のコマンドを発行します。例えば、現在の履歴ファイルに含まれていないような非常に古い時点までリカバリーすることが必要な場合、ユーザーはその期間以降を含む履歴ファイルのバージョンを提供する必要があります。ユーザーがその期間以降の履歴ファイルを保存している場合、そのバージョンを使用してリカバリーを実行できます。

```
RECOVER DB SAMPLE TO 1999-12-31-04.00.00  
USING HISTORY FILE (/home/user/old1999files/db2rhist.asc)
```



単一パーティション・データベース環境で、リカバリーするデータベースが存在していない場合は、**USING HISTORY FILE** 節を使用して履歴ファイルを指定する必要があります。

- 履歴ファイルのバックアップを作成していなかったため、バックアップ・イメージ内のコピーしか利用できるバージョンがない場合には、コマンド **RESTORE DATABASE** を発行し、続いて **ROLLFORWARD** を発行することをお勧めします。一方、**RECOVER DATABASE** コマンドを使用する場合は、まず、イメージ内の履歴ファイルをどこかの場所 (例えば /home/user/fromimage/db2rhist.asc) に抽出してから、このコマンドを発行する必要があります。(このバージョンの履歴ファイルには、ロールフォワードに必要なログ・ファイルに関する情報が含まれていないため、この履歴ファイルは **RECOVER DATABASE** には適していません)。

```
RECOVER DB SAMPLE TO END OF LOGS
USING HISTORY FILE (/home/user/fromimage/db2rhist.asc)
```

- 履歴のバックアップ・コピーを周期的に、あるいは頻繁に作成している場合は、**USING HISTORY FILE** 節を使用して、履歴ファイルのこのバージョンを指定するようにしてください。ファイルが /home/user/myfiles/db2rhist.asc なら、発行するコマンドは次のようになります。

```
RECOVER DB SAMPLE TO 2001-12-31-04.00.00
USING HISTORY FILE (/home/user/myfiles/db2rhist.asc)
```

(この場合、要求されたポイント・イン・タイム (PIT) より前に取られたバックアップが含まれている限り、履歴ファイルの任意のコピーを使用できます。最新のものである必要はありません。)

パーティション・データベース環境で、データベースがすべてのデータベース・パーティション上に存在し、すべてのデータベース・パーティション上の **dftdbpath** に使用できる履歴ファイルが存在する場合、

- すべてのデータベース・パーティションでデータベースを特定の時点までリカバリーする場合。DB2 データベース・システムは、リストア操作を開始する前に、その PIT がどのデータベース・パーティションでも到達可能であるかどうかを検証します。

```
RECOVER DB SAMPLE TO 2001-12-31-04.00.00
```

- すべてのデータベース・パーティションでデータベースをこの時点までリカバリーする場合。DB2 データベース・システムは、リストア操作を開始する前に、指定された時点がどのデータベース・パーティションでも到達可能であるかどうかを検証します。各データベース・パーティションでのリカバリー操作は、単一パーティションのリカバリーの場合と同じです。

```
RECOVER DB SAMPLE TO END OF LOGS
```

- dftdbpath** に履歴ファイルの最新バージョンがありますが、複数の特定の履歴ファイルを使用することが望ましい場合があるかもしれません。特に指定しない場合、各データベース・パーティションでは /home/user/oldfiles/db2rhist.asc にローカルに存在する履歴ファイルが使用されます。例外はデータベース・パーティション 2 と 4 です。データベース・パーティション 2 では /home/user/node2files/db2rhist.asc が使用され、データベース・パーティション 4 では /home/user/node4files/db2rhist.asc が使用されます。

## RECOVER DATABASE

```
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)
```

- すべてのデータベース・パーティションではなくデータベース・パーティションのサブセットをリカバリーすることは可能ですが、その場合はポイント・イン・タイムのリカバリー操作は実行できず、リカバリーはログの最後まで実行する必要があります。

```
RECOVER DB SAMPLE TO END OF LOGS ON DBPARTITIONNUMS(2 TO 4, 7, 9)
```

パーティション・データベース環境で、データベースが存在しない場合、

- 履歴ファイルのバックアップを作成していなかったため、バックアップ・イメージ内のコピーしか利用できるバージョンがない場合には、**RESTORE DATABASE** を発行し、続いて **ROLLFORWARD** を発行することをお勧めします。一方、**RECOVER DATABASE** を使用する場合は、まず、イメージ内の履歴ファイルをどこかの場所 (例えば /home/user/fromimage/db2rhist.asc) に抽出してから、このコマンドを発行する必要があります。(履歴ファイルのそのバージョンには、ロールフォワードに必要なログ・ファイルに関する情報が含まれていないため、その履歴ファイルはリカバリーには適していません。)

```
RECOVER DB SAMPLE TO 2001-12-31-04.00.00
USING HISTORY FILE (/home/user/fromimage/db2rhist.asc)
```

- 履歴のバックアップ・コピーを周期的に、あるいは頻繁に作成している場合は、**USING HISTORY FILE** 節を使用して、履歴ファイルのこのバージョンを指定するようにしてください。ファイルが /home/user/myfiles/db2rhist.asc なら、次のコマンドを発行できます。

```
RECOVER DB SAMPLE TO END OF LOGS
USING HISTORY FILE (/home/user/myfiles/db2rhist.asc)
```

### 使用上の注意

- データベースをリカバリーするためには、磁気テープ装置を使用したロード・リカバリーが必要になる場合があります。別のテープを求める要求が出された場合は、次のいずれか 1 つで応答できます。
  - c 続行。警告メッセージを生成した装置の使用を続けます (例えば、新しいテープをマウントしたときなど)。
  - d 装置の終了。警告メッセージを生成した装置の使用を停止します (例えば、それ以上テープがない場合)。
  - t 終了。すべての装置を終了します。
- リカバリー操作のリストア部分において障害が発生した場合は、**RECOVER DATABASE** コマンドを再発行することができます。リストア操作が成功したが、ロールフォワード操作中にエラーが発生した場合は、リカバリー操作全体をやり直す必要はなく (それには非常に時間がかかる)、**ROLLFORWARD DATABASE** コマンドを発行することができます。
- パーティション・データベース環境で、リカバリー操作のリストア部分でエラーが発生した場合、単一データベース・パーティションでのエラーでしかないという可能性があります。**RECOVER DATABASE** コマンドを再発行するとすべてのデータベース・パーティションでデータベースがリストアされますが、その代わり

に、障害が発生したデータベース・パーティションに関する **RESTORE DATABASE** を発行してから、**ROLLFORWARD DATABASE** コマンドを発行するほうが効率的です。

- DB2 pureScale 環境では、(データベース・パーティション・リスト節またはログ・オーバーフロー節のいずれかの部分として) **ON DBPARTITIONNUMS** 節または **ON DBPARTITIONNUM** 節を指定して **RECOVER DATABASE** コマンドを使用する場合は、データベース・パーティション 0 を指定する必要があります。

## REDISTRIBUTE DATABASE PARTITION GROUP

データベース・パーティション・グループ内のパーティション間でデータを再配分します。このコマンドはデータベース・パーティション・グループにあるすべてのオブジェクトに影響を及ぼし、1つのオブジェクトだけに限定することはできません。

このコマンドを発行できるのは、カタログ・データベース・パーティションからのみです。どのデータベース・パーティションが各データベースのカタログ・データベース・パーティションになっているかを判別するには、**LIST DATABASE DIRECTORY** コマンドを使用します。

### 有効範囲

このコマンドは、データベース・パーティション・グループ内のすべてのデータベース・パーティションに影響を与えます。

### 許可

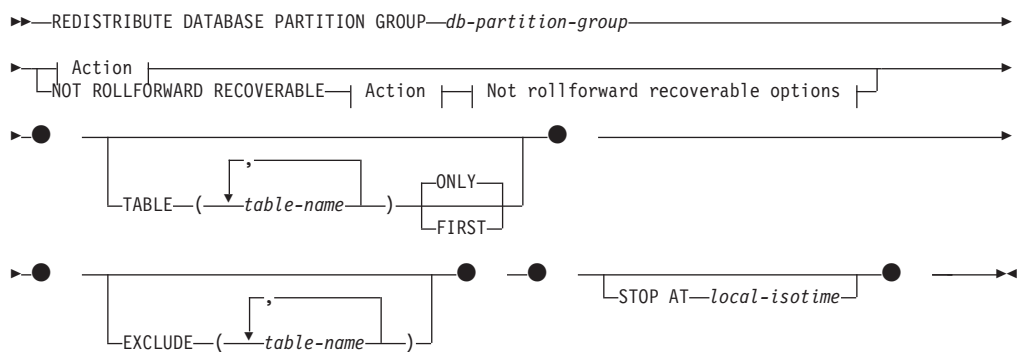
以下のいずれかの権限が必要です。

- SYSADM
- SYSCTRL
- DBADM

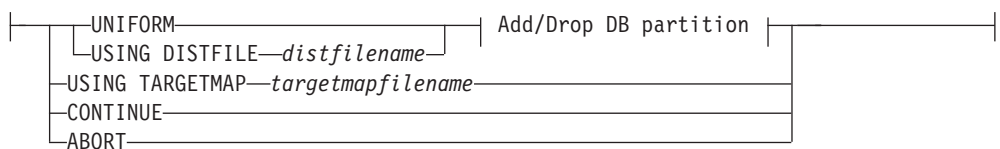
さらに、以下の権限のグループのいずれかも必要です。

- 再配分されるデータベース・パーティション・グループ内のすべての表に対する DELETE、INSERT、および SELECT 特権。
- DATAACCESS 権限

### コマンド構文

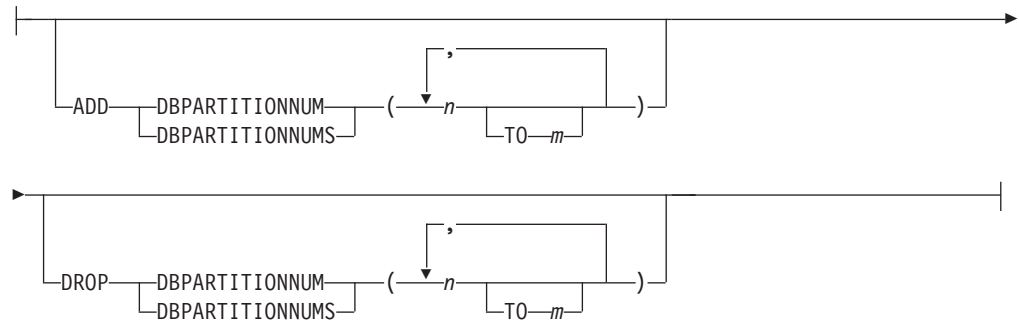


### Action:

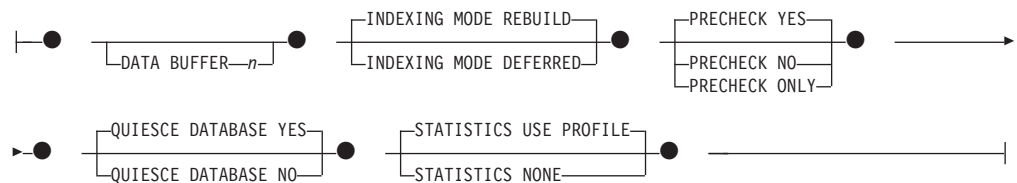


## REDISTRIBUTE DATABASE PARTITION GROUP

### Add/Drop DB partition:



### Not rollforward recoverable options:



## コマンド・パラメーター

### DATABASE PARTITION GROUP *db-partition-group*

データベース・パーティション・グループの名前。この 1 部構成の名前は、SYSCAT.DBPARTITIONGROUPS カタログ表に記述されたデータベース・パーティション・グループを識別します。データベース・パーティション・グループは、現在再配分を受けることはできません。

注: IBMCATGROUP および IBMTEMPGROUP データベース・パーティション・グループ内の表を再配分することはできません。

### NOT ROLLFORWARD RECOVERABLE

このオプションを使用すると、**REDISTRIBUTE DATABASE PARTITION GROUP** コマンドはロールフォワード・リカバリー可能ではありません。

- データは、内部での挿入および削除操作によってではなく、一括して移動されます。これにより、表のスキャンおよびアクセスの回数が減り、パフォーマンスが向上します。
- 挿入および削除操作それぞれに対するログ・レコードは必要ではなくなりました。このため、データの再配分を実行するときに、システム内で大容量のアクティブ・ログ・スペースおよびログ・アーカイブ・スペースを管理する必要がなくなりました。
- **REDISTRIBUTE DATABASE PARTITION GROUP** コマンドを **NOT ROLLFORWARD RECOVERABLE** オプションとともに使用する場合、再配分操作は XML 列の入った表に対して **INDEXING MODE DEFERRED** オプションを使用します。表に XML 列が含まれていない場合、再配分操作はコマンドの発行時に指定された索引付けモードを使用します。

## REDISTRIBUTE DATABASE PARTITION GROUP

このオプションが使用されない 場合には、すべての行移動に関する詳細なロギングが実行されるため、中断やエラーが起きた場合、またはほかにビジネス上の必要が生じた場合に、データベースを後からリカバリーすることができます。

### UNIFORM

データがハッシュ・パーティション間で均等に配分されることを指定します (つまり、それぞれのハッシュ・パーティションが同じ数の行を持つことが想定されます)。しかし、それぞれのデータベース・パーティションに同じ数のハッシュ・パーティションはマップされません。再配分後、データベース・パーティション・グループのすべてのデータベース・パーティションは、ほぼ同じ数のハッシュ・パーティションを持っています。

### USING DISTFILE *distfilename*

分散キー値の分散に偏りがある場合、このオプションを使用して、データベース・パーティション・グループのデータベース・パーティション全体にわたるデータの均一な再分散を行います。

*distfilename* を使用して、32 768 個のハッシュ・パーティションにわたる現行のデータの配分を指示します。

行カウント、バイト・ボリューム、または他の任意の尺度を使用して、各ハッシュ・パーティションで表示されたデータ量を示します。ユーティリティーは、パーティションに関連する整数値をそのパーティションの重みとして読み取ります。*distfilename* を指定した場合、ユーティリティーはターゲット分散マップを生成します。このマップは、データベース・パーティション・グループのデータベース・パーティション全体においてデータをできるだけ均一に再配分するために使用されます。再配分した後は、データベース・パーティション・グループ内の各データベース・パーティションの重みが、ほぼ同じになります (データベース・パーティションの重みは、そのデータベース・パーティションにマップするすべてのハッシュ・パーティションの重みの合計です)。

例えば、入力配布ファイルに以下の項目があるとします。

```
10223
1345
112000
0
100
...
```

例の中で、ハッシュ・パーティション 2 は 112000 の重みを持ち、パーティション 3 (重さは 0) には、マッピングするデータがまったくありません。

*distfilename* には、32 768 の正整数値が文字形式で入っていなければなりません。値の合計は、4 294 967 295 以下である必要があります。

*distfilename* のパスが指定されていない場合、現行ディレクトリーが使用されます。

### USING TARGETMAP *targetmapfilename*

*targetmapfilename* で指定されたファイルは、ターゲット分散マップとして使用されます。データの再配分はこのファイルに従って行われます。パスを指定していない場合、現行ディレクトリーが使用されます。

*targetmapfilename* には 32 768 個の整数を格納する必要があり、それぞれは有効なデータベース・パーティション番号を表します。各行の番号によって、ハッシュ値が特定のデータベース・パーティションにマップされます。つまり、行

## REDISTRIBUTE DATABASE PARTITION GROUP

$X$  に値  $Y$  が入っている場合、HASHEDVALUE() が  $X$  のすべてのレコードはデータベース・パーティション  $Y$  に配置されます。

ターゲット・マップに含まれるデータベース・パーティションがデータベース・パーティション・グループ中に存在しないと、エラーが戻されます。

**REDISTRIBUTE DATABASE PARTITION GROUP** コマンドを実行する前に、ALTER DATABASE PARTITION GROUP ADD DBPARTITIONNUM ステートメントを実行してください。

ターゲット・マップから除外されたデータベース・パーティションが、データベース・パーティション・グループにある場合、そのデータベース・パーティションはパーティションの中に含まれていません。このようなデータベース・パーティションは、**REDISTRIBUTE DATABASE PARTITION GROUP** コマンドの前か後に ALTER DATABASE PARTITION GROUP DROP DBPARTITIONNUM ステートメントを使用することによってドロップできます。

### CONTINUE

直前に失敗または停止した **REDISTRIBUTE DATABASE PARTITION GROUP** 操作を継続します。何も起こらなければ、エラーが戻されます。

### ABORT

直前に失敗または停止した **REDISTRIBUTE DATABASE PARTITION GROUP** 操作をアボートします。何も起こらなければ、エラーが戻されます。

### ADD

#### DBPARTITIONNUM $n$

##### TO $m$

$n$  または  $n$  TO  $m$  では、データベース・パーティション・グループに追加するデータベース・パーティション番号のリストを指定します。指定するパーティションは、データベース・パーティション・グループにすでに定義済みであってはなりません (SQLSTATE 42728)。ADD DBPARTITIONNUM 節を指定して ALTER DATABASE PARTITION GROUP ステートメントを実行する操作と等価です。

#### DBPARTITIONNUMS $n$

##### TO $m$

$n$  または  $n$  TO  $m$  では、データベース・パーティション・グループに追加するデータベース・パーティション番号のリストを指定します。指定するパーティションは、データベース・パーティション・グループにすでに定義済みであってはなりません (SQLSTATE 42728)。ADD DBPARTITIONNUM 節を指定して ALTER DATABASE PARTITION GROUP ステートメントを実行する操作と等価です。

### 注:

1. このオプションを使用してデータベース・パーティションを追加すると、表スペースのコンテナは、データベース・パーティション・グループ内で最も小さい番号を持つ既存のパーティション内の、対応する表スペースのコンテナに基づくこととなります。その結果、コンテナの名前の競合が発生する場合は、このオプションを使用しないでください。



## REDISTRIBUTE DATABASE PARTITION GROUP

い (新しいパーティションが既存のコンテナと同じ物理マシンにあると、そのような名前の競合が発生する可能性があります)。そのような場合は、**WITHOUT TABLESPACES** オプションを指定して **ALTER DATABASE PARTITION GROUP** ステートメントを使用してから、**REDISTRIBUTE DATABASE PARTITION GROUP** コマンドを実行してください。その後、適切な名前を指定して、表スペース・コンテナを手動で作成できます。

2. **ADD DBPARTITIONNUMS** パラメーターを指定すると、データ再配分によって、すべての新規データベース・パーティションに表スペースが作成される可能性があります。

### DROP

#### **DBPARTITIONNUM** *n*

**TO** *m*

*n* または *n TO m* では、データベース・パーティション・グループからドロップするデータベース・パーティション番号のリストを指定します。指定するパーティションは、データベース・パーティション・グループにすでに定義されている必要があります (SQLSTATE 42729)。**DROP DBPARTITIONNUM** 節を指定して **ALTER DATABASE PARTITION GROUP** ステートメントを実行する操作と等価です。

#### **DBPARTITIONNUMS** *n*

**TO** *m*

*n* または *n TO m* では、データベース・パーティション・グループからドロップするデータベース・パーティション番号のリストを指定します。指定するパーティションは、データベース・パーティション・グループにすでに定義されている必要があります (SQLSTATE 42729)。**DROP DBPARTITIONNUMS** 節を指定して **ALTER DATABASE PARTITION GROUP** ステートメントを実行する操作と等価です。

#### **TABLE** *tablename*

再配分処理する表の順番を指定します。

#### **ONLY**

表の順序の後に **ONLY** キーワード (デフォルト) を使用すると、指定した表だけが再配分の対象になります。残りの表は、**REDISTRIBUTE CONTINUE** コマンドによって後で処理できます。これはデフォルトです。

#### **FIRST**

表の順序の後に **FIRST** キーワードを使用すると、指定した表が指定の順序で再配分処理を受け、データベース・パーティション・グループ内の残りの表は、ランダムな順序で再配分処理を受けることになります。

#### **EXCLUDE** *tablename*

再配分処理をしない表を指定します。例えば、データ再配分の要件を表が満たすように構成できるようになるまで、その表を一時的に対象外にできます。対象から外した表は、**REDISTRIBUTE CONTINUE** コマンドを使用して後で処理できます。

#### **STOP AT** *local-isotime*

このオプションを指定すると、各表のデータ再配分を開始する前に、



## REDISTRIBUTE DATABASE PARTITION GROUP

*local-isotime* と現在のローカル・タイム・スタンプが比較されます。指定した *local-isotime* が現在のローカル・タイム・スタンプと同じか、それよりも早いと、ユーティリティは処理を停止して、警告メッセージを生成します。停止時に進行中であった表のデータ再配分の処理は中断されずに完了します。表の新規のデータ再配分の処理は開始されません。未処理の表の再配分を実行するには、**CONTINUE** オプションを使用します。この *local-isotime* 値は、日付と時刻の組み合わせを識別する 7 部構成の文字ストリングのタイム・スタンプとして指定します。形式は、現地時間の *yyyy-mm-dd-hh.mm.ss.nnnnnn* (年、月、日、時、分、秒、マイクロ秒) です。

### DATA BUFFER *n*

ユーティリティ内でデータを転送するためのバッファ・スペースとして使用する 4 KB ページの数を指定します。このコマンド・パラメーターは、**NOT ROLLFORWARD RECOVERABLE** パラメーターも指定された場合のみ使用できます。

指定された値がサポートされている最小値よりも小さい場合には、最小値が使用され、警告は戻されません。**DATA BUFFER** 値を指定しないと、実行時に各表の処理を開始する時点で、ユーティリティによって適切なデフォルトが計算されます。具体的には、表の再配分の開始時点でユーティリティ・ヒープで使用可能になっているメモリーの 50% を使用することを基本にしなが、さまざまな表プロパティを考慮に入れることによって、デフォルトを計算することになります。

このメモリーは、ユーティリティ・ヒープから直接に割り当てられ、そのサイズは **util\_heap\_sz** データベース構成パラメーターで修正可能です。システムにさらに使用可能なメモリーがある場合、**REDISTRIBUTE DATABASE PARTITION GROUP** コマンドの **DATA BUFFER** パラメーターの値は、一時的に **util\_heap\_sz** を超える場合があります。

### INDEXING MODE

再配分時の索引の保守方法を指定します。このコマンド・パラメーターは、**NOT ROLLFORWARD RECOVERABLE** パラメーターも指定された場合のみ使用できます。

有効な値は以下のとおりです。

#### REBUILD

索引が最初から再作成されます。このオプションを使用する場合は、索引が有効である必要はありません。このオプションを使用する結果として、「索引」ページがディスク上で一緒にクラスター化されます。

#### DEFERRED

再配分で索引の維持を試行しません。リフレッシュが必要であることを示すマークが索引に付けられます。そのような索引に最初にアクセスした時点で再作成が強制実行されるか、データベースの再始動時に索引が再作成されることとなります。

注: 非 MDC 表および非 ITC 表の場合、表に無効な索引があると、**INDEXING MODE DEFERRED** が指定されていないければ、**REDISTRIBUTE DATABASE PARTITION GROUP** コマンドによって自動的に索引が再作成されます。MDC 表または ITC 表の場合、**INDEXING MODE DEFERRED** が指定されている場合でも、無効な複合索引は表の再配分が始まる前に再作成されます。なぜなら、ユーティリティは MDC 表または ITC 表を処理するために複合索引を必要とするからです。

## REDISTRIBUTE DATABASE PARTITION GROUP

### PRECHECK

データベース・パーティション・グループを再配分できるかどうかを検査します。このコマンド・パラメーターは、**NOT ROLLFORWARD RECOVERABLE** パラメーターも指定された場合のみ使用できます。

### YES

これはデフォルト値です。再配分操作が開始されるのは、検査が正常に終了した場合のみです。検査に失敗すると、このコマンドは終了して、失敗した最初の検査に関連したエラー・メッセージを返します。

**NO** 再配分操作はすぐに開始されます。検査は行われません。

### ONLY

検査実行後に、コマンドは終了します。再配分は行われません。デフォルトでは、データベースは静止されません。**QUIESCE DATABASE** コマンド・パラメーターが **YES** に設定された場合、またはデフォルト値の **YES** の場合には、データベースは静止状態のままになります。データベースへの接続を復元するには、再配分操作を実行するか、**UNQUIESCE DATABASE** コマンドを発行します。

### QUIESCE DATABASE

データベースからすべてのユーザーを強制的に切断して静止モードにするように指定します。このコマンド・パラメーターは、**NOT ROLLFORWARD RECOVERABLE** パラメーターも指定された場合のみ使用できます。

### YES

これはデフォルト値です。**SYSADM**、**SYSMAINT**、または **SYSCTRL** の権限を持つユーザーと、**QUIESCE\_CONNECT** 権限を付与されたユーザーだけが、データベースとそのオブジェクトにアクセスできるようになります。再配分が正常に完了すると、データベースは静止解除されます。

**NO** 再配分操作によって、データベースが静止状態になることはありません。ユーザーがデータベースから強制的に切断されることもありません。

詳しくは、**QUIESCE DATABASE** コマンドを参照してください。

### STATISTICS

ユーティリティーが、統計プロファイルのある表の統計を収集するように指定します。このコマンド・パラメーターは、**NOT ROLLFORWARD RECOVERABLE** パラメーターも指定された場合のみ使用できます。

このオプションを指定するほうが、データ再配分の完了後に **RUNSTATS** コマンドを別途実行するよりも効率的です。

### USE PROFILE

統計プロファイルのある表の統計を収集します。統計プロファイルのない表については、何も実行されません。これはデフォルトです。

### NONE

表の統計を収集しません。

## 例

データ分散ファイル `distfile_for_dbpg_1` によって現在のデータ分散を提供することにより、データベース・パーティション・グループ `DBPG_1` を再配分します。データを 2 つの新しいデータベース・パーティション 6 および 7 に移動します。

```
REDISTRIBUTE DATABASE PARTITION GROUP DBPG_1
  USING DISTFILE /home/user1/data/distfile_for_dbpg_1
  ADD DATABASE PARTITION (6 TO 7)
```

データベース・パーティション・グループ `DBPG_2` に対して以下のような再配分を行います。

- 再配分はロールフォワード・リカバリー可能ではない。
- データは複数のハッシュ・パーティションにわたって一様に配分される。
- 索引は最初から再作成される。
- 統計は収集されない。
- データ転送のためのバッファ・スペースとして、180,000 個の 4 KB ページが使用される。

```
REDISTRIBUTE DATABASE PARTITION GROUP DBPG_2
  NOT ROLLFORWARD RECOVERABLE
  UNIFORM
  INDEXING MODE REBUILD
  DATA BUFFER 180000
  STATISTICS NONE
```

この再配分操作では、**QUIESCE DATABASE** および **PRECHECK** コマンド・パラメーターのデフォルト値により、データベースの静止および事前検査も実行されます。

## 使用上の注意

- 再配分操作を開始する前に、表が通常状態にあり、「ロード・ペンディング」状態でも「REORG ペンディング」状態でもないことを確認してください。表状態は **LOAD QUERY** コマンドを使って調べることができます。
- **NOT ROLLFORWARD RECOVERABLE** オプションが指定され、データベースがリカバリー可能である場合、ユーティリティーが最初に表スペースにアクセスした時点で、表スペースは **BACKUP PENDING** 状態になります。表スペースのバックアップが作成されるまで、その表スペースに含まれているすべての表は読み取り専用になります。表スペースのバックアップは、その表スペース内のすべての表の再配分処理が完了したときのみ可能になります。
- 再配分操作の実行中に、その再配分操作に関する一般情報、および各表の処理開始時刻と処理終了時刻などの情報を含むイベント・ログ・ファイルが作成されます。このイベント・ログ・ファイルは、以下のように書き込まれます。
  - Linux および UNIX オペレーティング・システムの場合は、`homeinst/sqllib/redis` ディレクトリー。サブディレクトリーとファイル名の形式は、`database-name.database-partition-group-name.timestamp.log` になります。
  - Windows オペレーティング・システムの場合は、**DB2INSTPROF**`¥instance¥redis` ディレクトリー (**DB2INSTPROF** は、**DB2INSTPROF** レジストリー変数の値です)。サブディレクトリーとファイル名の形式は、`database-name.database-partition-group-name.timestamp.log` になります。
  - タイム・スタンプ値は、コマンドが発行された時の時刻です。

## REDISTRIBUTE DATABASE PARTITION GROUP

- このユーティリティーは、処理中に断続的な COMMIT を実行します。
- 再配分を受けた表と従属関係があるすべてのパッケージは無効になります。データベース・パーティション・グループの再配分操作が完了した後で、そのようなパッケージを明示的に再バインドすることをお勧めします。明示的な再バインドにより、無効パッケージに対する最初の SQL 要求の実行での初期遅延がなくなります。再配分メッセージ・ファイルには、再配分を受けたすべての表のリストが入ります。
- 統計プロファイルがある表については、再配分ユーティリティーの実行時に、デフォルトで統計が更新されます。統計プロファイルがない表の場合は、表や索引の統計を別途更新することをお勧めします。そのためには、再配分操作の完了後に、db2Runstats API を呼び出すか、**RUNSTATS** コマンドを実行できます。
- 複製されたマテリアライズ照会表や DATA CAPTURE CHANGES を用いて定義された表を含むデータベース・パーティション・グループは、再配分することができません。
- データベース・パーティション・グループに、既存の宣言済み一時表または作成済み一時表を含む USER TEMPORARY 表スペースがある場合、再配分を行うことはできません。
- **INDEXING MODE** などのオプションは、適用対象にならない表では無視されます (警告も生成されません)。例えば、**INDEXING MODE** は、索引のない表では無視されます。
- **REDISTRIBUTE DATABASE PARTITION GROUP** コマンドは、データベース・パーティションの追加サーバー要求が保留中または進行中の場合は、失敗する可能性があります (SQLSTATE 55071)。さらに、新規データベース・パーティション・サーバーがオンラインでインスタンスに追加された場合、およびすべてのアプリケーションが新規データベース・パーティション・サーバーを認識しているわけではない場合、このコマンドは失敗する可能性があります (SQLSTATE 55077)。

### 互換性

DB2 パージョン 9.5 以前の XML レコード・フォーマットを使用する XML 列が入った表は、再配分できません。そうした表を新しいフォーマットに移行するには、ADMIN\_MOVE\_TABLE ストアド・プロシージャを使用します。

## REFRESH LDAP

Lightweight Directory Access Protocol (LDAP) の情報が変更された場合に、更新されたその情報に合わせてローカル・マシンのキャッシュを最新表示します。

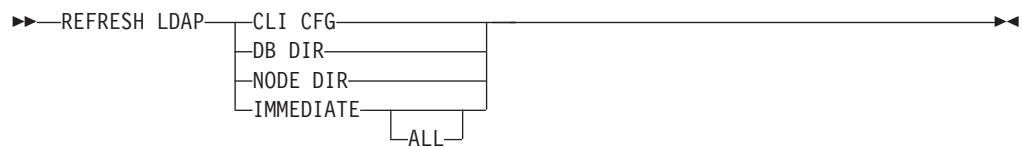
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### CLI CFG

CLI 構成を更新するよう指定します。このパラメーターは AIX または Solaris オペレーティング・システムではサポートされていません。

#### DB DIR

データベース・ディレクトリーを更新するよう指定します。

#### NODE DIR

ノード・ディレクトリーを更新するよう指定します。

#### IMMEDIATE

ローカル・データベースおよびノード・ディレクトリーを即座に更新するよう指定します。

**ALL** LDAP サーバー内に含まれているデータベースおよびノード項目すべてを、ローカル・データベースおよびノード・ディレクトリーに追加するよう指定します。

### 使用上の注意

最新表示中に LDAP のオブジェクトが除去されると、それに対応するローカル・マシンの LDAP 項目も除去されます。LDAP の情報が変更されると、それに対応する LDAP 項目もそれに応じて変更されます。DB2CLI.INI ファイルを手動で更新する場合は、**REFRESH LDAP CLI CFG** コマンドを実行して、現行ユーザーのキャッシュを更新してください。

**REFRESH LDAP DB DIR** および **REFRESH LDAP NODE DIR** コマンドは、ローカル・データベースまたはノード・ディレクトリーで検出される LDAP データベースまたはノード項目を削除します。データベース項目またはノード項目は、ユーザーが、LDAP で検出されるデータベースに接続したりインスタンスにアタッチしたりした場合

## REFRESH LDAP

に、**DB2LDAPCACHE** が未設定または YES に設定されていれば、ローカル・データベースまたはノード・ディレクトリーに再度追加されます。

**REFRESH LDAP IMMEDIATE** コマンドは、LDAP で検出された最新情報を使ってローカル・データベースおよびノード・ディレクトリーの項目を更新します。この更新は、**DB2LDAPCACHE** が有効になっているかどうかにかかわらず、ただちに実行されます。更新されるのは、LDAP に由来するデータベースおよびノード項目だけです。手動で追加された項目は未変更のままになります。

**REFRESH LDAP IMMEDIATE ALL** コマンドは、LDAP で検出されたすべての情報を、ローカル・データベースおよびノード・ディレクトリーに即座に読み込みます。LDAP で検出された項目が既存のローカル項目と一致する場合、コマンドは項目を更新します。この更新が行われるのは、ローカル項目が LDAP に由来する場合だけです。手動で追加された項目は未変更のままになります。この更新は、**DB2LDAPCACHE** が有効になっているかどうかにかかわらず、実行されます。

LDAP が使用不可であるときに **REFRESH LDAP IMMEDIATE** または **REFRESH LDAP IMMEDIATE ALL** のいずれかを実行すると、SQLCODE -3279 (LDAP が使用できないため、コマンドが正常に完了しませんでした) になります。

## REGISTER

ネットワーク・ディレクトリー・サーバーに DB2 サーバーを登録します。

### 許可

なし

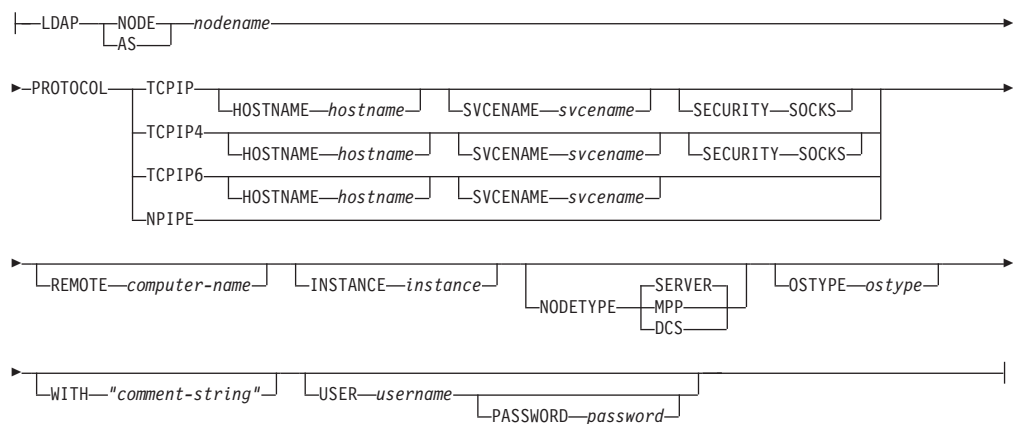
### 必要な接続

なし

### コマンド構文



### LDAP path:



### コマンド・パラメーター

**IN** DB2 サーバーを登録するネットワーク・ディレクトリー・サーバーを指定します。有効な値は、LDAP (Lightweight Directory Access Protocol) ディレクトリー・サーバーの場合、LDAP です。

### ADMIN

管理サーバー・ノードを登録することを指定します。

### NODE | AS *nodename*

LDAP 内の DB2 サーバーを表すショート・ネームを指定します。ノード項目は、このノード名によって LDAP にカタログされます。クライアントは、このノード名によってサーバーにアタッチできます。この LDAP ノード項目に関連付けられるプロトコルは、**PROTOCOL** パラメーターで指定します。

### PROTOCOL

LDAP ノード項目に関連付けるプロトコル・タイプを指定します。データベース・サーバーは複数のプロトコル・タイプをサポートできるため、この値にはクライアント・アプリケーションが実際に使用するプロトコル・タイプを指定します。DB2 サーバーはプロトコルごとに 1 つずつ登録しなければ



ばなりません。有効な値は、TCPIP、TCPIP4、TCPIP6、および NPIPE です。Windows の Named PIPE を使用する場合は、NPIPE を指定します。NPIPE は Windows オペレーティング・システムでのみサポートされています。

**HOSTNAME** *hostname*

TCP/IP ホスト名 (または IP アドレス) を指定します。TCP/IP プロトコルの使用時には、IP アドレスは、IPv4 のアドレスでも IPv6 のアドレスでもかまいません。TCPIP4 プロトコルの使用時には、IP アドレスは IPv4 アドレスでなければなりません。TCPIP6 プロトコルの使用時には、IP アドレスは IPv6 アドレスでなければなりません。

**SVCENAME** *svcename*

TCP/IP サービス名またはポート番号を指定します。

**SECURITY SOCKS**

TCP/IP SOCKS を使用することを指定します。このパラメーターは、IPv4 でのみサポートされます。プロトコルとして TCPIP が指定された場合、基礎となるプロトコルとして使用されるのは IPv4 です。

**REMOTE** *computer-name*

DB2 サーバーが常駐するマシンのコンピューター名を指定します。このパラメーターを指定する必要があるのは、リモート DB2 サーバーを LDAP に登録する場合だけです。この値は、サーバー・マシンを LDAP に追加したときに指定した値と同じでなければなりません。Windows オペレーティング・システムの場合、これはコンピューター名です。UNIX ベースのシステムの場合、これは TCP/IP ホスト名です。

**INSTANCE** *instance*

DB2 サーバーのインスタンス名を指定します。リモート・インスタンスの場合 (つまり、**REMOTE** パラメーターの値が指定されている場合)、インスタンス名は必ず指定しなければなりません。

**NODETYPE**

データベース・サーバーのノード・タイプを指定します。有効な値は以下のとおりです。

**SERVER**

DB2 Enterprise Server Edition の SERVER ノード・タイプを指定します。これはデフォルトです。

**MPP** DB2 Enterprise Server Edition - Extended (パーティション・データベース) サーバーの MPP ノード・タイプを指定します。

**DCS** ホスト・データベース・サーバーを登録するときに、DCS ノード・タイプを指定します。

**OSTYPE** *ostype*

サーバー・マシンのオペレーティング・システムのタイプを指定します。有効な値は、AIX、NT、HPUX、SUN、MVS、OS400、VM、VSE、および LINUX です。オペレーティング・システムのタイプが指定されない場合、ローカル・サーバーに対してはローカルのオペレーティング・システムのタイプが使用され、リモート・サーバーに対してはオペレーティング・システムのタイプは使用されません。



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

DB2 サーバーについて記述します。ネットワーク・ディレクトリーに登録されているサーバーについての記述を補足する、任意のコメントを入力できます。最大長は 30 文字です。復帰文字や改行文字は許可されません。コメント・テキストは必ず二重引用符で囲んでください。

**使用上の注意**

DB2 サーバーは、サーバーがサポートするプロトコルごとに 1 つずつ登録します。

DB2 サーバー・インスタンスごとに **REGISTER** コマンドを 1 回ずつ公開して、サーバーをディレクトリー・サーバーに登録してください。通信パラメーター・フィールドを再構成する場合や、サーバー・ネットワーク・アドレスを変更する場合には、ネットワーク・ディレクトリー・サーバーで DB2 サーバーを更新してください。

LDAP の DB2 サーバーを更新するには、変更が実施された後に **UPDATE LDAP NODE** コマンドを使用します。

DB2 サーバーをローカルに登録するときにプロトコル構成パラメーターが指定されていると、データベース・マネージャー構成ファイルに指定されている値がオーバーライドされます。

**REGISTER** コマンドを、ローカル DB2 インスタンスを LDAP に登録するために使用し、**NODETYPE** および **OSTYPE** の 1 つまたは両方を指定する場合、それらはローカル・システムから検索された値で置き換えられます。**REGISTER** コマンドを、リモート DB2 インスタンスを LDAP に登録するために使用し、**NODETYPE** および **OSTYPE** の 1 つまたは両方を指定しない場合、デフォルト値の **SERVER** と **Unknown** が使用されます。

**REGISTER** コマンドを使用してリモートの DB2 サーバーを LDAP に登録する場合、リモート・サーバーが使用する通信プロトコルとともに、リモート・サーバーのコンピューター名とインスタンス名も指定する必要があります。

ホスト・データベース・サーバーを登録する場合には、**NODETYPE** パラメーターに値 **DCS** を指定しなければなりません。

## REGISTER XMLSCHEMA

XML スキーマを XML スキーマ・リポジトリ (XSR) に登録します。

### 許可

以下の権限のいずれか。

- DBADM
- IMPLICIT\_SCHEMA データベース権限 (SQL スキーマが存在しない場合)。
- CREATEIN 特権 (SQL スキーマが存在する場合)。

### 必要な接続

データベース

### コマンド構文

```

▶▶ 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] ) ) |

```

### コマンド・パラメーター

#### *schema-URI*

登録される XML スキーマの URI を、XML インスタンス文書で参照されるとおりに指定します。

#### **FROM** *content-URI*

XML スキーマ文書が置かれている URI を指定します。ファイル・スキーム URI で指定されたローカル・ファイルだけがサポートされます。

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

XML スキーマのプロパティ文書の URI を指定します。ファイル・スキーム URI で指定されたローカル・ファイルだけがサポートされます。

#### **AS** *relational-identifier*

登録される XML スキーマを参照するために使用できる名前を指定します。リレーショナル名は 2 つの部分の SQL ID として指定することができます。これは、SQL スキーマと XML スキーマ名から成り、SQLschema.name というフォーマットを持ちます。スキーマが指定されない

場合、CURRENT SCHEMA 特殊レジスターで定義されたとおりに、デフォルトのリレーショナル・スキーマが使用されます。名前が提供されない場合、固有値が生成されます。

**COMPLETE**

これ以上の XML スキーマ文書は追加されないことを示します。これが指定される場合、スキーマの妥当性検査が行われ、エラーが見つからなければ使用できるものとしてマークされます。

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

XML スキーマのプロパティ文書の URI を指定します。ファイル・スキーム URI で指定されたローカル・ファイルだけがサポートされます。

**ENABLE DECOMPOSITION**

このスキーマが XML 文書の分解のために使用されることを指定します。

**ADD** *document-URI*

このスキーマに追加される XML スキーマ文書の URI を指定します。この文書は別の XML 文書から参照されることがあるからです。

**FROM** *content-URI*

XML スキーマ文書が置かれている URI を指定します。ファイル・スキーム URI で指定されたローカル・ファイルだけがサポートされます。

**WITH** *properties-URI*

XML スキーマのプロパティ文書の URI を指定します。ファイル・スキーム URI で指定されたローカル・ファイルだけがサポートされます。

**例**

```
REGISTER XMLSCHEMA 'http://myPOschema/PO.xsd'
FROM 'file:///c:/TEMP/PO.xsd'
WITH 'file:///c:/TEMP/schemaProp.xml'
AS user1.POschema
```

**使用上の注意**

- XML スキーマ文書を参照し、妥当性検査およびアノテーションのために使用できるようになるには、その前にまず XSR に登録する必要があります。このコマンドは、基本 XML スキーマ文書を登録することにより、XML スキーマ登録プロセスの最初のステップを実行します。XML スキーマ登録プロセスの最終ステップでは、**COMPLETE XMLSCHEMA** コマンドが XML スキーマに対して正常に実行される必要があります。あるいは、その他の XML スキーマ文書が組み込まれない場合、**COMPLETE** キーワードを指定して **REGISTER XMLSCHEMA** コマンドを発行し、登録を 1 ステップで完了してください。
- データベースで XML スキーマを登録する際に、XML スキーマのサイズによっては、より大きなアプリケーション・ヒープ (**applheapsz**) が必要になる場合があります。推奨されるサイズは 1024 ですが、スキーマが大きくなると追加メモリーが必要になります。

## REGISTER XSROBJECT

データベース・カタログに XML オブジェクトを登録します。サポートされるオブジェクトは、DTD および外部エンティティーです。

### 許可

以下の権限のいずれか。

- DBADM
- IMPLICIT\_SCHEMA データベース権限 (SQL スキーマが存在しない場合)。
- CREATEIN 特権 (SQL スキーマが存在する場合)。

### 必要な接続

データベース

### コマンド構文

```

▶▶ REGISTER XSROBJECT system-ID [PUBLIC public-ID] FROM content-URI
▶ [AS relational-identifier] [DTD EXTERNAL ENTITY]

```

### コマンド・パラメーター

*system-ID*

XML オブジェクト宣言で指定されているシステム ID を指定します。

**PUBLIC** *public-ID*

XML オブジェクト宣言内のオプションの PUBLIC ID を指定します。

**FROM** *content-URI*

XML スキーマ文書の内容が置かれている URI を指定します。ファイル・スキーム URI で指定されたローカル・ファイルだけがサポートされます。

**AS** *relational-identifier*

登録される XML オブジェクトを参照するために使用できる名前を指定します。リレーショナル名は 2 つの部分の SQL ID として指定することができます。これは、ピリオドで区切られたリレーショナル・スキーマと名前から成ります (例えば、"JOHNDOE.EMPLOYEEEDTD")。リレーショナル・スキーマが指定されない場合、特殊レジスター CURRENT SCHEMA で定義されているデフォルトのリレーショナル・スキーマが使用されます。名前を指定しない場合、自動的に生成されます。

**DTD** 登録されるオブジェクトがデータ・タイプ定義文書 (DTD) であることを指定します。

**EXTERNAL ENTITY**

登録されるオブジェクトが外部エンティティーであることを指定します。

### 例

1. 以下のサンプル XML 文書は外部エンティティーを参照します。

```
<?xml version="1.0" standalone="no" ?>
<!DOCTYPE copyright [
  <!ELEMENT copyright (#PCDATA)>
]
]>
<copyright>c</copyright>
```

この文書を正常に XML 列に挿入するには、その前に外部エンティティを登録する必要があります。以下のコマンドは、エンティティ・コンテンツがローカルの C:¥TEMP に保管されているエンティティを登録しています。

```
REGISTER XSROBJECT 'http://www.xmlwriter.net/copyright.xml'
FROM 'c:¥temp¥copyright.xml' EXTERNAL ENTITY
```

2. 以下の XML 文書フラグメントは DTD を参照します。

```
<!--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>
  ...
```

この文書を正常に XML 列に挿入するには、その前に DTD を登録する必要があります。以下のコマンドは、DTD 定義がローカルの C:¥TEMP に保管されており、DTD に関連付けるリレーショナル ID が "TEST.SUBJECTS" である DTD を登録します。

```
REGISTER XSROBJECT 'http://www.xmlwriter.net/subjects.dtd'
FROM 'file:///c:/temp/subjects.dtd' AS TEST.SUBJECTS DTD
```

3. 以下のサンプル XML 文書は public 外部エンティティを参照します。

```
<?xml version="1.0" standalone="no" ?>
<!DOCTYPE copyright [
  <!ELEMENT copyright (#PCDATA)>
]
]>
<copyright>c</copyright>
```

この文書を正常に XML 列に挿入するには、その前に public 外部エンティティを登録する必要があります。以下のコマンドは、エンティティ・コンテンツがローカルの C:¥TEMP に保管されているエンティティを登録しています。

```
REGISTER XSROBJECT 'http://www.w3.org/xmlspec/copyright.xml'
PUBLIC '-//W3C//TEXT copyright//EN' FROM 'file:///c:/temp/copyright.xml'
EXTERNAL ENTITY
```

## REORG INDEXES/TABLE

索引または表を再編成します。

フラグメント化されていない物理的に連続したページに索引データを再構築することによって、表に定義されたすべての索引を再編成することができます。データ・パーティション表では、パーティション表で特定の非パーティション索引を再編成すること、もしくは特定のデータ・パーティションですべてのパーティション索引を再編成することができます。

索引節の **CLEANUP** オプションを指定すると、索引を再作成しないでクリーンアップが実行されます。このコマンドを宣言済み一時表または作成済み一時表の索引に対して使用することはできません (SQLSTATE 42995)。

表オプションは、フラグメント化されたデータを消去するために行を再作成、および情報を縮小化することによって、表を再編成します。パーティション表では、単一パーティションを再編成できます。

### 有効範囲

このコマンドは、データベース・パーティション・グループ内のすべてのデータベース・パーティションに影響を与えます。

### 許可

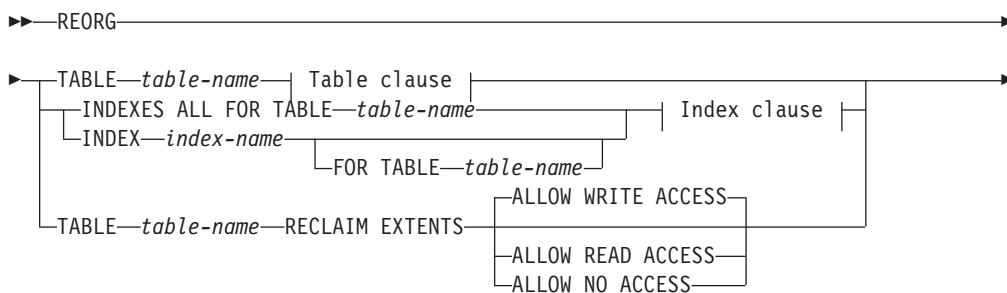
以下の権限のいずれか。

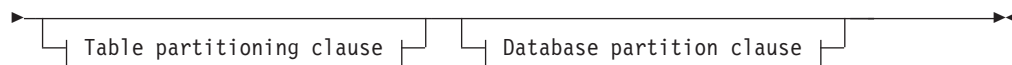
- SYSADM
- SYSCTRL
- SYSMOINT
- DBADM
- SQLADM
- 表に対する CONTROL 特権

### 必要な接続

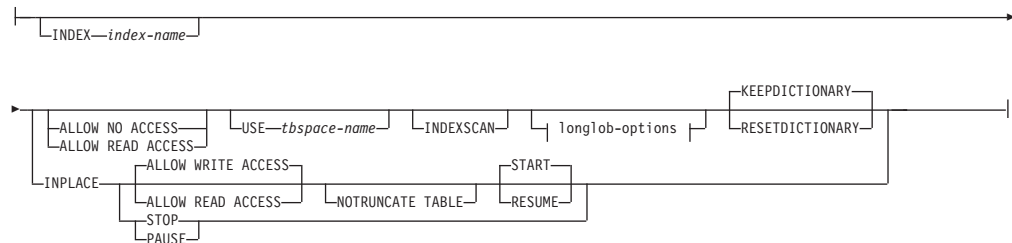
データベース

### コマンド構文

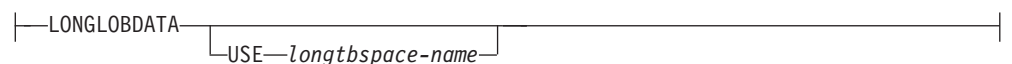




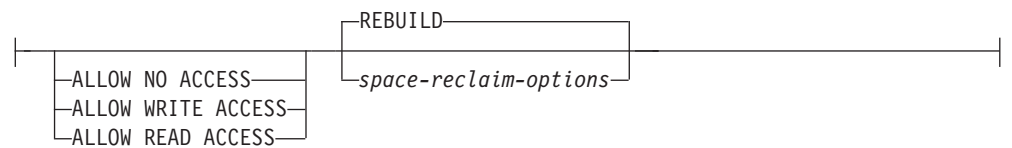
**Table clause:**



**longlob-options:**



**Index clause:**



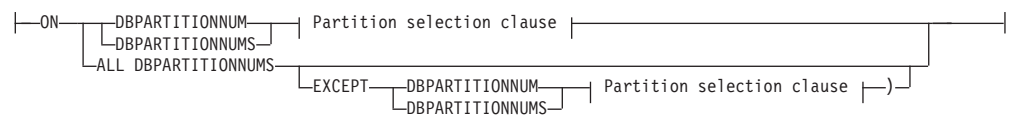
**space-reclaim-options:**



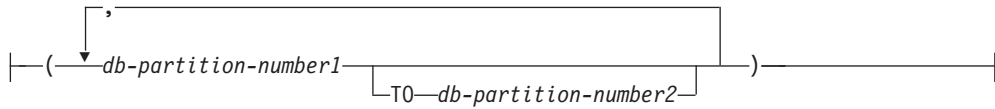
**Table partitioning clause:**



**Database partition clause:**



**Partition selection clause:**



**コマンド・パラメーター**

**INDEXES ALL FOR TABLE** *table-name*

索引を再編成する表を指定します。表は、ローカルまたはリモート・データベースにあるものです。

**INDEX** *index-name*

データ・パーティション表上の再編成する個別索引を指定します。個別索引の再編成は、パーティション表上の非パーティション索引についてのみサポートされています。このパラメーターは、ブロック索引に関してはサポートされていません。

**FOR TABLE** *table-name*

非パーティション索引 *index-name* が作成される表の名前を指定します。索引名がデータベースを通じて固有であるなら、このパラメーターはオプションです。

**ALLOW NO ACCESS**

**REORG INDEXES** の場合、索引が再編成される間に、他のユーザーがその表にアクセスできないことを指定します。パーティション表に対して **ON DATA PARTITION** 節が指定されている場合、その指定されたパーティションのみがこのアクセス・モード・レベルに制限されます。

**REORG INDEX** の場合、非パーティション索引が再編成される間に、他のユーザーがその表にアクセスできないことを指定します。

**ALLOW READ ACCESS**

**REORG INDEXES** の場合、索引が再編成される間に、他のユーザーがその表に対して、読み取り専用でアクセスできることを指定します。**CLEANUP** オプションまたは **RECLAIM EXTENTS** オプション、あるいは **ON DATA PARTITION** 節が指定されていない限り、パーティション表の **REORG INDEXES** に対して **ALLOW READ ACCESS** モードはサポートされません。パーティション表に対して **ON DATA PARTITION** 節が指定されている場合、その指定されたパーティションのみがこのアクセス・モード・レベルに制限されます。

**REORG INDEX** の場合、非パーティション索引が再編成される間に、その表に読み取り専用でアクセスできることを指定します。

**ALLOW WRITE ACCESS**

**REORG INDEXES** の場合、索引が再編成される間に、他のユーザーがその表から読み取ったりそこに書き込んだりできることを指定します。**CLEANUP** オプションまたは **RECLAIM EXTENTS** オプション、あるいは **ON DATA PARTITION** 節が指定されていない限り、パーティション表に対して **ALLOW WRITE ACCESS** モードはサポートされませ



ん。パーティション表に対して **ON DATA PARTITION** 節が指定されている場合、その指定されたパーティションのみがこのアクセス・モード・レベルに制限されます。

**REORG INDEX** の場合、非パーティション索引が再編成される間に、その表から読み取ったりそこに書き込んだりできることを指定します。

**CLEANUP** オプションまたは **RECLAIM EXTENTS** オプションが指定されている場合を除き、**ALLOW WRITE ACCESS** モードは、マルチディメンション・クラスタリング (MDC) 表、挿入時クラスタリング (ITC) 表、および拡張索引ではサポートされません。

**ON DATA PARTITION** 節が **REORG INDEXES ALL** コマンドと共に指定されている場合、以下の項目がデータ・パーティション表に適用されます。

- 指定されたデータ・パーティションのみがこのアクセス・モード・レベルに制限されます。指定されたパーティションのパーティション索引が再編成される間に、ユーザーは表の他のパーティションから読み取ったりそこに書き込んだりできます。

以下の表に、**ON DATA PARTITION** 節が指定された時にサポートされるアクセス・モードおよび表の他のパーティションで許可される並行アクセスを示します。

表 39. **REORG INDEXES ALL** と共に **ON DATA PARTITION** 節が指定された時にサポートされるアクセス・モードおよび許可される並行アクセス

アクセス・モード	指定されたパーティションで許可される並行アクセス	他のパーティションで許可される並行アクセス
<b>ALLOW NO ACCESS</b>	アクセスできません	読み取りおよび書き込みアクセス
<b>ALLOW READ ACCESS</b>	索引が更新されるまでのパーティションにおける読み取り	読み取りおよび書き込みアクセス
<b>ALLOW WRITE ACCESS</b>	索引が更新されるまでのパーティションにおける読み取りおよび書き込みアクセス	読み取りおよび書き込みアクセス

- 指定されたパーティションのパーティション索引のみが再編成されます。非パーティション索引は再編成されません。

「無効」または「再作成」とマークされた表に非パーティション索引がある場合、これらの索引は再編成前に再作成されます。それ以外の場合、索引オブジェクトが「無効」または「再作成」とマークされているなら、指定されたパーティションのパーティション索引のみが再編成されるか、または再作成されます。

- CLEANUP** オプションまたは **RECLAIM EXTENTS** オプションも指定されている場合、指定されたパーティションのパーティション索引のみがクリーンにされます。

以下の表は、パーティションおよび非パーティション表における索引の再編成に対してサポートされるアクセス・モードを示します。

## REORG INDEXES/TABLE

表 40. パーティションおよび非パーティション表における索引の再編成に対してサポートされるアクセス・モード

コマンド	表タイプ	表パーティション節	index (索引) 節に指定されたその他のパラメーター	サポートされるアクセス・モード
REORG INDEXES	非パーティション表	該当なし	任意	ALLOW NO ACCESS, ALLOW READ ACCESS <sup>1</sup> , ALLOW WRITE ACCESS
REORG INDEX	パーティション表	該当なし	任意	ALLOW NO ACCESS, ALLOW READ ACCESS <sup>1</sup> , ALLOW WRITE ACCESS
REORG INDEXES	パーティション表	なし	REBUILD (何も指定されていない場合のデフォルト)	ALLOW NO ACCESS <sup>1</sup>
REORG INDEXES	パーティション表	ON DATA PARTITION	REBUILD (何も指定されていない場合のデフォルト)	ALLOW NO ACCESS, ALLOW READ ACCESS <sup>1</sup> , ALLOW WRITE ACCESS
REORG INDEXES	パーティション表	ON DATA PARTITION 節ありまたはなし	CLEANUP または RECLAIM EXTENTS を指定	ALLOW NO ACCESS, ALLOW READ ACCESS <sup>1</sup> , ALLOW WRITE ACCESS

### 注:

1. アクセス節が指定されていない場合のデフォルト・モード。

### CLEANUP

**CLEANUP** が要求されると、**REBUILD** ではなくクリーンアップが実行されます。索引は再作成されません。解放されたページはこの表に定義された索引だけが再使用できます。

**ALL** コミット済み疑似削除済みキーおよびコミット済み疑似空白キーを除去することにより、索引をクリーンアップすることを指定します。

**CLEANUP ALL** オプションはコミット済み疑似空白ページを解放して、コミット済み疑似削除済みキーを疑似空白ではないページから除去します。このオプションは、隣接する複数のリーフ・ページをマージすると少なくとも **PCTFREE** のフリー・スペースを持つマージ済みリーフ・ページが生じる場合に、そのマージを試行します。**PCTFREE** は、索引作成時に索引に定義されたフリー・スペースのパーセントです。デフォルトの **PCTFREE** は 10% です。2 つのページがマージ可能な場合、そのうちの 1 つのページが解放されます。疑似空白ページにあるものを除く、索引内の疑似削除済みキーの数は、**RUNSTATS** を実行してから **NUMRIDS DELETED** を **SYSCAT.INDEXES** から選択することによって判別できます。**ALL** オプションは、コミット済みと判別された場合に **NUMRIDS DELETED** および **NUM EMPTY LEAFS** を消去します。

**PAGES** コミット済み疑似空白ページを索引ツリーから除去することを指定

します。これは、疑似空白ではないページ上の疑似削除済みキーはクリーンアップしません。これは疑似空白リーフ・ページだけをチェックするので、ほとんどの場合に **ALL** オプションを使用するよりも相当速くなります。

**CLEANUP PAGES** オプションは、コミット済み疑似空白ページを検索して解放します。コミット済み疑似空白ページとは、ページ上のすべてのキーに削除済みのマークが付いていて、それらすべての削除がコミット済みとして知られているページのことです。索引内の疑似空白ページの数、**RUNSTATS** を実行して **SYSCAT.INDEXES** の **NUM EMPTY LEAFS** 列を調べることにより判別できます。 **PAGES** オプションは、コミット済みと判別された場合に **NUM EMPTY LEAFS** を消去します。

**ALLOW READ ACCESS** または **ALLOW WRITE ACCESS** オプションを使用して、索引が再編成されている間に、他のトランザクションに表に対する読み取り専用または読み取り/書き込みのいずれかのアクセス権限を許可することができます。索引の再編成されたコピーを使用できる期間中に索引を再作成する場合、表へのアクセスは許可されません。

#### **INDEX** *index-name* **REBUILD**

**REBUILD** オプションがデフォルトです。このオプションは、**CLEANUP** 節および **CONVERT** 節が指定されていない場合の以前のリリースにおける索引再編成で提供されるのと同じ機能を表します。索引再編成の **REBUILD** オプションにより、物理的に連続したページに索引データが再作成されます。デフォルトのアクセス・モードは表タイプによって異なります。

#### **INDEX** *index-name* **RECLAIM EXTENTS**

再編成する索引を指定し、使用されていないエクステントを再利用します。このアクションによって、索引オブジェクト内の索引ページを移動して空エクステントが作成され、索引オブジェクトの排他的使用からこれらの空エクステントを解放され、表スペース内の他のデータベース・オブジェクトがスペースを使用できるようにします。エクステントは、索引オブジェクトから表スペースに戻されて再利用されます。 **ALLOW READ ACCESS** はデフォルトですが、すべてのアクセス・モードがサポートされています。

#### **TABLE** *table-name* **RECLAIM EXTENTS**

再編成する表を指定し、使用されていないエクステントを再利用します。*table-name* 変数では、マルチディメンション・クラスタリング (MDC) 表または挿入時クラスタリング (ITC) 表を指定する必要があります。*schema.table-name* 形式の名前あるいは別名を使用することができます。*schema* には、表作成時のユーザー名が入ります。スキーマ名を省略した場合、デフォルトのスキーマが想定されます。

**REORG TABLE RECLAIM EXTENTS** の場合、**ON DATA PARTITION** 節が指定されるたびに、アクセス節は指定されたパーティションに対してのみ適用されません。指定されたパーティションのエクステントが再利用される間に、ユーザーは表の残りの部分から読み取ったりそこに書き込んだりできます。この状態も、デフォルトのアクセス・レベルに適用されます。

## REORG INDEXES/TABLE

### ALLOW NO ACCESS

**REORG TABLE RECLAIM EXTENTS** の場合、エクステントが再利用される間に、他のユーザーがその表にアクセスできないことを指定します。

### ALLOW READ ACCESS

**REORG TABLE RECLAIM EXTENTS** の場合、エクステントが再利用される間に、他のユーザーがその表に読み取り専用アクセスができることを指定します。

### ALLOW WRITE ACCESS

**REORG TABLE RECLAIM EXTENTS** の場合、エクステントが再利用される間に、他のユーザーがその表で読み取り/書き込みができることを指定します。

### TABLE *table-name*

再編成する表を指定します。表は、ローカルまたはリモート・データベースにあるものです。 *schema.table-name* 形式の名前あるいは別名を使用することができます。 *schema* には、表作成時のユーザー名が入ります。スキーマ名を省略した場合、デフォルトのスキーマが想定されます。

型付き表の場合、指定する表名は階層のルート表の名前でなければなりません。

マルチディメンション・クラスタリング (MDC) 表または挿入時クラスタリング (ITC) 表の再編成に対して索引を指定することはできません。表のインプレース再編成を MDC 表または ITC 表に対して使用することはできません。

データ・パーティション表における表の再編成に対して **ON DATA PARTITION** 節が指定されている場合、指定されたデータ・パーティションのみが再編成されます。

- 表に非パーティション索引が定義されていない場合 (システム生成された XML パス索引を除く)、アクセス・モードは指定されたパーティションにのみ適用され、ユーザーは表の他のパーティションから読み取ったりそこに書き込んだりできます。
- 表に非パーティション索引が定義されている場合 (システム生成された XML パス索引を除く)、**ALLOW NO ACCESS** モードがデフォルトとなり、サポートされる唯一のアクセス・モードになります。この場合、その表は **ALLOW NO ACCESS** モードになります。**ALLOW READ ACCESS** が指定されている場合、SQL1548N が戻されます (SQLSTATE 5U047)。

表 41. 非パーティションおよびパーティション表における表の再編成に対してサポートされるアクセス・モード

コマンド	表タイプ	表パーティション節	サポートされるアクセス・モード
<b>REORG TABLE</b>	非パーティション表	該当なし	<b>ALLOW NO ACCESS</b> , <b>ALLOW READ ACCESS</b> <sup>1</sup>
<b>REORG TABLE</b>	パーティション表	指定なし	<b>ALLOW NO ACCESS</b> <sup>1</sup>
<b>REORG TABLE</b> (索引がない、または表にパーティション索引のみが定義されている)	パーティション表	<b>ON DATA PARTITION</b>	<b>ALLOW NO ACCESS</b> , <b>ALLOW READ ACCESS</b> <sup>1</sup>

表 41. 非パーティションおよびパーティション表における表の再編成に対してサポートされるアクセス・モード (続き)

コマンド	表タイプ	表パーティション節	サポートされるアクセス・モード
REORG TABLE (システム生成された XML パス索引を除き、表に非パーティション索引が定義されていない)	パーティション表	ON DATA PARTITION	ALLOW NO ACCESS <sup>1</sup>

## 注:

1. アクセス節が指定されていない場合のデフォルト・モード。

データ・パーティション表では、表を再編成すると、表の再編成後にその表の非パーティション索引およびパーティション索引が再作成されます。 **ON DATA PARTITION** 節を使用して、データ・パーティション表の特定のデータ・パーティションを再編成する場合、表の再編成によって、指定したパーティションのみで非パーティション索引およびパーティション索引が再作成されます。

**INDEX** *index-name*

表を再編成する際に使用する索引を指定します。 *schema.index-name* 形式の完全修飾名を指定しない場合、デフォルトのスキーマが想定されます。 *schema* は、その索引が作成された時のユーザー名です。データベース・マネージャーは、再編成している表のレコードを物理的に再配列する索引を使用します。

表のインプレース再編成では、クラスタリング索引が表に定義されて、索引が指定されている場合、それはクラスタリング索引でなければなりません。インプレース・オプションが指定されない場合、指定された任意の索引が使用されます。索引名を指定しない場合には、そのレコードは順番に関係なく再編成されます。しかし、表にクラスタリング索引が定義されている場合、索引が指定されていない場合は、クラスタリング索引が使用されて表がクラスタリングされます。MDC 表または ITC 表を再編成しているときには、索引を指定できません。

**INDEX** および **ON DATA PARTITION** 節の両方を使用して表が再編成される場合、指定したパーティションのみが、索引 *index-name* を使用して再編成されます。

**ALLOW NO ACCESS**

表が再編成される間に、他のユーザーがその表にアクセスできないことを指定します。

**ON DATA PARTITION** 節を使用せずにパーティション表を再編成する場合、**ALLOW NO ACCESS** モードがデフォルトになり、唯一のサポートされるアクセス・モードになります。

データ・パーティション表に対して **ON DATA PARTITION** 節が指定されている場合、指定されたデータ・パーティションのみが再編成されます。

- 表に非パーティション索引が定義されていない場合 (システム生成された XML パス索引を除く)、指定されたパーティションのみが **ALLOW NO ACCESS** モードに制限されます。ユーザーは表の他のパーティションから読み取ったりそこに書き込んだりできません。
- 表に非パーティション索引が定義されている場合 (システム生成された XML パス索引を除く)、**ALLOW NO ACCESS** モードがデフォルトとなり、サポートされる唯一のアクセス・モードになります。この場合、その表は **ALLOW NO ACCESS** モードになります。

#### ALLOW READ ACCESS

再編成の際に表に対する読み取りアクセスだけを許可します。

**ALLOW READ ACCESS** モードが、非パーティション表のデフォルト・モードになります。

データ・パーティション表に対して **ON DATA PARTITION** 節が指定されている場合、指定されたデータ・パーティションのみが再編成されます。

- 表に非パーティション索引が定義されていない場合 (システム生成された XML パス索引を除く)、**ALLOW READ ACCESS** モードがデフォルト・モードとなり、指定されたパーティションのみがそのアクセス・モード・レベルに制限されます。ユーザーは表の他のパーティションから読み取ったりそこに書き込んだりできません。
- 表に非パーティション索引が定義されている場合 (システム生成された XML パス索引を除く)、**ALLOW READ ACCESS** モードはサポートされません。この場合に **ALLOW READ ACCESS** が指定されると、SQL1548N が戻されます (SQLSTATE 5U047)。

#### INPLACE

ユーザー・アクセスを許可しながら、表を再編成します。

インプレース表再編成が可能なのは、拡張索引を持たず、表内の XML 列に対して索引が定義されていない非パーティション表、非 MDC 表、および非 ITC 表のみです。表のインプレース再編成は、サイズが少なくとも 3 ページである表に対してのみ実行できます。

表のインプレース再編成は非同期に発生するので、即時に有効にならないことがあります。

#### ALLOW READ ACCESS

再編成の際に表に対する読み取りアクセスだけを許可します。

#### ALLOW WRITE ACCESS

再編成の際に表に対する書き込みアクセスを許可します。これがデフォルトの動作です。

#### NOTRUNCATE TABLE

インプレース再編成の後に表を切り捨てないでください。切り捨ての際に、表は S ロックされます。



- START** インプレース **REORG** 処理を開始します。これがデフォルトなので、このキーワードはオプションです。
- STOP** インプレース **REORG** 処理を現時点で停止します。
- PAUSE** インプレース **REORG** を当面の間、中断または一時停止します。
- RESUME** 以前に一時停止した、表のインプレース再編成を継続または再開します。オンライン再編成が再開された時点で、再編成の一時停止時と同じオプションを指定したい場合は、再開時にそれらのオプションを再び指定する必要があります。

**USE** *tblspace-name*

再編成されている表の一時コピーを保管する **SYSTEM**

**TEMPORARY** 表スペースの名前を指定します。表スペースの名前を指定しない場合、データベース・マネージャーは、再編成しようとする表を含む表スペースにその表の作業用コピーを保管します。

8 KB、16 KB、または 32 KB の表オブジェクトの場合、指定した **SYSTEM TEMPORARY** 表スペースのページ・サイズが、表データの存在する表スペースのページ・サイズと一致していなければ、**DB2** データベース製品は、**LONG/LOB** オブジェクトのサイズが正しい **TEMPORARY** 表スペースを検出しようとします。再編成が正常に実行されるためには、そのような表スペースが存在していなければなりません。

パーティション表の場合、表に含まれるデータ・パーティションの再編成において、**TEMPORARY** 表スペースが一時ストレージとして使用されます。パーティション表全体の再編成では、一度に 1 つのデータ・パーティションが再編成されます。**TEMPORARY** 表スペースは、表全体ではなく、表に含まれる最大のデータ・パーティションを保持できる必要があります。**ON DATA PARTITION** 節が指定される場合、**TEMPORARY** 表スペースは指定されたパーティションを保持できる必要があります。

パーティション表の表スペース名を指定しない場合、各データ・パーティションの存在する表スペースが、そのデータ・パーティションの一時ストレージとして使用されます。各データ・パーティションの表スペースには、そのデータ・パーティションのコピーが入るだけの十分なフリー・スペースがなければなりません。

**INDEXSCAN**

クラスタリング **REORG** では、索引スキャンが使用されて表レコードが再配列されます。索引を介して表にアクセスすることにより、表の行を再編成します。デフォルトの方法は、必要に応じて **TEMPORARY** 表スペースを使用しながら、表をスキャンして結果をソートし表を再編成することです。索引キーはソートの順序に配列していますが、スキャンおよびソートはまず索引から行 ID を読み取って行をフェッチするよりも通常は高速です。

**LONGLOBDATA**

長いフィールドおよび **LOB** データが再編成されます。

表に長い列または LOB 列が含まれる場合でも、これは必要ではありません。これは時間がかかり、クラスタリングを改善しないために、デフォルトではこれらのオブジェクトを再編成しません。しかし、XML 列を持つ表に対して **LONGLOBDATA** オプションを指定して再編成を実行すると、未使用のスペースが再利用されるため、XML ストレージ・オブジェクトのサイズが削減されます。

このパラメーターは、既存の LOB データをインライン化された LOB データに変換するときが必要です。

#### **USE** *longtbspace-name*

これはオプション・パラメーターであり、LONG データを再作成するために使用する TEMPORARY 表スペースの名前を指定するために使用できます。表オブジェクトについても LONG オブジェクトについても TEMPORARY 表スペースが指定されていない場合、現在それらのオブジェクトが存在している表スペース中にそれらのオブジェクトが構成されることとなります。表の TEMPORARY 表スペースが指定されているが、このパラメーターが指定されていない場合には、ページ・サイズが異なるのでない限り、基本再編成データのために使用される表スペースが使用されることとなります。ページ・サイズが異なる場合、DB2 データベース・システムは、LONG オブジェクトを作成するために適切なページ・サイズの一時コンテナを選択することを試みます。

**USE** *longtbspace-name* が指定されている場合、**USE** *tbspace-name* も指定する必要があります。そうでない場合、*longtbspace-name* 引数は無視されます。

#### **KEEPDICTIONARY**

表の COMPRESS 属性が YES であり、表にコンプレッション・ディクショナリーがある場合、新しいディクショナリーは作成されません。再編成中に処理されるすべての行は、既存のディクショナリーを使用して圧縮されます。COMPRESS 属性が YES であり、表にコンプレッション・ディクショナリーが存在しない場合、ディクショナリーは、表が特定のサイズ (およそ 1 から 2 MB) で、この表に十分なデータがある場合のみ、このシナリオで作成されます (さらに表が圧縮されます)。代わりに明示的に **REORG RESETDICTIONARY** を指定した場合には、表に少なくとも 1 行あれば、ディクショナリーが作成されます。表の COMPRESS 属性が NO であり、表にコンプレッション・ディクショナリーがある場合、再編成処理によりそのディクショナリーは保存され、新たに編成された表のすべての行は非圧縮形式になります。基本表の行に保管されない LOB などの一部のデータは圧縮できません。

**LONGLOBDATA** オプションを指定しない場合、表の行データだけが再編成されます。以下の表は、**LONGLOBDATA** オプションが指定されていないときの、**REORG** コマンドの **KEEPDICTIONARY** 構文の動作を説明しています。



表 42. REORG KEEPDICTIONARY

圧縮	ディクショナリーが存在するかどうか	結果と効果
Y	Y	ディクショナリーはそのまま、行圧縮
Y	N	ディクショナリーを作成、行圧縮
N	Y	ディクショナリーはそのまま、全行圧縮解除
N	N	影響なし、全行圧縮解除

以下の表は、**LONGLOBDATA** オプションが指定されているときの、**REORG** コマンドの **KEEPDICTIONARY** 構文の動作を説明しています。

表 43. LONGLOBDATA オプションが指定されているときの REORG KEEPDICTIONARY.

圧縮	表の行データ・ディクショナリーが存在する	XML ストレージ・オブジェクト・ディクショナリーが存在する <sup>1</sup>	コンプレッション・ディクショナリー	データ圧縮
Y	Y	Y	ディクショナリーを保存します。	既存のデータは、圧縮されます。新規データは、圧縮されます。
Y	Y	N	表の行ディクショナリーを保存し、XML ストレージ・オブジェクト・ディクショナリーを作成します。	既存のデータは、圧縮されます。新規データは、圧縮されます。
Y	N	Y	表の行ディクショナリーを作成し、XML ディクショナリーを保存します。	既存のデータは、圧縮されます。新規データは、圧縮されます。
Y	N	N	表の行ディクショナリーおよび XML ディクショナリーを作成します。	既存のデータは、圧縮されます。新規データは、圧縮されます。
N	Y	Y	表の行ディクショナリーおよび XML ディクショナリーを保存します。	表データは圧縮解除されます。新規データは、圧縮されません。
N	Y	N	表の行ディクショナリーを保存します。	表データは圧縮解除されます。新規データは、圧縮されません。
N	N	Y	XML ディクショナリーを保存します。	表データは圧縮解除されます。新規データは、圧縮されません。
N	N	N	影響なし。	表データは圧縮解除されます。新規データは、圧縮されません。

注:

1. コンプレッション・ディクショナリーは、XML 列が DB2 V9.7 以降の表に追加された場合または表が ONLINE\_TABLE\_MOVE ストアード・プロシージャを使用してマイグレーションされた場合にのみ、表の XML ストレージ・オブジェクトに対して作成できます。

表の圧縮属性が NO の場合、(置換操作などで) 表の再初期設定または切り捨てが発生したなら、ディクショナリーが存在していればそれは廃棄されます。逆に、表の圧縮属性が YES の場合にディクショナリーが存在しているなら、切り捨てではディクショナリーが保たれ、廃棄は実行されません。リカバリーのことを考慮して、また将来データ・キャプチャーの変更 (つまりレプリケーション) をサポートすることを考慮して、ディクショナリーは全体としてログ記録されます。

#### RESETDICTIONARY

表の COMPRESS 属性が YES の場合、新しい行コンプレッション・ディクショナリーが作成されます。再編成の際に処理されるすべての行は、この新しいディクショナリーによる圧縮の対象になります。前のすべてのディクショナリーは、このディクショナリーに置き換わります。表の COMPRESS 属性が NO であり、表にコンプレッション・ディクショナリーが存在している場合、再編成処理によりそのディクショナリーは除去され、新たに編成された表のすべての行は非圧縮形式になります。基本表の行に保管されない LOB などの一部のデータは圧縮できません。

**LONGLOBDATA** オプションを指定しない場合、表の行データだけが再編成されます。以下の表は、**LONGLOBDATA** オプションが指定されていないときの、**REORG** コマンドの **RESETDICTIONARY** 構文の動作を説明しています。

表 44. REORG RESETDICTIONARY

圧縮	ディクショナリーが存在するかどうか	結果と効果
Y	Y	ディクショナリー新規作成*、行圧縮。DATA CAPTURE CHANGES オプションが CREATE TABLE または ALTER TABLE ステートメントで指定されていると、現在のディクショナリーが保持されます (これは履歴コンプレッション・ディクショナリーと呼ばれます)。
Y	N	ディクショナリー新規作成、行圧縮
N	Y	ディクショナリー除去、全行圧縮解除。DATA CAPTURE NONE オプションが CREATE TABLE または ALTER TABLE ステートメントで指定されていると、指定された表の履歴コンプレッション・ディクショナリーも除去されます。
N	N	影響なし、全行圧縮解除

\* - ディクショナリーが存在し、圧縮属性が有効になっていても、現在は表にデータがない場合は、**RESETDICTIONARY** 操作でも既存のディクショナリーが維持されます。内部の最小レコード長よりサイズの小さい行と、圧縮が試みられたときにレコード長の節減が図られない行は、このような場合は「十分でない」とみなされます。

以下の表は、**LONGLOBDATA** オプションが指定されているときの、**REORG** コマンドの **RESETDICTIONARY** 構文の動作を説明しています。

表 45. **LONGLOBDATA** オプションが指定されているときの **REORG RESETDICTIONARY**.

圧縮	表の行データ・ディクショナリーが存在する	XML ストレージ・オブジェクト・ディクショナリーが存在する <sup>1</sup>	データ・ディクショナリー	データ圧縮
Y	Y	Y	ディクショナリーの作成 <sup>2 3</sup> 。	既存のデータは、圧縮されます。新規データは、圧縮されます。
Y	Y	N	新規の表の行ディクショナリーを作成し、新規の XML ディクショナリーを作成します <sup>3</sup> 。	既存のデータは、圧縮されます。新規データは、圧縮されます。
Y	N	Y	表の行データ・ディクショナリーを作成し、新規 XML ディクショナリーを作成します。	既存のデータは、圧縮されます。新規データは、圧縮されます。
Y	N	N	ディクショナリーを作成します。	既存のデータは、圧縮されます。新規データは、圧縮されます。
N	Y	Y	ディクショナリーを除去します。既存および新規のデータは圧縮されません。	既存の表データは圧縮解除されます。新規データは、圧縮されません。
N	Y	N	表の行ディクショナリーを除去します。すべてのデータは圧縮解除されます。	既存の表データは圧縮解除されます。新規データは、圧縮されません。
N	N	Y	XML ストレージ・オブジェクト・ディクショナリーを除去します。	既存の表データは圧縮解除されます。新規データは、圧縮されません。
N	N	N	影響なし。	既存の表データは圧縮解除されます。新規データは、圧縮されません。

注:

1. コンプレッション・ディクショナリーは、XML 列が DB2 V9.7 以降の表に追加された場合または表が Online Table Move を使

- 用してマイグレーションされた場合にのみ、表の XML ストレージ・オブジェクトに対して作成できます。
2. ディクショナリーが存在していて圧縮属性が有効であるが、現在のところ表中にデータが存在しない場合、**RESETDICTIONARY** 操作では既存のディクショナリーがそのまま保たれます。この場合、不適切とみなされるのは、内部最小レコード長よりもサイズが小さい行、および圧縮してもレコード長の節約にならない行です。
  3. **DATA CAPTURE CHANGES** オプションが **CREATE TABLE** または **ALTER TABLE** ステートメントで指定されていると、現在のデータ・ディクショナリーが保持されます (これは履歴コンプレッション・ディクショナリーと呼ばれます)。

### ON DATA PARTITION *partition-name*

データ・パーティション表では、再編成するデータ・パーティションを指定します。

DB2 V9.7 フィックスパック 1 以降のリリースでは、この節を **REORG INDEXES ALL** コマンドで使用して、特定のパーティションのパーティション索引を再編成することができます。また、**REORG TABLE** コマンドで使用して、特定のパーティションのデータを再編成することもできます。

パーティション表で **REORG TABLE** または **REORG INDEXES ALL** コマンドでこの節を使用するときに、パーティション *partition-name* が指定した表にない場合、再編成は失敗し、理由コード 1 と共に **SQL2222N** を戻します。パーティション *partition-name* がアタッチまたはデタッチ状態の場合、再編成は失敗し、理由コード 3 と共に **SQL2222N** を戻します。

**ON DATA PARTITION** 節を指定して **REORG INDEX** コマンドが発行された場合、再編成は失敗し、理由コード 2 と共に **SQL2222N** を戻します。

パーティション表が **REORG** ペンディング状態にあり、かつ表に非パーティション索引が定義されている場合、**REORG TABLE** コマンドは失敗し、**SQL1549N (SQLSTATE 5U047)** を戻します。

### ALL DBPARTITIONNUMS

**db2nodes.cfg** ファイルに指定されているすべてのデータベース・パーティションで操作が実行されることを指定します。データベース・パーティション節が指定されていない場合、これがデフォルトです。

**EXCEPT** データベース・パーティション・リストに指定されているものを除き、**db2nodes.cfg** ファイルに指定されているすべてのデータベース・パーティションで操作が実行されることを指定します。

### ON DBPARTITIONNUM | ON DBPARTITIONNUMS

データベース・パーティションのセットに対して操作を実行します。

#### *db-partition-number1*

データベース・パーティション・リスト内のデータベース・パーティション番号を指定します。

#### *db-partition-number2*

2 番目のデータベース・パーティション番号を指定し、

*db-partition-number1* から *db-partition-number2* までのすべてのデータベース・パーティションがデータベース・パーティション・リストに含まれるようにします。

## 例

表を再編成して、スペースを再利用し、TEMPORARY 表スペース `mytemp1` を使用するには、以下のコマンドを入力します。

```
db2 reorg table homer.employee use mytemp1
```

4 パーティション・システムのデータベース・パーティション 1、2、3、および 4 から構成されるパーティション・グループの表を再編成するには、以下のコマンドのいずれかを入力できます。

```
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)
```

他のトランザクションに表の読み取りおよび更新を許可しながら、EMPLOYEE 表にあるすべての索引内の疑似削除済みキーおよび疑似空白ページをクリーンアップするには、次のように入力します。

```
db2 reorg indexes all for table homer.employee allow write
access cleanup only
```

他のトランザクションに表の読み取りおよび更新を許可しながら、EMPLOYEE 表にあるすべての索引内の疑似空白ページをクリーンアップするには、次のように入力します。

```
db2 reorg indexes all for table homer.employee allow write
access cleanup only pages
```

作業域として SYSTEM TEMPORARY 表スペース `TEMPSPACE1` を使用して EMPLOYEE 表を再編成するには、次のように入力してください。

```
db2 reorg table homer.employee use tempSPACE1
```

(これまでの例で明示的に指定した) デフォルト・スキーマ `HOMER` を使って EMPLOYEE 表のインプレース REORG を開始、一時停止、および再開するには、以下のコマンドを入力します。

```
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
```

REORG を再開するコマンドには、読み取りアクセスのみを指定して切り捨てステップをスキップし、表を共有ロックする追加のキーワードが含まれています。

## 使用上の注意

制約事項:

- **REORG** ユーティリティでは、ニックネームの使用はサポートされません。
- **REORG TABLE** コマンドは、宣言済み一時表または作成済み一時表に対してサポートされていません。
- **REORG TABLE** コマンドは、ビューに対しては使用できません。

- 表の再編成は、範囲がクラスター化された表との互換性がありません。表の範囲領域は常に、クラスター化されているからです。
- DMS 表スペース中のパーティション表が属している表スペース (LOB や索引を含む) のオンライン・バックアップが実行されている間は、**REORG TABLE** をその表に対して使用することはできません。
- **REORG TABLE** は、索引拡張子に基づく索引を使用できません。
- 表が **REORG** ペンディング状態なら、その表に対してインプレース再編成を実行することはできません。
- 同じ **TEMPORARY DMS** 表スペースの共有による表の再編成の同時実行はサポートされていません。
- イベント・モニターの書き込み先である表に対して再編成を実行する場合は、その前にその表に関係するイベント・モニターを非活動化しておく必要があります。
- データ・パーティション表では、以下のようになります。
  - **SYSCAT.TABLES** の中でその表の **ACCESS\_MODE** はフル・アクセス権限でなければなりません。
  - 再編成では、アタッチまたはデタッチ操作のため、制限状態にあるデータ・パーティションがスキップされます。**ON DATA PARTITION** 節が指定されている場合、そのパーティションは完全にアクセス可能でなければなりません。
  - 表の再編成中にエラーが発生した場合、一部の索引または索引パーティションは無効のままとなる可能性があります。表の非パーティション索引は、再編成が最初のデータ・パーティションの置換フェーズに達するかそれをパスすると、無効のマークが付きます。置換フェーズに既に達したかまたはそれを通過したデータ・パーティションの索引パーティションには、無効のマークが付きます。索引は次回表またはデータ・パーティションにアクセスしたときに再作成されます。
  - **ALLOW NO ACCESS** モードが使用されている場合、索引の再編成中にエラーが発生すると、表の一部の索引が無効のままになる場合があります。表の非パーティション **RID** 索引については、障害発生時点で再編成中の索引のみが無効のままになります。非パーティション・ブロック索引がある **MDC** 表については、エラーが発生した場合、1 つ以上のブロック索引が無効のままになる可能性があります。パーティション索引を持つ **MDC** 表または **ITC** 表の場合、再編成中のデータ・パーティションの索引オブジェクトだけが無効のままになります。無効のマークが付いたすべての索引は、次回表またはデータ・パーティションにアクセスしたときに再作成されます。
  - 表に定義されたパーティション索引のみを使用するデータ・パーティション表が **REORG** ペンディング状態にある場合、**ON DATA PARTITION** 節を指定して **REORG TABLE** コマンドを発行すると、指定されたデータ・パーティションのみ **REORG** ペンディング状態ではなくなります。その表の残りのパーティションを **REORG** ペンディング状態から戻すためには、表全体に対して **REORG TABLE** コマンド (**ON DATA PARTITION** 節を使用しないで) を発行するか、または残っているパーティションそれぞれに対して **ON DATA PARTITION** 節を指定して **REORG TABLE** コマンドを発行します。



表の再編成の現在の進行状態に関する情報は、データベース活動の履歴ファイルに書き込まれます。履歴ファイルには、再編成イベントごとの記録が含まれています。このファイルを表示するには、再編成している表を含むデータベースに対して **LIST HISTORY** コマンドを実行します。

さらに、表スナップショットを使用して表の再編成の進行状況をモニターすることもできます。表の再編成のモニター・データは、「データベース・モニター表スイッチ (Database Monitor Table Switch)」の設定値に関係なく記録されます。

エラーが生じた場合、SQLCA ダンプが履歴ファイルに書き込まれます。表のインプレース再編成の場合、状況が PAUSED として記録されます。

索引付き表が何回も変更されると、索引内のデータがフラグメント化されることがあります。表が索引に関してクラスター化されている場合、表および索引をクラスターの順序から取り出すことができます。これら両方の要素は索引を使用するスキヤンのパフォーマンスを低下させ、索引ページの事前取り出しの効果に影響を与ることがあります。 **REORG INDEX** または **REORG INDEXES** で REBUILD オプションを指定して、表の 1 つの索引またはすべての索引を再編成できます。索引を再編成して再作成すると、フラグメントが除去され、リーフ・ページに物理クラスタリングがリストアされます。 **REORGCHK** を使用すると、索引に再編成が必要かどうかを判別するために役立ちます。すべてのデータベース操作が完了し、すべてのロックを解放したことを確かめてから、索引の再編成を呼び出してください。これは、WITH HOLD でオープンされた、すべてのカーソルをクローズした後で COMMIT または ROLLBACK を発行することによって行われます。

従来の表の再編成 (オフライン再編成) では、再編成の最後のフェーズで索引が再作成されます。TEMPORARY 表スペースが複数存在する場合は、表の再編成処理に伴って追加のソートを行う際に、**REORG TABLE** コマンドで指定した TEMPORARY 表スペースに加えて、それ以外の TEMPORARY 表スペースが 1 つ使用される可能性もあります。ただし、表のインプレース再編成 (オンライン再編成) では索引は再作成されません。インプレース表再編成の完了後に、**REORG INDEXES** コマンドを発行することをお勧めします。インプレース表再編成は非同期であるため、**REORG INDEXES** コマンドを発行する前にインプレース表再編成が完了していることを確認するように注意しなければなりません。インプレース表再編成が完了する前に **REORG INDEXES** コマンドを発行すると、再編成が失敗するおそれがあります (SQLCODE -2219)。

何回も修正されてデータがフラグメント化して、アクセス・パフォーマンスが大幅に低下した表も **REORG TABLE** コマンドの対象になります。構造化タイプ列のインラインの長さを変更後、このユーティリティーもまた呼び出して、変更の有用性を確認してください。 **REORGCHK** コマンドを使用して、表の再編成が必要であるかどうか判別してください。すべてのデータベース操作が完了し、すべてのロックが解放されていることを確かめてから、**REORG TABLE** を呼び出してください。これは、WITH HOLD でオープンされた、すべてのカーソルをクローズした後で COMMIT または ROLLBACK を発行することによって行われます。表の再編成の後で、**RUNSTATS** を使用して表統計を更新し、**REBIND** を使用してこの表を使用するパッケージを再バインドします。再編成ユーティリティーは、暗黙的にすべてのカーソルをクローズします。

DB2 V9.7 フィックスパック 1 以降では、1 つのデータ・パーティション表に対して **REORG TABLE** コマンドおよび **REORG INDEXES ALL** コマンドを発行し、同時に複数のデータ・パーティション、または 1 つのパーティション上の複数のパーティション索引を再編成することができます。複数のデータ・パーティション、または 1 つのパーティション上の複数のパーティション索引を並行して再編成する場合、ユーザーは影響を受けないパーティションにはアクセスできますが、影響を受けるパーティションにはアクセスできません。同一表に対して同時に操作する複数の **REORG** コマンドを発行するには、以下の基準をすべて満たす必要があります。

- 各 **REORG** コマンドは、**ON DATA PARTITION** 節を指定して、それぞれ別個のパーティションを指定しなければならない。
- 各 **REORG** コマンドは、**ALLOW NO ACCESS** モードを使用して、データ・パーティションに対するアクセスを制限しなければならない。
- **REORG TABLE** コマンドを発行する場合、パーティション表にはパーティション索引のみが含まれていなければならない。非パーティション索引が表に定義されているはなりません (システム生成された XML パス索引を除く)。

非パーティション索引がなく (システムが生成した XML パス索引を除く)、パーティション P1、P2、P3、P4 を使用するパーティション表 T1 の場合、以下の複数の **REORG** コマンドを並行して実行することができます。

```
REORG INDEXES ALL FOR TABLE T1 ALLOW NO ACCESS ON DATA PARTITION P1
REORG TABLE T1 ALLOW NO ACCESS ON DATA PARTITION P2
REORG INDEXES ALL FOR TABLE T1 ALLOW NO ACCESS ON DATA PARTITION P3
```

並行 **REORG** コマンドを使用する場合、以下のような操作はサポートされません。

- **ON DATA PARTITION** 節を指定せずに、表に対して **REORG** コマンドを使用する。
- データ・パーティションを追加、アタッチ、またはデタッチするのに、表で **ALTER TABLE** ステートメントを使用する。
- データを表にロードする。
- 表を含むオンライン・バックアップを実行する。

表の値圧縮をアクティブ化または非アクティブ化したために表に混合した行形式が含まれている場合、オフラインで表を再編成することによって、既存の行すべてをターゲットの行形式に変換することができます。

表がいくつかのデータベース・パーティションに分散している場合、影響を受けるデータベース・パーティションのいずれかで表または索引の再編成が失敗すると、失敗したデータベース・パーティションでのみ表または索引の再編成がロールバックされます。

再編成が成功しなかった場合には、一時ファイルを削除すべきではありません。データベース・マネージャーは、これらのファイルを使用し、データベースをリカバリします。

索引の名前が指定されると、データベース・マネージャーはその索引の順番に従って、データを再編成します。パフォーマンスを最大にするため、SQL 照会で頻繁に使用される索引を指定してください。索引の名前が指定されておらず、クラスタリング索引が存在する場合、データはクラスタリング索引に従って配列されます。



表の PCTFREE 値は、ページごとに指定されたフリー・スペースの量を決定します。この値が設定されていない場合、ユーティリティーは、それぞれのページにできるだけ大きなスペースを割り当てます。

表の再編成の後に表スペースのロールフォワード・リカバリーを完了させるには、通常の表スペースと LARGE 表スペースの両方で、ロールフォワード・リカバリーが有効になっていなければなりません。

その表が、**COMPACT** オプションを使用しない LOB 列を含む場合、LOB DATA ストレージ・オブジェクトは、表の再編成に従いかなり大きくなることができます。これは、行が再編成された順序、および使用される (SMS または DMS) 表スペースのタイプの結果になります。

**REORG INDEXES/TABLE** コマンドにより、XML データに対する索引が再作成される場合があります。詳細については、『XML データに対する索引の再作成』を参照してください。

## REORGCHK

データベースでの統計を計算し、表または索引もしくはその両方が、再編成またはクリーンアップされる必要があるか判別します。

### 有効範囲

このコマンドは、db2nodes.cfg ファイル中のどのデータベース・パーティションからでも発行できます。これを使用して、カタログ中の表および索引統計を更新できます。

### 許可

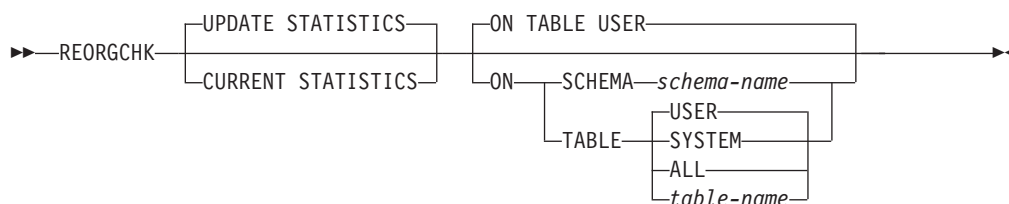
以下の権限のいずれか。

- SYSADM または DBADM 権限
- 表に対する CONTROL 特権

### 必要な接続

データベース

### コマンド構文



### コマンド・パラメーター

#### UPDATE STATISTICS

**RUNSTATS** ルーチン呼び出して、表および索引の統計を更新してから、更新後の統計を使用して、表または索引の再編成が必要かどうかを判別します。

**REORGCHK** が発行されたデータベース・パーティション上に表の一部が置かれている場合、ユーティリティはそのデータベース・パーティションに対して実行されます。このデータベース・パーティションに表が存在しない場合、その要求は、表の一部を保持しているデータベース・パーティション・グループ中の最初のデータベース・パーティションに送信されます。その後、このデータベース・パーティションに対して **RUNSTATS** が実行されます。

#### CURRENT STATISTICS

現在の表の統計を使用して、表の再編成が必要かどうかを判別します。

#### ON SCHEMA *schema-name*

指定のスキーマの下で作成されたすべての表を検査します。

#### ON TABLE

**USER** ランタイム許可 ID が所有する表を検査します。

**SYSTEM** システムの表を検査します。

**ALL** すべてのユーザーおよびシステムの表を検査します。

*table-name*

検査する表を指定します。 *schema.table-name* 形式の名前あるいは別名を使用することができます。 *schema* には、表作成時のユーザー名が入ります。スキーマ名を省略した場合、現行スキーマであると想定されます。指定された表がシステム・カタログの表である場合には、 *schema* は **SYSIBM** です。型付き表の場合、指定する表名は階層のルート表の名前でなければなりません。

## 例

**SAMPLE** データベースに対して以下のコマンドを発行します。

```
db2 reorgchk update statistics on table system
```

結果出力の中で、表統計の用語 (公式 1 から 3 まで) の意味は以下のとおりです。

**CARD** (CARDINALITY) 基本表の行数。

**OV** (OVERFLOW) オーバーフローした行数。

**NP** (NPAGES) データを含むページ数。

**FP** (FPAGES) ページの合計数。

### ACTBLK

マルチディメンション・クラスタリング (MDC) 表または挿入時クラスタリング (ITC) 表のアクティブ・ブロックの合計数。このフィールドは、**ORGANIZE BY** 節を使用して定義された表に対してのみ適用できます。これは、データを含む表のブロック数を示します。

**TSIZE** 表サイズ (バイト数)。表 (**CARD**) 内の行数と行の長さの平均を基にして計算されます。行の長さの平均は、列の長さの平均 (**SYSCOLUMNS** の **AVGCOLLEN**) の合計に、行のオーバーヘッドとして 10 バイトを加えたものを基にして計算されます。長フィールドと **LOB** の場合には、記述子のおおよその長さだけが使用されます。実際の長フィールドまたは **LOB** データは、**TSIZE** にはカウントされません。

### TABLEPAGESIZE

表データが存在する表スペースのページ・サイズ。

### NPARTITIONS

これがパーティション表である場合は、パーティション数。そうでない場合は 1。

**F1** 公式 1 の結果。

**F2** 公式 2 の結果。

**F3** 公式 3 の結果。この公式は、表の中で無駄になっているスペース量を示します。この量の測定基準は、空のページ数と、表のページ内に存在するデータを含んだページ数です。マルチディメンション・クラスタリング (MDC) 表または挿入時クラスタリング (ITC) 表の場合は、空のブロック数と、データを含んだブロック数が測定されます。

**REORG**

この列に表示されている各ハイフン (-) は、計算結果が、対応する公式の設定範囲内であったことを示しています。各アスタリスク (\*) は、計算結果が、その対応する公式の設定範囲を超えたことを示しています。

- 列の左側の - または \* は F1 (公式 1) に対応しています。
- 列の中央の - または \* は F2 (公式 2) に対応しています。
- 列の右側の - または \* は F3 (公式 3) に対応しています。

表の再編成は、その計算結果が公式によって設定された範囲を超える場合に、提案されます。

例えば、--- は、F1、F2、および F3 の公式の結果がその公式の設定範囲内であるために、再編成が推奨される表はないことを示しています。表記 \*-\* は、F2 の結果がその設定範囲内であっても、F1 と F3 の結果が表の再編成を提案していることを示しています。表記 \*-- は、F1 の公式のみが、その範囲を超えていることを示しています。

表名は 30 文字で切り捨てられ、31 列目の ">" 記号が表名の切り捨て位置を表します。表名への "\*" 接尾部は、これが MDC または ITC 表であることを示します。索引名への "\*" 接尾部は、これが MDC または ITC ディメンション索引であることを示します。

索引統計の用語 (公式 4 から 8) の意味は、次のとおりです。

**INDCARD**

(INDEX CARDINALITY) 索引中の索引項目数。索引によっては、これは、表のカーディナリティーとは異なることがあります。例えば、XML 列上の索引の場合、索引のカーディナリティーは、表のカーディナリティーより大きいと考えられます。

**LEAF** 索引リーフ・ページ (NLEAF) の合計数。この値は、索引がパーティション化されているかどうかによって、SYSCAT.INDEXES または SYSCAT.SYSINDEXPARTITIONS の NLEAF 列から得られます。

**ELEAF**

疑似空白リーフ・ページ (NUM\_EMPTY\_LEAFS) の数

疑似空白索引リーフ・ページは、すべての RID に削除済みのマークが付いていますが、それらが物理的には削除されていないページです。

**NDEL** 疑似削除された RID の数 (NUMRIDS\_DELETED)

疑似削除された RID とは、削除済みのマークが付いた RID のことです。この統計は、疑似空白ではないリーフ・ページ上の疑似削除された RID に関して報告します。すべての RID に削除済みのマークが付いたリーフ・ページ上の、削除マークの付いた RID は含まれません。

**KEYS** 削除マークの付いていないユニーク索引項目の数 (FULLKEYCARD)

**LEAF\_RECSIZE**

リーフ・ページ上の索引項目のレコード・サイズ。これは、すべてのオーバーヘッドを除外した索引項目の平均サイズであり、索引に関与するすべての列から得た列の平均の長さから計算されます。

**NLEAF\_RECSIZE**

非リーフ・ページ上の索引項目のレコード・サイズ。これは、すべてのオーバーヘッドを除外した索引項目の平均サイズであり、索引に關与するすべての列 (INCLUDE 列を除く) から得た列の平均の長さから計算されます。

**LEAF\_PAGE\_OVERHEAD**

内部使用に備えて索引リーフ・ページ上で予約されているスペース。

**NLEAF\_PAGE\_OVERHEAD**

内部使用に備えて予約されている索引の非リーフ・ページ上のスペース。

**INDEXPAGESIZE**

索引が置かれている表スペースのページ・サイズ。表または索引の作成時に指定します。指定しなかった場合、INDEXPAGESIZE の値は TABLEPAGESIZE の値と同じになります。

**LVLS** 索引レベルの数 (NLEVELS)**PCTFREE**

各索引ページでフリー・スペースのままにしておくパーセントを指定します。値は索引の定義時に割り当てられます。値の範囲は 0 から 99 です。デフォルト値は 10 です。

**LEAF\_RECSIZE\_OVERHEAD**

リーフ・ページ上の索引レコード・オーバーヘッド。LARGE 表スペース内の表上の索引の場合、パーティション表のオーバーヘッドは 11、他の表のオーバーヘッドは 9 です。REGULAR 表スペース内の表上の索引の場合、それぞれの値は、パーティション表の場合は 9、他の表の場合は 7 です。このルール唯一的例外は XML パスおよび XML 領域の索引であり、この場合のオーバーヘッドは常に 9 です。この情報は、容易に参照できるように、下記の表にも示されています。

**NLEAF\_RECSIZE\_OVERHEAD**

非リーフ・ページ上の索引レコード・オーバーヘッド。LARGE 表スペース内の表上の索引の場合、パーティション表のオーバーヘッドは 14、他の表のオーバーヘッドは 12 です。REGULAR 表スペース内の表上の索引の場合、それぞれの値は、パーティション表の場合は 12、他の表の場合は 10 です。このルール唯一的例外は XML パスおよび XML 領域の索引であり、この場合のオーバーヘッドは常に 12 です。この情報は、容易に参照できるように、下記の表にも示されています。

**DUPKEYSIZE**

索引リーフ・ページ上の重複キーのサイズ。LARGE 表スペース内の表上の索引の場合、パーティション表の DUPKEYSIZE は 9、他の表の DUPKEYSIZE は 7 です。REGULAR 表スペース内の表上の索引の場合、それぞれの値は、パーティション表の場合は 7、他の表の場合は 5 です。このルール唯一的例外は XML パスおよび XML 領域の索引であり、この場合の DUPKEYSIZE は常に 7 です。この情報は、容易に参照できるように、下記の表にも示されています。

## REORGCHK

表 46. *LEAF\_RECSIZE\_OVERHEAD*、*NLEAF\_RECSIZE\_OVERHEAD*、および *DUPKEYSIZE* の各値は、索引タイプ、表パーティション、および表スペース・タイプ (*REGULAR* 表スペース) の関数です。

変数	通常の表 (XML パスまたは領域の索引)	通常の表 (他のすべての索引)	パーティション表 (すべての索引)
<i>LEAF_RECSIZE_OVERHEAD</i>	9	7	9
<i>NLEAF_RECSIZE_OVERHEAD</i>	12	10	12
<i>DUPKEYSIZE</i>	7	5	7

表 47. *LEAF\_RECSIZE\_OVERHEAD*、*NLEAF\_RECSIZE\_OVERHEAD*、および *DUPKEYSIZE* の各値は、索引タイプ、表パーティション、および表スペース・タイプ (*LARGE* 表スペース\*\*) の関数です。

変数	通常の表 (XML パスまたは領域の索引)	通常の表 (他のすべての索引)	パーティション表 (すべての索引)
<i>LEAF_RECSIZE_OVERHEAD</i>	9	9	11
<i>NLEAF_RECSIZE_OVERHEAD</i>	12	12	14
<i>DUPKEYSIZE</i>	7	7	9

\*\**LARGE* 表スペース内の表上の索引の場合、索引はラージ RID を持つと見なされます。そのため、表の表スペースが *LARGE* に変換されたにも関わらず索引の再作成または再編成が未完了であると、公式によっては不正確な結果を生じることがあります。

**F4** 公式 4 の結果。

**F5** 公式 5 の結果。+++ の表記は、結果が 999 を超えるので無効であることを示しています。 **UPDATE STATISTICS** オプションを指定して **REORGCHK** を再実行するか、 **REORGCHK** コマンドに続けて **RUNSTATS** を発行してください。

**F6** 公式 6 の結果。+++ の表記は、結果が 999 を超えるので無効である可能性があることを示しています。 **UPDATE STATISTICS** オプションを指定して **REORGCHK** を再実行するか、 **REORGCHK** コマンドに続けて **RUNSTATS** を発行してください。統計が現行のものであり、有効であれば、再編成してください。

**F7** 公式 7 の結果。

**F8** 公式 8 の結果。

### REORG

この列に表示されている各ハイフン (-) は、計算結果が、対応する公式の設定範囲内であったことを示しています。各アスタリスク (\*) は、計算結果が、その対応する公式の設定範囲を超えたことを示しています。

- 左列の - または \* は F4 (公式 4) に対応しています。
- 左から 2 番目の列の - または \* は F5 (公式 5) に対応しています。
- 列の中央の - または \* は F6 (公式 6) に対応しています。
- 右から 2 番目の列の - または \* は F7 (公式 7) に対応しています。
- 右列の - または \* は F8 (公式 8) に対応しています。

索引を再編成する際の提案を以下に示します。



- 公式 1、2、および 3 の計算結果がその公式によって設定された境界を超えないで、公式 4、5、または 6 の計算結果が設定された境界を超える場合、索引を再編成することをお勧めします。
- 公式 7 の計算結果だけが設定された境界を超えて、公式 1、2、3、4、5、および 6 の結果は設定された境界内にある場合、索引再編成の **CLEANUP** オプションを使用して索引をクリーンアップすることをお勧めします。
- 公式 8 の計算結果だけが設定された境界を超える場合、索引再編成の **CLEANUP PAGES** オプションを使用して索引の疑似空白ページをクリーンアップすることをお勧めします。
- **admin\_get\_tab\_info** および **admin\_get\_index\_info** 関数内の **RECLAIMABLE\_SPACE** 列に、表および索引の再利用可能なスペースの量 (キロバイト) が示されます。この値を使用して、**RECLAIM EXTENTS** オプションを指定した再編成をいつ実行すべきか判断することができます。**RECLAIMABLE\_SPACE** の詳細、およびスペース再利用の有効性を評価する方法については、関連リンクを参照してください。

パーティション表の場合、統計の収集時によっては、公式の結果 (5 から 8) が誤ったものになる恐れがあります。データ・パーティションがデタッチされる時、そのデタッチされるパーティションの索引キーは直ちにクリーンアップされるわけではありません。ここでは、クリーンアップは保留され、キーは最終的に索引クリーナーによって除去されます。この索引クリーナーはバックグラウンドで非同期に作動しています (これは非同期の索引クリーンアップまたは **AIC** と呼ばれます)。クリーンアップが保留されている索引キーが索引内に存在すると、そのような索引キーは、統計の中でキーの一部としてカウントされません。この理由は、そのような索引キーは不可視であり、既に表の一部ではなくなっているからです。その結果、非同期の索引クリーンアップを実行する前に収集された統計は誤ったものとなります。非同期の索引クリーンアップが完了する前に **REORGCHK** コマンドを発行した場合、不正確な統計に基づいて、索引の再編成や索引のクリーンアップを指示する誤ったアラームが生成される可能性があります。非同期の索引クリーンアップの実行を開始すると、クリーンアップを必要とするデタッチされたデータ・パーティションにまだ属しているすべての索引キーが除去されるので、索引の再編成の必要がなくなることもあります。

パーティション表の場合、デタッチされたデータ・パーティションがあるなら、正確な索引統計が生成されるように、非同期の索引クリーンアップが完了した後で **REORGCHK** を発行することをお勧めします。表の中にデタッチされているデータ・パーティションがあるかどうかを調べるには、**SYSCAT.DATAPARTITIONS** カタログ・ビューの状況フィールドを確認して、値 **I** (索引クリーンアップ)、**L** (論理的にデタッチ済み)、または **D** (デタッチ済みで従属 **MQT** 付き) を探してください。

## 使用上の注意

このコマンドは、宣言済み一時表または作成済み一時表の統計情報は表示しません。

このユーティリティでは、ニックネームの使用はサポートされません。



新しいサーバー・リリースでは、新しい表と索引のフィーチャーが導入される可能性があります。こうした新規フィーチャーは **REORGCHK** ロジック、つまり **REORGCHK** が **REORG** 推奨値を計算する仕方に影響を与えることがあります。**REORGCHK** がサーバーと同レベルにないクライアントから発行される場合には、サーバーと同レベルのクライアントからの場合とは異なる結果が報告される可能性があります。

**REORGCHK** はクライアント・アプリケーションであるため、**REORGCHK** はサーバーと同レベルで実行されているクライアントから実行する必要があります。そのようにすると、最も正確なレポートが生成されます。サーバー管理作業の場合、同レベルのクライアントとサーバーを通常は使用します。

**CURRENT STATISTICS** オプションを指定していなければ、**REORGCHK** はデフォルトのオプションだけを使用してすべての列から統計を収集します。特に、列グループは収集されません。さらに、**LIKE** 統計が以前に収集されている場合、それらは **REORGCHK** によっては収集されません。収集される統計は、カタログ表に現在保管されている統計の種類によって異なります。

- いずれかの索引のカタログ内に詳細索引統計が存在する場合、すべての索引の表統計および詳細索引統計が収集されます (サンプリングは行われません)。
- 詳細索引統計が収集されない場合、すべての索引の表統計および通常の索引統計が収集されます。
- 分散統計が検出された場合、その表についての分散統計が収集されます。分散統計が収集された場合、頻度および変位値の数はデータベース構成パラメーターの設定値によって異なります。

**REORGCHK** は、8 つの異なる公式から得た統計を計算し、表またはその索引の再編成によってパフォーマンスが低下するか、または改善できるのかを判別します。表が ( **NPARTITIONS** \* 1 エクステンツ・サイズ) より小か等しいページを使用する場合、各公式に基づいて、表の再編成が推奨されなくなります。具体的には、以下のようになります。

- 非パーティション表 ( **NPARTITIONS** =1 ) の場合、しきい値は以下のようになります。

(**FPAGES** <= 1 extent size)

- パーティション表では、以下のようになります。

(**FPAGES** <= **NPARTITIONS** \* 1 extent size)

- 複数パーティションのデータベースでは、表のデータベース・パーティション・グループのデータベース・パーティションの数を考慮した後、表の再編成が推奨されないこのしきい値を次のように変更します。

**FPAGES** <= 'number of database partitions in a database partition group of the table' \* **NPARTITIONS** \* 1 extent size

**TSIZE** の計算時には、長フィールドまたは **LOB** データは検討の対象にはなりません。

**REORGCHK** は、次の公式を使用して、行の物理的なロケーションおよび表のサイズを分析します。

- 公式 F1:

$100 * \text{OVERFLOW} / \text{CARD} < 5$

表のオーバーフロー行の合計数は、行の合計数の 5% 以下でなければなりません。オーバーフロー行は、行が更新されて、新しい行のバイト数が古い行 (VARCHAR フィールド) のそれより大きくなる場合、または列が既存の表に追加される場合に作成されます。

- 公式 F2:

通常の表では、以下のようになります。

$$100 * \text{TSIZE} / ((100 - \text{TPCTFREE}) / 100 * (\text{FPAGES} - \text{NPARTITIONS}) * (\text{TABLEPAGESIZE} - 68)) > 70$$

バイト単位の表のサイズ (TSIZE) は、表に割り当てられた合計スペースの 70 % を超えていなければなりません (フリー・スペースは 30% 以下でなければなりません)。表に割り当てられる合計スペースは、表データが存在する表スペースのページ・サイズによって決まります (オーバーヘッド分の 68 バイトを差し引きします)。データ・オブジェクト内で割り当てられる最終ページは、通常はいっぱいにならないため、各パーティションごとに FPAGES から 1 を引きます (これは、FPAGES-NPARTITIONS と同じです)。

MDC 表では、以下のようになります。

$$100 * \text{TSIZE} / ((\text{ACTBLK} - \text{FULLKEYCARD}) * \text{EXTENTSIZE} * (\text{TABLEPAGESIZE} - 68)) > 70$$

FULLKEYCARD は、MDC 表の複合ディメンション索引のカーディナリティーを示します。エクステント・サイズは、ブロックごとのページ数です。バイト単位の表サイズが、必要最低限のブロック数を差し引いた後の残りのブロックの 70 パーセントを超えているかどうか、公式でチェックされます。

- 公式 F3:

$$100 * \text{NPAGES} / \text{FPAGES} > 80$$

まったく行を含まないページ数は、ページ合計数の 20% より少ない値にします (行が削除された後では、ページは空になります)。上述のとおり、(FPAGES <= NPARTITIONS \* 1 エクステント・サイズ) の場合には、表の再編成は推奨されません。したがって、F3 は計算されません。非パーティション表の場合、NPARTITIONS = 1 です。複数パーティションのデータベースでは、この条件は「FPAGES = '表のデータベース・パーティション・グループ内のデータベース・パーティション数' \* NPARTITIONS \* 1 エクステント・サイズ」に変更されません。

MDC または ITC 表の場合、式は次のようになります。

$$100 * \text{activeblocks} / (((\text{fpages\_adjust} / \text{tExtentSize}) - (\text{numberOfTablePartitions} * \text{numberOfDatabasePartitions}))$$

**REORGCHK** は、次の公式を使用して、索引および表データに対する索引のリレーションシップを分析します。

- 公式 F4:

- 非パーティション表では、以下のようになります。

$$\text{CLUSTERRATIO or normalized CLUSTERFACTOR} > 80$$

グローバル CLUSTERFACTOR および CLUSTERRATIO では、索引キーと分散キーの相関関係が考慮に入れます。クラスタリング索引比率は、80% より大きくします。複数の索引が 1 つの表に定義される場合は、これらの索引のいくつかは、低いクラスター比率を持っています (索引順序は、表の順序と

同じではありません)。これを避けることはできません。表を再編成する際に、必ず最も重要な索引を指定してください。そのクラスター比率は、通常、数の多い複写キーおよび数の多い項目を含む索引には最適ではありません。

- パーティション表では、以下のようになります。

AVGPARTITION\_CLUSTERRATIO or normalized AVGPARTITION\_CLUSTERFACTOR > 80

AVGPARTITION\_CLUSTERFACTOR および

AVGPARTITION\_CLUSTERRATIO 値は、索引キーから見て、データ・パーティションの中でデータがどの程度クラスタリングされているかを反映します。パーティション表は特定の索引キーに関して各データ・パーティションの中で完全にクラスタリングすることが可能ですが、それでも CLUSTERFACTOR および CLUSTERRATIO の値は低くなります。なぜなら、索引キーは表パーティション・キーの接頭部ではないからです。最も重要な索引キーを表パーティション・キーの接頭部として使用して、表および索引を設計してください。それに加えて、オプティマイザーは複数のデータ・パーティションを対象とした照会に関する決定を下すにあたり、グローバルなクラスタリング度を表す値を使用するため、クラスタリングの再編成を実行しつつも、キーが適切でない場合はオプティマイザーがクラスタリング索引を選択しないようにすることができます。

注: 先読みプリフェッチが有効になっている場合、公式 F4 は、REORG コマンドを実行するかどうかの良い指標とはならない可能性があります。REORG コマンドを実行するかどうかは、照会のパフォーマンスによって判断するほうが適切です。

- 公式 F5:

- 単一データベース・パーティションの場合:

$$100 * (\text{KEYS} * (\text{LEAF\_RECSIZE} + \text{LEAF\_RECSIZE\_OVERHEAD}) + (\text{INDCARD} - \text{KEYS}) * \text{DUPKEYSIZE}) / ((\text{LEAF\_NUM\_EMPTY\_LEAFS} - 1) * (\text{INDEXPAGESIZE} - \text{LEAF\_PAGE\_OVERHEAD})) > \text{MIN}(50, (100 - \text{PCTFREE}))$$

索引のリーフ・レベルで使用されるスペースは、50 パーセントと 100-PCTFREE パーセントのうち小さい方より大きくなければなりません (これは LEAF>1 のときのみ検査されます)。

- 複数パーティション・データベース環境の場合:

$$100 * (\text{KEYS} * (\text{LEAF\_RECSIZE} + \text{LEAF\_RECSIZE\_OVERHEAD}) + (\text{INDCARD} - \text{KEYS}) * \text{DUPKEYSIZE}) / ((\text{LEAF\_NUM\_EMPTY\_LEAFS} - \text{NPARTITIONS}) * (\text{INDEXPAGESIZE} - \text{LEAF\_PAGE\_OVERHEAD})) > \text{MIN}(50, (100 - \text{PCTFREE}))$$

- 公式 F6:

$$((100 - \text{PCTFREE}) * ((\text{FLOOR}((100 - \text{LEVEL2PCTFREE}) / 100 * (\text{INDEXPAGESIZE} - \text{NLEAF\_PAGE\_OVERHEAD}) / (\text{NLEAF\_RECSIZE} + \text{NLEAF\_RECSIZE\_OVERHEAD}))) * (\text{FLOOR}((100 - \text{MIN}(10, \text{LEVEL2PCTFREE})) / 100 * (\text{INDEXPAGESIZE} - \text{NLEAF\_PAGE\_OVERHEAD}) / (\text{NLEAF\_RECSIZE} + \text{NLEAF\_RECSIZE\_OVERHEAD})) ** (\text{NLEVELS} - 3)) * (\text{INDEXPAGESIZE} - \text{LEAF\_PAGE\_OVERHEAD})) / (\text{KEYS} * (\text{LEAF\_RECSIZE} + \text{LEAF\_RECSIZE\_OVERHEAD}) + (\text{INDCARD} - \text{KEYS}) * \text{DUPKEYSIZE})) < 100$$

索引の再作成がツリーのレベル数を減少させるかどうかを判別するために、この公式は現行のツリーよりも 1 つ低いレベルの索引ツリー内にあるスペースの量と必要なスペースの量との比率を検査します。1 つ低いレベルのツリーを作成しても PCTFREE が使用できるなら、再編成をお勧めします。索引項目の実際の数、NLEVELS-1 索引ツリーが取り扱える項目数の (100-PCTFREE) パーセントよ

り大きくなければなりません (NLEVELS>2 の場合のみ検査されます)。  
 NLEVELS = 2 の場合に索引の再編成が必要かどうかを判断するには、他の  
**REORGCHK** 公式に頼る必要があります。

簡易な形式の場合、公式 F6 は以下の形式に書き直すことができます。

$$\frac{\text{Amount of space needed for an index if it was one level smaller}}{\text{Amount of space needed for all the entries in the index}} < 1$$

上記の左辺が > 1 の場合、それは、既存の索引中のすべての索引項目が既存の索引より 1 つ下位のレベルの索引に収まりきる可能性があることを意味します。この場合、索引の REORG をお勧めします。

NLEVELS-1 索引に必要なスペース量は、次のように計算されます。

(The max number of leaf pages that a NLEVELS-1 index can have) \*  
 (Amount of space available to store index entries per leaf page)

詳細は次のとおりです。

The max number of leaf pages that a NLEVELS-1 index can have =  
 (No. of entries a level 2 index page can have) \*  
 (No. of entries per page on levels greater than 2) \*\*  
 (No. of levels in the intended index - 2) =

$$\left\{ \text{FLOOR} \left( \frac{(100 - \text{LEVEL2PCTFREE})}{100} \right) * \right.$$

$$\left. \left[ \frac{(\text{PageSize} - \text{Overhead})}{(\text{Avg. size of each nonleaf index entry})} \right] * \right.$$

$$\text{FLOOR} \left( \frac{(100 - \text{MIN}(10, \text{LEVEL2PCTFREE}))}{100} \right) * \left. \right.$$

$$\left. \left[ \frac{(\text{PageSize} - \text{Overhead})}{(\text{Avg. size of each nonleaf index entry})} \right] ** \right.$$

$$\left. (\text{NLEVELS} - 3) \right\}$$

(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-PCTFREE)/100 \* (INDEXPAGESIZE - LEAF\_PAGE\_OVERHEAD)) =  
 ( Used space per page \* (PageSize - Overhead) )

The amount of space needed for all index entries:  
 KEYS \* (LEAF\_REC\_SIZE + LEAF\_REC\_SIZE\_OVERHEAD) +  
 (INDCARD - KEYS) \* DUPKEYSIZE

$(KEYS * (LEAF\_RECSIZE + LEAF\_RECSIZE\_OVERHEAD))$  は、索引内の各キー値の最初の出現箇所に使用されるスペースを表し、 $((INDCARD - KEYS) * DUPKEYSIZE)$  は、キー値のその後の (重複) 出現箇所に使用されるスペースを表します。

- 公式 F7:

$$100 * (NUMRIDS\_DELETED / (NUMRIDS\_DELETED + IND CARD)) < 20$$

疑似空白ではないページ上の疑似削除された RID の数は 20% 未満でなければなりません。

- 公式 F8:

$$100 * (NUM\_EMPTY\_LEAFS/LEAF) < 20$$

疑似空白リーフ・ページの数、リーフ・ページの合計数の 20% 未満でなければなりません。

多数の表で統計を実行すると、表が大きい場合には特に時間がかかります。

### 索引圧縮に関する使用上の注意

公式 F5 は、キーが必要とするスペース量と、割り振られているスペース量の比率を判別します。公式 F6 は、索引の再作成がツリーのレベルを減少させるかどうかを判別します。以下の公式は現行のツリーよりも 1 つ低いレベルの索引ツリー内にあるスペースの量と必要なスペースの量との比率を検査します。この公式は、すべての索引項目に必要なスペース量に基づいています。どちらの公式も、すべての索引項目に必要なスペース量を使用します。

圧縮解除索引内にあるすべての索引項目に必要なスペース量は、以下のようになります。

$$KEYS * (LEAF\_RECSIZE + LEAF\_RECSIZE\_OVERHEAD) + (INDCARD - KEYS) * DUPKEYSIZE$$

ここで、LEAF\_RECSIZE は索引キーの平均サイズで、DUPKEYSIZE は RID のサイズです。

圧縮索引の場合、LEAF\_RECSIZE は接頭部圧縮の影響を受けます。DUPKEYSIZE は、索引上の重複キーのサイズを測定する確実な方法とは言えません。圧縮索引に必要なスペース量は、圧縮解除されているすべての索引項目で必要とされるスペース量に索引圧縮比率を乗算して求めます。

$$(KEYS * (LEAF\_RECSIZE + LEAF\_RECSIZE\_OVERHEAD) + (INDCARD - KEYS) * DUPKEYSIZE) * COMPRESSION\_RATIO$$

ここで、COMPRESSION\_RATIO は索引における見積もり索引圧縮比率です。COMPRESSION\_RATIO は、以下のようにして計算されます。

$$(100 - PCT\_PAGES\_SAVED) / 100$$

ここで、PCT\_PAGES\_SAVED は索引圧縮によって節約されるリーフ・ページのパーセントの見積もりです。この値は、カタログから取得されます。統計が収集されない場合には、カタログ内で PCT\_PAGES\_SAVED は -1、COMPRESSION\_RATIO は 1 となります。

**REORGCHK** コマンドと **REORGCHK\_IX\_STATS** プロシージャはどちらも **PCT\_PAGES\_SAVED** 値を示します。

## パーティション表に関する使用上の注意

データ・パーティション表に関して、**REORGCHK** は表および表のデータ・パーティションの統計と再編成推奨値を戻します。

表統計および再編成推奨値に関して、**REORGCHK** は、表の **SCHEMA.NAME**、表レベル統計、および **REORG** 推奨値を含む表情報をリストします。表情報に続いて、各データ・パーティション情報に関する情報がリストされます。各パーティションに対して、表の **SCHEMA.NAME**、パーティション名、パーティションの表統計、およびパーティションの再編成推奨値などの情報があります。

索引統計および再編成推奨値に関して、**REORGCHK** は、各表の **SCHEMA.NAME** に続けて、完全修飾された索引名、および表の各非パーティション索引に対する索引統計と索引の再編成推奨値を戻します。表にパーティション索引が含まれる場合、**REORGCHK** は、非パーティション索引の後にパーティション索引の情報を戻します。**REORGCHK** は、完全修飾された索引名、パーティション名、パーティションの索引統計、およびパーティションに対する索引の再編成推奨値など表のデータ・パーティションごとの情報を戻します。

データ・パーティション表のデータ可用性を高めるために、特定のデータ・パーティションの再編成を行うことができます (推奨された場合)。**ON DATA PARTITION** 節を指定した **REORG TABLE** では、1 つの表の 1 つのパーティションに対する再編成がサポートされます。

パーティション索引の場合、**ON DATA PARTITION** 節を指定して **REORG INDEXES ALL** を使用し、特定のデータ・パーティションの全索引に対する索引再編成を行うことができます (推奨された場合)。

**REORG INDEX** コマンドを使用して、データ・パーティション表の非パーティション索引を再編成することができます。

データ・パーティション表の再編成推奨値に関しては、**REORG** の欄を参照してください。

以下に、パーティション表 **MYDPARTT1** に対する表統計の出力例を示します。この表には、デフォルト・パーティション名 **PART0**、**PART1**、および **PART2** という 3 つのデータ・パーティションがあります。

Table statistics:

SCHEMA.NAME	CARD	OV	NP	FP	ACTBLK	TSIZE	F1	F2	F3	REORG
Table: USER1.MYDPARTT1	-	-	-	-	-	-	-	-	-	----
Table: USER1.MYDPARTT1 Data Partition: PART0	-	-	-	-	-	-	-	-	-	----
Table: USER1.MYDPARTT1 Data Partition: PART1	-	-	-	-	-	-	-	-	-	----
Table: USER1.MYDPARTT1	-	-	-	-	-	-	-	-	-	----

## REORGCHK

Data Partition: PART2

以下に、パーティション表 MYDPARTT1 の索引統計の出力例を示します。この表には、3 つのデータ・パーティション、1 つの非パーティション索引 MYNONPARTIDX1、および 1 つのパーティション索引 MYDPARTIDX1 があります。

Index statistics:

SCHEMA.NAME	INDCARD	LEAF	ELEAF	LVLS	NDEL	KEYS ...	F4	F5	F6	F7	F8	REORG
-----												
Table: USER1.MYDPARTT1						...						
Index: USER1.MYNONPARTIDX1												
Index: USER1.MYDPARTIDX1	-	-	-	-	-	- ...	-	-	-	-	-	-----
Data Partition: PART0												
Index: USER1.MYDPARTIDX1	-	-	-	-	-	- ...	-	-	-	-	-	-----
Data Partition: PART1												
Index: USER1.MYDPARTIDX1	-	-	-	-	-	- ...	-	-	-	-	-	-----
Data Partition: PART2												
	-	-	-	-	-	- ...	-	-	-	-	-	-----
-----												
						...						



## RESET ADMIN CONFIGURATION

接続先のノードの DB2 Administration Server (DAS) 構成ファイルにある項目をリセットします。DAS は、DB2 サーバーのリモート管理を有効にする特別な管理ツールです。

この値は、常にリモート・クライアントのサーバーになっているノード・タイプによってリセットされます。DAS パラメーターのリストは、**UPDATE ADMIN CONFIGURATION** コマンドの説明を参照してください。

**重要:** DB2 Administration Server (DAS) は、バージョン 9.7 で非推奨となり、将来のリリースで除去される可能性があります。DAS は、DB2 pureScale環境ではサポートされていません。リモート管理のためには、Secure Shell プロトコルを使用するソフトウェア・プログラムを使用してください。詳しくは、『DB2 Administration Server (DAS) が推奨されなくなった』( )を参照してください。

### 有効範囲

このコマンドは、接続先のシステムの管理ノードで DAS 構成ファイルをリセットします。

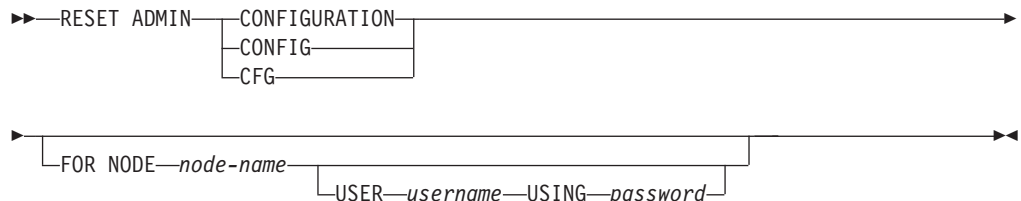
### 許可

DASADM

### 必要な接続

パーティション。リモート・システムの DAS 構成をリセットする場合は、**FOR NODE** オプションと管理ノード名を使用してシステムを指定します。

### コマンド構文



### コマンド・パラメーター

**FOR NODE** *node-name*

DAS 構成パラメーターをリセットする管理ノードの名前をここに入力します。

**USER** *username* **USING** *password*

リモート・システムへの接続にユーザー名とパスワードが必要な場合は、この情報を入力します。

## RESET ADMIN CONFIGURATION

### 使用上の注意

リモート・システムの DAS 構成パラメーターをリセットするには、管理ノード名を **FOR NODE** オプションの引数にしてシステムを指定し、そのノードへの接続にユーザー名とパスワードの許可が必要な場合は、ユーザー名とパスワードを指定します。

DAS 構成パラメーターのリストの表示または印刷を行うには、**GET ADMIN CONFIGURATION** コマンドを使用してください。ADMIN パラメーターの値を変更するには、**UPDATE ADMIN CONFIGURATION** コマンドを使用してください。

オンラインで更新可能な DAS 構成パラメーターへの変更は、即時に行われます。それ以外の変更は、**db2admin** コマンドで DAS が再始動し、変更がメモリーにロードされた後に有効になります。

エラーが生じた場合には、DAS 構成ファイルは変更されません。

DAS 構成ファイルは、そのチェックサムが無効であると、リセットすることができません。このような状況は、適切なコマンドを使用せずに手作業で DAS 構成ファイルが変更された場合などに起こります。チェックサムが無効な場合は、DAS を一度ドロップしてから再作成し、その構成ファイルをリセットする必要があります。

## RESET ALERT CONFIGURATION

特定のオブジェクトのヘルス・インディケーター設定を、そのオブジェクト・タイプの現行デフォルトにリセットするか、またはオブジェクト・タイプの現行のデフォルトのヘルス・インディケーター設定を、インストール時のデフォルトにリセットします。

**重要:** バージョン 9.7 でヘルス・モニターが非推奨になったため、このコマンドおよび API は推奨されておらず、将来のリリースで除去される可能性があります。これはDB2 pureScale環境ではサポートされていません。詳しくは、『ヘルス・モニターが推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055045.html>) を参照してください。

### 許可

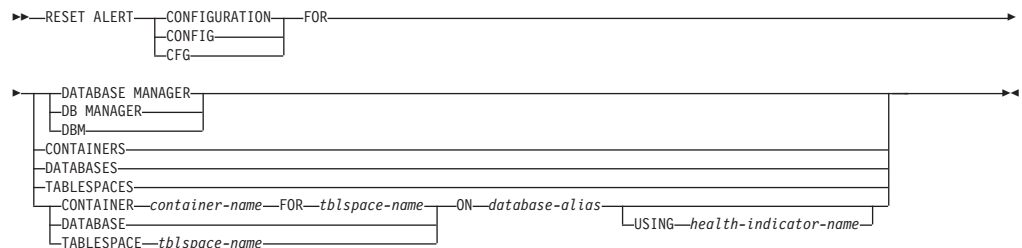
以下の権限のいずれか。

- SYSADM
- SYSMANT
- SYSCTRL

### 必要な接続

インスタンス。明示的なアタッチは必要ありません。

### コマンド構文



### コマンド・パラメーター

#### DATABASE MANAGER | DB MANAGER | DBM

データベース・マネージャーでアラート設定をリセットします。

#### CONTAINERS

データベース・マネージャーが管理するすべての表スペース・コンテナのデフォルトのアラート設定を、インストール時のデフォルトにリセットします。これは、カスタム設定を持たないすべての表スペース・コンテナに適用される設定です。カスタム設定は、**CONTAINER container-name FOR tblspace-name ON database-alias** 節を使って定義されます。

#### DATABASES

データベース・マネージャーが管理するすべてのデータベースのアラート設定をリセットします。これは、カスタム設定を持たないすべてのデータベースに適用される設定です。カスタム設定は、**DATABASE ON database-alias** 節を使って定義されます。

## RESET ALERT CONFIGURATION

### TABLESPACES

データベース・マネージャーが管理するすべての表スペースのデフォルトのアラート設定を、インストール時のデフォルトにリセットします。これは、カスタム設定を持たないすべての表スペースに適用される設定です。カスタム設定は、**TABLESPACE** *tblspace-name* **ON** *database-alias* 節を使って定義されます。

### CONTAINER *container-name* **FOR** *tblspace-name* **ON** *database-alias*

**ON** *database-alias* 節を使って指定したデータベース上で、**FOR** *tblspace-name* 節を使って指定した表スペースの、*container-name* という名前の表スペース・コンテナのアラート設定をリセットします。この表スペース・コンテナにカスタム設定がある場合、これらの設定は除去され、現行の表スペース・コンテナのデフォルトが使用されます。

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

**ON** *database-alias* 節を使って指定したデータベースのアラート設定をリセットします。このデータベースにカスタム設定がある場合、これらの設定は除去され、インストール時のデフォルトが使用されます。

### TABLESPACE *tblspace-name* **ON** *database-alias*

**ON** *database-alias* 節を使って指定したデータベース上で、*tblspace-name* という名前の表スペースのアラート設定をリセットします。この表スペースにカスタム設定がある場合、これらの設定は除去され、インストール時のデフォルトが使用されます。

### USING *health-indicator-name*

アラート構成がリセットされるヘルス・インディケーターのセットを指定します。ヘルス・インディケーター名は 2 文字のオブジェクト ID で構成され、その後にインディケーターの測定対象を説明する名前が続きます。以下に例を示します。

```
db.sort_privmem_util
```

このオプションを指定しない場合は、指定したオブジェクトまたはオブジェクト・タイプのすべてのヘルス・インディケーターがリセットされます。

## RESET DATABASE CONFIGURATION

特定データベースの構成をシステム・デフォルトにリセットします。

### 有効範囲

このコマンドはデフォルトですべてのメンバーを更新します。ただし、**MEMBER** が 1 つのメンバーのみリセットするように指定されている場合は除きます。

### 許可

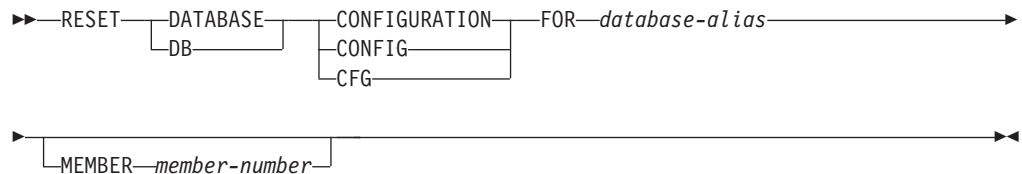
以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMANT

### 必要な接続

インスタンス。明示的なアタッチは必要ありません。データベースがリモートとして示されている場合、リモート・ノードへのインスタンス・アタッチはコマンドの持続期間の間、ずっと確立されたままになります。

### コマンド構文



### コマンド・パラメーター

#### FOR *database-alias*

構成がシステム・デフォルトにリセットされるそのデータベースの別名を指定します。

#### MEMBER *member-number*

データベース構成のリセットが特定のメンバーに適用される場合、このパラメーターが使用されることがあります。このパラメーターが指定されていない場合、リセットはすべてのメンバーに対して有効です。

### 例

#### マルチパーティション・インスタンス上でのデータベース構成のリセット

ユーザーは、`db2nodes.cfg` で定義された次の 4 個のデータベース・パーティションがあるパーティション・データベース環境を持っています。

```

10 gilera 0
20 gilera 1
30 motobi 0
40 motobi 1
  
```

## RESET DATABASE CONFIGURATION

ユーザーはインスタンス上で **SAMPLE** データベースを作成しました。 **SAMPLE** のカタログ・パーティションは、データベース・パーティション番号 10 にあります。ここでは、ユーザーがシステム **motobi** にログオンするとします。

次のコマンドは、すべてのデータベース・パーティション上でデータベース **SAMPLE** についてのすべてのデータベース構成値をリセットします。

```
db2 reset db cfg for sample
```

次のコマンドは、データベース・パーティション番号 30 のみ、データベース **SAMPLE** についてのすべてのデータベース構成値をリセットします。

```
db2 reset db cfg for sample dbpartitionnum 30
```

または

```
export DB2NODE=30
db2 reset db cfg for sample
```

### 使用上の注意

データベース構成パラメーターのリストを表示または印刷するには、**GET DATABASE CONFIGURATION** コマンドを使用してください。

構成可能なパラメーターの値を変更するには、**UPDATE DATABASE CONFIGURATION** コマンドを使用してください。

データベース構成ファイルへの変更は、ファイルがメモリーにロードされた後のみ有効になります。これを行う前にすべてのアプリケーションはデータベースから切断されている必要があります。

エラーが発生した場合、データベース構成ファイルは変更されません。

チェックサムが無効である場合には、データベースの構成ファイルは、リセットできません。適当なコマンドを使用しないでデータベース構成ファイルを変更するとこれが発生することがあります。これが発生する場合、データベースをリストアしてデータベース構成ファイルをリセットする必要があります。

**RESET DATABASE CONFIGURATION** コマンドを実行すると、データベース構成パラメーターが文書化されているデフォルトの構成値にリセットされ、**auto\_runstats** は ON になります。 **Self\_tuning\_mem** は、非パーティション・データベース環境の場合は ON にリセットされ、パーティション・データベース環境の場合は OFF にリセットされます。

### 互換性

以前のバージョンとの互換性:

- **DB2\_ENFORCE\_MEMBER\_SYNTAX** レジストリー変数が ON に設定されている場合を除き、**DBPARTITIONNUM** を **MEMBER** の代わりに使用できます。

## RESET DATABASE MANAGER CONFIGURATION

現在接続されているデータベースを含むインスタンスについて、データベース・マネージャの構成ファイルのパラメーターをシステム・デフォルトこの値は、ノード・タイプに基づいたデフォルト値にリセットされます。

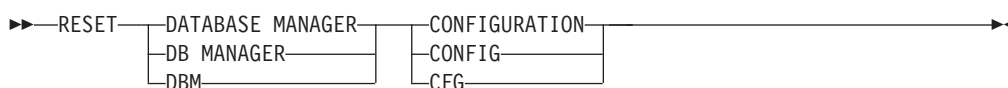
### 許可

SYSADM

### 必要な接続

なし、またはインスタンス。インスタンス接続は、ローカル・データベース・マネージャ構成操作を実行する場合には必ずしも必要ではありませんが、リモート・データベース・マネージャ構成操作を実行する場合には必須です。リモート・インスタンスに対するデータベース・マネージャ構成を更新するためには、最初にそのインスタンスにアタッチする必要があります。構成パラメーターをオンラインで更新する場合も、まずインスタンスにアタッチする必要があります。

### コマンド構文



### コマンド・パラメーター

なし

### 使用上の注意

このコマンドは、インストール・プログラムによって設定されたすべてのパラメーターをリセットします。パラメーターがリセットされると、DB2 を再始動するときエラー・メッセージが戻される原因となる場合があります。例えば、**svcsname** パラメーターがリセットされると、DB2 を再始動しようとすると、ユーザーは SQL5043N エラー・メッセージを受け取ります。

このコマンドを実行する前に、既存の設定値を参照できるようにするために、**GET DATABASE MANAGER CONFIGURATION** コマンドの出力をファイルに保存します。個々の設定値は、**UPDATE DATABASE MANAGER CONFIGURATION** コマンドを使用して更新できます。

インストール・プログラムで設定される **svcsname** パラメーターの、ユーザーによる修正は推奨されません。

データベース・マネージャ構成パラメーターのリストの表示または印刷を行うには、**GET DATABASE MANAGER CONFIGURATION** コマンドを使用してください。構成可能なパラメーターの値を変更するには、**UPDATE DATABASE MANAGER CONFIGURATION** コマンドを使用してください。

これらのパラメーターについての詳細は、構成パラメーターおよび個々のパラメーターについてのサマリー・リストを参照してください。



## RESET DATABASE MANAGER CONFIGURATION

データベース・マネージャー構成ファイルへの変更の一部は、ファイルがメモリーにロードされた後にのみ有効になります。オンラインで構成できるパラメーターと構成できないパラメーターについては、構成パラメーターの一覧をご覧ください。即時にリセットされないサーバー構成パラメーターは、**db2start** の実行中にリセットされます。クライアント構成パラメーターの場合、パラメーターは次にアプリケーションを開始するときのリセットされます。クライアントがコマンド行プロセッサである場合は、**TERMINATE** を呼び出す必要があります。

エラーが生じた場合には、データベース・マネージャー構成ファイルは変更されません。

データベース・マネージャー構成ファイルは、そのチェックサムが無効であると、リセットすることができません。このような状況は、適切なコマンドを使用せずに手動で構成ファイルが編集された場合などに発生します。チェックサムが無効な場合は、インスタンスを再作成する必要があります。

## RESET MONITOR

指定されたデータベース、またはすべてのアクティブ・データベースの、内部のデータベース・システム・モニター・データ域をゼロにリセットします。内部のデータベース・システム・モニター・データ域には、データベース用のデータ域のほかに、データベースに接続されるすべてのアプリケーション用のデータ域が含まれます。

### 有効範囲

このコマンドは、現在アタッチされているメンバーに対して呼び出され、デフォルトではそのメンバーについての情報のみを返します。現在アタッチされているメンバーは、デフォルトでは、コマンドが実行されるホストになります。ホストに対して複数のメンバーがある場合、現在アタッチされているメンバーは、そのホスト上の `db2nodes.cfg` ファイルにリストされている最初のメンバーです。

現在アタッチされているメンバーではないメンバーに対してコマンドを実行する場合、特定のメンバーに対してこのコマンドを実行するには **AT MEMBER** オプションを指定し、すべてのメンバーに対してこのコマンドを実行して集約された結果を受け取るには **GLOBAL** オプションを指定してください。

後続のコマンドを実行する対象の、現在アタッチされているメンバーを変更するには、**SET CLIENT** コマンドの **ATTACH\_MEMBER** パラメーターを使用します。このクライアント設定を有効にするには、アプリケーションを **DETACH** してから再び **ATTACH** する必要があります。

### 許可

以下の権限のいずれか。

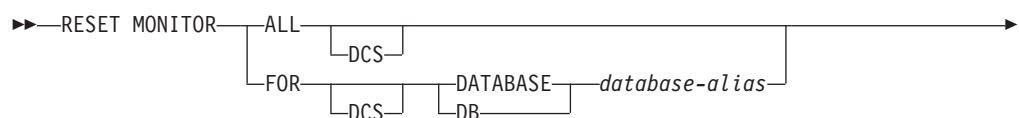
- SYSADM
- SYSCTRL
- SYSMANT
- SYSMON

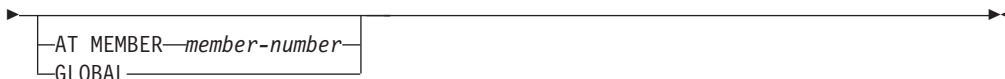
### 必要な接続

インスタンス。インスタンス接続が存在しない場合は、デフォルトのインスタンス接続が作成されます。

リモート・インスタンス (または別のローカル・インスタンス) 用のモニター・スイッチをリセットするには、まず最初にそのインスタンスにアタッチすることが必要です。

### コマンド構文





## コマンド・パラメーター

**ALL** このオプションは、すべてのデータベースについて内部カウンターがリセットされることを指示します。

**FOR DATABASE** *database-alias*

このオプションは、別名 *database-alias* を持つデータベースの内部カウンターのみがリセットされることを指示します。

**DCS** このキーワードは、指定された節に従って、次のいずれかの内部カウンターをリセットします。

- すべての DCS データベース
- 特定の DCS データベース

**AT MEMBER** *member-number*

内部カウンターのリセットの対象となるメンバーを指定します。

**GLOBAL**

パーティション・データベース環境または DB2 pureScale 環境におけるすべてのメンバーの内部カウンターをリセットします。

## 使用上の注意

各プロセス (アタッチ) には、モニター・データについての独自のプライベート・ビューがあります。あるユーザーがモニター・スイッチをリセットしたり、オフにしたりしても、他のユーザーはその影響を受けません。モニター・スイッチ構成パラメーターの設定を、モニター・スイッチの一括変更で変更してください。

**ALL** を指定すると、一部のデータベース・マネージャー情報が、戻されるデータの一貫性を維持するためにリセットされ、一部のメンバー・レベルのカウンターもリセットされます。

## 互換性

以前のバージョンとの互換性:

- **DB2\_ENFORCE\_MEMBER\_SYNTAX** レジストリー変数が ON に設定されている場合を除き、**DBPARTITIONNUM** または **NODE** を **MEMBER** の代わりに使用できます。

## RESTART DATABASE

異常終了し、矛盾した状態のままであるデータベースを再始動します。ユーザーが CONNECT 特権を持っている場合には、**RESTART DATABASE** が正常終了すると、アプリケーションはデータベースに接続されたままとなります。

### 有効範囲

このコマンドは、それが実行されたデータベース・パーティションに対してだけ影響を与えます。DB2 pureScale 環境では、このコマンドはすべての中断状態のメンバーについて、入出力書き込み操作を再開します。

DB2 pureScale 環境では、このコマンドは必要に応じて、グループ・クラッシュ・リカバリーをトリガーします。それはグループのメンバーすべてに対するクラッシュ・リカバリー、またはメンバー・クラッシュ・リカバリーを実行します。

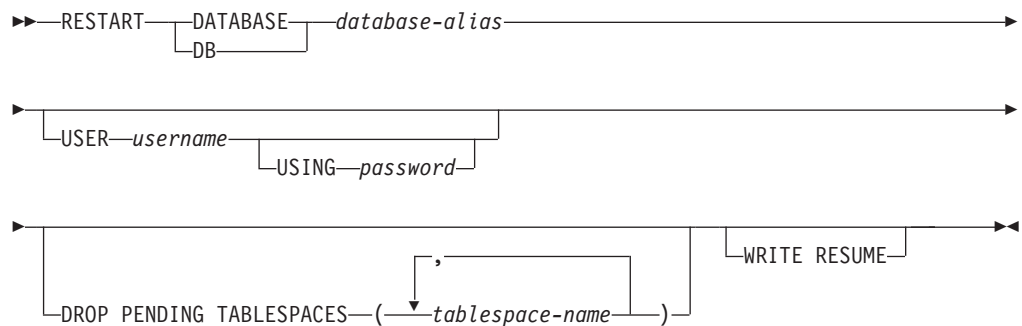
### 許可

なし。

### 必要な接続

このコマンドは、データベース接続を確立します。

### コマンド構文



### コマンド・パラメーター

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

再始動するデータベースを識別します。

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

データベースが再始動する際のユーザー名を識別します。

#### **USING** *password*

*username* を認証するのに使用されるパスワード。パスワードを省略すると、ユーザーに入力を求めるプロンプトが出ます。

#### **DROP PENDING TABLESPACES** *tablespace-name*

表スペース・コンテナに問題が発生しても、データベース再始動操作を正常に完了するよう指定します。

指定した表スペース用のコンテナで再始動処理中に問題が発生した場合、それに対応する表スペースは、再始動操作後に使用できなくなります (ドロ

## RESTART DATABASE

ップ・ペンディング状態になります)。表スペースがドロップ・ペンディング状態のとき、実行可能なアクションは表スペースをドロップすることだけです。

メンバー・クラッシュ・リカバリー中にコンテナで問題が生じた場合、メンバー・クラッシュ・リカバリー操作は失敗します。この状態を修正するには、**db2stop -force** コマンドを使用してすべてのメンバーをシャットダウンし、**RESTART DATABASE** コマンドを再発行することにより、グループ・クラッシュ・リカバリーを手動で開始します。グループ・クラッシュ・リカバリーの後は、問題のコンテナに対応する表スペースは使用不可になります(詳しくは、損傷した表スペースでのリカバリーに関するトピックを参照してください)。

循環ロギングの場合には、問題の発生している表スペースが再始動障害の原因となります。問題の発生している表スペース名のリストは、データベースの再始動操作が失敗した原因がコンテナの問題である場合には、管理通知ログに含まれています。データベースに **SYSTEM TEMPORARY** 表スペースが 1 つしかなく、しかもその **SYSTEM TEMPORARY** 表スペースがドロップ・ペンディング状態である場合は、データベースの再始動操作が正常に完了したらすぐに新しい **TEMPORARY** 表スペースを作成する必要があります。

### WRITE RESUME

入出力書き込み操作が中断状態だったときにクラッシュしたデータベースの再始動を強制します。クラッシュ・リカバリーを実行する前に、このパラメーターは入出力書き込み操作を再開します。DB2 pureScale 環境では、このパラメーターはすべての中断状態のメンバーについて、入出力書き込み操作を再開します。

入出力書き込み操作を中断するのに使用した接続が現在ハングしていて、その後に接続を試行してもすべてハングしてしまう場合にも、**WRITE RESUME** パラメーターを使用できます。このケースでパラメーターを使用する場合、**RESTART DATABASE** コマンドは、クラッシュ・リカバリーを実行せずに、データベースに対する入出力書き込み操作を再開します。**RESTART DATABASE** コマンドに **WRITE RESUME** パラメーターを指定して使用すると、データベースのクラッシュ後に使用する場合にのみ、クラッシュ・リカバリーを実行します。

### 使用上の注意

データベースへの接続を試行すると、データベースを再始動する必要があることを示すエラー・メッセージが戻される場合、このコマンドを発行してください。このエラー・メッセージは、このデータベースを用いた前のセッションが異常終了した(例えば、電源障害により)場合にのみ生成されます。

#### パーティション・データベース環境

パーティション・データベース・システムでは、未確定トランザクションを解決するために、以下の例で示すように、すべてのデータベース・パーティション上で **RESTART DATABASE** コマンドを発行する必要があります。

```
db2_all "db2 restart database database-alias"
```

パーティション・データベース環境内の 1 つのデータベース・パーティションだけでデータベースを再始動した場合、データベースを再始動する必要があることを示すメッセージが、後続のデータベース照会で戻されることがあります。このメッセージは、照会が依存しているデータベース・パーティションのデータベースを再始動しなかったために戻されます。すべてのデータベース・パーティションでデータベースを再始動すれば、問題を解決できます。

## DB2 pureScale 環境

DB2 pureScale 環境では、**RESTART DATABASE** コマンドは、クラッシュ・リカバリーを手動で呼び出す唯一の方法です。データベース・マネージャは、グループ・クラッシュ・リカバリーまたはメンバー・クラッシュ・リカバリーが必要かどうかを自動的に判別します。グループ・クラッシュ・リカバリーの完了後に、グループ・クラッシュ・リカバリーを実行しなかったいずれかのメンバーに未確定トランザクションが存在する場合には、そうした未確定トランザクションの解決を可能にするために、それらのメンバーにメンバー・クラッシュ・リカバリーを実行する必要があります。

クラッシュの時点で、データベースに対する書き込み操作が中断の途中だった場合には、**RESTART DATABASE** コマンドを使用することにより、クラッシュ・リカバリーを実行してください。SET WRITE SUSPEND 操作が別のメンバーで実行されている場合、クラッシュ・リカバリーを実行する前に、SET WRITE SUSPEND 操作の完了を待機する必要があるかもしれません。それが完了した後に、**RESTART DATABASE** コマンドを **WRITE RESUME** パラメーター付きで発行して、データベースを再始動し、すべての中断状態メンバーの書き込み操作を再開してください。

クラッシュの時点で、データベースでの書き込み操作が正常に中断状態に入っていた場合には、クラッシュ・リカバリーを実行できる 2 つの方法があります。

- **RESTART DATABASE** コマンドに **WRITE RESUME** パラメーターを付けて使用する。
- アクティブなメンバーから、**SET WRITE** コマンドに **RESUME** パラメーターを付けて発行し、次いでクラッシュしたメンバーから **RESTART DATABASE** コマンドを発行する。

---

## RESTORE DATABASE

**RESTORE DATABASE** コマンドは、DB2 バックアップ・ユーティリティーを使用してバックアップされた、損傷のある、または破壊されたデータベースを再作成します。リストアされたデータベースは、バックアップ・コピーが行われた時と同じ状態になります。

このユーティリティーは、以下のサービスも実行できます。

- データベースに別のイメージを上書きしたり、バックアップ・コピーを新しいデータベースにリストアしたりします。
- DB2 バージョン 9.8 のリストア・ユーティリティーを使用して、他のいずれかの DB2 ソフトウェアのバージョンでバックアップされたバックアップ・イメージをリストアすることはできません。
- DB2 バージョン 9.5 でバックアップされたバックアップ・イメージを DB2 バージョン 9.7 でリストアします。
  - データベース・アップグレードが必要な場合、これはリストア操作の終了時に自動的に起動されます。
- バックアップ操作の時点でデータベースのロールフォワード・リカバリーが有効になっていた場合は、リストア操作が正常に完了した後に、ロールフォワード・ユーティリティーを起動することによって、データベースを元の状態に戻すことができます。
- 表スペース・レベルのバックアップをリストアします。
- **TRANSPORT** オプションを使用して、データベース・バックアップ・イメージに含まれる表スペース、ストレージ・グループ、および SQL スキーマのセットをデータベースに転送します (DB2 バージョン 9.7 フィックスパック 2 以降のフィックスパック)。TRANSPORT オプションは DB2 pureScale 環境ではサポートされません。
- このコマンドの発行時にそのデータベース名が存在している場合には、バックアップ・イメージが作成された時点のとおり、すべてのストレージ・グループを置換して再定義します (ユーザーが別のリダイレクトを指定していない限り)。

異なるオペレーティング・システムおよびハードウェア・プラットフォームの間で DB2 データベース・システムによってサポートされるリストア操作の詳細については、「データ・リカバリーと高可用性 ガイドおよびリファレンス」の『異なるオペレーティング・システムおよびハードウェア・プラットフォーム間のバックアップおよびリストア操作』を参照してください。

オペレーティング・システムまたはワード・サイズ (32 ビットか 64 ビットか) が異なる場合、増分イメージおよび「差分イメージ」(以前のキャプチャー時との差異だけをキャプチャーするイメージ) をリストアすることはできません。

ある環境から別の環境へのリストア操作を行った後は、非増分バックアップが実施されるまで、増分バックアップまたは差分バックアップを実行できません。(同じ環境でのリストア環境の場合、この制限はありません。)

ある環境から別の環境へのリストア操作が成功した場合でも、いくつかの注意事項があります。パッケージは、使用する前に再バインドする必要があります (**BIND** コマンド、**REBIND** コマンド、または **db2rbind** ユーティリティーを使用)。SQL プロ



シージャーは、ドロップしてから再作成する必要があります。また、外部ライブラリーは、新しいプラットフォーム上ですべて再ビルドする必要があります。(同じ環境にリストアする場合、これらの点は該当しません。)

既存のデータベースと既存のコンテナに対して実行されるリストア操作では、同じコンテナと表スペース・マップが再利用されます。

新規のデータベースに対して実行されるリストア操作では、すべてのコンテナが再取得され、最適化された表スペース・マップが再作成されます。既存のデータベースに対して実行されるリストア操作でも 1 つ以上のコンテナがない場合は、すべてのコンテナが再取得され、最適化された表スペース・マップが再作成されます。

## 有効範囲

このコマンドは、それが実行されたノードに対してだけ影響を与えます。

オンラインで SYSCATSPACE をリストアすることはできません。

## 許可

既存のデータベースにリストアするには、次の権限のいずれかが必要です。

- SYSADM
- SYSCTRL
- SYSMMAINT

新規のデータベースにリストアするには、次の権限のいずれかが必要です。

- SYSADM
- SYSCTRL

ユーザー名を指定する場合、そのユーザーにはデータベースに対する CONNECT 権限が必要です。

## 必要な接続

必要な接続は、リストア・アクションの種類によって異なります。

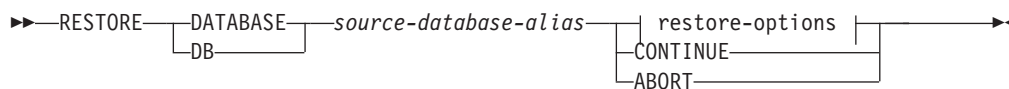
- 既存のデータベースにリストアするには、データベース接続が必要です。このコマンドは、指定されたデータベースへの排他接続を自動的に確立します。
- 新しいデータベースにリストアするには、インスタンスおよびデータベース接続が必要です。データベースを作成するには、インスタンス接続が必要です。

現行のインスタンスとは異なるインスタンスで新規のデータベースへのリストアを行うには、まず、新規のデータベースを存在させるインスタンスにアタッチすることが必要です。新規インスタンスは、ローカルでもリモートでもかまいません。現行のインスタンスは、**DB2INSTANCE** 環境変数の値によって定義されます。

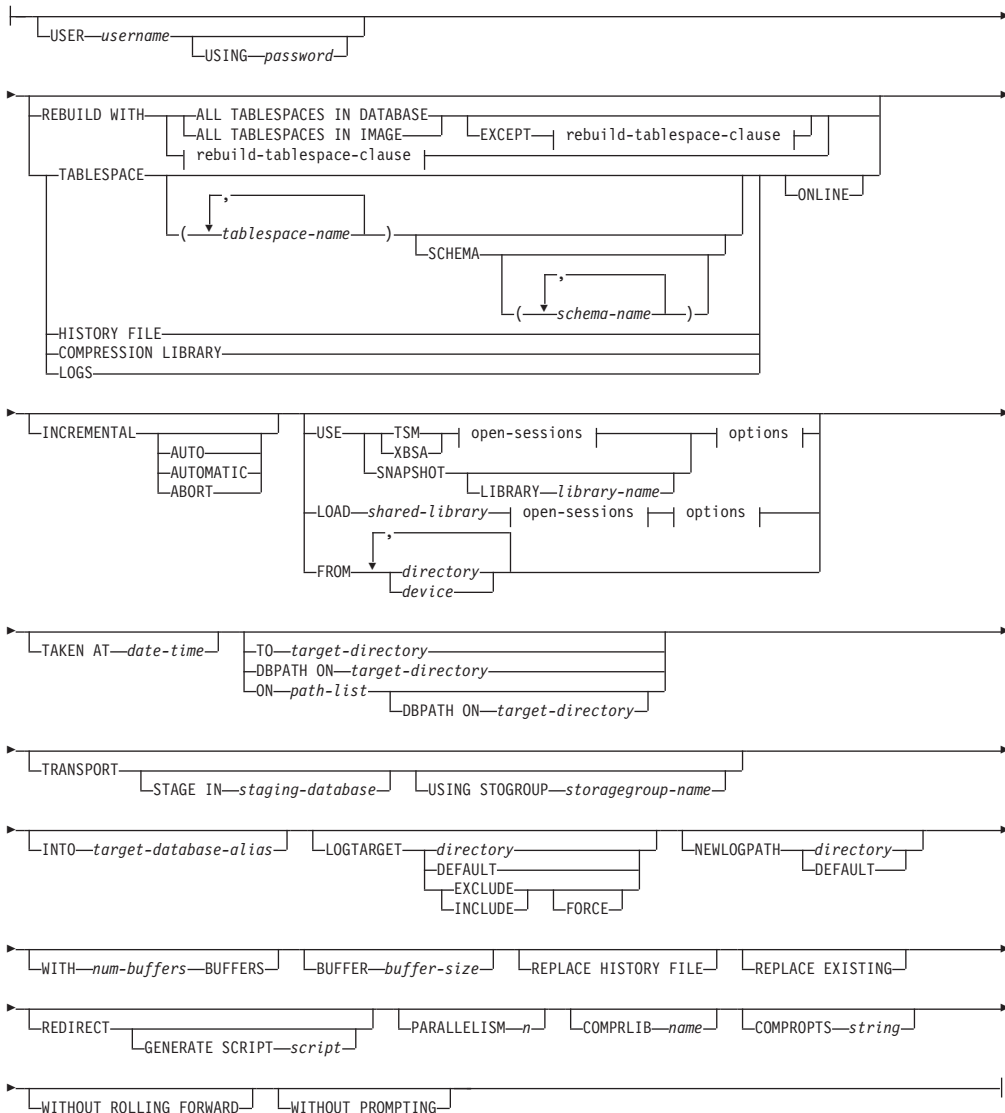
- スナップショット・リストアの場合、インスタンス とデータベース の接続が必要です。

# RESTORE DATABASE

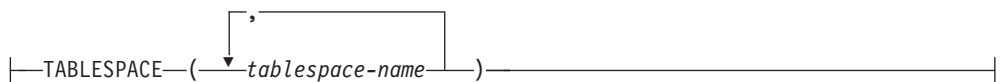
## コマンド構文



### restore-options:



### rebuild-tablespace-clause:



**open-sessions:**

```
|-----|
| OPEN num-sessions SESSIONS |
|-----|
```

**options:**

```
|-----|
| OPTIONS "options-string" |
| @file-name |
|-----|
```

**コマンド・パラメーター****DATABASE** *source-database-alias*

バックアップが取得されるソース・データベースの別名です。

**CONTINUE**

コンテナが再定義されていること、およびリダイレクトしたリストア操作の最終ステップを実行する必要があることを指定します。

**ABORT**

このパラメーターは以下を指定します。

- リダイレクトしたリストア操作を停止します。これは、1 つ以上のステップを繰り返す必要があるエラーが発生したときに便利です。 **ABORT** オプションを指定して **RESTORE DATABASE** を発行した後、**REDIRECT** オプションを指定した **RESTORE DATABASE** を含む、リダイレクトしたリストア操作の各ステップを繰り返す必要があります。
- 完了する前に増分リストア操作を終了します。

**USER** *username*

データベースへの接続を試みる際に使用するユーザー名を指定します。

**USING** *password*

ユーザー名を認証するために使用するパスワード。パスワードを省略すると、ユーザーに入力を求めるプロンプトが出ます。

**REBUILD WITH ALL TABLESPACES IN DATABASE**

イメージをリストアする時点でデータベースが認識しているすべての表スペースを使って、データベースをリストアします。データベースが既に存在する場合、このリストアによってそれが上書きされます。

**REBUILD WITH ALL TABLESPACES IN DATABASE EXCEPT**

*rebuild-tablespace-clause*

イメージをリストアする時点でデータベースが認識しているすべての表スペースを使って、データベースをリストアします。ただし、リストで指定されているものは除外されます。データベースが既に存在する場合、このリストアによってそれが上書きされます。

**REBUILD WITH ALL TABLESPACES IN IMAGE**

リストアされるイメージに含まれる表スペースだけを使ってデータベースをリストアします。データベースが既に存在する場合、このリストアによってそれが上書きされます。

## RESTORE DATABASE

### REBUILD WITH ALL TABLESPACES IN IMAGE EXCEPT

#### *rebuild-tablespace-clause*

リストアされるイメージに含まれる表スペースだけを使ってデータベースをリストアします。ただし、リストで指定されているものは除外されます。データベースが既に存在する場合、このリストアによってそれが上書きされません。

### REBUILD WITH *rebuild-tablespace-clause*

指定された表スペースのリストだけを使ってデータベースをリストアします。データベースが既に存在する場合、このリストアによってそれが上書きされます。

### TABLESPACE *tablespace-name*

リストアされる表スペースを指定するときに使用する名前のリストです。

**TRANSPORT** オプションを指定する場合、表スペース名は必須です。

### SCHEMA *schema-name*

リストアされるスキーマを指定するために使用される名前のリスト。

**TRANSPORT** オプションを指定する場合、スキーマ名は必須です。 **SCHEMA** オプションは、**TRANSPORT** オプションが指定された場合のみ有効です。

### ONLINE

このキーワードは、表スペース・レベルのリストア操作を行う場合のみ適用でき、これを指定するとオンラインでバックアップ・イメージがリストアできます。これは、他のエージェントが、バックアップ・イメージのリストア中にデータベースに接続できることや、指定された表スペースのリストア中に他の表スペースのデータを使用できることを意味します。

### HISTORY FILE

このキーワードは、バックアップ・イメージから履歴ファイルのみをリストアする場合に指定します。

### COMPRESSION LIBRARY

このキーワードは、バックアップ・イメージから圧縮ライブラリーだけをリストアする場合に指定します。バックアップ・イメージの中にオブジェクトが存在している場合、それはデータベース・ディレクトリーの中にリストアされます。バックアップ・イメージの中にオブジェクトが存在しない場合、リストア操作は失敗します。

**LOGS** このキーワードは、バックアップ・イメージに含まれている一連のログ・ファイルだけをリストアする場合に指定します。バックアップ・イメージの中にログ・ファイルが含まれていない場合、リストア操作は失敗します。このオプションを指定する場合は、**LOGTARGET** オプションも指定する必要があります。

### INCREMENTAL

**INCREMENTAL** は、追加のパラメーターを使用しないで手動累積リストア操作を指定します。手動リストアの際、ユーザーはリストアに含まれるイメージごとに各リストア・コマンドを手動で発行する必要があります。以下の順序でこれを行ってください。最後、1 番目、2 番目、以下同様に最後のイメージまで。

**INCREMENTAL AUTOMATIC/AUTO**

自動累積リストア操作を指定します。

**INCREMENTAL ABORT**

手動累積リストア操作を指定します。

**USE**

**TSM** ターゲット・デバイスとして Tivoli Storage Manager (TSM) を使用して、データベースがリストアされるように指定します。

**XBSA** XBSA インターフェースを使用するように指定します。バックアップ・サービス API (XBSA) は、バックアップやアーカイブの目的でデータ・ストレージ管理を必要とするアプリケーションまたは機能のための、オープン・アプリケーション・プログラミング・インターフェースです。

**SNAPSHOT**

データがスナップショット・バックアップからリストアされるように指定します。

**SNAPSHOT** パラメーターを以下のいずれかのパラメーターと一緒に使用することはできません。

- **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**

また、**SNAPSHOT** パラメーターは、表スペース・リストが関係するリストア操作で使用することはできません。これには、**REBUILD WITH** オプションが含まれます。

スナップショット・バックアップ・イメージからデータをリストアするときのデフォルト動作は、すべてのコンテナ、ローカル・ボリューム・ディレクトリー、およびデータベース・パス (DBPATH) を含む、データベースを構成するすべてのパスの完全データベース・オフライン・リストアです。**LOGTARGET INCLUDE** パラメーターを指定した場合を除き、ログはスナップショット・リストアから除

## RESTORE DATABASE

外されます。**LOGTARGET EXCLUDE** パラメーターが、すべてのスナップショット・リストアのデフォルトです。タイム・スタンプを指定すると、そのタイム・スタンプの付いたスナップショット・バックアップ・イメージがリストアに使用されます。

### **LIBRARY** *library-name*

IBM Data Server には、以下のストレージ・ハードウェアのための DB2 ACS API ドライバーが組み込まれています。

- IBM TotalStorage SAN ボリューム・コントローラー
- IBM Enterprise Storage Server Model 800
- IBM Storwize V7000
- IBM System Storage DS6000
- IBM System Storage DS8000
- IBM System Storage N Series
- IBM XIV

他のストレージ・ハードウェアを使用していて、そのストレージ・ハードウェア用の DB2 ACS API ドライバーがある場合、**LIBRARY** パラメーターを使用してその DB2 ACS API ドライバーを指定できます。

**LIBRARY** パラメーターの値は、完全修飾ライブラリー・ファイル名です。

## OPTIONS

### **"options-string"**

リストア操作で使用するオプションを指定します。ストリングは、二重引用符なしで、入力されたとおりに渡されます。

### **@file-name**

リストア操作で使用するオプションが、DB2 サーバー上のファイルに含まれていることを指定します。このストリングは、ベンダー・サポートのライブラリーに渡されます。ファイル名は完全修飾ファイル名でなければなりません。

**VENDOROPT** データベース構成パラメーターを使用してスナップショット・リストア操作でのベンダー固有のオプションを指定することはできません。代わりに、リストア・ユーティリティーの **OPTIONS** パラメーターを使用する必要があります。

## **OPEN** *num-sessions* **SESSIONS**

TSM またはベンダー製品とともに使用する入出力セッションの数を指定します。

## **FROM** *directory/device*

バックアップ・イメージがあるディレクトリーまたは装置の完全修飾パス名。 **USE TSM**、**FROM**、および **LOAD** を省略した場合のデフォルト値は、クライアント・マシンの現行作業ディレクトリーです。このターゲット・ディレクトリーまたは装置は、ターゲット・サーバー/インスタンス上に存在している必要があります。

複数の項目が指定され、項目の最後がテープ装置である場合には、他のテープが要求されます。有効な応答オプションは、次のとおりです。

- c** 続行。警告メッセージを生成した装置を使用し続けます (例えば、新しいテープがマウントされた場合)。
- d** 装置の終了。警告メッセージの原因となった装置の使用だけを停止します (例えば、もうテープがない場合に停止する、など)。
- t** 終了。ユーザーが、ユーティリティーによって要求された何らかのアクションを実行しなかった場合、リストア操作を異常終了します。

#### **LOAD** *shared-library*

使用するバックアップおよびリストア I/O ベンダー関数を含む共有ライブラリー (Windows オペレーティング・システムでは DLL) の名前。名前には絶対パスを含めることができます。絶対パスを指定しない場合、ユーザー出口プログラムが置かれているパスがデフォルト値として使われます。

#### **TAKEN AT** *date-time*

データベース・バックアップ・イメージのタイム・スタンプです。タイム・スタンプはバックアップ操作が正常に終了した後に表示され、バックアップ・イメージのパス名の一部になっています。 *yyyymmddhhmmss* の形式で指定されます。タイム・スタンプを部分的に指定することもできます。例えば、2 つの異なるタイム・スタンプ 20021001010101 および 20021002010101 で指定されるバックアップ・イメージが存在する場合、20021002 を指定することで、タイム・スタンプ 20021002010101 のイメージが使用できます。このパラメーターに値を指定しない場合は、ソース・メディア上のバックアップ・イメージは 1 つだけでなければなりません。

#### **TO** *target-directory*

このパラメーターは、ターゲット・データベース・ディレクトリーを指定します。ユーティリティーが存在するデータベースへリストアしている場合には、このパラメーターは無視されます。指定するドライブおよびディレクトリーは、ローカルのものでなければなりません。自動ストレージが有効になったデータベースがバックアップ・イメージに含まれる場合、データベース・ディレクトリーだけが変更され、そのデータベースに関連したストレージ・パスは変更されません。

#### **DBPATH ON** *target-directory*

このパラメーターは、ターゲット・データベース・ディレクトリーを指定します。ユーティリティーが存在するデータベースへリストアしている場合には、このパラメーターは無視されます。指定するドライブおよびディレクトリーは、ローカルのものでなければなりません。自動ストレージが有効になったデータベースがバックアップ・イメージに含まれ、**ON** パラメーターが指定されない場合、このパラメーターは **TO** パラメーターと同じ意味になり、データベース・ディレクトリーだけが変更されます。そのデータベースに関連したストレージ・パスは変更されません。

#### **ON** *path-list*

このパラメーターは、データベースに関連付けられているストレージ・パスを再定義します。データベースに複数のストレージ・グループが含まれている場合、このオプションは、定義された各ストレージ・グループがその新し



いストレージ・グループ・パスとして *path-list* を使用するように、すべてのストレージ・グループを指定のパスにリダイレクトします。ストレージ・グループが定義されていないデータベースや、自動ストレージが有効になっていないデータベースに対してこのパラメーターを使用した場合、エラー (SQL20321N) が発生します。バックアップ・イメージ内に定義された既存のストレージ・パスはもはや使用されなくなり、自動ストレージ表スペースは新しいパスに自動的にリダイレクトされます。自動ストレージ・データベースに対してこのパラメーターを指定しない場合、ストレージ・パスはバックアップ・イメージ内に定義されたままの状態になります。1つのパス、またはコンマで区切った複数のパスを指定できます。それぞれのパスは絶対パス名でなければならず、ローカルに存在しなければなりません。

このオプションを **REDIRECT** オプションと一緒に指定した場合、最初の **RESTORE ... REDIRECT** コマンドが呼び出し元に戻るまでこのオプションは有効です。ただし、**SET STOGROUP PATHS** ステートメントまたは **SET TABLESPACE CONTAINERS** ステートメントが発行されると無効になります。その後ストレージ・グループ・パスがリダイレクトされた場合には、そうした変更によって、最初の **RESTORE ... ON path-list** コマンドで指定されたあらゆるパスはオーバーライドされます。

リストア操作中に再定義されたパスを持つストレージ・グループについては、それ以降のロールフォワード操作においてストレージ・パス関連の操作が再生されることはありません。

データベースがディスクにまだ存在せず、**DBPATH ON** パラメーターが指定されていない場合には、最初のパスがターゲット・データベース・ディレクトリーとして使用されます。

複数パーティション・データベースの場合、「**ON path-list**」オプションを指定できるのはカタログ・パーティションについてだけです。**ON** オプションを使用する場合、カタログ・パーティションは、他のどのパーティションがリストアされるよりも前にリストアする必要があります。新しいストレージ・パスでカタログ・パーティションをリストアすると、非カタログ・データベース・パーティションのすべてが **RESTORE\_PENDING** 状態になります。その場合、非カタログ・データベース・パーティションは、**RESTORE** コマンドに **ON** 文節を指定することなく並列してリストアできます。

一般的に、複数パーティション・データベースでは、どのパーティションにも同じストレージ・パスを使用する必要があり、それらはすべて、**RESTORE DATABASE** コマンドの実行前に存在している必要があります。その例外の1つとして、ストレージ・パス内でデータベース・パーティション式を使用する場合があります。その使用によって、処理結果のパス名が各パーティションごとに異なるように、データベース・パーティション番号をストレージ・パスにおいて反映することができます。

**RESTORE** コマンドに **ON** 節を指定して使用すると、リダイレクト・リストア操作と同じ意味になります。

**ON** パラメーターを使用してスキーマ転送のストレージ・パスを再定義することはできません。スキーマ転送は、ターゲット・データベースの既存のストレージ・パスを使用します。

**INTO** *target-database-alias*

ターゲット・データベースの別名です。ターゲット・データベースが存在しない場合には、作成されます。

データベース・バックアップを既存のデータベースにリストアするとき、リストアされたデータベースは既存のデータベースの別名およびデータベース名を継承します。データベース・バックアップが存在していないデータベースにリストアするとき、新規のデータベースが指定した別名およびデータベース名を使用して作成されます。新しいデータベース名は、リストア先のシステムで固有のものでなければなりません。

**TRANSPORT INTO** *target-database-alias*

転送操作で使用する既存のターゲット・データベース別名を指定します。転送される表スペースとスキーマがデータベースに追加されます。

**TABSPACE** および **SCHEMA** オプションが、有効な転送可能セットを表す表スペース名とスキーマ名を指定していなければなりません。そうでない場合は、転送操作は失敗します (SQLCODE=SQL2590N rc=1)。

システム・カタログは転送できません。(SQLCODE=SQL2590N rc=4。)

**RESTORE** コマンドによってスキーマの妥当性検査が行われた後、転送される表スペース内のオブジェクトを表すシステム・カタログ項目が、ターゲット・データベースに作成されます。スキーマ再作成の完了後、ターゲット・データベースは物理表スペース・コンテナの所有権を得ます。

リストアされる表スペースに含まれる物理オブジェクトと論理オブジェクトがターゲット・データベースに再作成され、表スペースの定義とコンテナがターゲット・データベースに追加されます。オブジェクト作成時に障害が発生した場合、または DDL の再生が発生した場合は、エラーが返されません。

**STAGE IN** *staging-database*

転送操作のソースであるバックアップ・イメージのための一時ステージング・データベースの名前を指定します。 **STAGE IN** オプションを指定した場合、転送操作が完了しても一時データベースはドロップされません。このデータベースは、転送が完了したら必要ではなくなるので、DBA がドロップできます。

**STAGE IN** オプションを指定しない場合は、以下のようになります。

- データベース名は、SYSTGxxx の形式になります (xxx は整数値)。
- 転送操作完了後、一時ステージング・データベースはドロップされます。

**USING STOGROUP** *storagegroup-name*

自動ストレージ表スペースの場合、すべての表スペースに関連付けられるターゲット・ストレージ・グループが転送されることを指定します。ストレージ・グループが指定されない場合には、ターゲット・データベースの、現在指定されているデフォルトのストレージ・グループが使用されます。この節は自動ストレージ表スペースのみに適用され、スキーマの transport 操作中のみ有効です。

## RESTORE DATABASE

表スペース・データが格納されるストレージ・グループを指定します。  
*storagegroup-name* には、**TRANSPORT** 操作の *target-database-alias* に存在するストレージ・グループを指定する必要があります (SQLSTATE 42704)。これは、1 部構成の名前です。

### LOGTARGET *directory*

スナップショット以外のリストアの場合:

バックアップ・イメージからログ・ファイルを抽出する際のターゲット・ディレクトリーとして使用する、データベース・サーバー上の既存のディレクトリーの絶対パス名。このオプションを指定する場合、バックアップ・イメージ内のログ・ファイルは、そのターゲット・ディレクトリー内に抽出されます。このオプションを指定しない場合、バックアップ・イメージ内のログ・ファイルは抽出されません。バックアップ・イメージからログ・ファイルだけを抽出する場合は、**LOGS** オプションを指定してください。このオプションは、データベース・パーティション番号とログ・ストリーム ID を自動的にパスに付加します。

### DEFAULT

ログ・ファイルをバックアップ・イメージからデータベースのデフォルトのログ・ディレクトリー (/home/db2user/db2inst/NODE0000/SQL00001/LOGSTREAM0000 など) にリストアします。

スナップショット・リストアの場合:

### INCLUDE

スナップショット・イメージからログ・ディレクトリー・ボリュームをリストアします。このオプションが指定されていて、バックアップ・イメージにログ・ディレクトリーが含まれている場合、それらはリストアされます。ディスク上に既存のログ・ディレクトリーとログ・ファイルは、バックアップ・イメージ中のログ・ディレクトリーと競合するものでなければ、変更なしでそのままになります。ディスク上に既存のログ・ディレクトリーがバックアップ・イメージ中のログ・ディレクトリーと競合する場合は、エラーが戻されます。

### EXCLUDE

ログ・ディレクトリー・ボリュームをリストアしません。このオプションを指定すると、バックアップ・イメージからログ・ディレクトリーはリストアされません。ディスク上に既存のログ・ディレクトリーとログ・ファイルは、バックアップ・イメージ中のログ・ディレクトリーと競合するものでなければ、変更なしでそのままになります。データベースに属する 1 つのパスがリストアされ、そのために暗黙のうちに 1 つのログ・ディレクトリーがリストアされ、その結果、ログ・ディレクトリーが上書きされることになる場合、エラーが戻されます。

### FORCE

スナップショット・イメージをリストアする時に現行データベースの既存のログ・ディレクトリーを上書きおよび置換することを許可します。このオプションを使用しなければ、スナップショット・イメージのログ・ディレクトリーと矛盾するディスク上の既存のログ・ディレクトリーおよびログ・ファイルが原因で、リストアは失

敗します。このオプションを使用して、リストアでこれらの既存のログ・ディレクトリーを上書きおよび置換できるように指示します。

**注:** このオプションは注意して使用し、リカバリーに必要な可能性があるすべてのログを常にバックアップおよびアーカイブしてください。

スナップショット・リストアでは、ディレクトリー・オプションのデフォルト値は **LOGTARGET EXCLUDE** です。

#### **NEWLOGPATH** *directory*

リストア操作後にアクティブ・ログ・ファイルに使用されるディレクトリーの絶対パス名。このパラメーターの機能は **newlogpath** データベース構成パラメーターと同じです。このパラメーターは、バックアップ・イメージのログ・パスが、リストア後の使用に適していない場合に使用することができます。例えば、パスがもはや有効でない、または別のデータベースによって使用されている、という場合などです。

**注:** **newlogpath** コマンド・パラメーターが設定されると、**newlogpath** データベース構成パラメーターを更新する場合と同様に、**logpath** の値にはノード番号が自動的に追加されなくなります。詳細については、『**newlogpath** - データベース・ログ・パスの変更』を参照してください。

#### **DEFAULT**

リストアが完了した後、データベースはデフォルトのログ・ディレクトリー `/home/db2user/db2inst/NODE0000/SQL00001/LOGSTREAM0000` をロギングに使用します。

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

使用するバッファの数です。値を明示的に指定しない場合、DB2 データベース・システムはこのパラメーターの最適値を自動的に選択します。複数のソースが読み取られる場合や、**PARALLELISM** の値が増やされている場合は、パフォーマンスを向上させるために複数のバッファを使用することができます。

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

リストア操作に使用するバッファのサイズ (ページ数)。値を明示的に指定しない場合、DB2 データベース・システムはこのパラメーターの最適値を自動的に選択します。このパラメーターの最小値は 8 ページです。

リストア・バッファ・サイズは、バックアップ操作中に指定したバックアップ・バッファ・サイズに正の整数を乗算したサイズでなければなりません。誤ったバッファ・サイズを指定すると、許容可能な最小のサイズで割り振られます。

#### **REPLACE HISTORY FILE**

リストア操作において、ディスク上の履歴ファイルを、バックアップ・イメージの履歴ファイルで置換することを指定します。

#### **REPLACE EXISTING**

ターゲット・データベースの別名と同じ別名を持つデータベースが既に存在している場合、このパラメーターは、リストア・ユーティリティーが既存の

## RESTORE DATABASE

データベースをリストアしたデータベースに置換するように指定します。これはリストア・ユーティリティーを起動するスクリプトで便利です。コマンド行プロセッサは、ユーザーに既存のデータベースの削除を検証するよう求めるプロンプトを出さないためです。 **WITHOUT PROMPTING** パラメーターが指定された場合、**REPLACE EXISTING** を指定する必要はありませんが、その場合、ユーザー介入を標準的に必要とするイベントが起こるとこの操作は失敗します。

### REDIRECT

リダイレクトしたリストア操作を指定します。リダイレクトしたリストア操作を完了するには、このコマンドの後に 1 つ以上の **SET TABLESPACE CONTAINERS** コマンドまたは **SET STOGROUP PATHS** コマンドを続け、次に **CONTINUE** オプションを指定して **RESTORE DATABASE** コマンドを続ける必要があります。以下に例を示します。

```
RESTORE DB SAMPLE REDIRECT
```

```
SET STOGROUP PATHS FOR sg_hot ON '/ssd/fs1', '/ssd/fs2'  
SET STOGROUP PATHS FOR sg_cold ON '/hdd/path1', '/hdd/path2'
```

```
RESTORE DB SAMPLE CONTINUE
```

バックアップ・イメージの作成以降にストレージ・グループが名前変更された場合には、**SET STOGROUP PATHS** コマンドに指定されるストレージ・グループ名は、最新の名前ではなくバックアップ・イメージからのストレージ・グループ名を参照します。

同一のリダイレクトしたリストア操作に関連したコマンドはすべて、同じウィンドウまたは CLP セッションから起動しなければなりません。

### GENERATE SCRIPT *script*

指定されたファイル名を使用して、リダイレクト・リストア・スクリプトを作成します。スクリプト名は相対パスまたは絶対パスであり、そのスクリプトはクライアント・サイドで生成されます。クライアント・サイドでそのファイルを作成できない場合には、エラー・メッセージ (SQL9304N) が戻されます。ファイルが既に存在する場合は上書きされます。使用法については詳しくは、下記の例を参照してください。

### WITHOUT ROLLING FORWARD

データベースを、正常にリストアされた後ロールフォワード・ペンディング状態にしないように指定します。

正常なリストアに続いて、データベースがロールフォワード・ペンディング状態にある場合には、データベースが使用できるようになる前に、

**ROLLFORWARD** コマンドを起動する必要があります。

オンライン・バックアップ・イメージからのリストアでこのオプションを指定した場合、エラー SQL2537N が戻されます。

注: リカバリー可能データベースのバックアップ・イメージである場合、**REBUILD** オプションに **WITHOUT ROLLING FORWARD** を指定することはできません。

### PARALLELISM *n*

リストア操作中に作成されるバッファ・マニピュレーターの数を指定しま



す。値を明示的に指定しない場合、DB2 データベース・システムはこのパラメーターの最適値を自動的に選択します。

#### COMPRLIB *name*

解凍を実行するために使用するライブラリーの名前を指定します (例えば、Windows の場合は db2compr.dll、Linux または UNIX システムの場合は libdb2compr.so)。この名前は、サーバー上の 1 個のファイルを参照する完全修飾パスでなければなりません。このパラメーターを指定しない場合、DB2 データベース・システムはイメージ内に格納されているライブラリーの使用を試みます。バックアップが圧縮されていなかった場合、このパラメーターの値は無視されます。指定されたライブラリーをロードできない場合、リストア操作は失敗します。

#### COMPROPTS *string*

バイナリー・データのうち、解凍ライブラリーの初期設定ルーチンに渡すブロックを記述します。DB2 データベース・システムはこのストリングをクライアントからサーバーに直接渡すため、バイト反転やコード・ページ変換の問題がある場合は、解凍ライブラリーで処理されます。データ・ブロックの最初の文字が「@」なら、データの残りの部分は、サーバー上に存在するファイルの名前を指定するものとして、DB2 データベース・システムは解釈します。その場合、DB2 データベース・システムは *string* の内容をこのファイルの内容で置き換え、新しい値を初期設定ルーチンに渡します。ストリングの最大長は 1 024 バイトです。

#### WITHOUT PROMPTING

リストア操作を無人で実行するように指定します。通常はユーザー介入を必要とするアクションでは、エラー・メッセージが戻されます。テープやディスクセットなどの取り外し可能メディア装置を使用している場合、このオプションを指定していても、その装置が終わるとプロンプトが出されます。

#### 例

- 以下の例で、データベース WSDb は 0 から 3 までの番号が付けられた 4 つのデータベース・パーティションすべてに定義されています。パス /dev3/backup はすべてのデータベース・パーティションからアクセスできます。以下のオフライン・バックアップ・イメージは、/dev3/backup から入手可能です。

```
wsdb.0.db2inst1.DBPART000.200802241234.001
wsdb.0.db2inst1.DBPART001.200802241234.001
wsdb.0.db2inst1.DBPART002.200802241234.001
wsdb.0.db2inst1.DBPART003.200802241234.001
```

最初にカタログ・パーティションをリストアしてから WSDb データベースの他のすべてのデータベース・パーティションを /dev3/backup ディレクトリーからリストアするには、いずれかのデータベース・パーティションから以下のコマンドを実行します。

```
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
```

## RESTORE DATABASE

```
INTO wsdb REPLACE EXISTING'  
db2_a11 '<<+3< db2 RESTORE DATABASE wsdb FROM /dev3/backup  
TAKEN AT 20020331235235  
INTO wsdb REPLACE EXISTING'
```

**db2\_a11** ユーティリティーは、指定された各データベース・パーティションへのリストア・コマンドを出します。 **db2\_a11** を使用してリストアを実行する場合は、常に **REPLACE EXISTING** や **WITHOUT PROMPTING** を指定してください。これを指定しないと、プロンプトが表示された場合に操作がハングしたように見えます。それは、**db2\_a11** でユーザー・プロンプトがサポートされていないためです。

2. 以下は、別名が **MYDB** であるデータベースの典型的なリダイレクト・リストアのシナリオです。

- a. 次のように、**REDIRECT** オプションを指定して **RESTORE DATABASE** コマンドを発行する。

```
restore db mydb replace existing redirect
```

ステップ 1 が正常終了した後でステップ 3 が完了する前に、次を発行してリストア操作を打ち切ることができる。

```
restore db mydb abort
```

- b. 再定義する必要があるコンテナを持つ表スペースごとに、**SET TABLESPACE CONTAINERS** コマンドを発行する。以下に例を示します。

```
set tablespace containers for 5 using  
(file 'f:¥ts3con1' 20000, file 'f:¥ts3con2' 20000)
```

リストアしたデータベースのコンテナが、このステップで指定したものであることを検査するために、**LIST TABLESPACE CONTAINERS** コマンドを発行する。

- c. ステップ 1 および 2 が正常終了した後、次を発行する。

```
restore db mydb continue
```

これはリダイレクト・リストア操作の最終ステップです。

- d. ステップ 3 が失敗した場合、またはリストア操作を打ち切った場合、リダイレクト・リストアはステップ 1 から再始動できる。

3. 以下の例は、リカバリー可能データベース用の週次の増分バックアップ・ストラテジーのサンプルです。週 1 回の全データベース・バックアップ操作、1 日 1 回の非累積 (差分) バックアップ操作、および週 2 回の累積 (増分) バックアップ操作が含まれています。

```
(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
```

金曜日の午前中に作成されたイメージを自動データベース・リストアするには、次のようにします。

```
restore db mydb incremental automatic taken at (Fri)
```



金曜日の午前中に作成されたイメージを手動データベース・リストアするには、次のようにします。

```
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. リモート・サイトに移動することを意図したバックアップ・イメージを作成し、それにログを含めるには、次のようにします。

```
backup db sample online to /dev3/backup include logs
```

このバックアップ・イメージをリストアするには、**LOGTARGET** パスを指定し、**ROLLFORWARD** でそのパスを指定します。

```
restore db sample from /dev3/backup logtarget /dev3/logs
rollforward db sample to end of logs and stop overflow log path /dev3/logs
```

5. ログを含むバックアップ・イメージから、ログ・ファイルだけを取り出すには、

```
restore db sample logs from /dev3/backup logtarget /dev3/logs
```

6. 次の例では、データベース **SAMPLE** のバックアップ操作のために、同一のターゲット・ディレクトリーを 3 回指定しています。データは 3 つのターゲット・ディレクトリーに並行してバックアップされ、それら 3 つのバックアップ・イメージは拡張子 **.001**、**.002**、および **.003** が付けられて生成されます。

```
backup db sample to /dev3/backup, /dev3/backup, /dev3/backup
```

ターゲット・ディレクトリーからバックアップ・イメージをリストアするには、次を発行します。

```
restore db sample from /dev3/backup, /dev3/backup, /dev3/backup
```

7. リストア操作で使用する **TSM** 情報を指定するには、**USE TSM OPTIONS** キーワードを使用します。Windows プラットフォームでは、**-fromowner** オプションを指定しないでください。

- 区切り文字付きストリングを指定する場合、

```
restore db sample use TSM options '"-fromnode=bar -fromowner=dmcinnis"'
```

- 完全修飾ファイル名を指定する場合、

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

ファイル **myoptions.txt** には、**-fromnode=bar -fromowner=dmcinnis** というストリングが含まれています。

8. 以下に示すのは、新しいストレージ・パスによる、複数パーティション自動ストレージ対応データベースの簡単なリストアです。もともとこのデータベースは、1 つのストレージ・パス **/myPath0:** を使用して作成されたものです。

- カタログ・パーティションで、次のコマンドを発行します。**restore db mydb on /myPath1,/myPath2**

- カタログでないすべてのパーティションで、次のコマンドを発行します。

```
restore db mydb
```

9. 非自動ストレージ・データベースにおいて以下のコマンドを発行すると、そのスクリプト出力は、

## RESTORE DATABASE

restore db sample from /home/jseifert/backups taken at 20050301100417 redirect  
generate script SAMPLE\_NODE0000.clp

下記のようなものになります。

```
-- *****
-- ** 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/SAMPLE/NODE0000/LOGSTREAM0000/'
-- 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
```

## RESTORE DATABASE

```

-- ** 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
-- *****

```

10. 自動ストレージ・データベースにおいて以下のコマンドを発行すると、そのスクリプト出力は、

```

restore db test from /home/jseifert/backups taken at 20050304090733 redirect
generate script TEST_NODE0000.clp

```

下記のようなものになります。

## RESTORE DATABASE

```
-- *****
-- ** automatically created redirect restore script
-- *****
UPDATE COMMAND OPTIONS USING S ON Z ON TEST_NODE0000.out V ON;
SET CLIENT ATTACH_MEMBER 0;
SET CLIENT CONNECT_MEMBER 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/TEST/NODE0000/LOGSTREAM0000/'
-- WITH <num-buff> BUFFERS
-- BUFFER <buffer-size>
-- REPLACE HISTORY FILE
-- REPLACE EXISTING
REDIRECT
-- PARALLELISM <n>
-- WITHOUT ROLLING FORWARD
-- WITHOUT PROMPTING
;
-- *****
-- ** storage group definition
-- ** Default storage group ID          = 0
-- ** Number of storage groups          = 3
-- *****
-- ** Storage group name                 = SG_DEFAULT
-- ** Storage group ID                   = 0
-- ** Data tag                            = None
-- *****
-- SET STOGROUP PATHS FOR SG_DEFAULT
-- ON '/hdd/path1'
-- , '/hdd/path2'
-- ;
-- *****
-- ** Storage group name                 = SG_HOT
-- ** Storage group ID                   = 1
-- ** Data tag                            = 1
-- *****
-- SET STOGROUP PATHS FOR SG_HOT
-- ON '/ssd/fs1'
-- , '/ssd/fs2'
-- ;
-- *****
-- ** Storage group name                 = SG_COLD
-- ** Storage group ID                   = 2
-- ** Data tag                            = 9
-- *****
-- SET STOGROUP PATHS FOR SG_COLD
-- ON '/hdd/slowpath1'
-- ;
-- *****
-- ** 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
-- ** Storage group ID                   = 0
-- ** Source storage group ID            = -1
-- ** Data tag                            = None
-- ** Auto-resize enabled                 = Yes
-- ** Total number of pages              = 6144
-- ** Number of usable pages             = 6140
```

# RESTORE DATABASE

```

-- ** 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
-- ** Storage group ID = 1
-- ** Source storage group ID = -1
-- ** Data tag = 1
-- ** 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
-- ** Storage group ID = 2
-- ** Source storage group ID = -1
-- ** Data tag = 9
-- ** 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;

```

## RESTORE DATABASE

```
-- *****  
-- ** end of file  
-- *****
```

11. **SNAPSHOT** オプションを使用した **RESTORE DB** コマンドの例を以下に示します。

ログ・ディレクトリー・ボリュームをスナップショット・イメージからリストアし、プロンプトを出しません。

```
db2 restore db sample use snapshot LOGTARGET INCLUDE without prompting
```

ログ・ディレクトリー・ボリュームをリストアせず、プロンプトを出しません。

```
db2 restore db sample use snapshot LOGTARGET EXCLUDE without prompting
```

ログ・ディレクトリー・ボリュームをリストアせず、プロンプトを出しません。**LOGTARGET** が指定されていない場合には、デフォルトは **LOGTARGET EXCLUDE** です。

```
db2 restore db sample use snapshot without prompting
```

矛盾するログ・ディレクトリーが含まれるスナップショット・イメージをリストアする時に、プロンプトを出さずに現行データベースの既存のログ・ディレクトリーを上書きおよび置換することを許可します。

```
db2 restore db sample use snapshot LOGTARGET EXCLUDE FORCE without prompting
```

矛盾するログ・ディレクトリーが含まれるスナップショット・イメージをリストアする時に、プロンプトを出さずに現行データベースの既存のログ・ディレクトリーを上書きおよび置換することを許可します。

```
db2 restore db sample use snapshot LOGTARGET INCLUDE FORCE without prompting
```

12. **RESTORE** コマンドに **TRANSPORT REDIRECT** オプションを指定した転送操作の例を以下に示します。

ソース・データベース (TT\_SRC) のバックアップ・イメージのストレージ・パスが /src で、ターゲット・データベース (TT\_TGT) のストレージ・パスが /tgt と想定しています。

```
> RESTORE DB TT_SRC TABLESPACE (AS1) SCHEMA (KRODGER) TRANSPORT INTO TT_TGT REDIRECT
```

```
SQL1277W A redirected restore operation is being performed. Table space configuration can now be viewed and table spaces that do not use automatic storage can have their containers reconfigured.  
DB20000I The RESTORE DATABASE command completed successfully.
```

表スペース「AS1」が、/tgt/krodger/NODE0000/TT\_TGT/T0000003/C0000000.LRG のようなコンテナ・パスに転送されます。

転送する表スペースのターゲット・ストレージ・グループを指定するには、**RESTORE** コマンドの **USING STOGROUP** オプションを使用できます。以下の例では、表スペース TS1 および TS2 の両方が SG\_COLD ストレージ・グループにリストアされます。

```
RESTORE DB TT_SRC TABLESPACE (TS1, TS2) SCHEMA (KRODGER) TRANSPORT INTO TT_TGT USING STOGROUP SG_COLD
```

**注:** **RESTORE** コマンドの **USING STOGROUP** オプションが有効であるのは、転送操作のみであるため、その他のリストア操作中にターゲット・ストレージ・グループを指定するために使用することはできません。

ターゲット・データベースのデフォルト・ストレージ・グループへの転送を実行する場合、 **USING STOGROUP** オプションを指定する必要はありません。

```
RESTORE DB TT_SRC TABLESPACE (TS3) SCHEMA (KRODGER) TRANSPORT INTO TT_TGT
```

**TRANSPORT** 操作中に **RESTORE** コマンドで指定するストレージ・グループ名は、ターゲット・データベースに現在定義されているものでなければなりません。バックアップ・イメージやソース・データベースに定義されていなくても構いません。

ドロップされたデータベース・パーティションを再作成する (損傷したため) 目的でパラメーター **AT DBPARTITIONNUM** を使用した場合、このデータベース・パーティションのデータベースはリストア・ペンディング状態になります。データベース・パーティションを再作成した後、このデータベース・パーティション上のデータベースをただちにリストアする必要があります。

## 使用上の注意

- `db2 restore db name` という形式の **RESTORE DATABASE** コマンドは、データベース・イメージを使ったフル・データベース・リストアの実行と、表スペース・イメージ内に検出される表スペースの表スペース・リストア操作を実行します。  
`db2 restore db name tablespace` という形式の **RESTORE DATABASE** コマンドは、イメージ内に検出される表スペースの表スペース・リストアを実行します。さらに、そのようなコマンドで表スペースのリストが指定された場合、明示的にリストされるすべての表スペースがリストアされます。
- オンライン・バックアップのリストア操作を実行した後、ロールフォワード・リカバリーを実行する必要があります。
- プロキシ・ノードをサポートする TSM 環境では、**OPTIONS** パラメーターを使用してリストア操作を使用可能にすることができます。詳しくは、『Tivoli Storage Manager クライアントの構成』のトピックを参照してください。
- バックアップ・イメージが圧縮されているなら、DB2 データベース・システムはそのことを検出し、データはリストア前に自動的に解凍されます。 `db2Restore API` でライブラリーが指定されている場合、データの解凍にはそれが使用されます。そうでない場合、ライブラリーがバックアップ・イメージに保管されているかどうか検査され、ライブラリーが存在する場合にはそれが使用されます。最後に、ライブラリーがバックアップ・イメージに入っていない場合には、データを解凍できず、リストア操作が失敗します。
- バックアップ・イメージから圧縮ライブラリーをリストアする場合 (**COMPRESSION LIBRARY** オプションを指定して明示的に、または圧縮バックアップの通常のリストアを実行することにより暗黙的に)、そのリストア操作は、バックアップが作成されたのと同じプラットフォームおよびオペレーティング・システム上で実行する必要があります。バックアップ作成時のプラットフォームとリストア操作実行時のプラットフォームが違っていると、それらの 2 つのシステムの間クロスプラットフォーム・リストアが DB2 データベース・システムで通常にサポートされている場合でも、リストア操作は失敗します。
- バックアップした SMS 表スペースをリストアできるのは、SMS 表スペースに対してのみです。DMS 表スペースにはリストアできませんし、その逆も不可能です。



## RESTORE DATABASE

- ログ・ファイルを含むバックアップ・イメージからログ・ファイルをリストアする場合には、**LOGTARGET** オプションを指定する必要があります。その際、DB2 サーバー上に存在する有効な完全修飾パス名を指定する必要があります。それらの条件が満たされている場合、リストア・ユーティリティーは、イメージ内のログ・ファイルをターゲット・パスに書き込みます。ログを含まないバックアップ・イメージのリストア操作で **LOGTARGET** を指定した場合、表スペース・データのリストアが試行される前にエラーが戻されます。また、**LOGTARGET** に無効なパスや読み取り専用パスが指定された場合も、リストア操作が失敗してエラーが返されます。
- **RESTORE DATABASE** コマンド発行の時点で **LOGTARGET** パス内にログ・ファイルが存在している場合、警告プロンプトがユーザーに対して返されます。 **WITHOUT PROMPTING** が指定されている場合、この警告は戻されません。
- **LOGTARGET** を指定したリストア操作において、いずれかのログ・ファイルを抽出できない場合には、リストア操作が失敗してエラーが戻されます。バックアップ・イメージから抽出されるいずれかのログ・ファイルの名前が、**LOGTARGET** パス内に存在するファイル名と同じである場合には、リストア操作が失敗してエラーが返されます。データベース・リストア・ユーティリティーは、**LOGTARGET** ディレクトリー内の既存のログ・ファイルを上書きしません。
- 保存されているログ・セットだけをバックアップ・イメージからリストアすることも可能です。ログ・ファイルだけをリストアするように指定するには、**LOGTARGET** パスに加えて **LOGS** オプションを指定します。 **LOGTARGET** パスを指定しないで **LOGS** オプションを指定すると、エラーになります。この操作モードでログ・ファイルをリストアしようとして問題が発生した場合、そのリストア操作は即座に終了し、エラーが戻されます。
- 自動増分リストア操作においては、リストア操作のターゲット・イメージに含まれるログ・ファイルだけがバックアップ・イメージから取り出されます。増分リストア処理中に参照される中間イメージに含まれるログ・ファイルは、それらの中間バックアップ・イメージから抽出されません。手動増分リストア操作の場合、**LOGTARGET** パスは、最後に発行するリストア・コマンドにのみ指定してください。
- オフラインの全データベース・バックアップおよびオフラインの増分データベース・バックアップは、より新しいデータベース・バージョンにリストアできますが、オンライン・バックアップはそれができません。複数パーティション・データベースでは、まずカタログ・パーティションを別個にリストアしてから、その後に残りのデータベース・パーティションを（並列または逐次に）リストアする必要があります。ただし、リストア操作によって実行された暗黙的なデータベース・アップグレードは、失敗する可能性があります。複数パーティション・データベースでは、1 つ以上のデータベース・パーティションでそれが失敗することがあります。この場合、**RESTORE DATABASE** コマンドの後にカタログ・パーティションから発行する単一の **UPGRADE DATABASE** コマンドを続けて、データベースを正常にアップグレードすることができます。
- パーティション・データベース環境では、1 つの表スペースが、異なるデータベース・パーティション間で異なるストレージ・グループに関連付けられる可能性があります。リダイレクト・リストアで表スペース・コンテナが **DMS** から自動ストレージに変更されると、その表スペースはデフォルトのストレージ・グループに関連付けられます。異なるデータベース・パーティションに対してリダイレクト・リストアが行われてから次回また行われるまでの間に新しいデフォル

ト・ストレージ・グループが選択された場合には、その表スペースのストレージ・グループの関連付けは、パーティション・データベース環境において矛盾することになります。このような場合、必要であれば、ALTER TABLESPACE ステートメントを使用して、すべてのデータベース・パーティションの表スペースで自動ストレージを使用するように変更し、リバランスを行ってください。

- **TRANSPORT** オプションがサポートされるのは、クライアントとデータベースのコード・ページが同じ場合のみです。

### スナップショット・リストア

従来の (スナップショット以外の) リストアのように、スナップショット・バックアップ・イメージをリストアする時のデフォルトの動作は、ログ・ディレクトリーをリストアしない、**LOGTARGET EXCLUDE** です。

いずれかのログ・ディレクトリーのグループ ID がリストアする他のパスのいずれかと共有されていることが DB2 データベース・マネージャーにより検出された場合、エラーが返されます。この場合、ログ・ディレクトリーがリストアに含まれる必要があるため、**LOGTARGET INCLUDE** または **LOGTARGET INCLUDE FORCE** を指定する必要があります。

DB2 データベース・マネージャーは、バックアップ・イメージからのパスのリストアが行われる前に既存のログ・ディレクトリー (1 次、ミラー、およびオーバーフロー) を保存するために、すべての方法を試みます。

ログ・ディレクトリーをリストアする場合、ディスク上に事前に存在するログ・ディレクトリーがバックアップ・イメージ中のログ・ディレクトリーと競合することが DB2 データベース・マネージャーによって検出されたなら、DB2 データベース・マネージャーによってエラーが報告されます。その場合、**LOGTARGET INCLUDE FORCE** を指定した場合にはこのエラーは抑止され、イメージのログ・ディレクトリーがリストアされて、以前に存在していたログ・ディレクトリーはすべて削除されます。

特殊なケースとして **LOGTARGET EXCLUDE** オプションが指定されていて、ログ・ディレクトリーのパスがデータベース・ディレクトリー (例えば、/NODExxxx/SQLxxxxx/LOGSTREAMxxxxx/) の下にあるという場合があります。この場合は、リストアによりログ・ディレクトリーはデータベース・パスとして上書きされ、その下位にあるすべての内容はリストアされます。このシナリオに該当することが DB2 データベース・マネージャーによって検出された場合、そのログ・ディレクトリー中にログ・ファイルが存在しているなら、エラーが報告されます。**LOGTARGET EXCLUDE FORCE** を指定した場合には、このエラーは抑止され、ディスク上の矛盾するログ・ディレクトリーは、バックアップ・イメージのそれらのログ・ディレクトリーで上書きされます。

### 表スペースとスキーマの転送

表スペースとスキーマの完全なリストを指定する必要があります。

転送時にターゲット・データベースがアクティブでなければなりません。

オンライン・バックアップ・イメージを使用する場合、ステージング・データベースはバックアップの最後までロールフォワードされます。オフライン・バックアップ・イメージを使用する場合、ロールフォワード処理は行われません。

**dftdbpath** データベース・パラメーターで指定したパスの下に、バックアップ・イメージに含まれるシステム・カタログ表スペースから成るステージング・データベースが作成されます。このデータベースは、**RESTORE DATABASE** コマンドが完了した時点でドロップされます。ステージング・データベースは、転送される表スペース内のオブジェクトを再生成するために使用される **DDL** を抽出するのに必要となります。

表スペースの転送時、DB2 データベース・マネージャーはページ・サイズが一致する最初の使用可能バッファ・プールを、転送される表スペースに割り当てようとします。転送される表スペースとページ・サイズが一致するバッファ・プールがターゲット・データベースにない場合は、隠しバッファ・プールが割り当てられることがあります。隠しバッファ・プールとは、転送される表スペース用の一時的なプレースホルダーのことです。転送完了後に、転送された表スペースに割り当てられたバッファ・プールを確認することができます。**ALTER TABLESPACE** コマンドを発行することによって、バッファ・プールを更新できます。

データベースのロールフォワードで表スペース・スキーマ転送ログ・レコードが検出された場合、対応する転送済み表スペースはオフラインになってドロップ・ペンディング状態に変わります。これは、転送された表スペースとその内容を再作成するための転送済み表スペースの完全なログが、データベースにないためです。転送完了後にターゲット・データベースのフルバックアップを取ることができるので、その後のロールフォワードがログ・ストリーム内のスキーマ転送のポイントを通過することはありません。

### ストレージ・グループの転送

転送操作では、ターゲット・データベースに現在定義されているストレージ・グループを変更することはできません。また、転送時に新規ストレージ・グループを明示的に作成することはできません。

転送のデフォルトのターゲット・ストレージ・グループは、操作のターゲット・データベースのデフォルトのストレージ・グループです。転送操作中にリストアするすべての表スペースを、ターゲット・データベースの特定のストレージ・グループに明示的にリダイレクトすることもできます。

転送操作中、**RESTORE** コマンドに **TRANSPORT REDIRECT** オプションを使用して実行すると、自動ストレージ表スペースのデフォルトのストレージ・グループ構成は、バックアップ・イメージで設定されている構成ではなく、ターゲット・データベースのストレージ・グループおよびストレージ・グループ・パスになります。なぜなら、自動ストレージ表スペースは、ターゲット・データベースの定義に従い、既存のストレージ・グループ・パスにリストアして直接リダイレクトする必要があるからです。

## REWIND TAPE

ストリーミング磁気テープ装置へのバックアップおよびリストア操作のためにテープを巻き戻します。このコマンドは Windows オペレーティング・システムでのみサポートされています。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMANT

### 必要な接続

なし

### コマンド構文

```
▶▶ REWIND TAPE [ON device] ◀◀
```

### コマンド・パラメーター

**ON** *device*

有効なテープ装置名を指定します。デフォルト値は ¥\$.¥TAPE0 です。

## ROLLFORWARD DATABASE

ROLLFORWARD DATABASE コマンドは、データベースのログ・ファイルに記録されたトランザクションを適用することによって、データベースをリカバリーします。このコマンドは、データベースまたは表スペースのバックアップ・イメージがリストアされた後、またはメディア・エラーのためにいずれかの表スペースがデータベースによってオフラインにされた場合に実行できます。

データベースをロールフォワード・リカバリーによってリカバリーするためには、事前にデータベースをリカバリー可能にしておく（つまり、データベース構成パラメーター `logarchmeth1` または `logarchmeth2` を OFF 以外の値に設定しておく）必要があります。

### 有効範囲

パーティション・データベース環境では、このコマンドはカタログ・パーティションからのみ呼び出すことができますが、各パーティションがロールフォワード操作に関与します。指定された時点までのデータベースまたは表スペースのロールフォワード操作は、`db2nodes.cfg` ファイルにリストされたすべてのデータベース・パーティションに影響します。ログの末尾までのデータベースまたは表スペースのロールフォワード操作は、指定されたデータベース・パーティションに影響します。データベース・パーティションが指定されていない場合は、`db2nodes.cfg` ファイルにリストされたすべてのデータベース・パーティションに影響します。特定のパーティションにロールフォワード・リカバリーが必要ない場合、そのパーティションは無視されます。

DB2 pureScale 環境の場合、このコマンドは任意のメンバーから実行できます。また、オンラインの表スペース・レベルのロールフォワード操作は、他のメンバーをオンラインにしたまま実行できます。データベース・パーティションのサブセットでロールフォワードを行うようにユーザーが選択できるパーティション・データベース環境とは異なり、DB2 pureScale 環境では、すべてのメンバーからのログが自動的に適用されます。あるメンバーで障害が起こったときに、それに対してロールフォワード操作が進行中である場合には、**ROLLFORWARD** コマンドをどのメンバーからでも再発行することができます。オリジナルのメンバーで障害が起こったときにオフのままにされた地点から、ロールフォワードが再開されます。

パーティション表については、関連する表スペースも同じ時点までロールフォワードする必要があります。この要件は、表のデータ・パーティションが含まれている表スペースに適用されます。ある 1 つの表スペースにパーティション表の一部が含まれている場合も、ログの末尾までのロールフォワードは可能です。

以前の DB2 リリース・バージョンで作成されたログ・ファイルを介してロールフォワードすることはできません。この制限は、DB2 データベースの新規リリース・バージョンにアップグレードする際の重要な考慮事項です。

### 許可

以下の権限のいずれか。

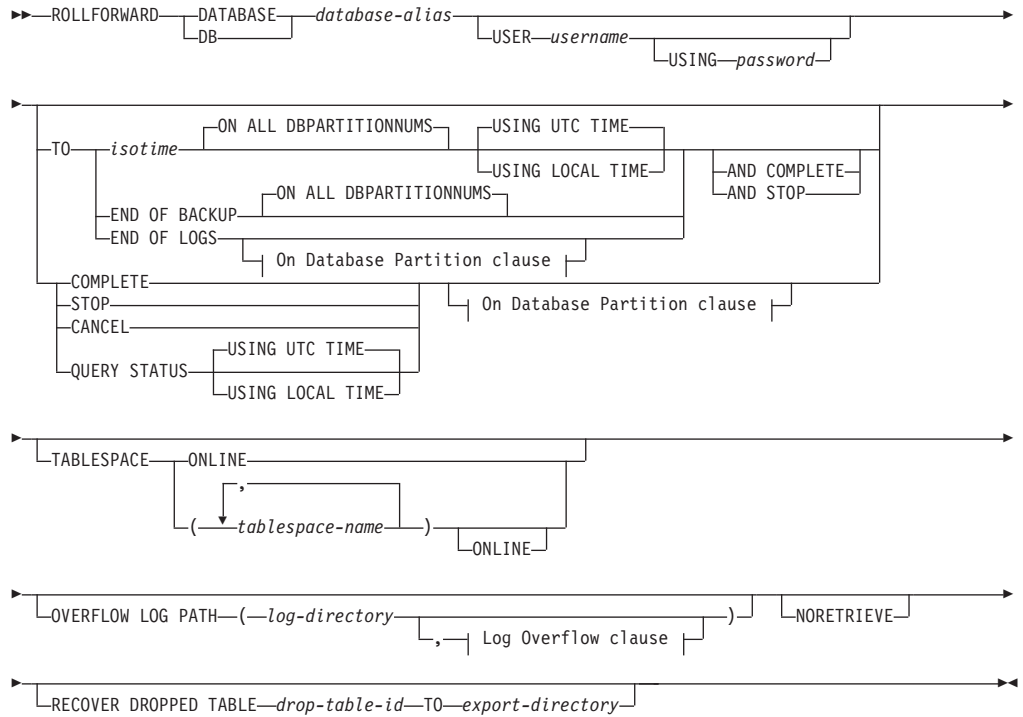
- SYSADM
- SYSCTRL

- SYSMMAINT

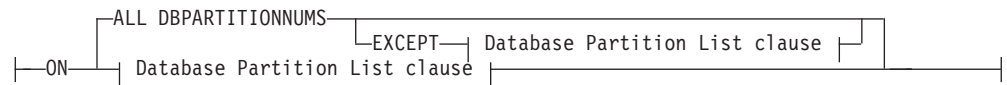
**必要な接続**

なし。このコマンドにより、排他的データベース接続が確立されます。

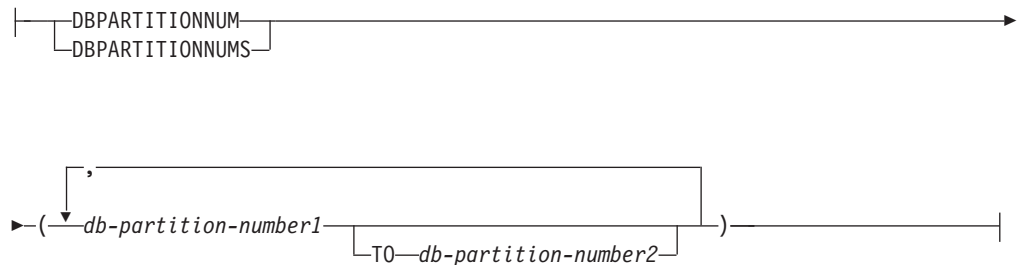
**コマンド構文**



**On Database Partition clause:**

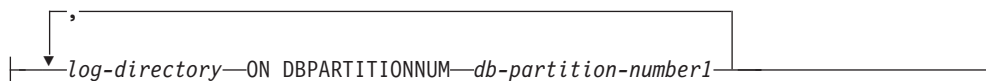


**Database Partition List clause:**





## Log Overflow clause:



## コマンド・パラメーター

### DATABASE *database-alias*

ロールフォワード・リカバリーされるデータベースの別名。

### USER *username*

データベースをロールフォワード・リカバリーするユーザー名。

### USING *password*

ユーザー名を認証するために使用するパスワード。パスワードを省略すると、入力を求めるプロンプトが出ます。

### TO

*isotime* この時点までに、コミットされたすべてのトランザクションがロールフォワードされます (その時点にコミットされたトランザクションと、それ以前にコミットされたすべてのトランザクションが含まれます)。特定時点までのロールフォワード操作で成功メッセージが返されるのは、それより大きいタイム・スタンプ値のトランザクションがログ・ファイルにある場合のみです。それより大きいタイム・スタンプのトランザクションがない場合であっても、COMPLETE オプションを指定したロールフォワード操作を実行できます。

この値は、タイム・スタンプとして指定されます。これは、日付と時刻の組み合わせを指定する 7 つの部分からなる文字ストリングです。形式は、*yyyy-mm-dd-hh.mm.ss* (年、月、日、時、分、秒) の協定世界時 (UTC、以前 GMT と呼ばれていたもの) です。UTC は、異なるログに同じタイム・スタンプが関連付けられるのを回避するために役立ちます (例えば、夏時間調整に伴う時間の変更などがあるため)。バックアップ・イメージのタイム・スタンプは、バックアップ操作が開始されたローカル時刻に基づいています。CURRENT TIMEZONE 特殊レジスターは、UTC とアプリケーション・サーバーのローカル時刻との差異を指定します。この差異は、時刻期間で表されます (最初の 2 文字が時間数を表し、次の 2 桁が分数を表し、最後の 2 桁が秒数を表す 10 進数です)。ローカル時刻から CURRENT TIMEZONE を減算すると、ローカル時刻を UTC に変換できます。

### USING UTC TIME

UTC 時刻として指定されたポイント・イン・タイムまでロールフォワードできます。これはデフォルト・オプションです。

### USING LOCAL TIME

UTC 時刻ではなく、サーバーのローカル時刻のポイント・イン・タイムまでロールフォワードできます。

注:



1. ロールフォワードにローカル時刻を指定すると、戻されるメッセージもすべてローカル時刻になります。すべての時刻はサーバー上で変換されます。パーティション・データベース環境では、カタログ・データベース・パーティションで変換されます。
2. タイム・スタンプ・ストリングは、サーバー上で UTC に変換されるため、この時刻はクライアントではなく、サーバーのタイム・ゾーンのローカル時刻になります。クライアントとサーバーのタイム・ゾーンが異なっている場合、サーバーのローカル時刻が使用されます。
3. タイム・スタンプ・ストリングが夏時間調整によるクロックの時刻変更に近い場合は、停止時間が時刻変更の前か後かを確認し、正しく指定しておくことが重要です。
4. このオプションが指定されていると、**USING LOCAL TIME** 節を指定できない後続の **ROLLFORWARD** コマンドは、すべてのメッセージをローカル時刻で戻します。
5. **USING LOCAL TIME** または **USING UTC TIME** (以前 GMT 時と呼ばれていたもの) を正しく選択することが大切です。指定されない場合、デフォルトは **USING UTC TIME** です。選択を誤ると、ロールフォワードが予想とは異なる時点に到達し、その時点より後のログが切り捨てられることがあります。ローカル・タイム・スタンプと UTC タイム・スタンプを取り違えると、必要なログが不用意に切り捨てられ、その誤った時点以降にロールフォワードを進められなくなる可能性があります。
6. データベースをリカバリーするときには有効なタイム・スタンプを指定してください。有効なタイム・スタンプは、パーティション・データベース・システムで最新のバックアップが完了した時刻になります。
7. **ROLLFORWARD DATABASE** コマンドを複数回発行するとき、2 回目以降の各コマンドで指定するタイム・スタンプは、その直前のコマンドで指定したタイム・スタンプより大きくする必要があります。

**END OF LOGS**

データベース構成パラメーター **logpath** にリストされたすべてのオンライン・アーカイブ・ログ・ファイルにある、コミットされたすべてのトランザクションが適用されることを指定します。

**END OF BACKUP**

パーティション・データベースの全パーティション、または DB2 pureScale 環境内のすべてのメンバーが、最小リカバリー時間までロールフォワードされることを指定します。その例については、ここを参照してください。

**ALL DBPARTITIONNUMS | ON ALL DBPARTITIONNUMS**

db2nodes.cfg ファイルに指定されているすべてのデータベース・パーティションでトランザクションがロールフォワードされることを指定します。データベース・パーティション節が指定されていない場合、これがデフォルトです。

## ROLLFORWARD DATABASE

**EXCEPT** データベース・パーティション・リストに指定されているものを除き、  
db2nodes.cfg ファイルに指定されているすべてのデータベース・パーティションに対してトランザクションがロールフォワードされることを指定します。

### ON DBPARTITIONNUM | ON DBPARTITIONNUMS

データベース・パーティションのセットに対してデータベースをロールフォワードします。

#### *db-partition-number1*

データベース・パーティション・リスト内のデータベース・パーティション番号を指定します。

#### *db-partition-number2*

2 番目のデータベース・パーティション番号を指定し、  
*db-partition-number1* から *db-partition-number2* までのすべてのデータベース・パーティションがデータベース・パーティション・リストに含まれるようにします。

### COMPLETE | STOP

ログ・レコードのロールフォワードを停止し、未完了トランザクションをロールバックし、データベースのロールフォワード・ペンディング状態をオフにすることによって、ロールフォワード・リカバリー処理を完了します。これによって、ロールフォワードされているデータベースまたは表スペースにアクセスできるようになります。これらのキーワードは同等です。いずれかを指定し、両方は指定しないでください。キーワード **AND** を使用すると、一度に複数の操作を指定することができます (例えば、`db2 rollforward db sample to end of logs and complete`)。ある時点まで表スペースをロールフォワードする場合、この表スペースはバックアップ・ペンディング状態になります。

**CANCEL** ロールフォワード・リカバリー操作をキャンセルします。これは、順方向リカバリーが開始されたデータベースまたはすべてのデータベース・パーティション上の 1 つ以上の表スペースを、リストア・ペンディング状態にします。

- データベース のロールフォワード操作が進行中ではない場合 (つまり、データベースがロールフォワード・ペンディング状態の場合)、このオプションはデータベースをリストア・ペンディング状態にします。
- 表スペース のロールフォワード操作が進行中ではない場合 (つまり、表スペースがロールフォワード・ペンディング状態の場合)、表スペースのリストが指定されている必要があります。リスト内のすべての表スペースが、リストア・ペンディング状態にされます。
- 表スペースのロールフォワード操作が進行中 の場合 (つまり、少なくとも 1 つの表スペースがロールフォワード進行中状態にある)、ロールフォワード進行中状態のすべての表スペースは、リストア・ペンディング状態にされます。表スペースのリストが指定されている場合、これにはロールフォワード進行中状態であるすべての表スペースが含まれていなければなりません。リスト内のすべての表スペースはリストア・ペンディング状態にされます。

- ある時点までロールフォワードしている場合、渡された表スペース名は無視され、ロールフォワード進行中状態であるすべての表スペースがリストア・ペンディング状態にされます。
- 表スペース・リストを使用してログの末尾までロールフォワードしている場合は、リストにある表スペースのみがリストア・ペンディング状態にされます。

このオプションは、実際に実行中のロールフォワード操作をキャンセルするために使用することはできません。このオプションは、進行中であってもその時点では実際には実行されていないロールフォワード操作をキャンセルするためにのみ使用できます。ロールフォワード操作が進行中であっても実行されていない場合とは、次のような場合です。

- 異常終了した。
- **STOP** オプションが指定されていない。
- エラーによって失敗した。リカバリー不能なロード操作によるロールフォワードなどの一部のエラーにより、表スペースがリストア・ペンディング状態にされることがあります。

このオプションを使用するときは注意してください。このオプションは、一部の表スペースがロールフォワード・ペンディング状態またはリストア・ペンディング状態にされたために、進行中のロールフォワード操作を完了できない場合のみ使用するようになっています。不明確な場合は、**LIST TABLESPACES** コマンドを使用して、ロールフォワード進行中状態またはロールフォワード・ペンディング状態にある表スペースを確認します。

## QUERY STATUS

ロールフォワード処理が開始してからデータベース・マネージャーが処理したログ・ファイル、必要な次のアーカイブ・ファイル、および最後にコミットされたトランザクションのタイム・スタンプ (UTC) をリストします。パーティション・データベース環境および DB2 pureScale 環境では、この状況情報は各データベース・パーティションまたはメンバーごとに戻されます。戻される情報には、次のフィールドが含まれます。

### メンバー番号

メンバー番号またはデータベース・パーティション番号。単一パーティション環境では、これは常に 0 です。

### ロールフォワード状況

状態は、データベースまたは表スペースのロールフォワード・ペンディング、データベースまたは表スペースのロールフォワード進行中、データベースまたは表スペースのロールフォワード処理停止、または非ペンディングのいずれかとなります。

### 読み込む予定の次のログ・ファイル

次に必要なログ・ファイルの名前が含まれるストリング。パーティション・データベース環境で、ロールフォワード・ユーティリティーに障害が起これば、ログ・ファイルの欠落を示す戻りコード、またはログ情報の不一致が生じたことを示す戻りコードが戻されたとき、この情報を使用します。

### 処理済みログ・ファイル

これ以上リカバリーに必要なく、そのディレクトリーから除去でき

る、処理済みのログ・ファイルの名前から成るストリング。DB2 pureScale 環境では、このストリングは、すべての DB2 メンバーで **ROLLFORWARD** コマンドによって処理されてきたすべてのログ・ファイルをリストします。例えば、最も古い非コミット・トランザクションがログ・ファイル  $x$  で始まる場合、古くなったログ・ファイルの範囲には  $x$  は含まれず、この範囲は、 $x - 1$  で終わります。このフィールドは、表スペース・ロールフォワード・リカバリー操作の場合には更新されません。

### 最後にコミットしたトランザクション

ISO 形式 ( $yyyy-mm-dd-hh.mm.ss$ ) の UTC または Local のいずれかが末尾に付けられたタイム・スタンプ (**USING LOCAL TIME** を参照) を含むストリング。このタイム・スタンプは、ロールフォワード・リカバリーの完了後に、最後にコミットされたトランザクションのタイム・スタンプです。このタイム・スタンプは、データベースに適用されます。表スペースのロールフォワード・リカバリーの場合、これはデータベースにコミットされた最後のトランザクションのタイム・スタンプになります。

**TO**、**STOP**、**COMPLETE**、または **CANCEL** 節が省略されている場合は、**QUERY STATUS** がデフォルト値になります。**TO**、**STOP**、または **COMPLETE** が指定されている場合、コマンドが正常に完了すると、状況情報が表示されます。個々の表スペースを指定する場合は、それらの表スペースは無視されます。状況要求は、指定された表スペースだけに適用されるものではないからです。

### TABLESPACE

このキーワードは、表スペース・レベルのロールフォワード・リカバリーに対して指定されます。

#### *tablespace-name*

特定時点までの表スペース・レベルのロールフォワード・リカバリーに必須です。これにより、ログの最後までロールフォワード・リカバリーに対して表スペースのサブセットを指定できます。パーティション・データベース環境では、リストの各表スペースがロールフォワードされている各データベース・パーティションに存在している必要はありません。もし表スペースが存在している場合、その表スペースは適正な状態になければなりません。

パーティション表の場合、パーティション表の一部が含まれた表スペースの特定時点ロールフォワードでは、その表がある他のすべての表スペースも同じ時点までロールフォワードする必要があります。索引パーティションが含まれている表スペースは、パーティション表の部分リストに組み込まれています。パーティション表の一部を収めた 1 つの表スペースのログの最後までロールフォワードすることも可能です。

パーティション化された表にアタッチまたはデタッチされたデータ・パーティションがある場合、**PIT** ロールフォワードには、これらのデータ・パーティションのすべての表スペースも含める必要があります。パーティション化された表にアタッチ、デタッチ、またはドロップされたデータ・パーティションがあるかどうかを判別するには、**SYSDATAPARTITIONS** カタログ表の状況フィールドを照会します。

パーティション表は、複数の表スペースにある可能性があるため、通常は複数の表スペースをロールフォワードする必要があります。ドロップされた表

のリカバリーによってリカバリーされたデータは、**ROLLFORWARD DATABASE** コマンドで指定されたエクスポート・ディレクトリーに書き込まれます。1つのコマンドですべての表スペースをロールフォワードすることも、関与する表スペースのサブセットに対してロールフォワード操作を繰り返すこともできます。**ROLLFORWARD DATABASE** コマンドを 1 つまたはいくつかの表スペースに対して実行すると、それらの表スペースにあった表のすべてのデータがリカバリーされます。**ROLLFORWARD DATABASE** コマンドが、表のすべてのデータを回復するために必要な表スペースの全セットを指定していないと、通知ログに警告が書き込まれます。表スペースのサブセットのロールフォワードを許可することによって、単一のエクスポート・ディレクトリーに収容できる分より多くのデータをリカバリーする場合に、処理が容易になります。

**ONLINE** このキーワードは、表スペース・レベルのロールフォワード・リカバリーをオンラインで実行できるようにするために指定されます。これは、ロールフォワード・リカバリーの進行中に、他のエージェントが接続できることを意味します。DB2 pureScale 環境で他のアプリケーションとの並行性を確保するため、表スペース・レベルのロールフォワード操作のテストを行って、特定の内部ロックを確認しなければならない場合があります。この環境でのロールフォワード操作は、**locktimeout** が -1 に設定されていても、ロック・タイムアウトによって失敗する可能性があります。

#### **OVERFLOW LOG PATH** *log-directory*

リカバリー中に、アーカイブ・ログを検索する代替のログ・パスを指定します。**logpath** データベース構成パラメーターによって指定されている以外の場所にログ・ファイルが移動された場合には、このパラメーターを使用してください。パーティション・データベース環境では、これはすべてのデータベース・パーティションの (完全修飾) デフォルト・オーバーフロー・ログ・パスになります。単一パーティション・データベースには、相対オーバーフロー・ログ・パスを指定できます。**OVERFLOW LOG PATH** コマンド・パラメーターは、データベース構成パラメーター **overflowlogpath** の値 (存在する場合) を上書きします。

#### *log-directory* **ON DBPARTITIONNUM**

パーティション・データベース環境では、これにより、特定のデータベース・パーティションのデフォルト・オーバーフロー・ログ・パスを別のログ・パスでオーバーライドできます。

#### **NORETRIEVE**

アーカイブ・ログの検索を無効にすることによって、ロールフォワードするログ・ファイルを制御できます。

#### **RECOVER DROPPED TABLE** *drop-table-id*

ロールフォワード操作中に、ドロップされた表をリカバリーします。表 ID は、出力リストのバックアップ ID 列で、**LIST HISTORY** コマンドを使用して取得できます。パーティション表の場合、*drop-table-id* は表全体を示します。したがって、表のすべてのデータ・パーティションを単一のロールフォワード・コマンドでリカバリーすることができます。



## ROLLFORWARD DATABASE

### TO *export-directory*

表データが含まれたファイルが書き込まれるディレクトリーを指定します。このディレクトリーは、すべてのデータベース・パーティションからアクセス可能でなければなりません。

### 例

- **例 1: ROLLFORWARD DATABASE** コマンドでは、それぞれをキーワード **AND** で区切ることによって、一度に複数の操作を指定することができます。例えば、ログの最後までロールフォワードして完了する場合、コマンドを別々に指定すると、次のようになります。

```
db2 rollforward db sample to end of logs
db2 rollforward db sample complete
```

これらは、次のように結合できます。

```
db2 rollforward db sample to end of logs and complete
```

これら 2 つは同等ですが、このような操作は 2 つのステップで実行することを推奨します。ロールフォワード操作が期待どおりに進行したことを確認してから、ロールフォワード操作を停止することが重要です。停止の際に、ログは失われる可能性があります。これは、ロールフォワード・リカバリー中に不良ログが検出され、不良ログが「ログの終わり」を意味すると解釈される場合は特に重要です。このような場合は、そのログの損傷していないバックアップ・コピーを使用して、それ以降のログのロールフォワード操作を続けることができます。しかし、ロールフォワード **AND STOP** オプションが使用されている場合、ロールフォワードでエラーが発生すると、そのエラーが戻されます。この場合、エラーがあってもロールフォワードを強制的に停止してオンラインにする（つまりログの中のそのエラーの前の時点でオンラインにする）ための唯一の方法は、**ROLLFORWARD STOP** コマンドを発行することです。

- **例 2:** ログの終わりまでロールフォワードします（2 つの表スペースがリストアされます）。

```
db2 rollforward db sample to end of logs
db2 rollforward db sample to end of logs and stop
```

これら 2 つのステートメントは同等です。ログの終わりまでの表スペースのロールフォワード・リカバリー操作を実行する場合、**AND STOP** や **AND COMPLETE** はいずれも不要です。表スペース名は必要ありません。表スペース名を指定しない場合、ロールフォワード・リカバリーが必要なすべての表スペースが含まれます。これらの表スペースのサブセットのみをロールフォワードする場合は、それらの名前を指定する必要があります。

- **例 3:** 3 つの表スペースがリストアされた後、1 つをログの終わりまでロールフォワードし、他の 2 つを特定時点までロールフォワードします（両方ともオンラインで行われます）。

```
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
```

2 つのロールフォワード操作を同時に実行することはできません。2 番目のコマンドは、最初のロールフォワード操作が正しく完了した後にのみ呼び出せます。

- 例 4: データベースのリストアが完了すると、ポイント・イン・タイムまでロールフォワードします。このとき、**OVERFLOW LOG PATH** を使用して、ユーザー出口がアーカイブ・ログを保存するディレクトリーを指定します。

```
db2 rollforward db sample to 1998-04-03-14.21.56 and stop
overflow log path (/logs)
```

- 例 5 (パーティション・データベース環境): 0、1、および 2 の、3 つのデータベース・パーティションがあります。表スペース **TBS1** はすべてのデータベース・パーティションで定義されており、表スペース **TBS2** はデータベース・パーティション 0 および 2 で定義されています。データベース・パーティション 1 でデータベースをリストアし、データベース・パーティション 0 および 2 で **TBS1** をリストアした後、データベース・パーティション 1 でデータベースをロールフォワードします。

```
db2 rollforward db sample to end of logs and stop
```

これにより、警告 **SQL1271** (「データベースは回復されましたが、データベース・パーティション 0 および 2 で 1 つ以上の表スペースがオフラインになっています。」) が戻されます。

```
db2 rollforward db sample to end of logs
```

これにより、データベース・パーティション 0 および 2 で **TBS1** がロールフォワードされます。この場合、節 **TABLESPACE(TBS1)** はオプションです。

- 例 6 (パーティション・データベース環境): データベース・パーティション 0 および 2 でのみ表スペース **TBS1** をリストアした後、データベース・パーティション 0 および 2 で **TBS1** をロールフォワードします。

```
db2 rollforward db sample to end of logs
```

データベース・パーティション 1 は無視されます。

```
db2 rollforward db sample to end of logs tablespace(TBS1)
```

データベース・パーティション 1 で **TBS1** がロールフォワード・リカバリーを行える状態になっていないため、これは失敗します。 **SQL4906N** を報告します。

```
db2 rollforward db sample to end of logs on dbpartitionnums (0, 2)
tablespace(TBS1)
```

これは正しく完了します。

```
db2 rollforward db sample to 1998-04-03-14.21.56 and stop
tablespace(TBS1)
```

データベース・パーティション 1 で **TBS1** がロールフォワード・リカバリーを行える状態になっていないため、これは失敗します。すべての部分は一緒にロールフォワードされなければなりません。表スペースをある時点までロールフォワードする場合、データベース・パーティション節は受け入れられません。ロールフォワード操作は、表スペースが存在するすべてのデータベース・パーティションで行う必要があります。

データベース・パーティション 1 で **TBS1** をリストアした後、次のようにします。

```
db2 rollforward db sample to 1998-04-03-14.21.56 and stop
tablespace(TBS1)
```



## ROLLFORWARD DATABASE

これは正しく完了します。

- 例 7 (パーティション・データベース環境): すべてのデータベース・パーティションで表スペースをリストアした後、時点 2 (pit2) までロールフォワードしますが、**AND STOP** は指定しません。ロールフォワード操作はまだ進行中です。それを取り消し、時点 1 (pit1) までロールフォワードします。

```
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)
```

- 例 8 (パーティション・データベース環境): db2nodes.cfg ファイルにリストされた 8 つのデータベース・パーティション (3 から 10 まで) にある表スペースをロールフォワード・リカバリーします。

```
db2 rollforward database dwtest to end of logs tablespace (tssprodt)
```

ログの終わり (特定時点ではなく) までのこの操作は正常に完了します。表スペースが存在するデータベース・パーティションは指定する必要がありません。ユーティリティーは、デフォルトとして db2nodes.cfg ファイルを使用します。

- 例 9 (パーティション・データベース環境): (データベース・パーティション 6 上の) 単一パーティションのデータベース・パーティション・グループに存在する 6 つの小さな表スペースをロールフォワード・リカバリーします。

```
db2 rollforward database dwtest to end of logs on dbpartitionnum (6)
tablespace(tsstore, tssbuyer, tsstime, tsswhse, tsslscat, tssvendor)
```

ログの終わり (特定時点ではなく) までのこの操作は正常に完了します。

- 例 10 (パーティション・データベース環境または DB2 pureScale 環境): **TO END OF BACKUP** 節を **ROLLFORWARD** コマンドで使用することにより、すべてのデータベース・パーティションまたはメンバーを最小のリカバリー時間にロールフォワードすることができます。最小リカバリー時間は、データベースが整合状態にあるとき (データベース・カタログにリストされるオブジェクトがディスク上に物理的に存在するオブジェクトと一致しているとき) の、ロールフォワード中の最も早い時点です。データベースをロールフォワードする正確な時点を手動で判別することは困難です (特にパーティション・データベースや DB2 pureScale インスタンスの場合)。**END OF BACKUP** オプションを使用することにより、これを簡単に行うことができます。

```
db2 rollforward db sample to end of backup and complete
```

- 例 11 (パーティション・データベース環境または DB2 pureScale 環境): ユーザーは、/backup\_logs にリストアされたログを含むオンライン・データベース・バックアップをリストアします。

```
db2 restore database dwtest ... logtarget /backup_logs ...
```

次にユーザーは、バックアップからのログを使用し、オーバーフロー・ログ・パスを通して、バックアップの終わりまで、ロールフォワードを行います。

```
db2 rollforward database dwtest to end of backup and stop overflow log path
(/backup_logs)
```

- 例 12 (DB2 pureScale 環境): ユーザーは、オーバーフロー・ログ・パスを使用するロールフォワード用のログ・ファイルの代替セットを用意しようと思ってい

ます。0、1、および2のログ・ストリーム3つがあると仮定します。これを行うためにユーザーは、必要なディレクトリーのセットを作成する必要があります。

```
mkdir -p /overflow/NODE0000/LOGSTREAM0000
mkdir -p /overflow/NODE0000/LOGSTREAM0001
mkdir -p /overflow/NODE0000/LOGSTREAM0002
```

次にユーザーは、使用予定のログ・ファイルをディレクトリーに入れる必要があります。

**注:** オーバーフロー・ログ・パス内のいずれかのログ・ファイルがアクティブ・ログ・パスにも存在する場合には、オーバーフロー・ログ・パス内のログ・ファイルの代わりに、アクティブ・ログ・パスからのログ・ファイルが使用されることとなります。

ディレクトリーにファイルが入れられた後、ユーザーは次のようにロールフォワード・コマンドを発行できます。

```
rollforward database dwtest to end of logs overflow log path (/overflow)
```

そのロールフォワードによって十分な前進が得られることを検証した後、ユーザーは次のコマンドを発行してロールフォワードを完了することができます。

```
rollforward database dwtest to end of logs and stop overflow log path (/overflow)
```

## 使用上の注意

オンライン・バックアップ操作中に作成されたイメージからリストアする場合は、ロールフォワード操作のために指定される時点は、オンライン・バックアップ処理が完了する時点より後でなければなりません。ロールフォワード操作がこのポイントを過ぎる前に停止すると、データベースはロールフォワード・ペンディング状態のままとなります。表スペースがロールフォワード中の場合は、ロールフォワード進行中状態のままとなります。

1つ以上の表スペースをある時点までロールフォワードしている場合は、ロールフォワード操作は、最低でも最小リカバリー時間（この表スペースまたはその表のシステム・カタログに関して最後に更新されたもの）まで継続する必要があります。表スペースの最小リカバリー時間（協定世界時、または UTC）は、**LIST TABLESPACES SHOW DETAIL** コマンドを使用して取得できます。DB2 pureScale 環境では、**LIST TABLESPACES** コマンドは推奨されません。次のモニター UDF を使用してください。SELECT \* FROM TABLE( SYSPROC.MON\_GET\_TABLESPACE('TBSpace\_1,0) )

DB2 pureScale 環境では、ロールフォワード操作を始める前に、リトリート・パス内に十分な空きディスク・スペースがあることを確認してください。それによって、パフォーマンスに影響することなく、DB2 pureScale 環境の必要に応じて、アーカイブから多数のファイルをリトリートする操作が可能になります。次の公式を使用して、すべてのメンバーのアクティブ・ログ・スペースをリトリートするのに必要なスペース量を計算します:  $(\text{logprimary} + \text{logsecond}) * \text{number of members}$ .

システム期間テンポラル表またはバイテンポラル表を含む表スペースが、ある時点までロールフォワードされる場合、**ROLLFORWARD** コマンドには、関連した履歴表を含む表スペースを組み込む必要があります。テンポラル表のための表スペースと履

## ROLLFORWARD DATABASE

歴表のための表スペースを別々にロールフォワードすることができますが、ロールフォワードできるのはログの終わりまでです。

データベースをロールフォワードするとき、テープ装置を使用するロード・リカバリーが必要な場合もあります。別のテープを求める要求が出された場合は、次の入力のいずれかで応答できます。

- c 続行。警告メッセージを生成した装置を使用し続けます (例えば、新しいテープをマウントした場合)。
- d 装置の終了。警告メッセージを生成した装置の使用を停止します (例えば、それ以上テープがない場合)。
- t 終了。影響を受けるすべての表スペースをオフラインにしますが、ロールフォワードの処理は継続します。

ロールフォワード・ユーティリティーが、必要とする次のログを検出できない場合は、そのログ名が **SQLCA** に戻され、ロールフォワード・リカバリーが停止します。さらに使用可能なログがない場合は、**STOP** オプションを使用してロールフォワード・リカバリーを終了します。未完了のトランザクションはロールバックされ、データベースまたは表スペースが確実に整合した状態になるようにします。

データベースのロールフォワードで表スペース・スキーマ転送のログ・レコードが検出された場合、対応する転送済み表スペースはオフラインになってドロップ・ペンディング状態に移ります。これは、転送された表スペースとその内容を再作成するための転送済み表スペースの完全なログが、データベースにないためです。転送完了後にターゲット・データベースのフルバックアップを取ることができるので、その後のロールフォワードがログ・ストリーム内のスキーマ転送のポイントを通過することはありません。

注: ログ・レコードにはデータの再分散については記録されないため、再分散操作のロールフォワードでは、データベースの内容はリストアできません。

『**REDISTRIBUTE DATABASE PARTITION GROUP** コマンド』を参照してください。

## RUNCMD

CLP 対話モード・コマンド履歴から指定のコマンドを実行します。

### 有効範囲

このコマンドは、CLP 対話モードでのみ実行できます。特に CLP コマンド・モードや CLP バッチ・モードでは実行できません。

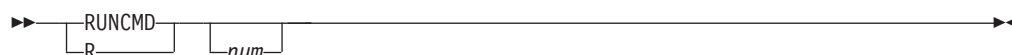
### 許可

なし

### 必要な接続

必要な接続は、実行するコマンドによって異なります。

### コマンド構文



### コマンド・パラメーター

*num* *num* が正の場合は、コマンド履歴の *num* に対応するコマンドを実行します。*num* が負の場合は、コマンド履歴の最新のコマンドから逆算する形で *num* に対応するコマンドを実行します。ゼロは、*num* の有効な値ではありません。このパラメーターを指定しない場合は、実行した最新のコマンドを実行します。(この動作は、*num* に -1 を指定した場合と同じです。)

### 使用上の注意

1. 通常は、最近実行したコマンドのリストを表示するために **HISTORY** コマンドを実行してから、そのリスト内に存在するコマンドを実行するために **RUNCMD** を実行します。
2. **RUNCMD** コマンドはコマンド履歴に記録されませんが、**RUNCMD** コマンドによって実行したコマンドはコマンド履歴に記録されます。

## RUNSTATS

表やそれに関連した索引の特性、あるいは統計ビューの特性に関する統計情報を更新します。これらの特性には、レコード数、ページ数、および平均レコード長が含まれます。オプティマイザーは、データへのアクセス・パスを判別するとき、これらの統計を使用します。

表の場合、表で数多くの更新が行われたとき、または表の再編成後に、RUNSTATS コマンドを呼び出します。統計ビューの場合、基礎表に対する変更がそのビューによって返される行に対してかなりの影響を及ぼしている場合に、RUNSTATS コマンドを呼び出します。そのビューは、それ以前に ALTER VIEW ステートメントを使用して、照会最適化で使用できる状態になっていなければなりません。

### 有効範囲

RUNSTATS コマンドは、db2nodes.cfg ファイル内のいずれのデータベース・パーティションからでも発行できます。この API は、カタログ・データベース・パーティション上のカタログを更新するために使用できます。

表の場合、このコマンドは、呼び出し元のデータベース・パーティションの表の統計を収集します。表がそのデータベース・パーティションに存在しない場合、データベース・パーティション・グループの最初のデータベース・パーティションが選択されます。

ビューの場合、このコマンドは、関連するすべてのデータベース・パーティションに含まれる表のデータを使用して、統計情報を収集します。

### 許可

表の場合、以下の権限のいずれか 1 つです。

- SYSADM
- SYSCTRL
- SYSMANT
- DBADM
- SQLADM
- 表に対する CONTROL 特権
- LOAD authority

このコマンドを使用する際には、接続内に存在する宣言された一時表のいずれにおいても明示特権は必要ありません。

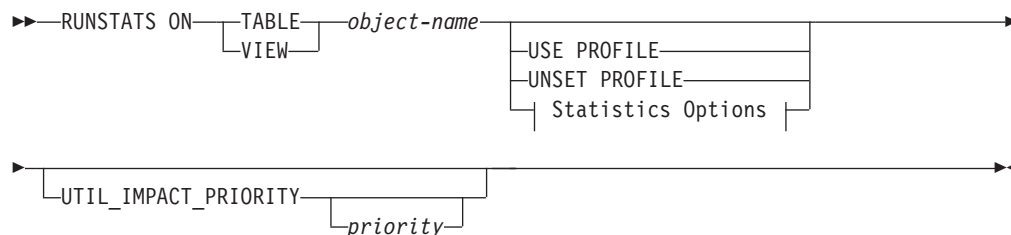
統計ビューの場合、以下の権限のいずれか 1 つです。

- SYSADM
- SYSCTRL
- SYSMANT
- DBADM
- SQLADM
- 統計ビューに対する CONTROL 特権

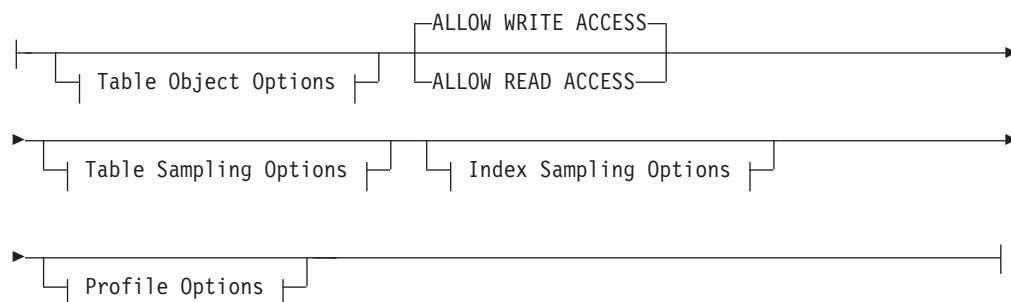
## 必要な接続

データベース

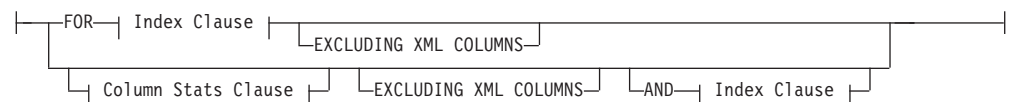
## コマンド構文



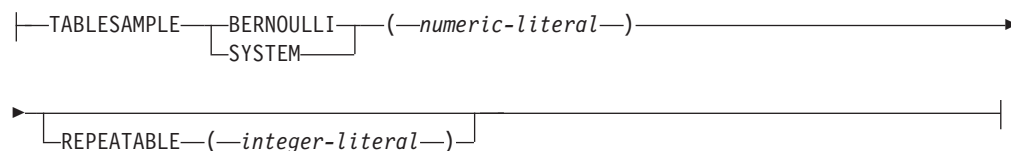
### Statistics Options:



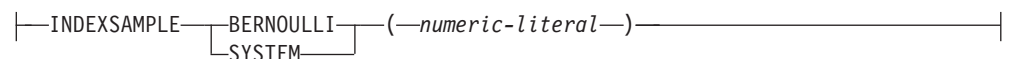
### Table Object Options:



### Table Sampling Options:



### Index Sampling Options:

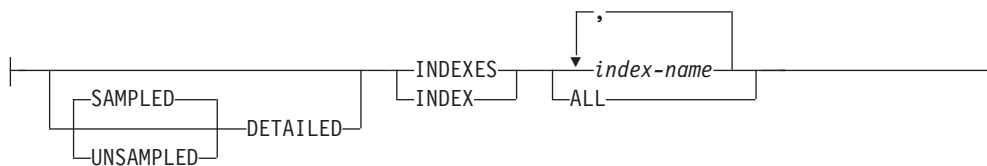


# RUNSTATS

## Profile Options:



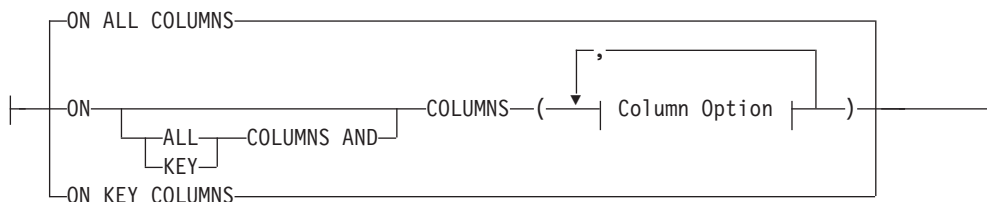
## Index Clause:



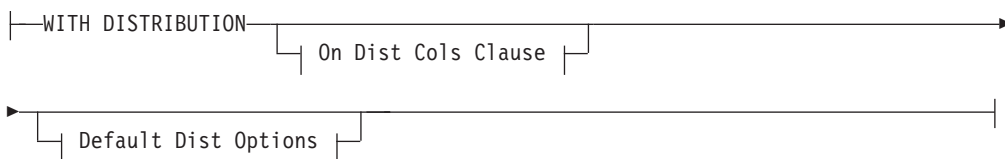
## Column Stats Clause:



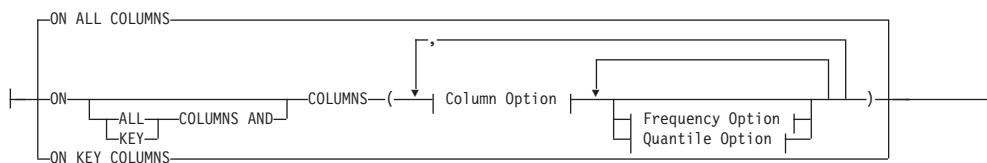
## On Cols Clause:



## Distribution Clause:

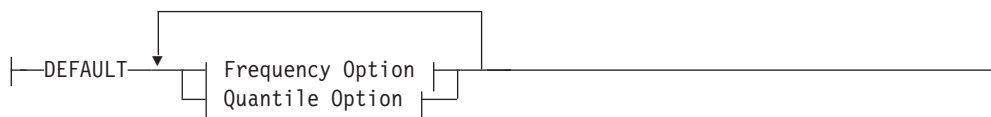


## On Dist Cols Clause:



## Default Dist Option:





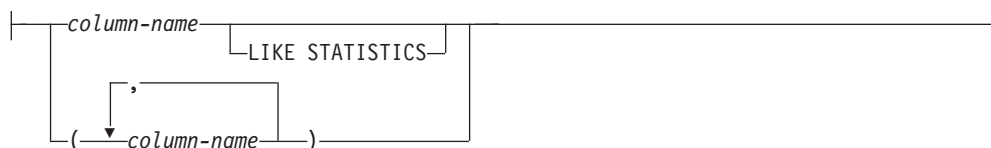
**Frequency Option:**



**Quantile Option:**



**Column Option:**



**コマンド・パラメーター**

*object-name*

統計が収集される表または統計ビューを識別します。このパラメーターは、階層表であってはなりません。型付き表の場合は、*object-name* は表階層のルート表の名前でなければなりません。 *schema.object-name* という形式の完全修飾名または別名を使用しなければなりません。 *schema* には、表作成時のユーザー名が入ります。

**USE PROFILE**

このオプションを使用した **RUNSTATS** は、以前に保管した統計プロファイルを使用して表または統計ビューの統計を収集します。統計プロファイルの作成には、**SET PROFILE** オプションを使用し、更新には **UPDATE PROFILE** オプションを使用します。

**UNSET PROFILE**

既存の統計プロファイルを除去する場合は、このオプションを指定します。以下に例を示します。

```
RUNSTATS ON tablemyschema.mytable UNSET PROFILE
```

**FOR INDEXES**

索引のみの統計を収集および更新します。表に、以前に収集された表統計がない場合は、基本表統計も収集されます。これらの基本統計には、分散統計は一切含まれません。ビューの場合、このオプションは使用できません。

**SAMPLED**

組み合わせて使用できるのは **DETAILED** パラメーターのみです。このオプションを指定しても、**DETAILED** によるデフォルトの機能は変更されません。このオプションは、以前のバージョンの DB2 との互換性のために残されています。ビューの場合、このオプションは使用できません。

**UNSAMPLED**

このオプションが **DETAILED** オプションと共に使用された場合、強制的に **RUNSTATS** が実行されることにより、索引内のすべての項目が調べられて拡張索引統計が計算されます。このオプションはビューに対しては使用できず、スキャン索引サンプリング (**INDEXSAMPLE** キーワード) と共に使用することもできません。このオプションを指定すると **RUNSTATS** のリソース使用量が大幅に増加させられます。一方、**DETAILED** オプションおよび **SAMPLED DETAILED** オプション (これらは等価です) では大幅な改善が得られることはほとんどありません。

**DETAILED**

拡張された索引統計を計算します。拡張された索引統計とは、比較的大規模な索引の場合に収集される、**CLUSTERFACTOR** 統計および **PAGE\_FETCH\_PAIRS** 統計のことです。パフォーマンスを改善するため、索引項目のすべてを調べるのではなく、CPU サンプリング技法が採用されます。ビューの場合、このオプションは使用できません。

*index-name*

表で定義されている既存の索引を識別します。 *schema.index-name* 形式の完全修飾名を指定しない場合、デフォルトのスキーマが想定されます。ビューの場合、このオプションは使用できません。

**EXCLUDING XML COLUMNS**

この節を使用して、XML タイプのすべての列を、統計情報収集から除外します。XML データを含めた場合、必要となるシステム・リソースが増加することがあるため、この節を使用すれば、XML 以外の列に関する統計情報の収集が容易になります。 **EXCLUDING XML COLUMNS** 節は、統計情報収集の対象として XML 列を指定する他の節よりも優先されます。例えば、**EXCLUDING XML COLUMNS** 節を使用する場合、**ON COLUMNS** 節で XML タイプの列を指定したり **ON ALL COLUMNS** 節を使用したりしても、統計情報収集において XML タイプの列はすべて無視されます。DB2 V9.7 フィックスパック 1 以降のリリースでは、この節が指定される場合、XML タイプの列に対する分散統計は収集されません。

**AND INDEXES**

表と索引両方の統計を収集および更新します。ビューの場合、このオプションは使用できません。

**ON ALL COLUMNS**

適格なすべての列で統計を収集するには、**ON ALL COLUMNS** 節を使用します。列は、基本統計の収集に指定するか (**On Co1s** 節)、または **WITH DISTRIBUTION** 節と組み合わせて指定 (**On Dist Co1s** 節) できます。これら列固有の節がどちらも指定されない場合は、デフォルト・オプションとして **ON ALL COLUMNS** が指定されます。

なお、**On Co1s** 節でこれが指定された場合は、特定の列が **WITH DISTRIBUTION** 節の一部として選択されない限り、すべての列で基本列統計だけが収集されます。 **WITH DISTRIBUTION** 節の一部として指定された列では、基本統計も分散統計も収集されます。

**WITH DISTRIBUTION ON ALL COLUMNS** が指定されている場合は、収集が行えるすべての列で基本統計と分散統計の両方が収集されます。 **On Co1s** 節での指定はすべて重複になるため、必要ありません。

**ON COLUMNS**

特定の列、列グループ、またはその両方に関する統計を収集するには、**ON COLUMNS** を使用します。列グループとは、結合した統計を収集しようとする複数の列を括弧でくくったコンマ区切りリストです。

列および列グループは括弧でくくったコンマ区切りリストとして指定します。

索引統計情報を収集することなく表に対して **RUNSTATS** コマンドを実行し、統計情報収集の対象として列のサブセットを指定した場合、

- **RUNSTATS** コマンドに指定されていないが、索引の最初の列である列の統計情報は、リセットされません。
- **RUNSTATS** コマンドで指定されていない他のすべての列の統計情報は、リセットされます。

この節は、On Cols 節と On Dist Cols 節の中で使用できます。列のグループに関する分散統計の収集は、現在サポートされていません。

列グループの中に XML タイプの列が指定されている場合は、そのグループについて重複のない異なる値を収集するため、XML タイプのそれらの列は無視されます。しかし、その列グループ中の XML タイプ列について、XML 列の基本統計情報は収集されます。

**ON KEY COLUMNS**

特定の列をリストする代わりに、表で定義されたすべての索引を構成する列の統計を収集することもできます。ここでは、照会に含まれる重要な列が、表での索引の作成にも使用されることが前提となっています。表に索引がない場合は、列がリストされず、列統計が収集されない場合と同様になります。これは、On Cols 節または On Dist Cols 節の中で使用できます。ただし、その両方の節で指定すると、**WITH DISTRIBUTION** 節で基本統計と分散統計の両方の収集が指定されているため、On Cols 節で重複が生じます。定義により XML タイプの列はキー列ではなく、**ON KEY COLUMNS** 節による統計情報収集には含まれません。ビューの場合、このオプションは使用できません。

*column-name*

表または統計ビューの中の列の名前。存在しない列が指定された場合や列名の入力を誤った場合など、統計収集を行えない列の名前が指定された場合は、エラー (-205) が戻されます。一方は配分なし、一方は配分ありで、2つの列のリストを指定できます。**WITH DISTRIBUTION** 節が関連付けられていないリストで列を指定する場合は、基本列統計だけが収集されます。列が両方のリストに含まれている場合は、分散統計が収集されます (**NUM\_FREQVALUES** および **NUM\_QUANTILES** がゼロに設定されていない限り)。

**LIKE STATISTICS**

このオプションを指定すると、コード・ページ属性が 1 バイト文字セット (SBCS)、FOR BIT DATA、または UTF-8 であるタイプ CHAR および VARCHAR の列について、追加の列統計が収集される場合があります。これらの統計が収集されるのは、**runstats** ユーティリティーが列値を分析した後に、これらの統計が有効であると判断した場合です。これらの統計とは、SYSSTAT.COLUMNS の SUB\_COUNT および SUB\_DELIM\_LENGTH 統計です。

照会オプティマイザーは、これらを使用して "column LIKE '%xyz'" や "column LIKE '%xyz%'" のようなタイプの述部に関する選択度の見積もりを向上させます。

#### WITH DISTRIBUTION

この節は、指定された列で基本統計と分散統計の両方を収集することを指定します。 **ON COLUMNS** 節が指定されていない場合は、表または統計ビューの中のすべての列 (CLOB や LONG VARCHAR といった、収集用には選択できない列を除く) で分散統計が収集されます。一方 **ON COLUMNS** 節が指定されている場合は、提供された列リストに対してのみ (統計収集用には選択できない列を除く) 分散統計が収集されます。なお、節が指定されなければ、基本統計だけが収集されます。

列のグループに関する分散統計の収集は、現在サポートされていません。

**WITH DISTRIBUTION ON COLUMNS** 節で列のグループが指定された場合は、分散統計は収集されません。

#### DEFAULT

**NUM\_FREQVALUES** または **NUM\_QUANTILES** を指定した場合、これらの値を **ON COLUMNS** 節で個々の列に対して指定していなければ、これらの値によって、すべての列に関して収集する度数と分位の統計の最大個数が決まります。

**DEFAULT** 節が指定されない場合は、対応するデータベース構成パラメーターにある値が使用されます。

#### NUM\_FREQVALUES

収集の頻度を示す値の最大値を定義します。これは、**ON COLUMNS** 節の中で、個々の列ごとに指定できます。個々の列に対して値が指定されない場合は、**DEFAULT** 節で指定されている頻度のしきい値が選出されます。どちらも指定されていない場合は、**num\_freqvalues** データベース構成パラメーターで設定されている値が、収集の頻度を指定する値の最大値になります。

#### NUM\_QUANTILES

収集する分散変位値の最大値を定義します。これは、**ON COLUMNS** 節の中で、個々の列ごとに指定できます。個々の列に対して値が指定されない場合は、**DEFAULT** 節で指定されている変位値のしきい値が選出されます。どちらも指定されていない場合は、**num\_quantiles** データベース構成パラメーターで設定されている値が、収集する分位値の最大個数になります。

DB2 V9.7 フィックスパック 1 以降のリリースでは、XML データに対する各索引の分散統計は、デフォルトで最大 250 の分位数を使用します。このデフォルトは、**ON COLUMNS** 節または **DEFAULT** 節の **NUM\_QUANTILES** パラメーターを指定して変更することができます。XML 分散統計を収集している間、**num\_quantiles** データベース構成パラメーターは無視されます。

#### ALLOW WRITE ACCESS

統計が計算される間に、他のユーザーが表から読み込んだりそこに書き込んだりできることを指定します。統計ビューの場合は、ビュー定義の中で参照されている基本表です。

多くの挿入、更新、または削除が同時に実行される表の場合、**ALLOW WRITE ACCESS** オプションは勧められていません。**RUNSTATS** コマンドは、まず表統計を実行した後、索引統計を実行します。表および索引の統計情報を収集している間に表の状態が変化した場合、不整合が発生することがあります。

照会の最適化のために最新の統計を収集することは重要ですが、整合性のある統計を収集することも重要です。したがって統計情報の収集は、挿入、更新、または削除の操作が最小になる時間に実行してください。

#### ALLOW READ ACCESS

統計が計算される間に、他のユーザーが表に対して読み取り専用のアクセスを行えることを指定します。統計ビューの場合は、ビュー定義の中で参照されている基本表です。

#### TABLESAMPLE BERNOULLI

このオプションを使用した **RUNSTATS** は、表または統計ビューから取られた行のサンプルに関する統計を収集します。 **BERNOULLI** (ベルヌーイ) サンプルングでは各行が個別に処理され、その際に  $P/100$  ( $P$  は数値リテラル値) の確率で行が含まれ、 $1-P/100$  の確率で行が除外されます。例えば、数値リテラルが値 10 (つまり、10 % のサンプル) と評価された場合は、0.1 の確率で各行が含まれ、0.9 の確率で行が除外されます。オプションの **REPEATABLE** 節を指定しない限り、**RUNSTATS** を実行するたびに、通常は異なった表のサンプルが作成されます。すべてのデータ・ページが表スキャンによって検索されますが、数値リテラル・パラメーターによって指定したパーセンテージの行だけが、統計収集に使用されます。

#### TABLESAMPLE SYSTEM

このオプションを使用した **RUNSTATS** は、表から取られたデータ・ページのサンプルに関する統計を収集します。 **SYSTEM** (システム) サンプルングでは各ページが個別に処理され、その際に  $P/100$  ( $P$  は数値リテラル値) の確率でページが含まれ、 $1-P/100$  の確率でページが除外されます。オプションの **REPEATABLE** 節を指定しない限り、**RUNSTATS** を実行するたびに、通常は異なった表のサンプルが作成されます。サンプルのサイズは、括弧内の数値リテラル・パラメーターによって制御し、表の約  $P$  % を戻すように指定します。数値リテラル・パラメーターによって指定したパーセンテージのデータ・ページだけが、検索されて統計収集に使用されます。

統計ビューでは、**SYSTEM** サンプルングは、ビュー定義で参照される単一の基本表にのみ適用できます。ビューに複数の表が含まれている場合、統計ビュー内のすべての表から、ある単一の表をそのビュー内で使用されている他の表の主キーまたはユニーク索引列すべてによって結合されているものとして特定できるのであれば、**SYSTEM** サンプルングが可能です。統計ビューがこれらの条件を満たしていない場合、**BERNOULLI** サンプルングが代わりに使用され、警告が戻されます。

#### *numeric-literal*

**numeric-literal** パラメーターでは、入手するサンプルのサイズを指定します ( $P$  %)。この値は 100 以下の正数でなければならず、1 と 0 の間の数を指定することもできます。例えば、値 0.01 は、1 % の 100 分の 1 を表します。この場合は、平均して 10,000 行のうちの 1 行がサンプルとして取られます。0 または 100 の値を指定した場合、DB2 データベース・システムでは、**TABLESAMPLE BERNOULLI** と **TABLESAMPLE SYSTEM** のどちらが指定されているかにかかわらず、サンプルングが指定されていない場合と同じように処理されます。100 より大きい値または 0 より小さい値は、DB2 データベース・システムではエラー (SQL1197N) として処理されます。



**REPEATABLE** (*integer-literal*)

**REPEATABLE** 節を **TABLESAMPLE** 節に追加すれば、**RUNSTATS** の反復実行時に必ず同じサンプルが戻されるようになります。 *integer-literal* パラメーターは、サンプリングで使用するシードを表す負以外の整数です。負のシードを渡した場合、エラー (SQL1197N) が発生します。 **TABLESAMPLE REPEATABLE** の最後の実行以降に行われた表または統計ビューに対する活動によって表または統計ビューのデータが変更された場合には、反復可能な **RUNSTATS** 呼び出しでサンプル・セットが変化する可能性があります。また、必ず整合した結果を得るためには、**BERNOULLI** (ベルヌーイ) または **SYSTEM** (システム) キーワードによって指定するサンプルの入手方法が同じでなければなりません。

**INDEXSAMPLE BERNOULLI**

このオプションを使用すると、索引内の行のサンプルに対する索引統計が収集されます。**BERNOULLI** (ベルヌーイ) サンプリングでは各行が個別に処理され、その際に  $P/100$  ( $P$  は数値リテラル値) の確率で行が含まれ、 $1-P/100$  の確率で行が除外されます。例えば、数値リテラルが値 10 (つまり、10 % のサンプル) と評価された場合は、0.1 の確率で各行が含まれ、0.9 の確率で行が除外されます。**RUNSTATS** は実行されるたびに、索引から異なるサンプルを得る結果になると考えられます。すべての索引ページが索引スキャンによって検索されますが、数値リテラル・パラメーターによって指定したパーセンテージの行だけが、統計収集に使用されます。このオプションは統計ビューではサポートされていません。

**INDEXSAMPLE SYSTEM**

このオプションを使用すると、索引ページのサンプルに対する統計が収集されます。**SYSTEM** (システム) サンプリングでは各ページが個別に処理され、その際に  $P/100$  ( $P$  は数値リテラル値) の確率でページが含まれ、 $1-P/100$  の確率でページが除外されます。通常は、**RUNSTATS** コマンドが実行されるたびに索引から異なるサンプルを得る結果になります。サンプルのサイズは、括弧内の数値リテラル・パラメーターによって制御し、索引の約  $P$  % を戻すように指定します。数値リテラル・パラメーターによって指定したパーセントの索引ページだけが、検索されて統計収集に使用されます。このオプションは統計ビューではサポートされていません。

**SET PROFILE NONE**

この **RUNSTATS** 呼び出しには統計プロファイルを設定しないことを指定します。

**SET PROFILE**

**RUNSTATS** は、特定の統計プロファイルを生成してシステム・カタログ表に保管し、**RUNSTATS** コマンド・オプションを実行して統計を収集します。

**SET PROFILE ONLY**

**RUNSTATS** は、特定の統計プロファイルを生成してシステム・カタログ表に保管しますが、**RUNSTATS** のコマンド・オプションを実行しません。

**UPDATE PROFILE**

**RUNSTATS** は、システム・カタログ表内の既存の統計プロファイルを変更し、その更新済みの統計プロファイルの **RUNSTATS** コマンド・オプションを実行して統計を収集します。UPDATE PROFILE オプションを使用して、統計プロファイルにある節を削除することはできません。

**UPDATE PROFILE ONLY**

**RUNSTATS** は、システム・カタログ表内の既存の統計プロファイルを変更しますが、その更新済みの統計プロファイルの**RUNSTATS** コマンド・オプションを実行しません。UPDATE PROFILE ONLY オプションを使用して、統計プロファイルにある節を削除することはできません。

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

*priority* に指定されているレベルで、**RUNSTATS** をスロットルすることを指定します。*priority* は 1 から 100 の範囲の数であり、100 が最高の優先順位、1 が最低の優先順位を表します。優先順位によって、ユーティリティのスロットルの量が決まります。優先順位が同じユーティリティはすべて同じ量のスロットルになり、優先順位の低いユーティリティは、優先順位の高いユーティリティよりも量が絞られます。*priority* を指定しない場合、**RUNSTATS** はデフォルトの優先順位 50 を使用します。

**UTIL\_IMPACT\_PRIORITY** キーワードを省略すると、スロットルのサポートなしで **RUNSTATS** ユーティリティが呼び出されます。

**UTIL\_IMPACT\_PRIORITY** キーワードを指定した場合でも、**util\_impact\_lim** 構成パラメーターが 100 に設定されていれば、スロットルなしでユーティリティが実行されます。

パーティション・データベースでは、**RUNSTATS** コマンドが表に対して使用された場合、1 つのデータベース・パーティションでしか統計を収集できません。**RUNSTATS** コマンドが実行されたデータベース・パーティションに表のパーティションがある場合、コマンドは、そのデータベース・パーティションで実行されます。それ以外の場合は、表がパーティションに分けられているデータベース・パーティション・グループの最初のデータベース・パーティションで実行されます。

**例**

1. 表でのみ統計を収集し、どの列でも分散統計は収集しません。

```
RUNSTATS ON TABLE employee
```

2. 表でのみ統計を収集し、列 **empid** と **empname** で分散統計を収集します。

```
RUNSTATS ON TABLE employee
WITH DISTRIBUTION ON COLUMNS (empid, empname)
```

3. 構成の設定から **num\_quantiles** を取り出しながら、表に指定された度数のしきい値を使用して、表でのみ、すべての列で分散統計を収集します。

```
RUNSTATS ON TABLE employee WITH DISTRIBUTION DEFAULT
NUM_FREQVALUES 50
```

4. 表でのみ統計を収集し、すべての列、および列グループ **JOB**、**WORKDEPT**、および **SEX** で分散統計を収集します。

```
RUNSTATS ON TABLE employee ON ALL COLUMNS AND COLUMNS ((JOB, WORKDEPT, SEX))
WITH DISTRIBUTION
```

5. 一連の索引で統計を収集します。

```
RUNSTATS ON TABLE employee for indexes emp11, emp12
```

6. すべての索引のみの統計を収集します。

```
RUNSTATS ON TABLE employee FOR INDEXES ALL
```

7. 詳細な索引統計収集のサンプリングを使用して、表とすべての索引に関する基本統計を収集します。

```
RUNSTATS ON TABLE employee AND SAMPLED DETAILED INDEXES ALL
```



これは、以下と同じ意味になります。

```
RUNSTATS ON TABLE employee AND DETAILED INDEXES ALL
```

8. 表の統計を収集し、列 `empid`、`empname`、`empdept`、および索引 `Xempid` および `Xempname` では分散統計を収集します。 `empdept` に関しては個別に分散統計のしきい値を設定し、その他の 2 つの列には共通のデフォルトを使用します。

```
RUNSTATS ON TABLE employee
  WITH DISTRIBUTION ON COLUMNS (empid, empname, empdept NUM_FREQVALUES
    50 NUM_QUANTILES 100)
  DEFAULT NUM_FREQVALUES 5 NUM_QUANTILES 10
  AND INDEXES Xempid, Xempname
```

9. 索引で使用されるすべての列と、すべての索引の統計を収集します。

```
RUNSTATS ON TABLE employee ON KEY COLUMNS AND INDEXES ALL
```

10. すべての索引とすべての列で統計を収集します。分散統計は、1 つの列を除いて収集しません。 `T1` には列 `c1`、`c2`、... `c8` が含まれているとと考えてください。

```
RUNSTATS ON TABLE 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 T1
  WITH DISTRIBUTION ON COLUMNS (c3 NUM_FREQVALUES 20 NUM_QUANTILES 40)
  AND INDEXES ALL
```

11. 表 `T1` で、個別の列 `c1` および `c5` と、列グループ (`c2`, `c3`) および (`c2`, `c4`) に関する統計を収集します。複数列のカーディナリティーは、照会オプション `OPTIMIZER` が、データの相関がある列の述部のフィルター係数を見積もるのに大変便利です。

```
RUNSTATS ON TABLE T1 ON COLUMNS (c1, (c2, c3),
  (c2, c4), c5)
```

12. 表 `T1` で、個別の列 `c1` および `c2` に関する統計を収集します。列 `c1` に関しては、`LIKE` 述部の統計も収集します。

```
RUNSTATS ON TABLE T1 ON COLUMNS (c1 LIKE STATISTICS, c2)
```

13. 構成の設定から `num_quantiles` を取り出しながら、表に指定された度数のしきい値を使用して、表でのみ、すべての列で分散統計を収集する統計プロファイルを登録します。このコマンドはさらに、指定のとおり統計を更新します。

```
RUNSTATS ON TABLE employee WITH DISTRIBUTION DEFAULT
  NUM_FREQVALUES 50 SET PROFILE
```

14. 構成の設定から `num_quantiles` を取り出しながら、表に指定された度数のしきい値を使用して、表でのみ、すべての列で分散統計を収集する統計プロファイルを登録します。統計は収集されません。

```
RUNSTATS ON TABLE employee WITH DISTRIBUTION
  DEFAULT NUM_FREQVALUES 50 SET PROFILE ONLY
```

15. `NUM_FREQVALUES` 値を 50 から 30 に変更することにより、以前に登録した統計プロファイルを変更します。このコマンドはさらに、指定のとおり統計を更新します。

```
RUNSTATS ON TABLE employee WITH DISTRIBUTION
  DEFAULT NUM_FREQVALUES 30 UPDATE PROFILE
```

16. `NUM_FREQVALUES` 値を 50 から 30 に変更することにより、以前に登録した統計プロファイルを変更します。統計は収集されません。

```
RUNSTATS ON TABLE employee WITH DISTRIBUTION
  DEFAULT NUM_FREQVALUES 30 UPDATE PROFILE ONLY
```

17. 列 `empl_address` と列グループ (`empl_title`, `empl_salary`) のオプションを追加することにより、以前に登録した統計プロファイルを変更します。このコマンドはさらに、指定のとおり統計を更新します。

```
RUNSTATS ON TABLE employee
  ON COLUMNS (empl_address, (empl_title, empl_salary))
  UPDATE PROFILE
```

18. 列 `empl_address` と列グループ (`empl_title`, `empl_salary`) のオプションを追加することにより、以前に登録した統計プロファイルを変更します。統計は収集されません。

```
RUNSTATS ON TABLE employee
  ON COLUMNS (empl_address, (empl_title, empl_salary))
  UPDATE PROFILE ONLY
```

19. 表の統計プロファイルに記録されたオプションを使用して、その表の統計を収集します。

```
RUNSTATS ON TABLE employee USE PROFILE
```

20. 表のカタログに保管されている以前に登録した統計プロファイルに対応する **RUNSTATS** コマンド・オプションを照会します。

```
SELECT STATISTICS_PROFILE FROM SYSCAT.TABLES WHERE TABNAME =
  'EMPLOYEE'
```

21. 30 % の行に関する分散統計などの統計を収集します。

```
RUNSTATS ON TABLE employee WITH DISTRIBUTION
  TABLESAMPLE BERNOULLI(30)
```

22. 統計を収集するためのサンプル・セットを制御し、同じサンプル・セットを繰り返し使用するには、以下のようにします。

```
RUNSTATS ON TABLE employee WITH DISTRIBUTION
  TABLESAMPLE BERNOULLI(30) REPEATABLE(4196)
```

上記のステートメントを実行すると、その間にデータが変更されていない限り、同じセットの統計が作成されます。

23. 1.5 % のデータ・ページに関する索引統計と表統計を収集します。表データ・ページだけがサンプルになり、索引ページはサンプルになりません。この例では、表統計の収集に 1.5 % の表データ・ページが使用され、索引統計にすべての索引ページが使用されます。

```
RUNSTATS ON TABLE employee AND INDEXES ALL TABLESAMPLE SYSTEM(1.5)
```

24. 1.5% のデータ・ページに関する表統計と、2.5% の索引ページに関する索引統計を収集します。表データ・ページと索引ページの両方がサンプルになります。

```
RUNSTATS ON TABLE employee AND INDEXES ALL TABLESAMPLE SYSTEM(1.5)
  INDEXSAMPLE SYSTEM(2.5)
```

25. 統計ビューの統計情報を、すべての列について、分散統計なしで収集します。

```
RUNSTATS ON VIEW product_sales_view
```

26. 統計ビューの統計情報を、列 `category`、`type`、および `product_key` に関する分散統計を含めて収集します。 `category` 列に関しては分散統計のしきい値を設定し、その他の列には共通のデフォルトを使用します。

```
RUNSTATS ON VIEW product_sales_view
WITH DISTRIBUTION ON COLUMNS (category NUM_FREQVALUES 100 NUM_QUANTILES 100,
type, product_key) DEFAULT NUM_FREQVALUES 50 NUM_QUANTILES 50
```

27. 行レベルのサンプリングを使用して、10 % の行に関する分散統計など、統計ビュー用の統計情報を収集します。

```
RUNSTATS ON VIEW product_sales_view
WITH DISTRIBUTION TABLESAMPLE_BERNOULLI (10)
```

28. データ・ページ・レベルのサンプリングを使用して、2.5% の行に関する分散統計など、統計ビュー用の統計情報を収集します。さらに、同じサンプル・セットを反復使用することを指定します。このコマンドが正常に実行されるためには、DB2 データベース・システムがデータ・ページ・サンプリングを 1 つ以上の表にプッシュする操作を正常に実行できるような照会でなければなりません。それ以外の場合、エラー (SQL 20288N) が発行されます。

```
RUNSTATS ON VIEW product_sales_view
WITH DISTRIBUTION TABLESAMPLE_SYSTEM (2.5)
```

29. ビューの統計情報を収集し、指定のとおりすべての列で分散統計を収集するための、統計プロファイルを登録します。

```
RUNSTATS ON VIEW product_sales_view
WITH DISTRIBUTION DEFAULT NUM_FREQVALUES 50 NUM_QUANTILES 50
SET PROFILE
```

30. 以前に登録した統計プロファイルを変更します。このコマンドはさらに、指定のとおり統計を更新します。

```
RUNSTATS ON VIEW salesdb.product_sales_view
WITH DISTRIBUTION DEFAULT NUM_FREQVALUES 25 NUM_QUANTILES 25
UPDATE PROFILE
```

## 使用上の注意

- パーティション表にデタッチされたパーティションが存在する場合、デタッチされたデータ・パーティションでクリーンアップの必要なものにまだ属している索引キーは、統計においてキーの一部としてカウントされません。それらのキーは、不可視であり表の一部ではなくなっているため、カウントされません。そのような索引キーは、最終的に非同期の索引クリーンアップによって索引から除去されます。その結果、非同期の索引クリーンアップを実行する前に収集された統計は誤ったものとなります。非同期索引クリーンアップの完了前に **RUNSTATS** コマンドが発行された場合、不正確な統計情報を基に、索引再編成または索引クリーンアップに対する誤ったアラームが生成されることがしばしばあります。非同期の索引クリーンアップの実行を開始すると、クリーンアップを必要とするデタッチされたデータ・パーティションにまだ属しているすべての索引キーが除去されるので、索引の再編成の必要がなくなることもあります。

パーティション表の場合、非同期索引クリーンアップ完了後に **RUNSTATS** コマンドを発行することをお勧めします。それは、デタッチされているデータ・パーティションの存在に関して正確な索引統計情報を生成するためです。表の中にデタッチされているデータ・パーティションがあるかどうかを調べるには、**SYSCAT.DATAPARTITIONS** カタログ・ビューの状況フィールドを確認して、値 L (論理的にデタッチ済み)、I (索引クリーンアップ)、または D (デタッチ済みで従属 MQT 付き) を探してください。

**RUNSTATS** コマンドは、パーティション索引のすべての索引パーティションの統計を収集します。パーティション索引の **SYSSTAT.INDEXES** ビュー内の統計は、索引パーティションを表します。ただし

**FIRSTKEYCARD**、**FIRST2KEYCARD**、**FIRST3KEYCARD**、**FIRST4KEYCARD**、および **FULLKEYCARD** 統計は除きます。これらの統計はカーディナリティーの見積もりで使用されるので、それらは索引全体に対するものであり、1 つの索引パーティションに対するものではありません。分散統計 (頻度および変位値) は、パーティション索引では収集されませんが、**RUNSTATS** が表上で実行される場合は収集されます。パーティション索引の先行列の統計は、非パーティション索引の先行列の統計ほど正確でない場合があります。

2. **RUNSTATS** コマンドは、次のような場合に実行することが勧められています。

- 表が大幅に変更されている場合 (例えば、多くの変更が行われている場合や、大量のデータが挿入または削除されている場合、あるいは、**LOAD** 時に統計オプションを指定しないで **LOAD**が行われた場合など)。
- 表が再編成されている場合 (**REORG**、**REDISTRIBUTE DATABASE PARTITION GROUP** を使用)。
- 行圧縮が実行された表の場合。
- 新しい索引が作成されている場合
- パフォーマンスが重要な意味を持つアプリケーションのバインドの前。
- プリフェッチ・サイズが変更された場合。
- 基礎表に実質的な変更が加えられたことにより、ビューによって戻される行が変更された統計ビューの場合。
- **STATISTICS** オプションを指定した **LOAD** が実行された後には、**RUNSTATS** コーティリティーを使用して、XML 列に関する統計情報を収集してください。 **LOAD** 実行時には、**LOAD** に **STATISTICS** オプションを指定して実行された場合であっても、XML 列に関する統計情報が収集されることは決してありません。 **RUNSTATS** を使用することにより、XML 列についてのみの統計情報を収集する場合、XML 以外の列に関して **LOAD** またはそれ以前に実行された **RUNSTATS** コーティリティーによって収集された既存の統計情報は、そのまま保持されます。以前に一部の XML 列に関する統計情報が収集されていた場合、ある XML 列に関する統計情報が現在のコマンドでは収集されないのであれば、その XML 列に関して以前に収集された統計情報はドロップされます。あるいは、その XML 列に関する統計が現在のコマンドで収集されるのであれば、置き換えられます。

3. オプションの選択は、特定の表やアプリケーションに合わせて行う必要があります。一般的なヒントとして、以下の点を考慮してください。

- 重要な照会に使用される非常に重要な表、比較的小規模な表、またはあまり変化がなく、システムそのものでの活動があまりない表には、可能な限り詳細に統計を収集する努力を費やす価値があります。
- 統計を収集する時間が限られている場合、表が比較的大規模な場合、または表が頻繁に更新される場合には、述部で使用される列セットに限って **RUNSTATS** を実行するのも良い方法かもしれません。このような方法を使用する場合には、より頻繁に **RUNSTATS** コマンドを実行できるでしょう。
- 統計を収集する時間が極めて限られており、表ごとに表の **RUNSTATS** コマンドを調整するのが時間の面で大きな問題となっている場合は、"KEY" 列だけ

の統計を収集することも考慮してください。索引に含まれている列セットは、表にとって重要で、述部に使用される確立が最も高いと考えられます。

- 統計を収集する時間が非常に限られている状況で表統計を収集する場合は、**TABLESAMPLE** オプションを使用して、表データのサブセットに関する統計を収集することを検討してください。
  - 統計を収集する時間が非常に限られている状況で索引統計を収集する場合は、**INDEXSAMPLE** オプションを使用して、索引データのサブセットに関する統計を収集することを検討してください。
  - 特定の列にスキューがあり、述部のタイプが "column = constant" である場合、その列にはより大きな **NUM\_FREQVALUES** 値を指定するほうが良い可能性があります。
  - 等式の述部で使用される列や、値の分散がスキューされる可能性のある列では、必ず分散統計を収集してください。
  - 範囲の述部を持つ列 (例えば、"column >= constant" や "column BETWEEN constant1 AND constant2" など) や、タイプ "column LIKE '%xyz'" の列では、**NUM\_QUANTILES** に指定する値を大きくしたほうが有益な場合があります。
  - ストレージ・スペースが関係している場合で、統計の収集にあまり時間をかけられない場合は、述部で使用されない列の **NUM\_FREQVALUES** 値や **NUM\_QUANTILES** 値をあまり高くしないでください。
  - 索引統計を要求したときに、索引を含む表についての統計がそれまで実行されていなかった場合、表と索引の両方に関する統計が計算されます。
  - 表に含まれる XML 列に関する統計情報が必要ない場合は、**EXCLUDING XML COLUMNS** オプションを使用することによって、XML 列をすべて除外することができます。このオプションは、統計情報収集の対象として XML 列を指定する他のどの節よりも優先されます。
4. コマンドを実行した後は、以下の点に注意してください。
- ロックを解除するには、**COMMIT** を発行する必要があります。
  - 新しいアクセス・プランを生成できるようにするには、ターゲット表を参照するパッケージを再バインドする必要があります。
  - 表で部分的にコマンドを実行すると、コマンドが最後に実行されてからの表での活動の結果として、不整合が生じる可能性があります。このような場合には、警告メッセージが戻されます。表でだけ **RUNSTATS** が実行されると、表レベルの統計と索引レベルの統計に不整合が生じます。例えば、ある表に関して索引レベルの統計を収集した後で、その表からかなりの数の行を削除してしまったとします。このような場合に、その表でだけ **RUNSTATS** を発行すると、表のカーディナリティーが **FIRSTKEYCARD** よりも小さくなってしまふ可能性があります。これは不整合です。これと同様に、作成した新しい索引で統計を収集した場合にも、表レベルの統計に不整合が生じることがあります。
5. **RUNSTATS** コマンドは、表統計が要求したときに、以前に収集された分散統計をドロップします。例えば、**RUNSTATS ON TABLE** または **RUNSTATS ON TABLE ... AND INDEXES ALL** は、以前に収集された分散統計がドロップされる原因になります。コマンドが索引でのみ実行される場合、以前に収集された分散統計は保持されます。例えば、**RUNSTATS ON TABLE ... FOR INDEXES ALL** は、以前に収



集された分散統計が保持される原因になります。 **RUNSTATS** コマンドが XML 列に対してのみ実行される場合、それ以前に収集された基本列統計および分散統計はそのまま保持されます。以前に一部の XML 列に関する統計情報が収集されていた場合、ある XML 列に関する統計情報が現在のコマンドでは収集されないのであれば、その XML 列に関して以前に収集された統計情報はドロップされます。あるいは、その XML 列に関する統計が現在のコマンドで収集されるのであれば、置き換えられます。

6. DB2 バージョン 9.7 フィックスパック 1 以降のリリースでは、XML 列に定義された XML データに対する索引で、分散統計が収集されます。表に対して、**RUNSTATS** コマンドを **WITH DISTRIBUTION** 節付きで実行する場合、タイプ XML の列に関する分散統計の収集は以下のようになります。

- 分散統計は、XML 列で指定されている XML データに対する各索引に関して収集されます。
- **RUNSTATS** コマンドでは、XML 列で定義された XML データに対する索引に関して分散統計を収集するため、分散統計と表統計の両方を収集する必要があります。XML 分散統計は表統計と共に保管されるので、分散統計を収集するためには表統計を収集しなければなりません。

XML 分散統計を収集するために、索引節は必要ありません。索引節だけを指定しても、XML 分散統計は収集されません。

XML 分散統計はデフォルトで、XML データに対する各索引に関して最大 250 の分位数を使用します。XML 列に対する分散統計を収集する場合、**ON COLUMNS** 節または **DEFAULT** 節の **NUM\_QUANTILES** パラメーターに値を指定することによって分位の最大数を変更することができます。

- 分散統計は、タイプ VARCHAR、DOUBLE、TIMESTAMP、および DATE の XML データの索引に関して収集されます。タイプ VARCHAR HASHED の索引に関して、分散統計は収集されません。
  - 分散統計は、パーティション表で定義された XML データに対するパーティション索引に関しては収集されません。
7. 範囲がクラスター化された表の場合、範囲がクラスター化された表の範囲配列プロパティを表す特殊なシステム生成索引がカタログ表内に存在します。この種の表の統計を収集するときに、統計収集の一部として表を組み込む場合は、システム生成索引用の統計も収集されます。この統計は、基本データ表と同じページ数を持つ 2 レベルの索引として索引を表現し、索引の順序に沿って完全に基本データをクラスター化することによって、範囲検索の高速アクセスを反映することになっています。
8. コマンド構文の **On Dist Cols** 節では、列 GROUPS に対する **Frequency Option** および **Quantile Option** のパラメーターの使用は、現在サポートされていません。これらのオプションは、単一の列でのみサポートされています。
9. DMS モードでの作業中に、計算できない 3 つのプリフェッチ統計があります。索引カタログ内の索引統計で、以下の統計の値は -1 になります。
- AVERAGE\_SEQUENCE\_FETCH\_PAGES
  - AVERAGE\_SEQUENCE\_FETCH\_GAP
  - AVERAGE\_RANDOM\_FETCH\_PAGES

10. プロファイルの設定またはプロファイルの更新のオプションを使用することによって、**RUNSTATS** コマンドで指定する表または統計ビューの統計プロファイルを設定または更新できます。統計プロファイルは、**STATISTICS\_PROFILE** システム・カタログ表の **SYSCAT.TABLES** 列に、可視ストリングのフォーマットで保管されます。これが **RUNSTATS** コマンドに相当します。
11. XML タイプの列に関する統計情報収集は、**DB2\_XML\_RUNSTATS\_PATHID\_K** および **DB2\_XML\_RUNSTATS\_PATHVALUE\_K** の 2 つの DB2 データベース・システム・レジストリー値によって制御されます。これらの 2 つのパラメーターは、収集する度数の数を指定するという点において **NUM\_FREQVALUES** パラメーターに似ています。設定されていない場合、どちらのパラメーターについてもデフォルトとして 200 が使用されます。
12. **RUNSTATS** は、開始時に、**SYSTABLES** に対して IX 表ロックを取得し、統計収集の対象となる表の行に対して U ロックを取得します。U ロックのかかった行を含め、**SYSTABLES** からの読み取り操作は可能です。また、U ロックのかかった行でない限り、書き込み操作も可能です。しかし、**RUNSTATS** が IX ロックを取得しているため、別の読み取りプログラムまたは書き込みプログラムが **SYSTABLES** に対する S ロックを取得することはできません。
13. 統計は、構造化タイプ of 列については収集されません。それらが指定されている場合、そのデータ・タイプの列は無視されます。
14. LOB または LONG データ・タイプの列では、AVGCOLLEN および NUMNULLS のみが収集されます。
15. AVGCOLLEN は、列がデータベース・メモリーまたは一時表に保管される場合の、バイト単位の平均スペースを表します。この値は、LOB または LONG データ・タイプのデータ記述子の長さを表します。ただし LOB データがデータ・ページ上でインライン化されている場合を除きます。

注: ディスク上で列を保管するための平均スペース所要量は、この統計により表される値とは異なる場合があります。

16. 索引統計の収集方法を変更するため、**UNSAMPLED DETAILED** オプションを使用できます。ただし、これを使用するのは、デフォルトと **DETAILED** では適正に機能しないことが明らかな場合に限るべきです。
17. **INDEXSAMPLE** キーワードを使用するときは、単一のコマンド内で索引に対する索引サンプリング率を別々に指定することはできません。例:

```
runstats on table orders and index o_ck indexsample system(5),
      index o_ok indexsample system(10)
```

この指定は無効です。必要な結果を得るには、以下の 2 つの **RUNSTATS** コマンドを使用できます。

```
runstats on table orders and index o_ck indexsample system(5)
runstats on table orders for index o_ok indexsample system(10)
```



## SET CLIENT

バックエンド・プロセス用の接続設定を指定します。

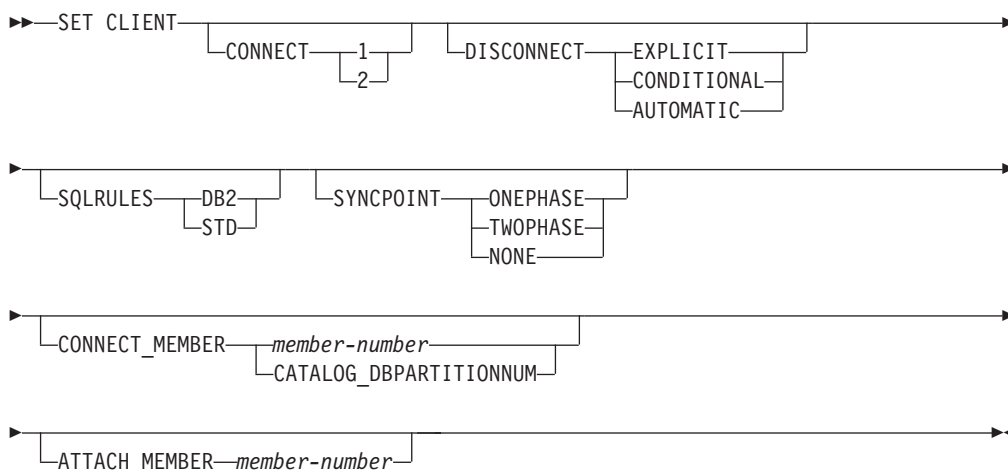
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### CONNECT

- 1 CONNECT ステートメントをタイプ 1 の CONNECT として処理するよう指定します。
- 2 CONNECT ステートメントをタイプ 2 の CONNECT として処理するよう指定します。

#### DISCONNECT

##### EXPLICIT

RELEASE ステートメントで明示的に解放をマークしたデータベース接続だけを、コミット時に切断するよう指定します。

##### CONDITIONAL

RELEASE をマークしたか、またはオープン状態の WITH HOLD カーソルをもたないデータベース接続を、コミット時に切断するよう指定します。

##### AUTOMATIC

コミット時にすべてのデータベース接続を切断するよう指定します。

#### SQLRULES

- DB2** タイプ 2 CONNECT が、DB2 規則に従って処理されることを指定します。
- STD** タイプ 2 CONNECT が、ISO/ANSI SQL92 に基づく標準 (STD) 規則に従って処理されることを指定します。

**SYNCPOINT**

複数のデータベース接続にまたがってコミットまたはロールバックを調整する仕方を指定します。このコマンド・パラメーターは無視されます。ここでは、後方互換性のためにこれを掲載しています。

**ONEPHASE**

2 フェーズ・コミットを実行するのに使用されるトランザクション・マネージャー (TM) がないことを指定します。複数のデータベース・トランザクションの各データベースが行う作業をコミットするときは、1 フェーズ・コミットが使用されます。

**TWOPHASE**

このプロトコルをサポートする複数のデータベースにまたがって 2 フェーズ・コミットを調整するのに TM が必要であることを指定します。

- NONE** 2 フェーズ・コミットを実行するのに使用される TM がなく、単一の更新プログラムと複数の読み込みプログラムという形を強制しないことを指定します。コミットは、関連する各データベースに送られます。コミットが失敗したときのリカバリーは、アプリケーションが行います。

**CONNECT\_MEMBER** (パーティション・データベースまたは DB2 pureScale 環境)*member-number*

接続が確立される先のメンバーを指定します。パーティション・データベース環境では、有効な値は 0 から 999 まで (両端を含む) で、DB2 pureScale 環境では、有効な値は 0 から 127 まで (両端を含む) です。DB2NODE 環境変数の値をオーバーライドします。

**CATALOG\_DBPARTITIONNUM**

パーティション・データベース環境では、この値を指定すると、クライアントはデータベース・パーティションの ID をあらかじめ認識していなくても、そのデータベースのカタログ・データベース・パーティションに接続できます。DB2 pureScale 環境では、このオプションの指定は許可されません (SQLSTATE 56038)。

**ATTACH\_MEMBER** *member-number* (パーティション・データベースまたは DB2 pureScale 環境)

アタッチが確立される先のメンバーを指定します。ATTACH\_MEMBER への変更を有効にするために、まず、DETACH コマンドを使用してインスタンスからデタッチし、次に ATTACH コマンドを使用してインスタンスにアタッチしてください。パーティション・データベース環境では、有効な値は 0 から 999 まで (両端を含む) で、DB2 pureScale 環境では、有効な値は 0 から 127 まで (両端を含む) です。DB2NODE 環境変数の値をオーバーライドします。

## 例

特定の値を設定するには、次のようになります。

```
db2 set client connect 2 disconnect automatic sqlrules std
syncpoint twophase
```

**SQLRULES** を DB2 規則に戻し、それ以外の設定はそのままにしておくには、次のようにします。

```
db2 set client sqlrules db2
```

接続の設定内容は、**TERMINATE** コマンドが出された後で、デフォルト値に戻ります。

## 使用上の注意

**SET CLIENT** は、アクティブな接続が 1 つ以上あると発行できません。

**SET CLIENT** が成功すると、後続の作業単位の接続は指定された接続の設定を使用します。**SET CLIENT** が異常であると、バックエンド・プロセスの接続設定内容が、未変更のままになります。

パーティション・データベースまたは DB2 pureScale 環境においては、この接続の設定はトラステッド接続の取得に影響を与えます。例えば、あるノードに **CONNECT\_MEMBER** オプションが設定されており、そのノードへの接続を確立するのに、中間ノード (ホップ・ノード) を通過する必要がある場合、トラステッド接続としてマークできるかどうかを判断します。その判断には、この中間ノードの IP アドレスと、ホップ・ノードと接続先ノード間の通信に使用される通信プロトコルが考慮されます。つまり、接続元となるノードは考慮されません。むしろ、ホップ・ノードが考慮されます。

## 互換性

以前のバージョンとの互換性:

- **DB2\_ENFORCE\_MEMBER\_SYNTAX** レジストリー変数が ON に設定されている場合を除き、**CONNECT\_DBPARTITIONNUM** または **CONNECT\_NODE** を **CONNECT\_MEMBER** の代わりに使用できます。
- **CATALOG\_NODE** は、**CATALOG\_DBPARTITIONNUM** の代わりとして使用できます。
- **DB2\_ENFORCE\_MEMBER\_SYNTAX** レジストリー変数が ON に設定されている場合を除き、**ATTACH\_DBPARTITIONNUM** または **ATTACH\_NODE** を **ATTACH\_MEMBER** の代わりに使用できます。



パーティション内並列処理のランタイム・レベルは、ステートメントが実行される  
ときに使用される並列処理の最大数を指定します。SQL ステートメントのパー  
ティション内並列処理のレベルは、ステートメントのコンパイル時に、**CURRENT  
DEGREE** 特殊レジスターまたは **DEGREE BIND** オプションを使用して指定するこ  
とができます。アクティブ・アプリケーションのパーティション内並列処理のラン  
タイムの最大並列度は、**SET RUNTIME DEGREE** コマンドを使用して指定するこ  
とができます。**max\_querydegree** データベース・マネージャ 構成パラメータは、デ  
ータベース・マネージャのこのインスタンスを実行する SQL ステートメントの  
ランタイムの最大並列度を指定します。

実際のランタイム並行性レベルは次の 3 つのうち、最も低いものです。

- **max\_querydegree** 構成パラメータ
- アプリケーションのランタイム並行性レベル
- SQL ステートメントのコンパイル度

**CURRENT DEGREE** 特殊レジスターの値および **intra\_parallel** 設定値は、ワー  
クロードで **MAXIMUM DEGREE** ワークロード属性の設定によってオーバーライドさ  
れます。

## SET SERVEROUTPUT

DBMS\_OUTPUT メッセージ・バッファの出力を標準出力にリダイレクトするかどうかを指定します。

### 許可

DBMS\_OUTPUT モジュールに対する EXECUTE 特権。

### 必要な接続

データベース

### コマンド構文

```

▶▶ SET SERVEROUTPUT {OFF|ON}

```

### コマンド・パラメーター

**ON** メッセージ・バッファ内のメッセージを標準出力にリダイレクトすることを指定します。

#### OFF

メッセージ・バッファ内のメッセージを標準出力にリダイレクトしないことを指定します。

### 例

DBMS\_OUTPUT メッセージ・バッファ内のメッセージを標準出力にリダイレクトするには、SET SERVEROUTPUT ON を指定します。この例では、PUT プロシージャは DBMS\_OUTPUT メッセージ・バッファに部分的な行を追加します。proc1 を実行すると、SET SERVEROUTPUT ON が指定されているため、DBMS\_OUTPUT メッセージ・バッファに格納されているテキストが表示されます。

```

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@

```

この例では、以下の出力が結果として戻ります。

```

SET SERVEROUTPUT ON
DB20000I The SET SERVEROUTPUT command completed successfully.

DROP PROCEDURE PROC1
DB20000I The SQL command completed successfully.

```

```

CREATE PROCEDURE procl()
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 procl( 10, 'Peter' )@

  Return Status = 0

p1 = 10
p2 = Peter

SET SERVEROUTPUT OFF
DB20000I The SET SERVEROUTPUT command completed successfully.

```

## 使用上の注意

PUT、PUT\_LINE、および NEW\_LINE の各プロシージャによって DBMS\_OUTPUT メッセージ・バッファにメッセージが追加されます。

コマンド SET SERVEROUTPUT ON を実行すると、DBMS\_OUTPUT.ENABLE プロシージャがデフォルト・バッファ・サイズ 20000 バイトを使用して呼び出され、コマンド・ライン・プロセッサ (CLP) または Command Line Processor Plus (CLPPlus) に内部フラグが設定されます。このフラグが有効な場合には、アプリケーションは各 SELECT または CALL ステートメントの実行後に GET\_LINES プロシージャを呼び出し、メッセージ・バッファのメッセージを標準出力にリダイレクトします。DBMS\_OUTPUT バッファ・プール・サイズを増やすには、SET SERVER OUTPUT ON を実行した後により大きなバッファ・サイズを使用して DBMS\_OUTPUT.ENABLE プロシージャを呼び出します。例えば、以下のようになります。CALL DBMS\_OUTPUT.ENABLE( 50000 );

コマンド SET SERVEROUTPUT OFF が実行されると、DBMS\_OUTPUT.DISABLE プロシージャが呼び出され、メッセージ・バッファ内にあるメッセージが破棄され、PUT、PUT\_LINE、および NEW\_LINE プロシージャに対する呼び出しは無視されます。DBMS\_OUTPUT.GET\_LINES プロシージャが、各 SELECT または CALL ステートメントの後に呼び出されることはありません。



## SET TABLESPACE CONTAINERS

リダイレクトされたリストア操作中に、表スペース・コンテナを設定します。

リダイレクトされたリストアとは、リダイレクトされたデータベースのストレージ (表スペース・コンテナまたはストレージ・グループ・パス) が、バックアップが行われた時のソース・データベースのストレージとは異なっているリストアのことです。このコマンドを使用すれば、リダイレクトされたリストア操作中に、表スペース・コンテナの追加、変更、または除去を実行できます。例えば、1 つ以上のコンテナが何かの理由でアクセス不能となった場合、リストアが別の表スペース・コンテナにリダイレクトされない限りリストアは失敗します。

このコマンドを使用して、既存の REGULAR データベース管理表スペースまたは LARGE データベース管理表スペースを変換して自動ストレージを使用することができます。また、既存の自動ストレージ表スペースを、データベースで使用可能なストレージ・パスにより均等に再ストライピングするためにも使用できます。

### 許可

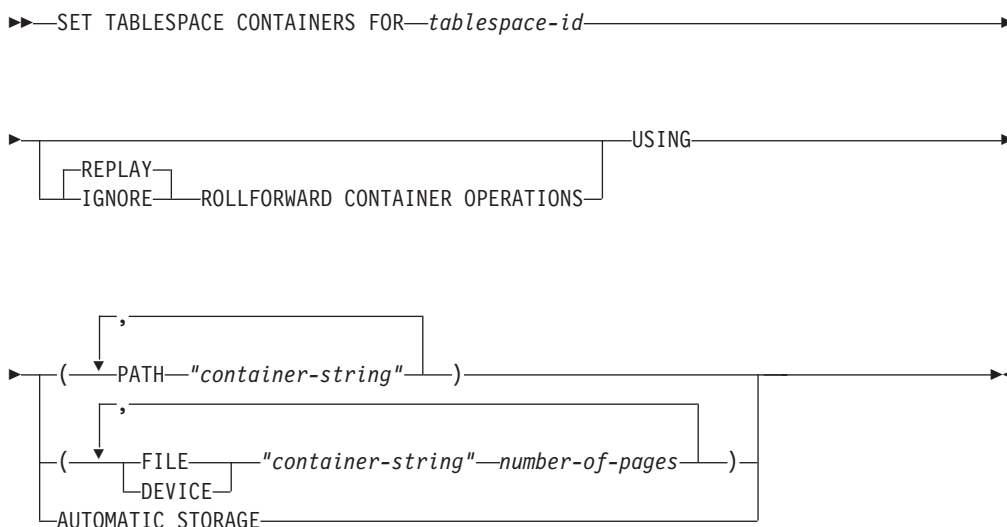
以下の権限のいずれか。

- SYSADM
- SYSCTRL

### 必要な接続

データベース

### コマンド構文



### コマンド・パラメーター

**FOR** *tablespace-id*

リストアされるデータベースが使用する表スペースを表す固有の整数。

**REPLAY ROLLFORWARD CONTAINER OPERATIONS**

データベースのバックアップが行われた後にこの表スペースに対して発行された ALTER TABLESPACE 操作が、それ以後のデータベースのロールフォワード処理中に再発行されることを指定します。

**IGNORE ROLLFORWARD CONTAINER OPERATIONS**

ロールフォワード中に、ログ内の ALTER TABLESPACE 操作が無視されることを指定します。

**USING PATH "container-string"**

SMS 表スペースに関して、表スペースに属し、表スペース・データの保管先となる、1 つ以上のコンテナを識別します。絶対または相対ディレクトリー名です。ディレクトリー名が絶対でない場合、データベース・ディレクトリーに対して相対的です。ストリング長は 240 バイト以下です。

**USING FILE | DEVICE "container-string" number-of-pages**

DMS 表スペースの場合、表スペースに属し、表スペース・データの保管先となる、1 つ以上のコンテナを識別します。コンテナ・タイプ (**FILE** または **DEVICE** のいずれか) およびそのサイズを指定します。FILE コンテナおよび DEVICE コンテナを混合して指定することができます。ストリング長は 254 バイト以下です。

FILE コンテナの場合、ストリングは絶対または相対ファイル名です。ファイル名が絶対ファイル名でない場合、データベース・ディレクトリーに対して相対的です。

DEVICE コンテナの場合、ストリングは装置名でなければなりません。また、装置が既に存在していなければなりません。

**USING AUTOMATIC STORAGE**

自動ストレージを使用するように表スペースが変換され、デフォルトのストレージ・グループに関連付けられるように指定します。データベースは、デフォルトのストレージ・グループのストレージ・パスに、新しいコンテナを作成することになります。表スペースが自動ストレージを使用するようにリダイレクトされると、表スペースにはコンテナ操作を適用できません。

このオプションを使用すると、既に自動ストレージによって管理されている表スペースのコンテナを再定義して、既存のストレージ・パスにおいてより良いストライピングを行えます。

注: 表スペースは、リストア中はオフラインになります。

このオプションは、システム管理表スペースではサポートされていません。

これは、DB2 pureScale 環境に指定できる唯一の **USING** 節です。

**例**

**RESTORE DATABASE** の例を参照してください。

**使用上の注意**

データベース、または 1 つ以上の表スペースのバックアップを取ることににより、バックアップしようとする表スペースが使用しているすべての表スペース・コンテナのレコードを維持できます。リストア中、バックアップにリストされたすべての

## SET TABLESPACE CONTAINERS

コンテナについて、現在存在しているかどうか、およびアクセス可能であるかどうかをチェックされます。何らかの理由で 1 つでもアクセス不能なコンテナがあると、リストアは失敗してしまいます。そのような場合であってもリストアを行えるようにするため、リストア中は表スペース・コンテナのリダイレクトがサポートされています。このサポートには、表スペース・コンテナの追加、変更、もしくは除去が含まれます。ユーザーはこのコマンドを使用して、これらのコンテナの追加、変更、または除去ができます。

**IGNORE/REPLAY ROLLFORWARD CONTAINER OPERATIONS** オプションは、**USING AUTOMATIC STORAGE** オプションが指定されている場合には無視されます。

**SET TABLESPACE CONTAINERS** コマンドの **USING AUTOMATIC STORAGE** オプションを使用して複数パーティション環境で表スペースのリダイレクト・リストアを行うと、リストア中のパーティションにある表スペースのみが自動ストレージに変換されます。別のデータベース・パーティション上のコンテナは再定義されません。

他のデータベース・パーティション上のコンテナは再定義されないため、表スペースの定義はパーティションごとに異なります。後からデータベース・パーティションを追加する場合、**LIKE DBPARTITIONNUM** オプションを指定して **ADD DBPARTITIONNUM** コマンドを使用します。このオプションで選択するデータベース・パーティションに応じて、新しいデータベース・パーティションには、自動ストレージを使用するように定義された表スペースまたは自動ストレージを使用しないように定義された表スペースのいずれかが含まれます。表スペースの定義間での不整合をなくし、新しいデータベース・パーティションが追加されるたびにどちらの定義にするか決定しなくて済むようにするために、表スペース定義がすべてのデータベース・パーティションで同じになるようにします。例えば、すべてのデータベース・パーティションをリダイレクト・リストアの対象とした後に、**SET TABLESPACE CONTAINERS** コマンドの **USING AUTOMATIC STORAGE** オプションを使用すると、すべてのデータベース・パーティション上にある表スペースが自動ストレージに変換されます。後から別のデータベース・パーティションを追加すると、他のデータベース・パーティションにある表スペースと同じ定義を持つこととなります。

パーティション・データベース環境では、**SET TABLESPACE CONTAINERS** コマンドによってストレージ・タイプを変更するリダイレクトで、1 つの表スペースがデータベース・パーティション・グループのサブセットに対してしかリダイレクトされなかった場合には、その表スペースが、一部のデータベース・パーティションでは **DMS** として、他のデータベース・パーティションでは自動ストレージとして、また別のデータベース・パーティションでは **DMS** と自動ストレージの組み合わせとして定義される可能性があります。

表スペースをリストアした後、続いて **ALTER TABLESPACE** ステートメントを実行して、カタログ・ビューの表スペースのストレージ・グループ **ID** を更新してください。

## SET TAPE POSITION

ストリーミング磁気テープ装置へのバックアップおよびリストア操作のためにテープの位置を設定します。このコマンドは Windows オペレーティング・システムでのみサポートされています。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMANT

### 必要な接続

なし

### コマンド構文

```
▶▶ SET TAPE POSITION [ON device] TO position ▶▶
```

### コマンド・パラメーター

#### ON *device*

有効なテープ装置名を指定します。デフォルト値は ¥¥.¥TAPE0 です。

#### TO *position*

テープ位置のマークを指定します。DB2 (Windows 版) は、バックアップ・イメージの度にテープ・マークを書き込みます。値 1 は 1 番目の位置、2 は 2 番目の位置、以下同じ手順で指定します。テープがテープ・マーク 1 に位置している場合、例えば、アーカイブ 2 がリストアされる位置に置かれます。

---

## SET UTIL\_IMPACT\_PRIORITY

実行中のユーティリティに関する影響設定を変更します。

このコマンドを使用して、以下を行うことができます。

- アンスロットル・モードで呼び出されたユーティリティにスロットルを設定します。
- スロットル・ユーティリティをアンスロットルします (スロットルを無効にします)。
- スロットル・ユーティリティの優先順位を再設定します (これは、複数のスロットル・ユーティリティを同時に実行している場合に便利です)。

### 有効範囲

#### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMOINT
- SYSMON

#### 必要な接続

インスタンス。ローカル・マシンに複数のパーティションがある場合、アタッチは正しいパーティションに行う必要があります。例えば、2 つのパーティションがあり、**LIST UTILITIES** コマンドで次の出力が得られたとします。

```
ID = 2
Type = BACKUP
Database Name = IWZ
Partition Number = 1
Description = online db
Start Time = 07/19/2007 17:32:09.622395
State = Executing
Invocation Type = User
Throttling:
Priority = Unthrottled
Progress Monitoring:
Estimated Percentage Complete = 10
Total Work = 97867649689 bytes
Completed Work = 10124388481 bytes
```

ID が 2 であるユーティリティに対して **SET UTIL\_IMPACT\_PRIORITY** コマンドを発行するには、インスタンスの接続はパーティション 1 に行う必要があります。これを行うには、環境に **DB2NODE=1** を設定してから、インスタンス接続コマンドを発行します。

#### コマンド構文

```
▶▶—SET UTIL_IMPACT_PRIORITY FOR—utility-id—TO—priority—▶▶
```

## コマンド・パラメーター

### *utility-id*

影響設定が更新されるユーティリティの ID。実行中のユーティリティの ID は、**LIST UTILITIES** コマンドで取得できます。

### **TO** *priority*

ユーティリティの実行に関連付けられている影響に関するインスタンス・レベルの制限を指定します。100 の値は、最も高い優先順位を表し、1 は最も低い優先順位を表します。*priority* を 0 に設定すると、スロットル・ユーティリティをアンスロットルして強制的に継続します。*priority* をゼロ以外の値に設定すると、アンスロットル・ユーティリティをスロットル・モードで継続します。

## 例

以下の例は、ID 2 のユーティリティをアンスロットルします。

```
SET UTIL_IMPACT_PRIORITY FOR 2 TO 0
```

以下の例は、ID 3 のユーティリティを優先順位 10 にスロットルします。変更前に優先順位が 0 だった場合、以前アンスロットルだったユーティリティはスロットルに設定されます。ユーティリティが以前スロットルだった (優先順位が 0 より大きい値に設定されていた) 場合、ユーティリティの優先順位が再設定されません。

```
SET UTIL_IMPACT_PRIORITY FOR 3 TO 10
```

## UTIL\_IMPACT\_LIM の設定と UTIL\_IMPACT\_PRIORITY の設定の関係

データベース・マネージャー構成パラメーター **util\_impact\_lim** は、スロットル・ユーティリティがマシンの全体的なワークロードに与える影響に制限を設定します。0-99 はスロットルのパーセンテージで、100 はスロットルなしです。

**SET UTIL\_IMPACT\_PRIORITY** コマンドは、スロットル・ユーティリティで使用できるリソース (**util\_impact\_lim** 構成パラメーターで定義される) に関して、特定のユーティリティが持つ優先順位を設定します。(0 = スロットルなし)。

例としてバックアップ・ユーティリティを取り上げます。**util\_impact\_lim=10** の場合、すべてのユーティリティがワークロード全体に与える影響は、スロットル・アルゴリズムを使用して評価された平均 10 % を超えてはなりません。例として、2 つのスロットル・ユーティリティを使用します。

- **util\_impact\_priority 70** のバックアップ
- **util\_impact\_priority 50** の RUNSTATS

両方のユーティリティを組み合わせた場合の合計ワークロードへの影響の平均は 10 % を超えず、優先順位の高いほうのユーティリティが、使用可能なワークロード・リソースをより多く獲得します。バックアップおよび **RUNSTATS** の操作は、両方とも各ユーティリティのコマンド行で影響の優先順位を宣言することもできます。**SET UTIL\_IMPACT\_PRIORITY** コマンドを発行しない場合、ユーティリティは (**util\_impact\_lim** の設定に関係なく) スロットルされない状態で実行します。

## SET UTIL\_IMPACT\_PRIORITY

実行中のユーティリティーに現在設定されている優先順位を表示するには、**LIST UTILITIES** コマンドを使用できます。

### 使用上の注意

スロットルを設定するには、**util\_impact\_lim** 構成パラメーターを設定して影響ポリシーを定義する必要があります。



## SET WORKLOAD

データベース接続の接続先として割り当てるワークロードを指定します。このコマンドは、データベースに接続する前に発行することができます。あるいは、接続が確立されてから現行接続の再割り当てをするために使用することができます。接続が確立されている場合は、次の作業単位の開始時にワークロードの再割り当てが行われます。

### 許可

なし。ただし、『使用上の注意』を参照してください。

### 必要な接続

なし

### コマンド構文

```

▶▶ SET WORKLOAD TO AUTOMATIC
SYSDEFAULTADMWORKLOAD

```

### コマンド・パラメーター

#### AUTOMATIC

サーバーが自動的に実行するワークロード計算に選ばれているワークロードに、データベース接続を割り当てるよう指定します。

#### SYSDEFAULTADMWORKLOAD

データベース接続を **SYSDEFAULTADMWORKLOAD** に割り当てて、*accessctrl* 権限、*dataaccess* 権限、*wlmadm* 権限、*secadm* 権限、または *dbadm* 権限を持つユーザーが通常のワークロード計算を迂回できるよう指定します。

### 例

接続を **SYSDEFAULTADMWORKLOAD** に割り当てる方法。

```
SET WORKLOAD TO SYSDEFAULTADMWORKLOAD
```

ワークロード割り当てをリセットして、サーバーが実行するワークロード計算に選ばれているワークロードを使用するようにする方法。

```
SET WORKLOAD TO AUTOMATIC
```

### 使用上の注意

データベース接続の SESSION 許可 ID に *accessctrl* 権限、*dataaccess* 権限、*wlmadm* 権限、*secadm* 権限、または *dbadm* 権限がない場合、接続を **SYSDEFAULTADMWORKLOAD** に割り当てることはできず、SQL0552N エラーが戻されます。SET WORKLOAD TO SYSDEFAULTADMWORKLOAD コマンドがデータベース接続前に発行される場合、データベース接続が確立された後の、次の作業単位の開始時に SQL0552N エラーが戻されます。データベース接続が確立されているときにコマンドが発行された場合は、ワークロードの再割り当てが実行される予定の、次の作業単位の開始時に SQL0552N エラーが戻されます。

## SET WRITE

**SET WRITE** コマンドを使用すると、ユーザーはデータベースへの入出力書き込み操作を中断したり、入出力書き込み操作を再開したりできます。

通常このコマンドは、ミラーリングされたデータベースを分割するという目的で使用します。このタイプのミラーリングは、ディスク・ストレージ・システムを使って行われます。

この新規の状態 `SUSPEND_WRITE` は、スナップショット・モニターから認識されます。この状態により、既存の書き込み操作は完了し、新規の書き込み操作は実行できないようになります。コマンドを正常に実行するためにすべての表スペースが `NORMAL` 状態になっている必要はありません。

### 有効範囲

このコマンドは、それが発行されたデータベース・パーティションに対してだけ影響を与えます。パーティション・データベース環境では、すべてのデータベース・パーティションにこれを発行する必要があります。DB2 pureScale 環境では、任意のメンバーからこれを発行して、すべてのメンバーについて入出力書き込み操作を中断したり、すべての中断状態のメンバーについて入出力書き込み操作を再開したりできます。

### 許可

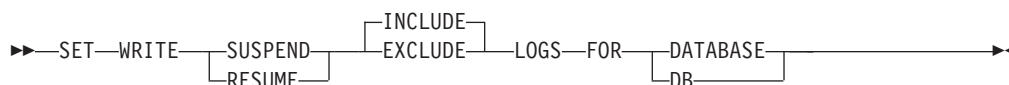
このコマンドの許可には、発行者が次の特権のうち 1 つを持っていることが必要です。

- SYSADM
- SYSCTRL
- SYSMMAINT

### 必要な接続

データベース

### コマンド構文



### コマンド・パラメーター

#### SUSPEND

ログへの書き込みを含む入出力書き込み操作を中断します。すべてのデータベース操作は、入出力書き込み操作の中断中に、オンライン・バックアップおよびリストアから離れて正常に機能します。しかし、操作によっては、バッファ・プールまたはログ・バッファからログへのダーティー・ページのフラッシュ試行中に待機するものもあります。それらの操作は、データベースに対する入出力書き込み操作を再開した後に続行されます。

**RESUME**

入出力書き込み操作を再開します。DB2 pureScale 環境では、このパラメーターはすべての中断状態のメンバーについて、入出力書き込み操作を再開します。

**INCLUDE LOGS**

データベースが書き込み中断状態にある場合はログ・ファイルへの書き込みを禁止することを指定します。これはデフォルトです。

**EXCLUDE LOGS**

データベースが書き込み中断状態にある場合に、ログ・ファイルに書き込みできる (ただし、ログ・ファイル・ヘッダー・ファイルやミラー・ログ・ファイル・ヘッダー・ファイルには書き込みできない) ようにすることを指定します。これにより、データベースに対して実行中の更新トランザクションが完了できるようにするための期間が設けられます。これは、データベースが書き込み中断状態になっているときに通常発生するはずの、ワークロードへの影響を削減するのに役立ちます。データベースが書き込み中断状態になっており、EXCLUDE LOGS オプションが指定されているときに取られたデータベースのコピーには、ログ・ファイルが含まれてはなりません。

注: 状況によっては、ログに記録された操作の処理がブロックされたままになることがあります。これは例えば、現在のアクティブなログ・ファイルが満杯の場合などに起こります。

**使用上の注意**

**suspend\_io** データベース構成パラメーターの設定値を表示することにより、入出力書き込み操作が中断されているかどうかを判別できます。データベース構成情報を表示するには、**GET DATABASE CONFIGURATION** コマンド、DBCFCG 管理ビュー、**-dbcfg** パラメーターを付けた **db2pd** コマンド、または **db2CfgGet** API を使用できます。

**SET WRITE** コマンドを使用する前に **FLUSH BUFFERPOOLS** ステートメントを使用して、スプリット・ミラー・データベースのリカバリー時間を最小化することができます。スプリット・ミラー・データベースをバックアップ・イメージとして使用することを計画している場合や、スプリット・ミラー・データベースを使用してバックアップ・イメージを作成することを計画している場合には、このステートメントを発行することが役立ちます。

ダーティー・ページをバッファ・プールからディスクにフラッシュする必要があるのに、**SET WRITE** コマンドを使用して入出力書き込み操作を再開することができない場合には、接続の試行が失敗します。接続の失敗を解決するには、**RESTART DATABASE** コマンドに **WRITE RESUME** パラメーターを付けて発行します。このシナリオでは、**RESTART DATABASE** コマンドは、クラッシュ・リカバリーを実行せずに書き込み操作を再開します。**RESTART DATABASE** コマンドに **WRITE RESUME** パラメーターを指定して使用すると、データベースのクラッシュ後にコマンドを使用する場合にのみ、クラッシュ・リカバリーを実行します。

このコマンドが正常に実行されるには、表スペースが **SQLB\_MOVE\_IN\_PROGRESS** または **SQLB\_BACKUP\_IN\_PROGRESS** などの過渡状態になっていても構いません。データベースが中断状態になっている場合に、スナップショット・モニターが

## SET WRITE

報告する可能性のある状態として REBAL\_IN\_PROGRESS もあることに注意してください。

## START DATABASE MANAGER

ターゲット・メンバーまたはすべてのメンバーで、データベース・マネージャーを開始します。DB2 pureScale 環境では、このコマンドはクラスター・キャッシング・ファシリティー (CF) も開始します。

このコマンドはクライアントでは無効です。

### 有効範囲

複数パーティション・データベース環境では、このコマンドは、`$HOME/sql1lib/db2nodes.cfg` ファイルにリストされているすべてのデータベース・パーティションに影響を与えます。ただし、**DBPARTITIONNUM** パラメーターが使用されていない場合に限りです。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMOINT

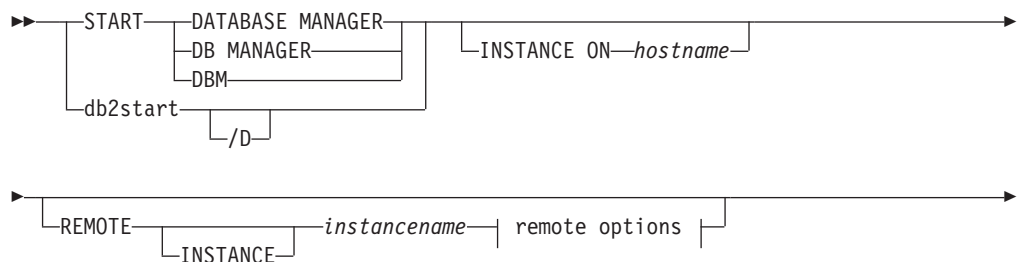
**ADD DBPARTITIONNUM** 開始オプションは、SYSADM または SYSCTRL 権限のどちらかを必要とします。

ユーザーはサービス開始に対する Windows オペレーティング・システム要件を満たす必要があります。拡張セキュリティーが無効になっている場合、ユーザーは Administrators、Server Operators、または Power Users グループのメンバーになっている必要があります。拡張セキュリティーが有効になっている場合、ユーザーはデータベースを始動するために Administrators グループまたは DB2ADMNS グループのメンバーになっている必要があります。

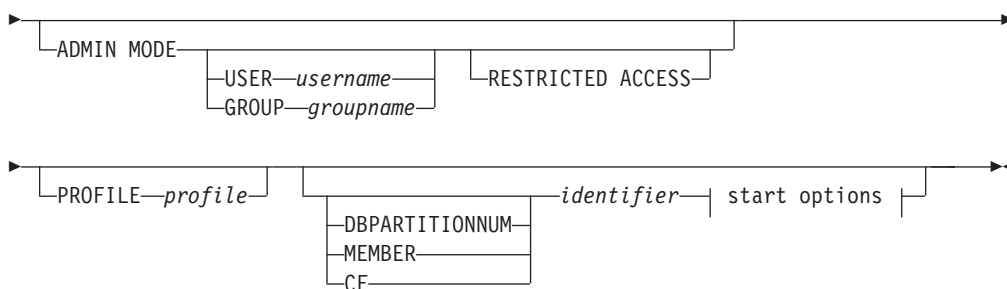
### 必要な接続

なし

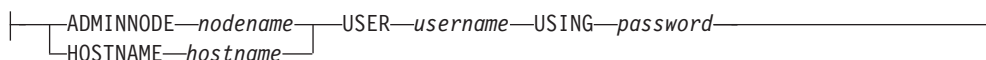
### コマンド構文



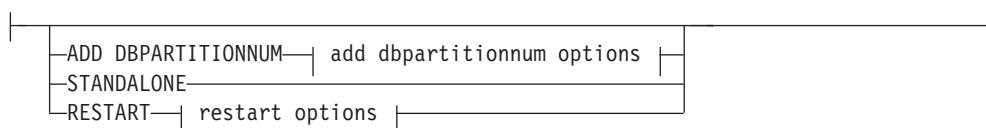
## START DATABASE MANAGER



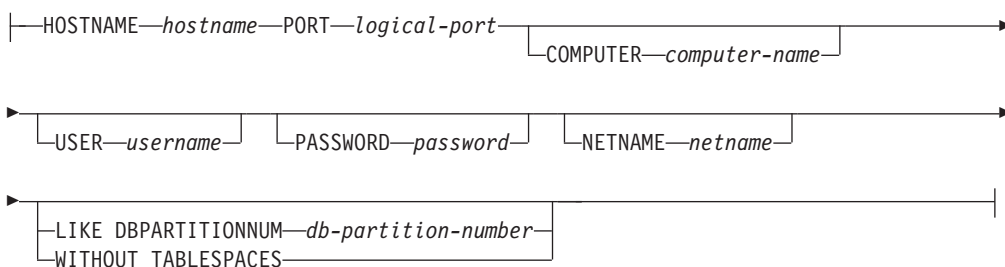
### remote options:



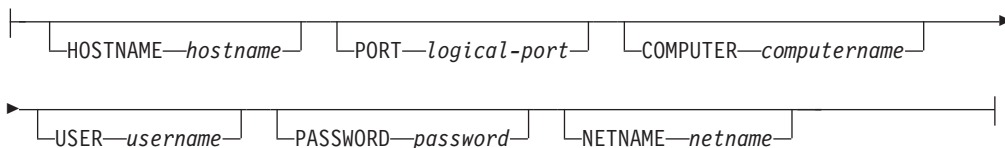
### start options:



### add dbpartitionnum options:



### restart options:



## コマンド・パラメーター

**/D** Windows 上にインストールされている DB2 製品がプロセスとして実行されることを許可します。パーティション・データベース環境では、**/D** パラメーターを使用しても、DB2 インスタンスをプロセスとして開始することはできないことに注意してください。

### INSTANCE ON *hostname*

メンテナンス操作が完了した後に、DB2 pureScale インスタンスの特定のホ

スト上で、DB2 インスタンスを開始するように指定します。DB2 pureScale 環境外で使用された場合には、SQL1695N エラーが返されます。

**REMOTE [INSTANCE]** *instancename*

開始したいリモート・インスタンスの名前を指定します。

**ADMINNODE** *nodename*

**REMOTE** または **REMOTE INSTANCE** では、管理ノードの名前を指定します。

**HOSTNAME** *hostname*

**REMOTE** または **REMOTE INSTANCE** では、ホスト・ノードの名前を指定します。

**USER** *username*

**REMOTE** または **REMOTE INSTANCE** では、ユーザーの名前を指定します。

**USING** *password*

**REMOTE** または **REMOTE INSTANCE**、および **USER** では、ユーザーのパスワードを指定します。

**ADMIN MODE**

管理の目的のために、インスタンスを静止モードで開始します。このパラメーターは、インスタンスが既に「活動中」でない場合以外は、**QUIESCE INSTANCE** コマンドと同じで、接続を強制クローズする必要はありません。

**ADMIN MODE** オプションのみを指定した場合、静止インスタンス内のデータベースは、データベースへのすべての接続試行に対する許可検査を行うためにアクティブ化されます。こうする必要があるのは、接続しようとするユーザー ID が DBADM 権限を持っているかどうかを判別するためです。この権限はデータベース・カタログに保管されているので、ユーザー ID がこの権限を持っているかどうかを判別するためには、そのデータベースをアクティブ化することが必要になります。この許可検査が行われないようにするには、**RESTRICTED ACCESS** オプションを指定します。

**USER** *username*

**ADMIN MODE** では、ユーザーの名前を指定します。

**GROUP** *groupname*

**ADMIN MODE** では、グループの名前を指定します。

**RESTRICTED ACCESS**

このオプションは、静止インスタンスのデータベースへのすべての接続試行に対する許可検査 (ユーザー ID が DBADM 権限を持っているかどうかを判別するための許可検査) が行われないようにする場合に指定します。この場合でも、インスタンス・レベルの許可検査は行われます。ユーザー ID の SYSADM、SYSCTRL、または SYSMAINT 権限の検査は、データベースのアクティブ化を必要としないからです。

以下のすべてのパラメーターは、Enterprise Server Edition (ESE) 環境でのみ有効です。



### PROFILE *profile*

DB2 環境を定義するために、各データベース・パーティションで実行しなければならないプロファイル・ファイル名を指定します。このファイルは、データベース・パーティションの開始前に実行されます。プロファイル・ファイルはインスタンス所有者の `sqllib` ディレクトリに常駐していなければなりません。プロファイル・ファイルの環境変数をすべてユーザー・セッションで定義する必要はありません。

### *identifier*

数値 ID を指定します。DBPARTITIONNUM、MEMBER、または CF パラメーターを指定する必要はありません。

### DBPARTITIONNUM *identifier*

このパラメーター・オプションは、パーティション・データベース環境でのみ有効です。開始するメンバーを指定します。ID が指定されない場合、ローカル・データベース・パーティションで通常の開始が行われます。有効な値は、0 から 999 (0 および 999 を含む) です。

### MEMBER *identifier*

開始するメンバーを指定します。DB2 pureScale 環境では、有効な値は 0 から 127 まで (両端を含む) です。

### CF *identifier*

このパラメーター・オプションは、DB2 pureScale 環境でのみ有効です。開始するクラスター・キャッシング・ファシリティ (CF) を指定します。有効な値は 128 および 129 です。DB2 pureScale 環境外で使用された場合には、SQL1695N エラーが返されます。

### ADD DBPARTITIONNUM

インスタンス所有者の `db2nodes.cfg` ファイルに、HOSTNAME および PORT パラメーター値とともに、新しいデータベース・パーティション・サーバーを追加することを指定します。このオプションは、パーティション・データベース環境でのみ有効です。

**HOSTNAME** および **PORT** パラメーター値の組み合わせが固有のものであることを確認してください。

データベース・パーティション・サーバー追加ユーティリティが内部で実行され、追加されたデータベース・パーティション・サーバーに既存のデータベースすべてが作成されます。新しいデータベース・パーティション・サーバーは `db2nodes.cfg` ファイルに自動的に追加されます。

**注:** 新規データベース・パーティションを追加する際に、アンカタログされたデータベースは認識されません。アンカタログされたデータベースは、新規データベース・パーティションには存在しません。新規データベース・パーティション上のデータベースに接続しようとする、エラー・メッセージ SQL1013N が戻されます。

アクティブなデータベース・パーティション・サーバーが複数ある環境で ADD 要求が行われると、ADD 処理の完了時に新しいデータベース・パーティション・サーバーがこの環境で可視になります。

データベース・パーティション・サーバーが 1 つで、そのサーバーがアクティブな環境で ADD 要求を行うと、ADD 処理の完了後、新しいデータバ

ース・パーティション・サーバーは非アクティブになります。新しいデータベース・パーティション・サーバーがパーティション・データベース環境に加わるには、その前に **db2stop** および **db2start** を用いてインスタンスを再始動する必要があります。データベース・パーティション・サーバーが 1 つあり、そのサーバーが非アクティブな環境で ADD 要求を行うと、ADD 処理の完了後、新しいデータベース・パーティション・サーバー (または複数を追加する場合には複数のサーバー) がアクティブになります。元のデータベース・パーティション・サーバーのみを開始する必要があります。

新しく追加されたデータベース・パーティションは、ADD 処理の際に以下のように構成されます。

1. 複数パーティション環境では、新しいデータベース・パーティションは、非カタログ・データベース・パーティションからのデータベース構成パラメーター値を用いて構成されます。
2. 単一パーティション環境では、新しいデータベース・パーティションは、カタログ・パーティションからのデータベース構成パラメーター値を用いて構成されます。
3. 新しいデータベース・パーティションへのデータベース構成パラメーター値のコピーの際に問題が生じると、新しいデータベース・パーティションはデフォルトのデータベース構成パラメーター値を用いて構成されます。

**HOSTNAME** *hostname*

**ADD DBPARTITIONNUM** パラメーターを使用して、db2nodes.cfg ファイルに追加するホスト名を指定します。

**PORT** *logical-port*

**ADD DBPARTITIONNUM** を使用して、db2nodes.cfg ファイルに追加する論理ポートを指定します。有効な値は 0 から 999 です。

**COMPUTER** *computername*

新しいデータベース・パーティションが作成されるマシンのコンピューター名。このパラメーターは、Windows では必須ですが、その他のオペレーティング・システムでは無視されます。

**USER** *username*

新しいデータベース・パーティション上のアカウントのユーザー名。このパラメーターは、Windows では必須ですが、その他のオペレーティング・システムでは無視されます。

**PASSWORD** *password*

新しいデータベース・パーティション上のアカウントのパスワード。このパラメーターは、Windows では必須ですが、その他のオペレーティング・システムでは無視されます。

**NETNAME** *netname*

db2nodes.cfg ファイルに追加する netname を指定します。指定されていない場合、**HOSTNAME** パラメーターで指定された値がデフォルトとなります。

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

SYSTEM TEMPORARY 表スペース用のコンテナが、インスタンス内の各データベース用に指定した db-partition-number 値のコンテ

ナーと同一になるように指定します。指定するデータベース・パーティションは、`db2nodes.cfg` ファイル中に既に指定してあるデータベース・パーティションでなければなりません。自動ストレージを使用する **SYSTEM TEMPORARY** 表スペースでは、これらのコンテナは、指定されたパーティションのコンテナと必ずしも一致しません。その代わりに、コンテナは、データベースに関連付けられたストレージ・パスに基づいてデータベース・マネージャーによって自動的に割り当てられます。この自動割り当ての結果として、それら 2 つのパーティションで同じコンテナが使用されることになる可能性があります。

### WITHOUT TABLESPACES

**SYSTEM TEMPORARY** 表スペースのコンテナがどのデータベースに対しても作成されないことを指定します。データベースを使用する前に、**ALTER TABLESPACE** ステートメントを使用して、**SYSTEM TEMPORARY** 表スペース・コンテナを各データベースに追加しなければなりません。自動ストレージを使用するよう定義された **SYSTEM TEMPORARY** 表スペース (これは **CREATE TABLESPACE** ステートメントの **MANAGED BY AUTOMATIC STORAGE** 節を使って作成されたか、あるいは **MANAGED BY CLAUSE** がまったく指定されなかった **SYSTEM TEMPORARY** 表スペースを示します) の場合、このオプションは無視されます。このような表スペースに関しては、コンテナ作成を先に延ばすことはできません。コンテナは、データベース・マネージャーにより、データベースに関連するストレージ・パスを基に自動的に割り当てられます。

### STANDALONE

このオプションは、パーティション・データベース環境でのみ有効です。データベース・パーティションが独立方式で開始されることを指定します。**FCM** は他のデータベース・パーティションとの接続を確立しようとはしません。このオプションはデータベース・パーティションの追加の時に使用します。

### RESTART

このオプションは、パーティション・データベース環境でのみ有効です。障害発生後、データベース・マネージャーを起動します。他のデータベース・パーティションの操作は続いており、このデータベース・パーティションは他のデータベース・パーティションとの接続を試みます。**HOSTNAME** および **LOGICAL-PORT** パラメーターがどちらも指定されない場合、データベース・マネージャーは、`db2nodes.cfg` でパーティションに対して指定されているホスト名とポートの値を使用して再始動します。どちらかの値が指定されている場合は、接続の確立時に新しい値が他のデータベース・パーティションに送信されます。`db2nodes.cfg` ファイルは、この情報に基づいて更新されます。**RESTART** オプションを使用して `db2nodes.cfg` ファイルを更新する場合、ポート 0 のデータベース・パーティション項目は、より大きいポート番号を持つ他のデータベース・パーティションが削除されるまで削除しないでください。

**HOSTNAME** *hostname*

**HOSTNAME** パラメーターを **RESTART** パラメーターとともに使用する

と、データベース・パーティションを、データベース・パーティション構成ファイル `db2nodes.cfg` で指定されたものとは別のマシンで再開できます。

#### 制約事項:

DB2 High Availability Feature を使用するときには、**HOSTNAME** オプションを **RESTART** パラメーターとともに使用してデータベース・パーティションを別のマシンで再開しないでください。DB2 pureScale インスタンス内の特定のマシンのデータベース・パーティションを別のマシンで再開する、または別のマシンに移動する場合には、DB2 高可用性インスタンス構成ユーティリティー (**db2haicu**) を使用します。

#### PORT *logical-port*

**RESTART** を使用して、データベース・パーティション構成ファイルにある論理ポート番号を上書きするために使用する論理ポート番号を指定します。指定されていない場合、このパラメーターは、`db2nodes.cfg` ファイルに指定されているデータベース・パーティションの論理ポート番号がデフォルトになります。有効な値は 0 から 999 です。

#### COMPUTER *computername*

新しいデータベース・パーティションが作成されるマシンのコンピューター名。このパラメーターは、Windows では必須ですが、その他のオペレーティング・システムでは無視されます。

#### USER *username*

新しいデータベース・パーティション上のアカウントのユーザー名。このパラメーターは、Windows では必須ですが、その他のオペレーティング・システムでは無視されます。

#### PASSWORD *password*

新しいデータベース・パーティション上のアカウントのパスワード。このパラメーターは、Windows では必須ですが、その他のオペレーティング・システムでは無視されます。

#### NETNAME *netname*

`db2nodes.cfg` ファイルで指定されたネット名を上書きするために使用される *netname* 値を指定します。指定されていない場合、`db2nodes.cfg` ファイルの論理データベース・パーティション番号に対応する、*netname* の値がデフォルトとなります。

## 例

以下に示すのは、10、20、および 30 のメンバーを持つ DB2 インスタンスに対して発行された **db2start** からの出力例です。

```
04-07-1997 10:33:05 10 0 SQL1063N DB2START processing was successful.
04-07-1997 10:33:07 20 0 SQL1063N DB2START processing was successful.
04-07-1997 10:33:07 30 0 SQL1063N DB2START processing was successful.
SQL1063N DB2START processing was successful.
```

### 使用上の注意

Microsoft Windows Vista 以降のバージョンで、ローカルまたはドメインの管理者権限を持つ ID を使用している場合には、完全な管理者特権で実行する DB2 コマンド・ウィンドウからこのコマンドを実行する必要があります。

このコマンドをクライアント・ノードで発行しない場合もあります。旧クライアントとの互換性が提供されていますが、データベース・マネージャーには何も影響ありません。

一度開始されると、データベース・マネージャーのインスタンスは、たとえそれを使用していたすべてのアプリケーション・プログラムが終了したとしても、ユーザーが停止しない限り稼働しています。

データベース・マネージャーが正常に開始されると、正常終了メッセージが標準出力装置に送られます。エラーが発生すると、処理は停止され、エラー・メッセージが標準出力装置に送られます。パーティション・データベース環境では、メッセージは **START DATABASE MANAGER** コマンドを発行したデータベース・パーティションに戻されます。

パーティション・データベース環境でパラメーターが指定されていない場合、データベース・パーティション構成ファイルで指定されたパラメーターを使用してすべてデータベース・パーティションでデータベース・マネージャーが開始されます。

**START DATABASE MANAGER** コマンド実行中の場合、データベースへの要求を発行する前に、適用可能なデータベース・パーティションが開始されていることを確認してください。

db2cshrc ファイルはサポートされておらず、環境の定義付けに使用できません。

インスタンスを静止状態で開始できます。これは、以下のいずれかを使用して行うことができます。

```
db2start admin mode
```

または

```
db2start admin mode user username
```

または

```
db2start admin mode group groupname
```

**RESTRICTED ACCESS** オプションを指定すると、許可検査を行うために静止インスタンス内のデータベースがアクティブ化されることがなくなります。データベースに接続しようとしているユーザー ID が、そのデータベースに対する DBADM 権限または QUIESCE\_CONNECT 特権を持っていても、接続は許可されません。

SYSADM、SYSCTRL、または SYSMANT 権限を持つユーザー ID、およびこのコマンドで指定されたユーザーまたはグループのみが、データベースへの接続を許可されます。

**RESTRICTED ACCESS** オプションは、静止インスタンス内のデータベースへの排他的接続が必要な場合に使用するようになっています。このようなケースとしては、オフライン・バックアップを行う場合や、他の保守アクティビティーを実行する場合が挙げられます。

新しいデータベース・パーティション・サーバーの追加時には、インスタンス内の各データベースに対して自動ストレージが有効になっているかどうかを **START DATABASE MANAGER** が判別する必要があります。これは、各データベースごとに、カタログ・パーティションとの通信によって行われます。自動ストレージが有効になっていれば、その通信によってストレージ・パス定義が得られます。同様に、データベース・パーティションに **SYSTEM TEMPORARY** 表スペースを作成する場合は、別のデータベース・パーティション・サーバーに常駐するデータベース・パーティションの表スペース定義を入手するために、**START DATABASE MANAGER** はそのサーバーと通信しなければならないかもしれません。 **start\_stop\_time** データベース・マネージャー構成パラメーターを使用して、時間 (分) を指定します。他のデータベース・パーティション・サーバーはこの時間内で自動ストレージおよび表スペース定義に応答する必要があります。この時間を超えると、このコマンドは失敗します。そのような場合は、**start\_stop\_time** の値を増やして、コマンドを再発行してください。

以下のいずれかのコマンド、ステートメント、または操作が進行中の場合、新しいデータベース・パーティション・サーバーを追加することはできません。それ以外の場合は、SQL6074N が戻されます。

- **QUIESCE INSTANCE**
- **UNQUIESCE INSTANCE**
- **STOP DB2 (db2stop)**
- **STOP DATABASE MANAGER DBPARTITIONNUM**
- **START DB2 (db2start)**
- **START DATABASE MANAGER DBPARTITIONNUM**
- **START DATABASE MANAGER** (再始動オプションを使用)
- **CREATE DATABASE**
- **DROP DATABASE**
- **QUIESCE DATABASE**
- **UNQUIESCE DATABASE**
- **ACTIVATE DATABASE**
- **DEACTIVATE DATABASE**
- データベース・オブジェクトに対する Z ロック
- すべてのデータベース・パーティション・サーバー上のデータベースのバックアップ
- データベースのリストア
- 表スペースの ALTER、または DROP
- 自動ストレージ・パスの更新

UNIX オペレーティング・システムでは、**START DATABASE MANAGER** コマンドは SIGINT 信号をサポートしています。これは CTRL+C を押すと発行されます。この



## START DATABASE MANAGER

信号が発生すると、進行中のすべての始動操作に割り込みが生じ、メッセージ (SQL1044N) が \$HOME/sql1lib/log/db2start.timestamp.log エラー・ログ・ファイルに戻されます。既に開始済みのメンバーには影響がありません。開始中のメンバーに対して CTRL+C が押された場合には、そのメンバーを再び開始しようとする前に、**db2stop** をそのメンバーに対して発行しなければなりません。

Windows オペレーティング・システムでは、開始に失敗した通信サブシステムがあっても、**db2start** コマンドおよび **NET START** コマンドは警告を戻しません。Windows 環境のデータベース・マネージャーは、サービスとしてインプリメントされます。サービスが正常に開始された場合にはエラーを戻しません。イベント・ログまたは db2diag ログ・ファイルを調べて、**db2start** の実行中にエラーが発生しなかったか確認してください。

### 互換性

以前のバージョンとの互換性:

- **DB2\_ENFORCE\_MEMBER\_SYNTAX** レジストリー変数が ON に設定されている場合を除き、**LIKE DBPARTITIONNUM** または **LIKE NODE** を **LIKE MEMBER** の代わりに使用できます。
- **DB2\_ENFORCE\_MEMBER\_SYNTAX** レジストリー変数が ON に設定されている場合を除き、**ADD DBPARTITIONNUM** または **ADD NODE** を **ADD MEMBER** の代わりに使用できません。
- **DB2\_ENFORCE\_MEMBER\_SYNTAX** レジストリー変数が ON に設定されている場合を除き、**DBPARTITIONNUM** または **NODENUM** を **MEMBER** の代わりに使用できます。**DBPARTITIONNUM** が DB2 pureScale 環境で指定され、**DB2\_ENFORCE\_MEMBER\_SYNTAX** レジストリー変数が ON に設定されている場合、SQL1694N エラーが戻されません。



## START HADR

データベースの HADR 操作を開始します。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMANT

### 必要な接続

インスタンス。このコマンドは、データベースが存在しない場合にデータベース接続を確立し、コマンドが完了するとデータベース接続をクローズします。

### コマンド構文

```

▶▶ START HADR ON DATABASE database-alias
                  |
                  | DB
▶
▶ USER user-name AS PRIMARY
                  |
                  | USING password
▶
▶ STANDBY BY FORCE

```

### コマンド・パラメーター

**DATABASE** *database-alias*

HADR 操作を開始するデータベース。

**USER** *user-name*

HADR 操作を開始するために使用するユーザー名。

**USING** *password*

*user-name* を認証するのに使用されるパスワード。

**AS PRIMARY**

データベースに対して HADR 1 次操作を開始することを指定します。データベース。1 次データベースが **hadr\_timeout** データベース構成パラメーターで指定された時間内に HADR スタンバイ・データベースに接続できないと、1 次データベースは開始されません。

**BY FORCE**

HADR 1 次データベースにおいて、スタンバイ・データベースがそれに接続するまで待機しないことを指定します。 **BY FORCE** オプションによる開始の後も 1 次データベースは、後でスタンバイ・データベースが使用できるようになった時点で、スタンバイ・データベースからの有効な接続を受け付けます。 **BY FORCE** が使用されるとき、データベース構成パラメーター **autorestart** の値には関係なく、データベースは必要に応じてクラッシュ・リカバリーを実行します。 1 次データベースを開始するその他の方法 (強制されない **START HADR** コマンド、 **ACTIVATE DATABASE** コマンド、クライアント接続など) は、 **autorestart** 設定値を優先します。

**注意:** START HADR コマンドに AS PRIMARY BY FORCE オプションを指定する場合には、十分な注意が必要です。スタンバイ・データベースが 1 次データベースに変更されてから、AS PRIMARY BY FORCE オプションを指定した START HADR コマンドの発行により元のデータベースが再始動した場合、データベースの 2 つのコピーが 1 次として独立して動作することになります。(これはスプリット・ブレイン、または二重 1 次 と呼ばれることがあります)。この場合、各 1 次データベースは複数の接続を受け入れたり複数のトランザクションを実行したりできますが、もう一方のデータベースによる更新は受け取ることも再生することもできません。そのため、データベースのそれら 2 つのコピーは、互いに矛盾することになります。

**AS STANDBY**

データベースに対して HADR スタンバイ操作を開始することを指定します。スタンバイ・データベースは、接続が正常に確立されるまで、または接続試行が 1 次データベースによって明示的に拒否されるまで、HADR 1 次データベースへの接続を試行します。(1 次データベースが接続を拒否する場合として考えられるのは、HADR 構成パラメーターが正しく設定されていない場合、またはデータベースのコピーが矛盾している場合であり、いずれにしても接続再試行を続けることは適当ではありません。)

**使用上の注意**

さまざまな条件におけるデータベースの動作を、次の表に示します。

データベースの状況	START HADR コマンド (AS PRIMARY オプション) での動作	START HADR コマンド (AS STANDBY オプション) での動作
非アクティブ標準データベース	HADR 1 次データベースとしてアクティブ化されます。	データベースがロールフォワード・ペンディング・モード (リストアまたはスプリット・ミラーなどの結果)、またはロールフォワード進行中モードの場合、データベースはスタンバイ・データベースとして開始します。それ以外の場合、エラーが戻されます。
アクティブ標準データベース	データベースは HADR 1 次の役割になります。	エラー・メッセージが戻されます。
非アクティブ 1 次データベース	HADR 1 次データベースとしてアクティブ化されます。	フェイルオーバーの後、障害の発生した 1 次を新しいスタンバイ・データベースとして HADR ペアに再び組み入れます。いくらかの制限があります。
アクティブ 1 次データベース	警告メッセージが発行されます。	エラー・メッセージが戻されます。
非アクティブ・スタンバイ・データベース	エラー・メッセージが戻されます。	データベースをスタンバイ・データベースとして開始します。
アクティブ・スタンバイ・データベース	エラー・メッセージが戻されます。	警告メッセージが発行されます。

**START HADR** コマンドの発行時に、対応するエラー・コード SQL1767N、SQL1769N、または SQL1770N と理由コード 98 が生成される場合があります。この理由コードは、コマンドが発行されたサーバー上に HADR のライセンスがインストールされていないことを示します。この問題を訂正するには、**db2licm** を使用して有効な HADR ライセンスをインストールするか、または配布の一部として有効な HADR ライセンスが含まれているサーバーのバージョンをインストールします。

## STOP DATABASE MANAGER

ターゲット・メンバーまたはすべてのメンバーで、データベース・マネージャーを停止します。DB2 pureScale 環境でこれを使用すると、クラスター・キャッシング・ファシリティー (CF) も停止します。

### 有効範囲

DB2 pureScale 環境またはパーティション・データベース環境では、このコマンドは、db2nodes.cfg ファイルにリストされているすべてのメンバーに作用します。ただし、**DBPARTITIONNUM**、**MEMBER**、および **CF** パラメーターがいずれも指定されていない場合に限りです。

アクティブ・データベース接続が存在する場合には、このコマンドはメンバーのシャットダウンを行いません。アクティブ・データベース接続がないものの、インスタンス接続がある場合には、まずインスタンス接続が強制的にオフにされてから、メンバーが停止されます。また、メンバーを停止させる前に、処理中のデータベースの活動を非アクティブ化します。このコマンドを使用して、特定のホスト上のインスタンス・サービスを停止することもできます。

このコマンドはクライアントでは無効です。

### 許可

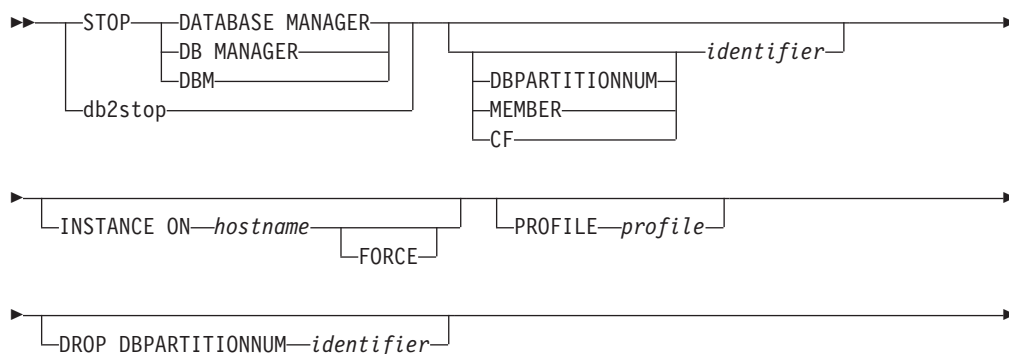
以下の権限のいずれか。

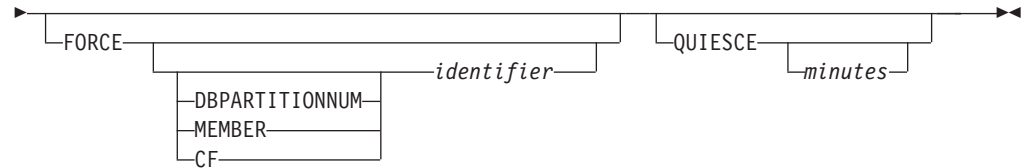
- SYSADM
- SYSCTRL
- SYSMAINT

### 必要な接続

なし

### コマンド構文





## コマンド・パラメーター

### INSTANCE ON *hostname*

メンテナンス操作のために、DB2 pureScale インスタンスの特定のホスト上で、DB2 インスタンスを停止するように指定します。このコマンドは、グローバル **db2start** コマンドによってホストが再始動されることを一時的に回避します。ホストは、*restart light* 状態のゲスト・ホストとしては機能できなくなります。メンテナンス操作の完了後にホスト上の DB2 インスタンスを再始動するには、**START DBM INSTANCE ON *hostname*** コマンドを発行してください。アクティブなメンバーまたはアクティブなクラスター・キャッシング・ファシリティ (CF) がホスト上で稼働している場合、このコマンドは失敗します。DB2 pureScale 環境外で使用された場合には、SQL1695N エラーが返されます。

**FORCE** 指定された場合、**FORCE** パラメーターはホスト上の DB2 インスタンスを即座に停止します。他のアクティブ・ホストにフェイルオーバーするホストで稼働しているアクティブ・メンバー、またはアクティブなクラスター・キャッシング・ファシリティ (CF) があれば、すべて強制停止されます。他のホストにフェイルオーバーされるメンバーは「*restart light* メンバー」と呼ばれます。それは、ホスト上の最小限のリソースを使用し、データベース接続を受け入れないことを意味しています。詳しくは、『*restart light*』を参照してください。

### *identifier*

数値 ID を指定します。**DBPARTITIONNUM**、**MEMBER**、または **CF** パラメーターを指定する必要はありません。

### **DBPARTITIONNUM** *identifier*

このパラメーター・オプションは、パーティション・データベース環境でのみ有効です。停止するメンバーを指定します。有効な値は、0 から 999 (0 および 999 を含む) です。

**注:** このパラメーターを指定せずに **FORCE** オプションを使用すると、すべてのデータベース・パーティションが停止される前に、すべてのデータベース・パーティション上のすべてのアプリケーションが強制的に停止されます。

**注:** *identifier* 値は、インスタンス所有者の *db2nodes.cfg* ファイルに存在していなければなりません。データベース・パーティション番号が指定されていない場合、構成ファイルで定義されているすべてのデータベース・パーティションが停止されます。

## STOP DATABASE MANAGER

### MEMBER *identifier*

停止するメンバーを指定します。DB2 pureScale 環境では、有効な値は 0 から 127 まで (両端を含む) です。

### CF *identifier*

このパラメーター・オプションは、DB2 pureScale 環境でのみ有効です。停止するクラスター・キャッシング・ファシリティー (CF)を指定します。有効な値は 128 および 129 です。DB2 pureScale 外で使用された場合には、SQL1695N エラーが返されます。

### PROFILE *profile*

このパラメーター・オプションは、パーティション・データベース環境でのみ有効です。開始されたデータベース・パーティション用に DB2 環境を定義するために、始動時に実行されたプロファイル・ファイル名を指定します。 **START DATABASE MANAGER** コマンドでプロファイルが指定されると、ここでも同じプロファイルを指定しなければなりません。プロファイル・ファイルはインスタンス所有者の `sql1lib` ディレクトリーに常駐していなければなりません。

### DROP DBPARTITIONNUM *identifier*

このパラメーター・オプションは、パーティション・データベース環境でのみ有効です。 `db2nodes.cfg` 構成ファイルからドロップするデータベース・パーティションを指定します。

**注:** **DROP DBPARTITIONNUM** パラメーターを使用する前に、**DROP DBPARTITIONNUM VERIFY** コマンドを実行して、このデータベース・パーティション環境にユーザー・データが存在しないことを確認してください。

**FORCE** 各データベース・パーティションでデータベース・マネージャーを停止する際に **FORCE APPLICATION ALL** を使用することを指定します。

### QUIESCE *minutes*

このパラメーター・オプションは、DB2 pureScale 環境でのみ有効で、**MEMBER** パラメーター節にのみ有効です。これは、メンバーがシャットダウンされる前に、特定のメンバー上でアクティブ・ワークロードを停止するために使用されます (クラスター・キャッシング・ファシリティーに対して使用することはできません)。アクティブ・アプリケーションが即座に中断される **db2stop FORCE** コマンドとは異なり、**QUIESCE** は、メンバー上のすべてのアプリケーションがそれぞれのアクティブ・トランザクションを完了するまで待機することにより、アクティブ・アプリケーションが正常終了するための余裕を与えます。オプションの *minutes* パラメーター値は、特定のメンバーから切断することをアプリケーションに通知する前にコマンドが待機する時間の長さを指定します。このタイムアウトに達すると、その時点で残っているアクティブな作業単位があれば、すべて中断されます。タイムアウトが指定されていない場合は、コマンドは、メンバーのシャットダウンを進める前に、すべてのアクティブ・アプリケーションがそれぞれの作業単位を終了するまで、無期限に待機します。

**重要:** メンバーをオンラインに戻すには、**db2start** コマンドをメンバーに対して発行する必要があります。

*minutes* パラメーター値の有効な値は、-1 から 1440 まで (両端を含む) です。デフォルト値は -1 で、すべてのアクティブ・ワークロードが終了するまで、指定されたメンバーが無期限に待機することを意味します。0 の値が指定された場合には、アクティブ・アプリケーションが即座に中断され、それらが終了すると、メンバーがシャットダウンされます。

**注: db2stop QUIESCE** コマンドは、割り込み可能ではありません。そのコマンドを発行した後は、さらに DB2 コマンドをそのメンバーに対して直接実行することはできません。静止されたメンバーに関する情報を照会したい場合には、別のアクティブ・メンバーから **LIST APPLICATIONS GLOBAL SHOW DETAIL** コマンドを発行する必要があります。静止されたメンバー上で実行されているアプリケーションを停止したい場合には、別のアクティブ・メンバーから **FORCE APPLICATION** '(*<app handle>*)' コマンドを発行する必要があります。

## 例

以下に示すのは、10、20、および 30 のメンバーを持つ DB2 インスタンスに対して発行された **db2stop** からの出力例です。

```
04-07-1997 10:32:53 10 0 SQL1064N DB2STOP processing was successful.
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.
```

## 使用上の注意

Microsoft Windows Vista 以降のバージョンでは、完全な管理者特権で実行する DB2 コマンド・ウィンドウからこのコマンドを実行する必要があります。

このコマンドをクライアント・ノードで発行しない場合もあります。旧クライアントとの互換性が提供されていますが、データベース・マネージャーには何も影響ありません。

一度開始されると、データベース・マネージャーのインスタンスは、たとえそれを使用していたすべてのアプリケーション・プログラムが終了したとしても、ユーザーが停止しない限り稼働しています。

データベース・マネージャーが停止されると、正常終了メッセージが標準出力装置に送られます。エラーが発生すると、処理は停止され、エラー・メッセージが標準出力装置に送られます。

アプリケーション・プログラムがまだデータベースに接続されているため、データベース・マネージャーが停止できない場合には、**FORCE APPLICATION** コマンドを使用して、まず最初にすべてのユーザーを切断するか、**FORCE** オプションで **STOP DATABASE MANAGER** コマンドを再発行してください。

次の情報は、パーティション・データベース環境にのみ適用されます。

- パラメーターが指定されない場合、データベース・マネージャーは構成ファイルにリストされている各データベース・パーティションで停止します。管理通知ログには、他のデータベース・パーティションが遮断されていることを示すメッセージが含まれています。



## STOP DATABASE MANAGER

- 前の **STOP DATABASE MANAGER** コマンドが発行されてからパーティション・データベース環境に追加されたデータベース・パーティションは、`db2nodes.cfg` ファイル内で更新されます。
- UNIX オペレーティング・システムでは、**start\_stop\_time** データベース・マネージャー構成パラメーターで指定された値に達すると、進行中のすべての停止操作が中断され、中断した各データベース・パーティションから `$HOME/sqllib/log/db2stop.timestamp.log` エラー・ログ・ファイルに、メッセージ `SQL6037N` が戻されます。既に停止しているデータベース・パーティションには影響がありません。
- `db2cshrc` ファイルはサポートされておらず、**PROFILE** パラメーターの値として指定することはできません。

**重要:** UNIX の `kill` コマンドを使用してデータベース・マネージャーを終了することはしないでください。このコマンドは、制御された終了もクリーンアップ処理も行わずに突然、データベース・マネージャー・プロセスを終了してしまうためです。

## STOP HADR

データベースの HADR 操作を停止します。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMANT

### 必要な接続

インスタンス。このコマンドは、データベースが存在しない場合にデータベース接続を確立し、コマンドが完了するとデータベース接続をクローズします。

### コマンド構文

```

▶▶ STOP HADR ON DATABASE database-alias
                   DB
▶
USER user-name
   USING password

```

### コマンド・パラメーター

**DATABASE** *database-alias*

HADR 操作を停止するデータベース。

**USER** *user-name*

HADR 操作を停止するために使用するユーザー名。

**USING** *password*

*user-name* を認証するのに使用されるパスワード。

### 使用上の注意

さまざまな条件におけるデータベースの動作を、次の表に示します。

データベースの状況	STOP HADR コマンドでの動作
非アクティブ標準データベース	エラー・メッセージが戻されます。
アクティブ標準データベース	エラー・メッセージが戻されます。
非アクティブ 1 次データベース	データベースの役割は標準に変更されます。データベース構成パラメーター <b>hadr_db_role</b> は STANDARD に更新されます。データベースはオフラインのままです。次回再始動時に標準の役割になります。

## STOP HADR

データベースの状況	STOP HADR コマンドでの動作
アクティブ 1 次データベース	HADR スタンバイ・データベースへのログ出力を停止し、HADR 1 次データベース上のすべての HADR EDU をシャットダウンします。データベースの役割は標準に変更され、データベースはオンラインのままです。 <b>AS PRIMARY</b> オプションを指定した明示的な <b>START HADR</b> コマンドが発行されるまで、データベースは標準の役割のままです。 <b>STOP HADR</b> コマンドは、オープンされているセッションとトランザクションには影響しません。データベースがオンラインのまま、 <b>STOP HADR</b> コマンドと <b>START HADR</b> コマンドを繰り返して発行できます。それらのコマンドは動的に影響を及ぼします。
非アクティブ・スタンバイ・データベース	データベースの役割は標準に変更されます。データベース構成パラメーター <b>hadr_db_role</b> は <b>STANDARD</b> に更新されます。データベースはオフラインのままです。データベースはロールフォワード・ペンディング・モードになります。
アクティブ・スタンバイ・データベース	エラー・メッセージが戻されます。スタンバイ・データベースを標準データベースに変更する前にそれを非アクティブにします。

**STOP HADR** コマンドの発行時に、対応するエラー・コード **SQL1767N**、**SQL1769N**、または **SQL1770N** と理由コード **98** が生成される場合があります。この理由コードは、コマンドが発行されたサーバー上に **HADR** のライセンスがインストールされていないことを示します。この問題を訂正するには、**db2licm** を使用して有効な **HADR** ライセンスをインストールするか、または配布の一部として有効な **HADR** ライセンスが含まれているサーバーのバージョンをインストールします。

## TAKEOVER HADR

HADR スタンバイ・データベースに対して、HADR ペアの新しい HADR 1 次データベースとしてテークオーバーするように指定します。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMANT

### 必要な接続

インスタンス。このコマンドは、データベースが存在しない場合にデータベース接続を確立し、コマンドが完了するとデータベース接続をクローズします。

### コマンド構文

```

▶▶ TAKEOVER HADR ON DATABASE database-alias
   |
   | DB
   |
▶
USER user-name
  |
  | USING password
  |
  | BY FORCE
  |
  | PEER WINDOW ONLY
  |
▶▶▶

```

### コマンド・パラメーター

#### DATABASE *database-alias*

HADR 1 次データベースとしてテークオーバーする現在の HADR スタンバイ・データベースを示します。

#### USER *user-name*

テークオーバー操作を開始するユーザー名を示します。

#### USING *password*

*user-name* を認証するのに使用されるパスワード。

#### BY FORCE

データベースが、オリジナルの HADR 1 次データベースがシャットダウンしたという確認を待たないことを指定します。このオプションは、HADR ペアがピア状態ではない場合に必要です。

#### PEER WINDOW ONLY

このオプションが指定された場合、コマンドが正常に実行された後、ピア・ウィンドウ期間が終了する前に 1 次データベースがダウンする場合 (データベース構成パラメーター `hadr_peer_window` をゼロ以外の値に設定) でも、コミット済みトランザクションの損失はありません。ピア・ウィンドウ期間が満了する前に 1 次データベースをダウンさせない場合、結果としてスプリット・ブレイン になります。HADR ペアがピア状態または切断済みピア状態 (ピア・ウィンドウが終了済み) でないときに **TAKEOVER BY FORCE PEER WINDOW ONLY** コマンドを実行した場合、エラーが戻されます。

同期モードが ASYNC または SUPERASYNC に設定されている場合には、**PEER WINDOW ONLY** オプションは使用できません。

注: 1 次データベース・クロックとスタンバイ・データベース・クロックが 5 秒以内の誤差で互いに同期されていない場合には、**PEER WINDOW ONLY** オプションを使ったテークオーバー操作が誤った動作をする可能性があります。つまり、操作が失敗するはずの場合に成功したり、成功するはずの場合に失敗したりします。時刻同期サービス (例えば NTP) を使用して、クロックを同じソースに同期させる必要があります。

### 使用上の注意

次の表は、考えられる状態とオプションの組み合わせと、その各組み合わせでアクティブ・スタンバイ・データベースに **TAKEOVER HADR** コマンドを発行した場合の動作を示しています。非アクティブのスタンバイ・データベースに対してこのコマンドを発行すると、エラー・メッセージが戻されます。

スタンバイ状態	BY FORCE オプションを使用	テークオーバー動作
切断済みピア	いいえ	NOT ALLOWED (SQL1770N)
切断済みピア	はい (BY FORCE のみ)	ALLOWED - NO ASSURANCE OF DATA CONSISTENCY 注: このほか、「トランザクション損失なし」テークオーバーは、必要な条件が保持されている限り、 <b>PEER WINDOW ONLY</b> オプションを指定せずに <b>TAKEOVER BY FORCE</b> コマンドを使用した場合 (つまり無条件フェイルオーバーの場合) にも可能です。1 次データベースの障害時に有効であったピア・ウィンドウの終了後、長い時間が経過した場合でも、そのようなフェイルオーバーを実行できます。
切断済みピア	はい (BY FORCE PEER WINDOW ONLY のみ)	ALLOWED - GREATER DEGREE OF DATA CONSISTENCY 次のような状況では、依然としてデータ損失が発生する可能性があります。 <ul style="list-style-type: none"> <li>ピア・ウィンドウの終了後に 1 次データベースが引き続きアクティブである場合、1 次データベースがスタンバイ・データベースにまだ接続していなければ、1 次データベースは切断済みピア状態から解除され、独立してトランザクション処理を再開します。</li> <li>NEARSYNC モードでは、スタンバイ・データベースが 1 次データベースからトランザクション・ログを受け取ったことを認識した後、そのトランザクション・ログ情報をディスクに書き込む前にスタンバイ・データベースに障害が発生した場合には、ログ受信バッファ内のそのトランザクション・ログ情報は失われる可能性があります。</li> </ul>
ローカル・キヤッチアップ	いいえ	エラー・メッセージが戻されます。

スタンバイ状態	BY FORCE オプションを使用	テークオーバー動作
ローカル・キヤッチアップ	はい	エラー・メッセージが戻されます。
ピア	いいえ	<p>1 次データベースとスタンバイ・データベースの役割が切り替えられます。</p> <p>テークオーバー中に障害が発生しなかった場合は、データ損失はありません。しかし、テークオーバー中に障害が検出されると、データ損失が発生する可能性があり、1 次とスタンバイの役割は変更されている場合もあれば、変更されていない場合もあります。以下は、1 次とスタンバイの役割が切り替えられるテークオーバー中に発生した障害を処理するためのガイドラインです。</p> <ol style="list-style-type: none"> <li>1. テークオーバー操作中に障害が発生した場合は、HADR データベースの役割が変更されていることもあれば、変更されていないこともあります。可能であれば、両方のデータベースをオンラインにします。使用できるデータベースの HADR 役割を調べるには、スナップショット・モニターを使用するか、またはデータベース構成パラメーター <code>hadr_db_role</code> の値を確認してください。</li> <li>2. 意図された新しい 1 次データベースがまだスタンバイの役割であり、それでもテークオーバーを発行したい場合は、<b>TAKEOVER HADR</b> コマンドを再発行してください (<b>BY FORCE</b> オプションについての下記のガイドラインを参照)。</li> <li>3. 両方のデータベースがスタンバイの役割になってしまう可能性もあります。その場合、1 次になっている方のノードで、<b>BY FORCE</b> オプションを使って <b>TAKEOVER HADR</b> コマンドを発行できます。この場合、2 つのスタンバイ・データベースが通常の HADR 1 次スタンバイ接続を確立することはできないため、<b>BY FORCE</b> オプションが必要です。</li> </ol>
ピア	はい	<p>古い 1 次が無効とマークされ、それ以上ログを書き込んだりトランザクションをコミットしたりできなくなります。</p> <p>次のログ書き込み試行により、データベースは停止します。ただし、古い 1 次でのセッションが読み取り専用照会のみを実行する場合、古い 1 次は無期限に稼働状態に保たれる可能性があります。既存のクライアント接続は、これらが読み取り専用操作を実行している限り開いたままとなり、新しいクライアント接続も受け入れられます。</p> <p>2 つの 1 次が存在する状況を回避するために、まず古い 1 次のトランザクション処理を停止してから、<b>BY FORCE</b> オプションを指定して <b>TAKEOVER HADR</b> コマンドを実行することを推奨します。</p>

スタンバイ状態	BY FORCE オプションを使用	テークオーバー動作
リモート・キャッチアップ	いいえ	SUPERASYNC モードの場合、リモート・キャッチアップ状態で非強制的テークオーバー操作を実行できます。その他の同期モードでは、エラーが戻されます。非強制的テークオーバー操作を開始する前に、1 次データベースとスタンバイ・データベース間のログ・ギャップを確認してください。スタンバイ・データベースはギャップにあるログを取り出して、それを再生する必要があるため、大きなギャップがあると、テークオーバー操作に長時間を要する原因となります。非強制的テークオーバー操作を実行するのは、ログ・ギャップが小さいときだけにすることをお勧めします。1 次データベースとスタンバイ・データベース間のログ・ギャップを小さくするには、1 次データベースでのワークロードを停止または削減することを考慮してください。
リモート・キャッチアップ	はい	SUPERASYNC モードの場合、リモート・キャッチアップ状態で強制テークオーバー操作を実行できます。その他の同期モードでは、エラーが戻されます。
リモート・キャッチアップ・ペンディング	いいえ	エラー・メッセージが戻されます。
リモート・キャッチアップ・ペンディング	はい	スタンバイ・データベースが 1 次データベースになります。

**TAKEOVER HADR** コマンドを発行すると、対応するエラー・コード

SQL1767N、SQL1769N、または SQL1770N と理由コード 98 が生成される可能性があります。この理由コードは、コマンドが発行されたサーバー上に HADR のライセンスがインストールされていないことを示します。この問題を訂正するには、**db2licm** を使用して有効な HADR ライセンスをインストールするか、または配布の一部として有効な HADR ライセンスが含まれているサーバーのバージョンをインストールします。

**TAKEOVER BY FORCE PEER WINDOW ONLY** コマンドを発行してこれが成功した場合 (1 次データベースがスタンバイ・データベースから切断済みでピア・ウィンドウに残っている時点でこれを発行した場合) には、スタンバイ・データベースにまだコピーされていないトランザクション情報は 1 次データベースには存在しません。

スタンバイ中の読み取りを使用可能にすると、スタンバイ・データベースに現在接続されているユーザー・アプリケーションは切断されて、テークオーバーが行えるようになります。スタンバイ・データベースでアクティブになっている読み取りプログラムの数によっては、スタンバイ・データベースに読み取りプログラムがない場合に比べて、テークオーバー操作を完了するのに時間が多少長くなる可能性があります。役割の切り替え中には新規接続はできません。テークオーバーでの役割の切り替え中に HADR スタンバイに接続しようとする、エラー (SQL1776N) を



受け取ります。

---

## TERMINATE

コマンド行プロセッサのバックエンド・プロセスを明示的に終了させます。

### 許可

なし

### 必要な接続

なし

### コマンド構文

▶—TERMINATE—◀

### コマンド・パラメーター

なし

### 使用上の注意

アプリケーションがデータベースに接続されているかまたは作業単位の途中で、**TERMINATE** コマンドを出すと、データベース接続は失われます。その場合、内部コミットは実行されます。

**TERMINATE** と **CONNECT RESET** は両方ともデータベースへの接続を中断しますが、**TERMINATE** のみがバックエンド・プロセスを終了します。

**TERMINATE** は **db2stop** コマンドを実行するよりも前に出すことをお勧めします。これによりバックエンド・プロセスは、使用できなくなったデータベース・マネージャーのインスタンスへの接続を保持することができなくなります。

セッション中に **DB2NODE** 環境変数が更新された場合、**MPP** システムにおけるバックエンド・プロセスも終了しなければなりません。この環境変数は、**MPP** の複数の論理データベース・パーティションの構成におけるコーディネーター・データベース・パーティション番号を指定する時に使用します。

## UNCATALOG DATABASE

データベース項目をデータベース・ディレクトリーから削除します。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL

### 必要な接続

なし。ディレクトリー操作は、ローカル・ディレクトリーだけに影響します。

### コマンド構文

```

▶▶—UNCATALOG—┬—DATABASE—┬—database-alias—▶▶
                  └—DB—┬—
  
```

### コマンド・パラメーター

**DATABASE** *database-alias*

アンカタログするデータベースの別名を指定します。

### 使用上の注意

アンカタログ可能なのはローカル・データベース・ディレクトリー内の項目だけです。システム・データベース・ディレクトリーにある項目は、**DROP DATABASE** コマンドを使用して削除できます。

インスタンス上のデータベースを再カタログするには、**UNCATALOG DATABASE** および **CATALOG DATABASE** コマンドを使用します。ノードにカタログされているデータベースをリストする場合は、**LIST DATABASE DIRECTORY** コマンドを使用してください。

最初にデータベースをアンカタログし、次に別のタイプを指定してデータベースを再カタログすることにより、以前のサーバーと通信する際に使用される、データベースの認証タイプを変更できます。

ディレクトリーのキャッシュが有効な場合、データベース、ノード、および DCS ディレクトリー・ファイルはメモリーにキャッシュされます。**GET DATABASE MANAGER CONFIGURATION** コマンドの構成パラメーター **dir\_cache** に関する情報を参照してください。アプリケーションのディレクトリー・キャッシュは、最初のディレクトリー検索時に作成されます。キャッシュはアプリケーションがディレクトリー・ファイルのいずれかを修正したときにのみ最新にされるため、他のアプリケーションが行ったディレクトリーの変更は、アプリケーションを再始動するまで有効にならないことがあります。

CLP のディレクトリー・キャッシュを最新表示するには、**TERMINATE** コマンドを使用します。データベース・マネージャーの共有キャッシュを最新表示するには、データベース・マネージャーを停止させてから (**db2stop**)、再始動させます

## UNCATALOG DATABASE

(**db2start**)。別のアプリケーション用のディレクトリー・キャッシュを最新にするには、そのアプリケーションを停止させてから再始動させてください。

**注:** データベース・パーティションをシステムに追加すると、インスタンス内のすべての既存のデータベースは新規データベース・パーティションに展開されます。ただし、アンカタログされたデータベースは、新規データベース・パーティションの追加時には認識されません。アンカタログされたデータベースは、新規データベース・パーティションには存在しません。新規データベース・パーティション上でデータベースに接続しようとする、エラー・メッセージ **SQL1013N** が戻されます。

## UNCATALOG DCS DATABASE

データベース接続サービス (DCS) ディレクトリーから項目を削除します。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL

### 必要な接続

なし。ディレクトリー操作は、ローカル・ディレクトリーだけに影響します。

### コマンド構文

```

▶▶—UNCATALOG DCS—┬─DATABASE—┬─database-alias—▶▶
                    └─DB—┘
  
```

### コマンド・パラメーター

**DATABASE** *database-alias*

アンカタログする DCS データベースの別名を指定します。

### 使用上の注意

DCS データベースは、リモート・データベースとしてシステム・データベース・ディレクトリーにもカタログされており、**UNCATALOG DATABASE** コマンドを使用してアンカタログすることができます。

DCS ディレクトリーでデータベースを再カタログする場合は、**UNCATALOG DCS DATABASE** および **CATALOG DCS DATABASE** コマンドを使用してください。ノードにカタログされている DCS データベースをリストする場合は、**LIST DCS DIRECTORY** コマンドを使用してください。

ディレクトリーのキャッシュが有効な場合、データベース、ノード、および DCS ディレクトリー・ファイルはメモリーにキャッシュされます。**GET DATABASE MANAGER CONFIGURATION** コマンドの出力で、構成パラメーター **dir\_cache** に関して提供されている情報を参照してください。アプリケーションのディレクトリー・キャッシュは、最初のディレクトリー検索時に作成されます。キャッシュはアプリケーションがディレクトリー・ファイルのいずれかを修正したときにのみ最新にされるため、他のアプリケーションが行ったディレクトリーの変更は、アプリケーションを再始動するまで有効にならないことがあります。

CLP のディレクトリー・キャッシュを最新表示するには、**TERMINATE** コマンドを使用します。DB2 の共有キャッシュを最新表示するには、データベースを停止してから (**db2stop**)、再始動します (**db2start**)。別のアプリケーション用のディレクトリー・キャッシュを最新にするには、そのアプリケーションを停止させてから再始動させてください。

## UNCATALOG LDAP DATABASE

Lightweight Directory Access Protocol (LDAP) からデータベース登録解除します。

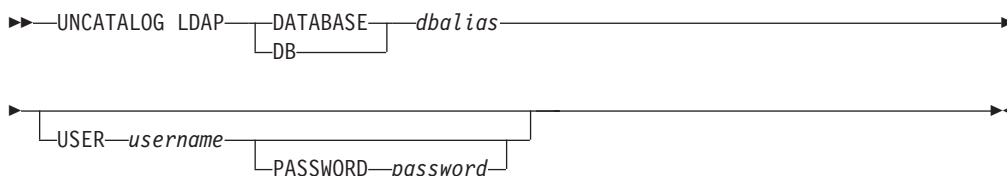
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### DATABASE *dbalias*

アンカタログする LDAP データベースの別名を指定します。

#### USER *username*

ユーザーの LDAP 識別名 (DN) を指定します。LDAP ユーザー DN には、LDAP ディレクトリーからオブジェクトを削除するための十分な権限が必要です。ユーザーの LDAP DN が指定されない場合、現行ログオン・ユーザーの資格情報が使用されます。

#### PASSWORD *password*

アカウント・パスワードを示します。

### 使用上の注意

データベースをドロップすると、データベース・オブジェクトも LDAP からドロップされます。データベースを管理するデータベース・サーバーが LDAP から登録解除されると、データベースも LDAP から自動的に登録解除されます。ただし、次のようなときには、データベースを LDAP から手動でアンカタログしなければならない場合もあります。

- データベース・サーバーが LDAP をサポートしない場合。データベースがドロップされるたびに、管理者はデータベースを LDAP から手動でアンカタログしなければなりません。
- **DROP DATABASE** を実行している間は、データベース・オブジェクトを LDAP からドロップすることができません (その間は LDAP にアクセスできないからです)。その場合、データベースはローカル・マシンから除去されますが、LDAP 内にある既存の項目は削除されません。

## UNCATALOG LDAP NODE

Lightweight Directory Access Protocol (LDAP) のノード項目をアンカタログします。

### 許可

なし

### 必要な接続

なし

### コマンド構文

```
▶▶—UNCATALOG LDAP—NODE—nodename—┬──┬──┬──▶  
└──USER—username—┬──┬──┬──  
└──PASSWORD—password—┬──┬──┬──
```

### コマンド・パラメーター

**NODE** *nodename*

アンカタログするノードの名前を指定します。

**USER** *username*

ユーザーの LDAP 識別名 (DN) を指定します。LDAP ユーザー DN には、LDAP ディレクトリーからオブジェクトを削除するための十分な権限が必要です。ユーザーの LDAP DN が指定されない場合、現行ログオン・ユーザーの資格情報が使用されます。

**PASSWORD** *password*

アカウント・パスワードを示します。

### 使用上の注意

DB2 サーバーが LDAP から登録解除されると、LDAP ノードは自動的にアンカタログされます。



---

## UNCATALOG NODE

ノード・ディレクトリーから項目を削除します。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL

### 必要な接続

なし。ディレクトリー操作は、ローカル・ディレクトリーだけに影響します。

### コマンド構文

▶▶—UNCATALOG NODE—*nodename*—————▶▶

### コマンド・パラメーター

**NODE** *nodename*

アンカタログするノード項目を指定します。

### 使用上の注意

**UNCATALOG NODE** はどのタイプのノードでも実行できますが、リモート・インスタンスや別のローカル・インスタンスへのアタッチがあっても、影響が及ぶのはローカル・ディレクトリーだけです。

ディレクトリーのキャッシュが有効な場合、データベース、ノード、および DCS ディレクトリー・ファイルはメモリーにキャッシュされます。ディレクトリー・キャッシングが有効であるかどうかを確認する場合は、**GET DATABASE MANAGER CONFIGURATION** コマンドの出力内の **dir\_cache** ディレクトリー・キャッシュ・サポート構成パラメーターの値を検査してください。アプリケーションのディレクトリー・キャッシュは、最初のディレクトリー検索時に作成されます。キャッシュはアプリケーションがディレクトリー・ファイルのいずれかを修正したときにのみ最新にされるため、他のアプリケーションが行ったディレクトリーの変更は、アプリケーションを再始動するまで有効にならないことがあります。

CLP のディレクトリー・キャッシュを最新表示するには、**TERMINATE** コマンドを使用します。データベース・マネージャーの共有キャッシュを最新表示するには、データベース・マネージャーを停止させてから (**db2stop**)、再始動させます (**db2start**)。別のアプリケーション用のディレクトリー・キャッシュを最新にするには、そのアプリケーションを停止させてから再始動させてください。

## UNCATALOG ODBC DATA SOURCE

ユーザーまたはシステム ODBC データ・ソースをアンカタログします。

ODBC (Open Database Connectivity) でのデータ・ソース という語は、特定のデータベースのユーザー定義名のことです。この名前は、ODBC API を介してデータベースにアクセスするときに使用されます。Windows では、ユーザー・データ・ソースまたはシステム・データ・ソースのいずれかのアンカタログができます。ユーザー・データ・ソースはそれをカタログしたユーザーにのみ可視になりますが、システム・データ・ソースは他のすべてのユーザーから可視であり使用できます。

このコマンドは Windows のみで使用できます。

### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

**USER** ユーザー・データ・ソースをアンカタログします。キーワードを指定しない場合、これがデフォルトです。

#### SYSTEM

システム・データ・ソースをアンカタログします。

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

アンカタログするデータ・ソースの名前を指定します。最大長は 32 文字です。

### 使用上の注意

Microsoft Windows Vista 以降のバージョンでは、完全な管理者特権で実行する DB2 コマンド・ウィンドウから **UNCATALOG SYSTEM ODBC DATA SOURCE** コマンドを実行する必要があります。

## UNQUIESCE

保守などの理由で静止状態になっていたインスタンスまたはデータベースに対するユーザー・アクセスを回復します。**UNQUIESCE** コマンドにより、シャットダウンおよびデータベースの再始動をしなくても、ユーザー・アクセスを回復できます。

### 有効範囲

**UNQUIESCE DB** は、静止データベース内のすべてのオブジェクトに対するユーザー・アクセスを回復します。

**UNQUIESCE INSTANCE** *instance-name* は、インスタンス *instance-name* 内のインスタンスおよびすべてのデータベースに対するユーザー・アクセスを回復します。

インスタンスを停止した後、そのインスタンスとそのすべてのデータベースの静止を解除するには、**db2stop** コマンドを発行します。DB2 を停止し、再開すると、すべてのインスタンスとデータベースの静止が解除されます。

### 許可

以下の権限のいずれか。

データベース・レベルの静止解除の場合:

- SYSADM
- DBADM

インスタンス・レベルの静止解除の場合:

- SYSADM
- SYSCTRL

### コマンド構文

```

▶▶ UNQUIESCE — DB —————▶▶
      |
      └─ INSTANCE — instance-name
  
```

### 必要な接続

データベース

(インスタンスの静止解除にはデータベース接続は必須ではありません。)

### コマンド・パラメーター

**DB** データベースの静止解除。データベース内のすべてのオブジェクトに対するユーザー・アクセスが回復されます。

**INSTANCE** *instance-name*  
 インスタンス *instance-name* およびインスタンス内のすべてのデータベースに対するアクセスが回復されます。

## 例：データベースの静止解除

次のコマンドは、以前に静止されていたデータベースの静止を解除します。

```
db2 unquiesce db
```

## インスタンスの静止解除

次のコマンドは、以前に静止されていたインスタンス `instA` の静止を解除します。

```
db2 unquiesce instance instA
```

## 使用上の注意

- DB2 pureScale 環境では、データベースの静止およびインスタンスの再始動の後、データベースはすべてのメンバーにわたり静止状態のままになります。静止状態を解除するには、明示的な **UNQUIESCE DATABASE** コマンド呼び出しが必要です。

### UPDATE ADMIN CONFIGURATION

DB2 Administration Server (DAS) 構成ファイル内の指定された項目を編集します。DAS は、DB2 サーバーのリモート管理を有効にする特別な管理ツールです。

**重要:** DB2 Administration Server (DAS) は、バージョン 9.7 で非推奨となり、将来のリリースで除去される可能性があります。DAS は、DB2 pureScale環境ではサポートされていません。リモート管理のためには、Secure Shell プロトコルを使用するソフトウェア・プログラムを使用してください。詳しくは、『DB2 Administration Server (DAS) が推奨されなくなった』( ) を参照してください。

DAS をインストールすると、構成ファイルのブランクのコピーが各物理データベース・パーティションに保管されます。この各コピーの中に項目を作成する必要があります。以下の DAS 構成パラメーターは、次に DAS を始動するときに使用するものとして指定できます。

- DB2 サーバー・システムの名前 - **db2system**
- DAS 管理者権限グループ名 - **dasadm\_group**
- スケジューラー・モード - **sched\_enable**
- ツール・カタログ・データベース・インスタンス - **toolscat\_inst**
- ツール・カタログ・データベース - **toolscat\_db**
- ツール・カタログ・データベース・スキーマ - **toolscat\_schema**
- 実行有効期限切れタスク - **exec\_exp\_task**
- スケジューラー・ユーザー ID - **sched\_userid**
- 認証タイプ DAS - **authentication**

以下の DAS 構成パラメーターは、元々指定可能なもので、後で DAS がオンラインになっているときに変更できます。

- DAS ディスカバリー・モード - **discover**
- SMTP サーバー - **smtp\_server**
- Java 開発キット・インストール・パス DAS - **jdk\_path**
- 連絡先リストのロケーション - **contact\_host**
- DAS コード・ページ - **das\_codepage**
- DAS テリトリー - **das\_territory**
- 診断エラー・キャプチャー・レベル - **diaglevel**

これらのパラメーターについての詳細は、個々のパラメーターの説明を参照してください。

#### 有効範囲

このコマンドは、ノードのパラメーター設定を指定または変更するときに、各管理ノードから発行してください。

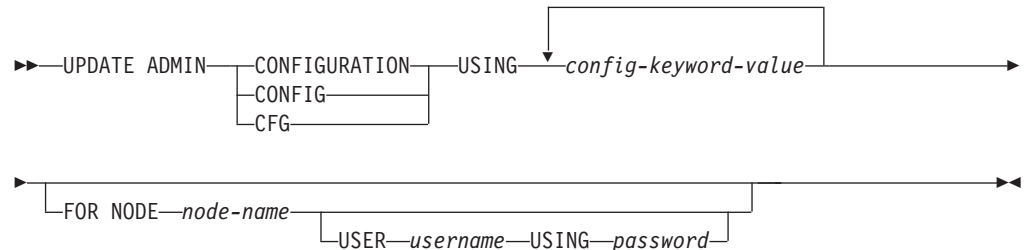
#### 許可

DASADM

## 必要な接続

ノード。リモート・システムの DAS 構成を更新する場合は、**FOR NODE** オプションと管理ノード名を使用します。

## コマンド構文



## コマンド・パラメーター

**USING** *config-keyword-value*

更新する admin 構成パラメーターを指定します。

**FOR NODE**

DAS 構成パラメーターを更新する管理ノードの名前をここに入力します。

**USER** *username* **USING** *password*

管理ノードへの接続にユーザー名とパスワードの許可が必要な場合は、この情報を入力します。

## 使用上の注意

DAS 構成パラメーターのリストの表示または印刷を行うには、**GET ADMIN CONFIGURATION** を使用してください。DAS 構成パラメーターを推奨されている DAS のデフォルトにリセットするには、**RESET ADMIN CONFIGURATION** を使用します。

構成パラメーターが有効になるタイミングは、標準の構成パラメーターを変更するか、オンラインでリセット可能なパラメーターの 1 つを変更するかによって異なります。標準の構成パラメーターの値は、**db2admin** コマンドが実行されたときにリセットされます。

エラーが生じた場合には、DAS 構成ファイルは変更されません。

**UPDATE ADMIN CONFIGURATION** を使用して DAS 構成を更新するには、DAS と同じインストール済みレベルにあるインスタンスからコマンド行プロセッサを使用する必要があります。

DAS 構成ファイルは、そのチェックサムが無効であると、更新することができません。このような状況は、適切なコマンドを使用せずに手作業で DAS 構成ファイルを変更した場合などに起こります。このような状況になったら、DAS を一度ドロップしてから再作成し、その構成ファイルをリセットする必要があります。

## UPDATE ALERT CONFIGURATION

ヘルス・インディケータのアラート構成設定を更新します。

**重要:** バージョン 9.7 でヘルス・モニターが非推奨になったため、このコマンドおよび API は推奨されておらず、将来のリリースで除去される可能性があります。これはDB2 pureScale環境ではサポートされていません。詳しくは、『ヘルス・モニターが推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055045.html>) を参照してください。

### 許可

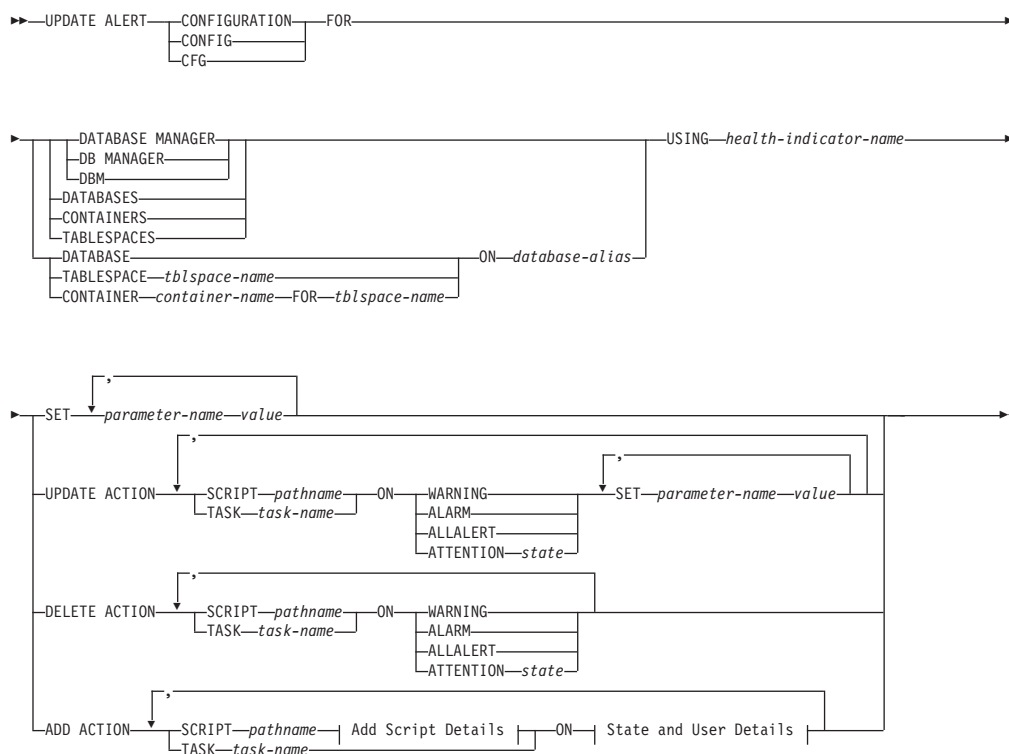
以下の権限のいずれか。

- SYSADM
- SYSMANT
- SYSCTRL

### 必要な接続

インスタンス。明示的なアタッチは必要ありません。

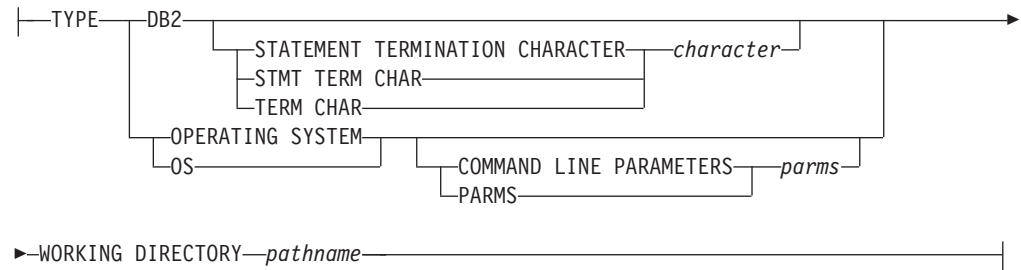
### コマンド構文



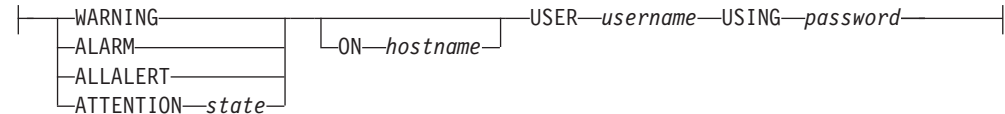
### Add Script Details:



## UPDATE ALERT CONFIGURATION



### State and User Details:



## コマンド・パラメーター

### DATABASE MANAGER

データベース・マネージャーのアラート設定を更新します。

### DATABASES

データベース・マネージャーによって管理されるすべてのデータベースのアラート設定を更新します。これは、カスタム設定を持たないすべてのデータベースに適用される設定です。カスタム設定は、**DATABASE ON database-alias** 節を使って定義されます。

### CONTAINERS

データベース・マネージャーによって管理されるすべての表スペース・コンテナのアラート設定を更新します。これは、カスタム設定を持たないすべての表スペース・コンテナに適用される設定です。カスタム設定は、**CONTAINER container-name ON database-alias** 節を使って定義されます。

### TABLESPACES

データベース・マネージャーによって管理されるすべての表スペースのアラート設定を更新します。これは、カスタム設定を持たないすべての表スペースに適用される設定です。カスタム設定は、**TABLESPACE tblspace-name ON database-alias** 節を使って定義されます。

### DATABASE ON database-alias

**ON database-alias** 節を使って指定したデータベースのアラート設定を更新します。このデータベースがカスタム設定を持つ場合、インスタンスの全データベースの設定をオーバーライドします。これは、**DATABASES** パラメーターを使って指定されます。

### CONTAINER container-name FOR tblspace-name ON database-alias

**ON database-alias** 節を使って指定したデータベース上で、**FOR tblspace-name** 節を使って指定した表スペースの、*container-name* という名前の表スペース・コンテナのアラート設定を更新します。この表スペース・コンテナがカスタム設定を持つ場合、データベースの全表スペース・コンテナの設定をオーバーライドします。これは、**CONTAINERS** パラメーターを使って指定されます。

## UPDATE ALERT CONFIGURATION

### **TABLESPACE** *tblspace-name* **ON** *database-alias*

**ON** *database-alias* 節を使って指定したデータベース上で、*name* という名前の表スペースのアラート設定を更新します。この表スペースがカスタム設定を持つ場合、データベースの全表スペースの設定をオーバーライドします。これは、**TABLESPACES** パラメーターを使って指定されます。

### **USING** *health-indicator-name*

アラート構成が更新されるヘルス・インディケーターの設定を指定します。ヘルス・インディケーター名は 2 文字のオブジェクト ID で構成され、その後にインディケーターの測定対象を説明する名前が続きます。以下に例を示します。

```
db.sort_privmem_util
```

### **SET** *parameter-name* *value*

ヘルス・インディケーターのアラート構成エレメント *parameter-name* を、指定した値に更新します。 *parameter-name* は以下の値のどれかになります。

- **ALARM:** *value* はヘルス・インディケーター・ユニット。
- **WARNING:** *value* はヘルス・インディケーター・ユニット。
- **SENSITIVITY:** *value* は秒単位。
- **ACTIONSENABLED:** *value* は YES または NO。
- **THRESHOLDSCHECKED:** *value* は YES または NO。

### **UPDATE ACTION SCRIPT** *pathname* **ON** [**WARNING** | **ALARM** | **ALLALERT** | **ATTENTION**

*state*] 絶対パス名 *pathname* を持つ定義済みスクリプトのスクリプト属性が以下の節に従って更新されるように指定します。

### **SET** *parameter-name* *value*

スクリプト属性 *parameter-name* を、指定した値に更新します。  
*parameter-name* は以下の値のどれかになります。

- **SCRIPTTYPE**

有効なタイプは OS または DB2。

- **WORKINGDIR**
- **TERMCHAR**
- **CMDLINEPARMS**

オペレーティング・システム・スクリプトに対して指定するコマンド行パラメーターが、デフォルトで指定されるパラメーターに先行します。オペレーティング・システム・スクリプトに送られるパラメーターは、以下のとおりです。

- ユーザーの指定するパラメーターのリスト
- ヘルス・インディケーターの短縮名
- 完全修飾オブジェクト名
- ヘルス・インディケーターの値
- アラート状態

- **USERID**
- **PASSWORD**

- SYSTEM

**UPDATE ACTION TASK** *task-name* **ON** [WARNING | ALARM | ALLALERT | ATTENTION *state*]  
名前 *name* を持つタスクのタスク属性が以下の節に従って更新されるように指定します。

**SET** *parameter-name value*

タスク属性 *parameter-name* を、指定した値に更新します。  
*parameter-name* は以下の値のどれかになります。

- USERID
- PASSWORD
- SYSTEM

**DELETE ACTION SCRIPT** *pathname* **ON** [WARNING | ALARM | ALLALERT | ATTENTION *state*]  
アラート・アクション・スクリプトから、絶対パス名 *pathname* を持つアクション・スクリプトを除去します。

**DELETE ACTION TASK** *task-name* **ON** [WARNING | ALARM | ALLALERT | ATTENTION *state*]  
アラート・アクション・タスクのリストから *name* という名前のアクション・タスクを除去します。

**ADD ACTION SCRIPT** *pathname* **ON** [WARNING | ALARM | ALLALERT | ATTENTION *state*]  
絶対パス名 *pathname* を持つ新規アクション・スクリプトが追加されるように指定します。その属性は、以下のように指定されます。

**TYPE** アクション・スクリプトは、DB2 コマンド・スクリプトか、オペレーティング・システム・スクリプトのいずれかでなければなりません。

- DB2
- OPERATING SYSTEM

DB2 コマンド・スクリプトの場合、以下の節を使用することにより、オプションで文字 *character* を指定することができます。この文字は、ステートメントを終了するのにスクリプト内で使用されません。

STATEMENT TERMINATION CHARACTER ;

オペレーティング・システム・スクリプトの場合、以下の節を使用することにより、オプションでコマンド行パラメーター *parms* を指定することができます。これは、呼び出しの際にスクリプトに渡されます。 **COMMAND LINE PARAMETERS** *parms*

**WORKING DIRECTORY** *pathname*

スクリプトが実行されるディレクトリーの絶対パス名 *pathname* を指定します。

**USER** *username* **USING** *password*

スクリプトが実行される際のユーザー・アカウント *username*、およびそれに関連したパスワード *password* を指定します。

**ADD ACTION TASK** *name* **ON** [WARNING | ALARM | ALLALERT | ATTENTION *state*]  
指定した条件を **ON** にして *name* という新規タスクを追加し、実行することを指定します。

## UPDATE ALERT CONFIGURATION

### ON [WARNING | ALARM | ALLALERT | ATTENTION *state*]

アクションまたはタスクが実行される条件を指定します。しきい値ベースのヘルス・インディケーター (HI) の場合、これは **WARNING** または **ALARM** になります。状態ベースの HI の場合、これは、各状態ベースの HI ごとに記されている数値状態になります (例えば、ts.ts\_op\_status ヘルス・インディケーターの場合は、表スペース状態の **tablespace\_state** モニター・エレメントを参照してください)。 **ALLALERTS** は、しきい値ベースの HI と状態ベースの HI に関する状態の変更を処理します (例えば、警告から正常への状態変更)。

### ATTENTION *state*

**ADD ACTION SCRIPT CLP** コマンド・オプションの例として、データベース・ヘルス・インディケーターのいくつかの状態に対応した有効な数値を以下のセクションにまとめます。

- 0 - アクティブ、正常 (ACTIVE)
- 1 - 静止ペンディング (QUIESCE\_PEND)
- 2 - 静止済み (QUIESCED)
- 3 - ロールフォワード (ROLLFWD)

その他の状態ベースのヘルス・インディケーターは、ヘッダー・ファイル `sqlmon.h` と `sqlutil.h` に定義されています。

## 使用上の注意

**ADD ACTION** オプションでは、入力した *username* および *password* は、SQL ステートメント・テキストがキャプチャーされる以下のさまざまな場所で公開される場合があります。

- ネットワーク (*username* と *password* は暗号化されずにワイヤーを經由して渡されます)
- **db2diag** ログ・ファイル
- トレース・ファイル
- ダンプ・ファイル
- スナップショット・モニター (動的 SQL スナップショット)
- システム・モニター・スナップショット
- いくつかのイベント・モニター (ステートメント、デッドロック)
- Explain 表
- **db2pd** 出力 (パッケージ・キャッシュおよびロック・タイムアウト機構など)
- DB2 監査レコード

## UPDATE ALTERNATE SERVER FOR DATABASE

データベース別名に関連した代替サーバーをシステム・データベース・ディレクトリー内で更新します。

### 有効範囲

このコマンドは、それが実行されたデータベース・パーティションに対してだけ影響を与えます。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL

### 必要な接続

なし

### コマンド構文

```

▶▶ UPDATE ALTERNATE SERVER FOR DATABASE database-alias USING DB
▶▶ HOSTNAME hostname PORT port-number

```

### コマンド・パラメーター

**DATABASE** *database-alias*

代替サーバーを更新するデータベースの別名を指定します。

**HOSTNAME** *hostname*

そのデータベースの代替サーバーが存在するノードの完全修飾ホスト名または IP アドレスを指定します。

**PORT** *port-number*

データベース・マネージャー・インスタンスの代替サーバーのポート番号を指定します。

### 例

以下の例は、ホスト名 `montero` とポート `20396` を使用して、`SAMPLE` データベースの代替サーバーを更新します。

```
db2 update alternate server for database sample using hostname montero port 20396
```

以下の 2 つの例は、`SAMPLE` データベースの代替サーバーをリセットします。

```
db2 update alternate server for database sample using hostname NULL port NULL
```

または

```
db2 update alternate server for database sample using hostname "" port NULL
```

## UPDATE ALTERNATE SERVER FOR DATABASE

### 使用上の注意

- このコマンドは、システム・データベース・ディレクトリーに対してのみ適用されます。
- このコマンドは、サーバー・インスタンスでのみ使用するべきです。クライアント・インスタンスで発行すると、このコマンドは無視され、メッセージ SQL1889W が返されます。
- このコマンドを発行するマシンで Lightweight Directory Access Protocol (LDAP) サポートが有効になっている場合、データベースの代替サーバーは自動的に LDAP ディレクトリーに登録されます。

## UPDATE ALTERNATE SERVER FOR LDAP DATABASE

データベースに関連した代替サーバーを Lightweight Directory Access Protocol (LDAP) 内で更新します。

### 許可

LDAP サーバーへの読み取り/書き込みアクセス。

### 必要な接続

なし

### コマンド構文

```

▶—UPDATE ALTERNATE SERVER FOR LDAP—DATABASE—database-alias—▶
      |
      |—DB—
▶—USING—NODE—node—▶
      |
      |—GWNODE—gwnode—|—USER—username—|—PASSWORD—password—▶
  
```

### コマンド・パラメーター

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

更新するデータベースの別名を指定します。

#### **NODE** *node*

データベースの代替サーバーが置かれているノード名を指定します。

#### **GWNODE** *gwnode*

データベースの代替ゲートウェイが置かれているノード名を指定します。

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

ユーザーの LDAP 識別名 (DN) を指定します。LDAP ユーザー DN には、LDAP ディレクトリーでオブジェクトを作成するための十分な権限が必要です。ユーザーの LDAP DN が指定されない場合、現行ログオン・ユーザーの資格情報が使用されます。

ユーザーの LDAP DN およびパスワードが **db21dcfg** を使用して指定されている場合、ユーザー名とパスワードをここで指定する必要はありません。

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

アカウント・パスワードを示します。

ユーザーの LDAP DN およびパスワードが **db21dcfg** を使用して指定されている場合、ユーザー名とパスワードをここで指定する必要はありません。



## UPDATE CLI CONFIGURATION

db2cli.ini ファイル内の指定されたセクションの内容を更新します。

db2cli.ini ファイルは、DB2 コール・レベル・インターフェース (CLI) 構成ファイルとして使用されます。このファイルには、CLI およびそれを使用するアプリケーションの動作を変更するために使用できるさまざまなキーワードと値が含まれます。このファイルは複数のセクションに分かれており、それぞれのセクションはデータベース別名に対応します。

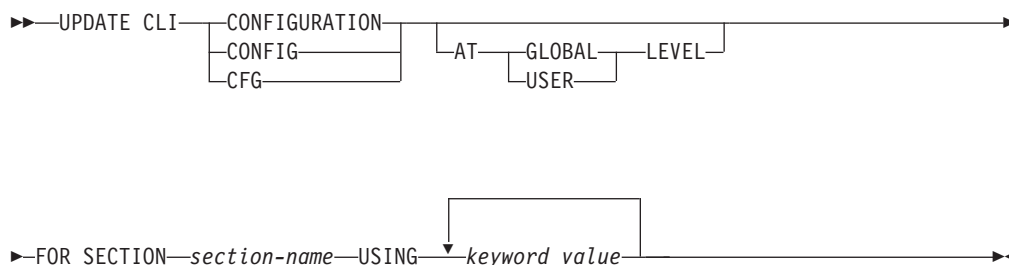
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### AT GLOBAL LEVEL

CLI 構成パラメーターをグローバル・レベルで更新するよう指定します。グローバル・レベルで CLI 構成パラメーターを更新するには、ユーザー ID に Active Directory システム権限がなければなりません。この権限がない場合、エラー SQL3267N で更新が失敗します。このパラメーターを適用できるのは、LDAP サポートが有効な場合だけです。

#### AT USER LEVEL

CLI 構成パラメーターをユーザー・レベルで更新するよう指定します。LDAP サポートが有効になっている場合は、同じ LDAP ユーザー ID を使用して別のマシンにログオンするときでも、この設定は変わりません。LDAP サポートが無効になっている場合は、同じオペレーティング・システム・ユーザー ID を使用して同じマシンにログオンするときだけ、この設定は変わりません。

#### FOR SECTION *section-name*

キーワードが更新されるセクションの名前。指定されたセクションが存在しない場合は、新しいセクションが作成されます。

#### USING *keyword value*

更新される CLI/ODBC パラメーターを指定します。

## 使用上の注意

このコマンドで指定されるセクション名とキーワードでは、大文字小文字が区別されません。ただし、キーワードの値では大文字小文字が区別されます。

キーワード値が単一引用符または組み込みブランクを含むストリングである場合には、ストリング全体を二重引用符で囲む必要があります。以下に例を示します。

```
db2 update cli cfg for section tstcli1x
    using TableType "'TABLE','VIEW','SYSTEM TABLE'"
```

**AT USER LEVEL** キーワードを指定した場合、指定されたセクションの CLI 構成パラメーターは現行ユーザーについてのみ更新されます。指定しなかった場合は、ローカル・マシン上のすべてのユーザーについて更新されます。ユーザー・レベルの CLI 構成は、LDAP ディレクトリーに保持され、ローカル・マシンでキャッシュされます。CLI 構成を読み取るとき、DB2 は常にキャッシュから読み取ります。キャッシュは、次のときに更新されます。

- ユーザーが CLI 構成を更新するとき。
- ユーザーが **REFRESH LDAP** コマンドを使用して、明示的に CLI 構成の最新表示を強制するとき。

LDAP 環境では、ユーザーは LDAP ディレクトリーにカタログされたデータベースに対して、デフォルト CLI 設定値のセットを構成することができます。LDAP カタログ・データベースが、DSN (データ・ソース名) として、ODBC 構成ユーティリティーを使用して追加されると、デフォルトの CLI 設定が LDAP ディレクトリーにある場合には、それらはローカル・マシン上のその DSN 用に構成されます。CLI パラメーターをデフォルト設定として構成するには、**AT GLOBAL LEVEL** 節を指定する必要があります。

LDAP 環境で CLI 構成パラメーターを削除するには、ユーザー ID に「システム/IBM」Active Directory コンテナにおける DELETE 権限がなければなりません。WRITE 権限と CREATE ALL CHILD OBJECTS 権限を持つユーザー ID の場合、CLI 構成パラメーターを無効にできますが、削除することはできません。

## UPDATE COMMAND OPTIONS

対話式セッションの間に、またはバッチ入力ファイルから、1 つ以上のコマンド・オプションを設定します。対話式セッションまたはバッチ入力ファイルが終了すると、設定値はシステム・デフォルトに戻ります (または **DB2OPTIONS** はオーバーライドされてシステム・デフォルトになります)。

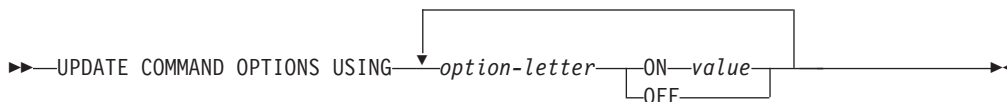
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

**USING** *option-letter*

次の *option-letters* を設定できます。

- a SQLCA の表示
- b DB2 V10.1 フィックスパック 2 以降、このオプションは、SQL ステートメントを実行するために、欠落パッケージや無効パッケージを自動バインドします。
- c SQL ステートメントを自動コミットします
- d XML データの XML 宣言を表示します。
- e SQLCODE および SQLSTATE を表示します
- i XQuery の結果を正しく字下げして表示します。
- l 履歴ファイルへのコマンドのログ書き出し
- m ステートメント INSERT、DELETE、UPDATE、または MERGE によって影響を受ける行数を表示します
- n 改行文字の除去
- o 標準出力への表示
- p DB2 対話式プロンプトの表示
- q 単一または二重引用符で区切られたストリング内の空白および改行を保持します。
- r 出力レポートのファイルへの保存
- s コマンド・エラー時の実行停止
- v 現行コマンドのエコー

- w SQL ステートメント警告メッセージの表示
- z 全出力のファイルへのリダイレクト

**ON value**

e、l、r、および z オプションは、オンにする場合に値が必要です。e オプションでは、*value* を、SQLCODE を表示する場合は c、SQLSTATE を表示する場合は s にできます。l、r、および z オプションの場合、*value* は、履歴ファイルまたはレポート・ファイルに対して使用する名前を表します。他のオプションは値を受け付けません。

**使用上の注意**

これらの設定値は、システム・デフォルト、**DB2OPTIONS** の設定値、およびコマンド行オプション・フラグで指定したオプションをオーバーライドします。

ファイル入力オプション (-f)、およびステートメント終了オプション (-t) は、このコマンドを使用して更新できません。

現行オプションの設定値を表示する場合は、**LIST COMMAND OPTIONS** コマンドを使用してください。

## UPDATE CONTACT

ローカル・システムで定義される連絡先の属性を更新します。連絡先とは、スケジューラーおよびヘルス・モニターがメッセージを送信する先のユーザーです。

連絡先を作成するには、**ADD CONTACT** コマンドを使用します。Database Administration Server (DAS) の **contact\_host** 構成パラメーターの設定により、リストがローカルかグローバルかが決まります。

### 許可

なし

### 必要な接続

なし。ローカル実行のみ: このコマンドはリモート接続では使用できません。

### コマンド構文

```

▶▶ UPDATE CONTACT name USING keyword value

```

### コマンド・パラメーター

**UPDATE CONTACT** *name*

更新される連絡先の名前。

**USING** *keyword value*

更新される連絡先パラメーター (*keyword*) および設定される値 (*value*) を指定します。有効なキーワードのセットは次のとおりです。

#### ADDRESS

SMTP サーバーが通知を送信するのに使用する E メール・アドレス。

**TYPE** アドレスが E メール・アドレスか、ページャーかを指定します。

#### MAXPAGELEN

ページャーが受信できる最大文字数。

#### DESCRIPTION

連絡先のテキスト記述。長さは、最大 128 文字です。

## UPDATE CONTACTGROUP

ローカル・システムで定義される連絡先グループの属性を更新します。連絡先グループは、スケジューラーおよびヘルス・モニターから通知を受け取るユーザーのリストです。

Database Administration Server (DAS) の **contact\_host** 構成パラメーターの設定により、リストがローカルかグローバルかが決まります。

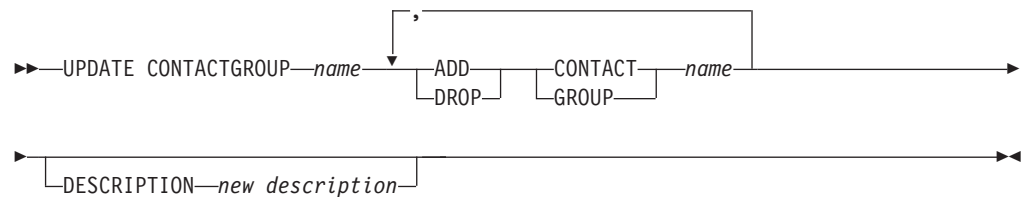
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

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

更新される連絡先グループの名前。

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

グループに追加される新しい連絡先の名前を指定します。グループに追加された後、**ADD CONTACT** コマンドを使用して連絡先を定義できます。

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

グループからドロップされる、グループ中の連絡先の名前を指定します。

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

グループに追加される新しい連絡先グループの名前を指定します。

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

グループからドロップされる、連絡先グループの名前を指定します。

#### **DESCRIPTION** *new description*

オプション。連絡先グループの新しいテキスト記述。

## UPDATE DATABASE CONFIGURATION

特定のデータベース構成ファイルの中の個々の項目を修正します。データベース構成ファイルは、データベースが作成されたデータベース・パーティションすべてに存在しています。

### 有効範囲

このコマンドはデフォルトですべてのデータベース・パーティションまたはメンバーを更新します。ただし、以下のオプションの節が指定されている場合は除きます。

- **MEMBER** を指定して、DB2 pureScale 環境の 1 つのデータベース・メンバーのみを更新する場合、またはパーティション・データベース環境の 1 つのデータベース・パーティションのみを更新する場合。

### 許可

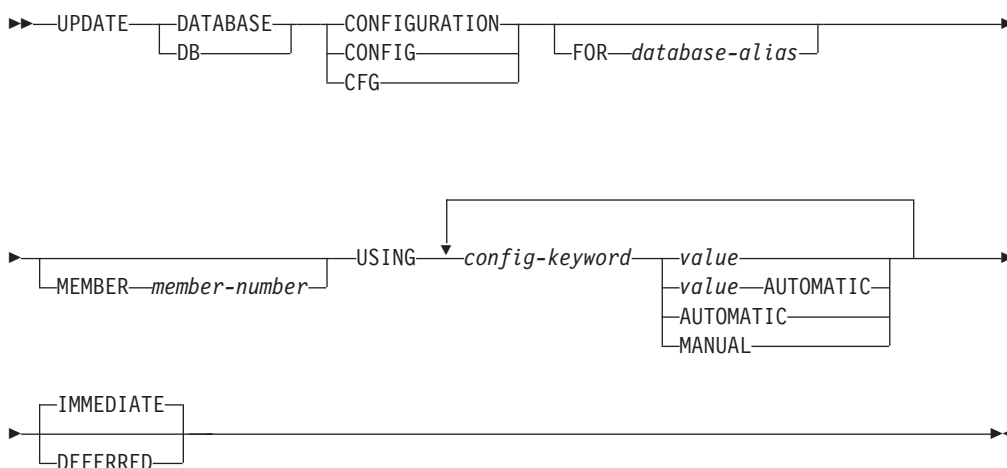
以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMANT

### 必要な接続

インスタンス。明示的なアタッチは必要ありませんが、データベースがアクティブになっているときは、データベース接続が勧められています。データベースがリモートとして示されている場合、リモート・ノードへのインスタンス・アタッチはコマンドの持続期間の間、ずっと確立されたままになります。パラメーターをオンラインにするには、データベースへの接続が必要です。

### コマンド構文





## コマンド・パラメーター

### FOR *database-alias*

構成を更新するデータベースの別名を指定します。データベース接続が既に確立されている場合は、データベース別名を指定する必要はありません。同じデータベース・インスタンスにある別のデータベースについては、その構成ファイルを更新できます。例えば、データベース db11 にのみ接続されている場合に `update db config for alias db22 using .... immediate` を発行すると、

- db22 上にアクティブな接続がない場合、更新の必要なのは構成ファイルだけであるため、更新は成功します。新しい接続 (それによりデータベースがアクティブになる) により、メモリー内で新しい変更が認識されるようになります。
- db22 上に他のアプリケーションからのアクティブな接続があるなら、更新はディスク上で動作しますが、メモリー内では動作しません。データベースを再始動する必要があることを示す警告を受け取るようになります。

### MEMBER *member-number*

**MEMBER** 節では、変更適用の対象にするメンバーを指定します。この節を省略すると、結果としてすべてのメンバーに対して変更が適用されることになります。

### USING *config-keyword value*

*config-keyword* は、更新するデータベース構成パラメーターを指定します。*value* は、パラメーターに割り当てる値を指定します。

### AUTOMATIC

いくつかの構成パラメーターは **AUTOMATIC** に設定できます。それにより、DB2 データベース・システムが自動的にこれらのパラメーターを調整し、現行のリソース要件を反映します。**AUTOMATIC** キーワードをサポートする構成パラメーターのリストは、構成パラメーターのサマリーを参照してください。**AUTOMATIC** キーワードと一緒に値を指定すると、自動計算に影響する可能性があります。この動作の具体的な詳細については、構成パラメーターの資料を参照してください。

**MANUAL** 構成パラメーターの自動チューニングを使用不可にします。パラメーターはその現行の内部値に設定され、自動的な更新は行われなくなります。

### IMMEDIATE

データベースが稼働している場合に、即時に変更を行います。**IMMEDIATE** はデフォルトのアクションですが、このアクションを実行するためにはデータベース接続が有効でなければなりません。

これは、CLPPlus インターフェースで操作する場合のデフォルトの節でもあります。**IMMEDIATE** は、CLPPlus プロセッサの使用時に呼び出す必要はありません。

### DEFERRED

構成ファイルでのみ変更を行います。したがって、加えられた変更は、次にデータベースが再活動化されるときに有効になります。

## UPDATE DATABASE CONFIGURATION

### 例

#### マルチパーティション・インスタンス上でのデータベース構成の更新

SAMPLE という名前のデータベースで、データベース構成パラメーター **MAXAPPLS** を 10 から 50 に更新する例を以下に示します。

ユーザーは、db2nodes.cfg で定義された次の 4 個のパーティションがあるマルチパーティション・インスタンスを持っています。

```
10 gilera 0
20 gilera 1
30 motobi 0
40 motobi 1
```

ユーザーはインスタンス上で **SAMPLE** データベースを作成しました。SAMPLE のカタログ・パーティションはメンバー 10 にあります。ユーザーがシステム **motobi** にログオンしたと想定しましょう。

マルチパーティション・インスタンスのデフォルトの振る舞いはすべてのデータベース・パーティション上でデータベース構成を更新することなので、ユーザーによって発行される次のコマンドにより、すべてのデータベース・パーティションで **MAXAPPLS** の値は同じになります。

```
db2 update db cfg for sample using maxappls 50
```

メンバー 30 だけで **MAXAPPLS** を更新する場合は、以下のコマンドを実行できます。

```
db2 update db cfg for sample member 30 using maxappls 50
```

### 使用上の注意

データベース構成パラメーターのリストを表示または印刷するには、**GET DATABASE CONFIGURATION** コマンドを使用してください。

すべてのデータベース構成パラメーターを推奨されているデフォルトにリセットするには、**RESET DATABASE CONFIGURATION** コマンドを使用してください。

データベース構成パラメーターを変更するには、**UPDATE DATABASE CONFIGURATION** コマンドを使用してください。例えば、ロギング・モードを「archival logging」に、**ZELLMART** というデータベースを含む単一パーティション・データベース環境で変更する場合は、次を使用します。

```
db2 update db cfg for zellmart using logarchmeth1 logretain
```

**logarchmeth1** 構成パラメーターが変更されたことをチェックするには、次を使用します。

```
db2 get db cfg for zellmart
```

例えば、レジストリー変数 **DB2\_UPDDBCFG\_SINGLE\_DBPARTITION** がデフォルトで **NULL** または **FALSE** に設定されている場合に、zellmart というデータベースが含まれている複数パーティション・データベース環境のすべてのパーティションでロギング・モードを **archival logging** に変更する場合は、以下のようなコマンドを使用します。

```
db2 update db cfg for zellmart using logarchmeth1 logretain
```

**Logarchmeth1** 構成パラメーターが、すべてのデータベース・パーティションで変更されたことをチェックするには、次を使用します。

```
db2_all ";db2 get db cfg for zellmart"
```

上記の例を使用して、ロギング・モードを 1 つの特定パーティション (30) だけで更新する場合は、以下のようなコマンドを使用します。

```
db2 update db cfg for zellmart member 30 using logarchmeth1 logretain
```

選択によっては、SYSIBMADM.DBCFG ビューを活用すれば、**db2\_all** を使用しなくても、すべてのパーティションからデータを取得することができます。

UNIX オペレーティング・システムで作業を行っており、**grep** コマンドがある場合は、以下のコマンドを使用して、**logarchmeth1** 値のみを表示できます。

```
db2_all ";db2 get db cfg for zellmart | grep -i logarchmeth1"
```

DB2 データベース構成パラメーターと、各種データベース・ノードに使用できる値についての詳細は、個々の構成パラメーターの説明を参照してください。これらのパラメーターの値は、構成するデータベース・ノードの各タイプ (サーバー、クライアント、またはリモート・クライアントを持つサーバー) によって異なります。

すべてのパラメーターを更新できるわけではありません。

データベース構成ファイルへの変更の一部は、ファイルがメモリーにロードされた後にのみ有効になります。これを行う前にすべてのアプリケーションはデータベースから切断されている必要があります。オンラインで構成できるパラメーターと構成できないパラメーターについては、構成パラメーターの一覧をご覧ください。

例えば、SALES データベースの **sortheap** データベース構成パラメーターをオンラインで変更するには、次のようなコマンドを入力します。

```
db2 connect to sales
db2 update db cfg using sortheap 1000
db2 connect reset
```

エラーが発生した場合、データベース構成ファイルは変更されません。チェックサムが無効な場合、データベース構成ファイルは更新できません。適切なコマンドを使用しないでデータベース構成ファイルを変更するとこれが発生することがあります。これが発生する場合、データベースをリストアしてデータベース構成ファイルをリセットする必要があります。

## 互換性

以前のバージョンとの互換性:

- **DB2\_ENFORCE\_MEMBER\_SYNTAX** レジストリー変数が ON に設定されている場合を除き、**DBPARTITIONNUM** を **MEMBER** の代わりに使用できます。

## UPDATE DATABASE MANAGER CONFIGURATION

データベース・マネージャーの構成ファイルの中の個々の項目を修正します。

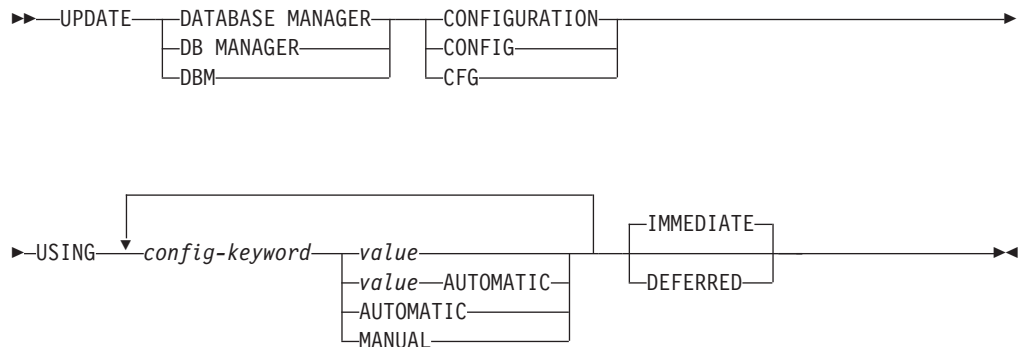
### 許可

SYSADM

### 必要な接続

なし、またはインスタンス。インスタンスとのアタッチは、ローカルの DBM 構成操作を実行する場合には必ずしも必要ではありませんが、リモートの DBM 構成操作の場合には必須です。リモート・インスタンスに対するデータベース・マネージャー構成を更新するためには、最初にそのインスタンスにアタッチする必要があります。構成パラメーターをオンラインで更新する場合も、まずインスタンスにアタッチする必要があります。

### コマンド構文



### コマンド・パラメーター

**USING** *config-keyword value*

更新するデータベース・マネージャー構成パラメーターを指定します。構成パラメーターのリストは、構成パラメーターのサマリーを参照してください。 *value* は、パラメーターに割り当てる値を指定します。

#### **AUTOMATIC**

いくつかの構成パラメーターは **AUTOMATIC** に設定できます。それにより、DB2 が自動的にこれらのパラメーターを調整し、現行のリソース要件を反映します。 **AUTOMATIC** キーワードをサポートする構成パラメーターのリストは、構成パラメーターのサマリーを参照してください。 **AUTOMATIC** キーワードと一緒に値を指定すると、自動計算に影響する可能性があります。この動作の具体的な詳細については、構成パラメーターの資料を参照してください。

注:

**MANUAL** 構成パラメーターの自動チューニングを使用不可にします。パラメーターはその現行の内部値に設定され、自動的な更新は行われなくなります。

#### **IMMEDIATE**

インスタンスが稼働している場合に、即時に、動的に変更を行います。

## UPDATE DATABASE MANAGER CONFIGURATION

**IMMEDIATE** はデフォルトですが、このアクションを実行するためにはインスタンスへのアタッチが有効になっていなければなりません。

### DEFERRED

構成ファイルでのみ変更を行います。したがって、加えられた変更は、インスタンスの再始動時に有効になります。

これは、CLPPlus インターフェースで操作する場合のデフォルトの節です。

**DEFERRED** は、CLPPlus プロセッサの使用時に呼び出す必要はありません。

### 使用上の注意

データベース・マネージャー構成パラメーターのリストの表示または印刷を行うには、**GET DATABASE MANAGER CONFIGURATION** コマンドを使用してください。データベース・マネージャー構成パラメーターを推奨されているデータベース・マネージャーのデフォルトにリセットするには、**RESET DATABASE MANAGER CONFIGURATION** コマンドを使用してください。データベース・マネージャーの構成パラメーターと、構成されている各種データベース・ノード (サーバー、クライアント、またはリモート・クライアントを持つサーバー) に適したこれらのパラメーターの値については、個々の構成パラメーターの説明を参照してください。

すべてのパラメーターを更新できるわけではありません。

データベース・マネージャー構成ファイルへの変更の一部は、ファイルがメモリーにロードされた後にのみ有効になります。オンラインで構成できるパラメーターと構成できないパラメーターについては、構成パラメーターの一覧をご覧ください。即時にリセットされないサーバー構成パラメーターは、**db2start** の実行中にリセットされます。クライアント構成パラメーターの場合、パラメーターは次にアプリケーションを開始するときにリセットされます。クライアントがコマンド行プロセッサである場合は、**TERMINATE** を呼び出すことが必要です。

例えば、データベース・マネージャーの **eastern** インスタンスの **DIAGLEVEL** データベース・マネージャー構成パラメーターをオンラインで変更するには、次のコマンドを入力します。

```
db2 attach to eastern
db2 update dbm cfg using DIAGLEVEL 1
db2 detach
```

エラーが生じた場合には、データベース・マネージャー構成ファイルは変更されません。

データベース・マネージャー構成ファイルは、そのチェックサムが無効であると、更新することができません。このような状況は、データベース・マネージャー構成ファイルが変更されて、適切なコマンドが使用されていない場合に起こります。チェックサムが無効な場合は、データベース・マネージャーを再インストールして、データベース・マネージャー構成ファイルをリセットする必要があります。

現行のインスタンスの **SVCENAME**、または **TPNAME** データベース・マネージャー構成パラメーターを更新するとき、LDAP サポートを使用することができて、このインスタンスに LDAP サーバーが登録されている場合は、LDAP サーバーが新しい値に更新されます。

## UPDATE HEALTH NOTIFICATION CONTACT LIST

インスタンスによって発行されるヘルス・アラートについての通知の連絡先リストを更新します。

### 許可

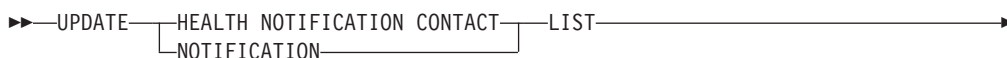
以下の権限のいずれか。

- SYSADM
- SYSCtrl
- SYSMaint

### 必要な接続

インスタンス。明示的なアタッチは必要ありません。

### コマンド構文



### コマンド・パラメーター

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

インスタンスのヘルスの通知を受ける新しい連絡先グループを追加します。

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

インスタンスのヘルスの通知を受ける新しい連絡先を追加します。

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

インスタンスのヘルスの通知を受ける連絡先のリストから、連絡先グループを除去します。

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

インスタンスのヘルスの通知を受ける連絡先のリストから、連絡先を除去します。

## UPDATE HISTORY

データベース履歴レコード項目にあるロケーション、装置タイプ、コメント、または状況を更新します。

### 許可

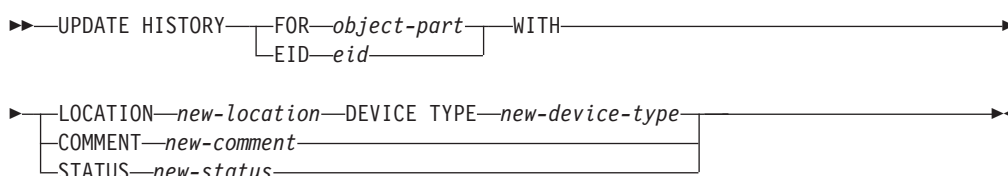
以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMAINT
- DBADM

### 必要な接続

データベース

### コマンド構文



### コマンド・パラメーター

#### FOR *object-part*

更新される履歴項目の ID を指定します。この ID は、タイム・スタンプと 001 から 999 までのオプションのシーケンス番号で構成されます。項目の状況を更新するためにこのパラメーターを使用することはできません。項目の状況を更新するには、代わりに EID を指定してください。

#### EID *eid*

履歴項目 ID を指定します。

#### LOCATION *new-location*

バックアップ・イメージの新しい物理ロケーションを指定します。このパラメーターの解釈は装置タイプに依存します。

#### DEVICE TYPE *new-device-type*

バックアップ・イメージを保管する新しい装置タイプを指定します。有効な装置タイプは次のとおりです。

- |          |                        |
|----------|------------------------|
| <b>D</b> | ディスク                   |
| <b>K</b> | ディスケット                 |
| <b>T</b> | テープ                    |
| <b>A</b> | Tivoli Storage Manager |
| <b>F</b> | スナップショット・バックアップ        |
| <b>U</b> | ユーザー出口                 |



## UPDATE HISTORY

P	パイプ
N	Null 装置
X	XBSA
Q	SQL ステートメント
O	その他

### COMMENT *new-comment*

項目を記述する新しい注釈を指定します。

### STATUS *new-status*

項目の新しい状況を指定します。バックアップ項目のみがその状況を更新できます。有効な値は以下のとおりです。

- A** アクティブ。バックアップ・イメージはアクティブ・ログ・チェーン上にあります。ほとんどの項目はアクティブです。
- I** 非アクティブ。現行のログ・シーケンス (現行のログ・チェーンとも言う) に対応しなくなったバックアップ・イメージには、非アクティブのフラグが立てられます。
- E** 期限切れ。アクティブ・イメージの数が `NUM_DB_BACKUPS` を超えたために不要になったバックアップ・イメージは、期限切れのフラグが立てられます。
- D** 削除済み。リカバリーに使用可能でなくなったバックアップ・イメージは、削除済みとしてマークされることとなります。
- X** 削除しません。DB2HISTORY\_STATUS\_DO\_NOT\_DELETE のマークが付いたりリカバリー・データベース履歴レコード項目は、**PRUNE HISTORY** コマンドの呼び出し、**PRUNE HISTORY** を指定した `ADMIN_CMD` プロシージャの実行、`db2Prune API` の呼び出し、またはリカバリー・データベース履歴レコードファイルの自動プルーニングにより整理されません。  
DB2HISTORY\_STATUS\_DO\_NOT\_DELETE 状況を使用すると、キー・リカバリー・ファイルの項目が整理されたり、それらに関連付けられたリカバリー・オブジェクトが削除されたりしないように保護できます。ログ・ファイル、バックアップ・イメージ、およびロード・コピー・イメージのみに  
DB2HISTORY\_STATUS\_DO\_NOT\_DELETE のマークを付けることができます。

## 例

1997 年 4 月 13 日午前 10 時 00 分にとった全データベース・バックアップのデータベース履歴レコード項目を更新するには、次のように入力します。

```
db2 update history for 19970413100000001 with
location /backup/dbbackup.1 device type D
```

## 使用上の注意

データベース履歴レコードの主な用途は情報を記録することですが、履歴に含まれるデータは、自動リストア操作で直接使用されます。 **AUTOMATIC** オプションを指

定したリストアにおいては、リストア・ユーティリティーによりバックアップ・イメージとそのロケーションの履歴が参照および使用されることにより、自動リストア要求が処理されます。自動リストア機能を使用する場合に、バックアップ・イメージが作成されてから再配置されているなら、現在のロケーションを反映するよう、それらのイメージのデータベース履歴レコードを更新することをお勧めします。データベース履歴の中のバックアップ・イメージのロケーションが更新されない場合、自動リストア処理においてはバックアップ・イメージを見つけることができなくなりますが、手動リストア・コマンドは正常に使用できます。

## UPDATE LDAP NODE

Lightweight Directory Access Protocol (LDAP) の DB2 サーバーを表すノード項目に関連したプロトコル情報を更新します。

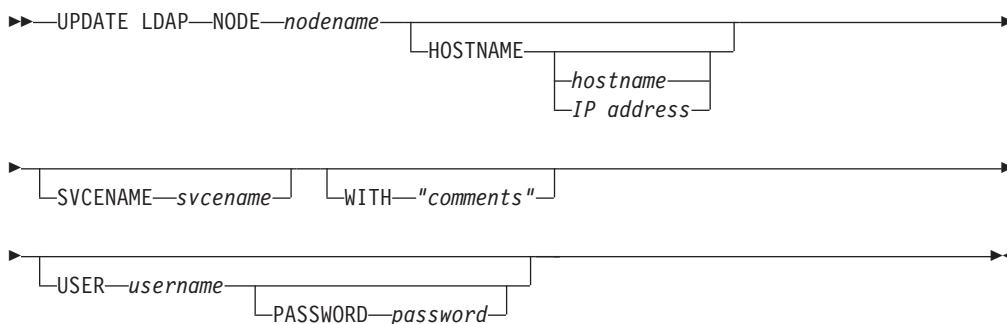
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

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

リモート DB2 サーバーを更新するときのノード名を指定します。ノード名は、DB2 サーバーを LDAP に登録するときに指定した値です。

#### **HOSTNAME** *hostname* | *IP address*

TCP/IP ホスト名または IP アドレスを指定します。

- TCPIP ノードである場合、ホスト名は IPv4 または IPv6 アドレスに解決されます。
- TCPIP4 ノードである場合、ホスト名は IPv4 アドレスにのみ解決されます。
- TCPIP6 ノードである場合、ホスト名は IPv6 アドレスにのみ解決されます。

#### **SVCENAME** *svcename*

TCP/IP サービス名またはポート番号を指定します。

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

DB2 サーバーについて記述します。ネットワーク・ディレクトリーに登録されているサーバーについての記述を補足する、任意のコメントを入力できます。最大長は 30 文字です。復帰文字や改行文字は許可されません。コメント・テキストは必ず二重引用符で囲んでください。

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

ユーザーの LDAP 識別名 (DN) を指定します。LDAP ユーザー DN には、LDAP ディレクトリーでオブジェクトを作成して更新するための十分

な権限が必要です。ユーザーの LDAP DN が指定されない場合、現行ログイン・ユーザーの資格情報が使用されます。

**PASSWORD** *password*

アカウント・パスワードを示します。

---

## UPDATE MONITOR SWITCHES

1 つ以上のデータベース・モニター記録スイッチをオンまたはオフにします。

データベース・マネージャーが開始するとき、6 個のスイッチの設定値が `dft_mon` データベース・マネージャー構成パラメーターによって判別されます。

データベース・モニターはいつでも基本情報セットを記録します。この基本情報以上の情報を必要とするユーザーは、該当するスイッチをオンにできますが、代わりにシステム性能は低下します。 `GET SNAPSHOT` コマンドから出力として利用できる情報の量は、存在するどのスイッチがオンになっているかを反映しています。

### 有効範囲

このコマンドは、現在アタッチされているメンバーに対して呼び出され、デフォルトではそのメンバーについての情報のみを返します。現在アタッチされているメンバーは、デフォルトでは、コマンドが実行されるホストになります。ホストに対して複数のメンバーがある場合、現在アタッチされているメンバーは、そのホスト上の `db2nodes.cfg` ファイルにリストされている最初のメンバーです。

現在アタッチされているメンバーではないメンバーに対してコマンドを実行する場合、特定のメンバーに対してこのコマンドを実行するには `AT MEMBER` オプションを指定し、すべてのメンバーに対してこのコマンドを実行して集約された結果を受け取るには `GLOBAL` オプションを指定してください。

後続のコマンドを実行する対象の、現在アタッチされているメンバーを変更するには、`SET CLIENT` コマンドの `ATTACH_MEMBER` パラメーターを使用します。このクライアント設定を有効にするには、アプリケーションを `DETACH` してから再び `ATTACH` する必要があります。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMANT
- SYSMON

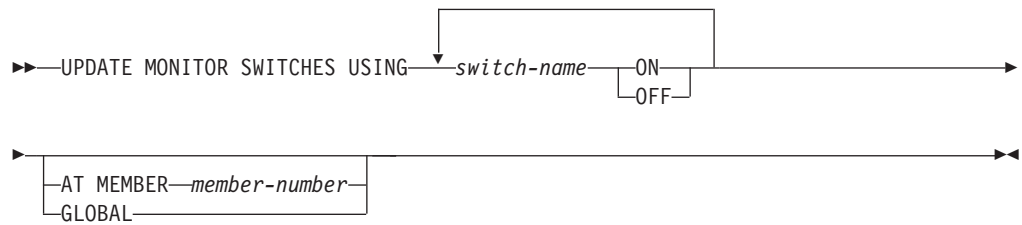
### 必要な接続

インスタンスまたはデータベース

- インスタンスへのアタッチや、データベースへの接続がない場合、デフォルトのインスタンス接続が作成されます。
- インスタンスへのアタッチとデータベース接続の両方がある場合、インスタンス接続が使用されます。

リモート・インスタンス 1 (または異なるローカル・インスタンス) のモニター・スイッチを更新するには、最初にそのインスタンスにアタッチする必要があります。

## コマンド構文



## コマンド・パラメーター

### USING *switch-name*

次のスイッチ名を使用できます。

#### **BUFFERPOOL**

バッファ・プール・アクティビティ情報

#### **LOCK** ロック情報

#### **SORT** ソート情報

#### **STATEMENT**

SQL ステートメント情報

#### **TABLE**

表活動情報

#### **TIMESTAMP**

タイム・スタンプ情報のモニター

#### **UOW** 作業単位情報

### AT MEMBER *member-number*

モニター・スイッチの更新の対象となるメンバーを指定します。

### GLOBAL

すべてのメンバーでモニター・スイッチを更新します。

## 使用上の注意

データベース・マネージャーが収集するのは、スイッチがオンになってから後の情報だけです。db2stop が出されるか、UPDATE MONITOR SWITCHES コマンドを出したアプリケーションが終了するまで、スイッチは設定されたままです。特定のスイッチに関連した情報をクリアするには、まずスイッチをオフに設定し、それからオンにしてください。

あるアプリケーションでスイッチを更新しても、他のアプリケーションには影響がありません。

スイッチ設定値を表示するには、GET MONITOR SWITCHES コマンドを使用してください。

## UPDATE MONITOR SWITCHES

### 互換性

以前のバージョンとの互換性:

- **DB2\_ENFORCE\_MEMBER\_SYNTAX** レジストリー変数が ON に設定されている場合を除き、**DBPARTITIONNUM** または **NODE** を **MEMBER** の代わりに使用できます。



## UPDATE XMLSCHEMA

任意の XML スキーマを、XML スキーマ・リポジトリ (XSR) にある別のスキーマに更新します。

### 許可

以下の権限のいずれか。

- DBADM
- カタログ・ビュー SYSCAT.XSROBJECTS および SYSCAT.XSROBJECTCOMPONENTS に対する SELECT 特権、および次の特権セットの 1 つ:
  - 更新される XML スキーマに対する ALTERIN 特権、および新しい XML スキーマに対する DROPIN 特権 (**DROP NEW SCHEMA** オプションが指定されている場合)。
  - `xmlschema1` によって指定される XML スキーマの OWNER。

### 必要な接続

データベース

### コマンド構文

```

▶▶ UPDATE XMLSCHEMA xmlschema1 WITH xmlschema2
└── DROP NEW SCHEMA ─┘

```

### コマンド・パラメーター

**UPDATE XMLSCHEMA** *xmlschema1*

更新されるオリジナルの XML スキーマの SQL ID を指定します。

**WITH** *xmlschema2*

オリジナルの XML スキーマを更新するのに使用される新 XML スキーマの SQL ID を指定します。

**DROP NEW SCHEMA**

新しい XML スキーマがオリジナルの XML スキーマを更新するために使用された後、ドロップされるよう指定します。

### 例

```

UPDATE XMLSCHEMA JOHNDOE.OLDPROD
WITH JOHNDOE.NEWPROD
DROP NEW SCHEMA

```

XML スキーマ JOHNDOE.OLDPROD の内容が JOHNDOE.NEWPROD の内容に更新され、XML スキーマ JOHNDOE.NEWPROD はドロップされます。

### 使用上の注意

- オリジナルの XML スキーマと新しい XML スキーマには互換性がなければなりません。この互換性の要件について詳しくは、『XML スキーマの展開のための互換性要件』を参照してください。

## UPDATE XMLSCHEMA

- XML スキーマを更新するためには、その前にオリジナルのスキーマと新しいスキーマの両方が XML スキーマ・リポジトリ (XSR) に登録されている必要があります。

## UPGRADE DATABASE

旧バージョンの DB2 データベースを、このインスタンスで実行されているリリースに対応する形式に変換します。

**db2ckupgrade** コマンドは、データベースがアップグレードの準備ができていることを検査するために、インスタンスのアップグレードの前に発行する必要があります。**db2iupgrade** コマンドは、暗黙的に **db2ckupgrade** を呼び出します。アップグレード作業、および DB2 データベース製品の現行バージョンを Windows オペレーティング・システムにインストールする作業に先立って、すべてのデータベースをバックアップしておいてください。

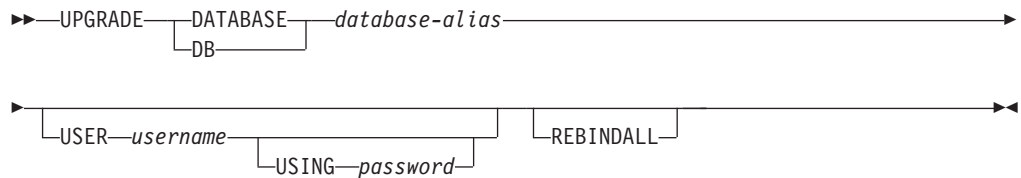
### 許可

SYSADM

### 必要な接続

このコマンドは、データベース接続を確立します。

### コマンド構文



### コマンド・パラメーター

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

現在インストールされているデータベース・マネージャーのバージョンにアップグレードするデータベースの別名を指定します。

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

データベースをアップグレードする際のユーザー名を識別します。

#### **USING** *password*

ユーザー名を認証するために使用するパスワード。ユーザー名を指定してパスワードを省略すると、ユーザーに入力を求めるプロンプトが出ます。

#### **REBINDALL**

アップグレード時に全パッケージの **REBIND** を実行するように指定します。アップグレード時に **REBIND** を自動実行することによって、この手順が漏れないようにします。また、これは、**REBIND** の完了前に他のアプリケーションが開始されないようにするためにも役立ちます。

### 例

次の例は、データベースの別名 **sales** を使ってカタログされたデータベースをアップグレードします。

```
db2 UPGRADE DATABASE sales
```

### 使用上の注意

このコマンドは、データベースを新しいバージョンにアップグレードするだけであり、アップグレード済みのデータベースを元のバージョンに戻すために使用することはできません。

アップグレードの前にデータベースをカタログする必要があります。

アップグレードの途中でエラーが発生する場合、提案されているユーザー応答を試みる前に、**TERMINATE** コマンドを発行することが必要になる場合があります。例えば、アップグレード中にログ・フル・エラーが発生した場合 (SQL1704: データベースのアップグレードは失敗しました。理由コード "3")、データベース構成パラメーターの **logprimary** と **logfilsiz** の値を大きくする前に、**TERMINATE** コマンドを発行する必要があります。データベースが既に再配置された後にアップグレードが失敗した場合、**CLP** はそのデータベース・ディレクトリー・キャッシュをリフレッシュする必要があります (「ログがいっぱい」エラーが戻される場合にこのようになる可能性があります)。

---

## 第 6 章 CLPPlus コマンド

CLPPlus フィーチャーには、広範なユーザー制御、カスタマイズ、および個人用設定を行える数多くのコマンドが含まれています。

注: 特に指定されない限り、CLPPlus コマンド名およびパラメーターは大/小文字を区別しないため、ユーザーは大文字または小文字で指定できます。

・ CLPPlus コマンドは、「何もしない」(NOOP または NOP) マシン言語コマンドに似ています。他の CLPPlus コマンドと一緒に実行されない場合には、このコマンドは無視されます。

・ CLPPlus コマンドを使用すると、現在実行されているコマンドをスキップして、次の SQL> プロンプトに移動することも可能です。このためには、現在のコード・ブロックの新規行にこのコマンドを入力します。このようにすると、間違った項目を入力した際にコマンドをキャンセルするのに役立ちます。キャンセルしたコマンドは履歴内で使用でき、アクセスおよび編集が可能です。

## 呼び出し

このコマンドは、CLPPlus インターフェイスから、または CLPPlus スクリプト・ファイルの中から実行する必要があります。

## 許可

なし

## 必要な接続

なし

## コマンド構文

▶▶.—————▶▶

## 例

この例では、. コマンドを使用して現在のコマンドをキャンセルします。

```
SQL> begin
       2 dbms_output.putline('wrong put_line');
       3 .
SQL>
```

!

V9.7 フィックスパック 3 以降、! CLPPlus コマンドは **HOST CLPPlus** コマンドのシノニムです。オペレーティング・システム・コマンドを実行します。

## 呼び出し

このコマンドは CLPPlus インターフェースで実行する必要があります。

## 許可

なし

## 必要な接続

なし

## コマンド構文

▶—!—*os\_command*————▶

## コマンド・パラメーター

*os\_command*

オペレーティング・システム・コマンドを指定します。

## 例

以下の例では、! コマンドは **dir** オペレーティング・システム・コマンドと一緒に実行されます。

```
SQL> ! dir
```

```
Volume in drive C has no label.
Volume Serial Number is 6806-ABBD
```

```
Directory of C:\USER_Backup\Office\Project\FP3\FP3\src
```

```
06/05/2010 22:18 <DIR>  .
06/05/2010 22:18 <DIR>  ..
06/05/2010 22:35      405.classpath
06/05/2010 17:20 <DIR>  com
                   1 File(s)          798 bytes
                   3 Dir(s)  33,397,190,656 bytes free
```







**@@** CLPPlus コマンドは、**START** CLPPlus コマンドの別名です。これは、CLPPlus スクリプト・ファイルの中で、別の CLPPlus スクリプト・ファイルを呼び出して実行する目的でのみ使用できます。

## 呼び出し

このコマンドは、CLPPlus スクリプトの中から実行する必要があります。

## 許可

なし

## 必要な接続

なし

## コマンド構文



## コマンド・パラメーター

### *path*

実行する SQL ステートメントおよびコマンドが含まれるスクリプト・ファイルへの、絶対パスまたは相対パスのいずれかを指定します。 *path* が指定されない場合、現行ディレクトリーが使用されます。

### *URL*

V9.7 フィックスパック 6 の時点では、実行する SQL ステートメントおよびコマンドが含まれるスクリプト・ファイルへの URL を指定します。 URL は `http://` または `https://` で始まらなければなりません。

### *script-file*

実行する SQL ステートメントおよびコマンドが含まれるスクリプト・ファイル名を指定します。

## 例

`dept_details.sql` というスクリプトから、`employee_count.sql` という別のスクリプトを呼び出します。 `dept_details.sql` の内容は次のとおりです。

```

ACCEPT dept_id PROMPT "Enter Department ID code : "
@@ employee_count &dept_id
  
```

## ACCEPT

**ACCEPT** CLPPlus コマンドは、指定された名前の変数を作成します。その変数には、CLPPlus インターフェースの中で、または実行されるスクリプトの一部として読み取られるパラメーターを介して、対話的に値を代入できます。**ACCEPT** コマンドは、SQL ステートメントまたは SQL バッファーでよく利用される値を保管するのに役立ちます。

デフォルトで、コマンドの出力は CLPPlus インターフェースの標準出力に表示されます。

### 呼び出し

このコマンドは、CLPPlus インターフェースから、または CLPPlus スクリプト・ファイルの中から実行する必要があります。

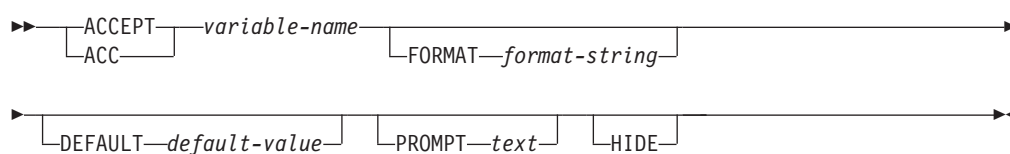
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### *variable-name*

変数名を定義します。スラッシュ (/) またはアットマーク (@) などの特殊記号および特殊文字は使用できません。

**ACCEPT** コマンドを発行すると、*variable-name* の値を入力するようプロンプトが表示されます。

#### **FORMAT** *format-string*

変数に割り当てられる書式を定義します。変数に割り当てようとする値は、指定された書式に従わなければなりません。

文字変数の場合、*format-string* の値は  $An$  です。ここで、 $n$  は、変数の表示に使用できる文字の数を指定します。指定された幅より大きい場合、データは折り返されます。

数値変数の場合、*format-string* の値は以下の 1 つ以上の文字になります。

- \$ 先行ドル記号を表示します。
- , 指定された位置にコンマを表示します。
- . 指定された位置に小数点を表示します。

## ACCEPT

- 0 指定された位置にゼロを表示します。
- 9 指定された位置に有効数字を表示します。

フォーマット設定のオーバーフローが原因で有効数字が失われる場合、# 文字が表示されます。

### DEFAULT *default-value*

ユーザーが入力を要求されたときに値を入力せずに ENTER キーを押したときに、このオプションで定義されたデフォルト値が変数に割り当てられます。

### PROMPT *text*

**ACCEPT** コマンドが入力されたときに、このオプションで定義された値がプロンプトに表示されます。

### HIDE

**HIDE** を指定すると、ユーザーが入力した値はコンソールにエコー出力されません。

## 例

以下の例では、**ACCEPT** コマンドは `my_name` という変数を作成し、値を入力するようプロンプトを表示します。値 `John Smith` はこの変数に保管されます。**DEFINE** コマンドはこの変数の値を表示します。

```
SQL> ACCEPT my_name
Enter value for my_name: John Smith
SQL> DEFINE my_name
DEFINE my_name = "John Smith"
```

次の例は、**ACCEPT** コマンドで使用されるすべてのオプションを示します。

```
SQL> ACCEPT lname FORMAT A10 DEFAULT 'Joy' PROMPT 'Enter Last Name [Joy]:' HIDE
```

**FORMAT** オプションは、`lname` の値を長さ 10 文字の英数字と指定します。ユーザーが入力を要求されたときに値を入力せず、ENTER キーを押したときに使用される **DEFAULT** は `JOY` です。**ACCEPT** コマンドが発行されたときのコマンド行のプロンプトは定義どおりに `Enter Last Name [JOY]:` になります。この場合、デフォルトはプロンプトの一部となっています。**HIDE** オプションは、ユーザーがコンソールで `lname` の値として入力した値をエコー出力しません。

以下の例は、CLPPlus スクリプト・ファイルの中で **ACCEPT** コマンドを使用する方法、また、定義されている変数に値を代入する複数の異なる方法を示すものです。これは `average_salary.sql` という名前のスクリプトであり、特定の部門に属する従業員 (`employee`) の給与 (`salary`) の平均を求めます。

```
ACCEPT dept_id PROMPT "Enter Department ID code : "
SELECT AVG(salary) FROM employee WHERE workdept = &dept_id;
```

このスクリプトを呼び出すには、2 つの異なる方法、つまり引数を指定する方法と引数を指定しない方法とがあります。

引数を指定して呼び出す場合、変数には、呼び出し時に渡される引数の値が代入されます。

```
SQL> start average_salary 'E21'
```

```
Original statement:SELECT AVG(salary) FROM employee WHERE workdept = &dept_id
```



## APPEND

**APPEND** CLPPLUS コマンドは、SQL バッファの現在行の終わりにテキストを追加します。このコマンドを使用して、SQL バッファでコマンドおよびステートメントをビルドすることができます。

### 呼び出し

これは、行エディター・コマンドであり、SQL バッファでコマンドをビルドするために使用できます。

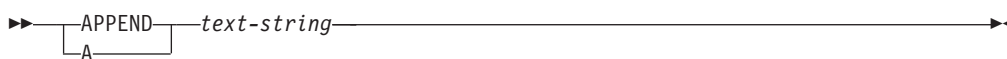
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

*text-string*

付加する文字ストリングを指定します。このストリングにはスペースおよび特殊文字を含めることができます。ストリングの大/小文字は保持されます。

### 例

以下の例では、**APPEND** コマンドはストリング `this text is appended.` を SQL バッファの現在行の終わりに追加します。

```
APPEND this text is appended.
```

以下の例は、**APPEND** コマンドを使用して **SELECT** ステートメントを SQL バッファ内でビルドする方法を示しています。DEPT と WHERE の間を SQL バッファ内で 1 つのスペースで区切るため、**APPEND** コマンドと WHERE 節の間には 2 つのスペースが置かれています。

```
SQL> APPEND SELECT * FROM DEPT
SQL> LIST
  1* SELECT * FROM DEPT
SQL> APPEND WHERE DEPTNO = 10
SQL> LIST
  1* SELECT * FROM DEPT WHERE DEPTNO = 10
```

**LIST** コマンドは、SQL ステートメントがビルドされたときに、SQL バッファの内容を表示します。



## BREAK

**BREAK** CLPPlus コマンドは、結果セット内の指定した個所に、改ページやブランク行を挿入します。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

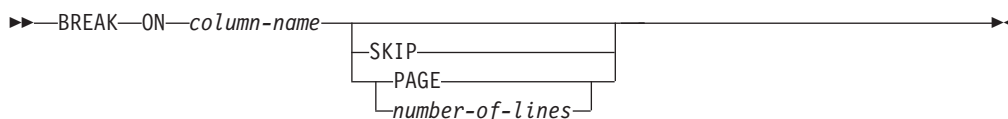
### 許可

なし

### 必要な接続

データベースに接続している必要があります。

### コマンド構文



### コマンド・パラメーター

*column-name*

ブレークの判別に使用する列を指定します。

**SKIP PAGE** | *number-of-lines*

この *number-of-lines* は整数です。

コマンドに **SKIP PAGE** を付加すると、出力が中断し次のページに送られます。

このコマンドに **SKIP number-of-lines** を付加すると、出力が中断し指定された *number-of-lines* に等しい数のブランク行が結果セットに挿入されます。

### 例

以下の例では、**SELECT** ステートメントが呼び出されて、**WORKDEPT** の値が変化した行において、**BREAK** コマンドが呼び出され、指定されたアクションが実行されます。この場合、**SKIP PAGE** が指定されていたため、値が変わった行は現行ページの残りの部分を飛ばして次のページに印刷されます。

```

SQL> BREAK ON WORKDEPT SKIP PAGE;
SQL> SELECT * FROM EMPLOYEE ORDER BY WORKDEPT;
  
```

以下の例では、前の例の動作に加えて、**JOB** 列の値が変わるたびにディスプレイにブランク行が 2 行出力されます。

```

SQL> BREAK ON WORKDEPT SKIP PAGE;
SQL> BREAK ON JOB SKIP 2;
SQL> SELECT * FROM EMPLOYEE ORDER BY WORKDEPT, JOB;
  
```

## BTITLE

**BTITLE** CLPPlus コマンドは、表示される各ページの下部にテキストを挿入します。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

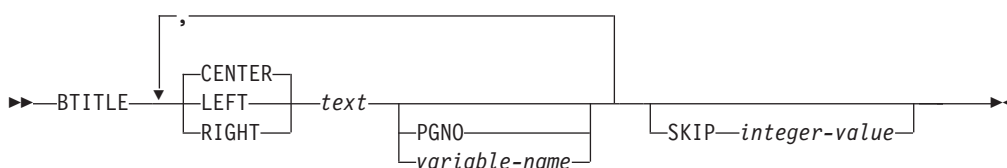
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

*text*

表示するテキストを指定します。

#### CENTER

ディスプレイで、テキストを各ページで中央揃えすることを指定します。

**CENTER**、**LEFT**、**RIGHT** のいずれも指定しない場合、中央揃えがデフォルトの動作になります。

#### LEFT

ディスプレイで、テキストを各ページで左揃えすることを指定します。

#### RIGHT

ディスプレイで、テキストを各ページで右揃えすることを指定します。

#### PGNO

現行のページ番号を指定します。

*variable-name*

*text* フィールドに続くユーザー定義変数を指定します。

#### SKIP *integer-value*

*integer-value* 値は、下部のタイトルの前に表示されるブランク行の行数を指定します。

### 例

次の例では、DEPT: (および変数の内容)、CONFIDENTIAL、Page No: (および現行のページ番号) が各ページの下部に表示されます。下部のタイトルの後には、ブランク行が 3 行続きます。

```
SQL> BREAK ON workdept SKIP PAGE;  
SQL> COLUMN workdept OLD_VALUE old_dept;  
SQL> BTITLE LEFT 'DEPT: ' old_dept, CENTER 'CONFIDENTIAL, RIGHT 'Page No: ' PGNO SKIP 3;
```

次の例では、Page No: タイトル (および現行のページ番号) が各ページの下部に、右揃えで表示されます。下部のタイトルの後には、ブランク行が 2 行続きます。

```
SQL> BTITLE RIGHT 'Page No: ' PGNO SKIP 2;
```

## CHANGE

**CHANGE** CLPPlus コマンドは、指定された内容を SQL バッファで変更します。バッファ内の特定の行にバッファ読み取りプログラムを設定した場合、コマンドはその行の指定された内容のみを変更します。

V9.7 フィックスパック 3 以降、変更を行うバッファ内の行を指定する際の **CHANGE** または **C** トークンは完全なコマンドでオプションになっています。変更を行うバッファ内の行番号と新規テキストを指定するだけでコマンドを発行することができます。

### 呼び出し

これは、SQL バッファでフィールドを変更するために使用される行エディター・コマンドです。

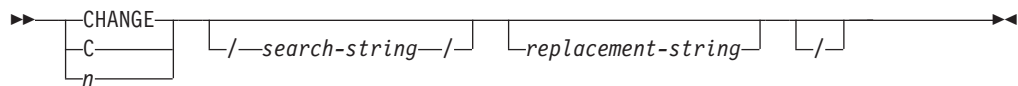
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

*n* 変更されるバッファ内の行番号を指定します。 *n* コマンドは、 *replacement-string* 変数のみを取ります。

#### *search-string*

置換または削除する SQL バッファ内のテキストを定義します。バッファに複数のテキスト行が含まれる場合、**CHANGE** コマンドを実行する前にプロンプトで行番号を入力することによって、変更する行を指定します。

検索対象のテキストにアスタリスク (\*) が含まれる場合、そのアスタリスクを単一引用符で囲んでください。

#### *replacement-string*

置換テキストを指定するか、あるいはテキストを除去することを指定します。 *replacement-string* に値を指定した場合、最初のオカレンスの *search-string* の値は *replacement-string* の値に置換されます。 *replacement-string* に値を指定しなかった場合、最初のオカレンスの *search-string* の値は除去されます。

置換テキストにアスタリスク (\*) が含まれる場合、そのアスタリスクを単一引用符で囲んでください。 V9.7 フィックスパック 3 以降、アスタリスク (\*) を単一引用符で囲む必要がなくなりました。

## 例

以下の例では、**LIST** コマンドはバッファの内容を表示します。SQL プロンプトで、バッファ読み取りプログラムをバッファ内の 3 番目の行の先頭に移動するために 3 を入力します。3 番目の行は、アスタリスクで示される新規の現在行になります。**CHANGE** コマンドは、ストリング 20 のオカレンスをストリング 30 に置換します。**LIST** コマンドは、バッファ内の変更されたテキストを表示します。

```
SQL> LIST
1 SELECT EMPNO, ENAME, JOB, SAL, COMM
2 FROM EMP
3 WHERE DEPTNO = 20
4* ORDER by EMPNO
SQL> 3
3* WHERE deptno = 20
SQL> CHANGE /20/30/
3* WHERE DEPTNO = 30
SQL> LIST
1 SELECT EMPNO, ENAME, JOB, SAL, COMM
2 FROM EMP
3* WHERE DEPTNO = 30
4 ORDER by EMPNO
```

次の例では、バッファに以下の単一ステートメントが含まれています。

```
SQL> SELECT EMPNO FROM EMPLOYEE
```

EMPNO が \* に置換されるようにステートメントを変更するために、1 を入力して、バッファ読み取りプログラムをバッファ内の最初の行の先頭に移動します。以下の **CHANGE** コマンドが発行されます。

```
SQL> CHANGE /empno/'*'/
```

このコマンドの出力は以下のとおりです。

```
1* SELECT * FROM EMPLOYEE
```

コマンド出力では、行番号の後にその行の新しい内容が表示されます。

V9.7 フィックスパック 3 以降、以下が可能になりました。拡張された **CHANGE** コマンドを使用して、変更を行うバッファ内の行番号、およびその変更後の値を指定することができます。

```
SQL> SELECT *
2 FROM
3 EMPLOKEE ;
ERROR near line 1:
SQL0204N "SCHEMA.EMPLOKEE" is an undefined name.
```

```
SQL> LIST
1 SELECT *
2 FROM
3* EMPLOKEE
```

```
SQL> 3 EMPLOYEE
3* EMPLOYEE
```

```
SQL> LIST
1 SELECT *
2 FROM
3* EMPLOYEE
```

```
SQL> /
```

## CLEAR

**CLEAR CLPPlus** コマンドは、SQL バッファの内容を除去したり、**COLUMN** コマンドで設定されたすべての列定義を削除したり、あるいは画面を消去したりします。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

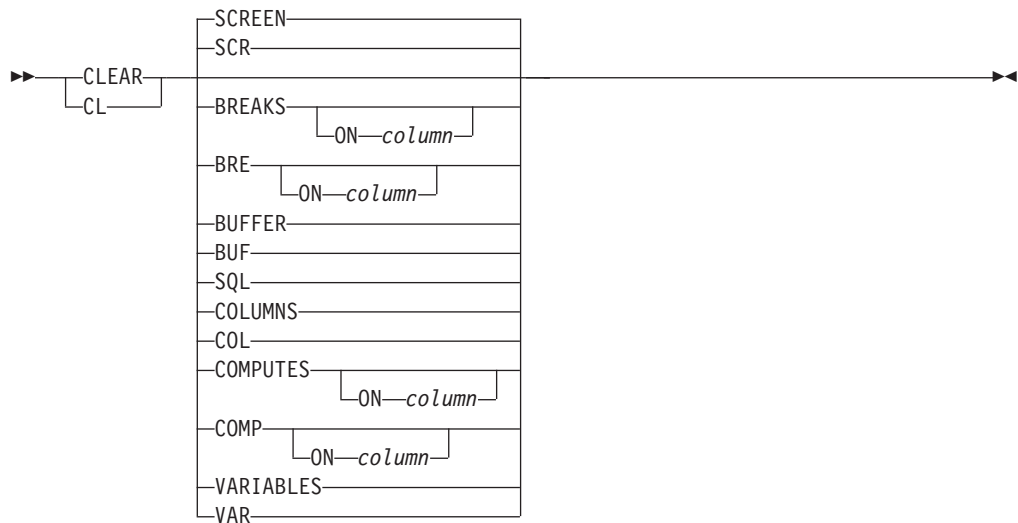
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### SCREEN | SCR

すべての SQL コマンド、現在表示されているデータ、および CLPPlus メッセージを画面から除去します。**CLEAR** コマンドをオプションなしで入力すると、デフォルトの動作により画面が消去されます。

#### BREAKS | BRE ON *column*

列を指定しないと、すべてのブレイクを消去します。列を指定した場合、その列に関連付けられたブレイクが消去され、その他のブレイクはすべて消去されません。

#### BUFFER | BUF and SQL

SQL バッファ内のすべてのテキストを削除します。**BUFFER** パラメーター (または **BUF** パラメーター) と **SQL** パラメーターは同義です。

#### COLUMNS | COL

SQL バッファ内の列定義を除去します。

**COMPUTES | COMP ON *column***

列を指定しないと、すべての計算定義を消去します。列を指定した場合、その列に関連付けられた計算定義が消去され、その他の計算定義はすべて消去されません。

**VARIABLES | VAR**

定義されているすべてのバインド変数を消去します。

## COLUMN

**COLUMN** CLPPlus コマンドは、表内の列の文字および数値の出力フォーマットを指定します。**COLUMN** コマンドで設定されたフォーマットは、現行セッション中でのみ有効です。現行セッション内で、同じ列のフォーマット設定を複数回変更または消去することができます。

データベース内の指定された列に対して **COLUMN** を発行すると、デフォルトで、フォーマット設定は CLPPlus インターフェースの標準出力を使用して表示されます。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

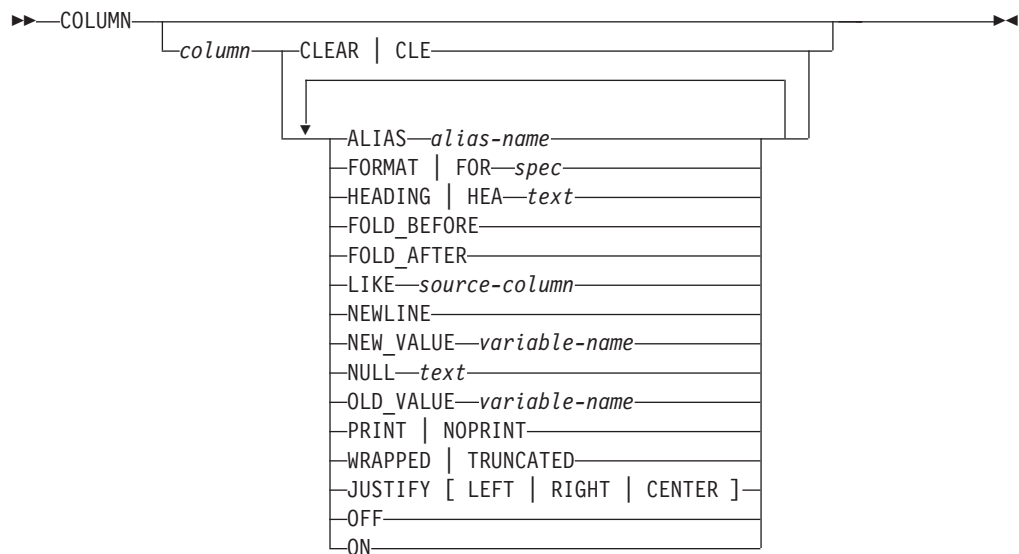
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### *column*

フォーマットが適用される表列の名前を指定します。*column* の後にどのパラメーターも指定しなかった場合、指定した *column* に関するデフォルト以外のフォーマット設定が表示されます。

#### **ALIAS** *alias-name*

*column* に別名を指定します。*alias-name* の最大長は、255 文字です。別名を使用できるその他のコマンドには、**COMMAND**、**COMPUTE**、**COLUMN** などがあります。



**CLEAR | CLE**

指定された列のすべてのフォーマット設定をデフォルト値に変更します。 **CLEAR** を指定した場合、*column* 以外のパラメーターを指定することはできません。

**FORMAT | FOR *spec***

指定された列に適用するフォーマットを指定します。これには、文字列と数値列の 2 つのタイプの列があります。

文字列の場合、*spec* の値は *An* です。ここで、*n* は、列の表示に使用できる文字の数を指定します。指定された幅より大きい場合、データは折り返されます。

数値列の場合、*spec* の値は以下の 1 つ以上の文字になります。

- \$ 先行ドル記号を表示します。
- ,
- .
- 0 指定された位置にゼロを表示します。
- 9 指定された位置に有効数字を表示します。

フォーマット設定のオーバーフローが原因で有効数字が失われる場合、# 文字が表示されます。

**HEADING | HEA *text***

指定された列の見出しを指定します。

**FOLD\_BEFORE**

指定の列の値を出力する前に、各行で改行フィードおよび復帰を行います。

**FOLD\_AFTER**

指定の列の値を出力してから、各行で改行フィードおよび復帰を行います。

**LIKE *source-column***

*source-column* の書式と表示属性は、*column* に適用されます。

**NEWLINE**

**FOLD\_AFTER** の同義語。指定の列の値を出力してから、各行で改行フィードおよび復帰を行います。

**NEW\_VALUE *variable\_name***

**BREAK** コマンドを使用して定義されたブレイク列に、新しい値を保持できる変数を定義します。この変数は、ページ上部タイトルである **TTITLE** コマンドとともに使用します。ブレイク列は、**SKIP PAGE** アクションを使用して定義する必要があります。

V9.7 フィックスパック 3 以降、**NEW\_VALUE** も現行セッション内のすべての場所で使用できるようになりました。置換変数と同様です。列に対して **NEW\_VALUE** 変数を定義すると、CLPPlus は常に、指定された変数名で置換変数を作成します。この変数は、列ブレイクごとに列値で更新されます。

**NULL *text***

指定した列の値が NULL の場合、*text* に指定された値を印刷します。 *text* の最大長は、255 文字です。

**OLD\_VALUE *variable\_name***

**BREAK** コマンドを使用して定義されたブレイク列に、古い値を保持できる変数を

## COLUMN

定義します。この変数は、ページ下部タイトルである **BTITLE** コマンドとともに使用します。ブレイク列は、**SKIP PAGE** アクションを使用して定義する必要があります。

V9.7 フィックスパック 3 以降、**OLD\_VALUE** も現行セッション内のすべての場所で使用できるようになりました。置換変数と同様です。列に対して **OLD\_VALUE** 変数を定義すると、CLPPlus は常に、指定された変数名で置換変数を作成します。この変数は、列ブレイクごとに列値で更新されます。

### PRINT | NOPRINT

指定された列のコンソール印刷を使用可能にするか、使用不可にするかを指定します。

### WRAPPED | TRUNCATED

指定されたフォーマットを超える場合に、CLPPlus 出力で列データを折り返すか、あるいは切り捨てるかを指定します。

### JUSTIFY [LEFT | RIGHT | CENTER]

列の位置調整を LEFT、RIGHT、または CENTER に指定します。

### OFF

フォーマット・オプションをデフォルト値に変更します。セッションで以前に列に指定した値は保存されており、そのセッションで後から使用することができます。

**ON** フォーマット・オプションを、**COLUMN** を最後に実行したときに指定の列に適用された値に変更します。

## 例

以下の例では、**SET PAGESIZE** コマンドは最大ページ長を 9999 に設定し、**COLUMN** コマンドは **JOB** 列の表示幅を 5 文字に変更します。**SELECT** ステートメントは、表内の指定された列を表示します。

```
SQL> SET PAGESIZE 9999
SQL> COLUMN JOB FORMAT A5
SQL> COLUMN JOB
COLUMN      JOB      ON
FORMAT      A5
WRAPPED
SQL> SELECT EMPNO, ENAME, JOB FROM EMP;
```

EMPNO	ENAME	JOB
7369	SMITH	CLERK
7499	ALLEN	SALES MAN
7521	WARD	SALES MAN
7566	JONES	MANAG ER
7654	MARTING	SALES MAN
7698	BLAKE	MANAG ER
7782	CLARK	MANAG ER
7788	SCOTT	ANALY ST
7839	KING	PRESI DENT

```

7844 TURNER SALES
      MAN
7876 ADAMS  CLERK
7900 JAMES  CLERK
7902 FORD   ANALY
      ST
7934 MILLER CLERK
    
```

14 rows received.

以下の例では、**COLUMN** コマンドは数値のフォーマットを SAL 列に適用します。

```

SQL> COLUMN SAL FORMAT $99,999.00
SQL> COLUMN
COLUMN JOB ON
FORMAT A5
WRAPPED

COLUMN SAL ON
FORMAT $99,999.00
WRAPPED
SQL> SELECT EMPNO, ENAME, JOB, SAL FROM EMP;
    
```

EMPNO	ENAME	JOB	SAL
7369	SMITH	CLERK	\$800.00
7499	ALLEN	SALES MAN	\$1,600.00
7521	WARD	SALES MAN	\$1,250.00
7566	JONES	MANAG ER	\$2,975.00
7654	MARTIN	SALES MAN	\$1,250.00
7698	BLAKE	MANAG ER	\$2,850.00
7782	CLARK	MANAG ER	\$2,450.00
7788	SCOTT	ANALY ST	\$3,000.00
7839	KING	PRESI DENT	\$5,000.00
7844	TURNER	SALES MAN	\$1,500.00
7876	ADAMS	CLERK	\$1,100.00
7900	JAMES	CLERK	\$950.00
7902	FORD	ANALY ST	\$3,000.00
7934	MILLER	CLERK	\$1,300.00

14 rows retrieved.

以下の例では、改善された **NEW\_VALUE** パラメーターの動作が示されています。新しい **OLD\_VALUE** 動作は同一です。これは V9.7 フィックスパック 3 以降でのみ有効です。

## COLUMN

```
SQL> break on empno skip 1
SQL> column empno new_value highest_sal
SQL> select empno from employee order by salary;

EMPNO
-----
200340
*****

000290
*****

200330
*****

000310
*****

...
...

000070
*****

000030
*****

000010
*****

SQL>DEFINE
DEFINE HIGHEST_SAL = 000010

SQL> select EMPNO, FIRSTNME, MIDINIT, LASTNAME from employee where empno=&highest_sal;
EMPNO  FIRSTNME      MIDINIT  LASTNAME
-----
000010 CHRISTINE     I        HAAS
```

## COMPUTE

**COMPUTE** CLPPlus コマンドは、定義された列の集合値に対して指定した関数を実行します。このコマンドは、**BREAK** CLPPlus コマンドとともに実行します。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

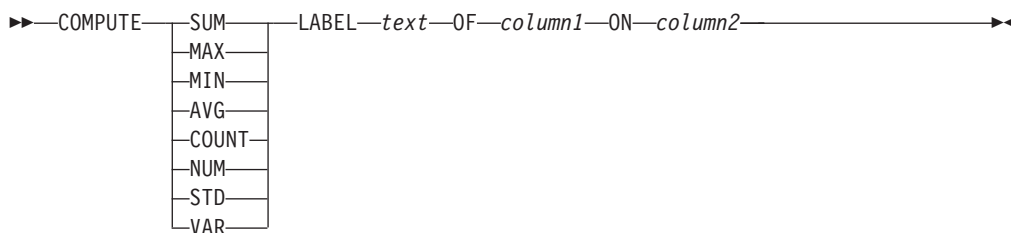
### 許可

なし

### 必要な接続

データベースに接続している必要があります。

### コマンド構文



### コマンド・パラメーター

#### SUM

SUM 関数は、指定された列の集約値を合計します。

#### MIN

MIN 関数は、指定された列の集約値の最小値を返します。

#### MAX

MAX 関数は、指定された列の集約値の最大値を返します。

#### AVG

AVG 関数は、指定された列の集約値の平均を返します。

#### COUNT

COUNT 関数は、指定された列の NULL 以外の値の数をカウントします。

#### NUM

NUM 関数は、指定された列について処理された集約行の数を返します。

#### STD

STD 関数は、指定された列の集約値の標準偏差を返します。

#### VAR

VAR 関数は、指定された列の集約値の分散を返します。

#### LABEL text

指定された関数の出力の前に印刷されるテキスト・ラベルを定義します。

## COMPUTE

*column1*

関数を実行する対象の列を指定します。

*column2*

**BREAK** コマンドを実行する対象の列を指定します。

### 例

次の例は、**COMPUTE** コマンドを **BREAK** コマンドとともに使用方法を示しています。

```
SQL> BREAK ON WORKDEPT SKIP 2;
SQL> COMPUTE AVG LABEL "Average" OF SALARY ON WORKDEPT;
SQL> COMPUTE MAX LABEL "Maximum" OF SALARY ON WORKDEPT;
SQL> SELECT WORKDEPT, EMPNO, SALARY FROM EMPLOYEE ORDER BY WORKDEPT;
```

この例のコマンド出力は、次のとおりです。

WORKDEPT	EMPNO	SALARY
-----	-----	-----
A01	00100	75000.00
A01	00101	80000.00
A01	00102	70000.00
*****		-----
Average		75000.00
Maximum		80000.00
A02	00103	80000.00
A02	00104	90000.00
A02	00105	85000.00
*****		-----
Average		85000.00
Maximum		90000.00

## CONNECT

**CONNECT CLPPlus** コマンドは、データベースに接続されているユーザー ID を変更したり、別のデータベースに接続したり、あるいはその両方を行います。このコマンドは、変更の結果も表示します。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

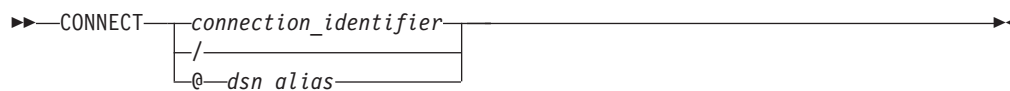
### 許可

なし

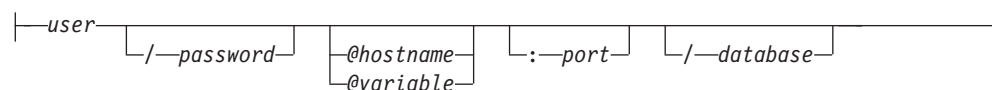
### 必要な接続

なし

### コマンド構文



#### connection\_identifier:



### コマンド・パラメーター

パラメーターの間にはスペースを入れないでください。

#### *user*

データベースへ接続するユーザー ID を指定します。

#### *password*

ユーザー ID に対応するパスワードを指定します。

#### *hostname*

データベースが稼働するコンピューター名を指定します。例えば、ascender という名前のコンピューターの場合は、@ascender と指定します。

#### *variable*

CLPPlus の関連情報を格納する変数の名前を指定します。この変数は接続ストリング・タイプ情報を定義でき、始動時に CLPPlus によって読み取られる login.sql という名前のファイルで定義する必要があります。

#### *port*

データベース・サーバーがインストールされたコンピューター上の接続受信ポート番号を指定します。デフォルトは 50000 です。

#### *database*

接続先のデータベース名を指定します。デフォルトは SAMPLE です。

- / データベースへの接続に、現在のオペレーティング・システムのログイン・ユーザー ID を使用するように指定します。

#### *dsn\_alias*

別名 *dsn\_alias* を含む `db2dsdriver.cfg` ファイルから、データベース接続情報を読み取るように指定します。*dsn\_alias* が `db2dsdriver.cfg` ファイルにない場合は、*dsn\_alias* というストリングがデータベース名として使用され、その他すべての接続パラメーターは対話的に取得されます。V9.7 フィックスパック 5 以降、指定した `db2dsdriver.cfg` ファイルで LDAP ディレクトリー・サーバーを構成すると、次のステップが適用されます。

1. 別名が *dsn\_alias* である DSN の `db2dsdriver.cfg` ファイルから、データベース接続情報が読み取られます。
2. *dsn\_alias* が `db2dsdriver.cfg` ファイルにない場合は、構成済みの LDAP ディレクトリー・サーバーで、*dsn\_alias* という名前の項目が検索されます。
3. *dsn\_alias* が LDAP ディレクトリー・サーバーで見つからない場合は、*dsn\_alias* がデータベース名として使用され、その他すべての接続パラメーターは対話的に取得されます。

注: CLPPlus は、DSN 別名を使用したデータベースへの接続試行、および `db2dsdriver.cfg` 構成ファイルで定義される認証メカニズムをサポートします。`db2dsdriver.cfg` ファイルで認証メカニズムを定義しない場合、デフォルトの認証メカニズムが CLPPlus によって使用されます。デフォルトの認証メカニズムは、JCC の認証メカニズムと同等です。

## 例

以下の例では、データベース接続は、ユーザー名 `smith` およびポート `5445` のローカル・ホスト上のデータベース `DB2` に変更されます。

```
SQL> CONNECT smith/mypassword@localhost:5445/db2
Connected to CLPPlus 1.1.0.10 (localhost:5445/db2) AS smith
```

同じセッションで、接続はユーザー名 `CLPPlus` に変更されます。この接続では、`localhost`、`5444`、および `DB2` は、ホスト、ポート、およびデータベースの値として保持されます。

```
SQL> CONNECT CLPPlus/password
Connected to CLPPlus 1.1.0.10 (localhost:5444/db2) AS CLPPlus
```

以下の例では、まず `db2dsdriver.cfg` ファイルを見つけて、データベースへの接続を試行します。このファイルが見つかり、ホスト、ポート、データベースの各値に関して `default_dsn` が読み取られます。接続の試行時には、現在のログオン ID が使用されます。`db2dsdriver.cfg` ファイルが見つからない場合は、必須パラメーターすべてが対話的に要求されます。

```
SQL> CONNECT /
```

以下の例では、`db2dsdriver.cfg` ファイルの `data_dsn` 別名からパラメーターを取り出して、データベースへの接続を試行します。接続には、`db2admin` ユーザー ID が使用されます。読み取れないパラメーターに関しては、対話的に要求されます。

```
SQL> CONNECT db2admin@data_dsn
```



次の例では、login.sql ファイルにデータベースへの接続試行に使用される変数の定義が含まれます。login.sql ファイルには、define connStr = localhost:50000/sample が含まれ、次のように **CONNECT CLPPlus** コマンドで使用できます。

```
SQL> CONNECT db2admin@connStr
```

db2admin ユーザーによる接続は、リスナー・ポート番号が 50000 のローカル・ホスト上のサンプル・データベースで試行されます。

## COPY

**COPY** CLPPlus コマンドは、ソースのデータベースと表からターゲットのデータベースと表にデータをコピーします。

### 呼び出し

このコマンドは、CLPPlus インターフェースから、または CLPPlus スクリプト・ファイルの中から実行する必要があります。

### 許可

なし

### 必要な接続

なし

### コマンド構文

```

▶▶ COPY FROM srcdb TO destdb
    ┌ APPEND
    ├── CREATE
    ├── INSERT
    └── REPLACE
▶
└─ dest_table [(col1, col2, ...)] ── USING query ──▶▶

```

#### srcdb, destdb:

```

┌ connection_identifier ───┐
├ / ───┤
└ @ dsn_alias ───┘

```

#### connection\_identifier:

```

┌ user ───┐
├ / password ───┤
├ @ host ───┤
├ : port ───┤
└ / database ───┘

```

### コマンド・パラメーター

#### FROM srcdb

データのコピー元となる接続の詳細情報とデータベース名を定義します。

**注:** **COPY** コマンドでは、**FROM** パラメーターか **TO** パラメーターのいずれか、あるいはその両方を指定する必要があります。 **FROM** を指定しないで、**TO** を指定する場合、データベース接続が存在すると、その接続先のデータベースがソース・データベースに使用されます。

#### TO destdb

データのコピー先となる接続の詳細情報とデータベース名を定義します。

注: **COPY** コマンドでは、FROM パラメーターか TO パラメーターのいずれか、あるいはその両方を指定する必要があります。TO を指定しないで、FROM を指定すると、データベース接続が存在する場合にはその接続先のデータベースがターゲット・データベースに使用されます。

#### APPEND

データを *dest\_table* に挿入します。 *dest\_table* が存在しない場合は、*dest\_table* 変数および [(*col1*, *col2*, ...)] 変数を使用して、宛先表定義を指定する必要があります。

#### CREATE

*dest\_table* を作成して、データを挿入します。 *dest\_table* 変数および [(*col1*, *col2*, ...)] 変数を使用して、宛先表定義を指定する必要があります。 *dest\_table* が存在する場合には、エラーが戻ります。

#### INSERT

データを *dest\_table* に挿入します。 *dest\_table* が存在しない場合には、エラーが戻ります。

#### REPLACE

*dest\_table* 変数および [(*col1*, *col2*, ...)] 変数を使用して、宛先表定義を指定する必要があります。 *dest\_table* は、ドロップされて再作成され、データが挿入されます。

#### *dest\_table*

データが挿入されるターゲット・データベース表です。

#### *query*

ソース・データベースからデータを取得するために使用する SQL 照会です。

#### *user*

データベースへ接続するユーザー ID を指定します。

#### *password*

ユーザー ID に対応するパスワードを指定します。

#### *hostname*

データベースが稼働するコンピューター名を指定します。例えば、ascender という名前のコンピューターの場合は、@ascender と指定します。

#### *port*

データベース・サーバーがインストールされたコンピューター上の接続受信ポート番号を指定します。デフォルトは 50000 です。

#### *database*

接続先のデータベース名を指定します。デフォルトは SAMPLE です。

#### *dsn\_alias*

db2dsdriver.cfg ファイルから、別名が *dsn\_alias* である DSN のデータベース接続情報を読み取るように指定します。指定した *dsn\_alias* が db2dsdriver.cfg ファイルにないと、*dsn\_alias* というストリングをデータベース名として使用し、他のすべての接続パラメーターは対話的に取得します。

### 例

以下のコマンドでは、db1 データベースの emp 表の行をコピーして、db2 データベースの emp 表に追加します。

```
COPY FROM u1@db1 TO u2@db2 APPEND emp USING SELECT * FROM emp;
```

以下のコマンドでは、db1 データベースの emp 表の行をコピーして、db2 データベースの emp 表に追加します。ターゲット表が db2 という名前のデータベースに存在しないため、コマンドで表定義を指定する必要があります。

```
COPY FROM u1@db1 TO u2@db2 APPEND emp (EmpId integer, name varchar(20))  
USING SELECT * FROM emp;
```

以下のコマンドでは、db1 データベースの emp 表の行をコピーし、db2 データベースに emp 表を作成し、db2 で新しく定義した表に行を挿入します。

```
COPY FROM u1@db1 TO u2@db2 CREATE emp (EmpId integer, name varchar(20))  
USING SELECT * FROM emp;
```

以下のコマンドでは、db1 データベースの emp 表の行をコピーし、db2 データベースの emp 表に挿入します (このターゲット表は既に存在する)。

```
COPY FROM u1@db1 TO u2@db2 INSERT emp USING SELECT * FROM emp;
```

以下のコマンドでは、db1 データベースの emp 表の行をコピーし、db2 データベースに emp 表を再作成し、行を置き換えます。

```
COPY FROM u1@db1 TO u2@db2 REPLACE emp (EmpId integer, name varchar(20))  
USING SELECT * FROM emp;
```

## DEFINE

**DEFINE** CLPPlus コマンドは、ユーザー変数 (置換変数とも呼ばれる) を作成し、その値を指定します。このコマンドは、1 つ以上のユーザー変数の値も表示します。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### *variable*

変数の名前を指定します。*text* なしで *variable* を指定した場合、その変数の名前とその値が表示されます。*variable* を指定しなかった場合、すべての変数の名前とその値が表示されます。

#### *text*

*variable* で指定された変数に割り当てるテキストを指定します。テキストにスペースが含まれる場合は、二重引用符または単一引用符で囲む必要があります。テキストにスペースが含まれない場合、引用符はオプションです。

### 例

以下の例では、**DEFINE** コマンドは、DB2、DEPT、および NAME 変数を定義してから、すべての変数の値を表示します。

```

SQL> DEFINE DEPT = 20
SQL> DEFINE NAME = 'John Smith'
SQL> DEFINE DB2 = 'localhost:5445/sample'
SQL> DEFINE
DEFINE DB2 = "localhost:5445/sample"
DEFINE DEPT = "20"
DEFINE NAME = "John Smith"

```

## DEL

DEL コマンドは、1 つ以上の行を SQL バッファから削除します。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

このパラメーターを使用して、SQL バッファから削除する行の範囲の開始および終了を指定することができます。どのパラメーターも指定しなかった場合は、現在行が削除されます。

*n* 行番号を指定します。

*n m*

2 つの行番号を指定します。ここで、*m* の値は *n* の値より大きくします。

\* 現在行を示します。

LAST | L

SQL バッファ内の最後の行を示します。

### 例

以下の例では、5 番目の行 (列 SAL が入っている) および 6 番目の行 (列 COMM が入っている) が SQL バッファ内の SELECT ステートメントから削除されます。

```

SQL> LIST
1  SELECT
2    EMPNO
3    ,ENAME
4    ,JOB
5    ,SAL
6    ,COMM
  
```

```
7      ,DEPTNO
8* FROM EMP
SQL> DEL 5 6
SQL> LIST
1 SELECT
2     EMPNO
3     ,ENAME
4     ,JOB
5     ,DEPTNO
6* FROM EMP
```

行 7 だった内容は行 5 の内容になり、行 8 だった内容は行 6 の内容になります。現在行の内容は変更されていませんが、アスタリスクのマークが付けられている現在行は行 8 から行 6 に変更されています。

## DESCRIBE

**DESCRIBE** CLPPlus コマンドは、列とそのデータ・タイプおよび表ビューの長さのリスト、プロシージャまたは関数のパラメーターのリスト、あるいはパッケージのプロシージャと関数およびそれらのパラメーターのリストを表示します。

V9.7 フィックスパック 3 以降、**DESCRIBE** CLPPlus コマンドが機能拡張され、表示するデータベース・オブジェクトのタイプを指定できるようになりました。表示するオブジェクトのタイプを指定しないと、指定の名前とスキーマを持つすべてのオブジェクトが表示されます。デフォルト・スキーマは *CURRENT SCHEMA* です。

V9.7 フィックスパック 4 以降、**DESCRIBE** CLPPlus コマンドの機能が拡張され、テンポラル表をサポートするようになりました。テンポラル表は、DB2 z/OS バージョン 10 の新規機能です。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行することができます。

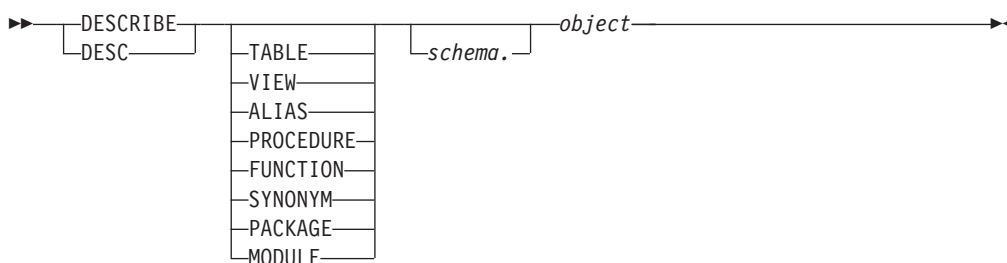
### 許可

なし

### 必要な接続

データベースに接続している必要があります。

### コマンド構文



### コマンド・パラメーター

#### TABLE

表示するデータベース・オブジェクトのタイプを表に指定します。V9.7 フィックスパック 3 以降、使用可能です。

#### VIEW

表示するデータベース・オブジェクトのタイプをビューに指定します。V9.7 フィックスパック 3 以降、使用可能です。

#### ALIAS

表示するデータベース・オブジェクトのタイプを別名に指定します。V9.7 フィックスパック 3 以降、使用可能です。



**PROCEDURE**

表示するデータベース・オブジェクトのタイプをプロシージャに指定します。  
V9.7 フィックスパック 3 以降、使用可能です。

**FUNCTION**

表示するデータベース・オブジェクトのタイプを関数に指定します。V9.7 フィックスパック 3 以降、使用可能です。

**SYNONYM**

表示するデータベース・オブジェクトのタイプをシノニムに指定します。V9.7 フィックスパック 3 以降、使用可能です。

**PACKAGE**

表示するデータベース・オブジェクトのタイプをパッケージに指定します。V9.7 フィックスパック 3 以降、使用可能です。

**MODULE**

表示するデータベース・オブジェクトのタイプをモジュールに指定します。V9.7 フィックスパック 3 以降、使用可能です。

*schema*

表示するオブジェクトが属するスキーマの名前を指定します。デフォルト・スキーマは *CURRENT SCHEMA* です。

*object*

表示する表、ビュー、プロシージャ、関数、またはパッケージの名前を指定します。

**例**

以下の例では、**DESCRIBE** コマンドを実行して、**ABCD** という名前の表の詳細情報を取得します。

```
SQL> DESCRIBE TABLE ABCD ;
```

```
TABLE - ABCD
```

Name	Data Type	Type schema	Length	Scale	Nulls	Hidden
ROLL	INTEGER	SYSIBM	4	0	Y	N
NAME	CHARACTER	SYSIBM	10	0	Y	P

以下の例では、**DESCRIBE** コマンドを実行して、バイトエンポラル表の詳細を取得します。

```
SQL > create table policy
(
  policy_id int NOT NULL,
  coverage int NOT NULL IMPLICITLY HIDDEN,
  bus_start date NOT NULL,
  bus_end date NOT NULL,
  system_start TIMESTAMP(12) generated always as row begin NOT NULL,
  system_end TIMESTAMP(12) generated always as row end NOT NULL,
  trans_start generated always as transaction start ID,
  period BUSINESS_TIME(bus_start, bus_end),
  period SYSTEM_TIME (system_start, system_end)
);
```

```
DB250000I: The command completed successfully.
```

## DESCRIBE

```
SQL> create table policy_hist LIKE policy;
DB250000I: The command completed successfully.

SQL> ALTER TABLE policy ADD VERSIONING USE HISTORY TABLE policy_hist;
DB250000I: The command completed successfully.

SQL> describe policy
TABLE - POLICY
*****
Name          Data Type      Type schema    Length  Scale  Nulls Hidden
-----
POLICY_ID     INTEGER        SYSIBM         4       0 N    Not
COVERAGE     INTEGER        SYSIBM         4       0 N    Implicit
BUS_START    TIMESTAMP      SYSIBM         7       0 N    Not
BUS_END      TIMESTAMP      SYSIBM         7       0 N    Not
SYSTEM_START  TIMESTAMP      SYSIBM        13      12 N    Not
SYSTEM_END    TIMESTAMP      SYSIBM        13      12 N    Not
TRANS_START   TIMESTAMP      SYSIBM        13      12 Y    Not
```

Temporal Type : Bitemporal

Table is versioned and has the following periods

```
-----
Name          Type Begin Column  End Column
-----
SYSTEM_TIME   S   SYSTEM_START    SYSTEM_END
BUSINESS_TIME A   BUS_START        BUS_END
```

以下の例では、**DESCRIBE** コマンドを実行して、システム期間があるもののバージョン管理されていない表の詳細を取得します。

```
SQL> create table demo_nontemp
(
  policy_id int NOT NULL,
  coverage int NOT NULL IMPLICITLY HIDDEN,
  system_start TIMESTAMP(12) generated always as row begin NOT NULL,
  system_end TIMESTAMP(12) generated always as row end NOT NULL,
  trans_start generated always as transaction start ID,
  period SYSTEM_TIME (system_start, system_end)
);
```

DB250000I: The command completed successfully.

```
SQL> describe demo_nontemp
```

```
TABLE - TEMPTAB
*****
Name          Data Type      Type schema    Length  Scale  Nulls Hidden
-----
POLICY_ID     INTEGER        SYSIBM         4       0 N    Not
COVERAGE     INTEGER        SYSIBM         4       0 N    Implicit
SYSTEM_START  TIMESTAMP      SYSIBM        13      12 N    Not
SYSTEM_END    TIMESTAMP      SYSIBM        13      12 N    Not
TRANS_START   TIMESTAMP      SYSIBM        13      12 Y    Not
```

Table has the following periods

Name	Type	Begin Column	End Column
SYSTEM_TIME	S	SYSTEM_START	SYSTEM_END

\*\*\*\*\*

以下の例では、**DESCRIBE** コマンドを実行して、アプリケーション期間テンポラル表の詳細を取得します。

```
SQL> create table demo_app
(
  policy_id int NOT NULL,
  coverage int NOT NULL IMPLICITLY HIDDEN,
  bus_start date NOT NULL,
  bus_end date NOT NULL,
  period BUSINESS_TIME(bus_start, bus_end));
```

DB250000I: The command completed successfully.

```
SQL> describe demo_app
```

TABLE - DEMO\_APP

\*\*\*\*\*

Name	Data Type	Type schema	Length	Scale	Nulls	Hidden
POLICY_ID	INTEGER	SYSIBM	4	0	N	Not
COVERAGE	INTEGER	SYSIBM	4	0	N	Implicit
BUS_START	TIMESTAMP	SYSIBM	7	0	N	Not
BUS_END	TIMESTAMP	SYSIBM	7	0	N	Not

Temporal Type : Application period temporal

Table has the following periods

Name	Type	Begin Column	End Column
BUSINESS_TIME	A	BUS_START	BUS_END

\*\*\*\*\*

以下の例では、**DESCRIBE** コマンドを実行して、システム期間テンポラル表の詳細を取得します。

```
SQL> create table demo_sys
(
  policy_id int NOT NULL,
  coverage int NOT NULL IMPLICITLY HIDDEN,
  system_start TIMESTAMP(12) generated always as row begin NOT NULL,
  system_end TIMESTAMP(12) generated always as row end NOT NULL,
  trans_start generated always as transaction start ID,
  period SYSTEM_TIME (system_start, system_end)
);
```

DB250000I: The command completed successfully.

```
SQL> create table demo_sys_history like demo_sys ;
```

DB250000I: The command completed successfully.

```
SQL> ALTER TABLE DEMO_SYS ADD VERSIONING USE HISTORY TABLE DEMO_SYS_HISTORY;
```

DB250000I: The command completed successfully.

## DESCRIBE

```
SQL> desc demo_sys
```

```
TABLE - DEMO_SYS
```

```
*****
```

Name	Data Type	Type schema	Length	Scale	Nulls	Hidden
POLICY_ID	INTEGER	SYSIBM	4	0	N	Not
COVERAGE	INTEGER	SYSIBM	4	0	N	Implicit
SYSTEM_START	TIMESTAMP	SYSIBM	13	12	N	Not
SYSTEM_END	TIMESTAMP	SYSIBM	13	12	N	Not
TRANS_START	TIMESTAMP	SYSIBM	13	12	Y	Not

```
Temporal Type : System period temporal
```

```
Table is versioned and has the following periods
```

```
-----
```

Name	Type	Begin Column	End Column
SYSTEM_TIME	S	SYSTEM_START	SYSTEM_END

```
*****
```

---

## DISCONNECT

**DISCONNECT** CLPPlus コマンドは、現在のデータベース接続を閉じますが、CLPPlus セッションは終了しません。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行することができます。

### 許可

なし

### 必要な接続

データベースに接続している必要があります。

### コマンド構文

```
▶▶ DISCONNECT  
DISC ▶▶
```

## EDIT

**EDIT** CLPPlus コマンドは、ファイルまたは SQL バッファの内容を大幅に変更する目的で、外部エディターを開始します。

CLPPlus はシステム環境変数 **EDITOR** および **PATH** を読み、**EDIT** コマンドが開始された場合にどの外部エディターを使用するかを判別します。**EDITOR** システム環境変数には、システムにインストールされている任意のエディターを指定できます。**EDITOR** システム環境変数の位置情報として指定されている外部エディターのバイナリー・ファイルの位置は、**PATH** システム環境変数に含まれるものでなければなりません。それらの変数が設定されていない場合、または正しく設定されていない場合、デフォルト・エディターが使用されます。それは、Windows オペレーティング・システムの場合は「メモ帳 (Notepad)」です。UNIX オペレーティング・システム、および Linux オペレーティング・システムの場合は vi です。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

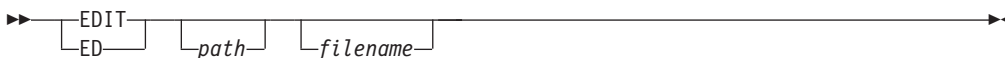
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

*path*

変数 *filename* で指定されるファイルへの、絶対パスまたは相対パスのいずれかを指定します。path が指定されない場合、現行ディレクトリーが使用されます。

*filename*

開くファイルの名前を指定します。ファイル拡張子を指定しない場合、拡張子 .sql が使用されます。*filename* パラメーターを指定しない場合、SQL バッファの内容がエディターに送られます。

## EXECUTE

**EXECUTE CPPPlus** コマンドは、現在接続が確立されているデータベースで定義されたプロシージャを実行します。変数を定義して、単一行 PL/SQL ステートメントを実行するためにも使用されます。DB2 データベース接続の場合、このコマンドを実行すると、DB2 CALL ステートメントが発行されます。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

### 許可

ステートメントの許可 ID によって保持されている特権には、少なくとも以下のいずれかが含まれていなければなりません。

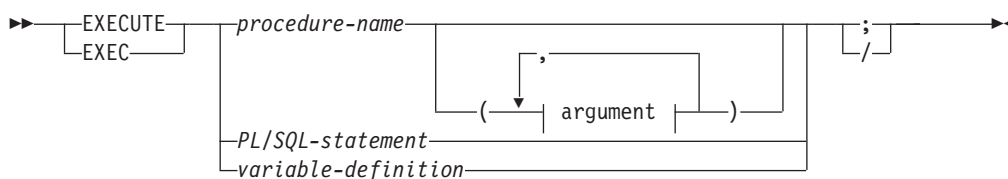
- プロシージャでの EXECUTE 特権
- DATAACCESS 権限

一致するプロシージャが存在しても、ステートメントの許可 ID がそのプロシージャの実行を許可されていない場合は、エラーが戻されます (SQLSTATE 42501)。

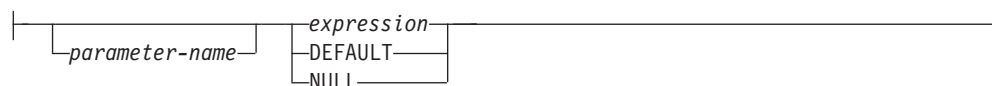
### 必要な接続

データベースに接続している必要があります。

### コマンド構文



### argument:



### コマンド・パラメーター

#### *procedure-name*

呼び出すプロシージャを指定します。プロシージャはカタログされていなければなりません。DB2 データ・サーバーの場合は、スキーマ名、モジュール名、またはスキーマ名とモジュール名の両方で名前を修飾できます。開始するプロシージャは、プロシージャ解決を使用して選択されます。DB2 データベースの場合、**EXECUTE** コマンドの実行 (プロシージャ解決を含む) は、DB2 CALL ステートメントと同じです。

#### *PL/SQL-statement*

実行する PL/SQL ステートメントを指定します。

*variable-definition*

変数の定義を指定します。

*argument**parameter-name*

DB2 データ・サーバーの場合のみ、値を割り当てるパラメーターの名前を指定します。パラメーターに名前を割り当てる場合は、後続の値割り当てでもすべて名前で行う必要があります。

プロシージャを実行するために使用するすべての名前付きパラメーターが、DB2 プロシージャ定義内に存在しなければなりません。

パラメーター名は固有でなければなりません。

カタログされていない DB2 プロシージャを、名前付きパラメーターを使用して実行することはできません。

*value*

パラメーターに関連付けられた値を指定します。*n* 番目の名前なしの値が、プロシージャの **CREATE PROCEDURE** ステートメントで定義された *n* 番目のパラメーターに対応します。名前付き値は、指定する順序に関係なく、同じ名前付きパラメーターに対応します。

*expression*

ユーザー指定のパラメーター値リストを、呼び出されるプロシージャに渡します。

**NULL**

パラメーター値として NULL を渡します。

**DEFAULT**

**CREATE PROCEDURE** ステートメントでデフォルト値が定義されている場合は、そのデフォルト値をパラメーター値として渡します。それ以外の場合は NULL を渡します。

DB2 データベースの場合は、デフォルト値が定義されていないパラメーターそれぞれに、値を指定する必要があります (SQLSTATE 428HF)。また、DB2 データベースの場合、各値はプロシージャ定義内の対応するパラメーターと以下のような互換性がなければなりません。

- IN パラメーター
  - パラメーターに割り当て可能な値でなければなりません。
  - ストリング引数の割り当てには、ストレージ割り当て規則を使用します。
- OUT パラメーター
  - 値は単一の変数またはパラメーター・マーカーでなければなりません (SQLSTATE 42886)。
  - パラメーターに割り当て可能な値でなければなりません。
  - ストリング値の割り当てには、検索割り当て規則を使用します。

注: CLPPlus インターフェースでは、出力データ・タイプが行、配列、連想配列、およびブールの値を表示することはできません。

- INOUT パラメーター



- 値は単一の変数またはパラメーター・マーカーでなければなりません (SQLSTATE 42886)。
- パラメーターに割り当て可能な値でなければなりません。
- ストリング引数の割り当てには、呼び出しについてはストレージ割り当て規則、および戻りに関しては検索割り当て規則を使用します。

## 例

1. **CREATE PROCEDURE** ステートメントを利用して、save\_tester\_details\_PROC というプロシージャを作成します。このプロシージャを **EXECUTE** コマンドで実行します。

```
> SQL> CREATE PROCEDURE save_tester_details_PROC
      (tno, IN integer, tname IN varchar, tadd IN varchar)
      AS
      BEGIN
        INSERT INTO tester1 VALUES
          (tno, tname, tadd);
      END;
/
```

```
> The SQL command completed successfully.
```

```
> SQL> EXECUTE save_tester_details_PROC(1, 'John Smith', 'Address1');
> DB250000I: The SQL command completed successfully.
```

2. **EXECUTE** コマンドは複数行にまたがっており、ブロック終了文字 / を使用して処理のためにコマンドをサブミットします。ブロック終了文字 / は、複数行にまたがるコマンドの最後で使用する必要があります。

```
SQL> exec dbms_output.put_line('test serveroutput')
2 /
test serveroutput
DB250000I: The command completed successfully.
```

3. **EXECUTE** コマンドで、単一の PL/SQL ステートメントを実行します。

```
SQL> Exec BEGIN dbms_output.put_line('TEST EXEC'); END
2
/
DB250000I: The command completed successfully.
```

4. **EXECUTE** コマンドで、変数を定義します。

```
SQL> Variable bindvar varchar(20)
SQL> Execute :bindvar := 'value' ;
```

## EXIT

**EXIT** CLPPlus コマンドは、CLPPlus セッションを終了し、制御をオペレーティング・システムに戻します。このコマンドは、**QUIT** コマンドと同義です。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

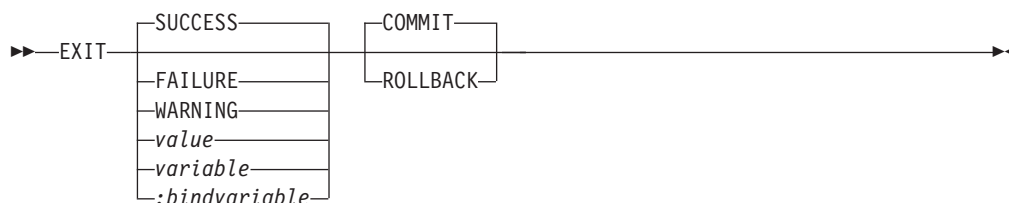
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### SUCCESS

成功を示すオペレーティング・システム依存の戻りコードを戻します。

#### FAILURE

失敗を示すオペレーティング・システム依存の戻りコードを戻します。

#### WARNING

警告を示すオペレーティング・システム依存の戻りコードを戻します。

#### *value*

値が戻りコードとして戻される、**DEFINE** コマンドで作成される変数を指定します。

#### *variable*

値が戻りコードとして戻される、**DEFINE** コマンドで作成される置換変数値を指定します。V9.7 フィックスパック 3 以降、使用可能です。

#### *:bindvariable*

値が戻りコードとして戻される、**DEFINE** コマンドで作成されるバインド変数値を指定します。V9.7 フィックスパック 3 以降、使用可能です。

#### COMMIT

コミットされていない更新が CLPPlus セッションの終了時にコミットされます。

**ROLLBACK**

コミットされていない更新が CLPPlus セッションの終了時にロールバックされます。

**例**

以下の例では、**EXIT** コマンドに **SUCCESS** パラメーターを指定して実行します。

```
SQL> exit success
C:¥>echo %errorlevel%
0
```

以下の例では、**EXIT** コマンドに **WARNING** パラメーターと **COMMIT** パラメーターを指定して実行します。

```
SQL> exit warning commit
C:¥>echo %errorlevel%
2
```

以下の例では、置換変数 *exit\_value* を **EXIT** コマンドで定義、設定、使用します。この機能は V9.7 フィックスパック 3 以降、使用可能です。

```
SQL> variable exit_value integer
DB250000I: The command completed successfully.
SQL> exec :exit_value:=5 ;
DB250000I: The command completed successfully.
SQL> exit :exit_value rollback
C:¥>echo %errorlevel%
5
```

```
SQL> define exit_value = 4
SQL> exit exit_value commit
C:¥>echo %errorlevel%
4
```

## EXPLAIN PLAN

**EXPLAIN PLAN** CLPPlus コマンドは、任意の単一の SQL ステートメントの Explain Plan を取り出します。

V9.7 フィックスパック 5 以降、**EXPLAIN PLAN** が DB2 for z/OS および IBM Informix でサポートされるようになりました。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

### 許可

なし

### 必要な接続

データベースに接続している必要があります。

### 制約事項

IBM Informix のサポートには、次のような制約事項があります。

- SELECT ステートメントのみがサポートされます。
- ONCONFIG ファイルの **SBSPACENAME** 構成パラメーターに、デフォルトの `sbspace` 名を作成および指定する必要があります。この `sbspace` は、プランの Explain が作成される際に、BLOB オブジェクトを作成するために使用されます。
- Informix サーバーから統計データを取得するには、**Informix** 特権またはそれと同等の特権が必要です。この特権を持つユーザーのみが、統計データにアクセスできます。

### 構文ダイアグラム

```
►►—EXPLAIN—PLAN—FOR—SQL-statement—◄◄
```

### コマンド・パラメーター

*SQL-statement*

Explain 情報が取り出される SQL ステートメント。IBM Informix では、**SELECT** ステートメントのみがサポートされます。

### 例

```
SQL> explain plan for select * from emp where bonus > 1000 and salary>10000;
```

ID	TYPE	OBJECT_SCHEMA	OBJECT_NAME	PREDICATE_TEXT
1	RETURN			
2	TBSCAN	MANSHANB	EMPLOYEE	(10000 < Q1.SALARY)
2	TBSCAN	MANSHANB	EMPLOYEE	(1000 < Q1.BONUS)

## GET

**GET** CLPPlus コマンドは、テキスト・ファイルの内容を CLPPlus SQL バッファにロードします。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

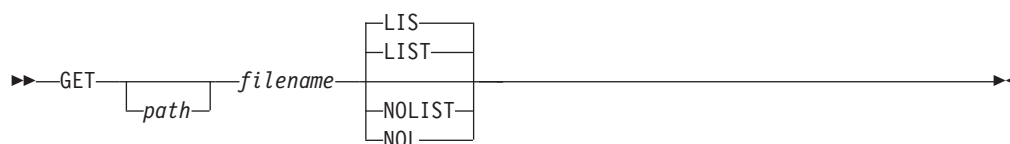
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### *path*

変数 *filename* で指定されるファイルへの、絶対パスまたは相対パスのいずれかを指定します。*path* が指定されない場合、現行ディレクトリーが使用されます。

#### *filename*

SQL バッファにロードするファイルの名前を指定します。ファイル拡張子を指定しない場合、拡張子 `.sql` が使用されます。

#### **LIST | LIS**

ファイルがロードされた後に SQL バッファの内容を表示します。

#### **NOLIST | NOL**

ファイルがロードされた後に SQL バッファの内容が表示されないようにします。

---

## HELP

**HELP** コマンドは、CLPPlus のトピックの一覧を表示するか、特定の CLPPlus コマンドのヘルプを表示します。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### INDEX

すべての CLPPlus ヘルプ・トピックの一覧を表示します。

#### *topic*

特定の CLPPlus コマンドのヘルプを表示します (例、**ACCEPT**)。

## HOST

**HOST** CLPPlus コマンドは、CLPPlus インターフェースでオペレーティング・システム・コマンドを実行します。

### 呼び出し

このコマンドは CLPPlus インターフェースで実行する必要があります。

### 許可

なし

### 必要な接続

なし。

### コマンド構文

▶ `HOST` `os_command` ▶  
HO

### コマンド・パラメーター

`os_command`

オペレーティング・システム・コマンドを指定します。

## IMPORT CLPPlus コマンド

V9.7 フィックスパック 5 以降、**IMPORT** がリモート CLPPlus クライアントからサポートされるようになりました。リモート・クライアント上のインポート・ファイルが処理されます。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

### 許可

なし

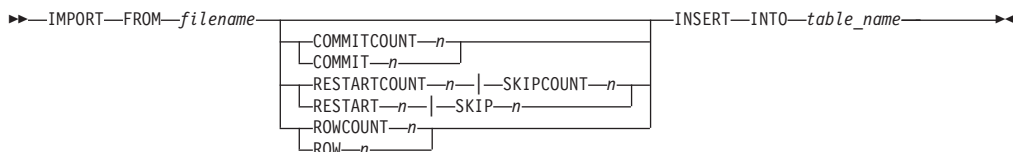
### 必要な接続

データベースに接続している必要があります。

### 制約事項

- CLPPlus インターフェースで **IMPORT** コマンドに使用可能なパラメーターは、CLP インターフェースで **IMPORT** コマンドに使用可能なパラメーターのサブセットです。
- データは、区切り文字で区切られているファイルからのみインポートできます。区切り文字で区切られているファイルは、.del、.ixf または .txt など、いずれのファイル・タイプのもので可能です。デフォルトの区切り文字は、**,** です。**SET CLPPlus** コマンドを使用して、区切り文字を別の文字に設定できます。
- システム表、宣言済み一時表、およびサマリー表をターゲット・データベース表にすることはできません。

### 構文ダイアグラム



### コマンド・パラメーター

#### filename

インポート・データを含むファイルを指定します。*filename* 変数の一部としてパスを指定できます。パスは、絶対パスまたは現行ディレクトリーに対する相対パスとして指定できます。パスを省略すると、現行ディレクトリーが検索されます。

#### COMMITCOUNT | COMMIT *n*

指定された場合、**IMPORT** は *n* 個のレコードが読み取りおよびインポートされる度にデータをコミットします。

#### RESTARTCOUNT | RESTART *n*

指定された場合、**IMPORT** は *n+1* 番目のレコードから開始します。最初の *n* 個



のレコードはスキップされます。このオプションは、**SKIPCOUNT** と機能上同等です。**RESTARTCOUNT** と **SKIPCOUNT** は相互に排他的です。

#### **SKIPCOUNT | SKIP *n***

指定された場合、**IMPORT** は  $n+1$  番目のレコードから開始します。最初の  $n$  個のレコードはスキップされます。このオプションは、**RESTARTCOUNT** と機能上同等です。**RESTARTCOUNT** と **SKIPCOUNT** は相互に排他的です。

#### **ROWCOUNT | ROW *n***

指定された場合、*filename* の先頭から  $n$  個の物理レコードがインポートされません。**ROWCOUNT** が **RESTARTCOUNT** または **SKIPCOUNT** と共に指定された場合、**IMPORT** により、**RESTARTCOUNT** または **SKIPCOUNT** で定義されたレコード以降の  $n$  行が *filename* から読み取られます。

#### *table\_name*

**IMPORT** 操作のターゲット・データベース表を指定します。この表に、システム表、宣言済み一時表、およびサマリー表を指定することはできません。スキーマにより完全修飾されない場合のデフォルト・スキーマは現行 ID です。

## 例

次の **IMPORT** コマンドは、`c:%data.txt` ファイルの最初の 100 行を読み取り、`db2admin.emptab` 表にデータを挿入します。

```
import from c:%data.txt rowcount 100 insert into db2admin.emptab;
```

次の **IMPORT** コマンドは、`data.txt` ファイルの 11 行目以降のデータを読み取り、`emptab` 表にデータを挿入します。

```
import from data.txt skip 10 insert into emptab;
```

次の **IMPORT** コマンドは、ディレクトリー・ツリー内で、現行ディレクトリーより 1 つ上のディレクトリーにある `data.txt` ファイルの 11 行目以降のデータを読み取ります。このコマンドにより、`emptab` 表にデータが挿入されます。

```
import from ../data.txt restart 10 insert into emptab;
```

---

## INPUT

**INPUT** 行エディター・コマンドは、テキストの行を SQL バッファの現在行の後に追加します。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

### 許可

なし

### 必要な接続

なし

### コマンド構文

▶▶ INPUT *text* ◀◀  
I

### コマンド・パラメーター

*text*

SQL バッファに挿入されるテキストを指定します。

### 例

以下の例では、**INPUT** コマンドのシーケンスによって **SELECT** ステートメントが構成されます。これは、**LIST** コマンドで表示されます。

```
SQL> INPUT SELECT empno, ename, job, sal, comm
SQL> INPUT FROM emp
SQL> INPUT WHERE deptno = 20
SQL> INPUT ORDER BY empno
SQL> LIST
 1 SELECT empno, ename, job, sal, comm
 2 FROM emp
 3 WHERE deptno = 20
 4* ORDER BY empno
```

## LIST

**LIST** 行エディター・コマンドは、SQL バッファ内すべての行または特定の範囲の行を表示します。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

*n* SQL バッファ内の指定された行を示します。

*n m*

SQL バッファ内の指定された範囲の行を表示します。ここで、*m* の値は *n* の値より大きくします。

\* SQL バッファ内の現在行を示します。

**LAST | L**

SQL バッファ内の最後の行を表示します。

### 例

以下の例では、**L** コマンドを使用して、行バッファの内容を表示します。

```
SQL> LIST
 1 SELECT
 2 *
 3 FROM
 4* EMPLOYEE
SQL> L2
2* *
SQL> L4
4* EMPLOYEE
SQL> L5
DB250400E: The buffer line number is invalid. Valid values can be between '1' and '4'.
SQL>
```

## LIST

以下の例では、**LIST** コマンドの変形版を使用して、行バッファの内容を表示します。

```
SQL> begin
2 dbms_output.put_line('list command ');
3 end;
4 /
```

DB250000I: The command completed successfully.

```
SQL> list
1 begin
2 dbms_output.put_line('list command ');
3* end
```

```
SQL> 2
2* dbms_output.put_line('list command ');
```

```
SQL>
```

前述の例で、2 という数字がそれ自体でどのように入力されているかに注意してください。結果として、2 行目のバッファの内容が表示されます。バッファに任意の有効な行番号を入力して、行の内容を表示させることができます。

## PAUSE

**PAUSE** CLPPlus コマンドは、CLPPlus 処理を中断し、メッセージを表示し (オプション)、ENTER キーが押されるまで処理の再開を待機します。

### 呼び出し

このコマンドは CLPPlus インターフェースで実行する必要があります。

### 許可

なし

### 必要な接続

データベースに接続している必要があります。

### コマンド構文

▶▶ `PAUSE` *optional-text* ▶▶  
    └── PAU ─┘

### コマンド・パラメーター

*optional-text*

メッセージを指定します。メッセージは、スペースおよび特殊文字を含めることができる文字ストリングです。メッセージを引用符で囲んだ場合、引用符もコマンド出力またはステートメント出力に含まれます。**optional-text** として指定された文字列は、入力されたとおりに表示されます。

---

## PRINT

**PRINT** CLPPlus コマンドは、バインド変数の値を表示します。バインド変数はリテラル値の代わりに使用されます。SELECT ステートメントを複数回発行する場合、バインド変数を使用してリテラル値の数を減らすことができます。

### 呼び出し

このコマンドは CLPPlus インターフェースで実行することができます。

### 許可

なし

### 必要な接続

データベースに接続している必要があります。

### コマンド構文

```
▶▶ PRINT bind-variable-name ◀◀  
PRI
```

### コマンド・パラメーター

#### *bind-variable-name*

スペースおよび特殊文字を含めることができる文字ストリングを指定します。*bind-variable-name* を指定しない場合、すべてのバインド変数の値が表示されます。

*bind-variable-name* のデータ・タイプが **REFCURSOR** の場合、*bind-variable-name* の指し示す結果セットの全体が読み取られ、現在の CLPPlus セッションで指定されているレポート書式に従ってすべての行が出力されます。

## PROMPT

**PROMPT** CLPPlus コマンドは、特定のメッセージをディスプレイに表示します。

**PROMPT** コマンドの出力は、CLPPlus 出力に表示されます。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

*optional-text*

メッセージを指定します。メッセージは、スペースおよび特殊文字を含めることができる文字ストリングです。メッセージを引用符で囲んだ場合、引用符もコマンド出力またはステートメント出力に含まれます。文字の大/小文字は保持されます。

---

## QUIT

**QUIT** CLPPlus コマンドは、CLPPlus セッションを終了し、制御をオペレーティング・システムに戻します。このコマンドは、**EXIT** コマンドと同義です。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

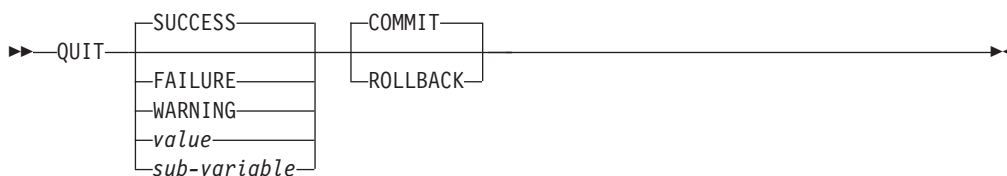
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### **SUCCESS**

成功を示すオペレーティング・システム依存の戻りコードを戻します。

#### **FAILURE**

失敗を示すオペレーティング・システム依存の戻りコードを戻します。

#### **WARNING**

警告を示すオペレーティング・システム依存の戻りコードを戻します。

#### *value*

値が戻りコードとして戻される、**DEFINE** コマンドで作成される変数を指定します。

#### *sub-variable*

情報を戻すために使用できる置換変数を指定します。

#### **COMMIT**

コミットされていない更新が CLPPlus セッションの終了時にコミットされます。

#### **ROLLBACK**

コミットされていない更新が CLPPlus セッションの終了時にロールバックされます。



---

## REMARK

**REMARK** CLPPlus コマンドは、スクリプト内のコメントのコーディングを許可します。これは、2 つのダッシュでコーディングされたコメント行と同じです。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。これは、CLPPlus スクリプトにのみ含めることができます。

### 許可

なし

### 必要な接続

なし

### コマンド構文

►► 

REMARK
REM

*optional-text* ◄◄

### コマンド・パラメーター

*optional-text*

スペースおよび特殊文字を含めることができる文字ストリングを指定します。以下の例に示されている規則を使用できます。

```
/*  
 * This is a three-line comment  
*/
```

## REORGCHK

**REORGCHK** CLPPlus コマンドは、表または索引、あるいはその両方の再編成またはクリーンアップが必要かどうかを決定するために、データベースの統計を計算します。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

### 許可

カタログ表に対する **SELECT** 特権が必要です。REORGCHK\_IX\_STATS プロシージャに対する **EXECUTE** 特権が必要です。

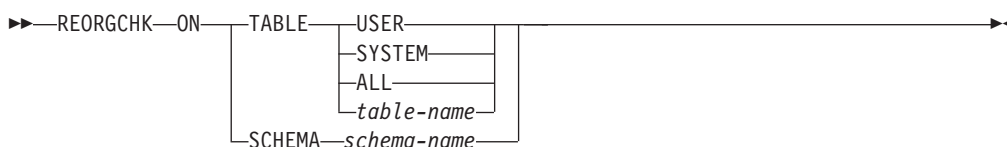
### 制約事項

このコマンドのサポートは、DB2 for Linux, UNIX, and Windows サーバーに制限されます。

### 必要な接続

データベースに接続している必要があります。

### 構文ダイアグラム



### コマンド・パラメーター

**TABLE [ USER | SYSTEM | ALL | table-name ]**

ここで、**USER** は実行時許可 ID に所有されている表をチェックし、**SYSTEM** はシステム表をチェックし、**ALL** はすべてのユーザーおよびシステム表をチェックします。そして、*table-name* はどの表をチェックするのかを指定します。*table-name* を使用する際には、完全修飾名または別名 (例えば、*schema.tablename*) を使用しなければなりません。schema には、表作成時のユーザー名が入ります。指定された表がシステム・カタログ表である場合、スキーマは **SYSIBM** です。型付き表の場合、指定された表名は、階層のルート表の名前でなければなりません。

**SCHEMA** *schema-name*

指定のスキーマの下で作成されたすべての表を検査します。

### 例

- この例は、システム表に **REORGCHK** を実行します。  
SQL> reorgchk on table system
- この例は、スキーマ manshanb の表 EMPLOYEE に **REORGCHK** を実行します。  
SQL> reorgchk on table manshanb.EMPLOYEE

3. この例は、すべてのユーザーおよびシステム表に **REORGCHK** を実行します。

```
SQL> reorgchk on table all
```

4. この例は、実行時許可 ID により所有されている表に **REORGCHK** を実行します。

```
SQL> reorgchk on table user
```

5. この例は、manshanb という名前のスキーマに **REORGCHK** を実行します。

```
SQL> reorgchk on schema manshanb
```

## REPFOOTER

**REPFOOTER** CLPPlus コマンドは、レポートの終わりに、レポート・フッターを印刷します。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

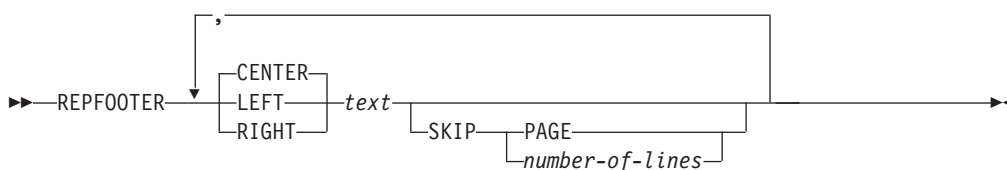
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

*text*

レポートの終わりに表示するテキストを指定します。

#### CENTER

ディスプレイで、レポート・フッターを中央揃えすることを指定します。

**CENTER**、**LEFT**、**RIGHT** のいずれも指定しない場合、中央揃えがデフォルトの動作になります。

#### LEFT

ディスプレイで、レポート・フッターのテキストを左揃えすることを指定します。

#### RIGHT

ディスプレイで、レポート・フッターのテキストを右揃えすることを指定します。

#### SKIP

**PAGE**

レポート・フッターを次のページに表示することを指定します。

*number-of-lines*

スキップする行数を指定します。

### 例

次の例では、レポート・フッター **END SALARY REPORT** を、レポートの終わりの次のページに中央揃えで表示します。

```
SQL> REPFOOTER CENTER 'END SALARY REPORT' SKIP PAGE;
```

次の例では、レポート・フッター `Company Name` を表示し、2 行スキップして、次のページに `End of Report` を中央揃えで表示します。

```
SQL> REPFOOTER CENTER "Company Name" SKIP 2, CENTER "End of Report" SKIP PAGE
```

## REPHEADER

**REPHEADER** CLPPlus コマンドは、レポートの先頭に、レポート見出しを 1 回印刷します。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

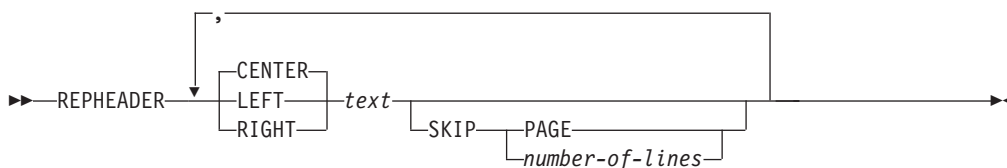
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

*text*

レポート・ヘッダーに表示するテキストを指定します。

#### **CENTER**

ディスプレイで、レポート・ヘッダーを中央揃えすることを指定します。

**CENTER**、**LEFT**、**RIGHT** のいずれも指定しない場合、中央揃えがデフォルトの動作になります。

#### **LEFT**

ディスプレイで、レポート・ヘッダーのテキストを左揃えすることを指定します。

#### **RIGHT**

ディスプレイで、レポート・ヘッダーのテキストを右揃えすることを指定します。

#### **SKIP**

**PAGE**

レポート・ヘッダーを表示して、レポート・データが次のページから開始されることを指定します。

*number-of-lines*

スキップする行数を指定します。

### 例

次の例では、レポート・ヘッダー SALARY REPORT を左揃えで表示して、レポート・データが次のページから開始されることを指定しています。

```
SQL> REPHEADER LEFT 'SALARY REPORT' SKIP PAGE;
```

次の例では、レポート・ヘッダー COMPANY NAME を表示し、2 行スキップして、SALARY REPORT をすべて中央揃えで表示します。レポート・データは次のページから開始します。

```
SQL> REPHEADER CENTER 'COMPANY NAME' SKIP 2, CENTER 'SALARY REPORT' SKIP PAGE;
```

---

## RUN

**RUN** CLPPlus コマンドは、SQL バッファに保管されている照会を実行します。SQL バッファはコマンドおよびステートメントを複数行に保管できるため、バッファ内の現在行にあるコマンドまたはステートメントが実行されます。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

### 許可

なし

### 必要な接続

データベースに接続している必要があります。

### コマンド構文

▶—RUN—◀

### 例

以下の例では、SQL バッファの現在行の内容は `SELECT EMPNO FROM EMP` です。**RUN** コマンドは、EMPNO 列の値を EMP 表から選択するステートメントを発行します。

```
SQL> run
1* SELECT EMPNO FROM EMP
```

出力は以下のとおりです。

```
EMPNO
-----
000010
000020
...
...
```



## SAVE

SAVE 行エディター・コマンドは、SQL バッファの内容を新規または既存のファイルに保管します。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

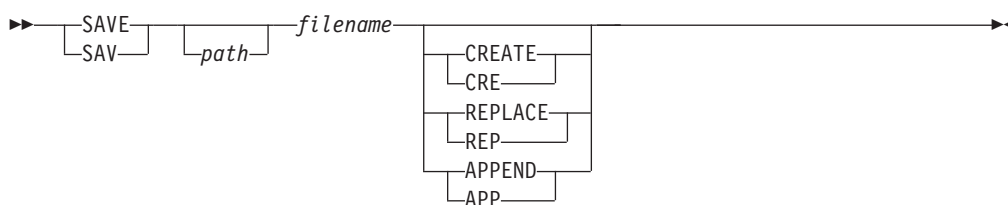
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### *path*

使用されるファイルへの、絶対パスまたは相対パスのいずれかを指定します。  
*path* が指定されない場合、現行ディレクトリーが使用されます。

#### *filename*

バッファの内容が書き込まれるファイルの名前を指定します。ファイル拡張子を指定しない場合、拡張子 `.sql` が使用されます。

#### CREATE | CRE

指定されたファイルが存在しない場合に作成します。

#### REPLACE | REP

指定されたファイルを上書きします。

#### APPEND | APP

SQL バッファの内容を指定されたファイルの最後に追加します。

---

## SET

**SET** CLPPlus コマンドはセッション・レベルの変数を制御します。

### 重要:

- **SET** コマンドの呼び出しごとに、パラメーターを 1 つのみ指定できます。
- バッチ・スクリプトでは、**SET** コマンドを連続して発行できます。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

### 許可

なし

### 必要な接続

なし

### 構文ダイアグラム

→→SET	
ARRAYSIZE	<i>integer-value</i>
AUTOCOMMIT	ON
AUTO	OFF
	IMMEDIATE
	<i>statement_count</i>
AUTOTRACE	ON
AUTOT	OFF
	TRACEONLY
	TRACE
	EXPLAIN
	EXP
	STATISTICS
	STAT
BGCOLOR	<i>color</i>
	<i>r,g,b</i>
COLOR	<i>color</i>
	<i>r,g,b</i>
COLSEP	<i>column-separator</i>
DUPLICATES	ON
	OFF
ECHO	ON
	OFF
ENVARSUBSTITUTION	ON
ENVARSUBST	OFF
ESCAPE	ON
	OFF
	<i>character</i>
FEEDBACK	ON
FEED	OFF
	<i>row-threshold</i>
FLUSH	ON
FLU	OFF
FONT	<i>name</i> , <i>style</i> , <i>size</i>
HEADING	ON
HEAD	OFF
HEA	
HEADSEP	
HEADS	
LOGMODE	<i>logmode-value</i>
JCCLOGMODE	<i>jcc logmode-value</i>
LINESIZE	<i>width-of-line</i>
LIN	
LOGPATH	<i>log-path</i>
LONG	<i>integer-value</i>
NEWPAGE	<i>lines-per-page</i>
NEWP	
NULL	<i>null-string</i>
NUMFORMAT	<i>format-string</i>
NUMWIDTH	OFF
	<i>integer-value</i>
PAGESIZE	<i>lines-per-page</i>
PAGES	
SQLCASE	MIXED
SQLC	MIX
	UPPER
	UP
	LOWER
	LO
PAUSE	ON
PAU	OFF
RECSEP	ON
	OFF
RECSEPCHAR	<i>character</i>
SERVEROUTPUT	ON
	OFF
	UNLIMITED
	SIZE <i>n</i>
	FORMAT
	WRAPPED
	WORD_WRAPPED
	TRUNCATED
SQLPROMPT	<i>prompt</i>
SQLP	
TERMOUT	ON
TERM	OFF
TIMING	ON
TIMI	OFF
TRIMOUT	ON
TRIMO	OFF
TRIMSPool	ON
TRIMS	OFF
UNDERLINE	ON
	OFF
VERBOSE	ON
	OFF
VERIFY	ON
VER	OFF
WRAP	ON
	OFF

## コマンド・パラメーター

### ARRAYSIZE *integer-value*

サーバーから一度にフェッチされる行数を定義します。このオプションを使用して、照会のパフォーマンスを調整することができます。デフォルト値は 10 です。値の有効範囲は 1 から 10,000 です。

### AUTOCOMMIT | AUTO

CLPPlus の SQL ステートメントのコミット動作を制御します。CLPPlus は常に、DDL ステートメントを自動的にコミットします。

#### ON | IMMEDIATE

SQL ステートメントの自動コミットメントを使用可能にします。

#### OFF

DDL ステートメント以外の SQL ステートメントの自動コミットメントを使用不可にします。

### AUTOTRACE | AUTOT

CLPPlus セッション中の SQL ステートメントのプランの Explain と統計情報の表示を有効にします。いったん AUTOTRACE が ON に設定されると、セッションが終了するか AUTOTRACE が OFF になるまで、CLPPlus は Explain 情報を表示し続けます。

V9.7 フィックスパック 5 以降、**AUTOTRACE** パラメーターが DB2 for z/OS でサポートされるようになりました。V9.7 フィックスパック 5 以降、**AUTOTRACE** パラメーターが IBM Informix でもサポートされるようになりましたが、以下の制限があります。

- EXPLAIN オプションは、SELECT ステートメントでのみサポートされます。
- EXPLAIN オプションを指定する場合、ONCONFIG ファイルの **SBSPACENAME** 構成パラメーターに、デフォルトの `sbspace` 名を作成および指定する必要があります。この `sbspace` 名は、プランの Explain が作成される際に、BLOB を作成するために使用されます。
- Informix サーバーから統計データを取得するには、Informix 特権またはそれと同等の特権が必要です。

**ON** AUTOTRACE を使用可能にします。

#### OFF

AUTOTRACE を使用不可にします。

### TRACEONLY | TRACE

照会実行出力の表示を使用不可にします。

### EXPLAIN | EXP

プランの Explain の表示を使用可能にします。

### STATISTICS | STAT

ステートメントの統計情報の表示を使用可能にします。

### BGCOLOR *color|r,g,b*

バージョン 10.1 フィックスパック 2 以降、このパラメーターはウィンドウ・モードの背景色を設定します。

*color*

*color* 変数の有効な値は、BLACK|black、BLUE|blue、CYAN|cyan、DARK\_GRAY|darkGray、GRAY|gray、GREEN|green、LIGHT\_GRAY|lightGray、MAGENTA|magenta、ORANGE|orange、PINK|pink、RED|red、WHITE|white、YELLOW|yellow です。

*r,g,b*

指定された赤、緑、および青の値を使用して、不透明な RGB 色を設定します。赤、緑、および青の値として有効な範囲は、0 から 255 です。無効な値はすべて 255 として扱われます。

**COLOR** *color*|*r,g,b*

バージョン 10.1 フィックスパック 2 以降、このパラメーターはウィンドウ・モードのフォントの色を設定します。

*color*

*color* 変数の有効な値は、BLACK|black、BLUE|blue、CYAN|cyan、DARK\_GRAY|darkGray、GRAY|gray、GREEN|green、LIGHT\_GRAY|lightGray、MAGENTA|magenta、ORANGE|orange、PINK|pink、RED|red、WHITE|white、YELLOW|yellow です。

*r,g,b*

指定された赤、緑、および青の値を使用して、不透明な RGB 色を設定します。赤、緑、および青の値として有効な範囲は、0 から 255 です。無効な値はすべて 255 として扱われます。

**COLSEP** *column-separator*

表の列間に、指定された区切り文字を配置します。区切り文字は、特殊文字またはスペース (デフォルト値) などの文字でなければなりません。

**DUPLICATES**

**BREAK** コマンドを使用したときに指定されるブレイク列の重複列の値の印刷を制御します。

**ON** 重複列の値の印刷ができます。

**OFF**

重複列の値の印刷はできません。

**ECHO**

すべてのコマンドおよび関連の出力を CLPPlus インターフェースの標準出力に表示するかどうかを制御します。

**ON** コマンドとその出力の表示を使用可能にします。

**OFF**

コマンドとその出力の表示を使用不可にします。

**ENVVARSUBSTITUTION | ENVVARSUBST**

CLPPlus インターフェースが環境変数置換をサポートするかどうかを制御します。デフォルトの値は ON です。

**ON** 環境変数置換を使用可能にします。CLPPlus は、\$ 文字が接頭部に付いているすべてのテキスト、および % 文字でラップされているテキストを環境変数として処理し、関連付けられた値に置き換えます。

**OFF**

環境変数置換を使用不可にします。

**ESCAPE**

CLPPPlus インターフェイスでエスケープ文字を使用するよう設定するかどうかを制御します。

**ON** デフォルトのエスケープ文字「¥」を使用可能にします。

**OFF**

現在定義されているエスケープ文字を使用不可にします。

*character*

*character* に指定された値でエスケープ文字を使用可能にします。

**FEEDBACK | FEED**

SQL ステートメントを発行した後で、対話式情報の表示を制御します。

**ON** 対話式情報の表示を使用可能にします。これはデフォルトのアクションです。

**OFF**

対話式情報の表示を使用不可にします。

*row-threshold*

フィードバックを可能にするために必要な最低行数を指定します。

**FLUSH | FLU**

出力バッファが外部プログラムにアクセス可能であるかどうかを制御します。出力バッファが追加されている間は、**FLUSH** はアクティブなままです。ただし、ステートメントを呼び出したり、あるいはコマンドを実行したりする際、このプロセスによってより大きな処理能力が要求されます。

**ON** バッファが外部プログラムにアクセスできるようにします。

**OFF**

バッファが外部プログラムにアクセスできないようにします。

**FONT**

バージョン 10.1 フィックスパック 2 以降、このパラメーターはウィンドウ・モードのフォントの名前、スタイル、およびサイズを設定します。

*name*

設定するフォントの名前を指定します。指定可能な値は、*serif*、*sansserif*、*monospaced*、または有効なシステム・フォントの名前です。

*style*

フォント・スタイルを指定します。指定可能な値は、プレーン・テキストの場合は 0、太字の場合は 1、イタリックの場合は 2、太字のイタリックの場合は 3 です。

*size*

フォント・サイズを指定します。受け入れられる値は整数です。

**HEADING | HEA**

SELECT ステートメントの列見出しを表示するかどうかを決定します。

**ON** 列見出しの表示を使用可能にします。

**OFF**

列見出しの表示を使用不可にします。

**HEADSEP | HEADS**

**COLUMN HEADING** コマンドで使用される見出し区切り文字を設定します。デフォルトの文字は | です。

**LOGMODE** *logmode-value*

**CLPPlus** クライアント・レイヤーおよび JDBC ドライバー・レイヤー (JCC) のトレースおよびロギングを制御します。

**CLPPLUS**

**CLPPlus** クライアント・レイヤーのみのトレースおよびロギングを実行します。

**JCC**

**JDBC** クライアント・レイヤーのみのトレースおよびロギングを実行します。

**BOTH**

**CLPPlus** クライアント・レイヤーおよび **JDBC** クライアント・レイヤーのトレースおよびロギングを実行します。

**NONE**

すべてのトレースおよびロギングを使用不可にします。

**JCCLOGMODE** *jcclogmode-value*

トレースとロギングのいずれかまたは両方の対象となる **JCC** クライアント・レイヤー・フィーチャーを指定します。 *jcclogmode-value* を使用するには、**LOGMODE** を 1 または 2 に設定する必要があります。

**0** (TRACE\_NONE)

**1** (TRACE\_CONNECTION\_CALLS)

**2** (TRACE\_STATEMENT\_CALLS)

**4** (TRACE\_RESULT\_SET\_CALLS)

**16** (TRACE\_DRIVER\_CONFIGURATION)

**32** (TRACE\_CONNECTS)

**64** (TRACE\_DRDA\_FLOWS)

**128**

(TRACE\_RESULT\_SET\_META\_DATA)

**256**

(TRACE\_PARAMETER\_META\_DATA)

**512**

(TRACE\_DIAGNOSTICS)

**1024**

(TRACE\_SQLJ)

**2048**

(TRACE\_XA\_CALLS)

**-1 (TRACE\_ALL)**。デフォルトで、-1 の設定が使用され、すべてのレイヤーがトレース対象であることを意味します。

#### **LINESIZE** | **LIN** *width-of-line*

キーワード **LINESIZE** とその省略形 **LIN** のどちらを使用しても構いません。

**LINESIZE** は行の幅を文字数で指定します。

この値は、有効範囲が **1** から **32767** である文字数による行の幅です。デフォルトでは、これは **80** に設定されています。

#### **LOGPATH**

**LOGPATH** は、データベース・サーバー上でログ・レコードが書き込まれるファイルのパス名を設定します。

##### *log-path*

**LOGMODE** および **JCCLOGMODE** を使用して、トレースのロギング・レコードを保持するために使用されるファイル名のパス。

#### **LONG** *integer-value*

**CLOB** や **XML** など、ラージ・テキスト・オブジェクトに表示する文字数を定義します。デフォルト値は **50** です。有効範囲は、**1** 文字から **2,147,483,647** 文字までです。

#### **NEWPAGE** | **NEWP**

**NEWPAGE** は、改ページの後に表示されるブランク行の数を制御します。

##### *lines-per-page*

負の数値または **100** より大きい値を含まない整数値。デフォルトでは、これは **1** に設定されており、改ページの後にブランク行が **1** 行表示されることを示しています。コマンド **SET NEWPAGE 0** は、それぞれの新規ページの先頭に改ページ制御文字を出力します。

#### **NULL**

キーワード **NULL** は、出力バッファーに表示される列値で **NULL** 値を示すときに表示されるストリングです。

##### *null-string*

スペースおよび特殊文字を含めることができる文字ストリング。デフォルトでは、これはスペースに設定されています。*null-string* を引用符で囲んでも、引用符は表示されません。文字の大/小文字は保持されます。

#### **NUMFORMAT** *format-string*

数字を表示するときに使用するデフォルトの書式制御ストリングを設定します。サポートされる書式は、**COLUMN FORMAT** *format-string* の場合と同じです。

#### **NUMWIDTH**

数字の表示に使用するデフォルト幅を設定します。デフォルト値は **OFF** です。

#### **PAGESIZE** | **PAGES** *lines\_per\_page*

**1** ページに収まる表示行の数を設定します。デフォルト値は **25** です。有効な値は **0** および **2** から **50000** の値です。

#### **SQLCASE** | **SQLC**

サーバーに伝送される **SQL** ステートメントの文字を大文字に変換するか、小文字に変換するかを制御します。



**MIXED | MIX**

文字ストリングに大文字および小文字を含めることができることを指定します。

**UPPER | UP**

文字ストリングをすべて大文字に変換します。

**LOWER | LO**

文字ストリングに小文字のみを含めることができることを指定します。

**PAUSE | PAU**

それぞれの改ページの前に処理を停止するかどうかを決定します。出力を 1 つのページに表示できない場合、それぞれの改ページの前にメッセージ Hit ENTER to continue とともにプロンプトが出されます。

**ON** 出力の表示を一時停止します。

**OFF**

一時停止せずに出力を表示します。

**RECSEP**

結果セット内の各レコードを印刷した後に、**RECSEPCHAR** オプションで指定するレコード分離文字セットを表示するかどうかを指定します。

**ON** レコード分離文字を、結果セット内の各レコードの後に印刷します。

**OFF**

レコード分離文字を印刷しません。

**RECSEPCHAR character**

**RECSEP** オプションとともに使用するレコード分離文字を指定します。単一文字は、このオプションの値として受け入れられます。

**SERVEROUTPUT**

サーバー・サイドのプロシージャーからの出力メッセージが取り出されて、クライアント・コンソールに表示されるかどうかを指定します。V9.7 フィックスパック 3 以降、**SERVEROUTPUT** では **SIZE** と **FORMAT** をオプションとして指定できます。

**ON** サーバー・サイド・プロシージャーの出力メッセージが取り出されて表示されます。

**OFF**

サーバー・サイド・プロシージャーの出力メッセージの取り出しおよび表示は実行されません。

**SIZE**

画面に表示する文字数を指定します。デフォルト値は **UNLIMITED** です。 $n$  は任意の正整数です。

**FORMAT**

コンソールにサーバー出力を表示する際に使用する形式を指定します。

**WRAPPED** の場合、必要なときにはテキストを次の行にオーバーフローさせることが可能です。**WORD\_WRAPPED** の場合、テキストを次の行にオーバーフローさせますが、その際、単語が 2 行にまたがって分割されるこ

とがなくなります。TRUNCATED では、行サイズを超過するテキストは切り捨てられます。FORMAT の使用時には、このうちのいずれかのオプションを指定する必要があります。

**SQLPROMPT | SQLP prompt**

CLPPlus インターフェースのプロンプトを指定します。デフォルトでは、プロンプトは SQL> です。このプロンプトはストリングでなければならず、特殊文字およびスペースを含めることができます。ストリングを引用符で囲んでも、引用符は表示されません。また、文字の大/小文字は保持されます。

**TERMOUT | TERM**

出力を CLPPlus インターフェースの標準出力に表示するかどうかを決定します。

**ON** 画面に出力を表示します。

**OFF**

出力は表示されません。

**TIMING | TIMI**

SQL ステートメントが発行された後に、SQL ステートメントごとに経過時間を表示するかどうかを制御します。

**ON** 経過時間を表示することを指定します。

**OFF**

経過時間を表示しないことを指定します。

**TRIMOUT | TRIMO**

出力がコンソールに書き込まれる前に、出力から末尾ブランク・スペースを削除するかどうかを制御します。デフォルトでは、末尾ブランク・スペースは削除されません。

**ON** 末尾ブランク・スペースが削除されることを指定します。

**OFF**

末尾ブランク・スペースが削除されないことを指定します。

**TRIMSPool | TRIMS**

スプール出力がスプール・ファイルに書き込まれる前に、スプール出力から末尾ブランク・スペースを削除するかどうかを制御します。デフォルトでは、末尾ブランク・スペースは削除されません。

**ON** 末尾ブランク・スペースが削除されることを指定します。

**OFF**

末尾ブランク・スペースが削除されないことを指定します。

**UNDERLINE**

列見出しに下線を付けるかどうかを指定します。

**ON** 列見出しに下線を付けます。

**OFF**

列見出しに下線を付けません。

**VERBOSE**

CLPPlus のすべてのメッセージがコンソールに出力されるかどうかを決定します。デフォルト値は OFF です。

**ON** CLPPlus のすべてのメッセージをコンソールに出力することを指定します。

**OFF**

メッセージのうち特定のサブセットのみコンソールに出力することを指定します。

#### VERIFY | VER

置換変数を使用したときに SQL ステートメントの元の値と新しい値を表示するかどうかを決定します。

**ON** 古い値と新しい値を表示することを指定します。

**OFF**

古い値と新しい値を表示しないことを指定します。

#### WRAP

列の値を表示するときに使用するデフォルトの桁合わせを設定します。

**ON** 列幅を超える列値を折り返します。

**OFF**

列幅を超える列値を切り捨てます。

---

## SPOOL

**SPOOL** CLPPlus コマンドは、CLPPlus 出力をエコー出力し、ファイルにログ記録します。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

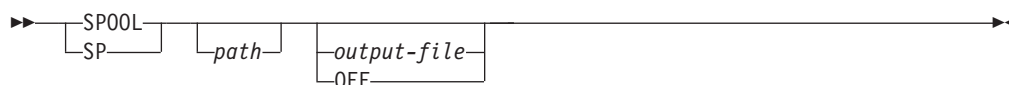
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### *path*

出力ファイルへの、絶対パスまたは相対パスのいずれかを指定します。 *path* が指定されない場合、現行ディレクトリーが使用されます。

#### *output-file*

変数 *output-file* が設定されている場合、**SPOOL** はオンになります。コマンド出力は、CLPPlus インターフェースの標準出力に表示されると同時に、指定の *output-file* にログとして記録されます。

#### **OFF**

キーワード・オプション **OFF** は **SPOOL** をオフにし、デフォルト (すべての出力を CLPPlus インターフェースの標準出力のみに送信する) に戻します。

## SHOW

**SHOW** CLPPlus コマンドは、CLPPlus インターフェースにおけるセッション・レベル変数の現在の設定値、またはサーバー・サイド・プロシージャから返されるエラーを表示します。設定は、**SET** コマンドを使用して制御されます。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

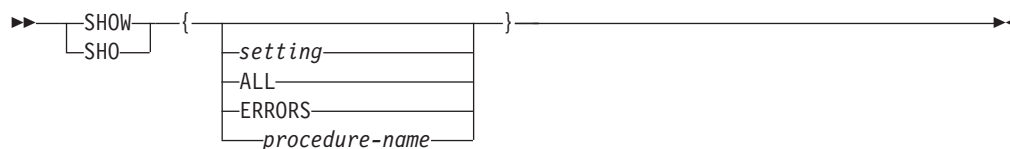
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### *setting*

指定されたセッション・レベル変数の名前および設定を表示します。

#### **ALL**

すべてのセッション・レベル変数の名前および設定を表示します。

#### **ERRORS**

現在の CLPPlus セッションで実行されるすべてのサーバー・サイド・プロシージャでのエラーを表示します。

#### *procedure-name*

**SHOW ERRORS** コマンドに付加して指定した場合、*procedure-name* についてのみエラーを表示します。

## START

**START** CLPPlus コマンドは、CLPPlus スクリプト・ファイルを実行します。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### *path*

実行する SQL ステートメントおよびコマンドが含まれるスクリプト・ファイルへの、絶対パスまたは相対パスのいずれかを指定します。 *path* が指定されない場合、現行ディレクトリーが使用されます。

#### *URL*

V9.7 フィックスパック 6 の時点では、実行する SQL ステートメントおよびコマンドが含まれるスクリプト・ファイルへの URL を指定します。 URL は `http://` または `https://` で始まらなければなりません。

#### *script-file*

実行する SQL ステートメントおよびコマンドが含まれるスクリプト・ファイル名を指定します。

### 例

この例では、CLPPlus が IP アドレス 9.124.159.144 のコンピューター上にある `demo.sql` という名前のスクリプトを開始する様子を示します。

```
SQL> Start http://9.124.159.144/demo.sql
```

## TTITLE

TTITLE CLPPlus コマンドは、表示される各ページの上部に、テキストを挿入します。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

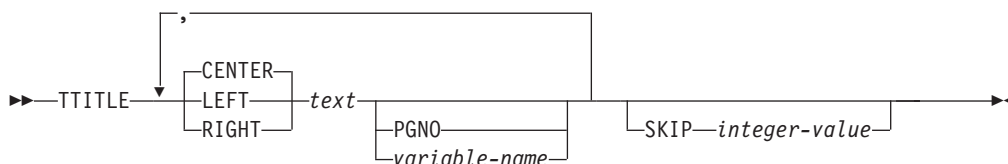
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

*text*

表示するテキストを指定します。

#### CENTER

ディスプレイで、テキストを各ページで中央揃えすることを指定します。

**CENTER**、**LEFT**、**RIGHT** のいずれも指定しない場合、中央揃えがデフォルトの動作になります。

#### LEFT

ディスプレイで、テキストを各ページで左揃えすることを指定します。

#### RIGHT

ディスプレイで、テキストを各ページで右揃えすることを指定します。

#### PGNO

現行のページ番号を指定します。

*variable-name*

*text* フィールドに続くユーザー定義変数を指定します。

#### SKIP *integer-value*

*integer-value* 値は、上部のタイトルの後に表示されるブランク行の行数を指定します。

### 例

## TTITLE

次の例では、DEPT: (および変数の内容)、CONFIDENTIAL、Page No: (および現行のページ番号) が各ページの上部に表示されます。上部のタイトルの後には、ブランク行が 1 行続きます。

```
SQL> BREAK ON workdept SKIP PAGE;  
SQL> COLUMN workdept NEW_VALUE new_dept;  
SQL> TTITLE LEFT 'DEPT: ' new_dept, CENTER 'CONFIDENTIAL', RIGHT 'Page No: ' PGNO SKIP 1;
```

次の例では、Page No: タイトル (および現行のページ番号) が各ページの上部に、右揃えで表示されます。上部のタイトルの後には、ブランク行が 2 行続きます。

```
SQL> TTITLE RIGHT 'Page No: ' PGNO SKIP 2;
```



## UNDEFINE

**UNDEFINE** CLPPlus コマンドは、**DEFINE** CLPPlus コマンドによって作成された変数をクリアおよび削除します。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

### 許可

なし

### 必要な接続

データベースに接続している必要があります。

### コマンド構文



### コマンド・パラメーター

*variable-name*

クリアおよび削除する変数の名前を指定します。

## WHENEVER OSERROR

V9.7 フィックスパック 3 以降、**WHENEVER OSERROR** CLPPlus コマンドは、オペレーティング・システムでエラーが発生した場合に実行する CLPPlus アクションを指定します。このコマンドを使用すると、エラーをトラップし、**EXIT** や **CONTINUE** などの指定のアクションを実行して、CLPPlus 動作を制御できます。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

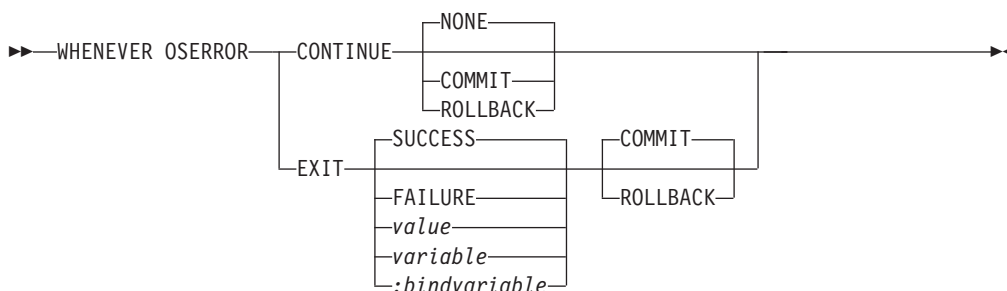
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### CONTINUE

SQL または PL/SQL エラーが発生した場合に、CLPPlus が指定のアクションを実行して続行するように指示します。

#### NONE

**WHENEVER OSERROR CONTINUE** コマンドで使用されるデフォルト値です。エラーが発生した SQL ブロックで何もアクションを実行しません。

#### COMMIT

**WHENEVER OSERROR CONTINUE** コマンドで **COMMIT** を指定すると、現在の SQL ブロックで実行された、可能なすべての作業がコミットされます。

#### ROLLBACK

**WHENEVER OSERROR CONTINUE** コマンドで **ROLLBACK** を指定すると、現在の SQL ブロック内のすべての作業がロールバックされます。

#### EXIT

オペレーティング・システムでエラーが発生した場合に、CLPPlus が終了するように指定します。このオプションの機能は、単体の **EXIT** コマンドと同じです。

**SUCCESS**

成功を示すオペレーティング・システム依存の戻りコードを戻します。これは、1 番目のデフォルトの **EXIT** パラメーターです。

**FAILURE**

失敗を示すオペレーティング・システム依存の戻りコードを戻します。

*value*

値が戻りコードとして戻される、**DEFINE** コマンドで作成される変数を指定します。

*variable*

値が戻りコードとして戻される、**DEFINE** コマンドで作成される置換変数値を指定します。

*:bindvariable*

値が戻りコードとして戻される、**DEFINE** コマンドで作成されるバインド変数値を指定します。

**COMMIT**

コミットされていない更新が CLPPlus セッションの終了時にコミットされます。これは、2 番目のデフォルトの **EXIT** パラメーターです。

**ROLLBACK**

コミットされていない更新が CLPPlus セッションの終了時にロールバックされます。

**例**

以下の例は、**EXIT** と終了エラー値が指定された場合のこのコマンドの動作を示しています。

```
SQL> whenever oserror exit -1
SQL> get c:%nonexistingfile.sql
DB250204E: An attempt to locate a file 'c:%nonexistingfile.sql' failed. The command cannot be processed.
c:%>echo %errorlevel%
-1
```

以下の例は、**CONTINUE** が指定された場合のこのコマンドの動作を示しています。

```
SQL> whenever oserror continue
SQL> get c:%nonexistingfile.sql
DB250204E: An attempt to locate a file 'c:%nonexistingfile.sql' failed. The command cannot be processed.
SQL>
```

## WHENEVER SQLERROR

V9.7 フィックスパック 3 以降、**WHENEVER SQLERROR** CLPPlus コマンドは、SQL または PL/SQL で SQL エラーが生じた場合に実行する CLPPlus アクションを指定します。このコマンドを使用すると、エラーをトラップし、**EXIT** や **CONTINUE** などの指定のアクションを実行して、CLPPlus 動作を制御できます。

### 呼び出し

このコマンドは CLPPlus インターフェースから実行する必要があります。

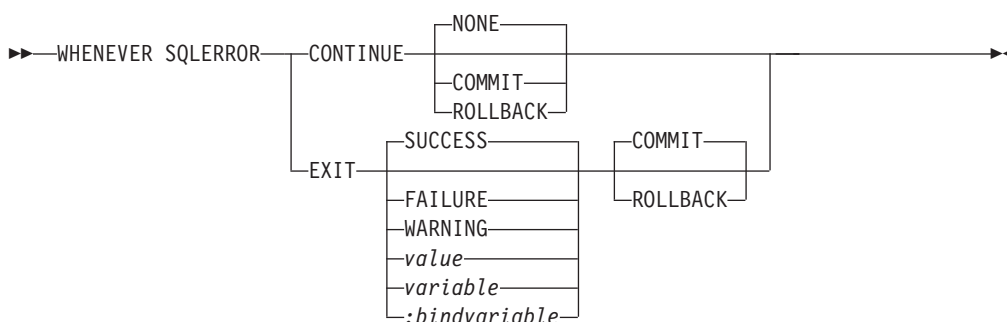
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### CONTINUE

SQL または PL/SQL エラーが発生した場合に、CLPPlus が指定のアクションを実行して続行するように指示します。

#### NONE

**WHENEVER SQLERROR CONTINUE** コマンドで使用されるデフォルト値です。エラーが発生した SQL ブロックで何もアクションを実行しません。

#### COMMIT

**WHENEVER SQLERROR CONTINUE** コマンドで **COMMIT** を指定すると、現在の SQL ブロックで実行された、可能なすべての作業がコミットされます。

#### ROLLBACK

**WHENEVER SQLERROR CONTINUE** コマンドで **ROLLBACK** を指定すると、現在の SQL ブロック内のすべての作業がロールバックされます。

#### EXIT

SQL または PL/SQL エラーが発生した場合に、CLPPlus が終了するように指定します。このオプションの機能は、単体の **EXIT** コマンドと同じです。

**SUCCESS**

成功を示すオペレーティング・システム依存の戻りコードを戻します。これは、1 番目のデフォルトの **EXIT** パラメーターです。

**FAILURE**

失敗を示すオペレーティング・システム依存の戻りコードを戻します。

**WARNING**

警告を示すオペレーティング・システム依存の戻りコードを戻します。

*value*

値が戻りコードとして戻される、**DEFINE** コマンドで作成される変数を指定します。

*variable*

値が戻りコードとして戻される、**DEFINE** コマンドで作成される置換変数値を指定します。

*:bindvariable*

値が戻りコードとして戻される、**DEFINE** コマンドで作成されるバインド変数値を指定します。

**COMMIT**

コミットされていない更新が CLPPlus セッションの終了時にコミットされます。これは、2 番目のデフォルトの **EXIT** パラメーターです。

**ROLLBACK**

コミットされていない更新が CLPPlus セッションの終了時にロールバックされます。

**例**

以下の例は、**WHENEVER SQLERROR CONTINUE** コマンドの動作を示しています。CLPPlus プロンプトに戻り、CLPPlus を引き続き使用できます。

```
SQL> whenever sqlerror continue
SQL> select * from nonexistingtable;
SQL0204N "SCHEMA.NONEXISTINGTABLE" is an undefined name.
SQL>
```

SQL エラーが発生する場合、アクションをコミットするか、ロールバックするか、アクションをとらないことも可能です。

```
SQL> whenever sqlerror continue commit
SQL>
```

```
SQL> whenever sqlerror continue commit
SQL>
```

```
SQL> whenever sqlerror continue none
SQL>
```

以下の例では、**EXIT** オプションを使用して CLPPlus アプリケーションを終了します。

```
SQL> whenever sqlerror exit
SQL> select * from nonexistingtable;
SQL0204N "SCHEMA.NONEXISTINGTABLE" is an undefined name.
```

```
C:¥>
```

## WHENEVER SQLERROR

以下の場合、終了時に戻されるエラー・コードが指定されます。この動作は、EXIT CLPPlus コマンドと同じです。

```
SQL> whenever sqlerror exit success
SQL> whenever sqlerror exit failure
SQL> select * from nonexistenttable;
SQL0204N "SCHEMA.NONEXISTINGTABLE" is an undefined name.
```

```
C:¥echo %errorlevel%
1
```

```
SQL> define exit_value=6
SQL> whenever sqlerror exit exit_value
SQL> select * from nonexistenttable;
SQL0204N "SCHEMA.NONEXISTINGTABLE" is an undefined name.
```

```
C:¥echo %errorlevel%
6
```

EXIT CLPPlus コマンドと同様、CLPPlus アプリケーション終了の間に、コミットするのか、ロールバックするのかを指定できます。

```
SQL> whenever sqlerror exit 2 commit
SQL> whenever sqlerror exit 2 rollback
```

---

## 第 7 章 システム・コマンド

## dasauto - DB2 Administration Server 自動始動

DB2 Administration Server の自動始動を有効または無効にします。

**重要:** バージョン 9.7 で DB2 Administration Server (DAS) が非推奨になったため、このコマンドは推奨されておらず、将来のリリースで削除される可能性があります。詳しくは、『DB2 Administration Server (DAS) が推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0059276.html>) を参照してください。

このコマンドは Linux および UNIX システムでのみ使用できます。これは、*DB2DIR*/das/adm ディレクトリーにあります。*DB2DIR* は、DB2 データベース製品の現行バージョンがインストールされている場所です。

### 許可

DASADM

### 必要な接続

なし

### コマンド構文

```
→ dasauto [-h] [-?] [-on] [-off] →
```

### コマンド・パラメーター

**-h | -?**

ヘルプ情報を表示します。このオプションを指定すると、他のすべてのオプションは無視され、ヘルプ情報だけが表示されます。

**-on**

DB2 Administration Server の自動始動を有効にします。次にシステムが再開したときに、DB2 Administration Server は自動的に開始します。

**-off**

DB2 Administration Server の自動始動を無効にします。次にシステムが再開しても、DB2 Administration Server は自動的に開始しません。



## dasrcrt - DB2 Administration Server の作成

DB2 Administration Server は、リモート管理や、レプリケーション・センターなどの DB2 管理ツールをサポートするサービスを提供します。システムに DAS がない場合、このコマンドを使って手動で生成できます。

**重要:** バージョン 9.7 で DB2 Administration Server (DAS) が非推奨になったため、このコマンドは推奨されておらず、将来のリリースで削除される可能性があります。詳しくは、『DB2 Administration Server (DAS) が推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0059276.html>) を参照してください。

**dasrcrt** コマンドは *DB2DIR/instance* ディレクトリーにあります。*DB2DIR* は、DB2 データベース製品の現行バージョンがインストールされている場所です。

このコマンドは Linux および UNIX オペレーティング・システムでのみ使用できます。Windows オペレーティング・システムでは、同じ目的で **db2admin create** コマンドを使用できます。

**注:** DB2 Administration Server (DAS) は、バージョン 9.7 で非推奨となり、将来のリリースで除去される可能性があります。DAS は、DB2 pureScale環境ではサポートされていません。リモート管理のためには、Secure Shell プロトコルを使用するソフトウェア・プログラムを使用してください。詳しくは、『DB2 Administration Server (DAS) が推奨されなくなった』( ) を参照してください。

### 許可

root ユーザー権限

### 必要な接続

なし

### コマンド構文

```
▶▶ dasrcrt -u DASuser [-d] ▶▶
```

### コマンド・パラメーター

**-u** *DASuser*

*DASuser* は、DAS を作成するときに使用するユーザー ID です。DAS は、*/home/DASuser/das* ディレクトリーの下に作成されます。

以下の制限が適用されます。

- 既存の ID を使用して DB2 DAS を作成する場合は、ID がロックされておらず、パスワードが有効期限切れでないことを確認してください。

**-d** DB2 サービスで使用するために、デバッグ・モードに入れます。

### 使用上の注意

- AIX 6.1 (またはそれ以上) では、このコマンドは、システム・ワークロード・パーティション (WPAR) グローバル環境内の共有 DB2 コピーから実行する場合、root ユーザーとして実行されなければなりません。WPAR は DB2 pureScale 環境ではサポートされていません。

## dasdrop - DB2 Administration Server の除去

Linux および UNIX オペレーティング・システムでのみ、DB2 Administration Server (DAS) を除去します。

**重要:** バージョン 9.7 で DB2 Administration Server (DAS) が非推奨になったため、このコマンドは推奨されておらず、将来のリリースで削除される可能性があります。詳しくは、『DB2 Administration Server (DAS) が推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0059276.html>) を参照してください。

DB2 Administration Server は、リモート管理や、レプリケーション・センターなどの DB2 管理ツールをサポートするサービスを提供します。Windows オペレーティング・システムでは、同じ目的で **db2admin drop** コマンドを使用できます。

**注:** DB2 Administration Server (DAS) は、バージョン 9.7 で非推奨となり、将来のリリースで除去される可能性があります。DAS は、DB2 pureScale環境ではサポートされていません。リモート管理のためには、Secure Shell プロトコルを使用するソフトウェア・プログラムを使用してください。詳しくは、『DB2 Administration Server (DAS) が推奨されなくなった』( ) を参照してください。

### 許可

root ユーザー権限

### 必要な接続

なし

### コマンド構文

```
▶▶ dasdrop [ -d ]
```

### コマンド・パラメーター

**-d** DB2 サービスで使用するために、デバッグ・モードに入れます。

### 使用上の注意

- **dasdrop** コマンドは *DB2DIR/instance* ディレクトリーにあります。*DB2DIR* は、DB2 データベース製品の現行バージョンがインストールされている場所です。
- AIX 6.1 (またはそれ以上) では、このコマンドは、システム・ワークロード・パーティション (WPAR) グローバル環境内の共有 DB2 コピーから実行する場合、root ユーザーとして実行されなければなりません。WPAR は DB2 pureScale 環境ではサポートされていません。



## dasmigr - DB2 Administration Server のマイグレーション

Windows オペレーティング・システムの場合

**-h** 使用法情報を表示します。

**-p** *path\_override*

DAS プロファイルも同様に移動しなければならないことを示します。

*path\_override* は、デフォルトの DAS プロファイル・パスの代わりに使用されるユーザー指定パスです。

### 例

Linux および UNIX オペレーティング・システムの場合

*DB2DIR*/instance/dasmigr

Windows オペレーティング・システムの場合

*DB2DIR*\bin\dasmigr

### 使用上の注意

- AIX 6.1 (またはそれ以上) では、このコマンドは、システム・ワークロード・パーティション (WPAR) グローバル環境内の共有 DB2 コピーから実行する場合、root ユーザーとして実行されなければなりません。WPAR は DB2 pureScale 環境ではサポートされていません。

## dasupdt - DAS の更新

Linux および UNIX オペレーティング・システムで、関連した DB2 データベース・システム・インストールが更新された場合、DB2 Administration Server (DAS) を更新します。Linux、UNIX、および Windows オペレーティング・システムでは、このユーティリティを使用して、両方が DB2 データベース・システムの同じバージョン内にある場合に、DAS を 1 つのインストール・ロケーションから、別のロケーションに移動することもできます。

**重要:** バージョン 9.7 で DB2 Administration Server (DAS) が非推奨になったため、このコマンドは推奨されておらず、将来のリリースで削除される可能性があります。詳しくは、『DB2 Administration Server (DAS) が推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0059276.html>) を参照してください。

このユーティリティは Linux および UNIX オペレーティング・システムの `DB2DIR/instance` ディレクトリーにあります。`DB2DIR` は、DB2 データベース製品の現行バージョンがインストールされている場所です。Windows オペレーティング・システムでは、`dasupdt` コマンドは `DB2DIR\bin` ディレクトリーにあります。

Windows オペレーティング・システムでは、このコマンドは、同じ DB2 データベース・バージョン内の 1 つの DB2 コピーから別のコピーに DAS を更新します。DAS を以前のバージョンからアップグレードするには、`dasmigr` コマンドを使用します。`dasupdt` を使用すると、DAS は `dasupdt` コマンドの実行元の DB2 コピーに更新されます。

システム上の DAS が、`installFixPack` によって更新される DB2 データベース製品のインストール・パスに関連している場合には、フィックスパックのインストール後に、`dasupdt` コマンドは自動的に実行されます。

### 許可

Linux および UNIX オペレーティング・システム上の root ユーザー権限、または Windows オペレーティング・システム上のローカル管理者権限

### 必要な接続

なし

### コマンド構文

Linux および UNIX オペレーティング・システムの場合

```

▶▶ dasupdt [-d] [-D] [-h] [-?]

```

Windows オペレーティング・システムの場合

```

▶▶ dasupdt [-h] [-p path override]

```

## コマンド・パラメーター

### Linux および UNIX オペレーティング・システムの場合

- d デバッグ・モードを設定し、問題分析に使用します。
- D 1 つのパスのより高いコード・レベルから、別のパスにインストールされたより低いコード・レベルに DAS を移します。
- h | -? 使用法情報を表示します。

### Windows オペレーティング・システムの場合

- h 使用法情報を表示します。
- p *path\_override*  
DAS プロファイルも同様に移動しなければならないことを示します。*path\_override* は、デフォルトの DAS プロファイル・パスの代わりに使用されるユーザー指定パスです。

## 例

DAS が 1 つの DB2 データベース製品のインストール・パスで稼働しており、その DAS をレベルを下げた別のインストール・パスに (ただしこの 2 つのインストール・パスは同じバージョンの DB2 データベース・システムにある) 移動する場合、以下のコマンドをその低レベルのインストール・パスから発行します。

```
dasupdt -D
```

## 使用上の注意

- AIX 6.1 (またはそれ以上) では、このコマンドは、システム・ワークロード・パーティション (WPAR) グローバル環境内の共有 DB2 コピーから実行する場合、root ユーザーとして実行されなければなりません。WPAR は DB2 pureScale 環境ではサポートされていません。

## db2\_deinstall - DB2 データベース製品、フィーチャー、または言語のアンインストール

コマンド・パラメーターと **db2\_deinstall** コマンドの実行場所に応じて、DB2 データベース製品製品、フィーチャー、または言語をアンインストールします。このコマンドは、Linux および UNIX のオペレーティング・システムでのみ利用できません。

**db2\_deinstall** コマンドは *DB2DIR/install* にあります。*DB2DIR* は、DB2 データベース製品の現行バージョンがインストールされている場所です。**db2\_deinstall** コマンドは、DB2 データベース製品メディア上でも使用できます。**db2\_deinstall** コマンドは、インストール・パスに関連した DB2 データベース製品のアンインストールにのみ使用できます。

- 特定の DB2 データベース・インストール・パスから **db2\_deinstall -a** を実行すると、ローカル・ホストのその同じパスからすべて、または特定のフィーチャーや言語をアンインストールできます。
- DB2 データベース製品メディアから **db2\_deinstall -a** を実行する場合は、**-b** オプションを使用してパスを指定する必要があります。その後、そのローカル・ホストのそのインストール・パスからすべて、または特定のフィーチャーや言語をアンインストールできます。

### 許可

root インストールには root ユーザー権限が必要です。

### 必要な接続

なし

### コマンド構文

```

▶▶ db2_deinstall --F feature-name [-a] [-s GPFS] [-s TSAMP] [-r response-file]
▶ [-f sqllib] [-b installpath] [-l log-file] [-t trace-file]
▶ [-h] [-?]

```

### コマンド・パラメーター

**-F feature-name**

1 つのフィーチャーの除去を指定します。複数のフィーチャーのアンインストールを示すには、このパラメーターを複数回指定します。例えば、**-F feature1 -F feature2** とします。

1 つの事例を除いて、**-a** と組み合わせて使用できません。除去されるフィーチャーが IBM Tivoli System Automation for Multiplatforms (SA MP) であ



## db2\_deinstall - DB2 データベース製品、フィーチャー、または言語のアンインストール

り、ユーザーに root ユーザー権限がある場合は、**-F TSAMP** と **-a** を組み合わせて使用することで、SA MP および DB2 データベース製品の両方をまとめて除去できます。

1 つの事例を除いて、**-r** と組み合わせて使用できます。除去されるフィーチャーが IBM Tivoli System Automation for Multiplatforms (SA MP) である場合は、**-F TSAMP** と **-r** を組み合わせて使用することはできません。

DB2 データベース・アンインストーラーは、一部の DB2 フィーチャーを除去した後に、関連した DB2 インスタンスを自動的に更新します。インスタンスの更新が失敗したことがログ・ファイル中で報告された場合は、

**db2iupdt** (ルート・インスタンス) または **db2nrupdt** (非ルート・インスタンス) コマンドを使用して、関連した DB2 インスタンスを手動で更新する必要があります。

### **-f sqllib**

このパラメーターは、非 root インストールの場合のみ有効です。**-a** と併用すると、インスタンスのトップ・ディレクトリーとその下のすべてのディレクトリーが除去されます。

### **-a**

**db2\_deinstall** コマンドを IBM DB2 pureScale Feature コピーに対して実行する場合は、このパラメーターは必須です。現在場所にあるすべてのインストール済み DB2 データベース製品を除去します。1 つの事例を除いて、**-F** と組み合わせて使用できません。除去されるフィーチャーが IBM Tivoli System Automation for Multiplatforms (SA MP) であり、ユーザーに root ユーザー権限がある場合は、**-F TSAMP** と **-a** を組み合わせて使用することで、SA MP および DB2 データベース製品の両方をまとめて除去できます。

**-r** パラメーターと組み合わせることはできません。

非 root インストールでは、**-a** と **-f sqllib** を併用すると、非ルート・インスタンスも除去されます。この除去の対象には、\$HOME/sqllib ディレクトリーも含まれます。

### **-s GPFS**

IBM General Parallel File System (GPFS™) が DB2 pureScale Feature インストール済み環境の一部としてインストールされている場合には、GPFS も自動的にアンインストールされます。このパラメーターを指定すると、GPFS の削除がスキップされます。GPFS クラスタとそのファイル・システムを保持するものの、DB2 コピーを削除したい場合に、このパラメーターを使用することができます。このパラメーターは、**-a** パラメーターとのみ、組み合わせて指定できます。

### **-s TSAMP**

IBM Tivoli System Automation for Multiplatforms (SA MP) が DB2 pureScale Feature インストール済み環境の一部としてインストールされている場合は、SA MP も自動的にアンインストールされます。このパラメーターを指定すると、SA MP の削除がスキップされます。RSCT ピア・ドメインを保持するものの、DB2 コピーを削除したい場合に、このパラメータ

## db2\_deinstall - DB2 データベース製品、フィーチャー、または言語のアンインストール

ーを使用することができます。このパラメーターは、**-a** または **-F** パラメーターとのみ組み合わせることができます。

### **-r** *response-file*

応答ファイルで指定された内容に基づいて、製品、フィーチャー、または言語のアンインストールを実行します。例えば、`db2_deinstall -r db2un.rsp`。DB2 製品イメージには、すぐに使用可能なサンプル応答ファイルが含まれていて、デフォルト項目が設定されています。製品、フィーチャー、言語をアンインストールするためのサンプル応答ファイルは、`db2un.rsp` です。 `db2un.rsp` ファイルは、`DB2DIR/install` パスにあります。

**-a** パラメーターまたは **-s** パラメーターと組み合わせることはできません。

1 つの事例を除いて、**-F** パラメーターと組み合わせて使用できます。除去されるフィーチャーが IBM Tivoli System Automation for Multiplatforms (SA MP) である場合は、**-F** TSAMP と **-r** を組み合わせて使用することはできません。

**-r** パラメーターと **-F** パラメーターの両方が指定されている場合、**-F** パラメーターで指定された DB2 フィーチャーが応答ファイルの **REMOVE\_COMP** キーワードをオーバーライドします。

DB2 データベース・アンインストーラーは、一部の DB2 フィーチャーを除去した後に、関連した DB2 インスタンスを自動的に更新します。インスタンスの更新が失敗したことがログ・ファイル中で報告された場合は、**db2iupdt** (ルート・インスタンス) または **db2nrupdt** (非ルート・インスタンス) コマンドを使用して、関連した DB2 インスタンスを手動で更新する必要があります。

### **-b**

このパラメーターは、DB2 データベース製品メディアからコマンドを実行する場合に有効です。アンインストールする DB2 データベース製品がインストールされている場所の絶対パスを指定します。このパラメーターが指定されない場合、コマンドはパスを求めるプロンプトを出します。

### **-l** *log-file*

ログ・ファイルを指定します。 root インストールの場合、デフォルト・ログ・ファイルは `/tmp/db2_deinstall.log$$` です (`$$` はプロセス ID)。

非 root インストールの場合、デフォルトのログ・ファイルは `/tmp/db2_deinstall_userID.log` です (`userID` は非 root インストールを所有するユーザー ID)。除去されるフィーチャーが IBM Tivoli System Automation for Multiplatforms (SA MP) である場合、SA MP のインストール・ログ・ファイルは DB2 データベース・ログ・ファイルと同じディレクトリに入れられます。

### **-t** *trace-file*

デバッグ・モードをオンにします。デバッグ情報は、*trace-file* として指定されたファイル名に書き込まれます。

### **-h | -?**

## db2\_deinstall - DB2 データベース製品、フィーチャー、または言語のアンインストール

ヘルプ情報を表示します。

### 例

ある場所 (*DB2DIR*) にインストールされているすべての DB2 データベース製品をアンインストールするには、以下のように *DB2DIR/install* ディレクトリーで **db2\_deinstall** コマンドを発行します。

```
DB2DIR/install/db2_deinstall -a -l /tmp/db2_deinstall.log  
-t /tmp/db2_deinstall.trc
```

### 使用上の注意

- **db2\_deinstall -a** を実行した場合、DB2 データベース・インストーラーによってインストールされたコンポーネントおよびフィーチャーだけが削除されます。
- **db2\_deinstall** コマンドを実行する前に、IBM General Parallel File System (GPFS) クラスタおよびファイル・システムを手動で削除する必要があります。あるいは、**db2\_deinstall** コマンドに **-s** パラメーターを付けて実行して、GPFS バイナリー・ファイルの削除をスキップしてください。また、**db2\_deinstall** コマンドに **-s** パラメーターを付けて実行して、SA MP の削除をスキップすることもできます。

## db2\_install - DB2 データベース製品のインストール

DB2 データベース製品のすべてのフィーチャーを、指定されたパスにインストールします。このコマンドは UNIX オペレーティング・システムでのみ使用できます。

**重要:** このコマンドは推奨されなくなりました。今後のリリースで除去される可能性があります。代わりに **db2setup** コマンドを使用してください。

### 許可

root インストールには root ユーザー権限が必要です。非 root インストールの場合、非 root インストールを所有するユーザー ID でログオンしなければなりません。

### 必要な接続

なし

### コマンド構文

```

▶▶ db2_install [-b install-path] [-f NOTSAMP] [-f PURESCALE] [-f nobackup]
                [-f ignoreType1] [-p productShortName] [-c NLPACK_location]
                [-n] [-m] [-L language] [-l log-file] [-t trace-file]
                [-h]
                [-?]

```

### コマンド・パラメーター

#### **-b** *install-path*

DB2 データベース製品をインストールするパスを指定します。*install-path* は絶対パス名にする必要があり、その最大長は 128 文字に制限されます。このパラメーターは、**-n** パラメーターが指定されるときには必須です。

**-b** オプションは、DB2 データベース製品の非 root インストールの場合は必須ではありませんが、root インストールで **-n** オプションを使用する場合には依然として必須です。非 root インストールで **-b** を使用する場合、パスが有効でなければならない、ユーザーの \$HOME/sqllib になる必要があります。root インストールと非 root インストールの両方で、絶対インストール・パスの長さは 128 バイトに制限されます。

#### **-f** NOTSAMP または **-f** PURESCALE

パラメーター **-f** PURESCALE と **-f** NOTSAMP は相互排他的です。

PURESCALE オプションは、IBM DB2 pureScale Feature をサポートする製品にのみ適用されます。PURESCALE オプションは、DB2 pureScale Feature をサポートしない製品では無視されます。

**NOTSAMP**

IBM Tivoli System Automation for Multiplatforms (SA MP) をインストールまたは更新しないことを指定します。

**PURESCALE**

DB2 pureScale Feature をインストールするように指定します。

**-f nobackup**

これは非ルート・アップグレードのみに適用されます。コンポーネントの更新時に、**db2\_install** に強制的にインストール・ファイルをバックアップさせないようにします。ファイルをバックアップしないことを選択すると、インストール・ディレクトリーのスペース所要量が削減されます。ただし、ファイルをバックアップしないことを選択すると、エラーが発生した場合は、DB2 データベース・インストーラーがロールバック操作を実行できなくなります。この場合、ファイルを手動でクリーンアップし、製品を再インストールする必要があります。

**-f ignoreType1**

これは非ルート・アップグレードのみに適用されます。データベース状況の検査中に、**db2\_install** に強制的にタイプ 1 索引を無視させます。

**-p productShortName**

インストールする DB2 データベース製品を指定します。このパラメーターは大/小文字を区別しません。また、**-n** パラメーターが指定される際には必須です。製品の短縮名 (*productShortName*) は、メディア上の *db2/plat* (*plat* はインストール先のプラットフォーム名) サブディレクトリーに入っているファイル *ComponentList.htm* 中 (製品のフルネームの下) にあります。一度にインストールできる製品は 1 つのみです。

**-c NLPACK\_location**

関連した DB2 各国語パック (NLPACK) の絶対パスの場所を指定します。このパラメーターは、**-n** が指定されるときには必須です。以下の条件がすべて満たされている場合は、DB2 NLPACK の場所を明示的に指定する必要があります。

- **-n** オプションが指定されている。
- インストールには、各国語 (英語以外) サポートが必要。
- DB2 NLPACK が、DB2 データベース製品イメージ内になく、DB2 データベース製品イメージと同じサブディレクトリー中にもない。

**-n** 非対話式モードを指定します。指定する場合は、**-b**、**-p**、**-c** のいずれかまたはすべても指定しなければなりません。

**-m** このオプションは、非 root インストールにのみ適用されます。非ルート・コピーのアップグレードを指定します。アップグレード中に、現行パスにある既存のすべての DB2 データベース製品は除去されます。アップグレードにより、指定された製品がインストールされます。アップグレードに続いて、その他の DB2 データベース製品を別個にインストールする必要があります。

**-L language**

各国語サポートを指定します。DB2 データベース製品の英語以外のバージョンをインストールできます。しかし、このコマンドは、各国語パック CD からではなく、製品 CD から実行する必要があります。デフォルトでは、

## db2\_install - DB2 データベース製品のインストール

常に英語がインストールされます。そのため、英語は指定する必要がありません。複数の言語が必要な場合、このパラメーターは必須です。複数の言語を示すには、このパラメーターを複数回指定します。例えば、フランス語とドイツ語の両方をインストールするには、`-L FR -L DE` と指定します。このパラメーターには、大/小文字の区別がありません。

### **-l** *log-file*

ログ・ファイルを指定します。root インストールの場合、デフォルト・ログ・ファイルは `/tmp/db2_install.log$$` です (`$$` はプロセス ID)。非 root インストールの場合、デフォルトのログ・ファイルは `/tmp/db2_install_userID.log` です (`userID` は非 root インストールを所有するユーザー ID)。**db2\_install** コマンドを使用して IBM Tivoli System Automation for Multiplatforms (SA MP) をインストールまたは更新する場合、対応するログ・ファイルは、DB2 データベース・ログ・ファイルと同じディレクトリーに入れられます。

### **-t** *trace-file*

デバッグ・モードをオンにします。デバッグ情報は、*trace-file* として指定されたファイル名に書き込まれます。

### **-h | -?**

使用法情報を表示します。

## 例

- `/mnt/cdrom` 内のイメージからインストールし、すべての必要な入力のためのプロンプトが出されるようにするか、または DB2 Enterprise Server Edition を `/mnt/cdrom` 内のイメージからインストールするには、以下のコマンドを発行します。

```
cd /mnt/cdrom
./db2_install
```

- DB2 Enterprise Server Edition を `/mnt/cdrom` 内のイメージから `/db2/newlevel` に英語で非対話式にインストールするには、以下のコマンドを発行します。

```
cd /mnt/cdrom
./db2_install -p ese -b /db2/newlevel -n
```

- DB2 Enterprise Server Edition を DB2 pureScale Featureと共に `/mnt/cdrom` 内のイメージから `/db2/newlevel` に、英語で非対話式にインストールするには、以下のコマンドを発行します。

```
cd /mnt/cdrom
./db2_install -p ese_dsf -b /db2/newlevel -n
```

## 使用上の注意

IBM PowerHA<sup>®</sup> SystemMirror for AIX クラスターが実行中の場合、IBM Tivoli System Automation for Multiplatforms (SA MP) のインストール、アップグレード、更新は実行できません。SA MP には、PowerHA SystemMirror に依存する Reliable Scalable Cluster Technology (RSCT) ファイル・セットが組み込まれているためです。SA MP インストールをスキップするには、**db2\_install** コマンドか **installFixPack** コマンドを使用します。PowerHA SystemMirror クラスターを使用した SA MP のインストールまたはアップグレードについては、「Upgrade guide for DB2 Servers in HACMP™ Environments」というホワイト・ペーパーを参照してください。この資料は、IBM サポートおよびダウンロード Web サイト

## db2\_install - DB2 データベース製品のインストール

(<http://www.ibm.com/support/docview.wss?uid=swg21045033>) から入手できます。



## db2\_local\_ps - Linux/UNIX の DB2 プロセス状況

Linux および UNIX システムで `db2_local_ps` コマンドを使用すると、インスタンスで実行されているすべての DB2 プロセスを表示できます。

### 許可

なし

### 必要な接続

なし

### コマンド構文

▶▶—db2\_local\_ps—▶▶

### コマンド・パラメーター

#### db2\_local\_ps

インスタンスで実行されているすべての DB2 プロセスを出力します。

### 例

```
[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 db2lfr (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
```



## db2\_local\_ps - Linux/UNIX の DB2 プロセス状況

```
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 ...

### 使用上の注意

インスタンスが停止すると、プロセスは表示されません。プロセスがリスト表示されない場合には、**db2start** コマンドを実行します。

## db2acsutil - DB2 スナップショット・バックアップ・オブジェクトの管理

DB2 スナップショット・バックアップ・オブジェクトをリスト、削除、およびモニターします。

**db2acsutil** は、以下の 3 種類の方法で DB2 スナップショット・バックアップ・オブジェクトを管理する場合に使用します。

- データベースをリストアするために使用できる DB2 スナップショット・バックアップをリストする
- **BACKUP** コマンド、db2Backup API、または **BACKUP DATABASE** パラメーターを指定した **ADMIN\_CMD** ストアド・プロシージャを使用して生成された DB2 スナップショット・バックアップを削除する
- DB2 スナップショット・バックアップの状況をモニターする

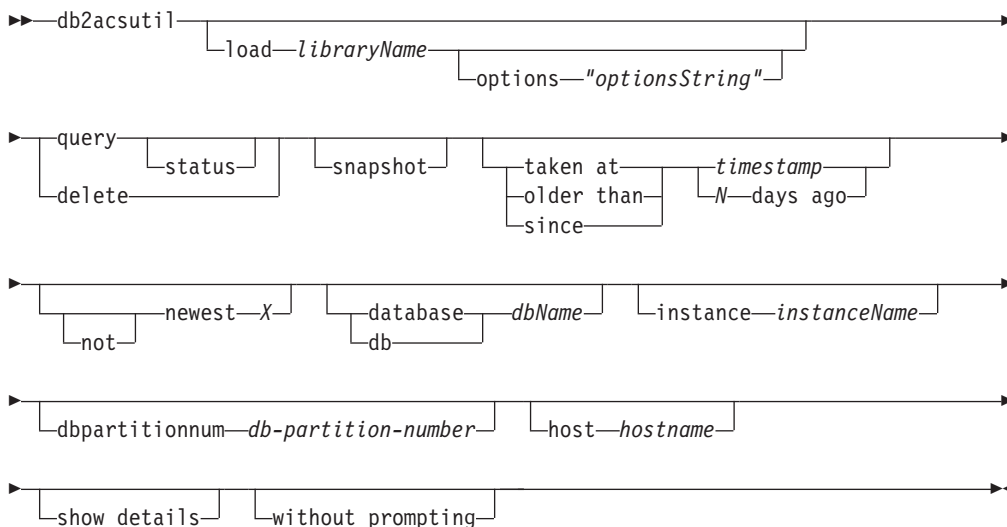
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### load libraryName

DB2 スナップショット・バックアップの中で使用されるベンダー高速コピー・テクノロジーを含む共有ライブラリーの名前。このパラメーターには、絶対パスを含めることができます。絶対パスが指定されていない場合のデフォルト・パスは、**BACKUP DB** および **RESTORE DB** コマンドの場合と同じライブラリーです (~/sqllib/acs の中)。

### **options** *"optionsString"*

このユーティリティーで使用するオプションを指定します。このストリングは、入力されたとおりの (ただし二重引用符を除いた) ベンダー・サポートのライブラリーに渡されます。

### **query**

ACS リポジトリを照会し、既知のオブジェクトの表を戻します。

### **status**

ACS リポジトリを照会し、既知のオブジェクトとその現在の状況の表を戻します。

### **delete**

これは、DB2 スナップショット・オブジェクトを削除します。また、その削除後に ACS リポジトリからそのレコードを除去します。

### **snapshot**

戻されるレコードまたは操作の対象となるレコードをフィルターに掛けて、スナップショット・オブジェクトだけになるようにします。

### **taken at | older than | since**

これらのオプションは、ユーティリティーの結果をフィルターに掛けて、指定された時刻範囲のものだけになるようにします。

### *timestamp*

形式 *YYYYMMDDhhmmss* のタイム・スタンプ。

### *N days ago*

何日前かの指定。現在の日付の *N* 日前を指定します。

### **[not] newest X**

ユーティリティーの結果をフィルターに掛けて、タイム・スタンプを基準として最も新しい *X* 個のレコードだけを考慮するようにします。 **NOT** キーワードを指定した場合は、最も新しい *X* 個以外のすべてのレコードが考慮の対象になります。

### **database | db** *dbName*

指定したデータベース名に関連したオブジェクトだけを対象にします。

### **instance** *instanceName*

管理作業の対象となる DB2 スナップショット・バックアップ・オブジェクトに関連するデータベース・マネージャー・インスタンスの名前。

### **dbpartitionnum** *db-partition-number*

指定したデータベース・パーティション番号で作成されたオブジェクトだけを対象にします。

### **host** *hostname*

指定した *hostname* によって作成されたオブジェクトだけを対象にします。例えば、多くの場合、DB2 サーバーの TCP/IP ホスト名を指定します。

### **show details**

ACS リポジトリに含まれている詳細オブジェクト情報を表示します。このオプションを使用する場合、1 つの簡潔なレコードを 1 行とする表ではなく、1 つの ACS オブジェクトごとに詳細なスタンプが生成されます。

**without prompting**

ユーティリティを無人実行し、通常ならユーザー介入を必要とするすべてのアクションでエラー・メッセージを戻すようにすることを指定します。

**例**

アクティブ・バックグラウンド・コピーを含むスナップショット・バックアップの出力例。

```
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)
```

完了したバックグラウンド・コピーを伴うスナップショット・バックアップの出力例。

```
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
```

**使用上の注意**

db2acsutil は、**BACKUP** コマンド、db2Backup API、または **BACKUP DATABASE** パラメーターを指定した **ADMIN\_CMD** ストアード・プロシージャーを使用して作成された DB2 スナップショット・バックアップを削除するための唯一の手段です。DB2 スナップショット・バックアップを削除するのに、自動リカバリー・オブジェクト削除または **AND DELETE** パラメーターを指定した **PRUNE HISTORY** コマンドを使用することはできません。フィルター/ストレージ・システムを使用して手動でバックアップを削除することもできません。

DB2 スナップショット・バックアップのユーザビリティ状態は、DB2 スナップショットに対してどんな操作を実行できるかを示します。表 1 は、DB2 スナップショット・バックアップのユーザビリティ状態として可能性のあるものとその説明のリストです。

表 48. DB2 スナップショット・バックアップについて戻されるユーザビリティ状態

ユーザビリティ状態	説明
LOCALLY_MOUNTABLE	バックアップ・データをローカル・マシンからマウントできます。

表 48. DB2 スナップショット・バックアップについて戻されるユーザビリティ状態 (続き)

ユーザビリティ状態	説明
REMOTELY_MOUNTABLE	バックアップ・データをリモート・マシンからマウントできます。
REPETITIVELY_RESTORABLE	バックアップ・データをリストアするために、DB2 スナップショット・バックアップ・イメージを複数回使用できます。
DESTRUCTIVELY_RESTORABLE	バックアップ・データをリストアするために、DB2 スナップショット・バックアップ・イメージを一度だけ使用できます。バックアップ・データがリストアされた後、その DB2 スナップショット・イメージ (および場合によってはそれ以外のもの) は破棄されます。
SWAP_RESTORABLE	ボリュームに直接アクセスができますが、 <b>RESTORE DB</b> コマンドは実行できず、バックアップ・データをソース・ボリュームに再びコピーすることはできません。
PHYSICAL_PROTECTION	スナップショットは、ソース・ボリュームにおいて物理的失敗から保護されています。
FULL_COPY	データの完全コピーが作成されました。DB2 スナップショット・バックアップ・イメージを使用して、バックアップ・データをリストアすることができます。
DELETED	バックアップに削除マークが付けられたことを示します。DELETED バックアップに関連するスナップショット・ストレージは、バックグラウンドで実行されているメンテナンス・プロセスの実行を通じて取り消されます。それが完了すると、バックアップは ACS リポジトリから除去されます。
FORCED_MOUNT	AIX JFS ファイル・システムをマウントすることにより、ファイル・システム整合性の検査を待機中。
BACKGROUND_MONITOR_PENDING	ACS バックグラウンド進捗モニターにより状況がモニターされています。
TAPE_BACKUP_PENDING	オフロード・テープ・バックアップを待機中。
TAPE_BACKUP_IN_PROGRESS	オフロード・テープ・バックアップが現在進行中。
TAPE_BACKUP_COMPLETE	オフロード・テープ・バックアップが完了しました。

## db2addicons - DB2 ツールのメインメニュー項目の作成

DB2 ツールのメインメニュー項目を作成します。

Linux オペレーティング・システムで、**db2addicons** コマンドは、現在のユーザーに関する DB2 ツールのメインメニュー項目を作成します。**db2addicons** コマンドを手動で実行すると、DB2 ツールのメインメニュー項目が作成されます。DB2 インスタンス所有者の場合、DB2 インスタンスの作成、更新、またはアップグレード時に、メニュー項目はインスタンス・ユーティリティーによって自動的に作成されます。別のユーザーのデスクトップ上でメインメニュー項目が必要な場合は、その特定のユーザーとして **db2addicons** コマンドを実行できますが、コマンドの実行前にまずそのユーザーの環境内でインスタンス環境を設定しなければなりません。

### 許可

なし

### コマンド構文

```
▶▶ db2addicons [-h] ▶▶
```

### コマンド・パラメーター

**-h**      使用法情報を表示します。

## db2admin - DB2 Administration Server

このユーティリティーは、DB2 Administration Server (DAS) の管理に使用します。パラメーターを指定せず、かつ DAS が存在する場合、このコマンドは DAS の名前を戻します。

**重要:** バージョン 9.7 で DB2 Administration Server (DAS) が非推奨になったため、このコマンドは推奨されておらず、将来のリリースで削除される可能性があります。詳しくは、『DB2 Administration Server (DAS) が推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0059276.html>) を参照してください。

Linux および UNIX オペレーティング・システムの場合、**db2admin** コマンドの実行可能ファイルは *DASHOME*/*das*/*bin* ディレクトリーにあります (*DASHOME* は DAS ユーザーのホーム・ディレクトリー)。Windows オペレーティング・システムの場合、**db2admin** の実行可能ファイルは *DB2PATH*\*bin* ディレクトリーにあります (*DB2PATH* は DB2 コピーのインストール場所)。

### 許可

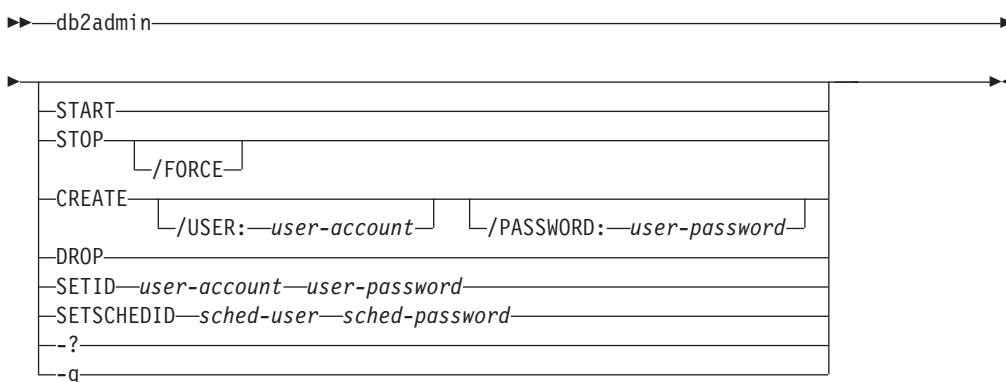
UNIX オペレーティング・システムでは DASADM (ただし 64 ビット・インスタンスには関連付けられていないもの)。

Windows オペレーティング・システムではローカル管理者。

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### START

DAS を開始します。

#### STOP /FORCE

DAS を停止します。force オプションを使用した場合、要求のサービスを処理中であるかどうかに関係なく、強制的に DAS を停止します。

**CREATE /USER:** *user-account /PASSWORD:* *user-password*

DAS を作成します。ユーザー名およびパスワードを指定した場合、DAS はこのユーザー・アカウントに関連付けられます。指定した値が無効であると、ユーティリティーは認証エラーを戻します。指定するユーザー・アカウントは、有効な SQL ID でなければならず、セキュリティ・データベース内に存在していなければなりません。DAS のすべての機能にアクセスできるようにするために、ユーザー・アカウントを指定することをお勧めします。UNIX オペレーティング・システム上に DAS を作成するには、**dasCRT** コマンドを使用します。

**DROP** DAS を削除します。UNIX オペレーティング・システム上の DAS をドロップするには、**dasdrop** コマンドを使用する必要があります。

**SETID** *user-account/user-password*

DAS に関連付けられたユーザー・アカウントを設定または変更します。

**SETSCHEDID** *sched-user/sched-password*

ツール・カタログ・データベースに接続するためにスケジューラーで使用するログオン・アカウントを確立します。これは、スケジューラーが有効になっている場合で、ツール・カタログ・データベースが DAS から見てリモートである場合にのみ必要です。スケジューラーについての詳細は、「管理ガイド」を参照してください。

**-?** ヘルプ情報を表示します。このオプションを指定すると、他のすべてのオプションは無視され、ヘルプ情報だけが表示されます。

**-q** **db2admin** コマンドを静止モードで実行します。コマンドが実行されるときにもメッセージは表示されません。このオプションは、他のすべてのコマンド・オプションと組み合わせて使用できます。



## db2adutl - TSM 内の DB2 オブジェクトの管理

Tivoli Storage Manager (TSM) を使用して保存した、バックアップ・イメージ、ログ、およびロード・コピー・イメージの、照会、抽出、検査、および削除をユーザーに許可します。また、ユーザーが TSM サーバー上のオブジェクトへのアクセスを付与したり取り消したりできるようにします。

UNIX オペレーティング・システムでは、このユーティリティーは `sqllib/adsm` ディレクトリーにあります。Windows オペレーティング・システムでは、これは `sqllib¥bin` にあります。

### 許可

なし

### 必要な接続

なし

### コマンド構文

```
▶▶ db2adutl [db2-object-options] [access-control-options]
```

#### db2-object-options:

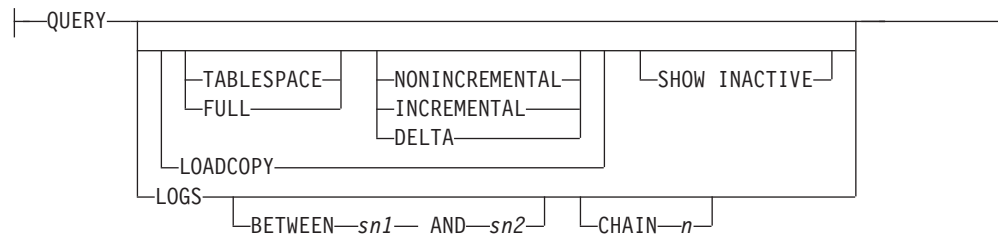
```

[QUERY-options] [EXTRACT-options] [UPLOAD-options] [DELETE-options] [VERIFY-options]
[COMPRLIB—decompression-library]
[COMPROPTS—decompression-options] [VERBOSE]
[DATABASE—database_name] [DBPARTITIONNUM—db-partition-number]
[DB] [LOGSTREAM—log-stream-number]
[OPTIONS—tsm_options] [PASSWORD—password] [NODENAME—node_name]
[OWNER—owner] [WITHOUT PROMPTING]

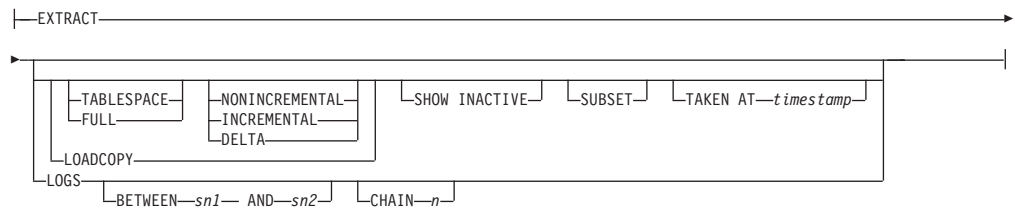
```

## db2adutl - TSM 内の DB2 オブジェクトの管理

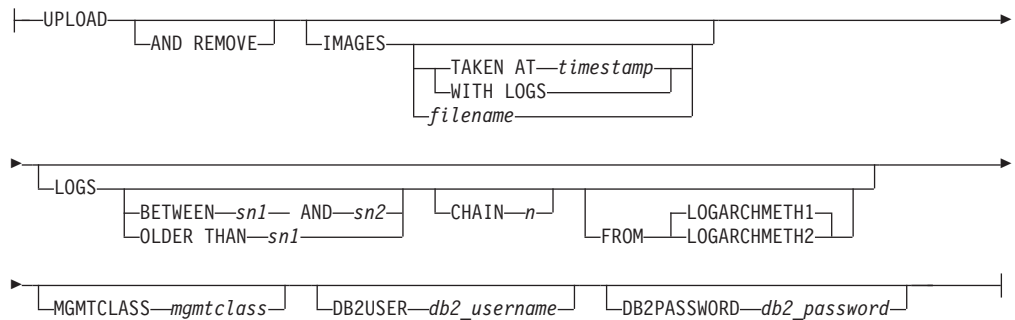
### QUERY-options:



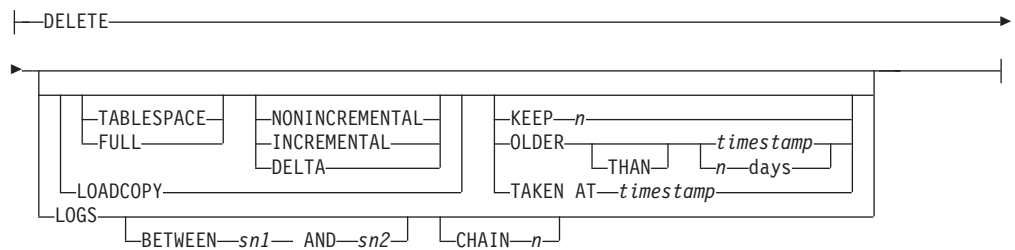
### EXTRACT-options:



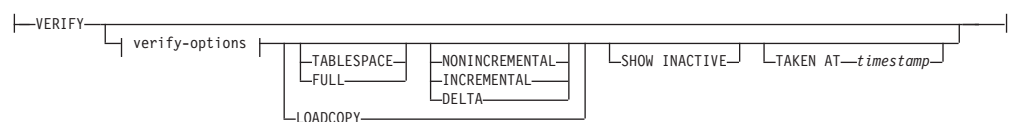
### UPLOAD-options:



### DELETE-options:



### VERIFY-options:



**verify-options:**

ALL
CHECK
CHECKDMS
DMS
HEADER
LFH
TABLESPACES
SGF
HEADERONLY
TABLESPACESONLY
SGFONLY
OBJECT
PAGECOUNT

**access-control-options:**

GRANT	ALL	ON	ALL	FOR	DATABASE	database_name
	USER—user_name		NODENAME—node_name		DB	
REVOKE	ALL	ON	ALL	FOR	ALL	database_name
	USER—user_name		NODENAME—node_name		DATABASE	
					DB	
QUERYACCESS	FOR	ALL			database_name	
		DATABASE				
		DB				
PASSWORD	password					

**コマンド・パラメーター**

**QUERY** TSM サーバーで DB2 オブジェクトを照会します。

**EXTRACT**

TSM サーバーからの DB2 オブジェクトを、ローカル・マシンにある現行ディレクトリーにコピーします。

**UPLOAD** ディスクに格納されているバックアップ・イメージまたはアーカイブ・ログを TSM サーバーにアップロードします。このオプションを使用するときには、データベース名を指定する必要があります。

**DELETE** バックアップ・オブジェクトを非活動化するか、または TSM サーバーにあるログ・アーカイブを削除します。

**VERIFY** サーバー上のバックアップ・コピーに対して整合性検査を実行します。このパラメーターを指定すると、バックアップ・イメージ全体がネットワークを介して転送されます。

**ALL** 使用できるすべての情報を表示します。

**CHECK** チェックビットおよびチェックサムの結果を表示します。

**CHECKDMS**

**DMS** および AS 表スペース・ページの追加の妥当性検査を実行します。このオプションは、暗黙指定されません。また、**ALL** オプションでは使用可能にはなりません。

**DMS** DMS 表スペース・データ・ページのヘッダーからの情報を表示します。

**HEADER** メディア・ヘッダー情報を表示します。

**HEADERONLY**

HEADER と同じ情報を表示します。ただし、イメージの先頭から 4 K メディア・ヘッダー情報のみを読み取ります。イメージの妥当性検査は実行しません。

**LFH** ログ・ファイル・ヘッダー (LFH) データを表示します。

**OBJECT** オブジェクト・ヘッダーからの詳細情報を表示します。

**PAGECOUNT**

イメージの中にある各オブジェクト・タイプのページ数を表示します。

**SGF** イメージ内の自動ストレージ・パスを表示します。

**SGFONLY**

イメージ内の自動ストレージ・パスを表示するだけで、イメージの妥当性検査は実行しません。

**TABLESPACES**

コンテナ情報など、イメージ中の表スペースに関する詳細情報を表示します。

**TABLESPACESONLY**

**TABLESPACES** と同じ情報を表示しますが、イメージの妥当性検査は実行しません。

**TABLESPACE**

表スペース・バックアップ・イメージだけを組み込みます。

**FULL** 完全データベース・バックアップ・イメージだけを組み込みます。

**NONINCREMENTAL**

非増分バックアップ・イメージだけを組み込みます。

**INCREMENTAL**

増分バックアップ・イメージだけを組み込みます。

**DELTA** 増分差分バックアップ・イメージだけを組み込みます。

**LOADCOPY**

ロード・コピー・イメージだけを組み込みます。

**LOGS** ログ・アーカイブ・イメージだけを組み込みます。

**BETWEEN sn1 AND sn2**

ログ・シーケンス番号 1 とログ・シーケンス番号 2 の間のログの使用を指定します。

**CHAIN n**

使用するログのチェーン ID を指定します。

**SHOW INACTIVE**

非活動化されているバックアップ・オブジェクトを組み込みます。

**SUBSET** イメージからファイルにページを抽出します。ページを抽出するには、入力ファイルと出力ファイルが必要です。デフォルトの入力ファイルは extractPage.in という名前です。デフォルトの入力ファイル名は、

**DB2LISTFILE** 環境変数に絶対パスを設定することによりオーバーライドできます。入力ファイルの形式は、次のとおりです。

SMS 表スペースの場合、

```
S <tbspID> <objID> <objType> <startPage> <numPages>
```

注:

1. <startPage> は、オブジェクト相対のオブジェクト・ページ番号です。

DMS 表スペースの場合、

```
D <tbspID> <objType> <startPage> <numPages>
```

注:

1. <objType> が必要なのは、DMS ロード・コピー・イメージを検証する場合だけです。
2. <startPage> は、プール相対のオブジェクト・ページ番号です。

ログ・ファイルの場合、

```
L <log num> <startPos> <numPages>
```

その他のデータ (初期データなど) の場合、

```
O <objType> <startPos> <numBytes>
```

デフォルトの出力ファイルは `extractPage.out` です。デフォルトの出力ファイル名は、**DB2EXTRACTFILE** 環境変数に絶対パスを設定することによりオーバーライドできます。

#### **TAKEN AT** *timestamp*

TSM にアップロードするバックアップ・イメージのタイム・スタンプを指定します。

**KEEP** *n* タイム・スタンプで最新の *n* 個を除き、指定したタイプのすべてのオブジェクトを非活動化します。

#### **OLDER THAN** *timestamp or n days*

*timestamp* または *n* 日より前のタイム・スタンプが付けられているオブジェクトを非活動化することを指定します。

#### **AND REMOVE**

バックアップ・イメージとログ・ファイルを TSM に正常にアップロードした後、それらを削除するように指定します。

**IMAGES** TSM にアップロードするバックアップ・イメージを指定します。イメージのファイル名を指定しても、**db2adutl** は履歴・ファイルの照会を引き続き試行します。対応する項目が履歴・ファイルで見つかると、**db2adutl** は指定のファイル名が履歴・ファイル内の場所と一致する場合にのみ、イメージをアップロードします。対応する項目が見つからない場合には、指定のパスから直接イメージをアップロードし、完了時に履歴・ファイル更新は実行されません。**IMAGES** を **UPLOAD** オプションと一緒に指定する場合、データベース名を指定する必要があります。

#### **WITH LOGS**

バックアップ・イメージと一緒に使用するアーカイブ・ログを指定します。

### *filename*

イメージ・ファイルの名前を指定します。このオプションを指定しない場合、データベース名を指定する必要があります。

### **LOGARCHMETH1** または **LOGARCHMETH2**

アップロードするログ・ファイルのアーカイブ場所を指定します。**LOGARCHMETH1** がデフォルトです。

### **MGMTCLASS** *mgmtclass*

アップロードを実行する TSM 管理クラスを指定します。

### **DB2USER** *db2\_username*

リカバリー履歴ファイルを更新するために作成する必要がある DB2 接続で使用するユーザー ID を指定します。

### **DB2PASSWORD** *db2\_password*

リカバリー履歴ファイルを更新するために作成する必要がある DB2 接続で使用するユーザー ID のパスワードを指定します。

### **COMPRLIB** *decompression-library*

解凍を実行するために使用するライブラリーの名前。この名前は、サーバー上の 1 個のファイルを参照する完全修飾パスでなければなりません。このパラメーターを指定しない場合、DB2 はイメージ内に格納されているライブラリーの使用を試みます。バックアップが圧縮されていなかった場合、このパラメーターの値は無視されます。指定されたライブラリーをロードできない場合、操作は失敗します。

### **COMPROPTS** *decompression-options*

バイナリー・データのうち、解凍ライブラリーの初期設定ルーチンに渡すブロックを記述します。DB2 はこのストリングをクライアントからサーバーに直接渡すため、バイト反転やコード・ページ変換の問題がある場合は解凍ライブラリーで処理する必要があります。データ・ブロックの最初の文字が「@」である場合、DB2 は、データの残りの部分をサーバー上に存在するファイルの名前と解釈します。その場合 DB2 は、データ・ブロックの内容をそのファイルの内容で置き換え、そのようにして得られる新しい値を初期設定ルーチンに渡します。このストリングの最大長は 1024 バイトです。

### **DATABASE** *database\_name*

指定したデータベース名に関連したオブジェクトだけを対象にします。

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

指定したデータベース・パーティション番号で作成されたオブジェクトだけを対象にします。

### **LOGSTREAM** *log-stream-number*

指定したログ・ストリーム番号で作成されたオブジェクトだけを対象にします。

### **OPTIONS** “*tsm\_options*”

TSM セッションの初期化時に TSM サーバーに渡すオプションを指定します。**OPTIONS** は、二重引用符なしで、入力されたとおりに TSM サーバーに渡されます。**OPTIONS** パラメーターを使用した場合、**db2adutl** コマンドは TSM サーバーが生成したエラーを返します。

**PASSWORD** *password*

このノードの TSM クライアント・パスワードを指定します (必要な場合)。データベースが指定されたもののパスワードが提供されない場合には、**tsm\_password** データベース構成パラメーターに指定した値が TSM に渡されます。渡されない場合には、パスワードは使用されません。

**NODENAME** *node\_name*

特定の TSM ノード名に関連したイメージだけを対象にします。

**重要: NODENAME** パラメーターと **-asnodename** 値が指定された **OPTIONS** パラメーターは互換性がないため、これらを同時に使用することはできません。プロキシ・ノード構成をサポートする TSM 環境の場合は **OPTIONS** **"-asnodename"** パラメーターを使用し、その他のタイプの TSM 構成の場合は **NODENAME** パラメーターを使用するようにしてください。

**OWNER** *owner*

指定した所有者により作成されたオブジェクトだけを対象にします。

**重要: OWNER** パラメーターと **-asnodename** 値が指定された **OPTIONS** パラメーターは互換性がないため、これらを同時に使用することはできません。プロキシ・ノード構成をサポートする TSM 環境の場合は **OPTIONS** **"-asnodename"** パラメーターを使用し、その他のタイプの TSM 構成の場合は **OWNER** パラメーターを使用するようにしてください。

**WITHOUT PROMPTING**

オブジェクトの削除の前に、確認を求めるプロンプトが出ないようにします。

**VERBOSE**

付加的なファイル情報を表示します。

**GRANT ALL | USER** *user\_name*

現在の TSM ノード上の TSM ファイルに対するアクセス権を、すべてのユーザーまたは指定したユーザーに付与します。アクセス権をユーザーに付与すると、指定されたデータベースに関連する現在のファイルと将来のファイルのすべてのアクセス権を付与することになります。

**REVOKE ALL | USER** *user\_name*

現在の TSM ノード上の TSM ファイルに対するアクセス権を、すべてのユーザーまたは指定したユーザーから削除します。

**QUERYACCESS**

現在のアクセス・リストを取り出します。ユーザーと TSM ノードのリストが表示されます。

**ON ALL | NODENAME** *node\_name*

アクセス権を変更する TSM ノードを指定します。

**FOR ALL | DATABASE** *database\_name*

対象となるデータベースを指定します。

**例**

1. 以下は、コマンド `db2 backup database rawsamp1 use tsm` からのサンプル出力です。

## db2adutl - TSM 内の DB2 オブジェクトの管理

Backup successful. The timestamp for this backup is : 20031209184503

以下に示すのは、バックアップ操作の後で発行された db2adutl query コマンドの出力例です。

```
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, Log stream: 0,
  Taken at 2003-12-09-18.46.13
  Log file: S0000051.LOG, Chain Num: 0, Log stream: 0,
  Taken at 2003-12-09-18.46.43
  Log file: S0000052.LOG, Chain Num: 0, Log stream: 0,
  Taken at 2003-12-09-18.47.12
  Log file: S0000053.LOG, Chain Num: 0, Log stream: 0,
  Taken at 2003-12-09-18.50.14
  Log file: S0000054.LOG, Chain Num: 0, Log stream: 0,
  Taken at 2003-12-09-18.50.56
  Log file: S0000055.LOG, Chain Num: 0, Log stream: 0,
  Taken at 2003-12-09-18.52.39
```

2. 以下に示すのは、db2adutl delete full taken at 20031209184503 db rawsampl コマンドの出力例です。

```
Query for database RAWSAMPL

Retrieving FULL DATABASE BACKUP information.
  Taken at: 20031209184503 Log stream: 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
```

以下に示すのは、全バックアップ・イメージを削除した操作の後で発行された db2adutl query コマンドの出力例です。バックアップ・イメージのタイム・スタンプに注意してください。



Query for database RAWAMPL

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 RAWAMPL

Retrieving DELTA DATABASE BACKUP information.

No DELTA DATABASE BACKUP images found for RAWAMPL

Retrieving TABLESPACE BACKUP information.

No TABLESPACE BACKUP images found for RAWAMPL

Retrieving INCREMENTAL TABLESPACE BACKUP information.

No INCREMENTAL TABLESPACE BACKUP images found for RAWAMPL

Retrieving DELTA TABLESPACE BACKUP information.

No DELTA TABLESPACE BACKUP images found for RAWAMPL

Retrieving LOCAL COPY information.

No LOCAL COPY images found for RAWAMPL

Retrieving log archive information.

Log file: S0000050.LOG, Chain Num: 0, Log stream: 0,

Taken at 2003-12-09-18.46.13

Log file: S0000051.LOG, Chain Num: 0, Log stream: 0,

Taken at 2003-12-09-18.46.43

Log file: S0000052.LOG, Chain Num: 0, Log stream: 0,

Taken at 2003-12-09-18.47.12

Log file: S0000053.LOG, Chain Num: 0, Log stream: 0,

Taken at 2003-12-09-18.50.14

Log file: S0000054.LOG, Chain Num: 0, Log stream: 0,

Taken at 2003-12-09-18.50.56

Log file: S0000055.LOG, Chain Num: 0, Log stream: 0,

Taken at 2003-12-09-18.52.39

3. 以下に示すのは、db2adutl queryaccess for all コマンドの出力例です。

Node	User	Database Name	type
bar2	jchisan	sample	B
<all>	<all>	test	B

Access Types: B - Backup images L - Logs A - both

4. 以下に示すのは、DB2 pureScale 環境における 3 つのメンバーのバックアップ・イメージから表示される出力例です。

BufAddr	MemberNum	PoolID	Token	Type	Offset	FileSize	ObjectSize	OrigSize	Object Name
00000000:	0	0	0	19	0	268	268	0	"BACKUP.START. RECORD.MARKER"

numTbpsInDB : 3  
numTbpsInImg : 3

Total members : 3  
Member numbers: 0,1,2

## 使用上の注意

以下の各グループから 1 つのパラメーターを使用して、操作に組み込むバックアップ・イメージのタイプを制限できます。

### 細分:

- FULL - データベース・バックアップ・イメージだけを組み込みます。

## db2adutl - TSM 内の DB2 オブジェクトの管理

- TABLESPACE - 表スペースのバックアップ・イメージだけを組み込みます。

### 累積性:

- NONINCREMENTAL - 非増分バックアップ・イメージだけを組み込みます。
- INCREMENTAL - 増分バックアップ・イメージだけを組み込みます。
- DELTA - 増分差分バックアップ・イメージだけを組み込みます。

TSM 環境でプロキシ・ノードを使用する場合、プロキシ・ノードの使用時に取られたバックアップ・イメージまたはログ・アーカイブを表示するためには、

**OPTIONS** パラメーターで `asnodename` オプションを使用して共有 TSM プロキシ・ノード値を指定する必要があります (例えば、**OPTIONS**

`"-asnodename=cluster1"`)。 **OPTIONS** パラメーターは、バージョン 9.8 フィックスパック 3 以降のフィックスパックで使用可能です。

TSM は、オブジェクトの所有者または `root` ユーザーに削除権限を付与します。他のユーザーへの削除権限は制限する可能性があります。

それぞれのログ・ファイル名の形式は次のとおりです。

`S0*****.LOG`

バージョン 9.8 フィックスパック 3 より前では、TSM サーバー上のログ・ファイルは `./NODE0***/TESTLOG/C0*****/` ディレクトリーに書き込まれていました。バージョン 9.8 フィックスパック 3 以降のフィックスパックでは、TSM サーバー上のログ・ファイルは `./NODE****/LOGSTREAM****/C0*****/` ディレクトリーに書き込まれます。

## db2advis - DB2 設計アドバイザー

DB2 設計アドバイザーは、マテリアライズ照会表 (MQT) と索引の作成、表の再パーティション化、マルチディメンション・クラスター化 (MDC) 表への変換、未使用オブジェクトの削除に関して、ユーザーにアドバイスを提示します。

推奨事項は、ユーザーが指定する 1 つ以上の SQL ステートメントに基づきます。関連 SQL ステートメントのグループは、ワークロードと呼ばれます。ユーザーは、ワークロード内の各ステートメントの重要性をランク付けし、ワークロード内の各ステートメントが実行される頻度を指定することができます。設計アドバイザーは、推奨オブジェクトを作成するための CREATE INDEX、CREATE SUMMARY TABLE (MQT)、CREATE TABLE の各ステートメントを組み込んだ DDL CLP スクリプトを出力します。

構造化タイプ列は、このコマンドの実行時には考慮されません。

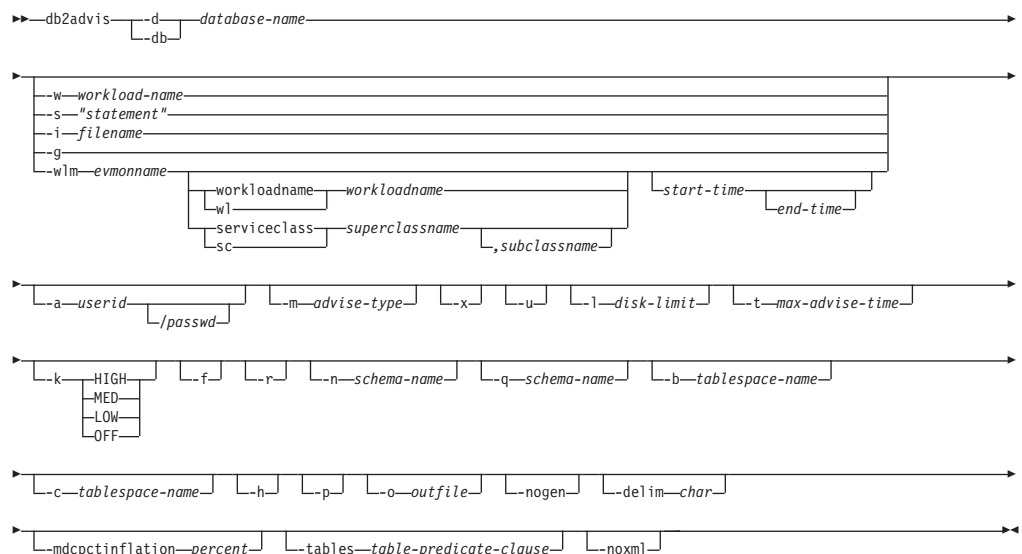
### 許可

データベースへの読み取りアクセス。 Explain 表への読み取りおよび書き込みアクセス。マテリアライズ照会表 (MQT) を使用する場合、CREATE TABLE 許可、および MQT に対する読み取り/書き込みアクセスが必要です。

### 必要な接続

なし。このコマンドは、データベース接続を確立します。

### コマンド構文



### コマンド・パラメーター

**-d database-name**

接続の確立先のデータベースの名前を指定します。

**-w workload-name**

設計アドバイザーによって評価され、索引を提案されるワークロードの名前

を指定します。この名前は ADVISE\_WORKLOAD 表で使用されます。このオプションは、**-g**、**-i**、または **-s** の各オプションと一緒に指定できません。

**-s "statement"**

設計アドバイザーによって評価され、索引を提案される単一 SQL ステートメントのテキストを指定します。ステートメントは必ず二重引用符で囲んでください。このオプションは、**-g**、**-i**、**-w** の各オプションと一緒に指定できません。

**-i filename**

1 つ以上の SQL ステートメントが入っている入力ファイルの名前を指定します。デフォルトは標準入力です。注釈テキストは、各行の先頭に 2 つのハイフンを付けて `-- comment` で表します。ステートメントは必ずセミicolonで区切ってください。

ワークロード中の各ステートメントが実行される頻度は、次の行を入力ファイルに挿入することによって変更できます。

```
--#SET FREQUENCY x
```

頻度は、ファイル中何回でも更新できます。このオプションは、**-g**、**-s**、**-w** の各オプションと一緒に指定できません。

**-g** 動的 SQL スナップショットからの SQL ステートメントの検索を指定します。**-p** コマンド・パラメーターと結合されている場合は、SQL ステートメントは ADVISE\_WORKLOAD 表に保持されます。このオプションは、**-i**、**-s**、**-w** の各オプションと一緒に指定できません。

**-wlm evmonname**

ACTIVITY および ACTIVITYSTMT 論理データ・グループに対応するイベント名 *evmonname* の表名を SYSCAT.EVENTTABLES から取得するように指定し、PARTIAL\_RECORD = 0 (トランザクション完了) のあるレコードの ACTIVATE\_TIMESTAMP、ACTIVITY\_ID、および ACTIVITY\_SECONDARY\_ID 上で一緒に結合します。オプションの *start-time* および *end-time* タイム・スタンプを追加し、*start-time* の時点かその後ステートメントを取得し、さらにオプションで *end-time* の時点かその前にも取得できます。*start-time* および *end-time* は、ACTIVITY 表中の TIME\_COMPLETED 列に関連付けられます。

**workloadname | wl workloadname**

SYSCAT.WORKLOADS 中で検索する *workloadname* を指定します。ACTIVITY イベント・モニター表が SYSCAT.WORKLOADS とワークロード ID 上で結合され、これらのステートメントが入手されます。

**serviceclass | sc superclassname**

SYSCAT.SERVICECLASSES 中のサービス・クラス情報を指定します。サブクラスを指定しないと、サービス・スーパークラスに関するステートメントがすべて取り出されます。基本的にサービス・スーパークラスは SYSCAT.SERVICECLASSES 中の PARENTSERVICECLASS です。ACTIVITY イベント・モニター表が SYSCAT.SERVICECLASSES とサービス・クラス ID 上で結合され、これらのステートメントが入手されます。

*,subclassname*

*superclassname* を指定する場合に *subclassname* を指定します。コンマで区切ります。このパラメーターはオプションです。

*start-time*

開始タイム・スタンプを指定します。

*end-time*

終了タイム・スタンプを指定します。このパラメーターはオプションです。

**-a** *userid/passwd*

データベースへの接続に使用する名前およびパスワード。パスワードが指定される場合、斜線 (/) を含めなければなりません。 **-x** オプションを指定した場合は、パスワードを指定しません。

**-m** *advise-type*

アドバイザーが戻す推奨のタイプを指定します。 I、M、C、P を任意に組み合わせて指定できます (大/小文字のどちらも可能)。例えば、 **db2advis -m PC** は、パーティション化と MDC 表を推奨します。パーティション・データベース環境で **-m P** または **-m M** を使用した場合、 `advise_partition` 表には最終的なパーティション推奨事項が入ります。選択可能な値は以下のとおりです。

**I** 新規索引を推奨します。これはデフォルトです。

**M** 新規マテリアライズ照会表 (MQT) および MQT の索引を推奨します。パーティション・データベース環境では、MQT のパーティション化も推奨します。

**C** 標準表からマルチディメンション・クラスター化 (MDC) 表に変換するよとの推奨、または表のクラスタリング索引を作成するよとの推奨。

**P** 既存表の再パーティション化を推奨します。

**-x** 端末からパスワードを読み取るか、ユーザーがパスワードを入力するかを指定します。

**-u** アドバイザーが据え置き MQT の推奨を考慮することを指定します。増分 MQT は推奨されません。このオプションを指定した場合は、DDL CLP スクリプト内のコメントによって、どの MQT を即時更新 MQT (Refresh Immediate MQT) に変換できるかが示されます。パーティション・データベース環境で即時更新 MQT (Refresh Immediate MQT) が推奨される場合は、デフォルトの分散キーが MQT の暗黙のユニーク・キーになります。

**-l** *disk-limit*

既存のスキーマ内のすべての推奨された索引とマテリアライズ・ビューに使用できるメガバイト数を指定します。可能な最大サイズを使用する場合は、 **-1** を指定します。デフォルト値は、データベースの合計サイズの 20% です。

**-t** *max-advise-time*

操作を完了するための最大許可時間 (分) を指定します。このオプションに

値を指定しない場合、操作は完了するまで継続します。無制限の時間を指定する場合は、ゼロの値を入力します。デフォルトはゼロです。

- k ワークロードを圧縮する度合いを指定します。圧縮を行うと、アドバイザーは実行処理を簡略化しますが、ワークロード全体の場合と同様の結果を生成します。HIGH の場合、アドバイザーはワークロードの小さなサブセットを対象にします。MED の場合、アドバイザーはワークロードの中程度のサブセットを対象にします。LOW の場合、アドバイザーはワークロードの大きなサブセットを対象にします。OFF の場合、圧縮は行われず、すべての照会が考慮されます。デフォルトは MED です。
- f 以前存在していたシミュレーション・カタログ表をドロップします。
- r 仮想 MQT とパーティション化の選択に詳細な統計を使用することを指定します。このオプションを指定しない場合は、デフォルトで MQT にオプティマイザー統計が使用されます。詳細な統計のほうが正確ですが、これを得るには時間がかかるので、**db2advis** の実行時間が長くなります。-r コマンド・パラメーターは MQT およびパーティション化に関連した統計を取得するためにサンプリングを使用します。MQT に関しては、サンプル照会が失敗するか行を 1 つも戻さない場合には、オプティマイザー見積もりが使用されます。

### -n *schema-name*

シミュレーション・カタログ表の修飾名、および新しい索引と MQT の修飾子を指定します。デフォルトのスキーマ名は、呼び出し側のユーザー ID です。ただし、カタログ・シミュレーション表の場合は例外で、デフォルトのスキーマ名は SYSTOOLS になります。デフォルトでは、新しい索引が索引のベースのスキーマ名を継承します。

### -q *schema-name*

ワークロード内の非修飾名の修飾名を指定します。これは、**db2advis** の実行時に CURRENT SCHEMA に使用するスキーマ名になります。デフォルトのスキーマ名は、コマンドの実行者のユーザー ID です。

### -b *tablespace-name*

新規 MQT を作成する表スペースの名前を指定します。指定しない場合、アドバイザーは、存在する表スペースのセットから表スペースを選択します。

### -c *tablespace-name*

シミュレーション・カタログ表を作成する表スペースの名前を指定します。表スペースは任意のタイプが可能で、例えばファイル名やディレクトリーを使用できます。この表スペースは、カタログ・データベース・パーティション・グループでのみ作成されなければなりません。デフォルトは USERSPACE1 です。

デフォルトの USERSPACE1 を使用する代わりに、シミュレーションに使用する表スペースを作成することをお勧めします。さらに、**db2advis** ユーティリティーのパフォーマンスを改善するために、ALTER TABLESPACE DROPPED TABLE RECOVERY OFF ステートメントをこの表スペースに対して実行してください。このユーティリティーが完了すると、表スペースの

履歴が元に戻ります。パーティション・データベース環境では、通常、USERSPACE1 がすべてのパーティション・グループにまたがって作成されるため、このオプションは必須です。

- h** ヘルプ情報を表示します。このオプションを指定すると、他のすべてのオプションは無視され、ヘルプ情報だけが表示されます。
- p** Explain 表でツールを実行した際に生成されたプランを維持します。**-p** コマンド・パラメーターを使用すると、**-g** のワークロードが ADVISE\_WORKLOAD 表に保存され、最終推奨を使用するワークロード照会のプランが Explain 表に保存されます。
- o outfile**  
推奨されたオブジェクトを作成するためのスクリプトを *outfile* に保存します。
- nogen** 生成列をマルチディメンション・クラスタリング推奨に含めないことを指定します。
- delim char**  
ワークロード・ファイル入力の中のステートメント区切り文字 *char* を示します。デフォルトは ; です。
- mdcptinflation percent**  
表のディスク・サイズを MDC 推奨で増やすことができるパーセントの最大値を指定します。例えば、これで、表を MDC 表に変換する場合に、その元のサイズの  $1+percent/100$  倍まで増やすことを許可することを指定します。*percent* は、デフォルト値が 10 の浮動小数点数です。
- tables table-predicate-clause**  
すべての既存の表のサブセットだけを考慮することを指示します。  
*table-predicate-clause* は、SYSCAT.TABLES に対するクエリーの WHERE 節で使用できる述部でなければなりません。db2advis によって考慮される表は、このクエリーからの表とワークロード内の表の論理積になります。  
このコマンド・パラメーターは、新しい MQT に関する推奨事項には当てはまりません。
- noxml** 推奨事項テキストの後に詳細な XML 出力をコンソールに書き込まないように指定します。

## 例

1. 次の例では、ユーティリティーはデータベース PROTOTYPE に接続し、解決策において制限なしで表 ADDRESSES に索引を推奨します。  

```
db2advis -d prototype -s "select * from addresses a
  where a.zip in ('93213', '98567', '93412')
  and (company like 'IBM%' or company like '%otus')"
```
2. 次の例では、ユーティリティーは PROTOTYPE データベースに接続し、ADVISE\_WORKLOAD 表の照会用として 53 MB を超えない索引を推奨します。ワークロード名は "production" と同じです。解決策を見つけるための最大許可時間は 20 分です。  

```
db2advis -d prototype -w production -l 53 -t 20
```
3. 次の例では、db2advis.in という入力ファイルに SQL ステートメント、および各ステートメントが実行される頻度の指定が含まれています。



```
--#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;
```

ユーティリティーはデータベース SAMPLE に接続し、入力ファイル内の照会によって参照される各表ごとに索引を推奨します。解決策を見つけるための最大許可時間は 5 分です。

```
db2advis -d sample -i db2advis.in -t 5
```

4. 次の例では、表スペース SPACE1 内に MQT を作成します。シミュレーション表スペースは SPACE2 です。ワークロード内の非修飾名の修飾名は SCHEMA1、新規 MQT の推奨を行うスキーマ名は SCHEMA2 です。使用するワークロード圧縮は HIGH、ディスク・スペースは無制限です。MQT にはサンプル統計を使用します。以下のコマンドを実行すると、MQT の推奨が行われ、パーティション・データベース環境では、索引とパーティション化の推奨も行われます。

```
db2advis -d prototype -w production -l -l -m M -b space1 -c space2 -k
HIGH -q schema1 -n schema2 -r
```

推奨された MQT と、MQT と基本表の両方の索引、パーティション化、および MDC を取得するには、**-m** オプションの値 IMCP を指定して以下のようにコマンドを発行します。

```
db2advis -d prototype -w production -l -l -m IMCP -b space1 -c space2 -k
HIGH -q schema1 -n schema2 -r
```

5. 次の例では、ユーティリティーはデータベース SAMPLE に接続し、EMPLOYEE と DEPT 用の表に MDC の候補がそれらの元のサイズの 30.5% 増加することが許可される MDC を推奨します。

```
db2advis -d sample -type C -disklimit 100 -i db2advis.in
-tables "TABNAME IN ('EMPLOYEE','DEPT')" -mdcpctinflation 30.5
```

## 使用上の注意

DDL CLP スクリプト、データベースのパーティション化、マルチディメンション・クラスタリング、クラスター索引を実行するには、これらのフィーチャーを事前にセットアップする必要があるため、推奨内容は、戻される DDL CLP スクリプト内でコメント化されます。表を推奨される DDL へと変換する操作はユーザーが行います。例えば ALTER TABLE ストアード・プロシージャを使ってこれを行うことができますが、RENAME ステートメントの場合と同様の制約事項があります。

バージョン 9.7 以降、設計アドバイザーではパーティション索引は推奨されていません。すべての索引で、NOT PARTITIONED 節を使用することが推奨されています。この推奨事項の場合、索引を作成するために PARTITIONED (デフォルト) または NOT PARTITIONED を使用するかどうかは、アプリケーション・シナリオに基づいて、およびパーティション索引によって得られる利点に基づいて選択します。

動的 SQL ステートメントの場合、ステートメントが実行される頻度は、次のようにモニターから獲得されます。

1. 次のコマンドを発行する。



```
db2 reset monitor for database database-alias
```

適切な時間間隔にわたって待機します。

2. 次のコマンドを発行する。

```
db2advis -g other-options
```

**-p** パラメーターが **-g** パラメーターと共に使用されている場合、取得される動的 SQL ステートメントは ADVISE\_WORKLOAD 表に、タイム・スタンプを含む生成されたワークロード名と共に置かれます。

ワークロードの各 SQL ステートメントのデフォルト頻度は 1 で、デフォルトの重要度も 1 です。generate\_unique() 関数は、その SQL ステートメントのより分かりやすい説明になるように、ユーザーによって更新できるステートメントに、固有 ID を割り当てます。

すべての **db2advis** エラー情報は **db2diag** ログ・ファイルにも入っています。

アドバイザーの実行が始まると、アドバイザーを識別する行が ADVISE\_INSTANCE 表に含まれるようになります。メイン・アドバイザー行は、アドバイザーの開始時間を示す START\_TIME によって識別されます。この行の STATUS は「STARTED」です。

**db2advis** コマンドを発行した結果、「Cannot insert into DB2ADVIS\_INSTANCE」というエラーが発生した場合には、db2advis.bnd をバインドし、**-1** オプションを使って **db2advis** コマンドを実行する必要があります。バインド操作を実行するには、以下のコマンドを発行します。

```
db2 bind db2advis.bnd blocking all grant public
```

アドバイザーが完了すると、ADVISE\_INSTANCE 表内で適切な START\_TIME を持つ関連行を検査することができます。STATUS が「COMPLETED」であれば、アドバイザーは正常に実行されました。STATUS がまだ「STARTED」で、実行中の **db2advis** プロセスが存在しない場合は、アドバイザーが異常終了しました。STATUS に「EX」が含まれる場合、アドバイザーの失敗原因を判別するための「SQLCODE」も表示されます。

**-1 disk-limit** オプションを指定しない場合、GET\_DBSIZE\_INFO ストアード・プロシージャを使ってデータベース最大サイズを判別するためには、SYSADM、SYSCTRL、SYSMAINT、または SYSMON 権限のうち少なくとも 1 つが必要です。

**-tables** パラメーターの *table-predicate-clause* は、SYSCAT.TABLES の照会とアドバイザーが考慮する表の判別に使用されます。基本表または既存の MQT しか考慮できませんが、*table-predicate-clause* では、基本表名または MQT のリストを返すために別名と論理ビューを使用することができます。例えば、「TV」で始まるビューを持つ表のサブセットを指定するには、**-tables "(tablename, tabschema) in (SELECT bname, bschema FROM SYSCAT.TABDEP WHERE TABNAME LIKE 'TV%')"** と指定します。

バージョン 9.7 では、照会オプティマイザーは節約できる入出力コスト、およびコスト・モデルにおけるキー値と RID の圧縮解除コストを測定します。そのようにして、索引アドバイザーは圧縮索引サイズの見積もりを行うことができます。

## db2audit - 監査機能管理者用ツール

DB2 データベース・システムには、未知または予期しないデータ・アクセスの検出を支援する監査機能が備わっています。DB2 監査機能は、事前定義された一連のデータベース・イベントの監査証跡を生成し、その保守を許可します。

この機能で生成されたレコードは、監査ログ・ファイルに保持されます。これらのレコードを分析すると、システムの誤用を識別する使用パターンが明らかになります。識別することができれば、システムのそのような誤用を削減または除去する処置をとることができます。監査機能はインスタンス・レベルとデータベース・レベルの両方で動作し、インスタンスまたはデータベースに基づいてすべてのアクティビティを別々のログに個別に記録します。

DB2 データベース・システムには、インスタンス・レベルおよび個々のデータベース・レベルで別個に監査する機能があります。インスタンス・レベルの監査を構成して、そのような監査情報をいつ収集するかを制御するには、**db2audit** ツールを使用します。個々のデータベースの監査要件を構成して制御するには、AUDIT SQL ステートメントを使用します。**db2audit** ツールを使用して、インスタンスおよびデータベースの監査ログをアーカイブしたり、いずれかの種類のアーカイブ・ログから抽出したりすることができます。

パーティション・データベース環境で作業している場合、監査可能イベントの多くは、ユーザーが接続しているデータベース・パーティション (コーディネーター・パーティション) か、またはカタログ・パーティション (それらが同じデータベース・パーティションではない場合) で発生します。したがって、監査レコードが複数のデータベース・パーティションで生成される場合があるということになります。各監査レコードの一部には、コーディネーター・パーティションおよび発信元データベース・パーティションの ID に関する情報が含まれています。

インスタンスの監査ログ (`db2audit.instance.log.node_number[.timestamp]`) はインスタンスの `security/auditdata` サブディレクトリーにあり、監査構成ファイル (`db2audit.cfg`) はインスタンスの `security` サブディレクトリーにあります。データベースの監査ログは、`db2audit.db.dbname.log.node_number[.timestamp]` という名前になります。インスタンスの作成時点では、オペレーティング・システムにより、それらのファイルに対して可能な限り読み取り/書き込み権限が設定されています。デフォルトでは、その権限はインスタンスの所有者にとってのみの読み取り/書き込み権限です。それらの権限は変更しないようにしてください。

監査機能の許可ユーザーは、**db2audit** を使用することにより、監査機能内で以下の処置を制御することができます。

- DB2 インスタンス内で監査可能イベントの記録を開始する。データベース・レベルの活動は、これに含まれません。
- DB2 インスタンス内で監査可能イベントの記録を停止する。
- 監査機能の振る舞いを構成する (インスタンス・レベルのみ)。
- 記録する監査可能イベントのカテゴリを選択する (インスタンス・レベルのみ)。
- インスタンスに関する現在の監査構成の説明を要求する。

- ペンディング中の監査レコードをインスタンスからフラッシュし、監査ログに書き込む。
- インスタンス、またはインスタンスの下のデータベースに関する現在の監査ログからアーカイブ監査レコードを生成する。
- 監査レコードを書式設定してフラット・ファイルまたは ASCII 区切りファイルにコピーすることにより、監査レコードをアーカイブ監査ログから抽出する。抽出は、ログ・レコード分析の準備として実行されます。

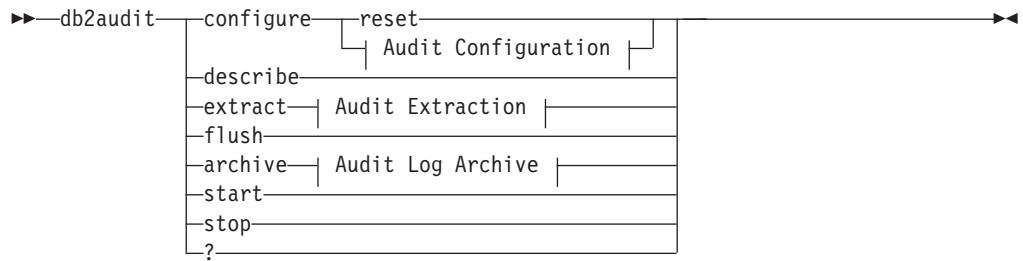
**許可**

SYSADM

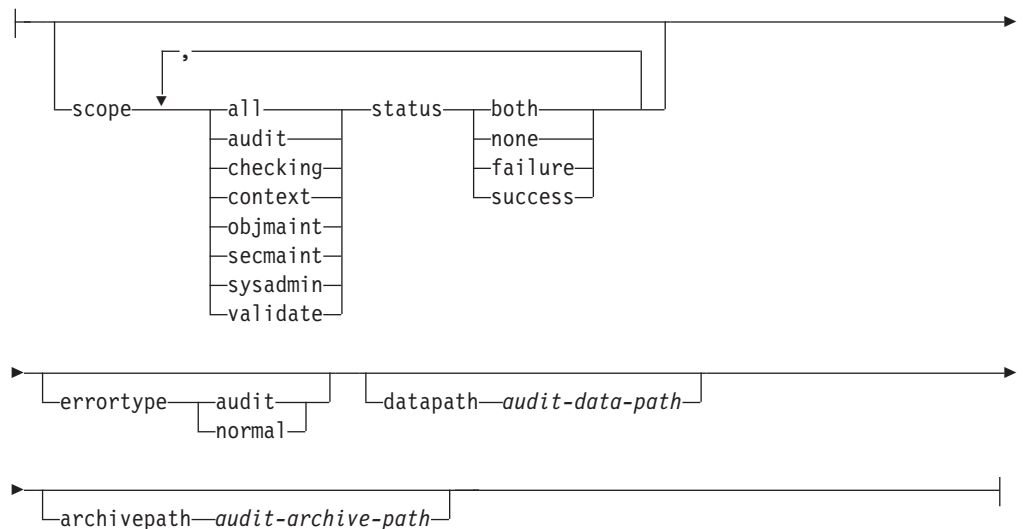
**必要な接続**

なし

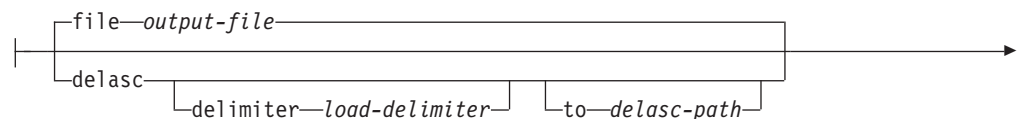
**コマンド構文**

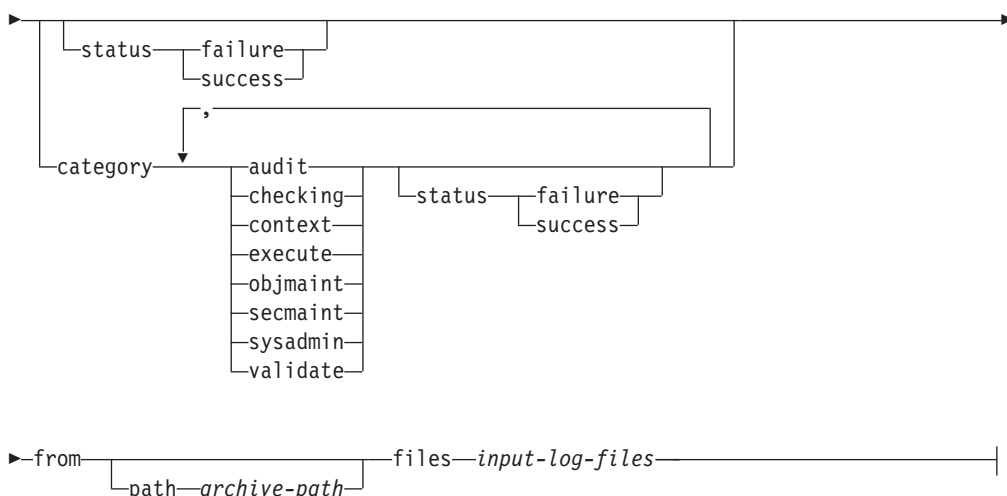


**Audit Configuration:**

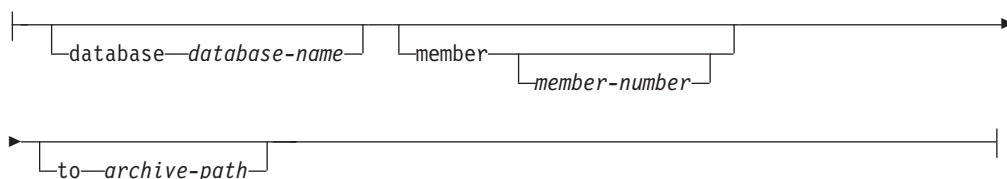


**Audit Extraction:**





### Audit Log Archive:



### コマンド・パラメーター

#### configure

このパラメーターを使用すると、インスタンスの `security` サブディレクトリーにある構成ファイル `db2audit.cfg` を変更できます。このファイルに対する更新は、インスタンスが停止されている場合でも発生することがあります。インスタンスがアクティブである場合に発生する更新は、DB2 インスタンスによって実行される監査に動的に影響を与えます。監査機能が開始されていて、監査可能イベントの **audit** (監査) カテゴリーが監査されている場合、構成ファイルに対して **configure** (構成) アクションが実行されると、監査レコードが作成されます。(データ・パスとアーカイブ・パスを除く) すべての構成オプションは、データベース・レベルの監査イベントではなく、インスタンス・レベルの監査イベントにのみ適用されます。パスのオプションは、インスタンスおよびインスタンス内のすべてのデータベースに適用されます。

構成ファイルに対して可能なアクションは、以下のとおりです。

**reset** このアクションが実行されると、構成ファイルが初期構成に戻りません (**scope** は `context` を除く全カテゴリー、各カテゴリーの **status** は `failure`、**errortype** は `normal`、インスタンス・レベルのイベントの監査は `off`)。オリジナルの監査構成ファイルが失われたか壊れている場合にこのアクションが実行されると、監査構成ファイルが新たに作成されます。監査データ・パスとアーカイブ・パ

スはブランクになります。このオプションは、監査ポリシー、およびデータベース・レベルでのこれらのポリシーの使用をリセットしません。

**scope** このアクションは、どのカテゴリーを監査するか、および対象となる各カテゴリーの状況を指定します。

**status** このアクションは、ログ記録の対象が成功したイベントだけなのか、失敗したイベントだけなのか、それとも成功したイベントと失敗したイベントの両方なのかを指定します。

**status** には次のオプションがあります。

**both** 成功したイベントと失敗したイベントがどちらも監査されます。

なし このカテゴリーのイベントは監査されません。

**failure**

失敗したイベントだけが監査されます。

**success**

成功したイベントだけが監査されます。

**configure** ステートメントで指定されたカテゴリーだけが変更されます。他のすべてのカテゴリーの状況は保持されます。

#### 注:

- デフォルトの **scope** は **context** を除く全カテゴリーであり、その場合には短時間で大量のレコードが生成されてしまう可能性があります。カテゴリーの選択は、モード (同期か非同期か) と共にパフォーマンス低下の大きな原因となる可能性があります。ディスク要件がどれだけ増大するかを大きく左右することになります。ログ記録の対象となるイベントの数と種類を可能な限り限定することをお勧めします。そうしないと、監査ログのサイズがすぐに肥大化してしまいます。また、このアクションを実行することにより、監査を特定の対象に絞ることができ、ログの肥大化を抑えることができます。
- **context** イベントは、操作の状況が認識されるより前に発生します。したがって、そのようなイベントは、**status** が **none** でない限り、このパラメーターに関連付けられた値に関係なくログに記録されます。
- 同じカテゴリーが繰り返された場合、または **all** キーワードによってカテゴリーが重複して指定されている場合には、構文エラーが戻されます。

#### errortype

このアクションは、監査エラーがユーザーに戻されるのか、それとも無視されるのかを指定します。このパラメーターの値として可能なのは、以下のとおりです。

**audit** 監査機能内で発生するエラーも含めて、すべてのエラーが DB2 データベースによって管理され、負の SQLCODE はすべて呼び出し元に報告されます。

**normal db2audit** によって生成されるエラーは無視され、実行されている操作に関連したエラーの SQLCODE だけがアプリケーションに戻されます。

**datapath** *audit-data-path*

これは、DB2 データベース・システムによって生成された監査ログが書き込まれるディレクトリーです。デフォルトは `sqllib/security/auditdata` (Windows では `インスタンス・パス¥instance¥security¥auditdata`) です。このパラメーターは、データベース・レベルの監査を含め、インスタンス内のすべての監査に影響を与えます。これは相対パスではなく、絶対パスでなければなりません。インスタンス所有者は、このディレクトリーに対する書き込み権限を持っている必要があります。Windows では、**db2start**、**db2audit**、**db2 update dbm cfg** などのローカル・インスタンス・コマンドを発行するユーザーは、そのコマンドに対して監査が行われる必要がある場合に、このディレクトリーに対する書き込み権限を取得している必要があります。パーティション・データベース環境では、このディレクトリーとして NFS 共有ディレクトリーを指定できますが、そうする必要があるわけではありません。非共有ディレクトリーを指定した場合、各メンバーが固有のディスクに書き込むことになるため、パフォーマンスが向上します。パスの最大長は、UNIX または Linux では 971 バイト、Windows オペレーティング・システムでは 208 バイトです。

パスを "" と指定した場合、パスはデフォルトに更新されます。

**db2audit describe** には設定されるパスが表示されず、デフォルト・パスが使用されます。なお、シェルによって引用符が解釈されるのを防ぐために、ほとんどの場合、例えば

```
db2audit configure datapath ¥"¥"
```

のように引用符をエスケープする必要があります。

データ・パスは必須です。パーティション・データベース環境では、各メンバーに同じデータ・パスが使用されます。特定のメンバーに対して固有のデータ・パスのセットを指定することはできません。ただし、データ・パス名にデータベース・パーティション式を含める場合はこれが可能です。その使用によって、処理結果のパス名が各メンバーごとに異なるように、メンバー番号をストレージ・パスにおいて反映することができます。

**archivepath** *audit-archive-path*

これは、アーカイブ・オプション用および抽出オプション用のデフォルト・ディレクトリーです。パーティション・データベース環境では、すべてのメンバーからアクセス可能な NFS 共有ディレクトリーをこのディレクトリーとして指定することをお勧めします。デフォルトは `sqllib/security/auditdata` (Windows では `sqllib¥instance¥security¥auditdata`) です。これは相対パスではなく、絶対パスでなければなりません。インスタンス所有者は、このディレクトリーに対する書き込み権限を持っている必要があります。パスの最大長は、UNIX または Linux では 971 バイト、Windows オペレーティング・システムでは 208 バイトです。

アーカイブ・パスは必須です。アーカイブ・パスではデータベース・パーティション式を使用できません。



**describe**

このパラメーターは、現在のインスタンス・レベルの監査の構成情報と状況を標準出力に表示します。

以下の項目が表示されます。

- 監査がアクティブかどうか。
- 各カテゴリーの状況。
- エラー・タイプ (エラーに対する SQLCA が戻されるかどうかという形式)。
- データ・パスとアーカイブ・パス。

**describe** の出力は、例えば次のようになります。

DB2 AUDIT SETTINGS:

```

監査がアクティブ: "FALSE"
監査イベントのログ: "SUCCESS"
チェック・イベントのログ: "FAILURE"
オブジェクト保守イベントのログ: "BOTH"
セキュリティ保守イベントのログ: "BOTH "
システム管理者イベントのログ: "NONE"
妥当性検査イベントのログ: "FAILURE"
コンテキスト・イベントのログ: "NONE"
監査エラーで SQLCA を戻す: "TRUE"
Audit Data Path: "/auditdata"
Audit Archive Path: "/auditarchive"

```

AUD0000I 操作が成功しました。

**extract**

このパラメーターを使用すると、監査レコードを監査ログから指定された宛先に移動することができます。監査ログは、データベースのコード・ページで作成されます。抽出の実行時に、すべてのフィールドは現在のアプリケーション・コード・ページに変換されます。

抽出時に使用できるオプションは、以下のとおりです。

**file output-file**

抽出された監査レコードが *output-file* に格納されます。ディレクトリーが指定されていない場合、*output-file* は現行作業ディレクトリーに書き込まれます。ファイルが既に存在する場合、出力はそれに付加されます。ファイル名が指定されない場合、レコードは、監査構成ファイルで指定されたアーカイブ・パスの db2audit.out ファイルに書き込まれます。

**delasc**

抽出された監査レコードは、区切り ASCII フォーマットになります。これは、DB2 データベースのリレーショナル表にロードするのに適しています。出力はカテゴリーごとに 1 つずつ、別個のファイルに入れます。さらに、監査データに含まれる lob を保持するための auditlobs というファイルも作成されます。ファイル名は次のとおりです。

- audit.del
- checking.del
- objmaint.del

- secmaint.del
- sysadmin.del
- validate.del
- context.del
- execute.del
- auditlobs

ファイルが既に存在する場合、出力はそれに付加されます。 `auditlobs` ファイルは、**context** または **execute** カテゴリが抽出される場合に作成されます。 `auditlobs` ファイル内の LOB を参照するために、LOB ロケーション指定子が `.del` ファイルに格納されます。

### **delimiter** *load-delimiter*

監査ログからの抽出時に、デフォルトの監査文字ストリング区切り (二重引用符 ") をオーバーライドできます。 監査レコードが入る表の中にロードするための準備段階で使用する新しい区切り文字を、**delimiter** の後に指定できます。新しいロード区切り文字として、単一の文字 (例えば !)、または 16 進数表記の 4 文字ストリング (例えば 0x3b) が可能です。

### **to** *delasc-path*

区切りファイルの書き込み場所のパスを指定できます。これが指定されない場合、監査構成ファイル内の監査アーカイブ・パス・オプションで指定されたディレクトリーにファイルが書き込まれます。

### **category**

監査イベントのうち指定されたカテゴリの監査レコードが抽出されます。これが指定されていない場合、すべてのカテゴリが抽出対象になります。

### **status**

指定された状況の監査レコードが抽出されます。これが指定されていない場合、すべてレコードが抽出対象になります。

### **path**

アーカイブ監査ログの場所を示すパス。これが指定されない場合、監査構成の中のアーカイブ・パスが使用されます。ファイル名に絶対パスが含まれる場合には、このパスは使用されません。

### **files**

抽出対象の監査ログ・ファイルのリスト。これには、単一のファイル、または複数ファイルからなるリストを指定できます。これらのファイルは、抽出時に変更されません。ファイル名が完全修飾されていない場合、ファイル名は **path** と結合して完全修飾ファイル名として認識されます。リストでは、標準的なシェル・ワイルドカードを使って複数のファイルを指定できます。

**flush** このパラメーターを使用すると、保留中の監査レコードが強制的に監査ログに書き込まれます。さらに、監査機能がエラー状態になっている場合には、監査状態が「ログ記録不可能」から「ログ記録可能」状態にリセットされず。



**archive**

このパラメーターは、個々のデータベース用またはインスタンス用の現在の監査ログを、アーカイブ用および抽出用の新しい場所に移動します。ファイル名には現在のタイム・スタンプが付加されます。監査ログに現在書き込まれている途中のすべてのレコードは、レコード全体が分割されないように、ログのアーカイブ前に完了します。アーカイブ処理中に作成されたすべてのレコードは、アーカイブが完了した後、アーカイブ・ログではなく現在の監査ログに書き込まれます。

アーカイブ時に使用できるオプションは、以下のとおりです。

**database database-name**

監査ログをアーカイブする対象のデータベースの名前。データベース名が提供されない場合、インスタンス・レベルの監査ログがアーカイブされます。

**member**

現在のメンバーに対してのみアーカイブ・コマンドを実行すること、および現在のメンバーはモニター・エレメント **node\_number** によって表されることを指示します。

注: 現行の *member-number* の使用は、DB2 pureScale 環境およびパーティション・データベース環境でオプションです。db2audit archive .... node コマンドが渡され、**DB2NODE** が設定されていれば、そのノード値が使用されます。**DB2NODE** が設定されていなければ、0 が使用されます。

**member-number**

現在、実行対象となっているメンバーを、db2audit 実行可能プログラムに示します。

注: 現行の *member-number* の使用は、DB2 pureScale 環境およびパーティション・データベース環境でオプションです。db2audit archive .... node X コマンドが渡された場合、**DB2NODE** が設定されているかどうかにかかわらず、ノード値 (X) が使用されます。

**to archive-path**

アーカイブ監査ログの作成場所となるディレクトリー。このディレクトリーは既に存在しなければならず、インスタンス所有者はこのディレクトリーに対する作成権限を持っていないければなりません。これが提供されない場合、監査構成の中のアーカイブ・パスが使用されます。

作成されるファイル名は、次のような形式になります。

- インスタンス・ログの場合は  
db2audit.instance.log.member\_number[.YYYYMMDDHHMMSS]
- データベース・ログの場合は  
db2audit.db.dbname.log.member\_number[.YYYYMMDDHHMMSS]

ここで YYYY は年、MM は月、DD は日、HH は時間、MM は分、SS は秒です。時刻は現地時間になります。インスタンス監査ログにはデータベース名の部分がありません。非パーティション・データベース環境でのメンバー番号は 0 になります。ファイルが既に存在する場合には、それに付加されます。

このタイム・スタンプは、ログ内の最後のレコードを 100% 正確に表すわけではありません。このタイム・スタンプは、アーカイブ・コマンドが実行された時点を表します。ログ・ファイルが移動される前に、ログ・ファイルに現在書き込まれている途中の項目が完了しなければなりません。これらの項目のタイム・スタンプは、ファイル名のタイム・スタンプより後の時刻を表す可能性があります。

**member** オプションが指定されない場合、すべてのメンバーに対する監査ログがアーカイブされます。この場合、データベース・サーバーが開始済みでなければなりません。データベース・サーバーが開始済みでない場合、アーカイブは各メンバーに対して実行されます。 **archive** の実行対象のメンバーを示すには **member** オプションを指定する必要があります (AUD0029)。

**archive** オプションを指定すると、アーカイブ実行対象の各メンバーから結果およびファイル名が出力されます。

**start** このパラメーターを使用すると、インスタンスのみに対して、db2audit.cfg ファイルの内容に基づくイベント監査が監査機能によって開始されます。DB2 パーティション・データベース・インスタンスでは、この節が指定されている場合、インスタンス・レベルおよびクライアント・レベルのアクティビティに対する、すべてのデータベース・パーティションの監査が開始されます。監査対象としてイベントの「**audit**」(監査) カテゴリが指定されている場合、監査機能が開始された時点で監査レコードがログに記録されます。これは、(AUDIT DDL ステートメントを介して制御される) データベース・レベルの監査には影響を与えません。

**stop** このパラメーターを使用すると、インスタンスのみに対するイベント監査が監査機能によって停止されます。DB2 パーティション・データベース・インスタンスでは、この節が指定されている場合、インスタンス・レベルおよびクライアント・レベルのアクティビティに対する、すべてのデータベース・パーティションの監査が停止します。監査対象としてイベントの「**audit**」(監査) カテゴリが指定されている場合、監査機能が停止された時点で監査レコードがログに記録されます。これは、(AUDIT DDL ステートメントを介して制御される) データベース・レベルの監査には影響を与えません。

? このパラメーターは、**db2audit** コマンドのヘルプ情報を表示します。

## 例

次の例は、パーティション・データベース環境において区切り ASCII ファイルをアーカイブして抽出する一般的な方法を示しています。UNIX の削除コマンド (**rm**) は、古い区切り ASCII ファイルを削除します。

```
rm /auditdelasc/*.del
db2audit flush
db2audit archive database mydb to /auditarchive
```

(次の手順で使用されるファイルが示されます)

```
db2audit extract delasc to /auditdelasc from files /auditarchive
/db2audit.db.mydb.log.*.20070514102856
```

.del ファイルを DB2 表にロードします。

## 使用上の注意

- データベース・レベルの監査は、**AUDIT** ステートメントによって制御されます。
- インスタンス・レベルの監査機能は、明示的に停止および開始する必要があります。開始時に監査機能は、既存の監査構成情報を使用します。監査機能は DB2 データベース・サーバーとは独立した機能なので、インスタンスが停止した場合でもアクティブのままです。事実、インスタンスが停止した時点で、監査レコードが監査ログ中に生成されることがあります。
- 監査のさまざまなユーティリティーを使用する前に、**db2audit start** コマンドを発行することによって、監査機能が確実にオンであるようにしてください。
- 生成される監査レコードには、いくつかの異なるカテゴリーがあります。監査するために使用可能なイベントのカテゴリーに関する以下の記述では、各カテゴリーの名前に続いて、カテゴリー・タイプの識別に使用される 1 つの単語のキーワードがあることに注意してください。監査のために使用できるイベントのカテゴリーは、以下のとおりです。
  - 監査 (**audit**)。監査設定値が変更された場合、または監査ログへのアクセスがあった場合に、レコードが生成されます。
  - 許可検査 (**checking**)。DB2 データベース・オブジェクトまたは関数にアクセスしたり、それらを操作したりする試みに関する許可検査中に、レコードが生成されます。
  - オブジェクト保守 (**objmaint**)。データ・オブジェクトの作成時またはドロップ時にレコードが生成されます。
  - セキュリティー保守 (**secmaint**)。オブジェクトまたはデータベースの特権または DBADM 権限を付与したり取り消したりする時点で、レコードが生成されます。データベース・マネージャーのセキュリティ構成パラメーター **sysadm\_group**、**sysctrl\_group**、または **sysmaint\_group** が変更されたときにもレコードが生成されます。
  - システム管理 (**sysadmin**)。SYSADM、SYSMAINT、または SYSCTRL の権限を必要とする操作が実行された時点で、レコードが生成されます。
  - ユーザー妥当性検査 (**validate**)。ユーザー認証時、またはシステムのセキュリティ情報の検索時に、レコードが生成されます。
  - 操作コンテキスト (**context**)。インスタンス操作が実行された時点で、その操作のコンテキストを示すレコードが生成されます。このカテゴリーにより、監査ログ・ファイルをより良い方法で解釈できます。ログのイベント相関関係子フィールドと共に使用すれば、一群のイベントを単一のデータベース操作に関連付けることができます。
  - 監査の対象として指定できるのは、失敗、成功、その両方、または「どちらも監査しない」です。
- インスタンスに対する 1 つの操作により、複数のレコードが生成されることがあります。生成されて監査ログに移されるレコードの実際数は、監査機能の構成での指定内容に基づいて記録されるイベントのカテゴリーの数によって異なります。また、監査の対象が成功だけか、失敗だけか、それともその両方かによっても異なります。そのため、監査の対象となるイベントが選択できるようになっていることは非常に重要です。

## db2audit - 監査機能管理者用ツール

- 監査ログをクリーンアップまたは表示するには、**archive** を定期的に行い、アーカイブ・ファイルに対して **extract** を実行して役立つ情報を保存します。その後、ファイル・システムの標準的な削除コマンドを使って監査ログを削除できます。

## db2batch - ベンチマーク・ツール

フラット・ファイルまたは標準入力から SQL ステートメントおよび XQuery ステートメントを読み取り、ステートメントを動的に準備および記述し、応答セットを戻します。

このツールは、単一パーティション・データベースと複数パーティション・データベースの両方で機能できます。

このツールのオプション・パラメーターにより、応答セットからフェッチする行数、出力ファイルや標準出力に送信するフェッチ済み行数、および戻されるパフォーマンス情報のレベルを制御できます。

出力のデフォルトは、標準出力を使用する設定になっています。結果サマリーの出力ファイルの名前を指定することができます。

### 許可

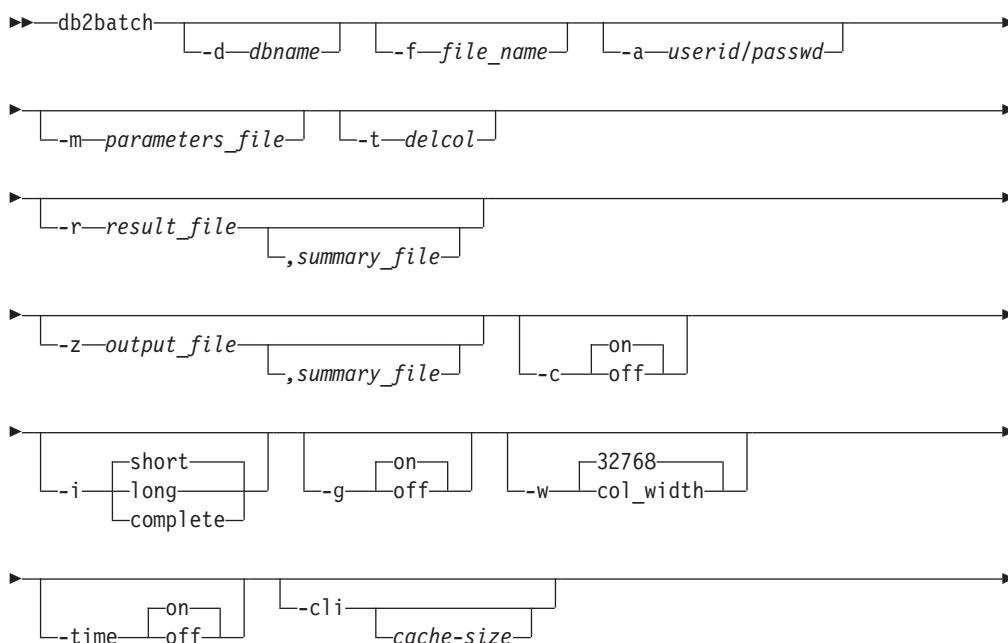
読み取られる SQL ステートメントまたは XQuery ステートメントが必要とするものと同じ権限レベル。

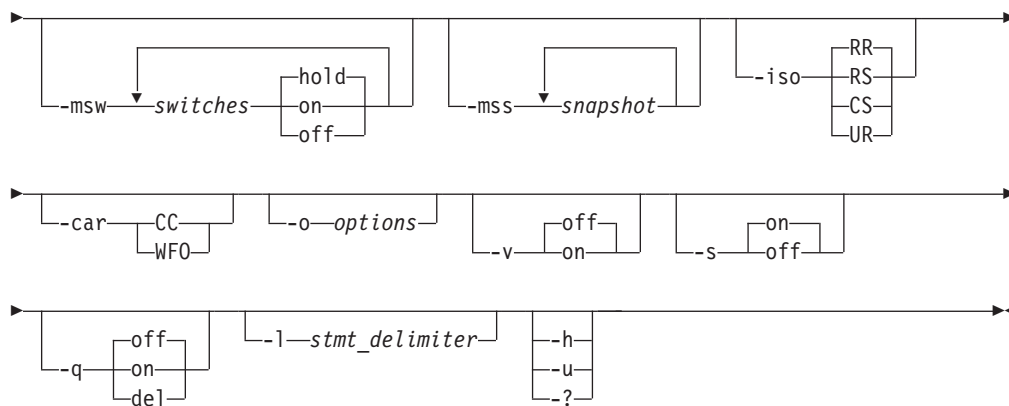
パフォーマンス情報のレベルを指定する **-o p** オプションを使用する場合、または Explain モードを設定する **-o e** オプションを使用する場合には、SYSMON 権限が必要です。

### 必要な接続

なし。このコマンドは、データベース接続を確立します。

### コマンド構文





## コマンド・パラメーター

### -d *dbname*

SQL ステートメントまたは XQuery ステートメントが適用されるデータベースの別名。このオプションが指定されない場合、**DB2DBDFT** 環境変数の値が使用されます。

### -f *file\_name*

SQL ステートメントおよび XQuery ステートメントが入っている入力ファイルの名前。デフォルトは標準入力です。

コメント・テキストの前に 2 つのハイフンを付けて **--コメント** のように指定すると、コメント・テキストが識別されます。2 つのハイフンの後から行の終わりまでのすべてのテキストは、コメントとして扱われます。隣接する 2 つのハイフンを含む文字列を単一または二重引用符で区切った場合、その文字列はコメントではなく、文字列定数として扱われます。出力にコメントを含めるには、**--#COMMENT comment** のようにマークを付けます。

ブロックとは、単体として扱われる複数の SQL ステートメントおよび XQuery ステートメントの集まりです。デフォルトでは、ブロック内のステートメントの情報は (1 つずつではなく) すべて一度で収集されます。照会ブロックの開始は、**--#BGBLK** で表します。照会ブロックの終了は、**--#EOBLK** で表します。ブロック定義時に **--#BGBLK repeat\_count** のように繰り返しカウントを指定することにより、繰り返しループの中に照会ブロックを含めることができます。ブロック内のステートメントは、ループの最初の反復時にのみ準備されます。

ブロックの特定のステートメントや特定の反復でのパラメーター値を指定するために、**#PARAM** ディレクティブまたはパラメーター・ファイルを使用することができます。詳しくは、**-m** オプションに関する下記のセクションを参照してください。

1 つ以上の制御オプションを指定するには、次のようにします。 **--#SET control option value**。有効な制御オプションは、以下のとおりです。

### **ROWS\_FETCH**

応答セットからフェッチする行数。有効な値は **-1** から **n** です。デフォルト値は **-1** (すべての行をフェッチする) です。

**ROWS\_OUT**

フェッチされた行のうち出力へ送られる行数。有効な値は -1 から  $n$  です。デフォルト値は -1 (フェッチされた行をすべて出力へ送る) です。

**PERF\_DETAIL** *perf\_detail*

戻されるパフォーマンス情報のレベルを指定します。有効な値は以下のとおりです。

- 0 時間情報やモニター・スナップショットをまったく戻さない。
- 1 経過時間のみ戻す。
- 2 経過時間とアプリケーションのスナップショットを戻す。
- 3 経過時間、およびデータベース・マネージャー、データベース、アプリケーションのスナップショットを戻す。
- 4 データベース・マネージャー、データベース、アプリケーション、およびステートメントのスナップショットを戻す。(自動コミットが OFF になっており、かつステートメント・ブロックではなく単一ステートメントを処理している場合にだけ、ステートメントのスナップショットが戻されます。) スナップショットにはハッシュ結合情報は組み込まれません。
- 5 データベース・マネージャー、データベース、アプリケーション、およびステートメントのスナップショットを戻す。(自動コミットが OFF になっており、かつステートメント・ブロックではなく単一ステートメントを処理している場合にだけ、ステートメントのスナップショットが戻されます。) バッファ・プール、表スペース、および FCM も戻します (FCM スナップショットはマルチ・データベース・パーティション環境でのみ使用できる)。スナップショットにはハッシュ結合情報は組み込まれません。

デフォルト値は 1 です。1 より大きい値は DB2 バージョン 2 と DB2 データベース・サーバーでのみ有効で、ホスト・マシンでは現在サポートされていません。

**ERROR\_STOP**

重大ではないエラーが発生したとき、**db2batch** を停止させるかどうかを指定します。有効な値は以下のとおりです。

- no** 重大ではないエラーが発生しても実行を続ける。これはデフォルト・オプションです。
- yes** 重大ではないエラーが発生したら実行を停止する。

**DELIMITER**

1 文字か 2 文字のステートメント終結区切り文字です。デフォルト値はセミコロン (;) です。

**SLEEP**

スリープの秒数。有効な値は 1 から  $n$  です。



**PAUSE**

継続するかどうかの入力を要求するプロンプトをユーザーに出します。

**SNAPSHOT** *snapshot*

取得するモニター・スナップショットを指定します。取得可能なスナップショットについては、**-mss** オプションを参照してください。

**TIMESTAMP**

タイム・スタンプを生成します。

**TIMING**

時間情報を出力します。有効な値は以下のとおりです。

**ON** 時間情報を出力する。これはデフォルトです。

**OFF** 時間情報を出力しない。

**-a** *userid/passwd*

データベースへの接続に使用するユーザー ID とパスワードを指定します。斜線 (/) を含めなければなりません。

**-m** *parameters\_file*

ステートメントの実行前に SQL ステートメント・パラメーター・マーカースにバインドされるパラメーター値が入っている入力ファイルを指定します。デフォルトでは、パラメーターをバインドしません。

パラメーター・ファイルを使用する場合、各行は、特定のステートメントやブロックの特定の反復でのパラメーター値を指定します。一方、**#PARAM** ディレクティブを使用した場合には、各ステートメントの各パラメーターごとに、複数の値やパラメーター範囲があらかじめ指定されます。ブロックのそれぞれの反復では、各パラメーターごとに指定された値セットからランダムな値が選択されます。**#PARAM** ディレクティブとパラメーター・ファイルを混合することはできません。

パラメーター値の形式:

```
-36.6      'DB2'      X'0AB2'   G'...'   NULL
12         'batch'    x'32ef'   N'...'   null
+1.345E-6  'db2 batch' X'afd4'   g'...'   Null
```

各パラメーターは SQL 定数のように定義され、空白文字によって他のパラメーターから分離されます。区切りなしテキストは数値を、' で単純に区切られたテキストは 1 バイト文字ストリングを、最初に x または X の付いたテキストの単一引用符 (') に囲まれた部分は 16 進数字の対としてエンコードされたバイナリー・ストリングを表します。さらに、最初に g、G、n、N が付いたテキストの単一引用符 (') 内は 2 バイト文字からなる GRAPHIC ストリングを、NULL (大/小文字を区別しない) はヌル値を表します。XML データを指定するには、( ) で区切られたテキストを使用します (例えば '<last>Brown</last>')。

パラメーター入力ファイルの形式:

第 X 行は、入力ファイル内の実行される X 番目の SQL ステートメントに提供されるパラメーター・セットをリストします。ステートメント・ブロックが反復されない場合、これは入力ファイルにリストされた X 番目の SQL ステートメントに対応します。ブランク行は、対応する SQL ステートメント



トメントのパラメーターを提供しないことを表します。パラメーターの数とタイプは、SQL ステートメントが必要とするパラメーターの数およびタイプと一致しなければなりません。

パラメーター・ディレクティブの形式:

```
--#PARAM [single | start:end | start:step:end] [...]
```

それぞれのパラメーター・ディレクティブは複数のパラメーター値からなる 1 つのセットを指定します。照会が実行されるたびに、このセットから 1 つのランダムな値が選択されます。このようなセットは、単一のパラメーター値の集まり、およびパラメーター値の範囲から構成されます。パラメーター値の範囲は、2 つの有効なパラメーター値の間にコロン (:) を付けることによって指定されます (または空白文字を区切り記号として使用することもできます)。開始値と終了値の間に 3 番目のパラメーター値を指定できます。これは、デフォルトをオーバーライドするステップ・サイズとして使用されます。それぞれのパラメーター範囲は、「開始値」、「開始値 + ステップ」、「開始値 + (ステップ x 2)」... 「開始値 + (ステップ x n)」を指定した場合と同じになります ( $n$  は「開始値 + (ステップ x n)  $\geq$  終了値」および「開始値 + ステップ x (n+1)  $>$  終了値」を満たす数として選択されます)。パラメーター・ディレクティブを使用すれば (NULL も含めて) あらゆるタイプのパラメーターの値セットを指定できますが、パラメーター値の範囲としてサポートされるのは数値 (整数と小数) だけです。

**-t delcol**

1 文字の列区切り記号を指定します。列区切り記号としてタブを指定するには **-t TAB**、スペースを指定するには **-t SPACE** をそれぞれ指定します。デフォルトでは、**-q on** オプションが設定されるとスペースが使用され、**-q del** オプションが設定されるとコンマが使用されます。

**-r result\_file [,summary\_file]**

照会結果が入る出力ファイルを指定します。デフォルトは標準出力です。エラー・メッセージは標準エラーで戻されます。オプションの *summary\_file* を指定した場合、サマリー表が含まれます。

**-z output\_file [,summary\_file]**

照会結果および戻されたエラー・メッセージが入る出力ファイルを指定します。デフォルトは標準出力です。エラー・メッセージは標準エラーでも戻されます。オプションの *summary\_file* を指定した場合、サマリー表が含まれます。このオプションは、バージョン 9.7 フィックスパック 1 から使用可能です。

**-c** 各ステートメントが実行された結果の変更内容を自動的にコミットします。デフォルトは ON です。

**-i** 経過時間インターバルを測定することを指定します。有効な値は以下のとおりです。

**short** 各ステートメント実行時の経過時間を測定します。これはデフォルトです。

**long** ステートメント間の追加の処理時間を含めて、各ステートメント実行の経過時間を測定します。

**complete**

各ステートメント実行時の経過時間を測定します。準備、実行、フェッチにかかった時間が別々に報告されます。

**-g** 時間をブロックごとまたはステートメントごとのどちらで報告するかを指定します。有効な値は以下のとおりです。

**on** ブロック全体のスナップショットを取得し、ブロックの時間だけがサマリー表に報告されます。これはデフォルトです。

**off** ブロック内の実行される各ステートメントごとにスナップショットが取得され、サマリー表に時間が報告されます。

**-w** 結果セットの列の最大幅を指定します (許容される範囲は 0 から 2 G)。データが切り捨て不能でない限り、データは表示時にこの幅になるように切り捨てられます。この設定値を増やして警告 **CLI0002W** が出ないようにすると、より正確なフェッチ時間を得ることができます。デフォルトの最大幅は 32768 列です。

**-time** 時間情報を報告するかどうかを指定します。有効な値は以下のとおりです。

**on** 時間を報告する。これはデフォルトです。

**off** 時間を報告しない。

**-cli** 組み込み動的 SQL モード (以前は **db2batch** のデフォルト・モードだったもの) は、サポートされなくなりました。このコマンドは、CLI モードでのみ実行されます。 **-cli** オプションは、後方互換性のためにのみ存在しています。これを (オプションとして *cache-size* 引数を含めて) 指定してもエラーにはなりません、内部では無視されます。

**-msw switch**

指定されたそれぞれのモニター・スイッチの状態を設定します。 *uow*、*statement*、*table*、*bufferpool*、*lock*、*sort*、*timestamp* のオプションのいずれも指定できます。特殊なスイッチ *all* は、上述のすべてのスイッチを設定します。指定するスイッチごとに、以下の値のいずれかを選択する必要があります。

**hold** スwitchの状態を変更しない。これはデフォルトです。

**on** スwitchを ON に切り替える。

**off** スwitchを OFF に切り替える。

**-mss snapshot**

(**-g** オプションに応じて) それぞれのステートメントまたはブロックの実行後に取得するモニター・スナップショットを指定します。一度に複数のスナップショットを取ることができます。この場合、すべてのスナップショットの情報が出力前に 1 つの大きな表に結合されます。可能なスナップショットは次のとおりです。 *applinfo\_all*、*dbase\_applinfo*、*dcx\_applinfo\_all*、*db2*、*dbase*、*dbase\_all*、*dcx\_dbase*、*dcx\_dbase\_all*、*dbase\_remote*、*dbase\_remote\_all*、*agent\_id*、*dbase\_appls*、*appl\_all*、*dcx\_appl\_all*、*dcx\_appl\_handle*、*dcx\_dbase\_appls*、*dbase\_appls\_remote*、*appl\_remote\_all*、*dbase\_tables*、*appl\_locks\_agent\_id*、*dbase\_locks*、*dbase\_tablespace*s、*bufferpools\_all*、*dbase\_bufferpools*、および *dynamic\_sql*。

特殊なスナップショット all は、上記のすべてのスナップショットを取得します。アプリケーション ID を扱うスナップショットはサポートされず、代わりに対応するエージェント ID (アプリケーション・ハンドル) が使用されます。デフォルトでは、モニター・スナップショットはまったく取得されません。

**-iso** データ・アクセス中にデータをロックして他のプロセスから分離する方法を決定する、分離レベルを指定します。デフォルトでは、**db2batch** は分離レベル RR を使用します。

db2cli.ini ファイル内の構成キーワード **TxnIsolation** は、**db2batch** に影響を与えません。分離レベルを RR 以外に設定してこのコマンドを実行するには、**-iso** パラメーターを指定する必要があります。

**RR** 反復可能読み取り (ODBC 逐次化可能)。これはデフォルトです。

**RS** 読み取り固定 (ODBC 反復可能読み取り)。

**CS** カーソル固定 (ODBC コミット読み取り)。

**UR** 非コミット読み取り (ODBC 非コミット読み取り)。

**-car** **db2batch** 操作に使用する並行アクセス解決方法を指定します。**-car** パラメーターには、正しく構成されたデータベース・サーバーおよび CS に設定された分離レベル・パラメーター **-iso** が必要になります。

**CC** 更新または削除の処理中に、**db2batch** 操作が、該当するスキャンに対して現在コミット済みバージョンのデータを使用することを指定します。挿入処理中の行はスキップできます。このオプションは、有効な分離レベルがカーソル固定または読み取り固定である場合に適用され (読み取り固定の場合、それはコミットされていない挿入だけをスキップします)、そうでない場合は無視されます。該当するスキャンには、非読み取り専用ステートメントに加え、読み取り専用ステートメントの一部にできる読み取り専用スキャンが含まれます。

**WFO** **db2batch** 操作が、操作の結果を待機するように指定します。カーソル固定およびさらに上位のスキャンで、更新処理中または削除処理中のデータが検出されたときには **db2batch** はコミットまたはロールバックを待機します。挿入処理中の行はスキップされません。

#### **-o options**

制御オプション。有効なオプションは以下のとおりです。

##### **f** *rows\_fetch*

応答セットからフェッチする行数。有効な値は -1 から *n* です。デフォルト値は -1 (すべての行をフェッチする) です。

##### **r** *rows\_out*

フェッチされた行のうち出力へ送られる行数。有効な値は -1 から *n* です。デフォルト値は -1 (フェッチされた行をすべて出力へ送る) です。

##### **p** *perf\_detail*

戻されるパフォーマンス情報のレベルを指定します。有効な値は以下のとおりです。

- 0 時間情報やモニター・スナップショットをまったく戻さない。
- 1 経過時間のみ戻す。
- 2 経過時間とアプリケーションのスナップショットを戻す。
- 3 経過時間、およびデータベース・マネージャー、データベース、アプリケーションのスナップショットを戻す。
- 4 データベース・マネージャー、データベース、アプリケーション、およびステートメントのスナップショットを戻す。(自動コミットが OFF になっており、かつステートメント・ブロックではなく単一ステートメントを処理している場合にだけ、ステートメントのスナップショットが戻されます。)
- 5 データベース・マネージャー、データベース、アプリケーション、およびステートメントのスナップショットを戻す。(自動コミットが OFF になっており、かつステートメント・ブロックではなく単一ステートメントを処理している場合にだけ、ステートメントのスナップショットが戻されます。) バッファ・プール、表スペース、および FCM も戻します (FCM スナップショットはマルチ・データベース・パーティション環境でのみ使用できる)。

デフォルト値は 1 です。1 より大きい値は DB2 バージョン 2 と DB2 データベース・サーバーでのみ有効で、ホスト・マシンでは現在サポートされていません。

**o** *query\_optimization\_class*

照会最適化クラスを設定する。有効な値は 0、1、2、3、5、7、または 9 です。デフォルトは -1 (現行の最適化クラスを使用) です。

**e** *explain\_mode*

**db2batch** 実行時の **explain** モードを設定する。このコマンドを使用する前に、**Explain** 表を作成しておく必要があります。有効な値は以下のとおりです。

**no** 照会のみ実行 (デフォルト)。

**explain**

**Explain** 表に書き込む。このオプションは、**Explain** 表に書き込み、**Explain** スナップショットを取得させます。

**yes** **Explain** 表に書き込み、照会を実行する。このオプションは、**Explain** 表に書き込み、**Explain** スナップショットを取得させます。

**s** *error\_stop*

重大ではないエラーが発生したとき、**db2batch** を停止させるかどうかを指定します。有効な値は以下のとおりです。

**no** 重大ではないエラーが発生しても実行を続ける。これはデフォルト・オプションです。

**yes** 重大ではないエラーが発生したら実行を停止する。

- v 冗長。照会処理中に標準エラーに情報を送信します。デフォルト値は OFF です。
- s サマリー表。照会または照会のブロックごとに、サマリー表を提供します。その表には、経過時間 (算術平均と幾何平均を含む)、フェッチされた行、および行出力が含まれます。
- q 照会の出力。有効な値は以下のとおりです。
  - off** 照会結果およびすべての関連情報を出力する。これはデフォルトです。
  - on** 照会結果だけを非区切り形式で出力する。
  - del** 照会結果だけを区切り形式で出力する。
- l *stmt\_delimiter*  
終了文字 (ステートメント区切り文字) を指定します。区切り文字は 1 文字または 2 文字が可能です。デフォルトはセミコロン (;) です。
- h | -u | -?  
ヘルプ情報を表示します。このオプションを指定すると、他のすべてのオプションは無視され、ヘルプ情報だけが表示されます。

## 例

1. 以下に示すのは、db2batch -d crystal -f update.sql コマンドの出力例です。

```
* Timestamp: Thu Feb 02 2006 10:06:13 EST
-----

* SQL Statement Number 1:

create table demo (c1 bigint, c2 double, c3 varchar(8));

* Elapsed Time is:      0.101091 seconds
-----

* SQL Statement Number 2:

insert into demo values (-9223372036854775808, -0.0000000000000005, 'demo');

* Elapsed Time is:      0.002926 seconds
-----

* SQL Statement Number 3:

insert into demo values (9223372036854775807, 0.0000000000000005, 'demodemo');

* Elapsed Time is:      0.005676 seconds
-----

* SQL Statement Number 4:

select * from demo;

C1                C2                C3
-----
-9223372036854775808 -5.000000000000000E-015 demo
9223372036854775807 +5.000000000000000E-015 demodemo

* 2 row(s) fetched, 2 row(s) output.
```

\* Elapsed Time is: 0.001104 seconds

\* SQL Statement Number 5:

drop table demo;

\* Elapsed Time is: 0.176135 seconds

\* Summary Table:

Type	Number	Repetitions	Total Time (s)	Min Time (s)	Max Time (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

Arithmetic Mean	Geometric Mean	Row(s) Fetched	Row(s) Output
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

\* Total Entries: 5  
 \* Total Time: 0.286932 seconds  
 \* Minimum Time: 0.001104 seconds  
 \* Maximum Time: 0.176135 seconds  
 \* Arithmetic Mean Time: 0.057386 seconds  
 \* Geometric Mean Time: 0.012670 seconds

\* Timestamp: Thu Feb 02 2006 10:06:13 EST

### 使用上の注意

- すべての SQL ステートメントは、--#SET DELIMITER コマンドによって設定される区切り文字 (デフォルトは「;」) で終わる必要があります。この区切り文字は 1 文字または 2 文字が可能です。
- SQL ステートメントの長さは、使用可能メモリー、および使用されるインターフェースによってのみ制限されます。ステートメントを複数の行に分割することは可能ですが、1 行に複数のステートメントを指定することはできません。
- 入力ファイルの行の長さは、使用可能メモリーによってのみ制限されます。
- **c** は自動的に CONNECT および CONNECT RESET ステートメントを発行します。
- タイミング・オプション **-i** に **long** が指定されている場合、PAUSE および SLEEP の時間が計測されます。
- explain オプションを使用するには、まず Explain 表を作成しておく必要があります。
- **db2batch** では、すべてのコマンド行オプションと入力ファイル・ステートメントは大/小文字を区別しません。

- **db2batch** がサポートするデータ・タイプは、INTEGER、CHAR、VARCHAR、LONG VARCHAR、FLOAT、SMALLINT、BIGINT、DECIMAL、DATE、TIME、TIMESTAMP、CLOB、GRAPHIC、VARGRAPHIC、LONGVARGRAPHIC、DBCLOB、BLOB、および XML です。
- `--#SET PERF_DETAIL perf_detail` (または `-o p perf_detail`) を使用すれば、モニター出力を素早く得ることができます。パフォーマンス詳細レベルが 1 より大きい場合、すべてのモニター・スイッチは **db2batch** によって内部的にオンに切り替えられます。モニター出力をより細かく制御する必要がある場合には、**-msw** および **-mss** オプション (または `--#SET SNAPSHOT`) を使用してください。
- **-r** オプションと **-z** オプションを同時に指定した場合は、**-r** オプションが指定しているものが **-z** オプションに含まれているので、**-r** オプションは無視されません。

## db2bfd - バインド・ファイル記述ツール

バインド・ファイルの内容を表示します。このユーティリティは、バインド・ファイルを作成する際に使用したプリコンパイル・オプションを表示するだけでなく、バインド・ファイル内の SQL ステートメントを調べ、検査するためにも使用できます。アプリケーションのバインド・ファイルに関連した問題を判別するのに役立ちます。

### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

- h** ヘルプ情報を表示します。このオプションを指定すると、他のすべてのオプションは無視され、ヘルプ情報だけが表示されます。
- b** バインド・ファイル・ヘッダーを表示します。
- s** SQL ステートメントを表示します。
- v** ホスト変数宣言を表示します。

*filespec*

内容が表示されるバインド・ファイルの名前です。



## db2caem - アクティビティ・イベント・モニター・データ・ツールのキャプチャー

**db2caem** ツールは、アクティビティ・イベント・モニターの作成手順を自動化します。

**db2caem** コマンドを実行してアクティビティ・イベント・モニターを作成し、SQL ステートメントのデータをキャプチャーできます。このデータは **db2support** コマンドで収集されます。**db2caem** ツールで収集され生成される情報には、以下のものが含まれます。

- モニター・メトリック (例えば、ステートメント実行の `total_cpu_time`) を含む、アクティビティ・イベント・モニターでキャプチャーされた詳細アクティビティ情報。
- フォーマット済み EXPLAIN 出力。これには、セクション実行時統計 (アクセス・プラン中の演算子別の統計) が含まれます。

**db2caem** ツールは、アクティビティ・イベント・モニターを使用して、ステートメントに関する情報をキャプチャーし、その後情報を抽出しフォーマットします。

**db2caem** ツールは、アクティビティ・イベント・モニターの作成プロセスを自動化します。

1. 対象となるステートメントのキャプチャーを使用可能にする。
2. ステートメントを呼び出す (各ステートメントは実行後、データベースへの副次作用を防ぐためロールバックされます)。
3. 出力情報をフォーマットする (対象となるステートメントのアクティビティ情報のエクスポートおよびキャプチャーされたセクションおよびセクション実行時統計からのフォーマットされた Explain 出力の生成を含む)。

### 許可

1. アクティビティ・イベント・モニターを作成するには、以下のいずれかの権限が必要です。

- DBADM 権限
- SQLADM 権限
- WLMADM 権限

および、WLM\_SET\_CONN\_ENV プロシージャの EXECUTE 特権。

2. アクティビティ・イベント・モニターを作成する必要がない場合は、以下の特権および権限が必要です。

- EXPLAIN\_FROM\_ACTIVITY プロシージャに対する EXECUTE 特権
- 指定されたスキーマ内の Explain 表に対する INSERT 特権
- ソース・アクティビティ・イベント・モニターに関するイベント・モニター表に対する SELECT 特権

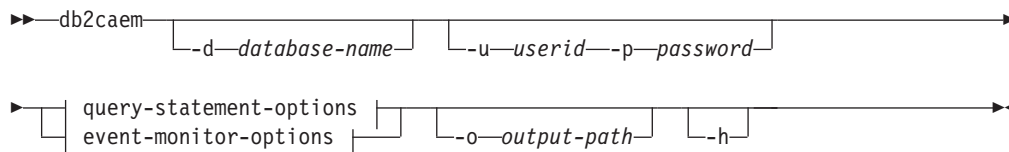
および、以下のいずれかの権限も必要になります。

- アクティビティ・イベント・モニター表の DATAACCESS 権限
- アクティビティ・イベント・モニター表の CONTROL または SELECT 特権

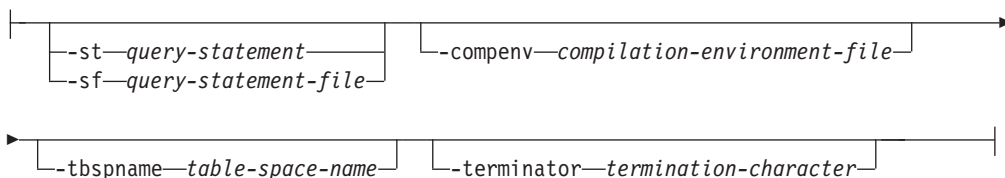
## 必要な接続

なし

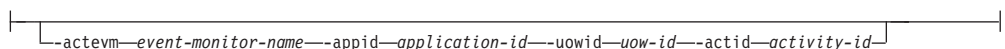
## コマンド構文



### query-statement-options:



### event-monitor-options:



## コマンド・パラメーター

### -d *database-name*

接続先データベースの名前を指定します。

### -u *userid*

データベースに接続する際に使用するユーザー ID を指定します。

### -p *password*

データベースに接続する際のユーザー ID のパスワードを指定します。

### -o *output-path*

db2caem の出力ファイルが、指定したパスに書き込まれます。

**-h** ヘルプ情報を表示します。このオプションを指定すると、他のすべてのオプションは無視され、ヘルプ情報だけが表示されます。

## query-statement-options

### -st *query-statement*

アクティビティ・イベント・モニター・データがキャプチャーされる SQL ステートメントを指定します。

注: SQL ステートメントは、ツールで指定されたデータベースで実行されます。

### -sf *query-statement-file*

アクティビティ・イベント・モニター・データがキャプチャーされる SQL ス

ステートメントを含むファイル・パスを指定します。**-terminator** オプションを使用して、SQL ステートメントの終了を示す文字を指定してください。

注: SQL ステートメントは、ツールで指定されたデータベースで実行されません。

### **-compenv** *compilation-environment-file*

SQL ステートメントを実行する際に使用されるコンパイル環境を指定します。コンパイル環境 (*comp\_env\_desc*) は BLOB データ・タイプであり、オプション入力としてファイルにより指定します。オプションが指定されない場合、SQL ステートメントを実行する際にはデフォルトのコンパイル環境が使用されます。

### **-tblspname** *table-space-name*

アクティビティ・イベント・モニターが作成される表スペース名を指定します。パーティション・データベース環境では、対象となる SQL ステートメントを実行するすべてのデータベース・パーティションにこの表スペースが存在しなければなりません。オプションが指定されない場合、アクティビティ・イベント・モニターを作成する必要がある際には、デフォルトの表スペースが使用されます。

### **-terminator** *termination-character*

**-sf** SQL ファイルに複数のステートメントがある場合に、ファイル中の SQL ステートメントの終了を示す文字を指定します。デフォルトはセミコロンです。

## event-monitor-options

以下のオプションは、既存のアクティビティ・イベント・モニターによって既にキャプチャーされている SQL ステートメントを一意に指定します。一緒に指定することにより、**アクティビティ・データ**および **Explain 出力**を抽出するステートメントを指定します。

注: ステートメントのセクションがキャプチャーされた場合にだけフォーマットされた Explain 出力は収集され、かつセクション実行時統計がステートメントに関してキャプチャーされた場合にだけフォーマットされた Explain 出力にセクション実行時統計が含まれます。

### **-actevm** *event-monitor-name*

対象となるステートメントのデータを含む既存のアクティビティ・イベント・モニターの名前を指定します。

### **-appid** *application-id*

対象となるステートメントを発行したアプリケーションを一意的に識別するアプリケーション ID (*appl\_id* モニター・エレメント) を指定します。

### **-uowid** *uow-id*

対象となるステートメントが実行された作業単位 ID (*uow\_id* モニター・エレメント) を指定します。

### **-actid** *activity-id*

対象となるステートメントのアクティビティ ID (*activity\_id* モニター・エレメント) を指定します。

### 例

以下の例は、**db2caem** ツールを使用してアクティビティ・イベント・モニターを作成し SQL ステートメントのデータをキャプチャーする方法を示しています。

- `db2caem -d sample -st "select * from staff"`

アクティビティ・イベント・モニターが作成され、詳細情報、セクションと値、および SQL ステートメント「select \* from staff」の実行時統計がキャプチャーされます。

- `db2caem -d sample -sf badquery.sql -terminator $`

アクティビティ・イベント・モニターが作成され、詳細情報、セクションと値、および badquery.sql ファイルで指定されている SQL ステートメントの実行時統計がキャプチャーされます。

- `db2caem -d sample -actevm mymon -appid *LOCAL.mikita.100203234904 -uowid 44 -actid 1`

アクティビティ・イベント・モニターの詳細情報、セクションと値、および既存のアクティビティ・イベント・モニターのイベント・モニター・オプションで指定される SQL ステートメントの実行時統計がキャプチャーされます。

**db2caem** ツールは、この例ではアクティビティ・イベント・モニターを作成しません。

### 使用上の注意

**db2caem** ツールを使用してアクティビティ・イベント・モニターを作成し、**db2support** コマンドで収集できるデータをキャプチャーできます。**db2caem** ツールがキャプチャーした全ての情報を格納するために、DB2CAEM\_<timestamp> ディレクトリーが生成されます。

## db2cap - CLI/ODBC 静的パッケージ・バインディング・ツール

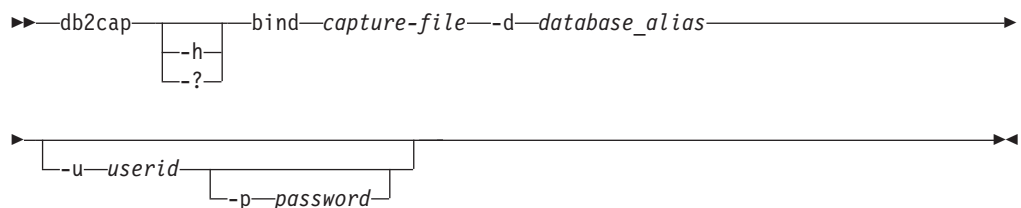
キャプチャー・ファイルを 1 つ以上の静的パッケージにバインドします。

キャプチャー・ファイルとは、CLI、ODBC、または .NET アプリケーションの静的プロファイル作成セッション中に生成されるもので、アプリケーションの実行中にキャプチャーされた SQL ステートメントが格納されます。このユーティリティーは、キャプチャー・ファイルを処理して、CLI、ODBC、または .NET ドライバーがアプリケーションの静的 SQL の実行時にそれを使用できるようにします。

### 許可

- キャプチャー・ファイルに記録されている SQL ステートメントが参照するデータベース・オブジェクトに対するアクセス権。
- **db2cap** コマンドを呼び出すための接続 ID が、それらと異なる場合は、OWNER や QUALIFIER といった BIND オプションを設定すれば十分な権限になります。
- そのパッケージが最初からバインドされている場合、BINDADD 権限になります。そうでなければ、BIND 権限が要求されます。

### コマンド構文



### コマンド・パラメーター

**-h | -?** コマンド構文のヘルプ・テキストを表示します。

#### **bind capture-file**

キャプチャー・ファイルからステートメントをバインドし、1 つ以上のパッケージを作成します。このキャプチャー・ファイルは .NET 用の pureQueryXML ファイルとも呼ばれます。

#### **-d database\_alias**

1 つ以上のパッケージを含んでいるデータベースに、データベース別名を指定します。

#### **-u userid**

データ・ソースに接続するために使うユーザー ID を指定します。ユーザー ID が指定されていない場合は、承認許可 ID をシステムから取得します。

#### **-p password**

データ・ソースに接続するためにパスワードを指定します。

### 使用上の注意

UNIX プラットフォーム上では、コマンドは小文字で入力する必要がありますが、Windows オペレーティング・システムでは、小文字も大文字も入力することができます。

ます。 .NET アプリケーション用の静的パッケージ・バインドは、Windows オペレーティング・システムでのみサポートされます。

このユーティリティーは、キャプチャー・ファイルの中で検索できる多くのユーザー指定 BIND オプションをサポートしています。 BIND オプションを変更するには、テキスト・エディターでキャプチャー・ファイルを開きます。

SQLERROR(CONTINUE) および VALIDATE(RUN) BIND オプションはパッケージを作成するために使用することができます。

パッケージを作成するために、このユーティリティーを使う場合、静的プロファイルは無効にしておく必要があります。

作成されるパッケージの数は、キャプチャー・ファイルで記録される SQL ステートメントに使用される分離レベルによって変わります。パッケージ名はキャプチャー・ファイルのパッケージ・キーワードから取った最初の 7 文字までと、1 文字の接尾部をつないで構成されます。

- 0 - 非コミット読み取り (UR)
- 1 - カーソル固定 (CS)
- 2 - 読み取り固定 (RS)
- 3 - 反復可能読み取り (RR)
- 4 - コミットなし (NC)

パッケージに関する特定の情報を獲得するには、ユーザーは以下の手順が必要です。

- キャプチャー・ファイルにある COLLECTION および PACKAGE キーワードを使用して、適切な SYSIBM カタログ表を照会します。
- キャプチャー・ファイルを表示します。

## db2cat - システム・カタログ分析

バック記述子の内容を分析します。データベース名および他の限定情報を指定すると、このコマンドはその情報をシステム・カタログで照会し、結果の書式を設定します。

サーバーで発行されなければなりません。

### 許可

なし

### 必要な接続

なし

### コマンド構文

```

db2cat [-d dbname] [-h] [-l] [-n name] [-o outfile]
        [-p descriptor] [-vi versionID] [-s schema] [-t] [-v]
        [-x] [-cb]

```

### コマンド・パラメーター

#### **-d** *dbname*

*dbname* は、コマンドがシステム・カタログを照会するデータベースの名前です。

**-h** 使用法情報を表示します。

**-l** オブジェクト名の大/小文字の区別をします。

#### **-n** *name*

オブジェクトの名前を指定します。

#### **-o** *outfile*

出力ファイルの名前を指定します。

#### **-p** *descriptor*

表示するバック記述子 (pd) の名前を指定します。ここで、*descriptor* は以下の値のうちの 1 つです。

#### **check**

表チェック制約のバック記述子を表示します。

#### **controls**

指定の表に関して有効になっているすべての権限およびマスクのバック記述子を表示します。

#### **rel**

参照整合性制約のバック記述子を表示します。

**table**

表のバック記述子を表示します。これには、表に少なくとも 1 つ存在する場合にはインライン長が含まれます。

**summary**

サマリー表のバック記述子を表示します。

**syscontrols**

指定のマスクまたは権限のバック記述子を表示します。

**trig**

表トリガーのバック記述子を表示します。

**view**

ビューのバック記述子を表示します。

**remote**

リモート非リレーショナル・データ・ソースのバック記述子を表示します。

**ast**

マテリアライズ照会表のバック記述子を表示します。

**routine**

ルーチンのバック記述子を表示します。

**sysplan**

パッケージのバック記述子を表示します。

**datatype**

構造化タイプのバック記述子を表示します。

**sequence**

シーケンスのバック記述子を表示します。

**esri**

キー変換スレッドおよび索引拡張のバック記述子を表示します。

**event**

イベント・モニターのバック記述子を表示します。

**server**

サーバーのバック記述子を表示します。

**auth**

このオブジェクトで被認可者が保持する特権の表示。

**-vi** *versionID*

パッケージのバージョン ID のバック記述子を指定します。 **-vi** は、 **-p sysplan** が指定された場合にのみ有効です。 *versionID* が省略された場合、デフォルトは空ストリングです。

**-s** *schema*

オブジェクト・スキーマの名前を指定します。

**-t** 端末出力を表示します。

**-v** バック記述子を妥当性検査します。このパラメーターは、表のバック記述子のみで有効です。



-x カタログ内の表スペース・エクステンツ・サイズを妥当性検査します (表名は必要ない)。

-cb

SYSCAT.BUFFERPOOLDBPARTITIONS (表名は必要ない) から孤立行を除去します。

## 例

### 例 1

以下のコマンドは、表 `splt_tb1` (スキーマは `raguk`) のパック記述子情報を、データベース `testdb` から端末にプリントします。

```
db2cat -d testdb -s raguk -t -n splt_tb1
```

## 使用上の注意

- **db2cat** コマンドの実行時には、**-d**、**-s**、および **-n** の各オプションは必須です。
- 表名および表スキーマは、LIKE 述部形式で指定できます。この場合、パーセント記号 (%) と下線 (\_) をパターン・マッチング文字として使用して、1 つの呼び出しで複数のソースを選択できます。
- プロンプトは、指定されていない、または完全に指定されていないすべてのフィールド (**-h** および **-l** オプション以外) に対して表示されます。
- **-o** をファイル名なしで指定し、**-t** を指定しない場合は、ファイル名の入力を促すプロンプトが表示されます (デフォルト名は `db2cat.out`)。
- **-o** も **-t** も指定しない場合は、ファイル名の入力を促すプロンプトが表示されず (デフォルトは端末出力です)。
- **-o** と **-t** の両方を指定した場合は、出力が端末に送信されます。

## db2cfexp - 接続構成エクスポート・ツール

接続構成情報をエクスポート・プロファイルにエクスポートします。あとでそのプロファイルを、類似のインスタンス・タイプの別の DB2 データベース・ワークステーション・インスタンス (つまり、クライアント・インスタンスからクライアント・インスタンス) でインポートすることができます。

生成されるプロファイルには、現行の DB2 データベース・インスタンスに関連した構成情報だけが含まれます。このプロファイルのことを、クライアント 構成プロファイルまたはインスタンス の構成プロファイルといいます。

このユーティリティーは、接続構成情報を、構成プロファイルというファイルにエクスポートします。これは、指定されるエクスポート・オプションの要件を満たすのに必要な構成情報をすべてパッケージする、非対話式ユーティリティーです。エクスポートできる項目は次のとおりです。

- データベース情報 (DCS および ODBC 情報を含む)
- ノード情報
- プロトコル情報
- データベース・マネージャ構成設定
- レジストリー設定
- 共通 ODBC/CLI 設定

このユーティリティーは、複数の同様のリモート DB2 クライアントがインストール、構成、および維持される状況において特に役立ちます (例えば、クライアント構成のテンプレートを複製または作成する場合)。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL

### コマンド構文

```

▶▶ db2cfexp filename [TEMPLATE] [BACKUP] [MAINTAIN]

```

### コマンド・パラメーター

*filename*

ターゲット・エクスポート・ファイルの完全修飾名を指定します。このファイルは、構成プロファイルと呼ばれます。

#### TEMPLATE

同じインスタンス・タイプ (つまり、クライアント・インスタンスからクライアント・インスタンス) の他のインスタンス用のテンプレートとして使用される、構成プロファイルを作成します。プロファイルには次のことに関する情報が含まれます。

- 関連 ODBC および DCS 情報を含めたすべてのデータベース
- エクスポートされるデータベースに関連したすべてのノード
- 共通 ODBC/CLI 設定
- データベース・マネージャ構成の共通クライアント設定
- DB2 レジストリーの共通クライアント設定

### BACKUP

ローカル・バックアップの目的で、DB2 データベース・インスタンスの構成プロファイルを作成します。このプロファイルには、このローカル・インスタンスのみに関係のある、特定の性質の情報も含め、インスタンス構成情報がすべて含まれています。プロファイルには次のことに関する情報が含まれます。

- 関連 ODBC および DCS 情報を含めたすべてのデータベース
- エクスポートされるデータベースに関連したすべてのノード
- 共通 ODBC/CLI 設定
- データベース・マネージャ構成のすべての設定
- DB2 レジストリーのすべての設定
- すべてのプロトコル情報

### MAINTAIN

他のインスタンスを維持または更新するために、データベースおよびノードに関連した情報だけを含む構成プロファイルを作成します。

### 注:

**db2cfexp** コマンドは、クライアントからのファイル・データ・ソース・ロケーション情報をエクスポートしません。

デフォルトのロケーションを使用する場合、アクションは不要です。クライアント上でデフォルトのロケーションを変更する場合、接続情報をエクスポートするときにこのロケーションを手動でコピーする必要があります。

ファイル・データ・ソース・ロケーションをクライアント間でコピーするには、以下のようにします。

1. 接続情報をエクスポートするクライアント上で、`%DB2PATH%\TOOLS` ディレクトリを見つけます。
2. `CA.properties` ファイルをコピーします。
3. 接続情報をインポートするクライアント上で、`%DB2PATH%\TOOLS` ディレクトリを見つけます。
4. 既存の `CA.properties` ファイルを、コピー元のクライアントから取られたコピーで上書きします。

これでクライアント間でファイル・データ・ソース・ロケーションが複写されました。

## db2cfimp - 接続構成インポート・ツール

接続構成情報を、構成プロファイルというファイルからインポートします。これは、構成プロファイル中で見つかるすべての情報をインポートしようとする、非対話式ユーティリティです。

構成プロファイルには、次のような接続項目を含めることができます。

- データベース情報 (DB2 Connect および ODBC 情報を含む)
- ノード情報
- プロトコル情報
- データベース・マネージャ構成設定
- DB2 データベース・レジストリー設定
- 共通 ODBC/CLI 設定

このユーティリティを使用すると、前に構成された同様の別のインスタンスから (つまりクライアント・インスタンスからクライアント・インスタンスへ)、接続情報を複製することが可能になります。複数の同様のリモート DB2 クライアントがインストール、構成、および維持される状況において特に役立ちます (例えば、クライアント構成のテンプレートを複製または作成する場合)。インスタンスの複製時には、インポートするプロファイルは常に、1 つの DB2 データベース・インスタンスだけに関する構成情報を含んだクライアント構成プロファイルでなければなりません。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL

注:

- ルート ID を使ってこのツールを実行しないでください。
- 有効な ID を使ってこのツールを実行するには、構成プロファイルをインポートするための適切な権限が ID が必要です。

### コマンド構文

▶▶—db2cfimp—*filename*————▶▶

### コマンド・パラメーター

*filename*

インポートされる構成プロファイルの完全修飾名を指定します。有効なインポート構成プロファイルは、**db2cfexp** コマンドを使用して DB2 データベースまたは DB2 Connect 製品によって作成されたプロファイルです。

## db2chglibpath - 組み込みランタイム・ライブラリー検索パスの変更

組み込みランタイム・ライブラリー検索パスの値を、実行可能ファイルまたは共有ライブラリー・ファイルで変更します。これは、既存の値が有効でなくなった場合に、組み込みランタイム・ライブラリー検索パスの値を新規ユーザー指定値で置き換えるために使用できます。

**db2chglibpath** コマンドは、オペレーティング・システム・ライブラリー検索パス環境変数の使用要件を置換するために使用できます。変数には、**LIBPATH** (AIX)、**SHLIB\_PATH** (HPPA、HPIPF) および **LD\_LIBRARY\_PATH** (AIX、SUN、HPPA64、HPIPF および Linux) などが含まれます。このコマンドは Linux および UNIX オペレーティング・システムでのみサポートされています。これは、**DB2DIR/bin** ディレクトリーにあります。ここで、**DB2DIR** は DB2 データベースのインストール・ロケーションです。

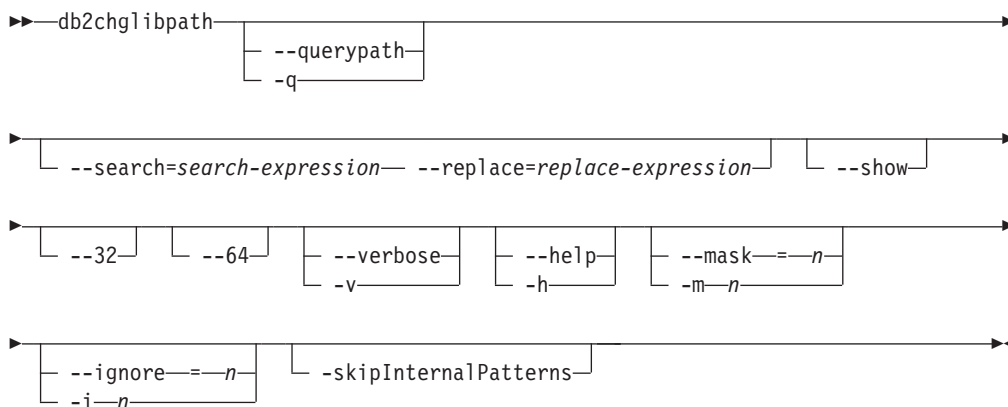
### 前提条件

- 共有ライブラリーまたは実行可能ファイルの変更には、読み取りおよび書き込みアクセスが必要です。
- バイナリーには、まず、組み込みライブラリー・パスがある必要があり、その組み込みパスは、既にバイナリーにあるパスよりも大きいものに変更することはできません。
- 組み込みランタイム・ライブラリー検索パス値を置き換えるユーザー指定値の長さは、既存の値の長さ以下でなければなりません。
- このコマンドは、共有ライブラリーまたは実行可能ファイルのバイナリー・コードを直接変更します。このコマンドを使用する前に、ファイルをバックアップすることを強くお勧めします。

### 必要な接続

なし

### コマンド構文



## コマンド・パラメーター

### **--querypath**

バイナリー内の組み込みライブラリー・パスを変更しないで照会を行うことを指定します。

### **--search | -s=***search-expression*

検索対象の式を指定します。

### **--replace | -r=***replace-expression*

*search-expression* を置換する式を指定します。

**--show** 検索および置換操作を、実際に変更をファイルに書き込むことはしないで行うことを指定します。

**--32** バイナリー形式が 32 ビットである場合に操作を実行します。

**--64** バイナリー形式が 64 ビットである場合に操作を実行します。

### **--verbose**

実行されている操作に関する情報を表示します。

**--help** 使用法情報を表示します。

### **--mask | -m**

終了値のエラー・メッセージを抑止します。このオプションは一度しか指定できません。 *mask* オプションの対象の終了値は *ignore* オプションの項目で示しています。

### **--ignore | -i**

特定のエラー・メッセージを抑止します。

*mask* オプションと *ignore* オプションの対象の終了値は以下のとおりです。

- 0 - パスは正常に変更されました
- 1 - 操作上の指定された検索と置換がすべて成功したわけではありません
- 2 - 正しいタイプのファイルでしたが *libpath* がありません。
- 3 - *libpath* を持つための正しいタイプのファイルではありませんでした
- >3 - その他のエラー

### **--skipInternalPatterns**

パターンの検索と置換のメソッドは、結果のパス上の潜在的なスペースを再利用するために内部で実行されます。このオプションは、この置換を回避するために使用します。

## 例

- *myexecutable* という名前の実行可能ファイル内の組み込みランタイム・ライブラリー検索パス値を、*/usr/opt/db2\_08\_01/lib* から */u/usr1/sql1lib/lib32* に変更するには、以下を発行します。

```
db2chglibpath --search=/usr/opt/db2_08_01/lib --replace=/u/usr1/sql1lib/lib32  
/mypath/myexecutable
```

新規値の長さが、元の値の長さと同じであることを注意してください。

### 使用上の注意

- このコマンドは、アプリケーションおよびルーチンをアップグレードする他の方法が使用できないか、または成功しなかった場合に、DB2 データベース・アプリケーションの実行可能ファイルおよび DB2 外部ルーチン共有ライブラリー・ファイルを更新する目的でのみ使用するものです。
- このコマンドは、DB2 サービス契約の下ではサポートされていません。これは現状のまま提供されており、そのため、このコマンドの本来の意図に反した使用、または悪意のある使用に対しては IBM は責任を負いません。
- このコマンドは、変更前に、共有ライブラリーまたは実行可能ファイルのバックアップは作成しません。このコマンドの発行前にファイルのバックアップ・コピーを作成することを強くお勧めします。

## db2chgpath - 組み込みランタイム・パスの変更

Linux および UNIX オペレーティング・システムで、関連した DB2 データベース・ライブラリーおよび実行可能ファイル内の組み込みランタイム・パスを更新するために DB2 データベース・インストーラーが使用します。DB2 データベース・インストール中にこのコマンドに関連したエラーがあった場合、IBM DB2 データベース・サポートの指示の下でコマンドを再発行できます。

**注:** Red Hat Enterprise Linux バージョン 5 (RHEL5) で DB2 データベースをインストールした後に SELinux (Security-enhanced Linux) が使用可能な場合、DB2 データベース・システムを正常に作動させるには現行リリースの DB2 データベース・インストールごとにこのコマンドを手動で実行する必要があります。詳しくは、『使用上の注意』というセクションを参照してください。

### 許可

root インストールには root ユーザー権限が必要です。非 root インストールの場合、非 root インストールを所有するユーザー ID でログオンしなければなりません。

### 必要な接続

なし

### コマンド構文

```
▶▶ db2chgpath [-d] [-f file-name] ▶▶
```

### コマンド・パラメーター

- d** デバッグ・モードをオンにします。このオプションは、DB2 データベース・サポートからの指示があった場合にのみ使用してください。
- f file-name**  
ランタイム・パスを更新するための特定のファイル名を指定します。  
*file-name* は、現行の DB2 データベース製品インストール位置のベースからの相対的なパス名を持っている必要があります。

### 例

- DB2 データベース製品インストール・パスの下のすべてのファイルを調べ、ランタイム・パス更新を実行するには、以下のコマンドを発行します。

```
DB2_installation_path/install/db2chgpath
```

- DB2\_installation\_path/lib64 ディレクトリー下の libdb2.a と呼ばれる特定のファイルのパスを更新するには、以下を発行します。

```
DB2_installation_path/install/db2chgpath -f lib64/libdb2.a
```



### 使用上の注意

RHEL5 システム上に DB2 データベース製品をインストール済みで、SELinux がアンインストールされているか、使用不可になっている場合に SELinux を使用できるようにするには、以下のステップを実行します。

- 必要に応じて、SELinux `rpms` をインストールします。
- `/etc/sysconfig/selinux` を変更して、状況を "permissive" または "enforcing" に設定します。
- マシンをリブートして、SELinux ラベルをすべてのファイルに適用します。
- **db2chgpath** を実行して、テキスト・リロケーションを使用した DB2 共有ライブラリーのロードを許可する SELinux 属性を設定します (`textrel_shlib_t`)。

## db2ckbkp - バックアップの検査

このユーティリティを使用すると、バックアップ・イメージの整合性をテストして、イメージがリストア可能かどうかを判別することができます。また、バックアップ・ヘッダーに保管されているメタデータを表示するために使用することもできます。

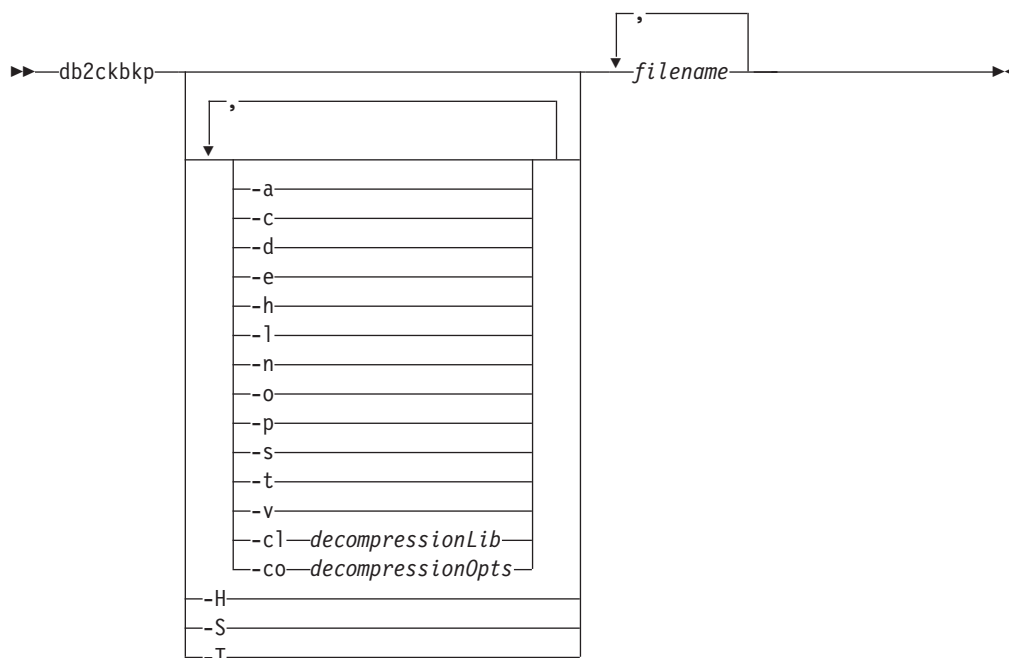
### 許可

このユーティリティにはすべてのユーザーがアクセス可能ですが、イメージ・バックアップに対してこのユーティリティを実行するには、それらの読み取り権限がなければなりません。

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

- a 使用できるすべての情報を表示します。
- c チェックビットおよびチェックサムの結果を表示します。
- d DMS 表スペース・データ・ページのヘッダーからの情報を表示します。
- e イメージからファイルにページを抽出します。ページを抽出するには、入力ファイルと出力ファイルが必要です。デフォルトの入力ファイルは `extractPage.in` という名前です。デフォルトの入力ファイル名は、**DB2LISTFILE** 環境変数に絶対パスを設定することによりオーバーライドできます。入力ファイルの形式は、次のとおりです。

SMS 表スペースの場合、

S <tbspID> <objID> <objType> <startPage> <numPages>

注:

1. <startPage> は、オブジェクト相対のオブジェクト・ページ番号です。

DMS 表スペースの場合、

D <tbspID> <objType> <startPage> <numPages>

注:

1. <objType> が必要なのは、DMS ロード・コピー・イメージを検証する場合だけです。
2. <startPage> は、プール相対のオブジェクト・ページ番号です。

ログ・ファイルの場合、

L <log num> <startPos> <numPages>

その他のデータ (初期データなど) の場合、

O <objType> <startPos> <numBytes>

デフォルトの出力ファイルは `extractPage.out` です。デフォルトの出力ファイル名は、`DB2EXTRACTFILE` 環境変数に絶対パスを設定することによりオーバーライドできます。

- h メディア・ヘッダー情報を表示します。これには、リストア・ユーティリティーで要求されるイメージの名前およびパスも含まれます。
- l ログ・ファイル・ヘッダー (LFH) およびミラー・ログ・ファイル・ヘッダー (MFH) データを表示します。
- n テープ・マウントのプロンプトを出します。1つの装置につき1つのテープが前提となります。
- o オブジェクト・ヘッダーからの詳細情報を表示します。
- p 各オブジェクト・タイプのページ数を表示します。DMS 表スペース・データに関するバックアップが実行された場合には、このオプションはさまざまなオブジェクト・タイプすべてのページ数を表示するものではありません。SQLUDMSTABLESPACEDATA として合計ページを示すだけです。SQLUDMSLOBDATA および SQLUDMSLONGDATA のオブジェクト・タイプは、DMS 表スペースではゼロです。
- s イメージ内の自動ストレージ・パスを表示します。
- t コンテナ情報など、イメージ中の表スペースに関する詳細情報を表示します。
- v DMS および AS 表スペース・ページの追加の妥当性検査を実行します。このオプションは、暗黙指定されません。また、`-a` (all) オプションでは使用可能にはなりません。

#### -c1 *decompressionLib*

解凍を実行するために使用するライブラリーの名前。この名前は、サーバー上の1個のファイルを参照する完全修飾パスでなければなりません。このパラメーターを指定しない場合、DB2 はイメージ内に格納されているライ

## db2ckbkp - バックアップの検査

ライブラリーの使用を試みます。バックアップが圧縮されていなかった場合、このパラメーターの値は無視されます。指定されたライブラリーをロードできない場合、操作は失敗します。

### **-co** *decompressionOpts*

バイナリー・データのうち、解凍ライブラリーの初期設定ルーチンに渡すブロックを記述します。DB2はこのストリングをクライアントからサーバーに直接渡すため、バイト反転やコード・ページ変換の問題がある場合は解凍ライブラリーで処理する必要があります。データ・ブロックの最初の文字が「@」である場合、DB2は、データの残りの部分をサーバー上に存在するファイルの名前と解釈します。その場合DB2は、*string*の内容をそのファイルの内容で置き換え、そのようにして得られる新しい値を初期設定ルーチンに渡します。*string*の最大長は1024バイトです。

**-H** **-h**と同じ情報を表示します。ただし、イメージの先頭から4Kメディア・ヘッダー情報のみを読み取ります。イメージの妥当性検査は実行しません。このオプションは他のオプションと併用できません。

**-S** **-s**と同じ情報を表示しますが、イメージの妥当性検査は実行しません。このオプションは他のオプションと併用できません。

**-T** **-t**と同じ情報を表示しますが、イメージの妥当性検査は実行しません。このオプションは他のオプションと併用できません。

### *filename*

バックアップ・イメージ・ファイルの名前。1つ以上のファイルを一度に検査できます。

### 注:

1. 完全バックアップが複数のオブジェクトで構成されている場合には、同時にすべてのオブジェクトを**db2ckbkp**を使用して妥当性検査する場合にのみ、妥当性検査は正常に実行できます。
2. イメージの複数の部分を検査する場合には、最初のバックアップ・イメージ・オブジェクト(.001)を最初に指定しなければなりません。

## 例

### 例 1 (UNIX オペレーティング・システムの場合)

```
db2ckbkp SAMPLE.0.krodger.DBPART000.19990817150714.001 SAMPLE.0.krodger.DBPART000.19990817150714.002
SAMPLE.0.krodger.DBPART000.19990817150714.003
```

```
[1] Buffers processed: ##
[2] Buffers processed: ##
[3] Buffers processed: ##
Image Verification Complete - successful.
```

注: "CATN####"の使用は、この形式の命名規則を使用した旧バージョンにのみ、適用可能です。

### 例 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/LOGSTREAM0000
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.

注: "CATN####" の使用は、この形式の命名規則を使用した旧バージョンにのみ、適用可能です。

例 3: 以下の例に示すのは、DB2 pureScale 環境における 3 つのメンバーのバックアップ・イメージから表示される出力例です。

```

BufAddr  MemberNum PoolID Token Type Offset FileSize ObjectSize OrigSize Object Name
-----
00000000:      0      0      0  19      0      268      268      0 "BACKUP.START.RECORD.MARKER"

numTbspsInDB : 3
numTbspsInImg : 3

Total members : 3
Member numbers: 0,1,2

```

## 使用上の注意

- 複数のセッションを使用してバックアップ・イメージを作成した場合には、**db2ckbkp** は同時にすべてのファイルを検査できます。シーケンス番号 001 のセッションが、最初に指定されるファイルであることを確認してください。
- このユーティリティーは、テープに保管されているバックアップ・イメージ (変数ブロック・サイズを指定して作成されたイメージは除く) も検査できます。これは、リストア操作の場合のようにテープを準備し、テープ装置名を指定してユーティリティーを起動することにより行えます。例えば、Linux および UNIX オペレーティング・システムでは以下のようにします。

```
db2ckbkp -h /dev/rmt0
```

## db2ckbkp - バックアップの検査

Windows では以下のようにします。

```
db2ckbkp -d ¥¥.¥tape1
```

3. イメージがテープ装置上にある場合、テープ装置パスを指定します。オプション **-n** を指定しない場合、マウント確認のプロンプトが出されます。テープが複数存在する場合、最初のテープを指定された最初の装置パスにマウントしなければなりません (これは、ヘッダー内の順序 001 のテープです)。

デフォルトでは、テープ装置が検出されるとテープのマウントを促すプロンプトが出されます。ユーザーは、プロンプトで選択します。以下は、プロンプトとオプションです。(指定された装置 I は、装置パス /dev/rmt0 上にあります)

```
Please mount the source media on device /dev/rmt0.  
Continue(c), terminate only this device(d), or abort this tool(t)?  
(c/d/t)
```

指定した装置ごとに、テープの終了時にプロンプトが出されます。

## db2cklog - アーカイブ・ログ・ファイルの妥当性の検査

データベースまたは表スペースのロールフォワード・リカバリー中にログ・ファイルを使用できるかどうかを判別するために、**db2cklog** コマンドを使用してアーカイブ・ログ・ファイルの妥当性を検査します。単一のアーカイブ・ログ・ファイル、またはある範囲のアーカイブ・ログ・ファイルを検査することができます。

DBT エラー・メッセージや警告が出されることなく **db2cklog** コマンドによる妥当性検査に合格したアーカイブ・ログ・ファイルは、ロールフォワード・リカバリー操作中に使用することができます。エラー・メッセージを伴ってアーカイブ・ログ・ファイルが妥当性検査に不合格になったり、警告が戻された場合には、そのログ・ファイルをロールフォワード・リカバリー中に使用してはなりません。

**db2cklog** コマンドによる妥当性検査中にエラーを戻したログ・ファイルは、リカバリー操作の失敗を引き起こす原因になります。ログ・ファイルの妥当性検査が警告を戻した場合には、ログ・ファイルがまだアクティブであるのでない限り、無効である可能性があります。正常に妥当性検査できるのは、アーカイブ・ログ・ファイルのように閉じられたログ・ファイルだけです。

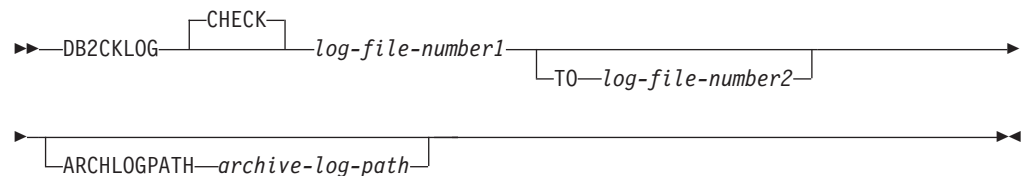
### 許可

誰でもこのコマンドを実行できますが、アーカイブ・ログ・ファイルに対する読み取りアクセス権がなければなりません。

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### CHECK

ファイルの内部妥当性の検査を実行することにより、アーカイブ・ログ・ファイル、またはある範囲のアーカイブ・ログ・ファイルの妥当性を検査します。これはデフォルトのアクションです。

#### *log-file-number1*

妥当性検査するログ・ファイルの数値 ID を指定します。例えば、S0000001.LOG ログ・ファイルの数値 ID は 1 です。TO *log-file-number2* パラメーターも指定した場合には、*log-file-number1* は検査する一定範囲のログ・ファイルにおける最初の数値 ID を表します。

#### TO *log-file-number2*

妥当性検査する番号付きログ・ファイルの範囲を指定します

## db2cklog - アーカイブ・ログ・ファイルの妥当性の検査

(*log-file-number1* から *log-file-number2* までの範囲)。 *log-file-number2* の番号が *log-file-number1* よりも小さい場合には、*log-file-number1* だけが検査されます。

### ARCHLOGPATH *archive-log-path*

アーカイブ・ログ・ファイルが格納される相対パスまたは絶対パスを指定します。デフォルト・パスは、現行ディレクトリーです。

### 例

次の例は、パス `tests` にあるアーカイブ・ログ・ファイル `S0000003.LOG` の妥当性検査の成功を示しています (出力は短縮されています)。このファイルをロールフォワード・リカバリー中に使用することができます。

```
$ db2cklog CHECK 3 ARCHLOGPATH tests
```

```
_____
          D B 2 C K L O G          _____
          DB2 Check Log File tool

...

"db2cklog": Finished processing log file "S0000003.LOG". Return code: "0".
```

次の例は、ある範囲のアーカイブ・ログ・ファイル (`S0000003.LOG` から `S0000005.LOG` まで) の妥当性検査の成功を示しています (出力は短縮されています)。各ファイルの妥当性検査の成功が出力に示されています。これらのファイルをロールフォワード・リカバリー中に使用することができます。

```
$ db2cklog 3 T0 5
```

```
_____
          D B 2 C K L O G          _____
          DB2 Check Log File tool

...

"db2cklog": Finished processing log file "S0000003.LOG". Return code: "0".

...

"db2cklog": Finished processing log file "S0000004.LOG". Return code: "0".

...

"db2cklog": Finished processing log file "S0000005.LOG". Return code: "0".
```

次の例は、ある範囲のアーカイブ・ログ・ファイルのうち、最初のログ・ファイルがエラーを戻し、妥当性検査に失敗した様子を示しています (出力は短縮されています)。いったんエラーが検出されると、DBT エラー・メッセージが戻され、**db2cklog** コマンドはそれ以降のログ・ファイルを処理せずに終了します。このログ・ファイルは、リカバリー操作が失敗する原因になるので、ロールフォワード・リカバリーに使用すべきではありません。

```
$ db2cklog 0 T0 1
```



## db2cklog - アーカイブ・ログ・ファイルの妥当性の検査

```
_____ D B 2 C K L O G _____  
DB2 Check Log File tool
```

...

DBT7053E Log file validation failed because the specified log file contains an invalid log page followed by another invalid log page.

DBT7048E The db2cklog utility determined that the current log file is invalid.

"db2cklog": Finished processing log file "S0000000.LOG". Return code: "-2000".

## db2ckrst - 増分リストア・イメージ順序の検査

データベース履歴を照会して、増分リストアに必要なバックアップ・イメージのタイム・スタンプのリストを生成します。手操作の増分リストアに使用する、単純化されたリストア構文も生成されます。

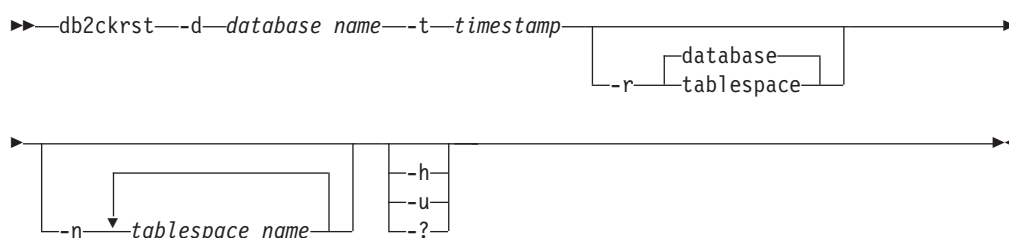
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### -d database name

リストアされるデータベースの別名を指定します。

#### -t timestamp

増分をリストアするバックアップ・イメージのタイム・スタンプを指定します。

**-r** 実行するリストアのタイプを指定します。デフォルトはデータベースです。  
**tablespace** を選択していながら表スペース名を指定しなかった場合、ユーティリティーは指定のイメージの履歴項目内を探索して、リストアを実行するためにリストされた表スペース名を使用します。

#### -n tablespace name

リストアされる 1 つ以上の表スペースの名前を指定します。データベース・リストア・タイプを選択して、表スペース名のリストを指定した場合、ユーティリティーは指定の表スペース名を使用して表スペース・リストアを続行します。

#### -h | -u | -?

ヘルプ情報を表示します。このオプションを指定すると、他のすべてのオプションは無視され、ヘルプ情報だけが表示されます。

### 例

```

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
```

```
Backup successful. The timestamp for this backup image is : 20001016001426
```

```

> db2 backup db mr incremental

Backup successful. The timestamp for this backup image is : 20001016001445

> 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
=====

```

## 使用上の注意

**db2ckrst** ユーティリティーの用途をデータベース再作成に拡張することはできません。履歴ファイルの制約のため、このユーティリティーは、複数のイメージから複数の表スペースをリストアする必要がある場合には正しいリストを提供できません。

このユーティリティーを使用するためには、データベース履歴が存在していなければなりません。データベース履歴が存在しない場合は、このユーティリティーを使用する前に、**RESTORE** コマンドで **HISTORY FILE** オプションを指定してください。

**PRUNE HISTORY** コマンドの **FORCE** オプションが使用されている場合、データベースの自動増分リストアに必要な項目を削除してしまう可能性があります。その場合でも手動リストアは正常に動作します。また、このコマンドを使用すると、**db2ckrst** ユーティリティーが、必要なバックアップ・イメージの完全なチェーンを正しく分析できなくなる可能性があります。**PRUNE HISTORY** コマンドのデフォルトの操作では、必要な項目を削除しないようになっています。**PRUNE HISTORY** コマンドの **FORCE** オプションは使用しないことをお勧めします。

このユーティリティーは、バックアップを記録するための代替手段として使用してはなりません。

## db2ckupgrade - アップグレードのためのデータベースの検査

データベースがマイグレーション可能であることを検査します。

### 有効範囲

パーティション・データベース環境では、**db2ckupgrade** コマンドは、各データベース・パーティションを検査します。

### 許可

SYSADM

### 必要な接続

なし

### コマンド構文

```
db2ckupgrade database [-e] [-l filename] [-not1] [-u userid] [-p password]
```

### コマンド・パラメーター

*database*

スキャンするローカル・データベースの別名を指定します。

**-e** スキャン対象のローカルにカタログ作成されたデータベースをすべて指定します。

**-l filename**

データベースのスキャンで生成されたエラーおよび警告のリストを維持するログ・ファイルを指定します。

**-not1** タイプ 1 索引の検査を使用不可にします。このオプションが指定されていない場合、**db2ckupgrade** コマンドはタイプ 1 索引を検査し、`type1_index_database-name.db2` スクリプト・ファイルを、ログ・ファイルに示されたのと同じディレクトリー内に作成します。このスクリプト・ファイルには、識別されたそれぞれのタイプ 1 索引に対する **ALLOW WRITE ACCESS** および **CONVERT** 節が指定された **REORG INDEXES ALL** コマンドが含まれます。

**-u userid**

システム管理者のユーザー ID を指定します。

**-p password**

システム管理者のユーザー ID のパスワードを指定します。

### 使用上の注意

**db2ckupgrade** コマンドを実行するには、以下のようにします。

## db2ckupgrade - アップグレードのためのデータベースの検査

- Linux および UNIX オペレーティング・システムでは、アップグレード先にする新規 DB2 コピーをインストールします。その後、*DB2DIR/bin* ディレクトリーから **db2ckupgrade** コマンドを実行します。ここで、*DB2DIR* は DB2 コピーがインストールされる場所です。
- Windows オペレーティング・システムでは、アップグレード先にする DB2 データベース製品 CD を挿入します。それから **db2ckupgrade** コマンドを、CD 上の *db2\Windows\Utilities* ディレクトリーから実行します。

**db2ckupgrade** コマンドは、リモート・データベースとしてカタログされているデータベースに対して実行できません。

このコマンドは、以下のすべての条件が真であることを検査します。

- カタログされたデータベースが実際に存在している。
- データベースが不整合な状態ではない。
- データベースがバックアップ・ペンディング状態ではない。
- データベースがリストア・ペンディング状態ではない。
- データベースがロールフォワード・ペンディング状態ではない。
- 表がロード・ペンディング状態ではない。
- 表が再配分ペンディング状態ではない。
- (バージョン 9.8 以降の場合) すべての表スペース・コンテナ・パスに同じマウント・ポイントを使用している。
- バージョン 9.8 フィックスパック 3 以降の場合、データベースの入出力書き込み操作がサスペンドされない。また、サスペンド中でない。
- システム・ビューに依存する MQT が存在しない。
- 表スペースが正常な状態である。
- データベースに、名前が ARRAY、BINARY、CURSOR、DECFLOAT、ROW、VARBINARY、または XML の、ユーザー定義タイプ (UDT) が含まれていない。
- データベースに、組み込みの DATALINK データ・タイプが含まれていない。
- データベースに、SYSPUBLIC という名前のスキーマがない。
- データベースに、データベース・アップグレード失敗の原因となる、システム・カタログ表内の孤立行がない。
- HADR 1 次データベースとして使用可能になっているデータベースが正常に接続できる。
- HADR データベースの役割がスタンバイでない。
- SYSCATSPACE が DMS 表スペースであり、AUTORESIZE が使用不可の場合、SYSCATSPACE には合計ページのうち少なくとも 50% の未使用ページがある。
- データベースが XML Extender には使用不可である。

アップグレード・プロセスに成功するには、ローカル・データベースはこれらすべてのチェック項目にパスする必要があります。 **db2iupgrade** コマンドは、**db2ckupgrade** コマンドを **-not1** パラメーターを指定して呼び出し、**db2ckupgrade** のログ・ファイルとして *update.log* を指定します。 **db2iupgrade** 用に作成されるデフォルトのログ・ファイルは */tmp/db2ckupgrade.log.processID* です。**db2ckupgrade** コマンドで上にリストした条件のいずれかが真でないことが検出され

## db2ckupgrade - アップグレードのためのデータベースの検査

た場合、**db2iupgrade** は失敗し、DBI1205E エラー・コードを戻します。インスタンスのアップグレード前にこれらのエラーを解決する必要があります。

**db2ckupgrade** コマンドは以下のいずれかの条件に対して、**-1** パラメーターで指定されたログ・ファイルに警告メッセージを書き込みます。

- 列名、ルーチン・パラメーター名、または変数名が **NULL** と呼ばれている。
- タイプ 1 索引がデータベース内に存在している。
- ワークロード接続属性にアスタリスク (\*) が含まれている。
- データベースが DB2 WebSphere® MQ 関数に使用可能である。

Windows オペレーティング・システムでのインストール中に、DB2 コピーを「既存の処理」ウィンドウで「アップグレード」アクションを指定して選択した場合、インスタンスにカタログされたローカル・データベースがあれば、メッセージ・ボックスによって **db2ckupgrade** コマンドを DB2 データベース製品 CD から実行しなければならないことが警告されます。それから、以下のいずれかのアクションを選択できます。

- メッセージを無視し、インストール・プロセスを続行する。
- **db2ckupgrade** コマンドを実行する。このコマンドが正常に実行された場合、インストール・プロセスを続行します。エラーを検出した場合は、インストール・プロセスを終了し、エラーを修正し、インストール・プロセスを再実行します。
- インストール・プロセスを終了します。

バージョン 9.5 データベースの場合、**-not1** パラメーターが省略されると、**db2ckupgrade** コマンドは **db2IdentifyType1** コマンドを呼び出してタイプ 1 索引を識別し、指定のデータベースのタイプ 1 索引をタイプ 2 索引に変換するスクリプトを生成します。**db2IdentifyType1** コマンドは、処理を完了するのに長い時間がかかる可能性があります。**db2IdentifyType1** コマンドの実行時間は、データベース内の表の数、およびデータベース・パーティション数に比例します。以下のパフォーマンスの考慮事項に注意してください。

- バージョン 9.5 データベースに多数の表がある場合、またはデータベース・パーティションの数が多い場合、あるいはその両方が当てはまる場合は、最初に **db2IdentifyType1** コマンドを特定のスキーマまたは表 (パラメーター **-s** または **-t** を使用) に対して実行します。これをすべての表を処理するまで行います。その後、**db2ckupgrade** コマンドに **-not1** パラメーターを指定して実行してください。
- バージョン 9.5 パーティション・データベース環境の場合、すべてのタイプ 1 索引を検出するには、1 つのデータベース・パーティション (より良いパフォーマンスを得るにはカタログ・パーティションが望ましい) から **db2ckupgrade** コマンドを実行します。

データベースがアップグレードの準備ができていることを確認するには、「DB2 バージョン 10.1 へのアップグレード」の『データベースがマイグレーションできる状態にあることの確認』を参照してください。

## db2cli - DB2 対話機能 CLI

コール・レベル・インターフェース (CLI) 環境での設計およびプロトタイピングのために、対話式 CLI 環境を開始します。

### 許可

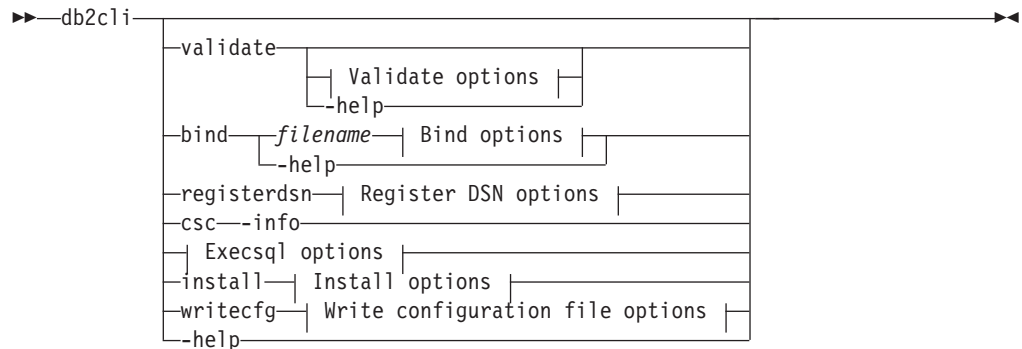
**db2cli** コマンドは、このドライバーをインストールしたディレクトリーの `bin` サブディレクトリーに配置されます。UNIX オペレーティング・システムの IBM Data Server Client では、このコマンドは `home_dir/sqlllib/bin` ディレクトリーに配置されます。`home_dir` はインスタンス所有者のホーム・ディレクトリーです。Windows オペレーティング・システムでは、このコマンドは `DB2PATH\bin` ディレクトリーに配置されます。`DB2PATH` は DB2 コピーのインストール場所です。

なし

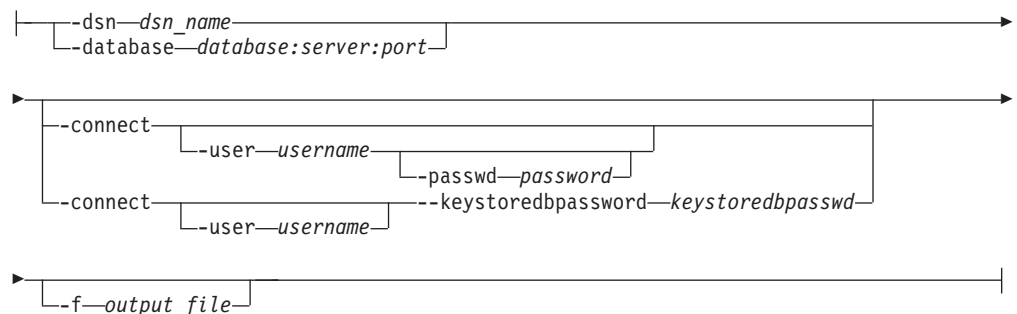
### 必要な接続

なし

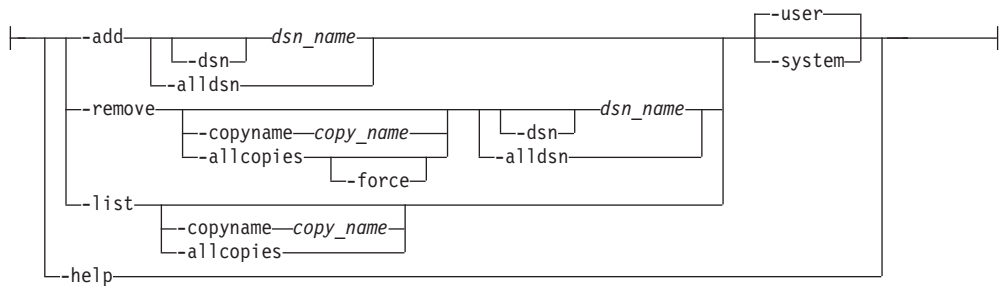
### コマンド構文



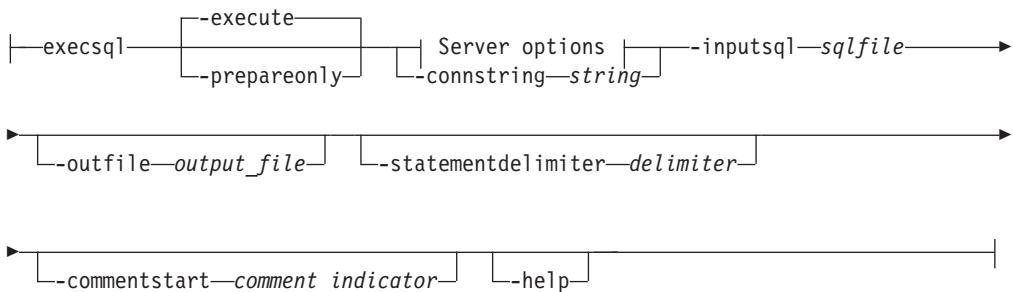
#### Validate options:



#### Register DSN options:



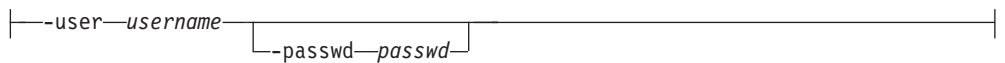
**Execsql options:**



**Server options:**



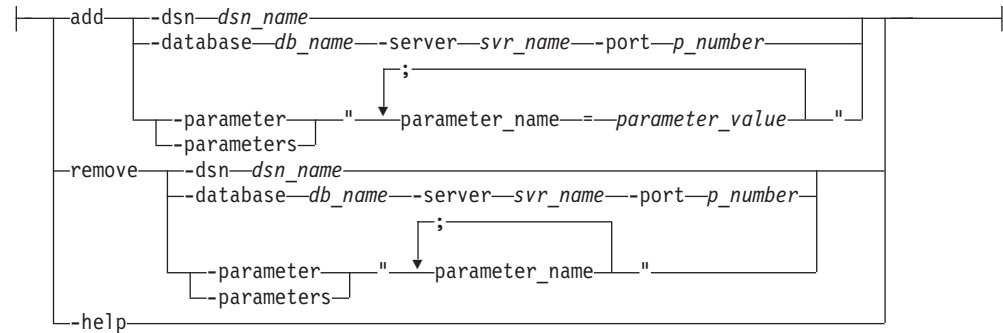
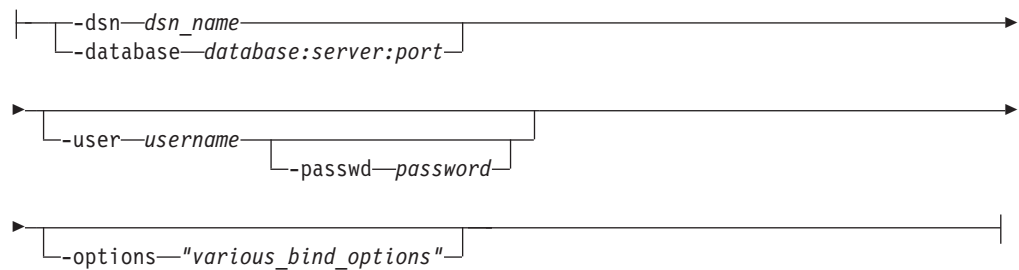
**Authentication options:**



**Install options (Windows でのみ使用可能):**





**Write configuration file options:****Bind options:****コマンド・パラメーター****validate**

CLI 環境構成の妥当性検査とテストを行います。このオプションにより、db2cli.ini ファイルおよび db2dsdriver.cfg ファイルで検出されたキーワードのリストが表示されます。特定のデータ・ソースまたはデータベース名に関して無効なキーワードがある場合、UNKNOWN としてリストされます。

IBM Data Server Client (IBM Data Server Client または IBM Data Server Runtime Client) では、**db2cli validate** コマンドは、Windows オペレーティング・システム上にインストール済みの DB2 クライアント・パッケージをリストします。IBM Data Server Client から **db2cli validate** コマンドを発行する場合、このリストは Windows オペレーティング・システム上にインストールされた IBM Data Server Client に限定されます。Windows オペレーティング・システム上にインストールされた IBM Data Server Driver for ODBC and CLI パッケージおよび IBM データ・サーバー・クライアント・パッケージをリストするには、**db2cli validate** コマンドを IBM Data Server Driver for ODBC and CLI インストール済み環境から発行する必要があります。リストでは、現行コピー名が [C]、デフォルトのコピー名が [D] で表されます。

**-dsn dsn\_name**

妥当性検査を行うデータ・ソース名 (DSN) を指定します。

**-database database:server:port**

妥当性検査を行うデータベース名、サーバー名、ポート番号を指定します。データベース名、サーバー名、およびポート番号はコロンで区切る必要があります。サーバー名は、完全修飾ドメイン名

(FQDN) で構成されている必要があります (例:  
TESTDB:dbserver.example.com:19677)。

**-connect**

**db2cli validate** コマンドの接続先となり、コマンド出力への接続試行に関する情報の書き込み先でもある DSN またはデータベースを指定します。

**-user *username***

接続に使用するユーザー名を指定します。

**-passwd *password***

接続に使用するパスワードを指定します。

**-keystoredbpassword *keystoredbpasswd***

**Authentication** パラメーターが CERTIFICATE に設定されている場合に、SSL 接続のパスワードを指定します。

**制約事項:** クライアントのバージョンよりも低位のバージョンのゲートウェイに接続する場合、そのクライアント/サーバーの組み合わせでサポートされている新フィーチャーを利用できない可能性があります。

**-f *OUTPUT\_FILE***

**db2cli** コマンドの結果用の出力ファイルを指定します (オプション)。

**-help *validate*** パラメーターの使用法に関連したヘルプ情報を表示します。

**registerdsn**

**db2cli** DSN 登録モードを指定します。このコマンド・パラメーターは、Windows オペレーティング・システムでの DSN を登録する場合に使用します。

**-add *dsn\_name*-alldsn**

Microsoft ODBC データ ソース アドミニストレーターに、システムまたはユーザーの ODBC データ・ソースを追加します。

***dsn\_name***

登録する DSN を指定します。*dsn\_name* の値は、db2cli.ini ファイルまたは db2dsdriver.cfg ファイルに定義されている DSN である必要があります、DBALIAS 名にすることはできません。**-add *dsn\_name*** パラメーターを使用すると、ローカル・データベース・ディレクトリー内にあるカタログ済みのデータベースを表すデータ・ソースを追加できます。*dsn\_name* では、db2cli.ini ファイルまたは db2dsdriver.cfg ファイルで定義されたデータ・ソースを指定することも、IBM Data Server Client のローカル・データベース・ディレクトリー内のデータベース別名を指定することもできます。

**-dsn** Microsoft ODBC データ ソース アドミニストレーターに追加する必要がある DSN 名を指定できるようにします。

**db2cli register -add -dsn** コマンドは、Windows プラットフォーム上の DB2 バージョン 10 フィックスパック 1 以降で使用可能です。

**-alldsn** db2cli.ini ファイルと db2dsdriver.cfg ファイルで定義されたすべてのデータ・ソースを登録します。このパラメーターは、**-add** パラメーターと組み合わせて使用する必要があります。

**registerdsn -add -alldsn** パラメーターを使用する場合には、db2cli.ini ファイルと db2dsdriver.cfg ファイルで使用可能なデータ・ソースのみを検索し、カタログされたデータベースのデータ・ソースは追加しません。

**-user** データ・ソースをユーザー ODBC データ・ソースとして登録します。パラメーターを指定しないと、データ・ソースはユーザー ODBC データ・ソースとして登録されます。

**-system**

データ・ソースをシステム ODBC データ・ソースとして登録します。

**-remove dsn\_name**

Microsoft ODBC データ ソース アドミニストレーターから、システムまたはユーザーの ODBC データ・ソースを削除します。

**-alldsn** 指定した DB2 コピーに含まれているすべてのユーザーまたはシステム ODBC データ・ソース項目を Microsoft ODBC データ ソース アドミニストレーターから削除します。

**db2cli register -remove -alldsn** コマンドは、Windows オペレーティング・システム上の DB2 バージョン 10 フィックスパック 1 以降で使用可能です。

**-copyname copy\_name**

指定した DB2 コピーに含まれているユーザーまたはシステム ODBC データ・ソース項目を Microsoft ODBC データ ソース アドミニストレーターから削除します。コピー名を指定しない場合、現行コピー名が使用されます。現行コピー名とは、**db2cli** ユーティリティが属しているクライアントのコピー名です。 **-copyname** オプションを **-alldsn** オプションと一緒に指定すると、Microsoft ODBC データ ソース アドミニストレーターのすべてのデータ・ソース項目が、指定した DB2 コピーから削除されます。**db2cli**

**register -remove -copyname** コマンドは、Windows オペレーティング・システム上の DB2 バージョン 10 フィックスパック 1 以降で使用可能です。

**-allcopies**

すべての DB2 コピーに含まれているユーザーまたはシステム ODBC データ・ソース項目をすべて Microsoft ODBC データ ソース アドミニストレーターから削除します。

**-force** オプションも指定している場合を除き、すべての ODBC データ・ソース項目を削除するかどうかを確認する

プロンプトが出されず、**db2cli register -remove -allcopies** コマンドは、Windows オペレーティング・システム上の DB2 バージョン 10 フィックスパック 1 以降で使用可能です。

- force** 確認を求めるプロンプトを出さずに、ユーザーまたはシステム ODBC データ・ソース項目をすべて削除します。**-force** オプションは、**-allcopies** オプションを指定する場合のみ使用できます。**db2cli register -remove -allcopies -force** コマンドは、Windows オペレーティング・システム上の DB2 バージョン 10 フィックスパック 1 以降で使用可能です。
- dsn** Microsoft ODBC データ ソース アドミニストレーターから削除する DSN 名を指定します。**db2cli register -remove -dsn** コマンドは、Windows オペレーティング・システム上の DB2 バージョン 10 フィックスパック 1 以降で使用可能です。デフォルト・オプションは **-user** であり、ユーザー DSN が削除されます。システム DSN を削除する場合は、**-system** オプションを指定する必要があります。
- list** Microsoft ODBC データ ソース アドミニストレーターに登録されているシステムまたはユーザーの IBM Data Server ODBC データ・ソースをすべてリストします。

**-copyname** *copy\_name*

Microsoft ODBC データ ソース アドミニストレーター内の指定した DB2 コピーに含まれているユーザーまたはシステム ODBC データ・ソース項目をリストします。コピー名を指定しない場合、現行コピー名が使用されます。現行コピー名とは、**db2cli** ユーティリティーが属しているクライアントのコピー名です。**db2cli register -list -copyname** コマンドは、Windows オペレーティング・システム上の DB2 バージョン 10 フィックスパック 1 以降で使用可能です。

**-allcopies**

Microsoft ODBC データ ソース アドミニストレーター内のすべての DB2 コピーに含まれている、ユーザーまたはシステム ODBC データ・ソース項目をすべてリストします。

**db2cli register -list -allcopies** コマンドは、Windows オペレーティング・システム上の DB2 バージョン 10 フィックスパック 1 以降で使用可能です。

- help** **registerdsn** パラメーターの使用法に関連したヘルプ情報を表示します。

**注:** 64 ビットの Windows マシン上で 32 ビットの ODBC データ・ソース項目を追加、削除、またはリストする場合は、**db2cli32** コマンドを使用してください。

**csc-info**

接続スーパーバイザー・クライアント (CSC) 情報を表示します。CSC は、エンドツーエンドのモニターを行うために CLI が使用する動的にロード可能なライブラリーです。

**-execsql**

入力ファイルで指定した SQL ステートメントを実行または準備します。また、ファイルに出力を保存することもできます。

**-execute**

SQL スクリプト・ファイル内の SQL ステートメントを準備して実行するように指定します。パラメーターを指定しない場合は、これがデフォルトになります。結果がコンソールに表示されます。ファイルに出力を保存するには、ファイルの絶対パスか相対パスとともにオプション **-output** を指定します。SQL スクリプト・ファイル内の SQL ステートメントに、パラメーター・マーカーを含めることはできません。

**-prepareonly**

**-inputSql** オプションで指定したファイル内の SQL ステートメントを準備するものの実行はしないように指定します。このオプションは、ステートメントを実行しないで SQL ステートメントの構文を検査する場合に使用します。

**-commentstart** *comment\_indicator*

コメント行を表す、行の先頭に表示する文字の組み合わせを指定します。*comment\_indicator* のデフォルト値は、2 つのダッシュ (--) です。コメントが複数行にまたがる場合、各行を *comment\_indicator* 文字の組み合わせで開始します。コメント行の最大長は 128 文字です。入力 SQL ファイルでは、コメント文字の後のテキストにステートメント・カーソルの属性を含めることができます。カーソル属性は、コメント直後の SQL ステートメントに適用されます。

**-connstring** *string*

ターゲット・データベースのデータベース名、サーバー、ポート番号を指定します。この情報を指定する場合は、データベース名、サーバー、ポート番号をコロンで区切る必要があります。例えば、**TESTDB:dbserver.example.com:19677** とします。**TESTDB** がデータベース名、**dbserver.example.com** がサーバー、**19677** がポート番号です。

**-dsn** *dsn\_name*

データ・ソース名を指定します。

**-help** 使用法情報の要約を表示します。

**-inputsql** *sqlfile*

SQL ステートメントが入っている入力ファイルを指定します。*sqlfile* の値は、ファイルの絶対パスか相対パスになります。このファイル内の SQL ステートメントは、区切り文字で区切られます。準備可能なステートメントのみが使用されます。**DESCRIBE TABLE**、**BIND** などの DB2 コマンドは使用できません。

入力 SQL ファイルに含めることができるのは、SQL ステートメントとコメントだけです。このファイルに、CLI 固有の属性やキーワードを含めることはできません。バッチ SQL ステートメントには対応していません。

**-outfile** *outputfile*

出力結果を格納するファイルの絶対パスまたは相対パスを指定します。このオプションを指定しない場合、結果はコンソールに表示されます。

**-passwd** *password*

データベースで認証を行うために使用するパスワードを指定します。

**-statementdelimiter** *delimiter*

ステートメントを分離するために使用する、入力 SQL ファイルで用いる文字の組み合わせを指定します。区切り文字のデフォルト値は、復帰文字です。

**-user** *username*

データベースで認証を行うユーザーを指定します。

## install

Windows レジストリーで、IBM Data Server Driver for ODBC and CLI の登録または登録解除を行います。

**-setup** 現行の IBM Data Server Driver for ODBC and CLI を、ODBC のもとで Windows レジストリーに登録します。また、アプリケーション・データ・パスに、必要な構成フォルダーとサンプル構成ファイルを作成します。

**-cleanup**

現行の IBM Data Server Driver for ODBC and CLI を、ODBC のもとで Windows レジストリーから登録解除します。その後、このクリーンアップは、現行のインストール環境のアプリケーション・データ・パスに作成されたフォルダー、構成サンプル・ファイル、ライセンス管理ファイルを削除します。 **-force** オプションを指定しない限り、ユーザー作成のファイルやフォルダーは削除されません。

**-force** **-cleanup** オプションと一緒に指定すると、インストール専用フォルダー全体がアプリケーション・データ・パスから削除されます。このフォルダーは、このインストールだけに属するフォルダーです。このフォルダー内にユーザー作成のファイルやフォルダーがある場合、プロンプトも警告も表示されずに削除されます。

**-help** **install** パラメーターの使用法に関連したヘルプ情報を表示します。

## writecfg addlremove

db2dsdriver.cfg 構成ファイルを更新します。

**add -dsn| -database | -parameter[s]**

データ・ソース、データベース、またはパラメーターに関する情報を db2dsdriver.cfg 構成ファイルに追加します。

**-dsn dsn\_name**

既存のデータベース・エレメントに追加する新規 DSN、またはパラメーター・エレメントを追加したり更新したりする既存の DSN を指定します。

構成ファイル内に *dsn\_name* が指定された *dsn* エレメントがない場合、データベース接続情報をユーザーが指定する必要があります。 *dsncollection* セクションに新しい *dsn* エレメントが、データベース用に追加されます。ユーザーがデータベース接続情報を指定しないと、SQL1531N エラー・メッセージが戻ります。

構成ファイル内に *dsn\_name* が指定された *dsn* エレメントが存在する場合、**-database** パラメーターを使用してデータベース接続情報を指定する必要があります。この接続情報は、構成ファイル内の既存の接続情報と一致しなければなりません。一致していないと、SQL1532N エラー・メッセージが戻ります。

**-database db\_name-server svr\_name-port port\_number**

データベース・エレメントの接続情報を指定します。この情報は、データベース名、データベースがあるサーバー名、データベース・サーバーのポート番号で構成されます。

db2dsdriver.cfg 構成ファイルに接続情報が同じデータベース・エレメントが存在すると、指定したパラメーター・エレメントがそのデータベース・エレメントに追加されます。その他の場合には、指定したパラメーター・エレメントは新しいデータベース・エレメントに追加されます。

**-parameter[s] “<par1=par1\_val[;...;parN=parN\_val]”**

db2dsdriver.cfg 構成ファイルに指定したデータベースまたはデータ・ソースのパラメーター・エレメントを追加または更新するためのパラメーター情報を指定します。

パラメーターをグローバル・セクションに追加するには、データベースまたはデータ・ソースを指定しないで、パラメーター情報を指定します。

構成ファイル内の指定のデータ・ソースまたはデータベースにパラメーター・エレメントがない場合、その指定のデータ・ソースまたはデータベースにパラメーター・エレメントが追加されます。

構成ファイル内の指定のデータ・ソースまたはデータベースにパラメーター・エレメントがある場合に

は、**-parameter[s]** で指定した値によって既存の値が更新されます。



値の中のパス区切り文字などの特殊文字は、「¥」エスケープ文字を前に置く必要があります。

**db2cli writecfg** コマンドは、構文検査も、指定のパラメータの値の妥当性検査も行いません。

#### **remove -dsn | -database | -parameter[s]**

データ・ソース、データベース、パラメータに関する情報を db2dsdriver.cfg 構成ファイルから削除します。

##### **-dsn** *dsn\_name*

パラメータ情報を削除する DSN、または構成ファイル内のデータ・ソース・セクション全体を削除するデータ・ソース名を指定します。

パラメータ情報を削除するには、対応する DSN とパラメータ情報を指定します。

データ・ソース・セクション全体を削除するには、パラメータ情報を指定しないで、DSN だけを指定します。

##### **-database** *db\_name-server svr\_name-port port\_number*

パラメータ情報を削除するデータベースの接続情報、または構成ファイル内のデータベース・セクション全体を削除するデータベースの接続情報を指定します。

パラメータ情報を削除するには、対応するデータベース・セクションのデータベース接続情報とパラメータ情報を指定します。

データベース・セクション全体を削除するには、パラメータ情報を指定しないで、データベース接続情報のみを指定します。

##### **-parameter[s]** “*par1[;par2;...;parN]*”

db2dsdriver.cfg 構成ファイルにある指定のデータベースまたはデータ・ソースから削除するパラメータ情報を指定します。

構成ファイルの指定のデータベース・セクションまたはデータ・ソース・セクションに、指示したパラメータがない場合、アクションは何も実行されません。

値ストリングの中のパス区切り文字などの特殊文字は、「¥」文字を前に置く必要があります。

**-help writecfg** パラメータの使用法に関連したヘルプ情報を表示します。

## **bind**

指定した *filename* をターゲット・データベースに対してバインドします。  
**db2cli bind** コマンドは、最初に **db2cli bind** コマンド・パラメータで提供される情報を使用してターゲット・データベースに接続し、バインド・アクションを実行します。バインド・ファイルの拡張子は *.bnd* です。

##### *filename*

*filename* には、バインド・ファイル、または複数のバインド・ファ



イルの名前を含むリスト・ファイルの名前を指定できます。リスト・ファイルを指定する場合、リスト・ファイル名の先頭に @ 文字を付ける必要があります。完全修飾バインド・ファイル名は二重引用符内に指定できます。

**-dsn** *dsn\_name*

バインド先のデータ・ソース名 (DSN) を指定します。このバインド・オプションを使用するときには、ターゲット・データベースに接続するために必要なすべての情報 (有効なユーザー ID とパスワードを除く) を以下に指定する必要があります。

- db2cli.ini ファイルのデータ・ソース名セクション。
- db2dsdriver.cfg ファイルのデータベース・セクションまたは DSN セクション。
- ローカル・カタログ項目。

**-database** *database:server:port*

バインド先のデータベース名、サーバー名、およびポート番号を指定します。データベース名、サーバー名、およびポート番号はコロンで区切る必要があります。サーバー名は、完全修飾ドメイン名 (FQDN) で構成されている必要があります (例: TESTDB:dbserver.example.com:19677)。

**-user** *username*

接続に使用するユーザー名を指定します。

**-passwd** *password*

接続に使用するパスワードを指定します。

**-options** "*various\_bind\_options*"

バインド・オプションおよびその値を指定します。使用可能なバインド・オプションは、次のとおりです。

- BLOCKING
- COLLECTION
- ENCODING
- GENERIC
- ISOLATION
- KEEP DYNAMIC
- REOPT

**-help** **bind** パラメーターの使用法に関連したヘルプ情報を表示します。

注: db2cli コマンドの **bind** オプションは、バージョン 10.1 フィックスパック 2 以降のフィックスパックで使用可能です。

**-help** **db2cli** コマンドのヘルプ情報を表示します。

## 使用上の注意

対話式 CLI インターフェースは、CLI 関数呼び出しの設計、プロトタイプ作成、およびテストに使用できる一連のコマンドで構成されています。このツールはプログラマーの便宜を図ったテスト用ツールであり、ツールのパフォーマンスに関して

は IBM の保証はありません。このインターフェースは、ユーザーを対象にはいないので、広範囲にわたるエラー・チェック機能はありません。

以下の 3 タイプのコマンドがサポートされています。

### CLI コマンド

CLI がサポートする各関数呼び出しに対応している (同じ名前の) コマンド。

### サポート・コマンド

CLI 関数には等価なものがないコマンド。

### 追加モード

**db2cli** コマンドの追加モードを使用して、CLI 環境構成の妥当性検査およびテストを行うことができます。

コマンドは対話式に発行することも、またはファイル内から発行することもできます。同様に、コマンドの出力は、端末に表示することもファイルに書き込むこともできます。IBM Data Server Driver for ODBC and CLI の便利なフィーチャーの 1 つに、セッション中に入力されたコマンドをすべてキャプチャーし、それをファイルに書き込むことによってコマンド・スクリプトを作成する機能があります。このスクリプトは後で再実行することができます。

Windows 64 ビットのオペレーティング・システム上の IBM Data Server Client パッケージの場合、64 ビット・バージョンの **db2cli** コマンドのほかに、32 ビット・バージョンの **db2cli** (**db2cli32.exe**) もサポートされています。

SQL ステートメントは、**SQLExecDirect()** 関数を使用して実行されます。SQL ステートメントの実行時に、**db2cli execsql** コマンドでは、**db2cli.ini** ファイルと **db2dsdriver.cfg** ファイルで指定されたデータベース設定値が使用されます。

**db2cli** コマンドによって戻されるエラー・メッセージは、**SQLGetDiagRec()** 関数と同じフォーマットでフォーマット設定されます。

## 例

### **db2cli validate**

以下の例では、ユーティリティーは、**db2cli.ini** ファイルの **[COMMON]** セクション、**db2cli.ini** ファイルにある **sample** DSN 名の **[DSN]** セクション、**db2dsdriver.cfg** ファイルにある **sample** DSN 名の **<dsn>** エlement を読み取ります。有効なキーワードと無効なキーワードの両方が表示されます。無効なキーワードは、**UNKNOWN** としてリストされます。

```
db2cli validate -dsn sample
```

次の例では、ユーティリティーは、**db2cli.ini** ファイルの **[COMMON]** セクション、データベース **dbname1** の **<database>** セクション、**db2dsdriver.cfg** ファイルにあるサーバー **server1.net1.com** とポート **50001** を読み取ります。有効なキーワードと無効なキーワードの両方が表示されます。無効なキーワードは、**UNKNOWN** としてリストされます。

```
db2cli validate -database dbname1:server1.net1.com:50001
```

**db2cli writectfg**

このセクションに記載されている例では、db2dsdriver.cfg ファイルの内容は以下のとおりであると想定します。

```
<configuration>
  <dsncollection>
    <dsn alias="alias1" name="name1" host="server1.net1.com" port="50001">
      <parameter name="DisableAutoCommit" value="TRUE"/>
    </dsn>
  </dsncollection>
  <databases>
    <database name="name1" host="server1.net1.com" port="50001">
      <parameter name="CurrentSchema" value="OWNER1"/>
    </database>
  </databases>
  <parameters>
    <parameter name="IsolationLevel" value="SQL_TXN_READ_COMMITTED"/>
  </parameters>
</configuration>
```

以下の例では、db2dsdriver.cfg 構成ファイルに新しいデータ・ソース・エレメントが追加されます。

```
$db2cli writectfg add -dsn alias2 -database name2 -server server1.net1.com -port 50001
```

このコマンドの結果、*dsncollection* セクションは以下のように変更されます。

```
<dsncollection>
  <dsn alias="alias1" name="name1" host="server1.net1.com" port="50001">
    <parameter name="DisableAutoCommit" value="TRUE"/>
  </dsn>
  <dsn alias="alias2" name="name2" host="server1.net1.com" port="50001"/>
</dsncollection>
```

以下の例では、構成ファイル内の既存のデータ・ソースにパラメーター情報が追加されます。

```
db2cli writectfg add -dsn alias2 -parameters "DisableAutoCommit=FALSE;CurrentSchema=OWNER2;pureQueryXml=C:¥clico"
```

このコマンドの結果、*dsncollection* セクションは以下のように変更されます。

```
<dsncollection>
  <dsn alias="alias1" name="name1" host="server1.net1.com" port="50001">
    <parameter name="DisableAutoCommit" value="TRUE"/>
  </dsn>
  <dsn alias="alias2" name="name2" host="server1.net1.com" port="50001">
    <parameter name="DisableAutoCommit" value="FALSE"/>
    <parameter name="CurrentSchema" value="OWNER2"/>
    ..<parameter name="pureQueryXml" value="C:¥clico"/>
  </dsn>
</dsncollection>
```

以下の例では、構成ファイル内のパラメーターに新しいデータベース・エレメントが追加されます。

```
db2cli writectfg add -database name2 -server server1.net1.com -port 50001 -parameters "LockTimeout=20;KeepAliveTimeout=20000"
```

このコマンドの結果、*databases* セクションは以下のように変更されます。

```
<databases>
  <database name="name1" host="server1.net1.com" port="50001">
    <parameter name="CurrentSchema" value="OWNER1"/>
  </database>
  <database name="name2" host="server1.net1.com" port="50001">
```

```

    parameter name=" LockTimeout" value="20"/>
    parameter name="KeepAliveTimeout" value="20000"/>
  </database>
</databases>

```

以下の例では、構成ファイル内の既存の *dsn* エレメントの既存のパラメーターが変更されます。

```
db2cli writecfg add -dsn alias1 -parameter "DisableAutoCommit=FALSE"
```

このコマンドの結果、*dsncollection* セクションは以下のように変更されます。

```

<dsncollection>
  <dsn alias="alias1" name="name1" host="server1.net1.com" port="50001">
    <parameter name="DisableAutoCommit" value="FALSE"/>
  </dsn>
  <dsn alias="alias2" name="name2" host="server1.net1.com" port="50001">
    <parameter name="DisableAutoCommit" value="FALSE"/>
    <parameter name="CurrentSchema" value="OWNER2"/>
    ..<parameter name="pureQueryXml" value="C:¥clico"/>
  </dsn>
</dsncollection>

```

以下の例では、構成ファイル内のグローバル・セクションにパラメーター・エレメントが追加されます。

```
db2cli writecfg add -parameter "RecieveTimeout=20000"
```

このコマンドの結果、グローバル・セクションが以下のように変更されます。

```

<parameters>
  <parameter name="IsolationLevel" value=" SQL_TXN_READ_COMMITTED"/>
  <parameter name="RecieveTimeout" value="20000"/>
</parameters>

```

以下の例では、構成ファイル内のデータベースからパラメーター・エレメントが削除されます。

```
db2cli writecfg remove -database name1 -server server1.net1.com -port 50001 -parameter "CurrentSchema"
```

このコマンドの結果、*databases* セクションは以下のように変更されます。

```

<databases>
  <database name="name1" host="server1.net1.com" port="50001">
  </database>
  <database name="name2" host="server1.net1.com" port="50001">
    parameter name="LockTimeout" value="20"/>
    parameter name="KeepAliveTimeout" value="20000"/>
  </database>
</databases>

```

以下の例では、構成ファイル内の *dsn* エレメントが削除されます。

```
db2cli writecfg remove -dsn alias1
```

このコマンドの結果、*dsncollection* セクションは以下のように変更されます。

```

<dsncollection>
  <dsn alias="alias2" name="name2" host="server1.net1.com" port="50001">
    <parameter name="DisableAutoCommit" value="FALSE"/>
    <parameter name="CurrentSchema" value="OWNER2"/>
    <parameter name="pureQueryXml" value="C:¥clico"/>
  </dsn>
</dsncollection>

```

**db2cli execsql**

以下の例では、SAMPLE データベースに以下の表およびプロシージャが作成されているとします。

```
create table employee(empid integer, empname varchar(100))

CREATE PROCEDURE proc1 ( )
DYNAMIC RESULT SETS 1 P1:
BEGIN
  DECLARE cursor1 CURSOR WITH RETURN FOR SELECT * FROM fprem;
  OPEN cursor1;
END P1

CREATE PROCEDURE PROC2(IN ID1 INTEGER,OUT NAME VARCHAR(20))
BEGIN
DECLARE CUR1 CURSOR WITH RETURN TO CALLER FOR SELECT * FROM EMPLOYEE1 WHERE ID=ID1;
  OPEN CUR1;
END
```

また、この例では、SQL ファイル test.sql に以下のテキストが含まれているとします。

```
--Populate table( employee )
insert into employee(empid, empname) values(1, 'Adam')
insert into employee(empid, empname) values(2, 'Atul')
select empid, empname from employee

--Execute the stored procedure
Call proc1( )
```

コンソール・ウィンドウに以下の **db2cli** コマンドを入力して、ファイル内の SQL ステートメントを実行します。

```
db2cli execsql -dsn sample -inputsql test.sql
```

以下のテキストが、コンソール・ウィンドウに表示されます。

```
IBM DATABASE 2 Interactive CLI Sample Program
(C) COPYRIGHT International Business Machines Corp. 1993,1996
All Rights Reserved
Licensed Materials - Property of IBM
```

```
insert into employee(empid, empname) values(1, 'Adam')
The SQL command completed successfully.
```

```
insert into employee(empid, empname) values(2, 'Atul')
The SQL command completed successfully.
```

```
select empid, empname from employee
```

```
EMPID EMPNAME
1, Adam
2, Atul
```

```
Call proc1()
```

```
EMPID EMPNAME
1, Adam
2, Atul
```

OUT 引数を持つストアド・プロシージャの CALL ステートメントを実行します。疑問符 (?) を OUT パラメーターとして使用することができます。

以下の例では、SQL スクリプト・ファイル test2.sql に以下のテキストが含まれているとします。

```
CALL PROC2( 1, ?)
```

コンソール・ウィンドウに以下の **db2cli** コマンドを入力して、ファイル内の SQL ステートメントを実行します。

```
db2cli execsql -dsn sample -inputsq1 test2.sql
```

以下のテキストが、コンソール・ウィンドウに表示されます。

```
Value of output parameters
```

```
-----
Parameter Name : NAME
Parameter Value : -
```

```
ID
```

```
-----
1
```

-prepareonly オプションを指定して、SQL ステートメントを実行せずに準備します。この SQL ステートメントに必要な DDL ステートメントは、-prepareonly オプションを指定して **db2cli execsql** コマンドを実行する前に、実行しておく必要があります。

以下の例では、SQL ファイル test3.sql に以下のテキストが含まれているとします。

```
--populate table( employee )
insert into employee(empid, empname) values(1, 'Adam');
insert into employee(empid, empname) values(2, 'Atul');
select empid, empname from employee;
```

また、表 EMPLOYEE がデータベースに作成されているとします。コンソール・ウィンドウに以下の **db2cli** コマンドを入力して、ファイル内の SQL ステートメントを準備します。

```
db2cli execsql -prepareonly -dsn sample -inputsq1 test3.sql
```

以下のテキストが、コンソール・ウィンドウに表示されます。

```
IBM DATABASE 2 Interactive CLI Sample Program
(C) COPYRIGHT International Business Machines Corp. 1993,1996
All Rights Reserved
Licensed Materials - Property of IBM
```

```
insert into employee(empid, empname) values(1, 'Adam')
The SQL command prepared successfully.
```

```
insert into employee(empid, empname) values(2, 'Atul')
The SQL command prepared successfully.
```

```
select empid, empname from employee
The SQL command prepared successfully.
```

If you place DDL statements that are required for DML statements in the same file, the DML statements that require the DDL statements fail.

For example, assume that the following text is in the file test4.sql, and assume that the EMPLOYEE table has not been created in the database:

```
--create and populate table( employee )
create table employee(empid integer, empname varchar(100));
insert into employee(empid, empname) values(1, 'Adam');
insert into employee(empid, empname) values(2, 'Atul');
select empid, empname from employee;
```

```
-- try to create another table with the same name
create table employee(empid integer, empname varchar(100));
```

INSERT および SELECT ステートメントを正常に実行するには、事前に CREATE TABLE ステートメントを実行しておく必要があります。

コンソール・ウィンドウに以下の **db2cli** コマンドを入力して、ファイル内の SQL ステートメントを準備します。

```
db2cli execsql -prepareonly -dsn sample -inputsq1 test4.sql
```

以下のテキストが、コンソール・ウィンドウに表示されます。

```
IBM DATABASE 2 Interactive CLI Sample Program
(C) COPYRIGHT International Business Machines Corp. 1993,1996
All Rights Reserved
Licensed Materials - Property of IBM

create table employee(empid integer, empname varchar(100))
The SQL command prepared successfully.

insert into employee(empid, empname) values(1, 'Adam')
The SQL command failed. During SQL processing it returned:
[IBM][CLI Driver][DB2/6000] SQL0204N "EMPLOYEE" is an undefined name.  SQLSTATE=42704

insert into employee(empid, empname) values(2, 'Atul')
The SQL command failed. During SQL processing it returned:
[IBM][CLI Driver][DB2/6000] SQL0204N "EMPLOYEE" is an undefined name.  SQLSTATE=42704

select empid, empname from employee
The SQL command failed. During SQL processing it returned:
[IBM][CLI Driver][DB2/6000] SQL0204N "EMPLOYEE" is an undefined name.  SQLSTATE=42704

create table employee(empid integer, empname varchar(100))
The SQL command prepared successfully.
```

この例では、2 つの CREATE SQL ステートメントが正常に準備されましたが、EMPLOYEE 表はデータベースに作成されませんでした。EMPLOYEE 表がデータベースに存在しなかったため、INSERT および SELECT ステートメントが正常に準備されませんでした。

### db2cli bind

以下の例では、db2cli.lst リスト・ファイルをバインドします。

```
$ db2cli bind @db2cli.lst -database "mydb:test.torolab.ibm.com:446"
  -options "BLOCKING unambig REOPT always ISOLATION RR"
LINE  MESSAGES FOR db2cli.lst
-----
      SQL0061W  The binder is in progress.
      SQL0091N  Binding was ended with "0" errors and "0" warnings.
```

## db2cmd - DB2 コマンド・ウィンドウのオープン

CLP 可能 DB2 ウィンドウをオープンし、DB2 コマンド行環境を初期化します。このコマンドを実行することは、「DB2 コマンド・ウィンドウ」アイコンをクリックすることと同じです。

このコマンドは Windows オペレーティング・システムでのみ使用できます。

### 許可

なし

### 必要な接続

なし

### コマンド構文

```
db2cmd [option-flag] [command]
```

### コマンド・パラメーター

**-c | /c** **-c** オプションの後のコマンドを新しい DB2 コマンド・ウィンドウで実行した後、終了します。例えば、**db2cmd -c dir** と指定すると、**dir** コマンドが新しい DB2 コマンド・ウィンドウで呼び出された後、DB2 コマンド・ウィンドウがクローズします。

**-w | /w**

**-w** オプションの後のコマンドを新しい DB2 コマンド・ウィンドウで実行した後、新しい DB2 コマンド・ウィンドウがクローズされるのを待機してから、処理を終了します。例えば **db2cmd /w dir** と指定すると、**dir** コマンドが呼び出されますが、新しい DB2 コマンド・ウィンドウがクローズされるまで処理は終了しません。

**-i | /i** 同じ DB2 コマンド・ウィンドウを共有しファイル・ハンドルを継承した状態で、**-i** オプションの後のコマンドを実行します。例えば **db2cmd -i dir** と指定すると、同じ DB2 コマンド・ウィンドウで **dir** コマンドが実行されます。

**-t | /t** **-t** オプションの後のコマンドを新しい DB2 CLP ウィンドウで実行します。指定されたコマンドはこの新しいウィンドウのタイトルになります。

### 使用上の注意

DB21061E (「コマンド行環境が初期化されていない。」) が、CLP が有効な DB2 ウィンドウを表示する際に戻される場合、オペレーティング・システムの環境スペースが足りない可能性があります。 **config.sys** ファイルで **SHELL** 環境セットアップ・パラメーターを調べ、状況に応じて値を増やしてください。以下に例を示します。

```
SHELL=C:¥COMMAND.COM C:¥ /P /E:32768
```



## db2cptsa - DB2 HA スクリプトのインストールまたは更新

UNIX および Linux のオペレーティング・システムにおいて、`/usr/sbin/rsct/sapolicies/db2` にある DB2 ハイ・アベイラビリティ (HA) スクリプトをインストールまたは更新します。これらの DB2 HA スクリプトは、IBM Tivoli System Automation for Multiplatforms (SA MP) を DB2 High Availability Feature とともに使用するために必要です。

### 許可

root ユーザー権限

### 必要な接続

なし

### コマンド構文

```

▶▶ db2cptsa [-c] [-f] [-r] [-h] [-?]

```

### コマンド・パラメーター

- c `/usr/sbin/rsct/sapolicies/db2` に DB2 HA スクリプトが存在していて、それらが適切なレベルであることを検証します。
- f `/usr/sbin/rsct/sapolicies/db2` 内の DB2 HA スクリプトの再インストールを強制的に実行します。この引数を指定しない場合、既にインストールされている DB2 HA スクリプトのバージョンが、これからインストールするスクリプトのバージョン以上なら、インストール済みスクリプトは上書きされません。
- r ディレクトリー `/usr/sbin/rsct/sapolicies/db2` を除去します。このディレクトリーは、SA MP の DB2 HA スクリプトのある場所です。それらのスクリプトおよびこのディレクトリーが除去されるのは、SA MP がインストールされていない場合だけです。
- h | -? ヘルプ情報を表示します。

### 使用上の注意

デフォルトでは、このユーティリティーにより DB2 HA スクリプトは、それらが `/usr/sbin/rsct/sapolicies/db2` にまだインストールされていない場合、あるいは、既にインストールされているスクリプトのバージョンがこれからインストールするスクリプトのバージョンよりも古い場合、そこにインストールされます。このユーティリティーは、SA MP が既にインストールされている場合、かつその場合に限り、DB2 HA スクリプトをインストールまたは更新します。

このコマンドは、DB2 インストール・メディアの中の `db2/plat/tsamp` ディレクトリーにあります。この `plat` は以下を表します。

- aix (DB2 for AIX 5L™ の場合)

## db2cptsa - DB2 HA スクリプトのインストールまたは更新

- linux (32 ビット AMD および Intel システム (x86) 上の DB2 for Linux の場合)
- linuxamd64 (AMD64 および Intel EM64T システム (x64) 上の DB2 for Linux の場合)
- linuxppc (POWER® (System i および pSeries®) システム上の DB2 for Linux の場合)
- linux390 (System z9® および zSeries® 上の DB2 for Linux の場合)

このコマンドは、*DB2DIR/install/tsamp* ディレクトリーにもあります (*DB2DIR* は UNIX および Linux のオペレーティング・システムにおける DB2 データベース製品のインストール・パス)

## db2dart - データベース分析およびレポート・ツール

データベースがアーキテクチャ的に正しいか調べ、エラーが発生した場合それを報告します。

**db2dart** コマンドによって生成されるレポートは、コマンドによって分析されるデータベースと同じコード・ページでエンコードされます。同様に、**db2dart** コマンドの一部として指定されるパラメーター値は、分析されるデータベースと同じコード・ページで解釈されます。**db2dart** レポート・ファイル内の文字ストリング、および入力値は、データベース・コード・ページでエンコードされます。

**db2dart** コマンドは、コード・ページ変換を実行しません。

**db2dart** コマンドを呼び出すときに指定できるアクションは、1 つだけです。アクションは、いくつかのオプションをサポートできます。

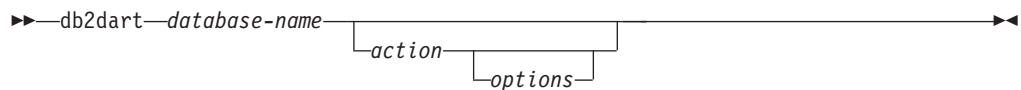
### 許可

**db2dart** コマンドを使用するには、SYSADM 権限が必要です。

### 必要な接続

なし。**db2dart** は、ユーザーがデータベースに接続していない状態で実行しなければなりません。

### コマンド構文



### コマンド・パラメーター

#### 検査アクション

- /DB** データベース全体を検査します。これはデフォルト・オプションです。
- /T** 1 つ以上の表を検査します。必須入力値は 2 つです。つまり、表スペース ID と、もう 1 つは表オブジェクト ID、表オブジェクト ID リスト、または表名のいずれかです。
- /TSF** ストレージ・グループおよび表スペース・ファイルと、コンテナーのみを検査します。
- /TSC** 1 つ以上の表スペースの表スペース構造を検査します。表の検査は行いません。必須入力値は 1 つです。つまり、表スペース ID または表スペース ID リストです。
- /TS** 1 つ以上の表スペースと、その表を検査します。必須入力値は 1 つです。つまり、表スペース ID または表スペース ID リストです。
- /ATSC** 全表スペースの構成を検査します。ただし、その表は検査しません。

#### データ・フォーマット・アクション

- /DD** フォーマット済み表データをダンプします。インライン LOB データがあれば、それも表示されます。必須入力値は、表オブジェクト ID または表名のいずれか、表スペース ID、開始ページ番号、ページ数、および冗長選択の 5 つです。
- /DI** フォーマット済み索引データをダンプします。必須入力値は、表オブジェクト ID または表名のいずれか、表スペース ID、開始ページ番号、ページ数、および冗長選択の 5 つです。
- パーティション表上の非パーティション索引の場合、**/DI** アクションは、**/OI** オプションおよび **/TSI** オプションに対する最初の 2 つの入力データとして、SYSCAT.INDEXES から INDEX\_OBJECTID と TBSpaceID を使用します。このアクションには表名 (**/TN**) オプションはサポートされていません。
  - パーティション表上のパーティション索引の場合、**/DI** アクションは SYSCAT.DATAPARTITIONS から PARTITIONOBJECTID と TBSpaceID を使用します。このアクションには表名 (**/TN**) オプションはサポートされていません。
- /DM** フォーマット済みブロック・マップ・データをダンプします。必須入力値は、表オブジェクト ID または表名のいずれか、表スペース ID、開始ページ番号、ページ数、および冗長選択の 5 つです。このデータは、マルチディメンション・クラスタリング (MDC) 表または挿入時クラスタリング表 (ITC) の空きブロックを再利用するための再編成の後に、表スペースでブロックが再利用されたかどうかを示します。
- /DP** ページを 16 進数でダンプします。
- **DMS** 表スペース内の永続オブジェクトの場合、アクション **/DP** に対して必要な入力値は、表スペース ID、開始ページ番号、およびページ数の 3 つです。
  - **SMS** 表スペース内の永続オブジェクトの場合、アクション **/DP** に対して必要な入力値は、表スペース ID、オブジェクト ID、開始ページ番号、ページ数、およびオブジェクト・タイプの 5 つです。
- /DTSF** フォーマット済み表スペースおよびストレージ・グループのファイル情報をダンプします。
- /DEMP** **DMS** 表のフォーマット済みエクステンツ・マップ・ページ (EMP) 情報をダンプします。必須入力値は、表スペース ID、および表オブジェクト ID または表名の 2 つです。
- /DDEL** 表データを区切り文字付き ASCII フォーマットでダンプします。必須入力値は、表オブジェクト ID または表名のいずれか、表スペース ID、開始ページ番号、およびページ数の 4 つです。
- ダンプされる区切り文字付き ASCII ファイルは、データベース・コード・ページでエンコードされます。 **db2dart** コマンドは、コード・ページ変換を実行しません。

**/DDEL** パラメーターは、以下の列データ・タイプだけをサポートします。他の何らかのデータ・タイプを持つ列が表に含まれている場合は、その列はスキップされ、区切り文字付き ASCII ファイルには組み込まれません。

- SMALLINT
- FLOAT
- REAL
- INTEGER
- TIME
- DECIMAL
- CHAR()
- VARCHAR()
- DATE
- TIMESTAMP
- BIGINT

タイプ CHAR および VARCHAR の列にバイナリー・データが入っている場合、またはそのような列に FOR BIT DATA が定義されている場合に、**/DDEL** パラメーターを指定すると、バイナリー・データを含む DEL ファイルが生成されます。**LOAD** コマンドを使用して、この DEL ファイルのデータを表にロードする場合は、必ず、modified by delprioritychar オプションを指定してください。**IMPORT** コマンドを使用して、この DEL ファイルのデータを表に挿入する場合は、必ず、modified by delprioritychar codepage=x オプションを指定してください。x は、入力データ・セットのデータのコード・ページです。

#### **/DHWM**

最高水準点情報をダンプします。必須入力値は、表スペース ID です。

**/DXA** フォーマット済み XML 列データを ASCII フォーマットでダンプします。必須入力値は、表オブジェクト ID または表名のいずれか、表スペース ID、開始ページ番号、ページ数、および冗長選択の 5 つです。

**/DXH** フォーマット済み XML 列データを HEX 形式でダンプします。必須入力値は、表オブジェクト ID または表名のいずれか、表スペース ID、開始ページ番号、ページ数、および冗長選択の 5 つです。

#### **/LHWM**

最高水準点を低くする方法を提案します。必須入力値は、表スペース ID およびページ数 (希望の最高水準点) の 2 つです。

#### 修復アクション

**/ETS** 可能な場合、表制限を 4 KB 表スペースに拡張します (DMS のみ)。必須入力値は、表スペース ID です。

**/MI** 索引に無効のマークを付けます。このパラメーターを指定するときは、データベースをオフラインにしなければなりません。必須入力

値は、表スペース ID および索引オブジェクト ID の 2 つです。  
パーティション索引の場合、こうした値は  
SYSCAT.INDEXPARTITIONS の INDPARTITIONOBJECTID および  
INDPARTITIONTBSPACEID から取得できます。

**/RHWM** 空の SMP エクステントを使用して、最高水準点を下げます。この  
パラメーターを指定するときは、データベースをオフラインにしな  
ければなりません。必須入力値は、表スペース ID です。

#### 状態の変更アクション

**/CHST** データベースの状態を変更します。このパラメーターを指定する  
ときは、データベースをオフラインにしなければなりません。必須入  
力値は、データベース・バックアップ・ペンディング状態です。

#### ヘルプ

**/H** ヘルプ情報を表示します。

#### 入力値オプション

##### **/OI** *object-id*

オブジェクト ID を指定します。 **/T** アクションの場合、最大 64  
個のオブジェクト ID をコンマ区切りのリストに指定できます。対  
応する **/TSI** オプションに、複数の入力 ID が入っている場合は、  
最初の ID のみが使用されます。重複する ID はスキップされま  
す。論理 ID を **/T** アクションに指定することができます。

##### **/TN** *table-name*

表名を指定します。

##### **/TSI** *tablespace-id*

表スペース ID を指定します。 **/TS** または **/TSC** アクションの場  
合、最大 64 個の物理表スペース ID をコンマ区切りのリストに指  
定できます。重複する ID はスキップされます。

##### **/ROW** *sum*

長フィールド記述子、LOB 記述子、および制御情報を検査するかと  
うか識別します。 1 つのオプションを指定することもできますし、  
値を追加して複数のオプションを指定することもできます。

- 1 行内の制御情報を検査します。
- 2 長フィールド記述子および LOB 記述子を検査します。

##### **/RPT** *path*

レポート出力ファイル用のオプションのパス。

##### **/RPTN** *file-name*

レポート出力ファイル用のオプションの名前。

##### **/PS** *number*

開始ページ番号を指定します。 **/DP** アクションと一緒に使用する場  
合、プール相対アドレッシングのために p 接尾部を使用すること  
ができます。 **/PS 0 /NP 0** を指定すると、指定されたオブジェクト中  
のすべてのページがダンプされます。

**/NP** *number*

ページ数を指定します。 /PS 0 /NP 0 を指定すると、指定されたオブジェクト中のすべてのページがダンプされます。

**/V** *option*

冗長オプションをインプリメントするかどうかを指定します。有効な値は以下のとおりです。

- Y** 冗長オプションをインプリメントすることを指定します。
- N** 冗長オプションをインプリメントしないことを指定します。

**/SCR** *option*

画面出力のタイプを指定します (もしあれば)。有効な値は以下のとおりです。

- Y** 通常の画面出力が生成されます。
- M** 最小化された画面出力が生成されます。
- N** 画面出力は生成されません。

**/RPTF** *option*

レポート・ファイル出力のタイプを指定します (もしあれば)。有効な値は以下のとおりです。

- Y** 通常のレポート・ファイル出力が生成されます。
- E** レポート・ファイルにエラー情報だけが生成されます。
- N** レポート・ファイル出力は生成されません。

**/ERR** *option*

生成するログのタイプを DART.INF に指定します (もしあれば)。有効な値は以下のとおりです。

- Y** 通常ログを DART.INF ファイルに生成します。
- N** 出力を最小化して DART.INF ファイルに記録します。
- E** DART.INF ファイルと画面出力を最小化します。レポート・ファイルにエラー情報だけが送信されます。

**/WHAT** **DBBP** *option*

データベース・バックアップ・ペンディング状態を指定します。有効な値は以下のとおりです。

- OFF** オフ状態。
- ON** オン状態。

**/QCK** *sum*

実行するクイック・オプションを指定します。オプションを 1 つ指定することもできますが、各値を加算して複数のクイック・オプションを実行することもできます。

- 1** /QCK 1 オプションは、アクション /DB、/T、および /TS のみ適用されます。このオプションは、DAT オブジェクトのページ 0 と、索引オブジェクトの一部を検査します (オブジェクト BMP、LOB、LF は検査せず、DAT オブジェク



トまたは INX オブジェクトの全体をトラバースすることはありません)。これはデフォルト・オプションです。

- 2 /QCK 2 オプションは、アクション /DB、/T、/TS、/DD、/DI、/DM、/DEMP、/DDEL、/DXA、および /DXH にのみ適用されます。このオプションは、非パーティション・データベース環境と、パーティション・データベース環境のカatalog・パーティションで、システム・Catalog表参照をスキップします。このオプションは、パーティション・データベース環境の非Catalog・パーティションには影響を与えません。  
/QCK 2 オプションは、/TN オプションに表名が指定されている場合、または /OI および /TSI オプションに論理 ID が指定されている場合は、前述のアクションに適用されません。
- 4 /QCK 4 オプションは、アクション /T、/TS、および /TSC にのみ適用されます。このオプションは、特殊システム・Catalog表検査またはシステム・Catalog表スペース検査をスキップします。アクション /TS および /TSC の場合、/QCK 4 オプションは、特殊システム・Catalog表検査をスキップします。/T アクションの場合、/QCK 4 オプションは、システム・Catalog表スペース構造の検査をスキップします。
- 8 /QCK 8 オプションは、アクション /T および /TS にのみ適用されます。このオプションは、コンテナの検査をスキップします。/T アクションの場合、/QCK 8 オプションは、すべてのコンテナ・ファイルの検査をスキップします。  
/TS アクションの場合、/QCK 8 オプションは、指定された表スペースに関連付けられているコンテナ・ファイルのみを検査します。

#### /TYP option

オブジェクトのタイプを指定します。有効な値は以下のとおりです。

**DAT** オブジェクト・タイプは DAT です。

**INX** オブジェクト・タイプは INDEX です。

**BKM** オブジェクト・タイプは BMP です。

### 例

#### 例 1

ID が 2 である表スペースにある非範囲パーティション表 (オブジェクト ID は 4) のフォーマット済み索引データを、ページ・ゼロから始めて 1000 ページ分ダンプするには、以下の **db2dart** コマンドを使用します。

```
db2dart IXMLDB /DI /TSI 2 /OI 4 /V Y /PS 0 /NP 1000
```

#### 例 2

範囲パーティション表に対するフォーマット済みブロック・マップ・データをダンプするには、以下の **db2dart** コマンドを使用します。

```
db2dart IXMLDB /DM /TSI 2 /OI 8 /V Y /PS 0 /NP 1000
```



ここで、8 は **partitionobjectid** の値で、2 は SYSCAT.DATAPARTITIONS からの **tbspaceid** の値です。

### 例 3

データベース testdb の表スペースおよびストレージ・グループ・ファイルのフォーマット済み情報をダンプするには、以下のコマンドを使用します。

```
db2dart testdb /DTSF
```

以下は、上記のコマンドで生成される出力の例です。

```
Storage group file (automatic storage) report phase start.
```

```
Header version:          33
Header flavour:          1
Checksum:                0x5402d18f
Number of storage groups: 1
Default storage group ID: 0
Header Last LSN:        0x0000000000003D4A4
```

```
Storage group ID:        0
Storage group name:      IBMSTOGRUP
Data Tag:                5
Flavour:                 3
Version:                 5
State flags:             0x0000000000000000
Last LSN:                0x0000000000003D4A4
Initial LSN:            0x0000000000000000
Checksum:                0x1e587275
```

```
Number of storage paths: 1
Storage path # 0:        /filesystem1 (id = 0, state = 0x0)
```

```
Storage group file (automatic storage) report phase end.
```

```
Tablespace file report phase start.
Tablespace information for current database:
-----
```

```
Number of defined tablespaces: 4
High water mark of used pools: 3
Number of disabled tablespaces: 0
```

```
Individual tablespace details:
-----
```

```
Information for Tablespace ID: 0
-----
```

```
Tablespace name: SYSCATSPACE
Table space flags (HEX): 3102
Table space type: Database Managed Space (DMS), Automatic Storage, Auto-Resize
Page size: 4096
Extent size: 4
Prefetch size: 4
Version: 104
Tablespace state: 0
Number of quiescers: 0
Storage Group ID: 0
Source Storage Group ID: -1
Data Tag: 0
Usable pages in tablespace: 25340
Total pages in tablespace: 25344
```

```

Initial Size: 1048576 bytes
Increment : -4096 bytes
Maximum Size: None
Last Resize: None
Last Resize Failed: No
SMP page for first free extent: 4
SMP page for last allocated tablespace extent. 4
SMP extent number of the last initialized SMP extent: 0
High Water Mark: 25164
.
.
.

```

#### 例 4

ID が 3、4、5、8、9、および 11 である表スペースを検査して、システム・カタログ表参照および特殊システム・カタログ表検査はスキップする場合は、以下の **db2dart** コマンドを使用します。指定された表スペースに関連付けられているコンテナのみが検査されます。

```
db2dart <dbname> /TS /TSI 3,4,5,6,9,11 /QCK 14
```

ここで、/QCK 14 はクイック・オプション 2、4、8 の加算を表します。  
/QCK 14 オプションは、クイック・オプション 2、4、8 のそれぞれの操作すべてを実行します。

#### 使用上の注意

- **db2dart** コマンドを呼び出すときに一部の必須入力値を指定しない場合、値を要求するプロンプトが出されます。/DDEL アクションの場合、コマンド行からオプションを指定できないため、**db2dart** によってプロンプトが出されたときに入力する必要があります。
- /ROW、/RPT、/RPTN、/SCR、/RPTF、/ERR、および /WHAT DBBP オプションはすべて、アクションに加えて呼び出すことができます。これらのオプションは、どのアクションでも必須ではありません。
- /DB、/T、および /TS オプションは、指定されたオブジェクトに関連した XML ストレージ・オブジェクトも含めて検査します。/DB オプションではデータベース中のすべての XML ストレージ・オブジェクトが含まれ、/T オプションでは指定された表に関連する XML ストレージ・オブジェクトが含まれ、/TS オプションでは指定された表スペースに親オブジェクトが存在するすべての XML ストレージ・オブジェクトが検査されます。また、/DEMP オプションはフォーマット済み EMP 情報を関連した XML ストレージ・オブジェクトに対するものも含めてダンプします。
- 単一の表スペースに対して **db2dart** が実行された場合には、その表スペースにある親表のすべての従属オブジェクトが、従属オブジェクトがどの表スペースにあるかにかかわらず、検査されます。ただし、指定された表スペースの外にある従属オブジェクトについては、エクステント・マップ・ページ (EMP) 情報はキャプチャーされません。親オブジェクトが指定された表スペース以外の表スペースにあるとしても、EMP 情報は、指定された表スペース内にある従属オブジェクトに対してキャプチャーされます。
- パーティション表の場合、/DD、/DM、/DEMP、/DDEL、/DP、/DXA、/DXH アクションは、特定のパーティションに関する表オブジェクト ID (/OI) および表スペース ID (/TSI) に対する最初の 2 つの入力データとして、syscat.datapartitions から partitionobjectid と tbspaceid を使用します。これらのアクションには表名オプション

ョン (/TN) はサポートされていません。/T アクションは、グローバル表スペース ID と併用して表全体をチェックする際に表名またはグローバル表オブジェクト ID をサポートし、/OI および /TSI に対する入力データとして syscat.datapartitions から partitionobjectid および tbspaceid を使用した特定のパーティションのチェックもサポートします。

- 一般に、**db2dart** コマンドは、データベースがオフラインのときに実行します。ただし、/DHWM または /LHWM アクションのいずれかを指定する場合には、オフライン・データベースは必要ありません。レポートは、データベースがオフラインでなくとも生成できますが、その結果の信頼性は、最近の書き込み/更新アクティビティーがどれほど発生したかによって変化します (アクティビティーが少ないほうが、結果の信頼性は高くなります)。

## db2daslevel - DAS レベルの表示

システム上の現在の DAS レベルを表示します。

このコマンドからの出力は、デフォルトでコンソールに表示されます。

**重要:** バージョン 9.7 で DB2 Administration Server (DAS) が非推奨になったため、このコマンドは推奨されておらず、将来のリリースで削除される可能性があります。詳しくは、『DB2 Administration Server (DAS) が推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0059276.html>) を参照してください。

### 許可

なし

### 必要な接続

なし

### コマンド構文

▶▶—db2daslevel—▶▶

### コマンド・パラメーター

なし

## db2dclgn - 宣言生成プログラム

指定されたデータベース表に宣言を生成し、文書中でそれらの宣言を検索する必要を省きます。生成された宣言は、必要に応じて変更できます。サポートされるホスト言語は C/C++、COBOL、JAVA、および FORTRAN です。

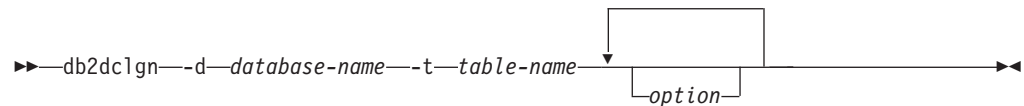
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### -d *database-name*

接続の確立先のデータベースの名前を指定します。

#### -t *table-name*

宣言を生成するために列情報が検索される表の名前を指定します。

#### *option*

以下のオプションのうちの 1 つ以上。

#### -a *action*

宣言が追加されるか、置換されるかを指定します。有効な値は ADD および REPLACE です。デフォルト値は ADD です。

#### -b *lob-var-type*

LOB 列に生成される変数のタイプを指定します。有効な値は以下のとおりです。

##### **LOB (デフォルト)**

例えば、C では SQL TYPE は CLOB(5K) x です。

##### **LOCATOR**

例えば、C では SQL TYPE は CLOB\_LOCATOR x です。

##### **FILE**

例えば、C では SQL TYPE は CLOB\_FILE x です。

-c 接頭部 (-n) の指定時に、列名が、フィールド名で接尾部として使用されるかどうかを指定します。接頭部が指定されない場合、このオプションは無視されます。デフォルトの動作は、列名を接尾部としては使用せず、代わりに 1 で始まる列番号を使用します。

-i 標識変数が生成されるかどうか指定します。ホスト構造は C および COBOL でサポートされるので、列の数に等しいサイズの標識表が生成され

ますが、JAVA および FORTRAN の場合は、個々の標識変数が各列ごとに生成されます。標識表および変数の名前は、表名および列名と同じで、"IND-" (COBOL の場合) または "ind\_" (その他の言語の場合) という接頭部が付きます。デフォルトの動作は、標識変数を生成しません。

### -l *language*

宣言が生成されるホスト言語を指定します。有効な値は C、COBOL、JAVA、および FORTRAN です。デフォルトの動作は、C 宣言を生成することで、C++ にも有効です。

### -n *name*

それぞれのフィールド名に接頭部を指定します。接頭部は、**-c** オプションが使用される場合に指定する必要があります。指定されないと、列名がフィールド名として使用されます。

### -o *output-file*

宣言用の出力ファイルの名前を指定します。デフォルトの動作は、生成されたホスト言語を反映した拡張子を使った、基本ファイル名として表名を使用します。

```
.h (C の場合)
.cb1 (COBOL の場合)
.java (JAVA の場合)
.f (FORTRAN (UNIX) の場合)
.for (FORTRAN (INTEL) の場合)
```

### -p *password*

データベースへの接続に使用するパスワードを指定します。ユーザー ID を指定する場合に指定する必要があります。デフォルトの動作では、接続の確立時にパスワードを提供しません。

### -r *remarks*

列の注釈が使用できる場合、宣言内のコメントとして使用される、フィールドのより詳細な記述を提供するかどうかを指定します。

### -s *structure-name*

宣言内のすべてのフィールドをグループ化するために生成される構造名を指定します。デフォルトの動作では、修飾なしの表名を使用します。

### -u *userid*

データベースへの接続に使用するユーザー ID を指定します。パスワードを指定する場合に指定する必要があります。デフォルトの動作では、接続の確立時にユーザー ID を提供しません。

**-v** ユーティリティーの状況 (例えば接続状況) が表示されるかどうかを指定します。デフォルトの動作では、エラー・メッセージのみを表示します。

### -w *DBCS-var-type*

sqldbchar または wchar\_t が、C で GRAPHIC、VARGRAPHIC、または DBCLOB 列に使用されるかどうかを指定します。

### -y *DBCS-symbol*

G または N が、COBOL で DBCS シンボルとして使用されるかどうかを指定します。

### -z *encoding*

encoding に、特定のサーバーに合わせたコーディング規則を指定します。

encoding は LUW または OS390 にすることができます。OS390 を指定した場合、生成されるファイルは、OS/390 で生成されるファイルと同じ外観を持つものになります。

### 例

```
db2dclgn -d sample -t emp_resume -l cobol -a replace
```

## db2diag - db2diag ログ分析ツール

このユーティリティーは、単一および循環 **db2diag** ログ・ファイルの両方をフィルター操作およびフォーマット設定するためのツールです。**diagsize** データベース・マネージャーの構成パラメーターが設定されている場合、**db2diag** ツールは、循環 **db2diag** ログ・ファイルから読み取ります。その他の場合、デフォルトでは、デフォルトの `db2diag.log` ファイルから読み取ります。

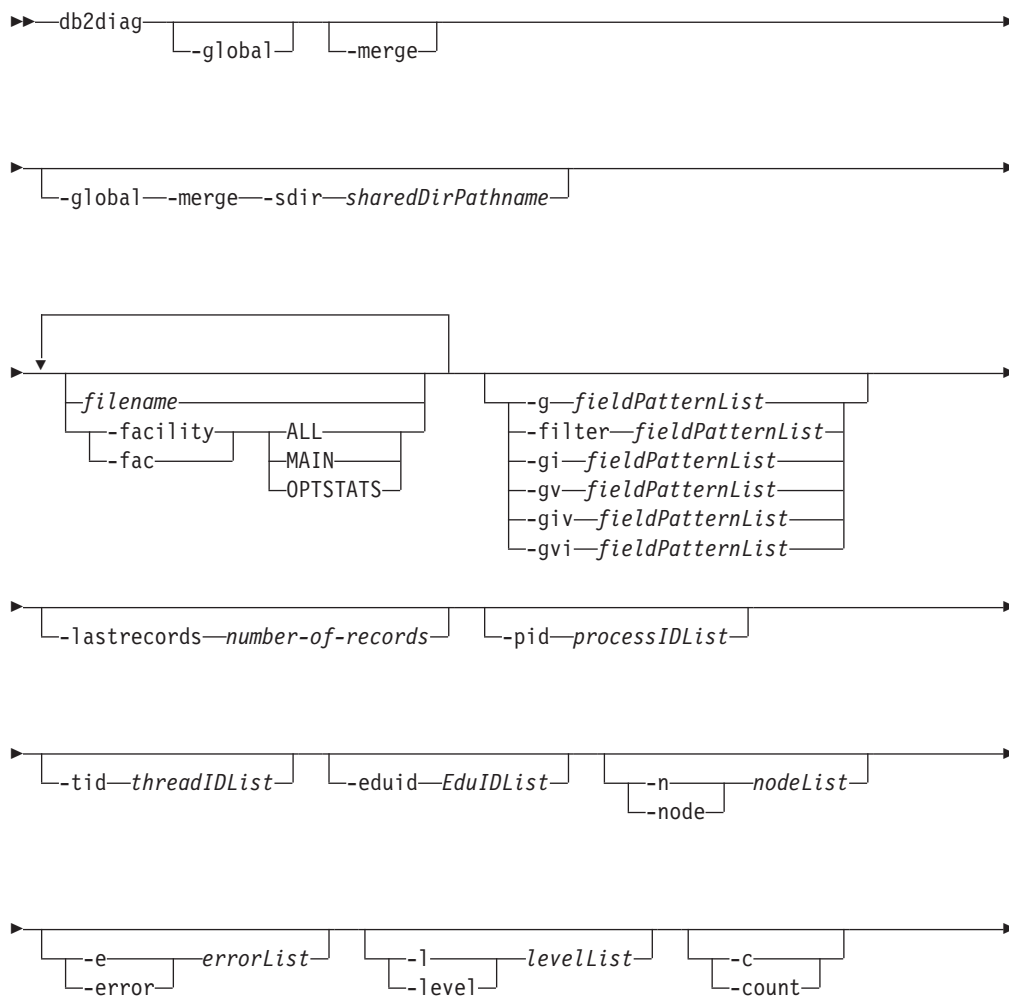
### 許可

なし

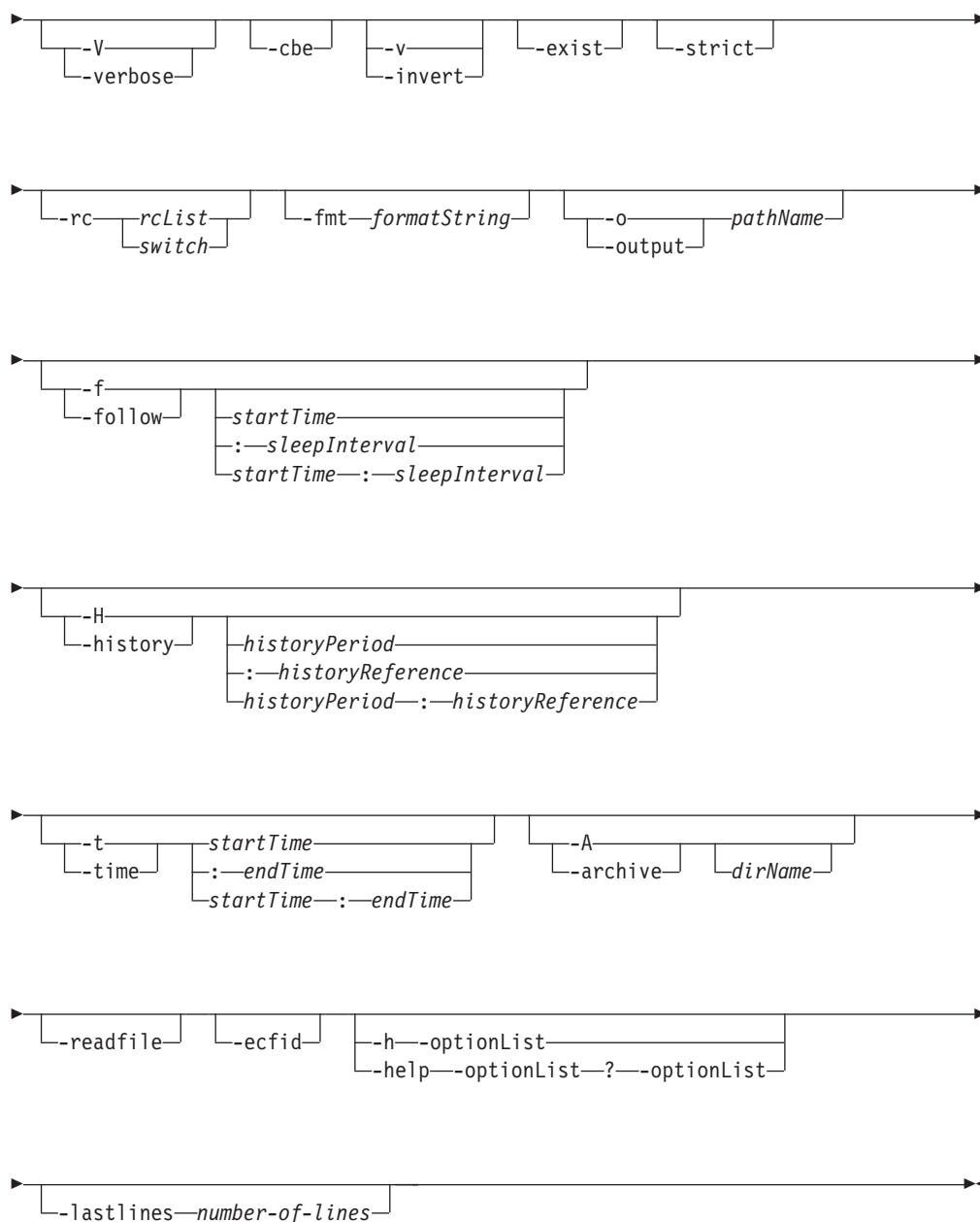
### 必要な接続

なし

### コマンド構文







## コマンド・パラメーター

**-global** ログ・ファイルの処理に、すべてのホスト上にあるすべてのデータベース・パーティションからの **db2diag** ログ・ファイルをすべて含めることを指定します。

**注:** このコマンド・パラメーターは、DB2 バージョン 9.8 フィックスパック 2 以降で使用できます。このオプションは、循環診断ログ・ファイルと、分割診断データ・ディレクトリー内のファイルをサポートします。DB2 バージョン 9.8 フィックスパック 2 では、このオプションを **-follow** オ

プションと組み合わせて使用することはできません。DB2 バージョン 9.8 フィックスパック 3 以降では、このオプションを **-follow** オプションと組み合わせて使用できます。

**-merge** 診断ログ・ファイルをマージし、レコードをタイム・スタンプに基づいてソートします。このオプションは、循環診断ログ・ファイルと、分割診断データ・ディレクトリー内のファイルをサポートします。

このコマンド・パラメーターの後に複数の *filename* 値がスペースで区切って指定されていない場合は、**diagpath** データベース・マネージャー構成パラメーターおよび **alt\_diagpath** データベース・マネージャー構成パラメーターで指定されたディレクトリー (複数可) 内にある **db2diag** ログ・ファイルがマージされます。診断データ・ディレクトリー・パスが複数のデータベース・パーティションにまたがって分割されている場合、現在のホストのデータベース・パーティションにある **db2diag** ログ・ファイルのみマージされます。

1 つの *filename* だけが指定されている場合、あるいは **diagpath** データベース・マネージャー構成パラメーターで指定されたパスに 1 つの診断ファイルしか存在せず、かつ **alt\_diagpath** データベース・マネージャー構成パラメーターが設定されていない場合は、**-merge** コマンド・パラメーターが指定されなかった場合と同様に、このコマンドによって 1 つの診断ログ・ファイルが処理されます。

**注:** このコマンド・パラメーターは、DB2 バージョン 9.7 フィックスパック 1 以降で使用できます。これは、**-facility**、**-follow**、または **-archive** オプションと組み合わせて使用することはできません。DB2 バージョン 9.7 フィックスパック 7 以降、**-merge** オプションは、**alt\_diagpath** ディレクトリー内の **db2diag.log** ファイルの自動マージをサポートします。

**-global -merge -sdir sharedDirPathname**

**-global** オプションと **-merge** オプションを一緒に指定すると、すべてのホスト上にあるすべてのデータベース・パーティションからの **db2diag** ログ・ファイルがすべてマージされ、タイム・スタンプに基づいてレコードがソートされます。このオプションは、循環診断ログ・ファイルと、分割診断データ・ディレクトリー内のファイルをサポートします。

**注:** 複数の異なるホストから取得されてマージされた診断ログ・ファイルを一時的に保管するには、**-sdir sharedDirPathname** オプションを指定しなければなりません。一時的にマージされた診断ログ・ファイルは、処理の完了後に削除されます。*sharedDirPathname* には、すべてのホストがアクセス許可と書き込み許可を持つ共有ディレクトリーを指定しなければなりません。

### *filename*

処理する DB2 診断ログのパス名を指定します (複数の場合はスペースで区切ります)。ファイル名が省略されると、現行ディレクトリーの **db2diag** ログ・ファイルが処理されます。ファイルが検出されない場合、**diagpath** データベース・マネージャーの構成パラメーターによって設定されたディレクトリーが検索されます。

**-facility | -fac**

対応するファシリティからファイルを読み取ります。ファシリティと

は、複数のレコードを論理的にグループ化したものです。例えば、すべてのオブティマイザー統計レコードは OPTSTATS ファシリティーにグループ化されます。これは、デフォルトではテキスト・フォーマットで出力されます。有効なファシリティー・オプションは次の値です。

**ALL** すべてのファシリティーからレコードを戻します。

**MAIN** DB2 の汎用診断ログ (例えば **db2diag** ログ・ファイルおよび循環イベント・ログなど) からレコードを戻します。

#### OPTSTATS

オブティマイザー統計に関連したレコードを返します。

#### -fmt *formatString*

フォーマット・ストリング *formatString* を使用して **db2diag** 出力のフォーマットを設定します。フォーマット・ストリングには、%field、{%field}、@field、または @{field} の形式でレコード・フィールドを含めることができます。{%field} および @{field} は、フィールド名に続く英数字 (または他の許可されている文字) からフィールド名を分離するために使用されます。フィールド名はすべて大文字小文字を区別しません。フィールド名は、フィールド名を固有に識別するために必要な範囲で、最初の数文字に短縮することができます。さらに、長い名前前のフィールドには別名を使用できます。フィールド名前の接頭部 % または @ は、フィールドが空の場合にフィールドの前のテキストを表示する (%) かしない (@) かを指定します。

現在、次のフィールドが使用可能です。

#### timestamp | ts

タイム・スタンプ。このフィールドはその構成要素フィールド (%tsyear、%tsmonth、%tsday、%tshour、%tsmin (分)、%tssec (秒)、%tsmsec (UNIX オペレーティング・システムの場合はマイクロ秒、Windows オペレーティング・システムの場合はミリ秒)) に分割できます。

#### timezone | tz

UTC (Universal Coordinated Time) との差 (分)。例えば、東部標準時の場合は -300 です。

#### recordid | recid

レコードの固有の英数字 ID (例えば I11455A696)。

#### audience

ログに記録されたメッセージの対象ユーザー。'E' は外部ユーザー (IBM のお客様、サービス・アナリスト、および開発者など) を示します。'I' は、内部ユーザー (サービス・アナリストおよび開発者など) を示します。'D' は開発者用のデバッグ情報を示します。

**level** メッセージの重大度レベル。Info、Warning、Error、Severe、または Event。

**source** ログに記録されたエラーの発信位置。Origin、OS、Received、または Sent です。

#### instance | inst

インスタンス名。

- node** データベース・パーティション・サーバー番号。
- database | db**  
データベース名
- pid** プロセス ID。
- tid** スレッド ID。
- eduid** EDU ID。
- eduname**  
EDU 名。
- process**  
プロセス ID に関連付けられた名前。二重引用符で囲みます。例えば、"db2sysc.exe" とします。
- product**  
製品名。例えば、DB2 COMMON となります。
- component**  
コンポーネント名。
- funcname**  
関数名。
- probe** プロブ番号。
- function**  
関数の詳細な説明。%prod、%comp、%funcname、probe:%probe で  
す。
- appid** アプリケーション ID。この値は **appl\_id** モニター・エレメント・  
データと同じです。この値を解釈する方法について、詳しくは  
『appl\_id アプリケーション ID : モニター・エレメント』を参照し  
てください。
- coordnode**  
コーディネーター・パーティション。
- coordindex**  
コーディネーター索引。
- apphdl** アプリケーション・ハンドル。%coordnode - %coordindex。
- message | msg**  
エラー・メッセージ。
- calledprod**  
エラーを戻した関数の製品名。
- calledcomp**  
エラーを戻した関数のコンポーネント名。
- calledfunc**  
エラーを戻した関数の名前。
- called** エラーを戻した関数の詳細な説明。 %calledprod、%calledcomp、  
%calledfunc。
- rcval** 戻りコード値 (32 バイト)。

**rcdesc** エラーの説明。

**retcode | rc**

呼び出された関数によって戻された戻りコード。%rcval %rcdesc。

**errno** システム・エラー番号。

**errname**

システム固有のエラー名。

**oserror**

システム呼び出しによって戻されるオペレーティング・システム・エラー。%errno %errname。

**callstack**

呼び出しスタック。

**datadesc**

データの説明。

**dataobject**

データ・オブジェクト。

**data** メッセージの詳細なデータ・セクション。%datadesc %dataobject。

**argdesc**

引数の説明。

**argobject**

引数オブジェクト。

**arg** エラーを戻した関数呼び出しの引数。 %argdesc %argobject。

**イベントの説明:**

**impact** ユーザーへの影響 (イベントのみ)。

**startevent**

開始イベントの説明 (\*)。

**stopevent**

停止イベントの説明 (\*)。

**changeevent**

変更イベントの説明 (\*)。

**init** 初期化イベントの説明 (\*)。

**fini** 完了/完結イベントの説明 (\*)。

**startup**

スタートアップ・イベントの説明 (\*)。

**terminate**

終了イベントの説明 (\*)。

**bringdown**

停止イベントの説明 (\*)。

**interrupt**

割り込みイベントの説明 (\*)。

**associate**

関連付けイベントの説明 (\*)。

**disassociate**

関連付け解除イベントの説明 (\*)。

**changecfg**

構成変更イベントの説明 (\*)。

**transfer**

転送イベントの説明 (\*)。

**dispatch**

ディスパッチ・イベントの説明 (\*)。

**switch** 切り替えイベントの説明 (\*)。

**report** レポート・イベントの説明 (\*)。

**get** 取得イベントの説明 (\*)。

**free** 解放イベントの説明 (\*)。

**open** オープン・イベントの説明 (\*)。

**close** クローズ・イベントの説明 (\*)。

**work** 作業イベントの説明 (\*)。

**wait** 待機イベントの説明 (\*)。

**available**

使用可能イベントの説明 (\*)。

**connect**

接続イベントの説明 (\*)。

**disconnect**

切断イベントの説明 (\*)。

**accept** 受け入れイベントの説明 (\*)。

**recv** 受信イベントの説明 (\*)。

**send** 送信イベントの説明 (\*)。

**create** 作成イベントの説明 (\*)。

**destroy**

破棄イベントの説明 (\*)。

**request**

要求イベントの説明 (\*)。

**reply** 応答イベントの説明 (\*)。

**dependency**

従属関係イベントの説明 (\*)。

**write** 書き込みイベントの説明 (\*)。

**read** 読み取りイベントの説明 (\*)。

**reset** リセット・イベントの説明 (\*)。

**collect**

収集イベントの説明 (\*)。

**add** 追加イベントの説明 (\*)。

**alter** 変更イベントの説明 (\*)。

**drop** ドロップ・イベントの説明 (\*)。

**invalidate**

無効化イベントの説明 (\*)。

**grant** 付与イベントの説明 (\*)。

**revoke** 取り消しイベントの説明 (\*)。

(\*) 各イベント・フィールドには、次のようなサブフィールドがあります。

**{event}type**

イベント・タイプ

(START、STOP、READ、WRITE、GET)。

**{event}desc**

イベント記述 (イベント情報付きのヘッダー)。

**{event}state**

イベントの状態 (成功、失敗、開始、停止、進行中、アイドル) またはイベントの進行状況 (%)。

**{event}attr**

イベントの属性 (ビジネス・レベル、キャッシュ、同期、非同期、内部、外部、論理、物理、自動、手動、一時、永続)。

**{event}objid**

固有のオブジェクト ID (TABLE、CFG、DBM)。

**{event}objname**

イベント・オブジェクト名 (例えば、  
"schema.tablename")。

**{event}objdata**

オブジェクト・データ (オブジェクトがストリングまたは単純な整数タイプでない場合、例えば、データ構造体や何らかの複合タイプである場合に使用されます)。

**{event}qtype**

イベント修飾子のタイプ

(FROM、TO、ON、FOR、AT、BY、CONTEXT)。

**{event}qname**

イベント修飾子の名前/値 (例えば FOR "DB  
ABC")。

**{event}qdhdr**

イベント修飾子のデータ・ヘッダー (データのタイ

プ、テキスト記述、およびサイズが含まれます。  
`%{event}qdata` フィールドと共に使用されます。

#### `{event}qdata`

イベント修飾子データ (修飾子がストリングまたは単純な整数タイプでない場合、例えば、何らかのデータ構造体や複合タイプである場合に使用されま  
す)。

上記のリストで、キーワード `{event}` を、特定のイベントのイベント・タイプに置き換えてください (例えば、開始、停止、変更、読み取り、書き込み)。

フィールド名の前のテキストを常に表示する (例えば、必須フィールドの場合など) には、`%` フィールド接頭部が使用されます。あるフィールドにデータが入っている場合に、そのフィールド名の前のテキストを表示するには、`@` 接頭部を使用します。必須フィールドおよびオプション・フィールドは、任意のテキスト記述と組み合わせることができます。

フォーマット・ストリング内で認識される特殊文字は、`¥n`、`¥r`、`¥f`、`¥v`、および `¥t` です。

他のフィールドとは対照的に、データおよび引数フィールドにはいくつかのセクションを含めることができます。特定のセクションを出力するには、`[n]` をフィールド名の後に追加します。ここで、`n` はセクション番号です ( $1 \leq n \leq 64$ )。例えば、1 番目のデータ・オブジェクトと 2 番目のデータ記述セクションを出力するには、`%{dataobj}[1]` と `%{datadesc}[2]` を使用します。`[n]` が使用されない場合、ログに記録されているすべてのセクションが事前にフォーマット設定されている記録データを使用して、ログ・メッセージで現れるとおりに出力されます。そのため、各データ・フィールド、引数フィールド、またはセクションの前に該当するテキスト記述および区別のための改行を追加する必要はありません。

#### `-filter fieldPatternList | -g fieldPatternList`

`fieldPatternList` は、フィールド・パターンの対のコンマで区切られたリストで、`fieldName operator searchPattern` の形式です。

演算子は以下の値の 1 つにすることができます。

- `=` 単語単位で一致する対象を含むレコードだけを選択します。(ワード検索。)
- `:=` 検索パターンが大きな対象の一部と一致する場合に、その対象を含むレコードだけを選択します。
- `!=` 一致しない行だけを選択します。(逆転ワード・マッチング。)
- `!:=` 検索パターンが大きな対象の一部と一致する場合に、その対象を含まない行だけを選択します。
- `^=` 指定された検索パターンで始まるフィールド値のレコードを選択します。
- `!^=` 指定された検索パターンで始まらないフィールド値のレコードを選択します。



**-fmt** オプションで説明したものと同一フィールドが使用できます。ただし、% および @ 接頭部はこのオプションに使用されません。

**-gi** *fieldPatternList*

**-g** と同じですが、大文字小文字を区別しません。

**-gv** *fieldPatternList*

指定されたパターンに一致しないメッセージを検索します。

**-gvi** | **-giv** *fieldPatternList*

**-gv** と同じですが、大文字小文字を区別しません。

**-lastrecords** *number-of-records*

db2diag ログ・ファイルにある最新の指定数のレコードを表示およびフィルター操作します。このパラメーターは、db2diag log ファイルごとに、指定した数のレコードを使用できるかどうかを検査します。ログ・ファイルで使用可能なレコードの数が指定レコード数より少ない場合、**db2diag** コマンドはそのファイルで使用可能なすべてのレコードを処理します。分割診断データのディレクトリー・パスを使用する場合、指定数の最新のレコードが各パス内のそれぞれの db2diag ログ・ファイルごとに戻されます。

**-pid** *processIDList*

リストに含まれているプロセス ID を持つログ・メッセージだけを表示します。

**-tid** *threadIDList*

リストに含まれているスレッド ID のログ・メッセージだけを表示します。

**-eduid** *EduIDList*

1 つまたは複数のコンマ区切り数値が含まれる EDU ID リストから、指定した EDU ID を持つすべてのレコードを検出します。

**-n** | **-node** *nodeList*

リストに含まれているデータベース・パーティション番号のログ・メッセージだけを表示します。

**-e** | **-error** *errorList*

リストに含まれているエラー番号のログ・メッセージだけを表示します。

**-l** | **-level** *levelList*

1 つまたは複数のコンマ区切りテキスト値 (情報、警告、エラー、重大、クリティカル、イベント) が含まれる重大度レベル・リストから、指定した重大度レベルを持つすべてのレコードを検出します。

**-c** | **-count**

検出されたレコードの数を表示します。

**-v** | **-invert**

パターン・マッチングを反転させて、指定されたパターンと一致しないすべてのレコードを選択します。

**-strict**

各行に 1 つのフィールド: 値の対を示すフォーマットを使用してレコードを表示します。空のフィールドはすべて省略されます。解析を単純化するために、これをスクリプト用に使用できます。

### **-V | -verbose**

すべてのフィールドを、空のフィールドも含めて出力します。

**-exist** 検索が要求されるとき、レコード内のフィールドが処理される方法を定義します。このオプションが指定される場合、処理されるフィールドが存在しなければなりません。

**-cbe** Common Base Event (CBE) Canonical Situation Data

### **-o | -output pathName**

完全修飾 *pathName* で指定されたファイルに出力を保存します。

### **-f | -follow**

入力ファイルが単一および循環 **db2diag** ログ・ファイルである場合、入力ファイルの最後のレコードが処理された後にツールが終了しないことを指定します。その代わりに、ツールは指定された期間 (*sleepInterval*) スリープ状態となり、その後、使用できる状態になるときに入力ファイルから追加のレコードの処理を試行します。バージョン 9.8 フィックスパック 3 の時点では、入力ファイルの最後の 8 キロバイトに含まれるレコードのみが処理されます。

**db2diag** コマンドを **-f** または **-follow** コマンド・パラメーターを指定して使用すると、循環 **db2diag** ログ・ファイルを処理します。例えば、このコマンドは、使用中の最新の循環診断ログ・ファイル (**db2diag.23.log**) を読み取り、**db2diag.23.log** ログ・ファイルがサイズ制限を満たす場合には次に作成された循環ログ・ファイル (**db2diag.24.log**) を読み取ります。

これは、別のプロセスによってファイルに書き込まれるレコードをモニターするときに使用できるオプションです。 *startTime* オプションは、この時刻の後にログに記録されたすべてのレコードを表示するために指定されます。 *startTime* オプションは以下のフォーマットで指定できます。

YYYY-MM-DD-hh.mm.ss.nnnnnn

YYYY 年を指定します。

MM 月を指定します (01 から 12)。

DD 日を指定します (01 から 31)。

hh 時間を指定します (00 から 23)。

mm 分を指定します (00 から 59)。

ss 秒を指定します (00 から 59)。

nnnnnn UNIX オペレーティング・システムではマイクロ秒で、Windows オペレーティング・システムではミリ秒で指定します。

年フィールドに続くフィールドのいくつかまたはすべては、省略できます。それらが省略されると、デフォルト値が使用されます。月および日のデフォルト値は 1 であり、他のすべてのフィールドのデフォルト値は 0 です。

レコードのタイム・スタンプに正確に一致するものが診断ログ・ファイルに存在しない場合、指定されたタイム・スタンプより前の、最も近い時刻が使用されます。

*sleepInterval* オプションは、スリープ・インターバルを秒数で指定します。さらに小さな単位が必要な場合、浮動小数点の値で指定できます。デフォルト値は 2 秒です。

#### **-H | -history**

指定された期間、ログに記録されたメッセージの履歴を表示します。このオプションには、以下のオプションを指定できます。

##### *historyPeriod*

ログに記録されたメッセージが、最後に記録されたレコードから始めて、*historyPeriod* で指定された期間に関して表示されるように指定します。*historyPeriod* オプションは、*Number timeUnit* 形式で指定されます。ここで、*Number* は時間単位の数値であり、*timeUnit* は時間単位のタイプ、M (月)、d (日)、h (時間)、m (分)、s (秒) です。*Number* のデフォルト値は 30 であり、*timeUnit* デフォルト値は m です。

##### *historyPeriod:historyReference*

*historyReference* で指定された開始時刻から後の期間に記録されたログ対象メッセージを表示する (明示的に正の値が *historyPeriod* に指定されている場合)、または *historyReference* で指定された終了時刻より前の期間に記録されたログ対象メッセージを表示する (負の値が *historyPeriod* に指定されている、またはデフォルトの場合) ように指定します。

フォーマットは、*YYYY-MM-DD-hh.mm.ss.nnnnnnn* です。

*YYYY* 年を指定します。

*MM* 月を指定します (01 から 12)。

*DD* 日を指定します (01 から 31)。

*hh* 時間を指定します (00 から 23)。

*mm* 分を指定します (00 から 59)。

*ss* 秒を指定します (00 から 59)。

*nnnnnn* マイクロ秒 (UNIX オペレーティング・システム) またはミリ秒 (Windows オペレーティング・システム) を指定します。

#### **-t | -time**

タイム・スタンプの値を指定します。このオプションには、以下のオプションの 1 つまたは両方を指定できます。

##### *startTime*

*startTime* 以降にログに記録されたすべてのメッセージを表示します。

##### *:endTime*

*endTime* の前にログに記録されたすべてのメッセージを表示します。

*startTime* と *endTime* の間にログに記録されたメッセージを表示するには、*-t startTime:endTime* を指定します。

フォーマットは、*YYYY-MM-DD-hh.mm.ss.nnnnnn* です。

*YYYY* 年を指定します。

*MM* 月を指定します (01 から 12)。

*DD* 日を指定します (01 から 31)。

*hh* 時間を指定します (00 から 23)。

*mm* 分を指定します (00 から 59)。

*ss* 秒を指定します (00 から 59)。

*nnnnnn* マイクロ秒 (UNIX オペレーティング・システム) またはミリ秒 (Windows オペレーティング・システム) を指定します。

年フィールドに続くフィールドのいくつかまたはすべては、省略できます。それらが省略されると、デフォルト値が使用されます。月および日のデフォルト値は 1 であり、他のすべてのフィールドのデフォルト値は 0 です。

レコードのタイム・スタンプに正確に一致するものが診断ログ・ファイルにない場合、指定されたタイム・スタンプに最も近い時刻が使用されます。

### **-A | -archive** *dirName*

単一の診断ログ・ファイルと循環診断ログ・ファイルの両方をアーカイブします。このオプションを指定した場合、他のすべてのオプションは無視されます。1 つ以上のファイル名が指定されると、各ファイルは個別に処理されます。*YYYY-MM-DD-hh.mm.ss* という形式のタイム・スタンプがファイル名に追加されます。

アーカイブされるファイルおよびディレクトリーの名前は指定できます。ディレクトリーが指定されない場合、そのファイルはそれが置かれているディレクトリーにアーカイブされ、ディレクトリー名はファイル名から抽出されます。

ディレクトリーを指定してファイル名を指定しない場合、現行のディレクトリー内で **db2diag** ログ・ファイルを検索します。ファイルが検出されると、ファイルは指定のディレクトリーにアーカイブされます。ファイルが検出されない場合、**diagpath** および **alt\_diagpath** 構成パラメーターによって指定されたディレクトリー内で **db2diag** ログ・ファイルが検索されます。ファイルが検出されると、指定のディレクトリー内にアーカイブされます。

ファイルまたはディレクトリーを指定しない場合、現行ディレクトリー内で **db2diag** ログ・ファイルを検索します。ファイルが検出されると、現行ディレクトリーにアーカイブされます。ファイルが検出されない場合、**diagpath** および **alt\_diagpath** 構成パラメーターによって指定されたディレクトリー内で **db2diag** ログ・ファイルが検索されます。ファイルが検出されると、**diagpath** または **alt\_diagpath** 構成パラメーターによって指定されているディレクトリーにアーカイブされます。

バージョン 9.7 フィックスパック 4 以降、**db2diag -archive** オプションを IBM Data Server Driver Package および IBM Data Server for ODBC and CLI で使用できます。このオプションを使用すると、インスタスのないクライアント上に診断ログ・ファイルをアーカイブできます。例えば、

```
$ db2diag -A
db2diag: Moving "/home/usr1/clidriver/db2dump/db2diag.log"
to "/home/usr1/clidriver/db2dump/db2diag.log_2010-09-14-01.16.26"
```

**-readfile**

端末入力を無視して、診断ログ・ファイルから読み取することを強制します。このオプションはスクリプトで使用して、特に、`stdin` が無効になっているか、または自動化ツールが使用される状況で、**db2diag** が端末からではなくファイルから読み取られることを保証することができます。 **rah** または **db2\_all** を使用して **db2diag** コマンドを実行するには、**-readfile** オプションも使用する必要があります。

**-rc rcList | switch**

特定の ZRC または ECF 16 進数または負の 10 進数戻りコードのスペース区切りリスト *rcList* に関する DB2 内部エラー戻りコードの説明を表示します。以下のスイッチの 1 つを指定することによって、ZRC または ECF 戻りコードの完全なリストを表示できます。

**zrc** DB2 ZRC 戻りコードに関する簡略説明を表示します。

**ecf** DB2 ECF 戻りコードに関する簡略説明を表示します。

**html** DB2 ZRC 戻りコードに関する簡略説明を HTML 形式で表示します。

このオプションを指定すると、他のすべてのオプションは無視され、出力はディスプレイに送られます。

**-ecfid ecfld**

数値 *ecfld* から抽出した関数情報を表示します。このオプションを指定した場合、他のすべてのオプションは無視されます。

**-h | -help | ?**

ヘルプ情報を表示します。このオプションを指定すると、他のすべてのオプションは無視され、ヘルプ情報だけが表示されます。1 つ以上のコマンド・パラメーター (複数の場合はコマンドで区切る) を含むオプション・リスト *optionList* が省略されると、使用できるすべてのオプションのリストが簡単な説明とともに表示されます。 *optionList* に指定された各オプションについては、さらに詳細な情報と使用例が表示されます。 *optionList* 引数の代わりに以下のスイッチの 1 つを使用することによって、ツールおよびその使用法の詳細が表示されるようにヘルプ出力を変更できます。

**brief** すべてのオプションのヘルプ情報を表示します。例は表示しません。

**examples**

ツールの使用に役立ついくつかの標準的な例を表示します。

**tutorial**

高度なフィーチャーを説明する例を表示します。

**notes** 使用上の注意と制限を表示します。

**all** すべてのオプションに関する完全な情報と、各オプションの使用例を表示します。

**-lastlines number-of-lines**

db2diag ログ・ファイルの最後から指定された行数を、表示およびフィルタ

ーに掛けます。このパラメーターを指定すると、db2diag log ファイルごとに指定された行数を取得できるかどうか検査されます。ログ・ファイル内にある行数が、指定されたレコード数より少ない場合、db2diag コマンドはファイル内にある全行を処理します。分割診断データ・ディレクトリー・パスを使用している場合、それぞれのパスの db2diag ログ・ファイルごとに、ファイルの最後から指定された行数が戻されます。

### 例

さまざまな状況下で **db2diag** コマンドを使用する方法を示す例を以下にリストします。

- 診断データ・ディレクトリー・パスにあるすべての **db2diag** ログ・ファイルをマージするには、次のコマンドを実行します。

```
db2diag -merge
```

診断データ・ディレクトリー・パスが複数のデータベース・パーティションに従って分割されている場合、このコマンドは現在のホストのすべてのデータベース・パーティションからの **db2diag** ログ・ファイルをマージします。診断データ・ディレクトリー・パスが分割されていない場合、**-merge** オプションが指定されていないかのように、単一診断ログ・ファイルがコマンドによって処理されます。

- この例では、以下のコマンドを使用して **diagpath** データベース・マネージャー構成パラメーターを設定することにより、物理ホストとデータベース・パーティションに従ってデフォルトの診断データ・ディレクトリー・パスが分割されました。

```
db2 update dbm cfg using diagpath "$h$n"
```

この例は、全診断ログからの全レコードの出力を入手し、2つのホスト **bower** および **horton** のそれぞれにある3つのデータベース・パーティションからの診断ログ・ファイルをマージする方法を示します。以下に示すのは、6つの **db2diag** ログ・ファイルのリストです。

```
- ~/sqllib/db2dump/HOST_bower/NODE0000/db2diag.log  
- ~/sqllib/db2dump/HOST_bower/NODE0001/db2diag.log  
- ~/sqllib/db2dump/HOST_bower/NODE0002/db2diag.log  
- ~/sqllib/db2dump/HOST_horton/NODE0003/db2diag.log  
- ~/sqllib/db2dump/HOST_horton/NODE0004/db2diag.log  
- ~/sqllib/db2dump/HOST_horton/NODE0005/db2diag.log
```

6つの **db2diag** ログ・ファイルのすべてからレコードを出力するには、以下のコマンドを実行します。

```
db2diag -global
```

ホスト **bower** および **horton** 上のそれぞれにある3つのデータベース・パーティションすべてからの診断データ・ディレクトリー・パスにある6つの **db2diag** ログ・ファイルすべてをマージし、タイム・スタンプに基づいて出力をフォーマットするには、次のコマンドを実行します。

```
db2diag -global -merge -sdir /temp/keon -fmt %{ts}
```



/temp/keon は、処理中に各ホストから一時的にマージされたファイルを保管するために、ホスト bower および horton で共有される共有ディレクトリです。

- すべてのクリティカル・エラー・メッセージを表示するには、以下のように入力します。

```
db2diag -level critical
```

または

```
db2diag -g 'level=Critical'
```

- プロセス ID (PID) 52356 のプロセスによって生成され、データベース・パーティション 1、2、または 3 に関するすべての重大エラー・メッセージを表示するには、以下のように入力します。

```
db2diag -g level=Severe,pid=952356 -n 1,2,3
```

- データベース SAMPLE およびインスタンス aabrashk を含むすべてのメッセージを表示するには、以下のように入力します。

```
db2diag -g db=SAMPLE,instance=aabrashk
```

- データベース・フィールドを含むすべての重大エラー・メッセージを表示するには、以下のように入力します。

```
db2diag -g db:= -gi level=severe
```

- DB2 ZRC 戻りコード 0x87040055 およびアプリケーション ID G916625D.NA8C.068149162729 を含むすべてのエラー・メッセージを表示するには、以下のように入力します。

```
db2diag -g msg:=0x87040055 -l Error | db2diag -gi appid^=G916625D.NA
```

- LOADID データを含んでいないすべてのメッセージを表示するには、以下のように入力します。

```
db2diag -gv data:=LOADID
```

- アプリケーション ID フィールドに LOCAL パターンが含まれていないログ・レコードだけを表示するには、以下のように入力します。

```
db2diag -gi appid!:=local
```

または

```
db2diag -g appid!:=LOCAL
```

一致しないすべてのレコードが表示されます。アプリケーション ID フィールドを持つメッセージだけを出力するには、以下のように入力します。

```
db2diag -gvi appid:=local -exist
```

- タイム・スタンプ 2003-03-03-12.16.26.230520 のメッセージ以降のログに記録されているすべてのメッセージを表示するには、以下のように入力します。

```
db2diag -time 2003-03-03-12.16.26.230520
```

- 3 日以内にログに記録された重大エラーを表示するには、以下のように入力します。

```
db2diag -gi "level=severe" -H 3d
```

- funcname フィールドが pdLog パターンに一致しないすべてのログ・メッセージを表示するには、以下のように入力します。

```
db2diag -g 'funcname!=pdLog'
```

## db2diag - db2diag ログ分析ツール

または

```
db2diag -gv 'funcn=pdLog'
```

- `base sys` で始まるコンポーネント名を含むすべての重大エラー・メッセージを表示するには、以下のように入力します。

```
db2diag -l severe | db2diag -g "comp^=base sys"
```

- `db2diag.log` ファイルへの追加分を表示するには、次のように入力します。

```
db2diag -f db2diag.log
```

このようにすると、現行ディレクトリーの `db2diag.log` ファイルに書き込まれるすべてのレコードが表示されます。レコードは、ファイルに追加されるときに表示されます。この情報は、`Ctrl-C` を押すまで表示されません。

- `db2diag.log` のコンテキストを、`/home/user/Logs` ディレクトリーにある `db2diag_123.log` ファイルに書き込むには、以下のように入力します。

```
db2diag -o /home/user/Logs/db2diag_123.log
```

- デフォルト設定を使用して **db2diag** を Perl スクリプトから呼び出すには、次のように入力します。

```
system("db2diag -readfile");
```

- これは、**diagpath** 構成パラメーターで指定されたディレクトリーからの `db2diag.log/db2diag.*.log` ファイル (データベース・マネージャー **diagsize** 構成パラメーターが設定されている場合には、循環ログ) を処理するように **db2diag** に強制します。

- 端末入力を見捨て、`db2diag.log1` ファイルを指定されたディレクトリーから読み取るには、次のように入力します。

```
system("db2diag -readfile /u/usr/sql1lib/db2dump/db2diag.log1");
```

- `ecfId = 0x1C30000E` に相当する関数情報を表示するには、次のように入力します。

```
db2diag -ecfid 0x1C30000E
```

これは以下のものと同等です。

```
db2diag -ecfid 472907790
```

これにより、関数名、コンポーネント、および製品名が表示されます。

- `eduid = 123` を含むログ・レコードだけを表示するには、次のように入力します。

```
db2diag -eduid 123
```

- `eduid = 123` または `eduid = 5678` を含むすべてのレコードを表示するには、次のように入力します。

```
db2diag -eduid "123,5678"
```

- `eduid = 15` を持つスレッドによって生成されたすべての重大エラー・メッセージを表示するには、次のように入力します。

```
db2diag -g "level=Severe, eduid=15"
```

あるいは、以下と同等です。

```
db2diag -g level=Severe | db2diag -eduid 15
```

- データベース・パーティション 1 からの最新の 5 つのフォーマット済みレコードを表示するには、以下のように入力します。



```
db2diag -lastrecords 5 -node 1 -fmt "%{ts} %{node}"
```

- すべての db2diag.log ファイルから最後の 10 行を読み取るには、以下のようになります。

```
db2diag -lastlines 10
```

- 各ログ・ファイルの最後の 20 行に含まれているレコードをマージするには、以下のようになります。

```
db2diag -merge file1 file2 file3... -lastlines 20
```

- すべてのホストの各 db2diag.log ファイルの最後の 20 行に含まれているレコードを表示するには、以下のようになります。

```
db2diag -global -lastlines 20
```

- 最後の 100 行に含まれているレコードのうち、Level=Error のレコードをすべて表示するには、以下のようになります。

```
db2diag -g level=Error -lastlines 100
```

## 使用上の注意

- 各オプションは一度だけ指定できます。オプションは任意の順番で指定でき、オプションのパラメーターを指定することができます。短いオプションを相互に組み合わせることはできません。例えば、**-l -e** を **-le** とすることはできません。
- デフォルトでは、**db2diag** は現行ディレクトリーで **db2diag** ログ・ファイルを探します。ファイルが検出されない場合、次に **diagpath** 構成パラメーターによって設定されたディレクトリーが検索されます。そこでも **db2diag** ログ・ファイルが検出されない場合、**db2diag** はエラーを戻し、終了します。
- フィルター操作オプションとフォーマット設定オプションを単一のコマンド行に組み合わせると、パイプを使用した複雑な検索を実行できます。フォーマット設定オプションである **-fmt**、**-strict**、**-cbe**、および **-verbose** は、標準的なフィールドを持つ元のログ・メッセージだけをフィルター操作して、ユーザーによって定義または省略されたフィールドを持つものは処理されないようにするために、すべてのフィルター操作の完了後に使用する必要があります。パイプを使用するときには、**-** を使用する必要はありません。
- パイプが使用され、1 つ以上のファイル名がコマンド行に指定される場合、**-** が指定されているかどうかによって、**db2diag** 入力の処理方法が異なります。**-** が省略されている場合、入力は指定のファイルからとられます。それに対して、**-** オプションが指定されている場合、ファイル名は無視され (コマンド行に指定されている場合でも)、端末からの入力を使用されます。パイプが使用され、ファイル名が指定されない場合、**db2diag** の入力は、コマンド行に **-** が指定されるかどうかに関係なく、同じ方法で処理されます。
- **-exist** オプションは、パターンに一致しないすべてのレコードが (適切なフィールドを含んでいるかどうかに関係なく) 出力される場合に、反転一致検索のデフォルト **db2diag** の動作をオーバーライドします。**-exist** オプションが指定される場合、要求されたフィールドを含むレコードだけが処理されて出力されます。
- **-fmt** (フォーマット設定) オプションが指定されない場合、すべてのメッセージ (フィルター操作されたものとされないもの) は診断ログ・ファイルに書き込まれるときに出力されます。出力レコード・フォーマットは、**-strict**、**-cbe**、および **-verbose** オプションを使用して変更できます。

## db2diag - db2diag ログ分析ツール

- **-fmt** オプションは、**-strict**、**-cbe**、および **-verbose** オプションをオーバーライドします。
- **-cbe** オプションが指定され、**db2diag** ログ・ファイルが元のコンピューターからネットワーク上で転送された場合、いくつかの制限が適用されます。**db2diag** ツールは、DB2 とコンピューター・ホスト名に関する情報をローカルで収集します。そのため、ローカル・システムの DB2 のバージョンおよびソースまたは報告機能 **componentID** の位置フィールドは、元のコンピューターで使用された対応する値と異なる可能性があります。
- **db2diag** をスクリプトで使用する際には、**-readfile** オプションを指定することが推奨されています。これで、端末入力を無視して、ファイルから読み取るようにします。
- 通常、一致が検出された場合、終了状況は 0 であり、一致が検出されない場合 1 です。入力データおよびパターンに構文エラーがある、入力ファイルがアクセス不能である、または他のエラーが検出される場合、終了状況は 2 です。
- DB2 Text Search の結果として発生する重大エラーは、**db2diag** ログ・ファイルのログに記録されています。
- 循環 **db2diag** ログ・ファイルを読み取ってフィルター処理するためにこのツールを使用する場合 (**diagsize** データベース構成パラメーターが非ゼロの場合)、すべての循環診断ログ・ファイル (最大 10 個の一連のファイル) が読み取られてフィルター処理されます。
- リストアの進行中にデータベースに接続しようとする時、**db2diag.log** にエラー・メッセージが書き込まれます。リストア・コマンドが成功していれば、そのようなエラー・メッセージは無視できます。

## db2drdat - DRDA トレース

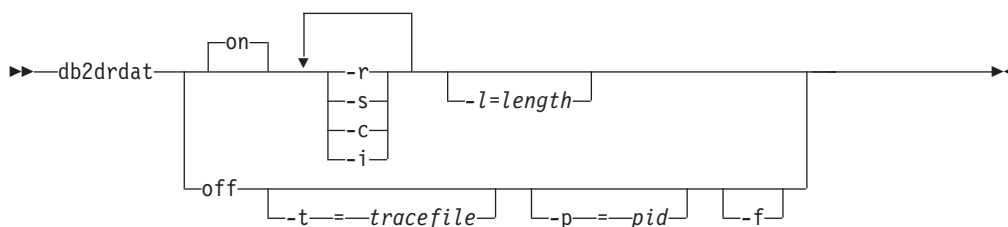
DB2 DRDA アプリケーション・リクエスター (AR) および DRDA アプリケーション・サーバー (AS) 間で交換された DRDA データ・ストリームをキャプチャーできます。

このツールは、アプリケーションの実行に必要な送信および受信の回数を判別することによって、問題判別でよく使用されますが、クライアント/サーバー環境でのパフォーマンス調整にも使用することができます。

### 許可

なし

### コマンド構文



### コマンド・パラメーター

- on** AS トレース・イベント (指定しない場合はすべて) をオンにします。
- off** AS トレース・イベントをオフにします。
- r** DRDA AR から受信した DRDA 要求をトレースします。
- s** DRDA AR に送信された DRDA 応答をトレースします。
- c** ホスト・システムの DRDA サーバーから受信した SQLCA をトレースします。これは、様式化して読みやすくした非 *NULL* SQLCA です。
- i** トレース情報にタイム・スタンプを含めます。
- l** トレース情報を格納するために使用されるバッファのサイズを指定します。
- p** このプロセスについてのみイベントをトレースします。 **-p** を指定しない場合、サーバー上の着信 DRDA 接続をもつエージェントがすべてトレースされます。トレースする *pid* は、**LIST APPLICATIONS** コマンドによって戻される *agent* フィールドにあります。
- t** トレースの宛先を指定します。ファイル名で、完全なパス名が指定されていない場合、脱落情報は現行パスから取られます。 *tracefile* が指定されていない場合、メッセージは現行ディレクトリーの *db2drdat.dmp* に送られます。
- f** 通信バッファをフォーマットします。

## 使用上の注意

**db2drdat** がアクティブな場合、 **db2trc** コマンドを発行しないでください。

**db2drdat** は、以下の情報を *tracefile* に書き込みます。

1. **-r**
  - DRDA 要求のタイプ
  - 受信バッファ
2. **-s**
  - DRDA 応答/オブジェクトのタイプ
  - 送信バッファ

コマンドは終了コードを戻します。ゼロ値はコマンドが正常に完了したことを示します。非ゼロ値はコマンドが正常に完了しなかったことを示します。 **db2drdat** が既に存在するファイルに出力を送信する場合、ファイルの権限で古いファイルの消去が禁止されているのではない限り、古いファイルは消去されます。古いファイルの消去が禁止されている場合は、オペレーティング・システムがエラーを返します。

## db2drvmp - DB2 データベース・ドライブのマップ

Microsoft Cluster Server (MSCS) のデータベース・ドライブをマップします。このコマンドは、Windows オペレーティング・システムでのみ使用できます。

### 許可

Windows レジストリーおよびクラスター・レジストリーに対する読み取り/書き込みアクセス。

### 必要な接続

インスタンス。デフォルトのインスタンス接続が存在しない場合は、アプリケーションによって作成されます。

### コマンド構文

```

▶▶ db2drvmp add dbpartition_number from_drive to_drive
    drop
    query
    reconcile

```

### コマンド・パラメーター

**add** 新しいデータベース・ドライブ・マップを割り当てます。

**drop** 既存のデータベース・ドライブ・マップを削除します。

**query** データベース・マップを照会します。

#### reconcile

レジストリーの内容が損傷を受けたり、意図せずにドロップされたりした場合に、レジストリーにデータベース・ドライブ・マッピングを再適用します。

#### dbpartition\_number

データベース・パーティション番号。このパラメーターは、追加およびドロップ操作に必要です。調整操作にこのパラメーターが指定されない場合、**db2drvmp** はすべてのデータベース・パーティションのマッピングを調整します。

#### from\_drive

マップ元のドライブ名。このパラメーターは、追加およびドロップ操作に必要です。調整操作にこのパラメーターが指定されない場合、**db2drvmp** はすべてのドライブのマッピングを調整します。

#### to\_drive

マップ先のドライブ名。このパラメーターは、追加操作に必要です。これは、他の操作には該当しません。

### 例

NODE0 に関して、データベース・ドライブを F: から E: にマッピングするには、以下のコマンドを入力します。

```
db2drvmp add 0 F E
```

## db2drvmp - DB2 データベース・ドライブのマップ

NODE1 に関して、データベース・ドライブを E: から F: にマッピングするには、以下のコマンドを入力します。

```
db2drvmp add 1 E F
```

### 使用上の注意

1. データベース・ドライブ・マッピングは、表スペース、コンテナ、または他のデータベース・ストレージ・オブジェクトには適用されません。
2. データベース・ドライブ・マッピングに対する設定または変更は、即時には有効になりません。データベース・ドライブ・マッピングを有効にするには、Microsoft Cluster Administrator ツールを使用して、DB2 リソースをオフラインにしてから、オンラインにします。
3. **TARGET\_DRVMAP\_DISK** キーワードを DB2MSCS.CFG ファイルに使用すると、ドライブ・マッピングを自動的に実行できます。

## db2empfa - 複数ページ・ファイル割り振りの有効化

データベースの複数ページ・ファイル割り振りの使用を有効にします。SMS 表スペースでの複数ページ・ファイル割り振りを有効にすると、ディスク・スペースは、一度に 1 ページではなく、一度に 1 エクステント割り振られます。

### 有効範囲

このコマンドは、それが実行されたデータベース・パーティションに対してだけ影響を与えます。

### 許可

SYSADM

### 必要な接続

なし。このコマンドは、データベース接続を確立します。

### コマンド構文

```
▶▶—db2empfa—database-alias—————▶▶
```

### コマンド・パラメーター

*database-alias*

複数ページ・ファイル割り振りを有効にするデータベースの別名を指定します。

### 使用上の注意

このユーティリティーは以下の処理を行います。

- (適用可能な) データベース・パーティションに排他モードで接続する
- すべての SMS 表スペースでは、空のページを割り振り、1 エクステントより大きなすべてのデータおよび索引ファイルで最後のエクステントを埋め込む
- データベース構成パラメーター **multipage\_alloc** の値を YES に変更する
- 切断する

**db2empfa** は、データベース・パーティションに排他モードで接続するので、カタログ・データベース・パーティションまたは他のどのデータベース・パーティションでも同時に実行することはできません。

## db2envar.bat - 現行コマンド・ウィンドウの環境の設定

**db2envar.bat** の実行元となる、DB2 コピーの現行コマンド・ウィンドウの環境を設定します。これは、コマンド行から異なる DB2 コピーを切り替える場合に役立ちます。

このコマンドは Windows オペレーティング・システムでのみ使用できます。

### 許可

なし

### 必要な接続

なし

### コマンド構文

▶▶—db2envar.bat—◀◀

### コマンド・パラメーター

なし

### 使用上の注意

マシン上に複数の DB2 コピーがある場合、絶対パスを使用して、どの **db2envar.bat** が実行されるのかを示す必要があります。例えば、e:¥sql1lib の下にインストールされている DB2 コピーの環境をセットアップする場合、e:¥sql1lib¥bin¥db2envar.bat を発行します。



## db2evmon - イベント・モニター生産性向上ツール

イベント・モニター・ファイルと Named PIPE をフォーマットし、それを標準出力に書き込みます。

### 許可

なし。ただし、データベースに接続している場合には (**-db -evm**)、以下のすべての権限が必要です。

- CONNECT 権限 (または CONNECT を暗黙的に含む権限)
- 以下のカタログ表に対する SELECT 特権 (それらのカタログ表に対する SELECT を暗黙的に含む権限)。
  - SYSIBM.SYSTABLES
  - SYSIBM.SYSEVENTMONITORS

イベント・モニターが db2detaildeadlock である場合、以下の権限または特権のいずれかが追加で必要になります。

- SYSMON
- SYSMOINT
- SYSCTRL
- SYSADM
- SNAPSHOT\_DATABASE 表関数に対する EXECUTE 特権。
- DATAACCESS

### 必要な接続

なし

### コマンド構文

```

▶▶ db2evmon [ -db database-alias -evm event-monitor-name ]
               [ -path event-monitor-target ]

```

### コマンド・パラメーター

#### **-db** *database-alias*

表示するデータのあるデータベースを指定します。このパラメーターには、大文字と小文字の区別があります。

#### **-evm** *event-monitor-name*

イベント・モニターの 1 部構成の名前です。普通の、または区切り SQL ID です。このパラメーターには、大文字と小文字の区別があります。

#### **-path** *event-monitor-target*

イベント・モニター・トレース・ファイルを含むディレクトリーを指定します。

### 使用上の注意

**db2evmon** コマンドは、データベースに接続して発行されたか、**path** オプションを指定して発行されたかに関係なく、同じ出力を生成します。

- インスタンスがまだ開始されていない状態で、**-db** および **-evm** オプションを指定して **db2evmon** を発行すると、そのコマンドがインスタンスを開始します。
- インスタンスがまだ開始されていない状態で、**-path** オプションを指定して **db2evmon** を発行しても、そのコマンドはインスタンスを開始しません。インスタンスは明示的に開始する必要があります。

データがファイルに書き込まれている場合、このツールは標準出力を使用した表示のためにファイルをフォーマットします。この場合、最初にモニターがオンになり、次にこのツールによってファイル中のイベント・データが表示されます。このツールを実行した後にファイルに書き込まれたデータをすべて表示させるには、**db2evmon** を再発行します。

データがパイプに書き込まれている場合、イベントが起きた時点で、このツールは標準出力を使用した表示のために出力をフォーマットします。この場合、ツールが開始した後 に、モニターがオンになります。

## db2evtbl - イベント・モニターのターゲット表定義の生成

CREATE EVENT MONITOR SQL ステートメントのサンプルを生成します。このステートメントは、SQL 表に書き込みを行うイベント・モニターを定義するときに使用できます。

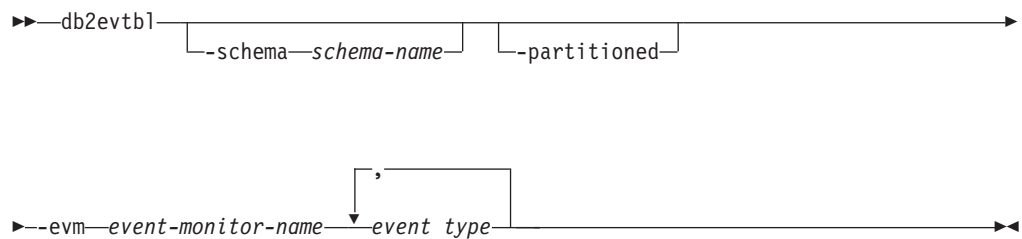
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

**-schema** *schema-name*

スキーマ名。指定しない場合、表の名前は修飾されません。

**-partitioned**

指定した場合は、パーティション・データベース環境だけに該当するイベントも生成されます。

**-evm** *event-monitor-name*

イベント・モニターの名前。

*event type*

CREATE EVENT MONITOR ステートメントの FOR 節に使用できるいずれかのイベント・タイプ。有効な値は以下のとおりです。

- ACTIVITIES
- BUFFERPOOLS
- CONNECTIONS
- CHANGE HISTORY
- DATABASE
- DEADLOCKS (DEADLOCKS WITH DETAILS、DEADLOCKS WITH DETAILS HISTORY、DEADLOCKS WITH DETAILS HISTORY VALUES も可)\*
- LOCKING
- PACKAGE CACHE
- STATEMENTS

## db2evtb1 - イベント・モニターのターゲット表定義の生成

- STATISTICS
- TABLES
- TABLESPACES
- THRESHOLD VIOLATIONS
- TRANSACTIONS\*
- UNIT OF WORK

\* このイベント・モニター・タイプは推奨されていません。

### 例

```
db2evtb1 -schema smith -evm test01 database, tables, tablespaces, bufferpools
```

### 使用上の注意

出力は標準出力に書き込まれます。

**db2evtb1** ツールを使用すると、WRITE TO TABLE イベント・モニターの定義がより簡単になります。例えば、イベント・モニターを定義およびアクティブ化するために、次の手順を実行できます。

1. **db2evtb1** を使って、CREATE EVENT MONITOR ステートメントを生成します。
2. SQL ステートメントを編集し、不必要な列を除去します。
3. CLP を使用して、SQL ステートメントを処理します。(CREATE EVENT MONITOR ステートメントを実行すると、ターゲット表が作成されます。)
4. SET EVENT MONITOR STATE を発行して、新しいイベント・モニターをアクティブ化します。

デッドロック・イベント・モニター以外のすべてのイベントは、フラッシュ可能であり、1 つのイベントにつき複数のレコードが作成されます。そのため、FLUSH EVENT MONITOR ステートメントを使用しないユーザーは、エレメント **evmon\_flushes** をターゲット表に入れる必要はありません。

LOCKING、UNIT OF WORK、または PACKAGE CACHE のイベント・モニターが、作成されるイベント・モニターに含まれている場合、**db2evtb1** によって生成される DDL は、UE 表ではなく通常の表を作成します。

## db2exfmt - Explain 表フォーマット

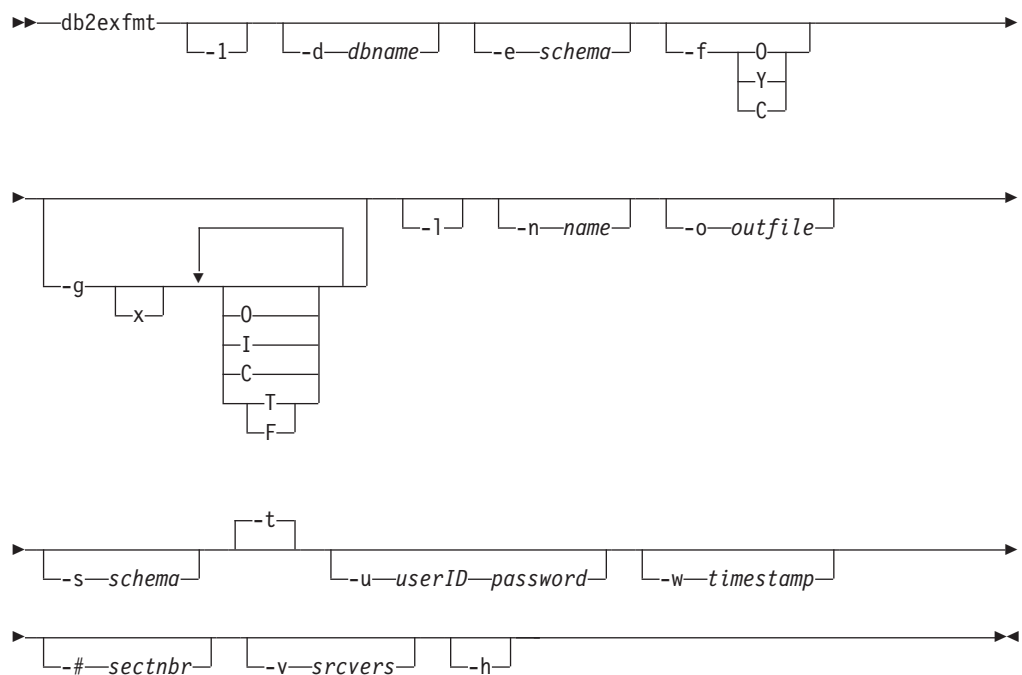
EXPLAIN 表の内容をフォーマットします。

このツールは、インスタンスの `sqllib` ディレクトリーの `misc` サブディレクトリーにあります。EXPLAIN スナップショットが使用可能であれば、このツールは EXPLAIN スナップショットの統計を使用します。

### 許可

このツールを使用するには、フォーマットする Explain 表に対する読み取りアクセスが必要です。

### コマンド構文



### コマンド・パラメーター

#### db2exfmt

オプションを指定しない場合、コマンドは対話モードに入り、ユーザーは入力するよう促されます。

**-l** デフォルト `-e % -n % -s % -v % -w -l -# 0` を使用します

Explain スキーマが提供されない場合、環境変数 `$USER` または `$USERNAME` の内容がデフォルトとして使用されます。この変数が見つからない場合は、Explain スキーマを指定するよう求めるプロンプトが出されます。

**-d dbname**

パッケージを含むデータベースの名前。

**-e schema**

Explain 表 SQL スキーマ。

## db2exfmt - Explain 表フォーマット

**-f** フォーマット・フラグ。複数のフラグを、ストリングとして結合することができます。例えば、旧バージョンの DB2 データベース製品と同様の出力を得るには、C オプションと Y オプションを結合して -f CY にすることができます。

**O** 演算子の要約。

**Y** 列 EXPLAIN\_STATEMENT.EXPLAIN\_TEXT にフォーマット設定が含まれている場合であっても、元のステートメントを強制的にフォーマット設定します。デフォルトの動作は、ステートメントのフォーマット設定が必要かどうかを自動的に検出し、元のフォーマット設定が存在するのであれば、それを使用します。

**C** ステートメントと述部をフォーマット設定する際によりコンパクトなモードを使用します。デフォルトは、読みやすい拡張モードです。Y が指定されていない場合、自動検出によってステートメントのフォーマット設定が必要であると判断された場合にのみ、C が効力を持ちます。

**-g** グラフ・プラン。

**x** オプションを OFF にします (デフォルトは ON です)。

**-g** だけを指定した場合は、グラフが生成され、その後すべての表に関するフォーマット済みの情報が生成されます。それ以外の場合は、以下の有効な値を任意に組み合わせて指定できます。

**O** グラフだけを生成します。表の内容はフォーマットしません。

**T** グラフ内の各演算子の下に合計コストを組み込みます。

**F** グラフ内の最初のタプルのコストを組み込みます。

**I** グラフ内の各演算子の下に I/O コストを組み込みます。

**C** グラフ内の各演算子の予期出力カーディナリティー (タプル数) を組み込みます。

これらのオプションを任意に組み合わせることができますが、相互に排他的な F と T は例外です。

**-l** パッケージ名の処理時に大文字小文字を区別します。

**-n name**

Explain 要求のソース名 (SOURCE\_NAME)。

**-s schema**

Explain 要求のソースの SQL スキーマまたは修飾子 (SOURCE\_SCHEMA)。

**-o outfile**

出力ファイル名。

**-t** 出力を端末に送信します。

**-u userID password**

データベースに接続時に、指定のユーザー ID とパスワードを使用します。

ユーザー ID とパスワードはいずれも、命名規則に従った有効な値でなければならず、データベースによって認識される値でなければなりません。

**-w** *timestamp*

Explain タイム・スタンプ。 **-1** を指定すれば、最新の Explain 要求を取得できます。

**-#** *sectnbr*

ソース内のセクション番号。すべてのセクションを要求するには、ゼロを指定します。

**-v** *srcvers*

Explain 要求のソースのソース・バージョン (デフォルトは %)

**-h**

ヘルプ情報を表示します。このオプションを指定すると、他のすべてのオプションは無視され、ヘルプ情報のみが返されます。

**使用上の注意**

パラメーター値を指定しなかった場合や、指定内容が完全でない場合は、値の入力を促すプロンプトが表示されます。ただし、**-h** オプションと **-1** オプションは除きます。

Explain 表 SQL スキーマを指定しない場合は、環境変数 **USER** の値がデフォルトとして使用されます。この変数が見つからない場合、Explain 表 SQL スキーマを指定するよう求めるプロンプトが出されます。

ソース名、ソース SQL スキーマ、Explain タイム・スタンプは、LIKE 述部形式で指定できます。この場合、パーセント記号 (%) と下線 ( ) をパターン・マッチング文字として使用して、1 つの呼び出しで複数のソースを選択できます。EXPLAIN された最新のステートメントの場合は、Explain タイム・スタンプを **-1** と指定できます。

**-o** をファイル名なしで指定し、**-t** を指定しない場合は、ファイル名の入力を促すプロンプトが表示されます (デフォルト名は **db2exfmt.out**)。 **-o** も **-t** も指定しない場合は、ファイル名の入力を求めるプロンプトが出されます (デフォルト・オプションは端末出力です)。 **-o** と **-t** の両方を指定した場合は、出力が端末に送信されます。

Explain スナップショットが使用可能であれば、**db2exfmt** コマンドは Explain スナップショットの統計を返します。使用可能でなければ、**db2exfmt** は、EXPLAIN\_OBJECT 表に保管されている統計と、システム・カタログから直接取得された統計を返します。

パーティションごとの使用量、転送速度、およびプリフェッチ・サイズとして返される値は、**db2exfmt** コマンドの実行時に取得されます。そのため、これらの値はステートメントが Explain されたときに使用された実際の値とは異なる可能性があります。

**db2exfmt** 出力に返される OVERHEAD 値、TRANSFERRATE 値および PREFETCHSIZE 値は、ステートメントがコンパイルされたときに使用された実際の値とは異なる可能性があります。

**例**

EXPLAIN スナップショットの例です。

## db2exfmt - Explain 表フォーマット

```
db2 explain plan with snapshot for query
db2exfmt
```

または、

```
db2 set current explain mode yes
db2 set current explain snapshot yes
query
db2exfmt
```



## db2exmig - Explain 表のマイグレーション

Explain 表をマイグレーションします。 Explain 表のマイグレーション・ツールは、既存の Explain 表を名前変更し、EXPLAIN.DDL を使用して表の新規セットを作成し、さらに既存の Explain 表の内容を新規の表にコピーします。最後に、既存の Explain 表をドロップします。

**db2exmig** コマンドを発行しているか、またはデータベースへの接続に使用されるユーザー ID に属する Explain 表がマイグレーションされます。

**db2exmig** コマンドは、Explain 表にあるユーザーが追加した列を保存します。

### 許可

**db2exmig** アプリケーション・パッケージが結合されている場合、必要となる許可は以下のいずれかの権限です。

- DBADM 権限
- **db2exmig** アプリケーション・パッケージでの EXECUTE 権限、および以下のシステム・カタログでの SELECT 特権または CONTROL 特権。
  - SYSCAT.COLUMNS
  - SYSCAT.TABLES
  - SYSCAT.REFERENCES

**db2exmig** アプリケーション・パッケージがバインドされていない場合に必要な許可は、DBADM 権限です。

### 必要な接続

なし

### コマンド構文

```
▶▶ db2exmig -d dbname -e explain_schema -u userID password ▶▶
```

### コマンド・パラメーター

- d *dbname*  
データベース名を指定します。
- e *explain\_schema*  
マイグレーションされる Explain 表のスキーマ名を指定します。
- u *userID password*  
現行ユーザーの ID およびパスワードを指定します。

### 使用上の注意

**db2exmig** アプリケーション・パッケージ名は、次のコマンドを使用して判別できます: db2bfd -b db2exmig.bnd。 db2exmig.bnd ファイルは、sqllib/bnd フォルダー内にあります。

## db2expln - SQL および XQuery Explain

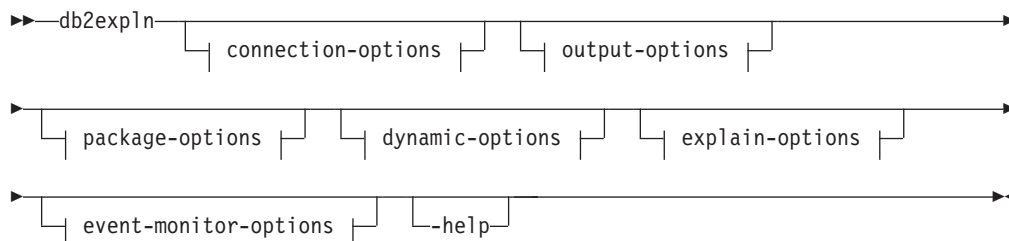
**db2expln** ツールは、SQL ステートメントおよび XQuery ステートメント用に選択されたアクセス・プランを記述します。EXPLAIN データがキャプチャーされなかったときに、このツールを使用して、選択したアクセス・プランの簡単な説明を取得します。静的 SQL ステートメントおよび XQuery ステートメントの場合、**db2expln** は、システム・カタログ表に保管されたパッケージを調べます。動的 SQL ステートメントおよび XQuery ステートメントの場合、**db2expln** は、照会キャッシュのセクションを調べます。

### 許可

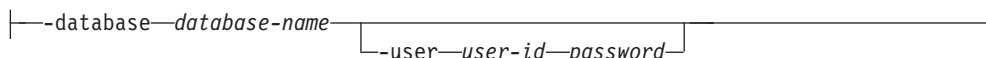
DBADM または以下のいずれかの権限か特権が必要です。

- 静的ステートメントの場合、カタログ表に対する SELECT 特権
- 動的ステートメントの場合、カタログ表に対する SELECT 特権と、以下のいずれかの権限か特権
  - ステートメントをコンパイルするための十分な特権
  - EXPLAIN 権限
  - SQLADM 権限

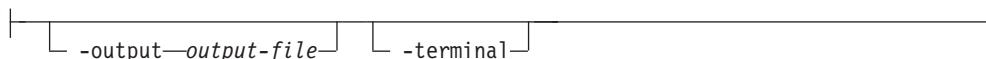
### コマンド構文



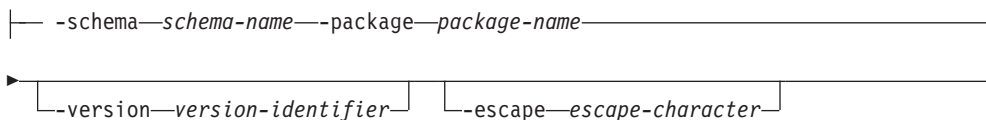
#### connection-options:

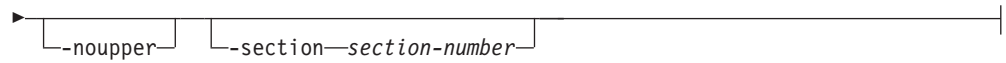
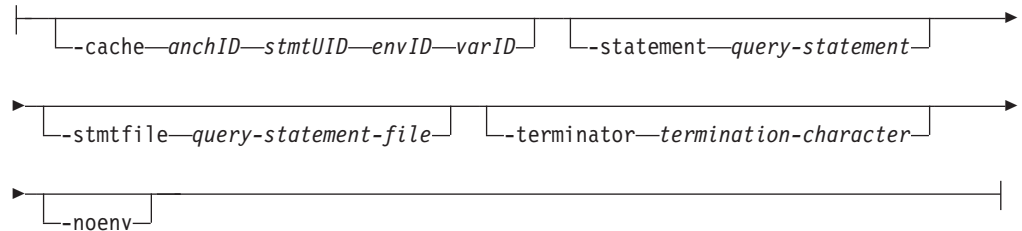
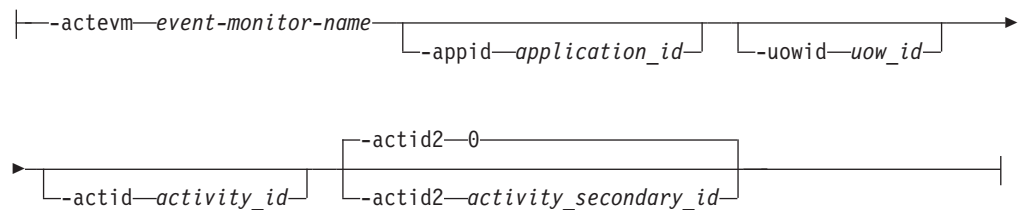


#### output-options:



#### package-options:



**dynamic-options:****explain-options:****event-monitor-options:****コマンド・パラメーター**

オプションは任意の順序で指定できます。

**connection-options:**

これらのオプションでは、接続先のデータベースと、その接続のために必要なオプションを指定します。接続オプションは、**-help** オプションを指定する場合以外は必須です。

**-database database-name**

Explain の対象パッケージが入っているデータベースの名前。

後方互換性のために、**-database** の代わりに **-d** を使用できます。

**-user user-id password**

データベース接続を確立するとき使用する許可 ID とパスワード。 *user-id* と *password* はいずれも、DB2 命名規則に従った有効な値でなければならず、データベースによって認識される値でなければなりません。

後方互換性のために、**-user** の代わりに **-u** を使用できます。

**output-options:**

## db2expln - SQL および XQuery Explain

これらのオプションでは、**db2expln** 出力の送信先を指定します。 **-help** オプションを指定する場合を除き、最低 1 つの出力オプションを指定しなければなりません。両方のオプションを指定すると、出力はファイルと端末の両方に送信されます。

### **-output** *output-file*

**db2expln** の出力は、指定したファイルに書き込まれます

後方互換性のために、**-output** の代わりに **-o** を使用できます。

### **-terminal**

**db2expln** 出力は、端末に送信されます。

後方互換性のために、**-terminal** の代わりに **-t** を使用できます。

### **package-options:**

これらのオプションでは、**Explain** の対象として 1 つ以上のパッケージとセクションを指定します。それらのパッケージとセクションの中にある静的照会だけが **Explain** の対象になります。

**LIKE** 述部の場合と同じく、パターン・マッチング文字としてパーセント記号 (%) と下線 ( \_ ) を使用して、 *schema-name*、 *package-name*、 *version-identifier* を指定できます。

### **-schema** *schema-name*

**Explain** 対象の 1 つ以上のパッケージの **SQL** スキーマ。

後方互換性のために、**-schema** の代わりに **-c** を使用できます。

### **-package** *package-name*

**Explain** の対象パッケージ (複数可) の名前。

後方互換性のために、**-package** の代わりに **-p** を使用できます。

### **-version** *version-identifier*

**Explain** の対象パッケージ (複数可) のバージョン ID。デフォルトのバージョンは、空ストリングです。

### **-escape** *escape-character*

*schema-name*、 *package-name*、 *version-identifier* でパターン・マッチングのエスケープ文字として使用する文字 *escape-character*。

例えば、パッケージ **TESTID.CALC%** を **Explain** する **db2expln** コマンドは、次のとおりです。

```
db2expln -schema TESTID -package CALC% ....
```

ただし、このコマンドは、**CALC** で始まる他のプランも **Explain** します。 **TESTID.CALC%** パッケージだけを **Explain** するには、エスケープ文字を使用しなければなりません。感嘆符 (!) をエスケープ文字として指定した場合のコマンドは、 **db2expln -schema TESTID -escape ! -package CALC!% ...** になります。この場合、**!** 文字をエスケープ文字として使用しているので、**!%** は「何にでもマッチングする」パターンではなく、**%** 文字として解釈されます。デフォルトのエスケープ文字はありません。

後方互換性のために、**-escape** の代わりに **-e** を使用できます。

問題を避けるため、オペレーティング・システムのエスケープ文字を **db2expln** のエスケープ文字として指定しないでください。

#### **-noupper**

マッチングするパッケージを検索する前に、*schema-name*、*package-name*、*version-identifier* を大文字に変換しないことを指定します。

デフォルトでは、パッケージの検索前に、これらの変数が大文字に変換されます。このオプションを指定すると、これらの値は入力のとおりに入力されます。

後方互換性のために、**-noupper** の代わりに **-1** (数字の 1 ではなく L の小文字) を使用できます。

#### **-section** *section-number*

選択したパッケージ (複数可) 内で Explain の対象にするセクションの番号。

各パッケージ内のすべてのセクションを Explain するには、数値ゼロ (0) を使用します。これがデフォルトの動作です。このオプションを指定しない場合や、*schema-name*、*package-name*、*version-identifier* のいずれかにパターン・マッチング文字を含めていない場合は、すべてのセクションが表示されます。

セクション番号を確認するには、システム・カタログ・ビュー SYSCAT.STATEMENTS を照会してください。システム・カタログ・ビューの説明については、「SQL リファレンス」を参照してください。

後方互換性のために、**-section** の代わりに **-s** を使用できます。

#### **dynamic-options:**

これらのオプションでは、Explain の対象である 1 つ以上の動的照会ステートメントを指定します。

#### **-cache** *anchID, stmtUID, envID, varID*

所定の識別子 (ID) で識別されるステートメントを取り出す動的 SQL キャッシュを指定します。この ID は、**db2pd** コマンドに **-dynamic** オプションを指定して使用すると取得できます。

#### **-statement** *query-statement*

Explain の対象として動的に準備される SQL または XQuery 照会ステートメント。複数のステートメントを Explain するには、**-stmtfile** オプションを使用することにより、Explain 対象の照会ステートメントを含むファイルを指定するか、**-terminator** オプションを使用することにより、**-statement** オプション内のステートメントを区切る終了文字を定義します。

#### **-stmtfile** *query-statement-file*

Explain の対象として動的に生成する 1 つ以上の照会ステートメントを含むファイル。デフォルトでは、ファイルの各行が別個の照会ステートメントと見なされます。ステートメントが複数行になる場合は、**-terminator** オプションを使用して、照会ステートメントの終了を示す文字を指定してください。

### **-terminator** *termination-character*

動的照会ステートメントの終わりを示す文字。デフォルトでは、**-statement** オプションの値は 1 つの照会ステートメントと見なされ、**-stmtfile** のファイルの各行は別個の照会ステートメントと見なされます。ここで指定する終了文字を使用して、**-statement** で複数の照会ステートメントを指定したり、**-stmtfile** ファイル内のステートメントを複数行にしたりすることができます。

### **-noenv** コンパイル環境を変更する動的ステートメントを Explain 後に実行しないことを指定します。

デフォルトでは、**db2expln** は、以下のステートメントを Explain 後に実行します。

```
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
```

これらのステートメントを実行すれば、**db2expln** の処理対象になる後続の動的照会ステートメントのために選択されているプランを変更できます。

**-noenv** を指定した場合、これらのステートメントは Explain されますが、実行されません。

動的照会を Explain するには、**-statement** または **-stmtfile** のいずれかを指定する必要があります。両方のオプションを **db2expln** の 1 つの呼び出しで指定できます。

### **explain-options:**

これらのオプションでは、Explain されたプランで提供する追加の情報を指定します。

### **-graph** オプティマイザー・プラン・グラフを表示します。各セクションが調査された後、元のオプティマイザー・プラン・グラフが構成されます。セクション・プランに含まれる情報に応じて、オプティマイザー・グラフにギャップが表示される可能性があります。

後方互換性のために、**-graph** の代わりに **-g** を使用できます。

### **-opids** Explain されるプラン内に演算子 ID 番号を表示します。

演算子 ID 番号によって、**db2expln** からの出力と Explain 機能からの出力との対応関係を示すことができます。ただし、すべての演算子が ID 番号を持つとは限らず、Explain 機能の出力に現れるいくつかの ID 番号は、**db2expln** の出力には現れません。

後方互換性のために、**-opids** の代わりに **-i** を使用できます。

### **-help** **db2expln** のヘルプ・テキストを表示します。このオプションを指定した場合、パッケージは Explain されません。

ほとんどのコマンド行は、**db2exsrv** ストアード・プロシージャで処理されます。使用できるすべてのオプションのヘルプを表示す

るには、**-help** と一緒に **connection-options** を指定する必要があります。例えば、次のように使用します。

```
db2expln -help -database SAMPLE
```

後方互換性のために、**-h** または **-?** を指定できます。

**-setup** *setup-file*

再コンパイルする必要のある動的ステートメントまたは静的ステートメントの環境をセットアップするのに必要な 1 つ以上のステートメントが含まれるファイル (宣言済み一時表を参照する静的ステートメントなど)。ファイル内の各ステートメントが実行され、エラーまたは警告があれば報告されます。ファイル内のステートメントは Explain されません。

**event-monitor-options:**

これらのオプションでは、Explain の対象として、アクティビティ・イベント・モニターから 1 つ以上のセクション環境を指定します。

**-actevm** *event-monitor-name*

アクティビティ・イベント・モニターの名前で、このモニターには、Explain の対象である、(**section\_env** モニター・エレメントにある) セクション環境が含まれる **activitystmt** 論理グループ化を指定します。

**-appid** *application-id*

Explain の対象であるセクション環境がある、アクティビティを発行したアプリケーションを一意的に識別するアプリケーション ID (**appl\_id** モニター・エレメント) を指定します。**-appid** を指定する場合は、**-actevm** を指定する必要があります。

**-uowid** *uow-id*

Explain の対象であるセクション環境がある、作業単位 ID (**uow\_id** モニター・エレメント) を指定します。作業単位 ID は、指定されたアプリケーション内のみで固有です。**-uowid** を指定する場合は、**-actevm** を指定する必要があります。

**-actid** *activity-id*

Explain の対象であるセクション環境がある、アクティビティ ID (**activity\_id** モニター・エレメント) を指定します。アクティビティ ID は、指定された作業単位内のみで固有です。**-actid** を指定する場合は、**-actevm** を指定する必要があります。

**-actid2** *activity-secondary-id*

Explain の対象であるセクション環境がある、アクティビティ 2 次 ID (**activity\_secondary\_id** モニター・エレメント) を指定します。指定されないと、これはデフォルトのゼロになります。**-actid2** を指定する場合は、**-actevm** を指定する必要があります。



### 使用上の注意

**-help** オプションを指定する場合以外は、`package-options` または `dynamic-options` のいずれかを指定しなければなりません。パッケージと動的 SQL の両方を **db2expln** の 1 つの呼び出しで Explain できます。

上にリストされたオプション・フラグのいくつかは、オペレーティング・システムに対して特別な意味を持つことがあるので、**db2expln** コマンド行の値が正確に解釈されない可能性があります。しかし、オペレーティング・システムのエスケープ文字を前に置けば、その種の文字を入力できます。詳細については、オペレーティング・システムの資料を参照してください。オペレーティング・システムのエスケープ文字を **db2expln** のエスケープ文字として間違えて指定しないように注意する必要があります。

**db2expln** によって生成されるヘルプと初期状況メッセージは、標準出力に書き込まれます。Explain ツールによって生成されるすべてのプロンプトと他の状況メッセージは、標準エラーに書き込まれます。Explain テキストは、選択した出力オプションに応じて、標準出力またはファイルに書き込まれます。

**db2expln** によって以下のメッセージが戻る可能性があります。

- 以下のデータベース・パッケージ・パターンのパッケージが見つかりませんでした: バージョン "`<version>`" の "`<creator>.<package>`" (No packages found for database package pattern: "`<creator>.<package>`" with version "`<version>`")

指定のパターンに一致するパッケージがデータベース内にない場合に、このメッセージが出力に表示されます。

- `db2expln.msg` にはバインド・メッセージがあります (Bind messages can be found in `db2expln.msg`)

`db2expln.bnd` のバインドが正常に行われなかった場合に、このメッセージが出力に表示されます。生じた問題の詳細については、現行ディレクトリーの `db2expln.msg` ファイルに記されます。

- 複数パッケージの可能性があるため、セクション番号が 0 (すべてのセクション) にオーバーライドされました (Section number overridden to 0 (all sections) for potential multiple packages)。

**db2expln** によって複数のパッケージが検出される可能性がある場合、このメッセージが出力に表示されます。いずれかのパターン・マッチング文字がパッケージ入力引数または作成者入力引数で使用されていると、このアクションが実行されます。

- `<bind file>` のバインド・メッセージは `<message file>` にあります (Bind messages for `<bind file>` can be found in `<message file>`)

指定のバインド・ファイルのバインドが正常に行われなかった場合に、このメッセージが表示されます。生じた問題の詳細については、データベース・サーバー上の指定のメッセージ・ファイルに記述されます。

- パッケージに適格な静的セクションがありません (No static sections qualify from package)。



指定のパッケージに動的照会ステートメントだけが含まれている、つまり静的セクションがない場合に、このメッセージが出力に表示されます。

- パッケージ "<creator>."<package>", "<version>", が無効です。パッケージを再バインドしてから、db2expln を再実行してください (Package "<creator>."<package>", "<version>", is not valid. Rebind the package and then rerun db2expln)。

指定のパッケージが現在無効な場合には、このメッセージが出力に表示されません。データベース内に有効なパッケージを再作成するためのプランに関して BIND コマンドまたは REBIND コマンドを再発行してから、**db2expln** を再実行します。

以下のステートメントは Explain されません。

- BEGIN/END COMPOUND
- BEGIN/END DECLARE SECTION
- CLOSE カーソル
- COMMIT および ROLLBACK
- CONNECT
- DESCRIBE
- 動的 DECLARE CURSOR
- EXECUTE
- EXECUTE IMMEDIATE
- FETCH
- INCLUDE
- OPEN カーソル
- PREPARE
- SQL 制御ステートメント
- WHENEVER

コンパウンド SQL ステートメント内の各サブステートメントは独自のセクションを持つことができ、こうしたセクションは **db2expln** によって Explain 可能です。

注: **db2expln** コマンドが XQuery ステートメントを除外することはありません。

## 例

**db2expln** の 1 回の呼び出しで複数のプランを Explain する場合は、**-package**、**-schema**、**-version** の各オプションを使用し、LIKE パターンを使用してパッケージと作成者に関するストリング定数を指定します。つまり、下線 (  ) を使用して 1 つの文字を表し、パーセント記号 (%) を使用してゼロ個以上の文字を表します。

SAMPLE という名前のデータベース内のすべてのパッケージのすべてのセクションを Explain して、その結果を my.exp ファイルに書き込むには、以下のようにします。

```
db2expln -database SAMPLE -schema % -package % -output my.exp
```

## db2expln - SQL および XQuery Explain

別の例として、ユーザーが "statements.db2" という名の CLP スクリプト・ファイルを持っていて、そのファイル内のステートメントを Explain するとします。ファイルには、以下のステートメントが含まれています。

```
SET PATH=SYSIBM, SYSFUN, DEPT01, DEPT93@
SELECT EMPNO, TITLE(JOBID) FROM EMPLOYEE@
```

これらのステートメントを Explain するには、以下のコマンドを入力します。

```
db2expln -database DEPTDATA -stmtfile statements.db2 -terminator @ -terminal
```

以下のステートメントを Explain します。

```
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
```

以下のコマンドを使用します。

```
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
```

これは、以下のものを戻します。

```
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

```

## db2extsec - DB2 オブジェクトの権限の設定

更新済みの DB2 データベース・システムのインストール・システムにある DB2 データベース・オブジェクト (ファイル、ディレクトリー、ネットワーク共有、レジストリー・キー、サービスなど) の権限を設定します。

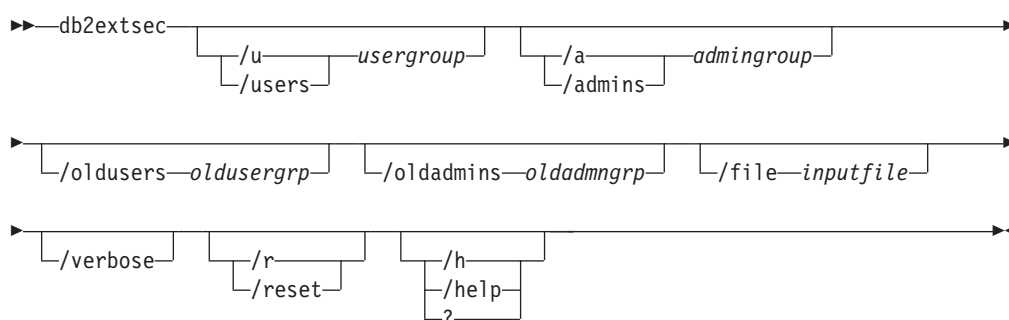
### 許可

SYSADM

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

**/u | /users** *usergroup*

追加するユーザー・グループの名前を指定します。このオプションを指定しない場合は、デフォルトの DB2 ユーザー・グループ (DB2USERS) が使用されます。*usergroup* は、ローカル・グループでもドメイン・グループでもかまいません。ローカル・グループを指定するには、グループ名と共にマシン名を指定してもしなくてもかまいません。例えば、DB2USERS または MYWKSTN¥DB2USERS とします。ドメイン・グループを指定するには、*usergroup* を DOMAIN¥GROUP の形式で指定します。例えば、MYDOMAIN¥DB2USERS とします。

**/a | /admins** *adminingroup*

追加する管理グループの名前を指定します。このオプションを指定しない場合は、デフォルトの DB2 管理グループ (DB2ADMNS) が使用されます。*adminingroup* は、ローカル・グループでもドメイン・グループでもかまいません。ローカル・グループを指定するには、グループ名と共にマシン名を指定してもしなくてもかまいません。例えば、DB2ADMNS または MYWKSTN¥DB2ADMNS とします。ドメイン・グループを指定するには、*adminingroup* を DOMAIN¥GROUP の形式で指定します。例えば、MYDOMAIN¥DB2ADMNS とします。

**注:** 拡張セキュリティー・グループ名を変更しようとしている際に、デフォルトの場所 (つまり、インストール・ディレクトリーまたはデータベース・ディレクトリー) 以外で作成されたファイルまたはディレクトリーのオブジェク

トがある場合は、**/oldusers**、**/oldadmins**、および **/file** の 3 つのパラメーターが必須です。**db2extsec** コマンドで行えるのは、権限を認識済みの DB2 ファイル・セットに変更することだけです。ユーザーが拡張セキュリティのある私用の DB2 ファイルを作成した場合、そのユーザーはこれらのファイルの場所を指定する必要があるため、**db2extsec** コマンドで新しい拡張セキュリティ・グループ名を指定してこれらのファイルに対する権限を変更できます。ファイルの場所は、**/file** オプションを使用した *inputfile* で指定します。

**/oldusers** *oldusergrp*

変更される古い DB2 ユーザー・グループ名。

**/oldadmins** *oldadmngrp*

変更される古い DB2 管理グループ名。

**/file** *inputfile*

権限を更新する必要がある追加のファイル/ディレクトリーをリストするファイル。

**/verbose**

出力追加情報。

**/r | /reset**

以前に実行した **db2extsec** による変更を元に戻すことを指定します。このオプションを指定した場合、他のすべてのオプションは無視されます。このオプションが有効なのは、**db2extsec** コマンドの発行後に他の DB2 コマンドを発行していない場合に限られます。

**/h | /help | ?**

コマンド・ヘルプ情報を表示します。

## 例

拡張セキュリティを使用可能にし、ドメイン・グループ *mydom%db2users* および *mydom%db2admns* を使用して、DB2 オブジェクトを保護するには、以下のようになります。

```
db2extsec /u mydom%db2users /a mydom%db2admns
```

拡張セキュリティを以前の設定にリセットするには、以下のようになります (上記の **/reset** オプションに関するセクションを参照)。

```
db2extsec /reset
```

拡張セキュリティを使用可能にしますが、さらに *c:%mylist.lst* にリストされているファイル/ディレクトリーのセキュリティ・グループをローカル・グループ *db2admns* および *db2users* からドメイン・グループ *mydom%db2admns* および *mydom%db2users* に変更するには、以下のようになります。

```
db2extsec /users mydom%db2users /admins mydom%db2admns /oldadmins db2admns
/oldusers db2users /file c:%mylist.lst
```

注: 入力ファイルの形式は、次のとおりです。

```
* This is a comment
D:%MYBACKUPDIR
D:%MYEXPORTDIR
D:%MYMISCFILE%myfile.dat
```

## db2extsec - DB2 オブジェクトの権限の設定

```
* This is another comment  
E:¥MYOTHERBACKUPDIR  
E:¥MYOTHEREXPORTDIR
```

```
* These are more comments
```

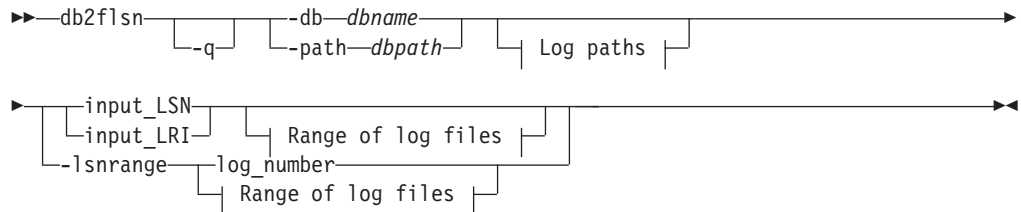
## db2flsn - ログ・シーケンス番号の検出

**db2flsn** コマンドは、指定されたログ・シーケンス番号 (LSN) で識別されるログ・レコードが含まれる、ログ・ストリーム内のファイルの名前を返します。

### 許可

なし

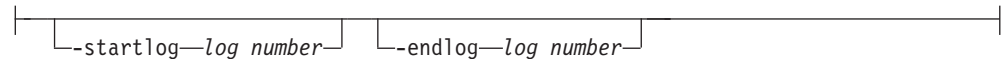
### コマンド構文



### Log paths:



### Range of log files:



### コマンド・パラメーター

**-q** ログ・ファイル名だけが印刷されます。エラー・メッセージも警告メッセージも出力されません。状況は戻りコードを介してのみ判別できます。有効なエラー・コードは以下のとおりです。

- -100 無効な入力
- -101 LFH ファイルをオープンできない
- -102 LFH ファイルの読み取りに失敗した
- -103 無効な LFH
- -106 無効なデータベース
- -108 LSN または LRI は、**db2flsn** で開くことのできる最も古いログ・ファイルよりも前のものです。
- -109 LSN または LRI は、**-startlog** パラメーターで指定されたログよりも前のものです。
- -110 ログ・ファイルを開けなかったため、LSN または LRI が見つかりませんでした。
- -120 エラーのため、LSN または LRI が見つかりませんでした。 **-lsnrange** オプションを使用してみてください。

他の有効な戻りコードは以下のとおりです。

## db2flsn - ログ・シーケンス番号の検出

- 0 正常な実行
- 100 警告: LSN または LRI は、さらに後のログ・ファイルにある可能性があります。 **db2flsn** で開くことのできる最後のファイルを返します。
- 101 警告: LSN または LRI は、さらに後のログ・ファイルにある可能性があります。 **-endlog** パラメーターで指定されたファイルを返します。正しいファイルを見つけるためには、 **-endlog** パラメーターで使用する値を大きくしてください。

### **-db** *dbname*

調査するデータベースの名前を指定します。

### **-path** *dbpath*

LFH ファイル `SQLLOGCTL.LFH.1` とそのミラー・コピー `SQLLOGCTL.LFH.2` が格納されるディレクトリーの絶対パスを指定します。

### *input\_LSN*

先行ゼロ付きの 8 バイト 16 進値を表す 16 文字ストリングを指定します。

### *input\_LRI*

先行ゼロ付きの 17 バイトの 16 進値を表す 34 文字ストリングを指定します。

### **-lsnrange**

ログ番号またはログ・ファイル番号の範囲を指定します。指定されたログ番号ごとに、LSN の値の範囲が戻されます。このオプションでは、ログ・ファイルは存在していて読み取り可能でなければなりません。

### *log\_number*

ログ・ファイルの番号を表す数値を指定します。例えば 52 は、ログ・ファイル `S0000052.LOG` の *log\_number* です。

### **-logpath** *primary\_logpath*

1 次ログ・パスを指定します。指定しない場合には、データベースの 1 次ログ・パスが使用されます。

### **-logpath2** *secondary\_logpath2*

2 次、つまり一時的なログ・パスを指定します。アーカイブにあるログ・ファイルを使用する場合は、リトリーブ用に一時ディレクトリーを使用します。その後、その一時パスを **-logpath2** パラメーターに指定すれば、指定されたログを **db2flsn** で読み取ることができます。 **logpath2** を指定しない場合、データベースのミラー・ログ・パスが使用されます (使用可能な場合)。

### **-startlog** *log\_number*

*log\_number* が指定された場合、**db2flsn** は、*log\_number* より前のログ・ファイルを検索しません。

### **-endlog** *log\_number*

*log\_number* が指定された場合、**db2flsn** は、*log\_number* より後のログ・ファイルを検索しません。

## 例

LSN の検索:

```
db2flsn 000000BF0030
Given LSN is contained in log file S0000002.LOG
```



```
db2flsn -q 000000BF0030
S0000002.LOG
```

```
db2flsn -db flsntest 0000000000FA0000
Given LSN is contained in log file S0000002.LOG
```

```
db2flsn -q -db flsntest 0000000000FA0000
S0000002.LOG
```

```
db2flsn -path /db2/NODE0000/SQL00001 0000000000FA4368
Given LSN is contained in log file S0000002.LOG
```

#### Using -lsnrange option:

##### db2flsn -lsnrange -startlog 20 -endlog 27

```
S0000020.LOG: has LSN range 0000000000023B3D to 0000000000023E0A
S0000021.LOG: has LSN range 0000000000023E0B to 00000000000240D8
S0000022.LOG: is not empty, but has no log records starting within its boundaries.
S0000023.LOG: has LSN range 00000000000240D9 to 00000000000243B9
S0000024.LOG: is unused/blank.
S0000025.LOG: is unused/blank.
S0000026.LOG: could not be opened or was invalid. Check file permissions.
S0000027.LOG: could not be opened or was invalid. Check file permissions.
```

#### Warning messages when an LSN could not be found, but it cannot be in an earlier log file:

##### db2flsn 00000000000243D7

```
Input LSN 00000000000243D7 is within or after log file S0000023.LOG:
```

```
S0000023.LOG: Log file starts with LSN 00000000000243BA.
S0000024.LOG: Log file could not be opened or was invalid/unused.
```

```
Log paths used:
(./LOGSTREAM0000) and
().
```

If this is not the end of the log stream, retrieve more log files into a temporary directory (to avoid overwriting active log files) and re-run db2flsn tool with -logpath2 option to use the log files from that temporary directory as well. Check access permissions on log files to make sure db2flsn can open necessary log files.

To see the first LSN of a log file, use the -lsnrange option:  
db2flsn -lsnrange <log file number> -logpath2 <temp log dir>

To see the first LSN of a set of log files, use:  
db2flsn -lsnrange -logpath2 <temp log dir> ¥  
-startlog <first log> -endlog <end log>

##### db2flsn -q 00000000000243D7

```
S0000023.LOG
```

=> Return code is: 100 (Warning, the LSN could be in a later log file, returning the last file that db2flsn could open).

##### db2flsn 00000000000243D7 -endlog 23

```
Input LSN 00000000000243D7 is after log file S0000023.LOG.
This log file has LSN range 00000000000243BA - 00000000000243D6.
To find the exact log file, try a larger value for the -endlog parameter.
```

```
Log paths used:
(./LOGSTREAM0000) and
().
```

To see the first LSN of a log file, use the -lsnrange option:  
db2flsn -lsnrange <log file number> -logpath2 <temp log dir>

## db2flsn - ログ・シーケンス番号の検出

To see the first LSN of a set of log files, use:  
db2flsn -lsnrange -logpath2 <temp log dir> ¥  
-startlog <first log> -endlog <end log>

```
db2flsn -q 0000000000243D7 -endlog 23
S0000023.LOG
=> Return code is: 101 (Warning, LSN could be in a later log file, returning
the file specified by -endlog parameter. To find the exact file, use a larger
value for the -endlog parameter.)
```

```
db2flsn -lsnrange 23
S0000023.LOG: has LSN range 0000000000243BA to 0000000000243D6
=> This shows that LSN 0000000000243D7 has not been produced yet on this log
stream. However, when it is generated, this LSN is guaranteed that it cannot be
in a log file before S0000023.LOG.
```

**Error messages, when an LSN could not be found. In this example, S0000000.LOG has already been archived and not found in history file either:**

```
db2flsn 000000000000001
Input LSN 0000000000000001 could not be found. Input LSN is before the earliest
LSN found 000000000021E6B:
```

```
S0000001.LOG: Log file starts with LSN 000000000021E6B.
S0000000.LOG: Log file could not be opened or was invalid/unused.
```

Log paths used:  
(./LOGSTREAM0000) and  
().

Retrieve log files before S0000001.LOG into a temporary directory (to avoid overwriting active log files) and re-run db2flsn tool with -logpath2 option to include the log files from that temporary directory:

```
db2flsn <LSN> -db <db> -logpath2 <temp log dir> -startlog <first log>
-endlog <last log>
```

To see the first LSN of a log file, use the -lsnrange option:  
db2flsn -lsnrange <log file number> -logpath2 <temp log dir>

To see the first LSN of a set of log files, use:  
db2flsn -lsnrange -logpath2 <temp log dir> ¥  
-startlog <first log> -endlog <eng log>

```
db2flsn -q 000000000000001
```

**=> with "-q" option, no output is produced as the log file could not be found. Error code returned is -108 (The LSN is before the earliest log file db2flsn could open).**

```
db2flsn 000000000000001 -startlog 1
Input LSN(0000000000000001) is before log file S0000001.LOG. This log file starts
with LSN 000000000021E6B.
```

Log paths used:  
(./LOGSTREAM0000) and  
().

To see the first LSN of a log file, use the -lsnrange option:  
db2flsn -lsnrange <log file number> -logpath2 <temp log dir>

To see the first LSN of a set of log files, use:  
db2flsn -lsnrange -logpath2 <temp log dir> ¥  
-startlog <first log> -endlog <end log>

```
db2flsn -q 000000000000001 -startlog 1
```

=> with "-q" option, no output is produced as the log file could not be found. Error code returned is -109 (The LSN is before the log specified by -startlog parameter.)

### 使用上の注意

- **-db** および **-path** が指定されていない場合には、**db2flsn** は、LFH ファイル (SQLOGCTL.LFH.1 および SQLOGCTL.LFH.2) が配置されているデータベース・パスとして現行ディレクトリーを取ります。**db2flsn** は、GLFH ファイル (SQLOGCTL.GLFH.1 および SQLOGCTL.GLFH.2) へのアクセスも必要とします。これらのファイルは、**-path** パラメーターで指定されたパスから、あるいは指定データベースのデフォルト・ロケーションから開きます。
- **db2flsn** ツールは、**-path** パラメーターによって提供されたディレクトリーから、あるいは所定のデータベースのヒストリー・ファイルが配置されているデフォルト・パスから、ヒストリー・ファイルを開こうと試みます。ヒストリー・ファイルには、ログ・アーカイブを使用するデータベース用のログ・ファイルの、開始 LSN が入っています。そのようなケースでは、**db2flsn** は、既にアーカイブされたログ・ファイルを開けないときに、ヒストリー・ファイルを調べます。**logarchmeth1** または **logarchmeth2** 構成パラメーターが OFF でも LOGRETAIN でもない値に設定されて、データベースが構成されている場合に、データベースはログ・アーカイブを使用します。
- このツールは、リカバリー可能データベースとリカバリー不能 (循環ロギング) データベースの両方で機能します。データベースがリカバリー可能なのは、そのデータベースの構成パラメーター **logarchmeth1** または **logarchmeth2** が OFF 以外の値に設定されている場合です。

## db2fm - DB2 障害モニター

DB2 障害モニター・デーモンを制御します。**db2fm** を使用すると、障害モニターを構成できます。

このコマンドは UNIX および Linux オペレーティング・システムで使用できます。

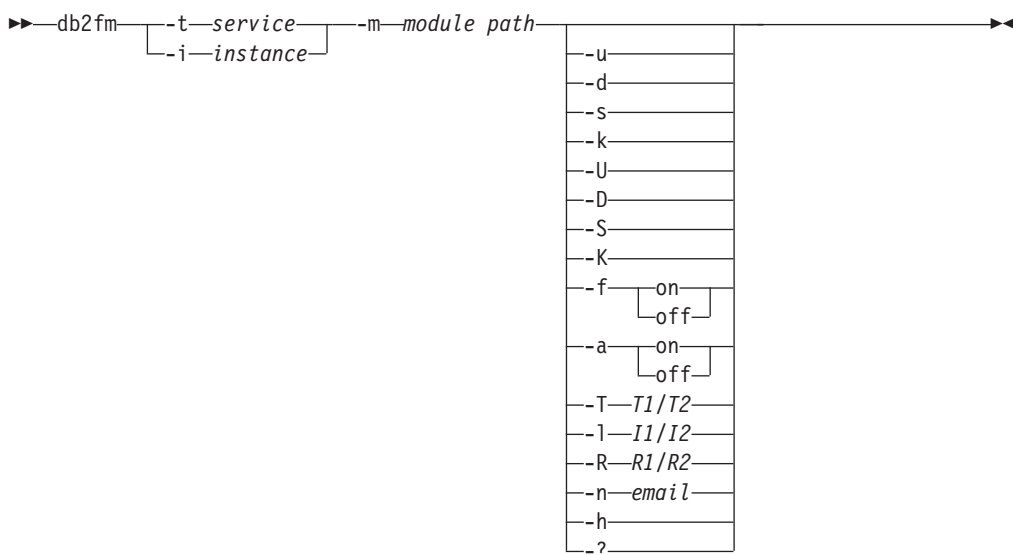
### 許可

コマンド実行対象のインスタンスに対する許可。

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

**-m** *module-path*

モニター対象製品の障害モニター共有ライブラリーの絶対パスを定義します。デフォルトは \$INSTANCEHOME/sql1lib/lib/libdb2gcf です。

**-t** *service*

固有のテキスト記述子をサービスに対して指定します。

**-i** *instance*

サービスのインスタンスを定義します。

**-u** サービスを開始します。

**-U**

障害モニター・デーモンを開始します。

**-d** インスタンスを停止します。

- D**  
障害モニター・デーモンを停止します。
- k** サービスを強制終了します。
- K**  
障害モニター・デーモンを強制終了します。
- s** サービスの状況に戻します。
- S** 障害モニター・デーモンの状況に戻します。サービスまたは障害モニターの状況は、次のいずれかになります。
  - 適切にインストールされていません (Not properly installed)、
  - 適切にインストールされていますが、活動状態にありません (INSTALLED PROPERLY but NOT ALIVE)、
  - 活動状態ですが、使用できません (保守) (ALIVE but NOT AVAILABLE (maintenance))、
  - 使用できます (AVAILABLE)、または
  - 不明 (UNKNOWN)
- f on | off**  
障害モニターを ON または OFF にします。このオプションが off に設定される場合、障害モニター・デーモンは開始されないか、デーモンが実行中の場合は終了されます。
- a on | off**  
障害モニターをアクティブ化または非アクティブ化します。このオプションが OFF に設定されると、障害モニターはアクティブなモニターを行いません。これは、サービスが停止する場合、再び始動しないことを意味します。
- T T1/T2**  
開始および停止タイムアウトを上書きします。  
以下に例を示します。
  - **-T 15/10** は、2 つのタイムアウトを更新します。
  - **-T 15** は、開始タイムアウトを 15 秒に更新します。
  - **-T /10** は、停止タイムアウトを 10 秒に更新します。
- I I1/I2**  
状況インターバル、タイムアウトを設定します。
- R R1/R2**  
中止する前に再試行される状況メソッドおよびアクションの回数を設定します。
- n e-mail**  
イベント通知用の E メール・アドレスを設定します。
- h | -?**  
コマンドの使用についてのヘルプを表示します。

### 使用上の注意

- 障害モニター・デーモンによって E メール通知が送信されるようセットアップする場合は、ローカル・ホスト上で SMTP を実行していなければなりません。DB2 障害モニターには、SMTP サーバーを指定するための設定がありません。障害モ

## db2fm - DB2 障害モニター

モニターにより E メール通知が送信されるようにするには、ホスト名を SMTP サーバーにバインドする必要があります。または次善策として、ホスト名を /etc/hosts ファイルでローカル・ホストの別名として指定します。

## db2fmcu - DB2 障害モニター・コントローラー

DB2 障害モニターは、異常終了の後に自動的にインスタンスを開始する DB2 データベースの機能です。それはまた、マシンのリブート時にインスタンスを自動的に再開します。Linux および UNIX オペレーティング・システム上で、DB2 障害モニター・コントローラー・コマンドを使用して DB2 障害モニターを構成できます。このコマンドは、システムの `inittab` ファイルにアクセスするので、`root` として実行する必要があります。

### 許可

root ユーザー権限

### 必要な接続

なし

### コマンド構文

```

▶▶ db2fmcu --u--p db2fmcu_path --d --f inittab

```

### コマンド・パラメーター

#### **-u -p** *db2fmcu\_path*

このオプションは、システムの始動時に障害モニター・コントローラー (FMC) を組み込むように `inittab` ファイルを再構成します。*db2fmcu\_path* は、FMC デモン (`db2fmcu`) オブジェクトへの完全なパスで、例えば `/opt/IBM/db2/bin/db2fmcu` となります。

Red Hat Enterprise Linux (RHEL) 6 では、`/etc/inittab` ファイルが非推奨になっています。このオプションを指定すると、`/etc/init` ディレクトリー下で `db2fmcu.conf` ファイルが作成または置換されます。

Solaris 10 では、`/etc/inittab` ファイルが非推奨になっています。このオプションを指定すると、`db2fm` SMF サービスが作成され、使用可能になります。

#### **-d** このオプションは、FMC がシステムの始動時に実行しないように、`inittab` ファイルの構成を変更します。

RHEL 6 では、`/etc/inittab` ファイルが非推奨になっています。このオプションを指定すると、`db2fmcu.conf` ファイルが削除されます。

Solaris 10 では、`/etc/inittab` ファイルが非推奨になっています。このオプションを指定すると、`db2fm` SMF サービスが使用不可になります。

#### **-f** *inittab*

このオプションは、`inittab` ファイルへのパスを指定します。

### 例

`inittab` ファイルを再構成してシステムの始動時に障害モニター・コントローラーが開始するようにするには、次のコマンドを実行します。

```
db2fmcu -u -p /opt/IBM/db2/bin/db2fmcu
```

システムの始動時に障害モニター・コントローラーが起動しないようにするには、次のコマンドを実行します。

```
db2fmcu -d
```

### 使用上の注意

- /etc/inittab を手動で変更した場合、SIGHUP をプロセス 1 に送信して、/etc/inittab を即時に再スキャンするように依頼する必要があります。そのようにしないと、次の再スキャンが生じるまでに時間がかかることがあります。/etc/inittab を **db2fmcu** によって更新した場合には、シグナルの送信は **db2fmcu** コマンドによって既に行われているので必要ありません。
- Solaris オペレーティング・システムでは、/etc/inittab ファイルにリストされているプロセスは、デフォルトのシステム・プロジェクト設定を継承しています。必ず、プロジェクトのカーネル・パラメーターを十分高く設定し DB2 プロセスを収容できるようにするか、または、適切なカーネル設定がなされた他のプロジェクトの下で **db2fmcu** プロセスを開始するようにしてください。別のプロジェクトの下で **db2fmcu** プロセスを開始するには、/etc/inittab ファイルの **fmc** 項目を編集して **newtask** を使用するようにしてください。例えば、/usr/bin/newtask -p を **db2fmcu** パスの前に追加します。カーネル・パラメーターの設定に関して詳しくは、**db2osconf** コマンドを参照してください。



## db2fodc - DB2 First Occurrence Data Collection

この **db2fodc** ユーティリティは、問題判別状態に役立つ、DB2 インスタンスに関する症状に基づくデータをキャプチャーします。潜在的なハング、パフォーマンス上の重大な問題、およびさまざまなタイプのエラーに関する情報を収集することが目的です。

### 目的

**db2fodc** コマンドは、ハングやパフォーマンス上の重大な問題など First Occurrence Data Collection (FODC) を自動的にトリガーできない問題に関する手動による FODC に使用できます。それは索引エラーに関するデータを収集するためにも使用できます。 **db2fodc** ツールはデータをキャプチャーして FODC パッケージに組み込み、デフォルトの診断パス、または **-fodcpath** パラメーターを使用して指定した FODC ディレクトリー・パスに作成された FODC パッケージ・ディレクトリー内にそのパッケージを入れます。

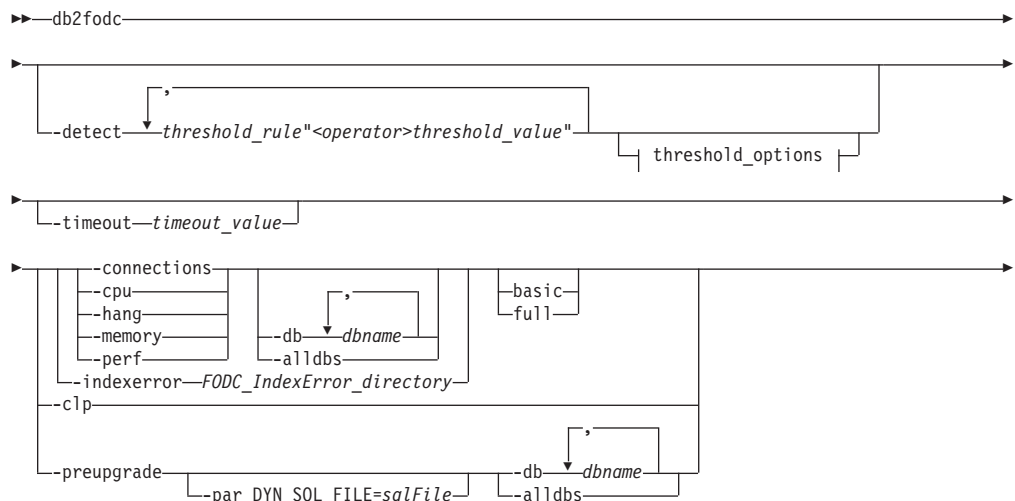
**db2fodc** ツールは、追加の手動収集タイプをサポートしていますし、ユーザー定義のしきい値条件を超える場合における自動診断データ収集のトリガーもサポートしています。

### 許可

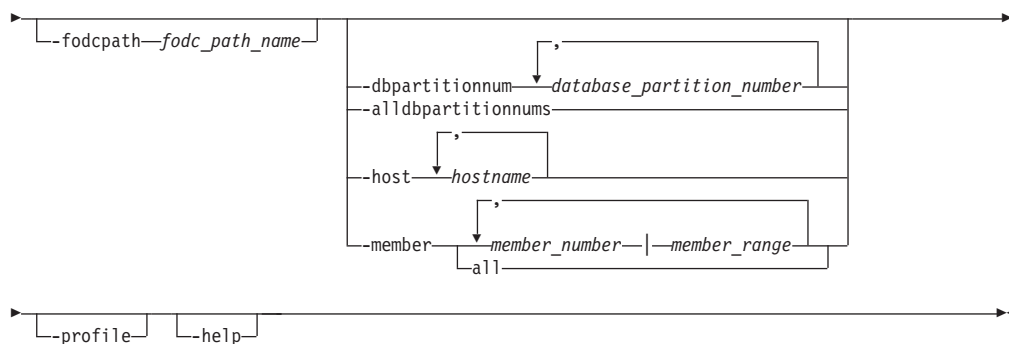
以下の権限のいずれか。

- Linux および UNIX システムの場合、*sysadm* 権限レベル。さらに、インスタンスの所有者でなければなりません。
- Windows オペレーティング・システムの場合、*sysadm* 権限レベル。

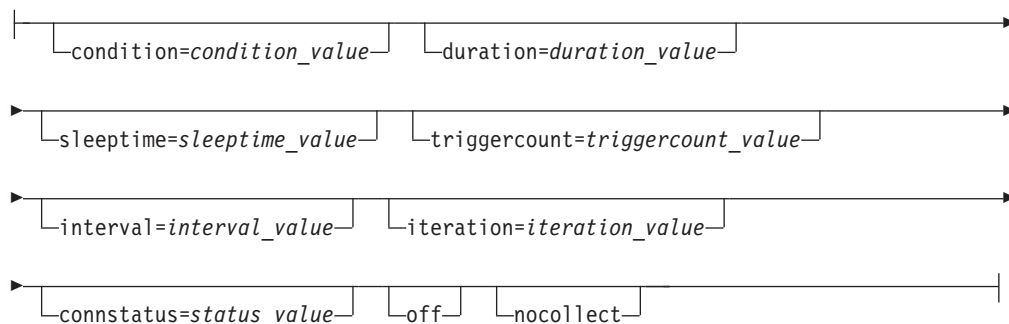
### コマンド構文



## db2fodc - DB2 First Occurrence Data Collection



### threshold\_options:



## コマンド・パラメーター

### -detect

指定した 1 つ以上のしきい値規則に基づいてトリガー条件を検出します。有効なしきい値規則を少なくとも 1 つ指定する必要があります。検出すると、db2diag ログ・ファイルに書き込まれます。triggercount オプションで指定した回数、トリガー条件が検出されると、診断データ収集が実行されます。

#### threshold\_rule

しきい値によって検出する条件を示す、サポートされるしきい値規則名。使用可能な規則は、プロセッサ使用量、メモリー使用量、データベース接続をカバーします。しきい値規則は複数指定できます。詳しくは、サポートされるしきい値検出の規則を参照してください。

#### <operator>

サポートされている演算子。>= (以上) または <= (以下) のいずれかです。

#### threshold\_value

指定するしきい値規則のための数値。指定できるのは、負ではない整数だけです。

#### condition=condition\_value

複数のしきい値規則が指定された場合に、しきい値規則を AND または OR で結合するための論理演算子。指定できるのは and または or です。デフォルトはしきい値規則を AND で結び付けます。

#### duration=duration\_value

しきい値検出と診断データ収集を有効にする期間を時間単位で指定します。

**sleeptime=sleeptime\_value**

次のしきい値検出および診断データ収集の反復を開始するまでに待機する時間を秒単位で指定します。デフォルトは 1 秒です。

**triggercount=triggercount\_value**

診断データ収集をトリガーする前にしきい値条件を検出する必要がある連続回数を指定します。デフォルトは 5 回です。トリガー条件が検出されるたびに、指定の間隔スリープ状態になってから検出が再開されます。

**interval=interval\_value**

現在の反復において、しきい値条件の検出を再開するまでに待機する時間を秒単位で指定します。デフォルトは 1 秒です。

**iteration=iteration\_value**

しきい値検出と診断データ収集を実行する最大反復回数を指定します。デフォルトでは反復は 1 回です。

**connstatus=status\_value**

接続の接続状況を指定します。デフォルトでは、データベースに接続されているアプリケーションのすべての接続状況を対象とします。有効な接続状況キーワードは、

CommitActive、Compiling、ConnectCompleted、ConnectPending、CreatingDatabase、Decoupled、DisconnectPending、FederatedRequestPending、HeuristicallyCommitted、HeuristicallyAborted、Lockwait、PendingRemoteRequest、PerformingLoad、PerformingBackup、PerformingUnload、QuiescingTable-space、RequestInterrupted、Recompiling、RestartingDatabase、RestoringDatabase、RollbackActive、RollbackToSavepoint、TransactionEnded、TransactionPrepared、UOW-Executing、UOW-Waiting、UOWQueued、Unknown、WaitAutonomous、WaitToDisableTablespace です。

**off**

すべてのしきい値検出を停止し、現在アクティブなしきい値規則をオフにします。off に指定して、他のオプションも指定している場合には、他のオプションは無視されます。しきい値検出をオフにしてから完全に有効になるまでには、最長で 60 秒必要です。

**nocollect**

しきい値条件が満たされる場合に検出だけを実行し、診断データ収集は実行しないように指定します。しきい値検出は db2diag ログ・ファイルに記録されます。

**-clp**

環境と構成に関する情報を収集し、インスタンス作成に関連した問題のトラブルシューティングに使用します。db2fodc -clp コマンドは -member パラメーターはサポートしていませんが、-host パラメーターはサポートしています。収集された情報は、FODC\_Clp\_timestamp\_member という新規ディレクトリーに格納されます。

## db2fodc - DB2 First Occurrence Data Collection

注: **-c1p** パラメーターがサポートされているのは、Linux および UNIX オペレーティング・システムのみです。Windows オペレーティング・システムでこのコマンドを発行しても、データは収集されません。

### **-connections**

接続関連の診断データを収集します。このデータは、実行状態やコンパイル状態のアプリケーション数に急激なスパイクが生じたり、新しいデータベース接続が拒否されたりするなどの問題を診断するのに使用します。収集された情報は、*FODC\_Connections\_timestamp\_member* という新規ディレクトリーに格納されます。

### **-cpu**

プロセッサ関連のパフォーマンス・データと診断データを収集します。これらのデータは、プロセッサ使用率が高くなったり、実行中のプロセス数が多くなったり、プロセッサの待機時間が長くなったりするなどの問題を診断するのに使用します。収集された情報は、*FODC\_Cpu\_timestamp\_member* という新規ディレクトリーに格納されます。

### **-hang**

潜在的なハング状態または重大なパフォーマンス上の問題に関連する FODC データを収集します。このオプションを実行すると、FODC パッケージが作成されます。このパラメーターが使用されると、インスタンスは使用不可と見なされ、再始動が必要になります。データ収集ができるだけ早く実行されて、できるだけ多くの情報が収集されます。

### **-indexerror** *FODC\_IndexError\_directory*

索引エラーに関連したデータを収集します。 *FODC\_IndexError\_directory* は必須で、*db2cos\_indexerror\_short(.bat)* スクリプトまたは *db2cos\_indexerror\_long(.bat)* スクリプトを含める必要があります。 **db2dart** レポートが同じ *FODC\_IndexError\_directory* 内に存在しないようにしてください。 *db2cos\_indexerror\_long(.bat)* スクリプトに **db2dart /t** コマンドが含まれていると、完全収集モードでは、**db2dart /t** コマンドの実行対象となるデータベースがオフラインであることが必要です。

### **-memory**

メモリー関連の診断データを収集します。このデータは、空きメモリーを使用できない、スワップ・スペースの使用率が高い、過剰なページングが生じている、メモリー・リークが疑われるといった問題を診断するのに使用します。収集された情報は、*FODC\_Memory\_timestamp\_member* という新規ディレクトリーに格納されます。

### **-perf**

パフォーマンスの問題に関連したデータを収集します。このオプションは、インスタンスがまだ使用可能であり、再始動が不必要な場合に使用してください。このオプションがシステムに与える影響は、**-hang** パラメーターよりも小さくなります。完全収集モードがユーザーと対話せずに実行され、**db2trc** コマンドからの出力や追加のスナップショットが含まれます。このオプションを実行すると、FODC パッケージが作成されます。

### **-preupgrade**

インスタンスのアップグレードや、新しいフィックスパックへの更新などの重要なアップグレードや更新を行う前に、パフォーマンス関連の情報を収集します。このようにすると、アップグレードや更新後に生じる場合がある問題のトラブル

シューティングに役立ちます。このパラメーターでは、データベースを指定する必要があります。データ収集が完了すると、結果は `FODC_Preupgrade_timestamp_member` というディレクトリーに配置されます。その後には生じる問題をトラブルシューティングするための十分なパフォーマンス・データを得るには、このコマンドをピーク時とアイドル使用時の両方で数回にわたって発行しなければならない場合があります。

**注:** このコマンドによるパフォーマンス・データ収集が完了するまでには、長い時間を要する可能性があります。

### **-profile**

`db2fodc.profile` ファイルが `~/sqllib/cfg` ディレクトリー内にある場合に、潜在的なハング状態に関連する部分的な FODC データを収集します。また、`db2fodc.profile` ファイルを構成してデータ収集をカスタマイズすることもできます。このパラメーターは、短期間のうちに特定の FODC データを収集する必要がある場合に特に有用です。

**注:** `-profile` パラメーターは、Windows NT プラットフォームではサポートされません。

### **オプション:**

#### **basic**

この基本収集モードはユーザーと対話せずに実行されます。

#### **full**

この完全収集モードは、ユーザーと対話せずに実行されます。このオプションを指定して実行するためには、基本的な収集モードよりもかなり多くのリソースおよび時間を必要とします。

#### **-alldbpartitionnums**

インスタンス内のすべてのアクティブなデータベース・パーティション・サーバーでこのコマンドを実行することを指定します。 `db2fodc` は、`db2fodc` が実行されているのと同じ物理マシン上のデータベース・パーティション・サーバーからの情報を報告します。

#### **-alldbs**

すべてのアクティブなデータベースに関連する FODC データを収集します。デフォルトで、このオプションはアクティブです。

#### **-db dbname**

指定のデータベース (複数可) に関連する FODC データを収集します。例:

```
db2fodc -hang -db sample,dbsample
```

#### **-dbpartitionnum database\_partition\_number**

すべての指定のデータベース・パーティション番号に関連する FODC データを収集します。

#### **-fodcpath fodc\_path\_name**

FODC データ・パッケージが作成されるディレクトリーの絶対パスを指定します。指定するパスは、マシン上のメンバーが書き込み可能な場所でなければなりません。さらに、そのメンバーまたはパーティション上で実行される `fmp` プロセスで書き込み可能な場所でなければなりません。

### **-help**

使用法とヘルプ情報を表示します。このオプションを指定すると、他のすべてのオプションは無視され、ヘルプ情報だけが表示されます。

### **-host hostname**

コマンド発行先ホスト (複数も可) を指定します。このコマンドは、ホスト上に存在するすべてのメンバーに対して実行されます。ホスト名を指定しないと、このコマンドはローカル・ホスト上のデフォルト・メンバーに対して発行されます。複数のホストを指定する場合、すべてのホスト名が有効でないと、コマンドを完了できません。このオプションを、**-member** オプションと一緒に指定することはできません。

### **-member member\_number | member\_range**

コマンド発行先メンバー (複数も可) を指定します。このオプションを指定しない場合、コマンドは現行メンバーに対して発行されます。複数のメンバーを、*member\_number* のコンマ区切りリスト (*member1*, *member2*) として、または *member\_range* (*member\_range* はメンバーの範囲 (*member1*-*member3*)) を使用して指定できます。最初の 2 つの方式を組み合わせることもできます。このオプションを、**-host** オプションと一緒に指定することはできません。

### **all**

これを指定すると、コマンドは、*db2nodes.cfg* で定義されたすべてのメンバーに対して発行されます。

### **-par DYN\_SQL\_FILE=sqlFile**

ユーザーによって指定された SQL ファイルに関連した FODC データを収集します。併用できるのは、**-preupgrade** パラメーターのみです。

### **-timeout timeout\_value**

**db2fodc** コマンドで呼び出すコールアウト・スクリプトのタイムアウト期間を指定します。コールアウト・スクリプトによる診断データの収集が完了する前にタイムアウトに達すると、そのコールアウト・スクリプトのプロセスは強制終了されます。デフォルトのタイムアウトはありません。タイムアウトは、*nh ym xs* と指定します。*n* は時間を、*y* は分を、そして *x* は秒を表します。h、m、および n のいずれの接尾部も指定しない場合、タイムアウトの単位は秒と見なされます。有効なタイムアウト期間は、**-timeout 2h 30m 45s** や **-timeout 600** などです。

注: **-timeout** オプションは、Windows オペレーティング・システムではサポートされていません。

## サポートされるしきい値検出の規則

以下のプロセッサ使用率およびメモリーのしきい値規則がサポートされています。

表 49. 種々のオペレーティング・システム用のプロセッサ使用量とメモリの規則

オペレーティング・システム	AIX <sup>1</sup>	Linux <sup>2</sup>	HP-UX <sup>3</sup>	Solaris <sup>4</sup>	Windows
実行キュー (rqueue)	r	r	r	r	r
ブロック・キュー (bqueue)	b	b	b	b	該当なし
アクティブ・メモリー (avm)	avm	active	avm	該当なし	used
空きメモリー (free)	fre	free	free	free	free
ページイン (pi)	pi	該当なし	pi	pi	pi
ページアウト (po)	po	該当なし	po	po	po
スワップイン (si)	該当なし	si	si	si	該当なし
スワップアウト (so)	該当なし	so	so	so	該当なし
スキャンされたページ (sr)	sr	該当なし	sr	sr	該当なし
ユーザー CPU (us)	us	us	us	us	usr
システム CPU (sy)	sy	sy	sy	sy	sys
ユーザーおよびシステム CPU (us_sy)	us+sy	us+sy	us+sy	us+sy	us+sy
アイドル CPU (id)	id	id	id	id	idl
コンテキスト・スイッチ (cs)	cs	cs	cs	cs	cs/s
使用されたコマンド	<b>vmstat</b>	<b>vmstst -a</b>	<b>vmstat -S</b>	<b>vmstat -S</b>	<b>db2pd -vmstat</b>



表 49. 種々のオペレーティング・システム用のプロセッサ使用量とメモリーの規則 (続き)

オペレーティング・システム	AIX <sup>1</sup>	Linux <sup>2</sup>	HP-UX <sup>3</sup>	Solaris <sup>4</sup>	Windows
注:					
1. AIX オペレーティング・システムの場合					
<ul style="list-style-type: none"> <li>• メモリー: 仮想メモリーと実メモリーの使用量に関する情報。仮想ページは、アクセスされた場合にアクティブであると見なされます。実メモリーの大部分は、ファイル・システム・データのキャッシュとして使用されます。フリー・リストのサイズが小さいままであるのは珍しいことではありません。1 ページは 4096 バイトです。</li> <li>• pi: ページング・スペースからページインされたページ数。</li> <li>• po: ページング・スペースにページアウトしたページ数。</li> <li>• sr: ページ置換アルゴリズムによってスキャンされたページ数。</li> <li>• avm: アクティブな仮想ページ数。</li> <li>• fre: フリー・リストのサイズ。</li> </ul>					
2. Linux オペレーティング・システムの場合					
<ul style="list-style-type: none"> <li>• メモリー: すべての linux ブロックは現在 1024 バイトです。</li> <li>• si: ディスクからスワップインされたメモリーの秒当たりの量。</li> <li>• so: ディスクへスワップされたメモリーの秒当たりの量。</li> <li>• inact: 非アクティブなメモリーの量。</li> <li>• active: アクティブなメモリーの量。</li> </ul>					
3. HP-UX オペレーティング・システムの場合					
<ul style="list-style-type: none"> <li>• メモリー: 仮想メモリーと実メモリーの使用量に関する情報。仮想ページは、実行中のプロセス、または直近の 20 秒間実行されていたプロセスに属している場合にアクティブと見なされます。</li> <li>• si: スワップインしたプロセス数。</li> <li>• so: スワップアウトしたプロセス数。</li> <li>• pi: ページインしたページ数。</li> <li>• po: ページアウトしたページ数。</li> <li>• sr: クロック・アルゴリズムによってスキャンされた秒当たりのページ数。</li> <li>• avm: アクティブな仮想ページ数。</li> <li>• free: フリー・リストのサイズ。</li> </ul>					
4. Solaris オペレーティング・システムの場合					
<ul style="list-style-type: none"> <li>• si: Kbps 単位で測定されたスワップイン数をレポートします。</li> <li>• so: Kbps 単位で測定されたスワップアウト数をレポートします。</li> <li>• pi: ページインされた KB 数。</li> <li>• po: ページアウトされた KB 数。</li> <li>• sr: クロック・アルゴリズムによってスキャンされたページ数。</li> <li>• swap: KB 単位で測定された、現在使用可能なスワップ・スペース量。</li> <li>• free: フリー・リストのサイズ。</li> </ul>					



表 50. 種々のオペレーティング・システム用のその他のメモリー規則

オペレーティング・システム	使用済みスワップ・スペース (swaped)	使用されたコマンド
AIX	Percent Used 列の下にあるパーセンテージ値	<b>lsps -s</b>
Linux	(合計スワップ・スペース/使用されたスワップ・スペース)*100%	<b>free</b>
HP-UX	PCT USED 列の下にある合計パーセント値	<b>swapinfo -tam</b>
Solaris	(使用されたスワップ・スペース/(使用可能なスワップ・スペース + 使用されたスワップ・スペース))*100%	<b>swap -s</b>
Windows	該当なし	該当なし

表 51. 種々のオペレーティング・システム用のデータベース有効範囲内の接続規則

オペレーティング・システム	接続	使用されたコマンド
AIX	<b>connstatus</b> オプションによって指定された状況での接続されたアプリケーションの数。	<b>db2pd -application</b>
Linux	<b>connstatus</b> オプションによって指定された状況での接続されたアプリケーションの数。	<b>db2pd -application</b>
HP-UX	<b>connstatus</b> オプションによって指定された状況での接続されたアプリケーションの数。	<b>db2pd -application</b>
Solaris	<b>connstatus</b> オプションによって指定された状況での接続されたアプリケーションの数。	<b>db2pd -application</b>
Windows	<b>connstatus</b> オプションによって指定された状況での接続されたアプリケーションの数。	<b>db2pd -application</b>

## 例

データベース・マネージャーを停止することなく、潜在的なハングの際のデータを収集するには、以下のようにします。

```
db2fodc -hang -all dbs
```

デフォルトの DB2FODC レジストリー変数とパラメーターが使用されます。FODC\_Hang\_ という接頭部を持つ新しいディレクトリーが、現行の診断パスの下に作成されます (既にこの名前のディレクトリーが存在する場合には、エラーが生成さ

## db2fodc - DB2 First Occurrence Data Collection

れます)。 db2cos\_hang スクリプトが実行されて、1 つ以上のファイルに FODC データが手動で収集され、ディレクトリーに入れられます。

特定のデータベースのデータを収集するには、次のようにします。

```
db2fodc -db SAMPLE -hang
```

データ収集が行われるのは、データベース SAMPLE に限定されます。FODC\_Hang\_ という接頭部を持つ新しいディレクトリーが、現行の診断パスの下に自動的に作成されます。 db2cos\_hang スクリプトが実行され、FODC パッケージに FODC データが手動で収集されて、ディレクトリーに保管されます。

完全収集スクリプトを使用して、パフォーマンス問題の際に特定のデータベースからデータを収集するには、次のようにします。

```
db2fodc -db SAMPLE -perf full
```

データ収集が行われるのは、データベース SAMPLE に限定されます。FODC\_Perf\_ という接頭部を持つ新しいディレクトリーが、現行の診断パスの下に作成されます。 db2cos\_perf スクリプトが実行されて、1 つ以上のファイルに FODC データが手動で収集され、ディレクトリーに入れられます。

データベース・マネージャーを停止しないでデフォルトの (基本) モードで、索引エラーに関するデータを収集するには、次のようにします。

```
db2fodc -indexerror FODC_IndexError_directory
```

db2cos\_indexerror\_short(.bat) スクリプトが実行されて、1 つ以上のファイルに FODC データが手動で収集されます。それらのファイルは、ディレクトリー *FODC\_IndexError\_directory* に入れられます。

パフォーマンス問題の際にメンバー 10、11、12、13、15 に関するデータを収集するには、次のようにします。

```
db2fodc -perf -member 10-13,15
```

ローカル・ホストでハング状況が生じた可能性がある場合にデータを収集するには、次のようにします。

```
db2fodc -hang
```

しきい値条件を検出したり、しきい値条件を複数回超過した場合に自動診断データ収集をトリガーしたりするには、以下の照会のようなしきい値規則を作成します。

```
db2fodc -memory basic -detect free<=10" connections">=1000" sleeptime="30" iteration="10" interval="10" triggercount="4" duration="5" -member 3
```

このしきい値規則の効果は次のようになります。メンバー 3 で検出が実行されて、しきい値規則 *free<=10* と *connections>=1000* によって指定された条件が満たされるかどうか検査されます。これらのしきい値規則は、フリー・リストのサイズが 10 以下でなければならないこと、および 1000 以上の接続が検出される必要があることを指定しています。トリガー・カウント用のこれらのしきい値条件が満たされると、FODC メモリー収集がメンバー 3 でトリガーされます。FODC 収集をトリガーするには、トリガー条件が 40 秒間 (*triggercount 4 x interval 10 秒 = 40 秒*) 存在しなければなりません。反復ごとに検出は 30 秒間スリープ状態になり、その検

出は合計 5 時間有効です。FODC メモリー収集がトリガーされると、FODC\_Memory\_ という接頭部を持つ新しいディレクトリーが現行の診断パスの下に作成されます。

しきい値条件の検出を停止し、現在アクティブなしきい値規則をすべてオフにするには、以下のコマンドを発行します。このコマンドを発行してからすべての検出が停止されるまで最大 60 秒かかる可能性があります。

```
db2fodc -detect off
```

## 使用上の注意

**db2fodc** コマンドを手動で呼び出す場合、このコマンドと一緒に **-fodcpath** パラメーター・オプションを指定することによって、FODC パッケージ・ディレクトリーが作成される場所を指示できます。自動 FODC の場合、または **-fodcpath** パラメーターを **db2fodc** コマンドで指定しないで、パーティションまたはメンバーのリストを指定した場合には、**db2fodc** コマンドは、指定のリストにある最初のパーティションまたはメンバーの **DB2FODC** レジストリー変数の **FODCPATH** パラメーター設定を使用します。その **FODCPATH** パラメーターの値が設定されていない場合、**db2fodc** は、インスタンス・レベル **FODCPATH** の設定値を使用します。**-fodcpath** パラメーターを指定せず、パーティションまたはメンバーのリストも指定しない場合には、**db2fodc** コマンドは、現行のパーティションまたはメンバーの **FODCPATH** パラメーター設定をまず使用しようと試行します。そのパラメーター設定がない場合には、インスタンス・レベルの設定が使用されます。デフォルトでは、FODC パスを指定しない場合には、First Occurrence Data Capture によって現在の診断ディレクトリー・パス (**diagpath** または **alt\_diagpath**) に診断情報が送信されます。

**db2fodc -hang** および **db2fodc -perf** は、複数の物理データベース・パーティションがあるマルチ・パーティション環境で実行できます。この環境では

```
db2fodc -perf <full | basic> -member all other_options
```

または

```
db2fodc -perf <full | basic> -alldbpartitionnums other_options
```

を使用して、ハングまたは重大なパフォーマンスの問題が起きている可能性がある場合に、すべての DB2 メンバーに対して 1 回の呼び出しで **db2fodc** を呼び出す必要があります。ユーザーとの対話を必要としない収集モードを設定するには、サブオプション **full** または **basic** を使用します。**-member all** オプションと **-member** オプションは、ローカル・メンバーまたはリモート・メンバーに対して作用します。**-member** も **-host** も指定しないと、デフォルトでは現行メンバーの情報だけが収集されます。

**db2fodc -indexerror** は、複数の物理または論理データベース・パーティションがあるマルチ・パーティション環境で実行できます。この環境では、特定の DB2 メンバーでの索引エラーに関する情報を収集するために、

```
db2_all "<<+member#< db2fodc -indexerror FODC_IndexError_directory <basic | full>"
```

を使用して、**db2fodc** を呼び出す必要があります。member# は、特定のメンバーの番号に置き換えてください。この番号は、ディレクトリー名 <FODC\_IndexError\_timestamp\_PID\_EDUID\_Member#> での最後の番号です。絶対パスのある **FODC\_IndexError\_directory** を指定する必要があり、サブオプションの **full**

## db2fodc - DB2 First Occurrence Data Collection

または **basic** を使用して収集モードを設定できます。絶対パスが必要となるのは、**db2fodc -indexerror** を **db2\_all** コマンドと共に使用するときだけです。

**db2fodc -indexerror** はコマンドの進行状況を出力して、メッセージがある場合にはそれを **db2diag** ログ・ファイルのログに記録します。

**db2fodc -indexerror** には、**db2cos\_indexerror\_short(.bat)** スクリプト、**db2cos\_indexerror\_long(.bat)** スクリプト、またはその両方を入力として含む **FODC\_IndexError\_directory** が必要です。

**db2cos\_indexerror\_short(.bat)** および **db2cos\_indexerror\_long(.bat)** スクリプトは、自動索引付けエラー FODC 処理の際に作成された **FODC\_IndexError\_directory** 内にあります。スクリプトには、複数の **db2dart** コマンドが含まれます。スクリプトの実行後に、生成された **db2dart** レポートは、スクリプトが存在する **FODC\_IndexError\_directory** と同じ場所に置かれます。手動の **db2fodc -indexerror** コマンドを発行しても、新しいディレクトリーは作成されません。**db2fodc -indexerror** は、新しい **db2dart** レポートを、自動索引付けエラー FODC 処理によって作成された同じ **FODC\_IndexError\_directory** 内に生成します。

**FODC\_IndexError\_directory** は、名前変更したり移動したりしないでください。スクリプト内の **db2dart** コマンドは、レポートを正しく生成するためにこのディレクトリー・パスを必要とします。

**db2fodc -indexerror** を手動で実行する必要がある場合、**FODC\_IndexError\_directory** に既存の **db2dart** レポートがあるかどうかを調べてください。レポートには、拡張子 **.rpt** および **.rpthex** があります。レポートが存在する場合、**db2fodc -indexerror** を手動で実行する前に、それらのレポートを名前変更するか、または **FODC\_IndexError\_directory** の下のサブディレクトリーに移動する必要があります。これにより、既存のレポートは **db2support** ツールによって収集されるときのために保存されて、**db2fodc -indexerror** は新しい **db2dart** レポートを作成できるようになります。

## db2fopt - 照会オブティマイザー・パラメーターの指定

**db2fopt** コマンドは、照会オブティマイザーが使用するパラメーターを指定します。このコマンドは、物理リソースが実動システムよりも少ないテスト・システムをセットアップする場合に使用できます。

例えば、実動システムが **sortheap=20000** という設定で稼働し、テスト・システムは **sortheap=5000** という設定でのみ動作できる場合は、テスト・システムで **db2fopt** を使用することによって **opt\_sortheap** オブティマイザー・パラメーターを 20000 に設定できます。これによりオブティマイザーは、**sortheap** データベース構成パラメーターが 5000 に設定されていても、アクセス・プランを評価する際にソート・ヒープ値として 20000 を使用するようになります。

### 有効範囲

このコマンドは、それが実行されたデータベース・パーティションに対してだけ影響を与えます。

**許可** **get** オプションを使用してパラメーターを照会する場合: なし

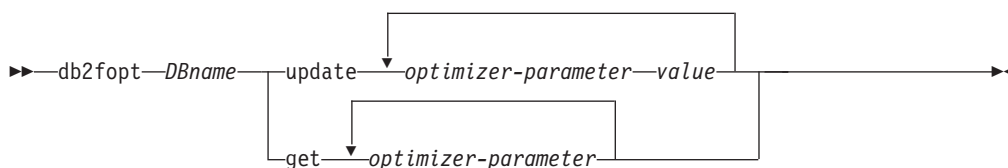
パラメーターを更新する場合 (**update**) は、以下のいずれかの権限が必要です。

- SYSADM
- SYSCTRL
- SYSMANT

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### *DBname*

データベースの別名。

#### **update** *optimizer-parameter value*

このコマンドは、オブティマイザー・パラメーターを更新する場合に使用します。

- **opt\_buffpage**
- **opt\_sortheap**
- **opt\_locklist**
- **opt\_maxlocks**

#### **get** *optimizer-parameter*

このコマンドは、オブティマイザー・パラメーターの値を照会する場合に使用します。

- opt\_buffpage
- opt\_sortheap
- opt\_locklist
- opt\_maxlocks

### 使用上の注意

このツールは、異種データベース・パーティション構成から成るパーティション・データベース環境で使用されることがあります。この場合、ステートメント・コンパイルはコーディネーター・データベース・パーティションで行われますが、コーディネーター・データベース・パーティションのデータベース構成の設定値は、照会処理が行われるインスタンス内のデータベース・パーティションのものとは異なることがあります。

オブティマイザー・パラメーターの値が 0 の場合、オブティマイザー値は指定されていません。ステートメント・コンパイルでは、データベース構成での値が使用されます。

オブティマイザー・パラメーターの値を 0 に更新すると、前回更新した値がリセットされます。

更新アクションで非数値または負の値を指定すると、値は 0 に設定されます。

更新を有効にするには、データベース・パーティションのすべての接続を終了するとともに、データベース・パーティションを非アクティブ化する必要があります (アクティブ化状態になっている場合)。

オブティマイザー・パラメーターは、ステートメント・コンパイルにのみ使用されます。パーティション・データベース環境では、コーディネーター・データベース・パーティションに対して設定する必要があります。

テスト・システムで指定する実際の値を判断するには、**db2exfmt** ツールを使用して実動システムから Explain 出力を取得し、データベース・コンテキスト・セクションを参照します。

### 例

例 1: **opt\_sortheap**、**opt\_locklist**、および **opt\_maxlocks** の値を照会します。

```
db2fopt testdb get opt_sortheap opt_locklist opt_maxlocks
```

例 2: **opt\_buffpage** および **opt\_sortheap** パラメーターを設定します。

例えば、次のようなデータベース・コンテキスト・セクションが実動システムから返されたとします。

Database Context:

-----

Parallelism:	None
CPU Speed:	1.456395e-07
Comm Speed:	0
Buffer Pool size:	89000
Sort Heap size:	10000
Database Heap size:	1200

## db2fopt - 照会オプティマイザー・パラメーターの指定

```
Lock List size:      8000
Maximum Lock List:  10
Average Applications: 1
Locks Available:    93030
```

テスト・システムで **db2fopt** コマンドを使用して、**opt\_buffpage** を 89000 に、**opt\_sortheap** を 10000 に、**opt\_locklist** を 8000 に、**opt\_maxlocks** を 10 に、それぞれ設定します。

```
db2fopt testdb update opt_buffpage 89000 opt_sortheap 10000 opt_locklist 8000
                                     opt_maxlocks 10
```

## db2fs - ファースト・ステップ

DB2 製品の学習と使用を開始するために必要な各機能へのリンクが含まれている、「ファースト・ステップ」のインターフェースを立ち上げます。

UNIX オペレーティング・システムでは、**db2fs** は `sqllib/bin` ディレクトリーにあります。Windows オペレーティング・システムでは、`db2fs.exe` は `DB2PATH%bin` ディレクトリーにあります。

**db2fs** コマンドを発行するためには、以下のいずれかのブラウザがインストール済みでなければなりません。

- Internet Explorer 6.0 以上
- Mozilla 1.7 以上
- Firefox 2.0 以上

### 許可

SYSADM

### コマンド構文

UNIX オペレーティング・システムの場合

```
▶▶ db2fs [-h] [-b browser]
```

Windows オペレーティング・システムの場合

```
▶▶ db2fs
```

### コマンド・パラメーター

UNIX オペレーティング・システムの場合

**-h** コマンドの使用情報を表示します。

**-b browser**

使用するブラウザを指定します。指定されない場合、**db2fs** は **PATH** で指定されたディレクトリー内でブラウザを検索します。

Windows オペレーティング・システムの場合

なし



## db2gcf - DB2 インスタンスの制御

HA (高可用性) クラスタなどの中にある DB2 インスタンスの開始と停止とモニターを (通常は自動スクリプトによって) 実行します。

UNIX オペレーティング・システムでは、このコマンドは *INSTHOME*/sql/lib/bin にあります。*INSTHOME* はインスタンス所有者のホーム・ディレクトリーです。

Windows オペレーティング・システムでは、このコマンドは sql/lib/bin サブディレクトリーにあります。

### 許可

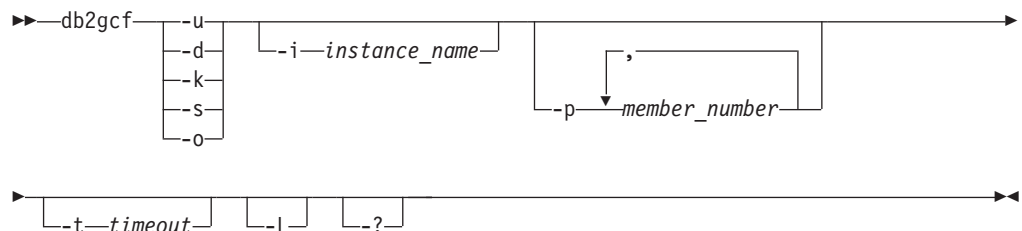
以下の権限のいずれか。

- インスタンス所有者
- Linux および UNIX オペレーティング・システム上の root ユーザー権限、または Windows オペレーティング・システム上のローカル管理者権限

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

- u 指定のインスタンスの指定のメンバーを開始します。
- d 指定のインスタンスの指定のメンバーを停止します。
- k 指定のインスタンスに関連するすべてのプロセスを除去します。 DB2 Enterprise Server Edition システムでは、このパラメーターを、リモート・シェル・ユーティリティー rsh または ssh を単一パーティション・データベース・インスタンスまたは複数パーティション・データベース・インスタンスのいずれかにセットアップする必要があります。
- s 指定のメンバーと指定のインスタンスの状況に戻します。戻される状態は、以下のとおりです。
  - Available: 指定のインスタンスの指定のメンバーは、プロセスの対象として使用できる状態です。
  - Operable: このインスタンスはインストールされていますが、現在使用できる状態ではありません。
  - Not operable: このインスタンスは、使用できる状態にできません。

## db2gcf - DB2 インスタンスの制御

- o それぞれの可能なアクションにデフォルトのタイムアウト値を戻します。-t パラメーターの値を指定すれば、すべてのデフォルトのタイムアウト値をオーバーライドできます。
- i *instance\_name*  
アクションの実行対象のインスタンス名。インスタンス名を指定しない場合は、**DB2INSTANCE** の値が使用されます。インスタンス名を指定しない場合に、**DB2INSTANCE** が設定されていなければ、以下のエラーが戻されます。  
db2gcf Error: Neither DB2INSTANCE is set nor instance passed.
- p *member\_number*  
パーティション・データベース環境で、ローカル・ノードのみに対してアクションを実行するメンバー番号を指定します (このコマンドではリモート・メンバーはモニターされません)。スペースを使用せずに、コンマで区切ってメンバー番号を指定してください。値を指定しない場合、デフォルトは 0 です。単一パーティション・データベース環境では、この値は無視されません。
- t *timeout*  
秒単位のタイムアウト値。指定の時間内に処理が完了しない場合、**db2gcf** コマンドは失敗します。それぞれの可能なアクションにデフォルトのタイムアウト値がありますが、-t パラメーターの値を指定すれば、すべてのデフォルトのタイムアウト値をオーバーライドできます。
- L エラー・ロギングを有効にします。インスタンス固有の情報は、インスタンス・ログ・ディレクトリーの **db2diag** ログ・ファイルに書き込まれます。インスタンス固有でない情報は、システム・ログ・ファイルに書き込まれます。
- ? ヘルプ情報を表示します。このオプションを指定すると、他のすべてのオプションは無視され、ヘルプ情報だけが表示されます。

### 例

1. 以下の例は、メンバー 0 のインスタンス **stevera** を開始します。

```
db2gcf -u -p 0 -i stevera
```

以下の出力が戻されます。

```
Instance : stevera
DB2 Start : Success
Member 0 : Success
```

2. 以下の例は、メンバー 0 のインスタンス **stevera** の状況を戻します。

```
db2gcf -s -p 0 -i stevera
```

以下の出力が戻されます。

```
Instance : stevera
DB2 State
Member 0 : Available
```

3. 以下の例は、メンバー 0 のインスタンス **stevera** を停止します。

```
db2gcf -d -p 0 -i stevera
```

以下の出力が戻されます。

```
Instance : stevera  
DB2 Stop : Success  
Member 0 : Success
```

### 使用上の注意

共に使用すると、**-k** パラメーターおよび **-p** パラメーターは、指定されたメンバーからのすべてのプロセスが除去されることがないようにします。むしろ、インスタンス (全メンバー) 上のすべてのプロセスが除去されます。

### 戻りコード

以下に示すのは、このコマンドに関するすべての戻りコードのリストです。

```
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
```

## db2gov - DB2 管理プログラム

データベースに対して実行しているアプリケーションの振る舞いをモニターし変更します。デフォルトでは、デーモンはすべてのデータベース・パーティションで開始されますが、特定のデータベース・パーティションで単一のデーモンを開始する場合には、フロントエンド・ユーティリティーを使用できます。

**重要:** DB2 バージョン 9.5 で導入されたワークロード管理フィーチャーにより、DB2 ガバナー・ユーティリティーはバージョン 9.7 で非推奨となり、将来のリリースで除去される可能性があります。これはDB2 pureScale環境ではサポートされていません。詳しくは、『DB2 ガバナーと Query Patroller が推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0054901.html>) を参照してください。

**注:** DB2 for Linux, UNIX, and Windows バージョン 10.1 リリース以降、AIX 5.3 オペレーティング・システムはサポートされません。DB2 for Linux, UNIX, and Windows バージョン 9.7 が、AIX 5.3 オペレーティング・システムをサポートしている最後のリリースです。AIX 6.1 オペレーティング・システムが、サポートされる最小レベルです。

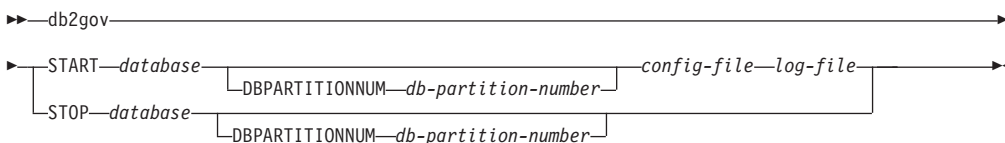
### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL

db2nodes.cfg ファイルを定義したインスタンスのある環境では、**db2\_a11** コマンドを呼び出すための権限も必要です。db2nodes.cfg ファイルを定義した環境には、db2nodes.cfg にデータベース・パーティションを定義したパーティション・データベース環境と単一パーティション・データベース環境が含まれます。

### コマンド構文



### コマンド・パラメーター

#### START database

管理プログラム・デーモンを開始して、指定されたデータベースをモニターします。データベース名またはデータベース別名のいずれかを指定できます。指定する名前は、管理プログラム構成ファイルに指定する名前と同じでなければなりません。モニターされる各データベースにつき 1 つのデーモンが実行されます。パーティション・データベース環境では、各データベース・パーティションにつき 1 つのデーモンが実行されます。複数のデータベースに対して管理プログラムが実行されている場合には、データベース・サーバーでも複数のデーモンが実行されます。

**DBPARTITIONNUM** *db-partition-number*

管理プログラム・デーモンを開始または停止するデータベース・パーティションを指定します。この番号は、データベース・パーティション構成ファイルで指定した番号と同じでなければなりません。

*config-file*

データベースをモニターする際に使用する構成ファイルを指定します。構成ファイルのデフォルトのロケーションは、`sql1lib` ディレクトリーです。指定したファイルがこのディレクトリーにない場合、フロントエンドは、指定したこの名前をファイルの絶対パス名であると見なします。

*log-file* 管理プログラムがログ記録を書き込むファイルのベース名を指定します。ログ・ファイルは、`sql1lib` ディレクトリーのログ・サブディレクトリーに保管されます。管理プログラムが実行されているデータベース・パーティションの数は、自動的にログ・ファイル名に追加されます。例えば、`mylog.0`, `mylog.1`, `mylog.2` となります。

**STOP** *database*

指定したデータベースをモニターしている管理プログラム・デーモンを停止します。パーティション・データベース環境では、フロントエンド・ユーティリティーは、データベース・パーティション構成ファイル `db2nodes.cfg` を読み取ることによって、すべてのデータベース・パーティション上の管理プログラムを停止します。

**使用上の注意**

管理プログラム構成ファイルの `[action]` 節では、`nice nnn` パラメーターを設定して、アプリケーションに対して動作しているエージェントの相対的な優先度を上下させることができます。追加情報は、「データベース・パフォーマンスのチューニング」というガイドの『ガバナーの規則エレメント』を参照してください。

注: AIX 6.1 以降では、アプリケーションに対して動作しているエージェントの相対的な優先度を上げられるようにするには、インスタンス所有者に `CAP_NUMA_ATTACH` の機能が必要です。この機能を付与するには `root` としてログオンし、次のコマンドを実行します。

```
chuser capabilities=CAP_NUMA_ATTACH,CAP_PROPAGATE
```

Solaris 10 以降では、アプリケーションに対して動作しているエージェントの相対的な優先度を上げられるようにするには、インスタンス所有者に `proc_prioctl` 特権が必要です。この特権を付与するには、`root` としてログオンし、次のコマンドを実行します。

```
usermod -K defaultpriv=basic,proc_prioctl db2user
```

この例では、`proc_prioctl` がユーザー `db2user` のデフォルトの特権セットに追加されます。

加えて、DB2 が Solaris の非グローバル・ゾーンで実行される場合は、`proc_prioctl` 特権をゾーンの制限特権セットに追加する必要があります。この特権をゾーンに付与するには、`root` としてログオンし、次のコマンドを実行します。

```
global# zonecfg -z db2zone
zonecfg:db2zone> set limitpriv="default,proc_prioctl"
```

## db2gov - DB2 管理プログラム

この例では、proc\_prioctl がゾーン db2zone の制限特権セットに追加されます。

Solaris 9 では、エージェントの相対優先順位を上げるための DB2 用の機能がありません。DB2 ガバナーの ACTION NICE 節を使用するには、Solaris 10 以降にアップグレードします。



## db2gpmmap - 分散マップの取得

データベースが既にセットアップされていて、そのデータベースのデータベース・パーティション・グループが定義されている場合、**db2gpmmap** は、カタログ・パーティション・データベース・サーバーから、データベース表またはデータベース・パーティション・グループの分散マップを取得します。

### 許可

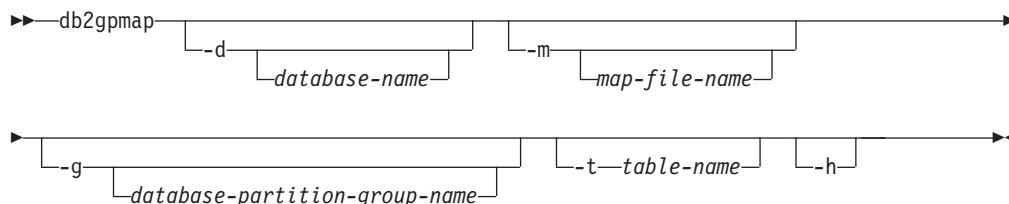
以下の権限の両方が必要です。

- システム・カタログ表に対する読み取りアクセス
- db2gpmmap.bnd に対するパッケージの BIND 特権と EXECUTE 特権

### 必要な接続

**db2gpmmap** を使用する前に、データベース・マネージャーを開始し、**db2gpmmap.bnd** をデータベースにバインドしておく必要があります。バインドしていない場合、**db2gpmmap** はそのファイルをバインドしようとします。

### コマンド構文



### コマンド・パラメーター

#### **-d** *database-name*

分散マップを生成するデータベースの名前を指定します。データベース名を指定しない場合は、**DB2DBDFT** 環境変数の値が使用されます。**DB2DBDFT** が設定されていない場合のデフォルトは、**SAMPLE** データベースです。

#### **-m** *map-file-name*

分散マップを保存するファイルの完全修飾ファイル名を指定します。デフォルトは **db2split.map** です。

#### **-g** *database-partition-group-name*

分散マップを生成するデータベース・パーティション・グループの名前を指定します。デフォルトは **IBMDEFAULTGROUP** です。

#### **-t** *table-name*

表名を指定します。

#### **-h**

使用法情報を表示します。

### 例

以下の例は、**SAMPLE** データベースの **ZURBIE.SALES** 表の分散マップを **C:%pmaps%zurbie\_sales.map** というファイルに抽出します。

```
db2gpmmap -d SAMPLE -m C:%pmaps%zurbie_sales.map -t ZURBIE.SALES
```



## db2iauto - インスタンスの自動開始

毎回のシステム再始動後のインスタンスの自動開始を有効にするか、または無効にします。このコマンドは Linux および UNIX オペレーティング・システムでのみ使用できます。このコマンドは DB2 pureScale インスタンスではサポートされません。

### 許可

以下の権限のいずれか。

- root ユーザー権限
- SYSADM

### 必要な接続

なし

### コマンド構文

```
▶▶ db2iauto [-on|-off] instance-name ▶▶▶▶
```

### コマンド・パラメーター

**-on** 指定したインスタンスの自動開始を有効にします。

**-off** 指定したインスタンスの自動開始を無効にします。

*instance-name*

インスタンスのログイン名。

## db2iclus - Microsoft Cluster Server

ユーザーが Microsoft Cluster Server (MSCS) 環境で、インスタンスおよび DB2 Administration Server (DAS) を追加、ドロップ、マイグレーション、マイグレーション解除を行うためのコマンドです。このコマンドは Windows オペレーティング・システムでのみ使用できます。

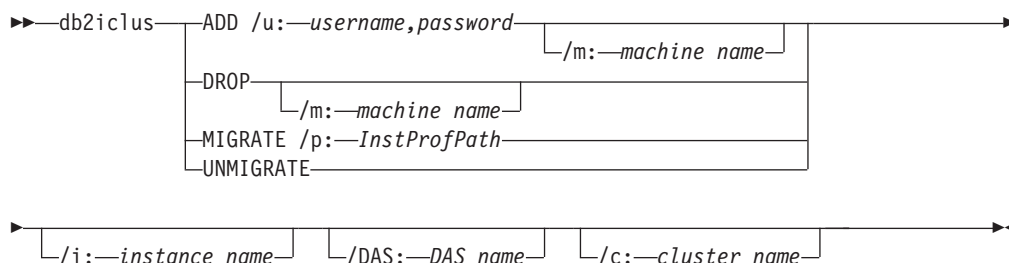
### 許可

タスクが実行されるマシンで、ローカル管理者権限が必要です。リモート・マシンをインスタンス追加するか、またはインスタンスからリモート・マシンを除去する場合、ターゲット・マシンでローカル管理者権限が必要です。

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

**ADD** MSCS ノードを DB2 MSCS インスタンスに追加します。

**DROP** MSCS ノードを DB2 MSCS インスタンスから除去します。

#### MIGRATE

非 MSCS インスタンスを MSCS インスタンスにマイグレーションします。

#### UNMIGRATE

MSCS マイグレーションを取り消します。

#### /DAS:DAS name

DAS 名を指定します。このオプションは、DB2 Administration Server に対してクラスター操作を実行する際に必要です。

#### /c:cluster name

デフォルトまたは現行のクラスターの名前と違う場合に MSCS クラスター名を指定します。

#### /p:instance profile path

インスタンス・プロファイル・パスを指定します。このパスは、クラスター・ディスクにあるはずなので、DB2 が MSCS クラスター中のマシンのいずれかでアクティブである場合アクセス可能です。このオプションは、非 MSCS インスタンスを MSCS インスタンスにマイグレーションする際に必要です。

**/u:username,password**

DB2 サービスのアカウント名およびパスワードを指定します。このオプションは、別の MSCS ノードを DB2 MSCS パーティション・データベース・インスタンスに追加する場合に必要です。

**/m:machine name**

MSCS ノードの追加または除去に使用するリモート・コンピューターの名前を指定します。

**/i:instance name**

デフォルトまたは現行のクラスタの名前と違う場合にインスタンス名を指定します。

**例**

この例は、**db2iclus** コマンドを使用して DB2 インスタンスを手動で構成し、2 つのマシン WA26 と WA27 から成るホット・スタンバイ構成で実行する方法を示します。

1. 開始するには、MSCS および DB2 Enterprise Server Edition が、両方のマシンにインストールされている必要があります。
2. マシン WA26 に、DB2 という新しいインスタンスを作成します。

```
db2icrt DB2
```

3. 「Windows サービス (Windows Services)」ダイアログ・ボックスから、手動で開始できるようにインスタンスが構成されていることを確認します。
4. DB2 インスタンスが実行中である場合、**DB2STOP** コマンドで停止します。
5. WA26 から DB2 リソース・タイプをインストールします。

```
c:>db2wo1fi i
ok
```

**db2wo1fi** コマンドが「エラー : 183」を戻す場合、既にインストールされているということです。確認するために、リソース・タイプを一度ドロップして再度追加することができます。また、リソース・タイプが存在しないと、クラスタ管理には表示されません。

```
c:>db2wo1fi u
ok
c:>db2wo1fi i
ok
```

6. WA26 から、**db2iclus** コマンドを使用して、DB2 インスタンスをクラスタ・インスタンスに変換します。

```
c:¥>db2iclus migrate /i:db2 /c:mycluster /m:wa26 /p:p:¥db2profs
```

```
DBI1912I The DB2 Cluster command was successful.
Explanation: The user request was successfully processed.
User Response: No action required.
```

ディレクトリー `p:¥db2profs` はクラスタ・ドライブにあるはずで、既に存在していなければなりません。このドライブも、現在マシン WA26 の所有である必要があります。

7. WA26 から、**db2iclus** コマンドを使用して、他のマシンを DB2 クラスタ・リストに追加します。

```
c:\>db2iclus add /i:db2 /c:mycluster /m:wa27
```

```
DBI1912I The DB2 Cluster command was successful.
```

```
Explanation: The user request was successfully processed.
```

```
User Response: No action required.
```

このコマンドは、クラスター中で連続する各マシンごとに実行されます。

8. クラスター管理から、「DB2 Group」という名前の新しいグループを作成します。
9. クラスター管理から、物理ディスク・リソース、ディスク O とディスク P を DB2 Group に移動します。
10. クラスター管理から、公衆ネットワークにある「mscs5」というタイプ「IP Address」の新しいリソース・タイプを作成します。このリソースも、DB2 Group に所属していなければなりません。これは非常に有効な IP アドレスで、このアドレスはネットワーク上のどのマシンにも対応してはなりません。IP アドレスのリソース・タイプをオンラインにし、そのアドレスがリモート・マシンから確実に PING できるようにします。
11. クラスター管理から、DB2 Group に所属する、タイプ「DB2」の新しいリソース・タイプを作成します。このリソースの名前は、インスタンス名とまったく同一でなければならないので、ここでは DB2 という名前になります。クラスター管理が DB2 リソースに関連する従属関係のプロンプトを出すので、DB2 リソースがディスク O、ディスク P、および mscs5 に依存していることを確認します。
12. 必要なら、クラスター管理を介して、また **DB2\_FALLBACK** プロファイル変数を使用して、フォールバック用に DB2 Group を構成します。
13. すべてのデータをディスク O およびディスク P に入れ、すべてのデータベースを作成するか、またはリストアします。
14. フェイルオーバー構成をテストします。

### 使用上の注意

MSCS フェイルオーバー環境で実行するためにインスタンスを移動するには、まず現行マシンでインスタンスをマイグレーションしてから、**ADD** オプションを指定した **db2iclus** を使って他の MSCS ノードをインスタンスに追加する必要があります。

MSCS インスタンスを正規のインスタンスに復帰するには、まず、**DROP** オプションを指定した **db2iclus** を使用して、インスタンスから他のすべての MSCS ノードをドロップすることが必要です。次に、現行マシン上のインスタンスのマイグレーションを取り消してください。

## db2icrt - インスタンスの作成

DB2 pureScale インスタンスを含めて DB2 インスタンスを作成します。また、このコマンドを使用して、DB2 pureScale インスタンスの作成の一環として、初期 DB2 メンバーおよびクラスター・キャッシング・ファシリティーを作成することもできます。

Linux および UNIX オペレーティング・システムでは、**db2icrt** は *DB2DIR/instance* にあります。*DB2DIR* は、DB2 データベース・システムがインストールされているインストール・ディレクトリを表します。Windows オペレーティング・システムでは、**db2icrt** は *DB2PATH%bin* にあります。*DB2PATH* は DB2 コピーがインストールされているディレクトリです。

**db2icrt** コマンドは、インスタンス所有者のホーム・ディレクトリーに DB2 インスタンスを作成します。DB2 pureScale インスタンスは、DB2 pureScale 環境ごとに 1 つのみ作成できます。

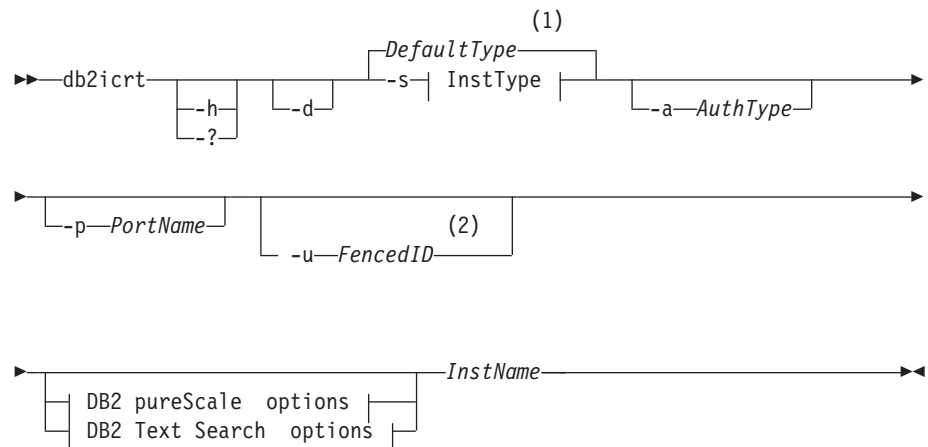
注: このコマンドは、DB2 データベース製品の非 root インストールには利用できません。

### 許可

Linux および UNIX オペレーティング・システムでは root ユーザー権限、Windows オペレーティング・システムではローカル管理者権限が必要です。

### コマンド構文

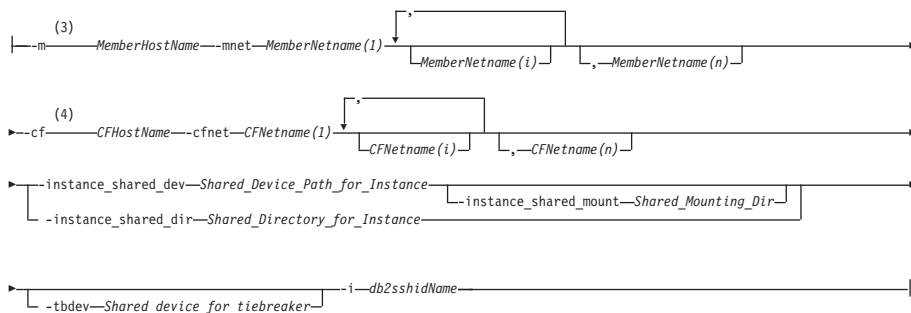
Linux および UNIX オペレーティング・システムの場合



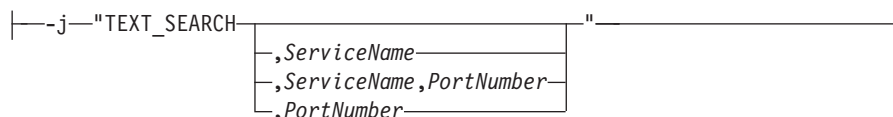
#### InstType:



**DB2 pureScale options:**



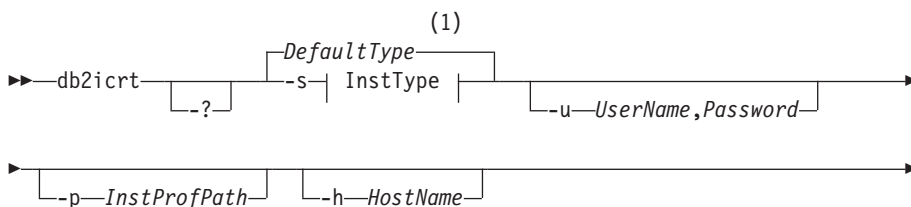
**DB2 Text Search options:**

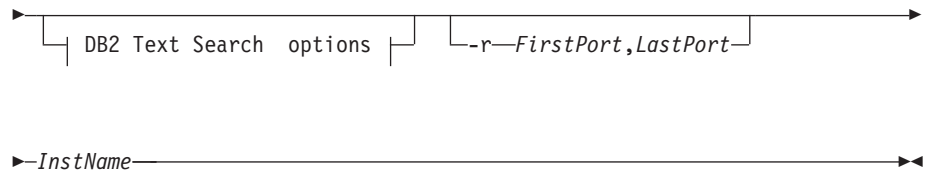
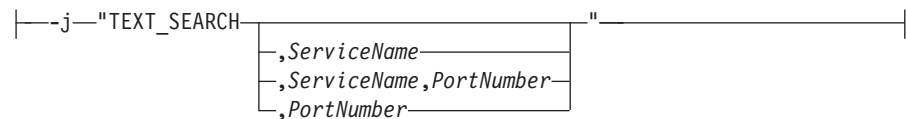


**注:**

- 1 インスタンス・タイプを `-s` で指定しなかった場合には、作成されるデフォルトのインスタンス・タイプは、DB2 コピーのライセンスで作成が許可されているタイプのうち、*InstType* にリストされているタイプの中で最も高位のタイプです。
- 2 クライアント・インスタンスを作成する場合、`-u FencedID` は有効なオプションではありません。
- 3 *MemberHostName:MemberNetname* 形式は、`-m` オプションでは推奨されなくなりました。今後廃止される可能性があります。DB2 pureScale Feature での IPv6 サポートには、`-m` および `-mnet` オプションの両方を使用する新しい形式が必要です。
- 4 *CFHostName:CFNetames* 形式は、`-cf` オプションでは推奨されなくなりました。今後廃止される可能性があります。DB2 pureScale Feature での IPv6 サポートには、`-cf` および `-cfnet` オプションの両方を使用する新しい形式が必要です。

**Windows オペレーティング・システムの場合**



**InstType:****DB2 Text Search options:****注:**

- 1 インスタンス・タイプを `-s` で指定しなかった場合には、作成されるデフォルトのインスタンス・タイプは、DB2 コピーのライセンスで作成が許可されているタイプのうち、*InstType* にリストされているタイプの中で最も高位のタイプです。

**コマンド・パラメーター****Linux および UNIX オペレーティング・システムの場合**

- `-?` 使用情報を表示します。
- `-h` 使用情報を表示します。
- `-d` デバッグ・モードをオンにします。db2icrt.trc.ProcessID として、/tmp 内にデフォルト名でトレース・ファイルを保存します。このオプションは、DB2 データベース・サポートからの指示があった場合にのみ使用してください。
- `-a AuthType`  
インスタンスの認証タイプ (SERVER、CLIENT、または SERVER\_ENCRYPT) を指定します。デフォルトは SERVER です。
- `-j "TEXT_SEARCH"`  
DB2 Text Search サーバーを、生成されたサービス名と TCP/IP ポート番号のデフォルト値を使用して構成します。インスタンス・タイプがクライアント の場合、このパラメーターは使用できません。
- `-j "TEXT_SEARCH, servicename"`

DB2 Text Search サーバーを、提供されたサービス名と自動的に生成されたポート番号を使用して構成します。サービス名に `services` ファイル内で割り当てられたポート番号がある場合、それはその割り当てられたポート番号を使用します。

**-j "TEXT\_SEARCH, servicename, portnumber"**

DB2 Text Search サーバーを、提供されたサービス名とポート番号を使用して構成します。

**-j "TEXT\_SEARCH, portnumber"**

DB2 Text Search サーバーを、デフォルトのサービス名と提供されたポート番号を使用して構成します。有効なポート番号は 1024 - 65535 の範囲になければなりません。

**-p <TCP/IP PortName>**

インスタンスが使用する TCP/IP ポート名または番号を指定します。さらに、このオプションによって、データベース・マネージャ構成パラメーター **SVCENAME** を DB2 インスタンスに対して構成します。

**-m MemberHostName:NetName1**

インスタンスの作成時に DB2 メンバーとしてセットアップするホストを指定します。このパラメーターは、DB2 pureScale 環境で必須です。 **db2icrt** コマンドでセットアップできる DB2 メンバーは 1 つだけです。 **db2iupdt -add** コマンドを使用して、さらに DB2 メンバーを追加することができます。 *NetName1* 構文は推奨されなくなりました。今後のリリースで廃止される可能性があります。代わりに **-mnet** パラメーターを使用してください。

*MemberHostName* は正規のホスト名にしてください (例えば、'hostname' コマンドの出力はローカル・ホスト上で実行します)。ここに指定される *NetName1* 値は、**-cf** パラメーターで指定されているのと同じサブネットに属している必要があります。

**-mnet MemberNetName**

このパラメーターは、推奨されなくなった **-m MemberHostName:NetName1** パラメーターの *:NetName1* 構文に取って代わるものです。クラスター相互接続ネットワーク名を指定します。これは、DB2 pureScale インスタンスで、メンバーとクラスター・キャッシング・ファシリティー (CF と呼ばれる) との高速通信に使用される相互接続のホスト名です。

*MemberNetName* は、**-cf** パラメーターで指定された同じサブネットの 1 つに属していて、クラスター相互接続ネットワーク名に対応している必要があります (例えば *db2\_<hostname\_ib0>*)。

**-cf CFHostName:NetName2**

インスタンス作成時にクラスター・キャッシング・ファシリティー (CF ともいう) としてセットアップするホストを指定します。このパラメーターは、DB2 pureScale 環境で必須です。 **db2icrt** コマンドでセットアップできる CF は 1 つだけです。 **db2iupdt -add** コ



マンドを使用して、さらに CF を追加することができます。

*NetName2* 構文は推奨されなくなりました。今後のリリースで廃止される可能性があります。代わりに **-cfnet** パラメーターを使用してください。

**-cfnet** *CFNetName*

このパラメーターは、推奨されなくなった **-cf**

*CFHostName:NetName2* パラメーターの *:NetName2* 構文に取って代わるものです。クラスター相互接続ネット名を指定します。これは、DB2 pureScale インスタンスで、メンバーと CF との高速通信に使用される相互接続のホスト名です。

*CFNetName* は、**-m** パラメーターで指定されたサブネットと同じサブネットに属していて、クラスター相互接続ネット名に対応している必要があります (例えば *db2\_<hostname\_ib0>*)。

**-instance\_shared\_dev** *Shared\_Device\_Path\_for\_Instance*

インスタンス共有ファイルとデフォルトのデータベース・パスを保持する DB2 pureScale インスタンスをセットアップするのに必要な共有ディスク装置パスを指定します。例えば、*dev/hdisk1* と指定します。DB2 pureScale インスタンスについて、共有ディレクトリーは、すべてのホストでアクセス可能でなければなりません。このオプションの値は、**-tbdev** オプションと同じ値であってはなりません。

**-instance\_shared\_dev** パラメーターを指定すると、DB2 インストーラーによって、DB2 クラスター・ファイル・システムが作成されます。

**-instance\_shared\_dev** パラメーターと **-instance\_shared\_dir** パラメーターは、相互に排他的です。

**-instance\_shared\_mount** *Shared\_Mounting\_Dir*

新規 IBM General Parallel File System (GPFS) ファイル・システム用のマウント・ポイントを指定します。指定するパスは、新規で空のパスでなければならず、既存の GPFS ファイル・システムの内部にネストされたパスであってはなりません。

**-instance\_shared\_dir** *Shared\_Directory\_for\_Instance*

インスタンス共有ファイルとデフォルトのデータベース・パスを保持する DB2 pureScale インスタンスをセットアップするのに必要な、共有ファイル・システム (GPFS) 内のディレクトリーを指定します。例えば、*/sharedfs* となります。DB2 pureScale インスタンスについて、そのディスクはすべてのホストでアクセス可能でなければなりません。このオプションの値は、**-tbdev** オプションと同じ値であったり、インストール・パスであってはなりません。

**-instance\_shared\_dir** パラメーターを指定すると、DB2 インストーラーによって、ユーザー管理ファイル・システムが作成されます。ユーザー管理のファイル・システムはすべてのホストで使用可能でなければならず、GPFS ファイル・システムでなければなりません。

**-instance\_shared\_dir** パラメーターと **-instance\_shared\_dev** パラメーターは、相互に排他的です。

**-tbdev** *Shared\_device\_for\_tiebreaker*

データの保全性を確実に維持するために、DB2 pureScale 環境でタイブレーカーとして機能するデバイスの共有装置パスを指定します。このオプションの値は、**-instance\_shared\_dev** オプションまたは **-instance\_shared\_dir** オプションと同じ値であってはなりません。DB2 クラスター・サービスのタイブレーカーが初めて作成されるときには、このオプションは必須です。ディスク装置には、それに関連付けられたファイル・システムがあってはなりません。DB2 クラスター・サービスのピア・ドメインが既に存在する場合は、このオプションは無効です。

**-i** *db2sshidName*

ホスト間でセキュア・シェル (SSH) ネットワーク・プロトコルを使用するために必要な非 root ユーザー ID を指定します。指定されたユーザー ID は、特別な特権を持たないユーザーである必要があります。DB2 管理 GPFS ファイル・システムに対してのみ有効です。

**-s** *InstType*

作成するインスタンスのタイプを指定します。**-s** オプションは、**db2icrt** を実行しているインストール済みの製品に関連したデフォルト以外のインスタンスを作成する場合にのみ指定してください。有効な値は以下のとおりです。

**dsf** ローカルおよびリモート・クライアントで DB2 データベース・サーバーの DB2 pureScale インスタンスを作成するために使用します。このオプションは、IBM DB2 pureScale Feature のデフォルトのインスタンス・タイプです。

**ese** ローカルおよびリモート・クライアントでデータベース・サーバーのインスタンスを作成するために使用します。このオプションは、DB2 Enterprise Server Edition または DB2 Advanced Enterprise Server Edition のデフォルトのインスタンス・タイプです。

**wse** ローカルおよびリモート・クライアントでデータベース・サーバーのインスタンスを作成するために使用します。このオプションは DB2 Workgroup Server Edition、DB2 Express® Edition または DB2 Express-C、および DB2 Connect Enterprise Edition のデフォルトのインスタンス・タイプです。

**standalone**

ローカル・クライアントでデータベース・サーバーのインスタンスを作成するために使用します。

**client** クライアントのインスタンスを作成するために使用します。このオプションは IBM Data Server Client、IBM Data Server Runtime Client、および DB2 Connect Personal Edition のデフォルトのインスタンス・タイプです。

DB2 データベース製品は、デフォルトのインスタンス・タイプと、デフォルトより下位のインスタンス・タイプをサポートしています。例えば、DB2 Enterprise Server Edition は、ese、wse、standalone、および client のインスタンス・タイプをサポートしています。

**-u Fenced ID**

fenced ユーザー定義関数および fenced ストアード・プロシージャを実行するユーザー ID の名前を指定します。クライアント・インスタンスを作成しない場合は、**-u** オプションは必須です。

*InstName*

オペレーティング・システムの既存ユーザー名であるインスタンス名を指定します。インスタンス名は、**db2icrt** コマンドの最後の引数でなければなりません。

**Windows オペレーティング・システムの場合**

*InstName*

インスタンスの名前を指定します。

**-s InstType**

作成するインスタンスのタイプを指定します。現在 4 種類の DB2 インスタンス・タイプがあります。有効な値は以下のとおりです。

**client** クライアントのインスタンスを作成するために使用します。このオプションは IBM Data Server Client、IBM Data Server Runtime Client、および DB2 Connect Personal Edition のデフォルトのインスタンス・タイプです。

**standalone**

ローカル・クライアントでデータベース・サーバーのインスタンスを作成するために使用します。

**ese** パーティション・データベース環境サポートのあるローカルおよびリモート・クライアントでデータベース・サーバーのインスタンスを作成するために使用します。

`-s ese -u Username, Password`

ESE インスタンス・タイプおよびパーティション・データベース環境インスタンスを作成するには、**db2icrt** でオプションを使用する必要があります。

**wse** ローカルおよびリモート・クライアントでデータベース・サーバーのインスタンスを作成するために使用します。このオプションは DB2 Workgroup Server Edition、DB2 Express Edition または DB2 Express-C、および DB2 Connect Enterprise Edition のデフォルトのインスタンス・タイプです。

DB2 データベース製品は、デフォルトのインスタンス・タイプと、デフォルトより下位のインスタンス・タイプをサポートしています。例えば、DB2 Enterprise Server Edition は、ese、wse、standalone、および client のインスタンス・タイプをサポートしています。

## db2icrt - インスタンスの作成

### **-u** *Username, Password*

DB2 サービスのアカウント名およびパスワードを指定します。このオプションはパーティション・データベース・インスタンスを作成する時に必要です。

### **-p** *InstProfPath*

インスタンス・プロファイル・パスを指定します。

### **-h** *HostName*

現行のマシンに対して複数のデフォルトの TCP/IP ホスト名がある場合、それらをオーバーライドします。TCP/IP ホスト名は、デフォルト・データベース・パーティション (データベース・パーティション 0) を作成する際に使用されます。このオプションは、パーティション・データベース・インスタンスに対してのみ有効です。

### **-r** *PortRange*

MPP モードで実行する場合に、パーティション・データベース・インスタンスによって使用される TCP/IP ポートの範囲を指定します。例えば、**-r 50000,50007**。このオプションを指定した場合、ローカル・マシンの `services` ファイルは、以下の項目で更新されます。

```
DB2_InstName      baseport/tcp
DB2_InstName_END  endpoint/tcp
```

### **/j** *"TEXT\_SEARCH"*

DB2 Text Search サーバーを、生成されたサービス名と TCP/IP ポート番号のデフォルト値を使用して構成します。インスタンス・タイプがクライアント の場合、このパラメーターは使用できません。

### **/j** *"TEXT\_SEARCH, servicename"*

DB2 Text Search サーバーを、提供されたサービス名と自動的に生成されたポート番号を使用して構成します。サービス名に `services` ファイル内で割り当てられたポート番号がある場合、それはその割り当てられたポート番号を使用します。

### **/j** *"TEXT\_SEARCH, servicename, portnumber"*

DB2 Text Search サーバーを、提供されたサービス名とポート番号を使用して構成します。

### **/j** *"TEXT\_SEARCH, portnumber"*

DB2 Text Search サーバーを、デフォルトのサービス名と提供されたポート番号を使用して構成します。有効なポート番号は 1024 - 65535 の範囲になければなりません。

**-?** 使用法情報を表示します。

## 例

1. インスタンス所有者 `db2sdin1`、fenced ユーザー `db2sdfe1` に対して DB2 pureScale インスタンスを作成するには、次のコマンドを実行します。

```
DB2DIR/instance/db2icrt
-cf host1.domain.com -cfnet host1.domain.com-ib0
-m host2.domain.com -mnet host2.domain.com-ib0
```

```
-instance_shared_dev /dev/hdisk1
-tbdev /dev/hdisk2
-u db2sdfel
db2sdin1
```

ここで、*DB2DIR* は、DB2 のコピーのインストール・ロケーションを示します。DB2 pureScale インスタンスである db2sdin1 は、host1 上に CF を持ち、host2 上にメンバーを持つことになります。さらにこのコマンドは /dev/hdisk1 を使用して、インスタンス共有ファイルを保管するための共有ファイル・システムを作成し、タイプレーカーの共有装置パスとして /dev/hdisk2 をセットアップします。

2. ユーザー ID db2inst1 に対して DB2 Enterprise Server Edition インスタンスを作成するには、次のコマンドを実行します。

```
DB2DIR/instance/db2icrt -s ese -u db2fenc1 db2inst1
```

ここで、*DB2DIR* は、DB2 のコピーのインストール・ロケーションを示します。

3. インスタンス所有者 db2sdin1 および fenced ユーザー db2sdfel で、DB2 製品によって管理される既存のファイル・システム (GPFS) を使用する DB2 pureScale インスタンスを作成するには、以下のコマンドを実行します。

```
DB2DIR/instance/db2icrt
-cf host1.domain.com -cfnet host1.domain.com-ib0
-m host2.domain.com -mnet host2.domain.com-ib0
-tbdev /dev/hdisk2
-u db2sdfel
db2sdin1
```

ここで、*DB2DIR* は、DB2 のコピーのインストール・ロケーションを示します。

4. インスタンス所有者 db2sdin1 および fenced ユーザー db2sdfel で、ユーザー管理の既存の GPFS ファイル・システム (/gpfs\_shared\_dir) を使用して DB2 pureScale インスタンスを作成するには、次のコマンドを実行します。

```
DB2DIR/instance/db2icrt
-cf host1.domain.com -cfnet host1.domain.com-ib0
-m host2.domain.com -mnet host2.domain.com-ib0
-instance_shared_dir /gpfs_shared_dir
-tbdev /dev/hdisk2
-u db2sdfel
db2sdin1
```

ここで、*DB2DIR* は、DB2 のコピーのインストール・ロケーションを示します。

5. AIX マシン上でユーザー ID db2inst1 のインスタンスを作成するには、以下のコマンドを発行します。

クライアント・マシンの場合:

```
DB2DIR/instance/db2icrt db2inst1
```

サーバー・マシンの場合:

```
DB2DIR/instance/db2icrt -u db2fenc1 db2inst1
```

ここで、db2fenc1 は、fenced ユーザー定義関数および fenced ストアード・プロシージャを実行するユーザー ID です。

### 使用上の注意

- インスタンス・ユーザーはすべてのホスト上に存在していて、それらは UID、GID、グループ名、およびホーム・ディレクトリー・パスが同じでなければなりません。同じ規則が `fenced` ユーザーに適用されます。 `db2icrt` コマンドが正常に実行された後、DB2 インストーラーは、全ホストにわたってインスタンス・ユーザーのための SSH をセットアップします。
- `db2icrt` コマンドを使用するときは、インスタンスの名前が既存ユーザーの名前に一致している必要があります。
- DB2 pureScale 環境ごとに、存在できるインスタンスは 1 つだけです。
- DB2 インスタンスを作成する際には、以下の制限事項を考慮してください。
  - 既存の ID を使用して DB2 インスタンスを作成する場合は、ID がロックされておらず、パスワードが期限切れでないことを確認してください。
- また、`db2isetup` コマンドを使用すると、グラフィカル・インターフェースで、DB2 インスタンスの作成および更新、複数ホストの追加を実行することができます。
- `root` ユーザーになるために `login` コマンドではなく `su` コマンドを使用している場合は、`su` コマンドを `-` オプションを指定して発行して、あたかも `login` コマンドを使用してシステムにログインしているかのように、プロセス環境を設定することを指示する必要があります。
- `root` ユーザー用の DB2 インスタンス環境を入手しないでください。DB2 インスタンス環境を入手している場合に、`db2icrt` を実行することはサポートされていません。
- 以前に DB2 pureScale インスタンスを作成してドロップしたことがある場合、`-instance_shared_dev` パラメーター指定を使ってそれを再作成することはできません。それは、DB2 クラスタ・ファイル・システムが既に作成されている可能性があるからです。以前に作成された共有ファイル・システムを指定するには、以下のようにします。
  - 既存の GPFS 共有ファイル・システムが DB2 pureScale Feature によって作成および管理されている場合には、`-instance_shared_dev` パラメーターと `-instance_shared_dir` パラメーターはいずれも使用しないでください。
  - 既存の GPFS 共有ファイル・システムが DB2 pureScale Feature によって作成および管理されているのではない場合には、`-instance_shared_dir` パラメーターを使用してください。
- AIX 6.1 (またはそれ以上) では、このコマンドは、システム・ワークロード・パーティション (WPAR) グローバル環境内の共有 DB2 コピーから実行する場合、`root` ユーザーとして実行されなければなりません。WPAR は DB2 pureScale 環境ではサポートされていません。





### コマンド・パラメーター

#### Linux および UNIX オペレーティング・システムの場合

- d DB2 データベース・サポートによって使用されるデバッグ・モードに入ります。
- h | -? 使用情報を表示します。
- g **db2idrop** が DB2 pureScale インスタンスで使用される場合には、このパラメーターは必須です。すべてのホスト上で DB2 pureScale インスタンスをドロップすることを指定します。このパラメーターは、DB2 pureScale インスタンス内のすべてのホストで、すべての DB2 メンバーと、すべてのクラスター・キャッシング・ファシリティーが停止されていることを必要とします。他のすべてのインスタンス・タイプのドロップについては、このオプションは無視されません。
- f このパラメーターは、推奨されていません。  
強制アプリケーション・フラグを指定します。このフラグを指定すると、このインスタンスを使用しているすべてのアプリケーションが強制的に終了させられます。このパラメーターは DB2 pureScale 環境ではサポートされません。

*InstName*

インスタンスの名前を指定します。

#### Windows オペレーティング・システムの場合

- f 強制アプリケーション・フラグを指定します。このフラグを指定すると、このインスタンスを使用しているすべてのアプリケーションが強制的に終了させられます。
- h 使用法情報を表示します。

*InstName*

インスタンスの名前を指定します。

### 例

Linux または UNIX オペレーティング・システム上で、以下のコマンドを発行することによって db2inst1 を作成した場合、

```
/opt/IBM/db2/copy1/instance/db2icrt -u db2fenc1 db2inst1
```

db2inst1 をドロップするには、以下のコマンドを実行する必要があります。

```
/opt/IBM/db2/copy1/instance/db2idrop db2inst1
```

### 使用上の注意

- インスタンスをドロップする前に、すべてのホストで DB2 データベース・マネージャーが停止していること、およびそのインスタンスにアクセスする DB2 データベース・アプリケーションが切断され、終了していることを確認してください。必要なら、そのインスタンスに関連する DB2 データベースをバックアップし、構成データを将来参照できるように保存することができます。



- **db2idrop** コマンドではデータベースは除去されません。データベースがもはや不要であれば、まずそれを除去してください。データベースを除去しない場合、それらはいつでも同じリリースの別の DB2 コピーの下でカタログして、使用を継続することができます。
- DB2 Text Search の構成を保存して、インスタンス・データベースを再利用する予定がある場合、**db2idrop** コマンドを発行する前に、config ディレクトリー (UNIX では `instance_home/sqlllib/db2tss/config`、Windows では `instance_profile_path¥instance_name¥db2tss¥config`) または config ディレクトリー内容を保存するという、追加のステップを実行する必要があります。新しいインスタンスが作成された後に、config ディレクトリーをリストアできます。ただし、config ディレクトリーのリストアが有効になるのは、作成される新しいインスタンスが同じリリースおよびフィックスパック・レベルのものである場合だけです。
- Linux および UNIX オペレーティング・システム上で非 root インストールされたインスタンスをドロップすることはできません。ユーザーがこの DB2 インスタンスを除去するには、**db2\_deinstall -a** を実行して DB2 の非ルート・コピーをアンインストールするオプションのみ使用できます。
- Linux および UNIX オペレーティング・システムの場合、root ユーザーになるために **login** コマンドではなく **su** コマンドを使用している場合は、**su** コマンドを - オプションを指定して発行し、あたかも **login** コマンドを使用してシステムにログインしているかのように、プロセス環境を設定することを指示する必要があります。
- Linux および UNIX オペレーティング・システムの場合、root ユーザー用の DB2 インスタンス環境を入手しないでください。DB2 インスタンス環境を入手している場合に、**db2idrop** を実行することはサポートされていません。
- DB2 pureScale 環境では、**-g** パラメーターは必須です。この場合、すべてのホストでインスタンスがドロップされます。ただし、インストール開始ホスト (IHH) 上の IBM General Parallel File System (GPFS) については削除されず、GPFS ファイル・システムについても同様です。ファイル・システムを手動で削除し、GPFS をアンインストールする必要があります。
- Windows オペレーティング・システムでは、インスタンスが Microsoft Cluster Service (MSCS) でクラスター化されている場合、**db2mcs** または **db2iclus** コマンドを発行してそのインスタンスをクラスター解除してからインスタンスを廃棄できます。
- AIX 6.1 (またはそれ以上) では、このコマンドは、システム・ワークロード・パーティション (WPAR) グローバル環境内の共有 DB2 コピーから実行する場合、root ユーザーとして実行されなければなりません。WPAR は DB2 pureScale 環境ではサポートされていません。

### db2IdentifyType1 - タイプ 1 索引識別ツール

出力ファイルを識別し、指定されたデータベースのタイプ 1 索引をタイプ 2 索引に変換するために使用できる **ALLOW WRITE ACCESS** および **CONVERT** 節を指定した適切な **REORG INDEXES ALL** コマンドでそれを生成します。

このコマンドはユーザー定義索引のみを検査するもので、バージョン 9.7 以降のリリースにまだアップグレードしていないデータベースで実行することを目的としています。

**重要:** **db2IdentifyType1** コマンドは、バージョン 10.1 で非推奨となっており、将来のリリースで除去される可能性があります。タイプ 1 索引はバージョン 9.7 で廃止されました。バージョン 9.5 からバージョン 10.1 にデータベースをアップグレードする前に、タイプ 1 索引を変換するためにこのコマンドを使用することは引き続き可能です。詳しくは、「*DB2 バージョン 10.1 の新機能*」の『*db2IdentifyType1* コマンドが推奨されなくなった』を参照してください。

バージョン 9.7 からのアップグレードの場合は、このコマンドを実行する必要はありません。バージョン 9.7 データベースには、いずれのタイプ 1 索引も含まれていません。デフォルトでは、バージョン 8 以降で作成されたすべての新しい索引はタイプ 2 索引です。ただし、既にタイプ 1 索引がある表に索引を作成した場合や、**DB2\_INDEX\_TYPE2** レジストリー変数を OFF に設定した場合は例外です。また、DB2 Universal Database バージョン 7 以前で作成され、まだ変換されていないデータベースでタイプ 1 の索引を持っている場合もあります。

タイプ 1 索引を使用する表に最初にアクセスする際のパフォーマンスへの影響を最小限にとどめるには、データベースをバージョン 9.7 にアップグレードする前に **db2IdentifyType1** を実行してこうした索引を変換します。データベースのアップグレード時にいくつかの表にタイプ 1 索引が依然として存在する場合、それらの索引には無効というマークが付けられます。アップグレード後、表に初めてアクセスするとき (**indexrec** 構成パラメーターが **RESTART** に設定されている場合は、データベースの再始動時) にそれらの索引はタイプ 2 索引に変換されます。索引が再作成されるまで、表にはアクセスできません。その他の重要なパフォーマンス上の注意、およびパーティション・データベース環境におけるこのコマンドの実行の影響に関する詳細については、『*使用上の注意*』のセクションを参照してください。

このコマンドは、索引変換を実行しません。出力ファイルの生成後、それを確認し、必要であれば内容を編集して、以下のコマンドを使用して出力ファイルを実行できます。

```
db2 -tvf filename
```

以下のディレクトリーから **db2IdentifyType1** コマンドを実行します。

- Linux および UNIX オペレーティング・システムの場合、DB2DIR/bin ディレクトリーから。DB2DIR は、DB2 コピーがインストールされている場所です
- Windows オペレーティング・システムの場合、製品 CD の db2¥Windows¥Utilities ディレクトリーから

## 許可

バージョン 9.7 にアップグレードされていないデータベースについては、以下のいずれかの許可レベルが必要です。

- SYSADM
- DBADM

バージョン 9.7 にアップグレードされたデータベースの場合:

- DBADM

## 必要な接続

データベース。このコマンドは、指定されたデータベースへの接続を自動的に確立します。

## コマンド構文

```

▶▶ db2IdentifyType1 -d database-name -o filename
                                     └-s schema-name┘
└-t table-name┘ └-h┘
  
```

## コマンド・パラメーター

### -d database-name

照会するデータベースの名前を指定します。

### -o filename

REORG INDEX コマンドを書き込むファイルのパスと名前を指定します。

db2IdentifyType1.err ログ・ファイルは、同じパスに作成されます。

db2IdentifyType1.err ログ・ファイルにはトラブルシューティング情報が含まれており、**db2IdentifyType1** コマンドが失敗した場合のみ作成されます。絶対または相対のパス名を使用して、ファイル名を修飾できます。ただし、ファイル名は 246 文字を超過しないでください。同じ名前のファイルが存在する場合は、上書きされます。

### -s schema-name

照会する表のスキーマ (作成者ユーザー ID) を指定します。スキーマを指定しない場合、すべてのスキーマや表が照会されます。一度に複数のスキーマ名を指定することはできません。スキーマ名 ID はシステム・カタログ表に表示されるように、大文字を使用して指定する必要があります。区切りスキーマ名 ID は二重引用符で囲む必要があります。

### -t table-name

照会する表の名前を指定します。表名を指定しない場合、すべての表が照会されます。一度に複数の表名を指定することはできません。表名 ID はシステム・カタログ表に表示されるように、大文字を使用して指定する必要があります。区切り表名 ID は二重引用符で囲む必要があります。

### -h

ヘルプ情報を表示します。このオプションを指定した場合、他のすべてのオプションは無視されます。

### 使用上の注意

**CONVERT** 節がバージョン 10.1 で廃止されたので、生成された出力ファイルはバージョン 9.5 クライアントから実行する必要があります。

データベース内にタイプ 1 索引がある場合や、データベースにタイプ 1 索引が含まれているかどうか不確かな場合を除いては、このツールを使用する必要はありません。タイプ 1 索引が多数含まれていることが分かっている場合、**REORG INDEXES** コマンドを使用して特定の表を識別するのではなく、このコマンドに **CONVERT** オプションを指定して、データベース内のすべての表か、タイプ 1 索引のある特定の表に対して実行することを選択できます。索引が既にタイプ 2 の場合には、**REORG INDEXES** に **CONVERT** オプションを指定しても効果はありません。

データベースに多数の表がある場合、**db2IdentifyType1** コマンドが処理を完了するには長い時間がかかる可能性があるので、**-s** オプションか **-t** オプションを使用して、一度に特定のサブセットだけをスキャンしてください。

パーティション・データベース環境におけるパフォーマンスと使用方法に関する考慮事項: パーティション・データベース環境では、表が存在するデータベース・パーティションのすべてに必ずしもタイプ 1 索引があるとは限りません。データベース・パーティションのサブセットでのみ索引がタイプ 2 索引に変換された場合には、このような状況が生じ得ます。また、すべてのデータベースパーティションを含んでいないデータベース・パーティション・グループに表が存在している場合もあります。バージョン 9.5 パーティション・データベース環境では、タイプ 1 索引がすべてのデータベース・パーティションにあるかどうかに関係なく、**db2IdentifyType1** コマンドを 1 つのデータベース・パーティション上で実行すれば、すべてのタイプ 1 索引が検出されます。最善のパフォーマンスを得るには、カタログ・データベース・パーティション上で実行してください。データベースにパーティションが多数ある場合、このコマンドを完了するには長い時間がかかる場合があります。各パーティションからデータが収集されるためです。

## db2ilist - インスタンスのリスト

**db2ilist** コマンドを実行しているのと同じ DB2 コピー・ロケーションから、**db2icrt** コマンドを使用することによって作成されるすべてのインスタンスのリストを表示します。

Linux および UNIX オペレーティング・システムでは、このユーティリティーは *DB2DIR/instance* ディレクトリーにあります (*DB2DIR* は DB2 コピーのインストール先インスタンス・ディレクトリー)。Windows オペレーティング・システムでは、このユーティリティーは *DB2PATH\bin* ディレクトリーにあります (*DB2PATH* は DB2 データベース・システムの現行バージョンがインストールされているインストール・ロケーション)。

### 許可

なし

### コマンド構文

```
▶▶ db2ilist [ -h ] ▶▶
```

### コマンド・パラメーター

**-h** 使用法情報を表示します。

### 使用上の注意

- Linux および UNIX オペレーティング・システムの場合、root ユーザーになるために **login** コマンドではなく **su** コマンドを使用している場合は、**su** コマンドを **-** オプションを指定して発行し、あたかも **login** コマンドを使用してシステムにログインしているかのように、プロセス環境を設定することを指示する必要があります。
- Linux および UNIX オペレーティング・システムの場合、root ユーザー用の DB2 インスタンス環境を入手しないでください。DB2 インスタンス環境を入手している場合に、**db2ilist** を実行することはサポートされていません。
- AIX 6.1 (またはそれ以上) では、このコマンドは、システム・ワークロード・パーティション (WPAR) グローバル環境内の共有 DB2 コピーから実行する場合、root ユーザーとして実行されなければなりません。WPAR は DB2 pureScale 環境ではサポートされていません。

## db2inidb - ミラーリングされたデータベースの初期化

スプリット・ミラー環境のミラーリングされたデータベースを初期化します。ミラーリングされたデータベースは、ロールフォワード・ペンディング状態にある 1 次データベースの複製として初期化したり、1 次データベースをリストアするためのバックアップ・イメージとして使用できます。

スプリット・ミラー・データベースを使用できるようにするには、まずこのコマンドを発行する必要があります。

### 許可

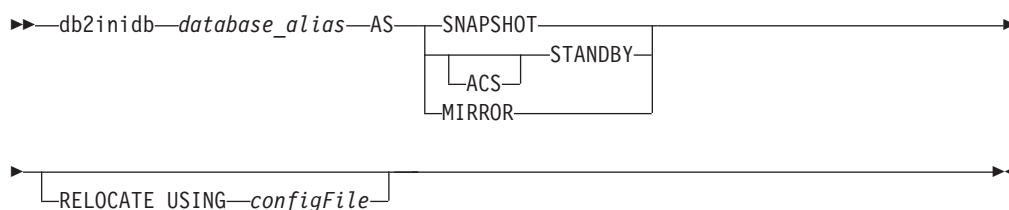
以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMAINT

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

*database\_alias*

初期設定するデータベースの別名を指定します。

#### SNAPSHOT

ミラーリングされたデータベースは、1 次データベースの複製として初期化されることを指定します。

#### STANDBY

データベースをロールフォワード・ペンディング状態にすることを指定します。1 次データベースから新しいログ・ファイルをフェッチ、スタンバイ・データベースに適用することが可能です。スタンバイ・データベースは、1 次データベースがダウンした場合に、その代わりに使用できます。

#### ACS

**db2inidb** コマンドを、データベースの ACS スナップショット・コピーに対して使用して **STANDBY** アクションを実行するように指定します。このオプションが必要となる理由は、**db2inidb** コマンドを発行できる対象が **SET WRITE SUSPEND | RESUME** コマンドによって作成されたスプリット・ミラー・データベース・スナップショットに限定されているためです。



また、**ACS STANDBY** オプションを使用すると ACS スナップショット・コピーが開始されてロールフォワード・ペンディング状態になるので、**DB2 BACKUP** コマンドをそのスナップショット・イメージに対して正常に発行できます。このようにしないと、スナップショット・イメージに接続しようとするデータベースのコピーが **RESTORE\_PENDING** 状態になり、将来リカバリーする際にバックアップ・コピーとして使用できなくなってしまいます。

このフィーチャーは、ACS スナップショットに基づいて DB2 オフロード・バックアップを作成するために、IBM Tivoli Storage FlashCopy® Manager などのストレージ・マネージャーと連携するという目的で特に導入されました。その他の目的のためにこのオプションを使用し、ACS スナップショット・コピーの内容をマウントしたり変更したりすると、DB2 バックアップを作成する場合であっても、将来に予期しない動作を引き起こす恐れがあります。

**MIRROR** ミラー・データベースを、1 次データベースのリストアに使用できるバックアップ・イメージにすることを指定します。

### **RELOCATE USING** *configFile*

データベースがスナップショット、スタンバイ、またはミラーとして初期化される前に、*configFile* にリストされた情報に基づいてデータベース・ファイルを再配置することを指定します。*configFile* の形式については、1277 ページの『db2relocatedb - データベースの再配置』を参照してください。

## 使用上の注意

**db2inidb***database\_alias* **as mirror** コマンドを発行する前に **db2 connect todatabase-alias** 操作を発行しないでください。初期化する前にスプリット・ミラー・データベースに接続すると、ロールフォワード・リカバリーに必要なログ・ファイルが消去されてしまいます。その接続によって、データベースは、以前にそのデータベースを中断した時点の状態に戻ります。中断の時点でデータベースが整合とマーク付けされると、DB2 データベース・システムはクラッシュ・リカバリーの必要はないと判断して、将来の利用のためにログを空にします。ログが空になった場合、ロールフォワードしようとする、SQL4970N エラー・メッセージが戻されます。

パーティション・データベース環境では、すべてのデータベース・パーティションに対して **db2inidb** コマンドを発行する必要があります。その後、任意のデータベース・パーティションのスプリット・ミラーを使用できるようになります。

**db2\_a11** コマンドを使用すれば、すべてのデータベース・パーティションに対して同時に **db2inidb** を実行することができます。

しかし、**RELOCATE USING** オプションを使用する場合は、**db2\_a11** コマンドを使用して全パーティションに対して同時に **db2inidb** を実行することはできません。パーティションごとにそれぞれ別個の構成ファイル (変更対象のデータベース・パーティションの **NODENUM** 値が含まれる) を用意する必要があります。例えば、データベースの名前を変更する場合は、すべてのデータベース・パーティションが影響を受けることになり、各データベース・パーティションごとに別個の構成ファイルを用意して **db2relocatedb** コマンドを実行する必要があります。単一データベー

## db2inidb - ミラーリングされたデータベースの初期化

ス・パーティションに属するコンテナを移動する場合は、そのデータベース・パーティションに対して一度だけ **db2relocatedb** コマンドを実行することが必要です。

**RELOCATE USING** *configFile* パラメーターが指定されており、データベースの再配置が正常に実行されたなら、指定された *configFile* はデータベース・ディレクトリーにコピーされ、その名前が *db2path.cfg* に変更されます。それ以降のクラッシュ・リカバリーまたはロールフォワード・リカバリーにおいて、このファイルは、ログ・ファイルの処理時にコンテナ・パスの名前を変更するために使用されます。

クローン・データベースを初期化している場合、指定された *configFile* は、クラッシュ・リカバリー完了後にデータベース・ディレクトリーから自動的に除去されます。

スタンバイ・データベースまたはミラーリングされたデータベースを初期化している場合、指定された *configFile* ファイルは、ロールフォワード・リカバリーの完了後またはキャンセル後に、データベース・ディレクトリーから自動的に除去されます。**db2inidb** 実行後には、新しいコンテナ・パスを *db2path.cfg* ファイルに追加できます。元のデータベースに対して **CREATE** 操作または **ALTER TABLESPACE** 操作を実行し、スタンバイ・データベース上で異なるパスを使用しなければならぬ場合には、このことが必要になります。

**HADR** プライマリーまたはスタンバイから取得されたスプリット・ミラー・データベースの初期化を実行する場合、以下のいずれかに該当するのであれば、**STANDBY** パラメーターを使用してください。

- 新しいデータベースは **HADR** ペアにして使用する予定であり、その新しいペアの **HADR** 構成設定は元のペアの設定とは同じではない。
- データベースは、スタンドアロン・データベースとして初期設定する予定である。

**DB2 pureScale** 環境では、いずれかのメンバーから **db2inidb** コマンドを発行でき、コマンドを 1 回だけ発行する必要があります。



## db2inspf - 検査結果のフォーマット

**INSPECT CHECK** の結果のデータを、ASCII フォーマットにします。このユーティリティを使用して、検査の詳細を表示します。

**db2inspf** ユーティリティによるフォーマットは、表または表スペースに対するものとして行うことができ、エラー、警告、およびサマリーを単独で、または任意の組み合わせで指定できます。

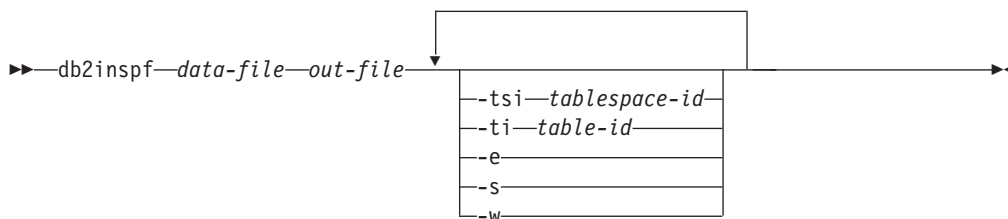
### 許可

このユーティリティにはすべてのユーザーがアクセス可能ですが、結果ファイルに対してこのユーティリティを実行するには、それらの読み取り権限がなければなりません。

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

*data-file*

フォーマットを必要とする不定形式の検査結果ファイル。

*out-file* フォーマット済み出力の出力ファイル。

**-tsi** *table-space-id*

表スペース ID。この表スペースの表のみをフォーマットします。

**-ti** *table-id*

表 ID。この ID を持つ表のみをフォーマットします。表スペース ID も指定する必要があります。

**-e** エラーのみをフォーマットします。

**-s** サマリーのみ。

**-w** 警告のみ。

### 例

すべてのエラー、警告、およびサマリーをデータ・ファイル `tbschk.log` からフォーマットするには、以下の照会を実行します。

```
db2inspf tbschk.log tbschk_esw.txt -e -s -w
```

## db2iprun - インストール・イメージ・サイズの削減

**db2iprun** コマンドは、DB2 製品のインストール・イメージのサイズを、インストールの前に削減できます。

このツールは、DB2 データベース製品の大規模デプロイメント、およびアプリケーション内への DB2 の組み込みに役立ちます。入力ファイルである `.prn` ファイルは、除去可能な製品、コンポーネント、および言語の完全なリストを含んでおり、これを使用してインストール・イメージから除去するものを指定できます。

**db2iprun** コマンドは入力ファイルを呼び出し、それらのコンポーネントおよび言語に関連したファイルを除去します。その結果、通常の DB2 インストール方式でインストール可能な、新しくより小さい DB2 インストール・イメージが生成されます。

すべての製品を整理できるわけではありません。少なくとも、1 つの製品は依然として、生成されるイメージの一部でなければなりません。

### 許可

なし

### 必要な接続

なし

### コマンド構文

```

▶▶ db2iprun -r input_file_path -p root_directory_path
▶ -o destination_directory_path -c -t trace_file
▶ -l log_filename -h -?

```

### コマンド・パラメーター

#### -r *input\_file\_path*

使用される入力ファイルへの絶対パスを指定します。入力ファイルである `.prn` ファイルは、除去可能なコンポーネントの完全なリストを含んでいて、インストール・イメージから除去する製品、コンポーネント、および言語を指定するために使用できます。

#### -p *root\_directory\_path*

(Windows オペレーティング・システムの場合のみ。) ソース・インストール・イメージのルート・ディレクトリーへの絶対パスを指定します。このディレクトリーは **setup** を含んでいて、DB2 インストール DVD のルート・ディレクトリーとなります。

#### -o *destination\_directory\_path*

整理された新しい DB2 イメージのコピー先を絶対パスで指定します。このディレクトリーに対する書き込み権限があることを確認してください。

**-c** ソース・インストール・イメージを直接整理することを指定します。ソース・インストール・イメージ・ディレクトリーが書き込み可能であることを確認してください。

**-l** *log\_filename*

エラー・ロギングを有効にします。Linux および UNIX オペレーティング・システムでは、**-l** パラメーターが指定されていない場合、デフォルトのログ・ファイル名は `tmpdir/db2iprun_username.log` です。Windows オペレーティング・システムでは、ログ・ファイル `db2iprun.log` が宛先ディレクトリーに書き込まれます。

**-t** *trace\_file*

(Linux および UNIX オペレーティング・システムの場合のみ。) デバッグ・モードをオンにします。デバッグ情報は、名前を指定したファイルに書き込まれます。

**-? | -h** 使用情報を表示します。

### 例

Windows オペレーティング・システムでは、入力ファイルが `c:%db2client.prn` にあり、DB2 **setup.exe** ファイルが `d:¥` にあり、整理された IBM Data Server Client イメージを `e:%compact_client` ディレクトリーにコピーする場合、IBM Data Server Client イメージを整理するには、コマンド・プロンプトに以下のコマンドを入力します。

```
db2iprun.exe -r c:%db2client.prn -p d:¥ -o e:%compact_client
```

Linux および UNIX オペレーティング・システムでは、入力ファイルが `/tmp/db2client.prn` にあり、DB2 ソース・ファイルが `/mnt/cdrom` にあり、整理された IBM Data Server Client イメージを `/compact_client` ディレクトリーにコピーする場合、IBM Data Server Client イメージを整理するには、コマンド・プロンプトに以下のコマンドを入力します。

```
db2iprun -r /tmp/db2client.prn /mnt/cdrom -o /compact_client
```

### 使用上の注意

**db2iprun** コマンドおよびサンプルの入力ファイルは、インストール DVD で提供されます。

#### Windows オペレーティング・システムの場合

```
dvd_drive:%db2%Windows%utilities%db2iprun
```

#### Linux および UNIX オペレーティング・システムの場合

```
product image/db2/platform/utilities/db2iprun
```

**db2iprun** コマンドは DB2 pureScale 環境ではサポートされていません。

## db2isetaup - インスタンス作成インターフェースの開始

DB2 インスタンス・セットアップ・ウィザードを開始します。これは、インスタンスを作成、構成、および拡張するためのグラフィカル・ユーザー・インターフェース (GUI) ツールです。

### 許可

root インストールの場合、コマンドが発行されるシステムでの root ユーザー権限が必要です。非 root インストールの場合、非 root インストールを所有するユーザー ID でログオンしなければなりません。

コマンドが発行されるシステムでの root ユーザー権限が必要です。

### 必要な接続

なし

### コマンド構文

```
▶▶ db2isetaup [-i language-code] [-l logfile] [-t tracefile]
▶ [-r response_file] [--?] [-h]
```

### コマンド・パラメーター

#### -i language-code

インストールを実行する際の希望する言語の 2 文字から成るコード。指定されないと、このパラメーターは現行ユーザーのロケールにデフォルト設定されます。

#### -l logfile

名前を指定したファイルにログを書き込みます。root インストールの場合、パスとファイル名はデフォルトの /tmp/db2isetaup.log になります。非 root インストールの場合、デフォルトのログ・ファイルは /tmp/db2isetaup\_userID.log です (userID は非 root インストールを所有するユーザー ID)。

#### -t tracefile

tracefile に指定するトレース・ファイルの絶対パスおよび名前。

#### -r response\_file

使用される応答ファイルの絶対パスとファイル名。GUI インターフェースの使用を計画している場合には、このパラメーターを指定しないでください。

#### -? | -h

出力使用情報。

## 使用上の注意

1. DB2 インスタンス・セットアップ・ウィザードは、DB2 セットアップ・ウィザードが提供する機能のサブセットを提供します。DB2 セットアップ・ウィザード (インストール・メディアから実行する) によって、DB2 データベース・コンポーネントのインストール、DAS 作成や構成などのシステム・セットアップ・タスクの実行、およびインスタンスのセットアップ、構成、拡張を行うことができます。DB2 インスタンス・セットアップ・ウィザードは DB2 セットアップ・ウィザードと同じ機能を提供しますが、DB2 データベース・コンポーネントのインストールはできません。
2. IBM DB2 pureScale Featureの必須コンポーネントがインスタンスの作成または拡張の際にインストールされていない場合には、DB2 インスタンス・セットアップ・ウィザードは、それらの必須コンポーネントすべてをいずれかのリモート・ホストにインストールします。
3. このコマンドの実行可能ファイルは、*DB2DIR/instance* ディレクトリにあります。これは標準インストールでは使用できませんが、コンパクト・インストールでは使用できません。
4. **db2isetup** は、サポートされているすべての Linux および UNIX のオペレーティング・システムで実行できます。
5. AIX 6.1 (またはそれ以上) では、このコマンドは、システム・ワークロード・パーティション (WPAR) グローバル環境内の共有 DB2 コピーから実行する場合、root ユーザーとして実行されなければなりません。
6. DB2 pureScale 環境では、
  - a. メンバーまたは クラスター・キャッシング・ファシリティ (CF) を追加またはドロップする前に、DB2 pureScale インスタンスをすべてのホスト上で完全に停止しておく必要があります。
  - b. サイレント・モードでは、DB2 pureScale インスタンスの既存のホストが、以下のケースで使用可能です (これらは DB2 セットアップ・ウィザードには当てはまりません)。
    - 2 番目の CF を DB2 pureScale インスタンスに追加するには、既存のメンバーの 1 つを更新して CF およびメンバー・ホストの両方に入れるか、または新しいホストを CF として追加します。
    - さらに別のメンバーを DB2 pureScale インスタンスに追加するには、既存の CF を更新して CF およびメンバー・ホストの両方に入れるか、または新しいホストをメンバーとして追加します。
    - **db2isetup** コマンドを使用してメンバーまたは CF をドロップすることはできません。
  - c. 常に少なくとも 1 つのアクティブな CF ホストが DB2 pureScale インスタンス内になければなりません。

## db2iupdt - インスタンスの更新

**db2iupdt** コマンドを使用して、リリース内のさらに高いレベルにインスタンスを更新したり、DB2 pureScale インスタンス以外のインスタンスを DB2 pureScale インスタンスへ更新したり、DB2 pureScale インスタンスのスケールを変えたりできます。

**db2iupdt** コマンドを実行する前に、インスタンスを停止し、そのインスタンスについて実行されているすべてのプロセスを停止する必要があります。このコマンドを使用することにより DB2 pureScale インスタンス以外のインスタンスを更新してメンバーまたはクラスター・キャッシング・ファシリティを追加する場合は、前提条件とプリインストール・チェックリストを検討して、インスタンスおよびホストが準備していることを確認してください。

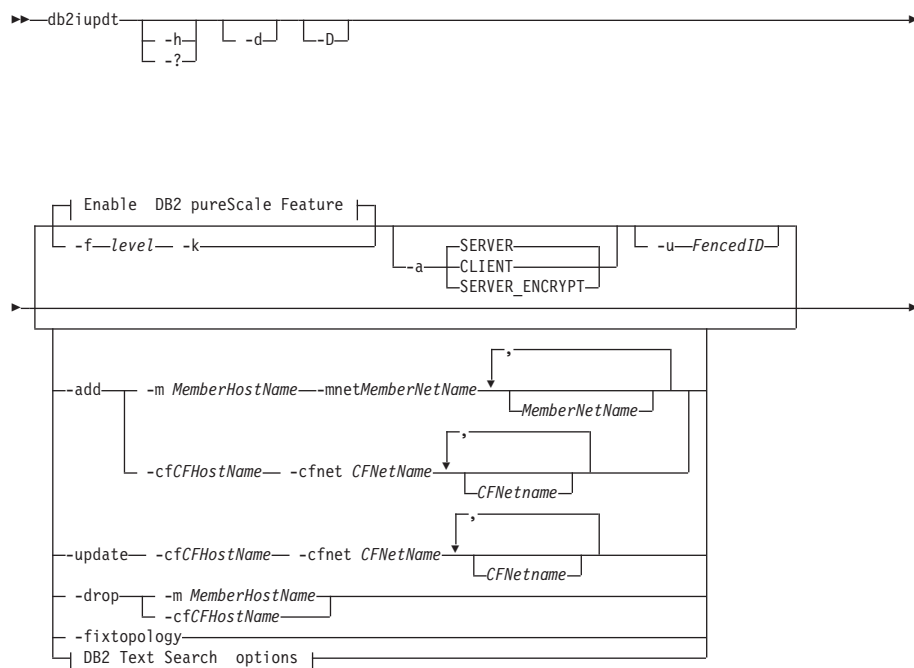
注: DB2 pureScale インスタンスでは、構成クォーラム に達していないと、つまり絶対多数のノードがオンラインになっていないとリソース・モデルに変更を加えることができません。2つのホストで構成されるセットアップの場合、いずれかがオフラインになっていると **db2iupdt** コマンドを使用できません。

### 許可

UNIX および Linux オペレーティング・システムでは root ユーザー権限、Windows オペレーティング・システムではローカル管理者。

### コマンド構文

UNIX および Linux オペレーティング・システムの場合



▶—InstName—

### Enable DB2 pureScale Feature:

```

-m MemberHostName -mnetMemberNetName MemberNetName
-cfCFHostName -cfnet CFNetName CFNetname
-instance_shared_dirShared_Directory_for_Instance
-instance_shared_devShared_Device_Path_for_Instance instance_shared_mount—Shared_Mounting_Dir
-i—db2sshidName

```

### DB2 Text Search options:

```

-j—"TEXT_SEARCH"
    ,ServiceName
    ,ServiceName,PortNumber
    ,PortNumber

```

### Windows オペレーティング・システムの場合

```

db2iupdt—InstName—/u:—username,password
/p:—instance-profile-path /r:—baseport,endport
/h:—hostname /s /q /a:—authType
DB2 Text Search options /?

```

### DB2 Text Search options:

```

-j—"TEXT_SEARCH"
    ,ServiceName
    ,ServiceName,PortNumber
    ,PortNumber

```

### コマンド・パラメーター

#### UNIX および Linux オペレーティング・システムの場合

-h | -?

使用情報を表示します。

## db2iupdt - インスタンスの更新

### **-a AuthType**

インスタンスの認証タイプ (SERVER、SERVER\_ENCRYPT、または CLIENT) を指定します。デフォルトは SERVER です。

### **-d** デバッグ・モードをオンにします。

### **-k** 更新時に現行のインスタンス・タイプが変更されないようにします。

### **-D** このパラメーターは推奨されなくなりました。今後のリリースで除去される可能性があります。このパラメーターは **-f level** パラメーターに置き換えられます。あるパスにインストールされた高いコード・レベルのインスタンスから別のパスにインストールされた低いコード・レベルのインスタンスに更新します。

### **-f level**

互換性のために、高い DB2 バージョンのインスタンス・タイプから、低い DB2 バージョンのインスタンス・タイプへ、インスタンスを更新します。

### **-add -m MemberHostName:NetName1 | -cf CFHostName:NetName2**

DB2 pureScale インスタンスに追加するホスト (メンバーまたはクラスター・キャッシング・ファシリティー) を指定します。

追加するホストのタイプを指定するには、メンバーの場合は **-m** を、クラスター・キャッシング・ファシリティーの場合は **-cf** を使用します。両方ではなく、一方を選択する必要があります。

**db2iupdt -add** コマンドは、既に DB2 pureScale インスタンスの一部になっているホストから実行する必要があります。

**-m MemberHostName:NetName1** および **-cf CFHostName:NetName2** 構文は推奨されなくなりました。今後のリリースでは除去される可能性があります。代わりに、構文 **-add -m MemberHostName -mnet MemberNetName** または **-add -cf CFHostName -cfnet CFNetName** を使用してください。

### **-add -m MemberHostName -mnet MemberNetName | -add -cf CFHostName**

### **-cfnet CFNetName**

クラスター相互接続ネット名を指定します。これは、DB2 メンバーとクラスター・キャッシング・ファシリティーとの高速通信に使用される相互接続のホスト名です。追加する各ホストのネット名は、インスタンス内のすべての既存のメンバーまたは CF と同じサブネット上にある必要があります。

追加するホストのタイプを指定するには、メンバーの場合は **-m MemberHostName -mnet MemberNetName** を、クラスター・キャッシング・ファシリティーの場合は **-cf CFHostName -cfnet CFNetName** を使用します。両方ではなく、一方を選択する必要があります。

**db2iupdt -add** コマンドは、既に DB2 pureScale インスタンスの一部になっているホストから実行する必要があります。

### **-add** DB2 pureScale Feature インスタンスに追加するホストのホスト名およびクラスター相互接続ネット名を指定します。 **db2iupdt -add** コ



マンドは、既に DB2 pureScale インスタンスの一部になっているホストから実行する必要があります。

**-m MemberHostName -mnet MemberNetName**

クラスター相互接続ネット名 *MemberNetName* を指定して、ホスト名 *MemberHostName* のホストを、DB2 pureScale Feature インスタンスに追加します。*MemberHostName* に複数のクラスター相互接続ネットワーク・アダプター・ポートがある場合は、各クラスター相互接続ネット名をコンマで区切ったリストを *MemberHostName* に指定することができます。

**-cf CFHostName -cfnet CFNetName**

クラスター相互接続ネット名 *CFNetName* を指定して、ホスト名 *CFHostName* のホストを、DB2 pureScale Feature インスタンスにクラスター・キャッシング・ファシリティーとして追加します。*CFHostName* に複数のクラスター相互接続ネットワーク・アダプター・ポートがある場合は、各クラスター相互接続ネット名をコンマで区切ったリストを *CFNetName* に指定することもできます。

**-drop -m MemberHostName | -cf CFHostName**

DB2 pureScale インスタンスからドロップするホスト (メンバーまたはクラスター・キャッシング・ファシリティー) を指定します。ドロップするホストのタイプを指定するには、メンバーの場合は **-m** オプションを、クラスター・キャッシング・ファシリティーの場合は **-cf** オプションを使用します。両方ではなく、一方を選択する必要があります。このパラメーターを使用して、最後のメンバーおよび最後の CF を DB2 pureScale インスタンスからドロップすることはできません。このパラメーターは、**-add** パラメーターと組み合わせて使用しないでください。

**-instance\_shared\_dev Shared\_Device\_Path\_for\_Instance**

インスタンス共有ファイルとデフォルトのデータベース・パスを保持する DB2 pureScale インスタンスをセットアップするのに必要な共有ディスク装置パスを指定します。例えば、装置パス *dev/hdisk1* を指定します。DB2 pureScale インスタンスについて、共有ディレクトリーは、すべてのホストでアクセス可能でなければなりません。このパラメーターの値は、**-tbdev** パラメーターと同じ値であってはなりません。このパラメーターと **-instance\_shared\_dir** は、相互に排他的です。

DB2 pureScale インスタンス以外のインスタンスを DB2 pureScale インスタンスに更新する場合にのみ、このパラメーターが必要です。

**-instance\_shared\_mount Shared\_Mounting\_Dir**

新規 IBM General Parallel File System (GPFS) ファイル・システム用のマウント・ポイントを指定します。指定するパスは、新規で空のパスでなければならず、既存の GPFS ファイル・システムの内部にネストされたパスであってはなりません。

### **-instance\_shared\_dir** *Shared\_Directory\_for\_Instance*

インスタンス共有ファイルとデフォルトのデータベース・パスを保持する DB2 pureScale インスタンスをセットアップするのに必要な、共有ファイル・システム (GPFS) 内のディレクトリーを指定します。例えば、/sharedfs となります。DB2 pureScale インスタンスについて、そのディスクはすべてのホストでアクセス可能でなければなりません。このパラメーターの値は、**-tbdev** パラメーターと同じ値であってはなりません。このパラメーターと **-instance\_shared\_dev** は、相互に排他的です。

DB2 pureScale インスタンス以外のインスタンスを DB2 pureScale インスタンスに更新する場合にのみ、このパラメーターが必要です。

### **-tbdev** *Shared\_device\_for\_tiebreaker*

データの保全性を維持するために、DB2 pureScale 環境でタイブレーカーとして機能する共有装置パスを指定します。このパラメーターの値は、**-instance\_shared\_dev** パラメーターや

**-instance\_shared\_dir** パラメーターと同じ値であってはなりません。このパラメーターが必要になるのは、DB2 クラスター・サービスのタイブレーカーが初めて作成される時や、DB2 pureScale インスタンス以外のインスタンスを DB2 pureScale インスタンスに更新する場合です。DB2 クラスター・サービスのピア・ドメインが存在する場合は、このパラメーターは無効です。

### **-i** *db2sshidName*

ホスト間でセキュア・シェル (SSH) ネットワーク・プロトコルを使用するために必要な非 root ユーザー ID を指定します。指定されたユーザー ID は、特別な特権を持たないユーザーである必要があります。DB2 管理 GPFS ファイル・システムに対してのみ有効です。

### **-j** "TEXT\_SEARCH"

DB2 Text Search サーバーを、生成されたサービス名と TCP/IP ポート番号のデフォルト値を使用して構成します。インスタンス・タイプがクライアントまたは dsf の場合、このパラメーターは使用できません。

#### **-j** "TEXT\_SEARCH, servicename"

DB2 Text Search サーバーを、提供されたサービス名と自動的に生成されたポート番号を使用して構成します。サービス名に services ファイル内で割り当てられたポート番号がある場合、それはその割り当てられたポート番号を使用します。

#### **-j** "TEXT\_SEARCH, servicename, portnumber"

DB2 Text Search サーバーを、提供されたサービス名とポート番号を使用して構成します。

#### **-j** "TEXT\_SEARCH, portnumber"

DB2 Text Search サーバーを、デフォルトのサービス名と提供されたポート番号を使用して構成します。有効なポート番号は 1024 - 65535 の範囲になければなりません。

**-u** *Fenced ID*

*fenced* ユーザー定義関数および *fenced* ストアード・プロシージャを実行するユーザー ID の名前を指定します。このパラメーターが必要になるのは、インスタンスのタイプをクライアント・インスタンスからクライアント以外のインスタンスに変換する場合だけです。現在のインスタンス・タイプを判別するには、**GET DBM CFG** コマンドからの出力のノード・タイプ・パラメーターを参照してください。インスタンスがすでにクライアント以外のインスタンスである場合や、インスタンスがクライアント・インスタンスであり、なおかつクライアント・インスタンスとして保たれる場合 (例えば、**-k** パラメーターを使用する場合) は、**-u** パラメーターは必要ありません。**-u** パラメーターによって、既存のインスタンスの *fenced* ユーザーを変更できます。

**-fixtopology**

失敗した追加操作またはドロップ操作を手動で修正する場合に使用します。追加操作の場合、このパラメーターはすべての変更をロールバックして、以前のトポロジーに戻します。ドロップ操作の場合、このパラメーターはドロップ操作を完了させます。このパラメーターは、**-d** を除く他のパラメーターと組み合わせて使用することはできません。

*InstName*

インスタンスの名前を指定します。

**Windows オペレーティング・システムの場合**

*InstName*

インスタンスの名前を指定します。

**/u:username,password**

DB2 サービスのアカウント名およびパスワードを指定します。

**/p:instance-profile-path**

更新されたインスタンス用の新しいインスタンス・プロファイル・パスを指定します。

**/r:baseport,endport**

MPP モードで実行する場合に、パーティション・データベース・インスタンスによって使用される TCP/IP ポートの範囲を指定します。このオプションを指定した場合、ローカル・マシンのサービス・ファイルは、以下の項目で更新されます。

```
DB2_InstName      baseport/tcp
DB2_InstName_END  endport/tcp
```

**/h:hostname**

現行のマシンに対して複数のデフォルトの TCP/IP ホスト名がある場合、それらをオーバーライドします。

**/s** インスタンスを更新してパーティション・インスタンスにします。

**/q db2iupdt** コマンドを静止モードで発行します。

**/a:authType**

インスタンスの認証タイプ *authType* (SERVER、CLIENT、または SERVER\_ENCRYPT) を指定します。

**/j "TEXT\_SEARCH"**

DB2 Text Search サーバーを、生成されたサービス名と TCP/IP ポート番号のデフォルト値を使用して構成します。インスタンス・タイプがクライアントの場合、このパラメーターは使用できません。

**/j "TEXT\_SEARCH, servicename"**

DB2 Text Search サーバーを、提供されたサービス名と自動的に生成されたポート番号を使用して構成します。サービス名に *services* ファイル内で割り当てられたポート番号がある場合、それはその割り当てられたポート番号を使用します。

**/j "TEXT\_SEARCH, servicename, portnumber"**

DB2 Text Search サーバーを、提供されたサービス名とポート番号を使用して構成します。

**/j "TEXT\_SEARCH, portnumber"**

DB2 Text Search サーバーを、デフォルトのサービス名と提供されたポート番号を使用して構成します。有効なポート番号は 1024 - 65535 の範囲になければなりません。

**/?** **db2iupdt** コマンドの使用方法を表示します。

### 例

#### UNIX および Linux オペレーティング・システムの場合

インスタンス *db2inst2* は、*DB2DIR1* にインストールされている DB2 データベース製品の DB2 コピーに関連しています。同じコンピューター上の *DB2DIR2* に、*DB2DIR1* にインストールされている DB2 データベース製品と同じバージョンの DB2 データベース製品の別のコピーがあります。インスタンスを更新して、*DB2DIR1* にインストールされている DB2 コピーから、*DB2DIR2* にインストールされている DB2 コピーに対して実行するには、以下のコマンドを発行します。

```
DB2DIR2/instance/db2iupdt db2inst2
```

*DB2DIR2* にインストールされている DB2 コピーが *DB2DIR1* でインストールされている DB2 コピーより低いレベルのものであるなら、発行するコマンドは次のようになります。

```
DB2DIR2/instance/db2iupdt -D db2inst2
```

#### リリース内のさらに高いレベルに、インスタンスを更新する

DB2 インスタンスをさらに高いレベルに更新する場合や、1 つの DB2 インストール・パスから別のインストール・パスに更新する場合には、次のようなコマンドを入力します。

```
DB2DIR/instance/db2iupdt db2inst1
```

ここで、*DB2DIR* は、DB2 のコピーのインストール・ロケーションを示します。このコマンドが DB2 pureScale Feature コピーから実行される場合、既存の *db2inst1* のインスタンス・タイプは *dsf* でなければなりません。*db2inst1* インスタンスが DB2 pureScale インスタンスである場合、この例により、それを 1 つのレベルから、DB2 pureScale Feature を持つ DB2 ESE の異なるレベルに更新することができます。この例は、*ese* タイプのインスタンスを DB2 pureScale インスタンスに更新することには適用されません。次の例は、この手順を略述しています。

### DB2 pureScale インスタンス以外のインスタンスを DB2 pureScale インスタンスに更新する

あるインスタンスを DB2 pureScale インスタンスに更新するには、以下のようになります。

```
DB2DIR/instance/db2iupdt
-cf host2
-cfnet host2-ib0
-m host1
-mnet host1-ib0
-instance_shared_dev /dev/hdisk1
-tbdev /dev/hdisk2
-u db2fenc1
db2inst1
```

ここで、*DB2DIR* は、DB2 のコピーのインストール・ロケーションを示します。

さらにこのコマンドは */dev/hdisk1* を使用して、インスタンス共有ファイルを保管するための共有ファイル・システムを作成し、タイプレーカーとして機能する共有装置パスとして */dev/hdisk2* をセットアップします。

**-tbdev** パラメーターの値は、**-instance\_shared\_dev** パラメーターの値とは異なっていなければなりません。

### DB2 pureScale インスタンスの拡張と縮小 (db2iupdt -add または db2iupdt -drop を使用)

以下の例は DB2 pureScale 環境に適用されます。

- DB2 pureScale インスタンスを更新して、メンバーを追加します。*host1* というメンバーを、ネット名 *host1-ib0* で、*db2sdin1* という DB2 pureScale インスタンスに追加するには、次のようなコマンドを入力します。

```
DB2DIR/instance/db2iupdt -d -add -m host1 -mnet host1-ib0 db2sdin1
```

ここで、*DB2DIR* は、DB2 のコピーのインストール・ロケーションを示します。

- DB2 pureScale インスタンスを更新して、第 2 のクラスター・キャッシング・ファシリティを追加します。*host2* というクラスター・キャッシング・ファシリティを、ネット名 *host2-ib0* で、*db2sdin1* という DB2 pureScale インスタンスに追加するには、次のようなコマンドを入力します。

```
DB2DIR/instance/db2iupdt -d -add -cf host2 -cfnet host2-ib0 db2sdin1
```

ここで、*DB2DIR* は、DB2 のコピーのインストール・ロケーションを示します。

- DB2 pureScale インスタンスから、メンバーをドロップします。 host1 というメンバーを、db2sdin1 という DB2 pureScale インスタンスからドロップするには、次のようなコマンドを入力します。

```
DB2DIR/instance/db2iupdt -d -drop -m host1 db2sdin1
```

ここで、*DB2DIR* は、DB2 のコピーのインストール・ロケーションを示します。 host1 が同じインスタンス内で CF ロールを持っていない場合には、host1 以外のホストからコマンドを実行する必要があります。

### 使用上の注意

#### サポートされているすべてのオペレーティング・システムの場合

- **db2iupdt** コマンドを使用して、DB2 データベース製品の同じバージョンの、ある DB2 コピーから別の DB2 コピーに DB2 インスタンスを更新する場合、旧 DB2 コピーのインストール・パス内で定義されている DB2 グローバル・プロファイル変数は、新しいインストール・ロケーションに更新されません。インスタンス固有の DB2 インスタンス・プロファイル変数は、インスタンスの更新後に実行されます。

#### UNIX および Linux オペレーティング・システムの場合

- DB2DB2 Enterprise Server Edition のみ、**db2iupdt** コマンドを使用して更新できます。
- メンバーの追加やドロップなど、メンバー・トポロジーを変更する場合、データベースにアクセスする前にオフライン・バックアップをとる必要があります。オフライン・バックアップをとる前にデータベースにアクセスしようとする、データベースはバックアップ・ペンディング状態に入ります。

複数のメンバーを追加したり、複数のメンバーをドロップすることができ、変更ごとに後でバックアップをとらなければならないわけではありません。例えば、メンバーを 3 つ追加する場合、バックアップをとる必要があるのは、すべての追加操作が完了した後だけです。ただし、2 つのメンバーを追加した後に 1 つのメンバーをドロップしたり、2 つのメンバーをドロップした後に 1 つのメンバーを追加したりする場合には、何らかの追加のトポロジー変更を実行する前に、バックアップをとる必要があります。

- **db2iupdt** コマンドは *DB2DIR/instance* ディレクトリーにあります。*DB2DIR* は、DB2 データベース製品の現行バージョンがインストールされている場所です。
- 非ルート・インスタンスを更新しない場合は、**db2nrupdt** 非 root インストール・インスタンス更新コマンドを参照してください。**db2iupdt** は、非ルート・インスタンスの更新をサポートしません。
- root ユーザーになるために **login** コマンドではなく **su** コマンドを使用している場合は、**su** コマンドを - オプションを指定して発行して、あたかも **login** コマンドを使用してシステムにログインしているかのように、プロセス環境を設定することを指示する必要があります。
- root ユーザー用の DB2 インスタンス環境を入手しないでください。DB2 インスタンス環境を入手している場合に、**db2iupdt** を実行することはサポートされていません。



- AIX 6.1 (またはそれ以上) では、このコマンドは、システム・ワークロード・パーティション (WPAR) グローバル環境内の共有 DB2 コピーから実行する場合、root ユーザーとして実行されなければなりません。WPAR は DB2 pureScale 環境ではサポートされていません。
- **db2iupdt** を実行して、リリース内のさらに高いレベルへインスタンスを更新する場合には、ルーチンとライブラリーが各メンバーから共有ロケーションにコピーされます。名前が同じものの、各ホストで内容が異なるライブラリーがある場合には、共有ロケーション内のライブラリー・コンテンツは、**db2iupdt** コマンドを最後に実行したホストのライブラリー・コンテンツになります。

### Windows オペレーティング・システムの場合

- **db2iupdt** コマンドは **DB2PATH\bin** ディレクトリーにあります。**DB2PATH** は、DB2 データベース製品の現行バージョンがインストールされている場所です。
- インスタンスは、**db2iupdt** コマンドの発行元から DB2 コピーに更新されます。ただし、インスタンス・プロファイルを現在の場所から別の場所に移動するには、**/p** オプションを使用し、インスタンス・プロファイル・パスを指定します。こうしない場合、インスタンスの更新後にインスタンス・プロファイルが元の場所に残ります。代わりに **db2iupgrade** コマンドを使用して、前のリリースから現行リリースにアップグレードします。

## db2iupgrade - インスタンスのアップグレード・

インスタンスを前のリリースの DB2 コピーから、現行リリースの DB2 コピーにアップグレードします。 **db2iupgrade** コマンドの実行元の DB2 コピーは、アップグレードする DB2 コピーからのインスタンス・サポートをアップグレードしていません。

Linux および UNIX オペレーティング・システムでは、このコマンドは *DB2DIR/instance* ディレクトリにあります。*DB2DIR* は、DB2 データベース・システムの新規リリースがインストールされている場所です。このコマンドは、非 root インストールのインスタンス・アップグレードはサポートしません。

Windows オペレーティング・システムでは、このコマンドは *DB2PATH\bin* ディレクトリにあります。**DB2PATH** は、DB2 コピーがインストールされている場所です。インスタンス・プロファイルを現在の場所から別の場所に移動するには、**/p** オプションを使用し、インスタンス・プロファイル・パスを指定します。こうしない場合、アップグレード後にインスタンス・プロファイルが元の場所に残ります。

### 許可

Linux および UNIX オペレーティング・システム上の root ユーザー権限、または Windows オペレーティング・システム上のローカル管理者

### コマンド構文

Linux および UNIX オペレーティング・システムの場合

```

▶▶ db2iupgrade [-d] [-k]
                [-j "TEXT_SEARCH [,servicename] [,portnumber]"
                [-a AuthType] [-u FencedID] InstName

```

Windows オペレーティング・システムの場合

```

▶▶ db2iupgrade InstName /u:username,password
                [/p:instance-profile-path] [/q] [/a:authType]
                [/j "TEXT_SEARCH [,servicename] [,portnumber]" [/?]

```

### コマンド・パラメーター

Linux および UNIX オペレーティング・システムの場合



## db2iupgrade - インスタンスのアップグレード

**-d** デバッグ・モードをオンにします。このオプションは、DB2 データベース・サポートからの指示があった場合にのみ使用してください。

**-k** **db2iupgrade** コマンドの実行元の DB2 コピーでサポートされている場合は、アップグレード前のインスタンス・タイプを保持します。このパラメーターが指定されていない場合、インスタンス・タイプは、サポートされるデフォルトのインスタンス・タイプにアップグレードされます。

### **-j "TEXT\_SEARCH"**

DB2 Text Search サーバーを、サービス名と TCP/IP ポート番号に対する生成済みのデフォルト値を使用して構成します。インスタンス・タイプがクライアントの場合、このパラメーターは使用できません。

### **-j "TEXT\_SEARCH, servicename"**

DB2 Text Search サーバーを、提供されたサービス名と自動的に生成されたポート番号を使用して構成します。サービス名に `services` ファイル内で割り当てられたポート番号がある場合、それはその割り当てられたポート番号を使用します。

### **-j "TEXT\_SEARCH, servicename, portnumber"**

DB2 Text Search サーバーを、提供されたサービス名とポート番号を使用して構成します。

### **-j "TEXT\_SEARCH, portnumber"**

DB2 Text Search サーバーを、デフォルトのサービス名と提供されたポート番号を使用して構成します。有効なポート番号は 1024 - 65535 の範囲になければなりません。

### **-a AuthType**

インスタンスの認証タイプ (SERVER、CLIENT、または SERVER\_ENCRYPT) を指定します。デフォルトは SERVER です。

### **-u FencedID**

fenced ユーザー定義関数および fenced ストアード・プロシージャを実行するユーザー ID の名前を指定します。このオプションは、DB2 クライアント・インスタンスを DB2 サーバー・インスタンスにアップグレードする際に必要です。

### *InstName*

インスタンスの名前を指定します。

## Windows オペレーティング・システムの場合

### *InstName*

インスタンスの名前を指定します。

### **/u:username,password**

DB2 サービスのアカウント名およびパスワードを指定します。パーティション・インスタンスをアップグレードする場合には、このオプションは必須です。

## db2iupgrade - インスタンスのアップグレード

**/p:instance-profile-path**

アップグレードされるインスタンス用の新しいインスタンス・プロファイル・パスを指定します。

**/q db2iupgrade** コマンドを静止モードで発行します。

**/a:authType**

インスタンスの認証タイプ (SERVER、CLIENT、または SERVER\_ENCRYPT) を指定します。

**/j "TEXT\_SEARCH"**

DB2 Text Search サーバーを、サービス名と TCP/IP ポート番号に対する生成済みのデフォルト値を使用して構成します。インスタンス・タイプがクライアントの場合、このパラメーターは使用できません。

**/j "TEXT\_SEARCH, servicename"**

DB2 Text Search サーバーを、提供されたサービス名と自動的に生成されたポート番号を使用して構成します。サービス名に `services` ファイル内で割り当てられたポート番号がある場合、それはその割り当てられたポート番号を使用します。

**/j "TEXT\_SEARCH, servicename, portnumber"**

DB2 Text Search サーバーを、提供されたサービス名とポート番号を使用して構成します。

**/j "TEXT\_SEARCH, portnumber"**

DB2 Text Search サーバーを、デフォルトのサービス名と提供されたポート番号を使用して構成します。有効なポート番号は 1024 - 65535 の範囲になければなりません。

**/? db2iupgrade** コマンドの使用方法を表示します。

### 使用上の注意

**db2iupgrade** コマンドを使用してアップグレードできるのは、DB2 Enterprise Server Edition のインスタンス (インスタンス・タイプ `ese`) のみです。

**db2iupgrade** コマンドは、**db2ckupgrade** コマンドを **-not1** パラメーターを指定して呼び出し、**db2ckupgrade** のログ・ファイルとして `upgrade.log` を指定します。**db2iupgrade** 用に作成されるデフォルトのログ・ファイルは `/tmp/db2ckupgrade.log.processID` です。インスタンスをアップグレードする前に、ローカル・データベースがアップグレードの準備ができていることを確認してください。**-not1** パラメーターは、タイプ 1 索引の検査を使用不可にします。ログ・ファイルは、Linux および UNIX オペレーティング・システムの場合はインスタンスのホーム・ディレクトリー、Windows オペレーティング・システムの場合は現行ディレクトリーに作成されます。**db2ckupgrade** コマンドが何らかのエラーを戻した場合、インスタンスのアップグレードは続行されません。

パーティション・データベース環境では、**db2iupgrade** コマンドを実行する前に、**db2ckupgrade** コマンドを実行します。**db2ckupgrade** コマンドは、すべてのパーティションを検査して、パーティションで検出されたエラーを返します。アップグレ

ードの準備ができていないかすべてのデータベース・パーティションをチェックしない場合、インスタンスのアップグレードが成功したとしても、それに続くデータベースのアップグレードで失敗する可能性があります。詳しくは、**db2ckupgrade** を参照してください。

### Linux および UNIX オペレーティング・システムの場合

- **db2iupgrade** コマンドを使用することによって旧バージョンの DB2 インスタンスを DB2 データベース・システムの現行バージョンにアップグレードする場合、旧 DB2 データベース・インストール・パス内で定義されている DB2 グローバル・プロファイル変数は、新しいインストール・ロケーションにアップグレードされません。インスタンス固有の DB2 インスタンス・プロファイル変数のアップグレードは、インスタンスのアップグレード後に実行されます。
- root ユーザーになるために **login** コマンドではなく **su** コマンドを使用している場合は、**su** コマンドを - オプションを指定して発行し、あたかも **login** コマンドを使用してシステムにログインしているかのように、プロセス環境を設定することを指示する必要があります。
- root ユーザー用の DB2 インスタンス環境を入手しないでください。DB2 インスタンス環境を入手している場合に、**db2iupgrade** コマンドを実行することはサポートされていません。
- AIX 6.1 (またはそれ以上) では、このコマンドは、システム・ワークロード・パーティション (WPAR) グローバル環境内の共有 DB2 コピーから実行する場合、root ユーザーとして実行されなければなりません。WPAR は DB2 pureScale 環境ではサポートされていません。

## db2jdbcbind - DB2 JDBC パッケージ・バインド・プログラム

このユーティリティーは、DB2 データベースに JDBC パッケージをバインドまたは再バインドするために使用されます。

JDBC および CLI は同じパッケージを共有します。CLI パッケージが既にデータベースにバインドされている場合、このユーティリティーを実行する必要はなく、そうでない場合は実行する必要があります。

### 許可

以下の権限のいずれか。

- DBADM
- パッケージが存在しない場合は、BINDADD 特権および以下のどちらかが必要です。
  - パッケージのスキーマ名が存在しない場合は、データベースに対する IMPLICIT\_SCHEMA 権限
  - パッケージのスキーマ名が存在している場合、そのスキーマに対する CREATEIN 特権。
- パッケージが存在する場合は、スキーマに対する ALTERIN 特権
- パッケージに対する BIND 特権 (パッケージが存在する場合)

### 必要な接続

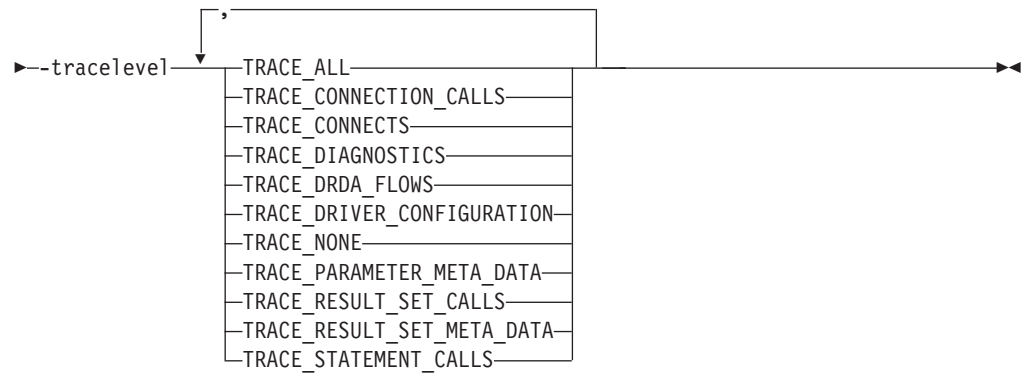
このコマンドは、データベース接続を確立します。

### コマンド構文

```

▶ db2jdbcbind [-help] -url jdbc:db2://servername:portnumber/dbname
▶ --user username --password password [-collection collection ID]
▶ [-size number of packages]

```



## コマンド・パラメーター

**-help** ヘルプ情報を表示します。その他のすべてのオプションは無視されます。

**-url jdbc:db2://servername:portnumber/dbname**

データベース接続の設定に使用する JDBC URL を指定します。接続の確立には DB2 JDBC タイプ 4 ドライバーが使用されます。

**-user username**

データベースに接続するときに使用する名前を指定します。

**-password password**

ユーザー名のパスワードを指定します。

**-collection collection ID**

コレクション ID (CURRENT PACKAGESET)。パッケージに使用します。デフォルトは NULLID です。これは、パッケージ・セットの複数のインスタンスを作成するために使用します。このオプションは、Connection または DataSource プロパティ `currentPackageSet` とともにのみ使用できます。

**-size number of packages**

それぞれの DB2 トランザクション分離レベルおよびカーソルの HOLD 設定ごとにバインドする内部パッケージの数。デフォルトは 3 です。DB2 分離レベルが 4 つ、カーソルの HOLD 設定が 2 つあるため、バインドされる動的パッケージ数は、このオプションに指定した数の  $4 \times 2 = 8$  倍になります。また、単一の静的パッケージは、必ず内部使用用にバインドされます。

**-tracelevel**

トレースのレベルを識別します。トラブルシューティングの場合にのみ必要です。

## db2ldcfg - LDAP 環境の構成

現行のログオン・ユーザー用の Lightweight Directory Access Protocol (LDAP) ユーザー識別名 (DN) およびパスワードを、IBM LDAP クライアントを使用する LDAP 環境で構成します。

### 許可

なし

### 必要な接続

なし

### コマンド構文

```
db2ldcfg -u userDN -w password -r
```

### コマンド・パラメーター

#### -u *userDN*

LDAP ディレクトリーにアクセスする際に使用する LDAP ユーザーの識別名を指定します。以下の例に示すように、識別名はいくつかの部分に分かれており、jdoe などのユーザー名や、ドメイン・ネーム、組織名、また com または org などの接尾部があります。

#### -w *password*

パスワードを指定します。

#### -r

マシン環境からユーザーの DN およびパスワードを除去します。

#### 例:

```
db2ldcfg -u "uid=jdoe,dc=mydomain,dc=myorg,dc=com" -w password
```

### 使用上の注意

IBM LDAP クライアントを使用する LDAP 環境では、現行のログオン・ユーザー用のデフォルト LDAP ユーザーの DN およびパスワードを構成できます。LDAP ユーザーの DN およびパスワードが構成されると、その DN およびパスワードがユーザーの環境に保存され、DB2 データベースから LDAP ディレクトリーへのアクセス時に毎回使用されます。LDAP ユーザーのデフォルトの DN およびパスワードを構成すると、LDAP コマンドや API の発行時に LDAP ユーザーの DN およびパスワードを指定する必要がなくなります。ただし、コマンドまたは API が発行される際に LDAP ユーザーの DN およびパスワードが指定されると、デフォルト設定は上書きされてしまいます。

このコマンドは、IBM LDAP クライアントを使用する場合にのみ実行できます。Microsoft LDAP クライアントでは、現行のログオン・ユーザーの資格情報が使用されます。

バージョン 10 フィックスパック 1 以降では、CLI アプリケーションを使用する場合、db2dsdriver.cfg ファイルの ldapsrvr セクション (<ldapsrvr>) でキーワ

ード **UserID** と **Password** を使用することにより、LDAP サーバー資格情報を提供できます。

---

## db2level - DB2 サービス・レベルの表示

インストール済み DB2 製品の現行バージョンおよびサービス・レベルを表示します。このコマンドからの出力は、デフォルトでコンソールに表示されます。

### 許可

なし

### 必要な接続

なし

### コマンド構文

▶▶—db2level—◀◀

### コマンド・パラメーター

なし

### 例

Windows オペレーティング・システムでは、**db2level** コマンドで DB2 コピー名が表示されます。以下に例を示します。

```
DB21085I Instance "DB2" uses "32" bits and DB2 code release "SQL09010" with
level identifier "01010107".
Informational tokens are "DB2 v9.1.0.189", "n060119", "", and Fix Pack "0".
Product is installed at "c:¥SQLLIB" with DB2 Copy Name "db2build".
```

Linux および UNIX ベースのオペレーティング・システムでは、**db2level** コマンドで DB2 コピー名は表示されません。以下に例を示します。

```
DB21085I Instance "wqzhuang" uses "64" bits and DB2 code release "SQL09010"
with level identifier "01010107".
Informational tokens are "DB2 v9.1.0.0", "n060124", "", and Fix Pack "0".
Product is installed at "/home/wqzhuang/sqllib".
```

### 使用上の注意

コマンドによる情報出力には、リリース、レベル、およびさまざまな情報トークンが含まれます。





## db2licm - ライセンス管理ツール

指定します。これは、ライセンス製品 CD から得られます。または IBM 担当員か認定販売店にお問い合わせください。

### **-e** *product-identifier*

システム上の制約ポリシーを更新します。有効な値は以下のとおりです。

**HARD** ライセンスなしの要求が許可されないことを指定します。

**SOFT** ライセンスなしの要求がログに記録されるが、制限はされないことを指定します。

### **-p** *product-identifier*

システムで使用するライセンス・ポリシー・タイプを更新します。

#### **CONCURRENT**

同時ユーザー・ポリシーに関して指定します。

**OFF** すべてのポリシーをオフにするよう指定します。

### **-r** *product-identifier*

製品のライセンスを除去します。特定の製品の製品 ID を得るには、**-l** オプションを指定してコマンドを呼び出してください。

### **-u** *product-identifier num-users*

お客様が購入したユーザー・ライセンスの数を更新します。製品 ID とユーザー数を指定してください。

### **-c** *product-identifier num-connectors*

お客様が購入したコネクタ・ライセンスの数を更新します。製品 ID とコネクタ・ライセンス数を指定してください。

### **-g** *filename*

準拠レポートを生成します。出力を保管するファイル名を指定してください。

**-x** ライセンス準拠レポートの生成のために、ライセンス準拠情報をリセットします。

**-l** すべての製品および (製品 ID を含む) 使用できるライセンス情報をリストします。

#### **SHOW DETAIL**

ライセンス・フィーチャーに関する詳細情報 (存在する場合) を表示するよう指定します。

**-v** バージョン情報を表示します。

### **-h | -?**

ヘルプ情報を表示します。このオプションを指定すると、他のすべてのオプションは無視され、ヘルプ情報だけが表示されます。

## 例

```
db2licm -a db2ese_u.lic
db2licm -p db2consv concurrent
db2licm -r db2ese
db2licm -u db2wse 10
db2licm -e db2ese SOFT
```

以下の出力例は、すべての製品および (製品 ID を含む) 使用できるライセンス情報をリストします。

```
C:¥Program Files¥IBM¥SQLLIB¥BIN>db2licm -l
Product name:          "DB2 Enterprise Server Edition"
License type:          "Trial"
Expiry date:           "08/31/2009"
Product identifier:    "db2ese"
Version information:   "10.1"
```

**注:** Linux では、**db2licm -l** コマンドは、以下の通知メッセージを /var/log/messages に書き込みます。

```
kernel: Program db2licm tried to access /dev/mem between 1f0000000f0000->10100000000.
```

## db2listvolumes - すべてのディスク・ボリュームの GUID の表示

Windows オペレーティング・システム上で定義されたすべてのディスク・ボリュームの GUID を表示します。

このコマンドにより、ツールの発行元のディレクトリーに 2 つのファイルが作成されます。1 つ目のファイルは `volumes.xml` と言い、XML 対応ブラウザで簡単に表示するために、XML でエンコードされた各ディスク・ボリュームの情報が入っています。2 つ目のファイルは `tablespace.ddl` と言い、表スペース・コンテナを指定するために必要な構文が入っています。このファイルは、表スペース定義に必要な残りの情報を記入するために更新する必要があります。`db2listvolumes` コマンドは、コマンド行引数を必要としません。このコマンドは Windows オペレーティング・システムでのみ使用できます。

### 許可

Administrator

### 必要な接続

なし

### コマンド構文

▶—db2listvolumes—▶

### コマンド・パラメーター

なし

---

## db2logsForRfwd - ロールフォワード・リカバリーに必要なログのリスト

DB2TSCHG.HIS ファイルを解析します。このユーティリティを使用すると、ユーザーは、表スペース・ロールフォワード操作に必要なログ・ファイルを見つけることができます。

このユーティリティは `sqllib/bin` にあります。

### 許可

なし

### 必要な接続

なし

### コマンド構文

```
▶▶ db2logsForRfwd path -all ▶▶
```

### コマンド・パラメーター

*path* DB2TSCHG.HIS ファイルの絶対パスおよび名前。

**-all** 詳細情報を表示します。

### 例

```
db2logsForRfwd /home/hotel73/roecken/roecken/NODE0000/SQL00001/MEMBER0000/DB2TSCHG.HIS
db2logsForRfwd DB2TSCHG.HIS -all
```

## db2look - DB2 統計および DDL 抽出ツール

テスト・データベース上に実動データベースのデータベース・オブジェクトを再作成するのに必要な、データ定義言語 (DDL) ステートメントを抽出します。 **db2look** コマンドは、オブジェクト・タイプごとに DDL ステートメントを生成します。なお、このコマンドは、SYSTOOLS スキーマのすべてのオブジェクトを無視します。ただし、ユーザー定義関数とストアド・プロシージャについては例外です。

テスト・システムに実動システムのデータのサブセットを含めておくと、便利なが多くあります。しかし、そのようなテスト・システム用に選択したアクセス・プランが、必ずしも実動システム用に選択したアクセス・プランと同じであるとは限りません。ただし、**db2look** ツールを使用すると、実動システムで使用するものとアクセス・プランが類似しているテスト・システムを作成することができます。このツールを使用して、テスト・データベース上の実動データベースにあるオブジェクトのカタログ統計を複製するのに必要な UPDATE ステートメントを生成することができます。また、このツールを使用して、**UPDATE DATABASE CONFIGURATION**、**UPDATE DATABASE MANAGER CONFIGURATION**、および **db2set** コマンドを生成することで、テスト・データベース上の照会オプティマイザー関連の構成パラメーターとレジストリー変数の値を、実動データベースでの値に合うように設定できます。

**db2look** コマンドによって生成される DDL ステートメントは、元の SQL オブジェクトのすべての特性を複製するとは限らないため、生成された DDL ステートメントを必ず確認するようにしてください。パーティション・データベース環境の表スペースでは、アクティブでないデータベース・パーティションが存在する場合、DDL が完全でない可能性があります。 **ACTIVATE DATABASE** コマンドを使用することによって、すべてのデータベース・パーティションがアクティブであることを確認してください。

### 許可

システム・カタログ表に対する SELECT 特権。

表スペース・コンテナ DDL を生成する場合などでは、以下の権限のいずれか 1 つが必要です。

- SYSADM
- SYSCTRL
- SYSMOINT
- SYSMON
- DBADM
- ADMIN\_GET\_STORAGE\_PATHS 表関数に対する EXECUTE 特権

### 必要な接続

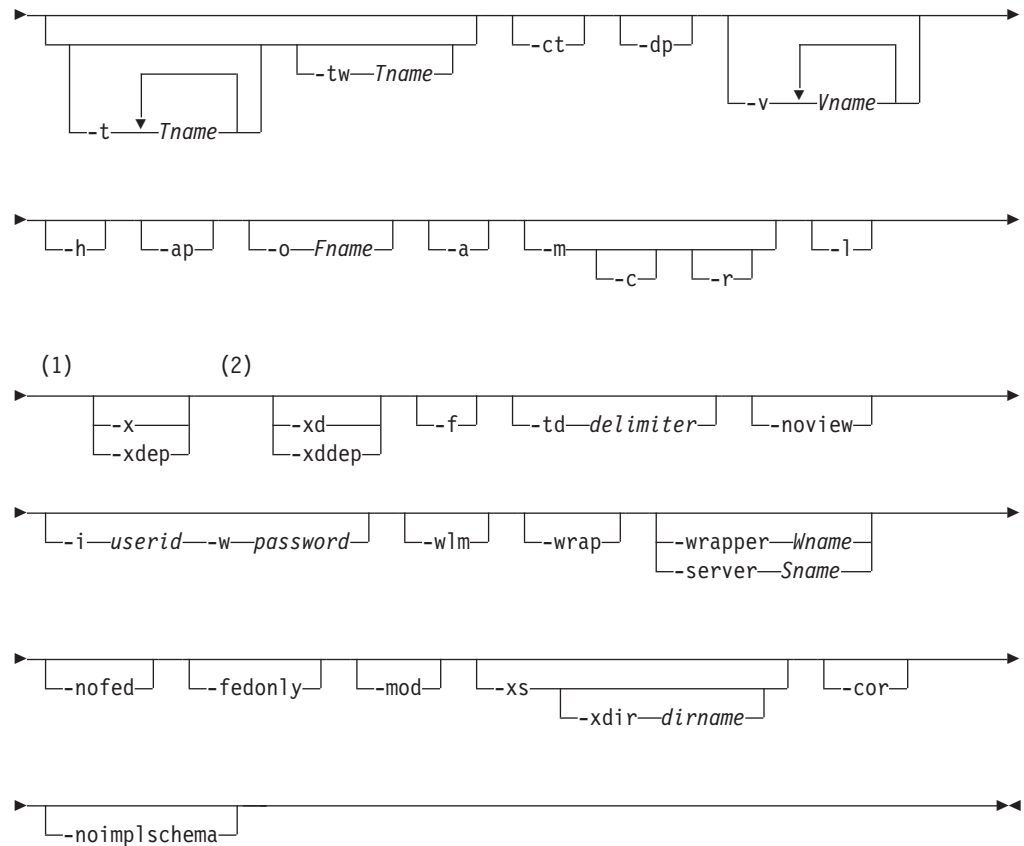
なし

### コマンド構文

```

▶▶ db2look -d DBname [-e] [-u creator] [-z schema]

```



## 注:

- 1 **-x** パラメーターと **-xdep** パラメーターを両方とも指定することはできません。
- 2 **-xd** パラメーターと **-xddep** パラメーターを両方とも指定することはできません。

## コマンド・パラメーター

**-d** *DBname*

照会する実動データベースの別名。 *DBname* としては、DB2 for Linux, UNIX, and Windows または DB2 Version 9.1 for z/OS データベースの名前を指定できます。 *DBname* が DB2 for z/OS データベースである場合には、**db2look** コマンドは、以下のステートメントを OS/390 および z/OS オブジェクト用に生成します。

- 表、索引、ビュー、およびユーザー定義特殊タイプ用の DDL ステートメント
- 表、列、列分散および索引用の UPDATE 統計ステートメント

これらの DDL および UPDATE 統計は、DB2 for Linux, UNIX, and Windows データベースには適用できますが、DB2 for z/OS データベースには適用できません。これらのステートメントは、OS/390 および z/OS のオブジェクトを抽出して、それらを DB2 for Linux, UNIX, and Windows データベース内に再作成しようとするユーザーに役立ちます。

- e 以下のデータベース・オブジェクト用の DDL ステートメントを抽出します。
  - 別名
  - 監査ポリシー
  - チェック制約
  - 関数マッピング
  - 関数テンプレート
  - グローバル変数
  - 索引 (パーティション表上のパーティション索引を含む)
  - SPECIFICATION ONLY 指定の索引
  - マテリアライズ照会表 (MQT)
  - ニックネーム
  - 主キー、参照整合性、およびチェック制約
  - 参照整合性制約
  - ロール
  - スキーマ
  - セキュリティー・ラベル
  - セキュリティー・ラベル・コンポーネント
  - セキュリティー・ポリシー
  - 順序
  - サーバー
  - ストアード・プロシージャ
  - 表

注: SYSIBM.SYSTABLES カタログ表の列 STATISTICS\_PROFILE の値は含まれません。

- トリガー
- トラストッド・コンテキスト
- タイプ・マッピング
- ユーザー・マッピング
- ユーザー定義特殊タイプ
- ユーザー定義関数
- ユーザー定義メソッド
- ユーザー定義構造化タイプ
- ユーザー定義トランスフォーム
- ビュー
- ラッパー

**db2look** コマンドによって生成された DDL ステートメントを使用してユーザー定義関数を再作成する場合、その関数を使用できる状態にするには、関数が参照するソース・コード (EXTERNAL NAME 節など) が使用可能でなければなりません。



**-u creator**

指定された作成者 ID で作成されたオブジェクトに対する DDL ステートメントを生成します。出力を、指定された作成者 ID で作成されたオブジェクトに制限します。出力に作動不能オブジェクトは含まれません。作動不能オブジェクトを表示するには、**-a** パラメーターを使用します。**-a** パラメーターを指定した場合、**-u** パラメーターは無視されます。

**-z schema**

指定されたスキーマ名を持つオブジェクトに対する DDL ステートメントを生成します。出力を、指定されたスキーマ名を持つオブジェクトに制限します。出力に作動不能オブジェクトは含まれません。作動不能オブジェクトを表示するには、**-a** パラメーターを使用します。**-z** パラメーターを指定しない場合には、すべてのスキーマ名のオブジェクトが抽出されます。**-a** パラメーターを指定した場合、**-z** パラメーターは無視されます。また、このパラメーターも、フェデレーテッド DDL ステートメントでは無視されません。

**-t Tname1 Tname2 ... TnameN**

指定した表とその従属オブジェクトに対する DDL ステートメントを生成します。表リストに指定された表だけに出力を制限し、ユーザー指定のすべての表の、すべての従属オブジェクトについて、DDL ステートメントを生成します。表の最大数は 30 です。リストを以下のように指定します。

- 表名をブランク・スペースで区切ります。
- 大/小文字の区別のある名前と 2 バイト文字セット (DBCS) 名は、円記号 (¥) および二重引用符 (" ") で囲む必要があります (例えば ¥" MyTable ¥")。
- 複数語の表名は、円記号と 2 セットの二重引用符 (例えば "¥"My Table¥") で囲むことにより、その対が一語ごとにコマンド行プロセッサ (CLP) で評価されないようにします。1 セットの二重引用符だけを使用した場合 (例えば "My Table") には、すべての語が大文字に変換され、**db2look** コマンドは大文字の表名 (例えば MY TABLE) を探すこととなります。

**-t** パラメーターを **-l** パラメーターと共に指定すると、パーティション表がサポートされます。

**-z schema** パラメーターを使用せずに表名を完全修飾するために、*schema.table* という形式の 2 部から成る表名を使用することができます。表が持っている従属オブジェクトが、表のスキーマとは異なるスキーマにあるときに、その従属オブジェクトのために生成される DDL ステートメントを必要とする場合には、2 部から成る表名を使用してください。**-z schema** パラメーターを使用してスキーマを指定する場合、そのパラメーターは同じ親スキーマを持たない従属オブジェクトを除外するので、そのような従属オブジェクトのための DDL ステートメントは生成されないこととなります。

**-tw Tname**

*Tname* で指定するパターンに一致する名前を持つ表について DDL ステートメントを生成し、それらの表のすべての従属オブジェクトについて、DDL ステートメントを生成します。*Tname* は、1 つの値だけでなければなりません。*Tname* 内の下線文字 ( ) は、任意の 1 文字を表します。パーセン

ト記号 (%) は、ゼロ個以上の文字のストリングを表します。-tw を指定した場合は、-t オプションが無視されます。

-z *schema* パラメーターを使用せずに表名を完全修飾するために、*schema.table* という形式の 2 部から成る表名を使用することができます。表が持っている従属オブジェクトが、表のスキーマとは異なるスキーマにあるときに、その従属オブジェクトのために生成される DDL ステートメントを必要とする場合には、2 部から成る表名を使用してください。-z *schema* パラメーターを使用してスキーマを指定する場合、そのパラメーターは同じ親スキーマを持たない従属オブジェクトを除外するので、そのような従属オブジェクトのための DDL ステートメントは生成されないことになります。

-ct オブジェクト作成時刻に基づいて DDL ステートメントを生成します。オブジェクト DDL ステートメントは、正しい従属順序では表示されない可能性があります。-ct パラメーターを指定する場合、db2look コマンドは以下の追加パラメーターだけをサポートします: -e、-a、-u、-z、-t、-tw、-v、-l、-noview、および -wlm。-ct パラメーターを、-z および -t パラメーターと共に使用した場合、db2look コマンドは、表、統計ビュー、列、および索引についての統計を複製するのに必要な UPDATE ステートメントを生成します。

-dp CREATE ステートメントの前に DROP ステートメントを生成します。ドロップされるオブジェクトに依存するオブジェクトが存在する場合、その DROP ステートメントは機能しない可能性があります。例えば、スキーマに依存する表がある場合には、そのスキーマをドロップすることはできません。また、ユーザー定義タイプまたはユーザー定義関数に依存するタイプ、関数、トリガー、または表がある場合、それらのユーザー定義タイプまたはユーザー定義関数をドロップすることはできません。型付き表の場合、DROP TABLE HIERARCHY ステートメントはルート表についてのみ生成されます。表がドロップされると索引、主キーと外部キー、および制約も常にドロップされるため、それらについての DROP ステートメントは生成されません。RESTRICT ON DROP 属性を持つ表はドロップできません。

-v *Vname1 Vname2 ... VnameN*

指定したビューに対する DDL ステートメントを生成しますが、それらの従属オブジェクトに対しては生成しません。ビューの最大数は 30 です。大/小文字の区別、DBCS、および複数語表名を制御する規則は、ビュー名にも適用されます。-t パラメーターを指定した場合、-v パラメーターは無視されます。

ビューを完全修飾するために、*schema.view* フォーマットの 2 部から成るビュー名を使用できます。

-h ヘルプ情報を表示します。このパラメーターを指定した場合、他のすべてのパラメーターは無視されます。

-ap 監査ポリシーを他のデータベース・オブジェクトに関連付けるのに必要な AUDIT USING ステートメントを生成します。

-o *Fname*

出力を *Fname* ファイルに書き込みます。拡張子を指定しない場合、拡張子 .sql が使用されます。このパラメーターを指定しない場合、出力は標準出力に書き込まれます。

- a** 作動不能オブジェクトを含め、他のユーザーによって作成されたオブジェクトに対する DDL ステートメントを生成します。例えば、このパラメーターを **-e** パラメーターと共に指定した場合、データベース内のすべてのオブジェクト用の DDL ステートメントが抽出されます。このパラメーターを **-m** パラメーターと共に指定した場合には、データベース内のすべてのユーザー作成の表および索引用の UPDATE 統計ステートメントが抽出されます。
- u** および **-a** パラメーターをいずれも指定しない場合には、USER 環境変数が使用されます。UNIX オペレーティング・システム上では、この変数を明示的に設定する必要はありません。ただし、Windows オペレーティング・システム上では、USER 環境変数のデフォルト値がありません。したがって、SYSTEM 変数にユーザー変数を設定するか、またはセッションに対して **set USER=username** コマンドを発行する必要があります。
- m** 表、統計ビュー、列、および索引についての統計を複製するのに必要な UPDATE ステートメントを生成します。 **-m** パラメーターを使用することを、模倣モードでの実行といいます。
- c** このオプションを指定した場合、**db2look** コマンドは、COMMIT、CONNECT、および CONNECT RESET ステートメントを生成しません。デフォルト・アクションでは、これらのステートメントを生成します。 **-m** または **-e** パラメーターも指定されているのであれば、このオプションは無視されます。
- r** このオプションを **-m** パラメーターと共に指定する場合、**db2look** コマンドは **RUNSTATS** コマンドを生成しません。デフォルト・アクションでは、**RUNSTATS** コマンドを生成します。
- 重要:** 別のデータベースに対して、**-m** パラメーター付きの **db2look** コマンドを使用して作成されたコマンド・プロセッサ・スクリプトを実行する予定の場合 (例えば、テスト・データベースのカatalog統計を実動環境のカatalog統計と突き合わせる場合)、両方のデータベースが同じコード・セットとテリトリーを使用する必要があります。
- l** 以下のデータベース・オブジェクト用の DDL ステートメントを生成します。
- ユーザー定義表スペース
  - ユーザー定義ストレージ・グループ
  - ユーザー定義データベース・パーティション・グループ
  - ユーザー定義バッファー・プール
- x** GRANT ステートメントなどの許可 DDL ステートメントを生成します。サポートされている許可には、以下のものが含まれます。
- 列: UPDATE、REFERENCES
  - データベース: ACCESSCTRL、BINDADD、CONNECT、CREATETAB、CREATE\_EXTERNAL\_ROUTINE、CREATE\_NOT\_FENCED\_ROUTINE、DATAACCESS、DBADM、EXPLAIN、IMPLICIT\_SCHEMA、LOAD、QUIESCE\_CONNECT、SECADM、SQLADM、WLMADM
  - 免除
  - グローバル変数

- 索引: CONTROL
- パッケージ: CONTROL、BIND、EXECUTE
- ロール
- スキーマ: CREATEIN、DROPIN、ALTERIN
- セキュリティー・ラベル
- シーケンス: USAGE、ALTER
- ストアード・プロシージャ: EXECUTE
- 表: ALTER、SELECT、INSERT、DELETE、UPDATE、INDEX、REFERENCE、CONTROL
- ビュー: SELECT、INSERT、DELETE、UPDATE、CONTROL
- ユーザー定義関数 (UDF): EXECUTE
- ユーザー定義メソッド: EXECUTE
- 表スペース: USE
- ワークロード: USAGE

**注:** **-t** または **-tw** パラメーターのいずれかと共に使用した場合、このパラメーターは従属オブジェクトのための許可 DDL ステートメントを生成しません。親オブジェクトと従属オブジェクトに対して許可 DDL ステートメントを生成する場合は、**-xdep** パラメーターを使用してください。

**-xdep** 親オブジェクトと従属オブジェクトに対して許可 DDL ステートメント (例えば GRANT ステートメント) を生成します。このパラメーターは、**-t** および **-tw** パラメーターがいずれも指定されていない場合には、無視されます。サポートされている許可には、以下のものが含まれます。

- 列: UPDATE、REFERENCES
- 索引: CONTROL
- ストアード・プロシージャ: EXECUTE
- 表: ALTER、SELECT、INSERT、DELETE、UPDATE、INDEX、REFERENCE、CONTROL
- 表スペース: USE
- ユーザー定義関数 (UDF): EXECUTE
- ユーザー定義メソッド: EXECUTE
- ビュー: SELECT、INSERT、DELETE、UPDATE、CONTROL

**-xd** オブジェクト作成時に SYSIBM によって許可が付与されたオブジェクト用の許可 DDL ステートメントを含む、許可 DDL ステートメントを生成します。システム・カタログ表とカタログ・ビューに対しては、許可 DDL は生成されません。

**注:** **-t** または **-tw** パラメーターのいずれかと共に使用した場合、このパラメーターは従属オブジェクトのための許可 DDL ステートメントを生成しません。親オブジェクトと従属オブジェクトに対して許可 DDL ステートメントを生成する場合は、**-xddep** パラメーターを使用してください。

**-xddep** オブジェクト作成時に SYSIBM によって権限を付与されたオブジェクトの許可 DDL ステートメントを含めて、親オブジェクトおよび従属オブジェク

トに対して、すべての許可 DDL ステートメントを生成します。このパラメーターは、**-t** および **-tw** パラメーターがいずれも指定されていない場合には、無視されます。

**-f** 照会オプティマイザーに影響を与える構成パラメーターおよびレジストリー変数を抽出します。

**-td delimiter**

**db2look** コマンドによって生成される SQL ステートメントのステートメント区切り文字を指定します。デフォルトの区切り文字はセミコロン (;) です。抽出されたオブジェクトにはトリガーまたは SQL ルーチンが含まれる可能性があるため、**-e** パラメーターを指定する場合にはこのパラメーターを使用してください。

**-noview**

CREATE VIEW DDL ステートメントを抽出しないことを指定します。

**-i userid**

リモート・システムにログオンするために **db2look** コマンドが使用するユーザー ID を指定します。このパラメーターと **-w** パラメーターを指定すると、**db2look** コマンドをリモート・システム上のデータベースに対して実行することができます。ローカルとリモートのデータベースが、同じ DB2 バージョンを使用する必要があります。

**-w password**

リモート・システムにログオンするために **db2look** コマンドが使用するパスワードを指定します。このパラメーターと **-i** パラメーターを指定すると、**db2look** コマンドをリモート・システム上のデータベースに対して実行することができます。ローカルとリモートのデータベースが、同じ DB2 バージョンを使用する必要があります。

**-wlm**

WLM 固有の DDL 出力を生成します。この出力は、以下のものに関する CREATE および ALTER ステートメントの生成に使用できます。

- ヒストグラム
- サービス・クラス
- しきい値
- WLM イベント・モニター
- ワークロード
- 作業アクション・セット
- 作業クラス・セット

**-wrap**

ルーチン、トリガー、ビュー、PL/SQL パッケージの DDL ステートメントの難読化バージョンを生成します。

**-wrapper Wname**

指定したラッパーに適用するフェデレーテッド・オブジェクト用の DDL ステートメントを生成します。生成される可能性のあるフェデレーテッド DDL ステートメントには、以下のものがあります。

- CREATE FUNCTION ... AS TEMPLATE
- CREATE FUNCTION MAPPING
- CREATE INDEX SPECIFICATION

- CREATE NICKNAME
- CREATE SERVER
- CREATE TYPE MAPPING
- CREATE USER MAPPING
- CREATE WRAPPER
- GRANT (ニックネーム、サーバー、索引に対する特権)

ラッパー名を指定しない場合や、複数指定した場合には、エラーが戻されません。

### **-server** *Sname*

指定したサーバーに適用するフェデレーテッド・オブジェクト用の DDL ステートメントを生成します。生成される可能性のあるフェデレーテッド DDL ステートメントには、以下のものがあります。

- CREATE FUNCTION ... AS TEMPLATE
- CREATE FUNCTION MAPPING
- CREATE INDEX SPECIFICATION
- CREATE NICKNAME
- CREATE SERVER
- CREATE TYPE MAPPING
- CREATE USER MAPPING
- CREATE WRAPPER
- GRANT (ニックネーム、サーバー、索引に対する特権)

サーバー名を指定しない場合や、複数指定した場合には、エラーが戻されません。

**-nofed** フェデレーテッド DDL ステートメントが生成されないことを指定します。このパラメーターを指定した場合、**-wrapper** および **-server** パラメーターは無視されます。

### **-fedonly**

フェデレーテッド DDL ステートメントのみ生成されることを指定します。

**-mod** 各モジュール、および各モジュールに定義されているすべてのオブジェクト用に DDL ステートメントを生成します。

**-xs** XML スキーマと DTD をターゲット・データベースに登録するために必要なすべてのファイルをエクスポートし、それらに登録するための該当するコマンドを生成します。エクスポートされる XSR オブジェクトのセットは、**-u**、**-z**、**-a** の各パラメーターによって制御されます。

### **-xdir** *dirname*

XML 関連ファイルを指定されたパスにエクスポートします。このパラメーターを指定しない場合、XML 関連ファイルはすべて現行ディレクトリーにエクスポートされます。

**-cor** DDL ステートメントを CREATE OR REPLACE 節とともに生成します (ステートメントがその節をもともと含んでいるかどうかに関係なく)。



**-noimplschema**

暗黙的に作成されたスキーマに関する CREATE SCHEMA DDL ステートメントを生成しないように指定します。このパラメーターを指定する場合は **-e** パラメーターも指定する必要があります。

**例**

以下の例は、**db2look** コマンドの使用方法を示しています。

- データベース DEPARTMENT でユーザー walid によって作成されたオブジェクト用の DDL ステートメントを生成します。出力は、db2look.sql ファイルに送られます。

```
db2look -d department -u walid -e -o db2look.sql
```

- ianhe というスキーマ名を持ち、データベース DEPARTMENT でユーザー walid によって作成されたオブジェクト用の DDL ステートメントを生成します。出力は、db2look.sql ファイルに送られます。

```
db2look -d department -u walid -z ianhe -e -o db2look.sql
```

- UPDATE ステートメントを生成して、データベース DEPARTMENT でユーザー walid によって作成されたデータベース・オブジェクトの統計を複製します。出力は、db2look.sql ファイルに送られます。

```
db2look -d department -u walid -m -o db2look.sql
```

- ユーザー walid によって作成されたオブジェクト用の DDL ステートメントおよび UPDATE ステートメントの両方を生成して、同じユーザーによって作成されたデータベース・オブジェクトについての統計を複製します。出力は、db2look.sql ファイルに送られます。

```
db2look -d department -u walid -e -m -o db2look.sql
```

- データベース DEPARTMENT ですべてのユーザーによって作成されたオブジェクトの DDL ステートメントを生成します。出力は、db2look.sql ファイルに送られます。

```
db2look -d department -a -e -o db2look.sql
```

- すべてのユーザー定義のデータベース・パーティション・グループ、バッファークラス・プール、および表スペース用の DDL ステートメントを生成します。出力は、db2look.sql ファイルに送られます。

```
db2look -d department -l -o db2look.sql
```

- オプティマイザー関連のデータベースおよびデータベース・マネージャーの構成パラメーター用の UPDATE ステートメント、およびデータベース DEPARTMENT にあるオプティマイザー関連のレジストリー変数用の **db2set** コマンドを生成します。出力は、db2look.sql ファイルに送られます。

```
db2look -d department -f -o db2look.sql
```

- オプティマイザー関連のレジストリー変数と、データベース DEPARTMENT のための以下のステートメントに対して **db2set** コマンドを生成します。
  - すべてのデータベース・オブジェクトのための DDL ステートメント
  - すべての表と索引についての統計を複製するための UPDATE ステートメント
  - GRANT 許可ステートメント
  - オプティマイザー関連のデータベースおよびデータベース・マネージャーの構成パラメーターに対する UPDATE ステートメント

## db2look - DB2 統計および DDL 抽出ツール

- オプティマイザー関連のレジストリー変数に対する **db2set** コマンド
- すべてのユーザー定義のデータベース・パーティション・グループ、バッファ・プール、および表スペースに対する DDL ステートメント

出力は、db2look.sql ファイルに送られます。

```
db2look -d department -a -e -m -l -x -f -o db2look.sql
```

- オリジナルの作成者によって作成されたオブジェクトも含む、データベース DEPARTMENT 内のすべてのオブジェクトのすべての許可 DDL ステートメントを生成します。(この場合には、オブジェクトの作成時に SYSIBM によって権限が付与されました。) 出力は、db2look.sql ファイルに送られます。

```
db2look -d department -xd -o db2look.sql
```

- データベース DEPARTMENT ですべてのユーザーによって作成されたオブジェクトの DDL ステートメントを生成します。出力は、db2look.sql ファイルに送られます。

```
db2look -d department -a -e -td % -o db2look.sql
```

出力は CLP によって読み取ることができます。

```
db2 -td% -f db2look.sql
```

- データベース DEPARTMENT 内のオブジェクト用に、CREATE VIEW ステートメントを除く DDL ステートメントを生成します。出力は、db2look.sql ファイルに送られます。

```
db2look -d department -e -noview -o db2look.sql
```

- 指定した表に関連するデータベース DEPARTMENT 内のオブジェクト用に、DDL ステートメントを生成します。出力は、db2look.sql ファイルに送られます。

```
db2look -d department -e -t tab1 "¥"My TaBlE2¥" -o db2look.sql
```

- フェデレーテッド・データベース FEDDEPART にすべてのオブジェクト (フェデレーテッドおよび非フェデレーテッド) 用の DDL ステートメントを生成します。フェデレーテッド DDL ステートメントでは、指定されたラッパー FEDWRAP に適用されるもののみが生成されます。出力が標準出力に書き込まれます。

```
db2look -d feddepart -e -wrapper fedwrap
```

- 非フェデレーテッド DDL ステートメントのみを含むスクリプト・ファイルを生成します。以下のシステム・コマンドは、フェデレーテッド・データベース FEDDEPART に対して実行でき、フェデレーテッドではないデータベースの実行時に検出されたような出力を生成するだけです。出力は、out.sql ファイルに送られます。

```
db2look -d feddepart -e -nofed -o out
```

- データベース DEPARTMENT の中でスキーマ名が valid であるオブジェクト用の DDL ステートメントを生成します。組み込み XML スキーマおよび DTD を登録するために必要なファイルは、現行ディレクトリーにエクスポートされます。出力は、db2look.sql ファイルに送られます。

```
db2look -d department -z valid -e -xs -o db2look.sql
```

- データベース DEPARTMENT ですべてのユーザーによって作成されたオブジェクトの DDL ステートメントを生成します。組み込み XML スキーマおよび



DTD を登録するために必要なファイルは、/home/ofer/ofer/ ディレクトリーにエクスポートされます。出力が標準出力に書き込まれます。

```
db2look -d department -a -e -xs -xdir /home/ofer/ofer/
```

- データベース DEPARTMENT に対する WLM 固有の DDL ステートメントだけを生成します。

```
db2look -d department -wlm
```

- データベース DEPARTMENT 内のすべてのオブジェクトに対する DDL ステートメントを生成します。

```
db2look -d department -wlm -e -l
```

- スキーマ TABLES 内の親表 TAB1 と、TAB1 の従属ビュー (データベース DB1 内の VIEWS スキーマにあり VIEW1 と呼ばれる) の両方に対して DDL ステートメントを生成します。出力は、db2look.sql ファイルに送られます。

```
db2look -d DB1 -t TABLES.TAB1 -e -o db2look.sql
```

- スキーマ TABLES 内の親表 TAB1 と、TAB1 の従属ビュー (データベース DB1 内の VIEWS スキーマにあり VIEW1 と呼ばれる) に対して DDL ステートメントおよび許可 DDL ステートメントを生成します。出力は、db2look.sql ファイルに送られます。

```
db2look -d DB1 -t TABLES.TAB1 -e -xdep -o db2look.sql
```

- 模倣モードで TABLE1 表に RUNSTATS DDL ステートメントを生成します。出力は、db2look.sql ファイルに送られます。このコマンドで使用可能な RUNSTATS 節すべてがサポートされているわけではありません。

```
db2look -d DB1 -t TABLE1 -m -e -o db2look.sql
```

## 使用上の注意

Windows オペレーティング・システムでは、**db2look** コマンドは DB2 コマンド・ウィンドウから発行する必要があります。

デフォルトでは、インスタンス所有者には **db2look** パッケージに対する EXECUTE 特権があります。その他のユーザーが **db2look** コマンドを実行する場合、インスタンス所有者は **db2look** パッケージに対する EXECUTE 特権を付与する必要があります。 **db2look** パッケージ名を判別するには、以下のようにして **db2bfd** コマンドを使用できます。

```
cd ../sql1lib/bnd
db2bfd -b db2look.bnd
db2bfd -b db21kfun.bnd
db2bfd -b db21ksp.bnd
```

フェデレーテッド・オブジェクトのための DDL ステートメントを作成するためには、データベース・マネージャ構成でフェデレーテッド・システムを使用可能に設定する必要があります。 **db2look** コマンドがスクリプト・ファイルを生成した後、そのスクリプトを実行する前に、**federated** 構成パラメーターを YES に設定する必要があります。以下の **db2look** コマンド・パラメーターがフェデレーテッド環境でサポートされます。

### -ap

AUDIT USING ステートメントを生成します。

- e フェデレーテッド・オブジェクト用の DDL ステートメントを生成します。

## db2look - DB2 統計および DDL 抽出ツール

- f フェデレーテッド関連情報をデータベース・マネージャー構成から抽出します。
- m ニックネームの統計を抽出します。
- x フェデレーテッド・オブジェクトに対する特権を付与する GRANT ステートメントを生成します。

### -xd

フェデレーテッド・オブジェクトに対してシステムが付与する特権を追加する DDL ステートメントを生成します。

### -wlm

WLM 固有の DDL ステートメントを生成します。

ニックネーム列とリモート表列が異なるデータ型である場合には、**db2look** コマンドは、ニックネーム列に対する ALTER COLUMN ステートメントを生成します。

出力スクリプトを変更して、CREATE USER MAPPING ステートメントのリモート・パスワードを追加する必要があります。

DB2 ファミリー・インスタンスをデータ・ソースとして定義するのに使用される、これらの CREATE SERVER ステートメントに、AUTHORIZATION および PASSWORD を追加して、**db2look** コマンド出力スクリプトを変更する必要があります。

-tw オプションの使用法は、次のとおりです。

- abc で始まる名前を持つ表に関連した、DEPARTMENT データベース内のオブジェクトの DDL ステートメントを生成し、その出力を db2look.sql ファイルに送信するには、次のようにします。

```
db2look -d department -e -tw abc% -o db2look.sql
```

- 名前の 2 番目の文字が d である表に関連した、DEPARTMENT データベース内のオブジェクトの DDL ステートメントを生成し、その出力を db2look.sql ファイルに送信するには、次のようにします。

```
db2look -d department -e -tw _d% -o db2look.sql
```

- **db2look** コマンドは、LIKE 述部を使用して、どの表名が *Tname* 引数に指定されたパターンに一致するかを評価します。LIKE 述部を使用する以上、\_ 文字または % 文字のいずれかが表名の一部である場合には、\_ または % のすぐ前に円記号 (¥) エスケープ文字を置かなければなりません。この場合、\_ も % も、*Tname* 内でワイルドカード文字として使用することはできません。例えば、名前の最初でも最後でもない場所にパーセント (%) 記号を持つ表に関連した、DEPARTMENT データベース内のオブジェクトの DDL ステートメントを生成するには、次のようにします。

```
db2look -d department -e -tw string¥%string
```

- 大/小文字の区別がある名前、DBCS 名、複数語表名および複数語ビュー名は、円記号および二重引用符の両方で囲む必要があります。以下に例を示します。

```
¥"My Table¥"
```

マルチバイト文字セット (MBCS) 名または 2 バイト文字セット (DBCS) 名が円記号と二重引用符の区切り文字で囲まれていない場合、小文字と同じバイトが含

まれていると大文字に変換され、**db2look** は変換された名前を使用してデータベース・オブジェクトを探します。その結果、DDL ステートメントは抽出されません。

- **-tw** オプションは、**-x** オプション (GRANT 特権を生成する場合)、**-m** オプション (表統計および列統計を戻す場合)、および **-l** オプション (ユーザー定義表スペース、データベース・パーティション・グループ、およびバッファー・プールの DDL を生成する場合) とともに使用できます。 **-t** オプションを **-tw** オプションとともに指定すると、**-t** オプション (およびそれに関連した *Tname* 引数) が無視されます。
- フェデレーテッド・データ・ソース上、または DB2 Universal Database for z/OS and OS/390、DB2 for i、または DB2 Server for VSE & VM 上にある表 (およびそれらに関連したオブジェクト) の DDL を生成するために **-tw** オプションを使用することはできません。
- **-tw** オプションは、CLP でのみサポートされています。

パーティション・データベース環境のあるシステムで DDL ステートメントを生成しようとする、非アクティブ・データベース・パーティション上にある表スペースに関しては、DDL ステートメントの代わりに警告メッセージが表示されます。すべての表スペースに対して正しい DDL ステートメントが作成されるようにするため、すべてのデータベース・パーティションをアクティブにする必要があります。

DB2 クライアントから、クライアントと同じかそれ以降のリリースのデータベースには **db2look** コマンドを発行できますが、クライアントからクライアントより前のリリースのデータベースにはこのコマンドは発行できません。例えば、**db2look** コマンドをバージョン 9.8 クライアントからバージョン 10.1 データベースに発行することはできますが、このコマンドをバージョン 10.1 クライアントからバージョン 9.8 データベースに発行することはできません。

**db2look** コーティリティーを呼び出すと、**db2look** コマンドによって、非コミット・トランザクションを使用して作成されたオブジェクトに対して DDL ステートメントが生成されます。

タイプ配列のセキュリティー・ラベル・コンポーネントのための DDL ステートメントを抽出する場合、抽出されたステートメントが生成するコンポーネントの内部表記 (その配列内のエレメントのエンコード) が、**db2look** コマンドの発行対象のデータベース内のコンポーネントの内部表記と一致しないことがあります。この不一致は、セキュリティー・ラベル・コンポーネントに 1 つ以上のエレメントを追加することによってコンポーネントを変更した場合に発生する可能性があります。そのような場合には、ある表から抽出されて **db2look** 出力から作成した別の表に移動したデータには、対応するセキュリティー・ラベル値がないので、新しい表の保護が危険にさらされる可能性があります。

#### 関連情報:

ニックネーム列および索引の名前

マイグレーションのためのアプリケーションの変更

## db2ls - インストール済みの DB2 製品およびフィーチャーのリスト

Linux および UNIX システムにインストールされているDB2 製品およびフィーチャーをリストします。これには、DB2 HTML 文書が含まれます。

システム上に DB2 製品の複数コピーをインストールする機能と、選択したパスに DB2 製品およびフィーチャーをインストールする柔軟性を活用し、**db2ls** コマンドを使用して、以下のものをリストすることができます。

- DB2 製品がシステム上のどこにインストールされているか、および DB2 製品レベルをリストする。
- 特定のインストール・パスにあるすべての、または特定の DB2 製品およびフィーチャー。

**db2ls** コマンドは、インストール・メディアとシステム上の DB2 インストール・コピーの両方にあります。**db2ls** コマンドはこのどちらかの場所から実行できます。**db2ls** コマンドは、IBM Data Server Driver Package を除くすべての製品のインストール・メディアから実行できます。

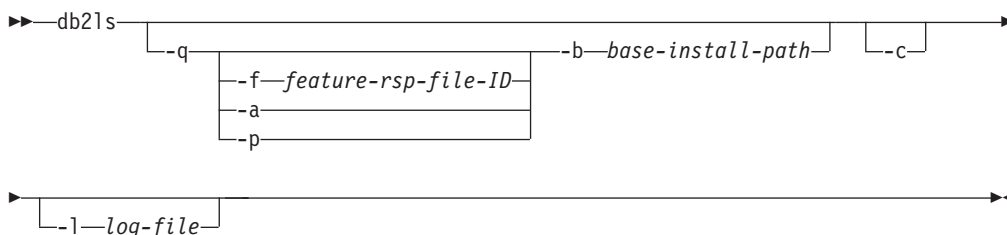
### 許可

なし

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

**-q** 照会がインストール済みの DB2 製品およびフィーチャーをリストするためであることを示します。デフォルトでは、**-a** パラメーターも指定されているのでなければ、可視のコンポーネント (フィーチャー) のみ表示されません。

#### **-f** *feature-rsp-file-ID*

特定のフィーチャーがインストールされている場合、それを照会します。インストールされていない場合、プログラムからの戻りコードはゼロ以外です。その他の場合は、戻りコードはゼロです。

**-a** 可視のフィーチャーだけでなく、非表示のコンポーネントもすべてをリストします。デフォルトでは、**db2ls** コマンドは、可視のフィーチャーのみをリストします。

## db2ls - インストール済みの DB2 製品およびフィーチャーのリスト

**-p** 製品のみをリストします。フィーチャーをリストするのではなく、お客様がインストールした製品の簡潔なリストを提供します。

**-b** *base-install-path*

`/usr/local/bin` でグローバル **db2ls** コマンドを使用する場合、照会するディレクトリーを指定する必要があります。グローバル **db2ls** コマンドは、単にそのインストール・パスから **db2ls** コマンドを呼び出し、パラメーターの残りを渡すだけです。

**-c** 項目の列ベース・リストではなく、コロン区切りリストとして出力を印刷します。これにより、情報をプログラマチックに処理することができます。出力の最初の行は、各項目を説明するトークンのコロン区切りリストになります。この最初の行はハッシュ文字 ("#") で始まり、プログラマチックに無視するのを容易にします。

**-l** *log-file*

デバッグの目的で使用するために、ログ・ファイルをトレースします。

### 例

• 特定のパスにインストールされている DB2 データベース・フィーチャーを照会するには、以下のコマンドを発行します。

```
db2ls -q -b /opt/ibm/ese/V10.1
```

• 特定のパスにインストールされたすべての DB2 データベース・フィーチャーを見るには、以下のコマンドを発行します。

```
db2ls -q -a -b /opt/ibm/ese/V10.1
```

• 特定の DB2 データベース・フィーチャーがインストールされているかどうかを調べるには、以下のコマンドを発行します。

```
db2ls -q -b /opt/ibm/ese/V10.1 -f feature
```

### 使用上の注意

- Windows オペレーティング・システムでは、**db2ls** コマンドを使用できません。
- `root` が `/usr/local/bin` への書き込み権限を持っているか、または `/usr/local/bin` を作成できる場合には、DB2 バージョン 9 以降を初めてシステムにインストールする際に `DB2DIR/install/db2ls` を指すシンボリック・リンク `/usr/local/bin/db2ls` が作成されます。複数の DB2 コピーをインストールする場合は、ルートはシステムにインストールされる DB2 のうち最高のバージョンおよびレベルを指すリンクを更新します。

非 `root` インストールでは、`/usr/local/bin/db2ls` の作成や変更は行われません。この場合、**db2ls** を実行するには、以下の 2 つのうちいずれかを行う必要があります。

- ユーザーのパスに `inst_home/sqlllib/install` を追加します。その後、非ルート・ユーザーとして **db2ls** を実行できます。
- `inst_home/sqlllib/install/db2ls` のような、コマンドの正確なパスを渡します。
- バージョン 9 以降では、**db2ls** コマンドは、DB2 製品を照会するための唯一の方法です。Linux または UNIX オペレーティング・システム固有のユーティリティー (`pkgadd`、`rpm`、`SMIT`、または `swinstall` など) を使用して DB2 製品を照会

## db2ls - インストール済みの DB2 製品およびフィーチャーのリスト

することはできません。DB2 インストール環境とのインターフェースや照会に使用する既存のスクリプトで、固有のインストール・ユーティリティーを含むものは、変更する必要があります。

- 入手するフィーチャーのリストは、DB2 のインストール方法がルートか非ルートかによっても違いますし、コマンドを実行するユーザーによっても違います。
  - q オプションを指定しないと、以下のようになります。
    - 非 root インストール・インスタンス・ユーザー以外のユーザーの場合、このコマンドはルート・ユーザーがインストールしたコピーをすべて表示します。
    - 非 root インストール・インスタンス・ユーザーの場合、このコマンドはルート・ユーザーがインストールしたすべての DB2 コピーと、非ルート・ユーザーが所有する非ルート・コピーを表示します。
  - q オプションを指定すると、以下のようになります。
    - userB が DB2 をインストールしたかどうかを userA が知りたい場合、userA は `db2ls -q -b $userBHomeDir/sqllib` を実行できます。userA にアクセス権限がある場合は、userB がインストールした DB2 フィーチャーが表示されます。ない場合は、アクセス権限が拒否されたことを示すエラー・メッセージが戻されます。
    - -b オプションを指定しないで `db2ls -q` を実行する場合、**db2ls** が属しているインストール・パス中にインストールされているフィーチャーが表示されます。
- ディレクトリーが読み取り専用の場合は、**db2ls** コマンドに `/usr/local/bin` ディレクトリーからリンクできません。システム・ワークロード・パーティション (WPAR) で実行している場合、インストール・イメージ・ルート・ディレクトリーにある **db2ls** コマンドを使用して、インストールされたコピーのリストを照会できます。



## db2move - データベース移動ツール

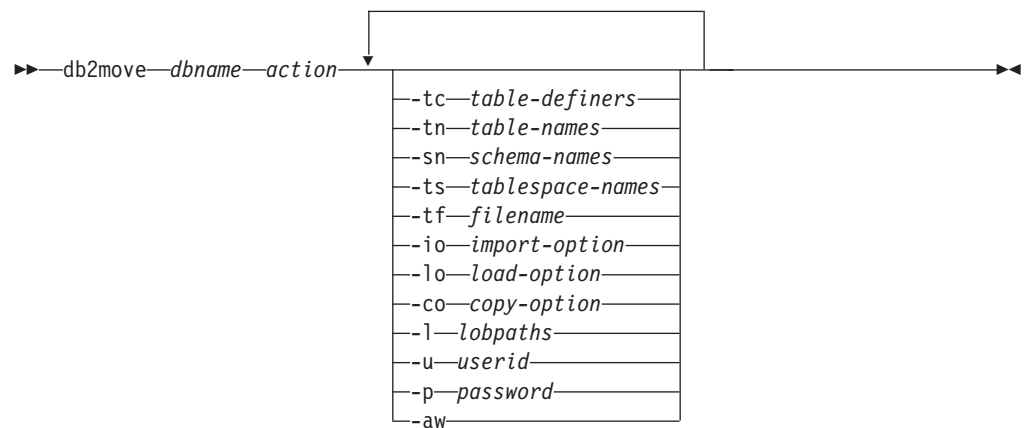
このツールを EXPORT、IMPORT、LOAD いずれかのモードで使用すると、ワークステーション上にある DB2 データベース間で、大量の表を簡単に移動することができます。COPY モードで使用すると、このツールによってスキーマの複写が容易になります。

また、特定のデータベースのシステム・カタログ表を照会し、すべてのユーザー表のリストをコンパイルします。そして、これらの表を PC/IXF フォーマットでエクスポートします。PC/IXF ファイルは、同じシステム上の別のローカル DB2 データベースにインポートまたはロードするか、または別のワークステーション・プラットフォームに転送し、そのプラットフォームで DB2 データベースにインポートまたはロードすることができます。構造化タイプ列がある表は、このツールを使用しても移動しません。

### 許可

このツールは、ユーザーから要求されるアクションにしたがって、DB2 エクスポート、インポート、およびロード API を呼び出します。したがって、要求元ユーザー ID には、これらの API に求められる正しい権限がなければなりません。この権限がないと、要求は失敗します。

### コマンド構文



### コマンド・パラメーター

*dbname*

データベースの名前。

*action* 以下のうちの 1 つでなければなりません。

#### EXPORT

*options* のフィルター基準を満たすすべての表をエクスポートします。*options* の指定がない場合には、すべての表をエクスポートします。内部ステージング情報は db2move.lst ファイルに保管されます。

### IMPORT

内部ステージング・ファイル `db2move.lst` にリストされているすべての表をインポートします。IMPORT の特定のアクションには、**-io** オプションを使用します。

**LOAD** 内部ステージング・ファイル `db2move.lst` にリストされているすべての表をロードします。LOAD の特定のアクションには、**-lo** オプションを使用します。

**COPY** ターゲット・データベースにスキーマを複製します。ターゲット・データベースはローカル・データベースでなければなりません。**-sn** オプションを使用して、1 つ以上のスキーマを指定します。**COPY** 特定オプションの **-co** オプションを参照してください。**-tn** または **-tf** オプションを使用して、LOAD\_ONLY モードの表をフィルターします。ADMIN\_COPY\_SCHEMA() ストアド・プロシージャを使用する場合も、**-COPY** オプションを指定した **db2move** ユーティリティを使用する場合も、SYSTOOLSPACE という名前の表スペースを使用する必要があります。

各アクション中に生成されるファイルのリストについては、下記のセクションを参照してください。

#### **-tc** *table-definers*

デフォルトはすべての定義者です。

これは EXPORT アクションのみです。指定されると、このオプションでリストされる定義者が作成する表のみがエクスポートされます。指定されない場合、デフォルトではすべての定義者を使用します。複数の定義者を指定する場合、それぞれをコンマで区切る必要があります。定義者 ID 間にブランクを入れることはできません。このオプションを **-tn table-names** オプションとともに使用すると、エクスポートする表を選択できます。

アスタリスク (\*) は、ストリング中のどこにでも入れられるワイルドカード文字として使用できます。

#### **-tn** *table-names*

デフォルトはすべてのユーザー表です。

これは EXPORT または COPY アクションのみです。

EXPORT アクションで指定されると、指定されたストリング内の表と名前が一致する表だけがエクスポートされます。指定されない場合、デフォルトではすべてのユーザー表を使用します。複数の表名を指定する場合、それぞれをコンマで区切る必要があります。表名間にブランクを入れることはできません。スキーマをフィルター操作するには、修飾なしで表名をリストし、**-sn** オプションを使用する必要があります。

エクスポートの場合、アスタリスク (\*) は、ストリング中のどこにでも入れられるワイルドカード文字として使用できます。

COPY アクションで指定する場合、これに加えて **-co "MODE" LOAD\_ONLY copy-option** も指定する必要があります。指定された表のデータだけがターゲット・データベースで再挿入されます。表名は、スキーマ修飾子と共に "schema"."table" の形式でリストします。



**-sn** *schema-names*

EXPORT のデフォルトは全スキーマです (COPY ではない)。

これが指定されると、一致するスキーマ名の表だけがエクスポートまたはコピーされます。複数のスキーマ名を指定する場合は、それぞれの名前をコンマで区切る必要があります。複数のスキーマ名の間にブランクを使用することはできません。8 文字より短いスキーマ名は、8 文字の長さになるまで埋め込まれます。

エクスポートの場合: スキーマ名の部分にアスタリスク・ワイルドカード文字 (\*) が使用された場合は、それがパーセント記号 (%) に変更され、WHERE 節の LIKE 述部にパーセント付きの表名が使用されます。指定されない場合、デフォルトではすべてのスキーマを使用します。-tn または -tc オプションと合わせて使用する場合、db2move は、スキーマが指定されたスキーマ名と一致し、定義者が指定された定義者と一致する表に対してのみ実行されます。fred のようなスキーマ名の場合、アスタリスクを使用するときは、-sn fr\*d ではなく -sn fr\*d\* のような指定が必要になります。

**-ts** *tablespace-names*

デフォルトはすべての表スペースです。

これは EXPORT アクションのみです。このオプションが指定されると、指定した表スペースにある表だけがエクスポートされます。表スペース名の部分にアスタリスク・ワイルドカード文字 (\*) が使用された場合は、それがパーセント記号 (%) に変更され、WHERE 節の LIKE 述部にパーセント付きの表名が使用されます。-ts オプションが指定されない場合、デフォルトではすべての表スペースを使用します。複数の表スペース名を指定する場合は、それぞれの名前をコンマで区切る必要があります。複数の表スペース名の間にブランクを使用することはできません。8 文字より短い表スペース名は、8 文字の長さになるまで埋め込まれます。例えば、mytb のような表スペース名の場合、アスタリスクを使用するときは、-sn my\*b ではなく -ts my\*b\* のような指定が必要になります。

**-tf** *filename*

EXPORT アクションで指定されると、指定されたファイル内の名前と正確に一致する名前の表だけがエクスポートされます。指定されない場合、デフォルトではすべてのユーザー表を使用します。表は 1 行に 1 つずつリストする必要がある、各表は完全に修飾する必要があります。ストリング内には、ワイルドカード文字を使用できません。以下は、ファイルの内容の例です。

```
"SCHEMA1"."TABLE NAME1"  
"SCHEMA NAME77"."TABLE155"
```

COPY アクションで指定する場合、これに加えて -co "MODE" LOAD\_ONLY copy-option も指定する必要があります。ファイル内に指定された表のデータだけが、ターゲット・データベースで再挿入されます。表名は、スキーマ修飾子と共に "schema"."table" の形式でリストします。

**-io** *import-option*

デフォルトは REPLACE\_CREATE です。インポート作成機能の制限については、『IMPORT コマンドの推奨されなくなったオプション CREATE および REPLACE\_CREATE』を参照してください。

有効なオプションは、INSERT、INSERT\_UPDATE、REPLACE、CREATE、および REPLACE\_CREATE です。

### **-lo** *load-option*

デフォルトは INSERT です。

有効なオプションは、INSERT および REPLACE です。

### **-co** **db2move** アクションが COPY である場合、以下の **-co** 追加オプションを使用できます。

**"TARGET\_DB db name [USER userid USING password]"**

ユーザーがターゲット・データベースの名前、ユーザー ID、およびパスワードを指定できるようにします。(ソース・データベースのユーザー ID およびパスワードは、既存の **-p** および **-u** オプションを使用して指定できます。) **USER USING** 節はオプションです。

**USER** が **userid** を指定する場合は、**USING** 節の後にパスワードを指定します。パスワードが指定されない場合、**db2move** はパスワード情報を求めるプロンプトを出します。プロンプトが出されるのは、下記のセクションで説明するセキュリティ上の理由によります。

**TARGET\_DB** は COPY アクションには必須のオプションです。

**TARGET\_DB** は、ソース・データベースと同じにすることはできませんし、ローカル・データベースでなければなりません。同じデータベース内のスキーマをコピーするには、**ADMIN\_COPY\_SCHEMA** プロシージャを使用できます。COPY アクションには、少なくとも 1 つのスキーマ (**-sn**) または 1 つの表 (**-tn** または **-tf**) の入力が必要です。

複数の **db2move** コマンドを実行してスキーマを 1 つのデータベースから別のデータベースにコピーすると、デッドロックになります。一度に 1 つのみの **db2move** コマンドを発行してください。コピー処理中にソース・スキーマ内の表を変更すると、ターゲット・スキーマのデータがコピー後に同一のものにならないことがあります。

## **"MODE"**

### **DDL\_AND\_LOAD**

ソース・スキーマの、すべてのサポートされるオブジェクトを作成し、ソース表データを表に追加します。これはデフォルト・オプションです。

### **DDL\_ONLY**

ソース・スキーマの、すべてのサポートされるオブジェクトを作成しますが、表にデータを再設定しません。

### **LOAD\_ONLY**

指定されたすべての表をソース・データベースからターゲット・データベースへロードします。表はターゲットに既に存在していなければなりません。**LOAD\_ONLY** モードでは、**-tn** または **-tf** オプションを使って少なくとも 1 つの表を入力する必要があります。

これは、COPY アクションでのみ使用される任意指定のオプションです。

#### "SCHEMA\_MAP"

ターゲットへコピーするときにユーザーがスキーマをリネームできるようにします。ソース・ターゲット間のスキーマ・マッピングをコンマで区切り、大括弧で囲んだリストを提供します。例えば、`schema_map ((s1, t1), (s2, t2))` のようになります。これは、スキーマ `s1` からのオブジェクトはターゲットのスキーマ `t1` にコピーされ、スキーマ `s2` からのオブジェクトはターゲットのスキーマ `t2` へコピーされることを意味します。ターゲット・スキーマ名がソース・スキーマ名であるのがデフォルトで、推奨されています。この理由は、**db2move** がオブジェクト本体内に修飾オブジェクトのあるスキーマを変更しないことにあります。したがって、異なるターゲット・スキーマ名を使用すると、オブジェクト本体内に修飾オブジェクトがある場合に問題が生じるおそれがあります。

以下に例を示します。`create view F00.v1 as 'select c1 from F00.t1'`

この場合、スキーマ `FOO` の `BAR` へのコピー、`v1` は以下のように再生成されます。`create view BAR.v1 as 'select c1 from F00.t1'`

これは、スキーマ `FOO` がターゲット・データベースに存在しないため失敗するか、または `FOO` が `BAR` と異なるために予期しない結果になります。ソースと同じスキーマ名を保つことにより、これらの問題を避けることができます。スキーマ間に相互従属関係がある場合、すべての相互に従属するスキーマがコピーされなければなりません。あるいは、相互従属関係のあるオブジェクトのコピーでエラーになります。

以下に例を示します。`create view F00.v1 as 'select c1 from BAR.t1'`

この場合、`v1` のコピーは `BAR` がコピーされない場合に失敗するか、または、ターゲットの `BAR` がソースからの `BAR` と異なる場合、予期しない結果になります。**db2move** はスキーマの相互従属関係を検出しようとはしません。

これは、COPY アクションでのみ使用される任意指定のオプションです。

ターゲット・スキーマが既に存在する場合には、ユーティリティは失敗します。 `ADMIN_DROP_SCHEMA` プロシージャを使用して、スキーマと、そのスキーマに関連付けられたすべてのオブジェクトをドロップしてください。

#### "NONRECOVERABLE"

このオプションにより、ユーザーはロードのデフォルト動作をオーバーライドし、ロードが `COPY-NO` で行われるようにすることができます。デフォルトの動作では、ユーザーはロードされる各表スペースをバックアップするよう強制されます。この **NONRECOVERABLE** キーワードを指定すると、ユーザーは表スペースをバックアップするように即時に強制されることがありません。ただし、新しく作成

された表が正しくリカバリーできるように、できるだけ早くバックアップを取ることを強くお勧めします。これは、COPY アクションで使用できる任意指定のオプションです。

### "OWNER"

正常にコピーした後、ターゲット・スキーマに作成された各新規オブジェクトの所有者をユーザーが変更できるようにします。ターゲット・オブジェクトのデフォルト所有者は接続ユーザーになりますが、このオプションが指定された場合、所有権は新規所有者に移されます。これは、COPY アクションで使用できる任意指定のオプションです。

### "TABLESPACE\_MAP"

ユーザーは、コピー中に使用する表スペース名のマッピングを、ソース・システムの表スペースの代わりに指定できます。これは、大括弧で囲まれた表スペース・マッピングが配列されたものです。例えば、`tablespace_map ((TS1, TS2),(TS3, TS4))` のようにします。これは、表スペース TS1 からのすべてのオブジェクトはターゲット・データベースの表スペース TS2 にコピーされ、表スペース TS3 からのオブジェクトはターゲットの表スペース TS4 へコピーされることを意味します。((T1, T2),(T2, T3)) の場合、ソース・データベースの T1 にあるすべてのオブジェクトはターゲット・データベースの T2 に再作成され、ソース・データベースの T2 にあるどのオブジェクトもターゲット・データベースの T3 に再作成されることとなります。デフォルトでは、ソースの表スペース名と同じ表スペース名を使用しますが、その場合には、表スペースのマッピング入力が必要ありません。指定された表スペースが存在しない場合、その表スペースを使用したオブジェクトのコピーは失敗し、エラー・ファイルにログされます。

ユーザーには、`SYS_ANY` キーワードを使用して、ターゲット表スペースの選択にデフォルトの表スペース選択アルゴリズムの使用を指定するオプションもあります。この場合、**db2move** は使用できる表スペースをどれでもターゲットとしての使用に選択することができます。`SYS_ANY` キーワードはすべての表スペースに対して使用できます。例えば、`tablespace_map SYS_ANY` とします。さらに、ユーザーは特定のマッピングを表スペースのいくつかに指定し、残りにデフォルトの表スペース選択アルゴリズムを指定することもできます。例えば、`tablespace_map ((TS1, TS2),(TS3, TS4), SYS_ANY)` のようにします。これは、表スペース TS1 は TS2 に、TS3 は TS4 にマップされるが、残った表スペースはデフォルトの表スペース・ターゲットを使用することを意味します。「SYS」で始まる表スペースはあり得ないため、`SYS_ANY` キーワードが使用されます。これは、COPY アクションで使用できる任意指定のオプションです。

### "PARALLEL" *number-of-threads*

このオプションを指定すると、スキーマ内の表のロード操作がいくつかのスレッドに分散されます。*number-of-threads* の値の範囲は 0 から 16 です。

- PARALLEL を指定しない場合、スレッドは使用されず、ロード操作は逐次実行されます。
  - スレッド数を設定しないで PARALLEL を指定した場合、**db2move** ユーティリティーによって適切な値が選択されます。
  - *number-of-threads* を設定して PARALLEL を指定した場合、指定した数のスレッドが使用されます。*number-of-threads* が 0 または 1 である場合、ロード操作は逐次実行されます。
  - *number-of-threads* に指定できる最大値は 16 です。
- これは、COPY アクションで使用できる任意指定のオプションです。

#### -l lobpaths

IMPORT および EXPORT の場合、このオプションが指定されると、これは XML パスにも使用されます。デフォルトは、現行ディレクトリです。

このオプションは、LOB または XML ファイルが (EXPORT の一部として) 作成されるか、または (IMPORT または LOAD の一部として) 検索される絶対パス名を指定します。複数のパスを指定する場合、それぞれをコンマで区切る必要があります。パス間に空白を入れることはできません。複数のパスが指定された場合、EXPORT はラウンドロビン方式でそれらを使用します。つまり、1 つの LOB 文書を最初のパスに書き込み、それから 2 番目のパスに、という順に最後まで書き込み、その後最初のパスに戻ります。XML 文書でも同じです。最初のパスでファイルが見つからない場合 (IMPORT または LOAD 中)、2 番目のパスが使用される、という方法でパスが使用されます。

#### -u userid

デフォルトはログオン・ユーザー ID です。

ユーザー ID とパスワードはどちらも任意指定です。しかし、一方を指定した場合、他方も必ず指定する必要があります。コマンドがリモート・サーバーに接続するクライアント上で実行される場合、ユーザー ID とパスワードを指定する必要があります。

#### -p password

デフォルトはログオン・パスワードです。ユーザー ID とパスワードはどちらも任意指定です。しかし、一方を指定した場合、他方も必ず指定する必要があります。-p オプションが指定されてもパスワードが指定されていない場合、**db2move** はパスワードを求めるプロンプトを出します。これは、セキュリティの理由によります。コマンド行にパスワードを入力するとセキュリティ問題が生じます。例えば、**ps -ef** コマンドがパスワードを表示します。しかし、**db2move** がスクリプトを通して呼び出される場合は、パスワードを供給する必要があります。コマンドがリモート・サーバーに接続するクライアント上で発行される場合、ユーザー ID とパスワードを指定する必要があります。

#### -aw

警告を許します。-aw が指定されていない場合、エクスポート中に警告があった表は **db2move.lst** ファイルに組み込まれません (表の .ixf ファイルと .msg ファイルが生成されている)。しかし、あるシナリオ (データ切り捨てなど) では、そのように警告があった表でも **db2move.lst** ファイルに



組み込んでしまいたい場合があります。そのようなとき、このオプションを指定すると、エクスポート中に警告を受け取った表を .lst ファイルに組み込むことができます。

### 例

- SAMPLE データベースのすべての表をエクスポートするには (すべてのオプションにデフォルト値を使用)、以下を発行します。

```
db2move sample export
```

- userid1 または us%rid2 のようなユーザー ID で作成され、tbname1 という名前、または %tbname2 のような表名を持つすべての表をエクスポートするには、以下を発行します。

```
db2move sample export -tc userid1,us*rid2 -tn tbname1,*tbname2
```

- すべての表を SAMPLE データベースにインポートするには、以下を発行します (LOB パス D:¥LOBPATH1 および C:¥LOBPATH2 で LOB ファイルが検索されます。この例は、Windows オペレーティング・システムにのみ該当します)。

```
db2move sample import -l D:¥LOBPATH1,C:¥LOBPATH2
```

- SAMPLE データベースのすべての表をロードするには、以下を発行します。 (/home/userid/lobpath サブディレクトリーと tmp サブディレクトリーで、LOB ファイルが検索されます。この例は Linux および UNIX システムにのみ該当します)。

```
db2move sample load -l /home/userid/lobpath,/tmp
```

- SAMPLE データベースのすべての表を、指定されたユーザー ID およびパスワードを使用して REPLACE モードでインポートするには、以下を発行します。

```
db2move sample import -io replace -u userid -p password
```

- スキーマ schema1 をソース・データベース dbsrc からターゲット・データベース dbtgt へ複写するには、以下を発行します。

```
db2move dbsrc COPY -sn schema1 -co TARGET_DB dbtgt USER myuser1 USING mypass1
```

- スキーマ schema1 をソース・データベース dbsrc からターゲット・データベース dbtgt へ複写し、そのターゲット上でスキーマを newschema1 に名前変更し、ソース表スペース ts1 をターゲットの ts2 へマップするには、以下を発行します。

```
db2move dbsrc COPY -sn schema1 -co TARGET_DB dbtgt USER myuser1 USING mypass1  
SCHEMA_MAP ((schema1,newschema1)) TABLESPACE_MAP ((ts1,ts2), SYS_ANY))
```

### 使用上の注意

- 1 つ以上のスキーマをターゲット・データベースにコピーする場合、各スキーマは、互いに依存しないものである必要があります。そうでない場合、オブジェクトの一部がターゲット・データベースに正常にコピーされない可能性があります。
- XML 列を含む表へのデータのロードは、LOAD 操作でのみサポートされており、COPY 操作ではサポートされていません。回避策は、手動で IMPORT または EXPORT コマンドを実行するか、db2move Export および db2move Import 動作を使用することです。それらの表に GENERATED ALWAYS ID 列も含まれている場合は、表にデータをインポートできません。
- db2move EXPORT に続いて db2move IMPORT または db2move LOAD を指定すると、簡単に表データを移動できます。表に関連した他のすべてのデータベース・オブ

ジェクト (別名、ビュー、トリガーなど)、およびこれらの表が依存するオブジェクト (ユーザー定義タイプ、ユーザー定義関数など) を手動で移動する必要があります。

- **CREATE** または **REPLACE\_CREATE** オプションを指定した **IMPORT** アクションを使ってターゲット・データベース上に表を作成する場合 (どちらのオプションも非推奨になり、今後のリリースで除去される可能性があります)、『インポート済みの表の再作成』で説明されている制約が適用されます。 **REPLACE\_CREATE** オプションの使用時の **db2move** インポート・フェーズ中に想定外のエラーが生じた場合、該当する **tabnnn.msg** メッセージ・ファイルを調べて、表の作成に対する制限事項が原因でエラーが起きたかどうかを確かめてください。
- **db2move** を使用して、ID 列 **GENERATED ALWAYS** を含む表をインポートまたはロードすることはできません。ただし、手動でこれらの表をインポートまたはロードすることは可能です。詳しくは、『ID 列のロードに関する考慮事項』または『ID 列のインポートに関する考慮事項』を参照してください。
- エクスポート、インポート、またはロード API が **db2move** によって呼び出されると、**FileTypeMod** パラメーターが **lobsinfile** に設定されます。つまり、LOB データが各表に対して、**PC/IXF** ファイルとは別のファイルに保持されます。
- **LOAD** コマンドは、データベースおよびデータ・ファイルが常駐するマシンでローカルに実行する必要があります。
- **db2move LOAD** を使用する場合、データベースの **logarchmeth1** が使用可能 (データベースがリカバリー可能) であれば、次のようになります。
  - **NONRECOVERABLE** オプションが指定されない場合、**db2move** はデフォルトの **COPY NO** オプションを使って **db2Load** API を呼び出します。ロードされた表が格納される表スペースは、ユーティリティ完了時にバックアップ・ペンディング状態に置かれます (表スペースをバックアップ・ペンディング状態から解除するには、データベース全体または表スペース全体のバックアップが必要です)。
  - **NONRECOVERABLE** オプションが指定されている場合、表スペースはバックアップ・ペンディング状態に置かれませんが、ロールフォワード・リカバリーが後で実行された場合、表はアクセス不能とマーク付けられるため、表をドロップする必要があります。ロード・リカバリー可能性オプションの詳細については、『ロードのパフォーマンスを改善するためのオプション』を参照してください。
- **IMPORT** または **LOAD** アクションを使用する **db2move** コマンドのパフォーマンスは、デフォルトのバッファ・プール **IBMDEFAULTBP** を変更し、構成パラメーター **sortheap**、**util\_heap\_sz**、**logfilsiz**、および **logprimary** を更新することによって、改善できます。
- **export** や **db2move** などのデータ移動ユーティリティを実行する際に、照会コンパイラーが、基礎照会を基本表よりも **MQT** に対して実行する方が効率が良いと判別することがあります。そのような場合、照会は据え置きリフレッシュの **MQT** に対して実行され、ユーティリティの結果は基礎表内のデータを正確に表していない可能性があります。
- **db2move** コマンドは、DB2 クライアントでは使用できません。クライアント・マシンから **db2move** コマンドを発行すると、エラー・メッセージ「db2move は、内部コマンドまたは外部コマンド、 操作可能なプログラムまたはバッチ ファイル

として認識されていません。」を受け取ります。この問題を避けるために、サーバー上で直接 **db2move** コマンドを発行できます。

### EXPORT 使用時に必要とされるファイル/生成されるファイル

- 入力: なし。
- 出力:

#### **EXPORT.out**

EXPORT アクションの結果の要約。

#### **db2move.lst**

オリジナル表名のリスト、その対応する PC/IXF ファイル名 (tabnnn.ixf)、およびメッセージ・ファイル名 (tabnnn.msg)。このリスト、エクスポートされた PC/IXF ファイル、および LOB ファイル (tabnnnc.yyy) は、**db2move** IMPORT または LOAD アクションへの入力として使用されます。

#### **tabnnn.ixf**

特定の表の、エクスポートされる PC/IXF ファイル。

#### **tabnnn.msg**

対応する表のエクスポート・メッセージ・ファイル。

#### **tabnnnc.yyy**

特定の表の、エクスポートされる LOB ファイル。

*nnn* は表番号です。*c* はアルファベットの文字です。*yyy* は 001 から 999 の範囲内の数値です。

これらのファイルは、エクスポートされている表に LOB データが入っている場合のみ作成されます。作成されると、これらの LOB ファイルは *lobpath* ディレクトリーに入れられます。LOB ファイルには、合計 26,000 の可能な名前があります。

#### **system.msg**

ファイルまたはディレクトリー・コマンドを作成または削除するための、システム・メッセージの入ったメッセージ・ファイル。これは、アクションが EXPORT で、LOB パスが指定される場合のみ使用されます。

### IMPORT 使用時に必要とされるファイル/生成されるファイル

- 入力:

#### **db2move.lst**

EXPORT アクションからの出力ファイル。

#### **tabnnn.ixf**

EXPORT アクションからの出力ファイル。

#### **tabnnnc.yyy**

EXPORT アクションからの出力ファイル。

- 出力:

#### **IMPORT.out**

IMPORT アクションの結果の要約。



**tabnnn.msg**

対応する表のインポート・メッセージ・ファイル。

**LOAD 使用時に必要とされるファイル/生成されるファイル**

- 入力:

**db2move.lst**

EXPORT アクションからの出力ファイル。

**tabnnn.ixf**

EXPORT アクションからの出力ファイル。

**tabnnnc.yyy**

EXPORT アクションからの出力ファイル。

- 出力:

**LOAD.out**

LOAD アクションの結果の要約。

**tabnnn.msg**

対応する表の **LOAD** メッセージ・ファイル。

**COPY 使用時に必要とされるファイル/生成されるファイル**

- 入力: なし

- 出力:

**COPYSCHEMA.msg**

COPY 操作中に生成されたメッセージを含む出力ファイル。

**COPYSCHEMA.err**

COPY 操作中に発生した各エラーに関するエラー・メッセージが含まれる出力ファイル。これには、ターゲット・データベース上に再作成できなかった各オブジェクトに関する DDL ステートメントが含まれます。

**LOADTABLE.msg**

ロード・ユーティリティのそれぞれの呼び出しによって生成されたメッセージが含まれる出力ファイル (ターゲット・データベースでのデータ再挿入に使用されます)。

**LOADTABLE.err**

ロード中に失敗した表の名前、またはターゲット・データベースにまだデータ挿入する必要がある表の名前が含まれる出力ファイル。詳しくは『スキーマのコピー操作が失敗した場合の再開方法』のトピックを参照してください。

これらのファイルは、タイム・スタンプされ、1 つの実行から生成されたすべてのファイルには同一のタイム・スタンプが付きます。

### db2mq1sn - MQListener

非同期 MQListener を呼び出します。この MQListener は、一群の WebSphere MQ メッセージ・キューをモニターして、キューに着信したメッセージを構成済みの DB2 ストアード・プロシージャーに渡します。また、関連する管理タスクや構成タスクを実行することもできます。

MQListener 構成情報は DB2 データベースに保管されており、その構成情報の中には一群の名前付きの構成とデフォルトが含まれています。それぞれの構成の中身は、一群のタスクです。MQListener タスクは、メッセージの検索元になるメッセージ・キューと、メッセージの引き渡し先になるストアード・プロシージャーによって定義します。メッセージ・キューの記述には、メッセージ・キューの名前と、キュー・マネージャーがデフォルトでなければそのキュー・マネージャーを含める必要があります。ストアード・プロシージャーの情報には、ストアード・プロシージャーの定義が入っているデータベース、そのデータベースにアクセスするためのユーザー名とパスワード、プロシージャーの名前とスキーマを含める必要があります。

Linux および UNIX オペレーティング・システムでは、このユーティリティーは *DB2DIR/instance* ディレクトリーにあります。*DB2DIR* は、DB2 データベース製品の現行バージョンがインストールされている場所です。

Windows オペレーティング・システムでは、このユーティリティーは *DB2PATH%sql1lib%bin* ディレクトリーにあります。*DB2PATH* は、DB2 データベース製品の現行バージョンがインストールされている場所です。

WebSphere MQ オブジェクトのアクセス制御の詳細については、「*WebSphere MQ システム管理 (SC88-9239-00)*」を参照してください。

#### 許可

- **db2mq1sn admin** 以外のすべてのオプションは、*configDB* データベース内の MQListener 構成にアクセスします。*configUser* として接続するか、ユーザーを指定しない場合は暗黙接続を試みることになります。接続に使用するユーザー名は、パッケージ *mq1Confi* に対する EXECUTE 特権を持っている必要があります。
- **db2mq1sn run** オプションと **db2mq1sn admin** オプションで MQ オブジェクトにアクセスする場合、プログラムを実行するユーザーは、該当する MQ オブジェクトを開く権限を持っている必要があります。
- **db2mq1sn run** オプションを正常に実行するには、そのタスクを作成する **db2mq1sn add** オプションに指定する *dbUser* が、指定のストアード・プロシージャーに対する EXECUTE 特権と、*dbName* データベース内のパッケージ *mq1Run* に対する EXECUTE 特権を持っている必要があります。

#### コマンド構文



**-configDB** *configuration database*

構成情報が入っているデータベースの名前。

**-configUser** *user ID* **-configPwd** *password*

構成データベースにアクセスするための権限情報。

**-config** *configuration name*

個々のタスクを 1 つの構成にグループ化できます。そうすれば、それらのタスクを 1 つのグループとしてまとめて実行できます。構成の名前を指定しない場合、このユーティリティはデフォルト構成を実行します。

### run

**-adminQueue** *admin queue name* **-adminQMGr** *admin queue manager*

これは、MQListener が管理コマンドを聴取しようとしているキューです。キュー・マネージャーの名前を指定しない場合、このユーティリティは構成済みのデフォルト・キュー・マネージャーを使用します。adminQueue を指定しない場合、このアプリケーションはメッセージ・キューから管理コマンド (**shutdown** や **restart** など) を受け取りません。

### add

**-inputQueue** *input queue name* **-queueManager** *queue manager name*

これは、MQListener がこのタスクのためのメッセージを聴取しようとしているキューです。キュー・マネージャーの名前を指定しない場合、このユーティリティは WebSphere MQ に構成されているデフォルト・キュー・マネージャーを使用します。

**-procSchema** *stored procedure schema* **-procName** *stored procedure name*

MQListener がメッセージの着信時にそのメッセージを引き渡すストアード・プロシージャ。

**-dbName** *stored procedure database*

MQListener はストアード・プロシージャにメッセージを引き渡します。これは、そのストアード・プロシージャが定義されているデータベースです。

**-dbUser** *user ID* **-dbPwd** *password*

ストアード・プロシージャの呼び出しに使用するユーザー。

**-mqCoordinated**

WebSphere MQ メッセージ・キューに対する読み取りと書き込みを、DB2 ストアード・プロシージャ呼び出しと一緒に 1 つのトランザクションとして統合することを指定します。そのトランザクション全体を WebSphere MQ コーディネーターが調整します。(このようにしてトランザクションを調整するには、キュー・マネージャーも構成する必要があります。詳細については、WebSphere MQ の資料を参照してください。) デフォルトでは、ストアード・プロシージャを呼び出すトランザクションにメッセージ・キュー操作は含まれません。

**-numInstances** *number of instances to run*

この構成内の実行対象タスクの重複インスタンスの数。値を指定しない場合は、1 つのインスタンスだけが実行されます。

**remove**

**-inputQueue** *input queue name* **-queueManager** *queue manager name*

これは、構成から除去するタスクを定義するキューとキュー・マネージャーです。入力キューとキュー・マネージャーの組み合わせは、1つの構成内で固有なものになっています。

**admin**

**-adminQueue** *admin queue name* **-adminQueueList** *namelist of admin queue names* **-adminQMgr** *admin queue manager*

管理コマンドを送信するためのキューまたはキュー名の名前リスト。キュー・マネージャーを指定しない場合、このユーティリティは WebSphere MQ に構成されているデフォルト・キュー・マネージャーを使用します。

**-adminCommand** *admin command*

コマンドを実行します。コマンドは、**shutdown** か **restart** のいずれかです。**shutdown** の場合は、実行中の MQListener が現在のメッセージの処理を終えた時点でリスナーを終了します。**restart** の場合は、シャットダウンを実行してから、再び構成を読み込んで再始動します。

**例**

```
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
```

## db2mcs - Windows フェイルオーバー・ユーティリティのセットアップ

Microsoft Cluster Server (MSCS) を使用する Windows で DB2 フェイルオーバーをサポートするためインフラストラクチャーを作成します。このユーティリティを使用すると、単一パーティション環境とパーティション・データベース環境の両方でフェイルオーバーが可能になります。

### 許可

ユーザーは、MSCS クラスタ内の各マシンの Administrators グループに属するドメイン・ユーザー・アカウントにログオンする必要があります。

### コマンド構文

```
db2mcs [-f input_file] [-u instance_name] [-l user_name] [-p password]
```

### コマンド・パラメーター

#### -f *input\_file*

MSCS ユーティリティによって使用される入力ファイルを指定します。このパラメーターが指定されている場合、**db2mcs** ユーティリティは、入力ファイルとしてファイル名を使用します。このパラメーターが指定されない場合、**db2mcs** ユーティリティは、現行ディレクトリーにある DB2MSCS.CFG ファイルの使用を試みます。

#### -u *instance\_name*

このオプションを使用すると、**db2mcs** 操作を取り消し、インスタンスを *instance\_name* で指定された非 MSCS インスタンスに復帰させることができます。

#### -l *user\_name*

DB2 サービス用ドメイン・アカウントのユーザー名を指定します。既に DB2MSCS.CFG ファイル内に **DB2\_LOGON\_USERNAME** パラメーターが指定されている場合に、このパラメーターを指定すると、**DB2\_LOGON\_USERNAME** パラメーターの値は無視されます。どちらのパラメーターも指定しない場合、DB2 サービスは、リモート・マシンのローカル管理者アカウントで作成されます。

セキュリティについて懸念している場合には、**-l** パラメーターを使用してください。**DB2\_LOGON\_USERNAME** パラメーターを指定しないでください。

#### -p *password*

DB2 サービス用ドメイン・アカウントのパスワードを指定します。既に DB2MSCS.CFG ファイル内に **DB2\_LOGON\_PASSWORD** パラメーターが指定されている場合に、このパラメーターを指定すると、**DB2\_LOGON\_PASSWORD** パラメーターの値は無視されます。どちらのパラメーターも指定しない場合、パスワードの入力を求めるプロンプトが出ます。

セキュリティについて懸念している場合には、**-p** パラメーターを使用してください。**DB2\_LOGON\_PASSWORD** パラメーターを指定しないでください。

### 使用上の注意

**db2mcs** ユーティリティーは、非 MSCS インスタンスを MSCS インスタンスにトランスフォームするのに使用できる、スタンドアロン型のコマンド行ユーティリティーです。このユーティリティーは、すべての MSCS グループ、リソース、およびリソース依存関係を作成します。また、このユーティリティーは、Windows レジストリーに保管されているすべての DB2 情報をレジストリーのクラスター部分にコピーし、インスタンス・ディレクトリーを共有クラスター・ディスクに移動します。**db2mcs** ユーティリティーは、ユーザーから渡される構成ファイルを、クラスターのセットアップ方法を指定する入力として受け取ります。DB2MSCS.CFG ファイルは、ASCII テキスト・ファイルで、**db2mcs** ユーティリティーが読み取るパラメーターが含まれています。各入力パラメーターは、それぞれ別々の行に `PARAMETER_KEYWORD=parameter_value` というフォーマットで指定します。以下に例を示します。

```
CLUSTER_NAME=FINANCE
GROUP_NAME=DB2 Group
IP_ADDRESS=9.21.22.89
```

2 つのサンプル構成ファイルが、DB2 データベース製品インストール・ディレクトリーの CFG サブディレクトリーにあります。1 つは DB2MSCS.EE というファイルで、これは単一パーティション・データベース環境の例になっています。もう 1 つは DB2MSCS.EEE で、これは、パーティション・データベース環境の例です。

DB2MSCS.CFG ファイルのパラメーターは次のようになっています。

#### DB2\_INSTANCE

DB2 インスタンスの名前。このパラメーターは、グローバルな有効範囲を持っているため、DB2MSCS.CFG ファイル内で一度だけ指定します。**db2mcs** を実行する前に **DB2\_INSTANCE** を停止する必要があります。このパラメーターの最大長は 8 バイトです。

#### DAS\_INSTANCE

DB2 Administration Server インスタンスの名前。このパラメーターは、MSCS 環境で稼働するように DB2 Administration Server をマイグレーションする場合に指定します。このパラメーターは、グローバルな有効範囲を持っているため、DB2MSCS.CFG ファイル内で一度だけ指定します。このパラメーターの最大長は 8 バイトです。

#### CLUSTER\_NAME

MSCS クラスターの名前。この行より後に指定されるすべてのリソースは、別の **CLUSTER\_NAME** パラメーターが指定されるまでこのクラスターに作成されません。このパラメーターの最大長は 15 バイトです。

#### DB2\_LOGON\_USERNAME

DB2 サービス用ドメイン・アカウントのユーザー名 (`domain\user` のように指定)。このパラメーターは、グローバルな有効範囲を持っているため、DB2MSCS.CFG ファイル内で一度だけ指定します。このパラメーターを指定しない場合、DB2 サービスは、リモート・マシンのローカル管理者アカウントで作成されます。このパラメーターの最大長は 256 バイトです。

#### DB2\_LOGON\_PASSWORD

DB2 サービス用ドメイン・アカウントのパスワード。このパラメーターは、グローバルな有効範囲を持っているため、DB2MSCS.CFG ファイル内で一度だけ指



定します。 **DB2\_LOGON\_USERNAME** を指定し、**DB2\_LOGON\_PASSWORD** を指定しない場合、パスワードの入力を求めるプロンプトが出ます。このパラメーターの最大長は 256 バイトです。

### **GROUP\_NAME**

MSCS グループの名前。このパラメーターが指定されたときに、指定された名前の MSCS グループが存在していない場合は、そのグループが新しく作成されます。むろん、グループが既に存在している場合は、そのグループがターゲット・グループになります。このパラメーターより後に指定された MSCS リソースは、別の **GROUP\_NAME** パラメーターが指定されるまで、このグループに作成または移動されます。このパラメーターは、各グループにつき 1 つ指定してください。MSCS グループは、MSCS クラスター内に作成できます。このパラメーターの最大長は 15 バイトです。

### **DB2\_NODE**

現行の MSCS グループに組み込むデータベース・パーティション・サーバー (またはデータベース・パーティション) のデータベース・パーティション番号。同じマシン上に複数の論理データベース・パーティションが存在する場合は、データベース・パーティションごとに別々の **DB2\_NODE** パラメーターが必要です。DB2 リソースが正しい MSCS グループに作成されるよう、このパラメーターは **GROUP\_NAME** パラメーターの後に指定してください。このパラメーターは、複数パーティション・データベース環境に必要です。このパラメーターの最大長は 15 バイトです。

### **IP\_NAME**

IP アドレス・リソースの名前。 **IP\_NAME** の値は任意ですが、クラスター内で固有な値でなければなりません。このパラメーターが指定されると、IP アドレス・タイプの MSCS リソースが作成されます。このパラメーターは、リモート TCP/IP 接続で必要です。単一パーティション・データベース環境の場合、このパラメーターはオプションです。推奨されている名前は、その IP アドレスに対応するホスト名です。このパラメーターの最大長は 256 バイトです。

### **IP\_ADDRESS**

前述の **IP\_NAME** パラメーターで指定した IP リソースの TCP/IP アドレス。  
**IP\_NAME** パラメーターを指定するときはこのパラメーターが必要です。新しい、ネットワーク内のいかなるマシンでも使用されていない IP アドレスが使用されます。このパラメーターの最大長は 15 バイトです。

### **IP\_SUBNET**

前述の **IP\_NAME** パラメーターで指定した IP リソースの TCP/IP サブネット・マスク。**IP\_NAME** パラメーターを指定するときはこのパラメーターが必要です。このパラメーターの最大長は 15 バイトです。

### **IP\_NETWORK**

前述の IP アドレス・リソースが属している MSCS ネットワークの名前。このパラメーターはオプションです。このパラメーターが指定されない場合は、システムが最初に検出した MSCS ネットワークが使用されます。MSCS ネットワークの名前は、「クラスター管理 (Cluster Administrator)」の Networks の分岐の下に示されている通りに、正確に入力してください。前述の 4 つの IP キーワードは、IP アドレス・リソースの作成に使用されます。このパラメーターの最大長は 256 バイトです。



## db2mcs - Windows フェイルオーバー・ユーティリティーのセットアップ

### NETNAME\_NAME

ネットワーク名リソースの名前。このパラメーターは、ネットワーク名リソースを作成する場合に指定してください。単一パーティション・データベース環境では、このパラメーターはオプションです。しかし、パーティション・データベース環境で DB2 インスタンス・ディレクトリーがあるマシンを所有するインスタンスには、必ずこのパラメーターを指定する必要があります。このパラメーターの最大長は 256 バイトです。

### NETNAME\_VALUE

ネットワーク名リソースの値。 **NETNAME\_NAME** パラメーターを指定する場合には、このパラメーターの指定が必要です。このパラメーターの最大長は 256 バイトです。

### NETNAME\_DEPENDENCY

ネットワーク名リソースが依存する IP リソースの名前。各ネットワーク名リソースには、必ず IP アドレス・リソースへの依存関係が必要です。このパラメーターはオプションです。このパラメーターが指定されない場合、ネットワーク名リソースは、グループ内の最初の IP リソースに依存するようになります。このパラメーターの最大長は 256 バイトです。

### SERVICE\_DISPLAY\_NAME

汎用サービス・リソースの表示名。このパラメーターは、汎用サービス・リソースを作成する場合に指定します。このパラメーターの最大長は 256 バイトです。

### SERVICE\_NAME

汎用サービス・リソースのサービス名。 **SERVICE\_DISPLAY\_NAME** パラメーターを指定する場合には、このパラメーターの指定が必要です。このパラメーターの最大長は 256 バイトです。

### SERVICE\_STARTUP

汎用サービス・リソース用のオプション始動パラメーター。このパラメーターの最大長は 256 バイトです。

### DISK\_NAME

現行グループに移動させる物理ディスク・リソースの名前。必要な分だけのディスク・リソースを指定してください。ディスク・リソースは、あらかじめ存在するものでなければなりません。 **db2mcs** ユーティリティーがフェイルオーバー・サポート用に DB2 インスタンスを構成する場合は、グループ内の最初の MSCS ディスクにインスタンス・ディレクトリーがコピーされます。インスタンス・ディレクトリーに別の MSCS ディスクを指定する場合は、 **INSTPROF\_DISK** パラメーターを使用してください。なお、ディスク名は、「クラスター管理 (Cluster Administrator)」で示されている通りに、正確に入力してください。このパラメーターの最大長は 256 バイトです。

### INSTPROF\_DISK

DB2 インスタンス・ディレクトリーを入れる MSCS ディスクを指定するための、オプション・パラメーター。このパラメーターが指定されない場合、 **db2mcs** ユーティリティーは、同じグループに属する最初のディスクを使用します。このパラメーターの最大長は 256 バイトです。

### INSTPROF\_PATH

インスタンス・ディレクトリーのコピー先の正確なパスを指定するための、オプ

## db2mcs - Windows フェイルオーバー・ユーティリティのセットアップ

ション・パラメーター。IPSHAdisks、つまり ServerRAID Netfinity® ディスク・リソース (例、INSTPROF\_PATH=p:¥db2profs) を使用する場合には、必ずこのパラメーターを指定する必要があります。INSTPROF\_PATH と INSTPROF\_DISK の両方が指定されている場合は、INSTPROF\_PATH の方が優先順位が上です。このパラメーターの最大長は 256 バイトです。

### TARGET\_DRVMAP\_DISK

複数パーティション・データベース環境のためのデータベース・ドライブ・マッピングのターゲット MSCS ディスクを指定する、オプション・パラメーター。このパラメーターは、データベースの作成コマンドで指定されたドライブからディスクをマップすることにより、データベースが作成されるディスクを指定します。このパラメーターを指定しない場合は、db2drvmp ユーティリティを使用して手動でデータベース・ドライブ・マッピングを登録する必要があります。このパラメーターの最大長は 256 バイトです。

### DB2\_FALLBACK

DB2 リソースがオフラインにされたときにアプリケーションを強制的にオフにするかどうかを制御する、オプション・パラメーター。このパラメーターが指定されなければ、DB2\_FALLBACK の設定は YES になります。アプリケーションを強制的にオフにしない場合は、DB2\_FALLBACK を NO に設定してください。

## db2mtrk - メモリー・トラッカー

インスタンス、データベース、エージェント、アプリケーションなどの、完全なメモリー状況レポートを提供します。

このコマンドは、以下のメモリー・プール割り振り情報を出力します。

- 現在のサイズ
- 最大サイズ (ハード限界)
- 最大サイズ (最高水準点)
- タイプ (メモリーが使用される機能を示す ID)
- プールを割り振ったエージェント (プールが私用の場合のみ)
- アプリケーション

スナップショット・モニターからも同じ情報を入手できます。

### 有効範囲

パーティション・データベース環境では、このコマンドは、`db2nodes.cfg` ファイルに定義されているどのデータベース・パーティションからでも呼び出すことができます。インスタンス・レベルの情報に戻る場合は別として、このコマンドはそのデータベース・パーティションに関する情報のみを戻します。リモート・サーバーの情報は戻しません。

### 許可

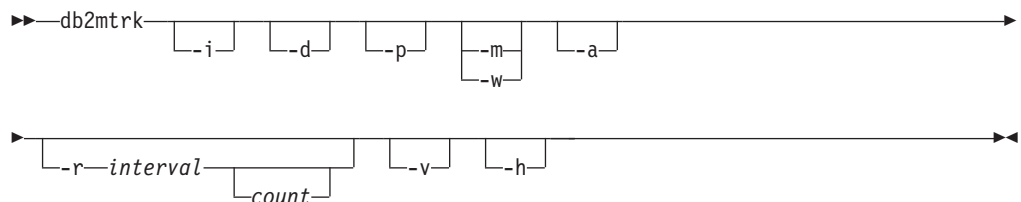
以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMOINT
- SYSMON

### 必要な接続

インスタンス。デフォルトのインスタンス接続が存在しない場合は、アプリケーションによって作成されます。

### コマンド構文



### コマンド・パラメーター

- i     インスタンス・レベルのメモリーを表示します。
- d     データベース・レベルのメモリーを表示します。

## db2mtrk - メモリー・トラッカー

- a アプリケーション・メモリーの使用量を表示します。
- p 非推奨。専用メモリーを表示します。  
-a パラメーターに置き換えると、アプリケーション・メモリーの使用量を表示します。
- m 各プールの最大値を表示します。
- w 各プールの最高水準点を表示します。
- r 反復モード  
*interval* 次のメモリー・トラッカーの呼び出しまでの待機秒数 (反復モード)。  
*count* 反復回数。
- v 冗長出力。
- h ヘルプ画面を表示します。 -h を指定する場合、ヘルプ画面だけが表示され、他の情報は表示されません。

### 例

以下の呼び出しは、データベースおよびインスタンスの通常の値を戻し、10 秒ごとに反復します。

```
db2mtrk -i -d -v -r 10
```

以下の出力サンプルを参考にしてください。

コマンド `db2mtrk -i -d` では、以下の出力が表示されます。

```
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
```

コマンド `db2mtrk -a -i -d` では、以下の出力が表示されます。

```
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

```

コマンド `db2mtrk -a -v -i -d` では、以下の出力が表示されます。

```

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

```

## db2mtrk - メモリー・トラッカー

```
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
```

### 使用上の注意

注:

1. フラグを指定しない場合は、使用量が戻されます。
2. **-d**、**-h**、**-i**、**-p** または **-a** からフラグを 1 つ指定する必要があります。
3. **-p** パラメーターを指定すると、エージェント ID ごとにグループ分けされた詳細な専用メモリーの使用量の情報が戻されます。
4. **-a** パラメーターを指定すると、アプリケーション ID ごとにグループ分けされた詳細なアプリケーション・メモリーの使用量の情報が返されます。
5. 報告される「Other Memory (その他のメモリー)」とは、データベース管理システムを操作するために使用するメモリーのことです。
6. 表示される最大サイズが構成パラメーターに割り当てられた値より大きい場合があります。例えばパッケージ・キャッシュなどの場合がそうです。このような場合、構成パラメーターに割り当てられた値は「ソフト限界」として使用され、実際のプール・メモリー使用量は構成済みのサイズを上回る可能性があります。
7. バッファー・プール・ヒープの場合、括弧内に示された番号がバッファー・プール ID であるか、またはこのバッファー・プールがシステム・バッファー・プールの 1 つであることを示しています。
8. アプリケーション・ヒープの場合、括弧で指定した番号が、アプリケーション ID になります。

9. いくつかのヒープに関してメモリー・トラッカーが報告する最大サイズは、マシン上の物理メモリーの量となります。これらのヒープは限りないヒープと呼ばれ、無制限の最大サイズが宣言されています。それは、そのヒープが宣言されたときに、ピーク時に必要とするメモリー・サイズが不明であったためです。これらのヒープはマシン上の物理メモリーによって厳密には束縛されませんが、それが適切な近似値となるために、最大サイズとして報告されます。
10. バッファ・プール・ヒープは常に完全に割り振られるので、メモリー・トラッカーが報告するこれらのヒープの現在のサイズと最大サイズは同じ値になります。バッファ・プール・サイズが自動的に設定されている場合は、バッファ・プール・ヒープの現在のサイズと最大サイズはワークロードと使用可能メモリーに基づいて時の経過とともに調整されます。

## db2nchg - データベース・パーティション・サーバー構成の変更

データベース・パーティション・サーバー構成を変更します。これには、あるマシンから別のマシンへのデータベース・パーティション・サーバーの移動、マシンのTCP/IP ホスト名の変更、データベース・パーティション・サーバー用の別の論理ポート番号または別のネットワーク名の選択も含まれます。

このコマンドが使用できるのは、データベース・パーティション・サーバーが停止している場合だけです。

このコマンドは、Windows のオペレーティング・システムのみで使用できます。

### 許可

ローカル管理者

### コマンド構文

```

db2nchg /n:dbpartitionnum /i:instance_name
/u:username,password /p:logical_port /h:host_name
/m:machine_name /g:network_name

```

### コマンド・パラメーター

#### **/n:dbpartitionnum**

変更するデータベース・パーティション・サーバー構成のデータベース・パーティション番号を指定します。

#### **/i:instance\_name**

このデータベース・パーティション・サーバーが参加するインスタンスを指定します。パラメーターが指定されていない場合、デフォルトは現行のインスタンスになります。

#### **/u:username,password**

ユーザー名およびパスワードを指定します。パラメーターが指定されない場合、既存のユーザー名とパスワードが設定されます。

#### **/p:logical\_port**

データベース・パーティション・サーバー用の論理ポートを指定します。データベース・パーティション・サーバーを別のマシンに移動させるには、このパラメーターを指定する必要があります。パラメーターが指定されない場合、論理ポート番号は変更されません。

#### **/h:host\_name**

内部通信用に FCM によって使用される TCP/IP ホスト名を指定します。パラメーターが指定されない場合、ホスト名は変更されません。

#### **/m:machine\_name**

データベース・パーティション・サーバーが常駐するマシンを指定します。



## db2nchg - データベース・パーティション・サーバー構成の変更

インスタンスに既存のデータベースがない場合にのみ、データベース・パーティション・サーバーを移動させることができます。

### */g:network\_name*

データベース・パーティション・サーバーのネットワーク名を変更します。このパラメーターは、マシンに複数の IP アドレスがある場合に、特定の IP アドレスをデータベース・パーティション・サーバーに適用するために使用できます。ネットワーク名または IP アドレスを入力できます。

### 例

インスタンス TESTMPP に参加する、データベース・パーティション 2 に割り当てられている論理ポートを論理ポート 3 に変更するには、以下のコマンドを入力します。

```
db2nchg /n:2 /i:TESTMPP /p:3
```

## db2ncrt - インスタンスへのデータベース・パーティション・サーバーの追加

データベース・パーティション・サーバーをインスタンスに追加します。

このコマンドは Windows オペレーティング・システムでのみ使用できます。

### 有効範囲

既にインスタンスが存在しているコンピューターにデータベース・パーティション・サーバーが追加される場合には、データベース・パーティション・サーバーはコンピューターへの論理データベース・パーティションとして追加されます。インスタンスが存在していないコンピューターにデータベース・パーティション・サーバーが追加される場合には、インスタンスが追加され、そのコンピューターは新しい物理データベース・パーティション・サーバーになります。インスタンスにデータベースがある場合には、このコマンドを使用してはなりません。代わりに、**START DATABASE MANAGER** コマンドを **ADD DBPARTITIONNUM** オプションを指定して発行してください。こうすると、新しいデータベース・パーティション・サーバーにデータベースが確実に正しく追加されます。データベースが作成されたインスタンスにデータベース・パーティション・サーバーを追加することも可能です。db2nodes.cfg ファイルは編集するべきではありません。このファイルを変更すると、パーティション・データベース環境に不整合が生じる可能性があるためです。

### 許可

新しいデータベース・パーティション・サーバーが追加されるコンピューターに対するローカル管理者権限。

### コマンド構文

```
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—
```

### コマンド・パラメーター

**/n:dbpartitionnum**

データベース・パーティション・サーバーを識別する固有のデータベース・パーティション番号。入力できる数値の範囲は 1 から 999 までです。

**/u:username,password**

DB2 のログオン・アカウント名およびパスワードを指定します。

**/i:instance\_name**

インスタンス名を指定します。パラメーターが指定されていない場合、デフォルトは現行のインスタンスになります。

**/m:machine\_name**

データベース・パーティション・サーバーが常駐する Windows ワークステ

## db2ncrt - インスタンスへのデータベース・パーティション・サーバーの追加

ーションのコンピューター名を指定します。データベース・パーティション・サーバーをリモート・コンピューター上に追加している場合、このパラメーターは必須です。

### */p:logical\_port*

データベース・パーティション・サーバーに使用する論理ポート番号を指定します。このパラメーターが指定されていない場合、割り当てられる論理ポート番号は 0 です。論理データベース・パーティション・サーバーを作成する際には、このパラメーターを指定しなければならず、使用していない論理ポート番号を選択しなければなりません。以下の制限事項に注意してください。

- すべてのコンピューターには、論理ポートが 0 のデータベース・パーティション・サーバーがなければなりません。
- ポート番号は、`x:%winnt%system32%drivers%etc% ディレクトリーで FCM 通信用に予約されているポート範囲内でなければなりません`。例えば、4 個のポートが現行のインスタンスに予約されている場合には、最大のポート番号は 3 になります。ポート 0 は、デフォルトの論理データベース・パーティション・サーバー用に使用されます。

### */h:host\_name*

内部通信用に FCM によって使用される TCP/IP ホスト名を指定します。データベース・パーティション・サーバーをリモート・コンピューター上に追加する場合、このパラメーターは必須です。

### */g:network\_name*

データベース・パーティション・サーバーのネットワーク名を指定します。パラメーターが指定されていない場合、システムで検出される最初の IP アドレスが使用されます。このパラメーターは、コンピューターに複数の IP アドレスがある場合に、特定の IP アドレスをデータベース・パーティション・サーバーに適用するために使用できます。ネットワーク名または IP アドレスを入力できます。

### */o:instance\_owning\_machine*

インスタンスを所有しているコンピューターのコンピューター名を指定します。デフォルトはローカル・コンピューターです。インスタンス所有コンピューターではない任意のコンピューターで **db2ncrt** コマンドが呼び出される場合、このパラメーターは必須です。

## 例

インスタンス所有のコンピューター SHAYER 上で、インスタンス TESTMPP に新しいデータベース・パーティション・サーバーを追加する場合、新しいデータベース・パーティション・サーバーがデータベース・パーティション 2 で、論理ポート 1 を使用する場合には、次のコマンドを入力します。

```
db2ncrt /n:2 /u:QBPAULZ%paulz,g1reeky /i:TESTMPP /m:TEST /p:1 /o:SHAYER /h:TEST
```

## db2ndrop - インスタンスからのデータベース・パーティション・サーバーのドロップ

データベースのないインスタンスからデータベース・パーティション・サーバーをドロップします。データベース・パーティション・サーバーがドロップされた場合には、このデータベース・パーティション番号を新しいデータベース・パーティション・サーバーで再使用できます。

このコマンドが使用できるのは、データベース・パーティション・サーバーが停止している場合だけです。

このコマンドは、Windows のオペレーティング・システムのみで使用できます。

### 許可

データベース・パーティション・サーバーをドロップするマシンに対するローカル管理者権限。

### コマンド構文

```
db2ndrop /n:dbpartitionnum [/i:instance_name]
```

### コマンド・パラメーター

*n*:*dbpartitionnum*

データベース・パーティション・サーバーを識別する固有のデータベース・パーティション番号。

*i*:*instance\_name*

インスタンス名を指定します。パラメーターが指定されていない場合、デフォルトは現行のインスタンスになります。

### 例

```
db2ndrop /n:2 /i=KMASCI
```

### 使用上の注意

インスタンスの所有するデータベース・パーティション・サーバー (*dbpartitionnum* 0) がインスタンスからドロップされると、このインスタンスは使用できなくなります。インスタンスをドロップするには、**db2idrop** コマンドを使用します。

このインスタンスにデータベースがある場合には、このコマンドを使用してはなりません。代わりに、**db2stop drop dbpartitionnum** コマンドを使用する必要があります。こうすると、パーティション・データベース環境からデータベース・パーティション・サーバーを確実に除去することができます。データベースが存在するインスタンスでデータベース・パーティション・サーバーをドロップすることも可能です。 *db2nodes.cfg* ファイルは編集するべきではありません。このファイルを変更すると、パーティション・データベース環境に不整合が生じる可能性があるためです。

## db2ndrop - インスタンスからのデータベース・パーティション・サーバーのドロップ

複数の論理データベース・パーティション・サーバーを実行しているマシンから、論理ポート 0 に割り当てられたデータベース・パーティション・サーバーをドロップするには、他の論理ポートに割り当てられている他のすべてのデータベース・パーティション・サーバーを最初にドロップする必要があります。各データベース・パーティション・サーバーには、論理ポート 0 に割り当てられているデータベース・パーティション・サーバーが必ず必要です。

## db2nrcfg - 非 root インストール構成ツール

DB2 の非 root インストールに使用する構成ツールです。

### 許可

非 root インストールを所有する非 root ID。

### 必要な接続

なし

### コマンド構文

```
db2nrcfg [-a AuthType] [-d] [-p PortName] [-s InstType] [-j "TEXT_SEARCH, portnumber"] [-h | -?]
```

### コマンド・パラメーター

#### -a *AuthType*

インスタンスの認証タイプ (SERVER、CLIENT、または SERVER\_ENCRYPT) を設定します。

-d デバッグ・モードをオンにします。

#### -p *PortName*

このインスタンスが使用するポート名またはポート番号を設定します。

#### -s *InstType*

作成するインスタンスのタイプ (wse、ese、または client) を設定します。

#### -j "TEXT\_SEARCH"

DB2 Text Search サーバーを、サービス名と TCP/IP ポート番号に対する生成済みのデフォルト値を使用して構成します。インスタンス・タイプがクライアントの場合、このパラメーターは使用できません。

#### -j "TEXT\_SEARCH, *portnumber*"

DB2 Text Search サーバーを、デフォルトのサービス名と提供されたポート番号を使用して構成します。有効なポート番号は 1024 - 65535 の範囲になければなりません。

#### -h | -?

ヘルプ情報を表示します。

### 使用上の注意

このコマンドは、非 root インストールの際に DB2 インストーラーによって自動的に実行されます。

root インストールが使用する **db2icrt**、**db2iupdt**、および **db2iupgrade** は非 root インストールでは使用できません。

## db2nrupdt - 非 root インストールを行ったインスタンスの更新

DB2 の非 root インストールによって作成されたインスタンス用に使用する更新ツール。

### 許可

非 root インストールを所有する非 root ID。

### 必要な接続

なし

### コマンド構文

```
db2nrupdt [-a AuthType] [-d] [-k] [-j "TEXT_SEARCH, portnumber"] [-h | -?]
```

### コマンド・パラメーター

#### -a *AuthType*

インスタンスの認証タイプ (SERVER、CLIENT、または SERVER\_ENCRYPT) を設定します。

-d デバッグ・モードをオンにします。

#### -k

更新時に現行のインスタンス・タイプが変更されないようにします。

#### -j "TEXT\_SEARCH"

DB2 Text Search サーバーを、サービス名と TCP/IP ポート番号に対する生成済みのデフォルト値を使用して構成します。インスタンス・タイプがクライアントの場合、このパラメーターは使用できません。

#### -j "TEXT\_SEARCH, portnumber"

DB2 Text Search サーバーを、デフォルトのサービス名と提供されたポート番号を使用して構成します。有効なポート番号は 1024 - 65535 の範囲になければなりません。

#### -h | -?

ヘルプ情報を表示します。

### 使用上の注意

root インストールが使用する db2icrt、db2iupdt、および db2iupgrade コマンドは非 root インストールでは使用できません。

## db2nrupgrade - 非 root インスタンスのアップグレード

旧バージョンの DB2 データベース・システムの非 root インスタンスを、**db2nrupgrade** コマンドを実行している現行バージョンの DB2 コピーにアップグレードします。

このコマンドは、Linux および UNIX のオペレーティング・システムでのみ利用できます。

このコマンドは、*DB2DIR/instance* ディレクトリーにあります。*DB2DIR* は、新規リリースの DB2 データベース・システムがインストールされているインストール・ロケーションを表します。このコマンドは、root インストール用のインスタンス・アップグレードをサポートしません。

### 許可

非 root インストール・コピーを所有する非 root ID。

### コマンド構文

```
db2nrupgrade [-d] [-a AuthType] [-b backup_dir] [-j "TEXT_SEARCH, portnumber"] [-h | -?]
```

### コマンド・パラメーター

- d デバッグ・モードをオンにします。このオプションは、DB2 サポートからの指示があった場合にのみ使用してください。
- a *AuthType*  
インスタンスの認証タイプ (SERVER、CLIENT、または SERVER\_ENCRYPT) を指定します。デフォルトは SERVER です。
- b *backup\_dir*  
このパラメーターは、必須です。旧バージョンの DB2 の構成ファイルが保管されているディレクトリーを指定します。バックアップ・ディレクトリーは、*sqllib\_vVR* という形式で **db2setup** ログに組み込まれます。ここで、*V* はバージョン番号で、*R* は古いコピーのリリース番号です。例えば、V9.5 がインストールされていて、その後 **db2setup** コマンドを使用して V9.7 をインストールする場合には、**db2setup** ログ・ファイルに *sqllib\_v95* という名前のバックアップ・ディレクトリーが見つかります。
- j "TEXT\_SEARCH"  
DB2 Text Search サーバーを、サービス名と TCP/IP ポート番号に対する生成済みのデフォルト値を使用して構成します。インスタンス・タイプがクライアントの場合、このパラメーターは使用できません。
- j "TEXT\_SEARCH, portnumber"  
DB2 Text Search サーバーを、デフォルトのサービス名と提供されたポート番号を使用して構成します。有効なポート番号は 1024 - 65535 の範囲になければなりません。
- h | -? ヘルプ情報を表示します。



### 使用上の注意

- このコマンドはコピーのアップグレード時に自動的に実行されます。コピーのアップグレードが失敗したのではない限り、このコマンドを手動で実行する必要はありません。

## db2osconf - カーネル・パラメーター値のためのユーティリティー

システムのサイズに基づいてカーネル・パラメーター値の推奨値を作成します。推奨値の大きさは、指定のシステムで一般的なワークロードの大部分を処理するために十分なものとなります。現在このコマンドを使用できるのは、Solaris オペレーティング・システム上の DB2 データベースと、HP-UX 上の 64 ビット DB2 インスタンスだけです。

### 許可

- DB2 Database for HP-UX の場合、権限は必要ありません。 **db2osconf** ユーティリティーが推奨する変更を行うには、 `root` ユーザー権限を持っている必要があります。
- Solaris オペレーティング・システム版の DB2 データベースの場合は、 `root` ユーザー権限を持っているか、 `sys` グループのメンバーである必要があります。

### コマンド構文

現在サポートされているオプションのリストを入手するには、 **db2osconf -h** と入力してください。

```
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
```

### コマンド・パラメーター

- c オプションは、クライアントだけをインストールするためのものです。このオプションは、Solaris オペレーティング・システム上の DB2 データベース・システムにのみ使用できます。
- f 現行のカーネル・パラメーターと **db2osconf** ユーティリティーで推奨される値との比較に使用します。 **db2osconf** コマンドで他のオプションを入力しない場合は、 **-f** オプションがデフォルトになります。 Solaris オペレーティング・システムの場合は、異なるカーネル・パラメーターだけが表示されます。 HP-UX の場合、 **-f** オプションは、推奨パラメーターのリストと、パラメーター値に対する推奨変更のリストを戻します。  

```
***** Please Change the Following in the Given Order *****
```

```
WARNING [<parameter name>] should be set to <value>
```
- l 現行のカーネル・パラメーターをリストします。
- m 物理メモリーのサイズを GB 単位でオーバーライドします。通常、 **db2osconf** ユーティリティーは物理メモリーのサイズを自動的に判断します。このオプションは、Solaris オペレーティング・システム上の DB2 データベース・システムにのみ使用できます。
- n システムの CPU の数をオーバーライドします。通常、 **db2osconf** ユーティ

## db2osconf - カーネル・パラメーター値のためのユーティリティー

リティーは CPU の数を自動的に判断します。このオプションは、Solaris オペレーティング・システム上の DB2 データベース・システムにのみ使用できます。

- p SYSV メッセージ・キューのパフォーマンス・レベルを設定します。0 (ゼロ) がデフォルトで、3 が最高の設定値です。この値をより高く設定すると、メッセージ・キュー機能のパフォーマンスは向上しますが、より多くのメモリーが使用されます。
- s スケール係数を設定します。デフォルトのスケール係数は 1 であり、大部分のワークロードはこの値で十分です。スケール係数が 1 では不十分な場合、そのワークロードを処理するにはシステムが小さすぎる可能性があります。スケール係数はカーネル・パラメーターの推奨値を、現行システムのサイズよりもその比率だけ大きなサイズのシステムの値に設定します。例えば、スケール係数が 2.5 の場合、現行システムのサイズよりも 2.5 倍大きいシステムのカーネル・パラメーターを推奨します。
- t semsys:seminfo\_semume および shmsys:shminfo\_shmseg カーネル・パラメーター値の推奨値を作成します。このオプションは、Solaris オペレーティング・システム上の DB2 データベース・システムにのみ使用できます。相当な数の接続を持つマルチスレッド・プログラムでは、これらのカーネル・パラメーターをデフォルト値よりも大きな値に設定する必要がある場合があります。それらをリセットする必要があるのは、それらを必要とするマルチスレッド・プログラムがローカル・アプリケーションである場合だけです。

### semsys:seminfo\_semume

任意の 1 プロセスが使用できるセマフォア取り消し構造の制限

### shmsys:shminfo\_shmseg

任意の 1 プロセスが作成できる共有メモリー・セグメント数の制限

これらのパラメーターは、/etc/system ファイルで設定されます。以下のセクションは値を設定するためのガイドであり、**db2osconf** ユーティリティーはこれらの値を推奨します。ローカル接続ごとに、DB2 は 1 つのセマフォアと 1 つの共有メモリー・セグメントを使用して通信します。マルチスレッドのアプリケーションがローカル・アプリケーションであり、DB2 データベースに対して  $X$  個の接続がある場合、DB2 データベース・システムと通信するためには、アプリケーション (プロセス) に  $X$  個の共有メモリー・セグメントと  $X$  個のセマフォア取り消し構造が必要になります。それで、2 つのカーネル・パラメーターの値は  $X + 10$  に設定してください (プラス 10 は安全のためのマージンとなります)。

-l または -f オプションを指定しないと、**db2osconf** ユーティリティーは /etc/system ファイルの構文を使用してカーネル・パラメーターを表示します。人為的なエラーを回避するために、出力を /etc/system ファイルに直接カット・アンド・ペーストすることができます。

カーネル・パラメーターは、CPU の数およびシステム上の物理メモリー量の両方に基づいて推奨されます。一方が不釣合に小さい場合、推奨値は 2 つのうちの小さい方に基づいたものになります。

## 例

-t オプションを 500 スレッドに設定して **db2osconf** ユーティリティーを実行した場合に生成される出力例を以下に示します。受け取る結果はマシンに特定のものなので、受け取る結果は使用する環境によって異なります。

```
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 および set shmsys:shminfo\_shmseg のための推奨値は、db2osconf -t 500 を実行して入手できる追加の値です。

## 使用上の注意

特定の DB2 データベース・ワークロードに基づくカーネル・パラメーターを推奨することは可能ですが、このレベルの正確さには利点がありません。カーネル・パラメーター値が実際に必要な値と過度に接近していて、ワークロードが将来変更される場合、DB2 データベース・マネージャーにはプロセス間通信 (IPC) リソースの不足の問題が生じることがあります。IPC リソースが不足すると DB2 データベース・マネージャーに計画外の停止が生じて、カーネル・パラメーターを増加させるためにリブートが必要になります。カーネル・パラメーターをある程度高い値に設定することにより、将来その値を変更する必要性を少なくするか、なくすことができます。カーネル・パラメーターの推奨値によって消費されるメモリーの量は、システムのサイズと比較して極めて小さいものです。例えば、4 GB の RAM および 4 つの CPU を備えたシステムでは、推奨されるカーネル・パラメーターのメモリー量は 4.67 MB つまり 0.11% となります。カーネル・パラメーターに使用されるこの小さなメモリー部分は、利点を考えると受け入れられるものです。

Solaris オペレーティング・システムの場合、**db2osconf** ユーティリティーには、64 ビット・カーネル用と 32 ビット・カーネル用の 2 つのバージョンがあります。このユーティリティーは、以下の特別装置にアクセスするので (アクセスは読み取り専用です)、root として実行するか、グループ sys によって実行する必要があります。

```
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
```

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

DB2 データベース・システム・メモリー・セット内の情報を取り出します。

### 許可

以下のいずれかの権限レベルが必要です。

- SYSADM 権限レベル。
- SYSCTRL 権限レベル。
- SYSMOINT 権限レベル。
- SYSMON 権限レベル。

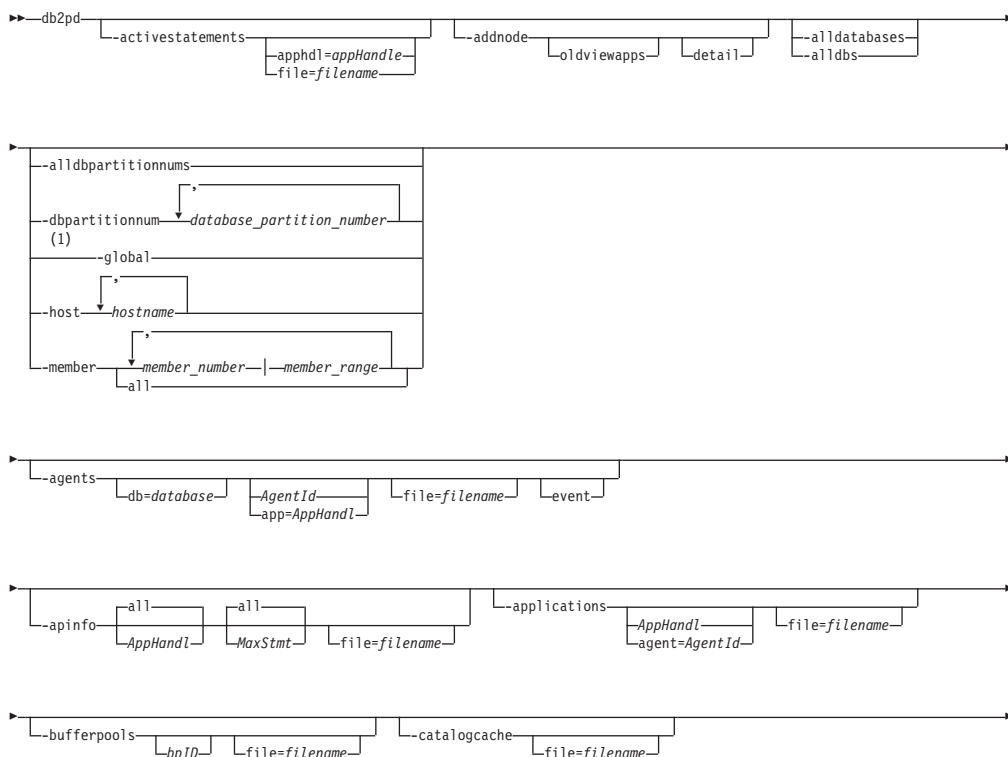
SYSMON 権限レベルが付与されている場合、以下のオプションは使用できません。

- **dump**
- **memblocks**
- **stack**

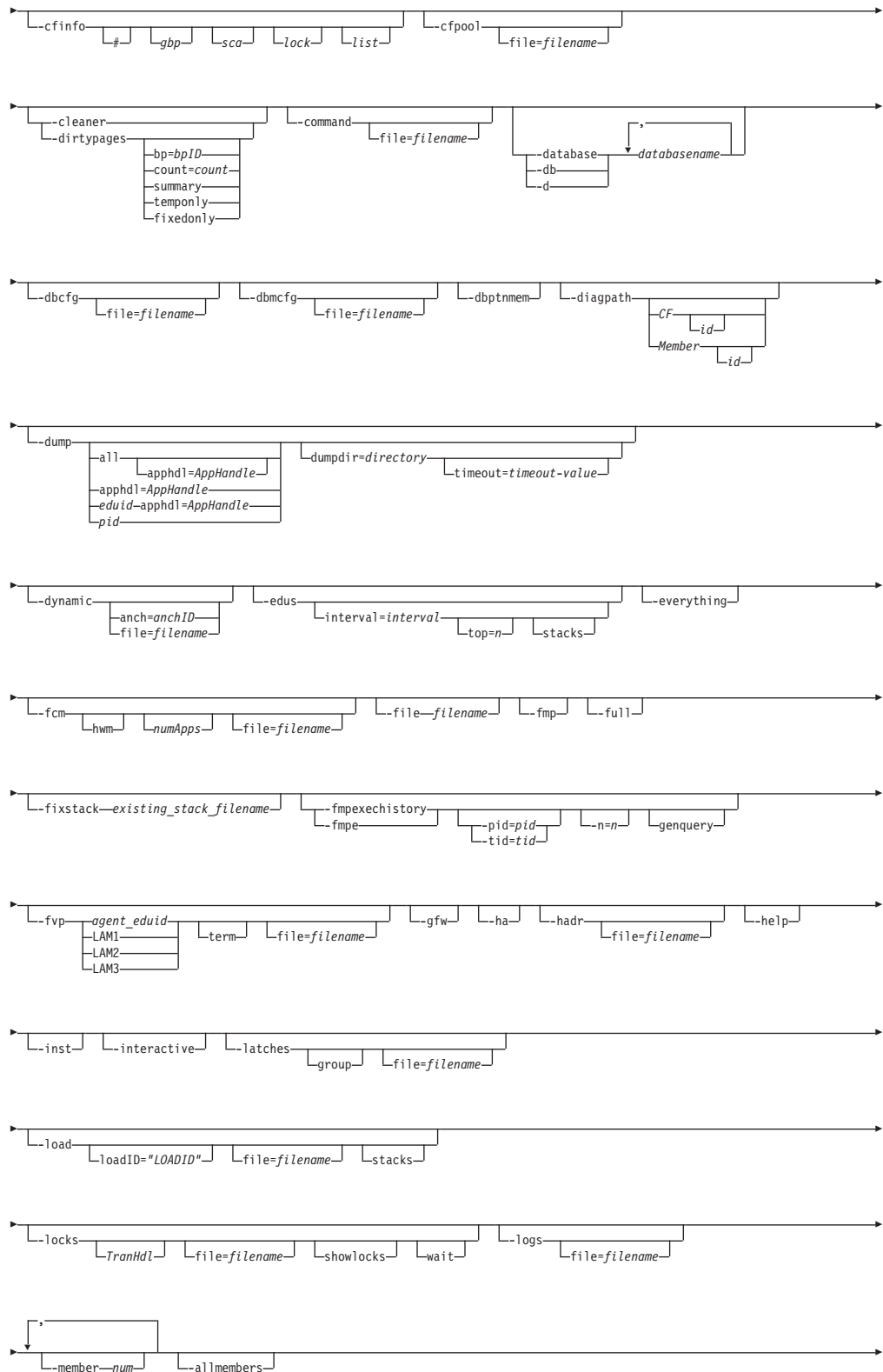
### 必要な接続

接続に必要な最低限の要件はありません。しかし、データベースの有効範囲オプションが指定されている場合、コマンドが要求された情報を戻す前に、そのデータベースをアクティブにする必要があります。

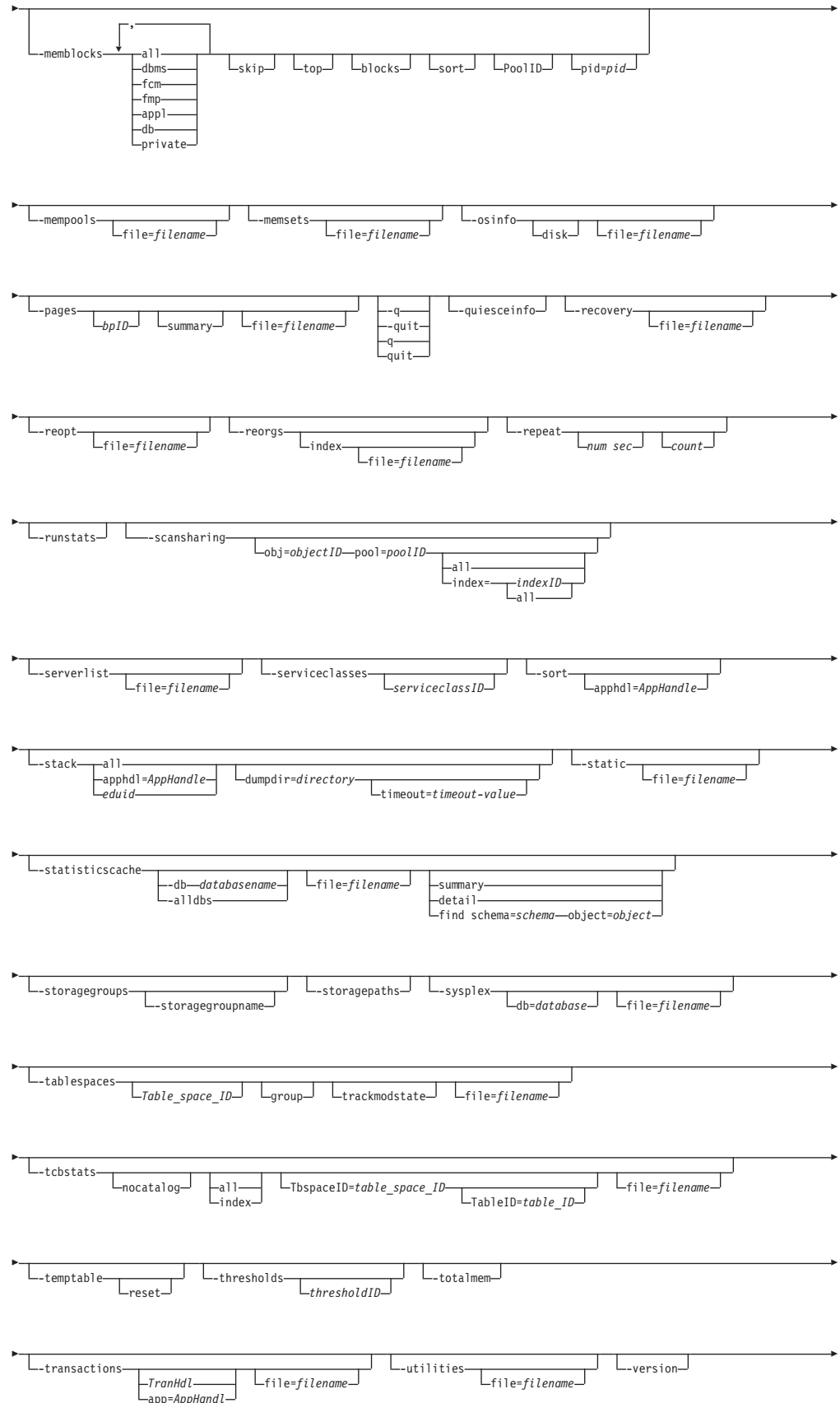
### コマンド構文

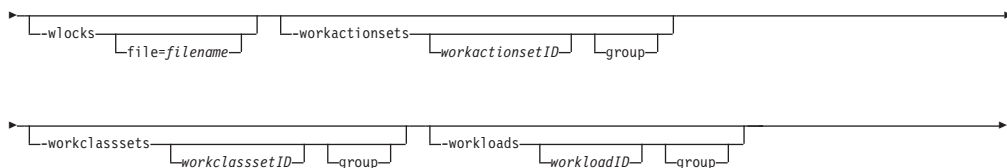


## db2pd - DB2 データベースのモニターおよびトラブルシューティング



## db2pd - DB2 データベースのモニターおよびトラブルシューティング





注:

- 1 **-global** パラメーターは使用すべきではありません。 **-member all** パラメーター・オプションを使用すると、情報をグローバルに取得することができます。

### コマンド・パラメーター

#### **-activestatements**

接続中のアプリケーションのために現在処理されているアクティビティーに関する情報を返します。このようなアプリケーションの例として、動的 SQL ステートメント、静的 SQL ステートメント、およびロードなどがあります。

#### **apphdl=appHandle**

アプリケーション・ハンドルが指定される場合、その特定のアプリケーションに関する情報が戻されます。

#### **file=filename**

**-activestatements** 出力を指定のファイルに送信します。

**-activestatement** の使用上の注意を参照してください。

#### **-addnode**

データベース・パーティション・サーバーの追加操作に関する進行情報を戻します。このパラメーターが情報を戻すのは、追加中のデータベース・パーティション・サーバーで発行された場合だけです。進行情報は、再始動するまで新しいデータベース・パーティション・サーバーにあります。既存のデータベース・パーティション・サーバーで発行すると、このパラメーターは何も情報を戻しません。

**db2pd -addnode** コマンドの出力例を参照してください。

#### **-alldatabases | -alldb**

すべてのデータベースのすべてのメモリー・セットにコマンドがアタッチします。

#### **-alldbpartitionnums**

ローカル・ホスト上のすべてのアクティブなデータベース・パーティション・サーバーでこのコマンドを実行するように指定します。このパラメーターは、**db2pd** が実行されている物理マシン上のデータベース・パーティション・サーバーからの情報のみを報告します。

#### **-allmembers**

DB2 pureScale 環境 のすべてのアクティブなメンバーでこのコマンドを実行することを指定します。 **db2pd** は、**db2pd** が実行しているものと同じ物理マシン上のデータベース・メンバーからの情報だけを報告します。

#### **-agents**

エージェントに関する情報を戻します。



## db2pd - DB2 データベースのモニターおよびトラブルシューティング

エージェント ID が指定される場合、そのエージェントに関する情報が戻されます。アプリケーション ID が指定される場合、そのアプリケーションの作業を実行するすべてのエージェントに関する情報が戻されます。出力の有効範囲として設定したいデータベースを選択した場合、このコマンド・パラメーターを **-inst** パラメーターとともに指定します。

### event

このオプションは、エージェントによって処理されているイベントのメトリックを返します。返されるメトリックには、イベントが最後に変更された時刻、イベントの状態、イベントのタイプ、イベントのオブジェクト、およびイベント・オブジェクト名が含まれます。

-agents の使用上の注意を参照してください。

### -apinfo

現在の作業単位 (UOW) の動的 SQL ステートメントの実行 (それが適用される場合) を含む、アプリケーションについての詳細情報を表示します。

#### AppHandl

アプリケーション・ハンドルが指定される場合、その特定のアプリケーションに関する情報が戻されます。デフォルトでは、そのパーティションで実行中のすべてのアプリケーションに関する情報が表示されます。

#### MaxStmt

最大ステートメント数が指定されている場合、指定された最大数と等しい最新の SQL ステートメントに関する情報が戻されます。デフォルトでは、実行されたすべての SQL ステートメントに関する情報が表示されます。

#### file=filename

**-apinfo** 出力を指定のファイルに送信します。

**db2pd -apinfo** コマンドの出力例を参照してください。

**注:** アプリケーションの SQL ステートメント・テキストを含む、作業単位 (UOW) の過去の履歴を取り込むには、ステートメント履歴節を使用してデッドロック・イベント・モニターをアクティブ化します。例えば、以下のステートメントの 1 つを使用します。

```
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 ステートメントには、データの書き込み先となる表スペースおよび表の名前を指定する機能など、追加のオプションがあります。詳しくは、CREATE EVENT MONITOR ステートメントの説明を参照してください。ステートメント履歴の機能を持つイベント・モニターは、すべてのアプリケーションに影響を与えるので、DB2 データベース・マネージャーによるモニター・ヒープの使用量が増加します。

**-apinfo** の使用上の注意を参照してください。

### -applications

アプリケーションに関する情報を戻します。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

アプリケーション ID が指定される場合、そのアプリケーションに関する情報が戻されます。

エージェント ID が指定される場合、アプリケーションの代わりに作動しているエージェントに関する情報が戻されます。 **-applications** の使用上の注意を参照してください。

### **-bufferpools**

バッファ・プールに関する情報を戻します。バッファ・プール ID が指定される場合、そのバッファ・プールに関する情報が戻されます。

**-bufferpools** の使用上の注意を参照してください。

### **-catalogcache**

メモリー内の統計を維持する、カタログ・キャッシュに関する情報を戻します。

**db2pd -catalogcache** コマンドの出力例を参照してください。

SYSTABLES の出力には、同じ表に対して複数の項目が含まれる可能性があります (上記の出力の DEPT を参照してください)。複数の項目は、同じ表に関する異なるバージョンの統計に対応しています。使用ロックの名前は、同じオブジェクトの複数の項目の間で固有です。ソフト無効項目には「S」というマークが付けられます。 **-catalogcache** の使用上の注意を参照してください。

### **-cfinfo**

パフォーマンスや他の問題を診断するときに役立つ CF 情報をダンプします。 **gbp**、**sca**、**lock**、または **list** のいずれかのサブオプションを使用することにより、ダンプする情報の特定の構造を指定することができます。例えば、**db2pd -cfinfo 2 sca** を実行すると、CF #2 からの SCA 構造情報をダンプします。

### **-cfpool**

現行メンバー上の各 CF 接続プール・エントリーとその状況のリストを表示します。各エントリーの状況には、それが使用中かどうか、それを使用している DB2 エンジン・ディスパッチ可能単位 (EDU)、およびその使用対象になっている機能が含まれています。

**cfpool** オプションは、コマンド接続のモニターおよび Host Channel Adapter (HCA) ポート・マッピング情報の表示に使用できます。この情報を使用すると、HCA ポート間のロード・バランシングが予期されるとおりに動作していることを確認できます。また、この情報を使用して、HCA フェイルオーバーが予期されるとおりに機能していることを検証することもできます (例えば、オフライン接続からの接続の排出や、ポートがオンラインに戻った後の接続の再確立など)。

さらに、**cfpool** オプションを使用する場合、XI およびロック通知接続が確立される HCA ポートのクラスター相互接続ネット名に関する情報が、**db2pd** の出力に含まれます。

### **-cleaner**

ページ・クリーナーの関連情報をデータベースからダンプします。このオブ

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

ションの前にアクティブ・データベースを指定する必要があります。これは、**-database** または **-db** オプションに適切なアクティブ・データベース名を指定することによって行います。

**-cleaner** オプションの出力例を参照してください。

### **-command filename**

ファイルに指定されている **db2pd** コマンド・オプションを読み取って実行します。

### **-database | -db | -d databasename**

指定されたデータベースのデータベース・メモリー・セットにコマンドがアタッチします。別名ではなくデータベース名を指定してください。

**-dbcfg** データベース構成パラメーターの設定を戻します。 **-dbcfg** の使用上の注意を参照してください。

### **-dbmcfg**

データベース・マネージャー構成パラメーターの設定を戻します。

出力の有効範囲として設定したいデータベースを選択した場合、このオプションに **-inst** コマンド・パラメーターを指定します。 **-dbmcfg** の使用上の注意を参照してください。

### **-dbpartitionnum number**

指定したローカルまたはリモートのデータベース・パーティション・サーバー上でコマンドを実行するように指定します。

### **-dbptnmem**

データベース・パーティションのメモリー統計をリストします。

### **-diagpath**

完全に解決された、分割された診断パスを返します。 DB2 pureScale 環境では、これは、すべてのメンバーおよび CF の診断パスを返します。

### **CF|member**

メンバーまたは CF のどちらかの診断パスを返します。 *id* が指定されていない場合、すべてのメンバーまたは CF の診断ログ・パスを返します。

### **-dirtypages**

データベース内の各バッファー・プールからダーティー・ページをダンプします。このオプションの前にアクティブ・データベースを指定する必要があります。これは、**-database** または **-db** オプションに適切なアクティブ・データベース名を指定することによって行います。

**bpID** 指定バッファー・プールからダーティー・ページをダンプする場合にこのオプションを指定します。

**count** 各バッファー・プール内の最初の *count* 個のダーティー・ページをダンプする場合にこのオプションを指定します。

### **summary**

各バッファー・プールのリカバリー関連情報をダンプする場合にこのオプションを指定します。

**temponly**

各バッファ・プールから一時ダーティ・ページをダンプする場合にこのオプションを指定します。

**fixedonly**

修正されたダーティ・ページを各バッファ・プールからダンプする場合にこのオプションを指定します。

**-dirtypages** オプションの出力例を参照してください。

**-dump diagpath** ディレクトリー内にスタック・トレース・ファイルおよびバイナリー・ダンプ・ファイルを生成します。 UNIX オペレーティング・システムでのみ使用できます。

- **a11** コマンド・パラメーターと共に指定すると、現行のデータベース・パーティション内にあるすべてのエージェントに対してスタック・トレース・ファイルおよびバイナリー・ダンプ・ファイルを生成します。
- **a11** パラメーターおよび **apphd1=appHandle** パラメーターと共に指定すると、指定した *appHandle* アプリケーションに関連付けられたすべての EDU を返します。
- **eduid** の EDU ID および **apphd1=appHandle** パラメーターと共に指定すると、指定した EDU が *appHandle* アプリケーションと関連している場合に、その EDU に関する情報を返します。
- **apphd1=appHandle** パラメーターと共に指定すると、*appHandle* アプリケーションのコーディネーター・エージェントの EDU のみ返します。
- **pid** オプションと共に指定すると、特定のエージェントに対してスタック・トレース・ファイルおよびバイナリー・ダンプ・ファイルを生成します。

上記のパラメーターと共に以下のパラメーターを指定できます。

**dumpdir=directory**

スタック・ファイルのリダイレクト先ディレクトリーを指定します。絶対パスを指定する必要があり、指定ディレクトリーは存在するものでなければなりません。このオプションは、UNIX および Linux オペレーティング・システムでのみ使用可能です。

**timeout=timeout-value**

スタック・ファイルが指定ディレクトリーにリダイレクトされる時間範囲を秒数で指定します。

**-dynamic**

動的 SQL の実行に関する情報を戻します。 **-dynamic** の使用上の注意を参照してください。

**anch=anchID**

アンカー ID が指定される場合、その特定の動的 SQL に関する情報が戻されます。

**file=filename**

**-dynamic** 出力を指定のファイルに送信します。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

**-edus** インスタンス内のすべての EDU をリストします。維持されているトラップの場合、このオプションを指定すると、EDU が中断されたことを示す EDU Name が出力されます。

### **interval=interval**

UNIX オペレーティング・システムでのみ使用できます。インターバルを指定した場合は、EDU の 2 つのスナップショットが *interval* 秒の間隔を置いて取られます。その後、EDU のリストに 2 つの新しい列が組み込まれます。1 つは USR DELTA で、この列には *interval* で指定した間隔での CPU ユーザー時間の差分が表示されます。もう 1 つは SYS DELTA で、この列には *interval* で指定した間隔での CPU システム時間の差分が表示されます。*interval* で指定した間隔の途中で追加された EDU はリストに含められ、追加された時点からの差分が計算されます。*interval* で指定した間隔の途中で削除された EDU は、リストに一切含められません。

### **top=n**

*n* 個の EDU を表示するように指定します。*n* は整数値です。指定間隔内で最も多く CPU 時間を消費した EDU が 1 番目に表示されます。

### **stacks**

表示される EDU のスタックをダンプします。

**-edus** の使用上の注意を参照してください。**db2pd -edus** コマンドの出力例も参照してください。

### **-everything**

サーバーに対してローカルなすべてのデータベース・パーティション・サーバーのすべてのデータベースに対してすべてのオプションを実行します。

**-fcm** 高速コミュニケーション・マネージャーに関する情報を戻します。

- 出力の有効範囲として設定したいデータベースを選択した場合、このパラメーターに **-inst** パラメーターを指定します。
- DB2 インスタンス開始以来の、アプリケーションによる FCM バッファおよびチャネル消費量の最高水準点を取得するには、このパラメーターに **hwm** パラメーターを指定します。アプリケーションの消費量の最高水準点値は、アプリケーションが既にデータベースから切断されていても保持されます。
- **db2pd** コマンドが現行および HWM 消費量統計でレポートするアプリケーションの最大数を制限するには、このパラメーターに **numApps** オプションを指定します。

**-fcm** の使用上の注意を参照してください。

### **-file filename**

指定されたファイルに出力を書き込むことを指定します。

**-fmp** fenced ルーチンが実行されたプロセスに関する情報を戻します。**-fmp** の使用上の注意を参照してください。

### **-fixstack existing\_stack\_filename**

既存のスタック・ファイルを読み取り、同じファイル名でファイル拡張子

.fmt が追加された新規ファイルを、同じ場所に生成します。生成された新規ファイル .fmt では、スタック・トレースに含まれるフレームのいくつかに関して、シンボルの詳細が改善されています (シンボルが定義されているライブラリーが、このコマンドの実行時に使用可能である場合)。

注: Linux オペレーティング・システムでのみ適用できます。

### **-fmpexechistory | -fmpe**

ロードして実行することが試みられた fenced ルーチン履歴を表示します。このパラメーターは、フィックスパック 1 から使用可能です。

#### **pid=pid**

特定の fenced プロセス ID に関するスレッドの詳細情報を表示します。何も指定しない場合は、すべてのプロセスの詳細情報が表示されます。スレッド・セーフ FMP プロセスの場合は、スレッドごとに 1 つの実行履歴リストが表示されます。スレッドは、アクティブ・スレッド、プールされたスレッド、強制されたスレッドの 3 つのグループで示されます。非スレッド・セーフ FMP プロセスの場合は、プロセスごとに実行履歴リストが 1 つだけ表示されます。

#### **tid=tid**

特定のスレッド ID を使用しているスレッド・セーフ・ルーチンの履歴の詳細を表示します。非スレッド・セーフ・ルーチンの場合、スレッド ID 値は 1 になります。

**n=n** このオプションは、FMP プロセスごとに表示されるルーチン実行履歴の数を指定する場合に使用します。最大値は 128 です。指定されない場合は、最後のルーチン履歴のみがデフォルトで戻されます。

#### **genquery**

ルーチン固有 ID に基づいてルーチンのスキーマ、モジュール、名前、および特定名を戻す select 照会を生成します。

**-fmpexechistory | -fmpe** の使用上の注意を参照してください。

**-full** すべての出力がその最大長まで拡張されます。これが指定されない場合、出力は切り捨てられて、ディスプレイ上のスペースを節約します。

**-fvp** fenced ベンダー・プロセス情報を表示し、応答していない fenced ベンダー・プロセスを終了できます。これは、ベンダー・メディア装置が使用されているバックアップ、リストア、整理履歴、ロード、ロード・コピー (ロールフォワード)、およびログ・マネージャーに該当します。

注: 情報収集のために適切なメモリー・セットに接続するには、このパラメーターと共に **-database database** コマンド・パラメーターを使用する必要があります。

#### **agent\_eduid**

バックアップ、リストア、整理履歴、ロード、またはロード・コピー (ロールフォワード) エージェントの DB2 EDU ID に関する fenced ベンダー・プロセス情報を表示します。

**LAM1 logarchmeth1** に関する fenced ベンダー・プロセス情報を表示します。



## db2pd - DB2 データベースのモニターおよびトラブルシューティング

- LAM2** **logarchmeth2** に関する fenced ベンダー・プロセス情報を表示します。
- LAM3** 特殊なケースに関する fenced ベンダー・プロセス情報を表示します。これは、現在のログ・アーカイブ方式構成パラメーターが **VENDOR** に設定されていないため、以前のベンダー・アーカイブ方式からログを取り出すために、**ROLLFORWARD DATABASE** 中に fenced ベンダー・プロセスを一時的に作成する必要があるような場合です。
- term** このオプションを使用すると、fenced ベンダー・プロセス情報の表示に加えて、特定の fenced ベンダー・プロセスが終了されます。

**注:** これは Windows オペレーティング・システムには影響を与えません。

### **-global**

リモート・ホストに対しても **db2pd** を実行することを指定します。 **-file** パラメーターを指定した場合、リモート・ホスト上のすべての個別ファイルから成る単一ファイルが、**db2pd** コマンドが発行されたコンピューター上に作成されます。

**注:** このコマンド・パラメーターは、DB2 バージョン 9.8 フィックスパック 3 以降で使用できます。このパラメーターは、DB2 バージョン 9.7 フィックスパック 4 およびそれ以降のフィックスパックでは非推奨です。

### **-dbp database\_partition\_number**

指定したデータベース・パーティションのリモート・ホストに対して **db2pd** を実行することを指定します。 **-global** オプションでデータベース・パーティションを指定しなかった場合、メンバーが存在するホストでのみ **db2pd** が実行されます。

**-gfw** 現在アクティブな、または何らかの理由で非アクティブにされたイベント・モニターのリストを戻します。また、高速書き込みプログラムの独立コーデイネーターごとのデータを、イベント・モニターが書き込むターゲットについての統計と情報も戻します。

**-ha** 高可用性の統計を報告します。

**-hadr** 高可用性災害時リカバリー (HADR) 情報を報告します。報告される一部のエレメントの説明が、「データベースのモニタリング ガイドおよびリファレンス」の高可用性災害時リカバリーのセクションに記載されています。

**-hadr** の使用上の注意を参照してください。

### **-h | -help**

オンライン・ヘルプ情報を表示します。

### **-host hostname**

コマンド発行先ホスト (複数も可) を指定します。このコマンドは、ホスト上にあるすべてのメンバーに対して発行されます。このオプションを指定しない場合、コマンドはローカル・ホストに対して発行されます。複数のホストを指定する場合、すべてのホスト名が有効でないと、コマンドを完了できません。このオプションは、**-member** オプションと一緒に指定することはできません。

**-inst** インスタンス範囲のすべての情報を戻します。

### -interactive

**db2pd** コマンドを実行するときに、**DB2PDOPT** 環境変数に指定されている値をオーバーライドします。

### -latches

すべてのラッチ・ホルダーとすべてのラッチ・ウェイターについて報告します。

**group** ホルダーのリスト、およびその後ウェイターのリストを単純に出力します。

**file=filename**

**-latches** 出力を *filename* に送信します。

**-latche** の使用上の注意を参照してください。

**-load** すべてのロード EDU 情報を表示します。このパラメーターを、パラメーター **-host** または **-member** と組み合わせると、ホストまたはメンバーに固有のロード EDU 情報を表示することができます。アクティブなデータベースを指定する必要があります。

**loadID="LOADID"**

*LOADID* で指定した特定のロード操作のために使用されているロード EDU をすべて表示します。指定した *LOADID* が存在しない場合、情報は表示されません。

**file=filename**

出力を、スタック・ファイルを除いて、指定のファイルにリダイレクトします。

**stacks** 表示されるロード EDU のスタック・トレースを、**diagpath** ディレクトリにダンプします。このオプションを **loadID** オプションと一緒に使用すると、指定したロード操作のために使用されているロード EDU のスタックがダンプされます。このオプションは、UNIX および Linux オペレーティング・システムでのみ使用可能です。

**-load** の使用上の注意を参照してください。

**-locks** ロックに関する情報を戻します。

特定のトランザクションによって保持されているロックに関する情報を得るには、そのトランザクション・ハンドルを指定します。

ロック名に関する詳細を戻すには、**showlocks** コマンド・パラメーターを指定します。パーティション表および個々のデータ・パーティション上の行およびブロックのロックについて、**showlocks** はデータ・パーティション ID を行の一部としてロック情報と共に表示します。

待ち状態にあるロックおよびそれらのロックの所有者を戻すには、**wait** コマンド・パラメーターを指定します。

**-locks** の使用上の注意を参照してください。

**-logs** ログ・ファイルに関する情報を戻します。 **-logs** の使用上の注意を参照してください。 **db2pd -logs** コマンドの出力例も参照してください。

この情報は、**MON\_GET\_TRANSACTION\_LOG** 表関数を実行して取得することもできます。



## db2pd - DB2 データベースのモニターおよびトラブルシューティング

### **-member** *member\_number* | *member\_range*

コマンド発行先メンバー (複数も可) を指定します。このオプションを指定しない場合、コマンドは現行メンバーに対して発行されます。複数のメンバーを、*member\_number* のコンマ区切りリスト (*member1*, *member2*) として、または *member\_range* (*member\_range* はメンバーの範囲 (*member1*-*member3*)) を使用して指定できます。最初の 2 つの方式を組み合わせ使用することもできます。このオプションは、**-host** オプションと一緒に指定することはできません。

**all** リモート・ホスト上のメンバーを含め、すべてのメンバーに対してコマンドを発行するように指定します。

### **-memblocks**

メモリー・セットに関する情報を戻します。 **-memblocks** パラメーターを使用したスコープに基づいて、特定のメモリー・セットが返されます。

- このパラメーターを **-inst** および **-alldbs** パラメーターと一緒に発行すると、メモリー・セット **dbms**、**fcm**、**fmp**、**appl**、および **db** の情報が返されます。次のコマンドは、すべてのデータベースについて、インスタンス・スコープとデータベース・スコープのメモリー・セットの情報を返します。

```
db2pd -inst -alldbs -memblocks
```

- このパラメーターを **-inst** および **-db** パラメーターと一緒に発行すると、指定したデータベースについて、メモリー・セット **dbms**、**fcm**、**fmp**、**appl**、および **db** の情報が返されます。次のコマンドは、データベース *sample* について、インスタンス・スコープとデータベース・スコープのメモリー・セットの情報を返します。

```
db2pd -inst -db sample -memblocks
```

- このパラメーターをデータベース・スコープ (**-db**) で発行すると、メモリー・セット **appl** および **db** の情報が返されます。次のコマンドは、データベース *sample* について、データベース・スコープのメモリー・セットの情報を返します。

```
db2pd -db sample -memblocks
```

- このパラメーターを単独で **db2pd** コマンドに指定して発行すると、インスタンス・スコープのメモリー・セットの情報が返されます。これには、メモリー・セット **dbms**、**fcm**、および **fmp** が含まれます。これがデフォルトの動作です。次のコマンドは、インスタンス・スコープのメモリー・セットの情報を返す **db2pd -inst -memblocks** と同じ情報を返します。

```
db2pd -memblocks
```

- このパラメーターを以下のいずれかのパラメーター・オプションと一緒に発行すると、返される情報は、そのオプションのメモリー・セットの情報のみです。次のコマンドは、**fmp** メモリー・セットの情報のみを返します。

```
db2pd -memblocks -fmp
```

### **dbms**

データベース・マネージャー・システムのメモリー・セットのメモリー・ブロックのみを報告します。このメモリー・セットは、インスタンス・スコープのメモリー・セットの一部です。

- fcm** 高速コミュニケーション・マネージャーのメモリー・セットのメモリー・ブロックのみを報告します。このメモリー・セットは、インスタンス・スコープのメモリー・セットの一部です。
- fmp** fenced モード・プロセスのメモリー・セットのメモリー・ブロックのみを報告します。このメモリー・セットは、インスタンス・スコープのメモリー・セットの一部です。
- app1** アプリケーションのメモリー・セットのメモリー・ブロックのみを報告します。このメモリー・セットは、データベース・スコープのメモリー・セットの一部です。
- db** データベースのメモリー・セットのメモリー・ブロックのみを報告します。このメモリー・セットは、データベース・スコープのメモリー・セットの一部です。
- a11** すべてのメモリー・セットのメモリー・ブロックを報告します。これには、インスタンス・スコープ (**-inst**) のメモリー・セットに加え、Windows オペレーティング・システムの場合に限り専用メモリー・セットのメモリー・ブロックも含まれます。

**注:** データベース・スコープのメモリー・セット (データベースおよびアプリケーションのメモリー・セット) のメモリー・ブロックを含めるためには、データベース・スコープ (**-db** または **-alldb**) を指定する必要があります。

- top** 各セットでの最大メモリー・コンシューマーを報告します。
- blocks** 各セットのメモリー・ブロックを報告します。
- sort** 各セット内でプールごとにソートされたメモリー・ブロックを報告します。
- PoolID** 特定プールからのメモリー・ブロックを報告します。
- pid=pid**  
特定のプロセス ID からのメモリー・ブロックを報告します (UNIX オペレーティング・システムのみ)。
- private**  
専用メモリー・セットからのメモリー・ブロックを報告します (Windows オペレーティング・システムのみ)。
- skipfreedwithpool | skip**  
プールによって解放されるものをスキップしたメモリー・ブロックを報告します。

**-memblocks** の使用上の注意を参照してください。

### **-mempools**

メモリー・プールに関する情報を戻します。

戻される情報にインスタンス範囲のすべての情報を組み込むには、このオプションに **-inst** オプションを指定します。 **-mempools** の使用上の注意を参照してください。

### **-memsets**

メモリー・セットに関する情報を戻します。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

戻される情報にインスタンス範囲のすべての情報を組み込むには、このコマンド・パラメーターに **-inst** コマンド・パラメーターを指定します。  
**-memsets** の使用上の注意を参照してください。

### **-osinfo**

オペレーティング・システム情報を戻します。ディスク・パスが指定される場合、ディスクに関する情報が出力されます。 **-osinfo** の使用上の注意を参照してください。

**-pages** バッファ・プール・ページに関する情報を戻します。

*bpID* バッファ・プール ID が指定されている場合、指定されたバッファ・プールからのページだけが戻されます。

### **summary**

このオプションを指定すると、サマリー情報のセクションだけが表示されます。

**-pages** の使用上の注意を参照してください。 **db2pd -pages** コマンドの出力例も参照してください。

### **-q | -quit | q | quit**

終了します。 **db2pd** キーワードだけを単独で発行した場合、**db2pd** は対話モードで実行されます。 **quit** コマンドはこのモードを終了させて、標準のコマンド・プロンプトに戻します。

### **-quiesceinfo**

インスタンスおよびデータベースの現在の静止状況を指定します。

### **-recovery**

リカバリー・アクティビティーに関する情報を戻します。 **-recovery** の使用上の注意を参照してください。

**-reopt REOPT ONCE** オプションを使用して再度最適化された、キャッシュに入れている SQL ステートメントに関する情報を戻します。 **-reopt** の使用上の注意を参照してください。

### **-reorgs**

表およびデータ・パーティションの再編成に関する情報を戻します。 **index** パラメーターがコマンドに追加されている場合、表およびデータ・パーティションの再編成情報と共に、索引再編成情報が戻されます。

**注:** DB2 バージョン 9.8 フィックスパック 3 以降、**db2pd -reorgs index** コマンドは、DB2 V9.8 フィックスパック 3 から報告している非パーティション索引の索引再編成統計に加えて、パーティション索引の索引再編成統計も報告するようになりました。

**注:** **REORG INDEXES** コマンドの **CLEANUP ONLY** パラメーターを指定した場合は、データベースの索引再編成操作の進行状況をモニターできません。

**-reorgs** の使用上の注意を参照してください。 **db2pd -reorgs index** コマンドの出力例も参照してください。

### **-repeat num sec count**

コマンドが指定の秒数の後、繰り返されます。秒数の値が指定されない場

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

合、コマンドは 5 秒ごとに繰り返されます。出力が繰り返される回数も指定できます。 *count* に値を指定しない場合、コマンドは割り込みが入るまで繰り返されます。

### -runstats

表および関連索引に実行している RUNSTATS ユーティリティの状況に関する情報を返します。最初に表統計収集の状況が表示され、続いて索引統計集合の状況が表示されます。 -runstats オプションの出力例は以下のようになります。

```
db2pd -runstats
```

```
Table Runstats Information:
```

```
Retrieval Time: 08/13/2009 20:38:20
TbspaceID: 2      TableID: 4
Schema: SCHEMA   TableName: TABLE
Status: Completed Access: Allow write
Sampling: No     Sampling Rate: -
Start Time: 08/13/2009 20:38:16 End Time: 08/13/2009 20:38:17
Total Duration: 00:00:01
Cur Count: 0      Max Count: 0
```

```
Index Runstats Information:
```

```
Retrieval Time: 08/13/2009 20:38:20
TbspaceID: 2      TableID: 4
Schema: SCHEMA   TableName: TABLE
Status: Completed Access: Allow write
Start Time: 08/13/2009 20:38:17 End Time: 08/13/2009 20:38:18
Total Duration: 00:00:01
Prev Index Duration [1]: 00:00:01
Prev Index Duration [2]: -
Prev Index Duration [3]: -
Cur Index Start: 08/13/2009 20:38:18
Cur Index: 2      Max Index: 2      Index ID: 2
Cur Count: 0      Max Count: 0
```

### -scansharing

指定のデータベース内で表またはブロック索引のスキャン共有があるすべての表に関するスキャン共有情報を返します。

**obj=objectID pool=poolID**

指定の表に関するスキャン共有情報を返します。

**a11** すべての表のスキャン共有情報を返します。表ごとに、表または範囲のスキャン共有情報が戻ります。加えて、MDC 表の場合、ブロック索引のスキャン共有情報も戻ります。

**index=**

*indexID*

指定の表のスキャン共有情報と、指定のブロック索引のブロック索引スキャン共有情報が戻ります。

**a11** すべてのブロック索引のブロック索引スキャン・シェアリング情報を返します。

**db2pd -scansharing** コマンドの出力例を参照してください。

**-scansharing** の使用上の注意を参照してください。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

### -serverlist

どのメンバーが使用可能か、およびそれらの各メンバーの相対負荷に関する情報を返します。

次のように、1 つ以上のデータベースに関して出力が返されない場合もあります。

- アクティブなデータベースが存在しない
- 指定したデータベースがアクティブではない
- 指定したデータベースはアクティブであるが、サーバー・リストがまだキャッシュされていない
- **db2pd** コマンドが DB2 pureScale 環境以外の環境で実行されている
- リモート・クライアントがデータベースに接続していない

**-serverlist** の使用上の注意を参照してください。

**db2pd -serverlist** コマンドの出力例を参照してください。

### -serviceclasses *serviceclassID*

データベースのサービス・クラスに関する情報を戻します。 *serviceclassID* は、特定の 1 つのサービス・クラスの情報を取得するためのオプション・パラメーターです。 *serviceclassID* が指定されない場合、すべてのサービス・クラスの情報取得されます。

**-serviceclasses** の使用上の注意を参照してください。 **db2pd**

**-serviceclasses** コマンドの出力例も参照してください。

**-sort** フィックスパック 5 以降では、このオプションによって、アプリケーション・ソート操作に関する情報を戻します。アプリケーション・ハンドル ID が指定される場合、指定されたアプリケーションのソート操作に関する情報が戻されます。

**-sort** の使用上の注意を参照してください。

### -stack **a11** | **apphd1=appHandle** | *eduid*

エンジンがハングした場合は、スタック・トレース・ファイルを使用して DB2 の状態に関する情報を取得できます。このコマンドにより、**diagpath** ディレクトリ内にスタック・トレース・ファイルが生成されます。UNIX および Linux オペレーティング・システムでのファイルの命名規則は *pid.tid.node.stack.txt* です。Windows オペレーティング・システムでは、命名規則 *pid.tid.stack.bin* に従ったスタック・トレース・ファイルに EDU から情報がダンプされます。**-stack a11** は、Windows オペレーティング・システムでのみサポートされているオプションであることに注意してください。

**a11** このオプションを指定すると、現行のデータベース・パーティション内にあるすべてのプロセスに対してスタック・トレース・ファイルを生成します。

注: **a11** オプションを使用する場合には、適切に動作するために十分なメモリーが必要になります。

### **apphd1=appHandle**

このオプションを指定すると、*appHandle* に等しいアプリケーション

ン・ハンドルについてのみスタック・トレース・ファイルを生成します。このオプションは、UNIX および Linux オペレーティング・システムでのみ使用可能です。

*eduid* 特定の ID を持つ EDU だけを出力するよう制限します。フォーマット済みイベントおよび関連データは、db2dump ディレクトリー内の *pid.tid/EDUID.node.trap.txt* トラップ・ファイルにダンプされます。このオプションは、UNIX および Linux オペレーティング・システムでのみ使用可能です。

イベント・スタックは、以下の順序で出力されます。

最後のイベント (イベント・スタックの先頭)

- イベント・タイプと簡略説明
- 顧客への影響
- オブジェクト ID
- ECF ID、プローブ
- 先頭イベント・ヘッダー
- 先頭イベント修飾子 (存在する場合)
- 先頭イベント・データ (存在する場合)

最初のイベント (イベント・スタックの末尾)

- イベント・タイプと簡略説明
- 顧客への影響
- オブジェクト ID
- ECF ID、プローブ
- 末尾イベント・ヘッダー
- 末尾イベント修飾子 (存在する場合)
- 末尾イベント・データ (存在する場合)

上記のリストで、ECF ID は ECF の識別子 (*product*、*component*、*function* というフォーマット) です。プローブは、コード行または何らかの固有の数値です (関数の場合)。

イベント・フロー (記録されたイベントの「履歴」) は、以下の順序で出力されます。

最初のイベント・レコード

- イベント・タイプと簡略説明
- 顧客への影響
- オブジェクト ID
- ECF ID、プローブ
- イベント・ヘッダー
- オブジェクト・データ (ストリングまたは整数でない場合)

最後のイベント・レコード

- イベント・タイプと簡略説明
- 顧客への影響

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

- オブジェクト ID
- ECF ID、プローブ
- イベント・ヘッダー
- オブジェクト・データ (ストリングまたは整数でない場合)

### **dumpdir=directory**

スタック・ファイルのリダイレクト先ディレクトリーを指定します。絶対パスを指定する必要がある、指定ディレクトリーは存在するものでなければなりません。このオプションは、UNIX および Linux オペレーティング・システムでのみ使用可能です。

### **timeout=timeout-value**

スタック・ファイルが指定ディレクトリーにリダイレクトされる時間範囲を秒数で指定します。

### **-static**

静的 SQL の実行とパッケージに関する情報を戻します。 **-static** の使用上の注意を参照してください。

### **-statisticscache**

データベース・レベルの統計キャッシュに関する情報を戻します。

#### **summary**

統計キャッシュを要約します。データベース *sample* に関する統計キャッシュ要約をダンプ出力するには、次のコマンドを発行します。

```
db2pd -db sample -statisticscache summary
```

**detail** リアルタイム統計収集によって集められた最新の統計を含む、統計キャッシュに保管されたすべての表に関する詳しい統計情報をダンプ出力するには、このオプションを指定します。統計キャッシュに保管されたすべてのデータベースに関する詳しい統計情報をダンプ出力するには、次のコマンドを発行します。

```
db2pd -statisticscache detail -alldbs
```

#### **find schema=schema object=object**

*schema* (スキーマ名) と *object* (表名) を指定して特定の表に関する詳しい統計情報をダンプ出力するには、このオプションを指定します。データベース *sample* の表 *USER1.T1* に関する詳しい統計情報をダンプ出力するには、次のコマンドを発行します。

```
db2pd -db sample -statisticscache find schema=USER1 object=T1
```

**-statisticscache** の使用上の注意を参照してください。

### **-storagegroups**

データベースに対して定義されたストレージ・グループに関する情報を戻します。

*Storagegroup ID* コマンド・パラメーターと共に指定して、特定のストレージ・グループとそのパスに関する情報を表示します。

**-storagegroups** の使用上の注意を参照してください。 **db2pd**

**-storagegroups** コマンドの出力例も参照してください。



### **-storagepaths**

データベースに対して定義されたすべてのストレージ・グループの、自動ストレージ・パスに関する情報を戻します。 **storagegroups** パラメーターとは異なり、このパラメーターはストレージ・グループ ID を入力として受け入れません。

**-storagepaths** の使用上の注意を参照してください。 **db2pd -storagepaths** コマンドの出力例も参照してください。

### **-sysplex**

**db** パラメーターによって示されるデータベース別名に関連付けられたサーバーのリストに関する情報を戻します。 **-database** コマンド・パラメーターが指定されない場合、すべてのデータベースに関する情報が戻されます。

出力の有効範囲として設定したいデータベースを選択した場合、このコマンド・パラメーターに **-inst** コマンド・パラメーターを指定します。

**-sysplex** の使用上の注意を参照してください。

### **-tablespaces**

表スペースに関する情報を戻します。

表スペースとグループ化されているその表スペースのコンテナに関する情報を表示するには、 **group** コマンド・パラメーターを指定します。

**Table\_space\_ID** コマンド・パラメーターと一緒に指定して、特定の表スペースとそのコンテナに関する情報を表示します。

最後のバックアップまたは次のバックアップに関する表スペースの状態を表示するには、 **trackmodstate** コマンド・パラメーターを指定します。このパラメーターを使用するためには、 **trackmod** 構成パラメーターが Yes に設定されている必要があります。

**-tablespaces** の使用上の注意を参照してください。 **db2pd -tablespaces** コマンドの出力例も参照してください。

### **-tcbstats**

表と索引に関する情報を戻します。さらに、表の更新の総数、UDI とリアルタイム統計 UDI カウンター (RTSUDI) も戻します。

**TbpaceID=table\_space\_ID**

特定の表スペースに関する情報を表示するには、このオプションを指定します。

**TableID=table\_ID**

特定の表に関する情報を表示するには、このオプションを指定します。 **TableID** オプションを使用するとき、 **TbpaceID** オプションが必須となります。

### **nocatalog**

すべての非カタログ表に関連した、表および索引の情報を表示する場合に、このオプションを指定します。

**-tcbstats** の使用上の注意を参照してください。

### **-temptable**

デフォルトで、一時表に関する以下の情報を戻します。



## db2pd - DB2 データベースのモニターおよびトラブルシューティング

- **Number of Temp Tables**。データベース・マネージャー始動後、またはカウンターの最終リセット後に作成およびドロップした一時表の総数。
- **Comp Eligible Temps**。データベース・マネージャーが *query type*、*minimum row size*、および *minimum expected table size* の 3 つのプロパティに基づいて圧縮に適切であると判断した一時表。
- **Compressed Temps**。実際に圧縮された一時表の総数。これは、一時表にコンプレッション・ディクショナリーが作成できるほど十分なデータが表にあることを暗に示します。
- **Total Stored Temp Bytes**。ディスク上に格納される、一時表の実際の行データの合計数。これは、圧縮一時表と非圧縮一時表の両方からの数です。
- **Total Bytes Saved**。行圧縮によって節約された合計バイト。
- **Total Compressed Rows**。圧縮を使用して少なくとも 1 バイトが節約された行数の累積カウント。
- **Total Temp Table Rows**。圧縮されているかどうかに関わらず、一時表すべてに挿入された合計行数。圧縮一時表に挿入されている必ずしもすべての行が、圧縮されている訳ではありません。

### reset

すべてのカウンターをゼロにリセットする場合に、このオプションを指定します。

**-temptable** の使用上の注意を参照してください。**db2pd -temptable** コマンドの出力例も参照してください。

### -thresholds thresholdID

しきい値に関する情報を戻します。 *thresholdID* の指定はオプションですが、しきい値 ID を指定すると特定のしきい値に関する情報が戻されます。*thresholdID* が指定されない場合、すべての有効および無効なしきい値に関する情報が取得されます。

**-thresholds** の使用上の注意を参照してください。**db2pd -thresholds** コマンドの出力例を参照してください。

### -totalmem

DB2 ホストに割り振られているメモリーの合計量に関する情報を戻します。具体的には次のとおりです。

- ホスト上で事前割り振りされた、予約済みの *restart light* メモリーの量。
- ホストの常駐メンバーおよびゲスト・メンバーによる合計メモリー消費量。

**-totalmem** オプションは、現在アクセスされているホストについての情報のみを報告します。

### -transactions

アクティブなトランザクションに関する情報を戻します。トランザクション・ハンドルが指定される場合、そのトランザクション・ハンドルに関する情報が戻されます。アプリケーション・ハンドルが指定される場合、そのトランザクションのアプリケーション・ハンドルに関する情報が戻されます。

**-transactions** の使用上の注意を参照してください。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

### **-utilities**

ユーティリティー情報を報告します。報告される各エレメントについての説明は、「データベースのモニタリング ガイドおよびリファレンス」のユーティリティーのセクションにあります。

**-utilities** の使用上の注意を参照してください。

### **-v | -version**

インストールされた DB2 データベース製品の現行のバージョンとサービス・レベルを表示します。

### **-wlocks**

待機対象となっている各ロックについての所有者および待機者の情報を表示します。 **db2pd -wlocks** コマンドの出力例で、ロック状況 (Sts) 値の G はロックの所有者を指定し、Sts 値の W はそのロックの待機者を指定します。

**file=filename**

**-wlocks** 出力を指定のファイルに送信します。

**-wlocks** の使用上の注意を参照してください。

### **-workactionsets** *workactionsetID*

すべての有効な作業アクション・セットに関する情報、およびこれらのセット内にあるすべての有効なワーク・アクションに関する情報を返します。

**group** 作業アクション・セットでグループ化された同じ情報を返します。

**-workactionsets** の使用上の注意を参照してください。

### **-workclasssets** *workclasssetID*

有効な作業アクション・セットによって参照されているすべての作業クラス・セットに関する情報、およびこれらの作業クラス・セット内のすべての作業クラスに関する情報を返します。

**group** 作業クラス・セットでグループ化された同じ情報を返します。

**db2pd -workclasssets** コマンドの出力例を参照してください。**-workclasssets** の使用上の注意を参照してください。

### **-workloads** *workloadID*

コマンド実行時にメモリー内にあるワークロード定義、ユーザー特権の所有者、およびローカル・パーティションのワークロード統計のリストを返します。

**group** ワークロードでグループ化された同じ情報を返します。

**db2pd -workloads** コマンドの出力例を参照してください。

**-workloads** の使用上の注意を参照してください。

## 例

クライアント要求にサービスを提供しているエージェントに関する情報を得るには、以下のようにしてコマンド行から **db2pd** コマンドを使用します。

```
db2pd -agents
```

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

クライアント要求にサービスを提供しているエージェントに関する情報を得るには、以下のようにしてコマンド行から **db2pd** コマンドを使用します。この例では、**db2pd** コマンドを呼び出す前に、**-agents** パラメーターを指定して **DB2PDOPT** 環境変数が設定されます。このコマンドは、実行時に環境変数に設定される情報を使用します。

```
export DB2PDOPT="-agents"  
db2pd
```

クライアント要求にサービスを提供しているエージェントに関する情報を得るには、以下のようにしてコマンド行から **db2pd** コマンドを使用します。この例では、**db2pd** コマンドを呼び出す前に、**-agents** パラメーターが **file.out** ファイル内で設定されます。 **-command** パラメーターが指定されているため、コマンドはその実行時に **file.out** ファイルの情報を使用します。

```
echo "-agents" > file.out  
db2pd -command file.out
```

すべてのデータベースとインスタンス範囲の情報を得るには、以下のようにしてコマンド行から **db2pd** コマンドを使用します。

```
db2pd -inst -alldbs
```

**fenced** ベンダー・プロセス状態の情報を取得するには、コマンド行から **db2pd -fvp** コマンドを次のように使用します。

### ログ・マネージャーの場合:

- **SAMPLE** というデータベースの **logarchmeth1** が TSM に設定されています。任意の時点で、以下を発行します。

```
db2pd -db sample -fvp lam1
```

その結果、以下のように出力されます。

```
-----  
Fenced ベンダー・プロセス状態情報:  
-----
```

```
ログ・マネージャー:  
-----
```

```
LOGARCHMETH1 available.
```

```
Vendor EDU is available and running.  
  startTime: 1155581841 20060814145721  
  function: sqluvint
```

これは、**fenced** ベンダー・プロセスがベンダー関数 **sqluvint** で 2006 年 8 月 14 日 14:57 から実行中であることを示しています。ここで、この実行期間が長すぎると思われる場合、あるいはこのプロセスが TSM リソースを待機して停止状態にあると判断される場合には、以下を発行することにより **fenced** ベンダー・プロセスを終了できます。

```
db2pd -db sample -fvp lam1 term
```

その結果、以下のように出力されます。

```
-----  
Fenced ベンダー・プロセス状態情報:  
-----
```

```
ログ・マネージャー:
```

```
-----  
LOGARCHMETH1 available.
```

```
Vendor EDU is available and running.  
  startTime: 1155581841 20060814145721  
  function: sqluvint  
This fenced vendor process has been sent a signal to terminate.
```

ここでは、前の出力と同じ情報が示されているのに加えて、終了要求が送信されたことがわかります。少し待てば、要求の効果が現れます。

- fenced ベンダー・プロセスが実行中であっても、ベンダー・コードで実行されていない場合には、通常の表示要求として以下のように表示されます。

```
-----  
Fenced ベンダー・プロセス状態情報:  
-----
```

```
ログ・マネージャー:
```

```
-----  
LOGARCHMETH1 available.
```

```
Vendor EDU is available and running.  
No vendor code being run.
```

### バックアップの場合:

注: 以下のセクションに示す方法の代わりに、**FORCE APPLICATION** コマンドを使用できることに留意してください。

- 2 つのセッションを使って **SAMPLE** というデータベースを **TSM** にバックアップしようとしています。 **db2pd -edus** または **DB2** 診断ログを介して、バックアップ・エージェント **EDU ID** を見つける必要があります。これが見つかったら、以下を発行できます。

```
db2pd -db sample -fvp 149
```

その結果、以下のように出力されます。

```
-----  
Fenced ベンダー・プロセス状態情報:  
-----
```

```
バックアップ:
```

```
-----  
メディア・コントローラー:  
-----
```

```
  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.
```

これによると、**DB2** メディア・コントローラー 0 (**EDU ID: 504**) はベンダー・コードで実行中です。一方、**DB2** メディア・コントローラー 1 (**EDU ID: 505**) には fenced ベンダー・プロセスがありますが、ベンダー・コードを実行していません。ここで、この実行期間が長すぎると思われる場合、あるいはこのプロセス

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

が TSM リソースを待機して停止状態にあると判断される場合には、以下を発行することにより fenced ベンダー・プロセスを終了できます。

```
db2pd -db sample -fvp 149 term
```

その結果、以下のように出力されます。

```
-----  
Fenced ベンダー・プロセス状態情報:  
-----
```

```
バックアップ:  
-----
```

```
メディア・コントローラー:  
-----
```

```
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.
```

ここでは前の出力と同じ情報が示されていますが、両方の fenced ベンダー・プロセスに終了要求が送られて、間もなく終了することが示されています。

### 使用上の注意

以下のセクションでは、db2pd の各種パラメーターを指定した場合に生成される出力について説明します。

- **-activestatements**
- **-agents**
- **-apinfo**
- **-applications**
- **-bufferpools**
- **-catalogcache**
- **-dbcfg**
- **-dbmcfg**
- **-dynamic**
- **-edus**
- **-fcm**
- **-fmp**
- **-fmpexechistory | -fmpe**
- **-hadr**
- **-latches**
- **-load**
- **-locks**
- **-logs**
- **-memblocks**
- **-mempools**

- -memsets
- -osinfo
- -pages
- -recovery
- -reopt
- -reorgs
- -scansharing
- -serviceclasses
- -sort
- -static
- -statisticscache
- -storagegroups
- -storagepaths
- -sysplex
- -tablespaces
- -tcbstats
- -temptable
- -thresholds
- -transactions
- -utilities
- -wlocks
- -workactionsets
- -workclasssets
- -workloads

### **-activestatement** パラメーター

**-activestatement** パラメーターについては、以下の情報が戻されます。

#### **Address**

現行アクティビティのアドレス。

#### **AppHandl**

アプリケーション・ハンドル

**UOW-ID** 実行の開始時の UOW-ID。

**StmtID** UOW-ID 内のステートメントのアクティビティ ID。

**AnchID** ステートメントのアンカー ID。

#### **StmtUID**

アンカー内のステートメントのユニーク ID。

**EffISO** 有効分離レベル。

#### **EffLockTOut**

開始時の有効ロック・タイムアウト。

#### **EffDegree**

開始時の有効 SMP 並列性の度合い。

#### **StartTime**

ステートメントが実行されたときの開始時刻。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

### LastRefTime

最後のアプリケーション参照時刻。

### -agents パラメーター

-agents パラメーターについては、以下の情報が戻されます。

### AppHandl

アプリケーション・ハンドル (ノードと索引を含む)。

### AgentPid

エージェント・プロセスのプロセス ID。

### Priority

エージェントの優先順位。

**Type** エージェントのタイプ。

**State** エージェントの状態。

### ClientPid

クライアント・プロセスのプロセス ID。

**Userid** エージェントを実行するユーザー ID。

### ClientNm

クライアント・プロセスの名前。

### Rowsread

エージェントによって読み取られた行数。

### Rowswrtn

エージェントによって書き込まれた行数。

**LkTmOut** エージェントのロック・タイムアウト設定。

### LastAppID

プールされたエージェントが最後に処理したアウトバウンド・アプリケーション ID。

### LastPooled

エージェントがプールされたタイム・スタンプ。

event オプションが **-agents** パラメーターと一緒に指定されると、以下の追加情報が戻ります。この情報を使用して、エージェントが同じタスクを引き続き処理するのか、エージェントが次のタスクに移行するのかを判別します。

### AGENT\_STATE\_LAST\_UPDATE\_TIME(ティック値)

エージェントによって処理されているイベントが最後に変更された時刻。エージェントによって現在処理されているイベントは、EVENT\_STATE、EVENT\_TYPE、EVENT\_OBJECT、およびEVENT\_OBJECT\_NAME 列で示されます。

### EVENT\_STATE

このエージェントによって最後に処理されたイベントの状態。可能な値はEXECUTING と IDLE です。

**EVENT\_TYPE**

このエージェントによって最後に処理されたイベントのタイプ。可能な値は ACQUIRE、PROCESS、および WAIT です。

**EVENT\_OBJECT**

このエージェントによって最後に処理されたイベントのオブジェクト。可能な値は、

COMP\_DICT\_BUILD、IMPLICIT\_REBIND、INDEX\_RECREATE、LOCK、LOCK\_ESCALATION、QP\_QUEUE、REMOTE\_REQUEST、REQUEST、ROUTINE、WLM\_QUEUE です。

**EVENT\_OBJECT\_NAME**

イベント・オブジェクト名。EVENT\_OBJECT の値が LOCK である場合、この列の値は、エージェントが待機するロックの名前です。

EVENT\_OBJECT の値が WLM\_QUEUE である場合、この列の値は、エージェントがキューに入れられている WLM しきい値の名前です。それ以外の場合、値は NULL です。

EVENT\_STATE、EVENT\_TYPE、EVENT\_OBJECT および EVENT\_OBJECT\_NAME 列値の可能な組み合わせを、以下の表にリストします。

表 52. EVENT\_STATE、EVENT\_TYPE、EVENT\_OBJECT および EVENT\_OBJECT\_NAME 列値の可能な組み合わせ

イベント記述	EVENT_STATE 値	EVENT_TYPE 値	EVENT_OBJECT 値	EVENT_OBJECT_NAME 値
ロックの獲得	IDLE	ACQUIRE	LOCK	ロック名
ロックのエスカレート	EXECUTING	PROCESS	LOCK_ESCALATION	NULL
要求の処理	EXECUTING	PROCESS	REQUEST	NULL
新規要求の待機	IDLE	WAIT	REQUEST	NULL
リモート・パーティションで処理される要求の待機	IDLE	WAIT	REMOTE_REQUEST	NULL
WLM threshold キューの待機	IDLE	WAIT	WLM_QUEUE	しきい値名
ルーチンの処理	EXECUTING	PROCESS	ROUTINE	NULL
索引の再作成	EXECUTING	PROCESS	INDEX_RECREATE	NULL
コンプレッション・ディクショナリーの作成	EXECUTING	PROCESS	COMP_DICT_BLD	NULL
暗黙的な再バインド	EXECUTING	PROCESS	IMPLICIT_REBIND	NULL

**-apinfo パラメーター**

-apinfo パラメーターについては、以下の情報が戻されます。

**AppHand1**

アプリケーション・ハンドル (ノードと索引を含む)。

**Application PID**

アプリケーションのプロセス ID。

**Application Node Name**

アプリケーション・ノードの名前。



## db2pd - DB2 データベースのモニターおよびトラブルシューティング

### IP Address

データベース接続の確立に使用された IP アドレス。

### Connection Start Time

アプリケーションの接続が開始したときのタイム・スタンプ。

### Client User ID

クライアント・ユーザー ID。

### System Auth ID

これは、接続のシステム許可 ID です。

### Coordinator EDU ID

アプリケーションのコーディネーター・エージェントの EDU ID。

### Coordinator Partition

アプリケーションのコーディネーター・エージェントのパーティション番号。

### Number of Agents

アプリケーションに代わって機能するエージェントの数。

### Locks timeout value

アプリケーションのロック・タイムアウト値。

### Locks Escalation

ロック・エスカレーション・フラグは、アプリケーションによって使用されるフラグがエスカレートされているかどうかを示します。

### Workload ID

ワークロード ID。

### Workload Occurrence ID

ワークロード・オカレンス ID。

### Trusted Context

接続が暗黙的のトラステッド接続または明示的のトラステッド接続である場合の、接続に関連したトラステッド・コンテキストの名前。

### Connection Trust Type

接続の信頼タイプ。これは非トラステッド接続、暗黙的なトラステッド接続、または明示的なトラステッド接続のいずれか 1 つです。

### Role Inherited

これは、トラステッド接続を介して継承されたロールです (存在する場合)。

### Application Status

アプリケーションの状況。

### Application Name

アプリケーションの名前。

### Application ID

アプリケーション ID。この値は `appl_id` モニター・エレメント・データと同じです。この値を解釈する方法について、詳しくは『`appl_id` アプリケーション ID : モニター・エレメント』を参照してください。

**UOW-ID** アプリケーションの現行 UOW の ID。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

### Activity ID

UOW 内のアクティビティ ID。

### Package Schema

パッケージ・スキーマ。

### Package Name

パッケージ名。

### Package Version

パッケージ・バージョン。

### Section Number

SQL ステートメントのセクション番号。

### SQL Type

SQL のタイプ: 動的または静的。

### Isolation

アプリケーションに設定された分離モード。

### Statement Type

ステートメント操作のタイプ。DML や DDL など。

### Statement

SQL ステートメント。

### ClientUserID

トランザクションのクライアント・ユーザー ID。これは **tpmon\_client\_userid** (TP Monitor Client User ID モニター・エレメント) と同じです。

### ClientWrkstnName

トランザクションのクライアント・ワークステーション名。これは **tpmon\_client\_wrkstn** (TP Monitor Client Workstation Name モニター・エレメント) と同じです。

### ClientApp1Name

トランザクションを実行しているクライアント・アプリケーション名。これは **tpmon\_client\_app** (TP Monitor Client Application モニター・エレメント) と同じです。

### ClientAcctng

トランザクションを実行しているクライアントのアカウントリング・ストリング。これは **tpmon\_acc\_str** (TP Monitor Client Accounting String モニター・エレメント) と同じです。

**db2pd -apinfo** コマンドの出力例を参照してください。

## **-applications** パラメーター

**-applications** パラメーターについては、以下の情報が戻されます。

### App1Hand1

アプリケーション・ハンドル (ノードと索引を含む)。

### NumAgents

アプリケーションに代わって機能するエージェントの数。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

### CoorPid

アプリケーションのコーディネーター・エージェントのプロセス ID。

**Status** アプリケーションの状況。

**Appid** アプリケーション ID。この値は **appl\_id** モニター・エレメント・データと同じです。この値の解釈の方法の詳細については、**appl\_id** モニター・エレメントの資料を参照してください。

### ClientIPAddress

データベース接続の確立に使用された IP アドレス。

### EncryptionLvl

接続によって使用されるデータ・ストリーム暗号化。NONE、LOW、または HIGH のいずれかです。NONE は、データ・ストリーム暗号化が使用されていないことを示します。LOW は、データベース・サーバー **authentication** タイプが **DATA\_ENCRYPT** に設定されていることを示します。HIGH は、SSL が使用されていることを示します。

### SystemAuthID

これは、接続のシステム許可 ID です。

### ConnTrustType

接続の信頼タイプ。これは非トラステッド、暗黙的なトラステッド接続、または明示的なトラステッド接続のいずれか 1 つです。

### TrustedContext

接続が暗黙的なトラステッド接続または明示的なトラステッド接続である場合の、接続に関連したトラステッド・コンテキストの名前。

### RoleInherited

これは、トラステッド接続を介して継承されたロールです (存在する場合)。

## -bufferpools パラメーター

**-bufferpools** パラメーターについては、以下の情報が戻されます。

### First Active Pool ID

最初のアクティブ・バッファ・プールの ID。

### Max Bufferpool ID

すべてのアクティブ・バッファ・プールの最大 ID。

### Max Bufferpool ID on Disk

ディスクに定義されているすべてのバッファ・プールの最大 ID。

### Num Bufferpools

使用できるバッファ・プールの数。

**ID** バッファ・プールの ID。

**Name** バッファ・プールの名前。

**PageSz** バッファ・プール・ページのサイズ。

### PA-NumPgs

バッファ・プールのページ領域にあるページ数。

### **BA-NumPgs**

バッファーク・プールのブロック領域にあるページ数。バッファーク・プールでブロック・ベースの入出力を使用できない場合、この値は 0 です。

### **BlkSize**

バッファーク・プールのブロック領域にあるブロックのブロック・サイズ。バッファーク・プールでブロック・ベースの入出力を使用できない場合、この値は 0 です。

### **NumTbsp**

バッファーク・プールを使用する表スペースの数。

### **PgsLeft**

バッファーク・プールのサイズが減少している場合、そのバッファーク・プールに残されている削除されるページ数。

### **CurrentSz**

ページ内のバッファーク・プールの現行サイズ。

### **PostAlter**

バッファーク・プールが再度開始される時の、ページ内のバッファーク・プールのサイズ。

### **SuspndTSCt**

バッファーク・プールにマップされる、現在 I/O が中断されている表スペースの数。すべてのバッファーク・プールに関して 0 が戻される場合、データベース I/O は中断されていません。

### **Automatic**

セルフチューニング自動状況を表示します。「True」は、このバッファーク・プールのセルフチューニングが有効になっていることを意味します。「False」は、このバッファーク・プールのセルフチューニングが有効ではないことを意味します。

### **DatLRds**

バッファーク・プール・データ論理読み取り。REGULAR および LARGE 表スペースに対して、バッファーク・プール (論理) から要求されたデータ・ページの数を示します。

### **DatPRds**

バッファーク・プール・データ物理読み取り。REGULAR および LARGE 表スペースに対して、表スペース・コンテナ (物理) から読み取られたデータ・ページの数を示します。

### **HitRatio**

公式 1 ( $\text{DatPRds} / \text{DatLRds}$ ) の使用による、バッファーク・プール内のデータ・ページのヒット率。

### **TmpDatLRds**

バッファーク・プール一時データ論理読み取り。TEMPORARY 表スペースに対して、バッファーク・プール (論理) から要求されたデータ・ページの数を示します。

### **TmpDatPRds**

バッファークール一時データ物理読み取り。TEMPORARY 表スペースに対して、表スペース・コンテナ (物理) から読み取られたデータ・ページの数を示します。

### **HitRatio**

公式 1 ( $\text{TmpDatPRds} / \text{TmpDatLRds}$ ) の使用による、バッファークール内の一時データ・ページのヒット率。

### **IdxLRds**

バッファークール索引論理読み取り。REGULAR および LARGE 表スペースに対して、バッファークール (論理) から要求された索引ページの数を示します。

### **IdxPRds**

バッファークール索引物理読み取り。REGULAR および LARGE 表スペースに対して、表スペース・コンテナ (物理) から読み取られた索引ページの数を示します。

### **HitRatio**

公式 1 ( $\text{IdxPRds} / \text{IdxLRds}$ ) の使用による、バッファークール内の索引ページのヒット率。

### **TmpIdxLRds**

バッファークール一時索引論理読み取り。TEMPORARY 表スペースに対して、バッファークール (論理) から要求された索引ページの数を示します。

### **TmpIdxPRds**

バッファークール一時索引物理読み取り。TEMPORARY 表スペースに対して、表スペース・コンテナ (物理) から読み取られた索引ページの数を示します。

### **HitRatio**

公式 1 ( $\text{TmpIdxPRds} / \text{TmpIdxLRds}$ ) の使用による、バッファークール内の一時索引ページのヒット率。

### **DataWrts**

バッファークール・データ書き込み。バッファークールのデータ・ページがディスクに物理的に書き込まれた回数を示します。

### **IdxWrts**

バッファークール索引書き込み。バッファークールの索引ページがディスクに物理的に書き込まれた回数を示します。

**DirRds** データベースからの直接読み取り。バッファークールを使用しない読み取り操作の回数。

### **DirRdReqs**

直接読み取り要求。データの 1 つ以上のセクターを直接読み取る要求の回数。

### **DirRdTime**

直接読み取り時間。直接読み取りを実行するために必要な経過時間 (ミリ秒)。

### **DirWrts**

データベースへの直接書き込み。バッファー・プールを使用しない書き込み操作の回数。

### **DirWrtReqs**

直接書き込み要求。データの 1 つ以上のセクターを直接書き込む要求の回数。

### **DirWrtTime**

直接書き込み時間。直接書きこみを実行するために必要な経過時間 (ミリ秒)。

### **AsDatRds**

バッファー・プール非同期データ読み取り。すべてのタイプの表スペースに対して、非同期エンジン・ディスパッチ可能単位 (EDU) によって表スペース・コンテナ (物理) から読み取られたデータ・ページの数を示します。

### **AsDatRdReq**

バッファー・プール非同期読み取り要求。非同期読み取り要求の数。

### **AsIdxRds**

バッファー・プール非同期索引読み取り。すべてのタイプの表スペースに対して、非同期エンジン・ディスパッチ可能単位 (EDU) によって表スペース・コンテナ (物理) から読み取られた索引ページの数を示します。

### **AsIdxRdReq**

バッファー・プール非同期索引読み取り要求。索引ページに対する非同期読み取り要求の数。

### **AsRdTime**

バッファー・プール非同期読み取り時間。すべてのタイプの表スペースに対して、非同期エンジン・ディスパッチ可能単位 (EDU) によって表スペース・コンテナ (物理) からデータ・ページおよび索引ページを読み取るために要した合計時間を示します。この値はマイクロ秒単位で示されます。

### **AsDatWrts**

バッファー・プール非同期データ書き込み。非同期ページ・クリーナーまたはプリフェッチャーのどちらかによって、バッファー・プールのデータ・ページがディスクに物理的に書き込まれた回数。プリフェッチャーは、プリフェッチするページ用のスペースを作るために、ディスクにダーティー・ページを書き込んでいることがあります。

### **AsIdxWrts**

バッファー・プール非同期索引書き込み。非同期ページ・クリーナーまたはプリフェッチャーのどちらかによって、バッファー・プールの索引ページがディスクに物理的に書き込まれた回数。プリフェッチャーは、プリフェッチするページ用のスペースを作るために、ディスクにダーティー・ページを書き込んでいることがあります。

### **AsWrtTime**

バッファー・プール非同期書き込み時間。データベース・マネージャーのページ・クリーナーによって、バッファー・プールからディスクにデータ・ページまたは索引ページを書き込むために要した合計経過時間。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

### TotRdTime

バッファ・プール物理読み取り時間の合計。すべてのタイプの表スペースに対して、表スペース・コンテナ (物理) からデータ・ページおよび索引ページを読み取るために要した合計時間を示します。この値はマイクロ秒単位で示されます。

### TotWrtTime

バッファ・プール物理書き込み時間の合計。バッファ・プールからディスクにデータ・ページまたは索引ページを物理的に書き込むために要した合計時間を示します。経過時間はマイクロ秒単位で示されます。

### VectIORds

ベクトル化入出力によって読み取られたページ数の合計。ベクトル化入出力によってバッファ・プールのページ領域に読み取られた合計ページ数。

### VectIOReq

ベクトル化入出力要求数。ベクトル化した入出力の要求の数。さらに具体的には、DB2 データベース製品がページをバッファ・プールのページ領域に順次プリフェッチする回数。

### BlockIORds

ブロック入出力によって読み取られたページ数の合計。ブロック入出力によってバッファ・プールのブロック領域に読み取られた合計ページ数。

### BlockIOReq

ブロック入出力要求数。ブロック入出力の要求の数。さらに具体的には、DB2 データベース製品がページをバッファ・プールのブロック領域に順次プリフェッチする回数。

### PhyPgMaps

物理ページ・マップ数。物理ページのマップの数。

### FilesClose

閉じられたデータベース・ファイル。閉じられたデータベース・ファイルの総数。

### NoVictAvl

ビクティム・バッファのないバッファ・プール。事前選択されたビクティム・バッファをエージェントが使用できなかった回数。

### UnRdPFetch

未読プリフェッチ・ページ。プリフェッチャーが読み取ったページで、一度も使用されなかったページの数。

## -catalogcache パラメーター

-catalogcache パラメーターについては、以下の情報が戻されます。

### Catalog Cache:

#### Configured Size

**catalogcache\_sz** データベース構成パラメーターで指定したバイト数。

#### Current Size

カタログ・キャッシュで使用される現在のバイト数。

**Maximum Size**

キャッシュに使用できるメモリの最大量 (データベース・グローバル・メモリの最大量まで)。

**High Water Mark**

処理中の最大物理サイズ。

**SYSTABLES:**

**Schema** 表のスキーマ修飾子。

**Name** 表の名前。

**Type** 表のタイプ。

**TableID**

表 ID。

**TbpaceID**

表が置かれている表スペースの ID。

**LastRefID**

表を参照した最後のプロセス ID。

**CatalogCache LoadingLock**

キャッシュ項目のロックをロードするカタログ・キャッシュの名前。

**CatalogCache UsageLock**

キャッシュ項目の使用ロックの名前。

**Sts** 項目の状況。可能な値は次のとおりです。

- V (有効)。
- I (無効)。
- S (ソフト無効。リアルタイム統計収集によって統計が更新されたとき、カタログ・キャッシュ項目はソフト無効 になります。データベース・エージェントでこのようなカタログ・キャッシュ項目を引き続き使用できますが、新しいカタログ・キャッシュ要求には無効です。ソフト無効になった項目が使用されなくなると、除去されます。新しいカタログ・キャッシュ要求では有効な項目が使用されます。)

**SYSRTNS:****RoutineID**

ルーチン ID。

**Schema** ルーチンのスキーマ修飾子。

**Name** ルーチンの名前。

**LastRefID**

ルーチンを参照した最後のプロセス ID。

**CatalogCache LoadingLock**

キャッシュ項目のロックをロードするカタログ・キャッシュの名前。

**CatalogCache UsageLock**

キャッシュ項目の使用ロックの名前。



## db2pd - DB2 データベースのモニターおよびトラブルシューティング

- Sts** 項目の状況。可能な値は次のとおりです。
- V (有効)。
  - I (無効)。

### SYSRTNS\_PROCSCHEMAS:

**RtnName**

ルーチンの名前。

**ParmCount**

ルーチン内のパラメーターの数。

**LastRefID**

PROCSCHEMAS 項目を参照した最後のプロセス ID。

**CatalogCache LoadingLock**

キャッシュ項目のロックをロードするカタログ・キャッシュの名前。

**CatalogCache UsageLock**

キャッシュ項目の使用ロックの名前。

- Sts** 項目の状況。可能な値は次のとおりです。
- V (有効)。
  - I (無効)。

### SYSDATATYPES:

**TypID** タイプ ID。

**LastRefID**

タイプを参照した最後のプロセス ID。

**CatalogCache LoadingLock**

キャッシュ項目のロックをロードするカタログ・キャッシュの名前。

**CatalogCache UsageLock**

キャッシュ項目の使用ロックの名前。

- Sts** 項目の状況。可能な値は次のとおりです。
- V (有効)。
  - I (無効)。

### SYSCODEPROPERTIES:

**LastRefID**

SYSCODEPROPERTIES 項目を参照する最後のプロセス ID。

**CatalogCache LoadingLock**

キャッシュ項目のロックをロードするカタログ・キャッシュの名前。

**CatalogCache UsageLock**

キャッシュ項目の使用ロックの名前。

- Sts** 項目の状況。可能な値は次のとおりです。
- V (有効)。
  - I (無効)。

**SYSNODEGROUPS:**

**PMapID** 分散マップ ID。

**RBalID** データ再分散に使用された分散マップの ID。

**CatalogCache LoadingLock**

キャッシュ項目のロックをロードするカタログ・キャッシュの名前。

**CatalogCache UsageLock**

キャッシュ項目の使用ロックの名前。

**Sts** 項目の状況。可能な値は次のとおりです。

- V (有効)。
- I (無効)。

**SYSDBAUTH:**

**AuthID** 許可 ID (*authid*)。

**AuthType**

許可タイプ。

**LastRefID**

キャッシュ項目を参照する最後のプロセス ID。

**CatalogCache LoadingLock**

キャッシュ項目のロックをロードするカタログ・キャッシュの名前。

**SYSRTNAUTH:**

**AuthID** 許可 ID (*authid*)。

**AuthType**

許可タイプ。

**Schema** ルーチンのスキーマ修飾子。

**RoutineName**

ルーチンの名前。

**RtnType**

ルーチンのタイプ。

**CatalogCache LoadingLock**

キャッシュ項目のロックをロードするカタログ・キャッシュの名前。

**SYSROLEAUTH:**

**AuthID** 許可 ID (*authid*)。

**AuthType**

許可タイプ。

**Roleid** ロール ID (許可 ID がロールである場合)。

**LastRefID**

キャッシュ項目を参照する最後のプロセス ID。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

### CatalogCache LoadingLock

キャッシュ項目のロックをロードするカタログ・キャッシュの名前。

### TABLESPACES:

**Schema** 表のスキーマ修飾子。

**Name** 表の名前。

**Type** 表のタイプ。

### TableID

表 ID。

### TbpaceID

表が置かれている表スペースの ID。

### LastRefID

表を参照した最後のプロセス ID。

### CatalogCache LoadingLock

キャッシュ項目のロックをロードするカタログ・キャッシュの名前。

### CatalogCache UsageLock

キャッシュ項目の使用ロックの名前。

**Sts** 項目の状況。可能な値は次のとおりです。

- V (有効)。
- I (無効)。
- S (ソフト無効。リアルタイム統計収集によって統計が更新されたとき、カタログ・キャッシュ項目はソフト無効 になります。データベース・エージェントでこのようなカタログ・キャッシュ項目を引き続き使用できますが、新しいカタログ・キャッシュ要求には無効です。ソフト無効になった項目が使用されなくなると、除去されます。新しいカタログ・キャッシュ要求では有効な項目が使用されます。)

**db2pd -catalogcache** コマンドの出力例を参照してください。

### -dbcfg パラメーター

**-dbcfg** パラメーターについては、データベース構成パラメーターの現在の値が戻されます。

### -dbmcfg パラメーター

**-dbmcfg** パラメーターについては、データベース・マネージャー構成パラメーターの現在の値が戻されます。

### -dynamic パラメーター

**-dynamic** パラメーターについては、以下の情報が戻されます。

動的キャッシュ:

**Current Memory Used**

パッケージ・キャッシュによって使用されるバイト数。

**Total Heap Size**

パッケージ・キャッシュに内部的に構成されるバイト数。

**Cache Overflow flag state**

パッケージ・キャッシュがオーバーフロー状態にあるかどうかを示すフラグ。

**Number of references**

パッケージ・キャッシュの動的な部分が参照された回数。

**Number of Statement Inserts**

パッケージ・キャッシュへのステートメント挿入の数。

**Number of Statement Deletes**

パッケージ・キャッシュからのステートメントの削除の数。

**Number of Variation Inserts**

パッケージ・キャッシュへのバリエーション挿入の数。

**Number of statements**

パッケージ・キャッシュ内のステートメントの数。

**動的 SQL ステートメント:**

**AnchID** ハッシュ・アンカー ID。

**StmtID** ステートメント ID。

**NumEnv** ステートメントに属する環境の数。

**NumVar** ステートメントに属するバリエーションの数。

**NumRef** ステートメントが参照された回数。

**NumExe** ステートメントが実行された回数。

**Text** SQL ステートメントのテキスト。

**動的 SQL 環境:**

**AnchID** ハッシュ・アンカー ID。

**StmtID** ステートメント ID。

**EnvID** 環境 ID。

**Iso** 環境の分離レベル。

**QOpt** 環境の照会最適化レベル。

**Blk** 環境のブロッキング因数。

**動的 SQL バリエーション:**

**AnchID** ハッシュ・アンカー ID。

**StmtID** このバリエーションのステートメント ID。

**EnvID** このバリエーションの環境 ID。

**VarID** バリエーション ID。

**NumRef** このバリエーションが参照された回数。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

**Typ** バリエーション・セクションの内部ステートメント・タイプ。

**Lockname**

バリエーション・ロック名。

**Val** バリエーション有効フラグ。以下の値を使用することができます。

**Y** オブジェクトは有効です。

**N** オブジェクトは無効です。

**X** オブジェクトは操作不能です。

**?** オブジェクトの再検査が必要です。

**Insert Time**

バリエーションがパッケージ・キャッシュに挿入された時刻。

**Sect Size**

セクション・データの長さ。

### -edus パラメーター

-edus パラメーターについては、以下の情報が戻されます。

**EDU ID** エンジン・ディスパッチ可能単位 (EDU) のユニーク ID。Linux オペレーティング・システムの場合を除き、EDU ID はスレッド ID にマップされます。Linux オペレーティング・システムでは、EDU ID は DB2 生成によるユニーク ID です。

**TID** スレッド ID。Linux オペレーティング・システムの場合を除き、スレッド ID は特定のスレッドのユニーク ID です。Linux オペレーティング・システムでは、これは DB2 生成によるユニーク ID です。

**Kernel TID**

サービス中のオペレーティング・システム・カーネル・スレッドのユニーク ID。

**EDU Name**

EDU の DB2 固有の名前。

**USR** EDU によって消費される合計 CPU ユーザー時間。

**SYS** EDU によって消費される合計 CPU システム時間。

**USR Delta**

指定された時間間隔にわたっての CPU ユーザー時間の差分を示します。

**SYS Delta**

指定された時間間隔にわたっての CPU システム時間の差分を示します。

db2pd -edus コマンドの出力例を参照してください。

### -fcm パラメーター

-fcm パラメーターについては、以下の情報が戻されます。

**FCM 使用統計**

**Total Buffers**

空いているバッファおよび使用中のバッファを含むバッファの総数。

**Free Buffers**

空いているバッファの数。

**Buffers LWM**

空いているバッファの最低数。

**Max Buffers**

インスタンス開始時に予約された仮想メモリー量に基づいて割り振ることができるバッファの最大数。

**Total Channels**

空いているチャンネルおよび使用中のチャンネルを含むチャンネルの総数。

**Free Channels**

空いているチャンネルの数。

**Channels LWM**

空いているチャンネルの最低数。

**Max Channels**

インスタンス開始時に予約された仮想メモリー量に基づいて割り振ることができるチャンネルの最大数。

**Total Sessions**

空いているセッションおよび使用中のセッションを含むセッションの総数。

**Free Sessions**

空いているセッションの数。

**Sessions LWM**

空いているセッションの最低数。

**Partition**

データベース・パーティション・サーバーの数。

**Bufs Sent**

**db2pd** コマンドが実行中のデータベース・パーティション・サーバーから、出力に示されているデータベース・パーティション・サーバーに送られる FCM バッファの総数。

**Bufs Recv**

**db2pd** コマンドが実行中のデータベース・パーティション・サーバーが、出力に示されているデータベース・パーティション・サーバーから受け取る FCM バッファの総数。

**Status db2pd** コマンドが実行されているデータベース・パーティション・サーバーと、出力にリストされている他のデータベース・パーティション・サーバーの間の論理接続状況。可能な値は次のとおりです。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

- **Inactive:** データベース・パーティション・サーバーは db2nodes.cfg ファイルで定義されていますが、現在活動状態ではありません (例えば、ユーザーがパーティションを停止した)。
- **Active:** データベース・パーティション・サーバーはアクティブです。
- **Undefined:** データベース・パーティション・サーバーは db2nodes.cfg ファイルで定義されていません。これはエラーを示す場合があります。
- **Unknown:** データベース・パーティション・サーバーは不明な状態です。これはエラーを示します。

### バッファの現在使用量

#### AppHandl

アプリケーション・ハンドル (ノードと索引を含む)。

#### TimeStamp

アプリケーション・ハンドルの使用法の固有 ID。

#### Buffers In-use

現在アプリケーションに使用されているバッファの数。

### チャンネルの現在使用量

#### AppHandl

アプリケーション・ハンドル (ノードと索引を含む)。

#### TimeStamp

アプリケーション・ハンドルの使用法の固有 ID。

#### Channels In-use

現在アプリケーションに使用されているチャンネルの数。

### バッファ使用量 HWM

#### AppHandl

アプリケーション・ハンドル (ノードと索引を含む)。

#### TimeStamp

アプリケーション・ハンドルの使用法の固有 ID。

#### Buffers Used

インスタンス開始以来の、アプリケーションによって使用されたバッファ数の最高水準点。

### チャンネル使用量 HWM

#### AppHandl

アプリケーション・ハンドル (ノードと索引を含む)。

#### TimeStamp

アプリケーション・ハンドルの使用法の固有 ID。

#### Channels Used

インスタンス開始以来の、アプリケーションによって使用されたチャンネル数の最高水準点。

## -fmp パラメーター

-fmp パラメーターについては、以下の情報が戻されます。

- Pool Size: FMP プール内の現在の FMP プロセスの数。
- Max Pool Size: FMP プール内の FMP プロセスの最大数。
- Keep FMP: **keepfenced** データベース・マネージャー構成パラメーターの値。
- Initialized: FMP が初期設定されました。指定可能な値は Yes および No です。
- Trusted Path: トラストッド・プロシージャのパス
- Fenced User: fenced ユーザー ID

### FMP プロセス:

- FmpPid: FMP プロセスのプロセス ID
- Bit: ビット・モード。値は 32 ビットまたは 64 ビットです。
- Flags: FMP プロセスの状態フラグ。可能な値は以下のとおりです。
  - 0x00000000 - JVM 初期化済み
  - 0x00000002 - スレッド化されている
  - 0x00000004 - フェデレーテッド・ラッパーの実行に使用済み
  - 0x00000008 - ヘルス・モニターに使用済み
  - 0x00000010 - シャットダウン用にマークされており、新規のタスクを受け入れない
  - 0x00000020 - **db2sysc** によるクリーンアップ用にマークされている
  - 0x00000040 - エージェント・クリーンアップ用にマークされている
  - 0x00000100 - プロセスのすべての IPCS が除去されている
  - 0x00000200 - .NET ランタイムが初期化済み
  - 0x00000400 - JVM がデバッグ用に初期化されている
  - 0x00000800 - 終了フラグ
- ActiveTh: FMP プロセス内で実行しているアクティブ・スレッドの数。
- PooledTh: FMP プロセスに保持されているプールされたスレッドの数。
- Active: FMP プロセスのアクティブ状態。値は Yes または No です。

### アクティブ・スレッド:

- FmpPid: アクティブ・スレッドを所有する FMP プロセス ID。
- EduPid: このスレッドが作業している EDU プロセス ID。
- ThreadId: アクティブ・スレッド ID。

### プールされたスレッド:

- FmpPid: プールされたスレッドを所有する FMP プロセス ID。
- ThreadId: プールされたスレッド ID。

## -fmpexechistory | -fmpe パラメーター

-fmpexechistory | -fmpe パラメーターについては、以下の情報が戻されます。

### FMP プロセス:



## db2pd - DB2 データベースのモニターおよびトラブルシューティング

- FmpPid - FMP プロセスのプロセス ID
- Bit - ビット・モード。値は 32 ビットまたは 64 ビットです。
- Flags - FMP プロセスの状態フラグ。可能な値は以下のとおりです。
  - 0x00000000 - JVM 初期化済み
  - 0x00000002 - スレッド化されている
  - 0x00000004 - フェデレーテッド・ラッパーの実行に使用済み
  - 0x00000008 - ヘルス・モニターに使用済み
  - 0x00000010 - シャットダウン用にマークされており、新規のタスクを受け入れない
  - 0x00000020 - **db2sysc** によるクリーンアップ用にマークされている
  - 0x00000040 - エージェント・クリーンアップ用にマークされている
  - 0x00000100 - プロセスのすべての IPCS が除去されている
  - 0x00000200 - .NET ランタイムが初期化済み
  - 0x00000400 - JVM がデバッグ用に初期化されている
  - 0x00000800 - 終了フラグ
- ActiveThrd - FMP プロセス内で実行しているアクティブ・スレッドの数。
- PooledThrd - FMP プロセスに保持されているプールされたスレッドの数。
- ForcedThrd - FMP プロセスによって生成された強制スレッドの数。
- Active - FMP プロセスのアクティブ状態。値は Yes または No です。

### アクティブ・スレッド:

- EduPid - このスレッドが作業している EDU プロセス ID。
- ThreadId - アクティブ・スレッド ID。
- RoutineID - ルーチン ID。
- Timestamp - アプリケーション・ハンドルの使用を表す固有 ID。

### プールされたスレッド:

- ThreadId - プールされたスレッド ID。
- RoutineID - ルーチン ID。
- Timestamp - アプリケーション・ハンドルの使用を表す固有 ID。

### 強制されたスレッド:

- ThreadId - 強制されたスレッド ID。
- RoutineID - ルーチン ID。
- Timestamp - アプリケーション・ハンドルの使用を表す固有 ID。

**db2pd -catalogcache** コマンドの出力例を参照してください。

## **-hadr パラメーター**

**-hadr** パラメーターで戻されるのは、高可用性災害時リカバリーに関連した情報です。現行設定に関連する情報のみが表示されるため、例えば、スタンバイ・データベースの読み取りが使用可能でない場合には再生専用時間枠に関する情報は表示されません。

### **HADR\_ROLE**

ローカル・データベースの現行の HADR 役割。可能な値は以下のとおりです。

- PRIMARY
- STANDBY

### **REPLAY\_TYPE**

データベースの HADR レプリケーションのタイプ。可能な値は次のとおりです。

- PHYSICAL

### **HADR\_SYNCMODE**

ローカル・データベースの現行の HADR 同期モード。可能な値は以下のとおりです。

- ASYNC
- NEARSYNC
- SUPERASYNC
- SYNC

注: 複数スタンバイ・モードの場合には、スタンバイの HADR\_SYNCMODE 値は、1 次データベースがスタンバイ・データベースに接続するまでは、空ストリング (長さがゼロのストリング) として表示されます。

### **STANDBY\_ID**

現行のセットアップに含まれているすべてのスタンバイの ID。この値は、コマンドがプライマリー上で実行された場合のみ意味があります。スタンバイ上で実行すると、必ず 0 が返されます。複数スタンバイ・モードであっても、互いに他のスタンバイは認識できないためです。単一スタンバイ・モードではスタンバイに常に 1 の ID が割り当てられますが、複数スタンバイ・モードでは 1 はプリンシパル・スタンバイを示します。

### **LOG\_STREAM\_ID**

1 次データベースからの転送対象のログ・ストリームの ID。

### **HADR\_STATE**

データベースの現在の HADR 状態。可能な値は以下のとおりです。

- DISCONNECTED
- DISCONNECTED\_PEER
- LOCAL\_CATCHUP
- PEER
- REMOTE\_CATCHUP
- REMOTE\_CATCHUP\_PENDING

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

### PRIMARY\_MEMBER\_HOST

ログ・ストリームを処理する基本メンバーのローカル・ホスト (**hadr\_local\_host** 構成パラメーターで示される)。

### PRIMARY\_INSTANCE

ログ・ストリームを処理する基本メンバーのインスタンス名。

### PRIMARY\_MEMBER

ログ・ストリームを処理する基本メンバー。

### STANDBY\_MEMBER\_HOST

ログ・ストリームを処理するスタンバイ・メンバーのローカル・ホスト (**hadr\_local\_host** 構成パラメーターで示される)。

### STANDBY\_INSTANCE

ログ・ストリームを処理するスタンバイ・メンバーのインスタンス名。

### STANDBY\_MEMBER

ログ・ストリームを処理するスタンバイ・メンバー。

### HADR\_CONNECT\_STATUS

データベースの現在の HADR 接続状況。可能な値は以下のとおりです。

- CONGESTED
- CONNECTED
- DISCONNECTED

### HADR\_CONNECT\_STATUS\_TIME

現行の HADR 接続状況が始まった時刻。HADR\_CONNECT\_STATUS 値に応じて、HADR\_CONNECT\_STATUS\_TIME 値は以下を示します。

- 輻輳開始時刻
- 接続開始時刻
- 切断時刻

### HEARTBEAT\_INTERVAL

ハートビート間隔 (秒)。これは、構成パラメーター **hadr\_timeout** や **hadr\_peer\_window** などのさまざまな因子に基づいて計算されます。HEARTBEAT\_INTERVAL エレメントは、1 次データベースとスタンバイ・データベースがモニター情報を交換する頻度を示します。

### HADR\_TIMEOUT

HADR データベースがパートナーのデータベースからメッセージを受信できる時間の長さ (秒)。この期間が過ぎると、HADR データベース・サーバーは、データベース間の接続に障害が発生して切断されたと見なします。

### TIME\_SINCE\_LAST\_RECV

最後にメッセージを受信してから経過した時間 (秒)。この数値が大きいほど、メッセージ送達の遅延が大きいということになります。

TIME\_SINCE\_LAST\_RECV 値が HADR\_TIMEOUT 値に等しいと、データベース間の接続はクローズされます。

### PEER\_WAIT\_LIMIT

ロギングがブロックされて、スタンバイ・データベースに HADR ログが転送されるのを待機している場合、1 次データベースのピア状態が解除されるまで待機する秒単位の時間。値 0 はタイムアウトがないことを示します。

### LOG\_HADR\_WAIT\_CUR

ロガーが HADR ログ転送要求に関して待機した時間の長さ (秒)。ロガーが待機していない場合は、値 0 が戻されます。待機時間が PEER\_WAIT\_LIMIT フィールドに返される値に達すると、HADR のピア状態は解除され、1 次データベースは非ブロック化されます。

### LOG\_HADR\_WAIT\_RECENT\_AVG

各ログがフラッシュされる平均時間 (秒単位)。

### LOG\_HADR\_WAIT\_ACCUMULATED

HADR のログ転送をロガーが待機した時間の累積 (秒)。

### LOG\_HADR\_WAITS\_COUNT

ロガーの HADR 待機イベントの総数。この数は、ロガーが HADR ログ転送に関して待機を開始する度に増分されます。これには、待機状態から即時に復帰した場合も含まれます。このため、実質上このカウントは、データベースがピア状態にある間のログ・フラッシュの回数です。

### SOCK\_SEND\_BUF\_REQUESTED,ACTUAL

- 要求されたソケット送信バッファ・サイズ (SOCK\_SEND\_BUF\_REQUESTED) のバイト数。値 0 は、要求がない (システム・デフォルトを使用する) ことを示します。
- 実際のソケット送信バッファ・サイズ (SOCK\_SEND\_BUF\_ACTUAL) のバイト数。

### SOCK\_RECV\_BUF\_REQUESTED,ACTUAL

- 要求されたソケット受信バッファ・サイズ (SOCK\_RECV\_BUF\_REQUESTED) のバイト数。値 0 は、要求がない (システム・デフォルトを使用する) ことを示します。
- 実際のソケット受信バッファ・サイズ (SOCK\_RECV\_BUF\_ACTUAL) のバイト数。

### PRIMARY\_LOG\_FILE, PAGE, POS

- 1 次データベース上のログ・ストリームの現行ログ・ファイルの名前 (PRIMARY\_LOG\_FILE)。
- 1 次 HADR データベース上の現在のログ位置を示す、現行ログ・ファイル内のページ番号。このページ番号はログ・ファイル内の位置に対する相対値です。例えば、ページ 0 は、ファイルの先頭です (PRIMARY\_LOG\_PAGE)。
- 1 次データベース上のログ・ストリームの現在の受信ログ位置 (バイト・オフセット) (PRIMARY\_LOG\_POS)。

### STANDBY\_LOG\_FILE, PAGE, POS

- ログ・ストリームのスタンバイ受信ログ位置に対応するログ・ファイル名 (STANDBY\_LOG\_FILE)。
- スタンバイ受信ログ位置に対応するページ番号 (ログ・ファイル内での位置に対する相対値) (STANDBY\_LOG\_PAGE)。
- スタンバイ HADR データベースの現在のログ位置 (STANDBY\_LOG\_POS)。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

### HADR\_LOG\_GAP

PRIMARY\_LOG\_POS 値と STANDBY\_LOG\_POS 値の間のギャップの移動平均 (バイト単位)。

### STANDBY\_REPLAY\_LOG\_FILE,PAGE,POS

- ログ・ストリームのスタンバイ適用ログ位置に対応するログ・ファイルの名前 (STANDBY\_REPLAY\_LOG\_FILE)。
- スタンバイ適用ログ位置に対応するスタンバイ適用ログ・ファイル内のページ番号 (STANDBY\_REPLAY\_LOG\_PAGE)。このページ番号はログ・ファイル内の位置に対する相対値です。例えば、ページ 0 はファイルの先頭です。
- ログ・ストリームのスタンバイ適用ログ位置のバイト・オフセット (STANDBY\_REPLAY\_LOG\_POS)。

### STANDBY\_RECV\_REPLAY\_GAP

スタンバイ・ログ受信位置とスタンバイ・ログ適用位置との間のギャップの平均 (バイト数)。このギャップの値がスタンバイの受信バッファー・サイズとスタンバイのスプール制限の合計値に達すると、スタンバイはログの受信を停止し、プライマリーをブロックします (プライマリーがピア状態の場合)。

### PRIMARY\_LOG\_TIME

1 次データベース上のログ・ストリームの中で最新のトランザクションのタイム・スタンプ。

### STANDBY\_LOG\_TIME

スタンバイ・データベース上で受信した、ログ・ストリームのログの中で最新のトランザクションのタイム・スタンプ。

### STANDBY\_REPLAY\_LOG\_TIME

スタンバイ・データベースで適用中のログのトランザクション・タイム・スタンプ。

### STANDBY\_RECV\_BUF\_SIZE

スタンバイの受信バッファー・サイズ (ページ数)。

### STANDBY\_RECV\_BUF\_PERCENT

現在使用されているスタンバイ・ログ受信バッファーのパーセンテージ。この値が 100 で、受信バッファーが満杯であることを示していても、ログ・スプーリングを有効にしていれば、スタンバイでログの受信を継続することができます。

### STANDBY\_SPOOL\_LIMIT

スプールするページの最大数。値 0 は、ログのスプーリングが無効であることを示し、値 -1 は、制限がないことを示します。

### PEER\_WINDOW

**hadr\_peer\_window** データベース構成パラメーターの値。

### READS\_ON\_STANDBY\_ENABLED

HADR スタンバイ・データベースの読み取りフィーチャーが有効であるかどうかの標識。可能な値は以下のとおりです。

- Y

- N

### STANDBY\_REPLAY\_ONLY\_WINDOW\_ACTIVE

スタンバイ・データベースで (DDL または保守操作適用に起因する) 再生専用時間枠が進行中 (つまり、スタンバイ・データベースで読み取りプログラムは使用できない) かどうかを示す標識。可能な値は以下のとおりです。

- Y
- N

### PEER\_WINDOW\_END

1 次データベースがアクティブである限り、この時点まで 1 次データベースがピア状態または切断ピア状態にとどまるという時点。このフィールドは、ピア・ウィンドウが有効になっている場合にのみ表示されます。

### STANDBY\_REPLAY\_DELAY

**hadr\_replay\_delay** データベース構成パラメーターの値を示します。

### TAKEOVER\_APP\_REMAINING\_PRIMARY

非強制テークオーバー中にプライマリーからまだ強制切断されていない現在のアプリケーション数。このフィールドは、非強制テークオーバーが進行中である場合にのみ表示されます。

### TAKEOVER\_APP\_REMAINING\_STANDBY

テークオーバー中に読み取り可能スタンバイからまだ強制切断されていない現在のアプリケーション数。このフィールドは、テークオーバーが進行中である場合にのみ表示されます。

### STANDBY\_REPLAY\_ONLY\_WINDOW\_START

現在の適用専用ウィンドウがアクティブになった時刻。このフィールドは、読み取り可能スタンバイにアクティブな適用専用ウィンドウがある場合にのみ表示されます。

### STANDBY\_REPLAY\_ONLY\_WINDOW\_TRAN\_COUNT

現行の適用専用ウィンドウで現在までに実行された、既存の非コミット DDL またはメンテナンス・トランザクションの合計数。このフィールドは、読み取り可能スタンバイにアクティブな適用専用ウィンドウがある場合にのみ表示されます。

## -latches パラメーター

**-latches** パラメーターについては、以下の情報が戻されます。

### Address

仮想アドレス・スペースにおける保留ラッチのアドレス。

**Holder** ラッチを保留している EDU の EDU ID。

**Waiter** ラッチを待機している EDU の EDU ID。

### Filename

ラッチが獲得されたソース・ファイル名。

**LOC** ラッチが獲得されたファイル名によって示される、ファイル内のコードの行。

### LatchType

保留されているラッチの ID。

### -load パラメーター

**-load** パラメーターについては、以下の情報が返されます。

**LoadID** 特定のロード操作の ID。

**EDU ID** エンジン・ディスパッチ可能単位 (EDU) のユニーク ID。Linux オペレーティング・システムの場合を除き、EDU ID はスレッド ID にマップされます。Linux オペレーティング・システムでは、EDU ID は DB2 生成によるユニーク ID です。

**EDU Name**

EDU の DB2 固有の名前。

**TableName**

表の名前。

**SchemaName**

表の名前を修飾するスキーマ。

**AppHandl**

アプリケーション・ハンドル (ノードと索引を含む)。

**Application ID**

アプリケーション ID。この値は **app1\_id** モニター・エレメント・データと同じです。

**StartTime**

ロード操作が最初に呼び出された日時。

**LoadPhase**

ロード操作の現在のフェーズ。

### -locks パラメーター

**-locks** パラメーターについては、以下の情報が返されます。

**TranHdl**

ロックを要求しているトランザクション・ハンドル。

**Lockname**

ロックの名前。

**Type** ロック・タイプ。可能な値は次のとおりです。

- 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** ロック・モード。可能な値は次のとおりです。

- IS
- IX
- S
- SIX
- X
- IN
- Z
- U
- NS
- NW

**Sts** ロック状況。可能な値は次のとおりです。

- G (譲渡)
- C (変換)
- W (待機)

**Owner** ロックを所有するトランザクション・ハンドル。

**Dur** ロックの期間。

### **HoldCount**

ロックに付された保留の数。保留のあるロックは、トランザクションがコミットされるときに解放されません。

**Att** ロックの属性。可能な値は以下のとおりです。

- 0x01: 使用できるようになるまで待機。
- 0x02: エスカレーションによる取得。
- 0x04: ブロック「内の」RR ロック。
- 0x08: 挿入ロック。
- 0x10: RR スキャンによるロック。
- 0x20: 行の更新/削除のロック。
- 0x40: 新規ロック要求の許可。
- 0x80: 新規ロックのリクエスト。

### **ReleaseFlg**

ロック解放フラグ。可能な値は以下のとおりです。

- 0x80000000: SQL コンパイラーによるロック。
- 0x40000000: 非ユニークな、トラックされないロック。



## db2pd - DB2 データベースのモニターおよびトラブルシューティング

**rrIID** RR ロック (上記の 0x10 属性) が取得された索引の IID。可能な値は以下のとおりです。

- 0: 単一の特定の索引に関連していない (あるいは RR ロックではない)。
- <>0: ロックを獲得するために使用された、特定の索引 IID。

### -logs パラメーター

-logs パラメーターについては、以下の情報が戻されます。

#### Current Log Number

現在アクティブなログの数。

#### Pages Written

現在のログに書き込まれている現行ページ。

#### Cur Commit Disk Log Reads

(ログ・バッファーではなく) ディスクからのログ読み取りにより、現在コミット済みバージョンの行が検索された回数。

#### Cur Commit Total Log Reads

現在コミット済みバージョンの行が、ログ (ログ・バッファーおよびディスク) から検索された合計回数。

#### Method 1 Archive Status

最新のログ・アーカイブの試行結果。可能な値は Success または Failure です。

#### Method 1 Next Log to Archive

次にアーカイブされるログ・ファイル。

#### Method 1 First Failed

アーカイブが失敗した最初のログ・ファイル。

#### Method 2 Archive Status

最新のログ・アーカイブの試行結果。可能な値は Success または Failure です。

#### Method 2 Next Log to Archive

次にアーカイブされるログ・ファイル。

#### Method 2 First Failed

アーカイブが失敗した最初のログ・ファイル。

#### StartLSN

開始ログ・シーケンス番号。

#### StartLSO

ログ・ファイルの最初の LSO。

**State** 0x00000020 はログがアーカイブされていることを示します。

**Size** ログのエクステントのサイズ (ページ数)。

**Pages** ログ内のページ数。

#### Filename

ログのファイル名。

**Log Chain ID**

ログ・チェーン番号の ID。

**Current LSN**

現行のログ・シーケンス番号 (LSN)。

**Current LSO**

現在の LSO。

**db2pd -logs** コマンドの出力例を参照してください。

### **-memblocks パラメーター**

**-memblocks** パラメーターでは、メモリー・セットの個別ブロック、メモリー・プールごとにグループ化されてソートされた合計、およびメモリー・セットのソートされた合計の、3 つの出力のセクションがあります。

メモリー・ブロック:

**PoolID** メモリー・ブロックを所有するメモリー・プール ID。

**PoolName**

メモリー・ブロックを所有するメモリー・プール名。

**BlockAge**

メモリー・ブロックのブロック経過時間。これはブロックが割り振られるときに割り当てられる増分カウンターです。

**Size** メモリー・ブロックのサイズ (バイト単位)。

**I** 割り振りのタイプ。値の 1 はブロックが個別に解放されることを示し、値の 0 はプールと共に解放されることを示します。

**LOC** メモリー・ブロックを割り振ったコードの行。

**File** ブロックが割り振られたファイル名のハッシュ値。

メモリー・プールごとに報告されてソートされた合計:

**PoolID** メモリー・ブロックを所有するメモリー・プール ID。

**PoolName**

メモリー・ブロックを所有するメモリー・プール名。

**TotalSize**

同じコード行およびファイルから割り振られたブロックの合計サイズ (バイト数)。

**TotalCount**

同じコード行およびファイルから割り振られたブロックの数。

**LOC** メモリー・ブロックを割り振ったコードの行。

**File** ブロックが割り振られたファイル名のハッシュ値。

メモリー・セットごとに報告されてソートされた合計:

**PoolID** メモリー・ブロックを所有するメモリー・プール ID。

**PoolName**

メモリー・ブロックを所有するメモリー・プール名。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

### TotalSize

同じコード行およびファイルから割り振られたブロックの合計サイズ (バイト数)。

**%Bytes** 同じコード行およびファイルから割り振られたブロックのパーセンテージ・バイト。

### TotalCount

同じコード行およびファイルから割り振られたブロックの数。

**%Count** 同じコード行およびファイルから割り振られたブロックのパーセンテージ・カウント。

**LOC** メモリー・ブロックを割り振ったコードの行。

**File** ブロックが割り振られたファイル名のハッシュ値。

## -mempools パラメーター

**-mempools** パラメーターについては、以下の情報が戻されます。(サイズはバイトで指定)

**MemSet** メモリー・プールを所有するメモリー・セット。

### PoolName

メモリー・プールの名前。

**Id** メモリー・プール ID。

### SecondId

同じタイプの複数のメモリー・プールを区別するための 2 番目のメモリー・プール ID。

### Overhead

プール構造に必要な内部使用量情報。

**LogSz** プール・メモリー要求の現在の合計。

**LogHWM** 論理サイズの最高水準点。

**PhySz** 論理サイズに必要な物理メモリー。

**PhyHWM** 処理中の最大物理サイズ。

### CfgSize

メモリー・プールの構成サイズ。

**Bnd** メモリー・プールに固定上限があるかどうかを指定します。

**BlkCnt** メモリー・プール内に割り振られたブロックの現在の数。

### CfgParm

報告されるプールのサイズを宣言する構成パラメーター。

## -memsets パラメーター

**-memsets** パラメーターについては、以下の情報が戻されます。

**Name** メモリー・セットの名前。

### Address

メモリー・セットのアドレス。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

**Id** メモリー・セット ID。

**Size(Kb)**

キロバイト単位で設定されたメモリーのサイズ。

**Key** メモリー・セット・キー (UNIX オペレーティング・システムのみ)。

**DBP** メモリー・セットを所有するデータベース・パーティション・サーバー。

**Type** メモリー・セットのタイプ。

**Unrsv(Kb)**

特定のプールのために予約されていないメモリー。セット内の任意のプールは、必要に応じてこのメモリーを使用できます。

**Used(Kb)**

現在メモリー・プールに割り振られているメモリー。

**HWM(Kb)**

メモリー・プールに割り振られたことのある最大メモリー。

**Cmt(Kb)**

DB2 データベースによってコミットされて、物理 RAM またはページング・スペース、あるいはその両方に場所を占めるすべてのメモリー。

**Uncmt(Kb)**

現在使用されていない、そして DB2 データベースによって非コミットとしてマークされているメモリー。オペレーティング・システムに応じて、このメモリーは物理 RAM またはページング・スペース、あるいはその両方に場所を占めることがあります。

**CmtRt(Kb)**

使用可能なコミット済みメモリーの最大連続区域。

**DcmtRt(Kb)**

使用可能な非コミット・メモリーの最大連続区域。

**HoldRt(Kb)**

揮発性要求に使用可能なコミット済みメモリーの最大連続区域。

**Sngl** 高速割り振りに使用可能な事前割り振り領域の数。

### -osinfo パラメーター

-osinfo パラメーターについては、以下の情報が戻されます。

**CPU 情報:** (Windows、AIX、HP-UX、Solaris および Linux オペレーティング・システム)

**TotalCPU**

CPU の総数。

**OnlineCPU**

オンラインの CPU 数。

**ConfigCPU**

構成済みの CPU 数。

**Speed(MHz)**

CPU の速度 (MHz)。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

### HMTDegree

ハードウェア・マルチスレッド化をサポートするシステムは、オペレーティング・システムに存在すると想定されるプロセッサの数を示す値を戻します。ハードウェア・マルチスレッド化をサポートしないシステムでは、この値は常に 1 です。ハードウェア・マルチスレッド化をサポートするシステムでは、合計は論理 CPU の数となります。物理 CPU の数を取得するには、合計を THREADING DEGREE で除算します。

### Timebase

時間基準のレジスタ増分の周波数 (Hz)。これは、Linux PPC でのみサポートされます。

### Cores/Socket

ソケットごとのコア数。

メガバイト単位の物理メモリーおよびスワップ: (Windows、 AIX、 HP-UX、 Solaris および Linux オペレーティング・システム)

### TotalMemTotal

メモリーのサイズ (MB)。

### FreeMem

空きメモリーの容量 (MB)。

### AvailMem

製品が使用できるメモリー容量 (MB)。

### TotalSwap

スワップ・スペースの合計 (MB)。

### FreeSwap

空きスワップ・スペースの合計 (MB)。

メガバイト単位の仮想メモリー (Windows、 AIX、 HP-UX、 および Solaris オペレーティング・システム)

**Total** システム上の仮想メモリーの総量 (MB)。

### Reserved

予約済みの仮想メモリーの量 (MB)。

### Available

使用できる仮想メモリーの量 (MB)。

**Free** 空き仮想メモリーの量 (MB)。

オペレーティング・システム情報 (Windows、 AIX、 HP-UX、 Solaris および Linux オペレーティング・システム)

**OSName** オペレーティング・システム・ソフトウェアの名前。

### NodeName

システムの名前。

### Version

オペレーティング・システムのバージョン。

### Machine

マシン・ハードウェア ID。

メッセージ・キュー情報 (AIX、HP-UX、および Linux オペレーティング・システム)

- MsgSeg** システム全体での SysV メッセージ・セグメントの合計。
- MsgMax** システム全体でのメッセージの最大サイズ。
- MsgMap** システム全体でのメッセージ・マップ内の項目の数。
- MsgMni** システム全体でのシステム用メッセージ・キュー ID の数。
- MsgTq1** システム全体でのメッセージ・ヘッダーの数。
- MsgMnb** メッセージ・キューの最大バイト数。
- MsgSsz** メッセージ・セグメント・サイズ。

共有メモリー情報 (AIX、HP-UX、および Linux オペレーティング・システム)

- ShmMax** システム全体での共有メモリー・セグメントの最大サイズ (バイト単位)。
- ShmMin** システム全体での共有メモリー・セグメントの最小サイズ (バイト数)。
- ShmIds** システム全体での共有メモリー ID の数。
- ShmSeg** プロセス全体でのプロセスごとの共有メモリー・セグメントの最大数。

セマフォ情報: (AIX、HP-UX、および Linux オペレーティング・システム)

- SemMap** システム全体でのセマフォ・マップ内の項目の数。
- SemMni** システム全体でのセマフォ ID の最大数。
- SemMns** システム全体でのシステム上のセマフォの最大数。
- SemMnu** システム全体でのシステムの取り消し構造の最大数。
- SemMs1** システム全体での ID ごとのセマフォの最大数。
- SemOpm** システム全体での semop 呼び出しごとの操作の最大数。
- SemUme** システム全体でのプロセスごとの取り消し構造の最大数。
- SemUsz** システム全体での取り消し構造のサイズ。semume から派生されません。
- SemVmx** システム全体でのセマフォの最大値。
- SemAem** システム全体での終了値の最大調整。

CPU ロード情報 (Windows、AIX、HP-UX、Solaris、および Linux オペレーティング・システム)

- shortPeriod**  
直前の 1 分間における、実行可能プロセスの数。
- mediumPeriod**  
直前の 5 分間における、実行可能プロセスの数。
- longPeriod**  
直前の 15 分間における、実行可能プロセスの数。

ディスク情報

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

### **BkSz(bytes)**

ファイル・システム・ブロック・サイズ (バイト単位)。

### **Total(bytes)**

デバイス上の合計バイト数 (バイト単位)。

### **Free(bytes)**

デバイス上の空きバイト数 (バイト単位)。

**Inodes** i ノードの総数。

**FSID** ファイル・システム ID。

### **DeviceType**

装置タイプ。

**FSName** ファイル・システム名。

### **MountPoint**

ファイル・システムのマウント・ポイント。

## **-pages パラメーター**

**-pages** パラメーターについては、以下の情報がページごとに戻されます。

**BPID** ページを含むバッファ・プール ID。

### **TbpaceID**

ページを含む表スペース ID。

### **TbpacePgNum**

表スペース内の論理ページ番号 (DMS のみ)。

**ObjID** ページを含むオブジェクト ID。

### **ObjPgNum**

オブジェクト内の論理ページ番号。

### **ObjClass**

ページに含まれるオブジェクトのクラス。可能な値は、Perm、Temp、Reorg、Shadow、および EMP です。

### **ObjType**

ページに含まれるオブジェクトのタイプ。値は、Data、Index、LongField、XMLData、SMP、LOB、LOBA、または BlockMap のいずれかです。

**Dirty** ページがダーティーであるかどうかを示します。可能な値は Y および N です。ページ出力のサマリー情報セクションで、この値はダーティー・ページの数を示します。

### **Permanent**

ページ出力のサマリー情報セクションで、この値は PERMANENT ページの数を示します。

### **Temporary**

ページ出力のサマリー情報セクションで、この値は TEMPORARY ページの数を示します。

### **Prefetched**

ページがプリフェッチされているかどうかを示します。可能な値は Y および N です。

**db2pd -pages** コマンドの出力例を参照してください。

### **-recovery パラメーター**

**-recovery** パラメーターについては、以下の情報が戻されます。

#### **Database State**

パーティション・データベース環境内のカタログ・パーティションに障害が発生した場合における、そのデータベース・カタログ・パーティションの状態。データベース・カタログ・パーティションに障害が発生した場合、CATALOGNODEFAIL 状態が戻されます。それ以外の場合、情報は戻されません。この状態は、任意のデータベース・パーティションから表示できます。

#### **Recovery Status**

内部リカバリー状況。

#### **Current Log**

リカバリー操作によって使用される現行ログ。

#### **Current LSN**

現行ログのシーケンス番号。

#### **Current LRI**

現在の LRI。

#### **Current LSO**

現在の LSO。

#### **Job Type**

実行されるリカバリーのタイプ。可能な値は次のとおりです。

- 5: クラッシュ・リカバリー。
- 6: データベースまたは表スペースでのロールフォワード・リカバリー。

#### **Job ID** ジョブ ID。

#### **Job Start Time**

リカバリー操作が開始される時刻。

#### **Job Description**

リカバリー・アクティビティーの説明。可能な値は次のとおりです。

- Tablespace Rollforward Recovery
- Database Rollforward Recovery
- Crash Recovery

#### **Invoker Type**

リカバリー操作の呼び出し方法。可能な値は次のとおりです。

- User
- DB2

#### **Total Phases**

リカバリー操作を完了するために必要なフェーズの数。



## db2pd - DB2 データベースのモニターおよびトラブルシューティング

### Current phase

リカバリー操作の現在のフェーズ。

**Phase** リカバリー操作での現在のフェーズ番号。

### Forward phase

ロールフォワード・リカバリーの最初のフェーズ。このフェーズは、REDO フェーズとも呼ばれます。

### Backward phase

ロールフォワード・リカバリーの 2 番目のフェーズ。このフェーズは、UNDO フェーズとも呼ばれます。

**Metric** 作業単位。可能な値は次のとおりです。

- 1: バイト。
- 2: エクステント。
- 3: 行。
- 4: ページ。
- 5: 索引。

### TotWkUnits

リカバリー操作のこのフェーズに実行される作業単位 (UOW) の総数。

### TotCompUnits

完了した UOW の総数。

## -reopt パラメーター

**-reopt** パラメーターについては、以下の情報が戻されます。

### Dynamic SQL Statements

**-dynamic** を参照。

### Dynamic SQL Environments

**-dynamic** を参照。

### Dynamic SQL Variations

**-dynamic** を参照。

### Reopt Values

指定の SQL ステートメントを再最適化するために使用された変数に関する情報を表示します。使用されなかった変数に関する情報は戻されません。有効な値は以下のとおりです。

**AnchID** ハッシュ・アンカー ID。

**StmtID** このバリエーションのステートメント ID。

**EnvID** このバリエーションの環境 ID。

**VarID** バリエーション ID。

### OrderNum

SQL ステートメントの再最適化に使用された変数の序数。

### SQLZType

変数のタイプ。

**CodPg** 変数のコード・ページ。  
**NullID** 値がヌル終了かどうかを示すフラグ。  
**Len** 可変値の長さ (バイト単位)。  
**Data** 変数に使用される値。

### -reorgs パラメーター

**-reorgs** パラメーターについては、以下の情報が戻されます。

#### 索引 Reorg 統計:

##### Retrieval time

このセットの索引 REORG 統計情報の取得時間。

##### TabSpaceID

表スペース ID。

##### TableID

表 ID。

**Schema** 表のスキーマ。

##### TableName

表の名前。

##### MaxPartition

処理中の表のパーティションの総数。パーティション・レベルの再編成の場合、再編成されているのは単一のパーティションのみであるため、MaxPartition の値は常に 1 になります。このフィールドは、パーティション索引の場合のみ表示されます。

##### PartitionID

処理中のパーティションのデータ・パーティション ID。このフィールドは、パーティション索引の場合のみ表示されます。

**Access** アクセス・レベルの可能な値は以下のとおりです。

- Allow none
- Allow read
- Allow write

**Status** 再編成の現在の状況。次のいずれかです。

- In Progress (操作が進行中)
- Completed (操作が正常に完了した)
- Stopped (エラーまたは割り込みのために操作が停止している)

##### Start time

この再編成セッションの開始時刻。

##### End time

この再編成セッションの終了時刻。

##### Total duration

この再編成セッションの合計所要時間。

##### Prev Index Duration

直前の (完了した) 索引の再編成所要時間。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

### Cur Index Start

現在の (進行中の) 索引の再編成開始時刻。

### Cur Index

現在の (進行中の) 索引のシーケンス番号。

### Max Index

モニターされている索引の総数。システムが生成した索引の中にはモニターされないものもあるため、表の索引の総数と同じではありません。

### Index ID

現在の (進行中の) 索引の索引 ID。

### Cur Phase

現在のフェーズのシーケンス番号。中括弧内は現在のフェーズの名前であり、以下のいずれかです。

- Scan (表がスキャンされており、1 データ・ページずつソートされている)
- Build (ソート済み入力から索引が 1 行ずつ作成されている)
- Catchup (索引の作成中に発生したトランザクションが再生されている。アクセス権限レベルが ALLOW WRITE の索引再編成の場合のみ示される)

### Max Phase

現在の (進行中の) 索引のフェーズの総数。索引のタイプに応じて異なります。

### CurCount

これまでに処理された作業単位数。単位の意味は、以下のように再編成フェーズごとに異なります。

- Scan フェーズ: スキャンされたデータ・ページ数
- Build フェーズ: 処理された行数
- Catchup: 再生されたトランザクション・ログ・レコード数

### MaxCount

現在のフェーズの単位の総数 (単位の説明については CurCount を参照)。

### Total Row Count

処理された行の総数。フェーズと索引タイプによって、示されたり示されなかったりします。

**db2pd -reorgs index** コマンドの出力例を参照してください。

### 表 Reorg 統計:

#### Address

16 進値。

#### TableName

表の名前。

**Start** 表の再編成が開始された時刻。

**End** 表の再編成が終了した時刻。

**PhaseStart**

表の再編成フェーズの開始時刻。

**MaxPhase**

再編成の間に発生する再編成フェーズの最大数。この値はオフラインの表再編成にのみ適用されます。

**Phase** 表再編成のフェーズ。この値はオフラインの表再編成にのみ適用されます。可能な値は次のとおりです。

- Sort
- Build
- Replace
- InxRecreat

**CurCount**

完了した表の再編成の量を示す進行単位。この値によって表される進行量は、表の再編成に必要な作業の合計量を示す **MaxCount** の値に対する相対的なものです。

**MaxCount**

表の再編成に必要な作業の全体量を示す値。この値を **CurCount** とともに使用することによって、表の再編成の進行具合を判別できます。

**Status** オンラインの表の再編成の状況。この値は、オフラインの表の再編成には適用されません。可能な値は次のとおりです。

- Started
- Paused
- Stopped
- Done
- Truncat

**Completion**

表の再編成の成否の標識。可能な値は次のとおりです。

- 0: 表の再編成は正常に完了しました。
- -1: 表の再編成は失敗しました。

**PartID** データ・パーティション ID。データ・パーティションごとに 1 行が戻され、再編成の情報を示します。

**MasterTbs**

パーティション表の場合、これはパーティション表が属する論理表スペース ID です。非パーティション表の場合、この値は TbspaceID に対応します。

**MasterTab**

パーティション表の場合、これはパーティション表の論理表 ID です。非パーティション表の場合、この値は TableID に対応します。

**Type** 再編成のタイプ。可能な値は次のとおりです。

- Online
- Offline

**IndexID**

表の再編成に使用される索引の ID。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

### TempSpaceID

表が再編成される表スペース。

### -scansharing パラメーター

**-scansharing** パラメーターの場合、見出しに関しては、以下のようなフィールドが戻されます。

#### 個別の共有スキャン

- エージェント ID
- アプリケーション ID
- ScanMode (prewrap または postwrap)
- IsScanWrappable
- スキャン速度
- スロットルされていた時間
- グループ内のページのスキャンの相対的位置 (ブロック索引のスキャンの場合)。ページ内のスキャンの絶対的位置 (表スキャンと範囲スキャンの場合)
- 予測される速度カテゴリー (SLOW または FAST)
- 処理する残りのページ (表スキャンと範囲スキャンの場合に正確)。ブロック索引のスキャンの場合、オプティマイザーの見積もりが代わりに戻ります。

**db2pd -scansharing** コマンドの出力例を参照してください。

#### 共有セット

- 表スペース ID
- 表 ID
- スキャン・オブジェクト (表スキャンの場合は 0、特定のブロック索引の場合は索引 ID)
- グループの数
- ページ内の共有セットの占有スペース
- ページ内の表サイズ (非パーティション表上の表スキャンとブロック索引スキャン、およびパーティション表上の範囲スキャンの場合。パーティション表上のブロック索引スキャンの場合の値は unknown です)
- 高速スキャン速度 (FAST スキャンの実行速度)
- 低速スキャン速度 (SLOW スキャンの実行速度)

#### 共有グループ

- グループ内のスキャン数
- グループの占有スペース (ページ数)

### -serverlist パラメーター

**-serverlist** パラメーターについては、以下の情報が返されます。

**Time** サーバー・リストがキャッシュされた時刻

**Database Name**

データベースの名前

**Count** サーバー・リストの項目数

**Hostname**

メンバーの TCP/IP ホスト名

**Non-SSL Port**

メンバーがクライアント接続を listen する非 SSL ポート

**SSL Port**

メンバーがクライアント接続を listen する SSL TCP/IP ポート

**Priority**

重みとも呼ばれる、メンバーの相対負荷。メンバー (A) の値が別のメンバー (B) よりも高い場合は、メンバー A により多くの処理を向けるようにクライアントに指示します。

**-serviceclasses パラメーター**

**-serviceclasses** パラメーターの場合、見出しに関しては、以下のようなフィールドが戻されます。

**サービス・クラスのフィールド:**

- Service Class Name: サービス・クラスの名前
- Service Class ID: システムが生成するサービス・クラス ID
- Service Class Type: サービス・クラスの種類 (スーパークラスまたはサブクラス)
- Service Class State (Effective and Catalog): サービス・クラスの状態 (使用可能または使用不可)
- Effective Agent Priority および Catalog Agent Priority: SYSCAT.SERVICECLASSES に記録された優先順位にマップされるサービス・クラスの有効なエージェント優先順位設定
- Effective Prefetch Priority および Catalog Prefetch Priority: SYSCAT.SERVICECLASSES に記録された優先順位にマップされるサービス・クラスの有効なプリフェッチ優先順位設定
- Effective Bufferpool Priority および Catalog Bufferpool Priority: SYSCAT.SERVICECLASSES に記録された優先順位にマップされるサービス・クラスの有効なバッファー・プール優先順位設定
- Effective Outbound Correlator および Catalog Outbound Correlator: SYSCAT.SERVICECLASSES に記録された相関関係子にマップされるサービス・クラスの有効なアウトバウンド相関関係子設定
- Last Statistics Reset Time: サービス・クラスに関する統計が最後にリセットされたときのタイム・スタンプ

**サービス・スーパークラスのフィールド:**

- Default Subclass ID: デフォルト・サブクラスのサービス・クラス ID
- Work Action Set ID: サービス・スーパークラスに関連した作業アクション・セットの ID

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

- Collect Request Metrics: サービス・クラスの COLLECT REQUEST METRICS オプションの設定
- Num Connections: サービス・スーパークラス内のコーディネーター接続およびリモート接続の現在の数
- Num Coordinator Connections: サービス・スーパークラス内のコーディネーター接続の現在の数
- Coordinator Connections HWM: 最後に統計がリセットされた以降の、コーディネーター接続の最高水準点
- Associated Workload Occurrences (WLO): サービス・スーパークラス内の現在のワークロード・オカレンスのリスト

### サービス・サブクラスのフィールド:

- Parent Superclass ID: 親スーパークラスのサービス・クラス ID
- Collect Activity Opt: サービス・サブクラスに関する COLLECT ACTIVITY DATA オプションの設定
- Collect Aggr Activity Opt: サービス・サブクラスに関する COLLECT AGGREGATE ACTIVITY オプションの設定
- Collect Aggr Request Opt: サービス・サブクラスに関する COLLECT AGGREGATE REQUEST オプションの設定
- Act Lifetime Histogram Template ID: アクティビティ存続時間ヒストグラム・テンプレートの ID
- Act Queue Time Histogram Template ID: アクティビティ・キュー時間ヒストグラム・テンプレートの ID
- Act Execute Time Histogram Template ID: アクティビティ実行時間ヒストグラム・テンプレートの ID
- Act Estimated Cost Histogram Template ID: アクティビティ見積コスト・ヒストグラム・テンプレートの ID
- Act Interarrival Time Histogram Template ID: アクティビティ到着間隔ヒストグラム・テンプレートの ID
- Request Execute Time Histogram Template ID: 要求実行時間ヒストグラム・テンプレートの ID
- Access Count: サービス・サブクラス内の現在のアクティビティ数
- Activities HWM: このサブクラスによってシステムに入れられたアクティビティと、REMAP ACTIVITY しきい値アクションによってこのサブクラスに再マップされたアクティビティの両方に関して、最後に統計をリセットして以降のアクティビティの最高水準点。
- Activities Completed: 最後に統計がリセットされた以降に完了したアクティビティの総数。アクティビティが完了する前に REMAP ACTIVITY アクションを使用して異なるサブクラスにそのアクティビティを再マップする場合、そのアクティビティのカウント対象となるのは、それが完了したサブクラスでの合計だけです。
- Activities Rejected: 最後に統計がリセットされた以降に拒否されたアクティビティの総数
- Activities Aborted: 最後に統計がリセットされた以降にアボートされたアクティビティの総数。アクティビティがアボートされる前に

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

REMAP ACTIVITY アクションを使用して異なるサブクラスにそのアクティビティを再マップする場合、そのアクティビティのカウンタ対象となるのは、それがアボートしたサブクラスでの合計だけです。

- Associated Agents: サービス・サブクラス内で現在作業中のエージェントのリスト
- Associated Non-agent threads: サービス・サブクラス内で現在作業中の非エージェント・エンティティのリスト

**db2pd -serviceclasses** コマンドの出力例を参照してください。

### **-sort** パラメーター

**-sort** パラメーターについては、以下の情報が戻されます。

#### **AppHandl**

アプリケーション・ハンドル (ノードと索引を含む)。

**SortCB** ソート制御ブロックのアドレス

#### **MaxRowSize**

ソートされる行のすべての列の最大長の合計

#### **EstNumRows**

オブティマイザーによって見積もられた、ソートに挿入される行数

#### **EstAvgRowSize**

オブティマイザーによって見積もられた、ソートされる行の平均長さ

#### **NumSMPSorts**

そのソートを処理する並行サブエージェントの数

#### **NumSpills**

そのソートをディスクへスピルした回数の総数

#### **KeySpec**

ソートされる各列のタイプおよび長さに関する記述

#### **SortheapMem**

そのソートによって予約され、割り振られるソート・ヒープ・メモリーの KB 数

#### **NumSpilledRows**

そのソートでディスクへスピルされた行の総数

#### **NumBufferedRows**

最後にスピルされてからそのソートに挿入された行の総数

### **-static** パラメーター

**-static** パラメーターについては、以下の情報が戻されます。

静的キャッシュ:

#### **Current Memory Used**

パッケージ・キャッシュによって使用されるバイト数。

#### **Total Heap Size**

パッケージ・キャッシュに内部的に構成されるバイト数。



## db2pd - DB2 データベースのモニターおよびトラブルシューティング

### Cache Overflow flag state

パッケージ・キャッシュがオーバーフロー状態にあるかどうかを示すフラグ。

### Number of References

パッケージ・キャッシュ内のパッケージに対する参照の数。

### Number of Package Inserts

パッケージ・キャッシュへのパッケージ挿入の数。

### Number of Section Inserts

パッケージ・キャッシュへの静的セクション挿入の数。

### パッケージ:

**Schema** パッケージの修飾子。

### PkgName

パッケージの名前。

### Version

パッケージのバージョン ID。

### UniqueID

パッケージに関連した整合性トークン。

**NumSec** ロードされたセクションの数。

### UseCount

キャッシュに入れられたパッケージの使用回数。

**NumRef** キャッシュに入れられたパッケージの参照回数。

**Iso** パッケージの分離レベル。

**Q0pt** パッケージの照会最適化。

**B1k** パッケージのブロック化因数。

### Lockname

パッケージのロック名。

### セクション:

**Schema** セクションが属するパッケージの修飾子。

### PkgName

セクションが属するパッケージ名。

### UniqueID

セクションが属するパッケージに関連した整合性トークン。

**SecNo** セクション番号。

**NumRef** キャッシュに入れられたセクションが参照された回数。

### UseCount

キャッシュに入れられたセクションの使用回数。

### StmtType

キャッシュに入れられたセクションの内部ステートメント・タイプ値。

**Cursor** カーソル名 (該当する場合)。

**W-Hld** カーソルが WITH HOLD カーソルかどうかを示す。

### -statisticscache パラメーター

**-statisticscache** パラメーターに関しては、以下の情報が戻されます。

**Current Size**

統計キャッシュで使用されている現在のバイト数。

**Address**

統計キャッシュ内の項目のアドレス。

**Schema** 表のスキーマ修飾子。

**Name** 表の名前。

**LastRefID**

表を参照した最後のプロセス ID。

**LastStatsTime**

表に関して最後に統計が収集された時間。

**Sts** 項目の状況。可能な値は次のとおりです。

- V (有効)。
- I (無効)。

IBM サポートが問題の分析とトラブルシューティングを行うために役立つ追加情報が返されることもあります。

**-statisticscache** コマンド・パラメーターを使用した場合に返される情報の詳細については、「問題判別およびデータベース・パフォーマンスのチューニング」のトピック『カタログ統計の表』を参照してください。

### -storagegroups パラメーターと storagepaths パラメーター

**-storagegroups** パラメーターと **-storagepaths** パラメーターの両方とも、以下の情報を返します。

**Storage Group Configuration:**

**SGID** ストレージ・グループ ID。

**Deflt** ストレージ・グループが、現在指定されているデフォルトのストレージ・グループであるかどうかを示します。

**DataTag**

データを一意的に特定し、グループ化するのに使用される識別タグ。

**Name** ストレージ・グループの名前。

**Storage Group Statistics:**

**SGID** ストレージ・グループ ID。

**State** ストレージ・グループの状態。以下の値のいずれか。

- 0x0000000000000000 - SQLB\_STORAGEGROUP\_STATE\_NORMAL
- 0x0000000000000001 - SQLB\_STORAGEGROUP\_ALTER\_PENDING
- 0x0000000000000002 - SQLB\_STORAGEGROUP\_SKIP\_ALTERS

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

- 0x0000000000000004 - SQLB\_STORAGEGROUP\_KEEP\_ON\_DISK\_PATHS
- 0x0000000000000008 -  
SQLB\_STORAGEGROUP\_REDEFINE\_CONTAINERS
- 0x0000000000000010 - SQLB\_STORAGEGROUP\_CREATE\_PENDING
- 0x0000000000000020 - SQLB\_STORAGEGROUP\_DROP\_PENDING
- 0x0000000000000040 - SQLB\_STORAGEGROUP\_RENAME\_PENDING

### NumPaths

このストレージ・グループに定義されているストレージ・パスの数。

### NumDropPen

ドロップ保留状態にあるストレージ・パスの数。

### ストレージ・グループ・パス:

**SGID** ストレージ・グループ ID。

**PathID** ストレージ・パス ID。

### PathState

ストレージ・パスの現在の状態: NotInUse、InUse、または DropPending。

### PathName

データベースに対して定義された自動ストレージ・パスの名前。パスにデータベース・パーティション式が含まれる場合、拡張パスの後に括弧内に入れます。

**db2pd -storagegroups** コマンドおよび **db2pd -storagepaths** コマンドの出力例を参照してください。

## -sysplex パラメーター

**-sysplex** パラメーターについては、以下の情報が戻されます。

**Alias** データベース別名。

### Location Name

データベース・サーバーの固有名。

**Count** サーバーにリストで検出された項目の数。

### IP Address

サーバーの IP アドレス。

**Port** サーバーによって使用される IP ポート。

### Priority

正規化されたワークロード・マネージャー (WLM) の重み。

### Connections

このサーバーに対するアクティブな接続の数。

**Status** 接続の状況。可能な値は次のとおりです。

- 0: 良好。
- 1: 不良。サーバーはリストにありますが、接続は確立できません。現在、この項目は接続が確立されるときには考慮されません。

- 2: 不良。サーバーは以前には使用できませんでしたが、現在は接続が確立されるときに考慮されます。

**PRDID** 最後の接続のサーバーの製品 ID。

### **-tablespaces** パラメーター

**-tablespaces** パラメーターについては、出力が次の 4 つのセグメントに編成されます。

表スペース構成:

**Id** 表スペース ID。

**Type** 表スペースのタイプ。可能な値は次のとおりです。

- SMS
- DMS

**Content**

内容のタイプ。可能な値は次のとおりです。

- Regular
- Large
- SysTmp
- UsrTmp

**PageSz** 表スペースに使用されるページ・サイズ。

**ExtentSz**

エクステンツのサイズ (ページ数)。

**Auto** プリフェッチ・サイズが **AUTOMATIC** に設定されているかどうかを示します。可能な値は次のとおりです。

- Yes
- No

**Prefetch**

各範囲プリフェッチ要求の表スペースから読み取られるページ数。

**BufID** この表スペースのマップ先のバッファー・プールの ID。

**BufIDDisk**

次の始動時のこの表スペースのマップ先のバッファー・プールの ID。

**FSC** これはファイル・システム・キャッシングを表し、**CREATE TABLESPACE** または **ALTER TABLESPACE** の時点でバッファー I/O がユーザーによって指定されたかどうかを示します。可能な値は次のとおりです。

- Yes
- No

**NumCnts**

表スペースが所有するコンテナーの数。

**MaxStripe**

表スペースに現在定義されている最大ストライプ・セット (DMS 表スペースにのみ適用)。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

### LastConsecPg

最後の連続したオブジェクト表エクステンツ。

**Name** 表スペースの名前。

### 表スペース統計:

**Id** 表スペース ID。

### TotalPages

DMS 表スペースの場合、表スペースの各コンテナの総サイズの合計 (コンテナの合計ページ・フィールドで報告される)。

SMS 表スペースの場合、この値は表スペースが所有するファイル・システム内のページ数を反映します。

### UsablePgs

DMS 表スペースの場合、表スペースの各コンテナの純サイズの合計 (コンテナのうち使用できるページ・フィールドで報告される)。

SMS 表スペースの場合、この値は表スペースが所有するファイル・システム内のページ数を反映します。

### UsedPgs

DMS 表スペースの場合、表スペースで現在使用されているページの総数。

SMS 表スペースの場合、この値は表スペースが所有するファイル・システム内のページ数を反映します。

### PndFreePgs

使用できないものの、現在の未解決のトランザクションがすべてコミットすると使用できるようになるページ数。

### FreePgs

DMS 表スペースの場合、表スペース内の使用できるページ数。

SMS 表スペースの場合、この値は常に 0 です。

**HWM** 表スペース内の最高割り振りページ。

### Max HWM

インスタンスが開始されて以来の表スペースの最大 HWM。

### State

- 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

- 0x0000800 - BACKUP IN PROGRESS
- 0x0001000 - STORAGE MUST BE DEFINED
- 0x0002000 - RESTORE IN PROGRESS
- 0x0004000 - OFFLINE
- 0x0008000 - DROP PENDING
- 0x0010000 - WRITE SUSPENDED
- 0x0020000 - LOAD IN PROGRESS
- 0x0200000 - STORAGE MAY BE DEFINED
- 0x0400000 - STORAGE DEFINITION IS IN FINAL STATE
- 0x0800000 - STORAGE DEFINITION CHANGED PRIOR TO ROLLFORWARD
- 0x1000000 - DMS REBALANCER IS ACTIVE
- 0x2000000 - DELETION IN PROGRESS
- 0x4000000 - CREATION IN PROGRESS

### MinRecTime

表スペースの最小リカバリー時間。

### NQuiescers

静止者の数。

### PathsDropped

自動ストレージ表スペースの場合、ドロップされたストレージ・パス上に、1 つ以上のコンテナがあるかどうかを示します。可能な値は次のとおりです。

- Yes
- No

### TrackmodState

最後のバックアップまたは次のバックアップに関する表スペースの変更状態。可能な値は次のとおりです。

- Clean - 直前のバックアップ以降、表スペースに変更は発生していません。今回、増分バックアップまたは差分バックアップを実行するのであれば、この表スペースのデータ・ページはバックアップされません。
- Dirty - 表スペースには、次のバックアップで取り出す必要があるデータが入っています。
- InIncremental - 表スペースには、増分バックアップにコピーされた変更が入っています。この状態は、今後の増分バックアップにこのプールのページを含める必要があるフルバックアップにとっては、Dirty な状態です。また、この状態は、今後の差分バックアップにはこのプールのページを含める必要がないため、Clean な状態でもあります。
- ReadFull - 正常に完了しなかった可能性があるフルバックアップ、または現在進行中のフルバックアップによって読み取られたダーティーな表スペースが原因で、最新の表スペース変更状態が変更されました。
- ReadIncremental - 正常に完了しなかった可能性がある増分バックアップ、または現在進行中の増分バックアップによって読み取られたダーティーな表スペースが原因で、最新の表スペース変更状態が変更されました。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

- n/a - **trackmod** 構成パラメーターが No に設定されています。このため、表スペース変更状態情報は利用できません。

### 表スペース自動サイズ変更の統計:

- Id** 表スペース ID。
- AS** 表スペースが自動ストレージを使用しているかどうかを示します。可能な値は次のとおりです。
- Yes
  - No
- AR** 表スペースの自動的なサイズ変更が有効になっているかどうかを示します。可能な値は次のとおりです。
- Yes
  - No

### InitSize

自動ストレージ表スペースの場合、このパラメーターの値は表スペースの初期サイズ (バイト) です。

### IncSize

このパラメーターの値が -1 の場合には、データベース・マネージャーが自動的に適切な値を判別します。自動的にサイズ変更される表スペースの場合、IIP フィールドの値が No であれば、このパラメーターの値は、表スペースがいっぱいスペース要求が出された場合に表スペース・サイズが自動変更されるときにサイズ増加単位 (バイト、データベース・パーティションごと) です。IIP フィールドの値が Yes であれば、このパラメーターの値はパーセントです。

- IIP** 自動的にサイズ変更される表スペースの場合、このパラメーターの値は、IncSize フィールドの増分値がパーセントかどうかを示します。可能な値は次のとおりです。
- Yes
  - No

### MaxSize

自動的にサイズ変更される表スペースの場合、このパラメーターの値は、表スペースをどこまで自動的に増加させることができるかの最大サイズ (バイト、データベース・パーティションごと) を指定します。値 NONE は、最大サイズが存在しないことを示します。

### LastResize

正常に行われた最後の自動サイズ変更操作のタイム・スタンプ。

- LRF** 最後に行われた自動サイズ変更操作が失敗したかどうかを示します。可能な値は次のとおりです。
- Yes
  - No

### 表スペース・ストレージ統計:

- Id** 表スペース ID。

### DataTag

データを一意的に特定し、グループ化するのに使用される識別タグ。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

### Rebalance

再平衡化がアクティブかどうかを示します。

**SGID** 自動ストレージ管理の表スペースの場合、表スペースが関連付けられているストレージ・グループを示します。

### SourceSGID

ストレージ・グループの関連を変更している自動ストレージ管理の表スペースの場合、表スペースが関連付けられていたソース・ストレージ・グループを示します。

### 表スペース・コンテナ:

**TspId** コンテナを所有する表スペースの ID。

### ContainNum

表スペース内のコンテナに割り当てられた番号。

**Type** コンテナのタイプ。可能な値は次のとおりです。

- Path
- Disk
- File
- Striped Disk
- Striped File

### TotalPgs

コンテナ内のページ数。

### UsablePgs

コンテナ内で使用できるページの数。

### StripeSet

コンテナが置かれるストライプ・セット (DMS 表スペースにのみ適用)。

### Container

コンテナの名前。

**PathID** 自動ストレージ表スペースの場合、コンテナが存在するストレージ・パスの ID。

**db2pd -tablespaces** コマンドの出力例を参照してください。

## -tcbstats パラメーター

**-tcbstats** パラメーターについては、以下の情報が戻されます。

### TCB 表情報:

#### TbpaceID

表スペース ID。

#### TableID

表 ID。

**PartID** パーティション表の場合、これはデータ・パーティション ID です。非パーティション表の場合、これは 'n/a' を表示します。



## db2pd - DB2 データベースのモニターおよびトラブルシューティング

### MasterTbs

パーティション表の場合、これはパーティション表が属する論理表スペース ID です。非パーティション表の場合、この値は TbspaceID に対応します。

### MasterTab

パーティション表の場合、これはパーティション表の論理表 ID です。非パーティション表の場合、この値は TableID に対応します。

### TableName

表の名前。

### SchemaNm

表の名前を修飾するスキーマ。

### ObjClass

オブジェクト・クラス。可能な値は次のとおりです。

- Perm (永続)。
- Temp (一時)。

### DataSize

データ・オブジェクト内のページ数。

**LfSize** 長いフィールド・オブジェクトのページ数。

### LobSize

ラージ・オブジェクトのページ数。

### XMLSize

XML オブジェクト内のページ数。

## TCB 表統計:

### TableName

表の名前。

### SchemaNm

表の名前を修飾するスキーマ。

**Scans** 表に対して実行されたスキンの数。

**UDI** 最後に表統計が (バックグラウンド統計収集プロセスによって、または **RUNSTATS** コマンドを手動で使用して) 更新された後に、表に対して実行された更新、削除、および挿入操作の数。

**RTSUDI** リアルタイム統計収集、バックグラウンド統計収集プロセス、または手動の **RUNSTATS**によって最後に表統計が更新された後に、表に対して実行された更新、削除、および挿入操作の数。

### PgReorgs

再編成が実行されたページ数。

### NoChgUpdts

表内の列を変更しなかった更新の数。

**Reads** 表のモニターがオンになっていたときに表から読み取られた行数。

### FscrUpdates

フリー・スペース制御レコードに対する更新の数。

### Inserts

表に対して実行された挿入操作の数。

### Updates

表に対して実行された更新操作の数。

### Deletes

表に対して実行された削除操作の数。

### OvFlReads

表のモニターがオンになっていたときに表に対して読み取られたオーバーフローの数。

### OvFlCrtes

作成された新しいオーバーフローの数。

### CCLogReads

表に関して現在コミット済みバージョンの行が検索された回数。

### StoredBytes

この列は、**db2pd -temptable** 出力の「Total stored temp bytes」に相当します。

### BytesSaved

この列は、**db2pd -temptable** 出力の「Total bytes saved」値に相当します。

### PgDictsCreated

正常に作成されたページ・レベルのディクショナリーの総数。

注 以下のデータは、**-tcbstats** パラメーターに **-all** または **-index** オプションを指定する場合にのみ表示されます。

### TCB 索引情報:

#### InxTbSpace

索引が置かれている表スペース。

#### ObjectID

索引のオブジェクト ID。

**PartID** パーティション表の場合、これはデータ・パーティション ID です。非パーティション表の場合、N/A が表示されます。

#### TbSpaceID

表スペース ID。

#### TableID

表 ID。

#### MasterTbs

パーティション表の場合、これはパーティション表が属する論理表スペース ID です。非パーティション表の場合、この値は TbSpaceID に対応します。

#### MasterTab

パーティション表の場合、これはパーティション表の論理表 ID です。非パーティション表の場合、この値は TableID に対応します。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

**TableName**

表の名前。

**SchemaNm**

表の名前を修飾するスキーマ。

**IID** 索引 ID。

**IndexObjSize**

索引オブジェクトのページ数。IndexObjSize に報告される値は推奨されなくなりました。エクステントを再利用するために再編成を実行した場合、IndexObjSize 出力は、索引オブジェクトのページ数を正しく反映しません。この値には、再編成中に解放されたページも含まれるためです。正しい値を取得するためには、代わりに ADMIN\_GET\_INDEX\_INFO 表関数の INDEX\_OBJECT\_P\_SIZE 列または INDEX\_OBJECT\_L\_SIZE 列を使用してください。

**TCB 索引統計:****TableName**

表の名前。

**IID** 索引 ID。

**PartID** パーティション表の場合、これはデータ・パーティション ID です。非パーティション表の場合、N/A が表示されます。

**EmpPgDel**

削除された空のリーフ・ノードの数。

**RootSplits**

索引ツリーを深くしたキーの挿入または更新操作の数。

**BndrySplits**

最低位または最高位のキーへの挿入操作を発生させる境界リーフの分割数。

**PseuEmptPg**

疑似の空としてマークされたリーフ・ノードの数。

**EmpPgMkdUsd**

再利用された疑似空ページの数。

**Scans** 索引に対するスキャンの数。

**IxOnlyScns**

スキャン中に読み取られたページ数にかかわらず、索引に対して実行された索引のみのスキャン (索引のみにアクセスして完了するスキャン) の数。

**KeyUpdates**

キーに対する更新の数。

**InclUpdats**

組み込まれた列の更新の数。

**NonBndSpts**

非境界リーフの分割数。

### PgAllocs

割り振られたページ数。

**Merges** 索引ページに実行されたマージの数。

### PseuDels

疑似削除としてマークされたキーの数。

### DelClean

削除された疑似削除キーの数。

### IntNodSp1

中間レベル分割の数。

## -temptable パラメーター

すべての一時表における累積圧縮比率を計算するには、以下の公式を使用できます。

$$\% \text{ Compression} = \frac{\text{Total Bytes Saved}}{\text{Total Bytes Saved} + \text{Total Stored Temp Bytes}}$$

注:

- 用語 Eligible は、圧縮基準を満たす一時表を示します。
- 用語 Compressed は、圧縮される十分なデータが最終的に挿入された一時表を示します。

```
hotel26:/home/billyp> db2pd -db billdb -temptable
```

```
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
```

すべてのカウンターは、非表示の `reset` オプションを使用するとゼロにリセットできます。

```
hotel26:/home/billyp> db2pd -db bill -temptable reset
Resetting counters to 0.
```

**db2pd -temptable** コマンドの出力例を参照してください。

## -thresholds パラメーター

**-thresholds** パラメーターについては、以下の情報が戻されます。

- Threshold Name: しきい値名
- Threshold ID: しきい値 ID
- Domain: しきい値ドメイン

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

- Domain ID: しきい値ドメイン ID
- Predicate ID: しきい値述部 ID
- Maximum Value: しきい値最大値
- Statement Text: ステートメントしきい値に関連付けられたステートメント・テキスト。
- Enforcement: しきい値強制の有効範囲
- Queuing: しきい値はキューイングしきい値です
- Queue Size: しきい値のキュー・サイズ設定
- Collect Flags: しきい値の COLLECT ACTIVITY DATA オプションの設定
- Partition Flags: COLLECT ACTIVITY オプション設定が適用されるパーティション
- Execute Flags: しきい値のアクション設定
- Enabled: しきい値の状態 (使用可能または使用不可)
- Check Interval (seconds): しきい値の頻度設定
- Remap Target Serv. Subclass: しきい値アクションの再マップ用のターゲット・サービス・サブクラス設定
- Log Violation Evmon Record: THRESHOLD VIOLATIONS イベント・モニター・ログの設定

さらに、しきい値がキューイングしきい値である場合、キュー・セクションには次の情報も表示されます。

- Queue information for threshold: しきい値名
- Max Concurrency: 最大並行性の設定
- Concurrency: 実際の並行性値
- Max Queue Size: 最大しきい値キュー・サイズ設定
- Agents Currently Queued: カタログ・データベース・パーティション上のしきい値キューで待機中のすべてのエージェントのリスト (エージェントがキューに入られている場合にのみ表示されます)

### -transactions パラメーター

**-transactions** パラメーターについては、以下の情報が戻されます。

#### **App1Handl**

トランザクションのアプリケーション・ハンドル。

#### **TranHdl**

トランザクションのトランザクション・ハンドル。

**Locks** トランザクションによって保持されているロックの数。

**State** トランザクションの状態。

**Tflag** トランザクション・フラグ。可能な値は次のとおりです。

- 0x00000002。この値は、2 フェーズ・コミット・アプリケーションのコーディネーター・ノードにのみ書き込まれ、すべての従属ノードが「コミットの準備」要求を送ったことを示します。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

- 0x00000020。トランザクションはキャプチャー・ソース表を変更する必要があります (データ・レプリケーションにのみ使用されます)。
- 0x00000040。クラッシュ・リカバリーは、トランザクションが準備状態にあると見なします。
- 0x00010000。この値は、パーティション・データベース環境のコーディネーター・パーティションにのみ書き込まれ、コーディネーター・パーティションが 2 フェーズ・コミット・トランザクションのすべての従属パーティションからまだコミット要求を受け取っていないことを示します。
- 0x00040000。トランザクションのロールバックはペンディング状態です。
- 0x01000000。トランザクションは、コーディネーター・パーティションではないデータベース・パーティション・サーバーで更新されました。
- 0x04000000。疎結合 XA トランザクションはサポートされています。
- 0x08000000。複数の分岐がこのトランザクションに関連付けられており、疎結合 XA プロトコルを使用しています。
- 0x10000000。トランザクションに参加する分岐が疎結合 XA プロトコルを使用できないことを示すデータ定義言語 (DDL) ステートメントが出力されました。

**Tf1ag2** トランザクション・フラグ 2。値は以下のとおりです。

- 0x00000004。トランザクションは、**num\_log\_span** データベース構成パラメーターによって指定された限界を超えました。
- 0x00000008。トランザクションは、DB2 ユーティリティーを実行した結果として発生しました。
- 0x00000020。トランザクションは、優先順位の高いアプリケーションにロックを譲渡します (通常、この値は DB2 データベース・システムがセルフチューニングおよび自己管理機能のために自動的に開始するジョブに関して発生します)。
- 0x00000040。トランザクションは、優先順位の高いアプリケーションに行レベルのロックを譲渡しません (通常、この値は DB2 データベース・システムがセルフチューニングおよび自己管理機能のために自動的に開始するジョブに関して発生します)。

### **Firstlsn**

トランザクションの最初の LSN。

### **Lastlsn**

トランザクションの最終 LSN。

### **Firstlso**

トランザクションの最初の LSO。

### **Lastlso**

トランザクションの最終 LSO。

### **SpaceReserved**

トランザクションのために予約されているログ・スペースの量。

### **LogSpace**

トランザクションに必要なログ・スペースの合計 (補正ログ・レコード用の使用済みスペースおよび予約済みスペースを含む)。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

**TID** トランザクション ID。

### **AxRegCnt**

グローバル・トランザクションに登録されているアプリケーションの数。ローカル・トランザクションの場合、この値は 1 です。

**GXID** グローバル・トランザクション ID。ローカル・トランザクションの場合、この値は 0 です。

### **ClientUserID**

トランザクションのクライアント・ユーザー ID。これは tpmon\_client\_userid (TP Monitor Client User ID モニター・エレメント) と同じです。

### **ClientWrkstnName**

トランザクションのクライアント・ワークステーション名。これは tpmon\_client\_wrkstn (TP Monitor Client Workstation Name モニター・エレメント) と同じです。

### **ClientApp1Name**

トランザクションを実行しているクライアント・アプリケーション名。これは tpmon\_client\_app (TP Monitor Client Application モニター・エレメント) と同じです。

### **ClientAcctng**

トランザクションを実行しているクライアントのアカウントリング・ストリング。これは tpmon\_acc\_str (TP Monitor Client Accounting String モニター・エレメント) と同じです。

## **-utilities パラメーター**

**-utilities** パラメーターについては、以下の情報が戻されます。

**ID** ユーティリティー呼び出しに対応するユニーク ID。

**Type** ユーティリティーのクラスを識別します。

**State** ユーティリティーの状態について記述します。

### **Invoker**

ユーティリティーの呼び出し方法について記述します。

### **Priority**

そのスロットルされたピアとの関連におけるスロットル・ユーティリティーの相対的な重要度を指定します。優先度 0 は、ユーティリティーがスロットルなしで実行することを暗黙に示します。ゼロ以外の優先度は 1 から 100 までの範囲になければならず、100 が最高の優先度、1 が最低の優先度を表します。

### **StartTime**

現行のユーティリティーが最初に呼び出された日時を指定します。

**DBName** ユーティリティーによる操作対象になったデータベースを識別します。

### **NumPhases**

ユーティリティーにあるフェーズの数を指定します。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

### CurPhases

現在実行中のフェーズを指定します。

### Description

ユーティリティーが実行している作業の要旨。これには、ロード操作 ID およびアプリケーション ID が含まれます。

### -wlocks パラメーター

**-wlocks** パラメーターについては、以下の情報が戻されます。

#### App1Hand1

アプリケーション・ハンドル (ノードと索引を含む)。

#### TranHd1

ロックを要求しているトランザクション・ハンドル。

#### LockName

ロックの名前。

**Type** ロック・タイプ。

**Mode** ロック・モード。可能な値は次のとおりです。

- IS
- IX
- S
- SIX
- X
- IN
- Z
- U
- NS
- NW

**Conv** ロック待機の終了後にロックが変換されるロック・モード。

**Sts** ロック状況。可能な値は次のとおりです。

- G (譲渡)
- C (変換)
- W (待機)

#### CoorEDU

アプリケーションのコーディネーター・エージェントの EDU ID。

#### AppName

アプリケーションの名前。

**AuthID** 許可 ID。

**AppID** アプリケーション ID。この値は **app1\_id** モニター・エレメント・データと同じです。

**db2pd -wlocks** コマンドの出力例を参照してください。



### **-workactionsets パラメーター**

**-workactionsets** パラメーターについては、以下の情報が戻されます。

- アドレス
- 作業アクション・セットの ID
- 作業アクション・セット名
- 関連する作業クラス・セットの ID
- 作業アクション・セットに関連したオブジェクトの種類 (データベースまたはサービス・クラス)
- 作業アクション・セットに関連したオブジェクト (サービス・クラスまたはデータベース) の ID
- 以下のような、作業アクション・セット内のすべての作業アクション:
  - アドレス
  - アクション ID
  - アクションの種類
  - 参照オブジェクト ID (アクションの種類に応じて、しきい値 ID、サービス・クラス ID、または NULL)

### **-workclasssets パラメーター**

**-workclasssets** パラメーターについては、以下の情報が戻されます。

- アドレス
- 作業クラス ID
- 参照カウンター (この作業クラス・セットを参照する異なる作業アクション・セットの数)
- 次のような、作業クラス・セット内のすべての作業クラス (評価順序で表示されます):
  - アドレス
  - クラス ID
  - クラス名
  - 属性
  - 作業タイプ
  - timeron コストの下限
  - timeron コストの上限

### **-workloads パラメーター**

**-workloads** パラメーターの場合、見出しに関しては、以下の情報が戻されます。

#### **ワークロード定義**

- ワークロード ID および名前
- ワークロード・オカレンスのデータベース・アクセス権
- 並列処理の最大度合い
- 並行ワークロード・オカレンスの数

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

- ワークロードしきい値
- 関連サービス・クラス
- 統計収集設定
- ヒストグラム・テンプレート ID

### 使用特権所有者

- ワークロード ID
- 所有者のタイプ
- 許可 ID

### ローカル・パーティションのワークロード統計

- ワークロード ID および名前
- ワークロード・オカレンス統計
- 最終統計リセット後の経過時間
- アクティビティ統計

**db2pd -workloads** コマンドの出力例を参照してください。

## 出力例

### -addnode

以下は、**db2pd -addnode** コマンドの出力例です。

```
-----
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

データベース・パーティション・サーバーの追加操作が行われる前に、インスタンス内にあるデータベース・パーティション・サーバーの数を参照するアプリケーションに関する情報を戻します。

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

以下は、**db2pd -addnode oldviewsapps** コマンドの出力例です。

```
-----  
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 db2pd** コマンドで使用する場合は、進行中のステップと、データベース・パーティション・サーバーの追加操作と互換性のないイベントを含む、データベース・パーティション・サーバーの追加操作に関する詳細情報を戻します。**oldviewapps** オプションと併用すると、最近追加されたデータベース・パーティション・サーバーが組み込まれていないインスタンスを調べるアプリケーションに関する情報も戻します。

以下は、**db2pd -addnode detail** コマンドの出力例です。

```
-----  
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
```

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

```
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]

-----

以下の例では、競合操作が示されています。

Total [01] Conflicting application handles for partition [20]

-----

Agents for app\_handle 00000000 00000052 : Activity occurrence:[1] time(s) ActivityName:[exclusive use of db]

以下は、db2pd -addnode oldviewapps detail コマンドの出力例です。

-----

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
```

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

```
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

以下は、**db2pd -apinfo** コマンドの出力例です。

```
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
```

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

```

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
ClientAcctng : 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

**db2pd -catalogcache** コマンドの SYSTABLES および TABLESPACES の出力例を以下に示します。

```

Catalog Cache:
Configured Size      819200
Current Size         537464
Maximum Size         4294901760
High Water Mark      589824

```

```

SYSTABLES:
Address          Schema  Name          Type TableID TbspaceID LastRefID CatalogCacheLoadingLock CatalogCacheUsageLock Sts
0x00002B3AD9CB5C40 SYSCAT TABLESPACES V 0 0 165 010000003A2B0000405CCBD9C3 000005000C110000405CCBD9C3 V
0x00002B3AD9E97680 DBUSER1 SALES T 16 2 164 010000003A2B00008076E9D9C3 000005000E1B00008076E9D9C3 S
0x00002B3AD9E5F920 DBUSER1 VSTAFAC2 V 0 0 164 010000003A2B000020F9E5D9C3 00000500001D000020F9E5D9C3 V
0x00002B3AD9BEDD60 DBUSER1 PRODUCT T 4 4 164 010000003A2B000060DDBED9C3 00000500061D000060DDBED9C3 S
0x00002B3AD9E62920 SYSIBM SYSPERIODS T 164 0 164 010000003A2B00002029E6D9C3 00000500050800002129E6D9C3 V
0x00002B3AD9E6E1A0 DBUSER1 EMP_PHOTO T 7 2 164 010000003A2B0000A0E1E6D9C3 00000500021B0000A0E1E6D9C3 V
0x00002B3AD9E5A500 SYSPUBLIC HMON_COLLECTION 0 0 0 164 010000003A2B0000A05E5D9C3 00000000000000000000000000000000 I
0x00002B3AD9E11C60 SYSIBM SYSTABLES T 5 0 164 010000003A2B0000601CE1D9C3 0000050004000000611CE1D9C3 V

0x00002B3AD9E6D060 DBUSER1 EMP_RESUME T 8 2 164 010000003A2B000060D0E6D9C3 00000500031B000060D0E6D9C3 V
0x00002B3AD9CB56A0 SYSTOOLS POLICY T 4 5 164 010000003A2B0000A056CB9C3 000005000D1D0000A056CB9C3 V
0x00002B3AD9E66C60 DBUSER1 EMPLOYEE T 6 2 164 010000003A2B0000606CE6D9C3 00000500001B0000606CE6D9C3 S
0x00002B3AD9CB6E00 SYSCAT TABLES V 0 0 164 010000003A2B000000E6CB9C3 000005000B11000000E6CB9C3 V
0x00002B3AD9E642E0 DBUSER1 EMPPROJECT T 11 2 164 010000003A2B0000E042E6D9C3 00000500071B0000E042E6D9C3 S
0x00002B3AD9DA26A0 DBUSER1 CUSTOMER T 6 4 164 010000003A2B0000A026DAD9C3 00000500081D0000A026DAD9C3 S
0x00002B3AD9E996E0 DBUSER1 ACT T 12 2 164 010000003A2B0000E096E9D9C3 000005000A1B0000E196E9D9C3 V

```

```

TABLESPACES:
Address          Name          TbspaceID LastRefID CatalogCacheLoadingLock CatalogCacheUsageLock Sts
0x00002B3AD9BED6C0 SYSCATSPACE 0 164 110000003A2B0000C0D6BED9C3 0000210004000000C0D6BED9C3 V
0x00002B3AD9BE3080 TEMPSPACE1 1 31 110000003A2B00008030BED9C3 00002100050000008030BED9C3 V
0x00002B3AD9BF2F00 USERSPACE1 2 164 110000003A2B0000002FBFD9C3 0000210006000000002FBFD9C3 V
0x00002B3AD9E62EC0 IBMDB2SAMPLEXML 4 164 110000003A2B0000C02EE6D9C3 0000210008000000C02EE6D9C3 V
0x00002B3AD9BF2E00 SYSTOOLSPACE 5 164 110000003A2B0000002EBFD9C3 0000210009000000002EBFD9C3 V

```

# db2pd - DB2 データベースのモニターおよびトラブルシューティング

## -cleaner

以下は **-cleaner** オプションの出力例です。

```
db2pd -db sample -cleaner

Database Partition 0 - Database SAMPLE - Active - Up 0 days 00:06:34 -
Date 08/09/2010 14:17:58

Page Cleaners:

Group ID   Clnr Idx   State Cycle   Task Pgs Gthr   Pgs Ga'd   IO  outstnd Max AIO   Pgs Thrsh   Pgs D Stl   Pgs Skppd
-1         0          Wait 0          None 0           0           0    32      0         0           0
-1         1          Wait 0          None 0           0           0    32      0         0           0
-1         2          Wait 0          None 0           0           0    32      0         0           0
-1         3          Wait 0          None 0           0           0    32      0         0           0
-1         4          Wait 0          None 0           0           0    32      0         0           0
...
Dirty lists for Bufferpool ID : 1
List ID    # Dirty   Clnr Idx   Trigger Target LSN      Pgs for Gap
0          4         0          None   000000000000000 0
0          0         0          None   000000000000000 0
1          8         1          None   000000000000000 0
1          0         1          None   000000000000000 0
2          2         2          None   000000000000000 0
2          0         2          None   000000000000000 0
3          1         3          None   000000000000000 0
```

## -dirtypages

以下は **-dirtypages** オプションの出力例です。

```
db2pd -db sample -dirtypages

Database Partition 0 -- Database SAMPLE -- Active -- Up 0 days 00:00:20 --
Date 08/09/2010 14:11:44

Bufferpool: 1
Dirty pages %      : 34 / 1000 (3.40% dirty)
Bufferpool minbuflsn: 000000000138800C

Oldest page info:
DirtyLst  TspID PPNum   ObjID OPNum   Typ UFlag fixcount   wgt CPC LSN          pgLtch
n/a       0       327        15   3         4 3 0           2 0 000000000138800C 0x07000000323508E8

Dirty pages:
DirtyLst  TspID PPNum   ObjID OPNum   Typ UFlag fixcount   wgt CPC LSN          pgLtch
0         0       272        14   0         0 3 0           2 0 000000000138881C 0x070000003236C2E8 hX:0 sH:0 xW:0 rC:0
0         0       273        14   1         0 3 0           1 0 000000000138881C 0x070000003236B228 hX:0 sH:0 xW:0 rC:0
0         0       7541       18   9         1 3 0           2 0 000000000138E237 0x07000000323678A8 hX:0 sH:0 xW:0 rC:0
0         0       7540       18   8         1 3 0           2 0 000000000138E402 0x0700000032367A28 hX:0 sH:0 xW:0 rC:0
1         0       6945       15   5         1 3 0           2 0 0000000001388107 0x070000003236F3E8 hX:0 sH:0 xW:0 rC:0
1         0       300        14   4         1 3 0           2 0 000000000138889D 0x070000003236B6A8 hX:0 sH:0 xW:0 rC:0
...
Recovery information:
lowtranlsn      : 000000000138E486
minbuflsn      : 000000000138800C
nextlsn        : 000000000138E4B0
LFH lowtranlsn : 000000000138E486
LFH minbuflsn  : 000000000138800C
LFH nextlsn    : 000000000138800C
Active Log bytes in use : 25764
Current Softmax : 4096000

DirtyLst - dirty list ID in this bufferpool
TspID - tablespace ID of this page
PPNum - pool page number
ObjID - object ID
OPNum - object page number
Typ - type of the object
UFlag - internal page flag
fixcount - number of active fixes on this page (in-use count)
wgt - weight of the page
CPC - clock
LSN - page LSN
pgLtch - page latch address, hX - held X, sH - # shard holders, xW - # of
X waiters
```

**-edus** 以下は、**db2pd -edus** コマンドの出力例です。

```
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
```

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

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

インターバル (例えば **db2pd -edus interval=10**) を含めた場合は、出力の SYS 列の右側に、次の 2 つの列がさらに追加されます。

...	USR DELTA	SYS DELTA
...	0.141799	0.045431
...	0.101154	0.045117
...	0.038113	0.020154
...	0.005668	0.007978
...	0.005139	0.004392
...	0.005003	0.004105
...	0.003913	0.004100
...	0.001785	0.001282
...	0.001083	0.001550
...	0.001005	0.000433
...	0.000181	0.000098
...	0.000095	0.000091
...	0.000000	0.000000
...	0.000000	0.000000
...	0.000000	0.000000

### -fmpexechistory | -fmpe

以下は、**db2pd -fmpexechistory** コマンドの出力例です。

```
db2pd -fmpexechistory -pid 761872 -n 10
```

```
Database Partition 0 -- Active -- Up 0 days 00:00:11
```

```
FMP Process:
```

```
FmpPid      Bit   Flags      ActiveThrd PooledThrd ForcedThrd Active
761872      64   0x00000002 2           1           1           YES
```

```
Active Threads:
```

```
EduPid: 123456 ThreadId: 987654
```

```
RoutineID  Timestamp
1          2009-05-06-17.12.30.000000
2          2009-05-06-17.12.30.005000
1          2009-05-06-17.12.30.100000
```

```
EduPid: 234567 ThreadId: 987000
```

```
RoutineID  Timestamp
1          2009-05-06-17.12.31.000000
3          2009-05-06-17.12.30.000000
```

```
Pooled Threads:
```

```
ThreadId: 540021
RoutineID  Timestamp
4          2009-05-06-17.10.30.000000
```



## db2pd - DB2 データベースのモニターおよびトラブルシューティング

```
Forced Threads:
ThreadId: 120021
RoutineID Timestamp
10          2009-05-06-15.10.30.000000
```

以下は、genquery オプションを指定した **db2pd -fmpexechistory** コマンドの出力例です。

```
db2pd -fmpExecHistory pid=761872 n=10 genquery
```

```
Database Partition 0 -- Active -- Up 0 days 00:00:11
```

```
WITH RTNHIST ( PID, TID, RTNID, RTNTIME) AS
  ( VALUES (761872, 987654, 1, TIMESTAMP('2009-07-13-16.17.10.818705')),
    (761872, 987654, 2, TIMESTAMP('2009-07-13-16.17.11.818710')),... )
SELECT R.PID, R.TID, R.RTNTIME, ROUTINESCHEMA, ROUTINEMODULENAME, ROUTINENAME, SPECIFICNAME, ROUTINEID
FROM syscat.routines, RTNHIST as R
WHERE ROUTINEID = R.RTNID
ORDER BY R.PID, R.TID, R.RTNTIME ;
```

### -logs

以下は、**db2pd -logs** コマンドの出力例です。

### -pages

以下は、summary パラメーターを指定しないときの **db2pd -pages** コマンドの出力例です。

```
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
```

以下は、summary パラメーターを指定するときの **db2pd -pages** コマンドの出力例です。

```
venus@baryon:/home/venus =>db2pd -pages summary -db pdtest
```

```
Database Partition 0 -- Database PDTEST -- Active -- Up 0 days 00:02:07
```

```
Bufferpool Pages:
```

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

```
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

### -reorgs index

以下のセクションは、**-reorgs index** パラメーターを使用して取得した出力の例で、2 つのパーティションを使用する範囲パーティション表の索引再編成の進行状況を報告しています。

**注:** 最初の出力は、非パーティション索引の索引再編成統計を報告しています。それに続く出力は、各パーティションのパーティション索引の索引再編成統計を報告しています。ここでは、出力ごとに 1 つのパーティションのみの索引再編成統計が報告されています。

Index Reorg Stats:

Retrieval Time: 02/08/2010 23:04:21

TbSpaceID: -6 TableID: -32768

Schema: ZORAN TableName: BIGRPT

Access: Allow none

Status: Completed

Start Time: 02/08/2010 23:03:55 End Time: 02/08/2010 23:04:04

Total Duration: 00:00:08

Prev Index Duration: -

Cur Index Start: -

Cur Index: 0 Max Index: 2 Index ID: 0

Cur Phase: 0 ( - ) Max Phase: 0

Cur Count: 0 Max Count: 0

Total Row Count: 750000

Retrieval Time: 02/08/2010 23:04:21

TbSpaceID: 2 TableID: 5

Schema: ZORAN TableName: BIGRPT

PartitionID: 0 MaxPartition: 2

Access: Allow none

Status: Completed

Start Time: 02/08/2010 23:04:04 End Time: 02/08/2010 23:04:08

Total Duration: 00:00:04

Prev Index Duration: -

Cur Index Start: -

Cur Index: 0 Max Index: 2 Index ID: 0

Cur Phase: 0 ( - ) Max Phase: 0

Cur Count: 0 Max Count: 0

Total Row Count: 375000

Retrieval Time: 02/08/2010 23:04:21

TbSpaceID: 2 TableID: 6

Schema: ZORAN TableName: BIGRPT

PartitionID: 1 MaxPartition: 2

Access: Allow none

Status: Completed

Start Time: 02/08/2010 23:04:08 End Time: 02/08/2010 23:04:12

Total Duration: 00:00:04

Prev Index Duration: -

Cur Index Start: -

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

```
Cur Index: 0          Max Index: 2          Index ID: 0
Cur Phase: 0          ( - )          Max Phase: 0
Cur Count: 0          Max Count: 0
Total Row Count: 375000
```

### -scansharing

以下のセクションは、**-scansharing** パラメーターを使用した出力の例です。出力には、2 つの共有セットが示されています。表スキャン・セットには 2 つのグループがあり、ブロック索引のスキャン・セットにはグループが 1 つあります。

```
Database Partition 0 -- Database SAMP -- 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	ApplID	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	ApplID	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	ApplID	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

### -serverlist

サーバー・リストの出力例を以下に示します。

```
db2pd -serverlist -db sample のサーバー・リスト出力例
```

```
Database Member 0 -- Active -- Up 0 days 00:10:43 -- Date 10/06/2010 12:22:39
```

#### Server List:

```
Time:          Wed Oct  6 12:13:17
Database Name: SAMPLE
Count:         2
```

Hostname	Non-SSL Port	SSL Port	Priority
coralxib23.torolab.ibm.com	49712	0	34
coralxib24.torolab.ibm.com	49712	0	65

```
db2pd -serverlist -alldbs のサービス・サブクラス出力例
```

```
Database Member 0 -- Active -- Up 0 days 00:06:15 -- Date 10/06/2010 12:18:11
```

#### Server List:

```
Time:          Wed Oct  6 12:13:17
Database Name: SAMPLE
Count:         2
```

Hostname	Non-SSL Port	SSL Port	Priority
coralxib23.torolab.ibm.com	49712	0	34
coralxib24.torolab.ibm.com	49712	0	65

```
Database Member 0 -- Active -- Up 0 days 00:06:15 -- Date 10/06/2010 12:18:11
```

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

### Server List:

Time: Wed Oct 6 12:17:00  
Database Name: SAMPLE2  
Count: 2

Hostname	Non-SSL Port	SSL Port	Priority
coralxib23.torolab.ibm.com	49712	0	56
coralxib24.torolab.ibm.com	49712	0	43

### -serviceclasses

以下は、1 つのサービス・スーパークラスとそのサブクラスに関するサービス・クラス情報の出力例です。

サービス・スーパークラスの出力例:

```
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
CPU Shares              = 1000
CPU Share Type          = Soft
CPU Limit               = 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

サービス・サブクラスの出力例:

```
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
```

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

```
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	db2pclnr
6834	2943347600	db2pfchr
6834	2942299024	db2pfchr
6834	2941250448	db2pfchr

## **-storagegroups** および **-storagepaths**

以下のセクションに示すのは、**-storagegroups** パラメーターまたは **-storagepaths** パラメーターを使用した出力の例です。

```
db2pd -db testdb -storagegroups
```

### Storage Group Configuration:

Address	SGID	Defltd	DataTag	Name
0x00002BA9E6CFF4C0	0	Yes	0	IBMSTOGROUP
0x00002BA9E6D0F4C0	1	No	1	SG_SSD
0x00002BA9E6D1DAE0	2	No	5	SG_IBMSAN

### Storage Group Statistics:

Address	SGID	State	NumPaths	NumDropPen
0x00002BA9E6CFF4C0	0	0x00000000	1	0
0x00002BA9E6D0F4C0	1	0x00000000	2	0
0x00002BA9E6D1DAE0	2	0x00000000	2	0

### Storage Group paths:

Address	SGID	PathID	PathState	PathName
0x00002BA99CD23540	0	0	InUse	/filesystem1
0x00002BA99CE13540	1	1024	InUse	/filesystem2
0x00002BA99CF03540	1	1025	InUse	/filesystem3
0x00002BA99D0F3540	2	2048	InUse	/filesystem4
0x00002BA99D1E3540	2	2049	InUse	/filesystem5

## **-tablespaces**

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

以下は、**db2pd -tablespaces** コマンドの出力例で、PathsDropped や PathID などの、データベースの該当する情報が表示されています (一部の列は、読みやすくするために省略されています)。

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:

...

Tablespace Storage Statistics:

Address	Id	DataTag	Rebalance	SGID	SourceSGID
0x00002BA9E6CFF4C0	0	0	No	0	-
0x00002BA9E6D0F4C0	1	5	No	0	-
0x00002BA9E6D1DAE0	2	1	Yes	1	0
0x00002BA9E73696C0	3	5	No	0	-

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

表が手動ストレージによって管理されている場合、SGID はダッシュ (-) を出力します。

新しい「Max HWM」列が **db2pd -tablespaces** 出力に追加され、この列はインスタンスの開始後の DMS 表スペースの最大 HWM を示します。出力内の「HWM」列は一時 DMS 表スペースに関する現在の HWM であり、使用されているディスク・スペース量の特定時点の値を表します。SMS 表スペースの場合、「HWM」と「Max HWM」には値が入りません。

照会が発行された後に **db2pd** を使用すると、最後のトランザクションで使用された一時表スペースに関するメモリー内の情報を使用できます。以下の例には、新しいこの列が太字で表示されています。「Max HWM」の値は常時、HWM 以上になります。

```
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	3	SYSCATSPACE
0x00002B9DCA583560	1	DMS	UsrTmp	4096	2	Yes	2	1	1	Off	1	0	1	DMSUSRTEMP
0x00002B9DCA5863E0	2	DMS	Large	4096	32	Yes	32	1	1	Off	1	0	31	USERSPACE1
0x00002B9DCA587220	3	DMS	SysTmp	4096	2	Yes	2	1	1	Off	1	0	1	DMSYSTEMP
0x00002B9DCA58A0A0	4	DMS	Large	4096	4	Yes	4	1	1	Off	1	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

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

### -temptable

またシステム・モニター・エレメントを使用してバッファ・プールの読み書き量を調べると、一時表圧縮の有効性を判別できます。以下は、**db2pd -temptable** コマンドの出力例です。

```
hotel26:/home/billyp> db2pd -db billdb -temptable
```

```
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
```

同じ情報が、システム一時表とユーザー一時表に格納されます。ただし、前述のすべてのカウンターは累積されてゆき、一時表がドロップされると更新されます。そのため、こうしたカウンターは履歴情報のみを表します。

### -thresholds

以下は、データベースしきい値とそのキューに関するしきい値情報の例です。

```
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
```

データベースしきい値のキュー出力例:

```
Database Threshold Tickets:
```

```
Ticket information for threshold: TH1 with threshold ID 1
```

```
Activity ID  UOW ID      Classification  AppHandl  [nod-index]
1            6              READ_DML      51        [000-00051]
```

```
Queue information for threshold: MAXDBACTIVITIES
```

```
Max Concurrency      = 2
Concurrency           = 2
Max Queue Size       = 0
```

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

Agents Currently Queued:

EDU ID	AppHandl	[nod-index]	Agent Type	Activity ID	UOW ID
36	14994	[000-14994]	1	4	1

以下に示すのは、ステートメントしきい値に関するしきい値情報の例です。

```
db2pd -thresholds -db sample db2
```

```
Database Member 0 -- Database SAMPLE -- Active -- Up 0 days 00:01:28 -- Date 04/13/2011 09:57:09
```

Statement Thresholds:

```
Threshold Name      = T_THRESHOLD_FIRING_23
Threshold ID        = 1
Domain              = 60
Domain ID           = 1
Predicate ID        = 30
Maximum Value       = 60
Enforcement         = D
Queueing            = N
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
Statement Text      = CREATE TABLE T2 (X INT)
```

```
$ db2 "select thresholdname, domain from sysibm.systhresholds"
```

THRESHOLDNAME	DOMAIN
T_THRESHOLD_FIRING_23	SQ

1 record(s) selected.

**-sort** 以下は、**db2pd -sort** コマンドの出力例です。

**-wlocks**

以下は、**db2pd -wlocks** コマンドの出力例です。

```
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	00020008000000000340000452	Row	..X	G	352614	db2bp	VENUS		*LOCAL.venus.071117030309
15	[000-00015]	9	00020008000000000340000452	Row	.NS	W	1176046	db2bp	VENUS		*LOCAL.venus.071117030358
12	[000-00012]	2	00020008000000000340000452	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

**-workclasssets**

以下に示すのは、基本作業クラス情報の出力例です。

Work Class Sets:

Address	ClassSetID	ReferenceCounter
0x00002BA89DDDF5AE0	1	1

Work Classes:

Address	ClassSetID	ClassID	ClassName
0x00002BA89DDDF5BC0	1	1	WCDML

Attributes:

Work Type: DML

Timeron Cost From: 1 Timeron Cost To: 1000

Address	ClassSetID	ClassID	ClassName
0x00002BA89DDDF5C40	1	2	WCDDL

Work Type: DML

**-workloads**

以下は、デフォルト・ワークロード **SYSDEFAULTUSERWORKLOAD** と **SYSDEFAULTADMWORKLOAD** の出力例です。



## db2pd - DB2 データベースのモニターおよびトラブルシューティング

Database Partition 0 -- Database SB -- Active -- Up 0 days 00:00:57

### Workload Definitions:

Address = 0x00002B3E772ACB40  
WorkloadID = 1  
WorkloadName = SYSDEFAULTUSERWORKLOAD  
DBAccess = ALLOW  
Maximum Degree = 4  
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  
Maximum Degree = DEFAULT  
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

## db2pd - DB2 データベースのモニターおよびトラブルシューティング

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

## db2pdcfg - 問題判別動作の DB2 データベースの構成

DB2 データベース・メモリー・セット内のフラグを、データベース・システムの問題判別用の動作に影響を与えるように設定します。

### 許可

以下のいずれかの権限レベルが必要です。

- SYSADM 権限レベル。
- SYSCTRL 権限レベル。
- SYSMANT 権限レベル。
- SYSMON 権限レベル。

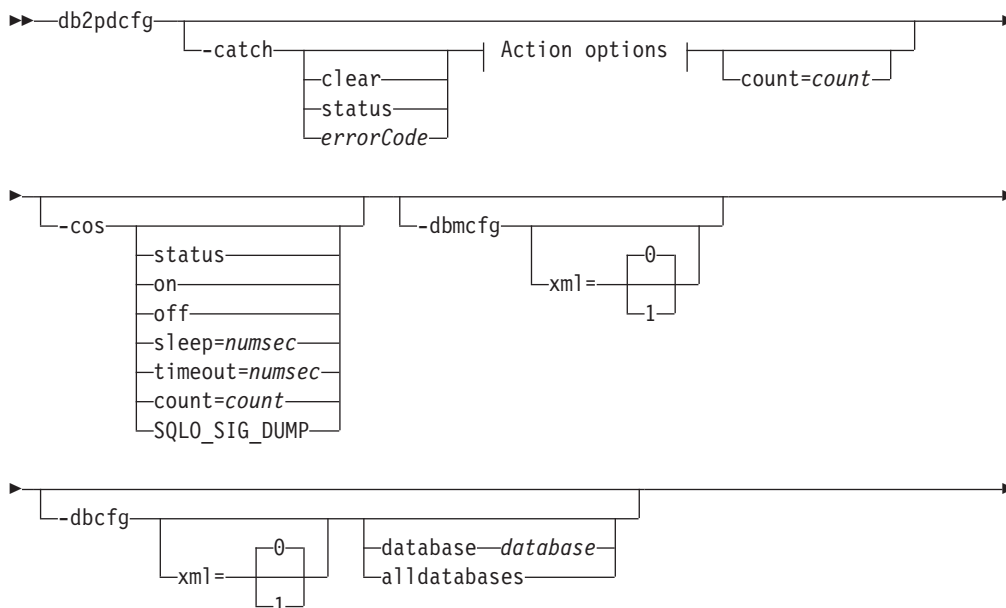
SYSMON 権限レベルが付与される場合、以下のオプションは使用不可になります。

- catch
- cos
- dbcfg
- dbmcf
- fodc
- trapresilience

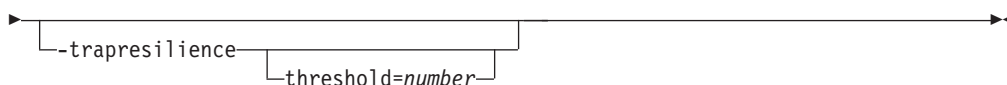
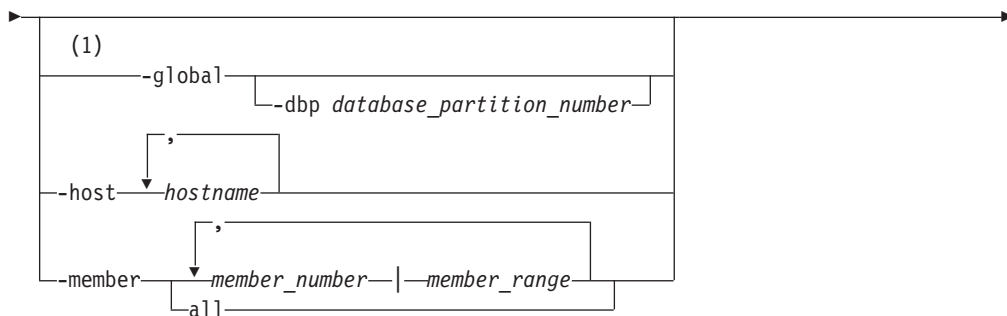
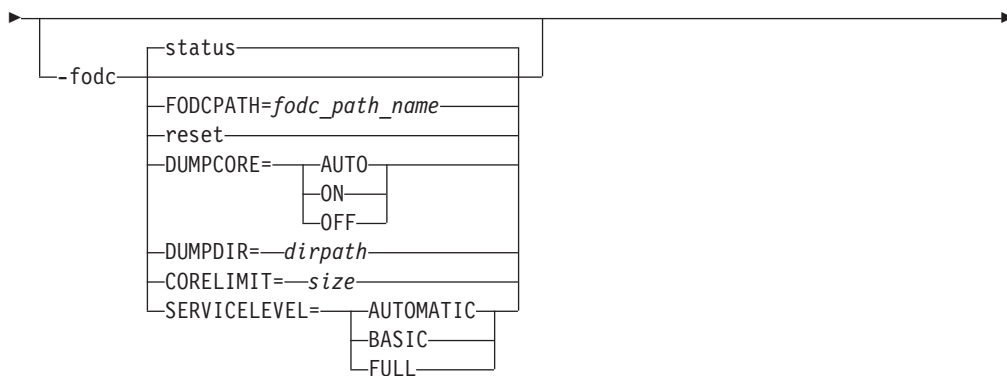
### 必要な接続

接続に必要な最低限の要件はありません。しかし、データベースの有効範囲オプションが指定されている場合、コマンドが要求された情報を戻す前に、そのデータベースをアクティブにする必要があります。

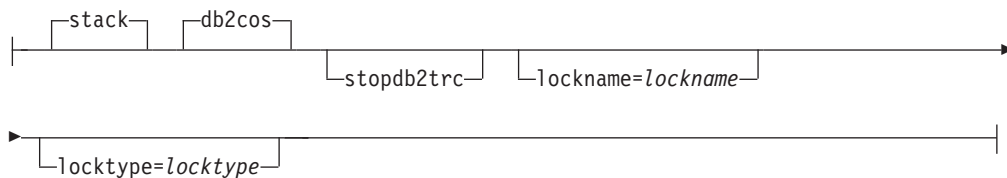
### コマンド構文



## db2pdcfg - 問題判別動作の DB2 データベースの構成



### Action options:



### 注:

- 1 **-global** パラメーターは使用すべきではありません。 **-member all** パラメーター・オプションを使用すると、情報をグローバルに取得することができます。

### コマンド・パラメーター

**-catch** エラーまたは警告をキャッチするようにデータベース・マネージャーに指示します。

**clear** 設定されているすべてのキャッチ・フラグをクリアします。

**status** 設定されているすべてのキャッチ・フラグを表示します。

*errorCode*

設定されている特定のフラグをキャッチします。

考えられる *errorCode* オプションは以下のとおりです。

## db2pdcfg - 問題判別動作の DB2 データベースの構成

- *sqlCode[,reasonCode]* / *sqlCode=sqlCode[,reasonCode]*
- ZRC (16 進数または整数)
- ZRC #define (SQLP\_LTIMEOUT など)
- ADM (ADM1611 など)
- スtring (diagstr="Hello World" など)
- ECF (16 進数または整数)
- "deadlock" または "locktimeout"

管理用通知メッセージ (ADM1611) や **db2diag** ログ・ファイルにある String ("Hello World") などの、指定された特定のフラグをキャッチすると、**db2cos** スクリプトはその情報をダンプします。

**stack db2diag** ログ・ファイルにスタック・トレースを作成します。デフォルト。

**db2cos bin** ディレクトリーにある **db2cos** コールアウト・スクリプトを実行します。デフォルト。

### stopdb2trc

**db2trc** コマンドを停止します。

### lockname=*lockname*

特定のロックをキャッチするためのロック名  
(lockname=000200030000001F0000000052)。

### locktype=*locktype*

特定のロックをキャッチするためのロック・タイプ (locktype=R または locktype=52)。

### count=*count*

データベース・マネージャーのトラップ中にデータベース・マネージャーが **db2cos** を実行する回数。デフォルトは 255 です。

**-cos** データベース・マネージャーのトラップ時に **db2cos** コールアウト・スクリプトを呼び出す方法を、データベース・マネージャーに指示します。

**status** 状況を印刷します。

**off** データベース・マネージャーのトラップ中の **db2cos** 呼び出しをオフにします。

**on** データベース・マネージャーのトラップ中の **db2cos** 呼び出しをオンにします。

### sleep=*numsec*

**db2cos** によって生成される出力ファイル・サイズの検査が終了した状態が続くときにスリープするまでの時間の長さ。デフォルトは 3 秒です。

### timeout=*numsec*

待機時に **db2cos** スクリプトが完了したと見なすまでの時間の長さ。デフォルトは 30 秒です。

### count=*count*

データベース・マネージャーのトラップ中に **db2cos** を実行する回数。デフォルトは 255 です。

### SQLO\_SIG\_DUMP

SQLO\_SIG\_DUMP シグナルの受信時に **db2cos** の実行を有効にします。

### -dbmcfg

DBM Config Reserved Bitmap を設定します。このオプションは、IBM DB2 サービスから取得できるパスワードで保護されています。

#### xml=0 | 1

値 0 (デフォルト) または 1 (インスタンスには xml データがある)。

**-dbcfg** Database Config Reserved Bitmap を設定します。このオプションは、IBM DB2 サービスから取得できるパスワードで保護されています。

#### xml=0 | 1

値 0 (デフォルト) または 1 (データベースには xml データがある)。

**-fodc** DB2 データベース・メモリー・セット内のフラグを設定します。First Occurrence Data Collection (FODC) を含む問題判別の状態にある間は、データベース・システムの動作に影響があります。

サポートされている **-fodc** オプションと、それらのオプションの有効な値とデフォルトを以下に示します。

**reset** すべての FODC オプションをデフォルトにリストアします。

**status** すべての FODC オプションの状況を表示します。これはデフォルト・オプションです。つまり、パラメーターを指定しないで **db2pdcfg** を呼び出すと、FODC 状況が表示されます。

### FODCPATH=*fodc\_path\_name*

FODC データ・パッケージを作成する既存ディレクトリーの絶対パス名を指定します。指定するパスは、マシン上のメンバーが書き込み可能な場所でなければなりません。さらに、そのメンバーまたはパーティション上で実行される **fmp** プロセスで書き込み可能な場所ではなければなりません。

### DUMPCORE=

UNIX および Linux オペレーティング・システム上でのみ、コア・ファイルの生成を有効または無効にします。

**AUTO** トラップを維持できず、インスタンスがシャットダウンされる場合に、コア・ファイルが生成されます。

**ON** コア・ファイルの生成を有効にして、**DB2RESILIENCE** レジストリー変数の設定をオーバーライドします。

**OFF** コア・ファイルの生成を無効にします。

### DUMPDIR=*dirpath*

コア・ファイルまたは共有メモリーのダンプが作成されるディレクトリーの絶対パス名を指定します。このオプションは、コア・ファイルや共有メモリーのダンプだけでなく、FODC パッケージ以外に保管する必要のある他の大規模なバイナリー・ダンプにも使用

## db2pdcfg - 問題判別動作の DB2 データベースの構成

できます。デフォルトは **diagpath** ディレクトリーで、**diagpath** が定義されていない場合はデフォルトの診断ディレクトリーです。

### **CORELIMIT=size**

作成されるコア・ファイルの最大サイズ。この値は、現在のコア・ファイルのサイズ制限の設定をオーバーライドします。unlimited のデフォルト値が適用されます。コア・ファイルは非常に大きくなる可能性があるため、使用可能なファイル・システム・スペースを考慮する必要があります。実際のサイズは、DB2 の構成、および問題発生時のプロセスの状態に応じて異なります。

**db2pdcfg** を使って **CORELIMIT** を変更する場合には、通常の UNIX アクセス権限の制約を受け、場合によっては **CORELIMIT** は **ulimit** 設定値を上回ることができません。**DB2FODC** レジストリー変数を使用して **db2start** でこの値を変更するか、DB2 製品の始動前に大きな **ulimit** 設定値を使用してください。

### **SERVICELEVEL=**

データ破損を示す可能性のあるパニック、トラップ、またはエラーの発生時におけるデータの収集方法を指定します。デフォルトの動作設定は **AUTOMATIC** です。**BASIC** または **FULL** のいずれかを動的に設定すると、前に行った **DUMPCORE** 設定がオーバーライドされることに注意してください。したがって、その設定をそのまま保持したい場合、これは **db2pdcfg** 設定の一部として指定する必要があります。この設定は動的に行っても、**CF** が障害により再始動した場合や、意図的に停止して再始動された場合を除き、**CF** には影響がありません。このパラメーターでは次のオプションがサポートされています。

#### **AUTOMATIC**

この設定では、有効な **SERVICELEVEL** 設定 (**BASIC** または **FULL**) が実行時 (メンバーの場合) および開始時 (**CF** プロセスの場合) に選択されるよう指定します。現時点では、**BASIC** が選択されるのは、複数のメンバーを持つ DB2 pureScale 環境の場合と、トラップ回復を目的とした場合のみです。

#### **BASIC**

この **SERVICELEVEL** 設定では、ダンプ出力される **FODC** データの最小量を指定します。コア・ダンプ処理はデフォルトで無効になっており (ただし、**COREDUMP** 設定によりオーバーライドできます)、診断は影響を受けるスレッドのみに限定され、コールアウト・スクリプトは無効になっています。

#### **FULL**

この **SERVICELEVEL** 設定では、ダンプ出力される **FODC** データの最大量を指定します。これには、コア・ダンプ、関連付けられているすべてのコンポーネント・ダンプ、およびコールアウト・スクリプトの呼び出しが含まれます。また、トラップの維持は試行されません。

### **-global**

リモート・ホストに対しても **db2pdcfg** を実行することを指定します。

## db2pdcfg - 問題判別動作の DB2 データベースの構成

**-file** パラメーターを指定した場合、リモート・ホスト上のすべての個別ファイルから成る単一ファイルが、**db2pdcfg** コマンドが発行されたコンピューター上に作成されます。

**注:** このコマンド・パラメーターは、DB2 バージョン 9.8 フィックスパック 3 以降で使用できます。このコマンド・パラメーターは、DB2 バージョン 9.7 フィックスパック 4 およびそれ以降のフィックスパックでは非推奨です。

### **-dbp** *database partition number*

指定したデータベース・パーティションのリモート・ホストに対して **db2pdcfg** を実行することを指定します。 **-global** オプションでデータベース・パーティションを指定しなかった場合は、すべてのリモート・ホストに対して **db2pdcfg** が実行されます。

### **-host** *hostname*

コマンド発行先ホスト (複数も可) を指定します。このコマンドは、ホスト上にあるすべてのメンバーに対して発行されます。このオプションを指定しない場合、コマンドはローカル・ホストに対して発行されます。複数のホストを指定する場合、すべてのホスト名が有効でないと、コマンドを完了できません。このオプションは、**-member** オプションと一緒に指定することはできません。

### **-member** *member\_number* | *member\_range*

コマンド発行先メンバー (複数も可) を指定します。このオプションを指定しない場合、コマンドは現行メンバーに対して発行されます。複数のメンバーを、*member\_number* のコンマ区切りリスト (*member1*, *member2*) として、または *member\_range* (*member\_range* はメンバーの範囲 (*member1*-*member3*)) を使用して指定できます。最初の 2 つの方式を組み合わせ使用することもできます。このオプションは、**-host** オプションと一緒に指定することはできません。

**all** リモート・ホスト上のメンバーを含め、すべてのメンバーに対してコマンドを発行するように指定します。

### **-trapresilience**

このオプションは、問題判別のために、トラップ回復パラメーターを表示または変更します。

以下は、このオプションを指定した場合の出力例です。

```
DB2 trap resilience is enabled.  
Current threshold setting : 0 (threshold disabled)  
Number of traps sustained : 0
```

### **threshold=number**

デフォルト値: 0 (しきい値は無効)

最小値: 0

最大値: 4294967295

**threshold=** オプションの後に数値を指定するとトラップの上限が設定され、これはインスタンスが存在する限り維持されます。このしきい値を超えると、次のトラップを維持できるかどうかにかかわらずインスタンスは停止します。

以下は、このオプションを指定した場合の出力例です。



## db2pdcfg - 問題判別動作の DB2 データベースの構成

```
db2pdcfg -trapresilience threshold=1
```

```
DB2 trap resilience threshold is set to 1
```

### 例

メンバー 11、12、13、および 15 の現行の FODC パッケージ設定を表示するには、以下のようにします。

```
db2pdcfg -fodc status -member 11-13,15
```

FODC パッケージをローカル・ホスト上のディレクトリーに送信するには、以下のようにします。

```
db2pdcfg -fodc FODCPATH=/home/hotel49/user/FODC/FODClcal
```

### 使用上の注意

**db2pdcfg** は、FODC オプションを (オンラインで) 動的に変更するためのメソッドです。

**db2pdcfg** は DB2 データベース・メモリー中のフラグを設定するので、**db2pdcfg** ツールを使用して加えられる変更は、インスタンスの稼働中のみアクティブになります。変更を永続させるには、**DB2FODC** レジストリー変数を使用してください。

**-fodc** オプションの一部の設定は、*variable=value* の形式で指定します。複数のオプションを 1 つのコマンド行で指定できます。

```
db2pdcfg -fodc DUMPCORE=ON -fodc CORELIMIT=8GB
```

またはその代わりに、スペースを使用して複数の設定を 1 つのコマンド行ストリングに連結することもできます。

```
db2pdcfg -fodc DUMPCORE=ON CORELIMIT=8GB
```

オプションを何も指定しないで **db2pdcfg** コマンドを実行すると、トラップ回復に関して情報を提供する以下の要約出力が提供されます (太字で強調表示されています)。

```
$ 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: 300
Current Count: 0
Max Count: 255
```

```
Current bitmap value: 0x0
```

```
Instance is not in a sleep state
```

```
DB2 trap resilience is enabled.
Current threshold setting : 0 ( disabled )
Number of traps sustained : 0
```

## db2pdcfg - 問題判別動作用の DB2 データベースの構成

Database Member 0

FODC (First Occurrence Data Capture) options:  
Dump directory for large objects (DUMPDIR)= /home/hotel85/vivmak/sql1lib/db2dump/  
Dump Core files (DUMPCORE)= AUTO  
Current hard core file size limit = Unlimited  
Current soft core file size limit = 0 Bytes

## db2perfc - データベース・パフォーマンス値のリセット

1 つ以上のデータベースのパフォーマンス値をリセットします。これは、Windows オペレーティング・システムのパフォーマンス・モニターで使用されます。

### 許可

ローカル管理者

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

**-d** DCS データベースのパフォーマンス値をリセットすることを指定します。

**dbalias** パフォーマンスの値をリセットするデータベースを指定します。何もデータベースが指定されない場合は、アクティブなデータベースすべてのパフォーマンスの値がリセットされます。

### 例

次の例では、アクティブな DB2 データベースすべてのパフォーマンス値をリセットします。

```
db2perfc
```

次の例では、特定の DB2 データベースのパフォーマンス値をリセットします。

```
db2perfc dbalias1 dbalias2
```

次の例では、アクティブな DB2 DCS データベースすべてのパフォーマンス値をリセットします。

```
db2perfc -d
```

次の例では、特定の DB2 DCS データベースのパフォーマンス値をリセットします。

```
db2perfc -d dbalias1 dbalias2
```

### 使用上の注意

アプリケーションが DB2 モニター API を呼び出したときに戻される値は、DB2 サーバーが開始して以来の累積値になります。しかし、パフォーマンス値をリセットし、テストを実行し、再び値をリセットしてからテストを再実行する方がよいでしょう。

## db2perf - データベース・パフォーマンス値のリセット

プログラムは、関係する DB2 サーバー・インスタンスのデータベース・パフォーマンス情報に現行でアクセスしているすべてのプログラム (つまり、**db2perf** を実行するセッションで **db2instance** に保持されているプログラム) の値をリセットします。また、**db2perf** を呼び出すと、コマンド実行時にリモート側から DB2 のパフォーマンス情報にアクセスしていたすべてのユーザーに表示されている値もリセットされます。

**db2ResetMonitor** では、グローバルにではなく、ローカルに特定のデータベースを参照している値をリセットできます。

## db2perfi - パフォーマンス・カウンター登録ユーティリティ

DB2 パフォーマンス・カウンターを Windows オペレーティング・システムに追加します。これは、Windows パフォーマンス・モニターが DB2 および DB2 Connect のパフォーマンス情報にアクセスできるようにするために実行する必要があります。

### 許可

ローカル管理者

### 必要な接続

なし

### コマンド構文

```

▶▶ db2perfi [-i] [-u]

```

### コマンド・パラメーター

- i DB2 パフォーマンス・カウンターを登録します。
- u DB2 パフォーマンス・カウンターの登録を解除します。

### 使用上の注意

**db2perfi -i** コマンドは、以下のアクションを行います。

1. Windows レジストリーに DB2 カウンター・オブジェクトの名前と説明を追加します。
2. Windows レジストリーの Services キーに、次のようにレジストリー・キーを作成します。

```

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

```

## db2perfr - パフォーマンス・モニター登録ツール

**db2perfr** コマンドは、Windows オペレーティング・システムのパフォーマンス・モニターにアクセスするときに、管理者のユーザー名およびパスワードを DB2 に登録するために使用します。

これにより、リモート・パフォーマンス・モニター要求は DB2 データベース・マネージャーによって正しく識別され、関連した DB2 パフォーマンス情報にアクセスできるようになります。パフォーマンス・ログ機能を使用してファイルにカウンター情報を記録する場合にも、管理者ユーザー名およびパスワードを登録する必要があります。

### 許可

ローカル管理者

### 必要な接続

なし

### コマンド構文

```
▶▶ db2perfr -r username password ▶▶  
           -u
```

### コマンド・パラメーター

- r ユーザー名およびパスワードを登録します。
- u ユーザー名およびパスワードを登録解除します。

### 使用上の注意

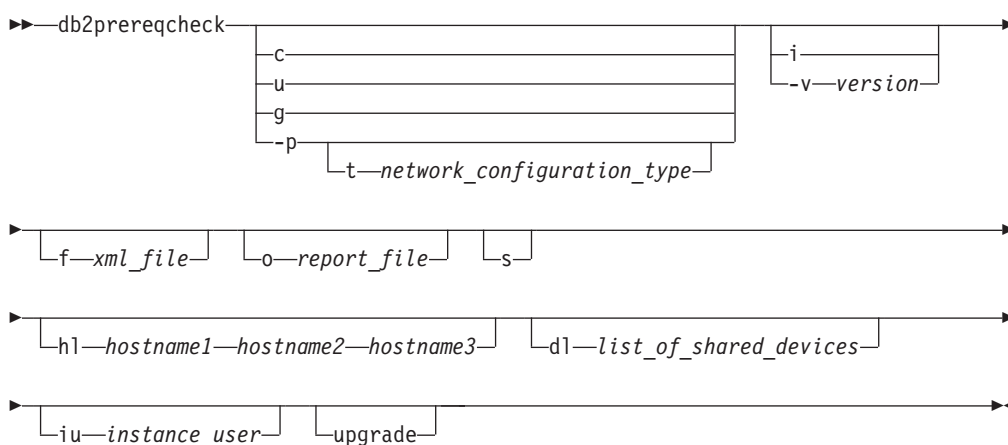
- いったんユーザー名とパスワードの組み合わせを DB2 データベース・システムに登録すると、パフォーマンス・モニターのローカル・インスタンスであっても、そのユーザー名とパスワードを使って明示的にログオンします。つまり、DB2 データベース・システムに登録されたユーザー名情報と一致しなければ、パフォーマンス・モニターのローカル・セッションに、DB2 のパフォーマンス情報は示されないこととなります。
- ユーザー名とパスワードの組み合わせは、Windows セキュリティー・データベースに保管されているユーザー名およびパスワードの値と常に一致していなければなりません。Windows セキュリティー・データベースのユーザー名またはパスワードの値が変更された場合、リモート・パフォーマンス・モニターで使用されるユーザー名とパスワードの組み合わせを再設定しなければなりません。
- デフォルトの Windows パフォーマンス・モニターのユーザー名 SYSTEM は、DB2 データベース製品の予約語であるため使用できません。

## db2prereqcheck - インストール前提条件の検査

特定のバージョンの DB2 for Linux, UNIX, and Windows をインストールするための前提条件を、対象のシステムが満たしているかどうか検査します。

このコマンドを使用することにより、該当の DB2 for Linux, UNIX, and Windows をダウンロードしてインストール・プロセスを開始する前に、対象のシステムが前提条件を満たしているかどうか判別できます。**db2prereqcheck** コマンドは、前提条件を含むリソース XML ファイルを使用します。XML ファイルのデフォルトのパスは、DB2 installation/cfg/DB2prereqs.xml です。この XML ファイルの読み取り権限または書き込み権限がなければなりません。XML ファイルの内容を変更しないでください。

### コマンド構文



### コマンド・パラメーター

- c シン・クライアントの前提条件を検査します。
  - c、-u、-g、および -p パラメーターは、相互に排他的です
- u DB2 pureScale 環境の uDAPL 要件を検査します (Linux および AIX のみ)
  - c、-u、-g、および -p パラメーターは、相互に排他的です。
  - RoCE ネットワークでは、AIX および uDAPL のソフトウェア前提条件が満たされていることを確認する必要があります。**db2prereqcheck** コマンドでは、これらのレベルが自動的に検証されることはありません。
- g GPL コンパイル要件を検査します (Linux のみ)
  - c、-u、-g、および -p パラメーターは、相互に排他的です
- p DB2 pureScale 環境の前提条件を検査します (Linux および AIX のみ)
  - c、-u、-g、および -p パラメーターは、相互に排他的です
- t *network configuration type*
  - t パラメーターを指定すると、特定のタイプのネットワーク構成に関する前提条件が検証されます (Linux のみ)。

## db2prereqcheck - インストール前提条件の検査

network configuration type は SINGLE\_IB\_PORT\_CLUSTER、MULT\_IB\_PORT\_CLUSTER、SINGLE\_ROCE\_PORT\_CLUSTER、または MULTI\_ROCE\_PORT\_CLUSTER でなければなりません。

**-s** 画面に前提条件の確認結果の要約を表示します。

**-i** リソース XML ファイルに定義されている DB2 最新バージョンの前提条件を検査します

XML ファイル内の最新バージョン番号を表示するには、**-s** を指定して **-i** を実行します。

**-i** と **-v** パラメーターは、相互に排他的です。

**-vversion**

特定の DB2 バージョンの前提条件を検査します。このパラメーターは、その他の **db2prereqcheck** コマンド・パラメーターとともに使用できます。

**-i** と **-v** パラメーターは、相互に排他的です。

**-fxml-file**

XML リソース・ファイルの名前を指定します。**-f** パラメーターを指定しないと、デフォルトの XML ファイル (ファイル・パス: DB2 installation/cfg/DB2prereqs.xml) が使用されます。

**-oreport-file**

レポート・ファイルの名前を指定します。例えば、db2prereqcheck.rpt と指定します。

**-h1 hostname1,hostname2,hostname3**

すべてのホスト間で root のパスワードなし SSH アクセスを指定します。

**-d1 list\_of\_shared\_devices**

すべてのホストからアクセスを受ける PVID (AIX) または WWN (Linux) のリストを指定します。

**-iu instance\_user**

すべてのホストのインスタンス・ユーザーおよび fenced ユーザーの同一ユーザー ID (UID) およびグループ ID (GID) を指定します。

**-upgrade**

バージョン 9.8 GA またはバージョン 9.8 フィックスパック 1 からアップグレードする際に、前提条件を確認して、netmon.cf ファイル内の必要な項目が正しいことを確認します。

### 例

システムが DB2 バージョン 10.1.0.0 の前提条件を満たしているかどうか検査するには、次のコマンドを発行します。

```
db2prereqcheck -v 10.1.0.0 -s
```

システムが DB2 バージョン 10.1 のシン・クライアントの前提条件を満たしているかどうか検査するには、次のコマンドを発行します。

```
db2prereqcheck -c -v 10.1.0.0 -s
```



## db2rbind - すべてのパッケージの再バインド

データベース内のパッケージを再バインドします。

### 許可

DBADM

### 必要な接続

なし

### コマンド構文

```

▶▶ db2rbind database -l logfile
    |_____| |_____|
    |all|   |-u-userid--p-password|
    |
    |_____|
    |_____|
    |conservative|
    |_____|
    |-r-|
    |any|
  
```

### コマンド・パラメーター

#### *database*

再び妥当性検査を行うパッケージが含まれているデータベースの別名を指定します。

#### **-l** *logfile*

パッケージの再妥当性検査プロセスを記録するときに使用するパス (任意指定) とファイル名 (必須) を指定します。

以下に例を示します。

```

cat <logfile>
Starting time .... Thu Jun 18 02:47:11 2009
Succeeded to rebind = 0
Failed to rebind = 0
Ending time .... Thu Jun 18 02:47:11 2009
  
```

#### **all**

すべての有効および無効パッケージの再バインドが実行されるように指定します。このオプションを指定しないと、データベース内のすべてのパッケージが検査されますが、アプリケーションの実行時に暗黙的に再バインドされることのないよう、無効のマークが付いたパッケージのみを再バインドしません。

#### **-u** *userid*

ユーザー ID。パスワードを指定する場合には、このパラメーターを指定しなければなりません。

#### **-p** *password*

パスワード。ユーザー ID を指定する場合には、このパラメーターを指定しなければなりません。

#### **-r**

解決方法。パッケージの再バインドの実行に、従来のバインド・セマンティクスを使用するかどうかを指定します。これは、解決に SQL パスを使用する新しいオブジェクトが、パッケージ内の静的 DML ステートメントの解

## db2rbind - すべてのパッケージの再バインド

決時に考慮されるかどうかに影響します。このオプションは DRDA ではサポートされていません。有効な値は以下のとおりです。

### conservative

オブジェクト解決に SQL パスを使用するオブジェクトに対する参照を解決するのに、最後の明示的バインドのタイム・スタンプより前に定義された SQL パスのオブジェクトのみが考慮されます。従来のバインド・セマンティクスを使用します。これはデフォルトです。このオプションは、作動不能パッケージではサポートされていません。

### any

オブジェクト解決にその SQL パスを使用するオブジェクトに対する参照を解決するのに、SQL パスにあるすべての可能な組み合わせが考慮されます。従来のバインド・セマンティクスは使用されません。

## 使用上の注意

- このコマンドは、データベース内の全パッケージの妥当性検査の再実行に、再バインド API (sqlarbind) コマンドを使用します。
- 必ずしも **db2rbind** を使用しなければならないわけではありません。
- 無効なパッケージについては、任意で、パッケージの初回使用時に暗黙的にパッケージの再妥当性検査を行わせることができます。パッケージの再妥当性検査には、**REBIND** コマンドと **BIND** コマンドのどちらを使用しても構いません。
- ただし、何らかのパッケージの再バインドでデッドロックが生じたり、ロックがタイムアウトになったりした場合は、すべてのパッケージの再バインドがロールバックされます。
- **db2rbind** コマンドを発行し、その際にインスタンスがアクティブな場合は、SQL1026N エラー・メッセージが出されます。

## db2relocatedb - データベースの再配置

このコマンドは、ユーザー提供の構成ファイルで指定されたとおりに、データベースを名前変更したり、データベースやデータベースの一部 (コンテナ、ログ・ディレクトリーなど) を再配置します。このツールは、DB2 インスタンスおよびデータベース・サポート・ファイルに、必要な変更を行います。

ターゲット・データベースをオフラインにしてからでなければ、**db2relocatedb** コマンドを実行してターゲット・データベースの制御ファイルとメタデータを変更してはなりません。

**db2relocatedb** コマンドがデータベースのファイルおよび制御構造に加える変更は、ログに記録されないため、リカバリー不能です。特に、保存されているログ・ファイルによってデータベースがリカバリー可能な場合、データベースに対するコマンドの実行後にフルバックアップを行うのは良い習慣です。

### 許可

なし

### 前提条件

このコマンドを使用する前に、次のコマンドを実行してファイルを移動しておく必要があります。

```
mv /home/db2inst1/db2inst1/NODE0000/X /home/db2inst1/db2inst1/NODE0000/Y
```

ここで、*X* は古いデータベース名を、*Y* は新しいデータベース名を表します。

この追加のステップを実行して、コマンド **db2relocatedb** が確実にエラー・メッセージなしで実行されるようにする必要があります。このステップは、DB2 バージョン 10.1 以降でのみ必要です。

### コマンド構文

```
►► db2relocatedb -f configFilename ◀◀
```

### コマンド・パラメーター

**-f configFilename**

データベースの再配置に必要な構成情報の入ったファイルの名前を指定します。これは、相対ファイル名でも絶対ファイル名でも構いません。構成ファイルのフォーマットは以下のとおりです。

```
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
...
FAILARCHIVE_PATH=newDirPath
```

```
LOGARCHMETH1=newDirPath
LOGARCHMETH2=newDirPath
MIRRORLOG_PATH=newDirPath
OVERFLOWLOG_PATH=newDirPath
...
```

ここで、

### **DB\_NAME**

再配置されるデータベースの名前を指定します。データベース名を変更する場合は、古い名前と新規の名前の両方を指定する必要があります。このフィールドは必須です。

### **DB\_PATH**

再配置されるデータベースの元のパスを指定します。データベース・パスが変更される場合、古いパスと新規のパスの両方を指定する必要があります。このフィールドは必須です。

### **INSTANCE**

データベースが存在する場所のインスタンスを指定します。データベースが新規のインスタンスに移動される場合、古いインスタンスと新規のインスタンスの両方を指定する必要があります。このフィールドは必須です。

### **NODENUM**

変更されるデータベース・ノードのノード番号を指定します。デフォルトは 0 です。

### **LOG\_DIR**

ログ・パスのロケーション内の変更を指定します。ログ・パスが変更される場合、古いパスと新しいパスの両方を指定する必要があります。ログ・パスがデータベース・パスの下にある場合、パスは自動的に更新されるので、この指定はオプションです。

### **CONT\_PATH**

表スペース・コンテナのロケーション内の変更を指定します。古いコンテナ・パスと新規のコンテナ・パスの両方を指定する必要があります。複数のコンテナ・パスを変更する場合、複数の **CONT\_PATH** 行を指定できます。コンテナ・パスがデータベース・パスの下にある場合、パスは自動的に更新されるので、この指定はオプションです。同じ古いパスが共通の新規パスで置換される場所で、複数のコンテナに変更を行う場合、単一の **CONT\_PATH** 項目が使用されます。このような場合、古いパス、新規パスの両方にアスタリスク (\*) をワイルドカードとして使用できます。

### **STORAGE\_PATH**

データベースのいずれかのストレージ・パスの場所を変更することを指定します。古いストレージ・パスと新しいストレージ・パスの両方を指定する必要があります。複数のストレージ・パスを変更する場合、複数の **STORAGE\_PATH** 行を指定できます。このパラメーターを指定して、すべてのストレージ・グループ内の任意のストレージ・パスを変更することができます。ただし、個々のストレージ・グループのストレージ・パスを変更するために、このパラメーターを指定することはできません。

注: このパラメーターは、AUTOMATIC STORAGE NO 節を指定して作成されたデータベースには適用されません。AUTOMATIC STORAGE NO 節を指定してデータベースを作成できますが、AUTOMATIC STORAGE 節は非推奨になり、将来のリリースで除去される可能性があります。

#### FAILARCHIVE\_PATH

データベース・マネージャーが 1 次アーカイブ・ロケーションにも 2 次アーカイブ・ロケーションにもログ・ファイルをアーカイブできない場合に、ログ・ファイルをアーカイブする新しいロケーションを指定します。このフィールドは、再配置されるデータベースに **failarchpath** 構成パラメーターが設定されている場合のみ指定してください。

#### LOGARCHMETH1

新しい 1 次アーカイブ・ロケーションを指定します。このフィールドは、再配置されるデータベースに **logarchmeth1** 構成パラメーターが設定されている場合のみ指定してください。

#### LOGARCHMETH2

新しい 2 次アーカイブ・ロケーションを指定します。このフィールドは、再配置されるデータベースに **logarchmeth2** 構成パラメーターが設定されている場合のみ指定してください。

#### MIRRORLOG\_PATH

ミラー・ログ・パスの新しいロケーションを指定します。ストリングがパス名を指していなければなりません。また、相対パス名ではなく、絶対パス名でなければなりません。このフィールドは、再配置されるデータベースに **mirrorlogpath** 構成パラメーターが設定されている場合のみ指定してください。

#### OVERFLOWLOG\_PATH

ロールフォワード操作に必要なログ・ファイルの検索、アーカイブから取り出されたアクティブ・ログ・ファイルの保管、db2ReadLog API が必要とするログ・ファイルの検索と保管を行う新しいロケーションを指定します。このフィールドは、再配置されるデータベースに **overflowlogpath** 構成パラメーターが設定されている場合のみ指定してください。

空白行またはコメント文字 (#) で始まる行は無視されます。

## 例

### 例 1

データベース TESTDB の名前を PRODDB に、パス /home/db2inst1 にあるインスタンス db2inst1 で変更するには、以下の構成ファイルを作成します。

```
DB_NAME=TESTDB,PRODDB
DB_PATH=/home/db2inst1
INSTANCE=db2inst1
NODENUM=0
```

## db2relocatedb - データベースの再配置

構成ファイルが作成されたら、すべての自動ストレージ・パスを、新規データベース名と一致するように変更する必要があります。

```
rename /home/db2inst1/db2inst1/TESTDB /home/db2inst1/db2inst1/PRODDB
```

構成ファイルを `relocate.cfg` として保存し、以下のコマンドを使用して、データベース・ファイルへの変更を行います。

```
db2relocatedb -f relocate.cfg
```

### 例 2

データベース `DATAB1` をパス `/dbpath` のインスタンス `jsmith` からインスタンス `prodinst` に移動するには、以下のようになります。

1. ディレクトリー `/dbpath/jsmith` 内のファイルを `/dbpath/prodinst` に移動します。
2. 以下の構成ファイルと **db2relocatedb** コマンドを使用して、データベース・ファイルに変更を行います。

```
DB_NAME=DATAB1
DB_PATH=/dbpath
INSTANCE=jsmith,prodinst
NODENUM=0
```

### 例 3

パス `/databases/PRODDB` のインスタンス `inst1` 内に存在するデータベース `PRODDB` です。2つの表スペース・コンテナのロケーションを、以下のように変更する必要があります。

- SMS コンテナ `/data/SMS1` を `/DATA/NewSMS1` に移動する必要があります。
- DMS コンテナ `/data/DMS1` を `/DATA/DMS1` に移動する必要があります。

物理ディレクトリーおよびファイルが、新規のロケーションに移動された後で、新規のロケーションを認識するように、以下の構成ファイルと **db2relocatedb** コマンドを使用して、データベース・ファイルに変更を行います。

```
DB_NAME=PRODDB
DB_PATH=/databases/PRODDB
INSTANCE=inst1
NODENUM=0
CONT_PATH=/data/SMS1,/DATA/NewSMS1
CONT_PATH=/data/DMS1,/DATA/DMS1
```

### 例 4

インスタンス `db2inst1` に存在するデータベース `TESTDB` は、パス `/databases/TESTDB` に作成されました。表スペースは、以下のコンテナと共に作成されました。

```
TS1
TS2_Cont0
TS2_Cont1
/databases/TESTDB/TS3_Cont0
/databases/TESTDB/TS4/Cont0
/Data/TS5_Cont0
/dev/rTS5_Cont1
```

TESTDB は新規システムに移動されます。新規システムのインスタンスは newinst になり、データベースのロケーションは /DB2 になります。

データベースを移動する場合、/databases/TESTDB/db2inst1 ディレクトリーに存在するすべてのファイルは、/DB2/newinst ディレクトリーに移動する必要があります。これは、最初の 5 つのコンテナが、この移動の一部として再配置されることを意味します。(最初の 3 つはデータベース・ディレクトリーに相対で、次の 2 つはデータベース・パスに相対です。) これらのコンテナがデータベース・ディレクトリーまたはデータベース・パス内にあるため、構成ファイルにリストする必要はありません。2 つの残りのコンテナが新規システムで異なるロケーションに移動された場合は、構成ファイルにリストする必要があります。

物理ディレクトリーおよびファイルが新規のロケーションに移動された後で、新規のロケーションを認識するように、以下の構成ファイルと **db2relocatedb** を使用して、データベース・ファイルに変更を行います。

```
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
```

#### 例 5

データベース TESTDB には、データベース・パーティション・サーバー 10 および 20 に 2 つのデータベース・パーティションがあります。このインスタンスは servinst で、データベース・パスは両方のデータベース・パーティション・サーバーで /home/servinst です。データベースの名前は SERVDB に変更され、データベース・パスは両方のデータベース・パーティション・サーバーで /databases に変更されます。さらに、ログ・ディレクトリーはデータベース・パーティション・サーバー 20 で、/testdb\_logdir から /servdb\_logdir に変更されます。

両方のデータベース・パーティションに変更が行われているため、構成ファイルは各データベース・パーティションに作成され、**db2relocatedb** は対応する構成ファイルを使用する各データベース・パーティション・サーバーで実行される必要があります。

データベース・パーティション・サーバー 10 では、以下の構成ファイルが使用されます。

```
DB_NAME=TESTDB,SERVDB
DB_PATH=/home/servinst,/databases
INSTANCE=servinst
NODENUM=10
```

データベース・パーティション・サーバー 20 では、以下の構成ファイルが使用されます。

```
DB_NAME=TESTDB,SERVDB
DB_PATH=/home/servinst,/databases
INSTANCE=servinst
NODENUM=20
LOG_DIR=/testdb_logdir,/servdb_logdir
```

#### 例 6



## db2relocatedb - データベースの再配置

パス /home/maininst のインスタンス maininst 内に存在するデータベース MAINDB です。4 つの表スペース・コンテナのロケーションを、以下のように変更する必要があります。

```
/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
```

物理ディレクトリーおよびファイルが、新規のロケーションに移動された後で、新規のロケーションを認識するように、以下の構成ファイルと **db2relocatedb** コマンドを使用して、データベース・ファイルに変更を行います。

同様の変更が、すべてのコンテナに対して行われました。すなわち、/maininst\_files/allconts/ が /MAINDB/ で置換され、ワイルドカード文字のある単一項目が使用できるようになります。

```
DB_NAME=MAINDB
DB_PATH=/home/maininst
INSTANCE=maininst
NODENUM=0
CONT_PATH=/maininst_files/allconts/*, /MAINDB/*
```

### 使用上の注意

データベースが属するインスタンスを変更する場合、インスタンスおよびデータベース・サポート・ファイルに確実に変更が加えられるようにするため、このコマンドを実行する前に以下の事柄を行う必要があります。

- データベースが他のインスタンスに移動されている場合は、新規のインスタンスを作成します。新規インスタンスは、現在データベースがあるインスタンスと同じリリース・レベルでなければなりません。
- 新しいインスタンスの所有者が現在のインスタンスの所有者でない場合は、新しいインスタンスの所有者へアクセス権を付与します。
- 新規インスタンスが常駐するシステムにコピーされるデータベースに属するファイルとデバイスをコピーします。パス名は必要に応じて変更する必要があります。ただし、データベース・ファイルの移動先のディレクトリー内にデータベースが既にある場合、不用意に既存の `sqlbdir` ファイルを上書きしてしまって、既存のデータベースへの参照を除去する可能性があります。そのような事態になった場合、**db2relocatedb** ユーティリティーを使用することはできません。その場合、**db2relocatedb** の代わりにリダイレクト・リストア操作を使用できます。
- インスタンス所有者に所有されるように、コピーされたファイル/デバイスのアクセス権を変更します。

複数のデータベースがあるデータベース・パスからデータベースを移動する場合には、そのデータベース・パス内の `sqlbdir` ディレクトリーを移動するのではなくコピーする必要があります。このディレクトリーは、移動されないデータベースを DB2 が見つけるために、そのまま古い場所に置いておくことが必要です。

`sqlbdir` ディレクトリーを新しい場所にコピーした後、`LIST DB DIRECTORY ON newPath` コマンドを実行すると、移動されなかったデータベースがリスト表示され



ます。これらの参照は削除できず、その名前を持つ新しいデータベースをこの同じパス上に作成することもできません。ただし、その名前を持つデータベースを別のパスに作成することはできます。

ALTER TABLESPACE MANAGED BY AUTOMATIC STORAGE ステートメントを用いて自動ストレージを使用するように変換された、表スペースの既存のユーザー作成コンテナを移動するために、**db2relocatedb** コマンドを使用することはできません。

インスタンスが変更されている場合、コマンドは新規のインスタンス所有者によって実行される必要があります。

パーティション・データベース環境では、変更が必要なすべてのデータベース・パーティションに対してこのツールを実行する必要があります。データベース・パーティションごとにそれぞれ別個の構成ファイル (変更対象のデータベース・パーティションの NODENUM 値が含まれる) を用意する必要があります。例えば、データベースの名前を変更する場合は、すべてのデータベース・パーティションが影響を受けることになり、各データベース・パーティションごとに別個の構成ファイルを用意して **db2relocatedb** コマンドを実行する必要があります。単一データベース・パーティションに属するコンテナを移動する場合は、そのデータベース・パーティションに対して一度だけ **db2relocatedb** コマンドを実行することが必要です。

**db2relocatedb** コマンドを使用して、ロードが進行中のデータベースや、**LOAD RESTART** または **LOAD TERMINATE** コマンドの完了を待機しているデータベースを再配置することはできません。

**制約事項:** パーティション・データベース環境では、同じ装置に常駐する複数のロジカル・パーティションの 1 つであるノードの全体を再配置することはできません。

## db2rfe - 非 root インストール用 root フィーチャー使用可能化

構成ファイルに基づいて DB2 データベース・システムの非 root インストールにおいてサポートされる root フィーチャーを使用可能にします。DB2 非 root インスタンスは、**db2rfe** コマンドの実行前に停止させる必要があります。

### 許可

root ユーザー権限

### 必要な接続

なし

### コマンド構文

```
▶▶ db2rfe -f db2rfe_config_file [ -h | -? ] ▶▶
```

### コマンド・パラメーター

**-f** *db2rfe\_config\_file*

root フィーチャーを使用可能にするために使用する構成ファイルを指定します。

**-h | -?**

ヘルプ情報を表示します。

### 使用上の注意

構成ファイル内の各 root フィーチャーが含まれるセクションは、それぞれ分かれています。各セクションには開始マークと終了マークがあり、そのセクションが使用可能にする事柄について説明したコメント、および root フィーチャーを使用可能にするためのコマンドが記されています。サンプル構成ファイル `db2rfe.cfg` が、`$DB2DIR/instance` ディレクトリーにインストールされます。

このサンプル構成ファイルは、以下のようになります (この例では、非 root インストール所有者は `db2inst3` です)。

```
** =====
**
** 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.
```

## db2rfe - 非 root インストール用 root フィーチャー使用可能化

```
** -----  
** 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

---

## db2rfpen - ロールフォワード・ペンディング状態にリセット

データベースをロールフォワード・ペンディング状態にします。高可用性災害時リカバリー (HADR) を使用している場合、データベースは標準データベースにリセットされます。

### 許可

なし

### 必要な接続

なし

### コマンド構文

```
▶▶ db2rfpen -ON database_alias -path global_log_control_files_dir ▶▶
```

### コマンド・パラメーター

#### *database\_alias*

ロールフォワード・ペンディング状態にするデータベースの名前を指定します。高可用性災害時リカバリー (HADR) を使用している場合、データベースは標準データベースにリセットされます。

#### **-path** *global\_log\_control\_files\_dir*

グローバル・ログ制御ファイル `SQLLOGCTL.GLFH.1` とそのミラー・コピー `SQLLOGCTL.GLFH.2` が格納されているディレクトリーの絶対パスを指定します。

---

## db2rmicons - メインメニューからの DB2 ツールの除去

DB2 ツールのメインメニュー項目を除去します。

Linux オペレーティング・システムで、**db2rmicons** コマンドは、現在のユーザーに関する DB2 ツールのメインメニュー項目を除去します。DB2 ツールのメインメニュー項目の除去は、**db2rmicons** コマンドを実行して手動で行うか、特定の DB2 コマンド (例えば、**db2\_deinstall** または **db2idrop**) の実行時に自動的に行われます。非 root インストールの場合、**db2\_deinstall** コマンドは非 root インストールに関連した DB2 インスタンスの項目を除去します。

### 許可

なし

### コマンド構文

```
▶▶ db2rmicons [-h]
```

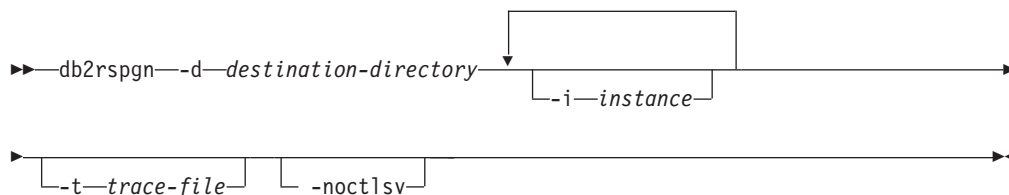
### コマンド・パラメーター

**-h** 使用法情報を表示します。

## db2rspgn - 応答ファイル生成プログラム

現行コピー用の応答ファイルとインスタンス構成プロファイルを生成します。これらの生成ファイルは、他のマシン上で正確なセットアップを再作成するために使用されます。

### コマンド構文



### コマンド・パラメーター

#### -d destination-directory

生成ファイル用の出力ディレクトリーへの絶対パスを指定します。指定された出力ディレクトリーが既存のディレクトリーである場合、ディレクトリーは空であり、書き込み可能でなければなりません。指定された出力ディレクトリーが存在しない場合、そのロケーションが書き込み可能であれば、新規ディレクトリーが作成されます。このパラメーターは、必須です。

#### -i instance

指定されたインスタンス構成を生成し、この情報を生成済み応答ファイルとインスタンス構成プロファイルに保存します。このパラメーターはオプションです。デフォルトでは、すべてのインスタンスが選択されます。複数インスタンスを指定するには、このパラメーターを複数回指定します。例えば、`-i db2inst1 -i db2inst3` とします。

#### -t trace-file

Linux および UNIX オペレーティング・システムのみ。デバッグ・モードをオンにします。デバッグ情報は、`trace-file` として指定されたファイル名に書き込まれます。

#### -noctlsv

Windows オペレーティング・システムのみ。コントロール・サーバー・インスタンスのためのインスタンス構成プロファイル・ファイルは生成しないことを指定します。このパラメーターはオプションです。

### 使用上の注意

**db2rspgn** コマンドは DB2 pureScale 環境ではサポートされていません。

## db2sampl - サンプル・データベースの作成

SAMPLE という名前のサンプル・データベースを作成します。

注: DB2 Workgroup Server Edition および DB2 Express Edition では、SAMPLE データベースにはマテリアライズ照会表 (MQT) とマルチディメンション・クラスター表 (MDC) が含まれており、これはライセンス違反の原因になります。この違反状態を除去する唯一の方法は、DB2 Enterprise Server Edition にアップグレードすることです。

このデータベースは、最初の作成時に自動的に構成されることはありません。ユーザーは、後で SAMPLE データベースに対して **AUTOCONFIGURE** コマンドを発行することができます。

### 許可

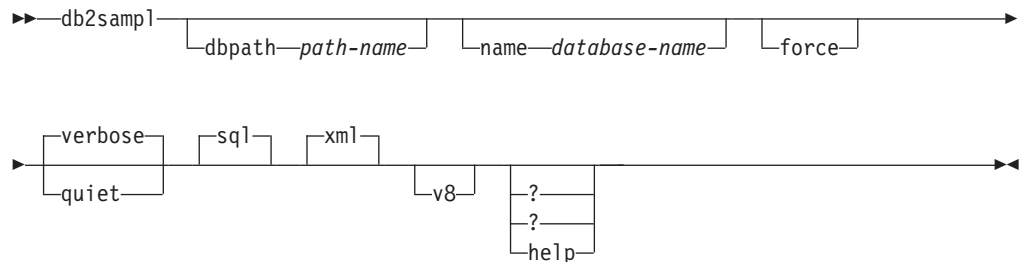
以下の権限のいずれか。

- SYSADM
- SYSCTRL

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### **dbpath** *path-name*

データベースを作成するパスを指定します。Windows オペレーティング・システムでは、データベースの作成先のドライブの文字を指定します。

*path-name* の最大長は 175 文字です。デフォルトでは、*path-name* は、データベース・マネージャ構成ファイル (**dftdbpath** パラメーター) に指定するデフォルトのパスです。

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

サンプル・データベースの名前を指定します。データベース名は、データベースの命名規則に準拠していなければなりません。デフォルトでは、*database-name* は SAMPLE です。

**force** サンプル・データベースに指定されたものと同じ名前の付いたインスタンス内の既存のデータベースを強制的にドロップし、再作成します。

## db2sampl - サンプル・データベースの作成

### verbose

状況メッセージを標準出力に出力します。

**quiet** 標準出力への状況メッセージの出力を抑制します。

**sql** 表、トリガー、関数、プロシージャを作成し、表にデータを追加します。

**xml** データ・タイプ XML の列を持つ表の作成、XML 列に対する索引の作成、XML スキーマの登録を行い、これらの表に XML 文書の値を含むデータを追加します。

このオプションがサポートされるのは、XML がサポートされている場合のみです。XML をサポートしない場合、このオプションは無視されます。

**v8** DB2 Universal Database バージョン 8 のサンプル・データベース、データベース・オブジェクトおよびデータを作成します。バージョン 8 サンプル・データベースは SAMPLE という名前の非 Unicode データベースであり、これはデータベース・マネージャ構成ファイルに指定する (**dftdbpath** パラメーター) デフォルトのパス内に作成されます。

### ? | ? | help

**db2sampl** コマンド構文ヘルプを戻します。

### db2sampl のデフォルトの動作

オプションの引数をまったく付けずに **db2sampl** コマンドを発行すると、環境がパーティション化されているかどうかに応じて、次のように動作が異なります。

非パーティション・データベース環境の場合:

- Unicode (UTF-8) コード・セットを使用して SAMPLE という名前のデータベースをデフォルト・データベース・パス内に作成します。
- 表、索引、制約、トリガー、関数、プロシージャ、マルチディメンション・クラスター表、およびマテリアライズ照会表を含め、リレーショナル・データベース・オブジェクトを作成します。
- リレーショナル表にデータを追加します。
- XML データ・タイプ列をもった表を作成します。
- XML データに対する索引を作成します。
- XML スキーマ文書を収容する XML スキーマ・リポジトリを作成します。
- すべてのデータベース・オブジェクト名は、CURRENT\_SCHEMA 特殊レジスターの値で修飾されます。

パーティション・データベース環境の場合:

- Unicode (UTF-8) コード・セットを使用して SAMPLE という名前のデータベースをデフォルト・データベース・パス内に作成します。
- 表、索引、制約、トリガー、関数、プロシージャ、マルチディメンション・クラスター表、およびマテリアライズ照会表を含め、リレーショナル・データベース・オブジェクトを作成します。
- 表にデータを追加します。
- すべてのデータベース・オブジェクト名は、CURRENT\_SCHEMA 特殊レジスターの値で修飾されます。



## 使用上の注意

- **db2sampl** コマンドは、DB2 データベース・サーバーがインストールされているコンピューター上でのみ発行できます。リモート IBM Data Server Client からは発行できません。
- サンプル・データベースは、データベース・マネージャー構成パラメーター **authentication** で指定したインスタンス認証タイプを使って作成されます。

## 例

- デフォルトの特性をもったサンプル・データベースを作成するには、以下を発行します。

```
db2sampl
```

- Windows オペレーティング・システムで、デフォルトのスキーマ内に SQL データベース・オブジェクトのみが入った **mysample** という名前のサンプル・データベースを E: ドライブ上に作成し、状況メッセージを表示するには、以下を発行します。

```
db2sampl -dbpath E -name mysample -sql -force -verbose
```

- DB2 バージョン 8 のサンプル・データベースを作成するには、以下を発行します。

```
db2sampl -v8
```

## db2schex - Active Directory スキーマ拡張

Microsoft Active Directory スキーマを拡張して、Windows Server 2003 以降で Lightweight Directory Access Protocol (LDAP) ディレクトリー・サーバー機能を使用するために必要な DB2 オブジェクト・クラスおよび属性定義を組み込みます。

このコマンドは、DB2 製品をインストールしてデータベースを作成する前に実行する必要があります。そうしないと、ノードとカタログを手動でデータベースに登録しなければなりません。詳しくは、『LDAP ディレクトリー・サービス用の Active Directory スキーマの拡張 (Windows)』のトピックを参照してください。

**db2schex** コマンドは、製品の DVD に収録されています。このコマンドの場所は、DVD のパス `x:%db2%\windows%\utilities` で、`x:` には DVD ドライブを指定します。

### 許可

Active Directory スキーマを更新するには、Schema Administrators グループのメンバーであるか、スキーマを更新する権限を委任されている必要があります。

### 必要な接続

ターゲット・ドメインの Windows ドメイン・コントローラー・サーバーへのアクセス

### コマンド構文

```
db2schex [-b bindDN] [-w password] [-k] [-u] [-x filename]
```

### コマンド・パラメーター

**-b** *bindDN*

ユーザーの識別名を指定します。

**-w** *password*

バインド・パスワードを指定します。

**-k**

エラーを無視して、強制的にアンインストールを続行します。

**-u**

スキーマをアンインストールします。

**-x** *filename*

このパラメーターは、ユーティリティーによって実行される、Active Directory スキーマへの変更内容をファイルに書き込むときに指定します。

### 例

DB2 スキーマをインストールするには、次のコマンドを実行します。

```
db2schex
```

DB2 スキーマをインストールしてバインド DN とパスワードを指定するには、次のコマンドを実行します。

```
db2schex -b "cn=A_Name,dc=toronto1,dc=ibm,dc=com" -w password
```

または、

```
db2schex -b Administrator -w password
```

DB2 スキーマをアンインストールするには、次のコマンドを実行します。

```
db2schex -u
```

DB2 スキーマをアンインストールしてエラーを無視するには、次のコマンドを実行します。

```
db2schex -u -k
```

### 使用上の注意

*bindDN* と *password* が指定されていない場合、**db2schex** は現在ログインしているユーザーとしてバインドします。

*bindDN* パラメーターは、Windows ユーザー名として指定できます。

DB2 スキーマ拡張コマンドは、以下のタスクを実行します。

- どのサーバーが Schema Master かを検出する
- Schema Master であるドメイン・コントローラーにバインドする
- ユーザーが、クラスと属性をスキーマに追加するための十分な権限を持つことを確実にする
- Schema Master を書き込み可能にする (すなわちレジストリーの安全インターロックを解除する)
- すべての新規属性を作成する
- すべての新規オブジェクト・クラスを作成する
- エラーを検出し、エラーが発生した場合、プログラムはスキーマへの変更をすべてロールバックする

## db2set - DB2 プロファイル・レジストリー

DB2 プロファイル変数の値を表示、設定、または削除します。 **db2set** コマンドは、DB2 Administration Server (DAS) を介したローカル管理およびリモート管理をサポートする外部環境レジストリー・コマンドです。

ENV\_GET\_REG\_VARIABLES 表関数を使用して、インスタンスが使用しているレジストリー変数の値や、レジストリーに格納されている値を取得することもできます。

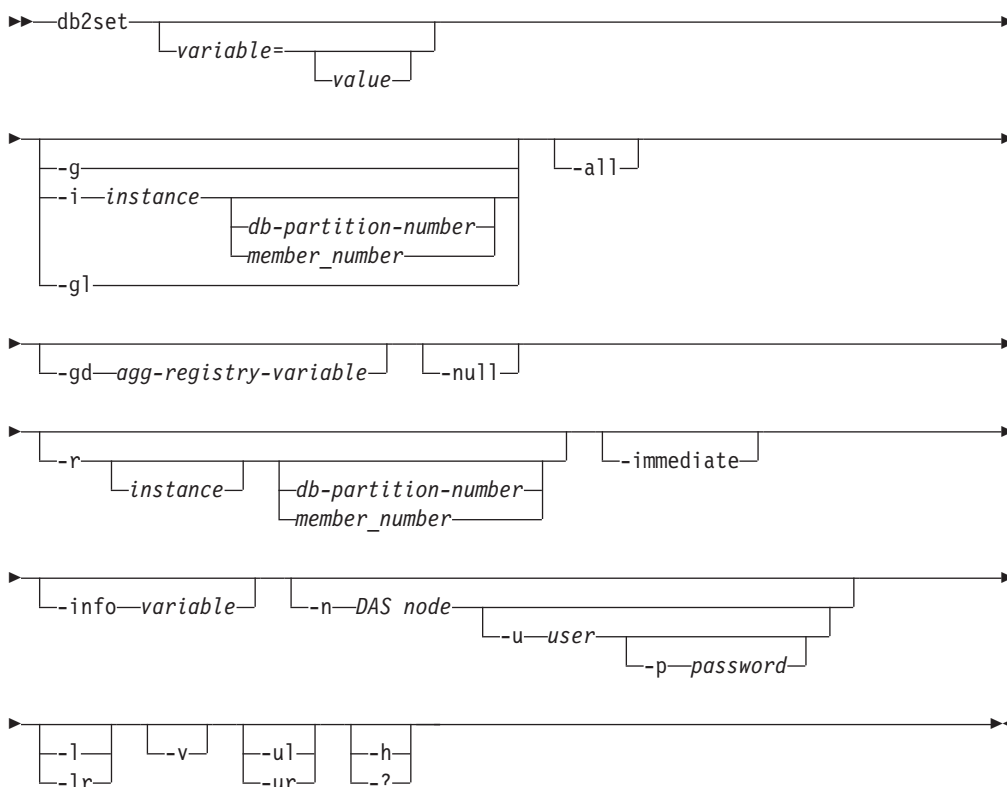
### 許可

SYSADM。加えて、**-g** コマンド・パラメーターには、root アクセス (Linux および UNIX オペレーティング・システムの場合) またはローカル管理者権限 (Windows オペレーティング・システムの場合) が必要になります。

### 必要な接続

変数を即時更新する場合は、ローカルのインスタンス接続が必要です。インスタンス接続がなければ、ローカルのインスタンス接続が作成されます。それ以外の場合、接続は不要です。リモート接続はサポートされません。

### コマンド構文



### コマンド・パラメーター

**variable=**  
指定された変数の値を表示します。

*value* 指定変数に指定値を設定します。等号 (=) の後に値を指定しなければ、変数の現行値が削除されます。デフォルトで即時有効になる一連のレジストリー変数と、**-immediate** パラメーターを指定すれば即時有効になる一連のレジストリー変数があります。どちらのタイプのレジストリー変数も、次の SQL ステートメントのコンパイル時に有効になります。あるいは、インスタンスを再始動した後に有効になります。

**-g** 特定の DB2 コピーに関係のあるすべてのインスタンスのグローバル・プロファイル・レジストリー変数にアクセスします。これによって、このレベルの変数の値を表示または変更することができます。

**-i** *instance*

使用するインスタンス・プロファイルを指定します。インスタンス・プロファイルを指定しない場合、現行インスタンスまたはデフォルトのインスタンスが使用されます。

*db-partition-number* または *member-number*

db2nodes.cfg ファイル内の番号を指定します。

**-immediate** パラメーターを *member-number* オプションと一緒に使用すると、特定のメンバーだけが次の SQL ステートメントのコンパイル時の更新を識別します。

**-gl** LDAP に保管されたグローバル・プロファイル変数にアクセスします。このパラメーターは、**DB2\_ENABLE\_LDAP** レジストリー変数を YES に設定した場合のみ有効です。

**-all** 以下の場所に定義されているローカル環境変数をすべて表示します。

- オペレーティング・システム環境 ([e] で示されます)
- ノード・レベル・レジストリー ([n] で示されます)
- インスタンス・レベル・レジストリー ([i] で示されます)
- グローバル・レベル・レジストリー ([g] で示されます)

**-gd** *agg-registry-variable*

集合レジストリー変数のグループ定義を表示します。詳しくは、「データ・サーバー、データベースおよびデータベース・オブジェクトのガイド」で『集合レジストリー変数』を参照してください。

**-null** 指定したレジストリー・レベルまたはデフォルトのレベル (レジストリー・レベルを指定しなければ、インスタンス・レベルになる) の変数の値を NULL に設定します。つまり DB2 は、この変数が設定されていないと見なし、レジストリー・レベルの優先順位の定義に従って次に使用可能なレジストリー・レベルの変数の値を検索する操作を行いません。

**-r** 特定のインスタンスのプロファイル・レジストリーをリセットします。

*instance*

プロファイルをリセットするインスタンスを指定します。インスタンスを指定しない場合、インスタンス接続が存在していれば、このオプションでは現行インスタンスのプロファイルがリセットされます。インスタンスを指定せず、接続も存在しない場合、このオプションでは **DB2INSTANCE** 環境変数に指定されているインスタンスのプロファイルがリセットされます。

*db-partition-number* または *member-number*

db2nodes.cfg ファイル内の番号を指定します。

**-immediate** パラメーターを *member-number* オプションと一緒に使用すると、指定したメンバーでのみ値がリセットされます。

#### **-immediate | -im**

このフィーチャーをサポートするレジストリー変数に対する更新が、次回の SQL ステートメントのコンパイル時に有効になるように指定します。

SQL コンパイラーに影響を与えるレジストリー変数に対する即時変更は、次回の SQL ステートメントのコンパイル時に有効になります。以下の 2 つのタイプの SQL ステートメントがあります。

#### 動的 SQL ステートメント

動的 SQL ステートメントが既にパッケージ・キャッシュに入っている場合、そのステートメントは無効化されないため、新しい設定値による再コンパイルは行われません。 **-immediate** パラメーターを有効にするには、**FLUSH PACKAGE CACHE** ステートメントを実行して古いステートメントをパッケージ・キャッシュから削除し、インスタンスを再始動しなくても SQL ステートメントを再コンパイルできるようにする必要があります。

#### 静的 SQL ステートメント

静的 SQL ステートメントが既にパッケージに入っている場合、そのステートメントは無効化されないため、新しい設定値による再コンパイルは行われません。 **-immediate** パラメーターを有効にするには、**BIND** コマンドまたは **REBIND** コマンドを実行し、インスタンスを再始動しなくてもパッケージが再コンパイルされるようにする必要があります。

このパラメーターは、**-g**、**-gl**、**-n** のいずれのパラメーターとも組み合わせられません。

#### **-info variable**

指定された変数のプロパティを返します。このプロパティは、変数が即時変更をサポートするかどうかと、変更がデフォルトで即時有効かどうかを示します。

#### **-n DAS node**

リモートの DAS ノード名を指定します。

#### **-u user**

Administration Server へのアタッチを使用する際のユーザー ID を指定します。

#### **-p password**

Administration Server へのアタッチを使用する際のパスワードを指定します。

**-l** DB2 製品インストール済み環境のすべてのインスタンス・プロファイルをリストします。

**-lr** サポートされているレジストリー変数をすべてリストします。

**-v** コマンドの実行中に、冗長出力を使用することを指定します。

- ul ユーザー・プロファイル変数にアクセスします。このパラメーターは Windows オペレーティング・システム上だけでサポートされます。
- ur ユーザー・プロファイル変数を最新表示します。これによって、同じインスタンス設定または同じ環境設定で作業する複数のユーザーが、それぞれ異なる変数設定で作業できるようになります。このパラメーターは Windows オペレーティング・システム上だけでサポートされます。
- h | -? ヘルプ情報を表示します。このパラメーターを指定した場合、他のすべてのパラメーターは無視されます。

## 例

以下の例は、**db2set** コマンドにさまざまなパラメーターを指定して発行する方法を示しています。

- 特定のインストール済み環境に属する定義済みのインスタンス・プロファイルをすべて表示するには、次のように入力します。

```
db2set -l
```

- サポートされているレジストリー変数をすべて表示するには、次のように入力します。

```
db2set -lr
```

- 特定のインストール済み環境に属するすべてのインスタンスから可視であるすべての定義済みグローバル変数を表示するには、次のように入力します。

```
db2set -g
```

- 現行インスタンスの定義済み変数をすべて表示するには、次のように入力します。

```
db2set
```

- 現行インスタンス用に定義された値をすべて表示するには、次のように入力します。

```
db2set -all
```

- 現行インスタンスの **DB2COMM** レジストリー変数に定義された値をすべて表示するには、次のように入力します。

```
db2set -all DB2COMM
```

- メンバー 3 のインスタンス INST 用に定義された変数をすべてリセットするには、次のように入力します。

```
db2set -r -i INST 3
```

- ユーザー ID に MYID、パスワードに MYPASSWD を使用し、DAS ノード RMTDAS を介してリモート・インスタンス RMTINST の変数 **DB2CHKPTR** レジストリー変数の値を削除するには、次のように入力します。

```
db2set -i RMTINST -n RMTDAS -u MYID -p MYPASSWD DB2CHKPTR=
```

- 特定のインストール済み環境に属するすべてのインスタンスの **DB2COMM** レジストリー変数に TCP/IP を設定するには、次のように入力します。

```
db2set -g DB2COMM=TCP/IP
```

- インスタンス MYINST に対してのみ **DB2COMM** レジストリー変数に TCP/IP を設定するには、次のように入力します。

```
db2set -i MYINST DB2COMM=TCP/IP
```

## db2set - DB2 プロファイル・レジストリー

- デフォルトのレベルで **DB2COMM** レジストリー変数に **NULL** を設定するには、次のように入力します。デフォルトのレベルとは、インスタンス・レベルです。

```
db2set -null DB2COMM
```

- レジストリー変数 **DB2\_ANTIJOIN** の現行値を削除して、これが次の **SQL** ステートメントのコンパイル時に有効になるようにするには、次のように入力します。

```
db2set DB2_ANTIJOIN= -immediate
```

### 使用上の注意

4 つのレベル (オペレーティング・システム環境、ノード・インスタンス、インスタンス、およびグローバル) のうち、1 つ以上のレベルで変数を設定することができます。DB2 データベース・システムは、この優先順位を使用して変数設定にアクセスして解決します。既に変数の値をオペレーティング・システム環境レベルで設定している場合、その変数の値の更新は、**-immediate** パラメーターを指定しても即時に有効にはなりません。これは、オペレーティング・システム環境レベルが、各レジストリーよりも優先されるためです。

変数名を指定しない場合、定義済みのすべての変数の値が表示されます。変数名を指定した場合、その変数の値のみが表示されます。特定の変数の定義済みの値をすべて表示するには、*variable* と **-all** パラメーターを指定します。すべてのレジストリーにおける定義済み変数の値をすべて表示するには、**-all** パラメーターを指定します。

Windows オペレーティング・システムのリモート・レジストリーに対してレジストリー変数の変更を行うには、**db2set** コマンドとともに、**db2\_all** または **rah** コマンドを実行します。

このコマンドは、DB2 製品の root インストールと非 root インストールとで同じ動作をしますが、一部使用できないパラメーターがあります (例えば DAS ノード名を指定する **-n** パラメーターなど)。



## db2setup - DB2 データベース製品のインストール

DB2 データベース製品をインストールします。このコマンドは、Linux および UNIX のオペレーティング・システムでのみ利用できます。Windows オペレーティング・システム用のコマンドは **setup** です。

このユーティリティーは、DB2 データベース・インストール・メディアにあります。これを使って DB2 セットアップ・ウィザードを立ち上げ、インストールを定義して DB2 データベース製品をインストールします。 **-r** オプションを指定して呼び出すと、ユーザーの入力を必要としないインストールが実行されます。インストールの構成情報は応答ファイルから読み取られます。

### 許可

root 権限 Linux および UNIX オペレーティング・システムでは、root インストールには root ユーザー権限が必要です。非 root インストールの場合、非 root インストールを所有するユーザー ID でログオンしなければなりません。

### コマンド構文

```

▶▶ db2setup [-i language] [-l log_file] [-t trace_file]
▶ [-r response_file] [-c] [-f nobackup] [--?] [-h]

```

### コマンド・パラメーター

#### **-i** *language*

インストールを実行する言語の 2 文字から成る言語コード。

#### **-l** *log\_file*

名前を指定したファイルにログを書き込みます。root インストールの場合、デフォルトのログ・ファイルは /tmp/db2setup.log です。非 root インストールの場合、デフォルトのログ・ファイルは /tmp/db2setup\_userID.log です (*userID* は非 root インストールを所有するユーザー ID)。db2setup を使用して IBM Tivoli System Automation for Multiplatforms (SA MP) をインストールする場合、SA MP のインストール・ログ・ファイルは、DB2 データベース・ログ・ファイルと同じディレクトリーに入れられます。

#### **-t** *trace\_file*

インストール・トレース情報を含むファイルを生成します。

#### **-r** *response\_file*

使用される応答ファイルの絶対パスとファイル名。 **-c** パラメーターが指定されている場合、このパラメーターは必須です。

#### **-c**

インストールを実行せずに、応答ファイルの内容を妥当性検査します。妥当性検査の結果はログ・ファイルに出力されます。ログ・ファイルの場所は、実行後のメッセージに表示されます。このパラメーターを指定する場合は、 **-r** パラメーターも指定する必要があります。

## db2setup - DB2 データベース製品のインストール

### **-f nobackup**

これは非ルート・アップグレードのみに適用されます。コンポーネントの更新時に、**db2setup** に強制的にインストール・ファイルをバックアップさせないようにします。ファイルをバックアップしないことを選択すると、インストール・ディレクトリーのスペース所要量が削減されます。ただし、ファイルをバックアップしないことを選択すると、エラーが発生した場合、DB2 インストーラーがロールバック操作を実行できなくなります。この場合、ファイルを手動でクリーンアップし、製品を再インストールする必要があります。

### **-? | -h**

使用法情報を生成します。

## 使用上の注意

ユーザーは、適切な権限のある ID でログオンするか、**su** を "-" フラグとともに使用して (**su -**)、適切な権限のある ID でログオンしたかのように処理環境を設定する必要があります。処理環境が適切な権限のある ID に設定されていないと、インストール・プロセスはエラーなしで完了しますが、DB2 コピーを実行するときにエラーが発生します。

## db2snapcore - Linux および Solaris 用 DB2 snapcore

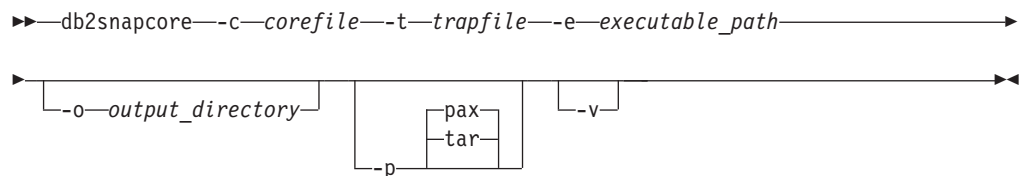
Linux および Solaris オペレーティング・システムでは、このコマンドは AIX オペレーティング・システムにおいて `snapcore` コマンドが提供する機能を模倣します。このコマンドは、EDU トラップ・ファイルから共有オブジェクト・リスト・セクションを抽出し、そのセクションをコア・ファイルと組み合わせて、出力をアーカイブに圧縮します。そのアーカイブを、IBM サポートに送信して分析してもらうことができます。

### 許可

以下の権限のいずれか。

- インスタンス所有者
- DBADM 権限

### コマンド構文



### コマンド・パラメーター

**-c** *corefile* | **-corefile** *corefile*

コア・ファイルの相対パスまたは絶対パスを指定します。

**-t** *trapfile* | **-trapfile** *trapfile*

トラップ・ファイルの相対パスまたは絶対パスを指定します。

**-e** *executable\_path* | **-executable** *executable\_path*

実行可能ファイルの相対パスまたは絶対パスを指定します。このファイルは、IBM サポートによって提供されるか、または IBM サポートのガイダンスに従ってトラップ・ファイルから入手できます。

**-o** *output\_directory* | **-outdir** *output\_directory*

圧縮出力ファイルの出力ディレクトリーを指定します。指定の出力ディレクトリーが存在していなければなりません。

**-p** | **-packCommand**

すべてのファイルをアーカイブに圧縮します。Linux オペレーティング・システムの場合、デフォルトで、ファイルは `pax` アーカイブに圧縮されます。Solaris オペレーティング・システムでは、`pax` オプションは使用できず、デフォルトで、ファイルは `tar` アーカイブに圧縮されます。

#### **pax**

ファイルを `pax` (ポータブル・アーカイブ交換) コマンドを使用して圧縮するように指定します。Solaris オペレーティング・システムの場合、`pax` オプションは使用できません。

## db2snapcore - Linux および Solaris 用 DB2 snapcore

### tar

ファイルを、**tar** (テープ・アーカイブ) コマンドを使用して圧縮するように指定します。**tar** コマンドがシステム上にインストールされている必要があります。

### -v | -verbose

冗長モードを使用可能にします。

### 例

EDU トラップ・ファイルとコア・ファイルから抽出された共有オブジェクト・リスト・セクションをアーカイブするには、次のようなコマンドを発行してください。必要に応じてコア・ファイルとトラップ・ファイルの名前を置き換えてください。

```
db2snapcore -corefile db2sysc.6270.98042.core -trapfile 28930.16.000.trap.txt -e ~/sql1lib/adm/db2sysc
```

使用するアーカイブ方式 (**pax** コマンドまたは **tar** コマンド) によって、結果として生じるアーカイブ・ファイル名は `db2snapcore.pax.gz` または `db2snapcore.tar.gz` のいずれかとなります。

## db2start - Start DB2

ターゲット・メンバーまたはすべてのメンバーで、データベース・マネージャーを開始します。DB2 pureScale 環境では、クラスター・キャッシング・ファシリティー (CF) を開始するために使用することもできます。

**db2start** は、システム・コマンドまたは CLP コマンドとして実行することができます。

データベースへの接続、アプリケーションのプリコンパイル、またはパッケージのデータベースへのバインドの前に、サーバーで DB2 を開始します。

**db2start** コマンドは、インストールされている DB2 データベース製品を Windows サービスとして起動します。ただし、**db2start** の呼び出し時に **/D** スイッチを指定すれば、Windows 上にインストールされている DB2 データベース製品をプロセスとして実行することもできます。また、「コントロール パネル」または **NET START** コマンドを使用して、インストールされている DB2 データベース製品をサービスとして開始することもできます。

**db2start** は Windows サービスを起動するので、ユーザーはサービス開始のための Windows 要件を満たす必要があります。拡張セキュリティが無効になっている場合、ユーザーは Administrators、Server Operators、または Power Users グループのメンバーになっている必要があります。拡張セキュリティが有効になっている場合、ユーザーはデータベースを始動するために Administrators グループまたは DB2ADMNS グループのメンバーになっている必要があります。

複数パーティション・データベースでの **db2start** 操作がデータベース・マネージャー構成パラメーター **start\_stop\_time** で指定される値以内に完了しない場合、タイムアウトになったデータベース・パーティションは、データベース・マネージャー・インスタンスのバックグラウンド・プロセスを開始しません (そのデータベース・パーティションに関連するリソースはすべて除去されます)。多くのデータベース・パーティションで **start\_stop\_timeout** に低い値が指定されている環境の場合、このような動作になる可能性があります。この動作による問題を解決するには、**start\_stop\_time** データベース・マネージャー構成パラメーターにさらに大きい値を指定してください。

root インストールの場合、DB2 はLinux およびUNIX オペレーティング・システムでコピーを行い、**db2start** コマンドはインスタンス所有者 ID の ulimit の永久設定を変更しないで、データベース・マネージャーが必要とされる ulimit 値を設定します。

非 root インストールの場合、'data' の ulimit を 'unlimited' に設定し、'nofiles' を 'unlimited' またはシステムで許可される最大値に設定する必要があります。

## db2stat - Windows の DB2 プロセス状況

Windows システムでは、すべてのインスタンスの下で実行されているすべての DB2 プロセスを、**db2stat** コマンドを使用して表示できます。

### 許可

なし

### 必要な接続

なし

### コマンド構文

▶▶—db2stat—◀◀

### コマンド・パラメーター

#### db2stat

すべてのアクティブな DB2 プロセスを出力します。

### 例

```
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

### 使用上の注意

Windows の場合の 1 つ注意事項は、DB2 はプロセス・ベースではなくスレッド・ベースなので、インスタンスのすべての EDU に対して 1 つのプロセス (DB2SYSCS) しか表示できないという点です。Windows では、Linux/UNIX システムの場合と同程度の情報を返せないことは明らかですが、実行中のすべてのインスタンスのプロセス ID を知りたい場合などには依然として役立ちます。例えば、Windows タスク マネージャというユーティリティを使用すると、指定のプロセス ID の CPU とメモリー使用量を判別できます。

---

## db2stop - DB2 の停止

ターゲット・メンバーまたはすべてのメンバーで、データベース・マネージャーを停止します。DB2 pureScale 環境では、クラスター・キャッシング・ファシリティー (CF) を停止するために使用することもできます。

**db2stop** は、システム・コマンドまたは CLP コマンドとして実行することができます。

複数パーティション・データベースでの **db2stop** 操作がデータベース・マネージャー構成パラメーター **start\_stop\_time** で指定される値以内に完了しない場合、タイムアウトになったデータベース・パーティションは内部で強制終了されます (そのデータベース・パーティションに関連するリソースはすべて除去されます)。多くのデータベース・パーティションで **start\_stop\_time** に低い値が指定されている環境の場合、このような動作になる可能性があります。この動作による問題を解決するには、**start\_stop\_time** にもっと大きい値を指定してください。

## db2support - 問題分析および環境収集ツール

クライアント・マシンまたはサーバー・マシンについての環境データを収集して、システム・データを含むファイルを圧縮ファイル・アーカイブに置きます。

DB2 インストール・イメージに同梱の **db2support** コマンドは、DB2 製品をインストールした後に使用できるコマンド・パラメーターのサブセットのみをサポートします。DB2 製品をインストールするまでは、使用できる **db2support** コマンド・パラメーターは、**-install** および **-host** パラメーターのみです。

このツールは、ユーザーとの対話式の質問と答えによって、問題の性質についての基本データを収集することもできます。

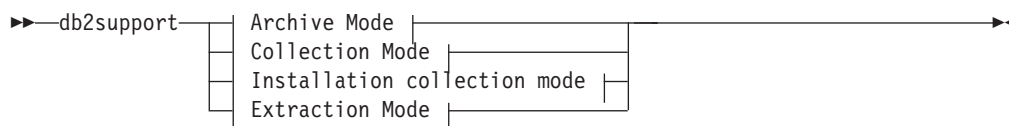
### 許可

最も完全な出力ができるように、このユーティリティーを **SYSADM** 権限で実行してください。システムに対するより限定された特権を持つユーザーはこのツールを実行できますが、データ収集アクションによっては、報告が少なくなったり出力が少なくなったりするものもあります。

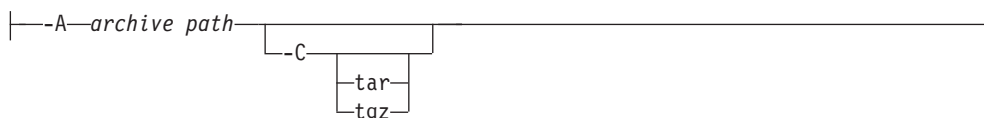
### 必要な接続

なし

### コマンド構文



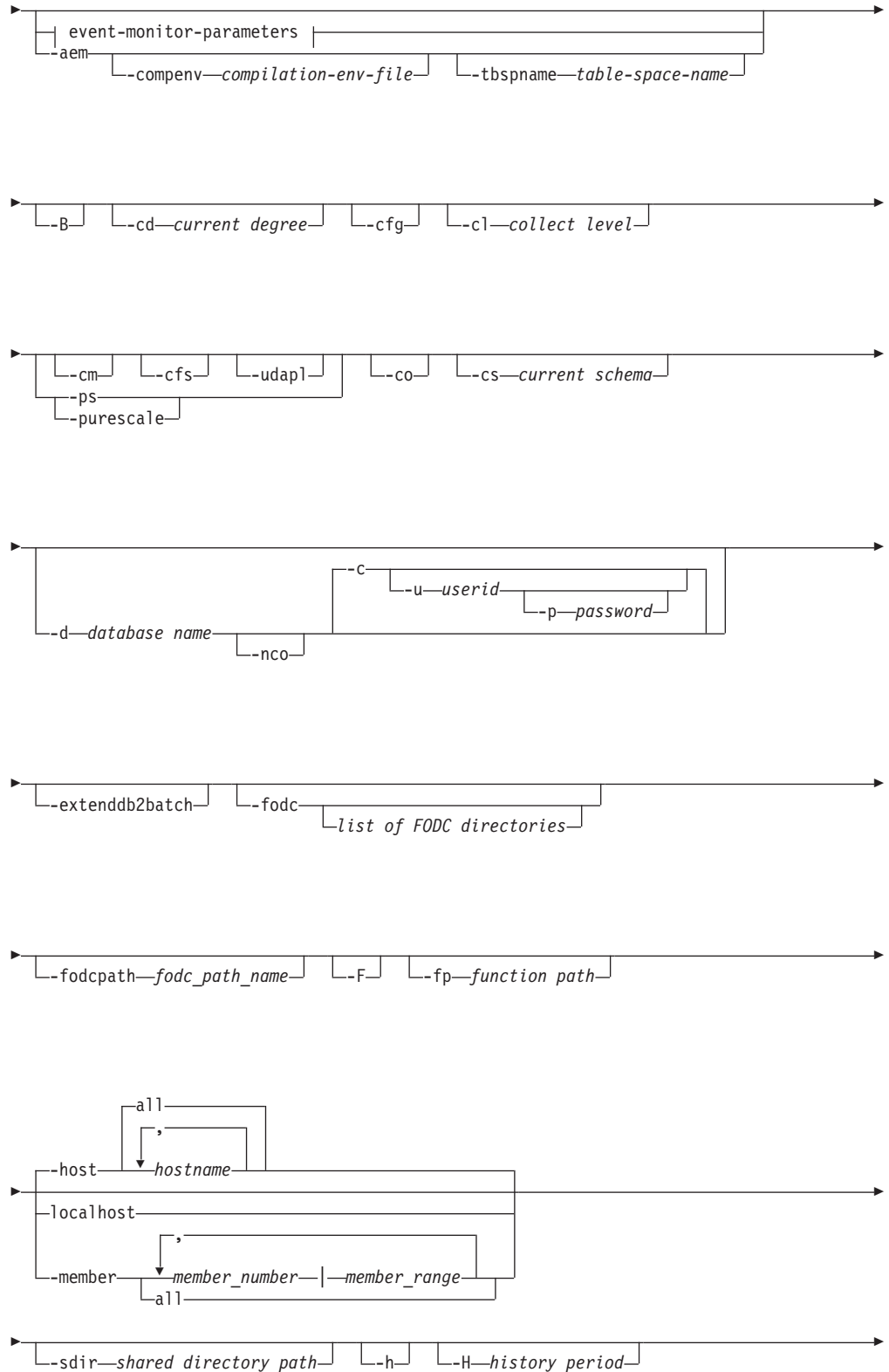
#### Archive Mode:



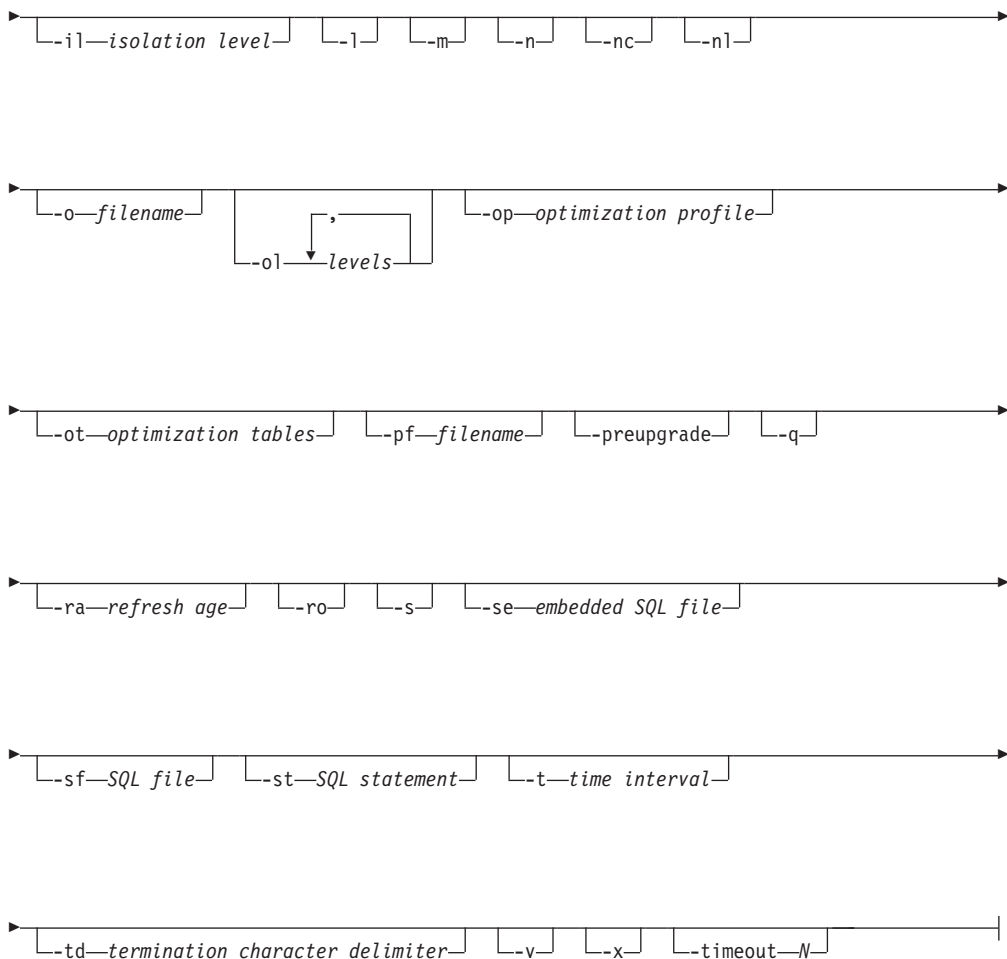
#### Collection Mode:



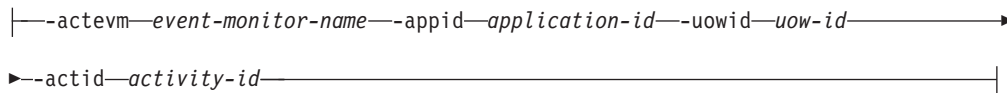




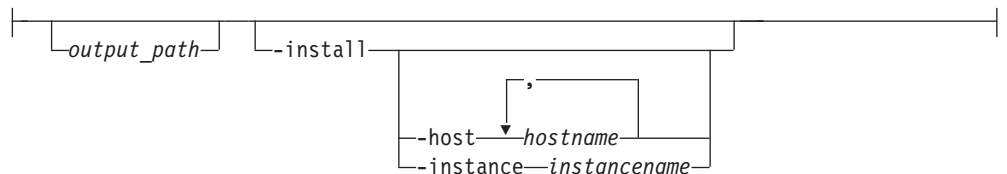
## db2support - 問題分析および環境収集ツール



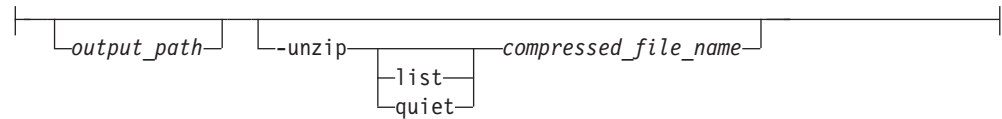
### event-monitor-parameters:



### Installation collection mode:



## Extraction Mode:



## 注:

1. **-st**、**-sf**、または **-se** パラメーターを指定した場合に限り、**db2support** ツールは不適切な照会に関連した情報を収集します。最適化中にエラーまたはトラップが発生する場合、不適切な照会を EXPLAIN せずにすべてのカタログ表と **db2look** 表の定義を収集するために、**-c1 0** (収集レベル・ゼロ) を使用してください。アクティビティー・イベント・モニター関連情報をオプティマイザー収集の一部として収集する必要がある場合、**-aem** (**-st** または **-sf** と共に) またはイベント・モニター・オプションを指定します。オプティマイザーの問題を処理するには、これらのオプションのいずれか 1 つを指定する必要があります。

注: オプティマイザーの収集に関するオプションはすべて、**-d <dbname>** オプションを使用して実行する必要があります。

2. ステートメント実行時に特殊レジスターがデフォルト以外の値に設定されている場合、正しい問題分析を行うために、これらの値を **db2support** コマンドに渡してください。

注: 特殊レジスター・オプションは、**db2caem** 収集では無視されます。

3. **-global** パラメーターは廃止されました。 **-host all** オプションが **db2support** コマンドのデフォルト動作であるため、すべてのホストからの情報がデフォルトで収集されます。
4. **db2support** ツールは、デフォルトで DB2 pureScale コンポーネントに固有のほとんどの診断データを収集するため実行に時間がかかります。  
**-purescale**、**-cm**、**-cfs**、**-udapl** のいずれかのパラメーターを指定した場合、**db2support** ツールは追加の診断データを収集します。この診断データのためにスペースが多く消費されたり、収集に時間がさらにかかったりしますが、DB2 pureScale 環境における問題判別プロセスの速度を上げるのに役立つ可能性があります。

## コマンド・パラメーター

*output path*

圧縮されたアーカイブ・ファイルを作成または解凍するパスを指定します。このパスは、アーカイブに組み込むために、ユーザー作成ファイルを配置する必要のあるディレクトリー、または **-unzip** パラメーターを指定した際にファイルの解凍先となるディレクトリーです。このパラメーターが指定されていないときは、現行ディレクトリーが使用されます。

**-A** *archive\_path* | **-archive** *archive\_path*

このパラメーターで、**diagpath** 構成パラメーターに指定されたディレクトリーからのすべてのデータを、ここで指定されたアーカイブ・パスにアーカイブします。新規ディレクトリーは、DB2DUMP にシステム・ホスト名とタイム・スタンプが付加された名前、指定されたアーカイブ・パスで作成されます。例えば、**DB2DUMP\_systemhostname\_2009-01-12-12.01.01** となります。

## db2support - 問題分析および環境収集ツール

このパラメーターで、**alt\_diagpath** 構成パラメーターに指定されたディレクトリーからのすべてのデータも、ここで指定されたアーカイブ・パスにアーカイブします。このディレクトリーの名前は **ALT\_DB2DUMP** です。さらに、**events/** サブディレクトリーにあるファイルは **ALT\_EVENTS** ディレクトリーにアーカイブされ、**stmmlog/** サブディレクトリーにあるファイルは **ALT\_STMM** ディレクトリーにアーカイブされます。

このパラメーターは Windows オペレーティング・システムでは使用できません。

### -aem

**-st** または **-sf** で指定された SQL ステートメント用の **db2caem** 情報を収集することを指定します。**-aem** は **-se** をサポートしていません。**db2support** のデフォルト動作では、**db2caem** 情報は収集されません。このパラメーターを指定すると、**db2caem** は要求情報を収集するためにアクティビティー・イベント・モニターを作成します。アクティビティー・イベント・モニターおよび作成された他の表は、システムからクリーンアップされます。

注: **db2support** 特殊レジスター・オプションは、**db2caem** ツールでは無視されます。**-compenv** を使用して、**db2caem** が使用するコンパイル環境を指定しなければなりません。指定された SQL ステートメントは **db2caem** ツールで実行されます。

### -B | -basic

収集をオプティマイザー情報のみに制限します。**db2supp\_opt.zip** ファイルに関する情報を除いて、他の情報は収集されません。**-basic** パラメーターは、**-st**、**-sf**、または **-se** パラメーターと共に使用する必要があります。そうしない場合、構文エラーが戻されます。

### -c | -connect

指定のデータベースに接続します。

### -cd | -curdegree

使用する **CURRENT DEGREE** 特殊レジスターの値を指定します。デフォルトは、**dft\_degree** データベース構成パラメーターの値です。

### -cfg

構成情報を収集し、その他のサポート関連データをすべて除外します。このパラメーターは、以下のパラメーターとのみ組み合わせることが可能です。

**-c**、**-connect**、**-d**、**-database**、**-m**、**-html**、**-n**、**-number**、**-o**、**-output**、**-p**、**-password**、**-u**、**-user**、**-v**、**-verbose**

注: このコマンド・パラメーターは、DB2 バージョン 9.7 フィックスパック 2 以降で使用できます。

### -cfs

クラスター・ファイル・システムの追加診断データが、生成される **.zip** ファイルにパッケージされるように指定します。このパラメーターは追加のクラスター・ファイル・システム・データのみを収集しますが、これにはスペースが大量に消費されたり時間がかかったりします。

### -cl | -collect

戻されるパフォーマンス情報のレベルを指定します。有効な値は以下のとおりです。

0 = collect only catalogs, **db2look**, **dbcfg**, **dbmcfg**, **db2set**  
 1 = collect 0 plus **exfmt**  
 2 = collect 1 plus **.db2service** (this is the default)  
 3 = collect 2 plus **db2batch**

注: イベント・モニター・パラメーター (**-actevm**、**-appid**、**-uowid**、**-actid**) を **-st**、**-sf**、**-se** なしで指定する場合の効率的な収集レベルは 1 で、**db2caem** 情報だけが収集され、**db2exfmt** は収集されません。

**-cm**

クラスター・マネージャーの追加診断データが、生成される .zip ファイルにパッケージされるように指定します。このパラメーターは追加のクラスター・マネージャー・データのみを収集しますが、これにはスペースが大量に消費されたり時間がかかったりします。

**-co**

データベース内のすべての表のカタログを収集します。デフォルトでは、問題のある照会で使用される表のカタログ情報のみを収集します。

**-compnv compilation-environment-file**

**db2caem** コマンドの実行時に使用されるコンパイル環境の名前が含まれているファイルの名前を指定します。コンパイル環境 (**comp\_env\_desc**) は BLOB データ・タイプであり、入力としてファイルにより指定します。パラメーターが指定されない場合、**db2caem** を実行する際にはデフォルトのコンパイル環境が使用されます。

**-cs | -curschema**

ステートメント内で修飾されていない表名を修飾するために使われる現行スキーマを指定します。デフォルト値は、現行セッション・ユーザーの許可 ID です。

**-C | -compress**

アーカイブ圧縮を使用可能にします。デフォルトで、アーカイブ・データは単一ファイルに圧縮されます。アーカイブ圧縮は、アーカイブ・モードでのみ使用可能であるため、**-A** パラメーターと一緒に指定する必要があります。そうしない場合、構文エラーが返されます。

**tar**

ファイルを **tar** (テープ・アーカイブ) コマンドを使用してアーカイブするように指定します。 **tar** パラメーターは、UNIX および Linux オペレーティング・システムでサポートされます。

**tgz**

ファイルを **tar** コマンドを使用してアーカイブし、**gzip** コマンドを使用して圧縮するように指定します。 **tgz** パラメーターは、UNIX および Linux オペレーティング・システムでサポートされます。

**-d database\_name | -database database\_name**

データが収集されているデータベースの名前を指定します。

注: デフォルトでは、指定のデータベースに接続するための試行を行います。この動作をオーバーライドするには、**-noconnect** または **-nco** パラメーターを指定します。

### **-nco | -noconnect**

指定のデータベースへの接続を試みないことを指定します。

### **-c | -connect**

指定のデータベースへの接続を試みることを指定します。

注: データベースを指定すると、このコマンド・パラメーターがデフォルトで組み込まれます。

### **event-monitor-parameters**

次のパラメーターは、アクティビティー・イベント・モニター・データが収集される SQL ステートメントを一意的に示します。それらは一緒に指定する必要があります。

#### **-actevm** *activity-event-monitor-name*

*activitystmt* 論理グループ化が収集対象データを含む既存のアクティビティー・イベント・モニターの名前を指定します。

#### **-appid** *application-id*

収集対象アクティビティーを発行したアプリケーションを一意的に識別するアプリケーション ID (**appl\_id** モニター・エレメント) を指定します。

#### **-uowid** *uow-id*

収集対象データのある作業単位 ID (**uow\_id** モニター・エレメント) を指定します。作業単位 ID は、指定されたアプリケーション内のみで固有です。

#### **-actid** *activity-id*

収集対象データのあるアクティビティー ID (**activity\_id** モニター・エレメント) を指定します。アクティビティー ID は、指定された作業単位内のみで固有です。

### **-extenddb2batch**

**-o1** または **-optlevel** パラメーターで指定されたすべての最適化レベルに関する **db2batch** 情報をキャプチャーするように指定します。 **-extenddb2batch** パラメーターとともに、**-o1** パラメーターの値を少なくとも 1 つ指定し、**-c1** パラメーター値を 3 に設定する必要があります。 そうしない場合、**db2support** コマンドは構文エラーを戻します。

### **-fodc**

FODC ディレクトリーと **db2diag** ログ・ファイルのみを収集することを指定します。ディレクトリーを指定しない場合は、**db2support** コマンドによってすべての FODC ディレクトリーのリストが表示され、そこから選択できます。ディレクトリーは使用時のタイム・スタンプに基づいて昇順の日時順にリストされるので、使用された日時が最近のディレクトリーほど目につきやすい位置に表示されます。

**db2support** コマンドが収集できる FODC ディレクトリーは、コマンドが実行された物理データベース・ホスト上にあるものだけです。したがって、**-global** パラメーターは、**-fodc** と連動しません。

時間間隔 (**-t** または **-time**) あるいは履歴 (**-H** または **-history**) パラメーターを指定できますが、指定した FODC ディレクトリーが指定時間フレームの外側だった場合、**db2support** はエラーを返します。

**-fodc** を使用する場合は、**archive (-A または -archive)** および **basic (-B または -basic)** パラメーターも指定できません。

*Trap | Panic | BadPage | Hang | IndexError | Perf | DBMarkedBad*  
 収集する FODC ディレクトリーのカテゴリーを指定します。

*list of FODC directories*

既存の FODC ディレクトリーのコンマ区切りリスト。

**-fodcpath** *fodc\_path\_name*

**db2support** コマンドによる FODC パッケージ検索の対象にできる既存ディレクトリーの絶対パス名を指定します。 **db2support** コマンドは、FODC パッケージを収集するために以下のパスを検索します。

- データベース・マネージャー構成パラメーターの **diagpath** および **alt\_diagpath** で指定された診断データ・ディレクトリー
- **DB2FODC** レジストリー変数のインスタンス・レベル **FODCPATH** 設定
- そのマシンの各メンバーの **FODCPATH** 設定
- メモリー内の **db2pdcfg** 設定
- **db2support -fodcpath** コマンドを使用して指定される *fodc\_path\_name*

**-F | -full**

**db2support** 情報およびオプティマイザー固有の情報をすべて、何も除外せずにキャプチャーすることを指定します。

**-fp | -funcpath**

修飾されていないユーザー定義関数およびタイプを解決するために使う関数パス特殊レジスターの値を指定します。デフォルト値は、「SYSIBM」、「SYSFUN」、「SYSPROC」、*X* です (*X* は二重引用符で区切った **USER** 特殊レジスターの値)。

**-h | -help**

ヘルプ情報を表示します。このパラメーターを指定すると、他のすべてのパラメーターは無視され、ヘルプ情報だけが表示されます。

**-H** *history\_period* | **-history** *history\_period*

収集されるデータを特定の時間間隔に制限します。 *history\_period* 変数を指定する必要があります。 *history\_period* 変数は、数値と時刻タイプで指定します。オプションで、コロンで区切った開始時刻の値を指定できます。使用可能なタイプは以下のとおりです。

d = 日  
 h = 時  
 m = 分  
 s = 秒

開始時刻の値は、タイム・スタンプ形式で指定します。タイム・スタンプ形式は、*YYYY-MM-DD-hh.mm.ss.nnnnnn* です。 *YYYY* には年、 *MM* には月 (01 から 12)、 *DD* には日 (01 から 31)、 *hh* には時 (00 から 23)、 *mm* には分 (00 から 59)、 *ss* には秒 (00 から 59)、 *nnnnnn* には UNIX オペレーティング・システムではマイクロ秒、Windows オペレーティング・システムではミリ秒を指定します。年フィールドに続くフィールドの一部またはすべてを省略できます。省略されたフィールドには、デフォルト値が使用されます。月および日のデフォルト値は 1 であり、他のすべてのフィールドのデフォルト値は 0 です。



数値と時刻タイプには、+ 符号または - 符号を使用して正または負を指定できます。数値と時刻タイプのみ指定された場合、デフォルトは負になります。数値と時刻タイプが指定され、かつ開始時刻の値が指定された場合、デフォルトは正になります。例えば、`-history 6d` は、過去 6 日間のデータを収集することになります。`-history 6d:2009` は、2009 年の最初の 6 日間のデータを収集します。

このパラメーターは、`-time` または `-t` パラメーターと共に使用できません。

### **-host**

コマンド発行先ホスト (複数も可) を指定します。このパラメーターを指定しない場合、コマンドはデフォルトですべてのホストに対して発行されます。

### **all**

コマンドが、すべてのホストに対して発行されるように指定します。この設定は、**db2support** コマンドのデフォルトの動作です。

### *hostname*

コマンド発行先ホスト (複数も可) を指定します。このオプションを指定しない場合、コマンドはすべてのホストに対して発行されます。複数のホストを指定する場合、すべてのホスト名が有効でないと、コマンドを完了できません。

DB2 pureScale 環境でもパーティション・データベース環境でもない環境で

**-host** オプションを指定すると、**db2support** はエラーを返します。

### **-il | -isolation**

データ・アクセス中にデータをロックして他のプロセスから分離する方法を決定するために使われる分離レベルを指定します。デフォルトでは、CURRENT ISOLATION 特殊レジスターがブランクに設定されます。

### **-install**

DB2 インストール・プロセスまたはインスタンス作成に関する問題のトラブルシューティングに必要な診断データを収集します。診断データは、`db2support.zip` ファイルに保管されます。`db2support.exe` ファイルをローカル・システムにコピーします。このようにすれば、**db2support** コマンドを発行したときに、`db2support.exe` のコピー先ディレクトリーに確実に `db2support.zip` ファイルが配置されるようになります。診断データを可能な限り多く収集するためには、root 権限でコマンドを実行してください。また、**-install** パラメーターと一緒に `output_path` 変数を指定して、`db2support.zip` ファイルの出力パスを示すこともお勧めします。

### **-host hostname | -host hostname\_list**

診断データ収集の対象ホストを指定します。リモート・ホストのデータ収集の場合は、SSH 接続が必要です。ホスト名を指定しない場合、ローカル・ホストの診断データが収集されます。

複数のホストの診断データを収集する場合は、**-host** パラメーターの後に `hostname_list` を指定します。`hostname_list` は、診断データを収集する各ホストのコンマ区切りのリストです。

### **-instance instancename**

診断データを収集するインスタンス名を指定します。このパラメーターを指定しない場合、デフォルトでは、**DB2INSTANCE** 環境変数に定義されている



インスタンスに関する診断データが収集されます。特定のインスタンスに関する診断データを収集するには、**-instance** パラメーターの後に *instancename* を指定します。*instancename* は、診断データを収集するインスタンスの名前です。

#### **-l | -logs**

アクティブ・ログをキャプチャーすることを指定します。

#### **-localhost**

コマンドが、ローカル・ホストに対して発行されるように指定します。このオプションを指定しない場合、コマンドはすべてのホストに対して発行されます。

#### **-m | -html**

すべてのシステム出力が、HTML 形式のファイルにダンプされることを指定します。デフォルトでは、すべてのシステム関連情報は、このパラメーターが使用されない場合はフラット・テキスト・ファイルにダンプされます。

#### **-member member\_number | member\_range**

コマンド発行先メンバー (複数も可) を指定します。このパラメーターを指定しない場合、コマンドは現行メンバーに対して発行されます。複数のメンバーを、*member\_number* のコンマ区切りリスト (*member1*, *member2*) として、または *member\_range* (*member\_range* はメンバーの範囲 (*member1*-*member3*)) を使用して指定できます。最初の 2 つの方式を組み合わせることもできます。

#### **all**

これを指定すると、コマンドは、*db2nodes.cfg* で定義されたすべてのメンバーに対して発行されます。

**db2support** ツールは、ホストごとに実行します。1 つのホストに複数のメンバーが存在する場合、**db2support** は対象のホストで一度だけ実行します。コマンドを実行するメンバーを指定すると、**db2support** を実行するホストを判別するためだけにその指定のメンバー番号が使用されます。

DB2 pureScale 環境でもパーティション・データベース環境でもない環境で

**-member** オプションを指定すると、**db2support** はエラーを返します。

#### **-n | -number**

現在の問題の問題管理レポート (PMR) 番号または ID を指定します。

#### **-nc | -nocatalog**

カタログ情報が収集されないことを指定します。デフォルトでは、カタログ情報は収集されます。

#### **-n1 | -nodb2look**

**db2look** コマンド情報が収集されないことを指定します。デフォルトでは、**db2support** コマンド情報は収集されます。

#### **-o filename**

**db2support** コマンドの発行後に生成される圧縮ファイルの新規名を指定します。新規ファイル名を指定するか、絶対パスか相対パスを記述することができます。ただし、このパラメーターを指定する前にこのパスが存在し、アクセス可能になっていることが必要です。そうでなければエラーが発生します。

このパラメーターが使用されない場合、生成されるデフォルトの圧縮ファイルは *db2support.zip* です。

このパラメーターを *output path* パラメーターと一緒に使用すると、*output path* で指定されたパスは無視されて、**-o** パラメーターで指定されたパスが使用されます。

### **-ol levels | -optlevel levels**

使用する最適化レベル特殊レジスターの値を指定します。デフォルトは、**dft\_queryopt** データベース構成パラメーターの値です。最適化レベルの値を単一値またはコンマで区切った複数の値として指定できます。

複数の値を指定した場合、最初の値ですべての最適化情報が収集されます。それぞれの追加の最適化レベルの値を指定した場合、**Explain** プランは、各レベルの収集の開始および終了時刻に従って、別個のファイルに収集され、保管されません。

### **-op | -optprofile**

使用する最適化プロファイル特殊レジスターの値を指定します。ステートメントのバインド時に有効な最適化プロファイルが存在した場合に限って、この値が必要です。デフォルトは "" (空ストリング) です。

### **-ot | -opttables**

動的 SQL 照会の処理を最適化する際に考慮される表のタイプを識別するために使われる、「CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION」という特殊レジスターの値を指定します。CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION の初期値は「SYSTEM」です。

### **-p password | -password password**

ユーザー ID のパスワードを指定します。

### **-ps | -purescale**

追加の DB2 診断データ、およびクラスター・ファイル・システム、クラスター・マネージャー、uDAPL の追加の診断データを収集するように指定します。このパラメーターは追加の診断データのみを収集しますが、これにはスペースが大量に消費されたり時間がかかったりします。このパラメーターを指定するのは、**-cm**、**-cfs**、および **-udapl** パラメーターを指定するのと同様です。

### **-pf filename | -profile filename**

代替プロファイル・ファイルを指定します。このパラメーターを使用する場合、この新規プロファイル・ファイルの正確な場所を取得するため、絶対パスを指定する必要があります。

デフォルト・プロファイルは `db2support_collection.profile` ファイルであり、`sqllib/adm` フォルダに格納されます。このプロファイル・ファイルを使用して、標準の **db2support** コマンド実行には組み込まれない追加の情報を収集します。以下に示すのは、任意の数のテンプレート 1 とテンプレート 2 の任意の組み合わせから成るプロファイル・ファイルの、可能なテンプレートです。

#### テンプレート 1

```
<COLLECTION>
<NAME>...</NAME>
<CMD>...</CMD>
<OUTFILE>...</OUTFILE>
<TIMEOUT>...</TIMEOUT>
</COLLECTION>
```

#### テンプレート 2

```
<COLLECTION>
<NAME>...</NAME>
<FILE>...</FILE>
<OUTFILE>...</OUTFILE>
<TIMEOUT>...</TIMEOUT>
</COLLECTION>
```

各コレクション項目は、名前、コマンドまたはファイル、出力ファイル、およびタイムアウト値によって記述されます。この情報は、追加情報を収集するために使用されます。各コレクション項目は、必ず **CMD** または **FILE** のいずれかの値を持ちますが、両方を持つことはありません。このファイルが解析される際には、すべてのブランク行はスキップされます。各コレクション項目に入っている値について、以下で説明します。

**NAME**

収集されるデータの名前。

**CMD**

追加情報を収集するためにコマンド行で使用されるコマンド。これは必須の値です。

**FILE**

収集するファイルの名前。これは必須の値です。

**OUTFILE**

収集される情報が格納される出力ファイルの名前。これは必須の値です。

**TIMEOUT**

コマンド行の実行で許可される最長時間 (秒単位)。デフォルト値は 180 秒です。

**OUTFILE** 値および **CMD/FILE** 値が欠落していると、**db2support** がプロファイル・ファイルを構文解析するときに構文解析エラーが発生します。これが発生すると、**db2support** はこの収集をスキップします。プロファイル・ファイルの構文解析が終了すると、収集された情報は **db2supp\_system.zip** ファイルの **USERCOLLECTION/OUTFILE** に格納されます。

**-preupgrade**

インスタンスのアップグレードや次のフィックスパックへの更新などの、重要なアップグレードまたは更新の前に、環境データと構成データを収集します。このパラメーターは、アップグレードや更新後に生じる場合がある問題のトラブルシューティングに役立ちます。このパラメーターと併用できるパラメーターは、**-d**、**-o**、**-fodcpath**、**-nl**、**-member**、および **-host** のみです。

データ収集が完了すると、結果は **db2support\_preupgrade.zip** というファイルに圧縮されます。

**-q | -question\_response**

対話式問題分析モードを使用することを指定します。

**-ra | -refreshage**

リフレッシュ経過時間特殊レジスターの値を指定します。ステートメント内の表を参照するマテリアライズ照会表 (MQT) が存在する場合にのみ、この値が該当します。CURRENT REFRESH AGE のデフォルト値はゼロです。

### **-ro | -reopt**

照会の EXPLAIN 時に、REOPT ONCE オプションを指定して EXPLAIN を使用するようにします。デフォルトでは、REOPT ONCE オプションを無視します。

### **-s | -system\_detail**

ハードウェアおよびオペレーティング・システムの詳細情報を収集することを指定します。

### **-se embedded SQL file | -sqlembed embedded SQL file**

データ収集に関連した SQL ステートメントが入っている組み込み SQL ファイルのパスを指定します。

### **-sdir shared directory path | -S shared directory path**

**db2support** 収集時に一時記憶域として使用される共有ディレクトリーを指定します。このオプションを指定しないと、データを一時的に格納するためにデフォルトの共有ディレクトリーが使用されます。デフォルトの共有ディレクトリーは、DB2 pureScale 環境の場合には `<db2 instance shared directory>/sqllib_shared` で、パーティション・データベース環境の場合には `path/sqllib` です。一時的に格納されたデータは、**db2support** コマンドの実行が完了すると削除されます。

注: DB2 pureScale 環境では `sqllib_shared` はデフォルトの共有ディレクトリーとして使用されるので、**db2support** 収集時にデータを格納する十分なスペースを `sqllib_shared` ディレクトリーに確保してください。 `sqllib_shared` ディレクトリーに必要な最小ディスク・スペースを計算するには、以下の公式を使用できます。

$$10GB + \text{number of member/CF host} * 2GB$$

### **-sf SQL file | -sqlfile SQL file**

データ収集に関連した SQL ステートメントが入っているファイル・パスを指定します。

### **-st SQL statement | -sqlstmt SQL statement**

データ収集に関連した SQL ステートメントを指定します。

### **-t time\_interval | -time time\_interval**

**db2support** によって収集されるデータを、`time_interval` 変数によって指定された特定の時間間隔に制限します。時間間隔は、開始時刻、終了時刻、またはその両方をコロンによって区切られたタイム・スタンプ形式で指定することができます。タイム・スタンプ形式は、`YYYY-MM-DD-hh.mm.ss.nnnnnn` です。YYYY には年、MM には月 (01 から 12)、DD には日 (01 から 31)、hh には時 (00 から 23)、mm には分 (00 から 59)、ss には秒 (00 から 59)、nnnnnn には UNIX オペレーティング・システムではマイクロ秒、Windows オペレーティング・システムではミリ秒を指定します。年フィールドに続くフィールドのいくつかまたはすべては、省略できます。省略されたフィールドには、デフォルト値が使用されます。月および日のデフォルト値は 1 であり、他のすべてのフィールドのデフォルト値は 0 です。

開始時刻のみを指定した場合 (-t 2009)、**db2support** コマンドは開始時刻よりも後に変更されたファイルを収集します。終了時刻のみを指定した場合 (-t :2009)、**db2support** コマンドは終了時刻よりも前に変更されたファイルを収集します。両方を入力して指定した場合 (-t 2008:2009)、**db2support** コマンドは

開始時刻と終了時刻の間に変更されたファイルを収集します。このパラメーターにデフォルト値はありません。少なくとも 1 つのタイム・スタンプを指定する必要があります。

このパラメーターは、**-history** または **-H** パラメーターと共に使用できません。

#### **-tbspname table-space-name**

**db2caem** コマンドがアクティビティ・イベント・モニターを作成する表スペース名を指定します。パーティション・データベース環境では、対象となる SQL ステートメントを実行するすべてのデータベース・パーティションにこの表スペースが存在しなければなりません。オプションが指定されない場合、アクティビティ・イベント・モニターを作成する際に **db2caem** コマンドはデフォルトの表スペースを使用します。

#### **-td | -delimiter**

ステートメントの終了文字を指定します。このコマンド・パラメーターの機能は、**db2** コマンドの **-td** パラメーターと同じです。デフォルトはセミコロンです。

#### **-timeout N**

**db2support** ツールが実行を停止するまでのタイムアウト期間を秒単位で指定します。 *N* 変数は、秒単位で指定する必要があります。 **timeout** には、実行開始後の合計実行時間を指定します。個別の収集のタイムアウトではありません。タイムアウトが発生すると、**db2support.zip** ファイルが作成され、エラー・メッセージが画面および **db2support.log** ファイルに書き込まれます。

このパラメーターは、他のどのパラメーターとも一緒に使用できます。

#### **-u userid | -user userid**

データベースへの接続に使用するユーザー ID を指定します。

#### **-udapl**

uDAPL の診断データが、生成される **.zip** ファイルにパッケージされるように指定します。このパラメーターは追加の uDAPL データのみを収集し、これにはスペースが大量に消費されたり時間がかかったりします。

#### **-unzip compressed\_file\_name**

指定した圧縮ファイルから内容を抽出します。 *compressed\_file\_name* には絶対パスまたは相対パスを指定する必要があります。このパラメーターは、抽出ユーティリティーを使用できないシステム・ファイル上の **db2support.zip** ファイルを抽出できます。 **-unzip** パラメーターは、他の **db2support** モードのパラメーターと組み合わせて使用することはできません。また、**db2support -unzip** コマンドは、 *compressed\_file\_name* 変数で **.ZIP** または **.zip** ファイル拡張子を付けて指定されたファイルのみを識別します。

*output\_path* 変数を **-unzip** パラメーターと一緒に指定すると、解凍されたファイルは *output\_path* ディレクトリーに配置されます。 **-unzip** パラメーターに *output\_path* 変数を指定しないと、現行ディレクトリーの下に *compressed\_file\_name* という新しいディレクトリーがデフォルトで作成され、この *compressed\_file\_name* ディレクトリー内に解凍されたファイルが配置されます。

### list

これを指定すると、圧縮ファイルの内容は標準出力にリストされますが、抽出されません。ファイル名、サイズ、および日付が表示されます。このパラメーターは、db2support.zip が大容量で、システムで使用可能なスペースがあまりない場合に役立ちます。

### quiet

**db2support** コマンドにおいて、抽出したファイルが *output\_path* (*output\_path* が発行されなかった場合には現行ディレクトリー) に既に存在する場合に入力を求めるプロンプトが出されるのを防ぎます。 **quiet** パラメーターを発行しなかった場合、指定ファイルを上書きするかどうかを尋ねるメッセージが出されます。このパラメーターを指定すると、プロンプトを出さずにすべての既存のファイルを上書きします。

### -v | -verbose

このツールの実行中に、冗長出力を使用することを指定します。

### -x | -xml\_generate

対話式問題分析モード (-q モード) 中に使用される判断ツリー論理全体を含む XML 文書を生成することを指定します。

## 例

### 例 1

**db2support** ツールは、以下のいずれかの方法によってオブティマイザー・モードで呼び出されます。

- コマンド行から SQL ステートメントとして。

```
db2support output_directory -d database_name -st sql_statement
```

**db2support** ツールは、bad\_query.sql というファイルに照会をコピーすることによって、オブティマイザー・ディレクトリーに照会を保管します。

- ファイル内に保管された SQL ステートメントとして。

```
db2support output_directory -d database_name -sf sql_file
```

このツールは、照会が入っているファイルをオブティマイザー・ディレクトリーにコピーします。

- 問題のある照会を含む組み込み静的 SQL ステートメントが入っているファイルとして。

```
db2support output_directory -d database_name -se embedded_sql_file
```

このツールは、照会が入っているファイルをオブティマイザー・ディレクトリーにコピーします。ファイルは現行ディレクトリーに存在する必要はありませんが、呼び出し元のユーザー ID がファイルを読み取ることができなければなりません。

- さまざまなレベルのパフォーマンス情報を戻すときに。

```
db2support output_directory -d database_name -collect 0
```

**db2support** ツールは、要求された詳細レベルに基づいてさまざまなレベルのパフォーマンス情報を収集します。0 から 3 まで値が増えるにつれて、よ



り多くの詳細が収集されます。レベル 0 を使用すると、実動データベース用にデータベース・オブジェクトを複製できるようにするカタログ情報と表定義が収集されます。

## 例 2

デフォルトで設定されたオプティマイザー関連特殊レジスターを使って動作の違いを診断するための情報を収集するには、以下を使用します。

```
db2support . -d sample -st "SELECT * FROM EMPLOYEE"
```

この例では、すべてのデータが db2support.zip ファイルに戻されます。(出力パスとして . が指定されたため) 現行ディレクトリーとサブディレクトリーの中に診断ファイルが作成されます。システム情報、オプティマイザー情報、および診断ファイルもまた収集されます。

## 例 3

オプティマイザー関連特殊レジスターにユーザー指定値を使用して前の例と同じ情報を収集するには、以下を使用します。

```
db2support . -d sample -st "SELECT * FROM EMPLOYEE" -cs db2usr -cd 3
-o1 5 -ra ANY -fp MYSCHEMA -op MYPROFSHEMA.MYPROFILE -ot ALL -il CS
```

## 例 4

オプティマイザー関連特殊レジスターに複数のユーザー指定値を使用して前の例と同じ情報を収集し、各オプティマイザー特殊レジスター値の db2batch 情報を収集するには、以下を使用します。

```
db2support . -d sample -st "SELECT * FROM EMPLOYEE" -cs db2usr -cd 3
-o1 3,5,7 -cl 3 -extenddb2batch -ra ANY -fp MYSCHEMA -op MYPROFSHEMA.MYPROFILE -ot ALL -il CS
```

この例では、特殊レジスターとして、現行スキーマが db2usr に、CURRENT DEGREE が 3 に、最適化レベルが 5 に、リフレッシュ経過時間が ANY に、関数パスがスキーマ MYSCHEMA に、最適化プロファイルが MYPROFSHEMA.MYPROFILE に、現在保守されている表タイプが ALL に、分離レベルが CS にそれぞれ設定されます。これらの値は、指定したデータベースとの間で db2support が確立する接続に関してのみ設定されます。ご使用の環境全体に影響することはありません。診断によって問題を修正する際、照会が実行されたときと同じ特殊レジストリー変数を使用することが重要です。

## 例 5

現在時刻から 3 日前の間に変更されたファイルにデータ収集を制限するには、以下を使用します。

```
db2support -H 3d
```

## 例 6

2009 年の最初 3 日間 (2009-01-01-00.00.00.000000 から 2009-01-04-00.00.00.000000 までの期間) に変更されたファイルにデータ収集を制限するには、以下を使用します。

```
db2support -H 3d:2009
```

## 例 7

2008-01-01-00.00.00.000000 から現在時刻までの期間に変更されたファイルにデータ収集を制限するには、以下を使用します。

```
db2support -t 2008
```

### 例 8

2009-01-01-00.00.00.000000 から 2009-03-01-00.00.00.000000 までの期間に変更されたファイルにデータ収集を制限するには、以下を使用します。

```
db2support -t 2009-01:2009-03
```

### 例 9

以下のセクションは、プロファイル・ファイルの例です。

```
<COLLECTION>
<NAME>List</NAME>
<CMD>ls -la $HOME</CMD>
<OUTFILE>list.out</OUTFILE>
</COLLECTION>
```

このプロファイル・ファイルでは、**db2support** は `ls -la $HOME` コマンドからの情報を収集し、結果は `db2supp_system.zip` ファイルの `USERCOLLECTION/list.out` に格納されます。タイムアウト値は必須ではないため、指定されていません。この場合、デフォルトのタイムアウト値である 180 秒が使用されます。

### 例 10

`db2support_hostname1.zip` ファイルの内容を解凍するには、次のようにします。

```
db2support -unzip db2support_hostname1.zip
```

このコマンドは現行ディレクトリの下に `db2support_hostname1` というディレクトリを作成し、この `db2support_hostname1` ディレクトリに `db2support_hostname1.zip` ファイルから解凍されたファイルが配置されます。

現行ディレクトリの `db2support.zip` を解凍し、`temp` ディレクトリに配置するには、以下のようになります。

```
db2support temp -unzip db2support.zip
```

解凍中のファイルの一部または全部が宛先ディレクトリに存在する場合は、ファイルを上書きするかどうかを選択するプロンプトが出されます。プロンプトが出ないようにするには、**-unzip** パラメーターと一緒に **quiet** パラメーターを発行します。

```
db2support temp -unzip quiet db2support.zip
```

### 例 11

パラメーター **-install** と **-host** の例を以下に示します。

現行ディレクトリに `db2support.zip` ファイルを作成するには、以下のようになります。

```
db2support -install
```

`db2support.zip` ファイルの出力パス `temp` を指定するには、以下のようになります。

```
db2support temp -install
```

診断データ収集の対象ホストを 1 つ指定するには、以下のようになります。

```
db2support -install -host myhost1
```



診断データ収集の対象ホストを複数指定するには、以下のようにします。

```
db2support -install -host myhost1,myhost2
```

#### 例 12

**db2support** 収集の合計タイムアウトを指定するには、次のようにします。

```
db2support -d sample -timeout 3
```

#### 例 13

ハードウェアおよびオペレーティング・システム情報の収集のタイムアウトを指定するには次のようにします。

```
db2support -d sample -c -s -timeout 15
```

#### 例 14

オブティマイザーの **db2support** 収集のタイムアウトを指定するには、次のようにします。

```
db2support -d sample -c -timeout 7 -st "select * from staff"
```

### db2support を使用した SQL ステートメントのアクティビティ・イベント・モニター・データの収集:

例えば、

- `db2support -d sample -st "select * from staff" -aem`

このコマンドは、**-c1 2** での現在の収集に加えて、**db2caem** コマンドを開始します。これにより、アクティビティ・イベント・モニターが作成され、詳細情報、セクション、値、および SQL ステートメント「select \* from staff」の実行時統計がキャプチャーされます。**db2support** コマンドは、すべての **db2caem** 出力を収集します。

- `db2support -d sample -sf badquery.sql -aem`

このコマンドは、**-c1 2** での現在の収集に加えて、**db2caem** を呼び出します。これにより、アクティビティ・イベント・モニターが作成され、詳細情報、セクション、値、および badquery.sql ファイルで指定された SQL ステートメントの実行時統計がキャプチャーされます。**db2support** コマンドは、すべての **db2caem** 出力を収集します。

- `db2support -d sample -actevm mymon -appid *LOCAL.amytang.100203234904 -uowid 44 -actid 1`

このコマンドは、**-c1 0** での現在の収集に加えて、**db2caem** コマンドを開始します。これにより、アクティビティ・イベント・モニター情報、つまり詳細情報、セクション、値、および既存のアクティビティ・イベント・モニターのイベント・モニター・オプションによって指定された SQL ステートメントの実行時統計がキャプチャーされます。**db2caem** コマンドは、この場合、アクティビティ・イベント・モニターを作成しません。**db2support** コマンドは、すべての **db2caem** 出力を収集します。

### DB2 pureScale 環境に固有の収集に関する db2support の例

#### 例 1

ホストでの実行:

```
host:~$ db2support
```

- このコマンドは、現行フォルダーに db2support.zip ファイルを作成します。このファイルには、DB2 診断データと、クラスター・マネージャー、クラスター・ファイル・システム、uDAPL などの DB2 pureScale コンポーネントに固有の追加の診断データ (すべてのホストから収集される) が含まれません。
- db2support.zip ファイルには PURESACLE というフォルダーが含まれます。PURESACLE フォルダーには、CFS、CM、UDAPL という 3 つの追加サブフォルダーがあり、それぞれに対応する情報ファイルが入ります。
- このコマンドに追加のパラメーターを指定して、より詳しい診断データを収集することができます。

### 例 2: db2support を -cm オプション付きで実行する

ホストでの実行:

```
host:~$ db2support -cm
```

- このコマンドは、デフォルトで db2support.zip ファイルを作成し、追加のクラスター・マネージャー・データを収集しますが、これにはスペースが大量に消費されたり時間がかかったりします。-cm オプションによって収集される追加のクラスター・マネージャー・データは、CM フォルダーに格納されます。このフォルダーは、db2support.zip ファイルの PURESACLE フォルダー内にあります。

### 例 3: db2support を -purescale オプション付きで実行する

ホストでの実行:

```
host:~$ db2support -purescale
```

- このコマンドは、デフォルトで db2support.zip ファイルを作成します。またこのコマンドは、クラスター・マネージャー、クラスター・ファイル・システム、uDAPL などの DB2 pureScale コンポーネントに固有の追加診断データを収集します。これには時間がかかったり、スペースが大量に消費されたりします。追加診断データは、db2support.zip ファイル内にある対応する PURESACLE、CFS、CM、UDAPL の各フォルダーに格納されます。
- このコマンドは、以下のコマンドによって収集されるデータと似た診断データを収集しますが、さらに DB2 pureScale 環境に固有の追加診断データも収集します。

```
host:~$ db2support -cm -cfs -udapl
```

## 使用上の注意

ビジネス・データのセキュリティを保護するために、このツールは、表データ、スキーマ (DDL)、またはログを収集しません。パラメーターによっては、スキーマおよびデータ (アーカイブ・ログなど) のいくつかの性質を組み込むことができるものもあります。データベースのスキーマまたはデータを公開するオプションは、注意して使用する必要があります。このツールが起動されると、機密データを扱う方法を示すメッセージが表示されます。

**db2support** ツールで収集されるデータは、ツールが実行されているマシンから取得できます。クライアント/サーバー環境では、データベースに関連した情報は、イン

スタンス接続またはデータベース接続を経由して、データベースが常駐するマシンから得られます。例えば、オペレーティング・システム情報またはハードウェア情報 (-s オプション)、および診断ディレクトリー (**diagpath** および **alt\_diagpath**) からのファイルは、**db2support** ツールを実行しているローカル・マシンから、そしてバッファ・プール、データベース構成、および表スペース情報などのデータは、データベースが物理的に常駐しているマシンから得られます。

**db2support** オプティマイザー・ツールで使用できる照会のタイプには、以下のような制限があります。

- 複数の照会はサポートされません。複数の照会を 1 つのファイル内に入れた場合、このツールはそれぞれの照会に必要なオブジェクトをすべて集めます。しかし、EXPLAIN されるのは最後の照会だけです。この状態は、組み込み静的 SQL ステートメントを含むファイルにも当てはまります。
- このツールはカスタマー・アプリケーションを実行しません。ただし、特定の不適切な照会または遅い照会を評価するための 3 つの方法のいずれかを使用している場合、**db2support** の実行時にアプリケーションを同時に実行することができません。
- ストアード・プロシージャはサポートされません。

**db2support** コマンドは、動的 SQL の EXPLAIN データを収集しません。

FODC パッケージが、デフォルトの診断パスとは異なるディレクトリー・パスに保管されている場合、または FODCPATH 設定で指定したパスに存在しない場合には、FODC パッケージを db2support.zip ファイルに含めるため、**-fodcpath** パラメーターを使用してその FODC パスを **db2support** コマンドに示す必要があります。

## db2swtch - デフォルト DB2 コピーとデータベース・クライアント・インターフェース・コピーの切り替え

デフォルト DB2 コピーとデフォルト・データベース・クライアント・インターフェース・コピーの両方を切り替えます。

デフォルト DB2 コピーとは、特定の DB2 コピーをターゲットとしていないアプリケーションによって使用されるコピーです。 **db2swtch** を発行すると、「デフォルト DB2 および IBM データベース・クライアント・インターフェース選択」ウィザードが起動します。このウィザードに従って、新しいデフォルト DB2 コピー、およびデフォルト・データベース・クライアント・インターフェース・コピーを設定することができます。このコマンドは Windows オペレーティング・システムでのみ使用できます。

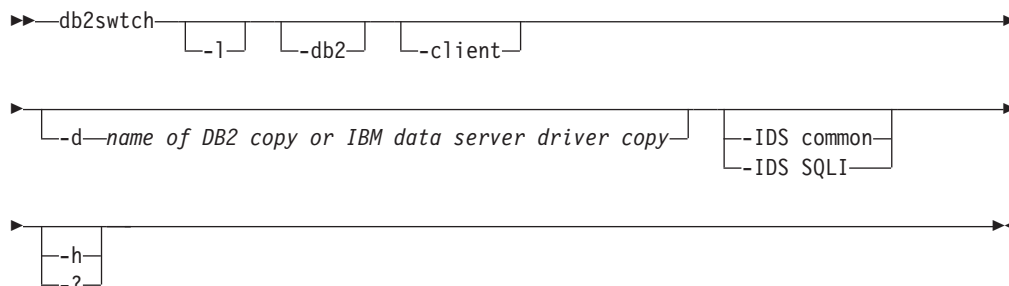
### 許可

ローカル管理者権限

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### 引数なし

ユーティリティーをグラフィック・モードで起動します。

**-l** システム上の DB2 コピーおよび IBM データ・サーバー・ドライバー・コピーのリストを表示します。

**-db2 -d** *DB2\_copy\_name*  
デフォルト DB2 コピーを、指定した名前のもに切り替えます。

db2swtch -db2 -d *DB2\_copy\_name*

**-client -d** *name of DB2 copy or IBM data server driver copy*  
デフォルト・クライアント・インターフェース・コピーを指定した名前のもに切り替えます。

db2swtch -client -d *DB2 コピーまたは IBM データ・サーバー・ドライバー・コピーの名前*

## db2swtch - デフォルト DB2 コピーとデータベース・クライアント・インターフェース・コピーの切り替え

### **-d** *DB2\_copy\_name*

デフォルト DB2 コピーとクライアント・インターフェース・コピーの両方を、指定した名前に切り替えます。

```
db2swtch -d DB2_copy_name
```

### **-IDS**

#### **common**

machine.config に含まれる IDS .NET データ・プロバイダーの参照を、共通 IDS .NET データ・プロバイダーにリダイレクトします。

**SQLI** machine.config に含まれる IDS .NET データ・プロバイダーの参照を、SQLI IDS .NET データ・プロバイダーにリダイレクトします。

**-h | -?** ヘルプ情報を表示します。

---

## db2sync - DB2 シンクロナイザーの開始

サテライトの初期構成および構成の変更が簡単になります。このコマンドは、同期化セッションの進行を開始、停止またはモニターしたり、サテライトの構成情報(例えば、通信パラメーター)をコントロール・サーバーにアップロードしたりすることにも使用できます。

### 許可

なし

### 必要な接続

なし

### コマンド構文

```
▶▶ db2sync --t  
--s application_version  
--g ▶▶
```

### コマンド・パラメーター

- t 管理者がサテライトのアプリケーション・バージョンまたは同期証明書のいずれかを変更できる、グラフィカル・ユーザー・インターフェースを表示します。
- s *application\_version* サテライトにアプリケーションのバージョンを設定します。
- g サテライトに現在設定されているアプリケーションのバージョンを表示します。

## db2sysray - DB2 システム・トレイの開始

DB2 システム・トレイ・ツールを開始します。これは、Windows オペレーティング・システム上の DB2 データベース・サービスの状況をモニターする、Windows オペレーティング・システムの通知アイコンです。 **db2sysray** は、サービスの開始および停止機能を持つとともに、そのサービスがいつ開始および停止されるのかを視覚的に示します。

**db2sysray** アイコンには、開始と停止の 2 つのモードがあります。モニターされているインスタンスが停止されると、アイコンには赤の正方形がオーバーレイされます。インスタンスが開始されると、赤の正方形は消えます。

パーティション・データベース環境では、すべてのパーティションが開始されることに限り、**db2sysray** アイコンは開始モードになります。1 つ以上のパーティションが停止される場合、**db2sysray** アイコンは停止モードになります。

複数の DB2 コピーが単一の Windows オペレーティング・システムにインストールされると、**db2sysray** はインストールされた DB2 コピーごとに DB2 インスタンスをモニターできます。デフォルト以外の DB2 コピーをモニターするには、モニター対象の DB2 コピーの **SQLLIB/bin** から **db2sysray.exe** アプリケーションを実行できます。

単一の DB2 インスタンスまたは複数のインスタンスを同時にモニターできます。複数インスタンスは、複数の **db2sysray** プロセスを使用してモニターできます。**db2sysray** によってモニターされるインスタンスごとに、システム・トレイに個別のアイコンが表示されます。各アイコン上にマウスを置くと、モニター対象の DB2 コピーの名前に続いて、その **db2sysray** アイコンによってモニターされた DB2 インスタンスの名前が表示されます。

**db2sysray** アイコンは、**db2sysray** コマンドを発行して、DB2 コマンド・ウィンドウから手動で起動することもできますし、Windows オペレーティング・システムの開始時に自動的に起動することもできます。**db2sysray** は、DB2 データベースのインストール時に、自動的に開始するように構成されています。しかし、**db2sysray** をシステムの開始時に自動的に開始するように構成しても、DB2 サービスを同時に開始しようとするにはなりません。それは、DB2 データベースの状況のモニターを自動的に開始することだけを意味します。

実行中の **db2sysray** プロセスによってモニターされているインスタンスに対して **db2idrop** コマンドを発行すると、**db2sysray** アプリケーションは、そのレジストリー項目をクリーンアップして終了するように強制されます。

**db2sysray** は、Windows オペレーティング・システムでのみ使用できます。

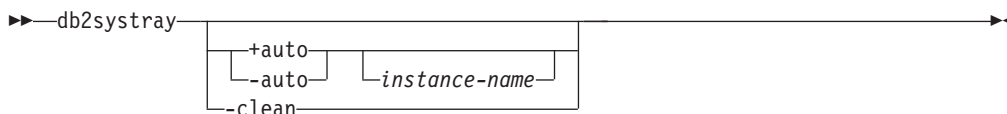
### 許可

**db2sysray** の開始には、特別な権限は必要ありません。処置を取るには適切な権限が必要です。

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

**+auto** Windows オペレーティング・システムの開始時に、指定されたインスタンスに対して **db2sysray** を自動的に開始します。また、**db2sysray** は、**db2sysray** メニュー・オプションの「スタートアップ時にツールを起動 (Launch Tool at Startup)」を有効にすることにより、自動的に起動するように構成することもできます。

**-auto** Windows オペレーティング・システムの開始時に、指定されたインスタンスに対して **db2sysray** を自動的に開始しないようにします。

#### *instance-name*

モニターされる DB2 インスタンスの名前。インスタンス名が指定されない場合、**db2sysray** はデフォルトのローカル DB2 インスタンスをモニターします。インスタンスが存在しない、または指定されたインスタンスが見つからない場合、**db2sysray** は自動的に終了します。

**-clean** **db2sysray** によってモニターされたすべての DB2 インスタンスのすべてのレジストリー項目をクリーンアップし、実行中のすべての **db2sysray.exe** プロセスを停止します。

### 例

1. C:¥SQLLIB¥bin> db2sysray

**DB2INSTANCE** 環境変数によって指定されたデフォルトの DB2 インスタンスに対して **db2sysray** を開始します。

2. C:¥SQLLIB¥bin¥> db2sysray DB2INST1

DB2INST1 という名前のインスタンスに対して **db2sysray** を開始します。

3. C:¥SQLLIB¥bin¥> db2sysray +auto

デフォルト DB2 インスタンスに対して **db2sysray** を開始し、Windows オペレーティング・システムの開始時にこのインスタンスのモニターを自動的に開始するように **db2sysray** を構成します。

4. C:¥SQLLIB¥bin¥> db2sysray +auto DB2INST1

DB2INST1 という名前のインスタンスに対して **db2sysray** を開始し、Windows オペレーティング・システムの開始時にこのインスタンスのモニターを自動的に開始するように **db2sysray** を構成します。

5. C:¥SQLLIB¥bin¥> db2sysray -auto

**DB2INSTANCE** 環境変数で定義されたデフォルト・インスタンスに対して自動開始オプションを無効にします。

6. C:¥SQLLIB¥bin¥> db2sysray -auto DB2INST1



インスタンス DB2INST1 に対して自動開始オプションを無効にします。

7. C:¥SQLLIB¥bin¥> db2sysstray -clean

**db2sysstray** によって作成されたすべてのレジストリー項目を除去し、実行中のすべての **db2sysstray.exe** プロセスを停止します。**db2sysstray.exe** プロセスが、インストール済みのその他の DB2 コピーに対して稼働している場合、それらはクリーンアップされません。クリーンアップする DB2 コピーごとに、SQLLIB/bin から db2sysstray -clean を実行する必要があります。

## db2tapemgr - テープ上のログ・ファイルの管理

DB2 ログ・ファイルの保管および取り出しをテープで行えるようにします。

**db2tapemgr** コマンドは **logarchmeth1** 値を読み取り、それらのログ・ファイルをディスクから指定の磁気テープ装置にコピーした後、コピーされたログ・ファイルの新しい位置によってリカバリー・ヒストリー・ファイルを更新します。

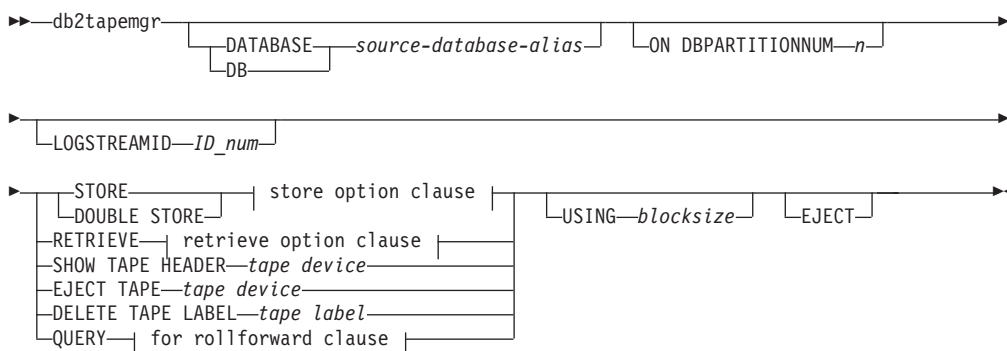
注: このコマンドは、DB2 pureScale 環境ではサポートされていません。

### 許可

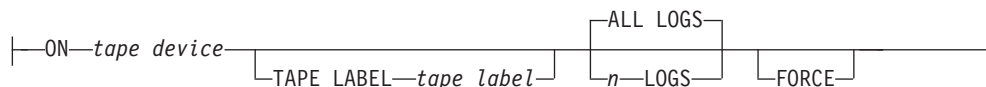
以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMANT

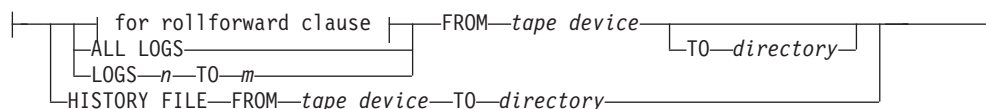
### コマンド構文



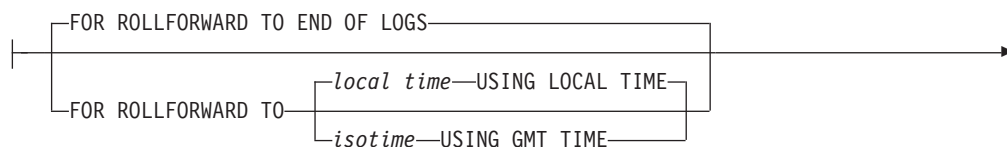
#### store option clause:



#### retrieve option clause:



#### for rollforward clause:



▶ ┌──USING HISTORY FILE──*history file*──┐

## コマンド・パラメーター

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

データベースの名前を指定します。値が指定されない場合、**DB2DBDFT** が使用されます。値が指定されず、**DB2DBDFT** が設定されていない場合、操作は失敗します。

### **ON DBPARTITIONNUM** *n*

対象とするデータベース・パーティションの番号を指定します。値が指定されない場合、**DB2NODE** が使用されます。

### **LOGSTREAMID** *ID\_num*

対象とするログ・ストリーム ID を指定します。値が指定されない場合、すべてのログ・ストリームが使用されます。

### **STORE ON** *tape device*

ログ・ファイルをテープに保管したり、ログ・ファイルをテープから削除したりします。

### **DOUBLE STORE ON** *tape device*

すべてのログ・ファイルが一度だけ保管されるようにして、ログ・ファイルが重複して保管されないようにします。テープに二重に保管されているログ・ファイルだけを削除して、それ以外のものはディスクに残されるようにします。

### **TAPE LABEL**

テープに適用されるラベルを指定します。**tape label** が指定されない場合、*database-alias**timestamp* の形式で自動的に生成されます (データベース別名が最大 8 文字、タイム・スタンプ (秒) が最大 14 文字、合計で最大 22 文字です)。

### **ALL LOGS** または *n LOGS*

コマンドをすべてのログに適用するか、指定した数のログに適用するかを指定します。

### **FORCE**

期限が切れていないテープを上書きすることを指定します。

### **USING** *blocksize*

テープ・アクセスのためのブロック・サイズを指定します。デフォルト・サイズは 5120 です。これは 512 の倍数でなければなりません。最小値は 512 です。

### **EJECT**

操作の完了後にテープがイジェクトされるように指定します。

### **RETRIEVE FOR ROLLFORWARD TO**

指定されたロールフォワードに必要なすべてのログに関して、ユーティリティーが対話式にプロンプトを表示して、テープからそれらを取り出すことを指定します。ディレクトリーが指定されない場合、**overflowlogpath** 構成

## db2tapemgr - テープ上のログ・ファイルの管理

パラメーターで指定されたパスが使用されます。ディレクトリーが指定されず、**overflowlogpath** が設定されていない場合、操作は失敗します。

### END OF LOGS

ログの最後までログ・ファイルを取り出すことを指定します。

### *isotime* USING GMT TIME

指定された時刻までログ・ファイルを取り出すことを指定します。

### *local time* USING LOCAL TIME

指定された時刻までログ・ファイルを取り出すことを指定します。

### USING HISTORY FILE *history file*

使用する代替の履歴ファイルを指定します。

### FROM *tape device*

ログ・ファイルの取り出し元の磁気テープ装置を指定します。

### TO *directory*

取り出したログ・ファイルのコピー先のディレクトリーを指定します。

### RETRIEVE ALL LOGS or LOGS *n TO m*

コマンドをすべてのログに適用するか、テープ上の指定した数のログに適用するかを指定します。

### FROM *tape device*

ログ・ファイルの取り出し元の磁気テープ装置を指定します。

### TO *directory*

取り出したログ・ファイルのコピー先のディレクトリーを指定します。

### RETRIEVE HISTORY FILE

履歴ファイルを取り出します。

### FROM *tape device*

ログ・ファイルの取り出し元の磁気テープ装置を指定します。

### TO *directory*

取り出したログ・ファイルのコピー先のディレクトリーを指定します。

### SHOW TAPE HEADER *tape device*

テープ・ヘッダー・ファイル DB2TAPEMGR.HEADER の内容を表示します。

### EJECT TAPE *tape device*

テープをイジェクトします。

### DELETE TAPE LABEL *tape label*

履歴ファイルから、指定のテープ・ラベルを示すすべての位置情報を削除します。

### QUERY FOR ROLLFORWARD TO

ロールフォワードに必要なログ・ファイルの位置を表示します。

### END OF LOGS

### *isotime* USING GMT TIME

指定の時刻まで操作がログを照会することを指定します。

*local time* **USING LOCAL TIME**

指定の時刻まで操作がログを照会することを指定します。

**USING HISTORY FILE** *history file*

使用する代替の履歴ファイルを指定します。

## db2tbst - 表スペース状態の獲得

16 進数の表スペース状態値を受け入れ、その状態を戻します。状態値は **LIST TABLESPACES** または **MON\_GET\_TABLESPACE** 表関数からの出力の一部です。

### 許可

なし

### 必要な接続

なし

### コマンド構文

▶▶—db2tbst—*tablespace-state*—————▶▶

### コマンド・パラメーター

*tablespace-state*

16 進数の表スペース状態値。

### 例

db2tbst 0x0000 の要求は、次の出力を生成します。

```
State = Normal
```

## db2tdbmgr - ツール・カタログ・データベースのマイグレーション

**db2tdbmgr** コマンドは、特定のツール・カタログ・データベースで **UPGRADE DATABASE** コマンドの実行後にそのツール・カタログ・データベース・オブジェクトをマイグレーションします。

### 許可

*sysadm*

### 必要な接続

このコマンドは、データベース接続を確立します。

### コマンド構文

```
▶▶ db2tdbmgr -d db_name -s schema_name [-u user_name]
[-p password]
```

### コマンド・パラメーター

**-d** *db\_name*

ツール・カタログ・データベース名。

**-s** *schema\_name*

ツール・カタログ・スキーマ名。

**-u** *user\_name*

ツール・カタログ・データベースを接続するために使用するユーザー名。

**-p** *password*

ツール・カタログ・データベースを接続するために使用するパスワード。

### 例

以下の例では、ツール・カタログ表をデータベース別名 `toolsdb`、スキーマ `systools` でマイグレーションします。

```
db2tdbmgr -d toolsdb -s systools -u db2inst1 -p *****
```

### 使用上の注意

このコマンドは、ツール・カタログ表を新しいバージョンにマイグレーションするだけであり、マイグレーション済みのツール・カタログ表を元のバージョンに戻すために使用することはできません。

マイグレーションの前にデータベースをカタログする必要があります。ほとんどの場合、マイグレーション・エラー・メッセージはエラー自体を説明するもので、エラーの場所に明確に言及します。エラー・メッセージにおいて表や列名などのオブジェクトに関して問題が指摘される場合、マイグレーションに指定したデータベース名のもとにある、報告されるオブジェクトが壊れているか欠落している可能性があります。

## db2top - DB2 モニター・ツール

**db2top** コマンドでは、AIX、Linux、HP-UX、および Solaris オペレーティング・システム上の複数パーティション・データベースまたは単一パーティション・データベースの統一された単一システム・ビューが提供されます。システムでの全体的な問題や、特定のデータベース・パーティションの問題を迅速に特定します。各データベース・パーティションからのスナップショット情報を結合することによって、稼働中の DB2 システムの動的なリアルタイム・ビューが提供されます。

### 有効範囲

**db2top** は、対話モードまたはバッチ・モードで実行することができます。

### 許可

以下の権限のいずれか。

- sysadm
- sysctrl
- sysmaint
- sysmon

### 必要な接続

インスタンス。再生モードおよびクライアント・モードでは、接続は必要ありません。

### コマンド構文

```

▶▶ db2top -A
  -a
  -B
  -b suboption
  -C suboption
  -D delimiter
  -d database name
  -f file </HH:MM:SS><+offset>
  -h
  -i interval in seconds
  -k
  -L
  -m duration in minutes
  -n node name
  -o outfile
  -P member_number
  -p password
  -R
  -s number
  -u username
  -V schema
  -x

```

### コマンド・パラメーター

**-A** 自動パフォーマンス分析を使用可能にします。**db2top** コマンドでは、上位



5 つのパフォーマンス報告が作成されます。このオプションは、再生モードとバックグラウンド・モード (**-b** パラメーター) で使用します。標準的な使用方法としては、以下のものがあります。

**db2top** の長時間 (例えば、4 時間) にわたる収集モードでの実行:

```
db2top -f collect.file -C -m 240
```

**db2top** の自動パフォーマンス分析を使用した再生モードでの実行。自動パフォーマンス分析は、バックグラウンド・モードでサポートされるすべての関数で使用でき、選択したサブオプション用の `.db2toprc` 構成ファイル内で指定されたデフォルトのソート基準に基づきます (自動パフォーマンス分析は、再生していない場合でもバックグラウンド・モードで使用することもできます)。例としては、最もアクティブなセッションを分析するために、以下のコマンドを発行します。

```
db2top -f collect.file -b l -A
```

**db2top** を再生モードで実行し、さらなる分析のために目的の時点へジャンプ:

```
db2top -f collect.file /HH:MM:SS
```

**-a** アクティブ・オブジェクトのみを表示することを指定します。

**-B** アクティブなオブジェクトをはっきりした (反転の) 色で表示します。これは、画面でカラーがサポートされない場合や、カラーが `.db2toprc` 構成ファイルでオフに設定されている場合に役立ちます。

**-b suboption**

**db2top** をバックグラウンド・モードで実行します。**-b** パラメーターを使用すると、**db2top** コマンドは情報を CSV 形式で表示します。**db2top** は、**-f file** パラメーターを使用した収集ファイルからのスナップショット・データの読み取りと組み合わせて、バックグラウンド・モードで実行することができます。バックグラウンド・モード (**-b**) での複数のサブオプションの実行は、サポートされません。**-b** パラメーターは、以下のいずれかのサブオプションの値を取ります。

**d** データベース  
**l** セッション  
**t** tablespaces  
**b** バッファ・プール  
**T** 表  
**D** 動的 SQL  
**s** ステートメント  
**U** ロック  
**u** ユーティリティ  
**F** フェデレーション (連合)  
**m** メモリー・プール

**-b** パラメーターで使用できるのは、パラメーター **-X**、**-L**、**-A**、**-s**、**-D**、**-o** だけです。

**-C db2top** をスナップショット・コレクター・モードで実行します。スナップショットのロー・データは、デフォルトでは `<db2snap-<dbname>-<Machine><bits><.bin>` に保存されます (**-f** を指定しない場合)。出力にはファイルの代わりにパイプも指定できます。コレクター・モード (**-C**) では、複数のサブオプションの指定がサポートされています。ロック情報を収集ファイルに含めるには、**-C** と共に **-x** を使用します。 **-C** パラメーターは、以下のいずれかのサブオプションの値を取ります。

- **b** : バッファ・プール
- **D** : 動的 SQL
- **d** : データベース
- **F** : フェデレーション
- **l** : セッション
- **s** : ステートメント
- **T** : 表
- **t** : 表スペース
- **U** : ロック

**db2top** コマンドは、**-C** コマンド・パラメーターを指定しない場合は、再生モードで実行されます。

### **-D delimiter**

フィールド区切り文字 (単一文字) を指定します。バックグラウンド・モード (**-b** パラメーター) で使用されます。

### **-d database name**

モニターするデータベースを指定します。

### **-f file </HH:MM:SS><+offset>**

• **-f** を指定しない場合は、**db2top** は再生モードで実行されます。 **-f** を指定した場合、これは出力ファイル名を示します。 **-d** で指定されるデータベース名は、存在している必要はありません。これは、EXPLAIN または Dump DB Struct を発行する場合にのみ参照されます。

再生モード (**-C** オプションを使用していない) で、スナップショット・データが前に *file* 内に収集されている場合は、offset (相対位置) によりファイル内の特定の時点にジャンプします。これは、時 (+10h)、分 (+10m) または秒 (+10s) 単位で表すことができます。 */HH:MM:SS* は、指定された時点までエントリーをスキップします。

• コレクター・モード (**-C** パラメーターを使用している) では、出力ファイル名を指定できます。 `</HH:MM:SS><+offset>` は、コレクター・モードの場合は無視されます。

**-h db2top** コマンドの使用方法の情報が表示されます。

### **-i interval in seconds**

画面更新の間の遅延を指定します。遅延時間は 1 秒未満にすることはできません。

- k 実際の値またはデルタ値 (差分の値) を表示します。デルタ・モードのすべての機能では、指定されたりフレッシュ・インターバルに関係なく、1 秒ごとにデルタ値が計算されて表示されます。
- L
 

現在実行しているステートメントか、または現行セッションからの最新のステートメントが **-i** パラメーターで指定されたそれぞれのレポート作成間隔で収集されることを指定します。SQL ステートメントが 1 つの間隔の間で開始して終了した場合、そのステートメントは含められません。

バックグラウンド・モードで、**-b 1 -L** パラメーターを使用して、セッション内で実行された SQL ステートメントをキャッチし、それらを現行作業ディレクトリー内にある ALL.sql ファイルに出力します。

現在実行中のステートメントか、接続しているセッションからの最新の SQL ステートメントが、**-i** パラメーターで指定された間隔ごとに収集されます。SQL ステートメントが 1 つの間隔の間で開始して終了した場合は、このステートメントはキャッチされません。
- m *duration in minutes*

**-b** と **-C** パラメーター用に **db2top** の処理時間を分単位で制限します。
- n*node name*

アタッチするノードを指定します。
- o 出力ファイル名を指定します。バックグラウンド・モード (**-b** パラメーター) で使用されます。
- P *dbpartition number*

スナップショットが実行されるデータベース・パーティション番号を指定します。**-P** パラメーターを指定して、*dbpartition number* を指定しないと、**db2top** はデフォルトのデータベース・パーティションにアタッチします。
- p データベースへのアクセスに使用するパスワードを指定します。これを省略した場合 **-u** パラメーターが指定されていると、ユーザーにパスワードの入力を求めるプロンプトが表示されます。
- R スナップショットを始動時にリセットします。
- s *number*

表示するサンプル数を指定します。バックグラウンド・モード (**-b** パラメーター) でのみサポートされます。
- u データベースへのアクセスに使用するユーザー名を指定します。
- V*schema*

EXPLAIN で使用するデフォルトのスキーマを指定します。
- x 追加のカウンターをセッションとアプリケーションの画面に表示する (セッションでより遅く実行される可能性があります) かどうかを指定します。
- X 出力の形式を XML に指定します。バックグラウンド・モードで、**-b** パラメーターと組み合わせて使用します。

## スナップショット・データ・コレクター

**db2top** モニター・ユーティリティーは、再生モードで実行できます。つまり、このユーティリティーはロー・バイナリーのスナップショット・データの保存済みコピーに対して実行できます。再生モードで実行するには、**db2top** をまずデータ・コレクター・モードで実行する必要があります。これは、コマンド行で **-C** スイッチを指定して **db2top** を実行するか、または、対話式セッションで **C** を押すことによりデータ収集をアクティブ化または非アクティブ化して、バッチ・モードで行います。これにより、ファイル `<db2snap-hostname.bin>` が現行ディレクトリーに作成されます。次に、**db2top** ユーティリティーを、**-f** 引数を使用して `<db2snap-<dbname>-<Machine><bits>.bin>` に対して実行することができます。

**db2top** モニター・ユーティリティーは再生モードでは、DB2 インスタンスにアタッチする必要はありません。これは、リモート・モニターに便利です。 **-C** パラメーターに使用できるいずれかのサブオプションを指定して、ストリーム・ファイルの内容とサイズを制限することができます。

### 例 (バッチ・モード)

コマンド・パラメーターは以下のとおりです。

```
-b 1          --> db2top 内の「セッション」に関連する情報をキャッチしながら、
                バックグラウンド・モードで実行します
-C           --> db2top をスナップショット・コレクター・モードで実行します
-d CUST      --> データベース
-i 3         --> 3 秒間隔
-m 60        --> db2top の処理時間を 60 分に制限します
-n node      --> ノード
-o db2top.xml --> 出力ファイルは db2top.xml
-p password  --> パスワード
-s 10000     --> 10000 のサンプルを収集します
-u userid    --> ユーザー ID
-V sv9       --> スキーマ
-x           --> セッションに追加のカウンターを表示します
```

下の例の各コマンドでは、「sv9」というスキーマを持つ「node」というノード上の「CUST」というデータベースをモニターします。データベース用のユーザー ID は「userid」で、パスワードは「password」です。

- 次の例のコマンドでは、データベースを 3 秒間隔で 60 分間バックグラウンド・モードでモニターし、セッション情報を `db2top.xml` というファイルに XML 形式で出力します。

```
db2top -d CUST -n node -u userid -p password -V sv9 -i 3 -b 1 -X -o db2top.xml
        -m 60
```

- 次の例のコマンドでは、データベースをバックグラウンド・モードとスナップショット収集モードでモニターし、セッション情報の 10000 個のサンプルを XML 形式で収集します。出力のバイナリー・ファイルは `db2snap-CUST-AIX64.bin` です。

```
db2top -d CUST -n node -u userid -p password -V sv9 -b 1 -x -s 10000 -X
        -f db2snap-CUST-AIX64.bin
```

- 次の例のコマンドでは、前の例のコマンドでの出力キャプチャーを再生します。

```
db2top -d CUST -f db2snap-CUST-AIX64.bin
```

詳しくは、『DB2 problem determination using db2top utility』(<http://www.ibm.com/developerworks/data/library/techarticle/dm-0812wang/>) を参照してください。

## db2trc - トレース

**db2trc** は、DB2 インスタンスや DB2 Administration Server (DAS) のトレース機能を制御します。トレース機能は、操作に関する情報を記録し、この情報を読みやすい形式にフォーマットします。

なお、トレース機能を有効にすると (デフォルトでは OFF になっている)、システムのパフォーマンスに影響を与える場合があります。したがって、トレース機能は、DB2 技術サポート担当者から指示された場合にのみ使用してください。指示されていない場合は、十分な情報を記録した後、トレースをオフにしてください。

DB2 のトレースは、繰り返し発生する問題や再現可能な問題を分析するとき特に役立ちます。これにより、サポート担当者の問題判別作業はかなり容易になります。

時折、DB2 データベース・システムを使用しているときに、エラー・メッセージが出されることがあります。例えば、「トレースを取得して、IBM ソフトウェア・サポートに電話してください」、「トレースをオンにして、トレース・レコードを調べてください」、「問題の説明、SQLCODE、SQLCA の内容 (可能であれば)、およびトレース・ファイル (可能であれば) などの情報を用意して、技術サポート担当者と連絡を取ってください」などのメッセージです。また、IBM ソフトウェア・サポートに問題を報告するときに、ご使用の環境に関する詳しい情報をキャプチャーするためにトレースを実行するように依頼される場合があります。

### 許可

UNIX オペレーティング・システムで DB2 インスタンスのトレースを行う場合は、以下のいずれかの権限が必要です。

- SYSADM
- SYSCTRL
- SYSMANT

UNIX オペレーティング・システムで DB2 Administration Server のトレースを行う場合は、以下の権限が必要です。

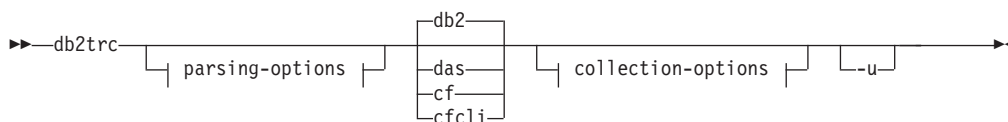
- DASADM

Windows オペレーティング・システムでは、権限は必要ありません。

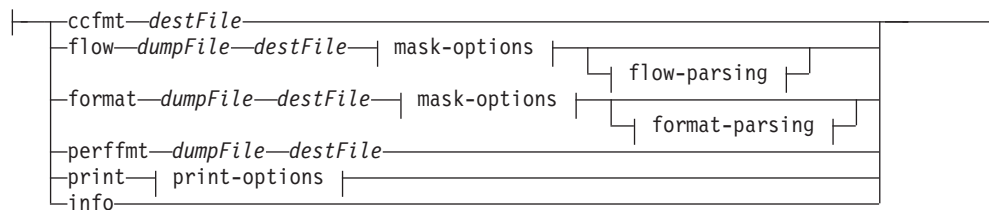
### 必要な接続

なし

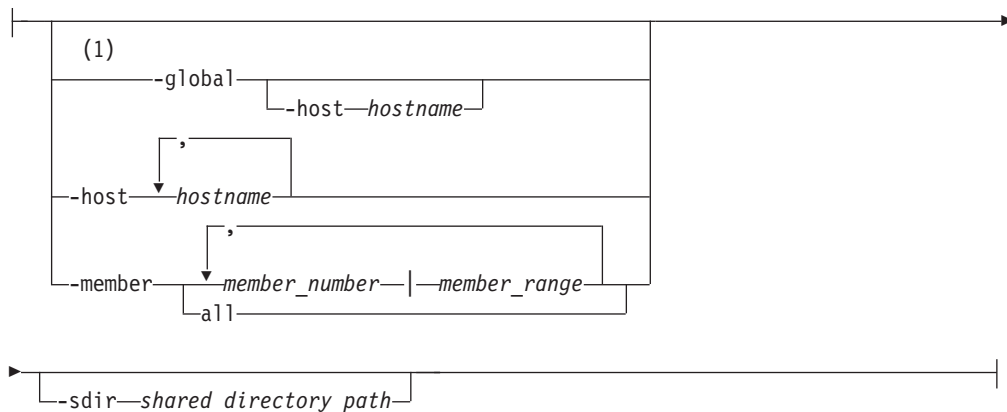
### コマンド構文



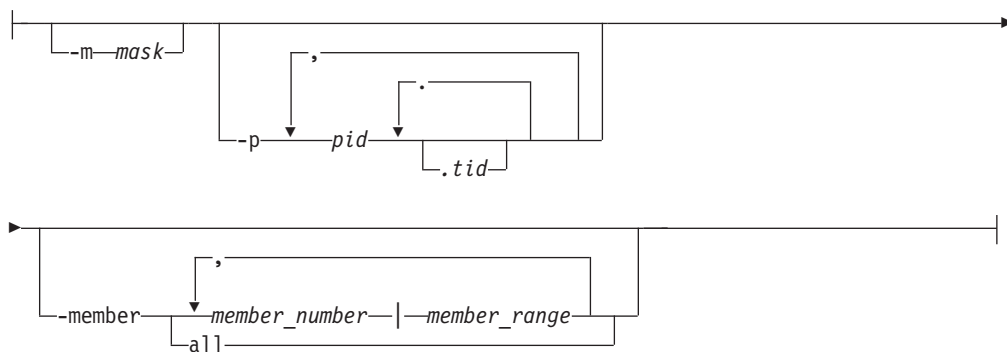
**parsing-options:**



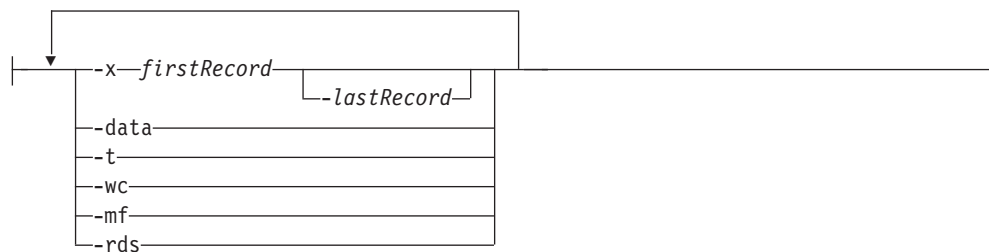
**location-options:**



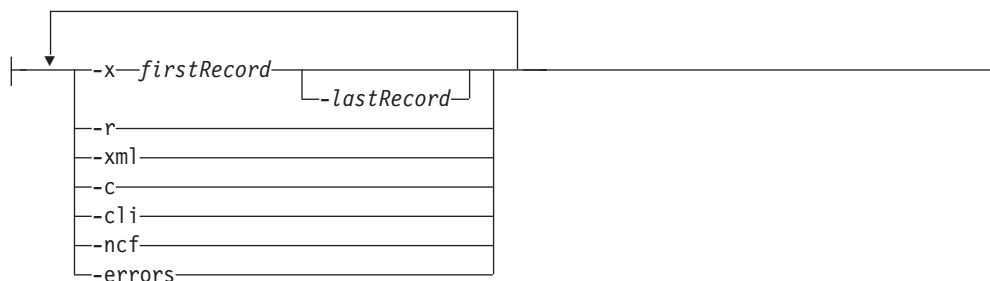
**mask-options:**



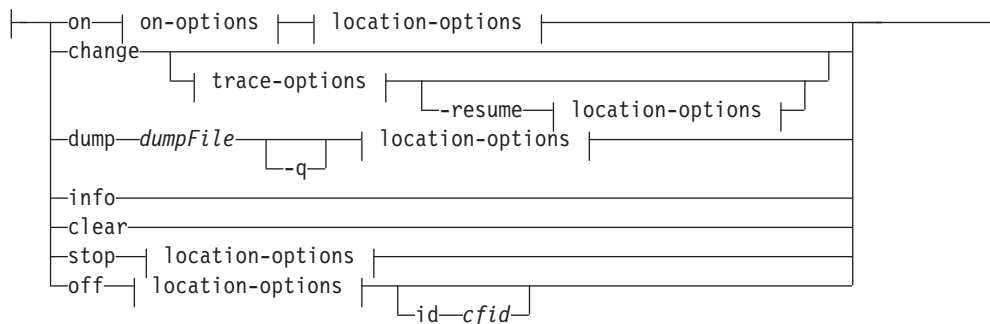
**flow-parsing:**



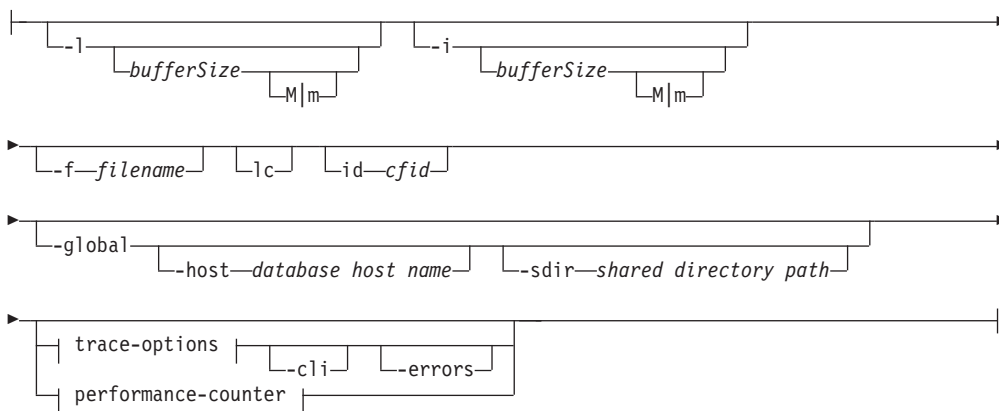
**format-parsing:**



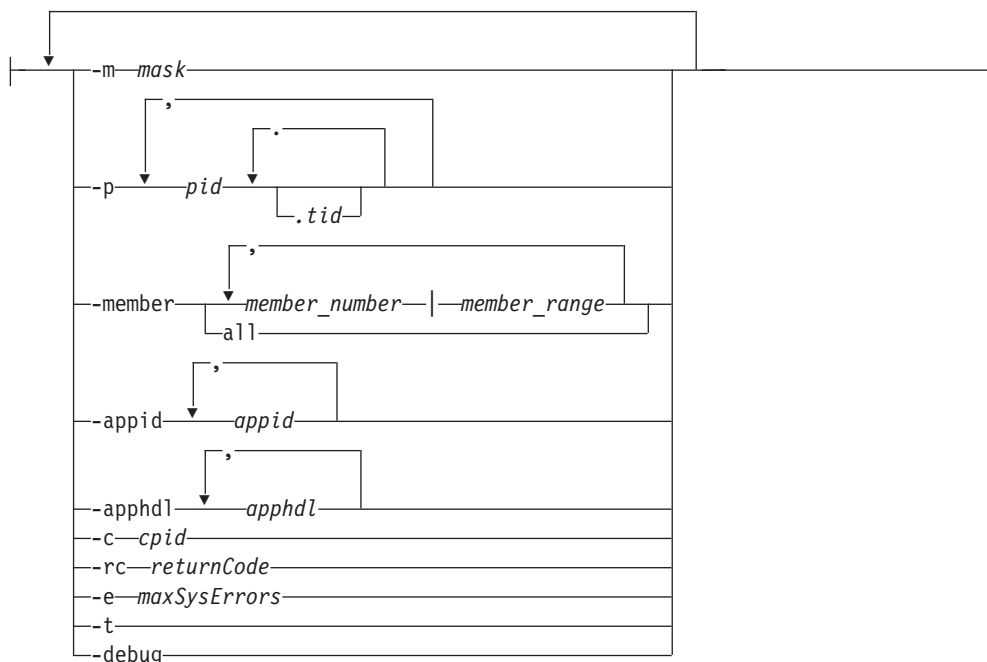
**collection-options:**



**on-options:**



**trace-options:**



**performance-counter:**



**print-options:**



**注:**

- 1 **-global** パラメーターは使用すべきではありません。 **-member all** パラメーター・オプションを使用すると、情報をグローバルに取得することができます。

**コマンド・パラメーター**

**-appid** トレースするアプリケーション ID を指定します。 **-appid** オプションは **on** および **change** オプションと共に指定された場合に有効になります。 **-appid** は **-perfcount** オプションと共に指定しても有効ではありません。

**-apphdl**  
 トレースするアプリケーション・ハンドルを指定します。 **-apphdl** オプションは **on** および **change** オプションと共に指定された場合に有効になります。 **-apphdl** は **-perfcount** オプションと共に指定しても有効ではありません。

**-member member\_number | member\_range**  
 どのデータベース・メンバー (パーティション) をトレースするか指定しま



す。このオプションを指定しない場合、このコマンドは **db2trc** コマンドが発行されるホスト上のすべての論理メンバーに対して実行されます。複数のメンバーを、*member\_number* のコンマ区切りリスト (*member1*, *member2*) として、または *member\_range* (*member\_range* はメンバーの範囲 (*member1*-*member3*)) を使用して指定できます。最初の 2 つの方式を組み合わせ使用することもできます。

**all** これを指定すると、コマンドは、*db2nodes.cfg* で定義されたすべてのメンバーに対して発行されます。

**-member** オプションは、**on**、**change**、**format (flow** および **format** 両方のオプション)、**stop**、および **off** の各オプションと共に指定した場合に有効になります。詳しくは、下記の例 2 および 3 を参照してください。**-member** オプションは、**-perfcnt** オプションと共に指定しても有効ではありません。

**db2** すべてのトレース操作を DB2 インスタンス上で実行することを指定します。これはデフォルトです。

**das** すべてのトレース操作を DB2 Administration Server インスタンス上で実行することを指定します。

**cf** すべてのトレース操作を CF サーバー上で実行することを指定します。このオプションを指定する場合、**-change**、**-clear**、および **-stop** オプションは機能しません。

**cfcli** すべてのトレース操作を メンバー 上で実行することを指定します。

注: CF クライアント・トレースを使用可能にするには、環境変数 **CA\_TRACE\_KEY\_FILE** を既存のファイルに設定する必要があります。このファイルは、SHARED MEMORY セットを作成するために CF クライアント・ライブラリーによって使用されます。このセットはトレース・データを収集するのに使用されます。加えて、このファイルのアクセス権と所有者情報は、作成される SHARED MEMORY のアクセス権と所有者情報を設定するために使用されます。同じ値に設定されるこの環境変数から始まるすべてのプロセスは、同じ SHARED MEMORY セグメントを使用することになります。DB2 によって使用されるデフォルト名は  $\${INSTHOME}/sql1lib/tmp/.CF\_DB2TRACE.key$  です。この環境変数は *db2profile* および *db2cshrc* ファイル内に自動的に設定されます。ただし、これらがソース生成されていない場合には、この値を手動で設定する必要があります。

注: CF クライアントのトレース・バッファは 10 MB を超えることができません。

**on** トレース機能を開始するには、このパラメーターを使用します。パラメーターのリストについては、下記の『共有トレース・オプション』のセクションを参照してください。

**-l** [*bufferSize*]

このオプションは、トレース・バッファのサイズと振る舞いを指定します。**-l** は、最後のトレース・レコードを保持することを指定します (つまり、バッファが満杯になると最初のレコードが上書きされます)。バッファ・サイズは、バイト単位でもメガバイト単位でも指定できます。メガバイト単位でバッファ・サイズを指

定する場合は、バッファ・サイズに文字 `M | m` を追加します。例えば、バッファ・サイズ `4 MB` で **db2trc** を開始する場合は、次のようにします。

```
db2trc on -l 4m
```

トレース・バッファ・サイズのデフォルト値や最大値は、プラットフォームによって異なります。最小バッファ・サイズは `1 MB` です。なお、バッファ・サイズは `2` の累乗でなければなりません。

#### **-i** [*bufferSize*]

このオプションは、トレース・バッファのサイズと振る舞いを指定します。**-i** は、最初のトレース・レコードを保持することを指定します (つまり、バッファがいったん満杯になると、レコードの書き込みはそれ以上行われません)。バッファ・サイズは、バイト単位でもメガバイト単位でも指定できます。メガバイト単位でバッファ・サイズを指定する場合は、バッファ・サイズに文字 `M | m` を追加します。

#### **-f** *filename*

ファイルにトレースするとき、完全修飾ファイル名を指定する必要があります。**-l** または **-i** と **-f** オプションを同時に使用する場合、それぞれの *bufferSize* 値がディスク上のファイルのサイズを制限します。**-l** は最後のトレース・レコードを保持し、ファイル内での折り返しを許可します。**-i** は最初のトレース・レコードを保持し、ファイル・サイズの限界に達したときにトレースを停止します。メガバイト単位でファイル・サイズを指定する場合は、**-i** および/または **-l** *bufferSize* に指定する値の後ろに `M | m` という文字を追加し、ギガバイト単位でファイル・サイズを指定する場合は同じ場所に `G | g` という文字を追加します。

**-cli** CLI/ODBC ドライバーと CLI アプリケーションをトレースします。

#### **-errors**

関数の出口でエラーおよびゼロ以外の戻りコードのみをトレースします。このオプションを **-debug** または **-perfcount** オプションと同時に指定することはできません。

**-lc** CF サーバーまたは CF クライアントのいずれかに有効な CF トレース・コンポーネントおよびトレース・レベルをすべてリストします。

#### **-id** [*cf-id*]

`db2nodes.cfg` ファイルに指定されている CF サーバー ID。デフォルトでは、すべての CF サーバーに対して操作が実行されます。ただし、このオプションが指定されている場合には、指定された CF サーバーだけが影響を受けます。

**change** この `collection` オプションを使用すると、有効になっているトレース・オプションを変更することができます。パラメーターのリストについては、下記の『共有トレース・オプション』のセクションを参照してください。

**-resume**

このオプションを使用すると、中断されていたプロセスの実行を再開することができます。 **-suspend** が有効になっていない場合は再開することはできません。

## 共有トレース・オプション

**on** と **change** の間で共有される共通のトレース・オプション。

**-m mask**

収集またはフォーマットされるデータの量を減らします。トレース・マスクの形式は、*types.products.components.functions.categories* です。

マスクの値は本来、IBM サポートによって提供されます。

マスクは 5 つの部分 (トレース・レコードのタイプ、製品、コンポーネント、関数、関数のカテゴリ) で構成されます。各部分は、コンマ区切りのリスト、ハイフン区切りの範囲、または単一の項目で構成できます。アスタリスク (\*) は、何にでもマッチングするものとして使用できます。フィールドの値は、その名前または対応する数値によって指定できます。フルネームの *products*、*components*、または *functions* 部分のいずれかの名前を指定したマスクの短形式を使用することができます。マスクを *\*.\*.\*.\*.\** と設定することは、マスクを指定しないのと同じことです。

例: `-m "entry,exit.*.SQL0,SQL.*.*"`

DB2 pureScale 環境の場合、CF サーバーまたは CF クライアント (あるいはその両方) のトレースを使用可能にするように、マスクは多重定義されています。マスクは、CF を扱う場合とは幾分異なる方法で使用されます。マスクの値は、**-lc** オプションを使用することによって検出できます。CF サーバーおよび CF クライアントの場合、トレース・マスクの各部分は、次のような意味を持ちます。

- *types* は常に \* になります。
- *products* は CF または CFCLI のいずれかになります (大/小文字は区別されません)。
- *components* は、1 つ以上の CF サーバーまたは CF クライアント・コンポーネントの任意の組み合わせになり、**-lc** フラグを使ってリストできます。
- *functions* は常に \* になります。CF はこのフィーチャーをサポートしていないからです。
- *categories* は、1 つ以上の CF サーバーまたは CF クライアント・トレース・レベルの任意の組み合わせになり、**-lc** フラグを使ってリストできます。

例: `-m ".*.CF.svr_list.*.CF_TRACE_ERROR,CF_TRACE_ALL"`

**-p pid [tid]**

指定したプロセス ID (*pid*) およびスレッド ID (*tid*) に対してのみトレース機能を有効にします。 *tid* を指定する場合は、必ずピリオド (.) を含めてください。1 つのプロセス ID に対して複数のスレッド ID を指定できます。1 つの *pid* において複数の *tid* を区切

るには、ピリオドを使用します。コンマで、*pid tid* のそれぞれのペアを区切ります。最大 64 個の *tid* を 1 つの *pid* とペアにできます。

例えば、プロセス 10、20、および 30 に対してトレースを有効にする場合は、次のような構文を使用します。

```
db2trc on -p 10,20,30
```

プロセス 100 のスレッド 33 とプロセス 200 のスレッド 66 に対してのみトレースを有効にする場合は、次のような構文を使用します。

```
db2trc on -p 100.33,200.66
```

プロセス 77 のスレッド 1、2、3、4 と、プロセス 88 のスレッド 5、6、7、8 でトレースを有効にする場合は、次の構文を使用します。

```
db2trc on -p 77.1.2.3.4,88.5.6.7.8
```

#### -c *cpid*

この比較プロセスのみをトレースまたはフォーマットします。

#### -rc *returnCode*

*returnCode* をシステム・エラーとして扱います。 *returnCode* を符号付き整数として指定する必要があります。

#### -e *maxSysErrors*

*maxSysErrors* システム・エラーの発生後にトレースを停止します。

#### -t

タイム・スタンプを組み込みます。

#### -debug

これは、デバッグの目的で IBM サポートによって使用される内部オプションです。この使用は推奨されていません。

**info** 以下は、このパラメーターを使用してリストされる環境情報の例です。

```
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
Trace starting time : 2011-03-25-15.03.58.909713-240

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
Member mask           : none
Application ID mask   : none
Application Handle mask : none
```

**dump** *dumpFile*

バッファに格納されるバイナリー形式のトレース情報をファイルにダンプします。次のコマンドは、トレース情報を現行ディレクトリーの `db2trc.dmp` というファイルに入れます。

```
db2trc dump db2trc.dmp
```

このパラメーターでダンプ・ファイル名を指定します。明示的にパスが指定されていない場合、そのバイナリー形式のダンプ・ファイルは現行ディレクトリーに保存されます。

**-q** 静止モード。

**ccfmt** *destFile*

コード・カバレッジ・トレースをダンプおよびフォーマットします。ダンプの宛先ファイル名を指定します。

**flow** *dumpFile destFile*

トレースをバイナリー・ファイルにダンプした後、読みやすいテキスト・ファイルにフォーマットします。 **flow** オプションを使用して、プロセス別またはスレッド別にソートされたレコードをフォーマットします。ダンプ・ファイルの名前および生成される宛先ファイルの名前を指定します。以下に例を示します。

```
db2trc flow db2trc.dmp db2trc.flw
```

**-x** *firstRecord* [-*lastRecord*]

レコード番号 *firstRecord* から *lastRecord* までのみを表示します。

**-data** すべてのトレース・レコード・データをフローに組み込みます。

**-t** 使用可能な場合に、タイム・スタンプ (秒:ナノ秒の形式) を組み込みます。

**-wc** 使用可能な場合に、壁時計タイム・スタンプを含めます。このオプションを使用できるようにするには、タイム・スタンプのキャプチャーを組み込む **-t** オプションも指定して、トレースを ON にしなければなりません。使用例については、例 1 を参照してください。

**-mf** 各フローごとに個別の宛先ファイルを生成します。

**-rds** 使用可能な場合に、RDS オペレーターの情報を組み込みます。

**format** *dumpFile destFile*

トレースをバイナリー・ファイルにダンプした後、読みやすいテキスト・ファイルにフォーマットします。 **format** オプションを使用して、レコードを古い順にフォーマットします。

**-x** *firstRecord* [-*lastRecord*]

レコード番号 *firstRecord* から *lastRecord* までのみを表示します。

**-r** 逆順で出力します。

**-xml** XML での構文解析が可能な形式でデータを出力します。

**-c** 通信バッファをフォーマットします。

- cli** CLI/ODBC ドライバーと CLI アプリケーション・バッファーをフォーマットします。
- ncf** コンポーネントのカスタム・フォーマットを使用しません。
- errors**  
関数の出口でエラーおよびゼロ以外の戻りコードのみをトレースします。
- global** リモート・ホストに対しても **db2trc** を実行することを指定します。このオプションは、DB2 バージョン 9.7 フィックスパック 4 およびそれ以降のフィックスパックでは非推奨です。
- host** *hostname*  
指定したホストに対してのみ **db2trc** を実行することを指定します。このパラメーターを指定しない場合、コマンドはローカル・ホストに対して発行されます。複数のホストを指定する場合、すべてのホスト名が有効でないと、コマンドを完了できません。
- sdir** *shared directory path*  
**-f** および **dump** パラメーターに該当するファイルを保存するために **db2trc** が使用する共有ディレクトリーを指定します。
- perffmt** *dumpFile destFile*  
パフォーマンス・トレース・フォーマッターは、パフォーマンス・カウンター・データを含むダンプ・ファイルを読みやすいテキストにフォーマットする構文解析オプションです。
- print -stack** *recordID flowFile*  
指定されたフロー・ファイルの指定レコード ID における、すべてのスタック・フレームのバックトレースを出力します。詳しくは、例 5 を参照してください。
- clear** トレース・バッファーの内容を、特に特定のデータベースに接続する直前にクリアします。このオプションは、収集された情報の量を減らすのに使用できます。これは、対象のデータベースとの接続を確立する前に、累積した不要な情報のバッファーをクリアすることによって行われます。
- stop** この **collection** オプションは、要求時にトレースを停止します。すべてのプロセスはトレースを中断しますが、トレース・バッファーの内容は後でダンプできるように保持されます。このアクションは、トレース機能を完全に無効にする **off** オプションと対照的です。
- off** トレース機能を無効にします。トレースをファイルにダンプした後、次のように入力してトレース機能を無効にします。  
db2trc off
- id** [*cf-id*]  
db2nodes.cfg ファイルに指定されている CF サーバー ID。デフォルトでは、すべての CF サーバーに対して操作が実行されます。ただし、このオプションが指定されている場合には、指定された CF サーバーだけが影響を受けます。
- u** ほとんどのコマンド行オプションに関する追加情報を提供します。コマンド



行の項目の一般的な形式については、下記の『使用上の注意』に示されています。以下は、DAS インスタンスの `dump` コマンドに関する詳しい情報を入手する例です。

```
db2trc das dump -u
```

## 例

**例 1** 壁時計タイム・スタンプを組み込んでトレースをキャプチャーするには、トレースを ON にする際に `-t` オプションを指定し、`-t -wc` オプションと `flow` オプションを指定して壁時計タイム・スタンプを読み取り可能なテキスト・ファイル内に出力しなければなりません。以下は実行できるステップの例です。

1. 以下のコマンドを実行し、トレースを ON にして、タイム・スタンプ情報をキャプチャーし、ダンプ・ファイル名を指定します。

```
db2trc on -t -f db2trc.dmp
```

2. トレース期間の終了後、以下のコマンドを実行してトレースを OFF にします。

```
db2trc off
```

3. バイナリー・ダンプ・ファイル (.dmp) を読み取り可能テキスト・ファイル (db2trc.flw など) にフォーマットするには、以下のコマンドを実行します。

```
db2trc flow -t -wc db2trc.dmp db2trc.flw
```

以下はフォーマットの完了時に表示される出力の例です。

```
Total number of trace records      : 3349
Trace truncated                     : NO
Trace wrapped                       : NO
Number of trace records formatted   : 43 (pid: 5414 tid 182967198368 node: 0)
Number of trace records formatted   : 2690 (pid: 29615 tid 182960067008 node: 0)
Number of trace records formatted   : 118 (pid: 5394 tid 183102335328 node: 0)
Number of trace records formatted   : 498 (pid: 29616 tid 182965078816 node: -1)
```

4. 読み取り可能テキスト・ファイル db2trc.flw の内容を表示するには、以下のコマンドを実行します。

```
more db2trc.flw
```

以下は、壁時計タイム・スタンプ情報を含む読み取り可能テキスト・ファイルの内容の読み取り時に表示される出力の例です。

```
pid = 5414 tid = 182967198368 node = 0
```

```
1      0.000000000  clp_bp_con data [probe 21] 2009-06-16-11.02.32.38407400
2      0.000038000  | sqllossig entry 2009-06-16-11.02.32.38411200
3      0.000050000  | sqllossig exit 2009-06-16-11.02.32.38412400
4      0.000057000  | sqlorque2 entry 2009-06-16-11.02.32.38413100
5      0.000062000  | sqlogmbkEx entry 2009-06-16-11.02.32.38413600
6      0.000070000  | | sqloGetPrivatePoolHandle entry 2009-06-16-11.02.32.38414400
7      0.000077000  | | sqloGetPrivatePoolHandle exit 2009-06-16-11.02.32.38415100
8      0.000088000  | sqlogmbkEx mbt [Marker:PD_OSS_ALLOCATED_MEMORY ] 2009-06-16-11.02.32.38416200
9      0.000092000  | sqlogmbkEx exit 2009-06-16-11.02.32.38416600
10     0.000094000  | sqlorqueInternal entry 2009-06-16-11.02.32.38416800
11     0.000096000  | | sqloSetAlarmApp entry 2009-06-16-11.02.32.38417000
12     0.000099000  | | | sqloSigMask entry 2009-06-16-11.02.32.38417300
13     0.000101000  | | | sqloSigMask exit 2009-06-16-11.02.32.38417500
14     0.000103000  | | | sqlohsig entry 2009-06-16-11.02.32.38417700
15     0.000105000  | | | sqlohsig exit 2009-06-16-11.02.32.38417900
16     0.000108000  | | sqloSetAlarmApp exit 2009-06-16-11.02.32.38418200
2825   5.000561000  | | sqloClearAlarmApp entry 2009-06-16-11.02.37.38463500
2826   5.000576000  | | | sqloSigMask entry 2009-06-16-11.02.37.38465000
2827   5.000579000  | | | sqloSigMask exit 2009-06-16-11.02.37.38465300
2828   5.000582000  | | | sqlohsig entry 2009-06-16-11.02.37.38465600
2829   5.000585000  | | | sqlohsig exit 2009-06-16-11.02.37.38465900
2830   5.000587000  | | | sqloClearAlarmApp exit 2009-06-16-11.02.37.38466100
2831   5.000589000  | | sqlorqueInternal exit [rc = 0x870F00B9 = -2029059911 = SQLQ_SEM_TIMEOUT] 2009-06-16-11.02.37.38466300
```

## db2trc - トレース

```
2832      5.000592000 | sqlofmbkEx entry      2009-06-16-11.02.37.38466600
2833      5.000597000 | sqlofmbkEx mbt [Marker:PD_OSS_FREED_MEMORY ] 2009-06-16-11.02.37.38467100
2834      5.000599000 | sqlofmbkEx exit      2009-06-16-11.02.37.38467300
2835      5.000601000 | sqlorque2 exit [rc = 0x870F00B9 = -2029059911 = SQLO_SEM_TIMEOUT] 2009-06-16-11.02.37.38467500
2836      5.000614000 | clp_bp_con data [probe 21] 2009-06-16-11.02.37.38468800
2837      5.000617000 | sqllossig entry     2009-06-16-11.02.37.38469100
2838      5.000620000 | sqllossig exit     2009-06-16-11.02.37.38469400
2839      5.000623000 | sqlorque2 entry    2009-06-16-11.02.37.38469700
2840      5.000626000 | sqlogmbkEx entry   2009-06-16-11.02.37.38470000
2841      5.000628000 | | sqloGetPrivatePoolHandle entry      2009-06-16-11.02.37.38470200
2842      5.000631000 | | sqloGetPrivatePoolHandle exit     2009-06-16-11.02.37.38470500
2843      5.000636000 | sqlogmbkEx mbt [Marker:PD_OSS_ALLOCATED_MEMORY ] 2009-06-16-11.02.37.38471000
2844      5.000638000 | sqlogmbkEx exit    2009-06-16-11.02.37.38471200
2845      5.000640000 | sqlorqueInternal entry 2009-06-16-11.02.37.38471400
2846      5.000643000 | | sqloSetAlarmApp entry      2009-06-16-11.02.37.38471700
2847      5.000646000 | | sqloSigMask entry  2009-06-16-11.02.37.38472000
2848      5.000647000 | | sqloSigMask exit  2009-06-16-11.02.37.38472100
2849      5.000649000 | | sqlohsig entry     2009-06-16-11.02.37.38472300
2850      5.000651000 | | sqlohsig exit     2009-06-16-11.02.37.38472500
2851      5.000654000 | | sqloSetAlarmApp exit 2009-06-16-11.02.37.38472800
```

例 2 以下は -member トレース・マスクの使用例です

- db2trc on -member n1[,n2,n3,n64]

このコマンドで、どのデータベース・メンバーつまりパーティションをトレースするかを指定できます。

注: 整数でなければなりません。複数のメンバーを指定する場合、それぞれのメンバーをコマンドで区切る必要があります。最大 64 メンバーまで指定できます。

- db2trc chg -member n1[,n2,n3,n64]

このコマンドで、指定されたデータベース・メンバーつまりパーティションのメンバー・マスクを変更できます。

- db2trc stop -member n1[,n2,n3,n64]

このコマンドで、メンバー・マスクからメンバーを削除できます。メンバー・マスクから最後のメンバーが削除されるとトレースは完全に停止し、**db2trc stop** を実行した場合と同じ効果があります。

- このコマンドで発行されるメンバーの番号は、現行のメンバー・マスクに存在していなければなりません。**db2trc info** を実行すると、現行のメンバー・マスクが表示されます)。
- **db2trc on** コマンドを **-member** オプションを指定せずに実行した場合、メンバー・マスクにはメンバーが何もありません。これは、すべてのメンバーがトレースされることを意味します。
- 現行ホストのすべてのメンバー (db2nodes.cfg ファイルで定義されている) がこのコマンドで指定されると、**db2trc stop** を実行 (**-member** オプションを指定しない **db2trc stop** を実行) したのと同じ効果があり、そのホストのすべてのメンバーのトレースを完全に停止します。

- db2trc off -member n1[,n2,n3,n64]

このコマンドで、メンバー・マスクからメンバーを削除できます。メンバー・マスクから最後のメンバーが削除されるとトレースはオフになり、**db2trc off** を実行した場合と同じ効果があります。

- このコマンドで発行されるメンバーの番号は、現行のメンバー・マスクに存在していなければなりません。**db2trc info** を実行すると、現行のメンバー・マスクが表示されます)。



- **db2trc on** コマンドを **-member** オプションを指定せずに実行した場合、メンバー・マスクにはメンバーが何もありません。これは、すべてのメンバーがトレースされることを意味します。
- 現行ホストのすべてのメンバー (db2nodes.cfg ファイルで定義されている) がこのコマンドで指定されると、**db2trc off** と同じ効果があります。**db2trc off** を **-member** オプションを指定せずに実行すると、そのホストのトレースはオフになります。
- db2trc flw -member n1[,n2,n3,n64] および db2trc fmt -member n1[,n2,n3,n64]

このコマンドを実行して、どのメンバーをフォーマットされたトレースに含めるかを指定できます。

**例 3** 以下は、**-member** トレース・マスクを、次のデータのように定義されている db2nodes.cfg で使用する例です。

```
0 host1 0
1 host1 1
2 host1 2
3 host2 0
4 host3 0
```

- db2trc on[ chg] -member 1,2,3

**-member** を実行する場合、**-member** で指定されたメンバーの番号は関連したホスト名にマップされ、関連したホストで **rah** (または **db2\_all**) コマンドにより実行されます。この例では、**db2trc -member 1,2** は host1 で実行され、**db2trc -member 3** は host2 で実行されます。

- db2trc on -host host1,host2

**db2trc on** は host1 と host2 の両方で実行されます

- db2trc on -member all

**db2trc on** は host1、host2 および host3 で実行されます

**例 4** 以下は、**-appid** と **-apphdl** の使用例です

- db2trc on -appid appid1,appid2

このコマンドは、特定のアプリケーション ID のトレースをオンにします。このコマンドは、最大 12 のアプリケーション ID までサポートします。

- db2trc chg -appid appid1,appid2

このコマンドは、特定のアプリケーション ID を変更 (またはリセット) します。リセットするには、**db2trc chg -appid none** コマンドを実行してください。このコマンドは、マスクのアプリケーション ID をすべて除去します。

- db2trc on -apphdl apphdl1,apphdl2,apphdl3

このコマンドは、特定のアプリケーション・ハンドルのトレースをオンにします。このコマンドは、最大 16 のアプリケーション・ハンドルまでサポートします。

- db2trc chg -apphd1 apphd11,apphd12,apphd13

このコマンドは、特定のアプリケーション・ハンドルを変更 (またはリセット) します。リセットするには、**db2trc chg -apphd1 none** コマンドを実行してください。このコマンドは、マスクのアプリケーション・ハンドルをすべて除去します。

**例 5** サンプルのフロー・ファイル `test.flw` を以下の例に示します。

```
pid = 1608 tid = 47604608002368 node = 0

1      sqlqsig entry [eduid 1 eduname db2sysc]
2      sqlqsig exit [rc = 0x840F0001 = -2079391743 = SQL0_ACCD]
3      sqkfFastCommManager::ResourceSelfTuning entry [eduid 1 eduname db2sysc]
4      | sqlogmt entry [eduid 1 eduname db2sysc]
5      | sqlogmt exit
6      sqkfFastCommManager::ResourceSelfTuning exit
7      sqkfFastCommManager::CollectResourceUsageStats entry [eduid 1 eduname db2sysc]
8      | sqlogmt entry [eduid 1 eduname db2sysc]
9      | sqlogmt exit
10     sqkfFastCommManager::CollectResourceUsageStats exit
11     sqkfFastCommManager::UpdateMemoryConsumptionStats entry [eduid 1 eduname db2sysc]
12     sqkfFastCommManager::UpdateMemoryConsumptionStats exit
13     sqleSyscUpdateDynamicVars entry [eduid 1 eduname db2sysc]
14     | sqloGetSysMonSetting entry [eduid 1 eduname db2sysc]
15     | | sqloGetEnvUnCached entry [eduid 1 eduname db2sysc]
16     | | | EnvPrfOpen entry [eduid 1 eduname db2sysc]
17     | | | | EnvKeyName entry [eduid 1 eduname db2sysc]

19     | | | | | sqloxltc_app entry [eduid 1 eduname db2sysc]
20     | | | | | sqloxltc_app exit
21     | | | | | sqloGetUserAttribByName data [probe 770]
22     | | | | | sqloGetUserAttribByName data [probe 820]
23     | | | | | sqloxult_app entry [eduid 1 eduname db2sysc]
24     | | | | | sqloxult_app exit
25     | | | | | sqloGetUserAttribByName exit
26     | | | | | EnvKeyName exit
27     | | | | | EnvPrfOpen exit
28     | | | | | sqloGetEnvUnCached exit
29     | | | | | sqloGetSysMonSetting exit
30     | | | | | sqleSyscUpdateDynamicVars exit
```

`test.flw` ファイルの行 5 におけるすべてのスタック・フレームのバックトレースを出力するには、次のコマンドを実行します。

```
db2trc print -stack 5 test.flw
```

または

```
db2trc print -s 5 test.flw
```

次の出力が表示されます。

```
pid = 1608 tid = 47604608002368 node = 0
3      sqkfFastCommManager::ResourceSelfTuning entry [eduid 1 eduname db2sysc]
4      | sqlogmt entry [eduid 1 eduname db2sysc]
5      | sqlogmt exit
```

`test.flw` ファイルには、行 18 が存在しない点に注目してください。指定されたレコード ID が存在しない場合は、以下のようにエラー・メッセージが表示されます。

```
$ db2trc print -stack 18 test.flw
ERROR: Unable to find the Record ID 18 . Exiting.
```

グローバル・オプション・コマンド構文に関する情報については、コマンド構文図を参照してください。さらに詳しい情報は、以下の CF および CFCLI 使用例セクションの『例 4』を参照してください。

## CF および CFCLI 使用例

**例 6** `xport_common` (CF\_TRACE\_WARNING)、`srv_init`

(CF\_TRACE\_PATH|CF\_TRACE\_ERROR)、および srv\_common (CF\_TRACE\_ALL) の各コンポーネントに対してトレースをオンにするには、以下のコマンドを実行します。

```
db2trc cf on -m "*.CF.xport_common,srv_init,srv_common.*.CF_TRACE_WARNING,0x09,CF_TRACE_ALL"
```

このコマンドを実行した後、他のすべてのコンポーネントに対してトレースがオフになります。

- 例 7** すべての CF クライアント・コンポーネントに対して CF\_TRACE\_ALL レベルでトレースをオンにするには、以下のコマンドを実行します。

```
db2trc cfcli on
```

- 例 8** srv\_list (CF\_TRACE\_ERROR) コンポーネントと他のすべてのコンポーネントに対して CF\_TRACE\_ALL レベルでトレースをオンにするには、以下のコマンドを実行します。

```
db2trc cf on -m "*.CF.srv_list.*.CF_TRACE_ERROR,CF_TRACE_ALL"
```

- 例 9** すべての CF コンポーネントに対して CF\_TRACE\_WARNING または CF\_TRACE\_DEBUG レベルでトレースをオンにするには、以下のコマンドを実行します。

```
db2trc cf on -m "*.CF.*.*.CF_TRACE_WARNING|CF_TRACE_DEBUG"
```

- 例 10** db2nodes.cfg ファイルにリストされているすべての CF サーバーと、すべての CF コンポーネントに対して CF\_TRACE\_ALL レベルでトレースをオンにするには、以下のコマンドを実行します。

```
db2trc cf on
```

- 例 11** db2nodes.cfg ファイルにリストされている CF サーバー 129 のみと、すべての CF コンポーネントに対して CF\_TRACE\_ALL レベルでトレースをオンにするには、以下のコマンドを実行します。

```
db2trc cf on -id 129
```

## 使用上の注意

**db2trc** コマンドは、トレースを実行する中で何度か発行する必要があります。一般的な手順としては、DB2 インスタンスを停止させた状態でまずトレースを **on** にします。これにより、DB2 インスタンスの開始直後に、指定されたデータの収集とそのデータのバッファへの格納が開始されます。次に、データベースに接続する前にバッファをクリアし、バイナリ形式のデータをダンプ・ファイルにダンプします。その後、トレースを **off** にして、最後にダンプ・ファイルを容易に解釈できるテキスト宛先ファイルにフォーマットします。以下は、SAMPLE データベースのトレースを行うために実行されるコマンドの例です。トレース・バッファの内容は、ファイル dmp に書き込まれています。

```
db2trc on -i 8m -m ".*.2.*.*" -t
db2start
db2trc clear
db2 connect to sample
db2trc dump dmp
db2trc off
```

注: Windows オペレーティング・システムでは、**db2trc** コマンドは、同じインストール済みコピーに属するすべての DB2 インスタンスをトレースします。Linux および Unix オペレーティング・システムでは、DB2 インスタンスを個別にトレースできます。

**db2trc** コマンドの一般的な構文を以下の例に示します。コマンド・オプションは大きく分けて、収集 (collection) と構文解析 (parsing) の 2 つの段階に分類されます。

- *Collection* オプションには、トレースをオンまたはオフにすること、トレース・バッファのサイズを指定すること、トレース・オプションを指定または変更すること、トレースをダンプすること、トレース・バッファをクリアすることなどが含まれます。
- *Parsing* オプションには、フォーマットされたトレース・レコードを古い順、またはプロセス別、スレッド別にソートすることなどが含まれます。

### STAGE #1 - COLLECTION

Usage: db2trc [facility] <command> [-u]

```
[facility]
  db2 - DB2 instance (default)
  das - DB2 Administration Server instance

<command>
  change - Change trace options
  clear   - Clear the trace buffer
  dump    - Generate trace dump file
  info    - Information
  off     - Disable the trace facility
  on      - Enable the trace facility
  stop    - Stop tracing
```

### STAGE #2 - PARSING

Usage: db2trc <command> [-u]

```
<command>
  ccfmt   - Dump and format a code coverage trace
  flow    - Generate control flow diagram
  format  - Format
  info    - Information
  perfmt  - Format a performance trace
```

For more information add the "-u" option to any of the above commands

上記の『Stage #2 - Parsing』セクションのコマンド **ccfmt** は「コード・カバレッジ・トレース」をダンプおよびフォーマットします。コード・カバレッジ・トレースは **db2trc** を拡張したもので、関数の入り口、出口、プローブ・ポイント、およびコード・パスの数を保持します。これは、頻繁に使用されている関数や、テスト時にタッチされていない関数に関する統計を収集するために使用することができます。

データベース・サーバーのトレースを実行する際は、トレース機能をオフにしてからデータベース・マネージャーを開始することをお勧めします。これは、UNIX および Linux プラットフォーム上で実行されるデータベース・マネージャーがトレースの変更を即座に認識するための最も信頼できる方法です。

トレースを ON にして DB2 Text Search に固有の情報を受け取るには、以下のよう  
に cie (155) のコンポーネント・コードを持つマスクを使用できます。

```
db2trc on -m " *.*.155.*.*"
```

特定のデータベース・パーティションが問題に関連していることが分かっているなら、そのデータベース・パーティションだけをトレースしてください。オプション **db2trc on -member NN** を使用して、どのデータベース・パーティションがトレースされるか指定できます。

問題が特定のアプリケーション ID に関連しているなら、db2trc オプション **db2trc on -appid <appID>** を使用して、トレースをその特定のアプリケーション ID に制限できます。

## db2trcoff - db2trc のトレース OFF オプション

**db2trcoff** コマンドは、**db2trc** コマンドを **off** オプションを付けて発行します。このコマンドを使用すると、**db2trc** コマンドを使用する場合に比べて少ないコマンド数で簡単にダンプ、フロー、およびフォーマット・ファイルを生成できます。

**db2trcoff** コマンドは、**db2trc** コマンドのすべてのパラメーターをサポートします。

### 許可

UNIX オペレーティング・システムで DB2 インスタンスのトレースを行う場合は、以下のいずれかの権限が必要です。

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

UNIX オペレーティング・システムで DB2 Administration Server のトレースを行う場合は、以下の権限が必要です。

- *dasadm*

### コマンド構文

```
db2trcoff [ -flw ] [ -fmt ] [ -help ]
```

### コマンド・パラメーター

#### -flw

トレース・フロー・レポートを生成します。フロー・ファイルの名前は、`db2trcoff<time-stamp>.flw` です。<time-stamp> の値は、`hours_minutes_seconds` の形式でなければなりません。

#### -fmt

トレース・フォーマット・レポートを生成します。フォーマット・ファイルの名前は、`db2trcoff<time-stamp>.fmt` です。<time-stamp> の値は、`hours_minutes_seconds` の形式でなければなりません。

#### -help

ヘルプ情報を表示します。

### 例

**db2trc** コマンドをオフにして、フォーマット・ファイルとフロー・ファイルの両方を生成するには、次のコマンドを発行します。

```
db2trcoff -flw -fmt
```

### 使用上の注意

**db2trcoff** コマンドは、常にダンプ・ファイルを生成します。ダンプ・ファイルの名前は `db2trcoff<time-stamp>.dmp` です。<time-stamp> の値は、`hours_minutes_seconds` の形式でなければなりません。

## db2trcon - db2trc のトレース ON オプション

**db2trcon** コマンドは、**db2trc** コマンドを **on** オプションを付けて発行します。このコマンドには、**db2trc** コマンドの使用に役立ついくつかのオプションがあります。**db2trcon** コマンドは、**db2trc** コマンドを一定期間発行することができます。さらに、**db2trc** コマンドを、最も CPU を消費する EDUs に対してオンにして発行することもできます。

**db2trcon** コマンドは、**db2trc** コマンドのすべてのパラメーターをサポートします。

### 許可

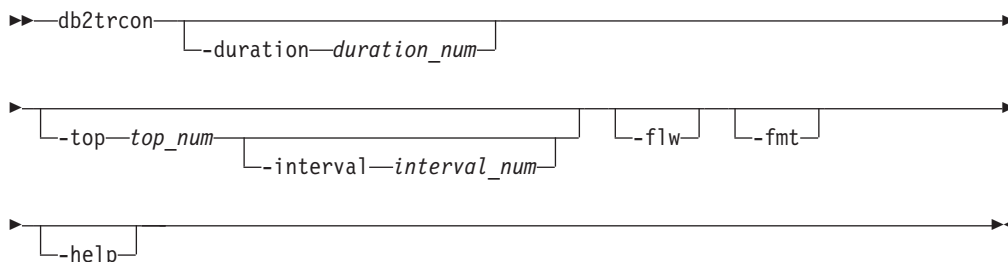
UNIX オペレーティング・システムで DB2 インスタンスのトレースを行う場合は、以下のいずれかの権限が必要です。

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

UNIX オペレーティング・システムで DB2 Administration Server のトレースを行う場合は、以下の権限が必要です。

- *dasadm*

### コマンド構文



### コマンド・パラメーター

#### **-duration** *duration\_num*

これを指定すると、指定時間 (秒単位) トレースがオンになります。

#### **-top** *top\_num*

これを指定すると、最も CPU を消費する EDUs に対してトレースがオンになります。トレースを使用可能にできる EDUs の最大数は 64 です。

#### **-interval** *-interval\_num*

最も CPU を消費する EDUs を収集する期間を秒単位で指定します。このオプションが指定されていない場合、デフォルトの期間は 10 秒です。

#### **-flw**

トレース・フロー・レポートを生成します。フロー・ファイルの名前は、`db2trcon.<time-stamp>.flw` です。<time-stamp> の値は、`hours_minutes_seconds` の形式でなければなりません。

## db2trcon - db2trc のトレース ON オプション

### -fmt

トレース・フォーマット・レポートを生成します。フォーマット・ファイルの名前は、db2trcon.<time-stamp>.fmt です。<time-stamp> の値は、hours\_minutes\_seconds の形式でなければなりません。

### -help

ヘルプ情報を表示します。

## 例

以下の例は、**db2trc** コマンドを 1 分間オンにします。

```
db2trcon -duration 60
```

デフォルト期間の 10 秒間、CPU を消費する上位 10 個の EDU に対して **db2trc** コマンドをオンにして、トレースを tmpDump.dmp に配置するには、次のコマンドを発行します。

```
db2trcon -top 10 -f tmpDump.dmp
```

以下の例は、**db2trc** コマンドを 45 秒間オンにして、プロセッサ時間を最も消費する上位 5 つの EDU に関して 15 秒間隔で収集します。**db2trc** コマンドをオフにすると、**db2trcon** によってダンプ・ファイル、フロー・ファイル、フォーマット・ファイルが生成されます。

```
db2trcon -duration 45 -top 5 -interval 15 -fodc -ff
```

## 使用上の注意

**db2trcon** コマンドに **-duration** パラメーターを付けて発行すると、常にダンプ・ファイルが生成されます。また **db2trcon** コマンドは、**db2trc** コマンドから **-f** パラメーターを指定しないで発行した場合にも、常にトレース・ダンプ・ファイルを生成します。ダンプ・ファイルの名前は db2trcon.<time-stamp>.dmp です。<time-stamp> の値は、hours\_minutes\_seconds の形式でなければなりません。



## db2unins - DB2 データベース製品、フィーチャー、または言語のアンインストール

1 つ以上の DB2 データベース製品、フィーチャー、または言語をアンインストールします。

**db2unins** は、インストール・メディアとシステム上の DB2 インストール・コピーの両方にあります。インストール・メディアから実行した場合、**-f**、**-l**、**-t**、および **-?** パラメーターのみ使用できます。DB2 インストール・コピーから実行した場合、**-f** 以外のすべてのオプションを使用できます。

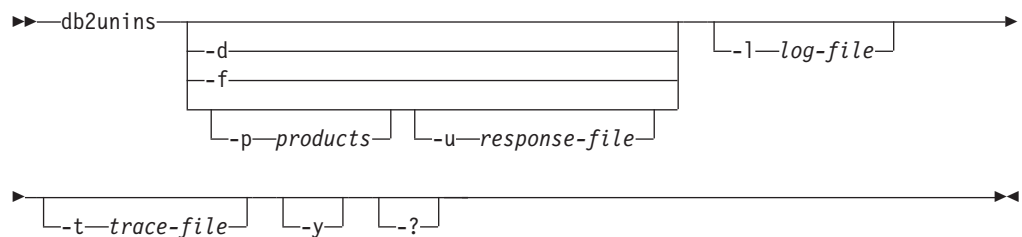
### 許可

SYSADM

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

注:

Windows オペレーティング・システムでは、**db2unins** コマンドで / または - の切り替え記号を交換可能なものとして使用できます。

**-?**、**-d**、**-p**、または **-u** パラメーターのいずれも指定しないで **db2unins** コマンドを実行すると、現行のインストール・ディレクトリーの下にあるすべての DB2 データベース製品が除去されます。

**-d** システム上の現行の DB2 コピーにインストールされている製品を表示します。このオプションは、DB2 データベース製品のインストール済みコピーから実行される場合にのみ使用できます。

**-f** システム上のすべての DB2 データベース製品の強制的なアンインストールを実行します。**db2unins -f** コマンドは、インストール・メディアからのみ発行できます。**db2unins -f** が正常に発行されると、システムがリポートします。これは、バージョン 9 より前のその他の DB2 製品がシステム上にインストールされていない場合にのみ発行できます。

**-p products**

アンインストールする製品を指定します。このパラメーターは、インストールされた DB2 コピーから実行する場合にのみ使用可能です。複数の製品を

アンインストールするには、製品のリストをセミコロンで分離し、二重引用符で囲みます。 **-p** と **-u** の両方のパラメーターを指定した場合、**-p** に指定された製品は、応答ファイルで指定された製品をオーバーライドします。例えば、`db2unins -p "ESE;QP" -u db2un.rsp` は、`db2un.rsp` 内の **REMOVE\_PROD** キーワード値に関係なく、DB2 ESE および QP の両方をアンインストールします。

### **-u response-file**

応答ファイルで指定された内容に基づいて、製品、フィーチャー、または言語のアンインストールを実行します。例えば、`db2unins -u db2un.rsp` とします。このパラメーターは、インストールされた DB2 コピーから実行する場合にのみ使用可能です。 **-p** と **-u** の両方のパラメーターを指定した場合、**-p** パラメーターに指定された DB2 製品は、応答ファイル内の **REMOVE\_PROD** キーワードをオーバーライドします。

クラスター環境を使用している場合、応答ファイルを使用して DB2 製品をアンインストールする前に、フェイルオーバー・インフラストラクチャーを作成するために当初 **db2mscs** コマンドを実行した同じサーバーから、まず **db2mscs** コマンドを **-u** オプションを指定して実行する必要があります。詳細については、**db2mscs** コマンドを参照してください。

### **-l log-file**

ログ・ファイルの位置を指定します。デフォルトのログ・ファイルの位置は、`My Documents\%DB2LOG%\db2un_<timestamp>.log` です。

### **-t trace-file**

トレース機能をオンにします。トレース・ファイルは、**db2unins** コマンドでの問題をデバッグするために使用されます。

**-y** アンインストール・プロセス中に確認が行われないようにします。

**-?** **db2unins** コマンドのヘルプを表示します。

## 使用上の注意

- Windows オペレーティング・システムでは、**db2unins** という名前のファイルはインストール・イメージのルート・ディレクトリで見つかります。また、インストール・パス `installPath\BIN` にもインストールされています。ただし、それらのファイル名は同じでも内容は同じではなく、動作も異なります。このファイルをインストール・イメージからコピーしてインストール・パス・ディレクトリに貼り付ける操作は行わないでください。この操作を行うと、問題が発生します。
- Windows オペレーティング・システムでは、絶対パスを指定する場合、パス名は二重引用符の中に含める必要があります。
- **db2unins -f** を使用してシステム上にあるすべての DB2 データベース製品を手動で削除する場合は、システム上の DB2 製品のバージョンのうち、最も高いバージョンのユーティリティーを使用してください。例えば、DB2 V9.1 の DB2COPY1 と、DB2 V9.5 の DB2COPY2 の 2 つのコピーがインストールされている場合、DB2 の 2 つのバージョンの両方を削除するには、DB2 V9.5 製品イメージから **db2unins -f** を実行します。DB2 V9.1 製品イメージから **db2unins -f** を実行した場合、マシンは完全にはクリーンアップされません。

## db2unins - DB2 データベース製品、フィーチャー、または言語のアンインストール

- Microsoft Cluster Service (MSCS) でクラスター化されたインスタンスがある場合、**db2mscs** または **db2iclus** コマンドを発行してインスタンスをクラスター解除してからアンインストールできます。

## db2untag - コンテナ・タグの解放

表スペース・コンテナの DB2 タグを除去します。

このタグは、DB2 が複数の表スペースで 1 つのコンテナを再利用できないようにするために使用します。コンテナが関連しているデータベースを識別して、コンテナ・タグについての情報を表示します。削除されたデータベースが最後に使用したコンテナを解放する必要があるときに便利です。タグが残されている場合、DB2 がそれ以後そのリソースを使用しないようにします。

**重要:** このツールは、経験のあるシステム管理者だけが使用してください。

### 許可

ユーザーには、データベースを作成した ID が所有する表スペースのコンテナへの読み取り/書き込みアクセス権が必要です。

### 必要な接続

なし

### コマンド構文

▶▶ db2untag [-f] filename ◀◀

### コマンド・パラメーター

*-f* db2untag コマンドが filename への操作を確認するプロンプトを出さないことを指定します。このオプションは、プロンプトや確認を表示しないで filename のタグを外すため、注意して使用する必要があります。

*filename*

DB2 タグが除去される、表スペース・コンテナの完全修飾名を指定します。

### 使用上の注意

データベースの作成、表スペースの作成または変更の操作から、SQLCODE -294 (使用中のコンテナでエラー) が戻されることがあります。通常それは、そのコンテナが別の表スペースによって既に使用中であるために、オペレーティング・システム・リソース名の指定エラーであることを示します。コンテナは、一度に 1 つの表スペースでしか使用できません。

最後にコンテナを使用したデータベースが削除されていることをシステムまたはデータベース管理者が検出した場合、コンテナのタグが除去されていなければ、db2untag ツールを使用できます。そのコンテナを解放する場合は、以下のアクションのどちらかを実行します。

- SMS コンテナの場合、適切な削除コマンドを使用して、ディレクトリーとそのコンテナを除去します。
- DMS ロー・コンテナの場合、ファイルまたは装置を削除するか、または db2untag でコンテナ・タグを除去します。ツールは、それ以外の点では DMS コンテナに何の修正も加えません。

## db2updserv - 製品の更新の表示

DB2 データベース製品に使用可能な製品の更新および機能強化を表示します。  
Windows では、このコマンドの出力先は Web ページとなり、UNIX では Java アプリケーションです。

### 許可

なし

### 必要な接続

インターネット接続が必要です。

UNIX および Linux システムで、HTTP プロキシ・サーバーを介してインターネットに接続する場合、**db2updserv** コマンドを実行する前に、HTTP プロキシ・サーバーのホスト名とポートを `DB2DIR</varname>/java/jdk64/jre/lib/net.properties` ファイルに指定する必要があります。詳しくは、使用上の注意のセクションを参照してください。

### コマンド構文

▶▶—db2updserv—◀◀

### コマンド・パラメーター

なし

### 使用上の注意

- UNIX および Linux システムの場合のみ

HTTP プロキシ・サーバーを使用するときは、**db2updserv** コマンドを使用する前に、HTTP プロキシのホスト名とポートをプロパティ・ファイルに設定する必要があります。HTTP プロキシ・サーバーのホスト名とポートをセットアップするには、`DB2DIR/java/jdk64/jre/lib/net.properties` ファイルを編集します。ここで、`DB2DIR` は、DB2 データベース製品の現行バージョンがインストールされている場所です。以下のパラメーターを含むようにファイルを編集します。

```
http.proxyHost=host name  
http.proxyPort=port number
```

ここで、*host name* はご使用の HTTP プロキシ・サーバーのホスト名で、*port number* はご使用の HTTP プロキシのポート番号です。

以下に例を示します。

```
http.proxyHost=proxy.mycompany.com  
http.proxyPort=80
```

他の Java アプリケーションが、HTTP プロキシ・サーバーを介してではなく他のサーバーに直接接続する必要がある場合には、`nonProxyHosts` パラメーターにサーバー・ホスト名を指定してください。

## db2val - DB2 コピー妥当性検査ツール

インストール・ファイル、インスタンス・セットアップ、およびローカル・データベース接続の状態をチェックして、DB2 コピーの基本機能を検証します。

### 許可

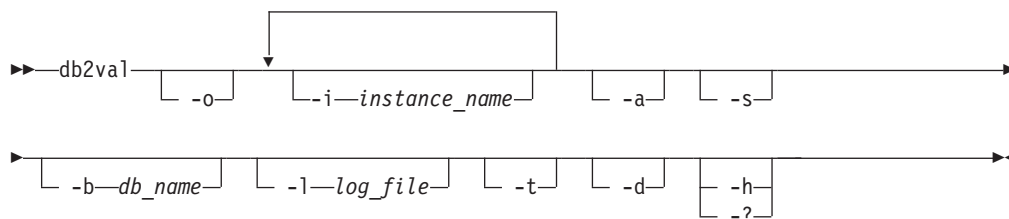
インスタンスの妥当性検査には、以下の権限のいずれかが必要です。

- root のコピーでは、Linux および UNIX オペレーティング・システムで root 権限が必要になります。
- SYSADM に加えて、以下の権限のいずれかが必要になります。
  - インスタンス所有者
  - Linux および UNIX オペレーティング・システム上の root アクセス、または Windows オペレーティング・システム上のローカル管理者権限

### 必要な接続

なし。

### コマンド構文



### コマンド・パラメーター

- o インストール・ファイルのみ妥当性検査されることを指定します。インスタンス、データベース、および拡張セキュリティの妥当性検査は実行されません。このパラメーターが指定される場合、-i、-a、-b、および -s パラメーターは無視されます。
- i *instance\_name* 妥当性検査するインスタンスの名前を指定します。複数のインスタンスを妥当性検査することを指定するには、このパラメーターを複数回指定します。例えば、-i inst1 -i inst2 とします。Windows オペレーティング・システムでは、このパラメーターが指定されない場合、現行のインスタンスがデフォルト値として使用されます。Linux および UNIX オペレーティング・システムでは、このパラメーターは、DB2 コピーの root インストールで root ユーザーのみが使用できます。
- a DB2 コピーのすべてのインスタンスを妥当性検査します。Linux および UNIX オペレーティング・システムでは、このパラメーターは、DB2 コピーの root インストールで root ユーザーのみが使用できます。このパラメーターは、パラメーター -i をオーバーライドします。

**-b <db\_name>**

データベースの作成および指定されたデータベースへの接続を妥当性検査します。アクティブな DB2 インスタンスのみが妥当性検査の対象となり、DB2 クライアントおよび DB2 pureScaleのインスタンスの場合はこのパラメーターは無視されます。

**-t <trace\_file>**

このパラメーターは、Linux および UNIX オペレーティング・システムにのみ適用されます。 trace\_file で指定するトレース・ファイルの絶対パスおよび名前を指定します。

**-d**

このパラメーターは推奨されなくなりました。今後のリリースでは除去される可能性があります。代わりに **-t** パラメーターを使用してください。Linux および UNIX オペレーティング・システムでのみ有効です。このパラメーターは、DB2 サポートからの指示があった場合にのみ使用してください。デバッグ・モードをオンにします。

**-s**

パーティション・データベース環境の一部である指定されたインスタンスについて DB2 データベース・マネージャーを開始します。

**-l <log\_file>**

名前を指定したファイルにログを書き込みます。-l パラメーターが指定されていない場合は、Linux および UNIX オペレーティング・システムでのデフォルト・ログ・パスは /tmp/db2valxx.log で、Windows オペレーティング・システムでは My Documents¥DB2LOG¥db2valxx.log です。ここで、xx は生成値です。

**-? | -h db2val** コマンドの使用方法の情報が表示されます。

**例**

インスタンス TEST1 およびデータベース DATA1 を妥当性検査するには、以下のコマンドを実行します。

```
db2val -i TEST1 -b DATA1
```

DB2 コピーのすべてのインスタンスを妥当性検査するには、以下のコマンドを実行します。

```
db2val -a
```

DB2 インストール・ファイルのみ妥当性検査するには、以下のコマンドを実行します。

```
db2val -o
```

**使用上の注意**

db2val コマンドは、DB2 pureScale 環境におけるインストール・ファイルの妥当性検査にのみ使用できます。

## db2xdbmig - XSR オブジェクトのマイグレーション

分解用に使用可能になっているすべての XML スキーマ・リポジトリ (XSR) オブジェクトを、コマンドを実行する DB2 コピーの現行バージョンおよびサービス・レベルにまでマイグレーションします。

このコマンドは、*DB2DIR/bin* ディレクトリにあります (*DB2DIR* は DB2 データベース・システムの現行バージョンがインストールされているインストール・ロケーション)。

### 許可

データベース内のすべての XSR オブジェクトに対する CREATE、ALTER、および DROP 特権。

### コマンド構文

▶—db2xdbmig—*database-alias*—————▶

### コマンド・パラメーター

*database-alias*

XSR オブジェクトが入っているデータベースの別名を指定します。

### 使用上の注意

- **db2xdbmig** コマンドは、分解が可能な XML スキーマにしか影響しません。
- DB2 バージョン 9.1 GA またはフィックスパック 1 コピーから DB2 バージョン 9.7 へのアップグレードを行う場合、**UPGRADE DATABASE** コマンドは暗黙的に **db2xdbmig** コマンドを実行します。このコマンドを DB2 バージョン 9.7 で実行する必要はありません。



## db2xprt - トラップ・ファイルのフォーマット

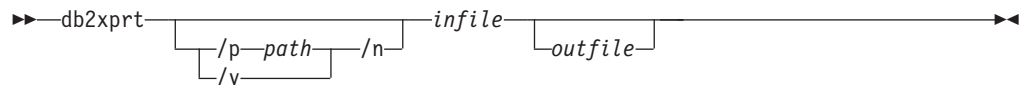
DB2 データベース・バイナリー・トラップ・ファイルを可読 ASCII ファイルにフォーマットします。

トラップ・ファイル (\*.TRP) は、デフォルトではインスタンス・ディレクトリー (DB2INSTPROF) にあります。 **diagpath** データベース・マネージャー構成パラメーターが設定されている場合は、診断データ・ディレクトリー・パスにあります。これは SQLLIB/BIN ディレクトリーの下にあります。 **db2xprt** コマンドは、トラップ・ファイルをフォーマットするために、DB2 シンボル・ファイル (.PDB) を使用します。

### 許可

**diagpath** ディレクトリーに対するアクセス権限を持っている必要があります。

### コマンド構文



### コマンド・パラメーター

#### */p path*

バイナリー・ファイルおよび PDB ファイルが置かれている位置 (複数可) を指す、セミコロン (;) で区切られたパス。

*/v* バージョン情報を表示します。

*/n* 行番号情報に関係なく、データをフォーマットします。

*infile* 入力ファイルを指定します。

*outfile* 出力ファイルを指定します。

### 例

DB30882416.TRP というトラップ・ファイルが **diagpath** に作成されている場合、それを以下のようにフォーマットできます。

```
db2xprt DB30882416.TRP DB30882416.FMT
```

## disable\_MQFunctions - WebSphere MQ 関数の無効化

指定したデータベースについて DB2 WebSphere MQ 関数の使用を無効にします。

### 許可

以下の権限のいずれか。

- SYSADM
- DBADM
- データベースに対する IMPLICIT\_SCHEMA (関数の暗黙または明示のスキーマ名が存在しない場合)
- スキーマに対する CREATEIN 特権 (スキーマ名 DB2MQ または DB2MQ1C が存在する場合)

### コマンド構文

```

▶▶ disable_MQFunctions -n database -u userid -p password

```

-v オプションの値のツリー図:
 

- v
  - all
  - 0pc
  - 1pc

### コマンド・パラメーター

- n *database*  
データベースの名前を指定します。
- u *userid*  
データベースに接続するためのユーザー ID を指定します。
- p *password*  
ユーザー ID のパスワードを指定します。
- v オプション。これは、トランザクション・ユーザー定義関数と非トランザクション・ユーザー定義関数のサポートのために使用します。値は all、0pc、または 1pc のいずれかです。0pc を指定すると、無効にする操作によってスキーマ db2mq が削除されます。1pc を指定すると、無効にする操作によってスキーマ db2mq1c が削除されます。all を指定すると、無効にする操作によって両方のスキーマ (db2mq と db2mq1c) が削除されます。このオプションを指定しない場合、無効にする操作のデフォルトは all オプションになります。

### 例

以下の例では、データベース SAMPLE について DB2MQ と DB2MQ1C の関数を無効にします。

```
disable_MQFunctions -n sample -u user1 -p password1
```

## doce\_deinstall - DB2 インフォメーション・センターのアンインストール

**doce\_deinstall** ツールと同じインストール・パスにある DB2 インフォメーション・センターをアンインストールします。このコマンドは Linux オペレーティング・システムでのみ使用できます。

**doce\_deinstall** コマンドは *DB2DIR/install* にあります。*DB2DIR* は、DB2 インフォメーション・センターの現行バージョンがインストールされている場所です。**doce\_deinstall** コマンドは DOCE DVD から使用可能です。DVD から実行する場合、**doce\_deinstall** コマンドには **-b** オプションが必要です。

### 許可

root ユーザー権限

### 必要な接続

なし

### コマンド構文

```

▶▶ doce_deinstall --a [ -b installpath ] [ -l log-file ]
▶ [ -t trace-file ] [ -r response_file ] [ -h ] [ -? ]

```

### コマンド・パラメーター

- a**      インフォメーション・センターを現在場所から除去します。
- b**      このオプションは、DB2 データベース製品メディアからコマンドを実行する場合に有効です。アンインストールする DB2 データベース製品がインストールされている場所の絶対パスを指定します。このオプションが指定されない場合、コマンドはパスを求めるプロンプトを出します。
- l log-file**  
ログ・ファイルを指定します。デフォルト・ログ・ファイルは /tmp/doce\_deinstall.log\$\$ です。\$\$ はプロセス ID です。
- t trace-file**  
デバッグ・モードをオンにします。デバッグ情報は、*trace-file* として指定されたファイル名に書き込まれます。
- r response\_file**  
応答ファイルを使用して インフォメーション・センター を除去します。例えば、doce\_deinstall -r db2un.rsp。 **-a** パラメーターと組み合わせることはできません。
- h | -?**  
使用法情報を表示します。

## doce\_deinstall - DB2 インフォメーション・センターのアンインストール

### 例

/opt/ibm/db2/doce にインストールされている DB2 インフォメーション・センターをアンインストールするには、以下を発行します。

```
cd /opt/ibm/db2/doce
doce_deinstall -a
```

## doce\_install - DB2 インフォメーション・センターのインストール

DB2 インフォメーション・センターをインストールします。パスが指定されない場合、DB2 インフォメーション・センター はデフォルトで /opt/ibm/db2ic/V10.1 にインストールされます。このコマンドは Linux オペレーティング・システムにのみ適用されます。

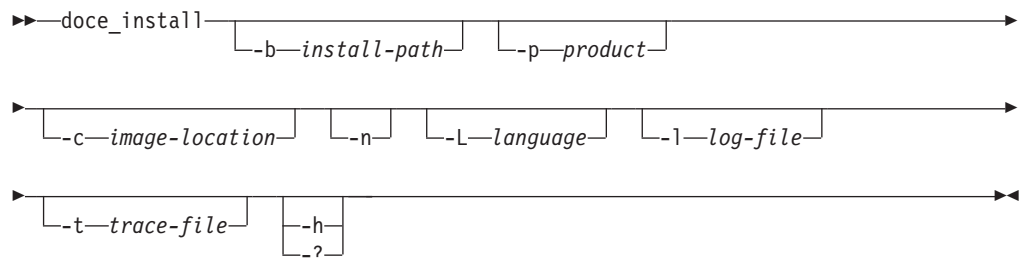
### 許可

root ユーザー権限

### 必要な接続

なし

### コマンド構文



### コマンド・パラメーター

#### **-b** *install-path*

DB2 インフォメーション・センターをインストールするパスを指定します。 *install-path* は絶対パス名にする必要があり、その最大長は 128 文字に制限されます。デフォルトのインストール・パスは /opt/ibm/db2ic/V10.1 です。このパラメーターは、 **-n** パラメーターが指定されるときには必須です。

#### **-p** *productID*

DB2 インフォメーション・センターの *productID* を指定します。 *productID* は接頭部として DB2 を必要としません。このパラメーターは、 **-n** パラメーターが指定されるときには必須です。

#### **-c** *image-location*

製品イメージ位置を指定します。複数のイメージ位置を示すには、このパラメーターを複数回指定します。例えば、 **-c** CD1 **-c** CD2 とします。このパラメーターは、 **-n** パラメーターが指定されている場合、インストールに複数の CD が必要な場合、および自動ディスクバリー用にイメージがセットアップされていない場合にのみ、必須です。それ以外の場合、次の CD が必要なときに、その位置を指定するように指示されます。

**-n** 非対話式モードを指定します。

#### **-L** *language*

各国語サポートを指定します。デフォルトは English です。複数の言語を同

## doce\_install - DB2 インフォメーション・センターのインストール

時にインストールするには、このパラメーターを複数回指定できます。例えば、英語とドイツ語の両方をインストールするには、`-L EN -L DE` と指定します。

### **-l** *log-file*

ログ・ファイルを指定します。デフォルト・ログ・ファイルは `/tmp/doce_install.log$$` です。\$\$ はプロセス ID です。

### **-t** *trace-file*

デバッグ・モードをオンにします。デバッグ情報は、*trace-file* として指定されたファイル名に書き込まれます。

### **-h | -?**

使用法情報を表示します。

## 例

- `/mnt/cdrom` 内のイメージからインストールし、すべての必要な入力のためのプロンプトが出されるようにするには、以下のコマンドを発行します。

```
cd /mnt/cdrom
./doce_install
```

- DB2 インフォメーション・センター を `/mnt/cdrom` 内のイメージから `/db2/V10.1` に英語で非対話式にインストールするには、以下のコマンドを発行します。

```
cd /mnt/cdrom
./doce_install -p doce -b /db2/V10.1 -n
```

## enable\_MQFunctions - WebSphere MQ 関数の有効化

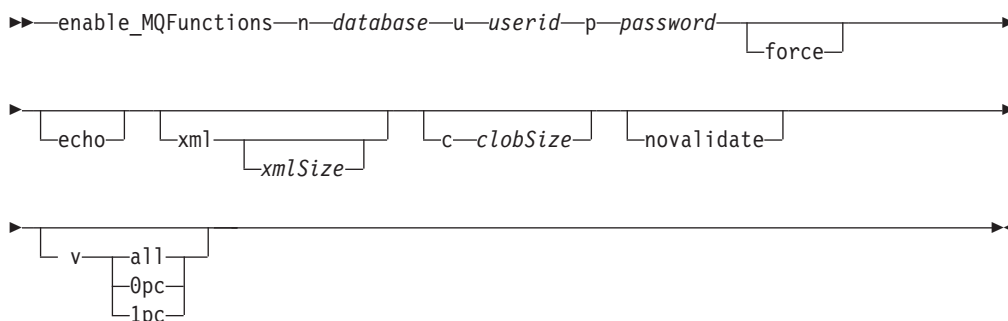
指定したデータベースについて DB2WebSphere MQ 関数を有効にし、DB2WebSphere MQ 関数を正常に実行できるかどうかを確認します。 WebSphere MQ と WebSphere MQ AMI のインストールと構成が行われていない場合、このコマンドは失敗します。

### 許可

以下の権限のいずれか。

- データベースに対する CREATE\_EXTERNAL\_ROUTINE 権限、および以下の少なくとも 1 つの権限
  - データベースに対する IMPLICIT\_SCHEMA 権限 (関数のスキーマ名が既存のスキーマを指していない場合)
  - スキーマに対する CREATEIN 特権 (スキーマ名 DB2MQ または DB2MQ1C が存在する場合)
- DBADM

### コマンド構文



### コマンド・パラメーター

**-n database**

有効にするデータベースの名前を指定します。

**-u userid**

データベースに接続するためのユーザー ID を指定します。

**-p password**

ユーザー ID のパスワードを指定します。

**-force** オプション。このオプションを使用すると、このユーティリティー・プログラムは既存の MQ UDF を無視します。言い換えれば、このプログラムは既存の関数をドロップしてから MQ UDF を再作成します。このオプションを指定しない場合、このコマンドは、MQ UDF が既に存在することを検出した時点で処理を終了します。

**-xml xmlSize**

オプション。これは、0pc 関数の XML バージョンを定義するためのものです。 **-v 1pc** オプションが指定されている場合、このオプションは無効です。

## enable\_MQFunctions - WebSphere MQ 関数の有効化

*xmlSize* は XML データの長さを指定します。最小の長さは 1 バイトです。最大の長さは 100M です。デフォルトは 1M です。この長さを *n* (バイト数)、*nK* (キロバイトでの長さ)、または *nM* (メガバイトでの長さ) として指定できます。

### -c *clobSize*

オプション。CLOB データの長さを指定します。最小の長さは 1 バイトです。これがデフォルトです。最大の長さは 100M です。この長さを *n* (バイト数)、*nK* (キロバイトでの長さ)、または *nM* (メガバイトでの長さ) として指定できます。

### -novalidate

オプション。DB2 MQSeries® 関数の妥当性検査を行わないことを指定します。

-v オプション。これは、トランザクション・ユーザー定義関数と非トランザクション・ユーザー定義関数のサポートのために使用します。値は *a11*、*0pc*、または *1pc* のいずれかです。*0pc* を指定すると、有効にすることによってスキーマ *db2mq* が作成されます。*1pc* を指定すると、有効にすることによってスキーマ *db2mq1c* が作成されます。*a11* を指定すると、有効にすることによってユーザー定義関数の下のすべてのスキーマ (*db2mq* と *db2mq1c*) が作成されます。このオプションを指定しない場合、有効にする処理のデフォルトは *a11* オプションになります。

-echo オプション。UDF の作成に使用した詳細な SQL または診断情報をプリントします。

## 例

以下の例は、トランザクション・ユーザー定義関数と非トランザクション・ユーザー定義関数を有効にします。ユーザーはデータベース *SAMPLE* に接続します。

```
enable_MQFunctions -n sample -u user1 -p password1
```

次の例では、ユーザーはデータベース *SAMPLE* に接続します。この例は、スキーマ *DB2MQ1C* によって *DB2MQ1C* 関数を作成します。

```
enable_MQFunctions -n sample -u user1 -p password1 -v 1pc
```

## 使用上の注意

DB2 MQ ユーザー定義関数は、このコマンドによって自動的に作成されるスキーマ *DB2MQ* または *DB2MQ1C* の下で実行されます。このコマンドを実行する前に、次のようにしてください。

- WebSphere MQ と WebSphere Application Messaging Interface (AMI) がインストールされていることと、WebSphere MQ のバージョンが 5.1 以上であることを確認します。
- 環境変数 *\$AMT\_DATA\_PATH* が定義されていることを確認します。
- トランザクション MQ UDF を使用する場合は、フェデレーテッド操作にデータベースが構成されていることを確認します。そのためには、以下のコマンドを実行します。

```
update dbm cfg using federated yes
```

- *DB2PATH* の *cfg* サブディレクトリーに移動します。



UNIX の場合:

- **db2set** を使用して、**DB2ENVLIST** に **AMT\_DATA\_PATH** を追加します。
- UDF 実行に関連するユーザー・アカウントが **mqm** グループのメンバーであることを確認します。
- このコマンドを呼び出すユーザーが **mqm** グループのメンバーであることを確認します。

注: AIX 4.2 は、MQSeries 5.2 でサポートされていません。

### installDSDriver - IBM Data Server Driver コンポーネントの抽出

IBM Data Server Driver (ds ドライバー) コンポーネントをインストールします。また、db2profile ファイルおよび db2cshrc ファイルを作成します。

Linux および UNIX オペレーティング・システムで、**installDSDriver** コマンドは、指定したディレクトリーで IBM Data Server Driver のすべてのコンポーネントをインストールおよびアップグレードし、抽出された .tar ファイルを削除します。

DB2 V10.1 フィックスパック 1 以降のフィックスパックでは、**installDSDriver** コマンドを実行すると、db2profile ファイル (Bash または Korn シェル用) および db2cshrc ファイル (C シェル用) が作成されます。db2profile ファイルおよび db2cshrc ファイルは、IBM Data Server Driver Package ソフトウェアのルート・インストール・パス (*INSTALL\_PATH*) の下に作成されます。

DB2 V10.1 フィックスパック 2 以降のフィックスパックでは、64 ビットの IBM Data Server Driver Package ソフトウェアから **installDSDriver** コマンドを実行すると、IBM Data Server Driver Package ソフトウェアのルート・インストール・パス (*INSTALL\_PATH*) 下に以下のスクリプト・ファイルが作成されます。

- 64 ビットの Bash または Korn シェル環境用の db2profile ファイル。
- 32 ビットの Bash または Korn シェル環境用の db2profile32 ファイル。
- 64 ビットの C シェル環境用の db2cshrc ファイル。
- 32 ビットの C シェル環境用の db2cshrc32 ファイル。

#### 許可

なし

#### コマンド構文

```
▶▶ installDSDriver [-upgrade <existing-dsdriver-path>] [-h] ▶▶
```

#### コマンド・パラメーター

**-h** 使用法情報を表示します。

**-upgrade** <existing-dsdriver-path>

<existing-dsdriver-path> パスで指定された既存の IBM Data Server Driver Package ソフトウェア・インストールをアップグレードします。以下に示すような、すべてのユーザー作成ファイルは、既存の IBM Data Server Driver Package ソフトウェアのインストール・パスに保存されています。

- db2cli.ini ファイル。
- db2dsdriver.cfg ファイル。
- DB2 Connect ライセンス・ファイル。

**-upgrade** <existing-dsdriver-path> オプションが指定されていない場合、**installDSDriver** スクリプトは IBM Data Server Driver を新規インストールとしてインストールします。

**-upgrade** <existing-dsdriver-path> オプションは、IBM Data Server Driver Package ソフトウェア バージョン 10.1 フィックスパック 2 以降のフィックスパックで使用可能です。

### 使用上の注意

- バージョン 10.1 フィックスパック 1 では、64 ビットの IBM Data Server Driver Package ソフトウェアの **installDSDriver** スクリプトを実行すると、64 ビット・ドライバーがインストールされ、64 ビット tar ファイルが削除されますが、パッケージ内の 32 ビット tar ファイルおよびディレクトリーはそのままになります。
- バージョン 10.1 フィックスパック 2 以降のフィックスパックでは 64 ビットの IBM Data Server Driver Package ソフトウェアの **installDSDriver** スクリプトを実行すると、32 ビットと 64 ビット両方のドライバーがインストールされ、32 ビットと 64 ビット両方の tar ファイルが削除されます。
- 32 ビットの IBM Data Server Driver Package ソフトウェア・インストールの場合、db2profile ファイルは Bash または Korn シェル用であり、db2cshrc ファイルは C シェル用です。

インストール・パスから db2profile スクリプトまたは db2cshrc スクリプトを実行するには、次のコマンドを発行します。

- Korn シェル環境または Bash シェル環境:

```
$ ./db2profile
```

- C シェル環境:

```
source ./db2cshrc
```

64 ビットの IBM Data Server Driver Package ソフトウェアをインストールした後で 32 ビットのドライバーを構成する場合は、以下のインストール・パスから db2profile32 または db2cshrc32 スクリプトを実行します。

- Korn シェル環境または Bash シェル環境:

```
$ ./db2profile32
```

- C シェル環境:

```
source ./db2cshrc32
```

## installFixPack - インストール済み DB2 データベース製品の更新

すべての UNIX および Linux プラットフォームにおいて、指定された位置にインストールされた DB2 データベース製品をイメージと同じレベルに更新します。マルチコピー DB2 データベース製品がインストールされている場合、**installFixPack** コマンドにより、指定されたパスに従って、一度に 1 コピーが更新されます。

このコマンドはイメージのトップ・ディレクトリーに置くことができます。

フィックスパックのインストール・パスに関連したすべてのインスタンス（およびパーティション・データベース環境内のすべてのデータベース・パーティション）のデータベース・マネージャー（DBM）が停止され、DB2 ライブラリーすべてがアンロードされていると、フィックスパックのインストールが続行します。前提条件がすべて満たされると、**installFixPack** はインストール・パスに関連するこれらのインスタンスと DAS を更新します。手動による追加更新は不要です。すべての UNIX および Linux オペレーティング・システムでは、データベースが再接続されると、またはアプリケーションが再開されると、**djxlink** バインド・コマンドが自動的に起動します。

場合によっては、別の強制オプションを指定してフィックスパックのインストールを続行することもできます。例えば、すべての DBM を停止させるわけではない場合や、DB2 ライブラリーがロードされたままになっている場合などです。

**installFixPack** は続行しますが、インスタンスと DAS の更新、およびアプリケーションの再開を手動で行うことが必要な場合があります。

パーティション・データベース環境インスタンスの場合、インスタンスの更新はインスタンス所有データベース・パーティションでのみ必要ですが、フィックスパックのインストールはすべてのデータベース・パーティションで行ってください。更新後もインスタンスの全機能を維持するには、全部の製品とフィーチャーをすべてのデータベース・パーティションに、少なくともインスタンス所有データベース・パーティションにインストールすることをお勧めします。

フィックスパックを dsf インスタンス・タイプ（つまり、DB2 pureScale インスタンス内の DB2 コピーに関連付けられている DB2 pureScale 環境）に適用する場合には、フィックスパックを新しいパスにインストールする必要があります。フィックスパックをオリジナルのインストール・パスに適用することはできません。フィックスパックがインストールされるパスは、既存のインストール済み環境とは異なってなければなりません。加えて、この新しいパスは、すべてのホスト上で同一でなければなりません。フィックスパックを新しい場所にインストールするときは、フィックスパックをインストールする前に DB2 pureScale インスタンスを停止する必要はありません。

フィックスパックを ese インスタンス・タイプに適用する場合には、フィックスパックをオリジナル・インストール・パス上に重ねて適用するか、またはフィックスパックを新しいパスにインストールすることができます。フィックスパックをオリジナル・インストール・パス上に重ねてインストールする場合には、フィックスパックを適用する前にインスタンスを停止する必要があります。詳しくは、「DB2 サ

## installFixPack - インストール済み DB2 データベース製品の更新

「オーバー機能 インストール」の『バージョン 9.8 ese または dsf インスタンス・タイプをアップデートするためのフィックスパックのインストール』を参照してください。

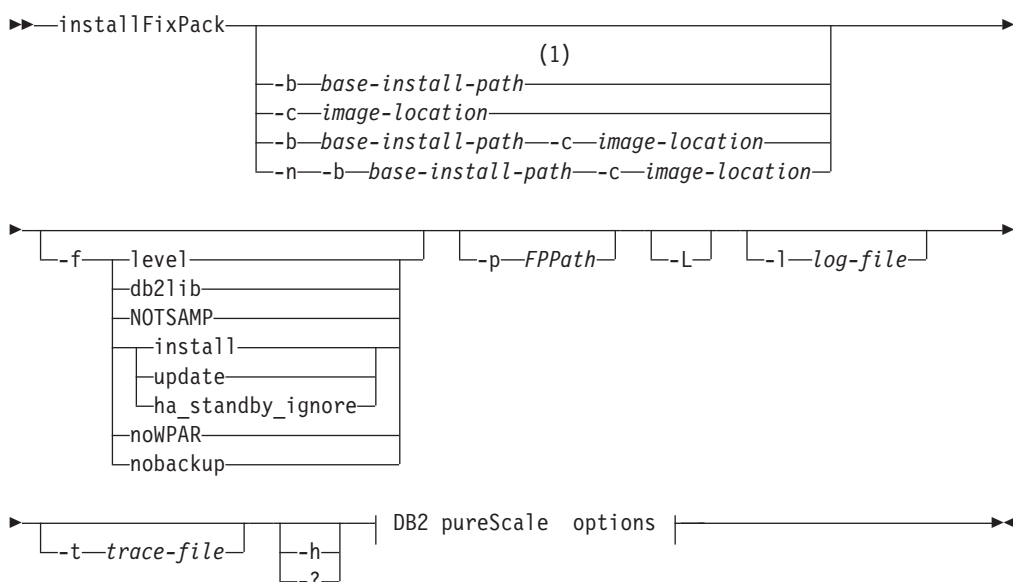
### 許可

root インストールには root ユーザー権限が必要です。非 root インストールの場合、非 root インストールを所有するユーザー ID でログオンしなければなりません。

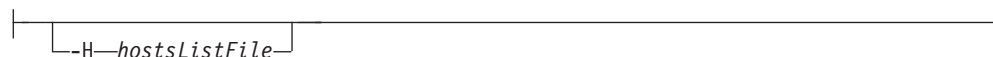
### 必要な接続

なし

### コマンド構文



### DB2 pureScale options:



### 注:

- 1 このオプションを省略すると、エラー・メッセージによって一時停止することなく必要な情報を求めるプロンプトが出されます。

### コマンド・パラメーター

- `-n` 非対話式モードを指定します。指定する場合は、`-b`、`-p`、および `-c` も指定しなければなりません。このモードは、コマンド行で必要とされる情報すべてを提供する無人モードのアプリケーションに使用できます。

## installFixPack - インストール済み DB2 データベース製品の更新

### **-b** *base-install-path*

DB2 データベース製品をインストールするパスを指定します。**-n** が指定される場合には必須です。パスの長さは 128 文字に制限され、絶対パス名です。

**-b** オプションは、DB2 製品の非 root インストールでは不要ですが、root インストールの場合には必須です。非 root インストールで **-b** を使用する場合、*base-install-path* の値はユーザーの *HOME/sql11ib* ディレクトリーでなければなりません。そうでない場合、パスは無効と見なされます。非 root インストールで **-b** が指定されていない場合、DB2 インストーラーはユーザーの *HOME/sql11ib* をインストール・パスとして使用して続行します。ただし **-b** を使用する場合も、指定されたインストール・パスに root ユーザーによって DB2 コピーがインストールされている場合は、DB2 コピーを更新できるのはそれをインストールしたユーザーだけであるため、そのインストール・パスは無効と見なされます。

### **-c** *NLPACK\_location*

関連する DB2 National Language Pack (NLPACK) のロケーションを指定します。このパラメーターは、**-n** が指定されるときには必須です。以下の条件がすべて満たされている場合は、DB2 NLPACK の場所を明示的に指定する必要があります。

- **-n** オプションが指定されている。
- インストールには、各国語 (英語以外) サポートが必要。
- DB2 NLPACK が DB2 DVD にも、インストールしている DB2 データベース製品と同じサブディレクトリー上にもない。

**-f** このパラメーターは、DB2 pureScale 環境では無視されます。強制オプション。引数のない **-f** はサポートされません。以下の強制引数を組み合わせて使用できます。例えば、**-f level -f db2lib** となります。

### **-f level**

下位レベルまたは同レベルのフィックスパック・インストールを強制します。フィックスパック・イメージがインストール済みの DB2 データベース製品より高位になっている場合、このオプションは無視されます。

### **-f db2lib**

このパラメーターは、DB2 pureScale 環境では無視されます。DB2 ライブラリーのロードについての検査を迂回するよう **installFixPack** を強制します。インストール後にインスタンスやアプリケーションが正常に作動するようにするには、すべての関連するインスタンス (関連するパーティション・データベース環境インスタンスのすべてのデータベース・パーティションを含む) に対して DBM を停止し、インストール・パスに関連するすべての DB2 ライブラリーをアンロードする必要があります。

### **-f NOTSAMP**

このパラメーターは、DB2 pureScale 環境では無視されます。SA MP を更新しないように指定します (root インストールにのみ適用)。

### **-f install**

**installFixPack** コマンドにおいて、DB2 ライブラリーのロードの検査と、インスタンスおよび DAS が停止されていることの検査をすべてバイパスするように強制します。インストール後にインスタンスやアプリケーションが

## installFixPack - インストール済み DB2 データベース製品の更新

正常に作動するようにするには、すべての関連するインスタンス (関連するパーティション・データベース環境インスタンスのすべてのデータベース・パーティションを含む) に対して DBM を停止し、インストール・パスに関連するすべての DB2 ライブラリーをアンロードする必要があります。このオプションを指定すると、インスタンスと DAS はいずれも更新されません。インストール後、インスタンスと DAS を手動で更新する必要があります。また、update、install、ha\_standby\_ignore の各オプションは相互に排他的なので、同じインストールに対して指定することはできません。

### -f update

**installFixPack** コマンドにおいて、DB2 ライブラリーのロードの検査と、インスタンスおよび DAS が停止されていることの検査をすべてバイパスするように強制します。インストール後にインスタンスやアプリケーションが正常に作動するようにするには、すべての関連するインスタンス (関連するパーティション・データベース環境インスタンスのすべてのデータベース・パーティションを含む) に対して DBM を停止し、インストール・パスに関連するすべての DB2 ライブラリーをアンロードする必要があります。このオプションを指定すると、インスタンスと DAS は両方とも更新されます。また、オプション update、install、および ha\_standby\_ignore は相互に排他的であるため、同じインストールに対して指定することはできないことに注意してください。

### -f ha\_standby\_ignore

このパラメーターは、**sql1lib** ディレクトリーの検査をバイパスするように **installFixPack** コマンドに強制することによって、この検査を無視します。例えば、**sql1lib** ライブラリーがスタンバイ・ノードにはマウントされないクラスター環境で、**sql1lib** ディレクトリーの検査を無視します。インストール後にインスタンスやアプリケーションが正常に作動するようにするには、すべての関連するインスタンス (関連するパーティション・データベース環境インスタンスのすべてのデータベース・パーティションを含む) に対して DBM を停止し、インストール・パスに関連するすべての DB2 ライブラリーをアンロードする必要があります。このオプションを指定すると、インスタンスと DAS はいずれも更新されません。インストール後、インスタンスと DAS を手動で更新する必要があります。また、update、install、ha\_standby\_ignore の各オプションは相互に排他的なので、同じインストールに対して指定することはできません。

### -f noWPAR

このパラメーターは、DB2 pureScale 環境では無視されます。グローバル環境の AIX 6.1 以降に適用されます。グローバル環境で更新中の DB2 コピーを共有する AIX システム・ワークロード・パーティション (WPAR) に対して検査またはアクションを実行しないよう **installFixPack** に強制します。-f noWPAR が指定されている場合、この DB2 コピーを共有する各システム WPAR 上でインスタンスおよび DAS を手動で更新する必要があります。

### -f nobackup

このパラメーターは、DB2 pureScale 環境では無視されます。コンポーネントの更新時に、**installFixPack** に強制的にインストール・ファイルをバックアップさせないようにします。ファイルをバックアップしないことを選択すると、インストール・ディレクトリーのスペース所要量が削減されま



## installFixPack - インストール済み DB2 データベース製品の更新

す。ただし、ファイルをバックアップしないことを選択すると、エラーが発生した場合、DB2 インストーラーがロールバック操作を実行できなくなります。この場合、ファイルを手動でクリーンアップし、フィックスパックを再インストールする必要があります。

### -p *FPPath*

フィックスパックをどこにインストールするかを指定します。DB2 pureScale 環境では、このパラメーターは必須です。このパラメーターを使用する場合は、以下の点を考慮してください。

- このパラメーターを指定する場合、インスタンスは自動更新されません。**db2iupdt** コマンドを使用してインスタンスを手動で更新する必要があります。また、DB2 pureScale 環境でフィックスパックを複数のホストに適用する場合、**db2iupdt** コマンドをすべてのホストで実行する必要があります。
- DB2 pureScale インスタンス (dsf タイプのインスタンス) を更新するフィックスパックをインストールしている場合、フィックスパックのインストール先のパスは、既存のインストール済み環境とは異なっていなければならない。なおかつすべてのホスト上で同一のパスでなければなりません。非 DB2 pureScale 環境でのフィックスパックのインストールの場合、パスは既存のインストール済み環境と異なるものにすることができます。

-L フィックスパックを現行のホストだけにローカルに適用することを示します。このパラメーターを指定する場合、-H パラメーターは指定できません。

### -H *hostsListFile*

このパラメーターは、DB2 pureScale 環境でのみ指定できます。このパラメーターを使用すると、複数ホスト間でフィックスパックを更新できます。ホスト・リスト・ファイルに指定するホストは、すべて同じサービス・レベルでなければなりません。このパラメーターを指定する場合、-L パラメーターは指定できません。

**注:** このパラメーターを指定して **installFixPack** コマンドを実行した後は、必ず、どのリモート・ホストにもインストール・エラーがないことをログ・ファイルで確認してください。

### -l *log-file*

ログ・ファイルを指定します。root インストールの場合、デフォルト・ログ・ファイルは /tmp/installFixPack.log\$\$ です。\$\$ はプロセス ID を表します。非 root インストールの場合、デフォルト・ログ・ファイルは /tmp/installFixPack\_userID.log です。userID は非 root インストールを所有するユーザー ID を表します。IBM Tivoli System Automation for Multiplatforms (SA MP) を **installFixPack** コマンドを使用してインストールまたは更新している場合、DB2 ログ・ファイルと同じディレクトリーに対応するログ・ファイルが配置されます。

### -t *trace-file*

デバッグ・モードをオンにします。デバッグ情報は、名前を指定したファイルに書き込まれます。

### -h | -?

ヘルプ情報を表示します。



## 使用上の注意

- DB2 Text Search をインストール済みで、Text Search が DB2 コピーに関連したインスタンス上で稼働している場合、インストールによって Text Search サービスの停止が試行されますが、インストールは続行します。
- ホスト・リスト・ファイルのフォーマットは、以下のセクションのようになります。

```
HOST=host1
host1.HOSTNAME=<hostname1>
HOST=host2
host2.HOSTNAME=<hostname2>
...
```

## 例

- DB2 Enterprise Server Edition German が /opt/ibm/db2/COPY1 にインストールされている場合に、フィックスパック 1 イメージを使用して、GA からフィックスパック 1 への対話式アップデートを実行するには、以下を発行します。  
`./installFixPack -b /opt/ibm/db2/COPY1`
- DB2 Enterprise Server Edition German が /opt/ibm/db2/COPY1 にインストールされている場合に、フィックスパック 1 イメージを使用して、GA からフィックスパック 1 へのサイレント・アップデートを実行するには、以下を発行します。  
`./installFixPack -b /opt/ibm/db2/COPY1 -c full_path_to_NLPACK_image -n`
- 何らかの理由でインストール済みの DB2 データベース製品ファイルが破損している場合、アンインストールしてから再インストールすることによってインストールをリフレッシュする代わりに、以下のコマンドを発行します。  
`./installFixPack -f level -b full_path_where_DB2_product_installed`
- インストール・ディレクトリーのスペース所要量を削減するには、以下のコマンドを発行します。  
`./installFixPack -f nobackup -b full_path_where_DB2_product_installed`
- DB2 pureScale インスタンス (dsf タイプのインスタンス) を更新するフィックスパックを適用する場合、フィックスパックのインストール先のパスは、既存のインストール済み環境とは異ならなければならない、なおかつすべてのホスト上で同一のパスでなければなりません。  
`./installFixPack -b full_path_where_DB2_product_installed -p path_where_DB2_fixpack_installed -L`
- フィックスパックを適用して ese インスタンス・タイプを更新する場合、パスは既存のインストール済み環境と同じであっても異なっていても構いません。  
`./installFixPack -b full_path_where_DB2_product_installed -p full_path_where_DB2_product_installed -L`
- DB2 pureScale 環境で、複数ホスト間でフィックスパック更新を適用するには、以下のようにします。  
`./installFixPack -b full_path_where_DB2_product_installed -p full_path_where_DB2_product_installed -H hostsListFile`

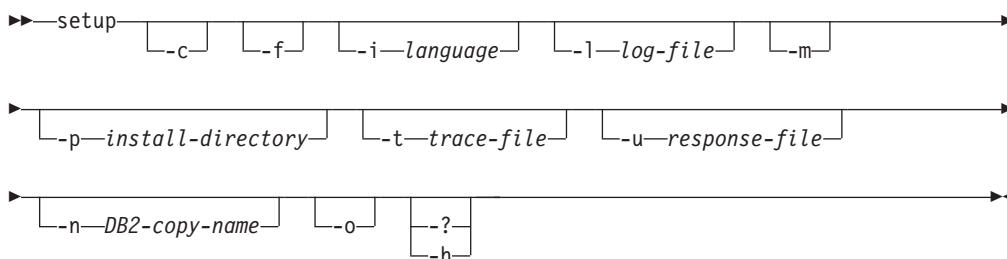
## setup - DB2 データベース製品のインストール

DB2 データベース製品をインストールします。このコマンドは Windows オペレーティング・システムでのみ使用できます。UNIX オペレーティング・システム用のコマンドは **db2setup** です。

このユーティリティーは、DB2 インストール・メディアにあります。これを使って DB2 セットアップ・ウィザードを立ち上げ、インストールを定義して DB2 データベース製品をインストールします。-u オプションを指定して呼び出すと、ユーザーの入力を必要としないインストールが実行されます。インストールの構成情報は応答ファイルから読み取られます。

Windows 上で IBM Data Server Runtime Client をインストールする際のセットアップ・オプションは、DB2 製品のインストールの場合とは異なります。該当するオプションについては、『IBM Data Server Runtime Client インストールのコマンド行オプション (Windows)』を参照してください。

### コマンド構文



### コマンド・パラメーター

注:

DB2 のインストール **setup** コマンドでは、切り替え記号は / および - のどちらを使用しても構いません。

- c インストールの開始直後に **setup.exe** を終了するようにします。このオプションを選択した場合、**setup.exe** の終了コードをモニターする際に、インストールの戻りコードは取得できません。
- f インストール前に DB2 プロセスを強制的に停止します。
- i *language*  
インストールを実行する言語の 2 文字から成る言語コードを指定します。
- l *log-file*  
使用されるログ・ファイルの絶対パスとファイル名。
- m これを -u オプションとともに使用すると、インストールの進行状況ダイアログが表示されます。ただし、何らかの入力を促すプロンプトは出されません。
- p *install-directory*  
製品のインストール・パスを変更します。このオプションを指定すると、応答ファイルで指定されたインストール・パスがオーバーライドされます。

**-t** *trace-file*

インストール・トレース情報を含むファイルを生成します。

**-u** *response-file*

使用される応答ファイルの絶対パスとファイル名を指定します。

**-n** *DB2-copy-name*

インストールで使用する DB2 コピー名を指定します。このオプションを指定すると、応答ファイルで指定されたコピー名がオーバーライドされます。

**-o** 生成されたデフォルトのコピー名を使用して、新しいコピーのインストールを常に実行します。このオプションは、Windows 上で IBM Data Server Driver Package をインストールする場合のみ使用できます。

**-? | -h** 使用法情報を生成します。

### 使用上の注意

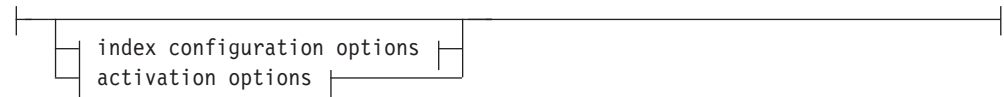
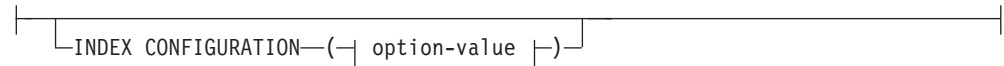
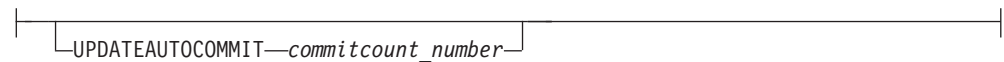
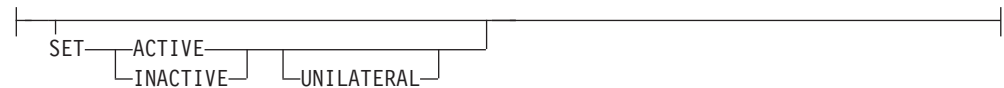
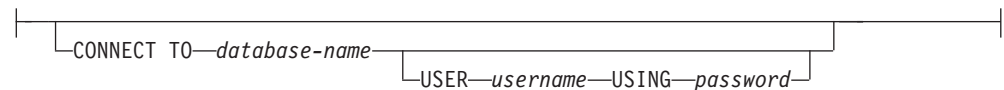
- 絶対パスを指定する場合、パス名は二重引用符の中に含める必要があります。

## setup - DB2 データベース製品のインストール

---

## 第 8 章 DB2 Text Search のコマンド



**options:****index configuration options:****option-value:****activation options:****connection options:****コマンド・パラメーター****ALTER INDEX** *index-name*

**CREATE INDEX** コマンドで指定されている索引のスキーマおよび名前。データベース内のテキスト検索索引を固有に識別します。

**UPDATE FREQUENCY**

索引の更新が行われる頻度を指定します。変更の数が少なくとも **UPDATE MINIMUM** に設定された値である場合に、索引が更新されます。更新頻度 **NONE** は、索引の更新はそれ以降行われなことを意味します。これは、変更されないデータを持つ表のテキスト列に役立ちます。さらに、これはユーザーが (**UPDATE INDEX** コマンドを使用して) 手動で索引を更新する予定の場合にも役立ちます。

**START FOR TEXT** コマンドが実行されていて、DB2 Text Search インスタンス・サービスが実行中の場合にのみ、自動更新が行われます。

デフォルトの頻度値は、ビュー **SYSIBMTS.TSDEFAULTS** の **DEFAULTNAME='UPDATEFREQUENCY'** のところから取得されます。

**NONE**

自動更新は、テキスト索引に適用されません。以降のすべての索引更新は手動で開始する必要があります。

**D** 索引が更新される曜日。

\* 毎日。

*integer1*

日曜日から土曜日の特定の曜日: 0 から 6

**H** 索引が更新される特定の日の時。

\* 毎時。

*integer2*

午前 0 時から午後 11 時までの特定の時: 0 から 23

**M** 索引が更新される特定の時刻の分。

*integer3*

正時 (0) として、または正時の後に 5 分の増分の倍数として指定:  
0、5、10、15、20、25、30、35、40、45、50、または 55

**UPDATE FREQUENCY** オプションを指定しない場合、頻度の設定は変更されないままとなります。

**UPDATE MINIMUM** *minchanges*

索引が増分的に更新される前に生じなければならない、テキスト文書への変更の最小数を指定します。同じテキスト文書への複数の変更は、別個の変更として扱われます。**UPDATE MINIMUM** オプションを指定しない場合、設定は変更されないままとなります。

**INDEX CONFIGURATION** (*option-value*)

バージョン 9.7 フィックスパック 3 以降のフィックスパックでは、これは、テキスト索引構成設定を変更できる **VARCHAR(32K)** タイプのオプションの入力引数です。以下のオプションがサポートされています。

表 53. オプション値の仕様

オプション	値	データ・タイプ	説明
SERIALUPDATE	<i>updatemode</i>	整数	パーティション化されたテキスト検索索引の更新処理を並列モードと直列モードのどちらで行うのかを指定します。並列モードの場合、実行が複数のデータベース・パーティションに分散され、それぞれのノードで独立して行われます。直列モードでは、実行は分散されずに行われ、障害が検出されると停止します。通常、直列モード実行の方が時間はかかりますが、必要とするリソースがかなり少なく済みます。  <ul style="list-style-type: none"> <li>• 0 = 並列モード</li> <li>• 1 = 直列モード</li> </ul>



表 53. オプション値の仕様 (続き)

オプション	値	データ・タイプ	説明
UPDATEAUTOCOMMIT	<i>commitcount_number</i>	整数	<p>ここで指定された回数の索引更新が行われると、コミットが実行され、イニシャル更新またはインクリメンタル更新のためにそれまでの作業が自動的に保存されます。</p> <ul style="list-style-type: none"> <li>イニシャル更新では、データ更新をキャプチャーするトリガーがアクティブ化された後の基本表からの文書のバッチを、索引更新が処理します。文書更新の回数が COMMITCOUNT に達すると、サーバーは暫定コミットを実行します。処理されていない文書により生成されたログ項目は、ステージング表から除去されます。イニシャル・テキスト索引更新に UPDATEAUTOCOMMIT オプションを使用すると、実行時間が大幅に増加します。</li> <li>インクリメンタル更新では、処理されているログ項目は、各暫定コミットで対応するステージング表から除去されます。COMMITCOUNT は更新される文書の数であり、ステージング表項目の数ではありません。</li> </ul>

*activation options*

バージョン 9.7 フィックスパック 3 以降のフィックスパックでは、この整数タイプの入力引数はテキスト索引の状況を設定します。

**ACTIVE**

テキスト索引状況をアクティブに設定します

**INACTIVE**

テキスト索引状況を非アクティブに設定します

**UNILATERAL**

DB2 Text Search 索引の状況にのみ影響する UNILATERAL 変更を指定します。この引数が指定された場合、DB2 Text Search 索引の状況だけがアクティブまたは非アクティブに変更されます。UNILATERAL 引数がない場合、DB2 Text Search 索引および DB2 Net Search Extender 索引のアクティブ化の状況はともに切り替えられ、テキスト索引の一方だけがアクティブ化されます。

**CONNECT TO *database-name***

この節は、接続が確立される先のデータベースを指定します。データベースはローカル・システム上になければなりません。この節が指定されている場合は、環境変数 DB2DBDFT よりも優先されます。以下のすべてが該当する場合、この節は省略できます。

- DB2DBDFT 環境変数が有効なデータベース名に設定されている。
- コマンドを実行しているユーザーが、データベース・サーバーに接続するのに必要な権限を持っている。

**USER** *username* **USING** *password*

この節は、接続を確立するのに使用するユーザー名およびパスワードを指定します。

## 使用上の注意

DB2 のデータベース・オブジェクトおよび照会に適用されるすべての制限および命名規則は、DB2 Text Search のフィーチャーおよび照会にも適用されます。DB2 Text Search 関連の ID は、DB2 の命名規則に準拠している必要があります。さらに、追加の制限事項がいくつかあります。例えば、これらの ID は以下の形式のみになります。

[A-Za-z][A-Za-z0-9@#\$\_]\*

または

"[A-Za-z ][A-Za-z0-9@#\$\_ ]\*"

複数のコマンドが競合する場合、テキスト検索索引でそれらを並行して実行することができないことに注意してください。競合するコマンドの実行中に、このコマンドが発行される場合、エラーが発生してコマンドは失敗します。その後、コマンドの実行を再試行することができます。以下に、競合するコマンドをいくつか示します。

- **ALTER INDEX**
- **CLEAR EVENTS FOR INDEX**
- **DROP INDEX**
- **UPDATE INDEX**
- **DISABLE DATABASE FOR TEXT**

データベースへの変更: DB2 Text Search のカタログ情報を更新します。

索引をアクティブにした結果は、元の索引の状況により異なります。次の表は、結果を示したものです。

表 54. 無効索引無しの状況変更:

イニシャルの DB2 Text Search または Net Search Extender の状況	ACTIVE 要求	ACTIVE UNILATERAL 要求	INACTIVE 要求	INACTIVE UNILATERAL 要求
アクティブ / 非 アクティブ	変更なし	変更なし	非アクティブ / アクティブ	非アクティブ / 非アクティブ
非アクティブ / アクティブ	アクティブ / 非 アクティブ	エラー	変更なし	変更なし
非アクティブ / 非アクティブ	アクティブ / 非 アクティブ	アクティブ / 非 アクティブ	非アクティブ / アクティブ	変更なし

SQL20427N および CIE0379E エラー・メッセージが、索引のアクティブ化競合の場合に戻されます。

## db2ts CLEANUP FOR TEXT

インスタンス内またはデータベース内の DB2 Text Search コレクションをクリーンアップします。

データベースに対してクリーンアップ操作を実行すると、無効なテキスト索引とそれらの関連コレクションがドロップされます。インスタンスに対してクリーンアップ操作を実行すると、廃止コレクションが削除されます。テキスト検索索引が作成されたデータベースにおいて、DB2 Text Search が無効化される前にデータベースがドロップされると、コレクションは廃止される可能性があります。

**注:** コマンドの操作対象はテキスト検索索引ですが、テキスト検索サーバーのツールの操作対象はテキスト検索コレクションです。テキスト検索コレクションとは、テキスト検索索引の基礎となる表現を指します。テキスト検索索引とその関連コレクションの関係は、非パーティション・セットアップの場合は 1:1 で、パーティション・セットアップの場合は 1:n です (n はデータ・パーティションの数)。テキスト検索索引のテキスト検索コレクションを調べるには、SYSIBMTS.TSCOLLECTIONNAMES カタログ表を照会してください。追加情報については、DB2 Text Search の管理ツールに関するトピックを参照してください。

実行するには、コマンド行でコマンドの前に **db2ts** を付ける必要があります。

### 許可

このコマンドをインスタンス・レベルで実行するためには、テキスト検索サーバー・プロセスの所有者でなければなりません。統合型テキスト検索サーバーの場合は、インスタンス所有者であることが必要です。

このコマンドをデータベース・レベルで実行するためには、ステートメントの許可 ID が持つ特権に、SYSTS\_ADM ロールおよび DBADM 権限が含まれている必要があります。

### 必要な接続

このコマンドは DB2 データベース・サーバーから発行されなければなりません。

### コマンド構文

Instance level

▶▶—CLEANUP FOR TEXT—▶▶

Database level

▶▶—CLEANUP FOR TEXT—connection-options—▶▶

### コマンド・パラメーター

なし

## db2ts CLEAR COMMAND LOCKS

特定のテキスト検索索引について、またはデータベース内のすべてのテキスト検索索引について、コマンド・ロックをすべて除去します。コマンド・ロックはテキスト検索索引コマンドの開始時に作成され、コマンドが完了すると破棄されます。それにより、異なるコマンド間の望ましくない競合が避けられます。

このコマンドの使用は、予期しないシステム動作のためにロックがそのまま残ってしまい、明示的にクリーンアップされる必要があるというまれなケースで、必要になります。

実行するには、コマンド行でコマンドの前に **db2ts** を付ける必要があります。

### 許可

索引のロックをクリアするために使用されるステートメントの許可 ID が保持する特権には、以下の両方の権限が含まれている必要があります。

- SYSTS\_MGR ロール
- DBADM 権限、または索引が定義されている基本表に対する CONTROL 特権

データベース接続のロックをクリアするために使用されるステートメントの許可 ID が保持する特権には、SYSTS\_ADM ロールが含まれている必要があります。

### 必要な接続

データベース

### コマンド構文

```

▶▶ CLEAR COMMAND LOCKS ─────────────────── FOR TEXT ───────────────────▶
└── FOR INDEX ─ index-name ─┘
▶ | connection options | ───▶

```

#### connection options:

```

┌──┐
└── CONNECT TO ─ database-name ─┘ ┌── USER ─ username ─ USING ─ password ─┘

```

### コマンド・パラメーター

**FOR INDEX** *index-name*

**CREATE INDEX** コマンドで指定されている索引の名前。

**CONNECT TO** *database-name*

この節は、接続が確立される先のデータベースを指定します。データベースはローカル・システム上になければなりません。この節が指定されている場合は、環境変数 **DB2DBDFT** よりも優先されます。以下のすべてが該当する場合、この節は省略できます。

- **DB2DBDFT** 環境変数が有効なデータベース名に設定されている。

- コマンドを実行しているユーザーが、データベース・サーバーに接続するのに必要な権限を持っている。

**USER** *username* **USING** *password*

この節は、接続を確立するのに使用する許可名およびパスワードを指定します。

### 使用上の注意

コマンド・ロックを所有している処理が非活動であるために、このコマンドを呼び出します。このケースでは、(ロックにより表される) コマンドが完了していない場合や、索引が操作できない場合があります。適切なアクションを取る必要があります。例えば、**DROP INDEX** コマンドを実行している処理が突然停止したとします。いくつかの索引データは削除されましたが、すべてのカタログおよびコレクション情報が削除された訳ではありません。コマンド・ロックは、そのまま残されています。**DROP INDEX** コマンド・ロックをクリアした後、**DROP INDEX** コマンドを再実行することができます。別の例として、**UPDATE INDEX** コマンドを実行しているプロセスが中断したとします。一部の文書の処理は完了しましたが、全部ではありません。また、コマンドのロックがまだ掛かっています。テキスト検索索引の状況を確認し、**UPDATE INDEX** コマンドのロックを解除した後、**UPDATE INDEX** コマンドを再実行できます。

このコマンドが発行されるとき、DB2 Text Search ビュー SYSIBMTS.TSLOCKS の内容が更新されます。

## db2ts CLEAR EVENTS FOR TEXT

このコマンドは、管理用に使用されている索引のイベント表から索引付けイベントを削除します。この表の名前は、ビュー SYSIBMTS.TSINDEXES の列 EVENTVIEWNAME にあります。

少なくとも 1 つの文書処理する索引の更新操作が行われるごとに、通知項目と、場合によってはエラー項目がイベント表に生成されます。自動更新の場合、この表を定期的に検査する必要があります。文書の固有のエラーは訂正されなければなりません (文書内容を変更することにより)。エラーを訂正した後、イベントをクリアすることができます (スペースを消費しすぎないために、クリアすべきです)。

実行するには、コマンド行でコマンドの前に **db2ts** を付ける必要があります。

### 許可

このステートメントの許可 ID には、以下の両方の特権が含まれている必要があります。

- SYSTS\_MGR ロール
- 索引が定義されている表に対する DATAACCESS 権限または CONTROL 特権を持つ DBADM

### 必要な接続

データベース

### コマンド構文

```
▶▶ CLEAR EVENTS FOR INDEX index-name FOR TEXT | connection options | ▶▶
```

**connection options:**

```
|
| CONNECT TO database-name
| USER username USING password
|
```

### コマンド・パラメーター

*index-name*

**CREATE INDEX** コマンドで指定されている索引の名前。索引名は、DB2 索引の命名上の制約に従う必要があります。

**CONNECT TO** *database-name*

この節は、接続が確立される先のデータベースを指定します。データベースはローカル・システム上になければなりません。この節が指定されている場合は、環境変数 DB2DBDFT よりも優先されます。以下のすべてが該当する場合、この節は省略できます。

- DB2DBDFT 環境変数が有効なデータベース名に設定されている。
- コマンドを実行しているユーザーが、データベース・サーバーに接続するのに必要な権限を持っている。

**USER** *username* **USING** *password*

この節は、接続を確立するのに使用する許可名およびパスワードを指定します。

## 使用上の注意

DB2 のデータベース・オブジェクトおよび照会に適用されるすべての制限および命名規則は、DB2 Text Search のフィーチャーおよび照会にも適用されます。DB2 Text Search 関連の ID は、DB2 の命名規則に準拠している必要があります。さらに、追加の制限事項がいくつかあります。例えば、これらの ID は以下の形式のみになります。

```
[A-Za-z][A-Za-z0-9@#$_]*
```

または

```
"[A-Za-z ][A-Za-z0-9@#$_ ]*"
```

通常の更新がスケジュールされている場合 (**CREATE INDEX** または **ALTER INDEX** コマンドの **UPDATE FREQUENCY** オプションを参照)、イベント表を定期的にチェックする必要があります。テキスト検索索引用の DB2 Text Search のイベント表をクリーンアップするには、そのイベントの理由をチェックし、エラーの原因を除去した後に、**CLEAR EVENTS FOR INDEX** コマンドを使用します。

イベント表で参照されているすべての行に、必ず変更を行ってください。ユーザー表の行を変更することにより、次の **UPDATE INDEX** の試行が、以前にエラーのあった文書を確実に正常に再索引付けできるようにします。

複数のコマンドが競合する場合、テキスト検索索引でそれらを並行して実行することができないことに注意してください。競合するコマンドの実行中に、このコマンドが発行される場合、エラーが発生してコマンドは失敗します。その後、コマンドの実行を再試行することができます。以下に、競合するコマンドをいくつか示します。

- **CLEAR EVENTS FOR INDEX**
- **UPDATE INDEX**
- **ALTER INDEX**
- **DROP INDEX**
- **DISABLE DATABASE FOR TEXT**

データベースへの変更: イベント表がクリアされます。

## db2ts CREATE INDEX

**db2ts CREATE INDEX** コマンドは、テキスト列のテキスト検索索引を作成します。その後、テキスト検索関数を使用して、列データを検索することができます。

テキスト検索の **UPDATE INDEX** コマンドを実行するか、または索引に定義された更新頻度に従って DB2 管理タスク・スケジューラーが **UPDATE INDEX** コマンドを実行するまで、テキスト検索索引の中にデータが入ることはありません。

実行するには、コマンド行でコマンドの前に **db2ts** を付ける必要があります。

### 許可

ステートメントの許可 ID には、SYSTS\_MGR ロール、データベースの CREATETAB 権限、および以下のいずれかの権限または特権が含まれている必要があります。

- その索引の定義されている表に対する CONTROL 特権。
- 索引が定義される表に対する INDEX 特権と、以下の権限のいずれか。
  - データベースに対する IMPLICIT\_SCHEMA 権限 (索引の暗黙または明示のスキーマ名が存在しない場合)
  - スキーマに対する CREATEIN 特権 (索引のスキーマ名が存在する場合)
- DBADM 権限

### 必要な接続

データベース

### コマンド構文

```

▶▶ CREATE INDEX index-name FOR TEXT ON schema-name table-name
▶└─( text-column-name )
└─( function-name ( text-column-name ) )
▶└─ text default information └─ update characteristics └─
▶└─ storage options └─ index configuration options └─ partition options └─
▶└─ connection options └─

```

#### text default information:

```

└─┬─ CODEPAGE code-page └─ LANGUAGE locale └─ FORMAT format

```

#### update characteristics:

```

└─ UPDATE FREQUENCY └─ NONE
└─ update frequency

```



► incremental update characteristics

#### update frequency:

D-(\*-integer1)-H-(\*-integer2)-M-(integer3)

#### incremental update characteristics:

UPDATE MINIMUM—*minchanges*

#### storage options:

COLLECTION DIRECTORY—*directory*

ADMINISTRATION TABLES IN—*tablespace-name*

#### index configuration options:

INDEX CONFIGURATION—(option-value)

#### option-value:

COMMENT—*text*  
 UPDATEAUTOCOMMIT—*commitcount\_number*  
 AUXLOG—*auxlog\_value*  
 CJKSEGMENTATION—*cjksegmentation-method*

#### connection options:

CONNECT TO—*database-name*—USER—*username*—USING—*password*

## コマンド・パラメーター

### CREATE INDEX *index-name*

作成される索引の名前を指定します (スキーマ修飾はオプション)。その名前はデータベース内のテキスト検索索引を固有に識別します。索引名は、DB2 索引の命名上の制約に従う必要があります。

### ON *table-name*

テキスト列を含む表名。テキスト検索索引を、以下の表に作成することはできません。

- フェデレーテッド表
- マテリアライズ照会表
- ビュー

### *text-column-name*

索引を付ける列の列名。列は、CHAR、VARCHAR、CLOB、DBCLOB、BLOB、GRAPHIC、VARGRAPHIC、または XML のデータ・タイプのいずれかでなければなりません。列のデータ・タイプがこれらのいずれのデータ・タイプでもない場合、*function-schema.function-name* で指定された変換関数を使用して、列タイプを有効なタイプのいずれかに変換します。あるいは、索引付けされるテキスト文書にアクセスするユーザー定義の外部関数を指定することができます。1 つのテキスト検索索引のみが、列に作成されます。

### *function-name(text-column-name)*

DB2 命名規則に準拠して、テキスト検索でサポートされるタイプではない列内のテキスト文書にアクセスする外部スカラー関数のスキーマ修飾名を指定します。その値のデータ・タイプ変換を実行し、テキスト検索でサポートされるデータ・タイプの 1 つとして値を戻します。このタスクは、列タイプの変換を実行します。この関数は、1 つのパラメーターのみを取り、1 つの値のみを戻す必要があります。

### CODEPAGE *code-page*

テキスト文書を索引付けするときに使用される DB2 コード・ページ (CODEPAGE) を指定します。デフォルト値は、ビュー SYSIBMTS.TSDEFAULTS の DEFAULTNAME='CODEPAGE' のところの値で指定されます (これがデータベース・コード・ページである場合もあります)。この引数はバイナリー・データ・タイプにのみ適用されます。例えば列タイプまたは変換関数からの戻りの型は、BLOB または文字タイプの FOR BIT DATA でなければなりません。

### LANGUAGE *locale*

索引付け中に、文書の言語固有の処理のために DB2 Text Search によって使用される言語を指定します。ロケールを指定しない場合は、データベース・テリトリーが LANGUAGE のデフォルト設定を判別します。ロケールを判別するために、文書が自動的にスキャンされるようにするには、*locale* を AUTO に指定してください。

### FORMAT *format*

列内のテキスト文書のフォーマットを指定します。サポートされているフォーマットには TEXT、XML、HTML、および INSO が含まれます。文書を索引付けするときに、DB2 Text Search はこの情報を必要とします。フォーマットを指定しない場合、デフォルト値が使用されます。デフォルト値は、ビュー SYSIBMTS.TSDEFAULTS の DEFAULTNAME='FORMAT' のところの値です。データ・タイプ XML の列の場合は、DEFAULTNAME の値に関係なく、デフォルトのフォーマット 'XML' が使用されます。INSO フォーマットを使用するためには、リッチ・テキスト・サポートがインストールされている必要があります。

**UPDATE FREQUENCY**

索引の更新が行われる頻度を指定します。変更の数が少なくとも **UPDATE MINIMUM** に設定された値である場合に、索引が更新されます。更新頻度 **NONE** は、索引の更新はそれ以降行われないことを意味します。**NONE** の更新頻度は、変更されないデータを持つ表のテキスト列に役立ちます。さらに、これはユーザーが (**UPDATE INDEX** コマンドを使用して) 手動で索引を更新する予定の場合にも役立ちます。**START FOR TEXT** コマンドが実行されていて、DB2 Text Search インスタンス・サービスが実行中の場合にのみ、自動更新が行われます。

デフォルトの頻度値は、ビュー **SYSIBMTS.TSDEFAULTS** の **DEFAULTNAME=&apos;UPDATEFREQUENCY&apos;** のところから取得されます。

**NONE**

これ以上の索引更新は行われません。更新を手動で開始する必要があります。

**D** 索引が更新される曜日。

\* 毎日。

*integer1*

日曜日から土曜日の特定の曜日: 0 から 6

**H** 索引が更新される指定日の時。

\* 毎時。

*integer2*

午前 0 時から午後 11 時までの特定の時: 0 から 23

**M** 索引が更新される指定時の分。

*integer3*

正時 (0) として、または正時の後に 5 分の増分の倍数として指定:  
0、5、10、15、20、25、30、35、40、45、50、または 55

**UPDATE MINIMUM** *minchanges*

**UPDATE FREQUENCY** で指定された時間に索引が増分的に更新される前に、テキスト文書に加えらる変更の最小数を指定します。正整数値だけを指定できます。デフォルト値はビュー **SYSIBMTS.TSDEFAULTS** から取られます。ここで、**DEFAULTNAME='UPDATEMINIMUM'** です。

注: この値は、**UPDATE INDEX** コマンドの実行中は無視されます (**USING UPDATE MINIMUM** オプションがそこで使用されていない場合)。小さい値は、表列とテキスト検索索引の間の整合性を増します。ただし、これによってシステムに余分の負荷がかかることにもなります。

**COLLECTION DIRECTORY** *directory*

テキスト検索索引コレクションを保管するディレクトリーを指定します。絶対パスを指定する必要があります。絶対パス名の最大長は 215 文字です。テキスト検索サーバーのインスタンス・サービスのプロセス所有者は、このディレクトリーに対する読み取りおよび書き込み権限を持っている必要があります。

**COLLECTION DIRECTORY** パラメーターは、統合テキスト検索サーバーのセットアップでのみサポートされます。コレクションの場所に関する追加情報については、使用上の注意を参照してください。

**ADMINISTRATION TABLES IN *tablespace-name***

索引用に作成される管理表のための、TEMPORARY 表スペース以外の既存の表スペースの名前を指定します。

非パーティション・データベースでは、表スペースが指定されていない場合は、索引を作成する基本表の表スペースが使用されます。

パーティション・データベースでは、テキスト索引を作成する際には、

**ADMINISTRATION TABLES** パラメーターを明示的に使用する必要があります。この表スペースは、テキスト検索索引のステージング表が対応する基本表と同じ方法で分散されるようにするには、基本表の表スペースと同じパーティション・グループ内になければなりません。

**INDEX CONFIGURATION (*option-value*)**

追加の索引関連値をオプション値のストリングのペアとして指定します。以下の値がサポートされています。

表 55. オプション値の仕様

オプション	値	データ・タイプ	説明
<b>COMMENT</b>	<i>text</i>	512 バイトより少ないストリング値	DB2 Text Search カタログ・ビュー TSINDEXES にある REMARKS 列にストリング・コメント値を追加します。ストリング・コメント値をコレクションの説明としても追加します。
<b>UPDATEAUTOCOMMIT</b>	<i>commitcount_number</i>	整数	このオプションは索引更新の回数を指定するもので、その回数に達すると、コミットが実行され、イニシャル更新またはインクリメンタル更新のためにそれまでの作業が自動的に保存されます。 <ul style="list-style-type: none"> <li>イニシャル更新では、データ更新をキャプチャーするトリガーがアクティブ化された後の基本表からの文書のバッチを、索引更新が処理します。文書更新の回数が COMMITCOUNT の数に達すると、サーバーは暫定コミットを実行します。処理されていない文書により生成されたログ項目は、ステージング表から除去されます。イニシャル・テキスト索引更新に <b>UPDATEAUTOCOMMIT</b> オプションを使用すると、実行時間が大幅に増加します。</li> <li>インクリメンタル更新では、処理されているログ項目は、各暫定コミットで対応するステージング表から除去されます。COMMITCOUNT は更新される文書の数であり、ステージング表項目の数ではありません。</li> </ul>
<b>AUXLOG</b>	<i>auxlog_value</i>	ストリング	トリガーに認識されない変更をキャプチャーするための、追加のログ・インフラストラクチャーの作成を制御します。範囲パーティション表のデフォルト設定は ON であり、それ以外の場合は OFF になります。非範囲パーティション表の場合は AuxLogNorm、範囲パーティション表の場合は AuxLogPart の設定によって、デフォルト表におけるデフォルト値を変更できます。

表 55. オプション値の仕様 (続き)

オプション	値	データ・タイプ	説明
<b>CJKSEGMENTATION</b>	<i>CJKSEGMENTATION_method</i>	512 バイトより少ないストリング値	<p>中国語、日本語、および韓国語 (zh_CN、zh_TW、ja_JP、ko_KR ロケール・セット) の文書のセグメンテーション方式を指定します。自動言語検出が有効 (LANGUAGE AUTO) になっている場合のこれらの文書も対象になります。オプションを指定しない場合、デフォルト表の <b>CJKSEGMENTATION</b> の値が適用されます。サポートされる値は、以下のとおりです。</p> <ul style="list-style-type: none"> <li>• MORPHOLOGICAL</li> <li>• NGRAM</li> </ul> <p>指定したセグメンテーション方式は <b>SYSIBMTS.TSCONFIGURATION</b> 管理ビューに追加されます。テキスト索引が作成された後に変更することはできません。</p>

**要確認:** コメントのような非数値は、単一引用符で囲む必要があります。ストリング値内の単一引用符の文字は、2 つの連続した単一引用符によって表記される必要があります。

**例:** INDEX CONFIGURATION (COMMENT 'Index on User''s Guide column')

### パーティション・オプション

IBM の内部使用のために予約済みです。

### CONNECT TO *database-name*

この節は、接続が確立される先のデータベースを指定します。データベースはローカル・システム上になければなりません。この節が指定されている場合は、環境変数 **DB2DBDFT** よりも優先されます。以下のすべてが該当する場合、この節は省略できます。

- **DB2DBDFT** 環境変数が有効なデータベース名に設定されている。
- コマンドを実行しているユーザーが、データベース・サーバーに接続するのに必要な権限を持っている。

### USER *username* USING *password*

この節は、接続を確立するのに使用する許可名およびパスワードを指定します。

## 使用上の注意

DB2 のデータベース・オブジェクトおよび照会に適用されるすべての制限および命名規則は、DB2 Text Search のフィーチャーおよび照会にも適用されます。DB2 テキスト関連の ID は、DB2 の命名規則に準拠している必要があります。さらに、追加の制限事項がいくつかあります。例えば、これらの ID は以下の形式のみになります。

[A-Za-z][A-Za-z0-9@#\$\_]\*

または

"[A-Za-z ][A-Za-z0-9@#\$\_ ]\*"

**CREATE INDEX** コマンドが正常に実行されると、以下のようになります。

- DB2 Text Search のサーバー・データが更新されます。以下の例のように、*instance\_database-name\_index-identifier\_number* という名前のコレクションがデータベース・パーティションごとに作成されます。

```
tigertail_MYTSDB_TS250517_0000
```

コレクション名は SYSIBMTS.TSCOLLECTIONNAMES ビュー (列 COLLECTIONNAME) から取られます。

- DB2 Text Search のカタログ情報が更新されます。索引のステージング表が、適切な DB2 索引のある指定された表スペースに作成されます。さらに、索引のイベント表が指定された表スペースに作成されます。auxlog on オプションが設定されている場合には、保全性処理を介して変更をキャプチャーするために、第 2 のステージング表が作成されます。
- DB2 Text Search と DB2 Net Search Extender が共に存在し、テーブル列のアクティブ Net Search Extender 索引が存在している場合、新規テキスト索引は非アクティブに設定されます。
- 新規に作成されたテキスト検索索引には、自動的にデータが追加されません。テキスト検索索引にデータが追加されるように、**UPDATE INDEX** コマンドが手動でまたは自動的に (**UPDATE FREQUENCY** オプションの指定により索引に定義されている更新スケジュールの結果として) 実行される必要があります。
- 頻度が指定されている場合、DB2 管理スケジューラー用のスケジュール・タスクが作成されます。

### 使用上の制限

- 主キーが表に対して定義されている必要があります。DB2 Text Search では、複数列の DB2 主キーをタイプの制限なしで使用できます。主キー列の数は、DB2 によって許可されている主キー列の数より 2 列少ない数に制限されています。
- DB2 Text Search 索引のある表のすべての主キー列の合計長は、DB2 によって許可されている主キー列の最大合計長より 15 バイト少ない長さに制限されています。DB2 CREATE INDEX ステートメントの制約事項を参照してください。

複数のコマンドは、競合する可能性がある場合は、テキスト検索索引に対して同時に実行することはできません。競合するコマンドの実行中に、このコマンドが発行される場合、エラーが発生してコマンドは失敗します。その後、コマンドの実行を再試行することができます。以下のコマンドが競合します。

- **DISABLE DATABASE FOR TEXT**

索引の作成後に、テキスト索引の補助ログ・プロパティを変更することはできません。

パーティション・データベース環境では、以下の条件が満たされている必要があります。

- テキスト検索索引に固有の管理表 (ステージング表など) およびテキスト検索索引は、対応する基本表と同様の仕方で分散されます。テキスト検索索引の作成中に **ADMINISTRATION TABLES IN** 節を使用すると、指定された表スペースが、基本表の表スペースと同じパーティション・グループに入れられます。

**CJKSEGMENTATION** オプションは、中国語、日本語、および韓国語のための、zh\_CN、zh\_TW、ja\_JP、および ko\_KR ロケール・セットに適用されます。セグメンテーション方式に指定された MORPHOLOGICAL または NGRAM オプションは、SYSIBMTS.TSCONFIGURATION 管理ビューに追加されます。

**LANGUAGE** パラメーターに AUTO オプションを設定して索引を作成すると、**CJKSEGMENTATION** 指定がオプションとして許可されます。指定されたセグメンテーション方式は、中国語、日本語、および韓国語の文書に適用されます。いったん索引作成が完了したら、**CJKSEGMENTATION\_method** 値の値セットを変更することはできません。

**LANGUAGE** パラメーターに AUTO を設定し、**CJKSEGMENTATION** に MORPHOLOGICAL を設定してテキスト検索索引を作成した場合、形態素索引で有効なストリングを検索すると、予期される結果が返されないことがあります。このような場合は、以下のサンプル・ステートメントに示すように **CONTAINS** 関数の **QUERYLANGUAGE** オプションを使用して結果を取得してください。

```
select bookname from ngrambooks where contains (story, '■■■','QUERYLANGUAGE=zh_CN') = 1
```



## db2ts DISABLE DATABASE FOR TEXT

このコマンドは、コマンド **ENABLE DATABASE FOR TEXT** によって行われた変更を元に戻します (例えば、テキスト検索関連の表およびビューをドロップします)。

このコマンドが発行される時、以下のことを行います。

- データベースの DB2 Text Search フィーチャーを使用不可にします。
- テキスト検索のカタログ表とビュー、および関連するデータベース・オブジェクトをドロップします。
- **FORCE** オプションを指定した場合、すべてのテキスト索引情報はデータベースから削除され、すべての関連する集合も削除されます。『db2ts DROP INDEX コマンド』を参照してください。

実行するには、コマンド行でコマンドの前に **db2ts** を付ける必要があります。

### 許可

このステートメントの許可 ID には、以下の両方の特権が含まれている必要があります。

- DBADM と DATAACCESS を併せ持つ権限。
- SYSTS\_ADM ロール。

### 必要な接続

データベース

### コマンド構文

```

▶▶—DISABLE DATABASE FOR TEXT—[FORCE]—| connection options |————▶▶

```

#### connection options:

```

|—————|
|CONNECT TO—database-name—|
|USER—username—USING—password—|
|—————|

```

### コマンド・パラメーター

#### FORCE

すべてのテキスト検索索引がデータベースから強制的にドロップされるように指定します。

このオプションが指定されておらず、テキスト検索索引がこのデータベースに対して定義されている場合、このコマンドは失敗します。

このオプションが指定されていても、DB2 Text Search サービスが開始されていない (db2ts **START FOR TEXT** コマンドが発行されていない) 場合、テキスト検索索引 (コレクション) はドロップされません。 **db2ts CLEANUP** コマンドを使用して手動でクリーンアップする必要があります。



**CONNECT TO** *database-name*

この節は、接続が確立される先のデータベースを指定します。データベースはローカル・システム上になければなりません。この節が指定されている場合は、環境変数 DB2DBDFT よりも優先されます。以下のすべてが該当する場合、この節は省略できます。

- DB2DBDFT 環境変数が有効なデータベース名に設定されている。
- コマンドを実行しているユーザーが、データベース・サーバーに接続するのに必要な権限を持っている。

**USER** *username* **USING** *password*

この節は、接続を確立するのに使用する許可名およびパスワードを指定します。

**使用上の注意**

このコマンドは、データベースにおける DB2 Net Search Extender の使用可能化状況には影響しません。このコマンドは **ENABLE FOR TEXT** コマンドにより作成された DB2 Text Search のカタログ表およびビューを削除します。

テキスト検索索引の定義を持つ DB2 データベースをドロップする前に、このコマンドを発行し、テキスト索引およびコレクションが正常に除去されたことを確認してください。

**FORCE** オプションを使用して、いくつかの索引が削除できなかった場合、コレクション名が **db2diag** ログ・ファイルに書き込まれます。

**注:** 結果がオーファン集合 (例えば残された集合がテキスト検索サーバー上に定義された状態であるが、DB2 には使用されない) となる使用法は推奨されていません。以下は、孤立したコレクションの原因となるいくつかのケースです。

- **DISABLE DATABASE FOR TEXT** コマンドを実行せずに **DROP DATABASE CLP** コマンドを実行するとき。
- **FORCE** オプションを使用して **DISABLE DATABASE FOR TEXT** コマンドを実行するとき。
- その他のいくつかのエラー条件。

複数のコマンドは、競合する可能性がある場合は、テキスト検索索引に対して同時に実行することはできません。競合するコマンドの実行中に、このコマンドが発行される場合、エラーが発生してコマンドは失敗します。その後、コマンドの実行を再試行することができます。以下に、競合するコマンドをいくつか示します。

- **DROP INDEX**
- **UPDATE INDEX**
- **CLEAR EVENTS FOR INDEX**
- **ALTER INDEX**
- **DISABLE DATABASE FOR TEXT**

## db2ts DROP INDEX

**db2ts DROP INDEX** コマンドは、既存のテキスト検索索引をドロップします。

実行するには、コマンド行でコマンドの前に **db2ts** を付ける必要があります。

### 許可

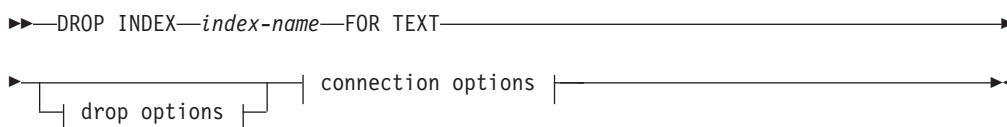
ステートメントの許可 ID が持つ特権には、SYSTS\_MGR ロールと、以下のいずれかの特権または権限が含まれている必要があります。

- その索引の定義されている表に対する CONTROL 特権。
- その索引の定義されているスキーマに対する DROPIN 特権。
- テキスト検索索引に既存のスケジュールがある場合、許可 ID は索引作成者と同じであるか、または DBADM 権限を持っている必要があります。

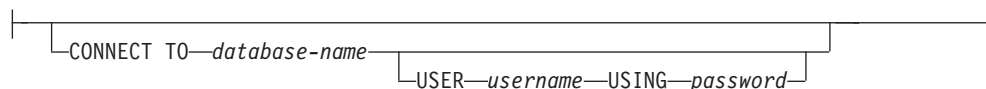
### 必要な接続

データベース

### コマンド構文



#### connection options:



### コマンド・パラメーター

#### **DROP INDEX** *index-name* **FOR TEXT**

**CREATE INDEX** コマンドで指定されている索引のスキーマおよび名前。データベース内のテキスト検索索引を固有に識別します。

#### *drop\_options*

IBM の内部使用のために予約済みです。

#### **CONNECT TO** *database-name*

この節は、接続が確立される先のデータベースを指定します。データベースはローカル・システム上になければなりません。この節が指定されている場合は、環境変数 DB2DBDFT よりも優先されます。以下のステートメントすべてが該当する場合、この節は省略できます。

- DB2DBDFT 環境変数が有効なデータベース名に設定されている。
- コマンドを実行しているユーザーが、データベース・サーバーに接続するのに必要な権限を持っている。

#### **USER** *username* **USING** *password*

この節は、接続を確立するのに使用する許可名およびパスワードを指定します。

## 使用上の注意

複数のコマンドは、競合する可能性がある場合は、テキスト検索索引に対して同時に実行することはできません。競合するコマンドの実行中に、このコマンドが発行される場合、エラーが発生してコマンドは失敗します。その後、コマンドの実行を再試行することができます。以下のコマンドは、一般的な競合コマンドです。

- **DROP INDEX**
- **UPDATE INDEX**
- **CLEAR EVENTS FOR INDEX**
- **ALTER INDEX**
- **DISABLE DATABASE FOR TEXT**

**DROP** 操作と並行して **STOP FOR TEXT** コマンドを実行しても、コマンドの競合に関するメッセージは出されません。その代わりに、**DROP** によってコレクションが削除される前にテキスト検索サーバーがシャットダウンされると、テキスト検索サーバーが使用できないというエラーが返されます。

テキスト検索索引がドロップされた後、テキスト検索は対応するテキスト列で使用できなくなります。同じテキスト列に新規のテキスト検索を作成する予定の場合、まずデータベースから切断し、再接続してから、新規のテキスト検索索引を作成する必要があります。

**db2ts DROP INDEX FOR TEXT** コマンドは、データベースに以下の変更を行います。

- DB2 Text Search のカタログ情報を更新します。
- 索引ステージング表およびイベント表をドロップします。
- ユーザー・テキスト表のトリガーを削除します。
- DB2 Text Search 索引定義に関連付けられたコレクションを破棄します。

## db2ts ENABLE DATABASE FOR TEXT

**db2ts ENABLE DATABASE FOR TEXT** コマンドは、現行データベースで DB2 Text Search を使用できるようにします。このコマンドは管理表および管理ビューを作成して各パラメーターにデフォルト値を設定します。データベース内の表の列にテキスト検索索引を作成できるようにするには、このコマンドを正常に実行する必要があります。コマンド行でコマンドの前に **db2ts** を付ける必要があります。

データベースを使用可能にした後、SYSIBMTS.TSSERVERS ビュー内の Text Search サーバーの接続情報を指定する必要があります。使用可能にする操作には、サーバーにデータを設定しようとする操作が含まれるので、サーバー構成にアクセスできない場合には警告が表示されます。どのような場合であれ、ビューの接続情報を確認することをお勧めします。詳細については、DB2 Text Search サーバー情報の更新に関するトピックを参照してください。

### 許可

- このステートメントの許可 ID が持つ特権には、SYSTS\_ADM ロールおよび DBADM 権限が含まれている必要があります。

### 必要な接続

データベース

### コマンド構文

```

▶▶—ENABLE DATABASE FOR TEXT—▶▶
|
|  ┌── ADMINISTRATION TABLES IN tablespace-name ──┐
|  └──┘
|
|  ┌── AUTOGRANT ──┐ ┌── connection options ──┐ ┌──▶▶
|  └──────────┘ └──────────┘ └──────────┘

```

### connection options:

```

┌──┐
└──CONNECT TO—database-name──┐ ┌──USER—username—USING—password──┐

```

### コマンド・パラメーター

#### ADMINISTRATION TABLES IN *tablespace-name*

データベースを DB2 Text Search に使用可能にする際に作成される管理表のための、既存の REGULAR 表スペースの名前を指定します。この表スペースはデータベース・パーティション・グループ IBMCATGROUP 内にいることが推奨されています。パーティション・データベースの場合、バッファー・プールおよび表スペースは 32 KB ページ・サイズで定義されている必要があります。

この節を指定しない場合、SYSTOOLSPACE がデフォルトの表スペースとして使用されます。この場合、SYSTOOLSPACE が既に存在していることを確認してください。存在しない場合は、SYSPROC.SYSINSTALLOBJECTS プロシージャを使用して作成できます。

注: 大/小文字の区別のある表スペース名を指定するには、引用符を使用してください。

#### AUTOGRANT

このオプションは非推奨で、インスタンス所有者に特権を付与しなくなりました。使用はもはや推奨されておらず、今後のリリースで除去される可能性があります。

#### CONNECT TO *database-name*

この節は、接続が確立される先のデータベースを指定します。データベースはローカル・システム上になければなりません。この節が指定されている場合は、環境変数 DB2DBDFT よりも優先されます。以下のステートメントすべてが該当する場合、この節は省略できます。

- DB2DBDFT 環境変数が有効なデータベース名に設定されている。
- コマンドを実行しているユーザーが、データベース・サーバーに接続するのに必要な権限を持っている。

#### USER *username* USING *password*

この節は、接続を確立するのに使用する許可名およびパスワードを指定します。

### 例

例 1: tsspace という名前の表スペース内に管理表を作成してデータベースを DB2 Text Search に使用できるようにし、すべてのエラー・メッセージを英語で戻します。

```
CALL SYSPROC.SYSTS_ENABLE('ADMINISTRATION TABLES IN tsspace', 'en_US', ?)
```

以下はこの照会の出力例です。

```
Value of output parameters
-----
Parameter Name : MESSAGE
Parameter Value : Operation completed successfully.

Return Status = 0
```

### 使用上の注意

正常に実行されるとき、このコマンドは以下のアクションを行います。

- データベースで DB2 Text Search フィーチャーを使用可能にします。
- ビュー SYSIBMTS.TSDEFAULTS に DB2 Text Search のデータベース構成デフォルト値を確立します。
- 以下の DB2 Text Search 管理ビューを SYSIBMTS スキーマに作成します。
  - SYSIBMTS.TSDEFAULTS
  - SYSIBMTS.TSLOCKS
  - SYSIBMTS.TSINDEXES
  - SYSIBMTS.TSCONFIGURATION
  - SYSIBMTS.TSCOLLECTIONNAMES
  - SYSIBMTS.TSSERVERS

## db2ts HELP

**db2ts HELP** は、使用可能な DB2 Text Search コマンドのリスト、または個々のコマンドの構文を表示します。

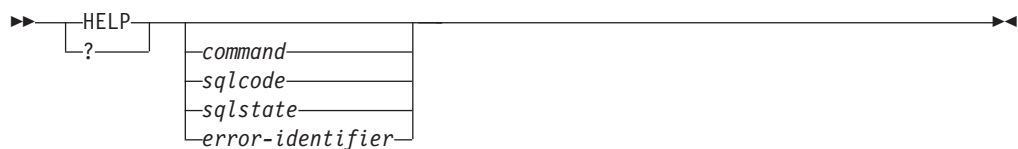
**db2ts HELP** コマンドを使用して、特定のエラー・メッセージに関するヘルプも表示できます。

実行するには、コマンド行でコマンドの前に **db2ts** を付ける必要があります。

### 許可

なし。

### コマンド構文



### コマンド・パラメーター

#### HELP | ?

コマンドまたは理由コードのヘルプ情報を表示します。

#### *command*

DB2 Text Search コマンドを識別する最初のキーワード:

- ALTER
- CLEANUP
- CLEAR (CLEAR COMMAND LOCKS と CLEAR EVENTS FOR INDEX の両方)
- CREATE
- DISABLE
- DROP
- ENABLE
- RESET PENDING
- START
- STOP
- UPDATE

#### *sqlcode*

(管理ストアード・プロシージャの内部または外部で) db2ts コマンドまたはテキスト検索照会によって戻されるメッセージの SQLCODE。

#### *sqlstate*

コマンド、管理ストアード・プロシージャ、またはテキスト検索照会で戻される sqlstate。

*error-identifier*

ID は、エラー・メッセージに組み込まれる *text-search-error-msg* の一部です。この ID は 'CIE' で始まり、*CIEnnnnn* の形式になります。*nnnnn* は数値です。この ID は、テキスト検索中のエラーの際に戻される特定のエラーを表します。これはまた、テキスト検索コマンドの終了時の通知メッセージ、またはテキスト検索管理プロシージャの完了時に印刷されるメッセージによって戻されることもあります。ID が 'CIE' から始まっていない場合、**db2ts help** は *error-identifier* に関する情報を提供できません。例えば、db2ts は IQQR0012E などの *error-identifier* によるメッセージのヘルプを提供できません。

**使用上の注意**

UNIX シェルを使用するときは、以下の例のように、二重引用符を使って **db2ts** に引数を指定する必要がある場合があります。

```
db2ts "? CIE00323"
```

引用符がないと、シェルはワイルドカードを作業ディレクトリーの内容と一致させようとするので、予期しない結果が生じることがあります。

いずれかの db2ts コマンドの最初のキーワードが指定された場合、識別されたコマンドの構文が表示されます。2 つの db2ts コマンドが同じ最初のキーワードを共有する場合 (**CLEAR COMMAND LOCKS** と **CLEAR EVENTS FOR INDEX**)、db2ts help clear を発行すると両方のコマンドの構文が表示されますが、db2ts help clear events などのように 2 番目のキーワードを追加してそれらを区別すると、各コマンドを特定して表示することができます。? または **HELP** の後にパラメーターが指定されていない場合、db2ts は使用可能なすべての db2ts コマンドをリストします。

*sqlcode*、*sqlstate*、または CIE *error-identifier* を指定すると、そのコード、状態、またはエラー ID に関する情報が戻されます。例えば、

```
db2ts help SQL20423
```

または

```
db2ts ? 38H10
```

または

```
db2ts ? CIE00323
```

## db2ts START FOR TEXT

**db2ts START FOR TEXT** コマンドは、他の DB2 Text Search 管理コマンドおよび SQL 照会内のテキスト検索索引を参照する機能をサポートする DB2 Text Search インスタンス・サービスを開始します。

サーバーにリッチ・テキスト・サポートが構成されている場合、**db2ts START FOR TEXT** コマンドで実行される操作には、DB2 Text Search サーバーを実行するホスト・マシン上のリッチ・テキスト・サポートのためのプロセスを開始する操作も含まれます。

このコマンドは DB2 データベース・サーバーから発行されなければなりません。

統合テキスト検索のセットアップを使用しているパーティション・データベース環境でインスタンス・サービスを開始する場合は、統合テキスト検索サーバー・ホスト・マシン上でコマンドを実行する必要があります。デフォルトでは、統合テキスト検索サーバー・ホスト・マシンは、最も小さい番号のデータベース・パーティション・サーバーのホストです。

### 許可

インスタンス所有者。データベース特権は要求されていない

### コマンド構文

```

▶▶—START FOR TEXT—┌STATUS┐—————▶▶
                    └VERIFY┘

```

### コマンド・パラメーター

#### STATUS

DB2 Text Search サーバーの状況を検査します。サーバーの STARTED 状況または STOPPED 状況を示す詳細情報メッセージが返されます。

#### VERIFY

DB2 Text Search サーバーの開始状況を検査し、操作が成功した場合にはそのことを示す戻りコード 0 を標準メッセージと共に返して終了します。テキスト検索サーバーのその他の状況の場合、または状況を検査できない場合には、ゼロ以外のコードが戻されます。

### 例

- テキスト検索サーバーが開始されたことを検査します。

```

Linux/UNIX:
$ db2ts START FOR TEXT VERIFY
CIE00001 Operation completed successfully.

```

```

$ echo $?
0

```

```

Windows:
C:¥> db2ts START FOR TEXT VERIFY

```



```
CIE00001 Operation completed successfully.
```

```
C:\> echo %ERRORLEVEL%
0
```

## 使用上の注意

- パーティション・データベース環境では、いずれか 1 つのパーティション・サーバー・ホストに対して、**STATUS** および **VERIFY** パラメーターを付けて **db2ts START FOR TEXT** コマンドを発行できます。インスタンス・サービスを開始するには、統合テキスト検索サーバー・ホスト・マシン上で **db2ts START FOR TEXT** コマンドを実行する必要があります。統合テキスト検索サーバー・ホスト・マシンは、最も小さい番号のデータベース・パーティション・サーバーのホストです。カスタム・コレクション・ディレクトリーが使用されている場合には、それより小さい番号のパーティションが後で作成されないように注意してください。この制限は特に、Linux および UNIX プラットフォームで該当します。インスタンスの作成時に DB2 Text Search を構成する場合には、構成によって最初に統合テキスト検索サーバー・ホストが判別されます。その構成が常に、最も小さい番号のデータベース・パーティション・サーバーのホストでなければなりません。
- Windows プラットフォームでは、DB2 Text Search 用の DB2 インスタンスごとに、関連付けられた Windows サービスが 1 つずつあります。次のコマンドを発行することにより、サービス名を判別することができます。

```
DB2TS - <instance name>[-<partition number>]
```

. **db2ts START FOR TEXT** コマンドを使用することのほかに、「コントロール パネル」または **NET START** コマンドを使用してサービスを開始することもできます。

## db2ts STOP FOR TEXT

**db2ts STOP FOR TEXT** コマンドは、DB2 Text Search インスタンス・サービスを停止します。実行中のサービスに、リッチ・テキスト・サポート用のプロセスが含まれている場合には、それらのサービスも停止されます。

このコマンドは DB2 データベース・サーバーから発行されなければなりません。

このコマンドをコマンド行から実行するときは、DB2 コマンド行で、コマンドの接頭部として **db2ts** を付けてください。

このコマンドを使用すると、独立型のテキスト検索サーバーを簡単に停止できます。そのサーバーがインストールされている環境で、付属スクリプトを使用して停止することもできます。インスタンス・サービスが既に停止している場合には、このコマンドはその状況をチェックしてユーザーに報告するだけです。

### 許可

インスタンス所有者。データベース特権は要求されていない

### コマンド構文

```

▶▶—STOP FOR TEXT—STATUS—
                        |
                        |—VERIFY—
  
```

### コマンド・パラメーター

#### STATUS

DB2 Text Search サーバーの状況を検査します。サーバーの **STARTED** または **STOPPED** 状況を示す詳細情報メッセージが戻されます。

#### VERIFY

DB2 Text Search サーバーの停止状況を検査します。コマンドが正常に実行されたことを示す場合は、標準メッセージおよび戻りコード **0** で終了します。それ以外の場合は、テキスト検索サーバーはゼロでないコードを戻して失敗を示します。

### 使用上の注意

- 現在実行中のコマンドの実行の中断を避けるため、**db2ts STOP FOR TEXT** コマンドを発行する前に、依然としてアクティブな **db2ts UPDATE INDEX FOR TEXT** コマンドのような他の管理コマンドがないことを確認してください。
- パーティション・データベース環境では、いずれか 1 つのパーティション・サーバー・ホストに対して、**STATUS** および **VERIFY** パラメーターを付けて **db2ts START FOR TEXT** コマンドを発行できます。
- 統合テキスト検索サーバーを使用している Windows プラットフォーム上のパーティション・データベース環境では、統合テキスト検索サーバー・ホスト・マシン上で **db2ts STOP FOR TEXT** コマンドを発行することにより、インスタンス・サービスを停止します。デフォルトでは、統合テキスト検索サーバー・ホスト・マシンは、最も小さい番号のデータベース・パーティション・サーバーのホストです。統合テキスト検索サーバー・ホスト・マシン上でこのコマンドを実行すれば、すべてのプロセスおよびサービスが確実に停止します。別のパーティショ

## **db2ts STOP FOR TEXT**

ン・サーバー・ホスト上でコマンドが実行される場合には、NET STOP などのコマンドを使用して、DB2TS サービスを停止する必要があります。

## db2ts UPDATE INDEX

**db2ts UPDATE INDEX** コマンドはテキスト検索索引 (DB2 Text Search のコレクション) を更新して、索引に関連付けられたテキスト列の現在の内容を反映します。更新が実行されている間も、検索は可能です。更新が完了するまでは、検索は部分的に更新された索引に対して実行されます。

実行するには、コマンド行でコマンドの前に **db2ts** を付ける必要があります。

### 許可

ステートメントの許可 ID が保持する特権には、SYSTS\_MGR ロールと以下の少なくとも 1 つの権限が含まれている必要があります。

- DATAACCESS 権限
- そのテキスト索引の定義されている表に対する CONTROL 特権。
- テキスト索引が定義されている基本表に対する、INDEX 特権および SELECT 特権。

また、初期更新の場合は、**CREATE TRIGGER** ステートメントで概略されている権限の要件が適用されます。

### 必要な接続

データベース

### コマンド構文

```

▶▶ UPDATE INDEX index-name FOR TEXT UPDATE OPTIONS
▶ | connection options |

```

#### connection options:

```

| CONNECT TO database-name USER username USING password |

```

### コマンド・パラメーター

#### UPDATE INDEX *index-name*

更新されるテキスト検索索引の名前を指定します。索引名は、DB2 索引の命名上の制約に従う必要があります。

#### UPDATE OPTIONS

更新オプションを指定する、タイプ VARCHAR(32K) の入力引数。オプションを指定しない場合、更新は無条件に開始されます。可能な値は次のとおりです。

UPDATE OPTIONS の値	説明
USING UPDATE MINIMUM	このオプションは、テキスト検索索引に定義されている UPDATE MINIMUM 値を使用するように強制して、指定された最小数の変更が発生したら更新を処理します。
FOR DATA REDISTRIBUTION	このオプションは、データ・パーティションが追加または削除され、後続のデータ再配分操作が完了した後に、パーティション・データベース内のテキスト検索索引をリフレッシュするように指定します。FOR DATA REDISTRIBUTION オプションでテキスト検索索引が更新されるまでは、検索結果に矛盾が生じる可能性があります。
ALLROWS	このオプションは、初期更新を無条件に試行するように指定します。
UPDATEAUTOCOMMIT	<p>このオプションは索引更新の回数を指定するもので、初期更新またはインクリメンタル更新のいずれかにおいてこの回数に達すると、コミットが実行され、それまでの作業が自動的に保存されます。</p> <ul style="list-style-type: none"> <li>• イニシャル更新では、データ更新をキャプチャーするトリガーがアクティブ化された後の基本表からの文書のバッチを、索引更新が処理します。文書更新の回数が COMMITCOUNT の数に達すると、サーバーは暫定コミットを実行します。処理されていない文書により生成されたログ項目は、ステージング表から除去されません。イニシャル・テキスト索引更新に UPDATEAUTOCOMMIT オプションを使用すると、実行時間が大幅に増加します。</li> <li>• インクリメンタル更新では、処理されているログ項目は、各暫定コミットで対応するステージング表から除去されます。COMMITCOUNT は更新される文書の数であり、ステージング表項目の数ではありません。</li> </ul>

**CONNECT TO** *database-name*

この節は、接続が確立される先のデータベースを指定します。データベースはローカル・システム上になければなりません。この節が指定されている場合は、環境変数 DB2DBDFT よりも優先されます。以下のステートメントすべてが該当する場合、この節は省略できます。

- DB2DBDFT 環境変数が有効なデータベース名に設定されている。
- コマンドを実行しているユーザーが、データベース・サーバーに接続するのに必要な権限を持っている。

**USER** *username* **USING** *password*

この節は、接続を確立するのに使用する許可名およびパスワードを指定します。

## 使用上の注意

DB2 のデータベース・オブジェクトおよび照会に適用されるすべての制限および命名規則は、DB2 Text Search のフィーチャーおよび照会にも適用されます。DB2 Text Search 関連の ID は、DB2 の命名規則に準拠している必要があります。さらに、追加の制限事項がいくつかあります。例えば、これらの ID は以下の形式のみになります。

```
[A-Za-z][A-Za-z0-9@#$_]*
```

または

```
"[A-Za-z ][A-Za-z0-9@#$_ ]*"
```

シノニム・ディクショナリーがテキスト索引用に作成されている場合には、ALLROWS および FOR DATA REDISTRIBUTION 更新オプションを実行すると、既存のコレクションからディクショナリーが削除されます。データベース・パーティションが追加された後には、テキスト索引に関連付けられた新しいコレクションを設けることができます。関連するすべてのコレクションのためのシノニム・ディクショナリーを再び追加する必要があります。

このコマンドは、すべての索引更新処理が完了するまで戻りません。この期間は、これから索引付けされる文書の数および既に索引付けされている文書の数に応じて異なります。索引のコレクション名は SYSIBMTS.TSCOLLECTIONNAMES ビュー(列 COLLECTIONNAME) から取られます。

複数のコマンドは、競合する可能性がある場合は、テキスト検索索引に対して同時に実行することはできません。競合するコマンドの実行中に、このコマンドを実行すると、エラーが発生してコマンドは失敗します。その後、コマンドの実行を再試行することができます。以下のコマンドは、一般的な競合コマンドの一部です。

- UPDATE INDEX
- CLEAR EVENTS FOR INDEX
- ALTER INDEX
- DROP INDEX
- DISABLE DATABASE FOR TEXT

**注:** 個々の文書エラーの場合、文書を訂正する必要があります。エラーがある文書の主キーは、索引のイベント表で参照できます。ユーザー表の対応する行が変更される場合、次の UPDATE INDEX コマンドはこれらの文書を再処理します。

データベースへの変更:

- イベント表に行を挿入します (DB2 Text Search からのパーサー・エラー情報を含む)。
- インクリメンタル更新の場合に、索引のステージング表から削除します。
- 最初の更新の前に、ユーザー・テキスト表にトリガーを作成します。
- コレクションが更新されます。
- 新規または変更された文書が構文解析および索引付けされます。
- 削除された文書が索引から破棄されます。

---

## 付録 A. 命名規則

## 命名規則

データベースおよび表などのデータベース・マネージャー・オブジェクトを命名する際の命名規則です。

データベース、表などのデータベース・マネージャー・オブジェクトの命名の際には、次のような規則が適用されます。

- データベース・マネージャー・オブジェクトの名前を表す文字ストリングには、a から z、A から Z、0 から 9、@、#、および \$ などが含まれます。
- 特に注記のない限り、名前は小文字で入力して構いません。ただし、データベース・マネージャーはそれらを大文字と見なして処理します。

ただし、システム・ネットワーク体系 (SNA) 下の名前を表す文字ストリングはこの規則の例外です。SNA は通信プロトコルとしてサポートされなくなりました。LU 名 (partner\_lu および local\_lu) など、値の多くは大文字小文字を区別します。こうした名前は、それらの用語に対応する SNA 定義に出ているとおりに入力してください。

- データベース名やデータベース別名は、前に説明した集合内の 1 つから 8 つの文字、数字、キーボード文字を含む固有の文字ストリングです。

データベースはシステム内にカタログされており、ローカル・データベース・ディレクトリーの別名が一つのフィールドに、元の名前が別のフィールドに入っています。ほとんどの機能の場合、データベース・マネージャーは、データベース・ディレクトリーの別名フィールドに入力された名前を使用します。(ただし、CHANGE DATABASE COMMENT および CREATE DATABASE は例外です。この場合は、ディレクトリー・パスを指定しなければなりません。)

- 表やビューの名前または別名は、長さが 1 バイトから 128 バイトまでの固有な文字ストリングで構成される SQL ID です。列名の長さは 1 バイトから 128 バイトまでです。

完全修飾された表名は、*schema.tablename* から成っています。スキーマ (schema) は固有のユーザー ID で、その下に表が作成されます。宣言一時表のスキーマ名は SESSION でなければなりません。

- ノード・ディレクトリーでカタログされるリモート・ノードのローカル別名の長さは、8 文字を超えてはなりません。
- ストリングの最初の文字はアルファベット、@、#、または \$ にしなければなりません。数字や文字順序列の SYS、DBM、IBM などは不可です。

ユーザー ID や認証 ID の命名の際には、次のような規則が適用されます。

- データベース・マネージャー・オブジェクトの名前を表す文字ストリングには、a から z、A から Z、0 から 9、@、#、および \$ などが含まれます。
- ユーザー ID およびグループには `_!%(){}-.^` という追加の文字を任意に含めることができます (ただし、これらの文字がセキュリティー・プラグインによってサポートされていることが前提になります)。
- `!%(){}-.^` という文字のいずれかを含むユーザー ID およびグループをコマンド行プロセッサで入力するときには、引用符で区切る必要があります。
- ストリングの最初の文字はアルファベット、@、#、または \$ にしなければなりません。数字や文字順序列の SYS、DBM、IBM などは不可です。



- 認証 ID の長さは、128 バイトを超えてはなりません。
- グループ ID の長さは、128 バイトを超えてはなりません。



---

## 付録 B. ファイル・タイプ修飾子および区切り文字

## エクスポート・ユーティリティ、インポート・ユーティリティ、ロード・ユーティリティで使用するファイル・タイプ修飾子

下記のリンクをクリックすると、コマンド・トピックの先頭に移動するので、そこから個々のファイル・タイプ修飾子のセクションへのクイック・リンクを見つけることができます。

200 ページの『EXPORT』

290 ページの『IMPORT』

425 ページの『LOAD』

## データ移動時の区切り文字に関する考慮事項

区切り文字が使用されている ASCII (DEL) ファイルを移動する場合は、区切り文字の認識方法にかかわる問題のために、移動するデータをうっかり変更してしまわないようにする必要があります。このようなエラーを回避するために、DB2 には、いくつかの制約が設定されており、そのためのファイル・タイプ修飾子が用意されています。

### 区切り文字に関する制約事項

特定の区切り文字が、移動するデータの一部として処理される事態を回避するために、いくつかの制約が設定されています。第 1 に、区切り文字は、相互に排他的です。第 2 に、区切り文字として、2 進ゼロ、改行文字、復帰文字、ブランク・スペースを使用することはできません。さらに、デフォルトの小数点 (.) をストリング区切り文字として使用することはできません。最後に、DBCS 環境では、ストリング区切り文字としてパイプ (|) がサポートされていません。

ASCII 系のコード・ページと EBCDIC 系のコード・ページでは、以下の文字の指定方法が異なります。

- EBCDIC MBCS データ・ファイルでは、シフトイン (0x0F) 文字とシフトアウト (0x0E) 文字を区切り文字として使用できません。
- MBCS コード・ページ、EUC コード・ページ、DBCS コード・ページでは、0x40 より大きな文字を区切り文字として使用できません。ただし、EBCDIC MBCS データのデフォルトの小数点 (0x4b) は例外です。
- ASCII コード・ページまたは EBCDIC MBCS コード・ページのデータ・ファイルのデフォルトの区切り文字は、以下のとおりです。
  - ストリング区切り文字: "(0x22、二重引用符)
  - 列区切り文字: ,(0x2c、コンマ)
- EBCDIC SBCS コード・ページのデータ・ファイルのデフォルトの区切り文字は、以下のとおりです。
  - ストリング区切り文字: "(0x7F、二重引用符)
  - 列区切り文字: ,(0x6B、コンマ)
- ASCII データ・ファイルのデフォルトの小数点は、0x2e (ピリオド) です。
- EBCDIC データ・ファイルのデフォルトの小数点は、0x4B (ピリオド) です。
- サーバーのコード・ページとクライアントのコード・ページが異なる場合は、デフォルト以外の区切り文字の 16 進表記を指定することをお勧めします。以下に例を示します。

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

### データ移動時の区切り文字に関する問題

#### 二重のストリング区切り文字

DEL ファイルの文字ベース・フィールドの場合は、フィールド内にストリング区切り文字が見つかったら、そのすべての出現箇所が二重のストリング区切り文字で表記される、というのがデフォルトの動作です。例えば、ストリング区切り文字が二重引用符の場合に、I am 6" tall. というテキストをエクスポートすると、DEL ファイルの出力テキストは、"I am 6"" tall." に

## データ移動時の区切り文字に関する考慮事項

なります。その逆に、DEL ファイルの入力テキストが "What a "nice" day!" になっていると、そのテキストは、What a "nice" day! という形でインポートされます。

### **nodoubledel**

インポート/エクスポート/ロード・ユーティリティーで二重のストリング区切り文字の動作を無効にするには、`nodoubledel` ファイル・タイプ修飾子を指定します。ただし、二重のストリング区切り文字の動作が存在しているのは構文解析エラーを回避するためである、という点は忘れないようにする必要があります。エクスポートで `nodoubledel` を使用するときは、ストリング区切り文字が文字フィールドにあっても、そのストリング区切り文字は二重になりません。インポートとロードで `nodoubledel` を使用すると、二重のストリング区切り文字は、ストリング区切り文字のリテラル・インスタンスとして解釈されなくなります。

### **nochardel**

エクスポートで `nochardel` ファイル・タイプ修飾子を使用すると、文字フィールドがストリング区切り文字で囲まれなくなります。インポートとロードで `nochardel` を使用すると、ストリング区切り文字は、特殊文字としてではなく実際のデータとして解釈されます。

### **chardel**

その他のファイル・タイプ修飾子を使用して、デフォルトの区切り文字とデータの混同を手動で回避することも可能です。`chardel` ファイル・タイプ修飾子では、二重引用符 (デフォルト) の代わりに使用する文字ストリング区切り文字として、`x` という 1 文字を指定します。

### **coldel**

同じように、列区切り文字としてデフォルトのコンマを使用したくない場合は、`coldel` を使用できます。この修飾子では、列データ区切り文字として、`x` という 1 文字を指定します。

### **delprioritychar**

DEL ファイルの移動に伴うもう 1 つの問題は、区切り文字の正しい優先順位を維持することです。区切り文字のデフォルトの優先順位は、行、文字 (桁)、列です。ただし、中には、文字 (桁)、行、列という優先順位に依存するアプリケーションもあります。例えば、デフォルトの優先順位を使用する次のような DEL データ・ファイルがあるとします。

```
"Vincent <row delimiter> is a manager",<row delimiter>
```

このデータは、Vincent と is a manager という 2 つの行として解釈されます。行区切り文字 (<row delimiter>) の方がストリング区切り文字 (") よりも優先順位が高いからです。一方、`delprioritychar` を使用すると、ストリング区切り文字 (") の方が行区切り文字 (<row delimiter>) よりも優先順位が高くなり、同じ DEL ファイルでも、この場合は Vincent is a manager という 1 行として (正しく) 解釈されます。

---

## 付録 C. DB2 技術情報の概説

DB2 技術情報は、さまざまな方法でアクセスすることが可能な、各種形式で入手できます。

DB2 技術情報は、以下のツールと方法を介して利用できます。

- DB2インフォメーション・センター
  - トピック (タスク、概念、およびリファレンス・トピック)
  - サンプル・プログラム
  - チュートリアル
- DB2 資料
  - PDF ファイル (ダウンロード可能)
  - PDF ファイル (DB2 PDF DVD に含まれる)
  - 印刷資料
- コマンド行ヘルプ
  - コマンド・ヘルプ
  - メッセージ・ヘルプ

**注:** DB2 インフォメーション・センターのトピックは、PDF やハードコピー資料よりも頻繁に更新されます。最新の情報を入手するには、資料の更新が発行されたときにそれをインストールするか、[ibm.com](http://ibm.com) にある DB2 インフォメーション・センターを参照してください。

技術資料、ホワイト・ペーパー、IBM Redbooks® 資料などのその他の DB2 技術情報には、オンライン ([ibm.com](http://ibm.com)) でアクセスできます。DB2 Information Management ソフトウェア・ライブラリー・サイト (<http://www.ibm.com/software/data/sw-library/>) にアクセスしてください。

### 資料についてのフィードバック

DB2 の資料についてのお客様からの貴重なご意見をお待ちしています。DB2 の資料を改善するための提案については、[db2docs@ca.ibm.com](mailto:db2docs@ca.ibm.com) まで E メールを送信してください。DB2 の資料チームは、お客様からのフィードバックすべてに目を通しますが、直接お客様に返答することはありません。お客様が関心をお持ちの内容について、可能な限り具体的な例を提供してください。特定のトピックまたはヘルプ・ファイルについてのフィードバックを提供する場合は、そのトピック・タイトルおよび URL を含めてください。

DB2 お客様サポートに連絡する場合には、この E メール・アドレスを使用しないでください。資料を参照しても、DB2 の技術的な問題が解決しない場合は、お近くの IBM サービス・センターにお問い合わせください。

## DB2 テクニカル・ライブラリー (ハードコピーまたは PDF 形式)

以下の表は、IBM Publications Center ([www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss](http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss)) から利用できる DB2 ライブラリーについて説明しています。英語および翻訳された DB2 バージョン 10.1 のマニュアル (PDF 形式) は、[www.ibm.com/support/docview.wss?rs=71&uid=swg27009474](http://www.ibm.com/support/docview.wss?rs=71&uid=swg27009474) からダウンロードできます。

この表には印刷資料が入手可能かどうかを示されていますが、国または地域によっては入手できない場合があります。

資料番号は、資料が更新される度に大きくなります。資料を参照する際は、以下にリストされている最新版であることを確認してください。

注: DB2 インフォメーション・センターは、PDF やハードコピー資料よりも頻繁に更新されます。

表 56. DB2 の技術情報

資料名	資料番号	印刷資料が入手可能かどうか	最終更新
管理 API リファレンス	SA88-4671-00	入手可能	2012 年 4 月
管理ルーチンおよびビ ュー	SA88-4672-01	入手不可	2013 年 1 月
コール・レベル・イン ターフェース ガイドお よびリファレンス 第 1 巻	SA88-4676-01	入手可能	2013 年 1 月
コール・レベル・イン ターフェース ガイドお よびリファレンス 第 2 巻	SA88-4677-01	入手可能	2013 年 1 月
コマンド・リファレン ス	SA88-4673-01	入手可能	2013 年 1 月
データベース: 管理の 概念および構成リファ レンス	SA88-4662-01	入手可能	2013 年 1 月
データ移動キューティ リ: ガイドおよび リファレンス	SA88-4693-01	入手可能	2013 年 1 月
データベースのモニタ リング ガイドおよび リファレンス	SA88-4663-01	入手可能	2013 年 1 月
データ・リカバリーと 高可用性 ガイドおよび リファレンス	SA88-4694-01	入手可能	2013 年 1 月
データベース・セキュ リティー・ガイド	SA88-4695-01	入手可能	2013 年 1 月



## DB2 テクニカル・ライブラリー (ハードコピーまたは PDF 形式)

表 56. DB2 の技術情報 (続き)

資料名	資料番号	印刷資料が入手可能かどうか	最終更新
DB2 ワークロード管理ガイドおよびリファレンス	SA88-4685-01	入手可能	2013 年 1 月
ADO.NET および OLE DB アプリケーションの開発	SA88-4665-01	入手可能	2013 年 1 月
組み込み SQL アプリケーションの開発	SA88-4666-01	入手可能	2013 年 1 月
Java アプリケーションの開発	SA88-4669-01	入手可能	2013 年 1 月
Perl、PHP、Python および Ruby on Rails アプリケーションの開発	SA88-4670-00	入手不可	2012 年 4 月
IBM データ・サーバー用の RDF アプリケーション開発	SA88-5083-00	入手可能	2013 年 1 月
SQL および外部ルーチンの開発	SA88-4667-01	入手可能	2013 年 1 月
データベース・アプリケーション開発の基礎	GI88-4279-01	入手可能	2013 年 1 月
DB2 インストールおよび管理 概説 (Linux および Windows 版)	GI88-4280-00	入手可能	2012 年 4 月
グローバリゼーション・ガイド	SA88-4696-00	入手可能	2012 年 4 月
DB2 サーバー機能 インストール	GA88-4679-01	入手可能	2013 年 1 月
IBM データ・サーバー・クライアント機能インストール	GA88-4680-00	入手不可	2012 年 4 月
メッセージ・リファレンス 第 1 巻	SA88-4688-01	入手不可	2013 年 1 月
メッセージ・リファレンス 第 2 巻	SA88-4689-01	入手不可	2013 年 1 月
Net Search Extender 管理およびユーザズ・ガイド	SA88-4691-01	入手不可	2013 年 1 月
パーティションおよびクラスタリングのガイド	SA88-4697-01	入手可能	2013 年 1 月
Preparation Guide for DB2 10.1 Fundamentals Exam 610	SC27-4540-00	入手不可	2013 年 1 月

## DB2 テクニカル・ライブラリー (ハードコピーまたは PDF 形式)

表 56. DB2 の技術情報 (続き)

資料名	資料番号	印刷資料が入手可能 かどうか	最終更新
<i>Preparation Guide for DB2 10.1 DBA for Linux, UNIX, and Windows Exam 611</i>	SC27-4541-00	入手不可	2013 年 1 月
<i>pureXML ガイド</i>	SA88-4686-01	入手可能	2013 年 1 月
<i>Spatial Extender ユーザーズ・ガイドおよびリファレンス</i>	SA88-4690-00	入手不可	2012 年 4 月
<i>SQL プロシージャ言語: アプリケーションのイネーブルメントおよびサポート</i>	SA88-4668-01	入手可能	2013 年 1 月
<i>SQL リファレンス 第 1 巻</i>	SA88-4674-01	入手可能	2013 年 1 月
<i>SQL リファレンス 第 2 巻</i>	SA88-4675-01	入手可能	2013 年 1 月
<i>Text Search ガイド</i>	SA88-4692-01	入手可能	2013 年 1 月
<i>問題判別およびデータベース・パフォーマンスのチューニング</i>	SA88-4664-01	入手可能	2013 年 1 月
<i>DB2 バージョン 10.1 へのアップグレード</i>	SA88-4678-01	入手可能	2013 年 1 月
<i>DB2 バージョン 10.1 の新機能</i>	SA88-4684-01	入手可能	2013 年 1 月
<i>XQuery リファレンス</i>	SA88-4687-01	入手不可	2013 年 1 月

表 57. DB2 Connect 固有の技術情報

資料名	資料番号	印刷資料が入手可能 かどうか	最終更新
<i>DB2 Connect Personal Edition</i> インストールおよび構成	SA88-4681-00	入手可能	2012 年 4 月
<i>DB2 Connect サーバー機能</i> インストールおよび構成	SA88-4682-01	入手可能	2013 年 1 月
<i>DB2 Connect ユーザーズ・ガイド</i>	SA88-4683-01	入手可能	2013 年 1 月

## コマンド行プロセッサから SQL 状態ヘルプを表示する

DB2 製品は、SQL ステートメントの結果として生じる可能性がある状態に対応した SQLSTATE 値を戻します。SQLSTATE ヘルプは、SQL 状態および SQL 状態クラス・コードの意味を説明します。

## 手順

SQL 状態ヘルプを開始するには、コマンド行プロセッサを開いて以下のように入力します。

```
? sqlstate または ? class code
```

ここで、*sqlstate* は有効な 5 桁の SQL 状態を、*class code* は SQL 状態の最初の 2 桁を表します。

例えば、? 08003 を指定すると SQL 状態 08003 のヘルプが表示され、? 08 を指定するとクラス・コード 08 のヘルプが表示されます。

---

## 異なるバージョンの DB2 インフォメーション・センターへのアクセス

他のバージョンの DB2 製品の資料は、ibm.com<sup>®</sup> のそれぞれのインフォメーション・センターにあります。

### このタスクについて

DB2 バージョン 10.1 のトピックを扱っている DB2 インフォメーション・センターの URL は、<http://publib.boulder.ibm.com/infocenter/db2luw/v10r1> です。

DB2 バージョン 9.8 のトピックを扱っている DB2 インフォメーション・センターの URL は、<http://publib.boulder.ibm.com/infocenter/db2luw/v9r8/> です。

DB2 バージョン 9.7 のトピックを扱っている DB2 インフォメーション・センターの URL は、<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/> です。

DB2 バージョン 9.5 のトピックを扱っている DB2 インフォメーション・センターの URL は、<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5> です。

DB2 バージョン 9.1 のトピックを扱っている DB2 インフォメーション・センターの URL は、<http://publib.boulder.ibm.com/infocenter/db2luw/v9/> です。

DB2 バージョン 8 のトピックについては、DB2 インフォメーション・センターの URL (<http://publib.boulder.ibm.com/infocenter/db2luw/v8/>) を参照してください。

---

## コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの更新

ローカルにインストールした DB2 インフォメーション・センターは、定期的な更新する必要があります。

### 始める前に

DB2 バージョン 10.1 インフォメーション・センターが既にインストール済みである必要があります。詳しくは、「DB2 サーバー機能 インストール」の『DB2 セットアップ・ウィザードによる DB2 インフォメーション・センターのインストール』のトピックを参照してください。インフォメーション・センターのインストールに適用されるすべての前提条件と制約事項は、インフォメーション・センターの更新にも適用されます。

## このタスクについて

既存の DB2 インフォメーション・センターは、自動で更新することも手動で更新することもできます。

- 自動更新は、既存のインフォメーション・センターのフィーチャーと言語を更新します。自動更新を使用すると、手動更新と比べて、更新中にインフォメーション・センターが使用できなくなる時間が短くなるというメリットがあります。さらに、自動更新は、定期的に行う他のバッチ・ジョブの一部として実行されるように設定することができます。
- 手動更新は、既存のインフォメーション・センターのフィーチャーと言語の更新に使用できます。自動更新は更新処理中のダウン時間を減らすことができますが、フィーチャーまたは言語を追加する場合は手動処理を使用する必要があります。例えば、ローカルのインフォメーション・センターが最初は英語とフランス語でインストールされており、その後ドイツ語もインストールすることにした場合、手動更新でドイツ語をインストールし、同時に、既存のインフォメーション・センターのフィーチャーおよび言語を更新できます。しかし、手動更新ではインフォメーション・センターを手動で停止、更新、再始動する必要があります。更新処理の間はずっと、インフォメーション・センターは使用できなくなります。自動更新処理では、インフォメーション・センターは、更新を行った後に、インフォメーション・センターを再始動するための停止が発生するだけで済みます。

このトピックでは、自動更新のプロセスを詳しく説明しています。手動更新の手順については、『コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの手動更新』のトピックを参照してください。

## 手順

コンピューターまたはイントラネット・サーバーにインストールされている DB2 インフォメーション・センターを自動更新する手順を以下に示します。

1. Linux オペレーティング・システムの場合、次のようにします。
  - a. インフォメーション・センターがインストールされているパスにナビゲートします。デフォルトでは、DB2 インフォメーション・センターは、`/opt/ibm/db2ic/V10.1` ディレクトリーにインストールされています。
  - b. インストール・ディレクトリーから `doc/bin` ディレクトリーにナビゲートします。
  - c. 次のように `update-ic` スクリプトを実行します。

```
update-ic
```
2. Windows オペレーティング・システムの場合、次のようにします。
  - a. コマンド・ウィンドウを開きます。
  - b. インフォメーション・センターがインストールされているパスにナビゲートします。デフォルトでは、DB2 インフォメーション・センターは、`<Program Files>%IBM%DB2 Information Center%バージョン 10.1` ディレクトリーにインストールされています (`<Program Files>` は「Program Files」ディレクトリーのロケーション)。

- c. インストール・ディレクトリーから doc¥bin ディレクトリーにナビゲートします。
- d. 次のように update-ic.bat ファイルを実行します。

```
update-ic.bat
```

## タスクの結果

DB2 インフォメーション・センターが自動的に再始動します。更新が入手可能な場合、インフォメーション・センターに、更新された新しいトピックが表示されます。インフォメーション・センターの更新が入手可能でなかった場合、メッセージがログに追加されます。ログ・ファイルは、doc¥eclipse¥configuration ディレクトリーにあります。ログ・ファイル名はランダムに生成された名前です。例えば、1239053440785.log のようになります。

---

## コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの手動更新

DB2 インフォメーション・センターをローカルにインストールしている場合は、IBM から資料の更新を入手してインストールすることができます。

### このタスクについて

ローカルにインストールされた **DB2 インフォメーション・センター** を手動で更新するには、以下のことを行う必要があります。

1. コンピューター上の **DB2 インフォメーション・センター** を停止し、インフォメーション・センターをスタンドアロン・モードで再始動します。インフォメーション・センターをスタンドアロン・モードで実行すると、ネットワーク上の他のユーザーがそのインフォメーション・センターにアクセスできなくなります。これで、更新を適用できるようになります。DB2 インフォメーション・センターのワークステーション・バージョンは、常にスタンドアロン・モードで実行されます。を参照してください。
2. 「更新」機能を使用することにより、どんな更新が利用できるかを確認します。インストールしなければならない更新がある場合は、「更新」機能を使用してそれを入手およびインストールできます。

**注:** ご使用の環境において、インターネットに接続されていないマシンに **DB2 インフォメーション・センター** の更新をインストールする必要がある場合、インターネットに接続されていて **DB2 インフォメーション・センター** がインストールされているマシンを使用して、更新サイトをローカル・ファイル・システムにミラーリングしてください。ネットワーク上の多数のユーザーが資料の更新をインストールする場合にも、更新サイトをローカルにミラーリングして、更新サイト用のプロキシを作成することにより、個々のユーザーが更新を実行するのに要する時間を短縮できます。

更新パッケージが入手可能な場合、「更新」機能を使用してパッケージを入手します。ただし、「更新」機能は、スタンドアロン・モードでのみ使用できます。

3. スタンドアロンのインフォメーション・センターを停止し、コンピューター上の **DB2 インフォメーション・センター** を再開します。

注: Windows 2008、Windows Vista (およびそれ以上) では、このセクションの後の部分でリストされているコマンドは管理者として実行する必要があります。完全な管理者特権でコマンド・プロンプトまたはグラフィカル・ツールを開くには、ショートカットを右クリックしてから、「**管理者として実行**」を選択します。

## 手順

コンピューターまたはイントラネット・サーバーにインストール済みの **DB2** インフォメーション・センター を更新するには、以下のようにします。

1. **DB2** インフォメーション・センター を停止します。
    - Windows では、「**スタート**」 > 「**コントロール パネル**」 > 「**管理ツール**」 > 「**サービス**」をクリックします。次に、「**DB2 インフォメーション・センター**」サービスを右クリックして「**停止**」を選択します。
    - Linux では、以下のコマンドを入力します。

```
/etc/init.d/db2icdv10 stop
```
  2. インフォメーション・センターをスタンドアロン・モードで開始します。
    - Windows の場合:
      - a. コマンド・ウィンドウを開きます。
      - b. インフォメーション・センターがインストールされているパスにナビゲートします。デフォルトでは、**DB2** インフォメーション・センター は、`Program_Files\IBM\DB2 Information Center\バージョン 10.1` ディレクトリーにインストールされています (`Program_Files` は Program Files ディレクトリーのロケーション)。
      - c. インストール・ディレクトリーから `doc\bin` ディレクトリーにナビゲートします。
      - d. 次のように `help_start.bat` ファイルを実行します。

```
help_start.bat
```
    - Linux の場合:
      - a. インフォメーション・センターがインストールされているパスにナビゲートします。デフォルトでは、**DB2** インフォメーション・センター は、`/opt/ibm/db2ic/V10.1` ディレクトリーにインストールされています。
      - b. インストール・ディレクトリーから `doc/bin` ディレクトリーにナビゲートします。
      - c. 次のように `help_start` スクリプトを実行します。

```
help_start
```
- システムのデフォルト Web ブラウザーが開き、スタンドアロンのインフォメーション・センターが表示されます。
3. 「**更新**」ボタン (🔄) をクリックします。(ブラウザーで JavaScript が有効になっている必要があります。) インフォメーション・センターの右側のパネルで、「**更新の検索**」をクリックします。既存の文書に対する更新のリストが表示されます。
  4. インストール・プロセスを開始するには、インストールする更新をチェックして選択し、「**更新のインストール**」をクリックします。
  5. インストール・プロセスが完了したら、「**完了**」をクリックします。

6. 次のようにして、スタンドアロンのインフォメーション・センターを停止します。

- Windows の場合は、インストール・ディレクトリーの `doc\bin` ディレクトリーにナビゲートしてから、次のように `help_end.bat` ファイルを実行します。

```
help_end.bat
```

注: `help_end` バッチ・ファイルには、`help_start` バッチ・ファイルを使用して開始したプロセスを安全に停止するのに必要なコマンドが含まれています。`help_start.bat` は、Ctrl-C や他の方法を使用して停止しないでください。

- Linux の場合は、インストール・ディレクトリーの `doc/bin` ディレクトリーにナビゲートしてから、次のように `help_end` スクリプトを実行します。

```
help_end
```

注: `help_end` スクリプトには、`help_start` スクリプトを使用して開始したプロセスを安全に停止するのに必要なコマンドが含まれています。他の方法を使用して、`help_start` スクリプトを停止しないでください。

7. **DB2 インフォメーション・センター** を再開します。

- Windows では、「スタート」 > 「コントロール パネル」 > 「管理ツール」 > 「サービス」をクリックします。次に、「**DB2 インフォメーション・センター**」サービスを右クリックして「開始」を選択します。

- Linux では、以下のコマンドを入力します。

```
/etc/init.d/db2icdv10 start
```

## タスクの結果

更新された **DB2 インフォメーション・センター** に、更新された新しいトピックが表示されます。

---

## DB2 チュートリアル

DB2 チュートリアルは、DB2 データベース製品のさまざまな機能について学習するための支援となります。この演習をとおして段階的に学習することができます。

### はじめに

インフォメーション・センター (<http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/>) から、このチュートリアルの XHTML 版を表示できます。

演習の中で、サンプル・データまたはサンプル・コードを使用する場合があります。個々のタスクの前提条件については、チュートリアルを参照してください。

### DB2 チュートリアル

チュートリアルを表示するには、タイトルをクリックします。

「*pureXML* ガイド」の『**pureXML**®』

XML データを保管し、ネイティブ XML データ・ストアに対して基本的な操作を実行できるように、DB2 データベースをセットアップします。



### DB2 トラブルシューティング情報

DB2 データベース製品を使用する際に役立つ、トラブルシューティングおよび問題判別に関する広範囲な情報を利用できます。

#### DB2 の資料

トラブルシューティング情報は、「問題判別およびデータベース・パフォーマンスのチューニング」または *DB2* インフォメーション・センター の『データベースの基本』セクションにあります。ここには、以下の情報が記載されています。

- DB2 診断ツールおよびユーティリティーを使用した、問題の切り分け方法および識別方法に関する情報。
- 最も一般的な問題のうち、いくつかの解決方法。
- DB2 データベース製品で発生する可能性のある、その他の問題の解決に役立つアドバイス。

#### IBM サポート・ポータル

現在問題が発生していて、考えられる原因とソリューションを見つけるには、IBM サポート・ポータルを参照してください。Technical Support サイトには、最新の DB2 資料、TechNotes、プログラム診断依頼書 (APAR またはバグ修正)、フィックスパック、およびその他のリソースへのリンクが用意されています。この知識ベースを活用して、問題に対する有効なソリューションを探し出すことができます。

IBM サポート・ポータル ([http://www.ibm.com/support/entry/portal/Overview/Software/Information\\_Management/DB2\\_for\\_Linux,\\_UNIX\\_and\\_Windows](http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/DB2_for_Linux,_UNIX_and_Windows)) にアクセスしてください。

---

## ご利用条件

これらの資料は、以下の条件に同意していただける場合に限りご使用いただけます。

**適用度:** これらのご利用条件は、IBM Web サイトのあらゆるご利用条件に追加で適用されるものです。

**個人使用:** これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、非商業的な個人による使用目的に限り複製することができます。ただし、IBM の明示的な承諾をえずに、これらの資料またはその一部について、二次的著作物を作成したり、配布 (頒布、送信を含む) または表示 (上映を含む) することはできません。

**商業的使用:** これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することができます。ただし、IBM の明示的な承諾をえずにこれらの資料の二次的著作物を作成したり、お客様の企業外で資料またはその一部を複製、配布、または表示することはできません。

**権利:** ここで明示的に許可されているもの以外に、資料や資料内に含まれる情報、データ、ソフトウェア、またはその他の知的所有権に対するいかなる許可、ライセンス、または権利を明示的にも黙示的にも付与するものではありません。



資料の使用が IBM の利益を損なうと判断された場合や、上記の条件が適切に守られていないと判断された場合、IBM はいつでも自らの判断により、ここで与えた許可を撤回できるものとさせていただきます。

お客様がこの情報をダウンロード、輸出、または再輸出する際には、米国のすべての輸出入関連法規を含む、すべての関連法規を遵守するものとします。

IBM は、これらの資料の内容についていかなる保証もしません。これらの資料は、特定物として現存するままの状態を提供され、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任なしで提供されます。

**IBM の商標:** IBM、IBM ロゴおよび [ibm.com](http://www.ibm.com) は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。



---

## 付録 D. 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。IBM 以外の製品に関する情報は、本書の最初の発行時点で入手可能な情報に基づいており、変更される場合があります。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510  
東京都中央区日本橋箱崎町19番21号  
日本アイ・ビー・エム株式会社  
法務・知的財産  
知的財産権ライセンス渉外

**以下の保証は、国または地域の法律に沿わない場合は、適用されません。** IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Canada Limited  
U59/3600  
3600 Steeles Avenue East  
Markham, Ontario L3R 9Z7  
CANADA

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、

利便性もしくは機能性があることをほのめかしたり、保証することはできません。サンプル・プログラムは、現存するままの状態を提供されるものであり、いかなる種類の保証も提供されません。IBM は、これらのサンプル・プログラムの使用から生ずるいかなる損害に対しても責任を負いません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生した創作物には、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。 © Copyright IBM Corp. \_年を入れる\_. All rights reserved.

## 商標

IBM、IBM ロゴおよび [ibm.com](http://ibm.com) は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。

以下は、それぞれ各社の商標または登録商標です。

- Linux は、Linus Torvalds の米国およびその他の国における商標です。
- Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。
- UNIX は The Open Group の米国およびその他の国における登録商標です。
- インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Celeron、Intel SpeedStep、Itanium、Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。
- Microsoft、Windows、Windows NT、および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。



# 索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

## [ア行]

アクセス・パス  
最適化 648  
アンカタログ  
システム・データベース・ディレクトリー 709  
データベース項目 709  
ホスト DCS データベース項目 711  
異常終了  
RESTART DATABASE コマンド 605  
一時ファイル  
ロード・ユーティリティ  
スペース所要量 425  
イベント・モニター  
アクティビティ・イベント・モニター・データ・ツールの  
キャプチャー 907  
db2caem ツール・コマンド 907  
db2evmon コマンド 995  
イベント・モニターのターゲット表定義の生成コマンド 997  
イメージ  
サイズの削減 1076  
インスタンスからのデータベース・パーティション・サーバ  
のドロップ・コマンド 1150  
インスタンス作成インターフェースの開始コマンド 1078  
インスタンスのアップグレード・コマンド 1090  
インスタンスの更新コマンド 1080  
インスタンスの作成コマンド 1055  
インスタンスの自動開始コマンド 1051  
インスタンスの除去コマンド 1065  
インスタンスのリスト・コマンド 1071  
インストール済み DB2 製品の更新コマンド 1382  
インストール済みの DB2 製品およびフィーチャーのリスト・  
コマンド 1118  
インストール前提条件の検査ツール・コマンド 1273  
インストール・イメージ  
サイズの削減 1076  
インポート  
データ 290  
エクスポート  
データ  
ファイル・タイプ修飾子 200  
EXPORT コマンド 200  
エラー  
チェックサム  
データベース構成ファイル 599, 734  
データベース・マネージャー構成ファイル 595

応答ファイル  
生成プログラム  
db2rspgn コマンド 1288

## [カ行]

カーソル固定 (CS)  
変更 145  
カーネル・パラメーター値のためのユーティリティ・コマン  
ド 1156  
開始  
DB2 1303  
カタログ作成  
データベース 122  
ホスト・データベース 126  
環境変数  
DB2OPTIONS 11  
監査機能  
db2audit コマンド 884  
行継続文字  
コマンド行プロセッサ (CLP) 2  
強調表記規則 x  
区切り文字  
データ移動時の制約事項 1431  
変更 1431  
組み込み SQL アプリケーション  
ランタイム・ライブラリー検索パス 919  
継続文字  
コマンド行プロセッサ (CLP) 2  
現行 DAS レベルの表示コマンド 966  
検索パス  
変更 922  
検査結果のフォーマット・コマンド 1075  
コード・ページ  
EXPORT コマンド 200  
IMPORT コマンド 290  
コード・ページ・ファイル・タイプ修飾子 425  
更新  
DAS の更新コマンド 848  
DB2 インフォメーション・センター 1437, 1439  
構成  
管理  
リセット、デフォルトに 595  
例 215  
データベース  
リセット、デフォルトに 599  
例 229  
UPDATE DATABASE CONFIGURATION コマンド 734  
データベース・マネージャー 242  
CLI 223

構文図

見方 xi

コマンド

構文ヘルプ 67

出力のリダイレクト 2

ACTIVATE DATABASE 72

ADD CONTACT 74

ADD CONTACTGROUP 75

ADD DBPARTITIONNUM 76

ARCHIVE LOG 81

ATTACH 84

AUTOCONFIGURE 86

BACKUP DATABASE 89

BIND 100

CATALOG DATABASE 122

CATALOG DCS DATABASE 126

CATALOG LDAP DATABASE 129

CATALOG LDAP NODE 132

CATALOG LOCAL NODE 133

CATALOG NAMED PIPE NODE 135

CATALOG ODBC DATA SOURCE 137

CATALOG TCP/IP NODE 139

CHANGE DATABASE COMMENT 143

CHANGE ISOLATION LEVEL 145

CLPPlus

サマリー 753

ACCEPT 759

APPEND 762

BREAK 763

BTITLE 764

CHANGE 766

CLEAR 768

COLUMN 770

COMPUTE 775

CONNECT 777

COPY 780

DEFINE 783

DEL 784

DESCRIBE 786

DISCONNECT 791

EDIT 792

EXECUTE 793

EXIT 796

EXPLAIN PLAN 798

GET 799

HELP 800

HOST 801

IMPORT 802

INPUT 804

LIST 805

PAUSE 807

PRINT 808

PROMPT 809

QUIT 810

REMARK 811

REORGCHK 812

コマンド (続き)

CLPPlus (続き)

REPFOOTER 814

REPHEADER 816

RUN 818

SAVE 819

SET 820

SHOW 831

SPOOL 830

START 832

TTITLE 833

UNDEFINE 835

WHENEVER OSERROR 836

WHENEVER SQLERROR 838

! 755

. 754

/ 756

@ 757

@@ 758

CREATE DATABASE 148

CREATE TOOLS CATALOG 168

dasauto 842

dasprt 843

dasdrop 845

dasmigr

DAS のマイグレーション 846

dasupdt 848

db2 9

db2acsutil 860

db2addicons 864

db2admin 865

db2adutl

詳細 867

db2advis 877

db2audit 884

db2batch 895

db2bfd 906

db2caem 907

db2cap 911

db2cfexp 916

db2cfimp 918

db2chglbpath 919

db2chgpath 922

db2ckbcp 924

db2cklog 929

db2ckrst 932

db2ckupgrade

詳細 934

db2cli 937

db2cmd 954

db2cptsa 955

db2dart

詳細 957

db2daslevel 966

db2dclgn 967



コマンド (続き)

db2diag  
 詳細 970  
 db2drdat  
 詳細 989  
 db2drvmp 991  
 db2empfa 993  
 db2envar.bat 994  
 db2evmon 995  
 db2evtbl 997  
 db2exmig  
 詳細 1003  
 db2expln 1004  
 db2extsec 1014  
 db2flsn 1017  
 db2fm 1022  
 db2fmcu 1025  
 db2fodc 1027  
 db2fopt 1039  
 db2fs  
 詳細 1042  
 db2gcf 1043  
 db2gov  
 詳細 1046  
 db2govlg 1049  
 db2gpmap 1050  
 db2iauto 1051  
 db2iclus 1052  
 db2icrt  
 詳細 1055  
 db2IdentifyType1  
 タイプ 1 索引の識別 1068  
 db2idrop  
 詳細 1065  
 db2ilist 1071  
 db2inidb 1072  
 db2inspf  
 詳細 1075  
 db2iprune 1076  
 db2isetup 1078  
 db2iupdt  
 詳細 1080  
 db2iupgrade  
 詳細 1090  
 db2jdbcbind 1094  
 db2ldcfg 1096  
 db2level  
 詳細 1098  
 db2licm  
 詳細 1099  
 db2listvolumes 1102  
 db2logsforrwd 1103  
 db2look  
 詳細 1104  
 db2ls  
 詳細 1118

コマンド (続き)

db2move 1121  
 db2mqslsn 1132  
 db2mscs 1136  
 db2mtrk 1141  
 db2nchg 1146  
 db2ncrt 1148  
 db2ndrop 1150  
 db2nrcfg 1152  
 db2nrupdt 1153  
 db2nrupgrade 1154  
 db2osconf  
 詳細 1156  
 db2pd  
 詳細 1159  
 db2pdcfg  
 詳細 1261  
 db2perfc 1269  
 db2perfi 1271  
 db2perfr 1272  
 db2prereqcheck 1273  
 db2rbind 1275  
 db2relocatedb 1277  
 db2rfe  
 詳細 1284  
 db2rfpen 1286  
 db2rmicons 1287  
 db2sampl  
 詳細 1289  
 db2schex 1292  
 db2set 1294  
 db2setup  
 詳細 1299  
 db2snapcore 1301  
 db2start 683, 1303  
 db2stat 1304  
 db2stop  
 詳細 1305  
 STOP DATABASE MANAGER 696  
 db2support  
 詳細 1306  
 db2swtch 1326  
 db2sync 1328  
 db2systray 1329  
 db2tapemgr 1332  
 db2tbst 1336  
 db2tdbmgr  
 詳細 1337  
 db2top 1338  
 db2trc  
 詳細 1343  
 db2trcoff 1360  
 db2trcon 1361  
 db2ts ALTER INDEX 1392  
 db2ts CLEANUP FOR TEXT 1397  
 db2ts CLEAR COMMAND LOCKS 1398

コマンド (続き)

db2ts CLEAR EVENTS FOR INDEX 1400  
 db2ts CREATE INDEX 1402  
 db2ts DISABLE DATABASE FOR TEXT 1410  
 db2ts DROP INDEX 1412  
 db2ts ENABLE DATABASE FOR TEXT 1414  
 db2ts HELP 1416  
 db2ts START FOR TEXT 1418  
 db2ts STOP FOR TEXT 1420  
 db2ts UPDATE INDEX 1422  
 db2unins 1363  
 db2untag 1366  
 db2updserv 1367  
 db2val  
   詳細 1368  
 db2xdbmig 1370  
 db2xpvt 1371  
 db2\_deinstall  
   詳細 850  
 db2\_install 854  
 db2\_local\_ps 858  
 DEACTIVATE DATABASE 171  
 DECOMPOSE XML DOCUMENT 173  
 DECOMPOSE XML DOCUMENTS 175  
 DEREGISTER 179  
 DESCRIBE  
   詳細 180  
 DETACH 190  
 disable\_MQFunctions 1372  
 doce\_deinstall  
   詳細 1373  
 doce\_install 1375  
 DROP CONTACT 191  
 DROP CONTACTGROUP 192  
 DROP DATABASE 193  
 DROP DBPARTITIONNUM VERIFY 195  
 DROP TOOLS CATALOG 196  
 ECHO 198  
 EDIT 199  
 enable\_MQFunctions 1377  
 EXPORT 200  
 FORCE APPLICATION 213  
 GET ADMIN CONFIGURATION 215  
 GET ALERT CONFIGURATION 217  
 GET CLI CONFIGURATION 223  
 GET CONNECTION STATE 225  
 GET CONTACTGROUP 226  
 GET CONTACTGROUPS 227  
 GET CONTACTS 228  
 GET DATABASE CONFIGURATION 229  
 GET DATABASE MANAGER CONFIGURATION 242  
 GET DATABASE MANAGER MONITOR SWITCHES 252  
 GET DESCRIPTION FOR HEALTH INDICATOR 255  
 GET HEALTH NOTIFICATION CONTACT LIST 256  
 GET HEALTH SNAPSHOT 257  
 GET INSTANCE 261

コマンド (続き)

GET MONITOR SWITCHES 262  
 GET RECOMMENDATIONS 265  
 GET ROUTINE 269  
 GET SNAPSHOT  
   詳細 271  
 HELP 288  
 HISTORY 289  
 IMPORT 290  
 INGEST 323  
 INITIALIZE TAPE 373  
 INSPECT  
   詳細 374  
 installDSDriver 1380  
 installFixPack  
   詳細 1382  
 LIST ACTIVE DATABASES 382  
 LIST APPLICATIONS 384  
 LIST COMMAND OPTIONS 387  
 LIST DATABASE DIRECTORY 388  
 LIST DATABASE PARTITION GROUPS 392  
 LIST DBPARTITIONNUMS 394  
 LIST DCS APPLICATIONS 395  
 LIST DCS DIRECTORY 397  
 LIST DRDA INDOUBT TRANSACTIONS 399  
 LIST HISTORY 401  
 LIST INDOUBT TRANSACTIONS 406  
 LIST NODE DIRECTORY 409  
 LIST ODBC DATA SOURCES 412  
 LIST PACKAGES/TABLES 413  
 LIST TABLESPACE CONTAINERS 416  
 LIST TABLESPACES 418  
 LIST UTILITIES 423  
 LOAD 425  
 LOAD QUERY 478  
 Microsoft Cluster Server 1052  
 MIGRATE DATABASE 484  
 MQListener 1132  
 PING 485  
 PRECOMPILE 487  
 PRUNE HISTORY/LOGFILE 517  
 PRUNE LOGFILE  
   詳細 517  
 PUT ROUTINE 520  
 QUERY CLIENT 522  
 QUIESCE 523  
 QUIESCE TABLESPACES FOR TABLE 527  
 QUIT 530  
 REBIND 531  
 RECOVER DATABASE 536  
 REDISTRIBUTE DATABASE PARTITION GROUP 544  
 REFRESH LDAP 553  
 REGISTER 555  
 REORG INDEXES/TABLE 562  
 REORGCHK 582  
 RESET ADMIN CONFIGURATION 595

## コマンド (続き)

RESET ALERT CONFIGURATION 597  
RESET DATABASE CONFIGURATION 599  
RESET DATABASE MANAGER CONFIGURATION 601  
RESET MONITOR 603  
RESTART DATABASE 605  
RESTORE DATABASE 608  
REWIND TAPE 633  
ROLLFORWARD DATABASE 634  
RUNCMD 647  
RUNSTATS 648  
SET CLIENT 665  
SET RUNTIME DEGREE 668  
SET SERVEROUTPUT 670  
SET TABLESPACE CONTAINERS 672  
SET TAPE POSITION 675  
SET UTIL\_IMPACT\_PRIORITY 676  
SET WORKLOAD  
    詳細 679  
SET WRITE 680  
setup 1388  
START DATABASE MANAGER 683  
START DB MANAGER 683  
START DBM 683  
START HADR 693  
STOP DATABASE MANAGER  
    詳細 696  
STOP DB MANAGER 696  
STOP DBM 696  
STOP HADR 701  
TAKEOVER HADR 703  
TERMINATE 708  
UNCATALOG DATABASE 709  
UNCATALOG DCS DATABASE 711  
UNCATALOG LDAP DATABASE 712  
UNCATALOG LDAP NODE 713  
UNCATALOG NODE 714  
UNCATALOG ODBC DATA SOURCE 715  
UNQUIESCE 716  
UPDATE ADMIN CONFIGURATION 718  
UPDATE ALERT CONFIGURATION 720  
UPDATE ALTERNATE SERVER FOR DATABASE 725  
UPDATE ALTERNATE SERVER FOR LDAP  
    DATABASE 727  
UPDATE CLI CONFIGURATION 728  
UPDATE COMMAND OPTIONS 730  
UPDATE CONTACT 732  
UPDATE CONTACTGROUP 733  
UPDATE DATABASE CONFIGURATION 734  
UPDATE DATABASE MANAGER CONFIGURATION 738  
UPDATE HEALTH NOTIFICATION CONTACT LIST 740  
UPDATE HISTORY FILE 741  
UPDATE LDAP NODE 744  
UPDATE MONITOR SWITCHES 746  
UPDATE XMLSCHEMA 749

## コマンド (続き)

UPGRADE DATABASE  
    詳細 751  
コマンド行プロセッサ (CLP)  
    オプション 11  
    概要 2  
    行継続文字 2  
    コマンド  
        構文 9  
        シェル・コマンド 9  
        終了 9, 530, 708  
        バックエンド処理 708  
        ヘルプ 9, 22  
        戻りコード 21  
        呼び出しコマンド 9  
        SQL ステートメント 23  
コミットなし (NC) 145  
ご利用条件  
    資料 1442  
コンテナ・タグの解放コマンド 1366

## [サ行]

### 最適化

REORG INDEXES/TABLE コマンド 562

### 索引

#### 統計

RUNSTATS コマンド 648

REORGCHK コマンド 582

サンプル・データベースの作成コマンド 1289

磁気テープ上のログの管理コマンド 1332

システム・カタログ

    分析 913

システム・データベース・ディレクトリー

    アンカタログ 709

    項目の削除 709

    変更、コメントの 143

### 終了

    異常 605

    コマンド行プロセッサのバックエンド処理 708

    通常 696

### 障害モニター

    db2fmcu コマンド 1025

### 資料

    印刷 1434

    概要 1433

    使用に関するご利用条件 1442

    PDF ファイル 1434

### ストレージ

    物理的 562

    すべてのディスク・ボリュームの GUID の表示コマンド 1102

    すべてのパッケージの再バインド・コマンド 1275

### 静止

    表スペース 527

    製品の更新の表示コマンド 1367

設計アドバイザー  
詳細 877  
接続構成インポート・ツール・コマンド 918  
接続構成エクスポート・ツール・コマンド 916  
設定  
CLPPlus での背景色 40  
CLPPlus でのフォントの色 39  
CLPPlus のフォント 38  
宣言生成プログラム・コマンド 967  
ゾーン 10 進ファイル・タイプ修飾子  
LOAD コマンド 425

## [タ行]

タイプ 1 索引識別ツール・コマンド 1068  
対話式 CLI コマンド 937  
大/小文字の区別  
コマンド 2  
命名規則 1426  
チュートリアル  
トラブルシューティング 1442  
問題判別 1442  
リスト 1441  
pureXML 1441  
ツール・カタログ・データベースのマイグレーション・コマンド 1337  
データ  
再分散  
REDISTRIBUTE DATABASE PARTITION GROUP コマンド 544  
整合性  
分離レベル 145  
フラグメント化 562  
データ移動  
区切り文字に関する制約事項 1431  
データベース間の 290  
データベース  
アップグレード  
UPGRADE DATABASE コマンド 751  
アンカタログ  
システム・データベース・ディレクトリー 709  
データベース接続サービス (DCS) 711  
カタログ作成  
CATALOG DATABASE コマンド 122  
間接 (間接) ディレクトリー項目 388  
再作成  
RESTORE DATABASE コマンド 608  
再始動 605  
再編成 582  
削除  
システム・データベース・ディレクトリー 709  
データベース接続サービス (DCS) 711  
DROP DATABASE コマンド 193  
作成  
CREATE DATABASE コマンド 148  
情報収集 271

データベース (続き)  
データの表へのロード 425  
統計 648  
ドロップ  
DROP DATABASE コマンド 193  
バックアップ  
履歴ファイル 517  
表からファイルへのエクスポート  
EXPORT コマンド 200  
ファイルから表へのインポート  
IMPORT コマンド 290  
ホーム (home) ディレクトリー項目 388  
マイグレーション  
MIGRATE DATABASE コマンド 484  
モニター  
db2top コマンド 1338  
RESET MONITOR コマンド 603  
リカバリー  
ROLLFORWARD DATABASE コマンド 634  
リストア 608  
リモート (remote) ディレクトリー項目 388  
ロールフォワード・リカバリー  
ROLLFORWARD DATABASE コマンド 634  
データベース移動ツール・コマンド 1121  
データベース構成ファイル  
値の取り出し 229  
値のリセット 599  
更新 734  
例 229  
データベース接続サービス (DCS) ディレクトリー  
項目の除去 711  
データベースの再配置コマンド 1277  
データベース分析およびレポート・ツール・コマンド  
詳細 957  
データベース・システム・モニター  
GET DATABASE MANAGER MONITOR SWITCHES コマンド 252  
GET MONITOR SWITCHES コマンド 262  
GET SNAPSHOT コマンド 271  
RESET MONITOR コマンド 603  
UPDATE MONITOR SWITCHES コマンド 746  
データベース・ディレクトリー  
サンプル内容 388  
詳細 388  
変更、コメントの 143  
データベース・パーティション・サーバー構成の変更コマンド  
1146  
データベース・パフォーマンス値のリセット・コマンド 1269  
データベース・マネージャー  
開始 683  
停止 696  
統計 271  
モニター・スイッチ  
GET DATABASE MANAGER MONITOR SWITCHES コマンド 252  
GET MONITOR SWITCHES コマンド 262

データベース・マネージャー構成ファイル  
 値の取り出し 242  
 サンプル・ファイル 242  
 テープ・バックアップ  
 BACKUP DATABASE コマンド 89  
 停止  
 DB2 1305  
 ディレクトリー  
 システム・データベース  
 項目の削除 709  
 データベースのアンカatalog 709  
 変更、コメントの 143  
 データベース接続サービス (DCS)  
 項目の削除 711  
 ノード  
 項目の削除 714  
 ローカル・データベース  
 変更、コメントの 143  
 デフォルト DB2 コピーとデータベース・クライアント・インターフェース・コピーの切り替えコマンド 1326  
 統計  
 再編成、索引の 582  
 データベース 648  
 データベース・マネージャー 271  
 REORGCHK コマンド 582  
 特記事項 1445  
 特権  
 データベース  
 データベース作成時に付与 148  
 トラップ・ファイルのフォーマット・コマンド 1371  
 トラブルシューティング  
 オンライン情報 1442  
 チュートリアル 1442  
 ログ・ファイル 929  
 db2pd コマンド 1159  
 トレース  
 アクティブ化 1343, 1360, 1361  
 ファイルへのダンプ 1343  
 command line processor plus (CLPPlus) 64  
 トレースのオフ  
 db2trcoff コマンド 1360  
 トレースのオン  
 db2trcon コマンド 1361  
 トレース・コマンド 1343

## [ナ行]

ノード  
 SOCKS  
 CATALOG TCPIP/TCPIP4/TCPIP6 NODE コマンド 139  
 ノード・ディレクトリー  
 項目の削除 714

## [ハ行]

バインド  
 暗黙的に作成されるスキーマ 100, 487  
 エラー 148  
 BIND コマンド 100  
 バインド変数  
 CLPPlus 54  
 バインド・ファイル  
 記述ツール・コマンド 906  
 パスワード  
 変更  
 ATTACH コマンド 84  
 CONNECT ステートメント 23  
 バックアップ  
 スナップショット・バックアップ・オブジェクト 860  
 バックアップの検査コマンド 924  
 バッケージ  
 再作成 531  
 パフォーマンス  
 索引  
 REORGCHK コマンド 582  
 表  
 再編成 562  
 REORGCHK コマンド 582  
 Windows  
 パフォーマンス・モニター登録ツール・コマンド 1272  
 パフォーマンス・カウンター登録ユーティリティー・コマンド 1271  
 反復可能読み取り (RR)  
 変更 145  
 非 root インスタンスのアップグレード・コマンド 1154  
 非 root インストール  
 インスタンスの更新 1153  
 構成 1152  
 root ベース・フィーチャーの使用可能化 1284  
 非コミット読み取り (UR) 分離レベル  
 変更 145  
 表  
 再編成  
 必要性の判別 582  
 REORG INDEXES/TABLE コマンド 562  
 統計  
 詳細 648  
 ファイルにエクスポートする 200  
 ファイルをインポートする 290  
 ロード 425  
 表スペース状態の獲得コマンド 1336  
 ファースト・ステップ  
 db2fs コマンド 1042  
 ファイル・タイプ修飾子  
 エクスポート・ユーティリティー 200  
 IMPORT コマンド 290  
 LOAD コマンド 425  
 ファイル・フォーマット  
 インポート、ファイルを表へ 290

ファイル・フォーマット (続き)  
ファイルへの表のエクスポート 200  
ファントム静止 527  
複数ページ・ファイル割り振りコマンド 993  
プロキシ・ノード  
Tivoli Storage Manager (TSM)  
db2adutl コマンド 867  
分散マップの取得コマンド 1050  
分離レベル  
CHANGE ISOLATION LEVEL コマンド 145  
ヘルプ  
コマンド 22  
メッセージ 22  
SQL ステートメント 1437  
ベンチマーク・ツール・コマンド 895  
ホスト・システム  
除去、DCS カタログ項目の 711  
データベースのカタログ作成 126  
DB2 Connect がサポートする接続  
CATALOG DCS DATABASE コマンド 126  
本書の構成 ix  
本書の対象読者 ix

## [マ行]

マイグレーション  
ツール・カタログ・データベース 1337  
DB2 Administration Server  
コマンド 846  
Explain 表  
コマンド 1003  
XSR オブジェクト 1370  
未確定トランザクション・フィールド 406  
ミラーリングされたデータベースの初期化コマンド 1072  
命名規則  
データベース・マネージャー・オブジェクト 1426  
メッセージ  
ヘルプ 9, 22  
メモリー・トラッカー・コマンド  
詳細 1141  
戻りコード  
コマンド行プロセッサ (CLP) 21  
モニター  
データベース 252, 262  
db2pd コマンド 1159  
問題判別  
診断ツール  
db2fodc コマンド 1027  
db2snapcore 1301  
db2support 1306  
チュートリアル 1442  
利用できる情報 1442  
問題判別動作の DB2 データベースの構成コマンド 1261

1456 コマンド・リファレンス

## [ヤ行]

読み取り固定 (RS)  
変更 145

## [ラ行]

ライセンス  
ライセンス管理ツール・コマンド 1099  
リカバリー  
データベース  
RESTORE DATABASE コマンド 608  
ロールフォワードなし 608  
ロールフォワードによる 634  
リストア  
旧バージョンの DB2 データベース 608  
ローカル・データベース・ディレクトリー  
変更、コメントの 143  
ロード・ユーティリティー  
一時ファイル  
スペース所要量 425  
ファイルをデータベース表へ 425  
ファイル・タイプ修飾子 425  
ロールフォワード・ベンディング状態リセット・コマンド  
1286  
ロールフォワード・リカバリーに必要なログのリスト・コマン  
ド 1103  
ログ  
ロールフォワード中のリスト 634  
ログ・シーケンス番号の検出コマンド 1017  
ログ・レコード  
command line processor plus (CLPPlus) 64  
ロック  
リセット、最大値をデフォルトに 599

## [ワ行]

ワークステーション  
リモート  
データベースのアンカATALOG 709  
データベースのカタログ作成 122  
ノードのアンカATALOG 714  
**A**  
ACCEPT コマンド 759  
ACTIVATE DATABASE コマンド  
詳細 72  
Active Directory  
スキーマ拡張コマンド 1292  
ADD CONTACT コマンド  
ADMIN\_CMD プロシージャーを使用しない 74  
ADD CONTACTGROUP コマンド  
ADMIN\_CMD プロシージャーを使用しない 75  
ADD DBPARTITIONNUM コマンド 76

ADD XMLSCHEMA DOCUMENT コマンド 79  
ALTER INDEX Text Search コマンド 1392  
anyorder ファイル・タイプ修飾子 425  
APPEND コマンド 762  
ARCHIVE LOG コマンド 81  
ASC インポート・ファイル・タイプ 290  
ATTACH コマンド  
    詳細 84  
AUTOCONFIGURE コマンド  
    ADMIN\_CMD プロシージャを使用しない 86

## B

BACKUP DATABASE コマンド  
    ADMIN\_CMD プロシージャを使用しない 89  
binarynumerics ファイル・タイプ修飾子  
    LOAD コマンド 425  
BIND コマンド  
    詳細 100  
BREAK コマンド 763  
BTITLE コマンド 764

## C

CALL ステートメント  
    CLP 23  
CATALOG DATABASE コマンド  
    詳細 122  
CATALOG DCS DATABASE コマンド 126  
CATALOG LDAP DATABASE コマンド 129  
CATALOG LDAP NODE コマンド 132  
CATALOG LOCAL NODE コマンド 133  
CATALOG NAMED PIPE NODE コマンド 135  
CATALOG ODBC DATA SOURCE コマンド 137  
CATALOG TCPIP/TCPIP4/TCPIP6 NODE コマンド 139  
CHANGE DATABASE COMMENT コマンド 143  
CHANGE ISOLATION LEVEL コマンド 145  
CHANGE コマンド 766  
chardel ファイル・タイプ修飾子  
    EXPORT コマンド 200  
    IMPORT コマンド 290  
    LOAD コマンド 425  
CLEANUP FOR TEXT Text Search コマンド 1397  
CLEAR COMMAND LOCKS Text Search コマンド 1398  
CLEAR EVENTS FOR INDEX Text Search コマンド 1400  
CLEAR コマンド 768  
CLI  
    構成  
        検索 223  
CLI/ODBC 静的パッケージ・バインディング・ツール・コマンド 911  
CLOSE ステートメント  
    CLP で実行 23  
CLP での終了コード 21  
CLP の True Type フォント要件 2

coldel ファイル・タイプ修飾子  
    EXPORT コマンド 200  
    IMPORT コマンド 290  
    LOAD コマンド 425  
COLUMN コマンド 770  
command line processor plus (CLPPlus)  
    エスケープ文字 52  
    開始 32  
    概要 31  
    コマンド  
        サマリー 753  
        実行のスキップ 50  
        実行の割り込み 50  
    ACCEPT 759  
    APPEND 762  
    BREAK 763  
    BTITLE 764  
    CHANGE 766  
    CLEAR 768  
    CLPPLUS 34  
    COLUMN 770  
    COMPUTE 775  
    CONNECT 777  
    COPY 780  
    DB2 コマンド 59  
    DEFINE 783  
    DEL 784  
    DESCRIBE 786  
    DISCONNECT 791  
    EDIT 792  
    EXECUTE 793  
    EXIT 796  
    EXPLAIN PLAN 798  
    GET 799  
    HELP 800  
    HOST 801  
    IMPORT 802  
    INPUT 804  
    LIST 805  
    PAUSE 807  
    PRINT 808  
    PROMPT 809  
    QUIT 810  
    REMARK 811  
    REORGCHK 812  
    REPFOOTER 814  
    REPHEADER 816  
    RUN 818  
    SAVE 819  
    SET 820  
    SHOW 831  
    SPOOL 830  
    START 832  
    TTITLE 833  
    UNDEFINE 835  
    WHENEVER OSERROR 836

command line processor plus (CLPPlus) (続き)

コマンド (続き)

WHENEVER SQLERROR 838

! 755

. 754

/ 756

@ 757

@@ 758

コメント 51

コンソール・タイプ 37, 38

スクリプト 49

制約事項 62

トラブルシューティング 63

トレース 64

背景色の設定 40

バインド変数 54

フォントの色の設定 39

フォントの設定 38

変数 58

ログ・レコード 64

CREATE DATABASE コマンド 61

DSN 別名 40

IBM Data Server Client におけるインストール 31

Kerberos 42

LDAP サポート 45

SERVER\_ENCRYPT 認証 43

UTF-8 サポート 38

COMPLETE XMLSCHEMA コマンド 147

compound ファイル・タイプ修飾子

IMPORT コマンド 290

COMPUTE コマンド 775

CONNECT コマンド 777

CONNECT ステートメント

CLP からの実行 23

COPY コマンド 780

CREATE DATABASE コマンド

詳細 148

CREATE INDEX Text Search コマンド 1402

CREATE TOOLS CATALOG コマンド 168

## D

DAS の自動始動コマンド 842

dasauto コマンド 842

dasct コマンド 843

dasdrop コマンド 845

dasmigr コマンド

詳細 846

dasupdt コマンド 848

dateformat ファイル・タイプ修飾子

IMPORT コマンド 290

LOAD コマンド 425

DB2 Administration Server (DAS)

管理 865

構成 215

作成 843

DB2 Administration Server (DAS) (続き)

ドロップ 865

DB2 Administration Server の除去コマンド 845

DB2 Administration Server のマイグレーション・コマンド  
846

DB2 Connect

サポートされる接続 126

DB2 HA スクリプトのインストールまたは更新コマンド 955

DB2 Text Search

コマンド

ALTER INDEX 1392

CLEANUP FOR TEXT 1397

CLEAR COMMAND LOCKS 1398

CLEAR EVENTS FOR TEXT 1400

CREATE INDEX 1402

DISABLE DATABASE FOR TEXT 1410

DROP INDEX 1412

ENABLE DATABASE FOR TEXT 1414

HELP 1416

START FOR TEXT 1418

STOP FOR TEXT 1420

UPDATE INDEX 1422

ALTER INDEX コマンド 1392

CLEAR COMMAND LOCKS コマンド 1398

CLEAR EVENTS FOR TEXT コマンド 1400

CREATE INDEX コマンド 1402

DISABLE DATABASE FOR TEXT コマンド 1410

DROP INDEX コマンド 1412

ENABLE DATABASE FOR TEXT コマンド 1414

HELP コマンド 1416

START FOR TEXT コマンド 1418

STOP FOR TEXT コマンド 1420

UPDATE INDEX コマンド 1422

DB2 インスタンスの制御コマンド 1043

DB2 インフォメーション・センター

更新 1437, 1439

バージョン 1437

DB2 インフォメーション・センターのアンインストール・コマ  
ンド 1373

DB2 インフォメーション・センターのインストール・コマンド  
1375

DB2 オブジェクトの権限の設定コマンド 1014

DB2 管理プログラム

開始 1046

停止 1046

ログ・ファイルの照会 1049

DB2 コピー

妥当性検査ツール 1368

db2 コマンド 9

DB2 コマンド・ウィンドウのオープン・コマンド 954

DB2 サーバー

管理

DAS 構成ファイルから値を検索 215

DAS 構成ファイルの値のリセット 595

DAS 構成ファイルの項目の更新 718

DB2 サービス・レベルの表示コマンド 1098



DB2 システム・トレイの開始コマンド 1329  
 DB2 障害モニター・コマンド 1022  
 DB2 シンクロナイザーの開始コマンド 1328  
 DB2 スナップショット・バックアップ・オブジェクトの管理コマンド 860  
 DB2 製品、フィーチャー、または言語のアンインストール・コマンド 850  
 DB2 製品のアンインストール・コマンド 1363  
 DB2 ツールのメインメニュー項目の作成コマンド 864  
 DB2 ツールのメインメニュー項目の除去コマンド 1287  
 DB2 データベース・ドライブのマップ・コマンド 991  
 DB2 統計および DDL 抽出ツール・コマンド 1104  
 DB2 のインストール・コマンド  
   db2iprune コマンド 1076  
   db2setup コマンド 1299  
   db2\_install 854  
   setup コマンド 1388  
 DB2 の開始コマンド 1303  
 DB2 の停止コマンド 1305  
 DB2 プロファイル・レジストリー・コマンド 1294  
 DB2 ワークロード管理  
   SET WORKLOAD コマンド 679  
 db2acsutil コマンド 860  
 db2addicons コマンド 864  
 db2admin コマンド 865  
 db2adutl コマンド  
   詳細 867  
 db2advis コマンド  
   詳細 877  
 db2audit コマンド 884  
 db2batch コマンド  
   詳細 895  
 db2bfd コマンド  
   詳細 906  
 db2caem コマンド 907  
 db2cap コマンド 911  
 db2cat コマンド  
   詳細 913  
 db2cfexp コマンド 916  
 db2cfimp コマンド 918  
 db2chglbpath コマンド 919  
 db2chgpath コマンド 922  
 db2ckbkb コマンド 924  
 db2cklog コマンド  
   詳細 929  
 db2ckrst コマンド 932  
 db2ckupgrade コマンド  
   詳細 934  
 db2cli コマンド 937  
 db2cmd コマンド 954  
 db2cptsa コマンド 955  
 db2dart コマンド  
   詳細 957  
 db2daslevel コマンド 966  
 db2dclgn コマンド  
   詳細 967  
 db2diag コマンド  
   詳細 970  
 db2drdat コマンド  
   詳細 989  
 db2drvmp コマンド 991  
 db2empfa コマンド 993  
 db2envar.bat コマンド  
   詳細 994  
 db2evmon コマンド  
   詳細 995  
 db2evtbl コマンド  
   詳細 997  
 db2exfmt コマンド  
   詳細 999  
 db2exmig コマンド  
   詳細 1003  
 db2expln コマンド  
   詳細 1004  
 db2extsec コマンド 1014  
 db2flsn コマンド 1017  
 db2fm コマンド  
   詳細 1022  
 db2fmcu コマンド 1025  
 db2fodc コマンド  
   詳細 1027  
 db2fopt コマンド 1039  
 db2fs コマンド 1042  
 db2gcf コマンド 1043  
 db2gov コマンド  
   詳細 1046  
 db2govlg コマンド 1049  
 db2gpmap コマンド 1050  
 db2iauto コマンド 1051  
 db2iclus コマンド 1052  
 db2icrt コマンド  
   詳細 1055  
 db2IdentifyType1 コマンド  
   詳細 1068  
 db2idrop コマンド  
   詳細 1065  
 db2ilist コマンド 1071  
 db2inidb コマンド  
   詳細 1072  
 db2inspf コマンド  
   詳細 1075  
 db2iprune コマンド  
   詳細 1076  
 db2isetup コマンド  
   詳細 1078  
 db2iupdt コマンド  
   詳細 1080  
 db2iupgrade コマンド  
   詳細 1090  
 db2jdbcbind コマンド 1094  
 db2ldcfg コマンド  
   詳細 1096

db2level コマンド  
     詳細 1098  
 db2licm コマンド  
     詳細 1099  
 db2listvolumes コマンド 1102  
 db2logsforrwd コマンド 1103  
 db2look コマンド  
     詳細 1104  
 db2ls コマンド  
     詳細 1118  
 db2move コマンド  
     詳細 1121  
 db2mqdsn コマンド  
     詳細 1132  
 db2mscs コマンド 1136  
 db2mtrk コマンド  
     詳細 1141  
 db2nchg コマンド  
     詳細 1146  
 db2ncrt コマンド  
     詳細 1148  
 db2ndrop コマンド  
     詳細 1150  
 db2nrcfg コマンド 1152  
 db2nrupdt コマンド 1153  
 db2nrupgrade コマンド 1154  
 DB2OPTIONS 環境変数  
     CLP オプションの設定 11  
 db2osconf コマンド  
     詳細 1156  
 db2pd コマンド  
     詳細 1159  
 db2pdcfg コマンド  
     詳細 1261  
 db2perfc コマンド  
     詳細 1269  
 db2perfi コマンド  
     詳細 1271  
 db2perfr コマンド  
     詳細 1272  
 db2prereqcheck コマンド  
     詳細 1273  
 db2rbind コマンド  
     詳細 1275  
 db2relocatedb コマンド  
     詳細 1277  
 db2rfe コマンド  
     詳細 1284  
 db2rfpen コマンド 1286  
 db2rmicons コマンド 1287  
 db2rspgn コマンド 1288  
 db2sampl コマンド 1289  
 db2schex コマンド 1292  
 db2set コマンド  
     詳細 1294  
 db2setup コマンド  
     詳細 1299  
 db2snapcore コマンド 1301  
 db2start コマンド  
     概要 1303  
     詳細 683  
 db2stat コマンド 1304  
 db2stop コマンド  
     概要 1305  
     詳細 696  
 db2support コマンド  
     詳細 1306  
 db2swtch コマンド 1326  
 db2sync コマンド 1328  
 db2systray コマンド 1329  
 db2tapemgr コマンド  
     詳細 1332  
 db2tbst コマンド 1336  
 db2tdbmgr コマンド  
     詳細 1337  
 db2top コマンド  
     詳細 1338  
 db2trc コマンド  
     詳細 1343  
 db2trcoff コマンド 1360  
 db2trcon コマンド 1361  
 db2ts コマンド  
     ALTER INDEX 1392  
     CLEANUP FOR TEXT 1397  
     CLEAR COMMAND LOCKS 1398  
     CLEAR EVENTS FOR INDEX 1400  
     CREATE INDEX 1402  
     DISABLE DATABASE FOR TEXT 1410  
     DROP INDEX 1412  
     ENABLE DATABASE FOR TEXT 1414  
     HELP 1416  
     START FOR TEXT 1418  
     STOP FOR TEXT 1420  
     UPDATE INDEX 1422  
 db2unins コマンド 1363  
 db2untag コマンド 1366  
 db2updsvr コマンド 1367  
 db2val コマンド  
     詳細 1368  
 db2xdbmig コマンド 1370  
 db2xpri コマンド 1371  
 db2\_deinstall コマンド  
     詳細 850  
 db2\_install コマンド  
     詳細 854  
 db2\_local\_ps コマンド 858  
 DCLGEN コマンド 967  
 DEACTIVATE DATABASE コマンド  
     詳細 171  
 DECLARE CURSOR ステートメント  
     CLP で実行 23

DECOMPOSE XML DOCUMENT コマンド 173  
 DECOMPOSE XML DOCUMENTS コマンド  
   詳細 175  
 decplusblank ファイル・タイプ修飾子  
   EXPORT コマンド 200  
   IMPORT コマンド 290  
   LOAD コマンド 425  
 decpt ファイル・タイプ修飾子  
   EXPORT コマンド 200  
   IMPORT コマンド 290  
   LOAD コマンド 425  
 DEFINE コマンド 783  
 DEL コマンド 784  
 delprioritychar ファイル・タイプ修飾子  
   IMPORT コマンド 290  
   LOAD コマンド 425  
 DEREGISTER コマンド 179  
 DESCRIBE CLPPlus コマンド 786  
 DESCRIBE コマンド  
   詳細 180  
 DETACH コマンド  
   詳細 190  
 DISABLE DATABASE FOR TEXT Text Search コマンド  
   1410  
 disable\_MQFunctions コマンド 1372  
 DISCONNECT コマンド 791  
 doce\_deinstall コマンド  
   詳細 1373  
 doce\_install コマンド  
   詳細 1375  
 DRDA トレース・コマンド 989  
 DROP CONTACT コマンド  
   詳細  
     ADMIN\_CMD を使用しない 191  
 DROP CONTACTGROUP コマンド  
   詳細  
     ADMIN\_CMD を使用しない 192  
 DROP DATABASE コマンド 193  
 DROP DBPARTITIONNUM VERIFY コマンド 195  
 DROP INDEX Text Search コマンド 1412  
 DROP TOOLS CATALOG コマンド 196  
 dumpfile ファイル・タイプ修飾子 425

## E

ECHO コマンド 198  
 EDIT CLPPlus コマンド 199  
 EDIT コマンド 792  
 ENABLE DATABASE FOR TEXT Text Search コマンド 1414  
 enable\_MQFunctions コマンド 1377  
 EXECUTE コマンド 793  
 EXIT コマンド 796  
 Explain 表  
   内容のフォーマット 999  
 Explain 表のマイグレーション・コマンド 1003

EXPORT コマンド  
   詳細  
     ADMIN\_CMD を使用しない 200

## F

fastparse ファイル・タイプ修飾子  
   LOAD コマンド 425  
 FETCH ステートメント  
   CLP で実行 23  
 FODC  
   db2fodc コマンド 1027  
 FORCE APPLICATION コマンド  
   ADMIN\_CMD を使用しない 213  
 forcein ファイル・タイプ修飾子  
   IMPORT コマンド 290  
   LOAD コマンド 425

## G

generatedignore ファイル・タイプ修飾子  
   IMPORT コマンド 290  
   LOAD コマンド 425  
 generatedmissing ファイル・タイプ修飾子  
   IMPORT コマンド 290  
   LOAD コマンド 425  
 generatedoverride ファイル・タイプ修飾子  
   LOAD コマンド 425  
 GET ADMIN CONFIGURATION コマンド 215  
 GET ALERT CONFIGURATION コマンド 217  
 GET CLI CONFIGURATION コマンド 223  
 GET CONNECTION STATE コマンド 225  
 GET CONTACTGROUP コマンド 226  
 GET CONTACTGROUPS コマンド 227  
 GET CONTACTS コマンド 228  
 GET DATABASE CONFIGURATION コマンド 229  
 GET DATABASE MANAGER CONFIGURATION コマンド  
   242  
 GET DATABASE MANAGER MONITOR SWITCHES コマンド  
   252  
 GET DESCRIPTION FOR HEALTH INDICATOR コマンド  
   255  
 GET HEALTH NOTIFICATION CONTACT LIST コマンド  
   256  
 GET HEALTH SNAPSHOT コマンド 257  
 GET INSTANCE コマンド 261  
 GET MONITOR SWITCHES コマンド 262  
 GET RECOMMENDATIONS コマンド 265  
 GET ROUTINE コマンド 269  
 GET SNAPSHOT コマンド  
   詳細 271  
     UPDATE MONITOR SWITCHES コマンドへの影響 746  
 GET コマンド 799

## H

HELP コマンド  
    CLPPlus 800  
    DB2 for Linux, UNIX, and Windows 288  
    Text Search 1416  
HISTORY コマンド 289  
HOST コマンド 801

## I

IBM Data Server Client  
    インストール  
        CLPPlus 31  
IBM Data Server Driver コンポーネントの抽出コマンド 1380  
identityignore ファイル・タイプ修飾子  
    IMPORT コマンド 290  
    LOAD コマンド 425  
identitymissing ファイル・タイプ修飾子  
    IMPORT コマンド 290  
    LOAD コマンド 425  
identityoverride ファイル・タイプ修飾子  
    LOAD コマンド 425  
implieddecimal ファイル・タイプ修飾子  
    IMPORT コマンド 290  
    LOAD コマンド 425  
IMPORT コマンド  
    詳細  
        ADMIN\_CMD プロシージャーを使用しない 290  
indexfreespace ファイル・タイプ修飾子  
    LOAD コマンド 425  
indexixf ファイル・タイプ修飾子  
    IMPORT コマンド 290  
indexschema ファイル・タイプ修飾子  
    IMPORT コマンド 290  
INGEST コマンド  
    詳細 323  
INITIALIZE TAPE コマンド  
    ADMIN\_CMD プロシージャーを使用しない 373  
INPUT コマンド 804  
INSPECT コマンド  
    詳細 374  
installDSDriver コマンド  
    詳細 1380  
installFixPack コマンド  
    詳細 1382  
IPX/SPX ノード  
    アンカタログ 714

## J

JDBC  
    パッケージ・バインド・プログラム・ユーティリティー・コ  
    マンド 1094

## K

keepblanks ファイル・タイプ修飾子  
    IMPORT コマンド 290  
    LOAD コマンド 425

## L

LDAP 環境の構成コマンド 1096  
Linux/UNIX の DB2 プロセス状況コマンド 858  
LIST ACTIVE DATABASES コマンド 382  
LIST APPLICATIONS コマンド 384  
LIST COMMAND OPTIONS コマンド 387  
LIST DATABASE DIRECTORY コマンド 388  
LIST DATABASE PARTITION GROUPS コマンド  
    詳細 392  
LIST DBPARTITIONNUMS コマンド 394  
LIST DCS APPLICATIONS コマンド  
    詳細 395  
LIST DCS DIRECTORY コマンド 397  
LIST DRDA INDOUBT TRANSACTIONS コマンド  
    詳細 399  
LIST HISTORY コマンド 401  
LIST INDOUBT TRANSACTIONS コマンド  
    詳細 406  
LIST NODE DIRECTORY コマンド 409  
LIST ODBC DATA SOURCES コマンド 412  
LIST PACKAGES コマンド  
    詳細 413  
LIST TABLES コマンド  
    詳細 413  
LIST TABLESPACE CONTAINERS コマンド  
    詳細 416  
LIST TABLESPACES コマンド  
    詳細 418  
LIST UTILITIES コマンド  
    詳細 423  
LOAD QUERY コマンド  
    詳細 478  
LOAD コマンド  
    詳細  
        ADMIN\_CMD プロシージャーを使用しない 425  
lobsinfile ファイル・タイプ修飾子  
    EXPORT コマンド 200  
    IMPORT コマンド 290  
    LOAD コマンド 425

## M

Microsoft Cluster Server (MSCS)  
    コマンド 1052  
MIGRATE DATABASE コマンド  
    詳細 484  
MQListener コマンド 1132

## N

nochecklengths ファイル・タイプ修飾子  
  IMPORT コマンド 290  
  LOAD コマンド 425  
nodefaults ファイル・タイプ修飾子  
  IMPORT コマンド 290  
nodoubledel ファイル・タイプ修飾子  
  EXPORT コマンド 200  
  IMPORT コマンド 290  
  LOAD コマンド 425  
noeofchar ファイル・タイプ修飾子  
  IMPORT コマンド 290  
  LOAD コマンド 425  
noheader ファイル・タイプ修飾子  
  LOAD コマンド 425  
norowwarnings ファイル・タイプ修飾子  
  LOAD コマンド 425  
notypeid ファイル・タイプ修飾子  
  IMPORT コマンド 290  
NULL  
  ストリング 2  
  SQL 値  
    コマンド行プロセッサの表記 2  
nullindchar ファイル・タイプ修飾子  
  IMPORT コマンド 290  
  LOAD コマンド 425

## O

off トレース・オプション 1360  
on トレース・オプション 1361  
OPEN ステートメント  
  CLP で実行 23

## P

packeddecimal ファイル・タイプ修飾子 425  
pagefreespace ファイル・タイプ修飾子 425  
PAUSE コマンド 807  
PING コマンド 485  
PRECOMPILE コマンド  
  詳細 487  
PREP コマンド 487  
PRINT コマンド 808  
PROMPT コマンド 809  
PRUNE HISTORY/LOGFILE コマンド  
  ADMIN\_CMD を使用しない 517  
PUT ROUTINE コマンド 520

## Q

QUERY CLIENT コマンド 522  
QUIESCE TABLESPACES FOR TABLE コマンド  
  ADMIN\_CMD を使用しない 527

QUIESCE コマンド 523  
QUIT CLPPPlus コマンド 810  
QUIT コマンド 530

## R

REBIND コマンド  
  詳細 531  
reclen ファイル・タイプ修飾子  
  IMPORT コマンド 290  
  LOAD コマンド 425  
RECOVER DATABASE コマンド  
  詳細 536  
REDISTRIBUTE DATABASE PARTITION GROUP コマンド  
  ADMIN\_CMD を使用しない 544  
REFRESH LDAP コマンド 553  
REGISTER XMLSCHEMA コマンド 558  
REGISTER XSROBJECT コマンド 560  
REGISTER コマンド 555  
REMARK コマンド 811  
REORG INDEXES コマンド  
  ADMIN\_CMD を使用しない 562  
REORG TABLE コマンド  
  ADMIN\_CMD を使用しない 562  
REORGCHK コマンド 582  
REPFOOTER コマンド  
  説明 814  
REPHEADER コマンド  
  説明 816  
RESET ADMIN CONFIGURATION コマンド 595  
RESET ALERT CONFIGURATION コマンド  
  ADMIN\_CMD を使用しない 597  
RESET DATABASE CONFIGURATION コマンド  
  ADMIN\_CMD を使用しない 599  
RESET DATABASE MANAGER CONFIGURATION コマンド  
  ADMIN\_CMD を使用しない 601  
RESET MONITOR コマンド 603  
RESTART DATABASE コマンド  
  詳細 605  
RESTORE DATABASE コマンド  
  詳細 608  
REWIND TAPE コマンド  
  ADMIN\_CMD を使用しない 633  
ROLLFORWARD DATABASE コマンド  
  詳細 634  
RUN コマンド 818  
RUNCMD コマンド 647  
RUNSTATS コマンド  
  詳細  
  ADMIN\_CMD を使用しない 648

## S

SAVE コマンド 819

SELECT ステートメント  
     CLP で実行 23  
     EXPORT コマンド 200  
 SET CLIENT コマンド 665  
 SET RUNTIME DEGREE コマンド 668  
 SET SERVEROUTPUT コマンド 670  
 SET TABLESPACE CONTAINERS コマンド 672  
 SET TAPE POSITION コマンド  
     ADMIN\_CMD を使用しない 675  
 SET UTIL\_IMPACT\_PRIORITY コマンド 676  
 SET WORKLOAD コマンド  
     詳細 679  
 SET WRITE コマンド  
     詳細 680  
 SET コマンド 820  
 setup コマンド 1388  
 SHOW コマンド 831  
 SIGALRM 信号  
     開始、データベース・マネージャーの 683  
 SIGINT 信号  
     開始、データベース・マネージャーの 683  
 snapcore コマンド 1301  
 SOCKS  
     ノード  
         パラメーター 139  
 SPOOL コマンド 830  
 SQL および XQuery Explain コマンド 1004  
 SQL ステートメント  
     コマンド行 23  
     ヘルプ  
         アクセス 9  
         表示 1437  
 START CLPPlus コマンド 832  
 START DATABASE MANAGER コマンド  
     詳細 683  
 START DB MANAGER コマンド 683  
 START DBM コマンド 683  
 START FOR TEXT Text Search コマンド 1418  
 START HADR コマンド  
     詳細 693  
 STOP DATABASE MANAGER コマンド  
     詳細 696  
 STOP DB MANAGER コマンド 696  
 STOP DBM コマンド 696  
 STOP FOR TEXT Text Search コマンド 1420  
 STOP HADR コマンド  
     詳細 701  
 striptblanks ファイル・タイプ修飾子  
     IMPORT コマンド 290  
     LOAD コマンド 425  
 striptnulls ファイル・タイプ修飾子  
     IMPORT コマンド 290  
     LOAD コマンド 425  
 subtableconvert ファイル・タイプ修飾子 425

## T

TAKEOVER HADR コマンド  
     詳細 703  
 TCP/IP  
     ノードのアンカタログ 714  
 TERMINATE コマンド 708  
 timeformat ファイル・タイプ修飾子  
     IMPORT コマンド 290  
     LOAD コマンド 425  
 timestampformat ファイル・タイプ修飾子  
     IMPORT コマンド 290  
     LOAD コマンド 425  
 Tivoli Storage Manager (TSM)  
     アーカイブ・イメージ 867  
 totalfreespace ファイル・タイプ修飾子  
     LOAD コマンド 425  
 TTITLE コマンド  
     説明 833

## U

UNCATALOG DATABASE コマンド 709  
 UNCATALOG DCS DATABASE コマンド 711  
 UNCATALOG LDAP DATABASE コマンド 712  
 UNCATALOG LDAP NODE コマンド 713  
 UNCATALOG NODE コマンド 714  
 UNCATALOG ODBC DATA SOURCE コマンド 715  
 UNDEFINE コマンド 835  
 UNQUIESCE コマンド 716  
 UPDATE ADMIN CONFIGURATION コマンド 718  
 UPDATE ALERT CONFIGURATION コマンド  
     ADMIN\_CMD を使用しない 720  
 UPDATE ALTERNATE SERVER FOR DATABASE コマンド  
     725  
 UPDATE ALTERNATE SERVER FOR LDAP DATABASE コ  
     マンド 727  
 UPDATE CLI CONFIGURATION コマンド 728  
 UPDATE COMMAND OPTIONS コマンド 730  
 UPDATE CONTACT コマンド  
     ADMIN\_CMD を使用しない 732  
 UPDATE CONTACTGROUP コマンド  
     ADMIN\_CMD を使用しない 733  
 UPDATE DATABASE CONFIGURATION コマンド  
     ADMIN\_CMD を使用しない 734  
 UPDATE DATABASE MANAGER CONFIGURATION コマン  
     ド  
         ADMIN\_CMD を使用しない 738  
 UPDATE HEALTH NOTIFICATION CONTACT LIST コマンド  
     ADMIN\_CMD を使用しない 740  
 UPDATE HISTORY FILE コマンド 741  
 UPDATE INDEX Text Search コマンド 1422  
 UPDATE LDAP NODE コマンド 744  
 UPDATE MONITOR SWITCHES コマンド 746  
 UPDATE XMLSCHEMA コマンド 749

UPGRADE DATABASE コマンド

詳細 751

usedefaults ファイル・タイプ修飾子

IMPORT コマンド 290

LOAD コマンド 425

## W

WHENEVER OSERROR コマンド 836

WHENEVER SQLERROR コマンド 838

Windows の DB2 プロセス状況コマンド 1304

Windows フェイルオーバー・ユーティリティーのセットアップ・コマンド 1136

## X

XML スキーマ・リポジトリ (XSR)

ADD XMLSCHEMA DOCUMENT コマンド 79

COMPLETE XMLSCHEMA コマンド 147

REGISTER XMLSCHEMA コマンド 558

REGISTER XSROBJECT コマンド 560

UPDATE XMLSCHEMA コマンド 749

XSR オブジェクトのマイグレーション・コマンド 1370

X/Open バックアップ・サービス API (XBSA)

BACKUP DATABASE コマンド・オプション 89

## [特殊文字]

. コマンド 754

/ コマンド 756

@ コマンド 757

@@ コマンド 758









Printed in Japan

SA88-4673-01



日本アイ・ビー・エム株式会社  
〒103-8510 東京都中央区日本橋箱崎町19-21

Spine information:

IBM DB2 10.1 for Linux, UNIX, and Windows

コマンド・リファレンス

