



コマンド・リファレンス
最終更新: 2009 年 4 月



コマンド・リファレンス
最終更新: 2009 年 4 月

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、953 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、www.ibm.com/shop/publications/order にある IBM Publications Center 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、www.ibm.com/planetwide にある IBM Directory of Worldwide Contacts 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： SC23-5846-02
DB2 Version 9.5
for Linux, UNIX, and Windows
Command Reference
Updated April, 2009

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 2009.3

© Copyright International Business Machines Corporation 1993, 2009.

目次

本書について	vii
本書の対象読者	vii
本書の構成	vii
強調表示規則	viii

構文図の見方	ix
------------------	----

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	1
db2 - コマンド行プロセッサの呼び出し	7
コマンド行プロセッサのオプション	8
コマンド行プロセッサの戻りコード	17
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	18
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	18

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 . 19

第 3 章 コマンド構文ヘルプの読み方 27

第 4 章 CLP コマンド 31

ACTIVATE DATABASE	31
ADD CONTACT	32
ADD CONTACTGROUP	33
ADD DBPARTITIONNUM	34
ADD XMLSCHEMA DOCUMENT	37
ARCHIVE LOG	38
ATTACH	40
AUTOCONFIGURE	42
BACKUP DATABASE	45
BIND	55
CATALOG DATABASE	74
CATALOG DCS DATABASE	78
CATALOG LDAP DATABASE	80
CATALOG LDAP NODE	83
CATALOG LOCAL NODE	84
CATALOG NAMED PIPE NODE	85
CATALOG ODBC DATA SOURCE	87
CATALOG TCP/IP/TCP/IP4/TCP/IP6 NODE	88
CHANGE DATABASE COMMENT	92
CHANGE ISOLATION LEVEL	93
COMPLETE XMLSCHEMA	95
CREATE DATABASE	96
CREATE TOOLS CATALOG	117
DEACTIVATE DATABASE	119
DECOMPOSE XML DOCUMENT	120
DEREGISTER	122
DESCRIBE	123

DETACH	128
DROP CONTACT	128
DROP CONTACTGROUP	129
DROP DATABASE	129
DROP DBPARTITIONNUM VERIFY	131
DROP TOOLS CATALOG	132
ECHO	133
EDIT	134
EXPORT	135
FORCE APPLICATION	147
GET ADMIN CONFIGURATION	149
GET ALERT CONFIGURATION	150
GET AUTHORIZATIONS	156
GET CLI CONFIGURATION	157
GET CONNECTION STATE	159
GET CONTACTGROUP	160
GET CONTACTGROUPS	160
GET CONTACTS	161
GET DATABASE CONFIGURATION	162
GET DATABASE MANAGER CONFIGURATION	167
GET DATABASE MANAGER MONITOR SWITCHES	172
GET DESCRIPTION FOR HEALTH INDICATOR	175
GET HEALTH NOTIFICATION CONTACT LIST	176
GET HEALTH SNAPSHOT	177
GET INSTANCE	179
GET MONITOR SWITCHES	180
GET RECOMMENDATIONS FOR HEALTH INDICATOR	182
GET ROUTINE	185
GET SNAPSHOT	186
HELP	202
HISTORY	203
IMPORT	204
INITIALIZE TAPE	235
INSPECT	236
LIST ACTIVE DATABASES	244
LIST APPLICATIONS	245
LIST COMMAND OPTIONS	247
LIST DATABASE DIRECTORY	248
LIST DATABASE PARTITION GROUPS	251
LIST DBPARTITIONNUMS	253
LIST DCS APPLICATIONS	254
LIST DCS DIRECTORY	256
LIST DRDA INDOUBT TRANSACTIONS	257
LIST HISTORY	258
LIST INDOUBT TRANSACTIONS	261
LIST NODE DIRECTORY	263
LIST ODBC DATA SOURCES	266
LIST PACKAGES/TABLES	266
LIST TABLESPACE CONTAINERS	269

LIST TABLESPACES	270
LIST UTILITIES	286
LOAD	288
LOAD QUERY	334
MIGRATE DATABASE	340
PING	341
PRECOMPILE	343
PRUNE HISTORY/LOGFILE	371
PUT ROUTINE	373
QUERY CLIENT	374
QUIESCE	375
QUIESCE TABLESPACES FOR TABLE	378
QUIT	381
REBIND	381
RECOVER DATABASE	385
REDISTRIBUTE DATABASE PARTITION GROUP	391
REFRESH LDAP	402
REGISTER	403
REGISTER XMLSCHEMA	406
REGISTER XSROBJECT	408
REORG INDEXES/TABLE	410
REORGCHK	421
RESET ADMIN CONFIGURATION	432
RESET ALERT CONFIGURATION	433
RESET DATABASE CONFIGURATION	434
RESET DATABASE MANAGER CONFIGURATION	436
RESET MONITOR	438
RESTART DATABASE	439
RESTORE DATABASE	441
REWIND TAPE	461
ROLLFORWARD DATABASE	462
RUNCMD	473
RUNSTATS	474
SET CLIENT	489
SET RUNTIME DEGREE	492
SET TABLESPACE CONTAINERS	493
SET TAPE POSITION	495
SET UTIL_IMPACT_PRIORITY	495
SET WORKLOAD コマンド	498
SET WRITE	499
START DATABASE MANAGER	500
START HADR	507
STOP DATABASE MANAGER	510
STOP HADR	513
TAKEOVER HADR	514
TERMINATE	518
UNCATALOG DATABASE	519
UNCATALOG DCS DATABASE	520
UNCATALOG LDAP DATABASE	521
UNCATALOG LDAP NODE	522
UNCATALOG NODE	522
UNCATALOG ODBC DATA SOURCE	523
UNQUIESCE	524
UPDATE ADMIN CONFIGURATION	525
UPDATE ALERT CONFIGURATION	527
UPDATE ALTERNATE SERVER FOR DATABASE	532

UPDATE ALTERNATE SERVER FOR LDAP DATABASE	533
UPDATE CLI CONFIGURATION	534
UPDATE COMMAND OPTIONS	536
UPDATE CONTACT	537
UPDATE CONTACTGROUP	538
UPDATE DATABASE CONFIGURATION	539
UPDATE DATABASE MANAGER CONFIGURATION	543
UPDATE HEALTH NOTIFICATION CONTACT LIST	545
UPDATE HISTORY	545
UPDATE LDAP NODE	548
UPDATE MONITOR SWITCHES	549
UPDATE XMLSCHEMA	551

第 5 章 システム・コマンド 553

dasauto - DB2 Administration Server の自動始動	553
dasrct - DB2 Administration Server の作成	553
dasdrop - DB2 Administration Server の除去	554
dasmigr - DB2 Administration Server のマイグレーション	555
dasupdt - DAS の更新コマンド	556
db2_deinstall - DB2 製品またはフィーチャーのアンインストール	557
db2_install - DB2 製品のインストール	559
db2_local_ps - Linux/UNIX の DB2 プロセス状況	562
db2admin - DB2 Administration Server	563
db2adutl - TSM 内の DB2 オブジェクトの管理	565
db2advis - DB2 設計アドバイザー	572
db2audit - 監査機能管理者用ツール	580
db2batch - ベンチマーク・ツール	589
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コマンド	599
db2bfd - バインド・ファイル記述ツール	603
db2ca - 構成アシスタントの開始	604
db2cap - CLI/ODBC 静的パッケージ・バインディング・ツール	605
db2cat - システム・カタログ分析	606
db2cc - コントロール・センターの開始	608
db2cfexp - 接続構成エクスポート・ツール	610
db2cfimp - 接続構成インポート・ツール	612
db2chglbpath - 組み込みランタイム・ライブラリー検索パスの変更	613
db2chgpath - 組み込みランタイム・パスの変更	615
db2ckbkp - バックアップの検査	616
db2ckmig - データベース事前マイグレーション・ツール・コマンド	620
db2ckrst - 増分リストア・イメージ順序の検査	622
db2cli - DB2 対話機能 CLI	624
db2cmd - DB2 コマンド・ウィンドウのオープン	625
db2cptsa - DB2 HA スクリプトのインストールまたは更新コマンド	626
db2dart - データベース分析およびレポート・ツール	628
db2daslevel - DAS レベルの表示	632

db2dclgn - 宣言生成プログラム	633	db2mcses - Windows フェイルオーバー・ユーティリ ティーのセットアップ	745
db2diag - db2diag.log 分析ツール	635	db2mtrk - メモリー・トラッカー	749
db2drdat - DRDA トレース	650	db2nchg - データベース・パーティション・サーバ ー構成の変更	753
db2drvmp - DB2 データベースのドライブ・マップ	651	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	754
db2empfa - 複数ページ・ファイル割り振りの使用可 能化	652	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	756
db2envar.bat - 現行コマンド・ウィンドウの環境の 設定	653	db2nrcfg - 非 root インストール構成ツール・コマ ンド	757
db2eva - イベント・アナライザー	654	db2rfe - 非 root インストール用 root フィーチャー 使用可能化コマンド	758
db2evmon - イベント・モニター生産性向上ツール	655	db2nrupdt - 非 root インストールを行ったインスタ ンスの更新コマンド	760
db2evtbl - イベント・モニターのターゲット表定義 の生成	656	db2osconf - カーネル・パラメーター値のためのユ ーティリティー	760
db2exfmt - Explain 表フォーマット	658	db2pd - DB2 データベース	763
db2exmig - Explain 表のマイグレーション	660	db2pdcfg - 問題判別動作の DB2 データベースの 構成	824
db2expln - SQL および XQuery Explain	661	db2perfc - データベース・パフォーマンス値のリセ ット	828
db2extsec - DB2 オブジェクトの許可の設定	667	db2perfi - パフォーマンス・カウンター登録ユーテ ィリティー	830
db2flsn - ログ・シーケンス番号の検出	669	db2perfr - パフォーマンス・モニター登録ツール	830
db2fm - DB2 障害モニター	671	db2rbind - すべてのパッケージの再バインド	831
db2fmcu - DB2 障害モニター・コントローラー・コ マンド	673	db2relocatedb - データベースの再配置	833
db2fodc - DB2 第 1 出現データ収集コマンド	674	db2rfpen - ロールフォワード・ペンディング状態に リセット	838
db2fs - ファースト・ステップ	680	db2rspgn - 応答ファイル生成プログラム (Windows)	838
db2gcf - DB2 インスタンスの制御	681	db2sampl - サンプル・データベースの作成	839
db2gov - DB2 管理プログラム	683	db2schex - Active Directory スキーマ拡張コマンド	842
db2govlg - DB2 管理プログラム・ログ照会	685	db2set - DB2 プロファイル・レジストリー	843
db2gpmap - 分散マップの取得	686	db2setup - DB2 のインストール	847
db2hc - ヘルス・センターの開始	687	db2sqljbind - SQLJ プロファイル・バインダー	848
db2iauto - インスタンスの自動開始	688	db2sqljcustomize - SQLJ プロファイル・カスタマイ ザー	854
db2iclus - Microsoft Cluster Server	688	db2sqljprint - SQLJ プロファイル・プリンター	867
db2icrt - インスタンスの作成	691	db2start - Start DB2	867
db2idrop - インスタンスの除去	695	db2stat - Windows の DB2 プロセス状況	868
db2ilist - インスタンスのリスト	698	db2stop - DB2 の停止	869
db2imigr - インスタンスのマイグレーション	699	db2support - 問題分析および環境収集ツール	869
db2inidb - ミラーリングされたデータベースの初期 化	701	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	875
db2inspf - 検査結果のフォーマット	703	db2sync - DB2 シンクロナイザーの開始	876
db2iprune - インストール・イメージ・サイズの削 減コマンド (Windows)	704	db2systray - DB2 システム・トレイの開始	877
db2isetup - インスタンス作成インターフェースの開 始	705	db2tapemgr - テープ上のログ・ファイルの管理	879
db2iupdt - インスタンスの更新	707	db2tbst - 表スペース状態の獲得	882
db2jdbcbind - DB2 JDBC パッケージ・バインド・ プログラム	710	db2tdbmgr - 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	883
db2ldcfg - LDAP 環境の構成	712	db2trc - トレース	884
db2level - DB2 サービス・レベルの表示	713	db2uiddl - V5 セマンティクスへのユニーク索引変 換の準備	891
db2licm - ライセンス管理ツール	714	db2unins - DB2 データベース製品のアンインスト ール	892
db2listvolumes - すべてのディスク・ボリュームの GUID の表示	716		
db2logsforrwd - ロールフォワード・リカバリーに 必要なログのリスト	717		
db2look - DB2 統計および DDL 抽出ツール	717		
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	729		
db2move - データベース移動ツール	731		
db2mqdsn - MQ Listener	741		

db2untag - コンテナ・タグの解放	894
db2xdbmig - XSR オブジェクトのマイグレーション・コマンド	895
db2xpvt - トラップ・ファイルのフォーマット	895
disable_MQFunctions - WebSphere MQ 関数の無効化コマンド	896
doce_deinstall - DB2 インフォメーション・センター	897
doce_install - DB2 インフォメーション・センターのインストール	898
enable_MQFunctions - WebSphere MQ 関数の有効化コマンド	899
installFixPack - インストール済み DB2 製品の更新	901
setup - DB2 のインストール	904
sqlj - SQLJ 変換プログラム	906

第 6 章 DB2 Text Search のコマンド 911

db2ts ALTER INDEX	911
db2ts CLEANUP FOR TEXT	913
db2ts CLEAR COMMAND LOCKS	914
db2ts CLEAR EVENTS FOR INDEX	916
db2ts CREATE INDEX	917
db2ts DISABLE DATABASE FOR TEXT	923
db2ts DROP INDEX	925
db2ts ENABLE DATABASE FOR TEXT	927
db2ts HELP	928
db2ts START FOR TEXT	930
db2ts STOP FOR TEXT	930
db2ts UPDATE INDEX	931

付録 A. 命名規則 935

命名規則	935
------	-----

付録 B. ファイル・タイプ修飾子および区切り文字 937

エクスポート、インポート、およびロード・ユーティリティー用のファイル・タイプ修飾子	937
データ移動のための区切り文字の考慮事項	937

付録 C. DB2 技術情報の概説 941

DB2 テクニカル・ライブラリー (ハードコピーまたは PDF 形式)	942
DB2 の印刷資料の注文方法	945
コマンド行プロセッサから SQL 状態ヘルプを表示する	946
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	946
DB2 インフォメーション・センターでの希望する言語でのトピックの表示	946
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	947
DB2 チュートリアル	949
DB2 トラブルシューティング情報	950
ご利用条件	950

付録 D. 特記事項 953

索引 957

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を記載しています。

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 テキスト・サーチの各コマンドを使用するために必要な参照情報を記載しています。

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含まれています。

コマンド使用法ヘルプ

- 第 3 章『コマンド・ヘルプ画面の読み方』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 4 章『CLP コマンド』では、すべてのデータベース・マネージャ・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 5 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 テキスト・サーチのコマンド

- 第 6 章『DB2 テキスト・サーチのコマンド』では、db2ts を接頭部としてオペレーティング・システム・コマンド・プロンプトに入力できるすべてのテキスト・サーチ・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・ユーティリティーのファイル・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。

強調表示規則

本書では、以下の強調表示規則を使用します。

太字	名前がシステムによって定義済みのコマンドやキーワードなどの項目を示します。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	以下のいずれかを示します。 <ul style="list-style-type: none">• ユーザーが指定する必要がある名前または値 (変数)• 一般的な強調• 初出の新規用語• 別の情報ソースへの参照
モノスペース	以下のいずれかを示します。 <ul style="list-style-type: none">• ファイルおよびディレクトリー• コマンド・プロンプトまたはウィンドウに入力するよう指示されている情報• 特定のデータ値の例• システムによって表示される内容に該当するテキストの例• システム・メッセージの例• プログラミング・コードのサンプル

構文図の見方

構文図の構造を理解するために、以下の情報を参考にできます。

構文図は、左から右、上から下に、線に沿って読みます。

▶▶— 記号は、構文図の始まりを示します。

—▶ 記号は、構文が次の行に続くことを示します。

▶— 記号は、構文が前の行から続いていることを示します。

—▶▶ 記号は、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。

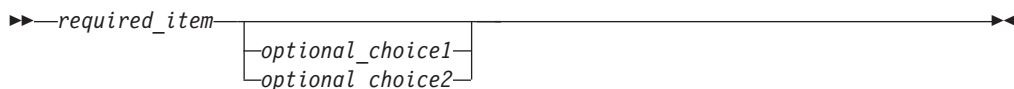


複数の項目からの選択が可能な場合、それらの項目を縦に並べて (スタックに) 示しています。

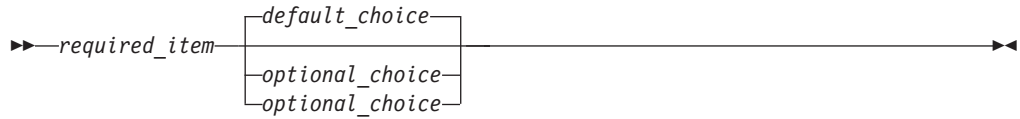
項目から 1 つを選択しなければならない場合、スタックの項目の 1 つはメインパス上に示されます。



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。



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。

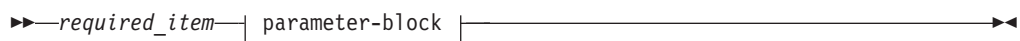


スタックの上部の反復の矢印の記号は、そのスタックの中から複数の項目を選択できること、または 1 つの選択項目を繰り返して選択できることを示します。

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。

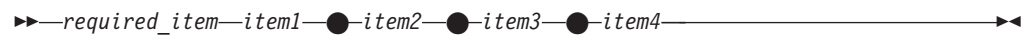
1 つの変数が、構文を構成する大きいフラグメントを表すことがあります。例えば次の図で、変数 `parameter-block` は、**parameter-block** というラベルの構文フラグメント全体を表します。



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
```

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

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。

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している必要があります。データベースに接続するには、以下のいずれかを行います。

- 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 と呼ばれる文字列を NULL 文字列として識別します。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 文字列としては解釈されず、null を含む文字列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は最大長 100 の任意のテキスト・文字列に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	ランタイムの値
%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、DB2INSTANCE レジストリー変数の値。Windows® プラットフォーム上でのみ、DB2INSTANCE レジストリー変数が設定されていない場合、DB2INSTDEF 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、DB2DBDFT 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。

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。
(@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'
- . foo (foo 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。

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

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ることを知らせま

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はしますが、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、上記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。

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。

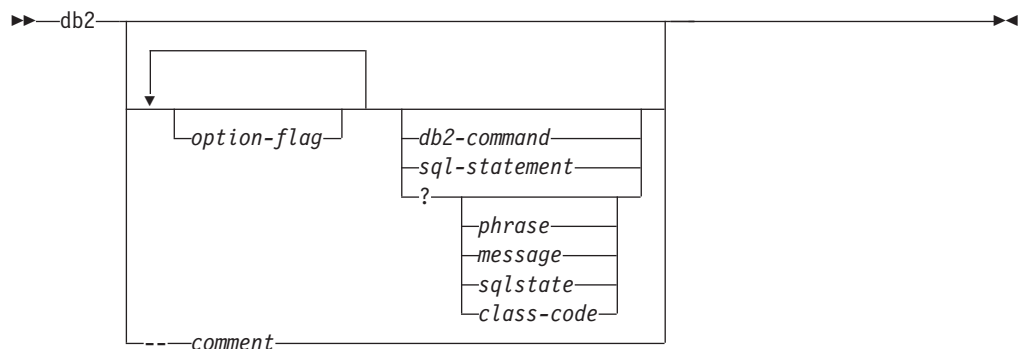
- 対話式入力モード。特徴は **db2 =>** 入力プロンプトです。
- コマンド・モード。各コマンドの前に db2 を付ける必要があります。
- バッチ・モード。-f ファイル入力オプションを使用します。

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、UNIX® ベースのシステム、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。

? 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ます。

9 ページの表 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 `auto-commit` 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
<code>-a</code>	コマンド行プロセッサは SQLCA データを表示します。	OFF
<code>-c</code>	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
<code>-d</code>	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
<code>-e{cls}</code>	コマンド行プロセッサは SQLCODE または SQLSTATE 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
<code>-f filename</code>	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
<code>-i</code>	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
<code>-l filename</code>	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
<code>-m</code>	このオプションは、コマンド行プロセッサが、INSERT/DELETE/UPDATE/MERGE で影響を受ける行の数を印刷するよう指示します。	OFF
<code>-n</code>	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、 <code>-t</code> オプションと共に使用する必要があります。	OFF
<code>-o</code>	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
<code>-p</code>	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
<code>-q</code>	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 <code>q</code> が ON のとき、オプション <code>n</code> は無視されます。	OFF
<code>-r filename</code>	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
<code>-s</code>	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
<code>-t</code>	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

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
-tdx または -tdxx	このオプションは、コマンド行プロセッサが <i>x</i> または <i>xx</i> 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、コマンド行プロセッサが FETCH/SELECT 警告メッセージを表示するよう指示します。	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 (LIST TABLES 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

例

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

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 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、他のコマンド行プロセッサ・オプションに影響することはありません。

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に失敗します。そのステートメントは自動コミット ON (デフォルト) で発行されたので、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。ステップ 3 が自動コミット OFF で発行されているためです。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
```

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。

整形印刷オプション (-i):

-i オプションにより、コマンド行プロセッサは、適切な字下げを使って XML データを「整形印刷」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 (+1 または -1-)。

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` オプションにより、コマンド行プロセッサは、INSERT、DELETE、UPDATE、または MERGE で影響を受ける行数を印刷するかどうかを決定します。

ON (-m) に設定した場合、INSERT/DELETE/UPDATE/MERGE のステートメントで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管されません。
- `-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 `-rfilename` 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管オプション (-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管オプションは、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。ネーム・スペース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、ネーム・スペース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。以下に例を示します。

```
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/SELECT) 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管オプション (-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管オプションは、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

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 を返し、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ければ、コマンドの実行は続きます。

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。

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。

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。

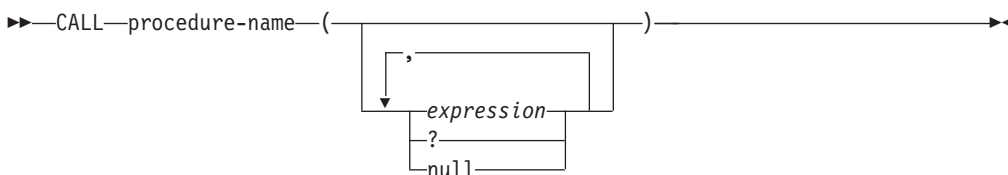
コマンド行から SQL ステートメントを使用できます。SQL を介して CLP コマンドを実行するストアード・プロシージャ (SYSPROC.ADMIN_CMD()) を使用できます。このストアード・プロシージャの使用方法についての詳細は、「SQL 管理ルーチン」を参照してください。

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。ネーム・スペース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、ネーム・スペース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、23 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、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異なります。これらのステートメントの CLP 構文を次に示します。

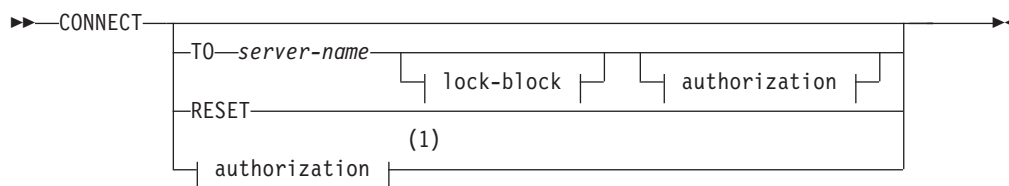
CALL



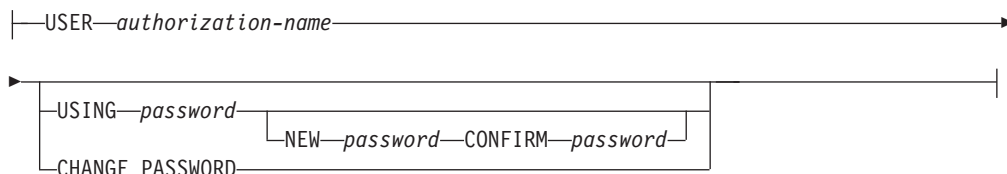
CLOSE



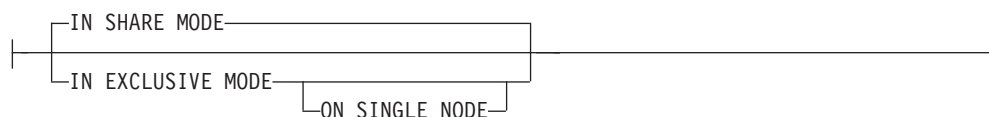
CONNECT



許可:



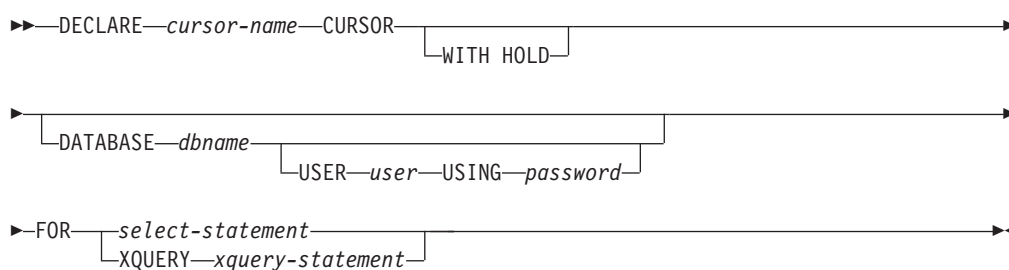
lock-block:



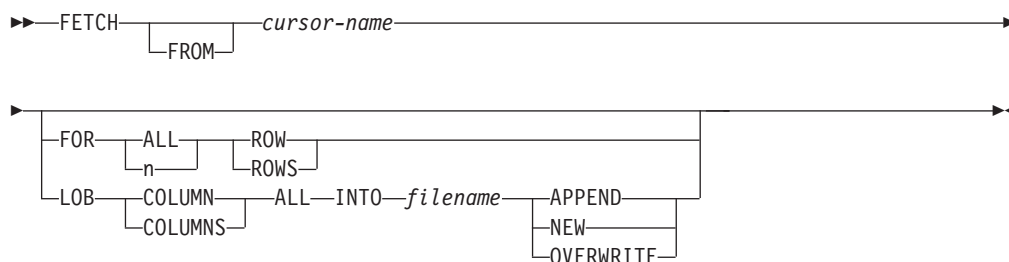
注:

- 1 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



FETCH



OPEN

注:

1. 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。

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

2. 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。

3. 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も適用されます。
4. 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。
5. 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。
6. 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
```

7. 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。
 - 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 Statement	動的 ¹	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 ³ (CLI)	SQL プロシ ャー
ALLOCATE CURSOR				X
割り当てステートメント				X
ASSOCIATE LOCATORS				X
ALTER { BUFFERPOOL, NICKNAME, ⁹ NODEGROUP, SERVER, ⁹ TABLE, TABLESPACE, USER MAPPING, ⁹ TYPE, VIEW }	X	X	X	
BEGIN DECLARE SECTION ²				
CALL	X	X	X	X
CASE ステートメント				X
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 ⁴	
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()	

表 5. SQL ステートメント (DB2) (続き)

SQL Statement	動的 ¹	コマンド行 プロセッサ ー (CLP)	コール・レベル・ (CLI)	インターフェース ³	SQL プロシ ージャー
CREATE { ALIAS, BUFFERPOOL, DISTINCT TYPE, EVENT MONITOR, FUNCTION, FUNCTION MAPPING, ⁹ INDEX, INDEX EXTENSION, METHOD, NICKNAME, ⁹ NODEGROUP, PROCEDURE, SCHEMA, SERVER, TABLE, TABLESPACE, TRANSFORM, TYPE MAPPING, ⁹ TRIGGER, USER MAPPING, ⁹ TYPE, VIEW, WRAPPER ⁹ }	X	X	X		X ¹⁰
DECLARE CURSOR ²		X	SQLAllocStmt()		X
DECLARE GLOBAL TEMPORARY TABLE	X	X	X		X
DELETE	X	X	X		X
DESCRIBE ⁸		X	SQLColAttributes(), SQLDescribeCol(), SQLDescribeParam() ⁶		
DISCONNECT		X	SQLDisconnect()		
DROP	X	X	X		X ¹⁰
END DECLARE SECTION ²					
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 ⁴		X
GET DIAGNOSTICS					X
GOTO ステートメント					X
GRANT	X	X	X		X
IF ステートメント					X
INCLUDE ²					
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

表 5. SQL ステートメント (DB2) (続き)

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

表 5. SQL ステートメント (DB2) (続き)

SQL Statement	動的 ¹	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 ³ (CLI)	SQL プロシ ージャー
---------------	-----------------	---------------------------	--	-----------------

注:

1. 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。
2. このステートメントは実行できません。
3. 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 DB2 CLI 機能がある場合は、機能名がリストされています。
4. このステートメントは動的ではないものの、DB2 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。
5. `CREATE TRIGGER` ステートメント内だけで使用できます。
6. `SQL DESCRIBE` ステートメントで出力の記述ができますが、DB2 CLI を使用すると、入力の記述も可能となります (`SQLDescribeParam()` 関数を使用する場合)。
7. `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、DB2 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。
8. `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。
9. 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。
10. 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 コマンド構文ヘルプの読み方

特定の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。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* オプション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。コマンド・ウィンドウに表示される便利なコマンド・ヘルプ画面の出力では、以下に示す構文規則を使用しています。

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

| 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: SQL10007N Message "227" could not be retrieved. Reason code: "4".
```

- 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、次のコマンドを入力します。

```
=> db2 ? help
```

第 4 章 CLP コマンド

ACTIVATE DATABASE

指定されたデータベースを活動化し、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、システム内のすべてのノードで、指定したデータベースを活動化します。データベースを活動化中に 1 つ以上のノード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ノードで、データベースは活動状態を維持します。

許可

以下のいずれか。

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

必要な接続

なし

コマンド構文

```
▶▶ ACTIVATE DATABASE database-alias
      DB
▶
  USER username
      USING password
▶▶
```

コマンド・パラメーター

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

データベースが開始していないときに、アプリケーションで `CONNECT TO` (または暗黙的な接続) が発行された場合、アプリケーションは、データベースに対して作業する前に、必要なデータベースをデータベース・マネージャーが開始する間待機しなければなりません。しかし、データベースが開始してしまえば、他のアプリケーションはデータベースの開始を待たずに、ただデータベースに接続するだけでその中のデータを使用することができます。

データベース管理者は、`ACTIVATE DATABASE` 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` で初期化したデータベースは、`DEACTIVATE DATABASE` コマンドまたは `db2stop` コマンドを使用してシャットダウンできます。

`CONNECT TO` (または暗黙的な接続) によってデータベースを開始し、その後、その同じデータベースに `ACTIVATE DATABASE` 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` を使用しなければなりません。`ACTIVATE DATABASE` 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。

`ACTIVATE DATABASE` コマンドは、再始動が必要なデータベース (例えば、不整合状態にあるデータベース) を処理する場面においては、`CONNECT TO` (または暗黙的な接続) と同じように機能します。`ACTIVATE DATABASE` で初期設定される前に、データベースは再始動します。再始動を実行できるのは、データベースが `AUTORESTART ON` で構成されている場合だけです。

`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。

ADD CONTACT

このコマンドは、システムでローカルに定義されるか、またはグローバル・リストで定義される、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) `contact_host` 構成パラメーターの設定は、リストがローカルかグローバルかを判別します。

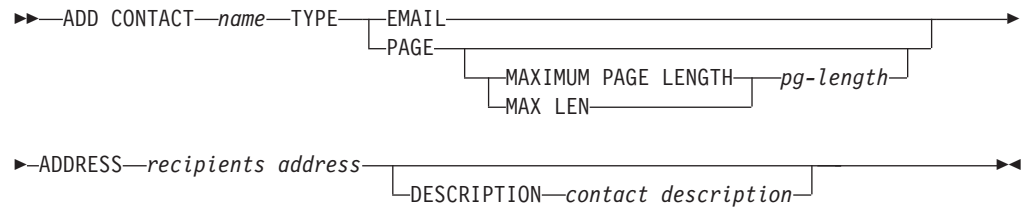
許可

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) **contact_host** 構成パラメーターの設定は、リストがローカルかグローバルかを判別します。

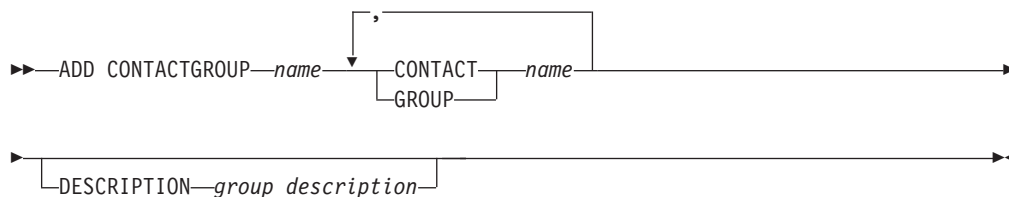
許可

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

新しいデータベース・パーティション・サーバーをパーティション・データベース環境に追加します。このコマンドは、新規のデータベース・パーティション・サーバーに置かれるすべてのデータベースのデータベース・パーティションも作成します。ユーザーは、新しいデータベース・パーティションとともに作成する SYSTEM TEMPORARY 表スペースの定義用のソース・データベース・パーティション・サーバーを指定できます。または SYSTEM TEMPORARY 表スペースを作成しないように指定することもできます。コマンドは、追加するデータベース・パーティション・サーバーから発行されなければなりません。

有効範囲

このコマンドは、それが実行されたマシンに対してだけ影響を与えます。

許可

以下のいずれか。

- *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 (つまり 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) 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、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。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 (つまり 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) 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

新規のデータベース・パーティション・サーバーを追加する前に、インスタンス内のすべてのデータベース用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。

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。

システム内のデータベースにおいて、XML 列を持つユーザー表が作成された場合や、XSR オブジェクトが登録された場合には、(そのような操作が成功したかどうかにかかわらず) このコマンドは失敗します。

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、`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。

互換性

バージョン 8 より前のバージョンとの互換性：

- キーワード `NODE` は、`DBPARTITIONNUM` の代わりとして使用できます。

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、既存の未完成の XML スキーマに追加します。

許可

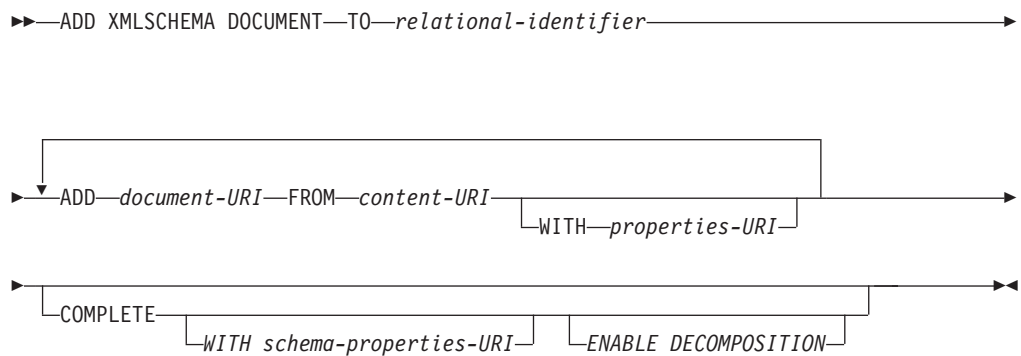
以下の権限が必要です。

- 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成の 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。

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

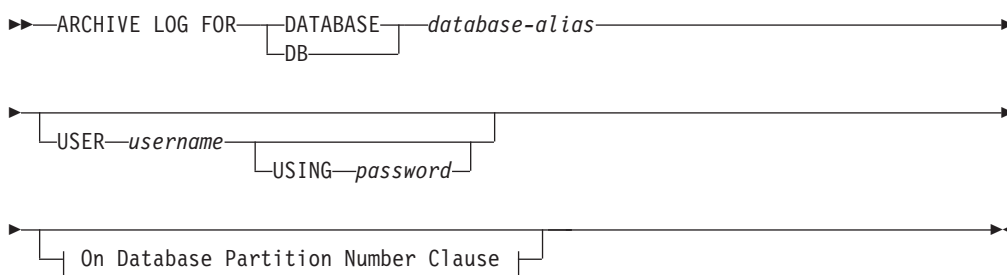
以下のいずれか。

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

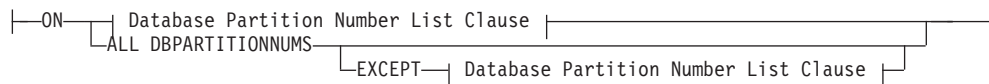
必要な接続

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。

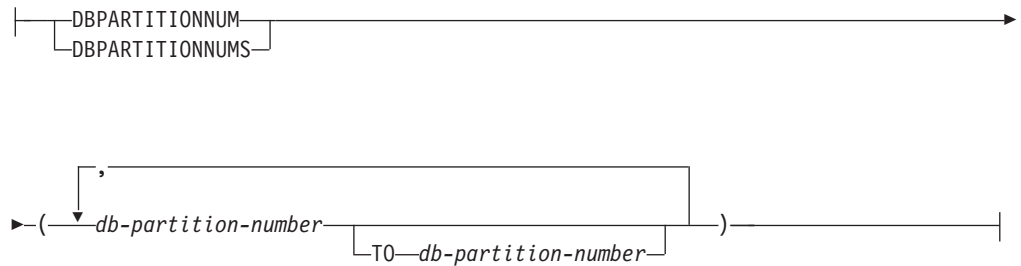
コマンド構文



On Database Partition Number Clause:



Database Partition Number List Clause:



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識別します。

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・ファイル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・ファイル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。

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

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。

互換性

バージョン 8 より前のバージョンとの互換性：

- ・ キーワード `NODE` は、`DBPARTITIONNUM` の代わりとして使用できます。
- ・ キーワード `NODES` は、`DBPARTITIONNUMS` の代わりとして使用できます。

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

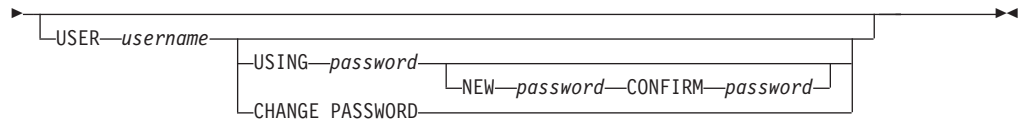
なし

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

```
▶▶ ATTACH [TO nodename]
```



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 (NetBIOS 形式) の名前であればなりません (最大長は 15 文字)。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。パスワードに関する追加情報は、「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。暗黙になります)、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 が実行されていない場合、インスタンス・レベル・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で実行されます。

AUTOCONFIGURE

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・マネージャの構成パラメータの初期値を計算し、これらの推奨値を適用するオプションと共に表示します。

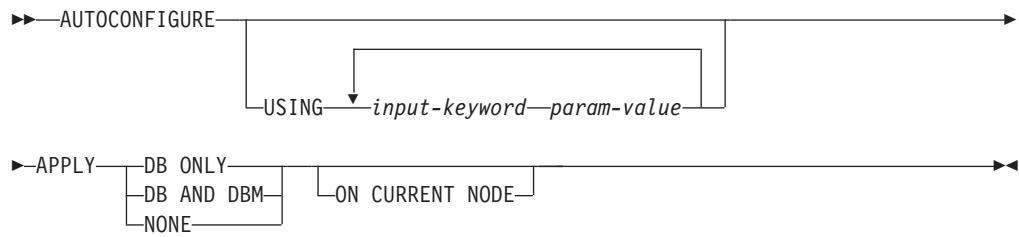
許可

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 6. 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	1-100	25	専用にするメモリーのパーセンテージ。他のアプリケーション (オペレーティング・システム以外) がこのサーバーで実行している場合、この値は 100 未満に設定してください。
workload_type	simple、mixed、complex	mixed	単純 (simple) ワークロードは入出力集約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 集約の傾向があり大部分が照会です。
num_stmts	1-1 000 000	10	作業単位ごとのステートメント数
tpm	1-200 000	60	1 分ごとのトランザクション
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
is_populated	yes、no	yes	データベースがデータで移植されるかどうか
num_local_apps	0-5 000	0	接続されたローカル・アプリケーションの数
num_remote_apps	0-5 000	10	接続されたリモート・アプリケーションの数

表 6. 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
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。
bp_resizeable	yes、no	yes	バッファーク・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ーク・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ーク・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、およびバッファーク・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 NODE

デフォルトでは、データベース・パーティション・フィーチャー (DPF) において、構成アドバイザーによりすべてのノード上のデータベース構成が更新されます。ON CURRENT NODE オプションを指定して実行すると、アドバイザーにより推奨データベース構成が適用される対象となるのは、コーディネーター (接続) ノードだけになります。

バッファーク・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ノードが影響を受けます。ON CURRENT NODE オプションはバッファーク・プールの推奨値に影響しません。

使用上の注意

- 複数のロジカル・パーティションのあるシステムでは、**mem_percent** パラメーターは、すべてのロジカル・パーティションによって使用されるメモリーのパーセントを参照します。例えば、DB2 がシステムの 25% のメモリーを使用している場合は、ロジカル・パーティションの数に関係なく、25% を指定します。データベースの推奨構成が作成されますが、これは 1 つのロジカル・パーティション用に調整されます。
- このコマンドは、現在接続されているデータベース (このデータベースがシステム上で唯一のアクティブ・データベースであると仮定) のための、推奨構成を作成します。複数のデータベースがシステムでアクティブな場合は、**mem_percent** パラメーターを調整して、現行データベースのメモリーの共用を反映します。例えば、DB2 データベースがシステム・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・データベースがシステムにある場合、40% (80% を 2 データベースで割る) をパラメーター **mem_percent** 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せん。
- 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。

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・ノードに対してだけです。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

以下のいずれか。

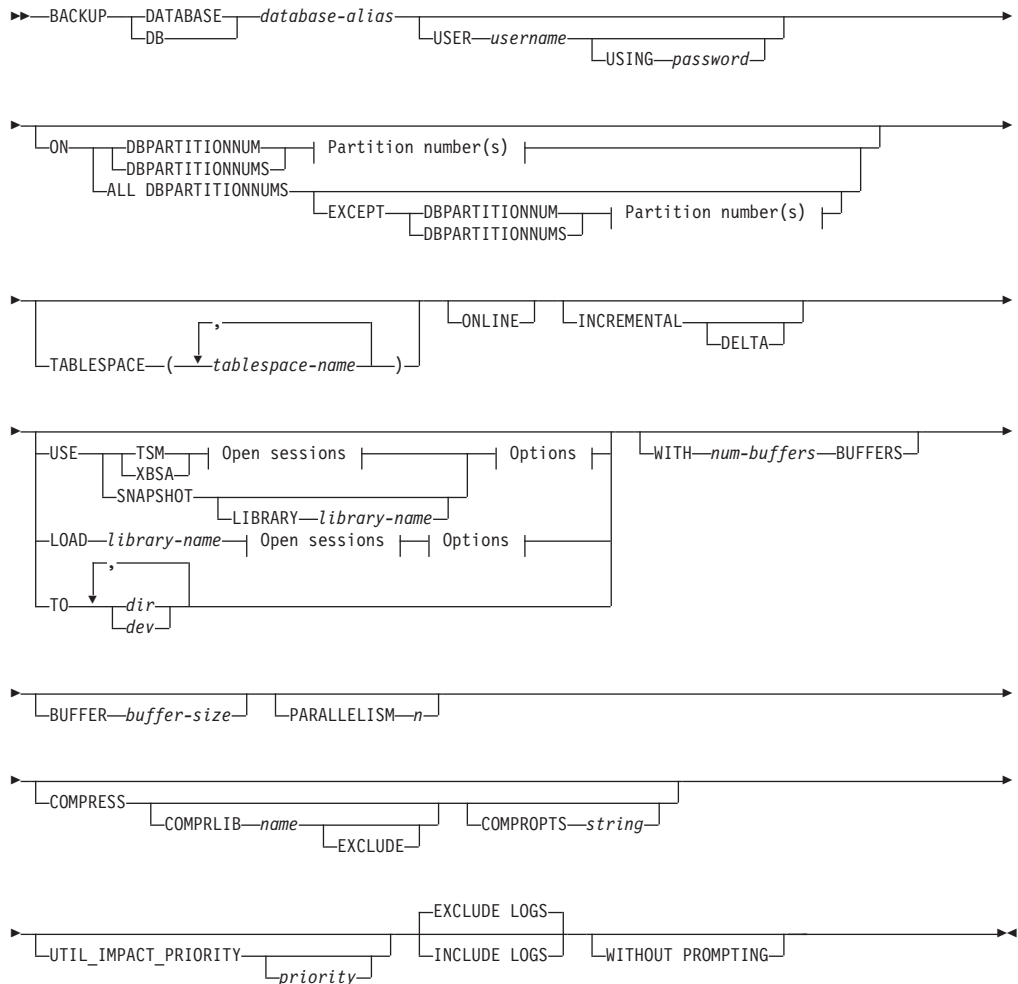
- *sysadm*
- *sysctrl*
- *sysmaint*

必要な接続

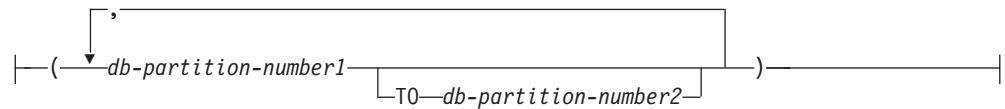
データベース。このコマンドは、指定されたデータベースへの接続を自動的に確立します。

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。

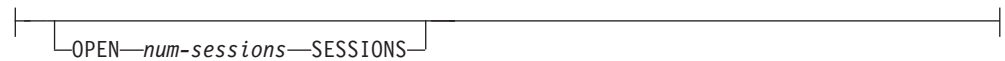
コマンド構文



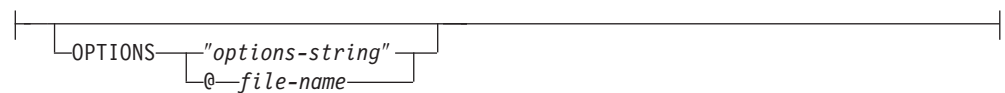
Partition number(s):



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レストラン。

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、*logretain* または *userexit* 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 (共用ロック) 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、正常に実行された全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 バックアップに Tivoli® Storage Manager (TSM) 出力を使用することを指定します。

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 パラメーターは、以下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、すべてのコンテナ、ローカル・ボリューム・ディレクトリ、データベース・パス (DBPATH)、および 1 次ログとミラー・ログのパスを含む、データベースを構成するすべてのパスの FULL DATABASE OFFLINE バックアップです (明示的に EXCLUDE LOGS が指定されているのでない限り、すべてのスナップショット・バックアップでのデフォルトは INCLUDE LOGS です)。

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 System Storage DS6000
- IBM System Storage DS8000
- IBM System Storage N Series
- NetApp V-series
- NetApp FAS

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。入力されたストリングは、二重引用符を付加せずに、そのまま DB2 ACS API ドライバーに渡されます。**VENDOROPT** データベース構成パラメーターでは、ベンダー固有のスナップショット・バックアップ操作の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・ユーティリティの **OPTIONS** パラメーターを使用する必要があります。

@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。

OPEN *num-sessions* **SESSIONS**

DB2 と TSM または他のバックアップ・ベンダー製品との間で作成され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。

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 が省略される場合には、バックアップ・イメージ用のデフォルト・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となります。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。オプションで共用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。

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 はこの

パラメーターの最適値を自動的に選択します。ただし、バックアップを複数の場所に作成する場合は、パフォーマンスを向上させるために多数のバッファを使用することができます。

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 の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。

$$\text{bufferpages} \leq \text{ST_MAX_BUFFERS} * \text{ST_BUFFER_BLOCKS} / 4$$

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示します (例えば、Windows の場合は db2compr.dll、Linux/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、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 '@' なら、データの残りの部分は、DB2によって、サーバー上に存在するファイルの名前として解釈されます。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。以下のバックアップ・シナリオの場合、デフォルトでログは含められません。

- 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
- シングル・システム・ビュー・バックアップを使用しない複数パーティション・データベースのオンラインまたはオフライン・バックアップ

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・バックアップは例外です。以下のバックアップ・シナリオの場合、デフォルトでログは含められます。

- 単一パーティション・データベースのオンライン・バックアップ
- 複数パーティション・データベースの、オンラインまたはオフラインのシングル・システム・ビュー (SSV) バックアップ
- オンラインまたはオフラインのスナップショット・バックアップ

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をバックアップするかが示されている場合、バックアップ操作はプロンプトを出さずに暗黙的に実行されます。
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せん。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・イメージはすべて `/dev3/backup` ディレクトリーに保管されます。このディレクトリーは、複数のパーティションからアクセス可能な共用ディレクトリーにすることもできますし、各パーティションから個別にアクセスできるローカルにマウントされるディレクトリーにすることもできます。また、この両方を組み合わせたディレクトリーにすることも可能です。

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` というストリングが含まれています。

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
```

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ます。

```
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。

- 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
```

- 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
-----
0    DB20000I The BACKUP DATABASE command completed successfully.
1    SQL2430W The database backup succeeded, but the following
      table spaces do not exist on this database partition: "T1".
```

Backup successful. このバックアップ・イメージのタイム・スタンプは、次のようになります。

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
-----
0    DB20000I The BACKUP DATABASE command completed successfully.
1    SQL2419N The target disk "/backups" has become full.
```

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

- 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
```

- 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。

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。

- *sysadm* または *dbadm* の権限
- パッケージが存在しない場合は、BINDADD 権限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)

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

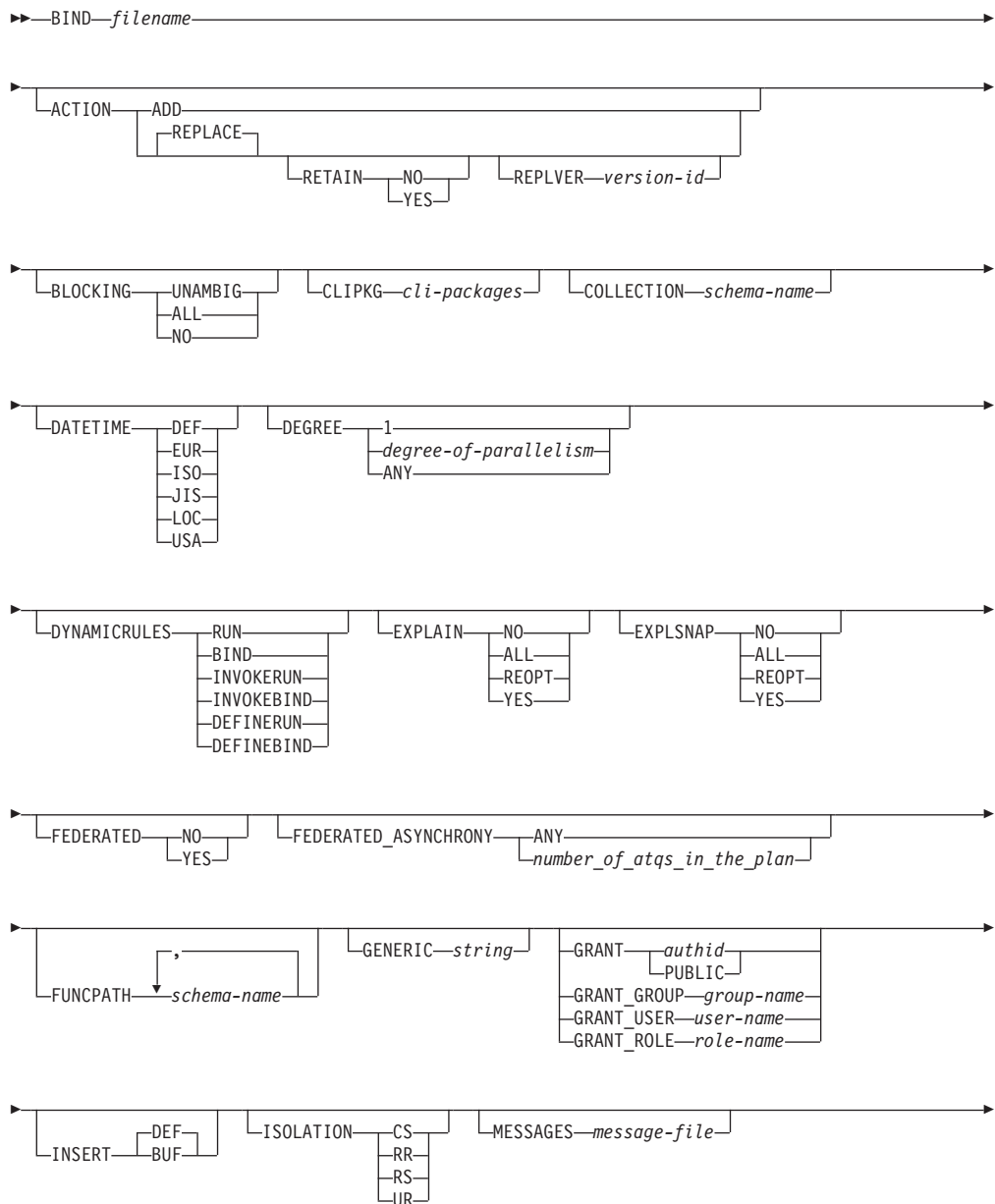
検査に使用されることはありません。ユーザーに `sysadm` 権限があつてバインドを完了させる明示特権がない場合、データベース・マネージャーは、明示的な `dbadm` 権限を自動的に付与します。

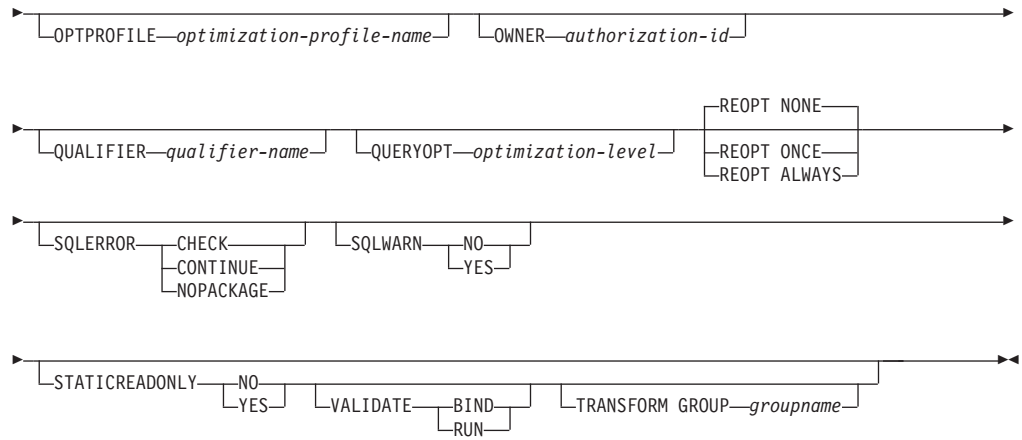
必要な接続

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。

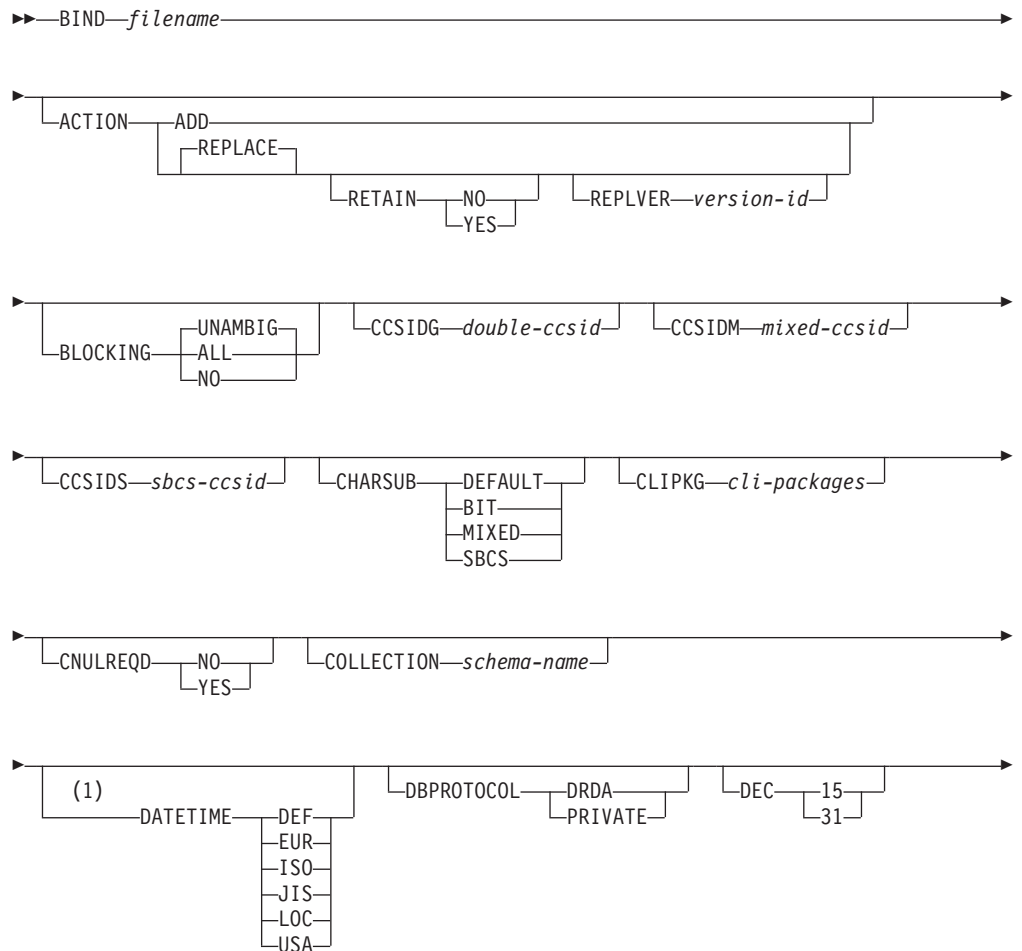
コマンド構文

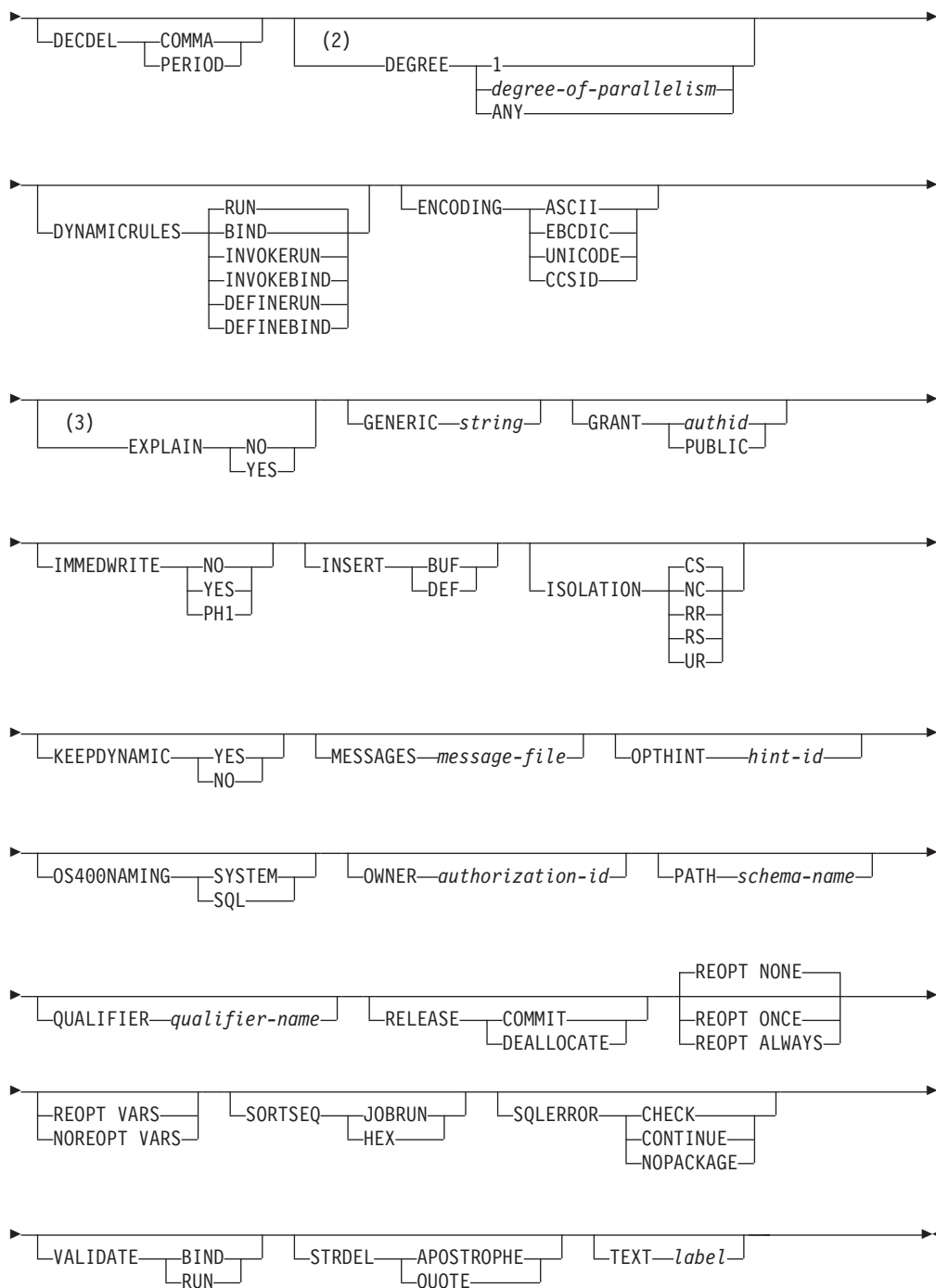
DB2 for Linux, Windows and UNIX では





Linux、Windows および UNIX 以外のサーバー上の DB2 では





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、

```
/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 **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いない場合、結合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。

BLOCKING

カーソルの行ブロッキングのタイプを指定します。LOB 列データ・タイプへの参照が含まれる行データのブロック化は、データベース・パーティション・フィーチャーが使用可能な環境で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。

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。なお、DB2 Database 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、システムが定義した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。なお、DB2 Database 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、システムが定義した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 Database 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、システムが定義した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。なお、DB2 Database 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。

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・システムが定義したデフォルト・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、DRDA でサポートされていない **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。なお、DB2 Database for Linux, UNIX, and Windows はこの DRDA BIND オプションをサポートしません。

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。

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。なお、DB2 Database 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。このオプションを指定しないと、DRDA サーバーは、システムが定義した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。なお、DB2 Database 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。このオプションを指定しないと、DRDA サーバーは、システムが定義した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 32767 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 とバインドし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 とバインドし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 とバインドし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 とバインドし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。DRDA では、このオプションの ALL 値がサポートされていません。

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 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)。

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 レジスターが NO に設定されていても、Explain 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)。DRDA では、EXPLAIN のこの値 (ALL) はサポートされていません。

EXPLSNAP

Explain 表に Explain スナップショットを保管します。この DB2 プリコンパイル/ BIND オプションは、DRDA ではサポートされていません。

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 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)。

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す。このオプションは、DRDA ではサポートされていません。

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 to 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。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

ターゲット・データベースで定義されていても、DRDA でサポートされていない新規 BIND オプションをサポートします。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。このオプションは、動的 SQL のパフォーマンスをかなり向上させることができます。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、ターゲット DRDA データベースが DB2 Universal Database バージョン 8 の場合、次のようにします。

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


これにより、**EXPLSNAP**、**QUERYOPT**、および **FEDERATED** の各オプションをバインドすることができます。

ストリングの最大長は 32768 バイトです。

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。GRANT ステートメントで grantee のタイプを指定するために USER、GROUP、または ROLE のいずれも指定されていない場合、SQL GRANT ステートメントとその規則を使用して authid 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を指定します。なお、DB2 Database 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。

KEEPDYNAMIC

コミット・ポイントの後に動的 SQL ステートメントを保持するかどうかを指定します。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。

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検証します。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[®] 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

ンを使用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。SYSADM または 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。この DB2 プリコンパイル/ BIND オプションは、DRDA ではサポートされていません。

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 Database 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。

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 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。

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 VARS | 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 として扱うかどうかを指定します。この DB2 プリコンパイル/ BIND オプションは、DRDA ではサポートされていません。

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。なお、DB2 Database 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。このオプションを指定しないと、DRDA サーバーは、システムが定義した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。

TEXT *label*

パッケージの記述。最大長は 255 文字です。また、デフォルト値は空白です。なお、DB2 Database 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。

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。このオプションは DRDA ではサポートされていません。

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。
- 上記のいずれの条件もない場合には、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 yyyyyy, 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 yyyyyy, 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 yyyyyy, 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、yyyyyy 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す。
3. 特殊レジスタ **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では、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したりできます。

パッケージのバインドに使用される許可 ID (**OWNER BIND** オプションの値) に付与されたロールからの特権または **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。 **BIND** オプションが明示的に指定されていない場合、**federated_async** 構成パラメーターの値が使用され、カタログには **FEDERATED_ASYNCHRONY** 列の値 -2 が示されます。

パッケージのバインド時に **FEDERATED_ASYNCHRONY BIND** オプションが明示的に指定されていない場合、およびこのパッケージが暗黙的または明示的に再バインドされている場合、**federated_async** 構成パラメーターの現行値を使用してパッケージが再バインドされます。

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、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

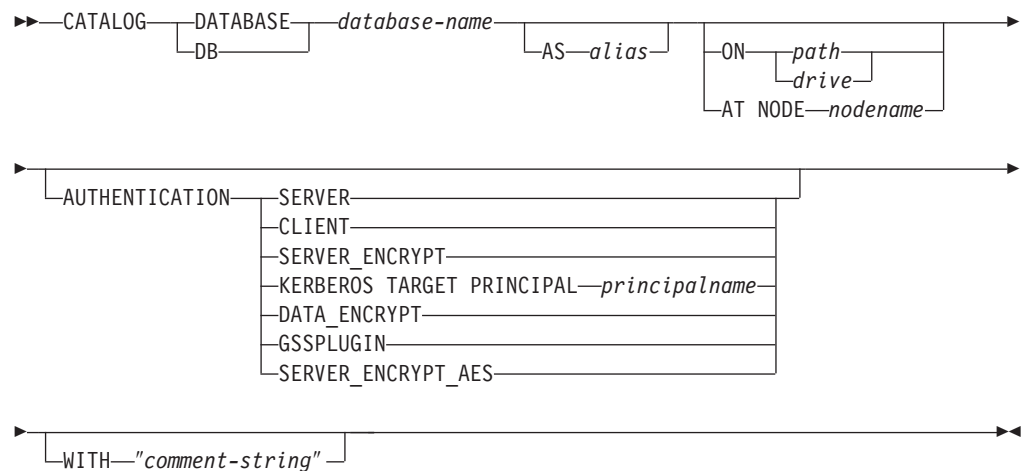
以下のいずれか。

- 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)。

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。この認証タイプは、DB2 バージョン 9.5 フィックスパック 3 以降で使用できます。

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。

ディレクトリーをキャッシュできる場合、データベースおよびノードのディレクトリー・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 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りリモート・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。UNIX ベースのシステムでは、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[®] 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[®] (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。

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 LDAP DATABASE

Lightweight Directory Access Protocol (LDAP) でデータベースを登録します。

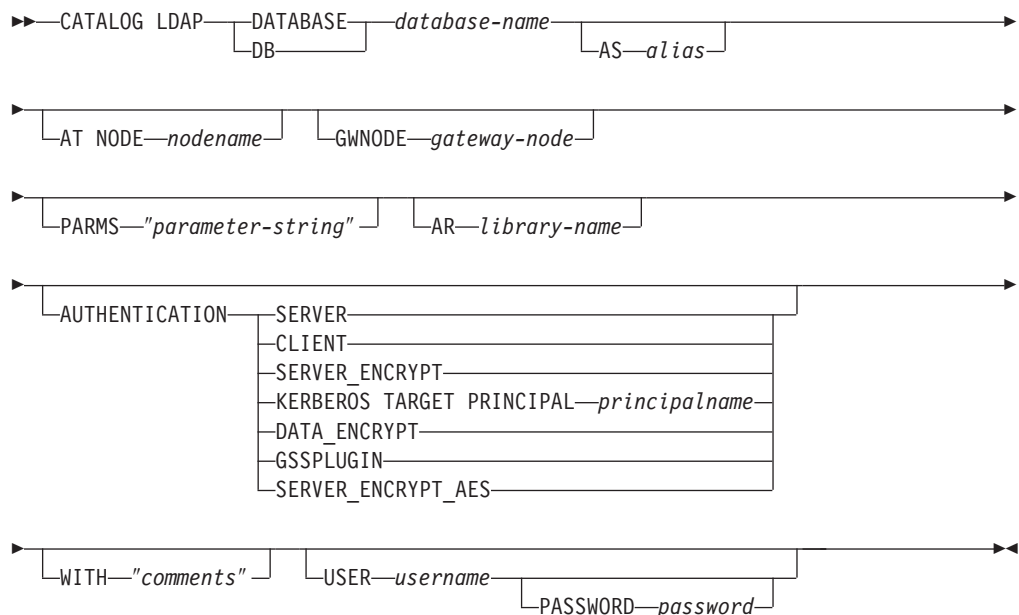
許可

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:sqllibdll` 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/sql/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、アプリケーションの呼び出し元であるデータベース・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たとおり、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、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・セキュリティー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。この認証タイプは、DB2 バージョン 9.5 フィックスパック 3 以降で使用できます。

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・ユーザーの信用証明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、zSeries® 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。
- LDAP での CREATE DATABASE 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。

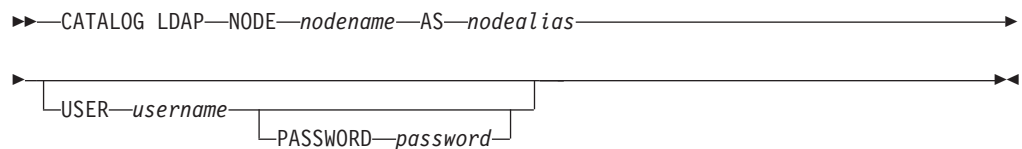
許可

なし

必要な接続

なし

コマンド構文



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・ユーザーの信用証明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

以下のいずれか。

- *sysadm*
- *sysctrl*

必要な接続

なし

コマンド構文

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

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。有効な値は次の通りです。AIX、WIN、HPUX、SUN、OS390、OS400、VM、VSE、SNI、SCO、LINUX、および DYNIX。

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 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、次のコマンドを出してください (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
```

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

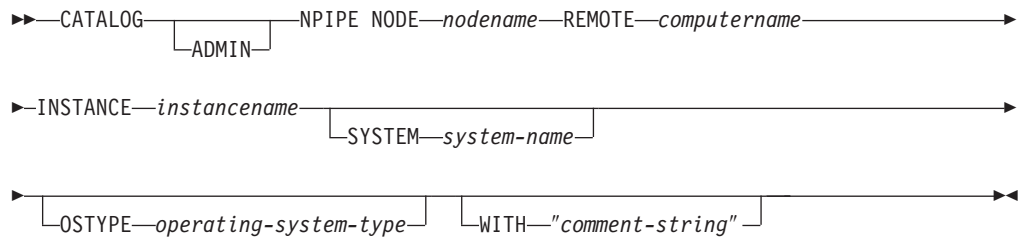
以下のいずれか。

- *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。有効な値は次のとおりです。AIX、WIN、HPUX、SUN、OS390、OS400、VM、VSE、SNI、SCO、および LINUX。

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。

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。

ODBC 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が同じである必要があります。そのため、データ・ソースの名前は、データベース名の最大長までに制限されます。

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。

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 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 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/TCP4/TCP6 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/TCP4/TCP6 NODE コマンドは、クライアントで実行されます。

許可

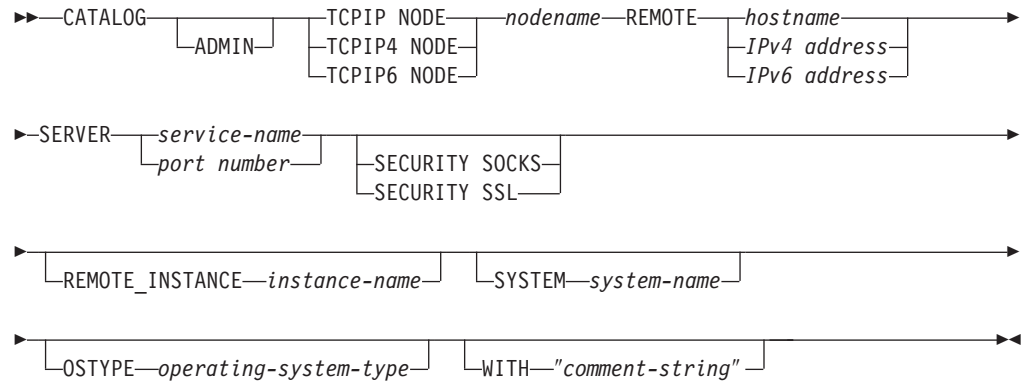
以下のいずれか。

- *sysadm*
- *sysctrl*

必要な接続

なし。ディレクトリ操作は、ローカル・ディレクトリだけに影響します。

コマンド構文



コマンド・パラメーター

ADMIN

カタログする TCP/IP Administration Server ノードを指定します。

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 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。

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

ターゲット・データベースが常駐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/小文字の区別があります。

サービス名を指定する場合、クライアントのサービス・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。サーバーのサービス・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・サービス・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。完全修飾ホスト名を解決するために SOCKS 化した IBM Data Server Client を使用できない場合、IPv4 アドレスが既に入力されたと想定されます。

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。

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。

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。

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 で 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 5.3 上でどのようにすればこの構成を行えるかについて確認するには、IPv4 構成の IPv6 へのアップグレードに従ってください。

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

以下のいずれか。

- *sysadm*
- *sysctrl*

必要な接続

なし

コマンド構文

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

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。

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。

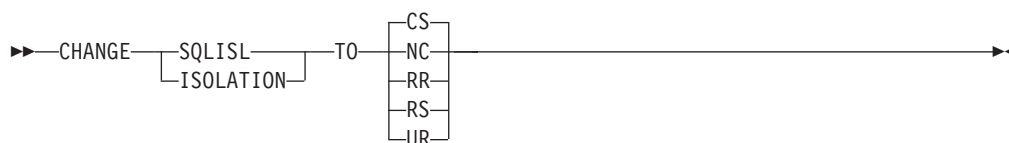
許可

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。

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 が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 が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 が既に宣言されて 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

このコマンドは、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、許可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・セットが CREATE DATABASE コマンドで指定されていない場合、許可される照合は次のとおりです。SYSTEM、IDENTITY_16BIT、UCA400_NO、UCA400_LSK、UCA400_LTH、*language-aware-collation*、および *locale-aware-collation* (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atalog・データベース・パーティションになります。

許可

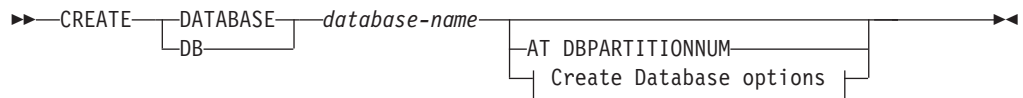
以下のいずれかが必要です。

- *sysadm*
- *sysctrl*

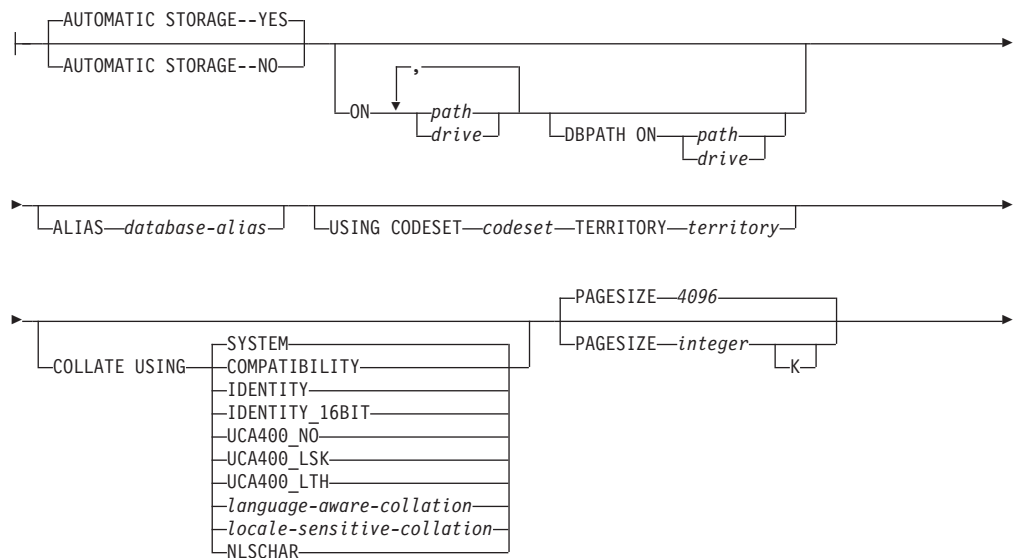
必要な接続

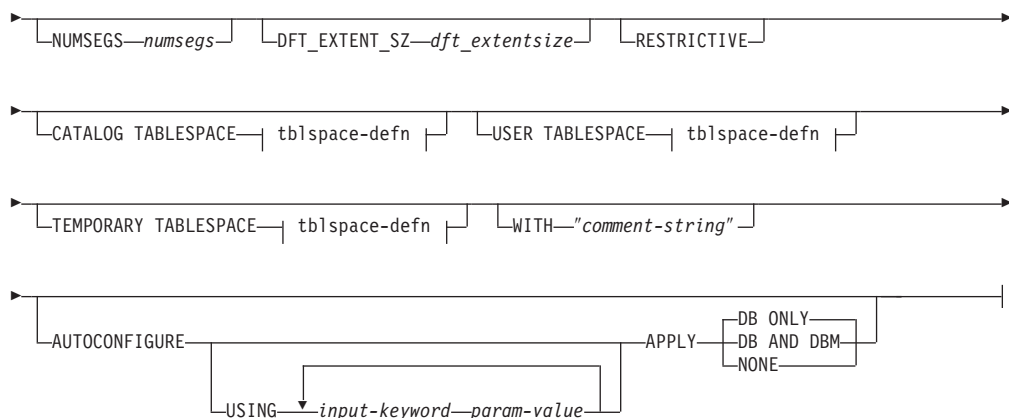
インスタンス。別の (リモート) データベース・パーティション・サーバーでデータベースを作成するには、まずそのサーバーにアタッチする必要があります。このコマンドの処理中、データベース接続が一時的に確立されます。

コマンド構文

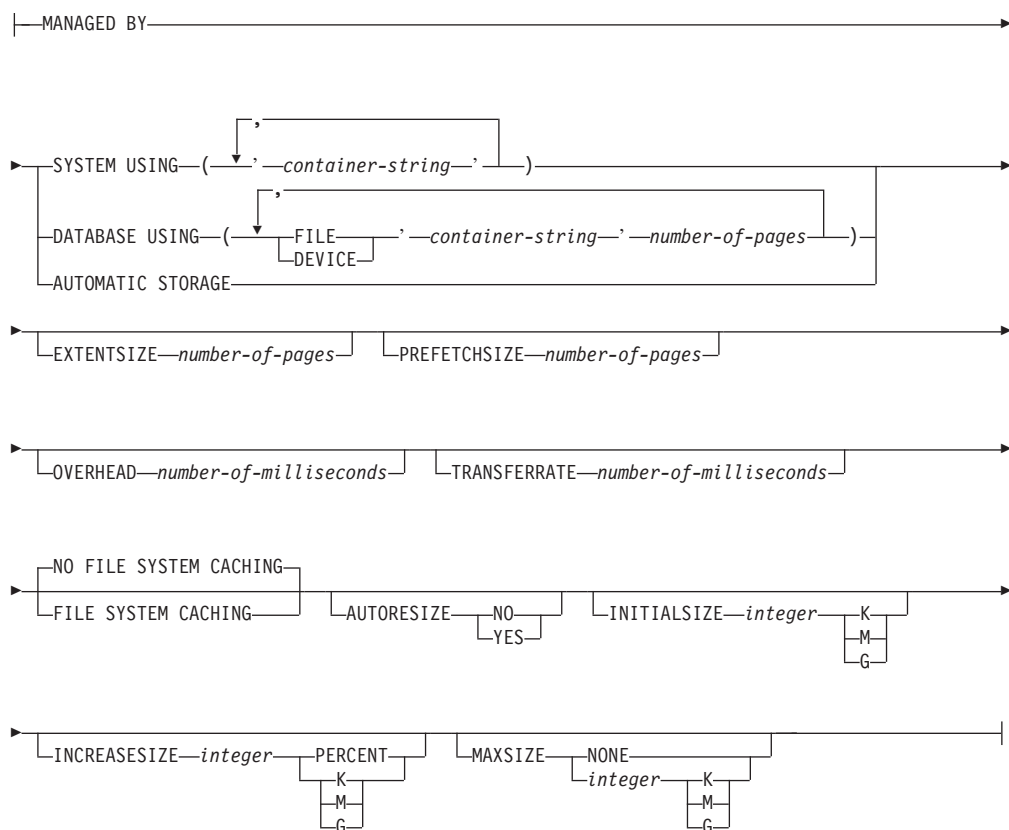


Create Database options:





tblspace-defn:



注:

1. 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。
2. コード・セットとテリトリ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 オプションには、*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、このオプション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 オプションを指定した 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。

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。

YES データベースの自動ストレージが有効です。

ON *path* または *drive*

このオプションの意味は、AUTOMATIC STORAGE オプション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** オプションにはただ 1 つのパスだけを含めることができ、このパスはデータベースの作成場所を指定します。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。この動作は、DB2® Universal Database™ バージョン 8.2 以前と同じです。
- 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。これらをストレージ・パスといい、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。

複数のパスを使用する場合、**DBPATH ON** オプションは、これらのパスのうちどこにデータベースを作成するかを指定します。**DBPATH ON** オプションを指定しない場合、データベースはリストの最初のパスに作成されます。パスを 1 つも指定しない場合、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・パスにデータベースが作成されます。また、これはデータベースに関連した単一ストレージ・パスの場所としても使用されます。

パスの最大長は 175 文字です。

MPP システムでは、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)。MPP システムでは、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** オプション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ない、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* または *drive*

自動ストレージが有効な場合、**DBPATH ON** オプションは、データベースの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** オプションが指定されていない場合、データベースは **ON** オプション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。このオプションは、それらの表の直前のバージョンを使用することを指定します。

IDENTITY

ストリングがバイト単位で比較される一致照合順序。Unicode データベースの場合は、これがデフォルトです。

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) で入手できます。このオプションは、Unicode データベースを作成する場合にのみ指定できます。

UCA400_NO

正規化された Unicode Standard バージョン 4.0.0 に基づく UCA (Unicode Collation Algorithm) 照合順序が、暗黙的に ON に設定されます。UCA の詳細は、Unicode Technical Standard #10 にあります。この仕様は Unicode Consortium Web サイト (www.unicode.org) で入手できます。このオプションは、Unicode データベースを作成する場合にのみ使用できます。

UCA400_LSK

Unicode Standard バージョン 4.0.0 に基づく UCA (Unicode Collation Algorithm) 照合順序。ただし、スロバキア語文字はすべて適切な順にソートされます。UCA の詳細は、Unicode Technical Standard #10 にあります。この仕様は Unicode Consortium Web サイト (www.unicode.org) で入手できます。このオプションは、Unicode データベースを作成する場合にのみ使用できます。

UCA400_LTH

Unicode Standard バージョン 4.0.0 に基づく UCA (Unicode

Collation Algorithm) 照合順序。ただし、タイ語文字はすべて Royal Thai Dictionary の順に配列されます。UCA の詳細は、Unicode Technical Standard #10 にあります。この仕様は Unicode Consortium Web サイト (www.unicode.org) で入手できます。このオプションは、Unicode データベースを作成する場合にのみ使用できます。この照合機能は、タイ語のデータを NLSCHAR 照合機能オプションとは異なる順序で配列することがあります。

language-aware-collation

このオプションは、Unicode データベースにのみ使用することができます。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 データの言語認識照合 (Language-aware collations for Unicode data)』を参照してください。

注: CREATE DATABASE コマンドがバージョン 9 サーバーに対して実行される場合、このオプションは使用できません。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。

locale-sensitive-collation

このオプションは、Unicode データベースにのみ使用することができます。詳細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システム定義の照合シーケンス。

このオプションは、タイ語コード・ページ (CP874) でのみ使用できます。このオプション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 が戻されます。

SYSTEM

これはデータベースを作成するときのデフォルト・オプションです。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、このオプションはクライアント・コード・セットおよびテリトリーに基づく言語認識照合にマップします。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・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 を使用しない場合の有効な整数の値は、4 096、

8 192、16 384、または 32 768 です。接尾部 K を使用する場合の有効な整数の値は、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 オプションを使用すると、**restrict_access** データベース構成パラメーターが YES に設定されることとなります。また、特権や権限が自動的に PUBLIC に認可されることがなくなります。**RESTRICTIVE** オプションを使用しないと、**restrict_access** データベース構成パラメーターは NO に設定され、以下のすべての特権が自動的に PUBLIC に認可されます。

- CREATETAB
- BINDADD
- CONNECT
- IMPLSCHEMA
- スキーマ SQLJ 内のすべてのプロシージャに対する EXECUTE WITH GRANT
- スキーマ SYSPROC 内のすべての関数およびプロシージャに対する EXECUTE WITH GRANT
- NULLID スキーマ内で作成されたすべてのパッケージに対する BIND
- NULLID スキーマ内で作成されたすべてのパッケージに対する EXECUTE
- スキーマ SQLJ に対する CREATEIN
- スキーマ NULLID に対する CREATEIN
- 表スペース USERSPACE1 に対する USE
- SYSIBM カタログ表に対する SELECT アクセス
- SYSCAT カタログ・ビューに対する SELECT アクセス
- SYSSTAT カタログ・ビューに対する SELECT アクセス
- SYSSTAT カタログ・ビューに対する UPDATE アクセス
- SYSDEFAULTUSERWORKLOAD ワークロードに対する USAGE

CATALOG TABLESPACE *tblspace-defn*

カタログ表 SYSCATSPACE を保持する、表スペース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には、自動ストレージ表スペースとして SYSCATSPACE 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 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には、ディレクトリーの **numsegs** 数をコンテナーとして使用し、**dft_extentsize** のエクステント・サイズを使用して、USERSPACE1 が 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には、ディレクトリーの **numsegs** 数をコンテナーとして使用し、**dft_extentsize** のエクステント・サイズを使用して、TEMPSPACE1 が 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/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ツ・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。

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。

`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 バイト以下です。

それぞれの *container-string* は絶対ディレクトリー名でも相対ディレクトリー名でも構いません。

ディレクトリー名が絶対ディレクトリー名でない場合、データベース・ディレクトリーに対して相対になり、そのデータベース・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に対するパス名別名 (UNIX システムではシンボリック・リンク) になることができます。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が 1 つも存在しない場合には、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・システムに依存します。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名はドライブ名とコロン (:) で始まり、UNIX 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** オプションを使用して無停電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。

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) およびそのサイズ (PAGESIZE ページ単位) 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 バイト以下です。

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* の正確な形式はオペレーティング・システムに依存します。

FILE *container-string number-of-pages*

FILE コンテナの場合、*container-string* ストリングは絶対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が 1 つも存在しない場合には、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。自動ストレージがデータベースに定義されていない場合、エラーが戻されます (SQLSTATE 55060)。

自動ストレージ表スペースは、システム管理スペース (SMS) 表スペースまたはデータベース管理スペース (DMS) 表スペースのいずれかとして作成されます。DMS が選択されたときに表スペースのタイプ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。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を判別します。

EXTENTSIZE *number-of-pages*

次のコンテナにスキップする前に、コンテナに書き込まれることになる **pagesize** ページの数を指定します。またエクステント・サイズの値は、整数値の後に **K** (キロバイト) または **M** (メガバイト) を付け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を実行する際に、表スペースから読み取られることになる **pagesize** ページの数を指定します。またプリフェッチ・サイズの値は、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) を付け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のオーバーヘッド、ディスク・シーク、および待ち時間を指定します。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) です。この値がすべてのコンテナーに同じでなければ、数値は表スペースに属するすべてのコンテナーの平均になるはずですが。

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オーバーヘッド、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 以上にマイグレーション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) です。この値がすべてのコンテナーに同じでなければ、数値は表スペースに属するすべてのコンテナーの平均になるはずですが。

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 以上にマイグレーション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・オプションです。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・オプションです。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 で、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。このオプションは自動ストレージ表スペースにのみ有効です。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、DB2 は十分なスペースになるまで **INCREASESIZE** によって表スペースの拡張を続けます。表スペースが自動サイズ変更可能であっても、**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。

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にサイズ増加単位 (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**。スペース要求が出された場合に表スペース・サイズの量をパーセントで指定します。PERCENT が指定された場合、整数値は 0 から 100 の間でなければなりません (SQLSTATE 42615)。
- **K** (キロバイト)、**M** (メガバイト)、または **G** (ギガバイト)。量をバイト単位で指定します。

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。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、バッファー・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 オプションが必要なのは、推奨内容を調整する場合のみです。

USING *input-keyword param-value*

表 7. 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	1-100	25	専用にするメモリーのパーセンテージ。他のアプリケーション (オペレーティング・システム以外) がこのサーバーで実行している場合、この値は 100 未満に設定してください。
workload_type	simple、 mixed、 complex	mixed	単純 (simple) ワークロードは入出力集約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 集約の傾向があり大部分が照会です。
num_stmts	1-1 000 000	25	作業単位ごとのステートメント数
tpm	1-200 000	60	1 分ごとのトランザクション
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	0-5 000	0	接続されたローカル・アプリケーションの数
num_remote_apps	0-5 000	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。Adaptive Self Tuning Memory および Auto Runstats が有効になり、**APPLY DB** または **APPLY DBM** オプションでの指示どおりに、データベース構成およびデータベース・マネージャー構成パラメーターが構成アドバイザーによって調整されます。
- あるデータベースに対して **CREATE DATABASE** コマンドで **AUTOCONFIGURE** オプションを指定すると、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
AUTOMATIC STORAGE YES
```

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。自動ストレージが有効になり、単一ストレージ・パスが使用されます (これもまた **dftdbpath** の値です)。

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。自動ストレージが暗黙的に有効になり、ストレージ・パスは **C:** および **D:** になります。

例 3:

```
CREATE DATABASE TESTDB15
AUTOMATIC STORAGE YES
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。自動ストレージが明示的に有効になり、ストレージ・パスは C: および D: になります。

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・マネージャーの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・パスが使用されます。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す。

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。

- 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タログ・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・マネージャーのバインド・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** オプションを選択しない場合、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 と呼ばれる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 つのファイルに関連した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 表スペースはカタログ・データベース・パーティションでのみ作成されます。
- 以下の権限や特権を付与します。
 - SYSFUN スキーマのすべての関数について、PUBLIC に対する EXECUTE WITH GRANT 特権
 - SYSIBM スキーマのすべてのプロシージャについて、PUBLIC に対する EXECUTE 特権
 - DBADM、CONNECT、CREATETAB、BINDADD、CREATE_NOT_FENCED、IMPLICIT_SCHEMA、および LOAD 権限をデータベース作成者に。
 - CONNECT、CREATETAB、BINDADD、および IMPLICIT_SCHEMA 権限を PUBLIC に。
 - USERSPACE1 表スペースの USE 特権を PUBLIC に。
 - 各システム・カタログに対する SELECT 特権を PUBLIC に。
 - 正常にバインドされたユーティリティーに対する BIND および EXECUTE 特権を PUBLIC に。
 - SYSFUN スキーマのすべての関数について、PUBLIC に対する EXECUTE WITH GRANT 特権。
 - SYSIBM スキーマのすべてのプロシージャについて、PUBLIC に対する EXECUTE 特権。

注: **RESTRICTIVE** オプションを使用すると、**restrict_access** データベース構成パラメーターが YES に設定されることとなります。また、特権や権限が自動的に PUBLIC に認可されることがなくなります。追加情報については、CREATE DATABASE コマンドの **RESTRICTIVE** オプション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、データベース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バ

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、**AUTOMATIC STORAGE NO** オプションを指定すれば、自動ストレージを無効にすることができます。

注意の必要な重要な点として、自動ストレージを有効にできるのはデータベースの作成時のみであり、データベースの作成後は有効にできません。また、自動ストレージを使用するようデータベースをいったん定義した後で無効にすることはできません。

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・マネージャーは、ストレージ・パス内に以下のディレクトリーまたはマウント・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・マネージャーは、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・ディレクトリー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したストレージ・パスです。
- *instance_name* は、データベースの配置先のインスタンスの名前です。
- *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、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・マネージャーは、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####* ディレクトリーのものを取り出すことを認識します。

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。

データベース・パーティション式を指示するには、引数 " \$N" ([blank]\$N) を使用します。データベース・パーティション式は、ストレージ・パス内のどこでも使用することができ、複数のデータベース・パーティション式を指定してもかまいません。データベース・パーティション式は、スペース文字で終了します。スペースの後に続くものはすべて、データベース・パーティション式の評価後にストレージ・パスに追加されます。データベース・パーティション式の後、ストレージ・パス内にスペース文字がない場合、そのストリングの残りは式の一部であるとみなされます。引数は、以下の形式のいずれかでのみ使用できます。

演算子は、左から右へ評価されます。% は、モジュラス演算子を表します。例中のデータベース・パーティション番号は 10 と想定されています。

構文	例	値
[blank]\$N	" \$N"	10
[blank]\$N+[number]	" \$N+100"	110
[blank]\$N%[number]	" \$N%5"	0
[blank]\$N+[number]%[number]	" \$N+1%5"	1
[blank]\$N%[number]+[number]	" \$N%4+2"	4
^a % はモジュラスです。		

dbadm 権限を使用すると、これらの特権を他のユーザーまたは PUBLIC に付与 (または取り消し) することができます。データベース上で *sysadm* または *dbadm* 権限を付与された別の管理者が上記の特権を取り消したとしても、データベース作成者はそれらの特権を保持します。

MPP 環境では、データベース・マネージャー が、すべてのデータベース・パーティションの指定したパスまたはデフォルト・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す。

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・モニターにも関連したオーバーヘ

ッドがあります。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 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。

互換性

バージョン 8 より前のバージョンとの互換性：

- キーワード **NODE** は、**DBPARTITIONNUM** の代わりとして使用できます。

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。データベースはローカルでなければなりません。

ツール・カタログには、タスク・センターおよびコントロール・センターなどのツールを使って構成する、管理タスクについての情報が入っています。

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、スケジューラーを更新します。

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の下に作成されます。

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・ディレクトリまたはシステム・データベース・ディレクトリ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 foobar 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) が必要です。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 で *dasadm* 権限が必要です。
- 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。
- マルチ・パーティション環境では、データベース・パーティション・フィーチャー (DPF) が使用可能に設定されているとき、32KB REGULAR 表スペースがカタログ・パーティション上に存在する必要があります。それが存在しない場合、(以下に示すような) コマンドは表スペースが指定されたときに失敗します。

```
db2 create tools catalog foobar use existing tablespace user32Ksp
in database toolsdb
```

DEACTIVATE DATABASE

指定したデータベースを停止させます。

有効範囲

MPP システムの場合、このコマンドはシステム内のすべてのデータベース・パーティションで、指定したデータベースを非活動化します。1 つ以上のデータベース・パーティションでエラーが検出されると、警告が戻されます。データベースが正常に非アクティブ化されるデータベース・パーティションもありますが、エラーが検出されたノードではデータベースはそのままアクティブを継続することがあります。

許可

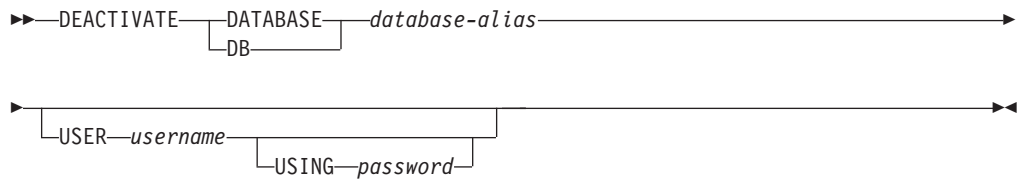
以下のいずれか。

- *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 *database-alias*

停止するデータベースの別名を指定します。

USER *username*

データベースを停止するユーザーを指定します。

USING *password*

ユーザー ID のパスワードを指定します。

使用上の注意

ACTIVATE DATABASE で初期化したデータベースは、DEACTIVATE DATABASE または db2stop によってシャットダウンすることができます。ACTIVATE DATABASE でデータベースを初期化した場合、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 を使用する必要があります。(この場合、db2stop を使用してデータベースをシャットダウンすることもできます。)

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。

DECOMPOSE XML DOCUMENT

このコマンドはストアド・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。

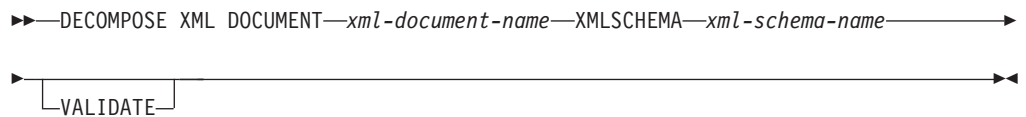
- 以下の特権すべて:

- 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)
- SELECT、INSERT、UPDATE、または DELETE 特権 (db2-xdb:expression または db2-xdb:condition アノテーションで参照される表に対して必要)
- VALIDATE オプションが指定されている場合、XML スキーマに対する USAGE 特権
- 以下の特権または権限のいずれか:
 - ターゲット表に対する CONTROL 特権
 - *sysadm* または *dbadm* の権限

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

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。

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"
```

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。

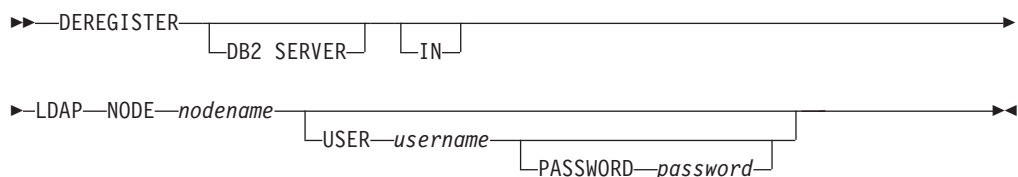
許可

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・ユーザーの信用証明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
す。

DESCRIBE

このコマンドは以下の事柄を行います。

- SELECT、CALL、または XQuery ステートメントに関する出力情報の表示
- 表またはビューの列の表示
- 表またはビューの索引の表示
- 表またはビューのデータ・パーティションの表示

許可

SELECT ステートメントに対する出力情報を表示するには、SELECT ステートメ
ント中で参照された表またはビューごとに、以下にリストされた特権または権限の
1 つが必要になります。

表またはビューの列、索引またはデータ・パーティションを表示するには、以下の
システム・カタログに関する SELECT 特権、CONTROL 特権、*sysadm* 権限、また
は *dbadm* 権限が必要です。

- SYSCAT.COLUMNS (DESCRIBE TABLE), SYSCAT.DATAPARTITIONEXPRESSION
(with SHOW DETAIL)
- GET_INDEX_COLNAMES() UDF に対する SYSCAT.INDEXES (DESCRIBE
INDEXES FOR TABLE) 実行特権 (および SHOWDETAIL)
- SYSCAT.DATAPARTITIONS (DESCRIBE DATA PARTITIONS FOR TABLE)

PUBLIC に、宣言されたグローバル一時表に対するすべての特権が付与されている
なら、ユーザーは、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され
たグローバル一時表に関する情報を表示できます。

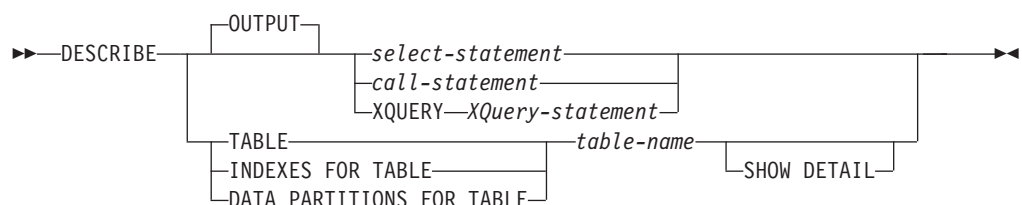
CALL ステートメントに関する出力情報を表示するには、下にリストされている特
権または権限のいずれかが必要です。

- ストアード・プロシージャでの EXECUTE 特権。
- *sysadm* または *dbadm* の権限

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ステートメントは、記述された照会の最終結果表の SELECT リストの一部として列が明示的に指定されている場合に、暗黙的に非表示にされている列に関する情報のみ戻します。

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。暗黙的に非表示にされている列に関する情報が戻されません。

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 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。

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
- 列カウント

DESCRIBE INDEXES FOR TABLE コマンドの場合、索引名が 18 バイトを超えると、切り捨てられます。出力には以下の追加情報が含まれます。

- 列名

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
- 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、元の出力の下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、元の出力の下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。

```
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ード・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型に 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型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 オプションのサポートされていないダウン・レベル・サーバーに対して DESCRIBE XQUERY コマンドが発行された場合、その機能がダウン・レベル・サーバーで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	Type schema	Type name	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下に表示されます。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	Type schema	Column number	Type name	Length
FIRST	SYSIBM	0	CHARACTER	10
LAST	SYSIBM	1	CHARACTER	10

Table is partitioned by range (ordered on the following column/s):

```
-----
LAST
FIRST
```

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。

```
db2 describe indexes for table user1.department
```

Table: USER1.DEPARTMENT

Index schema	Index name	Unique rule	Number of columns
USER1	IDX1	U	2

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	ObjectId	AccessMode	Status
0	PARTx	3	43	F		
1	PARTNew	13	13	N	A	
2	PART3	31	33	F		
3	PART4	23	34	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にない場合、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) **contact_host** 構成パラメーターの設定は、リストがローカルかグローバルかを判別します。

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) **contact_host** 構成パラメーターの設定は、リストがローカルかグローバルかを判別します。

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 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 DB2 ESE に付属のユーティリティーが使用するもので、汎用ではありません。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/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管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。

互換性

バージョン 8 より前のバージョンとの互換性 :

- キーワード NODE は、DBPARTITIONNUM の代わりとして使用できます。

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

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。

互換性

バージョン 8 より前のバージョンとの互換性：

- キーワード `NODE` は、`DBPARTITIONNUM` の代わりとして使用できます。

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。スケジューラーを活動化するには、前のツール・カタログを活動化するか、新しいツール・カタログを作成する必要があります。

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管され、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。

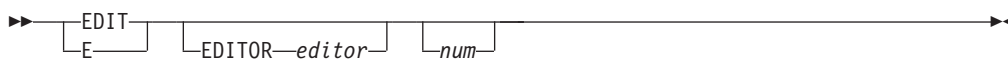
許可

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。

141 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

以下のいずれか。

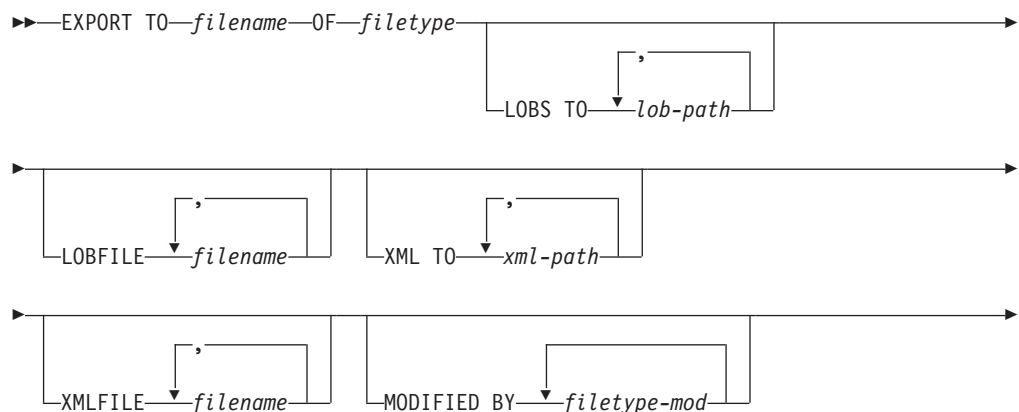
- `sysadm`
- `dbadm`

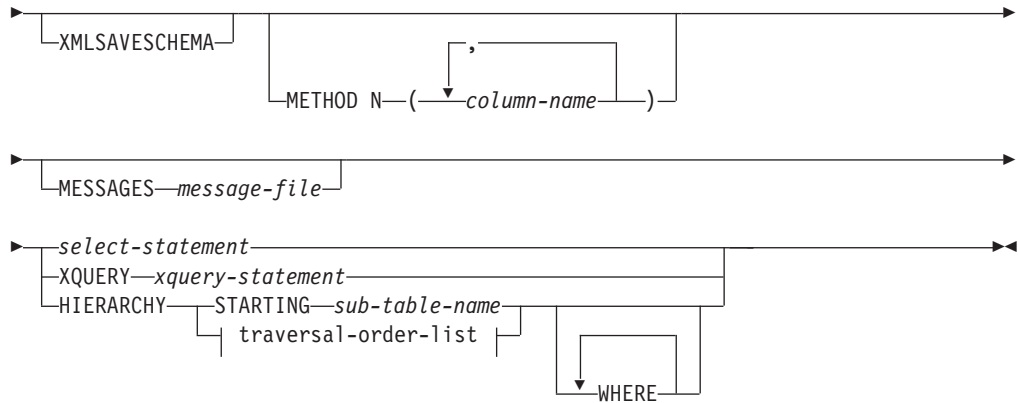
または、関係するそれぞれの表またはビューに対する `CONTROL` または `SELECT` 特権

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

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。

HIERARCHY STARTING sub-table-name

デフォルトの走査順序 (ASC、DEL、または WSF 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。

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。最初の名前のネーム・スペース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 動作が暗黙的に活動化されます。

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。

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 動作が暗黙的に活動化されます。

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。

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 WSF および 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。

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。 141 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。

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。
- WSF (ワークシート・フォーマット)。以下のプログラムで使用します。
 - Lotus® 1-2-3®
 - Lotus Symphony

BIGINT または DECIMAL データをエクスポートする場合、タイプ DOUBLE の範囲内の値のみが正確にエクスポートされます。この範囲内にはない値もエクスポートされますが、オペレーティング・システムによっては、これらの値のインポートまたはエクスポートの結果、データに間違いが生じる場合があります。

- 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。

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。

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。情報の追加は行いません。

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。最初の名前のネーム・スペース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。

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。

XMLSAVESCHEMA

すべての XML 列について XML スキーマ情報を保管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。

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 chardel'' 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、すべてのロックがベ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。
- 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。
- DEL フォーマット・ファイルへエクスポートするために 254 よりも長い文字データの列が選択されると、エクスポート・ユーティリティーは警告メッセージを生成します。
- 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 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に保管されます。後者を除く各ファイルでは、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 (または要約して "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。このリストが指定されていないと、階層のすべての表がエクスポートされ、OUTER 順序がデフォルトの順序になります。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。
- インポート操作時には、同じ全探索順序を使用してください。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 クレデンシャルのために、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 クレデンシャルのために、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。
- 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。

```
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 8. 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 <code>db2exp.001.123.456/</code> がデータ・ファイルに保管される場合、LOB はファイル <code>db2exp.001</code> のオフセット 123 に位置し、456 バイト長です。</p> <p>EXPORT の使用時に『lobsinfile』修飾子を指定した場合、LOB データは LOBS TO 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。LOBS TO 節は、LOB ファイルが保管されるディレクトリーに、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。LOBS TO または LOBFILE オプションによって、LOBSINFILE 動作が暗黙的に活動化されます。</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 <code>db2exp.001.7.-1/</code> です。</p>
xmlinsefiles	それぞれの XQuery Data Model (XDM) インスタンスは、別個のファイルに書き込まれます。デフォルトで、複数の値は同一ファイル内で連結されます。
lobsinsefiles	それぞれの LOB 値は、別個のファイルに書き込まれます。デフォルトで、複数の値は同一ファイル内で連結されます。
xmlnodeclaration	XDM インスタンスは、XML 宣言タグなしで書き込まれます。デフォルトで、XDM インスタンスはエンコード属性を含む XML 宣言タグを先頭に付けてエクスポートされます。
xmlchar	XDM インスタンスは、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 <i>codepage</i> 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、XDM インスタンスは Unicode で書き出されます。
xmlgraphic	<i>xmlgraphic</i> 修飾子が EXPORT コマンドで指定されている場合、エクスポートされた XML 文書は、アプリケーション・コード・ページまたは <i>codepage</i> ファイル・タイプ修飾子にかかわらず、UTF-16 コード・ページでエンコードされます。

表9. 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: 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。² 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、以下のように文字ストリングの区切り文字として指定できません。</p> <pre>modified by chardel''</pre>
codepage=x	<p>x は ASCII 文字ストリングです。この値は、出力データ・セット内のデータのコード・ページと解釈されます。エクスポート操作中に、文字データをアプリケーション・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。codepage 修飾子を lobsinfile 修飾子と一緒に使うことはできません。</p>
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。²</p> <p>以下の例では、coldel; によって、エクスポート・ユーティリティーがセミコロン文字 (;) 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	<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。²</p>
nochardel	<p>列データは区切り文字で囲まれません。データを DB2 を使用してインポートまたはロードするつもりの場合は、このオプションを指定しないでください。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 chardelx 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。</p>
nodoubledel	<p>二重になっている区切り文字の認識を抑制します。²</p>

表 9. 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: 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
striplzeros	<p>エクスポートされるすべての DECIMAL 列から先行ゼロを除去します。</p> <p>次のような例を考察してみます。</p> <pre data-bbox="602 390 1230 569">db2 create table decimalTable (c1 decimal(31, 2)) db2 insert into decimalTable values (1.1) db2 export to data of del select * from decimalTable db2 export to data of del modified by STRIPLZEROS select * from decimalTable</pre> <p>最初のエクスポート操作では、エクスポート・ファイル・データの内容は +000000000000000000000000000000000001.10 となります。 2 番目の操作は、striplzeros 修飾子以外は最初の操作と同一ですが、エクスポート・ファイル・データの内容は +1.10 となります。</p>

表9. 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: 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
timestampformat="x"	<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。⁴ 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) 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 および MMM とは相互に排他的) 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) D - 日 (1 から 31 の範囲の 1 桁または 2 桁の数) DD - 日 (1 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。 他の日または月のエレメントとは相互に排他的) 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。) HH - 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 2 桁の数。 H と相互に排他的) M - 分 (0 から 59 の範囲の 1 桁または 2 桁の数) MM - 分 (0 から 59 の範囲の 2 桁の数。 M (分) とは相互に排他的) S - 秒 (0 から 59 の範囲の 1 桁または 2 桁の数) SS - 秒 (0 から 59 の範囲の 2 桁の数。 S と相互に排他的) SSSSS - 夜中の 12 時から数えた秒数 (00000 から 86399 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) UUUUUU - マイクロ秒 (000000 から 999999 の範囲の 6 桁の数。 他のマイクロ秒エレメントとは相互に排他的) UUUUU - マイクロ秒 (00000 から 99999 の範囲の 5 桁の数。 000000 から 999990 の範囲にマップされる。 他のマイクロ秒エレメントとは相互に排他的) UUUU - マイクロ秒 (0000 から 9999 の範囲の 4 桁の数。 000000 から 999900 の範囲にマップされる。 他のマイクロ秒エレメントとは相互に排他的) UUU - マイクロ秒 (000 から 999 の範囲の 3 桁の数。 000000 から 999000 の範囲にマップされる。 他のマイクロ秒エレメントとは相互に排他的) UU - マイクロ秒 (00 から 99 の範囲の 2 桁の数。 000000 から 990000 の範囲にマップされる。 他のマイクロ秒エレメントとは相互に排他的) U - マイクロ秒 (0 から 9 の範囲の 1 桁の数。 000000 から 900000 の範囲にマップされる。 他のマイクロ秒エレメントとは相互に排他的) TT - 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>"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 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: 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・セット内のデータのコード・ページと解釈されます。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。codepage 修飾子を lobsinfile 修飾子と一緒に使うことはできません。</p>

表 11. 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: WSF ファイル・フォーマット

修飾子	説明
1	Lotus 1-2-3 リリース 1、または Lotus 1-2-3 リリース 1a との互換がある WSF ファイルを作成します。 ⁵ この値がデフォルトです。
2	Lotus Symphony リリース 1.0 と互換性のある WSF ファイルを作成します。 ⁵
3	Lotus 1-2-3 パージョン 2、または Lotus Symphony リリース 1.1 との互換がある WSF ファイルを作成します。 ⁵
4	DBCS 文字から成る WSF ファイルを作成します。

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・ユーティリティー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・ユーティリティーでの記述フォーマットは次のとおりです。
 - 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.uuuuuu* 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。以下に、いくつかのあいまいなタイム・スタンプ・フォーマットを示します。
 - "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))
```

5. *filetype-mod* パラメーター・ストリングの中で、Lotus 1-2-3 の場合は L、Symphony の場合は S を指定すれば、これらのファイルを特定の製品に送ることができます。指定できるのは、1 つの値または製品指定子だけです。
6. WSF ファイル形式は、XML 列ではサポートされません。
7. 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。
8. XMLNODEDECLARATION ファイル・タイプ修飾子が指定されていない場合、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。
9. 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、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。
10. 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。

11. エクスポート・ユーティリティーは、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
```


から開始します。999 番目の LOB ファイルまたは XML ファイルの後には、ID に 0 が埋め込まれることはなくなります (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 の後にはデータ・タイプを表す 3 文字のタイプ ID があり、それは

.lob

または

.xml

のどちらかです。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。

12. XQuery を指定すると、エクスポート・ユーティリティーは、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/sql/lib/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・データベース環境内の任意のノード (データベース・パーティション・サーバー) から発行することができます。

許可

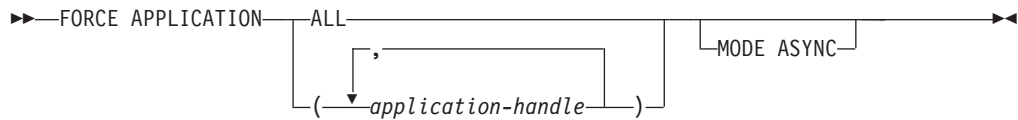
以下のいずれか。

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

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。

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せん。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) 構成パラメーターの値を戻します。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 構成パラメーターのリストについては、UPDATE ADMIN CONFIGURATION コマンドの説明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 構成パラメーターに関する情報を戻します。

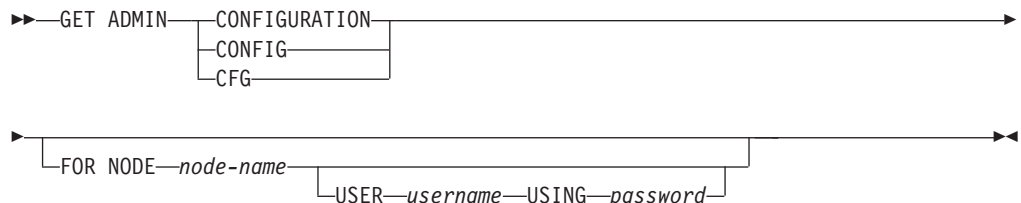
許可

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 構成パラメーター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

```

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。

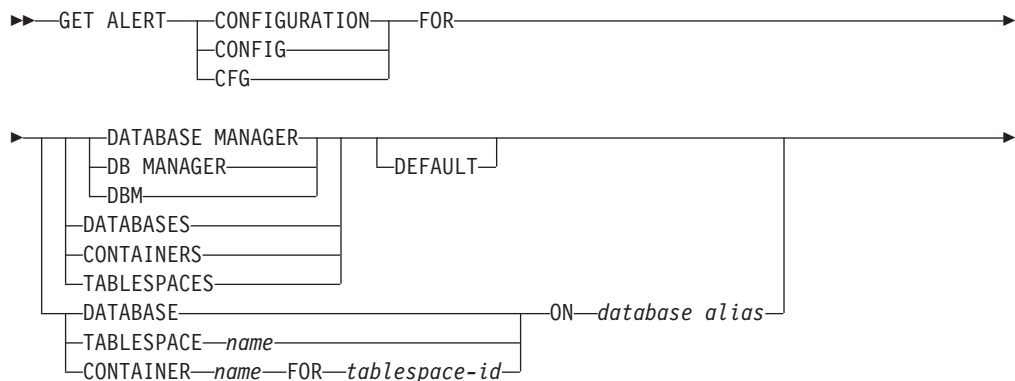
許可

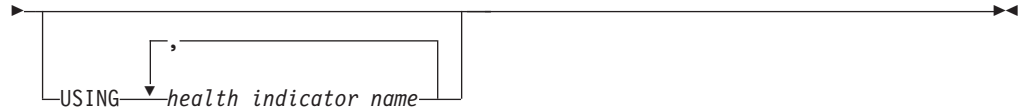
なし

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・インディケータのセットを指定します。ヘルス・インディケータ名は 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・インディケータが戻されます。

例

以下に示すのは、データベース・マネージャ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
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/sheaphres_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
Formula	= ((db.max_shr_sort_mem/ sheaphres_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
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

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)
                                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 AUTHORIZATIONS

データベース構成ファイルおよび許可システム・カタログ・ビュー (SYSCAT.DBAUTH) 内で検出した値から現行ユーザーの権限を取得します。

注: このコマンドは、推奨されていません。許可 ID に関する権限情報を検索するための新しい方法があります。それには、AUTH_LIST_AUTHORITIES_FOR_AUTHID 表関数を使用します。

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

▶▶ GET AUTHORIZATIONS ◀◀

コマンド・パラメーター

なし

例

以下に示すのは、GET AUTHORIZATIONS の出力例です。

```
Administrative Authorizations for Current User

Direct SYSADM authority           = NO
Direct SYSCtrl authority          = NO
Direct SYSMAINT authority         = NO
Direct DBADM authority            = YES
Direct CREATETAB authority        = YES
Direct BINDADD authority          = YES
Direct CONNECT authority          = YES
Direct CREATE_NOT_FENC authority  = YES
Direct IMPLICIT_SCHEMA authority  = YES
Direct LOAD authority             = YES
Direct QUIESCE_CONNECT authority  = YES
Direct CREATE_EXTERNAL_ROUTINE   = YES

Indirect SYSADM authority         = YES
Indirect SYSCtrl authority        = NO
Indirect SYSMAINT authority       = NO
Indirect DBADM authority          = NO
Indirect CREATETAB authority      = YES
Indirect BINDADD authority        = YES
Indirect CONNECT authority        = YES
Indirect CREATE_NOT_FENC authority = NO
```

```

Indirect IMPLICIT_SCHEMA authority      = YES
Indirect LOAD authority                  = NO
Indirect QUIESCE_CONNECT authority      = NO
Indirect CREATE_EXTERNAL_ROUTINE authority = NO

```

使用上の注意

- 権限をユーザー ID に付与する明示的なコマンドによって獲得される権限のことを直接権限といいます。それに対し、間接権限とは、ユーザーが所属するグループによって獲得された権限を基盤としている権限のことをいいます。(すべてのユーザーは、PUBLIC という名前の特殊グループに属します。)
- GET AUTHORIZATIONS コマンドは、現行ユーザーが SECADM 権限を保持しているかどうかを表示しません。SECADM 権限を保持するユーザーを調べるには、以下の照会を使用します。

```

SELECT GRANTEE FROM SYSCAT.DBAUTH
WHERE SECURITYADMAUTH = 'Y'

```

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、DB2 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 構成パラメーターを、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、構成アシスタント (CA) または CLI/ODBC 構成ユーティリティーのどちらか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
```

説明

```
-----  
Foo Widgets broadloom support unit
```

Name	Type
-----	-----
joe	contact
support	contact group
joline	contact

GET CONTACTGROUPS

このコマンドは、システムでローカルに定義されるか、またはグローバル・リストで定義される、連絡先グループのリストを提供します。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) **contact_host** 構成パラメーターの設定は、リストがローカルかグローバルかを判別します。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戻します。

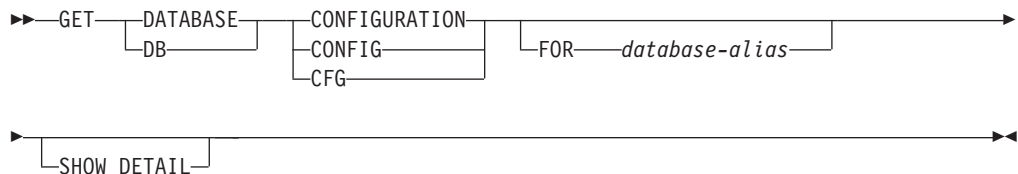
許可

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活動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。

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。
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。

以下に示すのは、GET DATABASE CONFIGURATION の出力例です (Windows で発行)。

Database Configuration for Database

```

Database configuration release level      = 0x0c00
Database release level                   = 0x0c00

Database territory                        = US
Database code page                       = 1252
Database code set                        = IBM-1252
Database country/region code            = 1
Database collating sequence              = UNIQUE
Alternate collating sequence              (ALT_COLLATE) =
Database page size                       = 8192

Dynamic SQL Query management              (DYN_QUERY_MGMT) = DISABLE

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|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
Size of database shared memory (4KB)      (DATABASE_MEMORY) = AUTOMATIC
Database memory threshold                 (DB_MEM_THRESH) = 10
Max storage for lock list (4KB)           (LOCKLIST) = 50
Percent. of lock lists per application    (MAXLOCKS) = 22
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
Catalog cache size (4KB)                  (CATALOGCACHE_SZ) = (MAXAPPLS*4)
Log buffer size (4KB)                     (LOGBUFSZ) = 8
Utilities heap size (4KB)                  (UTIL_HEAP_SZ) = 5000
Buffer pool size (pages)                  (BUFFPAGE) = 250
SQL statement heap (4KB)                  (STMHEAP) = AUTOMATIC
Default application heap (4KB)            (APPLHEAPSZ) = AUTOMATIC
Application Memory Size (4KB)             (APPL_MEMORY) = AUTOMATIC
Statistics heap size (4KB)                (STAT_HEAP_SZ) = AUTOMATIC

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
Number of I/O servers                     (NUM_IOSERVERS) = AUTOMATIC
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
Average number of active applications (AVG_APPLS) = AUTOMATIC
Max DB files open per application      (MAXFILOP) = 32768

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                      = D:¥DB2¥NODE0000¥SQL00001¥SQLLOGDIR¥
Overflow log path                      (OVERFLOWLOGPATH) =
Mirror log path                        (MIRRORLOGPATH) =
First active log file                  =
Block log on disk full                 (BLK_LOG_DSK_FUL) = 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) = 100
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

First log archive method               (LOGARCHMETH1) = OFF
Options for logarchmeth1              (LOGARCHOPT1) =
Second log archive method              (LOGARCHMETH2) = 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 statistics profiling        (AUTO_STATS_PROF) = OFF
  Automatic profile updates            (AUTO_PROF_UPD) = OFF
  Automatic reorganization             (AUTO_REORG) = OFF

```

Enable XML Character operations (ENABLE_XMLCHAR) = YES
WLM Collection Interval (WLM_COLLECT_INT) = 0

以下の例は、SHOW DETAIL オプションを指定した場合のコマンド出力の一部を示しています。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		= 0x0c00	
Database release level		= 0x0c00	
Database territory		= US	
Database code page		= 1252	
Database code set		= IBM-1252	
Database country/region code		= 1	
Database collating sequence		= UNIQUE	UNIQUE
Alternate collating sequence	(ALT_COLLATE)	=	
Database page size		= 8192	8192
Dynamic SQL Query management	(DYN_QUERY_MGMT)	= DISABLE	DISABLE
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	
Database is consistent		= 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(18096)	AUTOMATIC(18096)
Database memory threshold	(DB_MEM_THRESH)	= 10	10
Max storage for lock list (4KB)	(LOCKLIST)	= 50	50
Percent. of lock lists per application	(MAXLOCKS)	= 22	22
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(600)	AUTOMATIC(600)
Catalog cache size (4KB)	(CATALOGCACHE_SZ)	= (MAXAPPLS*4)	(MAXAPPLS*4)
Log buffer size (4KB)	(LOGBUFSZ)	= 8	8
Utilities heap size (4KB)	(UTIL_HEAP_SZ)	= 5000	5000
Buffer pool size (pages)	(BUFFPAGE)	= 250	250
SQL statement heap (4KB)	(STMTHEAP)	= AUTOMATIC(2048)	AUTOMATIC(2048)
Default application heap (4KB)	(APPLHEAPSZ)	= AUTOMATIC(256)	AUTOMATIC(256)
Application Memory Size (4KB)	(APPL_MEMORY)	= AUTOMATIC(496)	AUTOMATIC(10000)

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(1)	AUTOMATIC(1)
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) = 32768	32768
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 ¥SQL00001¥SQLOGDIR¥	D:¥DB2¥NODE0000 ¥SQL00001¥SQLOGDIR¥
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 ckckpt	(SOFTMAX) = 100	100
Log retain for recovery enabled	(LOGRETAIN) = OFF	OFF
User exit for logging enabled	(USEREXIT) = OFF	OFF
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
First log archive method	(LOGARCHMETH1) = OFF	OFF
Options for logarchmeth1	(LOGARCHOPT1) =	
Second log archive method	(LOGARCHMETH2) = 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
Automatic database backup	(AUTO_DB_BACKUP) =	OFF
Automatic table maintenance	(AUTO_TBL_MAINT) =	ON
Automatic runstats	(AUTO_RUNSTATS) =	ON
Automatic statistics profiling	(AUTO_STATS_PROF) =	OFF
Automatic profile updates	(AUTO_PROF_UPD) =	OFF
Automatic reorganization	(AUTO_REORG) =	OFF
Enable XML Character operations	(ENABLE_XMLCHAR) =	ON
WLM Collection Interval	(WLM_COLLECT_INT) =	0

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。

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 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ことが必要です。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。

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。

以下に示すのは、GET DATABASE MANAGER CONFIGURATION の出力例です (Windows で発行)。

```
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)   =

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
Max time difference between nodes (min) (MAX_TIME_DIFF) = 60

db2start/db2stop timeout (min)        (START_STOP_TIME) = 10

```

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・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)	=	
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 について指定されている AUTOMATIC 値は、インスタンス起動時の初期値であり、実行時に発生する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。

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

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・システム・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。モニター中の任意のアプリケーションがオンの場合、データベース・マネージャー・レベル・スイッチがオンになります。モニター中の任意のアプリケーション用に、現在データベース・システム・モニターがデータを収集しているかどうかを判別するために、このコマンドを使用します。

許可

以下のいずれか。

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

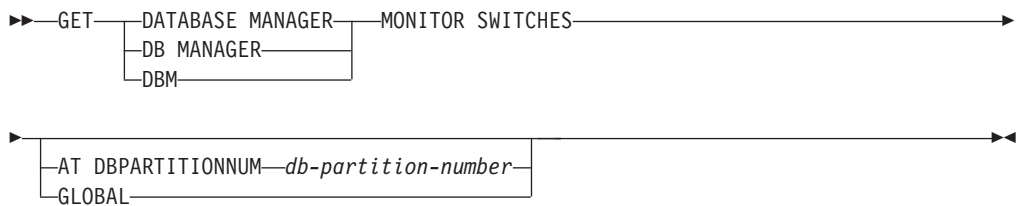
必要な接続

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、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 DBPARTITIONNUM *db-partition-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データベース・パーティションを指定します。

GLOBAL

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・パーティションの集約結果を戻します。

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 db partition number 1
Buffer Pool Activity Information (BUFFERPOOL) = ON 06-11-2003 10:11:01.738377

Lock Information	(LOCK) = OFF	
Sorting Information	(SORT) = ON	06-11-2003 10:11:01.738400
SQL Statement Information	(STATEMENT) = OFF	
Table Activity Information	(TABLE) = OFF	
Take Timestamp Information	(TIMESTAMP) = ON	06-11-2003 10:11:01.738525
Unit of Work Information	(UOW) = ON	06-11-2003 10:11:01.738353

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
- 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になります。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

バージョン 8 より前のバージョンとの互換性：

- キーワード NODE は、DBPARTITIONNUM の代わりとして使用できます。

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 ◀◀
    └── NOTIFICATION ───┘
```

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。

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。デフォルトでは、これは呼び出し元データベース・パーティションで活動します。GLOBAL オプションを使用する場合、すべてのデータベース・パーティションから統合された情報が抽出されます。

許可

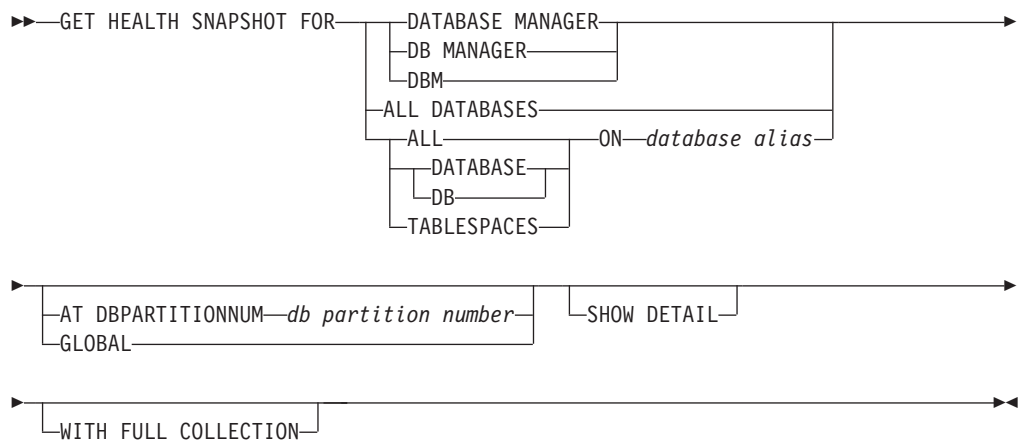
なし

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必要があります。

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・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 DBPARTITIONNUM *db-partition-number*

指定されたデータベース・パーティションの結果を戻します。

GLOBAL

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・パーティションの集約結果を戻します。

SHOW DETAIL

出力の中にそれぞれのヘルス・モニター・データ・エレメントごとの履歴データが (Timestamp, Value, Formula) という形式で含まれるように指定します。大括弧で囲まれたパラメーター {(Timestamp, Value, Formula)} は、戻されるそれぞれの履歴データ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討します。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 database partitions 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
```

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・システム・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。

許可

以下のいずれか。

- *sysadm*
- *sysctrl*
- *sysmaint*
- *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 DBPARTITIONNUM db-partition-number] GLOBAL ▶▶
```

コマンド・パラメーター

AT DBPARTITIONNUM *db-partition-number*

モニター・スイッチの状況を表示するデータベース・パーティションを指定します。

GLOBAL

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・パーティションの集約結果を戻します。

例

以下に示すのは、GET MONITOR SWITCHES の出力例です。

Monitor Recording Switches

```
Switch list for db partition number 1
Buffer Pool Activity Information (BUFFERPOOL) = ON 02-20-2003 16:04:30.070073
Lock Information (LOCK) = OFF
Sorting Information (SORT) = OFF
SQL Statement Information (STATEMENT) = ON 02-20-2003 16:04:30.070073
Table Activity Information (TABLE) = OFF
Take Timestamp Information (TIMESTAMP) = ON 02-20-2003 16:04:30.070073
Unit of Work Information (UOW) = ON 02-20-2003 16:04:30.070073
```

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`
- `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

バージョン 8 より前のバージョンとの互換性：

- キーワード NODE は、DBPARTITIONNUM の代わりとして使用できます。

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、推奨事項の記述を取得します。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。

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。GLOBAL パラメーターを指定していない限り、このコマンドは、そのデータベース・パーティション上でのみ機能します。

許可

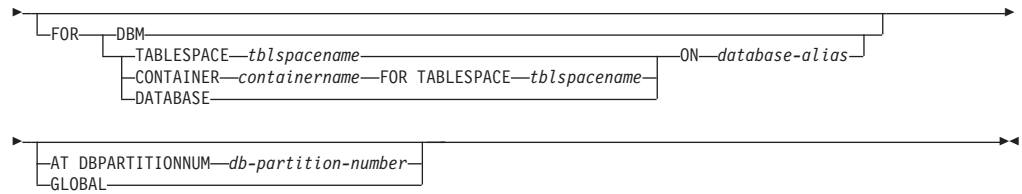
なし

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*—▶



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 DBPARTITIONNUM

ヘルス・インディケーターがアラート状態に入ったデータベース・パーティション番号を指定します。データベース・パーティション番号を指定せず、GLOBAL も指定しない場合、このコマンドは、現在接続しているデータベース・パーティションの情報を戻します。

GLOBAL

すべてのデータベース・パーティションで、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。別々のデータベース・パーティションに関する推奨事項が同じ場合は、それらの推奨事項が、影響を受けるデータベース・パーティションに関するヘルス・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:

Recommendation: Increase the database heap size.

Rank: 1

Increase the database configuration parameter dbheap sufficiently to move utilization to normal operating levels. To increase the value, set the new value of dbheap to be equal to $(\text{pool_cur_size} / (4096 * U))$ where U is the desired utilization rate. For example, if your desired utilization rate is 60% of the warning threshold level, which you have set at 75%, then $U = 0.6 * 0.75 = 0.45$ (or 45%).

Take one of the following actions:

Execute the following scripts at the DB2 server:

```
CONNECT TO SAMPLE;  
UPDATE DB CFG USING DBHEAP 149333;  
CONNECT RESET;
```

Launch DB2 tool: Database Configuration Window

The Database Configuration window can be used to view and update database configuration parameters.

To open the Database Configuration window:

1. From the Control Center, expand the object tree until you find the databases folder.
2. Click the databases folder. Any existing database are displayed in the contents pane on the right side of the window.
3. Right-click the database that you want in the contents pane, and click Configure Parameters in the pop-up menu. The Database Configuration window opens.

On the Performance tab, update the database heap size parameter as suggested and click OK to apply the update.

Recommendation: Investigate memory usage of database heap.

Rank: 2

There is one database heap per database and the database manager uses it on behalf of all applications connected to the database. The data area is expanded as needed up to the maximum specified by dbheap.

For more information on the database heap, refer to the DB2 Information Center.

Investigate the amount of memory that was used for the database heap over time to determine the most appropriate value for the database heap configuration parameter. The database system monitor tracks the highest amount of memory that was used for the database heap.

Take one of the following actions:

Launch DB2 tool: Memory Visualizer

The Memory Visualizer is used to monitor memory allocation within a DB2 instance. It can be used to monitor overall memory usage, and to update configuration parameters for individual memory components.

To open the Memory Visualizer:

1. From the Control Center, expand the object tree until you find the instances folder.

2. Click the instances folder. Any existing instances are displayed in the contents pane on the right side of the window.
3. Right-click the instance that you want in the contents pane, and click View Memory Usage in the pop-up menu. The Memory Visualizer opens.

To start the Memory Visualizer from the command line issue the db2memvis command.

The Memory Visualizer displays a hierarchical list of memory pools for the database manager. Database Heap is listed under the Database Manager Memory group for each database. On Windows, it is listed under the Database Manager Shared Memory group.

Click the check box on the Show Plot column for the Database Heap row to add the element to the plot.

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 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

SYSCAT.ROUTINES に対する 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

```
►► 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

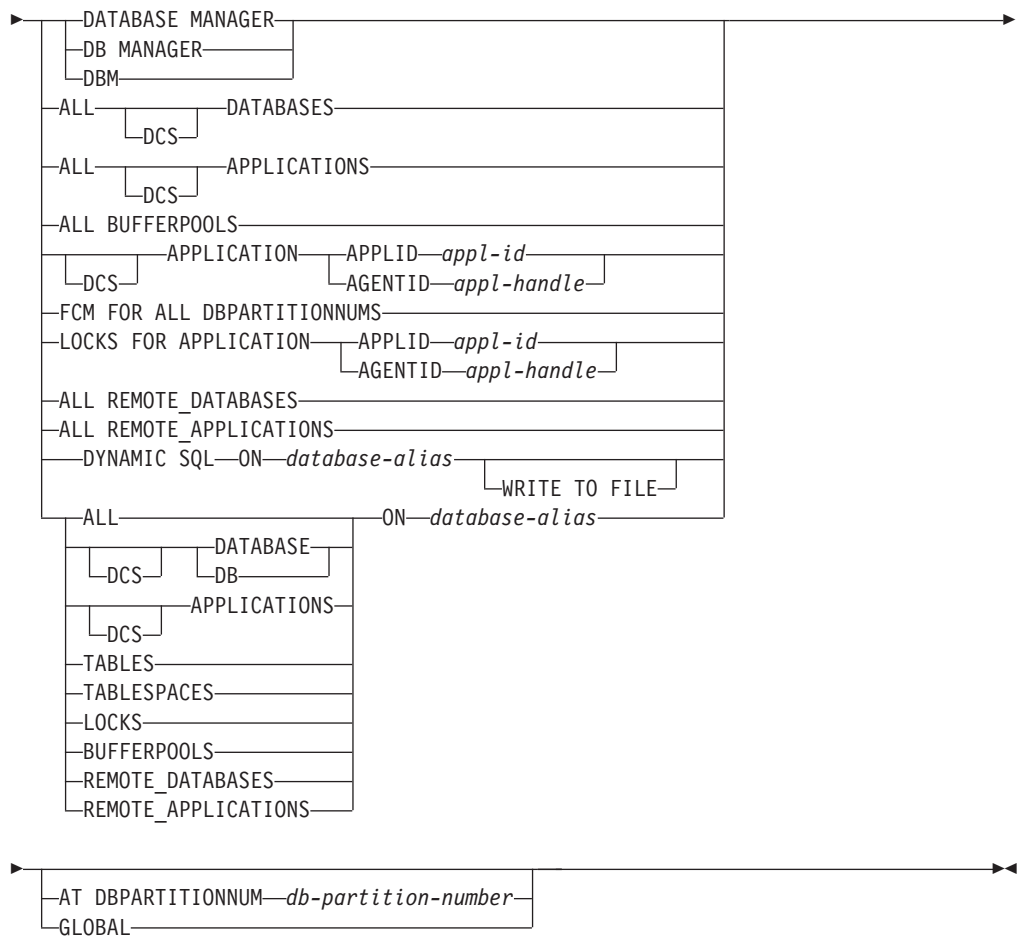
```
GET ROUTINE INTO procs/proc1.sar FROM PROCEDURE myappl.proc1;
```

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されます。"-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。



統計を収集するために、モニター・スイッチを ON 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一般統計を提供します。

ALL APPLICATIONS

現行データベース・パーティション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 DBPARTITIONNUMS

GET SNAPSHOT コマンドの発行対象のデータベース・パーティションと、パーティション・データベース環境の他のデータベース・パーティション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

現行データベース・パーティション上のアクティブ・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

現行データベース・パーティション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 ON 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。

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 DBPARTITIONNUM *db-partition-number*

指定されたデータベース・パーティションの結果を戻します。

GLOBAL

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・パーティションの集約結果を戻します。

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 database partitions 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 db partition 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        = 0
Free FCM buffers                          = 128
Free FCM buffers low water mark           = 128
Free FCM channels                         = 128
Free FCM channels low water mark          = 128

```

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す。

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
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
Coordinating database partition number = 0
Current database partition 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
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	= AAAAANBX
Package Version ID	=
Cursor name	= SQLCUR201
Statement database partition 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
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 = 0x0002000E00000000000000000054
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、

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
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 I/Os            = 0
Number read log I/Os             = 0
Number partial page log I/Os    = 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
    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
```

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.log 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 OFF である。
 - スイッチを ON 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。

- すべてのデータベース・パーティションからのスナップショット情報 (すべてのパーティション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、すべてのノードでの個々の最高水準点の合計ではなく、すべてのノード間で最大の、接続の最高水準点を表します。以下に例を示します。
 - ノード A 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。
 - ノード B 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。

上の例では、「High water mark for connections」値は 6 で、「Applications connected currently」値は 9 です。

互換性

バージョン 8 より前のバージョンとの互換性 :

- キーワード `NODE` は、`DBPARTITIONNUM` の代わりとして使用できます。
- キーワード `NODES` は、`DBPARTITIONNUMS` の代わりとして使用できます。
- 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:TRUE
```

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 ベースの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 character-string ▶▶
```

コマンド・パラメーター

HELP *character-string*

SQL コマンドや DB2 コマンド、またはインフォメーション・センターにリストされているその他の項目。

例

以下に示すのは、HELP コマンドの例です。

- 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。

- db2 help drop

このコマンドは、Web ブラウザーをオープンし、SQL DROP ステートメントに関する情報を表示します。これは、コマンド db2ic -j drop を呼び出すのと同じです。db2ic コマンドは、DROP と呼ばれるステートメントまたはコマンドを、最初に SQL リファレンスで検索し、次にコマンド・リファレンスで検索して、最初に検出された情報を表示します。

- db2 help 'drop database'

このコマンドは、より詳細な検索を開始し、DROP DATABASE コマンドに関する情報を表示します。

使用上の注意

インフォメーション・センターがユーザーのシステムにインストールされている必要があります。DB2 ライブラリーの HTML ブックは、`¥sqllib¥doc¥html` サブディレクトリーになければなりません。

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來ます。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來ます。

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_HISTSIZ` 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管されません。
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。

223 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

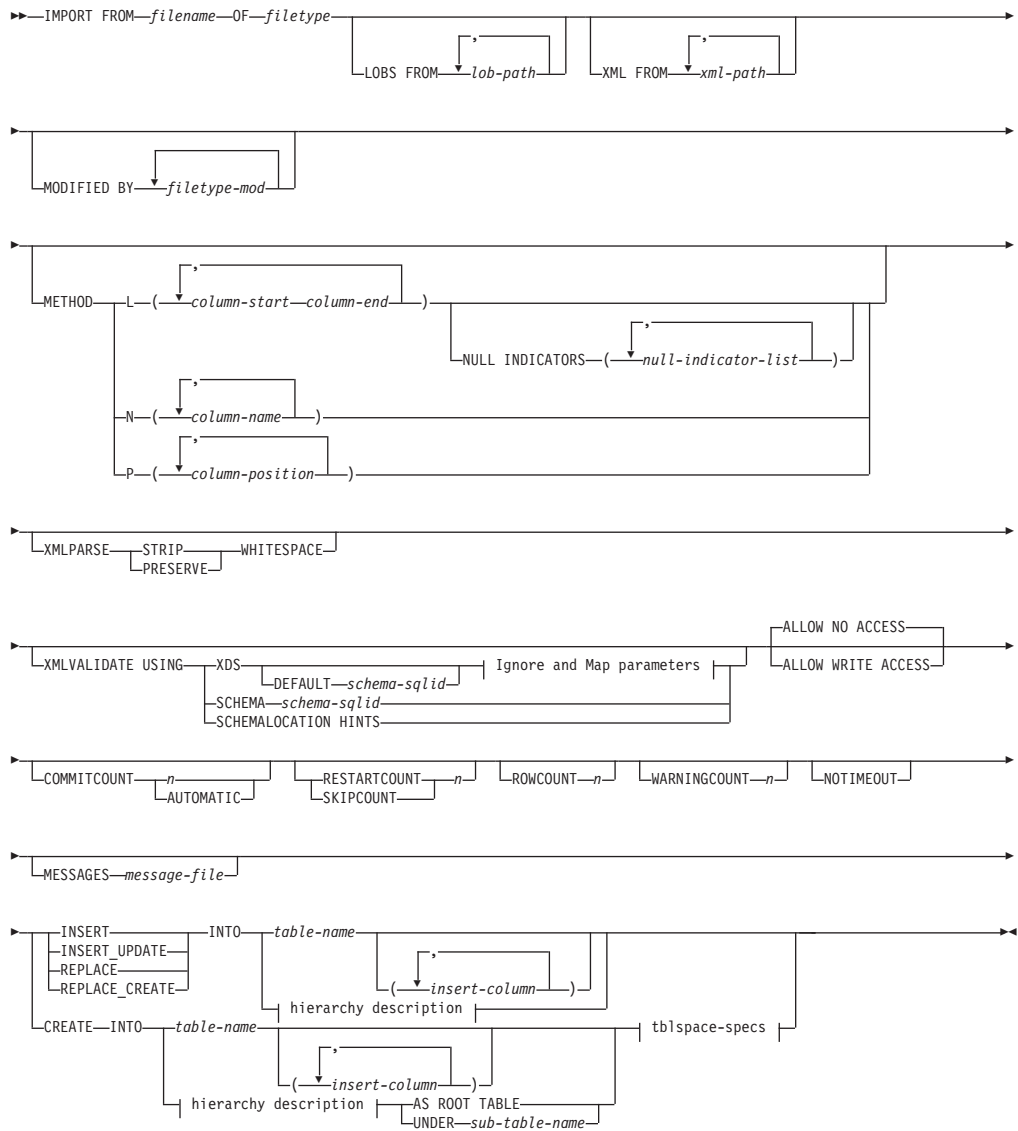
- INSERT オプションを使用して **IMPORT** する場合、以下のいずれかが必要です。
 - *sysadm*
 - *dbadm*
 - 関係するそれぞれの表、ビュー、またはニックネームに対する CONTROL 特権
 - 関係するそれぞれの表またはビューに対する INSERT および SELECT 特権
- INSERT_UPDATE オプションを使用して既存の表に **IMPORT** するには、以下のいずれかが必要です。
 - *sysadm*
 - *dbadm*
 - 関係するそれぞれの表、ビュー、またはニックネームに対する CONTROL 特権
 - 関係するそれぞれ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、以下のいずれかが必要です。
 - *sysadm*
 - *dbadm*
 - 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、以下のいずれかが必要です。
 - *sysadm*
 - *dbadm*

- 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。
 - 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、以下のいずれかが必要です。
 - *sysadm*
 - *dbadm*
 - 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。
 - 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、以下のどれかが必要です。
 - *sysadm*
 - *dbadm*
 - 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** クリデンシャルが必要です。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** クリデンシャルが必要です。
 - 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。

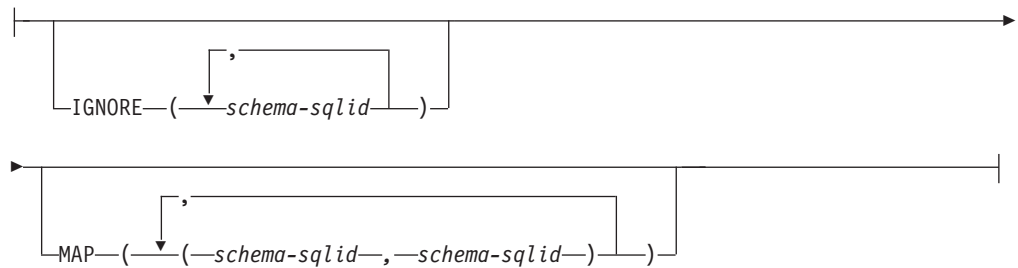
必要な接続

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



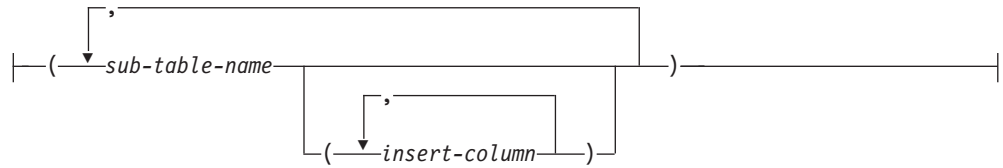
Ignore and Map parameters:



hierarchy description:



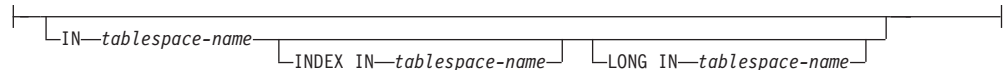
sub-table-list:



traversal-order-list:



tblspace-specs:



コマンド・パラメーター

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。

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)。

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。

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、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。

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。

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。

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。

HIERARCHY

階層データをインポートするよう指定します。

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。

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、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。

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。

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。

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、宣言一時表、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。

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 動作が暗黙的に活動化されます。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。

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ます、LARGE DMS 表スペースでなければなりません。

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、スキーマ検証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。

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。

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、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を指定します。

注: この方式は、IXF または DEL ファイルの場合にのみ使用でき、DEL ファイル・タイプに対してのみ有効なオプションです。

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。 223 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。

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。

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。

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。
- WSF (ワークシート・フォーマット)。以下のプログラムで使用します。
 - Lotus 1-2-3
 - Lotus Symphony
- 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。

ニックネームにインポートするときには、WSF ファイル・タイプはサポートされません。

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。

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ト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。

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。

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。

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。

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。

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。

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。

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。

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。

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。

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** 節を指定して **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。

以下に示す **XMLVALIDATE** オプションの例を参照してください。

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 PC/IXF 形式ファイルの H レコードには、製品 "DB2 01.00"、日付 "19970220"、および時刻 "140848" が入っています。

SQL3153N PC/IXF 形式ファイルの T レコードは、名前 "myfile"、修飾子 " "、およびソース " " を持っています。

SQL3109N ユーティリティが、ファイル "myfile" からデータのロードを開始しています。

SQL3110N ユーティリティーが処理を完了しました。"58" 行が、
入力ファイルから読み取られました。

SQL3221W ...COMMIT WORK が開始されました。入力レコード数 = "58"。
SQL3222W ...すべてのデータベース変更のコミットが成功しました。

SQL3149N "58" 行が、入力ファイルから処理されました。"58" 行が、
正常に表に挿入されました。"0" 行が、拒否されました。

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 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):

```
"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 およ
び 4 は、それぞれユーザー提供の IDENTITY 値 100 と 101 が割り当てられま
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' />
```

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。

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 COMMIT 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。自動の COMMIT が実行されない場合にログが満杯になると、ROLLBACK が実行されます。

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。
- バッファー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 COMMIT 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 COMMIT 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。

PC/IXF ファイルから表を再作成するときは、参照制約と外部キー定義は保存されません。(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、DEL、または WSF 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

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、通常の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・ページが異なっており、**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・ページが異なっており、**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・ページが異なっており、**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、宣言された一時表、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使用して、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 または WSF 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。

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・パラメーター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・パラメーター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 12. 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: すべてのファイル・フォーマット

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>

表 12. 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: すべてのファイル・フォーマット (続き)

修飾子	説明
generatedignore	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。
generatedmissing	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。
identityignore	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。
identitymissing	この修飾子を指定すると、ユーティリティーは、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。
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 のロケーションの文字列表現です。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、文字列 <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>LOBS FROM 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。LOBS FROM 節によって、LOBSINFILE 動作が自動的に活性化されます。LOBS FROM 節は、データのインポート中に、IMPORT ユーティリティーに 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> です。</p>
no_type_id	単一の副表にインポートするときのみ有効です。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。

表 12. 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: すべてのファイル・フォーマット (続き)

修飾子	説明
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"> • 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。 • 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。 • 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、ユーティリティーは処理を停止します。
norowwarnings	リジェクトされた行についての警告をすべて抑制します。
rowchangetimestampignore	<p>この修飾子は、インポート・ユーティリティーに、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、それらを見捨てるべきことを知らせます。この結果、すべての ROW CHANGE TIMESTAMP はユーティリティー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。</p>
rowchangetimestampmissing	<p>この修飾子を指定すると、ユーティリティーは、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。</p>
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・ラベル値のストリング形式であることを示します。IMPORT 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。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 (SQLSTATE 01H53) が戻されます。ストリング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。IMPORT 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>注: 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 12. 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: すべてのファイル・フォーマット (続き)

修飾子	説明
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。欠落データの例は次のとおりです。</p> <ul style="list-style-type: none"> • DEL ファイルの場合: 2 つの連続する列区切り文字 ("、"), または任意の数のスペース (" , ") で分離された 2 つの連続する列区切り文字が、列の値に指定されます。 • DEL/ASC/WSF 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。 <p>注: 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。NULL 列値は、数値、日付、およびタイム・スタンプ列の場合は全桁スペース文字、または任意のタイプの列の場合は NULL INDICATOR を使用して表現され、列が NULL であることを示します。</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"> • DEL/ASC/WSF 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。

表 13. 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><i>x</i> 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"> • 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。 • nullindchar には、標準の ASCII セットに組み込む (コード・ポイント x20 から x7F の範囲の) 記号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。 <p>注:</p> <ol style="list-style-type: none"> 1. codepage 修飾子を lobsinfile 修飾子と一緒に使うことはできません。 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。

表 13. 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。² 有効な日付要素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) M - 月 (1 から 12 の範囲の 1 桁または 2 桁の数) MM - 月 (1 から 12 の範囲の 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 (1 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。 他の日または月要素とは相互に排他的)</p> <p>指定されない各要素には、デフォルト値の 1 が割り当てられます。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。² 有効な時刻要素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) HH - 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 2 桁の数; H と相互に排他的) M - 分 (0 から 59 の範囲の 1 桁または 2 桁の数) MM - 分 (0 から 59 の範囲の 2 桁の数。 M とは相互に排他的) S - 秒 (0 から 59 の範囲の 1 桁または 2 桁の数) SS - 秒 (0 から 59 の範囲の 2 桁の数。 S と相互に排他的) SSSSS - 夜中の 12 時から数えた秒数 (00000 から 86399 の範囲の 5 桁の数。 他の時刻要素とは相互に排他的) TT - 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 13. 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。² 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 (1 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 0 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 (0 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 (0 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 86399 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>UUUUUU - マイクロ秒 (000000 から 999999 の範囲の 6 桁の数。 他のマイクロ秒エレメントとは相互に排他的)</p> <p>UUUUU - マイクロ秒 (00000 から 99999 の範囲の 5 桁の数。 000000 から 999990 の範囲にマップされる。 他のマイクロ秒エレメントとは相互に排他的)</p> <p>UUUU - マイクロ秒 (0000 から 9999 の範囲の 4 桁の数。 000000 から 999900 の範囲にマップされる。 他のマイクロ秒エレメントとは相互に排他的)</p> <p>UUU - マイクロ秒 (000 から 999 の範囲の 3 桁の数。 000000 から 999000 の範囲にマップされる。 他のマイクロ秒エレメントとは相互に排他的)</p> <p>UU - マイクロ秒 (00 から 99 の範囲の 2 桁の数。 000000 から 990000 の範囲にマップされる。 他のマイクロ秒エレメントとは相互に排他的)</p> <p>U - マイクロ秒 (0 から 9 の範囲の 1 桁の数。 000000 から 900000 の範囲にマップされる。 他のマイクロ秒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>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>"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。これらの値で大/小文字は区別されません。</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>

表 13. 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中に GRAPHIC データがある場合にのみ、表リカバリーのドロップ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>制約事項</p> <p>EXPORT 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 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 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>注: xmlgraphic 修飾子が IMPORT 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 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: 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 Y です。 ³ 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。

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: 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。³⁴ 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。³⁴</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。³⁴</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<row delimiter> "Vincent,<row delimiter>, is a manager", 4005,44.37<row delimiter></pre> <p>delprioritychar 修飾子が指定されている場合、このデータ・ファイルには 2 行しかありません。2 番目の <row delimiter> 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 <row delimiter> 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 <row delimiter> によって区切られます。</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: 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・ユーティリティーは、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	二重になっている区切り文字の認識を抑止します。

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: 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止するようにユーティリティーに指示します。 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ーに指示します。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。

表 17. `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 N であると見なされます。 警告: 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。 GRAPHIC データは、アプリケーション GRAPHIC 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。 警告: アプリケーション・コード・ページが 1 バイトの場合、 GRAPHIC データは、データベースにたとえ GRAPHIC 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 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。 警告: 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・ユーティリティー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は、区切り文字、または DEL ファイル・フォーマットのフィールド区切り文字と同じであってはなりません。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。

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。以下に、いくつかのあいまいなタイム・スタンプ・フォーマットを示します。

```
"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。

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. **WSF** ファイル形式は、XML 列ではサポートされません。
7. **CREATE** モードは、XML 列ではサポートされません。

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・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・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。

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。パーティション表では、CHECK DATABASE および CHECK TABLESPACE オプションには、個々のデータ・パーティションおよびパーティション化されていない索引が含まれます。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、以下のどれかになります。

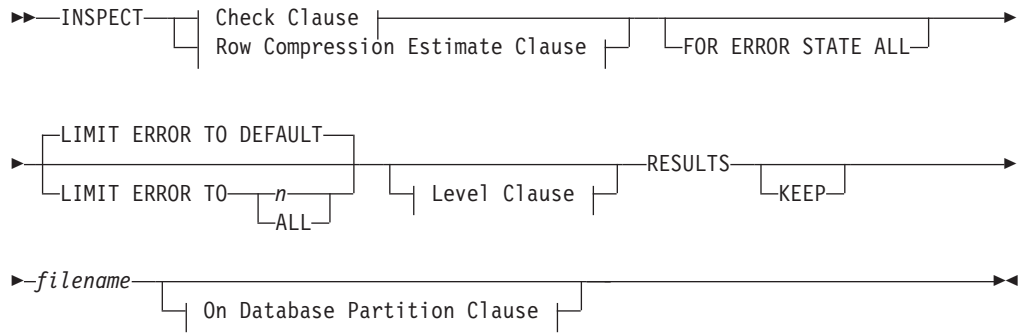
- *sysadm*
- *dbadm*
- *sysctrl*

- *sysmaint*
- 単一表の場合 CONTROL 特権。

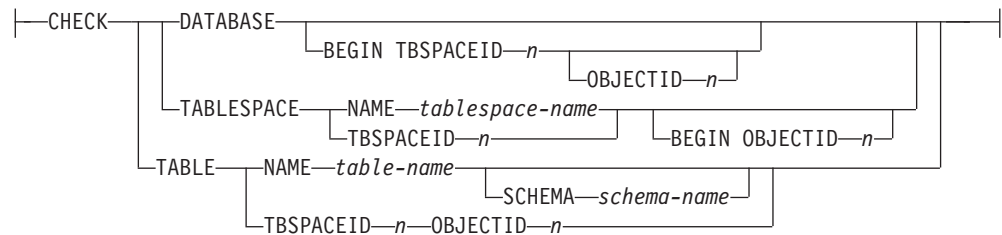
必要な接続

データベース

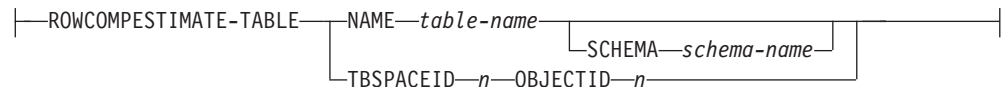
コマンド構文



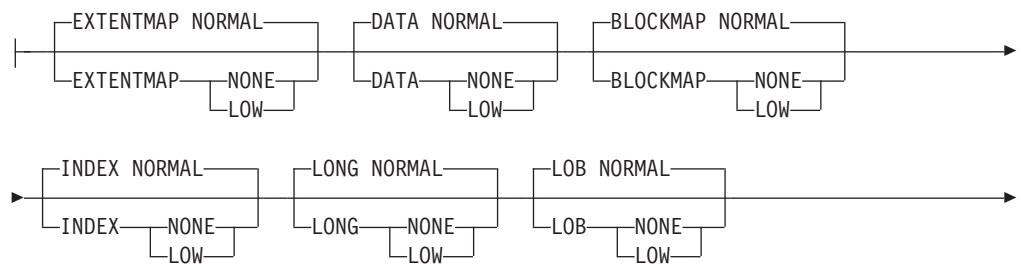
Check Clause:

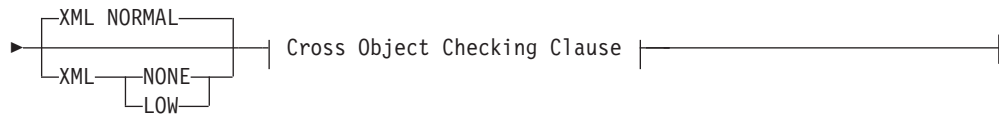


Row Compression Estimate Clause:



Level Clause:

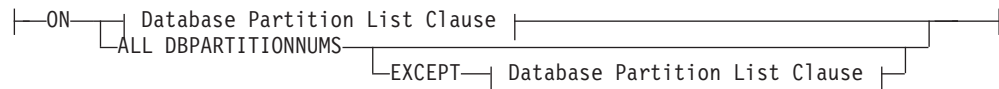




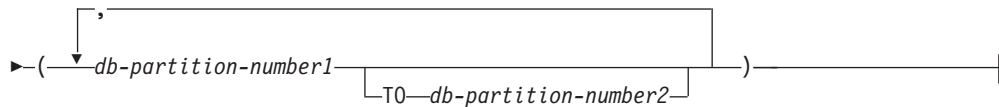
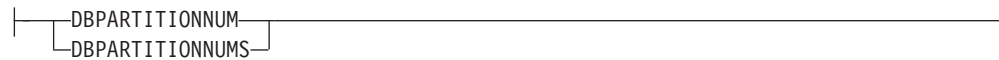
Cross Object Checking Clause:



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、この操作の対象となるデータベース・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。

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。ノード節が指定されていない場合、これがデフォルトです。

EXCEPT

ノード・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または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** というテーブル内のデータが圧縮されるとどの程度のストレージ・スペースが節約されるかを見積もるには、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

- 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。
- 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。
- 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。
- 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、ユーティリティー 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。

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、`$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

以下のいずれか。

- `sysadm`
- `sysctrl`
- `sysmaint`
- `sysmon`

コマンド構文

```
▶▶—LIST ACTIVE DATABASES—▶▶
┌──AT DBPARTITIONNUM—db-partition-number──┐
└──GLOBAL──┘
```

コマンド・パラメーター

AT DBPARTITIONNUM *db-partition-number*

モニター・スイッチの状況を表示するデータベース・パーティションを指定します。

GLOBAL

パーティション・データベース環境内のすべてのノードの集約結果を戻します。

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  = 0
Database path           = /home/smith/smith/NODE0000/SQL00002/

Database name           = SAMPLE
Applications connected  = 1
Database path           = /home/smith/smith/NODE0000/SQL00001/
```

互換性

バージョン 8 より前のバージョンとの互換性：

- キーワード `NODE` は、`DBPARTITIONNUM` の代わりとして使用できます。

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、それが発行されたデータベース・パーティションの情報だけを戻します。

許可

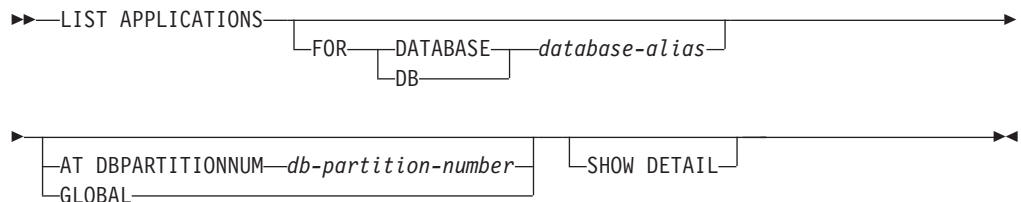
以下のいずれか。

- `SYSADM`
- `SYSCTRL`
- `SYSMAINT`
- `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データベース・パーティション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、以下のものが含まれます。

- 許可 ID
- アプリケーション名
- アプリケーション・ハンドル
- Application ID
- データベース名
- エージェントの数

AT DBPARTITIONNUM *db-partition-number*

モニター・スイッチの状況を表示するデータベース・パーティションを指定します。

GLOBAL

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・パーティションの集約結果を戻します。

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 DB パーティション番号
- コーディネーター・パーティション ID/スレッド
- Status
- 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。

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

バージョン 8 より前のバージョンとの互換性：

- キーワード **NODE** は、**DBPARTITIONNUM** の代わりとして使用できます。

LIST COMMAND OPTIONS

環境変数の現行の設定値のリストを表示します。

- DB2BQTIME
- DB2DQTRY
- DB2RQTIME
- DB2IQTIME
- DB2OPTIONS.

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      (DB2DQTRY) = 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
-c	Auto-Commit	ON

System Database Directory

Number of entries in the directory = 2

Database 1 entry:

Database alias	= SAMPLE
Database name	= SAMPLE
Local database directory	= /home/smith
Database release level	= 8.00
Comment	=
Directory entry type	= Indirect
Catalog database partition number	= 0
Alternate server hostname	= montero
Alternate server port number	= 29384

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。

ローカル・データベース・ディレクトリー

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す。
- *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

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、少なくとも以下の 1 つが必要です。

- *sysadm* または *dbadm* の権限
- CONTROL 特権
- SELECT 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

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
-----
IBMCATGROUP
IBMDEFAULTGROUP

```

2 record(s) selected.

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。

互換性

バージョン 8 より前のバージョンとの互換性：

- キーワード `NODEGROUPS` は、`DATABASE PARTITION GROUPS` に置き換えられます。

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・パーティションのリストを表示します。

有効範囲

このコマンドは、`$HOME/sql11lib/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

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

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.

互換性

バージョン 8 より前のバージョンとの互換性：

- キーワード NODES は、DBPARTITIONNUMS の代わりとして使用できます。

LIST DCS APPLICATIONS

DB2 Connect Enterprise Edition によってホスト・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情報を、標準出力に表示します。

許可

以下のいずれか。

- *sysadm*
- *sysctrl*
- *sysmaint*
- *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 requester 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・リクエスターで使用される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[®] 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 のインストール先パス/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 (l を入力)
- 未確定トランザクション番号 *x* のリスト表示 (l 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

履歴ファイルの中の項目のリストを表示します。履歴ファイルには、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。

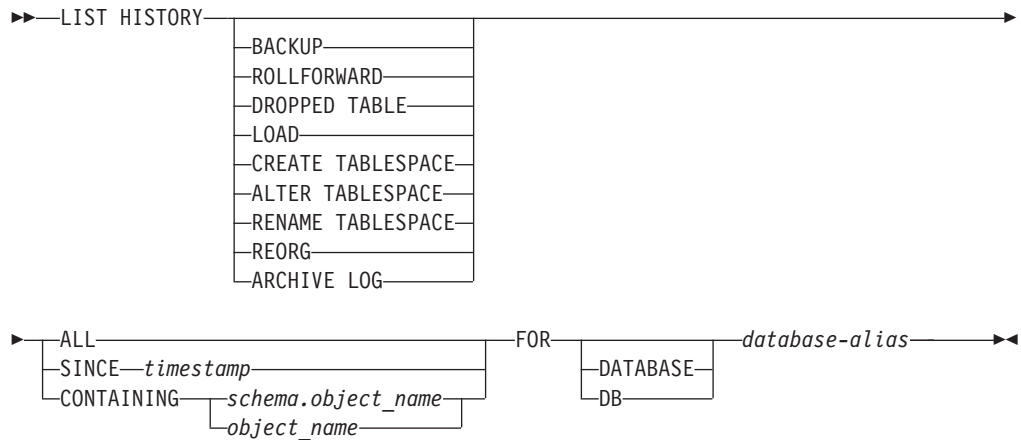
許可

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

現在履歴ファイル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 履歴ファイルのうち、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 (形式は *yyyymmddhhmmss*)、または先頭の接頭部 (最小値は *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*

リカバリー履歴ファイルをリスト表示するデータベースを指定します。

例

```
db2 list history since 19980201 for sample
db2 list history backup containing userspace1 for sample
db2 list history dropped table all for db sample
```

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。

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 - ドロップされた表
- F - ロールフォワード
- G - 表の再編成
- 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

タイプ

アーカイブ・ログ・タイプ:

- P - 1 次ログ・パス
- M - 2 次 (ミラー) ログ・パス

N - アーカイブ・ログ・コマンド
F - フェイルオーバー・アーカイブ・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・タイプ:

F - オフライン
N - オンライン
I - 増分オフライン
O - 増分オンライン
D - デルタ・オフライン
E - デルタ・オンライン
R - 再ビルド

ロールフォワード・タイプ:

E - ログの最後
P - ポイント・イン・タイム

ロード・タイプ:

I - 挿入
R - 置換

表スペースの変更タイプ:

C - コンテナの追加
R - 再調整

静止タイプ:

S - 静止共有
U - 静止更新
X - 静止排他
Z - 静止リセット

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のかわきりなくロールバックするかわきりありません。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。

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。

```
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

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

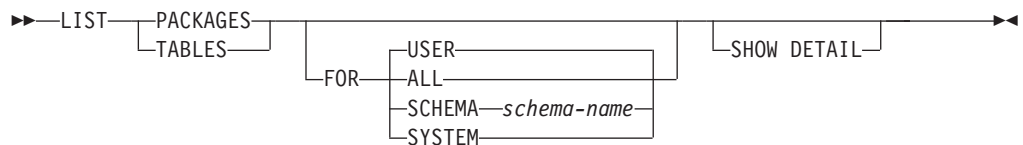
システム・カタログ SYSCAT.PACKAGES (LIST PACKAGES) と SYSCAT.TABLES (LIST TABLES) の場合には、少なくとも以下の 1 つが必要です。

- *sysadm* または *dbadm* の権限
- 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
where tabschema = 'SYSCAT'
order by tabschema, tabname;
```

```
select pkgname, pkgschema, pkgversion, unique_id, boundby, total_sect,
       valid, format, isolation, blocking
from syscat.packages
where pkgschema = 'NULLID'
order by pkgschema, pkgname, pkgversion;
```

```
select tabname, tabschema, type, create_time
from syscat.tables
where tabschema = 'USER'
order by tabschema, tabname;
```

```
select pkgname, pkgschema, pkgversion, unique_id, boundby, total_sect,  
       valid, format, isolation, blocking  
from syscat.packages  
where pkgschema = USER  
order by pkgschema, pkgname, pkgversion;
```

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。

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

```
▶▶—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)

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。

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戻します。

許可

以下のいずれか。

- SYSADM
- SYSCTRL
- SYSMAINT
- 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
0x20000	Load in progress
0x2000000	Storage may be defined
0x4000000	StorDef is in 'final' state
0x8000000	StorDef was change prior to roll forward
0x10000000	DMS rebalance in progress
0x20000000	Table space deletion in progress
0x40000000	Table space creation in progress

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)
- コンテナーの数
- 最小リカバリー時間 (0 以外の場合のみ表示)
- 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)
- 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
State change tablespace ID = 2
State change object ID    = 3
Number of quiesceners     = 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:
    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。

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。

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 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。

次の表では、サポートされているそれぞれの表スペース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 18. サポートされる表スペースの状態

State	16 進数の状態値	説明	例
バックアップ・ペンディング	0x20	<p>ポイント・イン・タイム表スペースのロールフォワード操作の後、または COPY NO オプションを指定した (リカバリー可能データベースに対する) ロード操作の後、表スペースはこの状態になります。表スペースを使用するには、その前に表スペース (またはデータベース全体) をバックアップしておく必要があります。表スペースをバックアップしないと、その表スペース内の表を照会することはできません。</p> <p>注: データベースは、ロールフォワード・リカバリーが有効になった直後にもバックアップする必要があります。データベースのリカバリーは、logretain データベース構成パラメーターを RECOVERY、または userexit データベース構成パラメーターを YES に設定した場合に行えます。そのデータベースをバックアップするまでは、活動化または接続を行えません。バックアップが取られた時点で、backup_pending 情報データベース構成パラメーターの値が NO に設定されます。</p>	<p>1. 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: update db cfg for sample using logretain recovery; backup db sample; connect to sample; load from staff_data.del of del messages load.msg insert into staff copy no; update staff set salary = 69000 where id = 11;</pre> <p>2.</p> <pre>update db cfg for sample using logretain recovery; connect to sample;</pre>

表 18. サポートされる表スペースの状態 (続き)

State	16 進数の状態値	説明	例
バックアップ進行中	0x800	これは、一時的な状態であり、バックアップ操作時のみ有効です。	<p>オンライン BACKUP DATABASE コマンドを発行します。</p> <pre>backup db sample online;</pre> <p>バックアップ操作の実行中に、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;</pre> <p>1.</p> <pre>list tablespaces show detail;</pre> <p>または</p> <p>2.</p> <pre>get snapshot for tablespaces on sample;</pre> <pre>connect reset;</pre> <p>USERSPACE1 について戻される情報は、この表スペースがバックアップ進行中状態にあることを示します。</p>
DMS 再平衡化進行中	0x10000000	これは、一時的な状態であり、データの再平衡化操作時のみ有効です。データベース管理スペース (DMS) として定義されている表スペースに新しいコンテナが追加されるか、または既存のコンテナが拡張されたときに、表スペースのデータの再平衡化が発生することがあります。再平衡化とは、データのストライピングを維持する試みにおいて、表スペースのエクステントを 1 つの場所から別の場所に移動する処理のことです。エクステントとは、コンテナ・スペースの単位 (測定基準はページ数) で、ストライプとは、表スペースの、コンテナのセット全体にわたるエクステントの層のことです。	<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;</pre> <pre>create tablespace ts1 managed by database using (file '/home/melnik/melnik/NODE0000/SQL00001/ts1c1' 1024);</pre> <pre>create table newstaff like staff in ts1;</pre> <pre>load from staffdata.del of del insert into newstaff nonrecoverable;</pre> <pre>alter tablespace ts1 add (file '/home/melnik/melnik/NODE0000/SQL00001/ts1c2' 1024);</pre> <pre>list tablespaces;</pre> <pre>connect reset;</pre> <p>TS1 について戻される情報は、この表スペースが DMS 再平衡化進行中状態にあることを示します。</p>

表 18. サポートされる表スペースの状態 (続き)

State	16 進数の 状態値	説明	例
使用不可ペ ンディング	0x200	データベースのロールフォワード操作時に表スペースはこの状態になる場合があります。ロールフォワード操作の終了までには表スペースのこの状態は解除されず。この状態は、表スペースがオフラインになり、トランザクションの補正ログ・レコードが作成されなかった結果として生じる条件によってトリガーされます。この表スペース状態になり、その後解除される過程は、ユーザーから認識されません。	この表スペース状態を示す例は、本書では扱われていません。
ドロップ・ ペンディング	0x8000	データベースの再始動操作時に、表スペースの 1 つ以上のコンテナで問題が見つかった場合に表スペースはこの状態になります (データベースの再始動は、このデータベースを用いた前のセッションが (例えば電源障害時に) 異常終了した場合に行う必要があります)。表スペースがドロップ・ペンディング状態にある場合は使用することができず、ドロップ以外に何も行えません。	この表スペース状態を示す例は、本書では扱われていません。

表 18. サポートされる表スペースの状態 (続き)

State	16 進数の状態値	説明	例
ロード中	0x20000	これは一時的な状態で、COPY NO オプションを指定した (リカバリー可能データベースに対する) ロード操作時のみ有効です。表のロード中状態も参照してください。	<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 logretain recovery; backup db sample; connect to sample; create table newstaff like staff; load from staffdata.del of del insert into newstaff copy no; connect reset;</pre> <p>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; list tablespaces; connect reset;</pre> <p>USERSPACE1 について戻される情報は、この表スペースがロード中 (およびバックアップ・ペンディング) 状態にあることを示します。</p>
通常	0x0	表スペースが通常以外の状態 (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、表スペースが作成された後に置かれる最初の状態です。	<pre>connect to sample; create tablespace ts1 managed by database using (file '/home/melnyk/melnyk/NODE0000/SQL00001 /tscl' 1024); list tablespaces show detail;</pre>

表 18. サポートされる表スペースの状態 (続き)

State	16 進数の状態値	説明	例
オフライン およびアクセス不可	0x4000	<p>表スペースの 1 つ以上のコンテナに問題があった場合に表スペースはこの状態になります。間違っ てコンテナの名前を変更したり、移動したり、損傷してしまうこともあり得ます。問題を正し、表スペースに関連付けられているコンテナに再びアクセスできるようになった後、すべてのアプリケーションをデータベースから切断し、データベースに再接続することによって、この異常な状態から抜け出すことができます。別の方法として、SWITCH ONLINE 節を指定して ALTER TABLESPACE ステートメントを発行することによって、他のアプリケーションをデータベースから切断せずに表スペースのオフラインおよびアクセス不可状態を解除することができます。</p>	<pre>connect to sample; create tablespace ts1 managed by database using (file '/home/melnyk/melnyk/NODE0000/SQL00001 /tsc1' 1024); alter tablespace ts1 add (file '/home/melnyk/melnyk /NODE0000/SQL00001/tsc2' 1024); export to st_data.del of del select * from staff; create table stafftemp like staff in ts1; import from st_data.del of del insert into stafftemp; connect reset;</pre> <p>表スペースのコンテナの名前を tsc1 から tsc3 に変更し、STAFFTEMP 表の照会を試行します。</p> <pre>connect to sample; select * from stafftemp;</pre> <p>照会は SQL0290N (表スペース・アクセスが許されていません) を戻し、LIST TABLESPACES コマンドによって TS1 の状態値 0x4000 (オフラインおよびアクセス不可) が戻されます。表スペースのコンテナの名前を tsc3 から tsc1 に戻します。これで照会は正常に実行されます。</p>
静止モード での排他	0x4	<p>表スペースの静止機能呼び出すアプリケーションに表スペースへの排他 (読み取りまたは書き込み) アクセスがあると、表スペースはこの状態になります。QUIESCE TABLESPACES FOR TABLE コマンドを発行することにより、表スペースを明示的に静止モードでの排他状態にすることができます。</p>	<pre>connect to sample; quiesce tablespaces for table staff reset; quiesce tablespaces for table staff exclusive; connect reset;</pre> <p>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; select * from staff where id=60; update staff set salary=50000 where id=60; list tablespaces; connect reset;</pre> <p>USERSPACE1 について戻される情報は、この表スペースが静止モードでの排他状態にあることを示します。</p>

表 18. サポートされる表スペースの状態 (続き)

State	16 進数の状態値	説明	例
静止モードでの共有	0x1	表スペースの静止機能呼び出すアプリケーションと同時にアプリケーションの両方に表スペースへの読み取りアクセスがあると (書き込みアクセスはない)、表スペースはこの状態になります。 QUIESCE TABLESPACES FOR TABLE コマンドを発行することにより、表スペースを明示的に静止モードでの共有状態にすることができます。	<p>表スペースの状態を静止モードでの共有に設定する前に通常にします。</p> <pre>connect to sample; quiesce tablespaces for table staff reset; quiesce tablespaces for table staff share; connect reset;</pre> <p>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; select * from staff where id=40; update staff set salary=50000 where id=40; list tablespaces; connect reset;</pre> <p>USERSPACE1 について戻される情報は、この表スペースが静止モードでの共有状態にあることを示します。</p>
静止モードでの更新	0x2	表スペースの静止機能呼び出すアプリケーションに表スペースへの排他書き込みアクセスがあると、表スペースはこの状態になります。 QUIESCE TABLESPACES FOR TABLE コマンドを発行することにより、表スペースを明示的に静止モードでの更新状態にすることができます。	<p>表スペースの状態を静止モードでの更新に設定する前に通常にします。</p> <pre>connect to sample; quiesce tablespaces for table staff reset; quiesce tablespaces for table staff intent to update; connect reset;</pre> <p>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; select * from staff where id=50; update staff set salary=50000 where id=50; list tablespaces; connect reset;</pre> <p>USERSPACE1 について戻される情報は、この表スペースが静止モードでの更新状態にあることを示します。</p>

表 18. サポートされる表スペースの状態 (続き)

State	16 進数の状態値	説明	例
REORG 進行中	0x400	これは、一時的な状態であり、REORG 操作時のみ有効です。	<p>REORG TABLE コマンドを発行します。</p> <pre>connect to sample; reorg table staff; connect reset;</pre> <p>REORG 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;</pre> <p>1. list tablespaces show detail;</p> <p>または</p> <p>2. get snapshot for tablespaces on sample; connect reset; <p>USERSPACE1 について戻される情報は、この表スペースが REORG 進行中状態にあることを示します。 注: SAMPLE データベースが関係する表の再編成操作は短時間で完了する可能性があり、結果として、このアプローチを用いて REORG 進行中状態を観察するのが難しくなる場合があります。</p> </p>
リストア・ペンディング	0x100	データベースの表スペースは、リダイレクト・リストア操作の最初の部分の後 (つまり、SET TABLESPACE CONTAINERS コマンドを発行する前)、この状態になります。表スペースを使用するには、その前に表スペース (またはデータベース全体) をリストアする必要があります。リストア操作が正常に完了するまでは、データベースへの接続を行えません。リストア操作が完了した時点で、 restore_pending 情報データベース構成パラメーターの値が NO に設定されます。	「ストレージを定義可能」状態にあるリダイレクト・リストア操作の最初の部分が完了すると、すべての表スペースがリストア・ペンディング状態になります。

表 18. サポートされる表スペースの状態 (続き)

State	16 進数の 状態値	説明	例
リストア進行中	0x2000	これは、一時的な状態であり、リストア操作時のみ有効です。	<pre>update db cfg for sample using logretain recovery; backup db sample; backup db sample tablespace (userspace1);</pre> <p>このバックアップ・イメージのタイム・スタンプは、次のようになります。</p> <p>20040611174124</p> <pre>restore db sample tablespace (userspace1) online taken at 20040611174124;</pre> <p>リストア操作の実行中に、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;</pre> <p>1.</p> <pre>list tablespaces show detail;</pre> <p>または</p> <p>2.</p> <pre>get snapshot for tablespaces on sample; connect reset;</pre> <p>USERSPACE1 について戻される情報は、この表スペースがリストア進行中状態にあることを示します。</p>

表 18. サポートされる表スペースの状態 (続き)

State	16 進数の 状態値	説明	例
ロールフォワード・ペンディング	0x80	<p>リカバリー可能データベースに対するリストア操作の後、表スペースはこの状態になります。表スペースを使用するには、その前に表スペース (またはデータベース全体) をロールフォワードする必要があります。データベースのリカバリーは、logretain データベース構成パラメーターを RECOVERY、または userexit データベース構成パラメーターを YES に設定した場合に行えます。ロールフォワード操作が正常に完了するまでは、データベースの活動化または接続を行えません。リストア操作が完了した時点で、rollfwd_pending 情報データベース構成パラメーターの値が NO に設定されます。</p>	<p>リストア進行中のオンライン表スペースのリストア操作が完了すると、表スペース USERSPACE1 はロールフォワード・ペンディング状態になります。</p>

表 18. サポートされる表スペースの状態 (続き)

State	16 進数の状態値	説明	例
ロールフォワード進行中	0x40	これは、一時的な状態であり、ロールフォワード操作時のみ有効です。	<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 logretain recovery; backup db sample; connect to sample; create tablespace ts1 managed by database using (file '/home/melnyk/melnyk/NODE0000/SQL00001/ts1c1' 1024); create table newstaff like staff in ts1; connect reset; backup db sample tablespace (ts1) online;</pre> <p>このバックアップ・イメージのタイム・スタンプは、次のようになります。</p> <p>20040630000715</p> <pre>connect to sample; load from staffdata.del of del insert into newstaff copy yes to /home/melnyk/backups; connect reset; restore db sample tablespace (ts1) online taken at 20040630000715; rollforward db sample to end of logs and stop tablespace (ts1) online;</pre> <p>ロールフォワ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;</pre> <p>1.</p> <pre>list tablespaces show detail;</pre> <p>または</p> <p>2.</p> <pre>get snapshot for tablespaces on sample; connect reset;</pre> <p>TS1 について戻される情報は、この表スペースがロールフォワード進行中状態にあることを示します。</p>
ストレージを定義可能	0x2000000	データベースの表スペースは、リダイレクト・リストア操作の最初の部分の後 (つまり、SET TABLESPACE CONTAINERS コマンドを発行する前)、この状態になります。これにより、必要に応じてコンテナを再定義することができます。	<pre>backup db sample;</pre> <p>このバックアップ・イメージのタイム・スタンプが 20040613204955 であるとします。</p> <pre>restore db sample taken at 20040613204955 redirect; list tablespaces;</pre> <p>LIST TABLESPACES コマンドによって戻される情報は、すべての表スペースが「ストレージを定義可能」および「リストア・ペンディング」状態にあることを示します。</p>

表 18. サポートされる表スペースの状態 (続き)

State	16 進数の状態値	説明	例
ストレージを定義する必要があります	0x1000	表スペース・コンテナの設定フェーズが省略されるか、またはそのフェーズ中に指定のコンテナを獲得できない場合、データベースの表スペースは新しいデータベースに対するリダイレクト・リストア操作時にこの状態になります。表スペース・コンテナの設定フェーズ中に指定のコンテナを獲得できないという状況は、例えば無効なパス名が指定されたり、十分なディスク・スペースがない場合に発生する可能性があります。	<pre>backup db sample;</pre> <p>このバックアップ・イメージのタイム・スタンプが 20040613204955 であるとします。</p> <pre>restore db sample taken at 20040613204955 into mydb redirect; set tablespace containers for 2 using (path 'ts2c1'); list tablespaces;</pre> <p>LIST TABLESPACES コマンドによって戻される情報は、表スペース SYSCATSPACE および表スペース TEMPSPACE1 が、「ストレージを定義する必要があります」、「ストレージを定義可能」、および「リストア・ペンディング」の状態にあることを示します。「ストレージを定義する必要があります」状態は、「ストレージを定義可能」状態よりも優先されます。</p>
表スペース作成の進行中	0x4000000	これは、一時的な状態であり、表スペースの作成操作時のみ有効です。	<pre>connect to sample; create tablespace ts1 managed by database using (file '/home/melnyk/melnyk/NODE0000/SQL00001/tsc1' 1024); create tablespace ts2 managed by database using (file '/home/melnyk/melnyk/NODE0000/SQL00001/tsc2' 1024); create tablespace ts3 managed by database using (file '/home/melnyk/melnyk/NODE0000/SQL00001/tsc3' 1024);</pre> <p>表スペースの作成操作の実行中に、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;</pre> <ol style="list-style-type: none"> list tablespaces show detail; <p>または</p> <ol style="list-style-type: none"> get snapshot for tablespaces on sample; connect reset; <p>TS1、TS2、および TS3 について戻される情報は、これらの表スペースが表スペース作成の進行中状態にあることを示します。</p>

表 18. サポートされる表スペースの状態 (続き)

State	16 進数の状態値	説明	例
表スペース削除の進行中	0x20000000	これは、一時的な状態であり、表スペースの削除操作時のみ有効です。	<pre>connect to sample; create tablespace ts1 managed by database using (file '/home/melnyk/melnyk/NODE0000/SQL00001 /tsc1' 1024); create tablespace ts2 managed by database using (file '/home/melnyk/melnyk/NODE0000/SQL00001 /tsc2' 1024); create tablespace ts3 managed by database using (file '/home/melnyk/melnyk/NODE0000/SQL00001 /tsc3' 1024); drop tablespace ts1; drop tablespace ts2; drop tablespace ts3;</pre> <p>表スペースの削除操作の実行中に、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;</pre> <p>1.</p> <pre>list tablespaces show detail;</pre> <p>または</p> <p>2.</p> <pre>get snapshot for tablespaces on sample; connect reset;</pre> <p>TS1、TS2、および TS3 について戻される情報は、これらの表スペースが表スペース削除の進行中状態にあることを示します。</p>

LIST UTILITIES

インスタンス上のアクティブなユーティリティーのリストを標準出力に表示します。各ユーティリティーの記述には、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

以下のいずれか。

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

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ーに関する詳細な進捗情報を表示します。

例

some_table 表に関する 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:
  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。

LOAD

データを DB2 表にロードします。サーバー上に存在するデータは、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存在するデータは、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既に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。

318 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、範囲クラスター表との互換性はありません。

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

以下のいずれか。

- *sysadm*
- *dbadm*
- データベースに対する LOAD 権限と以下のもの
 - 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)。
- 保護された列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、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 信用証明情報を持っていない限りなりません。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。
- 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。
 - 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、またはすべてのアクセスに関して、セッション許可 ID に付与された

こうしたセキュリティー・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。このセキュリティー・ラベルは、セッション許可 ID の 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ー・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、その行を保護するために使用されます。ただし、表を保護しているセキュリティー・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。

- 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。

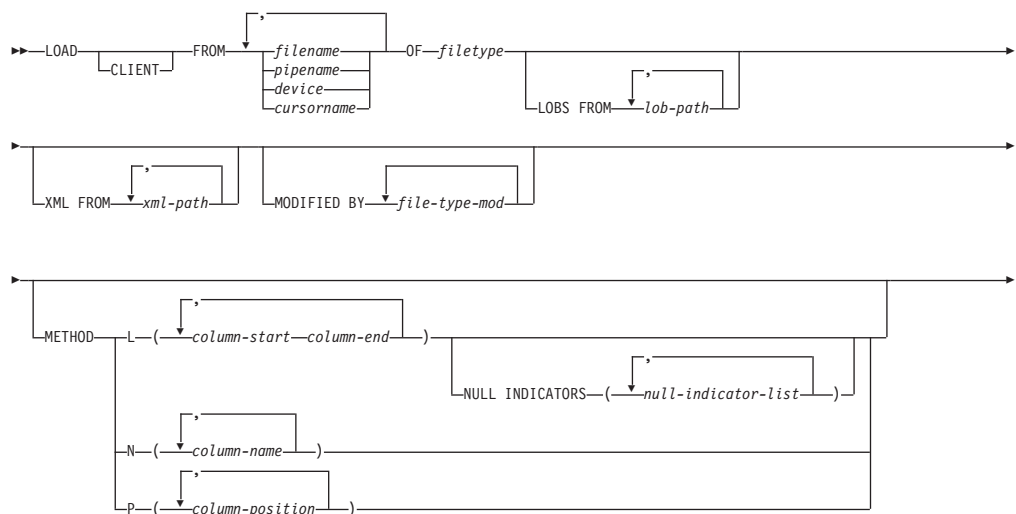
すべてのロード・プロセス (および一般にすべての DB2 サーバー・プロセス) はインスタンス所有者によって所有されており、それらのプロセスすべてにおいて、必要なファイルにアクセスするためにそのインスタンス所有者の ID を使用するため、インスタンス所有者には入力データ・ファイルに対する読み取りアクセス権が必要です。このコマンドをだれが呼び出すかには関係なく、それらの入力データ・ファイルをインスタンス所有者から読むことができなければなりません。

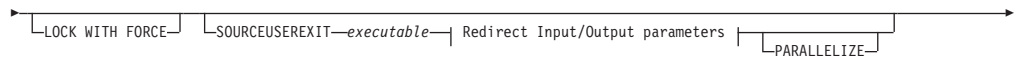
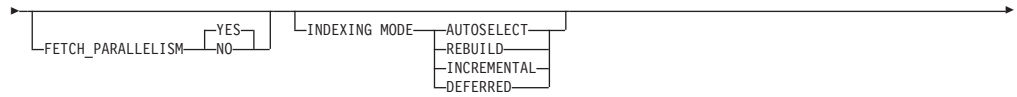
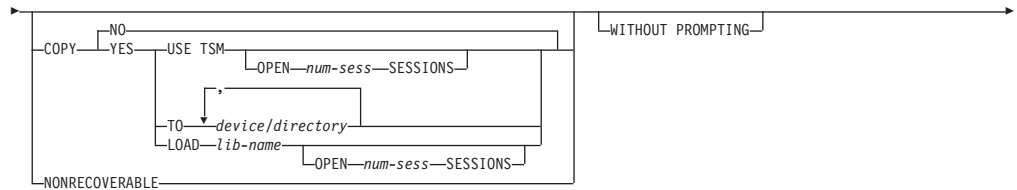
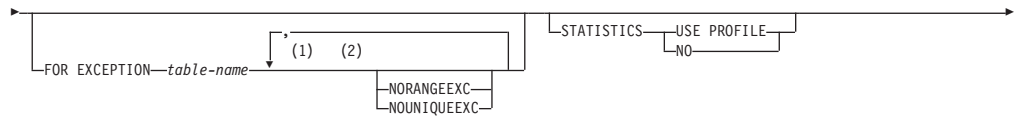
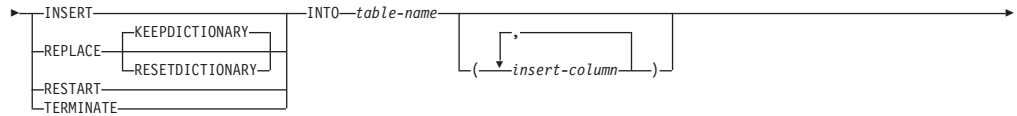
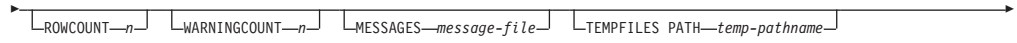
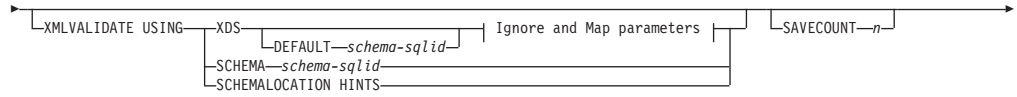
必要な接続

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。

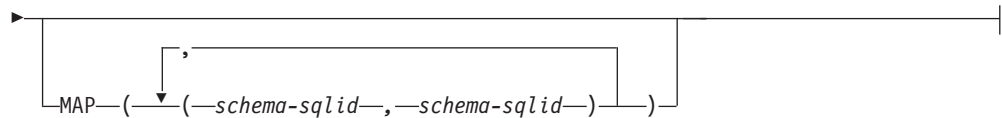
コマンド構文



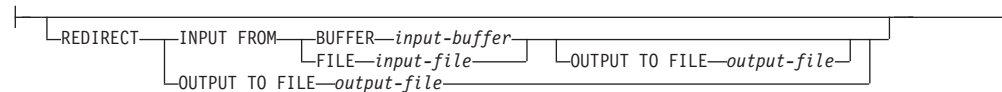


Ignore and Map parameters:





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:

1. 可能なかぎり完全修飾ファイル名を使用してください。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。
2. `ADMIN_CMD` プロシージャを使用する `EXPORT` コマンド を使っ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。
3. 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。(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。)
4. 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。

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` 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。
- `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)。

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` 動作が暗黙的に活動化されます。

`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。 318 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、そのファイル・タイプに対してのみ有効なメソッドです。

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・タイプに対してのみ有効な方式です。

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。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 IGNORE のスキーマ・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、スキーマ検証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において発行されるので、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、このオプションは無視されます。

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、ロード対象の各行ごとにロード・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、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。

KEEPDICTIONARY

既存のコンプレッション・ディクショナリーは、LOAD REPLACE 操作を介して保存されます。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。要約については、以下の表 1 を参照してください。

表 19. LOAD REPLACE KEEPDICTIONARY

圧縮	ディクショナリーの存在	結果
Y	Y	ディクショナリーを保存します。すべての入力行は既存のディクショナリーによる圧縮の影響を受けます。
Y	N	十分なユーザー・データがある場合にのみ新規ディクショナリーを表に挿入します。残りの行はディクショナリーの作成後に圧縮の影響を受けます。
N	Y	ディクショナリーを保存します。すべての入力行は圧縮されません。
N	N	影響なし。すべての行は圧縮されません。

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。ロードされるデータ・セットのサイズがゼロの場合、および前から存在するディクショナリーがある場合、ディクショナリーは保存されません。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、以下の表 2 を参照してください。

表 20. LOAD REPLACE RESETDICTIONARY

圧縮	ディクショナリーの存在	結果
Y	Y	新規ディクショナリーを作成します*。ロード対象の残りの行は、ディクショナリー作成後に圧縮の影響を受けます。
Y	N	新規ディクショナリーを作成します。残りの行は、ディクショナリー作成後に圧縮の影響を受けます。
N	Y	ディクショナリーを除去します。すべての入力行は圧縮されません。
N	N	影響なし。すべての行は圧縮されません。

* 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については、以下の表 3 を参照してください。

LOAD TERMINATE オプションでは、表スペースのバックアップ・ペンディング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については、以下の表 4 を参照してください。

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、システム表または宣言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。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。

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の列を解析できません。例えば、

```
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。

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。

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。なお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。

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 (つまり、`logretain` または `userexit` がオン) になっていれば、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完全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、表スペースをバックアップ・ペンディング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。バックアップ・ペンディング状態にある表スペース内に常駐する場合、索引はリフレッシュできません。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、または 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 (**logretain** または **logarchmeth1** が有効にされている) 場合には、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。このオプションは、フォワード・リカバリーが使用不可に設定されている場合には無効です。

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 (全バックアップまたは表スペースのバックアップ) からのみ、リストアすることができます。

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バックアップ・ペンディング状態になりません。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、または 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ない (**logretain** または **logarchmeth1** が有効にされていない) 場合には、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、表はロード・ペンディング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*

ユーティリティー内でデータを転送するためのバッファー・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、最小限必要なリソースが使用され、警告は戻されません。

このメモリーは、ユーティリティー・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。

値が指定されていない場合、実行時にユーティリティー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ー・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。指定される値は、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・ユーティリティー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・ユーティリティーは、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 より大きいなら、フラッシュ操作は非同期になり、ローダーは CPU を活用できます。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。

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・ユーティリティーは、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。

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リソースが必要となります。

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。

以下の条件がすべて真の場合、増分索引の作成はサポートされません。

- LOAD COPY オプションが指定されている (USEREXIT または LOGRETAIN オプションを指定した *logarchmeth1*)。
- 表が DMS 表スペース内に存在している。
- 索引オブジェクトの存在している表スペースが、ロードしようとしている表に属する他の表オブジェクトによって共有されている。

この制限を迂回するため、索引は別々の表スペースに置くようお勧めします。

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、表の状態はロード進行中に設定されます。

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、ターゲット表は共用モードでロックされます。表の状

態は、ロード中および読み取りアクセス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。

ALLOW READ ACCESS ロードの **LOAD TERMINATE** または **LOAD RESTART** はこのオプションを使用できますが、**ALLOW NO ACCESS** ロードの **LOAD TERMINATE** または **LOAD RESTART** はこのオプションを使用できません。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、ロード中、および読み取りアクセスの他に、**SET INTEGRITY** ペンディングに設定されます。ロードの終了時に、表のロード中状態は解除されますが、**SET INTEGRITY** ペンディングと読み取りアクセスは表の状態としてそのまま残ります。表の **SET INTEGRITY** ペンディングを解除するには、**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** ペンディングおよび読み取りアクセスの状態にある間、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留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。

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 ペンディング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、および下層即時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 ペンディング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 ペンディング読み取りアクセス状態だった下層外部キー表は、SET INTEGRITY ペンディング・アクセスなし状態になります。

DEFERRED

ロードされる表だけが、SET INTEGRITY ペンディング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 ペンディング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 ペンディング状態になります。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 ペンディング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 ペンディング状態になります。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 ペンディング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 ペンディング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留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 (SYSADM または SYSCTRL) と同じ権限を必要とします。

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ます。

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す。

詳しくは、カスタマイズ済みアプリケーション (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

予約済み。

USING *directory*

予約済み。

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   xyzN
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 の長さがそれぞれ 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。ある特定のレコードの、その列の NULL 標識位置が Y である場合、その列は NULL になります。N なら、入力レコード中のその列のデータ位置のデータ値 (L(.....)) で定義されるは、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、それはそのデータを 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

- 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られています。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) )"
```

以下のデータ・レコードをこの表にロードしようとする、

```
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 の中で 22 より後のデータは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):

```
"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 および 4 は、それぞれユーザー提供の IDENTITY 値 100 と 101 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 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. 識別に関係するファイル・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 列を持つ表のロード)

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、それぞれユーザー提供の 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、以下のコマンドに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。

```
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 コマンドの 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それぞれ 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
```

XML 文書からデータをロードする例

XML データのロード

例 1

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。これは次のようなものです。

```
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。

例 2

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
```

CURSOR 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;
```

次に、ユーザーはカーソル・タイプを使用して LOAD を発行できます。

```
LOAD FROM X1 of CURSOR INSERT INTO T2
```

XML 固有の LOAD オプションをカーソル・タイプに適用することは、ファイルからロードすることと同じです。

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) )
```

上記の 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、および MAP オプションの組み合わせを使用したいと思います。

```
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、

```
<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` 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。

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。特定の順序を希望する場合には、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・データの順序を保存する必要がない場合、以下のロード・ユーティリティーのファイル・タイプ修飾子のセクションで説明されている 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。
- 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、ロードする表に従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、整合性の設定ペンディング (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。
- クラスティング索引が表に存在する場合、ロード前にクラスティ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・クラスティング (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ー・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。セキュリティー・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、使い終え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。

- DATABASE キーワードが DECLARE CURSOR コマンド中に指定された CURSOR ファイル・タイプを使用してロードを実行する場合、現在の接続先である (ロードのために) データベースに対する認証に使用されるユーザー ID およびパスワードがソース・データベース (DECLARE CURSOR コマンドの DATABASE オプションで指定される) に対する認証にも使用されます。ロードするデータベースへの接続にユーザー ID またはパスワードが指定されなかった場合は、DECLARE CURSOR コマンド中に、ソース・データベースのユーザー ID およびパスワードを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、複数パートの 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。

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 21. LOAD TERMINATE ディクショナリー管理

表の COMPRESS 属性	LOAD 前のディク ショナリーの存在	TERMINATE: LOAD REPLACE KEEPDICTIONARY また は LOAD INSERT	TERMINATE: LOAD REPLACE RESETDICTIONARY
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。
YES	NO	何も保持しません。	何も保持しません。
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	何も行われません。	何も行われません。

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せん。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、以下の表 4 を参照してください。

表 22. LOAD RESTART デクショナリー管理

表の COMPRESS 属 性	LOAD 整合点の前 のデクショナリ ーの存在	RESTART: LOAD REPLACE KEEPDICTIONARY また は LOAD INSERT	RESTART: LOAD REPLACE RESETDICTIONARY
YES	YES	既存のデクショナリーを保持します。	既存のデクショナリーを保持します。
YES	NO	デクショナリーの作成は ADC の影響を受けます。	デクショナリーを作成します。
NO	YES	既存のデクショナリーを保持します。	既存のデクショナリーを除去します。
NO	NO	何も行われません。	何も行われません。

ロード・ユーティリティー用のファイル・タイプ修飾子

表 23. 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
anyorder	この修飾子は、 <code>cpu_parallelism</code> 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。 <code>cpu_parallelism</code> の値が 1 である場合、このオプションは無視されます。SAVECOUNT > 0 の場合、整合点後のクラッシュ・リカバリーでは、データを順番にロードする必要があるため、このオプションはサポートされません。
generatedignore	この修飾子は、ロード・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、それらは無視すべきことを知らせます。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、generatedmissing または generatedoverride 修飾子とともに使用することはできません。
generatedmissing	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、generatedignore または generatedoverride 修飾子とともに使用することはできません。

表 23. 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
generatedoverride	<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・ユーティリティーに指示します。これが役立つのは、別のデータベース・システムからデータを移行する場合や、ROLLFORWARD DATABASE コマンドで RECOVER DROPPED TABLE 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 ペンディング状態になります。ユーザー提供の値をチェックせずに表を SET INTEGRITY ペンディング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 < table-name > GENERATED COLUMN IMMEDIATE UNCHECKED</pre> <p>表の SET INTEGRITY ペンディング状態を解除し、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 < table-name > 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、LOAD コマンドが修飾子 generatedignore 自動的に変換し、ロードに進みます。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、generatedmissing または generatedignore 修飾子と共に使用することはできません。</p>
identityignore	<p>この修飾子はロード・ユーティリティーに対して、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 または identityoverride 修飾子とともに使用することはできません。</p>
identitymissing	<p>この修飾子を指定すると、ユーティリティーは、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 または identityoverride 修飾子とともに使用することはできません。</p>
identityoverride	<p>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。これが役立つのは、別のデータベース・システムからデータを移行するときに GENERATED ALWAYS として表を定義しなければならない場合や、ROLLFORWARD DATABASE コマンドで DROPPED TABLE RECOVERY 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 ID 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、identitymissing または identityignore 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。</p>

表 23. 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
indexfreespace= <i>x</i>	<p><i>x</i> は 0 から 99 の整数です。その値は、各索引ページの中で索引再作成ロード時のフリー・スペースとして残しておく部分の割合を示すパーセントとして解釈されます。ロードに INDEXING MODE INCREMENTAL を指定すると、このオプションは無視されます。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。 indexfreespace オプションは、索引のリーフ・ページにのみ影響を与えます。</p>
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。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>LOBS FROM 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。LOBS FROM 節によって、LOBSINFILE 動作が暗黙的に活性化されます。LOBS FROM 節は、データのロード中に、LOAD ユーティリティーに 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>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 db2exp.001.123.456/ がデータ・ファイルに保管される場合、LOB はファイル db2exp.001 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 db2exp.001.7.-1/ です。</p>
noheader	<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、noheader 修飾子は必要ありません。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。noheader 修飾子を使用する必要があるのは、ヘッダーのないファイルを使って LOAD_ONLY 操作を実行する場合だけです。</p>
norowwarnings	<p>リジェクトされた行についての警告をすべて抑制します。</p>

表 23. 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
pagefreespace= <i>x</i>	<p><i>x</i> は 0 から 100 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。最小行サイズのため、指定した値が無効である場合 (例えば、行の長さが少なくとも 3 000 バイトで、<i>x</i> の値が 50 である場合)、その行は新規のページに置かれます。値 100 が指定された場合には、各行がそれぞれ新しいページに置かれます。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 pagefreespace 値、または表の PCTFREE 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。pagefreespace によって設定された値は、表に対して指定されている PCTFREE 値をオーバーライドします。</p>
rowchangetimestampignore	<p>この修飾子はロード・ユーティリティーに対して、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ー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 または rowchangetimestampoverride 修飾子とともに使用することはできません。</p>
rowchangetimestampmissing	<p>この修飾子を指定すると、ユーティリティーは、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 または rowchangetimestampoverride 修飾子とともに使用することはできません。</p>
rowchangetimestampoverride	<p>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 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。これが役立つのは、別のデータベース・システムからデータを移行するときに GENERATED ALWAYS として表を定義しなければならない場合や、ROLLFORWARD DATABASE コマンドで DROPPED TABLE RECOVERY 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、rowchangetimestampmissing または rowchangetimestampignore 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 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。</p>

表 23. 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
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。LOAD 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、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>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、ロードは失敗して、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>このデータをロードまたはインポートするには、次のように SECLABELCHAR 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>
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。LOAD 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、ロードは失敗して、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>このデータをロードまたはインポートするには、次のように SECLABELNAME 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>注: 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 23. 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
totalfreespace= <i>x</i>	<p><i>x</i> 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、<i>x</i> が 20 で、データのロード後に、表に 100 のデータ・ページがある場合、20 の追加の空ページが付加されます。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。</p>
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。欠落データの例は次のとおりです。</p> <ul style="list-style-type: none"> • DEL ファイルの場合: 2 つの連続する列区切り文字 (",,")、または任意の数のスペース (" , ") で分離された 2 つの連続する列区切り文字が、列の値に指定されます。 • DEL/ASC/WSF 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。NULL 列値は、数値、日付、およびタイム・スタンプ列の場合は全桁スペース文字、または任意のタイプの列の場合は NULL INDICATOR を使用して表現され、列が NULL であることを示します。 <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"> • DEL/ASC/WSF 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。

表 24. 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
codepage= <i>x</i>	<p><i>x</i> 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"> • 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。 • 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。 • nullindchar には、標準の ASCII セットに組み込む (コード・ポイント x20 から x7F の範囲の) 記号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。 <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>

表 24. 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
dateformat="x"	<p>x はソース・ファイルの日付のフォーマットです。¹ 有効な日付要素は次のとおりです。</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 (1 から 12 の範囲の 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 (1 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>
dumpfile = x	<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許可をオーバーライドするには、dumpfileaccessall ファイル・タイプ修飾子を使用します。</p> <p>注:</p> <ol style="list-style-type: none"> 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。 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り割り込まれたりした場合、ディスクにコミットされたレコード数を正確に判別できないので、LOAD RESTART 後も一貫性が保たれるとは限りません。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。 指定されたファイルが既に存在する場合は、再作成されずに追加されます。
dumpfileaccessall	<p>ダンプ・ファイルの作成時に「OTHERS」への読み取りアクセスを認可します。</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"> dumpfile ファイル・タイプ修飾子と一緒に使用された場合。 ロード・ターゲット表に対してユーザーが SELECT 特権をもっている場合。 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。 <p>指定されたファイルが既に存在する場合、その許可は変更されません。</p>

表 24. 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
fastparse	<p>使用に際しては、注意が必要です。ユーザー提供の列値に対する構文検査を減らし、パフォーマンスを向上させてください。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、ユーザー提供のデータに :1>0-00-20-07.11.12.000000 という無効なタイム・スタンプ列値が含まれている場合、FASTPARSE が指定されていればこの値が表に挿入され、FASTPARSE が指定されていない場合はリジェクトされます。</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> <p>この修飾子は、packeddecimal 修飾子と共に使用することはできません。</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。1 有効な時刻エレメントは以下のとおりです。</p> <ul style="list-style-type: none"> 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) HH - 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 2 桁の数; H と相互に排他的) M - 分 (0 から 59 の範囲の 1 桁または 2 桁の数) MM - 分 (0 から 59 の範囲の 2 桁の数。M とは相互に排他的) S - 秒 (0 から 59 の範囲の 1 桁または 2 桁の数) SS - 秒 (0 から 59 の範囲の 2 桁の数。S と相互に排他的) SSSSS - 夜中の 12 時から数えた秒数 (00000 から 86399 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) TT - 午前/午後の指定子 (AM または PM) <p>デフォルト値の 0 が、指定されない各エレメントに割り当てられます。時刻フォーマットの例を以下に示します。</p> <pre>"HH:MM:SS" "HH.MM TT" "SSSSS"</pre>

表 24. 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
timestampformat="x"	<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。¹ 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 (1 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 0 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 (0 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 (0 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 86399 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>UUUUUU - マイクロ秒 (000000 から 999999 の範囲の 6 桁の数。 他のマイクロ秒エレメントとは相互に排他的)</p> <p>UUUUU - マイクロ秒 (00000 から 99999 の範囲の 5 桁の数。 000000 から 999990 の範囲にマップされる。 他のマイクロ秒エレメントとは相互に排他的)</p> <p>UUUU - マイクロ秒 (0000 から 9999 の範囲の 4 桁の数。 000000 から 999900 の範囲にマップされる。 他のマイクロ秒エレメントとは相互に排他的)</p> <p>UUU - マイクロ秒 (000 から 999 の範囲の 3 桁の数。 000000 から 999000 の範囲にマップされる。 他のマイクロ秒エレメントとは相互に排他的)</p> <p>UU - マイクロ秒 (00 から 99 の範囲の 2 桁の数。 000000 から 990000 の範囲にマップされる。 他のマイクロ秒エレメントとは相互に排他的)</p> <p>U - マイクロ秒 (0 から 9 の範囲の 1 桁の数。 000000 から 900000 の範囲にマップされる。 他のマイクロ秒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>

表 24. 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
timestampformat="x" (続き)	<p>YYYY、M、MM、D、DD、または DDD エlementが指定されていない場合、デフォルト値として 1 が割り当てられます。値が指定されていない MMM Elementには、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、デフォルト値として 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 Elementの有効な値は、「jan」、「feb」、「mar」、「apr」、「may」、「jun」、「jul」、「aug」、「sep」、「oct」、「nov」、および「dec」です。これらの値で大小文字は区別されません。</p> <p>TIMESTAMPFORMAT 修飾子が指定されていない場合、ロード・ユーティリティーは考えられる次の 2 つのフォーマットのいずれかを使用してタイム・スタンプ・フィールドをフォーマットします。</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。区切り文字がダッシュ '-' の場合、ロード・ユーティリティーはダッシュとドットの正規のフォーマット (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 か YYYY-MM-DD HH:MM:SS.UUUUUU のどちらかが受け入れられます。</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 24. 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
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。LOAD は、指定されている場合の codepage 修飾子、または codepage 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中に GRAPHIC データがある場合にのみ、表リカバリーのドロップ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>制約事項</p> <p>EXPORT 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 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せん。</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 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>

表 25. 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
binarynumerics	<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、reclen 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"> • 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。 • 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。 • FLOAT は、IEEE 浮動小数点フォーマットでなければなりません。 • 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。 <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>
nochecklengths	<p>nochecklengths 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>
nullindchar=x	<p>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 Y です。²</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、n も NULL 標識と認識されます。</p>

表 25. 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
packeddecimal	<p>binarynumerics 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、reclen 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、implieddecimal 修飾子と共に使用することはできません。</p>
reclen=x	<p>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。</p>
striptblanks	<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、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。</p>
striptnulls	<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、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。</p>
zoneddecimal	<p>BINARYNUMERICS 修飾子は DECIMAL フィールド・タイプでは構成されないため、ゾーン 10 進数データをロードします。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、RECLen 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、以下のいずれかになります。</p> <pre> + = 0xC 0xA 0xE 0xF - = 0xD 0xB </pre> <p>サポートされている数値は、0x0 から 0x9 です。</p> <p>サポートされているゾーン値は、0x3 および 0xF です。</p>

表 26. 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
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。²³ 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 chardel"</pre> <p>単一引用符 (') も、以下のように文字ストリングの区切り文字として指定できます。</p> <pre>modified by chardel'</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。²³</p>
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。²³</p>
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 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<row delimiter> "Vincent,<row delimiter>, is a manager", 4005,44.37<row delimiter></pre> <p>delprioritychar 修飾子が指定されている場合、このデータ・ファイルには 2 行しかありません。2 番目の <row delimiter> 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 <row delimiter> 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 <row delimiter> によって区切られます。</p>
keepblanks	<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 26. 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*) (続き)

修飾子	説明
nochardel	<p>ロード・ユーティリティーは、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。データが <i>DB2</i> 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 <i>nochardel</i> 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。</p> <p>このオプションを <i>chardelx</i>、<i>delprioritychar</i> または <i>nodoubledel</i> 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>
nodoubledel	二重になっている区切り文字の認識を抑制します。

表 27. 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: *IXF* ファイル・フォーマット

修飾子	説明
forcein	<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制するようにユーティリティーに指示します。</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。 <i>nochecklengths</i> 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>
nochecklengths	<p><i>nochecklengths</i> 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 <i>EUC</i> データがある場合、それがターゲットで 2 バイトの <i>DBCS</i> 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* を使用することはできません。フィールド区切り文字は、区切り文字、または *DEL* ファイル・フォーマットのフィールド区切り文字と同じであってはなりません。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。以下に、いくつかのあいまいなタイム・スタンプ・フォーマットを示します。

- "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. `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、以下のいずれかを使用します。

```
... modified by coldel# ...
... modified by coldel0x23 ...
... modified by coldelX23 ...
```

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. 暗黙的に隠された `row change timestamp` 列を含む表にインポートする場合、その列の暗黙的に隠されたプロパティは反映されません。そのため、列のデータがインポートするデータに含まれておらず、明示的な列リストが存在しない場合には、`rowchangetimestampmissing` ファイル・タイプ修飾子を `import` コマンドで指定する必要があります。

表 28. `codepage` および `usegraphiccodepage` 使用時の `LOAD` 動作

<code>codepage=N</code>	<code>usegraphiccodepage</code>	LOAD 動作
なし	なし	CLIENT オプションが指定があっても、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。 警告: 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。

表 28. *codepage* および *usegraphiccodepage* 使用時の *LOAD* 動作 (続き)

<i>codepage=N</i>	<i>usegraphiccodepage</i>	LOAD 動作
なし	あり	<p>CLIENT 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。 CLIENT オプションの指定があっても、GRAPHIC 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。</p> <p>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。</p> <p>警告: 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。</p>
あり	あり	<p>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。</p> <p>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。</p> <p>警告: 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。</p>

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—┬─
└─┬─SUMMARYONLY—┬─
▶┬─SHOWDELTA—┬─

```

コマンド・パラメーター

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。

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。

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。

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。

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・ユーティリティーは、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、整合性の設定ペンディング (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、表を整合性の設定ペンディング (Set Integrity Pending) 状態にします。

ロード中

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 ユーティリティのいずれか (例えばロード・ユーティリティやバックアップおよびリストア・ユーティリティ) 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 29. サポートされる表の状態

State	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 <code>staffdata.del</code> があるとします。ロード操作のターゲット表 (<code>NEWSTAFF</code> 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><code>LOAD QUERY</code> コマンドによって戻される情報は、<code>NEWSTAFF</code> 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>
ロード中	<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 <code>staffdata.del</code> 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><code>LOAD QUERY</code> コマンドによって戻される情報は、<code>NEWSTAFF</code> 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><code>LOAD QUERY</code> コマンドによって戻される情報は、<code>NEWSTAFF</code> 表が通常状態にあることを示します。</p>

表 29. サポートされる表の状態 (続き)

State	例
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 logretain recovery; backup db sample; connect to sample; create tablespace ts1 managed by database using (file '/home/meinyk/meinyk/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/meinyk/backups; connect reset; restore db sample taken at 20040629205935; rollforward db sample to end of logs and stop; connect to sample; load query table newstaff; connect reset;</pre> <p>LOAD QUERY 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/meinyk/backups; load query table newstaff; connect reset;</pre> <p>LOAD QUERY 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>
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>LOAD QUERY 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>

表 29. サポートされる表の状態 (続き)

State	例
SET INTEGRITY ペンディング	<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: connect to sample; alter table staff add constraint max_salary check (100000 - salary > 0); load from staff_data.del of del insert into staff; load query table staff;</pre> <p>LOAD QUERY コマンドによって戻される情報は、STAFF 表が SET INTEGRITY ペ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: update db cfg for sample using logretain recovery; 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>LOAD QUERY 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。

db2ckmig コマンドは、インスタンス・マイグレーションより前に発行する必要があります。db2imigr コマンドは、暗黙的に db2ckmig を呼び出します。マイグレーション作業、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。

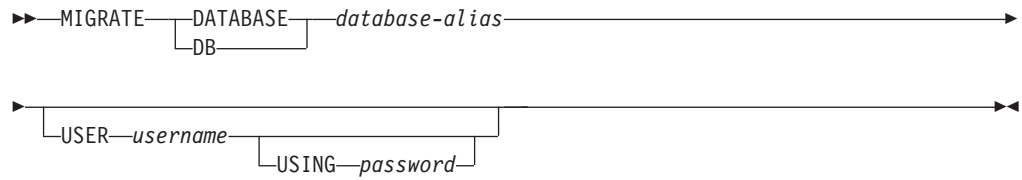
許可

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。

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マイグレーションします。

```
db2 migrate 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。

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。

マイグレーション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、マイグレーション中にログがいっぱいになるというエラーが生じる場合 (SQL1704: データベースのマイグレーションは失敗しました。理由コード "3")、データベース構成パラメーター `LOGPRIMARY` および `LOGFILSIZ` の値を増やす前に、`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マイグレーションが失敗した場合、`CLP` はそのデータベース・ディレクトリ・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)。

PING

クライアントと接続済みデータベース・サーバーの間の基礎接続のネットワーク応答時間をテストします。

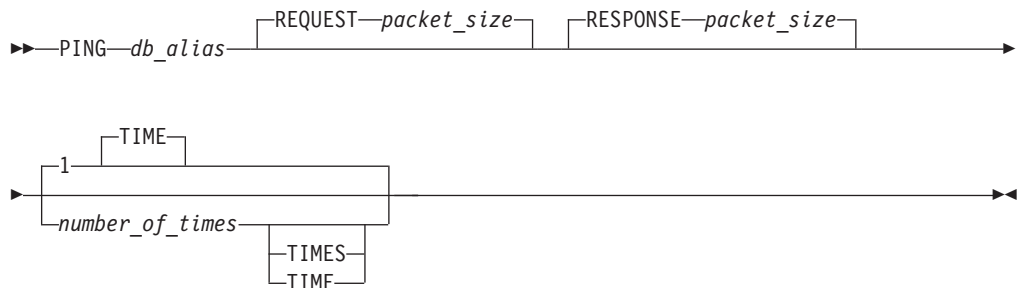
許可

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 Database 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*

クライアントに戻す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 Database 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。

戻される経過時間は、IBM Data Server Client と DB2 サーバーの間の接続に対するものです。

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。

- *sysadm* または *dbadm* の権限
- パッケージが存在しない場合は、BINDADD 権限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)

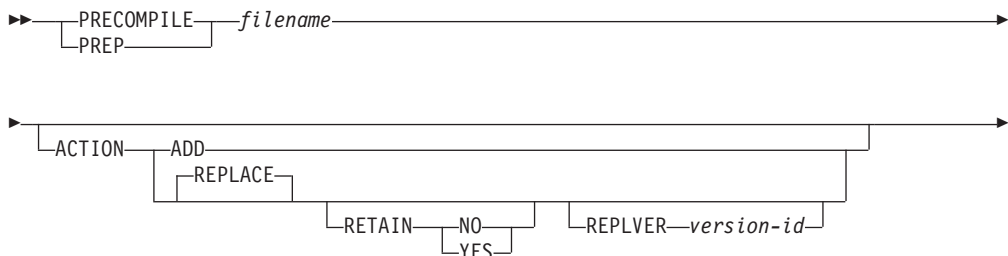
アプリケーション内の静的 SQL ステートメントをコンパイルするために必要な特権もすべて必要です。グループに認可された特権が、静的ステートメントの許可の検査に使用されることはありません。ユーザーに *sysadm* 権限があつてバインドを完了させる明示特権がない場合、データベース・マネージャは、明示的な *dbadm* 権限を自動的に付与します。

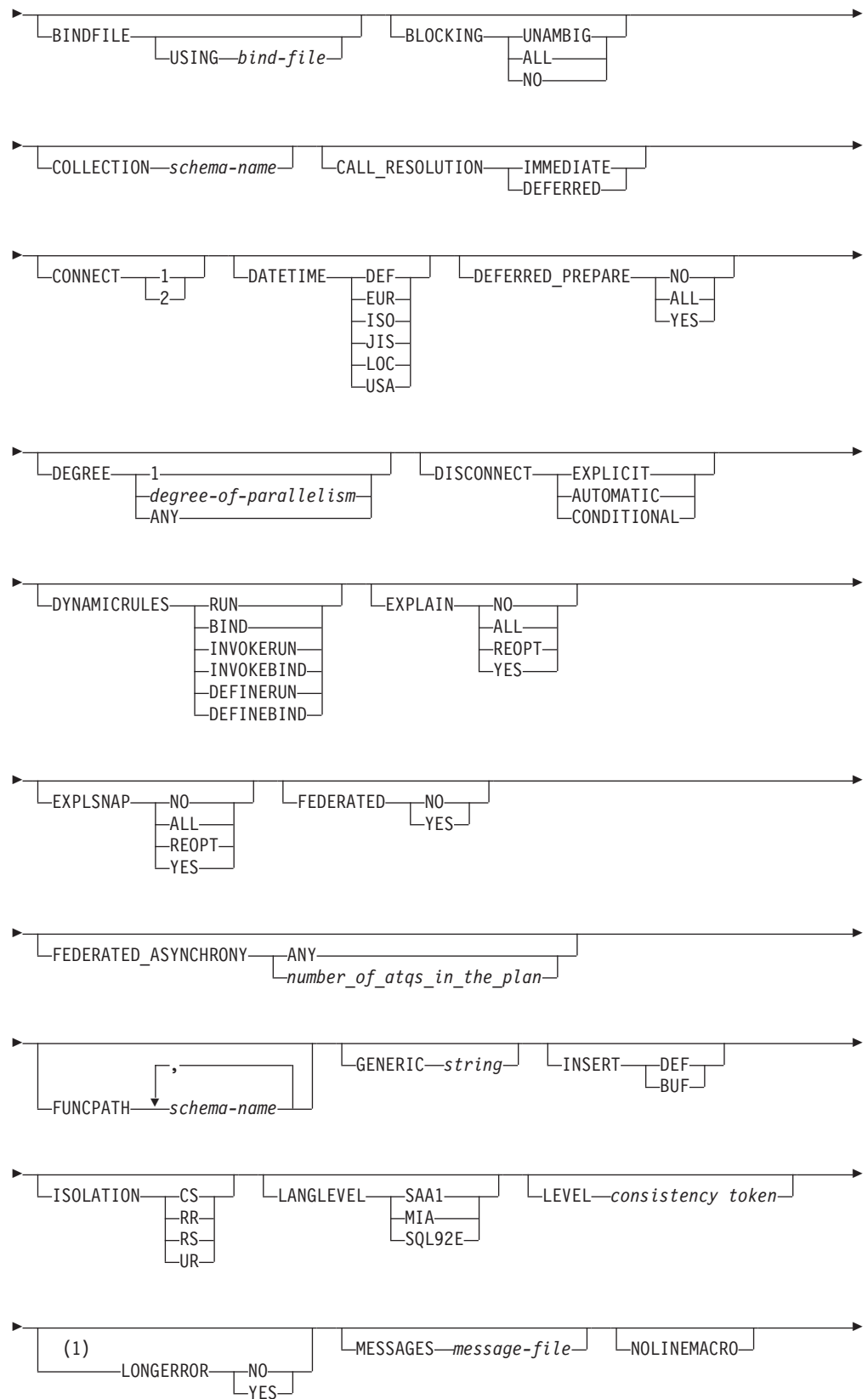
必要な接続

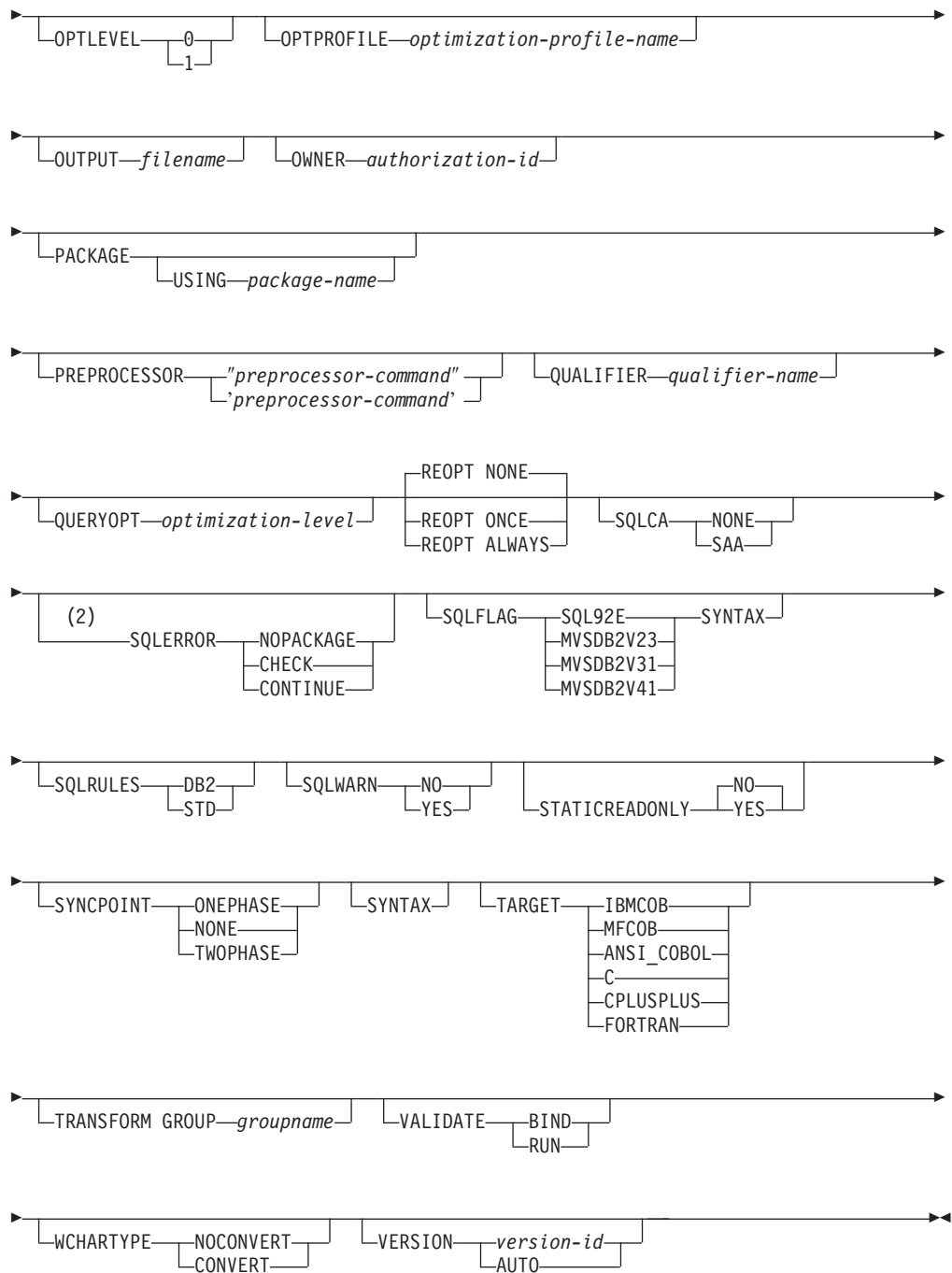
データベース。暗黙接続が可能な場合には、デフォルト・データベースへの接続が確立されます。

コマンド構文

DB2 Database for Linux, UNIX, and Windows では



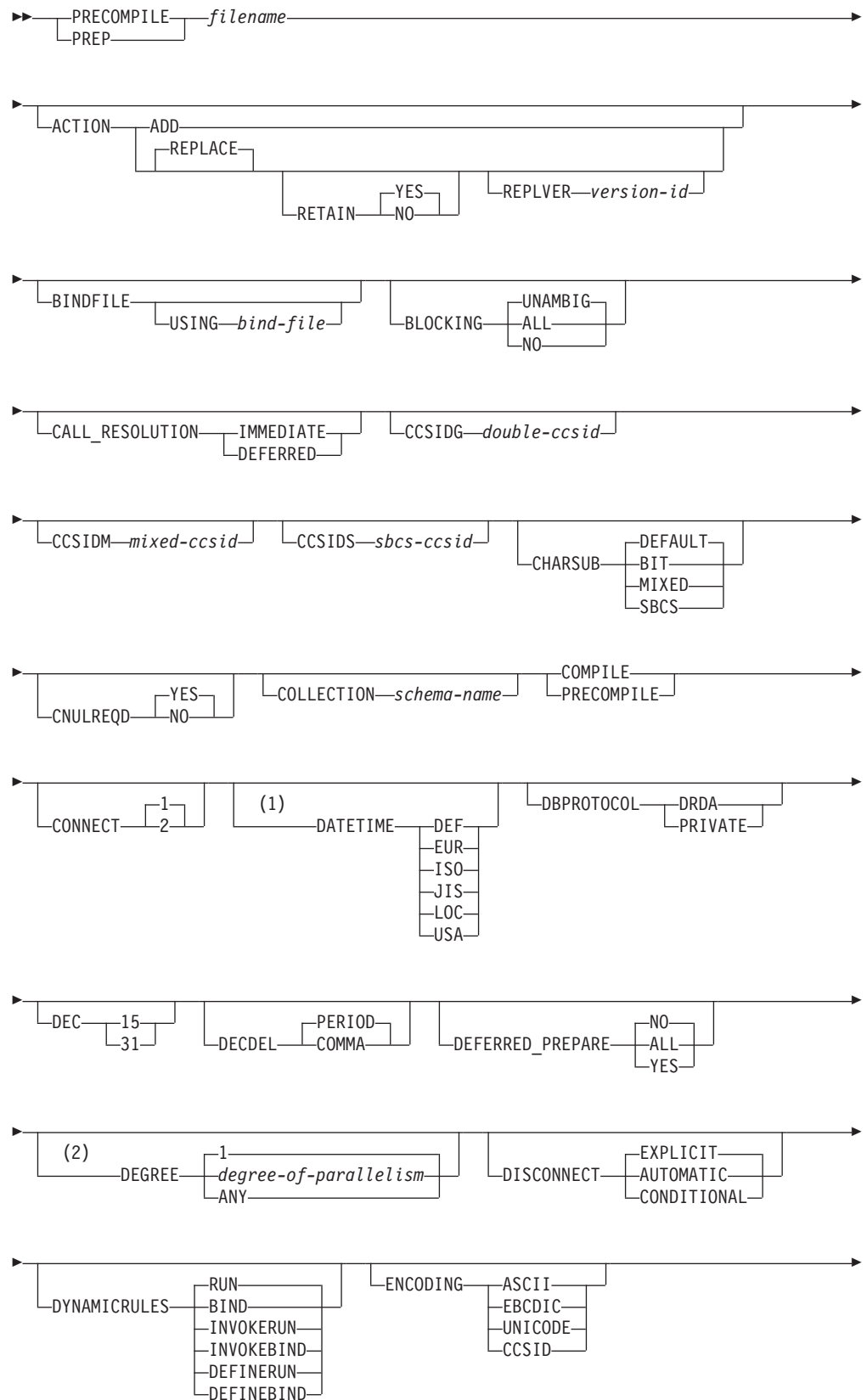


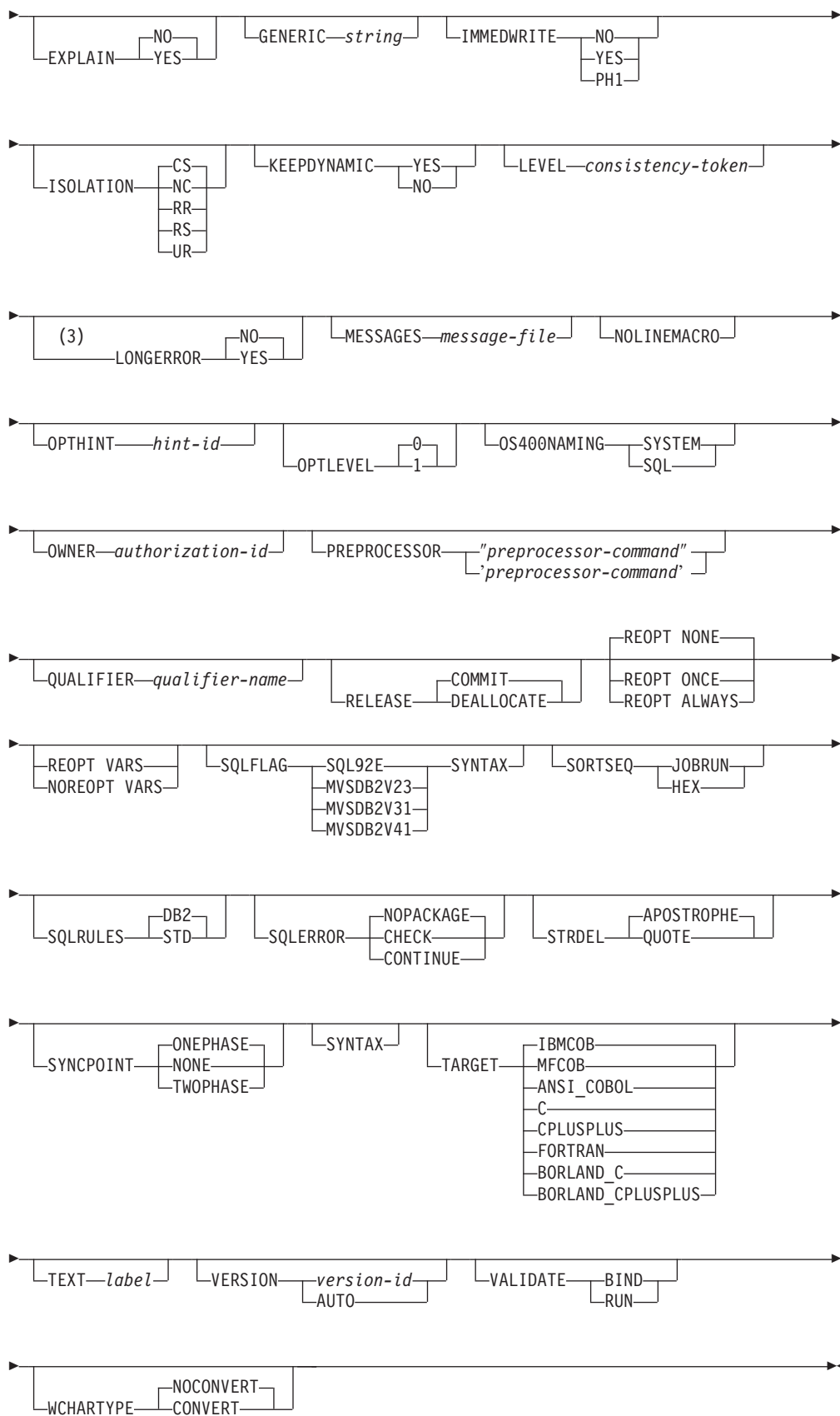


注:

- 1 NO は、32 ビット・システムと 64 ビット NT 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 および UNIX 以外のサーバー上の DB2 データベースでは





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 NT 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 (**.cbl** 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。

BINDFILE

バインド・ファイルが作成されます。**PACKAGE** オプションが共に指定されていない場合、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認証 **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。LOB 列データ・タイプへの参照が含まれる行データのブロック化は、データベース・パーティション・フィーチャー (DPF) が使用可能な環境で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ジャ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。なお、DB2 Database 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、システムが定義した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。なお、DB2 Database 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、システムが定義した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 Database 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、システムが定義した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。なお、DB2 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。

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・システムが定義したデフォルト・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、DRDA でサポートされていない **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。なお、DB2 はこの DRDA BIND オプションをサポートしません。

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。

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。なお、DB2 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。このオプションを指定しないと、DRDA サーバーは、システムが定義した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。なお、DB2 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。このオプションを指定しないと、DRDA サーバーは、システムが定義した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せん。

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、ただし、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 32767 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 とバインドし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** とバインドし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** とバインドし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** とバインドし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。DRDA では、このオプションの ALL 値がサポートされていません。

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 表に入れます。さらに、CURRENT EXPLAIN MODE 特殊レジスターが 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)。

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 特殊レジスターが NO に設定されていても、Explain 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)。

EXPLSNAP

Explain 表に Explain スナップショットを保管します。この DB2 プリコンパイル/ BIND オプションは、DRDA ではサポートされていません。

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 特殊レジスターが 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 特殊レジスターが 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)。

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す。

このオプションは DRDA サーバーではサポートされていません。

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 to 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。この DB2 プリコンパイル/ BIND オプションは、DRDA ではサポートされていません。

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。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

DB2 Enterprise Server Edition 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ていても、DRDA でサポートされていない新規 BIND オプションをサポートします。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。このオプションは、動的 SQL のパフォーマンスをかなり向上させることができます。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、ターゲット DRDA データベースが DB2 Universal Database バージョン 8 の場合、次のようにします。

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

これにより、**EXPLSNAP**、**QUERYOPT**、および **FEDERATED** の各オプションをバインドすることができます。

ストリングの最大長は 32768 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。

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。このオプションは DRDA サーバーではサポートされていません。

- 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、位置指定 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、位置指定 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なければ、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、位置指定 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)。

KEEPDYNAMIC

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。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。

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 NT 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が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を処理せず、名前が構造的に有効であることだけを検証します。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** オプションを使用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。SYSADM または 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。例えば、

```
x1c -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。この DB2 プリコンパイル/ BIND オプションは、DRDA ではサポートされていません。

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。なお、DB2 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。

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、グローバル変数、または特殊レジスターの実際の値によ

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 VARS | 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。

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 コンパイラから警告を戻しません。

YES SQL コンパイラ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。この DB2 プリコンパイル/ BIND オプションは、DRDA ではサポートされていません。

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。なお、DB2 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。このオプションを指定しないと、DRDA サーバーは、システムが定義した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。

SYNCPPOINT

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。なお、DB2 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。

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。このオプションは DRDA サーバーではサポートされていません。

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。
- 上記のいずれの条件もない場合には、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 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 yyy, 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 yyy, 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。

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いならばそうなります)、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 形式のデータが入れ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。

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い場合は、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。

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す。
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を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、静的および動的ステートメント・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・マネージャ構成パラメーターの値が特殊レジスタ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。

許可

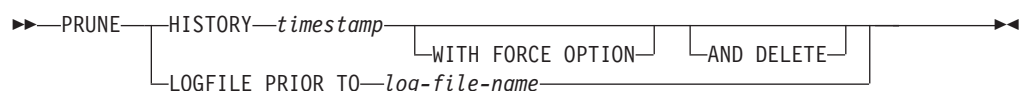
以下のいずれか。

- *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

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。

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・ファイル名を表す文字列 (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。 **logretain** データベース構成パラメーターは、RECOVERY または CAPTURE に設定する必要があります。

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限りなりません。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/proc1.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されます。"-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_DBPARTITIONNUM = CATALOG_DBPARTITIONNUM
ATTACH_DBPARTITIONNUM  = -1
```

CONNECT_DBPARTITIONNUM および ATTACH_DBPARTITIONNUM 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_DBPARTITIONNUM または ATTACH_DBPARTITIONNUM 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および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、静止モードにします。データベース・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および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。ただし、シャットダウンしたり、他のデータベースの開始を実行したりしないようにする必要があります。

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/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ユーザー/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。

許可

以下のいずれか。

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:

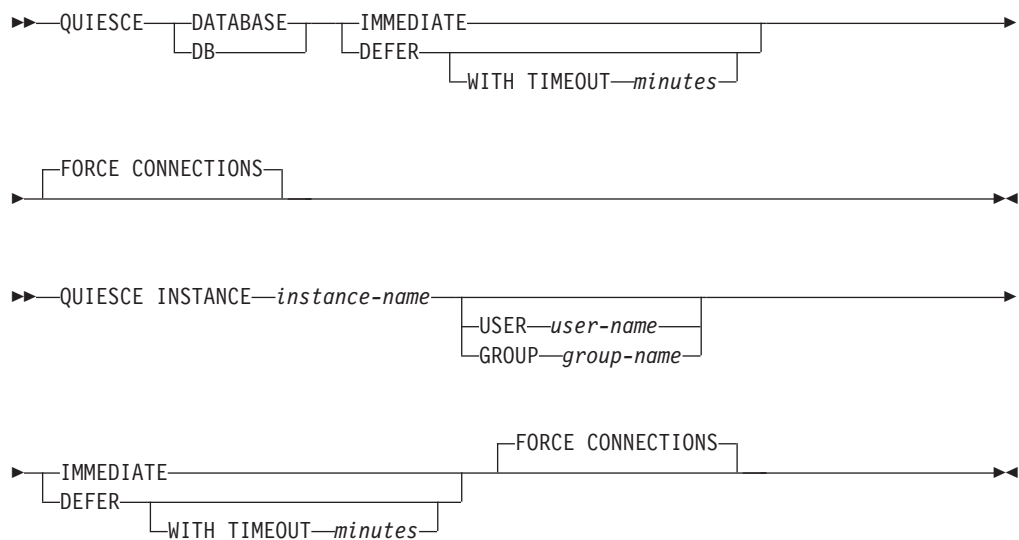
- *sysadm*
- *sysctrl*

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。

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。

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。
- QUIESCE DATABASE の後、*sysadm*、*sysmaint*、*sysctrl*、または *dbadm* 権限、および GRANT/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。

例えば、

```
grant quiesce_connect on database to <username/groupname>
revoke quiesce_connect on database from <username/groupname>
```

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、共用、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 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 がそれぞれ 'N'、'A'、または 'D') が静止されます。

許可

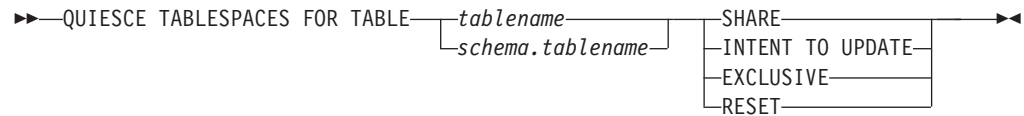
以下のいずれか。

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

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

静止が共用モードであることを指定します。

「静止モードでの共用」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用モードは、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。

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、宣言一時表に対して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、削除フェーズ中に停電によってロード操作が割り込まれると、ロードされていた表の表スペースは削除ペンディング、静止排他状態で残ります。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) アップグレードさせることができます。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す。つまり、状態がダウングレードされることはありません。

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・プロセスを終了しないか、データベース接続を中断します。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

以下のいずれか。

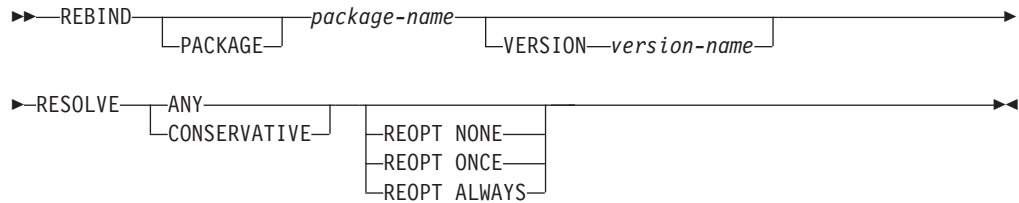
- *sysadm* または *dbadm* の権限
- スキーマに対する ALTERIN 特権
- パッケージに対する BIND 特権

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。

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。

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 は、RUNSTATS が実行された後に、新規統計を利用してパッケージを再作成するためにも使用することができます。
- 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・マイグレーションの後、データベースに格納されているすべてのパッケージが、MIGRATE DATABASE コマンドによって無効にされます。これは多数のパッケ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 を使用するようになっています。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 BIND オプションのいずれかを変更したい場合。REBIND は BIND オプションをサポートしていません。例えば、バインド・プロセスの一部として付与された、パッケージに対する特権をユーザーが所有したい場合、GRANT オプションが用意されているのは BIND なので、それ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 文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 は、EXPLSNAP BIND オプションが YES または ALL に設定されて作成されたパッケージ (値の設定は、SYSCAT.PACKAGES カタログ表項目の EXPLAIN_SNAPSHOT 列で示されます)、あるいは EXPLAIN BIND オプションが YES または ALL に設定されて作成されたパッケージ (値の設定は、SYSCAT.PACKAGES カタログ表項目の EXPLAIN_MODE 列で示されます) に再び Explain を実行します。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 ステートメントがエラーで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、問題が訂正されたとしても、そのステートメントには無効というマークが付けられます。REBIND しても、ステートメント

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では、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または権限の問題があるかどうかに応じて、REBIND を通じて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 ALWAYS を指定してパッケージを再バインドすると、静的および動的ステートメント・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 はプリコンパイル時またはバインド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。

許可

既存のデータベースにリカバリーするには、次のいずれかが必要です。

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

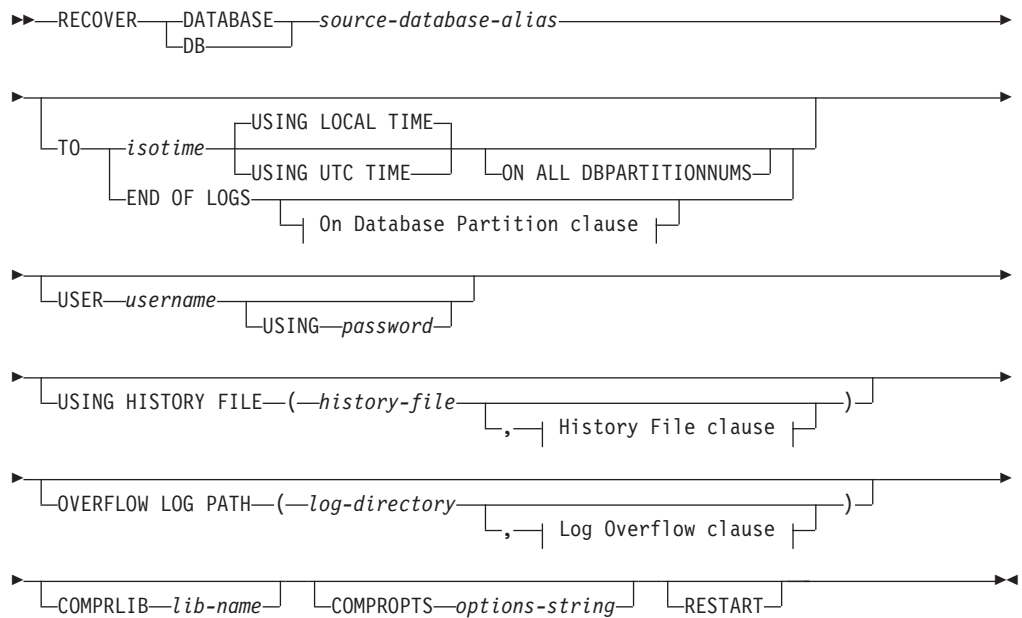
新規のデータベースにリカバリーするには、次のいずれかが必要です。

- *sysadm*
- *sysctrl*

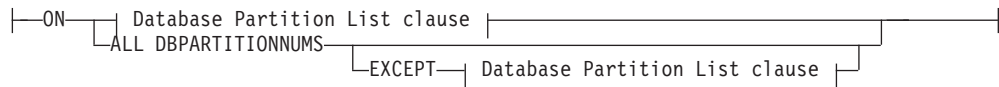
必要な接続

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。

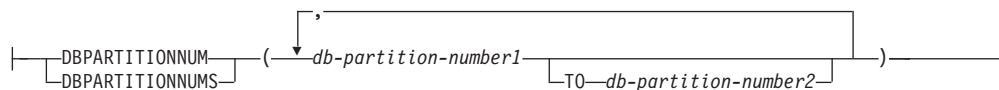
コマンド構文



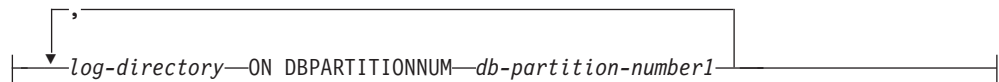
On Database Partition clause:



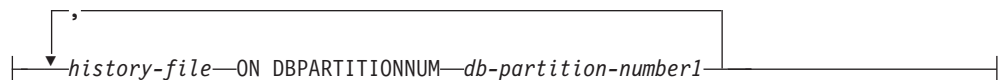
Database Partition List clause:



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。

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。

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、ちょうどその時点に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。

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。これは、クライアントのローカル時刻となるコントロール・センターのローカル時刻オプションとは異なります。
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。

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

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。

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。

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。

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 RECOVER 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。V9.1 からは、後続の RECOVER コマンドが、可能であれば前の RECOVER を継続しようとしています。

RESTART キーワードを使用すると、RECOVER にフレッシュ・リストアを開始してから、指定された PIT にロールフォワードするように強制します。

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。

例

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現在存在しており、履歴ファイルの最新バージョンが **dftdbpath** で使用できる場合、

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 PIT (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 PIT 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。履歴ファイルの保管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 PIT 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管している場合、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。

1. その履歴ファイルのバックアップを作成していないため、バックアップ・イメージ内のコピーしか利用できない場合には、**ROLLFORWARD** の後に **RESTORE**

を発行することをお勧めします。しかし、RECOVER を使用するためには、履歴ファイルをイメージからどこかの場所 (例えば /home/user/oldfiles/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 RECOVER 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 PIT
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、

- すべてのノードでデータベースを特定の PIT までリカバリーする場合。DB2 は、リストア操作を開始する前に、その PIT がどのノード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
```

- すべてのノードでデータベースをこの PIT までリカバリーする場合。DB2 は、リストア操作を開始する前に、その PIT がどのノードでも到達可能であるかどうかを検証します。各ノードでの RECOVER 操作は、単一パーティション RECOVER の場合と同一です。

```
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。例外はノード 2 と 4 です。ノード 2 では /home/user/node2files/db2rhist.asc が使用され、ノード 4 では /home/user/node4files/db2rhist.asc が使用されます。

```
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)
```

- すべてのノードではなく一部のノードをリカバリーすることも可能です。しかし、その場合は PIT RECOVER は実行できず、リカバリーは EOL 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、

1. その履歴ファイルのバックアップを作成していないため、バックアップ・イメージ内のコピーしか利用できない場合には、ROLLFORWARD の後に RESTORE を発行することをお勧めします。しかし、RECOVER を使用するためには、履歴ファイルをイメージからどこかの場所 (例えば /home/user/oldfiles/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 PIT
USING HISTORY FILE (/home/user/fromimage/db2rhist.asc)
```

2. 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。

REDISTRIBUTE DATABASE PARTITION GROUP

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・パーティション間でデータを再配分します。データの配分先は、均一 (デフォルト) にすることも、特定のシステム要件にかなうようにユーザー指定にすることもできます。

REDISTRIBUTE DATABASE PARTITION GROUP 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パーティション間でデータを再配分します。

これはデータベース・パーティション・グループ内に存在するすべてのオブジェクトに影響を与え、1つのオブジェクトのみに制限することはできません。

このコマンドは、カタログ・データベース・パーティションからしか発行できません。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

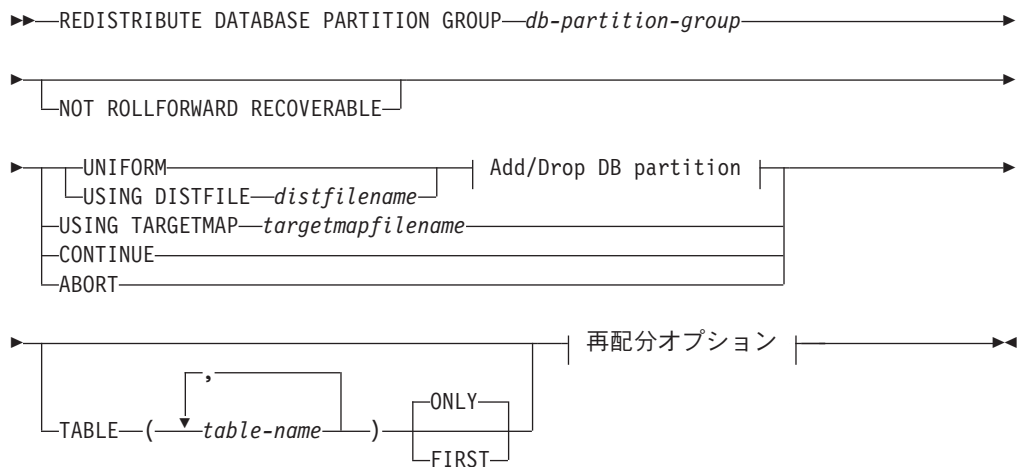
このコマンドは、データベース・パーティション・グループ内のすべてのデータベース・パーティションに影響を与えます。

許可

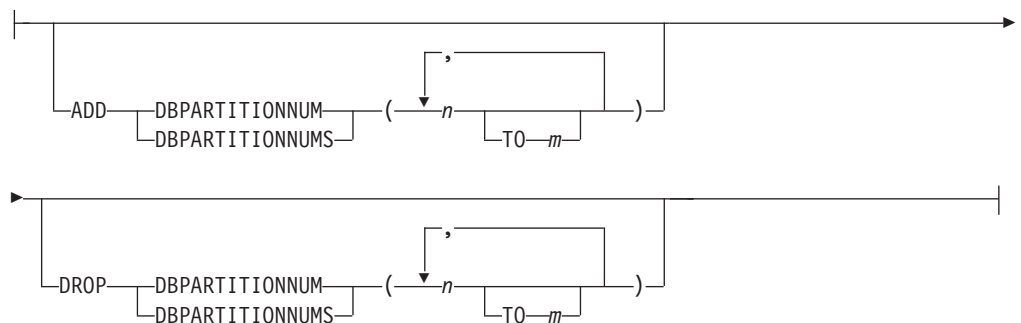
以下のいずれか。

- SYSADM
- SYSCtrl
- DBADM

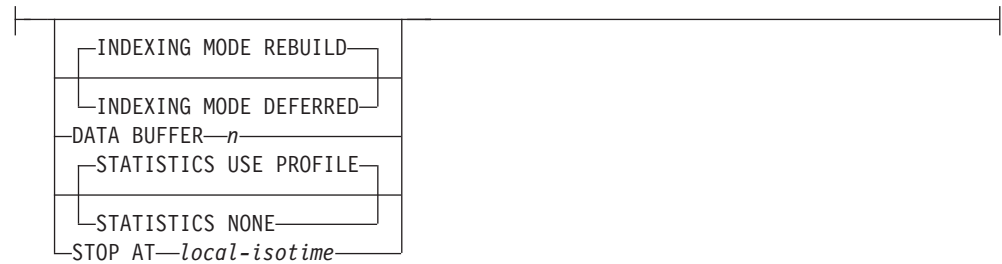
コマンド構文



Add/Drop DB partition:



再配分オプション:



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、現在再配布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

このオプションは、DB2 9.5 フィックスパック 1 をインストールしないと、使用できません。このオプションを使用する場合、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、バルク操作で移動されます。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ごとのログ・レコードが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。過去に、大容量のアクティブ・ログ・スペースとストレージの要件のために、単一のデータ再配分操作を複数の小規模の再配分タスクに分割することを強制され、その結果、エンドツーエンドのデータ再配分操作にさらに時間が要求されていた場合には、これは特に有益です。

このオプションを使用しない場合、すべての行移動の詳細なロギングが実行されるので、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、データベースを後でリカバリーすることができます。

UNIFORM

データがハッシュ・パーティションにわたって均等に分散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配分を行います。

distfilename を使用して、4096 個のハッシュ・パーティションにおける現行のデータの分散を示します。

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 (重み 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、4096 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、4294967295 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は、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。

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場合、エラーが戻されます。

ALTER DATABASE PARTITION GROUP ADD DBPARTITIONNUM ステートメントを発行してから、REDISTRIBUTE DATABASE PARTITION GROUP コマンド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ません。ALTER DATABASE PARTITION GROUP DROP DBPARTITIONNUM ステートメントを REDISTRIBUTE DATABASE PARTITION GROUP コマンドの前か後のいずれかで使用して、このようなデータベース・パーティションをドロップすることができます。

CONTINUE

直前に失敗または停止した REDISTRIBUTE DATABASE PARTITION GROUP 操作を続行します。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 操作を打ち切ります。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* は、データベース・パーティション・グループに追加されるデータベース・パーティション番号のリスト (1 つまたは複数) を指定します。ここで指定され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 (SQLSTATE 42728) です。すでに定義されているものであってはなりません。これは、ALTER DATABASE PARTITION GROUP ステートメントに ADD DBPARTITIONNUM 節を指定して実行する場合も同様です。

DBPARTITIONNUMS *n*

TO *m*

n または *n* TO *m* は、データベース・パーティション・グループに追加されるデータベース・パーティション番号のリスト (1 つまたは複数) を指定します。ここで指定され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 (SQLSTATE 42728) です。すでに定義されているものであってはなりません。これは、ALTER DATABASE PARTITION GROUP ステートメントに ADD DBPARTITIONNUM 節を指定して実行する場合も同様です。

注: このオプションを使用してデータベース・パーティションが追加される場合、表スペースのコンテナは、データベース・パーティション・グループ内の最も低い番号が付いた既存のパーティション上にある対応する表スペースのコンテナに基づきます。これによりコンテナ間で名前の競合が起きる場合、このオプションを使用しないでください (新規パーティションが既存のコンテナと同じ物理マシン上にある場合は、名前の競合が発生する可能性があります)。その代わりに、ALTER DATABASE PARTITION GROUP ステートメントを WITHOUT TABLESPACES オプションとともに使用してから、REDISTRIBUTE DATABASE PARTITION GROUP コマンドを発行してください。その後、適切な名前を手動で指定して、表スペース・コンテナを作成できます。

DROP

DBPARTITIONNUM *n*

TO *m*

n または *n* TO *m* は、データベース・パーティション・グループからドロップされるデータベース・パーティション番号のリスト (1

つまたは複数) を指定します。ここで指定され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 (SQLSTATE 42729) です。すでに定義されていなければなりません。これは、ALTER DATABASE PARTITION GROUP ステートメントに DROP DBPARTITIONNUM 節を指定して実行する場合も同様です。

DBPARTITIONNUMS *n*

TO *m*

n または *n TO m* は、データベース・パーティション・グループからドロップされるデータベース・パーティション番号のリスト (1 つまたは複数) を指定します。ここで指定され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 (SQLSTATE 42729) です。すでに定義されていなければなりません。これは、ALTER DATABASE PARTITION GROUP ステートメントに DROP DBPARTITIONNUM 節を指定して実行する場合も同様です。

TABLE *tablename*

再配分処理する表の順番を指定します。

ONLY 表順序の後に **ONLY** キーワードが続く (これがデフォルトです) 場合は、指定された表のみが再配分されます。残りの表は、後続の REDISTRIBUTE CONTINUE コマンドで後から処理できます。これはデフォルトです。

FIRST 表順序の後に **FIRST** キーワードが続く場合は、指定された表が特定の順序に従って再配分され、データベース・パーティション・グループに残っている表はランダムな順序で再配分されます。

INDEXING MODE

NOT ROLLFORWARD RECOVERABLE オプションが指定されている場合、このパラメーターで、再配分の際に索引を維持する方法を指定します。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。このような索引に最初にアクセスすると、再作成が強制されたり、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。

注: 非 MDC 表の場合、表に無効な索引があると、INDEXING MODE DEFERRED が指定されていなければ、**REDISTRIBUTE DATABASE PARTITION GROUP** コマンドが自動的に索引を再作成します。MDC 表の場合、INDEXING MODE DEFERRED が指定されている場合でも、表の再配分が始まる前に無効な複合索引が再作成されます。なぜなら、ユーティリティーは MDC 表を処理するために複合索引を必要とするからです。

DATA BUFFER *n*

ユーティリティー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。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。このメモリーは、ユーティリティー・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。値が指定されていない場合、実行時にユーティリティーによって適切なデフォルトが各表の処理の初めに計算されます。具体的に言うと、デフォルトは、表の再配分の開始時にユーティリティー・ヒープで使用可能なメモリーの 50% を使用し、さらに様々な表プロパティーも考慮することです。

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、*local-isotime* が現在のローカル・タイム・スタンプと比較されます。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、ユーティリティーは停止して、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、タイム・スタンプとして指定されます。これは、組み合わせられた日付と時刻を示す 7 つの部分からなる文字ストリングです。形式はローカル時間の `yyyy-mm-dd-hh.mm.ss.nnnnnn` (年、月、日、時、分、秒、マイクロ秒) で表されます。

STATISTICS

このオプションは、ユーティリティーで統計プロファイルを持つ表の統計を収集するよう指定します。データ再配分が完了した後から `RUNSTATS` コマンドを個別に発行するよりも、このオプションを指定した方が効率的です。

USE PROFILE

統計プロファイルを持つ表の統計が収集されます。統計プロファイルのない表では何も行われません。これはデフォルトです。

NONE 表の統計は収集されません。

例: 再配分の手順

ノード・グループからノードを追加またはドロップするとします。新規ノードをノード・グループに追加してデータを再配分する手順を以下に示します。追加されたデータベース・パーティションは分散マップにありませんが、データベース・パーティション・グループ内の表スペース用のコンテナは作成されています。データベース・パーティション・グループの再配分操作が正常に完了すると、データベース・パーティションは分散マップに追加されます。

1. 再配分する必要があるノード・グループを確認します。本書では、再配分する必要があるノード・グループを `sampleNodegrp` という名前にします。
2. 再配分する前に使用不可にする必要がある、または削除する必要があるオブジェクトを確認します。
 - a. 複製 MQT: このタイプの MQT は `REDISTRIBUTE` ユーティリティーの一部としてサポートされていません。再配分を実行する前にドロップし、後ほど再作成する必要があります。

```
SELECT tabschema, tabname FROM syscat.tables WHERE partition_mode = 'R'
```

- b. 表書き込みイベント・モニター: 自動開始する表書き込みイベント・モニターで、再配分するノード・グループにその表があるものがあれば、それらのイベント・モニターは無効にする必要があります。

```
SELECT distinct evmonname FROM syscat.eventtables E
JOIN syscat.tables T on T.tabname = E.tabname AND T.tabschema = E.tabschema
JOIN syscat.tablespaces S on S.tbspace = T.tbspace AND S.ngname = 'sampleNodegrp'
```

- c. Explain 表: Explain 表は単一パーティション・ノード・グループ内に作成することをお勧めします。しかし、再配分を必要とするノード・グループ内で定義される場合、今までに生成されたデータを維持する必要がないなら、再配分する前にそれらをドロップして、再配分が完了した時に再定義することも可能です。

- d. 表アクセス・モードおよびロード状態: 再配分するノード・グループ内のすべての表がフル・アクセス・モードであること、および、ロード・ペンディングまたはロード進行中状態でないことを確認してください。

```
SELECT distinct trim(T.creator) || '\.' || trim(T.name) as name, T.access_mode, A.load_status
FROM sysibm.systables T, sysibm.sysnodegroups N, sysibmadm.admintabinfo A
WHERE T.pmap_id = N.pmap_id
AND A.tabschema = T.creator
AND A.tabname = T.name
AND N.name = 'sampleNodegrp'
AND (T.access_mode <> 'F' or A.load_status is not null)
```

- e. 統計プロファイル: 統計プロファイルが表に定義されている場合、表統計は再配分プロセスの一環として更新することができます。REDISTRIBUTE ユーティリティを使用して表統計を更新すると、データすべてが再配分のためにスキャンされて RUNSTATS 用のデータの追加スキャンが必要なくなるため、入出力を削減できます。

```
RUNSTATS on table schema.table USE PROFILE runstats_profile SET PROFILE ONLY
```

3. データベース構成を確認します。**util_heap_sz** は、データベース・パーティション間のデータ移動処理にとって重要です。再配分時のために **util_heap_sz** にできるだけ多くのメモリーを割り振ってください。索引の再作成が再配分の一環として行われる場合、十分な **sorthheap** が必要です。**util_heap_sz** および **sorthheap** を必要に応じて増やして、再配分のパフォーマンスを向上させてください。
4. 新規データベース・パーティションに使用されるデータベース構成設定を取得します。データベース・パーティションを追加する時は、デフォルトのデータベース構成が使用されます。そのため、REDISTRIBUTE コマンドを発行する前に新規ノード上でデータベース構成を更新することにより、ウェアハウス全体が均衡のとれた構成になるようにすることが重要です。

```
SELECT name,
CASE WHEN deferred_value_flags = 'AUTOMATIC' THEN deferred_value_flags ELSE substr(deferred_value,1,20) END as deferred_value
FROM sysibmadm.dbcfg
WHERE dbpartitionnum = existing-node
AND deferred_value != ''
AND name not in ('hadr_local_host','hadr_local_svc','hadr_peer_window',
'hadr_remote_host','hadr_remote_inst','hadr_remote_svc',
'hadr_syncmode','hadr_timeout','backup_pending',
'codepage','codeset','collate_info','country',
'database_consistent','database_level',
'hadr_db_role','log_retain_status',
'loghead','logpath','multipage_alloc','numsegs',
'pagesize','release','restore_pending','restrict_access',
'rollfwd_pending','territory','user_exit_status','number_compat',
'varchar2_compat','database_memory')
```

5. 再配分を開始する前に、データベース (または再配分するノード・グループ内の表スペース) をバックアップして、最新のリカバリー・ポイントを確保します。
6. db2nodes.cfg ファイルを更新して新規データ BCU データベース・パーティションの指定を追加することにより、新規データ BCU を DB2 に定義します。そして ADD NODE WITHOUT TABLESPACES コマンドを使用して、新規のデータベース・パーティションを DB2 に定義します。

```
db2start nodenum x export DB2NODE=x
db2 add node without tablespaces
db2stop nodenum x
```

注: それがデータ BCU 上で最初の論理ポートではない場合は、後続の論理ポートのために、上記の一連のコマンドを実行する前と後に、最初の論理ポート番号の始動および停止を実行します。

7. 新たに定義されたデータベース・パーティションに SYSTEM TEMPORARY 表スペース・コンテナを定義します。

```
ALTER TABLESPACE tablespace_name ADD container_information ON dbpartitionnums (x to y)
```

8. 新規の論理データベース・パーティションを複数のデータ BCU にまたがるデータベース・パーティション・グループに追加します。

```
ALTER DATABASE PARTITION GROUP partition_group_name ADD dbpartitionnums (x to y) WITHOUT TABLESPACES
```

9. 新たに定義されたデータベース・パーティションに永続データの表スペース・コンテナを定義します。

```
ALTER TABLESPACE tablespace_name ADD container_information ON dbpartitionnums (x to y)
```

10. ステップ 4 で取得したデータベース構成設定を新規のデータベース・パーティションに適用します。(または、新規の DB2 9.5 の構成サポートの単一ビューを使用して、すべてのデータベース・パーティションに対して単一の UPDATE DB CFG コマンドを発行します。)

11. 再配分するデータベース・パーティション・グループに複製 MQT がある場合、それらの定義をキャプチャーしてから、複製 MQT をドロップします。

```
db2look -d dbname -e -z schema -t replicated_MQT_table_names -o repMQTs.clp
```

12. 再配分するデータベース・パーティション・グループにある表書き込みイベント・モニターを使用不可にします。

```
SET EVENT MONITOR monitor_name STATE 0
```

13. REDISTRIBUTE ユーティリティを実行してすべてのデータベース・パーティションに均等に再配分します。以下に単純な再配分コマンドを示します。

```
REDISTRIBUTE DATABASE PARTITION GROUP sampleNodegrp NOT ROLLFORWARD RECOVERABLE uniform;
```

また、REDISTRIBUTE コマンドへの入力として表リストを指定することによって、表が処理される順序を強制することも考慮する必要があります。REDISTRIBUTE ユーティリティはデータを移動させます (圧縮および短縮)。オプションで、索引の再作成と、統計プロファイルが定義されている場合は統計の更新とが行われます。そのため、以前のコマンドの代わりに次のスクリプトを実行できます。

```
REDISTRIBUTE DATABASE PARTITION GROUP sampleNodegrp
NOT ROLLFORWARD RECOVERABLE uniform TABLE (tab1, tab2,...) FIRST;
```

NOT ROLLFORWARD RECOVERABLE オプション使用の影響

REDISTRIBUTE DATABASE PARTITION GROUP コマンドが発行されて NOT ROLLFORWARD RECOVERABLE オプションが指定された場合、移動される各行のログ・レコードの書き込みを最小にする、最小限のロギング方式が使用されます。このタイプのロギングは、再配分操作のユーザビリティにとって重要です。なぜなら、大規模なシステムの場合、すべてのデータ移動を完全に記録する方法では、非現実的な量のアクティブ・ログ・スペースと永続ログ・スペースを必要とする可能性があり、一般にはパフォーマンス特性の低下をもたらすからです。しかし、この最小限のロギング・モデルの結果として、REDISTRIBUTE DATABASE PARTITION GROUP コマンドがロールフォワード・リカバリー可能ではないことをユーザーが認識しておくことは重要です。これは、再配分操作の結果としてデータベースがロールフォワードすることになる操作では、再配分操作の影響を受けるすべての表が UNAVAILABLE 状態のままになるという意味です。そうした表はドロップのみが可能です。つまり、それらの表の中のデータをリカバリーする方法はないということです。したがって、リカバリー可能なデータベースについては、REDISTRIBUTE DATABASE PARTITION GROUP ユーティリティが、NOT ROLLFORWARD RECOVERABLE オプションを指定して発行されると、対象になるすべての表スペースを BACKUP PENDING 状態に設定し、成功した再配分操作の最後に、再配分を受けたすべての表スペースのバックアップをユーザーが実行するように強制しています。再配分操作の後にバックアップが取られると、ユーザーは再配分操作自体によってロールフォワードする必要がなくなるはずですが、

再配分操作ユーティリティにロールフォワード・リカバリー可能性がない場合の重大な結果の 1 つとしてユーザーが知っておくべきことは、次のとおりです。再配分操作の実行中 (再配分の影響を受ける表スペースがユーザーによってバックアップされている再配分の終わりの期間を含む) にデータベース内の表 (再配分されるデータベース・パーティションの外の表であっても) に対する更新を許可することをユーザーが選択した場合、重大な障害が発生すると (例えばデータベース・コンテナが破壊されるなど)、そうした更新は失われることがあります。そうした更新が失われる理由は、再配分操作がロールフォワード・リカバリー可能でないからです。再配分操作の前に取られたバックアップからデータベースをリストアする必要がある場合、再配分操作中になされた更新を再生するためにログによるロールフォワードを実行するには、すでに説明したとおり、再配分された表が UNAVAILABLE 状態になる原因となった再配分操作についても、ロールフォワードする必要があります。そのため、この状況で行える唯一の事柄は、ロールフォワードしないで再配分する前にバックアップからデータベースをリストアすることです。その後、再配分操作を再度実行できます。残念ながら、元の再配分操作中に発生した更新はすべて失われます。

この点の重要性は強調し過ぎることはありません。再配分操作中に更新が絶対に失われないようにするには、以下のいずれかが該当しなければなりません。

- ユーザーが REDISTRIBUTE DATABASE PARTITION GROUP コマンドの操作中 (コマンドの終了後、影響を受ける表スペースがバックアップされている期間を含む) に更新を行わないようにします。
- 再配分操作中に適用される更新は反復可能ソースに由来します。つまり、そうした更新をいつでも再び適用できるという意味です。例えば、更新のソースがファイルに保管されたデータであり、更新がバッチ処理中に適用される場合、データ

ベースのリストアを要する障害が明らかに発生した場合でも、更新が失われることはありません。更新をいつでも再び適用すればよいからです。

再配分操作中にデータベースへの更新を許可することについて、ユーザーはそうした更新が該当するシナリオに適切かどうかを、データベースのリストア後に必要に応じて更新を反復できるかどうかに基づいて決定する必要があります。

注: REDISTRIBUTE DATABASE PARTITION GROUP コマンドの操作中のすべての障害でこの問題が発生するわけではありません。実際、ほとんどの場合は発生しません。 REDISTRIBUTE DATABASE PARTITION GROUP コマンドは十分に再始動可能です。これは、ユーティリティーが作業の途中で失敗する場合、**CONTINUE** または **ABORT** オプションを指定して容易に続行したりアボートすることが可能という意味です。前に言及した失敗とは、再配分操作の前に取られたバックアップからユーザーがリストアしなければならない失敗のことです。

使用上の注意

- **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データベースである場合、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。その表スペースにあるすべての表は、表スペースがバックアップされるまで読み取り専用になります。バックアップは、表スペースのすべての表の再配分が終了したときだけ行われます。
- 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、`homeinst/sqllib/redist` ディレクトリー。サブディレクトリーとファイル名については、`database-name.database-partition-group-name.timestamp.log` の形式が使用されます。
 - Windows オペレーティング・システムの場合は、**DB2INSTPROF**`¥instance¥redist` 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。

再配分イベント・ログについて詳しくは、『再配分イベント・ログ・ファイル』のトピックを参照してください。

- 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。
- デフォルトでは、再配分ユーティリティーが、統計プロファイルを持つ表の統計を更新します。統計プロファイルのない表の場合は、`db2Runstats` API を呼び出す

か、または再配分操作完了後に `RUNSTATS` コマンドを発行するかして、表とそれらの表の索引の統計を個別に更新することをお奨めします。

- 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、宣言された既存の一時表を含むユーザー `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、表では無視され適用されませんが、警告は出ません。例えば、`INDEXING MODE` は索引のない表では無視されます。
- 再配分操作を開始する前に、どの表もロード・ペンディング状態になっていないことを確認します。表の状態は、`LOAD QUERY` コマンドを使用してチェックすることができます。

互換性

バージョン 8 より前のバージョンとの互換性：

- キーワード `NODEGROUP` は、`DATABASE PARTITION GROUP` に置き換えられます。

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。

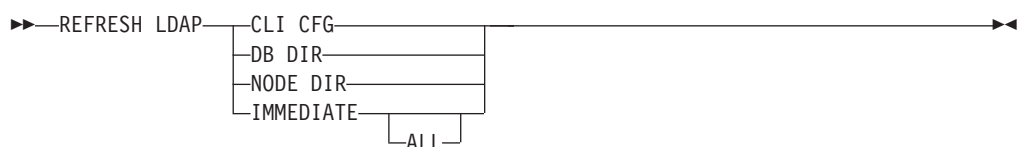
許可

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。データベースまたはノード項目は、ユーザーがデータベースに接続したり、LDAP で検出されるインスタンスにアタッチし、DB2LDAPCACHE が未設定または YES に設定されている場合に、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 が使用可能が否かに関係なく、即座に行われます。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 が使用可能が否かに関係なく実行されます。

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

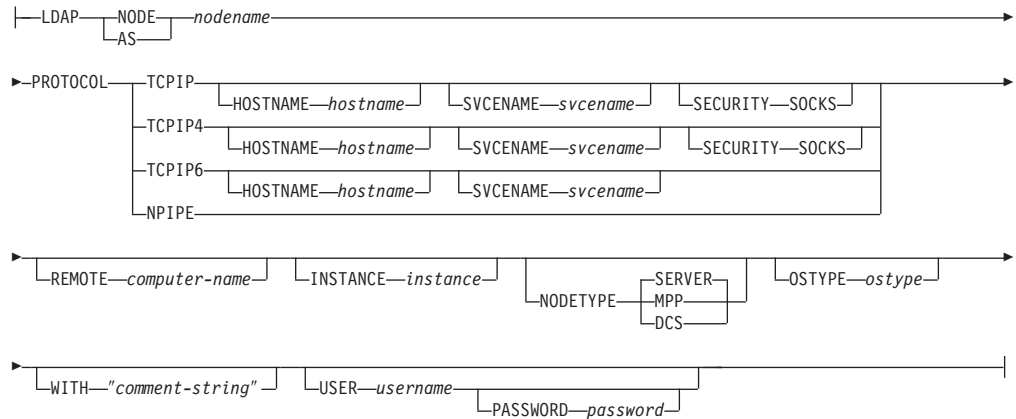
必要な接続

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 *svcname*

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ってください。

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 がそれぞれ使用されます。

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・サーバーに登録する場合には、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・リポジトリー (XSR) に登録します。

許可

以下のいずれか。

- *sysadm* または *dbadm*
- SQL スキーマが存在しない場合は IMPLICIT_SCHEMA データベース権限
- SQL スキーマが存在する場合は CREATEIN 特権

必要な接続

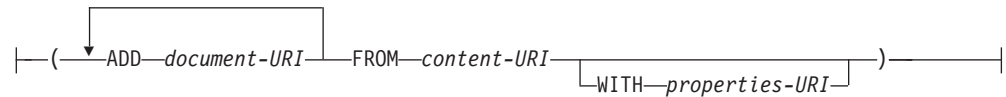
データベース

コマンド構文

```
▶▶ REGISTER XMLSCHEMA—schema-URI—FROM—content-URI————→  
▶┌—————┐┌—————┐  
  WITH—properties-URI  AS—relational-identifier
```



xml-document-subclause:



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です。

許可

以下のいずれか。

- *sysadm* または *dbadm*
- SQL スキーマが存在しない場合は IMPLICIT_SCHEMA データベース権限
- SQL スキーマが存在する場合は CREATEIN 特権

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に C:¥TEMP に保管されるように、エンティティを登録します。

```
REGISTER XSRBJECT '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 定義がローカルに C:¥TEMP に保管され、DTD に関連付けられるリレーショナル ID が "TEST.SUBJECTS" になるように、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に C:¥TEMP に保管されるように、エンティティを登録します。

```
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。

索引データを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再作成することによって、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。あるいは、範囲パーティション表上の特定の索引を再編成するという方法もあります。

索引節の CLEANUP ONLY 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。

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。

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

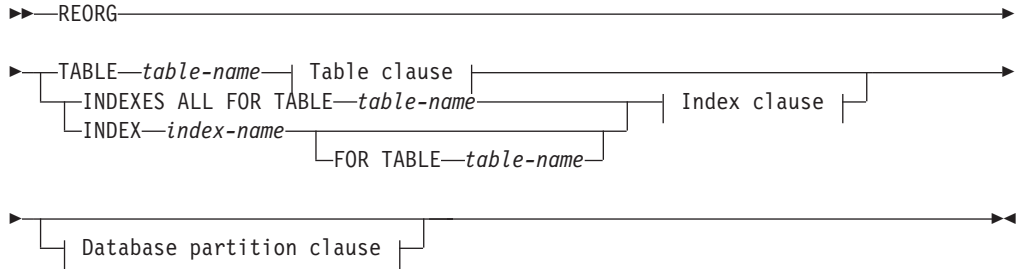
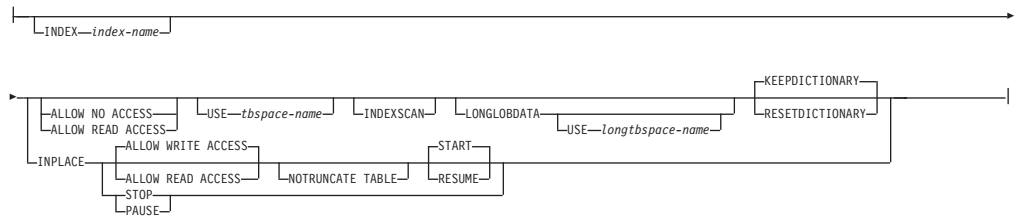
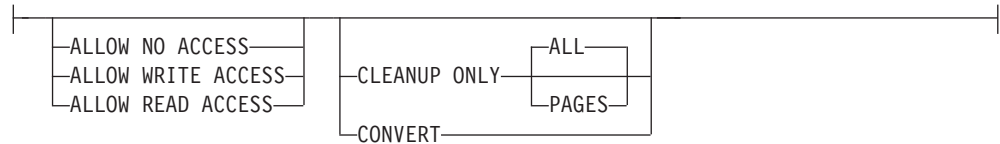


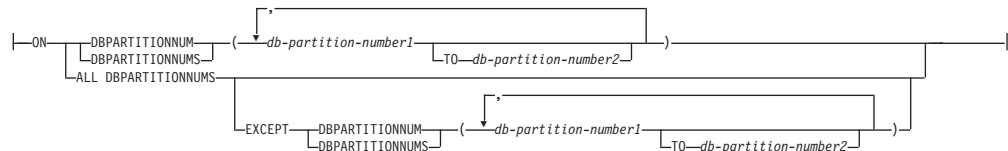
Table clause:



Index clause:



Database partition 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*

パーティション表に対して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*

パーティション表について再編成する個別索引の表名のロケーションを指定します。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

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

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、読み取り専用のアクセスを行うことができることを指定します。

CLEANUP ONLY オプションが指定されているのでない限り、パーティション表の REORG INDEXES に関してこのアクセス・レベルはサポートされていません。

ALLOW WRITE ACCESS

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込んだりそこに書き込んだりできることを指定します。 CLEANUP ONLY オプションが指定されているのでない限り、このアクセス・レベルは、マルチディメンション・クラスター (MDC) 表、パーティション表、拡張索引、または XML データ・タイプの列を含む表ではサポートされていません。

ACCESS モードが何も指定されていない場合は、以下のようにして自動的に選択されます。

表 30. コマンド、表タイプ、および *index* (索引) 節に指定されたその他のパラメーターに基づいて選択されるデフォルトの表アクセス

コマンド	表タイプ	<i>index</i> (索引) 節に 指定された その他のパラメーター	デフォルトの アクセス・モード
REORG INDEXES	非パーティション表	any	ALLOW READ ACCESS
REORG INDEXES	パーティション表	指定なし	ALLOW NO ACCESS
REORG INDEXES	パーティション表	CLEANUP ONLY が指定 されている	ALLOW READ ACCESS
REORG INDEX	パーティション表	any	ALLOW READ ACCESS

CLEANUP ONLY

CLEANUP ONLY が要求されると、完全な再編成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。

CLEANUP ONLY 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。

CLEANUP ONLY 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。

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。

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。

CONVERT

作業中の表の索引がタイプ 1 かタイプ 2 かが分からない場合で、タイプ 2 の索引を使用したいときは、CONVERT オプションを使用できます。索引がタイプ 1 であれば、このオプションはそれをタイプ 2 に変換します。索引が既にタイプ 2 であれば、このオプションは何も行いません。

バージョン 8 よりも前の DB2 によって作成されたすべての索引はタイプ 1 の索引です。バージョン 8 によって作成された索引は、すべてタイプ 2 の索引です。ただし、既にタイプ 1 の索引を持つ表に作成した索引は例外です。この場合、新しい索引もタイプ 1 索引になります。

INSPECT コマンドを使用して索引のタイプを判別する手順は時間がかかることがあります。CONVERT を使用すると、元のタイプを判別しなくても、新規の索引が必ずタイプ 2 となるようにすることができます。

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。ALLOW READ ACCESS および ALLOW WRITE ACCESS は表へのアクセスを許可しますが、索引の再編成済みコピーを使用できる間は、表へのアクセスが許可されません。

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 表に対して使用することはできません。

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 表を再編成しているときには、索引を指定できません。

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。パーティション表を再編成する場合、これがデフォルトです。パーティション表の再編成はオフラインで実行されます。

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。これは、非パーティション表の場合のデフォルトです。

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、非パーティションかつ非 MDC の表で、タイプ 2 の索引があり、拡張索引はなく、表内の XML 列に対して索引が定義されていない場合のみです。インプレース表再編成は、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インプレース表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。

8KB、16KB、または 32KB 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。

同じページ・サイズの 2 つの **TEMPORARY** 表スペースがあり、そのうちの 1 つを **USE** 節に指定している場合、再編成する表に索引があるなら、それらの表スペースはラウンドロビン方式で使用されます。 **tempSPACE1** および **tempSPACE2** の 2 つの表スペースがあり、それらのページ・サイズが同じである場合に、**USE** オプションを指定した **REORG** コマンドで **tempSPACE1** を指定したとします。初めて **REORG** を実行する際には、**tempSPACE1** が使用されます。2 回目には **tempSPACE2** が使用されます。3 回目には **tempSPACE1** が使用される、という具合になります。このような状況を避けるには、それらの一時表スペースのいずれか 1 つをドロップしてください。

パーティション表の場合、表に含まれるすべてのデータ・パーティションの再編成において、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。そのためのスペース所要量は、表全体ではなく、表に含まれる最大のデータ・パーティションに等しくなります。

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。

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、この表に十分の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。 **long**、**LOB**、索引、または **XML** オブジェクトを圧縮することはできません。

表 31. *REORG KEEPDICTIONARY*

圧縮	ディクショナリーの存在	結果、出力
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

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。 **long**、**LOB**、索引、または **XML** オブジェクトを圧縮することはできません。

表 32. *REORG RESETDICTIONARY*

圧縮	ディクショナリーの存在	結果、出力
Y	Y	ディクショナリー新規作成*、行圧縮
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

表 32. REORG RESETDICTIONARY (続き)

圧縮	ディクショナリーの存在	結果、出力
N	N	影響なし、全行圧縮解除

* - 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。

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。ノード節が指定されていない場合、これがデフォルトです。

EXCEPT

ノード・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 ノード・システムのノード 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 コーティリティーでは、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、宣言一時表に対しては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、範囲クラスター化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。
- 表が再編成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。
- パーティション表では、以下のようになります。
 - REORG は表レベルでサポートされています。個々のデータ・パーティションの再編成は、そのデータ・パーティションをデタッチし、その結果の非パーティション表を再編成した後、そのデータ・パーティションを再びアタッチすることによって実現できます。
 - 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。
 - 表の再編成中にエラーが発生した場合、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。索引は次回表にアクセスしたときに再作成されます。

- ALLOW NONE アクセス・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、表の一部の非パーティション索引が無効のままになる場合があります。表の RID 索引については、エラー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。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。無効のマークが付いたすべての索引は、次回表にアクセスしたときに再作成されます。
- 表の再編成操作が失敗した場合、再編成状態のデータ・パーティションとそうでないデータ・パーティションが混在している可能性があります。REORG TABLE コマンドを再発行すると、データ・パーティションの再編成状態には関係なくすべてのデータ・パーティションが再編成されます。

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えることがあります。REORG INDEX または REORG INDEXES は、表の 1 つまたはすべての索引の再編成を実行するのに使用できます。索引を再編成すると、フラグメントが除去され、リーフ・ページの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、すべてのロックが解放されていることを確かめてから、索引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。ただし、インプレース表再編成 (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。

表の値圧縮を活動化または非活動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ーします。

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には、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

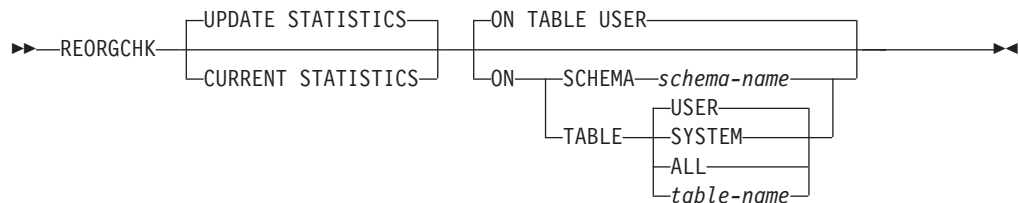
以下のいずれか。

- *sysadm* または *dbadm* の権限
- 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。使用する名前は、*schema.table-name* という形式の完全修飾名または別名で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基準になります。

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 表であることを示します。索引名への "*" 接尾部は、これが MDC 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。

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、索引に関与するすべての列 (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

内部使用に備えて予約されている索引リーフ・ページ上のスペース。

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。

表 33. 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・タイプの関数です。

変数	REGULAR 表スペース内のデータ			LARGE 表スペース内のデータ**		
	REGULAR 表		パーティション表	REGULAR 表		パーティション表
	XML パスまたは領域の索引	他のすべての索引	すべての索引	XML パスまたは領域の索引	他のすべての索引	すべての索引
LEAF_RECSIZE_OVERHEAD	9	7	9	9	9	11
NLEAF_RECSIZE_OVERHEAD	12	10	12	12	12	14

表 33. 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・タイプの関数です。(続き)

変数	REGULAR 表スペース内のデータ			LARGE 表スペース内のデータ**		
DUPKEYSIZE	7	5	7	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 を戻すか、REORGCHK コマンドに続けて RUNSTATS を発行してください。

注: この公式は、XML 列上の索引に対してはサポートされません。

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 を戻すか、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。

注: この公式は、XML 列上の索引に対してはサポートされません。

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 ONLY 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 ONLY 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。

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たデータ・パーティションがあるかどうかを確認するには、SYSDATAPARTITIONS 表内の状況フィールドを調べて、値 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。

使用上の注意

このコマンドは、宣言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、表および索引に関する新しいフィーチャーが導入される可能性があります。新規のフィーチャーは REORGCHK ロジックに、すなわち REORGCHK が REORG 推奨を計算する方法に影響を与える可能性があります。REORGCHK がサーバーと同じレベルではないクライアントから発行される場合、サーバーと同じレベルのクライアントから報告される結果とは異なる結果にな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 はサーバーと同じレベルのクライアントで実行される必要があります。そのようにすることによって、最も正確なレポートが確実に作成されるようになります。一般的に、サーバーの管理作業では、同じレベルのクライアントおよびサーバーを使用します。

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atalog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 エクステント・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 <= '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

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。

$$100 * \text{TSIZE} / ((\text{FPAGES} - \text{NPARTITIONS}) * (\text{TABLEPAGE SIZE} - 68)) > 70$$

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。

$$100 * TSIZE / ((ACTBLK - FULLKEYCARD) * EXTENTSIZE * (TABLEPAGESIZE - 68)) > 70$$

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:

$$100 * NPAGES / FPAGES > 80$$

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)。上で述べたとおり、(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 表の場合、式は次のようになります。

$$100 * \text{activeblocks} / ((\text{fpages} / \text{ExtentSize}) - 1)$$

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。

$$\text{AVGPARTITION_CLUSTERRATIO or normalized AVGPARTITION_CLUSTERFACTOR} > 80$$

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。

• 公式 F5:

– 単一データベース・パーティションでは、以下のようになります。

$$\frac{100 * (KEYS * (LEAF_RECSIZE + LEAF_RECSIZE_OVERHEAD) + (INDCARD - KEYS) * DUPKEYSIZE)}{(NLEAF - NUM_EMPTY_LEAFS - 1) * (INDEXPAGESIZE - LEAF_PAGE_OVERHEAD) } > \text{MIN}(50, (100 - PCTFREE))$$

リーフ・レベルの索引で使用されるスペースは、50 パーセントと 100-PCTFREE パーセントのうち最小のものより大きくなければなりません (これは NLEAF>1 のときのみ検査されます)。

– 複数パーティション・データベース環境では、以下のようになります。

$$\frac{100 * (KEYS * (LEAF_RECSIZE + LEAF_RECSIZE_OVERHEAD) + (INDCARD - KEYS) * DUPKEYSIZE)}{(NLEAF - NUM_EMPTY_LEAFS - NPARTITIONS) * (INDEXPAGESIZE - LEAF_PAGE_OVERHEAD) } > \text{MIN}(50, (100 - PCTFREE))$$

• 公式 F6:

$$\frac{(100 - PCTFREE) * (\text{FLOOR}((100 - \text{LEVEL2PCTFREE}) / 100 * (\text{INDEXPAGESIZE} - \text{NLEAF_PAGE_OVERHEAD}) / (\text{NLEAF_RECSIZE} + \text{NLEAF_RECSIZE_OVERHEAD})) * (\text{FLOOR}((100 - \text{MIN}(10, \text{LEVEL2PCTFREE})) / 100 * (\text{INDEXPAGESIZE} - \text{NLEAF_PAGE_OVERHEAD}) / (\text{NLEAF_RECSIZE} + \text{NLEAF_RECSIZE_OVERHEAD})) ** (\text{NLEVELS} - 3)) * (\text{INDEXPAGESIZE} - \text{LEAF_PAGE_OVERHEAD}) / (\text{KEYS} * (\text{LEAF_RECSIZE} + \text{LEAF_RECSIZE_OVERHEAD}) + (\text{INDCARD} - \text{KEYS}) * \text{DUPKEYSIZE})) < 100$$

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。

公式 F6 は以下のように簡潔な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。

$$\frac{\text{(The max number of leaf pages that a NLEVELS-1 index can have) * (Amount of space available to store index entries per leaf page)}}{\text{Amount of space needed for all the entries in the index}}$$

詳細は次のとおりです。

$$\frac{\text{The max number of leaf pages that a NLEVELS-1 index can have} = \text{(No. of entries a level 2 index page can have) * (No. of entries per page on levels greater than 2) ** (No. of levels in the intended index - 2) =$$

$$(100 - \text{LEVEL2PCTFREE})$$

$$\left\{ \left\lfloor \frac{\left[\frac{(\text{PageSize} - \text{Overhead})}{(\text{Avg. size of each nonleaf index entry})} \right]}{100} \right\rfloor * \left\lfloor \frac{(100 - \text{MIN}(10, \text{LEVEL2PCTFREE}))}{100} \right\rfloor * \left[\frac{(\text{PageSize} - \text{Overhead})}{(\text{Avg. size of each nonleaf index entry})} \right]^{**} \right\} (\text{NLEVELS}-3)$$

(100 - LEVEL2PCTFREE) is the percentage of used space on level 2 of the index.

Level 2 is the level immediately above the leaf level.

(100 - MIN(10, LEVEL2PCTFREE)) is the percentage of used space on all levels above the second level.

NLEVELS is the number of index levels in the existing index.

The amount of space available to store index entries per leaf page =
 $((100 - \text{PCTFREE}) / 100 * (\text{INDEXPAGESIZE} - \text{LEAF_PAGE_OVERHEAD})) =$
 (Used space per page * (PageSize - Overhead))

The amount of space needed for all index entries:
 $\text{KEYS} * (\text{LEAF_RECSIZE} + \text{LEAF_RECSIZE_OVERHEAD}) +$
 $(\text{INDCARD} - \text{KEYS}) * \text{DUPKEYSIZE}$

$(\text{KEYS} * (\text{LEAF_RECSIZE} + \text{LEAF_RECSIZE_OVERHEAD}))$ 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、 $((\text{INDCARD} - \text{KEYS}) * \text{DUPKEYSIZE})$ 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:

$$100 * (\text{NUMRIDS_DELETED} / (\text{NUMRIDS_DELETED} + \text{INDCARD})) < 20$$

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:

$$100 * (\text{NUM_EMPTY_LEAFS} / \text{NLEAF}) < 20$$

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。

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。

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 構成ファイルのリセットします。

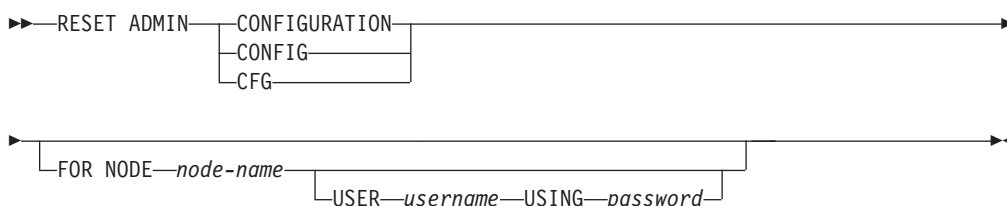
許可

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。

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。

許可

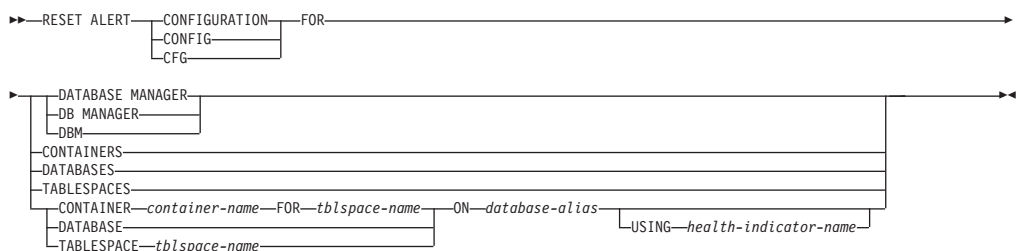
以下のいずれか。

- *sysadm*
- *sysmaint*
- *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。

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データベース・パーティションを更新します。ただし、**DBPARTITIONNUM** が 1 つのデータベース・パーティション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

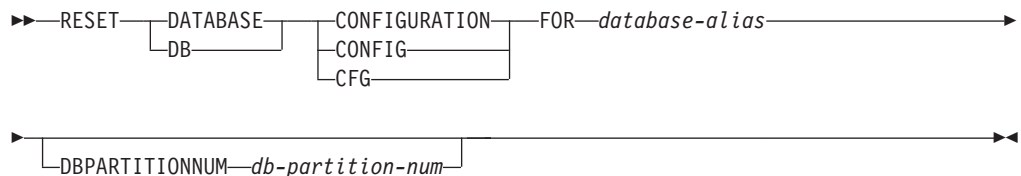
以下のいずれか。

- 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。

DBPARTITIONNUM *db-partition-num*

データベース構成のリセットが特定のデータベース・パーティションに適用される場合、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データベース・パーティションに対して有効です。

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
```

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。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。

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。

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・マネージャー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。

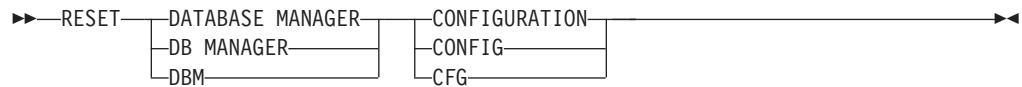
許可

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・マネージャー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・マネージャー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

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 を再始動するときにエラー・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、**svccname** 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き、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管します。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 **svccname** 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。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。

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。

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、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。

許可

以下のいずれか。

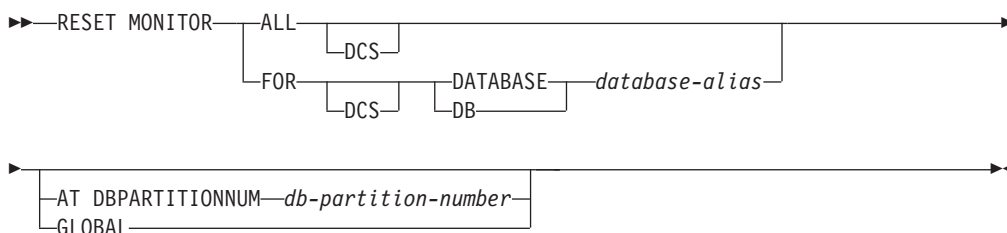
- *sysadm*
- *sysctrl*
- *sysmaint*
- *sysmon*

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必要があります。

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 DBPARTITIONNUM *db-partition-number*

モニター・スイッチの状況を表示するデータベース・パーティションを指定します。

GLOBAL

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・パーティションの集約結果を戻します。

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、一部のデータベース・パーティション・レベルのカウンターもリセットされます。

互換性

バージョン 8 より前のバージョンとの互換性：

- キーワード NODE は、DBPARTITIONNUM の代わりとして使用できます。

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、それが実行されたノードに対してだけ影響を与えます。

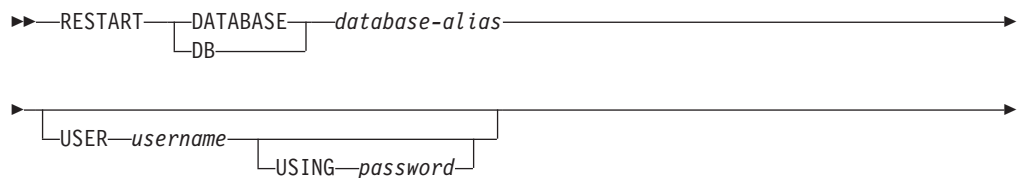
許可

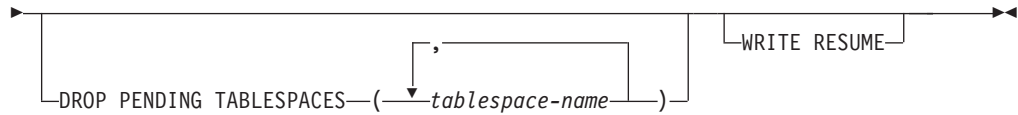
なし

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・コンテナー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ー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。

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ー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

入出力書き込みの中断中に障害の生じたデータベースに対して、データベースの再始動を強制することを可能にします。このオプションを使用すると、クラッシュ・リカバリーを実行する前に、データベース内のすべての表スペースから SUSPEND_WRITE 状態が除去され、入出力書き込みが再開されます。

WRITE RESUME オプションは、入出力書き込みを中断するのに使用する接続がハングし、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使用できます。このような状態で使用すると、**RESTART DATABASE** 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を再開します。**RESTART DATABASE** を **WRITE RESUME** オプションを指定して使用すると、データベースがクラッシュしてから使用する場合にはクラッシュ・リカバリーのみを実行します。**WRITE RESUME** パラメーターは、ミラーリングされたデータベースではなく、1 次データベースだけに適用できます。

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、このコマンドを実行してください。このアクションは、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) 場合にのみ起こります。

パーティション・データベース・システム上では、未確定トランザクションを解決するために、すべてのノード上で以下の例で示すように `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"
```

データベースが MPP システム中の単一ノードでのみ再始動する場合、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場合があります。これが起こるのは、照会が依存しているノード上のデータベース・パーティションも再始動しなければならないためです。すべてのノードでデータベースを再始動すれば、問題を解決できます。

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。また、このユーティリティーを使って、別のイメージをデータベースに上書きしたり、バックアップ・コピーを新しいデータベースにリストアすることもできます。

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リストア操作の詳細は、「[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。

さらに、リストア・ユーティリティーは、DB2 Universal Database バージョン 8 で作成されたバックアップ・イメージをリストアするのにも使用できます。マイグレーションが必要な場合、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。

このユーティリティーは、表スペース・レベルのバックアップをリストアすることもできます。

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 ユーティリティを使用)。SQL プロシージャは、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。

許可

既存のデータベースにリストアするには、次のいずれかが必要です。

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

新規のデータベースにリストアするには、次のいずれかが必要です。

- *sysadm*
- *sysctrl*

必要な接続

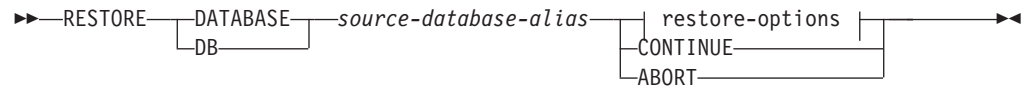
必要な接続は、リストア・アクションの種類によって異なります。

- 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。

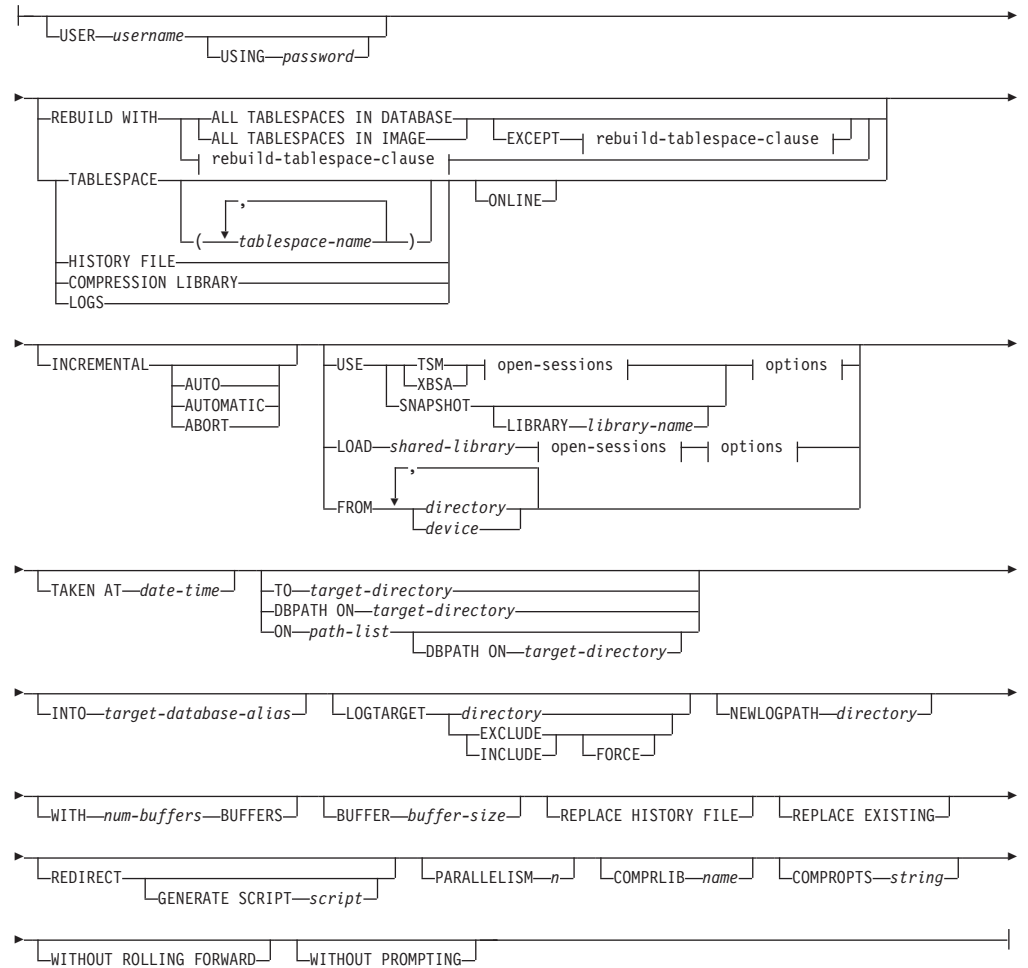
現行のインスタンスとは異なるインスタンスで新規のデータベースへのリストアを行うには、まず、新規のデータベースを存在させるインスタンスにアタッチすることが必要です。新規インスタンスは、ローカルでもリモートでもかまいません。現在のインスタンスは、DB2INSTANCE 環境変数の値によって定義されません。

- 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。

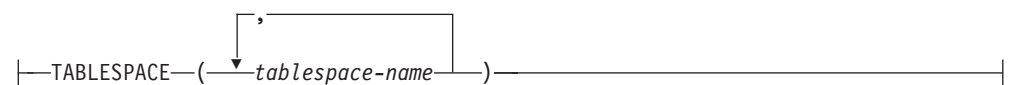
コマンド構文



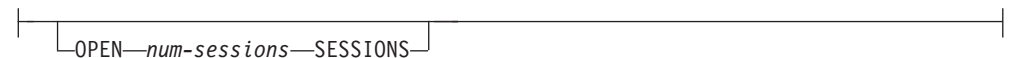
restore-options:



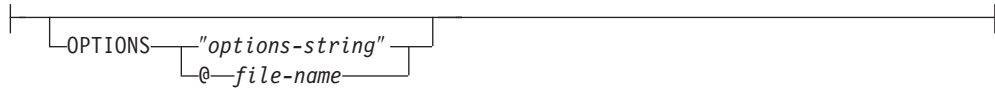
rebuild-tablespace-clause:



open-sessions:



オプション:



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*

データベースがリストアされ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。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。

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。

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のに指定されます。

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 データベースが Tivoli Storage Manager 管理の出力から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 パラメーターは、以下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・イメージからのデータ・リストア時のデフォルトの動作は、すべてのコンテナ、ローカル・ボリューム・ディレクトリー、データベース・パス (DBPATH)、最新のスナップショット・バックアップの 1 次ログとミラー・ログのパス (タイム・スタンプが指定されていない場合) を含む、データベースを構成するすべてのパスの FULL DATABASE OFFLINE リストアです (明示的に EXCLUDE LOGS が指定されているのでない限り、すべてのスナップショット・バックアップでのデフォルトは INCLUDE LOGS です)。タイム・スタンプが提供されている場合、そのスナップショット・バックアップ・イメージがリストアされます。

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 System Storage[™] DS6000[™]
- IBM System Storage DS8000[™]
- IBM System Storage N Series
- NetApp V-series
- NetApp FAS

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。このストリングは、二重引用符なしで、入力されたとおりに DB2 ACS API ドライバーに渡されます。 **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・ユーティリティの **OPTIONS** パラメーターを使用する必要があります。

@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。

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、自動ストレージ・データベースに関連したストレージ・パスを再定義します。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。

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・ノードのすべてが RESTORE_PENDING 状態になります。その場合、非カタログ・ノードは、リストア・コマンドに ON 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。

データベース・パーティション式を指示するには、引数 " \$N" ([blank]\$N) を使用します。データベース・パーティション式は、ストレージ・パス内のどこでも使用することができ、複数のデータベース・パーティション式を指定してもかまいません。データベース・パーティション式は、スペース文字で終了します。スペースの後に続くものはすべて、データベース・パーティション式の評価後にストレージ・パスに追加されます。データベース・パーティション式の後、ストレージ・パス内にスペース文字がない場合、そのstringの残りは式の一部であるとみなされます。引数は、以下の形式のいずれかでのみ使用できます。

表 34. : 演算子は、左から右へ評価されます。% は、モジュラス演算子を表します。例中のデータベース・パーティション番号は 10 と想定されています。

構文	例	値
[blank]\$N	" \$N"	10
[blank]\$N+[number]	" \$N+100"	110
[blank]\$N%[number]	" \$N%5"	0
[blank]\$N+[number]%[number]	" \$N+1%5"	1
[blank]\$N%[number]+[number]	" \$N%4+2"	4

^a % はモジュラスです。

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。

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。

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。

注: LOGTARGET がスナップショット以外のリストア用に指定されていない場合には、デフォルトの LOGTARGET ディレクトリーは 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。ただし、これが影響するのは、これを指定したリストア操作に限定されます。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。

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に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。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。自動ストレージが有効になっている表スペースに対して、リダイレクト・リストア操作を実行することはできません。

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。使用法に関する情報は、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・マネージャの数を指定します。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の名前 (例えば、Windows では **db2compr.dll**、Linux/UNIX システムでは **libdb2compr.so** です)。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、このパラメータの値は無視されます。指定されたライブラリ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

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。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.NODE0000.CATN0000.20020331234149.001
wsdb.0.db2inst1.NODE0001.CATN0000.20020331234427.001
wsdb.0.db2inst1.NODE0002.CATN0000.20020331234828.001
wsdb.0.db2inst1.NODE0003.CATN0000.20020331235235.001
```

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、データベース・パーティションの 1 つから以下のコマンドを出します。

```
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
INTO wsdb REPLACE EXISTING'
db2_all '<<+3< db2 RESTORE DATABASE wsdb FROM /dev3/backup
TAKEN AT 20020331235235
INTO wsdb REPLACE EXISTING'
```

`db2_all` 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_all` 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_all` 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. 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。

7. 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

8. 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 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/NODE0000/SQL00001/SQLLOGDIR/'
-- WITH <num-buff> BUFFERS
-- BUFFER <buffer-size>
-- REPLACE HISTORY FILE
-- REPLACE EXISTING
REDIRECT
-- PARALLELISM <n>
-- WITHOUT ROLLING FORWARD
-- WITHOUT PROMPTING
;
-- *****
-- ** tablespace definition
-- *****
-- ** Tablespace name = SYSCATSPACE
-- ** Tablespace ID = 0
-- ** Tablespace Type = System managed space
-- ** Tablespace Content Type = Any data
-- ** Tablespace Page size (bytes) = 4096
-- ** Tablespace Extent size (pages) = 32
-- ** Using automatic storage = No
-- ** Total number of pages = 5572
-- *****
SET TABLESPACE CONTAINERS FOR 0
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
USING (
PATH 'SQLT0000.0'
```

```

);
-- *****
-- ** Tablespace name                = TEMPSPACE1
-- ** Tablespace ID                  = 1
-- ** Tablespace Type                = System managed space
-- ** Tablespace Content Type        = System Temporary data
-- ** Tablespace Page size (bytes)   = 4096
-- ** Tablespace Extent size (pages) = 32
-- ** Using automatic storage        = No
-- ** Total number of pages          = 0
-- *****
SET TABLESPACE CONTAINERS FOR 1
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
USING (
  PATH 'SQLT0001.0'
);
-- *****
-- ** Tablespace name                = USERSPACE1
-- ** Tablespace ID                  = 2
-- ** Tablespace Type                = System managed space
-- ** Tablespace Content Type        = Any data
-- ** Tablespace Page size (bytes)   = 4096
-- ** Tablespace Extent size (pages) = 32
-- ** Using automatic storage        = No
-- ** Total number of pages          = 1
-- *****
SET TABLESPACE CONTAINERS FOR 2
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
USING (
  PATH 'SQLT0002.0'
);
-- *****
-- ** Tablespace name                = DMS
-- ** Tablespace ID                  = 3
-- ** Tablespace Type                = Database managed space
-- ** Tablespace Content Type        = Any data
-- ** Tablespace Page size (bytes)   = 4096
-- ** Tablespace Extent size (pages) = 32
-- ** Using automatic storage        = No
-- ** Auto-resize enabled            = No
-- ** Total number of pages          = 2000
-- ** Number of usable pages         = 1960
-- ** High water mark (pages)        = 96
-- *****
SET TABLESPACE CONTAINERS FOR 3
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
USING (
  FILE /tmp/dms1                1000
, FILE /tmp/dms2                1000
);
-- *****
-- ** Tablespace name                = RAW
-- ** Tablespace ID                  = 4
-- ** Tablespace Type                = Database managed space
-- ** Tablespace Content Type        = Any data
-- ** Tablespace Page size (bytes)   = 4096
-- ** Tablespace Extent size (pages) = 32
-- ** Using automatic storage        = No
-- ** Auto-resize enabled            = No
-- ** Total number of pages          = 2000
-- ** Number of usable pages         = 1960
-- ** High water mark (pages)        = 96
-- *****
SET TABLESPACE CONTAINERS FOR 4
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
USING (
  DEVICE '/dev/hdb1'            1000

```

```

, DEVICE '/dev/hdb2'                                1000
);
-- *****
-- ** start redirect restore
-- *****
RESTORE DATABASE SAMPLE CONTINUE;
-- *****
-- ** end of file
-- *****

```

9. 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。

```

-- *****
-- ** automatically created redirect restore script
-- *****
UPDATE COMMAND OPTIONS USING S ON Z ON TEST_NODE0000.out V ON;
SET CLIENT ATTACH_DBPARTITIONNUM 0;
SET CLIENT CONNECT_DBPARTITIONNUM 0;
-- *****
-- ** initialize redirected restore
-- *****
RESTORE DATABASE TEST
-- USER '<username>'
-- USING '<password>'
FROM '/home/jseifert/backups'
TAKEN AT 20050304090733
ON '/home/jseifert'
-- DBPATH ON <target-directory>
INTO TEST
-- NEWLOGPATH '/home/jseifert/jseifert/NODE0000/SQL00002/SQLLOGDIR/'
-- WITH <num-buff> BUFFERS
-- BUFFER <buffer-size>
-- REPLACE HISTORY FILE
-- REPLACE EXISTING
REDIRECT
-- PARALLELISM <n>
-- WITHOUT ROLLING FORWARD
-- WITHOUT PROMPTING
;
-- *****
-- ** tablespace definition
-- *****
-- *****
-- ** Tablespace name                = SYCATSPACE
-- ** Tablespace ID                  = 0
-- ** Tablespace Type                 = Database managed space
-- ** Tablespace Content Type        = Any data
-- ** Tablespace Page size (bytes)   = 4096
-- ** Tablespace Extent size (pages) = 4
-- ** Using automatic storage        = Yes
-- ** Auto-resize enabled             = Yes
-- ** Total number of pages          = 6144
-- ** Number of usable pages         = 6140
-- ** High water mark (pages)        = 5968
-- *****
-- *****
-- ** Tablespace name                = TEMPSPACE1
-- ** Tablespace ID                  = 1
-- ** Tablespace Type                 = System managed space
-- ** Tablespace Content Type        = System Temporary data
-- ** Tablespace Page size (bytes)   = 4096

```

```

-- ** Tablespace Extent size (pages) = 32
-- ** Using automatic storage = Yes
-- ** Total number of pages = 0
-- *****
-- *****
-- ** Tablespace name = USERSPACE1
-- ** Tablespace ID = 2
-- ** Tablespace Type = Database managed space
-- ** Tablespace Content Type = Any data
-- ** Tablespace Page size (bytes) = 4096
-- ** Tablespace Extent size (pages) = 32
-- ** Using automatic storage = Yes
-- ** Auto-resize enabled = Yes
-- ** Total number of pages = 256
-- ** Number of usable pages = 224
-- ** High water mark (pages) = 96
-- *****
-- *****
-- ** Tablespace name = DMS
-- ** Tablespace ID = 3
-- ** Tablespace Type = Database managed space
-- ** Tablespace Content Type = Any data
-- ** Tablespace Page size (bytes) = 4096
-- ** Tablespace Extent size (pages) = 32
-- ** Using automatic storage = No
-- ** Auto-resize enabled = No
-- ** Total number of pages = 2000
-- ** Number of usable pages = 1960
-- ** High water mark (pages) = 96
-- *****
SET TABLESPACE CONTAINERS FOR 3
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
USING (
    FILE '/tmp/dms1' 1000
    , FILE '/tmp/dms2' 1000
);
-- *****
-- ** Tablespace name = RAW
-- ** Tablespace ID = 4
-- ** Tablespace Type = Database managed space
-- ** Tablespace Content Type = Any data
-- ** Tablespace Page size (bytes) = 4096
-- ** Tablespace Extent size (pages) = 32
-- ** Using automatic storage = No
-- ** Auto-resize enabled = No
-- ** Total number of pages = 2000
-- ** Number of usable pages = 1960
-- ** High water mark (pages) = 96
-- *****
SET TABLESPACE CONTAINERS FOR 4
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
USING (
    DEVICE '/dev/hdb1' 1000
    , DEVICE '/dev/hdb2' 1000
);
-- *****
-- ** start redirect restore
-- *****
RESTORE DATABASE TEST CONTINUE;
-- *****
-- ** end of file
-- *****

```

10. 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
```

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。
- 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 つのシステムの間のクロスプラットフォーム・リストアが DB2 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。

- SMS 表スペースのバックアップは SMS 表スペースにのみリストアすることができます。SMS 表スペースのバックアップを DMS 表スペースにリストアすることはできません。その逆も同様です。
- 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戻されます。
- 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戻されます。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戻されます。データベース・リストア・ユーティリティーは、LOGTARGET ディレクトリー内に既存のログ・ファイルを上書きしません。
- 保管されているログ・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ことを指定するには、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、最終リストア・コマンドを発行する場合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のマイグレーションは、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 MIGRATE DATABASE コマンドを続けて、データベースを正常にマイグレーションすることができます。

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 マネージャーにより検出された場合、エラーが戻されます。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 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 マネージャーによって検出されたなら、DB2 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。

LOGTARGET EXCLUDE オプションが指定され、ログ・ディレクトリー・パスがデータベース・ディレクトリーの下位 (つまり /NODExxxx/SQLxxxxx/SQLLOGDIR/) にあるような特殊な事例もあります。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 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。

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

以下のいずれか。

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

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

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。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呼び出されます。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。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。

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。これには、表のデータ・パーティションが含まれた表スペースが該当します。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。これは、新規 DB2 リリース・バージョンにマイグレーションする際の重要な考慮事項です。

許可

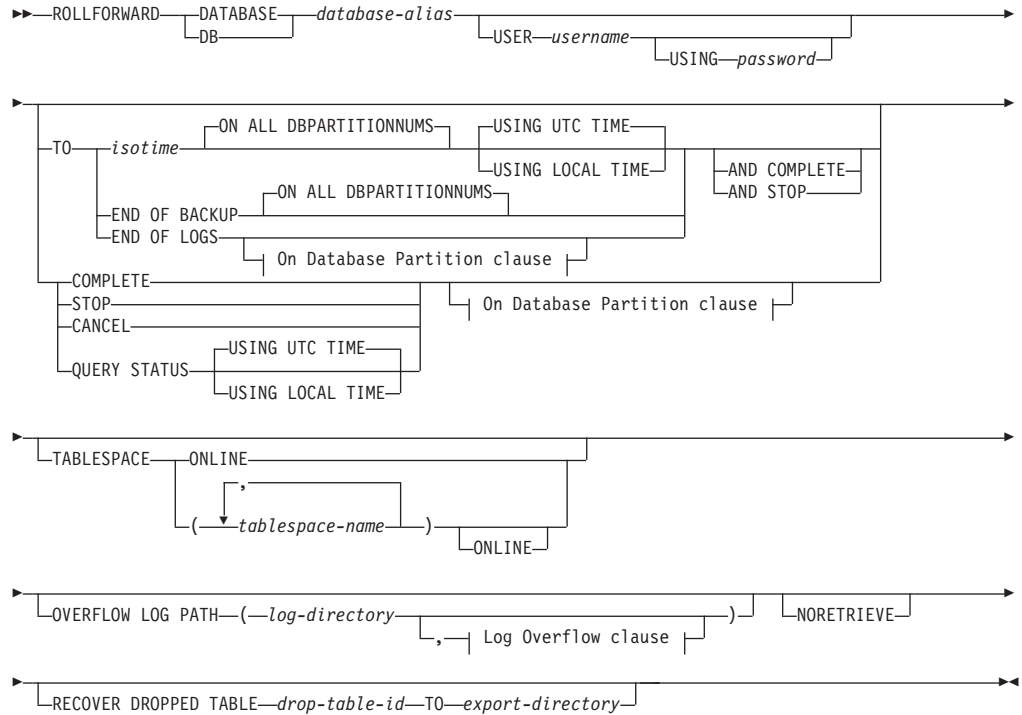
以下のいずれか。

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

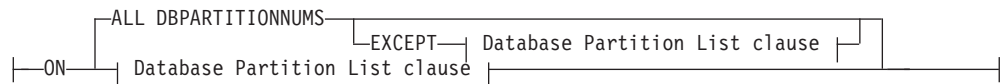
必要な接続

なし。このコマンドは、データベース接続を確立します。

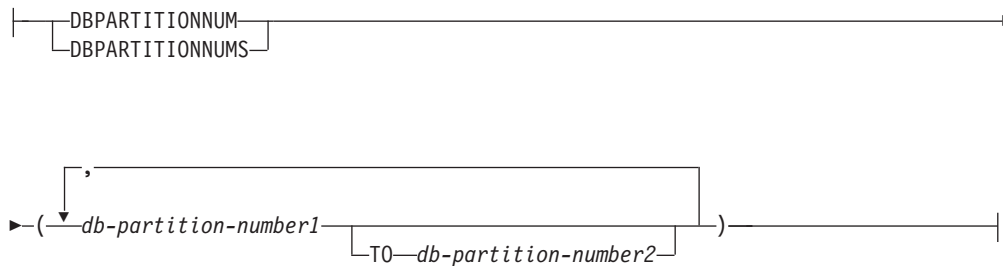
コマンド構文



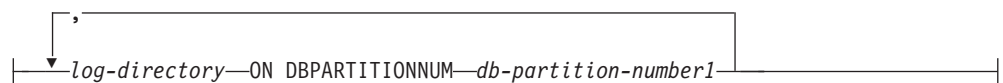
On Database Partition clause:



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)。

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。これは、クライアントのローカル時刻となるコントロール・センターのローカル時刻オプションとは異なります。

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。

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が、最小リカバリ時間 までロールフォワードされることを指定します。例については、以下の『例』のセクション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。

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に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・マネージャーがロールフォワードしたログ・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・データベース環境では、この状況情報は各データベース・パーティションごとに戻されます。戻される情報には、次のフィールドが含まれます。

データベース・パーティション番号

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。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、表スペースのロールフォワード・リカバリー操作の場合は更新されません。

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。パーティション表の一部が含まれた単一の表スペースのログの末尾までのロールフォワードも可能です。

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。

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できるようにして、スタンバイ・マシン上などのログ・ファイルをロールフォワードするかを制御できるようにします。これには以下の利点があります。

- ロールフォワードするログ・ファイルを制御することによって、スタンバイ・マシンが実動マシンより必ず X 時間遅れるようにして、両方のシステムに影響が及ぶことを防ぎます。
- スタンバイ・システムがアーカイブにアクセスできない場合 (例えば、TSM がアーカイブの場合にはオリジナル・マシンだけがこのファイルを検索できます)
- 実動システムがファイルをアーカイブしている間に、スタンバイ・システムが同一のファイルを検索して、不完全なログ・ファイルを取得することがあります。この問題は、**NORETRIEVE** を使用して解決できます。

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。

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)
```

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)

ROLLFORWARD コマンドで TO END OF BACKUP 節を使用することにより、パーティション・データベースの全パーティションを最小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
```

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。

データベースのロールフォワードには、テープ装置を使用したロード・リカバリーが必要な場合があります。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。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。

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。

互換性

バージョン 8 より前のバージョンとの互換性 :

- キーワード NODE は、DBPARTITIONNUM の代わりとして使用できます。
- キーワード NODES は、DBPARTITIONNUMS の代わりとして使用できます。
- V9.1 でパーティション表がサポートされるようになったため、V9.1 より前のクライアントにおいて特定時点ロールフォワードはサポートされていません。

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、このユーティリティーは、表が数多く更新される時、または表を再編成した後で、呼び出してください。統計ビューにおいては、基礎表に対する変更が、ビューによって戻される行に対して実質的に影響を与える場合に、このユーティリティーを呼び出す必要があります。そのビューは、それ以前に ALTER VIEW コマンドを使用して、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

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、以下のいずれか 1 つです。

- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*
- 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グローバル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、以下のいずれか 1 つです。

- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*
- 統計ビューに対する CONTROL 特権

さらに、統計ビューの行にアクセスするための適切な特権が必要です。特に、統計ビューの定義の中で参照されている各表、統計ビュー、またはニックネームごとに、ユーザーには以下の特権のいずれか 1 つが必要です。

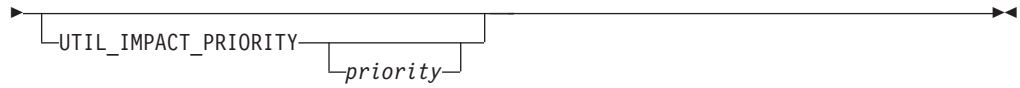
- *sysadm* または *dbadm*
- CONTROL
- SELECT

必要な接続

データベース

コマンド構文

```
►►—RUNSTATS—ON TABLE—object-name—  
|—USE PROFILE—|  
|—UNSET PROFILE—|  
| Statistics Options |
```



Statistics Options:

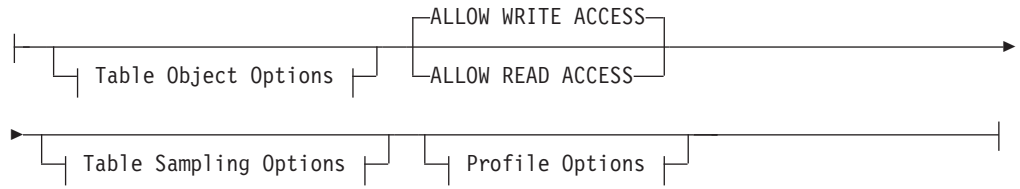


Table Object Options:

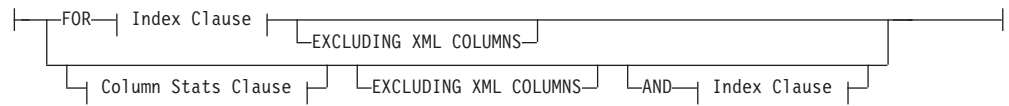
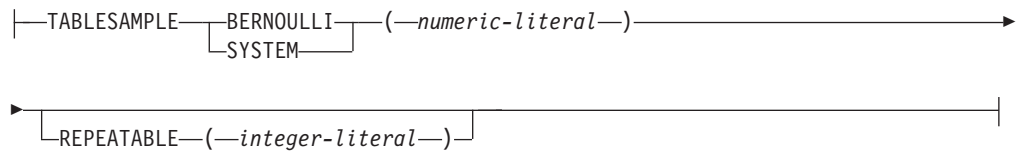


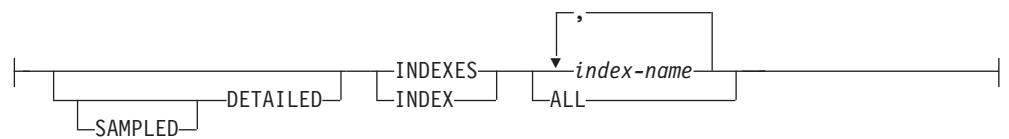
Table Sampling Options:



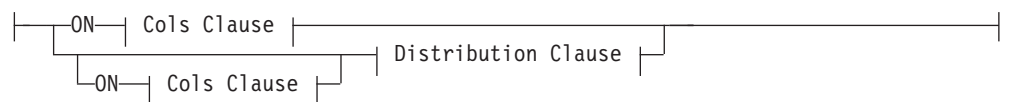
Profile Options:



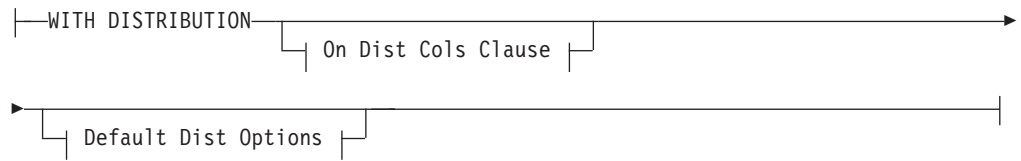
Index Clause:



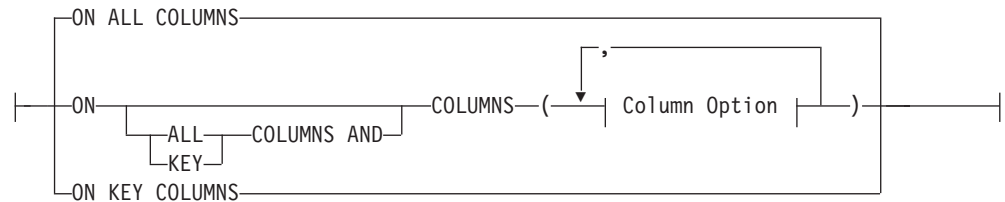
Column Stats Clause:



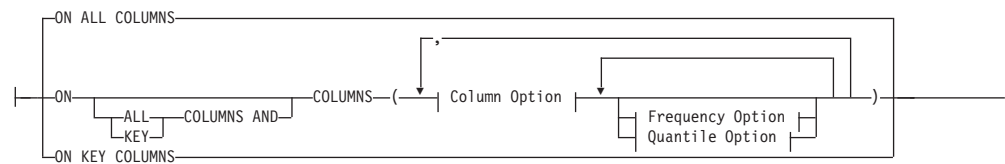
Distribution Clause:



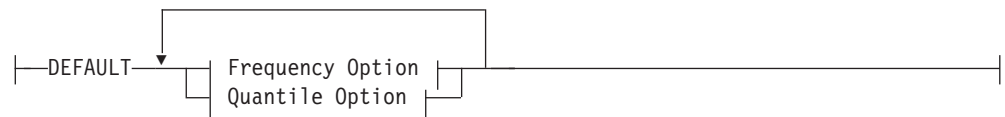
On Cols Clause:



On Dist Cols Clause:



Default Dist Option:



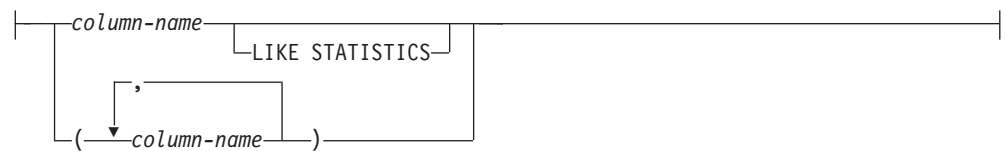
Frequency Option:



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。階層表は指定できません。型付き表の場合は、*object-name* を表階層のルート表の名前にしてください。使用する名前は、*schema.object-name* という形式の完全修飾名または別名で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。

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使用しなければなりません。ビューの場合、このオプションは使用できません。

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

このオプションを指定すると、既存の統計プロファイルが削除されます。例えば、

```
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。

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。

DETAILED

拡張された索引統計を計算します。これは CLUSTERFACTOR および PAGE_FETCH_PAIRS 統計で、比較的大規模な索引の場合に収集されます。ビューの場合、このオプションは使用できません。

SAMPLED

このオプションは、DETAILED オプションと合わせて使用することにより、拡張された索引統計をコンパイルする際に RUNSTATS で CPU のサンプリング方式を使用できるようにします。オプションが指定されていないときは、索引内のすべての項目で、拡張された索引統計を計算するかどうか調べられます。ビューの場合、このオプションは使用できません。

ON ALL COLUMNS

一部の列で統計の収集を行えますが、収集を行えない列もあります。LONG VARCHAR 列や CLOB 列では統計を収集できません。収集が可能なすべての列で統計を収集することが希望の場合は、ON ALL COLUMNS 節を使用できます。列は、基本統計の収集に指定するか (On Cols clause)、または WITH DISTRIBUTION 節と組み合わせて指定 (On Dist Cols Clause) 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 Cols Clause 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、すべての列では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 Cols Clause での指定はすべて重複になるため、必要ありません。

ON COLUMNS

この節では、統計の収集を行う列のリストを任意に指定できます。列のグループを指定した場合は、そのグループの特色となるいくつかの値が収集されます。索引統計情報を収集することなく表に対して RUNSTATS 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、

1. 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。
2. 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 Clause と On Dist Cols Clause 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。

EXCLUDING XML COLUMNS

この節を使用すると、XML タイプのすべての列を、統計情報収集から除外することができます。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。

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-clause または on-dist-cols-clause 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-clause 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められません。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。一方は分散なし、一方は分散ありで、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)。

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、収集する変位値の最大値になります。

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頻度と分位の統計の最大数が決まります。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。

LIKE STATISTICS

このオプションを指定すると、付加的な列統計が収集されます。収集されるのは、SYSSTAT.COLUMNS の SUB_COUNT および SUB_DELIM_LENGTH 統計です。これらの統計は、1 バイト文字セット (SBCS)、FOR BIT DATA、または UTF-8 のコード・ページ属性を持つタイプ CHAR および VARCHAR の列について収集されます。これらの統計は、タイプ "column LIKE '%xyz'" および "column LIKE '%xyz%'" の述部に関する選択度の評価を上げるために、照会オプティマイザーで使用されます。

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。統計情報を最新のものに保つことは照会の最適化の点で重要ですが、統計情報を一貫性のあるものに保つ点でも重要です。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うことが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。

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トの行だけが、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トのデータ・ページだけが、検索されて統計収集に使用されます。統計ビューの場合、SYSTEM (システム) サンプルングは、特定のクラスのビューに制限されます。単一の基本表かニックネームのいずれかにアクセスするビュー、あるいは参照保全リレーションシップを通じて結合された複数の基本表にアクセスするビューがあります。どちらの場合も、ビュー定義の中にローカル述部が含まれてはなりません。SYSTEM (システム) サンプルングのサポートが不可能なビューに対して、そのようなサンプルングが指定された場合、SQL20288N エラーが発生します。

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。

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、サンプリ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 ではエラー (SQL1197N) として扱われます。

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 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。

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、ユーティリティー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ー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ーは、優先順位の高いユーティリティー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。ビューの場合、このオプションは使用できません。

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 db2user.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 db2user.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、表に指定された頻度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 db2user.employee WITH DISTRIBUTION DEFAULT  
NUM_FREQVALUES 50
```

4. 一連の索引で統計を収集します。

```
RUNSTATS ON TABLE db2user.employee for indexes  
db2user.emp11, db2user.emp12
```

5. すべての索引のみの基本統計を収集します。

```
RUNSTATS ON TABLE db2user.employee FOR INDEXES ALL
```

6. 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 db2user.employee AND SAMPLED DETAILED INDEXES ALL
```

7. 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 db2user.employee  
WITH DISTRIBUTION ON COLUMNS (empid, empname, empdept NUM_FREQVALUES  
50 NUM_QUANTILES 100)  
DEFAULT NUM_FREQVALUES 5 NUM_QUANTILES 10  
AND INDEXES db2user.Xempid, db2user.Xempname
```

8. 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 db2user.employee ON KEY COLUMNS AND INDEXES ALL
```

9. 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` が含まれていると考えてください。

```
RUNSTATS ON TABLE db2user.T1  
WITH DISTRIBUTION ON COLUMNS (c1, c2, c3 NUM_FREQVALUES 20  
NUM_QUANTILES 40, c4, c5, c6, c7, c8)  
DEFAULT NUM_FREQVALUES 0, NUM_QUANTILES 0 AND INDEXES ALL
```

```
RUNSTATS ON TABLE db2user.T1
  WITH DISTRIBUTION ON COLUMNS (c3 NUM_FREQVALUES 20 NUM_QUANTILES 40)
  AND INDEXES ALL
```

10. 表 T1 で、個別の列 c1 および c5 と、列の組み合わせ (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、照会オプションマイザーが、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 db2user.T1 ON COLUMNS (c1, (c2, c3),
  (c2, c4), c5)
```

11. 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 db2user.T1 ON COLUMNS (c1 LIKE STATISTICS, c2)
```

12. 構成の設定から NUM_QUANTILES を取り出しながら、表に指定された頻度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に登録します。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 db2user.employee WITH DISTRIBUTION DEFAULT
  NUM_FREQVALUES 50 SET PROFILE
```

13. 構成の設定から NUM_QUANTILES を取り出しながら、表に指定された頻度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に登録します。統計は収集されません。

```
RUNSTATS ON TABLE db2user.employee WITH DISTRIBUTION
  DEFAULT NUM_FREQVALUES 50 SET PROFILE ONLY
```

14. 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 db2user.employee WITH DISTRIBUTION
  DEFAULT NUM_FREQVALUES 30 UPDATE PROFILE
```

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。統計は収集されません。

```
RUNSTATS ON TABLE db2user.employee WITH DISTRIBUTION
  DEFAULT NUM_FREQVALUES 30 UPDATE PROFILE ONLY
```

16. 列 emp1_address と列グループ (emp1_title, emp1_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 db2user.employee
  ON COLUMNS (emp1_address, (emp1_title, emp1_salary))
  UPDATE PROFILE
```

17. 列 emp1_address と列グループ (emp1_title, emp1_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 db2user.employee
  ON COLUMNS (emp1_address, (emp1_title, emp1_salary))
  UPDATE PROFILE ONLY
```

18. 表の統計プロファイルに登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 db2user.employee USE PROFILE
```

19. 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 SYSIBM.SYSTABLES WHERE NAME =  
'EMPLOYEE'
```

20. 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 db2user.employee WITH DISTRIBUTION  
TABLESAMPLE BERNOULLI(30)
```

21. 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 db2user.employee WITH DISTRIBUTION  
TABLESAMPLE BERNOULLI(30) REPEATABLE(4196)
```

上記と同じステートメントを実行すると、 interim 内の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。

22. 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 db2user.employee AND INDEXES ALL TABLESAMPLE SYSTEM(1.5)
```

23. 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 TABLE salesdb.product_sales_view
```

24. 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 TABLE salesdb.product_sales_view  
WITH DISTRIBUTION ON COLUMNS (category NUM_FREQVALUES 100 NUM_QUANTILES 100,  
type, product_key) DEFAULT NUM_FREQVALUES 50 NUM_QUANTILES 50
```

25. 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の統計情報を収集します。

```
RUNSTATS ON TABLE db2user.daily_sales  
WITH DISTRIBUTION TABLESAMPLE BERNOULLI (10)
```

26. 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。そうでない場合、エラー (SQL 20288N) が発生します。

```
RUNSTATS ON TABLE db2user.daily_sales  
WITH DISTRIBUTION TABLESAMPLE SYSTEM (2.5)
```

27. ビューについて、分散統計を指定してすべての列に関する統計情報を収集するための統計プロファイルを登録します。

```
RUNSTATS ON TABLE salesdb.product_sales_view  
WITH DISTRIBUTION DEFAULT NUM_FREQVALUES 50 NUM_QUANTILES 50  
SET PROFILE
```

28. 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 salesdb.product_sales_view
WITH DISTRIBUTION DEFAULT NUM_FREQVALUES 25 NUM_QUANTILES 25
UPDATE PROFILE
```

使用上の注意

1. 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たデータ・パーティションがあるかどうかを確認するには、`SYSDATAPARTITIONS` 表内の状況フィールドを調べて、値 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。

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` ユーティリティを使用して、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` ユーティリティ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。
 - 表に多くの索引があり、それらの索引に含まれる `DETAILED` (拡張) 情報がアクセス・プランを向上させる可能性がある場合は、統計の収集にかかる時間を減らすために、`SAMPLED` オプションを考慮してください。
 - 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"`、`"column BETWEEN constant1 AND constant2"` など) や、タイプ `"column LIKE '%xyz'"` の列では、より大きな `NUM_QUANTILES` 値を指定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います。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. 範囲クラスター表の場合、範囲クラスター表の範囲配列プロパティー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。
7. コマンド構文の On Dist Cols Clause では、列のグループ (GROUPS) に対する Frequency Option および Quantile Option 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。
8. 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
9. TABLESAMPLE による RUNSTATS サンプルングは、索引ページではなく表データ・ページでのみ行われます。索引統計とサンプルングが要求された場合は、統計収集のためにすべての索引ページがスキャンされます。そのようになるのは、TABLESAMPLE が適用される表統計の収集においてのみです。しかし、SAMPLED DETAILED オプションを使用すれば、詳細な索引統計をより効率的に収集できます。これは、TABLESAMPLE の場合とは別のサンプルング方法であり、索引統計の詳細なセットにのみ適用されます。
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 SYSIBM.SYSTABLES 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、収集する頻度値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 SYSTABLES に対して IX 表ロックを取得し、また、RUNSTATS 開始時には、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。それでも、U ロックのかかった行を含めて 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。

SET CLIENT

バックエンド・プロセス用の接続設定を指定します。

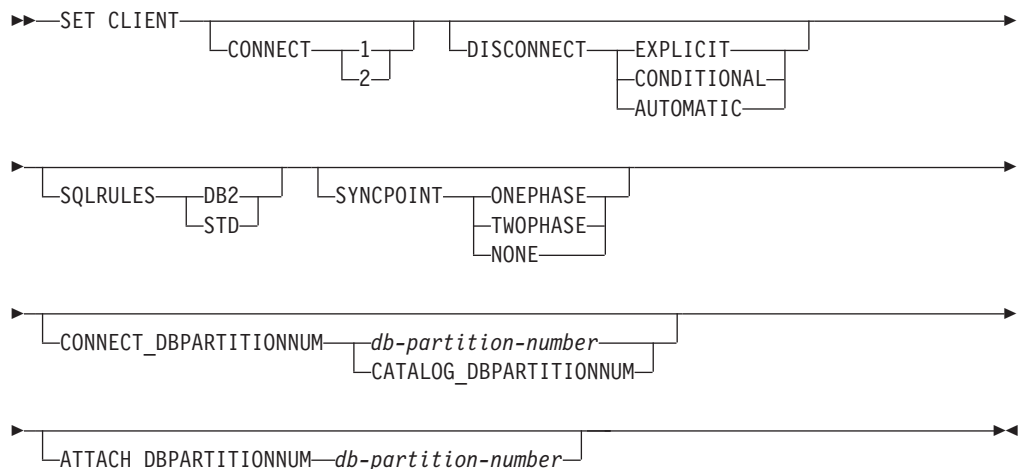
許可

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/ANS 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_DBPARTITIONNUM (パーティション・データベース環境のみ)

db-partition-number

接続先のデータベース・パーティションを指定します。値は 0 から 999 (0 および 999 を含む) です。環境変数 DB2NODE の値をオーバーライドします。

CATALOG_DBPARTITIONNUM

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。

ATTACH_DBPARTITIONNUM *db-partition-number* (パーティション・データベース環境のみ)

アタッチ先のデータベース・パーティションを指定します。値は 0 から 999 (0 および 999 を含む) です。環境変数 DB2NODE の値をオーバーライドします。

例えば、データベース・パーティション 1、2、および 3 が定義された場合、クライアントはこれらのうちの 1 つのデータベース・パーティションにアクセスできればよいことになります。データベースを含むデータベース・パーティション 1 がカタログされ、このパラメーターが 3 に設定されると、次のアタッチは、まずデータベース・パーティション 1 に行われ、次いでデータベース・パーティション 3 に行われます。

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 に戻し、それ以外の設定はそのままにしておくには、次のようになります。

```
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・データベース環境 (DPF) 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_DBPARTITIONNUM 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別します。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

バージョン 8 より前のバージョンとの互換性：

- キーワード CONNECT_NODE は、CONNECT_DBPARTITIONNUM の代わりとして使用できます。

- キーワード CATALOG_NODE は、CATALOG_DBPARTITIONNUM の代わりとして使用できます。
- キーワード ATTACH_NODE は、ATTACH_DBPARTITIONNUM の代わりとして使用できます。

SET RUNTIME DEGREE

指定したアクティブ・アプリケーションの SQL ステートメントについて、パーティション内並列処理のランタイムの最大並列度を設定します。

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。

許可

以下のいずれか。

- *sysadm*
- *sysctrl*

必要な接続

インスタンス。リモート・サーバーで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を変更するには、最初にそのサーバーにアタッチする必要があります。アタッチがない場合、SET RUNTIME DEGREE コマンドの実行は失敗します。

コマンド構文

```

▶▶ SET RUNTIME DEGREE FOR ALL TO degree
    (application-handle)
  
```

コマンド・パラメーター

FOR

ALL 指定した並列処理レベルはすべてのアプリケーションに適用されます。

application-handle

新しい並列処理レベルが適用されるエージェントを指定します。
LIST APPLICATIONS コマンドを使用して値をリストします。

TO *degree*

パーティション内並列処理のランタイムの最大並列度。

例

次の例では、*application-handle* の値が 41408 と 55458 の 2 つのユーザーについて、並列処理のランタイムの最大並列度を 4 に設定します。

使用上の注意

このコマンドは、アクティブなアプリケーションの最大並列処理レベルを修正するメカニズムを提供します。また、SQL ステートメント・コンパイル時間で決定された値を上書きするときにも使用します。

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・マネージャー 構成パラメーターは、データベース・マネージャー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** 構成パラメーター
- 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

SET TABLESPACE CONTAINERS

リダイレクトしたリストア とは、リストアされたデータベースの表スペース・コンテナのセットが、バックアップが行われた時の元のデータベースの表スペース・コンテナと異なるリストアのことです。このコマンドでは、リストアされるデータベースのための表スペース・コンテナの追加、変更、または除去が可能となります。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。

許可

以下のいずれか。

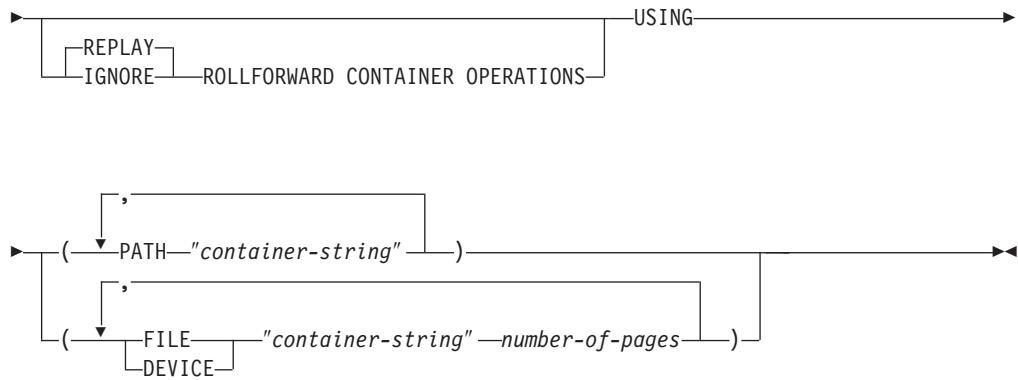
- *sysadm*
- *sysctrl*

必要な接続

データベース

コマンド構文

▶▶—SET TABLESPACE CONTAINERS FOR—*tablespace-id*————▶▶



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) およびコンテナ・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。

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

一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。

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

以下のいずれか。

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

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。

- スロットルが設定されたユーティリティーのスロットルを解除します (スロットルを無効にする)。
- スロットルが設定されたユーティリティーの優先順位を再度設定します(これは複数の同時スロットル・ユーティリティーを実行している場合に便利です)。

有効範囲

許可

以下のいずれか。

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

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く必要があります。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 2 のユーティリティーに対して SET UTIL_IMPACT_PRIORITY コマンドを発行するには、インスタンスのアタッチがパーティション 1 に行われる必要があります。このようにするためには環境に DB2NODE=1 を設定し、インスタンス・アタッチ・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ーの ID。実行中のユーティリティーの ID は、LIST UTILITIES コマンドで取得できます。

TO *priority*

ユーティリティー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 は最高の優先順位を表し、1 は最低の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

ティーをアンスロットルで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・ユーティリティーをスロットル・モードで継続します。

例

以下の例は、ID 2 のユーティリティー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 のユーティリティーを優先順位 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ーはスロットルに設定されます。ユーティリティー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、ユーティリティーの優先順位が再設定されます。

```
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・ユーティリティー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 から 99 はスロットルされるパーセンテージで、100 はスロットルなしです。

SET UTIL_IMPACT_PRIORITY コマンドは、**util_impact_lim** 構成パラメーターの定義に従ってスロットル・ユーティリティー群から使用可能なリソースに対し、特定のユーティリティー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・ユーティリティーを取り上げます。**util_impact_lim=10** の場合、スロットル・アルゴリズムの判定による、すべてのユーティリティーが合計ワークロードに及ぼす影響の平均は、10 % 以下になります。例として、2 つのスロットル・ユーティリティーを使用します。

- 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ー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ーが、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。また、バックアップと runstats のどちらの操作についても、影響の優先順位を各ユーティリティーのコマンド行の中で宣言することができます。SET UTIL_IMPACT_PRIORITY コマンドを発行しない場合、ユーティリティーは (**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。

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 コマンド

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、`dbadm` 権限または `sysadm` 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 `dbadm` 権限または `sysadm` 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を中断したり、入出力書き込み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。コマンドを正常に実行するためには、すべての表スペースが正常状態でなければなりません。正常でない状態の表スペースが 1 つでもあると、コマンドは失敗します。

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

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。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*
- *sysmaint*

必要な接続

データベース

コマンド構文

```
▶▶ SET WRITE {SUSPEND | RESUME} FOR {DATABASE | DB} ▶▶
```

コマンド・パラメーター

SUSPEND

入出力書き込みを中断し、すべての表スペースの状態を書き込み中断にします。ログへの書き込みもこのコマンドによって中断されます。すべてのデータベース操作は、データベース書き込みの中断中に、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するはずですが、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・ページのフラッシュ試行中に待機するものもあります。これらの操作は、データベースの書き込みが再開されると、通常に再開します。

RESUME

入出力書き込みを再開し、すべての表スペースから SUSPEND_WRITE 状態が除去され、表スペースの更新が可能になります。

使用上の注意

入出力書き込みは、中断されたのと同じ接続から再開することが推奨されています。この接続を入出力書き込みに使用できるようにするには、データベース書き込みが再開するまでこの接続からどんな操作も実行しません。操作を実行すると、バッファ・プールまたはログ・バッファからログにダーティ・ページをフラッシュしなければならない場合に、一部の操作が入出力書き込みの再開を待機する可能性があります。さらに、その後に試行される接続は、バッファ・プールからディスクにダーティ・ページをフラッシュしなければならない場合にハングする可能性があります。データベース入出力が再開すると、後続の接続は正常に完了します。接続しようとしてもハングし、入出力を中断した接続からの入出力の再開が不可能になった場合、`RESTART DATABASE` コマンドを `WRITE RESUME` オプションを指定して実行する必要があります。このような状態で使用すると、`RESTART DATABASE` コマンドはクラッシュ・リカバリーを実行せずに、入出力書き込みを再開します。`RESTART DATABASE` コマンドを `WRITE RESUME` オプションを指定して使用すると、データベースがクラッシュしてから使用する場合にはクラッシュ・リカバリーのみを実行します。

START DATABASE MANAGER

単一データベース・パーティション、または複数パーティション・データベース環境で定義されているすべてのデータベース・パーティションで、現行のデータベース・マネージャー・インスタンス・バックグラウンド・プロセスを開始します。

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

以下のいずれか。

- `sysadm`
- `sysctrl`
- `sysmaint`

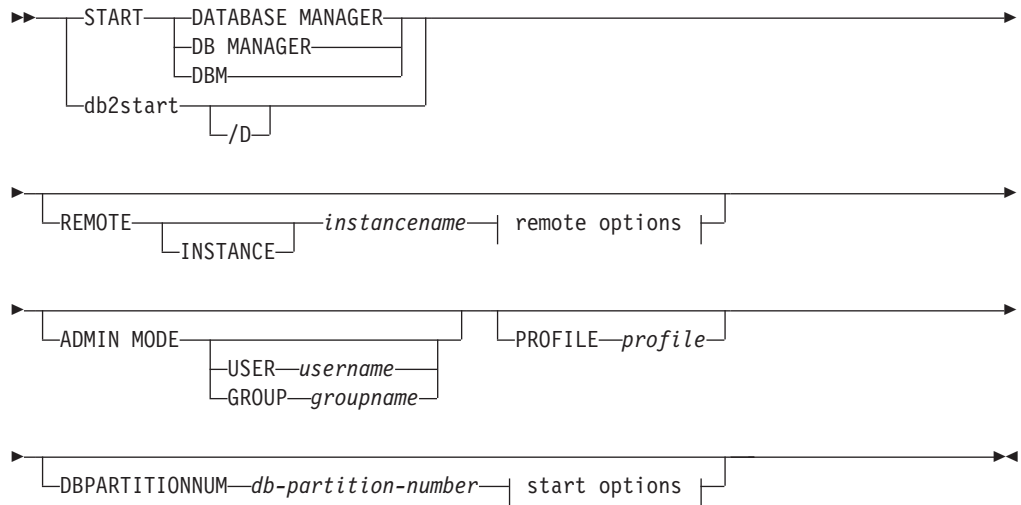
`ADD DBPARTITIONNUM` 開始オプションは、`sysadm` または `sysctrl` 権限のどちらかを必要とします。

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。

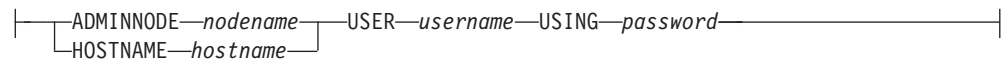
必要な接続

なし

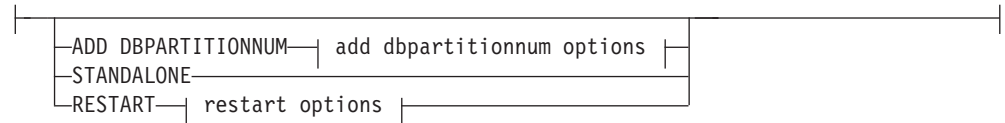
コマンド構文



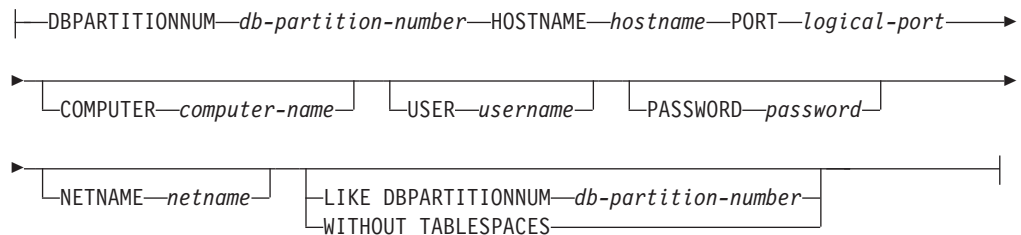
remote options:



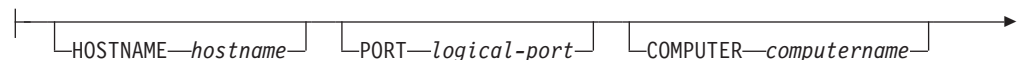
start options:

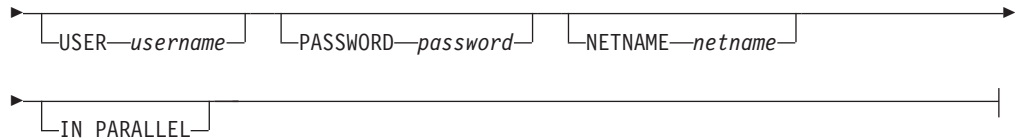


add dbpartitionnum options:



restart options:





コマンド・パラメーター

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。これは、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。

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。

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` ディレクトリー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。

DBPARTITIONNUM *db-partition-number*

開始するデータベース・パーティションを指定します。他のオプションが指定されていない場合、このデータベース・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。ADD

DBPARTITIONNUM が指定されていない場合、値はインスタンス所有者の

db2nodes.cfg ファイルに既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・パーティションが開始されます。

ADD DBPARTITIONNUM

インスタンス所有者の db2nodes.cfg ファイルに、*hostname* および *logical-port* 値とともに、新しいデータベース・パーティションを追加することを指定します。

hostname および *logical-port* 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・パーティション追加ユーティリティーが内部で実行され、追加されたデータベース・パーティションに既存のデータベースすべてが作成されます。データベース・パーティション追加後、db2stop が発行されるまで、db2nodes.cfg ファイルは新しいデータベース・パーティションで更新されません。db2stop に続く db2start が実行されるまで、データベース・パーティションは MPP システムの一部とは見なされません。

新しいノードにデータベース・パーティションが作成される場合、構成パラメーターはデフォルトに設定されます。

DBPARTITIONNUM *db-partition-number*

作成される新しいデータベース・パーティションのデータベース・パーティション番号の値を指定します。

HOSTNAME *hostname*

ADD DBPARTITIONNUM 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* で指定された値がデフォルトとなります。

IN PARALLEL

並列実行のための RESTART DATABASE コマンドを発行します。

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* のコンテナ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よう定義された SYSTEM TEMPORARY 表スペース (つまり 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) 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、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。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 (つまり 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) 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

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

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* と *logical-port* のどちらのパラメーターも指定されないと、データベース・マネージャーは、`db2nodes.cfg` で指定された *hostname* および *logical-port* 値を使用して再始動します。どちらかのパラメーター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。

HOSTNAME *hostname*

HOSTNAME オプションを RESTART パラメーターとともに使用する

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。データベース・パーティションをクラスター内のあるマシンから別のマシンに移したり、再開したりするには、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、`db2nodes.cfg` ファイルの *num* の値に対応する、*logical-port* の値がデフォルトとなります。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* を指定します。指定されていない場合、`db2nodes.cfg` ファイルの *db-partition-number* の値に対応する、*netname* の値がデフォルトとなります。

例

次に示すのは、データベース・パーティション 10、20、および 30 を使用する 3 データベース・パーティション・システムで発行された `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並列ノードでデータベース・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
```

新しいデータベース・パーティションの追加時には、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

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。

UNIX プラットフォームでは、`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。この信号が発行されると、進行中のすべての始動操作に割り込みが生じ、割り込みが生じた各データベース・パーティションから、`$HOME/sql1lib/log/db2start` にメッセージ (`SQL1044N`) が戻されます。`timestamp.log` エラー・ログ・ファイルに戻されます。既に開始済みのデータベース・パーティションには影響がありません。開始しているデータベース・パーティションに対して `CTRL+C` が押された場合、そのデータベース・パーティションを再始動する前に、そのデータベース・パーティションに対して `db2stop` が発行され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、開始に失敗した通信サブシステムがあっても、`db2start` コマンドまたは `NET START` コマンドから警告は戻されません。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.LOG` 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

バージョン 8 より前のバージョンとの互換性：

- キーワード `LIKE NODE` は、`LIKE DBPARTITIONNUM` の代わりとして使用できます。
- キーワード `ADDNODE` は、`ADD DBPARTITIONNUM` の代わりとして使用できます。
- キーワード `NODENUM` は、`DBPARTITIONNUM` の代わりとして使用できます。

START HADR

データベースの HADR 操作を開始します。

許可

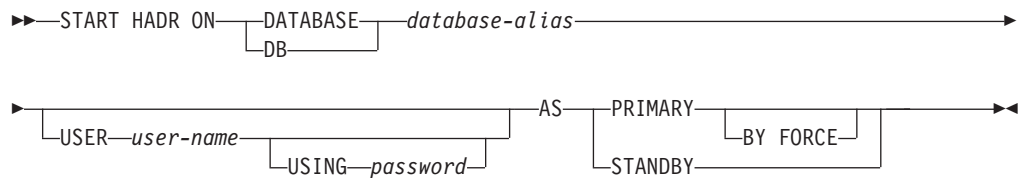
以下のいずれか。

- `sysadm`
- `sysctrl`
- `sysmaint`

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



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。

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** 設定値の影響を受けません。

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。(これは分割ブレイン、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活動化されます。	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活動化されます。	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

現行のデータベース・マネージャー・インスタンスを停止します。明示的に停止されない限り、データベース・マネージャーは、アクティブなまま続きます。データベースに接続されたアプリケーションがある場合、このコマンドはデータベース・マネージャーのインスタンスを停止しません。データベース接続はなくても、インスタンスへのアタッチ機構がある場合には、そのインスタンスへのアタッチ機構を強制的に使用して、データベース・マネージャーを停止させます。また、データベース・マネージャーを停止させる前に、処理中のデータベースの活動を非活動化します。

パーティション・データベース環境では、このコマンドは、特定のデータベース・パーティションまたはすべてのデータベース・パーティションにある現行のデータベース・マネージャーのインスタンスを停止します。すべてのデータベース・パーティションでデータベース・マネージャーを停止する場合、`db2nodes.cfg` 構成ファイルを使用して、各データベース・パーティションに関する情報を入手します。

このコマンドを使用して、`db2nodes.cfg` ファイルからデータベース・パーティションをドロップすることもできます (パーティション・データベース環境の場合のみ)。

このコマンドはクライアントでは無効です。

有効範囲

デフォルトおよびパーティション・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。

許可

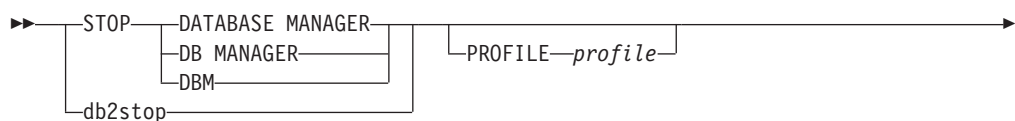
以下のいずれか。

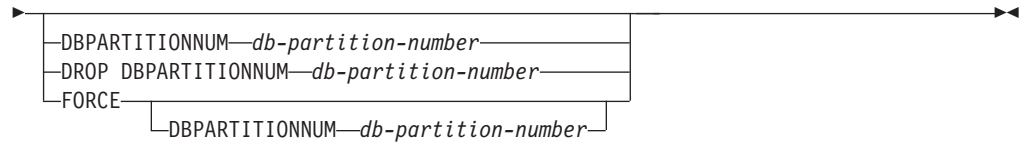
- `sysadm`
- `sysctrl`
- `sysmaint`

必要な接続

なし

コマンド構文





コマンド・パラメーター

PROFILE *profile*

パーティション・データベース環境のみ。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 sqllib ディレクトリーに常駐していなければなりません。

DBPARTITIONNUM *db-partition-number*

パーティション・データベース環境のみ。停止させるデータベース・パーティションを指定します。

有効な値は 0 から 999 で、その値が db2nodes.cfg ファイルに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。

DROP DBPARTITIONNUM *db-partition-number*

パーティション・データベース環境のみ。db2nodes.cfg 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。

この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。

このオプションを指定した場合、db2nodes.cfg ファイルにあるすべてのデータベース・パーティションが停止します。

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。

DBPARTITIONNUM *db-partition-number*

パーティション・データベース環境のみ。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た後で停止されるデータベース・パーティションを指定します。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。

例

次に示すのは、データベース・パーティション 10、20、および 30 を使用する 3 パーティション・システムで発行された 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。
- 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 プラットフォームでは、構成パラメーターの **start_stop_time** データベース・マネージャー の指定値に達すると、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/sql1lib/log/db2stop にメッセージ SQL6037N が戻されます。timestamp.log エラー・ログ・ファイルに戻されます。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 コマンドは、データベース・マネージャーを終了するための方法としては使用しないでください。このコマンドは、何の制御もクリーンアップ処理も行わずに突然データベース・マネージャーのプロセスを終了してしまいます。

STOP HADR

データベースの HADR 操作を停止します。

許可

以下のいずれか。

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

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 hadr_db_role 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。

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。AS PRIMARY オプションを指定した明示的な START HADR 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。 STOP HADR 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、STOP HADR コマンドと START HADR 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 <code>hadr_db_role</code> 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。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

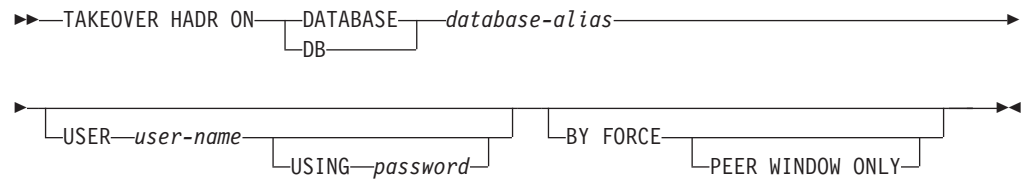
以下のいずれか。

- `sysadm`
- `sysctrl`
- `sysmaint`

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



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、結果として分割ブレイン 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。

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、PEER WINDOW ONLY オプションを指定せずに TAKEOVER BY FORCE 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"> 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。 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。
ローカル・キャッチアップ またはリモート・キャッチアップ	いいえ	エラー・メッセージが戻されます。
ローカル・キャッチアップ またはリモート・キャッチアップ	はい	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
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"> 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。 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、TAKEOVER HADR コマンドを再発行してください (BY FORCE オプションについての下記のガイドラインを参照)。 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、BY FORCE オプションを使って TAKEOVER HADR 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、BY FORCE オプションが必要です。
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、BY FORCE オプションを指定して TAKEOVER HADR コマンドを実行することを推奨します。</p>
リモート・キヤッチアップ・ペンディング	いいえ	エラー・メッセージが戻されます。

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・キヤッチアップ・ペンディング	はい	スタンバイが 1 次になります。

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。

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 の複数の論理ノード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システム・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。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 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) からデータベースを登録解除するのに使用します。

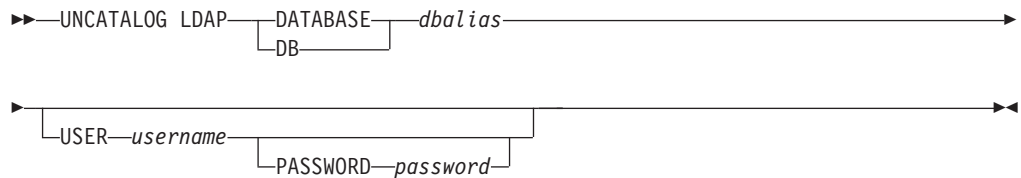
許可

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・ユーザーの信用証明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・ユーザーの信用証明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。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 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

保守またはその他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 は、シャットダウンしたりデータベースを再開したりせずにユーザー・アクセスを回復します。

特に指定がない限り、*sysadm*、*sysmaint*、または *sysctrl* 以外のユーザーは、静止中のデータベースにアクセスできません。そのため、静止データベースの一般アクセスを回復するには、UNQUIESCE が必要です。

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

```
db2 unquiesce db
```

このコマンドは、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。

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。

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**

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。

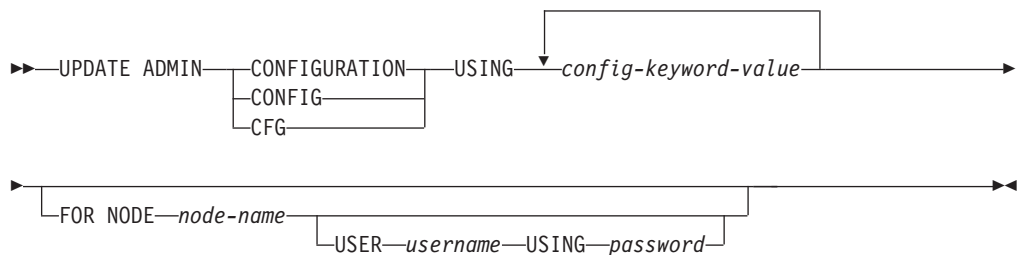
許可

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・インディケーターのアラート構成設定を更新します。

許可

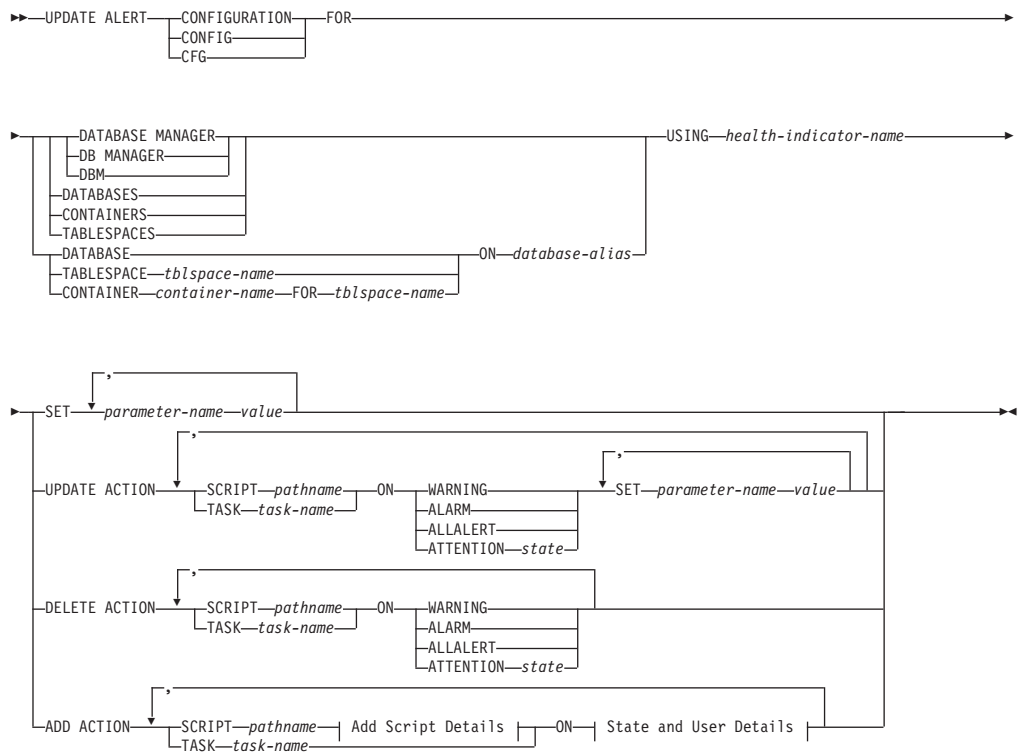
以下のいずれか。

- `sysadm`
- `sysmaint`
- `sysctrl`

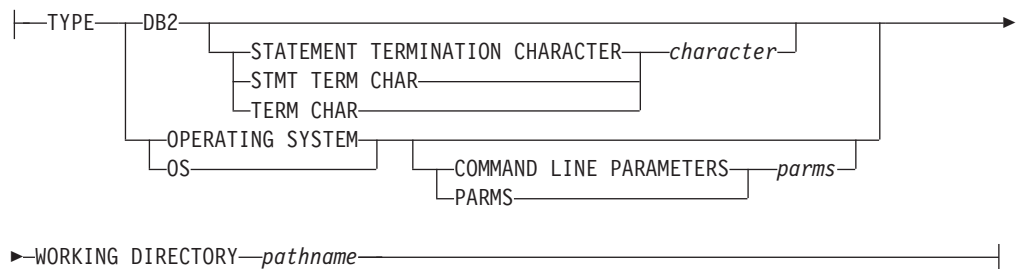
必要な接続

インスタンス。明示的なアタッチは必要ありません。

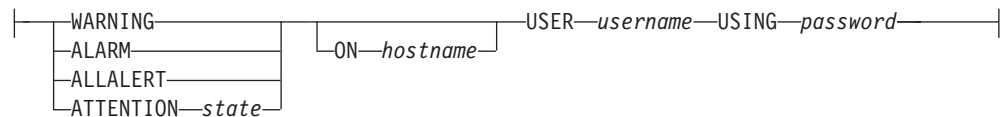
コマンド構文



Add Script Details:



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。

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
```

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 は以下のどれかになります。

- SCRIPTTYPE
- WORKINGDIR
- TERMCHAR
- CMDLINEPARMS

有効なタイプは OS または DB2。

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す。

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*]

指定した条件で、*name* という名前の新規タスクが追加され、実行されるように指定します。

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 (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)。

ATTENTION *state*

データベース・ヘルス・インディケータの一部の状態の有効な数値は、ADD ACTION SCRIPT CLP コマンド・オプションの例として以下に示されています。

- 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)

状態ベースの追加のヘルス・インディケータは、ヘッダー・ファイル `sqlmon.h` および `sqlutil.h` で定義されます。

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.log`
- 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)
- Query Patroller
- 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

以下のいずれか。

- `sysadm`
- `sysctrl`

必要な接続

なし

コマンド構文

```
▶▶—UPDATE ALTERNATE SERVER FOR—

|          |
|----------|
| DATABASE |
| DB       |

—database-alias—USING—▶▶
```

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
```

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` が戻されます。
- 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。

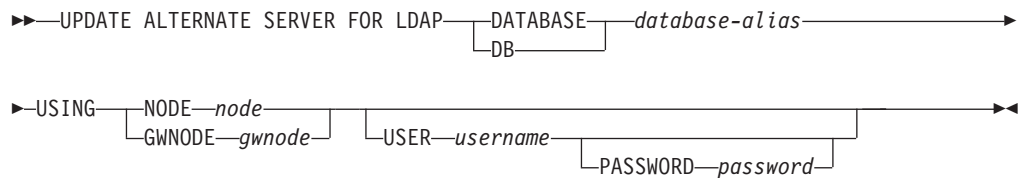
許可

`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



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・ユーザーの信用証明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。

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、DB2 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。

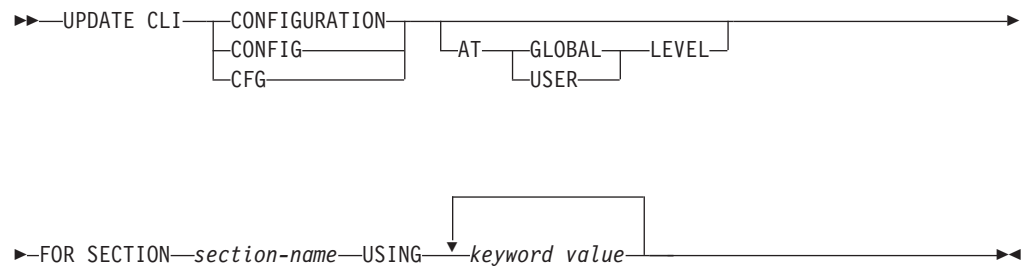
許可

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

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。

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。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。

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文字列である場合には、文字列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、CA (構成アシスタント) または ODBC 構成ユーティリティーのどちらか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。

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 (またはシステム・デフォルトは DB2OPTIONS の中でオーバーライドされます)。

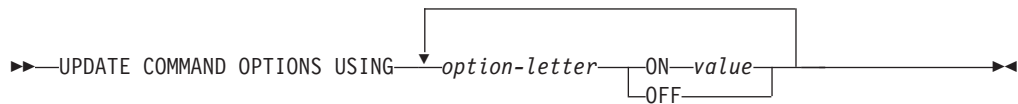
許可

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

USING *option-letter*

次の *option-letters* を設定できます。

- a** SQLCA の表示
- c** SQL ステートメントの自動コミット
- d** XML データの XML 宣言を表示します。
- e** SQLCODE/SQLSTATE の表示
- 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** INSERT、DELETE、UPDATE、または MERGE ステートメント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管
- 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) **contact_host** 構成パラメーターの設定は、リストがローカルかグローバルかを判別します。

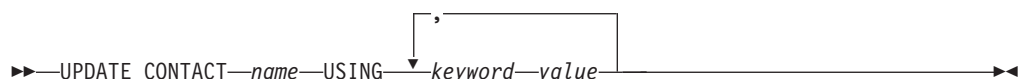
許可

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

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) **contact_host** 構成パラメーターの設定は、リストがローカルかグローバルかを判別します。

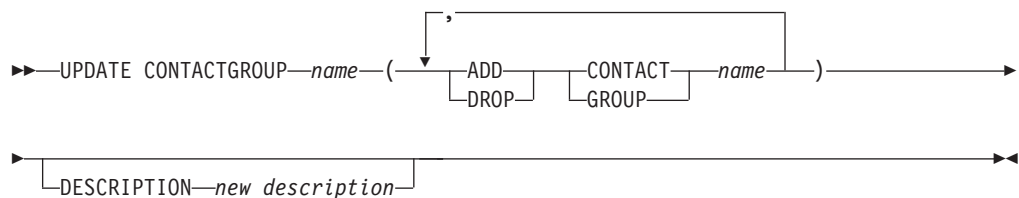
許可

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

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・パーティションを更新します。ただし、DBPARTITIONNUM が 1 つのデータベース・パーティションのみ更新するように指定されている場合は除きます。

許可

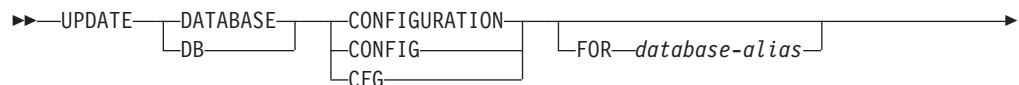
以下のいずれか。

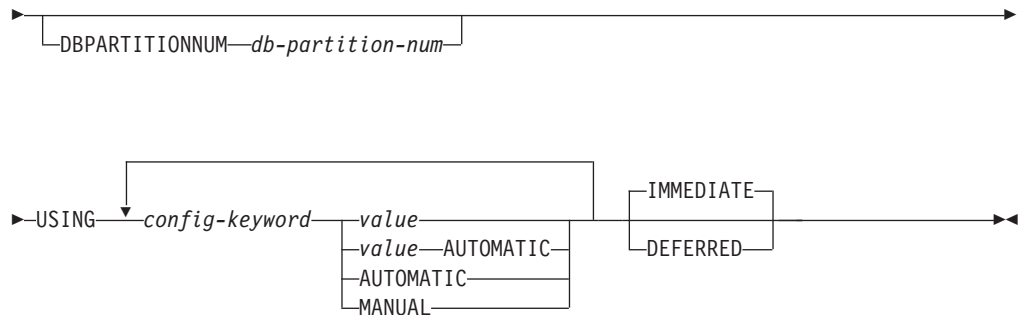
- *sysadm*
- *sysctrl*
- *sysmaint*

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

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。

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。

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。

DBPARTITIONNUM *db-partition-num*

データベース構成の更新が特定のデータベース・パーティションに適用される場合、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、更新はすべてのデータベース・パーティションに対して有効です。

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。

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。

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。

例

DPF (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

この例では、SAMPLE という名前のデータベースについて、データベース構成パラメーター **MAXAPPLS** を 10 から 50 に更新する方法を示します。

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 DPF 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 dbpartitionnum 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。

DPF 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
```

MAXAPPLS を dbpartitionnum 30 でのみ更新するには、以下のコマンドを発行してください。

```
db2 update db cfg for sample dbpartitionnum 30 using maxappls 50
```

または

```
export DB2NODE=30
db2 update db cfg for sample 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 logretain recovery
```

logretain 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、『zellmart』というデータベースを含む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』に変更するには (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)、次を使用します。

```
db2 update db cfg for zellmart using logretain recovery
```

logretain 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"
```

上記と同じ例を使用して、ロギング・モードを 1 つだけの特定のパーティション (30) に更新するには、次を使用します。

```
db2 update db cfg for zellmart dbpartitionnum 30 using logretain recovery
```

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、**logretain** 値のみを表示できます。

```
db2_all ";db2 get db cfg for zellmart | grep -i logretain"
```

DB2 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。

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。

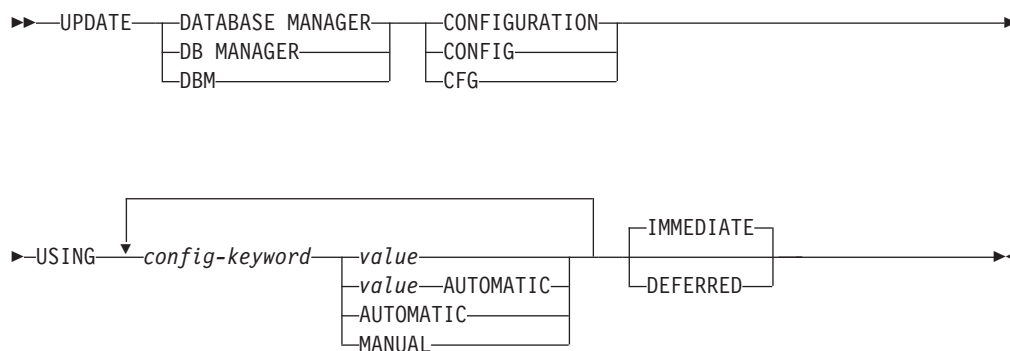
許可

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

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:

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。

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。 `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。

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。

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。

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

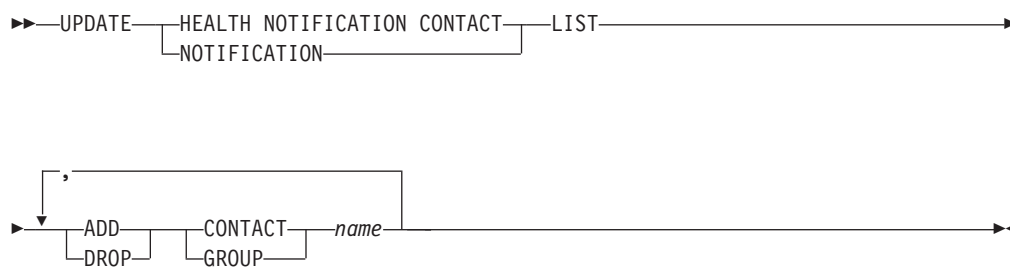
以下のいずれか。

- 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

履歴ファイル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

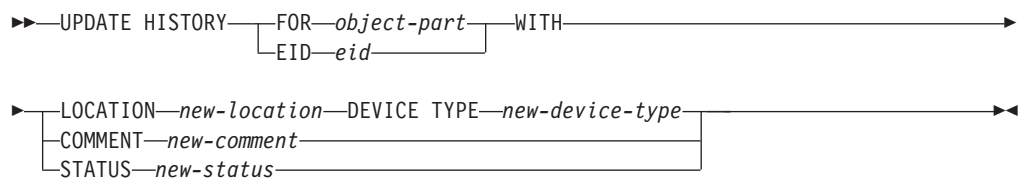
以下のいずれか。

- *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。

- | | |
|----------|---------------------|
| D | ディスク |
| K | ディスケット |
| T | テープ |
| A | Tivoli ストレージ・マネージャー |
| F | スナップショット・バックアップ |
| U | ユーザー出口 |
| 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履歴ファイル項目は、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、またはリカバリー履歴ファイルの自動ブ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・バックアップの履歴ファイル項目を更新するには、次のように入力します。

```
db2 update history for 199704131000000001 with
location /backup/dbbackup.1 device type d
```

使用上の注意

データベース履歴ファイル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。

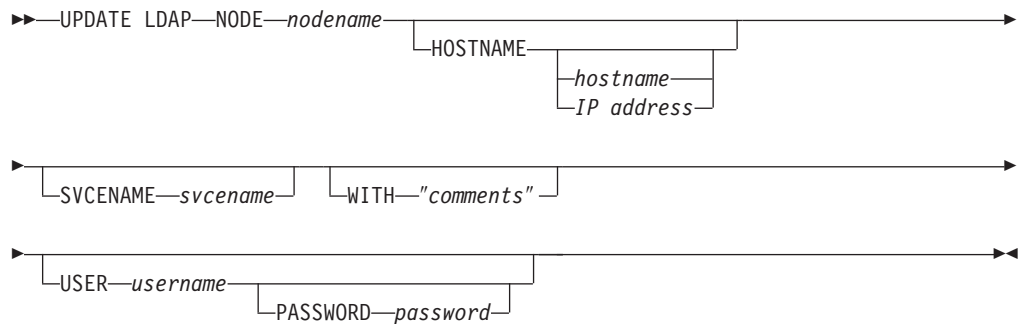
許可

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・ユーザーの信用証明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。

許可

以下のいずれか。

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

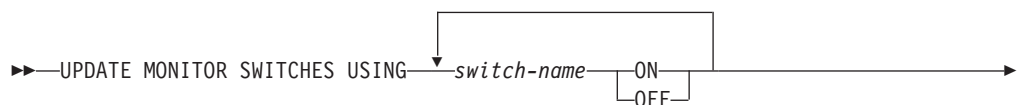
必要な接続

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



AT DBPARTITIONNUM *db-partition-number*
GLOBAL

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・アクティビティー情報

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 DBPARTITIONNUM *db-partition-number*

モニター・スイッチの状況を表示するデータベース・パーティションを指定します。

GLOBAL

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・パーティションの集約結果を戻します。

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。

互換性

バージョン 8 より前のバージョンとの互換性：

- キーワード NODE は、DBPARTITIONNUM の代わりとして使用できます。

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

以下のいずれか。

- *sysadm* または *dbadm*
- 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。
- 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。

第 5 章 システム・コマンド

dasauto - DB2 Administration Server の自動始動

DB2 Administration Server の自動始動を有効または無効にします。

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。

dasCRT - DB2 Administration Server の作成

DB2 Administration Server (DAS) は、コントロール・センターおよび構成アシスタントなどの DB2 ツールのサービスをサポートします。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。dasCRT 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 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せん。

許可

Root 権限

必要な接続

なし

コマンド構文

```
▶▶ dasdrt -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。

dasdrop - DB2 Administration Server の除去

Linux および UNIX ベースのシステムでのみ、DB2 Administration Server (DAS) を除去します。 Administration Server は、コントロール・センターおよび構成アシスタントなどの DB2 ツールのサービスをサポートします。 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。

許可

Root 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。

dasmigr - DB2 Administration Server のマイグレーション

システム上の DB2 Administration Server (DAS) を DB2 データベース・システム (DB2 データベース・システムの現行バージョンへのマイグレーションがサポートされている) の以前のバージョンから、dasmigr が発行されるパスに関連した DB2 データベース・レベルでの DB2 データベース・システムの現行バージョンへマイグレーションします。

DAS を 1 つの DB2 データベース・システムのインストール・ロケーションから、DB2 データベース・システムの同じバージョン内で別のロケーションに移動するには、dasupdt コマンドを使用します。以前のバージョンの DB2 データベース・システムの DAS は、現行バージョンの DB2 データベース・システムのインスタンスの管理には使用できません。

Linux および UNIX システムでは、このユーティリティーは DB2DIR/instance ディレクトリにあります。Windows オペレーティング・システムでは、これは 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・ロケーションを表します。

許可

UNIX オペレーティング・システムでは Root でアクセスし、Windows オペレーティング・システムではローカル管理者権限でアクセスします。

必要な接続

なし

コマンド構文

Linux および UNIX システムの場合

```
▶▶--dasmigr [ -d ]
```

Windows オペレーティング・システムの場合

```
▶▶--dasmigr [ -h ] [ -p—path override ]
```

コマンド・パラメーター

Linux および UNIX システムの場合

-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。

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 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
```

dasupdt - DAS の更新コマンド

Linux および UNIX システムで、このコマンドは、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。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のインストール先)。このユーティリティーを使用して、両方が DB2 データベース・システムと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。

システム上の DAS が、installFixPack によって更新される DB2 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。

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マイグレーションするには、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。このユーティリティーは、DB2IPATH%bin ディレクトリーにあります (DB2IPATH は DB2 データベース製品の現行バージョンのインストール先)。

許可

Linux および UNIX システムでは root でアクセスし、Windows オペレーティング・システムではローカル管理者権限でアクセスします。

必要な接続

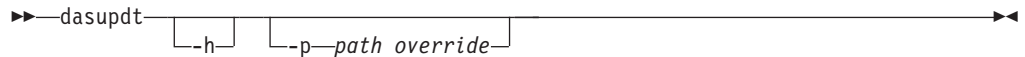
なし

コマンド構文

Linux および UNIX 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



コマンド・パラメーター

Linux および UNIX 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 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
```

db2_deinstall - DB2 製品またはフィーチャーのアンインストール

以下のように、インストールされている DB2 製品または DB2 フィーチャーをすべてアンインストールします。

- 特定の DB2 インストール・パスから db2_deinstall を実行すると、その同じパスからすべてまたは特定のフィーチャーをアンインストールできます。
- DB2 メディアから db2_deinstall 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をアンインストールできます。

このコマンドは、Linux および UNIX のシステムでのみ利用できます。

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 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 製品のアンインストールにのみ使用できます。

許可

ルート・インストールには root 権限が必要です。非ルート・インストールの場合、非ルート・インストールを所有するユーザー ID でログイ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_deinstall [-F feature-name] [-a] [-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。

指定した DB2 フィーチャーを除去した後に、DB2 のインストール場所に関連した DB2 インスタンスを手動で更新する必要があります。

1 つの事例を除いて、-a と組み合わせて使用できません。IBM Tivoli System Automation for Multiplatforms Base Component (TSAMP) フィーチャーを除去する際に、root 権限がある場合は、-F TSAMP と -a を組み合わせて使用できます。この場合、TSAMP と DB2 の両方が一緒に除去されません。

DB2 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*

このオプションは、非ルート・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 現在場所にあるすべてのインストール済み DB2 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。IBM Tivoli System Automation for Multiplatforms Base Component (TSAMP) フィーチャーを除去する際に、root 権限がある場合は、-F TSAMP と -a を組み合わせて使用できます。この場合、TSAMP と DB2 の両方が一緒に除去されます。

非ルート・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。

-b このオプションは、DB2 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 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。ルート・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)。

非ルート・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` は非ルート・インストールを所有するユーザー ID)。IBM Tivoli System Automation for Multiplatforms Base Component (TSAMP) フィーチャーを除去する際には、SA MP Base Component のインストール・ログ・ファイルは、DB2 ログ・ファイルと同じディレクトリーに入れられます。TSAMP を除去すると、すべての DB2 も除去されます。

-t *trace-file*

デバッグ・モードをオンにします。デバッグ情報は、*trace-file* として指定されたファイル名に書き込まれます。

-h | -? 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
```

使用上の注意

`db2_deinstall -a -f sqllib` を実行すると、`$HOME/sqllib` ディレクトリーが除去されます。このコマンドを実行する前に、このディレクトリーから、保管する必要のあるすべてのファイルのバックアップを取っていることを確認してください。

db2_install - DB2 製品のインストール

DB2 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 Linux および UNIX システムでのみ使用できます。

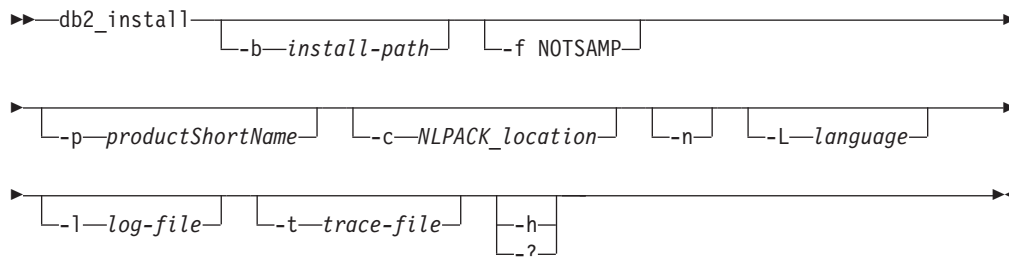
許可

ルート・インストールには root 権限が必要です。非ルート・インストールの場合、非ルート・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



コマンド・パラメーター

-b *install-path*

DB2 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 製品の非ルート・インストールの場合は必須ではありませんが、ルート・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。非ルート・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/sql1lib` になる必要があります。ルート・インストールと非ルート・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**

SA MP Base Component をインストールまたは更新しないことを指定します。

-p *productShortName*

インストールする DB2 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ときには必須です。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 製品イメージ内になく、DB2 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。

-L language

各国語サポートを指定します。DB2 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、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。ルート・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)。非ルート・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` は非ルート・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 Base Component (SA MP Base Component) 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 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
```

使用上の注意

非ルート DB2 インストールのデフォルトのログおよびトレースのファイル名にはユーザー ID が含まれているので、DB2 のルート・インストールの結果のファイルは上書きされません。

非ルート・インストールのデフォルトのファイル名は以下のようになります (`userID` は非ルート・インストールを実行する非ルート・ユーザーのユーザー名)。

- `/tmp/db2_install_userID.log`
- `/tmp/db2_install_userID.err`

- /tmp/db2_install_userID.trc

非ルート・インストールの完了後、ログ・ファイルは \$DB2DIR/install/logs ディレクトリーにコピーされます。このディレクトリーは、ルート・インストールの場合と同じディレクトリーです。

追加情報については、-l logfile コマンド・パラメーターを参照してください。

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 db2tccpm 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
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。

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。

Linux および UNIX ベースのシステムの場合、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 実行可能ファイルは DB2PATHbin 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先)。

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ません。

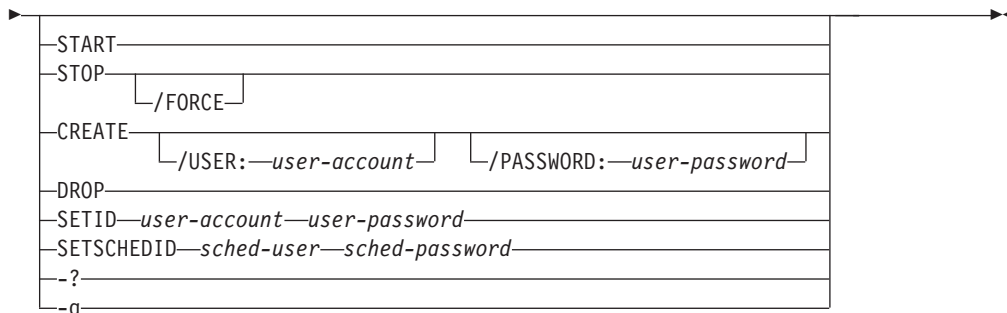
Windows オペレーティング・システムではローカル管理者。

必要な接続

なし

コマンド構文

▶▶—db2admin—————▶▶



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。指定されるユーザー・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、`dasrcrt` 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时候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) を使用して保管した、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` ディレクトリにあります。Windows オペレーティング・システムでは、これは `sqllibbin` にあります。

許可

なし

必要な接続

なし

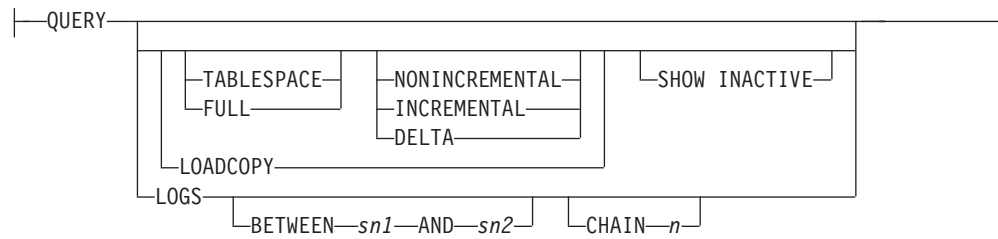
コマンド構文

```
▶▶ db2adutl [db2-object-options] [access-control-options]
```

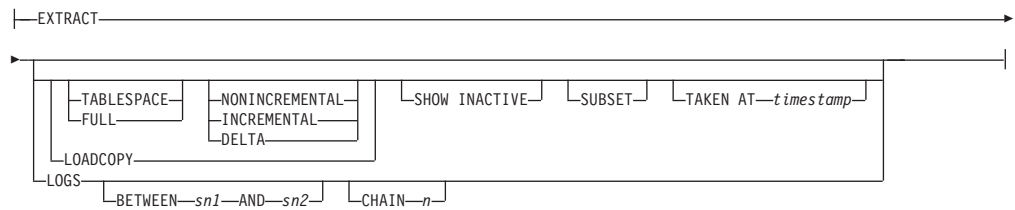
db2-object-options:

```
[QUERY-options] [EXTRACT-options] [DELETE-options] [VERIFY-options] [COMPROPTS—decompression-options] [VERBOSE] [COMPRLIB—decompression-library] [DATABASE—database_name] [DBPARTITIONNUM—db-partition-number] [DB] [PASSWORD—password] [NODENAME—node_name] [OWNER—owner] [WITHOUT PROMPTING]
```

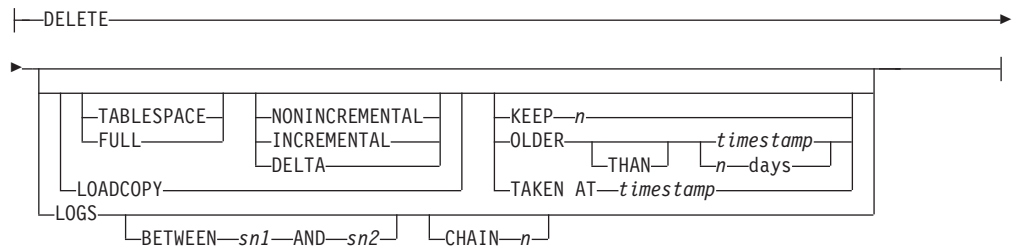
QUERY-options:



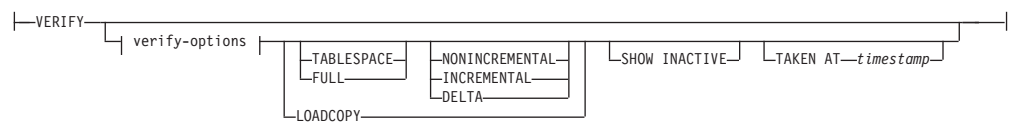
EXTRACT-options:



DELETE-options:



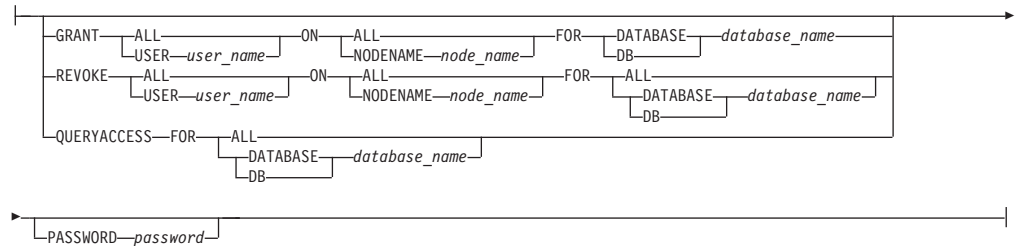
VERIFY-options:



verify-options:



access-control-options:



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。

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。

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、イメージの検証は実行しません。

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 <tbodyID> <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 <tbodyID> <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、

```
0 <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*

タイム・スタンプを基準としてバックアップ・イメージを指定します。

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* または *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。

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 '@' なら、データの残りの部分は、`DB2` によって、サーバー上に存在するファイルの名前として解釈されます。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。

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。

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。

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. The following is sample output from the command `db2 backup database`

```
rawsampl use tsm
```

```
Backup successful. このバックアップのタイム・スタンプは 20031209184503 です。
```

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, DB Partition Number: 0,

Taken at 2003-12-09-18.46.13

Log file: S0000051.LOG, Chain Num: 0, DB Partition Number: 0,

Taken at 2003-12-09-18.46.43

Log file: S0000052.LOG, Chain Num: 0, DB Partition Number: 0,

Taken at 2003-12-09-18.47.12

Log file: S0000053.LOG, Chain Num: 0, DB Partition Number: 0,

Taken at 2003-12-09-18.50.14

Log file: S0000054.LOG, Chain Num: 0, DB Partition Number: 0,

Taken at 2003-12-09-18.50.56

Log file: S0000055.LOG, Chain Num: 0, DB Partition Number: 0,

Taken at 2003-12-09-18.52.39

2. 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 DB Partition Number: 0 Sessions: 1

Do you want to delete this file (Y/N)? y

Are you sure (Y/N)? y

Retrieving INCREMENTAL DATABASE BACKUP information.

No INCREMENTAL DATABASE BACKUP images found for RAWSAMPL

Retrieving DELTA DATABASE BACKUP information.

No DELTA DATABASE BACKUP images found for RAWSAMPL

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に発行された 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 RAWSAMPL

Retrieving FULL DATABASE BACKUP information.

1 Time: 20031209184403, Oldest log: S0000050.LOG, Sessions: 1

Retrieving INCREMENTAL DATABASE BACKUP information.

No INCREMENTAL DATABASE BACKUP images found for RAWSAMPL

Retrieving DELTA DATABASE BACKUP information.

No DELTA DATABASE BACKUP images found for RAWSAMPL

Retrieving TABLESPACE BACKUP information.

No TABLESPACE BACKUP images found for RAWSAMPL

Retrieving INCREMENTAL TABLESPACE BACKUP information.

No INCREMENTAL TABLESPACE BACKUP images found for RAWSAMPL

Retrieving DELTA TABLESPACE BACKUP information.

No DELTA TABLESPACE BACKUP images found for RAWSAMPL

Retrieving LOCAL COPY information.

No LOCAL COPY images found for RAWSAMPL

Retrieving log archive information.

Log file: S0000050.LOG, Chain Num: 0, DB Partition Number: 0,
Taken at 2003-12-09-18.46.13
Log file: S0000051.LOG, Chain Num: 0, DB Partition Number: 0,
Taken at 2003-12-09-18.46.43
Log file: S0000052.LOG, Chain Num: 0, DB Partition Number: 0,
Taken at 2003-12-09-18.47.12
Log file: S0000053.LOG, Chain Num: 0, DB Partition Number: 0,
Taken at 2003-12-09-18.50.14
Log file: S0000054.LOG, Chain Num: 0, DB Partition Number: 0,
Taken at 2003-12-09-18.50.56
Log file: S0000055.LOG, Chain Num: 0, DB Partition Number: 0,
Taken at 2003-12-09-18.52.39

3. 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

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、何のバックアップ・イメージ・タイプを操作に組み込むかを制限できます。

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。
- 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。

互換性

バージョン 8 より前のバージョンとの互換性 :

- キーワード NODE は、DBPARTITIONNUM の代わりとして使用できます。

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。

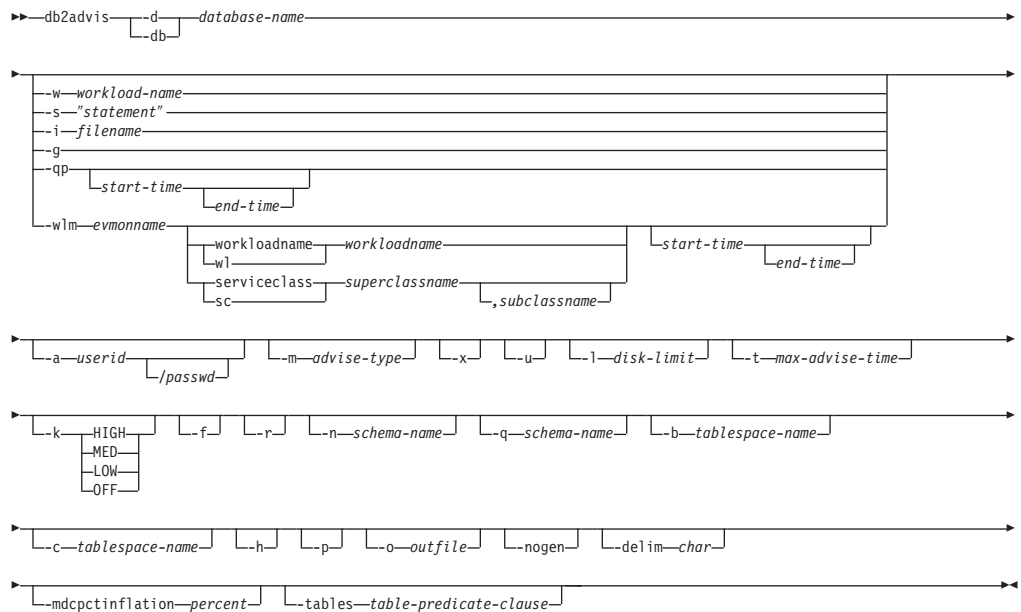
許可

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、-qp、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、-qp、-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**、**-qp**、**-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**、**-qp**、**-w** の各オプションと一緒に指定できません。

-qp ワークロードを DB2 Query Patroller から取り込むことを指定します。*start-time* および *end-time* オプションは、DB2QP.TRACK_QUERY_INFO 表の *time_completed* フィールドとの比較検査に使用されるタイム・スタンプです。*start-time* および *end-time* タイム・スタンプが指定されない場合、表の *completion_status* 列のすべての行は「D」(完了という意味)になります。*start-time* だけを指定した場合、TIME_COMPLETED が *start-time* の値より大または等しい行が戻されます。それに加えて *end-time* 値を指定した場合には、TIME_COMPLETED が *end-time* の値以下の行だけがさらに限定されて戻されます。このオプションは、**-w**、**-wlm**、**-s**、**-i**、または **-g** の各オプションと一緒に使用できません。

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。

-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。コマンドで区切ります。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で指定できます (大/小文字のどちらも可能)。例えば、db2advise -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) 表に変換するようにとの推奨、または表のクラスタリング索引を作成するようにとの推奨。

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어의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、**-qp** および **-g** のワークロードが ADVISE_WORKLOAD 表に保管され、最終推奨を使用するワークロード照会のプランが Explain 表に保管されます。
- 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* に保管します。
- 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。
- mdcpcinflation** *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。このコマンド・パラメーターは、DB2 バージョン 9.5 フィックスバック 2 以降で使用できます。
- 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。db2advisee 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、DB2 バージョン 9.5 フィックスバック 2 以降で使用できます。このコマンド・パラメーターは、新規 MQT に関する推奨には適用されません。

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。

```
db2advisee -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。

```
db2advise -d prototype -w production -l 53 -t 20
```

3. 次の例では、db2advise.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。

```
db2advise -d sample -f db2advise.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。

```
db2advise -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。

```
db2advise -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。

```
db2advise -d sample -type C -disklimit 100
-tables "TABNAME IN ('EMPLOYEE','DEPT')" -mdcpcctinflation 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。

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。

```
db2advise -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 db2advise エラー情報は db2diag.log にも入っています。

アドバイザの実行が始まると、アドバイザを識別する行が ADVISE_INSTANCE 表に含まれるようになります。メイン・アドバイザ行は、アドバイザの開始時間を示す START_TIME によって識別されます。この行の STATUS は「STARTED」です。

db2advise コマンドを発行した結果、「Cannot insert into DB2ADVISE_INSTANCE」というエラーが発生した場合には、db2advise.bnd をバインドし、**-I** オプションを使って db2advise 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、次のコマンドを発行します。

```
db2 bind db2advise.bnd blocking all grant public
```

アドバイザが完了すると、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、アドバイザは正常に実行されました。STATUS がまだ「STARTED」で、実行中の db2advise プロセスが存在しない場合は、アドバイザが異常終了しました。STATUS に「EX」が含まれる場合、アドバイザの失敗原因を判別するための「SQLCODE」も表示されます。

-I disk-limit オプションを指定しない場合、GET_DBSIZE_INFO ストアド・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、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。

db2audit - 監査機能管理者用ツール

DB2 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ーを別々のログに個別に記録します。

DB2 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/書き込み許可が設定されています。デフォルトでは、その許可はインスタンスの所有者にとってのみの読み取り/書き込み許可です。それらの許可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。

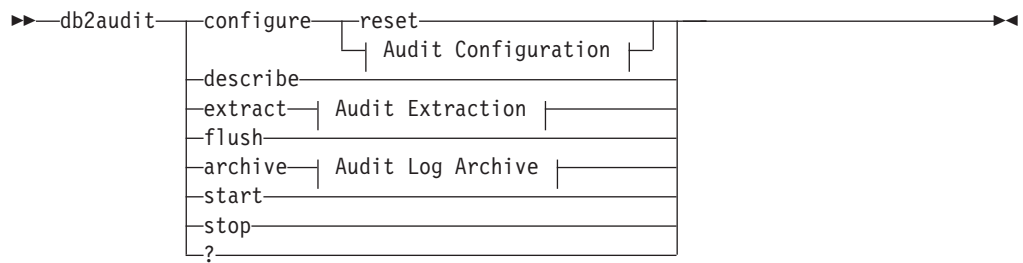
許可

SYSADM

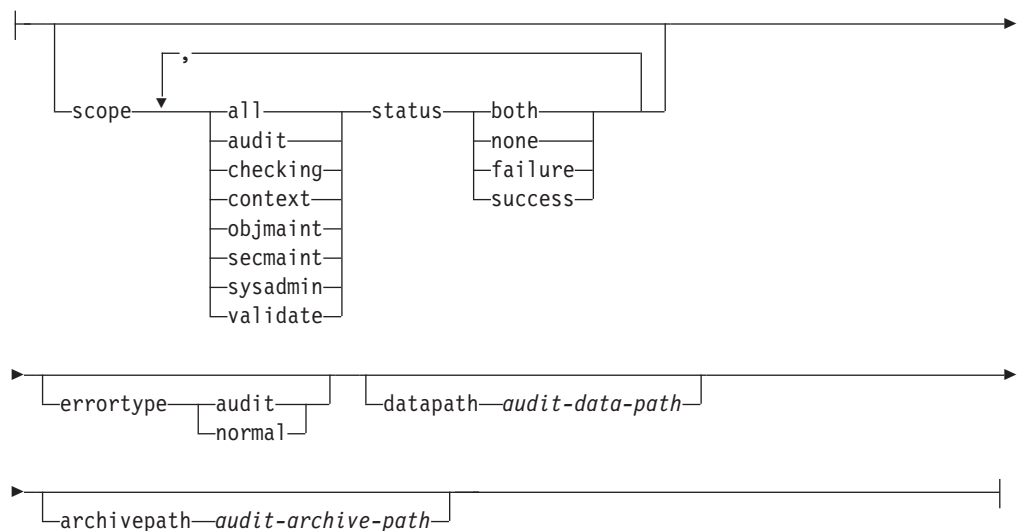
必要な接続

なし

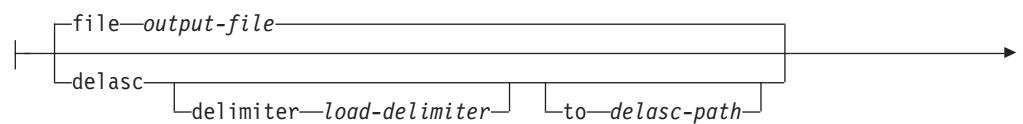
コマンド構文

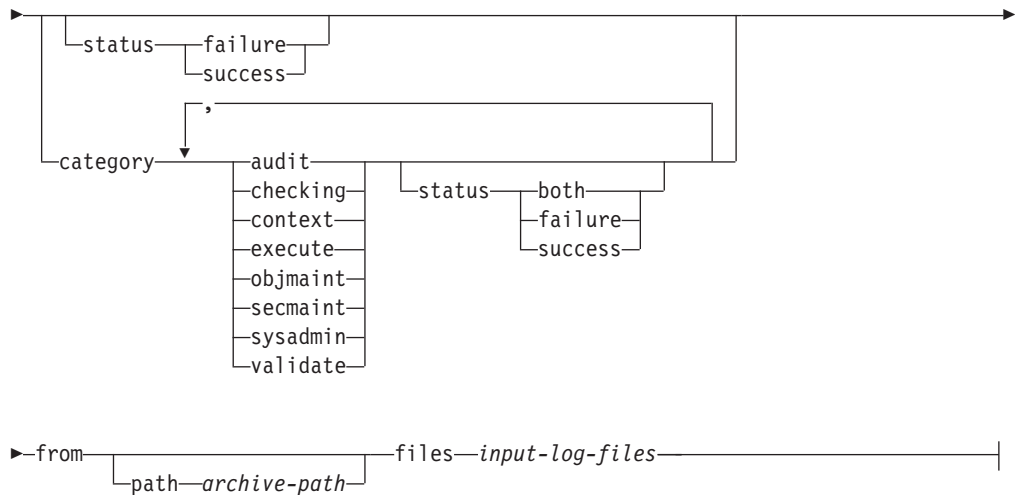


Audit Configuration:

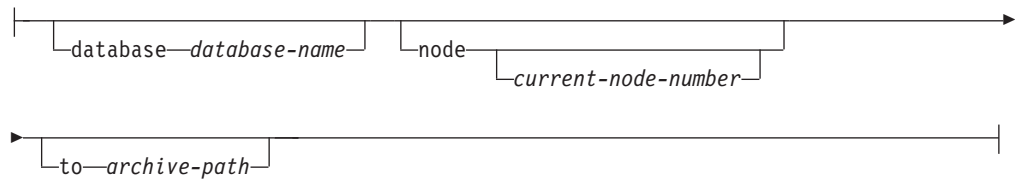


Audit Extraction:





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す (`scope` は `context` を除くすべてのカテゴリ、各カテゴリの `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、どのカテゴリを監査するか、および対象となる各カテゴリ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。

none このカテゴリ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だけが変更されます。他のすべてのカテゴリ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。

通常 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 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 `instance path¥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も、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持っている必要があります。DPF システムでは、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、そうする必要の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、各ノード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/Linux では 971 バイト、Windows 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。DPF 環境では、各ノードに同じデータ・パスが使用されます。特定のノード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。こうすれば、ストレージ・パスにノード番号が反映され、結果として生成されるパス名は各データベース・パーティションごとに異なります。データベース・パーティション式については、『自動ストレージ・データベース』を参照してください。

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。DPF 環境では、すべてのノード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許可を持っている必要があります。パスの最大長は、UNIX/Linux では 971 バイト、Windows 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"  
監査データ・パス: "/auditdata"  
監査アーカイブ・パス: "/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 形式になります。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られます。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 0xff) 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ます。

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。

node

現在のノード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、**node_number** モニター・エレメントによって現在のノードを表すことを示します。これは DPF システムでのみ必要です。

current-node-number

現在、実行対象となっているノードを、db2audit 実行可能プログラムに示します。 **DB2NODE** 環境変数に現在のノードが含まれない場合には、このパラメーターが必要です。

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.node_number[.YYYYMMDDHHMMSS]
- データベース・ログの場合は
db2audit.db.dbname.log.node_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。 DPF 以外の環境でのノード番号は 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。

node オプションが指定されない場合、すべてのノード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、アーカイブは各ノードに対して実行されます。 **archive** の実行対象のノードを示すには **node** 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ノード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、DPF 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。Windows 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があります。監査のために使用できるイベントのカテゴリの説明 (下記) において、各カテゴリの

名前は、カテゴリーのタイプを識別するために使用される 1 語のキーワードになっています。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、それらを実行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。
 - ユーザー検証 (**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。
 - 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・パラメータでは、応答セットからフェッチする行の数、出力ファイルや標準出力に送信するフェッチ済み行の数、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には、名前を付けることができます。

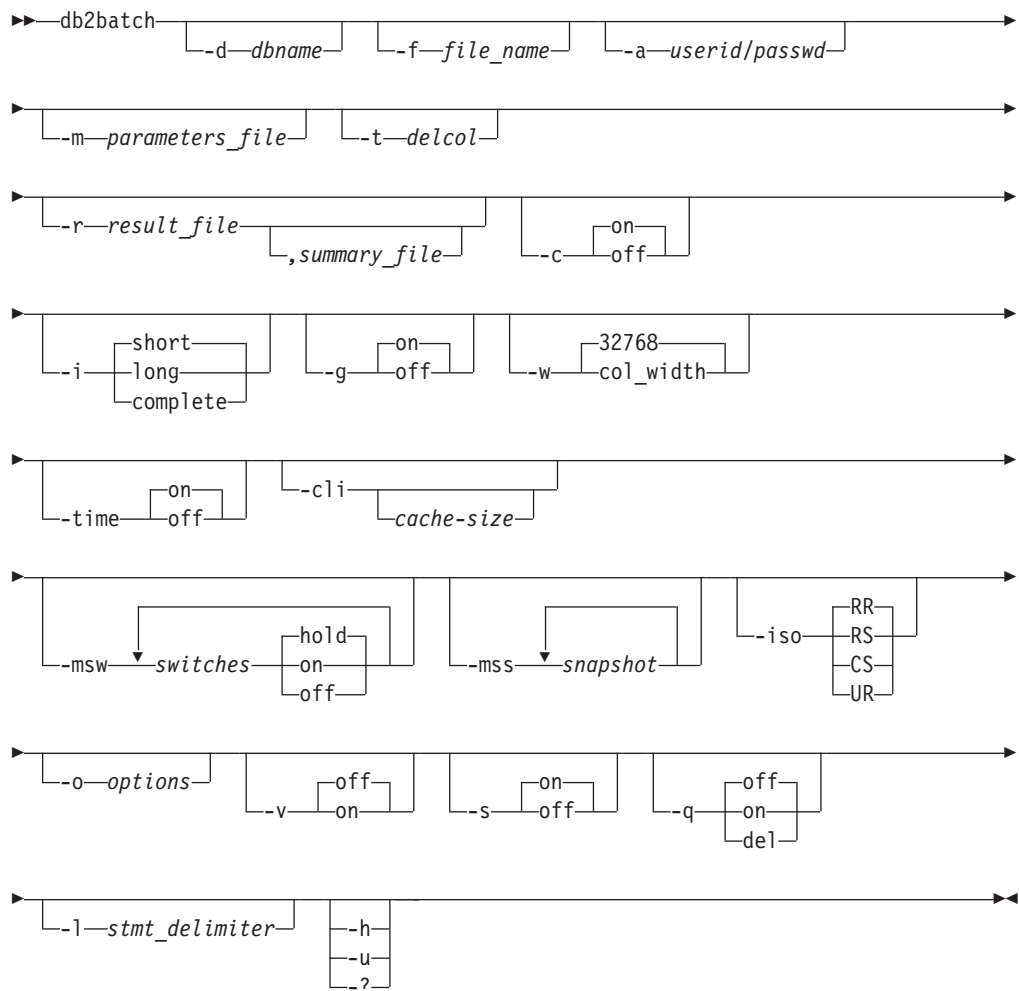
許可

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。

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ストリングを単一または二重引用符で区切った場合、そのストリングはコメントではなく、ストリング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、下記の `-m` オプ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。) バッファー・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

照会結果が入る出力ファイル。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。デフォルトは標準出力です。

-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オーバーヘッドを含めて、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切り捨てられます。この設定値を増やすと、警告 CLI0002W を抑止して、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 のいずれも指定できます。特殊なスイッチ all は、これらすべてのスイッチを設定します。指定する各スイッチごとに、以下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、dcs_applinfo_all、db2、dbase、dbase_all、dcs_dbase、dcs_dbase_all、dbase_remote、dbase_remote_all、agent_id、dbase_appls、appl_all、dcs_appl_all、dcs_appl_handle、dcs_dbase_appls、dbase_appls_remote、appl_remote_all、dbase_tables、appl_locks_agent_id、dbase_locks、dbase_tablespace、bufferpools_all、dbase_bufferpools、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)。

-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。デフォルトは -1 (現在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。

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 コマンド

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。

1. データベースをリストアするのに使用できる DB2 スナップショット・バックアップのリストを取得する
2. 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
3. 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

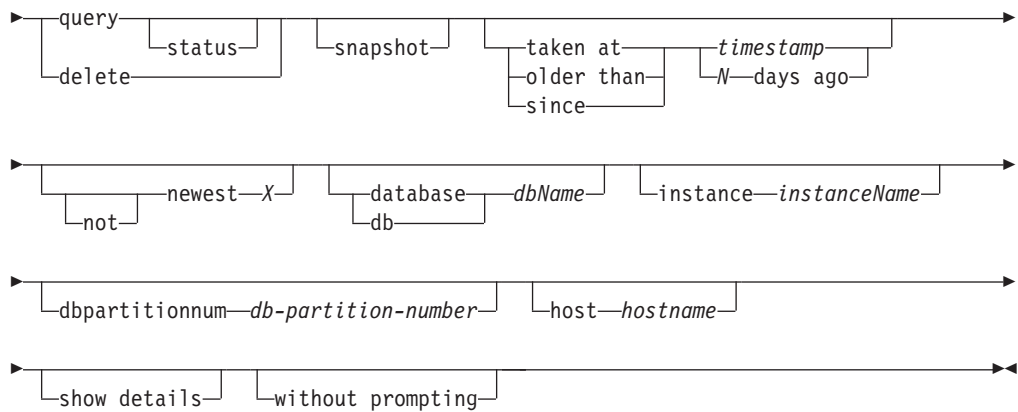
なし

コマンド構文

```

▶▶ db2acsutil [load—libraryName] [options—"optionsString"]

```



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用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。この文字列は、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

YYYYMMDDhhmmss の形式のタイム・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ザが生成されます。

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方法です。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 35. 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。
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_RESTOREABLE	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_RESTOREABLE	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_RESTOREABLE	ボリュームに直接アクセスができますが、RESTORE DB 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。

表 35. 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
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。

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。

db2ca - 構成アシスタントの開始

構成アシスタントを開始します。構成アシスタントは、データベース・マネージャ構成、DB2 レジストリー、ノード・ディレクトリー、データベース・ディレクトリー、および DCS ディレクトリーなどの DB2 データベース構成を管理するために使用されるグラフィカル・インターフェースです。

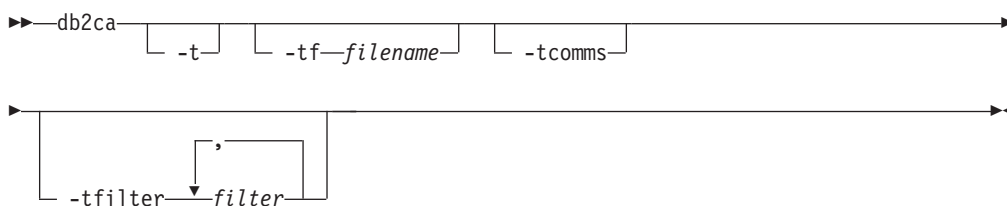
許可

sysadm

必要な接続

なし

コマンド構文



コマンド・パラメーター

- t** GUI トレースをオンにし、出力をコンソール・ウィンドウへ送信します。Windows オペレーティング・システムでは、db2ca コマンドにコンソール・ウィンドウはありません。したがって、このオプションは、Windows オペレーティング・システム上では無効です。
- tf filename** GUI トレースをオンにし、トレースの出力を指定されたファイルに保管します。出力ファイルは、Windows オペレーティング・システムでは <DB2 install path>%sqllib%tools に、Linux および UNIX のシステムでは /home/<userid>/sqllib/tools に保管されます。
- tcomms** トレースを通信イベントのみを対象とするように制限します。
- tfilter filter** トレースを、指定したフィルター (1 つ以上) を含む項目のみを対象とするように制限します。

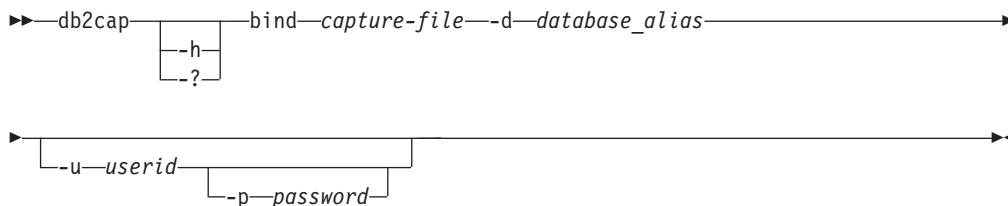
db2cap - CLI/ODBC 静的パッケージ・バイnding・ツール

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・ファイルは、CLI/ODBC/JDBC/.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、アプリケーションの実行中にキャプチャーされた SQL ステートメントを含みます。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、アプリケーションが静的 SQL を実行するために CLI/ODBC/JDBC/.NET ドライバーによって使用されます。

許可

- SQL ステートメントにより参照される、データベース・オブジェクトへのアクセス権はキャプチャー・ファイルの中に記録されています。
- 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 では 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 アプリケーションの静的パッケージ・バインディングは、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。

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

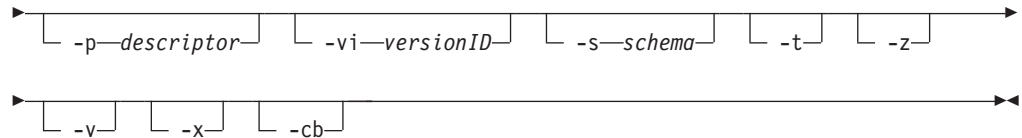
なし

必要な接続

なし

コマンド構文

```
db2cat [-d dbname] [-h] [-l] [-n name] [-o outfile]
```



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。

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。

summary

サマリー表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 のパ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。

-z キー・ストローク・プロンプトを無効にします。

-v パック記述子を妥当性検査します。このパラメーターは、表のパック記述子のみで有効です。

-x カタログ内の表スペース・エクステント・サイズ (表名は必要ない) を妥当性検査します。

-cb SYSCAT.BUFFERPOOLNODES (表名は必要ない) から孤立行を除去します。

使用上の注意

- 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ます (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。

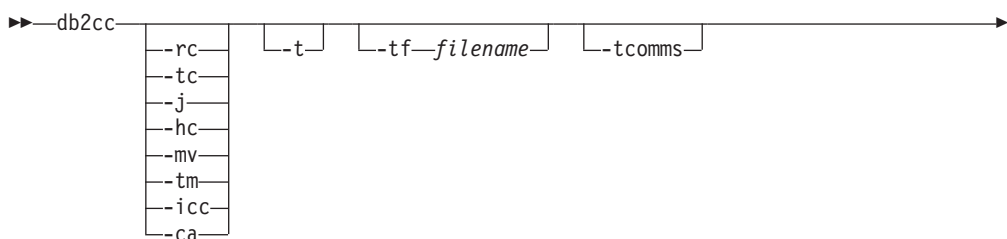
db2cc - コントロール・センターの開始

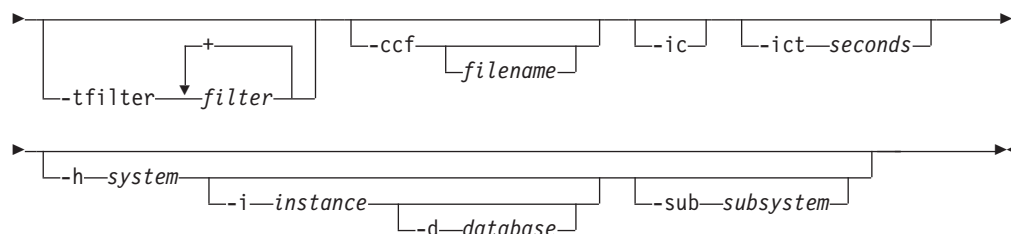
コントロール・センターを開始します。コントロール・センターは、データベース・オブジェクト (データベース、表、およびパッケージなど) とそれらの相互リレーションシップを管理するために使用する、グラフィカル・インターフェースです。

許可

sysadm

コマンド構文





コマンド・パラメーター

- rc** レプリケーション・センターをオープンします。
- hc** ヘルス・センターをオープンします。
- tc** タスク・センターをオープンします。
- j** ジャーナルをオープンします。
- mv** メモリー・ビジュアライザーをオープンします。
- tm** 「未確定トランザクション・マネージャーの識別 (Identify Indoubt Transaction Manager)」をオープンします。
- icc** インフォメーション・カタログ・マネージャーをオープンします。
- ca** 構成アシスタントを開きます。
- t** 初期化コードに対してコントロール・センターのトレースをオンにします。
Windows オペレーティング・システムでは、db2cc コマンドにコンソール・ウィンドウはありません。したがって、このオプションは、Windows オペレーティング・システム上では無効です。
- tf** 初期化コードに対してコントロール・センターのトレースをオンにし、トレースの出力を指定されたファイルに保管します。出力ファイルは、Windows では <DB2 install path>%sqllib%\tools に、UNIX オペレーティング・システムでは /home/<userid>/sqllib/tools に保管されます。
- tcomms**
トレースを通信イベントのみを対象とするように制限します。
- tfilter filter**
トレースを、指定したフィルター (1 つ以上) を含む項目のみを対象とするように制限します。
- ccf filename**
コマンド・エディターをオープンします。ファイル名が指定された場合は、このファイルの内容がコマンド・エディターの「スクリプト (Script)」ページにロードされます。ファイル名を指定するときは、ファイルの絶対パスを指定する必要があります。
- ic** インフォメーション・センターをオープンします。
- ict seconds**
アイドル接続タイマー。指定した秒数が経過すると、コントロール・センターによって保守されているプールにあるアイドル接続はクローズされます。デフォルト・タイマーは 30 分です。

- h** *system*
システムに関連してコントロール・センターをオープンします。
- i** *instance*
インスタンスに関連してコントロール・センターをオープンします。
- d** *database*
データベースに関連してコントロール・センターをオープンします。
- sub** *subsystem*
サブシステムに関連してコントロール・センターをオープンします。

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・マネージャー構成設定
- 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、特に DB2 構成アシスタントがインストールされていないワークステーションで、接続構成情報をエクスポートする場合や、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

以下のいずれか。

- *sysadm*
- *sysctrl*

コマンド構文



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・マネージャー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・マネージャー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` ディレクトリーを見つけます。
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` ディレクトリーを見つけます。
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ーです。

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ーを使用すると、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。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、DB2 構成アシスタント (CA) がインストールされていないワークステーションで特に役立ちます (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 に必要です。

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、構成アシスタント、コントロール・センター、または `db2cfexp` の使用により 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。

db2chglbpath - 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用ライブラリー・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。

`db2chglbpath` 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場所です。

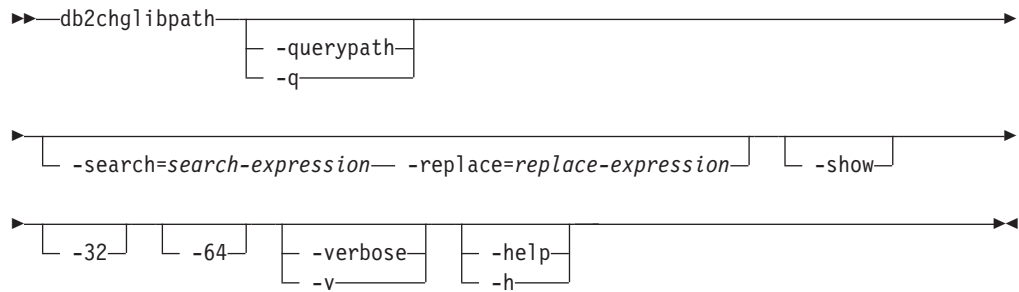
前提条件

- 共用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、共用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=*search-expression*

検索対象の式を指定します。

-replace=*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。

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マイグレーション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 外部ルーチン共用ライブラリー・ファイルを更新する目的でのみ使用するものです。アプリケーションおよびルーチンのマイグレーションに関するトピックの関連リンクを参照してください。
- 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、共用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 のシステムで、関連した DB2 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 インストーラーが使用します。DB2 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 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 をインストールした後に SELinux (Security-enhanced Linux) が使用可能な場合、DB2 を正常に作動させるには現行リリースの DB2 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、後述の『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

ルート・インストールには root 権限が必要です。非ルート・インストールの場合、非ルート・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 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 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 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 ディレクトリ下の 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 システム上にユーザーが DB2 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 rpm 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 共用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ーを使用すると、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。

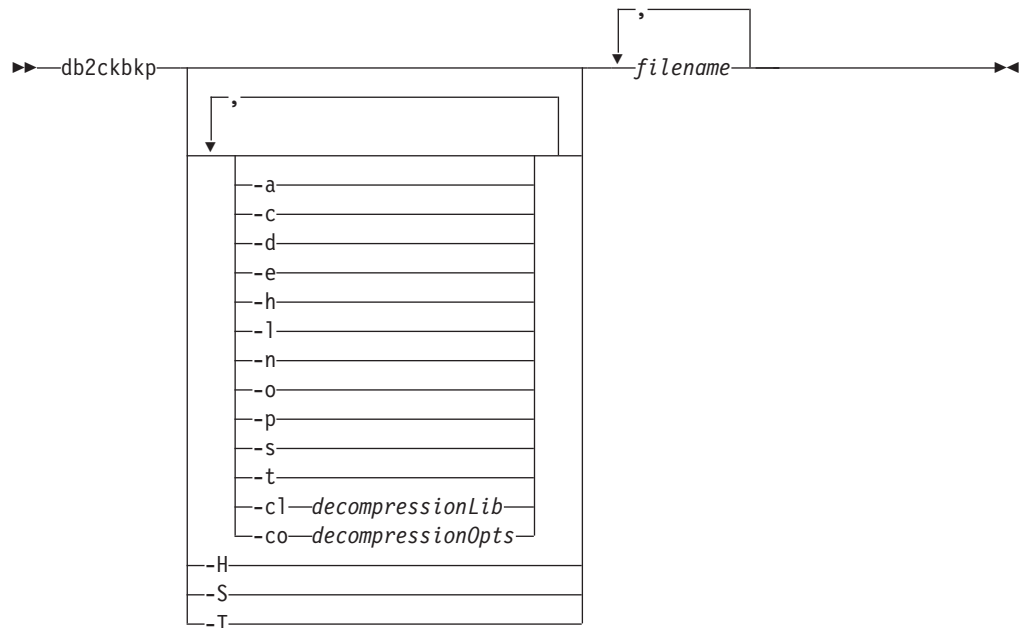
許可

このユーティリティー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ーを実行するには、それらの読み取り許可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。

-cl *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 '@' なら、データの残りの部分は、DB2 によって、サーバー上に存在するファイルの名前として解釈されます。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。

-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 <tbodyID> <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 <tbodyID> <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、

```
0 <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。
- H** `-h` と同じ情報を表示しますが、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。
- 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。
- S** `-s` と同じ情報を表示しますが、イメージの検証は実行しません。このオプションは他のオプションと併用できません。
- t** 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。
- T** `-t` と同じ情報を表示しますが、イメージの検証は実行しません。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 プラットフォーム)

```
db2ckbkp SAMPLE.0.krodger.NODE0000.CATN0000.19990817150714.001
SAMPLE.0.krodger.NODE0000.CATN0000.19990817150714.002
SAMPLE.0.krodger.NODE0000.CATN0000.19990817150714.003
```

```
[1] Buffers processed: ##
[2] Buffers processed: ##
[3] Buffers processed: ##
Image Verification Complete - successful.
```

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/SQLLOGDIR
Backup Buffer Size        -- 4194304
Number of Sessions       -- 1
Platform                  -- 0
```

The proper image file name would be:

SAMPLE2.0.krodger.NODE0000.CATN0000.19990818122909.001

```
[1] Buffers processed: ####  
Image Verification Complete - successful.
```

使用上の注意

1. 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。
2. 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、UNIX ベースのシステムでは以下のようにします。

```
db2ckbkp -h /dev/rmt0
```

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。

db2ckmig - データベース事前マイグレーション・ツール・コマンド

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

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。パーティション・データベース環境では、各データベース・パーティションに対してコマンドを実行します。

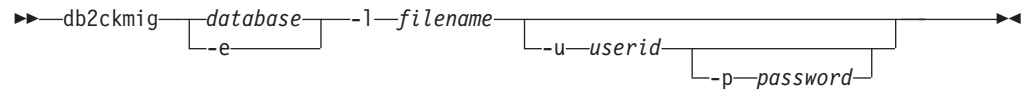
許可

sysadm

必要な接続

なし

コマンド構文



コマンド・パラメーター

database

スキャンするデータベース名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。必須パラメーター。

-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

db2imigr コマンドでインスタンスをマイグレーションするとき、**db2ckmig** がマイグレーションの一部として暗黙的に呼び出されます。

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**、**DATALINK**、**DECFLOAT**、**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、データベース・マイグレーション失敗の原因となる、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。
- データベースが **HADR** スタンバイの役割ではない。
- **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。

マイグレーション・プロセスに成功するには、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。 **db2ckmig** 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た場合、db2imigr は失敗し、DBI1205E エラー・コードを戻します。データベースのマイグレーション前にこの状態を解決する必要があります。

db2ckmig を手動で実行することにした場合は、DB2 インスタンスのインストール後、インスタンスをマイグレーションする前に、各データベースに対してそのツールを実行しなければなりません。Linux および UNIX システムにおいてこのユーティリティーは、DB2DIR/bin 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 プラットフォームでは、インストール時にマイグレーション・オプションを選択した場合、インスタンスのマイグレーションが行われ、db2ckmig を実行するためのプロンプトが表示されます。システムにローカル・データベースが存在する場合、CD の db2\Windows\Utilities にある db2ckmig を実行する必要があることを通知するメッセージ・ボックスが表示されます。メッセージ・ボックスが表示されたら、選択肢として、メッセージを無視するか、インストール・プロセスを中止することができます。db2ckmig を実行し、エラーがなければインストールを続行してください。エラーが発生したら、インストールを中止し、エラーを修正した後で再びインストールしてください。一方、新規インストール・オプションを選択した場合には、インスタンスをマイグレーションするために db2imigr を実行する必要があります。その際、db2ckmig も実行されます。

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は、db2ckmig は実行されません。

データベースの状態の検査は、以下の手順で行います。

1. インスタンス所有者としてログオンする。
2. db2ckmig コマンドを発行する。
3. ログ・ファイルをチェックする。db2imigr コマンドによって db2ckmig コマンドが実行される場合に指定されるログ・ファイルは、Linux および UNIX 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、また 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ある migration.log ファイルです。ログ・ファイルは、db2ckmig コマンドの実行時に起きたエラーを表示します。ログが何も記録されていないことをチェックしてから、マイグレーション・プロセスを継続してください。不明な sqlcode または理由コードをツールが報告した場合、詳細については、移行後のリリースの新しい資料を参照してください。

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。

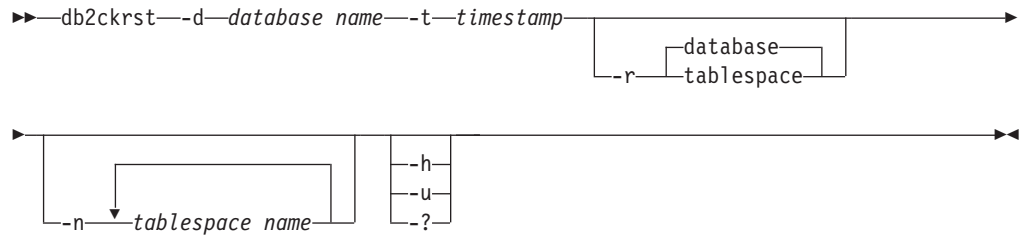
許可

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 tbsp1 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もあります。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。

db2cli - DB2 対話機能 CLI

CLI で、設計およびプロトタイピングのための対話式コール・レベル・インターフェース環境を開始します。これはデータベース・インスタンス所有者のホーム・ディレクトリーの `sql1lib/samples/cli/` サブディレクトリーにあります。IBM Data Server Driver for ODBC and CLI の場合、db2cli 実行可能ファイルは `clidriver/bin` ディレクトリーにあります。

許可

なし

必要な接続

なし

コマンド構文

▶▶—db2cli—◀◀

コマンド・パラメーター

なし

使用上の注意

DB2 対話機能 CLI は、CLI 関数呼び出しの設計、プロトタイプの実装、テストに使用できる一連のコマンドからなっています。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、ツールのパフォーマンスについては IBM の保証はありません。DB2 対話機能 CLI はエンド・ユーザーを対象にしてはいないので、広範囲にわたるエラー・チェック機能はありません。

2 種類のコマンドがサポートされています。

CLI コマンド

IBM 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。

コマンドは対話式で発行することも、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。CLI コマンド・ドライバーの便利なフィーチャーは、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み、それによってコマンド・スクリプトを作成することです。このスクリプトは後で再実行することができます。

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。

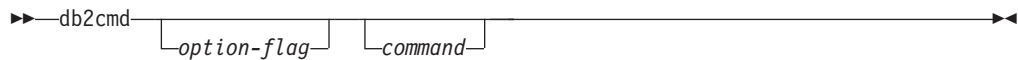
許可

なし

必要な接続

なし

コマンド構文



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・ウィンドウを共用しファイル・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コマンド

このユーティリティーは、UNIX および Linux のシステムにおいて、`/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。それらの DB2 HA スクリプトは、IBM Tivoli System Automation for Multiplatforms Base Component (SA MP Base Component) を DB2 HA フィーチャーと共に使用するのに必要です。

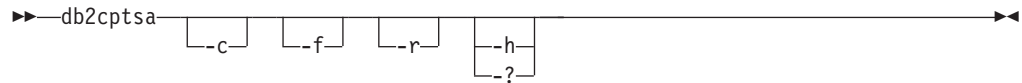
許可

このコマンドの実行には root 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

- 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 Base Component 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 Base Component 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 Base Component 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)
- 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。

許可

sysadm

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

```
db2dart database-name [action] [options]
```

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 つの表を検査します。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。
- /TSF** 表スペース・ファイルとコンテナのみを検査します。
- /TSC** 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。必須入力値は、表スペース ID です。
- /TS** 1 つの表スペースとその表を検査します。必須入力値は、表スペース ID 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。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。
 - パーティション表の場合、/DI アクションは、/OI 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 `tblspaceid` 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。

/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・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 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、およびページ数の 4 つです。

/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 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。

/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。

/RHWM

空の SMP エクステンツを使用して、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。

/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。

/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どうか識別します。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。ページ番号には、プール相対用に p という接尾部を付けなければなりません。/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 option

クイック・オプション。/DB、/T、および /TS アクションのみに適用されます。DAT オブジェクトのページ 0 のみと、索引オブジェクトの一部を検査します (BMP、LOB、LF オブジェクトは検査せず、DAT または INX オブジェクトの全体をトラバースしません)。

/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。

使用上の注意

1. 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。
2. 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。
3. /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。
4. /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。

5. 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。
6. 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 (/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。
7. 通常、db2dart 要求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 アクションおよび /LHWM アクションでは、オフライン・データベースが絶対に必要というわけではありません。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。

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。

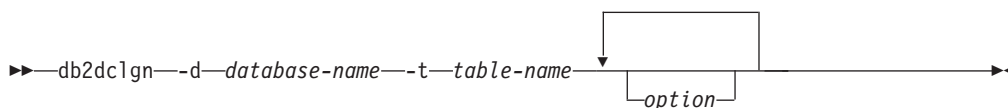
許可

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、そ

れぞれ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)
- .cbl (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

sqldbcchar または **wchar_t** が、C で **GRAPHIC/VARGRAPHIC/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、OS390 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

db2dc1gn -d sample -t emp_resume -l cobol -a replace

db2diag - db2diag.log 分析ツール

このユーティリティは、db2diag.log ファイル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。

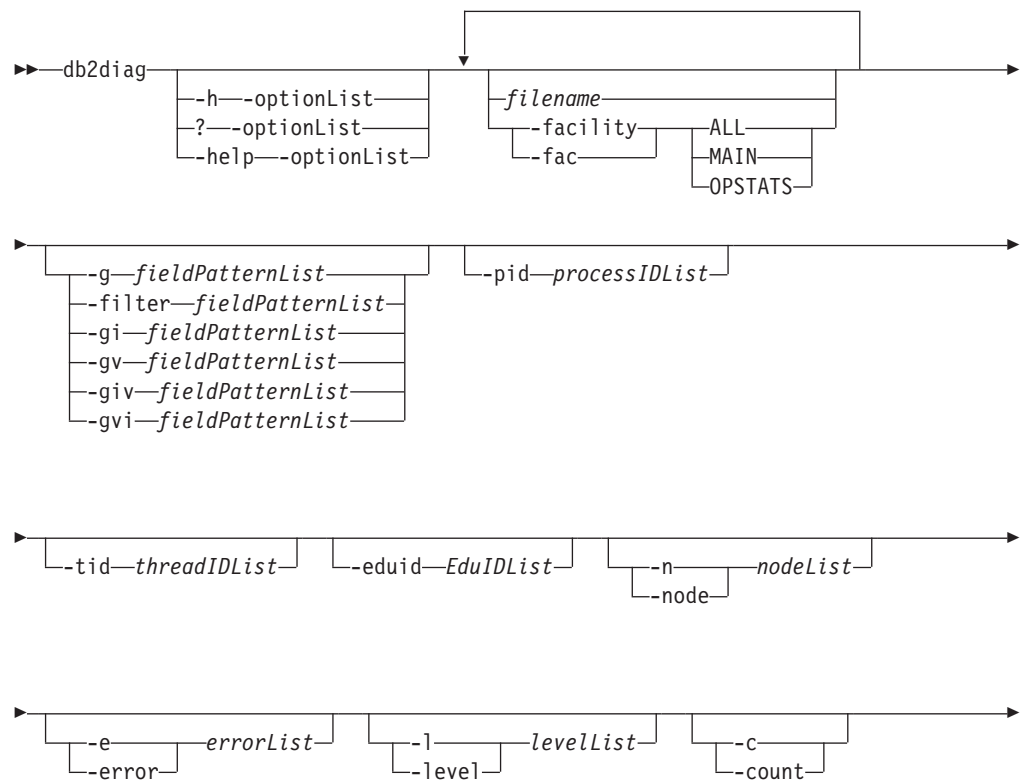
許可

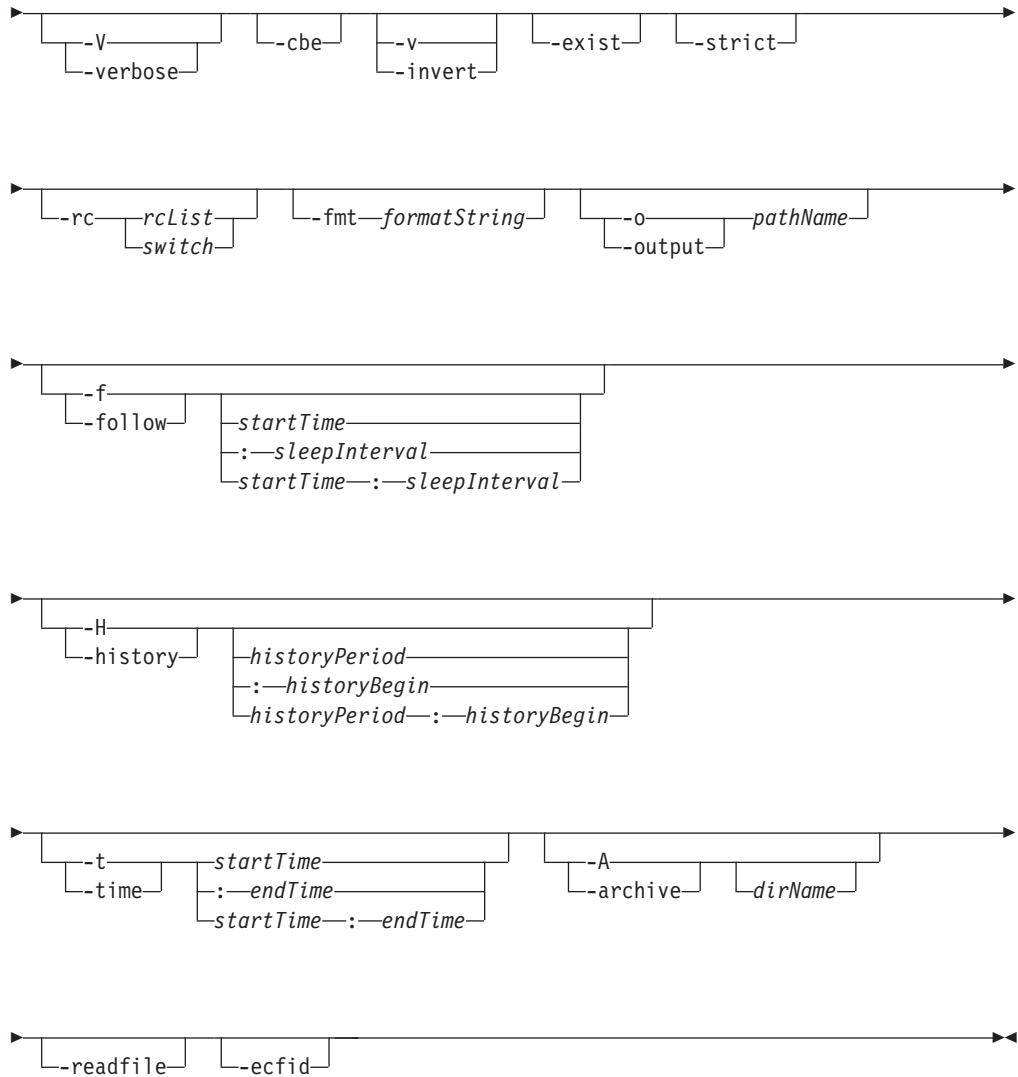
なし

必要な接続

なし

コマンド構文





コマンド・パラメーター

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.log` ファイルが処理されます。そのファイルが見つからない場合、`DIAGPATH` 変数によって設定されている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、すべてのオプティマイザー統計レコードは `OPTSTATS` ファシリティ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・オプションは次のとおりです。

ALL すべてのファシリティからレコードを戻します。

MAIN DB2 の汎用診断ログ (例えば db2diag.log、管理者通知ログ、ローテーション・イベント・ログなど) からレコードを戻します。

OPSTATS

オブティマイザー統計に関連したレコードを戻します。

-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 (複数の場合はコンマで区切る) 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。

例 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。

-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、長い名前のフィールドには別名を使用できます。フィールド名の前の接頭部 % 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。

rval 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。%rval %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 ("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、つまり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、つまり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す。 *fieldName operator searchPattern* の形式です。

演算子は以下の 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。
- 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管します。

-f | -follow

入力ファイルが正規の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。

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)、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:historyBegin

ログに記録されたメッセージが、`historyBegin` で指定された時刻から開始して、`historyPeriod` で指定された期間中に表示されることを指定します。

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。

-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*

診断ログ・ファイル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.log 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** 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.log 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.log 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** 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.log ファイルが検索されます。ファイルが検出されると、**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。

-readfile

端末入力を無視して、診断ログ・ファイルから読み取れ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。

-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 *ecfId*

数値 *ecfId* 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。

例

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、次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、ノード 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 or 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' or 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 ファイルへの項目の追加を表示するには、以下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す。

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、db2diag に、DIAGPATH 環境変数で指定されたディレクトリーからの db2diag.log を処理するように強制します。

端末入力を無視して、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
```

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.log` 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` レジストリー変数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.log` 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。
- `-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.log` 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ときには、`-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.log` ファイルのログに記録されています。

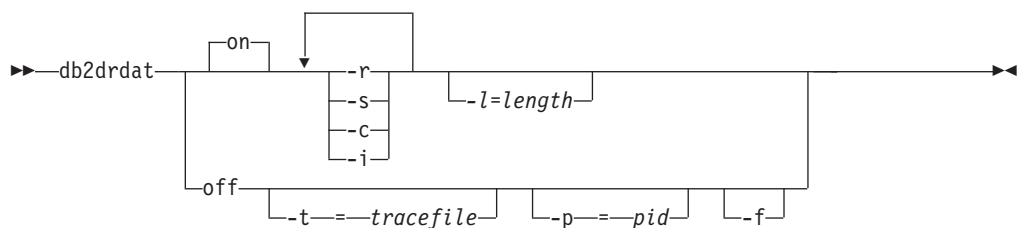
db2drdat - DRDA トレース

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、問題判別に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、ファイルの許可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 データベースのドライブ・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 プラットフォー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 | drop | query | reconcile} dbpartition_number from_drive to_drive ▶▶
```

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
```

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、コンテナー、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. DB2MSCS.CFG ファイルに TARGET_DRVMAP_DISK キーワードを使用すると、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使用可能化

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 エクステントに割り振られます。

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 db2envvar.bat が実行されるのかを示す必要があります。例えば、e:%sqllib 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:%sqllib%bin%db2envvar.bat を発行します。

db2eva - イベント・アナライザー

イベント・アナライザーを開始することにより、DB2 イベント・モニターが生成し表に送ったパフォーマンス・データをトレースできるようにします。

許可

イベント・アナライザーは、データベースと一緒に保管されているイベント・モニター表からデータを読み取ります。したがって、このデータにアクセスするために以下の権限が必要です。

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

必要な接続

データベース接続

コマンド構文

```
▶▶ db2eva [-db database-alias] [-evm evmon-name] ▶▶
```

コマンド・パラメーター

db2eva では、パラメーターの指定は必須ではありません。パラメーターが指定されない場合は、「イベント・アナライザーのオープン (Open Event Analyzer)」ダイアログ・ボックスで、データベースとイベント・モニターの名前を要求するプロンプトが出されます。

-db database-alias

イベント・モニター用に定義したデータベースの名前を指定します。

-evm evmon-name

トレースが分析されるイベント・モニターの名前です。

使用上の注意

必要なアクセスが行われないと、ユーザーは一切のイベント・モニター・データを取り出せません。

イベント・モニター・トレースの取り出しには、2通りの方法があります。

1. コマンド行から `db2eva` と入力すると、「イベント・アナライザーのオープン (Open Event Analyzer)」ダイアログ・ボックスをオープンできます。このダイアログ・ボックスで、ドロップダウン・リストからデータベースとイベント・モニターの名前を選択して「OK」をクリックすると、「イベント・アナライザー (Event Analyzer)」ダイアログ・ボックスがオープンします。
2. コマンド行から `-db` パラメーターと `-evm` パラメーターを指定すると、指定したデータベースで「イベント・アナライザー (Event Analyzer)」ダイアログがオープンします。

イベント・アナライザーはデータベースに接続し、`SYSIBM.SYSEVENTTABLES` から対象を選択する命令を発行して、イベント・モニター表を取得します。こうして必要なデータが取り出されると、接続は解放されます。

イベント・アナライザーは、アクティブなイベント・モニターが生成したデータを分析するのに使用できます。ただし、イベント・アナライザーが起動された後にキャプチャーされたイベント・モニターは、表示されない場合があります。データが適正に表示されるようにするには、イベント・モニターをオフにしてからイベント・アナライザーを起動してください。

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`)、以下のいずれかが必要です。

- `sysadm`
- `sysctrl`
- `sysmaint`
- `dbadm`

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/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/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、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。

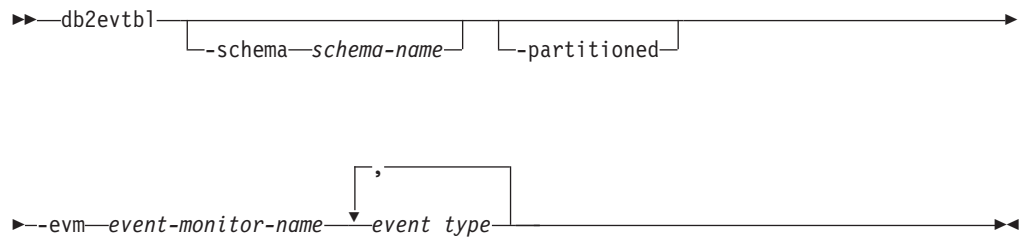
許可

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エレメ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 ステートメントで使用できるイベント・タイプのいずれか。例えば DATABASE、TABLES、TRANSACTIONS などです。

例

```
db2evtb1 -schema smith -evm foo 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活動化するために、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・モニターを活動化します。

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、1 つのイベントにつき複数のレコードを作成してフラッシュ可能なので、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_flush` をどのターゲット表に入れる必要もありません。

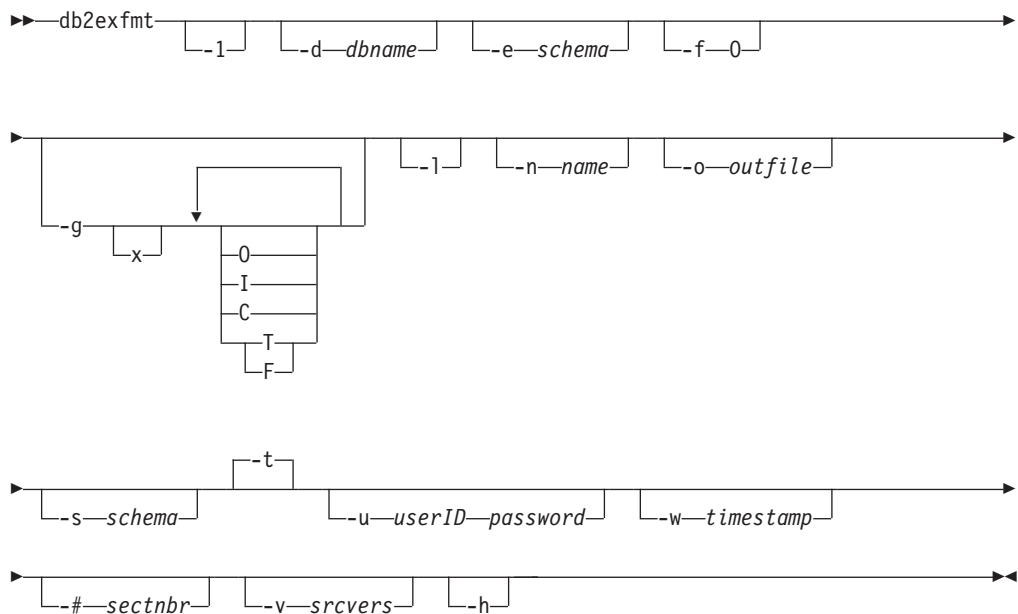
db2exfmt - Explain 表フォーマット

EXPLAIN 表の内容をフォーマットするには、db2exfmt ツールを使用します。このツールは、インスタンスの `sqlib` 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、ユーザーが Explain スキーマを提供するよう求められます。

-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。

-f

フォーマット・フラグ。このリリースでは、値 `O` (演算子サマリー) だけがサポートされています。

-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、ヘルプ情報だけが表示されます。

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 `-l` 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は、Explain 表 SQL スキーマの指定を促すプロンプトが表示されます。

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促すプロンプトが表示されます (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表示します。使用可能でない場合、`db2exfmt` は、`EXPLAIN_OBJECT` 表に保管された統計と、システム・カタログから直接取得されたいくつかの統計を表示します。

以下は EXPLAIN スナップショットの例です。

```
db2 explain plan with snapshot for query
db2exfmt
```

または、

```
db2 set current explain mode yes
db2 set current explain snapshot yes
run the 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。 `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。 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。

- *sysadm* 権限
- *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

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。

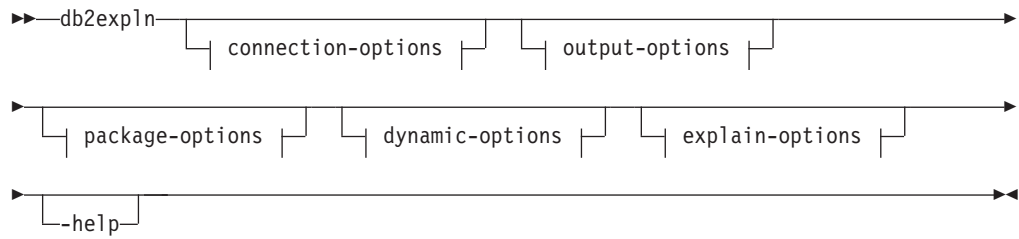
db2expln - SQL および XQuery Explain

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できます。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、照会キャッシュ内のセクションを調べます。

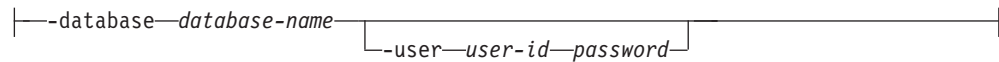
許可

sysadm または *dbadm*

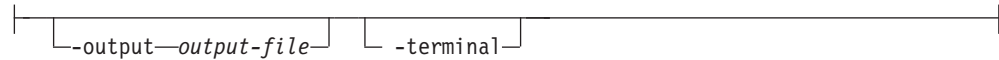
コマンド構文



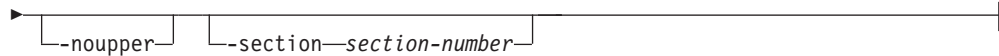
connection-options:



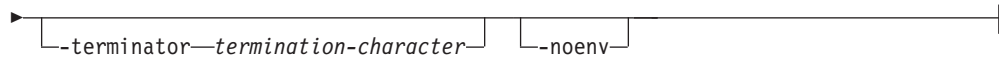
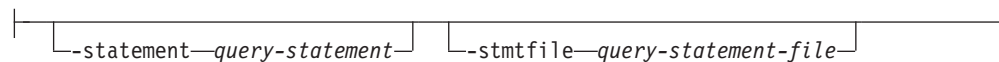
output-options:



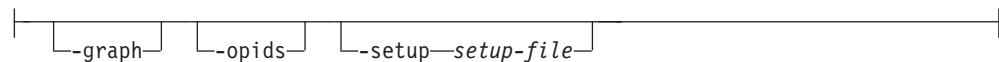
package-options:



dynamic-options:



explain-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に使用する許可 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:

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 사용됩니다。

後方互換性のために、-noupper の代わりに -l (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。

-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。

`dynexpln` との互換性のために、`-statement` の代わりに `-q` を使用できます。

-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。

`dynexpln` との互換性のために、`-stmtfile` の代わりに `-f` を使用できます。

-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。

`dynexpln` との互換性のために、`-terminator` の代わりに `-z` を使用できます。

-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、元のオプティマイザー・プラン・グラフが Visual Explain の場合と同じように構成されます。

生成されるグラフは、Visual Explain のグラフとまったく一致するとは限りません。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が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 (宣言済みグローバル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。

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。

上記のオプション・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ことがあります。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。

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
```

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, SYSPFUN, 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
```

db2extsec - DB2 オブジェクトの許可の設定

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 オブジェクト (ファイル、ディレクトリー、ネットワーク共用、レジストリー・キー、サービスなど) の許可を設定します。前のリリースでは、このコマンドの名前は `db2secv82` でした。コマンド名 `db2secv82` は推奨されませんが、`db2extsec` の代替名として使用することはできます。

DB2 バージョン 9 フィックスパック 2 以降、拡張セキュリティー用にドメイン・グループを使用できるようになりました。

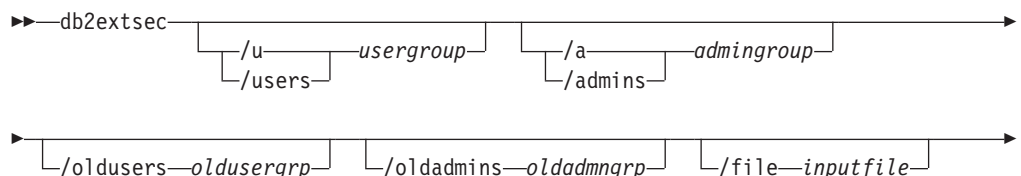
許可

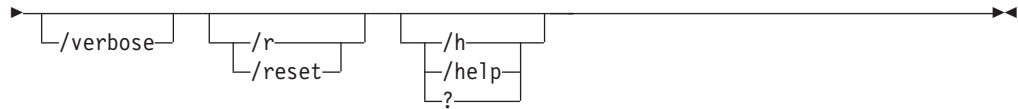
sysadm

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** *admingroup*

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。*admingroup* 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、*admingroup* 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・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、許可を認識済みの 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ので、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許可を変更できます。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*

許可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` オプションを参照)。

```
db2extsec /reset
```

上記の例と同じく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  
  
* 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

指定されたログ・シーケンス番号 (LSN) で識別されるログ・レコードを含むファイルの名前を戻します。

許可

なし

コマンド構文

```
db2flsn -q --db dbname --file LFH-file --path LFH-dir input_LSN
```

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・メッセージや警告メッセージは印刷されず、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
- -104 データベースがリカバリー可能でない
- -105 LSN が大きすぎる
- -106 無効なデータベース
- -500 論理エラー

他の有効な戻りコードは以下のとおりです。

- 0 正常な実行
- 99 警告: 結果は、分かっている最後のログ・ファイル・サイズに基づいている

-db *dbname*

調査するデータベースの名前を指定します。

-file *LFH-file*

ファイル名を含む、LFH ファイルの絶対パスを指定します。

-path *LFH-dir*

LFH ファイル SQLOGCTL.LFH.1 とそのミラー・コピー SQLOGCTL.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

先行ゼロ付きの内部 (6 または 8 バイト) 16 進数値を表す 12 または 16 文字のストリング。

例

```
db2flsn 000000BF0030
Given LSN is contained in log page 2 in log file S0000002.LOG
```

```
db2flsn -q 000000BF0030
S0000002.LOG
```

```
db2flsn 000000BE0030
Given LSN is contained in log page 2 in log file S0000001.LOG
```

```
db2flsn -q 000000BE0030
S0000001.LOG
```

```
db2flsn -db flsntest 0000000000FA0000
Warning: the result is based on the last known log file size (6
4K pages starting from log extent 10). The input_LSN might be before
the database becomes recoverable.
```

```
Given LSN is contained in log page 2 in log file S0000002.LOG
```

```
db2flsn -q -db flsntest 0000000000FA0000
S0000002.LOG
```

```
db2flsn -file C:¥DB2¥NODE0000¥SQL00001¥SQLOGCTL.LFH.1 0000000000FA4368
Given LSN is contained in log page 6 in log file S0000002.LOG
```

```
db2flsn -path C:¥DB2¥NODE0000¥SQL00001 0000000000FA4368
Given LSN is contained in log page 6 in log file S0000002.LOG
```

使用上の注意

- -db、-file、および -path が指定されない場合、このツールは、LFH ファイル SQLOGCTL.LFH.1 とミラー・コピー SQLOGCTL.LFH.2 が現行ディレクトリーに存在すると想定します。
- -file が指定されている場合、提供された LFH ファイルだけが使用されます。
- -file が指定されない場合、2 つの LFH ファイル (SQLOGCTL.LFH.1 とミラー・コピー SQLOGCTL.LFH.2) から得られる最新のログ制御レコードが使用されます。この 2 つのファイルのいずれかが欠落または破損している場合、もう 1 つのファイルが使用されます。両方のファイルが欠落または破損している場合、db2flsn は失敗します。
- このツールは、**logfilsiz** データベース構成パラメーターを使用します。DB2 は、このパラメーターの最新の 3 つの値と、各 **logfilsiz** 値によって作成された最初のログ・ファイルを記録します。このため、**logfilsiz** が変更されても、ツールは正しく動作することができます。指定された LSN の日付が、記録されている最も古い日付の **logfilsiz** の値よりも前の場合、ツールはこの値を使用し、警告を戻します。このツールは、DB2 Universal Database バージョン 5.2 より前のデータベース・マネージャーでも使用できます。その場合、(**logfilsiz** の値が変更されない場合に得られる) 正しい結果に対しても、警告が戻されます。
- このツールは、リカバリー可能データベースでのみ使用することができます。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 オペレーティング・システムでのみ使用できます。

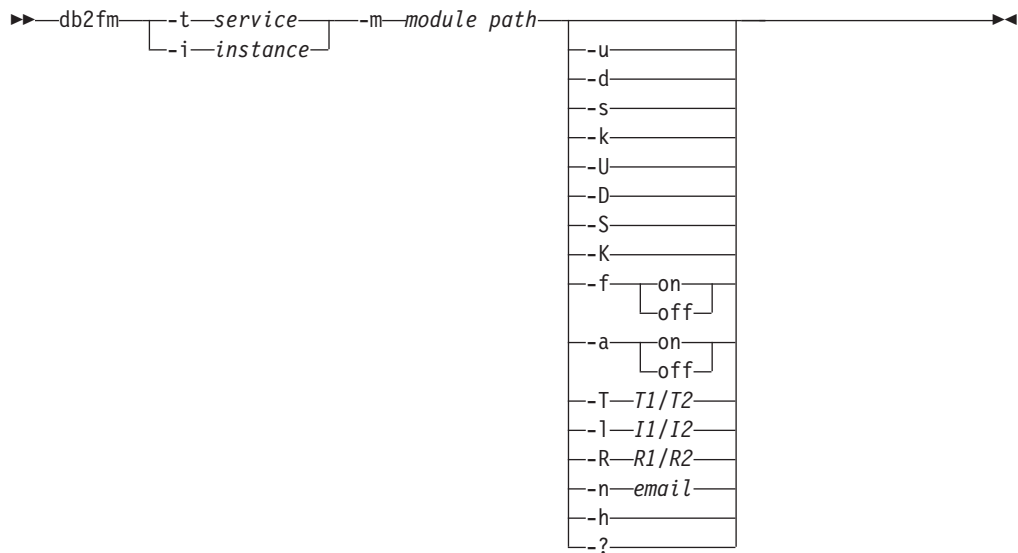
許可

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)
- 不明 (UNKNOWN)

-f on | off

障害モニターを ON または OFF にします。このオプションがオフに設定される場合、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

障害モニターを活動化または非活動化します。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それぞれ更新します。
- -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、タイムアウトをそれぞれ設定します。

-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 email

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。

db2fmcu - DB2 障害モニター・コントローラー・コマンド

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 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 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

必要な接続

なし

コマンド構文

```

▶▶ db2fmcu --u--p--db2fmcu_path-- --f--inittab
           --d
  
```

コマンド・パラメーター

-u -p *db2fmc_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。 *db2fmc_path* は、FMC デーモン (db2fmc) オブジェクトへの完全なパスで、例えば `/opt/IBM/db2/bin/db2fmc` となります。

-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。

-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/db2fmc
```

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` によって既に行われているので必要ありません。

db2fodc - DB2 第 1 出現データ収集コマンド

この `db2fodc` ユーティリティーは、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

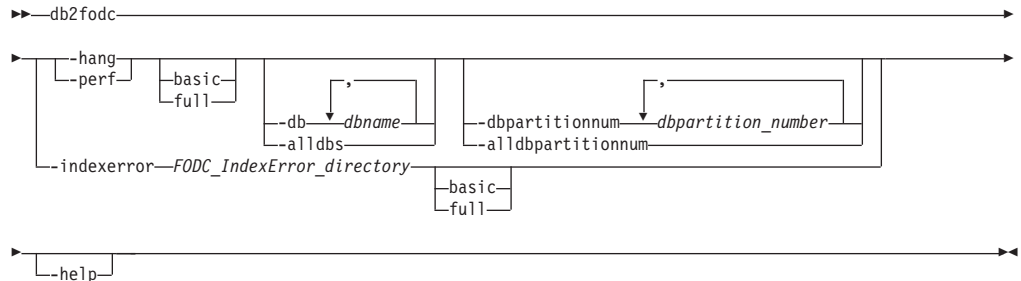
このツールは、他の PD ツールの機能を取り込んで、それらのツールに取って代わっています。`db2fodc` コマンドは、ハングやパフォーマンス上の重大な問題など第 1 出現データ収集 (FODC) を自動的にトリガーできない問題に関する手動による FODC に使用できます。それは索引エラーに関するデータを収集するためにも使用できます。`db2fodc` ツールはデータをキャプチャーして FODC パッケージに組み込み、現在の *DIAGPATH* に作成された *FODC_symptom* ディレクトリー内にそのパッケージを入れます。ここで、*symptom* は問題の症状を表します。

許可

以下のいずれか。

- 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



コマンド・パラメーター

-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。

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。

-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。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。

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。このオプションには、db2trc コマンドおよび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を必要とします。

-hang および -perf のサブオプション

-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。以下に例を示します。

```
db2fodc -hang -db sample,dbsample
```

-alldbpartitionnum

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。

-dbpartitionnum *dbpartition_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。

-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* が必要であり、db2cos_indexerror_short(.bat) スクリプト、db2cos_indexerror_long(.bat) スクリプト、またはその両方を含む必要があります。

basic

基本収集モードが実行されます。同じ *FODC_IndexError_directory* 内に db2dart レポートが存在しないようにしてください。

full

完全収集モードが実行されます。同じ *FODC_IndexError_directory* 内に db2dart レポートが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 コマンドが含まれる場合、完全モードでは、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。

-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。

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 -alldbs
```

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。新しいディレクトリー FODC_hang_<timestamp> が、現在の診断パスの下に作成されます (既に存在する場合にはエラーが生成されます)。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、FODC_hang_<timestamp> 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。新規ディレクトリー FODC_hang_<timestamp> が現在の診断パスの下に自動的に作成されます。

タイム・スタンプは `db2fodc` が呼び出された時間になります。`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、`FODC_hang_<timestamp>` 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。新しいディレクトリー `FODC_perf_<timestamp>` が、現在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、`FODC_perf_<timestamp>` 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。

使用上の注意

`db2fodc -hang` および `db2fodc -perf` は、複数の物理ノードがあるマルチ・パーティション環境で実行できます。この環境では、すべての DB2 DPF ノードでの潜在的なハングまたは重大なパフォーマンス問題の際に、単一の呼び出しで

```
rah ";db2fodc -hang <full | basic> -alldbpartitionum other_options"
```

または

```
rah ";db2fodc -perf <full | basic> -alldbpartitionum other_options"
```

を使用して `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。オプション `-alldbpartitionum` と `-dbpartitionnum` は、論理パーティション番号 (別の物理マシンではない) に対してのみ作用します。デフォルトでは、現在のパーティション番号の情報だけが収集されます。

`db2fodc -hang` および `db2fodc -perf` は、`FODC_symptom` ディレクトリー内に置かれたログ・ファイル `db2fodc_symptom.log` を使用します。ここで、`symptom` は `hang` または `perf` のどちらかです。また `db2fodc` は、FODC サブディレクトリー内の FODC パッケージについて説明した状況情報およびメタデータをこのファイルに保管します。このファイルには、FODC のタイプに関する情報、データ収集の開始または終了 (あるいはその両方) のタイム・スタンプ、および FODC パッケージの分析に役立つその他の情報が入れられます。

`db2fodc -hang` and `db2fodc -perf` によるデータ収集の動作は、パラメーターによって制御でき、実行するスクリプトを変更してカスタマイズできます。UNIX システムでデータ収集をカスタマイズする場合、`/bin/db2cos_symptom` に置かれているスクリプトを `/adm/db2cos_symptom` にコピーします。ここで、`symptom` は `hang` または `perf` のどちらかです。この新規ディレクトリーに置いてから、スクリプトを自由に

変更します。Windows システムでは、デフォルト・スクリプト `¥bin¥db2cos_symptom.bat` を変更するだけです。UNIX システムの場合、`db2fodc` はまず `/adm/db2cos_symptom` にあるスクリプトを実行しようとし、それが見つからないと `/adm/db2cos_symptom` 内の元のスクリプトを実行します。Windows システムでは、必ずスクリプト `¥bin¥db2cos_symptom.bat` を実行します。

収集されたデータは、デフォルトで `FODC_symptom_timestamp` という名前の診断パスの新規または既存のサブディレクトリーに保管されているいくつかのファイルに書き込まれます。`symptom` は問題の症状で、`timestamp` は自動または手動による FODC 呼び出しの時間です。`db2diag.log` 診断メッセージはログに記録され、この特定の自動または手動による FODC に使用したディレクトリー名を知らせてくれます。

`db2fodc -indexerror` は、複数の物理または論理ノードがあるマルチ・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 DPF パーティション番号での索引エラーに関する情報を収集するために、

```
db2_all "<<+node#< 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。`node#` は、特定の DPF ノード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_Node#>` 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` 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.log` 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ため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。

FODC パッケージの作成

db2fodc コマンドの実行が完了すると、db2support ツールを実行して結果の診断ファイルを収集し、IBM サポートに送信するための FODC パッケージを作成する必要があります。

デフォルトでは、db2support は、診断データのディレクトリー・パスの下にあるすべての FODC_XXX_XXX ディレクトリーを収集します。診断情報について IBM サポートからさらに情報を要求されないようにするためです。

db2fodc 診断データの収集

db2fodc -hang は、以下の情報を収集します。

db2fodc -hang は、以下の情報を収集します。

- 基本オペレーティング・システム情報。問題は OS レベル、パッチなどに起因する可能性があります。
- 基本 DB2 構成情報。
- オペレーティング・システムは、vmstat、netstat、iostat などの情報をモニターします。
 - 保管されているタイム・スタンプを使用して少なくとも 2 度繰り返されます。
- 部分的な呼び出しスタック。CPU の上位エージェントの DB2 スタック・トレース。
- オペレーティング・システム・トレース。AIX でトレースします。
- db2pd によって収集される診断情報。
- DB2 トレース。
- 全 DB2 呼び出しスタック。
- 2 回目の DB2 構成情報。
 - 2 度目の DB2 トレース・コレクションが含まれます。
- スナップショット情報。db2 get snapshot for データベース、アプリケーション、表など。
 - 複数の論理ノードがある場合、ノードごとに情報が収集されます。

db2fodc -perf は、以下の情報を収集している可能性のあるシステムをモニターします。

- スナップショット
- スタック・トレース
- 仮想メモリー (Vmstat)
- 入出力情報 (Iostat)
- トレース
- 状況に応じた他のいくつかの情報。詳しくは、スクリプトを参照してください。

db2fodc -indexerror は、以下の情報を収集します。

- 基本モード
 - `db2cos_indexerror_short(.bat)` スクリプトが実行されます。詳しくは、スクリプトを参照してください。
 - 該当する `db2dart` コマンドがスクリプト内に存在する場合、`db2dart /DD` および / または `db2dart /DI` データ・フォーマット・アクションはページ数を 100 に限定して実行されます。
- 完全モード
 - `db2cos_indexerror_short(.bat)` および `db2cos_indexerror_long(.bat)` スクリプトが実行されます。詳しくは、スクリプトを参照してください。
 - 該当する `db2dart` コマンドがスクリプト `db2cos_indexerror_short(.bat)` 内に存在する場合、`db2dart /DD` および / または `db2dart /DI` データ・フォーマット・アクションはページ数を 100 に限定して実行されます。
 - 該当する `db2dart` コマンドがスクリプト `db2cos_indexerror_long(.bat)` 内に存在する場合、`db2dart /DD` および / または `db2dart /DI` データ・フォーマット・アクションはページ数の制限なしで実行されます。
 - 該当する `db2dart` コマンドが `db2cos_indexerror_long(.bat)` スクリプト内に存在する場合、`db2dart /T` コマンドが実行されます。このコマンドでは、データベースがオフラインであることが必要です。

db2fs - ファースト・ステップ

「ファースト・ステップ」のインターフェースを立ち上げて、ユーザーの学習と DB2 の使用を開始するのに必要な機能を開始します。

UNIX オペレーティング・システムでは、`db2fs` は `sqlllib/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 5.0 以上
- Mozilla 1.4 以上
- Firefox 1.0 以上
- Netscape 7.0 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/sqlllib/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 システムでは、このコマンドは sqlllib¥bin サブディレクトリーにあります。

許可

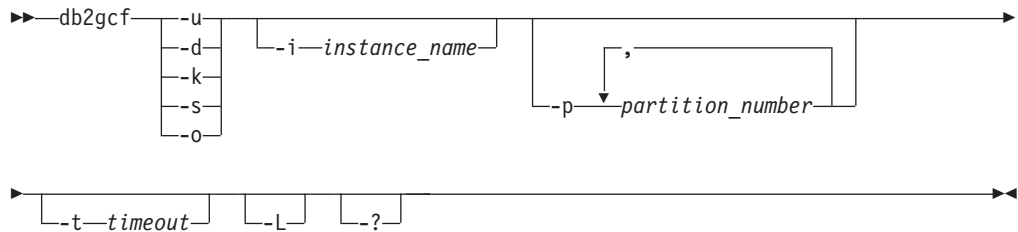
以下のいずれか。

- インスタンス所有者
- Linux および UNIX システムでは root でアクセスし、Windows オペレーティング・システムではローカル管理者権限でアクセスします。

必要な接続

なし

コマンド構文



コマンド・パラメーター

- u** 現在のデータベース・パーティション・サーバー (ノード) の指定のインスタンスの指定のデータベース・パーティションを開始します。
- d** 指定のインスタンスの指定のデータベース・パーティションを停止します。
- 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。
- s** 指定のデータベース・パーティション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データベース・パーティションは、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。
- o** それぞれの可能なアクションのデフォルトのタイムアウト値に戻します。 **-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 partition_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データベース・パーティション番号を指定します (このコマンドではリモート・パーティションはモニターされません)。スペースを使用せずに、コンマで区切ってパーティション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す。
- 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.log` 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、パーティション 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
Partition 0 : Success
```

2. 以下の例は、パーティション 0 のインスタンス `stevera` の状況に戻します。

```
db2gcf -s -p 0 -i stevera
```

以下の出力が戻されます。

```
Instance : stevera
DB2 State
Partition 0 : Available
```

3. 以下の例は、パーティション 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
Partition 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、指定されたパーティションからのすべてのプロセスが除去されないようにします。むしろ、インスタンス (全パーティション) 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。

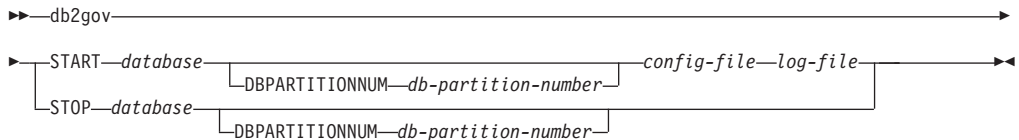
許可

以下のいずれか。

- *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_all 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、sqllib 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、sqllib 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

イリティーは、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 5.3 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
```

互換性

バージョン 8 より前のバージョンとの互換性 :

- キーワード NODENUM は、DBPARTITIONNUM の代わりとして使用できます。

db2govlg - DB2 管理プログラム・ログ照会

指定したタイプのレコードを管理プログラム・ログ・ファイルから抽出します。DB2 管理プログラムは、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。

許可

なし

コマンド構文

```
▶▶ db2govlg log-file dbpartitionnum db-partition-number ▶▶
▶▶ rectype record-type ▶▶
```

コマンド・パラメーター

log-file 照会するログ・ファイル (複数可) のベース名。

dbpartitionnum *db-partition-number*

管理プログラムを実行しているデータベース・パーティションの番号。

rectype *record-type*

照会するレコードのタイプです。次のレコード・タイプが有効です。

- START
- FORCE

- NICE
- ERROR
- WARNING
- READCFG
- STOP
- ACCOUNT

互換性

バージョン 8 より前のバージョンとの互換性：

- キーワード `nodenum` は、`dbpartitionnum` の代わりとして使用できます。

db2gppmap - 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、`db2gppmap` 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

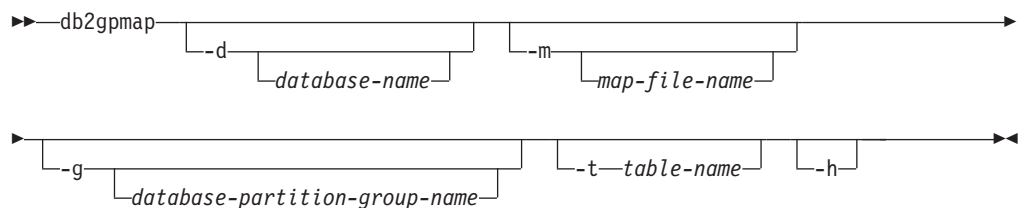
以下の両方が必要です。

- システム・カタログ表に対する読み取りアクセス
- `db2gppmap.bnd` に対するパッケージの `BIND` 特権と `EXECUTE` 特権

必要な接続

`db2gppmap` を使用する前に、データベース・マネージャーを開始し、`db2gppmap.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、`db2gppmap` 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*

分散マップを保管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
```

db2hc - ヘルス・センターの開始

ヘルス・センターを開始します。ヘルス・センターとは、データベース・システムのヘルス状態の全体を表示するのに使用されるグラフィカル・インターフェースです。ヘルス・センターを使用して、ヘルス・インディケータ上のアラートに関する詳細や推奨事項を表示し、アラート解決のための処置を取ることができます。

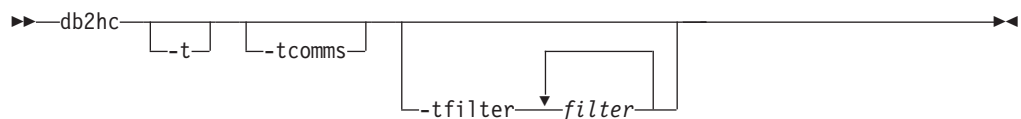
許可

情報の表示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

インスタンス

コマンド構文



コマンド・パラメーター

-t 初期化コードに対する NavTrace をオンにします。このオプションを使用するのは、DB2 サポートにそうするよう指示された場合のみです。

-tcomms

トレースを通信イベントのみを対象とするように制限します。このオプションを使用するのは、DB2 サポートにそうするよう指示された場合のみです。

-tfilter *filter*

トレースを、指定したフィルター (1 つ以上) を含む項目のみを対象とするように制限します。このオプションを使用するのは、DB2 サポートにそうするよう指示された場合のみです。

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 システムでのみ使用できます。

許可

以下のいずれか。

- Root 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 プラットフォームでのみ使用できます。

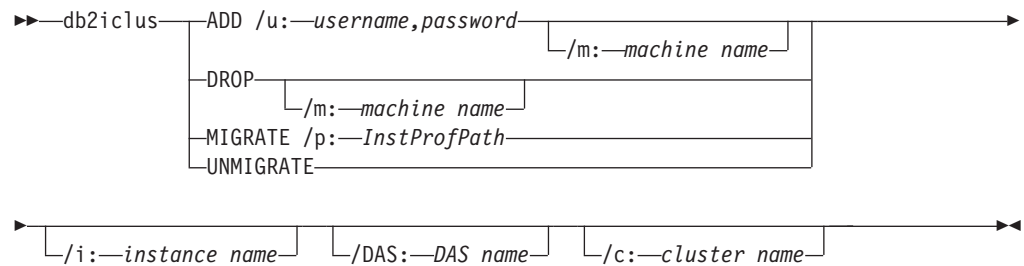
許可

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ー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:>db2wolfi i
ok
```

db2wolfi 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、クラスター管理には表示されません。

```
c:>db2wolfi u
ok
c:>db2wolfi 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ー・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 はクラスター・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 の所有であることが必要です。

7. WA26 から、db2iclus コマンドを使用して、他のマシンを DB2 クラスター・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 インスタンスを作成します。

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\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。

db2icrt コマンドにより、インスタンス所有者のホーム・ディレクトリに DB2 インスタンスが作成されます。

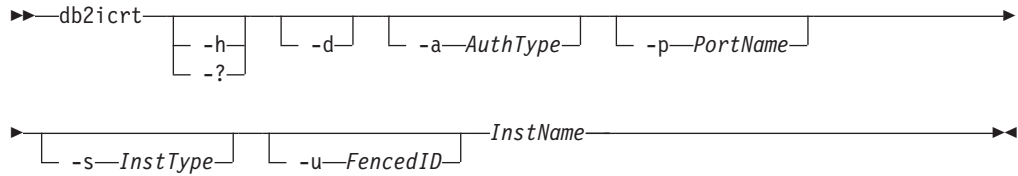
注: このコマンドは、Linux および UNIX オペレーティング・システム上の DB2 の非ルート・インストールには利用できません。

許可

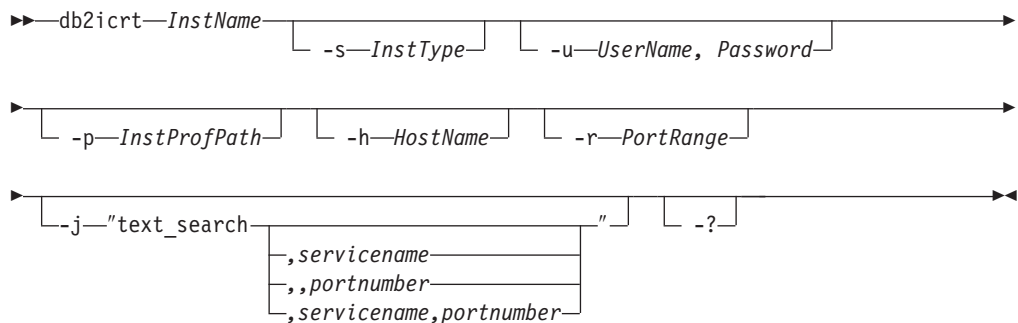
Linux および UNIX オペレーティング・システムでは root でアクセスし、Windows オペレーティング・システムではローカル管理者権限でアクセスします。

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



Windows オペレーティング・システムの場合



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

-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。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。

-p PortName

インスタンスが使用するポート名または番号を指定します。このオプションは、クライアント・インスタンスには適用されません。

-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。

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。これは、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。これは、DB2 Personal Edition のデフォルトのインスタンス・タイプです。

ese DPF 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。これは、DB2 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。これは、DB2 Workgroup Server Edition、DB2 Express または Express-C Edition、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。

-u Fenced ID

fenced ユーザー定義関数および fenced ストアド・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。これは、db2icrtコマンドの最後の引数にする必要があります。

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。

-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。これは、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。これは、DB2 Personal Edition のデフォルトのインスタンス・タイプです。

ese DPF 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。ESE インスタンス・タイプおよび DPF インスタンスを作成するには、

```
-s ese -u Username, Password
```

オプションを db2icrt とともに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。これは、DB2 Workgroup Server Edition、DB2 Express または Express-C Edition、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。

-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
```

-j *"text_search"*

DB2 Text Search をデフォルトのサービス名および生成されたポート値で構成することを試行します。

-j "text_search"

,servicename

自動生成されたポート番号で、またはそれがサービス・ファイル内に既に予約されている場合はそのサービス名に割り当てられたポート番号と同じ番号で、サービス名を予約します。

-j "text_search,servicename"

,portnumber

ポート番号をデフォルトのサービス名で予約します。有効なポートは、1024 - 65535 の範囲になければなりません。

-j "text_search,,portnumber"

-? 使用法情報を表示します。

例

- 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

- `instance_home/sqlllib/db2tss/config` フォルダは、db2icrt によって Linux および UNIX オペレーティング・システム上に作成されます。sqlllib ディレクトリーの外部の領域へのシンボリック・リンクを使用することをお勧めします。
- Linux および UNIX オペレーティング・システムでは、`-s` オプションは、システムの全機能を使用しないインスタンスの作成を意図して用意されています。例えば、UNIX オペレーティング・システム上で Enterprise Server Edition (ESE) を使用していてもパーティション機能を使用する意思がない場合は、オプション `-s WSE` を使用して Workgroup Server Edition (WSE) インスタンスを作成できます。
- Microsoft Cluster Server をサポートする DB2 インスタンスを作成するには、まずインスタンスを作成し、それから `db2mcs` コマンドを使用してそれが MSCS インスタンスで稼働するようにマイグレーションします。
- Linux および UNIX オペレーティング・システムでは、1 つのユーザー名について作成できるインスタンスは 1 つだけです。既にインスタンスのあるユーザー名に対してインスタンスを作成するには、新しいインスタンスを作成する前に既存のインスタンスをドロップする必要があります。
- Windows オペレーティング・システムでは、複数のインスタンスを作成できます。
- DB2 インスタンスを作成するには、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。
- `db2isetup` コマンドを使用して、サポートされるすべての Linux および UNIX オペレーティング・システムで、グラフィカル・インターフェースを使って DB2 インスタンスを作成および更新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、`db2icrt` を実行することはサポートされていません。

db2idrop - インスタンスの除去

`db2icrt` によって作成された DB2 インスタンスを除去します。

ドロップできるインスタンスは、db2idrop を発行している発行元と同じ DB2 コピーに関する db2ilist によってリスト表示されるインスタンスだけです。

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・システム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\bin* ディレクトリーにあります。*DB2PATH* は、DB2 コピーがインストールされている場所です。

注: このコマンドを使用して、Linux および UNIX オペレーティング・システムに非ルート・インストールされた DB2 インスタンスをドロップすることはできません。非ルート DB2 コピーをアンインストールするオプションのみあります。詳しくは、以下の『使用上の注意』を参照してください。

許可

Linux および UNIX オペレーティング・システムでは root アクセス、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

Linux および UNIX オペレーティング・システムの場合

```
db2idrop InstName [-d] [-f] [-h | -?]
```

Windows オペレーティング・システムの場合

```
db2idrop InstName [-f] [-h]
```

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

InstName

インスタンスの名前を指定します。

- d DB2 Service によって使用されるデバッグ・モードに入ります。
- 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 | -? 使用情報を表示します。

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。

- 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。

例

- Linux および 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

- パーティション・データベース環境では、ドロップするインスタンスに複数のデータベース・パーティションが属する場合には、各データベース・パーティション上で db2idrop コマンドを実行し、各データベース・パーティション上の DB2 レジストリーを更新しなければなりません。
- インスタンスをドロップしてもデータベースはドロップされないため、データを失うことはありません。インスタンスをドロップした後にインスタンスを再作成し、それからデータベースをカタログしてデータベースを使用可能にすることができます。db2cfexp および db2cfimp コマンドを使用して、インスタンス・プロファイルをエクスポートおよびインポートすることができます。
- インスタンスをドロップする前に、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管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/sql/lib/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バック・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・システム上で非ルート・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。

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。

db2imigr - インスタンスのマイグレーション

Linux および UNIX オペレーティング・システムでは、DB2 データベース・システムのインスタンスを、旧バージョン (DB2 データベース・システムの現行バージョンへのマイグレーションがサポートされているもの) から、db2imigr コマンドを実行している DB2 コピーの現行バージョンにマイグレーションします。Windows オペレーティング・システムでは、過去のリリースから現行リリースにインスタンスをマイグレーションします。

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 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場所です。

注: このコマンドは、Linux および UNIX オペレーティング・システム上の DB2 の非ルート・インストールには利用できません。

Windows オペレーティング・システムでは、db2imigr コマンドは、インスタンスのマイグレーション先の DB2 コピーから実行してください。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、マイグレーション後にインスタンス・プロファイルが元の場所に残ります。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 コピーのインストール先)。

インスタンスのマイグレーション前に、ローカル・データベースのマイグレーション準備ができていないことを検証するために、db2imigr コマンドは db2ckmig コマンドを呼び出します。db2ckmig コマンドが何らかのエラーを戻した場合、マイグレーションは続行されません。

許可

Linux および UNIX オペレーティング・システムでは root アクセス、Windows オペレーティング・システムではローカル管理者。

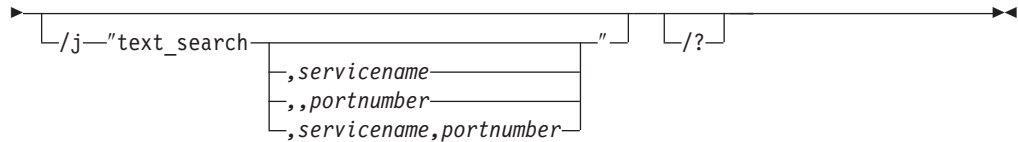
コマンド構文

Linux および UNIX オペレーティング・システムの場合

```
db2imigr [-d] [-a AuthType] [-u FencedID] InstName
```

Windows オペレーティング・システムの場合

```
db2imigr InstName /u:username,password [-p:instance-profile-path]
/q
/a:authType
```



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。

-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・インスタンスにマイグレーションする際に必要です。

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・インスタンスをマイグレーションする場合には、このオプションは必須です。

/p:instance-profile-path

マイグレーション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 db2imigr 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 をデフォルトのサービス名および生成されたポート値で構成することを試行します。

```
/j "text_search"
```

```
,servicename
```

自動生成されたポート番号で、またはそれがサービス・ファイル内に既に予約されている場合はそのサービス名に割り当てられたポート番号と同じ番号で、サービス名を予約します。

```
/j "text_search,servicename"
```

,portnumber

ポート番号をデフォルトのサービス名で予約します。有効なポートは、1024 - 65535 の範囲になければなりません。

`/j "text_search,,portnumber"`

`/? db2imigr` コマンドの使用方法を表示します。

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

- `db2imigr` コマンドは、マイグレーション前のバージョンの `/usr/lib` および `/usr/include` に存在するシンボリック・リンクをすべて除去します。オペレーティング・システムのライブラリー環境変数を使用して `libdb2` を検索するのではなく、`/usr/lib` からそれを直接ロードするアプリケーションがある場合、`db2imigr` の実行後、そのアプリケーションが正常に実行できなくなる可能性があります。
- `db2imigr` 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マイグレーションする場合、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・ロケーションにマイグレーションされません。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マイグレーションは、インスタンスのマイグレーション後に実行されます。
- `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、`db2imigr` コマンドを実行すること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。このコマンドはスプリット・ミラー・データベースに対してのみ実行可能であり、スプリット・ミラーを使用するには、その前にこのコマンドを実行しておく必要があります。

許可

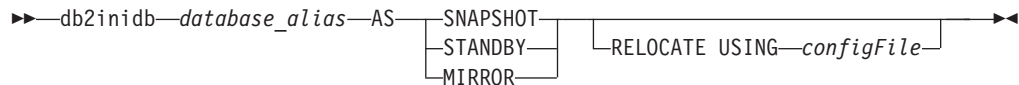
以下のいずれか。

- `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。

MIRROR

ミラーリングされたデータベースを、1 次データベースをリストアするために使用できるバックアップ・イメージとして使用することを指定します。

RELOCATE USING *configFile*

データベースをスナップショット、スタンバイ、またはミラーとして初期化する前に、指定された *configFile* の中でリストされている情報に基づいて、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、833 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 to *database-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せん。

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 を実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_all 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_all 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

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。

`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` 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ない場合には、このことが必要になります。

db2inspf - 検査結果のフォーマット

このユーティリティーは、`INSPECT CHECK` 結果からのデータを ASCII 形式にフォーマットします。このユーティリティーを使用して、検査の詳細を表示します。`db2inspf` ユーティリティー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することができ、エラー、警告、おわ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。

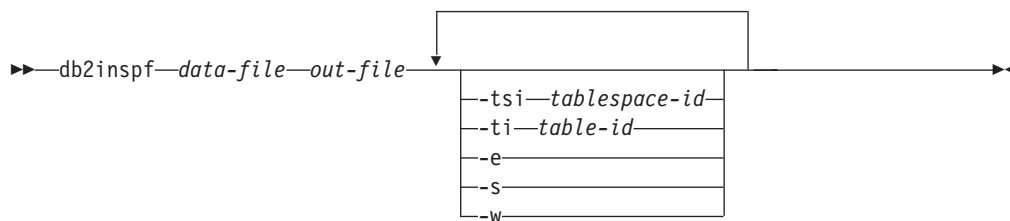
許可

このユーティリティー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ーを実行するには、それらの読み取り許可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 *tablespace-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、以下を実行します。

```
db2inspf tbschk.log tbschk_esw.txt -e -s -w
```

db2iprune - インストール・イメージ・サイズの削減コマンド (Windows)

`db2iprune` コマンドは、DB2 Windows製品のインストール・イメージのサイズを、DB2 製品をインストールする前に削減します。

このツールは、DB2 の大規模デプロイメント、および DB2 のアプリケーション内への組み込みに役立ちます。これは、入力ファイルおよび `db2iprune` 実行可能ファイル (`db2iprune.exe`) から構成されます。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フィーチャーおよび言語を指定するために使用できます。 `db2iprune` コマンドは、それらのフィーチャーおよび言語に関連付けられたキャビネット (`.cab`) 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。

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—————►
```

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フィーチャーおよび言語を指定するために使用できます。

-p *root_directory_path*

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.exe` 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。

例

入力ファイルが `c:\db2rtcl.prn` にあり、DB2 `setup.exe` ファイルが `d:*` にあり、整理された IBM Data Server Runtime Client イメージを `e:\compact_rtcl` ディレクトリーにコピーする場合、IBM Data Server Runtime 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:\db2rtcl.prn -p d:\* -o e:\compact_rtcl
```

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 の次のディレクトリーにあります。

```
dvd_drive:\db2\windows\utilities\db2iprun
```

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、インスタンスの作成および既存のインスタンスでの新機能の構成のためのグラフィック・ツールです。

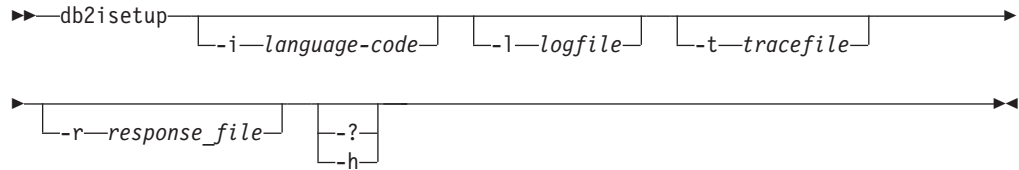
許可

ルート・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 権限が必要です。非ルート・インストールの場合、非ルート・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



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。ルート・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/db2isetup.log` になります。非ルート・インストールの場合、デフォルトのログ・ファイルは `/tmp/db2isetup_userID.log` です (*userID* は非ルート・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。

-? | -h 出力使用情報。

使用上の注意

- このインスタンス・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 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、インスタンス・セットアップに関連する機能を提供するだけです。
- 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。
- `db2isetup` は、サポートされているすべての Linux および UNIX のシステムで実行できます。

db2iupdt - インスタンスの更新

Linux および UNIX オペレーティング・システムでこのコマンドは、指定された DB2 インスタンスを更新します。Windows オペレーティング・システムの場合、このコマンドはインスタンスのリリース・レベルを更新します。また、ある DB2 コピーから別のコピーにインスタンスを移動する場合にも使用します。

Linux および UNIX システムで、db2iupdt コマンドは、同じまたは別の DB2 データベース・インストール・ディレクトリーに関連するインスタンスのうち、同じバージョンのものに対して発行できます。いずれの場合でも、db2iupdt コマンドの発行元と同じディレクトリー内にあるコードに対して実行されるよう、インスタンスが更新されます。このコマンドを発行することが必要なのは、以下の場合です。

- 新しい DB2 データベース製品またはフィーチャーを、DB2 インスタンスに関連するインストール・ディレクトリーにインストールする場合 (フィックスパックのインストール後を除く)。
- DB2 インスタンスを、あるインストール・パスから、DB2 データベース・システムの同じバージョンの現行パスに移す場合。

注: このコマンドは、Linux および UNIX オペレーティング・システム上の DB2 の非ルート・インストールには利用できません。

Linux および UNIX オペレーティング・システムでは、これは *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。

Linux および UNIX オペレーティング・システム上にフィックスパックをインストールした後は、自動的に db2iupdt コマンドが実行されます。

Windows オペレーティング・システムでは、インスタンスは、db2iupdt 実行元の DB2 コピーに移動します。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、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。メジャー・リリースを別のものに変更する場合には、このコマンドではなく db2imigr コマンドを使用してください。このユーティリティーは *DB2PATH\sqlib\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のインストール先)。

db2iupdt を使用してインスタンスを更新するには、まず、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。

許可

UNIX および Linux オペレーティング・システムでは root アクセス、Windows オペレーティング・システムではローカル管理者。

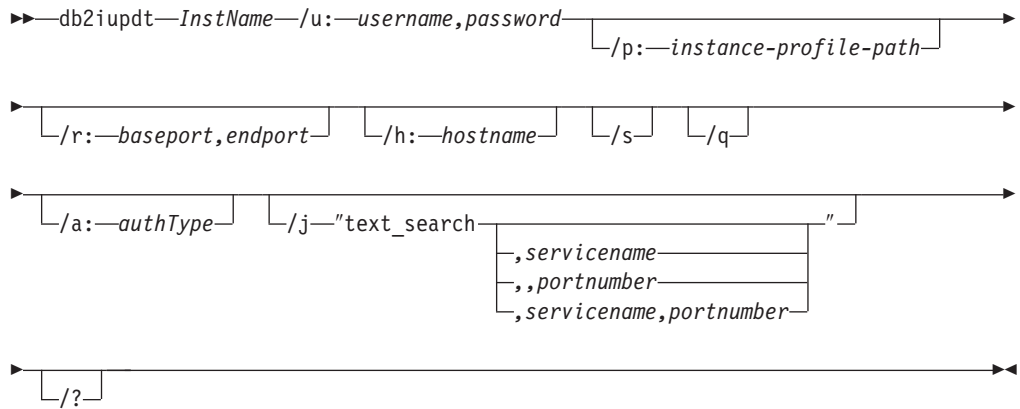
コマンド構文

UNIX および Linux オペレーティング・システムの場合

```
db2iupdt [-h] [-d] [-k] [-D] [-s] [-a AuthType]
```



Windows オペレーティング・システムの場合



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。
- 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 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。
- s 既存の SPM ログ・ディレクトリーを無視します。
- 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。

-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。このオプションが必要になるのは、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 オプションを使用する場合) は、-u オプションは必要ありません。-u オプションによって、既存のインスタンスの fenced ユーザーを変更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。

-e すべてのインスタンスを更新します。

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 をデフォルトのサービス名および生成されたポート値で構成することを試行します。

/j "text_search"

,servicename

自動生成されたポート番号で、またはそれがサービス・ファイル内に既に予約されている場合はそのサービス名に割り当てられたポート番号と同じ番号で、サービス名を予約します。

/j "text_search,servicename"

,portnumber

ポート番号をデフォルトのサービス名で予約します。有効なポートは、1024 - 65535 の範囲になければなりません。

/j "text_search,,portnumber"

/? db2iupdt コマンドの使用方法を表示します。

例 (UNIX および Linux)

- インスタンス db2inst2 は、インストール・パス *DB2DIR1* に関連しています。*DB2DIR2* の同じシステム上に、*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別の DB2 データベース製品がインストールさ

れています。インストールされている DB2 データベース製品を使用するためのインスタンスを、*DB2DIR1* から *DB2DIR2* に更新するには、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 データベース製品が *DB2DIR1* のもの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
```

使用上の注意

UNIX、Linux、および Windows オペレーティング・システムの場合

- db2iupdt コマンドを使用することによって DB2 インスタンスをあるインストール・ロケーションから現行のインストール・ロケーションに更新する場合、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

- 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、db2iupdt を実行すること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。DB2 バージョン 8 データベースには、既に JDBC パッケージがインストールされているので、通常このコマンドを必要とするのは以前のサーバーだけです。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

以下のいずれか。

- *sysadm*
- *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



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

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。4 つの DB2 分離レベルと 2 つのカーソルの HOLD 設定があるので、このオプションで指定されるバインドされる動的パッケージ数は $4 \times 2 = 8$ 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。一度構成すると、LDAP ユーザーの DN およびパスワードがこのユーザーの環境に保管され、それらは DB2 が LDAP ディレクトリーにアクセスする際に必ず使用されます。こうすると、LDAP コマンドまたは API を発行する際に、LDAP ユーザーの DN およびパスワードを指定する必要はなくなります。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・ユーザーの証明書が使用されます。

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

コントロール・センターがない場合に基本ライセンス機能を実行します。ローカル・システムにインストールされたライセンスおよびポリシーを追加、除去、リスト、および変更します。

注: プロセッサ値単位 (PVU) ライセンス交付構造の下で、個々のプロセッサ・コアに特定の数の値単位が割り当てられます。ソフトウェア・プログラムをデプロイするプロセッサ・コアごとのプロセッサ値単位の総数を取得しなければなりません。IBM 社は引き続きプロセッサがチップ上の個々のプロセッサ・コアになるように定義しています。例えば、二重コアのチップにはプロセッサ・コアが 2 つ含まれます。

個々のソフトウェア・プログラムには、値単位ごとに固有の価格があります。個々のソフトウェア・プログラムをデプロイする場合の総コストを判別するには、値単位当たりのプログラム価格に、必要なプロセッサ値単位の総数を乗算します。

許可

UNIX および Linux オペレーティング・システムでは、ライセンスを除去する場合に限り *sysadm* 権限が必要です。

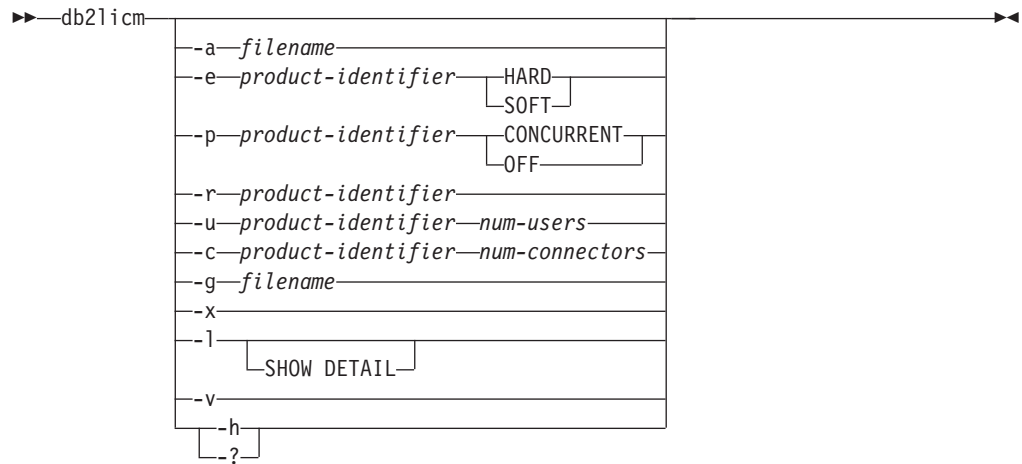
Windows オペレーティング・システムの場合

- `-a`、`-r`、または `-x` コマンド・パラメーターを使用するには、ローカル管理者グループまたはパワー・ユーザー・グループに属している必要があります。
- `-c`、`-e`、`-p`、`-r`、または `-u` コマンド・パラメーターを使用するには *sysadm* 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

-a *filename*

製品のライセンスを追加します。有効なライセンス情報を含むファイル名を指定します。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.lic
db2licm -p db2consv concurrent
db2licm -r db2ese
db2licm -u db2wse 10
```

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:           "08/31/2008"
Product identifier:    "db2ese"
Version information:   "9.5"
```

使用上の注意

II (Information Integrator) 製品の製品 ID 値には、以下のような変更が加えられています。

表 36. 変更された II (Information Integrator) 製品の製品 ID 値

製品名	古い製品 ID	新しい製品 ID
WebSphere® Replication Server	wsiir	wsrs
WebSphere フェデレーション・サーバー	wsiif	wsfs
WebSphere Data Event Publisher	wsiip	wsep

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ーを使用すると、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ーは `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/ofer/ofer/NODE0000/S0000001/DB2TSCHG.HIS
```

```
db2logsForRfwd DB2TSCHG.HIS -all
```

db2look - DB2 統計および DDL 抽出ツール

必要とされるデータ定義言語 (DDL) ステートメントを抽出して、テスト・データベース上に実働データベースのデータベース・オブジェクトを再生成します。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。

このツールは、テスト・データベース内のオブジェクトに関する統計を複製するために必要な UPDATE ステートメントを生成できます。また、このツールを使用して、UPDATE DATABASE CONFIGURATION および UPDATE DATABASE MANAGER CONFIGURATION コマンドおよび db2set コマンドを生成し、テスト・データベース上の照会オブティマイザー関連の構成パラメーターとレジストリ変数を、実働データベースでの値に合わせて設定できます。

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、便利ながことが多くあります。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。テスト・システム用のカタログ統計と構成パラメーターの両方が、実動システムのものと同じように更新されていなければなりません。このツールを使用すると、アクセス・プランが、実動システムで使用されるものと類似しているテスト・データベースを作成することが可能になります。

db2look コマンドによって生成される DDL 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 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。

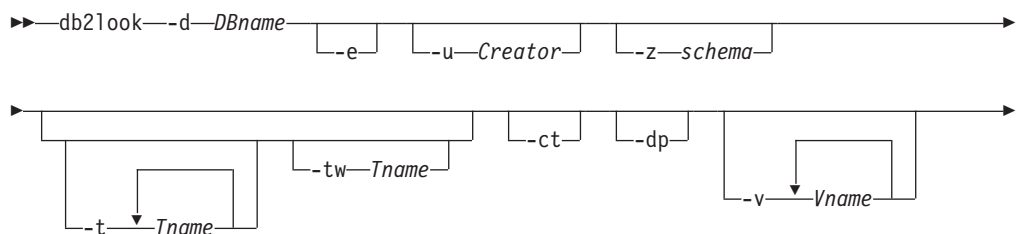
表スペース・コンテナ DDL を生成する場合（つまり、sqlbotcq、sqlbftcq、および sqlbctcq の API を呼び出す場合）などでは、以下のいずれか 1 つが必要です。

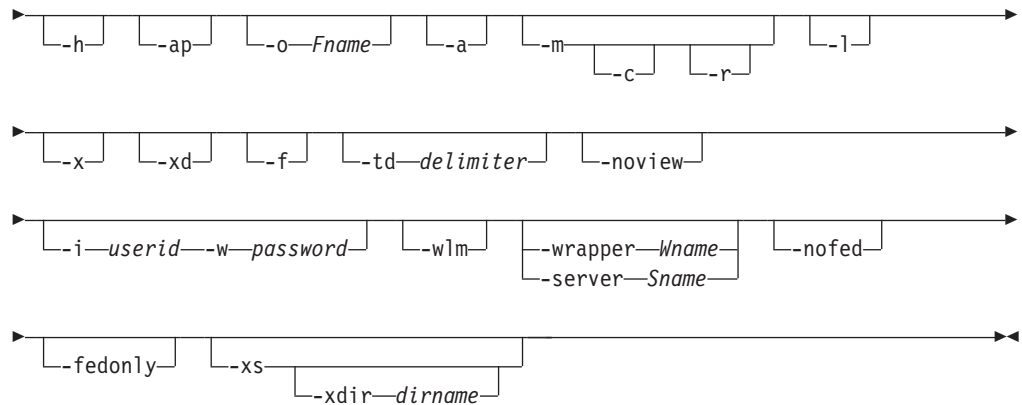
- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*

必要な接続

なし

コマンド構文





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 Database for Linux, UNIX, and Windows または DB2 Version 9.1 for z/OS (DB2 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 データベースの場合には、 db2look ユーティリティーは、 OS/390 オブジェクトおよび z/OS オブジェクト用の DDL および UPDATE 統計ステートメントを抽出します。これらの DDL および UPDATE 統計ステートメントは、 DB2 Database for Linux, UNIX, and Windows データベースには適用できませんが、 DB2 for z/OS データベースには適用できません。これは、 OS/390 オブジェクトおよび z/OS オブジェクトを抽出して、それらを DB2 Database for Linux, UNIX, and Windows データベースで再作成しようとするユーザーに役立ちます。

DBname が DB2 for z/OS データベースの場合、 db2look コマンドの出力は以下のものに制限されます。

- 表、索引、ビュー、およびユーザー定義特殊タイプ用の DDL の生成
- 表、列、列分散および索引用の UPDATE 統計ステートメントの生成

-e 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。 -e オプションを使用する場合には、以下のデータベース・オブジェクト用の DDL を抽出します。

- 監査ポリシー
- スキーマ
- 表
- ビュー
- マテリアライズ照会表 (MQT)
- 別名
- 索引
- トリガー
- シーケンス
- ユーザー定義特殊タイプ
- 主キー、参照整合性、およびチェック制約

- ユーザー定義構造化タイプ
- ユーザー定義関数
- ユーザー定義メソッド
- ユーザー定義トランスフォーム
- ラッパー
- サーバー
- ユーザー・マッピング
- ニックネーム
- タイプ・マッピング
- 関数テンプレート
- 関数マッピング
- SPECIFICATION ONLY 指定の索引
- ストアード・プロシージャ
- ロール
- トラストッド・コンテキスト
- グローバル変数
- 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
- セキュリティー・ラベル

db2look コマンドによって生成される DDL を使用して、ユーザー定義関数を正常に再作成することができます。ただし、ユーザー定義関数を使用できる状態にするには、特定のユーザー定義関数 (EXTERNAL NAME 節など) が参照するユーザー・ソース・コードが使用できる状態でなければなりません。

-u *Creator*

作成者 ID。出力をこの作成者 ID があるオブジェクトだけに制限します。オプション **-a** を指定した場合、このパラメーターは無視されます。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** オプションを使用します。

-z *schema*

スキーマ名。出力をこのスキーマ名の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** オプションを使用します。このパラメーターが指定されない場合は、すべてのスキーマ名のオブジェクトが抽出されます。**-a** オプションを指定した場合、このパラメーターは無視されます。このオプションは、フェデレーテッド DDL では無視されます。

-t *Tname1 Tname2 ... TnameN*

表名のリストです。表のリストにある特定の表への出力を制限します。表の最大数は 30 です。表名はブランク・スペースで区切られます。大文字と小文字を区別する名前および 2 バイト文字セット (DBCS) 名は、`¥"MyTabLe ¥"` のように、円記号と二重引用符の区切り文字で囲む必要があります。複数語表名の場合、区切り文字は ("`¥"My Table¥"`") のように引用符の中に置く必要があります、そのようにしてその対が一語ごとにコマンド行プロ

セッターで評価されないようにします。複数語表名が ("My Table" のように) 円記号と二重引用符で囲まれていない場合、すべての語は大文字に変換され、db2look コマンドは ("MY TABLE" のように) 大文字の表を探します。-t が -l と共に使用された場合、その組み合わせでは DB2 バージョン 9.5 のパーティション表がサポートされます。

-tw *Tname*

Tname に指定したパターン基準に一致する表名の DDL を生成します。また、戻されたすべての表のすべての従属オブジェクトの DDL も生成します。*Tname* は、1 つの値だけでもかまいません。*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。*Tname* 内の他のすべての文字は、その文字そのものを表します。-tw を指定した場合は、-t オプションが無視されます。

-ct

オブジェクト作成時刻に基づいて DDL を生成します。オブジェクト作成時刻に基づいて DDL を生成した場合、オブジェクト DDL のすべてが従属関係の点で正しい順序で表示されることが保証されません。db2look コマンドで、-ct オプションと共に指定することがサポートされているオプションは、-e、-a、-u、-z、-t、-tw、-v、-l、-noview、-wlm だけです。

-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、スキーマをドロップする場合、そのスキーマに依存する表が存在するなら、そのドロップは失敗します。あるいは、ユーザー定義のタイプ/関数をドロップする場合、それに依存する他のタイプ、関数、トリガー、または表が存在するなら、そのドロップは失敗します。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。表に RESTRICT ON DROP 属性が設定されている場合、それはドロップできません。

-v *Vname1 Vname2 ... VnameN*

指定したビューの DDL を生成します。ビューの最大数は 30 です。-t オプションを指定した場合は、-v オプションが無視されます。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。

-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。

-ap

必須の AUDIT USING ステートメントを生成し、監査ポリシーを他のデータベース・オブジェクトと関連付けます。

-o *Fname*

出力を *filename.sql* に書き込みます。このオプションを指定しない場合、出力は標準出力に書き込まれます。ファイル名が拡張子付きで指定されている場合、出力はそのファイルに書き込まれます。

-a

このオプションが指定されている場合には、特定の作成者 ID で作成されたオブジェクトだけに出力が制限されることはありません。すべてのユーザーによって作成されたすべてのオブジェクト (作動不能オブジェクトを含む)

が対象になります。例えば、このオプションと `-e` オプションが共に指定される場合、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。このオプションと `-m` オプションが共に指定される場合、データベース内のすべてのユーザー作成表および索引用の UPDATE 統計ステートメントが抽出されます。 `-u` と `-a` のどちらも指定しない場合には、環境変数 `USER` が使用されます。UNIX オペレーティング・システムでは、この変数を明示的に設定する必要はありません。しかし Windows システムの場合、`USER` 環境変数にデフォルト値がありません。`SYSTEM` 変数の中のユーザー変数を設定するか、または `set USER=username` をセッションに発行する必要があります。

- m** 必要な UPDATE ステートメントを生成して、表、統計ビュー、列、および索引についての統計を複製します。
- c** このオプションを `-m` オプションと共に指定する場合、`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。
- r** このオプションと `-m` オプションを共に指定する場合、`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。

注: 模倣モード (`-m` オプション) で `db2look` を使用して作成されたコマンド・プロセッサ・スクリプトを他のデータベースに対して実行しようとする場合 (例えば、テスト・データベースのカatalog統計を実動データベースと一致させる場合)、データベースは両方とも同じコード・セットおよびデリトリリーを使用していなければなりません。

- l** このオプションを指定すると、`db2look` コマンドは、ユーザー定義の表スペース、データベース・パーティション・グループ、およびバッファ・プール用の DDL を生成します。以下のデータベース・オブジェクト用の DDL は、`-l` オプションを使用すると抽出されます。
 - ユーザー定義表スペース
 - 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・プール
- x** このオプションを指定すると、`db2look` コマンドは、許可 DDL (`GRANT` ステートメントなど) 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。
 - 表: `ALTER`、`SELECT`、`INSERT`、`DELETE`、`UPDATE`、`INDEX`、`REFERENCE`、`CONTROL`
 - ビュー: `SELECT`、`INSERT`、`DELETE`、`UPDATE`、`CONTROL`
 - 索引: `CONTROL`
 - スキーマ: `CREATEIN`、`DROPIN`、`ALTERIN`
 - データベース: `CREATEDB`、`BINDADD`、`CONNECT`、`CREWATE_NOT_FENCED`、`IMPLICIT_SCHEMA`
 - ユーザー定義関数 (UDF): `EXECUTE`
 - ユーザー定義メソッド: `EXECUTE`

- ストアード・プロシージャ: EXECUTE
 - パッケージ: CONTROL、BIND、EXECUTE
 - 列: UPDATE、REFERENCES
 - 表スペース: USE
 - シーケンス: USAGE、ALTER
 - ワークロード: USAGE
 - グローバル変数
 - ロール
 - セキュリティー・ラベル
 - 免除
- xd** このオプションを指定すると、オブジェクトの作成時に SYSIBM によって権限を付与されたオブジェクトの権限 DDL を含むすべての権限 DDL が、db2look コマンドによって生成されます。
- f** このオプションを使用して、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。このオプションが指定されていない場合のデフォルトはセミコロン (;) です。このオプションは、-e オプションを指定した場合に使用することをお勧めします。この場合、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ります。
- noview**
このオプションを指定すると、CREATE VIEW DDL ステートメントが抽出されません。
- i userid**
リモート・データベースで作業する場合には、このオプションを使用してください。
- w password**
-i オプションと共にこのパラメーターを使用すると、ユーザーは、リモート・システムにあるデータベースに対して db2look コマンドを実行できるようになります。db2look コマンドでは、リモート・システムにログオンするために、ユーザー ID およびパスワードが使用されます。リモート・データベースで作業を行っている場合は、リモート・データベースは、ローカル・データベースと同じバージョンである必要があります。db2look コマンドには、下位レベルまたは上位レベル・サポートはありません。
- wlm** このオプションは、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
 - WLM イベント・モニター
 - サービス・クラス
 - ワークロード
 - しきい値

- 作業クラス・セット
- 作業アクション・セット

-wrapper *Wname*

この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、以下のものが含まれます: CREATE WRAPPER、CREATE SERVER、CREATE USER MAPPING、CREATE NICKNAME、CREATE TYPE MAPPING、CREATE FUNCTION ... AS TEMPLATE、CREATE FUNCTION MAPPING、CREATE INDEX SPECIFICATION、および GRANT (ニックネーム、サーバー、索引への特権)。1 つのラッパー名のみがサポートされています。1 つも指定されない場合、または複数指定された場合は、エラーが返されます。このオプションでは、非リレーショナル・データ・ソースはサポートされません。

-server *Sname*

この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、以下のものが含まれます: CREATE WRAPPER、CREATE SERVER、CREATE USER MAPPING、CREATE NICKNAME、CREATE TYPE MAPPING、CREATE FUNCTION ... AS TEMPLATE、CREATE FUNCTION MAPPING、CREATE INDEX SPECIFICATION、および GRANT (ニックネーム、サーバー、索引への特権)。1 つのサーバー名のみがサポートされています。1 つも指定されない場合、または複数指定された場合は、エラーが返されます。このオプションでは、非リレーショナル・データ・ソースはサポートされません。

-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。このオプションが指定された場合、-wrapper および -server オプションは無視されます。

-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。

-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 の各オプションによって制御されます。

-xdir *dirname*

エクスポートされた XML 関連ファイルを、指定されたパスに配置します。このオプションが指定されていない場合、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。

例

- 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。db2look の出力は、以下のようにしてファイル db2look.sql に書き込みます。

```
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。 `db2look` の出力は、以下のようにしてファイル `db2look.sql` に書き込みます。

```
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、以下のようにしてファイル `db2look.sql` に書き込みます。

```
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。 `db2look` の出力は、以下のようにしてファイル `db2look.sql` に書き込みます。

```
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。 `db2look` の出力は、以下のようにしてファイル `db2look.sql` に書き込みます。

```
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。 `db2look` の出力は、以下のようにしてファイル `db2look.sql` に書き込みます。

```
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` ステートメントを生成します。 `db2look` の出力は、以下のようにしてファイル `db2look.sql` に書き込みます。

```
db2look -d department -f -o db2look.sql
```

- データベース `DEPARTMENT` にあるすべてのオブジェクト用の `DDL`、データベース `DEPARTMENT` にあるすべての表および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データベースおよびデータベース・マネージャー構成パラメーター用の `UPDATE` ステートメント、オプティマイザー関連レジストリー変数用の `db2set` ステートメント、およびデータベース `DEPARTMENT` にある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` を生成します。出力は、以下のようにしてファイル `db2look.sql` に書き込みます。

```
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。) `db2look` の出力は、以下のようにしてファイル `db2look.sql` に書き込みます。

```
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。 db2look の出力は、以下のようにしてファイル db2look.sql に書き込みます。

```
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。 db2look の出力は、以下のようにしてファイル db2look.sql に書き込みます。

```
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。 db2look の出力は、以下のようにしてファイル db2look.sql に書き込みます。

```
db2look -d department -e -t tab1 ¥"My TaB1E2¥" -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。 db2look 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。 db2look 出力がファイル out.sql に書き込まれます。

```
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。 db2look の出力は、以下のようにしてファイル db2look.sql に書き込みます。

```
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、ディレクトリー /home/ofer/ofer/ にエクスポートされます。 db2look 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 中の WLM 固有の DDL を排他的に生成します。

```
db2look -d department -wlm
```

データベース DEPARTMENT 中のすべてのオブジェクトに関する DDL を生成します。

```
db2look -d department -wlm -e -l
```

使用上の注意

Windows オペレーティング・システムにおいて、db2look コマンドは DB2 コマンド・ウィンドウから実行される必要があります。

既存のオプションのいくつかは、フェデレーテッド環境をサポートします。以下の db2look コマンド行オプションがフェデレーテッド環境で使用されます。

- -ap

使用されると、AUDIT USING ステートメントが生成されます。

- -e

使用されると、フェデレーテッド DDL ステートメントが生成されます。

- -x

使用されると、フェデレーテッド・オブジェクトへの特権を付与するために、GRANT ステートメントが生成されます。

- -xd

使用されると、システム付与の特権をフェデレーテッド・オブジェクトに追加するために、フェデレーテッド DDL ステートメントが生成されます。

- -f

使用されると、フェデレーテッド関連情報がデータベース・マネージャー構成から抽出されます。

- -m

使用されると、ニックネームの統計が抽出されます。

- -wlm

使用されると、WLM 固有の DDL が出力されます。

フェデレーテッド・システムを使用する能力は、フェデレーテッド DDL ステートメントを作成するために、データベース・マネージャー構成で有効にされる必要があります。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。

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・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 i5/OS[®]、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。

データベース・パーティション・フィーチャーを使用するシステム上で DDL を要求する場合、非アクティブ・データベース・パーティション上に存在する表スペースに関しては、DDL 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ついて確実に正しい DDL が生成されるようにするためには、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。

タイプ配列のセキュリティー・ラベル・コンポーネントに関する DDL を抽出する際に、抽出された DDL により生成されるセキュリティー・ラベル・コンポーネントの内部表記 (つまり、配列中の要素のエンコード) が、db2look の抽出元のデータベース中のセキュリティー・ラベル・コンポーネントの内部表記と正確に一致しないことがあります。この種の不一致は、タイプ配列のセキュリティー・ラベル・コンポーネントに変更が加えられ、1 つ以上の要素が追加された場合に

発生することがあります。この場合、ある表から抽出され、db2look 出力から作成された別の表に移動する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 バージョン 9 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。

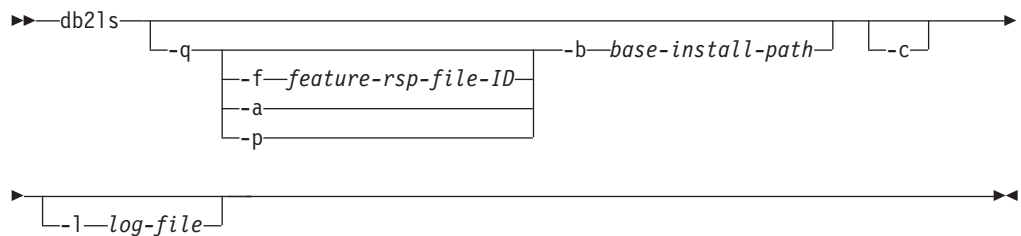
許可

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) のみ表示されます。

-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リストします。
- 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/v9
```

- 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/v9
```

- 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/v9 -f <feature>
```

使用上の注意

- ルートに /usr/local/bin 中での書き込み許可があるか、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。

非ルート・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。
- コマンドの正確なパス、つまり *inst_home/sqlllib/install/db2ls* 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

- (pkgadd、rpm、SMIT、または swinstall など) を使用して DB2 製品を照会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。
 - 非ルート・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。
 - 非ルート・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 にアクセス許可がある場合は、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、アクセス許可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。
- 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。

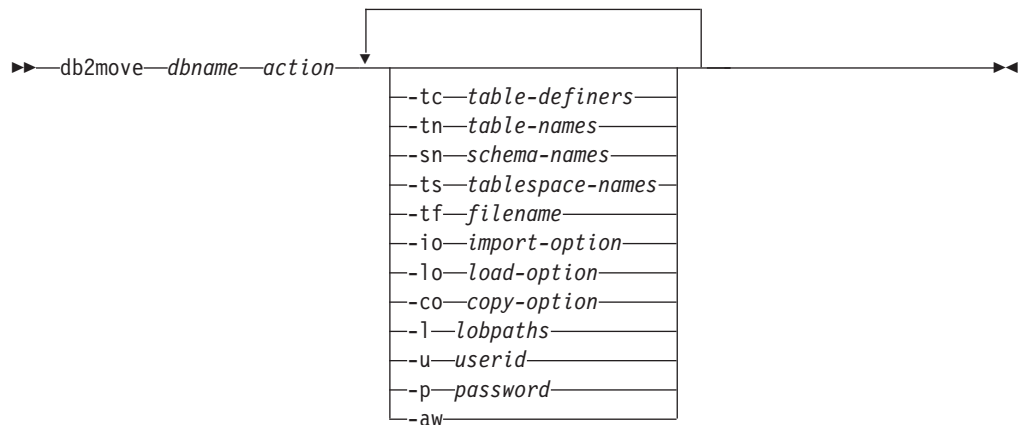
db2move - データベース移動ツール

このツールは、EXPORT/IMPORT/LOAD モードで使用された場合、ワークステーション上にある DB2 データベース間で、大量の表の移動を容易にします。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。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。

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 スキーマをターゲット・データベースへ複製します。 `-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。

下記は、各アクション中に生成されるファイルのリストです。

-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。表名間に空白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空白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空白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り、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 アクションがコピーである場合、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とユーザー/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/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、下記で説明するセキュリティの理由によります。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。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。

「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。

-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、現行ディレクトリーです。

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、それぞれをコマンド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ても)。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、**tbyname1** という名前、または **%tbyname2** 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 tbyname1,*tbyname2
```

- **SAMPLE** データベースのすべての表を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

- **db2move EXPORT** の後に **db2move IMPORT/LOAD** を指定すると、表データの移動が可能になります。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)、『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 `logretain` 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 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。

db2mqIsn - MQ Listener

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・プロシージャの情報には、ストアード・プロシージャ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。

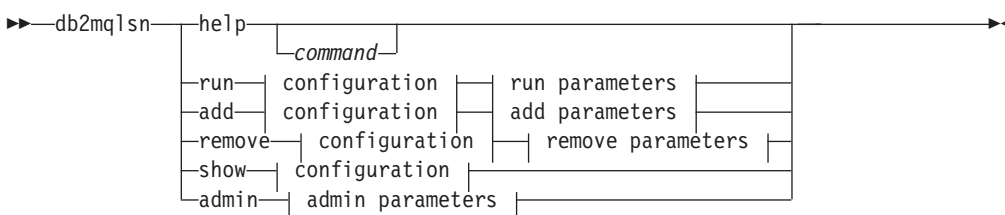
Windows オペレーティング・システムでは、このユーティリティは、DB2PATH%sqllib%bin ディレクトリーにあります。DB2PATH は、DB2 データベース製品の現行バージョンがインストールされている場所です。

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。

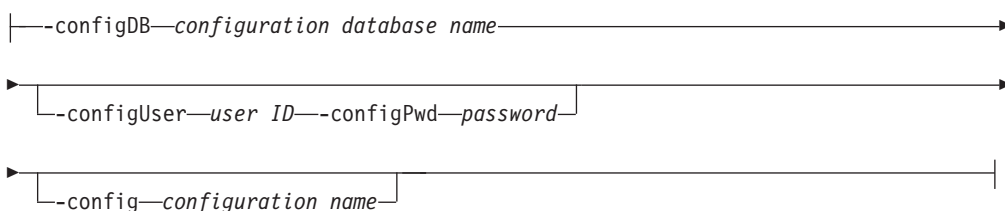
許可

- `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。

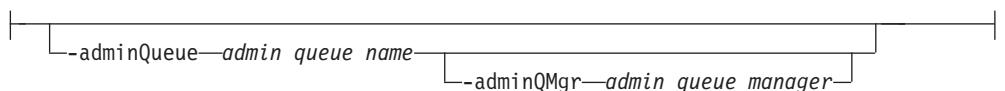
コマンド構文



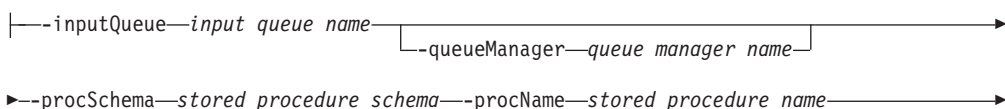
configuration:



run parameters:



add parameters:



```

▶--dbName--stored procedure database--dbUser--user ID--dbPwd--password--
└──────────────────────────────────────────────────────────────────────────┘
└──────────────────────────────────────────────────────────────────────────┘
└mqCoordinated┘ └numInstances--number of instances to run┘

```

remove parameters:

```

└--inputQueue--input queue name--queueManager--queue manager name--

```

admin parameters:

```

└--adminQueue--admin queue name--adminQueueList--namelist of admin queue names--adminQMGr--admin queue manager--
▶--adminCommand--shutdown┘
└──────────────────────────────────────────────────────────────────────────┘
└restart┘

```

コマンド・パラメーター

help *command*

特定のコマンドに関する詳細情報を提供します。コマンド名を指定しない場合は、一般的なヘルプ・メッセージが表示されます。

-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ー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ー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ーは 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ーは 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
```

db2mscs - 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

```
db2mscs --f:--input_file --u:--instance_name
```

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。パラメーターが指定されている場合、db2mscs ユーティリティはそのファイル名を入力ファイルとして使用します。このパラメーターが指定されない場合、db2mscs 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、db2mscs 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。

使用上の注意

db2mscs ユーティリティは、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です。このユーティリティは、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・ディスクに移動します。db2mscs ユーティリティは、ユーザーから渡される構成ファイルを、クラスタ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、db2mscs ユーティリティ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
```

DB2 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は、2 つのサンプル構成ファイルがあります。1 つは DB2MSCS.WSE 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.ESE で、これは、パーティション・データベース環境の例です。

db2mscs コマンドを発行する前にコントロール・センターがアクティブでないことを確認します。コントロール・センターがアクティブである場合、db2mscs ユーティリティーはエラーで終了します。

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。

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。

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 (*domainuser* 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。

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。

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。むしろ、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。

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。

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 接続が必要です。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。

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。

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。

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。

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。

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。

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。

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。

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。

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。

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。 **db2mscs** 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。

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、**db2mscs** 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。

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。なお、**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。

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** ユーティリティー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。

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・レベルの情報が戻される場合とは別に、このコマンドは、そのデータベース・パーティションのみに関する情報を戻します。リモート・サーバーの情報は戻しません。

許可

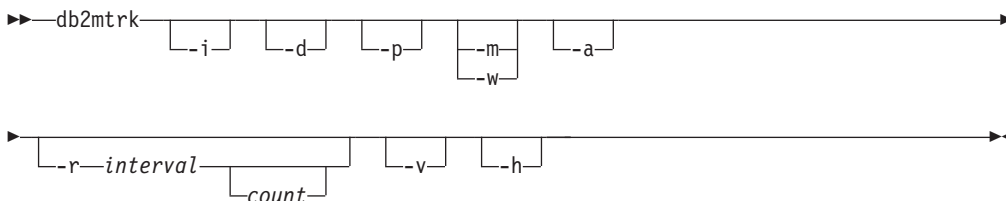
以下のいずれか。

- `sysadm`
- `sysctrl`
- `sysmaint`

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。
- 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
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 (他のメモリー)」とは、データベース管理システムの操作のオーバーヘッドに関連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・プール・ヒープの場合、括弧内に示された番号がバッファ・プール ID であるか、またはこのバッファ・プールがシステム・バッファ・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。

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（ノード）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（ノード）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。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 (ノード) 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

```
db2nrcrt -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。1 から 999 の範囲の番号を指定できます。

/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 通信に予約されているポートの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 `db2nrcrt` 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。

```
db2nrcrt /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(ノード)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 nodenum 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。

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・コマンド

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] [-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。

-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 db2imigr は非 root インストールでは使用できません。

db2rfe - 非 root インストール用 root フィーチャー使用可能化コマンド

このコマンドは、構成ファイルに基づいて DB2 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 特権を持つユーザー。

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.

** -----
** 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
```

db2nrupdt - 非 root インストールを行ったインスタンスの更新コマンド

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] [-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。
- 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 db2imigr は非 root インストールでは使用できません。

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のものとなります。現在このコマンドを使用できるのは、64 ビット・インスタンスの HP-UX 上と Solaris オペレーティング・システム上の DB2 だけです。

許可

- DB2 for HP-UX 版の DB2 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 アクセスの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 の場合は、 root アクセスの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・システム版の DB2 でのみ使用できます。
- 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。現行のカーネル・パラメーターはライブ・カーネルから直接取得されるので、それらは /etc/system 内の Solaris システム仕様ファイルにあるものと一致しないことがあります。ライブ・カーネルから取得されたカーネル・パラメーターが /etc/system にリストされているものと異なる場合、 /etc/system ファイルはリブートされないで変更されたか、またはファイル内に構文エラーが存在する可能性があります。 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・システム版の DB2 でのみ使用できます。
- 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・システム版の DB2 でのみ使用できます。

- 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・システム版の DB2 でのみ使用できます。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。以下は値を設定するためのガイドであり、`db2osconf` ユーティリティ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 に対して X 個の接続がある場合、DB2 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て行われます。

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 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 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 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ポ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、なくすることができます。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、4GB 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 4.67MB 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

DB2 データベース・システムのメモリー・セットから情報を取り出します。

許可

以下のいずれか。

- Linux および UNIX の場合、SYSADM、SYSMAINT、SYSCTRL、または SYSMON 権限レベル。

インスタンス所有者の 1 次グループとルートの 1 次グループとが同じであることはできません。

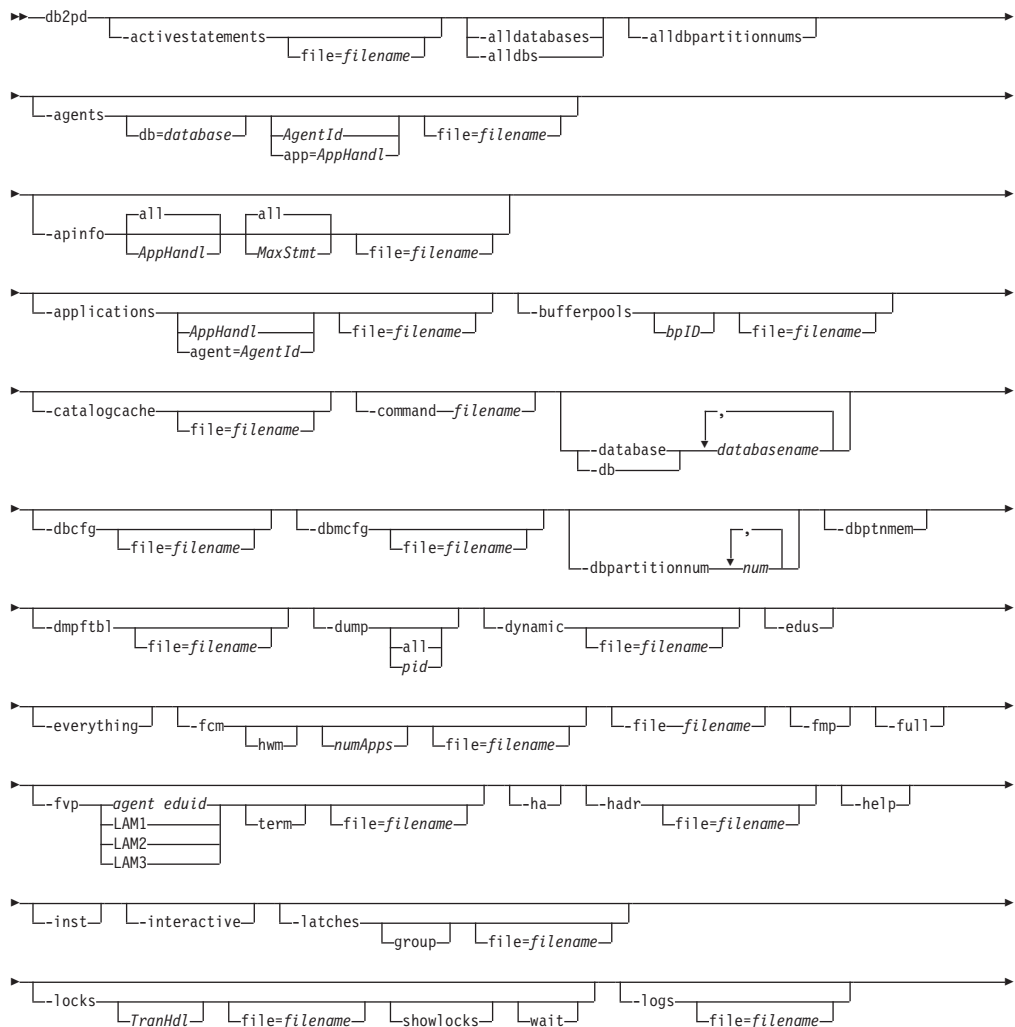
SYSMON 許可レベルが使用される時、以下のオプションは使用不可になります。

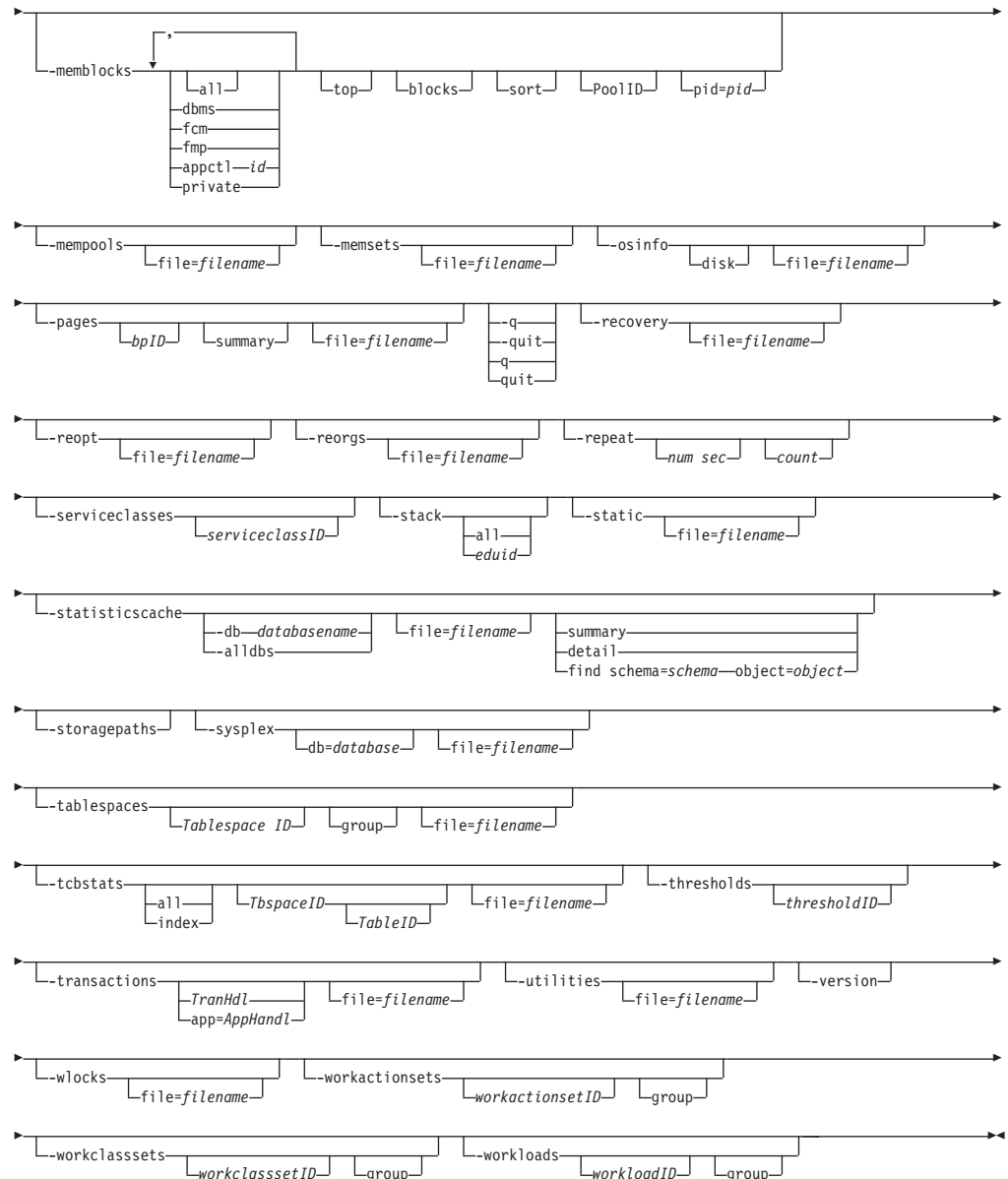
- **dump**
- **stack**
- Windows オペレーティング・システムの場合、SYSADM 権限レベル。

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





コマンド・パラメーター

-activestatements

アクティブ・ステートメント・リストに関する情報を戻します。

-alldatabases | -alldbs

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

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。 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・パーティション・サー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。

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、このオプションに **-inst** オプションを指定します。

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。

DB2 V9.5 フィックスパック 1 で使用可能です。

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・モニターを活動化します。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。

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。例えば、

```
db2pd -catalogcache -db sample
```

db2pd -catalogcache コマンドの出力例を参照してください。

戻り情報の定義については、**-catalogcache** 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 をご覧ください)。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。

-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 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。

-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 num

指定されたデータベース・パーティション・サーバーでコマンドを実行します。

-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。

-dmpftbl

ファイル表の内容をダンプ出力します。

-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。

- **all** 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。
- **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。

-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。

-edus インスタンス内のすべての EDU をリストします。

-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 バッファー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。

-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。

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。

-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。報告される各エレメントについての説明は、「システム・モニター 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あります。

-h | -help

オンライン・ヘルプ情報を表示します。

-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。

-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。

-memblocks

メモリー・プールに関する情報を戻します。

dbms dbms メモリー・セット内のブロックだけを報告します。

fcm 高速コミュニケーション・マネージャーのメモリー・セット内のブロックだけを報告します。

fmp fenced モード・プロシージャのメモリー・セット内のブロックだけを報告します。

appctl id

アプリケーション制御セット内のブロックだけを報告します。

all すべてのメモリー・セットからのブロックを報告します。

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)。

-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。

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。DB2 バージョン 9.5 フィックスパック 1 以降で使用可能です。

-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。

-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、キャッシュに入られている 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。

-reorgs の使用上の注意を参照してください。

-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。

-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。

-stack

エンジン停止の場合には、イベント・スタックとイベント・ヒストリー (イベント・フロー) を使用して DB2 状態に関する情報を得ることができます。イベント・スタックとヒストリーのトレース・ファイルが **diagpath** ディレクトリーに生成されます。トラップ EDU とパニック EDU を除いて、他のすべての EDU はスタック・トレース・ファイル *pid.tid.node.stack.txt* にダンプ出力されます。UNIX オペレーティング・システムでは **-stack**、Windows オペレーティング・システムでは **-stack all** を使用してください。

all 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。

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アタッチされたデータは、db2dump ディレクトリ内の関連する `pid.tid/EDUID.node.trap.txt trap` ファイルにダンプ出力されます。

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 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
- 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)

-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。

-storagepaths

データベースに対して定義された自動ストレージ・パスに関する情報を戻します。

-storagepaths の使用上の注意を参照してください。

-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。

特定の表スペースとそのコンテナに関する情報を表示するには、 *Tablespace ID* コマンド・パラメーターを指定します。

-tablespaces の使用上の注意を参照してください。

-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。特定の表スペースに関する情報を表示するには、このコマンド・パラメーターに *TbpaceID* コマンド・パラメーターを指定します。

特定の表に関する情報を表示するには、このコマンド・パラメーターに *TableID* コマンド・パラメーターを指定します。 *TableID* コマンド・パラメーターを使用するとき、*TbpaceID* コマンド・パラメーターが必須となります。

-tcbstats の使用上の注意を参照してください。

-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。

-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。

-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、「システム・モニター 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。

-v | -version

インストールされた DB2 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。DB2 V9.5 フィックスパック 1 以降で使用可能です。

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作業アクションに関する情報を戻します。

group 作業アクション・セット ID 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 ID でグループ化された同じ情報を返します。

-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 ID でグループ化された同じ情報を返します。

-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
```

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 -all dbs
```

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、上記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: 下記の方法のほかに、代替的な方法として FORCE APPLICATION コマンドを使用できることに注意してください。

- 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.
```

ここでは上記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。

- **-agents**
- **-apinfo**
- **-applications**
- **-bufferpools**
- **-catalogcache**
- **-dbcfg**
- **-dbmcfg**
- **-dynamic**
- **-fcm**
- **-fmp**
- **-hadr**
- **-locks**
- **-logs**
- **-memblocks**
- **-mempools**
- **-memsets**
- **-osinfo**
- **-pages**
- **-recovery**
- **-reopt**
- **-reorgs**
- **-serviceclasses**
- **-static**
- **-statisticscache**
- **-storagepaths**
- **-sysplex**
- **-tablespaces**
- **-tcbstats**
- **-thresholds**
- **-transactions**
- **-wlocks**
- **-workactionsets**
- **-workclasssets**
- **-workloads**

-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。

AppHand1

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。

LkTm0t エージェントのロック・タイムアウト設定。

-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。

AppHandl

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。

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

接続が暗黙的トラステッド接続または明示的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。

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。

ClientAppName

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会計情報ストリング。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。

App1Handl

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。

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・サーバー認証タイプが 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

接続が暗黙的トラステッド接続または明示的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・プールのブロック領域にあるページ数。ブロック・ベースの I/O に関してバッファ・プールを使用できない場合、この値は 0 です。

BlkSize

バッファ・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。ブロック・ベースの I/O に関してバッファ・プールを使用できない場合、この値は 0 です。

NumTbsp

バッファ・プールを使用する表スペースの数。

PgsLeft

バッファ・プールのサイズが減少している場合、そのバッファ・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・プールの現行サイズ。

PostAlter

バッファ・プールが再度開始される時の、ページ内のバッファ・プールのサイズ。

SuspndTSCt

バッファ・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・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。

DatLRds

バッファ・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・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・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・プール内のデータ・ページのヒット率。

TmpDatLRds

バッファ・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・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・プ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・プール内の一時データ・ページのヒット率。

IdxLRds

バッファ・プ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・プール (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・プ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・プール内の索引ページのヒット率。

TmpIdxLRds

バッファ・プ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・プール (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。バッファーク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

バッファーク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

バッファークール非同期読み取り要求。非同期読み取り要求の数。

AsIdxRds

バッファーク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

バッファーク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・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・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・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・ページを書き込んでいることがあります。

AsIdxWrts

バッファ・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・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・ページを書き込んでいることがあります。

AsWrtTime

バッファ・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・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。

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・バッファーのないバッファー・プール。事前選択されたビクティム・バッファー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。

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。

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。

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。

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。

-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。

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。

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。

- **Inactive:** データベース・パーティション・サーバーはデータベース・パーティション化フィーチャー構成で定義されていますが、現在活動状態にありません (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・サーバーはデータベース・パーティション化フィーチャー構成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。

-hadr パラメーター

-hadr パラメーターについては、以下の情報が戻されます。

Role データベースの現在の HADR の役割。指定可能な値のリストについては、**hadr_role** モニター・エレメントを参照してください。

State データベースの現在の HADR の状態。指定可能な値のリストについては、**hadr_state** モニター・エレメントを参照してください。

SyncMode

データベースの現在の HADR 同期モード。指定可能な値のリストについては、**hadr_syncmode** モニター・エレメントを参照してください。

HeartBeatsMissed

HADR 接続上で失われたハートビートの数。データベースが HADR 1 次またはスタンバイの役割の場合、このエレメントは HADR 接続の正常性を示します。詳しくは、**hadr_heartbeat** モニター・エレメントを参照してください。

LogGapRunAvg

1 次ログのシーケンス番号 (LSN) とスタンバイ・ログ LSN の間のギャップの現行の平均。ギャップはバイト数で測定されます。詳しくは、**hadr_log_gap** モニター・エレメントを参照してください。

ConnectStatus

データベースの現在の HADR 接続状況。指定可能な値のリストについては、**hadr_connect_status** モニター・エレメントを参照してください。

ConnectTime

現在の HADR 接続状況が開始された時刻。詳しくは、**hadr_connect_time** モニター・エレメントを参照してください。

Timeout

HADR データベース・サーバーで、そのパートナーからの通信がなく、パートナーとの間の接続が失敗したと見なすまでに要する秒数。詳しくは、**hadr_timeout** モニター・エレメントを参照してください。

LocalHost

ローカル HADR のホスト名または IP アドレス。詳しくは、**hadr_local_host** モニター・エレメントを参照してください。

LocalService

ローカル HADR TCP のサービス名またはポート番号。詳しくは、**hadr_local_service** モニター・エレメントを参照してください。

RemoteHost

リモート HADR のホスト名または IP アドレス。詳しくは、**hadr_remote_host** モニター・エレメントを参照してください。

RemoteService

リモート HADR TCP のサービス名またはポート番号。詳しくは、**hadr_remote_service** モニター・エレメントを参照してください。

RemoteInstance

リモート HADR インスタンス名。詳しくは、**hadr_remote_instance** モニター・エレメントを参照してください。

PrimaryFile

1 次 HADR データベース上の現在のログ・ファイルの名前。詳しくは、**hadr_primary_log_file** モニター・エレメントを参照してください。

PrimaryPg

1 次 HADR データベース上の現在のログの位置を示す現在のログ・ファイルのページ番号。詳しくは、**hadr_primary_log_page** モニター・エレメントを参照してください。

PrimaryLSN

1 次 HADR データベースの現在のログの位置。ログ・シーケンス番号 (LSN) とは、データベースのログ・ストリーム内でのバイト単位のオフセット (相対位置) のことです。詳しくは、**hadr_primary_log_lsn** モニター・エレメントを参照してください。

StandByFile

スタンバイ HADR データベース上の現在のログ・ファイルの名前。詳しくは、**hadr_standby_log_file** モニター・エレメントを参照してください。

StandByPg

スタンバイ HADR データベース上の現在のログの位置を示す現在のログ・ファイルのページ番号。詳しくは、**hadr_standby_log_page** モニター・エレメントを参照してください。

StandByLSN

スタンバイ HADR データベースの現在のログの位置。詳しくは、**hadr_standby_log_lsn** モニター・エレメントを参照してください。

StandByRcvBufUsed

スタンバイの使用されているログ受信バッファのパーセント。このプロパティは、HADR スタンバイ・データベースに関してのみ報告されます。高い値は、HADR スタンバイ・データベースが十分な速さでログを処理していないことを示す可能性があります。この値が頻繁に 100 に達する場合は、1 次データベース上のパフォーマンスに悪影響を与えるおそれがあります。反対に、この値が常に低い場合は、スタンバイのログ受信バッファが十分に使用されていません。この場合、バッファ・サイズを減らすことを検討してください。スタンバイのログ受信バッファ・サイズの調整については、**DB2_HADR_BUF_SIZE** レジストリー変数を参照してください。

-locks パラメーター

-locks パラメーターについては、以下の情報が戻されます。

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
- NX
- W
- 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ター。

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。

-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。

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。

Pages Written

現在のログに書き込まれている現行ページ。

StartLSN

開始ログ・シーケンス番号。

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。

-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:

Poo1ID メモリー・ブロックを所有するメモリー・プール ID。

Poo1Name

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。

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。

Overhead プール構造に必要な内部オーバーヘッド。

LogSz プール・メモリー要求の現在の合計。

LogUpBnd 現在の論理サイズの上限。

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。

PhyUpBnd 現在の物理サイズの上限。

PhyHWM 処理中の最大物理サイズ。

Bnd 内部バインド・ストラテジー。

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。

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。

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。

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。

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。

-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 の数。

ConfigCPU

構成された CPU の数。

Speed (MHz)

CPU の速度 (MHz)。

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ー増分の周波数 (Hz)。これは、Linux PPC でのみサポートされます。

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。

MsgTql 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。

SemMsl 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。Derived from semume.
- 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

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。可能な値は、Data、Index、LongField、XMLData、SMP、LOB、LOBA、および MDC_BMP 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。

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。

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。

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。
NulID	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。

表 Reorg 情報:

TabSpaceID	表スペース ID。
TableID	表 ID。
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。
TableName	表の名前。
Type	再編成のタイプ。可能な値は次のとおりです。 <ul style="list-style-type: none"> • Online • Offline
IndexID	表の再編成に使用される索引の ID。
TempSpaceID	表が再編成される表スペース。

表 Reorg 統計:

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。

-serviceclasses parameter

-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: サービス・クラスの状態 (有効または無効)
- Agent Priority: サービス・クラスのエージェント優先度の設定
- Prefetch Priority: サービス・クラスのプリフェッチ優先度の設定
- Outbound Correlator: サービス・クラスのアウトバウンド相関関係子の設定

- 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
- 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 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
- Num Activities: サービス・サブクラス内の現在のアクティビティの数
- Activities HWM: 最後に統計がリセットされた以降の、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
- 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

- 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。

-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。

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。

QOpt 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)。

-statisticscache コマンド・パラメーターを使用した場合に戻される情報の詳細については、データベース・パフォーマンスのチューニングにある『カタログ統計の表』のトピックを参照してください。

-storagepaths パラメーター

-storagepaths パラメーターについては、以下の情報が戻されます。

Number of Storage Paths
データベースに対して定義された自動ストレージ・パスの数。

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。

-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クラスタの 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クラスタの 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クラスタ 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)。

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。

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
- 0x00000800 - BACKUP IN PROGRESS
- 0x00001000 - STORAGE MUST BE DEFINED
- 0x00002000 - RESTORE IN PROGRESS
- 0x00004000 - OFFLINE
- 0x00008000 - DROP PENDING
- 0x00010000 - WRITE SUSPENDED
- 0x00020000 - LOAD IN PROGRESS
- 0x00020000 - STORAGE MAY BE DEFINED
- 0x00040000 - STORAGE DEFINITION IS IN FINAL STATE
- 0x00080000 - STORAGE DEFINITION CHANGED PRIOR TO ROLLFORWARD
- 0x00100000 - DMS REBALANCER IS ACTIVE
- 0x00200000 - DELETION IN PROGRESS
- 0x00400000 - 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。

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

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時のサイズ増加単位 (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・コンテナー:

TspId コンテナーを所有する表スペースの ID。

ContainNum

表スペース内のコンテナーに割り当てられた番号。

Type

コンテナーのタイプ。可能な値は次のとおりです。

- Path
- Disk
- File
- Striped Disk
- Striped File

TotalPgs

コンテナー内のページ数。

UsablePgs

コンテナー内で使用できるページの数。

StripeSet

コンテナー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

コンテナーの名前。

-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:

TbSpaceID

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。

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。

Scans 表に対して実行されたスキャンの数。

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。

0vFlReads

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。

0vFlCrtes

作成された新しいオーバーフローの数。

Note 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。

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。

TableName

表の名前。

SchemaNm

表の名前を修飾するスキーマ。

IID 索引 ID。

IndexObjSize

索引オブジェクト内のページ数。

TCB 索引統計:**TableName**

表の名前。

IID 索引 ID。

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。

Scans 索引に対するスキャンの数。

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。

IntNodSpl

中間レベル分割の数。

-thresholds

-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

- Domain ID: しきい値ドメイン ID
- Predicate ID: しきい値述部 ID
- Maximum Value: しきい値の最大値
- Enforcement: しきい値の施行有効範囲
- Queuing: しきい値がキューイングしきい値である
- Queue Size: しきい値キュー・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: しきい値アクション設定
- Enabled: しきい値の状態。enabled (有効) または disabled (無効)

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 (しきい値のキュー情報): Threshold Name (しきい値名)
- Max Concurrency: 最大の並行性の設定
- Concurrency: 実際の並行性の値
- Max Queue Size: 最大のしきい値キュー・サイズ設定

-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。

AppHandl

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。
- 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されました。

Tflag2 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。

LogSpace

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。

SpaceReserved

トランザクションに予約されているログ・スペースの合計 (使用済みスペースおよびすべての適合レコードを含む)。

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_wkstn** (TP Monitor Client Workstation Name モニター・エレメント) と同じです。

ClientAppName

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。

ClientAccntng

トランザクションを実行しているクライアントの会計情報ストリング。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。

-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。

AppHndl

アプリケーション・ハンドル (ノードと索引を含む)。

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。

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
- NX
- W
- 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 **appl_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
 - クラスの種類
 - スキーマ名
 - FROM 値
 - TO 値
 - 範囲の単位

-workloads パラメーター

-workloads パラメーターについては、以下の情報が戻されます。

- ワークロード定義リスト。
- ワークロード権限情報。
- 各ワークロードの並行ワークロード・オカレンス数を示すカウンター。

出力例

-apinfo

以下は、db2pd -apinfo コマンドの出力例です。

```
db2pd -apinfo 2836 -db pdtest
```

```
Database Partition 0 -- Database PDTEST -- Active -- Up 0 days 00:02:46
```

```
Application :
```

```
Address : 0x0000000200D07080
AppHndl [nod-index] : 2836 [000-02836]
TranHdl : 2
Application PID : 6176
Application Node Name : hotel11
IP Address: n/a
Connection Start Time : (1220565478)Thu Sep 4 17:57:58 2008
Client User ID : juntang
System Auth ID : JUNTANG
Coordinator EDU ID : 32
Coordinator Partition : 0
Number of Agents : 1
Locks timeout value : NotSet
Locks Escalation : No
Workload ID : 1
Workload Occurrence ID : 1
Trusted Context : n/a
Connection Trust Type : non trusted
Role Inherited : n/a
Application Status : UOW-Waiting
Application Name : db2bp
Application ID : *LOCAL.juntang.080904215757
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 : 4
Activity ID : 3
Package Schema : NULLID
Package Name : SQLC2G14
Package Version :
Section Number : 201
```

```

SQL Type :      Dynamic
Isolation :     CS
Statement Type : DML, Select (blockable)
Statement :     select * from t2

UOW-ID :       4
Activity ID :   2
Package Schema : NULLID
Package Name :  SQLC2G14
Package Version :
Section Number : 203
SQL Type :     Dynamic
Isolation :    CS
Statement Type : DDL, (not Set Constraints)
Statement :    create table t2 ( c1 int )

UOW-ID :       4
Activity ID :   1
Package Schema : NULLID
Package Name :  SQLC2G14
Package Version :
Section Number : 203
SQL Type :     Dynamic
Isolation :    CS
Statement Type : DML, Insert/Update/Delete
Statement :    insert into t1 values 1

```

-catalogcache

以下は、db2pd -catalogcache コマンドの SYSTABLES 出力例です。

```

Database Partition 0 -- Database SAMPLE -- Active -- Up 0 days 00:05:34

Catalog Cache:
Configured Size      1064960
Current Size         78272
Maximum Size         4294901760
High Water Mark      131072

SYSTABLES:
Address              Schema Name      Type TableID TbspaceID LastRefID CatalogCacheLoadingLock CatalogCacheUsageLock Sts
0x07800000232FF820 SYSIBM SYSTABLES T 5 0 19288214 0001000007800000232FF82043 0000000500001804232FF82043 V
0x07800000232FD360 SYSCAT TABLES V 0 0 19288214 0001000007800000232FD36043 000000050000CC907232FD36043 V
0x07800000232FFB60 KEON014 EMPLOYEE 0 0 0 19288214 0001000007800000232FFB6043 000000050013AE07232FFB6043 I
0x07800000232FC500 SYSTOOLS POLICY 0 0 0 19288214 0001000007800000232FC50043 00000000000000000000000000000000 I
0x07800000232FCF40 KEON014 DEPT T 4 2 19288214 0001000007800000232FCF4043 000000050013AE06232FCF0343 V
0x07800000238FCF40 KEON014 DEPT T 4 2 19288214 0001000007800000238FCF4043 000000050013AE06238FCF0143 S
0x07800000234433A0 KEON014 SALARY 0 0 0 19288214 0001000007800000234433A043 000000050013AF00234433A043 I

```

-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:
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

```

-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      = TOFU
Service Class ID       = 24
Service Class Type     = Service Superclass
Default Subclass ID   = 25
Service Class State    = Enabled
Agent Priority         = Default
Prefetch Priority     = Default
Outbound Correlator   = None
Work Action Set ID    = N/A
Collect Activity Opt  = None

```

```

Num Connections          = 1
Last Statistics Reset Time = 01/26/2007 07:48:40.000000
Num Coordinator Connections = 1
Coordinator Connections HWM = 1

```

```

Associated Workload Occurrences (WLO):
AppHandl [nod-index] WL ID WLO ID UOW ID WLO State
25 [000-00025] 6 1 3 undefined
24 [000-00024] 1 2 2 undefined
7 [000-00007] 1 3 2 undefined

```

サービス・サブクラスの出力例

```

Service Class Name      = SUBTOFU
Service Class ID       = 26
Service Class Type     = Service Subclass
Parent Superclass ID   = 24
Service Class State    = Enabled
Agent Priority         = -10
Prefetch Priority     = High
Outbound Correlator   = None
Collect Activity Opt  = None
Collect Aggr Activity Opt = Base
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

Num Activities                            = 0
Last Stats Reset Time                     = 12/31/1969 19:00:00.000000
Activities HWM                             = 1
Activities Completed                       = 1
Activities Rejected                       = 0
Activities Aborted                         = 0

Associated Agents:
EDU ID      AppHandl [nod-index]  WL ID  WLO ID  UOW ID  Activity ID
17179869216 25      [000-00025]  6      1      3      1
17172839348 24      [000-00024]  6      2      2      1

Associated Non-agent threads:
PID          TID          Thread Name
11834        183320439136  db2fmp
11834        183316244832  db2fmp

```

-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      0002000B00000000340000452 Row  ..X  G   352614  db2bp   VENUS  *LOCAL.venus.071117030309
15      [000-00015]  9      0002000B00000000340000452 Row  .NS  W   1176046  db2bp   VENUS  *LOCAL.venus.071117030358
12      [000-00012]  2      0002000B00000000340000452 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

```

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。

- Linux および UNIX の場合、*sysadm*、*sysmaint*、*sysctrl*、または *sysmon* 権限レベル。

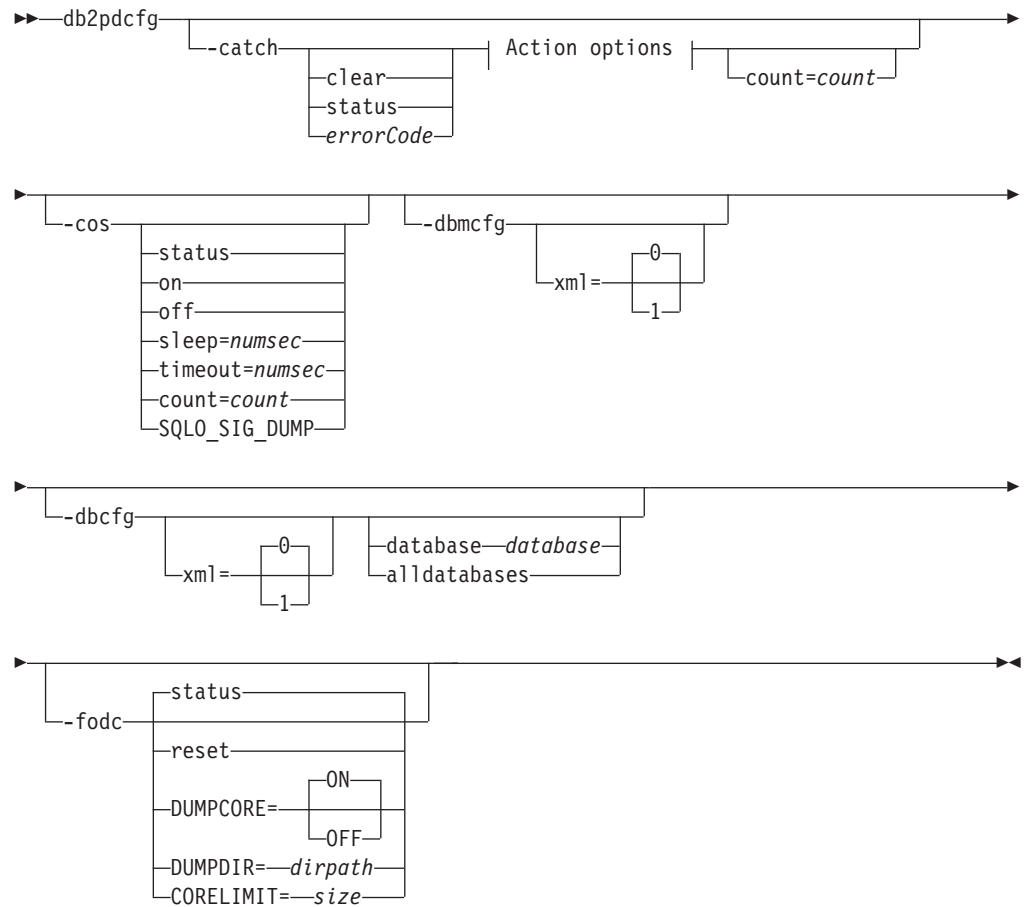
sysmon 許可レベルが使用される場合は、*-status* サブオプションだけが使用可能になります。

- Windowsオペレーティング・システムの場合、*sysadm* 権限レベル。

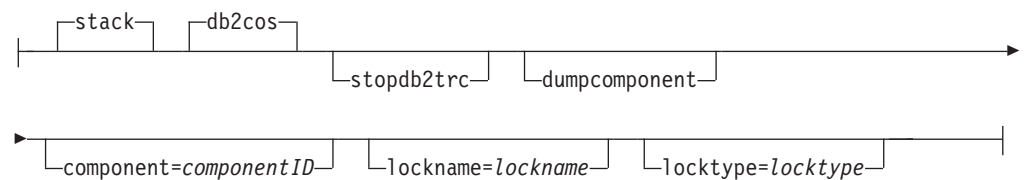
必要な接続

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



Action options:



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。

- `sqlCode[,reasonCode] / sqlCode=sqlCode[,reasonCode]`

- ZRC (16 進数または整数)
- ZRC #define (SQLP_LTIMEOUT など)
- ECF (16 進数または整数)
- "deadlock" または "locktimeout"

stack db2diag.log 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。

dumpcomponent

コンポーネント・フラグをダンプします。

component=componentID

コンポーネント ID。

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中に 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に db2cos へのデータベース・マネージャー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に db2cos へのデータベース・マネージャー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のサイズを検査する間にスリープする時間の長さ。デフォルトは 3 秒です。

timeout=numsec

db2cos スクリプトの完了が想定されるまで待機する時間の長さ。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。最初の出現箇所のデータ収集 (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。

DUMPCORE=

コア・ファイルの生成を有効または無効にします。

ON すべてのプラットフォームでデフォルトです。

OFF

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その他の大規模なバイナリー・ダンプにも使用できます。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。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。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。

CORELIMIT が設定されない場合、DB2 は現在の **ulimit** 設定と同じ値にコア・ファイル・サイズを設定します。AIX は例外で、コアの **ulimit** 設定値 (無制限) をオーバーライドするために 8 GB の値が使用されます。

db2pdcfg を使って **CORELIMIT** を変更する場合には、通常の UNIX アクセス許可の制約を受け、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。

AIX では、「無制限」のユーザー・コア制限は DB2 サーバー・プロセスのみに関する値 8 GB を使ってオーバーライドされます。これは、DPF 環境、または内部並列処理が使用可能な環境において大きな高速コミュニケーション・マネージャー (FCM) メモリー領域がダンプ出力されるのを防ぐためです。AIX では、(問題診断でほとんど必要とされない) このメモリー領域の大部分もまた、事前割り振り/非コミットされます (システム・メモリーによって支えられるものではありません)。この事前割り振りにより、FCM の動的メモリー要件がサポートされますが、このメモリーをコア・ファイルにダンプ出力すると、そのメモリーが即時にコミットされ、多くの場合、不必要に大きなコア・ファイルになります。8 GB より大きいコア・ダンプが必要な場合、unlimited 以外の何らかの大きな値 (例えば RAM のサイズ) にユーザー・コア制限を設定する必要があります。あるいは、CORELIMIT を十分に大きな値に設定することもできます。ユーザー・コア制限または CORELIMIT の変更内容は、DB2 インスタンスの次の再生まで有効にならないことに注意してください。

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 オプションでは、一部の設定は variable=value の形式で指定されます。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
```

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」で使用されます。

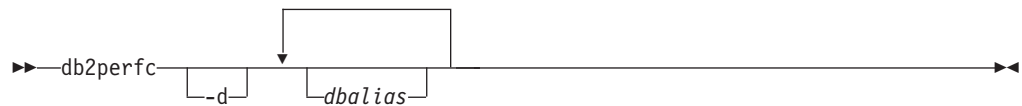
許可

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。

```
db2perf
```

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。

```
db2perf 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。

```
db2perf -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。

```
db2perf -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。しかしたいの場合、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。

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ー

Windows オペレーティング・システムに DB2 パフォーマンス・カウンターを追加します。これは、DB2 および DB2 Connect のパフォーマンス情報を、Windows パフォーマンス・モニターにアクセス可能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、以下を行います。

- 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。
- 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

Windows オペレーティング・システム上のパフォーマンス・モニターで使用されます。db2perfr コマンドは、パフォーマンス・カウンターにアクセスするときに、管理者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されます。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

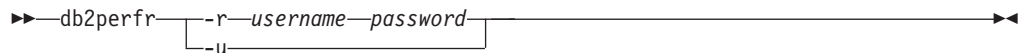
ローカル管理者

必要な接続

なし

コマンド構文

```
db2perfr -r username password
```



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 に登録したら、パフォーマンス・モニターのローカル・インスタンスも、そのユーザー名とパスワードを使用して明示的にログオンします。これは、DB2 に登録したユーザー名情報が一致しない場合、パフォーマンス・モニターのローカル・セッションは DB2 パフォーマンス情報を示さないことを意味します。
- 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・ユーザー名 SYSTEM は、DB2 予約語なので使用できません。

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。

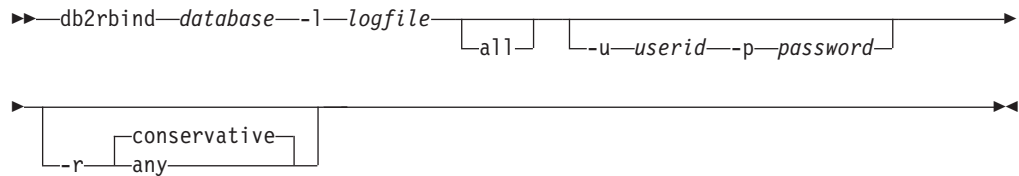
許可

sysadm

必要な接続

なし

コマンド構文



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シージャからのエラーを記録するとき使用するパス (任意指定) とファイル名 (必須) を指定します。

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す。

-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。

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。

許可

なし

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
...
```

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

これは、自動ストレージが有効になっているデータベースにのみ該当します。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。

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
```


構成ファイルを `relocate.cfg` として保管し、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、以下のようにします。

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。

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
NODE_NUM=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
NODE_NUM=20
LOG_DIR=/testdb_logdir,/servdb_logdir
```

例 6

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
NODE_NUM=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 `sqlbdbir` 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。

インスタンスが変更されている場合、ツール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。

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 ▶▶
    |--file--log_control_file--
    |--path--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。

-file *log_control_file*

ログ制御ファイルのパスとファイル名を指定します。

-path *log_control_files_dir*

ログ制御ファイル `SQLLOGCTL.LFH.1` とそのミラー・コピー `SQLLOGCTL.LFH.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。

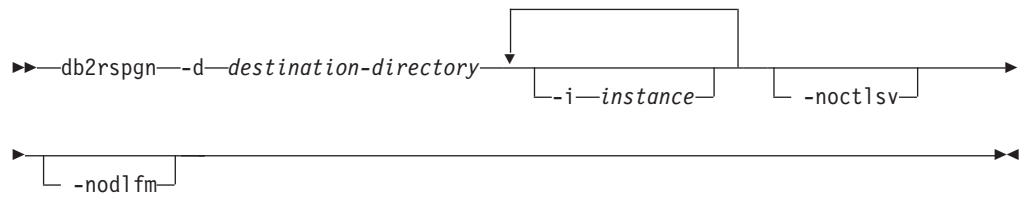
使用上の注意

`-file` パラメーターを使用した場合、指定されたログ制御ファイル (例えば `SQLLOGCTL.LFH.1` と `SQLLOGCTL.LFH.2` のどちらか) だけが更新されます。これにより、2 つのファイルは同期状態ではなくなります。このため、可能な限り、`-file` オプションではなく `database_alias` または `-path` オプションを使用することをお勧めします。

db2rspgn - 応答ファイル生成プログラム (Windows)

db2rspgn コマンドは、Windows 上でのみ使用できます。

コマンド構文



コマンド・パラメーター

-d *destination-directory*

応答ファイルとすべてのインスタンス・ファイルの宛先ディレクトリー。このパラメーターは必須です。

-i *instance*

プロファイルを作成するインスタンスのリスト。デフォルトでは、すべてのインスタンスのインスタンス・プロファイル・ファイルが生成されます。このパラメーターはオプションです。

-noctlsv

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・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。

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。

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

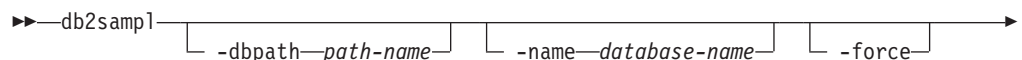
以下のいずれか。

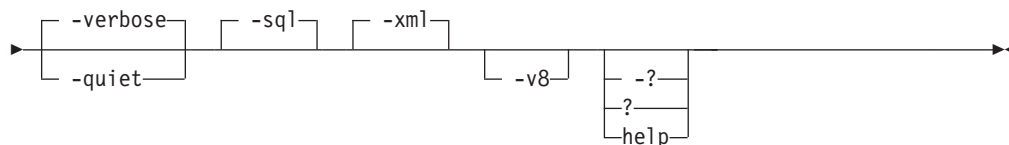
- *sysadm*
- *sysctrl*

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・マネージャー構成ファイル (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。

-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・マネージャー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・セットと UCA400_NO 照合および C (POSIX) テリトリリーを使用して 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・セットと UCA400_NO 照合および C (POSIX) テリトリリーを使用して 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。
- db2sampl コマンドを使用して、XML データベース・オブジェクトを使用するサンプル・データベースを作成すると、DB2 Enterprise Server Edition で利用できるデータベース・パーティション・フィーチャーを将来利用できなくなります。DB2 Enterprise Server Edition がインストールされている環境で db2sampl コマンドを発行すると、警告テキストが標準出力に送信されます。
- 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。
- デフォルトでは、db2sampl コマンドは Enterprise Server Edition インスタンスを Windows 上で作成します。この場合、XML 表はデフォルトでは作成されません。XML 表を Windows 上で作成するには、-xml オプションを使用する必要があります。

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。

```
db2samp1 -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。

```
db2samp1 -v8
```

db2schex - Active Directory スキーマ拡張コマンド

Lightweight Directory Access Protocol (LDAP) ディレクトリー・サーバー・フィーチャーを Windows Server 2003 以降で使用するために、Microsoft Active Directory スキーマを拡張して、必要な DB2 オブジェクト・クラスと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してください。そうしないと、ノードを手動で登録して、データベースをカタログすることが必要になります。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)』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 NT[®] ユーザー名として指定できます。

db2schex 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 プロファイル変数を表示、設定、または除去します。DB2 プロファイル・レジストリーには、DB2 Administration Server を経由しての、DB2 の環境変数のローカル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・コマンドが保管されています。

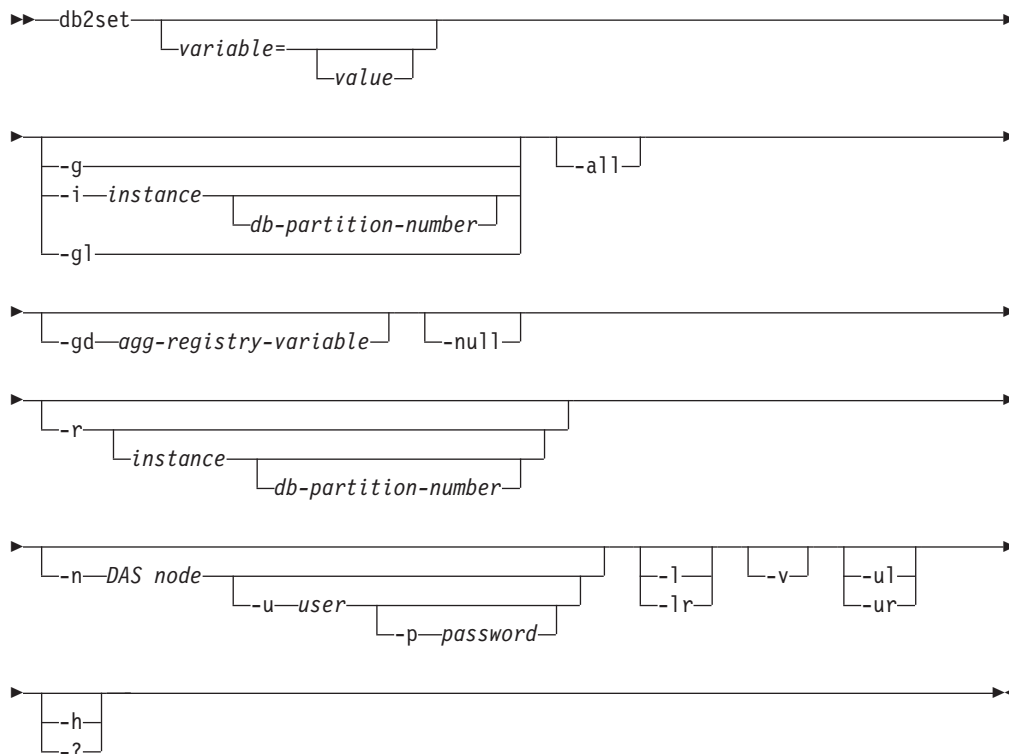
許可

```
sysadm
```

必要な接続

なし

コマンド構文



コマンド・パラメーター

variable=

指定された変数の値を表示します。

value 指定された変数を入力値に設定します。変数を削除する場合は、等号 (=) の後の指定の変数に値を指定しないでください。設定の変更は、インスタンスが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。

-i インスタンス・プロファイルを指定して、現行値またはデフォルトの代わりに使用します。

db-partition-number

db2nodes.cfg ファイルでリストされる番号を指定します。

-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。このオプションが有効なのは、レジストリー変数 DB2_ENABLE_LDAP が YES に設定されている場合だけです。

-all 以下の形式で定義されたローカル環境変数のオカレンスをすべて表示します。

- 環境 ([e] で表示)

- ノード・レベル・レジストリー ([n] で表示)
- インスタンス・レベル・レジストリー ([i] で表示)
- グローバル・レベル・レジストリー ([g] で表示)

-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・レベルの変数値を NULL 値に設定します。これによって探索順で定義された次のレジストリー・レベルの値を探索せずに済みます。

-r *instance*

指定した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。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、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しない場合、アタッチが存在しなければ、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。

-n *DAS node*

リモート DB2 Administration Server の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 現在の DB2 製品インストール済み環境のすべてのインスタンス・プロファイルの一覧を示します。

-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 冗長モードを指定します。

-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

- 特定のインストール済み環境に属する定義済みのプロファイル (DB2 インスタンス) 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のすべての定義済みの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 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 DB2COMM
```

- ノード 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 およびパスワードに 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 を設定解除するには、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に対してグローバルに、変数 DB2COMM を TCPIP に設定するには、次のように入力します。

```
db2set -g DB2COMM=TCPIP
```

- 変数 DB2COMM を、インスタンス MYINST 用の TCPIP のみで設定するには、次のように入力します。

```
db2set -i MYINST DB2COMM=TCPIP
```

- 変数 DB2COMM を指定したインスタンス・レベルで NULL 値に設定するには、次のように入力します。

```
db2set -null DB2COMM
```

使用上の注意

変数名を指定しないと、定義済み変数の値がすべて表示されます。変数名を 1 つだけ指定した場合、その変数の値だけが表示されます。変数の値をすべて表示するには、`variable -all` を指定します。すべてのレジストリーに定義されている変数をすべて表示するには、`-all` を指定します。

変数の値を修正するには、`variable=` の後に新規の値を指定します。変数の値に NULL を指定するには、`variable -null` を指定します。設定の変更は、インスタンスが再始動した後に有効になります。

変数を削除するには、値を指定せずに `variable=` だけにします。

このコマンドは DB2 の非ルート・インストールの場合と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パラメーターなど)。

db2setup - DB2 のインストール

DB2 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 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 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

Linux および UNIX システムでは、ルート・インストールには `root` 権限が必要です。非ルート・インストールの場合、非ルート・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] [--?] [-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。ルート・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。非ルート・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` は非ルート・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 Base Component (SA MP Base Component) をインストールする場合、SA MP Base Component のインストール・ログ・ファイルは、DB2 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。

`-? | -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。

db2sqljbind - SQLJ プロファイル・バインダー

db2sqljbind は、以前に db2sqljcustomize コマンドでカスタマイズされたシリアライズド・プロファイルの DB2 パッケージをバインドします。

IBM Data Server Driver for JDBC and SQLJ と共に実行するアプリケーションは、パッケージを必要としますが、プランは必要としません。db2sqljcustomize -automaticbind オプションが YES に指定されているか、デフォルトで YES になっている場合、db2sqljcustomize は -url パラメーターで指定されたデータ・ソースでパッケージのバインドを行います。しかし、-automaticbind が NO になっている場合、db2sqljcustomize の実行時にバインドが失敗した場合、あるいは、複数のロケーションで同じシリアライズド・プロファイルのための同一のパッケージを作成したい場合は、db2sqljbind コマンドを使用してパッケージをバインドできます。

許可

プロセスの特権セットには、以下の権限のうちの 1 つを組み込む必要があります。

- SYSADM authority
- DBADM 権限
- パッケージが存在しない場合の BINDADD 特権と、以下のいずれかの特権
 - CREATEIN 特権
 - 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
- パッケージが存在する場合に必要な特権:
 - スキーマに対する ALTERIN 特権
 - パッケージに対する BIND 特権

ユーザーには、アプリケーションで静的 SQL ステートメントをコンパイルするのに必要なすべての特権も必要です。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。ユーザーが、SYSADM 権限は持っていない場合、DB2 データベース・マネージャーによって明示的な DBADM 権限が自動的に認可されません。

コマンド構文

```
▶▶ db2sqljbind [-help] -url jdbc:db2://server[:port]/database[:property=value;]
```

▶ --user *user-ID* --password *password* [-bindoptions "*options-string*"]

▶ [-staticpositioned NO]
 [-staticpositioned YES] [-genDBRM] [-DBRMDir *directory-name*]

▶ [-tracefile *file-name*] [-tracelevel TRACE_SQLJ]

[-tracelevel]

- TRACE_NONE
- TRACE_CONNECTION_CALLS
- TRACE_STATEMENT_CALLS
- TRACE_RESULT_SET_CALLS
- TRACE_DRIVER_CONFIGURATION
- TRACE_CONNECTS
- TRACE_DRDA_FLOWS
- TRACE_RESULT_SET_META_DATA
- TRACE_PARAMETER_META_DATA
- TRACE_DIAGNOSTICS
- TRACE_SQLJ
- TRACE_XA_CALLS
- TRACE_TRACEPOINTS
- TRACE_ALL

▶ [-serialized-profile-name]

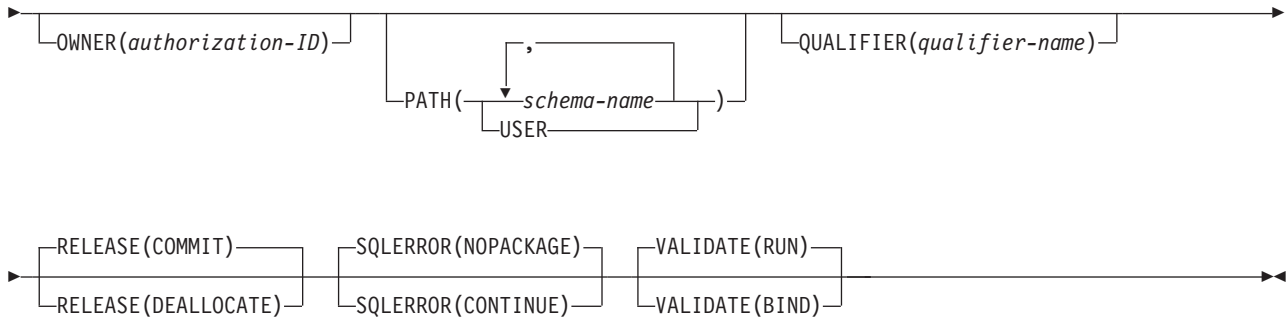
options-string:

▶▶ [-DB2-for-z/OS-options]
 [-DB2-Database-for-Linux-UNIX-and-Windows-options]

DB2 for z/OS options:

▶▶ (1) [ACTION(REPLACE)] [REPLVER(*version-id*)] [DBPROTOCOL(DRDA)] [DEGREE(1)]
 [ACTION(ADD)] [DBPROTOCOL(PRIVATE)] [DEGREE(ANY)]

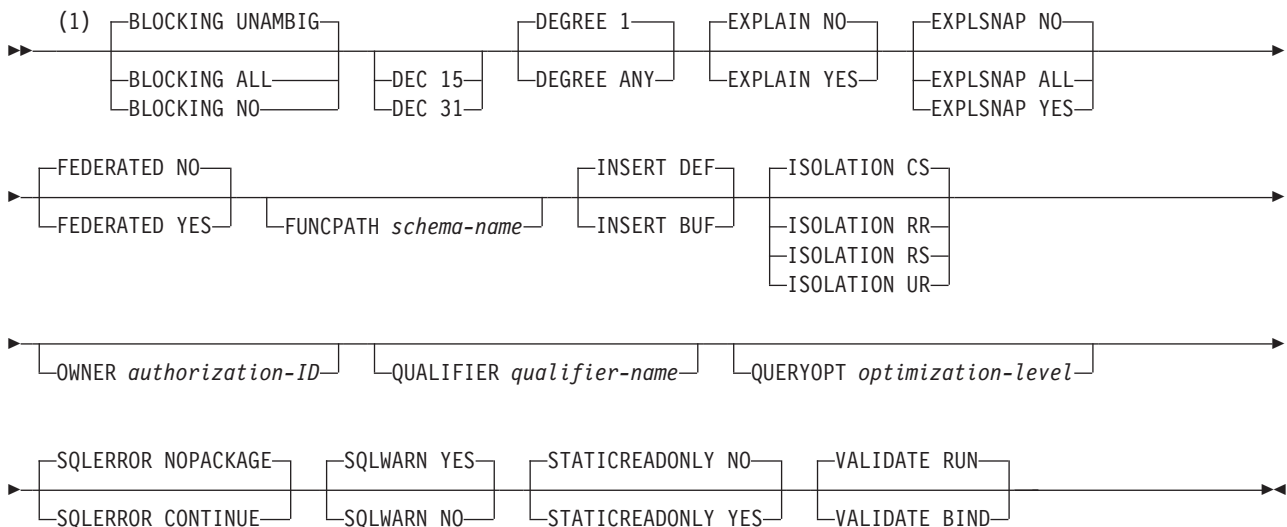
▶ [EXPLAIN(NO)] [IMMEDIATEWRITE(NO)] [ISOLATION(RR)] [NOREOPT(VARS)]
 [EXPLAIN(YES)] [IMMEDIATEWRITE(PH1)] [ISOLATION(RS)] [REOPT(VARS)] [OPTHINT(*hint-ID*)]
 [IMMEDIATEWRITE(YES)] [ISOLATION(CS)]
 [ISOLATION(UR)]



注:

- 1 これらのオプションは、どのような順序で指定してもかまいません。

DB2 Database for Linux, UNIX, and Windows options



注:

- 1 これらのオプションは、どのような順序で指定してもかまいません。

コマンド・パラメーター

-help

db2sqljbind はそれがサポートする各オプションについて記述することを指定します。 -help と共に他のオプションが指定されても、それらは無視されます。

-url

プロファイルがカスタマイズされるデータ・ソースの URL を指定します。 -automaticbind オプションまたは -onlinecheck オプションが YES に指定されているか、デフォルトで YES になった場合、この URL が表すデータ・ソースへの接続が確立されます。 -url 値の可変部は、以下のとおりです。

server

データベース・サーバーが存在するオペレーティング・システムのドメイン名または IP アドレス。

port

データベース・サーバーに割り当てられている TCP/IP サーバーのポート番号。デフォルトは 446 です。

database

プロファイルがカスタマイズされるデータベース・サーバーの名前。

接続を DB2 for z/OS サーバーに対して行う場合、*database* は、インストール中に定義される DB2 ロケーション名です。この値の中の文字はすべて大文字でなければなりません。サーバーで以下の SQL ステートメントを実行することにより、ロケーション名を判別できます。

```
SELECT CURRENT SERVER FROM SYSIBM.SYSDUMMY1;
```

接続を DB2 Database for Linux, UNIX, and Windows サーバーに対して行う場合、*database* は、インストール中に定義されるデータベース名です。

接続を IBM Cloudscape® サーバーに対して行う場合、*database* は、データベースを含むファイルの完全修飾名です。この名前は、二重引用符 (") で囲まなければなりません。以下に例を示します。

```
"c:/databases/testdb"
```

```
property=value;
```

JDBC 接続のプロパティ。

-user user-ID

パッケージのバインドのためにデータ・ソースに接続するときに使うユーザー ID を指定します。

-password password

パッケージのバインドのためにデータ・ソースに接続するときに使うパスワードを指定します。

-bindoptions options-string

スペースで区切ったオプションのリストを指定します。それらのオプションには、同名の DB2 プリコンパイル・オプションおよびバインド・オプションと同じ機能があります。DB2 for z/OS システム上で実行されるプログラムを準備する場合、DB2 for z/OS のオプションを指定します。DB2 Database for Linux, UNIX, and Windows システム上で実行されるプログラムを準備する場合、DB2 Database for Linux, UNIX, and Windows のオプションを指定します。

バインド・オプションに関する注:

- 以下の条件が該当する場合にのみ **VERSION** を指定してください。
 - パッケージを DB2 Database for Linux, UNIX, and Windows システムでバインドしている場合、システムがバージョン 8 以降である。
 - 関連したパッケージを新規の **VERSION** 値にバインドする前にプログラム上で変換プログラムを再実行する。
- **STATICREADONLY** をサポートするサーバーの場合は、**STATICREADONLY** の値は **YES**、その他のサーバーの場合は、**NO** となります。
STATICREADONLY YES を指定すると、DB2 は未確定カーソルを読み取り専用カーソルであるものとして処理します。イテレーター宣言のエラーのトラブルシューティングの場合、**STATICREADONLY NO** を明示的に指定する必要があります。あるいは、確定済みになるようにイテレーターを宣言しま

す。例えば、イテレーターを確定的に更新可能にする場合は、`sqlj.runtime.ForUpdate` をインプリメントするようにイテレーターを宣言します。イテレーターを読み取り専用にする場合、イテレーターを使用する `SELECT` ステートメントに `FOR READ ONLY` 節を付け加えます。

重要: パッケージのバインディングを行う場所であるデータ・ソースに適したプログラム準備オプションのみを指定してください。IBM Data Server Driver for JDBC and SQLJ の一部の値およびデフォルト値は、DB2 用の値およびデフォルト値とは異なります。

-staticpositioned NOIYES

イテレーターを使用する位置指定 `UPDATE` ステートメントと同じソース・ファイルに宣言されているイテレーターの場合、静的にバインドされたステートメントとしてその位置指定 `UPDATE` を実行するかどうかを指定します。デフォルトは `NO` です。`NO` は、動的に準備されるステートメントとして位置指定 `UPDATE` を実行することを意味します。この値は、シリアライズド・プロファイルの以前の `db2sqljcustomize` 呼び出しのための `-staticpositioned` 値と同じでなければなりません。

-genDBRM

`db2sqljbind` がシリアライズド・プロファイルからデータベース要求モジュール (DBRM) を生成すること、および `db2sqljbind` がリモート・バインド操作を行わないことを指定します。

`-genDBRM` は、DB2 for z/OS データベース・サーバーで実行されるプログラムにのみ適用されます。

-DBRMDir *directory-name*

`-genDBRM` を指定する場合、`-DBRMDir` は生成された DBRM ファイルを `db2sqljbind` が配置するローカル・ディレクトリーを指定します。デフォルトは、現行ディレクトリーです。

`-DBRMdir` は、DB2 for z/OS データベース・サーバーで実行されるプログラムにのみ適用されます。

-tracefile *file-name*

トレースを使用可能にし、トレース情報のための出力ファイルを示します。このオプションを指定できるのは、IBM ソフトウェア・サポートから指示された場合のみです。

-tracelevel

`-tracefile` を指定する場合、`db2sqljcustomize` の実行中に何をトレースするかを指示します。デフォルトは `TRACE_SQLJ` です。このオプションを指定できるのは、IBM ソフトウェア・サポートから指示された場合のみです。

serialized-profile-name

パッケージのバインド元である 1 つ以上のシリアライズド・プロファイルの名前を指定します。シリアライズド・プロファイル名の形式は、以下のようになります。

`program-name_SJProfileIDNumber.ser`

program-name は、拡張子 *.sqlj* の付いていない SQLJ ソース・プログラムの名前です。*n* は、0 から *m-1* までの整数です。ただし、*m* は、SQLJ ソース・プログラムから SQLJ 変換プログラムが生成したシリアライズド・プロファイルの数です。

単一の DB2 パッケージを複数のシリアライズド・プロファイルからバインドするために複数のシリアライズド・プロファイル名を指定する場合、`db2sqljcustomize` を実行したときに、同じシリアライズド・プロファイル名を同じ順序で指定している必要があります。

例

```
db2sqljbind -user richler -password mordecai
            -url jdbc:db2://server:50000/sample -bindoptions "EXPLAIN YES"
            pgmname_SJProfile0.ser
```

使用上の注意

db2sqljbind によって作成されるパッケージ名: `db2sqljbind` によって作成されるパッケージの名前は、`db2sqljcustomize` を実行したときに `-rootpkgname` または `-singlepkgname` パラメーターを使用して指定された名前です。 `-rootpkgname` または `-singlepkgname` を指定しなかった場合、パッケージ名は分離レベル文字を付加したプロファイル名の最初の 7 バイトです。

db2sqljbind の **DYNAMICRULES** 値: **DYNAMICRULES** バインド・オプションによって、DB2 パッケージの実行時の属性数が決まります。この属性のうち 2 つは、許可を検査するときに使用される許可 ID と、非修飾オブジェクト用に使用される修飾子です。SQLJ プログラムで動的に実行される位置指定 UPDATE および DELETE ステートメントの許可を正しいものにするために、`db2sqljbind` は常に DB2 パッケージを **DYNAMICRULES(BIND)** オプションにバインドします。このオプションは変更できません。 **DYNAMICRULES(BIND)** オプションを指定すると、**SET CURRENT SQLID** ステートメントおよび **SET CURRENT SCHEMA** ステートメントは SQLJ プログラムに対して無効になります。なぜなら、これらのステートメントは **BIND** 以外の **DYNAMICRULES** 値を使ってバインドされている動的ステートメントにのみ影響を与えるからです。

DYNAMICRULES(BIND) を指定する場合、動的 SQL ステートメント内の非修飾の表、ビュー、索引、および別名はバインド・オプション **QUALIFIER** の値を使って暗黙的に修飾されます。 **QUALIFIER** を指定しない場合、DB2 はパッケージ所有者の許可 ID を暗黙的な修飾子として使用します。この動作がご使用のプログラムに適していない場合、以下の手法のいずれかを使用して正しい修飾子を設定できます。

- 位置指定 UPDATE および DELETE ステートメントを静的に強制実行します。位置指定 UPDATE または DELETE ステートメントのカーソル (イテレーター) が位置指定 UPDATE または DELETE ステートメントと同じパッケージ内にある場合、`db2sqljcustomize` または `db2sqljbind` の `-staticpositioned YES` オプションを使用してこの強制実行を行えます。
- 位置指定 UPDATE および位置指定 DELETE ステートメント内の DB2 表名を完全修飾します。

db2sqljcustomize - SQLJ プロファイル・カスタマイザー

db2sqljcustomize は、組み込み SQL ステートメントを含む SQLJ プロファイルを処理します。

デフォルトでは、それぞれの分離レベルごとに 1 つずつ、4 つの DB2 パッケージが db2sqljcustomize によって作成されます。db2sqljcustomize では、実行時に使用できるように、DB2 の固有情報を使ってプロファイルが拡大されます。

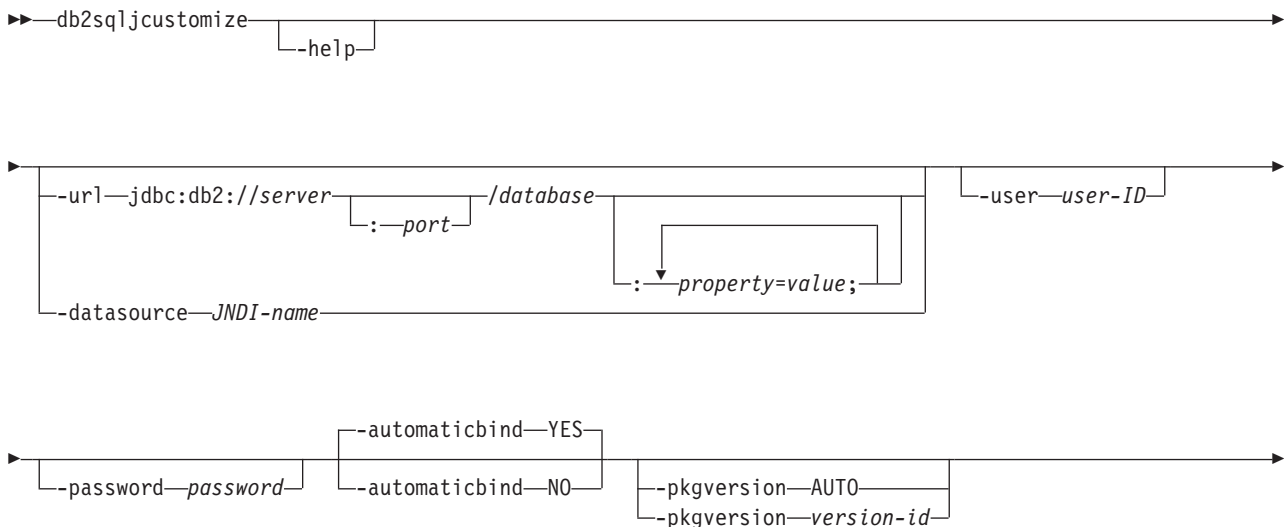
許可

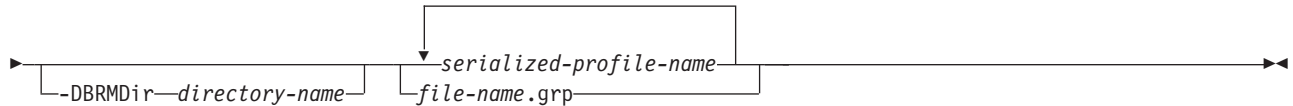
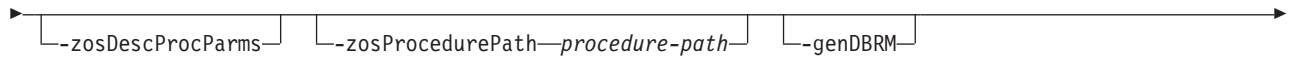
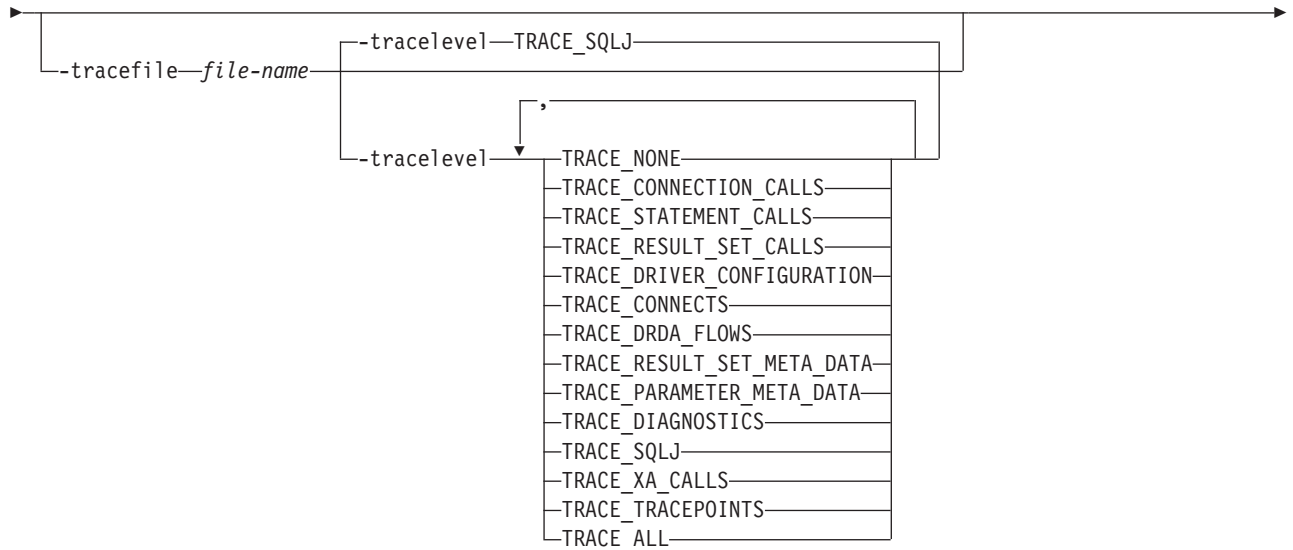
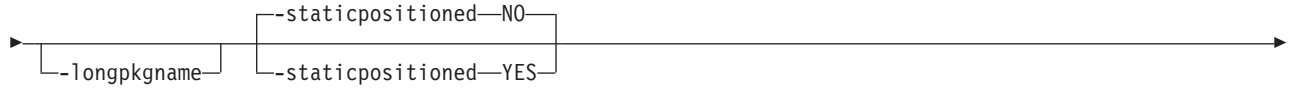
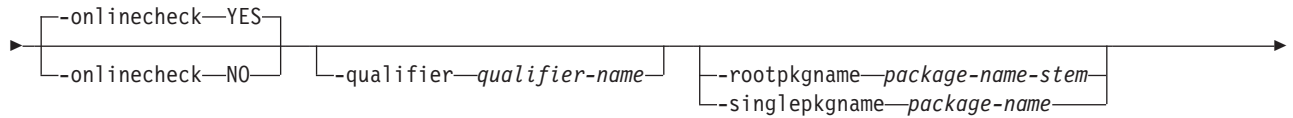
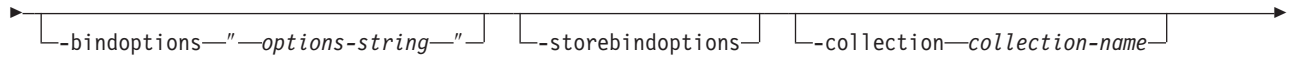
プロセスの特権セットには、以下の権限のうちの 1 つを組み込む必要があります。

- SYSADM authority
- DBADM 権限
- パッケージが存在しない場合の BINDADD 特権と、以下のいずれかの特権
 - CREATEIN 特権
 - 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
- パッケージが存在する場合に必要な特権:
 - スキーマに対する ALTERIN 特権
 - パッケージに対する BIND 特権

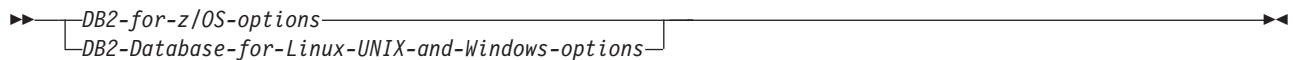
ユーザーには、アプリケーションで静的 SQL ステートメントをコンパイルするのに必要なすべての特権も必要です。グループに認可された特権が、静的ステートメントの許可の検査に使用されることはありません。ユーザーが、SYSADM 権限は持っていない場合、DB2 データベース・マネージャーによって明示的な DBADM 権限が自動的に認可されます。

コマンド構文

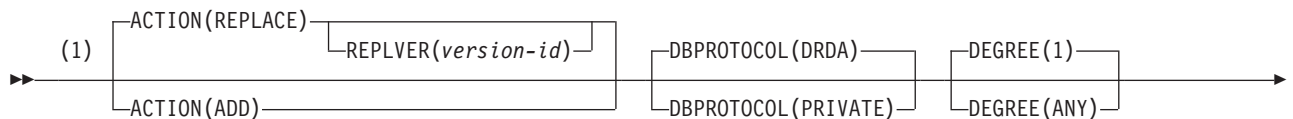


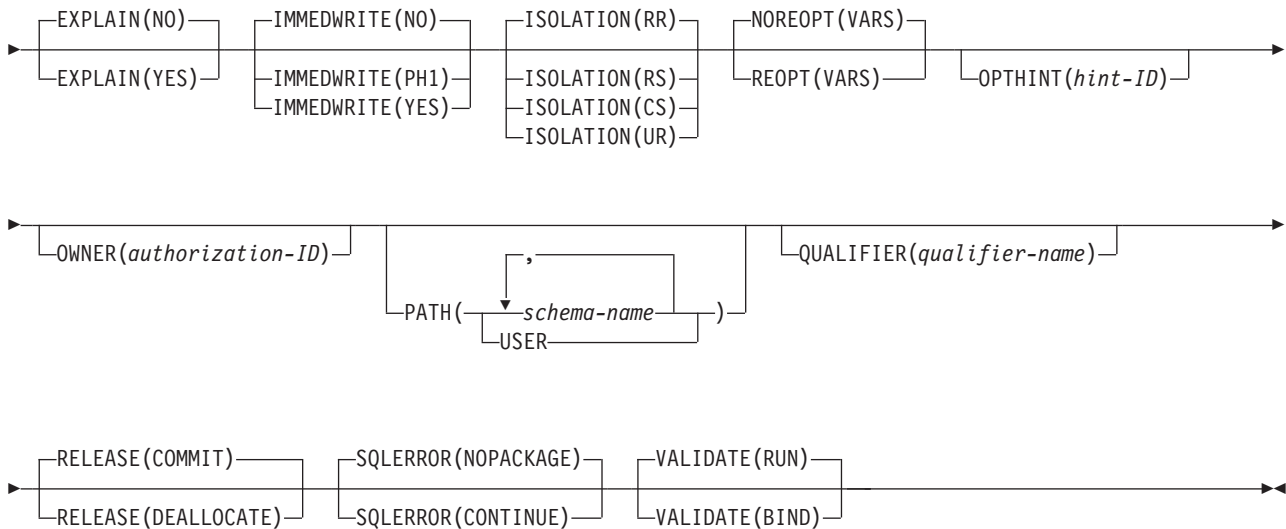


options-string:



DB2 for z/OS options:

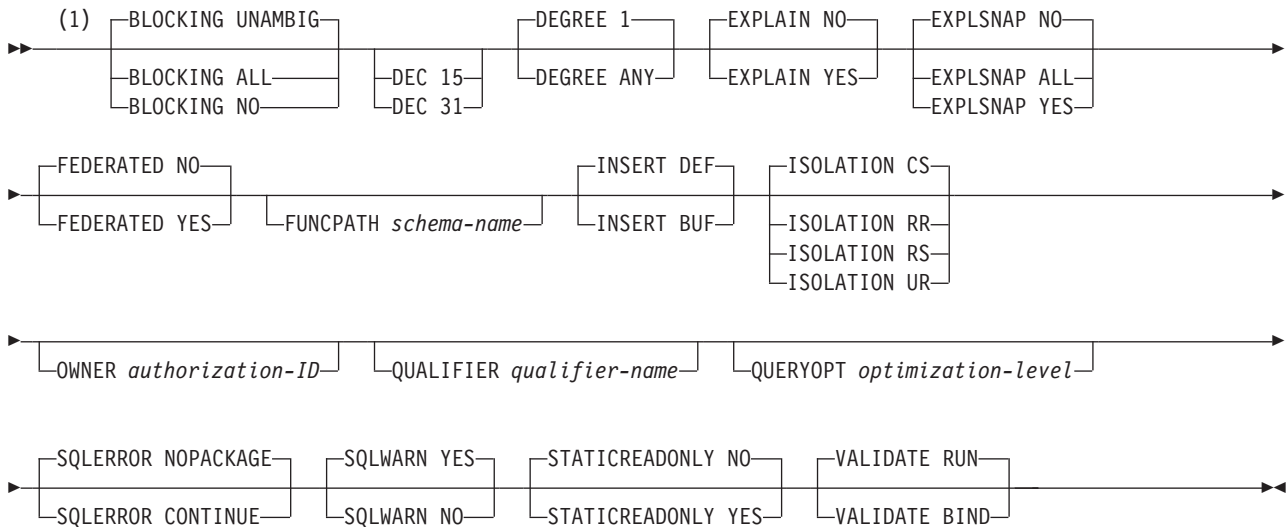




注:

- 1 これらのオプションは、どのような順序で指定してもかまいません。

DB2 Database for Linux, UNIX, and Windows options



注:

- 1 これらのオプションは、どのような順序で指定してもかまいません。

コマンド・パラメーター

-help

SQLJ カスタマイザーがサポートする各オプションについて、説明を表示するように指定します。-help と共に他のオプションが指定されても、それらは無視されます。

-url

プロファイルがカスタマイズされるデータ・ソースの URL を指定します。-automaticbind オプションまたは -onlinecheck オプションが YES に指定さ

れているか、デフォルトで YES になった場合、この URL が表すデータ・ソースへの接続が確立されます。-url 値の可変部は、以下のとおりです。

server

DB2 サブシステムが存在する z/OS システムのドメイン名または IP アドレス。

port

DB2 サブシステムに割り当てられている TCP/IP サーバーのポート番号。デフォルトは 446 です。

-url

プロファイルがカスタマイズされるデータ・ソースの URL を指定します。-automaticbind オプションまたは -onlinecheck オプションが YES に指定されているか、デフォルトで YES になった場合、この URL が表すデータ・ソースへの接続が確立されます。-url 値の可変部は、以下のとおりです。

server

データベース・サーバーが存在するオペレーティング・システムのドメイン名または IP アドレス。

port

データベース・サーバーに割り当てられている TCP/IP サーバーのポート番号。デフォルトは 446 です。

database

プロファイルがカスタマイズされるデータベース・サーバーの名前。

接続を DB2 for z/OS サーバーに対して行う場合、*database* は、インストール中に定義される DB2 ロケーション名です。この値の中の文字はすべて大文字でなければなりません。サーバーで以下の SQL ステートメントを実行することにより、ロケーション名を判別できます。

```
SELECT CURRENT SERVER FROM SYSIBM.SYSDUMMY1;
```

接続を DB2 Database for Linux, UNIX, and Windows サーバーに対して行う場合、*database* は、インストール中に定義されるデータベース名です。

接続を IBM Cloudscape サーバーに対して行う場合、*database* は、データベースを含むファイルの完全修飾名です。この名前は、二重引用符 (") で囲まなければなりません。以下に例を示します。

```
"c:/databases/testdb"
```

```
property=value;
```

JDBC 接続のプロパティー。

```
property=value;
```

JDBC 接続のプロパティー。

-datasource JNDI-name

JNDI に登録された DataSource オブジェクトの論理名を指定します。DataSource オブジェクトは、プロファイルがカスタマイズされるデータ・ソースを表します。-automaticbind オプションまたは -onlinecheck オプションが YES に指定されているか、デフォルトで YES になった場合、このデータ・ソースへの接続が

確立されます。-datasource は、-url を指定する代わりに指定できます。
DataSource オブジェクトは、IBM Data Server Driver for JDBC and SQLJ
Type 4 接続 を使用する接続を表す必要があります。

-user *user-ID*

オンライン検査またはパッケージのバインドのためにデータ・ソースに接続する
ときに使うユーザー ID を指定します。-url を指定する場合は、-user を指定し
なければなりません。-datasource を指定し、*JNDI-name* が表す DataSource オブ
ジェクトにユーザー ID が含まれない場合、-user を指定しなければなりません。

-password *password*

オンライン検査またはパッケージのバインドのためにデータ・ソースに接続する
ときに使うパスワードを指定します。-url を指定する場合は、-password を指
定しなければなりません。-datasource を指定し、*JNDI-name* が表す DataSource
オブジェクトにパスワードが含まれない場合、-password を指定しなければなり
ません。

-automaticbind YES|NO

-url パラメーターで指定されているデータ・ソースにおいてカスタマイザーが
DB2 パッケージをバインドするかどうかを指定します。

デフォルトは YES です。

パッケージ数と、それらのパッケージの分離レベルは、-rootpkgname および
-singlepkgname オプションで制御します。

バインド操作が功を奏するには、以下の条件が満たされる必要があります。

- TCP/IP と DRDA がターゲットのデータ・ソースにインストールされていなければなりません。
- -url、-username、および -password にそれぞれ有効な値を指定する必要があります。
- -username の値には、ターゲット・データ・ソースにおいてパッケージをバインドする許可がなければなりません。

-pkgversion AUTO|*version-id*

カスタマイズしようとしているシリアライズド・プロファイル用のパッケージが
サーバー側でバインドされるときに使用するパッケージ・バージョンを指定しま
す。db2sqljcustomize は、シリアライズド・プロファイル内と DB2 パッケージ
内にバージョン ID を保管します。実行時のバージョン検査では、バージョン
名ではなく、整合性トークンがベースになります。整合性トークンに基づいたバ
ージョン名を自動的に生成するには、-pkgversion AUTO を指定します。

デフォルトでは、バージョンはありません。

-bindoptions *options-string*

スペースで区切ったオプションのリストを指定します。それらのオプションに
は、同名の DB2 プリコンパイル・オプションおよびバインド・オプションと同
じ機能があります。DB2 for z/OS システム上で実行されるプログラムを準備
する場合、DB2 for z/OS のオプションを指定します。DB2 Database for
Linux, UNIX, and Windows システム上で実行されるプログラムを準備する場
合、DB2 Database for Linux, UNIX, and Windows のオプションを指定しま
す。

バインド・オプションに関する注:

- ISOLATION を指定するのは、-singlepkgname オプションも指定する場合だけです。
- STATICREADONLY をサポートするサーバーの場合は、STATICREADONLY の値は YES、その他のサーバーの場合は、NO となります。
STATICREADONLY YES を指定すると、DB2 は未確定カーソルを読み取り専用カーソルであるものとして処理します。イテレーター宣言のエラーのトラブルシューティングの場合、STATICREADONLY NO を明示的に指定する必要があります。あるいは、確定済みになるようにイテレーターを宣言します。例えば、イテレーターを確定的に更新可能にする場合は、`sqlj.runtime.ForUpdate` をインプリメントするようにイテレーターを宣言します。イテレーターを読み取り専用にする場合、イテレーターを使用する SELECT ステートメントに FOR READ ONLY 節を付け加えます。

重要: パッケージのバインディングを行う場所であるデータ・ソースに適したプログラム準備オプションのみを指定してください。IBM Data Server Driver for JDBC and SQLJ の一部の値およびデフォルト値は、DB2 用の値およびデフォルト値とは異なります。

-storebindoptions

-bindoptions および -staticpositioned パラメーターの値をシリアライズド・プロファイル内に保管することを指定します。-bindoptions または -staticpositioned パラメーターを使用しないで `db2sqljbind` を呼び出すと、シリアライズド・プロファイルに保管されている値がバインド操作で使用されます。1 回の `db2sqljcustomize` の呼び出しで複数のシリアライズド・プロファイルを指定すると、パラメーター値は、それぞれのシリアライズド・プロファイル内に保管されます。保管値は、`db2sqljprint` ユーティリティからの出力に表示されます。

-collection *collection-name*

`db2sqljcustomize` がバインドするパッケージの修飾子。 `db2sqljcustomize` は、カスタマイズされたシリアライズド・プロファイル内にその値を保管します。その値は、関連パッケージのバインディング時に使用されます。このパラメーターを指定しないと、`db2sqljcustomize` では NULLID のコレクション ID が使用されます。

-onlinecheck YES|NO

SQLJ プログラム内のデータ・タイプのオンライン検査を実行するかどうかを指定します。-url または -datasource オプションは、オンライン検査で使用されるデータ・ソースを決定します。-url または -datasource パラメーターを指定した場合のデフォルトは YES です。それ以外の場合のデフォルトは NO です。

-qualifier *qualifier-name*

オンライン検査で使用するための、SQLJ プログラム内の非修飾オブジェクトの修飾子を指定します。パッケージのバインディング時には、この値は修飾子として使用されません。

-rootpkgname|singlepkgname

プログラムに関連したパッケージの名前を指定します。-automaticbind が NO である場合、これらのパッケージ名が `db2sqljbind` 実行時に使用されます。各パラメーターの意味は、以下のとおりです。

-rootpkgname *package-name-stem*

カスタマイザーが、4 つの DB2 分離レベルのそれぞれに対して 1 つずつ、4 つのパッケージを作成することを指定します。4 つのパッケージの名前は以下のとおりです。

package-name-stem1

分離レベル UR 用

package-name-stem2

分離レベル CS 用

package-name-stem3

分離レベル RS 用

package-name-stem4

分離レベル RR 用

-longpkgname が指定されていない場合、*package-name-stem* は 7 バイト以下の英数字ストリングでなければなりません。

-longpkgname が指定されている場合、*package-name-stem* は 127 バイト以下の英数字ストリングでなければなりません。

-singlepkgname *package-name*

カスタマイザーが *package-name* という名前の 1 つのパッケージを作成することを指定します。このオプションを指定した場合、プログラムは 1 つの分離レベルでのみ実行されます。パッケージの分離レベルの指定は、-bindoptions オプション・ストリングの ISOLATION オプションで指定します。

-longpkgname が指定されていない場合、*package-name* は 8 バイト以下の英数字ストリングでなければなりません。

-longpkgname が指定されている場合、*package-name* は 128 バイト以下の英数字ストリングでなければなりません。

-singlepkgname オプションの使用は推奨されていません。

推奨: ターゲット・データ・ソースが DB2 for z/OS である場合は、*package-name-stem* または *package-name* の値には大文字を使用してください。特定の CCSID 値で定義された DB2 for z/OS システムでは、パッケージ名またはコレクション名内の小文字が許容されません。

-rootpkgname または -singlepkgname を指定しない場合、db2sqljcustomize はシリアライズド・プロファイル名に基づいた 4 つのパッケージ名を生成します。シリアライズド・プロファイル名の形式は、以下のようになります。

program-name_SJProfileIDNumber.ser

生成される 4 つのパッケージ名は、以下の形式になります。

Bytes-from-program-nameIDNumberPkgIsolation

861 ページの表 37 生成されるパッケージ名の各部分およびそれぞれのバイト数を示します。

パッケージ名の最大長は、*maxlen* です。-longpkgname が指定されない場合、*maxlen* は 8 です。-longpkgname が指定される場合は、*maxlen* は 128 です。

表 37. db2sqljcustomize によって生成されるパッケージ名の部分

パッケージ名の部分	バイト数	値
<i>Bytes-from-program-name</i>	$m=\min(\text{Length}(\text{program-name}), \text{maxlen}-1-\text{Length}(\text{IDNumber}))$	<i>program-name</i> の最初の m バイト。大文字。
<i>IDNumber</i>	$\text{Length}(\text{IDNumber})$	<i>IDNumber</i>
<i>PkgIsolation</i>	1	1、2、3、または 4。この値は、パッケージのトランザクション分離レベルを表します。表 38を参照してください。

表 38は、db2sqljcustomize によって生成されるパッケージ名の *PkgIsolation* 部分の値を示します。

表 38. *PkgIsolation* 値および関連した分離レベル

<i>PkgNumber</i> 値	パッケージの分離レベル
1	非コミット読み取り (UR)
2	カーソル固定 (CS)
3	読み取り固定 (RS)
4	反復可能読み取り (RR)

例: プロファイル名が *ThisIsMyProg_SJProfile111.ser* であるとしてします。db2sqljcustomize オプション *-longpkgname* が指定されていません。したがって、*Bytes-from-program-name* は、*ThisIsMyProg* の大文字に変換された最初の 4 バイト、*THIS* です。*IDNumber* は 111 です。4 つのパッケージ名は以下のとおりです。

```
THIS1111
THIS1112
THIS1113
THIS1114
```

例: プロファイル名が *ThisIsMyProg_SJProfile111.ser* であるとしてします。db2sqljcustomize オプション *-longpkgname* が指定されています。したがって、*Bytes-from-program-name* は、*ThisIsMyProg* が大文字に変換されたもの、つまり *THISISMYPROG* です。*IDNumber* は 111 です。4 つのパッケージ名は以下のとおりです。

```
THISISMYPROG1111
THISISMYPROG1112
THISISMYPROG1113
THISISMYPROG1114
```

例: プロファイル名が *A_SJProfile0.ser* であるとしてします。*Bytes-from-program-name* は *A* で、*IDNumber* は 0 です。したがって、4 つのパッケージ名は以下のとおりです。

```
A01
A02
A03
A04
```

db2sqljcustomize にパッケージ名を生成させることは、推奨されていません。生成されたパッケージ名に既存のパッケージの名前と同じものがある場合、

db2sqljcustomize は既存のパッケージを上書きします。パッケージ名が必ず固有になるようにするためには、-rootpkgname を指定します。

-longpkgname

128 バイトまでの DB2 パッケージ名を db2sqljcustomize で生成できることを指定します。このオプションは、長いパッケージ名をサポートするサーバーでパッケージをバインドする場合にのみ使用してください。-singlepkgname または -rootpkgname を指定する場合、以下の条件では、-longpkgname も指定しなければなりません。

- -singlepkgname の引数が 8 バイトよりも長い。
- -rootpkgname の引数が 7 バイトよりも長い。

-staticpositioned NOIYES

イテレーターを使用する位置指定 UPDATE ステートメントと同じソース・ファイルに宣言されているイテレーターの場合、静的にバインドされたステートメントとしてその位置指定 UPDATE を実行するかどうかを指定します。デフォルトは NO です。NO は、動的に準備されるステートメントとして位置指定 UPDATE を実行することを意味します。

-zosDescProcParms

DB2 for z/OS は、ストアード・プロシージャ・パラメーターに対する DESCRIBE 操作の実行を指定します。

-zosDescProcParms は、DB2 for z/OS データベース・サーバーで実行されるプログラムにのみ適用されます。

DESCRIBE 情報が入手可能な場合、SQLJ には INOUT および OUT パラメーターの長さおよび精度に関する情報が含まれるため、これらのパラメーターに必要なメモリー量のみが割り振られます。DESCRIBE 情報が入手できることによって最も大きな影響を受けるのは、INOUT、LOB OUT、および decimal OUT 文字パラメーターのためのストレージの使用です。

-zosDescProcParms を指定すると、DB2 データベース・サーバーは、指定された値または -zosProcedurePath のデフォルト値を使用して、DESCRIBE 情報が要求されるストアード・プロシージャの非修飾名を解決します。

-zosProcedurePath *procedure-path*

SQLJ プログラムのオンライン・チェック時に DB2 for z/OS が非修飾ストアード・プロシージャ名の解決に使用する、スキーマ名のリストを指定します。

-zosProcedurePath は、DB2 for z/OS データベース・サーバーで実行されるプログラムにのみ適用されます。

リストは、スキーマ名のコンマ区切りのリストを二重引用符で囲んだストリング値になります。DB2 データベース・サーバーは、そのリストを非修飾ストアード・プロシージャ名を解決するための SQL パスに挿入します。SQL パスは次のとおりです。

`SYSIBM, SYSFUN, SYSPROC, procedure-path, qualifier-name, user-ID`

qualifier-name は -qualifier パラメーターの値であり、*user-ID* は -user パラメーターの値です。

DB2 データベース・サーバーは、データベース・サーバーに存在するストアード・プロシージャの名前と一致するスキーマ名が見つかるまで、SQL パス内のスキーマ名を左から右に確認します。一致するものがあると、DB2 データベ

ース・サーバーはそのストアード・プロシージャのパラメーターに関する情報を DB2 カタログから取得します。DB2 データベース・サーバーが一致するものを検出しなかった場合、SQLJ は DB2 カタログ情報を使用せずにパラメーター・データを設定します。

-zosProcedurePath を指定しない場合、DB2 データベース・サーバーは次の SQL パスを使用します。

SYSIBM, SYSFUN, SYSPROC, *qualifier-name*, *user-ID*

-qualifier パラメーターを指定しない場合、SQL パスには *qualifier-name* は含まれません。

-genDBRM

db2sqljcustomize がデータベース要求モジュール (DBRM) を生成することを指定します。これらの DBRM は、DB2 for z/OS プランおよびパッケージを作成するのに使用できます。

-genDBRM は、DB2 for z/OS データベース・サーバーで実行されるプログラムにのみ適用されます。

-genDBRM および -automaticbind NO が指定されていると、db2sqljcustomize は DBRM を作成しますが、それらを DB2 パッケージにバインドすることはしません。-genDBRM および -automaticbind YES が指定されていると、db2sqljcustomize は DBRM を作成し、それらを DB2 パッケージにバインドします。

各 DB2 分離レベルごとに 1 つの DBRM が作成されます。生成された DBRM ファイルの命名規則は、パッケージの命名規則と同じものです。例えば、-rootpkgname SQLJSA0 が指定されており、-genDBRM も指定されている場合、4 つの DBRM ファイルの名前は次のようになります。

- SQLJSA01
- SQLJSA02
- SQLJSA03
- SQLJSA04

-DBRMDir *directory-name*

-genDBRM を指定する場合、-DBRMDir は生成された DBRM ファイルを db2sqljcustomize が配置するローカル・ディレクトリーを指定します。デフォルトは、現行ディレクトリーです。

-DBRMdir は、DB2 for z/OS データベース・サーバーで実行されるプログラムにのみ適用されます。

-tracefile *file-name*

トレースを使用可能にし、トレース情報のための出力ファイルを示します。このオプションを指定できるのは、IBM ソフトウェア・サポートから指示された場合のみです。

-tracelevel

-tracefile を指定する場合、db2sqljcustomize の実行中に何をトレースするかを指示します。デフォルトは TRACE_SQLJ です。このオプションを指定できるのは、IBM ソフトウェア・サポートから指示された場合のみです。

***serialized-profile-name*|*file-name*.grp**

カスタマイズしようとしている 1 つ以上のシリアライズド・プロファイルの名

前を指定します。指定されたシリアライズド・プロファイルは、CLASSPATH 環境変数で指定されたディレクトリーになければなりません。

シリアライズド・プロファイル名の形式は、以下のようになります。

```
program-name_SJProfileIDNumber.ser
```

シリアライズド・プロファイルの名前を指定するときは、.ser 拡張子を付けても付けなくてもかまいません。

program-name は、拡張子 .sqlj の付いていない SQLJ ソース・プログラムの名前です。*n* は、0 から *m-1* までの整数です。ただし、*m* は、SQLJ ソース・プログラムから SQLJ 変換プログラムが生成したシリアライズド・プロファイルの数です。

以下のいずれかの方法で、シリアライズド・プロファイルの名前を指定することができます。

- db2sqljcustomize コマンドで名前の一覧を表示します。複数のシリアライズド・プロファイル名は、スペースを使用して区切る必要があります。
- *file-name.grp* という名前のファイルに、シリアライズド・プロファイル名を 1 行に 1 つずつ指定し、db2sqljcustomize コマンドに *file-name.grp* を指定します。

複数のシリアライズド・プロファイル名を指定し、しかもデフォルト値の -automaticbind YES を指定または使用した場合、db2sqljcustomize によってそれらのプロファイルから 1 つの DB2 パッケージがバインドされます。

db2sqljcustomize を使用して、複数のシリアライズド・プロファイルから 1 つの DB2 パッケージを作成するには、-rootpkgname または -singlepkgname オプションも一緒に指定する必要があります。

複数のシリアライズド・プロファイル名を指定し、さらに -automaticbind NO も指定した場合に、db2sqljbind の実行時にそのシリアライズド・プロファイルを 1 つの DB2 パッケージにバインドするには、シリアライズド・プロファイル名が同じ順序で並んだ同じリストを db2sqljcustomize および db2sqljbind に指定する必要があります。

出力

db2sqljcustomize を実行すると、カスタマイズされたシリアライズド・プロファイルが作成されます。また、automaticbind 値が YES の場合は、DB2 パッケージも作成されます。

例

```
db2sqljcustomize -user richler -password mordecai
  -url jdbc:db2:/server:50000/sample -collection duddy
  -bindoptions "EXPLAIN YES" pgmname_SJProfile0.ser
```

使用上の注意

常にオンライン検査をお勧めします。 シリアライズド・プロファイルをカスタマイズする際には、オンライン検査を使用するよう強くお勧めします。オンライン検査によって、DB2 ホスト変数のデータ・タイプおよび長さに関する情報が確認されます。これは、以下の項目の場合特に重要です。

- java.lang.String ホスト変数および CHAR 列を備えた述部

他のホスト言語の文字変数とは違って、Java の String ホスト変数の宣言では、長さ属性は使用されません。文字ホスト変数が含まれる照会を正しく最適化するには、そのホスト変数の長さが DB2 で必要です。例えば、ある照会が述部をもっていて、そこで、String ホスト変数が CHAR 列と比較され、CHAR 列上で索引が定義されると仮定します。DB2 がホスト変数の長さを判別できない場合、索引スキャンではなく表スペース・スキャンを行うことがあります。オンライン検査は、対応する文字カラムの長さを提示することによって、このような問題が起きないようにします。

- java.lang.String ホスト変数および GRAPHIC 列を備えた述部

オンライン検査を行わないと、String ホスト変数を GRAPHIC 列と比較する述部が検出された場合、バインド・エラー (SQLCODE -134) が DB2 から発行されることがあります。

- リモート・サーバーでの SQLJ SELECT ステートメントの結果表内の列名:

オンライン検査を行わないと、リモートの SELECT の結果表の列名をドライバーが判別できません。

複数のシリアライズド・プロファイルの一括カスタマイズ: 複数のシリアライズド・プロファイルをまとめてカスタマイズし、1 つの DB2 パッケージを作成することができます。そのように作成する場合に、-staticpositioned YES を指定すると、カーソルの宣言とは異なるソース・ファイル内に UPDATE または DELETE ステートメントがあっても、パッケージ内で事前に宣言されているカーソルを参照する位置指定の UPDATE または DELETE ステートメントは静的に実行されます。複数のソース・ファイルでプログラムが構成されている場合に、-staticpositioned YES の動作が望ましい場合、パッケージ内で位置指定の UPDATE または DELETE ステートメントより前にカーソル宣言が出現するように、db2sqljcustomize コマンド内でプロファイルを配列する必要があります。そのためには、イテレーターを参照する位置指定の UPDATE または DELETE ステートメントを収めたプロファイルより前に、結果表をイテレーターに割り当てる SELECT ステートメントを収めたプロファイルをリストします。

データ・ソースでカスタマイズされたカスタマイズ済みシリアライズド・プロファイルの、別のデータ・ソースでの使用: db2sqljcustomize を実行して、ある 1 つのデータ・ソースで、SQLJ プログラム用にカスタマイズされたシリアライズド・プロファイルを作成してから、別のデータ・ソースでそのシリアライズド・プロファイルを使用することができます。それには、db2sqljcustomize を 1 回実行して作成したカスタマイズ済みシリアライズド・プロファイルに対して、db2sqljbind を複数回実行します。それらのデータ・ソースでプログラムを実行するときは、そのプログラムからアクセスする DB2 オブジェクトは、どちらのデータ・ソースでも同じでなければなりません。例えば、どのデータ・ソースの表も、同じコード化スキームおよび同じデータ・タイプと同じ列をもっていなければなりません。

-collection パラメーターの使用: db2sqljcustomize は、作成した各カスタマイズ済みシリアライズド・プロファイル内に DB2 コレクション名を保管します。SQLJ プログラムが実行されると、ドライバーは、カスタマイズ済みシリアライズド・プロファイル内に保管されているコレクション名を使用して、実行するパッケージを検索します。カスタマイズ済みシリアライズド・プロファイルに保管されている名前は、-collection パラメーターの値によって判別されます。シリアライズド・プロフ

ファイルに保管できるコレクション ID は 1 つだけです。ただし、`-bindoptions` パラメーター内で `COLLECTION` オプションを指定すれば、1 つのシリアライズド・プロファイルを複数のパッケージ・コレクションにバインドすることができます。シリアライズド・プロファイルに指定されているもの以外のコレクション内にあるパッケージを実行するには、プログラム内で `SET CURRENT PACKAGESET` ステートメントを使用します。

VERSION パラメーターの使用: 1 つの SQLJ プログラムの複数バージョンのパッケージを 1 つのコレクションにバインドするときは、`VERSION` パラメーターを使用します。これを実行するのは、SQLJ ソース・プログラムを変更した後で、新旧バージョンのプログラムを実行することになった場合です。

2 つのバージョンのパッケージを保守するには、次のステップを行います。

1. ソース・プログラム内のコードを変更します。
2. 新規のシリアライズド・プロファイルを作成するように、ソース・プログラムを変換します。オリジナルのシリアライズド・プロファイルを上書きしないよう気を付けてください。
3. `db2sqljcustomize` を実行し、シリアライズド・プロファイルをカスタマイズしてから、オリジナル・パッケージと同じパッケージ名を付けて同じコレクション内に DB2 パッケージを作成します。それには、オリジナル・パッケージの作成時に使用したのと同じ `-rootpkgname` および `-collection` の値を、新規のパッケージのバインド時に使用します。新規のカスタマイズ済みシリアライズド・プロファイル内と新規のパッケージ内にバージョン ID を書き込むには、`-bindoptions` パラメーターに `VERSION` オプションを指定します。

このステップを実行するときは、`VERSION` オプションを必ず指定してください。指定しないと、オリジナル・パッケージが上書きされてしまいます。

旧バージョンのプログラムを実行すると、旧バージョンのパッケージが DB2 でロードされます。新バージョンのプログラムを実行すると、新バージョンのパッケージが DB2 でロードされます。

DB2 for z/OS でのパッケージおよびプランのバインディング: `db2sqljcustomize` `-genDBRM` パラメーターを使用してローカル・システムに DBRM を作成することができます。続いて、それらの DBRM を DB2 for z/OS システムに転送して、それらをパッケージまたはプランにバインドすることができます。この手法を使用する場合、DBRM ファイルをバイナリー・ファイルとして z/OS システムに、つまりレコード形式 FB およびレコード長 80 のパーティション・データ・セットに転送する必要があります。パッケージまたはプランをバインドするときには、以下のバインド・オプションの値を指定する必要があります。

ENCODING(EBCDIC)

DB2 for z/OS 上の IBM Data Server Driver for JDBC and SQLJ では、パッケージおよびプランには EBCDIC エンコードが必要になります。

DYNAMICRULES(BIND)

このオプションにより、SQLJ が動的 SQL を使用するときの許可規則が一貫性のあるものになります。SQLJ は、複数の SQLJ プログラムが関係する位置指定 UPDATE または DELETE 操作に対して動的 SQL を使用します。

DBPROTOCOL(DRDA)

専用プロトコルは推奨されていません。DBPROTOCOL(DRDA) を全アプリケーションで使用してください。特に、3 部から構成されるリモート表名を使用する SQLJ アプリケーションの場合は、必ず DBPROTOCOL(DRDA) を使用してください。そうしないと、それらのアプリケーションは失敗する可能性があります。

db2sqljprint - SQLJ プロファイル・プリンター

db2sqljprint は、プロファイルのカスタマイズ済みバージョンの内容をプレーン・テキストとして印刷します。

許可

なし

コマンド構文

▶▶—db2sqljprint—*profilename*————▶▶

コマンド・パラメーター

profilename

SQLJ プロファイル・ファイルの相対名または絶対名を指定します。SQLJ ファイルが Java ソース・ファイルに変換されると、そこに含まれる SQL 操作に関する情報は、プロファイルと呼ばれる SQLJ が生成したリソース・ファイルに保管されます。プロファイルは、オリジナルの入力ファイルの名前の後に続く接尾部 *_SJProfileN* (N は整数) で識別されます。その拡張子は *.ser* です。プロファイル名を指定する際には、*.ser* 拡張子を付けても付けなくてもかまいません。

例

```
db2sqljprint pgmname_SJProfile0.ser
```

db2start - Start DB2

単一データベース・パーティションまたはパーティション・データベース環境で定義されているすべてのデータベース・パーティションで、現行のデータベース・マネージャー・インスタンス・バックグラウンド・プロセスを開始します。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 を始動します。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。

db2start コマンドは、インストールされている DB2 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 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 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。

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 the 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_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 にもっと大きい値を指定してください。

ルート・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・マネージャーで必要とされる ulimit 値を設定します。非ルート・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 db2stat 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

▶▶—db2stat—◀◀

コマンド・パラメーター

db2stat

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

```
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じ情報量を戻せないのは明らかですが、それでも実行中のインスタンスのプロセス ID を知りたい場合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 とメモリ使用量を判別できます。

db2stop - DB2 の停止

現行のデータベース・マネージャ・インスタンスを停止します。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_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 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。

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

ほとんどの完全出力の場合、このユーティリティはインスタンス所有者が起動する必要があります。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。

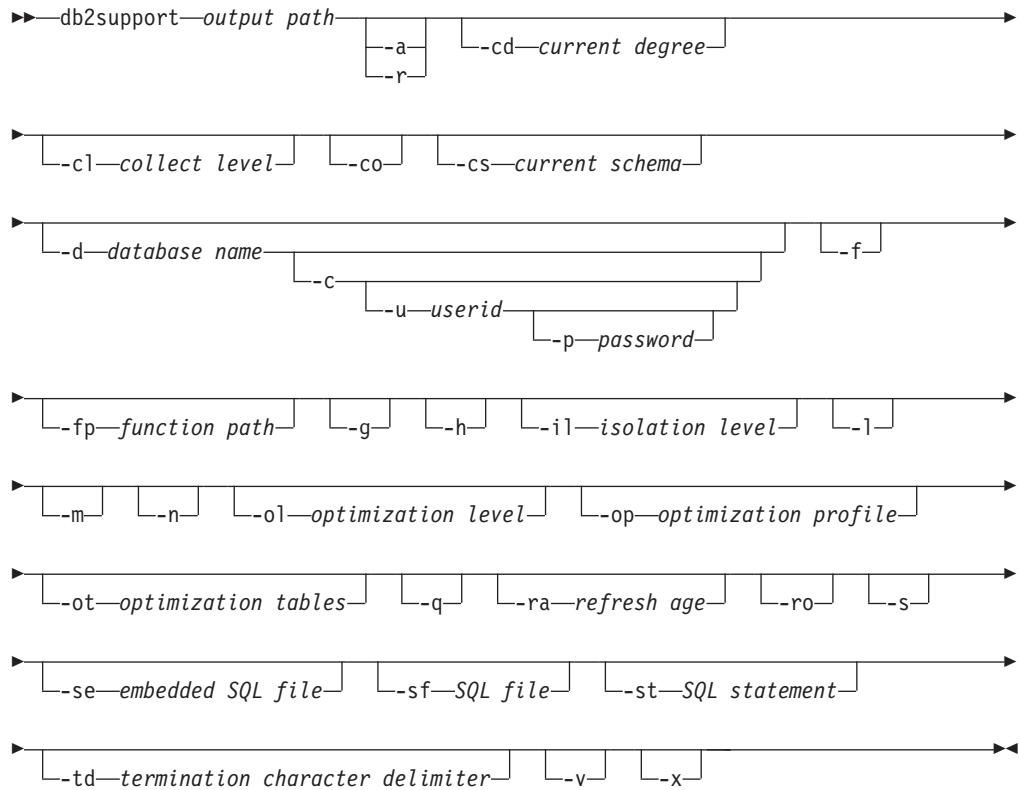
前提条件

db2support を実行する前にデータベースが活動化していなければなりません。そうでないと、情報を収集しても十分な情報が取られません。

必要な接続

なし

コマンド構文



注:

1. このツールをオプティマイザー・モードで呼び出すための独立したオプションはありません。
2. -st、-sf、または -se オプション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、-cl 0 (収集レベル・ゼロ) を使用してください。オプティマイザーの問題を処理するには、ここに示す 4 つ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。
3. ステートメント実行時に特殊レジスターの値がデフォルト以外に設定された場合、同じ値をパラメーターとして db2support ツールに渡すことは、正しい問題分析を行う上で非常に重要です。

コマンド・パラメーター

output path

アーカイブ・ライブラリーを作成するパスを指定します。これは、アーカイブに組み込むために、ユーザー作成ファイルが置かれるディレクトリーです。

-a | -all_core

すべてのコア・ファイルをキャプチャーすることを指定します。

-r | -recent_core

最新のコア・ファイルをキャプチャーすることを指定します。このオプションは、**-a** オプションを指定すると無視されます。

-c | -connect

指定のデータベースに接続するための試行を行うことを指定します。

-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。

-cl | -collect

戻されるパフォーマンス情報のレベルの値を指定します。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
```

-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だけを収集します。

-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の値を指定します。デフォルト値は、現行セッション・ユーザーの許可 ID です。

-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。

-f | -flow

ユーザーに対して、Press <Enter> key to continue という要求が出されたときに、休止を無視します。このオプションは、自動実行が要求される場合に、スクリプトまたはその他の自動プロシーチャーを経由して db2support ツールを実行したり、呼び出したりする際に役に立ちます。

-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われる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)。

-g | -get_dump

ダンプ・ディレクトリー内のすべてのファイル (コア・ファイルを除く) をキャプチャーすることを指定します。

- 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。
- 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。
- 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。
- 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。
- 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。
- ol | -optlevel**
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。
- 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、これが必要です。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。
- 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、これが該当します。CURRENT REFRESH AGE のデフォルト値はゼロです。
- ro | -reopt**
照会を EXPLAIN するとき、REOPT ONCE を指定した EXPLAIN を使用するかどうかを指定します。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。

-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。

-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** オプションと同じです。デフォルトはセミコロンです。

-u *userid* | **-user** *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。

-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

db2support ツールは、以下のいずれかの方法によってオプティマイザー・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、オプティマイザー・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プティマイザー・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プティマイザー・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せん。

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遅い照会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。システム情報と診断ファイルもまた収集されます。

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  
-ol 5 -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非常に重要です。

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。オプションによっては、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 ツールから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からのファイル (DIAGPATH) 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。組み込み静的 SQL ステートメントを含むファイルでも、これが該当します。

- 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する場合、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す。
- ストアド・プロシージャはサポートされません。

db2support は、動的 SQL の EXPLAIN データを収集しません。

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。そこで表示される指示に従うことによって、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。

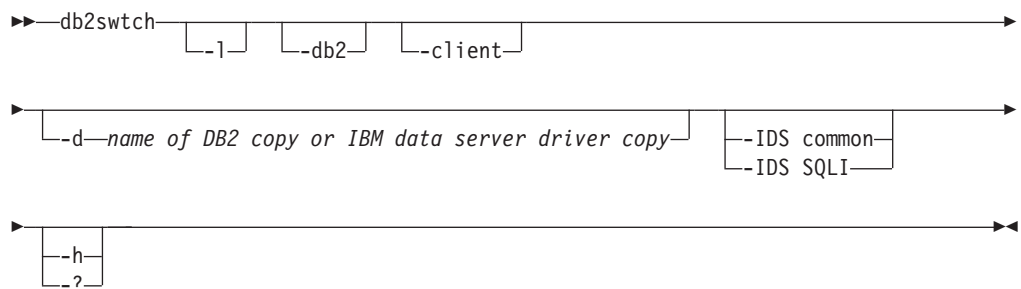
許可

sysadm

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 name of DB2 copy

デフォルト DB2 コピーを指定された名前のもに切り替えます。

db2swtch -db2 -d DB2 コピーの名前

-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のに切り替えます。

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

-d *name of DB2 copy*

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、指定した名前のものに切り替えます。

db2swtch -d *name of DB2 copy*

-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 [options]
    --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。また、DB2 コントロール・センターの起動ポイントも提供します。

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ときに限り、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 プラットフォームでのみ使用できます。

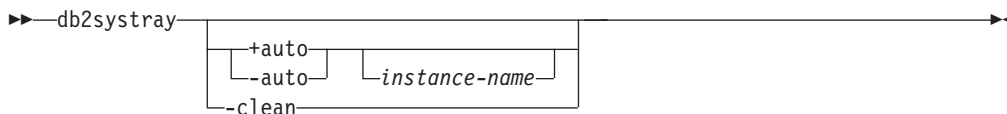
許可

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、「スタートアップ時にツールを起動 (Launch Tool at Startup)」db2sysray メニュー・オプション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 db2sysstray 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 db2sysstray を構成します。

5. C:\SQLLIB\bin\> db2sysstray -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\> db2sysstray -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

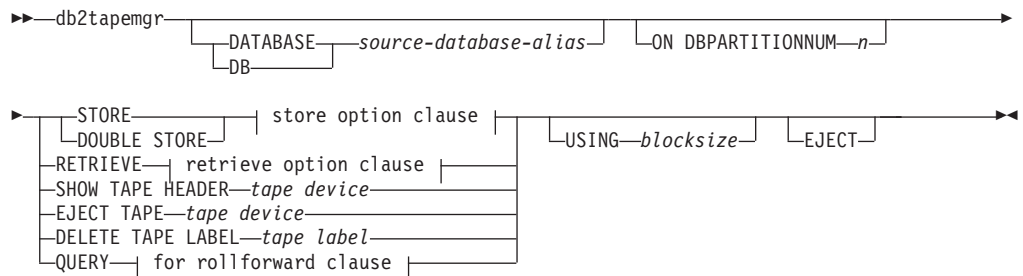
DB2 ログ・ファイルの保管および取り出しをテープで行えるようにします。テープでの位置は、履歴ファイルに保管されます。

許可

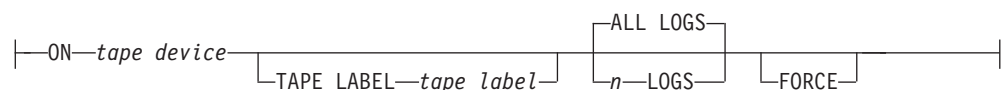
以下のいずれか。

- sysadm
- sysctrl
- sysmaint

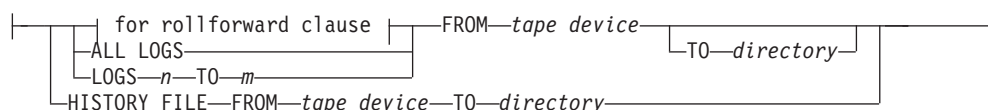
コマンド構文



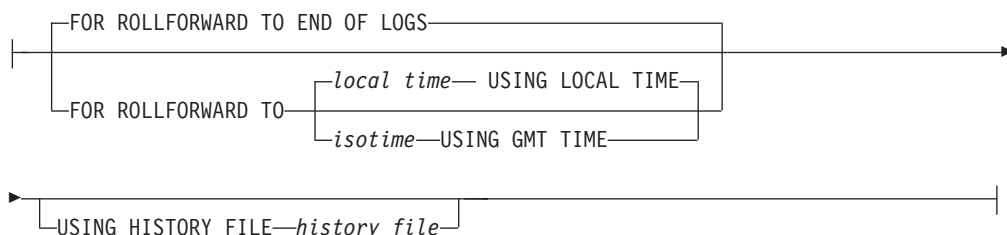
store option clause:



retrieve option clause:



for rollforward clause:



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

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。値を指定しない場合、DB2NODE 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-aliastimestamp* 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 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・コマンド

db2tdbmgr コマンドは、特定のツール・カタログ・データベースで MIGRATE 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。

db2trc - トレース

db2trc コマンドは、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します。

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ために特に役立ちます。DB2 製品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 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 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。

なお、トレース機能を有効にすると (デフォルトではオフになっている)、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。そして、十分な情報を記録した後にトレースをオフにしてください。

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

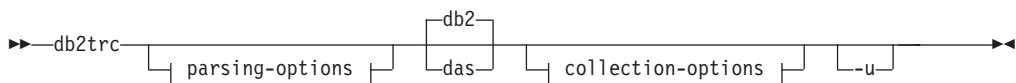
UNIX オペレーティング・システムで DAS のトレースを行うためには、DASADM 許可が必要です。

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。

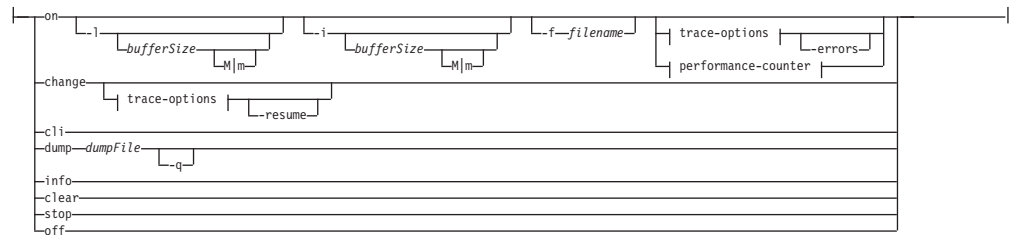
必要な接続

なし

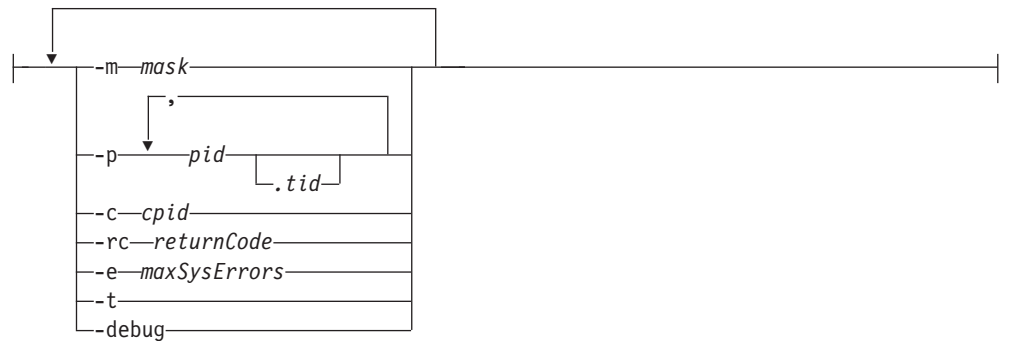
コマンド構文



collection-options:



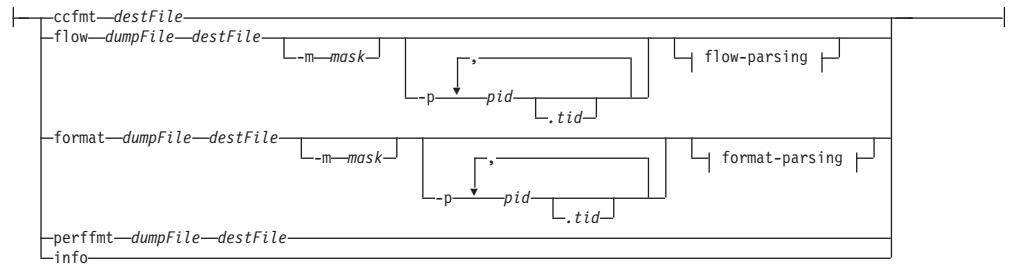
trace-options:



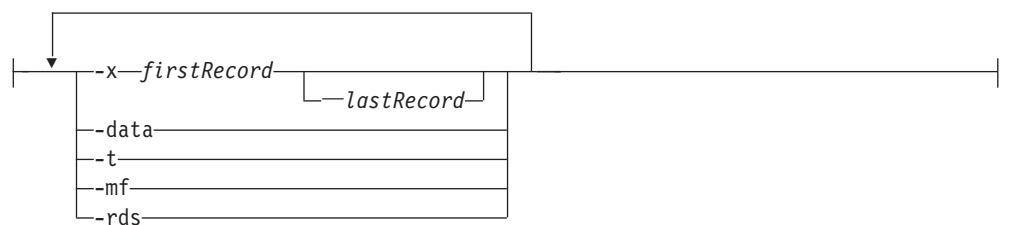
performance-counter:



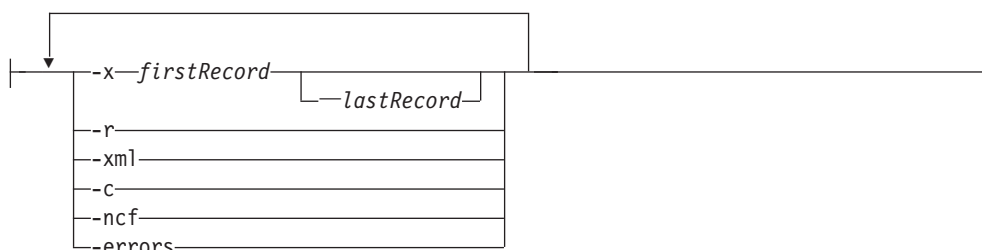
parsing-options:



flow-parsing:



format-parsing:



コマンド・パラメーター

- 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。
- 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、この後に出てくる『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に、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。

-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 `-perfcoun` オプションと同時に指定することはできません。

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、この後に出てくる『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 *.*.SQL0,SQL.*.entry,exit`

-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。最大 5 つの `pid.tid` の組み合わせがサポートされています。

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
```

-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。

cli この *collection* オプションによって、CLI アプリケーション・トレース情報を収集できます。

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

Buffer size           : 2097152 bytes (2 MB)
Allow buffer to wrap  : yes
Mask                  : *.*.*.*.*
Timestamps            : enabled
PID.TID mask          : all
Fixed data mask #1    : all
Fixed data mask #2    : all
Max system errors     : infinite
Treat this rc as sys err: none
```

dump dumpFile

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管されます。

-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。

-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。

-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。

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。

clear トレース・バッファの内容を、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
```

-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、以下の『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
```

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読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
```

`db2trc` コマンドの一般的な構文を以下に示します。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
perf fmt - Format a performance trace
```

For more information add the "-u" option to any of the above commands

上記の『Stage #2 - Parsing』の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 `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.*.*"
```

db2uidl - V5 セマンティクスへのユニーク索引変換の準備

ユーザー自身のスケジュールで、ユニーク索引を段階的にマイグレーションするための管理を容易にします。ユーザー表のユニーク索引のための `CREATE UNIQUE INDEX` ステートメントを生成します。

許可

sysadm

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

```
▶▶ db2uidl -d database-name -u table-schema -t table-name
▶▶ -o filename -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。

-u *table-schema*

処理する表のスキーマ (作成者ユーザー ID) を指定します。デフォルト・アクションでは、すべてのユーザー ID によって作成された表を処理します。

-t *table-name*

処理する表の名前。デフォルト・アクションでは、すべての表を処理します。

-o *filename*

出力を書き込むファイルの名前。デフォルト・アクションでは、出力は標準出力に書き込まれます。

-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。

使用上の注意

データベースに、バージョン 5 より前のバージョンの DB2 で稼働するデータベースで作成された索引がない場合、このツールを使用する必要はありません。このツールは、特定のタイプの名前を処理するようには設計されていません。特定の表名または表スキーマが、小文字、特殊文字、ブランクを含む区切り ID である場合、すべての表またはスキーマの処理を要求したほうがよいでしょう。出力結果は編集できます。

db2unins - DB2 データベース製品のアンインストール

1 つ以上の DB2 データベース製品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、すべてのオプションを使用できます。

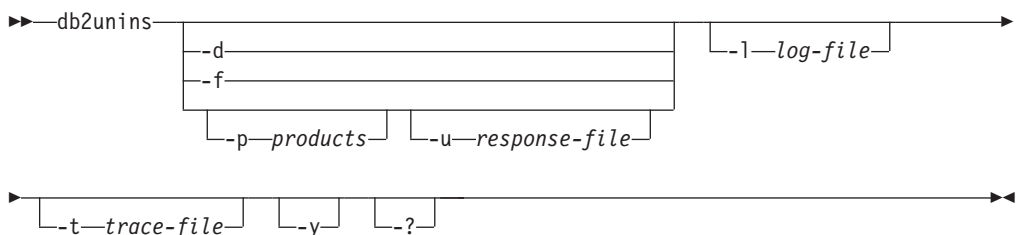
許可

sysadm

必要な接続

なし

コマンド構文



コマンド・パラメーター

-?, -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・メディアまたはマシン上のインストール・コピーのいずれかから発行できます。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べき製品を指定します。*products* は、DB2 データベース製品の省略形を二重引用符で囲んでセミコロンで区切ったリストです。例えば、-p "ESE;PE;QP" のようにします。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。

-u *response-file*

response-file に指定された内容に基づいて、アンインストールを実行します。また、このオプションはサイレント・アンインストールの実行にも使用され、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。

-l *log-file*

ログ・ファイルの位置を指定します。

-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

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、DB2COPY2 の db2unins -f を実行します。DB2COPY1 の db2unins -f を実行した場合、マシンが完全にきれいな状態にはなりません。

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 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、以下のどちらかを行います。

- 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。

db2xdbmig - XSR オブジェクトのマイグレーション・コマンド

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 9.5 へのマイグレーションを行う場合、MIGRATE DATABASE コマンドは自動的に db2xdbmig コマンドを実行します。このコマンドを DB2 バージョン 9.5 で実行する必要はありません。

db2xpert - 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) はデフォルトでインスタンス・ディレクトリ (DB2INSTPROF) にあるか、または **DIAGPATH** データベース・マネージャ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・パスにあります。これは *SQLLIB/BIN* ディレクトリの下にあります。db2xpert 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 ディレクトリに対するアクセスを持っている必要があります。

コマンド構文

▶▶—db2xpert—
 └─/p—*path*—/n—*infile*—
 └─/v— └─*outfile*—▶▶

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

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/doc/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。

許可

Root

必要な接続

なし

コマンド構文

```
▶▶ doce_deinstall --a [ -b installpath ] [ -l log-file ]
[ -t trace-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 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 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。

-h | **-?** 使用法情報を表示します。

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/V9 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。

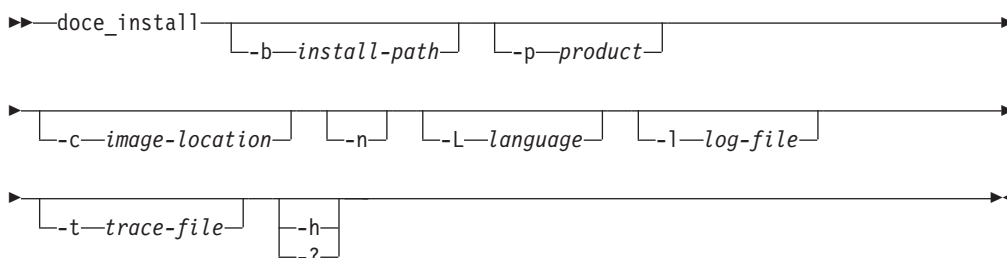
許可

Root

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/V9 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/v9.1` 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/v9.1 -n
```

enable_MQFunctions - WebSphere MQ 関数の有効化コマンド

指定したデータベースについて DB2 WebSphere MQ 関数を有効にし、DB2 WebSphere MQ 関数を正常に実行できるかどうかを検証します。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。

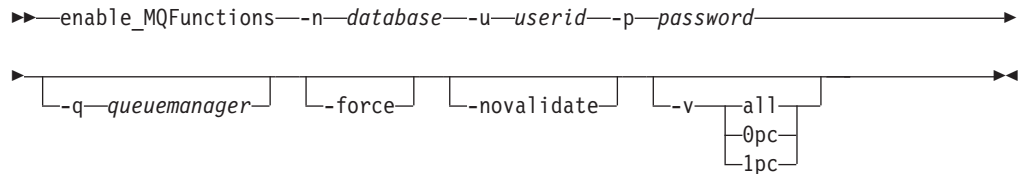
許可

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



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。

-q *queuemanager*

オプション。トランザクション MQ ユーザー定義関数をサポートするキュー・マネージャーの名前。名前を指定しない場合は、デフォルト・キュー・マネージャー DB2MQ_DEFAULT_MQM になります。このオプションを使用する場合、この関数は **-novalidate** パラメーターの使用を想定します。

-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。

-novalidate

オプション。DB2 MQSeries® 関数の検証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 *all*、*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。*all* 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 *all* オプションになります。

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 lpc
```

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。

installFixPack - インストール済み DB2 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 製品 (複数可) をイメージと同じレベルに更新します。マルチコピー DB2 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 DPF 内のすべてのノード) 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・パスに関連するそ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 プラットフォームでは、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。

DPF 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ノードでのみ必要ですが、フィックスパックのインストールはすべてのノードで行ってください。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ノードに、少なくともインスタンス所有ノードにインストールすることをお勧めします。

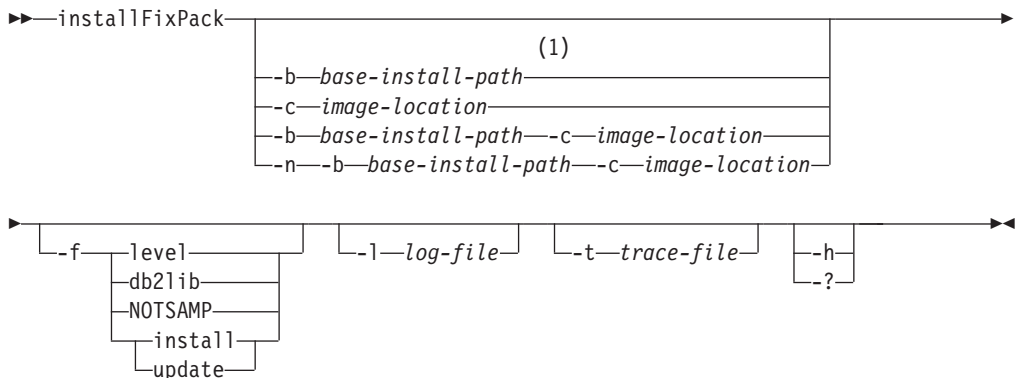
許可

ルート・インストールには `root` 権限が必要です。非ルート・インストールの場合、非ルート・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



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。指定する場合には、`-b`、または `-c` (あるいは両方) を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。

-b *base-install-path*

DB2 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/sql1lib 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/sql1lib 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 製品と同じサブディレクトリー上にもない。

-f 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 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

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 DPF インスタンスの全ノードも含まれる) で DBM を停止し、インストール・パスに関連する DB2ライブラリーすべてをアンロードする必要があります。

-f NOTSAMP

SA MP Base Component 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

DB2 ライブラリーのロード、インスタンス、および DAS が正しく停止しているかどうかについてのすべ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 DPF インスタンスの全ノードも含まれる) で DBM を停止し、インストール・パスに関連する DB2ライブラリーすべて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/DAS は更新されません。インストール後に、ユーザーが手動でインスタンス/DAS を更新しなければなりません。また、*update* オプションと *install* 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

DB2 ライブラリーのロード、インスタンス、および DAS が正しく停止し

ているかどうかについてのすべ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 DPF インスタンスの全ノードも含まれる) で DBM を停止し、インストール・パスに関連する DB2ライブラリーすべて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/DAS が更新されます。また、`update` オプションと `install` 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。

-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 Base Component (SA MP Base Component) 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。

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 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
```

setup - DB2 のインストール

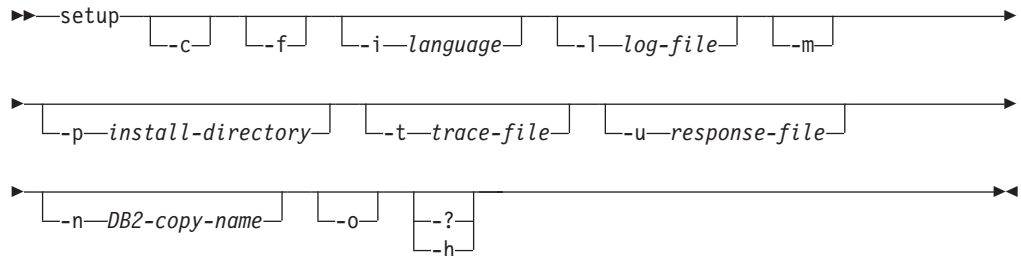
DB2 製品をインストールします。このコマンドは Windows オペレーティング・システムでのみ使用できます。UNIX オペレーティング・システム用のコマンドは `db2setup` です。

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 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・コマンド `setup` では、/ や - の切り替え記号を交換可能なものとして使用できます。

- c インストールの開始直後に、`setup.exe` が存在することを確認します。このオプションを選択すると、`setup.exe` の終了コードをモニターする場合にインストールの戻りコードが使用できなくなります。
- 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。

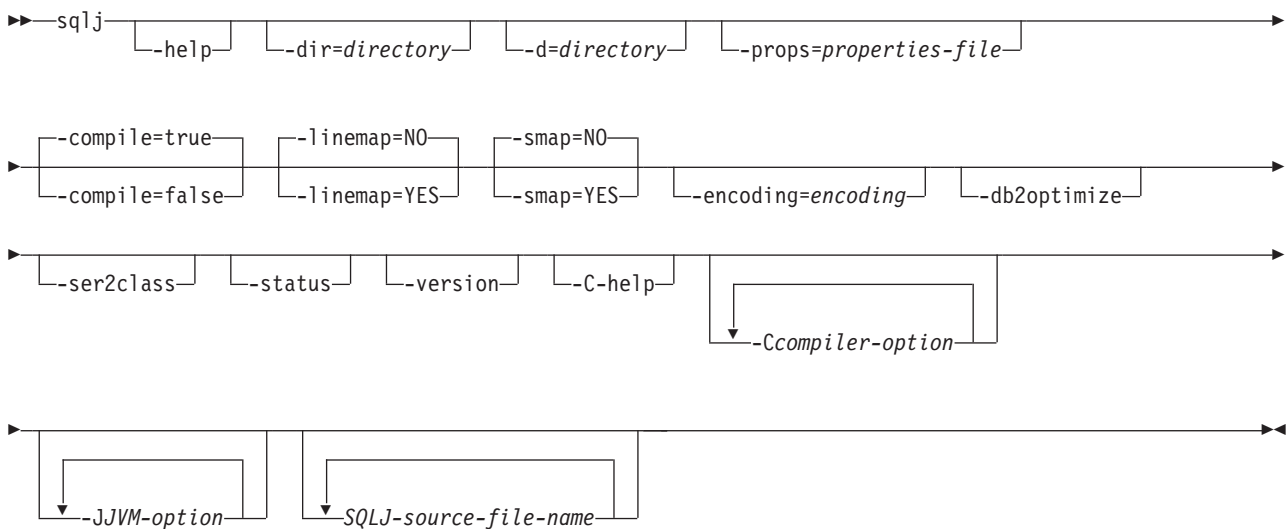
sqlj - SQLJ 変換プログラム

sqlj コマンドは SQLJ ソース・ファイルを Java ソース・ファイルおよび 0 個以上の SQLJ シリアライズド・プロファイルに変換します。デフォルトでは、sqlj コマンドは Java ソース・ファイルのコンパイルも行います。

許可

なし

コマンド構文



コマンド・パラメーター

-help

変換プログラムがサポートする各オプションが、SQLJ 変換プログラムによって説明されることを指定します。-help と共に他のオプションが指定されても、それらは無視されます。

-dir=directory

SQLJ が変換プログラムによって生成される .java ファイル、およびコンパイラによって生成される .class ファイルを入れるディレクトリーの名前を指定します。デフォルトは、SQLJ ソース・ファイルが入っているディレクトリーです。

変換プログラムは、生成ファイルをディレクトリーに配置するときに、SQLJ ソース・ファイルのディレクトリー構造を使用します。例えば、変換プログラムが次の 2 つのファイル进行处理するとします。

- Java パッケージにない file1.sqlj
- Java パッケージ sqlj.test にある file2.sqlj

また、変換プログラムを呼び出すときにパラメーター `-dir=/src` を指定するとします。変換プログラムは file1.sqlj 用の Java ソース・ファイルをディレクトリー `/src` に入れ、file2.sqlj 用の Java ソース・ファイルをディレクトリー `/src/sqlj/test` に入れます。

-d=directory

SQLJ が変換プログラムおよびコンパイラーによって生成されるバイナリー・ファイルを入れるディレクトリーの名前を指定します。これらのファイルには、.ser ファイル、name_SJProfileKeys.class ファイル、およびコンパイラーによって生成される .class ファイルが含まれます。

デフォルトは、SQLJ ソース・ファイルが入っているディレクトリーです。

変換プログラムは、生成ファイルをディレクトリーに配置するときに、SQLJ ソース・ファイルのディレクトリー構造を使用します。例えば、変換プログラムが次の 2 つのファイル进行处理するとします。

- Java パッケージにない file1.sqlj
- Java パッケージ sqlj.test にある file2.sqlj

また、変換プログラムを呼び出すときにパラメーター `-d=/src` を指定するとします。変換プログラムは file1.sqlj 用のシリアライズド・プロファイルディレクトリー `/src` に入れ、file2.sqlj 用のシリアライズド・プロファイルディレクトリー `/src/sqlj/test` に入れます。

-compile=truelfalse

SQLJ 変換プログラムが生成済みの Java ソースをバイトコードにコンパイルするかどうかを指定します。

true

変換プログラムは生成済みの Java ソース・コードをコンパイルします。これはデフォルトです。

false

変換プログラムは生成済みの Java ソース・コードをコンパイルしません。

-linemap=nolyes

Java 例外の行番号が、SQLJ ソース・ファイル (.sqlj ファイル) の行番号に一致するか、SQLJ 変換プログラムによって生成される Java ソース・ファイル (.java ファイル) の行番号に一致するかを指定します。

no Java 例外の行番号は Java ソース・ファイルの行番号に一致します。これはデフォルトです。

yes

Java 例外の行番号は SQLJ ソース・ファイルの行番号に一致します。

-smap=nolyes

SQLJ 変換プログラムが SQLJ ソース・ファイルごとにソース・マップ (SMAP) ファイルを生成するかどうかを指定します。SMAP ファイルはいくつかの Java 言語デバッグ・ツールで使用されます。このファイルは SQLJ ソース・ファイ

ルの行を、SQLJ 変換プログラムによって生成される Java ソース・ファイルの行にマップします。ファイルは Unicode UTF-8 コード化スキームです。その形式は、以下の Web サイトの Original Java Specification Request (JSR) 45 に説明されています。

<http://www.jcp.org>

no SMAP ファイルを生成しません。これはデフォルトです。

yes

SMAP ファイルを生成します。SMAP ファイルの名前は `SQLJ-source-file-name.java.smap` です。SQLJ 変換プログラムは SMAP ファイルを、生成された Java ソース・ファイルと同じディレクトリーに置きます。

-encoding=*encoding-name*

ソース・ファイルのエンコードを指定します。例として JIS または EUC があります。このオプションを指定しない場合、オペレーティング・システムのデフォルト・コンバーターが使用されます。

-db2optimize

SQLJ 変換プログラムが DB2 に合わせて最適化された接続コンテキスト・クラスのコードを生成することを指定します。-db2optimize は、デフォルト・コンテキストのコードではなく、ユーザー定義コンテキストのコードを最適化します。

SQLJ 変換プログラムを -db2optimize オプションを指定して実行する際に、アプリケーションが JDBC 3.0 またはそれ以前の関数を使用する場合、生成された Java アプリケーションをコンパイルするには IBM Data Server Driver for JDBC and SQLJ ファイル db2jcc.jar が CLASSPATH に入っていなければなりません。アプリケーションが JDBC 4.0 またはそれ以前の関数を使用する場合、生成された Java アプリケーションをコンパイルするには IBM Data Server Driver for JDBC and SQLJ ファイル db2jcc4.jar が CLASSPATH に入っていなければなりません。

-ser2class

SQLJ 変換プログラムが .ser ファイルを .class ファイルに変換することを指定します。

-status

SQLJ 変換プログラムが実行中に状況メッセージを表示することを指定します。

-version

SQLJ 変換プログラムが IBM Data Server Driver for JDBC and SQLJ のバージョンを表示することを指定します。情報の形式は次のとおりです。

IBM SQLJ xxxx.xxxx.xx

-C-help

SQLJ 変換プログラムが Java コンパイラーのヘルプ情報を表示することを指定します。

-C*compiler-option*

ダッシュ (-) で始まる有効な Java コンパイラー・オプションを指定します。-C とコンパイラー・オプションの間にはスペースを入れないでください。複数のコ

ンパイラー・オプションを指定する必要がある場合、各コンパイラー・オプションの前に `-C` を付けてください。以下に例を示します。

```
-C-g -C-verbose
```

すべてのオプションは Java コンパイラーに渡され、SQLJ 変換プログラムには使用されません。ただし、以下のオプションを除きます。

-classpath

SQLJ 変換プログラムおよび Java コンパイラーによって使用されるユーザー・クラス・パスを指定します。この値は CLASSPATH 環境変数をオーバーライドします。

-sourcepath

SQLJ 変換プログラムおよび Java コンパイラーがクラスまたはインターフェース定義を検索するソース・コード・パスを指定します。SQLJ 変換プログラムは、JAR または zip ファイル内ではなく、ディレクトリー内の `.sqlj` および `.java` ファイルのみ検索します。

-JVM-option

`sqlj` コマンドの実行場所である Java 仮想マシン (JVM) に渡されるオプションを指定します。オプションは、ダッシュ (-) で始まる有効な JVM オプションでなければなりません。-J と JVM オプションの間にはスペースを入れしないでください。複数の JVM オプションを指定する必要がある場合、各コンパイラー・オプションの前に -J を付けてください。以下に例を示します。

```
-J-Xmx128m -J-Xmine2M
```

SQLJ-source-file-name

変換される SQLJ ソース・ファイルのリストを指定します。これは必須パラメーターです。すべての SQLJ ソース・ファイル名には拡張子 `.sqlj` が付いていないければなりません。

出力

各ソース・ファイル `program-name.sqlj` ごとに、SQLJ 変換プログラムは以下のファイルを作成します。

- 生成済みソース・プログラム

生成済みソース・ファイルには `program-name.java` という名前が付けられます。

- SQLJ 実行可能節で使用される接続コンテキスト・クラスごとのシリアライズド・プロファイル・ファイル

シリアライズド・プロファイル名の形式は、以下のようになります。

```
program-name_SJProfileIDNumber.ser
```

- SQLJ 変換プログラムが Java コンパイラーを呼び出す場合、コンパイラーが生成するクラス・ファイル

例

```
sqlj -encoding=UTF8 -C-0 MyApp.sqlj
```


第 6 章 DB2 Text Search のコマンド

db2ts ALTER INDEX

このコマンドは、索引の更新特性を変更します。

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

テキスト検索索引が作成された表に対する CONTROL 特権は、データベース接続を確立するのに使用する許可 ID を持つユーザーによって保持される必要があります。

必要な接続

データベース

コマンド構文

▶▶ ALTER INDEX—*index-name*—FOR TEXT—| 更新特性 | | 接続オプション | ▶▶

更新特性:

| UPDATE FREQUENCY—NONE—| インクリメンタル更新特性 |
| | 更新頻度 |

更新頻度:

| D—(—*—) | H—(—*—) | M—(—*integer3*—) |
| | *integer1* | | *integer2* |

インクリメンタル更新特性:

| UPDATE MINIMUM—*minchanges*—|

接続オプション:

| CONNECT TO—*database-name*—|
| USER—*username*—USING—*password*—|

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、索引の更新がこれ以上行われなことを示します。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。また、ユーザーが索引を手動で更新する (UPDATE INDEX コマンドを使用して) つもりのときにも、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 から取られ、DEFAULTNAME='UPDATEFREQUENCY'となります。

NONE

自動更新は、テキスト索引に適用されません。今後の索引の更新は手動で開始しなければなりません。

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、頻度の設定は未変更のままです。

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、設定は未変更のままです。

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。競合するコマンドのいくつかは、以下のとおりです。

- ALTER INDEX
- CLEAR EVENTS FOR INDEX
- DROP INDEX
- UPDATE INDEX
- DISABLE DATABASE FOR TEXT
- STOP 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。

db2ts CLEANUP FOR TEXT

このコマンドは、廃止されたインスタンス内の DB2 Text Search コレクションをクリーンアップします。以下の場合に、コレクションが廃止されます。

- DB2 Text Search がデータベースで使用できなくなる前に、テキスト検索索引のあるデータベースがドロップされる場合。
- 表に関連付けられたテキスト検索索引をドロップする前に、表がドロップされる場合。

注: 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。テキスト検索コレクションとテキスト検索索引の間には 1 対 1 の関係があります。コマンドがテキスト検索索引に作動する一方、ツールはテキスト検索コレクションに作動します。SYSIBMTS.TSCOLLECTIONNAMES カタログ表を照会して、テキスト検索索引に対するテキスト検索コレクションを判別します。追加情報については、「DB2 Text Search の管理ツール」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

コマンドを正常に発行するには、ユーザーは現行インスタンスの DB2 インスタンス所有者でなければなりません。

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。このコマンドは、IBM Data Server Client からはサポートされていません。

コマンド構文

▶▶—CLEANUP FOR TEXT—◀◀

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。

活動状態ではない処理に関連付けられたすべてのロックが自動的にクリーンアップされます。これは、テキスト検索索引が新規の検索要求からアクセス可能になるために行われます。

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。

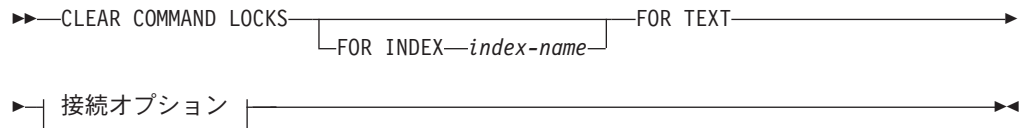
許可

索引名が指定されていない場合、データベース接続用の `username` は `dbadm` または `sysadm` 権限を持っている必要があります。特定の索引でコマンド・ロックをクリアする場合、データベース接続用の `username` が、テキスト検索索引が (DB2 カタログ・ビューに従って) 作成された表に対する CONTROL 特権を持っている必要があります。

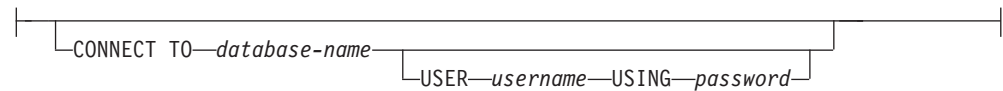
必要な接続

データベース

コマンド構文



接続オプション:



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。この場合、コマンド (ロックによって表される) は完了していない可能性があります。索引は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、CREATE 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。いくつかの索引カタログおよびコレクション情報は作成されましたが、すべてではありません。コマンド・ロックは、そのまま残されています。CREATE INDEX コマンド・ロックをクリアした後、DROP INDEX および CREATE INDEX コマンドを実行することができます。

このコマンドが発行される時、DB2 Text Search ビュー SYSIBMTS.TSLOCKS の内容が更新されます。

db2ts CLEAR EVENTS FOR INDEX

このコマンドは、管理用で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 つの文書进行处理する索引の更新操作はそれぞれ、情報項目と、ある場合にはエラー項目をイベント表に生成します。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

テキスト検索索引が作成された表に対する CONTROL 特権は、データベース接続を確立するのに使用する許可 ID を持つユーザーによって保持される必要があります。

必要な接続

データベース

コマンド構文

▶▶—CLEAR EVENTS FOR INDEX—*index-name*—FOR TEXT— | 接続オプション | ▶▶▶▶

接続オプション:

|
└──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。競合するコマンドのいくつかは、以下のとおりです。

- CLEAR EVENTS FOR INDEX
- UPDATE INDEX
- ALTER INDEX
- DROP INDEX
- DISABLE DATABASE FOR TEXT
- STOP 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

テキスト列に対するテキスト検索索引を作成します。これによって、テキスト検索関数を使用して列データを検索できるようになります。

テキスト検索 UPDATE INDEX コマンドがユーザーによって明示的に実行されるか、または索引に対して定義された更新頻度に従ってテキスト検索サービスによって暗黙的に発行されるまで、索引にはデータが何も含まれていません。

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

現行のデータベース接続を確立した許可 ID によってテキスト検索索引が作成される表に対して、CONTROL 特権が必要です。

必要な接続

データベース

コマンド構文

```
▶▶ CREATE INDEX index-name FOR TEXT ON table-name
▶ ( text-column-name )
  ( function-name ( text-column-name ) )
▶ | テキストのデフォルト情報 | | 更新特性 | | ストレージ・オプション |
▶ | 索引構成オプション | | 接続オプション |
```

テキストのデフォルト情報:

```
| CODEPAGE code-page | LANGUAGE locale | FORMAT format |
```

更新特性:

```
| UPDATE FREQUENCY NONE | インクリメンタル更新特性 |
| | 更新頻度 |
```

更新頻度:

```
| D ( * ) | H ( * ) | M ( integer3 ) |
| | integer1 | | integer2 |
```

インクリメンタル更新特性:

```
| UPDATE MINIMUM minchanges |
```

ストレージ・オプション:

```
| COLLECTION DIRECTORY directory |
```

```
ADMINISTRATION TABLES IN tablespace-name
```

索引構成オプション:

```
INDEX CONFIGURATION (option-value)
```

接続オプション:

```
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。

- 範囲パーティション表
- 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、LONG VARCHAR、CLOB、DBCLOB、BLOB、GRAPHIC、VARGRAPHIC、LONG 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 にある値によって指定されます。ここで、DEFAULTNAME='CODEPAGE'(ここではデータベースのコード・ページ) です。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。つまり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、データベースのテリトリーが、LANGUAGE のデフォルト設定を判別するのに使用されます。文書を自動的にスキャンしてロケールを判別する場合、*locale* を AUTO として指定します。

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、および HTML 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 にあります。ここで、DEFAULTNAME='FORMAT' 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。

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、索引の更新がこれ以上行われなことを示します。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。また、ユーザーが索引を手動で更新する (UPDATE INDEX コマンドを使用して) つもりのときにも、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 から取られ、DEFAULTNAME='UPDATEFREQUENCY'となります。

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 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る変更の最小数を指定します。正の整数値のみ使用できます。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。しかし、それはパフォーマンス・オーバーヘッドがより大きくなる原因ともなります。

COLLECTION DIRECTORY *directory*

テキスト検索索引が保管されるディレクトリー。デフォルトでは、コレクション・データは DBPATH/NODExxxx/SQLxxxx/db2collections/index *identifier*/data に置かれます。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。

ADMINISTRATION TABLES IN *tablespace-name*

索引用に作成された管理表の既存の REGULAR 表スペースの名前を指定します。指定しない場合、索引が作成されている基本表の表スペースが使用されます。

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。これらの値は単一引用符で囲まれなければなりません。

注: 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。以下の値がサポートされています。

表 39. オプション値の仕様

オプション	許可されている値 (デフォルト)	意味
COMMENT	512 バイトより短いストリング値。	DB2 Text Search のカタログ・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。

以下に例を示します。

```
INDEX CONFIGURATION (COMMENT 'Index on User''s Guide column')
```

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。
- 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。
- DB2 データベース・サーバー上の Text Search の索引データ・ファイルが更新されます。スケジュールされた更新情報は、インスタンス内の各索引ごとに記録されます。

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。競合するコマンドのいくつかは、以下のとおりです。

- DROP INDEX

- UPDATE INDEX
- CLEAR EVENTS FOR INDEX
- ALTER INDEX
- DISABLE DATABASE FOR TEXT
- STOP FOR TEXT

db2ts DISABLE DATABASE FOR TEXT

このコマンドは、コマンド `ENABLE DATABASE FOR TEXT` によってなされた変更を逆転させます (例えば、テキスト検索に関連した表およびビューのドロップ)。

このコマンドが発行される時、以下のことを行います。

- データベースで DB2 Text Search フィーチャーを使用不可にします。
- 次のような表およびビューを削除します。
 - SYSIBMTS.TSDEFAULTS
 - SYSIBMTS.TSLOCKS
 - SYSIBMTS.TSINDEXES
 - SYSIBMTS.TSCONFIGURATION
 - SYSIBMTS.TSCOLLECTIONNAMES

表は、データベースのデフォルト表スペース (IBMDEFAULTGROUP) から除去されます。

- `FORCE` オプションが指定されている場合、すべてのテキスト索引情報がデータベースから除去され、すべての関連するコレクションが削除されます。さらに、テキスト・サービスが更新され、残りの更新スケジュール情報が除去されます。『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

データベース接続用の `username` は `dbadm` または `sysadm` 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

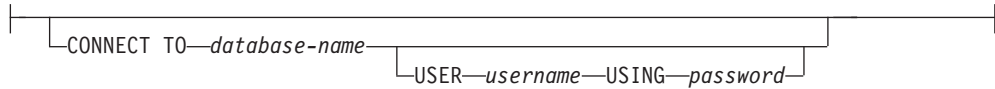
コマンド構文

```

▶▶—DISABLE DATABASE FOR TEXT—FORCE | 接続オプション |————▶▶

```

接続オプション:



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、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。

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.log に書き込まれます。CLP コマンド DROP DATABASE の前に、テキスト検索索引コマンド DISABLE DATABASE FOR TEXT が実行されない場合、テキスト検索索引サービスも CLEANUP FOR TEXT コマンドを使用してクリーンアップされる必要があります。索引のドロップについての詳細は DROP INDEX コマンドを参照し、テキスト検索コレクションおよびコレクションとテキスト検索索引との関連については CLEANUP FOR TEXT コマンドを参照してください。

注: ユーザーは、孤立したコレクション、つまりテキスト検索サーバーで定義されたまま、DB2 によって使用されないコレクションとなるような使用法は勧められていません。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` コマンドまたは `DROP TABLE` ステートメントを実行するとき。
- `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。`CLEANUP FOR TEXT` コマンドは、いくつかのシナリオで使用することができます。

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。競合するコマンドのいくつかは、以下のとおりです。

- `DROP INDEX`
- `UPDATE INDEX`
- `CLEAR EVENTS FOR INDEX`
- `ALTER INDEX`
- `DISABLE DATABASE FOR TEXT`
- `STOP FOR TEXT`

db2ts DROP INDEX

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

以下のいずれか。

- `dbadm` または `sysadm`: この権限は、`FORCE` オプションを持つ `DISABLE DATABASE FOR TEXT` コマンドの一部として、テキスト索引をドロップするのに使用されます。
- テキスト検索索引が作成された表に対する `CONTROL` 特権は、データベース接続を確立するのに使用する許可 ID を持つユーザーによって保持される必要があります。

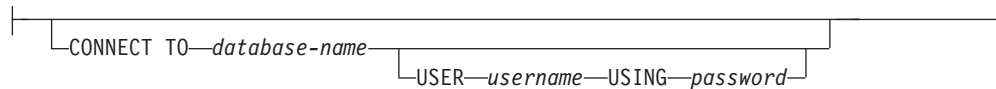
必要な接続

データベース

コマンド構文

▶▶ `DROP INDEX—index-name—FOR TEXT—` | 接続オプション | ◀◀

接続オプション:



コマンド・パラメーター

DROP 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。

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

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。競合するコマンドのいくつかは、以下のとおりです。

- DROP INDEX
- UPDATE INDEX
- CLEAR EVENTS FOR INDEX
- ALTER INDEX
- DISABLE DATABASE FOR TEXT
- STOP FOR TEXT

DB2 でのユーザー表のドロップは、テキスト検索索引のドロップを引き起こしません。表のドロップの前または後 (表のドロップの前が望ましい) に、手動でそれらをドロップする必要があります。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。

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。
- 索引のステージング/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。

- SYSIBMTS.TSDEFAULTS
- SYSIBMTS.TSLOCKS
- SYSIBMTS.TSINDEXES
- SYSIBMTS.TSCONFIGURATION
- SYSIBMTS.TSCOLLECTIONNAMES

表は、データベースのデフォルト表スペース (IBMDEFAULTGROUP) に作成されます。

DB2 サーバー上のファイル・システムへの変更: なし。

db2ts HELP

HELP は、使用可能な DB2 テキスト・サーチ・コマンドのリスト、または個別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:

- ENABLE
- DISABLE
- CREATE
- DROP
- ALTER
- UPDATE
- CLEAR (CLEAR COMMAND LOCKS と CLEAR EVENTS FOR INDEX の両方)
- CLEANUP

- START
- STOP

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、CIE##### の形式になります。##### 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
```

注: 以下の Text Search コマンドでは、リストされるコマンド・ヘルプ・オプションは現時点では無効です。

db2ts DISABLE DATABASE FOR TEXT
[SERVER]

db2ts ENABLE DATABASE FOR TEXT
[SERVER]

db2ts UPDATE INDEX
[REORGANIZE] および [PARSE ONLY]

db2ts START FOR TEXT

このコマンドは以下の事柄を行います。

- 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。
- DB2 データベース・サーバーを実行するホスト・マシン上のサービス (UNIX 上ではデーモン) を開始します。DB2 データベース・サーバー上でのテキスト検索索引の更新のスケジューリングと、テキスト検索エンジンの処理は、これらのサービスが行います。インスタンス・サービスは DB2 インスタンス所有者の許可名の下で開始されます。インスタンス・サービスがすでに実行されている場合、コマンドには何の効果もありません。

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

コマンドを正常に発行するには、ユーザーは現行インスタンスの DB2 インスタンス所有者でなければなりません。

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。このコマンドは、IBM Data Server Client からはサポートされていません。

コマンド構文

▶—START FOR TEXT—▶

コマンド・パラメーター

なし

db2ts STOP FOR TEXT

このコマンドは、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

コマンドを正常に発行するには、ユーザーは現行インスタンスの DB2 インスタンス所有者でなければなりません。

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。このコマンドは、IBM Data Server Client からはサポートされていません。

コマンド構文

▶—STOP FOR TEXT—▶

コマンド・パラメーター

なし

使用上の注意

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。現在実行中のコマンドの実行を中断しないために、この停止コマンドを発行する前に、UPDATE INDEX のような他の管理コマンドがアクティブでなくなっていることを必ず確認してください。競合するコマンドのいくつかは、以下のとおりです。

- DROP INDEX
- UPDATE INDEX
- CLEAR EVENTS FOR INDEX
- ALTER INDEX
- DISABLE DATABASE FOR TEXT
- STOP FOR TEXT

db2ts UPDATE INDEX

この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、検索は部分的に更新された索引上で作動します。

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

テキスト検索索引が作成された表に対する CONTROL 特権は、データベース接続を確立するのに使用する許可 ID を持つユーザーによって保持される必要があります。

必要な接続

データベース

コマンド構文

```
▶▶ UPDATE INDEX index-name FOR TEXT USING UPDATE MINIMUM
```

▶ | 接続オプション | ▶▶

接続オプション:

```
| 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。

USING UPDATE MINIMUM

索引を作成するのに使用される CREATE INDEX コマンドで指定された UPDATE MINIMUM *minchange* 設定値を使用するように指定し、指定された最小数の変更が生じた場合に、インクリメンタル更新を開始します。デフォルトでは、無条件に更新を開始します。

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@#$_ ]*"
```

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。所要時間は、索引付けされる文書の数と、すでに索引付けされた文書の数によって決まります。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。競合するコマンドのいくつかは、以下のとおりです。

- UPDATE INDEX
- CLEAR EVENTS FOR INDEX
- ALTER INDEX
- DROP INDEX
- DISABLE DATABASE FOR TEXT
- STOP 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。エラーのある文書の主キーは、索引のイベント表で検索できます。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、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tring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、インポート、およびロード・ユーティリティー用のファイル・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。

135 ページの『EXPORT』

204 ページの『IMPORT』

288 ページの『LOAD』

データ移動のための区切り文字の考慮事項

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tring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。例えば、


```
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String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String区切り文字が見つかると、そのすべての出現箇所が二重のString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、String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。**nodoublede1**

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String区切り文字の動作を無効にするには、**nodoublede1** ファイル・タイプ修飾子を指定します。ただし、二重のString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 **nodoublede1** を使用するとき、String区切り文字が文字フィールドにあっても、そのString区切り文字は二重になりません。インポートとロードで **nodoublede1** を使用すると、二重のString区切り文字は、String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String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、String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String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、**colde1** 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、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)
 - DB2 ツールのヘルプ
 - 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://www.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://www.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 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。

IBM Information Management 製品の使用をより容易にするために IBM にご協力いただける場合は、コンシューマビリティに関する次のアンケートにご回答ください。<http://www.ibm.com/software/data/info/consumability-survey/>

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、DB2 ライブラリーについて説明しています。DB2 ライブラリーに関する詳細な説明については、www.ibm.com/shop/publications/order にある IBM Publications Center にアクセスしてください。英語の DB2 バージョン 9.5 のマニュアル (PDF 形式) とその翻訳版は、www.ibm.com/support/docview.wss?rs=71&uid=swg2700947 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 40. 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	SC88-4431-02	入手可能	2009 年 4 月
管理ルーチンおよびビュー	SC88-4435-02	入手不可	2009 年 4 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	SC88-4433-02	入手可能	2009 年 4 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	SC88-4434-02	入手可能	2009 年 4 月
コマンド・リファレン ス	SC88-4432-02	入手可能	2009 年 4 月
データ移動ユーティリ ティー ガイドおよびリ ファレンス	SC88-4421-02	入手可能	2009 年 4 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	SC88-4423-02	入手可能	2009 年 4 月
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	SC88-4259-02	入手可能	2009 年 4 月
データベース・セキュ リティー・ガイド	SC88-4418-02	入手可能	2009 年 4 月
ADO.NET および OLE DB アプリケーション の開発	SC88-4425-02	入手可能	2009 年 4 月

表 40. 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
組み込み SQL アプリケーションの開発	SC88-4426-02	入手可能	2009 年 4 月
Java アプリケーションの開発	SC88-4427-02	入手可能	2009 年 4 月
Perl および PHP アプリケーションの開発	SC88-4428-02	入手不可	2009 年 4 月
SQL および外部ルーチンの開発	SC88-4429-02	入手可能	2009 年 4 月
データベース・アプリケーション開発の基礎	GC88-4430-02	入手可能	2009 年 4 月
DB2 インストールおよび管理 概説 (Linux および Windows 版)	GC88-4439-02	入手可能	2009 年 4 月
国際化対応ガイド	SC88-4420-02	入手可能	2009 年 4 月
メッセージ・リファレンス 第 1 巻	GI88-4109-01	入手不可	2009 年 4 月
メッセージ・リファレンス 第 2 巻	GI88-4110-01	入手不可	2009 年 4 月
マイグレーション・ガイド	GC88-4438-02	入手可能	2009 年 4 月
Net Search Extender 管理およびユーザズ・ガイド	SC88-4630-02	入手可能	2009 年 4 月
パーティションおよびクラスタリングのガイド	SC88-4419-02	入手可能	2009 年 4 月
Query Patroller 管理およびユーザズ・ガイド	SC88-4611-01	入手可能	2009 年 4 月
IBM データ・サーバー・クライアント機能概説およびインストール	GC88-4441-02	入手不可	2009 年 4 月
DB2 サーバー機能 概説およびインストール	GC88-4440-02	入手可能	2009 年 4 月
Spatial Extender および Geodetic Data Management Feature ユーザズ・ガイドおよびリファレンス	SC88-4629-02	入手可能	2009 年 4 月
SQL リファレンス 第 1 巻	SC88-4436-02	入手可能	2009 年 4 月
SQL リファレンス 第 2 巻	SC88-4437-02	入手可能	2009 年 4 月

表 40. 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
システム・モニター ガ イドおよびリファレン ス	SC88-4422-02	入手可能	2009 年 4 月
<i>Text Search</i> ガイド	SC88-4424-01	入手可能	2009 年 4 月
問題判別ガイド	GI88-4108-02	入手不可	2009 年 4 月
データベース・パフォー マンスのチューニン グ	SC88-4417-02	入手可能	2009 年 4 月
<i>Visual Explain</i> チュー トリアル	SC88-4449-00	入手不可	
新機能	SC88-4445-02	入手可能	2009 年 4 月
ワークロード・マネー ジャー ガイドおよびリ ファレンス	SC88-4446-02	入手可能	2009 年 4 月
<i>pureXML</i> ガイド	SC88-4447-02	入手可能	2009 年 4 月
<i>XQuery</i> リファレンス	SC88-4448-02	入手不可	2009 年 4 月

表 41. 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
DB2 Connect <i>Personal</i> <i>Edition</i> 概説およびイン ストール	GC88-4443-02	入手可能	2009 年 4 月
DB2 Connect サーバー 機能 概説およびインス トール	GC88-4444-02	入手可能	2009 年 4 月
DB2 Connect ユーザー ズ・ガイド	SC88-4442-02	入手可能	2009 年 4 月

表 42. Information Integration 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>Information Integration</i> : フェデレーテッド・シ ステム 管理ガイド	SC88-4166-01	入手可能	2008 年 3 月
<i>Information Integration</i> : レプリケーションおよ びイベント・パブリッ シングのための <i>ASNCLP</i> プログラム・ リファレンス	SC88-4167-02	入手可能	2008 年 3 月

表 42. Information Integration 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
Information Integration: フェデレーテッド・データ・ソース 構成ガイド	SC88-4185-01	入手不可	
Information Integration: SQL レプリケーションガイドおよびリファレンス	SC88-4168-01	入手可能	2008 年 3 月
Information Integration: レプリケーションとイベント・パブリッシング 概説	GC88-4187-01	入手可能	2008 年 3 月

DB2 の印刷資料の注文方法

DB2 の印刷資料が必要な場合、オンラインで購入することができますが、すべての国および地域で購入できるわけではありません。DB2 の印刷資料については、IBM 営業担当員にお問い合わせください。DB2 PDF ドキュメンテーション DVD の一部のソフトコピー・ブックは、印刷資料では入手できないことに留意してください。例えば、「DB2 メッセージ・リファレンス」はどちらの巻も印刷資料としては入手できません。

DB2 PDF ドキュメンテーション DVD で利用できる DB2 の印刷資料の大半は、IBM に有償で注文することができます。国または地域によっては、資料を IBM Publications Center からオンラインで注文することもできます。お客様の国または地域でオンライン注文が利用できない場合、DB2 の印刷資料については、IBM 営業担当員にお問い合わせください。DB2 PDF ドキュメンテーション DVD に収録されている資料の中には、印刷資料として提供されていないものもあります。

注: 最新で完全な DB2 資料は、DB2 インフォメーション・センター (<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5>) で参照することができます。

DB2 の印刷資料は以下の方法で注文することができます。

- 日本 IBM 発行のマニュアルはインターネット経由でご購入いただけます。詳しくは <http://www.ibm.com/shop/publications/order> の「ご注文について」をご覧ください。資料の注文情報にアクセスするには、お客様の国、地域、または言語を選択してください。その後、各ロケーションにおける注文についての指示に従ってください。
- DB2 の印刷資料を IBM 営業担当員に注文するには、以下のようになります。
 1. 以下の Web サイトのいずれかから、営業担当員の連絡先情報を見つけてください。
 - IBM Directory of world wide contacts (www.ibm.com/planetwide)
 - IBM Publications Web サイト (<http://www.ibm.com/shop/publications/order>)
国、地域、または言語を選択し、お客様の所在地に該当する Publications ホ

ーム・ページにアクセスしてください。このページから、「このサイトについて」のリンクにアクセスしてください。

2. 電話をご利用の場合は、DB2 資料の注文であることをご指定ください。
3. 担当者に、注文する資料のタイトルと資料番号をお伝えください。タイトルと資料番号は、942 ページの『DB2 テクニカル・ライブラリー (ハードコピーまたは PDF 形式)』でご確認いただけます。

コマンド行プロセッサから SQL 状態ヘルプを表示する

DB2 は、SQL ステートメントの結果の原因になったと考えられる条件の 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。

SQL 状態ヘルプを呼び出すには、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 or ? 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

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 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、バージョン 8 のインフォメーション・センターの URL <http://publib.boulder.ibm.com/infocenter/db2luw/v8/>にアクセスしてください。

DB2 インフォメーション・センターでの希望する言語でのトピックの表示

DB2 インフォメーション・センターでは、ブラウザの設定で指定した言語でのトピックの表示が試みられます。トピックがその指定言語に翻訳されていない場合は、DB2 インフォメーション・センターでは英語でトピックが表示されます。

- Internet Explorer Web ブラウザーで、指定どおりの言語でトピックを表示するには、以下のようにします。
 1. Internet Explorer の「ツール」->「インターネット オプション」->「言語...」ボタンをクリックします。「言語の優先順位」ウィンドウがオープンします。
 2. 該当する言語が、言語リストの先頭の項目に指定されていることを確認します。
 - リストに新しい言語を追加するには、「追加...」ボタンをクリックします。

注: 言語を追加しても、特定の言語でトピックを表示するのに必要なフォントがコンピューターに備えられているとはかぎりません。

- リストの先頭に新しい言語を移動するには、その言語を選択してから、その言語が言語リストに先頭に行くまで「上に移動」ボタンをクリックします。
- 3. ブラウザー・キャッシュを消去してから、ページを最新表示します。希望する言語で DB2 インフォメーション・センターが表示されます。
- Firefox または Mozilla Web ブラウザーの場合に、希望する言語でトピックを表示するには、以下のようになります。
 1. 「ツール」->「オプション」->「詳細」ダイアログの「言語」セクションにあるボタンを選択します。「設定」ウィンドウに「言語」パネルが表示されます。
 2. 該当する言語が、言語リストの先頭の項目に指定されていることを確認します。
 - リストに新しい言語を追加するには、「追加...」ボタンをクリックしてから、「言語を追加」ウィンドウで言語を選択します。
 - リストの先頭に新しい言語を移動するには、その言語を選択してから、その言語が言語リストに先頭に行くまで「上に移動」ボタンをクリックします。
 3. ブラウザー・キャッシュを消去してから、ページを最新表示します。希望する言語で DB2 インフォメーション・センターが表示されます。

ブラウザとオペレーティング・システムの組み合わせによっては、オペレーティング・システムの地域の設定も希望のロケールと言語に変更しなければならない場合があります。

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。

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。非管理者および非 root の DB2 インフォメーション・センター 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。インストールする更新がある場合は、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は、

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する必要があります。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 を再始動します。

注: Windows Vista の場合、下記の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・ツールを起動するには、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。

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/db2icdv95 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。*DB2* インフォメーション・センター は、デフォルトで *Program_files¥IBM¥DB2 Information Center¥Version 9.5* ディレクトリーにインストールされます。ここで、*Program_files* は Program Files ディレクトリーのロケーションを表します。

- 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/V9.5* 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 ブラウザーが起動し、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。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 (Find Updates)」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終了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終了しないでください。

- 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終了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 スクリプトを終了しないでください。

7. DB2 インフォメーション・センター を再始動します。

- 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/db2icdv95 start
```

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 製品のさまざまな機能について学習するのを支援します。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/db2help/>) 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。

「*Visual Explain* チュートリアル」の『**Visual Explain**』

Visual Explain を使用して、パフォーマンスを向上させるために SQL ステートメントを分析し、最適化し、調整します。

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 ドキュメンテーション

トラブルシューティング情報は、「DB2 問題判別ガイド」、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。ここでは、DB2 診断ツールおよびユーティリティを使用して、問題を切り分けて識別する方法、最も頻繁に起こる幾つかの問題に対するソリューションについての情報、および DB2 データベース製品を使用する際に発生する可能性のある問題の解決方法についての他のアドバイスがあります。

DB2 Technical Support の Web サイト

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検索したい場合は、DB2 Technical Support の Web サイト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。

DB2 Technical Support の Web サイト (http://www.ibm.com/software/data/db2/support/db2_9/) 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。

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。

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。

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。

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。

〒106-8711
東京都港区六本木 3-2-12
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。

本書は、IBM 以外の Web サイトおよびリソースへのリンクまたは参照を含む場合があります。IBM は、本書より参照もしくはアクセスできる、または本書からリンクされた IBM 以外の Web サイトもしくは第三者のリソースに対して一切の責任を負いません。IBM 以外の Web サイトにリンクが張られていることにより IBM が当該 Web サイトを推奨するものではなく、またその内容、使用もしくはサイトの所有者について IBM が責任を負うことを意味するものではありません。また、IBM は、お客様が IBM Web サイトから第三者の存在を知ることになった場合にも (もしくは、IBM Web サイトから第三者へのリンクを使用した場合にも)、お客様と第三者との間のいかなる取引に対しても一切責任を負いません。従って、お客様は、IBM が上記の外部サイトまたはリソースの利用について責任を負うものではなく、また、外部サイトまたはリソースからアクセス可能なコンテンツ、サービス、

製品、またはその他の資料一切に対して IBM が責任を負うものではないことを承諾し、同意するものとします。第三者により提供されるソフトウェアには、そのソフトウェアと共に提供される固有の使用条件が適用されます。

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
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
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。

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、ibm.com は、International Business Machines Corporation 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。現時点での IBM の商標リストについては、www.ibm.com/legal/copytrade.shtml の「Copyright and trademark information」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、Sun Microsystems, Inc. の米国およびその他の国における商標です。
- 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。
- Intel、Intel (ロゴ)、Intel Inside、Intel Inside (ロゴ)、Intel Centrino、Intel Centrino (ロゴ)、Celeron、Intel Xeon、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。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

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 474
アドバイザー
設計アドバイザー 572
db2advis 572
アンカタログ
システム・データベース・ディレクトリー 519
データベース項目 519
ホスト DCS データベース項目 520
暗黙接続 1
異常終了
再始動コマンド 439
一時ファイル
LOAD コマンド 288
移動, データの
区切り文字に関する制約事項 937
データベース間の 204
イベント・アナライザー・コマンド 654
イベント・モニターのターゲット表定義の生成コマンド 656
イベント・モニター・コマンド 655
イメージ 704
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・コマンド 756
インスタンス作成インターフェースの開始コマンド 705
インスタンスの更新コマンド 707
インスタンスの作成コマンド 691
インスタンスの自動始動コマンド 688
インスタンスの除去コマンド 696
インスタンスのマイグレーション・コマンド 699
インスタンスのリスト・コマンド 698
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
追加コマンド 754
インストール済み DB2 製品の更新コマンド 901
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 729
インストール・イメージ
イメージ・サイズの削減 704
インポート
データ 204
エクスポート
データ
ファイル・タイプ修飾子 135
EXPORT コマンド 135

エラー・メッセージ
チェックサム
データベース構成ファイル 434, 539
データベース・マネージャー構成ファイル 432
データベース構成ファイル 162
リモート・データベースのドロップ
DROP DATABASE コマンド 129
応答ファイル
生成プログラム
db2rspgn 838
オンライン検査
カスタマイズが必要 854
最適化の向上のため 854
制約事項 854

[カ行]

カーソル固定 (CS)
変更 93
カーネル・パラメーター値のためのユーティリティー・コマン
ド 760
開始
DB2
db2start コマンド 867
カタログ作成
データベース 75
ホスト・データベース 78
環境変数
DB2OPTIONS 8
監査機能管理者ツール・コマンド 580
行継続文字
コマンド行プロセッサ (CLP) 1
強調表示規則 viii
区切り文字
データ移動時の制約事項 937
変更 937
組み込みランタイム・ライブラリー検索パス
変更 613
継続文字
コマンド行プロセッサ (CLP) 1
現行 DAS レベルの表示コマンド 632
検索パスの変更コマンド 615
検査結果のフォーマット・コマンド 703
コード・ページ
EXPORT コマンド 135
IMPORT コマンド 204
コード・ページ・ファイル・タイプ修飾子 288
更新
DAS の更新コマンド 556
DB2 インフォメーション・センター 947

構成

管理

リセット、デフォルトに 432

例 149

データベース・マネージャ

例 167

CLI

例 157

database

更新 539

リセット、デフォルトに 434

例 162

構成アシスタント (CA)

開始 604

構文 911, 913, 914, 916, 917, 923, 925, 927, 928, 930, 931

コマンド行プロセッサ SQL ステートメント用 19

説明 ix

コマンド

構文ヘルプ 27

出力のリダイレクト 1

ヘルプの呼び出し 18

ACTIVATE DATABASE 31

ADD CONTACT 32

ADD CONTACTGROUP 33

ADD DBPARTITIONNUM 34

ARCHIVE LOG 38

ATTACH 40

AUTOCONFIGURE 42

BACKUP DATABASE 45

BIND 55

CATALOG DATABASE 75

CATALOG DCS DATABASE 78

CATALOG LDAP DATABASE 80

CATALOG LDAP NODE 83

CATALOG LOCAL NODE 84

CATALOG NAMED PIPE NODE 85

CATALOG ODBC DATA SOURCE 87

CATALOG TCP/IP NODE 88

CHANGE DATABASE COMMENT 92

CHANGE ISOLATION LEVEL 93

CREATE DATABASE 96

CREATE TOOLS CATALOG 117

dasauto 553

dasert 553

dasdrop 554

dasmigr 555

dasupdt 556

db2 7

DB2 JDBC パッケージ・バインド・プログラム・ユーティ

リティー 710

db2acsutil 599

db2admin 563

db2adutl 565

db2advis 572

db2audit 580

db2batch 589

コマンド (続き)

db2bfd 603

db2cap 605

db2cc 608

db2cfexp 610

db2cfimp 612

db2chglbpath 613

db2chgpath 615

db2ckbcp 616

db2ckmig 620

db2ckrst 622

db2cli 624

db2cmd 625

db2cptsa 626

db2dart 628

db2daslevel 632

db2dclgn 633

db2diag 635

db2drdat 650

db2drvmp 651

db2empfa 652

db2envar.bat 653

db2eva 654

db2evmon 655

db2evtbl 656

db2exmig 660

db2expln 661

db2extsec 667

db2flsn 669

db2fm 671

db2fmcu 673

db2fodc 674

db2fs 680

db2gcf 681

db2gov 683

db2govlg 685

db2gpmap 686

db2hc 687

db2iauto 688

db2iclus 688

db2icrt 691

db2idrop 696

db2ilist 698

db2imigr 699

db2inidb 701

db2inspf 703

db2iprune 704

db2isetup 705

db2iupdt 707

db2jdbcbind 710

db2ldcfg 712

db2level 713

db2licm 714

db2listvolumes 716

db2logsforrwd 717

db2look 717

コマンド (続き)

db2ls 729
 db2move 731
 db2mqdsn 741
 db2mscs 745
 db2mtrk 749
 db2nchg 753
 db2ncrt 754
 db2ndrop 756
 db2nrcfg 757
 db2nrupdt 760
 db2osconf 760
 db2pd 763
 db2pdcfg 824
 db2perfc 828
 db2perfi 830
 db2perfr 830
 db2rbind 831
 db2relocatedb 833
 db2rfe 758
 db2sampl 839
 db2schex 842
 db2set 843
 db2setup 847
 db2sqljbind 848
 db2sqljprint 867
 db2start 867
 db2stat 868
 db2stop 869
 db2support 869
 db2swtch 875
 db2sync 876
 db2systray 877
 db2tapemgr 879
 db2tbst 882
 db2tdbmgr 883
 db2trc 884
 db2ts ALTER INDEX 911
 db2ts CLEANUP FOR TEXT 913
 db2ts CLEAR COMMAND LOCKS 914
 db2ts CLEAR EVENTS FOR INDEX 916
 db2ts CREATE INDEX 917
 db2ts DISABLE DATABASE FOR TEXT 923
 db2ts DROP INDEX 925
 db2ts ENABLE DATABASE FOR TEXT 927
 db2ts HELP 928
 db2ts START FOR TEXT 930
 db2ts STOP FOR TEXT 930
 db2ts UPDATE INDEX 931
 db2uiddl 891
 db2unins 892
 db2untag 894
 db2xdbmig 895
 db2xprt 895
 db2_deinstall 557
 db2_install 559

コマンド (続き)

db2_local_ps 562
 DEACTIVATE DATABASE 119
 DECOMPOSE XML DOCUMENT 120
 DEREGISTER 122
 DESCRIBE 123
 DETACH 128
 disable_MQFunctions 896
 doce_deinstall 897
 doce_install 898
 DROP CONTACT 128
 DROP CONTACTGROUP 129
 DROP DATABASE 129
 DROP DBPARTITIONNUM VERIFY 131
 DROP TOOLS CATALOG 132
 ECHO 133
 EDIT 134
 enable_MQFunctions 899
 EXPORT 135
 FORCE APPLICATION 147
 GET ADMIN CONFIGURATION 149
 GET ALERT CONFIGURATION 150
 GET AUTHORIZATIONS 156
 GET CLI CONFIGURATION 157
 GET CONNECTION STATE 159
 GET CONTACTGROUP 160
 GET CONTACTGROUPS 160
 GET CONTACTS 161
 GET DATABASE CONFIGURATION 162
 GET DATABASE MANAGER CONFIGURATION 167
 GET DATABASE MANAGER MONITOR SWITCHES 172
 GET DESCRIPTION FOR HEALTH INDICATOR 175
 GET HEALTH NOTIFICATION CONTACT LIST 176
 GET HEALTH SNAPSHOT 177
 GET INSTANCE 179
 GET MONITOR SWITCHES 180
 GET RECOMMENDATIONS 182
 GET ROUTINE 185
 GET SNAPSHOT 186
 HELP 202
 HISTORY 203
 IMPORT 204
 INITIALIZE TAPE 235
 INSPECT 236
 installFixPack 901
 LIST ACTIVE DATABASES 244
 LIST APPLICATIONS 245
 LIST COMMAND OPTIONS 247
 LIST DATABASE DIRECTORY 248
 LIST DATABASE PARTITION GROUPS 251
 LIST DBPARTITIONNUMS 253
 LIST DCS APPLICATIONS 254
 LIST DCS DIRECTORY 256
 LIST DRDA INDOUBT TRANSACTIONS 257
 LIST HISTORY 258
 LIST INDOUBT TRANSACTIONS 261

コマンド (続き)

LIST NODE DIRECTORY 263
LIST ODBC DATA SOURCES 266
LIST PACKAGES/TABLES 266
LIST TABLESPACE CONTAINERS 269
LIST TABLESPACES 270
LIST UTILITIES 286
LOAD 288
LOAD QUERY 334
Microsoft Cluster Server 688
MIGRATE DATABASE 340
MQ Listener 741
PING 341
PRECOMPILE 344
PRUNE HISTORY/LOGFILE 371
PUT ROUTINE 373
QUERY CLIENT 374
QUIESCE 375
QUIESCE TABLESPACES FOR TABLE 378
QUIT 381
REBIND 381
RECOVER DATABASE 385
REDISTRIBUTE DATABASE PARTITION GROUP 391
REFRESH LDAP 402
REGISTER 403
REORG INDEXES/TABLE 410
REORGCHK 421
RESET ADMIN CONFIGURATION 432
RESET ALERT CONFIGURATION 433
RESET DATABASE CONFIGURATION 434
RESET DATABASE MANAGER CONFIGURATION 436
RESET MONITOR 438
RESTART DATABASE 439
RESTORE DATABASE 441
REWIND TAPE 461
ROLLFORWARD DATABASE 462
RUNCMD 473
RUNSTATS 474
SET CLIENT 489
SET RUNTIME DEGREE 492
SET TABLESPACE CONTAINERS 493
SET TAPE POSITION 495
SET UTIL_IMPACT_PRIORITY 495
SET WORKLOAD 498
SET WRITE 499
setup 904
sqlj 906
SQLJ プロファイル・バインド 848
SQLJ 変換プログラム 906
START DATABASE MANAGER 500
START HADR 507
STOP DATABASE MANAGER 510
STOP HADR 513
TAKEOVER HADR 514
TERMINATE 518
UNCATALOG DATABASE 519

コマンド (続き)

UNCATALOG DCS DATABASE 520
UNCATALOG LDAP DATABASE 521
UNCATALOG LDAP NODE 522
UNCATALOG NODE 522
UNCATALOG ODBC DATA SOURCE 523
UNQUIESCE 524
UPDATE ADMIN CONFIGURATION 525
UPDATE ALERT CONFIGURATION 527
UPDATE ALTERNATE SERVER FOR DATABASE 532
UPDATE ALTERNATE SERVER FOR LDAP
DATABASE 533
UPDATE CLI CONFIGURATION 534
UPDATE COMMAND OPTIONS 536
UPDATE CONTACT 537
UPDATE CONTACTGROUP 538
UPDATE DATABASE CONFIGURATION 539
UPDATE DATABASE MANAGER CONFIGURATION 543
UPDATE HEALTH NOTIFICATION CONTACT LIST 545
UPDATE HISTORY FILE 545
UPDATE LDAP NODE 548
UPDATE MONITOR SWITCHES 549
UPDATE XMLSCHEMA 551

コマンド行プロセッサ (CLP)

オプション 8
行継続文字 1
シェル・コマンド 7
終了 7, 381, 518
使用 1
バックエンド処理
終了 518
ヘルプへのアクセス 7
戻りコード 17
呼び出しコマンド 7
SQL ステートメント 19

コマンド構文

CLP コマンド 7

コミットなし (NC) 93

ご利用条件

資料の使用 950

コンテナ・タグの解放コマンド 894

コントロール・センター

開始 608

コントロール・センターの開始コマンド 608

[サ行]

最適化

REORG INDEXES/TABLE コマンド 410

索引

統計

RUNSTATS コマンド 474

REORGCHK コマンド 421

サンプル・データベースの作成コマンド 839

磁気テープ上のログ・ファイルの管理コマンド 879

磁気テープ・バックアップ 45

- システム・カタログ
 - 分析 606
- システム・データベース・ディレクトリー
 - アンカATALOG 519
 - 項目の削除 519
 - 変更、コメントの 292
- 修飾子
 - ファイル・タイプ
 - EXPORT コマンド 135
 - IMPORT コマンド 204
 - LOAD コマンド 288
- 終了
 - 異常 439
 - コマンド行プロセッサのバックエンド処理 518
 - 通常 510
- 終了コード
 - CLP 17
- 障害モニター・コントローラー・ユーティリティ 673
- 資料
 - 印刷 942
 - 注文 945
 - 概要 941
 - 使用に関するご利用条件 950
 - PDF 942
- スキーマ
 - 新規のデータベース 96
- ストレージ
 - 物理的 410
- すべてのディスク・ボリュームの GUID の表示コマンド 716
- すべてのパッケージの再バインド・コマンド 831
- 静止
 - ファントム 378
- 設計アドバイザー
 - db2advis コマンド 572
- 接続構成インポート・ツール・コマンド 612
- 接続構成エクスポート・ツール・コマンド 610
- 宣言生成プログラム・コマンド 633
- ゾーン 10 進ファイル・タイプ修飾子 288
- 増分リストア・イメージ順序の検査コマンド 622

[タ行]

- 第 1 出現データ収集 674
- 対話機能 CLI コマンド 624
- 大/小文字の区別
 - コマンド 1
 - 命名規則における 935
- チュートリアル
 - トラブルシューティング 950
 - 問題判別 950
 - Visual Explain 949
- 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 883

- データ
 - 再分散
 - REDISTRIBUTE DATABASE PARTITION GROUP コマンド 391
 - 整合性
 - 分離レベル 93
 - フラグメント化
 - 表の再編成を使用した除去 410
- データベース
 - アンカATALOG
 - システム・データベース・ディレクトリー 519
 - データベース接続サービス (DCS) 520
 - カタログ作成
 - CATALOG DATABASE コマンド 75
 - 間接 (間接) ディレクトリー項目 248
 - 許可検査 156
 - 再作成
 - RESTORE DATABASE コマンド 441
 - 再始動 439
 - 再編成 421
 - 削除
 - システム・データベース・ディレクトリー 519
 - データベース接続サービス (DCS) 520
 - DROP DATABASE コマンド 129
 - 作成
 - CREATE DATABASE コマンド 96
 - 情報収集 186
 - データの表へのロード 288
 - 統計 474
 - ドロップ
 - DROP DATABASE コマンド 129
 - バックアップ
 - 履歴ファイル 371
 - 表からファイルへのエクスポート
 - EXPORT コマンド 135
 - ファイルから表へのインポート
 - IMPORT コマンド 204
 - ホーム (home) ディレクトリー項目 248
 - マイグレーション
 - MIGRATE DATABASE コマンド 340
 - モニター 438
 - リカバリー
 - ROLLFORWARD DATABASE コマンド 462
 - リストア 441
 - リモート (remote) ディレクトリー項目 248
 - ロールフォワード・リカバリー
 - ROLLFORWARD DATABASE コマンド 462
- データベース移動ツール・コマンド 731
- データベース構成ファイル
 - 値の取り出し 162
 - 値のリセット 434
 - 更新 539
 - 例 162
- データベース事前マイグレーション・ツール・コマンド 620
- 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 520

データベースの再配置コマンド 833
データベース分析およびレポート・ツール・コマンド
説明 628
データベース・システム・モニター
記録スイッチの更新 549
GET DATABASE MANAGER MONITOR SWITCHES コマ
ンド 172
GET MONITOR SWITCHES コマンド 180
GET SNAPSHOT コマンド 186
RESET MONITOR コマンド 438
UPDATE MONITOR SWITCHES コマンド 549
データベース・ディレクトリー
サンプル内容 248
説明 248
変更、コメントの 92
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
753
データベース・パフォーマンス値のリセット・コマンド 828
データベース・マネージャー
開始 500
停止 510
統計 186
モニター・スイッチ
GET DATABASE MANAGER MONITOR SWITCHES コ
マンド 172
GET MONITOR SWITCHES コマンド 180
データベース・マネージャー構成ファイル
サンプル・ファイル 167
GET DATABASE MANAGER CONFIGURATION コマンド
を使用した値の取り出し 167
停止
DB2
db2stop コマンド 869
ディレクトリー
システム・データベース
項目の削除 519
データベースのアンカatalog (API) 519
変更、コメントの 92
データベース接続サービス (DCS)
項目の削除 520
ローカル・データベース
変更、コメントの 92
node
項目の削除 522
テキスト検索 911, 914, 916, 917, 923, 925, 927, 928, 930,
931
コマンド
ALTER INDEX 911
CLEANUP FOR TEXT 913
CLEAR COMMAND LOCKS 914
CLEAR EVENTS FOR INDEX 916
CREATE INDEX 917
DISABLE DATABASE FOR TEXT 923
DROP INDEX 925
ENABLE DATABASE FOR TEXT 927
HELP 928

テキスト検索 (続き)
コマンド (続き)
START FOR TEXT 930
STOP FOR TEXT 930
UPDATE INDEX 931
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 875
デフォルトの構成をリセット
データベース 434
DB2 Administration Server (DAS) 432
統計
再編成、索引の 421
データベース・マネージャー 186
database 474
REORGCHK 421
特記事項 953
特権
間接
概要 156
直接 156
レポート 156
database
作成時に付与 96
トラップ・ファイルのフォーマット・コマンド 895
トラブルシューティング
オンライン情報 950
チュートリアル 950
db2pd コマンド 763
トレース
アクティブ化 884
トレース・コマンド
説明 884

[ナ行]

ノード
SOCKS
CATALOG TCP/IP/TCP/IP4/TCP/IP6 NODE コマンド 88
ノード・ディレクトリー
項目の削除 522

[ハ行]

バージョン
プリコンパイル・オプション 344
バインド
暗黙的に作成されるスキーマ 55, 344
エラー 96
バインド・ファイル記述ツール・コマンド 603
パスワード
変更
ATTACH コマンド 40
CONNECT ステートメント 19
バックアップ
オブジェクト 599

- バックアップの検査コマンド 616
- パッケージ
 - 再作成 381
 - プリコンパイル・オプション 344
- パフォーマンス
 - 索引
 - REORGCHK コマンド 421
 - 表
 - 再編成 410
 - REORGCHK コマンド 421
 - Windows
 - パフォーマンス・モニター登録ツール・コマンド 830
- 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 830
- 反復可能読み取り (RR)
 - 変更 93
- 非 root インストール済み DB2 コマンド
 - インスタンス更新
 - db2nrupdt 760
 - configure
 - db2nrcfg 757
 - root フィーチャーの使用可能化
 - db2rfe 758
- 非コミット読み取り (UR)
 - 変更 93
- 表
 - 再編成
 - 必要性の判別 421
 - REORG INDEXES/TABLE コマンド 410
 - 統計
 - 説明 474
 - ファイルにエクスポートする 135
 - ファイルのロード 288
 - ファイルをインポートする 204
- 表スペース状態の獲得コマンド 882
- ファースト・ステップ 680
- ファイル・タイプ修飾子
 - エクスポート・ユーティリティー 135
 - exportimportload 937
 - IMPORT コマンド 204
 - LOAD コマンド 288
- ファイル・フォーマット
 - インポート、ファイルを表へ 204
 - ファイルへの表のエクスポート 135
- ファントム静止 378
- フェデレーテッド・プリコンパイル/BIND オプション 55, 344
- 複数ページ・ファイル割り振りの使用可能化 652
- 分散マップの取得コマンド 686
- 分離レベル
 - CHANGE ISOLATION LEVEL コマンド 93
- ヘルス・センターの開始コマンド 687
- ヘルプ
 - 言語の構成 946
 - SQL ステートメント 946
- ベンチマーク・ツール・コマンド 589

- ホスト・システム
 - 除去、DCS カタログ項目の 520
 - データベースのカタログ作成 78
 - DB2 Connect がサポートする接続
 - CATALOG DCS DATABASE コマンド 78
- 本書の構成 vii
- 本書の対象読者 vii

[マ行]

- 未確定トランザクション・フィールド 261
- ミラーリングされたデータベースの初期化コマンド 701
- 命名規則
 - データベース・マネージャー・オブジェクト 935
- メッセージ
 - プリコンパイル/BIND オプション 55, 344
 - ヘルプへのアクセス 7
- メッセージ・ヘルプ
 - 呼び出し 18
- メモリー・トラッカー・コマンド 749
- 戻りコード
 - コマンド行プロセッサ (CLP) 17
- モニター
 - データベース 172, 180
 - db2pd コマンド 763
- 問題判別
 - 診断ツール
 - db2fodc 674
 - db2support 869
 - チュートリアル 950
 - 利用できる情報 950
- 問題判別動作の DB2 データベースの構成コマンド 824

[ヤ行]

- ユーザー ID
 - 許可 156
- 呼び出し 18
 - コマンド・ヘルプ 18
- 読み取り固定 (RS)
 - 変更 93

[ラ行]

- ライセンス
 - ライセンス管理ツール・コマンド 714
- リカバリー
 - ロールフォワードなし 441
 - ロールフォワードを用いた 462
 - database 441
- リストア
 - 旧バージョンの DB2 データベース 441
- ローカル・データベース・ディレクトリー
 - 変更、コメントの 92

ロード
 ファイルをデータベース表へ 288
 ファイル・タイプ修飾子 288
ロード・ユーティリティー
 一時ファイル
 LOAD コマンド 288
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 717
ログ
 ロールフォワード中のリスト 462
ログ・シーケンス番号の検出コマンド 669
ロック
 リセット、最大値をデフォルトに 434

[ワ行]

ワークステーション
 remote
 データベースのアンカタログ 519
 データベースのカタログ作成 75
 ノードのアンカタログ 522
ワークロード管理
 SET WORKLOAD コマンド 498

A

action プリコンパイル/BIND オプション 55, 344
ACTIVATE DATABASE コマンド
 構文の説明 31
Active Directory
 スキーマ拡張
 コマンド 842
ADD CONTACT コマンド
 構文の説明 32
ADD CONTACTGROUP コマンド
 構文の説明 33
ADD DBPARTITIONNUM コマンド 34
ADD XMLSCHEMA DOCUMENT コマンド
 構文 37
ADMIN 構成
 ネットワーク・パラメーター値 525
 ファイル 149
 リセット、デフォルトに 432
 例 149
Administration Server
 構成 149
 作成 563
 ドロップ 563
anyorder ファイル・タイプ修飾子 288
APPC (詳細プログラム間通信)
 node
 アンカタログ 522
ARCHIVE LOG コマンド 38
ASC インポート・ファイル・タイプ 204
ATTACH コマンド 40

AUTOCONFIGURE コマンド 42

B

BACKUP DATABASE コマンド 45
binarynumerics ファイル・タイプ修飾子 288
BIND コマンド
 構文 55
bindfile プリコンパイル・オプション 344
blocking プリコンパイル/BIND オプション 55, 344

C

CALL ステートメント
 CLP からの実行 19
CATALOG DATABASE コマンド
 構文 75
CATALOG DCS DATABASE コマンド 78
CATALOG LDAP DATABASE コマンド 80
CATALOG LDAP NODE コマンド 83
CATALOG LOCAL NODE コマンド 84
CATALOG NAMED PIPE NODE コマンド 85
CATALOG ODBC DATA SOURCE コマンド 87
CATALOG TCP/IP NODE コマンド 88
CCSIDG プリコンパイル/BIND オプション 55, 344
CCSIDM プリコンパイル/BIND オプション 55, 344
CCSIDS プリコンパイル/BIND オプション 55, 344
CHANGE DATABASE COMMENT コマンド 92
CHANGE ISOLATION LEVEL コマンド 93
chardel ファイル・タイプ修飾子
 インポート 204
 エクスポート 135
 load 288
charsub プリコンパイル/BIND オプション 55, 344
CLI (コール・レベル・インターフェース)
 構成 157
CLIPKG プリコンパイル/BIND オプション 55
CLI/ODBC 静的パッケージ・バインディング・ツール・コマンド 605
CLOSE ステートメント
 CLP からの実行 19
CLP (コマンド行プロセッサ)
 コマンド
 構文 7
 終了 381, 518
cnulreqd プリコンパイル/BIND オプション 55, 344
coldel ファイル・タイプ修飾子
 インポート
 IMPORT コマンド 204
 エクスポート
 EXPORT コマンド 135
 load
 LOAD コマンド 288
collection プリコンパイル/BIND オプション 55, 344

COMPLETE XMLSCHEMA コマンド
 構文 95
 compound ファイル・タイプ修飾子 204
 CONNECT ステートメント
 CLP からの実行 19
 connect プリコンパイル・オプション 344
 CREATE DATABASE コマンド
 説明 96
 CREATE DATABASE ステートメントの RESTRICTIVE 節
 96
 CREATE TOOLS CATALOG コマンド 117

D

DAS の自動始動コマンド 553
 dasauto コマンド 553
 dasert コマンド 553
 dasdrop コマンド 554
 dasmigr コマンド
 説明 555
 dasupdt コマンド 556
 dateformat ファイル・タイプ修飾子
 IMPORT コマンド 204
 LOAD コマンド 288
 DB2 Administration Server (DAS)
 管理 563
 構成 149
 コマンド
 作成 553
 説明 563
 作成
 db2admin コマンド 563
 ドロップ 563
 DB2 Administration Server の除去コマンド 554
 DB2 Administration Server のマイグレーション・コマンド
 555
 DB2 Connect
 サポートされる接続 78
 DB2 for z/OS
 プランおよびパッケージのバインディング 854
 DB2 HA スクリプトのインストールまたは更新 626
 DB2 インスタンスの制御コマンド 681
 DB2 インフォメーション・センター
 言語 946
 更新 947
 バージョン 946
 別の言語で表示する 946
 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 897
 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
 898
 DB2 オブジェクトの許可の設定コマンド 667
 DB2 管理プログラム
 コマンド 683
 ログ照会コマンド 685
 db2 コマンド 7
 DB2 コマンド・ウィンドウのオープン・コマンド 625
 DB2 サービス・レベルの表示コマンド 713
 DB2 システム・トレイの開始コマンド 877
 DB2 障害モニター・コマンド 671
 DB2 資料の印刷方法 945
 DB2 シンクロナイザーの開始コマンド 876
 DB2 スナップショット・バックアップ・オブジェクトの管理
 599
 DB2 製品インストール・コマンド 559
 DB2 製品のアンインストール・コマンド 892
 DB2 製品またはフィーチャーのアンインストール・コマンド
 557
 DB2 データベース・ドライブのマップ・コマンド 651
 DB2 統計および DDL 抽出ツール・コマンド 717
 DB2 のインストール・コマンド 704, 847, 904
 DB2 の開始コマンド 867
 DB2 の停止コマンド 869
 DB2 プロファイル・レジストリー・コマンド 843
 db2acsutil コマンド 599
 db2admin コマンド 563
 db2adutl コマンド
 説明 565
 db2advis コマンド
 説明 572
 db2audit コマンド
 説明 580
 db2batch コマンド
 説明 589
 db2bfd コマンド
 説明 603
 db2ca コマンド 604
 db2cap コマンド 605
 db2cat コマンド 606
 db2cc コマンド 608
 db2cfexp コマンド 610
 db2cfimp コマンド 612
 db2chglbpath コマンド 613
 db2chgpath コマンド 615
 db2ckbkp コマンド 616
 db2ckmig コマンド
 説明 620
 db2ckrst コマンド 622
 db2cli コマンド 624
 db2cmd コマンド 625
 db2cptsa コマンド 626
 db2dart コマンド
 説明 628
 db2daslevel コマンド 632
 db2dclgn コマンド
 説明 633
 db2diag コマンド
 説明 635
 db2diag.log 分析ツール・コマンド 635
 db2drdat コマンド
 説明 650
 db2drvmp コマンド 651

db2empfa コマンド 652
 db2envar.bat コマンド 653
 db2eva コマンド 654
 db2evmon コマンド 655
 db2evtbl コマンド 656
 db2exfmt コマンド
 説明 658
 db2exmig コマンド
 説明 660
 db2expln コマンド
 説明 661
 db2extsec コマンド 667
 db2flsn コマンド 669
 db2fm コマンド
 説明 671
 db2fmcu コマンド 673
 db2fodc コマンド 674
 db2fs コマンド 680
 db2gcf コマンド 681
 db2gov コマンド
 説明 683
 db2govlg コマンド 685
 db2gpmap コマンド 686
 db2hc コマンド 687
 db2iauto コマンド 688
 db2iclus コマンド 688
 db2icrt コマンド
 説明 691
 db2idrop コマンド
 説明 696
 db2ilist コマンド
 説明 698
 db2imigr コマンド
 説明 699
 db2inidb コマンド
 説明 701
 db2inspf コマンド
 説明 703
 db2iprune コマンド 704
 db2isetup コマンド 705
 db2iupdt コマンド
 説明 707
 db2jdbcbind コマンド 710
 db2ldcfg コマンド
 説明 712
 db2level コマンド
 説明 713
 db2licm コマンド
 説明 714
 db2listvolumes コマンド 716
 db2logsforrwd コマンド 717
 db2look コマンド
 説明 717
 db2ls コマンド
 説明 729
 db2move コマンド
 説明 731
 db2mqlsn コマンド 741
 db2mscs コマンド 745
 db2mtrk コマンド
 説明 749
 db2nchg コマンド
 説明 753
 db2ncrt コマンド
 説明 754
 db2ndrop コマンド
 説明 756
 db2nrcfg コマンド 757
 db2nrupdt コマンド 760
 DB2OPTIONS 環境変数
 CLP オプションの設定 8
 db2osconf コマンド
 説明 760
 db2pd コマンド
 説明 763
 db2pdcfg コマンド
 説明 824
 db2perfc コマンド
 説明 828
 db2perfi コマンド
 説明 830
 db2perfr コマンド
 説明 830
 db2rbind コマンド
 説明 831
 db2relocatedb コマンド
 説明 833
 db2rfe コマンド
 説明 758
 db2rfpen コマンド 838
 db2rspgn 応答ファイル・ジェネレーター 838
 db2sampl コマンド 839
 db2schex コマンド 842
 db2set コマンド
 説明 843
 db2setup コマンド
 説明 847
 db2sqljbind コマンド 848
 db2sqljcustomize コマンド 854
 db2sqljprint コマンド
 説明 867
 db2start コマンド
 説明 500, 867
 db2stat コマンド 868
 db2stop コマンド
 説明 510, 869
 db2swtch コマンド 875
 db2sync コマンド 876
 db2systray コマンド 877
 db2tapemgr コマンド
 説明 879

db2tbst コマンド 882
 db2tdbmgr コマンド
 説明 883
 db2trc コマンド
 説明 884
 db2ts コマンド
 ALTER INDEX 911
 CLEANUP FOR TEXT 913
 CLEAR COMMAND LOCKS 914
 CLEAR EVENTS FOR INDEX 916
 CREATE INDEX 917
 DISABLE DATABASE FOR TEXT 923
 DROP INDEX 925
 ENABLE DATABASE FOR TEXT 927
 HELP 928
 START FOR TEXT 930
 STOP FOR TEXT 930
 UPDATE INDEX 931
 db2uiddl コマンド
 説明 891
 db2unins コマンド 892
 db2untag コマンド 894
 db2xdbmig コマンド 895
 db2xpvt コマンド 895
 db2_deinstall コマンド
 説明 557
 db2_install コマンド
 説明 559
 db2_local_ps コマンド 562
 DCLGEN コマンド 633
 DEACTIVATE DATABASE コマンド 119
 DECLARE CURSOR ステートメント
 CLP で実行 19
 DECOMPOSE XML DOCUMENT コマンド
 説明 120
 decplusblank ファイル・タイプ修飾子
 EXPORT コマンド 135
 IMPORT コマンド 204
 LOAD コマンド 288
 decpt ファイル・タイプ修飾子
 EXPORT コマンド 135
 IMPORT コマンド 204
 LOAD コマンド 288
 delprioritychar ファイル・タイプ修飾子
 IMPORT コマンド 204
 LOAD コマンド 288
 DEREGISTER コマンド 122
 DESCRIBE コマンド
 説明
 ADMIN_CMD プロシーチャーを使用しない 123
 DETACH コマンド
 説明 128
 disable_MQFunctions コマンド 896
 doce_deinstall コマンド
 説明 897

doce_install コマンド
 説明 898
 DRDA トレース・コマンド 650
 DROP CONTACT コマンド
 説明 128
 DROP CONTACTGROUP コマンド
 説明 129
 DROP DATABASE コマンド
 説明 129
 DROP DBPARTITIONNUM VERIFY コマンド 131
 DROP TOOLS CATALOG コマンド 132
 dumpfile ファイル・タイプ修飾子 288
 DYNAMICRULES プリコンパイル/BIND オプション
 BIND コマンド 55
 PRECOMPILE コマンド 344

E

ECHO コマンド 133
 EDIT コマンド 134
 enable_MQFunctions コマンド 899
 Explain 表
 内容のフォーマット 658
 Explain 表のマイグレーション・コマンド 660
 EXPORT コマンド
 説明
 ADMIN_CMD プロシーチャーを使用しない 135

F

fastparse ファイル・タイプ修飾子 288
 federated_asynchrony プリコンパイル/バインド・オプション
 55, 344
 FETCH ステートメント
 CLP からの実行 19
 FODC (第 1 出現データ・キャプチャー)
 db2pdcfg コマンドのオプション 824
 FORCE APPLICATION コマンド 147
 forcein ファイル・タイプ修飾子 204, 288
 funcpath プリコンパイル/BIND オプション 55, 344

G

generatedignore ファイル・タイプ修飾子 204, 288
 generatedmissing ファイル・タイプ修飾子 204, 288
 generatedoverride ファイル・タイプ修飾子 288
 generic プリコンパイル/BIND オプション 55, 344
 GET ADMIN CONFIGURATION コマンド 149
 GET ALERT CONFIGURATION コマンド 150
 GET AUTHORIZATIONS コマンド 156
 GET CLI CONFIGURATION コマンド 157
 GET CONNECTION STATE コマンド 159
 GET CONTACTGROUP コマンド 160
 GET CONTACTGROUPS コマンド 160
 GET CONTACTS コマンド 161

GET DATABASE CONFIGURATION コマンド 162
GET DATABASE MANAGER CONFIGURATION コマンド
167
GET DATABASE MANAGER MONITOR SWITCHES コマンド
172
GET DESCRIPTION FOR HEALTH INDICATOR コマンド
175
GET HEALTH NOTIFICATION CONTACT LIST コマンド
176
GET HEALTH SNAPSHOT コマンド 177
GET INSTANCE コマンド 179
GET MONITOR SWITCHES コマンド 180
GET RECOMMENDATIONS
コマンド 182
GET ROUTINE コマンド 185
GET SNAPSHOT コマンド 186
UPDATE MONITOR SWITCHES への影響 549
grant BIND オプション 55
grantgroup BIND オプション 55
grantuser BIND オプション 55

H

help
コマンド 18
メッセージ 18
HELP コマンド
DB2 Database for Linux, UNIX, and Windows 202
HISTORY コマンド 203

I

identityignore 204
ファイル・タイプ修飾子 288
identitymissing
ファイル・タイプ修飾子 204, 288
identityoverride
ファイル・タイプ修飾子 288
implieddecimal ファイル・タイプ修飾子 204, 288
IMPORT コマンド 204
indexfreespace ファイル・タイプ修飾子 288
indexixf ファイル・タイプ修飾子 204
indexschema ファイル・タイプ修飾子 204
INITIALIZE TAPE コマンド 235
insert プリコンパイル/BIND オプション 55, 344
INSPECT コマンド 236
installFixPack コマンド 901
IPX/SPX ノード
アンカタログ 522
isolation プリコンパイル/BIND オプション 55, 344

J

Java Database Connectivity (JDBC)
パッケージ・バインド・プログラム・ユーティリティ・コ
マンド 710

K

keepblanks ファイル・タイプ修飾子
ロード
LOAD コマンド 288
IMPORT コマンド 204

L

LANGLEVEL プリコンパイル・オプション
SQL92E 344
LDAP 環境の構成コマンド 712
level プリコンパイル・オプション 344
Linux/UNIX の DB2 プロセス状況コマンド 562
LIST ACTIVE DATABASES コマンド 244
LIST APPLICATIONS コマンド 245
LIST COMMAND OPTIONS コマンド 247
LIST DATABASE DIRECTORY コマンド 248
LIST DATABASE PARTITION GROUPS コマンド 251
LIST DBPARTITIONNUMS コマンド 253
LIST DCS APPLICATIONS コマンド 254
LIST DCS DIRECTORY コマンド 256
LIST DRDA INDOUBT TRANSACTIONS コマンド 257
LIST HISTORY コマンド 258
LIST INDOUBT TRANSACTIONS コマンド 261
LIST NODE DIRECTORY コマンド 263
LIST ODBC DATA SOURCES コマンド 266
LIST PACKAGES コマンド 266
LIST PACKAGES/TABLES コマンド 266
LIST TABLES コマンド 266
LIST TABLESPACE CONTAINERS コマンド 269
LIST TABLESPACES コマンド 270
LIST UTILITIES コマンド 286
LOAD QUERY コマンド 334
LOAD コマンド
概要 288
lobsinfile ファイル・タイプ修飾子
インポート 204
エクスポート 135
ロード 288
longerror プリコンパイル・オプション 344

M

Microsoft Cluster Server コマンド 688
MIGRATE DATABASE コマンド 340
MQ Listener コマンド 741

N

NetBIOS
ノード
 アンカタログ 522
nochecklengths ファイル・タイプ修飾子
 インポート 204
 ロード 288
nodefaults ファイル・タイプ修飾子
 インポート 204
nodoubledel ファイル・タイプ修飾子
 インポート 204
 エクスポート 135
 ロード 288
noeofchar ファイル・タイプ修飾子
 インポート 204
 ロード 288
noheader ファイル・タイプ修飾子
 ロード 288
NOLINEMACRO プリコンパイル・オプション 344
norowwarnings ファイル・タイプ修飾子
 LOAD コマンド 288
notypeid ファイル・タイプ修飾子
 IMPORT コマンド 204
NULL 値
 SQL
 コマンド行プロセッサの表記 1
NULL スtring 1
nullindchar ファイル・タイプ修飾子
 IMPORT コマンド 288
 LOAD コマンド 204

O

OPEN ステートメント
 CLP からの実行 19
optlevel プリコンパイル・オプション 344
optprofile プリコンパイル/BIND オプション 55
output プリコンパイル・オプション 344
owner プリコンパイル/BIND オプション 55, 344

P

packeddecimal ファイル・タイプ修飾子 288
pagefreespace ファイル・タイプ修飾子 288
PING コマンド
 説明 341
PRECOMPILE コマンド
 説明 344
PREP コマンド 344
PRUNE HISTORY/LOGFILE コマンド
 ADMIN_CMD プロシージャを使用しない 371
PUT ROUTINE コマンド 373

Q

qualifier プリコンパイル/BIND オプション 55, 344
QUERY CLIENT コマンド 374
queryopt プリコンパイル/BIND オプション
 BIND コマンド 55
 PRECOMPILE コマンド 344
QUIESCE TABLESPACES FOR TABLE コマンド 378
QUIESCE コマンド 375
QUIT コマンド 381

R

REBIND コマンド 381
reclen ファイル・タイプ修飾子 204
 ロード 288
RECOVER DATABASE コマンド 385
REDISTRIBUTE DATABASE PARTITION GROUP コマンド
 391
REFRESH LDAP コマンド 402
REGISTER XMLSCHEMA コマンド
 構文 406
REGISTER XSROBJECT コマンド
 構文 408
REGISTER コマンド 403
release プリコンパイル/BIND オプション 55, 344
REORG TABLE コマンド 410
REORGCHK コマンド 421
RESET ADMIN CONFIGURATION コマンド 432
RESET ALERT CONFIGURATION コマンド 433
RESET DATABASE CONFIGURATION コマンド 434
RESET DATABASE MANAGER CONFIGURATION コマンド
 436
RESET MONITOR コマンド 438
Reset rollforward pending state コマンド 838
RESTART DATABASE コマンド 439
RESTORE DATABASE コマンド 441
REWIND TAPE コマンド 461
ROLLFORWARD DATABASE コマンド 462
RUNCMD コマンド 473
RUNSTATS コマンド
 構文 474

S

SELECT ステートメント
 CLP からの実行 19
 EXPORT コマンド内の 135
SET CLIENT コマンド 489
SET RUNTIME DEGREE コマンド 492
SET TABLESPACE CONTAINERS コマンド 493
SET TAPE POSITION コマンド 495
SET UTIL_IMPACT_PRIORITY コマンド 495
SET WORKLOAD コマンド 498
SET WRITE コマンド 499
setup コマンド 904

SIGNALRM 信号
開始、データベース・マネージャーの 500
SIGINT 信号
開始、データベース・マネージャーの 500
SOCKS
node
パラメーター 88
SQL および XQuery Explain コマンド 661
SQL ステートメント
コマンド行の使用 19
ヘルプへのアクセス 7
ヘルプを表示する 946
sqlca プリコンパイル・オプション 344
sqlerror プリコンパイル/BIND オプション 55, 344
sqlflag プリコンパイル・オプション 344
sqlj コマンド 906
SQLJ プロファイル・バインド・プログラム・コマンド 848
SQLJ プロファイル・プリンター・コマンド 867
SQLJ 変換プログラム・コマンド 906
sqlrules プリコンパイル・オプション 344
sqlwarn プリコンパイル/BIND オプション 55, 344
START DATABASE MANAGER コマンド 500
START HADR コマンド 507
STOP DATABASE MANAGER コマンド 510
STOP HADR コマンド 513
strdel プリコンパイル/BIND オプション 55, 344
striptblanks ファイル・タイプ修飾子 204, 288
striptnulls ファイル・タイプ修飾子 204, 288
subtableconvert ファイル・タイプ修飾子 288
syncpoint プリコンパイル・オプション 344

T

TAKEOVER HADR コマンド 514
target プリコンパイル・オプション 344
TCP/IP
node
アンカタログ 522
TERMINATE コマンド 518
text プリコンパイル/BIND オプション 55, 344
timeformat ファイル・タイプ修飾子 204, 288
timestampformat ファイル・タイプ修飾子
IMPORT コマンド 204
LOAD コマンド 288
totalreespace ファイル・タイプ修飾子 288
transform group プリコンパイル/BIND オプション 55, 344
True Type フォント
コマンド行プロセッサの要件 1
TSM
アーカイブ・イメージ 565

U

UNCATALOG DATABASE コマンド 519
UNCATALOG DCS DATABASE コマンド 520

UNCATALOG LDAP DATABASE コマンド 521
UNCATALOG LDAP NODE コマンド 522
UNCATALOG NODE コマンド 522
UNCATALOG ODBC DATA SOURCE コマンド 523
UNQUIESCE コマンド 524
UPDATE ADMIN CONFIGURATION コマンド 525
UPDATE ALERT CONFIGURATION コマンド 527
UPDATE ALTERNATE SERVER FOR DATABASE コマンド
532
UPDATE ALTERNATE SERVER FOR LDAP DATABASE コ
マンド 533
UPDATE CLI CONFIGURATION コマンド 534
UPDATE COMMAND OPTIONS コマンド 536
UPDATE CONTACT コマンド 537
UPDATE CONTACTGROUP コマンド 538
UPDATE DATABASE CONFIGURATION コマンド 539
UPDATE DATABASE MANAGER CONFIGURATION コマン
ド 543
UPDATE HEALTH NOTIFICATION CONTACT LIST コマンド
545
UPDATE HISTORY FILE コマンド 545
UPDATE LDAP NODE コマンド 548
UPDATE MONITOR SWITCHES コマンド 549
UPDATE XMLSCHEMA コマンド
構文 551
usedefaults ファイル・タイプ修飾子 204, 288

V

V5 セマンティクスへのユニーク索引変換コマンド 891
validate プリコンパイル/BIND オプション
BIND コマンド 55
PRECOMPILE コマンド 344
Visual Explain
チュートリアル 949

W

WCHARTYPE プリコンパイラー・オプション
説明 344
Windows の DB2 プロセス状況コマンド 868
Windows フェールオーバー・ユーティリティのセットアッ
プ・コマンド 745

X

XSBA (バックアップ・サービス API) オプション
BACKUP DATABASE コマンド 45
XML スキーマ
リポジトリ
ADD XMLSCHEMA DOCUMENT コマンド 37
COMPLETE XMLSCHEMA コマンド 95
REGISTER XMLSCHEMA コマンド 406
REGISTER XSROBJECT コマンド 408
UPDATE XMLSCHEMA コマンド 551

XSR オブジェクトのマイグレーション・コマンド 895
X/Open バックアップ・サービス API (XBSA)
インターフェース 45



Printed in Japan

SC88-4432-02



日本アイ・ビー・エム株式会社
〒106-8711 東京都港区六本木3-2-12

Spine information:

DB2 Version 9.5 for Linux, UNIX, and Windows

コマンド・リファレンス

