

DB2®

IBM

DB2 Version 9
for Linux, UNIX, and Windows

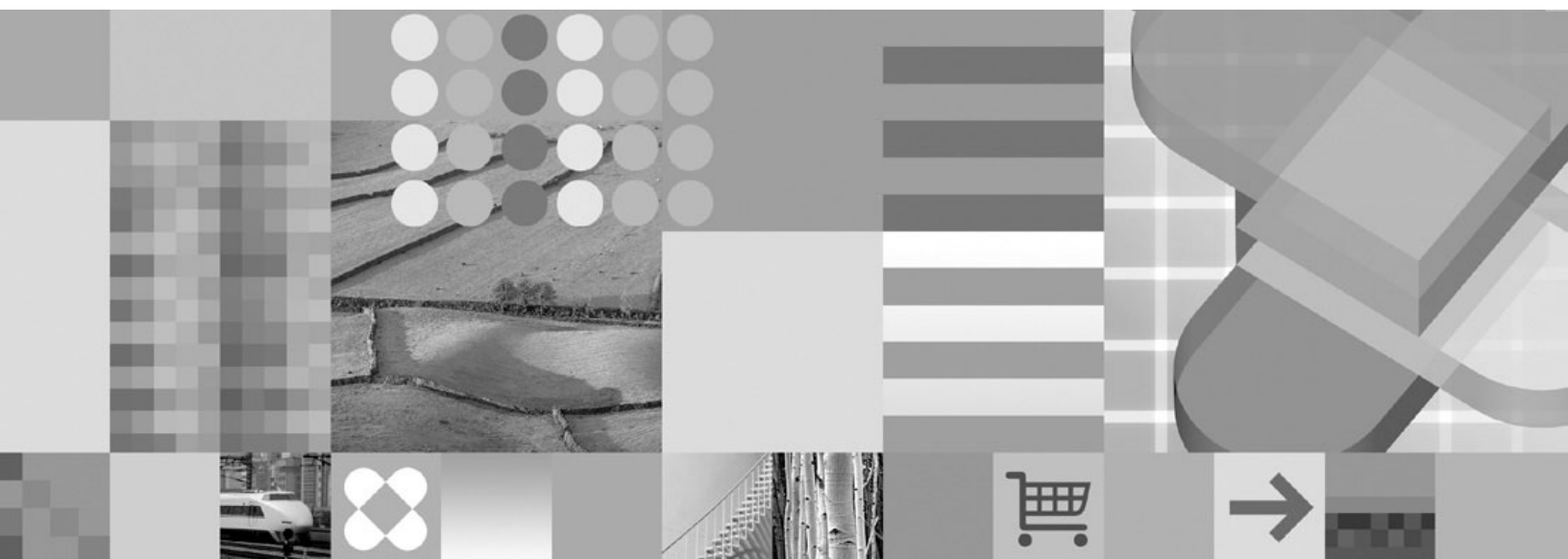


コマンド・リファレンス

DB2®

IBM

DB2 Version 9
for Linux, UNIX, and Windows



コマンド・リファレンス

本書および本書で紹介する製品をご使用になる前に、『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 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。

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

<http://www.ibm.com/jp/manuals/main/mail.html>

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

<http://www.ibm.com/jp/manuals/> の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

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： SC10-4226-00
Command Reference
DB2 Version 9 for Linux, UNIX, and Windows

発 行： 日本アイ・ビー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

第1刷 2006.7

この文書では、平成明朝体™W3、平成明朝体™W7、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体*)は、(財)日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注* 平成明朝体™W3、平成明朝体™W7、平成明朝体™W9、平成角ゴシック体™W3、
平成角ゴシック体™W5、平成角ゴシック体™W7

© Copyright International Business Machines Corporation 1993, 2006. All rights reserved.

© Copyright IBM Japan 2006

目次

本書について	vii
本書の対象読者	vii
本書の構成	vii

第 1 章 システム・コマンド 1

コマンドの説明の編成方法	1
dasauto - DB2 Administration Server の自動始動	3
dasprt - DB2 Administration Server の作成	4
dasdrop - DB2 Administration Server の除去	5
dasmigr - DB2 Administration Server の移行	6
dasupdt - DAS の更新	8
db2_deinstall - DB2 製品またはフィーチャーのアンインストール	10
db2_install - DB2 製品のインストール	11
db2admin - DB2 Administration Server	13
db2adutl - TSM 内の DB2 オブジェクトの管理	15
db2advis - DB2 設計アドバイザー	23
db2audit - 監査機能管理者用ツール	29
db2batch - ベンチマーク・ツール	35
db2bfd - バインド・ファイル記述ツール	45
db2ca - 構成アシスタントの開始	46
db2cap - CLI/ODBC 静的パッケージ・バインディング・ツール	47
db2cat - システム・カタログ分析	49
db2cc - コントロール・センターの開始	51
db2cfexp - 接続構成エクスポート・ツール	53
db2cfimp - 接続構成インポート・ツール	55
db2chglibpath - 組み込みランタイム・ライブラリー検索パスの変更	56
db2chgpath - 組み込みランタイム・パスの変更	58
db2ckbkp - バックアップの検査	59
db2ckmig - データベース事前移行ツール	63
db2ckrst - 増分リストア・イメージ順序の検査	65
db2cli - DB2 対話機能 CLI	67
db2cmd - DB2 コマンド・ウィンドウのオープン	68
db2dart - データベース分析およびレポート・ツール	70
db2daslevel - DAS レベルの表示	75
db2dclgn - 宣言生成プログラム	76
db2diag - db2diag.log 分析ツール	79
db2drdat - DRDA trace	91
db2drvmp - DB2 データベースのドライブ・マップ	93
db2empfa - 複数ページ・ファイル割り振りの使用可能化	95
db2eva - イベント・アナライザー	96
db2evmon - イベント・モニター生産性向上ツール	98
db2evtbl - イベント・モニターのターゲット表定義の生成	100
db2exfmt - Explain 表フォーマット	102
db2exmig - Explain 表の移行	104
db2expln - SQL および XQuery Explain	105
db2extsec - DB2 オブジェクトの許可の設定	111

db2flsn - ログ・シーケンス番号の検出	112
db2fm - DB2 障害モニター	114
db2fs - ファースト・ステップ	116
db2gcf - DB2 インスタンスの制御	117
db2gov - DB2 管理プログラム	120
db2govlg - DB2 管理プログラム・ログ照会	122
db2gpmmap - 分散マップの取得	123
db2hc - ヘルス・センターの開始	124
db2iauto - インスタンスの自動開始	125
db2iclus - Microsoft Cluster Server	126
db2icrt - インスタンスの作成	129
db2idrop - インスタンスの除去	132
db2ilist - インスタンスのリスト	134
db2imigr - インスタンスの移行	135
db2inidb - ミラーリングされたデータベースの初期化	137
db2inspf - 検査結果のフォーマット	140
db2isetup - インスタンス作成インターフェースの開始	141
db2iupdt - インスタンスの更新	143
db2jdbcbind - DB2 JDBC パッケージ・バインド・プログラム	147
db2ldcfg - LDAP 環境の構成	149
db2level - DB2 サービス・レベルの表示	151
db2licm - ライセンス管理ツール	152
db2listvolumes - すべてのディスク・ボリュームの GUID の表示	154
db2logsforrwd - ロールフォワード・リカバリーに必要なログのリスト	155
db2look - DB2 統計および DDL 抽出ツール	156
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	167
db2move - データベース移動ツール	169
db2mqslsn - MQ Listener	178
db2mscs - Windows フェイルオーバー・ユーティリティのセットアップ	182
db2mtrk - メモリー・トラッカー	186
db2nchg - データベース・パーティション・サーバー構成の変更	190
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	192
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	195
db2osconf - カーネル・パラメーター値のためのユーティリティ	197
db2pd - DB2 データベースのモニターおよびトラブルシューティング	201
db2pdcfg - 問題判別動作用の DB2 データベースの構成	243
db2perf - データベース・パフォーマンス値のリセット	246

db2perfi - パフォーマンス・カウンター登録ユーティリティー	248
db2perftr - パフォーマンス・モニター登録ツール	249
db2rbind - Rebind all packages	250
db2_recon_aid - 複数の表の調整	252
db2relocatedb - データベースの再配置	255
db2rfpen - ロールフォワード・ペンディング状態にリセット	260
db2rspgn - 応答ファイル生成プログラム (Windows)	261
db2sampl - サンプル・データベースの作成	262
db2set - DB2 プロファイル・レジストリー	265
db2setup - DB2 のインストール	268
db2sql92 - SQL92 準拠 SQL ステートメント・プロセッサ	269
db2sqljbind - SQLJ プロファイル・バインダー	272
db2sqljcustomize - SQLJ プロファイル・カスタマイザー	279
db2sqljprint - SQLJ プロファイル・プリンター	291
db2start - DB2 の開始	292
db2stop - DB2 の停止	293
db2support - 問題分析および環境収集ツール	294
db2swtch - デフォルト DB2 コピーの切り替え	300
db2sync - DB2 シンクロナイザーの開始	301
db2sysstray - DB2 システム・トレイの開始	302
db2tapemgr - テープ上のログ・ファイルの管理	305
db2tbtst - 表スペース状態の獲得	308
db2trc - トレース	309
db2uiddl - V5 セマンティクスへのユニーク索引変換の準備	314
db2undgp - 実行特権の取り消し	316
db2unins - DB2 データベース製品のアンインストール	317
db2untag - コンテナ・タグの解放	319
db2xpirt - トラップ・ファイルのフォーマット	320
doce_deinstall - DB2 インフォメーション・センターのアンインストール	321
doce_install - DB2 インフォメーション・センターのインストール	323
disable_MQFunctions	325
enable_MQFunctions	327
installFixPack - インストール済み DB2 製品の更新	330
setup - DB2 のインストール	332
sqlj - SQLJ 変換プログラム	334

第 2 章 コマンド行プロセッサ (CLP) 339

db2 - コマンド行プロセッサの呼び出し	339
コマンド行プロセッサのオプション	340
コマンド行プロセッサの戻りコード	349
コマンド行プロセッサのフィーチャー	350
コマンド行プロセッサのヘルプ	356
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	356
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	356

第 3 章 CLP コマンド 359

DB2 CLP コマンド	359
ACTIVATE DATABASE	363
ADD CONTACT	365
ADD CONTACTGROUP	367
ADD DBPARTITIONNUM	368
ADD XMLSCHEMA DOCUMENT	371
ARCHIVE LOG	373
ATTACH	376
AUTOCONFIGURE	378
BACKUP DATABASE	382
BIND	389
CATALOG DATABASE	408
CATALOG DCS DATABASE	412
CATALOG LDAP DATABASE	415
CATALOG LDAP NODE	419
CATALOG LOCAL NODE	421
CATALOG NAMED PIPE NODE	423
CATALOG ODBC DATA SOURCE	425
CATALOG TCPIP/TCPIP4/TCPIP6 NODE	426
CHANGE DATABASE COMMENT	430
CHANGE ISOLATION LEVEL	432
COMPLETE XMLSCHEMA	434
CREATE DATABASE	435
CREATE TOOLS CATALOG	451
DEACTIVATE DATABASE	454
DECOMPOSE XML DOCUMENT	456
DEREGISTER	458
DESCRIBE	460
DETACH	466
DROP CONTACT	467
DROP CONTACTGROUP	468
DROP DATABASE	469
DROP DBPARTITIONNUM VERIFY	471
DROP TOOLS CATALOG	473
ECHO	475
EDIT	476
EXPORT	478
FORCE APPLICATION	484
GET ADMIN CONFIGURATION	486
GET ALERT CONFIGURATION	488
GET AUTHORIZATIONS	494
GET CLI CONFIGURATION	496
GET CONNECTION STATE	498
GET CONTACTGROUP	499
GET CONTACTGROUPS	500
GET CONTACTS	501
GET DATABASE CONFIGURATION	502
GET DATABASE MANAGER CONFIGURATION	508
GET DATABASE MANAGER MONITOR SWITCHES	513
GET DESCRIPTION FOR HEALTH INDICATOR	516
GET HEALTH NOTIFICATION CONTACT LIST	518
GET HEALTH SNAPSHOT	519
GET INSTANCE	522
GET MONITOR SWITCHES	523
GET RECOMMENDATIONS	526

GET ROUTINE	530	SET CLIENT	784
GET SNAPSHOT	532	SET RUNTIME DEGREE	787
HELP	538	SET TABLESPACE CONTAINERS	789
HISTORY	540	SET TAPE POSITION	791
IMPORT	541	SET UTIL_IMPACT_PRIORITY	792
INITIALIZE TAPE	560	SET WRITE	794
INSPECT	562	START DATABASE MANAGER	796
LIST ACTIVE DATABASES	568	START HADR	803
LIST APPLICATIONS	570	STOP DATABASE MANAGER	806
LIST COMMAND OPTIONS	572	STOP HADR	810
LIST DATABASE DIRECTORY	573	TAKEOVER HADR	812
LIST DATABASE PARTITION GROUPS	577	TERMINATE	815
LIST DBPARTITIONNUMS	579	UNCATALOG DATABASE	816
LIST DCS APPLICATIONS	580	UNCATALOG DCS DATABASE	818
LIST DCS DIRECTORY	582	UNCATALOG LDAP DATABASE	820
LIST DRDA INDOUBT TRANSACTIONS	584	UNCATALOG LDAP NODE	822
LIST HISTORY	586	UNCATALOG NODE	823
LIST INDOUBT TRANSACTIONS	589	UNCATALOG ODBC DATA SOURCE	824
LIST NODE DIRECTORY	592	UNQUIESCE	825
LIST ODBC DATA SOURCES	595	UPDATE ADMIN CONFIGURATION	827
LIST PACKAGES/TABLES	596	UPDATE ALERT CONFIGURATION	829
LIST TABLESPACE CONTAINERS	599	UPDATE ALTERNATE SERVER FOR DATABASE	834
LIST TABLESPACES	601	UPDATE ALTERNATE SERVER FOR LDAP	
LIST UTILITIES	606	DATABASE	836
LOAD	608	UPDATE CLI CONFIGURATION	838
LOAD QUERY	630	UPDATE COMMAND OPTIONS	840
MIGRATE DATABASE	634	UPDATE CONTACT	842
PING	636	UPDATE CONTACTGROUP	843
PRECOMPILE	638	UPDATE DATABASE CONFIGURATION	844
PRUNE HISTORY/LOGFILE	665	UPDATE DATABASE MANAGER	
PUT ROUTINE	667	CONFIGURATION	847
QUERY CLIENT	669	UPDATE HEALTH NOTIFICATION CONTACT	
QUIESCE	670	LIST	849
QUIESCE TABLESPACES FOR TABLE	673	UPDATE HISTORY	850
QUIT	676	UPDATE LDAP NODE	852
REBIND	677	UPDATE MONITOR SWITCHES	854
RECONCILE	681		
RECOVER DATABASE	685	第 4 章 コマンド行 SQL ステートメン	
REDISTRIBUTE DATABASE PARTITION GROUP	692	トおよび XQuery ステートメントの使	
REFRESH LDAP	696	用	857
REGISTER	697		
REGISTER XMLSCHEMA	700	付録 A. 構文図の見方	865
REGISTER XSROBJECT	702		
REORG INDEXES/TABLE	705	付録 B. 命名規則	867
REORGCHK	716		
RESET ADMIN CONFIGURATION	726	付録 C. ファイル・タイプ修飾子	869
RESET ALERT CONFIGURATION	728	ロード・ユーティリティー用のファイル・タイプ修	
RESET DATABASE CONFIGURATION	730	飾子	869
RESET DATABASE MANAGER CONFIGURATION	732	インポート・ユーティリティー用のファイル・タイ	
RESET MONITOR	734	プ修飾子	884
RESTART DATABASE	736	エクスポート・ユーティリティー用のファイル・タ	
RESTORE DATABASE	738	イプ修飾子	896
REWIND TAPE	755	データ移動での区切り文字の制限.	902
ROLLFORWARD DATABASE	756		
RUNCMD	768	付録 D. DB2 Database 技術情報	905
RUNSTATS	769	DB2 技術情報の概説	905

資料についてのフィードバック	905
DB2 テクニカル・ライブラリー (PDF 形式)	906
DB2 の印刷資料の注文方法.	909
コマンド行プロセッサから SQL 状態ヘルプを表示する.	910
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	910
DB2 インフォメーション・センターにおける特定の言語でのトピックの表示	911
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	912

DB2 チュートリアル	914
DB2 トラブルシューティング情報	914
条件	915

付録 E. 特記事項	917
商標	919

付録 F. IBM と連絡をとる	921
-----------------------------------	------------

索引	923
---------------------	------------

本書について

本書は、データベース管理機能を実行するためのシステム・コマンドおよび DB2 コマンド行プロセッサ (CLP) の用法について説明します。

本書の対象読者

本書の読者は、データベース管理について理解しており、構造化照会言語 (SQL) の知識があることが前提となっています。

本書の構成

本書では、CLP を使用する際に必要となる参照情報を提供します。

以下のトピックについて説明します。

第 1 章

データベース・マネージャーにアクセスするためにオペレーティング・システム・コマンド・プロンプトまたはシェル・スクリプトに入力できるコマンドを記述します。

第 2 章

コマンド行プロセッサを呼び出し、使用方法と CLP オプションについて説明します。

第 3 章

すべてのデータベース・マネージャー・コマンドを説明します。

第 4 章

コマンド行から SQL ステートメントを使用する方法を示します。

付録 A

構文図の規則を説明します。

付録 B

データベースおよび表などのオブジェクトを命名するときの規則について説明します。

付録 C

ロード、インポート、およびエクスポート・ユーティリティー用のファイル・タイプ修飾子について説明します。

第 1 章 システム・コマンド

この章では、データベース・マネージャーへのアクセスおよび保守を行うためにオペレーティング・システム・コマンド・プロンプトまたはシェル・スクリプトで入力できるコマンドに関する情報を提供します。

注:

1. ディレクトリー・パスのスラッシュ (/) は UNIX ベース・システムに特有のものであり、Windows® オペレーティング・システムのディレクトリー・パスの円記号 (¥) に相当します。
2. Windows という用語は通常、サポートされているすべてのバージョンの Microsoft® Windows を指します。サポートされているバージョンには、Windows NT ベースのバージョンや Windows 9x ベースのバージョンも含まれます。その機能が Windows NT® 4、Windows 2000、Windows .NET および Windows XP ではサポートされているものの、Windows 9x ではサポートされていない場合、「Windows NT ベースのオペレーティング・システム」のように具体的に示されている場合もあります。特定のバージョンの Windows に固有の機能がある場合、有効なバージョン (複数の場合もある) のオペレーティング・システムが注記されます。

コマンドの説明の編成方法

各コマンドの短い説明の後に、以下の項目の一部またはすべてが続きます。

有効範囲:

インスタンス内でのコマンド操作の有効範囲。単一データベース・パーティション・システムでは、有効範囲はその単一データベース・パーティションに限定されます。マルチ・データベース・パーティション・システムでの有効範囲は、データベース・パーティション構成ファイル (db2nodes.cfg) に定義されている論理データベース・パーティションすべてです。

許可:

コマンドを正常に呼び出すために必要な権限。

必要な接続:

データベース、インスタンス、なし、接続が確立される、のいずれかです。関数にデータベース接続またはインスタンス接続機構が必要かどうか、または正常に操作を行うのに接続は必要ないかを示します。特定のコマンドを発行する前に、データベースへの明示的な接続またはインスタンスへのアタッチが必要である場合があります。データベース接続またはインスタンス・アタッチを必要とするコマンドは、ローカルまたはリモートのどちらかで実行することができます。データベース接続とインスタンス・アタッチのいずれも必要ではないコマンドはリモートには実行できません。そのため、そのようなコマンドをクライアント環境で発行すると、コマンドの影響はそのクライアント内にしか及びません。

システム・コマンド

コマンド構文:

構文図では、オペレーティング・システムが入力を正しく判別できるようなコマンドの指定方法を示します。構文図については、865 ページの『付録 A. 構文図の見方』を参照してください。

コマンド・パラメーター:

コマンドとともに使用可能なパラメーターの説明。

使用上の注意:

その他の情報です。

関連資料:

関連情報の相互参照です。

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。

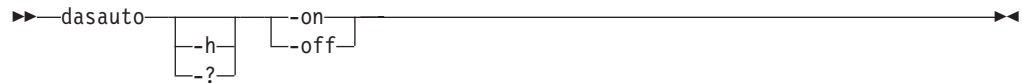
許可:

dasadm

必要な接続:

なし

コマンド構文:



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。

関連タスク:

- 「管理ガイド: インプリメンテーション」の『DB2 Administration Server (DAS) の構成』
- 「管理ガイド: インプリメンテーション」の『DB2 Administration Server (DAS) の作成』
- 「管理ガイド: インプリメンテーション」の『DB2 Administration Server (DAS) の開始と停止』

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 ベースのシステムでのみ使用できます。Windows オペレーティング・システムでは、同じ目的で **db2admin create** コマンドを使用できます。

許可:

Root 権限。

必要な接続:

なし。

コマンド構文:

```
▶▶ dasCRT -u DASuser -d
```

コマンド・パラメーター:

-u *DASuser*

DASuser は、DAS を作成するときに使用するユーザー ID です。DAS は、/home/*DASuser*/das ディレクトリーの下に作成されます。

-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。

関連タスク:

- 「管理ガイド: インプリメンテーション」の『DB2 Administration Server (DAS) の構成』
- 「管理ガイド: インプリメンテーション」の『DB2 Administration Server (DAS) の作成』
- 「管理ガイド: インプリメンテーション」の『DB2 Administration Server (DAS) の開始と停止』

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。

関連タスク:

- 「管理ガイド: インプリメンテーション」の『DB2 Administration Server (DAS) の削除』

dasmigr - DB2 Administration Server の移行

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移行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移行します。

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 ディレクトリーにあります。 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 DB2DIR\bin 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:

dasmigr - DB2 Administration Server の移行

Linux および UNIX システムの場合

DB2DIR/instance/dasmig

Windows オペレーティング・システムの場合

dasmigr db2as dasusr1

関連タスク:

- 「管理ガイド: インプリメンテーション」の『DB2 Administration Server (DAS) の構成』
- 「マイグレーション・ガイド」の『DB2 Administration Server (DAS) の移行』

関連資料:

- 8 ページの『dasupdt - DAS の更新』

dasupdt - DAS の更新

Linux および UNIX ベースのシステムで、関連した DB2 データベース・システム・インストールが更新された場合、DB2 Administration Server (DAS) を更新します。このユーティリティーは、DB2DIR/instance ディレクトリーにあります。DB2DIR は、DB2 データベース製品の現行バージョンがインストールされている場所です。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。

Windows オペレーティング・システムでは、同じ DB2 データベース・リリース内の 1 つの DB2 コピーから別のコピーに DAS を更新します。DAS を以前のリリースから移動するには、**dasmigr** コマンドを使用します。DAS は、**dasupdt** コマンドの実行元の DB2 コピーに移動されます。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 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:

Linux および UNIX ベースのシステムでは 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 ベースのシステムの場合

```

>> dasupdt [-d] [-D] [-h] [-?]

```

Windows オペレーティング・システムの場合

```

>> dasmigr [-h] [-p path override]

```

コマンド・パラメーター:

Linux および UNIX ベースのシステムの場合

- 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 または -?** 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:

フィックスパックに関連した DAS のある DB2 インストール・パスに適用した場合、以下のコマンドをインストール・パスから発行すると、適用されたフィックスパックに DAS を更新します。

```
dasupdt
```

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
```

関連タスク:

- 「管理ガイド: インプリメンテーション」の『DB2 Administration Server (DAS) の構成』
- 「管理ガイド: インプリメンテーション」の『ディスクバリーのための DB2 Administration Server (DAS) 構成の更新』
- 「管理ガイド: インプリメンテーション」の『UNIX での DB2 Administration Server (DAS) の更新』

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

db2_deinstall ツールと同じパスにあるインストール済みの DB2 製品またはフィーチャーをアンインストールします。これは Linux および UNIX システムでのみ使用できます。

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、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:

Root

必要な接続:

なし。

コマンド構文:

```

▶ db2_deinstall [-F feature-name] [-a] [-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。

-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。

-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/db2_deinstall.log\$\$ です。ここで \$\$ 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:

- 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** を発行します。

```
DB2DIR/install/db2_deinstall -a
```

関連資料:

- 167 ページの『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』

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 ベースのシステムでのみ利用できます。

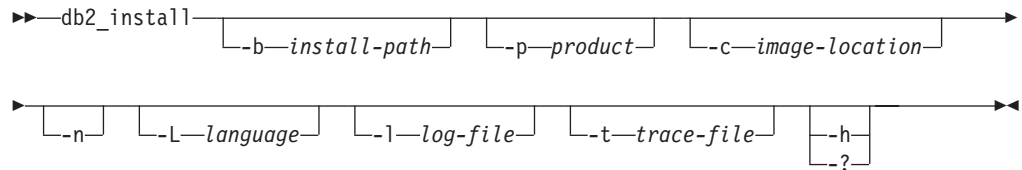
許可:

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 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。

-p *productID*

インストールする DB2 製品を指定します。 *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。

-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、および自動ディスカ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。複数インストール・イメージに関連した自動ディスカバリーについて詳しくは、複数の CD によるインストール (Linux と UNIX) を参照してください。

-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。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。

db2_install - DB2 製品のインストール

-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/db2_install.log\$\$ です。ここで \$\$ 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。 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
```

関連タスク:

- 「インストールおよび構成 補足」の『db2_install または doce_install コマンドによる DB2 製品のインストール (Linux および UNIX)』

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 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)。

許可:

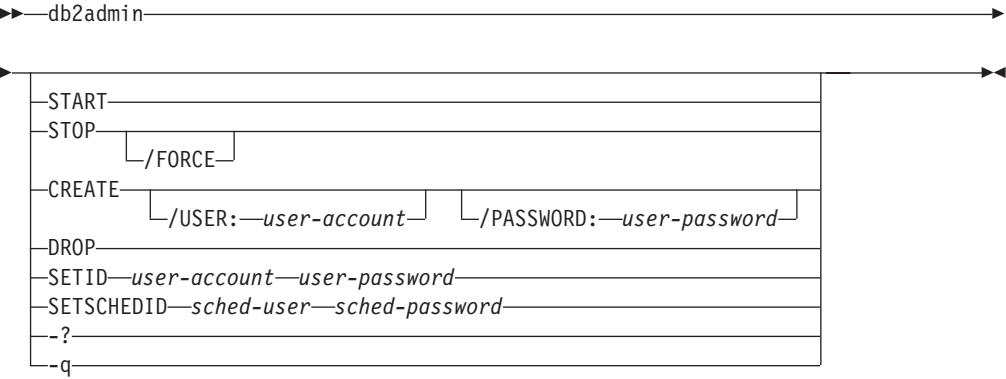
UNIX オペレーティング・システムでは *dasadm*、ただし 64 ビット・インスタンスには関連付けられません。

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:



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、**dasCRT** コマンドを使用します。

DROP DAS を削除します。UNIX オペレーティング・システム上の DAS をドロップするには、**dasdrop** コマンドを使用する必要があります。

SETID *user-account/user-password*

DAS に関連付けられたユーザー・アカウントを設定または変更します。

SETSCHEID *sched-user/sched-password*

ツール・カタログ・データベースに接続するためにスケジューラーで使用するログオン・アカウントを確立します。これは、スケジューラーが有効になっている場合で、ツール・カタログ・データベースが DAS から見てリモートである場合にのみ必要です。スケジューラーについての詳細は、「管理ガイド」を参照してください。

-? ヘルプ情報を表示します。このオプションを指定すると、他のすべてのオプションは無視され、ヘルプ情報だけが表示されます。

-q **db2admin** コマンドを静止モードで実行します。コマンドが実行されるときにもメッセージは表示されません。このオプションは、他のすべてのコマンド・オプションと組み合わせて使用できます。

関連タスク:

- 「管理ガイド: インプリメンテーション」の『ツール・カタログ・データベースと DB2 Administration Server (DAS) のスケジューラーのセットアップと構成』

関連資料:

- 4 ページの『dasCRT - DB2 Administration Server の作成』
- 5 ページの『dasdrop - DB2 Administration Server の除去』

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 `sqlllib/adsm` ディレクトリーにあります。Windows オペレーティング・システムでは、これは `sqlllib¥bin` にあります。

許可:

なし

必要な接続:

なし

コマンド構文:

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

db2-object-options:

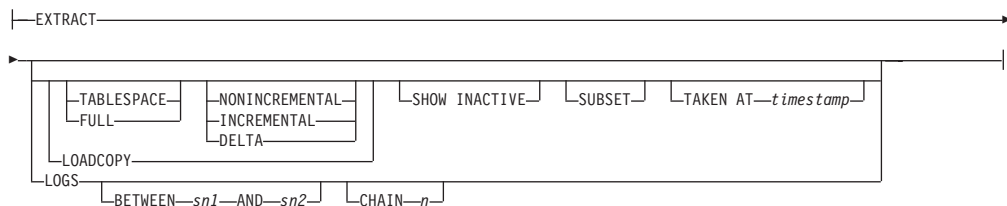
```
[QUERY-options] [EXTRACT-options] [DELETE-options] [VERIFY-options]
[COMPRLIB—decompression-library]
[COMPROPTS—decompression-options] [VERBOSE]
[DATABASE—database_name] [DBPARTITIONNUM—db-partition-number]
[DB]
[PASSWORD—password] [NODENAME—node_name] [OWNER—owner]
[WITHOUT PROMPTING]
```

QUERY-options:

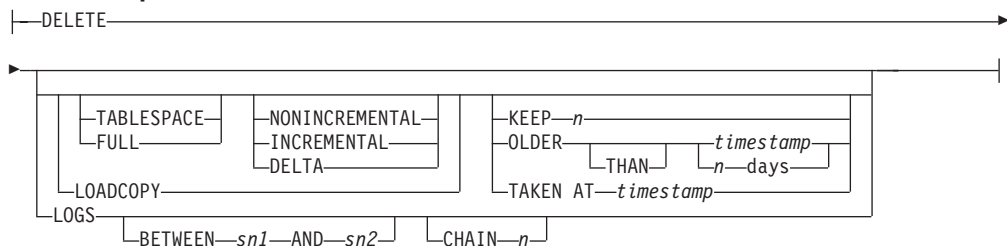
```
[QUERY]
[TABLESPACE—FULL] [NONINCREMENTAL—INCREMENTAL—DELTA] [SHOW INACTIVE]
[LOADCOPY]
[LOGS] [BETWEEN—sn1—AND—sn2] [CHAIN—n]
```

db2adutl - TSM 内の DB2 オブジェクトの管理

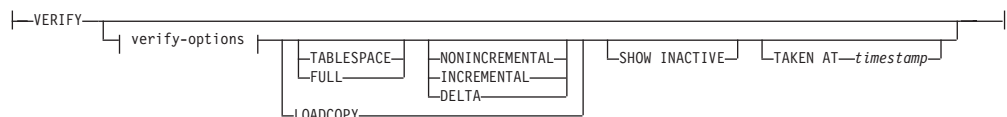
EXTRACT-options:



DELETE-options:



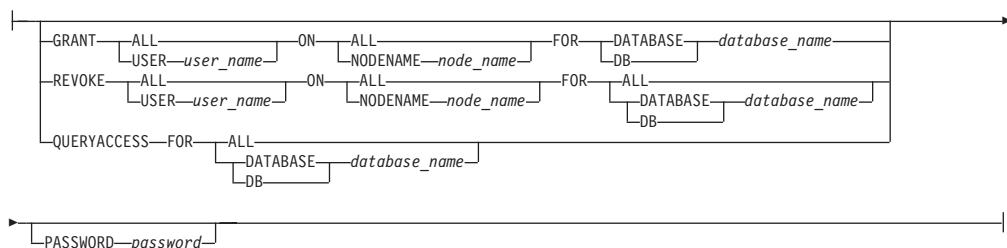
VERIFY-options:



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、

O <objType> <startPos> <numBytes>

デフォルトの出力ファイルは `extractPage.out` です。デフォルトの出力ファイル名は、**DB2EXTRACTFILE** 環境変数に絶対パスを設定することによりオーバーライドできます。

TAKEN AT *timestamp*

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、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

ユーザーまたは指定したユーザーに付与します。アクセス権をユーザーに付与すると、指定されたデータベースに関連する現在のファイルと将来のファイルのすべてのアクセス権を付与することになります。

REVOKE ALL / USER *user_name*

現在の TSM ノード上の TSM ファイルに対するアクセス権を、すべてのユーザーまたは指定したユーザーから削除します。

QUERYACCESS

現在のアクセス・リストを取り出します。ユーザーと TSM ノードのリストが表示されます。

ON ALL / NODENAME *node_name*

アクセス権を変更する TSM ノードを指定します。

FOR ALL / DATABASE *database_name*

対象となるデータベースを指定します。

例:

1. 以下に示すのは、db2 backup database rawsampl use tsm コマンドの出力例です。

```
Backup successful. The timestamp for this backup is : 20031209184503
```

以下に示すのは、バックアップ操作の後で発行された db2adutl query コマンドの出力例です。

```
Query for database RAWSAMPL
```

```
Retrieving FULL DATABASE BACKUP information.
```

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

```
Retrieving INCREMENTAL DATABASE BACKUP information.
```

```
No INCREMENTAL DATABASE BACKUP images found for RAWSAMPL
```

```
Retrieving DELTA DATABASE BACKUP information.
```

```
No DELTA DATABASE BACKUP images found for RAWSAMPL
```

```
Retrieving TABLESPACE BACKUP information.
```

```
No TABLESPACE BACKUP images found for RAWSAMPL
```

```
Retrieving INCREMENTAL TABLESPACE BACKUP information.
```

```
No INCREMENTAL TABLESPACE BACKUP images found for RAWSAMPL
```

```
Retrieving DELTA TABLESPACE BACKUP information.
```

```
No DELTA TABLESPACE BACKUP images found for RAWSAMPL
```

```
Retrieving LOCAL COPY information.
```

```
No LOCAL COPY images found for RAWSAMPL
```

```
Retrieving log archive information.
```

```
Log file: S0000050.LOG, Chain Num: 0, 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 DBPARTITIONNUM の代わりに NODE を使用できます。

関連概念:

- 「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『db2adutl コマンドと logarchopt1 および vendoropt データベース構成パラメーターを使ったノード間リカバリー』

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。

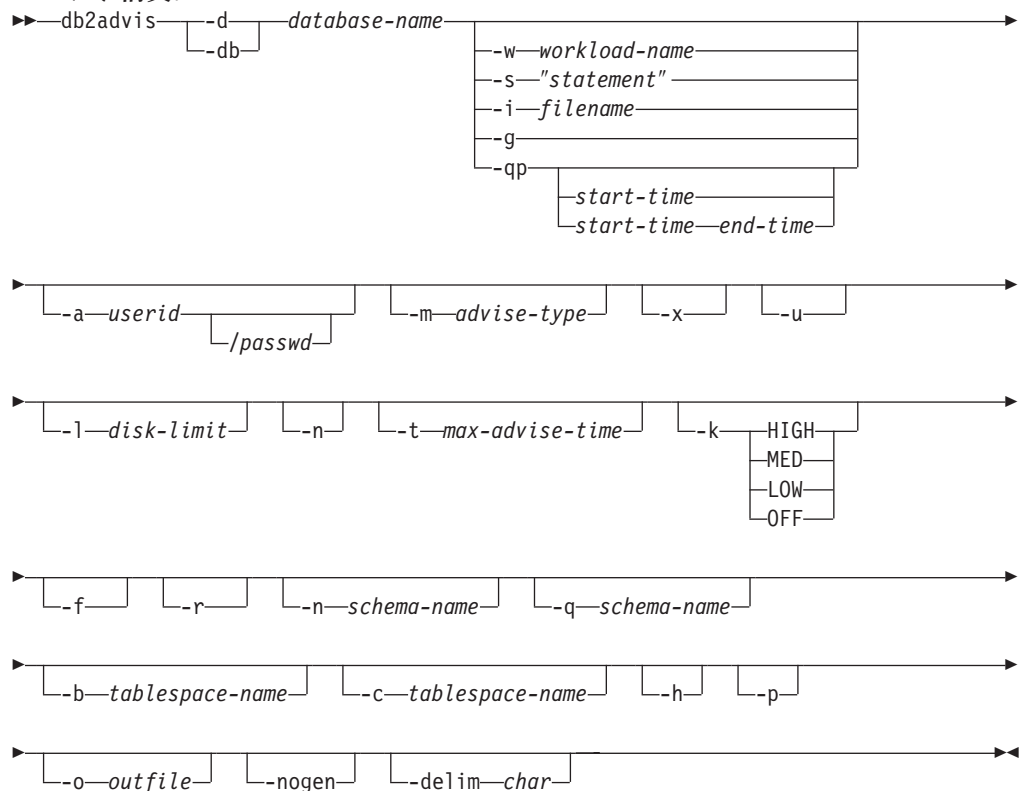
許可:

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、-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 -- <注釈> で表します。ステートメントは必ずセミコロン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 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、-s、-i、-g の各オプションと一緒に使用できません。

-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、db2advis -m PC は、パーティション化と MDC 表を推奨します。パーティション・

データベース環境で `-m P` または `-m M` を使用した場合、`advise_partition` 表には最終的なパーティション推奨事項が入ります。選択可能な値は以下のとおりです。

I 新規索引を推奨します。これがデフォルトです。

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、アドバイザーは実行処理を簡略化しますが、ワークロード全体の場合と同様の結果を生成します。**HIGH** の場合、アドバイザーはワークロードの小さなサブセットを対象にします。**MED** の場合、アドバイザーはワークロードの中程度のサブセットを対象にします。**LOW** の場合、アドバイザーはワークロードの大きなサブセットを対象にします。**OFF** の場合は、圧縮を行いません。デフォルトは **MED** です。

-f 以前存在していたシミュレーション・カタログ表をドロップします。

-r 仮想 MQT とパーティション化の選択に詳細な統計を使用することを指定します。このオプションを指定しない場合は、デフォルトで MQT にオプティマイザー統計が使用されます。詳細な統計のほうが正確ですが、これを得るには時間がかかるので、**db2advis** の実行時間が長くなります。`-r` コマンド・パラメーターは MQT およびパーティション化に関連した統計を取得するためにサンプリングを使用します。MQT に関しては、サンプル照会が失敗するか行を 1 つも戻さない場合には、オプティマイザー見積もりが使用されます。

-n *schema-name*

シミュレーション・カタログ表の修飾名、および新しい索引と MQT の修飾子を指定します。デフォルトのスキーマ名は、呼び出し側のユーザー ID です。ただし、カタログ・シミュレーション表の場合は例外で、デフォルトのスキーマ名は SYSTOOLS になります。デフォルトでは、新しい索引が索引のベースのスキーマ名を継承します。

-q *schema-name*

ワークロード内の非修飾名の修飾名を指定します。これは、**db2advis** の実行時に CURRENT SCHEMA に使用するスキーマ名になります。デフォルトのスキーマ名は、コマンドの実行者のユーザー ID です。

-b *tablespace-name*

新規 MQT を作成する表スペースの名前を指定します。指定しない場合、アドバイザーは、存在する表スペースのセットから表スペースを選択します。

-c *tablespace-name*

シミュレーション・カタログ表を作成する表スペースの名前を指定します。表スペースは任意のタイプが可能で、例えばファイル名やディレクトリーを使用できます。この表スペースは、カタログ・データベース・パーティション・グループでのみ作成されなければなりません。デフォルトは USERSPACE1 です。

デフォルトの USERSPACE1 を使用する代わりに、シミュレーションに使用する表スペースを作成することをお勧めします。さらに、db2advis ユーティリティーのパフォーマンスを改善するために、ALTER TABLESPACE DROPPED TABLE RECOVERY OFF ステートメントをこの表スペースに対して実行してください。このユーティリティーが完了すると、表スペースの履歴が元に戻ります。パーティション・データベース環境では、通常、USERSPACE1 がすべてのパーティション・グループにまたがって作成されるため、このオプションは必須です。

-h ヘルプ情報を表示します。 このオプションを指定すると、他のすべてのオプションは無視され、ヘルプ情報だけが表示されます。

-p Explain 表でツールを実行した際に生成されたプランを保持します。**-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。

例:

1. 次の例では、ユーティリティーはデータベース PROTOTYPE に接続し、解決策において制限なしで表 ADDRESSES に索引を推奨します。

```
db2advis -d prototype -s "select * from addresses a
where a.zip in ('93213', '98567', '93412')
and (company like 'IBM%' or company like '%otus')"
```

2. 次の例では、ユーティリティーは PROTOTYPE データベースに接続し、ADVISE_WORKLOAD 表の照会用として 53 MB を超えない索引を推奨します。ワークロード名は "production" と同じです。解決策を見つけるための最大許可時間は 20 分です。

```
db2advis -d prototype -w production -l 53 -t 20
```

3. 次の例では、db2advis.in という入力ファイルに SQL ステートメント、および各ステートメントが実行される頻度の指定が含まれています。

```
--#SET FREQUENCY 100
SELECT COUNT(*) FROM EMPLOYEE;
SELECT * FROM EMPLOYEE WHERE LASTNAME='HAAS';
--#SET FREQUENCY 1
SELECT AVG(BONUS), AVG(SALARY) FROM EMPLOYEE
GROUP BY WORKDEPT ORDER BY WORKDEPT;
```

ユーティリティーはデータベース SAMPLE に接続し、入力ファイル内の照会によって参照される各表ごとに索引を推奨します。解決策を見つけるための最大許可時間は 5 分です。

```
db2advis -d sample -f db2advis.in -t 5
```

4. 次の例では、表スペース SPACE1 内に MQT を作成します。シミュレーション表スペースは SPACE2 です。ワークロード内の非修飾名の修飾名は SCHEMA1、新規 MQT の推奨を行うスキーマ名は SCHEMA2 です。使用するワークロード圧縮は HIGH、ディスク・スペースは無制限です。MQT にはサンプル統計を使用します。以下のコマンドを実行すると、MQT の推奨が行われ、パーティション・データベース環境では、索引とパーティション化の推奨も行われます。

```
db2advis -d prototype -w production -l -l -m M -b space1 -c space2 -k
HIGH -q schema1 -n schema2 -r
```

推奨された MQT と、MQT と基本表の両方の索引、パーティション化、および MDC を取得するには、-m オプションの値 IMCP を指定して以下のようにコマンドを発行します。

```
db2advis -d prototype -w production -l -l -m IMCP -b space1 -c space2 -k
HIGH -q schema1 -n schema2 -r
```

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、これらの機能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 コマンド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. 次のように発行します。

```
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. 次のように発行します。

```
db2advis -g <other-options>
```

-p パラメーターが -g パラメーターと共に使用されている場合、取得される動的 SQL ステートメントは ADVISE_WORKLOAD 表に、タイム・スタンプを含む生成されたワークロード名と共に置かれます。

ワークロードの各 SQL ステートメントのデフォルト頻度は 1 で、デフォルトの重要度も 1 です。generate_unique() 関数は、その SQL ステートメントのより分かりやすい説明になるように、ユーザーによって更新できるステートメントに、固有 ID を割り当てます。

すべての db2advis エラー情報は db2diag.log にも入っています。

アドバイザーの実行が始まると、アドバイザーを識別する行が ADVISE_INSTANCE 表に含まれるようになります。メイン・アドバイザー行は、アドバイザーの開始時間を示す START_TIME によって識別されます。この行の STATUS は「STARTED」です。

db2advis コマンドを発行した結果、「Cannot insert into DB2ADVIS_INSTANCE」というエラーが発生した場合には、db2advis.bnd をバインドし、-l オプションを使って **db2advis** コマンドを実行する必要があります。バインド操作を実行するには、db2 bind db2advis.bnd blocking all grant public を発行します。

アドバイザーが完了すると、ADVISE_INSTANCE 表内で適切な START_TIME を持つ関連行を検査することができます。STATUS が「COMPLETED」であれば、アドバイザーは正常に実行されました。STATUS がまだ「STARTED」で、実行中の db2advis プロセスが存在しない場合は、アドバイザーが異常終了しました。STATUS に「EX」が含まれる場合、アドバイザーの失敗原因を判別するための「SQLCODE」も表示されます。

-l disk-limit option 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。

関連概念:

- 「パフォーマンス」の『設計アドバイザー』

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・レベルで動作し、すべてのインスタンス・レベルの活動とデータベース・レベルの活動を記録します。

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。

監査ログ (db2audit.log) および監査構成ファイル (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。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。
- 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。
- 記録する監査可能イベントのカテゴリーを選択する。
- 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。
- 形式設定して監査ログからコピーすることにより、監査レコードをフラット・ファイルまたは ASCII 区切りファイルに抽出する。抽出を行う理由は 2 つのうちどちらかです。ログ・レコードの分析を準備するためか、ログ・レコードの整理を準備するためです。
- 現在の監査ログから監査レコードを整理する。

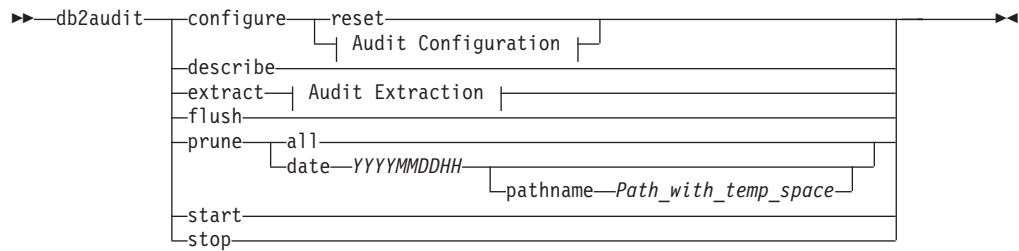
許可:

sysadm

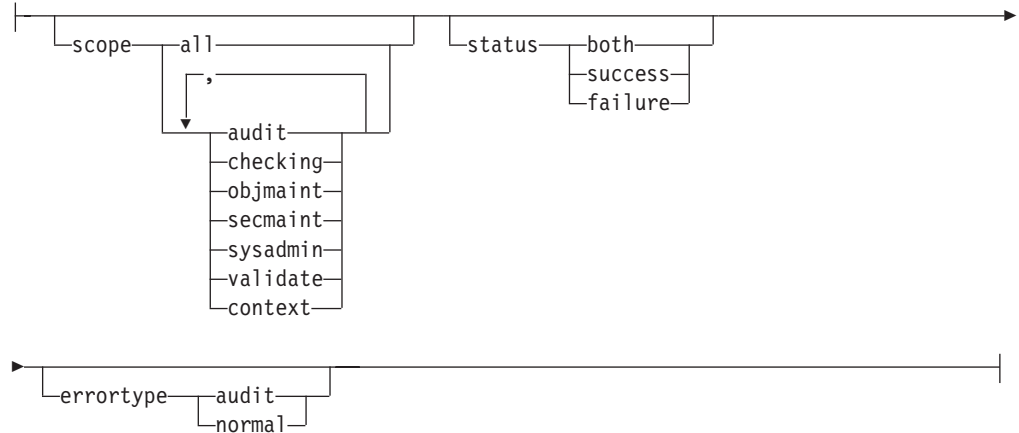
必要な接続:

なし。

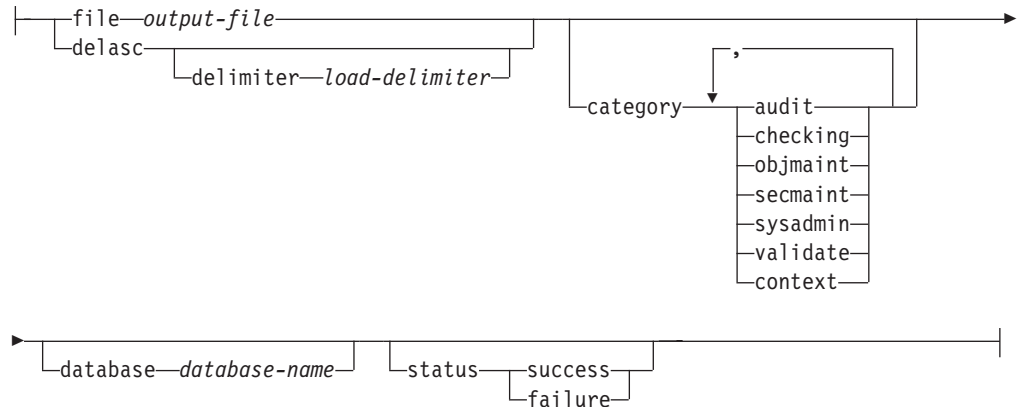
コマンド構文:



Audit Configuration:



Audit Extraction:



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、インスタンスがシャットダウン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、すべてのデータベース・パーティションを通じて DB2 データベー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。

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 を除く全カテゴリー、STATUS は FAILURE、ERRORTYPE は NORMAL、および監査機能は 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。
- **SCOPE**。このアクションは、イベントのどのカテゴリーが監査対象かを指定します。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。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が勧められています。そうしないと、監査ログはすぐに肥大化してしまいます。デフォルトの SCOPE は CONTEXT を除く全カテゴリーであり、その場合には短時間で大量のレコードが生成されてしまう可能性があります。カテゴリーの選択は、モード (同期か非同期か) と共にパフォーマンス低下の大きな原因となる可能性があり、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。
- **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。コンテキスト・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、このパラメーターに関連する値には関係なくログに記録する必要があります。
- **ERRORTYPE**。このアクションは、監査エラーがユーザーに戻されるのか、それとも無視されるのかを指定します。このパラメーターの値として可能なのは、以下のとおりです。
 - **AUDIT**。監査機能内で発生するエラーも含めて、すべてのエラーが DB2 データベースによって管理され、負の SQLCODE はすべて呼び出し元に報告されます。
 - **NORMAL**。db2audit によって生成されるエラーは無視され、実行されている操作に関連したエラーの SQLCODE だけがアプリケーションに戻されます。

describe

このパラメーターは、現行の監査の構成情報および状況を標準出力に出力します。

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。オプションの文節が何も指定されていない場合は、すべての監査レコードが抽出され、フラット・レポート・ファイルに入れられます。output_file が既に存在するなら、エラー・メッセージが戻されます。

抽出時に使用できるオプションは、以下のとおりです。

- **FILE**。抽出された監査レコードは、ファイル (output_file) に入れられます。ファイル名が指定されていない場合、レコードは、sqllib の security サブディレクトリの中にある db2audit.out ファイルに入れられます。ディレクトリが指定されていない場合、output_file は、現行作業ディレクトリに書き込まれます。

- 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

```

これらのファイルは、常に sqllib の security サブディレクトリーに書き込まれます。

さらに DELASC 選択項目を使用すると、監査ログからの抽出処理において、デフォルトの監査文字ストリング区切り (『0xff』) をオーバーライドできます。DELASC DELIMITER の後に、監査レコードが入れられる表にロードするための準備処理において使用する新しい区切り文字を指定できます。新しいロード区切り文字は、単一の文字 (! など) か、または 16 進数表記の 4 バイトのストリング (0xff など) のいずれかです。

- CATEGORY。監査イベントのうち指定されたカテゴリー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、すべてのカテゴリーが抽出対象になります。
- DATABASE。指定されたデータベース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、すべてのデータベース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、すべてのレコードが抽出対象になります。

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、エンジン内で監査状態が「ログ記録不可能」から「ログ記録可能」の状態にリセットされます。

prune このパラメーターを使用すると、監査ログから監査レコードを削除できます。監査機能がアクティブになっており、イベントの「audit」(監査) カテゴリーが監査対象として指定されている場合、監査レコードは監査ログの整理処理が実行された後にログに記録されます。

整理処理において使用できるオプションは、以下のとおりです。

- ALL。監査ログ内のすべての監査レコードを削除します。
- DATE yyyyymmddhh。ユーザーは、指定された日時以前に発生したすべての監査レコードを監査ログから削除するよう指定できます。ユーザーはオプションとして

pathname

を指定できます。これは、監査機能が監査ログの整理処理において一時スペースとして使用するパス名です。この一時スペースにより、監査ログが

含まれるディスクがいっぱいになって、整理操作のためのスペースが不足するようになった場合にも、監査ログの整理処理の実行が可能になります。

- start** このパラメーターを使用すると、db2audit.cfg ファイルの内容に基づいたイベント監査処理が、監査機能によって開始されます。この文節が指定されている場合、DB2 のパーティション・データベース・インスタンスにおいて、すべてのデータベース・パーティションに関して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。
- stop** このパラメーターを使用すると、監査機能によってイベントの監査処理が停止されます。この文節が指定されている場合、DB2 のパーティション・データベース・インスタンスにおいて、すべてのデータベース・パーティションに関して監査が停止します。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。

使用上の注意:

- 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は、以下のとおりです。
 - 監査 (AUDIT)。監査設定値が変更された場合、または監査ログへのアクセスがあった場合に、レコードが生成されます。
 - 許可検査 (CHECKING)。DB2 データベース・オブジェクトまたは関数にアクセスしたり、それらを操作したりする試みに関する許可検査中に、レコードが生成されます。
 - オブジェクト保守 (OBJMAINT)。データ・オブジェクトの作成時またはドロップ時にレコードが生成されます。
 - セキュリティ保守 (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。また、データベース・マネージャー・セキュリティ構成パラメーター SYSADM_GROUP、SYSCTRL_GROUP、または SYSMAINT_GROUP が変更された時点においても、レコードが生成されます。
 - システム管理 (SYSADMIN)。SYSADM、SYSMAINT、または SYSCTRL の権限を必要とする操作が実行された時点で、レコードが生成されます。
 - ユーザー妥当性検査 (VALIDATE)。ユーザー認証時、またはシステムのセキュリティ情報の検索時に、レコードが生成されます。

- 操作コンテキスト (CONTEXT)。データベー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ための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。例えば、動的照会の照会ステートメント、静的照会のパッケージ ID、または実行中の操作のタイプの標識 (CONNECT など) により、監査結果分析時に必要とされるコンテキストを提供できます。操作コンテキストを提供する SQL または XQuery ステートメントは非常に長いものになる場合がありますが、CONTEXT レコード内に完全に示されます。そのため、CONTEXT レコードは非常に大きくなる場合があります。
- 監査の対象として指定できるのは、失敗、成功、またはその両方です。
- データベースに対する 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。

関連タスク:

- 「管理ガイド: インプリメンテーション」の『DB2 監査データ・ファイルの作成』

関連資料:

- 「管理ガイド: インプリメンテーション」の『監査機能の使用方法』

db2batch - ベンチマーク・ツール

フラット・ファイルまたは標準入力から SQL ステートメントおよび XQuery ステートメントを読み取り、ステートメントを動的に準備および記述し、応答セットを戻します。

このツールは、単一パーティション・データベースと複数パーティション・データベースの両方で機能できます。

このツールのオプション・パラメーターでは、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。

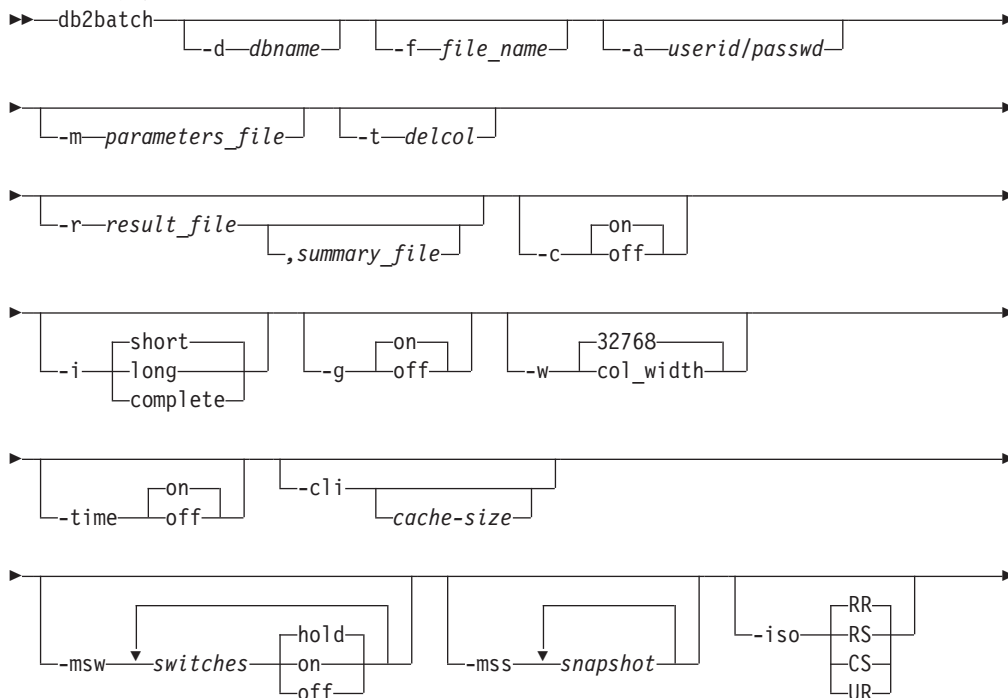
許可:

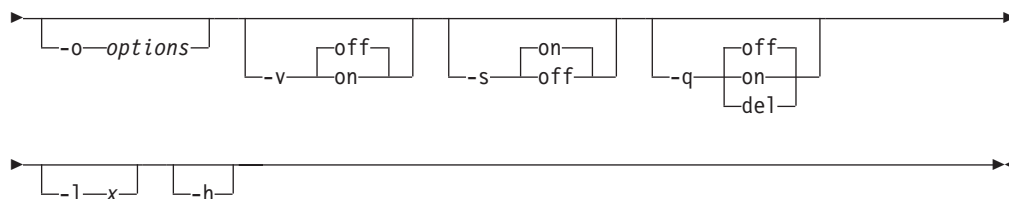
読み取られる SQL ステートメントまたは XQuery ステートメントが必要とするものと同じ権限レベル。

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 <コメント>` 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 <制御オプション> <値>`。有効な制御オプションは、以下のとおりです。

ROWS_FETCH

応答セットから取り出す行数。有効な値は -1 から *n*、デフォルト値は -1 (すべての行を取り出す) です。

ROWS_OUT

取り出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*、デフォルト値は -1 (取り出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。
(自動コミットがオフになっており、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は、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。
(自動コミットがオフになっており、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は、ステートメントのスナップショットが戻されます。) バッファ・プール、表スペース、および FCM も戻します
(FCM スナップショットはマルチ・データベース・パーティション環境でのみ使用できる)。スナップショットにはハッシュ結合情報は組み込まれません。

デフォルト値は 1 です。1 より大きい値は DB2 バージョン 2 と DB2 データベース・サーバーでのみ有効で、ホスト・マシンでは現在サポートされていません。

ERROR_STOP

重大ではないエラーが発生したとき、**db2batch** を停止させるかどうかを指定します。有効な値は以下のとおりです。

- no** 重大ではないエラーが発生しても実行を続ける。これは、デフォルト・オプションです。
- yes** 重大ではないエラーが発生したら実行を停止する。

DELIMITER

1 文字か 2 文字のステートメント終結区切り文字です。デフォルト値はセミコロン (;) です。

SLEEP

スリープの秒数。有効な値は 1 ～ *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 スナップショット

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。これがデフォルトです。

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'...'	Nu1l

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の (:) 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。これがデフォルトです。

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、取り出し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。これがデフォルトです。

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 2G)。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、より正確な取り出し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。これがデフォルトです。

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。これがデフォルトです。

on スwitchをオンに切り替える。

off スwitchをオフに切り替える。

-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*、*dcsl_applinfo_all*、*db2*、*dbase*、*dbase_all*、*dcsl_dbase*、*dcsl_dbase_all*、*dbase_remote*、*dbase_remote_all*、*agent_id*、*dbase_appls*、*appl_all*、*dcsl_appl_all*、*dcsl_appl_handle*、*dcsl_dbase_appls*、*dbase_appls_remote*、*appl_remote_all*、*dbase_tables*、*appl_locks_agent_id*、*dbase_locks*、*dbase_tablespace*s、*bufferpools_all*、*dbase_bufferpools*、および *dynamic_sql*。

特殊なスナップショット *all* は、上記のすべてのスナップショットを取得します。アプリケーション ID を扱うスナップショットはサポートされず、代わりに対応するエージェント ID (アプリケーション・ハンドル) が使用されます。デフォルトでは、モニター・スナップショットはまったく取得されません。

-iso データ・アクセス中にデータをロックして他のプロセスから分離する方法を決定する、分離レベルを指定します。デフォルトでは、**db2batch** は分離レベル *RR* を使用します。

db2cli.ini ファイル内の構成キーワード *TxnIsolation* は、**db2batch** に影響を与えません。分離レベルを *RR* 以外に設定してこのコマンドを実行するには、*-iso* パラメーターを指定する必要があります。

RR 反復可能読み取り (ODBC 逐次化可能)。これがデフォルトです。

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

応答セットから取り出す行数。有効な値は -1 から *n*、デフォルト値は -1 (すべての行を取り出す) です。

r rows_out

取り出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*、デフォルト値は -1 (取り出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。
(自動コミットがオフになっており、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は、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。
(自動コミットがオフになっており、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は、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)、取り出された行、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。これがデフォルトです。

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 x

終了文字 (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:

- 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, 0.000000000000005, 'demodemo');
```

* Elapsed Time is: 0.005676 seconds

* SQL Statement Number 4:

```
select * from demo;
```

C1	C2	C3
-9223372036854775808	-5.00000000000000E-015	demo
9223372036854775807	+5.00000000000000E-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。

db2batch - ベンチマーク・ツール

- `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。

関連概念:

- 「管理ガイド: プランニング」の『分離レベルについて』

関連資料:

- 269 ページの『db2sql92 - SQL92 準拠 SQL ステートメント・プロセッサ』

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。

関連概念:

- 「組み込み SQL アプリケーションの開発」の『組み込み SQL パッケージのデータベースへのバインド』
- 「問題判別ガイド」の『db2bfd ツールを使用したバインド・ファイルの内容の表示』

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・マネージャー構成、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。

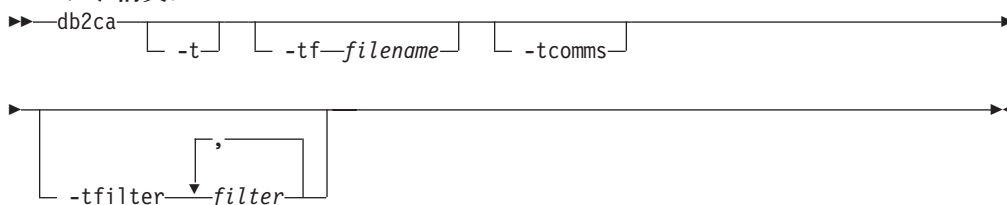
許可:

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** コマンドにはコンソール・ウィンドウがありません。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 ベースのシステムでは /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。

関連資料:

- 408 ページの『CATALOG DATABASE 』
- 412 ページの『CATALOG DCS DATABASE 』
- 426 ページの『CATALOG TCPIP/TCPIP4/TCPIP6 NODE 』
- 265 ページの『db2set - DB2 プロファイル・レジストリー』
- 502 ページの『GET DATABASE CONFIGURATION 』
- 730 ページの『RESET DATABASE CONFIGURATION 』
- 844 ページの『UPDATE DATABASE CONFIGURATION 』

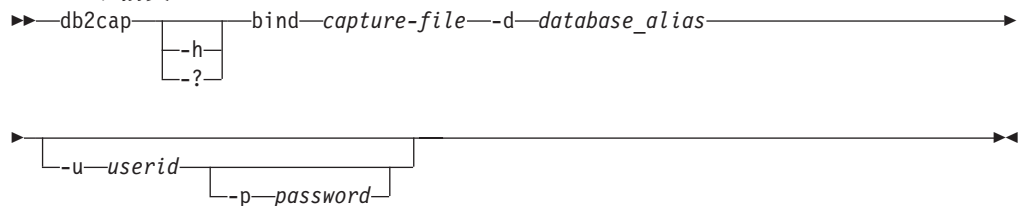
db2cap - CLI/ODBC 静的パッケージ・バインディング・ツール

キャプチャー・ファイルを 1 つ以上の静的パッケージにバインドします。キャプチャー・ファイルは、CLI/ODBC/JDBC 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はキャプチャー・ファイル进行处理して、アプリケーションが静的 SQL を実行するために CLI/ODBC/JDBC 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。

-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。

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。

関連タスク:

- 「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」の『CLI/ODBC/JDBC 静的プロファイル作成による静的 SQL の作成』

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。

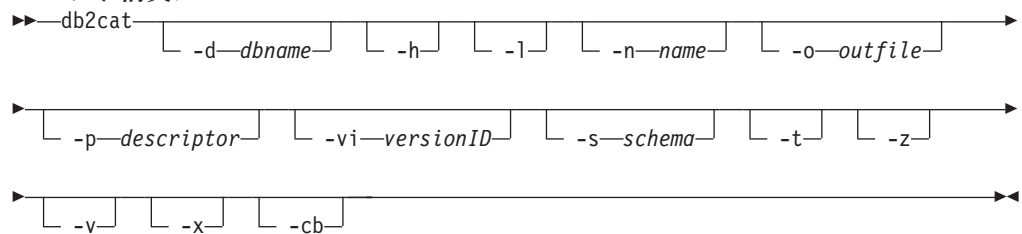
許可:

なし。

必要な接続:

なし。

コマンド構文:



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 オブジェクト名のケース・センシティブをオンにします。

-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 参照保全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 のバック記述子を指定します。 **-vi** は、**-p sysplan** が指定された場合にのみ有効です。 *versionID* が省略された場合、デフォルトは空ストリングです。

-s *schema*

オブジェクト・スキーマの名前を指定します。

-t 端末出力を表示します。

-z キー・ストローク・プロンプトを無効にします。

-v バック記述子を妥当性検査します。このパラメーターは、表のバック記述子のみで有効です。

-x カタログ内の表スペース・エクステンツ・サイズ (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。

関連資料:

- 「SQL リファレンス 第 1 巻」の『システム・カタログ・ビュー』

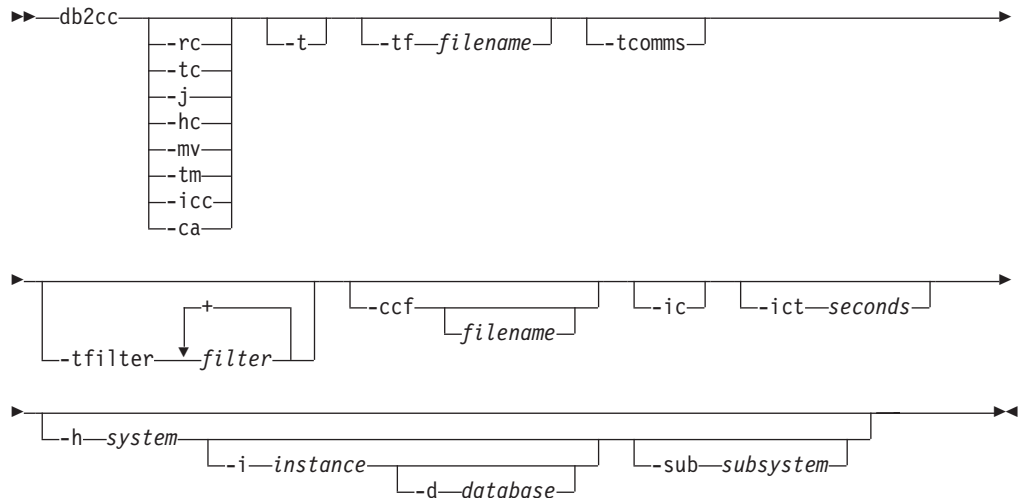
db2cc - コントロール・センターの開始

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 file name

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。

関連資料:

- 486 ページの『GET ADMIN CONFIGURATION 』
- 726 ページの『RESET ADMIN CONFIGURATION 』
- 827 ページの『UPDATE ADMIN CONFIGURATION 』

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 (例えば、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:

```
db2cfexp filename [TEMPLATE | BACKUP | MAINTAIN]
```

コマンド・パラメーター:

filename

ターゲット・エクスポート・ファイルの完全修飾名を指定します。このファイルは、構成プロファイルと呼ばれます。

TEMPLATE

同じインスタンス・タイプ (例えば、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。

関連タスク:

- 「インストールおよび構成 補足」の『プロファイルのエクスポートとインポート』

db2cfimp - 接続構成インポート・ツール

接続構成情報を、構成プロファイルというファイルからインポートします。これは、構成プロファイル中で見つかるすべての情報をインポートしようとする、非対話式ユーティリティです。

構成プロファイルには、次のような接続項目を含めることができます。

- データベース情報 (DB2 Connect および ODBC 情報を含む)
- ノード情報
- プロトコル情報
- データベース・マネージャ構成設定
- DB2 データベース・レジストリー設定
- 共通 ODBC/CLI 設定

このユーティリティを使用すると、前に構成された同様の別のインスタンスから (つまりクライアント・インスタンスからクライアント・インスタンスへ)、接続情報を複製することが可能になります。このユーティリティは、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*

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。

関連タスク:

- 「*DB2 Universal Database* クライアント機能 概説およびインストール」の『構成アシスタントによるクライアント・プロファイルを用いたデータベース接続の構成』
- 「インストールおよび構成 補足」の『プロファイルのエクスポートとインポート』

db2chglibpath - 組み込みランタイム・ライブラリー検索パスの変更

組み込みランタイム・ライブラリー検索パスの値を、実行可能ファイルまたは共用ライブラリー・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。

db2chglibpath コマンドは、オペレーティング・システム・ライブラリー検索パス環境変数の使用要件を置換するために使用できます。変数には、LIBPATH (AIX)、SHLIB_PATH (HPPA、HPIPF)、および LD_LIBRARY_PATH (AIX、SUN、HPPA64、HPIPF および Linux) などが含まれます。このコマンドは Linux および UNIX オペレーティング・システムでのみサポートされています。これは、DB2DIR/bin ディレクトリにあります。ここで、DB2DIR は DB2 データベースのインストール場所です。

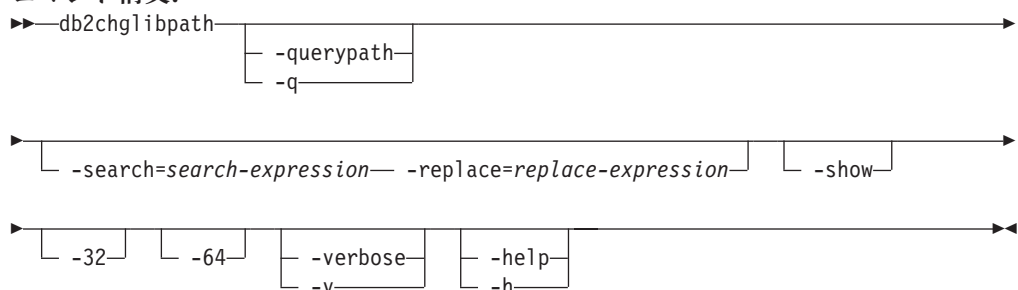
前提条件:

- 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。

db2chglbpath - 組み込みランタイム・ライブラリー検索パスの変更

-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/sqlllib/lib32 に変更するには、以下を発行します。

```
db2chglbpath -search=/usr/opt/db2_08_01/lib -replace=/u/usr1/sqlllib/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移行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移行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。

関連タスク:

- 「マイグレーション・ガイド」の『32 ビット・データベース・アプリケーションから 64 ビット・インスタンスへの移行』
- 「マイグレーション・ガイド」の『C、C++、および COBOL ルーチンの移行』
- 「マイグレーション・ガイド」の『組み込み SQL および CLI アプリケーションの移行』

db2chgpath - 組み込みランタイム・パスの変更

Linux および UNIX ベースのシステムで、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。

許可:

Root 権限。

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・パスの下すべてのファイルを調べ、ランタイム・パス更新を実行するには、以下のコマンドを発行します。

```
<DB2 installation path>/install/db2chgpath
```

- <DB2 installation path>/lib64 ディレクトリー下の libdb2.a と呼ばれる特定のファイルのパスを更新するには、以下を発行します。

```
<DB2 installation path>/install/db2chgpath -f lib64/libdb2.a
```

関連タスク:

- 「インストールおよび構成 補足」の『DB2 製品の手動インストール』
- 「インストールおよび構成 補足」の『ペイロード・ファイルの手動インストール (Linux および UNIX)』

db2ckbkp - バックアップの検査

このユーティリティを使用すると、バックアップ・イメージの保全性をテストして、イメージがリストア可能かどうかを判別することができます。また、バックアップ・ヘッダーに保管されているメタデータを表示するために使用することもできます。

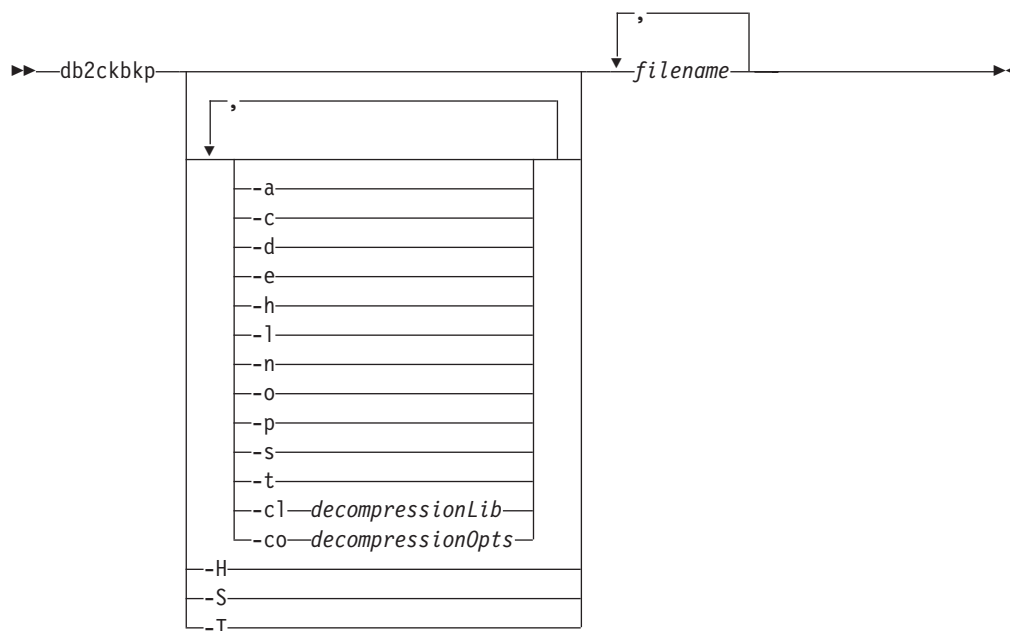
許可:

このユーティリティにはすべてのユーザーがアクセス可能ですが、イメージ・バックアップに対してこのユーティリティを実行するには、それらの読み取り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 decompresionLib

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 decompresionOpts

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、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 <tblspID> <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 <tblspID> <objType> <startPage> <numPages>
```

注:

1. <objType> が必要なのは、DMS ロード・コピー・イメージを検証する場合だけです。
2. <startPage> は、プール相対のオブジェクト・ページ番号です。

ログ・ファイルの場合、

```
L <log num> <startPos> <numPages>
```

その他のデータ (初期データなど) の場合、

```
O <objType> <startPos> <numBytes>
```

デフォルトの出力ファイルは *extractPage.out* です。デフォルトの出力ファイル名は、DB2EXTRACTFILE 環境変数に絶対パスを設定することによりオーバーライドできます。

- h メディア・ヘッダー情報を表示します。これには、リストア・ユーティリティーで要求されるイメージの名前およびパスも含まれます。
- H -h と同じ情報を表示します。ただし、イメージの先頭から 4K メディア・ヘッダー情報のみを読み取ります。イメージの妥当性検査は実行しません。このオプションは他のオプションと併用できません。
- 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)       --
```


db2ckbkp - バックアップの検査

```
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                 -- ISO8859-1
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。

関連資料:

- 15 ページの『db2adutl - TSM 内の DB2 オブジェクトの管理』

db2ckmig - データベース事前移行ツール

データベースが移行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:

```

➡ db2ckmig database -l filename -u userid -p password ➡

```

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 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保持するログ・ファイルを指定します。
- u システム管理者のユーザー ID を指定します。
- p 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移行するときに、**db2ckmig** が移行の一部として暗黙的に呼び出されます。 **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、インスタンスを移行する前に、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 ベースのシステムにおいてこのユーティリティは、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)。

Windows プラットフォームでは、インストール時に移行オプションを選択した場合、インスタンスの移行が行われ、**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

db2ckmig - データベース事前移行ツール

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、インスタンスを移行するために **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 ベースの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、移行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。

関連タスク:

- 「マイグレーション・ガイド」の『データベースが移行できる状態にあることの確認』

db2ckrst - 増分リストア・イメージ順序の検査

データベース履歴を照会して、増分リストアに必要なバックアップ・イメージのタイム・スタンプのリストを生成します。手操作の増分リストアに使用する、単純化されたリストア構文も生成されます。

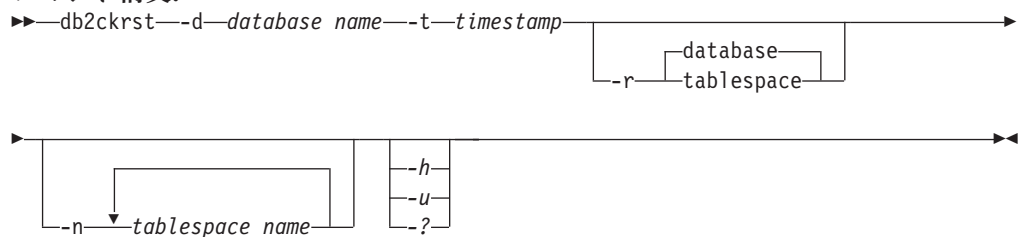
許可:

なし

必要な接続:

なし

コマンド構文:



コマンド・パラメーター:

-d *database name*

リストアされるデータベースの別名を指定します。

-t *timestamp*

増分をリストアするバックアップ・イメージのタイム・スタンプを指定します。

-r

実行するリストアのタイプを指定します。デフォルトはデータベースです。TABLESPACE を選択していながら表スペース名を指定しなかった場合、ユーティリティーは指定のイメージの履歴項目内を探索して、リストアを実行するためにリストされた表スペース名を使用します。

-n *tablespace name*

リストアされる 1 つ以上の表スペースの名前を指定します。データベース・リストア・タイプを選択して、表スペース名のリストを指定した場合、ユーティリティーは指定の表スペース名を使用して表スペース・リストアを続行します。

-h/-u/-?

ヘルプ情報を表示します。このオプションを指定すると、他のすべてのオプションは無視され、ヘルプ情報だけが表示されます。

例:

```

db2ckrst -d mr -t 20001015193455 -r database
db2ckrst -d mr -t 20001015193455 -r tablespace
db2ckrst -d mr -t 20001015193455 -r tablespace -n tbspl tbsp2

```

```
> db2 backup db mr
```

```
Backup successful. The timestamp for this backup image is : 20001016001426
```

db2ckrst - 増分リストア・イメージ順序の検査

```
> 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、**dbckrst** 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こともあります。 **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。

関連タスク:

- 「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『増分バックアップ・イメージからのリストア』

関連資料:

- 665 ページの『**PRUNE HISTORY/LOGFILE**』
- 738 ページの『**RESTORE DATABASE**』

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。

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、プロトタイプの実行、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機能は、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。

関連概念:

- ・ 「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」の『db2cli.ini 初期設定ファイル』

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 (DB2 Command Window)」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。

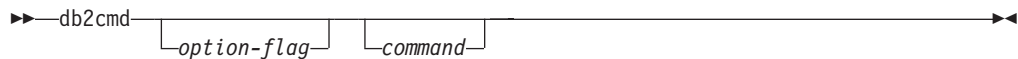
許可:

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 または **/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 または **/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 または **/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 または **/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。例えば、次のようなものがあります。

```
SHELL=C:¥COMMAND.COM C:¥ /P /E:32768
```

関連資料:

- 339 ページの『db2 - コマンド行プロセッサの呼び出し』

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。

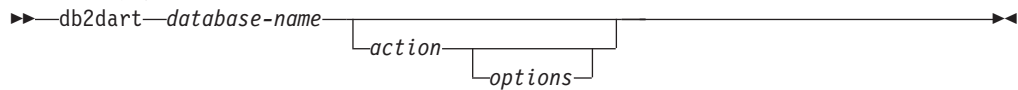
許可:

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:



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。
- /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ト・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。

/MT 表にドロップ・ペンディング状態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、表オブジェクト ID または表名のいずれか、およびパスワードの 3 つです。

/IP 表のデータ・ページを空として初期化します。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、表名または表オブジェクト ID、表スペース ID、開始ページ番号、ページ数、およびパスワードの 5 つです。

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。

/PW *password*

db2dart アクションの実行に必要なパスワード。有効なパスワードについては、DB2 サービスにお問い合わせください。

/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 および /IP 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、エクステンツ・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。

関連資料:

- 「管理ガイド: インプリメンテーション」の『rah および db2_all コマンドの説明』

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—◀◀

関連資料:

- 6 ページの『dasmigr - DB2 Administration Server の移行』
- 8 ページの『dasupdt - DAS の更新』
- 3 ページの『dasauto - DB2 Administration Server の自動始動』
- 4 ページの『dasprt - DB2 Administration Server の作成』
- 5 ページの『dasdrop - DB2 Administration Server の除去』

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。

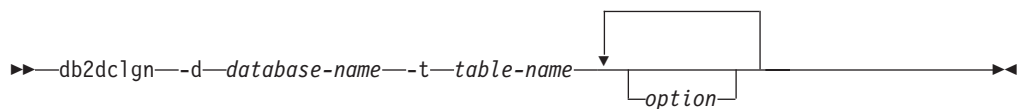
許可:

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

sqlbchar 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。

db2dclgn - 宣言生成プログラム

-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。

関連タスク:

- 「組み込み *SQL* アプリケーションの開発」の『db2dclgn 宣言生成プログラムを使用したホスト変数の宣言』

db2diag - db2diag.log 分析ツール

db2diag.log ファイルをフィルター操作して、フォーマット設定します。

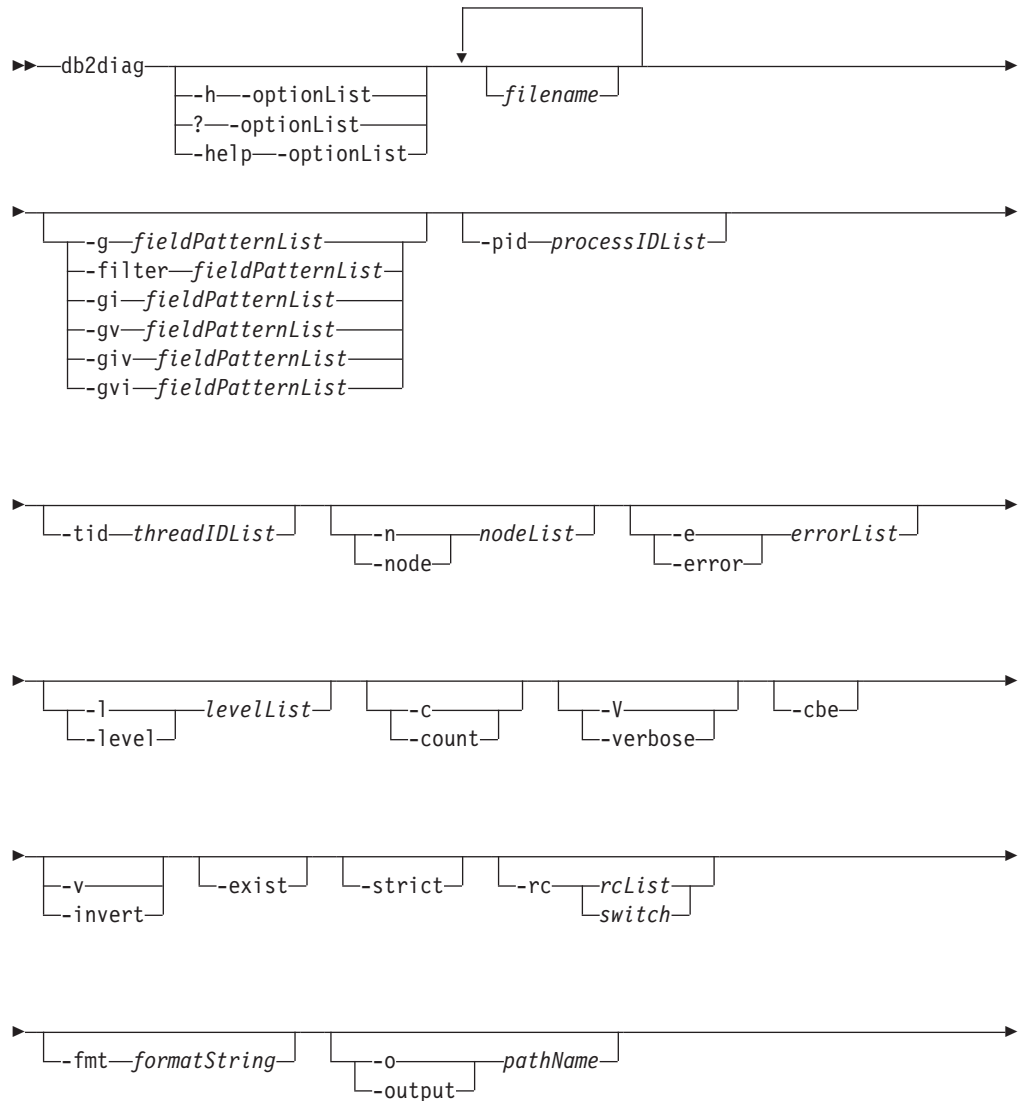
許可:

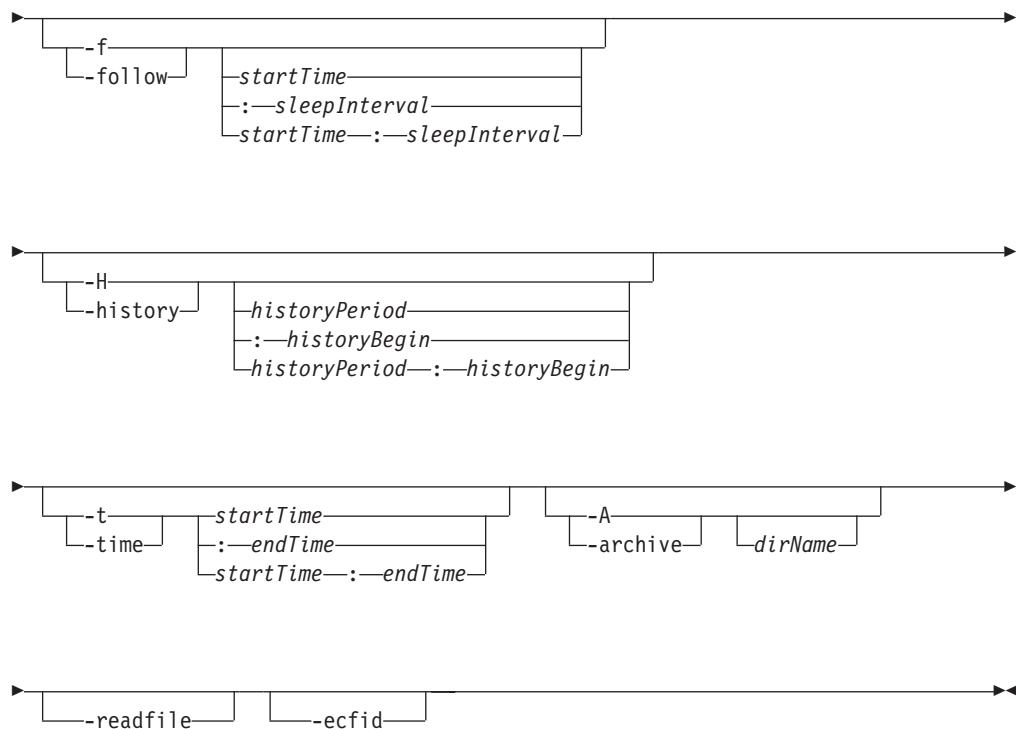
なし。

必要な接続:

なし。

コマンド構文:





コマンド・パラメーター:

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。

-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。

examples

ツールの使用に役立ついくつかの標準的な例を表示します。

tutorial

高度な機能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、以下に示すフィールドが使用できます (以下のリストは、接頭部 @ の場合にも有効です)。

%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

固有のレコード ID。

%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。

%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

プローブ番号。

%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。

%coordnode

コーディネーター・パーティション。

%coordindex

コーディネーター索引。

%apphdl

アプリケーション・ハンドル。%coordnode - %coordindex。

%message/%msg

エラー・メッセージ。

%calledprod

エラーを戻した関数の製品名。

%calledcomp

エラーを戻した関数のコンポーネント名。

%calledfunc

エラーを戻した関数の名前。

%called

エラーを戻した関数の詳細な説明。 %calledprod、%calledcomp、
%calledfunc。

%rcval

戻りコード値 (32 バイト)。

%rcdesc

エラーの説明。

%retcode/%rc

呼び出された関数によって戻された戻りコード。%rcval %rcdesc。

%errno

システム・エラー番号。

%errname

システム固有のエラー名。

%oserror

システム呼び出しによって戻されるオペレーティング・システム・エラー。
%errno %errname。

%callstack

呼び出しスタック。

%datadesc

データの説明。

%dataobject

データ・オブジェクト。

%data メッセージの詳細なデータ・セクション。
%datadesc %dataobject。

%argdesc

引数の説明。

%argobject

引数オブジェクト。

%arg エラーを戻した関数呼び出しの引数。
%argdesc %argobject。

%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。

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。

-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 @ 接頭部がこのオプションに使用されない場合は除きます。

-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。

-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*

示される重大度レベルのログ・メッセージだけを表示します。

-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。

-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。

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)。

nnnnnnn 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する、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 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:

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。

使用上の注意:

1. 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。
2. 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。
3. 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。

4. 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、入力是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。
5. `-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の **db2diag** の動作を上書きします。`-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。
6. `-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。
7. `-fmt` オプションは、`-strict`、`-cbe`、および `-verbose` オプションを上書きします。
8. `-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。
9. **db2diag** をスクリプトで使用するときは、`-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。
10. 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。

関連概念:

- 「問題判別ガイド」の『**db2diag** を使用した `db2diag.log` ファイルの分析』

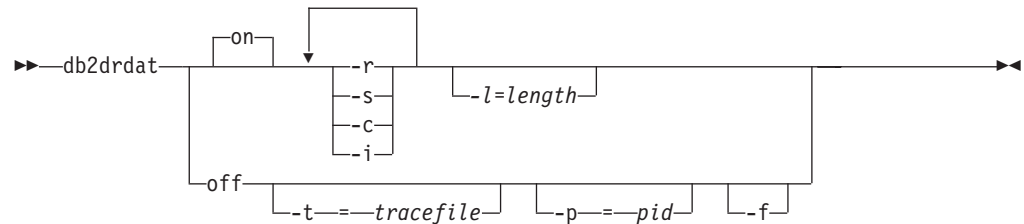
db2drdat - DRDA trace

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**

db2drdat - DRDA トレース

- 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。

関連資料:

- 570 ページの『LIST APPLICATIONS 』

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。

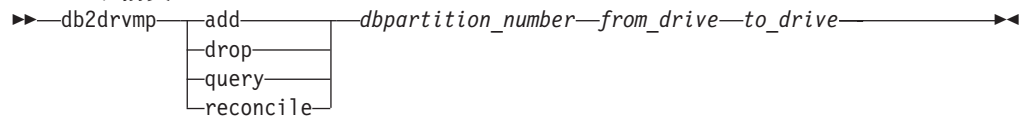
許可:

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:



コマンド・パラメーター:

add 新しいデータベース・ドライブ・マップを割り当てます。

drop 既存のデータベース・ドライブ・マップを削除します。

query データベース・マップを照会します。

reconcile

レジストリーの内容が損傷を受けたり、意図せずにドロップされたりした場合に、レジストリーにデータベース・ドライブ・マッピングを再適用します。

dbpartition_number

データベース・パーティション番号。このパラメーターは、追加およびドロップ操作に必要です。調整操作にこのパラメーターが指定されない場合、**db2drvmp** はすべてのデータベース・パーティションのマッピングを調整します。

from_drive

マップ元のドライブ名。このパラメーターは、追加およびドロップ操作に必要です。調整操作にこのパラメーターが指定されない場合、**db2drvmp** はすべてのドライブのマッピングを調整します。

to_drive

マップ先のドライブ名。このパラメーターは、追加操作に必要です。これは、他の操作には該当しません。

例:

NODE0 に関して、データベース・ドライブを F: から E: にマッピングするには、以下のコマンドを入力します。

```
db2drvmp add 0 F E
```

db2drvmp - DB2 データベースのドライブ・マップ

NODE1 に関して、データベース・ドライブを E: から F: にマッピングするには、以下のコマンドを入力します。

```
db2drvmp add 1 E F
```

使用上の注意:

1. データベース・ドライブ・マッピングは、表スペース、コンテナ、または他のデータベース・ストレージ・オブジェクトには適用されません。
2. データベース・ドライブ・マッピングに対する設定または変更は、即時には有効になりません。データベース・ドライブ・マッピングを有効にするには、Microsoft Cluster Administrator ツールを使用して、DB2 リソースをオフラインにしてから、オンラインにします。
3. 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。

関連タスク:

- 「マイグレーション・ガイド」の『Microsoft Cluster Server 環境での DB2 サーバーの移行』

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。

db2empfa を使用することで、レジストリー変数 **DB2_NO_MPFA_FOR_NEW_DB** が設定された後に作成されたデータベースに対して、複数ページ・ファイル割り振りを有効にできます。

関連概念:

- 「管理ガイド: プランニング」の『SMS 表スペース』

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。

関連資料:

- 「システム・モニター ガイドおよびリファレンス」の『データベース・システム・モニター・インターフェース』

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・ファイルと名前付きパイプをフォーマットし、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:

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。

データがファイルに書き込まれている場合、このツールは標準出力を使用した表示のためにファイルをフォーマットします。この場合、最初にモニターがオンになり、次にこのツールによってファイル中のイベント・データが表示されます。このツールを実行した後にファイルに書き込まれたデータをすべて表示させるには、**db2evmon** を再発行します。

データがパイプに書き込まれている場合、イベントが起きた時点で、このツールは標準出力を使用した表示のために出力をフォーマットします。この場合、ツールが開始した後 に、モニターがオンになります。

関連概念:

- 「システム・モニター ガイドおよびリファレンス」の『イベント・モニター自己記述型データ・ストリーム』

関連タスク:

- 「システム・モニター ガイドおよびリファレンス」の『コマンド行からのファイルまたはパイプ・イベント・モニター出力のフォーマット』

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。

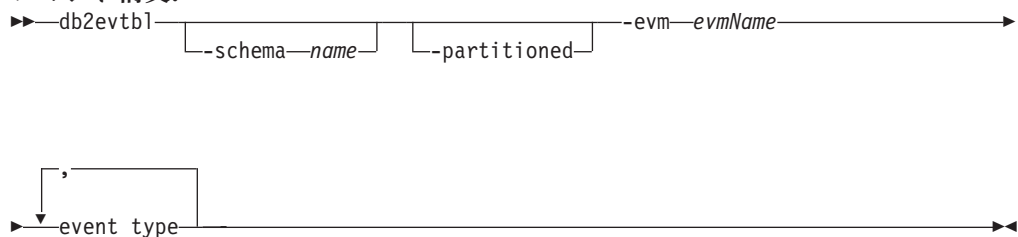
許可:

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

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 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えるイベント・タイプのいずれか。例えば DATABASE、TABLES、TRANSACTIONS などです。

例:

```
db2evtbl -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。

db2evtbl 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. **db2evtbl** 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・モニターを活性化します。

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。

関連概念:

- 「システム・モニター ガイドおよびリファレンス」の『イベント・モニター』

関連資料:

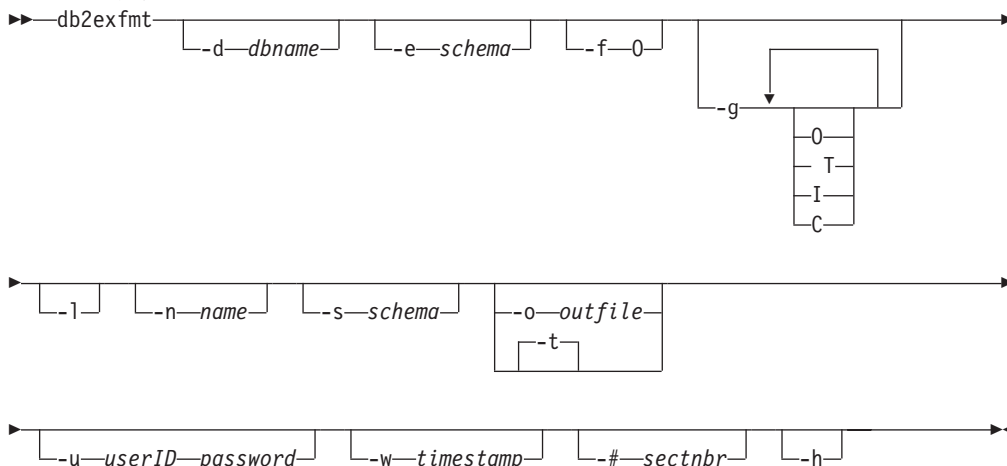
- 「SQL リファレンス 第 2 巻」の『CREATE EVENT MONITOR ステートメント』
- 「SQL リファレンス 第 2 巻」の『SET EVENT MONITOR STATE ステートメント』

db2exfmt - Explain 表フォーマット

db2exfmt ツールを使用して、Explain 表の内容をフォーマットします。このツールは、インスタンスの `sqllib` ディレクトリーの `misc` サブディレクトリーにあります。

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:

- 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 0 (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。 `-g` だけを指定した場合は、グラフが生成され、その後に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。
 - 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。
- 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。

-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 -1 オプションは除きます。

Explain 表 SQL スキーマを指定しない場合は、環境変数 **USER** の値がデフォルトとして使用されます。この変数が見つからない場合は、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。

関連概念:

- 「パフォーマンス」の『Explain ツール』
- 「パフォーマンス」の『Explain 情報のキャプチャーのガイドライン』
- 「パフォーマンス」の『Explain 情報の使用のガイドライン』

db2exmig - Explain 表の移行

Explain 表を移行します。 **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 表が移行されます。 Explain 表の移行ツールは、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:

以下のいずれか。

- *sysadm*
- *dbadm*

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*

移行される 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。

関連概念:

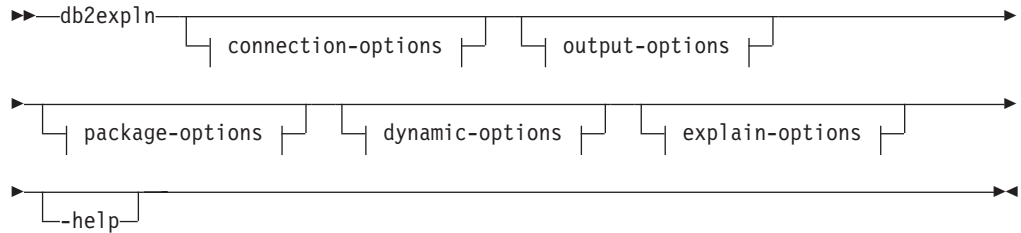
- 「パフォーマンス」の『Explain 表および Explain 情報の編成』

関連タスク:

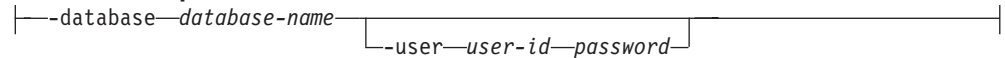
- 「マイグレーション・ガイド」の『Explain 表の移行』

db2expln - SQL および XQuery Explain

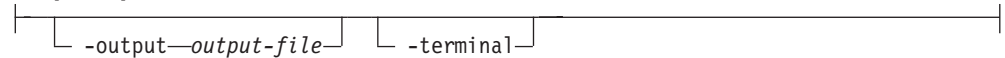
コマンド構文:



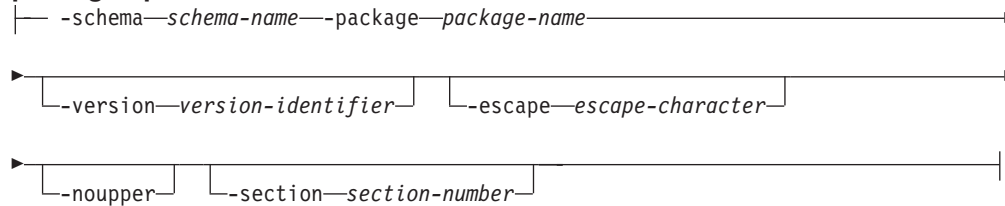
connection-options:



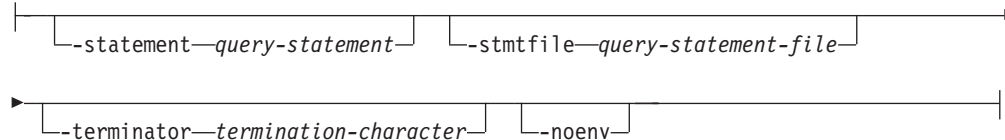
output-options:



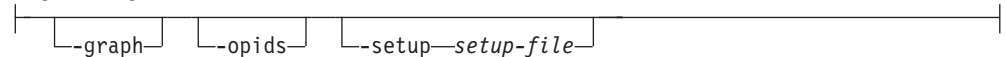
package-options:



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。

db2expln - SQL および XQuery Explain

-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*

動的ステートメントまたは再コンパイルする必要がある静的ステートメント(宣言されたグローバル一時表を参照する静的ステートメントなど)用に、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を含むファイル。ファイル内の各ステートメントが実行され、エラーまたは警告はすべて報告されます。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。

db2expln - SQL および XQuery Explain

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, SYSFUN, DEPT01, DEPT93@  
SELECT EMPNO, TITLE(JOBID) FROM EMPLOYEE@
```

これらのステートメントを Explain するには、以下のコマンドを入力します。

```
db2expln -database DEPTDATA -stmtfile statements.db2 -terminator @ -terminal
```

関連概念:

- 「パフォーマンス」の『db2expln および dynexpln 出力の説明』
- 「パフォーマンス」の『db2expln および dynexpln 出力の例』
- 「パフォーマンス」の『SQL および XQuery Explain ツール』

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。

許可:

- *sysadm*

必要な接続:

なし。

コマンド構文:

```

▶▶ db2extsec [ /u—usergroup ] [ /a—admingroup ] [ /r ]

```

コマンド・パラメーター:

/u 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。

/a 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。

/r 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。

関連概念:

- 「管理ガイド: インプリメンテーション」の『DB2ADMNS と DB2USERS グループの使用による拡張 Windows セキュリティー』

関連タスク:

- 「DB2 Universal Database サーバー機能 概説およびインストール」の『DB2ADMNS および DB2USERS ユーザー・グループへのユーザー ID の追加 (Windows)』

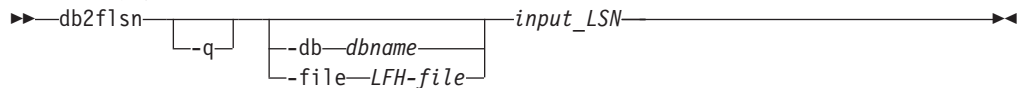
db2flsn - ログ・シーケンス番号の検出

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返します。

許可:

なし

コマンド構文:



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-name

ファイル名を含む LFH ファイル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\SQLLOGCTL.LFH 0000000000FA4368
Given LSN is contained in log page 6 in log file S0000002.LOG
```

使用上の注意:

- -db も -file も指定されない場合、ツールは LFH ファイルが現行ディレクトリ内の SQLLOGCTL.LFH であると想定します。
- 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、UDB 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、正しい結果 (*logfilsiz* の値が変更されない場合に得られる) につい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。

関連資料:

- 「管理 API リファレンス」の『DB2 log records』
- 「管理 API リファレンス」の『SQLU_LSN データ構造』

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。

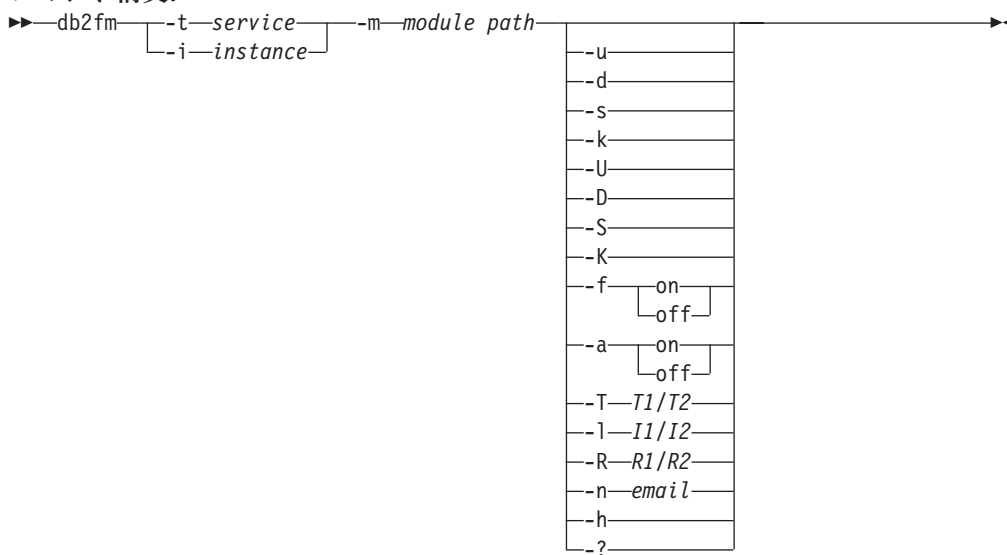
許可:

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/sqllib/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 サービスの状況を戻します。
- S 障害モニター・デーモンの状況を戻します。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*

障害モニターをオン/オフにします。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。このオプションがオフに設定されると、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。

例えば、次のようなものがあります。

 - -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 使用法を画面に表示します。
- ? 使用法を画面に表示します。

関連概念:

- 「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『Linux および UNIX 用の障害モニター機能』

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 `sqllib/bin` 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、db2fs.exe は `DB2PATH\bin` ディレクトリにあります。

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┐└┬-bbrowser┐▶▶

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 使用法情報を表示します。

-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

関連概念:

- 「DB2 Universal Database サーバー機能 概説およびインストール」の『ファースト・ステップ・インターフェース』

関連タスク:

- 「DB2 Universal Database サーバー機能 概説およびインストール」の『ファースト・ステップを使用した DB2 サーバーのインストールの検査 (Windows および Linux)』

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:

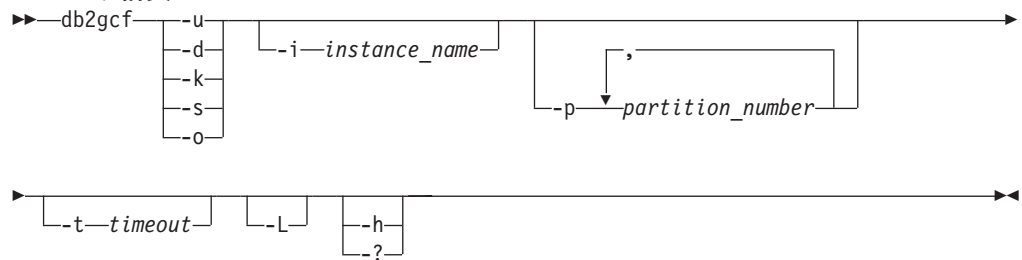
以下のいずれか。

- `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:

- 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を戻します。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を戻します。 `-t` パラメーターの値を指定すれば、すべてのデフォルトのタイムアウト値をオーバーライドできます。
- i *instance_name*
アクションの実行対象のインスタンス名。インスタンス名を指定しない場合

は、DB2INSTANCE の値が使用されます。インスタンス名を指定しない場合に、DB2INSTANCE が設定されていなければ、以下のエラーが戻されます。

```
db2gcf Error: Neither DB2INSTANCE is set nor instance passed.
```

-p *partition_number*

パーティション・データベース環境で、アクションの実行対象のデータベース・パーティション番号を指定します。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。

-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:

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* の状況を戻します。

```
db2gcf -s -p 0 -i stevera
```

以下の出力が戻されます。

```
Instance : stevera
DB2 State
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ることがないようにします。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。

関連概念:

- 「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『高可用性』

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。

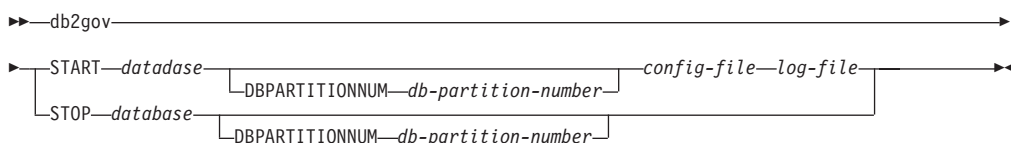
許可:

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・ユーティリティーは、データベース・パーティション構成ファイル `db2nodes.cfg` を読み取ることによって、すべてのデータベース・パーティション上の管理プログラムを停止します。

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` に置き換えられます。

関連概念:

- 「パフォーマンス」の『ガバナー・ユーティリティー』

関連タスク:

- 「パフォーマンス」の『ガバナーの構成』
- 「パフォーマンス」の『管理プログラムの開始と停止』

関連資料:

- 122 ページの『db2govlg - DB2 管理プログラム・ログ照会』

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` に置き換えられます。

関連資料:

- 120 ページの『db2gov - DB2 管理プログラム』

db2gpmmap - 分散マップの取得

データベースが既にセットアップされていて、そのデータベースのデータベース・パーティション・グループが定義されている場合、**db2gpmmap** は、カタログ・パーティション・データベース・サーバーから、データベース表またはデータベース・パーティション・グループの分散マップを取得します。

許可:

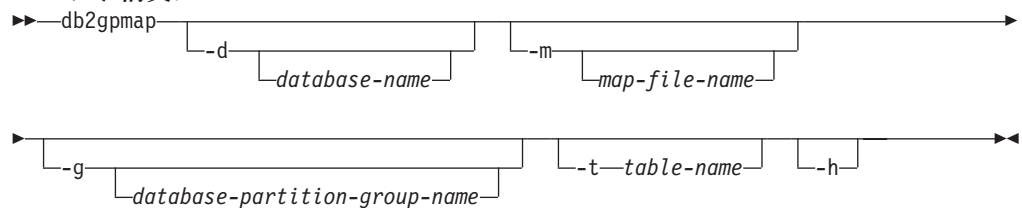
以下の両方が必要です。

- ・ システム・カタログ表に対する読み取りアクセス
- ・ db2gpmmap.bnd に対するパッケージの BIND 特権と EXECUTE 特権

必要な接続:

db2gpmmap を使用する前に、データベース・マネージャを開始し、db2gpmmap.bnd をデータベースにバインドしておく必要があります。バインドしていない場合、**db2gpmmap** はそのファイルをバインドしようとします。

コマンド構文:



コマンド・パラメーター:

- d** 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** 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** 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** 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
```

関連概念:

- ・ 「管理ガイド: プランニング」の『分散マップ』

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・インディケーター上のアラートに関する詳細や推奨事項を表示し、アラートを解決するためにその推奨処置を取ることができます。

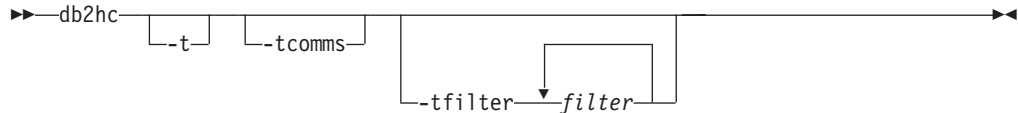
許可:

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。

関連概念:

- 「システム・モニター ガイドおよびリファレンス」の『ヘルス・モニターのグラフィック・ツール』

関連タスク:

- 「システム・モニター ガイドおよびリファレンス」の『ヘルス・センターを使用したヘルス・インディケーターの構成』

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 ベースの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。

関連タスク:

- 「管理ガイド: インプリメンテーション」の『インスタンスの自動開始』

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、移行、移行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。

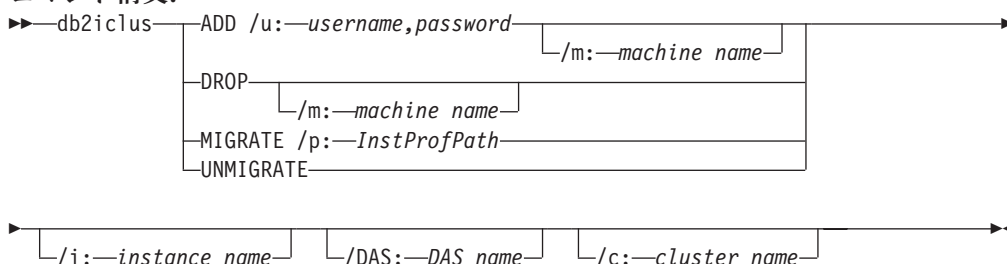
許可:

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 インスタンスに移行します。

UNMIGRATE

MSCS 移行を取り消します。

/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 インスタンスに移行する際に必要です。

/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

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 の所有である必要があります。

7. WA26 から、**db2iclus** コマンドを使用して、他のマシンを DB2 クラスター・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. クラスター管理から、DB2 Group に所属する、タイプ「DB2」の新しいリソース・タイプを作成します。このリソースの名前は、インスタンス名とまったく同一でなければならないので、ここでは DB2 という名前になります。クラスター管理が DB2 リソースに関連する従属関係のプロンプトを出すので、DB2 リソースがディスク O、ディスク P、および mscs5 に依存していることを確認します。
12. 必要なら、クラスター管理を介して、また DB2_FALLBACK プロファイル変数を使用して、フォールバック用に DB2 Group を構成します。
13. すべてのデータをディスク O およびディスク P に入れ、すべてのデータベースを作成するか、またはリストアします。
14. フェイルオーバー構成をテストします。

使用上の注意:

MSCS フェイルオーバー環境で実行するためにインスタンスを移動するには、まず現行マシンでインスタンスを移行してから、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移行を取り消してください。

関連資料:

- 129 ページの『db2icrt - インスタンスの作成』
- 132 ページの『db2idrop - インスタンスの除去』
- 135 ページの『db2imigr - インスタンスの移行』
- 293 ページの『db2stop - DB2 の停止』

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 ベースのシステムにおいて、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インストール・ロケーション)。
Windows オペレーティング・システムの場合、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)。

db2icrt コマンドは、**db2icrt** を発行する場所から、DB2 コピー・インストール・パスに関連した DB2 インスタンスを作成します。

許可:

Linux および UNIX ベースのシステムでは 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 ベースのシステムの場合

```

▶▶ db2icrt [-h] [-d] [-a AuthType] [-p PortName]
           [-s InstType] [-u FencedID] InstName

```

Windows オペレーティング・システムの場合

```

▶▶ db2icrt InstName [-s InstType] [-u UserName, Password]
                   [-p InstProfPath] [-h HostName] [-r PortRange] [-?]

```

コマンド・パラメーター:

Linux および UNIX ベースのシステムの場合

-h または -?

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。

db2icrt - インスタンスの作成

-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、システムの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。 DB2 Connect Personal Edition を使用している場合は、この値を使用してください。
- ese** ローカルおよびリモート・クライアントでデータベース・サーバーのインスタンスを作成するために使用します。
- wse** DB2 Workgroup Server Edition、DB2 Express Edition、および DB2 Connect Enterprise Edition のインスタンスを作成するために使用されます。

-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。

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。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。 DB2 Connect Personal Edition を使用している場合は、この値を使用してください。
- standalone** ローカル・クライアントでデータベース・サーバーのインスタンスを作成するために使用します。
- ese** 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。
- wse** DB2 Workgroup Server Edition、DB2 Express Edition、および DB2 Connect Enterprise Edition のインスタンスを作成するために使用されます。

-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
```

-? 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:

- **-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、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 **db2iclus** 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移行します。
- 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。既に関連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。

関連概念:

- 「管理ガイド: インプリメンテーション」の『ユーザー、ユーザー ID、およびグループの命名規則』

関連資料:

- 126 ページの『db2iclus - Microsoft Cluster Server』

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 ベースのシステムにおいて、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インストール・ロケーション)。Windows オペレーティング・システムの場合、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)。

許可:

Linux および UNIX ベースのシステムでは 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でアクセスします。

コマンド構文:

Linux および UNIX ベースのシステムの場合

```
▶▶ 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 ベースのシステムの場合

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** または **-?** 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 ベースのシステム上で、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。

関連資料:

- 129 ページの『db2icrt - インスタンスの作成』

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 ベースのシステムにおいて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・システムにおいて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 

コマンド・パラメーター:

-h 使用法情報を表示します。

関連タスク:

- 「管理ガイド: インプリメンテーション」の『インスタンスの除去』
- 「管理ガイド: インプリメンテーション」の『構成アシスタントおよびコントロール・センターを使用するための DB2 Administration Server (DAS) の設定』
- 「管理ガイド: インプリメンテーション」の『UNIX でのインスタンス構成の更新』

関連資料:

- 143 ページの『db2iupdt - インスタンスの更新』

db2imigr - インスタンスの移行

Linux および UNIX ベースのシステムにおいて、**db2imigr** コマンドを実行している場所から、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移行がサポートされているもの) から DB2 コピーの現行バージョンに移行します。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・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)。

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移行します。**db2imigr** コマンドは、インスタンスの移行先の 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、移行後にインスタンス・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)。

インスタンスの移行前に、データベースの移行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、移行は続行されません。

許可:

Linux および UNIX ベースのシステムでは 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 ベースのシステムの場合

```
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 ベースのシステムの場合

-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。DB2 クライアントだけをインストールする場合には、このオプションは任意指定です。

db2imigr - インスタンスの移行

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・インスタンスを移行する場合には、このオプションは必須です。

/p:instance-profile-path

移行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 authType (認証タイプ) として SERVER、CLIENT、または SERVER_ENCRYPT を指定します。

/? **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 ベースのシステムの場合

- **db2imigr** コマンドは、移行前のバージョンの /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移行する場合、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・ロケーションに移行されません。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移行は、インスタンスの移行後に実行されます。

関連概念:

- 「マイグレーション・ガイド」の『DB2 サーバーの移行の概要』

関連タスク:

- 「マイグレーション・ガイド」の『インスタンスの移行』

関連資料:

- 6 ページの『dasmigr - DB2 Administration Server の移行』
- 63 ページの『db2ckmig - データベース事前移行ツール』

db2init <database> as mirror コマンドを発行する前に **db2 connect to <database>** コマンドを発行しないでください。初期化する前に分割ミラー・データベースに接続すると、ロールフォワード・リカバリーに必要なログ・ファイルが消去されてしまいます。その接続によって、データベースは、以前にそのデータベースを中断した時点の状態に戻ります。中断の時点でデータベースが整合とマーク付けされると、DB2 データベース・システムはクラッシュ・リカバリーの必要はないと判断して、将来の利用のためにログを空にします。ログが空になった場合、ロールフォワードしようとする、SQL4970N エラー・メッセージが戻されます。

パーティション・データベース環境では、すべてのデータベース・パーティションに対して **db2inidb** 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。

関連タスク:

- 「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『スプリット・ミラーを使用したデータベースのクローン作成』

db2inidb - ミラーリングされたデータベースの初期化

- 「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『スプリット・ミラーをバックアップ・イメージとして使用する』
- 「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『スプリット・ミラーをスタンバイ・データベースとして使用する』

関連資料:

- 255 ページの『db2relocatedb - データベースの再配置』
- 「管理ガイド: インプリメンテーション」の『rah および db2_all コマンドの説明』

db2inspf - 検査結果のフォーマット

このユーティリティは、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を使用して、検査の詳細を表示します。db2inspf ユーティリティによるフォーマットには、表のみのフォーマット、表スペースのみのフォーマット、エラーのみのフォーマット、警告のみのフォーマット、またはサマリーのみのフォーマットがあります。

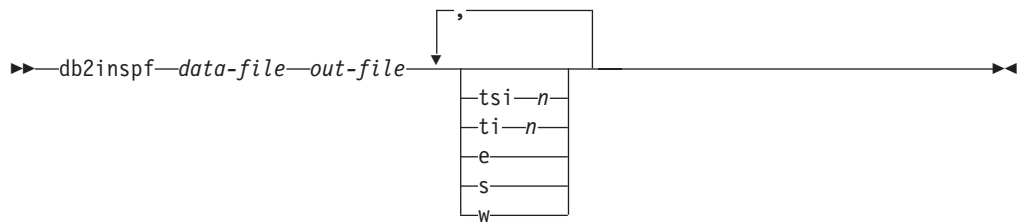
許可:

このユーティリティにはすべてのユーザーがアクセス可能ですが、結果ファイルに対してこのユーティリティを実行するには、それらの読み取り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 *n* 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 *n* 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。

関連資料:

- 562 ページの『INSPECT 』
- 「管理 API リファレンス」の『db2Inspect API - データベースの構造上の保全性の検査』

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:

コマンドが発行されるシステムでの root 権限。

必要な接続:

なし。

コマンド構文:

```

db2isetaup [-i language-code] [-l logfile] [-t tracefile]
           [-r response_file] [-?] [-h]

```

コマンド・パラメーター:

-i language-code

インストールを実行する際の希望する言語の 2 文字から成るコード。指定されないと、このパラメーターは現行ユーザーのロケールにデフォルト設定されます。

-l logfile

ログ・ファイルの絶対パスおよび名前。名前を指定しない場合、パスとファイル名はデフォルトの /tmp/db2isetaup.log になります。

-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・セッアップ・ウィザードは、DB2 セッアップ・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・セッアップ・タスクの実行、およびインスタンスのセッアップを行うことができます。DB2 インスタンス・セッアップ・ウィザードは、インスタンス・セッ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 ディレクトリーにあります。db2icrt および db2iupdt などの他のインスタンス・スクリプトも一緒にあります。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。他のインスタンス・

db2iseta - インスタンス作成インターフェースの開始

スクリプトと同様、この実行可能ファイルでは root 権限が必要です。これは標準インストールでは使用できますが、コンパクト・インストールでは使用できません。

3. db2iseta は、サポートされているすべての Linux および UNIX ベースのシステムで実行できます。

関連概念:

- 「*DB2 Universal Database* サーバー機能 概説およびインストール」の『DB2 のインストール方式』

関連資料:

- 143 ページの『db2iupdt - インスタンスの更新』
- 268 ページの『db2seta - DB2 のインストール』
- 129 ページの『db2icrt - インスタンスの作成』

db2iupdt - インスタンスの更新

Linux および UNIX ベースのシステムにおいてこのコマンドは、指定された DB2 インスタンスを更新します。 **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ックスパックを、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。
- 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。

Linux および UNIX ベースのシステムにおいてこれは、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のインストール先)。

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。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¥sqlllib¥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 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 オペレーティング・システムの場合

```

>> db2iupdt [-h] [-d] [-k] [-D] [-s] [-a AuthType]
              [-?]
              [-u FencedID] [-e InstName]

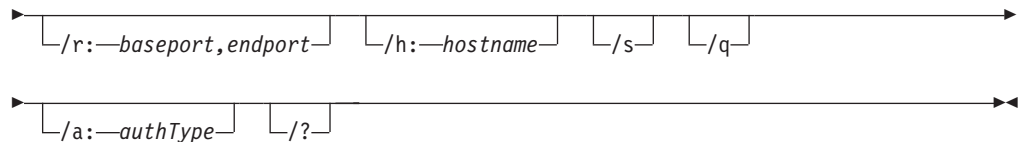
```

Windows オペレーティング・システムの場合

```

>> db2iupdt InstName /u:—username,password
                               /p:—instance-profile-path

```



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 または -?

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・インスタンスからサーバー・インスタンスに変換する場合だけです。インスタンスが既にサーバー・インスタンスである場合、またはインスタンスがクライアント・インスタンスであり、引き続きクライアント・インスタンスのままである場合 (-k オプションの使用による)、-u オプションは不要です。-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 authType (認証タイプ) として SERVER、CLIENT、または SERVER_ENCRYPT を指定します。

/? **db2iupdt** コマンドの使用方法を表示します。

例 (UNIX):

1. インストール・パス DB2DIR に関連したインスタンス db2inst1 があり、そのインストール・パスにフィックスパックを上書き適用した場合には、以下のコマンドを実行することによってそのインスタンスを更新することが必要な場合があります。

```
<DB2DIR>/instance/db2iupdt db2inst1
```

これにより、対象のインスタンスのタイプは、インスタンスとして最高のものになります。インスタンスを元のタイプのままにするには、-k オプションを使用することが必要な場合があります。

2. 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 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・プロファイル変数の更新は、インスタンスの更新後に実行されます。

関連タスク:

- 「管理ガイド: インプリメンテーション」の『Windows でのインスタンス構成の更新』

db2iupdt - インスタンスの更新

- 「管理ガイド: インプリメンテーション」の『UNIX でのインスタンス構成の更新』

関連資料:

- 134 ページの『db2ilist - インスタンスのリスト』

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下位レベル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:

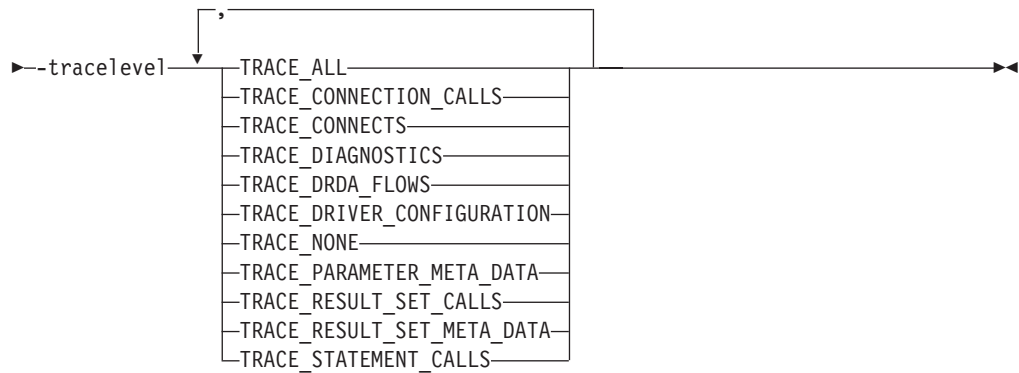
このコマンドは、データベース接続を確立します。

コマンド構文:

```

▶ db2jdbcbind [-help] [-url jdbc:db2://server:port/dbname] [-user username]
               [-password password] [-collection collection ID]
               [-size number of packages]

```

**コマンド・パラメーター:**

-help ヘルプ情報を表示します。その他のすべてのオプションは無視されます。

-url jdbc:db2://server:port/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 user's Distinguished Name -w password ➡
    |_____|
    -r
```

コマンド・パラメーター:

-u user's Distinguished Name

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。

関連タスク:

- 「管理ガイド: インプリメンテーション」の『DB2 アプリケーション用 LDAP ユーザーの構成』

db2ldcfg - LDAP 環境の構成

関連資料:

- 415 ページの『CATALOG LDAP DATABASE 』

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。

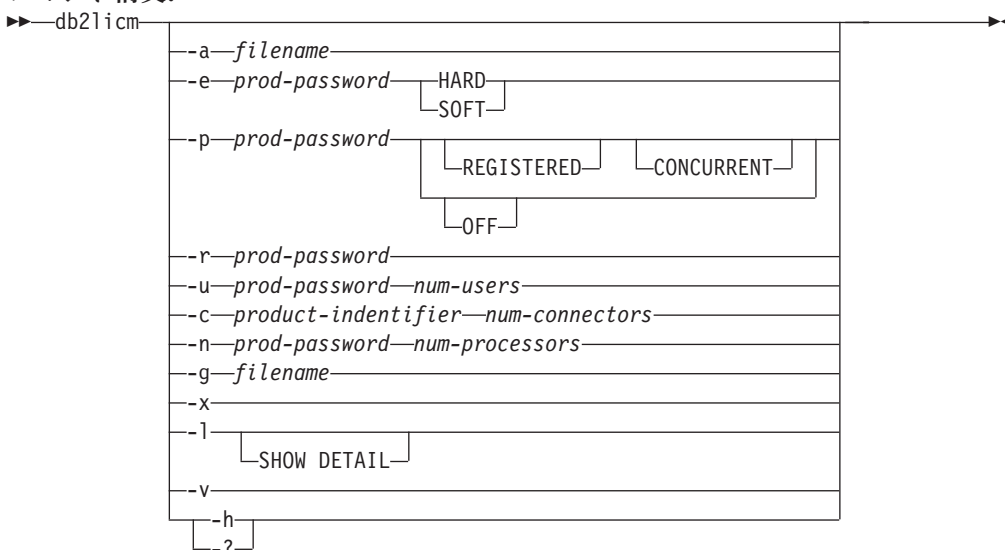
許可:

sysadm (ライセンス除去の場合のみ)。

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、HARD および SOFT です。HARD は、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 は、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 keyword

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。すべてのポリシーをオフにするには OFF を指定します。

-r product-identifier

製品のライセンスを除去します。ライセンスが除去された後は、製品は「試用版」モードで機能します。特定の製品のパスワードを得るには、-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。

-n product-identifier num-processors

ライセンスのあるプロセッサの数を更新します。製品 ID と、この製品での使用をライセンス許可されたプロセッサの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。ライセンス機能に関する詳細情報 (存在する場合) を表示するには、SHOW DETAIL を指定します。

-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 db2wse registered concurrent
db2licm -r db2ese
db2licm -u db2wse 10
db2licm -n db2ese 8
```

関連タスク:

- 「インストールおよび構成 補足」の『db2licm コマンドによる DB2 製品またはフィーチャー・ライセンス・キーの登録』

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。これは 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。

関連タスク:

- 「管理ガイド: インプリメンテーション」の『直接ディスク・アクセス・デバイスのアタッチ』

db2logsforffwd - ロールフォワード・リカバリーに必要なログのリスト

DB2TSCHG.HIS ファイルを解析します。このユーティリティを使用すると、ユーザーは、表スペース・ロールフォワード操作に必要なログ・ファイルを見つけることができます。このユーティリティは `sqllib/bin` にあります。

許可:

必要な接続:

なし。

コマンド構文:

▶▶ `db2logsforffwd` *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/ofef/ofef/NODE0000/S00000001/DB2TSCHG.HIS
```

```
db2logsForRfwd DB2TSCHG.HIS -all
```

関連概念:

- 「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『表スペースでの変更のロールフォワード』

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は一致する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す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、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:

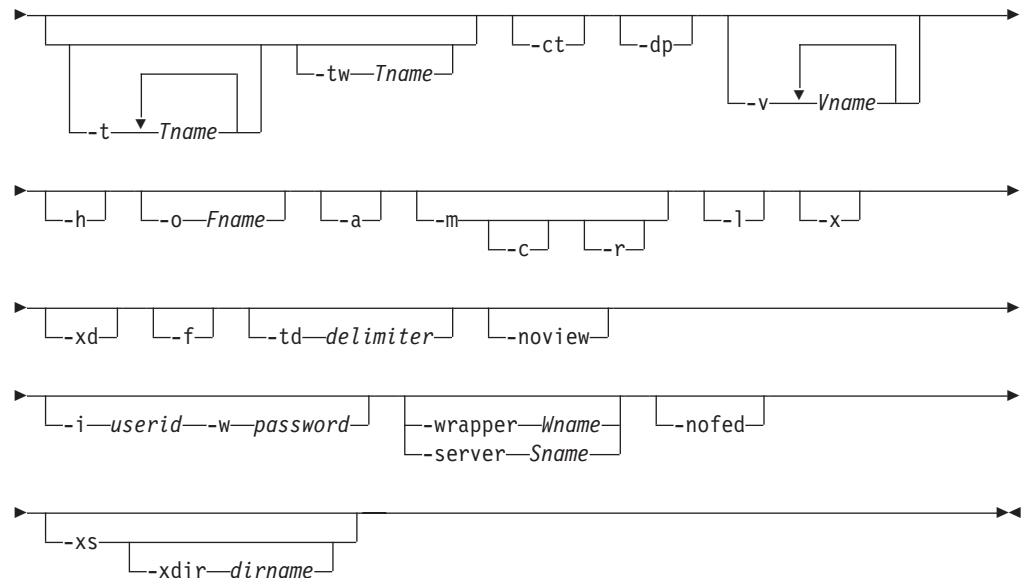
なし

コマンド構文:

```

▶▶ db2look -d DBname [-e] [-u Creator] [-z schema]

```



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すが、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。

- 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、参照保全、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
- 索引の仕様
- ストアード・プロシージャ

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 文節など) が参照するユーザー・ソース・コードが使用できる状態であればなりません。

-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は、`¥" MyTabLe ¥"` のように、円記号と二重引用符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サ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 9.1 のパーティション表がサポートされません。

-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** 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。

-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。

-o *Fname*

LaTeX 形式を使用する場合、*filename.tex* に出力を書き込みます。プレーン・テキスト形式を使用する場合、*filename.txt* に出力を書き込みます。そうでない場合は、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。
- 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、およびバッファー・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ー・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
- 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** 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。

db2look コマンドは、以下の構成パラメーター用の更新コマンドを生成します。

- データベース・マネージャー構成パラメーター
 - cpuspeed
 - intra_parallel
 - comm_bandwidth
 - nodetype
 - federated
 - fed_noauth
- データベース構成パラメーター
 - locklist
 - dft_degree
 - maxlocks
 - avg_appls
 - stmtheap
 - dft_queryopt

db2look コマンドは、以下の DB2 レジストリー変数用の **db2set** コマンドを生成します。

- DB2_PRED_FACTORIZE
- DB2_CORRELATED_PREDICATES
- DB2_LIKE_VARCHAR
- DB2_SORT_AFTER_TQ
- DB2_ORDERED_NLJN
- DB2_NEW_CORR_SQ_FF
- DB2_PART_INNER_JOIN
- DB2_INTERESTING_KEYS

-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。

-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。

-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、それらを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・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・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/
```

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。

- 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。

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複数の語で成る表名は、円記号および二重引用符の両方で囲む必要があります。例えば、

```
¥"My Table¥"
```

- `-tw` オプションは、`-x` オプション (GRANT 特権を生成する場合)、`-m` オプション (表統計および列統計を戻す場合)、および `-l` オプション (ユーザー定義表スペース、データベース・パーティション・グループ、およびバッファー・プールの DDL を生成する場合) とともに使用できます。 `-t` オプションを `-tw` オプションとともに指定すると、`-t` オプション (およびそれに関連した *Tname* 引数) が無視されます。
- フェデレーテッド・データ・ソース上、または DB2 Universal Database for z/OS and OS/390、DB2 Universal Database for iSeries、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・パーティション機能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。

関連概念:

- 「パフォーマンス」の『実動データベースのモデル化の統計』

関連資料:

- 「SQL リファレンス 第 1 巻」の『LIKE 述部』

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

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 文書および DB2 バージョン 8 製品が含まれます。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。

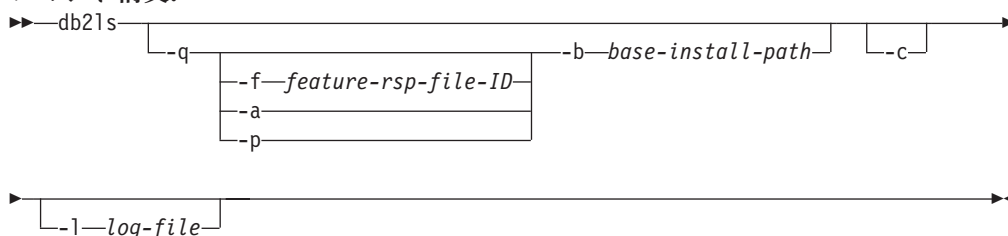
許可:

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

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

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:

- /usr/local/bin ディレクトリで **db2ls** コマンドへのシンボリック・リンクを使用するには、少なくとも 1 つの DB2 バージョン 9 製品が既にインストールされている必要があります。
- **db2ls** コマンドは、DB2 製品を照会するための唯一の方法です。Linux または UNIX オペレーティング・システム固有のユーティリティー (pkgadd、rpm、SMIT、または swinstall など) を使用して DB2 製品を照会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。
- 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。

関連資料:

- 「管理 SQL ルーチンおよびビュー」の『ENV_PROD_INFO 管理ビュー - インストール済みの DB2 製品に関する情報の検索』
- 10 ページの『db2_deinstall - DB2 製品またはフィーチャーのアンインストール』

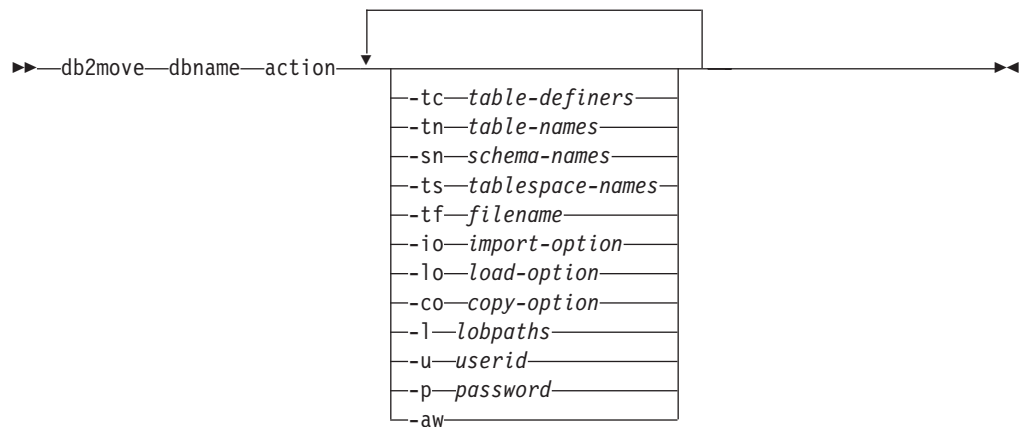
db2move - データベース移動ツール

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 表定義者です。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`」表名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 表名。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。指定されると、指定されたストリング中の名前と完全に一致する名前を持つ表のみがエクスポートまたはコピーされます。指定されない場合、デフォルトではすべてのユーザー表を使用します。複数の表名を指定する場合、それぞれをコンマで区切る必要があります。表名間にブランクを入れることはできません。COPY アクションを使用する場合、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。EXPORT アクションを使用する場合、表名は、非修飾でリストします。このオプションを「`-tc`」表定義者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。

db2move は、名前が指定された表名と一致し、かつ定義者が指定された表定義者と一致する表に対してのみ実行されます。

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。

-sn スキーマ名。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、完全に一致するスキーマ名の表だけがエクスポートまたはコピーされます。複数のスキーマ名を指定する場合は、それぞれの名前をコンマで区切る必要があります。複数のスキーマ名の間にブランクを使用することはできません。8 文字より短いスキーマ名は、8 文字の長さになるまで埋め込まれます。

エクスポートの場合は、以下ようになります。

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 文節の 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 表スペース名。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** 文節の **LIKE** 述部にパーセント付きの表名が使用されます。 **-ts** オプションが指定されない場合、デフォルトではすべての表スペースを使用します。複数の表スペース名を指定する場合は、それぞれの名前をコンマで区切る必要があります。複数の表スペース名の間にブランクを使用することはできません。 8 文字より短い表スペース名は、8 文字の長さになるまで埋め込まれます。例えば、'mytb' のような表スペース名の場合、アスタリスクを使用するときは、"-sn my*b" ではなく "-ts my*b*" のような指定が必要になります。

-tf filename

これは **EXPORT** または **COPY** アクションのみです。指定されると、特定のファイルにリストされている表だけがエクスポートまたはコピーされます。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。以下は、ファイルの内容の例です。

```
"SCHEMA1"."TABLE NAME1"
"SCHEMA NAME77"."TABLE155"
```

-io インポート・オプション。デフォルトは **REPLACE_CREATE** です。

有効なオプションは、**INSERT**、**INSERT_UPDATE**、**REPLACE**、**CREATE**、および **REPLACE_CREATE** です。

-lo ロード・オプション。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** 文節はオプションです。 **USER** が **userid** を指定する場合は、**USING** 文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。

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す。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。このオプションは、Q1/2006 配信の包含可能性により保留されていますが、このパラメーターは最初の設計に入ります。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になります。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 LOB パス。IMPORT および EXPORT の場合、このオプションが指定されると、これは XML パスにも使用されます。デフォルトは、現行ディレクトリです。

このオプションは、LOB または XML ファイルが (EXPORT の一部として) 作成されるか、または (IMPORT または LOAD の一部として) 検索される絶対パス名を指定します。複数のパスを指定する場合、それぞれをコマンドで区切る必要があります。パス間にブランク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 ユーザー ID。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 パスワード。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、**tbname1** という名前、または **%tbname2** のような表名を持つすべての表をエクスポートするには、以下を発行します。

```
db2move sample export -tc userid1,us*rid2 -tn tbname1,*tbname2
```

- **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 ベースの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:

- XML 列を含む表へのデータのロードはサポートされていません。この問題は、**IMPORT** または **EXPORT** コマンドを手動で発行するか、または **db2move -Export** および **db2move -Import** 動作を使用することにより解決できます。それらの表に、常に生成される ID 列も含まれる場合、表にデータをインポートすることはできません。
- このツールはユーザーが作成した表をエクスポート、インポート、またはロードします。データベースが、あるオペレーティング・システムから別のオペレーティング・システムに複製される場合、**db2move** によって表の移動が容易になります。表と関連する他のすべてのオブジェクト、例えば別名、ビュー、トリガー、ユーザー定義機能なども移動する必要があります。 **REPLACE_CREATE** オプションを指定したインポート・ユーティリティーを使って、ターゲット・デー

データベース上で表を作成する場合、『インポートを使用した、エクスポートされる表の再作成』に概略されている制限を受けます。 **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。

- 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。 **load** API が **db2move** によって呼び出されると、**NONRECOVERABLE** オプションが使用されます。 **logretain** がオンで、かつ **-lo** オプションが **INSERT** である場合、ロード操作によりその表はアクセス不能としてマークが付けられ、ドロップすることが必要になります。ロードされた表が常駐する表スペースはバックアップ・ペンディング状態にされ、アクセスできません。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、全データベースのバックアップまたは表スペースのバックアップが必要です。
IMPORT アクションにおける **DB2MOVE** のパフォーマンスは、デフォルトのバッファ・プール **IBMDEFAULTBP** を調整し、構成パラメーター **sortheap**、**util_heap_sz**、**logfilsize**、および **logprimary** を更新することによって改善できます。
- **NONRECOVERABLE** のリカバリーできるオプションに関して詳しくは、『データ移動ユーティリティ・ガイドおよびリファレンス』を参照してください。

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・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 アクションからの出力ファイル。

COPYSCHEMA.err
COPY アクションからの出力ファイル。

LOADTABLE.err
COPY アクションからの出力ファイル。

LOADTABLE.msg
COPY アクションからの出力ファイル。

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。

関連資料:

- 156 ページの『db2look - DB2 統計および DDL 抽出ツール』

db2mqdsn - 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。

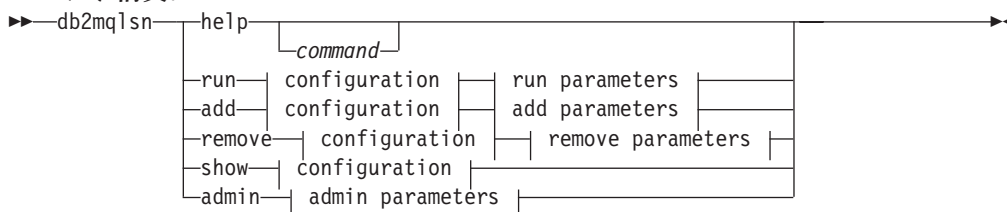
Windows オペレーティング・システムでは、このユーティリティは、DB2PATH\sql\lib\bin ディレクトリーにあります。DB2PATH は、DB2 データベース製品の現行バージョンがインストールされている場所です。

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:

- **db2mqdsn 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 mqlConfi に対する EXECUTE 特権を持っている必要があります。
- **db2mqdsn run** オプションと **db2mqdsn 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。
- **db2mqdsn 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 **db2mqdsn 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 mqlRun 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:

```

|--configDB--configuration database name----->
|
|└--configUser--user ID--configPwd--password--┘
|
|└--config--configuration name--┘

```

run parameters:

```

|└--adminQueue--admin queue name--┘
|└--adminQMgr--admin queue manager--┘

```

add parameters:

```

|--inputQueue--input queue name--┘
|└--queueManager--queue manager name--┘
|
|└--procSchema--stored procedure schema--procName--stored procedure name--┘
|
|└--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--┘└--adminQMgr--admin queue manager--┘
|└--adminQueueList--namelist of admin queue names--┘
|
|└--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

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 (シャットダウンや再始動など) 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、シャットダウンか再始動のいずれかです。シャットダウンの場合は、実行中の MQListener が現在のメッセージの処理を終えた時点で MQListener を終了します。再始動の場合は、シャットダウンを実行してから、再び構成を読み込んで再始動します。

例:

```
db2mq1sn show -configDB sampleDB -config nightlies
```

```
db2mq1sn add -configDB sampleDB -config nightlies -inputQueue app3  
-procSchema imauser -procName proc3 -dbName aDB -dbUser imauser -dbPwd aSecret
```

```
db2mq1sn run -configDB -config nightlies
```

関連資料:

- 325 ページの『disable_MQFunctions』
- 327 ページの『enable_MQFunctions』

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管理者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 DB2MSCS.CFG 入力ファイルを指定します。このパラメーターが指定されない場合、DB2MSCS ユーティリティは、現行のディレクトリーにある DB2MSCS.CFG ファイルを読み取ります。

-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。例えば、次のようなものがあります。

```
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.EE というファイルで、これは単一

db2mscs - Windows フェイルオーバー・ユーティリティーのセットアップ

パーティション・データベース環境の例になっています。もう 1 つは DB2MSCS.EEE で、これは、パーティション・データベース環境の例です。

DB2MSCS.CFG ファイルのパラメーターは次のようになっています。

DB2_INSTANCE

DB2 インスタンスの名前。このパラメーターは、グローバルな有効範囲を持っているため、DB2MSCS.CFG ファイル内で一度だけ指定します。

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 を移行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 (*domain\user* のように指定)。このパラメーターは、グローバルな有効範囲を持っているため、DB2MSCS.CFG ファイル内で一度だけ指定します。

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、指定された名前の MSCS グループが存在していない場合は、そのグループが新しく作成されます。むろん、グループが既に存在している場合は、そのグループがターゲット・グループになります。このパラメーターより後に指定された MSCS リソースは、別の GROUP_NAME パラメーターが指定されるまで、このグループに作成または移動されます。このパラメーターは、各グループにつき 1 つ指定してください。

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・システムに必要です。

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・システムのためのデータベース・ドライブ・マッピングのターゲット MSCS ディスクを指定する、オプション・パラメーター。このパラメーターは、データベースの作成コマンドで指定されたドライブからディスクをマップすることにより、データベースが作成されるディスクを指定します。このパラメーターを指定しない場合は、DB2DRVMP ユーティリティを使用して手動でデータベース・ドライブ・マッピングを登録する必要があります。

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。

関連資料:

- 93 ページの『db2drvmp - DB2 データベースのドライブ・マップ』

db2mtrk - メモリー・トラッカー

データベースやエージェントなどの、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。

- 現行サイズ
- 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)

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。このコマンドは、そのデータベース・パーティションの情報だけを返し、リモート・サーバーの情報は戻しません。

許可:

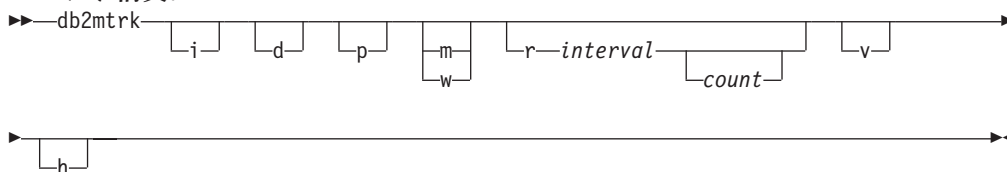
以下のいずれか。

- 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。
- p 専用メモリーを表示します。
- m 各プールの最大値を表示します。
- w 各プールの最高水準点を表示します。
- r 反復モード

interval

次のメモリー・トラッカーの呼び出しまでの待機秒数 (反復モード)。

count 反復回数。

-v 冗長出力。

-h ヘルプ画面を表示します。 **-h** を指定する場合、ヘルプ画面だけが表示され、他の情報は表示されません。

例:

以下の呼び出しは、データベースおよびインスタンスの通常の値を戻し、 10 秒ごとに反復します。

```
db2mtrk -i -d -v -r 10
```

以下の出力サンプルを参考にしてください。

db2mtrk -i -d -p コマンドは、以下の出力を表示します。

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	other		
128.0K	64.0K	9.6M	4.8M	192.0K		

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	other		
128.0K	64.0K	9.6M	4.8M	192.0K		

Memory for agent 970830

other	apph	appctlh
64.0K	64.0K	64.0K

Memory for agent 4460644

other	appctlh	apph
64.0K	64.0K	64.0K

db2mtrk -i -d -p -v コマンドは、以下の出力を表示します。

Tracking Memory on: 2006/01/17 at 15:25:36

Memory for instance

Database Monitor Heap is of size 589824 bytes
Other Memory is of size 8388608 bytes
Total: 8978432 bytes

Memory for database: AJSTORM

Backup/Restore/Util Heap is of size 65536 bytes
Package Cache is of size 655360 bytes
Catalog Cache Heap is of size 131072 bytes
Buffer Pool Heap (1) is of size 35913728 bytes
Buffer Pool Heap (System 32k buffer pool) is of size 589824 bytes

db2mtrk - メモリー・トラッカー

```
Buffer Pool Heap (System 16k buffer pool) is of size 327680 bytes
Buffer Pool Heap (System 8k buffer pool) is of size 196608 bytes
Buffer Pool Heap (System 4k buffer pool) is of size 131072 bytes
Shared Sort Heap is of size 65536 bytes
Lock Manager Heap is of size 10092544 bytes
Database Heap is of size 4980736 bytes
Other Memory is of size 196608 bytes
Total: 53346304 bytes
```

Memory for database: CMGARCIA

```
Backup/Restore/Util Heap is of size 65536 bytes
Package Cache is of size 655360 bytes
Catalog Cache Heap is of size 131072 bytes
Buffer Pool Heap (1) is of size 35913728 bytes
Buffer Pool Heap (System 32k buffer pool) is of size 589824 bytes
Buffer Pool Heap (System 16k buffer pool) is of size 327680 bytes
Buffer Pool Heap (System 8k buffer pool) is of size 196608 bytes
Buffer Pool Heap (System 4k buffer pool) is of size 131072 bytes
Shared Sort Heap is of size 65536 bytes
Lock Manager Heap is of size 10092544 bytes
Database Heap is of size 4980736 bytes
Other Memory is of size 196608 bytes
Total: 53346304 bytes
```

Memory for agent 970830

```
Other Memory is of size 65536 bytes
Application Heap is of size 65536 bytes
Application Control Heap is of size 65536 bytes
Total: 196608 bytes
```

Memory for agent 4460644

```
Other Memory is of size 65536 bytes
Application Control Heap is of size 65536 bytes
Application Heap is of size 65536 bytes
Total: 196608 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、または -p フラグのいずれかを指定する必要があります。
3. -p フラグが指定されると、エージェント ID によってオーダーされた専用メモリー使用量に関する詳細情報が戻されます。
4. 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。
5. 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。
6. 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。
7. 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。

関連概念:

- 「パフォーマンス」の『DB2 におけるメモリーの割り振り』

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 ベース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:user,password] [/p:logical_port] [/h:hostname]
                        [/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
```

関連資料:

- 192 ページの『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』
- 195 ページの『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』

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・システムに不整合が生じる可能性があるためです。

許可:

新しいデータベース・パーティション・サーバーが追加されるコンピューターに対するローカル管理者権限。

コマンド構文:

```
db2ncrt -n:—dbpartitionnum—/u:—username,password—
      /i:—instance_name— /m:—machine_name— /p:—logical_port—
      /h:—host_name— /g:—network_name— /o:—instance_owning_machine—
```

コマンド・パラメーター:

/n:dbpartitionnum

データベース・パーティション・サーバーを識別する固有のデータベース・パーティション番号。1～999 の範囲の番号を指定できます。

/u:domain_name¥username,password

DB2 のドメイン、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。

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

/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 **db2ncrt** コマンドが呼び出される場合、このパラメーターは必須です。

例:

インスタンス所有のコンピューター SHAYER 上で、インスタンス TESTMPP に新しいデータベース・パーティション・サーバーを追加する場合、新しいデータベース・パーティション・サーバーがデータベース・パーティション 2 で、論理ポート 1 を使用する場合には、次のコマンドを入力します。

```
db2ncrt /n:2 /u:QBPAULZ\paulz,g1reeky /i:TESTMPP /m:TEST /p:1 /o:SHAYER
```

関連資料:

- 190 ページの『db2nchg - データベース・パーティション・サーバー構成の変更』

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

- 195 ページの『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』

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 ベース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・システムからデータベース・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・システム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

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

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。

関連資料:

- 190 ページの『db2nchg - データベース・パーティション・サーバー構成の変更』
- 192 ページの『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』

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 '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 'f' オプションを使用して、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を比較します。 **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ずに変更されたか、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 'l' オプションは、現行のカーネル・パラメーターをリストします。
- m 'm' オプションは、物理メモリーのサイズを上書きします。単位は GB。通

常、db2osconf ユーティリティ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 'n' オプションは、システム上の CPU の数を上書きします。通常、db2osconf ユーティリティは 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 'p' オプションは、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 's' オプションはスケール因数を設定します。デフォルトのスケール因数は 1 であり、大部分のワークロードはこの値で十分です。スケール因数が 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。スケール因数はカーネル・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、スケール因数が 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 't' オプションは、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。一方が不釣り合いに小さい場合、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** ユーティリティーを実行した場合に生成される出力例を以下に示します。受け取る結果はマシンに特定のものですので、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ブートが必要になります。カーネル・パラメーターをある程度高い値に設定することにより、将来その値を変更する必要性を少なくするか、なくすることができます。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
```

関連タスク:

db2osconf - カーネル・パラメーター値のためのユーティリティー

- 「*DB2 Universal Database* サーバー機能 概説およびインストール」の『カーネル・パラメーターの変更 (Solaris オペレーティング環境)』

関連資料:

- 「*DB2 Universal Database* サーバー機能 概説およびインストール」の『推奨されるカーネル構成パラメーター (HP-UX)』

db2pd - DB2 データベースのモニターおよびトラブルシューティング

db2pd ユーティリティは、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:

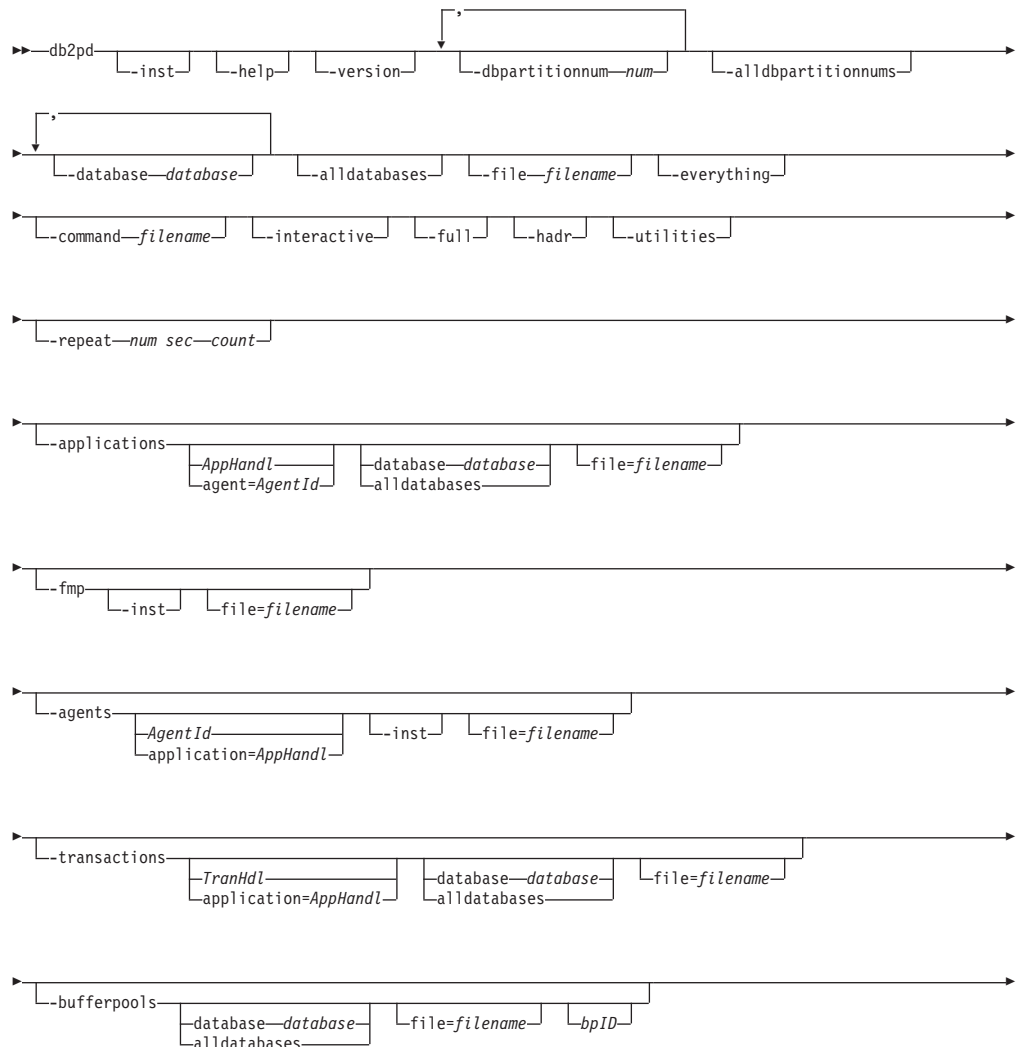
以下のいずれか。

- Linux および UNIX の場合、*sysadm* 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。

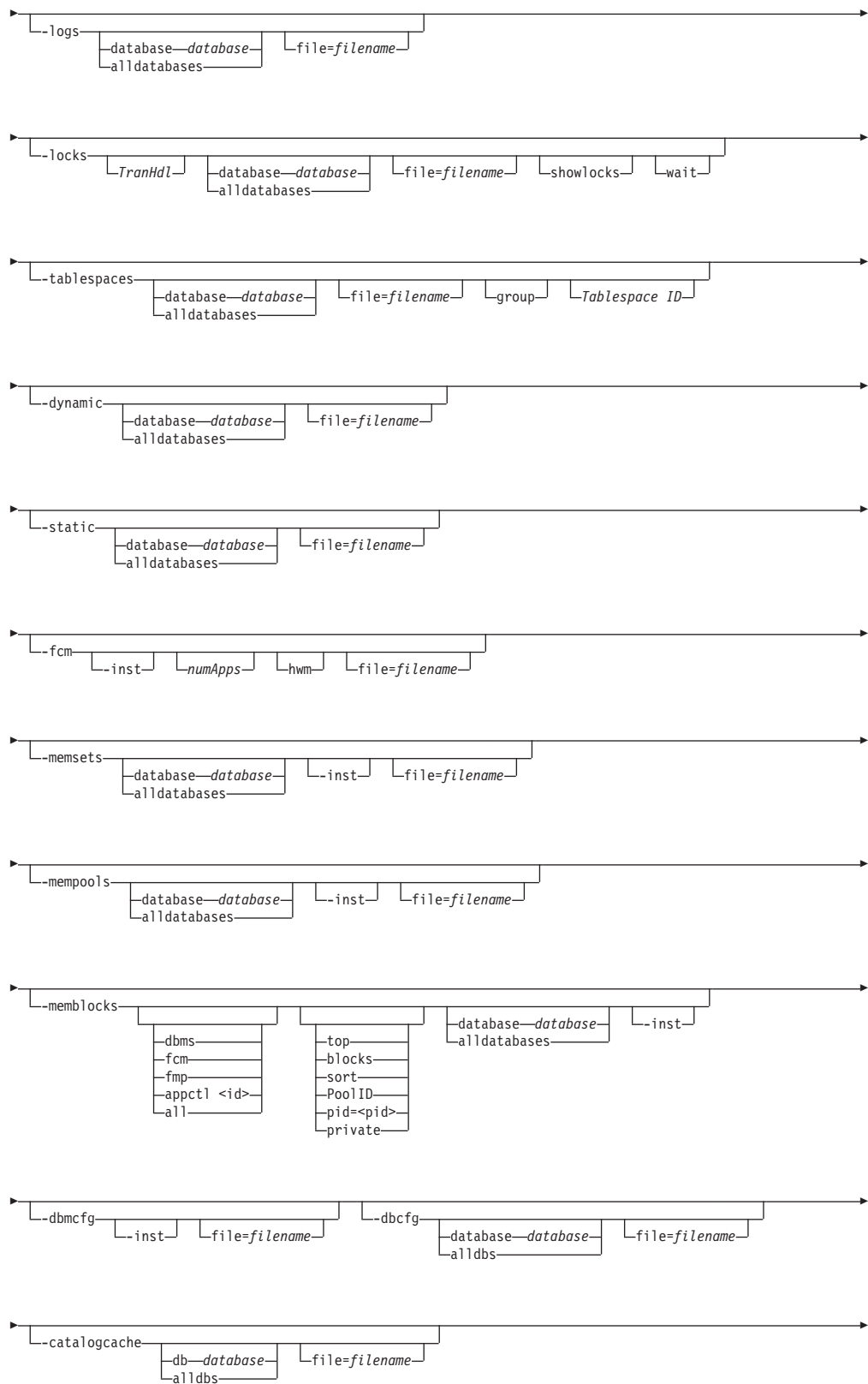
必要な接続:

接続に必要な最低限の要件はありません。しかし、データベースの有効範囲オプションが指定されている場合、コマンドが要求された情報を戻す前に、そのデータベースをアクティブにする必要があります。

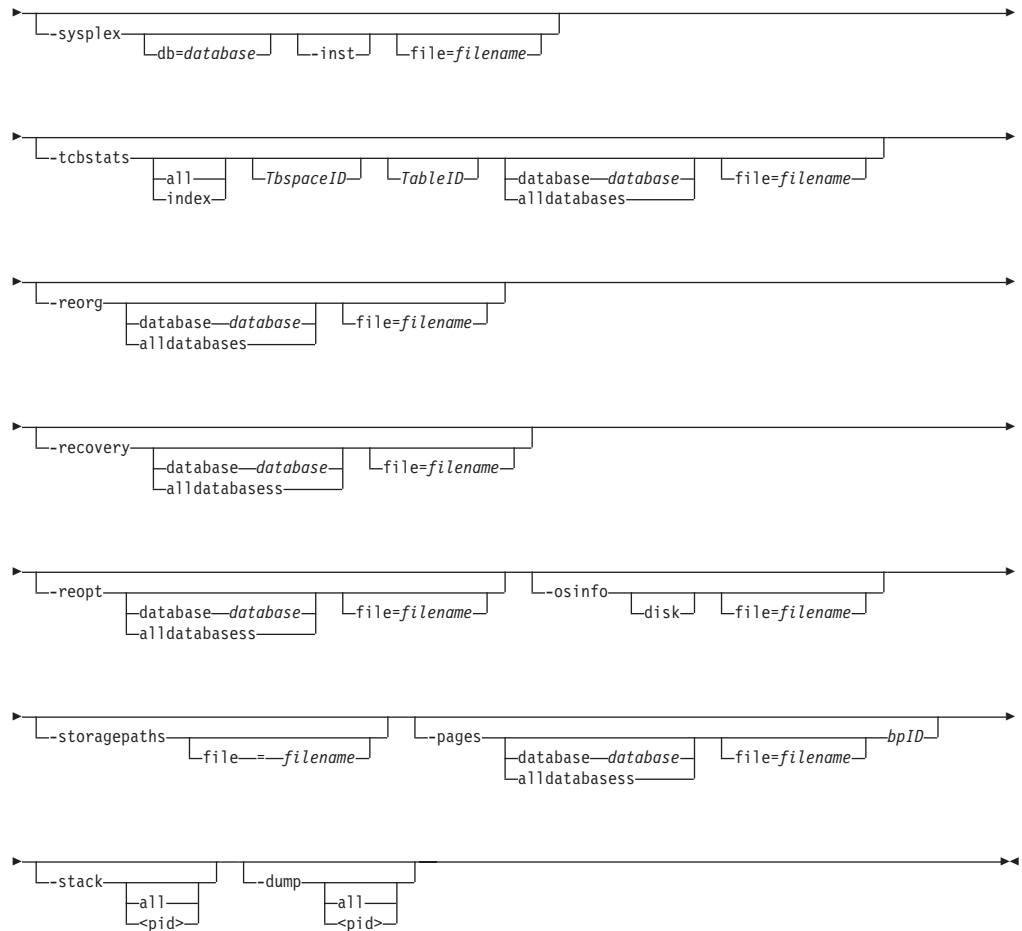
コマンド構文:



db2pd - DB2 データベースのモニターおよびトラブルシューティング



db2pd - DB2 データベースのモニターおよびトラブルシューティング



コマンド・パラメーター:

-inst インスタンス範囲のすべての情報を戻します。

-help オンライン・ヘルプ情報を表示します。

-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。

-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。

-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。

-database database

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。

-alldatabases

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。

-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。

-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。

-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。

-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。

-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。

-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。

-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。

-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。

-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。

-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。

-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。

db2pd - DB2 データベースのモニターおよびトラブルシューティング

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。

-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。

-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。

-logs ログ・ファイルに関する情報を戻します。

-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、このオプションに `showlocks` オプションを指定します。パーティション表および個々のデータ・パーティション上の行およびブロックのロックについて、`showlocks` はデータ・パーティション ID を行の一部としてロック情報と共に表示します。

待ち状態にあるロックおよびそれらのロックの所有者を戻すには、`wait` オプションを指定します。

-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、このオプションに `group` オプションを指定します。

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、このオプションに `tablespace` オプションを指定します。

-dynamic

動的 SQL の実行に関する情報を戻します。

-static 静的 SQL の実行とパッケージに関する情報を戻します。

-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、このオプションに `-inst` オプションを指定します。

- DB2 インスタンス開始以来の、アプリケーションによる FCM バッファおよびチャネル消費量の最高水準点を取得するには、このオプションに `hwm` オプションを指定します。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データベースから切断されていても、保存されます。

- 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、このオプションに `numApps` オプションを指定します。

-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、このオプションに `-inst` オプションを指定します。

-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。

-memblocks

メモリー・プールに関する情報を戻します。

- このオプションを `dbms` オプションと共に指定すると、`dbms` メモリー・セット内のブロックだけを報告します。
- このオプションを `fcm` オプションと共に指定すると、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。
- このオプションを `fmp` オプションと共に指定すると、`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。
- このオプションを `appctl <id>` オプションと共に指定すると、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。
- このオプションを `all` オプションと共に指定すると、すべてのメモリー・セットからのブロックを報告します。
- このオプションを `top` オプションと共に指定すると、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。
- このオプションを `blocks` オプションと共に指定すると、各セットのメモリー・ブロックを報告します。
- このオプションを `sort` オプションと共に指定すると、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。
- このオプションを `PoolID` オプションと共に指定すると、特定プールからのメモリー・ブロックを報告します。
- このオプションを `pid=<pid>` オプションと共に指定すると、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)
- このオプションを `private` オプションと共に指定すると、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)

-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` オプションを指定します。

-dbcfg データベース構成パラメーターの設定を戻します。

-catalogcache

カタログ・キャッシュに関する情報を戻します。

-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** 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、このオプションに **-inst** オプションを指定します。

-tcbstats

表と索引に関する情報を戻します。特定の表スペースに関する情報を表示するには、このオプションに **TbspaceID** オプションを指定します。

特定の表に関する情報を表示するには、このオプションと共に **TableID** オプションを指定します。 **TableID** オプションを使用するとき、**TbspaceID** オプションが必須となります。

-reorg 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。

-recovery

リカバリー・アクティビティに関する情報を戻します。

-reopt **REOPT ONCE** オプションを使用して再度最適化された、キャッシュに入れている **SQL** ステートメントに関する情報を戻します。

-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。

-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。

-pages バッファ・プール・ページに関する情報を戻します。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。

-stack **DIAGPATH** ディレクトリー内にスタック・トレース・ファイルを生成します。

- このオプションを **all** オプションと共に指定すると、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。
- このオプションを **<pid>** オプションと共に指定すると、特定のプロセス **ID** に対してスタック・トレース・ファイルを生成します。

-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。

- このオプションを **all** オプションと共に指定すると、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。
- このオプションを **<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。

例:

db2pd - DB2 データベースのモニターおよびトラブルシューティング

クライアント要求にサービスを提供しているエージェントに関する情報を得るには、以下のようにしてコマンド行から **db2pd** コマンドを使用します。

```
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
```

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。

- 209 ページの『**-applications**』
- 209 ページの『**-fmp**』
- 210 ページの『**-agents**』
- 210 ページの『**-transactions**』
- 212 ページの『**-bufferpools**』
- 216 ページの『**-logs**』
- 217 ページの『**-locks**』
- 218 ページの『**-tablespaces**』
- 222 ページの『**-dynamic**』
- 223 ページの『**-static**』
- 224 ページの『**-fcm**』
- 226 ページの『**-memsets**』
- 227 ページの『**-mempools**』
- 228 ページの『**-memblocks**』
- 229 ページの『**-dbmcfg**』
- 229 ページの『**-dbcfg**』
- 229 ページの『**-catalogcache**』
- 232 ページの『**-sysplex**』
- 233 ページの『**-tcbstats**』
- 235 ページの『**-reorg**』
- 237 ページの『**-recovery**』

- 238 ページの『-reopt』
- 239 ページの『-osinfo』
- 242 ページの『-storagepaths』
- 242 ページの『-pages』

-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。

ApplHandl

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。

-fmp パラメーター:

-fmp パラメーターに関して、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。

-agents パラメーター:

-agents パラメーターについては、以下の情報が戻されます。

AppHandl

アプリケーション・ハンドル (ノードと索引を含む)。

AgentPid

エージェント・プロセスのプロセス ID。

Priority

エージェントの優先順位。

Type エージェントのタイプ。

State エージェントの状態。

ClientPid

クライアント・プロセスのプロセス ID。

Userid エージェントを実行するユーザー ID。

ClientNm

クライアント・プロセスの名前。

Rowsread

エージェントによって読み取られた行数。

Rowswrtn

エージェントによって書き込まれた行数。

LkTmOt

エージェントのロック・タイムアウト設定。

-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。

ApplHandl

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。

db2pd - DB2 データベースのモニターおよびトラブルシューティング

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 ユーティリティーを実行した結果として発生しました。
- 0x00000020。トランザクションは、優先順位の高いアプリケーションにロックを譲渡します (通常、この値は DB2 データベース・システムが自己調整および自己管理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。

-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・バッファース・プールの ID。

Max Bufferpool ID

すべてのアクティブ・バッファース・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ース・プールの最大 ID。

Num Bufferpools

使用できるバッファース・プールの数。

ID バッファース・プールの ID。

Name バッファース・プールの名前。

PageSz

バッファース・プール・ページのサイズ。

PA-NumPgs

バッファース・プールのページ領域にあるページ数。

BA-NumPgs

バッファース・プールのブロック領域にあるページ数。バッファース・プールがブロック・ベースの I/O に関して使用できない場合、この値は 0 です。

BlkSize

バッファース・プールのブロック領域にあるブロックのブロック・サイズ。バッファース・プールがブロック・ベースの I/O に関して使用できない場合、この値は 0 です。

NumTbsp

バッファース・プールを使用する表スペースの数。

PgsLeft

バッファース・プールのサイズが減少している場合、そのバッファース・プールに残されている削除されるページ数。

CurrentSz

ページ内のバッファース・プールの現行サイズ。

PostAlter

バッファーク・プールが再度開始されるときの、ページ内のバッファーク・プールのサイズ。

SuspndTSCt

バッファーク・プールにマップされる、現在 I/O が中断されている表スペースの数。すべてのバッファーク・プールに関して 0 が戻される場合、データベース I/O は中断されていません。

DatLRds

バッファーク・プール・データ論理読み取り。REGULAR および LARGE 表スペースに対して、バッファーク・プール (論理) から要求されたデータ・ページの数を示します。

DatPRds

バッファーク・プール・データ物理読み取り。REGULAR および LARGE 表スペースに対して、表スペース・コンテナ (物理) から読み取られたデータ・ページの数を示します。

HitRatio

公式 1 ($\text{DatPRds} / \text{DatLRds}$) の使用による、バッファーク・プール内のデータ・ページのヒット率。

TmpDatLRds

バッファーク・プール一時データ論理読み取り。TEMPORARY 表スペースに対して、バッファーク・プール (論理) から要求されたデータ・ページの数を示します。

TmpDatPRds

バッファーク・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、バッファーク・プール内の一時データ・ページのヒット率。

IdxLRds

バッファーク・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、バッファーク・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

バッファーク・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、バッファーク・プール内の索引ページのヒット率。

TmpIdxLRds

バッファーク・プール一時索引論理読み取り。TEMPORARY 表スペースに対して、バッファーク・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・プ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・プール内の一時索引ページのヒット率。

DataWrts

バッファーク・プール・データ書き込み。バッファーク・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を示します。

IdxWrts

バッファーク・プール索引書き込み。バッファーク・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・プ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・プールを使用しない書き込み操作の回数。

DirWrtReqs

直接書き込み要求。データの 1 つ以上のセクターを直接書き込む要求の回数。

DirWrtTime

直接書き込み時間。直接書きこみを実行するために必要な経過時間 (ミリ秒)。

AsDatRds

バッファーク・プール非同期データ読み取り。すべてのタイプの表スペースに対して、非同期エンジン・ディスパッチ可能単位 (EDU) によって表スペース・コンテナ (物理) から読み取られたデータ・ページの数を示します。

AsDatRdReq

バッファーク・プール非同期読み取り要求。非同期読み取り要求の数。

AsIdxRds

バッファーク・プール非同期索引読み取り。すべてのタイプの表スペースに対して、非同期エンジン・ディスパッチ可能単位 (EDU) によって表スペース・コンテナ (物理) から読み取られた索引ページの数を示します。

AsIdxRdReq

バッファ・プール非同期索引読み取り要求。索引ページに対する非同期読み取り要求の数。

AsRdTime

バッファ・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ー・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ー・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・バッファのないバッファ・プール。事前選択されたビクティム・バッファをエージェントが使用できなかった回数。

UnRdPFetch

未読プリフェッチ・ページ。プリフェッチャーが読み取ったページで、一度も使用されなかったページの数。

-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 ログのエクステンツのサイズ (ページ数)。

Pages ログ内のページ数。

Filename

ログのファイル名。

-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

モード ロック・モード。可能な値は次のとおりです。

- no lock
- 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きに解放されません。

Att ロックの属性。可能な値は以下のとおりです。

- 0x01: 使用できるようになるまで待機。
- 0x02: エスカレーションによる取得。
- 0x04: ブロック「内の」RR ロック。
- 0x08: 挿入ロック。
- 0x10: RR スキャンによるロック。
- 0x20: 行の更新/削除のロック。
- 0x40: 新規ロック要求の許可。
- 0x80: 新規ロックのリクエスト。

ReleaseFlg

ロック解放フラグ。可能な値は以下のとおりです。

- 0x80000000: SQL コンパイラーによるロック。
- 0x40000000: 非ユニークな、トラックされないロック。

-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・プールの ID。

BufIDDisk

次の始動時のこの表スペースのマップ先のバッファ・プールの ID。

FSC これはファイル・システム・キャッシングを表し、**CREATE/ALTER TABLESPACE** の時点でバッファ 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

- 0x0000000 - NORMAL
- 0x0000001 - QUIESCED: SHARE
- 0x0000002 - QUIESCED: UPDATE
- 0x0000004 - QUIESCED: EXCLUSIVE
- 0x0000008 - LOAD PENDING
- 0x0000010 - DELETE PENDING
- 0x0000020 - BACKUP PENDING
- 0x0000040 - ROLLFORWARD IN PROGRESS
- 0x0000080 - ROLLFORWARD PENDING
- 0x0000100 - RESTORE PENDING
- 0x0000200 - DISABLE PENDING
- 0x0000400 - REORG IN PROGRESS
- 0x0000800 - BACKUP IN PROGRESS
- 0x0001000 - STORAGE MUST BE DEFINED
- 0x0002000 - RESTORE IN PROGRESS
- 0x0004000 - OFFLINE
- 0x0008000 - DROP PENDING
- 0x0010000 - WRITE SUSPENDED
- 0x0020000 - LOAD IN PROGRESS
- 0x0200000 - STORAGE MAY BE DEFINED
- 0x0400000 - STORAGE DEFINITION IS IN FINAL STATE
- 0x0800000 - STORAGE DEFINITION CHANGED PRIOR TO ROLLFORWARD
- 0x1000000 - DMS REBALANCER IS ACTIVE
- 0x2000000 - DELETION IN PROGRESS
- 0x4000000 - CREATION IN PROGRESS

MinRecTime

表スペースの最小リカバリー時間。

NQuiescers

静止者の数。

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

db2pd - DB2 データベースのモニターおよびトラブルシューティング

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ときのサイズ増加単位 (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:

TspId コンテナを所有する表スペースの ID。

ContainNum

表スペース内のコンテナに割り当てられた番号。

Type コンテナのタイプ。可能な値は次のとおりです。

- Path
- Disk
- File
- Striped Disk
- Striped File

TotalPgs

コンテナ内のページ数。

UsablePgs

コンテナ内で使用できるページの数。

StripeSet

コンテナが置かれるストライプ・セット (DMS 表スペースにのみ適用)。

Container

コンテナの名前。

-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。

-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。

Blk

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。

-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・パーティション化機能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・パーティション化機能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。

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チャンネル数の最高水準点。

-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 ベースのシステムのみ)。

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。

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。

-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。

-memblocks パラメーター:

-memblocks パラメーターでは、メモリー・セットの個別ブロック、メモリー・プールごとにグループ化されてソートされた合計、およびメモリー・セットのソートされた合計の、3 つの出力のセクションがあります。

メモリー・ブロック:

PoolID メモリー・ブロックを所有するメモリー・プール ID。

PoolName

メモリー・ブロックを所有するメモリー・プール名。

BlockAge

メモリー・ブロックのブロック経過時間。これはブロックが割り振られるときに割り当てられる増分カウンターです。

Size メモリー・ブロックのサイズ (バイト単位)。

I 割り振りのタイプ。値の 1 はブロックが個別に解放されることを示し、値の 0 はプールと共に解放されることを示します。

LOC メモリー・ブロックを割り振ったコードの行。

File ブロックが割り振られたファイル名のハッシュ値。

メモリー・プールごとに報告されてソートされた合計:

PoolID メモリー・ブロックを所有するメモリー・プール ID。

PoolName

メモリー・ブロックを所有するメモリー・プール名。

TotalSize

同じコード行およびファイルから割り振られたブロックの合計サイズ (バイト数)。

TotalCount

同じコード行およびファイルから割り振られたブロックの数。

LOC メモリー・ブロックを割り振ったコードの行。

File ブロックが割り振られたファイル名のハッシュ値。

メモリー・セットごとに報告されてソートされた合計:

PoolID メモリー・ブロックを所有するメモリー・プール ID。

PoolName

メモリー・ブロックを所有するメモリー・プール名。

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)。

db2pd - DB2 データベースのモニターおよびトラブルシューティング

%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。

-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。

-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。

-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・キャッシュで使わ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ーの最大量 (データベース・グローバル・メモリーの最大量まで)。

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)。

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。

-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。

-tcbstats パラメーター:

-tcbstats パラメーターについては、以下の情報が戻されます。

TCB 表情報:

TbpaceID

表スペース ID。

TableID

表 ID。

PartID パーティション表の場合、これはデータ・パーティション ID です。非パーティション表の場合、これは 'n/a' を表示します。

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 TbpaceID 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ンの数。

UDI **RUNSTATS** によって最後に表の統計が更新された後に、表に対して実行された更新、削除、および挿入操作の数。

PgReorgs

再編成が実行されたページ数。

NoChgUpdts

表内の列を変更しなかった更新の数。

Reads 表のモニターがオンになっていたときに表から読み取られた行数。

FscrUpdates

フリー・スペース制御レコードに対する更新の数。

Inserts 表に対して実行された挿入操作の数。

Updates

表に対して実行された更新操作の数。

Deletes

表に対して実行された削除操作の数。

OvFlReads

表のモニターがオンになっていたときに表に対して読み取られたオーバーフローの数。

OvFlCrtes

作成された新しいオーバーフローの数。

注意 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 TbpaceID 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:

db2pd - DB2 データベースのモニターおよびトラブルシューティング

TableName

表の名前。

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。

-reorg パラメーター:

-reorg 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。

- 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

db2pd - DB2 データベースのモニターおよびトラブルシューティング

- 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。

-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

222 ページの『-dynamic』を参照。

Dynamic SQL Environments

222 ページの『-dynamic』を参照。

Dynamic SQL Variations

222 ページの『-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 NULL 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。

db2pd - DB2 データベースのモニターおよびトラブルシューティング

-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。semume から派生します。

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。

-storagepaths パラメーター:

-storagepaths パラメーターに関して、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。

-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。

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。

関連タスク:

- ・ 「問題判別ガイド」の『待機中のロックの所有者を識別する』

関連資料:

- ・ 502 ページの『GET DATABASE CONFIGURATION 』
- ・ 508 ページの『GET DATABASE MANAGER CONFIGURATION 』
- ・ 243 ページの『db2pdcfg - 問題判別動作の DB2 データベースの構成』
- ・ 「SQL リファレンス 第 1 巻」の『SYSCAT.ROUTINES カタログ・ビュー』
- ・ 「SQL リファレンス 第 1 巻」の『SYSCAT.TABLES カタログ・ビュー』

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:

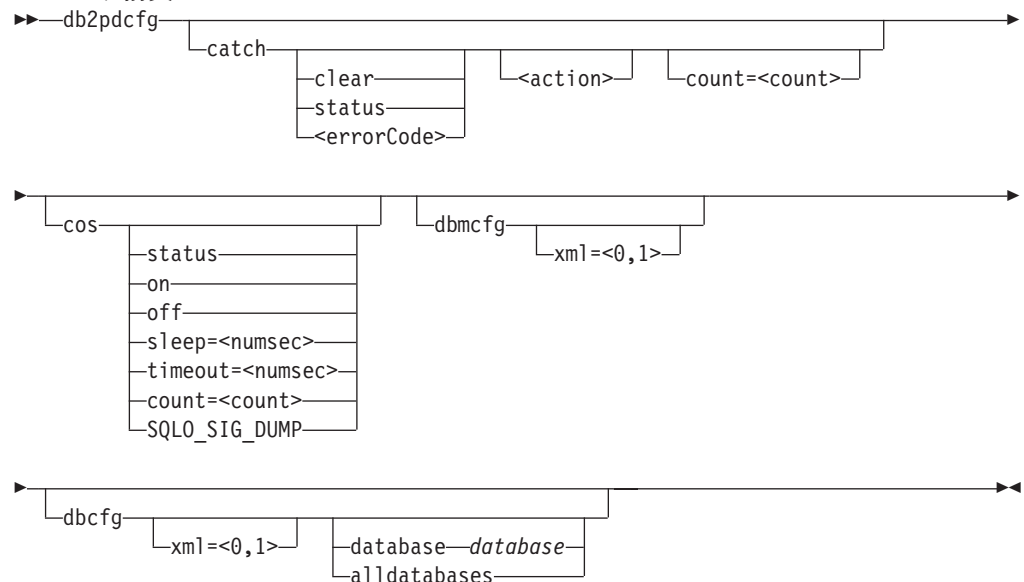
以下のいずれか。

- Linux および UNIX の場合、*sysadm* 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。

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:

-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。

- このオプションを *clear* オプションとともに指定すると、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。
- このオプションを *status* オプションとともに指定すると、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。
- このオプションを *<errorCode>* オプションとともに指定すると、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。

考えられる *errorCode* は以下のとおりです。

- *<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”
- このオプションを <action> オプションとともに指定すると、エラーまたは警告がデータベース・マネージャーによってキャッチされるときに希望するアクションに設定します。

考えられるアクションは以下のとおりです。

- [stack] (デフォルト) - db2diag.log にスタック・トレースを作成します
- [db2cos] (デフォルト) - sqllib/db2cos コールアウト・スクリプトを実行します
- [stopdb2trc] - db2trc 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=0002000300000001F0000000052)

- [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)

- このオプションを count=<count> オプションとともに指定すると、データベース・マネージャーのトラップ中に db2cos を実行する回数をデータベース・マネージャーに指示します。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。
- このオプションを status オプションとともに指定すると、状況を印刷します。
 - このオプションを off オプションとともに指定すると、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。
 - このオプションを on オプションとともに指定すると、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。
 - このオプションを sleep=<numsec> とともに指定すると、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をデータベース・マネージャーに指示します。デフォルトは 3 秒です。
 - このオプションを timeout=<numsec> オプションとともに指定すると、db2cos によって生成される出力ファイルのサイズが大きくなっているかどうかを調べるときの、タイムアウトの時間の長さをデータベース・マネージャーに指示します。デフォルトは 30 秒です。

- このオプションを count=<count> オプションとともに指定すると、データベース・マネージャーのトラップ中に db2cos を実行する回数をデータベース・マネージャーに指示します。デフォルトは 255 です。
- このオプションを SQL0_SIG_DUMP オプションとともに指定すると、SQL0_SIG_DUMP シグナルを受け取るたびに db2cos を実行するようにデータベース・マネージャーに指示します。

-dbmcfg

DBM Config Reserved Bitmap を値 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) に設定します。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。

-dbcfg xml=<0,1>

Database Config Reserved Bitmap を値 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) に設定します。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。

関連概念:

- 「問題判別ガイド」の『db2cos (コールアウト・スクリプト) 出力ファイル』

関連資料:

- 201 ページの『db2pd - DB2 データベースのモニターおよびトラブルシューティング』

db2perf - 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:

Windows のローカル管理者権限。

必要な接続:

なし

コマンド構文:



コマンド・パラメーター:

-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。

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。しかしたいていの場合、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 API 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。

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
```


db2perf - データベース・パフォーマンス値のリセット

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
```

関連資料:

- 「管理 API リファレンス」の『db2ResetMonitor API - データベース・システム・モニターのデータのリセット』

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:

Windows のローカル管理者権限。

必要な接続:

なし

コマンド構文:

```
db2perfi -i
```

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。

1. Windows レジストリーに DB2 カウンター・オブジェクトの名前と説明を追加します。
2. Windows レジストリーの Services キーに、次のようにレジストリー・キーを作成します。

```
HKEY_LOCAL_MACHINE
  ¥System
    ¥CurrentControlSet
      ¥Services
        ¥DB2_NT_Performance
          ¥Performance
            Library=Name of the DB2 performance support DLL
            Open=Open function name, called when the DLL is
              first loaded
            Collect=Collect function name, called to request
              performance information
            Close=Close function name, called when the DLL is
              unloaded
```

関連タスク:

- ・ 「管理ガイド: インプリメンテーション」の『Windows パフォーマンス・モニターへの DB2 の登録』

関連資料:

- ・ 246 ページの『db2perfc - データベース・パフォーマンス値のリセット』
- ・ 249 ページの『db2perfr - パフォーマンス・モニター登録ツール』

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:

Windows のローカル管理者権限。

必要な接続:

なし

コマンド構文:

```

>> db2perfr -r username password
           -u

```

コマンド・パラメーター:

- r ユーザー名およびパスワードを登録します。
- u ユーザー名およびパスワードを登録解除します。

使用上の注意:

- いったんユーザー名とパスワードの組み合わせを DB2 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。

関連タスク:

- 「管理ガイド: インプリメンテーション」の『DB2 パフォーマンス情報へのリモート・アクセスを使用可能にする』

関連資料:

- 246 ページの『db2perf - データベース・パフォーマンス値のリセット』
- 248 ページの『db2perfi - パフォーマンス・カウンター登録ユーティリティー』

db2rbind - Rebind all packages

データベース内のパッケージを再バインドします。

許可:

以下のいずれか。

- *sysadm*

必要な接続:

なし

コマンド構文:

```
db2rbind database -l logfile all -u userid -p password  
-r conservative  
-r any
```

コマンド・パラメーター:

database

再び妥当性検査を行うパッケージが含まれているデータベースの別名を指定します。

-l 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に使用するパス (任意指定) とファイル名 (必須) を指定します。

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 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 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、古典的なバインド・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、パッケージの静的 DML ステートメントの関数およびタイプの解決時に、新しい関数およびデータ・タイプを対象にす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

関数およびタイプの解決時に、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 パスにある関数とタイプだけを対象にします。古典的なバインド・セマンティクスを使用します。これがデフォルトです。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 関数およびタイプの解決時に、SQL パスにあるすべての関数とタイプを対象にします。古典的なバインド・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。

関連資料:

- 389 ページの『**BIND**』
- 638 ページの『**PRECOMPILE**』
- 677 ページの『**REBIND**』

db2_recon_aid - 複数の表の調整

db2_recon_aid ユーティリティは、DB2 RECONCILE ユーティリティへのインターフェースを提供します。RECONCILE ユーティリティは、一度に 1 つの表を操作し、その表のすべての DATALINK 列参照を検証 (およびその検証にしたがって「修理」) します。RECONCILE ユーティリティを複数の表に対して実行する必要が生じることがあります。db2_recon_aid はこの目的で提供されています。

RECONCILE ユーティリティと同様、db2_recon_aid ユーティリティは、調整される DATALINK 列を持つ表を含む DB2 サーバーで実行しなければなりません。

許可:

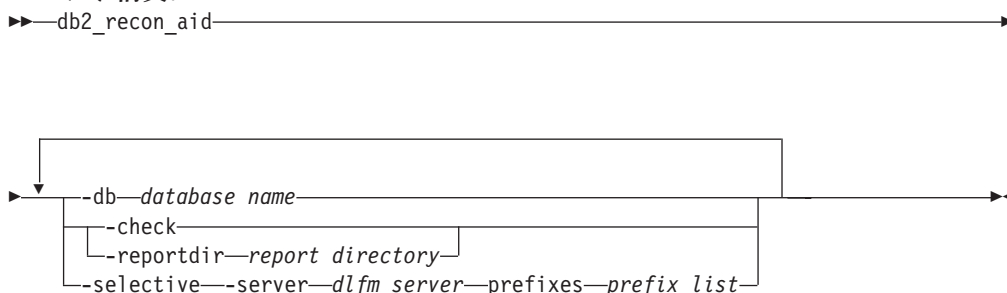
以下のいずれか。

- sysadm
- sysctrl
- sysmaint
- dbadm

必要な接続:

なし。このコマンドは、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:



ここで、*prefix list* はコロンで区切られた 1 つ以上の DLFS 接頭部で、例えば *prefix1:prefix2:prefix3* などです。

コマンド・パラメーター:

-db database name

調整する必要がある DATALINK 列を持つ表を含むデータベースの名前。このパラメーターは必須です。

-check 調整が必要な表をリストします。このパラメーターを使用する場合、調整操作は実行されません。このパラメーターは、-reportdir パラメーターが指定されない場合は必須です。

-reportdir

ユーティリティが調整操作ごとに報告書を入れるディレクトリーを指定します。調整が実行される各表ごとに、<tbchema>.<tbname>.<ext> というフォーマットでファイルが作成されます。ここで、

- <tbchema> は表のスキーマです。

- <tbname> は表の名前です。
- <ext> は .ulk または .exp です。 .ulk ファイルにはデータ・リンク・サーバーでリンク解除されたファイルのリスト、 .exp ファイルにはデータ・リンク・サーバー上で例外だったファイルのリストが入っています。

-check および -reportdir の両方が指定されると、-reportdir は無視されます。

-selective

指定された -server および -prefixes 基準に一致するファイル参照を含む DATALINK 列を持つ表だけを処理します。

- このパラメーターを使用する場合、-server および -prefixes パラメーターの両方を使用する必要があります。
- このパラメーターを使用しない場合、すべてのデータ・リンク・サーバーと、指定された DB2 データベースで登録されるその接頭部は調整されるか、調整が必要なものとしてフラグを付けられます。

-prefixes prefix list

-selective パラメーターの使用時に必要です。1 つ以上のデータ・リンク・ファイル・システム (DLFS) 接頭部の名前を指定します。接頭部値はスラッシュで始まっていなければならない、指定されたデータ・リンク・ファイル・サーバーで登録される必要があります。複数の接頭部名はコロン (:) で区切りますが、組み込みスペースは入れないでください。例えば、次のようなものがあります。

```
/dlfsdir1/smith/:/dlfsdir2/smith/
```

DATALINK 列値のパスは、リスト中の接頭部のいずれかがパスの左端のサブストリングである場合、接頭部リストに一致するものと見なされます。

このパラメーターが使用されない場合、指定された DB2 データベースで登録されるすべてのデータ・リンク・サーバーのすべての接頭部が調整されます。

-server 調整操作が実行されるデータ・リンク・サーバーの名前。パラメーター *dlfm server* は IP ホスト名を表します。このホスト名は、指定された DB2 データベースで登録される DLFM サーバー・ホスト名に完全に一致していなければなりません。

例:

```
db2_recon_aid -db STAFF -check
```

```
db2_recon_aid -db STAFF -reportdir /home/smith
```

```
db2_recon_aid -db STAFF -check -selective -server dlmserver.services.com  
-prefixes /dlfsdir1/smith/
```

```
db2_recon_aid -db STAFF -reportdir /home/smith -selective -server  
dlmserver.services.com -prefixes /dlfsdir1/smith/:/dlfsdir2/smith/
```

使用上の注意:

1. AIX システムまたは Solaris オペレーティング環境の場合、db2_recon_aid ユーティリティは INSTHOME/sql/lib/adm ディレクトリーにあります。INSTHOME はインスタンス所有者のホーム・ディレクトリーです。

2. Windows システムの場合、ユーティリティーは `x:\sqllib\bin` ディレクトリーにあります。x: は、DB2 Data Links Manager をインストールしたドライブです。
3. `db2_recon_aid` は、FILE LINK CONTROL 列属性をもつ DATALINK 列を含む指定されたデータベースにあるすべての表を識別できます。RECONCILE ユーティリティーを介してファイル参照の検証を必要とすることがある列は、このタイプです。`-check` オプションを指定することによって、対象の表を単にリストすることができます。`-reportdir` オプションを指定すると、実際に RECONCILE ユーティリティーが、この表のセットに対して自動的に実行されます。`-selective` オプションを指定すると、`db2_recon_aid` が調整の候補として識別する(特定のデータ・リンク・サーバーおよび 1 つ以上のデータ・リンク・ファイル・システムへの参照を含む、表の DATALINK 列に基づく)、表のセットの範囲を絞ることができます。
4. 解決しようとしている問題によっては、RECONCILE と `db2_recon_aid` ユーティリティーの実行のどちらかを選択する必要があります。オーバーライド時には、調整の必要のあり得る表の数を考慮に入れてください。例えば、次のようなものがあります。
 - DRP または DRNP のような状態の個別の表がある場合に必要なのは、その特定の表に RECONCILE を実行して、その表を通常の表にリストアすることだけかもしれません。
 - 指定されたデータ・リンク・サーバーでデータ・リンク・ファイル・システム(DLFS) が壊れているか、欠落している場合、`db2_recon_aid` (`-selective` オプションを指定) を使って、そのデータ・リンク・サーバーおよびその特定の「接頭部」(DLFS パス) を参照するすべての表を見つけ、これらの表ごとに調整を実行します。
 - 単にデータベース中の DATALINK ファイル参照をすべて検証する場合は、`db2_recon_aid` (`-selective` オプション指定なし) を実行します。
5. 各接頭部は絶対パス (つまりスラッシュで始まるパス) である必要があり、指定された DLFM サーバーで登録されなければなりません。
6. DATALINK 列値のパスは、リスト中の接頭部のいずれかがパスの左端のサブストリングである場合、接頭部リストに一致するものと見なされます。

関連資料:

- 681 ページの『RECONCILE』

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。

ブランク行またはコメント文字 (#) で始まる行は無視されます。

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内にデータベースが既にある場合、不用意に既存の sqldbdir 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。

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、以下のようになります。

1. ディレクトリー /dbpath/jsmith 内のファイルを /dbpath/prodinst に移動します。
2. 以下の構成ファイルと **db2relocatedb** コマンドを使用して、データベース・ファイルに変更を行います。

```
DB_NAME=DATAB1
DB_PATH=/dbpath
INSTANCE=jsmith,prodinst
NODENUM=0
```

例 3

パス /databases/PRODDB のインスタンス inst1 内に存在するデータベース PRODDB です。2 つの表スペース・コンテナのロケーションを、以下のように変更する必要があります。

- SMS コンテナ /data/SMS1 を /DATA/NewSMS1 に移動する必要があります。
- DMS コンテナ /data/DMS1 を /DATA/DMS1 に移動する必要があります。

物理ディレクトリーおよびファイルが、新規のロケーションに移動された後で、新規のロケーションを認識するように、以下の構成ファイルと **db2relocatedb** コマンドを使用して、データベース・ファイルに変更を行います。

```
DB_NAME=PRODDB
DB_PATH=/databases/PRODDB
INSTANCE=inst1
NODENUM=0
CONT_PATH=/data/SMS1,/DATA/NewSMS1
CONT_PATH=/data/DMS1,/DATA/DMS1
```

例 4

db2relocatedb - データベースの再配置

インスタンス db2inst1 に存在するデータベース TESTDB は、パス /databases/TESTDB に作成されました。表スペースは、以下のコンテナと共に作成されました。

```
TS1
TS2_Cont0
TS2_Cont1
/databases/TESTDB/TS3_Cont0
/databases/TESTDB/TS4/Cont0
/Data/TS5_Cont0
/dev/rTS5_Cont1
```

TESTDB は新規システムに移動されます。新規システムのインスタンスは newinst になり、データベースのロケーションは /DB2 になります。

データベースを移動する場合、/databases/TESTDB/db2inst1 ディレクトリーに存在するすべてのファイルは、/DB2/newinst ディレクトリーに移動する必要があります。これは、最初の 5 つのコンテナが、この移動の一部として再配置されることを意味します。(最初の 3 つはデータベース・ディレクトリーに相対で、次の 2 つはデータベース・パスに相対です。) これらのコンテナがデータベース・ディレクトリーまたはデータベース・パス内にあるため、構成ファイルにリストする必要はありません。2 つの残りのコンテナが新規システムで異なるロケーションに移動された場合は、構成ファイルにリストする必要があります。

物理ディレクトリーおよびファイルが新規のロケーションに移動された後で、新規のロケーションを認識するように、以下の構成ファイルと **db2relocatedb** を使用して、データベース・ファイルに変更を行います。

```
DB_NAME=TESTDB
DB_PATH=/databases/TESTDB,/DB2
INSTANCE=db2inst1,newinst
NODENUM=0
CONT_PATH=/Data/TS5_Cont0,/DB2/TESTDB/TS5_Cont0
CONT_PATH=/dev/rTS5_Cont1,/dev/rTESTDB_TS5_Cont1
```

例 5

データベース TESTDB には、データベース・パーティション・サーバー 10 および 20 に 2 つのデータベース・パーティションがあります。このインスタンスは servinst で、データベース・パスは両方のデータベース・パーティション・サーバーで /home/servinst です。データベースの名前は SERVDB に変更され、データベース・パスは両方のデータベース・パーティション・サーバーで /databases に変更されます。さらに、ログ・ディレクトリーはデータベース・パーティション・サーバー 20 で、/testdb_logdir から /servdb_logdir に変更されます。

両方のデータベース・パーティションに変更が行われているため、構成ファイルは各データベース・パーティションに作成され、**db2relocatedb** は対応する構成ファイルを使用する各データベース・パーティション・サーバーで実行される必要があります。

データベース・パーティション・サーバー 10 では、以下の構成ファイルが使用されます。

```
DB_NAME=TESTDB,SERVDB
DB_PATH=/home/servinst,/databases
INSTANCE=servinst
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/*
```

関連資料:

- 137 ページの『db2inidb - ミラーリングされたデータベースの初期化』

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 -log logfile_path ◀◀
```

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。

-log logfile_path

ログ・ファイルのパスを指定します。

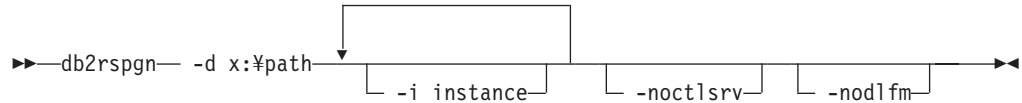
関連概念:

- 「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『高可用性災害時リカバリーの概要』

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** 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** 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。
- noctlsrv**
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。
- nodlfm**
データ・リンク・ファイル・マネージャー・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。

関連概念:

- 「インストールおよび構成 補足」の『応答ファイル生成プログラム (Windows)』

関連タスク:

- 「インストールおよび構成 補足」の『DB2 の応答ファイル・インストールの概要 (Windows)』

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:

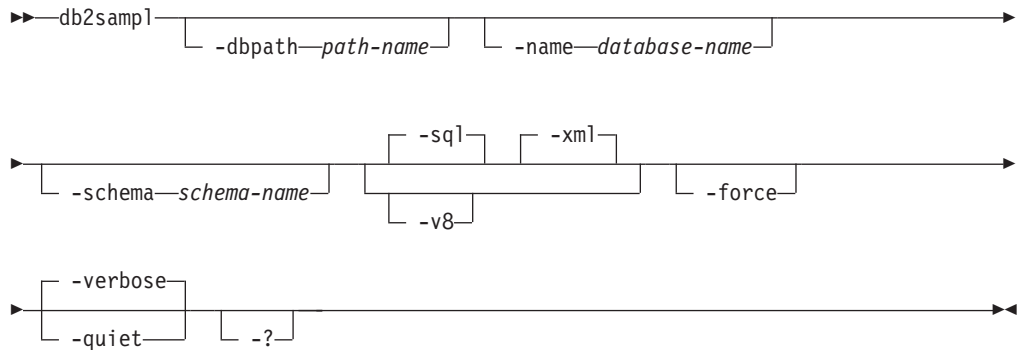
以下のいずれか。

- *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。

-schema *schema-name*

データベース・オブジェクトの作成で使用するデフォルト・スキーマを指定します。すべてのデータベース・オブジェクトの名前がスキーマ名で修飾されます。スキーマ名は、スキーマ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、*schema-name* は、現行ユーザーの許可 ID に対応する CURRENT_SCHEMA 特殊レジスターの値です。

-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。

このオプションが明示的に指定されているか、デフォルトで暗黙指定されている場合、サンプル・データベースは、ネイティブの XML フィーチャーを処理するための前提条件である Unicode (UTF-8) コード・セットで作成されます。これは、UCA400_NO 照合、および C (Canadian) テリトリリーを使用して作成されます。

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。

-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止します。

-? db2sampl コマンド構文ヘルプを戻します。

db2saml 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 **db2saml** 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 (Canadian) 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:

- オペレーティング・システム環境から派生したデフォルトのコード・セットおよび照合を使用して SAMPLE という名前のデータベースを作成します。

db2sampl - サンプル・データベースの作成

- 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 DB2 クライアントから発行することはできません。
- **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・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場合に、**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。

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、スキーマ *myschema* 内に SQL データベース・オブジェクトのみが入った *mysample* という名前のサンプル・データベースを E: ドライブ上に作成し、状況メッセージを表示するには、以下を発行します。

```
db2sampl -dbpath E -name mysample -schema myschema -sql -force -verbose
```

- DB2 バージョン 8 サンプル・データベースを作成するには、以下を発行します。

```
db2sampl -v8
```

関連タスク:

- サンプル・トピックの『サンプル・データベースの作成』

関連資料:

- サンプル・トピックの『SAMPLE データベース』
- 508 ページの『GET DATABASE MANAGER CONFIGURATION 』
- 867 ページの『付録 B. 命名規則』
- 435 ページの『CREATE DATABASE 』

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。

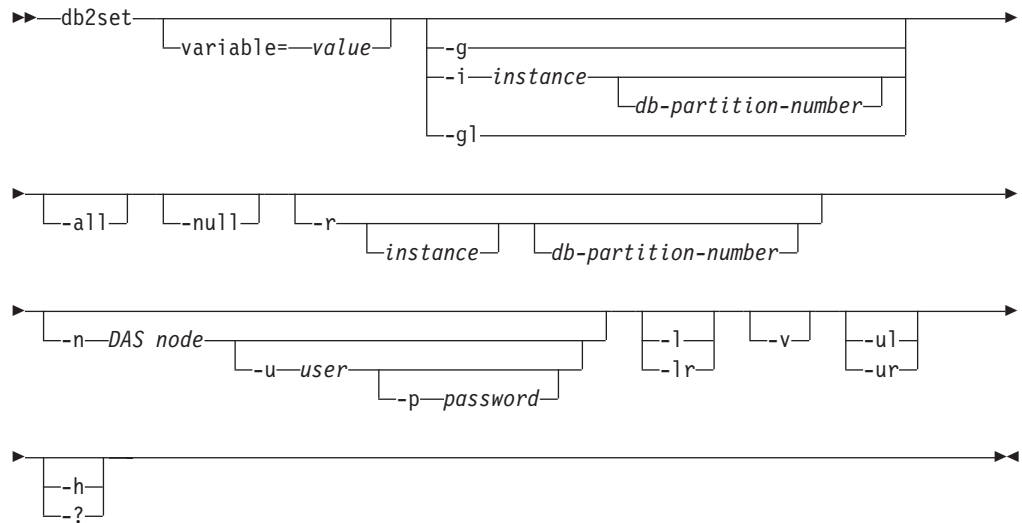
許可:

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= value

指定値に、指定される変数を設定します。変数を削除する場合は、指定される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。

db2set - DB2 プロファイル・レジストリー・コマンド

- 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)

-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。

関連資料:

- 「管理 SQL ルーチンおよびビュー」の『REG_VARIABLES 管理ビュー – 使用中の DB2 レジストリー設定の検索』

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 ベース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 ベースのシステムでは root アクセス。

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

使用されるログ・ファイルの絶対パスとファイル名。

-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 root でログオンするか、"- " フラグを指定した状態で su を使用して、root 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 root 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。

関連資料:

- 332 ページの『setup - DB2 のインストール』
- 「DB2 Universal Database サーバー機能 概説およびインストール」の『言語 ID (別の言語で DB2 セットアップ・ウィザードを実行する場合)』

db2sql92 - SQL92 準拠 SQL ステートメント・プロセッサ

フラット・ファイルまたは標準入力のどちらかから SQL ステートメントを読み取り、ステートメントを動的に記述、準備し、応答セットを戻します。複数のデータベースへの同時接続をサポートします。

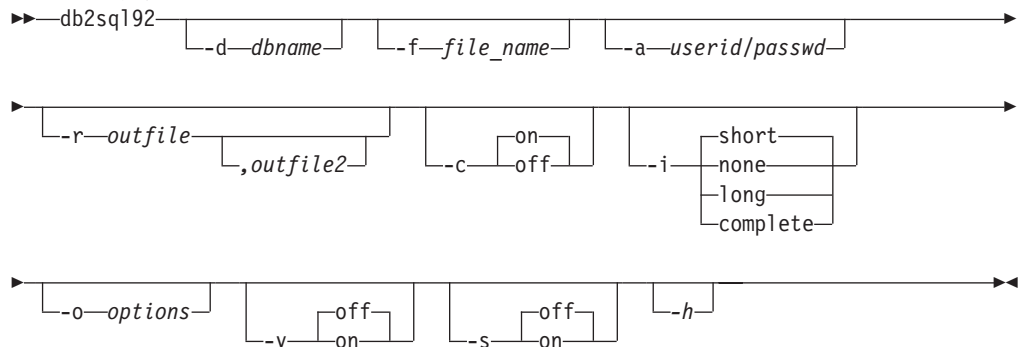
許可:

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:



コマンド・パラメーター:

-d dbname

SQL ステートメントが適用されるデータベースの別名。デフォルトは、**DB2DBDFT** 環境変数の値です。

-f file_name

SQL 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。

注釈テキストは、各行の先頭に 2 つのハイフンを付けて **-- <注釈>** で表します。注釈を出力にも含めるときは、次のように注釈にマークを付けます。
--#COMMENT <注釈>。

ブロック は、一まとまりとして処理されるいくつかの SQL ステートメントからなっています。つまり、ステートメントで使用する情報を 1 つずつ収集するのではなく、すべてのステートメントに必要な情報を一度に収集します。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。

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 <制御オプション> <値>**。有効な制御オプションは、以下のとおりです。

ROWS_FETCH

応答セットから取り出す行数。有効な値は **-1 ~ n** です。デフォルト値は **-1** (すべての行を取り出す) です。

ROWS_OUT

取り出された行のうち出力へ送られる行数。有効な値は $-1 \sim n$ です。デフォルト値は -1 (取り出された行をすべて出力へ送る) です。

AUTOCOMMIT

自動コミットのオン、またはオフを指定します。有効な値は、ON または OFF です。デフォルト値は、ON 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。

TIMESTAMP

タイム・スタンプを生成します。

-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。

-r outfile

照会結果が入る出力ファイル。任意指定の *outfile2* には、結果のサマリーが入ります。デフォルトは標準出力です。

-c 各 SQL ステートメントの実行による変更を自動的にコミットします。

-i 経過時間のインターバル (秒単位)。

none 時間情報を収集しないことを指定します。

short 照会のランタイム。

long 次の照会が開始する経過時間。

complete

準備、実行、および取り出しの時間。別個に表示します。

-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

応答セットから取り出す行数。有効な値は $-1 \sim n$ です。デフォルト値は -1 (すべての行を取り出す) です。

r rows_out

取り出された行のうち出力へ送られる行数。有効な値は $-1 \sim n$ です。デフォルト値は -1 (取り出された行をすべて出力へ送る) です。

-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。収集した値の算術平均と幾何平均の両方を含む、経過時間と CPU 時間のサマリーを提供します。

-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。

使用上の注意:

db2sql92 コマンド・プロンプトで以下を実行できます。

- すべての制御オプション
- SQL ステートメント
- CONNECT ステートメント
- コミット作業
- ヘルプ
- 終了

このツールは、プログラムの単一実行中での異なる複数のデータベース間の切り替えをサポートしています。これを行うために **CONNECT RESET** を発行し、次に **db2sql92** コマンド・プロンプト (stdin) で以下のどちらかを入力します。

```
connect to database
connect to database USER userid USING passwd
```

SQL ステートメントの長さの上限は、65 535 文字です。ステートメントの末尾はセミコロンにしなければなりません。

SQL ステートメントは、反復可能読取り (RR) 分離レベルで実行されます。

照会を実行する場合、LOB を組み込む結果セットのサポートはありません。

関連資料:

- 35 ページの『db2batch - ベンチマーク・ツール』

db2sqljbind - SQLJ プロファイル・バインダー

db2sqljbind は、事前に db2sqljcustomize コマンドを使ってカスタマイズされた直列化プロファイル用の DB2 パッケージをバインドします。

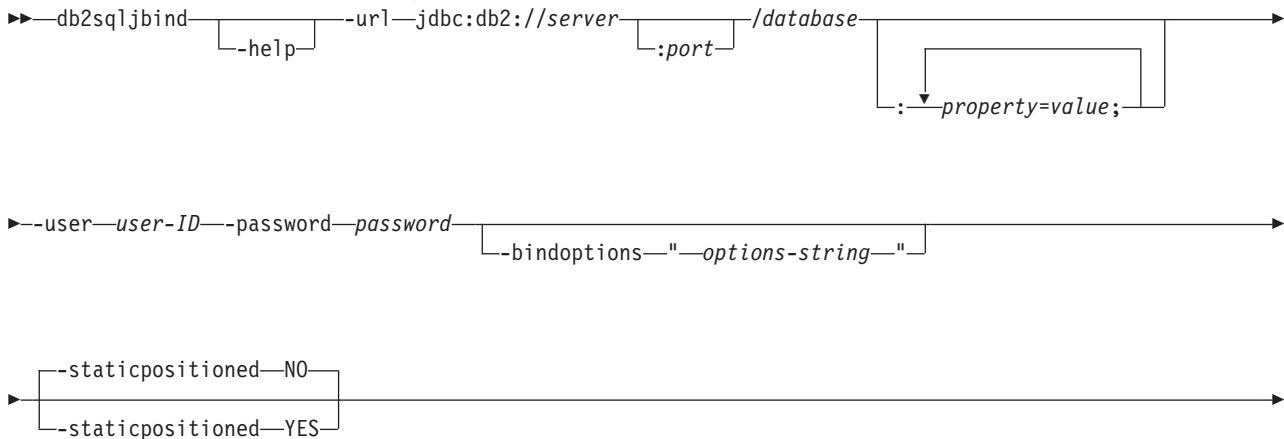
許可:

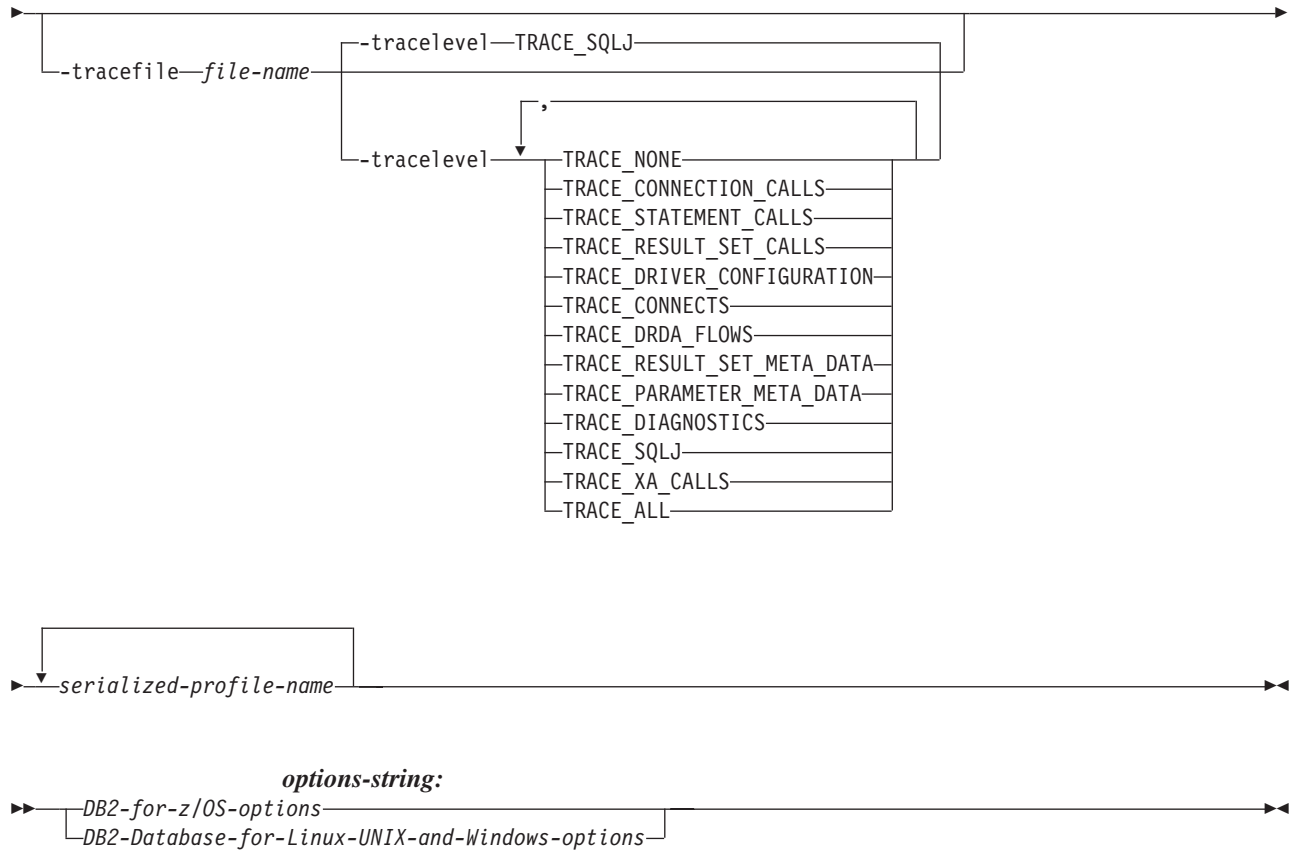
プロセスの特権セットには、以下の権限のうちの 1 つを組み込む必要があります。

- SYSADM 権限
- 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 権限は持ってもバインドを完了するための明示的な特権を持っていない場合、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。

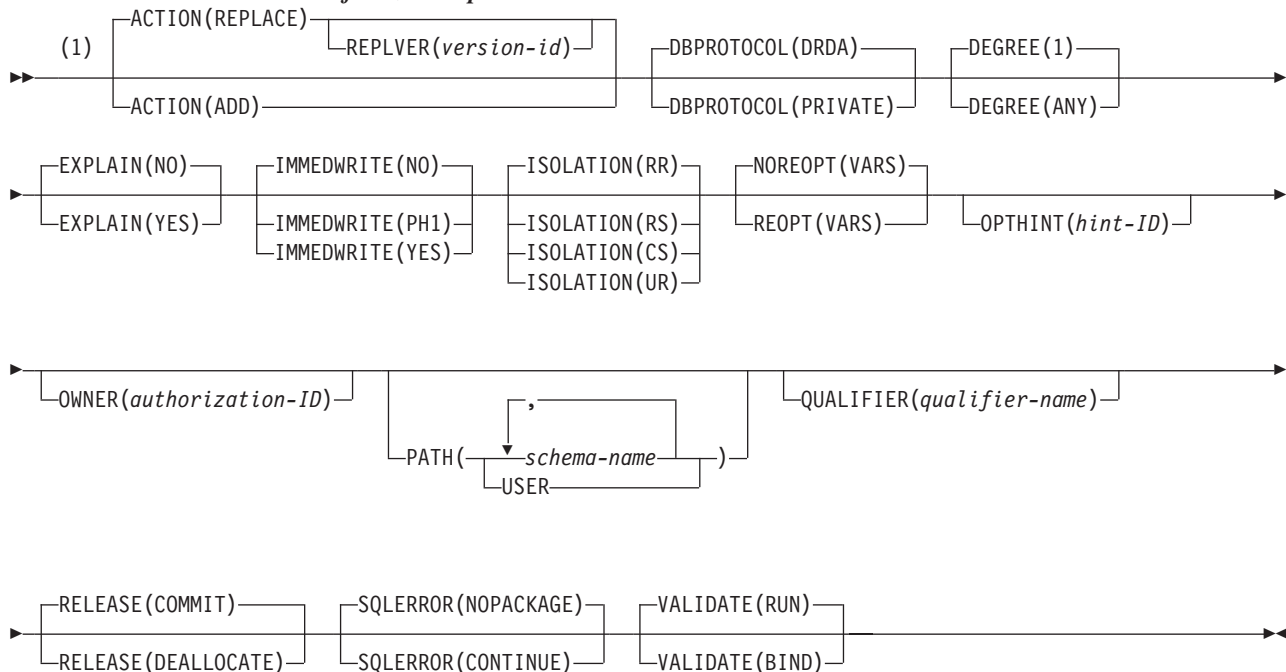
コマンド構文:





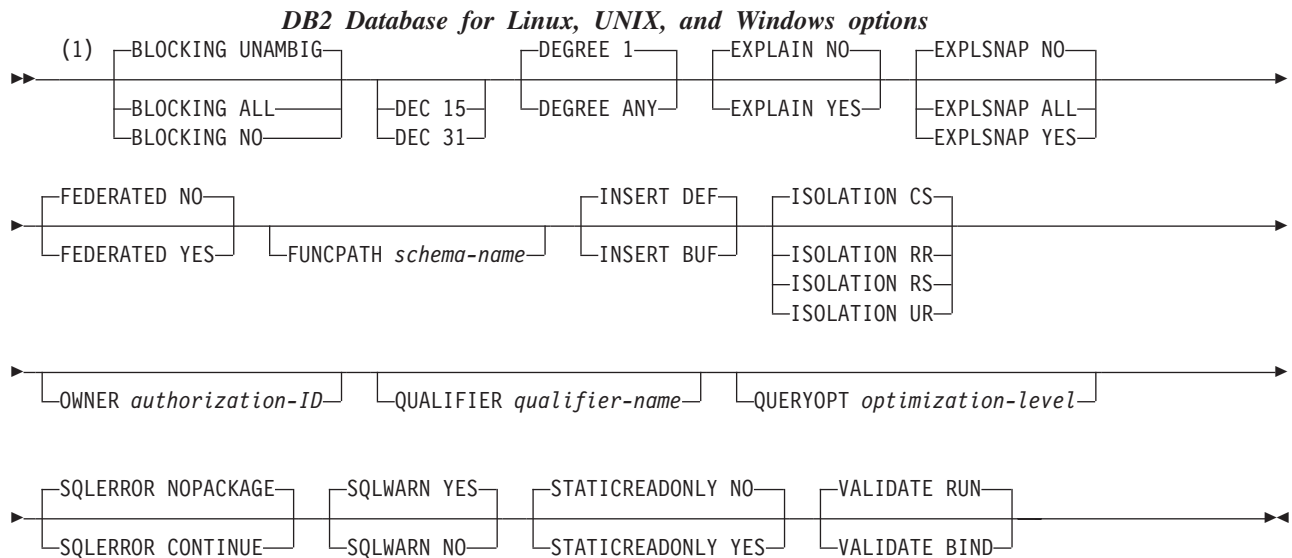
db2sqljbind - SQLJ プロファイル・バインダー

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。



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 ユーティリティが、サポートしている各オプションを説明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 または -onlinecheck オプションが YES の場合、この URL が使用されます。 -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 MVS 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。

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 サーバーへ行う場合、*database* 値は、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 サーバーへ行う場合、*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。例えば以下のようにします。

```
"c:/databases/testdb"
```

property=value;

JDBC 接続のプロパティ。これらのプロパティの定義については、『IBM DB2 Driver for JDBC and SQLJ のプロパティ』を参照してください。

-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意事項:

- 以下の条件が真の場合のみ **VERSION** を指定してください。
 - DB2 Database for Linux, UNIX, and Windows システムでパッケージをバインディングする場合、そのシステムはバージョン 8 またはそれ以上である。
 - 関連パッケージを新規の **VERSION** 値にバインドする前に、プログラムに対して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** 文節を付け加えます。

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 DB2 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、直列化プロファイル用に前に db2sqljcustomize を呼び出したときの -staticpositioned 値と同じでなければなりません。

-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となる 1 つ以上の直列化プロファイルの名前を指定します。直列化された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直列化プロファイルの数です。

いくつかの直列化プロファイルから 1 つの DB2 パッケージをバインドするために、複数の直列化プロファイルの名前を指定する場合、db2sqljcustomize の実行時にそれと同じ直列化プロファイル名を同じ順序ですでに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名は、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したものと同名になります。-rootpkgname または -singlepkgname を指定しなかった場合のパッケージ名は、プロファイル名の最初の 7 バイトに、分離レベル文字を付け加えた名前になります。

db2sqljbind 用の **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。この属性のうちの 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 プログラム内で動的に実行される位置指定の 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) オプションが原因で、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何の影響も与えないようになります。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) を使用した場合、動的 SQL ステートメント内の非修飾表、ビュー、索引、および別名は、バインド・オプション QUALIFIER の値を使って暗黙で修飾されます。QUALIFIER を指定しなかった場合、パッケージ所有者の許可 ID が DB2 で暗黙の修飾子として使用されます。この動作が適当でないプログラムの場合、以下の技法のいずれかを使用して、正しい修飾子を設定することができます。

- 位置指定の UPDATE および DELETE ステートメントが強制的に静的実行されるようにする。位置指定の UPDATE または DELETE ステートメントのカーソル (イテレーター) が、その位置指定の 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は、db2sqljcustomize または db2sqljbind の -staticpositioned YES オプションを使用して、このような措置をとることができます。
- 位置指定の UPDATE および DELETE ステートメント内の DB2 表名を完全修飾する。

関連資料:

- 389 ページの『BIND』
- 279 ページの『db2sqljcustomize - SQLJ プロファイル・カスタマイザー』
- 291 ページの『db2sqljprint - SQLJ プロファイル・プリンター』
- 334 ページの『sqlj - SQLJ 変換プログラム』

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。

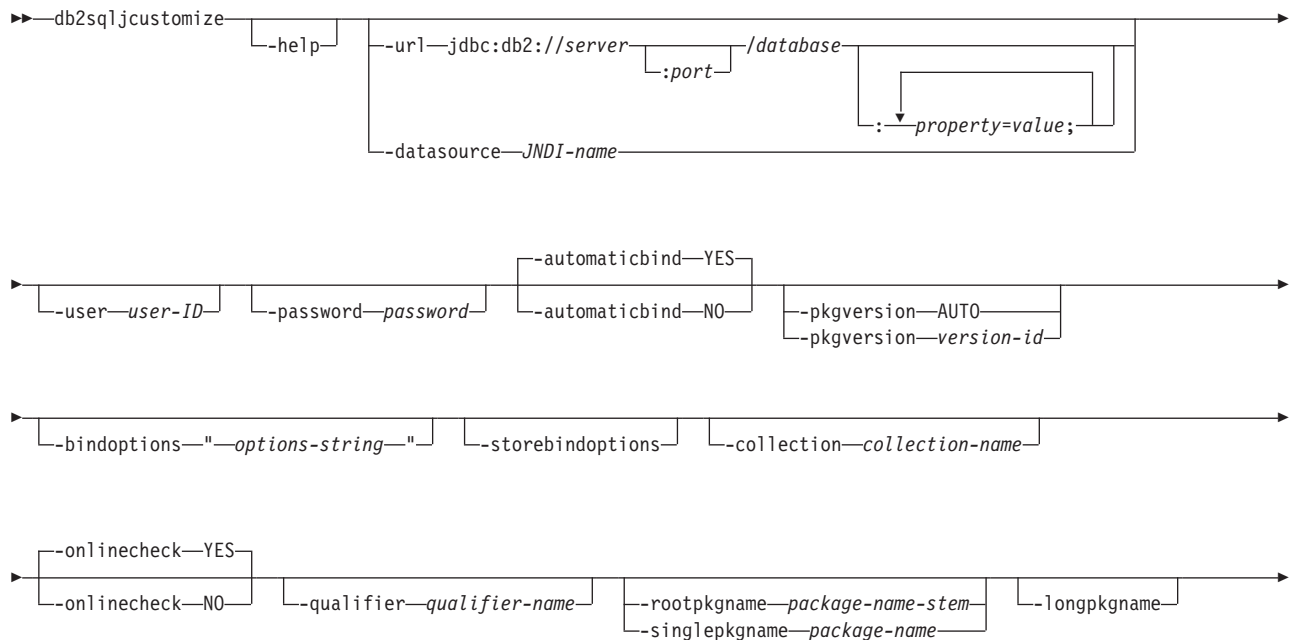
許可:

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。

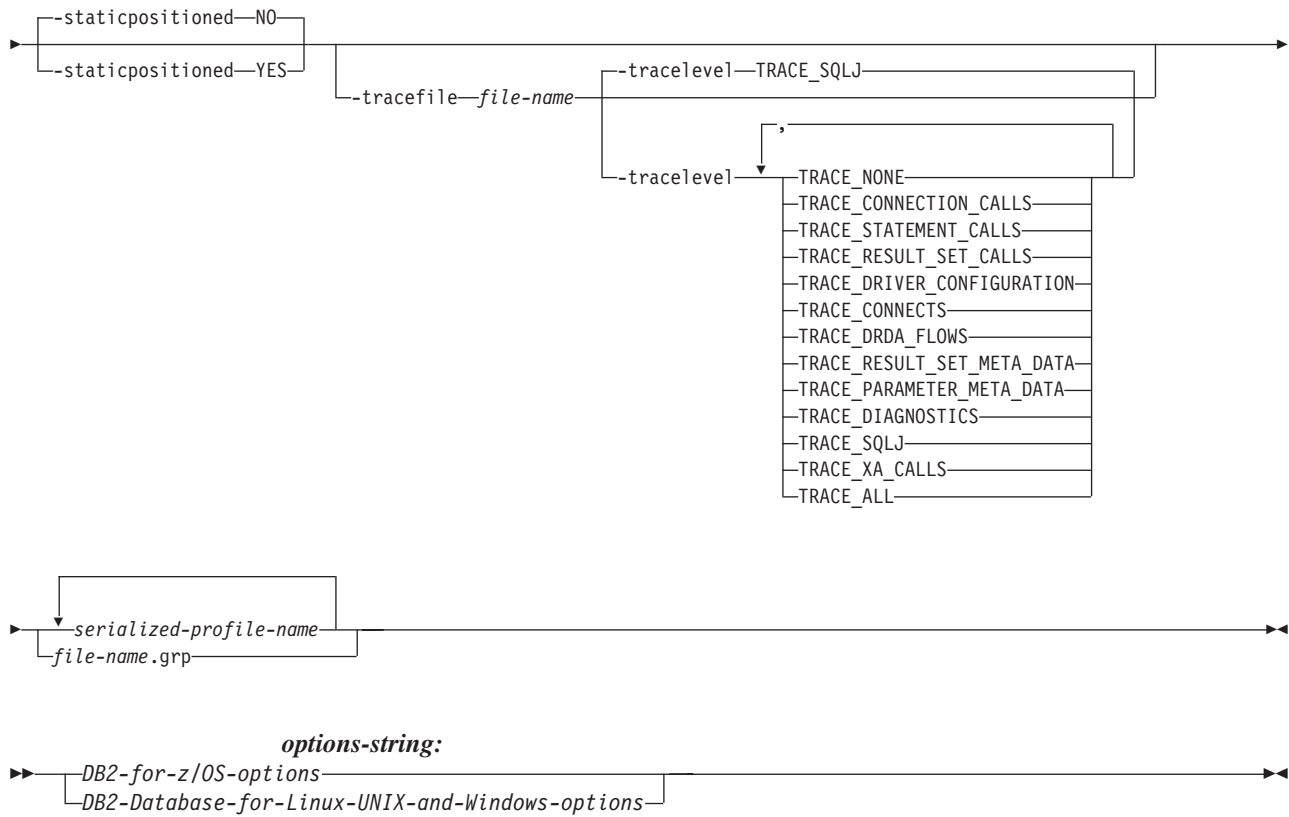
- SYSADM 権限
- DBADM 権限
- パッケージが存在しない場合の BINDADD 特権と、以下のいずれかの特権
 - CREATEIN 特権
 - パッケージのスキーマ名が存在しない場合は、データベースに対する IMPLICIT_SCHEMA 権限
- パッケージが存在する場合に必要な特権:
 - スキーマに対する ALTERIN 特権
 - パッケージに対する BIND 特権

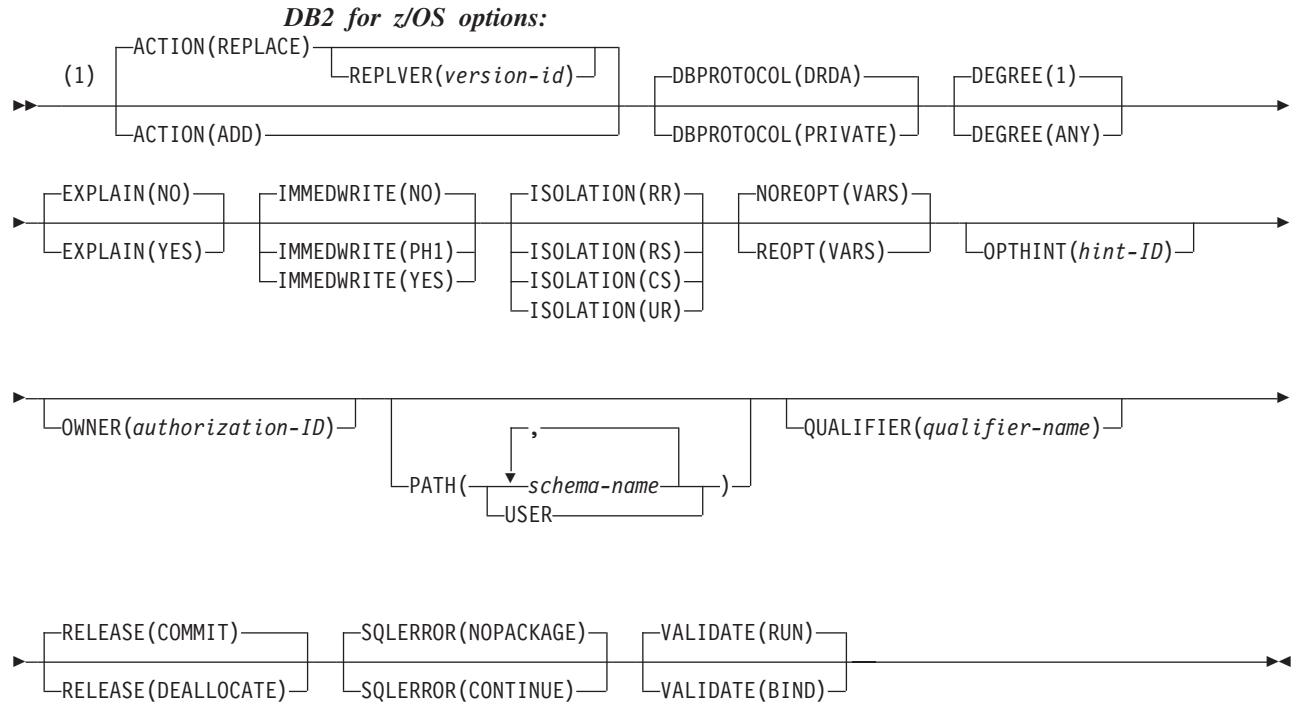
ユーザーには、アプリケーションで静的 SQL ステートメントをコンパイルするのに必要なすべての特権も必要です。グループに認可された特権が、静的ステートメントの許可の検査に使用されることはありません。ユーザーが、SYSADM 権限は持っていない場合、DB2 データベース・マネージャーによって明示的な DBADM 権限が自動的に認可されます。

コマンド構文:



db2sqljcustomize - SQLJ プロファイル・カスタマイザー

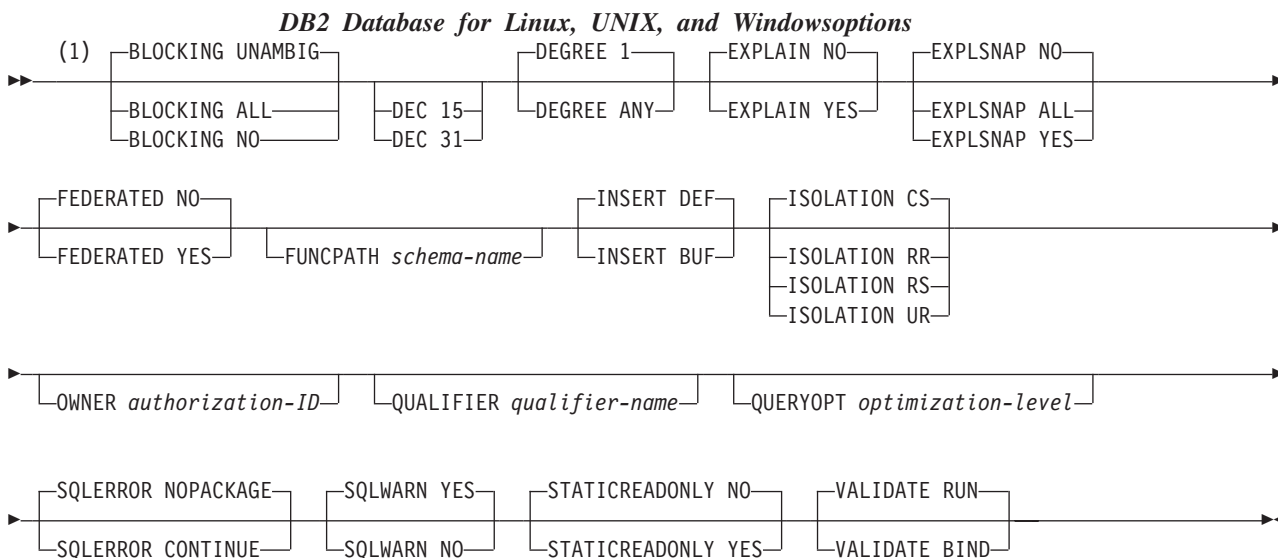




注:

- 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。

db2sqljcustomize - SQLJ プロファイル・カスタマイザー



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が、このカスタマイザー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。

-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 MVS 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。

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 サーバーへ行う場合、*database* 値は、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 サーバーへ行う場合、*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。例えば以下のようにします。

```
"c:/databases/testdb"
```

property=value;

JDBC 接続のプロパティ。これらのプロパティの定義については、『IBM DB2 Driver for JDBC and SQLJ のプロパティ』を参照してください。

-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 DB2 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直列化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、直列化プロファイル内と 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意事項:

- 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` 文節を付け加えます。

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 DB2 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直列化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、直列化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直列化プロファイルを指定すると、パラメーター値は、それぞれの直列化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直列化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 プログラム内の未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。

<i>package-name-stem1</i>	分離レベル UR 用
<i>package-name-stem2</i>	分離レベル CS 用
<i>package-name-stem3</i>	分離レベル RS 用
<i>package-name-stem4</i>	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、プログラムはただ 1 つの分離レベルで実行されます。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。

`-rootpkgname` または `-singlepkgname` を指定しない場合、db2sqljcustomize は直列化された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。直列化された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

db2sqljcustomize - SQLJ プロファイル・カスタマイザー

表 1 は、生成されるパッケージ名の部分およびそれぞれの部分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 1. *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 <i>m</i> 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 2 を参照してください。

表 2 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 2. *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 NOYES

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。

-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 つ以上の直列化プロファイルの名前を指定します。直列化された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

直列化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直列化プロファイルの数です。

以下のいずれかの方法で、直列化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。複数の直列化プロファイル名は、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、直列化プロファイル名を 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。

複数の直列化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、複数の直列化プロファイルから 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。

複数の直列化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 の実行時にその直列化プロファイルを 1 つの DB2 パッケージにバインドするには、直列化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直列化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。 直列化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。

複数の直列化プロファイルの一括カスタマイズ: 複数の直列化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直列化プロファイルの、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直列化プロファイルを作成してから、別のデータ・ソースでその直列化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直列化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・タイプの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直列化プロファイル内に 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、ドライバは、カスタマイズ済み直列化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。カスタマイズ済み直列化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。直列化プロ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 つの直列化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。直列化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. 新規の直列化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。オリジナルの直列化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、直列化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直列化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。

関連資料:

- 389 ページの『BIND』
- 291 ページの『db2sqljprint - SQLJ プロファイル・プリンター』
- 272 ページの『db2sqljbind - SQLJ プロファイル・バインダー』

db2sqljprint - SQLJ プロファイル・プリンター

db2sqljprint は、カスタマイズ済みバージョンの DB2 プロファイルの内容をプレーン・テキスト形式で印刷します。

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、その中に含まれている SQL 操作に関する情報は、プロファイルと呼ばれる SQLJ 生成のリソース・ファイルに保管されます。プロファイルは、元の入力ファイルの名前に続く接尾部 `_SJProfileN` (ここで `N` は整数) で識別されます。これらの拡張子は `.ser` です。プロファイル名を指定する際に、拡張子 `.ser` は付けても付けなくても構いません。

例:

```
db2sqljprint pgmname_SJProfile0.ser
```

関連資料:

- 279 ページの『db2sqljcustomize - SQLJ プロファイル・カスタマイザー』
- 272 ページの『db2sqljbind - SQLJ プロファイル・バインダー』

db2start - DB2 の開始

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に対する Windows 要件を満たす必要があります。拡張セキュリティが無効になっている場合、ユーザーは Administrators、Server Operators、または Power Users グループのメンバーになっている必要があります。拡張セキュリティが有効になっている場合、ユーザーはデータベースを始動するために Administratorsグループまたは DB2ADMNS グループのメンバーになっている必要があります。

複数パーティション・データベースでの **db2start** 操作がデータベース・マネージャー構成パラメーター the start_stop_timeout 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_timeout にもっと大きい値を指定してください。

関連資料:

- 796 ページの『START DATABASE MANAGER 』

db2stop - DB2 の停止

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。 **db2stop** は、システム・コマンドまたは CLP コマンドとして実行することができます。

複数パーティション・データベースでの **db2stop** 操作がデータベース・マネージャー構成パラメーター `start_stop_timeout` 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_timeout` にもっと大きい値を指定してください。

関連資料:

- 806 ページの『STOP DATABASE MANAGER 』

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ー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。

必要な接続:

なし

コマンド構文:

```

db2support output path
    -a
    -r
    -cd current degree
    -cl collect level
    -co
    -cs current schema
    -d database name
    -c
    -u userid
    -p password
    -f
    -fp function path
    -g
    -h
    -il isolation level
    -l
    -m
    -n
    -ol optimization level
    -op optimization profile
    -ot optimization tables
    -q
    -ra refresh age
    -ro
    -s
    -se embedded SQL file
    -sf SQL file
    -st SQL statement
    -td termination character delimiter
    -v
    -x
  
```

注:

1. このツールをオブティマイザー・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。オブ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 または **-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 または **-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 または **-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 または **-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 または **-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atalog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atalog情報だけを収集します。

-cs または **-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 または **-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 または **-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 または **-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 特殊レジスタの値)。

-g または -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 または -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 または -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 特殊レジスタがブランクに設定されます。

-l または -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 または -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 または -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 または -optlevel

使用する最適化レベル特殊レジスタの値を指定します。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 または -optprofile

使用する最適化プロファイル特殊レジスタの値を指定します。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 または -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の値を指定します。CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION の初期値は「SYSTEM」です。

-p *password* または -password *password*

ユーザー ID のパスワードを指定します。

-q または -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 または -refreshage

リフレッシュ経過時間特殊レジスタの値を指定します。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 または -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 または -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* または -sqlembded *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* または -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* または -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 または -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* または -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 または -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 または -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ブティマイザー・ディレクトリーにコピーします。ファイルは現行ディレクトリーに存在する必要はありませんが、呼び出し元のユーザー ID がファイルを読み取ることができなければなりません。

- さまざまなレベルのパフォーマンス情報を戻すときに。

```
db2support <output_directory> -d <database name> -collect 0
```

db2support ツールは、要求された詳細レベルに基づいてさまざまなレベルのパフォーマンス情報を収集します。0 から 3 まで値が増えるにつれて、より多くの詳細が収集されます。レベル 0 を使用すると、実動データベース用にデータベース・オブジェクトを複製できるようにするカタログ情報と表定義が収集されます。

デフォルトで設定されたオブ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。

オブ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、そしてバッファー・

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・ツールで利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。
- ストアード・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。

関連概念:

- 「問題判別ガイド」の『DB2 データベース の診断と OS の診断の結合』

関連タスク:

- 「問題判別ガイド」の『db2support を使用した環境情報の収集』

db2swtch - デフォルト DB2 コピーの切り替え

デフォルト DB2 コピーを切り替えます。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、DB2 コピー選択ウィザード (DB2 Copy Selection Wizard) が起動し、新規のデフォルト DB2 コピーを設定するために従うことができます。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:

```
▶▶ db2swtch [-l] [-d installation-name]
```

コマンド・パラメーター:

- l** システム上の DB2 データベース製品インストールのリストを表示します。
- d *installation-name***
デフォルト DB2 コピーを設定します。

関連資料:

- 143 ページの『db2iupdt - インスタンスの更新』

db2sync - DB2 シンクロナイザーの開始

サテライトの初期構成および構成の変更が簡単になります。このコマンドは、同期化セッションの進行を開始、停止またはモニターしたり、サテライトの構成情報 (例えば、通信パラメーター) をコントロール・サーバーにアップロードしたりすることにも使用できます。

許可:

なし

必要な接続:

なし

コマンド構文:

```
►► db2sync 

|                                |
|--------------------------------|
| --t                            |
| --s <i>application_version</i> |
| --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。

関連資料:

- 「管理 API リファレンス」の『db2SyncSatellite API - サテライト同期化の開始』
- 「管理 API リファレンス」の『db2SyncSatelliteStop API - サテライト同期化の一時停止』

db2sysstray - 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。 **db2sysstray** 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。

db2sysstray 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、**db2sysstray** 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、**db2sysstray** 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、**db2sysstray** 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 **db2sysstray.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 **db2sysstray** プロセスを使用してモニターできます。**db2sysstray** 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 **db2sysstray** 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。

db2sysstray アイコンは、**db2sysstray** 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。**db2sysstray** 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、**db2sysstray** 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 **db2sysstray** 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、**db2sysstray** 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。

db2sysstray は、Windows プラットフォームでのみ使用できます。

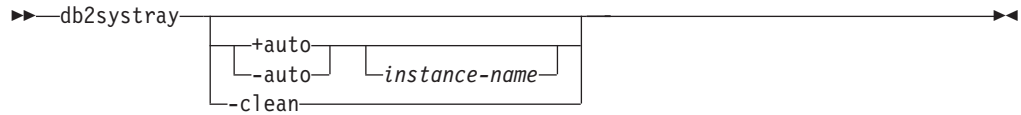
許可:

db2sysstray 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 **db2sysstray** を自動的に開始します。また、**db2sysstray** は、「スタートアップ時にツールを起動 (Launch Tool at Startup)」 **db2sysstray** 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 **db2sysstray** 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、**db2sysstray** 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、**db2sysstray** は自動的に終了します。

-clean **db2sysstray** 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 **db2sysstray.exe** プロセスを停止します。

例:

1. C:\SQLLIB\bin> db2sysstray

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 **db2sysstray** を開始します。

2. C:\SQLLIB\bin> db2sysstray DB2INST1

DB2INST1 という名前のインスタンスに対して **db2sysstray** を開始します。

3. C:\SQLLIB\bin> db2sysstray +auto

デフォルト DB2 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。

4. C:\SQLLIB\bin> db2sysstray +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

db2sysstray - DB2 システム・トレイの開始

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。

関連概念:

- 「管理ガイド: インプリメンテーション」の『コントロール・センターの概要』

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。

有効範囲:

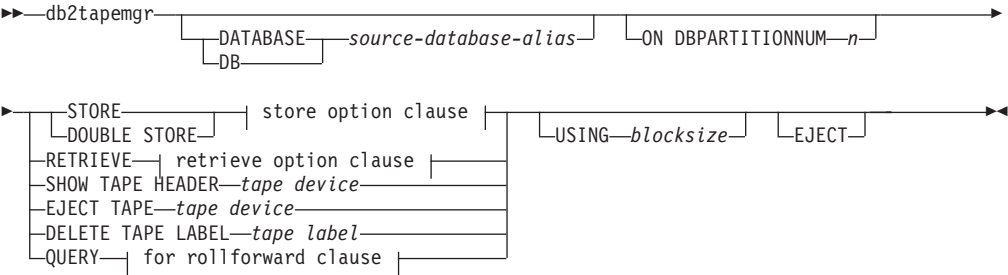
許可:

以下のいずれか。

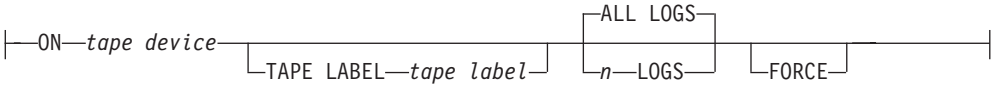
- *sysadm*
- *sysctrl*
- *sysmaint*

必要な接続:

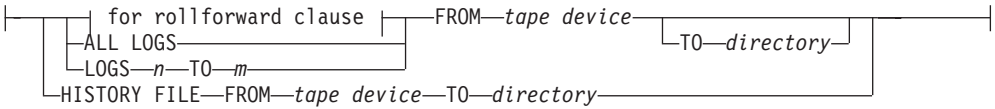
コマンド構文:



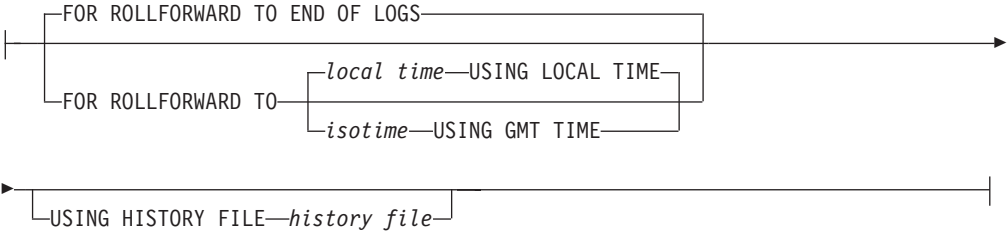
store option clause:



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 文字で、タイム・スタンプ (秒) が最大 14 文字で、合計最大 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 または **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。

例:

関連概念:

- 「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『db2tapemgrを使用したログ・アーカイブ』

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

関連資料:

- 601 ページの『LIST TABLESPACES 』

db2trc - トレース

DB2 インスタンスや DB2 Administration Server のトレース機能を制御します。トレース機能は、操作に関する情報を記録し、この情報を読みやすい形式にフォーマットします。なお、トレース機能を有効にすると、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 サポートの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。

許可:

UNIX オペレーティング・システムで DB2 インスタンスのトレースを行う場合は、以下のいずれかの権限が必要です。

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

UNIX オペレーティング・システムで DB2 Administration Server のトレースを行う場合は、以下の権限が必要です。

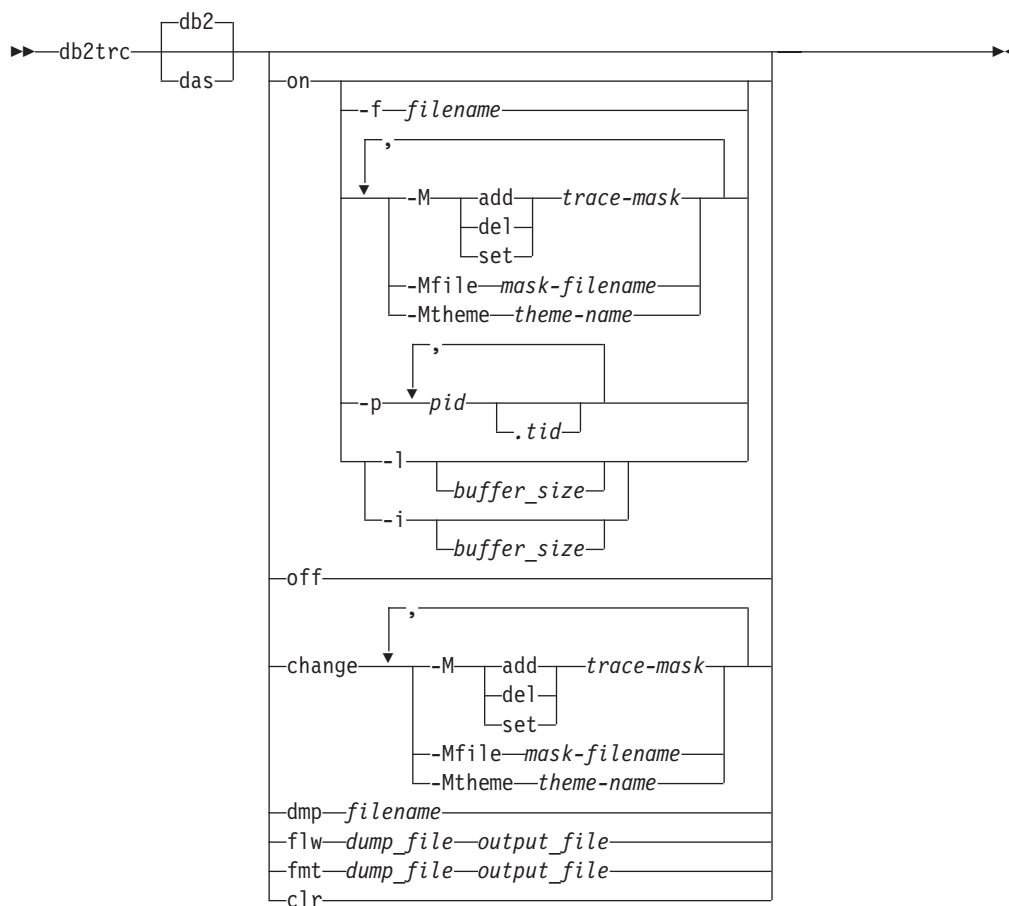
- *dasadm*

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:



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。これがデフォルトです。
- das** すべてのトレース操作を DB2 Administration Server 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。

-f filename

db2trc がオフにされるまで、継続的にトレース情報を指定されたファイルに書き込むことを指定します。このオプションを使用すると、ダンプ・ファイルが極めて大きなものになる可能性があります。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。

-M<action> trace-mask

トレース機能によって記録される操作を制限するトレース・マスクを有効にします。トレース・マスクは、必要に応じて DB2 サポートの技術サポート担当者から提供されます。<action> として可能な値は次のとおりです。

add トレース・マスク・エレメントを追加します。

del トレース・マスク・エレメントを削除します。

set トレース・マスクを特定の値に設定します。それまでのマスクの内容はすべて失われます。

-Mfile *mask-filename*

トレース・マスク・アクションのリストをファイル *mask-filename* からロードします。

-Mtheme *theme-name*

トレース・マスクのテーマをテーマ・ファイルからロードします。デフォルトでは、~/sql1lib/cfg 内のテーマ・ファイルが使用されます。

-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
```

-l [*buffer_size*] | **-i** [*buffer_size*]

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・レコードを保存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重ね書きされます)。'-i' は、最初のトレース・レコードを保存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" 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。

off 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停止します。

```
db2trc off
```

change 既に実行中のトレースのトレース・マスクに関連した値を変更します。トレース・マスクは、必要に応じて DB2 サポートの技術サポート担当者から提供されます。

-Madd *trace-mask*

トレース・マスク・エレメントを追加します。

-Mdel *trace-mask*

トレース・マスク・エレメントを削除します。

-Mset *trace-mask*

トレース・マスクを特定の値に設定します。それまでのマスクの内容はすべて失われます。

-Mfile *mask-filename*

トレース・マスク・アクションのリストをファイル *mask-filename* からロードします。

-Mtheme *theme-name*

トレース・マスクのテーマをテーマ・ファイルからロードします。

dmp 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 dmp db2trc.dmp
```

このパラメーターで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、そのファイルは現行ディレクトリーに保管されます。

flw | fmt

トレースをバイナリー・ファイルにダンプした後、それがテキスト・ファイルにフォーマットされたことを確認します。 **flw** オプション (プロセスやスレッド別にレコードをソートするフォーマット) か、 **fmt** オプション (イベントの発生順にレコードをソートするフォーマット) を使用できます。どちらのオプションの場合でも、ダンプ・ファイル名および生成される出力ファイルの名前を指定します。例えば、次のようなものがあります。

```
db2trc flw db2trc.dmp db2trc.flw
```

clr トレース・バッファーの内容をクリアします。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。ただし、このオプションは、ファイルへのトレース中は無効です。

使用上の注意:

db2trc コマンドは、トレースをオンにする時、ダンプ・ファイルを生成する時、ダンプ・ファイルをフォーマットする時、トレースを再びオフにする時など、何回か発行する必要があります。パラメーターのリストは、そのパラメーターを使用する順序になっています。

トレース・バッファー・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。最小バッファー・サイズは 1 MB です。

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。

関連概念:

- ・ 「問題判別ガイド」の『DB2 トレース・ファイルのダンプ』
- ・ 「問題判別ガイド」の『DB2 トレース・ファイルのフォーマット』

- 「問題判別ガイド」の『db2trc を使用した DB2 トレースの取得』

関連資料:

- 91 ページの『db2drdat - DRDA trace 』
- 「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」の『SkipTrace CLI/ODBC 構成キーワード』

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移行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。

関連資料:

- 「マイグレーション・ガイド」の『移行済みのデータベースのタイプ 1 の索引の変換』

db2undgp - 実行特権の取り消し

外部ストアド・プロシージャで実行特権を取り消します。このコマンドは、外部ストアド・プロシージャに対して使用できます。

データベースの移行中、すべての既存の機能、メソッド、および外部ストアド・プロシージャの EXECUTE 特権は、PUBLIC に付与されます。これによって、SQL データ・アクセスを含む外部ストアド・プロシージャの機密漏れが生じる恐れがあります。ユーザーが特権を持っていない可能性のある SQL オブジェクトにアクセスしないように、db2undgp コマンドを使用します。

コマンド構文:

```
db2undgp [-d dbname] [-h] [-o outfile] [-r]
```

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。その最大長は 8 文字です。

-h

コマンドのヘルプを表示します。

-o outfile

取り消しステートメントを指定されたファイルに出力します。ファイル名の長さは、<= 80 とする必要があります。

-r

取り消しを実行します。

使用上の注意:

少なくとも、-r または -o オプションのどちらか一方を必ず指定してください。

関連資料:

- 「マイグレーション・ガイド」の『移行済みのルーチンに関する PUBLIC の EXECUTE 特権の変更』

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。

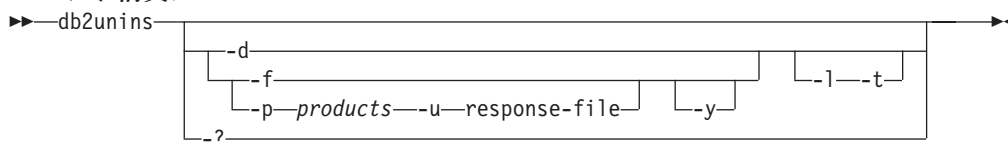
許可:

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、システムがリブートします。バージョン 9 より前の 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、このコマンドを発行できます。

-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"** とします。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。

-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。

-l ログ・ファイルの位置を指定します。

db2unins - DB2 データベース製品のアンインストール

- t 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の使用されます。
- ? **db2unins** のヘルプを表示します。

関連タスク:

- 「インストールおよび構成 補足」の『応答ファイルによる DB2 製品のアンインストール (Windows)』
- 「*DB2 Universal Database* サーバー機能 概説およびインストール」の『DB2 製品のアンインストール (Windows)』

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。

関連資料:

- 435 ページの『CREATE DATABASE 』

db2xpirt - 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・ディレクトリー (DB2INSTPROF) にあるか、または DIAGPATH データベース・マネージャー構成パラメーターが設定されている場合は診断データ・ディレクトリー・パスにあります。これは SQLLIB/BIN ディレクトリーの下にあります。 **db2xpirt** コマンドは、トラップ・ファイルをフォーマットするために、DB2 シンボル・ファイル (.PDB) を使用します。

許可:

DIAGPATH ディレクトリーに対するアクセスを持っている必要があります。

コマンド構文:

```
db2xpirt [ /p path /n ] infile outfile
```

コマンド・パラメーター:

/p path

バイナリー・ファイルおよび PDB ファイルが置かれている位置 (複数可) を指す、セミコロン (;) で区切られたパス。

/v

バージョン情報を表示します。

/n

行番号情報に関係なく、データをフォーマットします。

infile

入力ファイルを指定します。

outfile

出力ファイルを指定します。

例:

DB30882416.TRP というトラップ・ファイルが DIAGPATH に作成されている場合、それを以下のようにフォーマットできます。

```
db2xpirt DB30882416.TRP DB30882416.FMT
```

関連概念:

- 「問題判別ガイド」の『トラップ・ファイル』

関連タスク:

- 「問題判別ガイド」の『トラップ・ファイルのフォーマット (Windows)』

doce_deinstall - DB2 インフォメーション・センターのアンインストール

doce_deinstall ツールと同じインストール・パスにある DB2 インフォメーション・センターをアンインストールします。このコマンドは Linux オペレーティング・システムでのみ使用できます。

doce_deinstall コマンドは、DB2DIR/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] [-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。

-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

```

関連概念:

- 「DB2 Universal Database サーバー機能 概説およびインストール」の『DB2 インフォメーション・センターのインストール・オプション』

関連タスク:

- 「DB2 Universal Database サーバー機能 概説およびインストール」の『db2_deinstall または doce_deinstall コマンドを使用した DB2 製品の削除 (Linux および UNIX)』
- 「DB2 Universal Database サーバー機能 概説およびインストール」の『DB2 製品のアンインストール (Linux および UNIX)』

doce_deinstall - DB2 インフォメーション・センターのアンインストール

関連資料:

- 10 ページの『db2_deinstall - DB2 製品またはフィーチャーのアンインストール』
- 323 ページの『doce_install - DB2 インフォメーション・センターのインストール』

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。

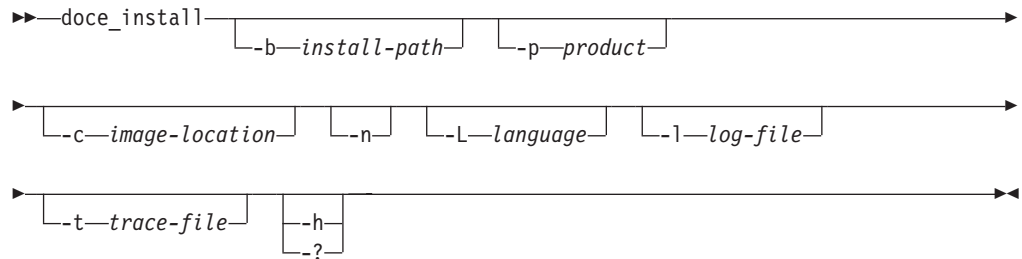
許可:

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、および自動ディスカ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。複数インストール・イメージに関連した自動ディスカバリーについて詳しくは、複数の CD によるインストール (Linux と UNIX) を参照してください。

-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。デフォルトは英語です。複数の言語を同時に

doce_install - DB2 インフォメーション・センターのインストール

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
```

関連概念:

- 「*DB2 Universal Database* サーバー機能 概説およびインストール」の『複数の CD によるインストール (Linux と UNIX)』

関連タスク:

- 「*DB2 Universal Database* サーバー機能 概説およびインストール」の『インフォメーション・センター・デーモンの開始または停止 (Linux)』

関連資料:

- 11 ページの『`db2_install` - DB2 製品のインストール』
- 321 ページの『`doce_deinstall` - DB2 インフォメーション・センターのアンインストール』

disable_MQFunctions

目的

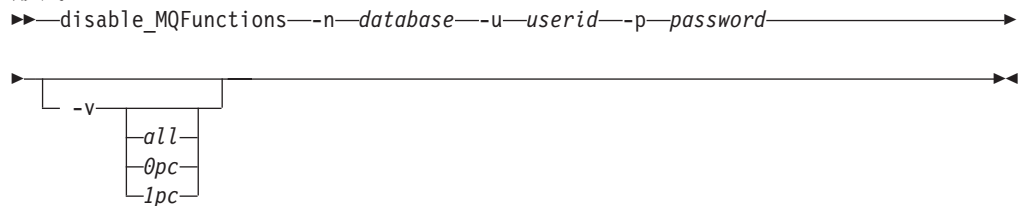
指定したデータベースについて DB2 WebSphere MQ 関数の使用を無効にします。

許可

以下のいずれか。

- *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)

形式



パラメーター

-n database

データベースの名前を指定します。

-u userid

データベースに接続するためのユーザー ID を指定します。

-p password

ユーザー ID のパスワードを指定します。

- v** オプション。これは、トランザクション・ユーザー定義関数と非トランザクション・ユーザー定義関数のサポートのために使用します。値は *all*、*0pc*、*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
```

関連概念:

disable_MQFunctions

- *WS II V9 - MQ and Web Services plugins (DOCE)* の『DB2 内で WebSphere MQ 機能を使用する方法』

関連資料:

- 178 ページの『db2mqdsn - MQ Listener 』
- 327 ページの『enable_MQFunctions』
- 「管理 SQL ルーチンおよびビュー」の『MQPUBLISH スカラー関数』
- 「管理 SQL ルーチンおよびビュー」の『MQREAD スカラー関数』
- 「管理 SQL ルーチンおよびビュー」の『MQREADALL 表関数』
- 「管理 SQL ルーチンおよびビュー」の『MQREADALLCLOB 表関数』
- 「管理 SQL ルーチンおよびビュー」の『MQREADCLOB スカラー関数』
- 「管理 SQL ルーチンおよびビュー」の『MQRECEIVE スカラー関数』
- 「管理 SQL ルーチンおよびビュー」の『MQRECEIVEALL 表関数』
- 「管理 SQL ルーチンおよびビュー」の『MQRECEIVEALLCLOB 表関数』
- 「管理 SQL ルーチンおよびビュー」の『MQRECEIVECLOB スカラー関数』
- 「管理 SQL ルーチンおよびビュー」の『MQSEND スカラー関数』
- 「管理 SQL ルーチンおよびビュー」の『MQSUBSCRIBE スカラー関数』
- 「管理 SQL ルーチンおよびビュー」の『MQUNSUBSCRIBE スカラー関数』

enable_MQFunctions

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:

```

▶▶ enable_MQFunctions --n=database --u=userid --p=password
▶
└─q=queuemanager┐ ┌-force┐ ┌-novalidate┐ ┌-v┐
                  └────────┘ └────────┘ └──┴──┘
                              all
                              0pc
                              1pc

```

コマンド・パラメーター:

- n** 有効にするデータベースの名前を指定します。
- u** 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** ユーザー ID のパスワードを指定します。
- q** 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*、*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

enable_MQFunctions

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 1pc
```

使用上の注意:

DB2 MQ ユーザー定義関数は、このコマンドによって自動的に作成されるスキーマ DB2MQ または DB2MQ1C の下で実行されます。このコマンドを実行する前に、次のようにしてください。

- WebSphere MQ と WebSphere Application Messaging Interface (AMI) がインストールされていることと、WebSphere MQ のバージョンが 5.1 以上であることを確認します。
- 環境変数 \$AMT_DATA_PATH が定義されていることを確認します。
- トランザクション MQ UDF を使用する場合は、フェデレーテッド操作用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。

関連概念:

- *WS II V9 - MQ and Web Services plugins (DOCE)* の『DB2 内で WebSphere MQ 機能を使用する方法』

関連資料:

- 「管理 SQL ルーチンおよびビュー」の『MQREADALL 表関数』
- 178 ページの『db2mq1sn - MQ Listener』
- 325 ページの『disable_MQFunctions』
- 「管理 SQL ルーチンおよびビュー」の『MQPUBLISH スカラー関数』
- 「管理 SQL ルーチンおよびビュー」の『MQREAD スカラー関数』

- 「管理 SQL ルーチンおよびビュー」の『MQREADALLCLOB 表関数』
- 「管理 SQL ルーチンおよびビュー」の『MQREADCLOB スカラー関数』
- 「管理 SQL ルーチンおよびビュー」の『MQRECEIVE スカラー関数』
- 「管理 SQL ルーチンおよびビュー」の『MQRECEIVEALL 表関数』
- 「管理 SQL ルーチンおよびビュー」の『MQRECEIVEALLCLOB 表関数』
- 「管理 SQL ルーチンおよびビュー」の『MQRECEIVECLOB スカラー関数』
- 「管理 SQL ルーチンおよびビュー」の『MQSEND スカラー関数』
- 「管理 SQL ルーチンおよびビュー」の『MQSUBSCRIBE スカラー関数』
- 「管理 SQL ルーチンおよびビュー」の『MQUNSUBSCRIBE スカラー関数』

installFixPack - インストール済み DB2 製品の更新

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。

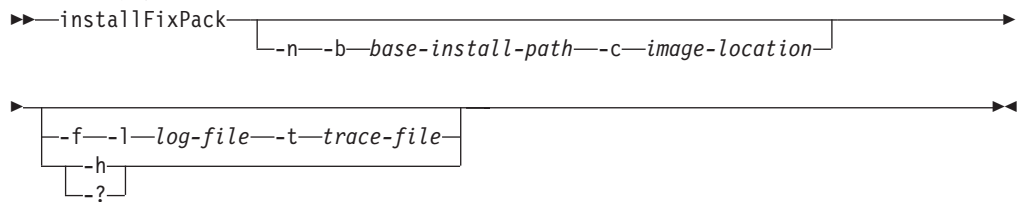
許可:

Root 権限。

必要な接続:

なし。

コマンド構文:



コマンド・パラメーター:

-n 非対話式モードを指定します。

-b *base-install-path*

更新する必要がある DB2 製品がインストールされているパスを指定します。パスの長さは 128 文字に制限され、絶対パス名です。

-c *image-location*

イメージ位置を指定します。単一のフィックスパック・イメージが複数の CD にまたがる場合に、必須です。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できます。例えば、イメージが複数 CD 上にあることを示すには、**-c** CD1 **-c** CD2 と指定します。

-f 強制オプション。このオプションは、インストール済み DB2 製品の高位フィックスパック・イメージの上に低位フィックスパック・イメージを強制的に更新するか、またはインストール済み DB2 製品を同じレベルにリフレッシュします。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。

-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/installFixPack.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、名前を指定したファイルに書き込まれます。

-h/-? 使用法情報を表示します。

例:

- DB2 ESE German が FP5 イメージから /opt/ibm/db2/V9.1 上にインストールされているときに、GA から FP5 への対話式更新を実行するには、以下のコマンドを発行します。

```
./installFixPack
```

- DB2 ESE German が FP5 イメージから /opt/ibm/db2/V9.1 上にインストールされているときに、GA から FP5 へのサイレント更新を実行するには、以下のコマンドを発行します。

```
./installFixPack -b /opt/ibm/db2/V9.1 -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 -b full_path_where_DB2_product_installed
```

関連タスク:

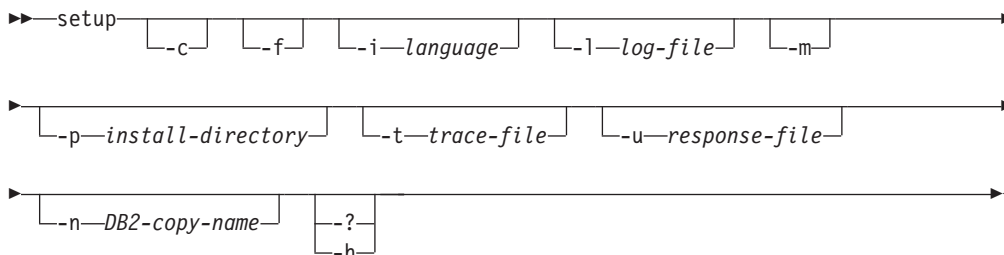
- 「*DB2 Universal Database* サーバー機能 概説およびインストール」の『フィックスパックの適用』

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。

コマンド構文:



コマンド・パラメーター:

- 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インストール・パスがオーバーライドされます。
- , -h 使用法情報を生成します。

関連資料:

- 268 ページの『db2setup - DB2 のインストール』
- 「*DB2 Universal Database* サーバー機能 概説およびインストール」の『言語 ID (別の言語で DB2 セットアップ・ウィザードを実行する場合)』

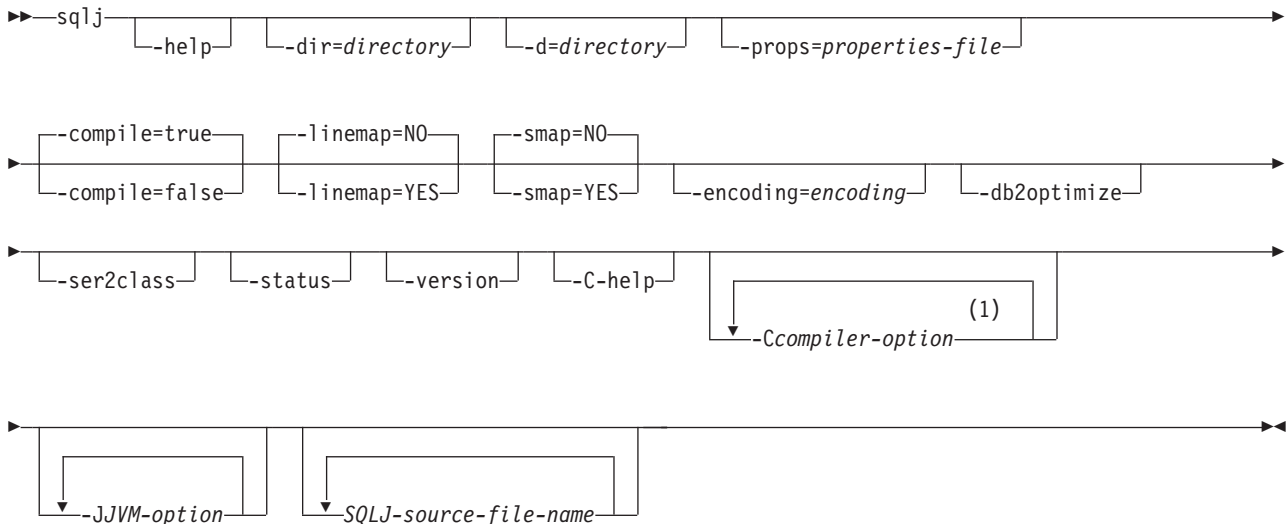
sqlj - SQLJ 変換プログラム

sqlj コマンドは、SQLJ ソース・ファイルを Java ソース・ファイルと、ゼロ個以上の直列化 SQLJ 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:



注:

- 1 `-C-classpath` および `-C-sourcepath` オプションは、SQLJ 変換プログラムのほかに Java コンパイラによっても使用されます。

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、変換プログラムで生成された .java ファイルを書き込むディレクトリ名を指定します。デフォルト・ディレクトリは、SQLJ ソース・ファイルが含まれているディレクトリです。

変換プログラムは、生成後のファイルをディレクトリに書き込むときに、SQLJ ソース・ファイルのディレクトリ構造を使用します。例えば、変換プログラムを使って以下の 2 つのファイルを処理しようとしています。

- file1.sqlj。これは Java パッケージ内にはありません。
- file2.sqlj。これは、Java パッケージ sqlj.test 内にあります。

また、変換プログラムの呼び出し時に、パラメーター `-dir=/src` を指定すると仮定します。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` に書き込み、`file2.sqlj` 用の Java ソース・ファイルをディレクトリー `/src/sqlj/test` に書き込みます。

-d=directory

SQLJ が、変換プログラムで生成されたバイナリー・ファイルを書き込むディレクトリー名を指定します。そのようなファイルとして次のものがあります。

- 直列化プロファイル・ファイル (`.ser` ファイル)
- `sqlj` コマンドが Java コンパイラーを呼び出す場合は、このコンパイラーによって生成されるクラス・ファイル (`.class files`)

デフォルト・ディレクトリーは、SQLJ ソース・ファイルが含まれているディレクトリーです。

変換プログラムは、生成後のファイルをディレクトリーに書き込むときに、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、変換プログラムを使って以下の 2 つのファイル进行处理しようとしているとします。

- `file1.sqlj`。これは Java パッケージ内にはありません。
- `file2.sqlj`。これは、Java パッケージ `sqlj.test` 内にあります。

また、変換プログラムの呼び出し時に、パラメーター `-d=/src` を指定すると仮定します。変換プログラムは、`file1.sqlj` 用の直列化プロファイルをディレクトリー `/src` に書き込み、`file2.sqlj` 用の直列化プロファイルをディレクトリー `/src/sqlj/test` に書き込みます。

-props=properties-file

SQLJ 変換プログラムがオプションのリストを入手するファイル名を指定します。

-compile=truelfalse

SQLJ 変換プログラムが、生成後の 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、生成後の Java ソースをバイトコードにコンパイルします。これがデフォルトです。

false

変換プログラムは、生成後の Java ソースをバイトコードにコンパイルしません。

-linemap=nolyes

Java 例外内の行番号が、SQLJ ソース・ファイル (`.sqlj` ファイル) 内の行番号、または SQLJ 変換プログラム・ファイルによって生成される Java ソース・ファイル (`.java` ファイル) 内の行番号と一致するかどうかを指定します。

no Java 例外内の行番号は Java ソース・ファイル内の行番号と一致します。これがデフォルトです。

yes

Java 例外内の行番号は SQLJ ソース・ファイル内の行番号と一致します。

-smap=nolyes

SQLJ 変換プログラムが各 SQLJ ソース・ファイルに対してソース・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・ツールによって使用されます。このファイルは、SQLJ ソース・ファイル内の行を、SQLJ 変換プログラムによって生成される Java ソース・ファイル内の行にマップします。このファイルは、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。これがデフォルトです。

yes

SMAP ファイルを生成します。SMAP ファイル名は、*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方式を指定します。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は、オペレーティング・システムのデフォルトの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。**-db2optimize** オプションを指定して SQLJ 変換プログラムを実行する場合、生成後の Java アプリケーションをコンパイルするため、IBM DB2 Driver for JDBC and SQLJ ファイル *db2jcc.jar* が、CLASSPATH 内に入っていないとなりません。

-ser2class

SQLJ 変換プログラムは *.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 DB2 Driver for JDBC and SQLJ のバージョンを表示することを指定します。その情報は、次のようなフォームになります。

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。

-Ccompiler-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** を付けます。例えば以下のようにします。

-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、以下のオプションを除いて、SQLJ 変換プログラムでは使用されません。

-classpath

SQLJ 変換プログラムおよび Java コンパイラーで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 コンパイラーによってクラスまたはインターフェースの定義を検索されるソース・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、ディレクトリー内でのみ、.sqlj および .java ファイルを検索します。

-JVM-option

sqlj コマンドの実行場所である Java 仮想マシン (JVM) に渡すオプションを指定します。このオプションは、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 オプションの間にスペースを入れないください。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 を付けます。例えば以下のようにします。

```
-J-Xmx128m -J-Xmine2M
```

SQLJ-source-file-name

変換対象の 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。SQLJ ソース・ファイル名にはすべて、拡張子 .sqlj が付いていなければなりません。

出力:

各ソース・ファイル *program-name.sqlj* ごとに、SQLJ 変換プログラムは次のようなファイルを作成します。

- 生成されたソース・プログラム

生成されたソース・ファイルには、*program-name.java* という名前が付きます。

- SQLJ 実行可能文節で使用する各接続コンテキスト・クラス用の直列化プロファイル・ファイル

直列化された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は、このコンパイラーによって生成されるクラス・ファイル

例:

```
sqlj -encoding=UTF8 -C-0 MyApp.sqlj
```

関連資料:

- 272 ページの『db2sqljbind - SQLJ プロファイル・バインダー』
- 279 ページの『db2sqljcustomize - SQLJ プロファイル・カスタマイザー』
- 291 ページの『db2sqljprint - SQLJ プロファイル・プリンター』

第 2 章 コマンド行プロセッサ (CLP)

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・ユーティリティー、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。

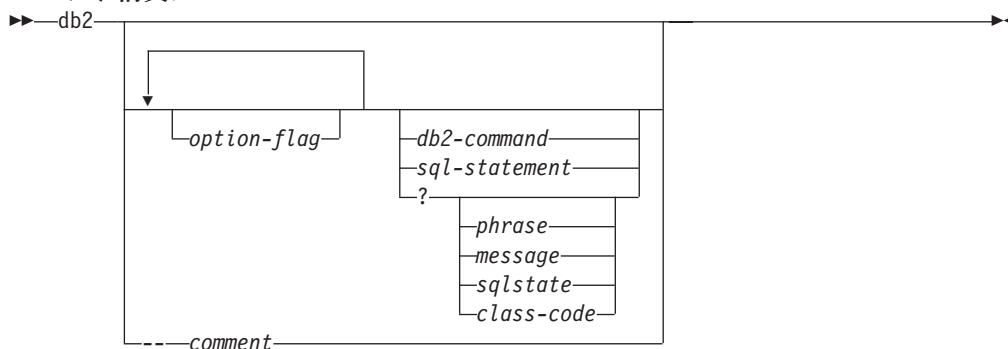
- 対話式入力モード。特徴は **db2 =>** 入力プロンプトです。
- コマンド・モード。各コマンドの前に **db2** を付ける必要があります。
- バッチ・モード。-f ファイル入力オプションを使用します。

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。

関連資料:

- 340 ページの『コマンド行プロセッサのオプション』
- 349 ページの『コマンド行プロセッサの戻りコード』
- 350 ページの『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。

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
-a	コマンド行プロセッサは SQLCA データを表示します。	オフ
-c	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。	オン
-d	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。	オフ
-e{cls}	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。	オフ
-ffilename	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。	オフ
-i	このオプションは、コマンド行プロセッサが、適切な字下げを使って XML データを「整形印刷」するよう指示します。このオプションは、XQuery ステートメントの結果セットにのみ影響を与えます。	オフ
-lfilename	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。	オフ
-m	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。	オフ
-n	単一の区切りトークン内の改行文字を除去します。このオプションを指定しない場合、改行文字はスペースで置換されます。このオプションは、-t オプションと共に使用する必要があります。	オフ
-o	コマンド行プロセッサは、出力データおよびメッセージを標準出力に表示します。	オン
-p	コマンド行プロセッサは、対話型入力モードのときにコマンド行プロセッサ・プロンプトを表示します。	オン

コマンド行プロセッサのオプション

表 3. CLP コマンド・オプション (続き)

オプション・フラグ	説明	デフォルト設定
-q	このオプションは、コマンド行プロセッサが、単一または二重引用符で区切られているストリング内の空白文字および改行文字を保持するよう指示します。オプション q が ON のとき、オプション n は無視されます。	オフ
-filename	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。	オフ
-s	コマンド行プロセッサは、バッチ・ファイルまたは対話式モードでコマンドを実行中にエラーが発生した場合、実行を停止します。	オフ
-t	コマンド行プロセッサはセミコロン (;) をステートメント終了文字として使用します。	オフ
-tdx または -tdxx	このオプションは、コマンド行プロセッサが x または xx をステートメント終了文字 (長さは 1 文字または 2 文字) として定義し、使用するよう指示します。	オフ
-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。	オフ
-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。	オン
-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。	オフ
-zfilename	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。	オフ

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 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。

コマンド行プロセッサのオプション

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、指定ファイルまでのディレクトリー・パスを含む絶対または相対ファイル名です。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。

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。例えば、次のようなものがあります。

```
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。行の最初の非ブランク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):

-lfilename 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、指定ファイルまでのディレクトリー・パスを含む絶対または相対ファイル名です。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。例えば、次のようなものがあります。

```
db2 -tv1 filename
```

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

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

影響を受ける行数の表示オプション (-m):

-m オプションにより、コマンド行プロセッサは、**INSERT**、**DELETE**、**UPDATE**、または **MERGE** で影響を受ける行の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。

コマンド行プロセッサのオプション

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。

- **-rfilename:** 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、**-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):

-rfilename 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、指定ファイルまでのディレクトリー・パスを含む絶対ま

たは相対ファイル名です。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。

-rfilename 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オプション (-t):

-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-)。

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、--#SET TERMINATOR ディレクティ

コマンド行プロセッサのオプション

ブを使用してステートメント終了文字を設定することもできます。例えば、`--#SET TERMINATOR%%` 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。

終了文字かどうか調べられるのが各入力行の最後の非空白文字だけであるため、コマンド行から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):

`-zfilename` 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、指定ファイルまでのディレクトリー・パスを含む絶対または相対ファイル名です。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。

-zfilename 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。

関連資料:

- 339 ページの『db2 - コマンド行プロセッサの呼び出し』
- 349 ページの『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。

コマンド行プロセッサの戻りコード

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。

関連資料:

- 339 ページの『db2 - コマンド行プロセッサの呼び出し』
- 340 ページの『コマンド行プロセッサのオプション』

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。

特定の CLP コマンドと SQL ステートメントを実行するには、サーバー・インスタンスが稼働中であり、データベース接続が存在する必要があります。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。

- SQL ステートメント `DB2 CONNECT TO database` を発行します。
- DB2 レジストリー変数 `DB2DBDFT` に定義されているデフォルト・データベースに暗黙接続を確立します。

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

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

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

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

コマンド行プロセッサのカスタマイズ:

DB2_CLPPROMPT レジストリー変数を使用することによって、対話式入力プロンプトをカスタマイズすることができます。このレジストリー変数は最大長 100 の任意のテキスト・ストリング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 5. 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, @)

コマンド行プロセッサのフィーチャー

- (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 "tyronnem" 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。
(TYRONNEM@DB2, @SAMPLE)
- 許可 ID "horman" 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, HORMAN@SAMPLE)
- 許可 ID "tyronnem" を使用してインスタンス "DB2" にインスタンス接続。許可 ID "horman" 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。
(TYRONNEM@DB2, HORMAN@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になります。

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 6. 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ったらバックエンド・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するのですが、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 文節にある *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。

関連概念:

- 「SQL ガイド」の『DB2 コマンド行プロセッサ (CLP)』

関連資料:

- 340 ページの『コマンド行プロセッサのオプション』
- 349 ページの『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のヘルプ

コマンド行プロセッサからのメッセージ・ヘルプの呼び出し

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

手順:

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

```
? XXXnnnnn
```

ここで、XXXnnnnn は有効なメッセージ ID を表します。

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

関連概念:

- 「メッセージ・リファレンス ボリューム 1」の『メッセージの概要』

関連資料:

- 339 ページの『db2 - コマンド行プロセッサの呼び出し』

コマンド行プロセッサからのコマンド・ヘルプの呼び出し

コマンド・ヘルプは、コマンド行プロセッサのコマンドの構文を説明します。

手順:

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

`? command`

ここで、`command` はキーワードまたはコマンド全体を表します。

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

関連タスク:

- 356 ページの『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』
- オンライン DB2 インフォメーション・センターの『DB2 ツール、ウィンドウ、ウィザード、またはアドバイザーからのヘルプへのアクセス』
- 910 ページの『コマンド行プロセッサから SQL 状態ヘルプを表示する』
- オンライン DB2 インフォメーション・センターの『DB2 インフォメーション・センターの開始』

関連資料:

- 339 ページの『db2 - コマンド行プロセッサの呼び出し』

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

第 3 章 CLP コマンド

この章では、アルファベット順に DB2 コマンドを説明します。これらのコマンドはシステムを対話式に制御するのに使用します。

ディレクトリー・パスのスラッシュ (/) は UNIX ベース・システムに特有のものであり、Windows オペレーティング・システムのディレクトリー・パスの円記号 (¥) に相当します。

DB2 CLP コマンド

以下の表では、機能カテゴリーごとにグループ化された CLP コマンドをリストしています。

表 7. DB2 CLP コマンド

CLP セッション制御
572 ページの『LIST COMMAND OPTIONS 』
840 ページの『UPDATE COMMAND OPTIONS 』
432 ページの『CHANGE ISOLATION LEVEL 』
787 ページの『SET RUNTIME DEGREE 』
815 ページの『TERMINATE 』
676 ページの『QUIT 』
データベース・マネージャー制御
796 ページの『START DATABASE MANAGER 』
806 ページの『STOP DATABASE MANAGER 』
508 ページの『GET DATABASE MANAGER CONFIGURATION 』
732 ページの『RESET DATABASE MANAGER CONFIGURATION 』
847 ページの『UPDATE DATABASE MANAGER CONFIGURATION 』
378 ページの『AUTOCONFIGURE 』
データベース制御
736 ページの『RESTART DATABASE 』
435 ページの『CREATE DATABASE 』
469 ページの『DROP DATABASE 』
634 ページの『MIGRATE DATABASE 』
363 ページの『ACTIVATE DATABASE 』
454 ページの『DEACTIVATE DATABASE 』
670 ページの『QUIESCE 』
825 ページの『UNQUIESCE 』
589 ページの『LIST INDOUBT TRANSACTIONS 』
584 ページの『LIST DRDA INDOUBT TRANSACTIONS 』
502 ページの『GET DATABASE CONFIGURATION 』
730 ページの『RESET DATABASE CONFIGURATION 』

表 7. DB2 CLP コマンド (続き)

844 ページの『UPDATE DATABASE CONFIGURATION 』
378 ページの『AUTOCONFIGURE 』
データベース・ディレクトリー管理
408 ページの『CATALOG DATABASE 』
816 ページの『UNCATALOG DATABASE 』
412 ページの『CATALOG DCS DATABASE 』
818 ページの『UNCATALOG DCS DATABASE 』
430 ページの『CHANGE DATABASE COMMENT 』
573 ページの『LIST DATABASE DIRECTORY 』
582 ページの『LIST DCS DIRECTORY 』
ODBC 管理
425 ページの『CATALOG ODBC DATA SOURCE 』
595 ページの『LIST ODBC DATA SOURCES 』
824 ページの『UNCATALOG ODBC DATA SOURCE 』
496 ページの『GET CLI CONFIGURATION 』
838 ページの『UPDATE CLI CONFIGURATION 』
クライアント/サーバー・ディレクトリー管理
421 ページの『CATALOG LOCAL NODE 』
423 ページの『CATALOG NAMED PIPE NODE 』
426 ページの『CATALOG TCPIP/TCPIP4/TCPIP6 NODE 』
823 ページの『UNCATALOG NODE 』
592 ページの『LIST NODE DIRECTORY 』
ネットワーク・サポート
697 ページの『REGISTER 』
458 ページの『DEREGISTER 』
852 ページの『UPDATE LDAP NODE 』
415 ページの『CATALOG LDAP DATABASE 』
820 ページの『UNCATALOG LDAP DATABASE 』
419 ページの『CATALOG LDAP NODE 』
822 ページの『UNCATALOG LDAP NODE 』
696 ページの『REFRESH LDAP 』
DB2 Administration Server
486 ページの『GET ADMIN CONFIGURATION 』
726 ページの『RESET ADMIN CONFIGURATION 』
827 ページの『UPDATE ADMIN CONFIGURATION 』
451 ページの『CREATE TOOLS CATALOG 』
473 ページの『DROP TOOLS CATALOG 』
リカバリー
373 ページの『ARCHIVE LOG 』
382 ページの『BACKUP DATABASE 』
681 ページの『RECONCILE 』

表 7. DB2 CLP コマンド (続き)

738 ページの『RESTORE DATABASE 』
756 ページの『ROLLFORWARD DATABASE 』
586 ページの『LIST HISTORY 』
665 ページの『PRUNE HISTORY/LOGFILE 』
850 ページの『UPDATE HISTORY 』
560 ページの『INITIALIZE TAPE 』
755 ページの『REWIND TAPE 』
791 ページの『SET TAPE POSITION 』
操作ユーティリティー
484 ページの『FORCE APPLICATION 』
596 ページの『LIST PACKAGES/TABLES 』
716 ページの『REORGCHK 』
705 ページの『REORG INDEXES/TABLE 』
769 ページの『RUNSTATS 』
データベース・モニター
523 ページの『GET MONITOR SWITCHES 』
854 ページの『UPDATE MONITOR SWITCHES 』
513 ページの『GET DATABASE MANAGER MONITOR SWITCHES 』
532 ページの『GET SNAPSHOT 』
734 ページの『RESET MONITOR 』
562 ページの『INSPECT 』
568 ページの『LIST ACTIVE DATABASES 』
570 ページの『LIST APPLICATIONS 』
580 ページの『LIST DCS APPLICATIONS 』
データ・ユーティリティー
478 ページの『EXPORT 』
541 ページの『IMPORT 』
608 ページの『LOAD 』
630 ページの『LOAD QUERY 』
ヘルス・センター
365 ページの『ADD CONTACT 』
367 ページの『ADD CONTACTGROUP 』
467 ページの『DROP CONTACT 』
468 ページの『DROP CONTACTGROUP 』
488 ページの『GET ALERT CONFIGURATION 』
499 ページの『GET CONTACTGROUP 』
500 ページの『GET CONTACTGROUPS 』
501 ページの『GET CONTACTS 』
516 ページの『GET DESCRIPTION FOR HEALTH INDICATOR 』
518 ページの『GET HEALTH NOTIFICATION CONTACT LIST 』
519 ページの『GET HEALTH SNAPSHOT 』

表 7. DB2 CLP コマンド (続き)

526 ページの『GET RECOMMENDATIONS 』
728 ページの『RESET ALERT CONFIGURATION 』
829 ページの『UPDATE ALERT CONFIGURATION 』
842 ページの『UPDATE CONTACT 』
843 ページの『UPDATE CONTACTGROUP 』
849 ページの『UPDATE HEALTH NOTIFICATION CONTACT LIST 』
アプリケーションの準備
638 ページの『PRECOMPILE 』
389 ページの『BIND』
677 ページの『REBIND 』
リモート・サーバー・ユーティリティー
376 ページの『ATTACH 』
466 ページの『DETACH 』
表スペース管理
599 ページの『LIST TABLESPACE CONTAINERS 』
789 ページの『SET TABLESPACE CONTAINERS 』
601 ページの『LIST TABLESPACES 』
673 ページの『QUIESCE TABLESPACES FOR TABLE 』
データベース・パーティション管理
368 ページの『ADD DBPARTITIONNUM 』
471 ページの『DROP DBPARTITIONNUM VERIFY 』
579 ページの『LIST DBPARTITIONNUMS 』
データベース・パーティション・グループ管理
577 ページの『LIST DATABASE PARTITION GROUPS 』
692 ページの『REDISTRIBUTE DATABASE PARTITION GROUP 』
データ・リンク
追加コマンド
460 ページの『DESCRIBE 』
475 ページの『ECHO 』
494 ページの『GET AUTHORIZATIONS 』
498 ページの『GET CONNECTION STATE 』
522 ページの『GET INSTANCE 』
530 ページの『GET ROUTINE 』
538 ページの『HELP 』
636 ページの『PING 』
667 ページの『PUT ROUTINE 』
669 ページの『QUERY CLIENT 』
784 ページの『SET CLIENT 』

ACTIVATE DATABASE

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。

関連概念:

- 「パフォーマンス」の『パフォーマンス・チューニングのためのクイック・スタート・ヒント』

関連資料:

- 806 ページの『STOP DATABASE MANAGER』
- 454 ページの『DEACTIVATE DATABASE』
- 「管理 API リファレンス」の『`sqlc_activate_db` API - データベースの活動化』

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。

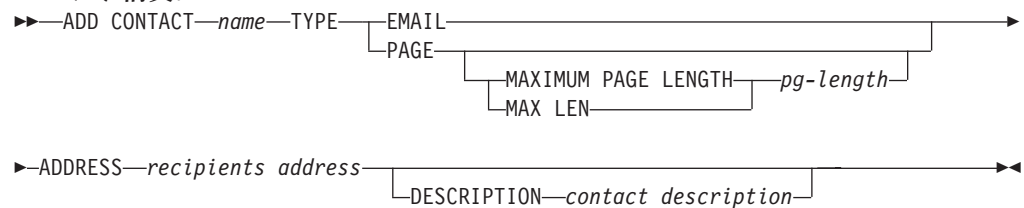
許可:

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:

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、DB2 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必要があります。

DESCRIPTION *contact description*

連絡先のテキスト記述。長さは、最大 128 文字です。

ADD CONTACT

関連タスク:

- 「システム・モニター ガイドおよびリファレンス」の『ヘルス・アラート通知の使用可能化』

関連資料:

- 「管理 API リファレンス」の『db2AddContact API - 通知メッセージを送信できる連絡先の追加』
- 「管理 SQL ルーチンおよびビュー」の『ADD CONTACT コマンド (ADMIN_CMD プロシージャを使用)』

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。

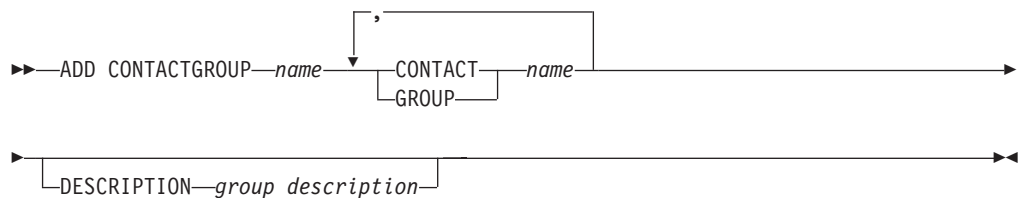
許可:

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:

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。

関連タスク:

- 「システム・モニター ガイドおよびリファレンス」の『ヘルス・アラート通知の使用可能化』

関連資料:

- 「管理 API リファレンス」の『db2AddContactGroup API - 通知メッセージを送信できる連絡先グループの追加』
- 「管理 SQL ルーチンおよびビュー」の『ADD CONTACTGROUP コマンド (ADMIN_CMD プロシージャを使用)』

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 文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、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 文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・

ADD DBPARTITIONNUM

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 **DBPARTITIONNUM** の代わりに **NODE** を使用できます。

関連概念:

- 「管理ガイド: インプリメンテーション」の『自動ストレージ・データベース』

関連資料:

- 796 ページの『START DATABASE MANAGER 』

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:

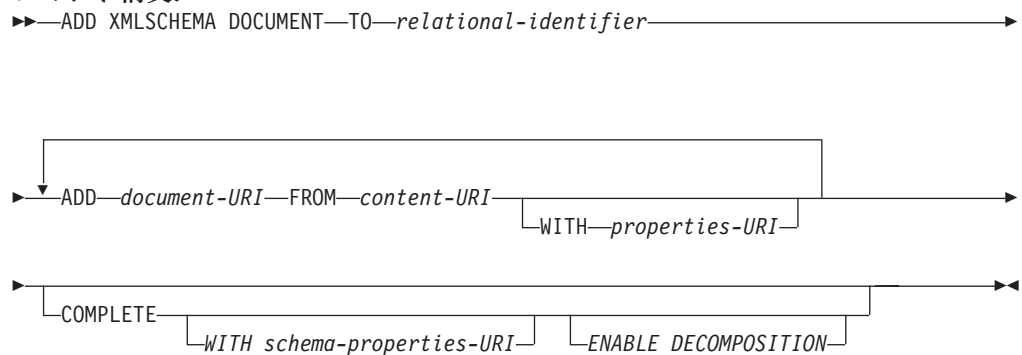
以下の権限が必要です。

- 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。

ADD XMLSCHEMA DOCUMENT

WITH *schema-properties-URI*

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

ENABLE DECOMPOSITION

このスキーマが XML 文書の分解のために使用されることを指定します。

例:

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

関連資料:

- 434 ページの『COMPLETE XMLSCHEMA 』
- 700 ページの『REGISTER XMLSCHEMA 』

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:

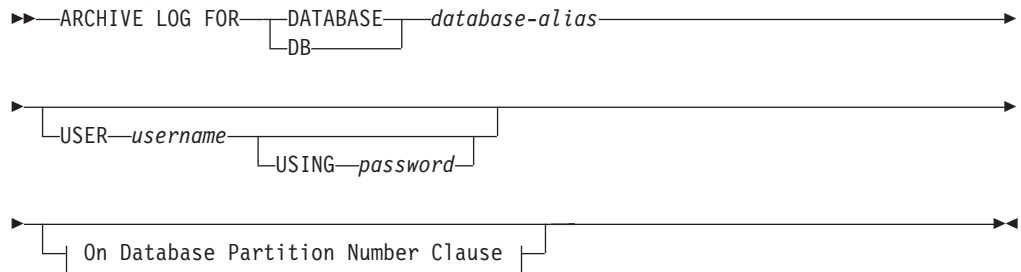
以下のいずれか。

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

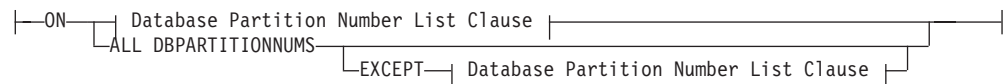
必要な接続:

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文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文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文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利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 DBPARTITIONNUM の代わりに NODE を使用できます。
- キーワード DBPARTITIONNUMS の代わりに NODES を使用できます。

関連資料:

- 「管理 API リファレンス」の『db2ArchiveLog API - アクティブ・ログ・ファイルのアーカイブ』

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指定することを可能にします。このインスタンスは、現在のインスタンス、同じワークステーション上の別のインスタンス、またはリモート・ワークステーションのインスタンスのいずれかになります。

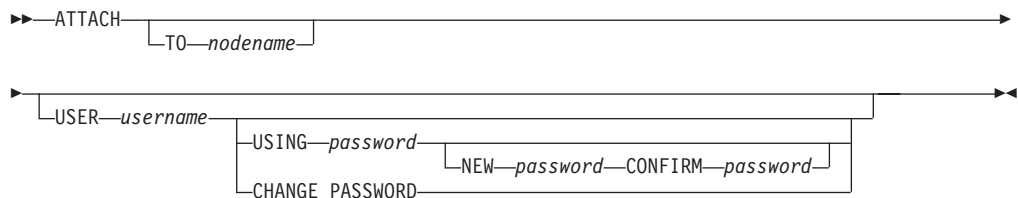
許可:

なし

必要な接続:

なし。このコマンドは、インスタンス接続を確立します。

コマンド構文:



コマンド・パラメーター:

TO nodename

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

USER username

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

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。

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。

関連タスク:

- 「管理ガイド: インプリメンテーション」の『データベース・マネージャーの非デフォルト・インスタンスに対するアタッチおよびデタッチ』

関連資料:

- 466 ページの『DETACH 』
- 「管理 API リファレンス」の『sqleatcp API - インスタンスへのアタッチとパスワードの変更』
- 「管理 API リファレンス」の『sqleatin API - インスタンスへのアタッチ』

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。

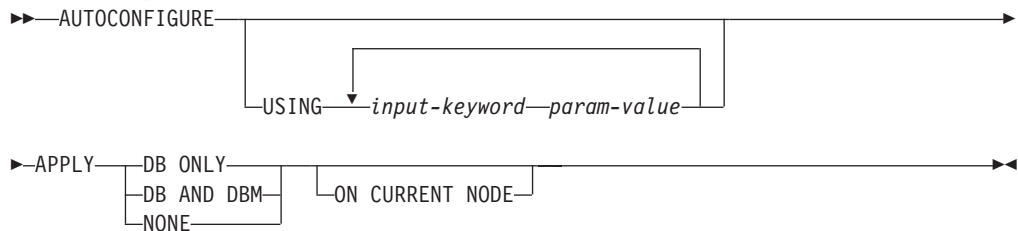
許可:

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 8. 有効な入力キーワードおよびパラメーター値

キーワード	有効値	デフォルト値	説明
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

表 8. 有効な入力キーワードおよびパラメーター値 (続き)

キーワード	有効値	デフォルト値	説明
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
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に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・プールのサイズが変更可能かどうか。

APPLY

DB ONLY

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

DB AND DBM

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

NONE 推奨される変更を表示しますが、適用はしません。

ON CURRENT NODE

デフォルトでは、データベース・パーティション機能 (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・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・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 *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す。
- データベースに対して **AUTOCONFIGURE** コマンドを実行すると、自己調整メモリー・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。

関連概念:

- 「パフォーマンス」の『構成パラメーター』

関連タスク:

- 「管理ガイド: インプリメンテーション」の『構成アドバイザーを使用して構成パラメーターの有効範囲を定義する』
- 「パフォーマンス」の『構成パラメーターによる DB2 の構成』

関連資料:

- 「管理 API リファレンス」の『db2AutoConfig API - 構成アドバイザーへのアクセス』
- 「管理 SQL ルーチンおよびビュー」の『ADMIN_CMD プロシージャ - 管理コマンドの実行』
- 「管理 SQL ルーチンおよびビュー」の『AUTOCONFIGURE コマンド (ADMIN_CMD プロシージャを使用)』

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、『関連概念』セクションにある『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。

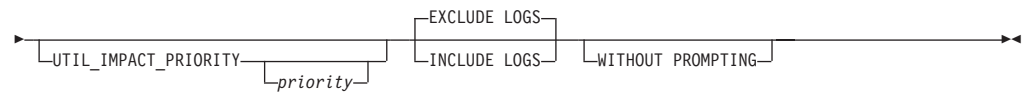
このコマンドは、それが実行されたデータベース・パーティションに対してだけ影響を与えます。

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

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。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。

The diagram illustrates the syntax for the `BACKUP` command, organized into five horizontal sections separated by lines. Each section contains a clause or option with its possible values or sub-clauses, connected by lines and arrows indicating the flow of the command syntax.

- Section 1:** Starts with `BACKUP`, followed by an optional `DATABASE database-alias` clause, and then an optional `USER username` clause. The `USER` clause is followed by an optional `USING password` clause.
- Section 2:** Contains the `TABSPACE (tablespace-name)` clause, which is followed by an optional `ONLINE` clause, an optional `INCREMENTAL` clause, and an optional `DELTA` clause.
- Section 3:** Contains the `USE` clause, which is followed by an optional `TSM XBSA` clause, an optional `OPTIONS "options-string" @file-name` clause, and an optional `OPEN num-sessions SESSIONS` clause. This is followed by a comma and the `TO` clause, which is followed by an optional `dir dev` clause.
- Section 4:** Contains the `LOAD library-name` clause, which is followed by an optional `OPTIONS "options-string" @file-name` clause and an optional `OPEN num-sessions SESSIONS` clause.
- Section 5:** Contains the `WITH num-buffers BUFFERS` clause, an optional `BUFFER buffer-size` clause, and an optional `PARALLELISM n` clause.
- Section 6:** Contains the `COMPRESS` clause, which is followed by an optional `COMPRLIB name` clause, an optional `EXCLUDE` clause, and an optional `COMPROPTS string` clause.



コマンド・パラメーター:

DATABASE 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。

TABLESPACE tablespace-name

バックアップを取る表スペースを指定するときに使用する名前のリスト。

ONLINE

オンライン・バックアップを指定します。デフォルトはオフライン・バックアップです。オンライン・バックアップは、*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。また、DB2 は SMS 表スペース中の LOB データに対する S (共用) ロックを取得します。

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 出力を使用することを指定します。

USE 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。

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。このストリングは、TSM などのベンダー・サポートのライブラリーに、引用符なしで入力された場合とまったく同じように渡されます。このオプションを指定すると、VENDOROPT 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値がオーバーライドされます。

@file-name

バックアップ操作で使用するオプションが、DB2 サーバー上のファイルに含まれていることを指定します。このストリングは、TSM などのベンダー・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。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。Windows オペレーティング・システムの FAT ファイル・システムにバックアップを取るには、ユーザーは 8.3 命名規則に適合するようにしなければなりません。

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。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、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。オフライン・バックアップ操作の実行の場合、このオプションが指定されていてもいなくても、ログは除外されます。

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。

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。

例:

- 以下の例で、データベース WSDb は 0 から 3 までの番号が付けられた 4 つのデータベース・パーティションすべてに定義されています。パス /dev3/backup はすべてのデータベース・パーティションからアクセスできます。データベース・パーティション 0 はカタログ・パーティションであり、これはオフライン・バックアップなので別個にバックアップする必要があります。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 BACKUP DATABASE wsdb TO /dev3/backup'
db2_all ' |<<-0< db2 BACKUP DATABASE wsdb TO /dev3/backup'
```

- 2 番目のコマンドで、db2_all ユーティリティーは同じバックアップ・コマンドを各データベース・パーティションに順々に出します (データベース・パーティション 0 を除く)。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。
- 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。

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・バッファー・サイズを指定する必要があります。パフォーマンスを最適化するには、使用している装置のブロック・サイズの最大限度と等しい値をバッファー・サイズとして使用しなければなりません。

関連概念:

BACKUP DATABASE

- 「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』
- 「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『バックアップとリカバリーの計画の作成』

関連タスク:

- 「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『バックアップの使用』

関連資料:

- 「管理 *SQL* ルーチンおよびビュー」の『BACKUP DATABASE コマンド (ADMIN_CMD プロシージャを使用)』

BIND

バインド・ユーティリティを呼び出し、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ていない場合、および以下のいずれかの場合には、BINDADD 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)

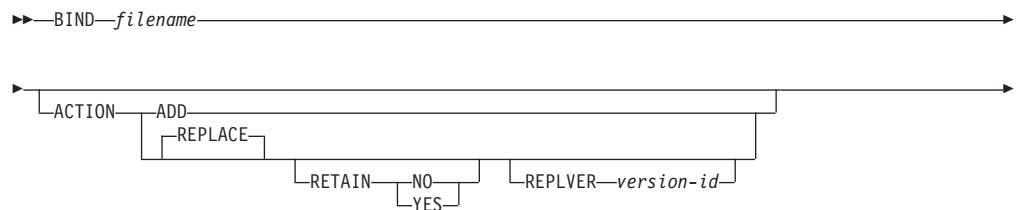
アプリケーション内の静的 SQL ステートメントをコンパイルするために必要な特権もすべて必要です。グループに認可された特権が、静的ステートメントの許可の検査に使用されることはありません。ユーザーに *sysadm* 権限があつてバインドを完了させる明示特権がない場合、データベース・マネージャーは、明示的な *dbadm* 権限を自動的に付与します。

必要な接続:

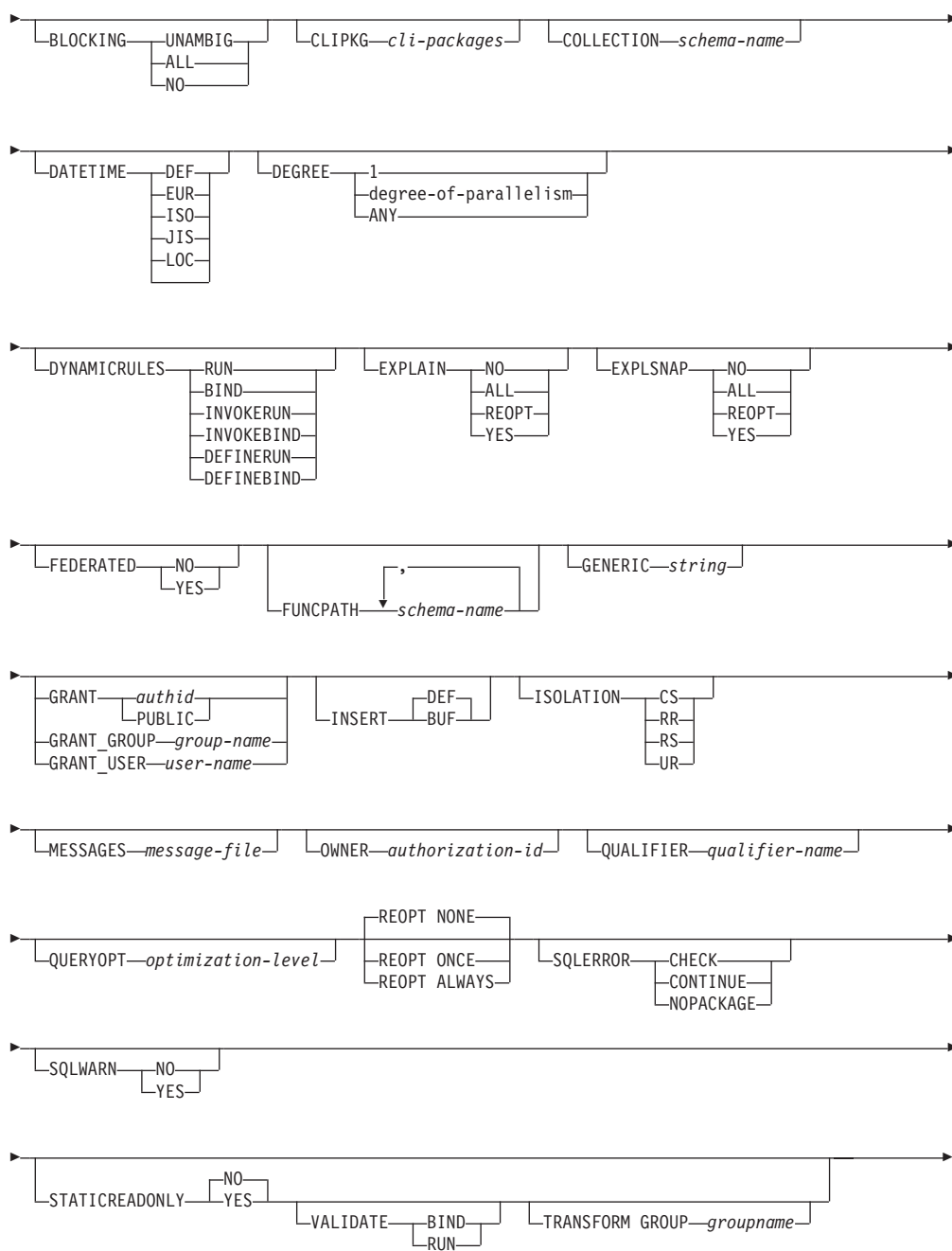
データベース。暗黙接続が可能な場合には、デフォルト・データベースへの接続が確立されます。

コマンド構文:

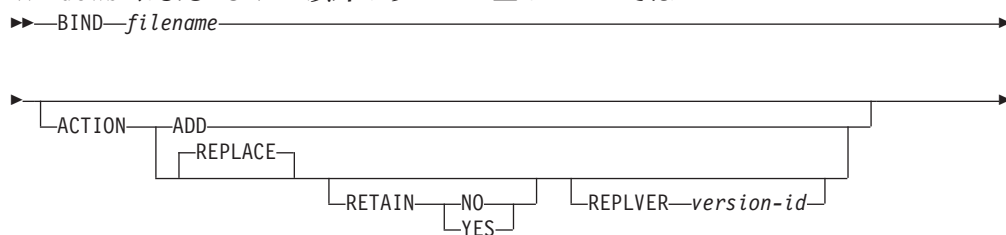
DB2 for Windows および DB2 for UNIX では

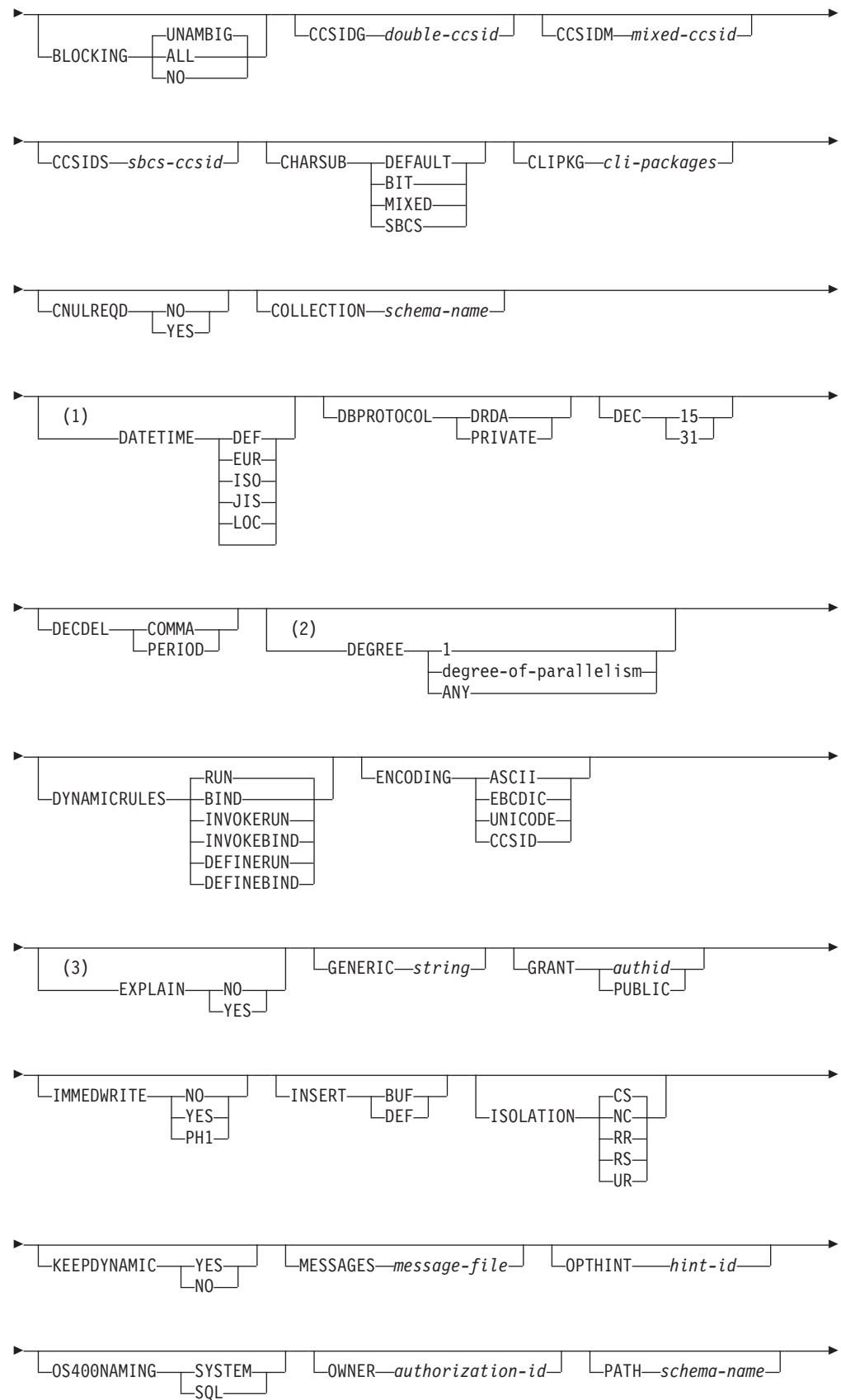


BIND

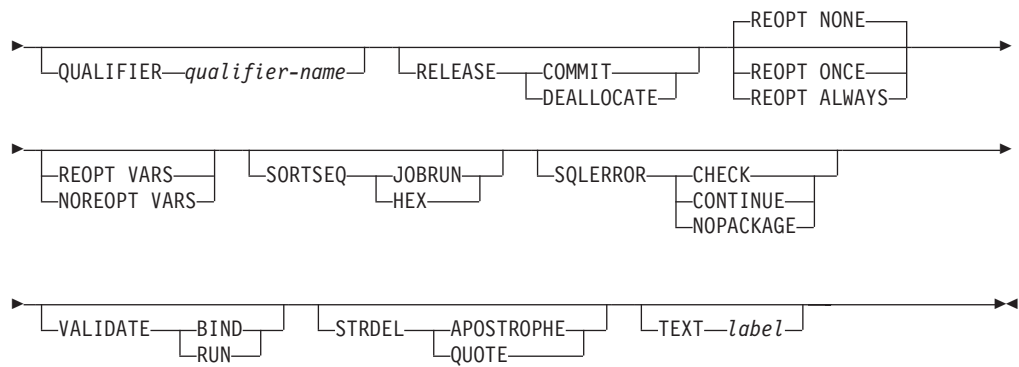


Windows および UNIX 以外のサーバー上の DB2 では





BIND



注:

- 1 サーバーが DATETIME DEF オプションをサポートしない場合、それは DATETIME ISO にマップされます。
- 2 DEGREE オプションは DRDA レベル 2 のアプリケーション・サーバーでしかサポートされていません。
- 3 DRDA は、EXPLAIN オプションが値 YES または NO を持つように定義します。サーバーが EXPLAIN YES オプションをサポートしない場合、この値は EXPLAIN ALL にマップされます。

コマンド・パラメーター:

filename

アプリケーション・プログラムをプリコンパイルしたときに生成されたバインド・ファイル、または複数のバインド・ファイル名を含むリスト・ファイルの名前を指定します。バインド・ファイルの拡張子は .bnd です。また、絶対パス名も指定できます。

リスト・ファイルを指定した場合、その名前の先頭文字は @ 文字でなければなりません。リスト・ファイルには、数行のバインド・ファイル名を含めることができます。同一行にリストするバインド・ファイルはプラス (+) 文字で区切る必要がありますが、各行の先頭ファイルの前や最後のファイルの後に + を挿入することはできません。例えば、

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

これは、次のバインド・ファイルを含むリスト・ファイルです。

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

ACTION

パッケージを追加または置換できるかどうかを示します。

ADD 名前付きパッケージが存在せず、新規パッケージを作成することを指示します。既にパッケージがある場合は、実行停止状態となり、診断エラー・メッセージが戻されます。

REPLACE

既存のパッケージを、パッケージ名および作成者が同じ新規パッケージと置き換えることを指示します。これは ACTION オプションのデフォルト値です。

RETAIN

パッケージを置き換えたときに BIND および EXECUTE 権限が保持されるかどうかを指示します。パッケージの所有権を変更した場合、新規所有者は前のパッケージ所有者に BIND 権限と EXECUTE 権限を付与します。

NO パッケージを置き換えたとき、BIND および EXECUTE 権限を保持しません。この値は DB2 ではサポートされていません。

YES パッケージを置き換えたとき、BIND および EXECUTE 権限を保持します。これはデフォルト値です。

REPLVER version-id

特定のバージョンのパッケージを置き換えます。バージョン ID は、どのバージョンのパッケージを置き換えるのかを指定するものです。指定されたバージョンが存在しない場合には、エラーが戻されます。REPLACE の REPLVER オプションが指定されていない場合、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ク化のタイプを指定します。

ALL 次のカーソルをブロック化することを指定します。

- 読み取り専用カーソル。
- FOR UPDATE OF と指定されていないカーソル。

未確定カーソルは、読み取り専用として扱われます。

NO どのカーソルもブロック化しないことを指定します。あいまいなカーソルは、更新可能として扱われます。

UNAMBIG

次のカーソルをブロック化することを指定します。

- 読み取り専用カーソル。
- FOR UPDATE OF と指定されていないカーソル。

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 文節は使用しない) を指定する整数。このオプションは、DB2 Database for

Linux, UNIX, and Windows ではサポートされていません。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 文節は使用しない) を指定する整数。このオプションは、DB2 Database for Linux, UNIX, and Windows ではサポートされていません。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 sbcs-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 文節は使用しない) を指定する整数。このオプションは、DB2 Database for Linux, UNIX, and Windows ではサポートされていません。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。DB2 for Windows および DB2 for UNIX 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 ~ 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。DB2 for Windows および DB2 for UNIX 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tring・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 30 文字の収集 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。DB2 for Windows および DB2 for UNIX 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。DB2 for Windows および DB2 for UNIX 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 ～ 32 767 です。

ANY ステートメントの実行時にデータベース・マネージャーで判別した程度で並列処理を行うよう指定します。

DYNAMICRULES

許可 ID に使用される値の初期設定、および非修飾オブジェクト参照の暗黙的な修飾の、実行時の動的 SQL に適用される規則を定義します。

RUN パッケージを実行するユーザーの許可 ID が動的 SQL ステートメントの許可検査に使用されるように指定します。許可 ID は、動的 SQL ステートメント内の非修飾オブジェクト参照を暗黙的に修飾するためのデフォルトのパッケージ修飾子としても使用されます。これはデフォルト値です。

BIND 許可および修飾の静的 SQL に適用されるすべての規則が、実行時に使用されるように指定します。つまり、パッケージ所有者の許可 ID が動的 SQL の許可検査に使用され、デフォルトのパッケージ修飾子が動的 SQL ステートメント内の非修飾オブジェクト参照の暗黙的な修飾に使用されます。

DEFINERUN

パッケージがルーチン・コンテキスト内で使用される場合、ルーチン定義者の許可 ID が許可検査およびルーチン内の動的 SQL ステートメント内の非修飾オブジェクト参照の暗黙的な修飾に使用されます。

パッケージがスタンドアロン・アプリケーションとして使用される場合、動的 SQL ステートメントはパッケージが DYNAMICRULES RUN 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。これが行われえない場合、パッケージ内の追加バインド・ステートメントはランタイム・エラーを生じます (SQLSTATE 42985)。

REOPT

再最適化可能な追加バインド SQL ステートメントの Explain 情報が、実行時に各 Explain 表に入れます。さらに、CURRENT EXPLAIN MODE レジスターが NO に設定されていても、Explain 情報は実行時に再最適化可能な動的 SQL ステートメント用に集められます。

パッケージがルーチンに使用される場合、そのルーチンは MODIFIES SQL DATA として定義されなければなりません。そうしなければ、パッケージ内の追加バインドおよび動的ステートメントはランタイム・エラーを生じます (SQLSTATE 42985)。

ALL 適格な静的 SQL ステートメントの Explain 情報が、prep/bind 時に各 Explain 表に入れます。適格な追加バインド SQL ステートメントの Explain 情報が、実行時に各 Explain 表に入れます。さらに、CURRENT EXPLAIN MODE レジスターが NO に設定されていても、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ます。

パッケージがルーチンに使用されるもので、パッケージに追加バインド・ステートメントが含まれる場合、そのルーチンは MODIFIES SQL DATA として定義されなければなりません。これが行われない場合、パッケージ内の追加バインド・ステートメントはランタイム・エラーを生じます (SQLSTATE 42985)。

REOPT

再最適化可能な追加バインド SQL ステートメントの Explain スナップショット情報が、実行時に各 Explain 表に入れます。さらに、CURRENT EXPLAIN SNAPSHOT レジスターが NO に設定されていても、Explain スナップショット情報は実行時に再最適化可能な動的 SQL ステートメント用に集められます。

パッケージがルーチンに使用される場合、そのルーチンは MODIFIES SQL DATA として定義されなければなりません。そうしなければ、パッケージ内の追加バインドおよび動的ステートメントはランタイム・エラーを生じます (SQLSTATE 42985)。

ALL 適格な各静的 SQL ステートメントの Explain スナップショット

が、prep/bind 時に Explain 表内に入れます。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。

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、254 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 1023 バイトです。

GRANT

authid 指定したユーザー名またはグループ ID に EXECUTE 特権と BIND 特権を付与します。

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 ID 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、GRANT_GROUP、および GRANT_USER 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。

INSERT

DB2 Enterprise Server Edition サーバーヘブ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。 DB2 for Windows および DB2 for UNIX 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

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。

OS400NAMING

DB2 UDB for iSeries 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。DB2 UDB for iSeries だけによってサポートされています。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 iSeries 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、iSeries システムの命名規則を使用する特定の SQL ステートメントに関して実行時に構文エラーを報告することがあります。例えば、OS400NAMING SYSTEM オプションを指定してプリコンパイルまたはバインドされているかどうかには関係なく、iSeries 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 30 文字の許可 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。

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 30 文字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。DB2 for Windows および DB2 for UNIX 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

iSeries システムで使用するソート・シーケンス表を指定します。DB2 UDB for iSeries だけによってサポートされています。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 iSeries 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、および特殊レジスターの値を使用して実行時にアクセス・パスを判別するようにするかどうかを指定します。有効な値は以下のとおりです。

NONE ホスト変数、パラメーター・マーカ、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、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、または特殊レジスターの既知の値によって、SQL ステートメントのアクセス・パスが必ずコンパイルおよび再最適化されます。

REOPT / NOREOPT VARS

これらのオプションは、REOPT ALWAYS および REOPT NONE によって置き換えられましたが、バックレベル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、パラメーター・マーカ、および特殊レジスターの値を使用して実行時にアクセス・パスを判別するようにするかどうかを指定します。サポートしているのは DB2 for OS/390 だけです。サポートされているオプション値のリストについては、DB2 for OS/390 の資料を参照してください。

SQLWARN

動的 SQL ステートメントのコンパイル時 (PREPARE または EXECUTE IMMEDIATE を通して)、または記述処理時 (PREPARE...INTO または DESCRIBE を通して) に警告を戻すかどうかを指示します。

NO SQL コンパイラから警告を戻しません。

YES SQL コンパイラ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** 文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。DB2 for Windows および DB2 for UNIX 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、デフォルト値はブランクです。DB2 for Windows および DB2 for UNIX 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 文字です。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** *yyyyy*, **SQLSTATE** *xxxxx*: 変換が必要ですが、静的変換グループが選択されていません。
- **SQLCODE** *yyyyy*, **SQLSTATE** *xxxxx*: 選択された変換グループには、交換するデータ・タイプに必要な変換が含まれていません (入力変数用の **TO SQL**、出力変数用の **FROM SQL**)。
- **SQLCODE** *yyyyy*, **SQLSTATE** *xxxxx*: **FROM SQL** 変換の結果タイプは、出力変数のタイプと互換性がありません。または、**TO SQL** 変換のパラメーター・タイプは、入力変数のタイプと互換性がありません。

これらのエラー・メッセージで、*yyyyy* は SQL エラー・コードによって置き換えられ、*xxxxx* は SQL 状態コードによって置き換えられます。

VALIDATE

データベース・マネージャーが、許可エラーとエラー未検出のオブジェクトをいつ検査するかを判別します。この妥当性検査には、パッケージ所有者の許可 ID が使用されます。

BIND プリコンパイル/バインド時に妥当性検査が実行されます。オブジェクトが 1 つもない場合、または権限がまったく保持されていない場合、エラー・メッセージが作成されます。 **SQLERROR** **CONTINUE** を指定した場合、エラー・メッセージにかかわらずパッケージ/バインド・ファイルは作成されますが、エラーのあるステートメントは実行できません。

RUN バインド時に妥当性検査が行われます。すべてのオブジェクトが存在し、全権限が保持されていれば、それ以上実行しても検査は行われません。

プリコンパイル/バインド時にオブジェクトが 1 つもない場合、または権限がまったく保持されていない場合、 **SQLERROR** **CONTINUE** オプションの設定とは無関係に警告メッセージが作成されて、パッケージは正常にバインドされます。ただし、プリコンパイル/バインド処理時に SQL ステートメントの権限検査と存在検査に障害が生じた場合、実行時に再実行される可能性があります。

例:

以下は、`myapp.bnd` (`myapp.sqc` プログラムのプリコンパイル時に生成されるバインド・ファイル) と接続が確立しているデータベースをバインドする例です。

```
db2 bind myapp.bnd
```

バインド処理で生じたすべてのメッセージは、標準出力に送信されます。

使用上の注意:

REOPT オプションに ONCE または ALWAYS 値を指定してパッケージをバインディングすると、静的および動的ステートメントのコンパイルおよびパフォーマンスが変化することがあります。

バインド処理は、アプリケーション・プログラムのソース・ファイルのプリコンパイル処理の一部として実行することもできますし、別のプロセスとして後から実行することもできます。バインドを別のプロセスとして実行するときは BIND を使用します。

パッケージを作成するのに使われた名前はバインド・ファイルに保管されます。その名前は、その名前が生成されたソース・ファイルの名前を基にして付けられます(既存のパスや拡張子は取り除かれます)。例えば、myapp.sql というプリコンパイル・ソース・ファイルは、myapp.bnd というデフォルト・バインド・ファイルとデフォルト・パッケージ名の MYAPP を生成します。ただし、**BINDFILE** および **PACKAGE** オプションを使用すれば、プリコンパイル時にバインド・ファイル名とパッケージ名を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 か 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に無視されます。

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。

関連概念:

- 「*SQL* リファレンス 第 1 巻」の『分離レベル』
- 「*SQL* ガイド」の『動的 SQL における許可に関する考慮事項』
- 「組み込み *SQL* アプリケーションの開発」の『動的 SQL における DYNAMICRULES BIND オプションの影響』
- 「組み込み *SQL* アプリケーションの開発」の『BIND コマンドの REOPT オプションを使用した場合のパフォーマンスの改善』

関連資料:

- 638 ページの『PRECOMPILE 』
- 「*SQL* リファレンス 第 2 巻」の『SET CURRENT QUERY OPTIMIZATION ステートメント』
- 「*SQL* リファレンス 第 1 巻」の『日付/時刻の値』
- 「*SQL* リファレンス 第 1 巻」の『特殊レジスター』
- 「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」の『DB2 CLI のバインド・ファイルおよびパッケージ名』
- 「管理 API リファレンス」の『sqlabndx API - アプリケーション・プログラムのバインドによるパッケージの作成』

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・ノードのどちらかに位置付けることができます。

有効範囲:

パーティション・データベース環境では、ローカル・データベースをシステム・データベース・ディレクトリーにカタログするときに、データベースが常駐するサーバー上のデータベース・パーティションでこのコマンドを発行する必要があります。

許可:

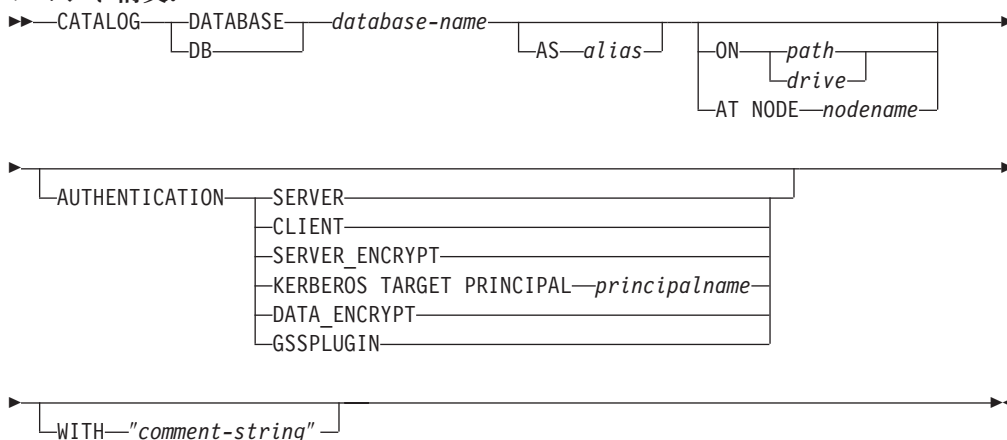
以下のいずれか。

- *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ノード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ノードで行われるということを指定します。

CLIENT

認証が、アプリケーションを呼び出すノードで行われるということを指定します。

SERVER_ENCRYPT

認証が、ターゲット・データベースを含むノードで行われるということ、および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、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。認証が Kerberos の場合は、SECURITY=NONE のみサポートされています。

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ノードで行われ、接続がデータ暗号化を使用しなければならないことを指定します。

GSSPLUGIN

外部 GSS API ベースのプラグイン・セキュリティ機構を使って認証が行われることを指定します。認証が GSSPLUGIN の場合は、SECURITY=NONE のみサポートされています。

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

CATALOG DATABASE

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・ノード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。

パス名もノード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 *dfidbpath* 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ノードのデータベースは、間接項目としてカタログされます。その他のノードのデータベースは、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。

関連タスク:

- 「管理ガイド: インプリメンテーション」の『データベースのカタログ作成』

関連資料:

- 「管理 API リファレンス」の『sqlcadb API - システム・データベース・ディレクトリーへのデータベースのカタログ』
- 508 ページの『GET DATABASE MANAGER CONFIGURATION 』
- 573 ページの『LIST DATABASE DIRECTORY 』
- 815 ページの『TERMINATE 』
- 816 ページの『UNCATALOG DATABASE 』

CATALOG DCS DATABASE

リモート・ホストまたは iSeries 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:

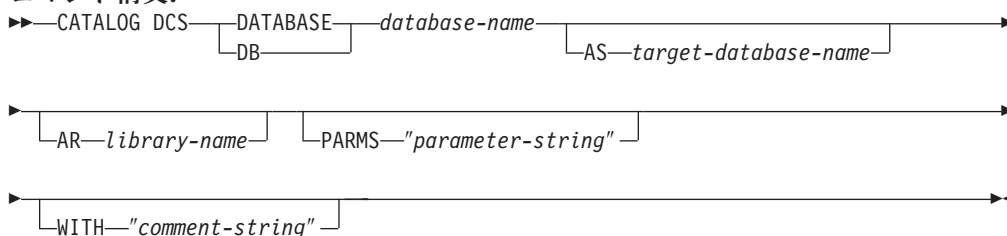
以下のいずれか。

- *sysadm*
- *sysctrl*

必要な接続:

なし

コマンド構文:



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・ノード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 iSeries 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。パラメーター・ストリングの詳細は、『**関連概念**』のセクションに示されている『DCS ディレクトリーの値』のトピックを参照してください。

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。
- AS/400 および iSeries コンピューター上の iSeries 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 DCS DATABASE

関連概念:

- 「*DB2 Connect ユーザーズ・ガイド*」の『DCS ディレクトリーの値』

関連資料:

- 「*管理 API リファレンス*」の『sqlegdad API - データベース接続サービス (DCS) ディレクトリーへのデータベースのカatalog』
- 508 ページの『GET DATABASE MANAGER CONFIGURATION 』
- 815 ページの『TERMINATE 』
- 818 ページの『UNCATALOG DCS DATABASE 』
- 582 ページの『LIST DCS DIRECTORY 』

CATALOG LDAP DATABASE

Lightweight Directory Access Protocol (LDAP) でデータベースを登録するのに使
います。

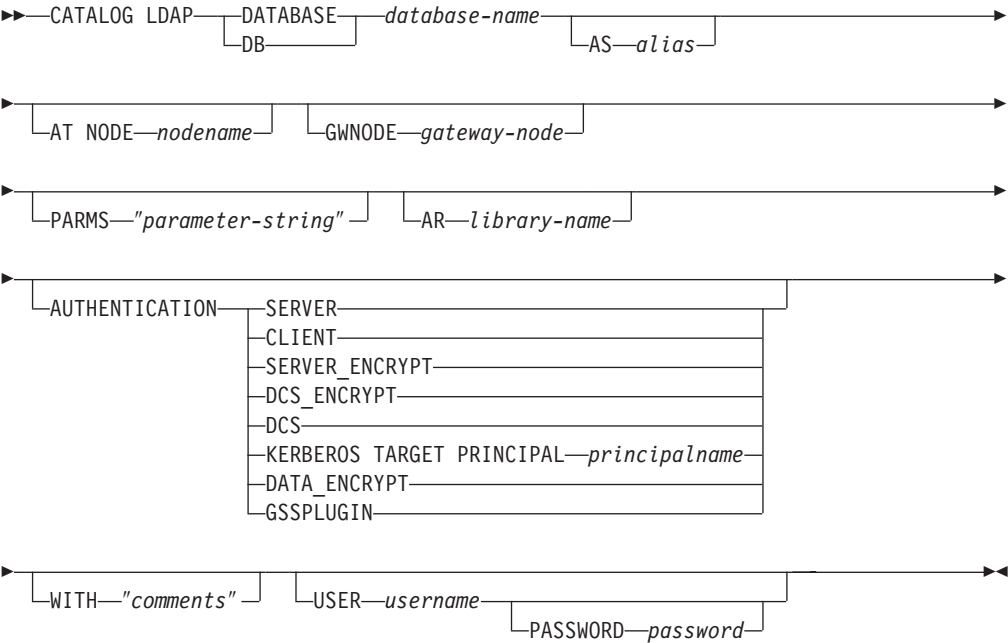
許可:

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りモート・データベースにアクセスするとき、ロードおよび使用されるアプリケーション・リクエスター・ライブラリーの名前を指定します。

DB2 Connect AR を使用する場合は、ライブラリー名を指定しないでください。デフォルト値で DB2 Connect を呼び出します。

DB2 Connect を使用しない場合は、AR のライブラリー名を指定します。そして、そのライブラリーをデータベース・マネージャー・ライブラリーと同じパスに置いてください。Windows オペレーティング・システムでは、そのパスは `drive:%sqllib%dll` です。UNIX 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。

AUTHENTICATION

認証レベルを指定します。有効な値は以下のとおりです。

SERVER

認証が、ターゲット・データベースを含むノードで行われるということを指定します。

CLIENT

認証が、アプリケーションの呼び出し元であるノードで行われるということを指定します。

SERVER_ENCRYPT

認証が、ターゲット・データベースを含むノードで行われるということ、および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、パスワードはターゲットで暗号化解除されます。

DCS_ENCRYPT

認証が、ターゲット・データベースを含むノードで行われるということを指定します。ただし、DB2 Connect の使用時は除きます。この場合、認証は DRDA アプリケーション・サーバー (AS) で行われます。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、パスワードは送信元で暗号化され、ターゲットで暗号化解除されます。

DCS 認証が、ターゲット・データベースを含むノードで行われるということを指定します。ただし、DB2 Connect の使用時は除きます。この場合、認証は DRDA アプリケーション・サーバー (AS) で行われます。

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。認証が Kerberos の場合は、SECURITY=NONE のみサポートされています。

TARGET PRINCIPAL *principalname*

ターゲット・サーバー用の完全修飾 Kerberos プリンシパル

名。つまり、DB2 サーバー・サービスのログオン・アカウントで、 `userid@xxx.xxx.xxx.com` または `domain#userid` の形式です。

DATA_ENCRYPT

認証が、ターゲット・データベースを含むノードで行われ、接続がデータ暗号化を使用しなければならないことを指定します。

GSSPLUGIN

外部 GSS API ベースのプラグイン・セキュリティ機構を使って認証が行われることを指定します。認証が GSSPLUGIN の場合は、SECURITY=NONE のみサポートされています。

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、ホストまたは iSeries データベース・サーバーに常駐します。この場合、管理者は LDAP でデータベースを登録し、GWNODE パラメーターを介してゲートウェイ・ノードを指定することができます。
- CREATE DATABASE IN LDAP の実行中、データベース名が既に LDAP に存在する場合。データベースは依然としてローカル・マシン上で作成されています。

(ローカル・アプリケーションからアクセス可能) が、LDAP にある既存の項目には、新規データベースの内容は反映されません。この場合、管理者は次のことを行えます。

- LDAP の既存のデータベース項目を除去し、手動で新しいデータベースを LDAP に登録する。
- 異なる別名を使って LDAP で新しいデータベースを登録する。

関連概念:

- 「管理ガイド: インプリメンテーション」の『Lightweight Directory Access Protocol (LDAP) 』

関連タスク:

- 「管理ガイド: インプリメンテーション」の『LDAP ディレクトリーへのデータベースの登録』

関連資料:

- 「管理 API リファレンス」の『db2LdapCatalogDatabase API - LDAP サーバーへのデータベースの登録』
- 697 ページの『REGISTER 』
- 820 ページの『UNCATALOG LDAP DATABASE 』
- 419 ページの『CATALOG LDAP NODE 』
- 822 ページの『UNCATALOG LDAP NODE 』
- 149 ページの『db2ldcfg - LDAP 環境の構成』

CATALOG LDAP NODE

Lightweight Directory Access Protocol (LDAP) で新しいノード項目をカタログします。

許可:

なし

必要な接続:

なし

コマンド構文:

```

▶—CATALOG LDAP—NODE—nodename—AS—nodealias—————▶
|
|—USER—username—|
|                  |—PASSWORD—password—|
|                  |
|—————|—————▶
  
```

コマンド・パラメーター:

NODE*nodename*

DB2 サーバーの LDAP ノード名を指定します。

AS *nodealias*

LDAP ノード項目に新しい別名を指定します。

USER *username*

ユーザーの LDAP 識別名 (DN) を指定します。LDAP ユーザー DN には、LDAP ディレクトリーでオブジェクトを作成するための十分な権限が必要です。ユーザーの LDAP DN が指定されない場合、現行ログオン・ユーザーの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。

関連概念:

- 「管理ガイド: インプリメンテーション」の『Lightweight Directory Access Protocol (LDAP) 』

関連タスク:

- 「管理ガイド: インプリメンテーション」の『インストール後の DB2 サーバーの登録』

関連資料:

- 「管理 API リファレンス」の『db2LdapCatalogNode API - LDAP サーバーにあるノード名の別名の指定』
- 415 ページの『CATALOG LDAP DATABASE 』

CATALOG LDAP NODE

- 820 ページの『UNCATALOG LDAP DATABASE 』
- 822 ページの『UNCATALOG LDAP NODE 』

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:

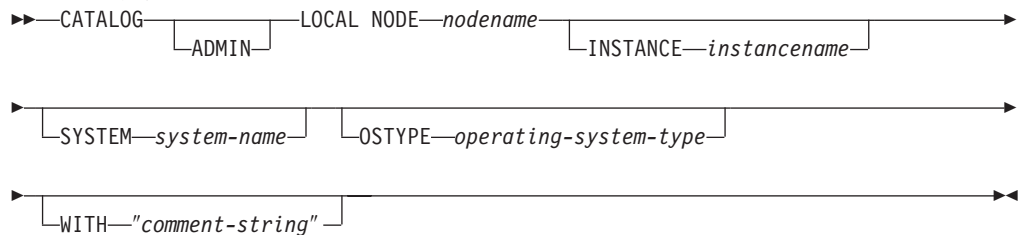
以下のいずれか。

- *sysadm*
- *sysctrl*

必要な接続:

なし

コマンド構文:



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。

NODE *nodename*

カタログするノードのローカル別名。これはユーザーのワークステーション上の任意の名前です。覚えやすい、意味のある名前にしてください。この名前は、データベース・マネージャーの命名規則に従う必要があります。

INSTANCE *instancetypename*

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

WITH “comment-string”

ノード・ディレクトリー内のノード項目について記述します。ノードの説明に役立つどんなコメントでも入力できます。最大長は 30 文字です。復帰文字または改行文字は使用できません。コメント・テキストは、単一または二重の引用符で囲む必要があります。

例:

CATALOG LOCAL NODE

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。

関連資料:

- 「管理 API リファレンス」の『sqlectnd API - ノード・ディレクトリーへの項目のカタログ』
- 508 ページの『GET DATABASE MANAGER CONFIGURATION 』
- 815 ページの『TERMINATE 』

CATALOG NAMED PIPE NODE

ノード・ディレクトリーに名前付きパイプ・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、この名前付きパイプを使用します。

このコマンドは Windows のみで使用できます。

許可:

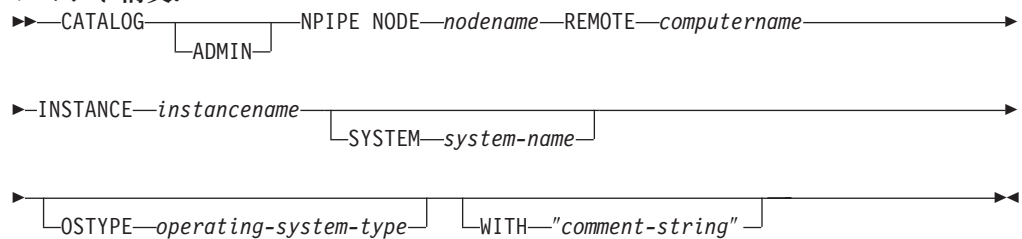
以下のいずれか。

- *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。

NODE *nodename*

カタログするノードのローカル別名。これはユーザーのワークステーション上の任意の名前です。覚えやすい、意味のある名前にしてください。この名前は、データベース・マネージャーの命名規則に従う必要があります。

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、リモート名前付きパイプ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。

WITH "comment-string"

ノード・ディレクトリー内のノード項目について記述します。ノードの説明

CATALOG NAMED PIPE NODE

に役立つどんなコメントでも入力できます。最大長は 30 文字です。復帰文字または改行文字は使用できません。コメント・テキストは、単一または二重の引用符で囲む必要があります。

例:

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

使用上の注意:

データベース・マネージャーは、最初のノードがカタログされたとき (つまり、CATALOG...NODE コマンドが最初に発行されたとき) にノード・ディレクトリーを作成します。Windows クライアントでは、そのクライアントをインストールしたインスタンス・サブディレクトリーに、ノード・ディレクトリーを保管して維持します。また、AIX クライアントでは、DB2 インストール・ディレクトリーにノード・ディレクトリーを作成します。

ローカル・ノード・ディレクトリーの内容をリストする場合は、LIST NODE DIRECTORY コマンドを使用してください。

ディレクトリーをキャッシュできる場合 (GET DATABASE MANAGER CONFIGURATION コマンドの構成パラメーター *dir_cache* を参照)、データベース、ノード、および DCS のディレクトリー・ファイルはメモリーにキャッシュされます。アプリケーションのディレクトリー・キャッシュは、最初のディレクトリー検索時に作成されます。キャッシュはアプリケーションがディレクトリー・ファイルのいずれかを修正したとき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。

関連資料:

- 「管理 API リファレンス」の『sqlectnd API - ノード・ディレクトリーへの項目のカタログ』
- 508 ページの『GET DATABASE MANAGER CONFIGURATION 』
- 592 ページの『LIST NODE DIRECTORY 』
- 815 ページの『TERMINATE 』

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:

```

▶▶ CATALOG [USER | SYSTEM] ODBC DATA SOURCE data-source-name ▶▶
  
```

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。最大長は 32 文字です。

関連資料:

- 595 ページの『LIST ODBC DATA SOURCES 』
- 824 ページの『UNCATALOG ODBC DATA SOURCE 』

426 コマンド・リファレンス

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 が使用されていて、かつ 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 DB2 クライアント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。

REMOTE_INSTANCE *instance-name*

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。

CATALOG TCPIP/TCPIP4/TCPIP6 NODE

OSTYPE *operating-system-type*

サーバー・マシンのオペレーティング・システムのタイプを指定します。有効な値は、AIX、WIN、HPUX、SUN、OS390、OS400、VM、VSE、および LINUX です。

WITH “comment-string”

ノード・ディレクトリー内のノード項目について記述します。ノードの説明に役立つどんなコメントでも入力できます。最大長は 30 文字です。復帰文字または改行文字は使用できません。コメント・テキストは、単一または二重の引用符で囲む必要があります。

例:

CATALOG TCPIP NODE コマンドを使用してホスト名を指定するには、次のようにコマンドを発行します。

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

CATALOG TCPIP4 NODE コマンドを使用して IPv4 アドレスを指定するには、次のようにコマンドを発行します。

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

この例では IPv4 アドレスを指定しています。 **CATALOG TCPIP4 NODE** コマンドでは、IPv6 アドレスを指定しないようにしてください。そのようにしてもカタログで障害が発生するわけではありませんが、カタログにおいて無効なアドレスが指定されたため、それ以降のアタッチまたは接続は失敗します。

CATALOG TCPIP6 NODE コマンドを使用して IPv6 アドレスを指定するには、次のようにコマンドを発行します。

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

この例では、server について IPv6 アドレスとポート番号を指定しています。

CATALOG TCPIP4 NODE コマンドでは、IPv6 アドレスを指定しないようにしてください。そのようにしてもカタログで障害が発生するわけではありませんが、カタログにおいて無効なアドレスが指定されたため、それ以降のアタッチまたは接続は失敗します。

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。

関連資料:

- 「管理 API リファレンス」の『sqlectnd API - ノード・ディレクトリーへの項目のカatalog』
- 508 ページの『GET DATABASE MANAGER CONFIGURATION 』
- 592 ページの『LIST NODE DIRECTORY 』
- 815 ページの『TERMINATE 』

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。

関連資料:

- 「管理 API リファレンス」の『sqledcgd API - 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・コメントの変更』
- 435 ページの『CREATE DATABASE 』

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。

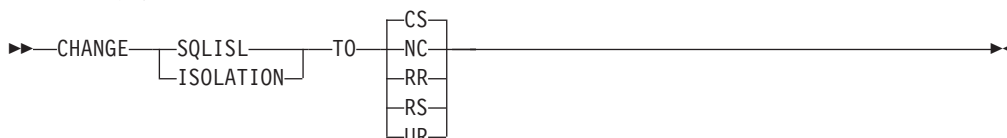
許可:

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保全性を維持します。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。

関連概念:

- 「SQL リファレンス 第 1 巻」の『分離レベル』

関連資料:

- 784 ページの『SET CLIENT 』
- 669 ページの『QUERY CLIENT 』
- 「管理 API リファレンス」の『分離レベルの変更』

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.P0schema 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。

関連資料:

- 371 ページの『ADD XMLSCHEMA DOCUMENT 』
- 700 ページの『REGISTER XMLSCHEMA 』

CREATE DATABASE

CREATE DATABASE コマンドは、オプションのユーザー定義照合順序を使って新しいデータベースを初期化し、3 つの初期表スペースを作成し、システム表を作成し、リカバリー・ログ・ファイルを割り振ります。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。

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* レジストリー変数の説明を参照してください。自動 **Runstats** を無効にするの詳細は、データベース構成パラメーター *auto_runstats* の説明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、適応自己調整メモリーもデフォルトで有効になっています。デフォルトで適応自己調整メモリーが無効になるようにすることに関する詳細は、データベース構成パラメーター *self_tuning_mem* の説明を参照してください (『関連資料』のセクションを参照)。複数パーティション・データベースの場合、適応自己調整メモリーはデフォルトで無効になっています。

XML フィーチャーを利用するには、このコマンドの **USING CODESET** オプションを使用して、コード・セットを UTF-8 に設定する必要があります。DB2 データベース・システムの将来のリリースでは、データベース作成時のデフォルト・コード・セットが変更されて UTF-8 になる予定です。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リーを指定する必要があります。

このコマンドはクライアントでは無効です。

有効範囲:

パーティション・データベース環境では、このコマンドは、*db2nodes.cfg* ファイルにリストされているすべてのデータベース・パーティションに影響を与えます。

このコマンドを発行したデータベース・パーティションは、新規データベースのカタログ・データベース・パーティションになります。

許可:

以下のいずれかが必要です。

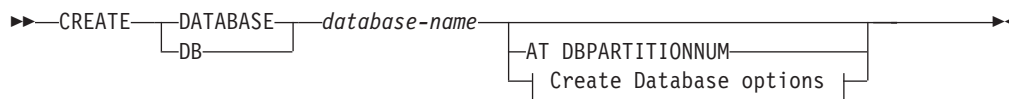
- *sysadm*
- *sysctrl*

必要な接続:

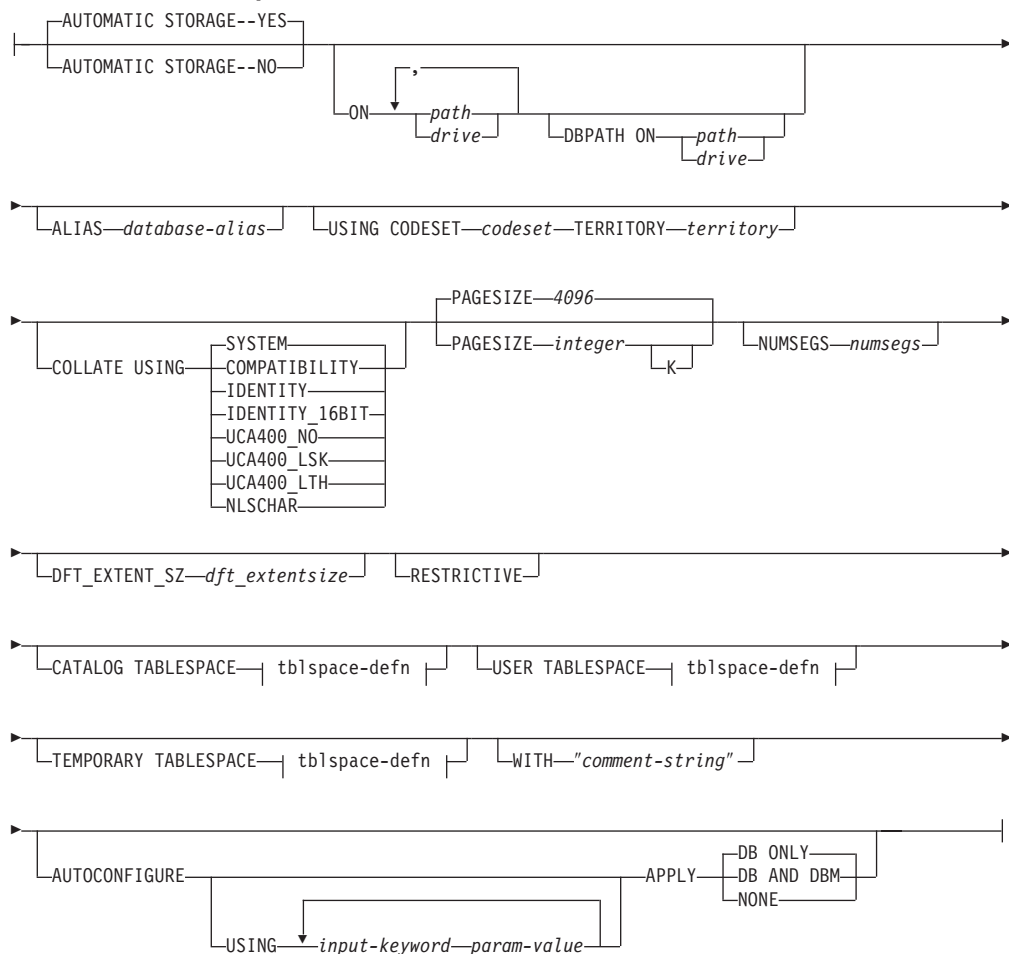
インスタンス。別の (リモート) ノードでデータベースを作成するには、まずそのノードにアタッチする必要があります。このコマンドの処理中、データベース接続が一時的に確立されます。

コマンド構文:

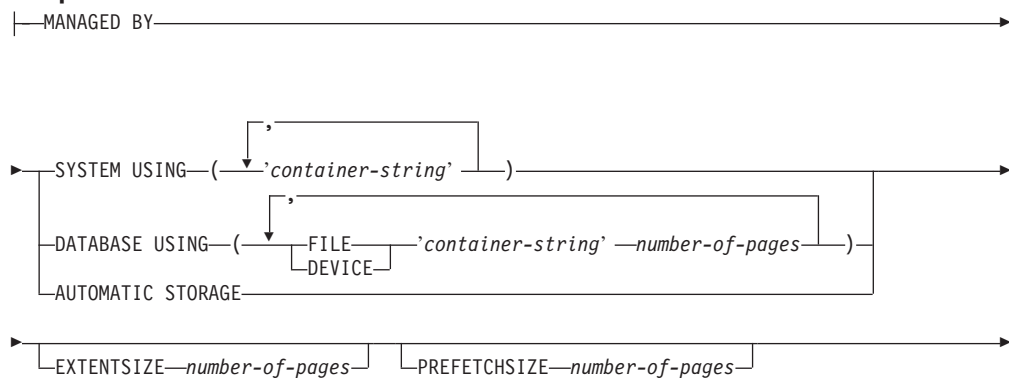
CREATE DATABASE

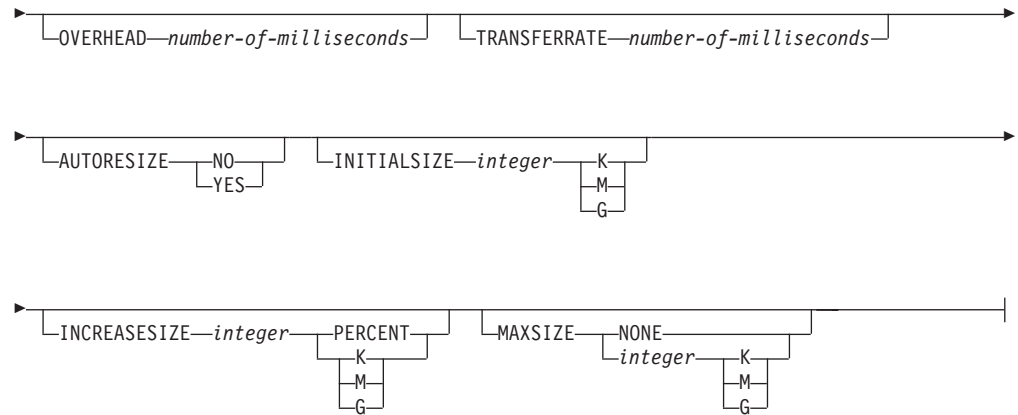


Create Database options:



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。このノード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、あるノードでデータベース・パーティショ

ンが損傷を受けたため再作成する必要がある場合、このパラメーターを **RESTORE DATABASE** コマンドと共に使用します。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 NOIYES

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** 文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 (*dfidbpath* 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 UDB 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 (*dfidbpath* 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、*dfidbpath* 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** ベースのシステムの場合、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。

XML フィーチャーを利用するには、コード・セットを UTF-8 に設定する必要があります。

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。

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。

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 4.00 に基づく UCA (Unicode Collation Algorithm) 照合順序が、暗黙的にオンに設定されます。UCA の詳細は、Unicode Technical Standard #10 にあります。

す。この仕様は Unicode Consortium Web サイト (www.unicode.org) で入手できます。これは、コード・セット UTF-8 を指定した場合のデフォルト照合であり、Unicode データベースの作成時にのみ使用できます。

UCA400_LSK

Unicode Standard バージョン 4.00 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 4.00 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。

NLSCHAR

特定のコード・セット/テリトリ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

Unicode 以外のデータベースの場合、これはデフォルト・オプションであり、その照合シーケンスは、データベース・テリトリをベースにします。Unicode データベースの場合、このオプションは IDENTITY オプションと同等です。

PAGESIZE integer

データベース作成時のデフォルト・バッファー・プールのページ・サイズ、および初期表スペース (SYSCATSPACE、TEMPSPACE1、USERSPACE1) を指定します。また、今後のすべての CREATE BUFFERPOOL および CREATE TABLESPACE ステートメントのデフォルト・ページ・サイズも表します。接尾部 K を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 に設定されることになります。また、特権が自動的に 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

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

CREATE DATABASE

ージ・パスにコンテナが作成されます。この表スペースのエクステント・サイズは 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ト・サイズは *dft_extentsize* です。

表スペース定義フィールドについての詳細は、CREATE TABLESPACE ステートメントを参照してください。

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・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 9. 有効な入力キーワードおよびパラメーター値

キーワード	有効値	デフォルト値	説明
mem_percent	1-100	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

CREATE DATABASE

表 9. 有効な入力キーワードおよびパラメーター値 (続き)

キーワード	有効値	デフォルト値	説明
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、自己調整メモリー・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。

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、*dfidbpath* パラメーターによってデータベース管理システム構成ファイルで定義されたデフォルトのデータベース・パス上のサーバーのローカル・データベース・ディレクトリー。ローカル・データベース・ディレクトリーは、データベースが入っている各ファイル・システムに常駐しています。
 - アタッチしたインスタンスのサーバーのシステム・データベース・ディレクトリー。結果のディレクトリー項目には、データベース名とデータベース別名が入ることになります。

コマンドがリモート・クライアントから発行された場合、クライアントのシステム・データベース・ディレクトリーもデータベース名と別名で更新されます。

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、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管理バインド・ファイルをデータベースにバインドします (このリストは、ユーティリティのバインド・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

CREATE DATABASE

うな場合にでも作成されます。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、下位レベルのサーバーに対してバインドできない 2 つの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 が下位レベルのサーバーに対してバインドされていると、これらの 2 つのファイルに関係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 権限、および CONNECT、CREATETAB、BINDADD、CREATE_NOT_FENCED、IMPLICIT_SCHEMA、および LOAD 特権をデータベース作成者に。
 - CONNECT、CREATETAB、BINDADD、および IMPLICIT_SCHEMA 特権を 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。

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ベースの自動ストレージを明示的に無効にすることができます。データベースの作成時に、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します。この場合、各パーティションごとに別々のファイル・システムを作成し、/db2data/<instance>/NODE#### にマウントすることができます。データベース・マネージャーは、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承知しています。

一般的に、複数パーティション・データベースでは、どのパーティションにも同じストレージ・パスを使用する必要があり、それらはすべて、**CREATE DATABASE** コマンドの実行前に存在している必要があります。その例外の 1 つとして、ストレージ・パス内でデータベース・パーティション式を使用する場合があります。その使用によって、処理結果のパス名が各パーティションごとに異なるように、データベース・パーティション番号をストレージ・パスにおいて反映することができます。

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。

CREATE DATABASE

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/NODE^{xxx} を作成します。 ^{xxx} 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 ~ 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・パスの下に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。

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 *dfidbpath* 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 *dfidbpath* 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。

例 4:

```
CREATE DATABASE TESTDB9 ON C:
USING CODESET UTF-8 TERRITORY US
```

データベース TESTDB9 は、ドライブ C: 上に作成されます。コード・セットは UTF-8 に設定されるので、データベース上でネイティブの XML 機能を使用できるようになります。

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 DBPARTITIONNUM の代わりに NODE を使用できます。

関連概念:

- 「SQL リファレンス 第 1 巻」の『分離レベル』
- 「管理ガイド: インプリメンテーション」の『自動ストレージ・データベース』
- 「管理ガイド: プランニング」の『DB2 Database for Linux, UNIX, and Windows での Unicode のインプリメンテーション』

関連タスク:

- 「管理ガイド: インプリメンテーション」の『データベースの作成』
- 「管理ガイド: インプリメンテーション」の『システム・カタログ・ビューのセキュリティ』
- 「管理ガイド: プランニング」の『タイ語文字の照合』

関連資料:

- 378 ページの『AUTOCONFIGURE』
- 389 ページの『BIND』

CREATE DATABASE

- 408 ページの『CATALOG DATABASE 』
- 415 ページの『CATALOG LDAP DATABASE 』
- 469 ページの『DROP DATABASE 』
- 「管理 API リファレンス」の『sqlecran API - データベース・パーティション・サーバー上へのデータベース作成』
- 「管理 API リファレンス」の『sqlecrea API - データベースの作成』
- 「SQL リファレンス 第 2 巻」の『CREATE TABLESPACE ステートメント』
- 738 ページの『RESTORE DATABASE 』
- 「パフォーマンス」の『auto_maint - 自動保守構成パラメーター』
- 「パフォーマンス」の『その他の変数』
- 「パフォーマンス」の『self_tuning_mem - 自己チューニング・メモリー構成パラメーター』

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 DB2 クライアントでは無効です。

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・ディレクトリーまたはシステム・データベース・ディレクトリーの他のデータベースから、そのデータベースを区別する、固有名でなければなりません。名前は、データベースの命名規則に適合していることが必要です。

EXISTING DATABASE *database-name*

ツール・カタログのホストになる既存のデータベースの名前。ローカル・データベースでなければなりません。

EXISTING TABLESPACE *tablespace-name*

DB2 ツール・カタログ表を作成するのに使用される、既存の 32K ページの表スペースを指定するのに使う名前。表を正常に作成するためには、32K ページ・サイズの TEMPORARY 表スペースも必要です。

FORCE

新しい表スペースでツール・カタログを作成する場合は、データベース・マネージャーを再開することが必要です。この場合、アプリケーションが接続してはなりません。FORCE オプションを使って、データベースに接続しているアプリケーションを確実になくします。アプリケーションが接続していると、既存の表スペースを指定しない限り、ツール・カタログの作成は失敗します。

KEEP INACTIVE

このオプションは、DB2 Administration Server 構成パラメーターを更新したり、スケジューラーを有効にしたりすることはありません。

例:

```
db2 create tools catalog cc create new database toolsdb

db2 create tools catalog 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) 必要です。さらに、既存の表スペースを指定しない限り、その表スペースには新しい 32 K のバッファ・プールが作成されます。これにはデータベース・マネージャーの再開が必要です。データベース・マネージャーを再開する必要がある場合、すべての既存のアプリケーションは強制クローズしなければなりません。新しい表スペースは、デフォルトのデータベース・ディレクトリー・パスのそれぞれで、1 つのコンテナを伴って作成されます。
- このコマンドの実行前に、この名前のアクティブ・カタログが存在する場合、そのカタログは非アクティブ化され、新しいカタログがアクティブ・カタログになります。
- 同じデータベースに複数の DB2 ツール・カタログが作成されることがあり、それらはカタログ名によって固有に識別されます。
- jdk_path* 構成パラメーターは、DB2 Administration Server (DAS) 構成で、最小サポート・レベルの Java 用 SDK に設定する必要があります。
- DAS 構成パラメーターを更新するには、DB2 Administration Server 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。

関連概念:

- 「管理ガイド: インプリメンテーション」の『DB2 Administration Server』

関連資料:

- 「パフォーマンス」の『jdk_64_path - 「64 ビット Software Developer's Kit for Java インストール・パスDAS」構成パラメーター』
- 「パフォーマンス」の『jdk_path - 「Software Developer's Kit for Java インストール・パス DAS」構成パラメーター』

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:

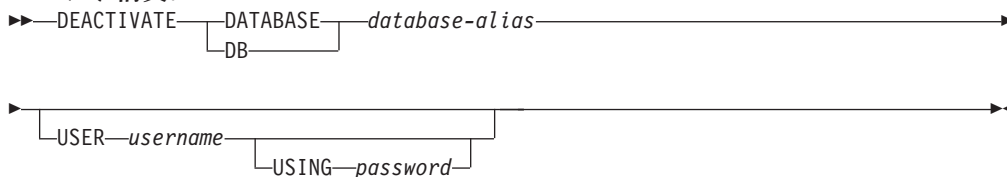
以下のいずれか。

- *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。

関連概念:

- 「パフォーマンス」の『パフォーマンス・チューニングのためのクイック・スタート・ヒント』

関連資料:

- 806 ページの『STOP DATABASE MANAGER 』
- 363 ページの『ACTIVATE DATABASE 』
- 「管理 API リファレンス」の『sqle_deactivate_db API - データベースの非活動化』

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)
- 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:

```

▶▶—DECOMPOSE XML DOCUMENT—xml-document-name—XMLSCHEMA—xml-schema-name————▶
|
|_VALIDATE_|

```

コマンド・パラメーター:

DECOMPOSE XML DOCUMENT *xml-document-name*

xml-document-name は、分解される入力 XML 文書のファイル・パスおよびファイル名です。

XMLSCHEMA *xml-schema-name*

xml-schema-name は、文書の分解に使用される、XML スキーマ・リポジトリに登録された既存の XML スキーマの名前です。 *xml-schema-name* は修飾 SQL ID で、オプションの SQL スキーマ名の後にピリオドと XML スキーマ名が続く形で構成されます。 SQL スキーマ名が指定されない場合、DB2 特殊レジスター CURRENT SCHEMA の値であると想定されます。

VALIDATE

このパラメーターは、入力 XML 文書が最初に妥当性検査され、その文書が有効な場合に限り、分解されることを示します。 VALIDATE が指定されない場合、入力 XML 文書は分解前に妥当性検査されません。

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 が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"
```

関連概念:

- 「XML ガイド」の『XML スキーマ・リポジトリ (XSR) を使用した、XML スキーマ、DTD、および外部エンティティ管理』

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。

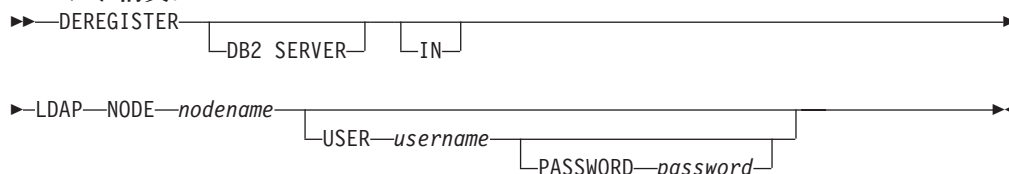
許可:

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。

NODEnodename

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。

関連概念:

- 「管理ガイド: インプリメンテーション」の『Lightweight Directory Access Protocol (LDAP) 』

関連タスク:

- 「管理ガイド: インプリメンテーション」の『DB2 サーバーの登録を解除する』

関連資料:

- 「管理 API リファレンス」の『db2LdapDeregister API - LDAP サーバーからの DB2 サーバーおよびカタログされたデータベースの登録解除』
- 697 ページの『REGISTER 』
- 852 ページの『UPDATE LDAP NODE 』

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 SQLDA 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 SQLDA 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 (および SHOW DETAIL)
- GET_INDEX_COLNAMES() UDF に対する SYSCAT.INDEXES (DESCRIBE INDEXES FOR TABLE) 実行特権 (および SHOW DETAIL)
- 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。

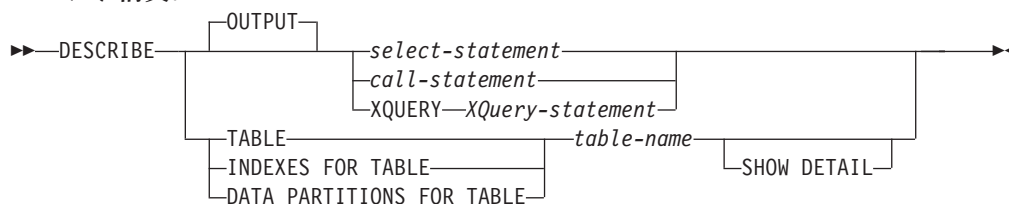
CALL ステートメントに関する出力 SQLDA 情報を表示するには、下にリストされている特権または権限のいずれかが必要です。

- 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、または 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。

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。 *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 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。 *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 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

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。

SHOW DETAIL

DESCRIBE TABLE コマンドの場合、以下の追加情報を出力に含めることを指定します。

- 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

DESCRIBE

- 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)

DESCRIBE INDEXES FOR TABLE コマンドは、出力に以下の追加情報を含めることを指定します。

- 列名

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

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"
```

SQLDA Information

```
sqldaid :SQLDA sqldabc:896 sqln:20 sqld:7
```

Column Information

sqltype	sqllen	sqlname.data	sqlname.length
500 SMALLINT	2	ID	2
449 VARCHAR	9	NAME	4
501 SMALLINT	2	DEPT	4
453 CHARACTER	5	JOB	3
501 SMALLINT	2	YEARS	5
485 DECIMAL	7,2	SALARY	6
485 DECIMAL	7,2	COMM	4

CALL ステートメントの出力の記述

次のステートメントでストアド・プロシージャが作成されたとします。

```
CREATE PROCEDURE GIVE_BONUS (IN EMPNO INTEGER,  
                             IN DEPTNO INTEGER,  
                             OUT CHEQUE INTEGER,  
                             INOUT BONUS DEC(6,0))  
...
```

次の例は、CALL ステートメントの出力を記述する方法を示しています。


```
db2 "describe output call give_bonus(123456, 987, ?, 15000.)"
```

SQLDA Information

```
sqldaid :SQLDA sqldabc:896 sqln:20 sqld:2
```

Column Information

sqltype	sqllen	sqlname.data	sqlname.length
497 INTEGER	4		
485 DECIMAL	6,0		

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'
```

SQLDA Information

```
sqldaid : SQLDA sqldabc: 1136 sqln: 20 sqld: 1
```

Column Information

sqltype	sqllen	sqlname.data	sqlname.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
```

DESCRIBE

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

関連資料:

- 「管理 SQL ルーチンおよびビュー」の『ADMIN_CMD プロシージャー - 管理コマンドの実行』
- 「管理 SQL ルーチンおよびビュー」の『DESCRIBE コマンド (ADMIN_CMD プロシージャーを使用)』
- 「SQL リファレンス 第 1 巻」の『SYSCAT.DATAPARTITIONS カタログ・ビュー』

DETACH

論理 DBMS インスタンス接続を除去し、この層を使用した論理接続がほかがない場合、物理通信接続を終了します。

許可:

なし

必要な接続:

なし。既存のインスタンス接続を除去します。

コマンド構文:

▶—DETACH—◀

コマンド・パラメーター:

なし

関連タスク:

- 「管理ガイド: インプリメンテーション」の『データベース・マネージャーの非デフォルト・インスタンスに対するアタッチおよびデタッチ』

関連資料:

- 376 ページの『ATTACH 』
- 「管理 API リファレンス」の『sqledtin API - インスタンスからのデタッチ』

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。

関連資料:

- 「管理 API リファレンス」の『db2DropContact API - 通知メッセージを送信できる連絡先リストからの連絡先の削除』
- 「管理 SQL ルーチンおよびビュー」の『DROP CONTACT コマンド (ADMIN_CMD プロシージャを使用)』

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。

関連資料:

- 「管理 API リファレンス」の『db2DropContactGroup API - 通知メッセージを送信できる連絡先リストからの連絡先グループの削除』
- 「管理 SQL ルーチンおよびビュー」の『DROP CONTACTGROUP コマンド (ADMIN_CMD プロシージャを使用)』

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・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:

```

▶ DROP DATABASE database-alias AT DBPARTITIONNUM
    DB
  
```

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。

DROP DATABASE

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。

このコマンドは、DATALINK 列を介してリンクされているすべてのファイルをリンク解除します。リンク解除操作は DB2 Data Links Manager で非同期に実行されるので、効果が DB2 Data Links Manager で即時に見られたり、リンク解除されたファイルが即時に他の操作で使えるようになったりするとは限りません。このコマンドを発行するときは、データベースに構成されたすべての DB2 Data Links Manager が使用できる状態でなければなりません。そうでない場合、データベースのドロップ操作は失敗します。

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 DBPARTITIONNUM の代わりに NODE を使用できます。

関連タスク:

- 「管理ガイド: インプリメンテーション」の『データベースのドロップ』

関連資料:

- 435 ページの『CREATE DATABASE 』
- 816 ページの『UNCATALOG DATABASE 』
- 「管理 API リファレンス」の『sqledpan API - データベース・パーティション・サーバーでのデータベースのドロップ』
- 「管理 API リファレンス」の『sqledrpd API - データベースのドロップ』
- 408 ページの『CATALOG DATABASE 』

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・システムからドロップする前に使用してください。

有効範囲:

このコマンドは、それが発行されたデータベース・パーティションに対してだけ影響を与えます。

許可:

sysadm

コマンド構文:

►—DROP DBPARTITIONNUM VERIFY—◄

コマンド・パラメーター:

なし

使用上の注意:

データベース・パーティションが使用中ではないことを示すメッセージが戻された場合、STOP DATABASE MANAGER コマンドに DROP DBPARTITIONNUM を指定して使用し、db2nodes.cfg ファイルからそのデータベース・パーティションの項目をドロップしてください。これでそのデータベース・パーティションはデータベース・システムからドロップされます。

そのデータベース・パーティションが使用中であることを示すメッセージが戻された場合、以下のアクションをとる必要があります。

1. そのデータベース・パーティションにデータがある場合、REDISTRIBUTE DATABASE PARTITION GROUP を使用してデータを再配分し、データベース・パーティションからデータを除去します。データベースの任意のデータベース・パーティション・グループからデータベース・パーティションをドロップするには、REDISTRIBUTE DATABASE PARTITION GROUP コマンドか ALTER DATABASE PARTITION GROUP ステートメントで、DROP DBPARTITIONNUM オプションを使用します。このアクションを、データベース・パーティション・グループ中にデータベース・パーティションを含むデータベースごとに行う必要があります。
2. データベース・パーティションで定義されているイベント・モニターをすべてドロップします。
3. DROP DBPARTITIONNUM VERIFY を実行し、データベースが使用中ではなくなったことを確認してください。

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 DBPARTITIONNUM の代わりに NODE を使用できます。

関連資料:

DROP DBPARTITIONNUM VERIFY

- 806 ページの『STOP DATABASE MANAGER 』
- 692 ページの『REDISTRIBUTE DATABASE PARTITION GROUP 』
- 「管理 API リファレンス」の『sqledrpn API - データベース・パーティション・サーバーがドロップ可能かどうかの検査』

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 DB2 クライアントでは無効です。

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ラができなくなり、スケジューラ済みのタスクも実行されません。スケジューラを活動化するには、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ことが必要です。

有効範囲:

このコマンドはデータベースに影響します。

許可:

以下のいずれか。

- 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。

DROP TOOLS CATALOG

- 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。

関連タスク:

- 「管理ガイド: インプリメンテーション」の『DB2 ツール・カタログのデータベースを作成する』

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:

▶ EDIT [E] [EDITOR=editor] [num] ▶

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。

関連資料:

- 540 ページの『HISTORY』

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。

許可:

以下のいずれか。

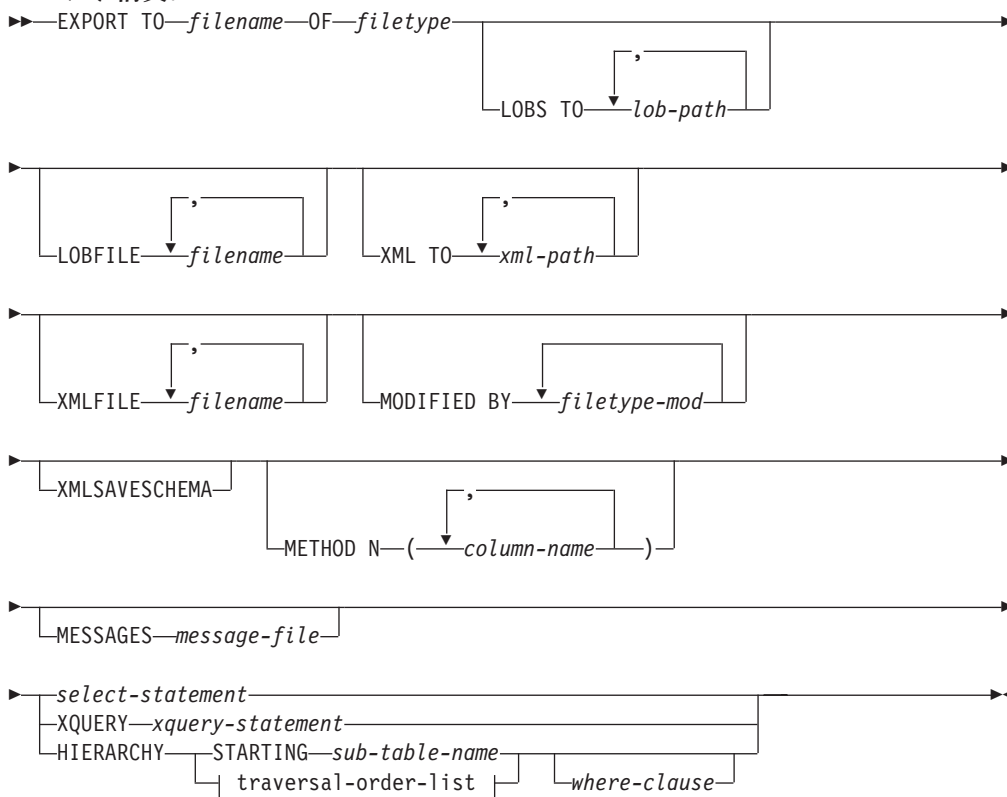
- *sysadm*
- *dbadm*

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。指定できるファイル名の最大数は 999 です。これによって、LOBSINFILE 動作が暗黙的に活動化されます。

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、それに 3 桁のシーケンス番号および 3 文字の ID *lob* を追加した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。

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する 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。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。

- SELECT ステートメントで表が指定してある場合を除き、ほとんどの表属性である IXF (統合交換フォーマット、PC バージョン) と、既存の索引が IXF ファイルに保管されます。このフォーマットを使用すると、表は再作成されます。一方、他のファイル・フォーマットを使用する場合、データをそこにインポートするには表が存在していなければなりません。

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 (QDM) インスタンスが含まれることになります。複数のパスが指定された場合、QDM 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は存在しない場合は、対応する XDS に SCH 属性は含められません。

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、整形式 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 lob to mylobs/
lobfile lobsl, lobsl2 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
lob to /db2expl/, /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。
- 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 は、DB2 for OS/390[®]、DB2 for VM and VSE、および DB2 for OS/400[®] などの DRDA[®] サーバーから表をエクスポートするために使用できます。PC/IXF エクスポートだけがサポートされています。
- エクスポート・ユーティリティは、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。
- エクスポート・ユーティリティは、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 文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とき、全選択と選択ステートメント 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 クリ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 クリ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返します。

```
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 文節を使用した export コマンドでは、期待される結果が返されません。

```
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
```

関連概念:

- 「データ移動ユーティリティ ガイドおよびリファレンス」の『エクスポートの概要』
- 「データ移動ユーティリティ ガイドおよびリファレンス」の『エクスポートの使用に必要な特権、権限、および許可』

関連タスク:

- 「データ移動ユーティリティ ガイドおよびリファレンス」の『データのエクスポート』

関連資料:

- 「管理 SQL ルーチンおよびビュー」の『ADMIN_CMD プロシージャ - 管理コマンドの実行』
- 「管理 SQL ルーチンおよびビュー」の『EXPORT コマンド (ADMIN_CMD プロシージャを使用)』
- 「データ移動ユーティリティ ガイドおよびリファレンス」の『エクスポート・セッション - CLP の例』
- 「データ移動ユーティリティ ガイドおよびリファレンス」の『インポートおよびエクスポートに関する LOB および XML ファイルの振る舞い』

484 コマンド・リファレンス

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。

データベースの保全性を確保するため、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。

データベースを作成しているユーザーは強制終了できません。

FORCE が出された後も、データベースはまだ接続要求を受諾します。すべてのユーザーを完全に強制終了するためには、追加の FORCE が必要になる場合があります。

関連資料:

- 570 ページの『LIST APPLICATIONS 』
- 376 ページの『ATTACH 』
- 「管理 API リファレンス」の『sqlefrce API - ユーザーおよびアプリケーションの強制終了』
- 「管理 SQL ルーチンおよびビュー」の『FORCE APPLICATION コマンド (ADMIN_CMD プロシージャを使用)』
- 「管理 SQL ルーチンおよびビュー」の『APPLICATIONS 管理ビュー - 接続されているデータベース・アプリケーション情報の検索』

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。

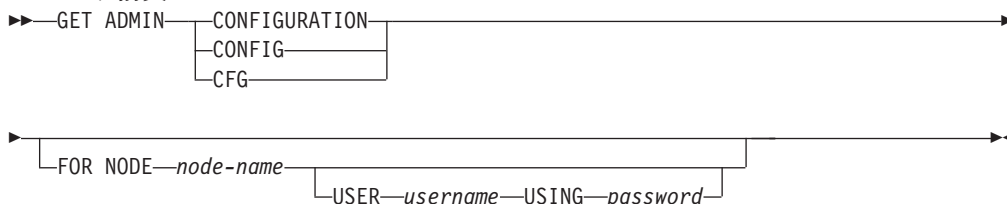
許可:

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

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:\sql\lib\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。

関連資料:

- 726 ページの『RESET ADMIN CONFIGURATION』
- 827 ページの『UPDATE ADMIN CONFIGURATION』
- 「パフォーマンス」の『構成パラメーターのサマリー』

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。

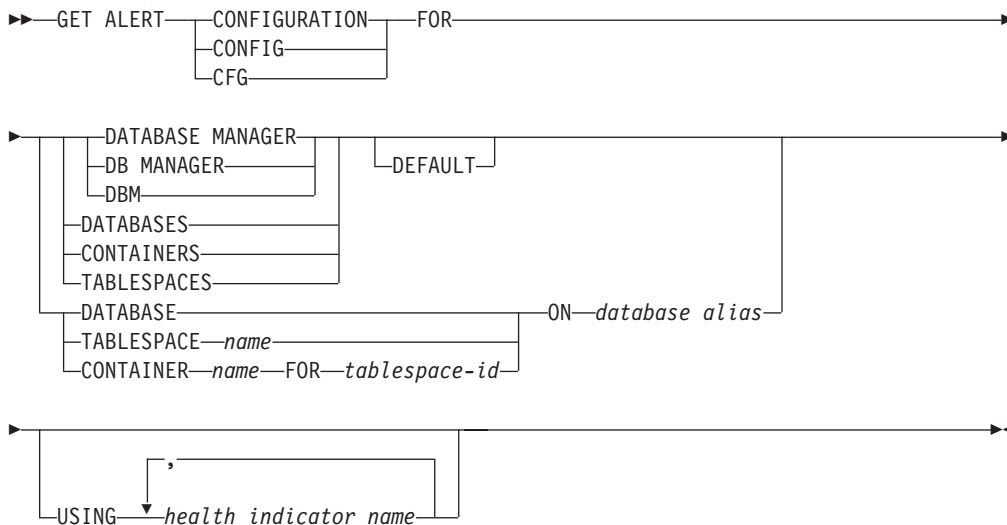
許可:

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` 文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"` 文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` 文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* 文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*" 文節を使って指定したデータベース上で、"FOR *tablespace-id*" 文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* 文節を使って指定したデータベース上で、*name* という名前の表スペースのアラート設定を検索します。この表スペースがカスタム設定を持たない場合、データベースの全表スペースの設定が戻されます。これは、TABLESPACES パラメーターと同じ結果です。

USING health indicator name

アラート構成情報が戻されるヘルス・インディケーターのセットを指定します。ヘルス・インディケーター名は 2 文字のオブジェクト ID で構成され、その後にインディケーターの測定対象を説明する名前が続きます。例えば、db.sort_privmem_util のようになります。これはオプションの文節で、これを使用しない場合は、指定したオブジェクトまたはオブジェクト・タイプのすべてのヘルス・インディケーターが戻されます。

例:

以下に示すのは、データベース・マネージャー情報を要求した結果として表示される一般的な出力です。

DB2 GET ALERT CFG FOR DBM

```

Alert Configuration
Indicator Name      = db2.db2_op_status
Default            = Yes
Type               = State-based
Sensitivity        = 0
Formula            = db2.db2_status;
Actions            = Disabled
Threshold or State checking = Enabled

Indicator Name      = db2.sort_privmem_util
Default            = Yes
Type               = Threshold-based
Warning            = 90
Alarm              = 100
Unit               = %
Sensitivity        = 0
Formula            = ((db2.sort_heap_allocated/sheapthres)
                    *100);
Actions            = Disabled
Threshold or State checking = Enabled

```

GET ALERT CONFIGURATION

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

GET ALERT CONFIGURATION

```

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

```

GET ALERT CONFIGURATION

```

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

```

GET ALERT CONFIGURATION

```

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

```

関連タスク:

- 「システム・モニター ガイドおよびリファレンス」の『クライアント・アプリケーションを使用したヘルス・インディケーターの構成』

関連資料:

- 「管理 API リファレンス」の『db2GetAlertCfg API - ヘルス・インディケーターのアラート構成設定の取得』
- 「管理 SQL ルーチンおよびビュー」の『HEALTH_GET_ALERT_ACTION_CFG 表関数 - ヘルス・アラート・アクション構成設定値の検索』
- 「管理 SQL ルーチンおよびビュー」の『HEALTH_GET_ALERT_CFG 表関数 - ヘルス・アラート構成設定値の検索』

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。

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 CREATE TAB 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 authority	= YES

Indirect SYSADM authority	= YES
Indirect SYSCtrl authority	= NO
Indirect SYSMaint authority	= NO
Indirect DBADM authority	= NO
Indirect CREATE TAB 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'
```

関連概念:

- 「管理ガイド: プランニング」の『許可』

関連資料:

- 「SQL リファレンス 第 1 巻」の『SYSCAT.DBAUTH カタログ・ビュー』

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 文節を指定する必要があります。

関連資料:

- 838 ページの『UPDATE CLI CONFIGURATION 』
- 696 ページの『REFRESH LDAP 』

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。

関連資料:

- 784 ページの『SET CLIENT 』
- 834 ページの『UPDATE ALTERNATE SERVER FOR DATABASE』

GET CONTACTGROUP

ローカル・システムで定義される、1 つの連絡先グループに含まれる連絡先を取得します。連絡先とは、スケジューラーおよびヘルス・モニターがメッセージを送信する先のユーザーです。ADD CONTACTGROUP コマンドを使用すると、名前付きの連絡先のグループを作成できます。

許可:

なし。

必要な接続:

なし。ローカル実行のみ: このコマンドはリモート接続では使用できません。

コマンド構文:

► GET CONTACTGROUP *name* ◄

コマンド・パラメーター:

CONTACTGROUP *name*

連絡先を検索するグループの名前。

例:

```
GET CONTACTGROUP support
```

Description

Foo Widgets broadloom support unit

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

関連資料:

- 「管理 API リファレンス」の『db2GetContactGroup API - 通知メッセージを送信できる単一の連絡先グループ中の連絡先のリストの取得』
- 「管理 SQL ルーチンおよびビュー」の『CONTACTGROUPS 管理ビュー - 連絡先グループのリストの検索』

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、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

関連資料:

- 「管理 API リファレンス」の『db2GetContactGroups API - 通知メッセージを送信できる連絡先グループのリストの取得』
- 「管理 SQL ルーチンおよびビュー」の『CONTACTGROUPS 管理ビュー - 連絡先グループのリストの検索』

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 ◀

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

関連資料:

- 「管理 API リファレンス」の『db2GetContacts API - 通知メッセージを送信できる連絡先リストの取得』
- 「管理 SQL ルーチンおよびビュー」の『CONTACTS 管理ビュー - 連絡先のリストの検索』

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。

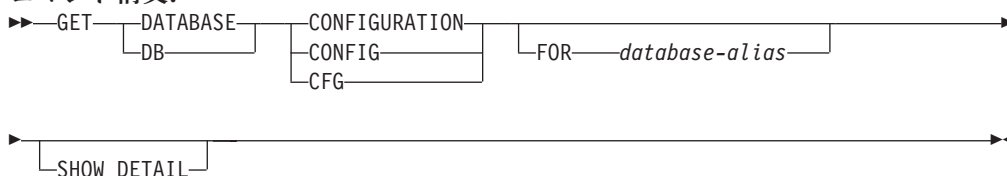
許可:

なし

必要な接続:

インスタンス。SHOW DETAIL 文節を使用するとき、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 (AIX で発行)。

Database Configuration for Database mick

Database configuration release level

= 0x0a00

GET DATABASE CONFIGURATION

```

Database release level                      = 0x0a00

Database territory                         = en_US
Database code page                         = 819
Database code set                         = ISO8859-1
Database country/region code              = 1
Database collating sequence               = UNIQUE
Alternate collating sequence              (ALT_COLLATE) =
Database page size                        = 4096
Dynamic SQL Query management              (DYN_QUERY_MGMT) = DISABLE

Discovery support for this database        (DISCOVER_DB) = ENABLE

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

Backup pending                            = NO

Database is consistent                     = YES
Rollforward pending                       = NO
Restore pending                           = NO

Multi-page file allocation enabled        = YES

Log retain for recovery status             = NO
User exit for logging status              = NO

Data Links Token Expiry Interval (sec)    (DL_EXPINT) = 60
Data Links Write Token Init Expiry Intvl (DL_WT_IEXPINT) = 60
Data Links Number of Copies               (DL_NUM_COPIES) = 1
Data Links Time after Drop (days)        (DL_TIME_DROP) = 1
Data Links Token in Uppercase             (DL_UPPER) = NO
Data Links Token Algorithm                 (DL_TOKEN) = MACO

Database heap (4KB)                       (DBHEAP) = 1200
Size of database shared memory (4KB)      (DATABASE_MEMORY) = AUTOMATIC
Catalog cache size (4KB)                  (CATALOGCACHE_SZ) = 64
Log buffer size (4KB)                     (LOGBUFSZ) = 8
Utilities heap size (4KB)                  (UTIL_HEAP_SZ) = 5000
Buffer pool size (pages)                   (BUFFPAGE) = 1000
Max storage for lock list (4KB)            (LOCKLIST) = 128

Max size of appl. group mem set (4KB)      (APPGROUP_MEM_SZ) = 30000
Percent of mem for appl. group heap        (GROUPHEAP_RATIO) = 70
Max appl. control heap size (4KB)          (APP_CTL_HEAP_SZ) = 128

Sort heap thres for shared sorts (4KB)     (SHEAPTHRES_SHR) = (SHEAPTHRES)
Sort list heap (4KB)                      (SORTHEAP) = 256
SQL statement heap (4KB)                   (STMHEAP) = 2048
Default application heap (4KB)             (APPLHEAPSZ) = 128
Package cache size (4KB)                   (PCKCACHESZ) = (MAXAPPLS*8)
Statistics heap size (4KB)                 (STAT_HEAP_SZ) = 4384

Interval for checking deadlock (ms)         (DLCHKTIME) = 10000
Percent. of lock lists per application      (MAXLOCKS) = 10
Lock timeout (sec)                         (LOCKTIMEOUT) = -1

Changed pages threshold                    (CHNGPGS_THRESH) = 60
Number of asynchronous page cleaners        (NUM_IOCLEANERS) = 1
Number of I/O servers                      (NUM_IOSERVERS) = 3
Index sort flag                            (INDEXSORT) = YES
Sequential detect flag                     (SEQDETECT) = YES

```

GET DATABASE CONFIGURATION

```

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) = 1
Max DB files open per application       (MAXFILOP) = 64

Log file size (4KB)                    (LOGFILSIZ) = 1000
Number of primary log files              (LOGPRIMARY) = 3
Number of secondary log files            (LOGSECOND) = 2
Changed path to log files                (NEWLOGPATH) =
Path to log files                       = /home/db2inst/db2inst
                                         /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
Percent of max primary log space by transaction (MAX_LOG) = 0
Num. of active log files for 1 active UOW (NUM_LOG_SPAN) = 0

Group commit count                     (MINCOMMIT) = 1
Percent log file reclaimed before soft ckcpt (SOFTMAX) = 100
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

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

TSM management class                   (TSM_MGMTCLASS) =
TSM node name                          (TSM_NODENAME) =
TSM owner                              (TSM_OWNER) =
TSM password                           (TSM_PASSWORD) =

Automatic maintenance                  (AUTO_MAINT) = OFF
Automatic database backup               (AUTO_DB_BACKUP) = OFF
Automatic table maintenance             (AUTO_TBL_MAINT) = OFF
Automatic runstats                      (AUTO_RUNSTATS) = OFF
Automatic statistics profiling          (AUTO_STATS_PROF) = OFF
Automatic profile updates               (AUTO_PROF_UPD) = OFF
Automatic reorganization                (AUTO_REORG) = OFF

```

GET DATABASE CONFIGURATION

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 mick			
Description	Parameter	Current Value	Delayed Value
Database configuration release level		= 0x0a00	
Database release level		= 0x0a00	
Database territory		= en_US	
Database code page		= 819	
Database code set		= ISO8859-1	
Database country/region code		= 1	
Database collating sequence		= UNIQUE	UNIQUE
Alternate collating sequence	(ALT_COLLATE)	=	
Database page size		= 4096	
Dynamic SQL Query management	(DYN_QUERY_MGMT)	= DISABLE	DISABLE
Discovery support for this database	(DISCOVER_DB)	= ENABLE	ENABLE
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
Backup pending		= NO	
Database is consistent		= YES	
Rollforward pending		= NO	
Restore pending		= NO	
Multi-page file allocation enabled		= YES	
Log retain for recovery status		= NO	
User exit for logging status		= NO	
Data Links Token Expiry Interval (sec)	(DL_EXPINT)	= 60	60
Data Links Write Token Init Expiry Intvl	(DL_WT_IXPINT)	= 60	60
Data Links Number of Copies	(DL_NUM_COPIES)	= 1	1
Data Links Time after Drop (days)	(DL_TIME_DROP)	= 1	1
Data Links Token in Uppercase	(DL_UPPER)	= NO	NO
Data Links Token Algorithm	(DL_TOKEN)	= MAC0	MAC0
Database heap (4KB)	(DBHEAP)	= 1200	1200
Size of database shared memory (4KB)	(DATABASE_MEMORY)	= AUTOMATIC (11516)	AUTOMATIC (11516)
Catalog cache size (4KB)	(CATALOGCACHE_SZ)	= 64	64
Log buffer size (4KB)	(LOGBUFSZ)	= 8	8
Utilities heap size (4KB)	(UTIL_HEAP_SZ)	= 5000	5000
Buffer pool size (pages)	(BUFFPAGE)	= 1000	1000
Max storage for lock list (4KB)	(LOCKLIST)	= 128	128
Max size of appl. group mem set (4KB)	(APPGROUP_MEM_SZ)	= 30000	30000
Percent of mem for appl. group heap	(GROUPHEAP_RATIO)	= 70	70
Max appl. control heap size (4KB)	(APP_CTL_HEAP_SZ)	= 128	128
Sort heap thres for shared sorts (4KB)	(SHEAPTHRES_SHR)	= (SHEAPTHRES)	(SHEAPTHRES)
Sort list heap (4KB)	(SORTHEAP)	= 256	256
SQL statement heap (4KB)	(STMHEAP)	= 2048	2048
Default application heap (4KB)	(APPLHEAPSZ)	= 128	128
Package cache size (4KB)	(PCKCACHEsz)	= (MAXAPPLS*8)	(MAXAPPLS*8)
Statistics heap size (4KB)	(STAT_HEAP_SZ)	= 4384	4384
Interval for checking deadlock (ms)	(DLCHKTIME)	= 10000	10000
Percent. of lock lists per application	(MAXLOCKS)	= 10	10

GET DATABASE CONFIGURATION

Lock timeout (sec)	(LOCKTIMEOUT) = -1	-1
Changed pages threshold	(CHNGPGS_THRESH) = 60	60
Number of asynchronous page cleaners	(NUM_IOCLEANERS) = 1	1
Number of I/O servers	(NUM_IOSERVERS) = 3	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) = 1	1
Max DB files open per application	(MAXFILOP) = 64	64
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	= home/db2inst /home /db2inst /db2inst /NODE0000 /db2inst /SQL00001 /NODE0000 /SQLOGDIR/ /SQL00001 /SQLOGDIR/ /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 of 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 chkpt	(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
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 (RESTART)	SYSTEM (RESTART)
Log pages during index build	(LOGINDEXBUILD) = OFF	OFF
Default number of loadrec sessions	(DFT_LOADREC_SES) = 1	1
Number of database backups to retain	(NUM_DB_BACKUPS) = 12	12
Recovery history retention (days)	(REC_HIS_RETENTN) = 366	366
TSM management class	(TSM_MGMTCLASS) =	
TSM node name	(TSM_NODENAME) =	

GET DATABASE CONFIGURATION

TSM owner	(TSM_OWNER) =	
TSM password	(TSM_PASSWORD) =	
Automatic maintenance	(AUTO_MAINT) =	OFF
Automatic database backup	(AUTO_DB_BACKUP) =	OFF
Automatic table maintenance	(AUTO_TBL_MAINT) =	OFF
Automatic runstats	(AUTO_RUNSTATS) =	OFF
Automatic statistics profiling	(AUTO_STATS_PROF) =	OFF
Automatic profile updates	(AUTO_PROF_UPD) =	OFF
Automatic reorganization	(AUTO_REORG) =	OFF

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。

関連タスク:

- 「管理ガイド: インプリメンテーション」の『ノードおよびデータベース構成ファイルの変更』
- 「パフォーマンス」の『構成パラメーターによる DB2 の構成』

関連資料:

- 730 ページの『RESET DATABASE CONFIGURATION』
- 844 ページの『UPDATE DATABASE CONFIGURATION』
- 「管理 API リファレンス」の『db2CfgGet API - データベース・マネージャー構成パラメーター、あるいはデータベース構成パラメーターの取得』
- 「パフォーマンス」の『構成パラメーターのサマリー』
- 「管理 SQL ルーチンおよびビュー」の『DBCFG 管理ビュー - データベース構成パラメーター情報の検索』

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。

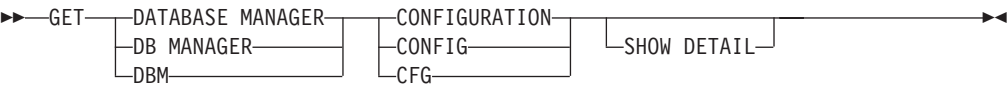
許可:

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必要があります。SHOW DETAIL 文節では、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 (AIX で発行)。

```
Database Manager Configuration

Node type = Database Server with local clients

Database manager configuration release level          = 0x0a00

CPU speed (millisec/instruction)                    (CPUSPEED) = 4.000000e-05

Max number of concurrently active databases          (NUMDB) = 8
Data Links support                                  (DATA LINKS) = NO
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) = /usr/java131

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
```

GET DATABASE MANAGER CONFIGURATION

```

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) =
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
Database manager authentication         (AUTHENTICATION) = SERVER
Cataloging allowed without authority    (CATALOG_NOAUTH) = YES
Trust all clients                      (TRUST_ALLCLNTS) = YES
Trusted client authentication           (TRUST_CLNTAUTH) = CLIENT
Bypass federated authentication         (FED_NOAUTH) = NO

Default database path                  (DFTDBPATH) = /home/db2inst

Database monitor heap size (4KB)        (MON_HEAP_SZ) = 90
Java Virtual Machine heap size (4KB)    (JAVA_HEAP_SZ) = 512
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

Sort heap threshold (4KB)               (SHEAPTHRES) = 20000

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
Max number of existing agents           (MAXAGENTS) = 200
Agent pool size                         (NUM_POOLAGENTS) = 100(calculated)
Initial number of agents in pool        (NUM_INITAGENTS) = 0
Max number of coordinating agents       (MAX_COORDAGENTS) = MAXAGENTS
Max no. of concurrent coordinating agents (MAXCAGENTS) = MAX_COORDAGENTS
Max number of client connections        (MAX_CONNECTIONS) = MAX_COORDAGENTS

Keep fenced process                     (KEEPFENCED) = YES
Number of pooled fenced processes        (FENCED_POOL) = MAX_COORDAGENTS
Initial number of fenced processes       (NUM_INITFENCED) = 0

Index re-creation time and redo index build (INDEXREC) = RESTART

Transaction manager database name        (TM_DATABASE) = 1ST_CONN
Transaction resync interval (sec)        (RESYNC_INTERVAL) = 180

SPM name                                (SPM_NAME) =
SPM log size                            (SPM_LOG_FILE_SZ) = 256
SPM resync agent limit                   (SPM_MAX_RESYNC) = 20
SPM log path                             (SPM_LOG_PATH) =

TCP/IP Service name                     (SVCENAME) =
Discovery mode                           (DISCOVER) = SEARCH
Discover server instance                 (DISCOVER_INST) = ENABLE

```


GET DATABASE MANAGER CONFIGURATION

```
Maximum query degree of parallelism (MAX_QUERYDEGREE) = ANY
Enable intra-partition parallelism (INTRA_PARALLEL) = NO

No. of int. communication buffers(4KB) (FCM_NUM_BUFFERS) = AUTOMATIC
No. of int. communication channels (FCM_NUM_CHANNELS) = AUTOMATIC
```

以下の出力例は、WITH 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** の値は、データベース・マネージャー・インスタンスを次回開始する際に有効になる値です。

```
Database Manager Configuration
Node type = Database Server with local clients
Description                               Parameter    Current Value    Delayed
Value

Database manager configuration release level      = 0x0a00

CPU speed (millisec/instruction)                (CPUSPEED) = 4.000000e-05    4.000000e-05
Max number of concurrently active databases      (NUMDB) = 8            8
Data Links support                             (DATA LINKS) = NO        NO
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) = /wsdb/v81    /usr
                                                    /bldsupp      /java131
                                                    /AIX/jdk1.3.1

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) = BUILD
SYSCTRL group name                             (SYSCTRL_GROUP) =
SYSMAINT group name                            (SYSMAINT_GROUP) =
SYSMON group name                              (SYSMON_GROUP) =

Client Userid-Password Plugin                   (CLNT_PW_PLUGIN) =
Client Kerberos Plugin                          (CLNT_KRB_PLUGIN) =
Group Plugin                                    (GROUP_PLUGIN) =
GSS Plugin for Local Authorization              (LOCAL_GSSPLUGIN) =
Server Plugin Mode                             (SRV_PLUGIN_MODE) = UNFENCED UNFENCED
Server List of GSS Plugins                      (SRVCON_GSSPLUGIN_LIST) =
Server Userid-Password Plugin                   (SRVCON_PW_PLUGIN) =
Server Connection Authentication                 (SRVCON_AUTH) = NOT_SPECIFIED NOT_SPECIFIED
Database manager authentication                 (AUTHENTICATION) = SERVER SERVER
Cataloging allowed without authority             (CATALOG_NOAUTH) = YES   YES
Trust all clients                               (TRUST_ALLCLNTS) = YES   YES
Trusted client authentication                   (TRUST_CLNTAUTH) = CLIENT CLIENT
Bypass federated authentication                 (FED_NOAUTH) = NO        NO
```


GET DATABASE MANAGER CONFIGURATION

Default database path	(DFTDBPATH) = /home /db2inst	/home /db2inst
Database monitor heap size (4KB)	(MON_HEAP_SZ) = 90	90
Java Virtual Machine heap size (4KB)	(JAVA_HEAP_SZ) = 512	512
Audit buffer size (4KB)	(AUDIT_BUF_SZ) = 0	0
Size of instance shared memory (4KB)	(INSTANCE_MEMORY) = AUTOMATIC (5386)	AUTOMATIC (20)
Backup buffer default size (4KB)	(BACKBUFSZ) = 1024	1024
Restore buffer default size (4KB)	(RESTBUFSZ) = 1024	1024
Sort heap threshold (4KB)	(SHEAPTHRES) = 20000	20000
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
Max number of existing agents	(MAXAGENTS) = 200	200
Agent pool size	(NUM_POOLAGENTS) = 100	100 (calculated)
Initial number of agents in pool	(NUM_INITAGENTS) = 0	0
Max number of coordinating agents	(MAX_COORDAGENTS) = 200	MAXAGENTS
Max no. of concurrent coordinating agents	(MAXCAGENTS) = 200	MAX_COORDAGENTS
Max number of client connections	(MAX_CONNECTIONS) = 200	MAX_COORDAGENTS
Keep fenced process	(KEEPFENCED) = YES	YES
Number of pooled fenced processes	(FENCED_POOL) = MAX_ COORDAGENTS	MAX_ COORDAGENTS
Initial number of fenced processes	(NUM_INITFENCED) = 0	0
Index re-creation time and redo index build	(INDEXREC) = RESTART	RESTART
Transaction manager database name	(TM_DATABASE) = 1ST_CONN	1ST_CONN
Transaction resync interval (sec)	(RESYNC_INTERVAL) = 180	180
SPM name	(SPM_NAME) =	
SPM log size	(SPM_LOG_FILE_SZ) = 256	256
SPM resync agent limit	(SPM_MAX_RESYNC) = 20	20
SPM log path	(SPM_LOG_PATH) =	
TCP/IP Service name	(SVCENAME) =	
Discovery mode	(DISCOVER) = SEARCH	SEARCH
Discover server instance	(DISCOVER_INST) = ENABLE	ENABLE
Maximum query degree of parallelism	(MAX_QUERYDEGREE) = ANY	ANY
Enable intra-partition parallelism	(INTRA_PARALLEL) = NO	NO
No. of int. communication buffers(4KB)	(FCM_NUM_BUFFERS) = AUTOMATIC	AUTOMATIC
No. of int. communication channels	(FCM_NUM_CHANNELS) = AUTOMATIC	AUTOMATIC

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。そのような場合には、データベース・マネージャーを再インストールしてリカバリーする必要があります。

GET DATABASE MANAGER CONFIGURATION

- 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。

関連タスク:

- 「管理ガイド: インプリメンテーション」の『ノードおよびデータベース構成ファイルの変更』
- 「パフォーマンス」の『構成パラメーターによる DB2 の構成』

関連資料:

- 732 ページの『RESET DATABASE MANAGER CONFIGURATION 』
- 847 ページの『UPDATE DATABASE MANAGER CONFIGURATION 』
- 「管理 API リファレンス」の『db2CfgGet API - データベース・マネージャー構成パラメーター、あるいはデータベース構成パラメーターの取得』
- 「パフォーマンス」の『構成パラメーターのサマリー』
- 「管理 SQL ルーチンおよびビュー」の『DBMCFG 管理ビュー - データベース・マネージャー構成パラメーター情報の検索』

GET DATABASE MANAGER MONITOR SWITCHES

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 DBPARTITIONNUM の代わりに NODE を使用できます。

関連概念:

- 「システム・モニター ガイドおよびリファレンス」の『システム・モニター・スイッチ』

関連タスク:

- 「システム・モニター ガイドおよびリファレンス」の『クライアント・アプリケーションからのモニター・スイッチの設定』

関連資料:

- 「管理 API リファレンス」の『db2MonitorSwitches API - モニター・スイッチ設定の取得あるいは更新』
- 532 ページの『GET SNAPSHOT 』
- 523 ページの『GET MONITOR SWITCHES 』
- 734 ページの『RESET MONITOR 』
- 854 ページの『UPDATE MONITOR SWITCHES 』
- 「管理 SQL ルーチンおよびビュー」の『SNAPSWITCHES 管理ビューおよび SNAP_GET_SWITCHES 表関数 - データベース・スナップショットのスイッチ状態情報の検索』

GET DESCRIPTION FOR HEALTH INDICATOR

指定されたヘルス・インディケーターの記述を取得します。ヘルス・インディケーターは、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。例えば、次のようなものがあります。

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.

関連資料:

GET DESCRIPTION FOR HEALTH INDICATOR

- 「システム・モニター ガイドおよびリファレンス」の『ヘルス・インディケーター』
- 「管理 *SQL* ルーチンおよびビュー」の『HEALTH_GET_IND_DEFINITION 表関数 - ヘルス・インディケーター定義の検索』

GET HEALTH NOTIFICATION CONTACT LIST

インスタンスのヘルスについて通知される連絡先および連絡先グループのリストを取得します。連絡先リストは、非常時ヘルス状況がインスタンスまたはそのデータベース・オブジェクトに示されたときに通知される、個人の電子メール・アドレスまたはページャー・インターネット・アドレスで構成されます。

許可:

なし。

必要な接続:

インスタンス。明示的なアタッチは必要ありません。

コマンド構文:

```
► GET HEALTH NOTIFICATION CONTACT LIST ◄
```

コマンド・パラメーター:

なし。

例:

GET NOTIFICATION LIST コマンドを実行した、出力結果です。

Name	Type
Joe Brown	Contact
Support	Contact group

関連資料:

- 「管理 *API* リファレンス」の『db2GetHealthNotificationList API - ヘルス・アラート通知を送信できる連絡先リストの取得』
- 「管理 *SQL* ルーチンおよびビュー」の『NOTIFICATIONLIST 管理ビュー - ヘルス通知の連絡先リストの検索』

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:

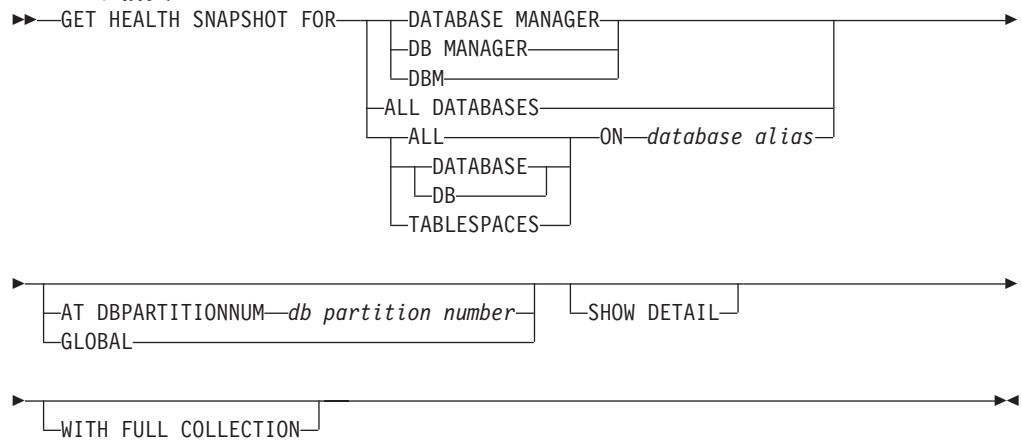
なし。

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。

GET HEALTH SNAPSHOT

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・システム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、次のようになります。

```
(03-19-2002 13:40:24.138865,50,((1-(4/8))*100)),  
(03-19-2002 13:40:13.1386300,50,((1-(4/8))*100)),  
(03-19-2002 13:40:03.1988858,0,((1-(3/3))*100))
```

ATTENTION または AUTOMATE FAILED の状態にあるすべてのコレクション・オブジェクトのコレクション・オブジェクト履歴が戻されます。

また、SHOW DETAIL オプションは、関連ヘルス・インディケーターの値とアラート状態を理解する上で役立つ追加のコンテキスト情報も提供します。例えば、表スペースのストレージ使用率ヘルス・インディケーターを使用して表スペースの使用率を判別する場合、表スペースの増大率も SHOW DETAIL によって提供されます。

WITH FULL COLLECTION

コレクションの状態に基づくすべてのヘルス・インディケーターの完全なコレクション情報を戻すことを指定します。このオプションは、名前およびサイズ・フィルターの両方の基準を検討します。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
```

関連タスク:

- 「システム・モニター ガイドおよびリファレンス」の『CLP を使用したデータベースのヘルス・スナップショットのキャプチャー』

関連資料:

- 「システム・モニター ガイドおよびリファレンス」の『ヘルス・モニター CLP コマンド』
- 「システム・モニター ガイドおよびリファレンス」の『ヘルス・モニター・インターフェースの論理データ・グループへのマッピング』
- 「システム・モニター ガイドおよびリファレンス」の『ヘルス・モニターの出力例』

- 「管理 API リファレンス」の『sqlgens API - 現行インスタンスの取得』

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必要があります。

コマンド構文:



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・システム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。

GET MONITOR SWITCHES

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 DBPARTITIONNUM の代わりに NODE を使用できます。

関連概念:

- 「システム・モニター ガイドおよびリファレンス」の『システム・モニター・スイッチ』

関連タスク:

- 「システム・モニター ガイドおよびリファレンス」の『クライアント・アプリケーションからのモニター・スイッチの設定』

関連資料:

- 「管理 API リファレンス」の『db2MonitorSwitches API - モニター・スイッチ設定の取得あるいは更新』
- 532 ページの『GET SNAPSHOT 』
- 513 ページの『GET DATABASE MANAGER MONITOR SWITCHES 』
- 734 ページの『RESET MONITOR 』
- 854 ページの『UPDATE MONITOR SWITCHES 』

GET RECOMMENDATIONS FOR HEALTH INDICATOR

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.

GET RECOMMENDATIONS FOR HEALTH INDICATOR

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。

関連タスク:

- 「システム・モニター ガイドおよびリファレンス」の『システムの正常性を保つための推奨事項をクライアント・アプリケーションを使用して検索』

関連資料:

- 「管理 API リファレンス」の『db2GetRecommendations API - アラート状態のヘルス・インディケーターを解決するための推奨の入手』
- 「管理 SQL ルーチンおよびビュー」の『AUTOCONFIGURE コマンド (ADMIN_CMD プロシージャを使用)』
- 「システム・モニター ガイドおよびリファレンス」の『ヘルス・インディケーター』
- 「システム・モニター ガイドおよびリファレンス」の『ヘルス・インディケーターの要約』

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:

$$dbadm$$

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

プロシージャーの名前。 `SPECIFIC` が指定された場合、これは特定の名前のプロシージャーになります。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ール) が正常に実行できない場合、エラー (SOLSTATE 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。

関連資料:

- 667 ページの『PUT ROUTINE 』
- 「管理 SQL ルーチンおよびビュー」の『PUT_ROUTINE_SAR プロシージャ ー』

GET SNAPSHOT

状況情報を収集して、ユーザー用に出力を形式設定します。戻された情報は、コマンドを発行した時点でのデータベース・マネージャー操作状況のスナップショットを表しています。

有効範囲:

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。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:

以下のいずれか。

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

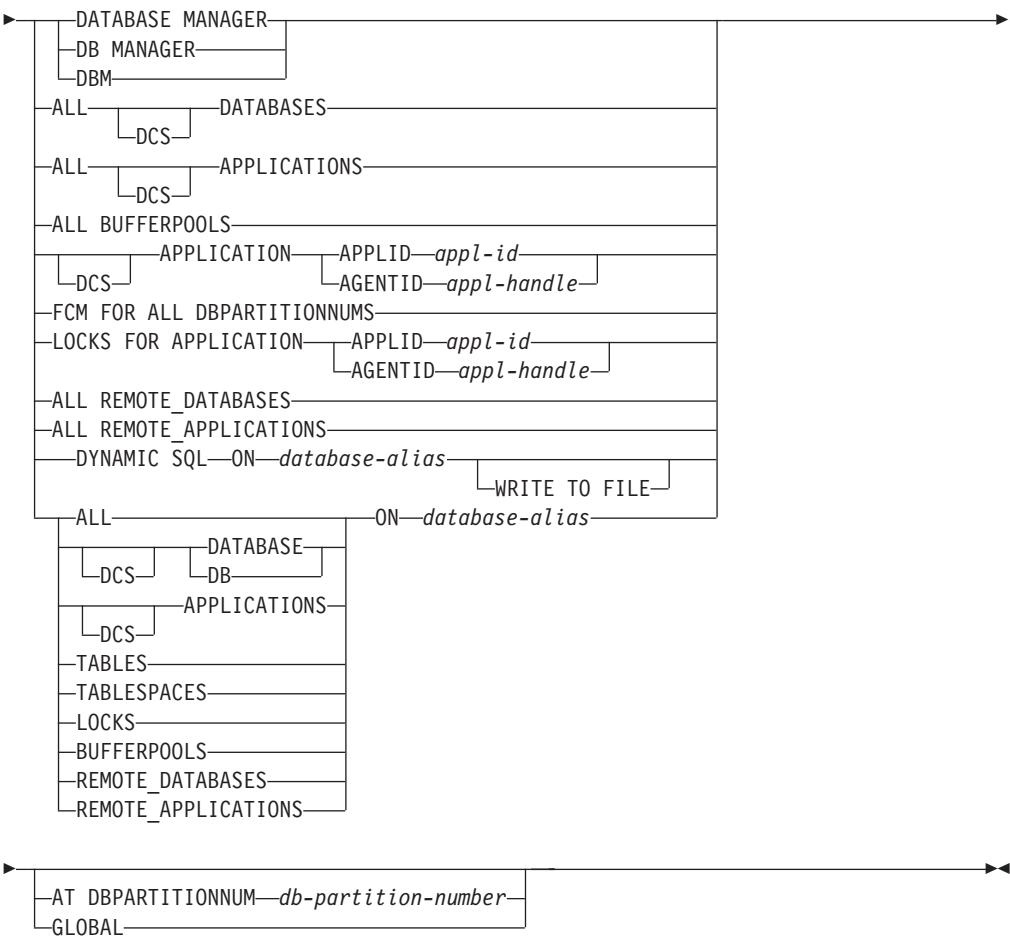
必要な接続:

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:

▶▶ GET SNAPSHOT FOR _____▶▶



統計を収集するために、モニター・スイッチをオンにする必要があります。

コマンド・パラメーター:

DATABASE MANAGER

アクティブ・データベース・マネージャー・インスタンスの統計を提供します。

ALL DATABASES

現行データベース・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 を獲得するには、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・ハンドルを知りたい場合には、LIST APPLICATIONS コマンドを使用してください。

FCM FOR ALL DBPARTITIONNUMS

GET SNAPSHOT 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 記録スイッチがオンになった後にアクセスのあった表だけが含まれます。

TABLESPACES ON database-alias

特定のデータベースの表スペースに関する情報を提供します。

LOCKS ON database-alias

特定のデータベースに接続された各アプリケーションが保留するロックすべてに関する情報を提供します。

BUFFERPOOLS ON database-alias

指定したデータベースのバッファークール活動に関する情報を提供します。

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 指定された文節に従って、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・システム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
```

- SAMPLE 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 database on sample
```

- 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オン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・モニターをオンにしない場合があります。記録スイッチ **TIMESTAMP** がオフに設定されている場合、タイム・スタンプに関連したエレメントは "Not Collected" を報告します。
- 以下の条件のいずれかが真の場合には、表情報を要求してもデータは返ってきません。
 - **TABLE** 記録スイッチがオフである。
 - スイッチをオンにして以来、アクセスのあった表がない。
 - 最後に **RESET MONITOR** コマンドを発行して以来、アクセスのあった表がない。

しかし、**REORG TABLE** が実行中であるか、またはこの期間中に実行された場合、一部の情報は戻されますが一部のフィールドは表示されません。

- 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。

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 **DBPARTITIONNUM** の代わりに **NODE** を使用できます。
- キーワード **DBPARTITIONNUMS** の代わりに **NODES** を使用できます。

関連概念:

- 「システム・モニター ガイドおよびリファレンス」の『スナップショット・モニター』

関連タスク:

- 「システム・モニター ガイドおよびリファレンス」の『クライアント・アプリケーションからのデータベース・スナップショットのキャプチャー』
- 「システム・モニター ガイドおよびリファレンス」の『スナップショット管理ビューおよび表関数を使用したデータベース・システムのスナップショットのキャプチャー』

関連資料:

- 「管理 API リファレンス」の『db2GetSnapshot API - 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の取得』
- 523 ページの『GET MONITOR SWITCHES 』
- 570 ページの『LIST APPLICATIONS 』

- 734 ページの『RESET MONITOR 』
- 854 ページの『UPDATE MONITOR SWITCHES 』

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。

関連タスク:

- 356 ページの『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』

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。

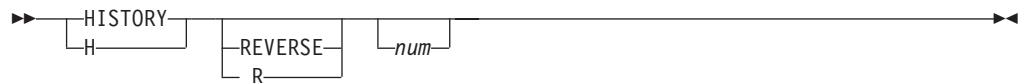
許可:

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

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。

関連資料:

- 476 ページの『EDIT 』
- 768 ページの『RUNCMD 』

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。

許可:

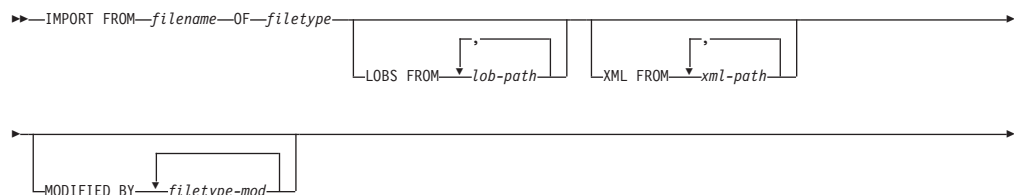
- **INSERT** オプションを使用して **IMPORT** する場合、以下のいずれかが必要です。
 - *sysadm*
 - *dbadm*
 - 関係するそれぞれの表、ビュー、またはニックネームに対する **CONTROL** 特権
 - 関係するそれぞれの表またはビューに対する **INSERT** および **SELECT** 特権
- **INSERT_UPDATE** オプションを使用して既存の表に **IMPORT** するには、以下のいずれかが必要です。
 - *sysadm*
 - *dbadm*
 - 関係するそれぞれの表、ビュー、またはニックネームに対する **CONTROL**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

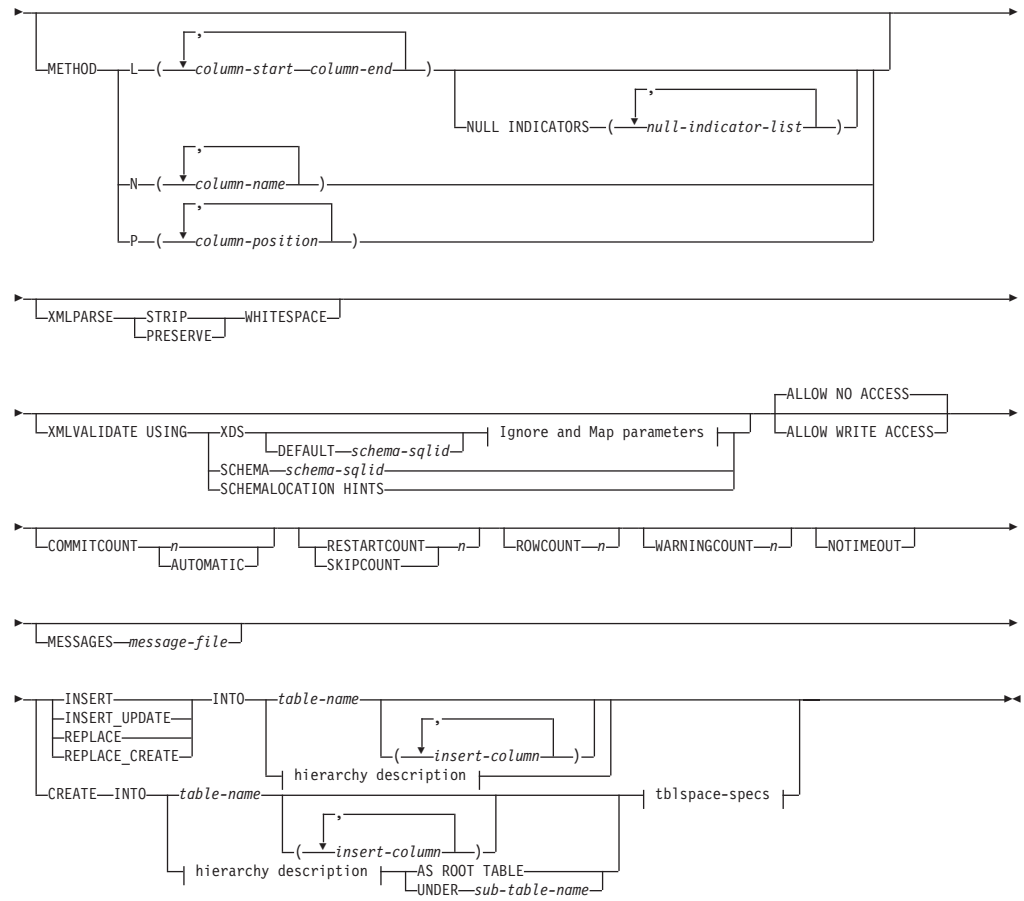
- データベースに対する CREATETAB 権限および表スペースに対する USE 特権と、以下のいずれか。
 - データベースに対する IMPLICIT_SCHEMA 権限 (表のスキーマ名が存在しない場合)
 - スキーマに対する CREATEIN 特権 (表のスキーマが存在する場合)
 - 階層全体に対して REPLACE_CREATE オプションが使用されている場合は、階層内のすべての副表に対する CONTROL 特権
 - REPLACE オプションを使用して既存の階層に IMPORT するには、以下のどれかが必要です。
 - sysadm
 - dbadm
 - 階層内のすべての副表に対する 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
- 行の挿入するラベル、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:

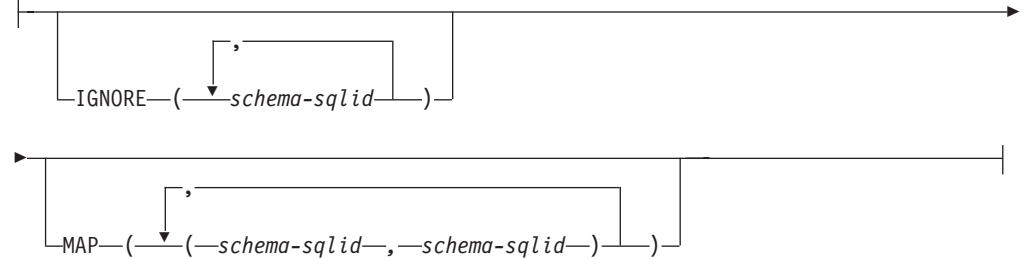
データベース。暗黙接続が可能な場合には、デフォルト・データベースへの接続が確立されます。Linux、UNIX、または Windows クライアントから Linux、UNIX、または Windows データベース・サーバーへのユーティリティ・アクセスは、DB2 Connect ゲートウェイまたはループバック環境を経由してではなく、エンジンを使用した直接接続でなければなりません。

コマンド構文:





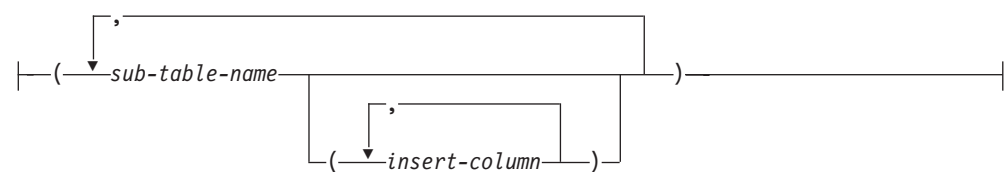
Ignore and Map parameters:

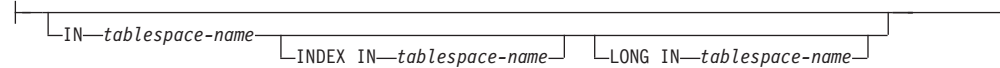


hierarchy description:



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 表ロックが外されるので、コミットの完了後に再びロックの設定が試みられます。ニックネームにインポートするときにはこのパラメーターが必要で、有効な数値を使って COMMITCOUNT を指定する必要があります (AUTOMATIC は有効なオプションとは見なされません)。

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。

IMPORT コマンドが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) を検出した場合、COMMITCOUNT *n* が指定されているなら、**IMPORT** は、無条件のコミットを実行することによって問題解決を試みた後、レコードの挿入または更新を再試行します。これでログ・フル条件が解決しない場合 (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。既にコミットされた行が処理されないようにするには、RESTARTCOUNT または SKIPCOUNT オプションを使用することができます。

CREATE

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 より少ない 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。制約事項のリストの詳細は、『インポートを使用した、エクスポートされる表の再作成』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 文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 文節は、IGNORE 文節および MAP 文節よりも優先されます。XDS が DEFAULT 文節を満たすなら、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 文節は、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 IGNORE のスキーマ・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 文節の中で指定されている場合、MAP 文節のスキーマ・ペアの左辺にそれを含めることはできません。

IGNORE 文節は XDS にのみ適用されます。あるスキーマが IGNORE 文節によって指定されていても、それが MAP 文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ます、REGULAR 表スペースでなければなりません。他の表スペースを指定しない場合、すべての表パーツはこの表スペースに保管されます。この文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 文節で指定される 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。

下位レベル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。修飾子付き表名は、*schema.tablename* の形式です。*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、LOBSINFIL 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 文節で指定した PRIMARY 表スペースが DMS 表スペースである場合のみ使用できます。指定した表スペースは存在している必要があり、LARGE DMS 表スペースでなければなりません。

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 文節は、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 文節には、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 文節のスキーマ・ペアの左辺で指定されている場合、IGNORE 文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ます。そのファイル・タイプに対してのみ有効なオプションです。

N インポートする列の名前を指定します。

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。インポート・ユーティリティ用のファイル・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・マネージャーで使います。
- WSF (ワークシート・フォーマット)。以下のプログラムで使います。
 - Lotus 1-2-3
 - Lotus Symphony
- IXF (統合交換フォーマット、PC バージョン)。同一のあるいは別の DB2 表からエクスポートされたことを意味します。IXF ファイルには、表定義および既存の索引定義も入ります。ただし、SELECT ステートメントに列が指定されている場合は除きます。

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) 文節、あるいは ALTER TABLE ステートメントの ACTIVE NOT LOGGED INITIALLY 文節は考慮されません。

NLI 文節が呼び出される 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 文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 UDB バージョン 7 および DB2 UDB バージョン 8 に適用されます。

REPLACE_CREATE

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。制約事項のリストの詳細は、『インポートを使用した、エクスポートされる表の再作成』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ことを指定します。最初の n 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・ユーティリティーによって警告が生成され、このためにインポートが失敗する可能性があります。 n をゼロにした場合や、このオプションを指定しない場合、発行された警告の回数に関係なくインポート操作は続行します。

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 文節によって 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 文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 文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 つの文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 文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 文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。同様に、MAP 文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 文節のペアの 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 文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 (IDENTITY 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 文節の使用例:

例 1 (XMLVALIDATE USING XDS)

以下の XMLVALIDATE 文節の場合、

```
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 文節の場合、

```
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 文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 文節でスキーマ "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。

例 3 (XMLVALIDATE USING XDS)

以下の XMLVALIDATE 文節の場合、

```
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 文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 文節の場合、

```
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 文節が適用されたため、それ以降 MAP 文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 文節でスキーマ "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 文節でスキーマ "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 文節の場合、

```
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' />
```

ファイルの文書

```
xmlfile.001.xml
```

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 文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 USING SCHEMALOCATION HINTS)

XML 列に以下の XDS が含まれているとすると、

```
<XDS FIL='xmlfile.001.xml' />
```

使用される XML スキーマは、文書の内容として含まれる schemaLocation 属性によって決まります。その属性が指定されていない場合、妥当性検査は実行されません。

XML 列に以下の XDS が含まれているとすると、

```
<XDS FIL='xmlfile.002.xml' SCH='S1.SCHEMA_A' />
```

使用される XML スキーマは、文書の内容として含まれる schemaLocation 属性によって決まります。その属性が指定されていない場合、妥当性検査は実行されません。妥当性検査は SCHEMALOCATION HINTS を使用して実行されるため、SCH 属性は無視されることに注意してください。

使用上の注意:

インポート操作を開始する前に、すべての表操作が完了し、すべてのロックがペンディング解除になっていることを確認してください。これは、WITH HOLD でオープンされた、すべてのカーソルをクローズした後で COMMIT または ROLLBACK を発行することによって行われます。

インポート・ユーティリティーは、SQL INSERT ステートメントを使用してターゲット表に行を追加します。このユーティリティーは、入力ファイル中の各行のデータにつき 1 つずつ INSERT ステートメントを発行します。INSERT ステートメントが失敗した場合、以下の 2 通りの結果のいずれかになります。

- 後続の INSERT ステートメントが成功すると予測される場合には、警告メッセージがメッセージ・ファイルに書き込まれ、処理が継続されます。
- 後続の INSERT ステートメントが失敗すると予測され、データベースが損傷する可能性がある場合には、エラー・メッセージがメッセージ・ファイルに書き込まれ、処理が停止されます。

このユーティリティーは、REPLACE または REPLACE_CREATE 操作時に以前の行が削除された後、自動 COMMIT を実行します。したがって、表オブジェクトが切り捨てられた後、システムに障害が起こったり、アプリケーションがデータベース・マネージャーに割り込んだりすると、元のデータがすべて失われてしまいます。これらのオプションを使用する前に、元のデータがもはや必要ないことを確認してください。

CREATE、REPLACE、または REPLACE_CREATE 操作時にログが満杯になると、このユーティリティーは挿入されたレコードに対して自動 COMMIT を実行します。自動 COMMIT の後に、システムに障害が起こるか、またはアプリケーションがデータベース・マネージャーに割り込むと、部分的にデータの挿入された表はデータベース内に残ります。REPLACE または REPLACE_CREATE オプションを使用してインポート操作全体をやり直すか、または正常にインポートされる行の数に設定した RESTARTCOUNT パラメーターを指定して INSERT を使用してください。

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、参照保全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い (60KB より長い) 場合、クライアント上でユーザーに戻されるメッセージ・ファイルがインポート操作中に欠落することがあります。メッセージ情報の最初の 30KB と最後の 30KB は、常に保存されます。

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セットにあるときよりも、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 を使用すると、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。

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・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ていて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用し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。

関連概念:

- 「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」の『インポートの概要』
- 「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」の『インポートの使用に必要な特権、権限、および許可』

関連タスク:

- 「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」の『データのインポート』

関連資料:

- 「SQL リファレンス 第 1 巻」の『XMLPARSE スカラー関数』
- 「管理 SQL ルーチンおよびビュー」の『ADMIN_CMD プロシージャー - 管理コマンドの実行』
- 156 ページの『db2look - DB2 統計および DDL 抽出ツール』
- 「管理 SQL ルーチンおよびビュー」の『IMPORT コマンド (ADMIN_CMD プロシージャーを使用)』
- 「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」の『インポート・セッション - CLP の例』
- 「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」の『インポートおよびエクスポートに関する LOB および XML ファイルの振る舞い』

INITIALIZE TAPE

ストリーミング磁気テープ装置へのバックアップおよびリストア操作のためにテープを初期化します。このコマンドは Windows オペレーティング・システムでのみサポートされています。

許可:

以下のいずれか。

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

必要な接続:

なし。

コマンド構文:

```
▶▶ INITIALIZE TAPE [ON=device] [USING=blksize] ▶▶
```

コマンド・パラメーター:**ON device**

有効なテープ装置名を指定します。デフォルト値は ¥¥.¥TAPE0 です。

USING blksize

装置のブロック・サイズを指定します (バイト単位)。値が装置のブロック・サイズとしてサポートされている範囲内であれば、装置は指定されたそのブロック・サイズを使用するよう初期化されます。

BACKUP DATABASE コマンドおよび RESTORE DATABASE コマンドで指定されるバッファ・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。

関連資料:

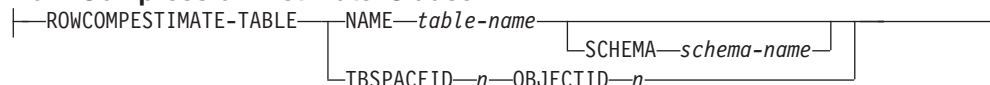
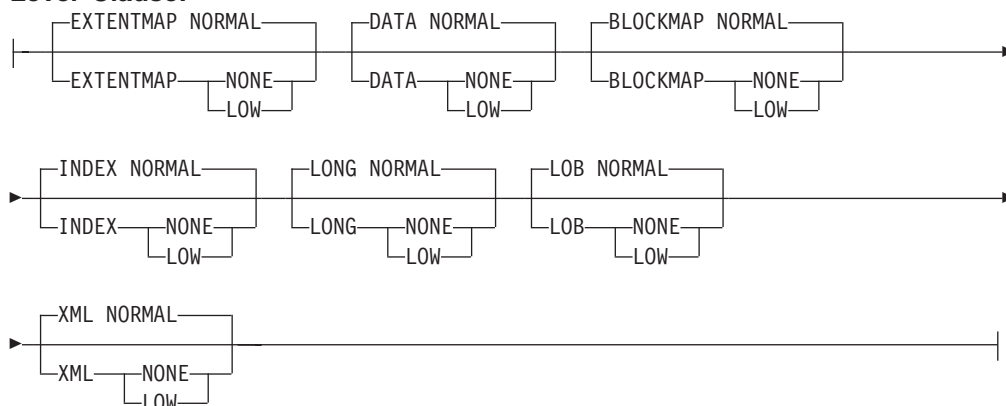
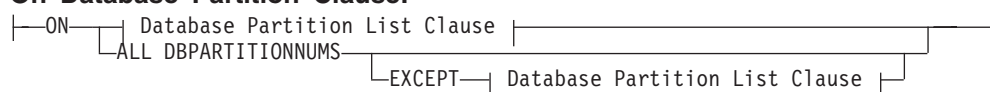
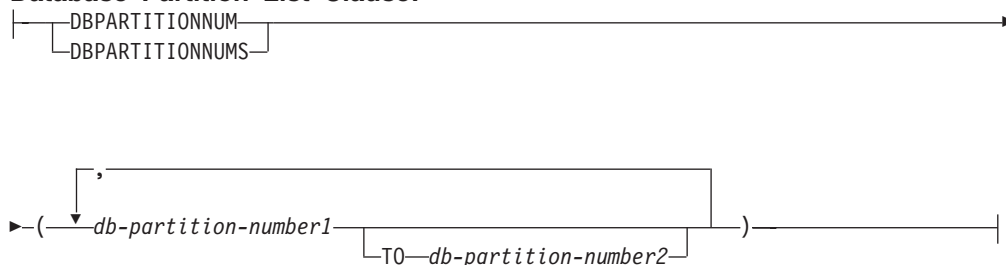
- 382 ページの『BACKUP DATABASE 』

- 738 ページの『RESTORE DATABASE 』
- 755 ページの『REWIND TAPE 』
- 791 ページの『SET TAPE POSITION 』
- 「管理 SQL ルーチンおよびビュー」の『INITIALIZE TAPE コマンド
(ADMIN_CMD プロシージャを使用)』

```

graph LR
    CHECK[CHECK] --- DB[DATABASE]
    CHECK --- TS[TABLESPACE]
    CHECK --- TAB[TABLE]
    DB --- TBSPID1[BEGIN TBSPACEID—n]
    DB --- OBJID1[OBJECTID—n]
    TS --- NAME1[NAME—tablespace-name]
    TS --- TBSPID2[TBSPACEID—n]
    TS --- OBJID2[BEGIN OBJECTID—n]
    TAB --- NAME2[NAME—table-name]
    TAB --- SCHEMA[SCHEMA—schema-name]
    TAB --- TBSPID3[TBSPACEID—n]
    TAB --- OBJID3[OBJECTID—n]
    TBSPID1 --- OBJID1
    TBSPID2 --- OBJID2
    TBSPID3 --- OBJID3
    TBSPID1 --- TBSPID2
    TBSPID2 --- TBSPID3
    TBSPID3 --- TBSPID1
    OBJID1 --- OBJID2
    OBJID2 --- OBJID3
    OBJID3 --- OBJID1
    OBJID1 --- TBSPID2
    OBJID2 --- TBSPID3
    OBJID3 --- TBSPID1
    NAME1 --- OBJID2
    NAME1 --- OBJID3
    NAME1 --- TBSPID2
    NAME1 --- TBSPID3
    NAME2 --- SCHEMA
    NAME2 --- TBSPID3
    NAME2 --- OBJID3
    SCHEMA --- TBSPID3
    SCHEMA --- OBJID3
    TBSPID3 --- OBJID3

```

Row Compression Estimate Clause:**Level 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。

BEGIN 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。

- 圧縮によって節約されるバイトのパーセンテージ
- 圧縮によって節約されるページのパーセンテージ
- データ・サイズが小さいため圧縮には不適格な行のパーセンテージ
- コンプレッション・ディクショナリーのサイズ
- 拡張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保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。

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。ノード文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。

LIMIT ERROR TO *n*

オブジェクトのエラー・ページ数を *n* まで報告できます。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。

LIMIT ERROR TO DEFAULT

オブジェクトのデフォルト・エラー・ページ数まで報告できます。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。

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。

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。

使用上の注意:

1. 表オブジェクトでのチェック操作に関して、処理レベルはオブジェクトに対して指定できます。デフォルトは **NORMAL** レベルです。オブジェクトに **NONE** を指定すると、そのオブジェクトは除外されます。 **LOW** を指定すると、**NORMAL** で行われるチェックのサブセットのチェック操作を行います。
2. 表スペースまたは表を識別する **ID** 値を指定することにより、データベースのチェックを特定の表スペースまたは表から開始するように指定できます。
3. 表を識別する **ID** 値を指定することにより、表スペースのチェックを特定の表から開始するように指定できます。
4. 表スペースの処理は、表スペース内にあるオブジェクトにのみ影響を与えます。
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。

関連資料:

- 「管理 API リファレンス」の『db2Inspect API - データベースの構造上の保全性の検査』

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:



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・システム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 currently	= 0
Database path	= /home/smith/smith/NODE0000/SQL00002/
Database name	= SAMPLE
Applications connected currently	= 1
Database path	= /home/smith/smith/NODE0000/SQL00001/

互換性:

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 DBPARTITIONNUM の代わりに NODE を使用できます。

関連資料:

- 532 ページの『GET SNAPSHOT 』
- 363 ページの『ACTIVATE DATABASE 』
- 454 ページの『DEACTIVATE DATABASE 』

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:

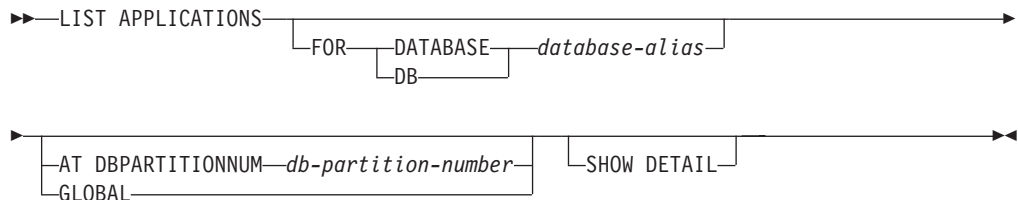
以下のいずれか。

- *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・プログラム名
- アプリケーション・ハンドル
- アプリケーション ID
- データベース名

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・システム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

出力には、以下の追加情報が含まれます。

- 順序番号
- アプリケーションの状況
- 状況変更時刻
- 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。

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 DBPARTITIONNUM の代わりに NODE を使用できます。

関連資料:

- 532 ページの『GET SNAPSHOT 』
- 484 ページの『FORCE APPLICATION 』
- 「管理 SQL ルーチンおよびビュー」の『APPLICATIONS 管理ビュー – 接続されているデータベース・アプリケーション情報の検索』
- 「管理 SQL ルーチンおよびビュー」の『FORCE APPLICATION コマンド (ADMIN_CMD プロシーチャーを使用)』

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
-d	XML declarations	OFF
-e	Display SQLCODE/SQLSTATE	OFF
-f	Read from input file	OFF
-l	Log commands in history file	OFF
-n	Remove new line character	OFF
-o	Display output	ON
-p	Display interactive input prompt	ON
-r	Save output to report file	OFF
-s	Stop execution on command error	OFF
-t	Set statement termination character	OFF
-v	Echo current command	OFF
-w	Display FETCH/SELECT warning messages	ON
-z	Save all output to output file	OFF

関連資料:

- 840 ページの『UPDATE COMMAND OPTIONS 』

LIST DATABASE DIRECTORY

システム・データベース・ディレクトリーの内容をリスト表示します。パスが指定されている場合、ローカル・データベース・ディレクトリーの内容が表示されます。

有効範囲:

ON *path* パラメーターを指定しないでこのコマンドを発行すると、システム・データベース・ディレクトリーが戻されます。この情報はすべてのデータベース・パーティションで同じです。

ON *path* パラメーターを指定すると、指定したパスのローカル・データベース・ディレクトリーが戻されます。この情報はデータベース・パーティションによって異なります。

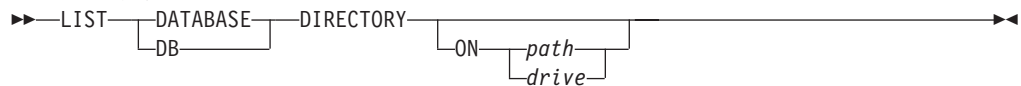
許可:

なし

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:

ON *path/drive*

情報を表示するローカル・データベース・ディレクトリーを指定します。これを指定しない場合、システム・データベース・ディレクトリーの内容が表示されます。インスタンス名は、パスに暗黙指定されることに注意してください。パスの一部としてインスタンス名を指定しないでください。

例:

次に示すのは、システム・データベース・ディレクトリーの場合の出力例です。

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
----------------	------------

LIST DATABASE DIRECTORY

```
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。多くの場合、この名前はデータベース作成時点での名前です。

Local Database Directory

データベースが存在しているパス。このフィールドが表示されるのは、システム・データベース・ディレクトリーがスキャンされた場合だけです。

Database directory

データベースが置かれているディレクトリーの名前。このフィールドが表示されるのは、ローカル・データベース・ディレクトリーがスキャンされた場合だけです。

Node name

リモート・ノードの名前。この名前は、データベースおよびノードのカタログ時に *nodename* パラメーターに入力した値に対応します。

Database release level

データベースに対して実行可能なデータベース・マネージャーのリリース・レベル。

Comment

データベースをカタログした時点で入力された、データベースに関連する注釈。

Directory entry type

データベースの存在ロケーション。

- *remote* 項目には、別のノードにあるデータベースについて記述されます。
- *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思なされるため、間接 (indirect) 項目と思なされます。
- *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。

LIST DATABASE DIRECTORY

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。

関連資料:

- 430 ページの『CHANGE DATABASE COMMENT 』
- 435 ページの『CREATE DATABASE 』
- 「管理 API リファレンス」の『db2DbDirGetNextEntry API - システム・データベース・ディレクトリー、あるいはローカル・データベース・ディレクトリー次項目の取得』
- 834 ページの『UPDATE ALTERNATE SERVER FOR DATABASE』

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:

```

▶▶—LIST DATABASE PARTITION GROUPS—┐
                                   └─SHOW DETAIL─┐

```

コマンド・パラメーター:

SHOW DETAIL

以下の情報を出力に含めることを指定します。

- 分散マップ ID
- データベース・パーティション番号
- 使用中フラグ

例:

以下に示すのは、LIST DATABASE PARTITION GROUPS コマンドの出力例です。

```

DATABASE PARTITION GROUP NAME
-----
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。

LIST DATABASE PARTITION GROUPS

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。

関連資料:

- 692 ページの『REDISTRIBUTE DATABASE PARTITION GROUP』

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/sql11ib/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 DBPARTITIONNUMS の代わりに NODES を使用できます。

関連資料:

- 692 ページの『REDISTRIBUTE DATABASE PARTITION GROUP 』

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:



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。

関連資料:

- 484 ページの『FORCE APPLICATION 』

582 コマンド・リファレンス

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分散データベース接続サービス・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
- iSeries 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

関連資料:

- 「管理 API リファレンス」の『sqlegdgt API - データベース接続サービス (DCS) ディレクトリーの項目の取得』
- 412 ページの『CATALOG DCS DATABASE 』

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:

sysadm

必要な接続:

インスタンス

コマンド構文:

▶—LIST DRDA INDOUBT TRANSACTIONS—▶
 |WITH PROMPTING|

コマンド・パラメーター:

WITH PROMPTING

未確定トランザクションを処理することを指定します。このパラメーターを指定すると、対話式ダイアログ・モードが開始され、未確定トランザクションのコミットまたはロールバックが可能になります。このパラメーターを指定しない場合、未確定トランザクションは標準出力装置に書き込まれ、対話式ダイアログ・モードは開始されません。

破棄 (forget) オプションはサポートされていません。未確定トランザクションをコミットまたはロールバックすると、そのトランザクションは自動的に破棄されます。

対話式ダイアログ・モードでは次のことが可能です。

- すべての未確定トランザクションのリスト表示 (1 を入力)
- 未確定トランザクション番号 *x* のリスト表示 (1 の後に有効なトランザクション番号を入力)
- 終了 (q を入力)
- トランザクション番号 *x* をコミット (c の後に有効なトランザクション番号を入力)
- トランザクション番号 *x* をロールバック (r の後に有効なトランザクション番号を入力)

コマンド文字と引数の間は、ブランク・スペースで区切る必要があります。

トランザクションのコミットまたはロールバックを実行する前に、トランザクション・データが表示され、アクションを確認するように求められます。

使用上の注意:

DRDA 未確定トランザクションが発生するのは、分散作業単位内のコーディネーターと参加者との間の通信が失われた場合です。分散作業単位では、ユーザーやアプリケーションが、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。

関連タスク:

- 「管理ガイド: プランニング」の『未確定トランザクションの手動での解決』

関連資料:

- 「管理 API リファレンス」の『sqlcspqy API - DRDA 未確定トランザクションのリスト』
- 「管理 API リファレンス」の『sqlxhfrg API - トランザクション状況の忘却』
- 「管理 API リファレンス」の『sqlxphcm API - 未確定トランザクションのロールバック』
- 「管理 API リファレンス」の『sqlxphrl API - 未確定トランザクションのロールバック』

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。

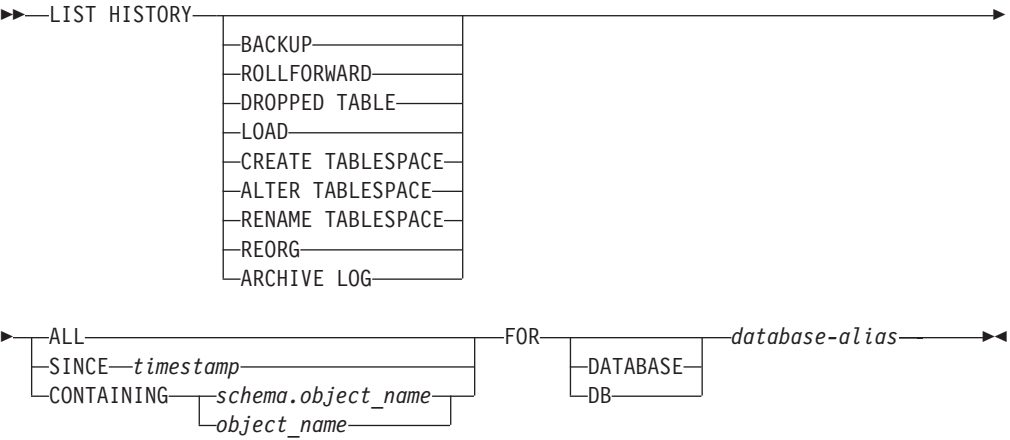
許可:

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 - ロードのコピー・ファイル取得
- 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

LIST HISTORY

X - アーカイブ・ログ

種類

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

関連概念:

- 「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『バックアップとリカバリーの計画の作成』

関連資料:

- 「管理 *SQL* ルーチンおよびビュー」の『DB_HISTORY 管理ビュー - 履歴ファイル情報の検索』

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)

LIST INDOUBT TRANSACTIONS

- 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それともロールバックするかがわかりません。未確定トランザクションは MPP 環境でも発生する可能性があります。

現在接続されているデータベースに対して LIST INDOUBT TRANSACTIONS が出された場合、そのデータベースの未確定トランザクションに関する情報が戻されます。

コミットできるのは、状況が未確定 (i)、またはコミット肯定応答欠落 (m)、またはフェデレーテッド・コミット肯定応答欠落 (d) のトランザクションだけです。

ロールバックできるのは、状況が未確定 (i)、フェデレーテッド・ロールバック肯定応答欠落 (b)、または終了済み (e) のトランザクションだけです。

取り消しができるのは、状況がコミット済み (c)、ロールバック (r)、フェデレーテッド・コミット肯定応答欠落 (d)、またはフェデレーテッド・ロールバック肯定応答欠落 (b) のトランザクションだけです。

2 フェーズ・コミットのコミット・フェーズでは、コーディネーター・ノードがコミットの肯定応答を待機します。(ノード障害などの理由で) 応答しないノードが 1 つ以上ある場合、そのトランザクションはコミット肯定応答欠落状態になります。

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。

関連概念:

- 「管理ガイド: プランニング」の『XA トランザクション・マネージャーの構成に関する考慮事項』

関連タスク:

- 「管理ガイド: プランニング」の『未確定トランザクションの手動での解決』

関連資料:

- 「管理 API リファレンス」の『db2XaListIndTrans API - 未確定トランザクションのリスト』
- 「管理 API リファレンス」の『sqlxhfrg API - トランザクション状況の忘却』
- 「管理 API リファレンス」の『sqlxphcm API - 未確定トランザクションのロールバック』
- 「管理 API リファレンス」の『sqlxphrl API - 未確定トランザクションのロールバック』

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:

```

▶▶—LIST—┐—NODE DIRECTORY—┐————▶▶
          └—ADMIN—┘           └—SHOW DETAIL—┘

```

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:

ノードは、個々のデータベース・クライアント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

LIST NODE DIRECTORY

- 名前付きパイプ
- TCPIP
- TCPIP4
- TCPIP6

関連資料:

- 「管理 API リファレンス」の『sqlengne API - ノード・ディレクトリー次項目の入手』
- 419 ページの『CATALOG LDAP NODE 』
- 421 ページの『CATALOG LOCAL NODE 』
- 423 ページの『CATALOG NAMED PIPE NODE 』
- 426 ページの『CATALOG TCPIP/TCPIP4/TCPIP6 NODE 』

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。

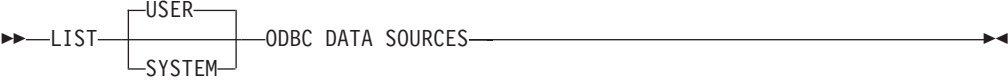
許可:

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

関連資料:

- 425 ページの『CATALOG ODBC DATA SOURCE 』
- 824 ページの『UNCATALOG ODBC DATA SOURCE 』

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:

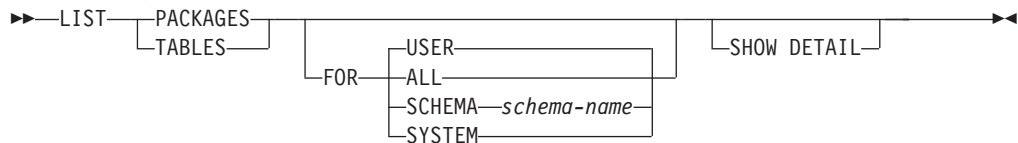
システム・カタログ SYSCAT.PACKAGES (LIST PACKAGES) と SYSCAT.TABLES (LIST TABLES) の場合には、少なくとも以下の 1 つが必要です。

- *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 文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

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 コマンドと 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
```

LIST PACKAGES/TABLES

```
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;
```

関連概念:

- 「*SQL* リファレンス 第1巻」の『カタログ・ビュー』
- 「パフォーマンス」の『効率的な SELECT ステートメント』

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 コマンドを使います。

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 の出力例です。

```
Tablespace Containers for Tablespace 0
```

```
Container ID                      = 0
```

LIST TABLESPACE CONTAINERS

Name	= /home/smith/smith/NODE0000/ SQL00001/SQLT0000.0
Type	= Path

次に示すのは、SHOW DETAIL を指定した場合の LIST TABLESPACE CONTAINERS 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

関連概念:

- 「システム・モニター ガイドおよびリファレンス」の『スナップショット・モニター』

関連資料:

- 「管理 API リファレンス」の『sqlbtcq API - すべての表スペース・コンテナの照会データの取得』
- 601 ページの『LIST TABLESPACES 』
- 「管理 SQL ルーチンおよびビュー」の『CONTAINER_UTILIZATION 管理ビュー - 表スペース・コンテナおよび使用率に関する情報の検索』
- 「管理 SQL ルーチンおよびビュー」の『TBSP_UTILIZATION 管理ビュー - 表スペースの構成および使用率に関する情報の検索』

LIST TABLESPACES

現行データベースの表スペース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*
- *load*

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

LIST TABLESPACES

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
0x2000000	Storage may be defined
0x4000000	StorDef is in 'final' state
0x8000000	StorDef was changed prior to rollforward
0x10000000	DMS rebalancer is active
0x20000000	TBS deletion in progress
0x40000000	TBS creation in progress
0x8	For service use only

このオプションを指定した場合は、各表スペースに関して下記の付加的な情報が表示されます。

- ページの合計数
- 使用できるページの数
- 使用されたページの数
- 未使用ページの数
- 最高水準点 (ページ単位)
- ページ・サイズ (バイト単位)
- エクステンツ・サイズ (ページ単位)
- プリフェッチ・サイズ (ページ単位)
- コンテナの数
- 最小リカバリー時間 (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 quiescers              = 1
    Quiescer 1:
        Tablespace ID             = 2
        Object ID                 = 3

```

DB21011I In a partitioned database server environment, only the table spaces on the current node are listed.

Tablespaces for Current Database

```

Tablespace ID                    = 0
Name                             = SYSCATSPACE
Type                             = System managed space
Contents                         = Any data
State                            = 0x0000
    Detailed explanation:
        Normal
Total pages                      = 1200
Useable pages                    = 1200
Used pages                       = 1200
Free pages                       = Not applicable
High water mark (pages)          = Not applicable
Page size (bytes)                = 4096
Extent size (pages)              = 32
Prefetch size (pages)            = 32
Number of containers              = 1
Tablespace ID                    = 1
Name                             = TEMPSPACE1
Type                             = System managed space
Contents                         = Temporary data

```

LIST TABLESPACES

```

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。

関連資料:

- 「管理 API リファレンス」の『sqlbmtsq API - すべての表スペースの照会データの取得』
- 599 ページの『LIST TABLESPACE CONTAINERS 』
- 308 ページの『db2tbst - 表スペース状態の獲得』
- 「管理 SQL ルーチンおよびビュー」の『CONTAINER_UTILIZATION 管理ビュー - 表スペース・コンテナおよび使用率に関する情報の検索』
- 「管理 SQL ルーチンおよびビュー」の『TBSP_UTILIZATION 管理ビュー - 表スペースの構成および使用率に関する情報の検索』

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、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*

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。

関連資料:

- 792 ページの『SET UTIL_IMPACT_PRIORITY』
- 「管理 SQL ルーチンおよびビュー」の『SNAPUTIL 管理ビューおよび SNAP_GET_UTIL 表関数 - utility_info 論理データ・グループ・スナップショット情報の検索』
- 「管理 SQL ルーチンおよびビュー」の『SNAPUTIL_PROGRESS 管理ビューおよび SNAP_GET_UTIL_PROGRESS 表関数 - progress 論理データ・グループ・スナップショット情報の検索』

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。

制約事項:

ロード・ユーティリティでは、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ロード権限と以下のもの
 - 表の INSERT 特権 (ロード・ユーティリティが 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必要ありません。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とできません。

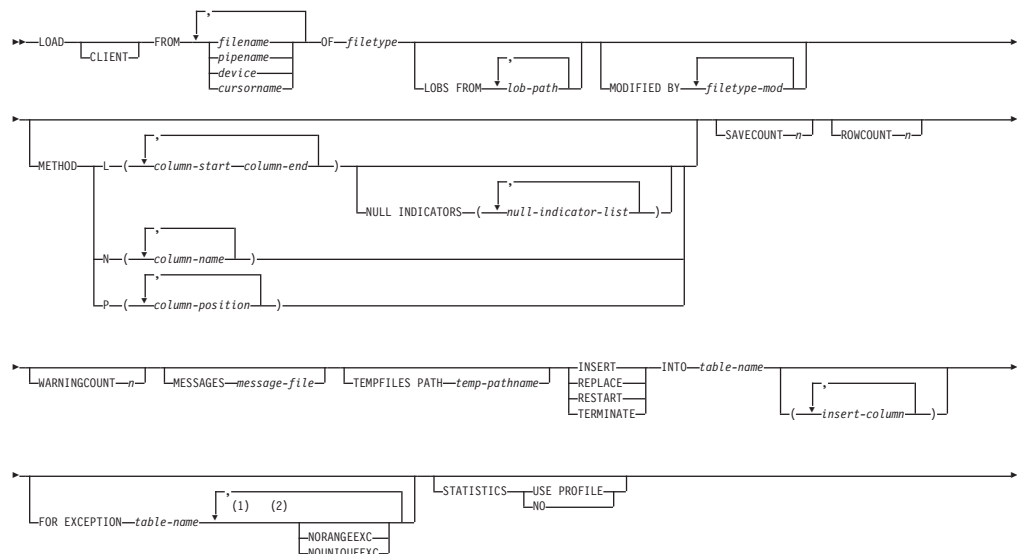
すべてのロード・プロセス (および一般にすべての DB2 サーバー・プロセス) はインスタンス所有者によって所有されており、それらのプロセスすべてにおいて、必要なファイルにアクセスするためにそのインスタンス所有者の ID を使用するため、インスタンス所有者には入力データ・ファイルに対する読み取りアクセス権が必要です。このコマンドをだれが呼び出すかには関係なく、それらの入力データ・ファイルをインスタンス所有者から読むことができればなりません。

必要な接続:

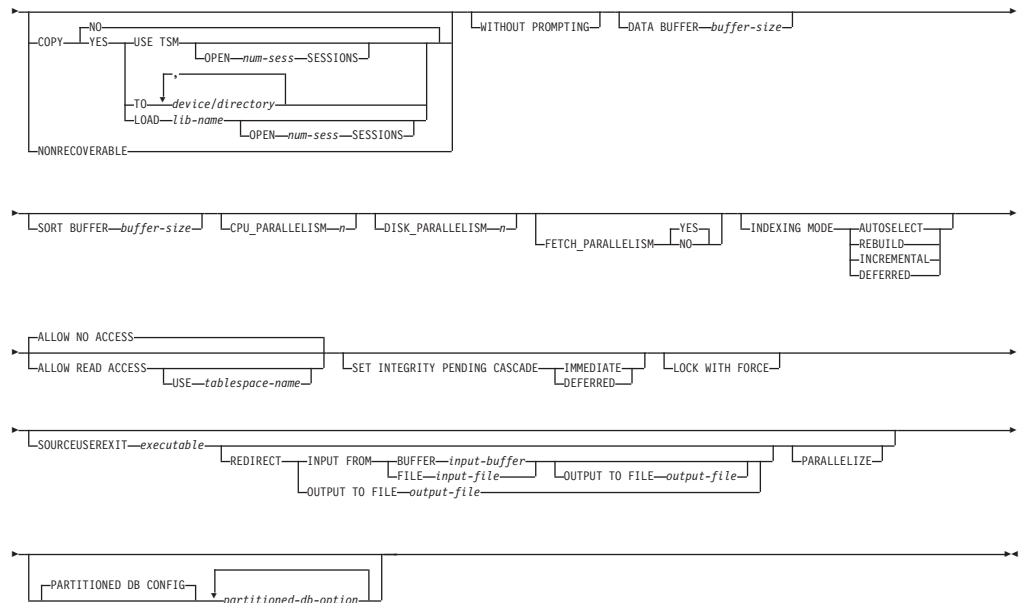
データベース。暗黙接続が可能な場合には、デフォルト・データベースへの接続が確立されます。

インスタンス。明示的なアタッチは必要ありません。データベースへの接続が確立されている場合には、ローカル・インスタンスへの暗黙的な接続が試みられます。

コマンド構文:



LOAD



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いければ、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. EXPORT コマンド (ADMIN_CMD プロシージャを使用) を使用してデータをファイルにエクスポートした場合、そのデータ・ファイルは fenced ユーザー ID によって所有されます。このファイルは通常、インスタンス所有者がアクセスすることはできません。LOAD を CLP または ADMIN_CMD プロシージャから実行するには、インスタンス所有者 ID はデータ・ファイルにアクセスできなければならないので、データ・ファイルに対する読み取りアクセス権限をインスタンス所有者に付与する必要があります。
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)。同一のあるいは別の DB2 表からエクスポートされたことを意味します
- 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。パスの最後は斜線 (/) 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 filetype-mod

ファイル・タイプ修飾子オプションを指定します。『ロード・ユーティリティー用のファイル・タイプ修飾子』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。

SAVECOUNT *n*

ロード・ユーティリティが *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・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、15KB のオーバーヘッド。INSERT オプションを指定した場合で、表の中に多量の長フィールドまたは LOB データが既にある場合には、この数値はこれよりもかなり大きくなる場合があります。

INSERT

ロード・ユーティリティーを実行できる 4 つのモードのうちの 1 つ。既存の表データを変更することなく、ロードされたデータを表に追加します。

REPLACE

ロード・ユーティリティーを実行できる 4 つのモードのうちの 1 つ。表から既存データをすべて削除し、ロードされたデータを挿入します。表定義および索引定義は変更されません。階層間でデータを移動する際にこのオプションを使用する場合は、階層全体に関係したデータだけが置き換えられます。副表は置き換えられません。

RESTART

ロード・ユーティリティーを実行できる 4 つのモードのうちの 1 つ。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。

TERMINATE

ロード・ユーティリティーを実行できる 4 つのモードのうちの 1 つ。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ロード REPLACE の場合、その表はロード 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ロード INSERT の場合、その表はロード TERMINATE 操作完了後も元のレコードをすべて保持します。

ロード終了オプションでは、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。

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。修飾子付き表名は、*schema.tablename* の形式です。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。これがデフォルトです。

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

LOAD

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 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。順方向リカバリーが使用禁止 (つまり *logretain* と *userexit* が両方ともオフ) であれば、このオプションは無効です。

USE TSM

Tivoli Storage Manager (TSM) を使ってコピーを保管することを指定します。

OPEN num-sess SESSIONS

TSM またはベンダー製品とともに使用する入出力セッションの数です。デフォルト値は 1 です。

TO device/directory

コピー・イメージを作成する先の装置またはディレクトリーを指定します。

LOAD lib-name

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

NONRECOVERABLE

ロード・トランザクションがリカバリー不能としてマークされており、それ以降のロールフォワード・アクションによってそれをリカバリーさせることは不可能であることを指定します。ロールフォワード・ユーティリティーは、そのトランザクションをスキップし、データのロード先の表に "invalid" (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。

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 4KB ページ数を設定します (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・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。

値を指定しないと、実行時にユーティリティによって適切なデフォルトが計算されます。デフォルトは、ローダーのインスタンス生成時にユーティリティ・ヒープで利用できるフリー・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・ユーティリティによって起動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、パーティション内並列処理を活用するために設計されています。これは、事前にソートされたデータをロードする際に役立ちます (ソース・データのレコード順序が保持されるため)。このパラメーターの値が 0 の場合や、このパラメーターを指定しなかった場合、ロード・ユーティリティは、実行時に自動的に計算された適切なデフォルト値 (通常は使用できる CPU の数に基づく) を使用します。

注:

1. LOB または LONG VARCHAR フィールドのどちらかの入った表でこのパラメーターを使用する場合、システムの CPU の数またはユーザーが指定した値には関係なく、値は 1 になります。
2. SAVECOUNT パラメーターに指定する値が小さいと、データと表のメタデータの両方をフラッシュするために、ローダーがさらに多くの入出力操作を実行することになります。CPU_PARALLELISM が 1 より大きいなら、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生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確保できます。

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とき、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保天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。

LOAD

ALLOW NO ACCESS 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 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

複数のユーザー出口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、『カスタマイズしたアプリケーション (ユーザー出口) を使用したデータの移動』を参照してください。

PARTITIONED DB CONFIG

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は、『パーティション・データベース環境でのロード構成オプション』に説明されています。

RESTARTCOUNT

予約済み。

USING directory

予約済み。

例:

例 1

TABLE1 に以下の 5 つの列があるとしてします。

- 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 6 つのエレメントがあるとしてします。

- ELE1、位置 01 から 20
- ELE2、位置 21 から 22
- ELE5、位置 23 から 23
- ELE3、位置 24 から 27
- ELE4、位置 28 から 31
- ELE6、位置 32 から 32
- ELE6、位置 33 から 40

データ・レコードは以下のとおりです。

```

1...5...10...15...20...25...30...35...40
Test data 1      XXN 123abcdN
Test data 2 and 3 QQY   wxyzN
Test data 4,5 and 6 WWN6789   Y

```

以下のコマンドは、ファイルから表をロードします。

```

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

```

注:

1. MODIFIED BY パラメーターで striptblanks を指定すると、VARCHAR 列の中のブランクが切り捨てられるようになります (例えば行 1、2、および 3 の長さがそれぞれ 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 32KB 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、

LOAD

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)  
db2load 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 (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
```

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。
- ロード・ユーティリティは、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。ユーティリティは、

参照保全を強制したり、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。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。

関連概念:

- 「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」の『ロードの概要』
- 「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」の『ロードの使用に必要な特権、権限、および許可』

関連タスク:

- 「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」の『データのロード』

関連資料:

- 673 ページの『**QUIESCE TABLESPACES FOR TABLE**』
- 「管理 **SQL** ルーチンおよびビュー」の『**LOAD** コマンド (**ADMIN_CMD** プロシージャーを使用)』
- 「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」の『ロード - **CLP** の例』

- 「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」の『パーティション・データベース環境でのロード構成オプション』

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。

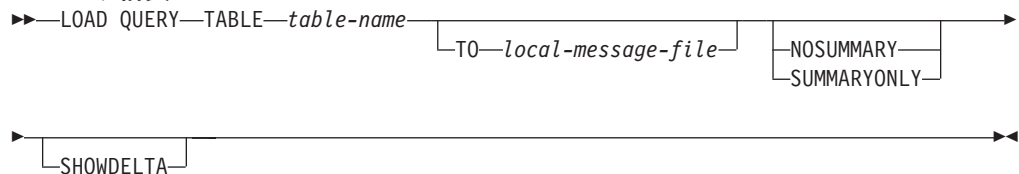
許可:

なし

必要な接続:

データベース

コマンド構文:



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 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 <database>
db2 load query table staff to /u/mydir/staff.tempsmsg

```


出力ファイル /u/mydir/staff.tempsmsg 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** 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** 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。

正常 表の状態は表には影響を及ぼしません。

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、保全性の設定ペンディ

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、表を保全性の設定ペンディング (Set Integrity Pending) 状態にします。

ロード進行中

この表では、現在ロードが進行中です。

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。

読み取りアクセス専用

表データを読み取りアクセス照会に使用することができます。ALLOW READ ACCESS オプションを使用するロード操作では、表は読み取り専用アクセス状態になります。

REORG ペンディング

REORG 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。

ロード操作の進行状況は、LIST UTILITIES コマンドを使ってモニターすることもできます。

関連概念:

- 「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」の『ロードの概要』
- 「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」の『LOAD QUERY コマンドを使用したパーティション・データベース環境でのロード操作のモニター』

- 「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」の『表ロック、表状態、および表スペース状態』

関連資料:

- 606 ページの『LIST UTILITIES』

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、インスタンス移行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。移行作業、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。

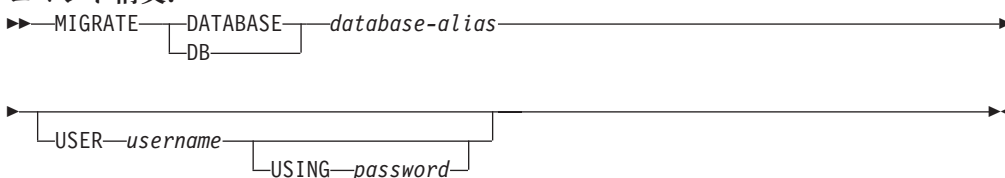
許可:

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移行するデータベースの別名を指定します。

USER username

データベースを移行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移行します。

```
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移行するだけであり、移行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。

移行の前にデータベースをカタログする必要があります。

移行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、移行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: データベースの移行は失敗しました。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移行が失敗した場合、**CLP** はそのデータベース・ディレクトリー・キャッシュをリフレッシュする必要があります (「ログがいっぱい」エラーが戻される場合にこのようになる可能性があります)。

関連タスク:

- 「マイグレーション・ガイド」の『データベースの移行』

関連資料:

- 815 ページの『**TERMINATE**』
- 「管理 API リファレンス」の『**sqlcmgdb** API - 前のバージョンの DB2 データベースの現行バージョンへのマイグレーション』

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。

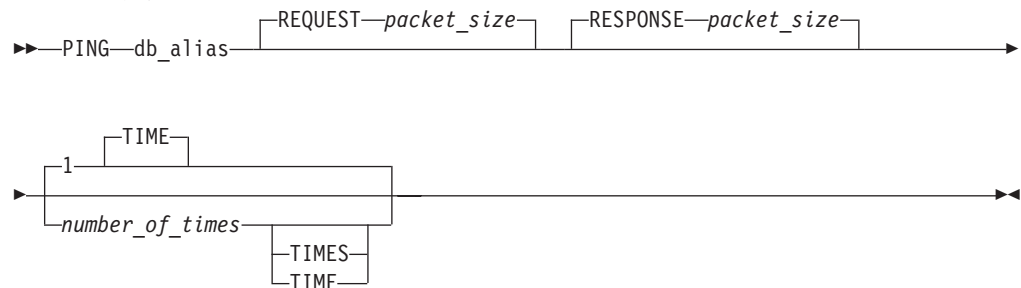
許可:

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 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。

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する必要があります。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、DB2 クライアントと 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。

関連資料:

- 「管理 API リファレンス」の『db2DatabasePing API - ネットワーク応答時間のテストのためのデータベースの ping』

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。

有效範圍:

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` ファイル中のどのデータベース・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** 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)

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。

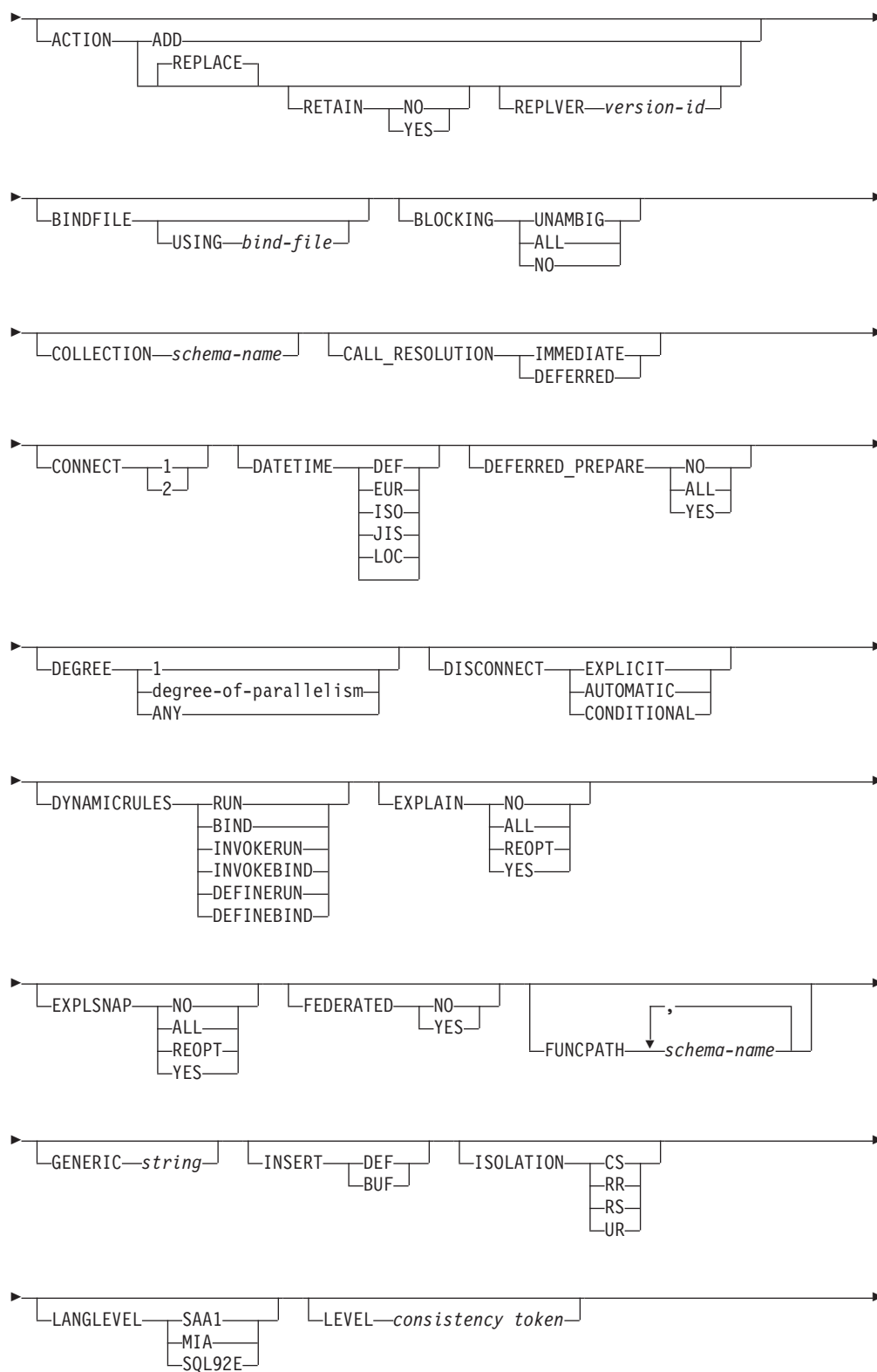
必要な接続:

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。

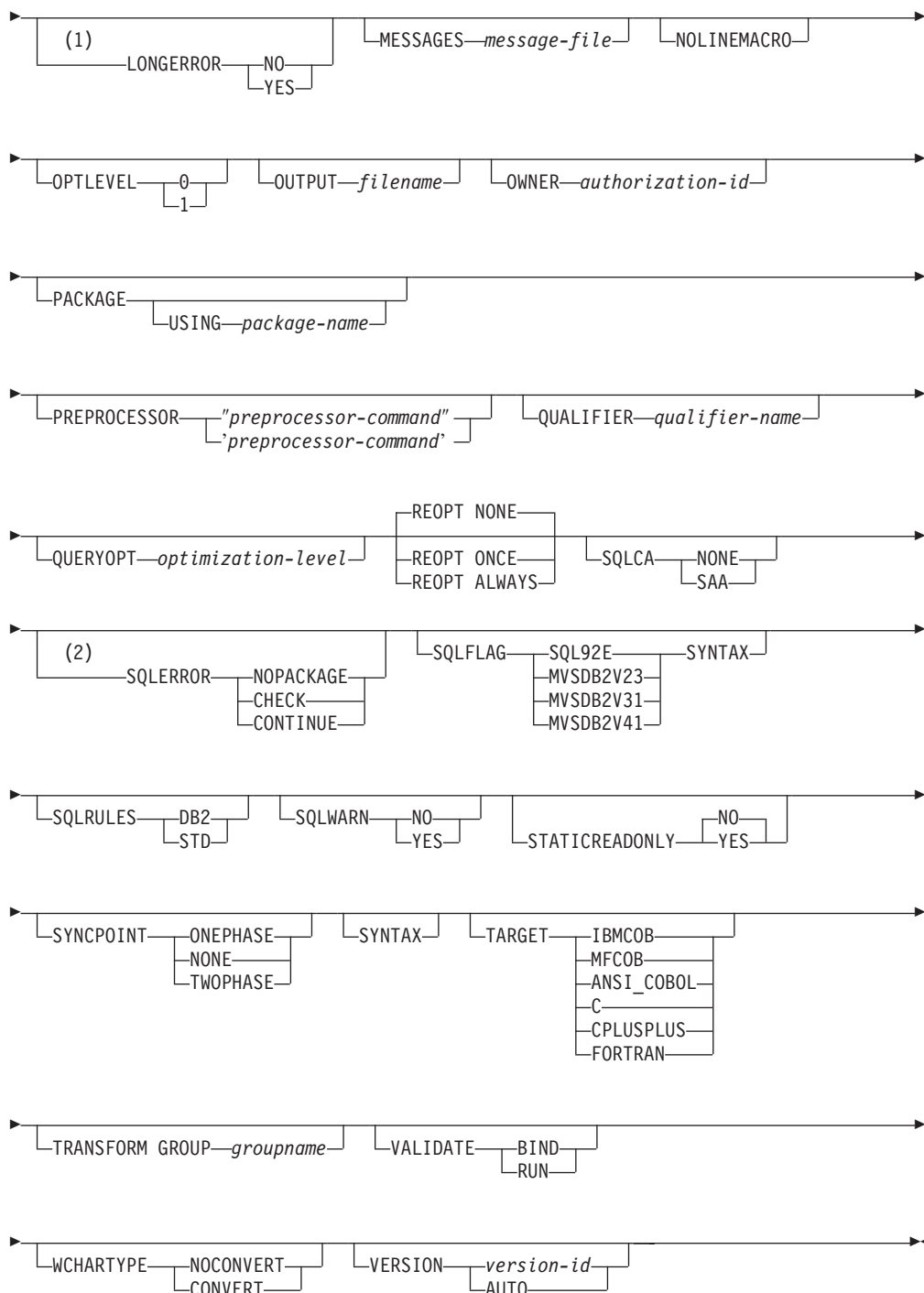
コマンド構文:

DB2 for Windows および DB2 for UNIX では

PRECOMPILE *filename*



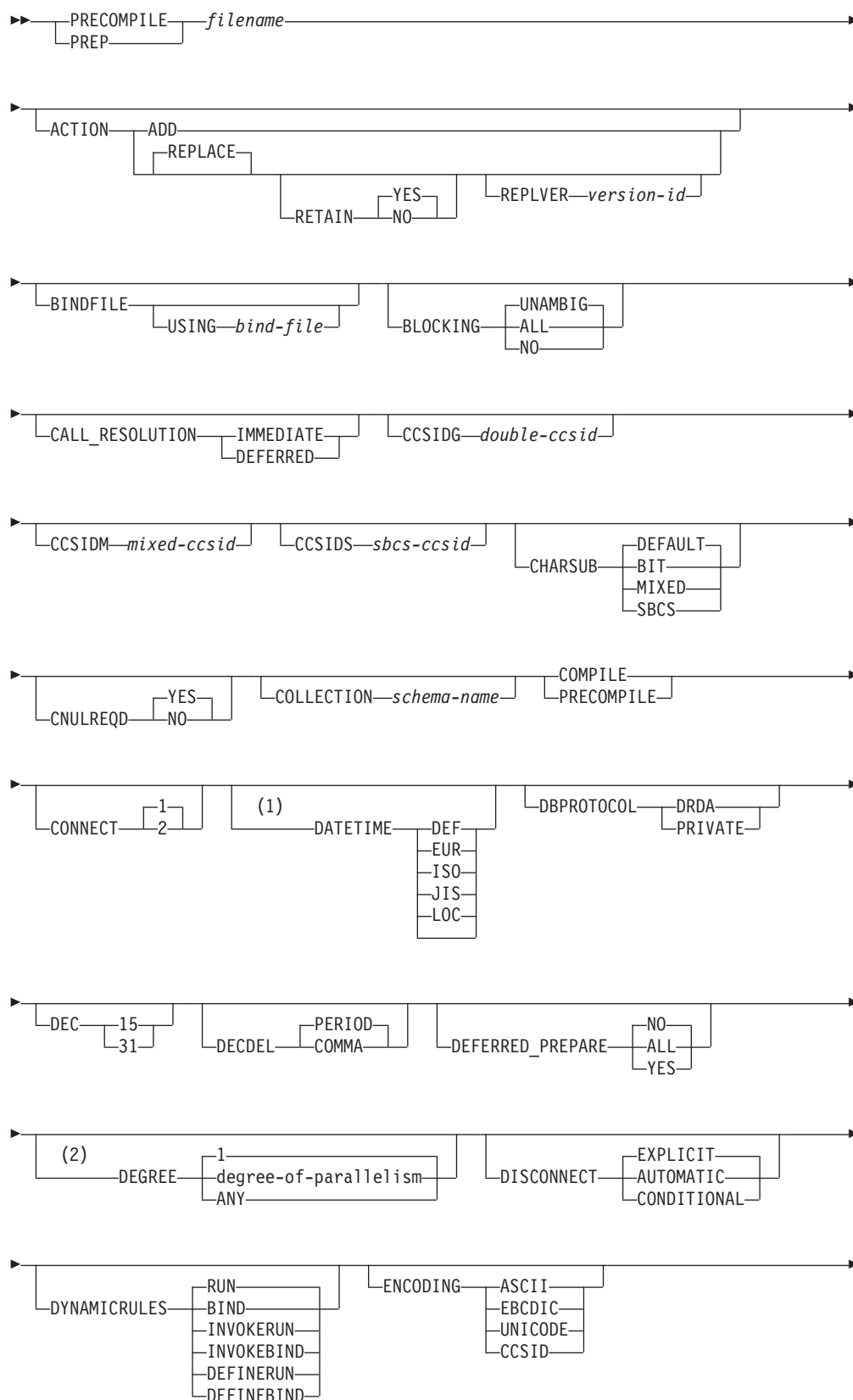
PRECOMPILE



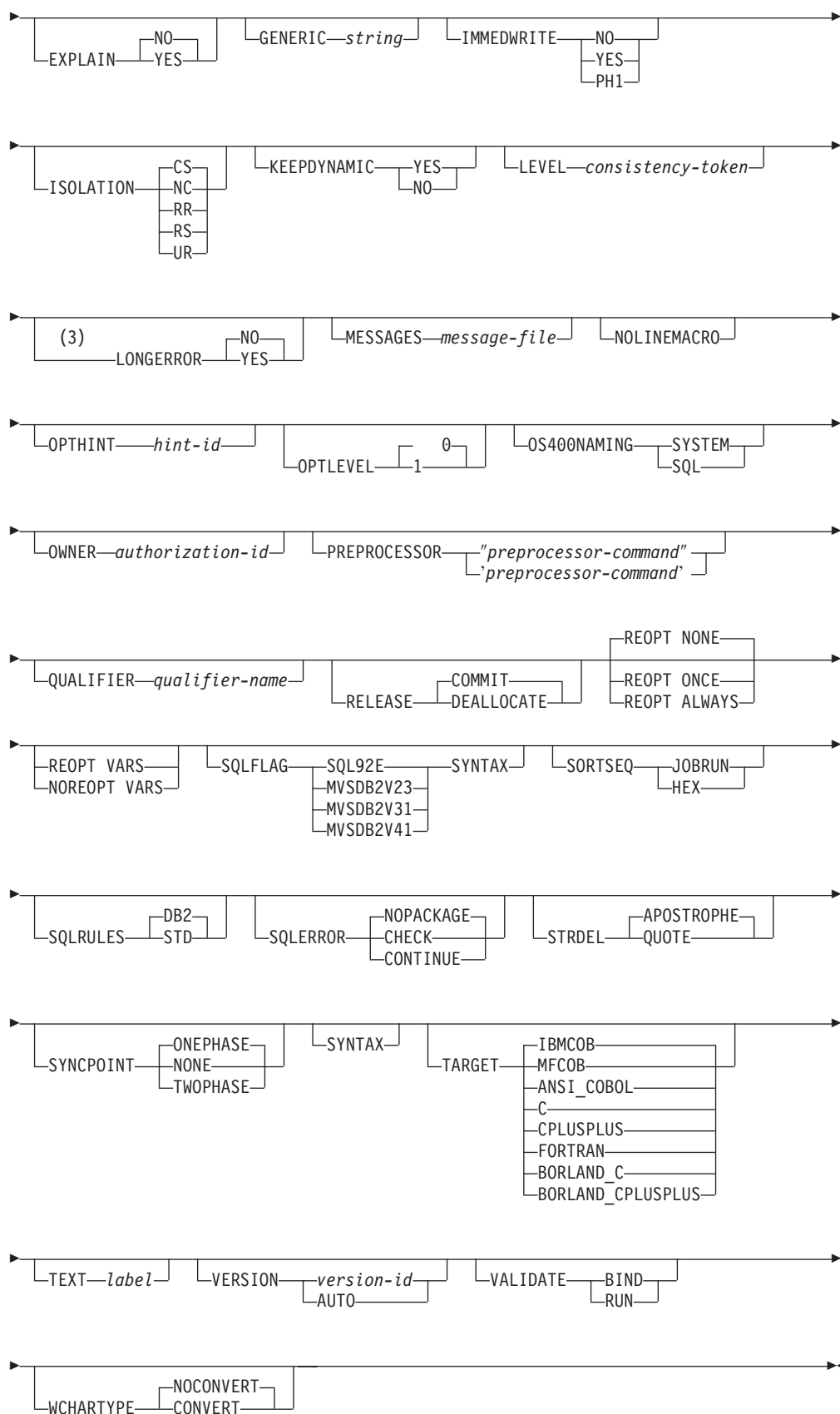
注:

- 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。

Windows および UNIX 以外のサーバー上の DB2 では



PRECOMPILE



注:

- 1 サーバーが DATETIME DEF オプションをサポートしない場合、それは DATETIME ISO にマップされます。
- 2 DEGREE オプションは DRDA レベル 2 のアプリケーション・サーバーでしかサポートされていません。
- 3 NO は、32 ビット・システムと 64 ビット 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 ベース・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 ベース・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 ベース・システムの場合は .f ファイルが生成されます)。

UNIX ベース・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 ベース・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ク化のタイプを指定します。

ALL 次のカーソルをブロック化することを指定します。

- 読み取り専用カーソル。
- FOR UPDATE OF と指定されていないカーソル。

未確定カーソルは、読み取り専用として扱われます。

NO どのカーソルもブロック化しないことを指定します。あいまいなカーソルは、更新可能として扱われます。

UNAMBIG

次のカーソルをブロック化することを指定します。

- 読み取り専用カーソル。

- FOR UPDATE OF と指定されていないカーソル。

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 文節は使用しない) を指定する整数。このオプションは、DB2 Database for Linux, UNIX, and Windows ではサポートされていません。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 文節は使用しない) を指定する整数。このオプションは、DB2 Database for Linux, UNIX, and Windows ではサポートされていません。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 sbcs-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 文節は使用しない) を指定する整数。このオプションは、DB2 Database for Linux, UNIX, and Windows ではサポートされていません。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・オプションに関連します。これは、C または C++ アプリケーションで作成されたバインド・ファイルの場合のみ有効です。この DRDA BIND オプションは、DB2 ではサポートされていません。

NO C スtring・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tring・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 30 文字の収集 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・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・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す。

YES PREPARE ステートメントの実行は、対応する OPEN、DESCRIBE、または EXECUTE ステートメントが発行されるまで据え置かれます。

PREPARE ステートメントは、INTO 文節 (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・マーカー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 文節を使用する場合、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 ~ 32 767 です。

ANY ステートメントの実行時にデータベース・マネージャーで判別した程度で並列処理を行うよう指定します。

DISCONNECT

AUTOMATIC

コミット時にすべてのデータベース接続を切断するよう指定します。

CONDITIONAL

RELEASE をマークしたか、またはオープン状態の WITH HOLD カーソルをもたないデータベース接続を、コミット時に切断するよう指定します。

EXPLICIT

RELEASE ステートメントで明示的に解放をマークしたデータベース接続だけを、コミット時に切断するよう指定します。

DYNAMICRULES

許可 ID に使用される値の初期設定、および非修飾オブジェクト参照の暗黙的な修飾の、実行時の動的 SQL に適用される規則を定義します。

RUN パッケージを実行するユーザーの許可 ID が動的 SQL ステートメントの許可検査に使用されるように指定します。許可 ID は、動的 SQL ステートメント内の非修飾オブジェクト参照を暗黙的に修飾するためのデフォルトのパッケージ修飾子としても使用されます。これはデフォルト値です。

BIND 許可および修飾の静的 SQL に適用されるすべての規則が、実行時に使用されるように指定します。つまり、パッケージ所有者の許可 ID が動的 SQL の許可検査に使用され、デフォルトのパッケージ修飾子が動的 SQL ステートメント内の非修飾オブジェクト参照の暗黙的な修飾に使用されます。

DEFINERUN

パッケージがルーチン・コンテキスト内で使用される場合、ルーチン定義者の許可 ID が許可検査およびルーチン内の動的 SQL ステートメント内の非修飾オブジェクト参照の暗黙的な修飾に使用されます。

パッケージがスタンドアロン・アプリケーションとして使用される場合、動的 SQL ステートメントはパッケージが DYNAMICRULES RUN 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。これが行われえない場合、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。これが行われない場合、パッケージ内の追加バインドおよび動的ステートメントはランタイム・エラーを生じます (SQLSTATE 42985)。

FEDERATED

パッケージ内の静的 SQL ステートメントがニックネームまたはフェデレーテッド・ビューを参照するかどうかを指定します。このオプションが指定されず、パッケージ内の静的 SQL ステートメントがニックネームまたはフェデレーテッド・ビューを参照する場合は、警告が返され、パッケージは作成されます。

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。

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、254 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 1023 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を指定します。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 文節は、定位置 UPDATE で更新されるすべての列において任意指定です。
- UPDATE または DELETE ステートメントの対象となる表の列が、検索条件の中や代入文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 文節は、定位置 UPDATE で更新されるすべての列において必須です。
- UPDATE または DELETE ステートメントの対象となる表が、そのステートメントの全選択から参照されるのであれば、検索 UPDATE または DELETE に、対象となる表に対する SELECT 特権は不要です。
- 索引を使って解決可能な列関数 (例えば MIN または MAX) は、NULL を検査せず、警告 SQLSTATE 01003 は戻されません。
- 警告が戻され、ユニーク制約が重複していても無視されます。
- 特権が付与されていないなら、エラーが戻されます。

SQL92E

以下のように ISO/ANS SQL92 規則を定義します。

- SQLCODE または SQLSTATE の値の検査をサポートするには、その名前の変数をホスト変数の宣言セクションで宣言できます (どちらも宣言されていないなら、プリコンパイル中には SQLCODE が指定されているものとされます)。
- C の NULL 終了ストリングにはブランクが埋め込まれ、切り捨てが実行された場合でも常に NULL 終了文字が含まれます。

- FOR UPDATE 文節は、定位置 UPDATE で更新されるすべての列において任意指定です。
- UPDATE または DELETE ステートメントの対象となる表の列が、検索条件の中や代入文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、リクエスト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。

NOLINEMACRO

出力 .c ファイルでの # 行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作成ツールでファイルを使用する場合に役に立ちます。このプリコンパイル・オプションは、C/C++ プログラム言語でのみ使用します。

OPTHINT

照会最適化ヒントを静的 SQL に使用するかどうかを制御します。サポートしているのは DB2 for OS/390 だけです。サポートされているオプション値のリストについては、DB2 for OS/390 の資料を参照してください。

OPTLEVEL

SQL ステートメントの中でホスト変数を使用されている場合に、内部 SQLDA の初期化を C/C++ プリコンパイラーで最適化するかどうかを指示します。そのように最適化すれば、密なループ内で単一の SQL ステートメント (FETCH など) を使う場合のパフォーマンスが向上します。

- 0** SQLDA の初期化を最適化しないようプリコンパイラーに指示します。
- 1** SQLDA の初期化を最適化するようプリコンパイラーに指示します。アプリケーションで以下のものを使っている場合は、この値を指定しないでください。

- 次の例に示すようなポインター・ホスト変数

```
exec sql begin declare section;
char (*name)[20];
short *id;
exec sql end declare section;
```

- 直接 SQL ステートメントに指定される C++ データ・メンバー

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 UDB for iSeries 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。DB2 UDB for iSeries だけによってサポートされています。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 iSeries 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は、iSeries システムの命名規則を使用する特定の SQL ステートメントに関して実行時に構文エラーを報告することがあります。例えば、OS400NAMING SYSTEM オプションを指定してプリコンパイルまたはバインドされているかどうかには関係なく、iSeries 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 30 文字の許可 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。

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 (拡張子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 8 文字です。

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、

```
xlc -P -DMYMACRO=0
```

QUALIFIER qualifier-name

パッケージに含まれる非修飾オブジェクトの 30 文字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 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、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。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・マーカー、または特殊レジスターを含む SQL ステートメントのアクセス・パスは、これらの変数の実際の値によって最適化されません。その代わりに、これらの変数のデフォルトの推定値が使用され、このプランがキャッシュされて使用されます。これがデフォルトの動作です。

ONCE 最初に照会が実行されるときに、ホスト変数、パラメーター・マーカー、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・マーカー、または特殊レジスターの既知の値によって、SQL ステートメントのアクセス・パスが必ずコンパイルおよび再最適化されます。

REOPT / NOREOPT VARS

これらのオプションは、REOPT ALWAYS および REOPT NONE に置き換えられましたが、以前のリリースとの互換性のために引き続きサポートされています。DB2 がホスト変数、パラメーター・マーカー、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するも、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、プリコンパイラー・リストに報告されます。

MVSD2V23 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、プリコンパイラー・リストに報告されます。

MVSD2V31 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、プリコンパイラー・リストに報告されます。

MVSD2V41 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、プリコンパイラー・リストに報告されます。

SORTSEQ

iSeries システムで使用するソート・シーケンス表を指定します。DB2 UDB for iSeries だけによってサポートされています。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 iSeries 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。
- ユーザーまたはアプリケーションが LOB 応答セット列の形式をどのように指定するか。

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。
- ユーザーまたはアプリケーションは、LOB 列の形式を最初のフェッチ要求においてしか指定できません。

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。
- ユーザーまたはアプリケーションは、LOB 列の形式を各フェッチ要求ごとに変更できます。

SQLWARN

動的 SQL ステートメントのコンパイル時 (PREPARE または EXECUTE IMMEDIATE を通して)、または記述処理時 (PREPARE...INTO または DESCRIBE を通して) に警告を戻すかどうかを指示します。

NO SQL コンパイラーから警告を戻しません。

YES SQL コンパイラーから警告を戻します。

SQLCODE +238 は例外です。これは、**SQLWARN** オプションの値が何であろうと戻されます。

STATICREADONLY

静的カーソルを READ ONLY として扱うかどうかを指定します。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 文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。

SYNCPOINT

複数のデータベース接続にまたがってコミットまたはロールバックを調整する仕方を指定します。このコマンド・パラメーターは無視されます。ここでは、後方互換性のためにこれを掲載しています。

NONE 2 フェーズ・コミットを実行するのにトランザクション管理機能 (TM) を使用しないことを指定し、単一更新機構、多重読み取り機構を適用しません。コミットは、関連する各データベースに送られます。コミットが失敗したときのリカバリーは、アプリケーションが行います。

ONEPHASE

2 フェーズ・コミットを実行するのに **TM** を使用しないことを指定します。複数のデータベース・トランザクションの各データベースが行う作業をコミットするときは、1 フェーズ・コミットが使用されます。

TWOPHASE

このプロトコルをサポートする複数のデータベースにまたがって 2 フェーズ・コミットを調整するのに **TM** が必要であることを指定します。

SYNTAX

プリコンパイル時に、パッケージまたはバインド・ファイルの作成を抑制します。このオプションを使用すれば、既存のパッケージまたはバインド・ファイルを修正したり変更したりしないで、ソース・ファイルの妥当性を検査できます。 **Syntax** は **sqlerror check** の同義語です。

syntax を **package** オプションと一緒に使うと、**package** は無視されます。

TARGET

現行のプラットフォームでサポートされているコンパイラーの 1 つに合わせて調整した修正コードを生成するように、プリコンパイラーに指示します。

IBMCOB

AIX では、IBM COBOL Set (AIX 版) のコンパイラーのためのコードが生成されます。

MFCOB

Micro Focus COBOL コンパイラー用のコードが生成されます。すべての UNIX オペレーティング・システムおよび Windows で、COBOL プリコンパイラーにおいて **target** 値が指定されなかった場合、この値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。

FORTTRAN

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 18 文字です。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ければそうなります)、そのタイム・スタンプは ISO 文字フォーマットに変換されて、バージョン ID として使用されます。

WCHARTYPE

GRAPHIC データのフォーマットを指定します。

CONVERT

wchar_t 基本タイプを使って宣言されたホスト変数には、wchar_t 形式のデータが入れられるものとして扱われます。この形式は、デ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。

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が使用される環境について DYNAMICRULES オプションで指定された動作をします。

SQL ステートメントがエラーであることが検出され、PRECOMPILE オプション SQLERROR CONTINUE が指定されている場合、このステートメントは無効とマークされます。この SQL ステートメントの状態を変えるには、さらに別の PRECOMPILE を発行する必要があります。暗黙的および明示的な再バインドでは、VALIDATE RUN でバインドされたパッケージ内の無効なステートメントの状態は変わりません。ステートメントは、再バインド時にオブジェクトが存在するかまたは権限の問題があるかどうかに応じて、暗黙的と明示的な再バインドとの両方で、静的バインドから追加バインドに変更したり、追加バインドを静的バインドに変更することができます。

REOPT ONCE または REOPT ALWAYS を指定してパッケージをバインディングすると、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。

関連概念:

- 「SQL ガイド」の『動的 SQL における許可に関する考慮事項』
- 「組み込み SQL アプリケーションの開発」の『動的 SQL における DYNAMICRULES BIND オプションの影響』
- 「組み込み SQL アプリケーションの開発」の『BIND コマンドの REOPT オプションを使用した場合のパフォーマンスの改善』
- 「組み込み SQL アプリケーションの開発」の『C および C++ 組み込み SQL アプリケーションでのグラフィック・データ用の WCHARTYPE プリコンパイラー・オプション』

関連タスク:

- 「パフォーマンス」の『オーバーヘッド削減のための行ブロッキングの指定』

関連資料:

- 389 ページの『BIND』
- 「SQL リファレンス 第 1 巻」の『日付/時刻の値』
- 「SQL リファレンス 第 2 巻」の『SET CURRENT QUERY OPTIMIZATION ステートメント』
- 「管理 API リファレンス」の『sqlaprep API - アプリケーション・プログラムのプリコンパイル』

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存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:

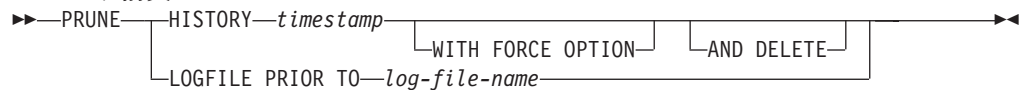
以下のいずれか。

- *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。

LOGFILE PRIOR TO log-file-name

ログ・ファイル名を表すストリング (例: *S0000100.LOG*) を指定します。指定したログ・ファイルより前のすべてのログ・ファイルは削除されます。指

PRUNE HISTORY/LOGFILE

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:

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、 **dbckrst** 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こともあります。 PRUNE HISTORY コマンドを FORCE オプション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。

履歴ファイルからバックアップ項目を整理すると、 DB2 Data Links Manager サーバー上にある関連ファイルのバックアップが削除されます。

関連概念:

- 「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『バックアップとリカバリーの計画の作成』

関連資料:

- 「管理 SQL ルーチンおよびビュー」の『PRUNE HISTORY/LOGFILE コマンド (ADMIN_CMD プロシージャを使用)』

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

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なりません。OWNER 文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。この文節が指定されない場合、ルーチンが定義されるときに、デフォルト・スキーマと SQL パスの設定が使用されます。ルーチン定義 (ルーチンの名前を含む) の非修飾オブジェクト名のスキーマ名として CURRENT SCHEMA が使用され、ルーチン定義の非修飾ルーチンとデータ・タイプを解決するために CURRENT PATH が使用されます。

例:

```
PUT ROUTINE FROM procs/procl.sar;
```

使用上の注意:

指定のスキーマの下で、複数のプロシージャが並行してインストールされることはありません。

GET ROUTINE または PUT ROUTINE 操作 (またはそれに対応するプロシージャ) が正常に実行できない場合、エラー (SQLSTATE 38000)、および失敗の原因に関する情報を示す診断テキストを毎回戻します。例えば、GET ROUTINE に指定されたプロシージャ名が SQL プロシージャを識別しない場合、"-204, 42704" という診断テキストが戻されます。"-204" は SQLCODE、"42704" は SQLSTATE

PUT ROUTINE

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。

関連資料:

- 530 ページの『`GET ROUTINE`』
- 「管理 *SQL* ルーチンおよびビュー」の『`PUT_ROUTINE_SAR` プロシージャ』

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。

関連資料:

- 「管理 API リファレンス」の『sqleqryc API - クライアント接続設定の照会』
- 784 ページの『SET CLIENT 』

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:

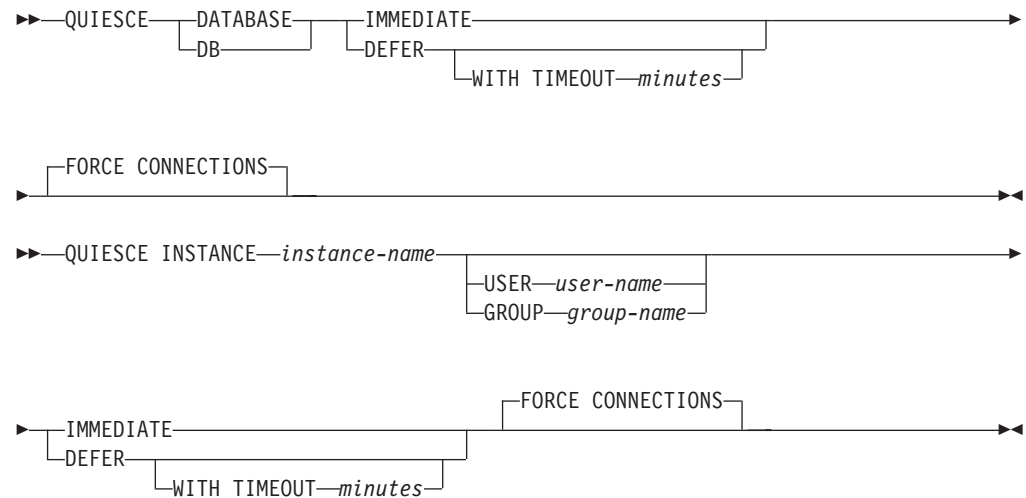
- *sysadm*
- *sysctrl*

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

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_timeout* 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 CONNECTION` 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 CONNECTION` 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` と同期され、`FORCE` 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、次のようになります。

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

関連資料:

- 825 ページの『UNQUIESCE』
- 「管理 API リファレンス」の『db2DatabaseQuiesce API - データベースの静止』
- 「管理 API リファレンス」の『db2InstanceQuiesce API - インスタンスの静止』
- 「管理 SQL ルーチンおよびビュー」の『QUIESCE DATABASE コマンド (ADMIN_CMD プロシージャを使用)』

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 の中のリストに含まれているすべての表スペースのうち、状況が「正常」、「アタッチ」、または「デタッチ」であるもの（例えば SYSDATAPARTITIONS.STATUS がそれぞれ '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:

```

▶▶ QUIESCE TABLESPACES FOR TABLE tablename [ schema.tablename ] {
    SHARE
    INTENT TO UPDATE
    EXCLUSIVE
    RESET
}
  
```

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。

QUIESCE TABLESPACES FOR TABLE

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、認められます。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、静止排他状態に残ります。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。例えば、次のようなものがあります。

```
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。

関連資料:

- 「管理 API リファレンス」の『sqluvqdp API - 表の表スペースの静止』
- 601 ページの『LIST TABLESPACES 』
- 608 ページの『LOAD 』
- 670 ページの『QUIESCE 』
- 825 ページの『UNQUIESCE 』
- 「管理 SQL ルーチンおよびビュー」の『QUIESCE TABLESPACES FOR TABLE コマンド (ADMIN_CMD プロシージャを使用)』

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。

関連資料:

- 815 ページの『**TERMINATE**』

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。

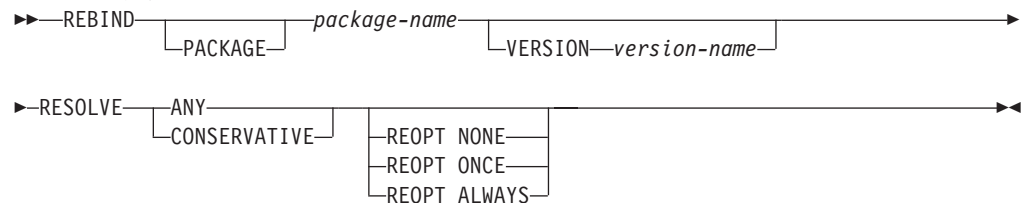
- `sysadm` または `dbadm` の権限
- スキーマに対する `ALTERIN` 特権
- パッケージに対する `BIND` 特権

`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 と異なっていても問題ありません。REBIND は、パッケージの作成時に指定されたのと同じ `BIND` オプションを使用します。

必要な接続:

データベース。データベース接続が存在しない場合で、暗黙の接続が有効な場合には、デフォルト・データベースへの接続が行われます。

コマンド構文:



コマンド・パラメーター:

PACKAGE package-name

再バインドされるパッケージを指定する修飾されている、または修飾されていない名前。

VERSION version-name

再バインドするパッケージのバージョン。バージョンが指定されない場合は、"" (空ストリング) と見なされます。

RESOLVE

パッケージの再バインドの実行に、古典的なバインド・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、パッケージの静的 DML ステートメントの関数およびタイプの解決時に、新しい関数およびデータ・タイプを対象にす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 関数およびタイプの解決時に、SQL パスにあるすべての関数とタイプを対象にします。古典的なバインド・セマンティクスは使用されません。これがデフォルトです。

CONSERVATIVE

関数およびタイプの解決時に、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 パスにある関数とタイプだけを対象にします。古典的なバインド・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、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、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、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、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

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以外に、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、移行の後、データベースに格納されているすべてのパッケージが、DB2 バージョン 8 の移行作業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 を使用するようにしてください。それは、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。しかしながら、次の場合には、BIND を使用しなければなりません。

- 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、ユーザーが、バインド・プロセスの部分として付与されたパッケージの特権を所有したい場合、BIND が、**grant** オプションを持っているために、それを使用しなければなりません。
- 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・エラーの検出が必要な場合。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。

REBIND が他のユーザーが使用中のパッケージで実行された場合、他のユーザーの LU の作業が終了するまで、再バインドは起こりません。再バインドの間、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保留になっているからです。

REBIND を実行する際、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

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 によっても、ステートメントが無効の状態は変更できません。VALIDATE RUN でバインドされたパッケージでは、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/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。

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。

関連資料:

- 389 ページの『BIND』
- 769 ページの『RUNSTATS 』
- 250 ページの『db2rbind - Rebind all packages 』
- 「管理 API リファレンス」の『sqlarbnd API - パッケージの再バインド』

RECONCILE

表の DATALINK データについてファイルへの参照を妥当性検査します。ファイルへの参照を確立できない行が例外表 (指定されている場合) にコピーされ、入力ファイルで変更されます。

RECONCILE を使用すると、UNIX ベース・システムのインスタンス・パスと Windows プラットフォームのインストール・パスに、メッセージ・ファイル (reconcil.msg) が生成されます。このファイルには、例外表の妥当性検査で生成された警告およびエラー・メッセージが入ります。

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:

```
►► RECONCILE—table-name—DLREPORT—filename—┐
                                                    └FOR EXCEPTION—table-name┘►►
```

コマンド・パラメーター:

RECONCILE table-name

調整を実行する表を指定します。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。修飾子付き表名は、*schema.tablename* の形式です。非修飾の表名を指定すると、その表は現行許可 ID で修飾されます。

DLREPORT filename

調整時にリンク解除されるファイルについての情報を含むファイルを指定します。名前は完全修飾する必要があります (例: /u/johnh/report)。調整ユーティリティは、指定されたファイル名に *.ulk* 拡張子を付け加えます (例:report.ulk)。FOR EXCEPTION 文節に表が指定されない場合は、例外レポート・ファイルに *.exp* のファイル拡張子が付け加えられます。

FOR EXCEPTION table-name

DATALINK 値へのリンク障害が発生している行がコピーされる例外表を指定します。表が指定されない場合は、"DLREPORT" オプションで指定されたディレクトリーに例外レポート・ファイルが生成されます。

例:

以下のコマンドを実行すると、表 DEPT の調整が行われ、ユーザーが既に作成している例外表 EXCPTAB に例外が書き込まれます。調整時にリンク解除されたファイルについての情報は、/u/johnh ディレクトリーに作成される report.ulk ファイ

ルに書き込まれます。FOR EXCEPTION excptab を指定しなかった場合は、/u/johnh ディレクトリーに作成される report.exp ファイルに例外情報が書き込まれます。

```
db2 reconcile dept dlreport /u/johnh/report for exception excptab
```

使用上の注意:

調整時には、表データに従って存在しているファイルで、データ・リンク・ファイル・マネージャーのメタデータに従って存在しているわけではないものをリンクしようとします。ただし、このことはこれ以外に競合がない場合です。必要な DB2 Data Links Manager は、表で参照される DATALINK 値を持っています。調整は、必要な DB2 Data Links Manager と、データベースに対して構成されていてデータ表の一部ではない他の DB2 Data Links Manager の可用性を許容します。

調整は表内のすべての DATALINK データに関して実行されます。ファイル参照を再確立できない場合は、違反行が例外表 (指定されている場合) に挿入されます。これらの行は、入力表からは削除されません。ファイル参照の健全性を確実にするために、問題の DATALINK 値は NULL にされます。列が NULL 可能ではないものとして定義されている場合、DATALINK 値はゼロの長さの URL に置き換えられます。

ファイルが WRITE PERMISSION ADMIN で定義された DATALINK 列の下にリンクしており、変更されてなおコミットされていない場合 (つまり、ファイルは依然として更新処理中の状態にある) は、調整プロセスによって、変更されたファイルの名前が .mod という接尾部付きの名前に変更されます。またこれにより、ファイルは更新処理中の状態でもなくなります。DATALINK 列が RECOVERY YES で定義されている場合は、直前のアーカイブ・バージョンがリストアされます。

例外表が指定されていない場合、ファイル参照が再確立できなかった各 DATALINK 列値のホスト名、ファイル名、列 ID、および理由コードが、例外レポート・ファイル (<filename>.exp) にコピーされます。DB2 Data Links Manager が使用できないこと、または DROP DATALINKS MANAGER コマンドを使用して DB2 Data Links Manager がデータベースからドロップされたことが原因でファイル参照を再確立できなかった場合、例外レポート・ファイルで報告されるファイル名は完全なファイル名ではありません。このファイル名には接頭部がありません。例えば、オリジナルの DATALINK 値が http://host.com/dlfs/x/y/a.b の場合、例外表で報告される値は http://host.com/x/y/a.b になります。接頭部名 'dlfs' は含まれません。

DATALINK 列が RECOVERY YES で定義されている場合は、直前のアーカイブ・バージョンがリストアされます。

調整プロセスの終了時、すべての必須 DB2 Data Links Manager で調整処理が完了した場合にのみ、表がデータ・リンク調整ペンディング (DRP) 状態になります。必須 DB2 Data Links Manager のいずれかで (その必須 DB2 Data Links Manager が使用できないことが原因で) 調整処理がペンディングになっている場合、表は DRP 状態になります。何らかの理由により、影響を受ける Data Links Manager の 1 つで調整が正常に完了できないなどの例外が生じた場合は、表も DRNP 状態に置かれることがあります。このような状態になった場合は、表をリストアできるだけの十分な参照保全を得るために、さらに手操作による介入が必要になります。

例外表 (指定されている場合) を作成してからでないと、調整ユーティリティは実行できません。調整ユーティリティで使用する例外表は、ロード・ユーティリティで使用する例外表と同じです。

例外表は、調整する表の定義を模倣します。例外表では、データ列の後に 1 つか 2 つのオプション列を加えることもできます。最初のオプション列は、TIMESTAMP 列です。この列には、調整操作が開始された時刻のタイム・スタンプが含まれます。2 番目のオプション列は、タイプ CLOB (32KB 以上) にする必要があります。この列には、リンク障害の発生した列の ID と、その障害の理由が含まれます。例外表の DATALINK 列は NO LINK CONTROL を指定しなければなりません。そのように指定することにより、行 (DATALINK 列を含む) が挿入されてもファイルはリンクされず、例外表から行が選択されてもアクセス・トークンが生成されずに済みます。

MESSAGE 列の情報は、以下の構造に従って編成されます。

フィールド 番号	内容	サイズ	注釈
1	違反の数 B	5 文字	右寄せ '0' を埋め込む
2	違反のタイプ	1 文字	'L' - DATALINK 違反
3	違反の長さ	5 文字	右寄せ '0' を埋め込む
4	違反している DATALINK 列の数	4 文字	右寄せ '0' を埋め込む
5	DATALINK 列番号 (最初の違反列の)	4 文字	右寄せ '0' を埋め込む
6	違反の理由	5 文字	右寄せ '0' を埋め込む
			フィールド 5 と 6 を、それぞれの 違反列ごとに繰り返す

以下に示すのは、可能性のある違反のリストです。

- 00001 - DB2 Data Links Manager がファイルを見つけられない。
- 00002 - ファイルは既にリンクされている。
- 00003 - ファイルは変更された状態にある。
- 00004 - 接頭部の名前が登録されていない。
- 00005 - ファイルが検索できなかった。
- 00006 - ファイル項目がない。これは、
RECOVERY NO、READ PERMISSION FS、WRITE PERMISSION FS DATALINK の列の
場合に生じます。
ファイルを再リンクするには更新を使用します。
- 00007 - ファイルがリンク解除状態にある。
- 00008 - リストアされたが変更されていないファイルは
<filename>.MOD にコピーされている。
- 00009 - ファイルは既に別の表にリンクされている。
- 00010 - DATALINK 値によって参照される DB2 Data Links Manager が、
DROP DATALINKS MANAGER コマンドを使用して
データベースからドロップされている。
- 00999 - ファイルをリンクできなかった。

例:

RECONCILE

00001L000220002000400002000500001

- 00001 - 違反の数が 1 であることを指定する。
- L - 違反のタイプが 'DATALINK 違反' であることを指定する。
- 00022 - 違反の長さが 12 バイトであることを指定する。
- 0002 - リンクの障害を検出した行には 2 つの列があることを指定する、
- 0004,00002
- 0005,00001 - 列 ID と違反の理由を指定する。

メッセージ列がある場合は、タイム・スタンプ列もあるはずです。

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しか呼び出せません。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`

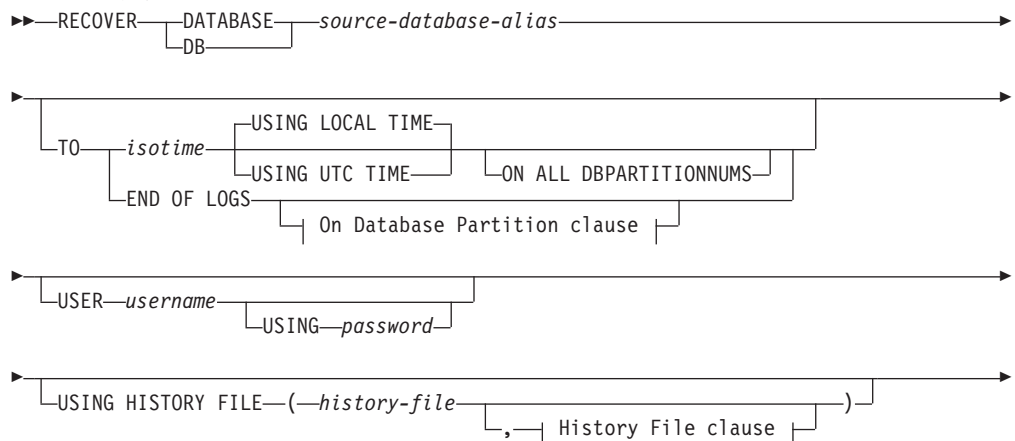
新規のデータベースにリカバリーするには、次のいずれかが必要です。

- `sysadm`
- `sysctrl`

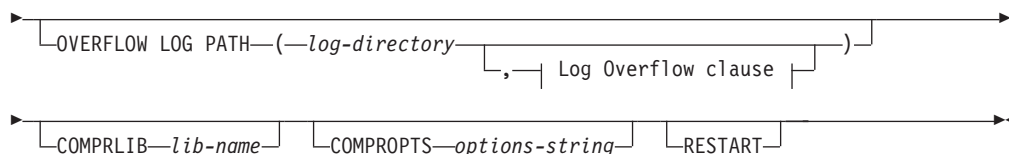
必要な接続:

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。

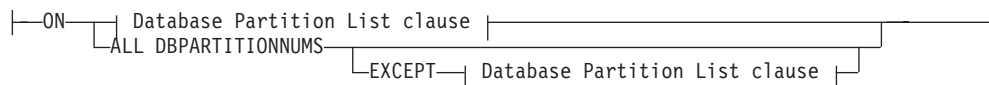
コマンド構文:



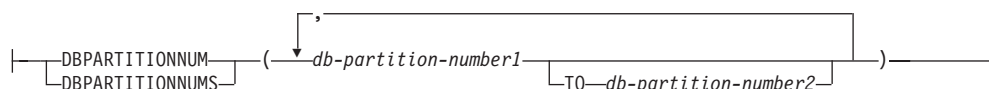
RECOVER DATABASE



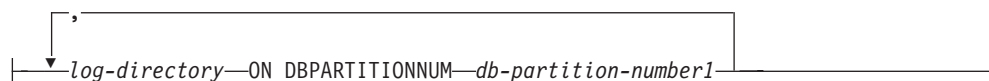
On Database Partition clause:



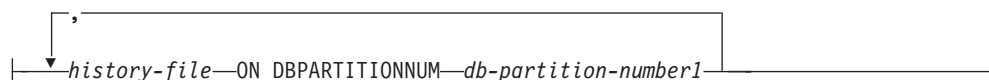
Database Partition List clause:



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、日付と時刻の組み合わせを識別する 7 つの部分の文字ストリングからなる、タイム・スタンプとして指定します。形式は、*yyyy-mm-dd-hh.mm.ss.nnnnnn* (年、月、日、時、分、秒、マイクロ秒) の協定世界時 (UTC。以前は GMT として知られる) です。UTC は、さまざまなログに関連したタイム・スタンプが (例えば、夏時間に関連して時間が変更されることによって) 同じにならないようにするのに役立ちます。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した地方時に基づいています。
CURRENT TIMEZONE 特殊レジスターは、UTC とアプリケーション

ン・サーバーの地方時との時差を指定します。時差は、時刻期間 (最初の 2 桁が時間数、次の 2 桁が分数、最後の 2 桁が秒数を表す 10 進数) で表されます。地方時から CURRENT TIMEZONE を減算すると、地方時を 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。

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、パーティション・データベース環境の場合にはカタログ・データベース・パーティション上で変換されます。
2. タイム・スタンプ・ストリングはサーバー上で UTC に変換されるので、時間はクライアントではなくサーバーの時間帯に基づく現地時間となります。クライアントの時間帯とサーバーの時間帯とが異なる場合、サーバーの現地時間を使用してください。これはコントロール・センターの現地時間オプションとは異なります。そのオプションは、クライアントの現地時間を使用します。
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

データベースの構成パラメーターである *logpath* にリストされる、すべてのオンライン・アーカイブ・ログ・ファイルからコミットされた全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 ファイルで指定されている、すべてのデータベース・パーティションについ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。データベース・パーティション文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パーティションを除き、db2nodes.cfg ファイルで指定されている、すべてのデータベース・パーティションについ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

一連のデータベース・パーティションについてデータベースをロールフォワードします。

RECOVER DATABASE

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て、*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*

リカバリー処理中にアーカイブされたログを探索する代替のログ・パスを指定します。ログ・ファイルが *logpath* データベース構成パラメーターで指定されるロケーション以外のロケーションに移動された場合に、このパラメーターを使用します。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・パスです。単一パーティション・データベースの場合は、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。v91 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、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 *dfidbpath* 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. 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** 文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)
```

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** 文節を使用して、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。)

RECOVER DATABASE

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、

1. 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
```

2. 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
```

3. *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)
```

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 文節を使用して、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。

関連概念:

- 「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『バックアップとリカバリーの計画の作成』

関連タスク:

- 「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『リカバリーの使用』

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分散します。現行のデータ分散（均等であるか、スキューであるかに関係なく）を指定できます。再分散アルゴリズムは、現行のデータ分散に基づいて移動するパーティションを選択します。

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:

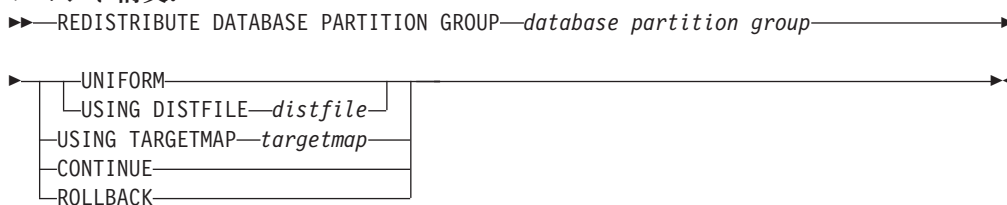
このコマンドは、データベース・パーティション・グループ内のすべてのデータベース・パーティションに影響を与えます。

許可:

以下のいずれか。

- `sysadm`
- `sysctrl`
- `dbadm`

コマンド構文:



コマンド・パラメーター:

DATABASE PARTITION GROUP *database 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。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分散することはできません。

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。再分散後、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 *distfile*

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、データバ

ース・パーティション・グループのデータベース・パーティション全体にわたるデータの均一な再分散を行います。

distfile を使用して、4 096 個のハッシュ・パーティションにわたる現行のデータの分散を指示します。

行カウント、バイト・ボリューム、または他の任意の尺度を使用して、各ハッシュ・パーティションで表示されたデータ量を示します。ユーティリティは、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。*distfile* を指定した場合、ユーティリティはターゲット分散マップを生成し、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データをできるだけ均一に再分散するために使用します。再分散の後、データベース・パーティション・グループ中の各データベース・パーティションの重みは、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、データベース・パーティションにマップしたすべてのデータベース・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 112 000 の重みを持ち、パーティション 3 (重さは 0) には、マッピングするデータがまったくありません。

distfile には、4 096 の正整数値が文字形式で入っているとします。値の合計は、4 294 967 295 以下である必要があります。

distfile のパスが指定されていない場合、現行ディレクトリーが使用されます。

USING TARGETMAP *targetmap*

targetmap 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。データの再分散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と、エラーが戻されます。

REDISTRIBUTE DATABASE PARTITION GROUP を実行する前に、ALTER DATABASE PARTITION GROUP ADD DBPARTITIONNUM を発行してください。

ターゲット・マップから除外されたデータベース・パーティションが、データベース・パーティション・グループにある場合、そのデータベース・パーティションはパーティションの中に含まれていません。このようなデータベース・パーティションは、REDISTRIBUTE DATABASE PARTITION GROUP の前か後に ALTER DATABASE PARTITION GROUP DROP DBPARTITIONNUM を使用することによってドロップできます。

REDISTRIBUTE DATABASE PARTITION GROUP

CONTINUE

直前に失敗した REDISTRIBUTE DATABASE PARTITION GROUP 操作を継続します。何も起こらなければ、エラーが戻されます。

ROLLBACK

直前に失敗した REDISTRIBUTE DATABASE PARTITION GROUP 操作をロールバックします。何も起こらなければ、エラーが戻されます。

使用上の注意:

- 再分散操作が実行されると、メッセージ・ファイルは次のディレクトリーに書き込まれます。
 - UNIX ベースのシステムの場合、`/sqllib/redis` 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、次の形式が使用されます。
database-name.database-partition-group-name.timestamp。
 - Windows オペレーティング・システムの場合、`¥sqllib¥redis¥` 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、次の形式が使用されます。
database-name¥first-eight-characters-of-the-database-partition-group-name¥date¥time。
- 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。
- このユーティリティは、処理中に断続的な COMMIT を実行します。
- ALTER DATABASE PARTITION GROUP ステートメントを使用して、データベース・パーティションをデータベース・パーティション・グループに追加します。このステートメントでは、データベース・パーティション・グループに関連付けられた表スペース用のコンテナを定義することができます。
- DB2 パラレル・エディション (AIX 版) バージョン 1 構文では、ADD DBPARTITIONNUM および DROP DBPARTITIONNUM オプションを指定することは、`sysadm` 権限または `sysctrl` 権限のあるユーザーに限りサポートされます。ADD DBPARTITIONNUM の場合、コンテナはデータベース・パーティション・グループ内の既存のノードの、ノード番号が最小のコンテナと同じように作成されます。
- 再分散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・グループの再分散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。再分散メッセージ・ファイルには、再分散を受けたすべての表のリストが入ります。
- 再分散データベース・パーティション・グループ操作が完了した後に、RUNSTATS を発行して統計を更新することもお勧めします。
- 複製されたサマリー表や 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、再分散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、再分散を行うことはできません。
- 再分散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。再分散操作の結果、間違ったデータベース・パーティションにデータが存在することがわかった場合には、2 つのオプションがあります。以下を行うことができます。

1. 表をアンロードし、ドロップしてから再ロードする、あるいは
2. 新しいターゲット・マップを使用してデータベース・パーティション・グループを再分散する。

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。

関連タスク:

- 「パフォーマンス」の『パーティション間でのデータの再分散』
- 「管理ガイド: インプリメンテーション」の『データベース・パーティション・グループの変更』

関連資料:

- 「管理 API リファレンス」の『`sqludrdt` API - データベース・パーティション・グループ内でのデータの再分散』
- 573 ページの『`LIST DATABASE DIRECTORY`』
- 769 ページの『`RUNSTATS`』
- 677 ページの『`REBIND`』
- 「管理 SQL ルーチンおよびビュー」の『`REDISTRIBUTE DATABASE PARTITION GROUP` コマンド (`ADMIN_CMD` プロシージャを使用)』

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。

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。

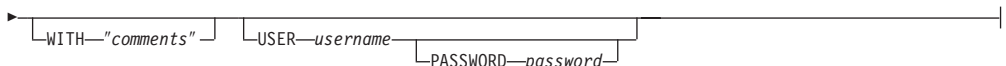
関連タスク:

- 「管理ガイド: インプリメンテーション」の『ローカル・データベースおよびノード・ディレクトリーの LDAP 項目をリフレッシュする』

許可:

必要な接続:

コマンド構文:



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。

管理サーバー・ノードを登録することを指定します。

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。

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 の名前付きパイプを使用する場合は、NPIPE を指定します。
NPIPE は Windows オペレーティング・システムでのみサポートされています。

HOSTNAME hostname

TCP/IP ホスト名 (または IP アドレス) を指定します。TCP/IP プロトコルの使用時には、IP アドレスは、IPv4 のアドレスでも IPv6 のアドレスでもかまいません。TCPIP4 プロトコルの使用時には、IP アドレスは IPv4 アドレスでなければなりません。TCPIP6 プロトコルの使用時には、IP アドレスは IPv6 アドレスでなければなりません。

SVCENAME svcename

TCP/IP サービス名またはポート番号を指定します。

SECURITY SOCKS

TCP/IP SOCKS を使用することを指定します。このパラメーターは、IPv4 でのみサポートされます。プロトコルとして TCPIP が指定された場合、基礎となるプロトコルとして使用されるのは IPv4 です。

REMOTE computer

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。これがデフォルトです。

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 "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。

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・サーバーを登録する場合には、NODETYPE パラメーターに値 DCS を指定しなければなりません。

関連概念:

- 「管理ガイド: インプリメンテーション」の『Lightweight Directory Access Protocol (LDAP) 』

関連タスク:

- 「管理ガイド: インプリメンテーション」の『インストール後の DB2 サーバーの登録』

関連資料:

- 「管理 API リファレンス」の『db2LdapRegister API - DB2 サーバーの LDAP サーバーへの登録』
- 458 ページの『DEREGISTER 』
- 852 ページの『UPDATE LDAP NODE 』

REGISTER XMLSCHEMA

XML スキーマを XML スキーマ・リポジトリ (XSR) に登録します。

許可:

以下のいずれか。

- 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]
    [COMPLETE [WITH schema-properties-URI] [ENABLE DECOMPOSITION]]
  
```

xml-document-subclause:

```

    >> ADD document-URI FROM content-URI [WITH properties-URI]
  
```

説明:

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た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。

関連資料:

- 371 ページの『ADD XMLSCHEMA DOCUMENT 』
- 434 ページの『COMPLETE XMLSCHEMA 』

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)>
  <!ENTITY c SYSTEM "http://www.xmlwriter.net/copyright.xml">
]>
<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 XSROBJECT 'http://www.xmlwriter.net/copyright.xml'
FROM 'c:\temp\copyright.xml' EXTERNAL ENTITY
```

2. 以下の XML 文書フラグメントは DTD を参照します。

```
<!--inform the XML processor
that an external DTD is referenced-->
<?xml version="1.0" standalone="no" ?>

<!--define the location of the
external DTD using a relative URL address-->
<!DOCTYPE document SYSTEM "http://www.xmlwriter.net/subjects.dtd">

<document>
  <title>Subjects available in Mechanical Engineering.</title>
  <subjectID>2.303</subjectID>
  <subjectname>Fluid Mechanics</subjectname>
  ...
```

この文書を正常に XML 列に挿入するには、その前に DTD を登録する必要があります。以下のコマンドは、DTD 定義がローカルに 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)>
  <!ENTITY c PUBLIC "-//W3C//TEXT copyright//EN"
    "http://www.w3.org/xmlspec/copyright.xml">
]>
<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
```

関連概念:

REGISTER XSROBJECT

- 「XML ガイド」の『XSR オブジェクト登録』
- 「XML ガイド」の『XSR オブジェクト』

関連資料:

- 700 ページの『REGISTER XMLSCHEMA 』

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。

索引文節の **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:

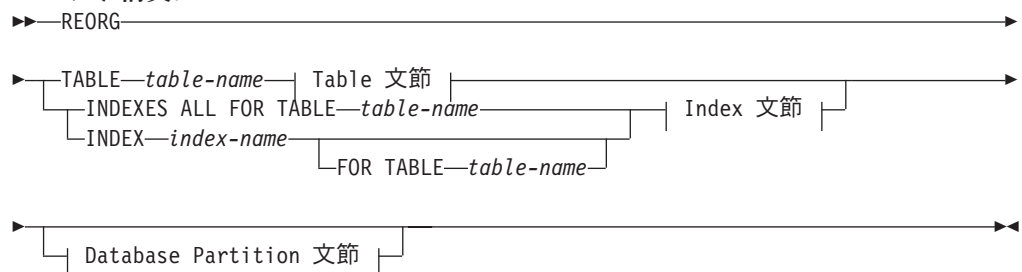
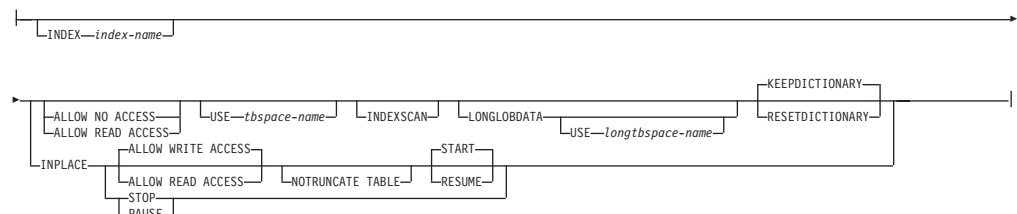
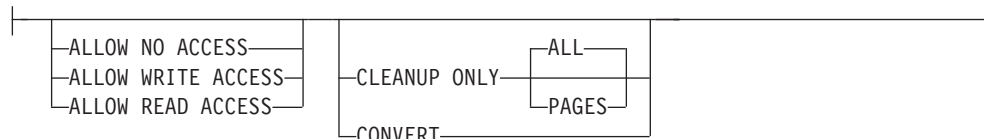
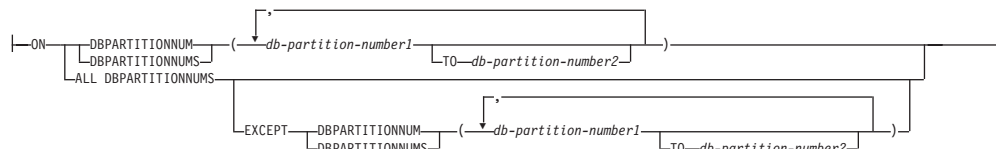


Table 文節:



Index 文節:**Database Partition 文節:****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 10. コマンド、表タイプ、および *index* (索引) 文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) 文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 列を調べることで判別できます。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。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、再開時にこ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 文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 を読み取って行を取り出す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。

USE longtblspace-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になります。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になります。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 が指定されている場合、USE-tbspace も指定する必要があります。そうでない場合、longtbspace 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。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 11. 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 12. REORG RESETDICTIONARY

Compress	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	
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、現在のところ新しいディクショナリーを作成するのに不適切なデータが表中に存在する場合、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。

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** ユーティリティでは、ニックネームの使用はサポートされません。
- **REORG TABLE** コマンドは、宣言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 表中のパーティション表が属している表スペース (LOB 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。
 - エラーが発生した場合、その表の非パーティション索引は不良としてマークされ、その表に次回アクセスする際に再作成されます。
 - 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。
 - パーティション表の索引を再編成する場合、非同期索引クリーンアップ完了後に RUNSTATS 操作を実行することをお勧めします。それは、デタッチされているデータ・パーティションの存在に関して正確な索引統計情報を生成するためです。表の中にデタッチされているデータ・パーティションがあるかどうかを調べるには、SYSDATAPARTITIONS の中の状況フィールドをチェックして、値が "I" (索引クリーンアップ) かそれとも "D" (依存 MQT からデタッチされている) かを調べることができます。

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ーマ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 INDEXES` 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。

インプレース `REORG TABLE` 操作の後には、データ・オブジェクトだけが再編成されて索引は再編成されないため、索引は最適ではないことがあります。インプレース `REORG TABLE` 操作の後に `REORG INDEXES` を実行することをお勧めします。しかし、索引はクラシック `REORG TABLE` の最終フェーズで完全に再作成されるので、索引の再編成は必要ありません。

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` 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せん。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。

関連概念:

- ・ 「管理ガイド: インプリメンテーション」の『データ行圧縮』
- ・ 「パフォーマンス」の『表の再編成』

関連資料:

- ・ 「管理 API リファレンス」の『db2Reorg API - 索引または表の再編成』
- ・ 532 ページの『GET SNAPSHOT 』
- ・ 677 ページの『REBIND 』
- ・ 716 ページの『REORGCHK 』
- ・ 「管理 SQL ルーチンおよびビュー」の『REORG INDEXES/TABLE コマンド (ADMIN_CMD プロシージャを使用)』
- ・ 「管理 SQL ルーチンおよびビュー」の『SNAPTAB_REORG 管理ビューおよび SNAP_GET_TAB_REORG 表関数 - 表再編成スナップショット情報の検索 』
- ・ 「管理 SQL ルーチンおよびビュー」の『REORGCHK_TB_STATS プロシージャ - 再編成の評価用の表統計の検索』
- ・ 「管理 SQL ルーチンおよびビュー」の『REORGCHK_IX_STATS プロシージャ - 再編成の評価用の索引統計の検索』

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:

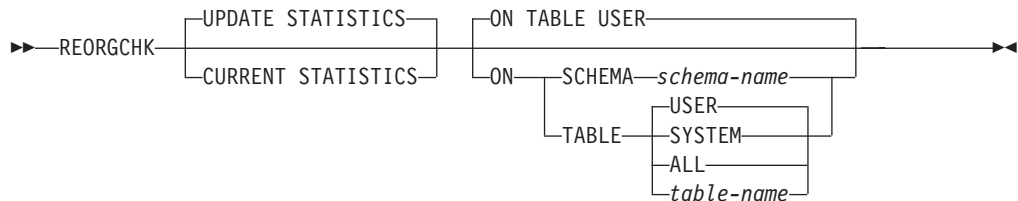
以下のいずれか。

- 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。名前は、*schema.table-name* 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** 文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 4~8) 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 0～99 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 13. 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
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、クリーンアップを必要とするデータタッチされたデータ・パーティションに属する非パーティション索引の中に索引キーがあると、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。パーティション表上にデタッチされたパーティションがあると、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。なぜなら、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。

CURRENT STATISTICS オプションを指定していなければ、**REORGCHK** はデフォルトのオプションだけを使用してすべての列から統計を収集します。特に、列グループは収集されません。さらに、**LIKE** 統計が以前に収集されている場合、それらは **REORGCHK** によっては収集されません。収集される統計は、カタログ表に現在保管されている統計の種類によって異なります。

- いずれかの索引のカタログ内に詳細索引統計が存在する場合、すべての索引の表統計および詳細索引統計が収集されます (サンプリングは行われません)。
- 詳細索引統計が収集されない場合、すべての索引の表統計および通常の索引統計が収集されます。
- 分散統計が検出された場合、その表についての分散統計が収集されます。分散統計が収集された場合、頻度および変位値の数はデータベース構成パラメーターの設定値によって異なります。

REORGCHK は、8 つの異なる公式から得た統計を計算し、表またはその索引の再編成によってパフォーマンスが低下するか、または改善できるのかを判別します。表が (**NPARTITIONS** * 1 エクステント・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 エクステント・サイズ)

- パーティション表では、以下のようになります。

(**FPAGES** <= **NPARTITIONS** * 1 エクステント・サイズ)

- 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含めた後、表の再編成が推奨されないこのしきい値を次のように変更します。

FPAGES

```
<= 'number of database partitions in a database partition group
of the table' * NPARTITIONS * 1 extent size
```

TSIZE の計算時には、長フィールドまたは LOB データは検討の対象にはなりません。

REORGCHK は、次の公式を使用して、行の物理的なロケーションおよび表のサイズを分析します。

- 公式 F1:

$$100 * \text{OVERFLOW} / \text{CARD} < 5$$

表のオーバーフロー行の合計数は、行の合計数の 5% 以下でなければなりません。オーバーフロー行は、行が更新されて、新しい行のバイト数が古い行 (VARCHAR フィールド) のそれより大きくなる場合、または列が既存の表に追加される場合に作成されます。

- 公式 F2:

通常の表では、以下のようになります。

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

バイト単位の表のサイズ (TSIZE) は、表に割り当てられた合計スペースの 70 % を超えていなければなりません (フリー・スペースは 30% 以下でなければなりません)。表に割り当てられる合計スペースは、表データが存在する表スペースのページ・サイズによって決まります (オーバーヘッド分の 68 バイトを差し引きします)。データ・オブジェクト内で割り当てられる最終ページは、通常はいっぱいにならないため、各パーティションごとに FPAGES から 1 を引きます (これは、FPAGES-NPARTITIONS と同じです)。

MDC 表では、以下のようになります。

$$100 * \text{TSIZE} / ((\text{ACTBLK} - \text{FULLKEYCARD}) * \text{EXTENTSIZE} * (\text{TABLEPAGESIZE} - 68)) > 70$$

FULLKEYCARD は、MDC 表の複合ディメンション索引のカーディナリティーを示します。エクステント・サイズは、ブロックごとのページ数です。バイト単位の表サイズが、必要最低限のブロック数を差し引いた後の残りのブロックの 70 パーセントを超えているかどうか、公式でチェックされます。

- 公式 F3:

$$100 * \text{NPAGES} / \text{FPAGES} > 80$$

まったく行を含まないページ数は、ページ合計数の 20% より少ない値にします (行が削除された後では、ページは空になります)。上で述べたとおり、(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は、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」に変更されます。

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、オブ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ブ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:

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

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)。

• 公式 F6:

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

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チェックします。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。

(The max number of leaf pages that a NLEVELS-1 index can have) *
(Amount of space available to store index entries per leaf page)

詳細は次のとおりです。

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

$$\begin{aligned} & \left(\frac{100 - \text{LEVEL2PCTFREE}}{100} \right) * \\ & \left(\frac{(\text{PageSize} - \text{Overhead})}{(\text{Avg. size of each nonleaf index entry})} \right) * \\ & \left(\frac{(100 - \text{MIN}(10, \text{LEVEL2PCTFREE}))}{100} \right) * \\ & \left(\frac{(\text{PageSize} - \text{Overhead})}{(\text{Avg. size of each nonleaf index entry})} \right) ** \\ & (\text{NLEVELS}-3) \} \end{aligned}$$

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

Level 2 is the level immediately above the leaf level.

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

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

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

The amount of space needed for all index entries:
KEYS * (LEAF_REC_SIZE + LEAF_REC_SIZE_OVERHEAD) +
(INDCARD - KEYS) * DUPKEYSIZE

$(KEYS * (LEAF_RECSIZE + LEAF_RECSIZE_OVERHEAD))$ は、索引内の各キー値の最初の出現箇所に使用されるスペースを表し、 $((INDCARD - KEYS) * DUPKEYSIZE)$ は、キー値のその後の (重複) 出現箇所に使用されるスペースを表します。

- 公式 F7:

$$100 * (NUMIDS_DELETED / (NUMIDS_DELETED + IND CARD)) < 20$$

疑似空白ではないページ上の疑似削除された RID の数は 20% 未満でなければなりません。

- 公式 F8:

$$100 * (NUM_EMPTY_LEAFS/NLEAF) < 20$$

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。

関連概念:

- 「パフォーマンス」の『表の再編成』

関連資料:

- 「管理 SQL ルーチンおよびビュー」の『REORGCHK_IX_STATS プロシージャー – 再編成の評価用の索引統計の検索』
- 「管理 SQL ルーチンおよびビュー」の『REORGCHK_TB_STATS プロシージャー – 再編成の評価用の表統計の検索』

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 ADMINISTRATION 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

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。

関連資料:

- 486 ページの『GET ADMIN CONFIGURATION 』
- 827 ページの『UPDATE ADMIN CONFIGURATION 』
- 「パフォーマンス」の『構成パラメーターのサマリー』

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:

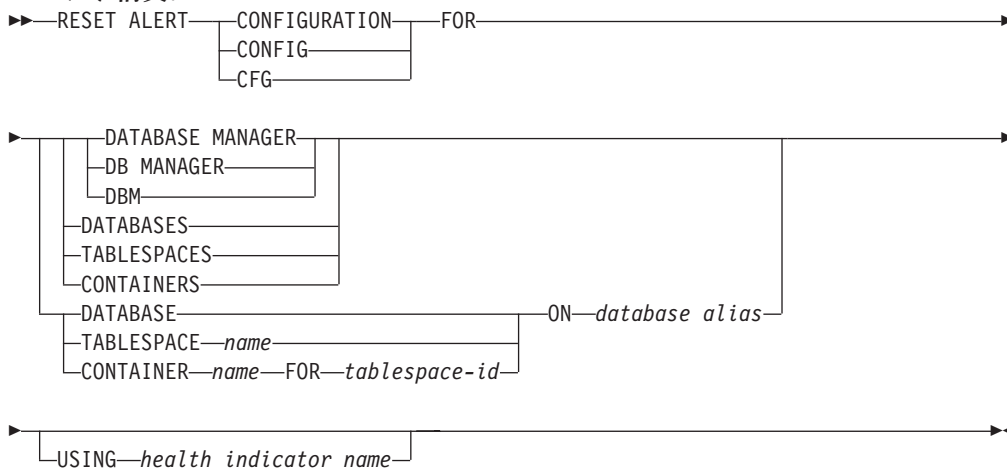
以下のいずれか。

- 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

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。

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` 文節を使って定義されます。

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 name ON database alias"` 文節を使って定義されます。

CONTAINER name FOR tablespace-id FOR tablespace-id ON database alias
 "ON database alias" 文節を使って指定したデータベース上で、 "FOR

tablespace-id" 文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。

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** *name* **ON** *database alias*" 文節を使って定義されます。

DATABASE *ON database alias*

ON database alias 文節を使って指定される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 *name* **ON** *database alias*

ON database alias 文節を使って指定されるデータベースで、 *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。例えば、次のようなものがあります。

```
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。

関連タスク:

- 「システム・モニター ガイドおよびリファレンス」の『クライアント・アプリケーションを使用したヘルス・インディケーターの構成』

関連資料:

- 「管理 API リファレンス」の『db2ResetAlertCfg API - ヘルス・インディケーターのアラート構成のリセット』
- 「管理 SQL ルーチンおよびビュー」の『RESET ALERT CONFIGURATION コマンド (ADMIN_CMD プロシージャを使用)』
- 「管理 SQL ルーチンおよびビュー」の『HEALTH_GET_ALERT_ACTION_CFG 表関数 - ヘルス・アラート・アクション構成設定値の検索』
- 「管理 SQL ルーチンおよびビュー」の『HEALTH_GET_ALERT_CFG 表関数 - ヘルス・アラート構成設定値の検索』
- 829 ページの『UPDATE ALERT CONFIGURATION 』

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、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:

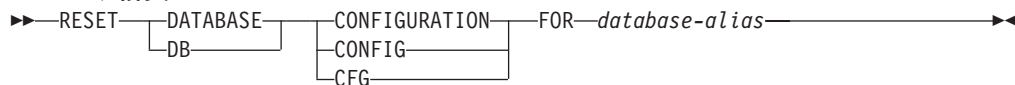
以下のいずれか。

- *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。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。

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データベース構成前の値にリセットされ、`AUTO_RUNSTATS` は `ON`

になります。SELF_TUNING_MEMORY 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。

関連タスク:

- 「パフォーマンス」の『構成パラメーターによる DB2 の構成』

関連資料:

- 502 ページの『GET DATABASE CONFIGURATION 』
- 844 ページの『UPDATE DATABASE CONFIGURATION 』
- 「管理 API リファレンス」の『db2CfgSet API - データベース・マネージャー構成パラメーター、あるいはデータベース構成パラメーターの設定』
- 「パフォーマンス」の『構成パラメーターのサマリー』
- 「管理 SQL ルーチンおよびビュー」の『RESET DATABASE CONFIGURATION コマンド (ADMIN_CMD プロシージャを使用)』
- 「管理 SQL ルーチンおよびビュー」の『DBCFCG 管理ビュー - データベース構成パラメーター情報の検索』

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、リモートのデータベース・マネージャー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:

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、SVCENAME 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 **SOL5043N 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 SVCENAME パラメーターの、ユーザーによる修正は推奨されません。

データベース・マネージャ構成パラメータ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** を呼び出す必要があります。

エラーが生じた場合には、データベース・マネージャー構成ファイルは変更されません。

データベース・マネージャー構成ファイルは、そのチェックサムが無効であると、リセットすることができません。これはデータベース・マネージャー構成ファイルを手動で編集して、適切なコマンドを使用しない場合に生じることがあります。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。

関連タスク:

- 「パフォーマンス」の『構成パラメーターによる DB2 の構成』

関連資料:

- 508 ページの『GET DATABASE MANAGER CONFIGURATION 』
- 815 ページの『TERMINATE 』
- 847 ページの『UPDATE DATABASE MANAGER CONFIGURATION 』
- 「管理 API リファレンス」の『db2CfgSet API - データベース・マネージャー構成パラメーター、あるいはデータベース構成パラメーターの設定』
- 「パフォーマンス」の『構成パラメーターのサマリー』
- 「管理 SQL ルーチンおよびビュー」の『RESET DATABASE MANAGER CONFIGURATION コマンド (ADMIN_CMD プロシージャを使用)』
- 「管理 SQL ルーチンおよびビュー」の『DBMCFG 管理ビュー - データベース・マネージャー構成パラメーター情報の検索』

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 DBPARTITIONNUM の代わりに NODE を使用できます。

関連資料:

- 「管理 API リファレンス」の『db2ResetMonitor API - データベース・システム・モニターのデータのリセット』
- 523 ページの『GET MONITOR SWITCHES 』
- 532 ページの『GET SNAPSHOT 』

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。

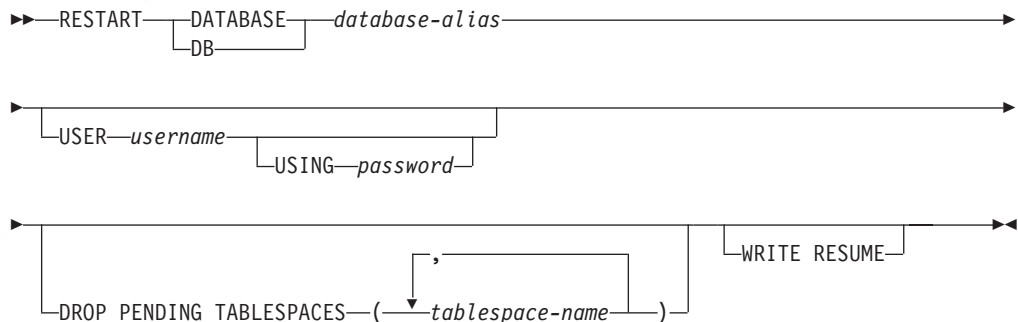
許可:

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・コンテナに問題が発生しても、データベース再始動操作を正常に完了するよう指定します。

指定した表スペース用のコンテナで再始動処理中に問題が発生した場合、それに対応する表スペースは、再始動操作後に使用できなくなります (ドロップ・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。

RESTART DATABASE の完了時に、ユーザーが CONNECT 特権を持っている場合には、データベースへの共有接続は維持され、未確定のトランザクションが存在する場合には、SQL の警告が発行されます。この場合、データベースはまだ使用できる状態ですが、未確定トランザクションが、データベースへの最終接続をドロップする前に解決されない場合には、別の RESTART DATABASE を発行してから、再度データベースを使用しなければなりません。LIST INDOUBT TRANSACTIONS コマンドを使用して、未確定トランザクションのリストを生成してください。

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。

関連タスク:

- 「管理ガイド: プランニング」の『未確定トランザクションの手動での解決』

関連資料:

- 589 ページの『LIST INDOUBT TRANSACTIONS 』
- 「管理 API リファレンス」の『db2DatabaseRestart API - データベースの再始動』

RESTORE DATABASE

RESTORE DATABASE コマンドは、DB2 バックアップ・ユーティリティを使用し、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、『関連概念』セクションにある『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・ユーティリティは、DB2 UDB 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。移行が必要な場合、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。)

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*

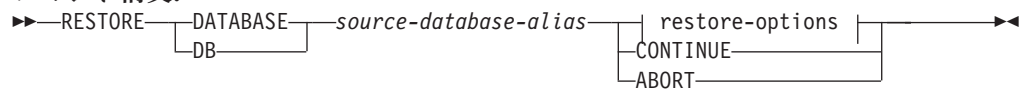
必要な接続:

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。

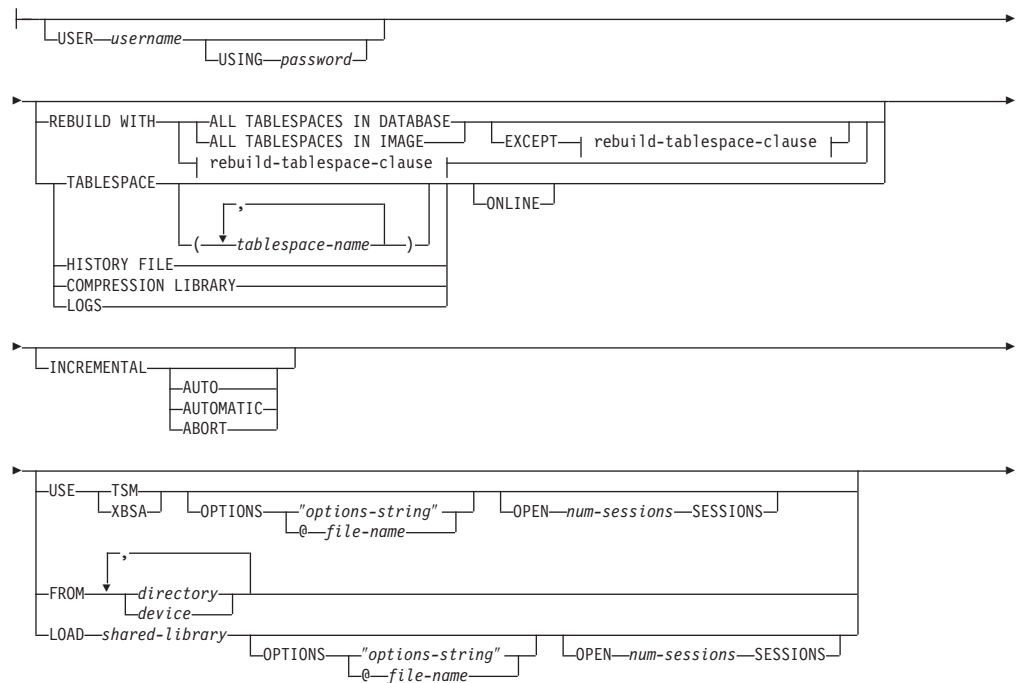
- 既存のデータベースにリストアするには、データベース接続が必要です。このコマンドは、指定されたデータベースへの排他接続を自動的に確立します。
- 新しいデータベースにリストアするには、インスタンスおよびデータベース接続が必要です。データベースを作成するには、インスタンス接続が必要です。

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す。

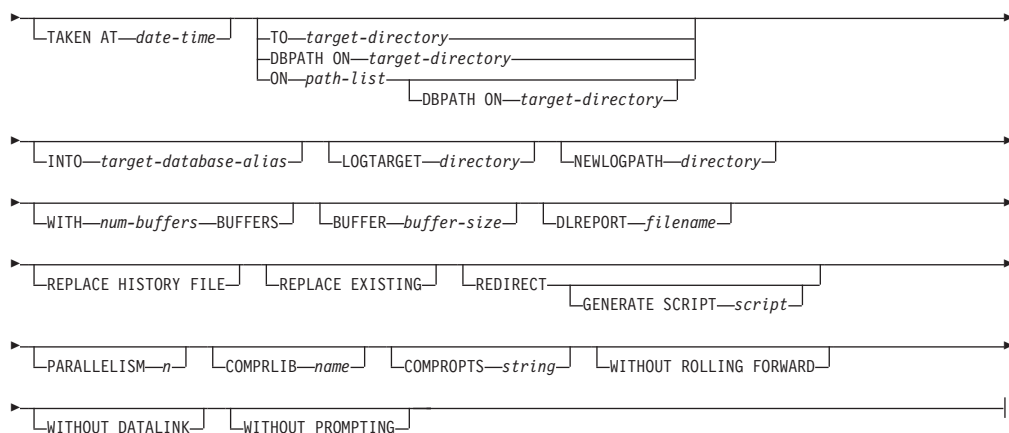
コマンド構文:



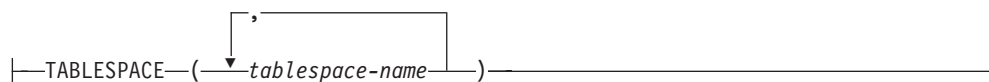
restore-options:



RESTORE DATABASE



rebuild-tablespace-clause:



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

コンテナー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

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す。

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。

RESTORE DATABASE

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 TSM 管理の出力からリストアされるように指定します。

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、TSM などのベンダー・サポートのライブラリーに、引用符なしで入力された場合とまったく同じように渡されます。このオプションを指定すると、VENDOROPT 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値がオーバーライドされます。

@file-name

リストア操作で使用するオプションが、DB2 サーバー上のファイルに含まれていることを指定します。このストリングは、TSM などのベンダー・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。

USE 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。

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。

Windows オペレーティング・システムでは、DB2 が生成するディレクトリーを指定してはなりません。例えば、次のようなコマンドを実行するとします。

```
db2 backup database sample to c:¥backup
db2 restore database sample from c:¥backup
```

これらのコマンドを使用すると DB2 データベース・システムは、c:¥backup ディレクトリーの下にいくつかのサブディレクトリーを生成し、それにより指定されたトップレベルのディレクトリーの中に複数のバックアップを入れることができるようにします。DB2 の生成するサブディレクトリーは無視してください。リストアするバックアップ・イメージを正確に指定するためには、TAKEN AT パラメーターを使用します。複数のバックアップ・イメージを同じパスに保管することもできます。

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、もうテープがない場合に停止する、など)。
- t** 終了。ユーザーが、ユーティリティーによって要求された何らかのアクションを実行しなかった場合、リストア操作を異常終了します。

LOAD *shared-library*

使用するバックアップおよびリストア I/O ベンダー関数を含む共有ライブラリー (Windows オペレーティング・システムでは DLL) の名前。名前には絶対パスを含めることができます。絶対パスを指定しない場合、ユーザー 出口プログラムが置かれているパスがデフォルト値として使われます。

TAKEN AT *date-time*

データベース・バックアップ・イメージのタイム・スタンプです。タイム・スタンプはバックアップ操作が正常に終了した後に表示され、バックアップ・イメージのパス名の一部になっています。 *yyyymmddhhmmss* の形式で指定されます。タイム・スタンプを部分的に指定することもできます。例えば、2 つの異なるタイム・スタンプ 20021001010101 および 20021002010101 で指定されるバックアップ・イメージが存在する場合、20021002 を指定することで、タイム・スタンプ 20021002010101 のイメージが使用できます。このパラメーターに値を指定しない場合は、ソース・メディア上のバックアップ・イメージは 1 つだけでなければなりません。

TO *target-directory*

このパラメーターは、ターゲット・データベース・ディレクトリーを指定します。ユーティリティーが存在するデータベースヘリストアしている場合には、このパラメーターは無視されます。指定するドライブおよびディレクトリーは、ローカルのものでなければなりません。自動ストレージが有効になったデータベースがバックアップ・イメージに含まれる場合、データベース・ディレクトリーだけが変更され、そのデータベースに関連したストレージ・パスは変更されません。

DBPATH ON *target-directory*

このパラメーターは、ターゲット・データベース・ディレクトリーを指定します。ユーティリティーが存在するデータベースヘ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。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 文節を指定することなく並列してリストアできます。

一般的に、複数パーティション・データベースでは、どのパーティションにも同じストレージ・パスを使用する必要があり、それらはすべて、

RESTORE DATABASE コマンドの実行前に存在している必要があります。その例外の 1 つとして、ストレージ・パス内でデータベース・パーティション式を使用する場合があります。その使用によって、処理結果のパス名が各パーティションごとに異なるように、データベース・パーティション番号をストレージ・パスにおいて反映することができます。

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。

表 14. : 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*

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。

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。

DLREPORT *filename*

ファイル名を指定する場合は、絶対パスとして指定しなければなりません。リストア操作中に高速調整が行われたためにリンク解除されたファイルを報告します。このオプションが使用されるのは、リストアする表に **DATALINK** 列タイプとリンク・ファイルが含まれている場合だけです。

RESTORE DATABASE

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。

WITHOUT DATALINK

DATALINK 列を持つ任意の表を **DataLink_Reconcile_Pending (DRP)** 状態にし、リンクされたファイルの調整を実行しないように指定します。

PARALLELISM *n*

リストア操作中に作成されるバッファ・マネージャの数指定しま

す。値を明示的に指定しない場合、DB2 データベース・システムはこのパラメーターの最適値を自動的に選択します。

COMPRLIB *name*

解凍を実行するために使用するライブラリーの名前。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 *string*

バイナリー・データのうち、解凍ライブラリーの初期設定ルーチンに渡すブロックを記述します。DB2 データベース・システムはこのストリングをクライアントからサーバーに直接渡すため、バイト反転やコード・ページ変換の問題がある場合は、解凍ライブラリーで処理されます。データ・ブロックの最初の文字が『@』なら、データの残りの部分は、サーバー上に存在するファイルの名前を指定するものとして、DB2 データベース・システムは解釈します。その場合、DB2 データベース・システムは *string* の内容をこのファイルの内容で置き換え、新しい値を初期設定ルーチンに渡します。ストリングの最大長は 1 024 バイトです。

WITHOUT PROMPTING

リストア操作を無人で実行するように指定します。通常はユーザー介入を必要とするアクションでは、エラー・メッセージが戻されます。テープやディスクセットなどの取り外し可能メディア装置を使用している場合、このオプションを指定していても、その装置が終わるとプロンプトが出されます。

例:

- 以下の例で、データベース WSDb は 0 から 3 までの番号が付けられた 4 つのデータベース・パーティションすべてに定義されています。パス /dev3/backup はすべてのデータベース・パーティションからアクセスできます。以下のオフライン・バックアップ・イメージは、/dev3/backup から入手可能です。

```
wsdb.0.db2inst1.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_a11 '<<+0< db2 RESTORE DATABASE wsdb FROM /dev3/backup
TAKEN AT 20020331234149
INTO wsdb REPLACE EXISTING'
db2_a11 '<<+1< db2 RESTORE DATABASE wsdb FROM /dev3/backup
TAKEN AT 20020331234427
INTO wsdb REPLACE EXISTING'
db2_a11 '<<+2< db2 RESTORE DATABASE wsdb FROM /dev3/backup
TAKEN AT 20020331234828
INTO wsdb REPLACE EXISTING'
db2_a11 '<<+3< db2 RESTORE DATABASE wsdb FROM /dev3/backup
TAKEN AT 20020331235235
INTO wsdb REPLACE EXISTING'
```


RESTORE DATABASE

db2_a11 ユーティリティーは、指定された各データベース・パーティションへのリストア・コマンドを出します。db2_a11 を使用してリストアを実行する場合は、常に REPLACE EXISTING や WITHOUT PROMPTING を指定してください。これを指定しないと、プロンプトが表示された場合に操作がハングしたように見えます。それは、db2_a11 でユーザー・プロンプトがサポートされていないためです。

2. 以下は、別名が MYDB であるデータベースの典型的なリダイレクト・リストアのシナリオです。

- a. 次のように、REDIRECT オプションを指定して RESTORE DATABASE コマンドを発行する。

```
restore db mydb replace existing redirect
```

ステップ 1 が正常終了した後でステップ 3 が完了する前に、次を発行してリストア操作を打ち切ることができる。

```
restore db mydb abort
```

- b. 再定義する必要があるコンテナを持つ表スペースごとに、SET TABLESPACE CONTAINERS コマンドを発行する。例えば、

```
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チェックするために、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'

```

RESTORE DATABASE

```
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

```

RESTORE DATABASE

```
-- BUFFER <buffer-size>
-- REPLACE HISTORY FILE
-- REPLACE EXISTING
REDIRECT
-- PARALLELISM <n>
-- WITHOUT ROLLING FORWARD
-- WITHOUT PROMPTING
;
-- *****
-- ** tablespace definition
-- *****
-- ** Tablespace name = SYSCATSPACE
-- ** Tablespace ID = 0
-- ** Tablespace Type = Database managed space
-- ** Tablespace Content Type = Any data
-- ** Tablespace Page size (bytes) = 4096
-- ** Tablespace Extent size (pages) = 4
-- ** Using automatic storage = Yes
-- ** Auto-resize enabled = Yes
-- ** Total number of pages = 6144
-- ** Number of usable pages = 6140
-- ** High water mark (pages) = 5968
-- *****
-- ** Tablespace name = TEMPSPACE1
-- ** Tablespace ID = 1
-- ** Tablespace Type = System managed space
-- ** Tablespace Content Type = System Temporary data
-- ** Tablespace Page size (bytes) = 4096
-- ** Tablespace Extent size (pages) = 32
-- ** Using automatic storage = Yes
-- ** Total number of pages = 0
-- *****
-- ** Tablespace name = USERSPACE1
-- ** Tablespace ID = 2
-- ** Tablespace Type = Database managed space
-- ** Tablespace Content Type = Any data
-- ** Tablespace Page size (bytes) = 4096
-- ** Tablespace Extent size (pages) = 32
-- ** Using automatic storage = Yes
-- ** Auto-resize enabled = Yes
-- ** Total number of pages = 256
-- ** Number of usable pages = 224
-- ** High water mark (pages) = 96
-- *****
-- ** Tablespace name = DMS
-- ** Tablespace ID = 3
-- ** Tablespace Type = Database managed space
-- ** Tablespace Content Type = Any data
-- ** Tablespace Page size (bytes) = 4096
-- ** Tablespace Extent size (pages) = 32
-- ** Using automatic storage = No
-- ** Auto-resize enabled = No
-- ** Total number of pages = 2000
-- ** Number of usable pages = 1960
-- ** High water mark (pages) = 96
-- *****
SET TABLESPACE CONTAINERS FOR 3
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
USING (
    FILE '/tmp/dms1' 1000
    , FILE '/tmp/dms2' 1000
);
-- *****
```

```

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

```

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・イメージに入っているかどうか検査され、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。
- 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・ユーティリティは、イメージ内のロ

RESTORE DATABASE

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。
- 別のオペレーティング・システムまたは別の DB2 データベース・バージョンにリストアする予定のバックアップは、オフライン・バックアップでなければならず、それをデルタまたは増分バックアップ・イメージにすることはできません。もっと新しいバージョンの DB2 データベースにリストアする予定のバックアップについても同じことが当てはまります。

関連概念:

- 「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』
- 「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『バックアップとリカバリーの計画の作成』

関連タスク:

- 「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『リストアの使用』

関連資料:

- 435 ページの『CREATE DATABASE 』
- 169 ページの『db2move - データベース移動ツール』

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。

関連資料:

- 560 ページの『INITIALIZE TAPE 』
- 791 ページの『SET TAPE POSITION 』
- 「管理 SQL ルーチンおよびビュー」の『REWIND TAPE コマンド (ADMIN_CMD プロシージャーを使用)』

ROLLFORWARD DATABASE

データベース・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、あるいはいくつかの表スペースがメディア・エラーのために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。ロールフォワード・リカバリーを使用してデータベースをリカバリーするには、前もってデータベースがリカバリー可能になっていなければなりません (すなわち、データベース構成パラメーター *logarchmeth1* または *logarchmeth2* が OFF 以外の値に設定されていなければなりません)。

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しか呼び出せません。指定された特定のポイント・イン・タイムへ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を与えます。データベース・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。特定のパーティションでロールフォワードが必要とされない場合には、そのパーティションは無視されます。

パーティション表の場合、関連する表スペースも同じポイント・イン・タイムにロールフォワードする必要があります。このことは、表のデータ・パーティションが含まれている表スペースについても適用されます。パーティション表の 1 つの部分が単一の表スペースに含まれている場合でも、ログの最後へのロールフォワードは可能です。

許可:

以下のいずれか。

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

必要な接続:

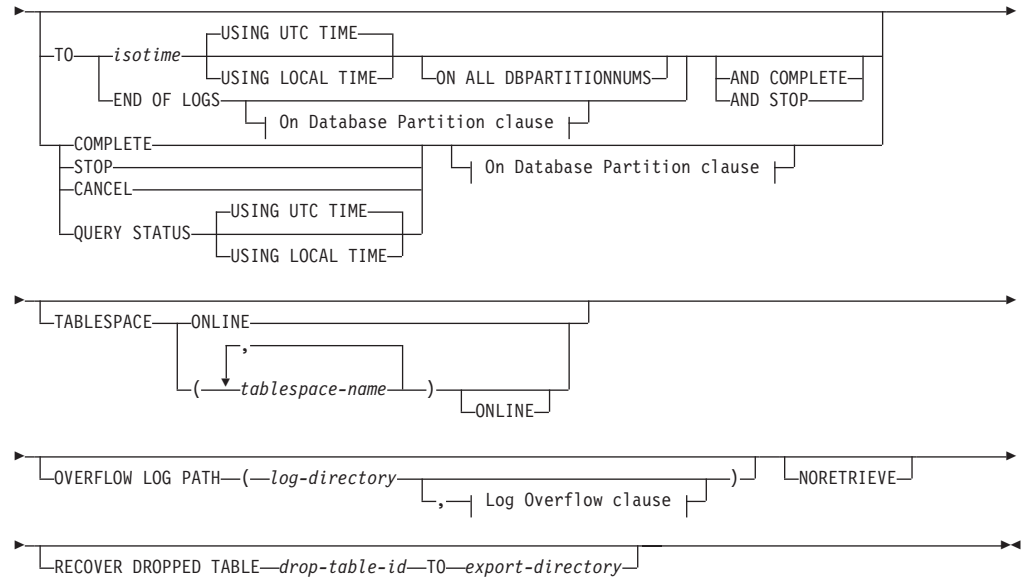
なし。このコマンドは、データベース接続を確立します。

コマンド構文:

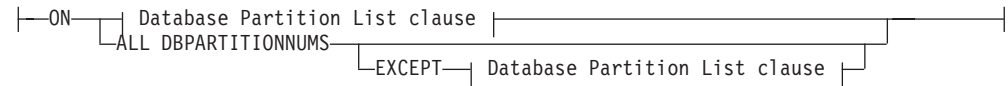
```

➤➤ROLLFORWARD DATABASE database-alias [USER=username [USING=password]]
    |_____|
    DB

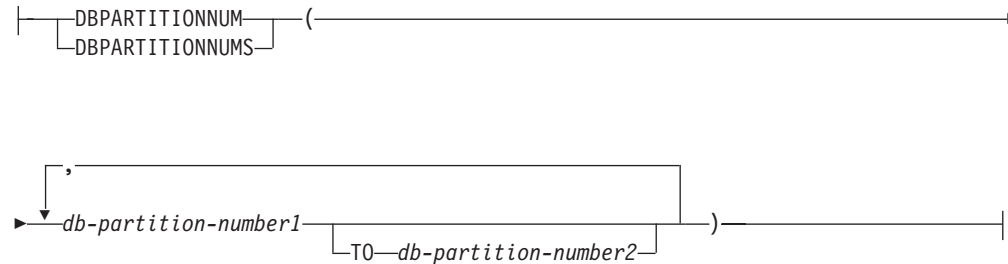
```



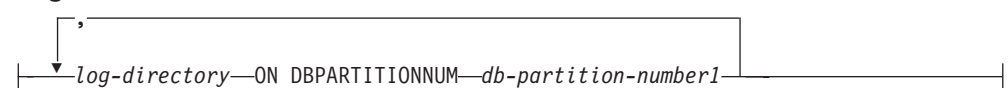
On Database Partition clause:



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・リカバリーするデータベースの別名。

USER username

データベースがロールフォワード・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、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) がロールフォワードされるポイント・イン・タイム。

この値は、日付と時刻の組み合わせを識別する 7 つの部分の文字ストリングからなる、タイム・スタンプとして指定します。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、さまざまなログに関連したタイム・スタンプが (例えば、夏時間に関連して時間が変更されることによって) 同じにならないようにするのに役立ちます。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した地方時に基づいています。CURRENT TIMEZONE 特殊レジスターは、UTC とアプリケーション・サーバーの地方時との時差を指定します。時差は、時刻期間 (最初の 2 桁が時間数、次の 2 桁が分数、最後の 2 桁が秒数を表す 10 進数) で表されます。地方時から CURRENT TIMEZONE を減算すると、地方時を UTC に変換できます。

USING LOCAL TIME

UTC 時間ではなくサーバーの現地時間を使用して特定の時点にロールフォワードすることを可能にします。

注:

1. ロールフォワードのために現地時間を指定した場合、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、パーティション・データベース環境の場合にはカタログ・データベース・パーティション上で変換されます。
2. タイム・スタンプ・ストリングはサーバー上で UTC に変換されるので、時間はクライアントではなくサーバーの時間帯に基づく現地時間となります。クライアントの時間帯とサーバーの時間帯とが異なる場合、サーバーの現地時間を使用してください。これはコントロール・センターの現地時間オプションとは異なります。そのオプションは、クライアントの現地時間を使用します。
3. タイム・スタンプ・ストリングが夏時間調整のための時間変更に接近している場合、停止時刻が時間変更の前か後かを判別して、それを適切に指定することが大切です。
4. USING LOCAL TIME 文節を指定することのできないそれ以降の ROLLFORWARD コマンドでこのオプションが指定されている場合、すべてのメッセージは現地時間で戻されます。

END OF LOGS

データベースの構成パラメーターである `logpath` にリストされる、すべてのオンライン・アーカイブ・ログ・ファイルからコミットされた全トランザクションが、適用されることを指定します。

ALL DBPARTITIONNUMS

`db2nodes.cfg` ファイルで指定されている、すべてのデータベース・パーテ

ィションについ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。データベース・パーティション文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パーティションを除き、db2nodes.cfg ファイルで指定されている、すべてのデータベース・パーティションについ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

一連のデータベース・パーティションについ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て、*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、未完了の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。これにより、既にロールフォワードされたデータベースまたは表スペースへのアクセスが認められます。これらのキーワードは同等です。どちらか一方を指定し、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)。 (end of logs ではなく) 特定のポイント・イン・タイムを指定して表スペースをロールフォワードすると、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す。

CANCEL

ロールフォワード・リカバリー操作を取り消します。これにより、順方向リカバリーが開始されている、すべてのデータベース・パーティション上のデータベースまたは 1 つ以上の表スペースがリストア・ペンディング状態になります。

- 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 (つまり、データベースがロールフォワード・ペンディング状態である) 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 (つまり、表スペースがロールフォワード・ペンディング状態である) 場合は、表スペース・リストを指定しなければなりません。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なります。
- 表スペースのロールフォワード操作が進行中である (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状態にある) 場合は、ロールフォワード進行中状態にあるすべての表スペースがリストア・ペンディング

状態になります。表スペース・リストを指定する場合、そのリストには、ロールフォワード進行中状態にある表スペースがすべて含まれていなければなりません。リストの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。

- あるポイント・イン・タイムまでロールフォワードする場合、渡される表スペース名はすべて無視され、ロールフォワード進行中状態にある表スペースがすべてリストア・ペンディング状態になります。
- 表スペース・リストを指定してログの終わりまでロールフォワードする場合は、リストされている表スペースのみがリストア・ペンディング状態になります。

このオプションは、実際に実行されている ロールフォワード操作を取り消すために使用することはできません。このオプションは、その時点で実際に実行されているロールフォワードではなく、進行中のロールフォワード操作を取り消すためにだけに使用できます。以下の場合に、ロールフォワード操作は実行中ではなく進行中の可能性があります。

- 異常終了した。
- STOP オプションが指定されなかった。
- エラーのために失敗した。表スペースをリストア・ペンディング状態にするエラーもあります。例えば、リカバリー不能ロード操作によるロールフォワードなどです。

このオプションを使用する場合は注意し、いくつか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なっているために、進行中のロールフォワード操作が完了しない場合だけ使用してください。はっきりと分からない場合は、表スペースがロールフォワード進行中状態になっているのか、ロールフォワード・ペンディング状態になっているのかを識別するために、LIST TABLESPACES コマンドを使用してください。

QUERY STATUS

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、次に必要とされるアーカイブ・ファイル、およびロールフォワード処理が開始されてから、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 形式) をリストを表示します。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・パーティションに関して戻されます。戻される情報には、次のフィールドが含まれています。

Database partition number

ロールフォワード状況

状態は次のいずれかです。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の停止、またはペンディングなし。

読み込む予定の次のログ・ファイル

次に必要なログ・ファイルの名前から成るストリング。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

に障害が起こり、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 で開始する場合は、古くなったログ・ファイルの範囲には x が含まれなくなり、範囲は $x - 1$ で終了します。

コミットされた最終トランザクション

ISO 書式のタイム・スタンプから成る文字列 (yyyy-mm-dd-hh.mm.ss) に、接尾部として “UTC” または “Local” を付けたもの (USING LOCAL TIME を参照)。このタイム・スタンプは、ロールフォワード・リカバリー処理の完了後に、コミットされた最終トランザクションを示しています。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・リカバリーでは、データベースにコミットされた最終トランザクションのタイム・スタンプを表します。

TO、STOP、COMPLETE、または CANCEL 文節を省略すると、QUERY STATUS がデフォルト値になります。TO、STOP、または COMPLETE を指定した場合は、コマンドが正常に完了したなら、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・リカバリーを行う場合に使用します。

tablespace-name

あるポイント・イン・タイムまでの表スペース・レベルのロールフォワード・リカバリーを行う場合は必須です。表スペースのサブセットに対してログの終わりまでのロールフォワード・リカバリーを指定できます。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あることを示します。

パーティション表の場合、パーティション表のいずれかの部分が入った表スペースのポイント・イン・タイム・ロールフォワードを行うためには、その表が置かれている他のすべての表スペースも、同じポイント・イン・タイムに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を収めた 1 つの表スペースのログの最後までロールフォワードすることも可能です。

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パーティション表の場合、そのようなデータ・パーティションの表スペースもすべて PIT ロールフォワードに含める必要があります。アタッチ、デタッチ、またはドロップされたデータ・パーティションがパーティション表にあるかどうかを判別するには、SYSDATAPARTITIONS カタログ表の Status 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内に置かれていることがあるので、一般的に、複数の表スペースのロールフォワードが必要になります。ドロップされた表のリカバリーを介してリカバリーされるデータは、

ROLLFORWARD DATABASE コマンドに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。すべての表スペースを 1 つのコマンドでロールフォワードできますが、関与する表スペースのサブセットのロールフォワード操作を繰り返してもかまいません。1 つまたは少数の表スペースに対して ROLLFORWARD DATABASE コマンドが実行されると、それらの表スペース中にある表のすべてのデータが回復されます。表のすべてのデータをリカバリーするのに必要な表スペースの完全セットを ROLLFORWARD DATABASE コマンドに指定しなかった場合、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が可能になれば、1 つのエクスポート・ディレクトリーに入り切らないほどのデータをリカバリーするようなケースの処理が簡単になります。

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行えるようにするために指定します。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

リカバリー処理中にアーカイブされたログを探索する代替のログ・パスを指定します。ログ・ファイルが *logpath* データベース構成パラメーターで指定されるロケーション以外のロケーションに移動された場合に、このパラメーターを使用します。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・パスです。単一パーティション・データベースの場合は、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、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・マシン上のどのログ・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、待機マシンが実動マシンより必ず X 時間遅れるようにして、どちらのシステムも影響を受けないようにすることができます。
- 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際に、スタンバイ・システムが同じファイルを検索していることも可能であり、その場合には不完全なログ・ファイルが取得されます。この問題は Noretrieve を使用して解決できます。

RECOVER DROPPED TABLE drop-table-id

ドロップされた表をロールフォワード操作中にリカバリーします。表 ID を取得するには、LIST HISTORY コマンドを使用します。パーティション表の場合、drop-table-id は表全体を識別するので、その表のすべてのデータ・パーティションを 1 つ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

表データが含まれているファイルを書き込むディレクトリーを指定します。指定するディレクトリーは、すべてのデータベース・パーティションにアクセスできるもの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、ログの終わりでロールフォワードし、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することができます。

```
db2 rollforward db sample to end of logs and complete
```

上記の 2 つは同じですが、このような操作は 2 つのステップで実行することをお勧めします。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。そうしない場合、ログが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、それ以降のログのロールフォワード操作を続けるために、そのログの損傷していないバックアップ・コピーを使用することができます。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、エラーにもかかわらず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 (つまりログの中で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、ロールフォワード 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ています)。

```
db2 rollforward db sample to end of logs
db2 rollforward db sample to end of logs and stop
```


これらの 2 つのステートメントは同じです。ログの終わりまで表スペースのロールフォワード操作を実行する場合、**AND STOP** または **AND COMPLETE** を使用する必要はありません。表スペース名は必須ではありません。指定しない場合には、ロールフォワード・リカバリーを必要としているすべての表スペースが組み込まれます。これらの表スペースの一部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しなければなりません。

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 つをあるポイント・イン・タイム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は、並行して実行できません。2 番目のコマンドは、最初のロールフォワード操作が正常に完了した場合にのみ起動することができます。

例 4

データベースをリストアした後、**OVERFLOW LOG PATH** でユーザー出口がアーカイブ・ログを保管するディレクトリを指定して、あるポイント・イン・タイムまでロールフォワードします。

```
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、文節 **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・リカバリー可能な状態になっていないため、これは失敗します。SQL4906N が報告されます。

```
db2 rollforward db sample to end of logs on dbpartitionnums (0, 2)
tablespace(TBS1)
```

これは正常に完了します。

```
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・リカバリー可能な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。表スペースをあるポイント・イン・タイムまでロールフォワードする場合、データベース・パーティション文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

```
db2 rollforward db sample to 1998-04-03-14.21.56 and stop
tablespace(TBS1)
```

これは正常に完了します。

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、特定時点 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、特定時点 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 ファイルにリスト表示されている 8 個のデータベース・パーティション (3 から 10 まで) に存在す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 (ポイント・イン・タイムではなく) 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 個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, tssvender)
```

ログの終わり (ポイント・イン・タイムではなく) までのこの操作は正常に完了します。

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ポイント・イン・タイムは、オンライン・バックアップの完了ポイント・イン・タイムより後でなければなりません。指定時刻の前にロールフォワード操作が停止する場合、データベースはロールフォワード・ペンディング状態になります。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になります。

1 つ以上の表スペースをあるポイント・イン・タイム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 (この表スペース用のシステム・カタログまたは表への最新の更新) 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 (UTC)) は、`LIST TABLESPACES SHOW DETAIL` コマンドを使用して検索できます。

データベースをロールフォワードするとき、テープ装置を使用するロード・リカバリーが必要な場合もあります。別のテープを求める要求が出された場合は、次のいずれか 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。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。使用できるログがなくなった場合は、ロールフォワード・リカバリーを終了するために `STOP` オプションを使用します。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。

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 `DBPARTITIONNUM` の代わりに `NODE` を使用できます。
- キーワード `DBPARTITIONNUMS` の代わりに `NODES` を使用できます。

- 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。

関連概念:

- 「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『バックアップとリカバリーの計画の作成』

関連タスク:

- 「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『ロールフォワードの使用』

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。

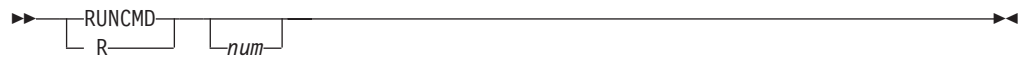
許可:

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。

関連資料:

- 476 ページの『EDIT 』
- 540 ページの『HISTORY 』

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は、表が数多く更新されるとき、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、照会最適化で利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 権限

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

RUNSTATS

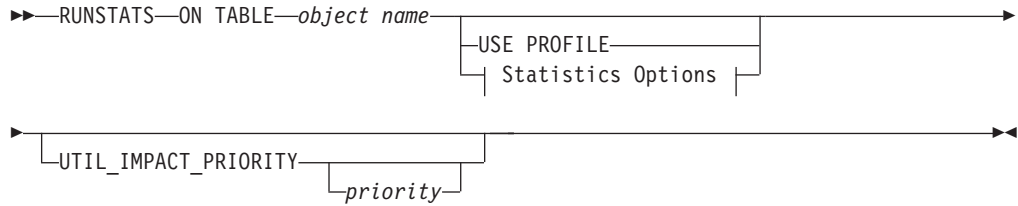
さらに、統計ビューの行にアクセスするための適切な特権が必要です。特に、統計ビューの定義の中で参照されている各表、統計ビュー、またはニックネームごとに、ユーザーには以下の特権のいずれか 1 つが必要です。

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

必要な接続:

データベース

コマンド構文:



Statistics Options:

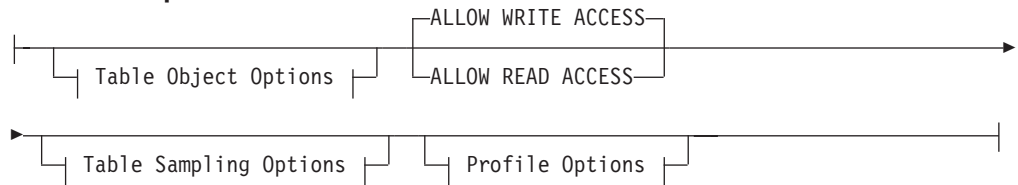


Table Object Options:

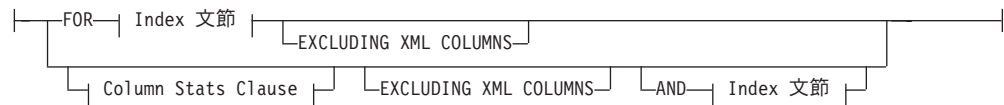
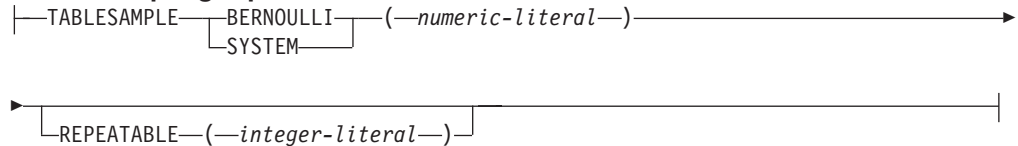


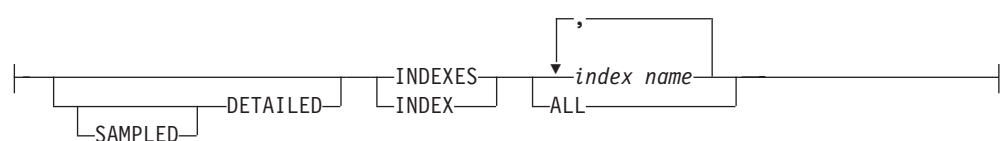
Table Sampling Options:



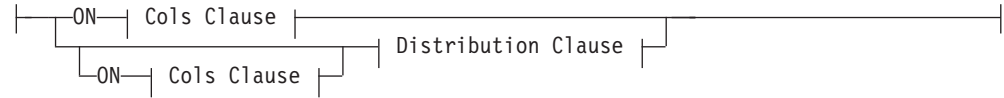
Profile Options:



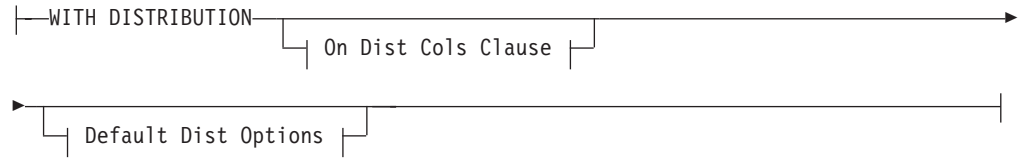
Index 文節:



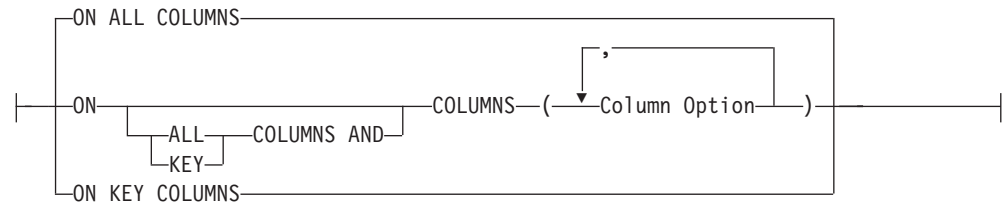
Column Stats Clause:



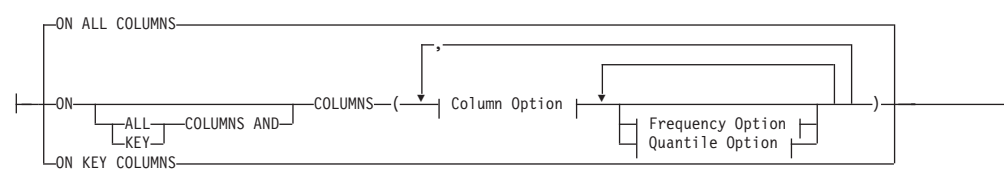
Distribution Clause:



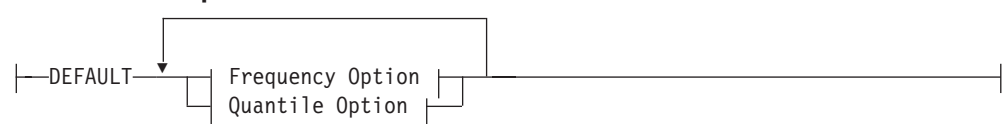
On Col's Clause:



On Dist Col's Clause:



Default Dist Option:



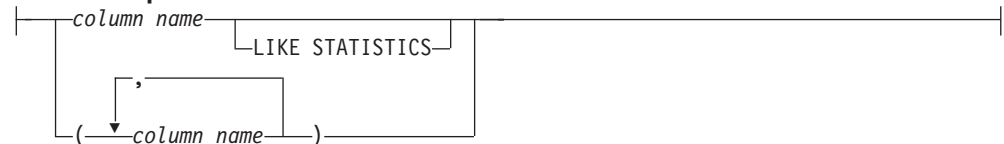
Frequency Option:



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。

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。名前は、`schema.index-name` 形式の完全修飾名を使用してください。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。

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` 文節を使用できます。列は、基本統計の収集に指定するか (`on-cols-clause`)、または `WITH DISTRIBUTION` 文節と組み合わせて指定 (`on-dist-cols-clause`) できます。これら列固有の文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` 文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` 文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

この文節では、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。

この文節は、`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

この文節を使用すると、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、この文節を使用すれば、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 文節は、統計情報収集の対象として XML 列を指定する他の文節よりも優先されます。例えば、EXCLUDING XML COLUMNS 文節を使用する場合、ON COLUMNS 文節で XML タイプの列を指定したり ON ALL COLUMNS 文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、その両方の文節で指定すると、WITH DISTRIBUTION 文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 文節による統計情報収集には含まれません。ビューの場合、このオプションは使用できません。

column-name

表または統計ビューの中の列の名前。存在しない列が指定された場合や列名の入力を誤った場合など、統計収集を行えない列の名前が指定された場合は、エラー (-205) が戻されます。一方は分散なし、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 文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 文節の中で、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 文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 文節の中で、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 文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

この文節は、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 文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 文節が指定されている場合は、指定されたリストにある列でのみ (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、文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 文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 文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 文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。これらの統計はストリング列に関してのみ収集され、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 文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 文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 文節を TABLESAMPLE 文節に追加すれば、**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、サンプル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、ユーティリティのスロットルの量が決まります。優先順位が同じユーティリティはすべて同じ量のスロットルになり、優先順位の低いユーティリティは、優先順位の高いユーティリティよりも量が絞られます。 *priority* を指定しない場合、**RUNSTATS** はデフォルトの優先順位 50 を使用します。

UTIL_IMPACT_PRIORITY キーワードを省略すると、スロットルのサポートなしで **RUNSTATS** ユーティリティが呼び出されます。

UTIL_IMPACT_PRIORITY キーワードを指定した場合でも、*util_impact_lim* 構成パラメーターが 100 に設定されていれば、スロットルなしでユーティリティが実行されます。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。

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、そのまま保たれます。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、述部で使用す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。SAMPLED オプションを使用するかどうかに関係なく、索引での詳細な統計の収集には時間がかかります。それで、これらの統計は、それが照会にとって有益であることが確かな場合以外、収集しないでください。
 - 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。
 - 等式の述部で使用する列や、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、述部で使われない列の 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文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う可能性があります。これは不整合です。これと同様に、作成した新しい索引で統計を収集した場合にも、表レベルの統計に不整合が生じることがあります。
5. **RUNSTATS** コマンドは、表統計が要求したときに、以前に収集された分散統計をドロップします。例えば、**RUNSTATS ON TABLE** または **RUNSTATS ON TABLE ... AND INDEXES ALL** は、以前に収集された分散統計がドロップされる原因になります。コマンドが索引でのみ実行される場合、以前に収集された分散統計は保存されます。例えば、**RUNSTATS ON TABLE ... FOR INDEXES ALL** は以前に収集された分散統計が保存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たれます。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** 文節では、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** サンプリングは、索引ページではなく表データ・ページでのみ行われます。索引統計とサンプリ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リング方法であり、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。

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.empl1, db2user.empl2
```
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. 索引で使われるすべての列と、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 empl_address と列グループ (empl_title, empl_salary) のオプションを追加することにより、以前に登録した統計プロファイルを変更します。このコマンドはさらに、指定のとおり統計を更新します。

```
RUNSTATS ON TABLE db2user.employee
  ON COLUMNS (empl_address, (empl_title, empl_salary)) UPDATE
  PROFILE
```

17. 列 empl_address と列グループ (empl_title, empl_salary) のオプションを追加することにより、以前に登録した統計プロファイルを変更します。統計は収集されません。

```
RUNSTATS ON TABLE db2user.employee
ON COLUMNS (empl_address, (empl_title, empl_salary)) UPDATE
PROFILE ONLY
```

18. 表の統計プロファイルに記録されたオプションを使用して、その表の統計を収集します。

```
RUNSTATS ON TABLE db2user.employee USE PROFILE
```

19. 表のカatalog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、 *interm* 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
```

関連概念:

- 「パフォーマンス」の『自動統計収集』
- 「パフォーマンス」の『表データのサンプルでの統計の収集』
- 「パフォーマンス」の『統計プロファイルを使用した統計の収集』

関連タスク:

- 「パフォーマンス」の『カタログ統計の収集』

関連資料:

- 「管理 *SQL* ルーチンおよびビュー」の『DMIN_COPY_SCHEMA プロシージャ – 特定のスキーマとそのオブジェクトのコピー』
- 「管理 *API* リファレンス」の『db2Runstats API - 表および索引の統計情報の更新』
- 「管理 *SQL* ルーチンおよびビュー」の『RUNSTATS コマンド (ADMIN_CMD プロシージャを使用)』

SET CLIENT

バックエンド・プロセス用の接続設定を指定します。

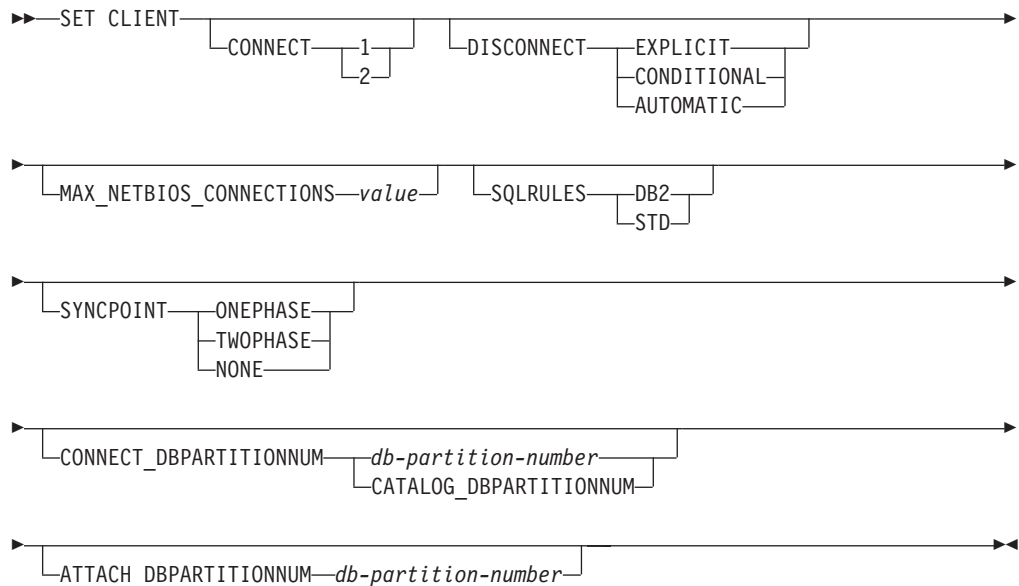
許可:

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。

MAX_NETBIOS_CONNECTIONS value

NetBIOS アダプターを使用したアプリケーションで可能な同時接続の最大数を指定します。最大値は、254 です。このパラメーターは、最初の NetBIOS 接続が行われる前に、設定する必要があります。最初の接続以降の変更は、無視されます。

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 0~999 (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atalog・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 0~999 (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

SET CLIENT

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、1 つ以上の接続がアクティブである場合には発行できません。

SET CLIENT が正常であると、それに続く作業単位の接続が、指定された接続の設定内容を使用します。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。

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 に置き換えられます。
- キーワード CATALOG_NODE は、CATALOG_DBPARTITIONNUM に置き換えられます。
- キーワード ATTACH_NODE は、ATTACH_DBPARTITIONNUM に置き換えられます。

関連資料:

- 「管理 API リファレンス」の『sqlesetc API - クライアント接続設定の指定』
- 815 ページの『TERMINATE 』
- 669 ページの『QUERY CLIENT 』

SET RUNTIME DEGREE

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。データベース・マネージャーの *max_querydegree* 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

関連タスク:

- 「管理ガイド: インプリメンテーション」の『照会のパーティション内並列処理
を使用可能にする』

関連資料:

- 「管理 API リファレンス」の『sqlesdeg API - SQL ステートメントの最大実行
時パーティション内並列処理レベル (つまり並列処理の程度) の設定』
- 570 ページの『LIST APPLICATIONS 』

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:

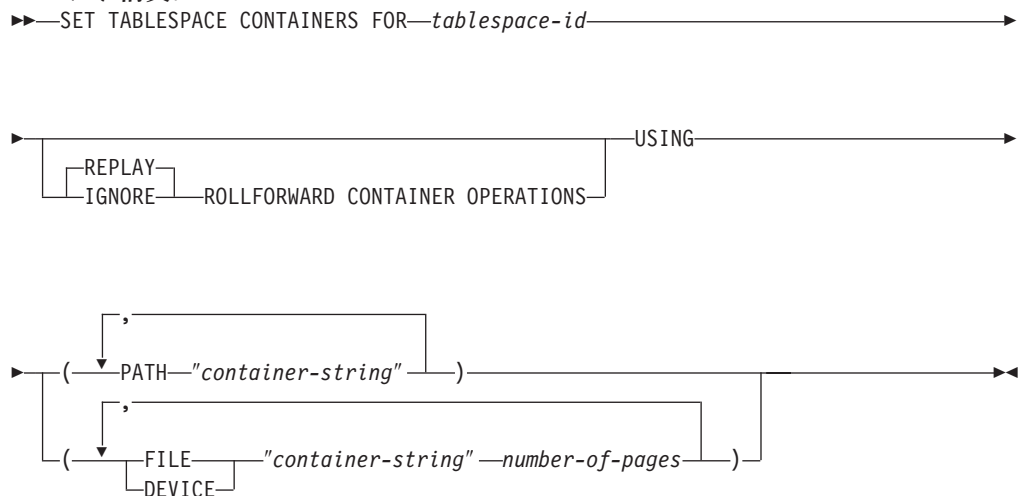
以下のいずれか。

- *sysadm*
- *sysctrl*

必要な接続:

データベース

コマンド構文:



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

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も可能です。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・コンテナのレコードを保持できます。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。

関連資料:

- 「管理 API リファレンス」の『sqlbstsc API - 表スペース・コンテナの設定』
- 382 ページの『BACKUP DATABASE 』
- 738 ページの『RESTORE DATABASE 』
- 756 ページの『ROLLFORWARD DATABASE 』

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。

関連資料:

- 「管理 SQL ルーチンおよびビュー」の『ADMIN_CMD プロシージャー – 管理コマンドの実行』
- 「管理 SQL ルーチンおよびビュー」の『SET TAPE POSITION コマンド (ADMIN_CMD プロシージャーを使用)』
- 560 ページの『INITIALIZE TAPE 』
- 755 ページの『REWIND TAPE 』

SET UTIL_IMPACT_PRIORITY

実行中のユーティリティー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ー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。

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
```

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。

関連資料:

- 606 ページの『LIST UTILITIES』
- 「パフォーマンス」の『util_impact_lim - インスタンス影響ポリシー構成パラメーター』

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われます。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クラッシュ・リカバリーのみを実行します。

関連概念:

- 「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『オンライン・スプリット・ミラーおよびサスペンド入出力サポートによる高可用性』

関連タスク:

- 「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『スプリット・ミラーをバックアップ・イメージとして使用する』
- 「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『スプリット・ミラーをスタンバイ・データベースとして使用する』
- 「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『スプリット・ミラーを使用したデータベースのクローン作成』

関連資料:

- 「管理 API リファレンス」の『db2SetWriteForDB API - データベースの入出力書き込みの中断または再開』

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/sql11ib/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`

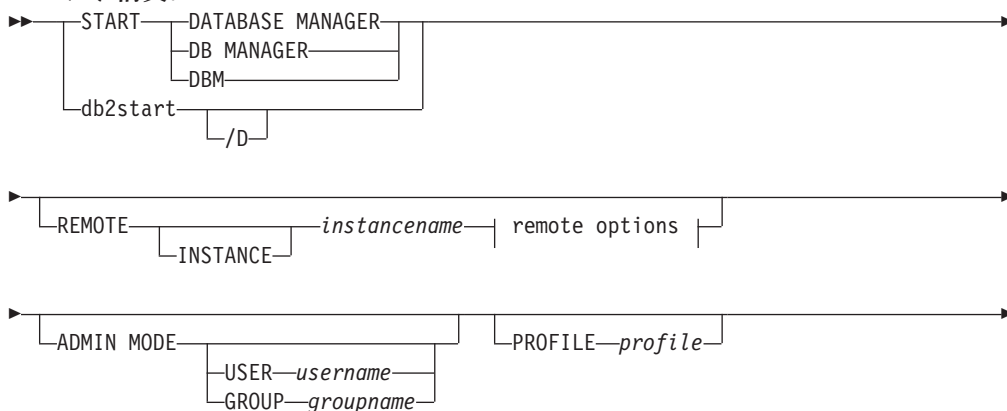
ADD DBPARTITIONNUM 開始オプションは、`sysadm` または `sysctrl` 権限のどちらかを必要とします。

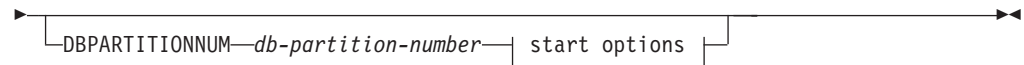
ユーザーはサービス開始に対する Windows オペレーティング・システム要件を満たす必要があります。拡張セキュリティーが無効になっている場合、ユーザーは Administrators、Server Operators、または Power Users グループのメンバーになっている必要があります。拡張セキュリティーが有効になっている場合、ユーザーはデータベースを始動するために Administratorsグループまたは DB2ADMNS グループのメンバーになっている必要があります。

必要な接続:

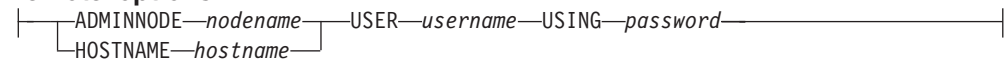
なし

コマンド構文:

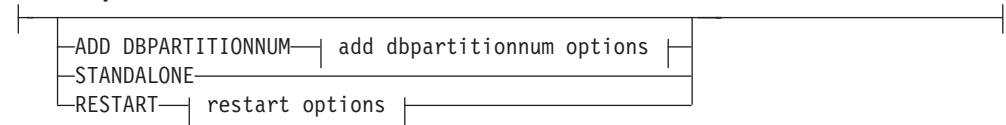




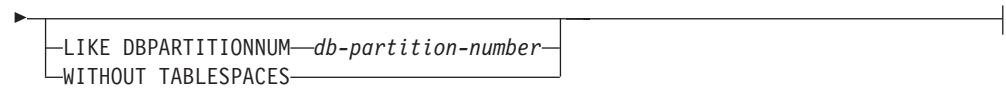
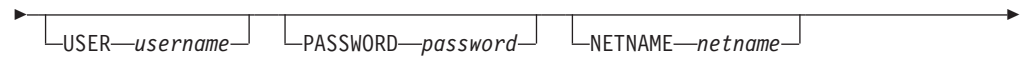
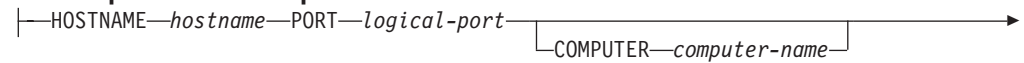
remote options:



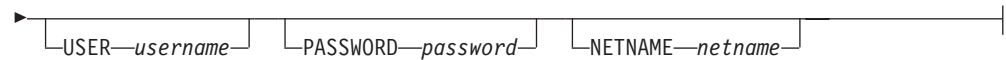
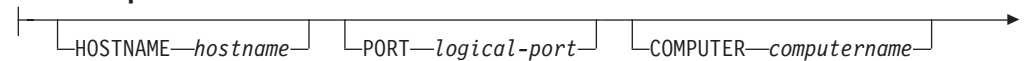
start options:



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

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 ～ 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。

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 ～ 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。

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 文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、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 文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*

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ホスト名を上書きするために使用するホスト名を指定します。

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 ～ 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:

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

START DATABASE MANAGER

ース・パーティションに SYSTEM TEMPORARY 表スペースを作成する場合は、別のデータベース・パーティション・サーバーに常駐するデータベース・パーティションの表スペース定義を入手するために、START DATABASE MANAGER はそのサーバーと通信しなければならないかもしれません。 start_stop_time データベース・マネージャー構成パラメーターを使用して、時間 (分) を指定します。他のデータベース・パーティション・サーバーはこの時間内で自動ストレージおよび表スペース定義に応答する必要があります。この時間を超えると、このコマンドは失敗します。そのような場合は、start_stop_time の値を増やして、コマンドを再発行してください。

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.timestamp.log エラー・ログ・ファイルにメッセージ (SQL1044N) が戻されます。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 に置き換えられます。
- キーワード ADDNODE は、ADD DBPARTITIONNUM に置き換えられます。
- キーワード NODENUM は、DBPARTITIONNUM に置き換えられます。

関連資料:

- 806 ページの『STOP DATABASE MANAGER 』
- 368 ページの『ADD DBPARTITIONNUM 』
- 「管理 API リファレンス」の『db2InstanceStart API - インスタンスの開始』

START HADR

データベースの HADR 操作を開始します。

許可:

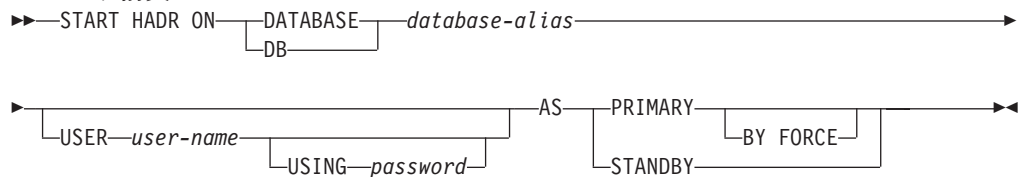
以下のいずれか。

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

必要な接続:

インスタンス。データベース接続が存在しない場合、このコマンドでデータベース接続が確立され、コマンド完了時にそのデータベース接続がクローズされます。

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 の認証パスワード。

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す。

注意: **START HADR** コマンドに **AS PRIMARY BY FORCE** オプションを指定する場合には、十分な注意が必要です。スタンバイ・データベースが 1 次データベースに変更されてから、**AS PRIMARY BY FORCE** オプションを指定した **START HADR** コマンドの発行により元のデータベースが再始動した場合、

データベースの 2 つのコピーが 1 次として独立して動作することになります。(これは分割ブレイン、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。いくつ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 コマンドを出すとき、対応するエラー・コードとして SQL01767N、SQL01769N、または SQL01770N に理由コード 98 が付いたものが生

成されることがあります。この理由コードは、コマンドの出されたサーバー上に HADR のためのインストール済みライセンスが存在しないことを示します。この問題を正すには、**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 ライセンスを含むサーバーのバージョンをインストールします。

関連タスク:

- 「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『高可用性災害時リカバリー (HADR) の初期設定』

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・システムでは、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・システムの場合のみ)。

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:

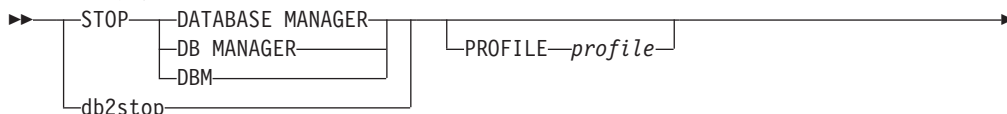
以下のいずれか。

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

必要な接続:

なし

コマンド構文:



STOP DATABASE MANAGER

```
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:

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・システム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、データベース・マネージャー構成パラメーターの *start_stop_time* 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。 **kill** 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。

関連資料:

- 484 ページの『**FORCE APPLICATION**』
- 796 ページの『**START DATABASE MANAGER**』
- 454 ページの『**DEACTIVATE DATABASE**』

- 471 ページの『DROP DBPARTITIONNUM VERIFY 』
- 「管理 API リファレンス」の『db2InstanceStop API - インスタンスの停止』

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。データベース接続が存在しない場合、このコマンドでデータベース接続が確立され、コマンド完了時にそのデータベース接続がクローズされます。

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 の認証パスワード。

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 <i>hadr_db_role</i> 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 <i>hadr_db_role</i> 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 コマンドを出すとき、対応するエラー・コードとして SQL01767N、SQL01769N、または SQL01770N に理由コード 98 が付いたものが生成されることがあります。この理由コードは、コマンドの出されたサーバー上に HADR のためのインストール済みライセンスが存在しないことを示します。この問題を正すには、**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 ライセンスを含むサーバーのバージョンをインストールします。

関連タスク:

- 「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『高可用性災害時リカバリ (HADR) ーの停止』

TAKEOVER HADR

HADR スタンバイ・データベースに対して、HADR ペアのための新しい HADR 1 次データベースとなるように指示します。

許可:

以下のいずれか。

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

必要な接続:

インスタンス。データベース接続が存在しない場合、このコマンドでデータベース接続が確立され、コマンド完了時にそのデータベース接続がクローズされます。

コマンド構文:

```

▶▶ TAKEOVER HADR ON DATABASE database-alias
                        DB
▶
  USER user-name
  USING password
  BY FORCE
▶▶

```

コマンド・パラメーター:

DATABASE *database-alias*

HADR 1 次データベースにする現在の HADR スタンバイ・データベース。

USER *user-name*

テークオーバー操作を開始するために使用するユーザー名。

USING *password*

user-name の認証パスワード。

BY FORCE

データベースが、元の HADR 1 次データベースがシャットダウンされたという確認を待機しないことを指定します。HADR ペアがピア状態でない場合には、このオプションが必要です。

使用上の注意:

TAKEOVER HADR コマンドをアクティブ・スタンバイに対して発行した場合に、可能性のある状態とオプションの各組み合わせごとにどんな動作になるかを、次の表に示します。このコマンドを非アクティブ・スタンバイ・データベースに対して発行すると、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	テークオーバーの動作
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。
ピア	いいえ	<p>1 次データベースとスタンバイ・データベースがその役割を切り替えます。</p> <p>テークオーバー中に障害が発生しなければ、データの損失はありません。しかし、テークオーバー中に障害が発生すると、データ損失の可能性があります。1 次とスタンバイの役割は変更される場合と変更されない場合があります。テークオーバー中に障害が発生して、1 次とスタンバイの役割が切り替えられた場合には、以下のガイドラインに従ってその障害を処理してください。</p> <ol style="list-style-type: none"> 1. テークオーバー中に障害が発生した場合、HADR データベースの役割は変更される場合と変更されない場合があります。可能なら、2 つのデータベースがオンラインであることを確認してください。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. 2 つのデータベースが両方共スタンバイの役割のまま終わることもあり得ます。その場合、1 次になっているほうのノードで TAKEOVER HADR コマンドを BY FORCE オプションで発行できます。その場合には、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。

スタンバイ 状態	BY FORCE オプション	テークオーバーの動作
ピア	はい	スタンバイ側は 1 次側に対して、それ自体 (1 次) をシャット ダウンするよう通知を送ります。スタンバイ側は 1 次側からの ログ受信を停止し、既に受信したログの再生を終了してから、1 次になります。スタンバイ側は、1 次側がテークオーバー通知 を受け取ったこと、またはそれがシャットダウンされたことを 確認するために、1 次側から確認通知が送られるのを待機しま せん。このため、テークオーバー時に 1 次側がトランザクショ ン処理中であった場合、1 次側を後にスタンバイとして再始動 できない可能性があります。1 次データベースをまずシャット ダウンして、それから TAKEOVER HADR コマンドを BY FORCE オプションで発行することをお勧めします。
リモート・ キャッチア ップ・ペン ディング	いいえ	エラー・メッセージが戻されます。
リモート・ キャッチア ップ・ペン ディング	はい	スタンバイ・データベースが 1 次データベースになります。

TAKEOVER HADR コマンドを出すとき、対応するエラー・コードとして SQL01767N、SQL01769N、または SQL01770N に理由コード 98 が付いたものが生成されることがあります。この理由コードは、コマンドの出されたサーバー上に HADR のためのインストール済みライセンスが存在しないことを示します。この問題を正すには、**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 ライセンスを含むサーバーのバージョンをインストールします。

関連タスク:

- 「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『高可用性災害時リカバリー (HADR) でのデータベース役割の切り替え』

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。

関連資料:

- 293 ページの『db2stop - DB2 の停止』

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、下位レベル・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。

関連資料:

- 「管理 API リファレンス」の『sqleuncd API - システム・データベース・ディレクトリーからのデータベースのアンカタログ』
- 408 ページの『CATALOG DATABASE 』
- 469 ページの『DROP DATABASE 』
- 508 ページの『GET DATABASE MANAGER CONFIGURATION 』
- 573 ページの『LIST DATABASE DIRECTORY 』
- 815 ページの『TERMINATE 』

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:

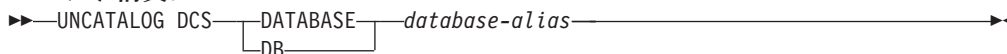
以下のいずれか。

- *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:



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。

関連資料:

- 「管理 API リファレンス」の『**sqlgdel API - データベース接続サービス (DCS) ディレクトリーからのデータベースのアンカタログ**』

- 412 ページの『CATALOG DCS DATABASE 』
- 508 ページの『GET DATABASE MANAGER CONFIGURATION 』
- 815 ページの『TERMINATE 』
- 816 ページの『UNCATALOG DATABASE 』
- 582 ページの『LIST DCS DIRECTORY 』

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。

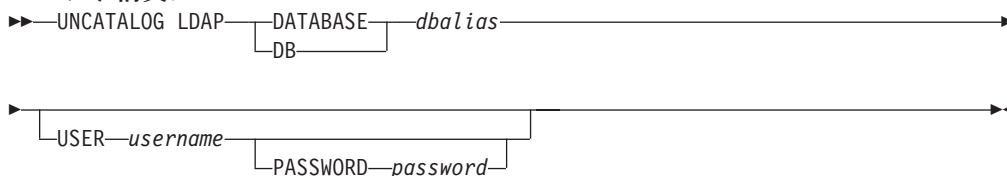
許可:

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。

関連概念:

- 「管理ガイド: インプリメンテーション」の『Lightweight Directory Access Protocol (LDAP) 』

関連タスク:

- 「管理ガイド: インプリメンテーション」の『LDAP ディレクトリーからのデータベースの登録解除』

関連資料:

- 「管理 API リファレンス」の『db2LdapUncatalogDatabase API - LDAP サーバーからのデータベース登録の解除』
- 415 ページの『CATALOG LDAP DATABASE 』
- 419 ページの『CATALOG LDAP NODE 』
- 822 ページの『UNCATALOG LDAP NODE 』

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。

関連概念:

- 「管理ガイド: インプリメンテーション」の『Lightweight Directory Access Protocol (LDAP) 』

関連資料:

- 「管理 API リファレンス」の『db2LdapUncatalogNode API - LDAP サーバーからのノード名に対応する別名の削除』
- 415 ページの『CATALOG LDAP DATABASE 』
- 820 ページの『UNCATALOG LDAP DATABASE 』
- 419 ページの『CATALOG LDAP NODE 』

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。

関連資料:

- 「管理 API リファレンス」の『**sqluncn API - ノード・ディレクトリーからの項目のアンカタログ**』
- 421 ページの『**CATALOG LOCAL NODE** 』
- 423 ページの『**CATALOG NAMED PIPE NODE** 』
- 426 ページの『**CATALOG TCPIP/TCPIP4/TCPIP6 NODE** 』
- 508 ページの『**GET DATABASE MANAGER CONFIGURATION** 』
- 815 ページの『**TERMINATE** 』

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:

```
►►—UNCATALOG—

|        |
|--------|
| USER   |
| SYSTEM |

—ODBC DATA SOURCE—data-source-name—►►
```

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。

関連資料:

- 425 ページの『CATALOG ODBC DATA SOURCE』
- 595 ページの『LIST ODBC DATA SOURCES』

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:

UNQUIESCE

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。

関連資料:

- 670 ページの『QUIESCE 』
- 「管理 API リファレンス」の『db2DatabaseUnquiesce API - データベースの非静止』
- 「管理 API リファレンス」の『db2InstanceUnquiesce API - インスタンスの静止解除』
- 「管理 SQL ルーチンおよびビュー」の『UNQUIESCE DATABASE コマンド (ADMIN_CMD プロシージャを使用)』

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 を始動するときに使用するものとして指定できます。

- DB2 サーバー・システムの名前 - db2system
- DAS 管理者権限グループ名 - dasadm_group
- スケジューラー・モード - sched_enable
- ツール・カタログ・データベース・インスタンス - toolscat_inst
- ツール・カタログ・データベース - toolscat_db
- ツール・カタログ・データベース・スキーマ - toolscat_schema
- 実行有効期限切れタスク - exec_exp_task
- スケジューラー・ユーザー ID - sched_userid
- 認証タイプ DAS - authentication

以下の DAS 構成パラメーターは、元々指定可能なもので、後で DAS がオンラインになっているときに変更できます。

- DAS ディスカバリー・モード - discover
- SMTP サーバー - smtp_server
- Java 開発キット・インストール・パス DAS - jdk_path
- 連絡先リストのロケーション -contact_host
- DAS コード・ページ - das_codepage
- DAS テリトリー - das_territory

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:

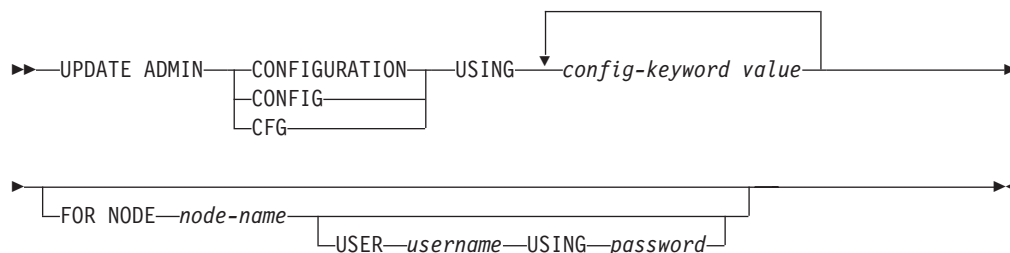
dasadm

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:

UPDATE ADMIN CONFIGURATION



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。

関連資料:

- 486 ページの『GET ADMIN CONFIGURATION 』
- 726 ページの『RESET ADMIN CONFIGURATION 』
- 「パフォーマンス」の『構成パラメーターのサマリー』

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:

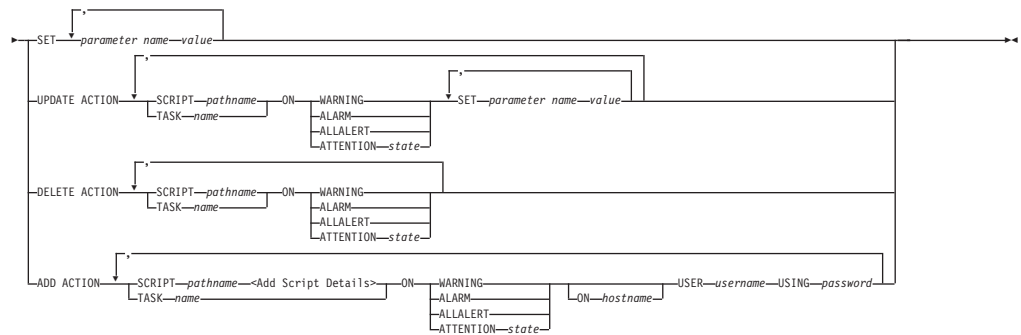
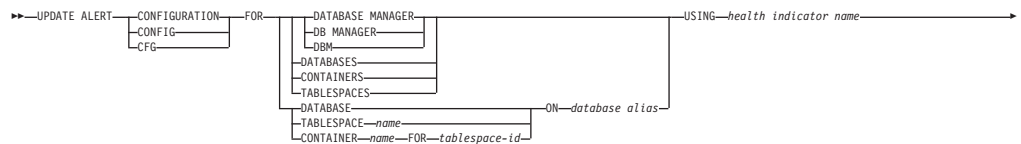
以下のいずれか。

- sysadm
- sysmaint
- sysctrl

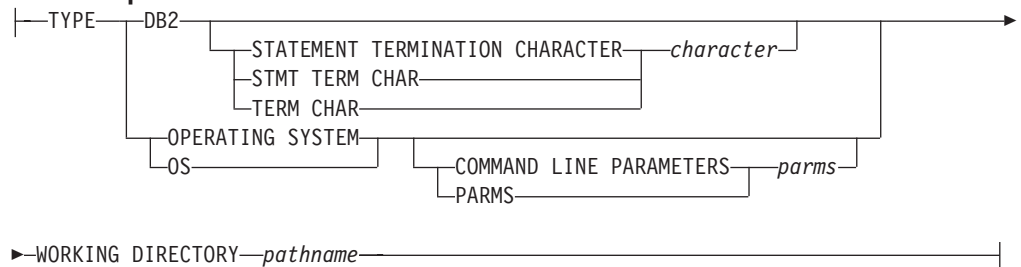
必要な接続:

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:



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" 文節を使って定義されます。

UPDATE ALERT CONFIGURATION

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 name ON database alias"` 文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 name ON database alias"` 文節を使って定義されます。

DATABASE ON *database alias*

`"ON database alias"` 文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 *name* FOR *tablespace-id* ON *database alias*

`"ON database alias"` 文節を使って指定したデータベース上で、`"FOR tablespace-id"` 文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 *name* ON *database alias*

`"ON database alias"` 文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。例えば、次のようなものがあります。

```
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文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

有効なタイプは OS または DB2。

- WORKINGDIR

- TERMCHAR

- CMDLINEPARMS

オペレーティング・システム・スクリプトに対して指定するコマンド行パラメーターが、デフォルトで指定されるパラメーターに先行します。オペレーティング・システム・スクリプトに送られるパラメーターは、以下のとおりです。

- ユーザーの指定するパラメーターのリスト
- ヘルス・インディケーターの短縮名
- 完全修飾オブジェクト名
- ヘルス・インディケーターの値
- アラート状態

- USERID

- PASSWORD

- SYSTEM

UPDATE ACTION TASK *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文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 *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。

UPDATE ALERT CONFIGURATION

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、以下の文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、以下の文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 | ATTENTION *state*]

アクションが実行される条件を指定します。しきい値ベースの 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、これは表に記されている、各状態ベース HI に提供される数値状態 (表スペース状態など) になります。

関連タスク:

- 「システム・モニター ガイドおよびリファレンス」の『クライアント・アプリケーションを使用したヘルス・インディケーターの構成』

関連資料:

- 「管理 SQL ルーチンおよびビュー」の『ADMIN_CMD プロシージャ - 管理コマンドの実行』
- 「管理 API リファレンス」の『db2UpdateAlertCfg API - ヘルス・インディケーターのアラート構成設定の更新』

UPDATE ALERT CONFIGURATION

- 「管理 SQL ルーチンおよびビュー」の『UPDATE ALERT CONFIGURATION コマンド (ADMIN_CMD プロシージャを使用)』

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—database-alias—USING————→
                                |
                                DB
▶—HOSTNAME—hostname—PORT—port-number————→▶

```

コマンド・パラメーター:

DATABASE *database-alias*

代替サーバーを更新するデータベースの別名を指定します。

HOSTNAME *hostname*

そのデータベースの代替サーバーが存在するノードの完全修飾ホスト名または IP アドレスを指定します。

PORT *port-number*

データベース・マネージャー・インスタンスの代替サーバーのポート番号を指定します。

例:

以下の例は、ホスト名 *montero* とポート *20396* を使用して、*SAMPLE* データベースの代替サーバーを更新します。

```
db2 update alternate server for database sample using hostname montero port 20396
```

以下の 2 つの例は、*SAMPLE* データベースの代替サーバーをリセットします。

```
db2 update alternate server for database sample using hostname NULL port NULL
```

または

```
db2 update alternate server for database sample using hostname "" port NULL
```

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。

UPDATE ALTERNATE SERVER FOR DATABASE

- 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。

関連資料:

- 「管理 API リファレンス」の『db2UpdateAlternateServerForDB API - 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別名の代替サーバーの更新』
- 435 ページの『CREATE DATABASE 』

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。

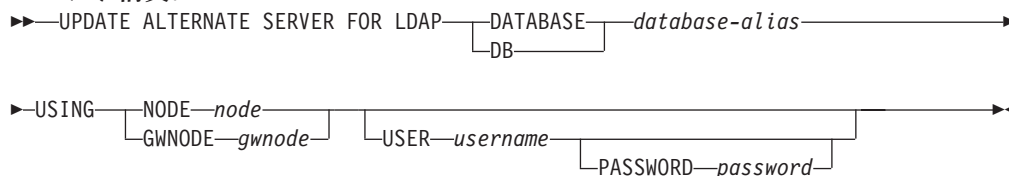
許可:

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。

関連タスク:

- 「管理ガイド: インプリメンテーション」の『他のサーバーへの LDAP クライアントのリルート』

関連資料:

- 415 ページの『CATALOG LDAP DATABASE』
- 149 ページの『db2ldcfg - LDAP 環境の構成』

UPDATE ALTERNATE SERVER FOR LDAP DATABASE

- 「管理 API リファレンス」の『db2LdapUpdateAlternateServerForDB API - LDAP サーバーでのデータベースに対応する代替サーバーの更新』

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。

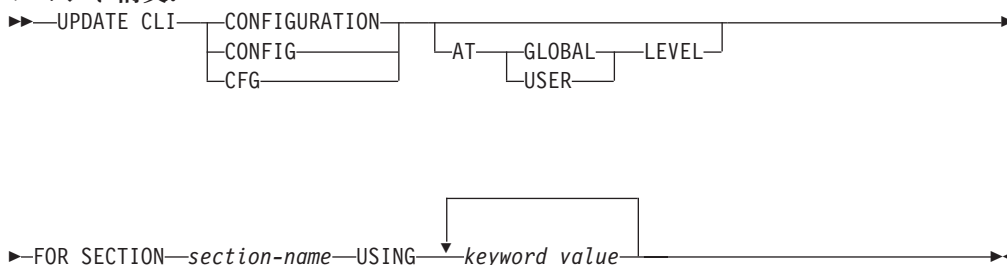
許可:

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ストリングである場合には、ストリング全体を二重引用符で囲む必要があります。例えば、次のようなものがあります。

```
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 文節を指定する必要があります。

関連資料:

- 496 ページの『GET CLI CONFIGURATION 』
- 696 ページの『REFRESH LDAP 』

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)。

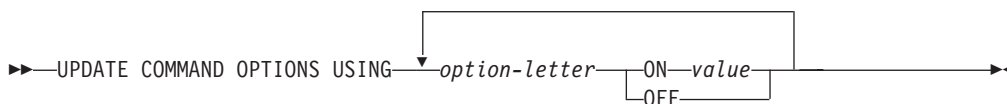
許可:

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

プシオンでは、*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。

関連資料:

- 572 ページの『LIST COMMAND OPTIONS 』

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:

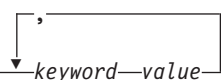
なし。

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:

►► UPDATE CONTACT *name* USING *keyword* *value* ◄◄



コマンド・パラメーター:

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。

関連資料:

- 「管理 API リファレンス」の『db2UpdateContact API - 連絡先の属性の更新』
- 「管理 SQL ルーチンおよびビュー」の『UPDATE CONTACT コマンド (ADMIN_CMD プロシージャを使用)』

第 3 章 CLP コマンド 843

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、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:

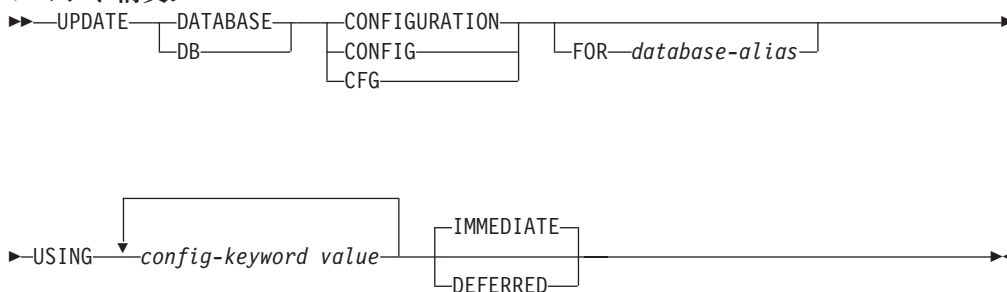
以下のいずれか。

- *sysadm*
- *sysctrl*
- *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:

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。

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。

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。

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
```

複数パーティション・データベース環境で構成パラメーターを変更するには、**db2_all** コマンドを使用してください。**db2_all** コマンドを使用すると、すべてのデータベース・パーティションに対して更新が発行されます。

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複数パーティション・データベース環境で変更する場合は、次を使用します。

```
db2_all ";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"
```

オプションとして、SYSIBMADM.DBCFG ビューを利用すれば、db2_all を使用することなく全パーティションからのデータを取得できます。

UPDATE DATABASE CONFIGURATION

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、ファイルがメモリにロードされた後にのみ有効になります。これを行う前にすべてのアプリケーションはデータベースから切断されている必要があります。オンラインで構成できるパラメーターと構成できないパラメーターについては、構成パラメーターの一覧をご覧ください。

例えば、SALES データベースの *sortheap* データベース構成パラメーターをオンラインで変更するには、次のようなコマンドを入力します。

```
db2 connect to sales
db2 update db cfg using sortheap 1000
db2 connect reset
```

エラーが発生した場合、データベース構成ファイルは変更されません。チェックサムが無効な場合、データベース構成ファイルは更新できません。適切なコマンドを使用しないでデータベース構成ファイルを変更するとこれが発生することがあります。これが発生する場合、データベースをリストアしてデータベース構成ファイルをリセットする必要があります。

関連概念:

- 「管理ガイド: インプリメンテーション」の『rah および db2_all コマンドの概要』

関連タスク:

- 「パフォーマンス」の『構成パラメーターによる DB2 の構成』

関連資料:

- 「管理 API リファレンス」の『db2CfgSet API - データベース・マネージャー構成パラメーター、あるいはデータベース構成パラメーターの設定』
- 「パフォーマンス」の『構成パラメーターのサマリー』
- 502 ページの『GET DATABASE CONFIGURATION 』
- 730 ページの『RESET DATABASE CONFIGURATION 』
- 「管理 SQL ルーチンおよびビュー」の『DBCFCG 管理ビュー - データベース構成パラメーター情報の検索』
- 「管理 SQL ルーチンおよびビュー」の『UPDATE DATABASE CONFIGURATION コマンド (ADMIN_CMD プロシーチャーを使用)』

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。

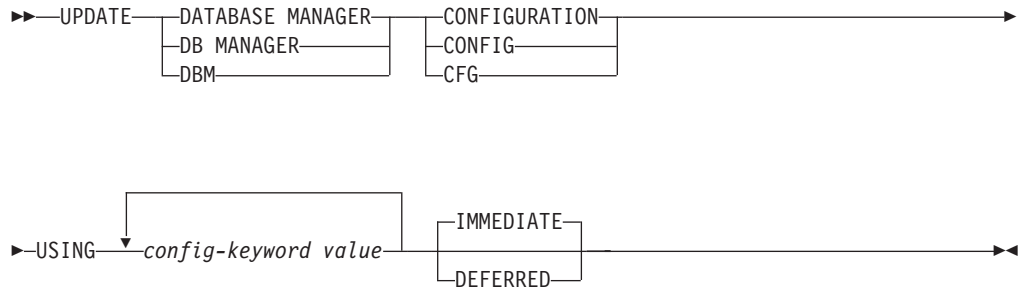
許可:

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:

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。

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。

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

UPDATE DATABASE MANAGER CONFIGURATION

パー、クライアント、またはリモート・クライアントを持つサーバー) に適したこれらのパラメーターの値については、個々の構成パラメーターの説明を参照してください。

すべてのパラメーターを更新できるわけではありません。

データベース・マネージャー構成ファイルへの変更の一部は、ファイルがメモリーにロードされた後にのみ有効になります。オンラインで構成できるパラメーターと構成できないパラメーターについては、構成パラメーターの一覧をご覧ください。即時にリセットされないサーバー構成パラメーターは、**db2start** の実行中にリセットされます。クライアント構成パラメーターの場合、パラメーターは次にアプリケーションを開始するときにリセットされます。クライアントがコマンド行プロセッサである場合は、**TERMINATE** を呼び出す必要があります。

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。

関連タスク:

- 「パフォーマンス」の『構成パラメーターによる DB2 の構成』

関連資料:

- 「管理 *API* リファレンス」の『db2CfgSet API - データベース・マネージャー構成パラメーター、あるいはデータベース構成パラメーターの設定』
- 「パフォーマンス」の『構成パラメーターのサマリー』
- 508 ページの『GET DATABASE MANAGER CONFIGURATION 』
- 732 ページの『RESET DATABASE MANAGER CONFIGURATION 』
- 815 ページの『TERMINATE 』
- 「管理 *SQL* ルーチンおよびビュー」の『DBMCFG 管理ビュー - データベース・マネージャー構成パラメーター情報の検索』
- 「管理 *SQL* ルーチンおよびビュー」の『UPDATE DATABASE MANAGER CONFIGURATION コマンド (ADMIN_CMD プロシージャーを使用)』

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:

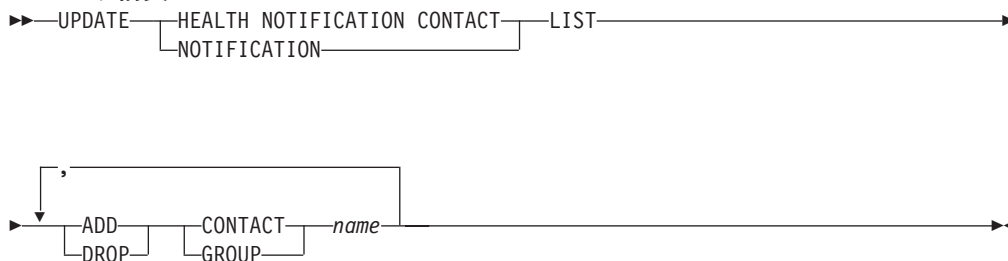
以下のいずれか。

- 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。

関連タスク:

- 「システム・モニター ガイドおよびリファレンス」の『ヘルス・アラート通知の使用可能化』

関連資料:

- 「管理 SQL ルーチンおよびビュー」の『ADMIN_CMD プロシージャ - 管理コマンドの実行』
- 「管理 API リファレンス」の『db2UpdateHealthNotificationList API - ヘルス・アラート通知を送信できる連絡先リストの更新』
- 「管理 SQL ルーチンおよびビュー」の『UPDATE HEALTH NOTIFICATION CONTACT LIST コマンド (ADMIN_CMD プロシージャを使用)』

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:

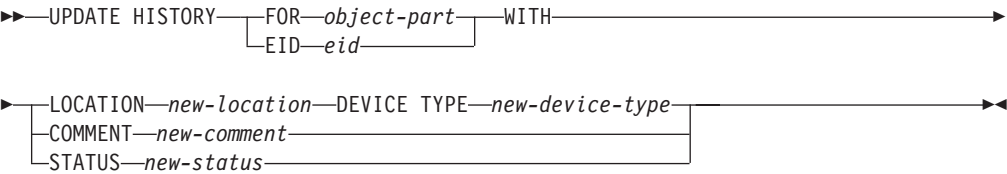
以下のいずれか。

- *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。

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 | TSM |
| 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。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。ほとんどの項目はアクティブです。

I 非アクティブ。アクティブ・ログ・チェーン上にもはやないバックアップ・イメージは、非アクティブになります。

E 期限切れ。バックアップ・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。リカバリーに使用できないバックアップ・イメージは、削除済みとしてマークされることになります。

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以来に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、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。

関連概念:

- 「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『バックアップとリカバリーの計画の作成』

関連資料:

- 「管理 SQL ルーチンおよびビュー」の『ADMIN_CMD プロシージャー – 管理コマンドの実行』
- 「管理 SQL ルーチンおよびビュー」の『UPDATE HISTORY コマンド (ADMIN_CMD プロシージャーを使用)』

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。

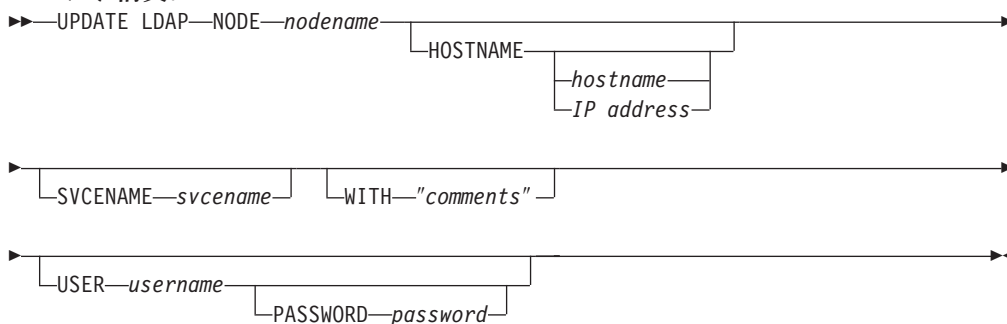
許可:

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* または **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。

関連資料:

- 「管理 API リファレンス」の『db2LdapUpdate API - LDAP サーバー上の DB2 サーバーの属性の更新』
- 697 ページの『REGISTER 』
- 458 ページの『DEREGISTER 』

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*

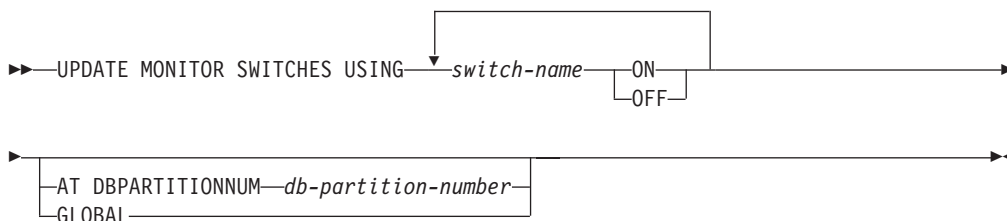
必要な接続:

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:



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

バッファー・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・システム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 DBPARTITIONNUM の代わりに NODE を使用できます。

関連概念:

- 「システム・モニター ガイドおよびリファレンス」の『システム・モニター・スイッチ』

関連タスク:

- 「システム・モニター ガイドおよびリファレンス」の『クライアント・アプリケーションからのモニター・スイッチの設定』

関連資料:

- 532 ページの『GET SNAPSHOT 』
- 523 ページの『GET MONITOR SWITCHES 』
- 「管理 API リファレンス」の『db2MonitorSwitches API - モニター・スイッチ設定の取得あるいは更新』

UPDATE MONITOR SWITCHES

第 4 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

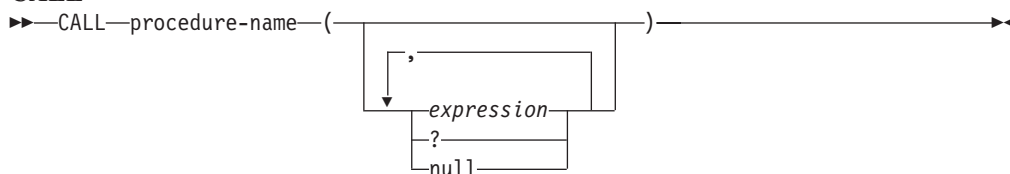
この節では、コマンド行から SQL (構造化照会言語) ステートメントを使用する方法について説明します。これらのステートメントは、オペレーティング・システムのコマンド・プロンプトから直接実行できるもので、コマンドをアプリケーション・プログラムに書き込む場合とほぼ同じようにして、データベースの表や索引やビューに格納されている情報を定義したり処理したりするのに使用できます。情報については追加や削除や更新ができます。レポートは表の内容から生成できます。

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。

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、861 ページの表 15 の CLP 列にリストされています。SQL ステートメントの構文は、コマンド行から実行できるものであれ、ソース・プログラムに組み込まれているものであれ、すべてが「SQL リファレンス」に記述されています。多くの場合、組み込み SQL ステートメントと CLP SQL ステートメントの構文は同じです。ただし、ホスト変数、パラメーター・マーカー、記述子名、およびステートメント名は、組み込み SQL にのみ適用できます。CALL、CLOSE、CONNECT、DECLARE CURSOR、FETCH、および OPEN の構文は、組み込み型のものと CLP によって実行されるものとは違います。これらのステートメントの CLP 構文を次に示します。

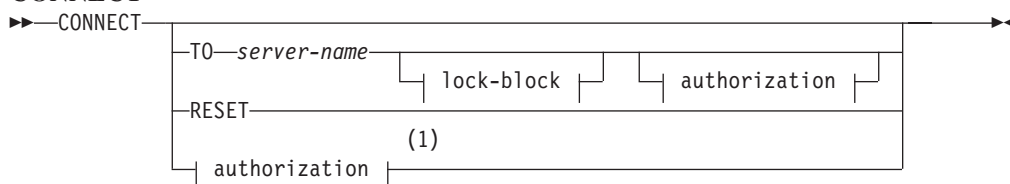
CALL



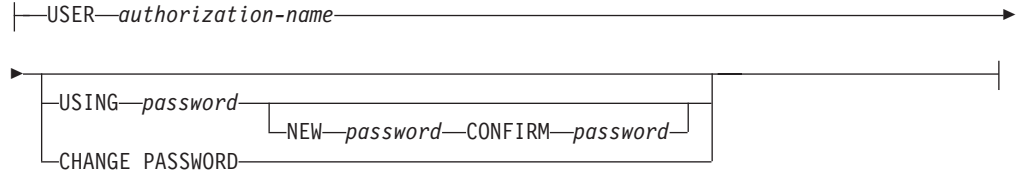
CLOSE



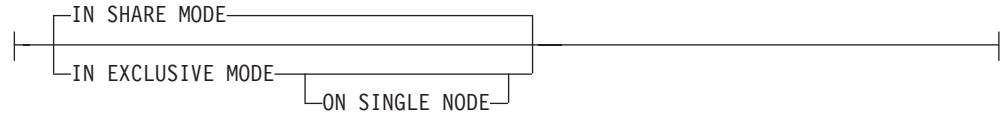
CONNECT



authorization:



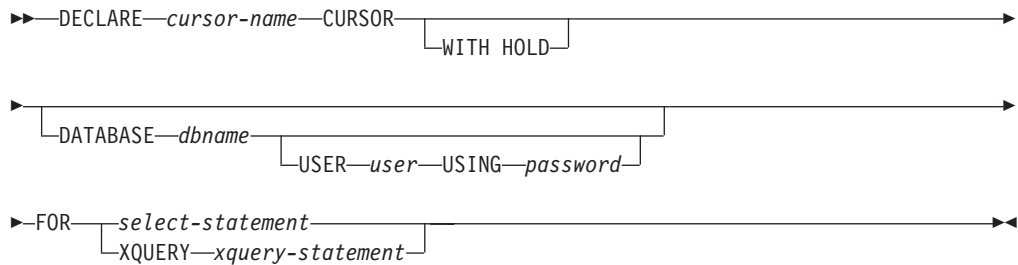
lock-block:



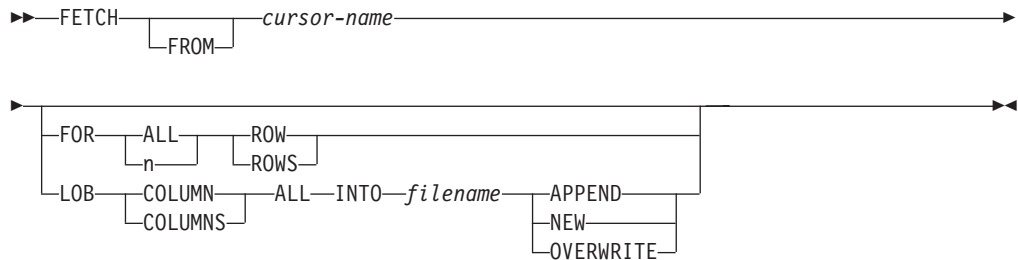
注:

- 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。
 - プロシーチャーの各 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。

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

- プロシーチャーの各 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・プロシーチャー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チャーが呼び出されると、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チャーを作成します。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チャー 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 文節は、それ以降のカーソルからのロード操作でカーソルが使用される場合にのみ適用されます。

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

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、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 文節を指定する場合は、次の行だけが取り出されます。
 - 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取り出されます。この場合、*filename* は LOB 文節に指定され、*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。) ファイルに取り出せる 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。

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

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。

```
db2clpcs.bnd +
db2clpr.r.bnd +
db2clpur.bnd +
db2clprs.bnd +
db2clpns.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 15. SQL ステートメント (DB2)

SQL ステートメント	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
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()	

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

表 15. SQL ステートメント (DB2) (続き)

SQL ステートメント	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
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 15. SQL ステートメント (DB2) (続き)

SQL ステートメント	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
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

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

表 15. SQL ステートメント (DB2) (続き)

SQL ステートメント	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 ヤー
-------------	-----------------	---------------------------	---	---------------------

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 方向に取り出すことができますが、DB2 CLI の `SQLExtendedFetch()` および `SQLFetchScroll()` を使用すると、配列の形で取り出すことができます。さらに、どの方向でも、また結果セットのどの位置でも取り出しができます。
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。

関連資料:

- 「SQL リファレンス 第 2 巻」の『CONNECT (タイプ 2) ステートメント』
- 「SQL リファレンス 第 2 巻」の『CLOSE ステートメント』
- 「SQL リファレンス 第 2 巻」の『CONNECT (タイプ 1) ステートメント』
- 「SQL リファレンス 第 2 巻」の『DECLARE CURSOR ステートメント』
- 「SQL リファレンス 第 2 巻」の『FETCH ステートメント』
- 「SQL リファレンス 第 2 巻」の『OPEN ステートメント』
- 「SQL リファレンス 第 2 巻」の『SELECT ステートメント』

付録 A. 構文図の見方

本書を通じて、構文の説明には次のように定義される構造の図が使用されます。

構文図は、左から右、上から下に、線に沿って読みます。

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

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

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

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

構文フラグメントは、記号 |— で始まり、記号 —| で終わります。

必須項目は、横線 (メインパス) 上に示されます。

\blacktriangleright —required_item— \blacktriangleleft

オプション項目は、メインパスの下に示されます。

\blacktriangleright —required_item—
 |optional_item|— \blacktriangleleft

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。

\blacktriangleright —required_item—
 |optional_item|— \blacktriangleleft

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。

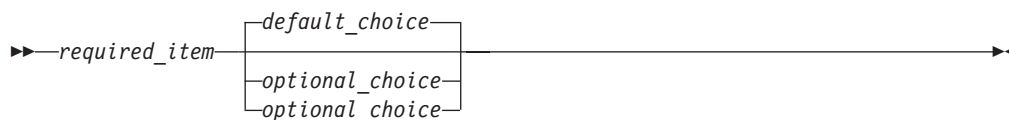
項目から 1 つを選択しなければならない 場合、スタックの項目の 1 つはメインパス上に示されます。

\blacktriangleright —required_item—
 |required_choice1|
 |required_choice2|— \blacktriangleleft

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。

\blacktriangleright —required_item—
 |optional_choice1|
 |optional_choice2|— \blacktriangleleft

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います (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
```

付録 B. 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～z、A～Z、0～9、@、#、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。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、元名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、1～128 文字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 1～30 文字です。

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～z、A～Z、0～9、@、#、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)。
- !、%、(、)、{、}、-、.、^ という追加の文字のいずれかを含むユーザー ID およびグループは、コマンド行プロセッサーで入力するときに引用符で区切る必要があります。
- ストリングの最初の文字はアルファベット、@、#、または \$ にしなければなりません。数字や文字順序列の SYS、DBM、IBM などは不可です。

- 認証 ID の長さは、Windows 32 ビット・オペレーティング・システムでは 30 文字、その他のオペレーティング・システムでは 8 文字を超えてはなりません。
- グループ ID の長さは、30 文字を超えてはなりません。

関連資料:

- 435 ページの『CREATE DATABASE 』
- 451 ページの『CREATE TOOLS CATALOG 』

付録 C. ファイル・タイプ修飾子

以下のトピックでは、ロード、インポート、およびエクスポート・ユーティリティのファイル・タイプ修飾子について説明します。

ロード・ユーティリティ用のファイル・タイプ修飾子

表 16. ロード・ユーティリティで有効なファイル・タイプ修飾子: すべてのファイル・フォーマット

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。 <code>SAVECOUNT > 0</code> 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によって生成されます。この修飾子は、 <code>generatedmissing</code> または <code>generatedoverride</code> 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、ユーティリティは、生成列のデータが入力データ・ファイルに入っていない (NULL も入っていない) ものを見なします。この結果、すべての生成列の値はユーティリティによって生成されます。この修飾子は、 <code>generatedignore</code> または <code>generatedoverride</code> 修飾子とともに使用することはできません。

ロード・ユーティリティー用のファイル・タイプ修飾子

表 16. 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) ものを見なし、行ごとに値を生成します。この動作は、GENERATED ALWAYS および GENERATED BY DEFAULT のどちらの ID 列の場合も同じです。この修飾子は、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 16. 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、レジストリー変数 DB2 INDEX FREE は indexfreespace 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 文節は、「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 文節によって、LOBSINFILE 動作が暗黙的に活動化されます。LOBS FROM 文節は、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	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。

ロード・ユーティリティー用のファイル・タイプ修飾子

表 16. 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、最も小さい行の大きさが 3 000 バイトで、<i>x</i> の値が 50 である場合)、その行は新しい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>
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。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。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、SQLCODE SQL3242W) 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、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>

表 16. 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
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>
totalreespace= <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>

ロード・ユーティリティー用のファイル・タイプ修飾子

表 16. 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
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、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 17. 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=x	<p>x は ASCII 文字ストリングです。この値は、入力データ・セット内のデータのコード・ページとして解釈されます。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>
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 有効な日付エレメントは次のとおりです。</p> <p>YYYY - 年 (0000 から 9999 の範囲の 4 桁の数) M - 月 (1 から 12 の範囲の 1 桁または 2 桁の数) MM - 月 (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。 他の日または月エレメントとは相互に排他的)</p> <p>デフォルト値の 1 が、指定されない各エレメントに割り当てられます。日付形式の例を以下に示します。</p> <p>"D-M-YYYY" "MM.DD.YYYY" "YYYYDDD"</p>

表 17. 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
dumpfile = <i>x</i>	<p><i>x</i> 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。
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。
fastparse	<p>ユーザー提供の列値に対して簡略化された構文チェックが実行され、パフォーマンスが向上します。このオプションの下でロードした表は、体系的な正確さが確保されて、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・チェックが必ずユーティリティーで実行されます。正しい形式のデータが正しくロードされます。</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>

ロード・ユーティリティー用のファイル・タイプ修飾子

表 17. 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
timeformat="x"	<p><i>x</i> 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> <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 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>TT - 午前/午後の指定子 (AM または PM)</p> <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> <p>"HH:MM:SS" "HH.MM TT" "SSSSS"</p>

表 17. 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・ファイル内のタイム・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、 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、 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、デフォルト値として 1 が割り当てられます。値が指定されていない MMM エレメントには、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>"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>

ロード・ユーティリティー用のファイル・タイプ修飾子

表 17. 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>

表 18. 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>

表 18. 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
nullindchar= <i>x</i>	<p><i>x</i> は単一文字です。 NULL 値を示す文字を <i>x</i> に変更します。 <i>x</i> のデフォルト値は Y です。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>
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= <i>x</i>	<p><i>x</i> は、32,767 以下の整数です。各行ごとに <i>x</i> 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 (後方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 (後方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>

ロード・ユーティリティー用のファイル・タイプ修飾子

表 18. 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
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> <p>+ = 0xC 0xA 0xE 0xF - = 0xD 0xB</p> <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 19. 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><i>x</i> は単一文字のストリング区切り文字です。デフォルト値は二重引用符 (") です。指定した文字は、文字ストリングを囲むために、二重引用符の代わりに使用されます。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> <p>modified by chardel""</p> <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> <p>modified by chardel''</p>
coldelx	<p><i>x</i> は単一文字の列区切り文字です。デフォルト値はコンマ (,) です。指定した文字は、列の終わりを表すために、コンマの代わりに使用されます。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>
decptr	<p><i>x</i> は、小数点文字としてピリオドの代わりに使用される単一の置換文字です。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>

表 19. 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
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 <i>DEL</i> 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 <i>CHAR</i>、<i>VARCHAR</i>、<i>LONG VARCHAR</i>、または <i>CLOB</i> 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 <i>NULL</i> 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、<i>TABLE1</i> 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>
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ペンダー・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止します。

表 20. 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止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>

ロード・ユーティリティー用のファイル・タイプ修飾子

表 20. 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 (続き)

修飾子	説明
nochecklengths	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。

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. この文字は、ソース・データのコード・ページで指定してください。

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。

表 21. *codepage* および *usegraphiccodepage* 使用時の *LOAD* 動作

codepage=N	usegraphiccodepage	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。
なし	あり	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。 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。 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。
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ロードすると壊れます。

関連資料:

- 「管理 API リファレンス」の『db2Load API - 表へのデータのロード』
- 902 ページの『データ移動での区切り文字の制限』
- 608 ページの『LOAD 』

インポート・ユーティリティ用のファイル・タイプ修飾子

表 22. 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

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いない場合、インポート操作は失敗します。トランザクション・ログは、COMMITCOUNT によって指定された行数か、または COMMITCOUNT が指定されていない場合はデータ・ファイルの行数を入れる十分な大きさが必要です。したがって、トランザクション・ログのオーバーフローを避けるために、COMMITCOUNT オプションを指定することをお勧めします。</p> <p>この修飾子は、INSERT_UPDATE モード、階層表、および修飾子 usedefaults、identitymissing、identityignore、generatedmissing、generatedignore とは互換性がありません。</p>
generatedignore	<p>この修飾子は、インポート・ユーティリティに、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、それらを見捨てるべきことを知らせます。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によって生成されます。この修飾子は、generatedmissing 修飾子と共に使用することはできません。</p>
generatedmissing	<p>この修飾子が指定されている場合、ユーティリティは、生成列のデータが入力データ・ファイルに入っていない (NULL も入っていない) ものを見なし、行ごとに値を生成します。この修飾子は、generatedignore 修飾子と共に使用することはできません。</p>
identityignore	<p>この修飾子は、インポート・ユーティリティに、ID 列のデータはデータ・ファイルに存在するが、それらを見捨てるべきことを知らせます。この結果として、すべて ID 値はこのユーティリティによって生成されます。この動作は、GENERATED ALWAYS および GENERATED BY DEFAULT のどちらの ID 列の場合も同じです。つまり、GENERATED ALWAYS 列の場合には、リジェクトされる行はありません。この修飾子は、identitymissing 修飾子とともに使用することはできません。</p>
identitymissing	<p>この修飾子を指定すると、ユーティリティは、ID 列のデータが入力データ・ファイルに入っていない (NULL も入っていない) ものを見なし、行ごとに値を生成します。この動作は、GENERATED ALWAYS および GENERATED BY DEFAULT のどちらの ID 列の場合も同じです。この修飾子は、identityignore 修飾子とともに使用することはできません。</p>

表 22. 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
lobsinfile	<p><i>lob-path</i> には、LOB データの入ったファイルへのパスを指定します。</p> <p>各パスには、データ・ファイル内で LOB ロケーション指定子 (LLS) によって示される 1 つ以上の LOB の入った、少なくとも 1 つのファイルが組み込まれます。LLS は、LOB ファイル・パスに保管されるファイル内の LOB のロケーションのストリング表現です。LLS の形式は、<i>filename.ext.nnn.mmm/</i> です。<i>filename.ext</i> は LOB を収めたファイルの名前、<i>nnn</i> はファイル内の LOB のオフセット (バイト単位)、<i>mmm</i> は LOB の長さ (バイト単位) を表します。例えば、ストリング <i>db2exp.001.123.456/</i> がデータ・ファイルに保管される場合、LOB はファイル <i>db2exp.001</i> のオフセット 123 に位置し、456 バイト長です。</p> <p>LOBS FROM 文節は、「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 文節によって、LOBSINFILE 動作が暗黙的に活動化されます。LOBS FROM 文節は、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	<p>単一の副表にインポートするときのみに有効です。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。</p>
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	<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>
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>

インポート・ユーティリティ用のファイル・タイプ修飾子

表 22. 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
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>
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は見なされず、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。

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 23. 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)

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、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。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"> codepage 修飾子を lobsinfile 修飾子と一緒に使うことはできません。 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。
dateformat=" <i>x</i> "	<p><i>x</i> 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 (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。 他の日または月エレメントとは 相互に排他的)</p> <p>デフォルト値の 1 が、指定されない各エレメントに割り当てられます。日付形式の例を以下に示します。</p> <p>"D-M-YYYY" "MM.DD.YYYY" "YYYYDDD"</p>
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>

インポート・ユーティリティ用のファイル・タイプ修飾子

表 23. 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) (続き)

修飾子	説明
timeformat="x"	<p><i>x</i> 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 23. 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、デフォルト値として 1 が割り当てられます。値が指定されていない MMM エレメントには、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>"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>

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 23. 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 文書进行处理するために役立ちます。</p> <p>各文書で、宣言タグが存在してエンコード属性が含まれる場合、そのエンコード方式は文字コード・ページと一致する必要があります。一致しない場合、その文書を含む行はリジェクトされます。文字コード・ページは codepage ファイル・タイプ修飾子で指定されている値であるか、または指定がない場合はアプリケーションのコード・ページであることに注意してください。デフォルトでは、文書は Unicode でエンコードされているか、またはエンコード属性のある宣言タグを含んでいます。</p>
xmlgraphic	<p>XML 文書が指定された GRAPHIC コード・ページでエンコードされていることを示します。</p> <p>このオプションは、特定の GRAPHIC コード・ページでエンコードされていてもエンコード宣言を含まない XML 文書进行处理するために役立ちます。</p> <p>各文書で、宣言タグが存在してエンコード属性が含まれる場合、そのエンコード方式は GRAPHIC コード・ページと一致する必要があります。一致しない場合、その文書を含む行はリジェクトされます。GRAPHIC コード・ページは codepage ファイル・タイプ修飾子で指定されている値のグラフィック・コンポーネントであるか、または指定がない場合はアプリケーションのコード・ページのグラフィック・コンポーネントであることに注意してください。デフォルトでは、文書は Unicode でエンコードされているか、またはエンコード属性のある宣言タグを含んでいます。</p> <p>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 24. 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	<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。³</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>
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	<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>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。</p> <pre>db2 import from myfile.asc of asc modified by striptblanks method 1 (1 10, 12 15) messages msgs.txt insert into staff</pre> <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 (後方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 (後方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>

インポート・ユーティリティ用のファイル・タイプ修飾子

表 25. インポート・ユーティリティで有効なファイル・タイプ修飾子: *DEL* (区切り文字で区切られている *ASCII* ファイル・フォーマット)

修飾子	説明
chardelx	<p><i>x</i> は単一文字のストリング区切り文字です。デフォルト値は二重引用符 (") です。指定した文字は、文字ストリングを囲むために、二重引用符の代わりに使用されます。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 charde1""</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、charde1' が指定されており、インポート・ユーティリティ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 charde1' method p (1, 4) insert into staff (id, years)"</pre>
coldelx	<p><i>x</i> は単一文字の列区切り文字です。デフォルト値はコンマ (,) です。指定した文字は、列の終わりを表すために、コンマの代わりに使用されます。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><i>x</i> は、小数点文字としてピリオドの代わりに使用される単一の置換文字です。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 charde1' 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 <i>DEL</i> 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 25. 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。
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 chardelx 、 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 26. 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。

インポート・ユーティリティ用のファイル・タイプ修飾子

表 27. `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	<p>ファイル内のすべてのデータは、コード・ページ <code>N</code> であると見なされます。</p> <p>警告: <code>N</code> が 1 バイト・コード・ページの場合、<code>GRAPHIC</code> データをデータベースにインポートすると、壊れます。</p>
なし	あり	<p>ファイル内の文字データは、アプリケーション・コード・ページであると見なされます。 <code>GRAPHIC</code> データは、アプリケーション <code>GRAPHIC</code> データのコード・ページであると見なされます。</p> <p>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。</p> <p>警告: アプリケーション・コード・ページが 1 バイトの場合、 <code>GRAPHIC</code> データは、データベースにたとえ <code>GRAPHIC</code> 列が収められていても、データベースにインポートされると壊れます。</p>
あり	あり	<p>文字データは、コード・ページ <code>N</code> であると見なされます。 <code>GRAPHIC</code> データは、<code>N</code> の <code>GRAPHIC</code> コード・ページであると見なされます。</p> <p><code>N</code> が 1 バイトまたは 2 バイト・コード・ページの場合は、すべてのデータは、コード・ページ <code>N</code> であると見なされます。</p> <p>警告: <code>N</code> が 1 バイト・コード・ページの場合、<code>GRAPHIC</code> データをデータベースにインポートすると、壊れます。</p>

注:

- サポートされていないファイル・タイプを `MODIFIED BY` オプションで使用しようとしても、インポート・ユーティリティは警告を出しません。この場合、インポート操作が失敗し、エラー・コードが戻されます。
- 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. この文字は、ソース・データのコード・ページで指定してください。

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
- dldeftype
- nodefaults
- usedefaults
- no_type_idfiletype
- 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。
8. すべての XML データは、メイン・データ・ファイルとは別の XML ファイル内に存在する必要があります。XML Data Specifier (XDS) (または NULL 値) が、メイン・データ・ファイル内の XML 列ごとに存在する必要があります。

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

- 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ければ、XML 文書は Unicode 形式であるか、またはエンコード属性のある宣言タグを含むと想定されます。
- 10. 整形式でない文書を含む行はリジェクトされます。
- 11. XMLVALIDATE オプションが指定されている場合、対応するスキーマに対して正常に検証された文書は、挿入される際にスキーマ情報がアノテーションとして付加されます。対応するスキーマに対する妥当性検査が失敗した文書を含む行は、リジェクトされます。妥当性検査を正常に行うためには、インポートを起動するユーザーの保持する特権に、次の 1 つ以上が含まれている必要があります。
 - SYSADM または DBADM の権限
 - 妥当性検査で使用する XML スキーマに対する USAGE 特権

関連資料:

- 902 ページの『データ移動での区切り文字の制限』
- 「管理 API リファレンス」の『db2Import API - 表、階層、ニックネーム、ビューへのデータのインポート』
- 541 ページの『IMPORT 』

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 28. エクスポート・ユーティリティーで有効なファイル・タイプ修飾子: すべてのファイル・フォーマット

修飾子	説明
lobsinfile	<p><i>lob-path</i> には、LOB データの入ったファイルへのパスを指定します。</p> <p>各パスには、データ・ファイル内で LOB ロケーション指定子 (LLS) によって示される 1 つ以上の LOB の入った、少なくとも 1 つのファイルが組み込まれます。LLS は、LOB ファイル・パスに保管されるファイル内の LOB のロケーションのストリング表現です。LLS の形式は、<i>filename.ext.nnn.mmm/</i> です。<i>filename.ext</i> は LOB を収めたファイルの名前、<i>nnn</i> はファイル内の LOB のオフセット (バイト単位)、<i>mmm</i> は LOB の長さ (バイト単位) を表します。例えば、ストリング <i>db2exp.001.123.456/</i> がデータ・ファイルに保管される場合、LOB はファイル <i>db2exp.001</i> のオフセット 123 に位置し、456 バイト長です。</p> <p>EXPORT の使用時に "lobsinfile" 修飾子を指定した場合、LOB データは LOBS TO 文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 文節は、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 <i>db2exp.001.7.-1/</i> です。</p>
xmlinsepfiles	それぞれの XQuery Data Model (QDM) 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。

表 28. エクスポート・ユーティリティーで有効なファイル・タイプ修飾子: すべてのファイル・フォーマット (続き)

修飾子	説明
lobsinsepfiles	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	QDM 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、QDM 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	QDM 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 <code>codepage</code> ファイル・タイプ修飾子で指定されている値であるか、または指定がない場合はアプリケーションのコード・ページであることに注意してください。デフォルトで、QDM インスタンスは Unicode で書き出されます。
xmlgraphic	xmlgraphic 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 <code>codepage</code> 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 29. 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
chardelex	<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 chardelex"</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 chardelex'</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>
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	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。
decptx	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

エクスポート・ユーティリティ用のファイル・タイプ修飾子

表 29. 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)

[illegible]

表 29. 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)</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>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>

エクスポート・ユーティリティ用のファイル・タイプ修飾子

表 30. エクスポート・ユーティリティで有効なファイル・タイプ修飾子: IXF ファイル・フォーマット

修飾子	説明
codepage= <i>x</i>	<p><i>x</i> 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 31. 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: 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.aaaaaa* の形式

エクスポート操作のために SELECT ステートメントで指定される日時列に組み込まれたデータも、これらの形式になります。

- タイム・スタンプ・フォーマットの場合、月の記述子と分の記述子のどちらも文字 M を使用するため、区別があいまいにならないように注意する必要があります。月のフィールドは、他の日付フィールドと隣接していなければなりません。分フィールドは、他の時刻フィールドに隣接していなければなりません。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" 文節または "XML TO" 文節のどちらも指定されていない場合でも、すべての QDM 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 ".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ければ、すべての QDM 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ければ、デフォルトで、すべての QDM 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・ユーティリティは、整形形式でない文書である QDM 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。

関連資料:

- 902 ページの『データ移動での区切り文字の制限』
- 「管理 API リファレンス」の『db2Export API - データベースからのデータのエクスポート』
- 478 ページの『EXPORT 』

データ移動での区切り文字の制限

区切り文字についての制約事項:

選択した区切り文字が、移動するデータの一部を成していないことを確かめるのは、ユーザーの責任です。区切り文字がデータの一部になっている場合、予期しないエラーが発生する場合があります。データを移動する際は、以下の制限が列、ストリング、DATALINK、および小数点区切り文字に適用されます。

- 区切り文字は相互に排他的である。
- 区切り文字としてバイナリー・ゼロ、改行文字、ブランク・スペースを使用することはできない。
- 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ない。
- 以下の文字は、ASCII ファミリー・コード・ページと EBCDIC ファミリー・コード・ページでは仕様が異なります。
 - シフトイン (0x0F) とシフトアウト (0x0E) 文字を、EBCDIC MBCS データ・ファイルの区切り文字として使用することはできない。

- MBCS、EUC、または DBCS コード・ページの区切り文字は、0x40 以下でなければならない (EBCDIC MBCS データのデフォルト小数点 0x4b は例外)。
- ASCII コード・ページまたは EBCDIC MBCS コード・ページでのデータ・ファイルのデフォルト区切り文字は、以下のとおりです。
 - " (0x22、二重引用符。ストリング区切り文字)
 - , (0x2c、コンマ; 列区切り文字)
- EBCDIC SBCS コード・ページにおけるデータ・ファイルのデフォルト区切り文字は、以下のとおりです。
 - " (0x7F、二重引用符。ストリング区切り文字)
 - , (0x6B、コンマ。列区切り文字)
- 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・ページがクライアントのコード・ページと異なっている場合は、非デフォルトの区切り文字を 16 進表示で指定するようお勧めします。例えば、以下のように指定します。

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

DEL ファイルでの二重区切り文字の認識サポートに関する以下の情報は、エクスポート、インポート、およびロード・ユーティリティーに適用されます。

- 区切り文字を、DEL ファイルの文字ベースのフィールド内で使用することができます。これは、タイプが CHAR、VARCHAR、LONG VARCHAR、または CLOB (lobsinfile が指定されている場合を除く) のフィールドに適用されます。区切り文字で囲まれている区切り文字の対は、データベースにインポートまたはロードされます。例えば、以下のように指定します。

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

これは、以下のようにインポートされます。

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

エクスポートの場合は、逆の規則が適用されます。例えば、以下のように指定します。

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

これは、以下のように DEL ファイルにエクスポートされます。

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

- DBCS 環境では、パイプ (|) 区切り文字はサポートされていません。

関連資料:

- 896 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』
- 884 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』
- 869 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』

付録 D. DB2 Database 技術情報

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
 - 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 CD 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
- サンプル・プログラム

IBM® は、定期的に資料の更新を入手できるようにしています。ibm.com® にある、DB2 インフォメーション・センターのオンライン版にアクセスする場合、そのバージョンは IBM によって最新の状態に保たれているので、資料の更新をインストールする必要はありません。DB2 インフォメーション・センターをインストール済みの場合は、資料の更新をインストールすることをお勧めします。資料の更新を使用すると、新しい情報が発表されたときに、DB2 インフォメーション・センター CD からインストール済みの情報、またはパスポート・アドバンテージからダウンロードした情報を更新することができます。

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 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、Redbooks™ などのその他の DB2 技術情報には、オンライン (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

ですが、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。

関連概念:

- ・ オンライン DB2 インフォメーション・センターの『DB2 インフォメーション・センターの機能』
- ・ サンプル・トピックの『サンプル・ファイル』

関連タスク:

- ・ 356 ページの『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』
- ・ 356 ページの『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』
- ・ 912 ページの『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』

関連資料:

- ・ 906 ページの『DB2 テクニカル・ライブラリー (PDF 形式)』

DB2 テクニカル・ライブラリー (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。

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。

こうした資料の情報は、すべての DB2 ユーザーに基本的なもので、プログラマーおよびデータベース管理者に加え、DB2 Connect または他の DB2 製品を使用するユーザーにとって役立つ内容です。

表 32. 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
管理ガイド： インプリメンテーション	SD88-6745-00	入手可能
管理ガイド： プランニング	SD88-6747-00	入手可能
管理 API リファレンス	SD88-6755-00	入手可能
管理 SQL ルーチンおよびビュー	SD88-6781-00	入手不可
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	SD88-6748-00	入手可能

表 32. 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
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	SD88-6749-00	入手可能
コマンド・リファレンス	SD88-6750-00	入手不可
データ移動ユーティリティ：ガイドおよびリファレンス	SD88-6751-00	入手可能
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	SD88-6752-00	入手可能
ADO.NET および OLE DB アプリケーションの開発	SD88-6754-00	入手可能
組み込み SQL アプリケーションの開発	SD88-6756-00	入手可能
SQL および 外部ルーチンの開発	SC88-4197-00	入手不可
Java (JDBC and SQLJ) アプリケーションの開発	SD88-6757-00	入手可能
Perl および PHP アプリケーションの開発	SD88-6758-00	入手不可
データベース・アプリケーション開発の基礎	SD88-6772-00	入手可能
DB2 インストールおよび管理 概説 (Linux および Windows 版)	GD88-6767-00	入手可能
メッセージ・リファレンス 第 1 巻	SD88-6760-00	入手不可
メッセージ・リファレンス 第 2 巻	SD88-6761-00	入手不可
マイグレーション・ガイド	GD88-6759-00	入手可能
Net Search Extender 管理およびプログラミング 注: この資料の HTML 版は、HTML ドキュメンテーション CD からインストールされません。	SH88-8592-00	入手可能
パフォーマンス・ガイド	SD88-6746-00	入手可能
Query Patroller 管理およびユーザーズ・ガイド	GD88-6763-00	入手可能
DB2 クライアント機能 概説 およびインストール	GD88-6764-00	入手不可
DB2 サーバー機能 概説およびインストール	GD88-6766-00	入手可能

表 32. 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
<i>Spatial Extender and Geodetic Data Management Feature</i> ユーザーズ・ガイドおよびリファレンス	SD88-6856-00	入手可能
<i>SQL</i> ガイド	SD88-6768-00	入手可能
<i>SQL</i> リファレンス 第 1 巻	SD88-6769-00	入手可能
<i>SQL</i> リファレンス 第 2 巻	SD88-6770-00	入手可能
システム・モニター ガイドおよびリファレンス	SD88-6771-00	入手可能
問題判別ガイド	GD88-6762-00	入手不可
<i>Visual Explain</i> チュートリアル	SC88-4162-00	入手不可
新機能	SD88-6773-00	入手可能
<i>XML Extender</i> 管理およびプログラミング	SD88-6857-00	入手可能
<i>XML</i> ガイド	SD88-6774-00	入手可能
<i>XQuery</i> リファレンス	SD88-6858-00	入手可能

表 33. 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
<i>DB2 Connect</i> ユーザーズ・ガイド	SD88-6753-00	入手可能
<i>DB2 Connect Personal Edition</i> 概説およびインストール	GC88-4131-00	入手可能
<i>DB2 Connect</i> サーバー機能 概説およびインストール	GD88-6765-00	入手可能

表 34. WebSphere® 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
<i>WebSphere Information Integration: フェデレーテッド・システム 管理ガイド</i>	SC88-4166-00	入手可能
<i>WebSphere Information Integration: レプリケーション およびイベント・パブリッシング ASNCLP プログラム・リファレンス</i>	SC88-4167-00	入手可能
<i>WebSphere Information Integration: フェデレーテッド・データ・ソース 構成ガイド</i>	SC88-4185-00	入手不可
<i>WebSphere Information Integration: SQL レプリケーション ガイドおよびリファレンス</i>	SC88-4168-00	入手可能

注: DB2 リリース情報は、ご使用の製品のリリースおよびフィックスパック・レベルに特有の追加情報を紹介します。詳しくは、関連リンクを参照してください。

関連概念:

- 905 ページの『DB2 技術情報の概説』
- 「リリース・ノート」の『リリース・ノートについて』

関連タスク:

- 909 ページの『DB2 の印刷資料の注文方法』

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 CD 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。たとえば、「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 CD 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 CD 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/db2help/>) で参照することができます。

印刷資料の注文方法:

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/jp/manuals/> の「ご注文について」をご覧ください。
- 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。
 - 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。
 - 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。
 - 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。

関連概念:

- 905 ページの『DB2 技術情報の概説』

関連資料:

- 906 ページの『DB2 テクニカル・ライブラリー (PDF 形式)』

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。

手順:

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。

たとえば、? 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。

関連タスク:

- 356 ページの『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』
- 356 ページの『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』

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

最新の DB2 製品情報の詳細については、
<http://publib.boulder.ibm.com/infocenter/db2help/> を参照してください。

DB2 バージョン 9.1 のトピックの場合、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 固有のトピックの場合は、バージョン 8 インフォメーション・センターの URL (<http://publib.boulder.ibm.com/infocenter/db2luw/v8/>) を参照してください。

関連タスク:

- 「管理ガイド: インプリメンテーション」の『DB2 コンテキスト・ヘルプと資料へのアクセスを設定する』

DB2 インフォメーション・センターにおける特定の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。

手順:

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、ページをリフレッシュし、使用する言語で DB2 インフォメーション・センターを表示します。

Firefox または Mozilla Web ブラウザーの場合に、使いたい言語でトピックを表示するには、以下のようにします。

1. 「ツール」 → 「オプション」 → 「言語」 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、ページをリフレッシュし、使用する言語で DB2 インフォメーション・センターを表示します。

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。

関連概念:

- 905 ページの『DB2 技術情報の概説』

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・センターがローカルにインストールされている場合、更新されたトピックをダウンロードして入手できます。ほとんどのトピックの最下部にある「最終更新」の値は、そのトピックの現在のレベルを示しています。

DB2 インフォメーション・センター全体に関する更新が入手可能かどうかを判別するには、インフォメーション・センターのホーム・ページの「最終更新」の値を調べてください。ローカルにインストールされたホーム・ページの値と、<http://www.ibm.com/software/data/db2/udb/support/icupdate.html> のダウンロード可能な更新の最終更新日の値を比較します。新しいダウンロード可能な更新がある場合、ローカルにインストールされているインフォメーション・センターを更新できます。

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、更新をダウンロードして適用できるようになります。
2. 「更新」機能を使用して、IBM から更新パッケージが入手可能かどうかを判別します。

注: 更新を CD で入手することもできます。CD から更新をインストールするようにインフォメーション・センターを構成する方法については、関連したリンクを参照してください。

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ダウンロードします。（「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・サービスを再開します。

手順:

コンピューターまたはイントラネット・サーバーにインストール済みの DB2 インフォメーション・センターを更新するには、以下のようになります。

1. DB2 インフォメーション・センター・サービスを停止します。
 - 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/db2icdv9 stop
```
2. インフォメーション・センターをスタンドアロン・モードで開始します。
 - Windows の場合:
 - a. コマンド・ウィンドウを開きます。

b. インフォメーション・センターがインストールされているパスにナビゲートします。デフォルトでは、DB2 インフォメーション・センターは C:\Program Files\IBM\DB2 Information Center\Version 9 ディレクトリにインストールされています。

c. 以下のように、DB2 インフォメーション・センターの完全修飾パスを使用して help_start.bat ファイルを実行します。

```
<DB2 Information Center dir>%doc%\bin\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 ディレクトリにインストールされています。

b. 以下のように、DB2 インフォメーション・センターの完全修飾パスを使用して help_start スクリプトを実行します。

```
<DB2 Information Center dir>/doc/bin/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. ダウンロード・プロセスを開始するには、ダウンロードする更新をチェックして選択し、「更新のインストール (Install Updates)」をクリックします。
5. ダウンロードおよびインストール・プロセスが完了したら、「完了」をクリックします。
6. 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 では、以下のように DB2 インフォメーション・センターの完全修飾パスを使用して help_end.bat ファイルを実行します。

```
<DB2 Information Center dir>%doc%\bin\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。Ctrl-C または他の方法を使用して、help_start.bat を終了しないでください。

• Linux では、以下のように DB2 インフォメーション・センターの完全修飾パスを使用して help_end スクリプトを実行します。

```
<DB2 Information Center dir>/doc/bin/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・サービスを再開します。

• Windows では、「スタート」→「コントロール パネル」→「管理ツール」→「サービス」をクリックします。次に、「DB2 インフォメーション・センター」サービスを右クリックして「開始」を選択します。

• Linux では、以下のコマンドを入力します。

```
/etc/init.d/db2icdv9 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。

関連概念:

- 「*DB2 Universal Database* サーバー機能 概説およびインストール」の『DB2 インフォメーション・センターのインストール・オプション』

関連タスク:

- 「*DB2 Universal Database* サーバー機能 概説およびインストール」の『インフォメーション・センター・デーモンの開始または停止 (Linux)』
- 「*DB2 Universal Database* サーバー機能 概説およびインストール」の『DB2 セットアップ・ウィザードを使用した DB2 インフォメーション・センターのインストール (Windows)』

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。

ネイティブ XML データ・ストア

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

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。

関連概念:

- 「管理ガイド: インプリメンテーション」の『Visual Explain の概要』

DB2 トラブルシューティング情報

DB2 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「サポートおよびトラブルシューティング」セ

クションにあります。ここには、DB2 診断ツールおよびユーティリティーを使用して、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 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/udb/support.html>) にアクセスしてください。

関連概念:

- 「問題判別ガイド」の『問題判別の概要』
- 905 ページの『DB2 技術情報の概説』

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 E. 特記事項

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
IBM World Trade Asia Corporation
Intellectual Property Law & Licensing

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で提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

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

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Canada Limited
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証できません。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

DB2 バージョン 9 ドキュメンテーション・ライブラリーの資料に記載されている会社名、製品名、またはサービス名は、IBM Corporation の商標である可能性があります。IBM Corporation の商標については、<http://www.ibm.com/legal/copytrade.shtml> を参照してください。

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。

Intel[®]、Itanium[®]、Pentium[®]、および Xeon[®] は、Intel Corporation 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または登録商標です。

UNIX[®] は、The Open Group の米国およびその他の国における登録商標です。

Linux は、Linus Torvalds 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。

付録 F. IBM と連絡をとる

ご自分の国または地域の IBM と連絡を取るには、<http://www.ibm.com/planetwide> で IBM の「Directory of Worldwide Contacts」を調べてください。

DB2 製品について理解を深めるには、<http://www.ibm.com/software/data/db2/> をご覧ください。

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 769
アドバイザー
設計アドバイザー 23
db2advis 23
アンカタログ
システム・データベース・ディレクトリー 816
データベース項目 816
ホスト DCS データベース項目 818
暗黙接続 350
異常終了
再始動コマンド 736
一時ファイル
LOAD コマンド 608
移動、データの
区切り文字についての制約事項 902
データベース間の 541
イベント・アナライザー・コマンド 96
イベント・モニター生産性向上ツール・コマンド 98
イベント・モニターのターゲット表定義の生成コマンド 100
印刷資料
配列 909
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 195
インスタンス作成インターフェースの開始コマンド 141
インスタンスの移行コマンド 135
インスタンスの更新コマンド 143
インスタンスの作成コマンド 129
インスタンスの自動始動コマンド 125
インスタンスの除去コマンド 132
インスタンスのリスト・コマンド 134
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 192
インストール済み DB2 製品の更新コマンド 330
インストール済みの DB2 製品およびフィチャーのリスト・コマンド 167

インフォメーション・センター
更新 912
バージョン 910
別の言語での表示 911
インポート
データ 541
インポート・ユーティリティー
ファイル・タイプ修飾子 884
ウィザード
設計アドバイザー 23
db2advis 23
エクスポート
データベース表ファイル 478
エクスポート・ユーティリティー
ファイル・タイプ修飾子 896
エラー・メッセージ
データベース構成ファイル 502
無効なチェックサム
データベース構成ファイル 730, 844
データベース・マネージャー構成ファイル 726
リモート・データベースのドロップ 469
応答ファイル
生成プログラム
db2rspgn 261
大文字小文字の区別
コマンド 350
命名規則における 867

【カ行】

カーソル固定 (CS)
変更 432
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 197
開始
DB2
db2start コマンド 292
カタログ作成
データベース 408
ホスト・データベース 412
環境変数
DB2OPTIONS 340
監査機能管理者ツール・コマンド 29
行継続文字
コマンド行プロセッサー (CLP) 350
区切り文字についての制約事項
移動、データの 902

継続文字
コマンド行プロセッサー (CLP) 350
現行 DAS レベルの表示コマンド 75
検索パスの変更コマンド 58
検査結果のフォーマット・コマンド 140
コード・ページ
EXPORT コマンド 478
IMPORT コマンド 541
コード・ページ・ファイル・タイプ修飾子 608
更新
インフォメーション・センター 912
DB2 インフォメーション・センター 912
構成
管理
リセット、デフォルトに 726
例 486
データベース
更新 844
リセット、デフォルトに 730
例 502
データベース・マネージャー、例 508
CLI、例 496
構文
コマンド行プロセッサー SQL ステートメント用 857
説明 865
コマンド
出力のリダイレクト 350
セットアップ 332
ヘルプの呼び出し 356
ACTIVATE DATABASE 363
ADD CONTACT 365
ADD CONTACTGROUP 367
ADD DBPARTITIONNUM 368
ARCHIVE LOG 373
ATTACH 376
AUTOCONFIGURE 378
BACKUP DATABASE 382
BIND 389
CATALOG DATABASE 408
CATALOG DCS DATABASE 412
CATALOG LDAP DATABASE 415
CATALOG LDAP NODE 419
CATALOG LOCAL NODE 421
CATALOG NAMED PIPE NODE 423
CATALOG ODBC DATA SOURCE 425
CATALOG TCP/IP NODE 426

コマンド (続き)

CHANGE DATABASE
 COMMENT 430
 CHANGE ISOLATION LEVEL 432
 CREATE DATABASE 435
 CREATE TOOLS CATALOG 451
 dasauto 3
 dasprt 4
 dasdrop 5
 dasmigr 6
 dasupdt 8
 db2 339
 DB2 JDBC パッケージ・バインド・プログラム・ユーティリティー 147
 DB2 SQLJ プロファイル・バインド・プログラム 272
 DB2 SQLJ 変換プログラム 334
 db2admin 13
 db2adutl 15
 db2advis 23
 db2audit 29
 db2batch 35
 db2bfd 45
 db2cap 47
 db2cc 51
 db2cfexp 53
 db2cfimp 55
 db2chgpath 58
 db2ckbkb 59
 db2ckmig 63
 db2ckrst 65
 db2cli 67
 db2cmd 68
 db2dart 70
 db2daslevel 75
 db2dclgn 76
 db2diag 79
 db2drdat 91
 db2empfa 95
 db2eva 96
 db2evmon 98
 db2evtbl 100
 db2exmig 104
 db2expln 105
 db2extsec 111
 db2flsn 112
 db2fm 114
 db2fs 116
 db2gcf 117
 db2gov 120
 db2govlg 122
 db2gpmap 123
 db2hc 124
 db2iauto 125
 db2iclus 126
 db2icrt 129

コマンド (続き)

db2idrop 132
 db2ilist 134
 db2imigr 135
 db2inidb 137
 db2inspf 140
 db2isetup 141
 db2iupdt 143
 db2jdbcbind 147
 db2ldcfg 149
 db2level 151
 db2licm 152
 db2listvolumes 154
 db2logsforrwd 155
 db2look 156
 db2ls 167
 db2move 169
 db2mqsln 178
 db2mscs 182
 db2mtrk 186
 db2nchg 190
 db2ncrt 192
 db2ndrop 195
 db2osconf 197
 db2pd 201
 db2pdcfg 243
 db2perfc 246
 db2perfi 248
 db2perfr 249
 db2rbind 250
 db2relocatedb 255
 db2sampl 262
 db2set 265
 db2setup 268
 db2sql92 269
 db2sqljbind 272
 db2sqljcustomize 279
 db2sqljprint 291
 db2start 292
 db2stop 293
 db2support 294
 db2swtch 300
 db2sync 301
 db2systray 302
 db2tapemgr 305
 db2tbst 308
 db2trc 309
 db2uiddl 314
 db2undgp 316
 db2unins 317
 db2untag 319
 db2xpvt 320
 db2_deinstall 10
 db2_install 11
 db2_recon_aid 252
 DEACTIVATE DATABASE 454

コマンド (続き)

DECOMPOSE XML DOCUMENT 456
 DEREGISTER 458
 DESCRIBE 460
 DETACH 466
 disable_MQFunctions 325
 doce_deinstall 321
 doce_install 323
 DROP CONTACT 467
 DROP CONTACTGROUP 468
 DROP DATABASE 469
 DROP DBPARTITIONNUM
 VERIFY 471
 DROP TOOLS CATALOG 473
 ECHO 475
 EDIT 476
 enable_MQFunctions 327
 EXPORT 478
 FORCE APPLICATION 484
 GET ADMIN CONFIGURATION 486
 GET ALERT CONFIGURATION 488
 GET AUTHORIZATIONS 494
 GET CLI CONFIGURATION 496
 GET CONNECTION STATE 498
 GET CONTACTGROUP 499
 GET CONTACTGROUPS 500
 GET CONTACTS 501
 GET DATABASE
 CONFIGURATION 502
 GET DATABASE MANAGER
 CONFIGURATION 508
 GET DATABASE MANAGER
 MONITOR SWITCHES 513
 GET DESCRIPTION FOR HEALTH
 INDICATOR 516
 GET HEALTH NOTIFICATION
 CONTACT LIST 518
 GET HEALTH SNAPSHOT 519
 GET INSTANCE 522
 GET MONITOR SWITCHES 523
 GET RECOMMENDATIONS 526
 GET ROUTINE 530
 GET SNAPSHOT 532
 HELP 538
 HISTORY 540
 IMPORT 541
 INITIALIZE TAPE 560
 INSPECT 562
 installFixPack 330
 LIST ACTIVE DATABASES 568
 LIST APPLICATIONS 570
 LIST COMMAND OPTIONS 572
 LIST DATABASE DIRECTORY 573
 LIST DATABASE PARTITION
 GROUPS 577
 LIST DBPARTITIONNUMS 579

コマンド (続き)

LIST DCS APPLICATIONS 580
LIST DCS DIRECTORY 582
LIST DRDA INDOUBT
TRANSACTIONS 584
LIST HISTORY 586
LIST INDOUBT
TRANSACTIONS 589
LIST NODE DIRECTORY 592
LIST ODBC DATA SOURCES 595
LIST PACKAGES/TABLES 596
LIST TABLESPACE
CONTAINERS 599
LIST TABLESPACES 601
LIST UTILITIES 606
LOAD 608
LOAD QUERY 630
Microsoft Cluster Server 126
MIGRATE DATABASE 634
MQ Listener 178
PING 636
PRECOMPILE 638
PRUNE HISTORY/LOGFILE 665
PUT ROUTINE 667
QUERY CLIENT 669
QUIESCE 670
QUIESCE TABLESPACES FOR
TABLE 673
QUIT 676
REBIND 677
RECONCILE 681
RECOVER DATABASE 685
REDISTRIBUTE DATABASE
PARTITION GROUP 692
REFRESH LDAP 696
REGISTER 697
REORG INDEXES/TABLE 705
REORGCHK 716
RESET ADMIN
CONFIGURATION 726
RESET ALERT
CONFIGURATION 728
RESET DATABASE
CONFIGURATION 730
RESET DATABASE MANAGER
CONFIGURATION 732
RESET MONITOR 734
RESTART DATABASE 736
RESTORE DATABASE 738
REWIND TAPE 755
ROLLFORWARD DATABASE 756
RUNCMD 768
RUNSTATS 769
SET CLIENT 784
SET RUNTIME DEGREE 787

コマンド (続き)

SET TABLESPACE
CONTAINERS 789
SET TAPE POSITION 791
SET UTIL_IMPACT_PRIORITY 792
SET WRITE 794
sqlj 334
START DATABASE MANAGER 796
START HADR 803
STOP DATABASE MANAGER 806
STOP HADR 810
TAKEOVER HADR 812
TERMINATE 815
UNCATALOG DATABASE 816
UNCATALOG DCS DATABASE 818
UNCATALOG LDAP
DATABASE 820
UNCATALOG LDAP NODE 822
UNCATALOG NODE 823
UNCATALOG ODBC DATA
SOURCE 824
UNQUIESCE 825
UPDATE ADMIN
CONFIGURATION 827
UPDATE ALERT
CONFIGURATION 829
UPDATE ALTERNATE SERVER FOR
DATABASE 834
UPDATE ALTERNATE SERVER FOR
LDAP DATABASE 836
UPDATE CLI CONFIGURATION 838
UPDATE COMMAND OPTIONS 840
UPDATE CONTACT 842
UPDATE CONTACTGROUP 843
UPDATE DATABASE
CONFIGURATION 844
UPDATE DATABASE MANAGER
CONFIGURATION 847
UPDATE HEALTH NOTIFICATION
CONTACT LIST 849
UPDATE HISTORY FILE 850
UPDATE LDAP NODE 852
UPDATE MONITOR SWITCHES 854

コマンド行プロセッサ (CLP)

行継続文字 350
コマンド行 339
シェル・コマンド 339
終了 339, 676, 815
使用 350
説明 339
対話式入力モード 339
データベースへのアクセス 339
バッチ・モード 339
ヘルプへのアクセス 339
戻りコード 349
呼び出し 339

コマンド行プロセッサ (CLP) (続き)

options 340
SQL ステートメント 857
コマンド行プロセッサ呼び出しコマンド
339
コマンド構文
CLP コマンド 339
コミットなし (NC) 432
コンテナ・タグの解放コマンド 319
コントロール・センター
開始 51
コントロール・センターの開始コマンド
51

[サ行]

最適化

REORG INDEXES/TABLE コマンド
705

索引

統計

RUNSTATS コマンド 769

REORGCHK コマンド 716

サンプル・データベースの作成コマンド
262

磁気テープ上のログ・ファイルの管理コマ
ンド 305

磁気テープ・バックアップ 382

システム・コマンド

概説 1

システム・データベース・ディレクトリー
アンカタログ 816

実行特権の取り消しコマンド 316

修飾子

ファイル・タイプ

インポート・ユーティリティ
884

エクスポート・ユーティリティ
896

ロード・ユーティリティ 869

EXPORT コマンド 478

IMPORT コマンド 541

LOAD コマンド 608

終了

異常 736

コマンド行プロセッサのバックエン
ド処理 815

normal 806

終了コード (CLP)

リスト 349

使用条件

資料の使用 915

スキーマ

新規データベースの 435

ストレージ

物理的 705

すべてのディスク・ボリュームの GUID
の表示コマンド 154
すべてのパッケージの再バインド・コマン
ド 250
静止、ファントム 673
設計アドバイザー 23, 526
接続構成インポート・ツール・コマンド
55
接続構成エクスポート・ツール・コマンド
53
切断
 コマンド行プロセッサ・フロントエン
 ド・プロセスおよびバックエン
 ド・プロセス 815
宣言生成プログラム・コマンド 76
ゾーン 10 値ファイル・タイプ修飾子
608
増分リストア・イメージ順序の検査コマン
ド 65

[タ行]

ダンプ、トレースをファイルへ 309
チュートリアル
 トラブルシューティングと問題判別
 914
 Visual Explain 914
データ
 フラグメント化、消去、表の再編成に
 よる 705
データのエクスポート
 ファイル・タイプ修飾子 478
データベース
 移行 634
 インポート、ファイルを表へ 541
 カタログ作成 408
 間接 (間接) ディレクトリー項目 573
 検査、許可の 494
 項目の除去 (アンカatalog) 816
 再始動 736
 再編成 716
 削除、ログ・ファイルによるリカバリ
 ーの確認 469
 情報 532
 除去、ホスト DCS 項目の 818
 統計 769
 ドロップ 469
 バックアップ履歴ファイル 665
 ファイルへの表のエクスポート 478
 変更、ディレクトリー内の注釈 430
 ホーム (home) ディレクトリー項目
 573
 モニター
 リセット 734
 リカバリー 756
 リストア (再構築) 738

データベース (続き)
 リモート (remote) ディレクトリー項目
 573
 ロード、ファイルを表へ 608
 ロールフォワード・リカバリー 756
データベース移動ツール・コマンド 169
データベース構成
 更新 844
 ネットワーク・パラメーター値 844
 リセット、デフォルトに 730
 例 502
データベース事前移行ツール・コマンド
63
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 818
データベースの再配置コマンド 255
データベース分析およびレポート・ツ
ール・コマンド 70
データベース・システム・モニター
 GET DATABASE MANAGER
 MONITOR SWITCHES コマンド
 513
 GET MONITOR SWITCHES コマンド
 523
 GET SNAPSHOT 532
 RESET MONITOR コマンド 734
 UPDATE MONITOR SWITCHES コマ
 ンド 854
データベース・ディレクトリー
 サンプル内容 573
 説明 573
 変更、コメントの 430
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 190
データベース・パフォーマンス値のリセッ
ト・コマンド 246
データベース・マネージャー
 インスタンス 522
 開始 796
 コマンド・プロンプトからのアクセス
 1
 システム・コマンド 1
 停止 806
 統計 532
 モニター・スイッチ 513, 523
データベース・マネージャー構成
 サンプル・ファイル 508
 GET DATABASE MANAGER
 CONFIGURATION コマンド 508
データベース・モニター
 説明 854
データ保全性
 維持、分離レベルを使用した 432

データ・スキュー
 データベース・パーティション・グル
 ープのデータの再分散 692
停止
 DB2
 db2stop コマンド 293
ディレクトリー
 アンカatalog 816
 項目の削除 823
 システム・データベース、除去 816
 データベース
 変更、コメントの 430
 データベース接続サービス (DCS)、項
 目のアンカatalog 818
 node
 項目の除去 823
デフォルト DB2 コピーの切り替えコマン
ド 300
デフォルトの構成
 データベース、リセット 730
 ADMIN、リセット 726
統計
 再編成、索引の 716
 データベース 769
 データベース・マネージャー 532
 REORGCHK 716
ドキュメンテーション 905, 906
 使用条件 915
特殊文字
 コマンド内 350
特記事項 917
特権
 間接 494
 直接 494
 データベース
 作成時に付与 435
 報告 494
トラップ・ファイルのフォーマット・コマ
ンド 320
トラブルシューティング
 オンライン情報 914
 チュートリアル 914
トレース
 アクティブ化 309
トレース・コマンド 309

[ナ行]

ノード
 SOCKS 426
ノード・ディレクトリー、項目の除去
823

【ハ行】

バイナリー・ファイル 309
バインド
 暗黙的に作成されるスキーマ 389,
 638
 エラー 435
バインド・ファイル記述ツール・コマンド 45
パスワード
 ATTACH コマンドを使って変更 376
 CONNECT を使用して変更 857
バックアップの検査コマンド 59
バックアップ・サービス API
 (XBSA) 382
パッケージ
 再作成 677
パフォーマンス
 調整
 表の再編成による 705
 REORGCHK コマンド 716
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 248
パフォーマンス・モニター登録ツール・コ
 マンド 249
反復可能読み取り (RR)
 変更 432
非コミット読み取り (UR)
 変更 432
表
 再編成
 必要性の判別 716
 REORG INDEXES/TABLE コマン
 ド 705
 統計
 説明 769
 ファイルにエクスポートする 478
 ファイルのロード 608
 ファイルをインポートする 541
表スペース状態の獲得コマンド 308
ファースト・ステップ 116
ファイル・タイプ修飾子
 インポート・ユーティリティー 884
 エクスポート・ユーティリティー
 478, 896
 ロード・ユーティリティー 869
 IMPORT コマンド 541
 LOAD コマンド 608
ファイル・フォーマット
 インポート、ファイルを表へ 541
 ファイルへの表のエクスポート 478
ファントム静止 673
フェデレーテッド・プリコンパイル/BIND
 オプション 389, 638
複数の表の RECONCILE コマンド 252

複数ページ・ファイル割り振りの使用可能
 化コマンド 95
分散マップの取得コマンド 123
分離レベル
 CHANGE ISOLATION LEVEL コマン
 ド 432
ヘルス・センターの開始コマンド 124
ベンチマーク・ツール・コマンド 35
ホスト・システム
 除去、DCS カタログ項目の 818
 データベースのカatalog作成 412
DB2 Connect がサポートする接続
 412

【マ行】

未確定トランザクション・フィールド
 589
ミラーリングされたデータベースの初期化
 コマンド 137
命名規則
 データベース・マネージャー・オブジ
 ェクト 867
メッセージ
 ヘルプへのアクセス 339
メッセージ・ヘルプ
 呼び出し 356
メモリー・トラッカー・コマンド 186
戻りコード
 コマンド行プロセッサ (CLP) 349
モニター
 データベース 513, 523
問題判別
 オンライン情報 914
 チュートリアル 914
問題判別動作の DB2 データベースの構
 成コマンド 243
問題分析および環境収集ツール・コマンド
 294

【ヤ行】

ユーザー ID
 許可 494
呼び出し 356
 コマンド・ヘルプ 356
読み取り固定 (RS)
 変更 432

【ラ行】

ライセンス管理ツール・コマンド 152
リカバリー
 データベース 738
 ロールフォワードなし 738

リカバリー (続き)
 ロールフォワードを用いた 756
リストア
 旧バージョンの DB2 データベース
 738
ロード、データの
 ファイルをデータベース表へ 608
 ファイル・タイプ修飾子 608
ロード・ユーティリティー
 一時ファイル 608
 ファイル・タイプ修飾子 869
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 155
ログ
 ロールフォワード中のリスト 756
ログ・シーケンス番号の検出コマンド
 112
ロック
 リセット、最大値をデフォルトに 730

【ワ行】

ワークステーション
 リモート
 アンカatalog、ローカル・ワークス
 テーションから 823
 除去、データベースのカatalog項目
 の 816
 データベースのカatalog作成 408

A

action プリコンパイル/BIND オプション
 389, 638
ACTIVATE DATABASE コマンド 363
ADD CONTACT コマンド 365
ADD CONTACTGROUP コマンド 367
ADD DBPARTITIONNUM コマンド 368
ADD XMLSCHEMA DOCUMENT コマン
 ド
 構文 371
ADMIN 構成
 ネットワーク・パラメーター値 827
 リセット、デフォルトに 726
 例 486
 file 486
Administration Server
 構成 486
 作成 13
 ドロップ 13
anyorder ファイル・タイプ修飾子 608
APPC (詳細プログラム間通信)
 node
 アンカatalog 823
ARCHIVE LOG コマンド 373

ASC インポート・ファイル・タイプ 541
ATTACH コマンド 376
AUTOCONFIGURE コマンド 378

B

BACKUP DATABASE コマンド 382
binarynumerics ファイル・タイプ修飾子 608
BIND コマンド
構文 389
bindfile プリコンパイル・オプション 638
blocking プリコンパイル/ BIND オプション 389, 638

C

CALL ステートメント
CLP からの実行 857
CATALOG DATABASE コマンド
構文 408
CATALOG DCS DATABASE コマンド 412
CATALOG LDAP DATABASE コマンド 415
CATALOG LDAP NODE コマンド 419
CATALOG LOCAL NODE コマンド 421
CATALOG NAMED PIPE NODE コマンド 423
CATALOG ODBC DATA SOURCE コマンド 425
CATALOG TCP/IP NODE コマンド 426
CCSIDG プリコンパイル/ BIND オプション 389, 638
CCSIDM プリコンパイル/ BIND オプション 389, 638
CCSIDS プリコンパイル/ BIND オプション 389, 638
CHANGE DATABASE COMMENT コマンド 430
CHANGE ISOLATION LEVEL コマンド 432
chardel ファイル・タイプ修飾子
インポート 541
エクスポート 478
ロード 608
charsub プリコンパイル/ BIND オプション 389, 638
CLI (コール・レベル・インターフェース)
構成 496
CLIPKG プリコンパイル/ BIND オプション 389
CLI/ODBC 静的パッケージ・バインディング・ツール・コマンド 47

CLOSE ステートメント
CLP からの実行 857
CLP (コマンド行プロセッサ)
コマンド構文 339
終了 676, 815
cnulreqd プリコンパイル/ BIND オプション 389, 638
coldel ファイル・タイプ修飾子
インポート 541
エクスポート 478
ロード 608
collection プリコンパイル/ BIND オプション 389, 638
command
db2drvmp 93
COMPLETE XMLSCHEMA コマンド
構文 434
compound ファイル・タイプ修飾子 541
CONNECT ステートメント
CLP からの実行 857
connect プリコンパイル・オプション 638
CREATE DATABASE コマンド
説明 435
CREATE DATABASE ステートメントの
RESTRICTIVE 文節 435
CREATE TOOLS CATALOG コマンド 451

D

DAS (DB2 Administration Server)
構成 486
作成 13
ドロップ 13
DAS の更新コマンド 8
DAS の自動始動コマンド 3
dasauto コマンド 3
dasprt コマンド 4
dasdrop コマンド 5
dasmigr コマンド 6
dasupdt コマンド 8
dateformat ファイル・タイプ修飾子 541, 608
datesiso ファイル・タイプ修飾子 478, 541, 608
DATETIME プリコンパイル/ BIND オプション 389, 638
db2
CMD の説明 339
DB2 Administration Server (DAS)
作成 13
ドロップ 13
DB2 Administration Server コマンド 13
作成 4
DB2 Administration Server の移行コマンド 6
DB2 Administration Server の除去コマンド 5
DB2 Connect
サポートされる接続 412
DB2 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 147
DB2 SQLJ プロファイル・カスタマイザー・コマンド 279
DB2 SQLJ プロファイル・バインド・プログラム・コマンド 272
DB2 SQLJ プロファイル・プリンター・コマンド 291
DB2 SQLJ 変換プログラム・コマンド 334
DB2 インスタンスの制御コマンド 117
DB2 インフォメーション・センター
更新 912
バージョン 910
別の言語での表示 911
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 321
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 323
DB2 オブジェクトの許可の設定コマンド 111
DB2 管理プログラム・コマンド 120
DB2 管理プログラム・ログ照会コマンド 122
db2 コマンド 339
DB2 コマンド・ウィンドウのオープン・コマンド 68
DB2 サービス・レベルの表示コマンド 151
DB2 索引アドバイザー 23
DB2 システム・トレイの開始コマンド 302
DB2 障害モニター・コマンド 114
DB2 資料の注文 909
DB2 シンクロナイザーの開始コマンド 301
DB2 製品インストール・コマンド 11
DB2 製品のアンインストール・コマンド 317
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 10
DB2 対話機能 CLI コマンド 67
DB2 データベースのモニターおよびトラブルシューティング・コマンド 201
DB2 データベース・ドライブのマップ・コマンド 93
DB2 統計および DDL 抽出ツール・コマンド 156

DB2 のインストール・コマンド 268, 332
DB2 の開始コマンド 292
DB2 の停止コマンド 293
DB2 プロファイル・レジストリー・コマンド 265
db2admin コマンド 13
db2adutl コマンド 15
db2advis 23, 526
db2audit コマンド 29
db2batch コマンド 35
db2bfd コマンド 45
db2cap コマンド 47
db2cc コマンド 51
db2cfexp コマンド 53
db2cfimp コマンド 55
db2chgpath コマンド 58
db2ckbkb コマンド 59
db2ckmig コマンド 63
db2ckrst コマンド 65
db2cli コマンド 67
db2cmd コマンド 68
db2dart コマンド 70
db2daslevel コマンド 75
db2dclgn 宣言生成プログラム
構文 76
db2diag コマンド 79
db2diag.log 分析ツール・コマンド 79
db2drdat コマンド 91
db2drvmp コマンド 93
db2empfa コマンド 95
db2eva コマンド 96
db2evmon コマンド 98
db2evtbl コマンド 100
db2exfmt ツール 102
db2exmig コマンド 104
db2expln コマンド 105
db2extsec コマンド 111
db2flsn コマンド 112
db2fm コマンド 114
db2fs コマンド 116
db2gcf コマンド 117
db2gov コマンド 120
db2govlg コマンド 122
db2gpmap コマンド 123
db2hc コマンド 124
db2iauto コマンド 125
db2iclus コマンド 126
db2icrt コマンド 129
db2idrop コマンド 132
db2ilist コマンド 134
db2imigr コマンド 135
db2inidb コマンド 137
db2inspf コマンド 140
db2isetup コマンド 141
db2iupdt コマンド 143

db2jdbcbind コマンド 147
db2ldcfg コマンド 149
db2level コマンド 151
db2licm コマンド 152
db2listvolumes コマンド 154
db2logsforrwd コマンド 155
db2look コマンド 156
db2ls コマンド 167
db2move コマンド 169
db2mqdsn コマンド 178
db2mscs コマンド 182
db2mtrk コマンド 186
db2nchg コマンド 190
db2ncrt コマンド 192
db2ndrop コマンド 195
DB2OPTIONS 340
db2osconf コマンド 197
db2pd コマンド 201
db2pdcfg コマンド 243
db2perfc コマンド 246
db2perfi コマンド 248
db2perfr コマンド 249
db2rbind コマンド 250
db2relocatedb コマンド 255
db2rfpen コマンド 260
db2rspgn 応答ファイル・ジェネレーター
261
db2sampl コマンド 262
db2set コマンド 265
db2setup コマンド 268
db2sql92 コマンド 269
db2sqljbind コマンド 272
db2sqljcustomize コマンド 279
db2sqljprint コマンド 291
db2start コマンド 292, 796
db2stop コマンド 293, 806
db2support コマンド 294
db2swtch コマンド 300
db2sync コマンド 301
db2sysstray コマンド 302
db2tapemgr コマンド 305
db2tbst コマンド 308
db2trc コマンド 309
db2uiddl コマンド 314
db2undgp コマンド 316
db2unins コマンド 317
db2untag コマンド 319
db2xpvt コマンド 320
db2_deinstall コマンド 10
db2_install コマンド 11
db2_recon_aid コマンド 252
DCLGEN コマンド
db2dclgn 宣言生成プログラム 76
DEACTIVATE DATABASE コマンド
454

dec プリコンパイル/BIND オプション
389, 638
decctl プリコンパイル/BIND オプション
389, 638
DECLARE CURSOR ステートメント
CLP からの実行 857
DECOMPOSE XML DOCUMENT コマ
ンド
説明 456
decplusblank ファイル・タイプ修飾子
478, 541, 608
decpt ファイル・タイプ修飾子 478, 541,
608
deferred_prepare プリコンパイル・オプシ
ョン 638
degree プリコンパイル/ BIND オプション
389, 638
delprioritychar ファイル・タイプ修飾子
541, 608
DEREGISTER コマンド 458
DESCRIBE コマンド 460
DETACH コマンド 466
disable_MQFunctions コマンド 325
disconnect プリコンパイル・オプション
638
dlldel ファイル・タイプ修飾子 478, 541,
608
doce_deinstall コマンド 321
doce_install コマンド 323
DRDA トレース・コマンド 91
DROP CONTACT コマンド 467
DROP CONTACTGROUP コマンド 468
DROP DATABASE コマンド
構文 469
DROP DBPARTITIONNUM VERIFY コマ
ンド 471
DROP TOOLS CATALOG コマンド 473
dumpfile ファイル・タイプ修飾子 608
DYNAMICRULES プリコンパイル/ BIND
オプション
BIND コマンド 389
PRECOMPILE コマンド 638

E

ECHO コマンド 475
EDIT コマンド 476
enable_MQFunctions コマンド 327
EXPLAIN BIND オプション 389, 638
Explain 表
データのフォーマット・ツール 102
Explain 表のマイグレーション・コマンド
104
explsnap プリコンパイル/BIND オプシ
ョン 389, 638
EXPORT コマンド 478

F

fastparse ファイル・タイプ修飾子 608
FETCH ステートメント
CLP からの実行 857
FORCE APPLICATION コマンド 484
forcein ファイル・タイプ修飾子 541,
608
funcpath プリコンパイル/BIND オプショ
ン 389, 638

G

generatedignore ファイル・タイプ修飾子
541, 608
generatedmissing ファイル・タイプ修飾子
541, 608
generatedoverride ファイル・タイプ修飾子
608
generic プリコンパイル/BIND オプション
389, 638
GET ADMIN CONFIGURATION コマン
ド 486
GET ALERT CONFIGURATION コマンド
488
GET AUTHORIZATIONS コマンド 494
GET CLI CONFIGURATION コマンド
496
GET CONNECTION STATE コマンド
498
GET CONTACTGROUP コマンド 499
GET CONTACTGROUPS コマンド 500
GET CONTACTS コマンド 501
GET DATABASE CONFIGURATION コマ
ンド 502
GET DATABASE MANAGER
CONFIGURATION コマンド 508
GET DATABASE MANAGER MONITOR
SWITCHES コマンド 513
GET DESCRIPTION FOR HEALTH
INDICATOR コマンド 516
GET HEALTH NOTIFICATION
CONTACT LIST コマンド 518
GET HEALTH SNAPSHOT コマンド
519
GET INSTANCE コマンド 522
GET MONITOR SWITCHES コマンド
523
GET RECOMMENDATIONS
command 526
GET ROUTINE コマンド 530
GET SNAPSHOT コマンド 532
UPDATE MONITOR SWITCHES への
影響 854
grant BIND オプション 389
grantgroup BIND オプション 389

grantuser BIND オプション 389

H

help
コマンドの 356
表示 911
メッセージの 356
SQL ステートメントの 910
HELP コマンド 538
HISTORY コマンド 540

I

IBM と連絡をとる 921
identityignore 541
identityignore ファイル・タイプ修飾子
608
identitymissing ファイル・タイプ修飾子
541, 608
identityoverride ファイル・タイプ修飾子
608
implieddecimal ファイル・タイプ修飾子
541, 608
IMPORT コマンド 541
indexfreespace ファイル・タイプ修飾子
608
indexixf ファイル・タイプ修飾子 541
indexschema ファイル・タイプ修飾子
541
INITIALIZE TAPE コマンド 560
insert プリコンパイル/BIND オプション
389, 638
INSPECT コマンド 562
installFixPack コマンド 330
IPX/SPX ノード
アンカタログ 823
isolation プリコンパイル / BIND オプシ
ョン 389, 638

J

JDBC パッケージ・バインド・プログラ
ム・ユーティリティー・コマンド 147

K

keepblanks ファイル・タイプ修飾子 541,
608

L

LANGLEVEL プリコンパイル・オプショ
ン 638

LANGLEVEL プリコンパイル・オプショ
ン (続き)

SQL92E 638

LDAP 環境の構成コマンド 149
level プリコンパイル・オプション 638
LIST ACTIVE DATABASES コマンド
568
LIST APPLICATIONS コマンド 570
LIST COMMAND OPTIONS コマンド
572
LIST DATABASE DIRECTORY コマンド
573
LIST DATABASE PARTITION GROUPS
コマンド 577
LIST DBPARTITIONNUMS コマンド
579
LIST DCS APPLICATIONS コマンド
580
LIST DCS DIRECTORY コマンド 582
LIST DRDA INDOUBT TRANSACTIONS
コマンド 584
LIST HISTORY コマンド 586
LIST INDOUBT TRANSACTIONS コマン
ド 589
LIST NODE DIRECTORY コマンド 592
LIST ODBC DATA SOURCES コマンド
595
LIST PACKAGES コマンド 596
LIST PACKAGES/TABLES コマンド 596
LIST TABLES コマンド 596
LIST TABLESPACE CONTAINERS コマ
ンド 599
LIST TABLESPACES コマンド 601
LIST UTILITIES コマンド 606
LOAD QUERY コマンド 630
LOAD コマンド 608
lobsinfile ファイル・タイプ修飾子 478,
541, 608
longerror プリコンパイル・オプション
638

M

messages プリコンパイル / BIND オプシ
ョン 389, 638
Microsoft Cluster Server コマンド 126
MIGRATE DATABASE コマンド 634
MQ Listener コマンド 178

N

NetBIOS
ノード
アンカタログ 823

nochecklengths ファイル・タイプ修飾子 541, 608
nodefaults ファイル・タイプ修飾子 541
nodoubledel ファイル・タイプ修飾子 478, 541, 608
noeofchar ファイル・タイプ修飾子 541, 608
noheader ファイル・タイプ修飾子 608
NOLINEMACRO プリコンパイル・オプション 638
norowwarnings ファイル・タイプ修飾子 608
notypeid ファイル・タイプ修飾子 541
NULL 値
 コマンド行プロセッサの表記 350
NULL スtring 350
nullindchar ファイル・タイプ修飾子 541, 608

O

OPEN ステートメント
 CLP からの実行 857
optlevel プリコンパイル・オプション 638
output プリコンパイル・オプション 638
owner プリコンパイル/BIND オプション 389, 638

P

packages プリコンパイル・オプション 638
packeddecimal ファイル・タイプ修飾子 608
pagefreespace ファイル・タイプ修飾子 608
PING コマンド 636
PRECOMPILE コマンド 638
PREP コマンド 638
preprocessor プリコンパイル・オプション 638
PRUNE HISTORY/LOGFILE コマンド 665
PUT ROUTINE コマンド 667

Q

qualifier プリコンパイル/BIND オプション 389, 638
QUERY CLIENT コマンド 669
queryopt プリコンパイル/BIND オプション
 BIND コマンド 389
 PRECOMPILE コマンド 638

QUIESCE TABLESPACES FOR TABLE
 コマンド 673
QUIESCE コマンド 670
QUIT コマンド 676

R

REBIND コマンド 677
reclen ファイル・タイプ修飾子 541
 ロード 608
RECONCILE コマンド
 構文 681
RECOVER DATABASE コマンド 685
REDISTRIBUTE DATABASE PARTITION
 GROUP コマンド 692
REFRESH LDAP コマンド 696
REGISTER XMLSCHEMA コマンド
 構文 700
REGISTER XSROBJECT コマンド
 構文 702
REGISTER コマンド 697
release プリコンパイル / BIND オプション 389, 638
REORG TABLE コマンド 705
REORGCHK コマンド 716
RESET ADMIN CONFIGURATION コマンド 726
RESET ALERT CONFIGURATION コマンド 728
RESET DATABASE CONFIGURATION
 コマンド 730
RESET DATABASE MANAGER
 CONFIGURATION コマンド 732
RESET MONITOR コマンド 734
Reset rollforward pending state コマンド 260
RESTART DATABASE コマンド 736
RESTORE DATABASE コマンド 738
REWIND TAPE コマンド 755
ROLLFORWARD DATABASE コマンド 756
RUNCMD コマンド 768
RUNSTATS コマンド
 構文 769

S

SELECT ステートメント
 CLP からの実行 857
 EXPORT コマンド内の 478
SET CLIENT コマンド 784
SET RUNTIME DEGREE コマンド 787
SET TABLESPACE CONTAINERS コマンド 789
SET TAPE POSITION コマンド 791

SET UTIL_IMPACT_PRIORITY コマンド 792
SET WRITE コマンド 794
setup コマンド 332
SIGALRM 信号
 開始、データベース・マネージャの 796
SIGINT 信号
 開始、データベース・マネージャの 796
SOCKS ノード
 パラメーター 426
SQL および XQuery Explain コマンド 105
SQL ステートメント
 コマンド行の使用 857
 ヘルプの表示 910
 ヘルプへのアクセス 339
SQL92 準拠 SQL ステートメント・プロセッサ・コマンド 269
sqlca プリコンパイル・オプション 638
sqlerror プリコンパイル/BIND オプション 389, 638
sqlflag プリコンパイル・オプション 638
sqlj コマンド 334
SQLJ プロファイル・バインド・プログラム・コマンド 272
SQLJ 変換プログラム・コマンド 334
sqlrules プリコンパイル・オプション 638
sqlwarn プリコンパイル/BIND オプション 389, 638
START DATABASE MANAGER コマンド 796
START HADR コマンド 803
STOP DATABASE MANAGER コマンド 806
STOP HADR コマンド 810
strdel プリコンパイル/BIND オプション 389, 638
striptblanks ファイル・タイプ修飾子 541, 608
striptnulls ファイル・タイプ修飾子 541, 608
subtableconvert ファイル・タイプ修飾子 608
syncpoint プリコンパイル・オプション 638

T

TAKEOVER HADR コマンド 812
target プリコンパイル・オプション 638
TCP/IP
 ノード
 アンカタログ 823

TERMINATE コマンド 815
text プリコンパイル / BIND オプション
389, 638
timeformat ファイル・タイプ修飾子 541,
608
timestampformat ファイル・タイプ修飾子
541, 608
totalfreespace ファイル・タイプ修飾子
608
transform group プリコンパイル/BIND オ
プション 389, 638
True Type フォント
コマンド行プロセッサの要件 350
TSM アーカイブ・イメージ 15
TSM アーカイブ・イメージによる作業コ
マンド 15

U

UNCATALOG DATABASE コマンド
816
UNCATALOG DCS DATABASE コマンド
818
UNCATALOG LDAP DATABASE コマン
ド 820
UNCATALOG LDAP NODE コマンド
822
UNCATALOG NODE コマンド 823
UNCATALOG ODBC DATA SOURCE コ
マンド 824
UNQUIESCE コマンド 825
UPDATE ADMIN CONFIGURATION コ
マンド 827
UPDATE ALERT CONFIGURATION コマ
ンド 829
UPDATE ALTERNATE SERVER FOR
DATABASE コマンド 834
UPDATE ALTERNATE SERVER FOR
LDAP DATABASE コマンド 836
UPDATE CLI CONFIGURATION コマン
ド 838
UPDATE COMMAND OPTIONS コマンド
840
UPDATE CONTACT コマンド 842
UPDATE CONTACTGROUP コマンド
843
UPDATE DATABASE CONFIGURATION
コマンド 844
UPDATE DATABASE MANAGER
CONFIGURATION コマンド 847
UPDATE HEALTH NOTIFICATION
CONTACT LIST コマンド 849
UPDATE HISTORY FILE コマンド 850
UPDATE LDAP NODE コマンド 852
UPDATE MONITOR SWITCHES コマン
ド 854

usedefaults ファイル・タイプ修飾子 541,
608

V

V5 セマンティクスへのユニーク索引変換
の準備コマンド 314
validate プリコンパイル/BIND オプション
389, 638
version プリコンパイル・オプション 638
Visual Explain
チュートリアル 914

W

WCHARTYPE プリコンパイラー・オブシ
ョン
プリコンパイル・コマンドを使用した
638
Windows フェールオーバー・ユーティリ
ティーのセットアップ・コマンド 182

X

XBSA (バックアップ・サービス
API) 382
XML スキーマ・リポジトリ
ADD XMLSCHEMA DOCUMENT コ
マンド 371
COMPLETE XMLSCHEMA コマンド
434
REGISTER XMLSCHEMA コマンド
700
REGISTER XSROBJECT コマンド
702

[特殊文字]

! シェル・コマンド 339



Printed in Japan

SD88-6750-00



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:

IBM DB2 DB2 Version 9

コマンド・リファレンス

