

IBM® DB2® Universal Database



## コマンド・リファレンス

バージョン 8.2



IBM® DB2® Universal Database



## コマンド・リファレンス

バージョン 8.2

**ご注意！**

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

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： SC09-4828-01  
IBM® DB2® Universal Database  
Command Reference  
Version 8.2

発 行： 日本アイ・ピー・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 2004.8

この文書では、平成明朝体™W3、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™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 1993-2004. All rights reserved.

© Copyright IBM Japan 2004

# 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
---	---

dasrct - 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	7
-------------------	---

db2admin - DB2 Administration Server	8
--------------------------------------	---

db2adutl - TSM 内の DB2 オブジェクトの管理	10
---------------------------------	----

db2advis - DB2 設計アドバイザー	17
-------------------------	----

db2atld - オートローダー	22
-------------------	----

db2audit - 監査機能管理者ツール	23
-----------------------	----

db2batch - ベンチマーク・ツール	24
-----------------------	----

db2bfd - バインド・ファイル記述ツール	30
-------------------------	----

db2cap - CLI/ODBC 静的パッケージ・バインディング・ツール	31
---------------------------------------	----

db2cc - コントロール・センターの開始	33
------------------------	----

db2cfexp - 接続構成エクスポート・ツール	35
---------------------------	----

db2cfimp - 接続構成インポート・ツール	37
--------------------------	----

db2cidmg - リモート・データベース移行	38
--------------------------	----

db2ckbkp - バックアップの検査	39
----------------------	----

db2ckmig - データベース事前移行ツール	43
--------------------------	----

db2ckrst - 増分リストア・イメージ順序の検査	45
-----------------------------	----

db2cli - DB2 対話機能 CLI	47
-----------------------	----

db2cmd - DB2 コマンド・ウィンドウのオープン	48
------------------------------	----

db2dart - データベース分析およびレポート・ツール	49
-------------------------------	----

db2dclgn - 宣言生成プログラム	53
----------------------	----

db2demigdbd - データベース・ディレクトリー・ファイルの逆マイグレーション	56
---	----

db2diag - db2diag.log 分析ツール	58
-----------------------------	----

db2dlm_upd_hostname - データ・リンク・ホスト名の更新	69
---------------------------------------	----

db2drdat - DRDA トレース	71
----------------------	----

db2drvmp - DB2 データベースのドライブ・マップ	73
--------------------------------	----

db2empfa - 複数ページ・ファイル割り振りの使用可能化	75
---------------------------------	----

db2eva - イベント・アナライザー	76
----------------------	----

db2evmon - イベント・モニター生産性向上ツール	78
------------------------------	----

db2evtbl - イベント・モニターのターゲット表定義の生成	79
----------------------------------	----

db2exfmt - Explain 表フォーマット	81
----------------------------	----

db2expln - SQL Explain	83
------------------------	----

db2flsn - ログ・シーケンス番号の検出	89
-------------------------	----

db2fm - DB2 障害モニター	91
--------------------	----

db2fs - ファースト・ステップ	93
--------------------	----

db2gcf - DB2 インスタンスの制御	94
------------------------	----

db2gov - DB2 管理プログラム	96
----------------------	----

db2govlg - DB2 管理プログラム・ログ照会	98
-----------------------------	----

db2gpmmap - パーティション・マップの取得	99
----------------------------	----

db2hc - ヘルス・センターの開始	100
---------------------	-----

db2iauto - インスタンスの自動開始	101
------------------------	-----

db2iclus - Microsoft Cluster Server	102
-------------------------------------	-----

db2icons - DB2 アイコンの追加	105
------------------------	-----

db2icrt - インスタンスの作成	106
---------------------	-----

db2idrop - インスタンスの除去	110
----------------------	-----

db2ilist - インスタンスのリスト	112
-----------------------	-----

db2imigr - インスタンスの移行	114
----------------------	-----

db2inidb - ミラーリングされたデータベースの初期化	116
--------------------------------	-----

db2inspf - 検査結果のフォーマット	118
------------------------	-----

db2isetup - インスタンス作成インターフェースの開始	119
---------------------------------	-----

db2iupdt - インスタンスの更新	121
----------------------	-----

db2jdbcbind - DB2 JDBC パッケージ・バインド・プログラム	124
---	-----

db2ldcfg - LDAP 環境の構成	126
-----------------------	-----

db2level - DB2 サービス・レベルの表示	127
----------------------------	-----

db2licm - ライセンス管理ツール	128
----------------------	-----

db2logsforrwd - ロールフォワード・リカバリーに必要なログのリスト	130
--	-----

db2look - DB2 統計および DDL 抽出ツール	131
-------------------------------	-----

db2move - データベース移動ツール	141
-----------------------	-----

db2mqslsn - MQ Listener	147
-------------------------	-----

db2mscs - Windows フェイルオーバー・ユーティリティのセットアップ	151
---	-----

db2mtrk - メモリー・トラッカー	155
----------------------	-----

db2nchg - データベース・パーティション・サーバー構成の変更	158
------------------------------------	-----

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	160
--	-----

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	163
--	-----

db2osconf - カーネル・パラメーター値のためのユーティリティ	165
-------------------------------------	-----

db2pd - DB2 のモニターおよびトラブルシューティング	168
---------------------------------	-----

db2perf - データベース・パフォーマンス値のリセット	197
--------------------------------	-----

db2perf - パフォーマンス・カウンター登録ユーティリティ	199
----------------------------------	-----

db2perf - パフォーマンス・モニター登録ツール	200
-----------------------------	-----

db2rbind - すべてのパッケージの再バインド	201
----------------------------	-----

db2_recon_aid - 複数の表の RECONCILE	203
---------------------------------	-----

db2relocatedb - データベースの再配置	206
----------------------------	-----

db2rfpen - ロールフォワード・ペンディング状態に設定	211
---------------------------------	-----

db2rmicons - DB2 アイコンの除去	212
--------------------------	-----

db2rspgn - 応答ファイル生成プログラム (Windows)	213
------------------------------------	-----

db2sampl - サンプル・データベースの作成 . . . . .	214
db2secv82 - DB2 オブジェクトの許可の設定 . . . . .	215
db2set - DB2 プロファイル・レジストリー . . . . .	216
db2setup - DB2 のインストール . . . . .	219
db2sql92 - SQL92 準拠 SQL ステートメント・プロセッサ . . . . .	220
db2sqljbind - DB2 SQLJ プロファイル・バインド・プログラム . . . . .	223
db2sqljcustomize - DB2 SQLJ プロファイル・カスタマイザー . . . . .	227
db2sqljprint - DB2 SQLJ プロファイル・プリンター . . . . .	234
db2start - DB2 の開始 . . . . .	235
db2stop - DB2 の停止 . . . . .	236
db2support - 問題分析および環境収集ツール . . . . .	237
db2sync - DB2 シンクロナイザーの開始 . . . . .	240
db2sysstray - DB2 システム・トレイの開始 . . . . .	241
db2tapemgr - テープ上のログ・ファイルの管理 . . . . .	242
db2tblst - 表スペース状態の獲得 . . . . .	245
db2trc - トレース . . . . .	246
db2undgp - 実行特権の取り消し . . . . .	249
db2uiddl - V5 セマンティクスへのユニーク索引変換の準備 . . . . .	250
db2untag - コンテナ・タグの解放 . . . . .	251
db2updv8 - バージョン 8 の現行レベルへのデータベースの更新 . . . . .	252
disable_MQFunctions . . . . .	254
enable_MQFunctions . . . . .	256
setup - DB2 のインストール . . . . .	259
sqlj - DB2 SQLJ 変換プログラム . . . . .	260

## 第 2 章 コマンド行プロセッサ (CLP) 263

db2 - コマンド行プロセッサの呼び出し . . . . .	263
コマンド行プロセッサのオプション . . . . .	264
コマンド行プロセッサの戻りコード . . . . .	272
コマンド行プロセッサ (CLP) . . . . .	273

## 第 3 章 CLP コマンド . . . . . 279

DB2 CLP コマンド . . . . .	279
ACTIVATE DATABASE . . . . .	283
ADD CONTACT . . . . .	285
ADD CONTACTGROUP . . . . .	286
ADD DATALINKS MANAGER . . . . .	287
ADD DBPARTITIONNUM . . . . .	289
ARCHIVE LOG . . . . .	291
ATTACH . . . . .	294
AUTOCONFIGURE . . . . .	296
BACKUP DATABASE . . . . .	299
BIND . . . . .	305
CATALOG APPC NODE . . . . .	324
CATALOG APPN NODE . . . . .	327
CATALOG DATABASE . . . . .	330
CATALOG DCS DATABASE . . . . .	334
CATALOG LDAP DATABASE . . . . .	337
CATALOG LDAP NODE . . . . .	341
CATALOG LOCAL NODE . . . . .	342
CATALOG NAMED PIPE NODE . . . . .	344

CATALOG NETBIOS NODE . . . . .	346
CATALOG ODBC DATA SOURCE . . . . .	348
CATALOG TCPIP NODE . . . . .	349
CHANGE DATABASE COMMENT . . . . .	353
CHANGE ISOLATION LEVEL . . . . .	355
CREATE DATABASE . . . . .	357
CREATE TOOLS CATALOG . . . . .	367
DEACTIVATE DATABASE . . . . .	370
DEREGISTER . . . . .	372
DESCRIBE . . . . .	373
DETACH . . . . .	377
DROP CONTACT . . . . .	378
DROP CONTACTGROUP . . . . .	379
DROP DATABASE . . . . .	380
DROP DATALINKS MANAGER . . . . .	382
DROP DBPARTITIONNUM VERIFY . . . . .	386
DROP TOOLS CATALOG . . . . .	388
ECHO . . . . .	390
EDIT . . . . .	391
EXPORT . . . . .	393
エクスポート用のファイル・タイプ修飾子 . . . . .	398
データ移動での区切り文字の制限 . . . . .	403
FORCE APPLICATION . . . . .	405
GET ADMIN CONFIGURATION . . . . .	407
GET ALERT CONFIGURATION . . . . .	409
GET AUTHORIZATIONS . . . . .	415
GET CLI CONFIGURATION . . . . .	416
GET CONNECTION STATE . . . . .	418
GET CONTACTGROUP . . . . .	419
GET CONTACTGROUPS . . . . .	420
GET CONTACTS . . . . .	421
GET DATABASE CONFIGURATION . . . . .	422
GET DATABASE MANAGER CONFIGURATION . . . . .	428
GET DATABASE MANAGER MONITOR SWITCHES . . . . .	433
GET DESCRIPTION FOR HEALTH INDICATOR . . . . .	436
GET HEALTH NOTIFICATION CONTACT LIST . . . . .	438
GET HEALTH SNAPSHOT . . . . .	439
GET INSTANCE . . . . .	442
GET MONITOR SWITCHES . . . . .	443
GET RECOMMENDATIONS . . . . .	446
GET ROUTINE . . . . .	450
GET SNAPSHOT . . . . .	452
HELP . . . . .	481
HISTORY . . . . .	482
IMPORT . . . . .	483
インポート用のファイル・タイプ修飾子 . . . . .	497
データ移動での区切り文字の制限 . . . . .	507
INITIALIZE TAPE . . . . .	509
INSPECT . . . . .	510
LIST ACTIVE DATABASES . . . . .	515
LIST APPLICATIONS . . . . .	517
LIST COMMAND OPTIONS . . . . .	519
LIST DATABASE DIRECTORY . . . . .	520
LIST DATABASE PARTITION GROUPS . . . . .	524
LIST DATALINKS MANAGERS . . . . .	526

LIST DBPARTITIONNUMS . . . . .	527	UNCATALOG DATABASE . . . . .	762
LIST DCS APPLICATIONS . . . . .	528	UNCATALOG DCS DATABASE . . . . .	764
LIST DCS DIRECTORY . . . . .	531	UNCATALOG LDAP DATABASE . . . . .	766
LIST DRDA INDOUBT TRANSACTIONS . . . . .	533	UNCATALOG LDAP NODE . . . . .	768
LIST HISTORY . . . . .	535	UNCATALOG NODE . . . . .	769
LIST INDOUBT TRANSACTIONS . . . . .	538	UNCATALOG ODBC DATA SOURCE . . . . .	770
LIST NODE DIRECTORY . . . . .	542	UNQUIESCE . . . . .	771
LIST ODBC DATA SOURCES . . . . .	545	UPDATE ADMIN CONFIGURATION . . . . .	773
LIST PACKAGES/TABLES . . . . .	546	UPDATE ALERT CONFIGURATION . . . . .	775
LIST TABLESPACE CONTAINERS . . . . .	549	UPDATE ALTERNATE SERVER FOR DATABASE . . . . .	779
LIST TABLESPACES . . . . .	551	UPDATE ALTERNATE SERVER FOR LDAP	
LIST UTILITIES . . . . .	556	DATABASE . . . . .	781
LOAD . . . . .	558	UPDATE CLI CONFIGURATION . . . . .	782
ロード用のファイル・タイプ修飾子 . . . . .	582	UPDATE COMMAND OPTIONS . . . . .	784
データ移動での区切り文字の制限 . . . . .	594	UPDATE CONTACT . . . . .	786
LOAD QUERY . . . . .	596	UPDATE CONTACTGROUP . . . . .	787
MIGRATE DATABASE . . . . .	599	UPDATE DATABASE CONFIGURATION . . . . .	788
PING . . . . .	601	UPDATE DATABASE MANAGER	
PRECOMPILE . . . . .	603	CONFIGURATION . . . . .	791
PRUNE HISTORY/LOGFILE . . . . .	630	UPDATE HEALTH NOTIFICATION CONTACT	
PUT ROUTINE . . . . .	632	LIST . . . . .	793
QUERY CLIENT . . . . .	634	UPDATE HISTORY FILE . . . . .	794
QUIESCE . . . . .	635	UPDATE LDAP NODE . . . . .	796
QUIESCE TABLESPACES FOR TABLE . . . . .	638	UPDATE MONITOR SWITCHES . . . . .	798
QUIT . . . . .	641		
REBIND . . . . .	642	<b>第 4 章 コマンド行 SQL ステートメン</b>	
RECONCILE . . . . .	646	<b>トの使用 . . . . . 801</b>	
RECOVER DATABASE . . . . .	650	<b>付録 A. 構文図の読み方 . . . . . 809</b>	
REDISTRIBUTE DATABASE PARTITION GROUP . . . . .	656	<b>付録 B. 命名規則 . . . . . 813</b>	
REFRESH LDAP . . . . .	660	<b>付録 C. DB2 Universal Database の技</b>	
REGISTER . . . . .	661	<b>術情報 . . . . . 815</b>	
REORG INDEXES/TABLE . . . . .	666	DB2 資料とヘルプ . . . . .	815
REORGCHK . . . . .	674	DB2 資料の更新 . . . . .	815
RESET ADMIN CONFIGURATION . . . . .	685	DB2 インフォメーション・センター . . . . .	816
RESET ALERT CONFIGURATION . . . . .	687	DB2 インフォメーション・センターのインストー	
RESET DATABASE CONFIGURATION . . . . .	689	ル・シナリオ . . . . .	818
RESET DATABASE MANAGER CONFIGURATION . . . . .	691	DB2 セットアップ・ウィザードを使用した DB2 イ	
RESET MONITOR . . . . .	693	ンフォメーション・センターのインストール	
RESTART DATABASE . . . . .	695	(UNIX) . . . . .	820
RESTORE DATABASE . . . . .	697	DB2 セットアップ・ウィザードを使用した DB2 イ	
REWIND TAPE . . . . .	708	ンフォメーション・センターのインストール	
ROLLFORWARD DATABASE . . . . .	709	(Windows) . . . . .	823
RUNCMD . . . . .	719	DB2 インフォメーション・センターの呼び出し	826
RUNSTATS . . . . .	720	コンピューターまたはイントラネット・サーバーへ	
SET CLIENT . . . . .	732	の DB2 インフォメーション・センターの更新イン	
SET RUNTIME DEGREE . . . . .	735	ストール . . . . .	827
SET TABLESPACE CONTAINERS . . . . .	737	DB2 インフォメーション・センターにおける特定	
SET TAPE POSITION . . . . .	739	の言語でのトピックの表示 . . . . .	828
SET UTIL_IMPACT_PRIORITY . . . . .	740	DB2 PDF 資料および印刷された資料 . . . . .	829
SET WRITE . . . . .	742	DB2 の基本情報 . . . . .	829
START DATABASE MANAGER . . . . .	744	管理情報 . . . . .	830
START HADR . . . . .	750	アプリケーション開発情報 . . . . .	831
STOP DATABASE MANAGER . . . . .	752	ビジネス・インテリジェンス情報 . . . . .	831
STOP HADR . . . . .	756		
TAKEOVER HADR . . . . .	758		
TERMINATE . . . . .	761		

DB2 Connect 情報 . . . . .	832	アクセス支援 . . . . .	841
入門情報 . . . . .	832	キーボードによる入力およびナビゲーション . . . . .	841
チュートリアル情報 . . . . .	833	アクセスしやすい表示 . . . . .	842
オプション・コンポーネント情報 . . . . .	833	支援テクノロジーとの互換性 . . . . .	842
リリース・ノート . . . . .	834	アクセスしやすい資料 . . . . .	842
PDF ファイルからの DB2 資料の印刷方法 . . . . .	835	ドット 10 進シンタックス・ダイアグラム . . . . .	842
DB2 の印刷資料の注文方法 . . . . .	836	DB2 Universal Database 製品の共通基準認証 . . . . .	845
DB2 ツールからコンテキスト・ヘルプを呼び出す . . . . .	836	<b>付録 D. 特記事項 . . . . . 847</b>	
コマンド行プロセッサからメッセージ・ヘルプを		商標 . . . . .	849
呼び出す . . . . .	838	<b>付録 E. IBM と連絡をとる . . . . . 851</b>	
コマンド行プロセッサからコマンド・ヘルプを呼		製品情報 . . . . .	851
び出す . . . . .	838	<b>索引 . . . . . 853</b>	
コマンド行プロセッサから SQL 状態ヘルプを呼			
び出す . . . . .	839		
DB2 チュートリアル . . . . .	839		
DB2 トラブルシューティング情報 . . . . .	840		



---

## 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 Universal Database 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。



---

## 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のために、オペレーティング・システムのコマンド・プロンプトで入力するか、またはシェル・スクリプトに含めることが可能なコマンドについて説明します。

注:

1. ディレクトリー・パス中の斜線 (/) は UNIX ベースのシステムだけに用いられるもので、Windows オペレーティング・システムの円記号 (¥) に相当します。
2. 用語 Windows は、通常 Microsoft Windows のサポートされるすべてのバージョンのことです。サポートされるバージョンには、Windows NT ベースのバージョン、および Windows 9x ベースのバージョンです。問題の機能が Windows NT 4、Windows 2000、Windows .NET および Windows XP ではサポートされていて、Windows 9x ではサポートされていない場合、「Window NT ベースのオペレーティング・システム」という特定の言及がなされることがあります。  
Windows の特定のバージョンに適用する機能がある場合、オペレーティング・システムの有効なバージョンが注記されます。

---

### 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。

**権限:**

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、または接続の確立のどれかです。機能が正常に作動するために、データベース接続またはインスタンス・アタッチが必要かどうか、または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もあります。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。

**コマンド構文:**

## システム・コマンド

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、809 ページの『付録 A. 構文図の読み方』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。

### 関連資料:

関連情報の相互参照です。



### 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。

このコマンドは、UNIX ベースのシステムのみで使用可能です。Windows システムでは、同じ目的でコマンド **db2admin create** を使用できます。

#### 権限:

Root 権限。

#### 必要な接続:

なし。

#### コマンド構文:

▶ `dasCRT -u DASuser` ◀

#### 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。

#### 使用上の注意:

- DB2 の前のバージョンでは、このコマンドは **dasicrt** でした。
- dasCRT** コマンドは、`DB2DIR/instance` ディレクトリーにあります。DB2DIR は、AIX 上では `/usr/opt/db2_08_01`、他のすべての UNIX ベース・システムでは `/opt/IBM/db2/V8.1` です。

---

## dasdrop - DB2 Administration Server の除去

UNIX オペレーティング・システムでのみ、DB2 Administration Server (DAS) を除去します。 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。

### 権限:

Root 権限。

### 必要な接続:

なし。

### コマンド構文:

▶— dasdrop —▶

### 使用上の注意:

- dasdrop コマンドは、インストール済み DB2 バージョンおよびリリースに特定のサブディレクトリーの下にある、instance サブディレクトリーにあります。
- フィックスパックまたはモディフィケーション・レベルが代替パスにインストールされている場合、インストール・パスから dasdrop ユーティリティーを実行することによって DAS をドロップできます。これを行うには、ドロップする DAS のインストール・パスにインストール・コードが置かれている必要があります。インストール・パスからインストール・コードを除去し、その後別のインストール・パスから dasdrop ユーティリティーを呼び出してそのパスにある DAS を除去しようとしても、DAS は除去できません。

### 関連タスク:

- 「管理ガイド: インプリメンテーション」の『DAS の除去』

# dasmigr - DB2 Administration Server の移行

インストールに続き、DB2 Administration Server を移行します。

UNIX ベース・システムでは、このユーティリティーは DB2DIR/instance ディレクトリにあります。DB2DIR は、AIX 上では /usr/opt/db2\_08\_01、他のすべての UNIX ベース・システムでは /opt/IBM/db2/V8.1 です。Windows オペレーティング・システムでは、このユーティリティーは sqllib¥bin サブディレクトリにあります。

### 権限:

UNIX ベース・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:

UNIX の場合:

```
▶▶ dasmigr—previous_das_name—new_das_name—————▶▶
```

Windows の場合:

```
▶▶ dasmigr—new_das_name—————▶▶
```

### コマンド・パラメーター:

*previous\_das\_name*

移行元のバージョンの DAS の名前。このパラメーターは、Windows では無効です。

*new\_das\_name*

移行先のバージョンの DAS の名前。

### 例:

```
dasmigr db2as dasusr1
```

### 使用上の注意:

DB2 Administration Server の移行では、ツール・カタログ・データベースを作成し、接続可能な状態にしておくことが必要です。

### 関連タスク:

- 「管理ガイド: インプリメンテーション」の『DAS の構成』
- 「DB2 Universal Database サーバー機能 概説およびインストール」の『DB2 Administration Server (DAS) の移行』

### 関連資料:

- 367 ページの『CREATE TOOLS CATALOG』



## dasupdt - DAS の更新

UNIX 系のオペレーティング・システムでは、プログラム一時修正 (PTF) またはコード・パッチのインストールによって DB2 を更新する場合、**dasupdt** によって各 DB2 Administration Server (DAS) を更新します。このツールは、DB2DIR/instance にあります。DB2DIR は、AIX 上では /usr/opt/db2\_08\_01、他のすべての UNIX ベース・システムでは /opt/IBM/db2/V8.1 です。

### 権限:

Root 権限。

### 必要な接続:

なし

### コマンド構文:

```

▶▶ dasupdt [-d] [-D] [-h] [-?]

```

### コマンド・パラメーター:

- d** デバッグ・モードを設定し、問題分析に使用します。
- D** 1 つのパスのより高いコード・レベルから、別のパスにインストールされたより低いコード・レベルに DAS を移します。
- h/-?** 使用情報を表示します。

### 例:

DAS がバージョン 8 インストール・パス内のバージョン 8.1.2 コードを実行しています。バージョン 8 インストール・パスにフィックスパック 3 がインストール済みの場合、バージョン 8 インストール・パスから以下のコマンドを呼び出すことにより、DAS をフィックスパック 3 に更新します。

```
dasupdt
```

DAS が代替インストール・パス内のバージョン 8.1.2 コードを実行しています。フィックスパック 1 を別の代替インストール・パスにインストール済みの場合、フィックスパック 1 の代替インストール・パスから以下のコマンドを呼び出すと、DAS は、フィックスパック 1 の代替インストール・パスから実行されるフィックスパック 1 に更新されます。

```
dasupdt -D
```

## db2admin - DB2 Administration Server

このユーティリティーは、DB2 Administration Server の管理に使用します。

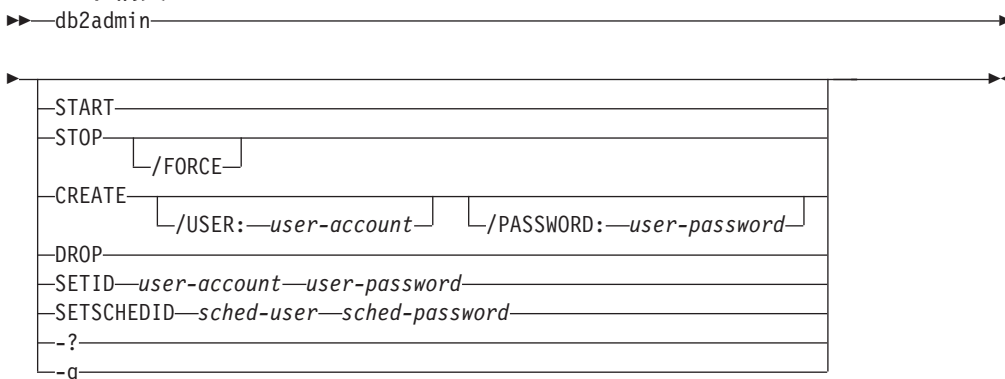
**権限:**

Windows ではローカル管理者、 UNIX ベースのシステムでは DASADM。

**必要な接続:**

なし

**コマンド構文:**



**コマンド・パラメーター:**

**注:** パラメーターを指定せず、かつ DB2 Administration Server が存在する場合、このコマンドは DB2 Administration Server の名前を戻します。

### START

DB2 Administration Server を開始します。

### STOP /FORCE

DB2 Administration Server を停止します。 force オプションを指定した場合は、要求のサービスの処理中であるかどうかに関係なく、強制的に DB2 Administration Server を停止させる場合に使用します。

### CREATE /USER: user-account /PASSWORD: user-password

DB2 Administration Server 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、DB2 Administration Server がこのユーザー・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。指定したユーザー・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・データベース内になければなりません。DB2 Administration Server の機能すべてにアクセスできるように、ユーザー・アカウントを指定することをお勧めします。

**注:** UNIX 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 DB2 Administration Server を削除します。

**注:** UNIX から DAS をドロップする場合は、dasdrop コマンドを使用する必要があります。

**SETID** *user-account/user-password*

DB2 Administration Server に関連付けられたユーザー・アカウントを設定または修正します。

**SETSCHEDID** *sched-user/sched-password*

ツール・カタログ・データベースに接続するためにスケジューラーで使用するログオン・アカウントを確立します。これは、スケジューラーが使用可能になっている場合で、ツール・カタログ・データベースが DB2 Administration Server のリモート側に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。

**使用上の注意:**

UNIX ベースのオペレーティング・システムの場合、db2admin コマンドの実行可能ファイルは、home/DASuser/das/bin ディレクトリーにあります (ここで、DASuser には DB2 Administration Server ユーザーの名前が入ります)。Windows の場合は、db2admin 実行可能ファイルは sqllib/bin ディレクトリーにあります。

**関連資料:**

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

## db2adutl - TSM 内の DB2 オブジェクトの管理

Tivoli Storage Manager 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 ベースのオペレーティング・システムでは、このユーティリティーは `sqllib/adsm` ディレクトリーにあります。Windows では、これは `sqllib\bin` にあります。

権限:

なし

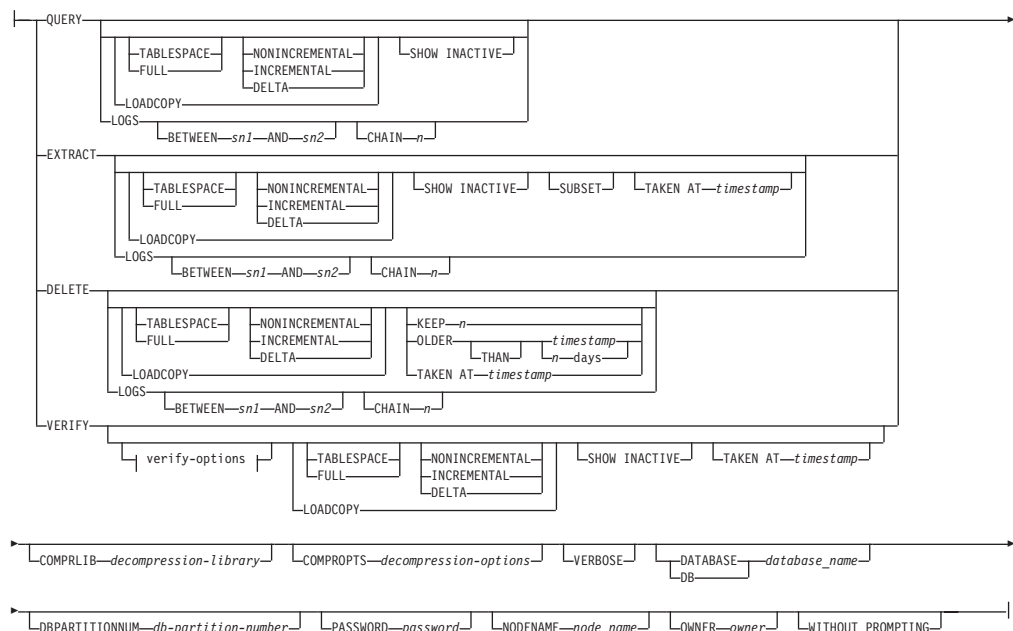
必要な接続:

なし

コマンド構文:

```
db2adutl db2-object-options access-control-options
```

**db2-object-options:**



**verify-options:**

ALL
CHECK
DMS
HEADER
LFH
TABLESPACES
HEADERONLY
TABLESPACESONLY
OBJECT
PAGECOUNT

**access-control-options:**

GRANT	ALL	ON	ALL	FOR	DATABASE	database_name
	USER—user_name		NODENAME—node_name		DB	
REVOKE	ALL	ON	ALL	FOR	ALL	
	USER—user_name		NODENAME—node_name		DATABASE	database_name
					DB	
QUERYACCESS	FOR	ALL				
		DATABASE	database_name			
		DB				
PASSWORD	password					

**コマンド・パラメーター:****QUERY**

TSM サーバーで DB2 オブジェクトを照会します。

**EXTRACT**

TSM サーバーからの DB2 オブジェクトを、ローカル・マシンにある現行ディレクトリーにコピーします。

**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。

**注:** 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** 使用可能な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。

**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>

DMS 表スペースの場合、

D <tbodyID> <objType> <startPage> <numPages>

注: <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。

ログ・ファイルの場合、

L <log num> <startPos> <numPages>

その他のデータ (初期データなど) の場合、

0 <objType> <startPos> <numBytes>

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

#### **TAKEN AT** *timestamp*

タイム・スタンプを基準としてバックアップ・イメージを指定します。

#### **KEEP** *n*

タイム・スタンプで最新の *n* 個を除き、指定したタイプのすべてのオブジェクトを非活動化します。

#### **OLDER THAN** *timestamp* または *n days*

*timestamp* または *n* 日より前のタイム・スタンプが付けられているオブジェクトを非活動化することを指定します。

#### **COMPRLIB** *decompression-library*

解凍を実行するために使用するライブラリーの名前。この名前は、サーバー上の 1 個のファイルを参照する完全修飾パスでなければなりません。このパラメーターを指定しない場合、DB2 はイメージ内に格納されているライブラリーの使用を試みます。バックアップが圧縮されていなかった場合、このパラメーターの値は無視されます。指定されたライブラリーをロードできない場合、操作は失敗します。

#### **COMPROPTS** *decompression-options*

バイナリー・データのうち、解凍ライブラリーの初期設定ルーチンに渡すブロックを記述します。DB2 はこのストリングをクライアントからサーバーに直接渡すため、バイト反転やコード・ページ変換の問題がある場合は解凍ライブラリーで処理する必要があります。データ・ブロックの最初の文字が '@' なら、データの残りの部分は、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で発行された db2adutl query コマンドの出力例です。バックアップ・イメージのタイム・スタンプに注意してください。

Query for database 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.

## db2adutl - TSM 内の DB2 オブジェクトの管理

No LOCAL COPY images found for RAWAMPL

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:

- 「管理ガイド: パフォーマンス」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

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、ワークロード中の各ステートメントの重要性をランク付けし、ワークロード中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。

構造型列は、このコマンドの実行時には考慮されません。

### 権限:

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。

### 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:

```

▶ db2advis [-d database-name] [-w workload-name] [-s "statement"] [-i filename] [-g] [-qp]
           [-a userid [/passwd]] [-m advise-type] [-x] [-u]
           [-l disk-limit] [-n] [-t max-advise-time] [-k HIGH | MED | LOW | OFF]
           [-f] [-r] [-n schema-name] [-q schema-name]
           [-b tablespace-name] [-c tablespace-name] [-h] [-p]
           [-o outfile]

```

### 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**

索引がアドバイス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"**

索引がアドバイスされる単一の 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。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。値は大文字で入力しなければなりません。たとえば、db2advis -m PC は、パーティション化と MDC 表を推奨します。

**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) 表への変換を推奨します。

**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 (Refresh Deffered MQT) の推奨を考慮することを指定します。即時更新 MQT (Refresh Immediate 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、デフォルトのパーティション・キーが、MQT の暗黙のユニーク・キーになります。

**-l** *disk-limit*

既存のスキーマですべての索引に使用可能な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、生成に時間がかかるので、**db2advis** の実行時間が長くなることに注意してください。

**-n** *schema-name*

シミュレーション・カタログ表の修飾名と、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。

**-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*

カタログ・データベース・パーティション・グループ上でシミュレーション・カタログ表スペースを作成する表スペースの名前 (ファイル名またはディレクトリ) を指定します。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上で実行してください。このユーティリティが完了すると、表スペースの履歴が元に戻ります。パーティション・データベース環境の場合、ユーザー作成の表スペースは、データベースのカatalog・パーティション上でのみ作成する必要があります。

- h** ヘルプ情報を表示します。 このオプションを指定すると、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。
- 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。

#### 例:

1. 次の例では、ユーティリティはデータベース **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、索引、MCT と基本表の両方のパーティション化と MDC を取得するには、次のように -m オプションに値 IMCP を指定して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ています。

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>
```

適切な時間間隔で待機します。

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、ユニークな ID を割り当てます。

#### 関連概念:

- 「管理ガイド: パフォーマンス」の『設計アドバイザー』

---

## db2atld - オートローダー

オートローダーは、MPP 環境でデータを区分化またはロードするツールです。このユーティリティーは以下の処理を行います。

- あるシステム (たとえば、MVS) から AIX システム (RS/6000 または SP2) へデータを転送する
- データを並列して区分化する
- 対応している複数のデータベース区分化でデータを同時にロードする

**関連資料:**

- 558 ページの『LOAD』



---

## db2audit - 監査機能管理者ツール

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。

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。

## 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。パーティション・データベースで作業していて、`-r` オプションを使用して出力ファイルに名前を付ける場合、各データベース・パーティションからの出力は、それらの各データベース・パーティションと同じ名前を持つ別々のファイルに入れられます。ただし、指定されたファイルが NFS でマウントされたファイル・システムにある場合は、例外です。複数パーティション・データベースでこのようなケースが生じた場合は、すべての結果がこのファイルに保持されます。

### 権限:

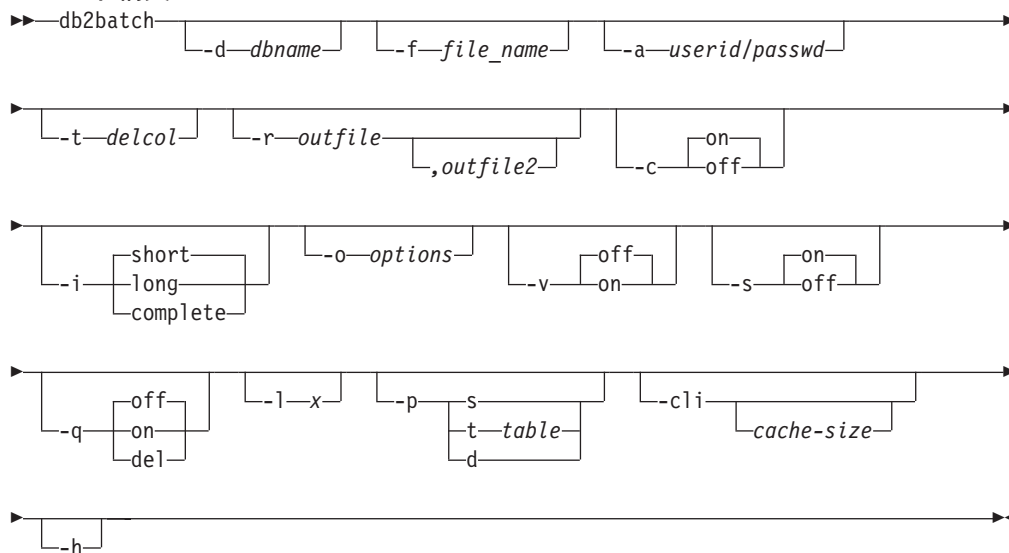
読み取られる SQL ステートメントが必要とするものと同じ権限レベル。

並列モードでは、ユーザーは **db2\_all** を実行する権限がなければなりません。

### 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。このオプションが指定されていない場合、**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  $-1 \sim n$  です。デフォルトは  $-1$  (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。デフォルトは  $-1$  (取り出された行をすべて出力へ送る) です。

**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** 時間情報を戻さない。
- 1** 経過時間のみ戻す。
- 2** 経過時間と CPU 時間を戻す。
- 3** モニター情報のサマリーを戻す。
- 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。)

**注:** 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可能)。

**注:** 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 UDB 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。

### 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。

### 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。斜線 (/) を含めなければなりません。

### -t delcol

1 文字の列区切り記号を指定します。

**注:** タブの列区切り文字を含めるには、-t TAB を使用します。

### -r outfile

照会結果が入る出力ファイル。任意指定の出力ファイル 2 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)。

**short** カーソルのオープン、取り出しの完了、およびカーソルのクローズの所要時間です。

**long** ある照会の開始から次の照会の開始までの経過時間です。PAUSE と SLEEP の時間、およびコマンド・オーバーヘッドを含みます。

### 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 ～  $n$  です。デフォルトは -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** 時間情報を戻さない。
- 1** 経過時間のみ戻す。
- 2** 経過時間と CPU 時間を戻す。
- 3** モニター情報のサマリーを戻す。
- 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可能)。

デフォルト値は  $1$  です。  $1$  より大きい値は DB2 パージョン  $2$  と DB2 UDB 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。

**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。

- 0** 照会のみ実行 (デフォルト)。
- 1** 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。
- 2** 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。

**-v** 冗長。照会処理中に標準エラーに情報を送信します。デフォルトは OFF です。

**-s** サマリーの表。照会または照会のブロックごとに、サマリーの表を提供します。その表には、経過時間 (選択時)、CPU 時間 (選択時)、取り出された行

数、印刷された行数が含まれます。経過時間および CPU 時間が収集された場合、その算術平均および幾何平均が算出されます。

**-q** 照会の出力。有効な値は以下のとおりです。

**on** 照会の区切られていない 出力のみを印刷する。

**off** 照会と関連情報すべての出力を印刷する。これがデフォルトです。

**del** 照会の区切られた 出力のみを印刷する。

**-l x** 終了文字を指定します。

**-p** 並列 (ESE のみ)。このモードでは SELECT ステートメントしかサポートされていません。出力名には、完全修飾パスがなければなりません。有効な値は以下のとおりです。

**s** 単一表または併置結合照会。SELECT ステートメントには列関数だけを含めることはできません。これは、照会に追加される DBPARTITIONNUM 関数の要件です。このオプションを指定すると、DBPARTITIONNUM 関数が照会の WHERE 文節に追加され、一時表は作成されません。このオプションは、照会の FROM 文節に単一表が入っている場合、または FROM 文節に含まれる複数表が併置である場合にのみ有効です。

このオプションが指定され、照会に GROUP BY 文節が含まれる場合、GROUP BY に指定された列は、表パーティション・キーのスーパーセットでなければなりません。

**t table** エクスポート・データを取り込むステージング表として使用する既存の表の名前を指定します。照会に FROM 文節の複数の表が入っていて、その表が併置でない場合、結果セットは指定した表に挿入され、SELECT が同時にすべてのパーティションに対して発行され、エクスポート・データが入るファイルが生成されます。

**d** INSERT INTO ステートメントで使用するシステム表を IBMDEFAULTGROUP に作成します。照会に FROM 文節の複数の表が入っていて、その表が併置でない場合、結果セットは指定した表に挿入され、SELECT が同時にすべてのパーティションに対して発行され、エクスポート・データが入るファイルが生成されます。

ローカル 出力ファイルを指定した (**-r** オプションを使用する) 場合、各データベース・パーティションからの出力は、各データベース・パーティションで同じ名前を持つ別々のファイルに入れられます。NFS マウント・ファイル・システム上のファイルが指定された場合、出力はすべてこのファイルに入れられます。

**-cli** CLI モードで **db2batch** を実行する。デフォルトでは、組み込み動的 SQL を使用するようにになっています。ステートメント・メモリーは、**cache-size** パラメーターを使用して、手動で設定できます。

**cache-size**

ステートメント・メモリーのサイズ。ステートメントの数で表されます。デ

フォルト値は 25 です。ユーティリティーがすでに準備済みになっている SQL ステートメントを検出した場合、古いプランを再利用します。このパラメーターは、**db2batch** を CLI モードで実行する場合だけ設定できます。

**-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:**

1. SQL ステートメントの長さは、65 535 文字まで可能です。入力ファイル中ではテキスト行が 3 898 文字を超えることはできないので、長いステートメントは数行に分割する必要があります。ステートメントの末尾は区切り文字 (デフォルトはセミコロン) にしなければなりません。
2. 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。
3. 出力に LOB 列を含む SQL 照会は、サポートされていません。

**関連資料:**

- 220 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ーは、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。

### 権限:

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。



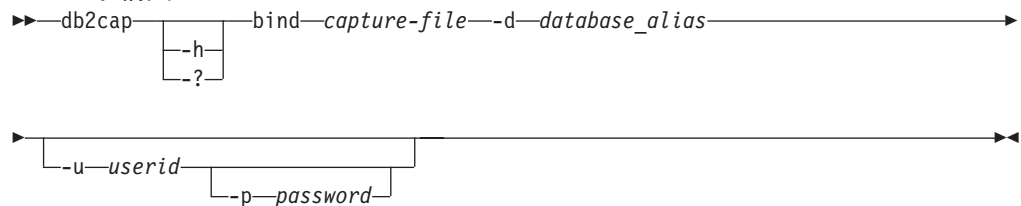
## db2cap - CLI/ODBC 静的パッケージ・バインディング・ツール

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・ファイルを処理して、アプリケーションが静的 SQL を実行するために CLI/ODBC/JDBC ドライバーによって使用されます。

### 権限:

- 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生成します。

#### **-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。

**注:** 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生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使用不可にしておきます。

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・ファイルからのパッケージ・キーワードの最初の 7 文字と、次の 1 文字の接尾部から構成されます。

- 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。

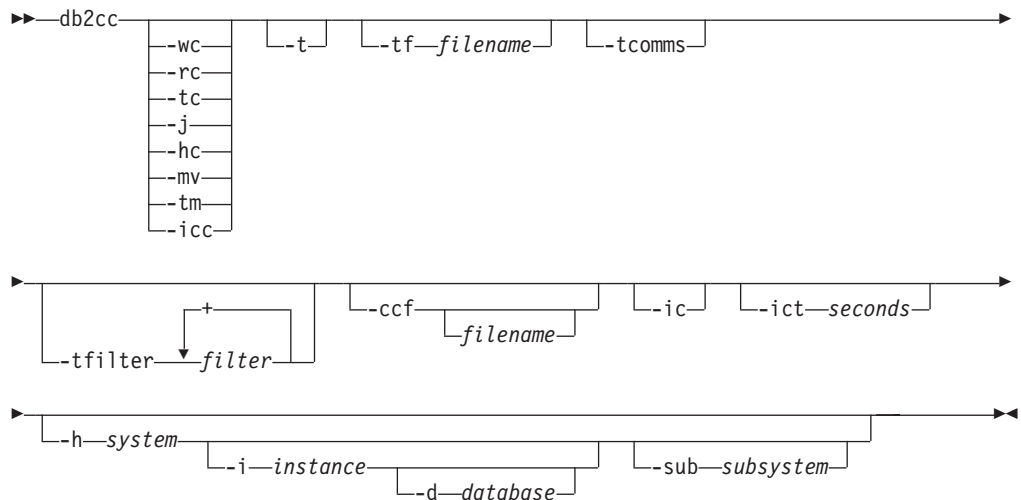
## db2cc - コントロール・センターの開始

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。

権限:

*sysadm*

コマンド構文:



コマンド・パラメーター:

- wc** データウェアハウス・センターをオープンします。
- 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。
- 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。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 ベースのプラットフォームでは `/home/<userid>/sqllib/tools` に保管されます。
- tcomms** トレースを通信イベントのみを対象とするように制限します。

**-tfilter filter**

トレースを、指定したフィルター (1 つ以上) を含む項目のみを対象とするように制限します。

**-ccf filename**

コマンド・エディターをオープンします。ファイル名が指定された場合は、このファイルの内容がコマンド・エディターの「スクリプト (Script)」ページにロードされます。ファイル名を指定するときは、ファイルの絶対パスを指定する必要があります。ご注意ください。

**-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:**

- 407 ページの『GET ADMIN CONFIGURATION』
- 685 ページの『RESET ADMIN CONFIGURATION』
- 773 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。後でそのプロファイルは、類似のインスタンス・タイプの別の DB2 Universal Database (UDB) ワークステーション・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 UDB 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 構成プロファイルといいます。

このユーティリティは、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です。エクスポートできる項目は次のとおりです。

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

このユーティリティは、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 UDB 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 (たとえば、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)。

### 権限:

以下のいずれかが必要です。

- *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 UDB 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 UDB 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 UDB 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。

## 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
- Universal Database (UDB) 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、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 UDB 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 (たとえば、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 UDB 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。

### 権限:

以下のいずれかが必要です。

- *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、DB2 UDB または DB2 Connect の接続構成エクスポート方式で作成されたプロファイル、またはサーバー・アクセス・プロファイルです。

## db2cidmg - リモート・データベース移行

構成、インストール、および配布 (CID) アーキテクチャー環境で、リモート操作による自動移行をサポートします。

### 権限:

以下のいずれかが必要です。

- *sysadm*
- *dbadm*

### コマンド構文:

```

▶▶ db2cidmg database [ /r=respfile ] [ /l1=logfile ] [ /b ]

```

### コマンド・パラメーター:

#### database

移行するデータベースの別名を指定します。この別名が指定されていない場合、プログラムを呼び出すためには、応答ファイルまたは */e* が指定されていなければなりません。データベースの別名はターゲット・ワークステーション上でカタログ作成しておかなければならないので、注意してください。ただし、別名はローカル・データベースのものでリモート・データベースのもので構いません。

**/r** CID 移行に使用する応答ファイルを指定します。応答ファイルとは、移行するデータベースのリストが入っている ASCII ファイルのことです。これが指定されていない場合、プログラムを呼び出すためには、データベース別名または */e* が指定されていなければなりません。

**/e** システム・データベース・ディレクトリーでカタログ作成されているそれぞれのデータベースをすべて移行します。 */e* を指定しない場合は、データベース別名か応答ファイルを指定しなければなりません。

**/l1** 移行作業が完了したら、リモート・ワークステーションからのエラー・ログ情報をコピーすることが可能なファイルのパス名を指定します。応答ファイルに複数のデータベースを指定する場合、各データベースの移行ログ情報は、ファイルの最後に追加されます。 */l1* が指定されているかどうかにかかわらず、 **DB2CIDMG.LOG** という名前のログ・ファイルが生成され、データベースの移行を実行したワークステーションのファイル・システムに保持されます。

**/b** データベースにあるパッケージはすべて、移行が完了すると再バインドされます。



## 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。

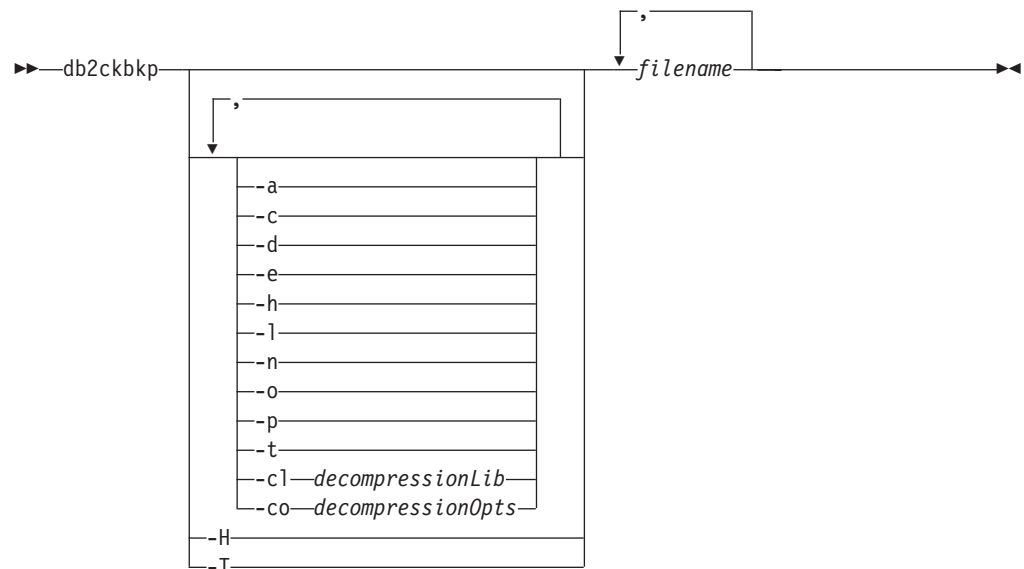
### 権限:

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** 使用可能な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** *deprecationLib*

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** *deprecationOpts*

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 <tbodyID> <objID> <objType> <startPage> <numPages>
```

DMS 表スペースの場合、

```
D <tbodyID> <objType> <startPage> <numPages>
```

**注:** <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。

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。

**注:** 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。

**-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。

I

注: 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 (Windows プラットフォーム)

```
db2ckbkp SAMPLE.0¥krodger¥NODE0000¥CATN0000¥19990817¥150714.001
SAMPLE.0¥krodger¥NODE0000¥CATN0000¥19990817¥150714.002
SAMPLE.0¥krodger¥NODE0000¥CATN0000¥19990817¥150714.003
```

```
[1] Buffers processed: ##
[2] Buffers processed: ##
[3] Buffers processed: ##
Image Verification Complete - successful.
```

#### 例 3

```
db2ckbkp -h SAMPLE2.0.krodger.NODE0000.CATN0000.19990818122909.001
```

```
=====
MEDIA HEADER REACHED:
=====
```

```
Server Database Name      -- SAMPLE2
Server Database Alias     -- SAMPLE2
Client Database Alias     -- SAMPLE2
Timestamp                 -- 19990818122909
Database Partition Number -- 0
Instance                  -- krodger
Sequence Number           -- 1
Release ID                 -- 900
Database Seed              -- 65E0B395
DB Comment's Codepage (Volume) -- 0
DB Comment (Volume)       --
DB Comment's Codepage (System) -- 0
DB Comment (System)       --
Authentication Value      -- 255
Backup Mode                -- 0
Include Logs               -- 0
Compression                -- 0
Backup Type                -- 0
Backup Gran.               -- 0
```

## db2ckbkp - バックアップの検査

```
Status Flags                -- 11
System Cats inc              -- 1
Catalog Database Partition No. -- 0
DB Codeset                   -- IS08859-1
DB Territory                 --
LogID                        -- 1074717952
LogPath                      -- /home/krodger/krodger/NODE0000/
                             SQL00001/SQL0GDIR
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。たとえば、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:

- 10 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。

### 権限:

*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:

UNIX ベースのプラットフォームでは、**db2imigr** 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 UDB のインストール後、インスタンスを移行する前に、各インスタンスに対してそのツールを実行しなければなりません。このコマンドはインストール・パスから実行します。このツールは、DB2DIR/bin にあります。DB2DIR は、AIX 上では /usr/opt/db2\_08\_01、他のすべての UNIX ベース・システム上では /opt/IBM/db2/V8.1 です。

Windows プラットフォームでは、インストール中に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。このツールは、DB2 UDB CD の db2/Windows/Utilities にあります。

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。

## db2ckmig - データベース事前移行ツール

3. ログ・ファイルをチェックする。

注: 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。

### 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『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。

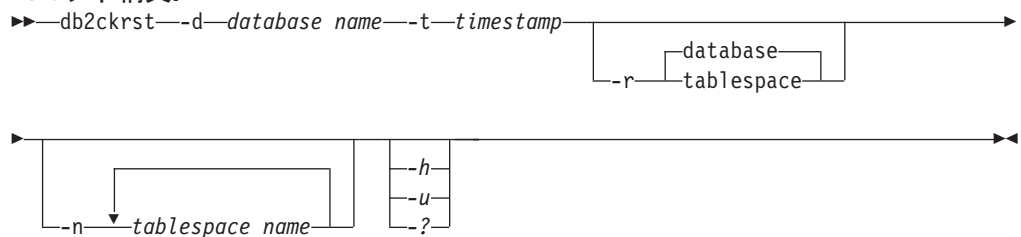
権限:

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。

**注:** 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。

**注:** データベース・リストア・タイプを選択して、表スペース名のリストを指定した場合、ユーティリティーは指定の表スペース名を使用して表スペース・リストアを続行します。

**-h/-u/-?**

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

例:

```

db2ckrst -d mr -t 20001015193455 -r database
db2ckrst -d mr -t 20001015193455 -r tablespace
db2ckrst -d mr -t 20001015193455 -r tablespace -n tbsp1 tbsp2

> db2 backup db mr

```

## db2ckrst - 増分リストア・イメージ順序の検査

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

> db2 backup db mr incremental

Backup successful. The timestamp for this backup image is : 20001016001445

> db2ckrst -d mr -t 20001016001445

Suggested restore order of images using timestamp 20001016001445 for
database mr.
=====
db2 restore db mr incremental taken at 20001016001445
db2 restore db mr incremental taken at 20001016001426
db2 restore db mr incremental taken at 20001016001445
=====

> db2ckrst -d mr -t 20001016001445 -r tablespace -n userspace1
Suggested restore order of images using timestamp 20001016001445 for
database mr.
=====
db2 restore db mr tablespace ( USERSPACE1 ) incremental taken at
20001016001445
db2 restore db mr tablespace ( USERSPACE1 ) incremental taken at
20001016001426
db2 restore db mr tablespace ( USERSPACE1 ) incremental taken at
20001016001445
=====
```

### 使用上の注意:

このユーティリティ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があります。**PRUNE HISTORY** コマンドのデフォルトの操作では、必要な項目を削除しないようになっています。**PRUNE HISTORY** コマンドの **FORCE** オプションは使用しないことをお勧めします。

このユーティリティは、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:

- 697 ページの『**RESTORE DATABASE**』
- 630 ページの『**PRUNE HISTORY/LOGFILE**』



---

## 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。

**権限:**

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。

## 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 でのみ使用できます。

### 権限:

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** コマンドを実行してから終了します。たとえば、"`db2cmd /c dir`" というコマンドを使用すると、"`dir`" コマンドがコマンド・ウィンドウに呼び出されてから、コマンド・ウィンドウがクローズします。
- w** `cmd.exe` プロセスが終了するまで待機します。たとえば、"`db2cmd /c /w dir`" というコマンドを使用すると、"`dir`" コマンドが呼び出され、コマンド・ウィンドウがクローズするまで、`db2cmd.exe` は終了しません。
- i** 同じコンソールを共用し、ファイル・ハンドルを継承して、コマンド・ウィンドウを実行します。たとえば、"`db2cmd /c /w /i db2 get dbm cfg > myoutput`" を使うと、`cmd.exe` が起動して DB2 コマンドを実行し、完了を待機します。新しいコンソールは割り当てられず、`stdout` がファイル "`myoutput`" にパイプ接続されます。
- t** "DB2 CLP" をコマンド・ウィンドウのタイトルに使用する代わりに、起動ウィンドウからタイトルを継承します。これは、たとえば "`db2cmd /t`" を起動する異なるタイトルでアイコンを設定したい場合などに役立ちます。

**注:** すべてのスイッチは、コマンドが実行される前に表示されるはずです。たとえば、`db2cmd /t db2` などです。

### 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 で使用可能になった DB2 ウィンドウを表示する際に戻される場合、または Windows 98 で CLP コマンドを実行する場合、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:

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

## 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。

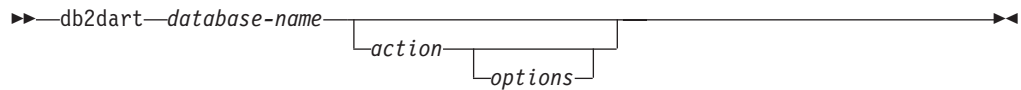
権限:

*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。必須入力値は、DMS 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。

**/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 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。

### **/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 および表オブジェクト 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。

注: 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。

**/NP** *number*

ページ数を指定します。

**/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** 冗長出力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。

### 使用上の注意:

**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。

**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。

/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。

### 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』

## 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。

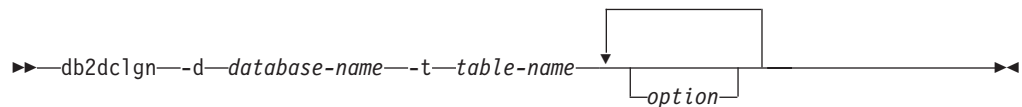
権限:

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**

以下のいずれかです (複数の場合もある)。

**-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)**

たとえば、C では SQL TYPE は CLOB(5K) x です。

**LOCATOR**

たとえば、C では SQL TYPE は CLOB\_LOCATOR x です。

**FILE** たとえば、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

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

列の注釈が使用可能である場合、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 (たとえば接続状況) 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

sqlwchar または wchar\_t が、C で GRAPHIC/VARGRAPHIC/DBCLOB 列に使用されるかどうかを指定します。



**-y DBCS-symbol**

G または N が、COBOL で DBCS シンボルとして使用されるかどうかを指定します。

**-z encoding**

encoding に、特定のサーバーに合わせたコーディング規則を指定します。 encoding は UDB か OS390 のいずれかにできます。 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。

## db2demigdbd - データベース・ディレクトリー・ファイルの逆マイグレーション

DB2 UDB バージョン 8.2 データベース・ディレクトリー・ファイルをバージョン 8.1 データベース・ディレクトリー・ファイル形式に変換します。

DB2 UDB バージョン 8.2 では、データベース項目構造に 2 つの新しいフィールドが追加され、データベース・ディレクトリー・ファイル構造が変更されました。バージョン 8.1 からバージョン 8.2 へのマイグレーションでは、データベース・ディレクトリー・ファイルは自動的にマイグレーションされます。しかし、バージョン 8.2 からバージョン 8.1 にデータベース・ディレクトリー・ファイルを逆マイグレーションするには、現在のデータベース・ディレクトリー・ファイルをバージョン 8.1 形式に変換するために **db2demigdbd** が必要です。

### 権限:

UNIX ベース・プラットフォームでは、以下のいずれかが必要です。

- インスタンス所有者
- 1 次グループのメンバー

Windows オペレーティング・システムでは、以下が必要です。

- DB2 管理者グループのメンバー

### 必要な接続:

なし

### コマンド構文:

```
db2demigdbd path [1] [2]
```

### コマンド・パラメーター:

*path* システムまたはローカル・データベース・ディレクトリーのパス名を指定します。

システム・データベース・ディレクトリーだけを逆マイグレーションする場合、パスは UNIX ベース・システムの場合は \$HOME/sqllib、Windows オペレーティング・システムの場合は \$DB2PATH¥instance です。

ローカル・データベース・ディレクトリーだけを逆マイグレーションする場合、パスは UNIX ベース・システムの場合は \$DBPATH/NODExxxx、Windows オペレーティング・システムの場合は \$HOME¥\$INSTANCE¥NODExxxx です。

システム・データベース・ディレクトリーとローカル・データベース・ディレクトリーの両方を逆マイグレーションする場合、パスは UNIX ベース・システムの場合は \$HOME/sqllib、Windows オペレーティング・システムの場合は \$DB2PATH¥instance です。

- 1 *path* にあるデータベース・ディレクトリー・ファイルを逆マイグレーションすることを指定します。このオプションを使用すると、ローカル・データベース・ディレクトリーのパスがシステム・データベース・ディレクトリ

## db2demigdbd - データベース・ディレクトリー・ファイルの逆マイグレーション

ー・ファイルのデータベース項目に存在しない場合、そのローカル・データベース・ディレクトリーを逆マイグレーションします。

- 2 *path* にあるインスタンスのシステム・データベース・ディレクトリー・ファイルとすべてのローカル・データベース・ディレクトリー・ファイルを逆マイグレーションすることを指定します。

### 例:

AIX 上のシステム・データベース・ディレクトリー・ファイルを逆マイグレーションするには、以下のようにします。

```
db2demigdbd $HOME/sql1lib 1
```

Windows プラットフォーム上のシステム・データベース・ディレクトリー・ファイルを逆マイグレーションするには、以下のようにします。

```
db2demigdbd d:%sql1lib%db2 1
```

ここで、db2 は現行インスタンスです。

AIX 上のローカル・データベース・ディレクトリー・ファイルを逆マイグレーションするには、以下のようにします。

```
db2demigdbd ~/user/NODE0000 1
```

Windows プラットフォーム上のローカル・データベース・ディレクトリー・ファイルを逆マイグレーションするには、以下のようにします。

```
db2demigdbd d:%db2%NODE0000 1
```

AIX 上のインスタンスにあるシステム・データベース・ディレクトリー・ファイルとすべてのローカル・データベース・ディレクトリー・ファイルを逆マイグレーションするには、以下のようにします。

```
db2demigdbd $HOME/sql1lib 2
```

Windows 上のインスタンスにあるシステム・データベース・ディレクトリー・ファイルとすべてのローカル・データベース・ディレクトリー・ファイルを逆マイグレーションするには、以下のようにします。

```
db2demigdbd d:%sql1lib%db2 2
```

### 使用上の注意:

このユーティリティーは、バージョン 8.2 から 8.1 に戻す前、または戻した後のどちらでも実行できます。

バージョン 8.2 からバージョン 8.1 に戻して、db2demigdbd ユーティリティーでデータベース・ディレクトリー・ファイルを逆マイグレーションしないと、データベースにアクセスするときに SQL10004 を受け取ります。

## 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。

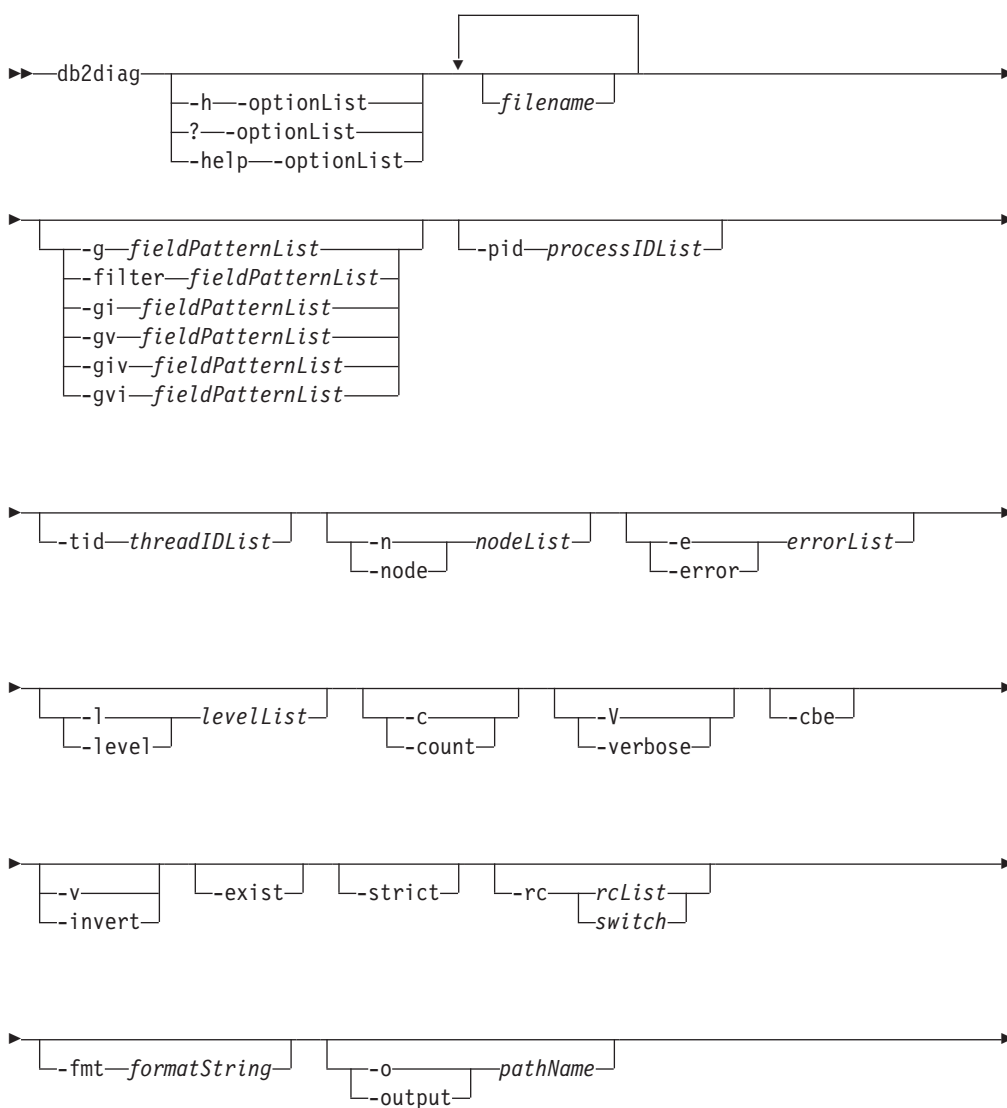
権限:

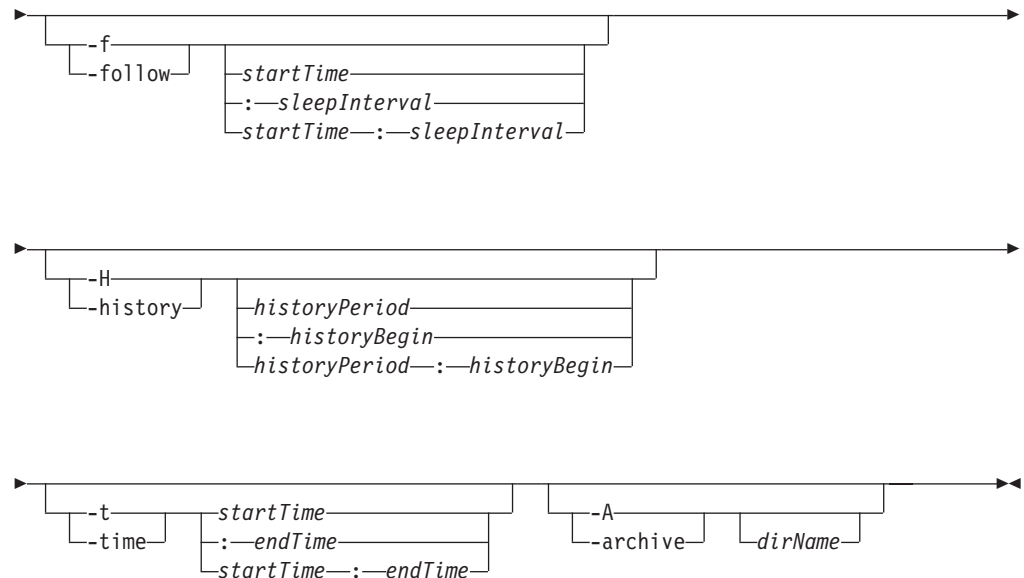
なし。

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、使用可能な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* 引き数の位置で以下のスイッチの 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 ベース・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)。たとえば、東部標準時の場合は -300 です。

#### **%recordid/%recid**

ユニークなレコード 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、 Critical、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。たとえば、 "db2sysc.exe" とします。

**%product**

製品名。たとえば、DB2 UDB または 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**

引き数の説明。

**%argobject**

引き数オブジェクト。

**%arg** エラーを戻した関数呼び出しの引き数。%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 (たとえば、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、データおよび引き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≤*n*≤64)。たとえば、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、引き数フィールド、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じ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、その後使用可能となると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 ベースの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 ベースのプラットフォーム) またはミリ秒 (Windows オペレーティング・システム) を指定します。

#### **-t/-time**

タイム・スタンプの値を指定します。このオプションには、以下のオプションの 1 つまたは両方を指定できます。

##### *startTime*

*startTime* 以降にログに記録されたすべてのメッセージを表示します。

##### *:endTime*

*endTime* の前にログに記録されたすべてのメッセージを表示します。

*startTime* と *endTime* の間にログに記録されたメッセージを表示するには、**-t *startTime:endTime*** を指定します。

フォーマットは、*YYYY-MM-DD-hh.mm.ss.nnnnnn* です。

*YYYY* 年を指定します。

*MM* 月を指定します (01 から 12)。

*DD* 日を指定します (01 から 31)。

*hh* 時間を指定します (00 から 23)。

*mm* 分を指定します (00 から 59)。

*ss* 秒を指定します (00 から 59)。

*nnnnnn* マイクロ秒 (UNIX ベースのプラットフォーム) 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、ファイル名にディレクトリー名は含められません。

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。

#### **-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。

#### **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
```

## 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。たとえば、`-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、`-` を使用する必要はないことに注意してください。
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、入力は指定のファイルからとられます。それに対して、`-` オプションが指定されている場合、ファイル名は無視され (コマンド行に指定されている場合でも)、端末からの入力を使用されます。パイプが使用され、ファイル名が指定されない場合、**db2diag** の入力は、コマンド行に `-` が指定されるかどうかに関係なく、同じ方法で処理されます。
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. 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。



## db2d1m\_upd\_hostname - データ・リンク・ホスト名の更新

Data Links Manager 環境で、DB2 サーバーまたは Data Links File Manager サーバーのホスト名の変更を登録します。DB2 サーバーまたは DLFM サーバーのいずれかを変更した場合 (たとえば、サーバーのホスト名を変更した場合や、HACMP フェイルオーバーによってホスト名が変更された場合) は、ホスト名の変更を登録する必要があります。DB2 サーバーのホスト名の変更があった場合に、DLFM サーバーでこのユーティリティーを実行すると、DB2 サーバーの新しいホスト名で DLFM\_DB データベースが更新されます。DLFM サーバーのホスト名の変更があった場合に、DB2 サーバーでこのユーティリティーを実行すると、DB2 サーバーの各データベース・ディレクトリーにある datalink.cfg ファイルが更新されます。

UNIX システムでは、**db2d1m\_upd\_hostname** は INSTHOME/sql/lib/bin 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。Windows システムでは、**db2d1m\_upd\_hostname** は x:\sql\lib\bin ディレクトリーにあります。x: は DB2 Data Links Manager がインストールされているドライブです。

### 権限:

DLFM サーバーを更新する場合:

- DLFM\_DB (DLFM\_DB を作成したユーザー ID) の *sysadm*

DB2 サーバーを更新する場合:

- *sysadm*

DLFM サーバーの場合は、DLFM インスタンスにアタッチしている状態で **db2d1m\_upd\_hostname** を実行する必要があります。DB2 サーバーの場合は、ホスト・データベースが存在するインスタンスにアタッチしている状態でこのユーティリティーを実行する必要があります。

### コマンド構文:

```
db2d1m_upd_hostname server dlfm oldhost name newhost name dbname name dbinst instance skipbkup yes
server db2 oldhost name newhost name newport port
```

### コマンド・パラメーター:

#### DLFM サーバーの場合:

**-server** *dlfm*

このユーティリティーを DLFM サーバーで実行することを指定します。

**-oldhost** *name*

DB2 サーバーの古いホスト名を指定します。

**-newhost** *name*

DB2 サーバーの新しいホスト名を指定します。

**-dbname** *name*

ホスト名を変更する必要があるデータベース名の項目 (DLFM\_DB データベースの dfm\_dbid 表の項目) を指定します。必ず -dbinst オプションと一緒に

## db2dlm\_upd\_hostname - データ・リンク・ホスト名の更新

に使用します。このオプションと `-dbinst` オプションを一緒に使用しない場合は、`dfm_dbid` 表内の該当するすべてのホスト名項目が更新されます。

### **-dbinst** *instance*

ホスト名を変更する必要があるデータベース・インスタンスの項目 (DLFM\_DB データベースの `dfm_dbid` 表の項目) を指定します。このオプションは、必ず `-dbname` オプションと一緒に指定します。

### **-skipbkup** *yes*

DLFM\_DB をバックアップしないことを指定します。このオプションを指定しない場合、DLFM\_DB データベースはバックアップされます。このオプションを指定しないようにお勧めします。

### **DB2 サーバーの場合:**

#### **-server** *db2*

このユーティリティを DB2 サーバーで実行することを指定します。

#### **-oldhost** *name*

DLFM サーバーの古いホスト名を指定します。

#### **-newhost** *name*

DLFM サーバーの新しいホスト名を指定します。

#### **-newport** *port*

DB2 と DLFM の間の通信に使用するポート番号を指定します。

### **例:**

#### **DB2 サーバーの場合:**

```
db2dlm_upd_hostname -server dlfm -oldhost dlfsc1.in.ibm.com
                    -newhost dlfstst.in.ibm.com

db2dlm_upd_hostname -server dlfm -oldhost dlfssrv.in.ibm.com
                    -newhost dlfstst.in.ibm.com -dbname tstdb -dbinst regress

db2dlm_upd_hostname -server dlfm -oldhost dlfsc1.in.ibm.com
                    -newhost dlfstst.in.ibm.com -skipbkup yes
```

#### **DLFM サーバーの場合:**

```
db2dlm_upd_hostname-server db2 -oldhost dlfsv.in.ibm.com
                    -newhost dlfsc1n.in.ibm.com

db2dlm_upd_hostname-server db2 -oldhost dlfsv.in.ibm.com
                    -newhost dlfsc1n.in.ibm.com -newport 5000
```



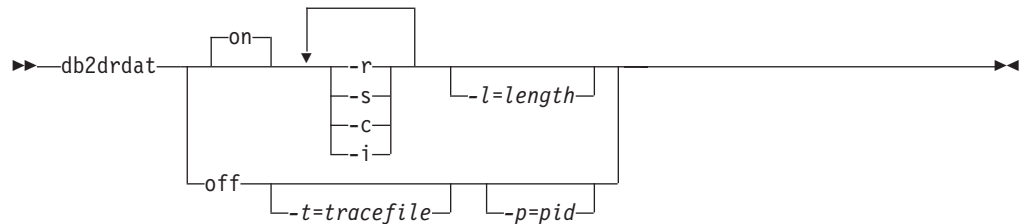
## db2drdat - DRDA トレース

DB2 UDB 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できます。このツールは、アプリケーションの実行に必要な送信および受信の回数を判別することによって、問題判別でよく使用されますが、クライアント/サーバー環境でのパフォーマンス調整にも使用することができます。

### 権限:

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。

注: 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。

注: *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。

### 使用上の注意:

**db2drdat** がアクティブな場合、**db2trc** コマンドを発行しないでください。

**db2drdat** は、以下の情報を *tracefile* に書き込みます。

1. -r
  - DRDA 要求のタイプ
  - 受信バッファー
2. -s
  - DRDA 応答/オブジェクトのタイプ
  - 送信バッファー
3. CPI-C エラー情報
  - 重大度
  - 使用したプロトコル
  - 使用した API
  - ローカル LU 名
  - 失敗した CPI-C 機能
  - CPI-C 戻りコード

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。

**注:** **db2drdat** がすで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:

- 517 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可能です。

### 権限:

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・アタッチが存在しない場合は、アプリケーションによって作成されます。

### コマンド構文:

```

▶ db2drvmp {
  add      dbpartition_number—from_drive—to_drive
  drop
  query
  reconcile
}
  
```

### コマンド・パラメーター:

**add** 新しいデータベース・ドライブ・マップを割り当てます。

**drop** 既存のデータベース・ドライブ・マップを削除します。

**query** データベース・マップを照会します。

### reconcile

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

### dbpartition\_number

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

### from\_drive

マップ元のドライブ・レター。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・レター。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。

## 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・ファイル割り振りを使用可能にします。SMS 表スペースでの複数ページ・ファイル割り振りを使用可能にすると、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。

### 権限:

*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使用可能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使用可能にできます。

### 関連概念:

- 「管理ガイド: パフォーマンス」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イベント・データをトレースできるようにします。

### 権限:

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:

注: *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。

**注:** 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。

78 コマンド・リファレンス



## 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。

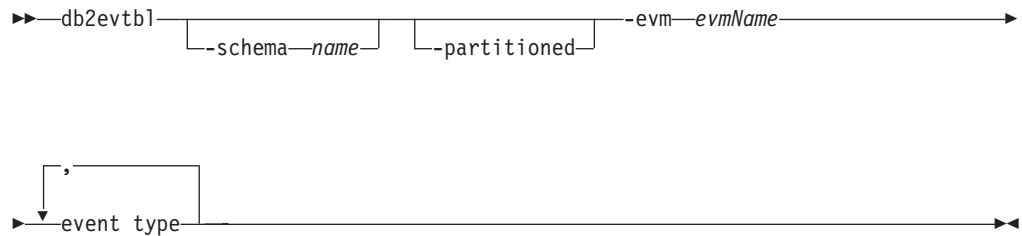
権限:

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用可能なイベント・タイプのいずれか。たとえば 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。たとえば、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:

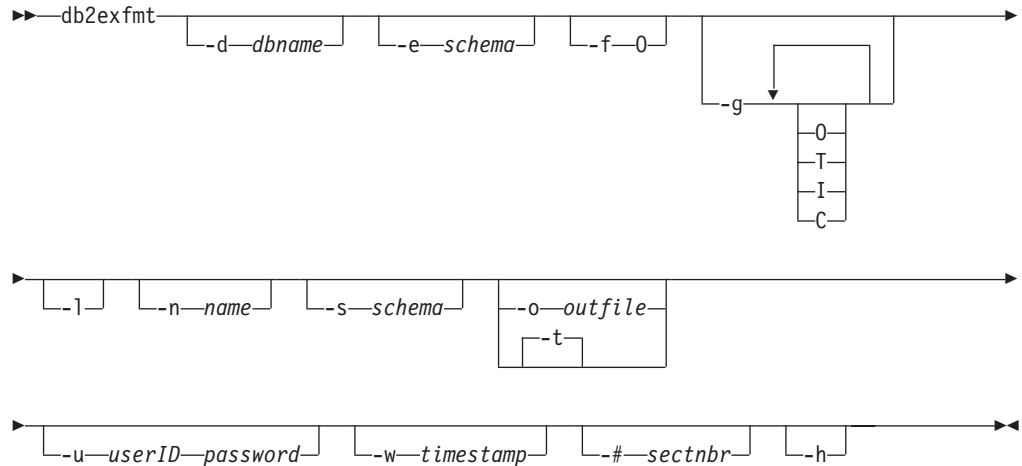
- 「SQL リファレンス 第 2 巻」の『CREATE EVENT MONITOR ステートメント』
- 「SQL リファレンス 第 2 巻」の『SET EVENT MONITOR STATE ステートメント』

## 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・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。

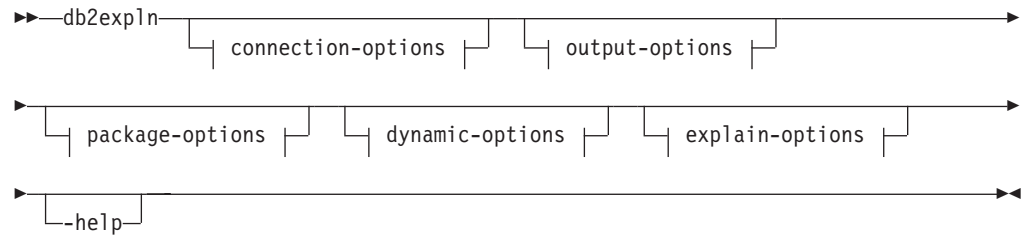
-o をファイル名なしで指定し、-t を指定しない場合は、ファイル名の入力を促すプロンプトが表示されます (デフォルト名は db2exfmt.out)。-o も -t も指定しない場合は、ファイル名の入力を促すプロンプトが表示されます (デフォルト・オプションは端末出力です)。-o と -t の両方を指定した場合は、出力が端末に送信されます。

### 関連概念:

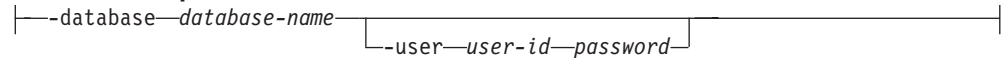
- ・ 「管理ガイド: パフォーマンス」の『Explain ツール』
- ・ 「管理ガイド: パフォーマンス」の『Explain 情報の使用のガイドライン』
- ・ 「管理ガイド: パフォーマンス」の『Explain 情報のキャプチャーのガイドライン』

## db2expln - SQL Explain

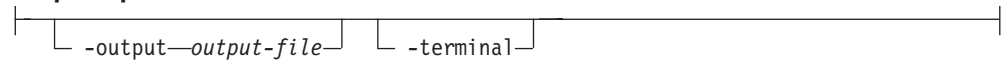
### コマンド構文:



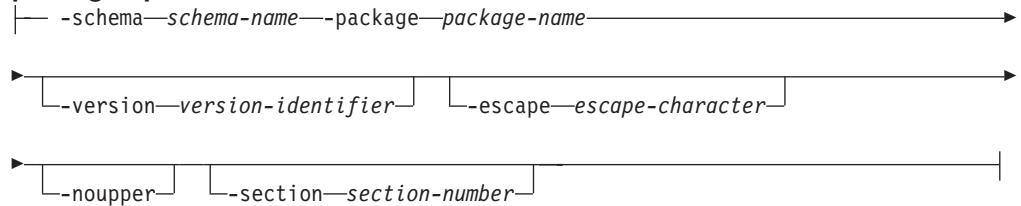
### connection-options:



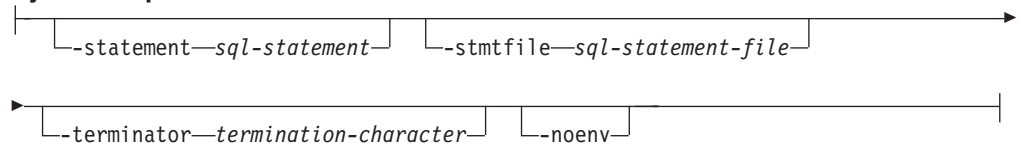
### output-options:



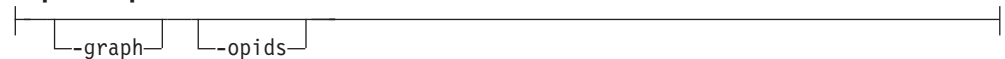
### package-options:



### 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。

### **-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 SQL だけが Explain の対象になります。

**注:** 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 の対象パッケージ (複数可) のスキーマ。

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*。

たとえば、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。

**注:** 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 の対象として 1 つ以上の動的 SQL ステートメントを指定します。

### **-statement** *sql-statement*

Explain の対象として動的に生成する SQL ステートメント。複数のステートメントを Explain するには、**-stmtfile** オプションを使用して、Explain の対象にする SQL ステートメントを含むファイルを指定するか、

**-terminator** オプションを使用して、**-statement** オプション内でステートメントを区切るための終止符を定義します。

dynexpln との互換性のために、**-statement** の代わりに **-q** を使用できません。

### **-stmtfile** *sql-statement-file*

Explain の対象として動的に生成する 1 つ以上の SQL 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 SQL 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、SQL ステートメントの終了を示す文字を指定してください。

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 SQL ステートメントの終了を示す文字。デフォルトでは、**-statement** オプションの値は 1 つの SQL 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 SQL 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 SQL 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 SQL 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 SQL 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。ただし、生成されるグラフは、Visual Explain のグラフと全く一致するとは限らないことに注意してください。

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 の出力には現れないことに注意してください。

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。使用可能な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。たとえば、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。

#### 使用上の注意:

**-help** オプションを指定する場合以外は、**package-options** または **dynamic-options** のいずれかを指定しなければなりません。パッケージと動的 SQL の両方を db2expln の 1 つの呼び出しで Explain できます。

上記のオプション・フラグのいくつかは、オペレーティング・システムに対して特別な意味を持つことがあるので、db2expln コマンド行の値が正確に解釈されないことがあります。しかし、オペレーティング・システムのエスケープ文字を前に置けば、その種の文字を入力できます。詳細については、オペレーティング・システムの資料を参照してください。オペレーティング・システムのエスケープ文字を db2expln のエスケープ文字として間違えて指定しないように注意する必要があります。

db2expln によって生成されるヘルプと初期状況メッセージは、標準出力に書き込まれます。Explain ツールによって生成されるすべてのプロンプトと他の状況メッセージは、標準エラーに書き込まれます。Explain テキストは、選択した出力オプションに応じて、標準出力またはファイルに書き込まれます。

#### 例:

db2expln の 1 回の呼び出しで複数のプランを Explain する場合は、**-package**、**-schema**、**-version** の各オプションを使用し、LIKE パターンを使用してパッケージと作成者に関するストリング定数を指定します。つまり、下線 ( \_ ) を使用して 1 つの文字を表し、パーセント記号 ( % ) を使用してゼロ個以上の文字を表します。

SAMPLE という名前のデータベース内のすべてのパッケージのすべてのセクションを Explain して、その結果を **my.exp** ファイルに書き込むには、以下のようにします。

```
db2expln -database SAMPLE -schema % -package % -output my.exp
```

## db2expln - SQL Explain

別の例として、ユーザーが "statements.db2" という名の CLP スクリプト・ファイルを持っていて、そのファイル内のステートメントを Explain するとします。ファイルには、以下のステートメントが含まれています。

```
SET PATH=SYSIBM, SYSFUN, DEPT01, DEPT93@  
SELECT EMPNO, TITLE(JOBID) FROM EMPLOYEE@
```

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

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

### 関連概念:

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

## db2flsn - ログ・シーケンス番号の検出

指定されたログ・シーケンス番号 (LSN) で識別されるログ・レコードを含むファイルの名前を戻します。

**権限:**

なし

**コマンド構文:**

```
db2flsn -q input_LSN
```

**コマンド・パラメーター:**

**-q** ログ・ファイル名だけが印刷されます。エラー・メッセージや警告メッセージは印刷されず、状況は戻りコードを介してのみ判別できます。有効なエラー・コードは以下のとおりです。

- -100 無効な入力
- -101 LFH ファイルをオープンできない
- -102 LFH ファイルの読み取りに失敗した
- -103 無効な LFH
- -104 データベースがリカバリー可能でない
- -105 LSN が大きすぎる
- -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

**input\_LSN**

ストリング付きの内部 (6 バイト) 16 進数値を表す 12 文字のストリング。

**例:**

```
db2flsn 000000BF0030
Given LSN is contained in log file S0000002.LOG

db2flsn -q 000000BF0030
S0000002.LOG

db2flsn 000000BE0030
Warning: the result is based on the last known log file size.
The last known log file size is 23 4K pages starting from log extent 2.

Given LSN is contained in log file S0000001.LOG

db2flsn -q 000000BE0030
S0000001.LOG
```

**使用上の注意:**

ログ・ヘッダー制御ファイル SQLLOGCTL.LFH が現行ディレクトリになければなりません。このファイルはデータベース・ディレクトリにあるので、データベー

## db2flsn - ログ・シーケンス番号の検出

ス・ディレクトリーからこのツールを実行するか、このツールが実行されるディレクトリーに制御ファイルをコピーすることができます。

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 *logretain* または *userexit* が ON の値に設定されている場合です。

## db2fm - DB2 障害モニター

DB2 障害モニター・デーモンを制御します。 **db2fm** を使用すると、障害モニターを構成できます。

このコマンドは、UNIX ベース・プラットフォームでのみ利用できます。

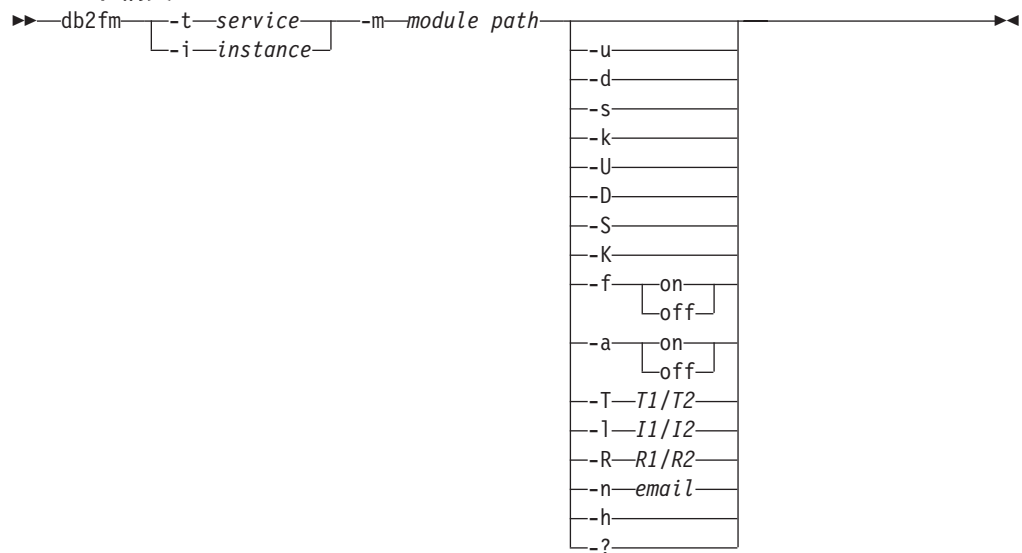
### 権限:

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/sql/lib/libdb2gcf です。

#### **-t** *service*

ユニークな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** サービスを停止します。

**-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。

注: 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))
- 使用可能です (AVAILABLE)
- 不明 (UNKNOWN)

**-f onloff**

障害モニターをオン/オフにします。

注: 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 onloff**

障害モニターを活動化または非活動化します。

注: 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、再び始動され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。

たとえば、

- -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。

---

## 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 ベースのシステムでは、db2fs は sqllib/bin ディレクトリーにあります。  
Windows オペレーティング・システムでは、db2fs.bat は \$DB2PATH¥bin ディレクトリーにあります。

権限:

*sysadm*

コマンド構文:

▶▶—db2fs—▶▶

コマンド・パラメーター:

なし

## db2gcf - DB2 インスタンスの制御

HA (高可用性) クラスタなどの中にある DB2 インスタンスの開始と停止とモニターを (通常は自動スクリプトによって) 実行します。

UNIX ベース・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。

### 権限:

以下のいずれかが必要です。

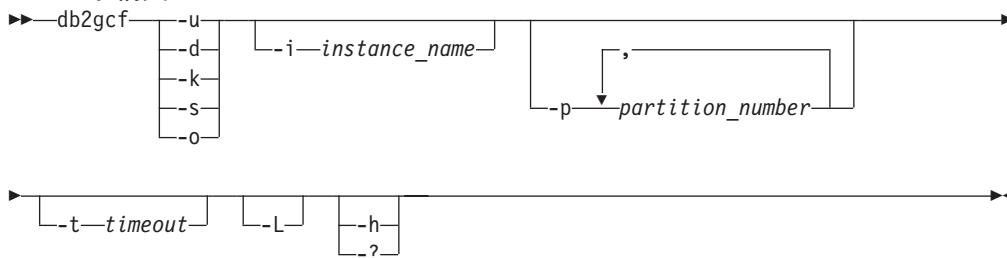
- `sysadm`
- `sysctrl`
- `sysmaint`

さらに、`db2_all` コマンドを実行できる必要があります。

### 必要な接続:

なし

### コマンド構文:



### コマンド・パラメーター:

- u** 現在のデータベース・パーティション・サーバー (ノード) の指定のインスタンスの指定の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パーティションは、プロセスの対象として使用可能な状態です。
  - *Operable*: このインスタンスはインストールされていますが、現在使用可能な状態ではありません。
  - *Not operable*: このインスタンスは、使用可能な状態にできません。
- 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・ロギングを使用可能にします。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
```

## 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。

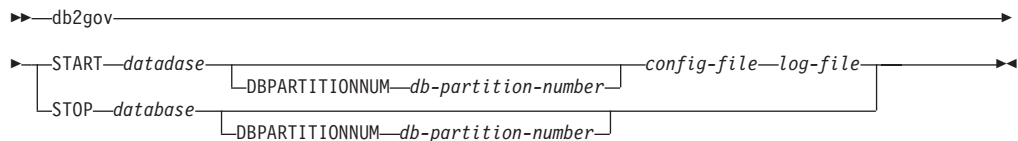
### 権限:

以下のいずれかが必要です。

- *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。

**注:** 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完全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。たとえば、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。

---

## 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。

権限:

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:

- 96 ページの『db2gov - DB2 管理プログラム』

## db2gpmmap - パーティション・マップの取得

データベースがすで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・グループのパーティション・マップを取得します。

### 権限:

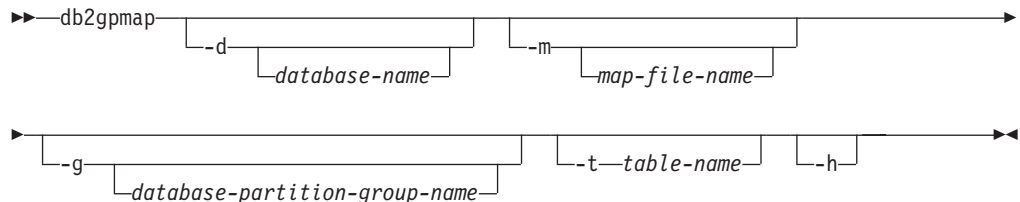
以下の両方が必要です。

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

### コマンド構文:



### コマンド・パラメーター:

- d** パーティション・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** パーティション・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** パーティション・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 表のパーティション・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『パーティション・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。

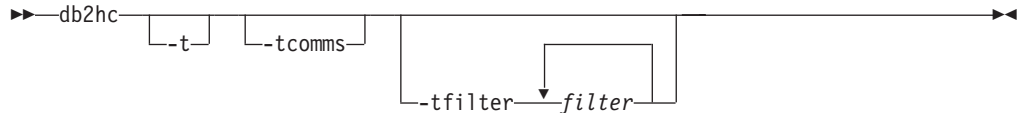
### 権限:

情報の表示には特別な権限は必要ありません。処置を取るには適切な権限が必要です。

### 必要な接続:

インスタンス

### コマンド構文:



### コマンド・パラメーター:

**-t** 初期化コードに対する NavTrace をオンにします。このオプションを使用するのは、DB2 サポートにそうするよう指示された場合のみです。

#### **-tcomms**

トレースを通信イベントのみを対象とするように制限します。このオプションを使用するのは、DB2 サポートにそうするよう指示された場合のみです。

#### **-tfilter** *filter*

トレースを、指定したフィルター (1 つ以上) を含む項目のみを対象とするように制限します。このオプションを使用するのは、DB2 サポートにそうするよう指示された場合のみです。

## db2iauto - インスタンスの自動開始

毎回のシステム再始動後のインスタンスの自動開始を使用可能にするか、使用不可にします。このコマンドは、UNIX ベースのシステムのみで使用可能です。

### 権限:

以下のいずれかが必要です。

- 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使用可能にします。
- off** 指定したインスタンスの自動開始を使用不可にします。

### instance-name

インスタンスのログイン名。

## db2iclus - Microsoft Cluster Server

ユーザーが Microsoft Cluster Server (MSCS) 環境で、インスタンスおよび DB2 Administration Server (DAS) を追加、ドロップ、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可能です。

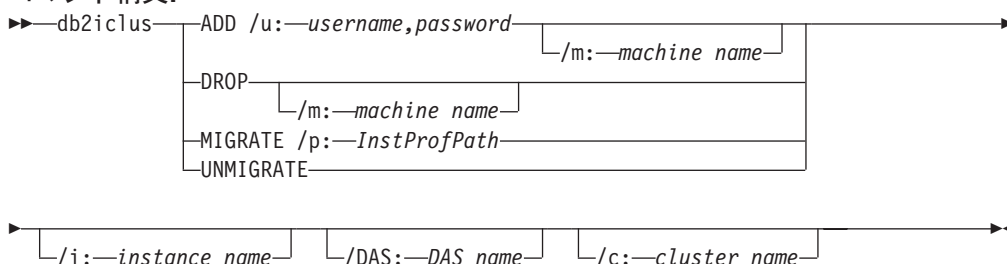
### 権限:

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 UDB 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、すで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.
```

**注:** 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、すで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。

## db2icons - DB2 アイコンの追加

Linux デスクトップに DB2 のアイコンとフォルダーを追加します。このコマンドは、サポートされている Intel ベース Linux ディストリビューションの Gnome と KDE のデスクトップについてのみ使用できます。このコマンドは DB2DIR/bin ディレクトリーにあります。DB2DIR は /opt/IBM/db2/V8.1 です。また、インスタンス所有者の home ディレクトリーの /sqlllib/bin ディレクトリーにもあります。

### 権限:

以下のいずれかが必要です。

- 他のユーザーのためにこのコマンドを呼び出す場合: root 権限、または指定のユーザーの home ディレクトリーに対する書き込み権限
- 自分のデスクトップのためにこのコマンドを呼び出す場合: なし

### 必要な接続:

なし

### コマンド構文:

```
db2icons user_name
```

### コマンド・パラメーター:

*user\_name*

デスクトップ・アイコンを追加するユーザー ID。

### 使用上の注意:

Gnome または KDE のデスクトップ環境の実行中にアイコンを生成した場合、手動でデスクトップをリフレッシュしないと、ユーザーに新しいアイコンが表示されないことがあります。

## db2icrt - インスタンスの作成

DB2 インスタンスを作成します。

Windows オペレーティング・システムでは、db2icrt ユーティリティーは %sqllib%bin サブディレクトリーにあります。

UNIX ベースのシステムでは、db2icrt ユーティリティーは DB2DIR/instance ディレクトリーにあります。ここで、DB2DIR には、AIX では /usr/opt/db2\_08\_01 が、それ以外の UNIX ベースのシステムでは /opt/IBM/db2/V8.1 が入ります。フィックスパックまたはモディフィケーション・レベルが代替パスにインストールされている場合、DB2DIR ディレクトリーは、AIX では usr/opt/db2\_08\_FPn に、それ以外のすべての UNIX ベースのシステムでは opt/IBM/db2/V8.FPn になります。ここで、n はフィックスパックまたはモディフィケーション・レベルの番号を表します。db2icrt ユーティリティーは、呼び出し元のディレクトリーにインスタンスを作成します。

### 権限:

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:

#### UNIX ベースのシステムの場合

```
db2icrt [-h] [-d] [-a AuthType] [-p PortName]
        [-s InstType] [-w WordWidth] [-u FencedID] InstName
```

#### Windows オペレーティング・システムの場合

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

### コマンド・パラメーター:

#### 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。

**-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。

**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。

**注:** このオプションは、PE データベース・システム、単一パーティション ESE データベース・システム、または DB2 Connect のインスタンスを作成する場合に指定します。

**WSE** Workgroup Server Edition サーバーのインスタンスを作成する際に使用してください。

**-w** *WordWidth*

作成されるインスタンスの幅 (バイト単位: 31、32、または 64) を指定します。適切な幅を選択できるようにするためには、DB2 の前提条件バージョン (31 ビット、32 ビット、または 64 ビット) がインストールされていなければなりません。デフォルト値は、サポートされる最低ビット幅ですが、これは DB2 UDB のインストールされているバージョン、その実行プラットフォーム、およびインスタンス・タイプによって異なります。このパラメーターは AIX 5L、HP-UX、および Solaris オペレーティング環境上だけで有効です。

**-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。-u オプションは、サーバー・インスタンスを作成する場合に必要です。

*InstName*

インスタンスの名前を指定します。

**Windows オペレーティング・システムの場合****-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。

**注:** 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。

## db2icrt - インスタンスの作成

**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。

**注:** このオプションは、PE データベース・システム、単一パーティション ESE データベース・システム、または DB2 Connect のインスタンスを作成する場合に指定します。

**WSE** Workgroup Server Edition サーバーのインスタンスを作成する際に使用してください。

**-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。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

*InstName*

インスタンスの名前を指定します。

**例:**

**例 1:**

| AIX マシン上でディレクトリー /u/db2inst1/sqllib/bin に "db2inst1" という名前  
| のインスタンスを作成するには、このディレクトリーから次のコマンドを発行しま  
| す。

| クライアント・マシンの場合:

| usr/opt/db2\_08\_01/instance/db2icrt db2inst1

| サーバー・マシンの場合:

| usr/opt/db2\_08\_01/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  
| ジャーを実行するユーザー ID です。

**例 2:**

AIX マシン上に代替フィックスパック 1 がインストールされている場合、次のコマンドを実行して、代替フィックスパック 1 のインストール・パスからフィックスパック 1 のコードを実行するインスタンスを作成します。

```
/usr/opt/db2_08_FP1/instance/db2icrt -u db2fenc1 db2inst1
```

### 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。たとえば、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 インスタンスで稼動するように移行します。

### 関連資料:

- 102 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。インスタンスのリストからインスタンス項目を除去します。

Windows オペレーティング・システムでは、db2idrop ユーティリティは %sqllib%bin サブディレクトリーにあります。UNIX ベースのシステムでは、このユーティリティは DB2DIR/instance ディレクトリーにあります。ここで、DB2DIR には、AIX では /usr/opt/db2\_08\_01 が、それ以外の UNIX ベースのシステムでは /opt/IBM/db2/V8.1 が入ります。フィックスパックまたはモディフィケーション・レベルが代替パスにインストールされている場合、DB2DIR ディレクトリーは、AIX では usr/opt/db2\_08\_FPn に、それ以外のすべての UNIX ベースのシステムでは opt/IBM/db2/V8.FPn になります。ここで、n はフィックスパックまたはモディフィケーション・レベルの番号を表します。

フィックスパックまたはモディフィケーション・レベルが代替パスにインストールされている場合、インストール・パスから db2idrop ユーティリティを実行することによってインスタンスをドロップできます。これを行うには、ドロップするインスタンスのインストール・パスにインストール・コードが置かれている必要があります。インストール・パスからインストール・コードを除去し、その後別のインストール・パスから db2idrop ユーティリティを呼び出してそのパスにあるインスタンスを除去しようとしても、インスタンスは除去できません。

たとえば、あるセットアップされたシステムで、DB2 バージョン 8 と DB2 バージョン 8 の代替フィックスパック 1 の両方がインストールされているとします。インスタンス db2inst1 はバージョン 8 のインストール・パスに作成され、インスタンス db2inst2 は代替フィックスパック 1 (AFP1) のパスに作成されました。

- どのインストール・コードも除去されなかった場合は、AFP1 パスから db2idrop ユーティリティを実行することによって、たとえバージョン 8 のパスのコードに対して実行される db2inst1 が作成される場合でも、両方のインスタンスを除去することができます。
- バージョン 8 のパスにあるインストール・コードが除去された場合、AFP1 パスから db2idrop を呼び出しても db2inst1 を除去できません。ただし、db2inst2 をドロップすることは可能です。

### 権限:

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:

#### UNIX ベースのシステムの場合

```
db2idrop -h InstName
```

#### Windows オペレーティング・システムの場合

```
db2idrop -f InstName
```

### コマンド・パラメーター:



**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。

**InstName**

インスタンスの名前を指定します。

**Windows オペレーティング・システムの場合****-f**

強制アプリケーション・フラグを指定します。このフラグを指定すると、このインスタンスを使用しているすべてのアプリケーションが強制的に終了させられます。

**InstName**

インスタンスの名前を指定します。

**例:**

AIX マシン上で、"db2inst1" という名前のインスタンスが、バージョン 8 のインストール・パスにあるバージョン 8 のコードを実行します。"db2inst2" という名前のインスタンスが、代替フィックスパック 1 のインストール・パスにあるバージョン 8 フィックスパック 1 のコードを実行します。db2inst1 をドロップするためのコマンドを、次の代替フィックスパック 1 インストール・パスから発行することができます。

```
/usr/opt/db2_08_FP1/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。

**関連資料:**

- 106 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

システムで使用可能なインスタンスをすべてリストします。

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 %sqllib%\bin サブディレクトリーにあります。UNIX ベースのシステムでは、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。ここで、DB2DIR には、AIX では /usr/opt/db2\_08\_01 が、それ以外の UNIX ベースのシステムでは /opt/IBM/db2/V8.1 が入ります。フィックスパックまたはモディフィケーション・レベルが代替パスにインストールされている場合、DB2DIR ディレクトリーは、AIX では usr/opt/db2\_08\_FPn に、それ以外のすべての UNIX ベースのシステムでは opt/IBM/db2/V8.FPn になります。ここで、n はフィックスパックまたはモディフィケーション・レベルの番号を表します。

このコマンドは、たとえば代替フィックスパック・コード・パスなどの、任意のコード・パスから発行できます。結果はすべて同じです。

権限:

なし

コマンド構文:

**Windows オペレーティング・システムの場合**

▶▶ db2ilist ◀◀

**UNIX ベースのシステムの場合**

▶▶ db2ilist   
           -w 31   
               32   
               64   
           -p   
           -a   
           inst\_name ◀◀

**コマンド・パラメーター:**

**-w** 31、32、または 64 ビット・インスタンスをリストする。-w オプションは -p オプションと共に指定でき、-a オプションに取り替えられます。

**-p** インスタンスが実行される DB2 インストール・パスをリストする。-p オプションは -a オプションと共に使用でき、-a オプションに取り替えられます。

**-a** インスタンスと関連した DB2 インストール・パス、およびそのビット幅 (32 か 64) を含む情報をリストします。32 ビットの場合に返される情報は、Linux 上 (S/390 および zSeries 上) の DB2 では 31 ビットを示します。

**-inst\_name**

指定したインスタンスの情報をリストします。インスタンスが指定されていない場合、db2ilist は現在の DB2 リリースのすべてのインスタンスに関する情報をリストします。

**-h** 使用情報を表示します。

**例:**

次の 4 つのインスタンスを持つ AIX システムを検討します。

- バージョン 8 コード・パスにインストールされた "db2inst1" という名前の 32 ビット・インスタンス。
- バージョン 8 コード・パスにインストールされた "db2inst2" という名前の 64 ビット・インスタンス。
- バージョン 8 代替フィックスパック 1 コード・パスにインストールされた "db2inst3" という名前の 32 ビット・インスタンス。
- バージョン 8 代替フィックスパック 1 コード・パスにインストールされた "db2inst4" という名前の 64 ビット・インスタンス。

db2ilist コマンドを発行すると、コマンドのパラメーターに応じたさまざまな出力が生成されます。

• db2ilist

```
db2inst1
db2inst2
db2inst3
db2inst4
```

• db2ilist -a

```
db2inst1 32 /usr/opt/db2_08_01
db2inst2 64 /usr/opt/db2_08_01
db2inst3 32 /usr/opt/db2_08_FP1
db2inst4 64 /usr/opt/db2_08_FP1
```

• db2ilist -w 64 -p

```
db2inst2 /usr/opt/db2_08_01
db2inst4 /usr/opt/db2_08_FP1
```

## db2imigr - インスタンスの移行

データベース・マネージャーのインストールの後に、既存のインスタンスを移行します。このコマンドは、UNIX ベースのシステムのみで使用可能です。Windows では、インスタンスの移行も通常の移行の中で暗黙的に実行されます。

このユーティリティは、DB2DIR/instance ディレクトリーにあります。ここで、DB2DIR の部分には、AIX では /usr/opt/db2\_08\_01 が、UNIX ベースのシステムでは /opt/IBM/db2/V8.1 が入ります。

**注:** DB2 バージョン 7 から、代替パスにインストール済みのバージョン 8 フィックスパックまたはモディフィケーション・レベルへのインスタンスの移行は、Enterprise Server Edition でのみサポートされています。バージョン 7 のインスタンスに Warehouse Manager や Data Links Manager などの他の製品が含まれている場合、それらの製品を、代替パスにインストール済みのフィックスパックまたはモディフィケーションに移行することはできません。

### 権限:

UNIX ベース・システムでは root 権限。

### コマンド構文:

```

db2imigr [-d] [-a AuthType] [-u FencedID] [-g dlfmxgrpID] InstName

```

### コマンド・パラメーター:

**-d** デバッグ・モードをオンにします。このオプションは、DB2 サポートからの指示があった場合にのみ使用してください。

**-a AuthType**  
インスタンスの認証タイプ (SERVER、CLIENT、または SERVER\_ENCRYPT) を指定します。デフォルトは SERVER です。

**-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 クライアントだけがインストールされている場合は、このオプションは必要ありません。

**-g dlfmxgrpID**  
dlfmxgrp ID を指定します。このオプションは、バージョン 7 以前のデータ・リンク・ファイル・マネージャーのインスタンスを移行する場合にのみ、使用してください。ここで指定されるシステム・グループ ID は、データ・リンク・ファイル・マネージャー専用の ID です。このグループのメンバーとして定義される唯一のシステム・ユーザー ID は、DLFM データベース・インスタンスの所有者 (デフォルトでは dlfm) です。

### InstName

インスタンスの名前を指定します。

### 使用上の注意:

| /usr/lib および /usr/include 内に、移行元のバージョンへのシンボリック・リン  
| クが存在する場合、 **db2imigr** はそれらを除去します。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。

### 関連概念:

- 「*Data Links Manager 概説およびインストール*」の『DB2 Data Links Manager をインストールする前に (AIX)』
- 「*Data Links Manager 概説およびインストール*」の『DB2 Data Links Manager をインストールする前に (Solaris オペレーティング環境)』

## db2inidb - ミラーリングされたデータベースの初期化

スプリット・ミラー環境のミラーリングされたデータベースを初期化します。ミラーリングされたデータベースは、ロールフォワード・ペンディング状態にある 1 次データベースの複製として初期化したり、1 次データベースをリストアするためのバックアップ・イメージとして使用できます。このコマンドはスプリット・ミラー・データベースに対してのみ実行可能であり、スプリット・ミラーを使用するには、その前にこのコマンドを実行しておく必要があります。

### 権限:

以下のいずれか。

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

### 必要な接続:

なし

### コマンド構文:

```

▶▶ db2inidb database_alias AS 

|          |
|----------|
| SNAPSHOT |
| STANDBY  |
| MIRROR   |

 RELOCATE USING configFile

```

### コマンド・パラメーター:

#### database\_alias

初期設定するデータベースの別名を指定します。

#### SNAPSHOT

ミラーリングされたデータベースは、1 次データベースの複製として初期化されることを指定します。

#### STANDBY

データベースをロールフォワード・ペンディング状態にすることを指定します。

**注:** 1 次データベースから新しいログファイルを取り出し、スタンバイ・データベースに適用することが可能です。スタンバイ・データベースは、1 次データベースがダウンした場合に、その代わりに使用できます。

#### MIRROR

ミラーリングされたデータベースを、1 次データベースをリストアするために使用できるバックアップ・イメージとして使用することを指定します。

#### RELOCATE USING configFile

データベースをスナップショット、スタンバイ、またはミラーとして初期化する前に、指定された *configFile* の中でリストされている情報に基づいて、データベース・ファイルを再配置することを指定します。 *configFile* の形式については、db2relocatedb - データベースの再配置コマンドを参照してください。

### 使用上の注意:

パーティション・データベース環境では、**db2inidb** は各パーティション上でそれぞれ実行する必要があります。それから、すべてのパーティションのスプリット・ミラーを使用することができます。**db2inidb** は、**db2\_all** コマンドを使用してすべてのパーティションで同時に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 (変更するパーティションの NODENUM 値が含まれる) を用意する必要があります。たとえば、データベースの名前を変更する場合は、すべてのパーティションが影響を受けることになり、各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、そのパーティション上で一度だけ **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『分割ミラーを使用したデータベースのクローン作成』
- ・「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『分割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『分割ミラーをバックアップ・イメージとして使用する』

### 関連資料:

- ・ 206 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ーは、INSPECT CHECK 結果からのデータを ASCII 形式にフォーマットします。このユーティリティーを使用して、検査の詳細を表示します。db2inspf ユーティリティーによるフォーマットには、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。

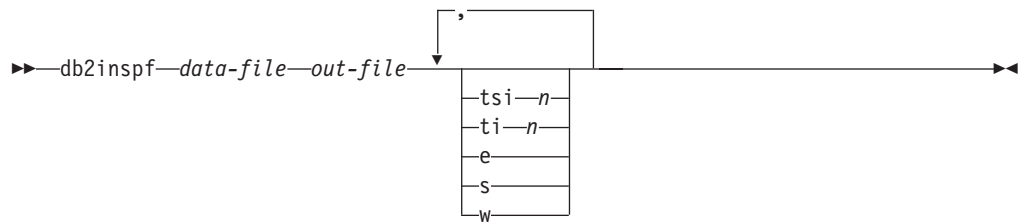
### 権限:

このユーティリティーにはすべてのユーザーがアクセス可能ですが、結果ファイルに対してこのユーティリティーを実行するには、それらの読み取り許可がなければなりません。

### 必要な接続:

なし

### コマンド構文:



### コマンド・パラメーター:

#### data-file

フォーマットを必要とする不定形式の検査結果ファイル。

#### out-file

フォーマット済み出力の出力ファイル。

- tsi 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。



第 1 章 システム・コマンド 119

## db2iseta - インスタンス作成インターフェースの開始

他のインスタンス・スクリプトも一緒にあります。他のインスタンス・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、UNIX 上の DB2 インスタンスのファイルではありません。

3. db2iseta は、サポートされているすべての UNIX プラットフォームで実行できます。

## db2iupdt - インスタンスの更新

Windows オペレーティング・システムでは、このコマンドは、パーティション・データベース・システムで使用する単一パーティションのインスタンスを更新します。これは、¥sqllib¥bin サブディレクトリーにあります。

UNIX ベースのシステムの場合、このコマンドは、新規システム構成の取得を可能にしたり、特定の製品オプションやフィックスパック、またはモディフィケーション・レベルのインストールまたは除去に関連した関数にアクセスできるようにするために、指定された DB2 インスタンスを更新します。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。ここで、DB2DIR の部分には、AIX では /usr/opt/db2\_08\_01 が、UNIX ベースのシステムでは /opt/IBM/db2/V8.1 が入ります。フィックスパックまたはモディフィケーション・レベルが代替パスにインストールされている場合、DB2DIR ディレクトリーは、AIX では usr/opt/db2\_08\_FpN に、それ以外のすべての UNIX ベースのシステムでは opt/IBM/db2/V8.FpN になります。ここで、n はフィックスパックまたはモディフィケーション・レベルの番号を表します。

フィックスパックまたはモディフィケーション・レベルが代替パスにインストールされている場合、インストール・パスから **db2iupdt** ユーティリティーを実行することによってインスタンスを更新できます。**db2iupdt** コマンドを実行するとインスタンスが更新され、このコマンドが出されたのと同じパスにインストールされたコードに対して実行されます。

### 権限:

UNIX ベースのシステムでは root 権限、Windows ではローカル管理者。

### コマンド構文:

#### UNIX ベースのシステムの場合

```
db2iupdt [-h] [-d] [-k] [-D] [-s] [-a AuthType]
          [-w WordWidth] [-u FencedID] [-e InstName]
```

#### Windows の場合

```
db2iupdt InstName /u:—username,password /p:—instance profile path
/r:—baseport,endport /h:—hostname
```

### コマンド・パラメーター:

#### UNIX ベースのシステムの場合

##### -h または -?

使用方法を表示します。

##### -d デバッグ・モードをオンにします。

## db2iupdt - インスタンスの更新

- 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。
- w WordWidth**  
作成するインスタンスの幅 (ビット単位) を指定します。有効な値は、31、32、および 64 です。このパラメーターは AIX、HP-UX、Linux for AMD64、および Solaris オペレーティング環境上だけで有効です。必要なバージョンの DB2 をインストールしておく必要があります (31 ビット、32 ビット、または 64 ビット)。デフォルト値は、更新されるインスタンスのビット幅です。
- 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。

### 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 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。

### 例 (UNIX):

1. インスタンス db2inst1 が、バージョン 8 インストール・パス内のバージョン 8 フィックスパック 1 コードを実行しています。バージョン 8.1.2 がバージョン

8 インストール・パスにインストール済みの場合、バージョン 8 インストール・パスから以下のコマンドを呼び出すことにより、db2inst1 をバージョン 8.1.2 に更新します。

```
db2iupdt db2inst1
```

2. インスタンス db2inst2 が、バージョン 8 インストール・パス内のバージョン 8.1.2 コードを実行しています。バージョン 8.1.2 を代替インストール・パスにインストールする場合、バージョン 8.1.2 の代替インストール・パスから以下のコマンドを呼び出すと、db2inst2 は、代替インストール・パスから実行されるバージョン 8.1.2 に更新されます。

```
db2iupdt db2inst2
```

3. インスタンス db2inst3 が、代替インストール・パス内のバージョン 8.1.2 コードを実行しています。フィックスパック 1 を別の代替インストール・パスにインストール済みの場合、フィックスパック 1 の代替インストール・パスから以下のコマンドを呼び出すと、db2inst3 は、フィックスパック 1 の代替インストール・パスから実行されるフィックスパック 1 に更新されます。

```
db2iupdt -D db2inst3
```

#### 使用上の注意:

インスタンスの実行元のインストール・パスを切り替えて、より高いコード・レベルからより低いコード・レベルにインスタンスを更新する場合 (たとえば、複数レベルの DB2 の間で切り替える場合) には、**db2iupdt** コマンドを **-D** オプションを指定して実行する必要があります。この状況は 3 番目の例で説明されています。

## 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、すでに 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。

**注:** JDBC および CLI は同じパッケージを共有します。CLI パッケージがすで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。

### 権限:

以下のいずれかになります。

- *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する場合は、スキーマに対する CREATEIN 特権
- パッケージが存在する場合は、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:

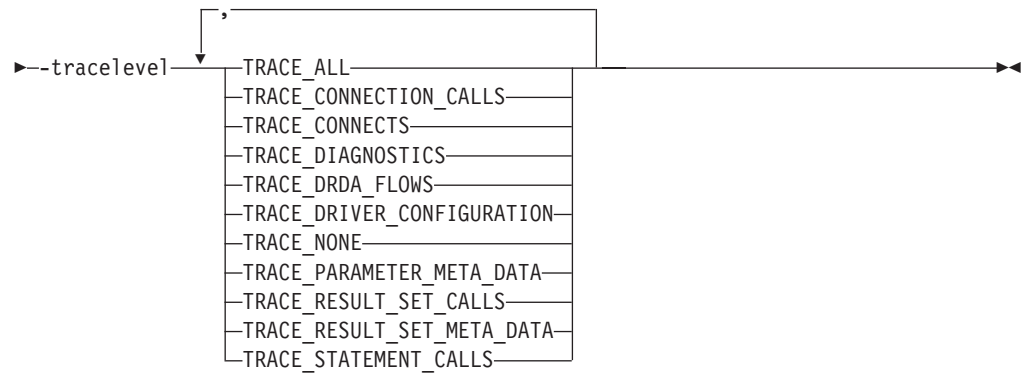
このコマンドは、データベース接続を確立します。

### 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に使用する名前を指定します。

**-password** *password*

ユーザー名のパスワードを指定します。

**-collection** *collection ID*

コレクション ID (CURRENT PACKAGESET)。パッケージに使用します。デフォルトは NULLID です。これは、パッケージ・セットの複数のインスタンスを作成するために使用します。このオプションは、Connection または DataSource プロパティ `currentPackageSet` とともにのみ使用できます。

**-size** *number of packages*

それぞれの DB2 トランザクション分離レベルおよびカーソルの HOLD 設定ごとにバインドする内部パッケージの数。デフォルトは 3 です。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。

権限:

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・ユーザーの認証が使用されません。



---

## db2level - DB2 サービス・レベルの表示

インストール済み DB2 製品の現行バージョンおよびサービス・レベルを表示します。このコマンドからの出力は、デフォルトでコンソールに表示されます。

**権限:**

なし。

**必要な接続:**

なし。

**コマンド構文:**

▶—db2level—◀

**例:**

Windows システムで db2level コマンドを実行すると、通常次のような結果になります。

```
DB21085I Instance "kirton" uses DB2 code release "SQL08010" with level  
identifier "01010106" and informational tokens "DB2 v8.1.0", "n020320" and "".
```

コマンドによる情報出力には、リリース、レベル、およびさまざまな情報トークンが含まれます。

权限:

必要な接続:

なし

コマンド構文:

```

db21icm
-a filename
-l
-p prod-password
    REGISTERED
    CONCURRENT
    INTERNET
    MEASURED
-r prod-password
-u prod-password num-users
-n prod-password num-processors
-e prod-password
    HARD
    SOFT
-v
-h
-?
  
```

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。

- すべての製品を使用可能なライセンス情報とともにリストします。

**-p prod-password** *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。キーワード CONCURRENT、REGISTERED、または CONCURRENT REGISTERED を指定できます。加えて、DB2 UDB Workgroup Server 製品には INTERNET を、DB2 Connect Unlimited 製品には MEASURED を指定できます。

**-r prod-password**

製品のライセンスを除去します。ライセンスが除去されると、製品機能は「試供版」モードになります。特定の製品のパスワードを取得するには、-i オプション付きで **db2licm** コマンドを呼び出します。

**-u prod-password 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。ユーザーの数、およびライセンスが購入された製品のパスワードを指定してください。

**-n prod-password num-processors**

お客様に DB2 を使用するライセンスが与えられている処理装置の数を更新します。

**-e prod-password**

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。

**-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 製品ライセンス・キーの登録』

## db2logsforrfwd - ロールフォワード・リカバリーに必要なログのリスト

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

権限:

必要な接続:

なし。

コマンド構文:

▶▶ `db2logsforrfwd` *path* -all ▶▶

コマンド・パラメーター:

*path*     DB2TSCHG.HIS ファイルの絶対パスおよび名前。

**-all**     詳細情報を表示します。

例:

```
db2logsForRfwd /home/ofer/ofer/NODE0000/S00000001/DB2TSCHG.HIS
```

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

## db2look - DB2 統計および DDL 抽出ツール

要求された 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** 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、**db2set** コマンドを生成し、テスト・データベース上の照会オプティマイザー関連の構成パラメーターとレジストリー変数を、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、便利なが多々あります。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。

**注:** 生成された 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りません。 **db2look** によって生成された DDL を確認してください。

### 権限:

システム・カタログ表に対する SELECT 特権。

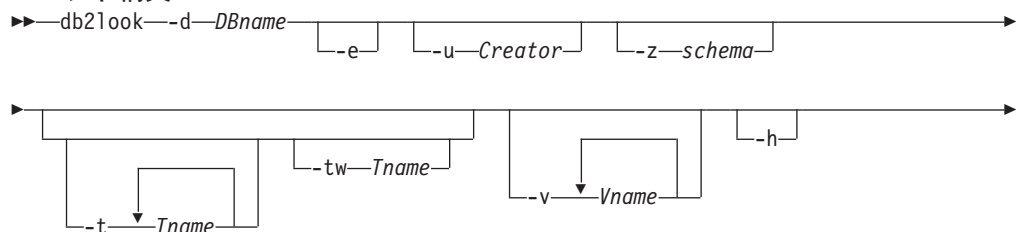
表スペース・コンテナ DDL を生成する場合 (つまり、sqlbotcq、sqlbftcq、sqlbctcq の API を呼び出す場合) などでは、以下のいずれか 1 つが必要です。

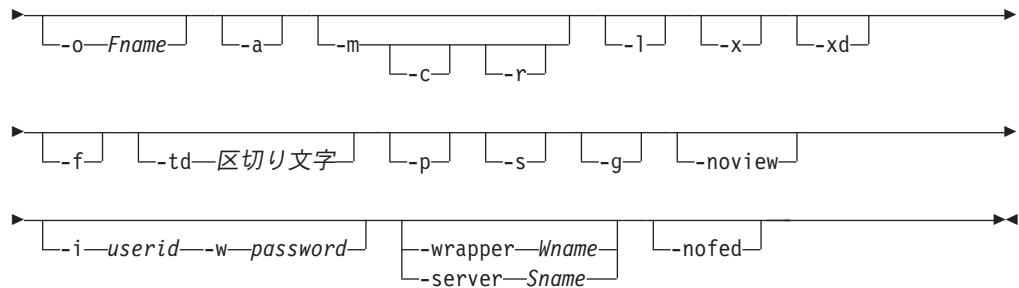
- sysadm
- sysctrl
- sysmaint
- dbadm

### 必要な接続:

なし。 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*

照会する実動データベースの別名。 *DBname* は、DB2 UDB for UNIX、Windows、または DB2 Universal Database for z/OS and OS/390 データベースの名前にすることができます。 *DBname* が DB2 Universal Database for z/OS and OS/390 データベースの場合には、**db2look** ユーティリティーは、OS/390 オブジェクトおよび z/OS オブジェクト用の DDL および UPDATE 統計ステートメントを抽出します。これらの DDL および UPDATE 統計ステートメントは、DB2 UDB データベースには適用できませんが、DB2 Universal Database for z/OS and OS/390 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 UDB 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 Universal Database for z/OS and OS/390 データベースの場合、**db2look** 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
- 自動サマリー表 (AST)
- 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
- 索引の指定
- ストアード・プロシージャ

**注:** **db2look** 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可能であるためには、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・コードが使用可能でなければなりません。

#### **-u** *Creator*

作成者 ID。出力をこの作成者 ID があるオブジェクトだけに制限します。オプション **-a** を指定した場合、このパラメーターは無視されます。 **-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。

#### **-z** *schema*

スキーマ名を指定します。出力をこのスキーマ名のオブジェクトに制限します。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。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。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** は (`"MY TABLE"` のように) 大文字の表を探します。

#### **-tw** *Tname*

*Tname* に指定したパターン基準に一致する表名の DDL を生成します。また、戻されたすべての表のすべての従属オブジェクトの DDL も生成します。 *Tname* は、1 つの値だけでもかまいません。 *Tname* 内の下線文字 (`_`) は、任意の 1 文字を表します。パーセント記号 (`%`) は、ゼロ個以上の文字のストリングを表します。 *Tname* 内の他のすべての文字は、その文字そのものを表します。 **-tw** を指定した場合は、**-t** オプションが無視されます。

#### **-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が対象になります。たとえば、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。

**注:** **-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 ベースのシステムでは、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、列、および索引についての統計を複製します。**-p**、**-g**、および **-s** オプションは、**-m** オプションが指定された場合は無視されます。**-c** オプションおよび **-r** オプションは、オプションで **-m** と共に使用されます。

**-c**

このオプションを **-m** オプションと共に指定する場合には、**db2look** は COMMIT、CONNECT、および CONNECT RESET ステートメントを生成しません。デフォルト・アクションでは、これらのステートメントを生成します。**-m** オプションを指定しない場合、**-c** オプションは無視されます。

**-r**

このオプションと **-m** オプションを共に指定する場合には、**db2look** 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。**-m** オプションを指定しない場合、**-r** オプションは無視されます。

**-l**

このオプションを指定すると、**db2look** ユーティリティーは、ユーザー定義の表スペース、データベース・パーティション・グループ、およびバッファ・プール用の DDL を生成します。以下のデータベース・オブジェクト用の DDL は、**-l** オプションを使用すると抽出されます。

- ユーザー定義表スペース
- ユーザー定義データベース・パーティション・グループ
- ユーザー定義バッファ・プール

**-x**

このオプションを指定すると、**db2look** ユーティリティーは、権限 DDL (GRANT ステートメントなど) を生成します。

**db2look** がサポートする許可には、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 ユーティリティ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** ユーティリティは、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** ユーティリティは、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\_HASH\_JOIN
- | • DB2\_ORDERED\_NLJN
- | • DB2\_NEW\_CORR\_SQ\_FF
- | • DB2\_PART\_INNER\_JOIN
- | • DB2\_INTERESTING\_KEYS

### **-td** *delimiter*

**db2look** 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。

**-p** プレーン・テキスト形式を使用します。

**-s** ポストスクリプト・ファイルを生成します。

#### 注:

1. このオプションは、すべての LaTeX ファイルと .tmp ポストスクリプト・ファイルを除去します。
2. 必要な非 IBM 製のソフトウェアは LaTeX と dvips です。
3. psfig.tex ファイルは、LaTeX 入力パスに置いておく必要があります。

**-g** 索引の取り出しページ・ペアを示すためにグラフを使用します。

#### 注:

1. このオプションは、LaTeX ファイルだけでなく、*filename.ps* ファイルを生成します。
2. 必要な非 IBM 製のソフトウェアは Gnuplot です。
3. psfig.tex ファイルは、LaTeX 入力パスに置いておく必要があります。

### **-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、リモート・システムに常駐するデータベースに対して **db2look** が実行可能になります。

**db2look** 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。

注: 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** ユーティリティーには、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 より少ない、または複数指定された場合は、エラーが返されます。このオプションは、-server オプションが使用されている場合は使用できません。

**-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 より少ない、または複数指定された場合は、エラーが返されます。このオプションは、-wrapper オプションが使用されている場合は使用できません。

**-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。

**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表および索引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表および索引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
```

- オブ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 にあるオブ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、オブ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、オブ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、フェデレートされてい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
```

#### 使用上の注意:

Windows システムでは、**db2look** 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。

**db2look** コマンド行オプションは、どのような順番でも指定できます。必須オプションで、有効なデータベース別名の指定に必要な **-d** オプションを除き、すべてのコマンド行オプションはオプションです。

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使用可能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** は、LIKE 述部を使用して、どの表名が *Tname* 引き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。たとえば、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
```

- 大文字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。たとえば、次のようにします。

```
¥"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、およびバッファーク・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* 引き数) が無視されます。
- -tw オプションは、*Tname* 引き数を 1 つしか受け入れません。
- 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。

### 関連資料:

- 「SQL リファレンス 第 1 巻」の『LIKE 述部』



## db2move - データベース移動ツール

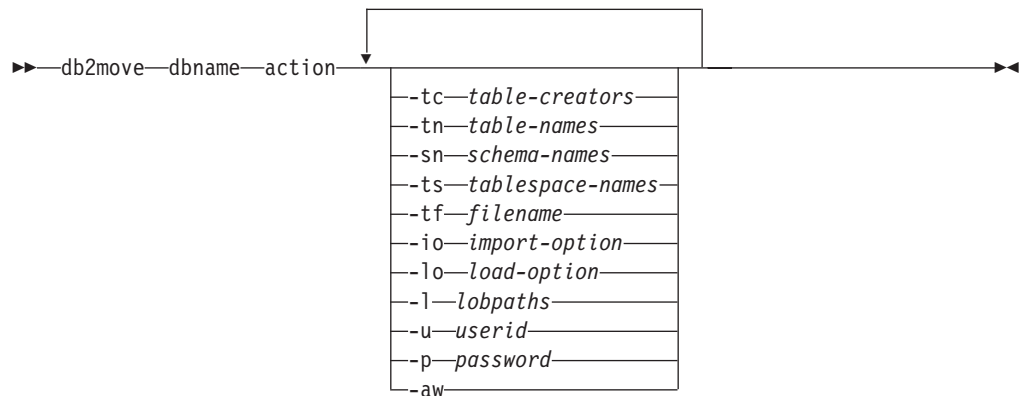
このツールは、ワークステーション上にある DB2 データベース間で、大量の表の移動を容易にします。また、特定のデータベースのシステム・カタログ表を照会し、すべてのユーザー表のリストをコンパイルします。そして、これらの表を PC/IXF フォーマットでエクスポートします。PC/IXF ファイルは、同じシステム上の別のローカル DB2 データベースにインポートまたはロードするか、または別のワークステーション・プラットフォームに転送し、そのプラットフォームで DB2 データベースにインポートまたはロードすることができます。

**注:** 構造型列がある表は、このツールを使用しても移動しません。

### 権限:

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** EXPORT、IMPORT、または LOAD のいずれかです。

**-tc** 表の作成者。デフォルトはすべての作成者です。

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作成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作成者を使用します。複数の作成者を指定する場合、それぞれをコンマで区切る必要があります。作成者 ID 間に空白を入れることはできません。指定できる作成者の最大数は 10 です。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 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されます。指定されない場合、デフォルトではすべてのユーザー表を使用します。複数の表名を指定する場合、それぞれをコンマで区切る必要があります。表名間に空白を入れることはできません。指定できる表名の最大数は 10 です。このオプションを 『-tc』 表作成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、かつ作成者が指定された表作成者と一致する表のみをエクスポートします。

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。

**-sn** スキーマ名。デフォルトはすべてのスキーマです。

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されます。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。複数のスキーマ名を指定する場合は、それぞれの名前をコンマで区切る必要があります。複数のスキーマ名の間に空白を使用することはできません。指定できるスキーマ名の最大数は 10 です。-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、作成者が指定された作成者と一致する表だけをエクスポートします。

**注:** 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。たとえば、'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空白を使用することはできません。指定できる表スペース名の最大数は 10 です。

**注:** 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。たとえば、'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 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されます。表は 1 行に 1 つずつリストする必要があります。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。

**-l** LOB パス。デフォルトは、現行ディレクトリーです。

このオプションは、LOB 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 LOB パスを指定する場合、それぞれをコンマで区切る必要があります。LOB パス間にブランクを入れることはできません。最初のパスでスペースを使い尽くした場合 (EXPORT 中)、または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。

アクションが EXPORT で LOB パスが指定される場合、LOB パス・ディレクトリーのすべてのファイルが削除され、ディレクトリーは除去され、新しいディレクトリーが作成されます。指定されない場合、現行ディレクトリーが LOB パスに使用されます。

**-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。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。

**-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:

- db2move sample export

## db2move - データベース移動ツール

この例では、SAMPLE データベースのすべての表をエクスポートします。すべてのオプションにデフォルトが使用されます。

- `db2move sample export -tc userid1,us*rid2 -tn tbnam1,*tbnam2`

この例では、『userid1』 または 『us%rid2』 のようなユーザー ID で作成され、『tbnam1』 という名前、または 『%tbnam2』 のような表名を持つすべての表がエクスポートされます。

- `db2move sample import -l D:¥LOBPATH1,C:¥LOBPATH2`

この例は、Windows オペレーティング・システムのみに適用されます。コマンドは、SAMPLE データベースのすべての表をインポートします。LOB パス『D:¥LOBPATH1』 および 『C:¥LOBPATH2』 で、LOB ファイルが検索されます。

- `db2move sample load -l /home/userid/lobpath,/tmp`

この例は、UNIX ベースのシステムのみに適用されます。コマンドは、SAMPLE データベースのすべての表をロードします。/home/userid/lobpath サブディレクトリーと tmp サブディレクトリーの両方で、LOB ファイルが検索されます。

- `db2move sample import -io replace -u userid -p password`

この例では、SAMPLE データベースのすべての表が REPLACE モードでインポートされます。指定されたユーザー ID およびパスワードが使用されます。

### 使用上の注意:

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、たとえば別名、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 PC/IXF ファイルとは別に保持されます。LOB ファイルでは、26,000 のファイル名を使用することができます。

LOAD アクションは、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。ロード API が **db2move** によって呼び出されると、CopyTargetList パラメーターが NULL に設定されます。つまり、コピーは行われません。logretain がオンである場合、ロード操作を後でロールフォワードすることはできません。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。

注: 'db2move import' 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*、*logfilsz*、および *logprimary* を更新することによって改善できます。

#### EXPORT 使用時に必要とされるファイル/生成されるファイル

- 入力: なし。

- 出力:

**EXPORT.out** EXPORT アクションの結果の要約。

**db2move.lst** オリジナル表名のリスト、その対応する PC/IXF ファイル名 (tabnnn.ixf)、およびメッセージ・ファイル名 (tabnnn.msg)。このリスト、エクスポートされた PC/IXF ファイル、および LOB ファイル (tabnnnc.yyy) は、**db2move IMPORT** または **LOAD** アクションへの入力として使用されます。

**tabnnn.ixf** 特定の表の、エクスポートされる PC/IXF ファイル。

**tabnnn.msg** 対応する表のエクスポート・メッセージ・ファイル。

**tabnnnc.yyy** 特定の表の、エクスポートされる LOB ファイル。

『nnn』 は表番号です。『c』 はアルファベットの文字です。  
『yyy』 は 001 から 999 の範囲内の数値です。

これらのファイルは、エクスポートされている表に LOB データが入っている場合のみ作成されます。作成されると、これらの LOB ファイルは 『lobpath』 ディレクトリに入れられます。LOB ファイルには、合計 26,000 の使用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:

## db2move - データベース移動ツール

**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。

### 関連資料:

- 131 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。

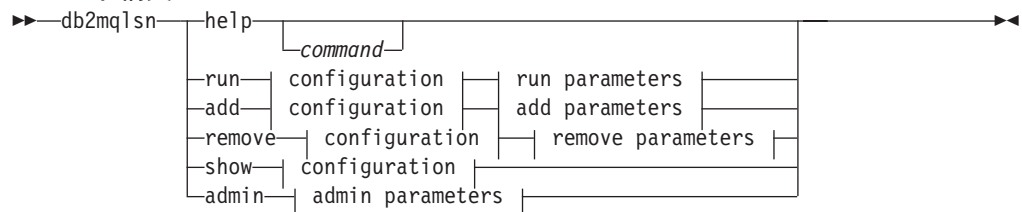
Windows オペレーティング・システムでは、**db2mqdsn** は `sqlllib\bin` サブディレクトリーにあります。UNIX ベース・システムでは、このコマンドは `DB2DIR/instance` ディレクトリーにあります。DB2DIR は、AIX 上では `/usr/opt/db2_08_01`、他のすべての UNIX ベース・システムでは `/opt/IBM/db2/V8.1` です。

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。

### 権限:

- **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 `mqIConfi` 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 `mqIRun` 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—┐
|└--adminQueueList—namelist of admin queue names—┐└--adminQMgr—admin queue manager—┐
|
|▶--adminCommand—┐└--shutdown—┐
|                  └--restart—┐
|
|
コマンド・パラメーター:

help command
    特定のコマンドに関する詳細情報を提供します。コマンド名を指定しない場
    合は、一般的なヘルプ・メッセージが表示されます。

-configDB configuration database
    構成情報が入っているデータベースの名前。

-configUser user ID -configPwd password
    構成データベースにアクセスするための権限情報。

-config configuration name
    個々のタスクを 1 つの構成にグループ化できます。そうすれば、それらの
    タスクを 1 つのグループとしてまとめて実行できます。構成の名前を指定
    しない場合、このユーティリティ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、MOListener 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。(ただし、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 つの構成内でユニークになっています。

## admin

**-adminQueue** *admin queue name* **-adminQueueList** *namelist of admin*

*queue names* **-adminQMGr** *admin queue manager*

管理コマンドを送信するためのキューまたはキュー名の名前リス



ト。キュー・マネージャーを指定しない場合、このユーティリティは WebSphere MQ に構成されているデフォルト・キュー・マネージャーを使用します。

### **-adminCommand** *admin command*

コマンドを実行します。コマンドは、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
```

### **関連概念:**

- 「*IBM DB2 Information Integrator Application Developer's Guide*」の『DB2 Information Integrator の非同期メッセージング』
- 「*IBM DB2 Information Integrator Application Developer's Guide*」の『DB2 内で WebSphere MQ 機能を使用する方法』



## 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。

### 権限:

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。たとえば、次のようにします。

```

CLUSTER_NAME=FINANCE
GROUP_NAME=DB2 Group
IP_ADDRESS=9.21.22.89

```

DB2 インストール・ディレクトリーの CFG サブディレクトリーには、2 つのサンプル構成ファイルがあります。1 つは DB2MSCS.EE というファイルで、これは単一

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

DB2MSCS.CFG ファイルのパラメーターは次のようになっています。

### **DB2\_INSTANCE**

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

### **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、グループがすで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、クラスター内でユニークな値でなければなりません。このパラメーターが指定されると、IP アドレス・タイプの MSCS リソースが作成されます。このパラメーターは、リモート TCP/IP 接続で必要です。単一パーティション環境の場合

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。

注: 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。

## 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。

### 権限:

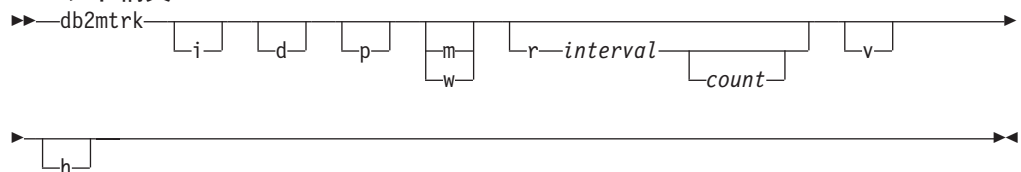
以下のどれかが必要です。

- 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・アタッチ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** UNIX プラットフォームの場合、インスタンス・レベルのメモリーを表示します。 Windows プラットフォームの場合、インスタンス・レベルおよびデータベース・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。  
Windows では使用できません。
- 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。

## db2mtrk - メモリー・トラッカー

**-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: 2002/02/25 at 02:14:10

Memory for instance

monh	other
168	3.1M

Memory for database: EKWAN

utilh	pckcacheh	catcacheh	lockh	dbh	other
56	588.8K	470.2K	432.8K	1.8M	5.1M

Memory for database: AJSTORM

utilh	pckcacheh	catcacheh	lockh	dbh	other
56	55.6K	38.3K	432.8K	1.7M	5.1M

Memory for agent 154374

apph	appctlh	stmth
357.1K	37.2K	209.5K

Memory for agent 213930

apph	appctlh
26.3K	4.0K

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

Tracking Memory on: 2002/02/25 at 17:19:12

Memory for instance

Database Monitor Heap is of size 168 bytes  
Other Memory is of size 3275619 bytes  
Total: 3275787 bytes

Memory for database: EKWAN

Backup/Restore/Util Heap is of size 56 bytes  
Package Cache is of size 56888 bytes  
Catalog Cache Heap is of size 39184 bytes  
Lock Manager Heap is of size 443200 bytes  
Database Heap is of size 1749734 bytes  
Other Memory is of size 5349197 bytes  
Total: 7638259 bytes

Memory for database: AJSTORM

Backup/Restore/Util Heap is of size 56 bytes  
Package Cache is of size 56888 bytes

```
Catalog Cache Heap is of size 39184 bytes
Lock Manager Heap is of size 443200 bytes
Database Heap is of size 1749734 bytes
Other Memory is of size 5349197 bytes
Total: 7638259 bytes
```

```
Memory for agent 154374
Application Heap is of size 26888 bytes
Application Control Heap is of size 4107 bytes
Total: 30995 bytes
```

```
Memory for agent 213930
Application Heap is of size 26888 bytes
Application Control Heap is of size 4107 bytes
Total: 30995 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. Windows プラットフォームでは、-h、-i、または -p フラグを指定する必要があります。 UNIX ベースのプラットフォームでは、-d、-h、-i、または -p フラグ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。たとえばパッケージ・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構成パラメーターは「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。



## 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 NT ベースのオペレーティング・システムのみで使用可能です。

### 権限:

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:**

- 160 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』
- 163 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可能です。

### 有効範囲:

すで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。

**注:** 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。

### 権限:

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ユニークなデータベース・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。

**注:** 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。たとえば、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:

## 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

- 158 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』
- 163 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 NT ベースのオペレーティング・システムのみで使用可能です。

### 権限:

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ユニークなデータベース・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。

**注:** 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。

## db2ndrop - インスタンスからのデータベース・パーティション・サーバーのドロップ

複数の論理データベース・パーティション・サーバーを実行しているマシンから、論理ポート 0 に割り当てられたデータベース・パーティション・サーバーをドロップするには、他の論理ポートに割り当てられている他のすべてのデータベース・パーティション・サーバーを最初にドロップする必要があります。各データベース・パーティション・サーバーには、論理ポート 0 に割り当てられているデータベース・パーティション・サーバーが必ず必要です。

### 関連資料:

- 158 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』
- 160 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 オペレーティング環境上の DB2 だけです。

### 権限:

- 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 オペレーティング環境版の 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 オペレーティング環境版の 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 オペレーティング環境の場合は、異なるカーネル・パラメーターだけが表示されます。現行のカーネル・パラメーターはライブ・カーネルから直接取得されるので、それらは /etc/system 内の Solaris システム仕様ファイルにあるものと一致しないことがあります。ライブ・カーネルから取得されたカーネル・パラメーターが /etc/system にリストされているものと異なる場合、/etc/system ファイルはリブートされないで変更されたか、またはファイル内に構文エラーが存在する可能性があります。HP-UX の場合、-f オプションは、推奨パラメーターのリストと、パラメーター値に対する推奨変更のリストを戻します。  
  
\*\*\*\*\* Please Change the Following in the Given Order \*\*\*\*\*  
  
WARNING [<parameter name>] should be set to <value>
- l '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 オペレーティング環境版の 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 オペレーティング環境版の 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。たとえば、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 オペレーティング環境版の 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 プロセスが使用可能な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** ユーティリティを実行した場合に生成される出力例を以下に示します。

**注:** 受け取る結果はマシンに特定のもののなので、受け取る結果は使用する環境によって異なります。

```
db2osconf -t 500

set msgsys:msginfo_msgmax = 65535
set msgsys:msginfo_msgmnb = 65535
set msgsys:msginfo_msgssz = 32
set msgsys:msginfo_msgseg = 32767
set msgsys:msginfo_msgmap = 2562
set msgsys:msginfo_msgmni = 2560
set msgsys:msginfo_msgtql = 2560
set semsys:seminfo_semmap = 3074
set semsys:seminfo_semmni = 3072
set semsys:seminfo_semmns = 6452
set semsys:seminfo_semmnu = 3072
set semsys:seminfo_semume = 600
set shmsys:shminfo_shmmax = 2134020096
set shmsys:shminfo_shmmni = 3072
set shmsys:shminfo_shmseg = 600

Total kernel space for IPC:
0.35MB (shm) + 1.77MB (sem) + 1.34MB (msg) == 3.46MB (total)
```

set semsys:seminfo\_semume および set shmsys:shminfo\_shmseg のための推奨値は、db2osconf -t 500 を実行して入手できる追加の値です。

#### 使用上の注意:

特定の DB2 ワークロードに基づくカーネル・パラメーターを推奨することは可能ですが、このレベルの正確さには利点がありません。カーネル・パラメーター値が実際に必要な値と過度に接近していて、ワークロードが将来変更される場合、DB2 にはプロセス間通信 (IPC) リソースの不足の問題が生じることがあります。IPC リソースが不足すると DB2 に計画外の停止が生じて、カーネル・パラメーターを増加させるためにリ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。たとえば、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 オペレーティング環境の場合、**db2osconf** ユーティリティには、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は、以下の特別装置にアクセスするので (アクセスは読み取り専用です)、root として実行するか、グループ sys によって実行する必要があります。

```
crw-r----- 1 root    sys      13,  1 Jul 19 18:06 /dev/kmem
crw-rw-rw-  1 root    sys      72,  0 Feb 19 1999 /dev/ksyms
crw-r----- 1 root    sys      13,  0 Feb 19 1999 /dev/mem
```

## db2pd - DB2 のモニターおよびトラブルシューティング

db2pd ユーティリティーは、DB2 メモリー・セットから情報を取り出します。

### 権限:

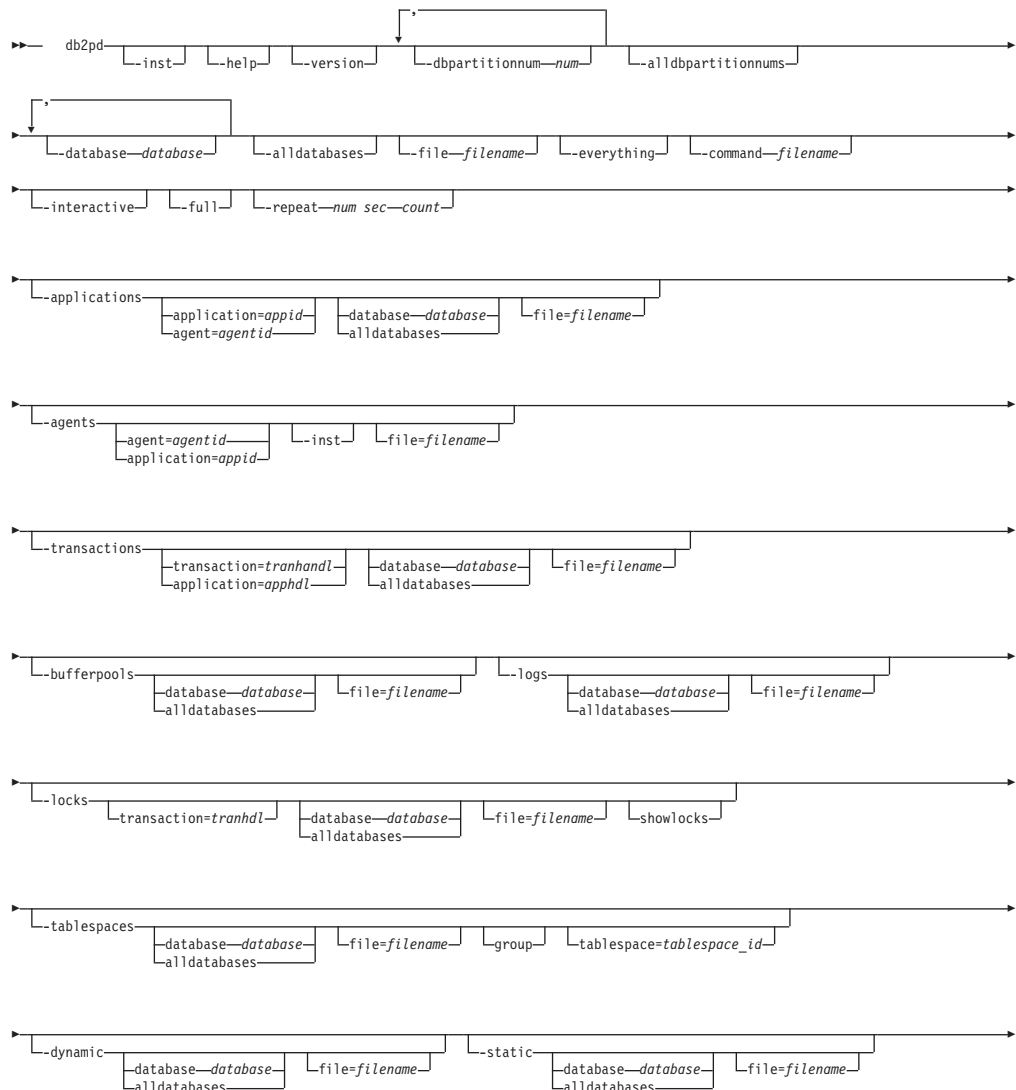
以下のいずれかが必要です。

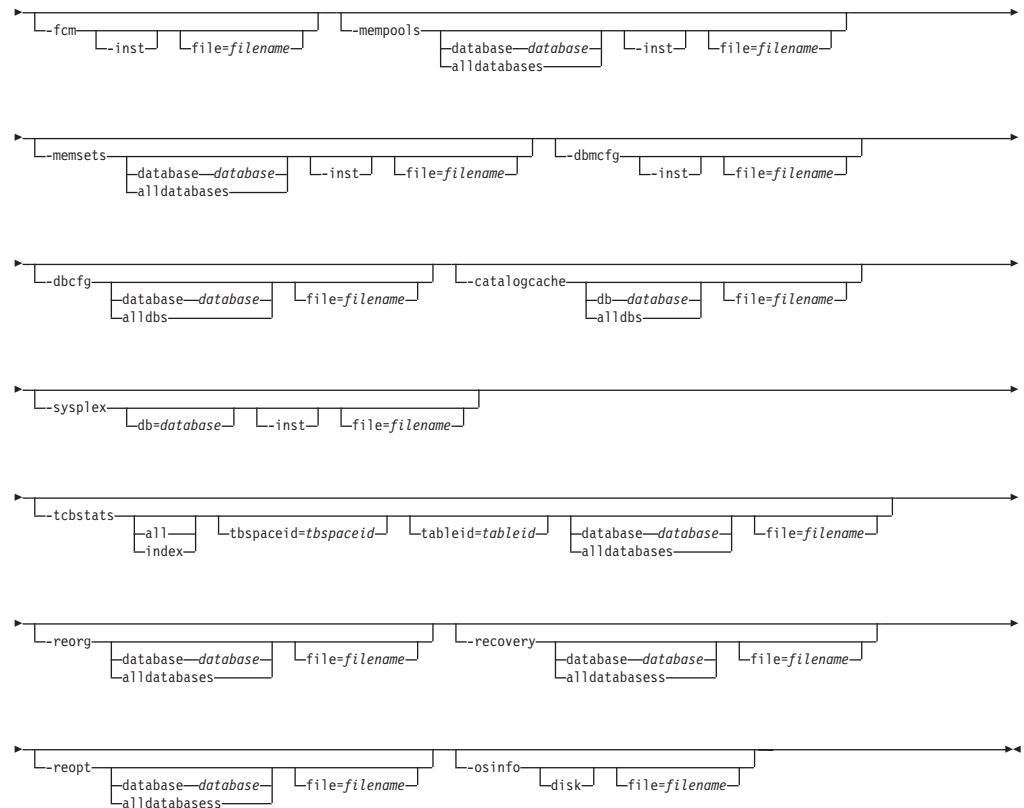
- Windows ベースのプラットフォームの場合、*sysadm* 権限レベル。
- UNIX ベースのプラットフォームの場合、*sysadm* 権限レベル。さらに、インスタンスの所有者でなければなりません。

### 必要な接続:

なし。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:

**-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**

インスタンスのすべてのデータベース・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**

マシン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** 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。

**-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。

**-agents**

エージェントに関する情報を戻します。

エージェント ID が指定される場合、そのエージェントに関する情報が戻されます。

アプリケーション ID が指定される場合、そのアプリケーションの作業を実行するすべてのエージェントに関する情報が戻されます。

出力の有効範囲として設定したいデータベースを選択した場合、このオプションに **-inst** オプションを指定します。

**-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。

**-logs** ログ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。

#### **-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。

#### **-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。

#### **-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。

#### **-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。

#### **-reorg**

表の再編成に関する情報を戻します。

#### **-recovery**

リカバリー・アクティビティー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。

### 例:

以下の例は、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ために、コマンド行から **db2pd** コマンドを呼び出す方法を示しています。

```
db2pd -agents
```

以下の例は、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ために、コマンド行から **db2pd** コマンドを呼び出す方法を示しています。この例では、**db2pd** コマンドを呼び出す前に、**DB2PDOPT** 環境変数が **-agents** パラメーターを指定して設定されています。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
```

以下の例は、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ために、コマンド行から **db2pd** コマンドを呼び出す方法を示しています。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** ファイルに設定されます。**-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
```

以下の例は、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ために、コマンド行から **db2pd** コマンドを呼び出す方法を示しています。

```
db2pd -inst -all dbs
```

### 使用上の注意:

**db2pd** の各種パラメーターを指定する場合に生成される出力について、以下のセクションで説明します。

- 173 ページの『-applications』
- 173 ページの『-agents』
- 174 ページの『-transactions』
- 175 ページの『-bufferpools』
- 176 ページの『-logs』
- 176 ページの『-locks』
- 177 ページの『-tablespaces』
- 180 ページの『-dynamic』
- 181 ページの『-static』
- 183 ページの『-fcm』
- 183 ページの『-mempools』
- 184 ページの『-memsets』
- 184 ページの『-dbmcfg』

- 185 ページの 『-dbcfg』
- 185 ページの 『-catalogcache』
- 188 ページの 『-sysplex』
- 188 ページの 『-tcbstats』
- 190 ページの 『-reorg』
- 191 ページの 『-recovery』
- 193 ページの 『-reopt』
- 193 ページの 『-osinfo』

#### **-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。

##### **AppHandl**

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。

#### **-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。

**タイプ** 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。

**State** トランザクションの状態。

**Tflag** トランザクション・フラグ。値は以下のとおりです。

- 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 ユーティリティの実行のために終了しました。
- 0x00000020。トランザクションは、アプリケーションに対する優先順位の高いロックを譲渡します (この値は、DB2 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通常発生します)。
- 0x00000040。トランザクションは、アプリケーションに対する優先順位の高い行レベルのロックを譲渡しません (この値は、DB2 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通常発生します)。



**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**

使用可能なバッファース・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。

**ES** 拡張記憶域がバッファース・プールに関して使用可能かどうかについて、Y または N で示します。

**NumTbsp**

バッファーク・プールを使用する表スペースの数。

**PgsLeft**

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

**CurrentSz**

ページ内のバッファーク・プールの現行サイズ。

**PostAlter**

バッファーク・プールが再度開始されるとき、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。

**-logs パラメーター:**

-logs パラメーターについては、以下の情報が戻されます。

**Current Log Number**

現在アクティブなログの数。

**Pages Written**

現在のログに書き込まれている現行ページ。

**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。

**タイプ** ロック・タイプ。値は以下のとおりです。

- Row
- Pool
- 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。値は以下のとおりです。

- no lock
- IS
- IX
- S
- SIX
- X
- IN
- Z
- U
- NS
- NX
- W
- NW

**Sts** ロック状況。値は以下のとおりです。

- 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。

**HldCnt**

現在保持されているロックの数。

**Att** ロックの属性。

**Rlse** ロック解放フラグ。

**-tablespaces** パラメーター:

-tablespaces パラメーターについては、以下の情報が戻されます。

**Id** 表スペース ID。

**タイプ** 表スペースのタイプ。値は以下のとおりです。

- SMS
- DMS

**Content**

内容のタイプ。値は以下のとおりです。

- Any
- Long
- SysTmp
- UsrTmp

**PageSize**

表スペースに使用されるページ・サイズ。

**ExtentSize**

エクステンツのサイズ (ページ数)。

**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。

**BufIID** この表スペースのマップ先のバッファー・プールの ID。

**BufIIDDisk**

次の始動時のこの表スペースのマップ先のバッファー・プールの ID。

**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
- 0x0020000 - 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

### TotPages

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、この値は常に 0 です。

### 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 (コンテナの使用可能ページ・フィールドで報告される)。

SMS 表スペースの場合、この値は常に 0 です。

### 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、この値は常に 0 です。

### PndFreePgs

使用できないものの、現在のすべての未解決のトランザクションがコミットするときに使用可能になるページ数。

### FreePgs

DMS 表スペースの場合、表スペース内の使用可能なページ数。

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。

### MinRecTime

表スペースの最小リカバリー時間。

### NQuiescers

静止者の数。

### 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)。

**Name** 表スペースの名前。

以下の出力は、コンテナについて説明します。

**Tspld** 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。

**タイプ** コンテナのタイプ。値は以下のとおりです。

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

**TotalPages**

コンテナ内のページ数。

**UsablePgs**

コンテナ内の使用可能なページ数。

**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。

**Dynamic Cache:**

**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。

**Dynamic SQL Statements:**

**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。**Dynamic SQL Environments:****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。**Dynamic SQL Variations:****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。**Static Cache:****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。

**Packages:**

**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。

**Sections:**

**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。

**Bufs** 高速コミュニケーション・マネージャー (FCM) の現在の空きバッファの数。

#### **BufLWM**

処理中の FCM 空きバッファの最低数。

#### **Anchors**

現在の空きメッセージ・アンカーの数。

#### **AnchLWM**

処理中の空きメッセージ・アンカーの最低数。

#### **Entries**

現在の空き接続項目の数。

#### **EntryLWM**

処理中の空き接続項目の最低数。

**RQBs** 現在の空き要求ブロックの数。

#### **RQBLWM**

処理中の空き要求ブロックの最低数。

**DBP** データベース・パーティション・サーバーの数。

#### **TotBufSnt**

**db2pd** 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。

#### **TotBufRcv**

**db2pd** 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** 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・サーバーの間の接続通信状況。値は以下のとおりです。

- Not Active
- Pending
- Pending Ack
- Active
- Congested
- Failed
- Reconnect

#### **-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。

**MemSet**

メモリー・プールを所有するメモリー・セット。

**PoolName**

メモリー・プールの名前。

**Id**      メモリー・プール ID。

**Overhead**

プール構造に必要な内部オーバーヘッド。

**LogSz**   プール・メモリー要求の現在の合計。

**LogUpBnd**

現在の論理サイズの上限。

**LogHWM**

論理サイズの最高水準点。

**PhySz**   論理サイズに必要な物理メモリー。

**PhyUpBnd**

現在の物理サイズの上限。

**PhyHWM**

処理中の最大物理サイズ。

**Bnd**      内部バインド・ストラテジー。

**BlkCnt**

メモリー・プール内に割り振られたブロックの現在の数。

**CfgParm**

報告されるプールのサイズを宣言する構成パラメーター。

**-memsets**   パラメーター:

-memsets   パラメーターについては、以下の情報が戻されます。

**Name**    メモリー・セットの名前。

**Address**

メモリー・セットのアドレス。

**Id**      メモリー・セット ID。

**Size**    メモリー・セットのサイズ (バイト単位)。

**Key**    メモリー・セット・キー (UNIX ベース・プラットフォームのみ)。

**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。

**タイプ**   メモリー・セットのタイプ。

**Ov**      プール・オーバーフローが許可されているかどうかについて、 Y または N  
で示します。

**OvSize**

メモリー・プールのオーバーフロー域のサイズ。

**-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・キャッシュで使用される現在のバイト数。

###### **Maximum Size**

キャッシュに使用できるメモリの最大量 (データベース・グローバル・メモリの最大量まで)。

###### **High Water Mark**

処理中の最大物理サイズ。

##### **SYSTABLES:**

###### **Schema**

表のスキーマ修飾子。

###### **Name** 表の名前。

###### **Type** 表のタイプ。

###### **TableID**

表 ID。

###### **TbpaceID**

表が置かれている表スペースの ID。

###### **LastRefID**

表を参照した最後のプロセス ID。

###### **CatalogCache LoadingLock**

キャッシュ項目のロックをロードするカタログ・キャッシュの名前。

###### **CatalogCache UsageLock**

キャッシュ項目の使用ロックの名前。

###### **Sts** 項目の状況。値は以下のとおりです。

- 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。値は以下のとおりです。

- 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。値は以下のとおりです。

- 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。値は以下のとおりです。

- 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。値は以下のとおりです。

- V (有効)。
- I (無効)。

**SYSNODEGROUPS:****PMapID**

パーティション・マップ ID。

**RBalID**

データ再分散に使用されたパーティション・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。値は以下のとおりです。

- 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・サーバーのユニークな名前。

**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。値は以下のとおりです。

- 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考慮されません。
- 2。不良。サーバーは以前使用不可能でしたが、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 Table Stats:**

**TbpaceID**

表スペース ID。

**TableID**

表 ID。

**TableName**

表の名前。

**SchemaNm**

表の名前を修飾するスキーマ。

**Scans** 表に対して実行されたスキャンの数。

**ObjClass**

オブジェクト・クラス。値は以下のとおりです。

- Perm (永続)。
- Temp (一時)。

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

**DataSize**

データ・オブジェクト内のページ数。

**IndexSize**

表の索引によって使用中のページ数。

**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**

表に実行された挿入の数。

**Updates**

表に対して実行された更新の数。

**Deletes**

表に対して実行された削除の数。

**OvFIReads**

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。

**OvFiCrtes**

作成された新しいオーバーフローの数。

**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。

**TCB Index Stats:**

注: 以下のデータは、-tcbstats パラメーターに -all または -index オプションを指定する場合にのみ表示されます。

**TbspacelD**

表スペース ID。

**TableID**

表 ID。

**SchemaNm**

表の名前を修飾するスキーマ。

**ID**      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の数。

**BndrySplits**

境界リーフの分割数 (最低位または最高位のキーに挿入されたものの)。

**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。

**TabSpaceID**

表スペース ID。

**TableID**

表 ID。



**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。値は以下のとおりです。

- 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。

**タイプ** 再編成のタイプ。値は以下のとおりです。

- Online
- Offline

**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。値は以下のとおりです。

- Started
- Paused
- Stopped
- Done
- Truncat

**Completion**

表の再編成の成否の標識。値は以下のとおりです。

- 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。

**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。

**-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。値は以下のとおりです。

- 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。値は以下のとおりです。

- Tablespace Rollforward Recovery
- Database Rollforward Recovery
- Crash Recovery

**Invoker Type**

リカバリー操作の呼び出し方法。値は以下のとおりです。

- 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** 現在のフェーズの番号。

**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。値は以下のとおりです。

- 1 バイト。
- 2 エクステンツ。
- 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**

180 ページの『-dynamic』を参照。

**Dynamic SQL Environments**

180 ページの『-dynamic』を参照。

**Dynamic SQL Variations**

180 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。

**NullID** 値がヌル終了かどうかを示すフラグ。

**Len** 可変値の長さ (バイト単位)。

**Data** 変数に使用される値。

**-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 information: (Windows, AIX, HP-UX, Sun, および Linux のみ)****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では、オペレーティング・システムに物理プロセッサとして示されるプロセッサの数。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。

**Physical memory and swap in megabytes:  
(Windows、AIX、HP-UX、Sun、Linux のみ)**

**TotalMemTotal**

メモリーのサイズ (MB)。

**FreeMem**

空き容量 (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)。

**Virtual memory in megabytes (Windows、AIX、HP-UX、および Sun のみ)**

**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**

使用可能な仮想メモリーの量 (MB)。

**Free** 空き仮想メモリーの量 (MB)。

**Operating system information (Windows、AIX、HP-UX、Sun、および Linux のみ)**

**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。

**Message queue information (AIX、HP-UX、および Linux のみ)**

**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のメッセージ・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。

**Shared memory information (AIX、HP-UX、および Linux のみ)**

**ShmMax**

システム全体での共用メモリー・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用メモリー・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用メモリー 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。

**Semaphore information: (AIX、HP-UX、および Linux のみ)**

**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 load information (Windows、AIX、HP-UX、Sun、および Linux のみ)**

**Short** 最小の期間。

**Medium**

中間の期間。

**Long** 長い期間。

**Disk information**

**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。

**関連タスク:**

- 「問題判別の手引き」の『待機中のロックの所有者を識別する』

**関連資料:**

- 「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』
- 422 ページの『GET DATABASE CONFIGURATION』
- 428 ページの『GET DATABASE MANAGER CONFIGURATION』

## db2perfc - データベース・パフォーマンス値のリセット

1 つ以上のデータベースのパフォーマンス値をリセットします。これは、Windows オペレーティング・システムの「パフォーマンス モニタ」で使用されます。

### 権限:

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 が開始して以来の累積値になります。しかしたいいていの場合、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、**db2perfc** 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。なお、**db2perfc** では、コマンドが実行されるときにリモート側で 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の値もリセットされます。

db2ResetMonitor API では、グローバルではなく、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。

```
db2perfc
```

次の例では、特定の DB2 データベースのパフォーマンス値をリセットします。

```
db2perfc dbalias1 dbalias2
```

次の例では、アクティブな DB2 DCS データベースすべてのパフォーマンス値をリセットします。

## db2perfc - データベース・パフォーマンス値のリセット

```
db2perfc -d
```

次の例では、特定の DB2 DCS データベースのパフォーマンス値をリセットします。

```
db2perfc -d dbalias1 dbalias2
```

### 関連資料:

- 「管理 API リファレンス」の『db2ResetMonitor - モニターのリセット』



## db2perfi - パフォーマンス・カウンター登録ユーティリティ

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。

### 権限:

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] [-u]
```

### コマンド・パラメーター:

- i** DB2 パフォーマンス・カウンターを登録します。
- u** DB2 パフォーマンス・カウンターの登録を解除します。

### 使用上の注意:

**db2perfi -i** コマンドは、以下を行います。

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

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

## db2perfr - パフォーマンス・モニター登録ツール

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したい場合にも、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。

### 権限:

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 NT 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一致していなければなりません。Windows NT 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る場合、リモート・パフォーマンス・モニターで 사용되는ユーザー名とパスワードの組み合わせをリセットしなければなりません。
- デフォルトの Windows パフォーマンス・モニター・ユーザー名 SYSTEM は、DB2 予約語なので使用できません。

## db2rbind - すべてのパッケージの再バインド

データベース内のパッケージを再バインドします。

権限:

以下のいずれかが必要です。

- *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
  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、古典的なバインド手法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。古典的なバインド手法を使用します。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。古典的なバインド手法は使用されません。

## db2rbind - すべてのパッケージの再バインド

### 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:

- 305 ページの『BIND』
- 603 ページの『PRECOMPILE』
- 642 ページの『REBIND』

## db2\_recon\_aid - 複数の表の RECONCILE

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。

### 権限:

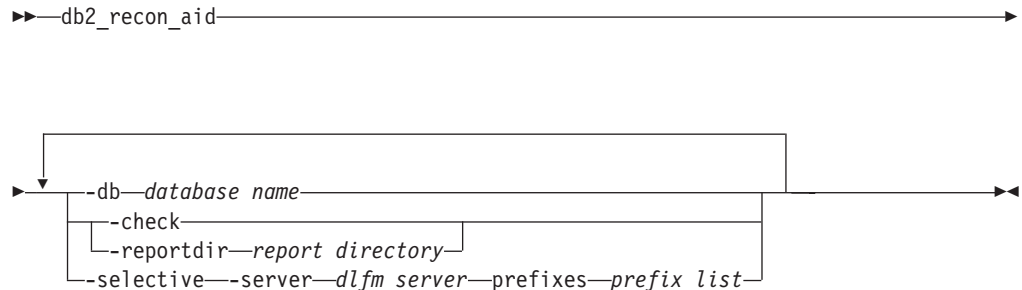
以下のどれかが必要です。

- 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、たとえば *prefix1:prefix2:prefix3* などです。

### コマンド・パラメーター:

#### **-db** *database name*

調整する必要のある DATALINK 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、<tbschema>.<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、

## db2\_recon\_aid - 複数の表の RECONCILE

- <tbschema> は表のスキーマです。
  - <tbyname> 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 は無視されません。

### -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。たとえば、次のようにします。

```
/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 dlmsvr.services.com
-prefixes /dlfsdir1/smith/

db2_recon_aid -db STAFF -reportdir /home/smith -selective -server
dlmsvr.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、ユーティリティは x:\sql\lib\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。たとえば、次のようになります。
  - 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ップ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。

## db2relocatedb - データベースの再配置

ユーザー提供の構成ファイルに指定されたように、データベースを名前変更、またはデータベース、またはデータベースの一部（たとえばコンテナおよびログ・ディレクトリ）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。

権限:

なし

コマンド構文:

▶▶ db2relocatedb -f *configFilename* ◀◀

コマンド・パラメーター:

**-f configFilename**

データベースを再配置するため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
...
```

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。データベース名が変更する場合は、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パスを指定します。このパスは、データベースが初めに作成された場所です。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・パスが変更されている場合、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。



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。古いコンテナと新規のコンテナの両方を指定する必要があります。複数コンテナ・パスが変更される場合、複数 CONT\_PATH 行が提供されます。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。

**注:** ブランク行またはコメント文字 (#) で開始され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が変更されている場合、インスタンスおよびデータベース・サポート・ファイルへの変更が行われるのを確認するために、このコマンドの実行前に以下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ているデータベースに属するファイル/デバイスをコピーします。パス名を変更する必要があります。ただし、データベース・ファイルの移動先のディレクトリー内にデータベースがすでにある場合、不用意に既存の sqlbdir ファイルを上書きしてしまって、既存のデータベースへの参照を除去する可能性があります。そのような事態になった場合、**db2relocatedb** ユーティリティーを使用することはできません。その場合、**db2relocatedb** の代わりにリダイレクト・リストアを使用します。
- 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、このツールは変更が必要なすべてのパーティションに対して実行される必要があります。パーティションごとにそれぞれ別個の構成ファイル (変更するパーティションの NODENUM 値が含まれる) を用意する必要があります。たとえば、データベースの名前を変更する場合は、すべてのパーティションが影響を受けることになり、各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、そのパーティション上で一度だけ **db2relocatedb** コマンドを実行することが必要です。

例:

#### 例 1

## db2relocatedb - データベースの再配置

データベース TESTDB の名前を PRODDB に、パス /home/db2inst1 にあるインスタンス db2inst1 で変更するには、以下の構成ファイルを作成します。

```
DB_NAME=TESTDB,PRODDB
DB_PATH=/home/db2inst1
INSTANCE=db2inst1
NODENUM=0
```

構成ファイルを relocate.cfg として保管し、以下のコマンドを使用して、データベース・ファイルへの変更を行います。

```
db2relocatedb -f relocate.cfg
```

### 例 2

データベース DATAB1 をパス /dbpath のインスタンス jsmith からインスタンス prodinst に移動するには、以下のようにします。

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

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

### 例 3

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

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

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

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

### 例 4

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

```
TS1
TS2_Cont0
TS2_Cont1
```

```

/databases/TESTDB/TS3_Cont0
/databases/TESTDB/TS4/Cont0
/Data/TS5_Cont0
/dev/rTS5_Cont1

```

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

データベースを移動する場合、/databases/TESTDB/db2inst1 ディレクトリーに存在するすべてのファイルは、/DB2/newinst ディレクトリーに移動する必要があります。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。

## db2relocatedb - データベースの再配置

```
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。

**注:** 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:

- 116 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設定

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。

権限:

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』

---

## db2rmicons - DB2 アイコンの除去

Linux デスクトップから DB2 のアイコンとフォルダーを除去します。このコマンドは、サポートされている Intel ベース Linux ディストリビューションの Gnome と KDE のデスクトップについてのみ使用できます。このコマンドは DB2DIR/bin ディレクトリーにあります。DB2DIR は /opt/IBM/db2/V8.1 です。また、インスタンス所有者の home ディレクトリーの /sqlllib/bin ディレクトリーにもあります。

### 権限:

以下のいずれかが必要です。

- 他のユーザーのためにこのコマンドを呼び出す場合: root 権限、または指定のユーザーの home ディレクトリーに対する書き込み権限
- 自分のデスクトップのためにこのコマンドを呼び出す場合: なし

### 必要な接続:

なし

### コマンド構文:

▶—db2rmicons—user\_name▶

### コマンド・パラメーター:

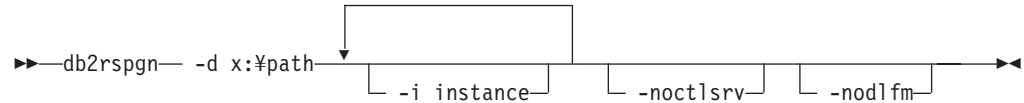
*user\_name*

デスクトップ・アイコンを除去するユーザー ID。

## db2rspgn - 応答ファイル生成プログラム (Windows)

**db2rspgn** コマンドは、Windows 上でのみ使用可能です。

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したい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・ジェネレーターについて (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。

権限:

以下のどれかが必要です。

- *sysadm*
- *sysctrl*

コマンド構文:

```
▶▶ db2sampl [path] [-k]
```

コマンド・パラメーター:

**path** SAMPLE データベースを作成するパスを指定します。 Windows の場合、パスは 1 文字のドライブ名になります。

パスを指定しない場合、SAMPLE はデフォルトのデータベース・パス (データベース・マネージャ構成ファイルの *dfidbpath* パラメーター) に作成されます。 UNIX ベースのシステムでは、デフォルトはインスタンス所有者の HOME ディレクトリーです。 Windows オペレーティング・システムでは、(DB2 がインストールされている) ルート・ディレクトリーになります。

**-k** 以下の SAMPLE 表について主キーを作成します。

表	主キー
-----	-----
DEPARTMENT	DEPTNO
EMPLOYEE	EMPNO
ORG	DEPTNUMB
PROJECT	PROJNO
STAFF	ID
STAFFG	ID (DBCS のみ)

注: パスは、このオプションの前に 指定しなければなりません。

使用上の注意:

このコマンドはサーバー・ノードからしか実行できません。 SAMPLE は、データベース・クライアントのみのノードに作成することはできません。

SAMPLE データベースは、データベース・マネージャ構成パラメーター *authentication* で指定したインスタンス認証タイプで作成されます。

SAMPLE 内の表の修飾子は、コマンドを実行しているユーザー ID が決めます。

SAMPLE がすでに存在する場合、**db2sampl** はコマンドを実行しているユーザー ID のために表を作成し、適切な特権を付与します。

関連資料:

- 428 ページの『GET DATABASE MANAGER CONFIGURATION』



## db2secv82 - DB2 オブジェクトの許可の設定

更新済みの DB2 Universal Database (UDB) 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。

権限:

- *sysadm*

必要な接続:

なし

コマンド構文:

►► db2secv82

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 db2secv82.exe 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。

**注:** このオプションが有効なのは、db2secv82.exe 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。

## 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。

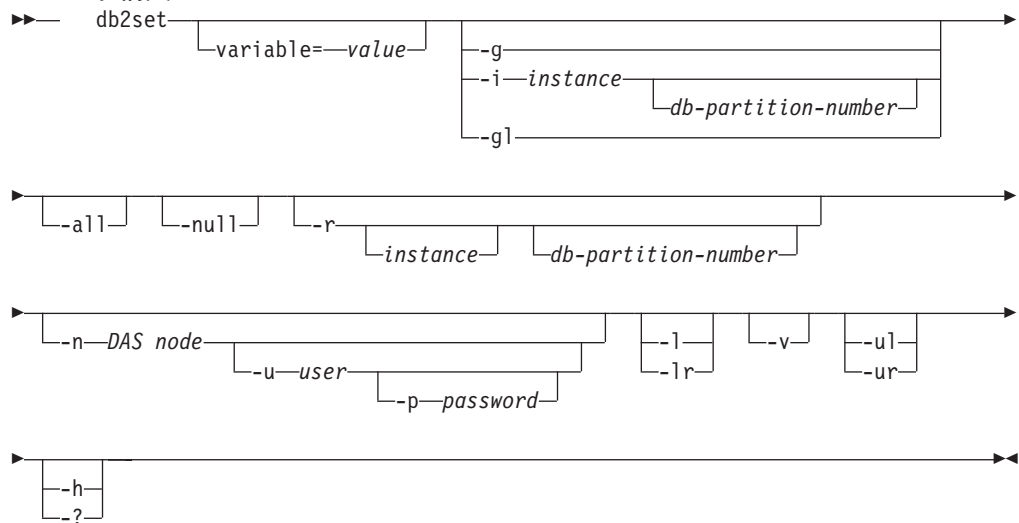
権限:

*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された後に有効になります。

**-g** グローバル・プロファイル変数にアクセスします。

**-i** インスタンス・プロファイルを指定して、現行値またはデフォルトの代わりに使用します。

**db-partition-number**

db2nodes.cfg ファイルでリストされる番号を指定します。

**-gl** LDAP に保管されたグローバル・プロファイル変数にアクセスします。このオプションが有効なのは、レジストリー変数 DB2\_ENABLE\_LDAP が YES に設定されている場合だけです。

**-all** 以下の形式で定義されたローカル環境変数のオカレンスをすべて表示します。

- 環境 ([e] で表示)

- ノード・レベル・レジストリー ([n] で表示)
- インスタンス・レベル・レジストリー ([i] で表示)
- グローバル・レベル・レジストリー ([g] で表示)

**-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・アタッチが存在していれば、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** すべてのインスタンス・プロファイルをリストします。

**-lr** サポートされているレジストリー変数をすべてリストします。

**-v** 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。

**注:** 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。

**注:** 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:**

- 定義済みプロファイル (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
```

- 定義済みグローバル変数をすべて表示するには、次のように入力します。

```
db2set -g
```

## db2set - DB2 プロファイル・レジストリー・コマンド

- 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=
```

- 変数 DB2COMM を TCPIP、IPXSPX、NETBIOS にグローバルに設定するには、次のように入力します。

```
db2set -g DB2COMM=TCPIP,IPXSPX,NETBIOS
```

- 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。

**注:** 設定の変更は、インスタンスが再始動され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。

## 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、UNIX ベース・システムでのみ利用できます。Windows オペレーティング・システム用のコマンドは **setup** です。

このユーティリティーは、DB2 インストール・メディアにあります。これを使って DB2 セットアップ・ウィザードを立ち上げ、インストールを定義して DB2 製品をインストールします。-r オプションを指定して呼び出すと、ユーザーの入力を必要としないインストールが実行されます。インストールの構成情報は応答ファイルから読み取られます。

### 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 使用情報を生成します。

### 関連資料:

- 259 ページの『setup - DB2 のインストール』

## 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。

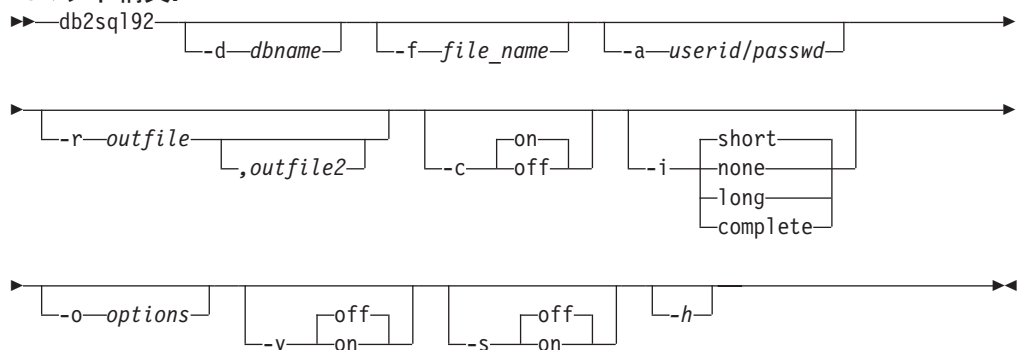
権限:

*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。

コメント文字は、各行の先頭に 2 つのハイフンを付けて `-- <コメント>` で表します。コメントを出力にも含めるときは、次のようにコメントにマークを付けます。 `--#COMMENT <コメント>`。

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:

- 24 ページの『db2batch - ベンチマーク・ツール』



## db2sqljbind - DB2 SQLJ プロファイル・バインド・プログラム

以前にカスタマイズされた SQLJ プロファイルをデータベースにバインドします。デフォルトでは、各分離レベルごとに 1 つずつで、4 つのパッケージが作成されます。カスタマイズに `-singlepkgname` オプションが使用されると、1 つのパッケージのみが作成され、`ISOLATION` オプションを使用する必要があります。このユーティリティーは、SQLJ アプリケーションがカスタマイズされた後に実行する必要があります。

### 権限:

以下のいずれかが必要です。

- `sysadm` または `dbadm` 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する場合は、スキーマに対する `CREATEIN` 特権
- パッケージが存在する場合は、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)

ユーザーには、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のに必要な特権もすべて必要になります。グループに付与された特権は、静的ステートメントの許可検査では使用されません。ユーザーに `sysadm` 権限があつてバインドを完成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:

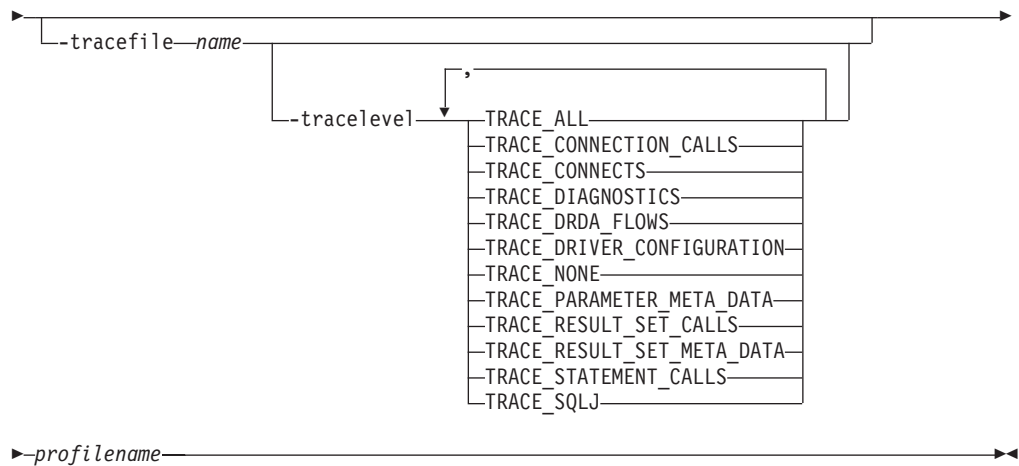
このコマンドは、データベース接続を確立します。

### コマンド構文:

```

▶ db2sqljbind [-help] [-url jdbc:db2://server:port/dbname] [-user username]
               [-password password] [-bindoptions "bind options"]
               [-staticpositioned {NO|YES}]

```



## 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に使用する名前を指定します。

**-password** *password*

ユーザー名のパスワードを指定します。

**-bindoptions** "*bind options*"

BIND オプションのリストを指定します。以下のオプションがサポートされています。詳細記述については、BIND コマンドを参照してください。

- DB2 (Windows および UNIX 版) の場合:
  - ACTION (ACTION RETAIN 以外)
  - BLOCKING
  - COLLECTION
  - DEGREE
  - EXPLAIN
  - EXPLSNAP
  - FEDERATED
  - FUNCPATH
  - INSERT
  - ISOLATION (db2sqljcustomize コマンドの -singlepkgname オプションを参照してください)
  - OWNER
  - QUALIFIER
  - QUERYOPT
  - REOPT
  - SQLERROR (SQLERROR CHECK 以外)

- SQLWARN
- STATICREADONLY
- VALIDATE
- VERSION
- Windows および UNIX 以外のサーバー上の DB2 の場合:
  - ACTION (ACTION RETAIN 以外)
  - BLOCKING
  - COLLECTION
  - DBPROTOCOL
  - DEGREE
  - EXPLAIN
  - IMMEDIATEWRITE
  - ISOLATION (db2sqljcustomize コマンドの -singlepkgname オプションを参照してください)
  - OPTHINT
  - OWNER
  - PATH
  - QUALIFIER
  - RELEASE
  - REOPT
  - SQLERROR
  - VALIDATE
  - VERSION

### **-staticpositioned**

値は YES および NO です。デフォルト値は NO です。db2sqljbind に指定する値は、前に db2sqljcustomize に指定した値と一致していなければなりません。

### **-tracefile** *name*

トレースを使用可能にし、トレース情報の出力ファイルを識別します。IBM サービス技術者の指示以外では、使用しないでください。

### **-tracelevel**

トレースのレベルを識別します。-tracelevel が省略された場合は、TRACE\_ALL が使用されます。

### **profilename**

SQLJ プロファイル・ファイルの相対または絶対名を指定します。SQLJ ファイルが Java ファイルに変換されるとき、ファイルに入っている SQL 操作に関する情報は、プロファイルと呼ばれる、SQLJ が生成したリソース・ファイルに保管されます。プロファイルは、元の入力ファイル名に接尾部 \_SJProfileN (N は整数) が付いた名前によって識別されます。拡張子は .ser です。プロファイル名を指定するとき、.ser 拡張子はあってもなくてもかまいません。

## db2sqljbind - DB2 SQLJ プロファイル・バインド・プログラム

複数のファイルをバインドして DB2 上の単一のパッケージにすることができます。コマンド行から db2sqljbind を呼び出すとき、プロファイル・リストを提供することができます。別の方法として、拡張子 .grp のファイル内で各行に 1 つのプロファイルをリストして、その .grp ファイル名を db2sqljbind に提供することもできます。バインド・プログラムはすべてのステートメントをバインドします。プロファイルのグループをまとめてバインドする場合は、同じ順序の同じファイル・リストを使用して、それらのプロファイルが事前に一緒にカスタマイズされている必要があります。

### 例:

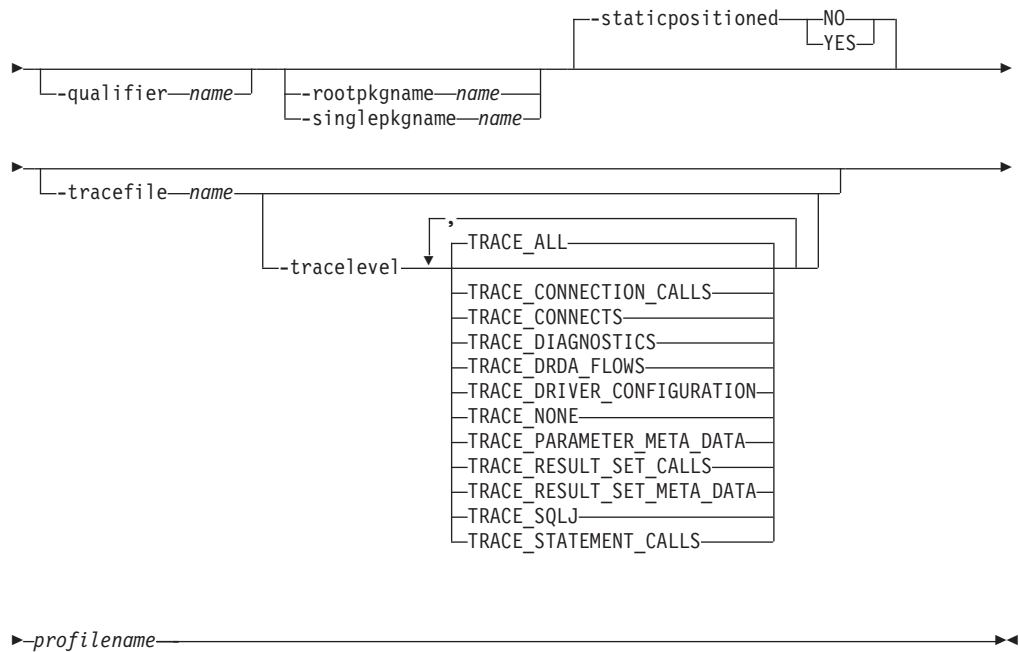
```
db2sqljbind -user richler -password mordecai
            -url jdbc:db2://server:50000/sample -bindoptions "EXPLAIN YES"
            pgmname_SJProfile0.ser
```

### 関連資料:

- 305 ページの『BIND』
- 227 ページの『db2sqljcustomize - DB2 SQLJ プロファイル・カスタマイザー』
- 234 ページの『db2sqljprint - DB2 SQLJ プロファイル・プリンター』
- 260 ページの『sqlj - DB2 SQLJ 変換プログラム』



## db2sqljcustomize - DB2 SQLJ プロファイル・カスタマイザー



### 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。オンライン・チェックまたは自動バインドが使用可能な場合に必要です。

**-datasource jndiName**

オンライン検査または自動バインディングのためのデータベース接続の確立に必要な、JNDI 登録済み DataSource 名を指定します。この登録名は、タイプ 4 接続用に構成された Universal Driver データ・ソースに対応していなければなりません。

**-user username**

データベースに接続するときに使用する名前を指定します。-datasource オプションを指定した場合、このパラメーターは不要です。

**-password password**

ユーザー名のパスワードを指定します。-datasource オプションを指定した場合、このパラメーターは不要です。

**-automaticbind**

**db2sqljbind** コマンドが、ターゲット・データベースにパッケージを作成するために、自動的に呼び出されるかどうかを決定します。有効な値は YES または NO です。デフォルトは YES です。使用可能な場合は、-url も指定する必要があります。

**-bindoptions "bind options"**

BIND オプションのリストを指定します。以下のオプションがサポートされています。詳細記述については、BIND コマンドを参照してください。

- DB2 (Linux、UNIX、および Windows 版) の場合:

## db2sqljcustomize - DB2 SQLJ プロファイル・カスタマイザー

- ACTION (ACTION RETAIN 以外)
- BLOCKING
- COLLECTION
- DEGREE
- EXPLAIN
- EXPLSNAP
- FEDERATED
- FUNCPATH
- INSERT
- ISOLATION (下記の -singlepkgname オプションを参照してください)
- OWNER
- QUALIFIER
- QUERYOPT
- REOPT
- SQLERROR (SQLERROR CHECK 以外)
- SQLWARN
- STATICREADONLY
- VALIDATE
- VERSION

• Linux、UNIX、および Windows 以外のサーバー上にある DB2 の場合:

- ACTION (ACTION RETAIN 以外)
- BLOCKING
- COLLECTION
- DBPROTOCOL
- DEGREE
- EXPLAIN
- IMMEDIATEWRITE
- ISOLATION (db2sqljcustomize コマンドの -singlepkgname オプションを参照してください)
- OPTHINT
- OWNER
- PATH
- QUALIFIER
- RELEASE
- REOPT
- SQLERROR
- VALIDATE
- VERSION

### **-collection** *name*

デフォルトのコレクション ID を指定します。指定されていない場合は、NULLID が使用されます。コレクションが SET CURRENT PACKAGESET ステートメントによって明示的に設定されていない場合は、実行時にデフォルト・コレクションが使用されます。

### **-onlinecheck**

オンライン・チェックが、-url オプションで指定されたデータベースを使用

して実行されるかどうかを決定します。有効な値は YES または NO です。デフォルトは `-url` が指定されている場合は YES です。そうでない場合はデフォルトは NO です。

#### **-pkgversion** *versionname*

直列化プロファイル用にサーバーでパッケージをバインディングするときに使用する、パッケージ・バージョン名を指定します。このバージョン名は、サーバーでパッケージの手動バージョン・マッチングを実行するときのために、直列化プロファイルに保管されます。ランタイム・バージョン検査は、バージョン名ではなく、整合性トークンに基づいています。整合性トークンに基づいてバージョン名を自動的に生成するには、このパラメーターに値 AUTO (すべて大文字) を入力してください。

#### **-qualifier** *name*

オンライン・チェックのデフォルト動的修飾子を提供します。提供された値は、オンライン・チェックが始まる前に、SET CURRENT SQLID の呼び出しの作成に使用されます。デフォルトは、動的 SQL のデフォルト修飾子です。動的デフォルト修飾子 (オンライン・チェックのために DB2 が使用) が、静的デフォルト修飾子 (ランタイムに DB2 が使用) と異なるため、`-qualifier` オプションの使用は、SQL ステートメントに非修飾のオブジェクトがある場合に、正しいオブジェクトがオンライン・チェックされたことを確認します。このオプションに提供された値は、バインドで自動的に使用されません。QUALIFIER BIND オプションは、`-bindoptions` ストリングに明示的に提供される必要があります。逆に、`-bindoptions` ストリングに提供された値は、`-qualifier` オプションが同様に提供されていない場合、オンライン・チェックには使用されません。

#### **-rootpkgname** *name*

SQLJ バインド・プログラムによって生成されるパッケージのルート名を指定します。このオプションが指定されていない場合は、ルート名はプロファイルの名前から派生します。最大長は 7 文字です。4 つの最終パッケージ名 (分離レベルごとに 1 つ) を作成するために、数字 1、2、3、および 4 がルート名に付加されます。

**注:** パッケージのルート名を指定することによって、パッケージが確実にユニークな名前を持つようにすることができます。既存のパッケージと同じ名前を持つパッケージを作成すると、既存のパッケージが上書きされます。

#### **-singlepkgname** *name*

SQLJ バインド・プログラムによって生成されるパッケージ名を指定します。最大長は 8 文字です。このオプションには、`-bindoptions` フラグで指定される ISOLATION バインド・フラグが必要です。これは、単一ランザクション分離レベルを使用するアプリケーションにのみ使用してください。

**注:** パッケージ名を指定することによって、名前を確実にユニークなものにすることができます。既存のパッケージと同じ名前を持つパッケージを作成すると、既存のパッケージが上書きされます。

#### **-staticpositioned**

更新ステートメントと同じプログラムでイテレーターを宣言する場合、このオプションを指定すると、位置決めされた更新 (UPDATE) が、動的に準備



されたステートメントではなく、静的にバインドされたステートメントを使用して行われるようになります。値は YES および NO です。デフォルト値は NO です。DB2 上の単一のパッケージに結合するために、複数のファイルをまとめてカスタマイズすることができます。この場合、結果パッケージより前の位置で宣言されたカーソルを参照する位置決めされた更新または削除ステートメントはすべて、`-staticpositioned` が YES に設定されていれば、静的に実行されます (同じソース・ファイルからのステートメントでない場合でも)。ただし、複数のソース・ファイルを結合するとき、この機能を有効にするには、プロファイル・リストに示される順序が重要です。更新または削除ステートメントよりも必ず前にカーソル・セクションが処理されるようにしなければなりません。そうするには、結果イテレーターを参照する位置決めされた更新または削除ステートメントを含むプロファイルよりも前に、照会ステートメントを含むプロファイルを一覧にしなければなりません。プロファイルを一覧にしない場合、`-staticpositioned` を YES に設定しても、パフォーマンスは改善されません。

#### **-tracefile** *name*

トレースを使用可能にし、トレース情報の出力ファイルを識別します。IBM サービス技術者の指示以外では、使用しないでください。

#### **-tracelevel**

トレースのレベルを識別します。`-tracelevel` が省略された場合は、TRACE\_ALL が使用されます。

#### *profilename*

SQLJ プロファイル・ファイルの相対または絶対名を指定します。SQLJ ファイルが Java ファイルに変換されるとき、ファイルに入っている SQL 操作に関する情報は、プロファイルと呼ばれる、SQLJ が生成したリソース・ファイルに保管されます。プロファイルは、元の入力ファイル名に接尾部 `_SJProfileN` (N は整数) が付いた名前によって識別されます。拡張子は `.ser` です。プロファイル名を指定するとき、`.ser` 拡張子はあってもなくてもかまいません。複数のファイルをまとめてカスタマイズしてバインドし、DB2 上の単一のパッケージにすることができます。コマンド行から **db2sqljcustomize** を呼び出すとき、プロファイル・リストを提供することができます。別の方法として、拡張子 `.grp` のファイル内で各行に 1 つのプロファイルを一覧にして、その `.grp` ファイル名を **db2sqljcustomize** に提供することもできます。カスタマイザーは、リスト内のプロファイルが単一の DB2 パッケージから実行されるように準備し、バインド・プログラムが暗黙的に呼び出されて、すべてのステートメントをバインドします。複数のプロファイルを結合するとき、`-rootpkgname` または `-singlepkgname` オプションのいずれかでパッケージ名を指定する必要があります。バインド・プログラムを別個に呼び出す場合、プロファイルを一覧のパッケージに結合するためには、同じファイル・リストを (同じ順序で) 提供する必要があります。

#### 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:

-staticpositioned YES オプションの使用の意味

SQLJ は、メソッド間で iterators を変数として渡すことを許可します。変数として渡され、位置決めされた更新 (UPDATE または DELETE) 用に使用される iterator は、実行時にのみ識別されます。また、同じ SQLJ 位置決めされた更新ステートメントは、実行時に異なる iterator で使用できます。静的に実行するために、SQLJ カスタマイザーが、位置決めされた更新ステートメントを準備する場合、どの照会がどの位置決めされた更新ステートメントに属するかを判別する必要があります。

SQLJ カスタマイザーは、照会ステートメントと位置決めされた UPDATE ステートメントとの間をマップする iterator のクラスを使用して、これを行います。iterator クラスが照会ステートメントと位置決めされた更新の間のユニーク・マッピングを提供しない場合、SQLJ カスタマイザーはどの照会と位置決めされた更新ステートメントが、互いに正確に属するかを判別できません。位置決めされた更新は、それぞれの一致する照会ごとに一度準備される必要があります (すなわち、同じ iterator クラスを位置決めされた更新として使用する照会ステートメント)。照会ステートメントから位置決めされた更新ステートメントまでのユニーク・マッピングがない場合は、これはバインド・エラーになります。以下のコード・フラグメントは、この点を表示します。

```
#sql iterator GeneralIter implements ForUpdate ( String );

public static void main( String args[] )
{
    ...
    GeneralIter iter = null;
    #sql [conn] iter = { SELECT CHAR_COL1 FROM TABLE1 };
    doUpdate( iter );
    ...
    #sql [conn] iter = { SELECT CHAR_COL2 FROM TABLE2 };
    ...
}

public static void doUpdate( GeneralIter iter )
{
    #sql [conn] { UPDATE TABLE1 ... WHERE CURRENT OF :iter };
}
```

この例では、iterator クラスは 1 つだけ定義されています。iterator の 2 つのインスタンスが作成され、それぞれが、異なる表からデータを検索する、異なる SELECT ステートメントに関連付けられます。iterator はメソッド doUpdate に変数として渡されるため、位置決めされた UPDATE に、どの iterator インスタンスが使用されるかは、実行時までわかりません。DB2 バインド処理は、両方の照会を位置決めされた UPDATE にバインドしようとしませんが、2 度目の SELECT でバインド・エラーが発生します。

上記のようなプログラムで、バインド時のエラーを避けるには、DB2 BIND オプション SQLERROR(CONTINUE) を指定します。しかし、より良い方法は、プログラムを書くことです。iterator クラス、照会、および位置決めされた UPDATE または DELETE 間のユニーク・マッピングがあるからです。下記の例は、これを行う方法を例示しています。このコーディングのメソッドでは、各 iterator クラスは 1 つの iterator インスタンスにのみ関連付けられています。そのため、DB2 バインド処理は常に位置決めされた UPDATE ステートメントを、正しい照会と関連付けます。

```
#sql iterator Table1Iter implements ForUpdate ( String );
#sql iterator Table2Iter ( String );

public static void main ( String args[] )
{
    ...
    Table1Iter iter1 = null;
    #sql [conn] iter1 = { SELECT CHAR_COL1 FROM TABLE1 };
    Table2Iter iter2 = null;
    #sql [conn] iter2 = { SELECT CHAR_COL2 FROM TABLE2 };
    ...
    updateTable1( iter1 );
}

public static void updateTable1 ( Table1Iter iter )
{
    #sql [conn] { UPDATE TABLE1 ... WHERE CURRENT OF :iter };
}
```

**db2prof** は DB2 バージョン 8 では推奨オプションではありませんが、**db2sqljcustomize** の代わりに指定することはできます。**db2prof** は DB2 バージョン 9 ではサポートされません。

### 関連資料:

- 305 ページの『BIND』
- 234 ページの『db2sqljprint - DB2 SQLJ プロファイル・プリンター』
- 223 ページの『db2sqljbind - DB2 SQLJ プロファイル・バインド・プログラム』

## db2sqljprint - DB2 SQLJ プロファイル・プリンター

DB2 のカスタマイズされたバージョンのプロファイルの内容をプレーン・テキストで印刷します。

権限:

なし

必要な接続:

なし

コマンド構文:

▶ db2sqljprint *profilename* ▶  
    └─db2profp┘

コマンド・パラメーター:

*profilename*

SQLJ プロファイル・ファイルの相対または絶対名を指定します。SQLJ ファイルが Java ファイルに変換されるとき、ファイルに入っている SQL 操作に関する情報は、プロファイルと呼ばれる、SQLJ が生成したリソース・ファイルに保管されます。プロファイルは、元の入力ファイル名に接尾部 *\_SJProfileN* (*N* は整数) が付いた名前によって識別されます。拡張子は *.ser* です。プロファイル名を指定するとき、*.ser* 拡張子はあってもなくてもかまいません。

例:

```
db2sqljprint pgmname_SJProfile0.ser
```

使用上の注意:

現在、**db2profp** は **db2sqljprint** の代わりに指定できます。ただし、**db2profp** は DB2 バージョン 9 では使用すべきではありません。

関連資料:

- 227 ページの『db2sqljcustomize - DB2 SQLJ プロファイル・カスタマイザー』
- 223 ページの『db2sqljbind - DB2 SQLJ プロファイル・バインド・プログラム』

---

## 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・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。

### 関連資料:

- 744 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。

**関連資料:**

- 752 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。

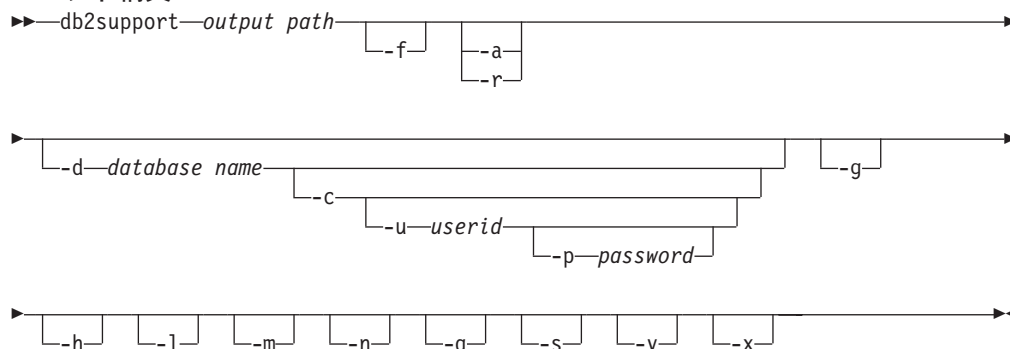
### 権限:

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:



### 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。

#### **-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ジャーを経由して **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。

#### **-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。

#### **-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。

**-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。

**-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。

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

ユーザー ID のパスワードを指定します。

**-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。

**-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。

**-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。

**-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。

**-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。

**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・アタッチ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。たとえば、オペレーティング・システム情報またはハードウェア情報 (-s オプション)、および診断ディレクトリーからのファイル



(DIAGPATH) は、 db2support ツールを実行しているローカル・マシンから、そしてバッファ・プール、データベース構成、および表スペース情報などのデータは、データベースが物理的に常駐しているマシンから得られます。

### 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停止またはモニターしたり、サテライトの構成情報 (たとえば、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。

#### 権限:

なし

#### 必要な接続:

なし

#### コマンド構文:

```
►► db2sync [ -t | -s application_version | -g ]
```

#### コマンド・パラメーター:

- t** 管理者がサテライトのアプリケーション・バージョンまたは同期化認証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。

## 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。**db2sysstray** は、ローカル DB2 インスタンスの状況をモニターする Windows システム・トレイ・ツールです。システム・トレイの中のアイコンは、モニターされるインスタンスの状態に基づいて変化します。**db2sysstray** は、ESE インスタンスの状況を 1 つ以上のデータベース・パーティションが停止するときに停止として、すべてのデータベース・パーティションが開始されるときに開始として表示します。インスタンスはこのツールを使用して開始および停止できます。**db2sysstray** を使用した ESE インスタンスの開始または停止は、インスタンス内のすべてのデータベース・パーティションを開始または停止します。**db2sysstray** は、接続しているアプリケーションがない場合、インスタンスの停止だけを行います。

**db2sysstray** は、Windows プラットフォームでのみ使用可能です。

### 権限:

**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:

```
►►—db2sysstray—┐  
                  └instance-name┘
```

### コマンド・パラメーター:

*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。

## 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:

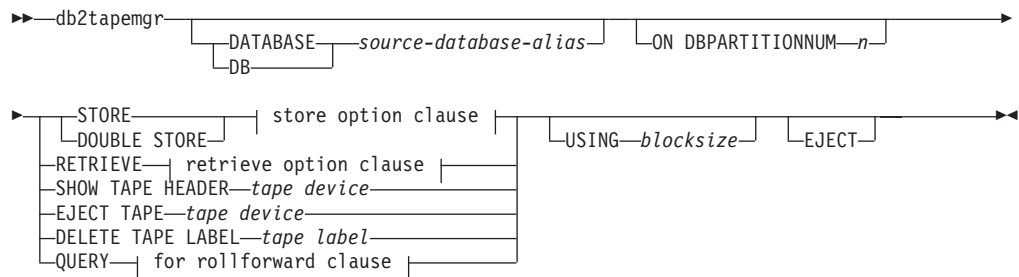
権限:

以下のいずれか。

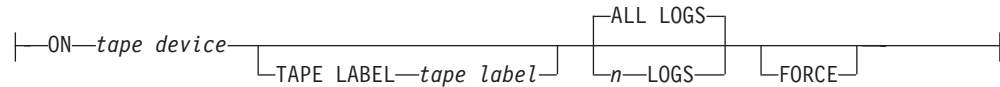
- *sysadm*
- *sysctrl*
- *sysmaint*

必要な接続:

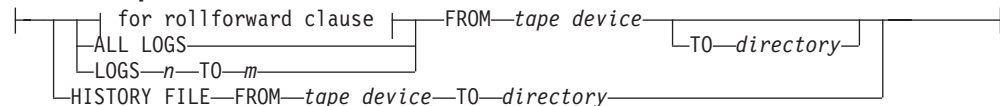
コマンド構文:



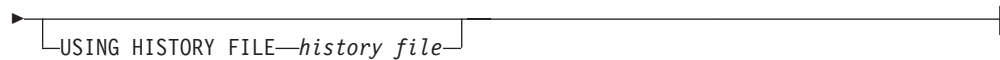
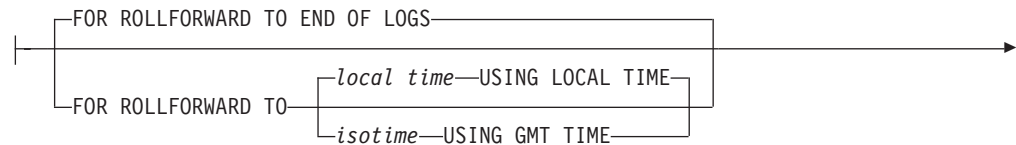
store option clause:



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***TO** *directory*

## db2tapemgr - テープ上のログ・ファイルの管理

**RETRIEVE ALL LOGS** または **LOGS *n* TO *m***

**FROM** *tape device*

**TO** *directory*

**RETRIEVE HISTORY FILE**

履歴ファイルを取り出します。

**FROM** *tape device*

**TO** *directory*

**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:

使用上の注意:

---

## 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。

権限:

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:

- 551 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、トレース機能を使用可能にすると、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。

### 権限:

UNIX ベースのシステムで 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 ベースのシステムで DB2 Administration Server のトレースを行う場合は、以下の権限が必要です。

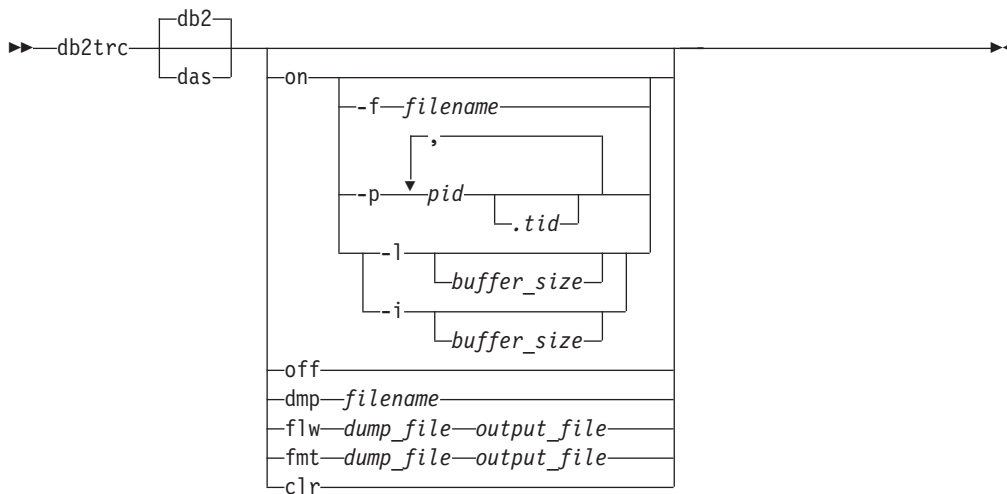
- *dasadm*

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。

**注:** 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。

**-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使用可能にします。tid を指定する場合は、必ずピリオド (.) を含めてください。最高で 5 つの pid.tid の組み合わせがサポートされています。

たとえば、プロセス 10、20、および 30 に対してトレースを使用可能にする場合は、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使用可能にする場合は、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。たとえば、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。

**注:** なお、バッファ・サイズは 2 の累乗でなければなりません。

**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。

**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
```

### 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。たとえば、次のようにします。

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

**clr** トレース・バッファ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・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。

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。

## 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 (最大 8 文字)

#### **-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 (ファイル名の長さ <= 80) に出力します。

#### **-r**

取り消しを実行します。

### 使用上の注意:

#### 注:

1. 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。

## 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。

権限:

*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:

このツールは、バージョン 5 以前の DB2 で稼動するデータベースで作成された索引がなければ、使用する必要はありません。

**注:** 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。

## 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。

**権限:**

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すで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:**

- 357 ページの『CREATE DATABASE』

## db2updv8 - バージョン 8 の現行レベルへのデータベースの更新

このコマンドは、以下の組み込みルーチンを使用可能にすることによって、現行レベルをサポートできるようにデータベース内のシステム・カタログを更新します。

- HEALTH\_DBM\_INFO
- HEALTH\_DBM\_HI
- HEALTH\_DBM\_HI\_HIS
- HEALTH\_DB\_INFO
- HEALTH\_DB\_HI
- HEALTH\_DB\_HI\_HIS
- HEALTH\_TBS\_INFO
- HEALTH\_TBS\_HI
- HEALTH\_TBS\_HI\_HIS
- HEALTH\_CONT\_INFO
- HEALTH\_CONT\_HI
- HEALTH\_CONT\_HI\_HIS
- GET\_WRAP\_CFG\_C
- LIST\_FILES\_C
- ODBC\_DISCOVERY
- HEALTH\_HI\_REC
- EXEC\_DB2\_SCRIPT
- HEALTH\_HI\_REC
- EXEC\_DB2\_SCRIPT

権限:

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:

```
db2updv8 -d database-name [-u userid -p password] [-h]
```

コマンド・パラメーター:

**-d database-name**

更新するデータベースの名前を指定します。

**-u userid**

ユーザー ID を指定します。

**-p password**

ユーザーのパスワードを指定します。

**-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。

### 例:

現行レベル (フィックスパックまたは新しいバージョン) をインストールした後、以下のコマンドを実行して、サンプル・データベース内のシステム・カタログを更新します。

```
db2updv8 -d sample
```

### 使用上の注意:

1. このツールは、DB2 バージョン 8.1.2 以降を実行しているデータベースでのみ使用できます。このコマンドを複数回発行した場合、エラーは報告されず、カタログのそれぞれの更新内容が 1 回だけ適用されます。
2. 新しい組み込み関数を使用可能にするには、すべてのアプリケーションをデータベースから切断し、データベースが活動状態になっていれば非活動状態にする必要があります。

指定したデータベースについて DB2 WebSphere MQ 関数を使用不可にします。

以下のいずれかが必要です。

- *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)

```

▶▶—disable_MQFunctions—-n—database—-u—userid—-p—password————▶
▶
└─-v┐
    │all│
    │opc│
    └─lpc┘

```

**-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、使用不可化によってスキーマ `db2mq` が削除されます。`1pc` を指定すると、使用不可化によってスキーマ `db2mq1c` が削除されます。`all` を指定すると、使用不可化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、使用不可化のデフォルトは `all` オプションになります。

以下の例では、データベース SAMPLE について DB2MQ と DB2MQ1C の関数を使用不可にします。

```
disable MQFunctions -n sample -u user1 -p password1
```

関連概念:



- 「アプリケーション開発ガイド クライアント・アプリケーションのプログラミング」の『MQSeries 使用可能性』
- 「アプリケーション開発ガイド クライアント・アプリケーションのプログラミング」の『WebSphere MQ 機能の概要』
- 「アプリケーション開発ガイド クライアント・アプリケーションのプログラミング」の『WebSphere MQ メッセージング』
- 「アプリケーション開発ガイド クライアント・アプリケーションのプログラミング」の『WebSphere MQ 関数を使用したメッセージ送信』
- 「アプリケーション開発ガイド クライアント・アプリケーションのプログラミング」の『WebSphere MQ 関数を使用したメッセージ取り出し』
- 「アプリケーション開発ガイド クライアント・アプリケーションのプログラミング」の『WebSphere MQ アプリケーション間の接続』
- 「アプリケーション開発ガイド クライアント・アプリケーションのプログラミング」の『WebSphere MQ 関数を使用した要求 / 応答通信』
- 「アプリケーション開発ガイド クライアント・アプリケーションのプログラミング」の『WebSphere MQ 関数を使用したパブリッシュ / サブスクライブ』
- 「IBM DB2 Information Integrator Application Developer's Guide」の『DB2 内で WebSphere MQ 機能を使用する方法』

#### 関連タスク:

- 「アプリケーション開発ガイド アプリケーションの構築および実行」の『DB2 WebSphere MQ 関数のセットアップ』

#### 関連資料:

- 「SQL Administrative Routines」の『MQSEND スカラー関数』
- 「SQL Administrative Routines」の『MQRECEIVE スカラー関数』
- 「SQL Administrative Routines」の『MQREAD スカラー関数』
- 「SQL Administrative Routines」の『MQPUBLISH スカラー関数』
- 「SQL Administrative Routines」の『MQSUBSCRIBE スカラー関数』
- 「SQL Administrative Routines」の『MQUNSUBSCRIBE スカラー関数』
- 「SQL Administrative Routines」の『MQREADALL 表関数』
- 「SQL Administrative Routines」の『MQRECEIVEALL 表関数』
- 「SQL Administrative Routines」の『MQREADCLOB スカラー関数』
- 「SQL Administrative Routines」の『MQRECEIVECLOB スカラー関数』
- 「SQL Administrative Routines」の『MQREADALLCLOB 表関数』
- 「SQL Administrative Routines」の『MQRECEIVEALLCLOB 表関数』
- 147 ページの『db2mqdsn - MQ Listener』
- 256 ページの『enable\_MQFunctions』

## enable\_MQFunctions

指定したデータベースについて DB2 WebSphere MQ 関数を使用可能にし、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。

### 権限:

以下のいずれかが必要です。

- *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** 使用可能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 がすで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、使用可能化によってスキーマ *db2mq* が作成されます。*1pc* を指定すると、使用可能化によってスキーマ *db2mq1c* が作成されます。*all* を指定すると、使用可能化によって

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、使用可能化のデフォルトは 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使用可能にします。ユーザーはデータベース SAMPLE に接続します。

```
enable_MQFunctions -n sample -u user1 -p password1
```

次の例では、ユーザーはデータベース SAMPLE に接続します。この例は、スキーマ DB2MQ1C によって DB2MQ1C 関数を作成します。

```
enable_MQFunctions -n sample -u user1 -p password1 -v 1pc
```

#### 使用上の注意:

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

- WebSphere MQ と WebSphere Application Messaging Interface (AMI) がインストールされていることと、WebSphere MQ のバージョンが 5.1 以上であることを確認します。
- 環境変数 \$AMT\_DATA\_PATH が定義されていることを確認します。
- トランザクション MQ UDF を使用する場合は、フェデレーテッド操作にデータベースが構成されていることを確認します。そのためには、以下のコマンドを実行します。

```
update dbm cfg using federated yes
```

- DB2PATH の cfg サブディレクトリーに移動します。

UNIX の場合:

- db2set を使用して、DB2ENVLIST に AMT\_DATA\_PATH を追加します。
- UDF 実行に関連するユーザー・アカウントが mqm グループのメンバーであることを確認します。
- このコマンドを呼び出すユーザーが mqm グループのメンバーであることを確認します。

**注:** AIX 4.2 は、MQSeries 5.2 でサポートされていません。

#### 関連概念:

- 「アプリケーション開発ガイド クライアント・アプリケーションのプログラミング」の『MQSeries 使用可能性』
- 「アプリケーション開発ガイド クライアント・アプリケーションのプログラミング」の『WebSphere MQ 機能の概要』
- 「アプリケーション開発ガイド クライアント・アプリケーションのプログラミング」の『WebSphere MQ メッセージング』
- 「アプリケーション開発ガイド クライアント・アプリケーションのプログラミング」の『WebSphere MQ 関数を使用したメッセージ送信』

- 「アプリケーション開発ガイド クライアント・アプリケーションのプログラミング」の『WebSphere MQ 関数を使用したメッセージ取り出し』
- 「アプリケーション開発ガイド クライアント・アプリケーションのプログラミング」の『WebSphere MQ アプリケーション間の接続』
- 「アプリケーション開発ガイド クライアント・アプリケーションのプログラミング」の『WebSphere MQ 関数を使用した要求 / 応答通信』
- 「アプリケーション開発ガイド クライアント・アプリケーションのプログラミング」の『WebSphere MQ 関数を使用したパブリッシュ / サブスクライブ』
- 「IBM DB2 Information Integrator Application Developer's Guide」の『DB2 内で WebSphere MQ 機能を使用する方法』

### 関連タスク:

- 「アプリケーション開発ガイド アプリケーションの構築および実行」の『DB2 WebSphere MQ 関数のセットアップ』

### 関連資料:

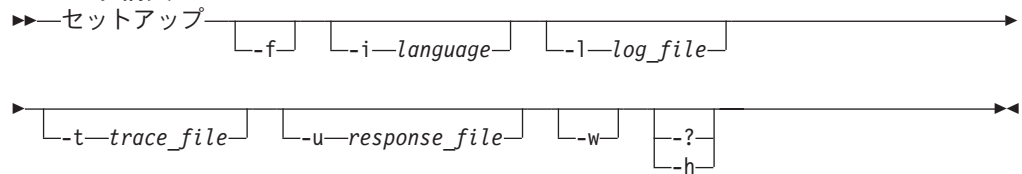
- 「SQL Administrative Routines」の『MQSEND スカラー関数』
- 「SQL Administrative Routines」の『MQRECEIVE スカラー関数』
- 「SQL Administrative Routines」の『MQREAD スカラー関数』
- 「SQL Administrative Routines」の『MQPUBLISH スカラー関数』
- 「SQL Administrative Routines」の『MQSUBSCRIBE スカラー関数』
- 「SQL Administrative Routines」の『MQUNSUBSCRIBE スカラー関数』
- 「SQL Administrative Routines」の『MQREADALL 表関数』
- 「SQL Administrative Routines」の『MQRECEIVEALL 表関数』
- 「SQL Administrative Routines」の『MQREADCLOB スカラー関数』
- 「SQL Administrative Routines」の『MQRECEIVECLOB スカラー関数』
- 「SQL Administrative Routines」の『MQREADALLCLOB 表関数』
- 「SQL Administrative Routines」の『MQRECEIVEALLCLOB 表関数』
- 147 ページの『db2mqIsn - MQ Listener』
- 254 ページの『disable\_MQFunctions』

## 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可能です。UNIX ベース・システムの場合のこのコマンドは、**db2setup** 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。-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:

- 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。
- 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。
- 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。
- w** インストールが終了するまで親プロセスを有効なまま保持します。
- , -h** 使用情報を生成します。

### 関連資料:

- 219 ページの『db2setup - DB2 のインストール』

## sqlj - DB2 SQLJ 変換プログラム

SQLJ ソース・ファイルを、Java ソース・ファイルおよびゼロ以上の SQLJ 直列化プロファイルに変換します。デフォルトでは、Java ソース・ファイルもコンパイルされます。

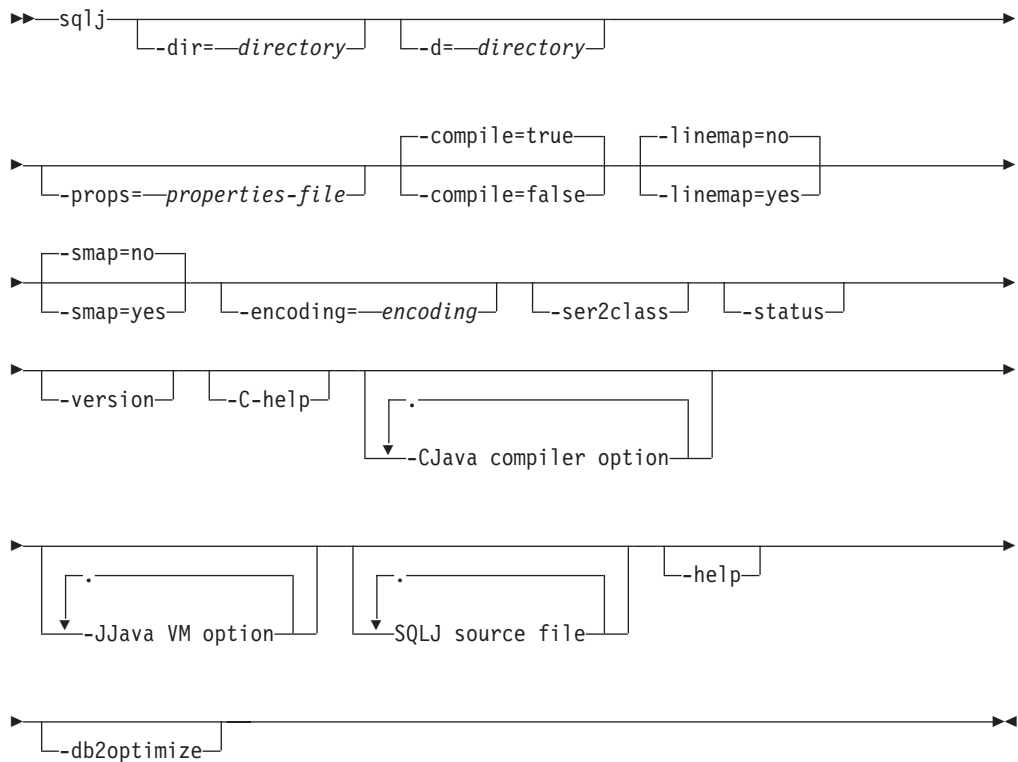
権限:

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:

**-dir=directory**

変換プログラムによって生成された .java ファイルを SQLJ が配置するときの配置先のディレクトリーの名前を指定します。デフォルトのディレクトリーは、SQLJ ソース・ファイルが入っているディレクトリーです。

**-d=directory**

生成された .ser ファイルが入っているルート・ディレクトリー。デフォルトは、現行ディレクトリーです。このオプションは、Java コンパイラーにも渡されます。

**-props=***properties-file*

SQLJ 変換プログラムがオプションのリストを取得するときの取得元のファイルの名前を指定します。

**-compile=**

**true** これは、生成された Java ファイルのコンパイルをオンにします。これはデフォルトです。

**false** これは、生成された Java ファイルのコンパイルをオフにします。

**-linemap=**

Java 例外の中の行番号を、SQLJ ソース・ファイル (.sqlj ファイル) 内の行番号に合わせるか、SQLJ 変換プログラムによって生成される Java ソース・ファイル (.java ファイル) 内の行番号に合わせるかを指定します。

**no** Java 例外の中の行番号を、Java ソース・ファイル内の行番号に合わせます。これはデフォルトです。

**yes** Java 例外の中の行番号を、SQLJ ソース・ファイル内の行番号に合わせます。

**-smap=**

SQLJ 変換プログラムによって、それぞれの SQLJ ソース・ファイルごとにソース・マップ (SMAP) ファイルを生成するかどうかを指定します。

SMAP ファイルは、一部の Java 言語デバッグ・ツールが使用するファイルです。このファイルには、SQLJ ソース・ファイル内の行番号と、SQLJ 変換プログラムによって生成される Java ソース・ファイル内の行番号との間のマッピングが記述されています。このファイルのコード化スキームは、Unicode UTF-8 です。そのフォーマットについては、[www.jcp.org](http://www.jcp.org) から入手できる Original Java Specification Request (JSR) 45 に説明があります。

**no** SMAP ファイルを生成しません。これはデフォルトです。

**yes** SMAP ファイルを生成します。SMAP ファイルの名前は、*SQLJ-source-file-name.java.smap* になります。SQLJ 変換プログラムは、生成する Java ソース・ファイルと同じディレクトリーにこの SMAP ファイルを配置します。

**-encoding=***encoding*

ファイルの入力および出力文字のエンコードを指定します。このオプションが指定されていない場合、ファイルのエンコードは、Java システム・プロパティー "file.encoding" からとられます。

**-ser2class**

直列化プロファイルをクラス・ファイルに変換します。これは、.ser ファイルのロード問題がある特定のブラウザで SQLJ 実行ファイルを実行するために必要になることがあります。

**-status**

SQLJ 処理の即時状況表示を印刷します。

**-version**

SQLJ 変換プログラムのビルド・バージョンを表示します。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。

### -C-help

SQLJ 変換プログラムによって、Java コンパイラーのヘルプ情報を表示するかどうかを指定します。

### -CJava compiler option

Java コンパイラーにオプションを渡します。変換プログラムが必要な従属関係ファイルを検出するために使用するクラス・パスとソース・パスを指定できます。指定するクラス・パスとソース・パスは、変換プログラムとコンパイラーの両方によって使用されます。

### -JJava VM option

Java 仮想マシンにオプションを渡します。

### SQLJ source file

1 つ以上の .sqlj ファイル。

**-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。

### -db2optimize

SQLJ 変換プログラムによって、DB2 用に最適化された接続コンテキスト・クラス用コードを生成することを指定します。このオプションは、デフォルト・コンテキスト用のコードではなくユーザー定義コンテキスト用のコードを最適化します。この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は、生成する Java アプリケーションをコンパイルするために、DB2 Universal JDBC Driver ファイル db2jcc.jar を CLASSPATH に入れておく必要があります。

### 例:

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

### 使用上の注意:

生成されるそれぞれの直列化プロファイルは、SQLJ ソース・コードで使用される接続コンテキスト・クラスに対応しています。プロファイルの名前は、pgmname\_SJProfile0.ser、pgmname\_SJProfile1.ser、pgmname\_SJProfile2.ser などです。ここで数字は、接続コンテキスト・クラスがソース・ファイル内で参照される順序を表しています。**db2sqljprint** コマンドは、プロファイルの内容を表示するために使用できます。それぞれのプロファイルは、**db2sqljcustomize** コマンドを使用して別々にカスタマイズする必要があります。

### 関連資料:

- 227 ページの『db2sqljcustomize - DB2 SQLJ プロファイル・カスタマイザー』
- 234 ページの『db2sqljprint - DB2 SQLJ プロファイル・プリンター』
- 223 ページの『db2sqljbind - DB2 SQLJ プロファイル・バインド・プログラム』



## 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、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。

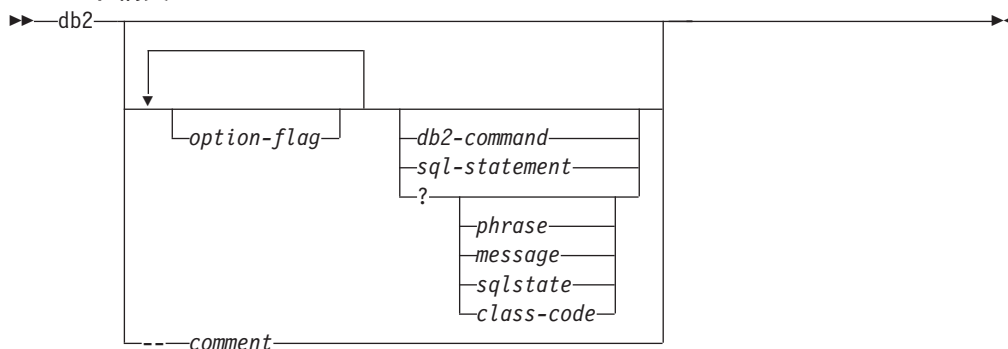
**注:** Windows で **db2cmd** を実行すると、CLP 対応の DB2 ウィンドウが開き、DB2 コマンド行環境が初期化されます。このコマンドを実行することは、「DB2 コマンド・ウィンドウ (DB2 Command Window)」アイコンをクリックすることと同じです。

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。

**注:** 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 (たとえば、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 (たとえば、**? 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。

**注:** 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。

### **関連概念:**

- 273 ページの『コマンド行プロセッサ (CLP)』

### **関連資料:**

- 264 ページの『コマンド行プロセッサのオプション』
- 272 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 1 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。たとえば、-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 1. 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。	OFF
-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。	ON
-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。	OFF
-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。	OFF
-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。	OFF
-n	単一の区切りトークン内の改行文字を除去します。このオプションを指定しない場合、改行文字はスペースで置換されます。このオプションは、-t オプションと共に使用する必要があります。	OFF
-o	コマンド行プロセッサは、出力データおよびメッセージを標準出力に表示します。	ON
-p	コマンド行プロセッサは、対話型入力モードのときにコマンド行プロセッサ・プロンプトを表示します。	ON
-rfilename	コマンド行プロセッサは、コマンドが生成したレポートをファイルに書き込みます。	OFF
-s	コマンド行プロセッサは、バッチ・ファイルまたは対話式モードでコマンドを実行中にエラーが発生した場合、実行を停止します。	OFF

## db2 - コマンド行プロセッサの呼び出し

表 1. 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
-t	コマンド行プロセッサはセミコロン (;) をステートメント終了文字として使用します。	OFF
-tdx	コマンド行プロセッサは <i>x</i> をステートメント終了文字として定義し、使用します。	OFF
-v	コマンド行プロセッサはコマンド・テキストを標準出力にエコーさせます。	OFF
-w	コマンド行プロセッサは SQL ステートメント警告メッセージを表示します。	ON
-x	コマンド行プロセッサは列名などのヘッダーなしでデータを戻します。	OFF
-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。	OFF

### 例

#### AIX コマンド

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

を実行すると、このセッションのデフォルト設定は以下のようになります。

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

以下に、これらのオプションの詳細を説明します。

#### SQLCA データ表示オプション (-a):

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

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

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

#### 自動コミット・オプション (-c):

このオプションは、各コマンドまたはステートメントを独立して処理するかどうかを指定します。ON (-c) に設定する場合、各コマンドまたはステートメントは自動的にコミットされるかまたはロールバックされます。コマンドまたはステートメントが正常に処理された場合、それ自体、およびそれ以前に自動コミット OFF (+c または -c-) で発行され、正常に発行されたコマンドおよびステートメントがすべてコミットされます。しかし、コマンドまたはステートメントが失敗した場合、それ自体、およびそれ以前に自動コミット 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ので失敗します。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。

### SQLCODE/SQLSTATE 表示オプション (-e):

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

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

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

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

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

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

### 入力ファイルからの読み取りオプション (-f):

-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:

```
db2 -tvf filename
```

## db2 - コマンド行プロセッサの呼び出し

**注:** 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全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。

**-ffilename** 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。

### 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すで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:

```
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。

### 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。

**注:** このオプションは、対話式モード内では変更できません。

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

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

### 出力表示オプション (-o):

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

このコマンド・オプションのデフォルト設定は **ON** です。

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

以下のオプションは、+o オプションの影響を受けることがあります。

- **-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。たとえば、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、-ffilename 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。

#### 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。



## db2 - コマンド行プロセッサの呼び出し

-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 2. 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、スラッシュ (/) 行継続文字は使用不能になります。

**注:** 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-)。

終了文字を定義するには、-td の後に選択した終了文字を指定します。たとえば、-tdx とすると x 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ブランク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 オプションを指定すると、コマンド行プロセッサは SQL ステートメント警告メッセージを表示します。

このコマンド・オプションのデフォルト設定は ON です。

### 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。

このコマンド・オプションのデフォルトの設定値は 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:

- 263 ページの『db2 - コマンド行プロセッサの呼び出し』
- 272 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。

たとえば、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 (-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。

戻りコードが利用可能なのは、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。

たとえば、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、コマンドの実行は継続します。

**関連資料:**

- 263 ページの『db2 - コマンド行プロセッサの呼び出し』
- 264 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 (CLP)

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

- SQL ステートメント `DB2® CONNECT TO database` を発行します。
- DB2 Universal Database™ (UDB) 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。たとえば、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。ランタイムにこれらのトークンが特定の値に置き換えられます。

## db2 - コマンド行プロセッサの呼び出し

表 3. 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・アタッチの許可 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・アタッチ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

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

### 例:

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

- インスタンス接続なし、およびデータベース接続なし。 DB2INSTANCE は "DB2" に設定される。 DB2DBDFT は設定されない。  
(@DB2, @)
- (Windows) インスタンス接続なし、およびデータベース接続なし。  
DB2INSTANCE と DB2DBDFT は設定されない。 DB2INSTDEF は "DB2" に設定される。  
(@DB2, @)
- インスタンス接続なし、およびデータベース接続なし。 DB2INSTANCE は "DB2" に設定される。 DB2DBDFT は "SAMPLE" に設定される。  
(@DB2, @SAMPLE)
- 許可 ID "tyronnem" を使用してインスタンス "DB2" にインスタンス・アタッチ。 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・アタッチ。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。

たとえば以下のように、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

## db2 - コマンド行プロセッサの呼び出し

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

### DB2BQTRY

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

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

### DB2RQTIME

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

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

### DB2IQTIME

バック・エンド・プロセスがフロントエンド・プロセスから要求を受信すると、バック・エンド・プロセスはフロントエンド・プロセスに肯定応答を送り、入力キューを介して入力を受け取る準備ができていることを知らせま



す。そして、バック・エンド・プロセスは入力キューで待機します。さらに、バッチ・ファイル (-f オプションで指定) が実行している間、またユーザーが対話式モードの間も、バック・エンド・プロセスは入力キューで待機します。

DB2IQTIME 変数は、フロントエンド・プロセス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・プロセスがすでに存在していない場合は、要求キューに戻って待機します。フロントエンド・プロセスが存在している場合は、バック・エンド・プロセスはフロントエンド・プロセスからの入力を待機します。

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

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

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

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

ユーザー B のフロントエンド・プロセス (コマンドが出されたオペレーティング・システム・ウィンドウ) の親が依然としてアクティブであるため、ユーザー A が開始したバック・エンド・プロセスは、ユーザー B が CLP を使用して開始した時点でもまだアクティブになっています。バック・エンド・プロセスは、ユーザー B が発行した新規コマンドを処理しようとしませんが、ユーザー B のフロントエンド・プロセスには、バック・エンド・プロセスのメッセージ・キューを使用するだけの十分な権限がありません。そのためにはバック・エンド・プロセスを作成したユーザー A の権限が必要だからです。CLP セッションを TERMINATE コマンドを使用して終了してから、ユーザーは同じオペレーティング・システム・ウィンドウを使用して、新規 CLP セッションを開始しなければなりません。これによって新規ユーザーごとに新しいバック・エンド・プロセスが作成されることになり、権限の問題は回避され、新規ユーザーのバック・エンド・プロセス内で環境変数 (DB2INSTANCE など) の値を訂正して設定できます。

#### CLP 使用上の注意:

コマンドはコマンド・プロンプトから大文字または小文字で入力できます。ただし、DB2 が大文字小文字を区別するパラメーターには、大文字小文字を正確に入力しなければなりません。たとえば、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。

## db2 - コマンド行プロセッサの呼び出し

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 (たとえば、\$ & \* ( ) ; < > ? ¥ ' " 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、疑問符またはエスケープ文字が必要です。

たとえば、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 (¥)、たとえば ¥\*、¥>、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。たとえば、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。たとえば、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。たとえば、ホスト変数およびステートメントはサポートしません。そのため、

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

このステートメント例は、:HostVar が有効なデータベース名ではないため、構文上は正しくありません。

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・ウィンドウから単一バイト (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。たとえば、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。



## 第 3 章 CLP コマンド

この章では、DB2 コマンドをアルファベット順に説明します。これらのコマンドを使用して、システムを対話式に制御できます。

注: ディレクトリー・パス中の斜線 (/) は UNIX ベースのシステムだけに用いられるもので、Windows オペレーティング・システムの円記号 (¥) に相当します。

### DB2 CLP コマンド

次の表では、CLP コマンドを機能別カテゴリーに分けて示しています。

表 5. DB2 CLP コマンド

CLP セッション制御
519 ページの『LIST COMMAND OPTIONS』
784 ページの『UPDATE COMMAND OPTIONS』
355 ページの『CHANGE ISOLATION LEVEL』
735 ページの『SET RUNTIME DEGREE』
761 ページの『TERMINATE』
641 ページの『QUIT』
データベース・マネージャー制御
744 ページの『START DATABASE MANAGER』
752 ページの『STOP DATABASE MANAGER』
428 ページの『GET DATABASE MANAGER CONFIGURATION』
691 ページの『RESET DATABASE MANAGER CONFIGURATION』
791 ページの『UPDATE DATABASE MANAGER CONFIGURATION』
296 ページの『AUTOCONFIGURE』
データベース制御
695 ページの『RESTART DATABASE』
357 ページの『CREATE DATABASE』
380 ページの『DROP DATABASE』
599 ページの『MIGRATE DATABASE』
283 ページの『ACTIVATE DATABASE』
370 ページの『DEACTIVATE DATABASE』
635 ページの『QUIESCE』
771 ページの『UNQUIESCE』
538 ページの『LIST INDOUBT TRANSACTIONS』
533 ページの『LIST DRDA INDOUBT TRANSACTIONS』
422 ページの『GET DATABASE CONFIGURATION』
689 ページの『RESET DATABASE CONFIGURATION』
788 ページの『UPDATE DATABASE CONFIGURATION』

表 5. DB2 CLP コマンド (続き)

296 ページの『AUTOCONFIGURE』
<b>データベース・ディレクトリー管理</b>
330 ページの『CATALOG DATABASE』
762 ページの『UNCATALOG DATABASE』
334 ページの『CATALOG DCS DATABASE』
764 ページの『UNCATALOG DCS DATABASE』
353 ページの『CHANGE DATABASE COMMENT』
520 ページの『LIST DATABASE DIRECTORY』
531 ページの『LIST DCS DIRECTORY』
<b>ODBC 管理</b>
348 ページの『CATALOG ODBC DATA SOURCE』
545 ページの『LIST ODBC DATA SOURCES』
770 ページの『UNCATALOG ODBC DATA SOURCE』
416 ページの『GET CLI CONFIGURATION』
782 ページの『UPDATE CLI CONFIGURATION』
<b>クライアント/サーバー・ディレクトリー管理</b>
342 ページの『CATALOG LOCAL NODE』
344 ページの『CATALOG NAMED PIPE NODE』
324 ページの『CATALOG APPC NODE』
327 ページの『CATALOG APPN NODE』
346 ページの『CATALOG NETBIOS NODE』
349 ページの『CATALOG TCPIP NODE』
769 ページの『UNCATALOG NODE』
542 ページの『LIST NODE DIRECTORY』
<b>ネットワーク・サポート</b>
661 ページの『REGISTER』
372 ページの『DEREGISTER』
796 ページの『UPDATE LDAP NODE』
337 ページの『CATALOG LDAP DATABASE』
766 ページの『UNCATALOG LDAP DATABASE』
341 ページの『CATALOG LDAP NODE』
768 ページの『UNCATALOG LDAP NODE』
660 ページの『REFRESH LDAP』
<b>DB2 Administration Server</b>
407 ページの『GET ADMIN CONFIGURATION』
685 ページの『RESET ADMIN CONFIGURATION』
773 ページの『UPDATE ADMIN CONFIGURATION』
367 ページの『CREATE TOOLS CATALOG』
388 ページの『DROP TOOLS CATALOG』
<b>リカバリー</b>
291 ページの『ARCHIVE LOG』

表 5. DB2 CLP コマンド (続き)

299 ページの『BACKUP DATABASE』
646 ページの『RECONCILE』
697 ページの『RESTORE DATABASE』
709 ページの『ROLLFORWARD DATABASE』
535 ページの『LIST HISTORY』
630 ページの『PRUNE HISTORY/LOGFILE』
794 ページの『UPDATE HISTORY FILE』
509 ページの『INITIALIZE TAPE』
708 ページの『REWIND TAPE』
739 ページの『SET TAPE POSITION』
操作ユーティリティー
405 ページの『FORCE APPLICATION』
546 ページの『LIST PACKAGES/TABLES』
674 ページの『REORGCHK』
666 ページの『REORG INDEXES/TABLE』
720 ページの『RUNSTATS』
データベース・モニター
443 ページの『GET MONITOR SWITCHES』
798 ページの『UPDATE MONITOR SWITCHES』
433 ページの『GET DATABASE MANAGER MONITOR SWITCHES』
452 ページの『GET SNAPSHOT』
693 ページの『RESET MONITOR』
510 ページの『INSPECT』
515 ページの『LIST ACTIVE DATABASES』
517 ページの『LIST APPLICATIONS』
528 ページの『LIST DCS APPLICATIONS』
データ・ユーティリティー
393 ページの『EXPORT』
483 ページの『IMPORT』
558 ページの『LOAD』
596 ページの『LOAD QUERY』
ヘルス・センター
285 ページの『ADD CONTACT』
286 ページの『ADD CONTACTGROUP』
378 ページの『DROP CONTACT』
379 ページの『DROP CONTACTGROUP』
409 ページの『GET ALERT CONFIGURATION』
419 ページの『GET CONTACTGROUP』
420 ページの『GET CONTACTGROUPS』
421 ページの『GET CONTACTS』
436 ページの『GET DESCRIPTION FOR HEALTH INDICATOR』

表 5. DB2 CLP コマンド (続き)

438 ページの『GET HEALTH NOTIFICATION CONTACT LIST』
439 ページの『GET HEALTH SNAPSHOT』
446 ページの『GET RECOMMENDATIONS』
687 ページの『RESET ALERT CONFIGURATION』
775 ページの『UPDATE ALERT CONFIGURATION』
786 ページの『UPDATE CONTACT』
787 ページの『UPDATE CONTACTGROUP』
793 ページの『UPDATE HEALTH NOTIFICATION CONTACT LIST』
アプリケーションの準備
603 ページの『PRECOMPILE』
305 ページの『BIND』
642 ページの『REBIND』
リモート・サーバー・ユーティリティー
294 ページの『ATTACH』
377 ページの『DETACH』
表スペース管理
549 ページの『LIST TABLESPACE CONTAINERS』
737 ページの『SET TABLESPACE CONTAINERS』
551 ページの『LIST TABLESPACES』
638 ページの『QUIESCE TABLESPACES FOR TABLE』
データベース・パーティション管理
289 ページの『ADD DBPARTITIONNUM』
386 ページの『DROP DBPARTITIONNUM VERIFY』
527 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
524 ページの『LIST DATABASE PARTITION GROUPS』
656 ページの『REDISTRIBUTE DATABASE PARTITION GROUP』
データ・リンク
287 ページの『ADD DATALINKS MANAGER』
382 ページの『DROP DATALINKS MANAGER』
526 ページの『LIST DATALINKS MANAGERS』
追加コマンド
373 ページの『DESCRIBE』
390 ページの『ECHO』
415 ページの『GET AUTHORIZATIONS』
418 ページの『GET CONNECTION STATE』
442 ページの『GET INSTANCE』
450 ページの『GET ROUTINE』
481 ページの『HELP』
601 ページの『PING』
632 ページの『PUT ROUTINE』

表 5. DB2 CLP コマンド (続き)

634 ページの『QUERY CLIENT』
732 ページの『SET CLIENT』

## ACTIVATE DATABASE

指定したデータベースを活動化し、必要なデータベース・サービスをすべて開始することにより、どのアプリケーションからでもそのデータベースに接続して使用できるようにします。

### 有効範囲:

このコマンドは、システム内のすべてのパーティションで、指定したデータベースを活動化します。データベースを活動化中に 1 つ以上のパーティションがエラーを検出すると、警告が戻されます。コマンドが正常に処理されたすべてのパーティションで、データベースは活動状態を維持します。

### 権限:

以下のいずれかが必要です。

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

### 必要な接続:

なし

### コマンド構文:

```

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

```

### コマンド・パラメーター:

#### database-alias

開始するデータベースの別名を指定します。

#### USER username

データベースを開始するユーザーを指定します。

#### USING password

ユーザー名のパスワードを指定します。

### 使用上の注意:

データベースが開始していないときに、アプリケーションで **CONNECT TO** (または暗黙的な接続) が発行された場合、アプリケーションは、データベースに対して作業する前に、必要なデータベースをデータベース・マネージャーが開始する間待機しなければなりません。しかし、データベースが開始してしまえば、他のアプリ

## ACTIVATE DATABASE

セッションはデータベースの開始を待たずに、ただデータベースに接続するだけでその中のデータを使用することができます。

データベース管理者は、選択したデータベースを開始する際に `ACTIVATE DATABASE` を使用することができます。そうしておけば、データベースの初期化に要するアプリケーション時間を節約することができます。

`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) によって開始したデータベースに続けて `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 (たとえば、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。データベースを再始動してからでないと、`ACTIVATE DATABASE` でそのデータベースを初期化できません。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。

**注:** `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。

### 関連資料:

- 752 ページの『`STOP DATABASE MANAGER`』
- 370 ページの『`DEACTIVATE DATABASE`』

## ADD CONTACT

このコマンドは、システムでローカルに定義されるか、またはグローバル・リストで定義される、連絡先リストに連絡先を追加します。連絡先とは、スケジューラーおよびヘルス・モニターなどのプロセスが、メッセージを送信する先のユーザーです。 Database Administration Server (DAS) *contact\_host* 構成パラメーターは、リストがローカルかグローバルかを判別します。

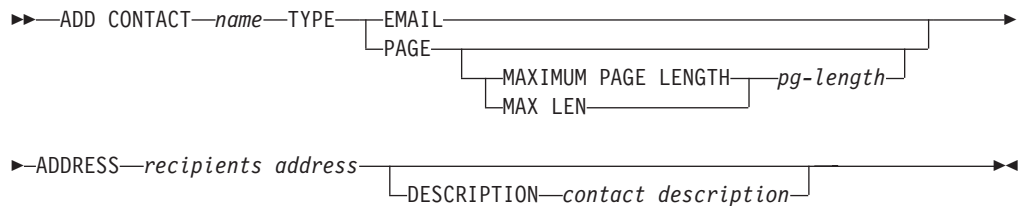
### 権限:

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。

**注:** 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 の役割です。

#### ADDRESS recipients-address

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

#### DESCRIPTION contact description

連絡先のテキスト記述。長さは、最大 128 文字です。

---

## ADD CONTACTGROUP

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

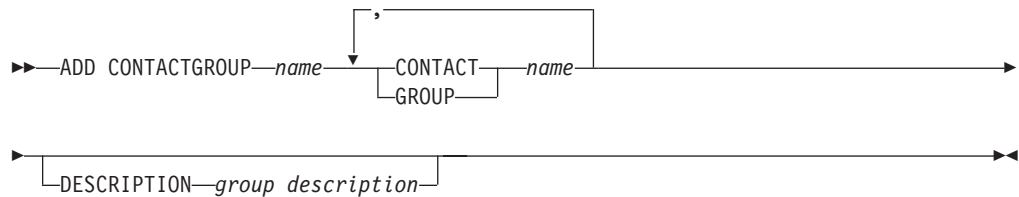
### 権限:

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ユニーク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。その連絡先をグループに含める前に、個々の連絡先を定義する必要はありません。

#### **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*

オプションです。連絡先グループのテキスト記述。





## ADD DATALINKS MANAGER

**注:** 同じ名前またはアドレスを使用する、重複した Data Links Manager が追加される場合には、コマンド行プロセッサはエラーを検出します。ただし、異なる IP 名またはアドレスを使用して複数回 Data Links Manager が追加される場合は、重複は検出されません。たとえば、Data Links Manager が 2 度追加されており、最初は `dln1.almaden.ibm.com` という名前、次は `dln1` というショート・ネームを使用しているとすると、上記のような障害が発生する可能性があります。

### 関連資料:

- 526 ページの『LIST DATALINKS MANAGERS』
- 382 ページの『DROP DATALINKS MANAGER』

## ADD DBPARTITIONNUM

パーティション・データベース環境に新規のデータベース・パーティションを追加します。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システム TEMPORARY 表スペースの定義用のデータベース・パーティションを指定できます。またはシステム 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。

### 権限:

以下のいずれかが必要です。

- *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*

新しいシステム 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。

#### WITHOUT TABLESPACES

システム 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。

**注:** オプションを指定しない場合、システム TEMPORARY 表スペース用のコンテナは各データベースのカatalog・パーティション上のコンテナと同じになります。カatalog・パーティションは、パーティション環境内の各データベースによって別々のデータベース・パーティションにできます。

### 使用上の注意:

## ADD DBPARTITIONNUM

新規のデータベース・パーティションを追加する前に、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、すで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。他のすべてのデータベース・パーティションにある既存データベース・パーティションは、影響を受けません。この終結処理のフェーズが失敗すると、終結処理は停止し、エラーが戻されます。

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

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

データベース・パーティションにシステム `TEMPORARY` 表スペースを作成する場合は、その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、`ADD DBPARTITIONNUM` に別のデータベース・パーティション・サーバーと通信させなければならない場合があります。他のデータベース・パーティション・サーバーが表スペース定義を応答しなければならない時間を、`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分単位で指定します。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。

### 関連資料:

- 744 ページの『START DATABASE MANAGER』

## ARCHIVE LOG

回復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。

权限:

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:

```

graph LR
    A[ARCHIVE LOG FOR] --- B[DATABASE]
    B --- DB[DB]
    B --- Alias[database-alias]
    A --- C[USER]
    C --- Username[username]
    A --- D[USING]
    D --- Password[password]
    A --- E[On Database Partition Number Clause]
  
```

The diagram illustrates the components of the `ARCHIVE LOG FOR` statement. It shows three main clauses separated by horizontal lines:

- Database Alias Clause:** `ARCHIVE LOG FOR DATABASE DB database-alias`. The `DATABASE` keyword is followed by `DB` (in brackets) and the `database-alias`.
- User/Password Clause:** `USER username USING password`. The `USER` keyword is followed by `username` (in brackets), and the `USING` keyword is followed by `password` (in brackets).
- On Database Partition Number Clause:** `On Database Partition Number Clause`.

### On Database Partition Number Clause:

```

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

```

**Database Partition Number List Clause:**

Database Partition Number List Class:

DBPARTITIONNUM  
DBPARTITIONNUMS

db-partition-number  
TO db-partition-number

コマンド・パラメーター:

**DATABASE** database-alias

アーカイブするアクティブ・ログを持つデータベースの別名を指定します。

**USER** username

接続を試みるユーザー名を識別します。

## 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すで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。

## ATTACH

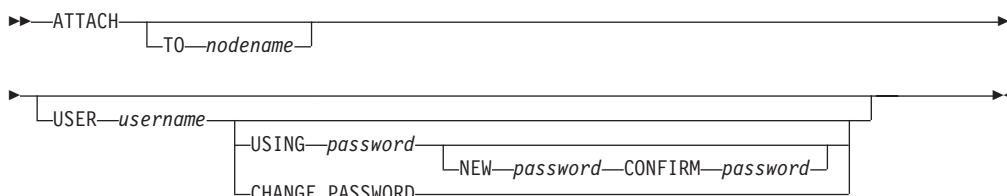
インスタンス・レベルのコマンド (たとえば CREATE DATABASE および FORCE APPLICATION) を実行するインスタンスを指定することを可能にします。このインスタンスは、現在のインスタンス、同じワークステーション上の別のインスタンス、またはリモート・ワークステーションのインスタンスのいずれかになります。

**権限:**

なし

**必要な接続:**

なし。このコマンドは、インスタンス・アタッチを確立します。

**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しようとしているインスタンスの別名です。このインスタンスには、ローカル・ノード・ディレクトリーの一致項目が必要です。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 Universal Database (UDB) インスタンスにアタッチする場合は、Microsoft Windows NT 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できます (たとえば、*domainname¥username*)。

**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、その後ディタッチします。

```
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、その後ディタッチします。

```
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、その後ディタッチします (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。

**関連資料:**

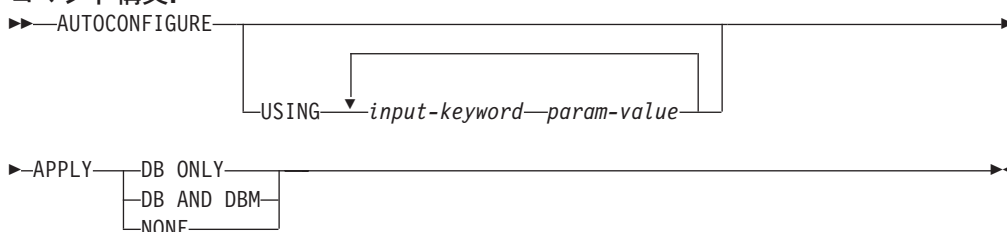
- 377 ページの『DETACH』

权限:

必要な接続:

データベース。

コマンド構文:



コマンド・パラメーター:

## USING input-keyword param-value

表 6. 有効な入力キーワードおよびパラメーター値

キーワード	有効値	デフォルト値	説明
mem_percent	1-100	80	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 6. 有効な入力キーワードおよびパラメーター値 (続き)

キーワード	有効値	デフォルト値	説明
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** 推奨される変更を表示しますが、適用はしません。

### 使用上の注意:

入力キーワードのいずれかが指定されないと、そのパラメーターにはデフォルト値が使用されます。

パーティション・データベース環境では、このコマンドは現行パーティションにしか適用しません。

複数のロジカル・パーティションのあるシステムでは、*mem\_percent* パラメーターは、すべてのロジカル・パーティションによって使用されるメモリーのパーセントを参照します。たとえば、DB2 がシステムの 80% 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、80% を指定します。データベースの推奨構成が作成されますが、これは 1 つのロジカル・パーティション用に調整されます。

このコマンドは、現在接続されているデータベース（このデータベースがシステム上で唯一のアクティブ・データベースであると仮定）のための、推奨構成を作成します。複数のデータベースがシステムでアクティブな場合は、*mem\_percent* パラメーターを調整して、現行データベースのメモリーの分散を反映します。たとえば、DB2 がシステム・メモリーの 80% を使用し、リソースを平等に共有する 2 つのアクティブ・データベースがシステムにある場合、40% (80% を 2 データベースで割る) をパラメーター *mem\_percent* に指定します。

# BACKUP DATABASE

データベースまたは表スペースのバックアップ・コピーを作成します。

## 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。

## 権限:

以下のいずれか。

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

## 必要な接続:

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

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

## コマンド構文:



## コマンド・パラメーター:

## BACKUP DATABASE

### 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* を使用可能にして構成されたデータベースにのみ、使用可能です。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。

**注:** 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

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。

**注:** 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名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・サーバー上に存在する必要があります。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 (たとえば、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ユニークなデバイス・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 (たとえば、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 (たとえば、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

使用するベンダーのバックアップおよびリストア I/O 機能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

パラメーターのため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ため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。

SCO UnixWare 7 上で磁気テープ装置を使用するときは、バッファ・サイズを 16 に指定します。

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ため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。

注: 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:

1. 以下の例で、データベース WSDb は 0 から 3 までの番号が付けられた 4 つのパーティションすべてに定義されています。パス /dev3/backup はすべてのパーティションからアクセスできます。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・パーティションに出します (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。

## BACKUP DATABASE

2. 以下の例で、データベース SAMPLE は TSM サーバーに 2 つの並行 TSM クライアント・セッションを使用してバックアップされます。この環境に最適なバッファ・サイズが DB2 によって計算されます。

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

3. 次の例では、データベース payroll の表スペース (syscatspace、userspace1) の表スペース・レベル・バックアップがテープに対して実行されます。

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

4. バックアップ操作で使用する TSM 情報を指定するには、USE TSM OPTIONS キーワードを使用します。次の例は、USE TSM OPTIONS キーワードを使用して、完全修飾ファイル名を指定する方法を示すものです。

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

ファイル myoptions.txt には、-fromnode=bar -fromowner=dmcinnis というストリングが含まれています。

5. 以下は、リカバリー可能データベース用の増分バックアップの週間予定のサンプルです。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。

### 関連資料:

- 697 ページの『RESTORE DATABASE』
- 709 ページの『ROLLFORWARD DATABASE』

# BIND

バインド・ユーティリティーを呼び出すことにより、プリコンパイラーに生成されるバインド・ファイルに SQL ステートメントを保管し、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・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。

## 権限:

以下のどれかが必要です。

- *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する場合は、スキーマに対する CREATEIN 特権
- パッケージが存在する場合は、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)

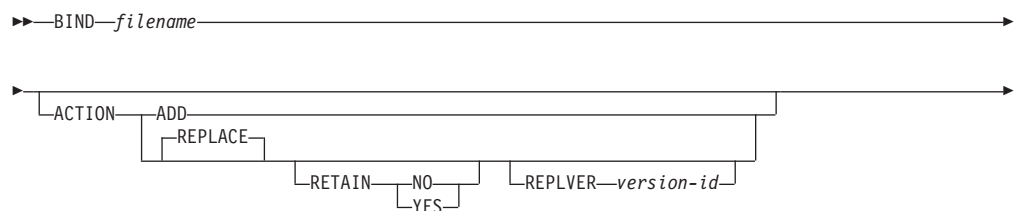
ユーザーには、アプリケーション内の静的 SQL ステートメントをコンパイルするのに必要な特権もすべて必要になります。グループに付与された特権は、静的ステートメントの許可検査では使用されません。ユーザーに *sysadm* 権限があつてバインドを完成させる明示特権がない場合、データベース・マネージャーは、明示的な *dbadm* 権限を自動的に付与します。

## 必要な接続:

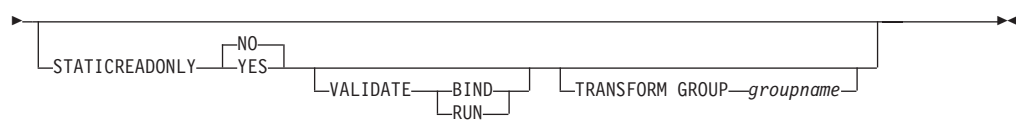
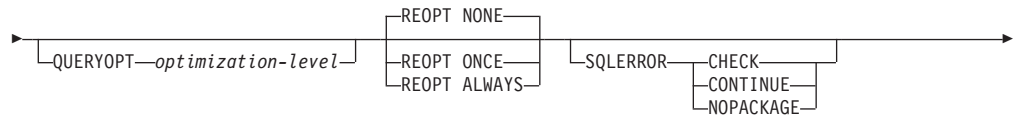
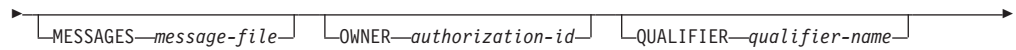
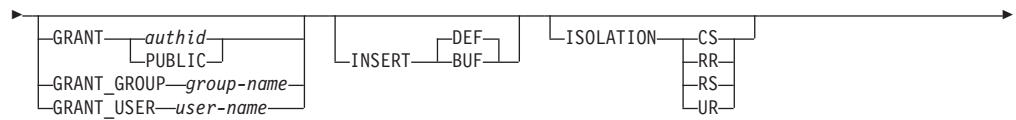
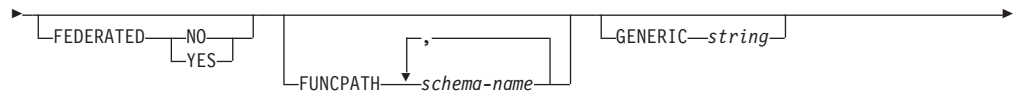
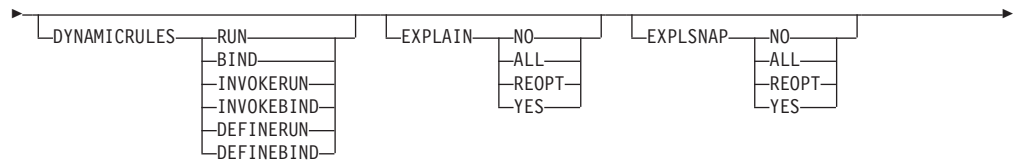
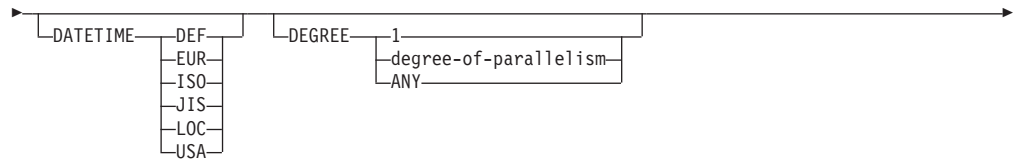
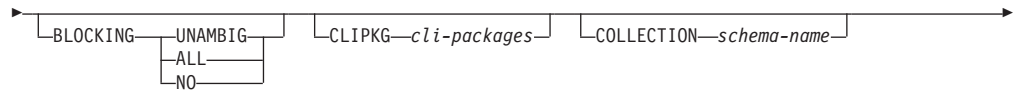
データベース。暗黙接続が可能な場合には、デフォルト・データベースへの接続が確立されます。

## コマンド構文:

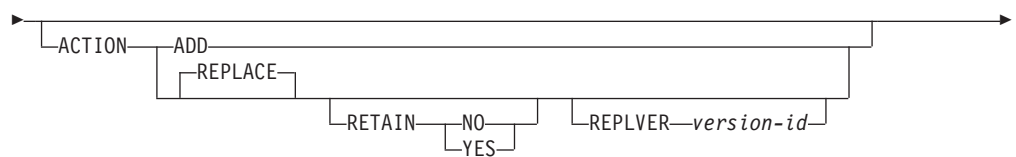
**DB2 for Windows および DB2 for UNIX では**

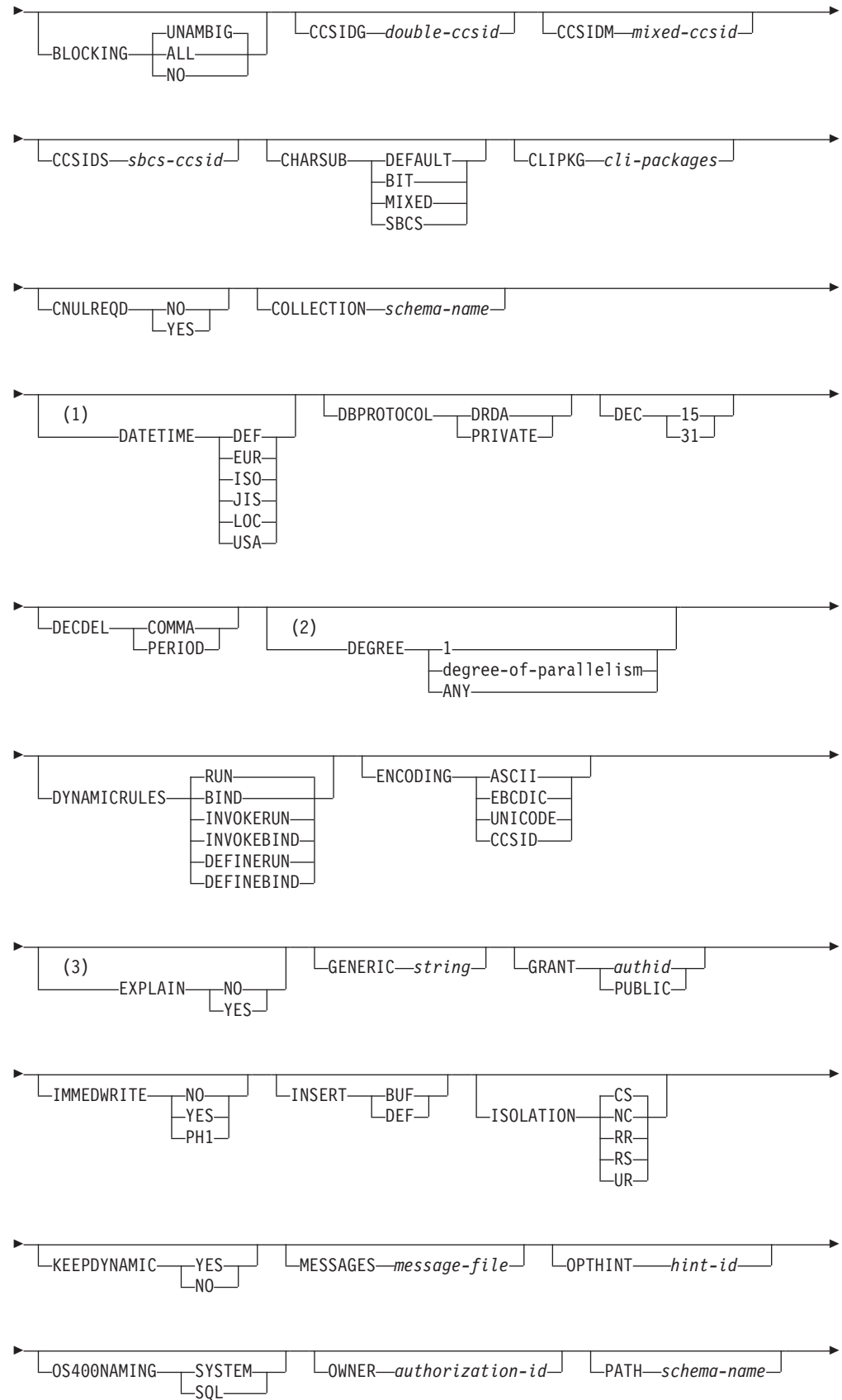


## BIND

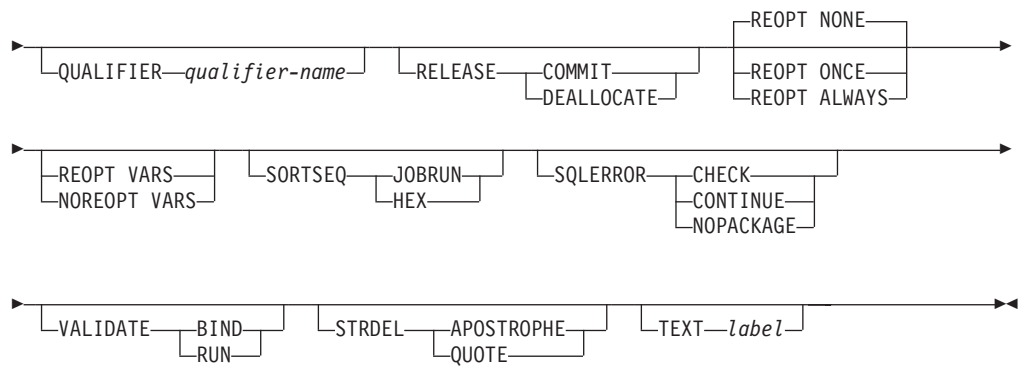


**Windows および UNIX 以外のサーバー上の DB2 では**





## 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。たとえば、次のように指定します。

```
/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ということを指示します。すで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。これがデフォルト値です。

## 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すでに存在すれば、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。

あいまいなカーソルは、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。なお、DB2 はこの DRDA プリコンパイラ/ 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。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。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。

### **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 パート名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、ランタイムの動的 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。これがデフォルト値です。

**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、ランタイム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。

**注:** 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、増分バインド・ステートメントの場合はランタイムに、選択されたアクセス・プランについての情報が取り込まれます。

パッケージがルーチンに使用されるもので、パッケージに追加バインド・ステートメントが含まれる場合、そのルーチンは MODIFIES SQL DATA として定義されなければなりません。これが行われない場合、パッケージ内の追加バインド・ステートメントはランタイム・エラーを生じます (SQLSTATE 42985)。

## REOPT

再最適化可能な増分バインド SQL ステートメントの Explain 情報が、ランタイムに各 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 情報はランタイム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。適格な増分バインド SQL ステートメントの Explain 情報が、ランタイムに各 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 情報はランタイムに適格な動的 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)。

注: 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、増分バインド・ステートメントの場合はランタイムに、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、Explain 表内に入れます。

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

再最適化可能な増分バインド SQL ステートメントの Explain スナップショット情報が、ランタイムに各 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ランタイム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。適格な増分バインド SQL ステートメントの Explain スナップショット情報が、ランタイムに各 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ランタイムに適格な動的 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)。

注: この値は DRDA ではサポートされていません。

## 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ニックネームを参照する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ニックネームを参照する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ニックネームを参照するビューは、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ニックネームを参照する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ニックネームを参照するビューは、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ニックネームを参照する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、ユーザー定義の個別タイプおよび機能を解析する際に使用する機能パスを指定します。このオプションを指定しなかった場合、デフォルトの機能パスは "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、機能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、処理結果の機能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。機能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 ..."
```

## BIND

各オプションと値は、1 つ以上のブランク・スペースで区切らなければなりません。たとえば、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。

**注:** 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** コミットしません。コミットメント制御が使用されないということ指定します。 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。

**MESSAGES message-file**

警告メッセージ、エラー・メッセージ、および完了状況メッセージの宛先を指定します。メッセージ・ファイルは、バインドが正常であるかどうかによって作成されます。メッセージ・ファイル名を指定しなかった場合、メッセージは標準出力に書き込まれます。ファイルへの完全パスを指定しなかった場合は、現行ディレクトリが使用されます。なお、既存ファイルの名前を指定すると、そのファイルの内容は上書きされます。

**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。

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、区切り記号としてスラッシュが使用されているために、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ので注意してください。たとえば、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、ユーザー定義の個別タイプおよび機能を解析する際に使用する機能パスを指定します。このオプションを指定しなかった場合、デフォルトの機能パスは "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、使用可能な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。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。ランタイム



で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ランタイムにアクセス・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。これがデフォルト値です。

**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ランタイムにアクセス・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。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** 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。

注: 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。

**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。これがデフォルト値です。

## 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型の値をホスト・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、構造型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・メッセージ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。

バインドは、アプリケーション・プログラム・ソース・ファイルのプリコンパイル処理の一部として、または後の分離ステップとして実行することができます。分離処理としてバインドを実行するときは、**BIND** を使用してください。

パッケージの作成に使用する名前は、それを生成したソース・ファイル名を基にして指定され (既存パスや拡張子は廃棄されます)、バインド・ファイルに保管されます。たとえば、myapp.sqc 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。

すで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ず。
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を使用される環境について **DYNAMICRULES** オプションで指定された動作を公開します。

SQL0020W メッセージに表示されるパラメーターは正しくエラーと示され、メッセージが示すとおりは無視されます。

SQL ステートメントがエラーであることが検出され、**BIND** オプション **SQLERROR CONTINUE** が指定されている場合、このステートメントは無効とマークされます。この **SQL** ステートメントの状態を変えるためには、さらに別の **BIND** を発行する必要があります。暗黙的および明示的な再バインドでは、無効なステートメントの状態は変わりません。**VALIDATE RUN** でバインドされたパッケージでは、ステートメントは、再バインド時にオブジェクトが存在するかまたは権限の問題があるかどうかに応じて、暗黙的と明示的な再バインドとの両方で、静的バインドから増分バインドに変更したり、増分バインドを静的バインドに変更することができます。

**関連概念:**

- 「*SQL* リファレンス 第 1 巻」の『分離レベル』
- 「アプリケーション開発ガイド クライアント・アプリケーションのプログラミング」の『動的 **SQL** における許可に関する考慮事項』
- 「アプリケーション開発ガイド クライアント・アプリケーションのプログラミング」の『動的 **SQL** における **DYNAMICRULES BIND** オプションの影響』
- 「アプリケーション開発ガイド クライアント・アプリケーションのプログラミング」の『静的 **SQL** における **REOPT** の影響』
- 「アプリケーション開発ガイド クライアント・アプリケーションのプログラミング」の『動的 **SQL** における **REOPT** の影響』

**関連資料:**

- 「*SQL* リファレンス 第 2 巻」の『**SET CURRENT QUERY OPTIMIZATION** ステートメント』
- 603 ページの『**PRECOMPILE**』
- 「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』
- 「*SQL* リファレンス 第 1 巻」の『特殊レジスター』
- 「*SQL* リファレンス 第 1 巻」の『日付/時刻の値』

## CATALOG APPC NODE

ノード・ディレクトリーに APPC 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には、APPC プロトコルを使用します。

### 権限:

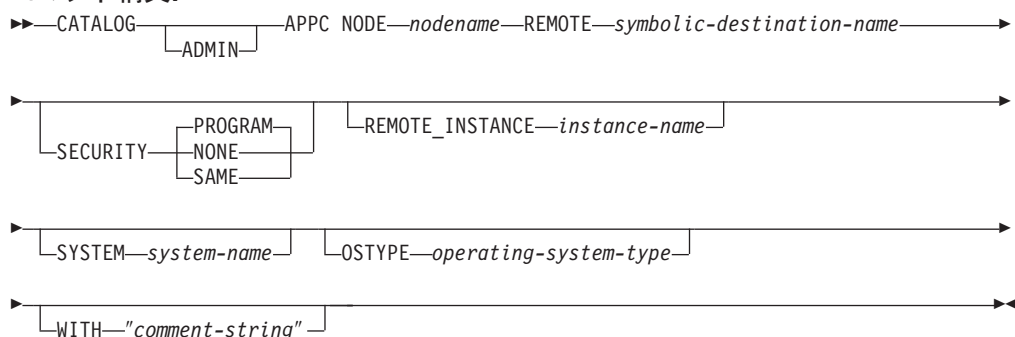
以下のどれかが必要です。

- *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

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。

#### REMOTE **symbolic-destination-name**

リモート・パートナー・ノードのシンボリック宛先名を指定します。この名前は、クライアントがサーバーへの APPC 接続をセットアップする際に必要な情報を含む、CPI 通信のサイド情報表の項目に対応します (パートナー LU 名、モード名、パートナー TP 名)。最大長は 8 文字です。

#### SECURITY

パートナー LU に送信する割り振り要求において、セキュリティー情報を使用する度合いを指定します。有効な値は以下のとおりです。

##### PROGRAM

パートナー LU に送信する割り振り要求に、ユーザー名とパスワードの両方が含まれるということを指定します。これがデフォルトです。

**NONE** パートナー LU に送信する割り振り要求に、セキュリティー情報が含まれないということを指定します。

**SAME** パートナー LU に送信する割り振り要求に、ユーザー名が含まれな



いということを指定します。これは、ユーザー名が「すでに検査済み」という標識で指定されます。パートナーは、「すでに検査済み」という保証を受け入れられるように構成されていなければなりません。

**注:** APPC を使用して DB2 (Windows NT 版) バージョン 7.1 (またはそれ以降) サーバーに接続する場合、8 バイトより長いユーザー ID は、SECURITY が NONE に指定されている場合に限りサポートされます。

#### **REMOTE\_INSTANCE instance-name**

リモート・サーバー・マシン上でアタッチを確立するインスタンスの実名を指定します。

#### **SYSTEM system-name**

サーバー・マシンを識別するために使用する名前を指定します。

#### **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、OS400、OS390、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 appc node db2appc1 remote db2inst1 security program
with "A remote APPC 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。

**注:** ディレクトリーをキャッシュできる場合、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参照の間に作成されます。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してから再始動してください。

**関連資料:**

- 428 ページの『GET DATABASE MANAGER CONFIGURATION』
- 542 ページの『LIST NODE DIRECTORY』
- 761 ページの『TERMINATE』



## CATALOG APPN NODE

通信プロトコルとして APPN を使用するリモート・ワークステーションに関する情報を、ノード・ディレクトリーに書き込みます。DB2 はこの情報を使用して、アプリケーションとこのノードでカタログ作成したりリモート・データベース間の接続を確立します。

このコマンドは、Windows、AIX、および Solaris でのみ使用可能です。

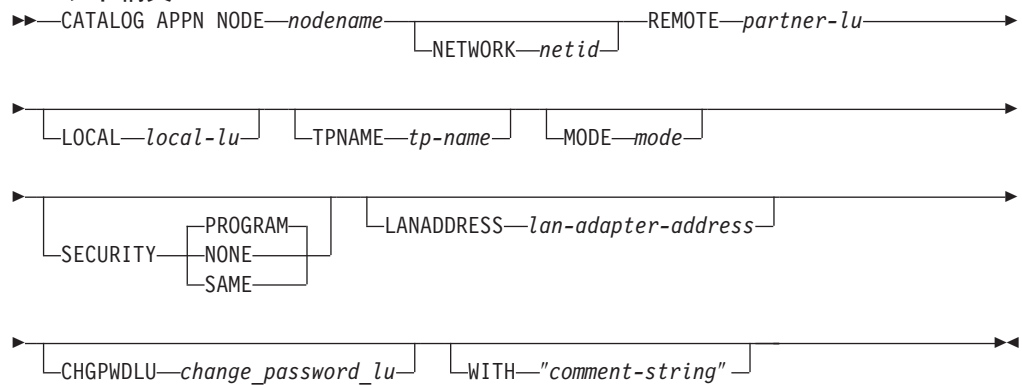
**権限:**

*sysadm*

**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*

カタログ作成するリモート・ワークステーションの名前を指定します。これは、(CATALOG DATABASE コマンドを使用して) カタログ化されるワークステーションにデータベースが常駐する場合に、ノード名パラメーターに入力したのと同じ名前です。名前は、DB2 命名規則に従っている必要があります。

### NETWORK *netid*

リモート LU が常駐する SNA ネットワークの ID を指定します。このネットワーク ID は、SNA の命名規則に従った 1 ～ 8 文字のストリングです。

### REMOTE *partner-lu*

接続で使った SNA パートナー LU を指定します。リモート・ノードの LU 名を入力してください。名前は、(コミュニケーション・マネージャー構成の) 対応する SNA 定義に現れるとおりに正確に入力する必要があります (大文字小文字を混合して使用)。名前は SNA 命名規則に従っている必要があります。

### LOCAL *local-lu*

接続に使った SNA ローカル LU の別名を指定します。これは 1 ～ 8

文字の非ブランク文字からなるストリングでなければなりません。別名は、(コミュニケーション・マネージャー構成の) 対応する SNA 定義に現れるとおりに正確に入力する必要があります (大文字小文字を混合して使用)。

### TPNAME **tp-name**

データベース・サーバーの APPC トランザクション・プログラム名を指定します。デフォルトは DB2DRDA です。

### MODE **mode**

接続で使った SNA 送信モードを指定します。名前は、SNA 命名規則に従っている必要があります。

この値を入力しないと、DB2 は、モード・タイプとして 8 文字のブランクを文字ストリングに保管します。

### SECURITY

パートナー LU に送信する割り振り要求において、セキュリティ情報を使用する度合いを指定します。有効な値は以下のとおりです。

#### PROGRAM

パートナー LU に送信する割り振り要求に、ユーザー名とパスワードの両方が含まれるということを指定します。これがデフォルトです。

**NONE** パートナー LU に送信する割り振り要求に、セキュリティ情報が含まれないということを指定します。

**SAME** パートナー LU に送信する割り振り要求に、ユーザー名が含まれないということを指定します。これは、ユーザー名が「すでに検査済み」という標識で指定されます。パートナーは、「すでに検査済み」という保証を受け入れられるように構成されていなければなりません。

注: APPN を使用して DB2 (Windows NT 版) バージョン 7.1 (またはそれ以降) サーバーに接続する場合、8 バイトより長いユーザー ID は、SECURITY が NONE に指定されている場合に限りサポートされます。

### LANADDRESS **lan-adapter-address**

DB2 サーバーの LAN アダプター・アドレス。

### CHGPWDLU **change\_password\_lu**

ホスト・データベース・サーバーのパスワード変更時に使用される、パートナー LU の名前を指定します。

### 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:

次の例は、APPN ノードをカタログ化します。

```
db2 catalog appn node remnode remote rlu with "Catalog APPN NODE"
```

関連資料:

- 330 ページの『CATALOG DATABASE』
- 294 ページの『ATTACH』

# 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。

**権限:**

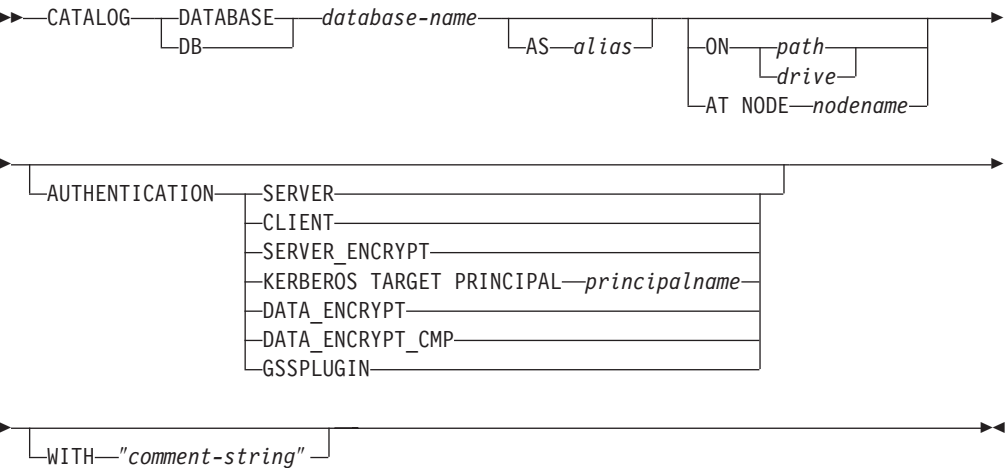
以下のどれかが必要です。

- *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**

UNIX ベースのシステムでは、カタログするデータベースが常駐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、カタログするデータベースが常駐するドライブ名を指定します。

**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、パフォーマンスの向上に役立ちます。

**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**

Kerberos セキュリティー・メカニズムを使用して行われる認証を指定します。認証が Kerberos の場合、アクセスには APPC 接続が使用され、SECURITY=NONE のみサポートされます。

**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、useridxxx.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。

**DATA\_ENCRYPT\_CMP**

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。このオプションは、データ暗号化をサポートしない下位レベルの製品との互換性のために提供されています。その場合、それらの製品は SERVER\_ENCRYPT を指定して接続できますが、ユーザー・データ

の暗号化はできません。データ暗号化をサポートする製品は、必ずデータ暗号化を使用することになります。

### GSSPLUGIN

認証が外部 GSS API ベースのプラグイン・セキュリティー・メカニズムを使用して行われることを指定します。認証が GSSPLUGIN の場合、アクセスには APPC 接続が使用され、SECURITY=NONE のみサポートされます。

### WITH "comment-string"

システム・データベース・ディレクトリー内のデータベースまたはデータベース項目について記述します。注釈列の最大長は 30 文字です。復帰文字や改行文字は許可されません。注釈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。

**注:** 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参照の間に作成されます。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してから再始動してください。

**関連資料:**

- 428 ページの『GET DATABASE MANAGER CONFIGURATION』
- 520 ページの『LIST DATABASE DIRECTORY』
- 761 ページの『TERMINATE』
- 762 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。これらのデータベースは、DB2 Connect などのアプリケーション・リクエスター (AR) を通してアクセスされます。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名前が 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、データベースが常駐するリモート・サーバーに SQL 要求を転送します。

### 権限:

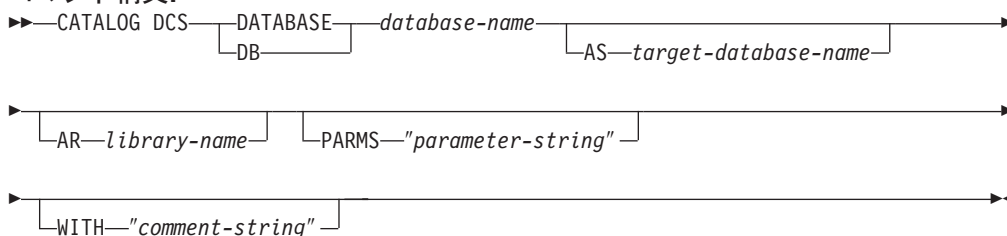
以下のどれかが必要です。

- *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。

**注:** 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。デフォルトで DB2 Connect を呼び出します。

DB2 Connect を使用しない場合は、AR のライブラリー名を指定します。そして、そのライブラリーをデータベース・マネージャー・ライブラリーと同じパスに置いてください。Windows オペレーティング・システムでは、そのパスは `drive:\sqllib\bin` です。UNIX ベースのシステムでは、パスはインスタンス所有者の `$HOME/sqllib/lib` になります。



**PARMS "parameter-string"**

呼び出したときに AR にパスされるパラメーター・ストリングを指定します。パラメーター・ストリングは、二重引用符で囲んでください。

**WITH "comment-string"**

DCS ディレクトリー項目について記述します。このディレクトリーにカタログするデータベースについて、任意の注釈を入力できます。最大長は 30 文字です。復帰文字や改行文字は許可されません。注釈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。

- システム/370 およびシステム/390 アーキテクチャー・ホスト・コンピューター上の DB2 for OS/390 and z/OS
- システム/370 およびシステム/390 アーキテクチャー・ホスト・コンピューター上の DB2 for VM and VSE
- 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。

**注:** ディレクトリーをキャッシュできる場合、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参照の間に作成されます。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してから再始動してください。

## CATALOG DCS DATABASE

### 関連資料:

- 428 ページの『GET DATABASE MANAGER CONFIGURATION』
- 761 ページの『TERMINATE』
- 764 ページの『UNCATALOG DCS DATABASE』
- 531 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。

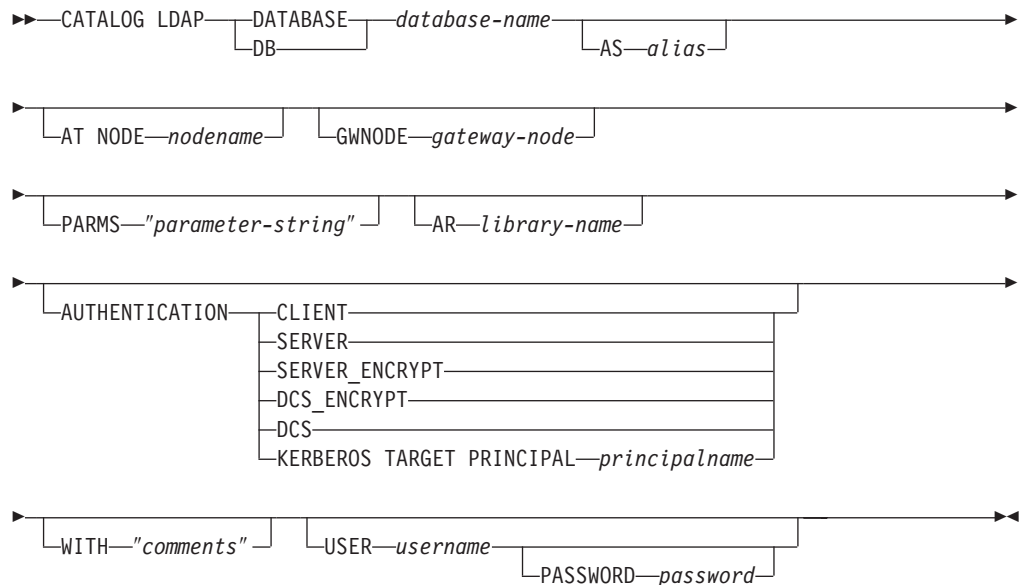
権限:

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と、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。

**注:** 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。

**注:** 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。

### 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

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。

### 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

Kerberos セキュリティー・メカニズムを使用して行われる認証を指定します。認証が Kerberos の場合、アクセスには APPC 接続が使用され、SECURITY=NONE のみサポートされます。

**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。

注: このパラメーターは Windows 2000 クライアント上でのみ有効です。

**WITH "comments"**

DB2 サーバーを記述します。ネットワーク・ディレクトリーで登録されるサーバー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・ユーザーの認証が使用されます。

注: 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。

注: 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、データベース名がすでに LDAP に存在する場合。この場合でもデータベースはローカル・マシン上に作成されます。

## CATALOG LDAP DATABASE

(かつローカル・アプリケーションによってアクセスできる) が、LDAP に存在する項目は新しいデータベースを反映して変更されることはありません。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。

### 関連資料:

- 661 ページの『REGISTER』
- 766 ページの『UNCATALOG LDAP DATABASE』
- 341 ページの『CATALOG LDAP NODE』
- 768 ページの『UNCATALOG LDAP NODE』
- 126 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。

**権限:**

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・ユーザーの認証が使用されます。

**PASSWORD** *password*

アカウント・パスワード。

**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。

**関連資料:**

- 337 ページの『CATALOG LDAP DATABASE』
- 766 ページの『UNCATALOG LDAP DATABASE』
- 768 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。

### 権限:

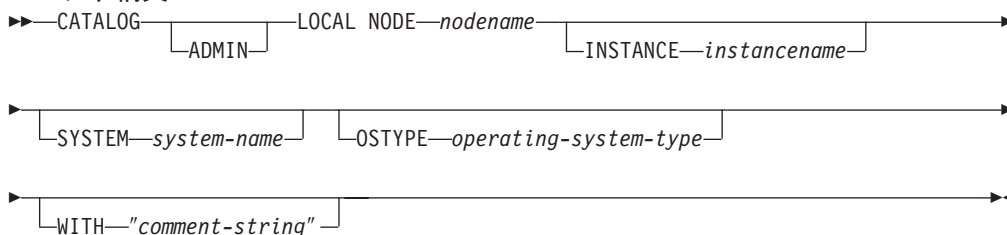
以下のいずれかが必要です。

- *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 *instancename*

アクセスするローカル・インスタンスの名前。

#### SYSTEM *system-name*

サーバー・マシンを識別するために使用する DB2 システム名を指定します。

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

サーバー・マシンのオペレーティング・システムのタイプを指定します。有効な値は次の通りです。AIX、WIN、HPUX、SUN、OS390、OS400、VM、VSE、SNI、SCO、LINUX、および DYNIX。

#### 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:



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:

**注:** ディレクトリーをキャッシュできる場合、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参照の間に作成されます。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してから再始動してください。

#### 関連資料:

- 428 ページの『GET DATABASE MANAGER CONFIGURATION』
- 761 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 のみで使用可能です。

### 権限:

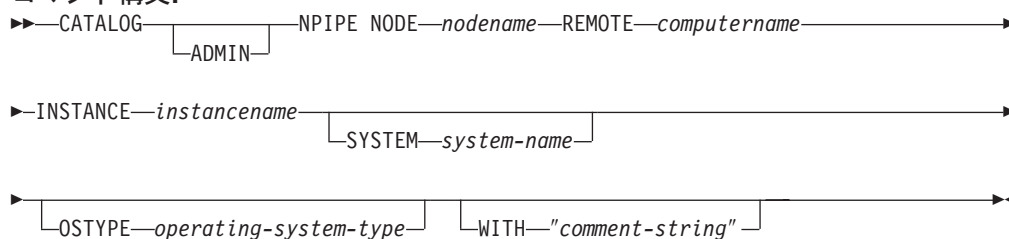
以下のいずれかが必要です。

- *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・データベースが常駐するノードのコンピューター名です。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

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。

**注:** 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参照の間に作成されます。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してから再始動してください。

**関連資料:**

- 428 ページの『GET DATABASE MANAGER CONFIGURATION』
- 542 ページの『LIST NODE DIRECTORY』
- 761 ページの『TERMINATE』

## CATALOG NETBIOS NODE

ノード・ディレクトリーに NetBIOS 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は、NetBIOS 通信プロトコルを使用します。

このコマンドは Windows のみで使用可能です。

### 権限:

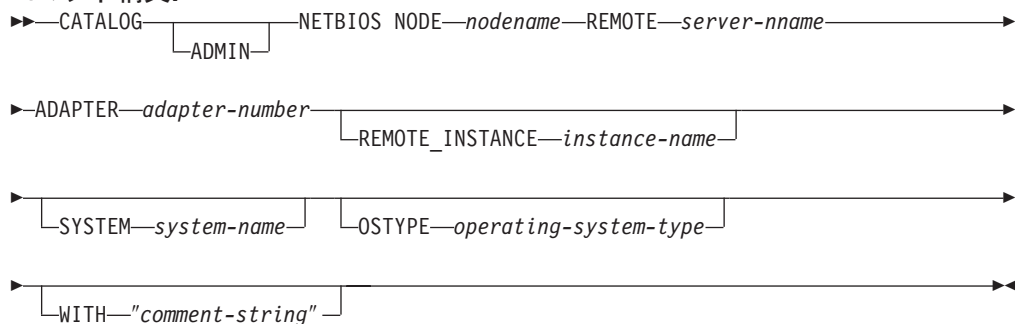
以下のいずれかが必要です。

- *sysadm*
- *sysctrl*

### 必要な接続:

なし。ディレクトリー操作は、ローカル・ディレクトリーだけに影響します。

### コマンド構文:



### コマンド・パラメーター:

#### ADMIN

Administration Server データベース・パーティションを指定します。

#### 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 server-nname

ターゲット・データベースが常駐するリモート・ワークステーションの名前。この名前は、データベース・マネージャーの命名規則に適合しなければなりません。これは、サーバー・ワークステーションのデータベース・マネージャー構成ファイルにあるワークステーション名 (*nname*) です。

#### ADAPTER adapter-number

ローカル、論理、出力 LAN アダプター番号を指定します。デフォルトはゼロです。

#### REMOTE\_INSTANCE instance-name

リモート・サーバー・マシン上でアタッチを確立するインスタンスの実名を指定します。

#### SYSTEM system-name

サーバー・マシンを識別するために使用する名前を指定します。

**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。サポートされているオペレーティング・システムで NetBios 接続が使用できるのは、現在のところ Windows だけです。したがって、現在 OSTYPE に使用できる唯一の値は、WIN 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:**

```
db2 catalog netbios node db2netb1 remote db2inst1 adapter 0
with "A remote NetBIOS 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。

**注:** ディレクトリーをキャッシュできる場合、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参照の間に作成されます。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してから再始動してください。

**関連資料:**

- 428 ページの『GET DATABASE MANAGER CONFIGURATION』
- 542 ページの『LIST NODE DIRECTORY』
- 761 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、指定した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可能です。

このコマンドは Windows プラットフォームでのみ使用可能です。

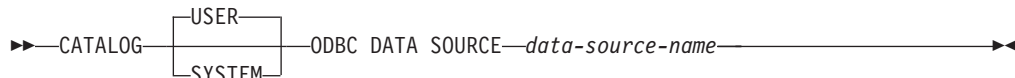
**権限:**

なし

**必要な接続:**

なし

**コマンド構文:**



**コマンド・パラメーター:**

**USER** ユーザー・データ・ソースをカタログします。キーワードを指定しない場合、これがデフォルトです。

**SYSTEM**

システム・データ・ソースをカタログします。

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

カタログするデータ・ソースの名前を指定します。最大長は 32 文字です。

**関連資料:**

- 545 ページの『LIST ODBC DATA SOURCES』
- 770 ページの『UNCATALOG ODBC DATA SOURCE』

## CATALOG TCPIP NODE

ノード・ディレクトリーに、伝送制御プロトコル/インターネット・プロトコル (TCP/IP) 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は、TCP/IP 通信プロトコルを使用します。 CATALOG TCPIP NODE コマンドは、クライアントで実行されます。

### 権限:

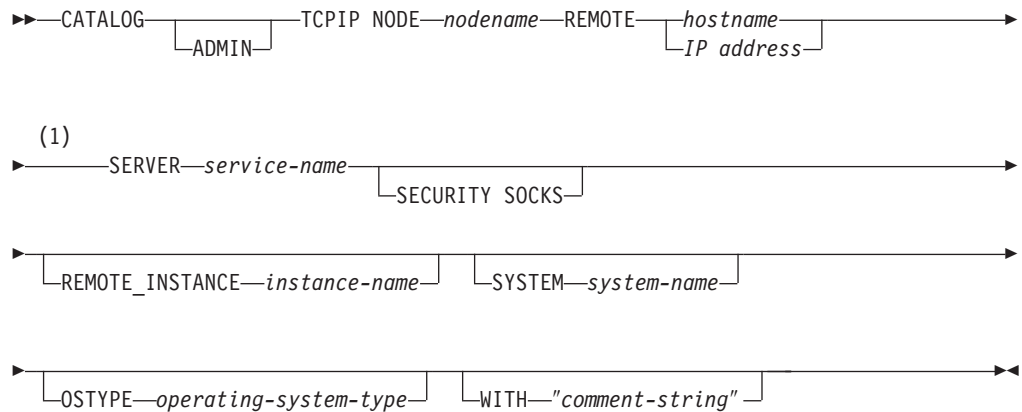
以下のどれかが必要です。

- *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:



### 注:

- 1 ADMIN ノードには SERVER を指定してはならない。ADMIN ノード以外には必須。

### コマンド・パラメーター:

#### ADMIN>

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

#### 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 hostname/IP address

ターゲット・データベースが常駐するノードのホスト名または IP アドレス。このホスト名とは、TCP/IP ネットワークに認識されているノードの名前のことです。ホスト名の最大長は 255 文字です。

### SERVER *service-name*

サーバー・データベース・マネージャー・インスタンスのサービス名またはポート番号を指定します。最大長は 14 文字です。このパラメーターには、大文字小文字の区別があります。

サービス名を指定する場合、クライアントのサービス・ファイルは、そのサービス名をポート番号にマップするのに使用されます。サービス名は、サーバーのデータベース・マネージャー構成ファイルで指定します。サーバーのサービス・ファイルは、このサービス名をポート番号にマップするのに使用されます。クライアントとサーバーのポート番号は一致していなければなりません。

サーバーのデータベース・マネージャー構成ファイルでサービス名の代わりにポート番号を指定することもできますが、お勧めできません。ポート番号を指定した場合、ローカル TCP/IP 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。

注: ADMIN ノードでは、このパラメーターを指定してはなりません。  
ADMIN ノードの値は、常に 523 です。

### SECURITY SOCKS

ノードを SOCKS 使用可能に指定します。

以下の環境変数は必須で、使用可能 SOCKS に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 IP アドレスでなければなりません。

#### SOCKS\_SERVER

完全修飾ホスト名または SOCKS サーバーの IP アドレスです。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、IP アドレスがすで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でなければならない。
- hosts ファイル中にリストされていなければならない。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。
- IP アドレス形式でなければならない。

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*

アタッチを確立するサーバー・インスタンスの名前を指定します。



**SYSTEM** *system-name*

サーバー・マシンを識別するために使用する DB2 システム名を指定します。これは、物理マシン、サーバー・システム、またはワークステーションの名前です。

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

サーバー・マシンのオペレーティング・システムのタイプを指定します。有効な値は次のとおりです。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 tcpip node db2tcp1 remote tcpghost server db2inst1
with "A remote TCP/IP node"

db2 catalog tcpip node db2tcp2 remote 9.21.15.235 server db2inst2
with "TCP/IP node using IP address"
```

**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。

**注:** ディレクトリーをキャッシュできる場合、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参照の間に作成されます。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してから再始動してください。

**関連資料:**

- 428 ページの『GET DATABASE MANAGER CONFIGURATION』
- 542 ページの『LIST NODE DIRECTORY』

## CATALOG TCPIP NODE

- 761 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。現行の注釈関連テキストは、新規の注釈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。

### 権限:

以下のどれかが必要です。

- *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

UNIX ベースのシステムでは、データベースが常駐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、データベースが常駐するドライブ名を指定します。

#### WITH "comment-string"

システム・データベース・ディレクトリーまたはローカル・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、任意の注釈を入力することができます。注釈列の最大長は 30 文字です。復帰文字や改行文字は許可されません。注釈テキストは必ず二重引用符で囲んでください。

### 例:

## CHANGE DATABASE COMMENT

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:

既存の注釈テキストは、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、既存の注釈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注釈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:

- 357 ページの『CREATE DATABASE』

## CHANGE ISOLATION LEVEL

データベースのアクセス中に、DB2 が他の処理からデータを分離する方法を変更します。

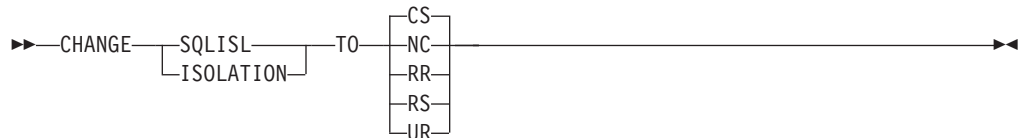
**権限:**

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。

## CHANGE ISOLATION LEVEL

次の例では、SAMPLE データベースの作成に続いて、ユーザーが DB2 対話式モードに入っています。

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

set client connect 2

connect to sample
connect to sample2

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

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

c1 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 がすでに宣言されて 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:

- 732 ページの『SET CLIENT』
- 634 ページの『QUERY CLIENT』

## CREATE DATABASE

新しいデータベースをオプションのユーザー定義照合順序で初期化し、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・ログを割り振ります。新規のデータベースを初期化するとき、**AUTOCONFIGURE** オプションを指定してバッファ・プール・サイズ、データベース、およびデータベース・マネージャーのパラメーターの初期値を表示することができます (オプションでこれを適用することもできます)。**AUTOCONFIGURE** オプションはパーティション・データベース環境では使用できません。

このコマンドはクライアントでは無効です。

### 有効範囲:

パーティション・データベース環境では、このコマンドは、`db2nodes.cfg` ファイルにリストされているすべてのデータベース・パーティションに影響を与えます。

このコマンドを発行したデータベース・パーティションは、新規データベースのカatalog・パーティションになります。

### 権限:

以下のどれかが必要です。

- `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 database-name
    [ AT DBPARTITIONNUM ]
    [ Create Database options ]
  
```

### Create Database options:

```

    [ ON path drive ]
    [ ALIAS database-alias ]

    [ USING CODESET codeset TERRITORY territory ]
    [ COLLATE USING
        SYSTEM
        COMPATIBILITY
        IDENTITY
        IDENTITY_16BIT
        UCA400_NO
        UCA400_LTH
        NLSCHAR ]

    [ NUMSEGS numsegs ]
    [ DFT_EXTENT_SZ dft_extentsize ]

    [ CATALOG TABLESPACE tblspace-defn ]
    [ USER TABLESPACE tblspace-defn ]
  
```

## CREATE DATABASE



### 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はありません。
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. パーティション・データベース環境では、AUTOCONFIGURE オプションを使用すると CREATE DATABASE コマンドが失敗します。パーティション・データベース環境で AUTOCONFIGURE オプションを使用したい場合、最初に AUTOCONFIGURE オプションを指定しないでデータベースを作成してから、パーティションごとに AUTOCONFIGURE コマンドを実行してください。
5. AUTOCONFIGURE オプションには、sysadm 権限が必要です。

### コマンド・パラメーター:

#### DATABASE database-name

新しいデータベースに割り当てられる名前。ローカル・データベース・ディ



レクトリーまたはシステム・データベース・ディレクトリーの他のデータベースから、そのデータベースを区別する、ユニークな名前であればなりません。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。

#### 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。

CREATE DATABASE コマンドに AT DBPARTITIONNUM オプションを指定して使用した後、この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。たとえば、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。

#### ON path/drive

UNIX ベースのシステムでは、データベースを作成するパスを指定します。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。最大長は 205 文字です。Windows オペレーティング・システムでは、データベースを作成するドライブの文字を指定します。

**注:** 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 (たとえば、UNIX ベースのシステムの場合は、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。

#### ALIAS database-alias

システム・データベース・ディレクトリーのデータベースに付けられる別名。別名が付けられないと、指定されたデータベース名が使用されます。

#### USING CODESET codeset

このデータベースに入るデータに使用するコード・セットを指定します。データベースを作成した後は、指定のコード・セットを変更できません。

#### TERRITORY territory

このデータベースに入るデータに使用するテリトリーを指定します。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、照合順序を認識します。

**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](http://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。この仕様は Unicode Consortium Web サイト ([www.unicode.org](http://www.unicode.org)) で入手できます。このオプションは、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](http://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に注意してください。

**NLSCHAR**

特定のコード・セット/テリトリ用のユニークな照合規則を使用するシステム定義の照合シーケンス。

**注:** 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**

データベースのテリトリに基づいた照合順序。このオプションは、Unicode データベースを作成する場合には指定できません。

**NUMSEGS numsegs**

デフォルト SMS 表スペース用に DAT、IDX、LF、LB、および LBA ファイルを保管する際に作成および使用するセグメント・ディレクトリーの数を指定します。このパラメーターは、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。

**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。指定しないと、SYSCATSPACE はディレクトリーの *numsegs* 数をコンテナーとして使用し、また *dft\_extentsize* のエクステント・サイズを使用して、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。たとえば、*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
```

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・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。指定しないと、USERSPACE1 はコンテナーとしてディレクトリーの *numsegs* 数を使用し、また *dft\_extentsize* のエクステント・サイズを使用して、SMS 表スペースとして作成されます。たとえば、*numsegs* に 5 を指定した場合、以下のコンテナーが作成されます。

```
/u/smith/smith/NODE0000/SQL00001/SQLT0001.0
/u/smith/smith/NODE0000/SQL00001/SQLT0001.1
/u/smith/smith/NODE0000/SQL00001/SQLT0001.2
/u/smith/smith/NODE0000/SQL00001/SQLT0001.3
/u/smith/smith/NODE0000/SQL00001/SQLT0001.4
```

**TEMPORARY TABLESPACE tblspace-defn**

初期 SYSTEM TEMPORARY 表スペース、TEMPSPACE1 の定義を指定します。指定しないと、TEMPSPACE1 はディレクトリーの *numsegs* 数をコンテナーとして使用し、また *dft\_extentsize* のエクステント・サイズを使用して、SMS 表スペースとして作成されます。たとえば、*numsegs* に 5 を指定した場合、以下のコンテナーが作成されます。

```
/u/smith/smith/NODE0000/SQL00001/SQLT0002.0
/u/smith/smith/NODE0000/SQL00001/SQLT0002.1
/u/smith/smith/NODE0000/SQL00001/SQLT0002.2
/u/smith/smith/NODE0000/SQL00001/SQLT0002.3
/u/smith/smith/NODE0000/SQL00001/SQLT0002.4
```

**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

### USING input-keyword param-value

表 7. 有効な入力キーワードおよびパラメーター値

キーワード	有効値	デフォルト値	説明
mem_percent	1-100	25	専用にするメモリーのパーセンテージ。他のアプリケーション (オペレーティング・システム以外) がこのサーバーで実行している場合、この値は 100 未満に設定してください。
workload_type	simple、mixed、complex	mixed	単純 (simple) ワークロードは入出力集約の傾向があり大部分がトランザクションであるのに対し、複雑 (complex) ワークロードは CPU 集約の傾向があり大部分が照会です。
num_stmts	1-1 000 000	25	作業単位ごとのステートメント数
tpm	1-200 000	60	1 分ごとのトランザクション
admin_priority	performance、recovery、both	both	より良いパフォーマンス (分あたりの多いトランザクション数) またはより良いリカバリー時間のための最適化
num_local_apps	0-5 000	0	接続されたローカル・アプリケーションの数
num_remote_apps	0-5 000	100	接続されたリモート・アプリケーションの数
isolation	RR、RS、CS、UR	RR	このデータベースに接続するアプリケーションの分離レベル (反復可能読み取り (RR)、読み取り固定 (RS)、カーソル固定 (CS)、非コミット読み取り (UR))
bp_resizeable	yes、no	yes	バッファ・プールのサイズが変更可能かどうか。

## APPLY

## DB ONLY

現行のデータベース・マネージャの構成に基づいて、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。

## 使用上の注意:

## CREATE DATABASE コマンド

- 指定し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・パーティションで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、`$DB2INSTANCE/NODExxxx` ディレクトリーを作成してください。非パーティション環境では、指定されたサブディレクトリーの下に `$DB2INSTANCE/NODE0000` ディレクトリーを作成してください。
- 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・ディレクトリーでデータベースをカタログします。
  - `path` によって示されるパスにサーバーのローカル・データベース・ディレクトリー、または、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るデフォルトのデータベース・パス。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。照合順序がユニーク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・パーティションになります。**IBMDEFAULTGROUP** および **IBMCATGROUP** の 2 つのデータベース・パーティションが自動的に作成されます。

## CREATE DATABASE

- 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、ユーザーは訂正の処置を取り、手動で失敗したファイルをバインドできます。データベースはどのような場合にでも作成されます。`CREATEIN` 特権が `PUBLIC` に付与されたバインドの実行時に、`NULLID` と呼ばれるスキーマが暗黙的に作成されます。

**注:** ユーティリティ・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・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` スキーマ内のすべての機能上の `PUBLIC` に対する `EXECUTE WITH GRANT` 権限
  - `SYSIBM` スキーマ内のすべてのプロシージャ上の `PUBLIC` に対する `EXECUTE` 権限
  - `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・カタログ上の `SELECT` 特権を `PUBLIC` に。
  - 正常にバインドされたユーティリティに対する `BIND` および `EXECUTE` 特権を `PUBLIC` に。
  - `SYSFUN` スキーマ内のすべての機能上の `PUBLIC` に対する `EXECUTE WITH GRANT` 権限。
  - `SYSIBM` スキーマ内のすべてのプロシージャ上の `PUBLIC` に対する `EXECUTE` 権限。

`dbadm` 権限を使用すると、これらの権限を他のユーザーまたは `PUBLIC` に `GRANT` (または取り消し) することができます。データベース上の `sysadm` 権限、または `dbadm` 権限を持つ別の管理者がそれらの特権を取り消す場合でも、データベース作成者はそれらを保存します。



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の \$DB2INSTANCE/NODExxxx を作成します。xxxx は db2nodes.cfg ファイルで定義されたデータベース・パーティション番号です (つまり、データベース・パーティション 0 が NODE00000 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使用可能である場合、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・オブジェクトがすでに 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すで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。

#### 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: プランニング」の『DB2 Universal Database 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: プランニング」の『タイ語文字の照合』
- 「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』

#### 関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE TABLESPACE ステートメント』
- 「管理 API リファレンス」の『sqlcra - データベースの作成』
- 305 ページの『BIND』
- 330 ページの『CATALOG DATABASE』
- 380 ページの『DROP DATABASE』

## CREATE DATABASE

- 697 ページの『RESTORE DATABASE』
- 337 ページの『CATALOG LDAP DATABASE』
- 296 ページの『AUTOCONFIGURE』



## 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。

**注:** 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。

### 権限:

以下のどれかが必要です。

- 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、ユニークな名前であればなりません。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、スケジューラーを使用可能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 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 サーバーで使用可能にします。
- 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 オペレーティング環境) 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『DB2 Administration Server』

#### 関連資料:

- 「管理ガイド: パフォーマンス」の『jdk\_path - 「Java Development Kit インストール・パス DAS」構成パラメーター』
- 「管理ガイド: パフォーマンス」の『jdk\_64\_path - 「64 ビット Java Development Kit 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します。

### 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パーティション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。

### 権限:

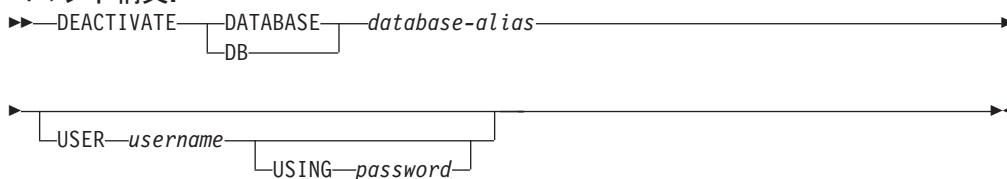
以下のいずれかが必要です。

- *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** によって非活動化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非活動化されないため、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非活動化することもできます。)

**注:** 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。

### 関連資料:

- 752 ページの『STOP DATABASE MANAGER』
- 283 ページの『ACTIVATE DATABASE』

## DEREGISTER

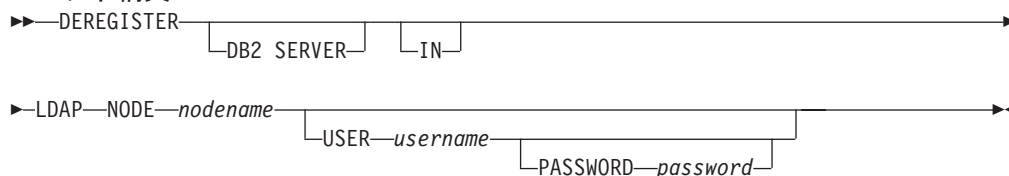
DB2 サーバーの登録をネットワーク・ディレクトリー・サーバーから取り消します。

**権限:**

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・ユーザーの認証が使用されます。

**PASSWORD password**

アカウント・パスワード。

**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。

**関連資料:**

- 661 ページの『REGISTER』
- 796 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 または CALL 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

### 権限:

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または索引を表示するには、システム・カタログ SYSCAT.COLUMNS (DESCRIBE TABLE) および SYSCAT.INDEXES (DESCRIBE INDEXES FOR TABLE) に対して、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。

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

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。

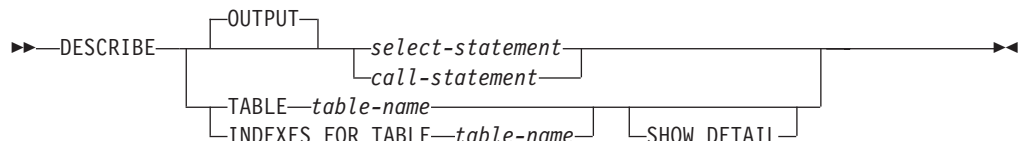
CALL ステートメントに関する出力 SQLDA 情報を表示するには、下にリストされている特権または権限のいずれかが必要です。

- 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 または call-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。

## DESCRIBE

### TABLE *table-name*

記述する表またはビューを指定します。完全修飾の名前の書式は、*schema.table-name* 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
- 位取り
- 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。完全修飾の名前の書式は、*schema.table-name* 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
- ユニークな規則
- 列カウント

### SHOW DETAIL

DESCRIBE TABLE コマンドの場合、出力に以下の追加情報を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
- パーティション・キー・シーケンス
- コード・ページ
- デフォルト

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

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		

**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

**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
```

## DESCRIBE

Table: USER1.DEPARTMENT

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

---

## DETACH

論理 DBMS インスタンス・アタッチを除去し、この層を使用した論理接続がほか  
にない場合、物理通信接続を終了します。

**権限:**

なし

**必要な接続:**

なし。 既存のインスタンス・アタッチを除去します。

**コマンド構文:**

▶▶—DETACH—◀◀

**コマンド・パラメーター:**

なし

**関連資料:**

- 294 ページの『ATTACH』

---

### 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。

**権限:**

なし。

**必要な接続:**

なし。

**コマンド構文:**

►►—DROP CONTACT—*name*—————►◄

**コマンド・パラメーター:**

**CONTACT name**

ローカル・システムからドロップされる連絡先の名前。

---

## DROP CONTACTGROUP

ローカル・システムで定義された連絡先のリストから、連絡先グループを除去します。連絡先グループには、スケジューラーおよびヘルス・モニターがメッセージを送信する先のユーザーのリストが入っています。

**権限:**

なし。

**必要な接続:**

なし。

**コマンド構文:**

►►—DROP CONTACTGROUP—*name*—————►◄

**コマンド・パラメーター:**

**CONTACTGROUP *name***

ローカル・システムからドロップされる連絡先グループの名前。

## DROP DATABASE

データベースの内容とそのすべてのログ・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。

### 権限:

以下のどれかが必要です。

- `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 ➡
```

### コマンド・パラメーター:

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

ドロップするデータベースの別名を指定します。データベースはシステム・データベース・ディレクトリー内にカタログされている必要があります。

#### **AT DBPARTITIONNUM**

DROP DATABASE コマンドを発行したデータベース・パーティションだけで、データベースが削除されることを指定します。このパラメーターは DB2 ESE に付属のユーティリティーが使用するもので、汎用ではありません。このパラメーターの使用が不適切であると、システム内に不整合が生じることもありえますので、ご使用の際には十分注意してください。

### 例:

次の例は、データベース別名 `SAMPLE` で参照されるデータベースを削除します。

```
db2 drop database sample
```

### 使用上の注意:

DROP DATABASE はすべてのユーザー・データとログ・ファイル、およびデータベースのバック/リストア履歴を削除します。リストア操作後のロールフォワード・リカバリーにログ・ファイルが必要である場合、またはデータベースのリストアにバックアップ履歴が必要である場合、このコマンドを実行する前にそれらのファイルを保管しておく必要があります。

データベースは使用中であってはなりません。データベースをドロップできるようになるには、すべてのユーザーがデータベースから切断してい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化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** が使用可能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。

**関連資料:**

- 330 ページの『**CATALOG DATABASE**』
- 357 ページの『**CREATE DATABASE**』
- 762 ページの『**UNCATALOG DATABASE**』

## DROP DATALINKS MANAGER

DB2 Data Links Manager を、指定されたデータベースの登録済み DB2 Data Links Manager のリストからドロップします。

### 権限:

以下のどれかが必要です。

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

### コマンド構文:

```
►► DROP DATALINKS MANAGER FOR DATABASE dbname USING name ◀◀
```

DB

### コマンド・パラメーター:

#### DATABASE *dbname*

データベース名を指定します。

#### USING *name*

LIST DATALINKS MANAGER コマンドによって表示された通りに DB2 Data Links Manager サーバーの名前を指定します。

### 例:

#### 例 1

いくつかのデータベース表に `micky.almaden.ibm.com` へのリンクがあるときに、ホスト `bramha.almaden.ibm.com` に存在するインスタンス `VALIDATE` の下で、データベース `TEST` から DB2 Data Links Manager `micky.almaden.ibm.com` をドロップするには、次のようにします。

1. データベース `TEST` のデータベース・バックアップを取ります。
2. `micky.almaden.ibm.com` へのリンクがある場合は、次のようにしてそのリンクを解除します。
  - a. `SYSADM_GROUP` に属するユーザー ID でログオンし、次のコマンドを発行してデータベース `TEST` への排他モード接続を取得します。

```
connect to test in exclusive mode
```

この接続が、上記のユーザー ID を使用した `TEST` への唯一の接続であるようにします。これにより、リンクが新たに作成されるのを防ぎます。

- b. 次のコマンドを発行して、すべての `FILE LINK CONTROL DATALINK` 列のリストと、その列を含むデータベース内の表を取得します。

```
select tabname, colname from syscat.columns where
substr(dl_features, 2, 1) = 'F'
```



- c. リスト内の FILE LINK CONTROL DATALINK 列ごとに SQL SELECT を発行して、micky.almaden.ibm.com へのリンクが存在するかどうかを判別します。たとえば、表 t の DATALINK 列 c の場合、SELECT ステートメントは次のようになります。

```
select count(*) from t where dlurlserver(t.c) = 'MICKY.ALMADEN.IBM.COM'
```

- d. リンクを含む FILE LINK CONTROL DATALINK 列ごとに SQL UPDATE を発行して、micky.almaden.ibm.com にリンクされている値をリンク解除します。たとえば、表 t の DATALINK 列 c の場合、UPDATE ステートメントは次のようになります。

```
update t set t.c = null where dlurlserver(t.c) = 'MICKY.ALMADEN.IBM.COM'
```

t.c が NULL 可能でない場合は、以下を使用できます。

```
update t set t.c = dlvalue('') where dlurlserver(t.c)
= 'MICKY.ALMADEN.IBM.COM'
```

- e. SQL UPDATE をコミットします。

```
commit
```

3. DROP DATALINKS MANAGER コマンドを発行します。

```
drop datalinks manager for db test using micky.almaden.ibm.com
```

4. 変更を有効にしてデータベースへの他の接続を許可するために、排他モード接続を終了します。

```
terminate
```

5. micky.almaden.ibm.com の TEST のために、バックアップ情報のリンク解除処理とガーベッジ・コレクションを開始します。DB2 Data Links Manager 管理者として、micky.almaden.ibm.com で次のコマンドを発行してください。

```
dlfm drop_dlm test validate bramha.almaden.ibm.com
```

これにより、ステップ 3 を呼び出す前のリンク解除をユーザーが忘れていた場合に、データベース TEST にまだリンクされているすべてのファイルがリンク解除されます。前にデータベース TEST にリンクされていたファイルのバックアップ情報 (アーカイブ・ファイル、メタデータなど) が micky.almaden.ibm.com にある場合、このコマンドはバックアップ情報のガーベッジ・コレクションを開始します。実際のリンク解除とガーベッジ・コレクションは非同期で実行されます。

## 例 2

DB2 Data Links Manager は、ドロップ後に再登録でき、それを新しいまったく別の DB2 Data Links Manager として扱えます。micky.almaden.ibm.com をドロップするために例 1 のステップを実行した場合は、古いバージョンへのリンクは存在しなくなります。そのようにしない場合は、ユーザーは下記のステップ 7 に示されているエラー SQL0368 を受け取ります。DB2 Data Links Manager を再登録するためのステップは、次のとおりです。

1. micky.almaden.ibm.com をデータベース TEST に登録します。

```
add datalinks manager for db test using node micky.almaden.ibm.com
port 14578
```

## DROP DATALINKS MANAGER

2. micky.almaden.ibm.com のファイルへのリンクを作成します。

```
connect to test
create table t(c1 int, c2 datalink linktype url file link
control mode db2options)
insert into t
values(1, dlvalue('file://micky.almaden.ibm.com/pictures/yosemite.jpg'))
commit
terminate
```

3. micky.almaden.ibm.com をデータベース TEST からドロップします。

```
drop datalinks manager for db test using micky.almaden.ibm.com
```

4. DATALINK 値を選択します。

```
connect to test
select * from t
terminate
```

ユーザーに対しては以下が表示されます。

```
SQL0368 The DB2 Data Links Manager "MICKY.ALMADEN.IBM.COM"
is not registered to the database. SQLSTATE=55022.
```

5. micky.almaden.ibm.com をもう一度データベース TEST に登録します。

```
add datalinks manager for db test using node micky.almaden.ibm.com
port 14578
```

6. DATALINK 値をさらに追加します。

```
connect to test
insert into t
values(2, dlvalue('file://micky.almaden.ibm.com/pictures/tahoe.jpg'))
commit
```

7. DATALINK 値を選択します。

```
select c2 from t where c1 = 2
```

このコマンドは、選択されている値が、micky.almaden.ibm.com の現在登録されているバージョンへのリンクなので成功します。

### 使用上の注意:

DROP DATALINKS MANAGER コマンドによる影響はロールバックできません。DROP DATALINKS MANAGER コマンドを使用するときは、例 1 で示したステップを行うことが重要です。

このコマンドは、データベースからすべてのアプリケーションが切断された後のみ有効です。

このコマンドが正常に完了すると、DB210201I メッセージによって、DB2 Data Links Manager で何も処理が行われていないことが通知されます。

DB2 Data Links Manager をドロップする前に、その DB2 Data Links Manager のファイルへのリンクがデータベースに含まれていないことを確認する必要があります。DB2 Data Links Manager がドロップされた後もリンクが存在する場合は、リンクを削除するために調整ユーティリティーを実行してください。これによって、NULL 可能リンクが NULL に設定され、NULL 不可能リンクが長さゼロの DATALINK 値に設定されます。このような値を含む行は例外表に挿入されます。DATALINK 値には元の接頭部名は組み込まれなくなります。この名前は、Data Links Manager がドロップされた後は使用できません。

データベースとドロップされた DB2 Data Links Manager との間のリンクに対応するファイルは、リンクされたままの状態になり、これらのファイルには、読み取り、書き込み、名前変更、削除、許可の変更、または所有権の変更などの操作ではアクセスできません。

DB2 Data Links Manager でリンク解除されたファイルのアーカイブされたコピーに対するガーベッジ・コレクションは、このコマンドでは行われません。ユーザーは、DB2 Data Links Manager で `dlfm drop_dlm` コマンドを使用して、リンク解除処理とガーベッジ・コレクションを明示的に開始することができます。

DB2 Data Links Manager をドロップする前に、データベースのバックアップを取っておくことをお勧めします。さらに、すべての複製サブスクリプションによって DB2 Data Links Manager に関連するすべての変更が複製されていることを確認してください。

DB2 Data Links Manager をデータベースからドロップする前にバックアップを取り、そのバックアップ・イメージを DB2 Data Links Manager のドロップ後のリストアに使用した場合には、リストアまたはロールフォワード処理によって特定の表がデータ・リンク調整ペンディング (DRP) 状態になる可能性があります。この場合、RECONCILE または **db2\_recon\_aid** ユーティリティーを実行し、DB2 データベースと Data Links Manager 上に保管されたファイルの間の不整合がないか確認して、もしあればそれを修復する必要があります。

**関連資料:**

- 526 ページの『LIST DATALINKS MANAGERS』
- 287 ページの『ADD DATALINKS MANAGER』

---

## 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、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・パーティションに対してだけ影響を与えます。

### 権限:

*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:

- 752 ページの『STOP DATABASE MANAGER』
- 656 ページの『REDISTRIBUTE DATABASE PARTITION GROUP』

---

## 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ルができなくなり、スケジュール済みのタスクも実行されません。スケジューラーを活動化するには、前のツール・カタログを活動化するか、新しいツール・カタログを作成することが必要です。

### 有効範囲:

このコマンドはデータベースに影響します。

### 権限:

以下のどれかが必要です。

- sysadm
- sysctrl

また、ユーザーには DB2 Administration Server (DAS) 構成パラメーターを更新するための、DASADM 権限も必要です。

### 必要な接続:

このコマンドの処理中、データベース接続が一時的に確立されます。

### コマンド構文:

►►—DROP TOOLS CATALOG—*catalog-name*—IN DATABASE—*database-name*—┐FORCE┘—►►

### コマンド・パラメーター:

#### CATALOG *catalog-name*

DB2 ツール・カタログを固有に識別するのに使用される名前。カタログ表はこのスキーマからドロップされます。

#### DATABASE *database-name*

カタログ表を含むローカル・データベースに接続するのに使用される名前。

#### FORCE

force オプションは、DB2 Administration Server のスケジューラーを強制的に停止させるのに使用されます。このオプションを指定しないと、スケジューラーが停止できない場合、ツール・カタログはドロップされません。

### 例:

```
db2 drop tools catalog cc in database toolsdb
db2 drop tools catalog in database toolsdb force
```

### 使用上の注意:

- *jdk\_path* 構成パラメーターは、DB2 Administration Server (DAS) 構成で、最小サポート・レベルの Java 用 SDK に設定する必要があります。

- このコマンドは、ローカル DAS でスケジューラーを使用不可にし、DAS ツール・カタログ・データベース構成に関連する DAS 構成パラメーターをリセットします。

---

## ECHO

ユーザーが文字ストリングを標準出力に書き込めるようにします。

**権限:**

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。さらに具体的に言えば、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。

### 権限:

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 プラットフォームではメモ帳、UNIX ベース・プラットフォームでは 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:

| • 482 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・フォーマットのどれかにデータをエクスポートします。ユーザーは、SQL SELECT ステートメントを提供するか、タイプ表の階層情報を提供して、エクスポートするデータを指定します。

### 権限:

以下のどれかが必要です。

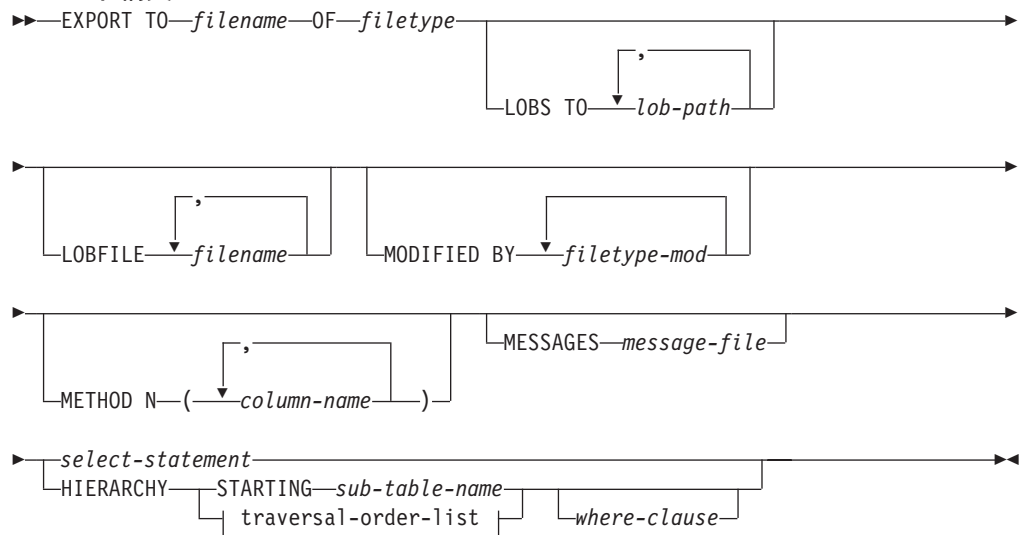
- *sysadm*
- *dbadm*

または、関係する各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。

### 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。最初の名前の名前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 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) に現行の基本名を追加し、それに 3 桁のシーケンス番号を追加したファイル名が構成されます。たとえば、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`、`/u/foo/lob/path/bar.002` (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 が組み込まれます。

**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。宛先ファイルがすで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

**注:** 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 ステートメントで列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** ステートメントを指定します。**SELECT** 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在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。

すで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。

### 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、まず **SAMPLE** データベースと接続しなければなりません。また、表の中の実際の列名は、**'department'** ではなく **'dept'** であることにも注意してください。

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

次の例は **LOB** を **DEL** ファイルにエクスポートする方法を示しています。

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

次の例は **LOB** を **DEL** ファイルにエクスポートする方法を示しています。ここでは、最初のディレクトリーにファイルを入れることができない場合のために 2 番目のディレクトリーを指定しています。

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

次の例はデータを **DEL** ファイルにエクスポートする方法を示しています。ここでは、単一引用符をストリング区切り文字として使用し、セミコロンを列の区切り文字として使用し、コンマを小数点として使用します。データを再びデータベースにインポートする場合、これと同じ規則を使用する必要があります。

```
db2 export to myfile.del of del
modified by charde1'' 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・ファイルに置かれるメッセージには、メッセージ検索サービスから戻される情報が入っています。各メッセージは改行してから始まります。

DEL フォーマット・ファイルへエクスポートするために 254 よりも長い文字データの列が選択されると、エクスポート・ユーティリティーは警告メッセージを生成します。

PC/IXF インポートはデータベース間でデータを移動する場合に使用します。行区切り文字の入った文字データを区切り文字付き ASCII (DEL) ファイルにエクスポートし、テキスト転送プログラムにより処理を行うと、行区切り文字の入ったフィールドは長さが伸縮します。

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・ユーティリティーは、AIX システムから呼び出される場合、複数部分からなる PC/IXF ファイルを作成しません。

エクスポート・ユーティリティーは、提供される SELECT ステートメントが、SELECT \* FROM tablename という形式である場合、IXF ファイルの表の NOT NULL WITH DEFAULT 属性を保管します。

タイプ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・フォーマットの場合は、走査順序リストを指定するようお勧めします。このリストは、階層を走査する方法やエクスポートする副表を DB2 に指示します。このリストを指定しない場合、階層内のすべての表がエクスポートされ、デフォルトの順序は OUTER 順序になります。または、OUTER 関数によって指定される順序である、デフォルトの順序を使用することができます。

**注:** インポート操作の間も同じ走査順序を使用します。ロード・ユーティリティーは、階層や副階層のロードをサポートしていません。

### DB2 Data Links Manager の考慮事項:

エクスポート時に、整合のとれた表のコピーと、 DATALINK 列によって参照される対応するファイルが確実にコピーされるようにするには、以下のようにします。

1. QUIESCE TABLESPACES FOR TABLE tablename SHARE コマンドを発行する。

これで、EXPORT の実行時に更新トランザクションが進行しなくなります。

2. EXPORT コマンドを発行します。

3. 各データ・リンク・サーバーで **dlfm\_export** ユーティリティを実行します。  
**dlfm\_export** ユーティリティへの入力制御ファイル名です。これは、エクスポート・ユーティリティによって生成されます。これにより、制御ファイル内にリストされるファイルの tar (または同等の) アーカイブが作成されます。

4. QUIESCE TABLESPACES FOR TABLE tablename RESET コマンドを発行する。

これにより、表は更新に使用できるようになります。

EXPORT は、SQL アプリケーションとして実行されます。SELECT ステートメントを満たす行と列がデータベースから抽出されます。DATALINK 列の場合、SELECT ステートメントはスカラー関数を指定できません。

EXPORT が正常に実行されると、以下のファイルが生成されます。

- EXPORT コマンドで指定したエクスポート・データ・ファイル。このファイルの DATALINK 列値は、インポートおよびロード・ユーティリティによって使用されるフォーマットと同じです。DATALINK 列の値が SQL NULL 値である場合、他のデータ・タイプと同じ処理が行われます。
- 各データ・リンク・サーバー用に生成される制御ファイル *server\_name*。  
Windows オペレーティング・システムでは、単一制御ファイル、*ctrlfile.lst* がすべてのデータ・リンク・サーバーによって使用されます。これらの制御ファイルは、ディレクトリー <data-file path>%dlfm %YYYYMMDD%HHMMSS に入れられます (Windows NT オペレーティング・システムの場合、*ctrlfile.lst* はディレクトリー <data-file path> %dlfm%YYYYMMDD%HHMMSS に入れられます)。YYYYMMDD は日付 (年月日)を、HHMMSS は時刻 (時、分、秒) を表します。

ファイルをデータ・リンク・サーバーからエクスポートするため、**dlfm\_export** ユーティリティが提供されています。このユーティリティが生成するアーカイブ・ファイルを使用して、ターゲットのデータ・リンク・サーバーにファイルをリストアすることができます。

#### 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『エクスポートの使用』

#### 関連資料:



- 「管理 API リファレンス」の『db2Export - エクスポート』
- 「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』
- 398 ページの『エクスポート用のファイル・タイプ修飾子』
- 403 ページの『データ移動での区切り文字の制限』

## エクスポート用のファイル・タイプ修飾子

表 8. エクスポートで有効なファイル・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。たとえば、ストリング <i>db2exp.001.123.456/</i> がデータ・ファイルに保存される場合、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 データは現行作業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 つまたは複数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。</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。たとえば、NULL LOB の LLS は <i>db2exp.001.7.-1/</i> です。</p>

表 9. エクスポートで有効なファイル・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。<sup>2</sup> 文字ストリング区切り文字として明示的に二重引用符を指定したい場合、次のように指定します。</p> <p style="text-align: center;">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 style="text-align: center;">modified by chardel''</p>



表 9. エクスポートで有効なファイル・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
codepage= <i>x</i>	<p><i>x</i> は <i>ASCII</i> 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。エクスポート操作時に、文字データをアプリケーションのコード・ページからこのコード・ページに変換します。</p> <p>DBCS のみ (GRAPHIC)、混合 DBCS、および EUC の場合、区切り文字の範囲は x00 から x3F に制限されます。</p> <p><b>注:</b> codepage 修飾子を lobsinfile 修飾子と一緒に使うことはできません。</p>
coldel <i>x</i>	<p><i>x</i> は単一文字カラム区切り文字です。デフォルト値はコンマ (,) です。指定した文字は、列の終わりを表すために、コンマの代わりに使用されます。 <sup>2</sup></p> <p>以下の例では、coldel; が指定されており、エクスポート・ユーティリティーは検出するすべてのセミコロン (;) を列の区切り文字として解釈します。</p> <pre>db2 "export to temp of del modified by coldel; select * from staff where dept = 20"</pre>
datesiso	日付形式。すべての日付データ値を ISO フォーマット ("YYYY-MM-DD ") でエクスポートします。 <sup>3</sup>
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。
decpt <i>x</i>	<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。 <sup>2</sup>
dlldel <i>x</i>	<p><i>x</i> は単一文字の DATALINK 区切り文字です。デフォルト値はセミコロン (;) です。指定した文字は、DATALINK 値のフィールド間区切り文字としてセミコロン (,) の代わりに使用されます。DATALINK 値には 2 つ以上の副値を指定できるので、この区切り文字が必要です。 <sup>2</sup></p> <p><b>注:</b> 行、列、または文字ストリング区切り文字と同じ文字を <i>x</i> に指定することはできません。</p>
nochardel	<p>列データは区切り文字で囲まれません。データを DB2 を使用してインポートまたはロードするつもりの場合は、このオプションを指定しないでください。これは、区切り文字を持たないペンダー・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 chardel<i>x</i> または nodoubledel と一緒に指定することはできません。これらは相互に排他的なオプションです。</p>
nodoubledel	二重になっている区切り文字 <sup>2</sup> の認識を抑止します。

## EXPORT

表 9. エクスポートで有効なファイル・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 9. エクスポートで有効なファイル・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。 <sup>4</sup> 有効なタイム・スタンプ・エレメントは以下のとおりです。</p> <p>YYYY - 年 (0000 から 9999 の範囲の 4 桁の数字)  M - 月 (1 から 12 の範囲の 1 桁または 2 桁の数)  MM - 月 (01 から 12 の 2 桁の数。  M と MMM とは相互に排他的)  MMM - 月 (大文字小文字を区別しない月名の 3 文字の省略形。  M と MM とは相互に排他的)  D - 日 (1 から 31 の範囲の 1 桁または 2 桁の数)  DD - 日 (1 から 31 の 2 桁の数字。  D とは相互に排他的)  DDD - 元日から数えた日数 (001 から 366 の 3 桁の数字。  他の日または月のエレメントとは相互に排他的)  H - 時 (12 時間制の場合は 0 から 12、24 時間制では 0 から 24 の範囲の 1 桁または 2 桁の数。)  HH - 時 (12 時間制の場合は 0 から 12、24 時間制では 0 から 24 の範囲の 2 桁の数。  H と相互に排他的)  M - 分 (0 から 59 の 1 桁または 2 桁の数字。)  MM - 分 (0 から 59 の範囲の 2 桁の数字。  M (分) とは相互に排他的)  S - 秒 (0 から 59 の 1 桁または 2 桁の数字。)  SS - 秒 (0 から 59 の範囲の 2 桁の数字。  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 5 桁の数字。  他の時刻エレメントとは相互に排他的)  UUUUUU - マイクロ秒 (000000 から 999999 の範囲の 6 桁の数。  他のマイクロ秒エレメントとは相互に排他的)  UUUUU - マイクロ秒 (00000 から 99999 の範囲の 5 桁の数。  000000 から 999990 の範囲にマップされる。  他のマイクロ秒エレメントとは相互に排他的)  UUUU - マイクロ秒 (0000 から 9999 の範囲の 4 桁の数字。  000000 から 999900 の範囲にマップされる。  他のマイクロ秒エレメントとは相互に排他的)  UUU - マイクロ秒 (000 から 999 の範囲の 3 桁の数。  000000 から 999000 の範囲にマップされる。  他のマイクロ秒エレメントとは相互に排他的)  UU - マイクロ秒 (00 から 99 の範囲の 2 桁の数字。  000000 から 990000 の範囲にマップされる。  他のマイクロ秒エレメントとは相互に排他的)  U - マイクロ秒 (0 から 9 の範囲の 1 桁の数。  000000 から 900000 の範囲にマップされる。  他のマイクロ秒エレメントとは相互に排他的)  TT - 午前/午後の指定子 (AM または PM)</p> <p>タイム・スタンプ・フォーマットの例を以下に示します。</p> <p>"YYYY/MM/DD HH:MM:SS.UUUUUU"</p> <p>MMM エレメントは、以下の値を生成します。「Jan」、「Feb」、「Mar」、  「Apr」、「May」、「Jun」、「Jul」、「Aug」、「Sep」、「Oct」、  「Nov」、および「Dec」。「Jan」は 1 月と等しく、「Dec」は 12 月と等しいです。</p> <p>以下の例は、「schedule」という表から、ユーザー定義のタイム・スタンプ・フォーマットを示すデータをエクスポートする方法を示しています。</p> <pre>db2 export to delfile2 of del modified by timestampformat="yyyy.mm.dd hh:mm tt" select * from schedule</pre>

表 10. エクスポートで有効なファイル・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。 <sup>5</sup> 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。 <sup>5</sup>
3	Lotus 1-2-3 バージョン 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。 <sup>5</sup>
4	DBCS 文字から成る WSF ファイルを作成します。

注:

- サポートされていないファイル・タイプを **MODIFIED BY** オプションで使用しようとしても、エクスポート・ユーティリティーは警告を出しません。サポートされていないファイル・タイプを使おうとすると、エクスポート操作は失敗し、エラー・コードが戻されます。
- データ移動のための区切り文字の制約事項に、区切り文字の指定変更として使用できる文字に適用される制限のリストが示されています。
- 通常、エクスポート・ユーティリティーでの記述フォーマットは次のとおりです。

- 日付データ: *YYYYMMDD* の形式
- 文字 (日付) データ: *YYYY-MM-DD* の形式
- 時刻データ: *HH.MM.SS* の形式
- タイム・スタンプ・データ: *YYYY-MM-DD-HH.MM.SS.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。

以下に示すのは、明確なタイム・スタンプ・フォーマットです。

"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))

- 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。

## 関連資料:

- ・ 「管理 API リファレンス」の『db2Export - エクスポート』
- ・ 393 ページの『EXPORT』
- ・ 403 ページの『データ移動での区切り文字の制限』

## 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 より大きくなければならない (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。たとえば、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

## EXPORT

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。たとえば、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。たとえば、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。



## FORCE APPLICATION

### 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:

**db2stop** は強制終了の間は実行できません。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:

- 517 ページの『LIST APPLICATIONS』
- 294 ページの『ATTACH』



## 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使用可能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。

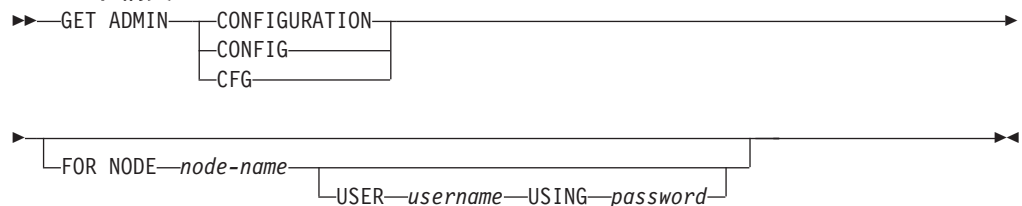
### 権限:

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。

## GET ADMIN CONFIGURATION

### 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:

- 685 ページの『RESET ADMIN CONFIGURATION』
- 773 ページの『UPDATE ADMIN CONFIGURATION』
- 「管理ガイド: 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。

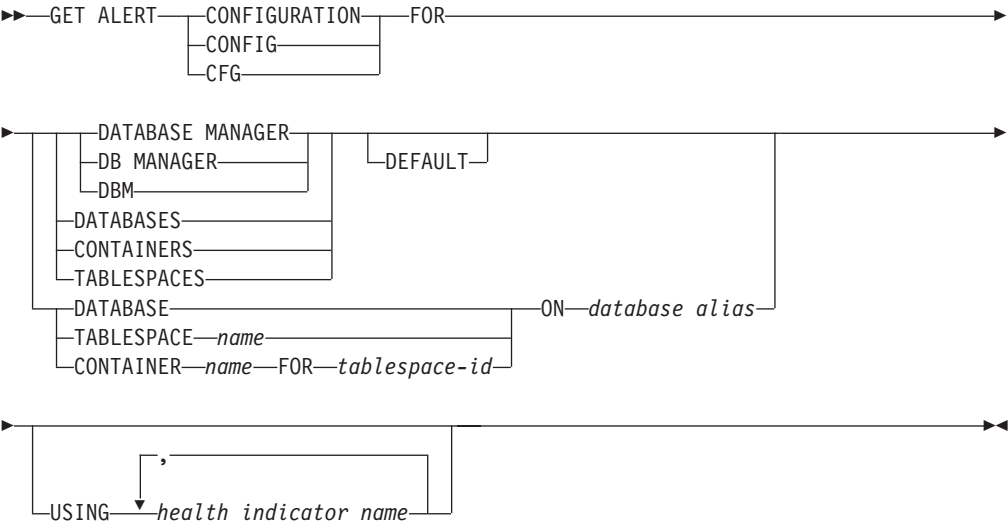
権限:

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、ユーザー定義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。ユーザー定義は、`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、ユーザー定義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。ユーザー定義は、`"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、ユーザー定義を持たないすべての表スペースに適用される設定です。ユーザー定義は、`TABLESPACE name ON database alias` 文節を使って定義されます。

## GET ALERT CONFIGURATION

### 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ユーザー定義を持たない場合、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・コンテナがユーザー定義を持たない場合、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。この表スペースがユーザー定義を持たない場合、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。たとえば、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
```

```

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
|
```



## 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。

**権限:**

なし

**必要な接続:**

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

**コマンド構文:**

►► GET AUTHORIZATIONS ◀◀

**コマンド・パラメーター:**

なし

**例:**

以下に示すのは、GET AUTHORIZATIONS の出力例です。

```
Administrative Authorizations for Current User

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

Indirect SYSADM authority         = YES
Indirect SYSCTRL authority        = NO
Indirect SYSMAINT authority       = NO
Indirect DBADM authority          = NO
Indirect CREATETAB authority      = YES
Indirect BINDADD authority        = YES
Indirect CONNECT authority        = YES
Indirect CREATE_NOT_FENC authority = NO
Indirect IMPLICIT_SCHEMA authority = YES
Indirect LOAD authority           = NO
Indirect QUIESCE_CONNECT authority = NO
Indirect CREATE_EXTERNAL_ROUTINE authority = NO
```

**使用上の注意:**

直接権限は、ユーザー ID に対する権限を与える明示コマンドによって獲得されます。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。

**注:** PUBLIC は、全ユーザーが所属することになる特殊なグループです。

## 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。

### 権限:

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。

**注:** 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、大文字小文字の区別がありません。たとえば、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) が使用可能な場合、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 カタログ化データベースが、データ・ソース名 (DSN) として、クライアント構成アシスタント (CCA) 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:

- 782 ページの『UPDATE CLI CONFIGURATION』
- 660 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 (暗黙接続が利用可能な場合に限る)

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。

**権限:**

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:**

- 732 ページの『SET CLIENT』
- 779 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のアドレスです。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。

### 権限:

なし。

### 必要な接続:

なし。 ローカル実行のみ: このコマンドはリモート接続では使用できません。

### コマンド構文:

► GET CONTACTGROUP *name* ◄

### コマンド・パラメーター:

#### CONTACTGROUP **name**

連絡先を検索するグループの名前。

### 例:

```
GET CONTACTGROUP support
```

Description

-----  
Foo Widgets broadloom support unit

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

# GET CONTACTGROUPS

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。

**権限:**

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 を実行した出力結果です。

Name	Description
-----	-----
support	Foo Widgets broadloom support unit
service	Foo Widgets service and support unit

GET CONTACTS

ローカル・システムで定義された連絡先のリストを取得します。連絡先とは、スケジューラーおよびヘルス・モニターなどのモニター・プロセスが、通知やメッセージを送信する先のユーザーです。

連絡先を作成するには、ADD CONTACT コマンドを使用します。

権限:

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

# 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、それが実行されたパーティションに対してだけ情報を戻します。

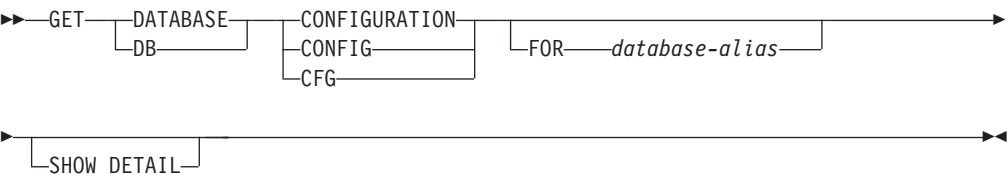
**権限:**

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すでに存在する場合、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
| Database release level                         = 0x0a00
```



## GET DATABASE CONFIGURATION

```

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) =

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) = MAC0

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
Extended storage segments size (4KB)   (ESTORE_SEG_SZ) = 16000
Number of extended storage segments    (NUM_ESTORE_SEGS) = 0
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)                (STMTHEAP) = 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

```

## GET DATABASE CONFIGURATION

```

Sequential detect flag                (SEQDETECT) = YES
Default prefetch size (pages)        (DFT_PREFETCH_SZ) = AUTOMATIC

Track modified pages                  (TRACKMOD) = OFF

Default number of containers          = 1
Default tablespace extentsize (pages) (DFT_EXTENT_SZ) = 32

Max number of active applications     (MAXAPPLS) = AUTOMATIC
Average number of active applications (AVG_APPLS) = 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 active 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 ckckpt (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

```

## GET DATABASE CONFIGURATION

```
Automatic statistics profiling (AUTO_STATS_PROF) = OFF
Automatic profile updates (AUTO_PROF_UPD) = OFF
Automatic reorganization (AUTO_REORG) = OFF
```

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)	=	
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_EXPINT)	= 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
Extended storage segments size (4KB)	(ESTORE_SEG_SZ)	= 16000	16000
Number of extended storage segments	(NUM_ESTORE_SEGS)	= 0	0
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)	(STMTHEAP)	= 2048	2048
Default application heap (4KB)	(APPLHEAPSZ)	= 128	128
Package cache size (4KB)	(PCKCACHESZ)	= (MAXAPPLS*8)	(MAXAPPLS*8)

## GET DATABASE CONFIGURATION

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
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 active log space by transaction	(MAX_LOG) = 0	0
Num. of active log files for 1 active UOW	(NUM_LOG_SPAN) = 0	0
Group commit count	(MINCOMMIT) = 1	1
Percent log file reclaimed before soft ckcpt	(SOFTMAX) = 100	100
Log retain for recovery enabled	(LOGRETAIN) = OFF	OFF
User exit for logging enabled	(USEREXIT) = OFF	OFF
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) =	

## GET DATABASE CONFIGURATION

```
| TSM node name (TSM_NODENAME) =  
| TSM owner (TSM_OWNER) =  
| TSM password (TSM_PASSWORD) =  
|  
| Automatic maintenance (AUTO_MAINT) = OFF OFF  
| Automatic database backup (AUTO_DB_BACKUP) = OFF OFF  
| Automatic table maintenance (AUTO_TBL_MAINT) = OFF OFF  
| Automatic runstats (AUTO_RUNSTATS) = OFF OFF  
| Automatic statistics profiling (AUTO_STATS_PROF) = OFF OFF  
| Automatic profile updates (AUTO_PROF_UPD) = OFF OFF  
| Automatic reorganization (AUTO_REORG) = OFF 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。

### 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』
- 「管理ガイド: 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:

- 689 ページの『RESET DATABASE CONFIGURATION』
- 788 ページの『UPDATE DATABASE CONFIGURATION』
- 「管理ガイド: パフォーマンス」の『構成パラメーターのサマリー』



## GET DATABASE MANAGER CONFIGURATION

```

Default database monitor switches
  Buffer pool                      (DFT_MON_BUFPOOL) = OFF
  Lock                           (DFT_MON_LOCK) = OFF
  Sort                           (DFT_MON_SORT) = OFF
  Statement                       (DFT_MON_STMT) = OFF
  Table                          (DFT_MON_TABLE) = OFF
  Timestamp                      (DFT_MON_TIMESTAMP) = ON
  Unit of work                    (DFT_MON_UOW) = OFF
Monitor health of instance and databases (HEALTH_MON) = ON

SYSADM group name                 (SYSADM_GROUP) =
SYSCTRL group name                (SYSCTRL_GROUP) =
SYSMAINT group name              (SYSMAINT_GROUP) =
SYSMON group name                (SYSMON_GROUP) =

Client Userid-Password Plugin    (CLNT_PW_PLUGIN) =
Client Kerberos Plugin           (CLNT_KRB_PLUGIN) =
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) (RQRIOLBK) = 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

```

## GET DATABASE MANAGER CONFIGURATION

```

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

Maximum query degree of parallelism (MAX_QUERYDEGREE) = ANY
Enable intra-partition parallelism (INTRA_PARALLEL) = NO

No. of int. communication buffers(4KB) (FCM_NUM_BUFFERS) = 512
Number of FCM request blocks      (FCM_NUM_RQB) = AUTOMATIC
Number of FCM connection entries  (FCM_NUM_CONNECT) = AUTOMATIC
Number of FCM message anchors     (FCM_NUM_ANCHORS) = AUTOMATIC

```

以下の出力例は、SHOW DETAIL オプションを指定したときに表示される情報を示しています。 **Delayed Value** 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                                  (DALINKS) = 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

```



## GET DATABASE MANAGER CONFIGURATION

```

Server List of GSS Plugins          (SRVCON_GSSPLUGIN_LIST) =

Server Userid-Password Plugin      (SRVCON_PW_PLUGIN) =

Server Connection Authentication    (SRVCON_AUTH) = NOT        NOT
                                   SPECIFIED    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

Default database path              (DFTDBPATH) = /home      /home
                                   /db2inst    /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 AUTOMATIC
                                   (5386)    (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      MAX
                                   COORDAGENTS 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) = 0        512
Number of FCM request blocks         (FCM_NUM_RQB) = AUTOMATIC AUTOMATIC
                                   (0)    (256)
Number of FCM connection entries      (FCM_NUM_CONNECT) = AUTOMATIC AUTOMATIC

```

## GET DATABASE MANAGER CONFIGURATION

			(-1)	(-1)
	Number of FCM message anchors	(FCM_NUM_ANCHORS) =	AUTOMATIC	AUTOMATIC
			(-1)	(-1)

### 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。

データベース・マネージャー出荷時のデフォルト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。

### 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』
- 「管理ガイド: 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:

- 691 ページの『RESET DATABASE MANAGER CONFIGURATION』
- 791 ページの『UPDATE DATABASE MANAGER CONFIGURATION』
- 「管理ガイド: パフォーマンス」の『構成パラメーターのサマリー』

## GET DATABASE MANAGER MONITOR SWITCHES

データベース・システム・モニター・スイッチの状況を表示します。モニター・スイッチは、データベース活動情報を収集するように、データベース・システム・マネージャーに指示します。データベース・システム・モニター・インターフェースを使用している各アプリケーションには、それ自体のモニター・スイッチの集合があります。モニター中の任意のアプリケーションがオンの場合、データベース・マネージャー・レベル・スイッチがオンになります。モニター中の任意のアプリケーション用に、現在データベース・システム・モニターがデータを収集しているかどうかを判別するために、このコマンドを使用します。

### 権限:

以下のどれかが必要です。

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

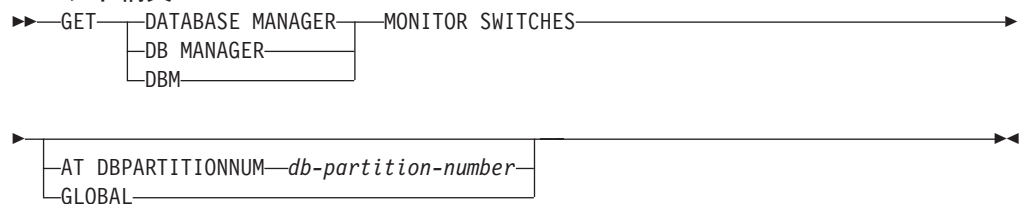
### 必要な接続:

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・アタッチが作成されます。
- 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・アタッチが使用されます。

リモート・インスタンス、または別のローカル・インスタンスの設定値を表示するには、まず最初にそのインスタンスとアタッチすることが必要です。

### コマンド構文:



### コマンド・パラメーター:

#### AT DBPARTITIONNUM db-partition-number

データベース・マネージャーのアプリケーションごと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:

## 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、このスイッチをオフにすることが重要になることに注意してください。このことが生じると、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。

### 関連資料:

- 452 ページの『GET SNAPSHOT』
- 443 ページの『GET MONITOR SWITCHES』
- 693 ページの『RESET MONITOR』
- 798 ページの『UPDATE MONITOR SWITCHES』

## 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。

### 権限:

なし。

### 必要な接続:

| インスタンス。インスタンス・アタッチがない場合、デフォルトのインスタンス・アタッチが作成されます。

| 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。たとえば、次のようになります。

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』

---

## 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、インスタンスまたはそのデータベース・オブジェクトのいずれかに、異常なヘルス状態がみられる場合に通知を受け取るユーザーの(メール・) アドレス、またはページャーのインターネット・アドレスから構成されます。

**権限:**

なし。

**必要な接続:**

インスタンス。明示的なアタッチは必要ありません。

**コマンド構文:**

```
▶▶ GET HEALTH NOTIFICATION CONTACT LIST ◀◀
```

**コマンド・パラメーター:**

なし。

**例:**

GET NOTIFICATION LIST コマンドを実行した、出力結果です。

Name	Type
Joe Brown	Contact
Support	Contact group



## GET HEALTH SNAPSHOT

データベース・マネージャーとそのデータベースのヘルス状況情報を検索します。戻された情報は、コマンドを発行した時点でのヘルス状態のスナップショットを表しています。

### 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、すべてのパーティションから統合された情報が抽出されます。

### 権限:

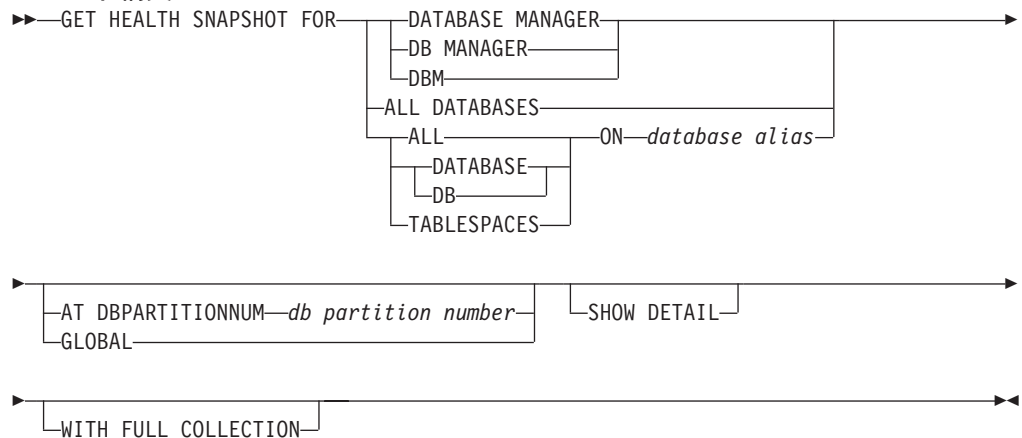
なし。

### 必要な接続:

インスタンス。 インスタンス・アタッチがない場合、デフォルトのインスタンス・アタッチが作成されます。

リモート・インスタンスのスナップショットを獲得するには、まず最初にそのインスタンスにアタッチすることが必要です。

### コマンド構文:



### コマンド・パラメーター:

#### DATABASE MANAGER

アクティブ・データベース・マネージャー・インスタンスの統計を提供します。

#### ALL DATABASES

現行データベース・パーティション上のアクティブ・データベースすべてに関する稼働状態を提供します。

## 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況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)} という形公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。たとえば、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。たとえば、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。

### 例:

以下に示すのは、データベース・マネージャー情報を要求した結果として表示される一般的な出力です。

```
D:>DB2 GET HEALTH SNAPSHOT FOR DBM
```

#### Database Manager Health Snapshot

Node name	=
Node type	= Enterprise Server Edition with local and remote clients
Instance name	= DB2
Snapshot timestamp	= 02/17/2004 12:39:44.818949

```

Number of database partitions in DB2 instance = 1
Start Database Manager timestamp             = 02/17/2004 12:17:21.000119
Instance highest severity alert state        = Normal

```

## Health Indicators:

```

Indicator Name                               = db2.db2_op_status
Value                                         = 0
Evaluation timestamp                         = 02/17/2004 12:37:23.393000
Alert state                                  = Normal

```

```

Indicator Name                               = db2.sort_privmem_util
Value                                         = 0
Unit                                          = %
Evaluation timestamp                         = 02/17/2004 12:37:23.393000
Alert state                                  = Normal

```

```

Indicator Name                               = db2.mon_heap_util
Value                                         = 6
Unit                                          = %
Evaluation timestamp                         = 02/17/2004 12:37:23.393000
Alert state                                  = Normal

```

---

### GET INSTANCE

**DB2INSTANCE** 環境変数の値を取得します。

**権限:**

なし

**必要な接続:**

なし

**コマンド構文:**

▶▶ GET INSTANCE ◀◀

**コマンド・パラメーター:**

なし

**例:**

以下に示すのは、GET INSTANCE の出力のサンプルです。

```
The current database manager instance is: smith
```

# GET MONITOR SWITCHES

現行セッションのデータベース・システム・モニター・スイッチの状況を表示します。モニター・スイッチは、データベース活動情報を収集するように、データベース・システム・マネージャーに指示します。データベース・システム・モニター・インターフェースを使用している各アプリケーションには、それ自体のモニター・スイッチの集合があります。このコマンドはそれらを表示します。データベース・マネージャー・レベル・スイッチを表示するには、 `GET DBM MONITOR SWITCHES` コマンドを使用します。

**権限:**

以下のどれかが必要です。

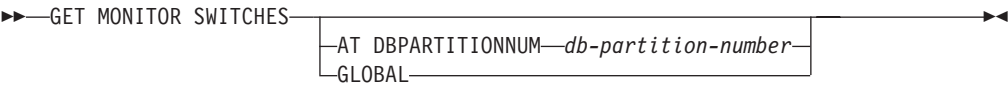
- `sysadm`
- `sysctrl`
- `sysmaint`
- `sysmon`

**必要な接続:**

インスタンス。インスタンス・アタッチがない場合、デフォルトのインスタンス・アタッチが作成されます。

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**

データベースごと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、このスイッチをオフにすることが重要になることに注意してください。このことが生じると、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。

#### 関連資料:

- 452 ページの『GET SNAPSHOT』
- 433 ページの『GET DATABASE MANAGER MONITOR SWITCHES』
- 693 ページの『RESET MONITOR』
- 798 ページの『UPDATE MONITOR SWITCHES』





**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パーティション番号を指定します。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、現在接続しているパーティションの情報を戻します。

**GLOBAL**

すべてのパーティションで、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、関係するすべての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 (this can be done using the EXEC\_DB2\_CMD stored procedure):

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

Launch DB2 tool: Database Configuration Window

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

To open the Database Configuration window:

1. From the Control Center, expand the object tree until you find the databases folder.
2. Click the databases folder. Any existing database are displayed

## GET RECOMMENDATIONS

- 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 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、そのオブジェクトのアラートに関する情報になります。

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・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』



されたプロシージャ名が SQL プロシージャを識別しない場合、 "-204, 42704" という診断テキストが戻されます。 "-204" は SQLCODE、"42704" は SQLSTATE で、それぞれ問題の原因を示します。この例の SQLCODE および SQLSTATE は、GET ROUTINE コマンドに指定されたプロシージャ名が未定義であることを示しています。

---

## GET SNAPSHOT

状況情報を収集して、ユーザー用に出力を形式設定します。戻された情報は、コマンドを発行した時点でのデータベース・マネージャー操作状況のスナップショットを表しています。

### 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、呼び出されたパーティション上でのみ活動します。

### 権限:

以下のどれかが必要です。

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

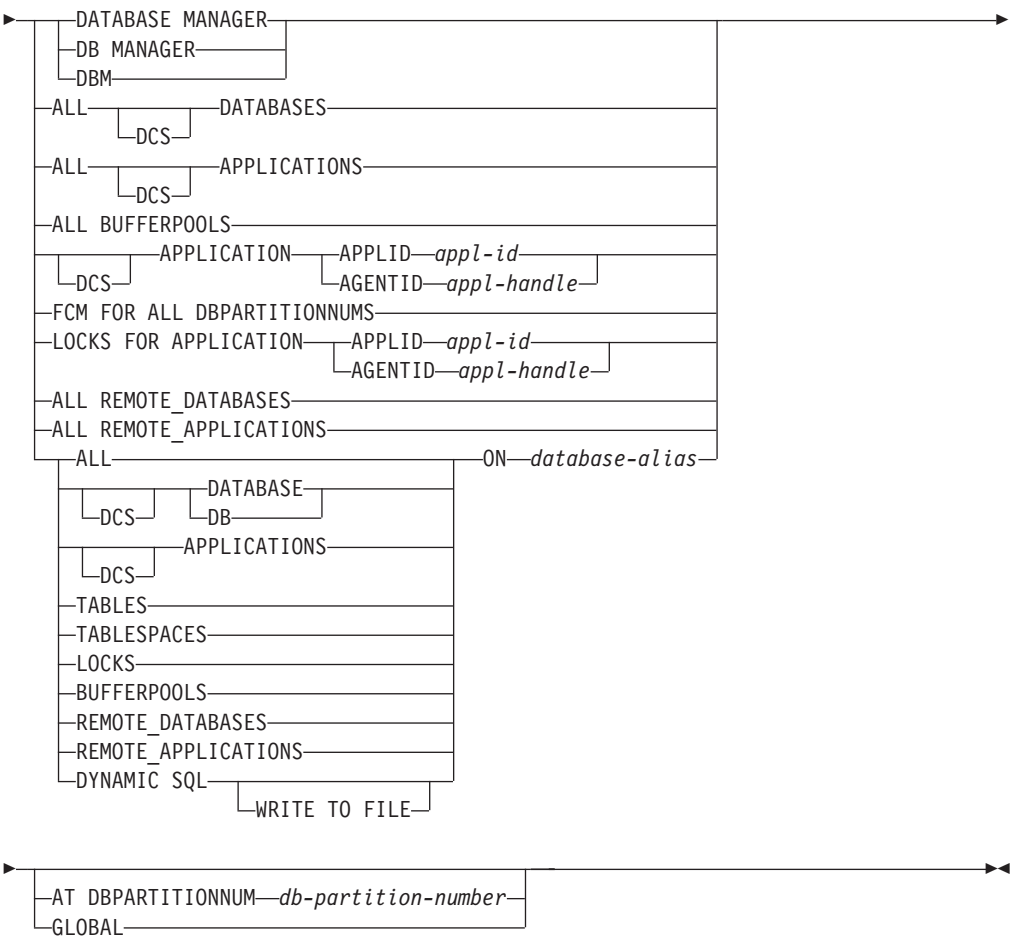
### 必要な接続:

インスタンス。インスタンス・アタッチがない場合、デフォルトのインスタンス・アタッチが作成されます。

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 \_\_\_\_\_▶



注:

- 1. 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・アクティビティーに関する情報を提供します。

**APPLICATION APPLID appl-id**

指定された ID を持つアプリケーションの情報だけを提供します。特定のアプリケーション ID を獲得するには、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:**

以下の出力例のリストでは、適切なデータベース・システム・モニターの記録スイッチがオンになっていないために、情報の一部が利用不可になっている場合があります。情報が利用できない場合、Not Collected が出力に表示されます。

以下に示すのは、データベース・マネージャー情報の一般的な出力結果です。

## Database Manager Snapshot

```

Node type = Database Server with local clients
Instance name = minweiw2
Number of database partitions in DB2 instance = 1
Database manager status = Active

Product name = DB2 v8.1.0.64
Service level = n040215 (U488485)

Private Sort heap allocated = 0
Private Sort heap high water mark = 0

```

## GET SNAPSHOT

```
Post threshold sorts                = 0
Piped sorts requested                = 0
Piped sorts accepted                 = 0

Start Database Manager timestamp     = 02/17/2004 12:17:21.493836
Last reset timestamp                  =
Snapshot timestamp                    = 02/17/2004 12:19:12.210537

Remote connections to db manager     = 0
Remote connections executing in db manager = 0
Local connections                    = 2
Local connections executing in db manager = 1
Active local databases                = 1

High water mark for agents registered = 3
High water mark for agents waiting for a token = 0
Agents registered                     = 3
Agents waiting for a token            = 0
Idle agents                           = 0

Committed private Memory (Bytes)     = 835584

Switch list for db partition number 0
Buffer Pool Activity Information (BUFFERPOOL) = ON 02/17/2004 12:17:21.493836
Lock Information (LOCK) = ON 02/17/2004 12:17:21.493836
Sorting Information (SORT) = ON 02/17/2004 12:17:21.493836
SQL Statement Information (STATEMENT) = ON 02/17/2004 12:17:21.493836
Table Activity Information (TABLE) = ON 02/17/2004 12:17:21.493836
Take Timestamp Information (TIMESTAMP) = ON 02/17/2004 12:17:21.493836
Unit of Work Information (UOW) = ON 02/17/2004 12:17:21.493836

Agents assigned from pool             = 1
Agents created from empty pool        = 4
Agents stolen from another application = 0
High water mark for coordinating agents = 3
Max agents overflow                   = 0
Hash joins after heap threshold exceeded = 0

Total number of gateway connections   = 0
Current number of gateway connections = 0
Gateway connections waiting for host reply = 0
Gateway connections waiting for client request = 0
Gateway connection pool agents stolen = 0

Memory usage for database manager:

Memory Pool Type                      = Database Monitor Heap
Current size (bytes)                  = 180224
High water mark (bytes)               = 180224
Configured size (bytes)               = 376832

Memory Pool Type                      = Other Memory
Current size (bytes)                  = 4751360
High water mark (bytes)               = 4751360
Configured size (bytes)               = 18071552
```

以下に示すのは、データベース情報の一般的な出力結果です。

```
Database Snapshot

Database name                        = SAMPLE
Database path                        = /home/minweiw2/minweiw2
                                      /NODE0000/SQL00001/
Input database alias                 = SAMPLE
Database status                      = Active
Catalog database partition number    = 0
```

```

Catalog network node name           =
Operating system running at database server= AIX
Location of the database             = Local
First database connect timestamp     = 02/17/2004 12:17:25.076527
Last reset timestamp                 =
Last backup timestamp                =
Snapshot timestamp                   = 02/17/2004 12:19:11.548218

High water mark for connections      = 2
Application connects                 = 2
Secondary connects total             = 0
Applications connected currently     = 2
Appls. executing in db manager currently = 1
Agents associated with applications   = 2
Maximum agents associated with applications= 2
Maximum coordinating agents         = 2

Locks held currently                 = 7
Lock waits                           = 1
Time database waited on locks (ms)  = 26039
Lock list memory in use (Bytes)      = 2304
Deadlocks detected                   = 0
Lock escalations                     = 0
Exclusive lock escalations           = 0
Agents currently waiting on locks    = 1
Lock Timeouts                        = 0
Number of indoubt transactions       = 0

Total Private Sort heap allocated    = 0
Total Shared Sort heap allocated     = 0
Shared Sort heap high water mark     = 0
Total sorts                          = 0
Total sort time (ms)                 = 0
Sort overflows                       = 0
Active sorts                         = 0

Buffer pool data logical reads       = 98
Buffer pool data physical reads      = 27
  Buffer pool temporary data logical reads = 0
  Buffer pool temporary data physical reads = 0
    Asynchronous pool data page reads = 0
    Buffer pool data writes            = 2
    Asynchronous pool data page writes = 0
  Buffer pool index logical reads     = 214
  Buffer pool index physical reads    = 91
  Buffer pool temporary index logical reads = 0
  Buffer pool temporary index physical reads = 0
    Asynchronous pool index page reads = 0
    Buffer pool index writes           = 0
    Asynchronous pool index page writes = 0
Total buffer pool read time (ms)     = 947
  Total buffer pool write time (ms)   = 3
  Total elapsed asynchronous read time = 0
  Total elapsed asynchronous write time = 0
  Asynchronous data read requests    = 0
  Asynchronous index read requests   = 0
  No victim buffers available        = 0
LSN Gap cleaner triggers             = 0
Dirty page steal cleaner triggers    = 0
Dirty page threshold cleaner triggers = 0
Time waited for prefetch (ms)       = 0
Unread prefetch pages               = 0
  Direct reads                       = 42
  Direct writes                       = 4
  Direct read requests                = 7
  Direct write requests               = 2
  Direct reads elapsed time (ms)     = 0

```

## GET SNAPSHOT

```

|         Direct write elapsed time (ms)           = 1
| Database files closed                           = 0
|         Data pages copied to extended storage    = 0
|         Index pages copied to extended storage    = 0
|         Data pages copied from extended storage  = 0
|         Index pages copied from extended storage = 0
|
| Host execution elapsed time                     = 0.069848
|
| Commit statements attempted                     = 2
| Rollback statements attempted                   = 0
| Dynamic statements attempted                    = 8
| Static statements attempted                     = 2
| Failed statement operations                     = 0
| Select SQL statements executed                   = 1
| Update/Insert/Delete statements executed        = 1
| DDL statements executed                         = 2
|
| Internal automatic rebinds                      = 0
| Internal rows deleted                           = 0
| Internal rows inserted                          = 0
| Internal rows updated                           = 0
| Internal commits                                = 2
| Internal rollbacks                              = 0
| Internal rollbacks due to deadlock              = 0
|
| Rows deleted                                   = 0
| Rows inserted                                  = 1
| Rows updated                                   = 0
| Rows selected                                  = 0
| Rows read                                      = 31
| Binds/precompiles attempted                   = 0
|
| Log space available to the database (Bytes)= 20395444
| Log space used by the database (Bytes)         = 4556
| Maximum secondary log space used (Bytes)       = 0
| Maximum total log space used (Bytes)           = 6031
| Secondary logs allocated currently              = 0
| Log pages read                                 = 0
| Log read time (sec.ns)                         = 0.000000004
| Log pages written                              = 6
| Log write time (sec.ns)                        = 0.000000004
| Number write log IOs                           = 6
| Number read log IOs                            = 0
| Number partial page log IOs                    = 4
| Number log buffer full                         = 0
| Log data found in buffer                       = 0
| Appl id holding the oldest transaction          = 7
| Log to be redone for recovery (Bytes)           = 4464
| Log accounted for by dirty pages (Bytes)        = 4424
|
| File number of first active log                 = 0
| File number of last active log                  = 2
| File number of current active log               = 0
| File number of log being archived               = Not applicable
|
| Package cache lookups                          = 10
| Package cache inserts                          = 8
| Package cache overflows                        = 0
| Package cache high water mark (Bytes)          = 207369
| Application section lookups                     = 8
| Application section inserts                     = 5
|
| Catalog cache lookups                          = 20
| Catalog cache inserts                          = 6
| Catalog cache overflows                        = 0
| Catalog cache high water mark                  = 0

```

## Workspace Information

Shared high water mark	= 0
Corresponding shared overflows	= 0
Total shared section inserts	= 0
Total shared section lookups	= 0
Private high water mark	= 17692
Corresponding private overflows	= 0
Total private section inserts	= 5
Total private section lookups	= 5
Number of hash joins	= 0
Number of hash loops	= 0
Number of hash join overflows	= 0
Number of small hash join overflows	= 0

## Memory usage for database:

Memory Pool Type	= Backup/Restore/Util Heap
Current size (bytes)	= 16384
High water mark (bytes)	= 16384
Configured size (bytes)	= 20496384
Memory Pool Type	= Package Cache Heap
Current size (bytes)	= 262144
High water mark (bytes)	= 262144
Configured size (bytes)	= 4294950912
Memory Pool Type	= Catalog Cache Heap
Current size (bytes)	= 65536
High water mark (bytes)	= 65536
Configured size (bytes)	= 4294950912
Memory Pool Type	= Buffer Pool Heap
Current size (bytes)	= 4259840
High water mark (bytes)	= 4259840
Configured size (bytes)	= 4294950912
Memory Pool Type	= Buffer Pool Heap
Current size (bytes)	= 540672
High water mark (bytes)	= 540672
Configured size (bytes)	= 4294950912
Memory Pool Type	= Buffer Pool Heap
Current size (bytes)	= 278528
High water mark (bytes)	= 278528
Configured size (bytes)	= 4294950912
Memory Pool Type	= Buffer Pool Heap
Current size (bytes)	= 147456
High water mark (bytes)	= 147456
Configured size (bytes)	= 4294950912
Memory Pool Type	= Buffer Pool Heap
Current size (bytes)	= 81920
High water mark (bytes)	= 81920
Configured size (bytes)	= 4294950912
Memory Pool Type	= Lock Manager Heap
Current size (bytes)	= 507904
High water mark (bytes)	= 507904
Configured size (bytes)	= 507904
Memory Pool Type	= Database Heap
Current size (bytes)	= 3637248

## GET SNAPSHOT

```
|
|           High water mark (bytes)           = 3637248
|           Configured size (bytes)          = 8339456
|
|           Memory Pool Type                  = Other Memory
|           Current size (bytes)              = 0
|           High water mark (bytes)           = 0
|           Configured size (bytes)           = 12353536
|
```

以下に示すのは、(アプリケーション ID、アプリケーション・ハンドル、全アプリケーション、または特定のデータベース上の全アプリケーションのいずれかを指定して) アプリケーション情報の一般的な出力結果です。

```
|
|           Application Snapshot
|
|           Application handle                 = 9
|           Application status                 = Lock-wait
|           Status change time                = 02/17/2004 12:18:45.508734
|           Application code page              = 850
|           Application country/region code    = 1
|           DUOW correlation token             = *LOCAL.minweiw2.0F7397171829
|           Application name                   = db2bp
|           Application ID                     = *LOCAL.minweiw2.0F7397171829
|           Sequence number                   = 0001
|           TP Monitor client user ID          =
|           TP Monitor client workstation name =
|           TP Monitor client application name =
|           TP Monitor client accounting string =
|
|           Connection request start timestamp = 02/17/2004 12:18:29.718212
|           Connect request completion timestamp = 02/17/2004 12:18:29.735915
|           Application idle time              = 0
|           CONNECT Authorization ID           = MINWEIW2
|           Client login ID                    = minweiw2
|           Configuration NNAME of client      =
|           Client database manager product ID = SQL08020
|           Process ID of client application   = 194360
|           Platform of client application     = AIX
|           Communication protocol of client   = Local Client
|
|           Inbound communication address      = *LOCAL.minweiw2
|
|           Database name                      = SAMPLE
|           Database path                      = /home/minweiw2/minweiw2/
|                                           NODE0000/SQL00001/
|           Client database alias              = SAMPLE
|           Input database alias               =
|           Last reset timestamp               =
|           Snapshot timestamp                 = 02/17/2004 12:19:13.260464
|           The highest authority level granted =
|               Direct DBADM authority
|               Direct CREATETAB authority
|               Direct BINDADD authority
|               Direct CONNECT authority
|               Direct CREATE_NOT_FENC authority
|               Direct LOAD authority
|               Direct IMPLICIT_SCHEMA authority
|               Direct CREATE_EXT_RT authority
|               Direct QUIESCE_CONN authority
|               Indirect SYSADM authority
|               Indirect CREATETAB authority
|               Indirect BINDADD authority
|               Indirect CONNECT authority
|               Indirect IMPLICIT_SCHEMA authority
|           Coordinating database partition number = 0
|           Current database partition number   = 0
|           Coordinator agent process or thread ID = 35384
|
```

```

Agents stolen = 0
Agents waiting on locks = 1
Maximum associated agents = 1
Priority at which application agents work = 0
Priority type = Dynamic

Lock timeout (seconds) = -1
Locks held by application = 4
Lock waits since connect = 1
Time application waited on locks (ms) = 27751
Deadlocks detected = 0
Lock escalations = 0
Exclusive lock escalations = 0
Number of Lock Timeouts since connected = 0
Total time UOW waited on locks (ms) = 27751

Total sorts = 0
Total sort time (ms) = 0
Total sort overflows = 0

Data pages copied to extended storage = 0
Index pages copied to extended storage = 0
Data pages copied from extended storage = 0
Index pages copied from extended storage = 0
Buffer pool data logical reads = 0
Buffer pool data physical reads = 0
Buffer pool temporary data logical reads = 0
Buffer pool temporary data physical reads = 0
Buffer pool data writes = 0
Buffer pool index logical reads = 0
Buffer pool index physical reads = 0
Buffer pool temporary index logical reads = 0
Buffer pool temporary index physical reads = 0
Buffer pool index writes = 0
Total buffer pool read time (ms) = 0
Total buffer pool write time (ms) = 0
Time waited for prefetch (ms) = 0
Unread prefetch pages = 0
Direct reads = 0
Direct writes = 0
Direct read requests = 0
Direct write requests = 0
Direct reads elapsed time (ms) = 0
Direct write elapsed time (ms) = 0

Number of SQL requests since last commit = 3
Commit statements = 0
Rollback statements = 0
Dynamic SQL statements attempted = 3
Static SQL statements attempted = 0
Failed statement operations = 0
Select SQL statements executed = 1
Update/Insert/Delete statements executed = 0
DDL statements executed = 0
Internal automatic rebinds = 0
Internal rows deleted = 0
Internal rows inserted = 0
Internal rows updated = 0
Internal commits = 1
Internal rollbacks = 0
Internal rollbacks due to deadlock = 0
Binds/precompiles attempted = 0
Rows deleted = 0
Rows inserted = 0
Rows updated = 0
Rows selected = 0
Rows read = 0

```

## GET SNAPSHOT

```

|
|      Rows written                                = 0
|
|      UOW log space used (Bytes)                  = 0
|      Previous UOW completion timestamp           = 02/17/2004 12:18:29.735915
|      Elapsed time of last completed uow (sec.ms)= 0.000000
|      UOW start timestamp                         = 02/17/2004 12:18:45.394125
|      UOW stop timestamp                         =
|      UOW completion status                      =
|
|      Open remote cursors                        = 0
|      Open remote cursors with blocking          = 0
|      Rejected Block Remote Cursor requests     = 0
|      Accepted Block Remote Cursor requests     = 1
|      Open local cursors                        = 1
|      Open local cursors with blocking          = 1
|      Total User CPU Time used by agent (s)      = 0.020000
|      Total System CPU Time used by agent (s)    = 0.100000
|      Host execution elapsed time                = 0.001853
|
|      Package cache lookups                      = 2
|      Package cache inserts                     = 1
|      Application section lookups               = 3
|      Application section inserts              = 1
|      Catalog cache lookups                    = 6
|      Catalog cache inserts                    = 0
|      Catalog cache overflows                  = 0
|      Catalog cache high water mark            = 0
|
|      Workspace Information
|
|      Shared high water mark                    = 0
|      Total shared overflows                   = 0
|      Total shared section inserts             = 0
|      Total shared section lookups            = 0
|      Private high water mark                  = 14976
|      Total private overflows                  = 0
|      Total private section inserts            = 1
|      Total private section lookups            = 1
|
|      Most recent operation                     = Fetch
|      Cursor name                              = SQLCUR201
|      Most recent operation start timestamp     = 02/17/2004 12:18:45.504828
|      Most recent operation stop timestamp      =
|      Agents associated with the application    = 1
|      Number of hash joins                     = 0
|      Number of hash loops                     = 0
|      Number of hash join overflows            = 0
|      Number of small hash join overflows      = 0
|      Statement type                           = Dynamic SQL Statement
|      Statement                                = Fetch
|      Section number                           = 201
|      Application creator                      = NULLID
|      Package name                             = SQLC2E03
|      Consistency Token                       = AAAAAJHR
|      Package Version ID                      =
|      Cursor name                             = SQLCUR201
|      Statement database partition number       = 0
|      Statement start timestamp                = 02/17/2004 12:18:45.504828
|      Statement stop timestamp                 =
|      Elapsed time of last completed stmt(sec.ms)= 0.001853
|      Total Statement user CPU time            = 0.000000
|      Total Statement system CPU time          = 0.000000
|      SQL compiler cost estimate in timerons   = 27
|      SQL compiler cardinality estimate        = 180
|      Degree of parallelism requested          = 1
|      Number of agents working on statement    = 1
|      Number of subagents created for statement = 1
|

```



```

Statement sorts                      = 0
Total sort time                      = 0
Sort overflows                      = 0
Rows read                          = 0
Rows written                        = 0
Rows deleted                       = 0
Rows updated                       = 0
Rows inserted                       = 0
Rows fetched                        = 0
Buffer pool data logical reads      = 0
Buffer pool data physical reads     = 0
Buffer pool temporary data logical reads = 0
Buffer pool temporary data physical reads = 0
Buffer pool index logical reads     = 0
Buffer pool index physical reads    = 0
Buffer pool temporary index logical reads = 0
Buffer pool temporary index physical reads = 0
Blocking cursor                     = YES
Dynamic SQL statement text:
select * from t1

Agent process/thread ID              = 35384

Agent process/thread ID              = 35384
Agent Lock timeout (seconds)         = -1
Memory usage for agent:

Memory Pool Type                    = Application Heap
Current size (bytes)                 = 147456
High water mark (bytes)              = 147456
Configured size (bytes)              = 1277952

Memory Pool Type                    = Application Control Heap
Current size (bytes)                 = 16384
High water mark (bytes)              = 16384
Configured size (bytes)              = 704512

ID of agent holding lock             = 7
Application ID holding lock          = *LOCAL.minweiw2.0307B7171724
Lock name                           = 0x0002000D00000000000000000054
Lock attributes                      = 0x00000000
Release flags                        = 0x00000001
Lock object type                     = Table
Lock mode                           = Exclusive Lock (X)
Lock mode requested                  = Intention Share Lock (IS)
Name of tablespace holding lock      = USERSPACE1
Schema of table holding lock         = MINWEIW2
Name of table holding lock           = T1
Lock wait start timestamp            = 02/17/2004 12:18:45.508738

Application Snapshot

Application handle                    = 7
Application status                    = UOW Waiting
Status change time                   = 02/17/2004 12:18:24.237397
Application code page                = 850
Application country/region code      = 1
DUOW correlation token               = *LOCAL.minweiw2.0307B7171724
Application name                     = db2bp
Application ID                       = *LOCAL.minweiw2.0307B7171724
Sequence number                     = 0003
TP Monitor client user ID            =
TP Monitor client workstation name   =
TP Monitor client application name   =
TP Monitor client accounting string  =

Connection request start timestamp   = 02/17/2004 12:17:25.076527

```

## GET SNAPSHOT

```

Connect request completion timestamp      = 02/17/2004 12:17:27.198920
Application idle time                     = 49
CONNECT Authorization ID                  = MINWEIW2
Client login ID                           = minweiw2
Configuration NNAME of client              =
Client database manager product ID        = SQL08020
Process ID of client application           = 209018
Platform of client application             = AIX
Communication protocol of client           = Local Client

Inbound communication address              = *LOCAL.minweiw2

Database name                             = SAMPLE
Database path                             = /home/minweiw2/minweiw2/
                                           NODE0000/SQL00001/
Client database alias                     = SAMPLE
Input database alias                      =
Last reset timestamp                      =
Snapshot timestamp                        = 02/17/2004 12:19:13.260464
The highest authority level granted        =
    Direct DBADM authority
    Direct CREATETAB authority
    Direct BINDADD authority
    Direct CONNECT authority
    Direct CREATE_NOT_FENC authority
    Direct LOAD authority
    Direct IMPLICIT_SCHEMA authority
    Direct CREATE_EXT_RT authority
    Direct QUIESCE_CONN authority
    Indirect SYSADM authority
    Indirect CREATETAB authority
    Indirect BINDADD authority
    Indirect CONNECT authority
    Indirect IMPLICIT_SCHEMA authority
Coordinating database partition number    = 0
Current database partition number          = 0
Coordinator agent process or thread ID    = 167996
Agents stolen                             = 0
Agents waiting on locks                   = 0
Maximum associated agents                 = 1
Priority at which application agents work  = 0
Priority type                             = Dynamic

Lock timeout (seconds)                    = -1
Locks held by application                  = 3
Lock waits since connect                   = 0
Time application waited on locks (ms)     = 0
Deadlocks detected                        = 0
Lock escalations                          = 0
Exclusive lock escalations                 = 0
Number of Lock Timeouts since connected   = 0
Total time UOW waited on locks (ms)       = 0

Total sorts                              = 0
Total sort time (ms)                      = 0
Total sort overflows                      = 0

Data pages copied to extended storage      = 0
Index pages copied to extended storage     = 0
Data pages copied from extended storage    = 0
Index pages copied from extended storage   = 0
Buffer pool data logical reads             = 98
Buffer pool data physical reads            = 27
Buffer pool temporary data logical reads   = 0
Buffer pool temporary data physical reads = 0
Buffer pool data writes                    = 2
Buffer pool index logical reads            = 214

```

```

|           Buffer pool index physical reads          = 91
|           Buffer pool temporary index logical reads = 0
|           Buffer pool temporary index physical reads = 0
|           Buffer pool index writes                  = 0
|           Total buffer pool read time (ms)          = 947
|           Total buffer pool write time (ms)         = 3
|           Time waited for prefetch (ms)            = 0
|           Unread prefetch pages                     = 0
|             Direct reads                            = 42
|             Direct writes                           = 4
|             Direct read requests                    = 7
|             Direct write requests                   = 2
|             Direct reads elapsed time (ms)          = 0
|             Direct write elapsed time (ms)          = 1
|
|           Number of SQL requests since last commit = 2
|           Commit statements                         = 2
|           Rollback statements                       = 0
|           Dynamic SQL statements attempted          = 5
|           Static SQL statements attempted           = 2
|           Failed statement operations                = 0
|           Select SQL statements executed             = 0
|           Update/Insert/Delete statements executed  = 1
|           DDL statements executed                   = 2
|           Internal automatic rebinds                = 0
|           Internal rows deleted                     = 0
|           Internal rows inserted                    = 0
|           Internal rows updated                     = 0
|           Internal commits                           = 1
|           Internal rollbacks                        = 0
|           Internal rollbacks due to deadlock        = 0
|           Binds/precompiles attempted              = 0
|             Rows deleted                            = 0
|             Rows inserted                           = 1
|             Rows updated                            = 0
|             Rows selected                           = 0
|             Rows read                               = 31
|             Rows written                            = 9
|
|           UOW log space used (Bytes)                = 159
|           Previous UOW completion timestamp         = 02/17/2004 12:18:13.052905
|           Elapsed time of last completed uow (sec.ms) = 0.137336
|           UOW start timestamp                      = 02/17/2004 12:18:18.844035
|           UOW stop timestamp                       =
|           UOW completion status                     =
|
|           Open remote cursors                      = 0
|           Open remote cursors with blocking         = 0
|           Rejected Block Remote Cursor requests    = 0
|           Accepted Block Remote Cursor requests     = 0
|           Open local cursors                       = 0
|           Open local cursors with blocking          = 0
|           Total User CPU Time used by agent (s)     = 0.300000
|           Total System CPU Time used by agent (s)   = 0.150000
|           Host execution elapsed time               = 0.067995
|
|           Package cache lookups                     = 8
|           Package cache inserts                     = 7
|           Application section lookups                = 5
|           Application section inserts                = 4
|           Catalog cache lookups                     = 14
|           Catalog cache inserts                     = 6
|           Catalog cache overflows                   = 0
|           Catalog cache high water mark              = 0
|
|           Workspace Information

```

## GET SNAPSHOT

```

Shared high water mark           = 0
Total shared overflows           = 0
Total shared section inserts     = 0
Total shared section lookups     = 0
Private high water mark          = 17692
Total private overflows          = 0
Total private section inserts    = 4
Total private section lookups    = 4

Most recent operation            = Execute Immediate
Most recent operation start timestamp = 02/17/2004 12:18:24.169317
Most recent operation stop timestamp = 02/17/2004 12:18:24.237312
Agents associated with the application = 1
Number of hash joins             = 0
Number of hash loops             = 0
Number of hash join overflows    = 0
Number of small hash join overflows = 0
Statement type                   = Dynamic SQL Statement
Statement                        = Execute Immediate
Section number                   = 203
Application creator              = NULLID
Package name                     = SQLC2E03
Consistency Token                = AAAAJHR
Package Version ID               =
Cursor name                      =
Statement database partition number = 0
Statement start timestamp        = 02/17/2004 12:18:24.169317
Statement stop timestamp         = 02/17/2004 12:18:24.237312
Elapsed time of last completed stmt(sec.ms) = 0.067995
Total Statement user CPU time    = 0.010000
Total Statement system CPU time  = 0.060000
SQL compiler cost estimate in timerons = 13
SQL compiler cardinality estimate = 1
Degree of parallelism requested  = 1
Number of agents working on statement = 0
Number of subagents created for statement = 1
    Statement sorts              = 0
Total sort time                  = 0
Sort overflows                   = 0
Rows read                        = 0
    Rows written                 = 1
    Rows deleted                 = 0
    Rows updated                 = 0
    Rows inserted                = 0
Rows fetched                     = 0
    Buffer pool data logical reads = 1
    Buffer pool data physical reads = 0
    Buffer pool temporary data logical reads = 0
    Buffer pool temporary data physical reads = 0
    Buffer pool index logical reads = 0
    Buffer pool index physical reads = 0
    Buffer pool temporary index logical reads = 0
    Buffer pool temporary index physical reads = 0
Blocking cursor                  = NO
Dynamic SQL statement text:
insert into t1 values(1)

Agent process/thread ID          = 167996
Agent Lock timeout (seconds)     = -1
Memory usage for agent:

    Memory Pool Type             = Application Heap
    Current size (bytes)         = 212992
    High water mark (bytes)      = 212992
    Configured size (bytes)      = 1277952

```

```

Memory Pool Type           = Application Control Heap
Current size (bytes)       = 16384
High water mark (bytes)   = 16384
Configured size (bytes)   = 704512

```

以下に示すのは、バッファ・プール情報の一般的な出力結果です。

#### Bufferpool Snapshot

```

Bufferpool name           = IBMDEFAULTBP
Database name             = SAMPLE
Database path             = /home/minweiw2/minweiw2/
                           NODE0000/SQL00001/
Input database alias      = SAMPLE
Snapshot timestamp        = 02/17/2004 12:19:14.265625

Buffer pool data logical reads      = 98
Buffer pool data physical reads     = 27
Buffer pool temporary data logical reads = 0
Buffer pool temporary data physical reads = 0
Buffer pool data writes             = 2
Buffer pool index logical reads     = 214
Buffer pool index physical reads    = 91
Buffer pool temporary index logical reads = 0
Buffer pool temporary index physical reads = 0
Total buffer pool read time (ms)    = 947
Total buffer pool write time (ms)   = 3
Asynchronous pool data page reads  = 0
Asynchronous pool data page writes = 0
Buffer pool index writes            = 0
Asynchronous pool index page reads  = 0
Asynchronous pool index page writes = 0
Total elapsed asynchronous read time = 0
Total elapsed asynchronous write time = 0
Asynchronous data read requests     = 0
Asynchronous index read requests    = 0
No victim buffers available         = 0
Direct reads                        = 42
Direct writes                       = 4
Direct read requests                = 7
Direct write requests               = 2
Direct reads elapsed time (ms)      = 0
Direct write elapsed time (ms)      = 1
Database files closed               = 0
Data pages copied to extended storage = 0
Index pages copied to extended storage = 0
Data pages copied from extended storage = 0
Index pages copied from extended storage = 0
Unread prefetch pages              = 0
Vectored IOs                       = 0
Pages from vectored IOs             = 0
Block IOs                           = 0
Pages from block IOs                = 0
Physical page maps                  = 0

Node number                       = 0
Tablespaces using bufferpool       = 3
Alter bufferpool information:
Pages left to remove               = 0
Current size                       = 1000
Post-alter size                    = 1000

```

以下に示すのは、表情報の一般的な出力結果です。

#### Table Snapshot

```

First database connect timestamp    = 02/17/2004 12:17:25.076527

```

## GET SNAPSHOT

```
Last reset timestamp      =
Snapshot timestamp       = 02/17/2004 12:19:10.785689
Database name            = SAMPLE
Database path            = /home/minweiw2/minweiw2/
                          NODE0000/SQL00001/
Input database alias     = SAMPLE
Number of accessed tables = 14
```

### Table List

```
Table Schema      = SYSIBM
Table Name        = SYSTABLES
Table Type        = Catalog
Data Object Pages = 27
Index Object Pages = 17
LOB Object pages  = 256
Rows Read        = 11
Rows Written     = 2
Overflows        = 0
Page Reorgs      = 0
```

```
Table Schema      = SYSIBM
Table Name        = SYSCOLUMNS
Table Type        = Catalog
Data Object Pages = 144
Index Object Pages = 71
Rows Read        = 2
Rows Written     = 2
Overflows        = 0
Page Reorgs      = 0
```

```
Table Schema      = SYSIBM
Table Name        = SYSPLAN
Table Type        = Catalog
Data Object Pages = 9
Index Object Pages = 5
LOB Object pages  = 320
Rows Read        = 1
Rows Written     = 0
Overflows        = 0
Page Reorgs      = 0
```

```
Table Schema      = SYSIBM
Table Name        = SYSDBAUTH
Table Type        = Catalog
Data Object Pages = 1
Index Object Pages = 3
Rows Read        = 3
Rows Written     = 0
Overflows        = 0
Page Reorgs      = 0
```

```
Table Schema      = SYSIBM
Table Name        = SYSTABAUTH
Table Type        = Catalog
Data Object Pages = 5
Index Object Pages = 13
Rows Read        = 1
Rows Written     = 2
Overflows        = 0
Page Reorgs      = 0
```

```
Table Schema      = SYSIBM
Table Name        = SYSEVENTMONITORS
Table Type        = Catalog
Data Object Pages = 1
Index Object Pages = 3
LOB Object pages  = 64
```

```

Rows Read          = 1
Rows Written       = 0
Overflows          = 0
Page Reorgs       = 0

Table Schema       = SYSIBM
Table Name         = SYSTABLESPACES
Table Type         = Catalog
Data Object Pages  = 1
Index Object Pages = 7
Rows Read          = 5
Rows Written       = 0
Overflows          = 0
Page Reorgs       = 0

Table Schema       = SYSIBM
Table Name         = SYSSCHEMATA
Table Type         = Catalog
Data Object Pages  = 1
Index Object Pages = 3
Rows Read          = 1
Rows Written       = 0
Overflows          = 0
Page Reorgs       = 0

Table Schema       = SYSIBM
Table Name         = SYSUSERAUTH
Table Type         = Catalog
Data Object Pages  = 8
Index Object Pages = 7
LOB Object pages   = 64
Rows Read          = 1
Rows Written       = 2
Overflows          = 0
Page Reorgs       = 0

Table Schema       = SYSIBM
Table Name         = SYSNODEGROUPS
Table Type         = Catalog
Data Object Pages  = 1
Index Object Pages = 3
Rows Read          = 1
Rows Written       = 0
Overflows          = 0
Page Reorgs       = 0

Table Schema       = SYSIBM
Table Name         = SYSBUFFERPOOLS
Table Type         = Catalog
Data Object Pages  = 1
Index Object Pages = 4
Rows Read          = 1
Rows Written       = 0
Overflows          = 0
Page Reorgs       = 0

Table Schema       = SYSIBM
Table Name         = SYSTBSPACEAUTH
Table Type         = Catalog
Data Object Pages  = 1
Index Object Pages = 4
Rows Read          = 1
Rows Written       = 0
Overflows          = 0
Page Reorgs       = 0

Table Schema       = SYSIBM

```

## GET SNAPSHOT

```
Table Name          = SYSVERSIONS
Table Type          = Catalog
Data Object Pages   = 1
Index Object Pages  = 3
Rows Read           = 1
Rows Written        = 0
Overflows           = 0
Page Reorgs         = 0

Table Schema        = MINWEIW2
Table Name          = T1
Table Type          = User
Data Object Pages   = 1
Rows Read           = 0
Rows Written        = 1
Overflows           = 0
Page Reorgs         = 0
```

以下に示すのは、表スペース情報の一般的な出力結果です。

### Tablespace Snapshot

```
First database connect timestamp = 02/17/2004 12:17:25.076527
Last reset timestamp             =
Snapshot timestamp               = 02/17/2004 12:19:10.105473
Database name                    = SAMPLE
Database path                    = /home/minweiw2/minweiw2/
                                NODE0000/SQL00001/
Input database alias             = SAMPLE
Number of accessed tablespaces  = 3

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
Automatic Prefetch size enabled  = Yes
Buffer pool ID currently in use  = 1
Buffer pool ID next startup      = 1
File system caching              = No
Tablespace State                 = 0x'00000000'
Detailed explanation:
Normal
Tablespace Prefetch size (pages) = 32
Total number of pages            = 4475
Number of usable pages           = 4475
Number of used pages             = 4475
Minimum Recovery Time            =
Number of quiescers              = 0
Number of containers             = 1

Container Name                   = /home/minweiw2/minweiw2/
                                NODE0000/SQL00001/SQLT0000.0
Container ID                     = 0
Container Type                   = Path
Total Pages in Container         = 4475
Usable Pages in Container        = 4475
Stripe Set                      = 0
Container is accessible          = Yes

Buffer pool data logical reads   = 93
Buffer pool data physical reads  = 26
Buffer pool temporary data logical reads = 0
Buffer pool temporary data physical reads = 0
Asynchronous pool data page reads = 0
```



```

Buffer pool data writes = 0
Asynchronous pool data page writes = 0
Buffer pool index logical reads = 214
Buffer pool index physical reads = 91
Buffer pool temporary index logical reads = 0
Buffer pool temporary index physical reads = 0
Asynchronous pool index page reads = 0
Buffer pool index writes = 0
Asynchronous pool index page writes = 0
Total buffer pool read time (ms) = 946
Total buffer pool write time (ms) = 0
Total elapsed asynchronous read time = 0
Total elapsed asynchronous write time = 0
Asynchronous data read requests = 0
Asynchronous index read requests = 0
No victim buffers available = 0
Direct reads = 42
Direct writes = 4
Direct read requests = 7
Direct write requests = 2
Direct reads elapsed time (ms) = 0
Direct write elapsed time (ms) = 1
Number of files closed = 0
Data pages copied to extended storage = 0
Index pages copied to extended storage = 0
Data pages copied from extended storage = 0
Index pages copied from extended storage = 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
Automatic Prefetch size enabled = Yes
Buffer pool ID currently in use = 1
Buffer pool ID next startup = 1
File system caching = No
Tablespace State = 0x'00000000'
Detailed explanation:
Normal
Tablespace Prefetch size (pages) = 32
Total number of pages = 1
Number of usable pages = 1
Number of used pages = 1
Minimum Recovery Time =
Number of quiescers = 0
Number of containers = 1

Container Name = /home/minweiw2/minweiw2/
NODE0000/SQL00001/SQLT0001.0
Container ID = 0
Container Type = Path
Total Pages in Container = 1
Usable Pages in Container = 1
Stripe Set = 0
Container is accessible = Yes

Buffer pool data logical reads = 0
Buffer pool data physical reads = 0
Buffer pool temporary data logical reads = 0
Buffer pool temporary data physical reads = 0
Asynchronous pool data page reads = 0
Buffer pool data writes = 0
Asynchronous pool data page writes = 0
Buffer pool index logical reads = 0

```

## GET SNAPSHOT

```

Buffer pool index physical reads      = 0
Buffer pool temporary index logical reads = 0
Buffer pool temporary index physical reads = 0
Asynchronous pool index page reads    = 0
Buffer pool index writes               = 0
Asynchronous pool index page writes   = 0
Total buffer pool read time (ms)      = 0
Total buffer pool write time (ms)     = 0
Total elapsed asynchronous read time  = 0
Total elapsed asynchronous write time = 0
Asynchronous data read requests       = 0
Asynchronous index read requests      = 0
No victim buffers available           = 0
Direct reads                          = 0
Direct writes                         = 0
Direct read requests                  = 0
Direct write requests                 = 0
Direct reads elapsed time (ms)        = 0
Direct write elapsed time (ms)        = 0
Number of files closed                 = 0
Data pages copied to extended storage = 0
Index pages copied to extended storage = 0
Data pages copied from extended storage = 0
Index pages copied from extended storage = 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
Automatic Prefetch size enabled       = Yes
Buffer pool ID currently in use       = 1
Buffer pool ID next startup            = 1
File system caching                   = No
Tablespace State                      = 0x'00000000'
Detailed explanation:
Normal
Tablespace Prefetch size (pages)      = 32
Total number of pages                  = 408
Number of usable pages                 = 408
Number of used pages                   = 408
Minimum Recovery Time                  =
Number of quiescers                    = 0
Number of containers                   = 1

Container Name                         = /home/minweiw2/minweiw2/
                                       NODE0000/SQL00001/SQLT0002.0
Container ID                           = 0
Container Type                         = Path
Total Pages in Container               = 408
Usable Pages in Container              = 408
Stripe Set                            = 0
Container is accessible                 = Yes

Buffer pool data logical reads         = 5
Buffer pool data physical reads        = 1
Buffer pool temporary data logical reads = 0
Buffer pool temporary data physical reads = 0
Asynchronous pool data page reads     = 0
Buffer pool data writes                = 2
Asynchronous pool data page writes    = 0
Buffer pool index logical reads        = 0
Buffer pool index physical reads       = 0
Buffer pool temporary index logical reads = 0
Buffer pool temporary index physical reads = 0

```

```

Asynchronous pool index page reads      = 0
Buffer pool index writes                 = 0
Asynchronous pool index page writes      = 0
Total buffer pool read time (ms)         = 1
Total buffer pool write time (ms)        = 3
Total elapsed asynchronous read time     = 0
Total elapsed asynchronous write time    = 0
Asynchronous data read requests          = 0
Asynchronous index read requests         = 0
No victim buffers available              = 0
Direct reads                            = 0
Direct writes                           = 0
Direct read requests                     = 0
Direct write requests                    = 0
Direct reads elapsed time (ms)           = 0
Direct write elapsed time (ms)           = 0
Number of files closed                   = 0
Data pages copied to extended storage    = 0
Index pages copied to extended storage   = 0
Data pages copied from extended storage  = 0
Index pages copied from extended storage = 0

```

以下の例は、ロック情報の一般的な出力結果です。分かりやすくするために、省略符号 (...) が除去された内部ロック情報に置き換わっています。1つの内部ロックについての情報だけが残っています。

#### Database Lock Snapshot

```

Database name      = SAMPLE
Database path      = /home/minweiw2/minweiw2/
                   NODE0000/SQL00001/
Input database alias = SAMPLE
Locks held         = 6
Applications currently connected = 2
Agents currently waiting on locks = 1
Snapshot timestamp = 02/17/2004 12:19:09.013588

Application handle = 9
Application ID      = *LOCAL.minweiw2.0F7397171829
  Sequence number   = 0001
  Application name   = db2bp
CONNECT Authorization ID = MINWEIW2
Application status  = Lock-wait
Status change time  = 02/17/2004 12:18:45.508734
Application code page = 850
Locks held          = 3
Total wait time (ms) = 23504

  ID of agent holding lock = 7
  Application ID holding lock = *LOCAL.minweiw2.0307B7171724
  Lock name                 = 0x0002000D0000000000000000000054
  Lock attributes           = 0x00000000
  Release flags             = 0x00000001
  Lock object type          = Table
  Lock mode                 = Exclusive Lock (X)
  Lock mode requested       = Intention Share Lock (IS)
  Name of tablespace holding lock = USERSPACE1
  Schema of table holding lock  = MINWEIW2
  Name of table holding lock   = T1
  Lock wait start timestamp   = 02/17/2004 12:18:45.508738

List Of Locks
Lock Name      = 0x000000001000000010001680056
Lock Attributes = 0x00000000
Release Flags  = 0x40000000

```

## GET SNAPSHOT

```

|
| Lock Count = 1
| Hold Count = 0
| Lock Object Name = 0
| Object Type = Internal Variation Lock
| Mode = S
|
| Lock Name = 0x53514C4332453033A95B579A41
| Lock Attributes = 0x00000000
| Release Flags = 0x40000000
| Lock Count = 1
| Hold Count = 0
| Lock Object Name = 0
| Object Type = Internal Plan Lock
| Mode = S
|
| Lock Name = 0x53514C4445464C540763DD2841
| Lock Attributes = 0x00000000
| Release Flags = 0x40000000
| Lock Count = 1
| Hold Count = 0
| Lock Object Name = 0
| Object Type = Internal Plan Lock
| Mode = S
|
|
| Application handle = 7
| Application ID = *LOCAL.minweiw2.0307B7171724
| Sequence number = 0003
| Application name = db2bp
| CONNECT Authorization ID = MINWEIW2
| Application status = UOW Waiting
| Status change time = 02/17/2004 12:18:24.237397
| Application code page = 850
| Locks held = 3
| Total wait time (ms) = 0
|
| List Of Locks
| Lock Name = 0x0000000200001A09409060F043
| Lock Attributes = 0x00000000
| Release Flags = 0x40000000
| Lock Count = 4
| Hold Count = 0
| Lock Object Name = 0
| Object Type = Internal Catalog Cache Lock
| Mode = S
|
| Lock Name = 0x53514C4332453033A95B579A41
| Lock Attributes = 0x00000000
| Release Flags = 0x40000000
| Lock Count = 1
| Hold Count = 0
| Lock Object Name = 0
| Object Type = Internal Plan Lock
| Mode = S
|
| Lock Name = 0x0002000D00000000000000000054
| Lock Attributes = 0x00000000
| Release Flags = 0x40000000
| Lock Count = 255
| Hold Count = 0
| Lock Object Name = 13
| Object Type = Table
| Tablespace Name = USERSPACE1
| Table Schema = MINWEIW2
| Table Name = T1
| Mode = X
|

```

以下のサンプルの一部に示されているように、LOCK スイッチがオンの場合には追加のアプリケーション情報が表示されます。

```

...
Application handle                = 2
Application ID                    = *LOCAL.mikew.07B492160951
Sequence number                  = 0001
Application name                  = db2bp
Authorization ID                  = MIKEW
Application status                = Lock-wait
Status change time               = Not Collected
Application code page            = 819
Locks held                       = 9
Total wait time (ms)             = 0

Subsection waiting for lock      = 0
ID of agent holding lock         = 3
Application ID holding lock      = *LOCAL.mikew.016A92161122
Lock name                       = 0x0002000200000000000000000054
Lock attributes                  = 0x00
Release flags                    = 0x40000001
Lock object type                 = Table
Lock mode                       = Intention Exclusive Lock (IX)
Lock mode held                   = Intention Exclusive Lock (IX)
Lock mode requested              = Exclusive Lock (X)
Name of tablespace holding lock  = USERSPACE1
Schema of table holding lock     = MIKEW
Name of table holding lock       = SNAPSHOT
Lock wait start timestamp       = Not Collected
Lock is a result of escalation  = NO
...

```

以下に示すのは、動的 SQL 情報の一般的な出力結果です。

#### Dynamic SQL Snapshot Result

```

Database name                    = SAMPLE

Database path                    = /home/minweiw2/minweiw2/
                                NODE0000/SQL00001/

Number of executions             = 1
Number of compilations           = 2
Worst preparation time (ms)      = 12
Best preparation time (ms)       = 12
Internal rows deleted            = 0
Internal rows inserted           = 0
Rows read                       = 16
Internal rows updated            = 0
Rows written                     = 4
Statement sorts                  = 0
Statement sort overflows         = 0
Total sort time                  = 0
Buffer pool data logical reads   = 46
Buffer pool data physical reads  = 17
Buffer pool temporary data logical reads = 0
Buffer pool temporary data physical reads = 0
Buffer pool index logical reads  = 124
Buffer pool index physical reads = 58
Buffer pool temporary index logical reads = 0
Buffer pool temporary index physical reads = 0
Total execution time (sec.ms)    = 0.894210
Total user cpu time (sec.ms)     = 0.120000
Total system cpu time (sec.ms)   = 0.050000
Statement text                   = drop table t1

```

## GET SNAPSHOT

```

Number of executions          = 0
Number of compilations        = 0
Worst preparation time (ms)   = 0
Best preparation time (ms)    = 0
Internal rows deleted         = 0
Internal rows inserted        = 0
Rows read                    = 0
Internal rows updated         = 0
Rows written                  = 0
Statement sorts               = 0
  Buffer pool data logical reads = 0
  Buffer pool data physical reads = 0
Buffer pool temporary data logical reads = 0
Buffer pool temporary data physical reads = 0
  Buffer pool index logical reads = 0
  Buffer pool index physical reads = 0
Buffer pool temporary index logical reads = 0
Buffer pool temporary index physical reads = 0
Total execution time (sec.ms) = 0.000000
Total user cpu time (sec.ms) = 0.000000
Total system cpu time (sec.ms) = 0.000000
Statement text                = SET CURRENT LOCALE LC_CTYPE = 'en_US'

```

```

Number of executions          = 1
Number of compilations        = 1
Worst preparation time (ms)   = 73
Best preparation time (ms)    = 73
Internal rows deleted         = 0
Internal rows inserted        = 0
Rows read                    = 0
Internal rows updated         = 0
Rows written                  = 0
Statement sorts               = 0
  Buffer pool data logical reads = 0
  Buffer pool data physical reads = 0
Buffer pool temporary data logical reads = 0
Buffer pool temporary data physical reads = 0
  Buffer pool index logical reads = 0
  Buffer pool index physical reads = 0
Buffer pool temporary index logical reads = 0
Buffer pool temporary index physical reads = 0
Total execution time (sec.ms) = 0.000000
Total user cpu time (sec.ms) = 0.000000
Total system cpu time (sec.ms) = 0.000000
Statement text                = select * from t1

```

```

Number of executions          = 1
Number of compilations        = 2
Worst preparation time (ms)   = 6
Best preparation time (ms)    = 6
Internal rows deleted         = 0
Internal rows inserted        = 0
Rows read                    = 1
Internal rows updated         = 0
Rows written                  = 0
Statement sorts               = 0
Statement sort overflows      = 0
Total sort time               = 0
Buffer pool data logical reads = 1
  Buffer pool data physical reads = 0
Buffer pool temporary data logical reads = 0
Buffer pool temporary data physical reads = 0
  Buffer pool index logical reads = 2
  Buffer pool index physical reads = 0
Buffer pool temporary index logical reads = 0

```

```

Buffer pool temporary index physical reads = 0
Total execution time (sec.ms)             = 0.011801
Total user cpu time (sec.ms)              = 0.010000
Total system cpu time (sec.ms)            = 0.000000
Statement text                             = lock table t1 in exclusive mode

```

```

Number of executions                       = 1
Number of compilations                    = 2
Worst preparation time (ms)               = 3
Best preparation time (ms)               = 3
Internal rows deleted                     = 0
Internal rows inserted                    = 0
Rows read                                = 4
Internal rows updated                     = 0
Rows written                             = 4
Statement sorts                          = 0
Statement sort overflows                  = 0
Total sort time                           = 0
Buffer pool data logical reads            = 26
Buffer pool data physical reads           = 3
Buffer pool temporary data logical reads = 0
Buffer pool temporary data physical reads = 0
Buffer pool index logical reads           = 44
Buffer pool index physical reads          = 10
Buffer pool temporary index logical reads = 0
Buffer pool temporary index physical reads = 0
Total execution time (sec.ms)             = 0.129477
Total user cpu time (sec.ms)              = 0.090000
Total system cpu time (sec.ms)            = 0.000000
Statement text                             = create table t1 (c1 int)

```

```

Number of executions                       = 1
Number of compilations                    = 1
Worst preparation time (ms)               = 64
Best preparation time (ms)               = 64
Internal rows deleted                     = 0
Internal rows inserted                    = 0
Rows read                                = 0
Internal rows updated                     = 0
Rows written                             = 1
Statement sorts                          = 0
Statement sort overflows                  = 0
Total sort time                           = 0
Buffer pool data logical reads            = 1
Buffer pool data physical reads           = 0
Buffer pool temporary data logical reads = 0
Buffer pool temporary data physical reads = 0
Buffer pool index logical reads           = 0
Buffer pool index physical reads          = 0
Buffer pool temporary index logical reads = 0
Buffer pool temporary index physical reads = 0
Total execution time (sec.ms)             = 0.067995
Total user cpu time (sec.ms)              = 0.010000
Total system cpu time (sec.ms)            = 0.060000
Statement text                             = insert into t1 values(1)

```

以下に示すのは、(DCS アプリケーション ID、DCS アプリケーション・ハンドル、全 DCS アプリケーション、または特定のデータベース上の全 DCS アプリケーションのいずれかを指定して) DCS アプリケーション情報の一般的な出力結果です。

#### DCS Application Snapshot

```

Client application ID                     = *LOCAL.andrewkm.010613200844
Sequence number                          = 0001

```

## GET SNAPSHOT

Authorization ID	= AMURCHIS
Application name	= db2bp
Application handle	= 5
Application status	= waiting for request
Status change time	= 12-31-1969 19:00:00.000000
Client node	=
Client release level	= SQL07021
Client platform	= AIX
Client protocol	= Local Client
Client codepage	= 850
Process ID of client application	= 36034
Client login ID	= andrewkm
Host application ID	= G9158067.CDF2.010613200845
Sequence number	= 0000
Database alias at the gateway	= GSAMPLE
DCS database name	= SAMPLE
Host database name	= SAMPLE
Host release level	= SQL07021
Host CCSID	= 850
Outbound communication address	= 9.21.115.179 17336
Outbound communication protocol	= TCP/IP
Inbound communication address	= *LOCAL.andrewkm
First database connect timestamp	= 06-13-2001 16:08:44.142656
Host response time (sec.ms)	= 0.271230
Time spent on gateway processing	= 0.000119
Last reset timestamp	=
Rows selected	= 0
Number of SQL statements attempted	= 1
Failed statement operations	= 0
Commit statements	= 1
Rollback statements	= 0
Inbound bytes received	= 184
Outbound bytes sent	= 10
Outbound bytes received	= 32
Inbound bytes sent	= 0
Number of open cursors	= 0
Application idle time	= 1 minute and 33 seconds
UOW completion status	= Committed - Commit Statement
Previous UOW completion timestamp	=
UOW start timestamp	= 06-13-2001 16:08:44.716911
UOW stop timestamp	= 06-13-2001 16:08:44.852730
Elapsed time of last completed uow (sec.ms)	= 0.135819
Most recent operation	= Static Commit
Most recent operation start timestamp	= 06-13-2001 16:08:44.716911
Most recent operation stop timestamp	= 06-13-2001 16:08:44.852730
Host execution elapsed time	= 0.000000
Statement	= Static Commit
Section number	= 0
Application creator	= NULLID
Package name	= SQLC2D02
SQL compiler cost estimate in timerons	= 0
SQL compiler cardinality estimate	= 0
Statement start timestamp	= 06-13-2001 16:08:44.716911
Statement stop timestamp	= 06-13-2001 16:08:44.852730
Host response time (sec.ms)	= 0.271230
Elapsed time of last completed stmt(sec.ms)	= 0.135819
Rows fetched	= 0
Time spent on gateway processing	= 0.000119
Inbound bytes received for statement	= 184
Outbound bytes sent for statement	= 10
Outbound bytes received for statement	= 32
Inbound bytes sent for statement	= 0



```

Blocking cursor                = NO
Outbound blocking cursor       = NO
Host execution elapsed time    = 0.000000

```

以下に示すのは、DCS データベース情報の一般的な出力結果です。

#### DCS Database Snapshot

```

DCS database name              = SAMPLE

Host database name             = SAMPLE
First database connect timestamp = 06-13-2001 16:08:44.142656
Most recent elapsed time to connect = 0.569354
Most recent elapsed connection duration = 0.000000
Host response time (sec.ms)     = 0.271230
Last reset timestamp           =
Number of SQL statements attempted = 1
Commit statements attempted      = 1
Rollback statements attempted    = 0
Failed statement operations      = 0
Total number of gateway connections = 1
Current number of gateway connections = 1
Gateway conn. waiting for host reply = 0
Gateway conn. waiting for client request = 1
Gateway communication errors to host = 0
Timestamp of last communication error = None
High water mark for gateway connections = 1
Rows selected                   = 0
Outbound bytes sent              = 10
Outbound bytes received          = 32
Host execution elapsed time      = 0.000000

```

#### 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) からスナップショットを取得するには、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常駐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・メッセージが返されます。

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くてはなら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。

#### 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。

## GET SNAPSHOT

### 関連資料:

- 443 ページの『GET MONITOR SWITCHES』
- 517 ページの『LIST APPLICATIONS』
- 693 ページの『RESET MONITOR』
- 798 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。

### 権限:

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、¥sql1lib¥doc¥html サブディレクトリーになければなりません。

コマンド行プロセッサは、コマンドが成功したかどうかを知ることができないため、エラー状態を報告できません。

## HISTORY

CLP 対話モード・セッション内で実行したコマンドの履歴を表示します。

### 有効範囲

このコマンドは、CLP 対話モードでのみ実行できます。さらに具体的に言えば、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。

### 権限:

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来ます。このパラメーターを指定しない場合は、コマンドのリストを発生順で表示します。つまり、実行した最新のコマンドがリストの末尾に来ます。

*num* 最新の *num* 個のコマンドだけを表示します。このパラメーターを指定しない場合は、最大 20 個のコマンドが表示されます。ただし、表示されるコマンドの数は、コマンド履歴に保管されているコマンドの数によっても制限されます。

### 使用上の注意:

1. コマンド履歴に保管できるコマンドの最大数は、DB2\_CLP\_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:

- 391 ページの『EDIT』
- 719 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、またはビューに挿入します。 **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。

### 権限:

- **INSERT** オプションを使用して **IMPORT** コマンドを実行する場合、以下のどれかが必要です。
  - *sysadm*
  - *dbadm*
  - 関係する表またはビューのそれぞれに対する **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** する場合、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
  - 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** する場合、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** する場合、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。

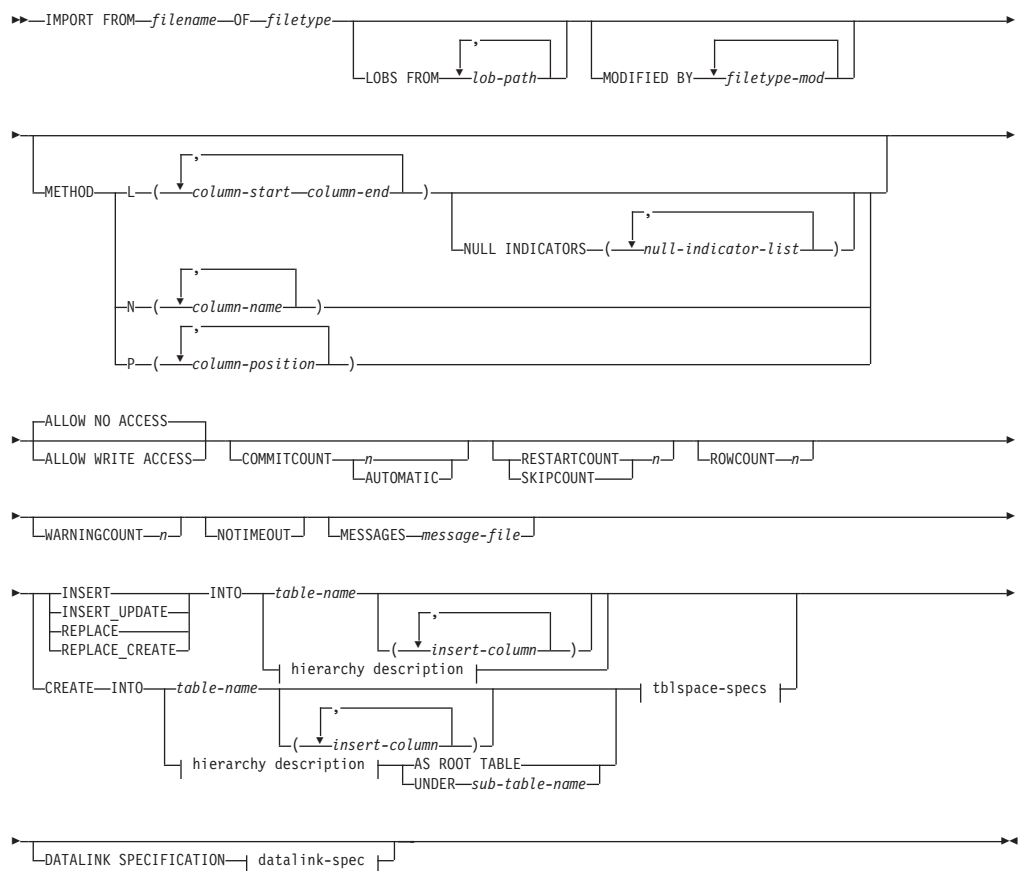
## IMPORT

- 表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
  - 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、以下のいずれかが必要です。
    - *sysadm*
    - *dbadm*
    - 階層内のすべての副表に対する **CONTROL** 特権

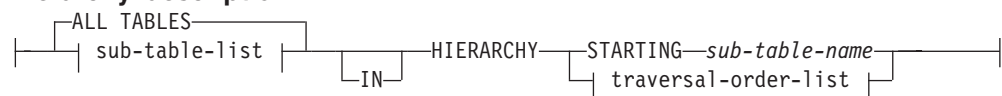
### 必要な接続:

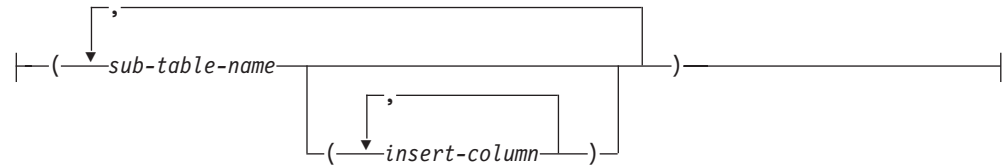
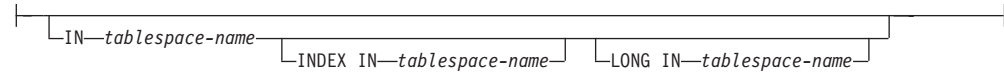
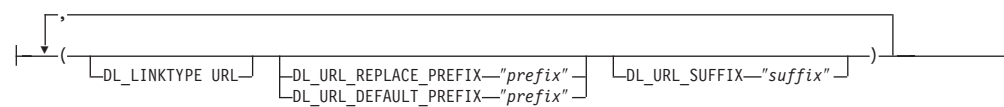
データベース。暗黙接続が可能な場合には、デフォルト・データベースへの接続が確立されます。

### コマンド構文:



### hierarchy description:



**sub-table-list:****traversal-order-list:****tblspace-specs:****datalink-spec:****コマンド・パラメーター:****ALL TABLES**

暗黙のキーワード (階層のみ)。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な排他 (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。

**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

## IMPORT

ど COMMIT が実行されます。コンパウンド挿入を使用した場合、ユーザー指定のコミット頻度  $n$  は、そのコンパウンド・カウント値に最も近い整数の倍数に切り上げられます。AUTOMATIC を指定すると、コミットの必要時期はインポート操作で内部的に判別されます。次の 2 つのうちのいずれかの理由で、このユーティリティーはコミットを行います。

- アクティブ・ログ・スペースを使いきらないようにするため。
- ロックが行レベルから表レベルにエスカレーションしないようにするため。

ALLOW WRITE ACCESS オプションを指定した場合に COMMITCOUNT オプションを指定しないと、インポート・ユーティリティーは、COMMITCOUNT AUTOMATIC が指定されたものとしてコミットを実行します。

## 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。

**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。

## DATALINK SPECIFICATION

各 DATALINK 列ごとに、それぞれ 1 つの列指定を括弧で囲んで指定できます。各列指定は、1 つ以上の DL\_LINKTYPE、接頭部、および DL\_URL\_SUFFIX 指定で構成されます。接頭部指定は、DL\_URL\_REPLACE\_PREFIX または DL\_URL\_DEFAULT\_PREFIX のどちらかになります。

DATALINK 列指定は、表で定義されている DATALINK 列の数と同じ数だけ指定できます。指定の順序は、挿入列 リストの中での DATALINK 列の順序、または表定義内での順序 (挿入列 リストが指定されていない場合) に従います。

## DL\_LINKTYPE

指定した場合は、列定義の LINKTYPE に一致していなければなりません。そうすることによって、列定義に LINKTYPE URL が指定されている場合に DL\_LINKTYPE URL が受け入れ可能になります。

## DL\_URL\_DEFAULT\_PREFIX "prefix"

これを指定すると、同じ列内のすべての DATALINK 値のデフォルト接頭部になります。ここでいう接頭部とは、URL 指定の「スキーム・ホスト・ポート」部分のことです。

接頭部の例



```
"http://server"
"file://server"
"file:"
"http://server:80"
```

列のデータの中に接頭部がない場合、DL\_URL\_DEFAULT\_PREFIX でデフォルトの接頭部が指定されているなら、列の値の接頭部としてそのデフォルト接頭部が付けられます (NULL でない場合)。

たとえば、DL\_URL\_DEFAULT\_PREFIX でデフォルト接頭部が "http://toronto" として指定されている場合、

- 列入力値 "/x/y/z" は "http://toronto/x/y/z" として保管されます。
- 列入力値 "http://coyote/a/b/c" は "http://coyote/a/b/c" として保管されます。
- 列入力値 NULL は NULL として保管されます。

#### **DL\_URL\_REPLACE\_PREFIX "prefix"**

この文節は、それ以前にエクスポート・ユーティリティによって生成されたデータをロードまたはインポートする際に、ユーザーが、データに入っているホスト名を別のホスト名に一括置換したい場合に便利です。指定する場合には、それがすべての 非 NULL 列値の接頭部になります。列値にすでに接頭部がある場合、それは置き換えられます。列値に接頭部がない場合、DL\_URL\_REPLACE\_PREFIX で指定される接頭部がその列値の接頭部になります。

たとえば、DL\_URL\_REPLACE\_PREFIX で接頭部が "http://toronto" として指定されている場合、

- 列入力値 "/x/y/z" は "http://toronto/x/y/z" として保管されます。
- 列入力値 "http://coyote/a/b/c" は "http://toronto/a/b/c" として保管されます。"coyote" は "toronto" に置き換えられます。
- 列入力値 NULL は NULL として保管されます。

#### **DL\_URL\_SUFFIX "suffix"**

これを指定すると、それはその列のすべての非 NULL 列値に付加されます。これは実際には、DATALINK 値の URL 部分の「パス」コンポーネントに付加されます。

#### **FROM filename**

インポートするデータのあったファイルを指定します。パスを省略すると、現行の作業ディレクトリが使用されます。

#### **HIERARCHY**

インポートする階層データを指定します。

#### **IN tablespace-name**

表を作成する表スペースを指定します。表スペースは存在している必要があります、REGULAR 表スペースでなければなりません。他の表スペースを指定しない場合、すべての表パーツはこの表スペースに保管されます。この文節を指定しない場合、表は許可 IDENTITIY によって作成された表スペース中に作成されます。何も検出されない場合、その表はデフォルト表スペースの USERSPACE1 に入れられます。USERSPACE1 がドロップされていた場合、表作成は失敗します。

### INDEX IN tablespace-name

表の索引の作成先の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、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、インポートし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。

下位のサーバーの場合を除き、INSERT、INSERT\_UPDATE、および 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。 *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。lobsinfile 修飾子が指定されていない場合、このオプションは無視されます。

### LONG IN tablespace-name

ロング列の値 (LONG VARCHAR、LONG VARGRAPHIC、LOB データ・タイプ、またはソース・タイプとしてこれらが指定されている特殊タイプ) を保管する表スペースを指定します。このオプションは、IN 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。

### 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。宛先ファイルがすで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。『インポート用のファイル・タイプ修飾子』を参照してください。

#### NOTIMEOUT

インポート・ユーティリティーは、ロック待ちの間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 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・マネージャーで使用します。
- WSF (ワークシート・フォーマット)。以下のプログラムで使用します。

- Lotus 1-2-3
- Lotus Symphony
- IXF (統合交換フォーマット、PC バージョン)。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。

## 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。DATALINK 列を備えた表では無効です。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。

## REPLACE\_CREATE

表がすでにある場合には、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。DATALINK 列を備えた表では無効です。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。

## 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

タイプ表で 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

タイプ表で 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走査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、警告は出ないはずであるけれども、正しいファイルと表が使用されているかどうかを検査した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を誤って指定した場合、インポート・ユーティリティーは、インポートを試みた行ごとに警告を生成して、それがインポートの失敗の原因になります。*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。

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 ユーティリティーが、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" 行が表に挿入され、成功しました。"0" 行が拒否されました。

#### 例 2

下記に示す例は、DEL 形式のデータをもった入力ファイル delfile1 から、表 MOVIE TABLE をインポートする方法を示す例です。

## IMPORT

```
db2 import from delfile1 of del
modified by dl1del|
insert into movietable (actorname, description, url_making_of,
url_movie) datalink specification (dl_url_default_prefix
"http://narang"), (dl_url_replace_prefix "http://bomdel"
dl_url_suffix ".mpeg")
```

注:

1. この表には下記の 4 つの列が示されます。

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

2. 入力ファイルの中の DATALINK データのサブフィールド区切り文字は、縦線 (|) 文字です。
3. url\_making\_of の列値に接頭部文字シーケンスが組み込まれていない場合、"http://narang" が使用されます。
4. url\_movie の非 NULL 列値には、接頭部として "http://bomdel" が付けられます。既存の値は置き換えられます。
5. url\_movie の非 NULL 列値のパスには、".mpeg" が付加されます。たとえば、url\_movie の列値が "http://server1/x/y/z" であれば "http://bomdel/x/y/z.mpeg" として格納されることになります。また、値が "/x/y/z" であれば "http://bomdel/x/y/z.mpeg" として格納されることになります。

### 例 3 (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 IDENTITY 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 IDENTITIY 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 IDENTITIY 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、IDENTITIY 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 を提供し、IDENTITIY 列が GENERATED ALWAYS であるからです。

#### 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・ユーティリティは、SQL INSERT ステートメントを使ってターゲット表に行を追加します。ユーティリティは入力ファイル中のデータの行ごとに 1 つの 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すると、以下のどちらかのアクションが起きます。

- 後続の INSERT ステートメントを正常に実行できると思われる場合、メッセージ・ファイルに警告メッセージが書き込まれ、処理は続行します。
- 後続の INSERT ステートメントが失敗しそうで、データベースの損傷の可能性がある場合、メッセージ・ファイルにエラー・メッセージが書き込まれ、処理は停止します。

ユーティリティは、REPLACE または REPLACE\_CREATE 操作中に、古い行が削除された後、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、古いデータがもう必要ないことを必ず確認してください。

ログが CREATE、REPLACE、または REPLACE\_CREATE 操作中にいっぱいになった場合、ユーティリティは挿入されたレコード上で自動 COMMIT を実行します。自動 COMMIT の後に、システムに障害が起こるか、またはアプリケーションがデータベース・マネージャーに割り込むと、部分的にデータの挿入された表はデータベース内に残ります。REPLACE または REPLACE\_CREATE オプションを使用してインポート操作全体をやり直すか、または正常にインポートされる行の数に設定した RESTARTCOUNT パラメーターを指定して INSERT を使用してください。

## IMPORT

デフォルトでは、自動 COMMIT は INSERT または INSERT\_UPDATE オプションには実行されません。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 を実行するときにはいつでも、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。1 つはコミットされるレコードの数を示し、もう 1 つは正常に終了した COMMIT の後に書き込まれます。失敗の後にインポート操作を再始動する場合、最後に正常に終了した COMMIT から決定されたとおり、スキップするレコードの数を指定してください。

インポート・ユーティリティーは、小さい非互換性問題 (たとえば、文字データは埋め込みまたは切り捨てを使ってインポートでき、数値データは異なる数値データ型を使ってインポートできる) は受け入れますが、大きな非互換性に関する問題は受け入れません。

それ自体以外への依存があるオブジェクト表や、基本表に何らかの依存 (それ自体も含めて) があるオブジェクト・ビューを、REPLACE または REPLACE\_CREATE することはできません。このような表またはビューを置換するには、次のようにします。

1. 表が親であ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。

参照制約および外部キー定義は、PC/IXF ファイルから表を作成する場合は保存されません。(主キー定義は、データが前に SELECT \* を使ってエクスポートされた場合、保存されます。)

リモート・データベースへのインポートには、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がサイズが大きくなる可能性に備えて、十分なディスク・スペースをサーバー上に確保する必要があります。

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。

ASC、DEL、または WSF のファイル形式をインポートするためには、それ以前にデータベース表または階層がすでに存在していなければなりません。ただし、表がまだ存在していない場合でも、IMPORT CREATE または IMPORT REPLACE\_CREATE を使えば、PC/IXF ファイルからデータをインポートする際に表が作成されます。タイプ表の場合、IMPORT CREATE によってタイプ階層と表階層も作成されます。

データ (階層データを含む) を別のデータベースに移動するには、PC/IXF インポートを使う必要があります。行区切り文字の入った文字データを区切り文字付き ASCII (DEL) ファイルにエクスポートし、テキスト転送プログラムにより処理を行うと、行区切り文字の入ったフィールドは長さが伸縮します。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。

ここでは、ASC ファイルおよび DEL ファイル内のデータは、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・ページのデータであると想定します。別々のコード・ページにデータをインポートする場合、PC/IXF ファイル (異なるコード・ページへのインポートが考慮されたファイル) の使用をお勧めします。PC/IXF ファイルとインポート・ユーティリティーが同一のコード・ページの場合、処理は通常のアプリケーションと同じようになります。両者が異なるコード・ページであっても、FORCEIN オプションが指定されている場合には、インポート・ユーティリティーにおいて、PC/IXF ファイル内のデータが、インポートを実行するアプリケーションと同一のコード・ページであると見なされます。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 UDB 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。このようになるのは、CHAR、VARCHAR、または CLOB 型の列の場合だけです。DEL または ASC ファイルのインポートでは、この制限は当てはまりません。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。PC/IXF インポート (INSERT オプション) だけがサポートされています。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。

タイプ表に対して CREATE オプションを使うと、PC/IXF ファイルの中で定義されているすべての副表が作成されます。副表定義は変更されません。タイプ表に対して CREATE 以外のオプションを使うと、走査順序リストによって、走査順序を指定できます。その場合、走査順序リストはエクスポート操作で使われたものと一致していなければなりません。PC/IXF ファイル・フォーマットの場合は、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走査順序を使用するだけです。

インポート・ユーティリティーを使って、前に PC/IXF ファイルにエクスポートされた表をリカバリーできます。表は、エクスポート時の状態に戻ります。

データは、システム表、宣言一時表、またはサマリー表にはインポートできません。

インポート・ユーティリティーを介してビューを作成することはできません。

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 PC/IXF ファイルのインポートはサポートされていません。
- 正しくない DEL フォーマット・ファイルの 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。

### DB2 Data Links Manager の考慮事項:

DB2 インポート・ユーティリティーを実行する場合、その前に下記のことを実行しておいてください。

1. 参照されるファイルを、適切なデータ・リンク・サーバーにコピーする。  
**dlfm\_import** ユーティリティーを使用することによって、**dlfm\_export** ユーティリティーの生成したアーカイブからファイルを取り出すことができます。
2. 必要な接頭部名を DB2 Data Links Manager に登録する。さらに、必要ならデータベースを登録するなど、その他の管理作業も必要になることがあります。
3. 必要なら、DATALINK 列の URL のデータ・リンク・サーバー情報を、SQL 表のエクスポート・データから更新します。(元の構成のデータ・リンク・サーバーがターゲットにおいても同じなら、そのデータ・リンク・サーバー名は更新する必要はありません。)
4. DB2 Data Links Manager 構成ファイルのターゲット構成で、Data Links サーバーを定義する。

インポート・ユーティリティーがターゲット・データベースに対して実行される場合、DATALINK 列データの参照するファイルが、該当するデータ・リンク・サーバー上でリンクされます。

挿入操作の間、DATALINK 列の処理では、ターゲット・データベースでの列指定に従って、該当するデータ・リンク・サーバー中のファイルがリンクされます。

### 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『インポートの使用』

## 関連資料:

- 「管理 API リファレンス」の『db2Import - インポート』
- 131 ページの『db2look - DB2 統計および DDL 抽出ツール』
- 「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』
- 558 ページの『LOAD』
- 497 ページの『インポート用のファイル・タイプ修飾子』
- 403 ページの『データ移動での区切り文字の制限』

## インポート用のファイル・タイプ修飾子

表 11. インポートで有効なファイル・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・ユーティリティーに、すべての生成列のデータはデータ・ファイルに存在するが、それらを見捨てる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ーによって生成されます。この修飾子は、generatedmissing 修飾子と共に使用することはできません。</p>
generatedmissing	<p>この修飾子が指定されている場合、ユーティリティーは、生成列のデータが入力データ・ファイルに入っていない (NULL も入っていない) ものを見なし、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・ユーティリティーに、IDENTITY 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、すべての識別値はユーティリティーによって生成されます。この動作は、GENERATED ALWAYS および GENERATED BY DEFAULT IDENTITY 列のどちらの場合も同じです。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>

## IMPORT

表 11. インポートで有効なファイル・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
identitymissing	この修飾子を指定すると、ユーティリティーは、IDENTITY 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 IDENTITY 列のどちらの場合も同じです。この修飾子は、identityignore 修飾子とともに使用することはできません。
lobsinfile	<p><i>lob-path</i> には、LOB データの入ったファイルへのパスを指定します。</p> <p>各パスには、データ・ファイル内で LOB ロケーション指定子 (LLS) によって示される 1 つ以上の LOB の入った、少なくとも 1 つのファイルが組み込まれます。LLS は、LOB ファイル・パスに保管されるファイル内の LOB のロケーションのストリング表現です。LLS のフォーマットは、<i>filename.ext.nnn.mmm/</i> です。<i>filename.ext</i> は LOB の入ったファイルの名前、<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。たとえば、ストリング <i>db2exp.001.123.456/</i> がデータ・ファイルに保存される場合、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 文節には、lobsinfile 修飾子のコンテキスト以外の意味はありません。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。たとえば、NULL LOB の LLS は <i>db2exp.001.7.-1/</i> です。</p>
no_type_id	単一の副表にインポートするときのみ有効です。これを使う場合として典型的な例は、REGULAR 表からデータをエクスポートした後、この修飾子を使ってインポート操作を呼び出してそのデータを単一の副表に変換する場合です。
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、デフォルトはロードされません。このオプションを指定せず、あるターゲット表列のためのソース列が明示的に指定されていない場合、以下のいずれかになります。</p> <ul style="list-style-type: none"> <li>• 列にデフォルトを指定できる場合、そのデフォルトがロードされます。</li> <li>• 列が NULL 可能で、デフォルトがその列に指定できない場合、NULL がロードされます。</li> <li>• 列が NULL 不可能で、デフォルト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。</li> </ul>
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。

表 11. インポートで有効なファイル・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 つまたは複数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、以下のとおりです。</p> <ul style="list-style-type: none"> <li>• DEL ファイルの場合、列に ".,," が指定された場合</li> <li>• ASC ファイルの場合、列の NULL 標識が yes に設定された場合</li> <li>• DEL/ASC/WSF ファイルの場合、列が不足している行、または元の指定では十分な長さのない行。</li> </ul> <p>このオプションが指定されていない場合、行インスタンスのソース列にデータがないと、以下のいずれかの処理が行われます。</p> <ul style="list-style-type: none"> <li>• 列が NULL 可能な場合、NULL がロードされます。</li> <li>• 列が NULL 不可能の場合、ユーティリティーはその行をリジェクトします。</li> </ul>

表 12. インポートで有効なファイル・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、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、文字データをアプリケーション・コード・ページからこのコード・ページに変換します。</p> <p>以下の規則が適用されます。</p> <ul style="list-style-type: none"> <li>• 純 DBCS (GRAPHIC)、混合 DBCS、および EUC では、区切り文字は x00 から x3F の範囲に制限されます。</li> <li>• nullindchar 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。</li> </ul> <p>注:</p> <ol style="list-style-type: none"> <li>1. codepage 修飾子を lobsinfile 修飾子と一緒に使うことはできません。</li> <li>2. コード・ページがアプリケーションのコード・ページからデータベースのコード・ページに変換されているときにデータの拡張が発生する場合は、データは切り捨てられ、データの消失が発生する可能性があります。</li> </ol>
dateformat="x"	<p>x はソース・ファイルの日付のフォーマットです。<sup>2</sup> 有効な日付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 桁の数字)  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 2 桁の数字。  D とは相互に排他的)  DDD - 元日から数えた日数 (001 から 366 の 3 桁の数字。  他の日または月エレメントと相互に排他)</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。たとえば、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>
noeofchar	<p>オプションのファイル終了文字 x'1A' は、ファイルの終わりとして認識されません。通常の文字の場合のように処理は継続されます。</p>

## IMPORT

表 12. インポートで有効なファイル・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>x はソース・ファイル内の時刻のフォーマットです。 <sup>2</sup> 有効な時刻エレメントは以下のとおりです。</p> <ul style="list-style-type: none"> <li>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)</li> <li>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)</li> <li>M     - 分 (0 から 59 の範囲の           1 桁または 2 桁の数)</li> <li>MM    - 分 (0 から 59 の範囲の 2 桁の数。           M と相互排他的)</li> <li>S     - 秒 (0 から 59 の範囲の           1 桁または 2 桁の数)</li> <li>SS    - 秒 (0 から 59 の範囲の 2 桁の数。           S と相互に排他的)</li> <li>SSSSS - 夜中の 12 時から数えた秒数           (00000 から 86399 の 5 桁の数。           他の時刻エレメントとは相互に排他的)</li> <li>TT    - 午前 / 午後の指定子 (AM または PM)</li> </ul> <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 12. インポートで有効なファイル・タイプ修飾子: ASCII ファイル・フォーマット (ASC/DEL) (続き)

修飾子	説明
timestampformat="x"	<p>x はソース・ファイル内のタイム・スタンプのフォーマットです。<sup>2</sup> 有効なタイム・スタンプ・エレメントは以下のとおりです。</p> <p>YYYY - 年 (0000 から 9999 の範囲の 4 桁の数字)  M - 月 (1 から 12 の範囲の 1 桁または 2 桁の数)  MM - 月 (01 から 12 の 2 桁の数。  M および MMM とは相互に排他的)  MMM - 月 (大文字小文字を区別しない月名の 3 文字の省略形。  M と MM とは相互に排他的)  D - 日 (1 から 31 の範囲の 1 桁または 2 桁の数)  DD - 日 (1 から 31 の 2 桁の数字。  D とは相互に排他的)  DDD - 元日から数えた日数 (001 から 366 の 3 桁の数字。  他の日または月のエレメントとは相互に排他的)  H - 時 (12 時間制の場合は 0 から 12、24 時間制では 0 から 24 の  範囲の 1 桁または 2 桁の数。)  HH - 時 (12 時間制の場合は 0 から 12、24 時間制では 0 から 24 の  範囲の 2 桁の数字。  H と相互に排他的)  M - 分 (0 から 59 の 1 桁または 2 桁の数字。)  MM - 分 (0 から 59 の範囲の 2 桁の数。  M (分) とは相互に排他的)  S - 秒 (0 から 59 の 1 桁または 2 桁の数字。)  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 5 桁の数。  他の時刻エレメントとは相互に排他的)  UUUUUU - マイクロ秒 (000000 から 999999 の範囲の 6 桁の数。  他のマイクロ秒エレメントとは相互に排他的)  UUUUU - マイクロ秒 (00000 から 99999 の範囲の 5 桁の数。  000000 から 999990 の範囲にマップされる。  他のマイクロ秒エレメントとは相互に排他的)  UUUU - マイクロ秒 (0000 から 9999 の範囲の 4 桁の数字。  000000 から 999900 の範囲にマップされる。  他のマイクロ秒エレメントとは相互に排他的)  UUU - マイクロ秒 (000 から 999 の範囲の 3 桁の数。  000000 から 999000 の範囲にマップされる。  他のマイクロ秒エレメントとは相互に排他的)  UU - マイクロ秒 (00 から 99 の範囲の 2 桁の数字。  000000 から 990000 の範囲にマップされる。  他のマイクロ秒エレメントとは相互に排他的)  U - マイクロ秒 (0 から 9 の範囲の 1 桁の数。  000000 から 900000 の範囲にマップされる。  他のマイクロ秒エレメントとは相互に排他的)  TT - 午前/午後の指定子 (AM または PM)</p> <p>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。これらの値は大文字小文字の区別をします。</p> <p>次の例では、ユーザー定義の日時フォーマットを指示するデータを、schedule という表にインポートする方法を示します。</p> <pre>db2 import from delfile2 of del modified by timestampformat="yyyy.mm.dd hh:mm tt" insert into schedule</pre>

## IMPORT

表 12. インポートで有効なファイル・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에만、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><b>制約事項</b></p> <p>usegraphiccodepage 修飾子は、EXPORT ユーティリティーで作成された DEL または ASC ファイルで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 13. インポートで有効なファイル・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	<p>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。たとえば、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。</p>
nullindchar=x	<p><math>x</math> は単一文字です。NULL 値を示す文字を <math>x</math> に変更します。<math>x</math> のデフォルト値は Y です。<sup>3</sup></p> <p>文字が 1 つの英字である場合を除いて、この修飾子は、EBCDIC データ・ファイルで大文字小文字を区別します。たとえば、NULL 標識文字が文字 N と指定されている場合、n も NULL 標識と認識されます。</p>
reclen=x	<p><math>x</math> は、最大値 32 767 の整数です。各行ごとに <math>x</math> 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>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。</p> <p><b>注:</b> このオプションは以前の t オプションを置き換えるもので、下位互換性のためにのみサポートされています。</p>



表 13. インポートで有効なファイル・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 (続き)

修飾子	説明
striptnulls	<p>可変長フィールドにデータをインポートする場合に、後書き NULL (0x00 文字) をすべて切り捨てます。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。</p> <p><b>注:</b> このオプションは、廃止された padwithzero オプション (下位互換性のためだけにサポートされる) に代わるものです。</p>

表 14. インポートで有効なファイル・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。<sup>34</sup> 文字ストリング区切り文字として明示的に二重引用符を指定したい場合、次のように指定します。</p> <pre>modified by chardel""</pre> <p>単一引用符 (') も、文字ストリングの区切り文字として指定できます。以下の例では、chardel' が指定されており、インポート・ユーティリティは検出するすべての単一引用符 (') を文字ストリングの区切り文字として解釈します。</p> <pre>db2 "import from myfile.del of del modified by chardel' method p (1, 4) insert into staff (id, years)"</pre>
coldelx	<p><i>x</i> は単一文字の列区切り文字です。デフォルト値はコンマ (,) です。指定した文字は、列の終わりを表すために、コンマの代わりに使用されます。<sup>34</sup></p> <p>以下の例では、coldel; が指定されており、インポート・ユーティリティ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>
datesiso	日付形式。すべての日付データ値を ISO フォーマットでインポートします。
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	<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。<sup>34</sup></p> <p>以下の例では、decpt; が指定されており、インポート・ユーティリティは検出するすべてのセミコロン (;) を小数点として解釈します。</p> <pre>db2 "import from myfile.del of del modified by chardel' decpt; messages msgs.txt insert into staff"</pre>

## IMPORT

表 14. インポートで有効なファイル・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
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>たとえば、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&lt;row delimiter&gt; "Vincent,&lt;row delimiter&gt;, is a manager", ... ... 4005,44.37&lt;row delimiter&gt;</pre> <p>delprioritychar 修飾子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 &lt;row delimiter&gt; は 2 番目の行の最初のデータ列の一部と解釈されますが、1 番目と 3 番目の &lt;row delimiter&gt; は実レコードの区切り文字と解釈されます。この修飾子が指定されていない場合、このデータ・ファイルでは 3 行になり、各行は &lt;row delimiter&gt; によって区切られます。</p>
dldelx	<p><i>x</i> は単一文字の <i>DATALINK</i> 区切り文字です。デフォルト値はセミコロン (;) です。指定した文字はセミコロンの代わりに、<i>DATALINK</i> 値のフィールド間区切り文字として使用されます。<i>DATALINK</i> 値には 2 つ以上の副値を指定できるので、この区切り文字が必要です。<sup>34</sup></p> <p><b>注:</b> <i>x</i> は、行、列、または文字ストリングの区切り文字とは異なる文字にしてください。</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空白を保持します。このオプションを指定しないと、区切り文字で囲まれていないすべての前後の空白は除去され、表のすべての空白・フィールドに <i>NULL</i> が挿入されます。</p>
nochardel	<p>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。文字切り文字は、列データの一部として構文解析されます。データが <i>DB2</i> を使用してエクスポートされている場合は、このオプションを指定しないでください (エクスポート時に <i>nochardel</i> が指定されない限り)。これは、区切り文字を持たないベンダー・データ・ファイルをサポートするために用意されています。不適切に使用すると、データが損失または破壊される場合があります。</p> <p>このオプションを <i>chardelx</i>、<i>delprioritychar</i> または <i>nodoubledel</i> と一緒に指定することはできません。これらは相互に排他的なオプションです。</p>
nodoubledel	<p>二重になっている区切り文字の認識を抑止します。</p>

表 15. インポートで有効なファイル・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・フィールドは、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> 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。</p>

表 15. インポートで有効なファイル・タイプ修飾子: IXF ファイル・フォーマット (続き)

修飾子	説明
indexixf	既存の表に現在定義されている索引をすべてドロップし、PC/IXF ファイルの索引定義に基づいて新しい索引を作成するようにユーティリティに指示します。このオプションを使用できるのは、表の内容を置換する場合だけです。ビューでは使用できません。また、 <i>insert-column</i> 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 <i>schema</i> を、索引作成時の索引名として使用します。 <i>schema</i> を指定しなかった場合 (しかしキーワード <i>indexschema</i> は指定した 場合) には、接続ユーザー <i>IDENTITY</i> 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	<i>nochecklengths</i> 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。たとえば、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。

**注:**

1. サポートされていないファイル・タイプを **MODIFIED BY** オプションで使用しようとしても、インポート・ユーティリティは警告を出しません。この場合、インポート操作が失敗し、エラー・コードが戻されます。
2. 日付形式ストリングは必ず二重引用符で囲まなければなりません。フィールド区切り文字には、*a* から *z*、*A* から *Z*、および *0* から *9* を使用することはできません。フィールド区切り文字は、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" (月)  
 "S:M" (分)  
 "M:YYYY:S:M" (月....分)  
 "M:H:YYYY:M:D" (分....月)

## IMPORT

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 (たとえば、¥) 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)。たとえば、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。

表 16. `codepage` および `usegraphiccodepage` 使用時の `IMPORT` 動作

<code>codepage=N</code>	<code>usegraphiccodepage</code>	<code>IMPORT</code> 動作
なし	なし	ファイル内のすべてのデータは、アプリケーション・コード・ページであると見なされます。
あり	なし	ファイル内のすべてのデータは、コード・ページ N であると見なされます。  <b>警告:</b> N が単一バイト・コード・ページの場合、GRAPHIC データをデータベースにインポートすると、壊れます。
なし	あり	ファイル内の文字データは、アプリケーション・コード・ページであると見なされます。 GRAPHIC データは、アプリケーション GRAPHIC データのコード・ページであると見なされます。  アプリケーション・コード・ページが単一バイトの場合は、すべてのデータはアプリケーション・コード・ページであると見なされます。  <b>警告:</b> アプリケーション・コード・ページが単一バイトの場合、GRAPHIC データは、データベースにたとえ GRAPHIC 列が収められていても、データベースにインポートされると壊れます。
あり	あり	文字データは、コード・ページ N であると見なされます。 GRAPHIC データは、N の GRAPHIC コード・ページであると見なされます。  N が単一バイトまたは 2 バイト・コード・ページの場合は、すべてのデータは、コード・ページ N であると見なされます。  <b>警告:</b> N が単一バイト・コード・ページの場合、GRAPHIC データをデータベースにインポートすると、壊れます。

### 関連資料:

- 「管理 API リファレンス」の『db2Import - インポート』
- 483 ページの『IMPORT』

- 403 ページの『データ移動での区切り文字の制限』

## 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 より大きくなければならない (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。たとえば、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。たとえば、以下のように指定します。

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

## IMPORT

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。たとえば、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。

## INITIALIZE TAPE

Windows NT ベースのオペレーティング・システムで実行する場合、DB2 は、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をサポートしています。このコマンドは、テープを初期化するのに使います。

**権限:**

なし

**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・サイズで初期化されます。

**注:** 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・サイズで初期化されます。値ゼロを指定した場合は、装置は可変長のブロック・サイズで初期化されます。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。

**関連資料:**

- 299 ページの『BACKUP DATABASE』
- 697 ページの『RESTORE DATABASE』
- 708 ページの『REWIND TAPE』
- 739 ページの『SET TAPE POSITION』

## INSPECT

データベースのページの整合性がとれているかどうかを調べることにより、データベースの構造上の保全性を検査します。この検査では、表オブジェクトの構造および表スペースの構造が有効かどうか調べられます。

### 有効範囲:

単一パーティション・システムでは、有効範囲はその単一パーティションに限定されます。パーティション・データベース・システムでは、`db2nodes.cfg` に定義されている論理パーティションすべてのコレクションです。

### 権限:

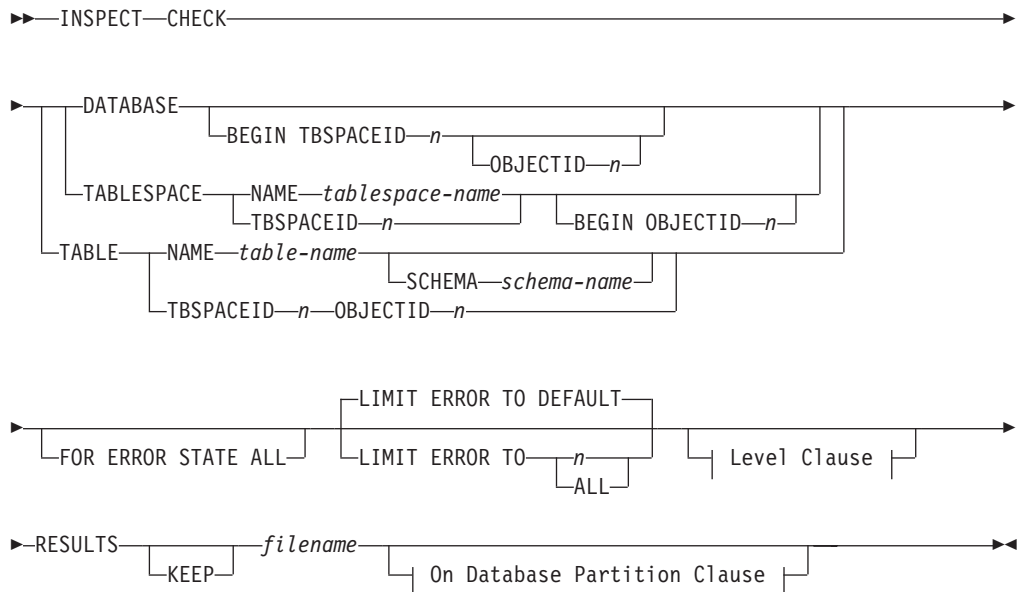
INSPECT CHECK の場合、以下のどれかになります。

- `sysadm`
- `dbadm`
- `sysctrl`
- `sysmaint`
- 単一表の場合 CONTROL 特権。

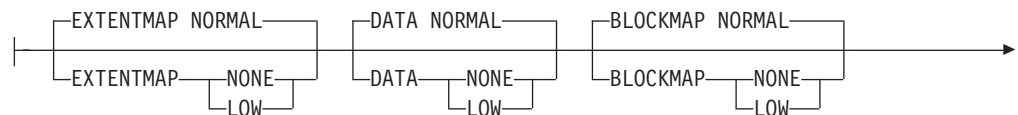
### 必要な接続:

データベース

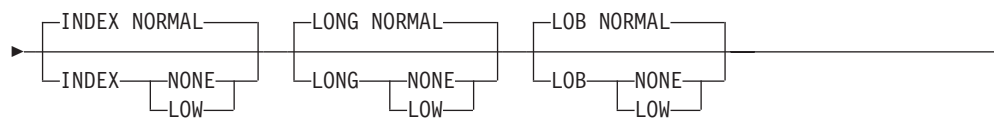
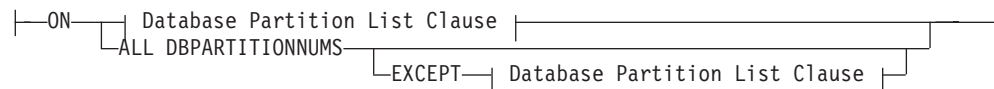
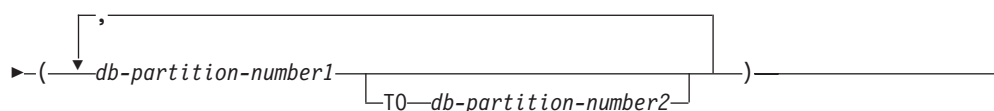
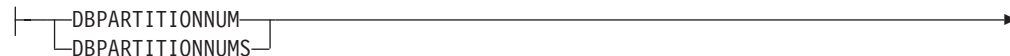
### コマンド構文:



### 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。

**FOR ERROR STATE ALL**

その内部状態がすで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、内部状態がすで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。

**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。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データベース・パーティションを除き、db2nodes.cfg ファイルに指定されているすべてのデータベース・パーティションで操作が実行されることを指定します。

**ON DBPARTITIONNUM / ON DBPARTITIONNUMS**

データベース・パーティションのセットに対して操作を実行します。

**db-partition-number1**

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

**db-partition-number2**

2 番目のデータベース・パーティション番号を指定し、db-partition-number1 から db-partition-number2 までのすべてのデータベース・パーティションがデータベース・パーティション・リストに含まれるようにします。

**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拡張子が付きます。パーティション・データベース環境では、各データベース・パーティションで、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。すで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。



## LIST ACTIVE DATABASES

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:

- 452 ページの『GET SNAPSHOT』
- 283 ページの『ACTIVATE DATABASE』
- 370 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。

### 権限:

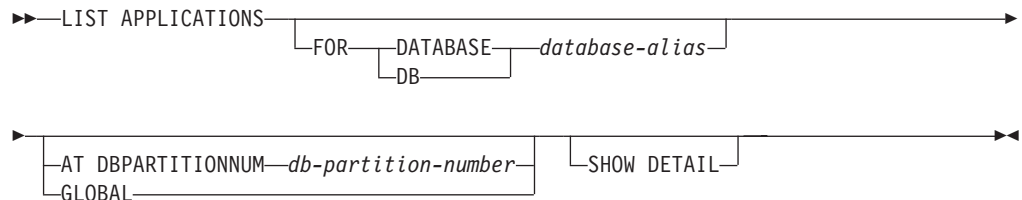
以下のどれかが必要です。

- *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

## LIST APPLICATIONS

### AT DBPARTITIONNUM db-partition-number

アプリケーションごと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

**注:** 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:

次に、LIST APPLICATIONS の出力例を示します。

Auth Id	Application Name	Appl. Handle	Application Id	DB Name	# of Agents
smith	db2bp_32	12	*LOCAL.smith.970220191502	TEST	1
smith	db2bp_32	11	*LOCAL.smith.970220191453	SAMPLE	1

### 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。アタッチがすで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:

- 452 ページの『GET SNAPSHOT』
- 405 ページの『FORCE APPLICATION』



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

権限:

なし

必要な接続:

なし

コマンド構文:

▶—LIST COMMAND OPTIONS—————▶▶

コマンド・パラメーター:

なし

例:

以下に示すのは、LIST COMMAND OPTIONS の出力例です。

Command Line Processor Option Settings		
Backend process wait time (seconds)	(DB2BQTIME) = 1	
No. of retries to connect to backend	(DB2BQTRY) = 60	
Request queue wait time (seconds)	(DB2RQTIME) = 5	
Input queue wait time (seconds)	(DB2IQTIME) = 5	
Command options	(DB2OPTIONS) =	
Option	Description	Current Setting
-a	Display SQLCA	OFF
-c	Auto-Commit	ON
-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:

- 784 ページの『UPDATE COMMAND OPTIONS』

520 コマンド・リファレンス

```

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/Database drive

データベースが存在しているディレクトリーまたはドライブ名。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

どのデータベース・パーティションをカタログ・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データベース・パーティションに対して割り当てられている番号を指定します。

### Alternate server hostname

データベースへの接続に通信障害がある場合に使用する代替サーバーのホスト名または IP アドレスを指定します。このフィールドは、システム・データベース・ディレクトリーに関してのみ表示されます。

### Alternate server port number

データベースへの接続に通信障害がある場合に使用する代替サーバーのポート番号を指定します。このフィールドは、システム・データベース・ディレクトリーに関してのみ表示されます。

### Alternate server node name

ディレクトリー項目タイプが LDAP の場合に、データベースへの接続に通信障害がある場合に使用する代替サーバーのノード名を指定します。

### Alternate server gateway node name

ディレクトリー項目タイプが LDAP の場合に、データベースへの接続に通信障害がある場合に使用する代替ゲートウェイのゲートウェイ・ノード名を指定します。

### 使用上の注意:

1 プロセスにつき、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させることができます。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:

- 353 ページの『CHANGE DATABASE COMMENT』
- 357 ページの『CREATE DATABASE』
- 779 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、`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。

### 権限:

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。

- パーティション・マップ ID
- データベース・パーティション番号
- 使用中フラグ

### 例:

以下に示すのは、LIST DATABASE PARTITION GROUPS コマンドの出力例です。

```

DATABASE PARTITION GROUP NAME
-----
IBMCATGROUP
IBMDEFAULTGROUP

```

2 record(s) selected.

以下に示すのは、LIST DATABASE PARTITION GROUPS SHOW DETAIL コマンドの出力例です。

```

DATABASE PARTITION GROUP NAME  PMAP_ID  DATABASE PARTITION NUMBER  IN_USE
-----
IBMCATGROUP                    0         0 Y
IBMDEFAULTGROUP                1         0 Y

```

2 record(s) selected.

各フィールドは、次のとおりです。

**DATABASE PARTITION GROUP NAME**

データベース・パーティション・グループの名前。この名前は、データベース・パーティション・グループのデータベース・パーティションごとに繰り返されます。

**PMAP\_ID**

パーティション・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・グループには追加されていますが、パーティション・マップには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 操作が正常に終了した場合、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・グループには追加されていますが、パーティション・マップには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:**

- 656 ページの『REDISTRIBUTE DATABASE PARTITION GROUP』

---

## LIST DATALINKS MANAGERS

指定されたデータベースに登録されている DB2 Data Links Manager のリストを表示します。

**権限:**

なし

**コマンド構文:**

►►—LIST DATALINKS MANAGERS FOR—

DATABASE
DB

—*dbname*—►◄

**コマンド・パラメーター:**

**DATABASE dbname**

データベース名を指定します。

**関連資料:**

- 287 ページの『ADD DATALINKS MANAGER』
- 382 ページの『DROP DATALINKS MANAGER』



# 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、 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。

**権限:**

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:**

- 656 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。

### 権限:

以下のどれかが必要です。

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

### 必要な接続:

インスタンス。 リモート・インスタンスの DCS アプリケーションをリスト表示するには、まず最初にそのインスタンスにアタッチする必要があります。

### コマンド構文:

```

▶▶—LIST DCS APPLICATIONS—
                                SHOW DETAIL
                                EXTENDED

```

### コマンド・パラメーター:

#### LIST DCS APPLICATIONS

デフォルトのアプリケーション情報の内容は、以下のとおりです。

- ホスト許可 ID (*username* )
- アプリケーション・プログラム名
- アプリケーション・ハンドル
- アウトバウンド・アプリケーション ID (*luwid*)

#### SHOW DETAIL

出力に以下の追加情報を含めることを指定します。

- クライアント・アプリケーション ID
- クライアント・シーケンス番号
- クライアント・データベース別名
- クライアント・ノード名 (*nname* )
- クライアント・リリース・レベル
- クライアント・コード・ページ
- アウトバウンド・シーケンス番号
- ホスト・データベース名
- ホスト・リリース・レベル

#### EXTENDED

拡張されたレポートを生成します。このレポートには、 SHOW DETAIL オプションを指定した場合に出力されるすべてのフィールドに加えて、以下の追加フィールドが含まれます。

- DCS アプリケーション状況

- 状況変更時刻
- クライアント・プラットフォーム
- クライアント・プロトコル
- クライアント・コード・ページ
- クライアント・アプリケーションのプロセス ID
- ホスト・コード化文字セット ID (CCSID)

例:

次に示すのは LIST DCS APPLICATIONS の出力例です。

Auth Id	Application Name	Appl. Handle	Outbound Application Id
-----	-----	-----	-----
DDCSUS1	db2bp_s	2	0915155C.139D.971205184245

次に示すのは LIST DCS APPLICATIONS EXTENDED の出力例です。

#### List of DCS Applications - Extended Report

Client application ID	= 09151251.0AD1.980529194106
Sequence number	= 0001
Authorization ID	= SMITH
Application name	= db2bp
Application handle	= 0
Application status	= waiting for reply
Status change time	= Not Collected
Client DB alias	= MVSDB
Client node	= antman
Client release level	= SQL05020
Client platform	= AIX
Client protocol	= TCP/IP
Client codepage	= 819
Process ID of client application	= 38340
Client login ID	= user1
Host application ID	= G9151251.GAD2.980529194108
Sequence number	= 0000
Host DB name	= GILROY
Host release level	= DSN05011
Host CCSID	= 500

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すでに確立され、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。

## LIST DCS APPLICATIONS

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:

- 405 ページの『FORCE APPLICATION』

第 3 章 CLP コマンド 531

## LIST DCS DIRECTORY

### 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 UDB データベース
- iSeries ホスト上の DB2 UDB データベース
- VSE & VM ホスト上の DB2 データベース

### 関連資料:

- 334 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。APPC コミット・プロトコルが使用されている場合は、パートナー LU 相互間の未確定トランザクションが表示されます。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。

**権限:**

*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。

**注:** 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)

**注:** コマンド文字と引き数の間は、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:**

## LIST DRDA INDOUBT TRANSACTIONS

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。APPC 接続では、LU 6.2 2 フェーズ・コミット・プロトコルを使います。



LIST HISTORY

履歴ファイルの中の項目のリストを表示します。履歴ファイルには、リカバリーと管理のさまざまなイベントの記録が含まれています。リカバリー・イベントには、データベース・レベルおよび表スペース・レベルのフル・バックアップ、増分バックアップ、リストア、およびロールフォワード操作が含まれます。さらにログ記録されるイベントには、表スペースの作成、変更、ドロップ、または名前変更、統計実行、表の再編成、表のドロップ、およびロードが含まれます。

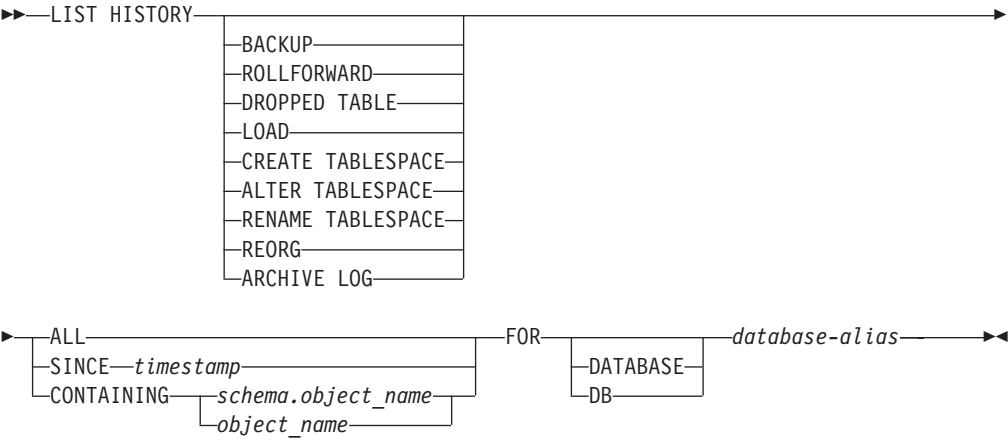
権限:

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を表示します。

ROLLFORWARD

ロールフォワード操作をリストを表示します。

DROPPED TABLE

ドロップした表レコードのリストを表示します。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。

LOAD ロード操作のリストを表示します。

### CREATE TABLESPACE

表スペースの作成およびドロップ操作のリストを表示します。

### RENAME TABLESPACE

表スペースの名前変更操作のリストを表示します。

### REORG

再編成操作のリストを表示します。

### ALTER TABLESPACE

表スペースの変更操作のリストを表示します。

### 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:

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
- X - アーカイブ・ログ

#### タイプ

アーカイブ・ログ・タイプ:

P - 1 次ログ・パス  
M - 2 次 (ミラー) ログ・パス  
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

## ロールフォワード・タイプ:

E - ログの最後  
P - ポイント・イン・タイム

## ロード・タイプ:

I - 挿入  
R - 置換

## 表スペースの変更タイプ:

C - コンテナの追加  
R - 再調整

## 静止タイプ:

S - 静止共有  
U - 静止更新  
X - 静止排他  
Z - 静止リセット

## LIST INDOUBT TRANSACTIONS

未確定トランザクションのリストを表示します。未確定トランザクションのコミット、ロールバック、または破棄を対話式で実行できます。

2 フェーズ・コミット・プロトコルは、以下のもので構成されます。

1. PREPARE フェーズ。このフェーズでは、リソース・マネージャーがログ・ページをディスクに書き込んでいるので、COMMIT または ROLLBACK プリミティブのどちらにも応答することができます。
2. COMMIT (または ROLLBACK) フェーズ。このフェーズでは、トランザクションの実際のコミットまたはロールバックが実行されます。

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。

### 権限:

*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:

```

▶—LIST INDOUBT TRANSACTIONS—┐
                               └─WITH PROMPTING─┘
  
```

### コマンド・パラメーター:

#### WITH PROMPTING

未確定トランザクションを処理することを指定します。このパラメーターを指定すると、対話式ダイアログ・モードが開始され、未確定トランザクションのコミット、ロールバック、または破棄が可能になります。このパラメーターを指定しない場合、未確定トランザクションは標準出力装置に書き込まれ、対話式ダイアログ・モードは開始されません。

対話式ダイアログ・モードでは次のことが可能です。

- すべての未確定トランザクションのリスト表示 (1 を入力)
- 未確定トランザクション番号 *x* のリスト表示 (1 の後に有効なトランザクション番号を入力)
- 終了 (q を入力)
- トランザクション番号 *x* をコミット (c の後に有効なトランザクション番号を入力)
- トランザクション番号 *x* をロールバック (r の後に有効なトランザクション番号を入力)

## LIST INDOUBT TRANSACTIONS

- 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)

**注:** コマンド文字と引き数の間は、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。

### 例:

次に示すのは、LIST INDOUBT TRANSACTIONS によって生成されるダイアログ例です。

In-doubt Transactions for Database SAMPLE

```
1.  originator: XA
    appl_id: *LOCAL.DB2.95051815165159      sequence_no: 0001 status: i
timestamp: 05-18-1997 16:51:59 auth_id: SMITH log_full: n type: RM
xid: 53514C2000000017 00000000544D4442 00000000002F93DD A92F8C4FF3000000
0000BD
```

```
2.  originator: XA
    appl_id: *LOCAL.DATABASE.950407161043 sequence_no: 0002 status: i
timestamp: 04-07-1997 16:10:43 auth_id: JONES log_full: n type: RM
xid: 53514C2000000017 00000000544D4442 00000000002F95FE B62F8C4FF3000000
0000C1
```

.  
.  
.

Enter in-doubt transaction command or 'q' to quit.

e.g. 'c 1' heuristically commits transaction 1.

c/r/f/l/q: c 1

```
1.  originator: XA
    appl_id: *LOCAL.DB2.95051815165159      sequence_no: 0001 status: i
timestamp: 05-18-1997 16:51:59 auth_id: SMITH log_full: n type: RM
xid: 53514C2000000017 00000000544D4442 00000000002F93DD A92F8C4FF3000000
0000BD
```

Do you want to heuristically commit this in-doubt transaction ? (y/n) y

DB20000I "COMMIT INDOUBT TRANSACTION" completed successfully

c/r/f/l/q: c 5

DB20030E "5" is not a valid in-doubt transaction number.

c/r/f/l/q: l

In-doubt Transactions for Database SAMPLE

```
1.  originator: XA
    appl_id: *LOCAL.DB2.95051815165159      sequence_no: 0001 status: c
timestamp: 05-18-1997 16:51:59 auth_id: SMITH log_full: n type: RM
xid: 53514C2000000017 00000000544D4442 00000000002F93DD A92F8C4FF3000000
0000BD
```

```
2.  originator: XA
    appl_id: *LOCAL.DATABASE.950407161043 sequence_no: 0002 status: i
```

## LIST INDOUBT TRANSACTIONS

```
timestamp: 04-07-1997 16:10:43 auth_id: JONES log_full: n type: RM
xid: 53514C2000000017 00000000544D4442 00000000002F95FE B62F8C4FF3000000
0000C1
```

```
.
.
```

```
c/r/f/l/q: r 2
```

```
2. originator: XA
   appl_id: *LOCAL.DATABASE.950407161043 sequence_no: 0002 status: i
timestamp: 04-07-1997 16:10:43 auth_id: JONES log_full: n type: RM
xid: 53514C2000000017 00000000544D4442 00000000002F95FE B62F8C4FF3000000
0000C1
```

```
Do you want to heuristically rollback this in-doubt transaction ? (y/n) y
```

```
DB20000I "ROLLBACK INDOUBT TRANSACTION" completed successfully
```

```
c/r/f/l/q: l 2
```

```
2. originator: XA
   appl_id: *LOCAL.DATABASE.950407161043 sequence_no: 0002 status: r
timestamp: 04-07-1997 16:10:43 auth_id: JONES log_full: n type: RM
xid: 53514C2000000017 00000000544D4442 00000000002F95FE B62F8C4FF3000000
0000C1
```

```
c/r/f/l/q: f 2
```

```
2. originator: XA
   appl_id: *LOCAL.DATABASE.950407161043 sequence_no: 0002 status: r
timestamp: 04-07-1997 16:10:43 auth_id: JONES log_full: n type: RM
xid: 53514C2000000017 00000000544D4442 00000000002F95FE B62F8C4FF3000000
0000C1
```

```
Do you want to forget this in-doubt transaction ? (y/n) y
```

```
DB20000I "FORGET INDOUBT TRANSACTION" completed successfully
```

```
c/r/f/l/q: l 2
```

```
2. originator: XA
   appl_id: *LOCAL.DATABASE.950407161043 sequence_no: 0002 status: f
timestamp: 04-07-1997 16:10:43 auth_id: JONES log_full: n type: RM
xid: 53514C2000000017 00000000544D4442 00000000002F95FE B62F8C4FF3000000
0000C1
```

```
c/r/f/l/q: q
```

**注:** 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 がもう一度使用可能になり、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それともロールバックするかわかりません。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) 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)、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。

破棄できるのは、状況がコミット済み (c) またはロールバック済み (r) のトランザクションだけです。

**注:** 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』

## 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。

権限:

なし

必要な接続:

なし

コマンド構文:



コマンド・パラメーター:

### ADMIN

Administration Server ノードを指定します。

### SHOW DETAIL

以下の情報を出力に含めることを指定します。

- リモート・インスタンス名
- システム
- オペレーティング・システムのタイプ

例:

次に示すのは LIST NODE DIRECTORY の出力例です。

Node Directory

Number of entries in the directory = 2

Node 1 entry:

Node name	= LANNODE
Comment	=
Directory entry type	= LDAP
Protocol	= TCPIP
Hostname	= LAN.db2ntd3.torolab.ibm.com
Service name	= 50000

Node 2 entry:

Node name	= TLBA10ME
Comment	=
Directory entry type	= LOCAL
Protocol	= TCPIP
Hostname	= tlba10me
Service name	= 447

次に示すのは LIST ADMIN NODE DIRECTORY の出力例です。

Node Directory

Number of entries in the directory = 2



## Node 1 entry:

```

Node name           = LOCALADM
Comment             =
Directory entry type = LOCAL
Protocol            = TCPIP
Hostname            = jaguar
Service name        = 523

```

## Node 2 entry:

```

Node name           = MYDB2DAS
Comment             =
Directory entry type = LDAP
Protocol            = TCPIP
Hostname            = peng.torolab.ibm.com
Service name        = 523

```

共通フィールドは、次のとおりです。

**Node name**

リモート・ノードの名前。これは、ノードのカタログ時に *nodename* パラメーターに入力された名前に対応します。

**Comment**

ノードのカタログ時に入力された、ノードに関連する注釈。ノード・ディレクトリー内の注釈を変更するには、そのノードをアンカタログしてから、別の注釈を付けてもう一度カタログします。

**Directory entry type**

LOCAL は、項目がローカル・ノードのディレクトリー・ファイルに見付かったことを意味します。LDAP は、項目が LDAP サーバーまたは LDAP キャッシュで見付かったことを意味します。

**Protocol**

ノード用にカタログされた通信プロトコル。

**注:** 個々のノード・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・アタッチが要求されると、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ディレクトリーを含めることができます。

- APPC
- APPCLU

## LIST NODE DIRECTORY

- APPN
- LDAP
- ローカル
- 名前付きパイプ
- NetBIOS
- TCP/IP

### 関連資料:

- 324 ページの『CATALOG APPC NODE』
- 349 ページの『CATALOG TCPIP NODE』
- 346 ページの『CATALOG NETBIOS NODE』
- 342 ページの『CATALOG LOCAL NODE』
- 327 ページの『CATALOG APPN NODE』
- 344 ページの『CATALOG NAMED PIPE NODE』
- 341 ページの『CATALOG LDAP NODE』

LIST ODBC DATA SOURCES

使用可能な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、指定した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可能です。

このコマンドは Windows のみで使用可能です。

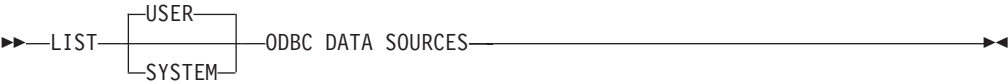
権限:

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:

- 348 ページの『CATALOG ODBC DATA SOURCE』
- 770 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。

### 権限:

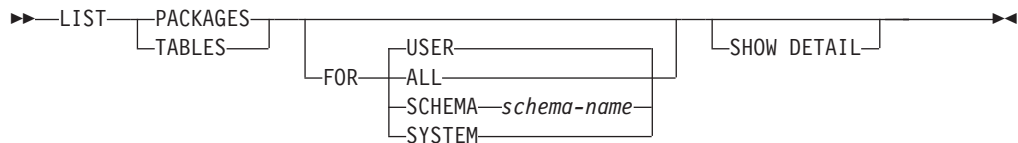
システム・カタログ SYSCAT.PACKAGES (LIST PACKAGES) と SYSCAT.TABLES (LIST TABLES) の場合には、少なくとも以下の 1 つが必要です。

- *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	Isolation Format 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』
- 「管理ガイド: 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。

**注:** 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。

### 権限:

以下のいずれかが必要です。

- *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
- 使用可能なページの数
- 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:

- 551 ページの『LIST TABLESPACES』



## 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。

**注:** このコマンドによって表示される情報は、表スペースのスナップショットでも使用できます。

### 有効範囲:

このコマンドは、それが実行されたデータベース・パーティションに対してだけ情報を戻します。

### 権限:

以下のいずれかが必要です。

- *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。たとえば、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

## LIST TABLESPACES

0x80	Roll forward pending
0x100	Restore pending
0x100	Recovery pending (not used)
0x200	Disable pending
0x400	Reorg in progress
0x800	Backup in progress
0x1000	Storage must be defined
0x2000	Restore in progress
0x4000	Offline and not accessible
0x8000	Drop pending
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
- 使用可能なページの数
- 使用されたページの数
- 未使用ページの数
- 限界値 (ページ単位)
- ページ・サイズ (バイト単位)
- エクステント・サイズ (ページ単位)
- プリフェッチ・サイズ (ページ単位)
- コンテナーの数
- 最小リカバリー時間 (0 以外の場合のみ表示)
- 状態変更表スペース ID (表スペース状態が "load pending" または "delete pending" の場合のみ表示)
- 状態変更オブジェクト ID (表スペース状態が "load pending" または "delete pending" の場合のみ表示)
- 静止者の数 (表スペース状態が "quiesced: SHARE"、"quiesced: UPDATE"、または "quiesced: EXCLUSIVE" の場合のみ表示)
- 各静止者ごとに表スペース ID とオブジェクト ID (静止者の数が 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                   = System 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                            = System 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

```

## LIST TABLESPACES

Type	= System managed space
Contents	= Temporary data
State	= 0x0000
Detailed explanation:	
Normal	
Total pages	= 1
Useable pages	= 1
Used pages	= 1
Free pages	= Not applicable
High water mark (pages)	= Not applicable
Page size (bytes)	= 4096
Extent size (pages)	= 32
Prefetch size (pages)	= 32
Number of containers	= 1
Tablespace ID	= 2
Name	= USERSPACE1
Type	= System managed space
Contents	= Any data
State	= 0x0000
Detailed explanation:	
Normal	
Total pages	= 1
Useable pages	= 1
Used pages	= 1
Free pages	= Not applicable
High water mark (pages)	= Not applicable
Page size (bytes)	= 4096
Extent size (pages)	= 32
Prefetch size (pages)	= 32
Number of containers	= 1
Tablespace ID	= 3
Name	= DMS8K
Type	= Database managed space
Contents	= Any data
State	= 0x0000
Detailed explanation:	
Normal	
Total pages	= 2000
Useable pages	= 1952
Used pages	= 96
Free pages	= 1856
High water mark (pages)	= 96
Page size (bytes)	= 8192
Extent size (pages)	= 32
Prefetch size (pages)	= 32
Number of containers	= 2
Tablespace ID	= 4
Name	= TEMP8K
Type	= System managed space
Contents	= Temporary data
State	= 0x0000
Detailed explanation:	
Normal	
Total pages	= 1
Useable pages	= 1
Used pages	= 1
Free pages	= Not applicable
High water mark (pages)	= Not applicable
Page size (bytes)	= 8192
Extent size (pages)	= 32
Prefetch size (pages)	= 32
Number of containers	= 1

DB21011I In a partitioned database server environment, only the table spaces on the current node are listed.

### 使用上の注意:

パーティション・データベース環境では、このコマンドがデータベースのすべての表スペースを戻すわけではありません。すべての表スペースのリストを表示するには、`SYSCAT.SYSTABLESPACES` を照会します。

表スペースのバランス調整操作中には、使用可能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、使用可能ページ数の値に等しくなります。

### 関連資料:

- 549 ページの『LIST TABLESPACE CONTAINERS』
- 245 ページの『db2tbst - 表スペース状態の獲得』

## 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表示します。

### 権限:

以下のいずれかが必要です。

- *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。たとえば、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。SET UTIL\_IMPACT\_PRIORITY コマンドでは、LIST UTILITIES コマンドで表示される ID をそのまま使用してください。

**関連資料:**

- 740 ページの『SET UTIL\_IMPACT\_PRIORITY』

---

## 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。ユーザー定義カーソルからデータをロードすることも可能です。

### 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。

### 権限:

以下のいずれかが必要です。

- *sysadm*
- *dbadm*
- データベースに対するロード権限と以下のもの
  - ロード・ユーティリティが INSERT モード、 TERMINATE モード (それまでのロード挿入操作を終了する)、または RESTART モード (以前のロード挿入操作を再開する) で呼び出された場合には、その表に対する INSERT 特権。
  - ロード・ユーティリティが REPLACE モード、 TERMINATE モード (それまでのロード置換操作を終了する)、または RESTART モード (以前のロード置換操作を再開する) で呼び出された場合には、その表に対する INSERT および DELETE 特権。
  - 例外表がロード操作の一部として使用される場合、その例外表に対する INSERT 特権。

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

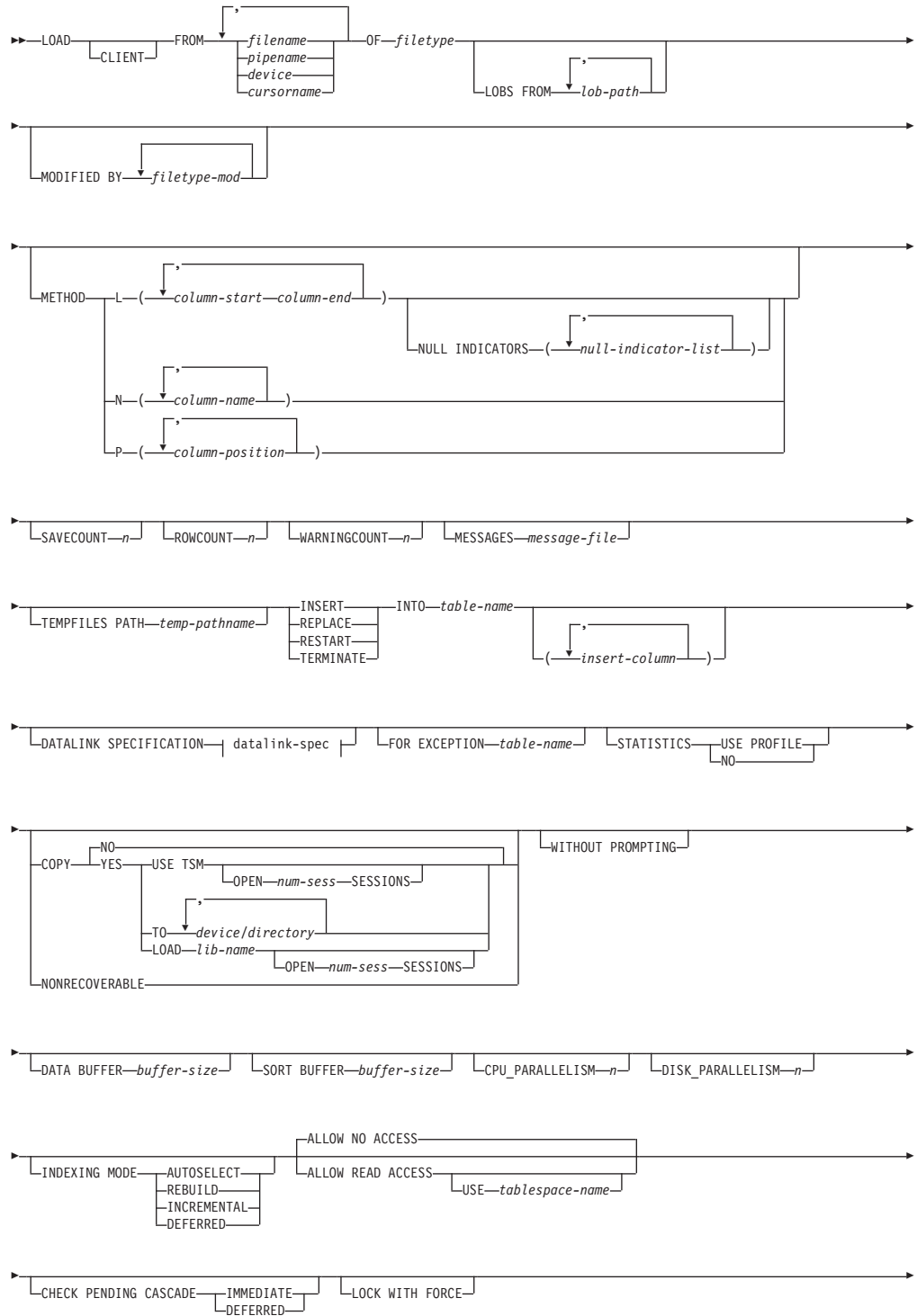
### 必要な接続:

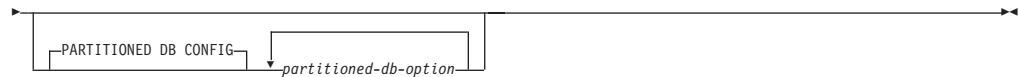
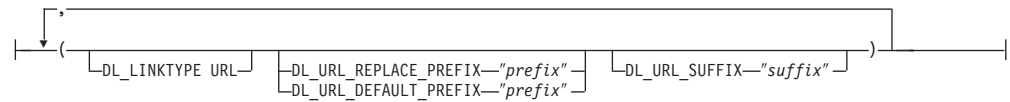
データベース。暗黙的な接続が可能である場合には、デフォルトのデータベースへの接続が確立されます。



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している場合には、ローカル・インスタンスへの暗黙的なアタッチが試行されます。

### コマンド構文:



**datalink-spec:****コマンド・パラメーター:****ALLOW NO ACCESS**

ロードを使用すると、ロード中に、排他的アクセスのためにターゲット表がロックされます。ロード中、表の状態は **LOAD IN PROGRESS** に設定されます。 **ALLOW NO ACCESS** はデフォルトの動作です。これは、**LOAD REPLACE** で唯一有効なオプションです。

表に制約があると、表の状態は、 **LOAD IN PROGRESS** の他に、**CHECK PENDING** に設定されます。表の **CHECK PENDING** を解除するには、**SET INTEGRITY** ステートメントを使用する必要があります。

**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、**LOAD IN PROGRESS** および **READ ACCESS** の両方に設定されます。表のロード中、データの非デルタ部分にアクセスすることができます。つまり、表を読み取る側はロードの開始前に存在していたデータにはアクセスができ、ロード中のデータはロードが完了するまで利用できない、ということです。 **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、 **LOAD IN PROGRESS**、**READ ACCESS** の他に、**CHECK PENDING** に設定されます。ロードの終了時に、表の状態 **LOAD IN PROGRESS** は解除されますが、 **CHECK PENDING** と **READ ACCESS** はそのまま残ります。表の **CHECK PENDING** を解除するには、**SET INTEGRITY** ステートメントを使用する必要があります。表が **CHECK PENDING** および **READ ACCESS** の状態にある間、データの非デルタ部分には引き続き読み取りアクセスできますが、データの新しい (デルタ) 部分には、 **SET INTEGRITY** ステートメントが完了するまでアクセス不能のままになります。ユーザーは、**SET INTEGRITY** ステートメントを発行しないで、同じ表上で複数のロードを実行できます。ただし、元の (チェック済み) データは、 **SET INTEGRITY** ステートメントが発行されるまで可視のままです。

**ALLOW READ ACCESS** は、以下の修飾子もサポートします。

**USE tablespace-name**

索引が再作成される場合、表スペース *tablespace-name* に索引のシャドー・コピーが作成され、ロード終了時の **INDEX COPY PHASE** で、元の表スペース上にコピーされます。このオプションと一緒に

使用できるのは、システム TEMPORARY 表スペースだけです。これを指定しないと、索引オブジェクトと同じ表スペースにシャドー索引が作成されます。索引オブジェクトと同じ表スペースにシャドー・コピーが作成される場合、シャドー索引オブジェクトは瞬間的に古い索引オブジェクトの上にコピーされます。シャドー・コピーが索引オブジェクトと別の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。

## CHECK PENDING CASCADE

LOAD によって表がチェック・ペンディング状態になる場合、CHECK 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

外部キー制約のチェック・ペンディング状態（読み取りまたは非アクセス・モード）が即時にすべての下層外部キー表に拡張されることを示します。表に下層即時マテリアライズ照会表または下層即時ステージング表がある場合、チェック・ペンディング状態は即時にマテリアライズ照会表およびステージング表に拡張されます。

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、チェック・ペンディング状態は下層外部キー表に拡張されないことに注意してください。

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、チェック・ペンディング読み取り状態だった下層外部キー表は、チェック・ペンディング非アクセス状態になります。

## DEFERRED

ロードされる表だけがチェック・ペンディング状態（読み取りまたは非アクセス・モード）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 コマンドの 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、後で暗黙的にチェック・ペンディング非アクセス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全性違反がチェックされる際、暗黙的にチェック・ペンディング非アクセス状態になります。従属表がチェック・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。この従属表がいつチェック・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。

CHECK 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、チェック・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、暗黙的にチェック・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チェック・ペンディング状態にならない場合、CHECK PENDING CASCADE オプションは無視されます。

### 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。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。

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
```

### COPY NO

順方向リカバリーが使用可能 (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。たとえば、

COPY NO を指定した LOAD および INDEXING MODE DEFERRED を実行すると、索引はリフレッシュが必要な状態になります。表での照会には、索引スキャンが必要なものがあり、索引がリフレッシュされるまで、成功しません。バックアップ・ペンディング状態にある表スペース内に常駐する場合、索引はリフレッシュできません。この場合、表へのアクセスは、バックアップが行われるまで許可されません。

**注:** 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、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。このオプションは DATALINK 列をもつ表ではサポートされません。

### 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 関数の入った共有ライブラリー (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。

### 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、ランタイムに自動的に計算された適切な値 (通常は使用可能な 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 IO 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。

#### DATA BUFFER buffer-size

ユーティリティー内でデータを転送するためのバッファ・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、ユーティリティー・ヒープから直接に割り当てられ、そのサイズは `util_heap_sz` データベース構成パラメーターで修正可能です。

値を指定しないと、ランタイムにユーティリティーによって適切なデフォルトが計算されます。デフォルトは、表の特性だけでなく、ローダーのインスタンス生成時にユーティリティー・ヒープ中で使用可能なフリー・スペースの割合に基づいています。

#### DATALINK SPECIFICATION

各 DATALINK 列ごとに、それぞれ 1 つの列指定を括弧で囲んで指定できます。各列指定は、1 つ以上の DL\_LINKTYPE、接頭部、および DL\_URL\_SUFFIX 指定で構成されます。接頭部指定は、DL\_URL\_REPLACE\_PREFIX または DL\_URL\_DEFAULT\_PREFIX のどちらかになります。

DATALINK 列指定の数は、表で定義されている DATALINK 列の数と同じだけ指定できます。指定の順序は、挿入列 リストの中での DATALINK 列の順序、または表定義内での順序 (挿入列 リストが指定されていない場合) に従います。

#### 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値を選択します。

#### DL\_LINKTYPE

指定した場合は、列定義の LINKTYPE に一致していなければなりません。そうすることによって、列定義に LINKTYPE URL が指定されている場合に DL\_LINKTYPE URL が受け入れ可能になります。

#### DL\_URL\_DEFAULT\_PREFIX "prefix"

これを指定すると、同じ列内のすべての DATALINK 値のデフォルト接頭部になります。ここでいう接頭部とは、URL 指定の「スキーム・ホスト・ポート」部分のことです。

接頭部の例

```
"http://server"
"file://server"
"file:"
"http://server:80"
```



列データの中に接頭部がない場合、DL\_URL\_DEFAULT\_PREFIX でデフォルトの接頭部が指定されている場合、列の値の接頭部としてそのデフォルト接頭部が付けられます (NULL でない場合)。

たとえば、DL\_URL\_DEFAULT\_PREFIX でデフォルト接頭部が "http://toronto" として指定されている場合、

- 列入力値 "/x/y/z" は "http://toronto/x/y/z" として保管されます。
- 列入力値 "http://coyote/a/b/c" は "http://coyote/a/b/c" として保管されます。
- 列入力値 NULL は NULL として保管されます。

#### **DL\_URL\_REPLACE\_PREFIX "prefix"**

この文節は、それ以前にエクスポート・ユーティリティーによって生成されたデータをロードまたはインポートする際に、ユーザーがデータに入っているホスト名を別のホスト名に一括置換したい場合に便利です。指定する場合には、それがすべての 非 NULL 列値の接頭部になります。列値にすでに接頭部がある場合、それは置き換えられます。列値に接頭部がない場合、DL\_URL\_REPLACE\_PREFIX で指定される接頭部がその列値の接頭部になります。

たとえば、DL\_URL\_REPLACE\_PREFIX で接頭部が "http://toronto" として指定されている場合、

- 列入力値 "/x/y/z" は "http://toronto/x/y/z" として保管されます。
- 列入力値 "http://coyote/a/b/c" は "http://toronto/a/b/c" として保管されます。"coyote" は "toronto" に置き換えられます。
- 列入力値 NULL は NULL として保管されます。

#### **DL\_URL\_SUFFIX "suffix"**

これを指定すると、それはその列のすべての非 NULL 列値に付加されます。これは実際には、DATALINK 値のデータ・ロケーション部分の「パス」コンポーネントに付加されます。

#### **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。DATALINK 例外も例外表にキャプチャーされます。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。

#### **FROM filename/pipename/device/cursorname**

ファイル、パイプ、装置、またはロードするデータを備えた SQL ステートメントを参照するカーソルを指定します。入力ソースがファイル、パイプ、または装置の場合、CLIENT オプションが指定されていなければ、データベースが存在するデータベース・パーティションになければなりません。複数の名前を指定すると、それらは順番に処理されます。最後に指定した項目がテープ装置の場合は、別のテープを使用するようユーザーに対してプロンプトが出ます。有効な応答オプションは、次のとおりです。

- c** 続行。警告メッセージを生成した装置を使用し続けます (たとえば、新しいテープがマウントされた場合)。
- d** 装置の終了。警告メッセージを生成した装置の使用を停止します (たとえば、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. 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 1 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。)
3. 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。

### 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 モードのどちらにするかを、ロード・ユーティリティーが自動的に決定します。

#### 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 (たとえば、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 (*logretain* または *userexit* が使用可能である)。
- 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。

この制限を回避するため、索引は別々の表スペースに置くようお勧めします。

## DEFERRED

このモードが指定されている場合、ロード・ユーティリティーは索引の作成を試みません。リフレッシュが必要であることを示すマークが索引に付けられます。ロード操作とは関係のないこのような索引に最初にアクセスするときは、再作成が強制的に実行されたり、データベースの再始動時に索引が再作成されたりする場合があります。このアプローチでは、最も大きい索引のキー部分をすべて処理できるだけのソート・スペースが必要です。索引を作成するためにその後かかる合計時間は、REBUILD モードの場合よりも長くなります。したがって、この索引作成据え置きモードで複数のロード操作を実行する場合、最初の非ロード・アクセス時に索引を再作成できるようにしておくよりも、順序列内の最後のロード操作で索引の再作成を実行できるようにした方が (パフォーマンスの観点から) 賢明であるといえます。

据え置き索引作成がサポートされるのは、非ユニークな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。

**注:** 据え置き索引作成は、DATALINK 列がある表ではサポートされません。

## 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。

### 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。たとえば、

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

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

### INTO table-name

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。

### 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。`filetype-mod` スtring内に `lobsinfile` が指定されていない場合、このオプションは無視されます。

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。

### LOCK WITH FORCE

ユーティリティーはロード・プロセス中に、表ロックなどの様々なロックを獲得します。ロックを獲得する際、このオプションを使用すると、ロードは待機することなく、またタイムアウトになることなく、ターゲット表に競合するロックを持つ他のアプリケーションを強制的にオフにします。システム・カタログ表に対する競合するロックを保留するアプリケーションは、ロード・ユーティリティーによって強制的にオフにされることはありません。強制されたアプリケーションは、ロールバックし、ロード・ユーティリティーが必要とするロックをリリースします。その後、ロード・ユーティリティーを続行できます。このオプションは、`FORCE APPLICATIONS` コマンドと同じ権限 (`SYSADM` または `SYSCTRL`) を必要とします。

`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。

### MESSAGES message-file

ロード操作中に生じ得る警告およびエラー・メッセージの宛先を指定します。メッセージ・ファイルを指定しなかった場合、メッセージは標準出力に書き込まれます。このファイルへの完全パスが指定されていない場合、ロード・ユーティリティーは現行のディレクトリーおよびデフォルトのドライブを宛先として使用します。すで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。

### 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。

**注:** このメソッドは、ASC ファイルの場合にのみ使用することができ、そのファイル・タイプに対してのみ有効なメソッドです。

**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。それらの列名の大文字小文字は、システム・カタログ中の対応する名前の大文字小文字と一致している必要があります。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。たとえば、データ・フィールドが F1、F2、F3、F4、F5、および F6 であり、表の列が C1 INT、C2 INT NOT NULL、C3 INT NOT NULL、および C4 INT の場合、method N (F2, F1, F4, F3) は有効な要求ですが、method N (F2, F1) は無効です。

**注:** この方式は、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。たとえば、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。

**注:** 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。

#### 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。『LOAD のファイル・タイプ修飾子』を参照してください。

#### 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。

FILE LINK CONTROL が指定された DATALINK 列が表に存在している場合や、そのような列を表に追加しようとしている場合には、このオプションを使用しないでください。

**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。

**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)。

**PARTITIONED DB CONFIG**

パーティション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。

```

HOSTNAME x
FILE_TRANSFER_CMD x
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・ロード構成オプション』に説明されています。

**REPLACE**

ロード・ユーティリティを実行できる 4 つのモードのうちの 1 つ。表の

既存データをすべて削除し、ロードされたデータを挿入します。表定義および索引定義は変更されません。階層間でデータを移動する際にこのオプションを使用する場合は、階層全体に関係したデータだけが置き換えられます。副表は置き換えられません。

このオプションは **DATALINK** 列をもつ表ではサポートされません。

## RESTART

ロード・ユーティリティを実行できる 4 つのモードのうちの 1 つ。以前に割り込みを受けたロード操作を再開します。ロード操作は、ロード、作成、または削除フェーズの最後の整合点から自動的に続行されます。

## RESTARTCOUNT

予約済み。

## 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。

## 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、エクステント・サイズのインターバルに切り上げられます。メッセージは整合点において発行されるので、**LOAD QUERY** を使用してロード操作をモニターする場合には、このオプションを選択する必要があります。 *n* の値が十分な大きさにない場合、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。

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。

## SORT BUFFER *buffer-size*

このオプションは、ロード操作時に **SORTHEAP** データベース構成パラメーターをオーバーライドする値を指定します。これは、索引とともに表をロードする場合、また **INDEXING MODE** パラメーターが **DEFERRED** として指定されていない場合にのみ関係があります。指定された値は **SORTHEAP** の値を超えることはありません。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。

## 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。

**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。

- DATALINK 値のある重複行またはリジェクト行ごとに 4 バイト
- 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 データがすでにある場合には、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。

**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。

**注:** このオプションは DATALINK 列をもつ表ではサポートされません。

**USING directory**

予約済み。

**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、警告は出ないはずであるけれども、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を誤って指定した場合、ロード・ユーティリティーは、ロードを試みた行ごとに警告を生成して、それがロードの失敗の原因になります。 $n$  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されます。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。

**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。

例:

**例 1**

TABLE1 に以下の 5 つの列があるとします。

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

ASCFILE1 に以下の 6 つのエレメントがあるとします。

- ELE1、位置 01 から 20
- ELE2、位置 21 から 22
- 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 reflen=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 列中の空白が切り捨てられるようになります (たとえば行 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 (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 形式と互換でない開始文字の入った列は、エラーを生成し、レコードはリジェクトされます。そのようなレコードは、ダンプ・ファイルに書き込むことができます。

区切り文字の外側にある列の DEL データは無視されますが、警告が生成されません。例:

## LOAD

```
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 (DATALINK データのロード)

下記のコマンドは、DEL 形式のデータをもった入力ファイル delfile1 から、表 MOVIE TABLE をロードします。

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

注:

1. この表には下記の 4 つの列が入っています。

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

2. 入力ファイルの中の DATALINK データのサブフィールド区切り文字は、縦線 (|) 文字です。
3. url\_making\_of の列値に接頭部文字シーケンスが組み込まれていない場合、"http://narang" が使用されます。
4. url\_movie の非 NULL 列値には、接頭部として "http://bomdel" が付けられます。既存の値は置き換えられます。
5. url\_movie の非 NULL 列値のパスには、".mpeg" が付加されます。たとえば、url\_movie の列値が "http://server1/x/y/z" であれば "http://bomdel/x/y/z.mpeg" として格納されることになります。また、値が "/x/y/z" であれば "http://bomdel/x/y/z.mpeg" として格納されることになります。
6. 表のロード中にユニーク索引または DATALINK の例外が発生すると、影響を受けるレコードは表から削除されて例外表 excptab に入れます。

### 例 5 (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 ID 値の提供がないので、それらの行のための ID 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 ID 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 6 (CURSOR ファイル・タイプを使用したロード)

表 ABC.TABLE1 には次の 3 つの列があります。

```
ONE INT
TWO CHAR(10)
THREE DATE
```

表 ABC.TABLE2 には次の 3 つの列があります。

## LOAD

```
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
```

### 使用上の注意:

データは、入力ファイル内に並んでいる順序でロードされます。特定の順序にしたい場合には、ロードが試行される前にデータをソートしてください。

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には、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サマリー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組み込まれた表は、チェック・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、チェック・ペンディング状態になります。表のチェック・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、複製されたサマリー表に対しては実行できません。

クラスタリング索引が表に存在する場合、ロード前にクラスタリング索引でデータをソートしてください。ただし、データはマルチディメンション・クラスタリング (MDC) 表にロードする前にソートする必要はありません。

### DB2 Data Links Manager の考慮事項:

各 DATALINK 列ごとに、括弧内にそれぞれ 1 つの列を指定できます。それぞれの列の指定は、1 つ以上の DL\_LINKTYPE、接頭部、および DL\_URL\_SUFFIX 指定で構成されます。接頭部 情報は、DL\_URL\_REPLACE\_PREFIX、または DL\_URL\_DEFAULT\_PREFIX の指定のいずれかになります。

DATALINK 列指定の数は、表で定義されている DATALINK の数と同じだけ定義できます。指定の順序は、挿入列リストの中での DATALINK 列の順序 (挿入列リストが INSERT INTO (insert-column, ...) で指定されている場合) か、または表定義内での順序 (insert-column が指定されていない場合) に従います。

たとえば、表に列 C1、C2、C3、C4、および C5 があり、そのうち C2 と C5 だけが DATALINK 型であり、挿入列 (insert-column) リストが (C1、C5、C3、C2) である場合、2 つの DATALINK 列を指定する必要があります。最初の列の指定は C5 用であり、2 番目の列の指定は C2 用です。挿入列リストを指定しない場合、最初の列の指定は C2 用になり、2 番目の列の指定は C5 用になります。

複数の DATALINK 列があり、一部の列では特別な指定が必要ではない場合、列の指定には、指定の順序をはっきりと示すために、少なくとも括弧を使用する必要があります。どの列にも指定が行われない場合は、空の括弧のリスト全体を除くことができます。したがって、デフォルトで十分な場合には、DATALINK を指定する必要はありません。

FILE LINK CONTROL で定義されている DATALINK 列を備えた表にデータをロードする場合は、ロード・ユーティリティを呼び出す前に、以下のステップを実行してください。(すべての DATALINK 列が NO LINK CONTROL として定義されている場合、これらのステップは必要ありません。)

1. DATALINK 列の値によって参照されるデータ・リンク・サーバーに、DB2 Data Links Manager がインストールされていることを確認する。
2. データベースが DB2 Data Links Manager に登録されていることを確認します。
3. DATALINK 値として挿入されるすべてのファイルを、適切なデータ・リンク・サーバーにコピーします。
4. データ・リンク・サーバー上の DB2 Data Links Manager に接頭部名を定義します。
5. (ロードする) DATALINK データによって参照されるデータ・リンク・サーバーを、DB2 Data Links Manager 構成ファイルに登録します。

ロード・ユーティリティの実行中に、DB2 とデータ・リンク・サーバーの間の通信に障害が発生することがあります。そのような場合、ロード操作は失敗します。その場合には、以下のようにしてください。

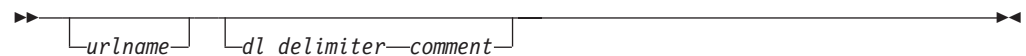
1. データ・リンク・サーバーおよび DB2 Data Links Manager を開始します。
2. ロード再開操作を起動します。

ロード操作中に失敗したリンクはデータ保全性違反と見なされ、ユニーク索引違反と同じような方法で処理されます。そのため、1 つ以上の DATALINK 列の入った表をロードする場合の特別な例外が定義されています。

### 入力ファイルでの DATALINK 情報の表現

LINKTYPE (現在のところ URL のみサポート) は、DATALINK 情報の一部として指定されていません。LINKTYPE は、LOAD または IMPORT コマンドで指定されますが、PC/IXF タイプの入力ファイルの場合は、適切な列記述子レコードの中で指定されます。

URL LINKTYPE の DATALINK 情報の構文は、以下のとおりです。



*urlname* と *comment* はいずれもオプションです。どちらも省略した場合、NULL 値が代入されます。

#### urlname

この URL 名は有効な URL 構文に適合していなければなりません。

#### 注:

1. 現在のところ、「http」、「file」、および「unc」がスキーマ名として許可されています。
2. URL 名の接頭部 (スキーマ、ホスト、およびポート) はオプションです。接頭部がない場合は、ロード・ユーティリティまたはインポート・ユーティリティの DL\_URL\_DEFAULT\_PREFIX または DL\_URL\_REPLACE\_PREFIX 指定の接頭部が使用されます。そのどちらも指定されていない場合、デフォルトの接頭部として "file://localhost" が

## LOAD

使われます。したがって、ローカル・ファイルの場合、LOAD または IMPORT コマンド内で DATALINK 列を指定せずに、絶対パス名で指定したファイル名を URL 名として入力することができます。

3. 接頭部が URL 名に付けられている場合も、ロードまたはインポート操作時には、DL\_URL\_REPLACE\_PREFIX で指定した異なる接頭部名によってオーバーライドされます。
4. (DL\_URL\_SUFFIX を指定した場合、それを追加した後の) 「path」は、リモート・サーバーにあるリモート・ファイルの絶対パス名です。相対パス名は使用できません。HTTP サーバーのデフォルトのパス接頭部は使用されません。

### dl\_delimiter

区切り付き ASCII (DEL) ファイル形式の場合、dlde1 修飾子で指定した文字、あるいは LOAD または IMPORT コマンドのデフォルトの文字。区切りなし ASCII (ASC) ファイル形式の場合、これを文字順序 ¥; (円記号とそれに続くセミコロン) に対応させる必要があります。空白文字 (ブランクやタブなど) を、このパラメーターに指定した値の前後に置くことができます。

### comment

DATALINK 値のコメント部分。区切り付き ASCII (DEL) ファイル形式で指定する場合、comment テキストは、文字ストリング区切り文字で囲む必要があります。文字ストリング区切り文字は、デフォルトでは二重引用符 (") です。この文字ストリング区切り文字は、LOAD または IMPORT コマンドで MODIFIED BY *filetype-mod* を指定することによりオーバーライドできます。

コメントを指定しない場合、このコメントはデフォルトでは長さがゼロのストリングになります。

次に示すのは、区切り付き ASCII (DEL) ファイル形式での DATALINK データの例です。

- `http://www.almaden.ibm.com:80/mrep/intro.mpeg; "Intro Movie"`

これは、以下のような部分から構成されています。

- スキーム = http
- サーバー = www.almaden.ibm.com
- パス = /mrep/intro.mpeg
- 注釈 = "Intro Movie"

- `file://narang/u/narang; "InderPal's Home Page"`

これは、次のような部分から構成されています。

- スキーム = file
- サーバー = narang
- パス = /u/narang
- 注釈 = "InderPal's Home Page"

次に示すのは、区切りなし ASCII (ASC) ファイル形式での DATALINK データの例です。

- `http://www.almaden.ibm.com:80/mrep/intro.mpeg`; Intro Movie

これは、以下のような部分から構成されています。

- スキーム = `http`
- サーバー = `www.almaden.ibm.com`
- パス = `/mrep/intro.mpeg`
- 注釈 = `"Intro Movie"`

- `file://narang/u/narang`; InderPal's Home Page

これは、次のような部分から構成されています。

- スキーム = `file`
- サーバー = `narang`
- パス = `/u/narang`
- 注釈 = `"InderPal's Home Page"`

以下に、DATALINK データの例を示します。列のロードまたはインポート指定が `DL_URL_REPLACE_PREFIX ("http://qso")` であるとしています。

- `http://www.almaden.ibm.com/mrep/intro.mpeg`

これは、以下のような部分から構成されています。

- スキーマ = `http`
- サーバー = `qso`
- パス = `/mrep/intro.mpeg`
- 注釈 = `NULL` スtring

- `/u/me/myfile.ps`

これは、次のような部分から構成されています。

- スキーマ = `http`
- サーバー = `qso`
- パス = `/u/me/myfile.ps`
- 注釈 = `NULL` スtring

#### 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『ロードの使用』

#### 関連資料:

- 638 ページの『QUIESCE TABLESPACES FOR TABLE』
- 22 ページの『db2atld - オートローダー』
- 「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『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・ロード構成オプション』
- 「管理 API リファレンス」の『db2Load - ロード』
- 582 ページの『ロード用のファイル・タイプ修飾子』

## ロード用のファイル・タイプ修飾子

表 17. ロードで有効なファイル・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 &gt; 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・ユーティリティーに、すべての生成列のデータはデータ・ファイルに存在するが、それらは無視すべきことを知らせます。この結果、すべての生成列の値はユーティリティーによって生成されます。この修飾子は、 <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、ユーティリティーは、生成列のデータが入力データ・ファイルに入っていない (NULL も入っていない) のものと見なします。この結果、すべての生成列の値はユーティリティーによって生成されます。この修飾子は、 <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。
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、<code>ROLLFORWARD DATABASE</code> コマンドで <code>RECOVER DROPPED TABLE</code> 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)。</p> <p><b>注:</b> この修飾子が使用される場合、表はチェック・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 <code>CHECK PENDING</code> 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 &lt; table-name &gt; GENERATED COLUMN IMMEDIATED UNCHECKED</pre> <p>表の <code>CHECK PENDING</code> 状態を解除し、ユーザー定義の値のチェックを強制するには、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 &lt; table-name &gt; IMMEDIATE CHECKED</pre> <p>この修飾子は、<code>generatedmissing</code> または <code>generatedignore</code> 修飾子と共に使用することはできません。</p>



表 17. ロードで有効なファイル・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
identityignore	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 ID 列のどちらの場合も同じです。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。
identitymissing	この修飾子を指定すると、ユーティリティーは、ID 列のデータが入力データ・ファイルに入っていない (NULL も入っていない) ものを見なし、行ごとに値を生成します。この動作は、GENERATED ALWAYS および GENERATED BY DEFAULT ID 列のどちらの場合も同じです。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。
identityoverride	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。 注: 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。
indexfreespace=x	x 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。  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。

## LOAD

表 17. ロードで有効なファイル・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。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』修飾子を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 文節には、“lobsinfile” 修飾子のコンテキスト以外の意味はありません。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 のフォーマットは、<i>filename.ext.nnn.mmm/</i> です。<i>filename.ext</i> は LOB の入ったファイルの名前、<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。たとえば、ストリング db2exp.001.123.456/ がデータ・ファイルに保存される場合、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。たとえば、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>オートローダー・ユーティリティーは、複数パーティションのデータベース・パーティション・グループの表にデータを提供している各ファイルに、ヘッダーを書き込みます。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。ヘッダーが組み込まれないファイルを使って操作を実行したい場合にのみ、noheader 修飾子を使用する必要があります。</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。
pagefreespace=x	<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 (たとえば、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。</p> <p>注: 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>

表 17. ロードで有効なファイル・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
subtableconvert	単一副表へのロードの場合のみ有効。これを使う場合として典型的な例は、正規の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ロード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。
totalfreespace= $x$	<p><math>x</math> 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。たとえば、<math>x</math> 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。</p> <p><b>注:</b> このオプションを指定して 2 つのロードが行われる場合、2 番目のロードは、最初のロードによって最後に付加された余分のスペースを再利用しません。</p>
usedefaults	<p>ターゲット表の列のソース列が指定されているが、1 つまたは複数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"> <li>• DEL ファイルの場合、列に ".,," が指定された場合</li> <li>• DEL/ASC/WSF ファイルの場合、列の数が不足している行、または元の指定に対して十分な長さではない行。</li> </ul> <p>このオプションが指定されていない場合、行インスタンスのソース列にデータがないと、以下のいずれかの処理が行われます。</p> <ul style="list-style-type: none"> <li>• その列が NULL 可能な場合は、NULL がロードされます。</li> <li>• 列が NULL 可能でない場合、ユーティリティーはその行をリジェクトします。</li> </ul>

表 18. ロードで有効なファイル・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><math>x</math> は ASCII 文字ストリングです。この値は、入力データ・セット内のデータのコード・ページとして解釈されます。ロード操作時に、文字データ (および文字内で指定された数値データ) は、このコード・ページからデータベースのコード・ページへ変換されます。</p> <p>以下の規則が適用されます。</p> <ul style="list-style-type: none"> <li>• 純粋な DBCS (GRAPHIC)、混合 DBCS、および EUC の場合、区切り文字は x00 から x3F の範囲に制限されます。</li> <li>• EBCDIC コード・ページで指定された DEL データの場合、区切り文字は DBCS のシフトイン文字およびシフトアウト文字と一致しない場合があります。</li> <li>• 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。</li> </ul> <p>ファイル・タイプが CURSOR の場合、このオプションはサポートされていません。</p>

表 18. ロードで有効なファイル・タイプ修飾子: ASCII ファイル・フォーマット (ASC/DEL) (続き)

修飾子	説明
dateformat="x"	<p>x は、ソース・ファイルの日付のフォーマットです。<sup>1</sup> 有効な日付要素は次のとおりです。</p> <p>YYYY - 年 (0000 から 9999 の範囲の 4 桁の数字)  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 2 桁の数字。  D とは相互に排他的)  DDD - 元日から数えた日数 (001 から 366 の 3 桁の数字。  他の日または月要素とは  相互に排他的)</p> <p>デフォルト値の 1 が、指定されない各要素に割り当てられます。日付形式の例を以下に示します。</p> <p>"D-M-YYYY"  "MM.DD.YYYY"  "YYYYDDD"</p>
dumpfile = x	<p>x は、リジェクトされた行を書き込む例外ファイルの (サーバー・データベース・パーティションによる) 完全修飾名です。 1 レコードにつき、最大で 32 KB のデータが書き込まれます。以下に、ダンプ・ファイルの指定方法の例を示します。</p> <pre>db2 load from data of del modified by dumpfile = /u/user/filename insert into table_name</pre> <p>ファイルは、インスタンスの所有者によって作成されて所有されます。デフォルトのファイル許可をオーバーライドするには、 dumpfileaccessall ファイル・タイプ修飾子を使用します。</p> <p>注:</p> <ol style="list-style-type: none"> <li>パーティション・データベース環境の場合、パスはロードを実行するデータベース・パーティションにローカルなものではありません。それによって、並行して実行される複数のロード操作が同じファイルに書き込むことを防ぐことができます。</li> <li>ファイルの内容は、非同期バッファ・モードでディスクに書き込まれます。ロード操作が失敗した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、 <b>LOAD RESTART</b> 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。</li> <li>この修飾子では、ファイル拡張子が複数あるファイル名はサポートされません。たとえば、以下のよう指定します。</li> </ol> <pre>dumpfile = /home/svtdbm6/DUMP.FILE</pre> <p>ロード・ユーティリティでは、上のファイル名は受け入れられます。</p> <pre>dumpfile = /home/svtdbm6/DUMP.LOAD.FILE</pre> <p>しかし、このファイル名は受け入れられません。</p>
dumpfileaccessall = x	<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"> <li>dumpfile ファイル・タイプ修飾子と一緒に使用された場合。</li> <li>ロード・ターゲット表に対してユーザーが <b>SELECT</b> 特権をもっている場合。</li> <li>UNIX ベースのオペレーティング・システムに置かれている DB2 サーバー・データベース・パーティション上で発行された場合。</li> </ol>
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> <p>たとえば、ASC ファイル内の整数列のフィールド項目として 123qwr4 の値が検出された場合、この値は有効な数値を表すものではないので、ロード・ユーティリティは通常、構文エラーのフラグを付けます。 fastparse を指定した場合、構文エラーは検出されず、整数フィールドに任意の数値がロードされます。この修飾子を使用する場合は、正しい形式のデータだけを使用するように注意してください。 ASCII データでこのオプションを使用した場合のパフォーマンスの向上は絶大です。</p> <p>ファイル・タイプが <b>CURSOR</b> または <b>IXF</b> の場合、このオプションはサポートされていません。</p>

表 18. ロードで有効なファイル・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
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。たとえば、値 12345 は、12345.00 ではなく、123.45 として DECIMAL(8,2) 列にロードされます。</p> <p>この修飾子は、packeddecimal 修飾子と共に使用することはできません。</p>
timeformat="x"	<p>x はソース・ファイル内の時刻のフォーマットです。 <sup>1</sup> 有効な時刻エレメントは以下のとおりです。</p> <p>H      - 時 (12 時間制の場合は 0 から 12、           24 時間制では 0 から 24 の範囲の           1 桁または 2 桁の数)</p> <p>HH     - 時 (12 時間制の場合は 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)</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 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 18. ロードで有効なファイル・タイプ修飾子: ASCII ファイル・フォーマット (ASC/DEL) (続き)

修飾子	説明
timestampformat="x"	<p>x はソース・ファイル内のタイム・スタンプのフォーマットです。<sup>1</sup> 有効なタイム・スタンプ・エレメントは以下のとおりです。</p> <p>YYYY - 年 (0000 から 9999 の範囲の 4 桁の数字)  M - 月 (1 から 12 の範囲の 1 桁または 2 桁の数)  MM - 月 (01 から 12 の 2 桁の数。  M と MMM とは相互に排他的)  MMM - 月 (大文字小文字を区別しない月名の 3 文字の省略形。  M と MM とは相互に排他的)  D - 日 (1 から 31 の範囲の 1 桁または 2 桁の数)  DD - 日 (1 から 31 の 2 桁の数字。  D とは相互に排他的)  DDD - 元日から数えた日数 (001 から 366 の 3 桁の数字。  他の日または月のエレメントとは相互に排他的)  H - 時 (12 時間制の場合は 0 から 12、24 時間制では 0 から 24 の  範囲の 1 桁または 2 桁の数。)  HH - 時 (12 時間制の場合は 0 から 12、24 時間制では 0 から 24 の  範囲の 2 桁の数字。  H と相互に排他的)  M - 分 (0 から 59 の 1 桁または 2 桁の数字。)  MM - 分 (0 から 59 の範囲の 2 桁の数。  M (分) とは相互に排他的)  S - 秒 (0 から 59 の 1 桁または 2 桁の数字。)  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 5 桁の数。  他の時刻エレメントとは相互に排他的)  UUUUUU - マイクロ秒 (000000 から 999999 の範囲の 6 桁の数。  他のマイクロ秒エレメントとは相互に排他的)  UUUUU - マイクロ秒 (00000 から 99999 の範囲の 5 桁の数。  000000 から 999990 の範囲にマップされる。  他のマイクロ秒エレメントとは相互に排他的)  UUUU - マイクロ秒 (0000 から 9999 の範囲の 4 桁の数字。  000000 から 999900 の範囲にマップされる。  他のマイクロ秒エレメントとは相互に排他的)  UUU - マイクロ秒 (000 から 999 の範囲の 3 桁の数。  000000 から 999000 の範囲にマップされる。  他のマイクロ秒エレメントとは相互に排他的)  UU - マイクロ秒 (00 から 99 の範囲の 2 桁の数字。  000000 から 990000 の範囲にマップされる。  他のマイクロ秒エレメントとは相互に排他的)  U - マイクロ秒 (0 から 9 の範囲の 1 桁の数。  000000 から 900000 の範囲にマップされる。  他のマイクロ秒エレメントとは相互に排他的)  TT - 午前/午後の指定子 (AM または PM)</p> <p>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。これらの値は大文字小文字の区別をします。</p> <p>次の例では、ユーザー定義の日時形式を指示するデータを、schedule という表に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>
noeofchar	オプションのファイル終了文字 x'1A' は、ファイルの終わりとして認識されません。それが普通の文字であるものとして処理は継続されます。



表 18. ロードで有効なファイル・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><b>制約事項</b></p> <p>usegraphiccodepage 修飾子は、EXPORT ユーティリティで作成された DEL または ASC ファイルで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 19. ロードで有効なファイル・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。 noeofchar オプションが前提となります。</p> <p>以下の規則が適用されます。</p> <ul style="list-style-type: none"> <li>• BIGINT、INTEGER、および SMALLINT を除き、データ・タイプ間の変換は実行されません。</li> <li>• データ長は、それぞれのターゲット列定義と一致している必要があります。</li> <li>• FLOAT は、IEEE 浮動小数点フォーマットでなければなりません。</li> <li>• ロード・ソース・ファイル中のバイナリー・データは、ロード操作を実行するプラットフォームに関係なく、ビッグ・エンディアンであると見なされます。</li> </ul> <p><b>注:</b> 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。たとえば、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。列定義が一致しなくてもソース・データがきちんと収まる事が明らかな場合に、このオプションは特に便利です。</p>
nullindchar=x	<p>x は単一文字です。 NULL 値を示す文字を x に変更します。 x のデフォルト値は y です。<sup>2</sup></p> <p>文字が 1 つの英字である場合を除いて、この修飾子は EBCDIC データ・ファイルで大文字小文字を区別します。たとえば、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>

## LOAD

表 19. ロードで有効なファイル・タイプ修飾子: ASC ファイル・フォーマット (区切り文字で区切られていない ASCII) (続き)

修飾子	説明
packeddecimal	<p>binarynumerics 修飾子は DECIMAL フィールド・タイプで構成されないため、パック 10 進数データを直接ロードします。</p> <p>このオプションがサポートされるのは、定位置 ASC において、reclen オプションによって固定長レコードが指定されている場合だけです。noeofchar オプションが前提となります。</p> <p>符号ニブル用にサポートされる値は以下のとおりです。</p> <pre> + = 0xC 0xA 0xE 0xF - = 0xD 0xB </pre> <p>この修飾子の影響を受ける列のデータに NULL があってはなりません。この修飾子を使用すると、ブランク (通常は NULL と解釈される) は、バイナリー値であると解釈されます。</p> <p>サーバーのプラットフォームには関係なく、ロードのソース・ファイルに入っているバイナリー・データのバイト順はビッグ・エンディアンであることが前提となっています。つまり、この修飾子を Windows オペレーティング・システムで使用する場合も、バイト順を逆にしてはなりません。</p> <p>この修飾子は、implieddecimal 修飾子と共に使用することはできません。</p>
reclen=x	<p>x は、32,767 以下の整数です。各行ごとに x 個の文字が読み取られ、行の終わりを示すのに改行文字は使用されません。</p>
striptblanks	<p>データを可変長フィールドにロードする際に、後書きブランク・スペースを切り捨てます。このオプションを指定しないと、ブランク・スペースは保持されます。</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。</p> <p>注: 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と、NULL は保持されます。</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。</p> <p>注: 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>
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。NOEOFCHAR オプションが前提となります。</p> <p>ハーフバイト符号値は、以下のいずれかになります。</p> <pre> + = 0xC 0xA 0xE 0xF - = 0xD 0xB </pre> <p>サポートされている数値は、0x0 から 0x9 です。</p> <p>サポートされているゾーン値は、0x3 および 0xF です。</p>



表 20. ロードで有効なファイル・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><math>x</math> は単一文字のストリング区切り文字です。デフォルトは二重引用符 (") です。指定した文字は、文字ストリングを囲むために、二重引用符の代わりに使用されます。<sup>23</sup> 文字ストリング区切り文字として明示的に二重引用符 (") を指定したい場合、以下のように指定します。</p> <pre>modified by charde1""</pre> <p>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 charde1''</pre>
coldelx	<p><math>x</math> は、単一文字の列区切り文字です。デフォルト値はコンマ (,) です。指定した文字は、列の終わりを表すために、コンマの代わりに使用されます。<sup>23</sup></p>
datesiso	日付形式。すべての日付データ値を ISO フォーマットでロードします。
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、10 進値に正符号の接頭部が付けられます。
decptx	<p><math>x</math> は、小数点文字としてピリオドの代わりに使用される単一の置換文字です。デフォルト値はピリオド (.) です。指定した文字は、小数点文字としてピリオドの代わりに使用されます。<sup>23</sup></p>
delprioritychar	<p>区切り文字の現在のデフォルト優先順位は、(1) レコード区切り文字、(2) 区切り文字、(3) 列区切り文字です。この修飾子は、区切り文字の優先順位を区切り文字、レコード区切り文字、列区切り文字と逆順にすることにより、以前の優先順位に依存する既存のアプリケーションを保護します。構文は以下のとおりです。</p> <pre>db2 load ... modified by delprioritychar ...</pre> <p>たとえば、以下のような DEL データ・ファイルがあるとします。</p> <pre>"Smith, Joshua",4000,34.98&lt;row delimiter&gt; "Vincent,&lt;row delimiter&gt;, is a manager", ... ... 4005,44.37&lt;row delimiter&gt;</pre> <p>delprioritychar 修飾子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 &lt;row delimiter&gt; は 2 番目の行の最初のデータ列の一部と解釈されますが、最初と 3 番目の &lt;row delimiter&gt; は実レコードの区切り文字と解釈されます。この修飾子を指定しなかった場合は、このデータ・ファイルは 3 行のままで、各行は &lt;row delimiter&gt; によって区切られます。</p>
dldelx	<p><math>x</math> は、単一文字の DATALINK 区切り文字です。デフォルト値はセミコロン (;) です。指定した文字は、DATALINK 値のフィールド間区切り文字として、セミコロンの代わりに使用されます。DATALINK 値には 2 つ以上の副値を指定できるので、この区切り文字が必要です。<sup>234</sup></p> <p><b>注:</b> 行、列、または文字ストリング区切り文字と同じ文字を <math>x</math> に指定することはできません。</p>

## LOAD

表 20. ロードで有効なファイル・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
keepblanks	<p>タイプが CHAR、VARCHAR、LONG VARCHAR、または CLOB の各フィールドの前後のブランクを保持します。このオプションを指定しないと、区切り文字で囲まれていないすべての前後のブランクは除去され、表のすべてのブランク・フィールドに NULL が挿入されます。</p> <p>以下の例では、データ・ファイルにある前後のブランクを保存しながら、TABLE1 という表にデータをロードする方法を示します。</p> <pre>db2 load from delfile3 of del       modified by keepblanks       insert into table1</pre>
nochardel	<p>ロード・ユーティリティは、列区切り文字と列区切り文字の間にあるすべてのバイトが列データの一部であると見なします。文字切り文字は、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。</p> <p>このオプションを chardelex、delprioritychar または nodoubledel と一緒に指定することはできません。これらのオプションは、相互に排他的です。</p>
nodoubledel	二重区切り文字の認識を抑止します。

表 21. ロードで有効なファイル・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。nochecklengths を指定した場合、そのような検査は実行されず、各行のロードが試みられます。</p>
nochecklengths	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。たとえば、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。列定義が一致しなくてもソース・データがきちんと収まることが明らかな場合に、このオプションは特に便利です。

### 注:

- 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" (月)
"S:M" (分)
"M:YYYY:S:M" (月....分)
"M:H:YYYY:M:D" (分....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 (たとえば、`¥`) を付けなければなりません。

- 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)。たとえば、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 ...
```

- 『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。
- DATALINK 区切り文字が URL 構文内で有効な文字である場合でも、ロード操作の有効範囲内では、その特別な意味はなくなります。
- 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 22. `codepage` および `usegraphiccodepage` 使用時の **LOAD** 動作

<b>codepage=N</b>	<b>usegraphiccodepage</b>	<b>LOAD 動作</b>
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 <b>N</b> であると見なされます。</p> <p><b>警告:</b> <b>N</b> が単一バイト・コード・ページの場合、<b>GRAPHIC</b> データをデータベースにロードすると壊れます。</p>

## LOAD

表 22. `codepage` および `usegraphiccodepage` 使用時の `LOAD` 動作 (続き)

<code>codepage=N</code>	<code>usegraphiccodepage</code>	LOAD 動作
なし	あり	<p>CLIENT オプションの指定があっても、ファイル内の文字データは、データベース・コード・ページであると見なされます。 CLIENT オプションの指定があっても、GRAPHIC データは、データベース GRAPHIC データのコード・ページであると見なされます。</p> <p>データベース・コード・ページが単一バイトの場合は、すべてのデータはデータベース・コード・ページであると見なされます。</p> <p><b>警告:</b> 単一バイト・データベースに GRAPHIC データをロードすると、壊れます。</p>
あり	あり	<p>文字データは、コード・ページ N であると見なされます。 GRAPHIC データは、N の GRAPHIC コード・ページであると見なされます。</p> <p>N が単一バイトまたは 2 バイト・コード・ページの場合は、すべてのデータは、コード・ページ N であると見なされます。</p> <p><b>警告:</b> N が単一バイト・コード・ページの場合、GRAPHIC データをデータベースにロードすると壊れます。</p>

### 関連資料:

- 558 ページの『LOAD』
- 「管理 API リファレンス」の『db2Load - ロード』
- 403 ページの『データ移動での区切り文字の制限』

---

## 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 より大きくなければならない (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。たとえば、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。たとえば、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。たとえば、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。

## 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。

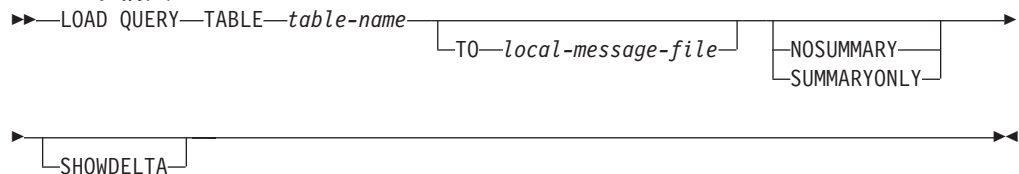
**権限:**

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) のレポートだけを生成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。ファイルがすでに存在する場合、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。表状態の判別には、LOAD QUERY コマンドを使用できます。LOAD QUERY は、現在ロードされていない表に対しても使用できます。LOAD QUERY で説明される表状態については、『表ロック、表状態、および表スペース状態』に説明されています。

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『表ロック、表状態、および表スペース状態』

#### 関連資料:

## LOAD QUERY

- 558 ページの『LOAD』
- 556 ページの『LIST UTILITIES』



## MIGRATE DATABASE

旧バージョンの DB2 データベースを現行の形式へ変換します。

**重要:** DB2 バージョン 8 のインストール (Windows オペレーティング・システムの場合)、またはインスタンスの移行 (UNIX ベース・システム) の前には、データベース事前移行ツールを実行しておく必要があります。それは DB2 バージョン 8 では実行できないためです。Windows では、事前以降ツールは db2ckmig です。UNIX システムでは、db2imigr が類似のタスクを実行します。移行作業、および Windows オペレーティング・システムへの DB2 バージョン 8 のインストール作業に先立って、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。

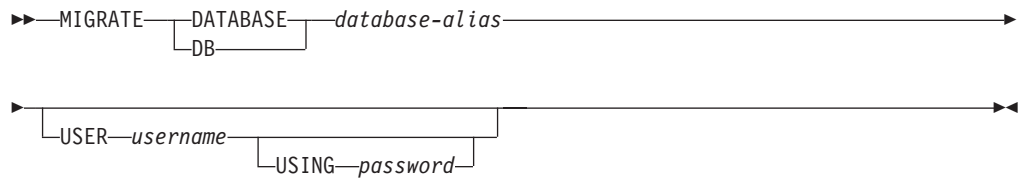
**権限:**

*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しておく必要があります。

## MIGRATE DATABASE

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。たとえば、移行中にログが一杯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。データベースがすで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 (「ログが一杯」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)。

データベースがバージョン 8 に移行される際に、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。

### 関連資料:

- 761 ページの『**TERMINATE**』

## 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。

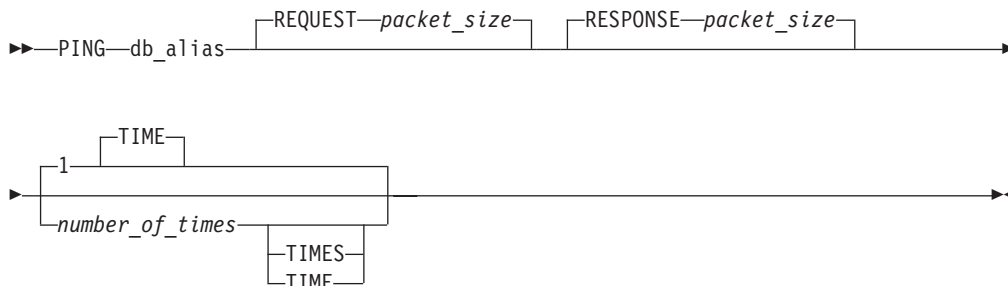
**権限:**

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 が設定されている DRDA サーバーのデータベース別名を指定します。

**注:** 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 Universal Database (UDB) for Linux、UNIX、および Windows のバージョン 8 以降、または DB2 UDB for z/OS のバージョン 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 Universal Database (UDB) for Linux、UNIX、および Windows のバージョン 8 以降、または DB2 UDB for z/OS のバージョン 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。値は、1 から 32767 までにしてください。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

## PING

ホスト・データベース *hostdb* への 1 回の接続に対するネットワーク応答時間をテストするには、次のように行います。

```
db2 ping hostdb 1
または
db2 ping hostdb
```

コマンドは、次の例と類似した出力を表示します。

```
Elapsed time: 7221 microseconds
```

### 例 2

ホスト・データベース *hostdb* への 5 回の接続に対するネットワーク応答時間をテストするには、次のように行います。

```
db2 ping hostdb 5
または
db2 ping hostdb 5 times
```

コマンドは、次の例と類似した出力を表示します。

```
Elapsed time: 8412 microseconds
Elapsed time: 11876 microseconds
Elapsed time: 7789 microseconds
Elapsed time: 10124 microseconds
Elapsed time: 10988 microseconds
```

### 例 3

100 バイトの REQUEST パケットおよび 200 バイトの RESPONSE パケットを使用して、ホスト・データベース *hostdb* への接続のネットワーク応答時間をテストするには、次のようにします。

```
db2 ping hostdb request 100 response 200
または
db2 ping hostdb request 100 response 200 1 time
```

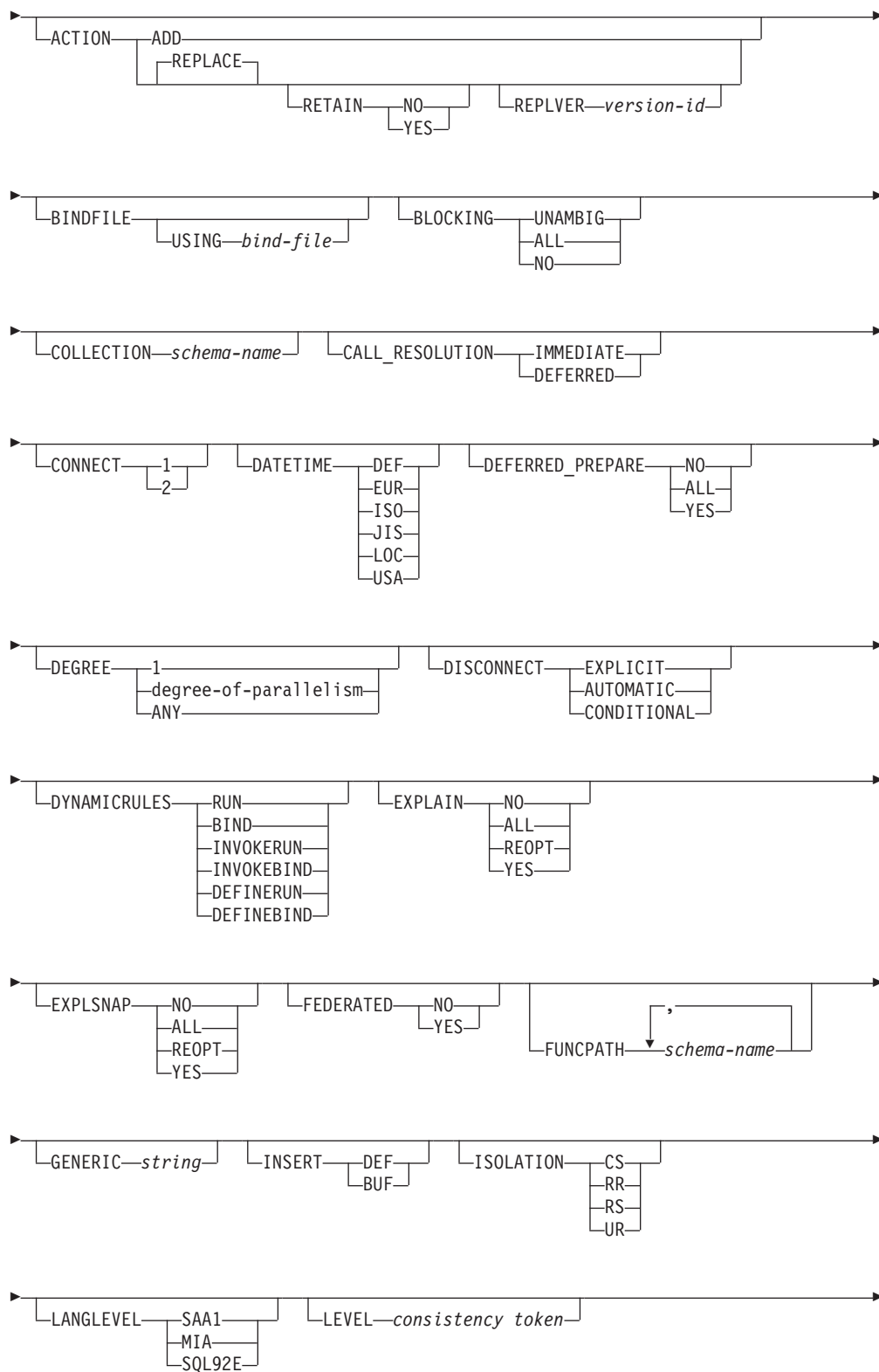
### 使用上の注意:

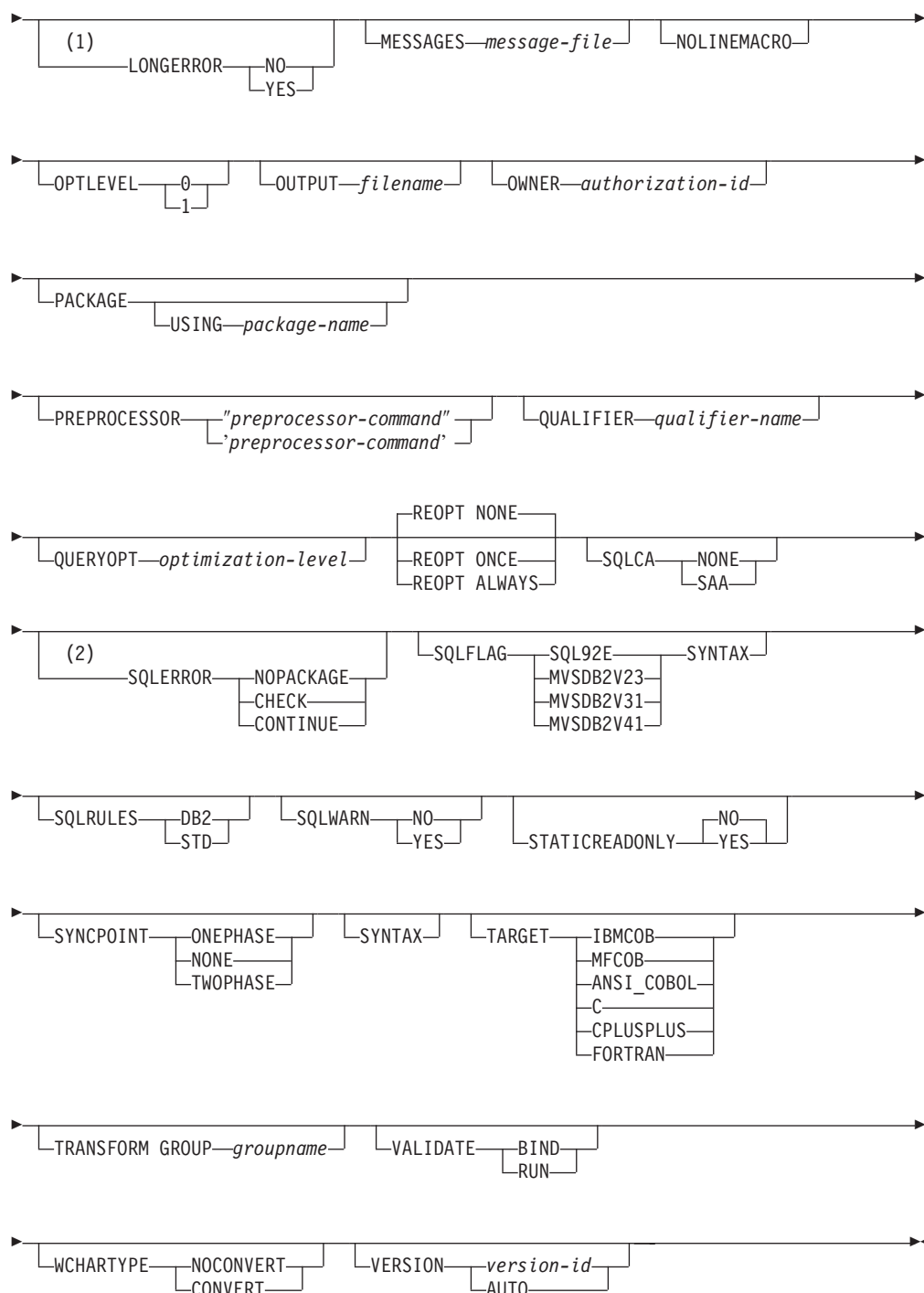
データベース接続は、このコマンドを呼び出す前に存在している必要があります。存在していない場合、エラーが起きます。

戻される経過時間は、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。

►► PRECOMPILE *filename*

## PRECOMPILE

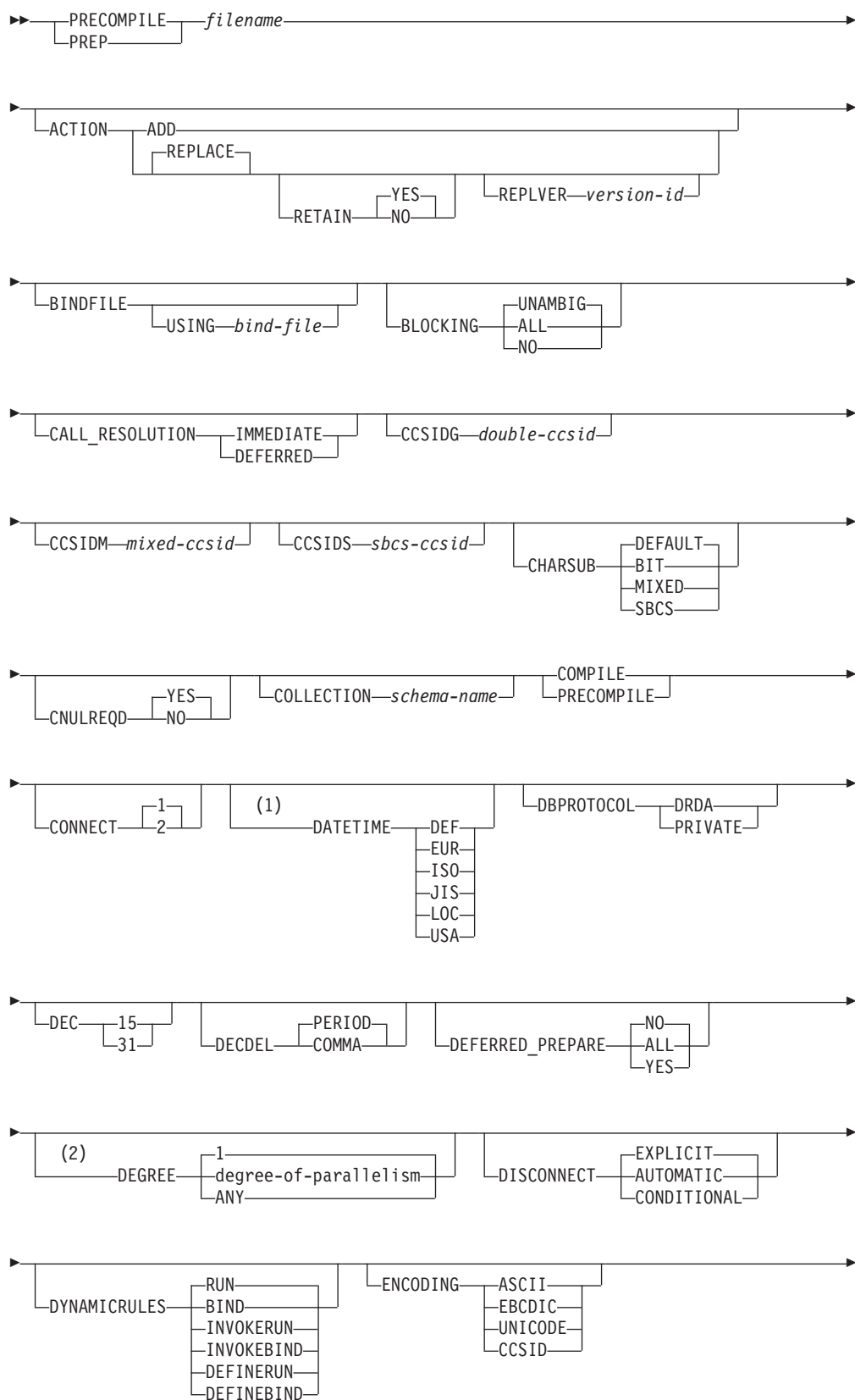


**注:**

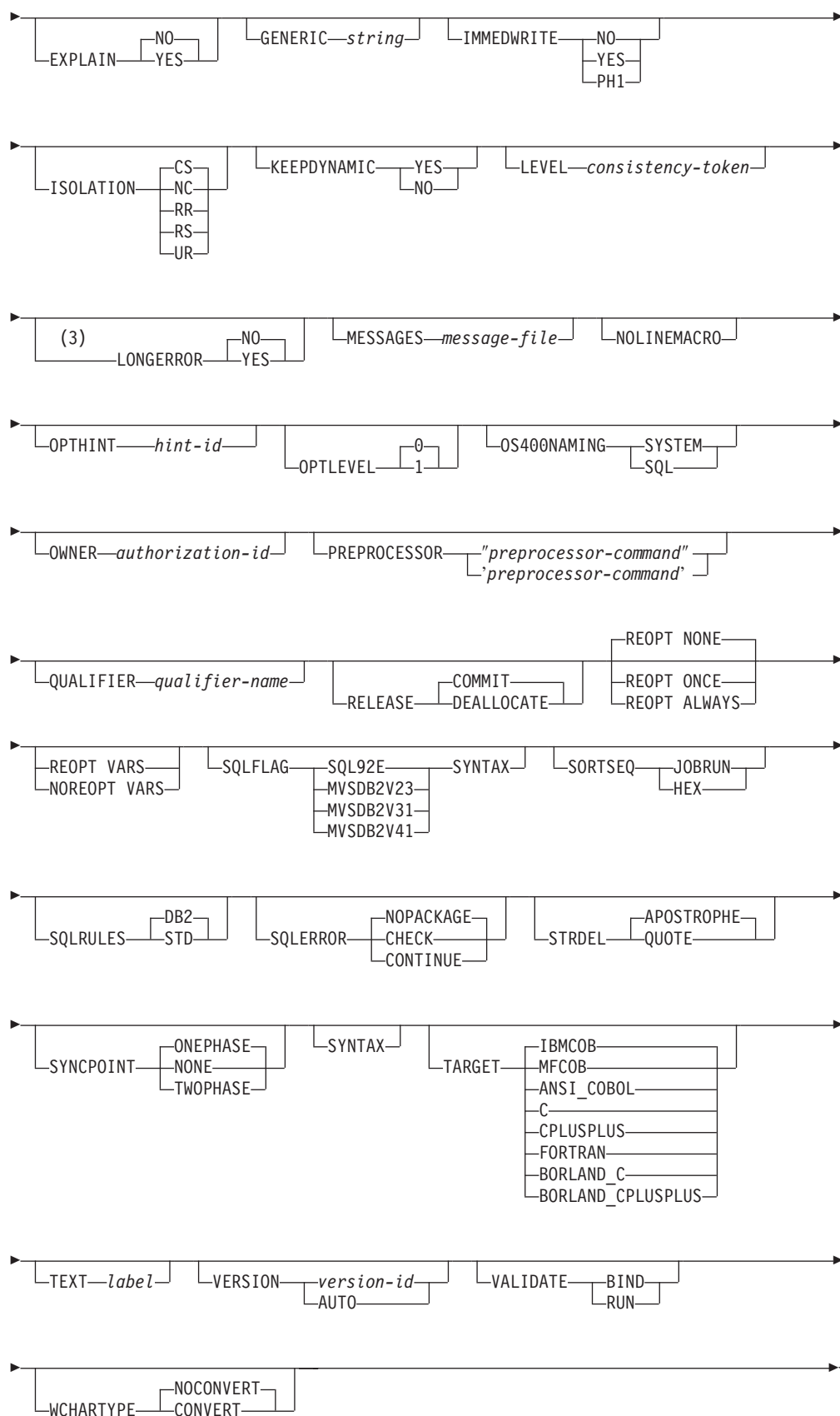
- 1 NO は、32 ビット・システムと 64 ビット NT システムのデフォルトです。それらのシステムでは、長いホスト変数を INTEGER 列の宣言として使用できます。YES は、64 ビット UNIX システムのデフォルトです。
- 2 SYNTAX は SQLERROR(CHECK) の同義語です。

**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。すで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。これがデフォルト値です。

**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すでに存在すれば、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。

あいまいなカーソルは、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。プリコンパイラが CALL\_RESOLUTION IMMEDIATE を指定した CALL ステートメントのプロシージャを解決できなかった場合に、SQL0204 が出されることに注意してください。

### 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。なお、DB2 はこの DRDA プリコンパイラ/ 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。なお、DB2 はこの DRDA プリコンパイラ/ 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。なお、DB2 はこの DRDA プリコンパイラ/ 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 プリコンパイラ/ BIND オプションをサポートしません。

**BIT** 明示的にサブタイプを指定しなかった場合、すべての新規文字カラムに FOR BIT DATA SQL 文字サブタイプが使用されます。

### DEFAULT

明示的にサブタイプを指定しなかった場合、すべての新規文字カラムにターゲット・システムが定義したデフォルト・サブタイプが使用されます。

### MIXED

明示的にサブタイプを指定しなかった場合、すべての新規文字カラムに FOR MIXED DATA SQL 文字サブタイプが使用されます。

**SBCS** 明示的にサブタイプを指定しなかった場合、すべての新規文字カラムに FOR SBCS DATA SQL 文字サブタイプが使用されます。

**CNULREQD**

このオプションは DRDA でサポートされていない **LANGLEVEL** プリコンパイル・オプションと関連します。これは、C または C++ アプリケーションで作成されたバインド・ファイルの場合のみ有効です。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** パート名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

## PRECOMPILE

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** このコマンドが実行されると同時に PREPARE ステートメントが実行されます。

**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。

SQLDA を即時に戻す必要がある INTO 文節を使用している場合、PREPARE ステートメントは据え置かれません。しかし、パラメーター・マーカを使用しないカーソルに対して PREPARE INTO ステートメントが発行される場合、その処理は PREPARE の実行時にカーソルを事前 OPEN することによって最適化されます。

**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 ステートメントで SQLDA を戻すために INTO 文節を使用している場合、OPEN、 DESCRIBE、または EXECUTE ステートメントが発行されて戻されるまで、アプリケーションでこの 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、ランタイムの動的 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。これがデフォルト値です。

**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、ランタイム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。

**注:** 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、増分バインド・ステートメントの場合はランタイムに、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

再最適化可能な増分バインド SQL ステートメントの Explain 情報が、ランタイムに各 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 情報はランタイム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。適格な増分バインド SQL ステートメントの Explain 情報が、ランタイムに各 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 情報はランタイムに適格な動的 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)。

注: 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、増分バインド・ステートメントの場合はランタイムに、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、Explain 表内に入れます。

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

再最適化可能な増分バインド SQL ステートメントの Explain スナップショット情報が、ランタイムに各 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ランタイム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。適格な増分バイン

ンド SQL ステートメントの Explain スナップショット情報が、ランタイムに各 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ランタイムに適格な動的 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ニックネームを参照する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ニックネームを参照する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。

**注:** 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ニックネームを参照するビューは、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ニックネームを参照する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ニックネームを参照するビューは、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ニックネームを参照する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、ユーザー定義の個別タイプおよび機能を解析する際に使用する機能パスを指定します。このオプションを指定しなかった場合、デフォルトの機能パスは "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、機能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、処理結果の機能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。機能パス内に含まれていなければ、最初のスキーマに暗黙的に想定されます。

## INSERT

DB2 Enterprise Server Edition サーバーへプリコンパイルまたはバインドされているプログラムが、パフォーマンス向上のために挿入データをバッファリングすることを要求できるようにします。

**BUF** アプリケーションからの挿入データをバッファリングすることを指定します。

**DEF** アプリケーションからの挿入データをバッファリングしないことを指定します。

### GENERIC string

ターゲット・データベースで定義されていても、DRDA でサポートされていない新規 BIND オプションをサポートします。BIND または PRECOMPILE で定義されている BIND オプションを渡すようにするには、このオプションを使用しないでください。このオプションは、動的 SQL のパフォーマンスをかなり向上させることができます。構文は次のとおりです。

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

各オプションと値は、1 つ以上の空白・スペースで区切らなければなりません。たとえば、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** コミットしません。コミットメント制御が使用されないということ指定します。なお、DB2 はこの分離レベルをサポートしません。

**RR** 反復可能読み取りを分離レベルとして指定します。

**RS** 読み取り固定を分離レベルとして指定します。読み取り固定は、パッケージ内での SQL ステートメントの実行を、他のアプリケーションが読み取りおよび変更を行った行に対する処理から分離させます。

**UR** 非コミット読み取りを分離レベルとして指定します。

### LANGLEVEL

アプリケーション内の静的および動的 SQL の構文およびセマンティクスに適用される SQL 規則を指定します。このオプションは DRDA サーバーではサポートされていません。

**MIA** 以下のように ISO/ANS SQL92 規則を選択します。

- エラー SQLCODE または SQLSTATE の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 (たとえば MIN または MAX) は、NULL も検査し、NULL があったなら警告 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 (たとえば 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 (たとえば MIN または MAX) は、NULL も検査し、NULL があったなら警告 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。

注: 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ー、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。

注: 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。

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、区切り記号としてスラッシュが使用されているために、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ので注意してください。たとえば、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 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。たとえば、

```
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、使用可能な最適化レベルの範囲の詳細を記述します。この DB2 プリコンパイル/ BIND オプションは、DRDA ではサポートされていません。

## RELEASE

リソースを、各 COMMIT ポイントで解放するか、アプリケーションの終了時に解放するかを指示します。なお、DB2 はこの DRDA プリコンパイル/ BIND オプションをサポートしません。

## COMMIT

各コミット点でリソースを解放します。これは、動的 SQL ステートメントに使用されます。

## DEALLOCATE

アプリケーションの終了時にだけリソースを解放します。

**REOPT**

DB2 がホスト変数、パラメーター・マーカ、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、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。

**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ランタイムにアクセス・パスを判別するようにするかどうかを指定します。サポートしているのは DB2 for OS/390 だけです。サポートされているオプション値のリストについては、DB2 for OS/390 の資料を参照してください。

**SQLCA**

FORTRAN アプリケーションでのみ使用します。その他の言語で使用しても、このオプションは無視されます。

**NONE** 修正されたソース・コードは SAA 定義と整合性がないことを指定します。

**SAA** 修正されたソース・コードは SAA 定義と整合性があることを指定します。

**SQLERROR**

エラーを検出した場合に、パッケージまたはバインド・ファイルを作成するかどうかを指示します。

**CHECK**

ターゲット・システムが、バインドされている SQL ステートメントのすべての構文、およびセマンティックの検査を行うことを指定します。この処理の一部としてパッケージが作成されることはありません。バインド中に、名前とバージョンが同じ既存パッケージを検出した場合、その既存パッケージはドロップも置換 (**ACTION REPLACE** を指定した場合) もされません。

**CONTINUE**

SQL ステートメントのバインド時にエラーが発生しても、パッケー



ジを作成します。許可または存在などの理由でバインドに失敗したこれらのステートメントは、**VALIDATE RUN** も指定されている場合は、実行時に増分でバインドすることができます。ランタイムで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** は呼び出されず、バインド・ファイルやパッケージが影響を受けることはありません。

### **SQL92E SYNTAX**

SQL ステートメントは、データベース・カタログにアクセスするのに必要な構文規則を除いて、ANSI または ISO SQL92 の項目レベル SQL 言語の書式および構文と比較して調べられます。相違があれば、プリコンパイラ・リストに報告されます。

### **MVSDDB2V23 SYNTAX**

SQL ステートメントは、MVS DB2 バージョン 2.3 SQL 言語構文と比較して調べられます。構文の相違があれば、プリコンパイラ・リストに報告されます。

### **MVSDDB2V31 SYNTAX**

SQL ステートメントは、MVS DB2 バージョン 3.1 SQL 言語構文と比較して調べられます。構文の相違があれば、プリコンパイラ・リストに報告されます。

### **MVSDDB2V41 SYNTAX**

SQL ステートメントは、MVS DB2 バージョン 4.1 SQL 言語構文と比較して調べられます。構文の相違があれば、プリコンパイラ・リストに報告されます。

### **SORTSEQ**

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指定できます。

**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。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** 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。

注: 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。

**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を使用します。

**APOSTROPHE**

ストリング区切り文字として、アポストロフィ (') を使用します。

**QUOTE**

ストリング区切り文字として、二重引用符 (") を使用します。

**SYNCPPOINT**

複数のデータベース接続にまたがってコミットまたはロールバックを調整する仕方を指定します。

**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、プリコンパイラに指示します。

**IBMCOB**

AIX では、IBM COBOL Set (AIX 版) のコンパイラのためのコードが生成されます。

**MFCOB**

Micro Focus COBOL コンパイラ用のコードが生成されます。すべての UNIX プラットフォームおよび Windows NT の COBOL プリコンパイラにおいて、**target** 値が指定されないなら、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 コンパイラと互換性のあるコードが生成されます。

### CPLUSPLUS

現行プラットフォーム上の DB2 によりサポートされる C++ コンパイラーと互換性のあるコードが生成されます。

### FORTRAN

現行プラットフォーム上の DB2 によりサポートされる FORTRAN コンパイラーと互換性のあるコードが生成されます。

### TEXT label

パッケージの記述。最大長は 255 文字です。また、デフォルトはブランクです。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型の値をホスト・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、構造型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・メッセージ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 バイト以下のバージョン 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。それには SQL ステートメントに相当するホスト言語ステートメントが入っています。パッケージは、デフォルトでは、接続が確立されているデータベース内に作成されます。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。

すで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。

致命的エラーが発生するか、100 以上のエラーが発生すると、プリコンパイルは停止します。致命的エラーが発生すると、ユーティリティーはプリコンパイルを停止し、すべてのファイルをクローズしてからパッケージを廃棄します。

パッケージがバインド動作を公開するとき、以下のとおりとなります。

1. BIND オプション OWNER の暗黙的または明示的な値は、動的 SQL ステートメントの許可検査に使用されます。
2. BIND オプション QUALIFIER の暗黙的または明示的な値は、動的 SQL ステートメント内の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。

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、静的バインドから増分バインドに変更したり、増分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:

- 「アプリケーション開発ガイド クライアント・アプリケーションのプログラミング」の『動的 SQL における許可に関する考慮事項』
- 「アプリケーション開発ガイド クライアント・アプリケーションのプログラミング」の『C および C++ での WCHARTYPE プリコンパイラー・オプション』
- 「アプリケーション開発ガイド クライアント・アプリケーションのプログラミング」の『動的 SQL における DYNAMICRULES BIND オプションの影響』
- 「アプリケーション開発ガイド クライアント・アプリケーションのプログラミング」の『静的 SQL における REOPT の影響』
- 「アプリケーション開発ガイド クライアント・アプリケーションのプログラミング」の『動的 SQL における REOPT の影響』

#### 関連タスク:

- 「管理ガイド: 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:

- 「SQL リファレンス 第 2 巻」の『SET CURRENT QUERY OPTIMIZATION ステートメント』
- 305 ページの『BIND』
- 「SQL リファレンス 第 1 巻」の『日付/時刻の値』

## 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。

### 権限:

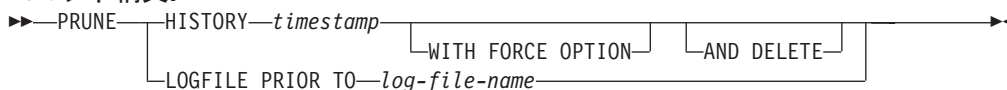
以下のいずれか。

- *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・スペースが確実に確保されるようにする上で、特に有用です。

**注:** 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



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
```

**注:** 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。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削除すると、 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。



という診断テキストが戻されます。"-204" は SQLCODE、"42704" は SQLSTATE で、それぞれ問題の原因を示します。この例の SQLCODE および SQLSTATE は、GET ROUTINE コマンドに指定されたプロシージャ名が未定義であることを示しています。

## QUERY CLIENT

アプリケーションの現在の接続設定を返します。

**権限:**

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:**

アプリケーション処理用の接続設定は、実行中にいつでも照会できます。

**関連資料:**

- 732 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をすべて強制的にオフにして、静止モードにします。静止モード中は、ユーザーはデータベース・エンジンの外部から接続できません。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。

### 権限:

以下のどれかが必要です。

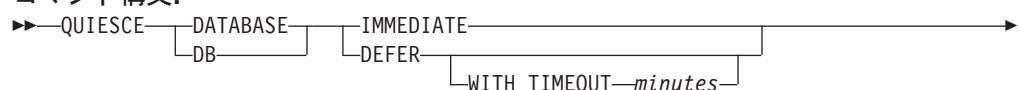
データベース・レベルの静止の場合:

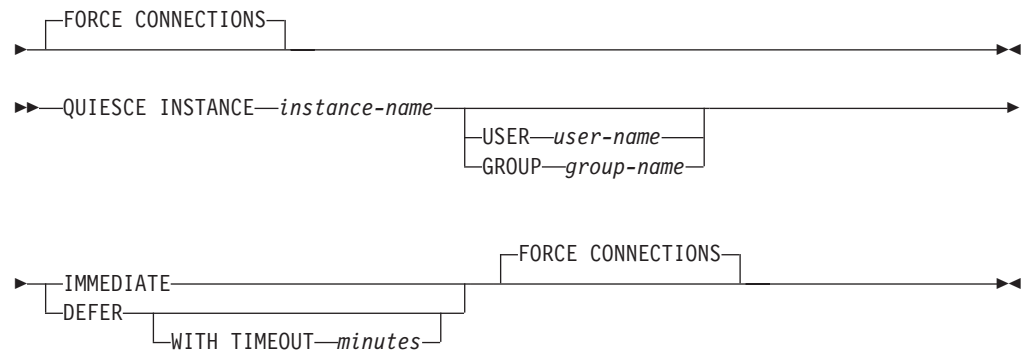
- *sysadm*
- *dbadm*

インスタンス・レベルの静止の場合:

- *sysadm*
- *sysctrl*

### コマンド構文:





**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・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、`start_stop_timeout` データベース・マネージャー構成パラメーターに指定されている値が使用されます。

**IMMEDIATE**

トランザクションがコミットされるのを待たず、即時にトランザクションをロールバックします。

**FORCE CONNECTIONS**

接続を強制的にオフにします。

**DATABASE**

データベースを静止します。データベース内のすべてのオブジェクトを静止モードにします。指定したグループのユーザーと、`sysadm`、`sysmaint`、および `sysctrl` 権限を持つユーザーだけが、データベースまたはそのオブジェクトにアクセスすることができます。

**INSTANCE** *instance-name*

インスタンス *instance-name*、およびインスタンス内のデータベースを静止モードにします。インスタンスにアクセスできるのは、`sysadm`、`sysmaint`、および `sysctrl` 権限を持つユーザーと、指定したグループのユーザーだけです。

**USER** *user-name*

インスタンスの静止中にそのインスタンスへのアクセスが許可されているユーザーの名前を指定します。

**GROUP** *group-name*

インスタンスの静止中にそのインスタンスへのアクセスが許可されているグループの名前を指定します。

**例:**

以下に示すのは、接続の強制がデフォルトの動作であるため、明示的な説明は必要なく、この例から省くことができる例です。

```
db2 quiesce instance crankarm user frank immediate force connections
```

以下の例は、データベースに接続しているすべてのユーザーを、強制的に切断にします。

```
db2 quiesce db immediate
```

- 最初の例は、インスタンス `crankarm` を静止し、ユーザー `frank` は引き続きデータベースを使用できるようにします。

2 番目の例は、アタッチしているデータベースを静止し、以下のいずれかの権限をもつユーザー以外のユーザーは、すべてアクセスできないようにします。

`sysadm`、`sysmaint`、`sysctrl` または `dbadm`

- このコマンドとともに **FORCE 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をすべて強制的にオフにします。**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。

たとえば、以下のようになります。

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

## QUIESCE TABLESPACES FOR TABLE

表の表スペースを静止させます。共用、更新意図、排他の 3 つの有効な静止モードがあります。静止機能の結果として生じる状態には、次の 3 つの状態、QUIESCED SHARE、QUIESCED UPDATE、および QUIESCED EXCLUSIVE があります。

### 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、このコマンドはノード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ノードに属する表スペースの部分のみを静止します。

### 権限:

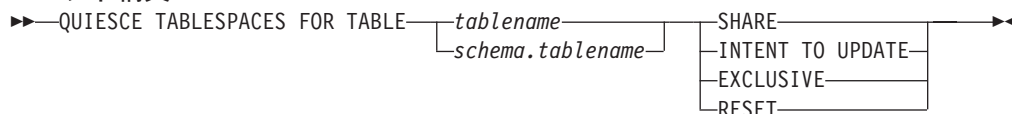
以下のいずれかが必要です。

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

### 必要な接続:

データベース

### コマンド構文:



### コマンド・パラメーター:

#### TABLE

##### tablename

非修飾表名を指定します。システム・カタログ表を指定することはできません。

##### schema.tablename

修飾表名を指定します。 *schema* が指定されない場合には、CURRENT SCHEMA が使用されます。システム・カタログ表を指定することはできません。

#### SHARE

静止が共用モードであることを指定します。

『静止モードでの共用』 要求を行うと、トランザクションは、表スペースに対して専用共用ロックを、および表に対して共用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。他のユーザーによって保留されているような対立状態がない場合には、その状態は、そのユーザーにのみ付与されます。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



一のデータベース・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。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、ロックは解放されますが、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、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。表スペースは、特別な排他モードで保護されているため、その表スペースへのアクセスが認められているその他のアクセスはありません。静止プログラム機能呼び出すユーザー (静止者) 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。

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。たとえば、削除フェーズ中に停電によってロード操作が割り込まれると、ロードされていた表の表スペースは削除ペンディング、静止排他状態で残ります。データベースの再始動時に、この静止は非所有 (ファントム) の状態になります。ファントム静止を取り除くには、静止モードが設定されたときに使用されたのと同じユーザー ID による接続が必要です。

ファントム静止を取り除くには、次のようにします。

1. 静止モードが設定されたときに使用されたのと同じユーザー ID でデータベースと接続する。
2. LIST TABLESPACES コマンドを使用して、静止させる表スペースを決定する。

## QUIESCE TABLESPACES FOR TABLE

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。たとえば、次のようにします。

```
db2 quiesce tablespaces for table mytable exclusive
```

完了すると、新しい接続が静止を所有するようになり、ロード操作を再開できるようになります。

いつでも、表スペース上での静止者の限度は 5 つです。

静止者は表スペースの状態を、制限の少ない状態から、より制限のある状態（たとえば、S から U へ、または U から X へ）へアップグレードすることができます。ユーザーがすでに保持されている状態より低い状態を要求すると、元の状態に戻されます。状態は、格下げされません。

### 関連資料:

- 558 ページの『LOAD』

---

## 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。

### 権限:

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:

- 761 ページの『TERMINATE』

## REBIND

ユーザーが、バインド・ファイルを必要とせずに、データベースに格納されているパッケージを、再作成することができます。

### 権限:

以下のいずれかが必要です。

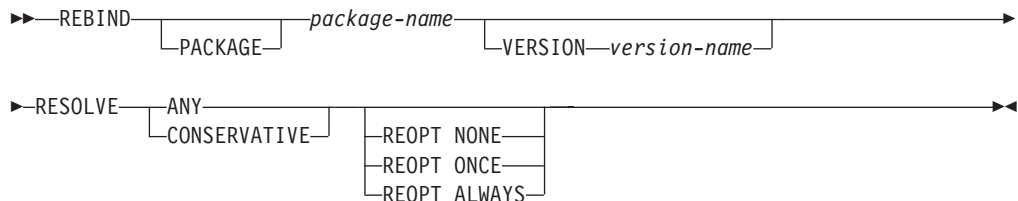
- *sysadm* または *dbadm* の権限
- スキーマに対する ALTERIN 特権
- パッケージに対する BIND 特権

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 と同じでなくてもよいことに注意してください。REBIND は、パッケージが作成された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、暗黙の接続が使用可能な場合には、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、古典的なバインド手法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。古典的なバインド手法は使用されません。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。古典的なバインド手法を使用します。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:

REBIND は、正常な再バインドに続いて、トランザクションを自動的にコミットしません。ユーザーは、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。これにより、ユーザーがある統計を更新する場合の、「what if」分析を使用可能にして、変更した内容を見るために、パッケージを再バインドしようとします。また、作業単位内の複数の再バインドも許可しています。

**注:** REBIND コマンドは、自動コミットが使用可能な場合には、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。

- パッケージを再作成する簡易的な方法を提供します。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。たとえば、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する方法を提供します。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して、明示的に再バインドする必要があります。パッケージが依存する機能

インスタンスがドロップされると、パッケージは作動不能とマークされます (SYSCAT.PACKAGES システム・カタログの VALID 列は、X と設定されます。)

- ユーザーに無効パッケージの再バインド以上の制御を与えます。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、最初の遅延を無くし、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。たとえば、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、1 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 のどちらを使用するかは、その環境によります。REBIND のパフォーマンスが BIND のパフォーマンスよりかなり良いため、BIND の使用を明示的に必要としない場合はいつでも、REBIND を使用するように推奨されています。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。

- プログラムを修正した場合 (たとえば、SQL 言語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)。
- ユーザーが再バインドの部分として BIND オプションのいずれかを修正する場合。REBIND は、BIND オプションをサポートしません。たとえば、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。
- パッケージが現在データベースに存在しない場合。
- 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 がエラーを検出すると、処理を停止してエラー・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 コマンド要求者のものであり、最初のバインド・コマンド送出者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増分バインドに変更したり、増分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:

- 305 ページの『BIND』
- 720 ページの『RUNSTATS』
- 201 ページの『db2rbind - すべての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。

### 権限:

以下のいずれかが必要です。

- *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

調整が実行される表を指定します。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たファイルに関する情報が入る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 値が関係したリンク障害が発生した行のコピー先とな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、ユーザーがすで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、存在するファイルのリンクは表データに従って試みられますが、存在しないファイルのリンクはデータ・リンク・ファイル・マネージャーのメタデータに従って試みられます (他の矛盾が存在しなければ)。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。これらの違反行は入力表から削除されません。ファイル参照の保全性を保証するため、問題となっている DATALINK 値は NULL にされます。列が NULL 不可として定義されている場合には、DATALINK 値が長さゼロの URL によって置換されます。

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 が使用可能でなかったり、DROP DATALINKS MANAGER コマンドで 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。たとえば、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	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 - ファイルはすで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 - ファイルはすで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:

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。

#### 関連概念:

- 「*DB2 Data Links Manager 管理ガイド*および*リファレンス*」の『障害とリカバリーの概要』

## 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パーティションに影響を与えます。

### 権限:

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・アタッチが必要です。

### コマンド構文:

```

➤ RECOVER DATABASE DB source-database-alias

```

```

TO isotime USING LOCAL TIME
END OF LOGS USING GMT TIME ON ALL DBPARTITIONNUMS
On Database Partition clause

```

```

USER username USING password OPEN num-sessions SESSIONS

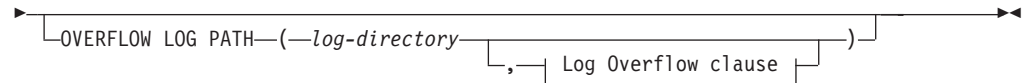
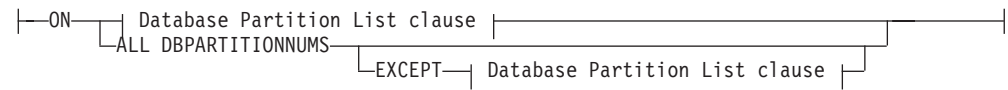
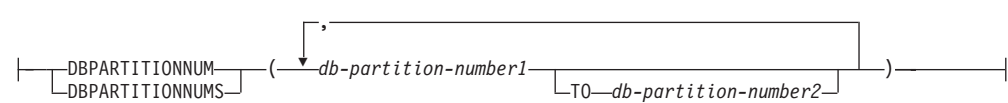
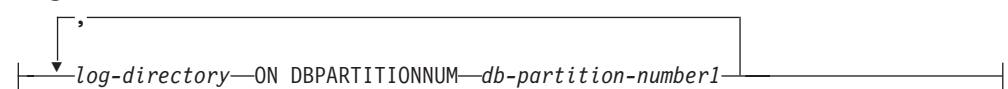
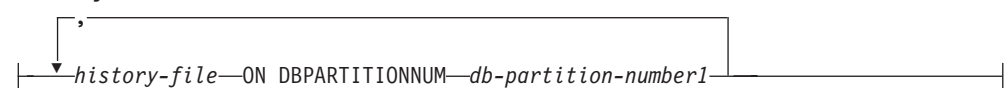
```

```

USING HISTORY FILE ( history-file )
History File clause

```

**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がリカバリーされる時点 (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、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)。

この値は、日付と時刻の組み合わせを識別する 7 つの部分の文字列からなる、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) です。UTC は、さまざまなログに関連したタイム・スタンプが (たとえば、夏時間に関連して時間の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

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 GMT 時間ではなくユーザーの現地時間を使用して特定の時点にリカバリーできます。これによって、ユーザーがローカル・マシンで特定の時点にリカバリーすることが容易になり、現地時間を GMT ポイント・イン・タイムに変換することによって生じる潜在的なユーザー・エラーの発生を除去します。

#### 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ることに注意してください。
2. タイム・スタンプ・ストリングはサーバー上で GMT 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 GMT 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

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。

**OPEN** *num-sessions* **SESSIONS**

Tivoli Storage Manager (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。

**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。

**注:** 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* で利用可能である場合、

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、使用可能な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。たとえば、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、ユーザーはその時



## RECOVER DATABASE

刻からの履歴ファイルのバージョンを提供できます。ユーザーがその時刻から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 (たとえば `/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。)

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` に利用可能な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したい場合があるかもしれません。特に指定しない場合、各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（たとえば /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（たとえば、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（たとえば、それ以上テープがない場合）。
  - t** 終了。すべての装置を終了します。
- リカバリー操作のリストア部分において障害が発生した場合は、RECOVER DATABASE コマンドを再発行することができます。リストア操作が成功したが、ロールフォワード操作中にエラーが発生した場合は、リカバリー操作全体をやり直す必要はなく（それには非常に時間がかかる）、ROLLFORWARD DATABASE コマンドを発行することができます。
- パーティション・データベース環境で、リカバリー操作のリストア部分でエラーが発生した場合、単一データベース・パーティションでのエラーでしかないという可能性があります。RECOVER DATABASE コマンドを再発行するとすべてのデータベース・パーティションでデータベースがリストアされますが、その代わりに、障害が発生したデータベース・パーティションに関する RESTORE DATABASE を発行してから、ROLLFORWARD DATABASE コマンドを発行するほうが効率的です。

## REDISTRIBUTE DATABASE PARTITION GROUP

データベース・パーティション・グループ内のデータベース・パーティション間でデータを再分散します。均一なあるいは偏った現行のデータ分散を指定できます。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:

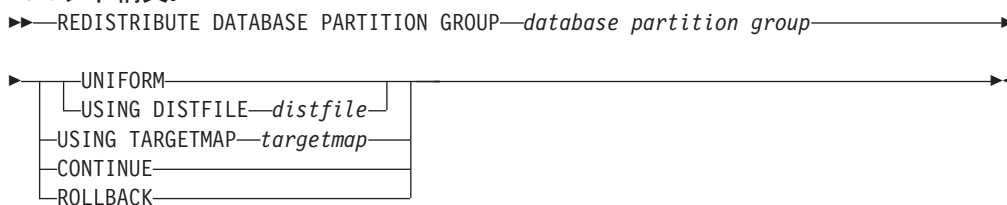
このコマンドは、データベース・パーティション・グループ内のすべてのデータベース・パーティションに影響を与えます。

### 権限:

以下のいずれかが必要です。

- `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 部分名は、SYSCAT.DBPARTITIONGROUPS カタログ表に記述されたデータベース・パーティション・グループを識別します。データベース・パーティション・グループは、現在再配布を受けることはできません。

**注:** IBMCATGROUP および IBMTEMPGROUP データベース・パーティション・グループ内の表を再分散することはできません。

#### **UNIFORM**

データがハッシュ区分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区分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区分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区分を持っています。

**USING DISTFILE distfile**

パーティション・キー値の分散に偏りがある場合、このオプションを使用して、データベース・パーティション・グループのデータベース・パーティション全体にわたるデータに均一な再分散を行います。

*distfile* を使用して、4 096 個のハッシュ区分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、各ハッシュ区分で表示されたデータ量を示します。ユーティリティーは、パーティションに関連する整数値をそのパーティションの重みとして読み取ります。*distfile* を指定した場合、ユーティリティーはターゲット・パーティション・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パーティションの重みの合計です)。

たとえば、入力配布ファイルに以下の項目があるとしします。

```
10223
1345
112000
0
100
...
```

例の中で、ハッシュ区分 2 は 112 000 の重みを持ち、区分 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・パーティション・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:

再分散操作が行われると、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、/sqlllib/redist ディレクトリー。サブディレクトリーとファイル名については、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、¥sqlllib¥redist¥ ディレクトリー。サブディレクトリーとファイル名については、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、処理の途中に断続的 COMMIT を実行します。

データベース・パーティション・グループにデータベース・パーティションを追加するには、ALTER DATABASE PARTITION GROUP ステートメントを使用します。このステートメントは、データベース・パーティション・グループに対応した表スペース用のコンテナを定義できるようにします。

**注:** 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されます。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、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。明示の再バインドによって、無効パッケージの最初の SQL 要求の実行を初期遅延させなくします。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っています。

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。

ユーザー TEMPORARY 表スペースと既存の宣言済み一時表とがデータベース・パーティション・グループに存在する場合、再分散は許可されません。

### 互換性:

## REDISTRIBUTE DATABASE PARTITION GROUP

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。

関連資料:

- 520 ページの『LIST DATABASE DIRECTORY』
- 720 ページの『RUNSTATS』
- 642 ページの『REBIND』

---

## 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。

**権限:**

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。

**注:** このパラメーターは AIX または Solaris オペレーティング環境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:**

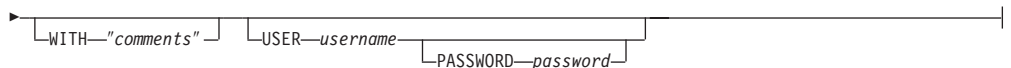
最新表示中に LDAP のオブジェクトが除去されると、それに対応するローカル・マシンの LDAP 項目も除去されます。LDAP の情報が変更されると、それに対応する LDAP 項目もそれに応じて変更されます。DB2CLI.INI ファイルを手動で更新する場合は、REFRESH LDAP CLI CFG コマンドを実行して、現行ユーザーのキャッシュを更新してください。

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。

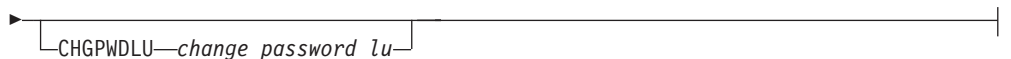
权限:

必要な接続:

コマンド構文:



**APPN path:**



コマンド・パラメーター:

**IN** DB2 サーバーを登録するネットワーク・ディレクトリー・サーバーを指定します。有効な値は、LDAP (Lightweight Directory Access Protocol) ディレクトリー・サーバーの場合、LDAP です。

## ADMIN

管理サーバー・ノードを登録することを指定します。

**NODE/AS** **nodename**

LDAP 内の DB2 サーバーを表すショート・ネームを指定します。ノード項目は、このノード名によって LDAP にカタログされます。クライアント

は、このノード名によってサーバーにアタッチできます。この LDAP ノード項目に関連付けられるプロトコルは、**PROTOCOL** パラメーターで指定します。

#### **NODEnodename**

LDAP 内の DB2 サーバーを表すショート・ネームを指定します。ノード項目は、このノード名によって LDAP にカタログされます。クライアントは、このノード名によってサーバーにアタッチできます。この LDAP ノード項目に関連付けられるプロトコルは、**PROTOCOL** パラメーターで指定します。

#### **PROTOCOL**

LDAP ノード項目に関連付けるプロトコル・タイプを指定します。データベース・サーバーは複数のプロトコル・タイプをサポートできるため、この値にはクライアント・アプリケーションが実際に使用するプロトコル・タイプを指定します。DB2 サーバーはプロトコルごとに 1 つずつ登録しなければなりません。有効な値は、TCP/IP、NETBIOS、APPN、および NPIPE です。Windows の名前付きパイプを使用する場合は、一番あとの値を指定してください。このプロトコル・タイプをサポートするのは、Windows オペレーティング・システムで稼働する DB2 サーバーだけです。

注: NETBIOS および NPIPE は、AIX および Solaris オペレーティング・システム上ではサポートされていません。ただし、これらのプロトコルは、Windows NT などのオペレーティング・システムを使用するリモート・サーバーに対しては登録できます。

#### **HOSTNAME hostname**

TCP/IP ホスト名 (または IP アドレス) を指定します。

#### **SVCENAME svcename**

TCP/IP サービス名またはポート番号を指定します。

#### **SECURITY SOCKS**

使用する TCP/IP ソケット・セキュリティを指定します。

#### **NNAME nname**

NetBIOS ワークステーション名を指定します。

#### **NETWORK net\_id**

APPN ネットワーク ID を指定します。

#### **PARTNERLU partner\_lu**

DB2 サーバー・マシン用の APPN パートナー LU 名を指定します。

#### **MODE mode**

APPN モード名を指定します。

#### **TPNAME tpname**

APPN トランザクション・プログラム名を指定します。デフォルトは DB2DRDA です。

#### **SECURITY**

APPN セキュリティ・レベルを指定します。有効な値は以下のとおりです。

**NONE** サーバーに送信する割り振り要求に、セキュリティ情報が含まれ



ないということを指定します。これは、DB2 UDB サーバーの場合のデフォルト・セキュリティです。

**SAME** サーバーに送信する割り振り要求に、ユーザー名が含まれないということを指定します。これは、ユーザー名が「すでに検査済み」という標識で指定されます。サーバーは、「すでに検査済み」という保証を受け入れられるように構成されていなければなりません。

#### PROGRAM

サーバーに送信する割り振り要求に、ユーザー名とパスワードの両方が含まれるということを指定します。これは、DB2 for OS/390 and z/OS、DB2 for iSeries などのホスト・データベース・サーバーの場合のデフォルト・セキュリティです。

#### LANADDRESS lan\_address

APPN ネットワーク・アダプター・アドレスを指定します。

#### CHGPWDLU change\_password\_lu

ホスト・データベース・サーバーのパスワード変更時に使用される、パートナー LU の名前を指定します。

#### 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 UDB Enterprise Edition サーバーの場合に、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 UDB Enterprise Extended Edition (パーティション・データベース) サーバーの場合に、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場合に、DCS ノード・タイプを指定します。これを指定すると、クライアントまたはゲートウェイはデータベース・プロトコルとして DRDA を使用します。

#### 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。有効な値は次のとおりです。AIX、NT、HPUX、SUN、MVS、OS400、VM、VSE、SNI、SCO、LINUX。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

ディング・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 サーバーを記述します。ネットワーク・ディレクトリーに登録されるサーバー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。たとえば、DB2 サーバーが NetBIOS と TCP/IP の両方をサポートする場合、REGISTER コマンドは次のように 2 回呼び出す必要があります。

```
db2 register db2 server in ldap as tcpnode protocol tcpip
db2 register db2 server in ldap as nbnode protocol netbios
```

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何らかのプロトコル構成パラメーターを指定すると、データベース・マネージャー構成ファイルに指定された値はオーバーライドされます。

APPN の場合、データベース・マネージャー構成ファイルには TPNAME だけがあります。APPN を正しく登録するためには、必要パラメーター (NETWORK、PARTNERLU、MODE、TPNAME、および SECURITY) に値を指定しなければなりません。オプション・パラメーター (LANADDRESS および CHGPWDLU) に値を指定することもできます。

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。

### 関連資料:

- 372 ページの『DEREGISTER』
- 796 ページの『UPDATE LDAP NODE』

## REORG INDEXES/TABLE

索引または表を再編成します。

索引オプションは、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します。索引オプションの **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実行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。

### 権限:

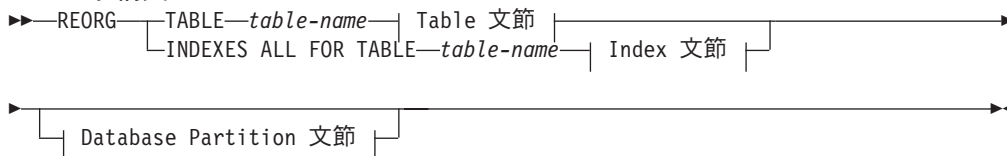
以下のどれかが必要です。

- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*
- 表に対する **CONTROL** 特権

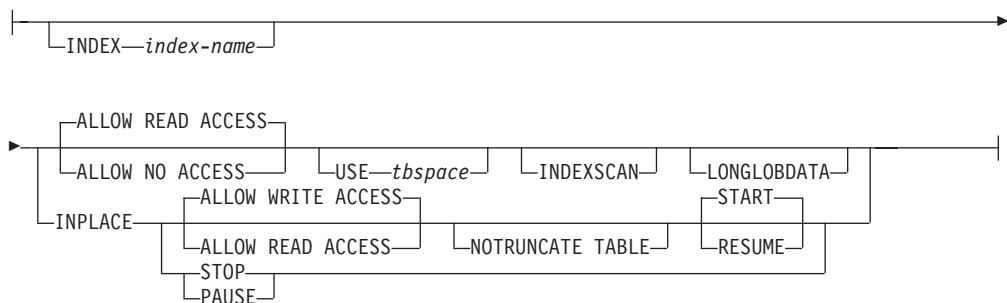
### 必要な接続:

データベース

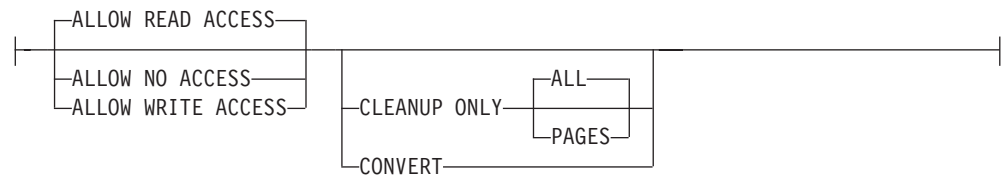
### コマンド構文:



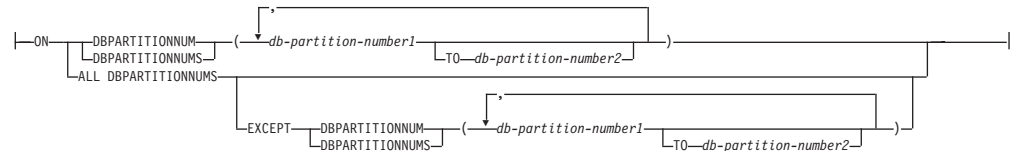
### Table 文節:



### 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。

#### 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。これがデフォルトです。

#### 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

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。索引がすでにタイプ 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、すでにタイプ 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が使用可能な期間は、表へのアクセスが許可されないことに注意してください。

### 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。名前または別名の書式は、`schema.table-name` 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。

**注:** 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。さらに、表のインプレース再編成は MDC 表に対して使用できないことに注意してください。

#### INDEX index-name

表を再編成する際に使用する索引を指定します。書式に完全修飾の名前 *schema.index-name* 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いて、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ていない場合、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ければ、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。

#### INPLACE

ユーザー・アクセスを許可しながら、表を再編成します。

表のインプレース再編成が可能なのは、タイプ 2 の索引があり拡張索引がない表だけです。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。

#### USE tablespace-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・サイズは、表データ (LONG または LOB 列データを含む) が存在する表スペースのページ・サイズと一致し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。

例:

DB2 バージョン 7 でのデフォルトのように REORG TABLE をクラス分けするには、以下のコマンドを入力します。

```
db2 reorg table employee index empid allow no access indexscan
longlobdata
```

DB2 バージョン 8 でのデフォルトは異なることに注意してください。

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 temp space1
```

デフォルト・スキーマ HOMER の指定された 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、読み取り専用アクセス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ることに注意してください。

#### 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 表がある表スペースのオンライン・バックアップが実行されている間は、REORG TABLE を DMS 表に使用できません。
- 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ンのパフォーマンスを低下させ、索引ページの事前取り出しの効果に影響を与えます。REORG INDEXES を使用して、表のすべての索引の再編成、フラグメントの除去、およびリーフ・ページの物理クラスターリングのリストアを行うことができます。REORGCHK を使用すると、索引に再編成が必要かどうかを判断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、REORG INDEXES 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で、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。構造型列のインラインの長さを変更後、このユーティリティもまた呼び出して、変更の有用性を確認してください。REORGCHK を使用して、表の再編成が必要かどうかを判断してください。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・パーティションに区分されている場合、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の再編成がロールバックされます。

**注:** 再編成が成功しなかった場合には、一時ファイルを削除すべきではありません。データベース・マネージャーは、これらのファイルを使用し、データベースをリカバリーします。

索引の名前が指定されると、データベース・マネージャーはその索引の順番に従って、データを再編成します。パフォーマンスを最善にするため、SQL 照会で頻繁に使用される索引を指定してください。索引の名前が指定されない場合、および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順序付けられます。

表の PCTFREE 値は、ページごとに指定されたフリー・スペース量を決定するものとなります。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で可能なかぎり多くのスペースを満たします。

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・リカバリーが使用可能で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あります。これは、行が再編成された順序、および使用される (SMS または DMS) 表スペースのタイプの結果になります。

**関連資料:**

- 452 ページの『GET SNAPSHOT』
- 674 ページの『REORGCHK』
- 720 ページの『RUNSTATS』
- 642 ページの『REBIND』
- 「*SQL Administrative Routines*」の『SNAPSHOT\_TBREORG 表関数』

## 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。

### 権限:

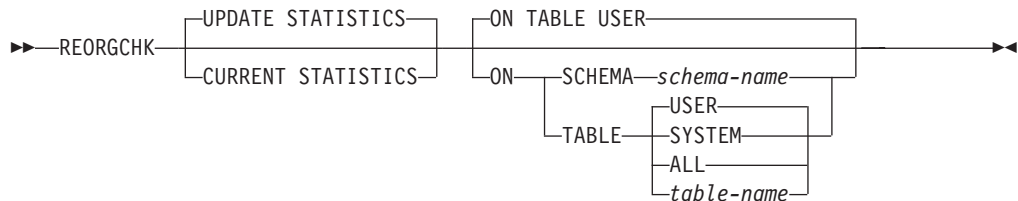
以下のどれかが必要です。

- 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、表の統計を更新してから、更新された統計を使用して、表の再編成が必要であるか判別します。

REORGCHK を発行したデータベース・パーティションに表パーティションが存在する場合、このデータベース・パーティションで RUNSTATS を実行します。このデータベース・パーティションに表パーティションが存在しない場合、その要求は表のパーティションを保留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。次にそのデータベース・パーティションで RUNSTATS を実行します。

#### CURRENT STATISTICS

現在の表の統計を使用して、表の再編成が必要であるか判別します。

#### 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。

**注:** 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:**

以下に示すのは、次のコマンドの出力例です。 *Index Statistics* 出力の一部だけが示されています。

```
db2 reorgchk update statistics on table system
```

SAMPLE データベースに対して実行すると、

```
Doing RUNSTATS ....
```

```
Table statistics:
```

```
F1: 100 * OVERFLOW / CARD < 5
```

```
F2: 100 * (Effective Space Utilization of Data Pages) > 70
```

```
F3: 100 * (Required Pages / Total Pages) > 80
```

# REORGCHK

SCHEMA	NAME	CARD	OV	NP	FP	ACTBLK	TSIZE	F1	F2	F3	REORG
SYSIBM	SYSATTRIBUTES	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSBUFFERPOOLNODES	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSBUFFERPOOLS	1	0	1	1	-	52	0	-	100	----
SYSIBM	SYSCHECKS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSCODEPROPERTIES	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSCOLAUTH	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSCOLCHECKS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSCOLDIST	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSCOLGROUPDIST	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSCOLGROUPDISTCO>	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSCOLGROUPS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSCOLGROUPSCOLS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSCOLOPTIONS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSCOLPROPERTIES	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSCOLUMNS	2812	34	141	143	-	553964	1	95	98	----
SYSIBM	SYSCOLUSE	4	0	1	1	-	156	0	-	100	----
SYSIBM	SYS COMMENTS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYS CONSTDEP	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYS DATATYPES	17	0	3	3	-	14399	0	100	100	----
SYSIBM	SYS DBAUTH	2	0	1	1	-	92	0	-	100	----
SYSIBM	SYS DEPENDENCIES	6	0	1	1	-	468	0	-	100	----
SYSIBM	SYS EVENTMONITORS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYS EVENTS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYS EVENTTABLES	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYS FUNCMAPOPTIONS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYS FUNCMAPPARMOPT>	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYS FUNC MAPPINGS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYS HIERARCHIES	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYS INDEXAUTH	6	0	1	1	-	408	0	-	100	----
SYSIBM	SYS INDEXCOLUSE	506	0	7	7	-	23782	0	97	100	----
SYSIBM	SYS INDEXES	188	89	17	31	-	159988	47	100	54	***
SYSIBM	SYS INDEXEXPLOITRU>	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYS INDEXEXTENSION>	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYS INDEXEXTENSION>	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYS INDEXEXTENSIONS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYS INDEXOPTIONS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYS JARCONTENTS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYS JAROBJECTS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYS KEYCOLUSE	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYS LIBRARIES	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYS LIBRARYAUTH	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYS LIBRARYBINDFIL>	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYS LIBRARYVERSIONS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYS NAME MAPPINGS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYS NODEGROUPDEF	2	0	1	1	-	60	0	-	100	----
SYSIBM	SYS NODEGROUPS	3	0	1	1	-	174	0	-	100	----
SYSIBM	SYS PARTITIONMAPS	4	0	1	1	-	160	0	-	100	----
SYSIBM	SYS PASSTHRUAUTH	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYS PLAN	81	0	7	7	-	66987	0	100	100	----
SYSIBM	SYS PLANAUTH	159	0	3	3	-	9858	0	100	100	----

SYSIBM	SYSPLANDEP	112	0	3	3	-	20720	0	100	100	---
SYSIBM	SYSPREDICATESPECS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSPROCOPTIONS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSPROCPARMOPTIONS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSRELS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSROUTINEAUTH	58	0	2	2	-	5046	0	100	100	---
SYSIBM	SYSROUTINEPARMS	1077	0	42	42	-	161550	0	96	100	---
SYSIBM	SYSROUTINEPROPERTY>	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSROUTINES	195	0	21	21	-	134160	0	100	100	---
SYSIBM	SYSSCHEMAAUTH	2	0	1	1	-	100	0	-	100	---
SYSIBM	SYSSCHEMATA	7	0	1	1	-	427	0	-	100	---
SYSIBM	SYSSECTION	133	0	5	5	-	55195	0	100	100	---
SYSIBM	SYSSEQUENCEAUTH	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSSEQUENCES	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSSEVEROPTIONS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSSEVERERS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSSTMT	133	0	5	5	-	51205	0	100	100	---
SYSIBM	SYSTABAUTH	242	0	5	5	-	17666	0	100	100	---
SYSIBM	SYSTABCONST	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSTABLES	243	0	27	27	-	388071	0	100	100	---
SYSIBM	SYSTABLESPACES	3	0	1	1	-	321	0	-	100	---
SYSIBM	SYSTABOPTIONS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSTBSPACEAUTH	1	0	1	1	-	54	0	-	100	---
SYSIBM	SYSTRANSFORMS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSTRIGGERS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSTYPEMAPPINGS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSUSERAUTH	242	0	8	8	-	52998	0	100	100	---
SYSIBM	SYSUSEROPTIONS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSVERSIONS	1	0	1	1	-	36	0	-	100	---
SYSIBM	SYSVIEWDEP	214	0	5	5	-	18404	0	100	100	---
SYSIBM	SYSVIEWS	144	0	7	7	-	37872	0	100	100	---
SYSIBM	SYSWRAPOPTIONS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSWRAPPERS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSXMLOBJECTAUTH	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSXMLOBJECTAUTHP>	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSXMLOBJECTPROPE>	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSXMLOBJECTRELDEP	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSXMLOBJECTS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSXMLOBJECTXMLDEP	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSXMLPHYSICALCOL>	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSXMLQUERIES	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSXMLRELATIONSHI>	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSXMLRSPROPERTIES	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSXMLSTATS	-	-	-	-	-	-	-	-	-	----

## Index statistics:

F4: CLUSTERRATIO or normalized CLUSTERFACTOR > 80  
 F5:  $100 * (KEYS * (ISIZE + 9) + (CARD - KEYS) * 5) / ((NLEAF - NUM\_EMPTY\_LEAFS) * INDEXPAGESIZE) > 50$   
 F6:  $(100 - PCTFREE) * ((INDEXPAGESIZE - 96) / (ISIZE + 12)) ** (NLEVELS - 2) * (INDEXPAGESIZE - 96) / (KEYS * (ISIZE + 9) + (CARD - KEYS) * 5) < 100$   
 F7:  $100 * (NUMRIDS DELETED / (NUMRIDS DELETED + CARD)) < 20$   
 F8:  $100 * (NUM\_EMPTY\_LEAFS / NLEAF) < 20$

# REORGCHK

SCHEMA	NAME	CARD	LEAF	ELEAF	LVLS	ISIZE	NDEL	KEYS	F4	F5	F6	F7	F8	REORG
-----														
Table: SYSIBM.SYSATTRIBUTES														
SYSIBM	IBM83	-	-	-	-	-	-	-	-	-	-	-	-	-----
SYSIBM	IBM84	-	-	-	-	-	-	-	-	-	-	-	-	-----
SYSIBM	IBM85	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSBUFFERPOOLNODES														
SYSIBM	IBM69	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSBUFFERPOOLS														
SYSIBM	IBM67	1	1	0	1	22	0	1	100	-	-	0	0	-----
SYSIBM	IBM68	1	1	0	1	10	0	1	100	-	-	0	0	-----
Table: SYSIBM.SYSCHECKS														
SYSIBM	IBM37	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSCODEPROPERTIES														
SYSIBM	IBM161	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSCOLAUTH														
SYSIBM	IBM42	-	-	-	-	-	-	-	-	-	-	-	-	-----
SYSIBM	IBM43	-	-	-	-	-	-	-	-	-	-	-	-	-----
SYSIBM	IBM64	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSCOLCHECKS														
SYSIBM	IBM38	-	-	-	-	-	-	-	-	-	-	-	-	-----
SYSIBM	IBM39	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSCOLDIST														
SYSIBM	IBM46	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSCOLGROUPDIST														
SYSIBM	IBM157	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSCOLGROUPDISTCOUNTS														
SYSIBM	IBM158	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSCOLGROUPS														
SYSIBM	IBM154	-	-	-	-	-	-	-	-	-	-	-	-	-----
SYSIBM	IBM155	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSCOLGROUPSCOLS														
SYSIBM	IBM156	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSCOLOPTIONS														
SYSIBM	IBM89	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSCOLPROPERTIES														
SYSIBM	IBM79	-	-	-	-	-	-	-	-	-	-	-	-	-----
SYSIBM	IBM80	-	-	-	-	-	-	-	-	-	-	-	-	-----
SYSIBM	IBM82	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSCOLUMNS														
SYSIBM	IBM01	2812	59	0	2	41	12	2812	95	58	2	0	0	-----
SYSIBM	IBM24	2812	7	0	2	23	3	11	82	50	25	0	0	-*---
Table: SYSIBM.SYSCOLUSE														
SYSIBM	IBM146	4	1	0	1	22	1	4	100	-	-	20	0	----*-
SYSIBM	IBM147	4	1	0	1	23	1	4	100	-	-	20	0	----*-
Table: SYSIBM.SYSCOMMENTS														
SYSIBM	IBM73	-	-	-	-	-	-	-	-	-	-	-	-	-----



```

Table: SYSIBM.SYSCONSTDEP
SYSIBM  IBM44          -    -    -    -    -    -    -    -    -    -    -    -    -    -
SYSIBM  IBM45          -    -    -    -    -    -    -    -    -    -    -    -    -    -
Table: SYSIBM.SYSDATATYPES
SYSIBM  IBM40          17    1    0    1    23    0    17 100    -    -    0    0    -
SYSIBM  IBM41          17    1    0    1    2    0    17 100    -    -    0    0    -
SYSIBM  IBM56          17    1    0    1    23    0    17 100    -    -    0    0    -
Table: SYSIBM.SYSDBAUTH
SYSIBM  IBM12          2     1    0    1    25    0     2 100    -    -    0    0    -
Table: SYSIBM.SYSDEPENDENCIES
SYSIBM  IBM51          6     1    0    1    66    0     6 100    -    -    0    0    -
SYSIBM  IBM52          6     1    0    1    25    0     5 100    -    -    0    0    -
Table: SYSIBM.SYSEVENTMONITORS
SYSIBM  IBM47          -    -    -    -    -    -    -    -    -    -    -    -    -
Table: SYSIBM.SYSEVENTS
SYSIBM  IBM48          -    -    -    -    -    -    -    -    -    -    -    -    -
Table: SYSIBM.SYSEVENTTABLES
SYSIBM  IBM165         -    -    -    -    -    -    -    -    -    -    -    -    -
SYSIBM  IBM167         -    -    -    -    -    -    -    -    -    -    -    -    -
Table: SYSIBM.SYSFUNCMAPOPTIONS
SYSIBM  IBM90          -    -    -    -    -    -    -    -    -    -    -    -    -
Table: SYSIBM.SYSFUNCMAPPARMOPTIONS
SYSIBM  IBM91          -    -    -    -    -    -    -    -    -    -    -    -    -
Table: SYSIBM.SYSFUNCMAPPINGS
SYSIBM  IBM92          -    -    -    -    -    -    -    -    -    -    -    -    -
SYSIBM  IBM93          -    -    -    -    -    -    -    -    -    -    -    -    -
SYSIBM  IBM94          -    -    -    -    -    -    -    -    -    -    -    -    -
SYSIBM  IBM95          -    -    -    -    -    -    -    -    -    -    -    -    -
SYSIBM  IBM96          -    -    -    -    -    -    -    -    -    -    -    -    -
Table: SYSIBM.SYSHIERARCHIES
SYSIBM  IBM86          -    -    -    -    -    -    -    -    -    -    -    -    -
SYSIBM  IBM87          -    -    -    -    -    -    -    -    -    -    -    -    -
Table: SYSIBM.SYSINDEXAUTH
SYSIBM  IBM17          6     2    0    2    57    96     6 100    4 909 100    0    -
SYSIBM  IBM18          6     2    0    2    32    151    6 100    3 +++ 100    0    -
Table: SYSIBM.SYSINDEXCOLUSE
SYSIBM  IBM139         506    9    0    2    36    114    506 96    61 15 20    0    -
Table: SYSIBM.SYSINDEXES
SYSIBM  IBM02          188    4    1    2    22    40    188 80    47 61 20    0    *
SYSIBM  IBM03          188    3    0    2    31    8     188 77    61 47 4    0    *
SYSIBM  IBM126         188    1    0    1    12    4     1 100    -    - 2    0    -
Table: SYSIBM.SYSINDEXEXPLOITRULES
SYSIBM  IBM97          -    -    -    -    -    -    -    -    -    -    -    -    -
Table: SYSIBM.SYSINDEXEXTENSIONMETHODS
SYSIBM  IBM100         -    -    -    -    -    -    -    -    -    -    -    -    -
SYSIBM  IBM101         -    -    -    -    -    -    -    -    -    -    -    -    -
Table: SYSIBM.SYSINDEXEXTENSIONPARMS
SYSIBM  IBM98          -    -    -    -    -    -    -    -    -    -    -    -    -
Table: SYSIBM.SYSINDEXEXTENSIONS
SYSIBM  IBM99          -    -    -    -    -    -    -    -    -    -    -    -    -
...

```

CLUSTERRATIO or normalized CLUSTERFACTOR (F4) will indicate REORG is necessary for indexes that are not in the same sequence as the base table. When multiple indexes are defined on a table, one or more indexes might be flagged as needing REORG. Specify the most important index for REORG sequencing.

Tables defined using the DIMENSION clause and the corresponding dimension indexes have a '\*' suffix to their names. The cardinality of a dimension index is equal to the Active blocks statistic of the table.

表統計の用語 (公式 1～3) 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。

### TABLEPAGE SIZE

表データが存在する表スペースのページ・サイズ。

**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、表の中でむだ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。

たとえば、--- 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。

**注:** 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。

**CARD** 基本表の行数。

**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。

**LVLS** 索引レベルの数 (NLEVELS)

**ISIZE** 索引サイズ、索引に関係するすべての列による列の平均の長さから計算される。

**KEYS** 削除マークの付いていないユニーク索引項目の数 (FULLKEYCARD)

#### **INDEXPAGESIZE**

表索引が存在する表スペースのページ・サイズ。表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。

#### **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。デフォルトは 10 です。

**F4** 公式 4 の結果。

**F5** 公式 5 の結果。表記法 +++ は、結果が 999 を超えて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。

**F6** 公式 6 の結果。表記法 +++ は、結果が 9999 を超えて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。

**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、REORG INDEXES 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、REORG INDEXES 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。

#### 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、通称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。

**重要:** 空の表 (TSIZE=0) が再編成を必要としている場合には、これらの統計を参考にせず、次の規準を当てはめてください。つまり、TSIZE=0 で FPAGE>0 の場合は、表の再編成が必要です。TSIZE=0 で FPAGE=0 の場合は、再編成は必要ありません。

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。

$$100 * TSIZE / ((FPAGES - 1) * (TABLEPAGE SIZE - 76)) > 68$$

バイトで表された表のサイズ (TSIZE) は、表に割り当てられた合計スペースの 68 % を超えていなければなりません (フリー・スペースを 32 以下の値にします)。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 76 バイトを差し引きします)。割り当てられている最終ページは、通常埋められていないため、FPAGES から 1 を引きします。

MDC 表では、以下のようになります。

$$100 * TSIZE / ((ACTBLK - FULLKEYCARD) * EXTENTSIZE * (TABLEPAGE SIZE - 76)) > 68$$

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。公式は、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表のブロックの 68% 以上かどうかをチェックします。

- 公式 F3:

$$100 * NPAGES / FPAGES > 80$$

全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)。

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:

$$\text{CLUSTERRATIO or normalized CLUSTERFACTOR} > 80$$

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。

- 公式 F5:

$$100 * (\text{KEYS} * (\text{ISIZE} + 9) + (\text{CARD} - \text{KEYS}) * 5) / ((\text{NLEAF} - \text{NUM\_EMPTY\_LEAFS}) * \text{INDEXPAGE SIZE}) > 50$$

索引項目に予約されたスペースの 50% 以下は空でなければなりません (NLEAF > 1 の場合のみ検査されます)。

- 公式 F6:

$$\frac{(100 - \text{PCTFREE}) * ((\text{INDEXPAGESIZE} - 96) / (\text{ISIZE} + 12)) * (\text{NLEVELS} - 2) * (\text{INDEXPAGESIZE} - 96) / (\text{KEYS} * (\text{ISIZE} + 9) + (\text{CARD} - \text{KEYS}) * 5)}{< 100}$$

索引の再作成がツリーのレベル数を減少させるかどうかを判別するために、この公式は現行のツリーよりも 1 つ低いレベルの索引ツリー内にあるスペースの量と必要なスペースの量との比率を検査します。1 つ低いレベルのツリーを作成しても PCTFREE が使用可能であれば、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うことができる項目の数の 90% 以上 (または 100-PCTFREE) でなければなりません (NLEVELS>1 の場合のみ検査されます)。

- 公式 F7:

$$100 * (\text{NUMRIDS\_DELETED} / (\text{NUMRIDS\_DELETED} + \text{CARD})) < 20$$

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:

$$100 * (\text{NUM\_EMPTY\_LEAFS} / \text{NLEAF}) < 20$$

疑似空白リーフ・ページの数、リーフ・ページの合計数の 20% 未満でなければなりません。

**注:** 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。

#### 関連資料:

- 666 ページの『REORG INDEXES/TABLE』
- 720 ページの『RUNSTATS』

## 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使用可能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。

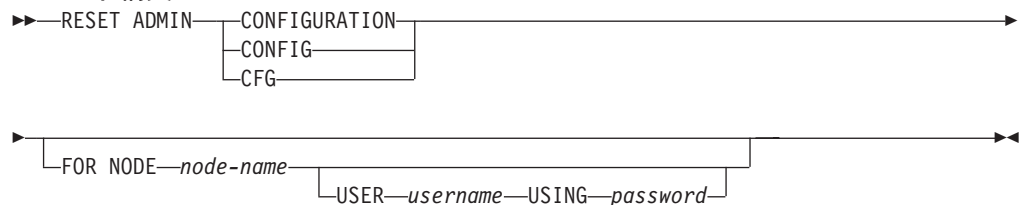
### 権限:

*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 オプションの引き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。



## RESET ADMIN CONFIGURATION

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 が再始動され、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:

- 407 ページの『GET ADMIN CONFIGURATION』
- 773 ページの『UPDATE ADMIN CONFIGURATION』
- 「管理ガイド: パフォーマンス」の『構成パラメーターのサマリー』



## RESET ALERT CONFIGURATION

ヘルス・インディケーターの設定を、特定のオブジェクトに関して、そのオブジェクト・タイプの現行のデフォルトにリセットするか、またはオブジェクト・タイプについて現行デフォルトのヘルス・インディケーターの設定を、インストール時のデフォルトにリセットします。

### 権限:

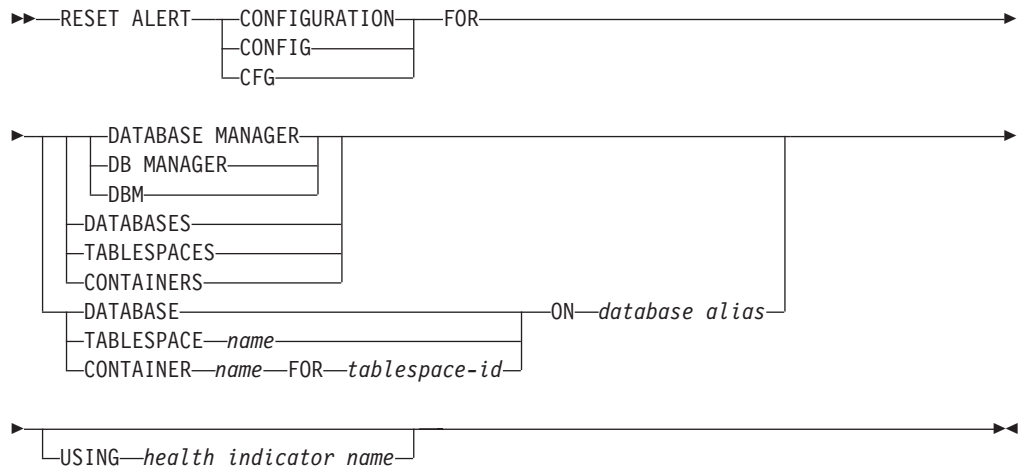
以下のどれかが必要です。

- 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

## RESET ALERT CONFIGURATION

*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。

### **BUFFERPOOL** *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* という名前のバッファ・プールのアラート設定をリセットします。このバッファ・プール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。たとえば、以下のようにします。

```
db.sort_privmem_util
```

このオプションを指定しない場合は、指定したオブジェクトまたはオブジェクト・タイプのすべてのヘルス・インディケーターがリセットされます。

## RESET DATABASE CONFIGURATION

特定データベースの構成をシステム・デフォルトにリセットします。

### 有効範囲:

このコマンドは、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。

### 権限:

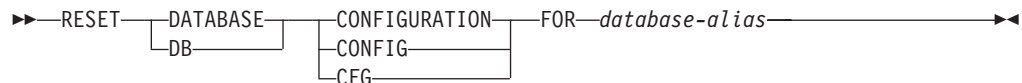
以下のどれかが必要です。

- *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。

### 関連タスク:

## RESET DATABASE CONFIGURATION

- 「管理ガイド: 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:

- 422 ページの『GET DATABASE CONFIGURATION』
- 788 ページの『UPDATE DATABASE CONFIGURATION』
- 「管理ガイド: パフォーマンス」の『構成パラメーターのサマリー』



## RESET DATABASE MANAGER CONFIGURATION

データベース・マネージャー構成ファイルへの変更の一部は、ファイルがメモリーにロードされた後にのみ有効になります。オンラインで構成できるパラメーターと構成できないパラメーターについては、構成パラメーターの一覧をご覧ください。即時にリセットされないサーバー構成パラメーターは、**db2start** の実行中にリセットされます。クライアント構成パラメーターの場合、パラメーターは次にアプリケーションを開始するときにリセットされます。クライアントがコマンド行プロセッサである場合は、**TERMINATE** を呼び出すことが必要です。

エラーが生じた場合には、データベース・マネージャー構成ファイルは変更されません。

データベース・マネージャー構成ファイルは、そのチェックサムが無効であると、リセットすることができません。これはデータベース・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:

- 「管理ガイド: 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:

- 428 ページの『GET DATABASE MANAGER CONFIGURATION』
- 761 ページの『TERMINATE』
- 791 ページの『UPDATE DATABASE MANAGER CONFIGURATION』
- 「管理ガイド: パフォーマンス」の『構成パラメーターのサマリー』

RESET MONITOR

指定されたデータベース、またはすべてのアクティブ・データベースの、内部のデータベース・システム・モニター・データ域をゼロにリセットします。内部のデータベース・システム・モニター・データ域には、データベース用のデータ域のほかに、データベースに接続されるすべてのアプリケーション用のデータ域が含まれます。

権限:

以下のどれかが必要です。

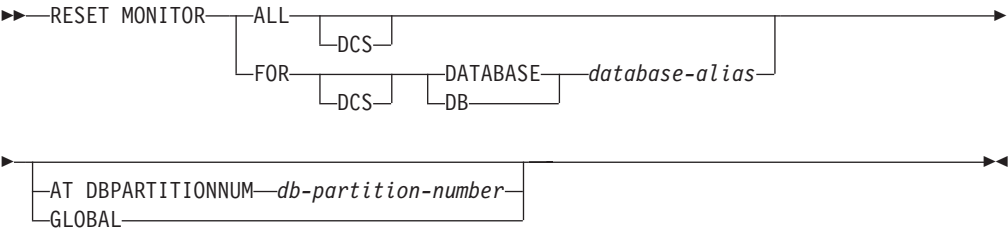
- *sysadm*
- *sysctrl*
- *sysmaint*
- *sysmon*

必要な接続:

インスタンス。インスタンス・アタッチがない場合、デフォルトのインスタンス・アタッチが作成されます。

リモート・インスタンス (または異なるローカル・インスタンス) 用のモニター・スイッチをリセットするには、最初にそのインスタンスにアタッチすることが必要です。

コマンド構文:



コマンド・パラメーター:

**ALL** このオプションは、すべてのデータベースについて内部カウンターがリセットされることを指示します。

**FOR DATABASE database-alias**

このオプションは、別名 *database-alias* を持つデータベースの内部カウンターのみがリセットされることを指示します。

**DCS** このキーワードは、指定された文節に従って、次のいずれかの内部カウンターをリセットします。

- すべての DCS データベース
- 特定の DCS データベース

**AT DBPARTITIONNUM db-partition-number**

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

### GLOBAL

パーティション・データベース・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:

各プロセス (アタッチ) は、モニター・データの私用ビューを持っています。あるユーザーがリセット、またはモニター・スイッチをオフにしても、その他のユーザーは影響されません。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 が指定されると、あるデータベース・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、あるパーティション・レベルのカウンターがリセットされます。

#### 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:

- 452 ページの『GET SNAPSHOT』
- 443 ページの『GET MONITOR SWITCHES』
- 791 ページの『UPDATE DATABASE MANAGER CONFIGURATION』



## RESTART DATABASE

異常終了し、矛盾した状態のままであるデータベースを再始動します。ユーザーが **CONNECT** 特権を持っている場合には、**RESTART DATABASE** が正常終了すると、アプリケーションはデータベースに接続されたままとなります。

### 有効範囲:

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

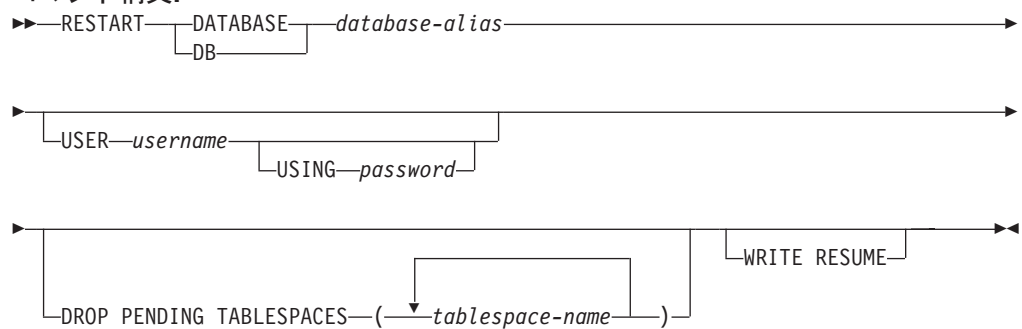
### 権限:

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され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

## RESTART DATABASE

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。

**注:** 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、このデータベースとの前のセッションが、異常に (たとえば、電源障害により) 終了した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可能ですが、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 システム中の単一データベース・パーティションでのみ再始動される場合、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・パーティションが再始動された場合です。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:

- 538 ページの『LIST INDOUBT TRANSACTIONS』

## RESTORE DATABASE

DB2 バックアップ・ユーティリティを使用してバックアップされた損傷のある、または破壊されたデータベースを再作成します。リストアされたデータベースは、バックアップ・コピーが行われた時と同じ状態になります。さらにこのユーティリティにより、データベースに異なるイメージを上書きしたり、新しいデータベースにリストアしたりすることもできます。

DB2 バージョン 8、32 ビット Windows 版で作成されたデータベースを、DB2 バージョン 8、64 ビット Windows 版にリストアしたり、その逆にリストアしたりすることが可能です。DB2 バージョン 8、32 ビット Linux (Intel) 版で作成されたデータベースを、DB2 バージョン 8、64 ビット Linux (Intel) 版にリストアしたり、その逆にリストアしたりすることが可能です。DB2 バージョン 8、AIX、HP-UX、または Solaris オペレーティング環境版 (32 ビットまたは 64 ビット) で作成されたデータベースを、DB2 バージョン 8、AIX、HP-UX、または Solaris オペレーティング環境版 (32 ビットまたは 64 ビット) にリストアできます。

さらにリストア・ユーティリティを使用すると、ワード・サイズ (32 ビットまたは 64 ビット) が同じである限り、DB2 の旧バージョン (2 バージョン前まで) で作成されたバックアップ・イメージをリストアすることもできます。DB2 の旧バージョンで作成されたバックアップ・イメージのクロスプラットフォーム・リストア操作は、サポートされていません。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ときに、データベースがすべてのロールフォワード・リカバリに対して使用可能である場合、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、データベースを損傷または破壊が起きる前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。

増分イメージと差分イメージは、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) が同じでない場合、リストアできません。

ある環境からそれとは異なる環境へのリストアの後には、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は実行できません。(同じ環境へのリストアの場合、この制限はありません。)

ある環境からそれとは異なる環境へのリストアが成功した場合も、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ことが必要です (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。

## RESTORE DATABASE

### 権限:

既存のデータベースにリストアするには、以下のいずれかが必要です。

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

新規データベースにリストアするには、以下のいずれかが必要です。

- *sysadm*
- *sysctrl*

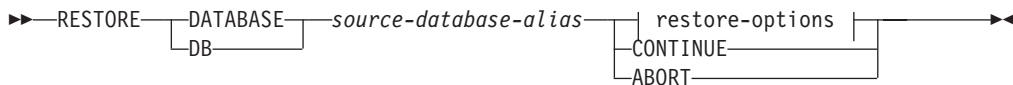
### 必要な接続:

必要な接続は、実行するリストア・アクションの種類によって異なります。

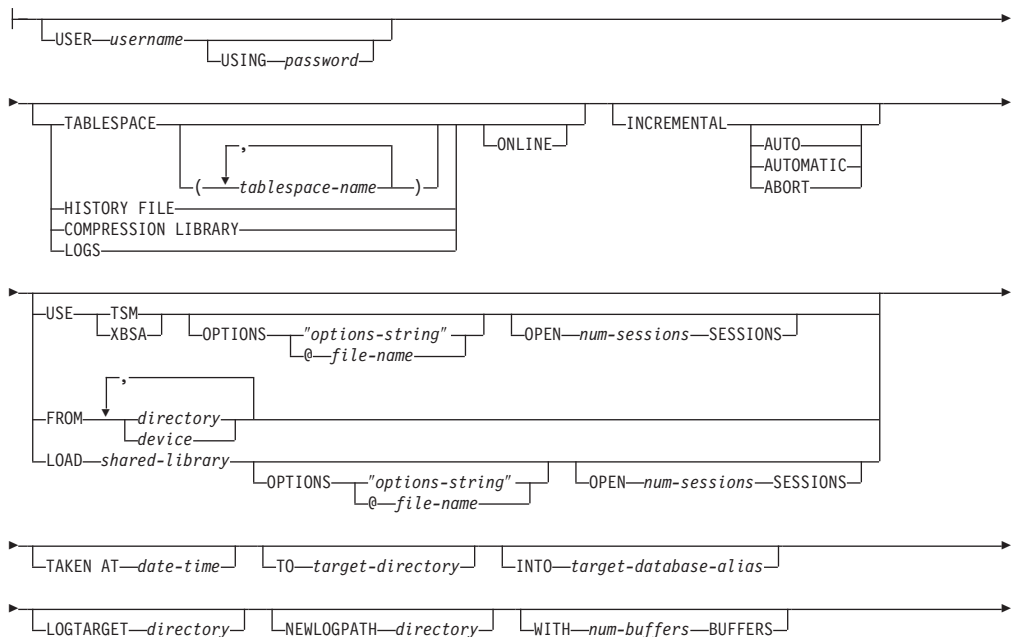
- データベース (既存のデータベースにリストアする場合)。このコマンドは、指定されたデータベースへの排他接続を自動的に確立します。
- インスタンスおよびデータベース (新規データベースにリストアする場合)。データベースを作成するには、インスタンス・アタッチが必要です。

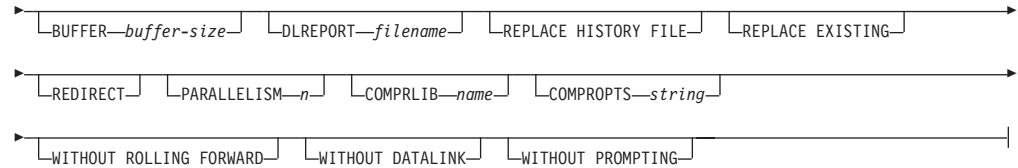
現行のインスタンスとは異なるインスタンスで新規のデータベースへのリストアを行うには、まず、新規のデータベースを存在させるインスタンスにアタッチすることが必要です。新規インスタンスは、ローカルでもリモートでもかまいません。現在のインスタンスは、DB2INSTANCE 環境変数の値によって定義されます。

### コマンド構文:



### restore-options:



**コマンド・パラメーター:****DATABASE** *source-database-alias*

バックアップが取得されるソース・データベースの別名です。

**CONTINUE**

コンテナが再定義されていること、およびリダイレクトしたリストア操作の最終ステップを実行する必要があることを指定します。

**ABORT**

このパラメーターは以下を指定します。

- リダイレクトしたリストア操作を停止します。これは、1 つまたは複数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。

**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ていない場合、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

## RESTORE DATABASE

にログ・ファイルが含まれていない場合、リストア操作は失敗します。このオプションを指定する場合は、LOGTARGET オプションも指定する必要があります。

### INCREMENTAL

INCREMENTAL は、追加のパラメーターを使用しないで手動累積リストア操作を指定します。手動リストアの際、ユーザーはリストアに含まれるイメージごとに各リストア・コマンドを手動で発行する必要があります。以下の順序でこれを行ってください。最後、1 番目、2 番目、以下同様に最後のイメージまで。

### INCREMENTAL AUTOMATIC/AUTO

自動累積リストア操作を指定します。

### INCREMENTAL ABORT

手動累積リストア操作を指定します。

### USE TSM

データベースが 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。

**注:** 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、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。



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。たとえば、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 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なります。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 (たとえば、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 (たとえば、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。たとえば、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*

ターゲット・データベース・ディレクトリーです。ユーティリティーが存在するデータベースへリストアしている場合には、このパラメーターは無視されます。指定するドライブおよびディレクトリーは、ローカルのものでなければなりません。

**注:** Windows オペレーティング・システムでは、このパラメーターを使用するときに、ドライブ名のみを指定してください。パスを指定すると、エラーが戻されます。

## RESTORE DATABASE

### **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。新規のデータベース名は、リストア先のシステムでユニークな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。たとえば、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** はこのパラメーターのため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** はこのパラメーターのため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。



**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すで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 つまたは複数の **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。

**注:** 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。

**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。

**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・マニピュレーターの数を指定します。値を明示的に指定しない場合、**DB2** はこのパラメーターのため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

## RESTORE DATABASE

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。

### WITHOUT PROMPTING

リストア操作を無人で実行するように指定します。通常はユーザー介入を必要とするアクションでは、エラー・メッセージが戻されます。テープやディスクセットなどの取り外し可能メディア装置を使用している場合、このオプションを指定していても、その装置が終わるとプロンプトが出されます。

例:

1. 以下の例で、データベース WSDb は 0 から 3 までの番号が付けられた 4 つの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'
```

db2\_all ユーティリティーは、指定された各データベース・パーティションにリストア・コマンドを出します。

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。

```
db2 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。

```
db2 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。たとえば、

```
db2 set tablespace containers for 5 using
(file 'f:¥ts3con1' 20000, file 'f:¥ts3con2' 20000)
```

リストアしたデータベースのコンテナが、このステップで指定したものであることを検査するために、`LIST TABLESPACE CONTAINERS` コマンドを発行する。

- c. ステップ 1 および 2 が正常終了した後、次を発行する。

```
db2 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増分バックアップの週間予定のサンプルです。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。

## RESTORE DATABASE

- 区切り文字付きストリングを指定する場合、

```
db2 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、

```
db2 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。

### 使用上の注意:

- db2 restore db <name> という形式の RESTORE DATABASE コマンドは、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・リストアを実行します。db2 restore db <name> tablespace という形式のすべての RESTORE DATABASE コマンドは、イメージ内で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。表スペースのリストが提供されるすべての RESTORE DATABASE コマンドは、明示的にリストされているすべての表スペースのリストアを実行します。
- オンライン・バックアップのリストアを実行したなら、その後にロールフォワード・リカバリーを実行してください。
- バックアップ・イメージが圧縮されているなら、そのことは DB2 によって検出され、リストア前に自動的にデータが解凍されます。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、バックアップ・イメージにライブラリーが格納されているなら、それが使用されます。そうでない場合、データは解凍できず、リストアは失敗します。
- 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 (リストアのタイプとして DB2RESTORE\_COMPR\_LIB 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は、LOGTARGET オプションを指定し、それに DB2 サーバー上に存在する有効な完全修飾パス名を指定する必要があります。それらの条件が満たされている場合、リストア・ユーティリティーは、イメージ内のログ・ファイルをターゲット・パスに書き込みます。ログを含まないバックアップ・イメージのリストアで LOGTARGET を指定した場合、リストア操作で表スペース・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、リストアからエラーが戻されます。
- RESTORE DATABASE コマンド発行時点に LOGTARGET パス内にログ・ファイルが存在している場合、ユーザーに対して警告プロンプトが戻されます。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、なんらかの理由で抽出できないログ・ファイルがあった場合、リストアは失敗し、エラーが戻されます。バックアップ・イメージから抽出するログ・ファイルの中に、LOGTARGET パス内に既存のファイルと同じ名前のもので 1 つでもあると、リストア操作は失敗し、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・イメージに含まれているログ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が、それらの中間バックアップ・イメージから抽出されることはありません。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。

**関連資料:**

- 299 ページの『BACKUP DATABASE』
- 709 ページの『ROLLFORWARD DATABASE』
- 141 ページの『db2move - データベース移動ツール』

---

## REWIND TAPE

Windows NT ベースのオペレーティング・システムで実行する場合、DB2 は、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をサポートしています。このコマンドを使用してテープを巻き戻します。

**権限:**

なし

**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。デフォルトは ¥¥.¥TAPE0 です。

**関連資料:**

- 509 ページの『INITIALIZE TAPE』
- 739 ページの『SET TAPE POSITION』

## ROLLFORWARD DATABASE

データベースのログ・ファイルに記録されたトランザクションを適用することによって、データベースをリカバリーします。データベースまたは表スペースのバックアップ・イメージがリストアされた後、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。指定された特定時刻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。

### 権限:

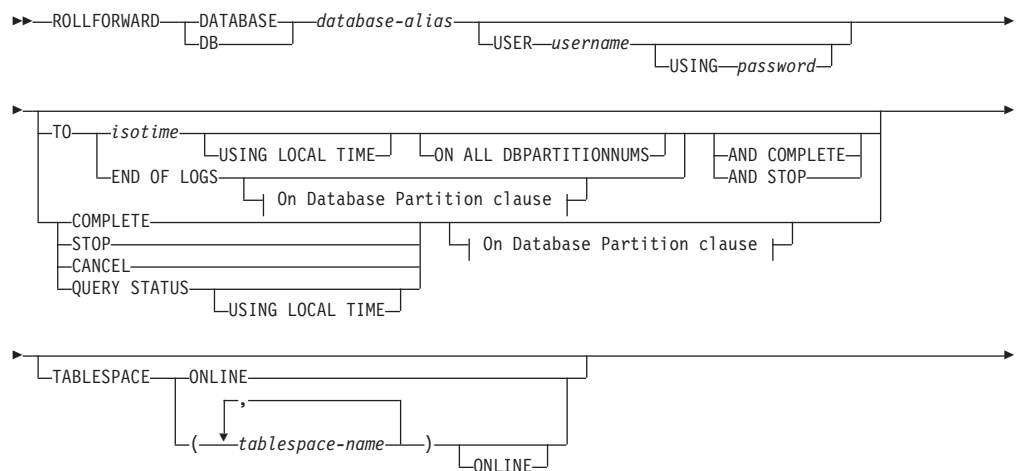
以下のいずれか。

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

### 必要な接続:

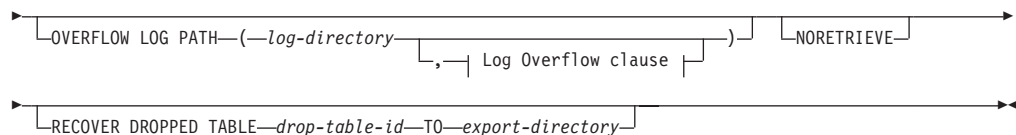
なし。このコマンドは、データベース接続を確立します。

### コマンド構文:

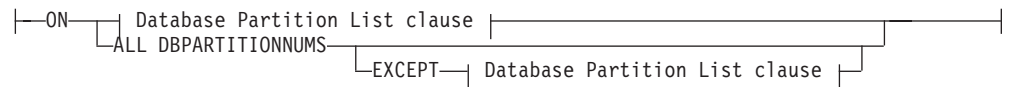




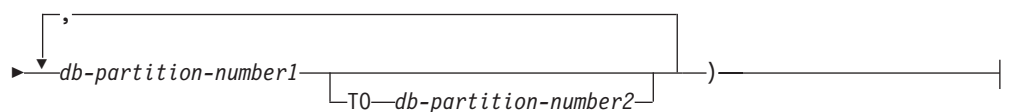
## ROLLFORWARD DATABASE



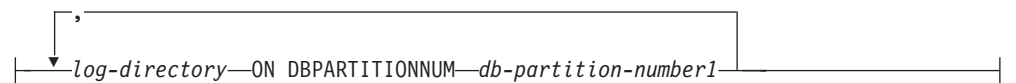
### On Database Partition clause:



### 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、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がロールフォワードされる時点 (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)。

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.nnnnnnn` (年、月、日、時、分、秒、マイクロ秒) の協定世界時 (UTC) です。UTC は、さまざまなログに関連したタイム・スタンプが (たとえば、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

ユーザーが GMT 時間ではなくユーザ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。これによって、ユーザーがローカル・マシンで特定の時点にロールフォワードすることが容易になり、現地時間を GMT ポイント・イン・タイムに変換することによって生じる潜在的なユーザー・エラーの発生を除去します。

#### 注:

1. ユーザーがロールフォワード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ることに注意してください。
2. タイム・スタンプ・ストリングはサーバー上で GMT 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。

### 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

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、すで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 (たとえば、`db2 rollforward db sample to end of logs and complete`)。

**注:** (end of logs ではなく) 特定時刻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、すべてのパーティション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。
- ある時点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。たとえば、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。

**注:** 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 (CUT 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。

### データベース・パーティション番号

#### ロールフォワード状況

状態は次のいずれかです。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、そのディレクトリーから除去できる、処理済みのログ・ファイルの名前から成るストリング。たとえば、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)。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。

注: 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、コマンドが正常に実行されれば、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

ある時点までの表スペース・レベルのロールフォワード・リカバリーを行う  
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。

### 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  
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。

注: 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

ユーザーがアーカイブ・ログの検索を使用不可にできるようにして、ユーザ  
ーがスタンバイ・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  
を制御することを可能にします。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り、スタンバ  
イ・マシンが実動マシンよりも必ず X 時間遅れているようにして、ユー  
ザーが両方のシステムに影響を与えることを防止できます。

- 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 (たとえば、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。

### 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。たとえば、ログの終わりま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。

#### 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 つをある時点までロールフォワードします (両方ともオンラインで行われます)。

```
db2 rollforward db sample to end of logs tablespace(TBS1) online
```

```
db2 rollforward db sample to 1998-04-03-14.21.56.245378 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ーを指定して、ある時点までロールフォワードします。

```
db2 rollforward db sample to 1998-04-03-14.21.56.245378 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 はすべてのパーティションで定義されており、表スペース TBS2 はパーティション 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 1998-04-03-14.21.56.245378 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。すべての部分と一緒にロールフォワードされなければなりません。

**注:** 表スペースをある時点までロールフォワードする場合、データベース・パーティション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 1998-04-03-14.21.56.245378 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、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、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 (ある時点ではなく) までのこの操作は正常に完了します。表スペースが存在するデータベース・パーティションは指定する必要がありません。ユーティリティーは、デフォルトとして db2nodes.cfg ファイルを使用します。

#### 例 9 (パーティション・データベース環境)

(データベース・パーティション 6 上の) 単一パーティションのデータベース・パーティション・グループに存在する 6 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(tssstore, tssbuyer, tsstime, tsswhse, tsslscat, tssvendor)
```

ログの終わり (ある時点ではなく) 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時点までロールフォワードしている場合は、ロールフォワード操作は、最低でも最小リカバリー時間（この表スペース用のシステム・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（たとえば、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（たとえば、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。

ロールフォワード・ユーティリティが、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。使用可能な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。

### 関連資料:

- 299 ページの『BACKUP DATABASE』
- 697 ページの『RESTORE DATABASE』



## 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。さらに具体的に言えば、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。

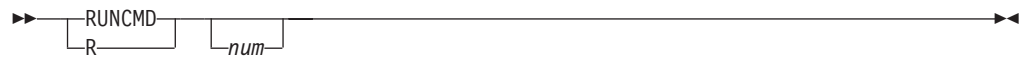
### 権限:

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:

- 391 ページの『EDIT』
- 482 ページの『HISTORY』

# RUNSTATS

表およびそれに関連した索引の物理的特性についての統計を更新します。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・パスを決定する際にこれらの統計を使用します。

このユーティリティは、表が数多く更新される時、または表を再編成した後で、呼び出してください。

**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。カタログ・データベース・パーティションのカタログを更新するのに使用します。

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。

**権限:**

以下のいずれかが必要です。

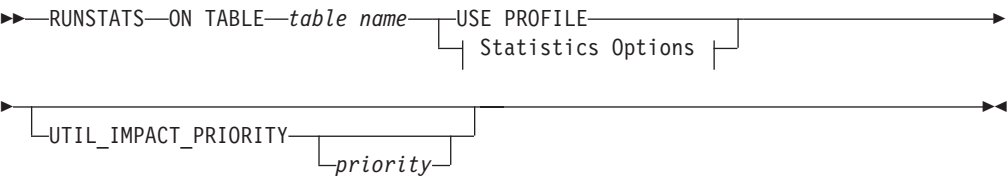
- sysadm
- sysctrl
- sysmaint
- dbadm
- 表に対する CONTROL 特権
- LOAD 権限

このコマンドを使用するうえでは、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。

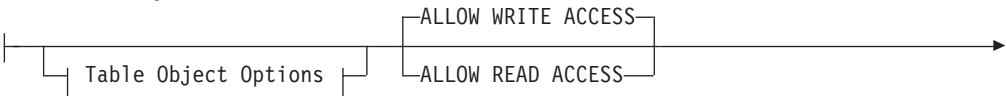
**必要な接続:**

データベース

**コマンド構文:**

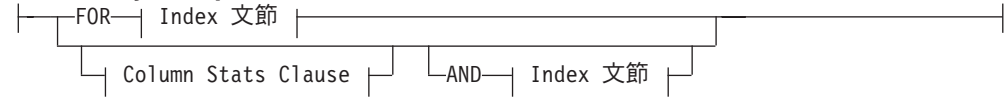


**Statistics Options:**

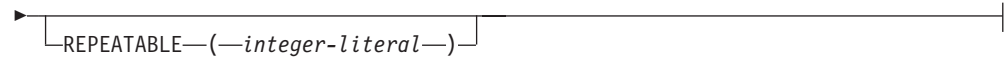
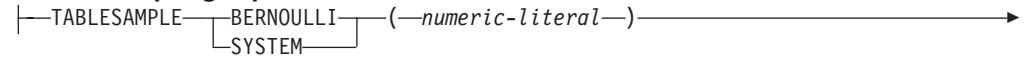




**Table Object Options:**



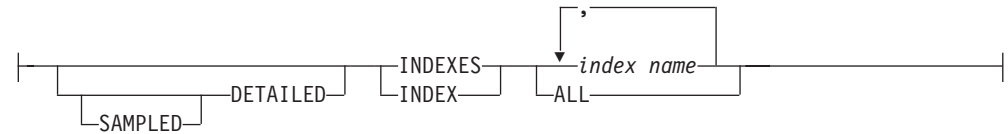
**Table Sampling Options:**



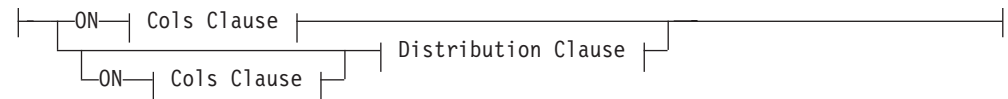
**Profile Options:**



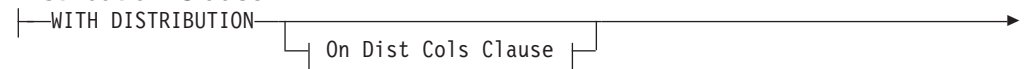
**Index Clause:**



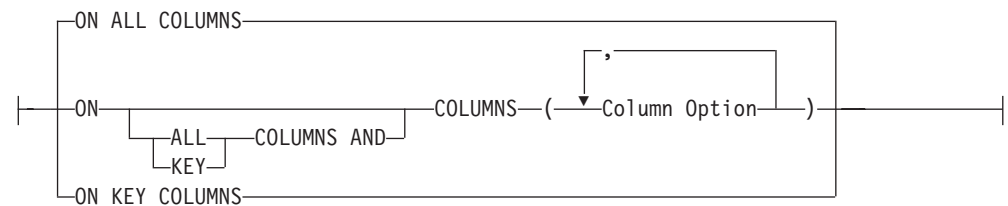
**Column Stats Clause:**

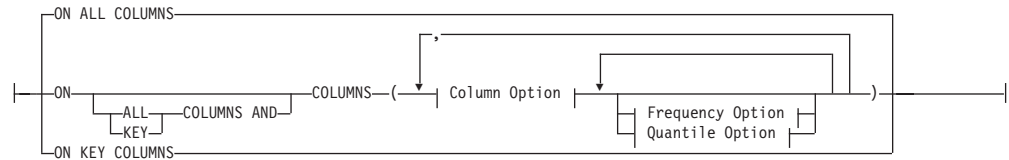
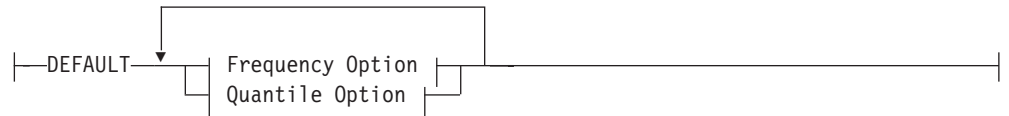
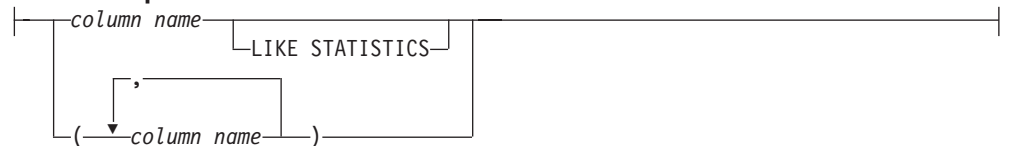


**Distribution Clause:**



**On Cols Clause:**



**On Dist Cols Clause:****Default Dist Option:****Frequency Option:****Quantile Option:****Column Option:****コマンド・パラメーター:****table-name**

統計が収集される表を識別します。これは、カタログで記述されている表でなければならず、ビューや階層表であってはなりません。タイプ表の場合は、`table-name` を表階層のルート表の名前にしてください。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。

**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。

**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。

**AND INDEXES**

表と索引両方の統計を収集および更新します。

**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。

**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 のサンプリング技術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。

**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 (`on-cols-clause`) こともできますし、`WITH DISTRIBUTION` 文節と組み合わせて指定する (`on-dist-cols-clause`) ことも可能です。これら列指定の文節がどちらも指定されない場合は、デフォルトで `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。リストしない列の統計は消去されます。この文節は、`on-cols-clause` と `on-dist-cols-clause` の中で使用できます。

**注:** 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。

**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。この文節は、`on-cols-clause` または `on-dist-cols-clause` で使用できます。ただし、`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` と `on-dist-cols-clause` の両方で指定を行うと、`on-cols-clause` で重複が生じます。

**column-name**

`column-name` は、表内の列の名前でなければなりません。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。値が個々の列でも DEFAULT 文節でも指定されていない場合は、NUM\_FREQVALUES データベース構成パラメーターで設定されている値が、収集の頻度を指定する値の最大値になります。

## NUM\_QUANTILES

収集する分散変位値の最大値を定義します。これは、ON COLUMNS 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。値が個々の列でも DEFAULT 文節でも指定されていない場合は、NUM\_QUANTILES データベース構成パラメーターで設定されている値が、収集する変位値の最大値になります。

## WITH DISTRIBUTION

この文節は、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、表内のすべての列 (CLOB や LONG VARCHAR といった、収集が不可能な列を除く) 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 (統計収集を行えない列を除く) 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。

**注:** 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ーにある値が使用されます。

## LIKE STATISTICS

このオプションを指定すると、付加的な列統計が収集されます。収集されるのは、SYSSTAT.COLUMNS の SUB\_COUNT および SUB\_DELIM\_LENGTH 統計です。これらの統計は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性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。

**ALLOW READ ACCESS**

統計が計算される間に、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。

**TABLESAMPLE BERNOULLI**

このオプションを使用した RUNSTATS は、表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 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。たとえば、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 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。

**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、表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。負のシードを引き渡すと、エラーになります (SQL1197N)。TABLESAMPLE REPEATABLE の最後の実行以降に行われた表に対する活動によって表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ることに注意してください。また、必ず整合した結果を得るためには、BERNOULLI または SYSTEM 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。たとえば、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 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、表のサンプルングが指定されていない場合と同じように処理されます。100 より大きい値または 0 未満の値は、DB2 ではエラーとして処理されます (SQL1197N)。



## 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 コマンド・オプションを実行しません。

## UPDATE PROFILE

RUNSTATS は、システム・カタログ表内の既存の統計プロファイルを変更し、その更新済みの統計プロファイルの RUNSTATS コマンド・オプションを実行して統計を収集します。

## UPDATE PROFILE ONLY

RUNSTATS は、システム・カタログ表内の既存の統計プロファイルを変更しますが、その更新済みの統計プロファイルの RUNSTATS コマンド・オプションを実行しません。

## UTIL\_IMPACT\_PRIORITY *priority*

*priority* に指定されているレベルで、RUNSTATS をスロットルすることを指定します。*priority* は 1 から 100 の範囲の数であり、100 が最高の優先順位、1 が最低の優先順位を表します。優先順位によって、ユーティリティーのスロットルの量が決まります。優先順位が同じユーティリティーはすべて同じ量のスロットルになり、優先順位の低いユーティリティーは、優先順位の高いユーティリティーよりも量が絞られます。*priority* を指定しない場合、RUNSTATS はデフォルトの優先順位 50 を使用します。

UTIL\_IMPACT\_PRIORITY キーワードを省略すると、スロットルのサポートなしで RUNSTATS ユーティリティーが呼び出されます。

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ーが実行されます。

**注:** パーティション・データベースでは、RUNSTATS コマンドは、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. 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 (たとえば、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)



- 新しい索引が作成された場合
  - 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
2. 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、システムそのものの活動があまりない表には、可能な限り詳細に統計を収集する努力を費やす価値があります。
  - 統計を収集する時間が限られている場合、表が比較的大規模な場合、または表が頻繁に更新される場合には、述部で使用される一連の列に限って 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一連の列は、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。
  - 等式の述部で使われる列や、値の分散がスキューされる可能性のある列では、必ず分散統計を収集してください。
  - 範囲の述部を持つ列 (たとえば、“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、述部で使されない列の NUM\_FREQVALUES 値や NUM\_QUANTILES 値をあまり高くしないでください。
  - 索引の統計が要求され、索引を含む表に対して統計が実行されたことがない場合は、表と索引の両方に関する統計が計算されることに注意してください。
3. コマンドを実行した後は、以下の点に注意してください。
- ロックは、COMMIT を発行すると解除されます。
  - 新規のアクセス・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。たとえば、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 を実行すると、表のカーディナリティーが FIRSTKEYCARD よりも小さくなってしまいます。可能性があります。これが不整合です。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。
- 4. 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。たとえば、"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。たとえば、"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。
- 5. 範囲クラスター表の場合、範囲クラスター表の範囲配列プロパティーを表す特殊なシステム生成索引がカタログ表内に存在します。この種の表の統計を収集するときに、統計収集の一部として表を組み込む場合は、システム生成索引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。
- 6. コマンド構文の "On Dist Cols" 文節では、列のグループに対する "Frequency Option" または "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。
- 7. 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、以下の統計に "-1" 値があります。
  - AVERAGE\_SEQUENCE\_FETCH\_PAGES
  - AVERAGE\_SEQUENCE\_FETCH\_GAP
  - AVERAGE\_RANDOM\_FETCH\_PAGES
- 8. TABLESAMPLE による RUNSTATS サンプルングは、索引ページではなく表データ・ページでのみ行われます。索引統計とサンプルングが要求された場合は、統計収集のためにすべての索引ページがスキャンされます。そのようになるのは、TABLESAMPLE が適用される表統計の収集においてのみです。ただし、SAMPLED DETAILED オプションを使用すれば、詳細な索引統計をより効率的に収集できます。これは、TABLESAMPLE の場合とは別のサンプルング方法であり、索引統計の詳細なセットにのみ適用されます。
- 9. 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、SYSIBM.SYSTABLES システム・カタログ表の STATISTICS\_PROFILE 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。

## 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. 表でのみ統計を収集し、構成の設定から NUM\_QUANTILES を取得した場合に表に指定される頻度のしきい値を使用して、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、照会オブティマイザーが、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. 表でのみ統計を収集する統計プロファイルを登録し、構成の設定から NUM\_QUANTILES を取得した場合に表に指定される頻度のしきい値を使用して、すべての列で分散統計を収集します。このコマンドはさらに、指定のとおり統計を更新します。

13. 表でのみ統計を収集する統計プロファイルを登録し、構成の設定から NUM\_QUANTILES を取得した場合に表に指定される頻度のしきい値を使用して、すべての列で分散統計を収集します。統計は収集されません。
- ```
RUNSTATS ON TABLE db2user.employee WITH DISTRIBUTION
NUM_FREQVALUES 50 SET PROFILE
```
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らないことに注意し

てください。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 TABLESAMPLE SYSTEM(1.5)
AND INDEXES ALL
```

**関連概念:**

- 「管理ガイド: 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』
- 「管理ガイド: パフォーマンス」の『表データのサンプルでの統計の収集』
- 「管理ガイド: パフォーマンス」の『自動統計収集』

**関連タスク:**

- 「管理ガイド: パフォーマンス」の『カタログ統計の収集』

## SET CLIENT

バック・エンド・プロセス用の接続設定を指定します。

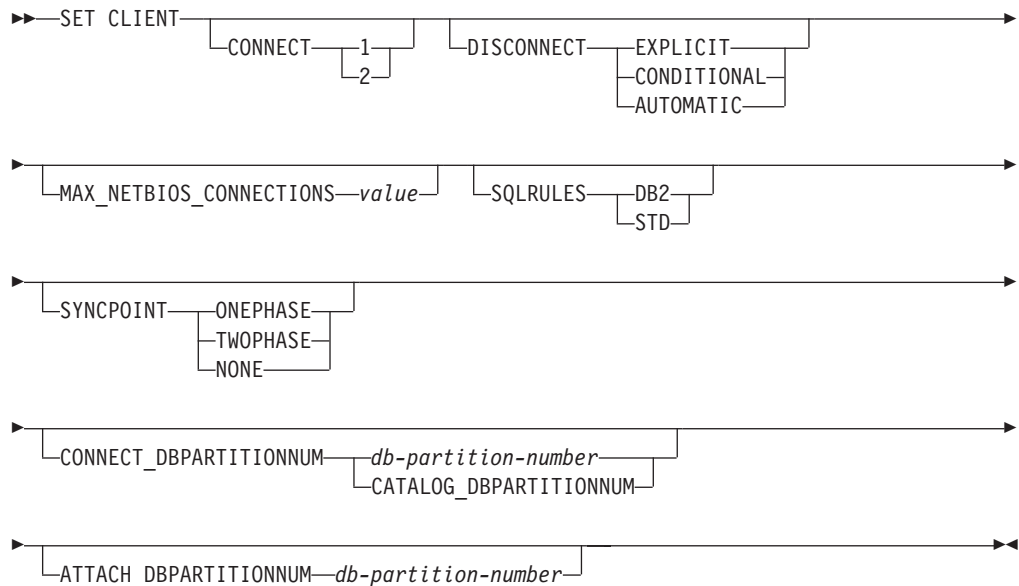
権限:

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。

**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 (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** の値を上書きします。

**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、そのカタログ・データベース・パーティションに接続できます。

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

アタッチ先のデータベース・パーティションを指定します。値は 0~999 (0 および 999 を含む) です。環境変数 **DB2NODE** の値を上書きします。

たとえば、データベース・パーティション 1、2、および 3 が定義された場合、クライアントはこれらのうちの 1 つのデータベース・パーティションにアクセスできればよいことになります。データベースを含むデータベース・パーティション 1 がカタログされ、このパラメーターが 3 に設定され

ると、次のアタッチは、まずデータベース・パーティション 1 に行われ、次いでデータベース・パーティション 3 に行われます。

### 例:

特定の値を設定するには、次のようになります。

```
db2 set client connect 2 disconnect automatic sqlrules std
syncpoint twophase
```

SQLRULES を DB2 に戻し、それ以外の設定はそのままにしておくには、次のようになります。

```
db2 set client sqlrules db2
```

**注:** 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:

- 761 ページの『TERMINATE』
- 634 ページの『QUERY CLIENT』



## SET RUNTIME DEGREE

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

### 有効範囲:

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。

### 権限:

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。 リモート・サーバーでのパーティション内並列処理の実行時の最大並列度を変更するには、最初にそのサーバーにアタッチする必要があります。アタッチがない場合、SET RUNTIME DEGREE コマンドの実行は失敗します。

### コマンド構文:

```

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

### コマンド・パラメーター:

#### FOR

**ALL** 指定した並列処理レベルはすべてのアプリケーションに適用されます。

#### application-handle

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

#### TO degree

パーティション内並列処理の実行時の最大並列度。

### 例:

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

```
db2 SET RUNTIME DEGREE FOR ( 41408, 55458 ) TO 4
```

### 使用上の注意:

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

## 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、ステートメントのランタイム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実行時の最大並列度は、**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 ステートメントの実行時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

### 関連資料:

- 517 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。たとえば、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。

**注:** ユーザー出口プログラムがリストアのために使用されている場合、リダイレクトしたリストアはできません。

### 権限:

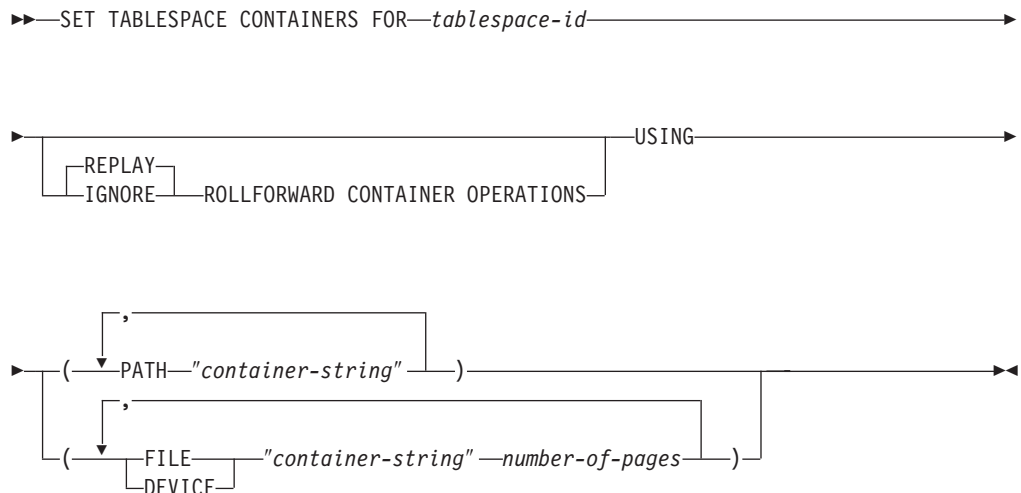
以下のいずれかが必要です。

- *sysadm*
- *sysctrl*

### 必要な接続:

データベース

### コマンド構文:



### コマンド・パラメーター:

#### FOR *tablespace-id*

リストアされるデータベースが使用する表スペースを表す固有の整数。

#### REPLAY ROLLFORWARD CONTAINER OPERATIONS

データベースのバックアップが行われた後にこの表スペースに対して発行された `ALTER TABLESPACE` 操作が、それ以後のデータベースのロールフォワード処理中に再発行されることを指定します。

#### IGNORE ROLLFORWARD CONTAINER OPERATIONS

ロールフォワード中に、ログ内の `ALTER TABLESPACE` 操作が無視されることを指定します。

### USING PATH "container-string"

SMS 表スペースに関して、表スペースに属し、表スペース・データの保管先となる、1 つ以上のコンテナを識別します。絶対または相対ディレクトリー名です。ディレクトリー名が絶対でない場合、データベース・ディレクトリーに対して相対的です。ストリング長は 240 バイト以下です。

### USING FILE/DEVICE "container-string" number-of-pages

DMS 表スペースの場合、表スペースに属し、表スペース・データの保管先となる、1 つ以上のコンテナを識別します。コンテナ・タイプ (FILE または DEVICE) およびコンテナ・サイズ (4KB ページ) を指定します。ファイル・コンテナおよび装置コンテナの混在も可能です。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。

装置コンテナの場合、ストリングは装置名でなければなりません。また、装置がすで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:

データベースのバックアップ、または 1 つ以上の表スペースには、バックアップされた表スペースにより使用されている表スペース・コンテナのすべての記録が保持されています。リストア操作中に、バックアップにリストされているすべてのコンテナは、現在存在し、アクセス可能かどうかチェックされます。1 つ以上のコンテナが何らかの理由でアクセス不能の場合、リストアは失敗します。そのようなときにリストアしたい場合、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、また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:

- 299 ページの『BACKUP DATABASE』
- 697 ページの『RESTORE DATABASE』
- 709 ページの『ROLLFORWARD DATABASE』

---

## SET TAPE POSITION

Windows NT ベースのオペレーティング・システムで実行する場合、DB2 は、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をサポートしています。このコマンドを使用して、テープの位置決めを行います。

### 権限:

なし

### 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、たとえば、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:

- 509 ページの『INITIALIZE TAPE』
- 708 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 (スロットルを使用不可にする)。
- 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:**

**権限:**

以下のいずれかが必要です。

- *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:**

- 「管理ガイド: 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』
- 556 ページの『LIST UTILITIES』

## 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。

**権限:**

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 つを持っている必要があります。

- 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



可能性があります。データベース入出力が再開すると、後続の接続は正常に完了します。接続しようとしてもハングし、入出力を中断した接続からの入出力の再開が不可能になった場合、RESTART DATABASE コマンドを WRITE RESUME オプションを指定して実行する必要があります。このような状態で使用すると、RESTART DATABASE コマンドはクラッシュ・リカバリーを実行せずに、入出力書き込みを再開します。RESTART DATABASE コマンドを WRITE RESUME オプションを指定して使用すると、データベースがクラッシュしてから使用する場合にはクラッシュ・リカバリーのみを実行します。

**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『オンライン分割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』

## 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、`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。

### 権限:

以下のいずれかが必要です。

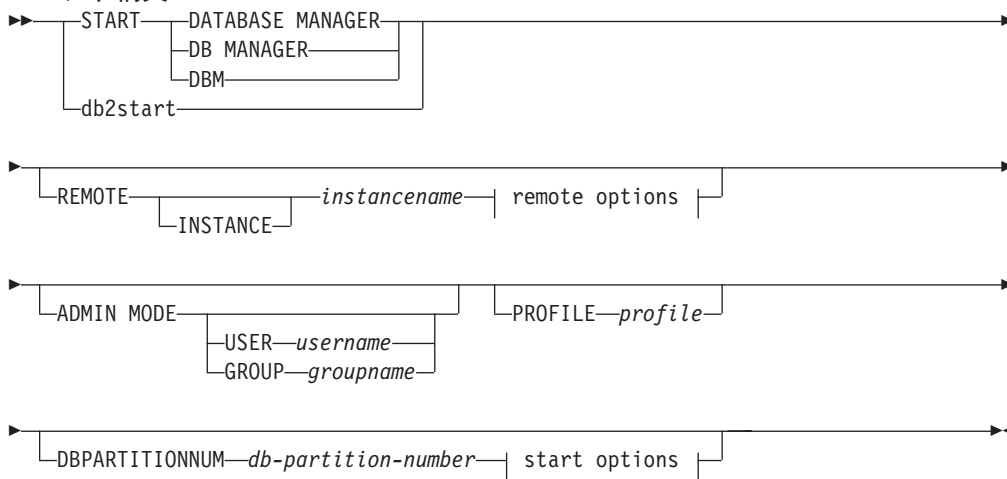
- `sysadm`
- `sysctrl`
- `sysmaint`

注: 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。

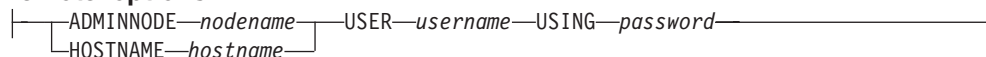
### 必要な接続:

なし

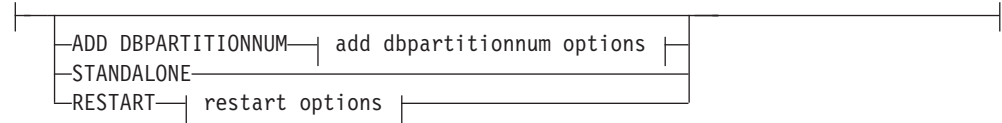
### コマンド構文:



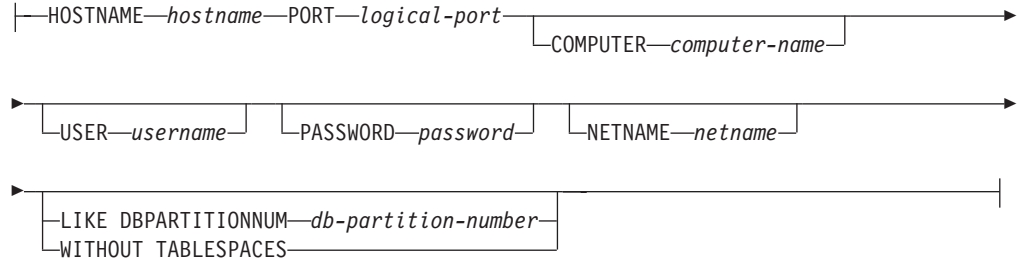
### remote options:



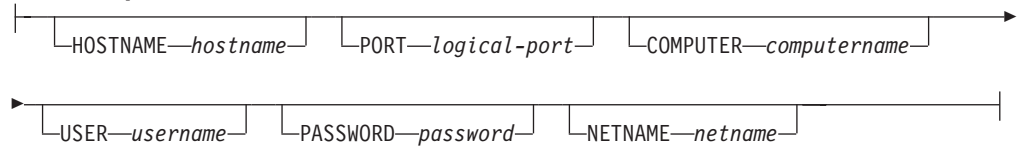
## start options:



## add dbpartitionnum options:



## restart options:



## コマンド・パラメーター:

### REMOTE [INSTANCE] instancename

開始したいリモート・インスタンスの名前を指定します。

#### ADMINNODE nodename

REMOTE または REMOTE INSTANCE では、管理ノードの名前を指定します。

#### HOSTNAME hostname

REMOTE または REMOTE INSTANCE では、ホスト・ノードの名前を指定します。

#### USER username

REMOTE または REMOTE INSTANCE では、ユーザーの名前を指定します。

#### USING password

REMOTE または REMOTE INSTANCE、および USER では、ユーザーのパスワードを指定します。

### ADMIN MODE

管理の目的のために、インスタンスを静止モードで開始します。これは、インスタンスがすでに『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。

注: 以下のすべてのパラメーターは、Enterprise Server Edition (ESE) 環境でのみ有効です。

### PROFILE profile

DB2 環境を定義するために、各データベース・パーティションで実行しなければならないプロファイル・ファイル名を指定します。このファイルは、データベース・パーティションの開始前に実行されます。プロファイル・ファイルはインスタンス所有者の `sqllib` ディレクトリーに常駐していなければなりません。

注: 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` ファイルにすで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* の組み合わせがユニーク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。

注: 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 NT 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 NT 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 NT 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**

システム 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 ファイル中にすで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。

**WITHOUT TABLESPACES**

システム 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。

**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* 値を用いて再起動されます。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 NT 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 NT 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 NT 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
```

UNIX プラットフォームでは、START DATABASE MANAGER コマンドは SIGINT および SIGALRM 信号をサポートしています。CTRL+C を押すと、SIGINT 信号が発行されます。db2start コマンド発行後にデータベース・マネージャー構成パラメーターの *start\_stop\_time* の指定値の時間が経過すると、SIGALRM 信号が発行されます。いずれかの信号が発行されると、データベース・パーティションの始動操作に割り込みが生じ、メッセージ (SIGINT の場合は SQL1044N、SIGALRM の場合は SQL6037N) が割り込みが生じたデータベース・パーティションから、\$HOME/sql1lib/log/db2start.timestamp.log エラー・ログ・ファイルに戻されます。すで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 NT オペレーティング・システムの場合、開始することを失敗した通信サブシステムがあったとしても、**db2start** コマンドも **NET START** コマンドも警告を戻しません。Windows NT 環境のデータベース・マネージャーは NT 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。NT 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:

- 752 ページの『STOP DATABASE MANAGER』
- 289 ページの『ADD DBPARTITIONNUM』

## START HADR

データベースの HADR 操作を開始します。

### 権限:

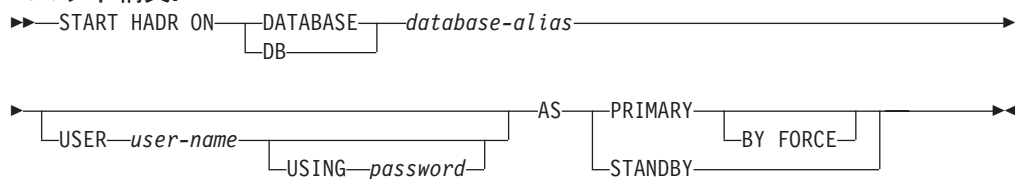
以下のいずれか。

- *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・データベースが利用可能になった時点で、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。

**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。                                                                                                        |

## 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。

### 権限:

以下のいずれかが必要です。

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

さらに、`db2_all` コマンドを実行する必要があります。

### 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、このコマンドは **SIGALRM** 信号をサポートします。**SIGALRM** 信号は、データベース・マネージャー構成パラメーターの *start\_stop\_time* の指定値が着信すると発行されます。この信号が発行されると、データベース・パーティションの停止操作に割り込みが生じ、メッセージ **SQL6037N** が割り込みが生じたデータベース・パーティションから、**\$HOME/sql1lib/log/db2stop timestamp.log** エラー・ログ・ファイルに戻されます。すで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:

- 405 ページの『**FORCE APPLICATION**』
- 744 ページの『**START DATABASE MANAGER**』

- 370 ページの『DEACTIVATE DATABASE』
- 386 ページの『DROP DBPARTITIONNUM VERIFY』

# STOP HADR

データベースの HADR 操作を停止します。

権限:

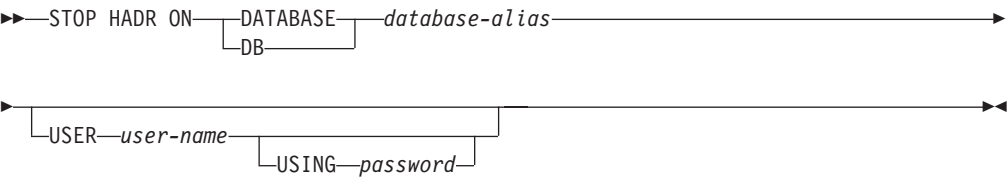
以下のいずれか。

- *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     | <p>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。</p> <p>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。</p> |
| 非アクティブ・スタンバイ・データベース | <p>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。</p>                                                                                                                                                                                            |
| アクティブ・スタンバイ・データベース  | <p>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。</p>                                                                                                                                                                                                                                                              |

## 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。

### 権限:

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 | 使用しない          | 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 | 使用する           | エラー・メッセージが戻されます。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| ピア                          | 使用しない          | <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"> <li>1. テークオーバー中に障害が発生した場合、HADR システム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。使用可能なデータベースの HADR 役割を調べるには、スナップショット・モニターを使用するか、またはデータベース構成パラメーター <code>hadr_db_role</code> の値を確認してください。</li> <li>2. 意図された新しい 1 次データベースがまだスタンバイの役割であり、それでもテークオーバーを発行したい場合は、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)。</li> <li>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 BY FORCE オプションを発行できます。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。</li> </ol> |
| ピア                          | 使用する           | <p>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。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

## TAKEOVER HADR

|  |       |          |                               |
|--|-------|----------|-------------------------------|
|  | スタンバイ | BY FORCE |                               |
|  | 状態    | オプション    | テークオーバーの動作                    |
|  | リモート・ | 使用しない    | エラー・メッセージが戻されます。              |
|  | キャッチア |          |                               |
|  | ップ・ペン |          |                               |
|  | ディング  |          |                               |
|  | リモート・ | 使用する     | 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。 |
|  | キャッチア |          |                               |
|  | ップ・ペン |          |                               |
|  | ディング  |          |                               |

---

## 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。

権限:

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処理単位の途中で、**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、それ以上使用不可能なデータベース・マネージャーのインスタンスへのアタッチ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、複数パーティション環境におけるバック・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、複数パーティション環境におけるコーディネーター・データベース・パーティション番号を指定する時に使用します。

---

## UNCATALOG DATABASE

データベース項目をシステム・データベース・ディレクトリーから削除します。

### 権限:

以下のいずれかが必要です。

- *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:

システム・データベース・ディレクトリーにある項目だけをアンカタログできます。ローカル・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、**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。

下位レベル・サーバーと通信するときに使用される、データベースの認証タイプの変更は、最初にデータベースをアンカタログし、次に別のタイプでもう一度カタログすることによって行えます。

**注:** ディレクトリーをキャッシュできる場合、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。 **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参照の間に作成されます。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してから再始動してください。

### 関連資料:

- 330 ページの『CATALOG DATABASE』
- 380 ページの『DROP DATABASE』
- 428 ページの『GET DATABASE MANAGER CONFIGURATION』
- 520 ページの『LIST DATABASE DIRECTORY』
- 761 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。

### 権限:

以下のいずれかが必要です。

- *sysadm*
- *sysctrl*

### 必要な接続:

なし。 ディレクトリー操作は、ローカル・ディレクトリーだけに影響します。

### コマンド構文:

```

▶▶—UNCATALOG DCS—┐
                     └─┬─ DATABASE ─ database-alias ─▶▶
                       └─ DB ─────────┘
  
```

### コマンド・パラメーター:

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

アンカタログする DCS データベースの別名を指定します。

### 使用上の注意:

DCS データベースは、リモート・データベースとしてシステム・データベース・ディレクトリーにもカタログされており、 UNCATALOG DATABASE コマンドを使用してアンカタログすることができます。

DCS ディレクトリーでデータベースを再カタログする場合は、 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。

**注:** ディレクトリーをキャッシュできる場合、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。 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参照の間に作成されます。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してから再始動してください。

### 関連資料:

- 334 ページの『CATALOG DCS DATABASE』

- 428 ページの『GET DATABASE MANAGER CONFIGURATION』
- 761 ページの『TERMINATE』
- 762 ページの『UNCATALOG DATABASE』
- 531 ページの『LIST DCS DIRECTORY』

## UNCATALOG LDAP DATABASE

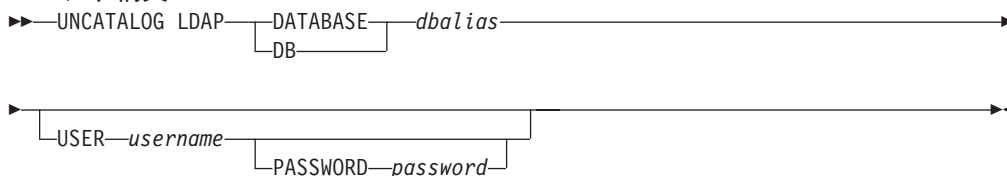
Lightweight Directory Access Protocol (LDAP) からデータベースを登録解除するのに使用します。

**権限:**

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・ユーザーの認証が使用されます。

**PASSWORD password**

アカウント・パスワード。

**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。

**関連資料:**

- 337 ページの『CATALOG LDAP DATABASE』
- 341 ページの『CATALOG LDAP NODE』



- 768 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。

権限:

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・ユーザーの認証が使用されます。

**PASSWORD** *password*

アカウント・パスワード。

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。

関連資料:

- 337 ページの『CATALOG LDAP DATABASE』
- 766 ページの『UNCATALOG LDAP DATABASE』
- 341 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。

### 権限:

以下のいずれかが必要です。

- *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。

**注:** ディレクトリーをキャッシュできる場合、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参照の間に作成されます。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してから再始動してください。

### 関連資料:

- 324 ページの『CATALOG APPC NODE』
- 349 ページの『CATALOG TCPIP NODE』
- 428 ページの『GET DATABASE MANAGER CONFIGURATION』
- 761 ページの『TERMINATE』
- 346 ページの『CATALOG NETBIOS NODE』
- 342 ページの『CATALOG LOCAL NODE』
- 327 ページの『CATALOG APPN NODE』
- 344 ページの『CATALOG NAMED PIPE NODE』

---

## 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、指定した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可能です。

このコマンドは Windows のみで使用可能です。

### 権限:

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:

- 348 ページの『CATALOG ODBC DATA SOURCE』
- 545 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。

### 権限:

以下のどれかが必要です。

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。

## 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使用可能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 検索モード - 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。

### 権限:

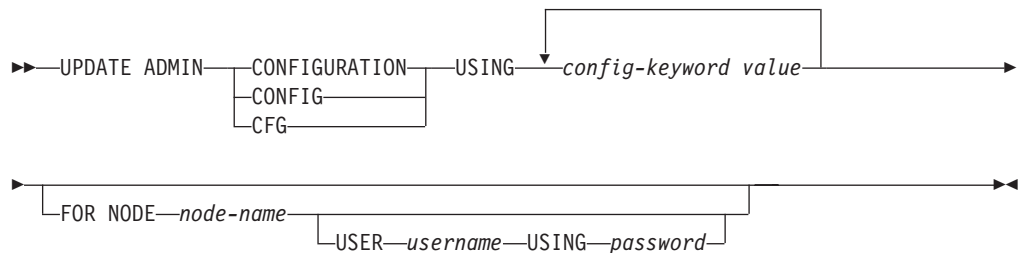
*dasadm*

### 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:

- 407 ページの『GET ADMIN CONFIGURATION』
- 685 ページの『RESET ADMIN CONFIGURATION』
- 「管理ガイド: 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。

### 権限:

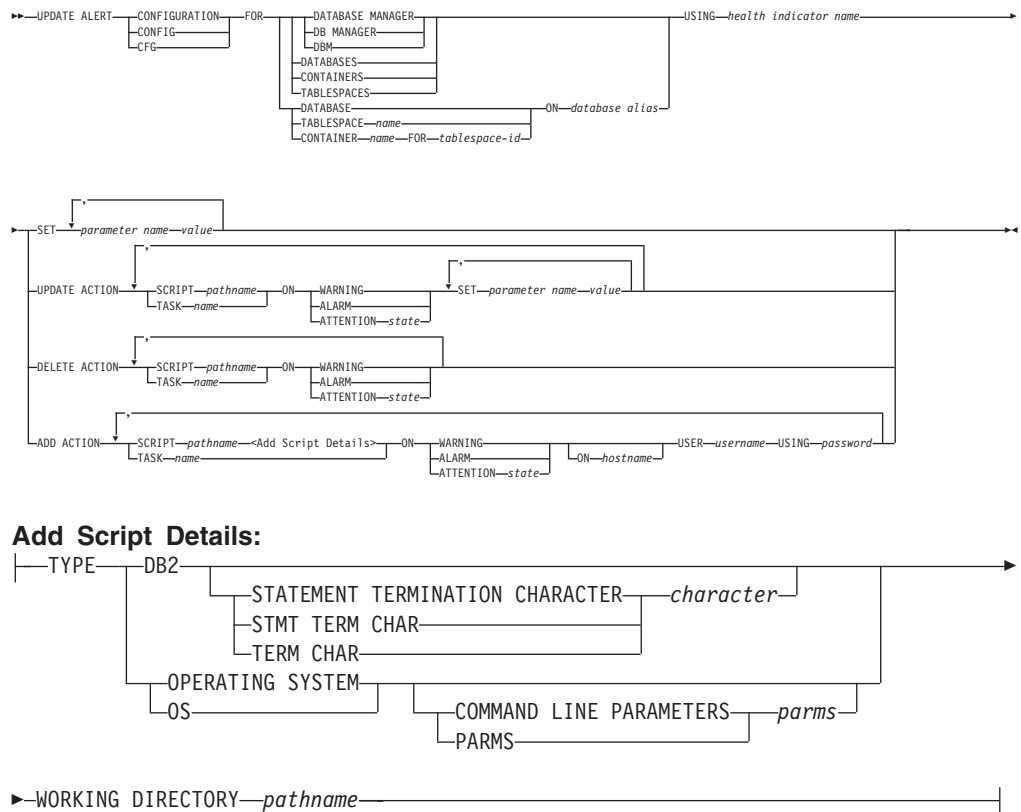
以下のどれかが必要です。

- 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。

#### 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

## UPDATE ALERT CONFIGURATION

テナー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。たとえば、次のようになります。

```
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
- WARNING
- SENSITIVITY
- ACTIONSENABLED
- THRESHOLDSCHECKED

### **UPDATE ACTION SCRIPT *pathname* ON [WARNING | ALARM | 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
- WORKINGDIR
- TERMCHAR
- CMDLINEPARMS
- USERID
- PASSWORD
- SYSTEM

**UPDATE ACTION TASK *name* ON [WARNING | ALARM | 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 | 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 | ATTENTION state]**

アラート・アクション・タスクのリストから *name* という名前のアクション・タスクを除去します。

**ADD ACTION SCRIPT *pathname* ON [WARNING | ALARM | 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*

## UPDATE ALERT CONFIGURATION

### **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** | **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。

## UPDATE ALTERNATE SERVER FOR DATABASE

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を更新します。

### 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。

### 権限:

以下のいずれかが必要です。

- *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) サポートが使用可能になっている場合、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:

- 357 ページの『CREATE DATABASE』

## UPDATE ALTERNATE SERVER FOR LDAP DATABASE

Lightweight Directory Access Protocol (LDAP) のデータベースの代替サーバーを更新します。

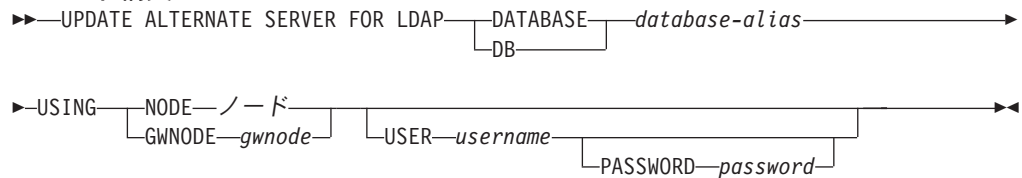
### 権限:

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・ユーザーの認証が使用されます。

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。

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。

### 関連資料:

- 337 ページの『CATALOG LDAP DATABASE』
- 126 ページの『db2ldcfg - LDAP 環境の構成』

## 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。

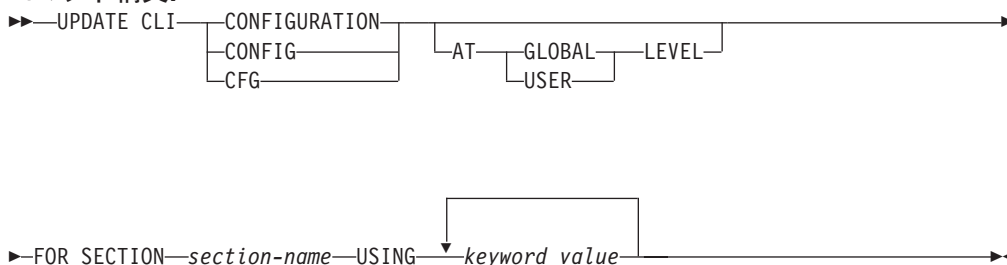
### 権限:

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。

**注:** このパラメーターを適用できるのは、LDAP サポートが使用可能な場合だけです。

#### 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。

**注:** LDAP サポートが使用可能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 サポートが使用不可能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。たとえば、次のようにします。

```
db2 update cli cfg for section tstcli1x
using TableType "'TABLE','VIEW','SYSTEM TABLE'"
```

AT USER LEVEL キーワードを指定した場合、指定されたセクションの CLI 構成パラメーターは現行ユーザーについてのみ更新されます。指定しなかった場合は、ローカル・マシン上のすべてのユーザーについて更新されます。ユーザー・レベルの CLI 構成は、LDAP ディレクトリーに保持され、ローカル・マシンでキャッシュされます。CLI 構成を読み取るとき、DB2 は常にキャッシュから読み取ります。キャッシュは、次のときに更新されます。

- ユーザーが CLI 構成を更新するとき。
- ユーザーが REFRESH LDAP コマンドを使用して、明示的に CLI 構成の最新表示を強制するとき。

LDAP 環境では、ユーザーは LDAP ディレクトリーにカタログされたデータベースに対して、デフォルト CLI 設定値のセットを構成することができます。LDAP カタログ・データベースが、DSN (データ・ソース名) として、CCA (クライアント構成アシスタント) 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:

- 416 ページの『GET CLI CONFIGURATION』
- 660 ページの『REFRESH LDAP』

784 コマンド・リファレンス

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:

- 519 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。

**権限:**

なし。

**必要な接続:**

なし。

**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。

# UPDATE CONTACTGROUP

ローカル・システムで定義される連絡先グループの属性を更新します。連絡先グループは、スケジューラーおよびヘルス・モニターから通知を受け取るユーザーのリストです。

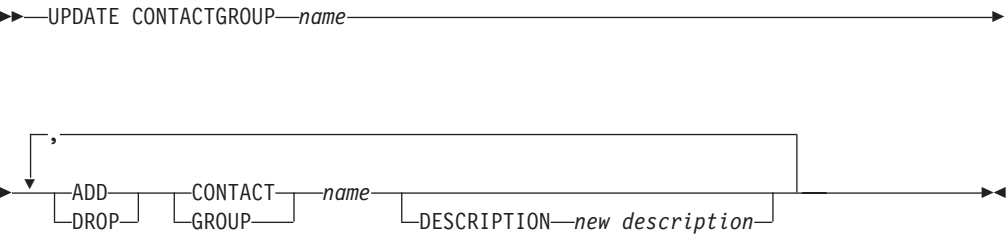
権限:

なし

必要な接続:

なし

コマンド構文:



コマンド・パラメーター:

**CONTACTGROUP** *name*  
更新される連絡先グループの名前。

**ADD CONTACT** *name*  
グループに追加される新しい連絡先の名前を指定します。グループへの追加前に、ADD CONTACT コマンドで連絡先を定義する必要はありません。

**DROP CONTACT** *name*  
グループからドロップされる、グループ中の連絡先の名前を指定します。

**ADD GROUP** *name*  
グループに追加される新しい連絡先グループの名前を指定します。

**DROP GROUP** *name*  
グループからドロップされる、連絡先グループの名前を指定します。

**DESCRIPTION** *new description*  
オプションです。連絡先グループの新しいテキスト記述。

## UPDATE DATABASE CONFIGURATION

特定のデータベース構成ファイルの中の個々の項目を修正します。

データベース構成ファイルは、データベースが作成されたノードすべてに常駐しています。

### 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。

### 権限:

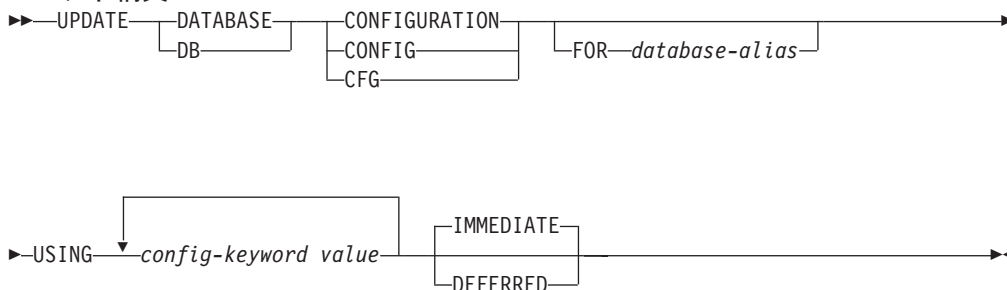
以下のいずれかが必要です。

- *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すで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。

#### IMMEDIATE

データベースが稼動している場合に、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。たとえば、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、すべてのパーティションに対して更新が発行されます。

たとえば、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"
```

UNIX オペレーティング・システムで作業を行っており、"grep" コマンドがある場合は、以下のコマンドを使用して、*logretain* 値のみを表示できます。

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

DB2 構成パラメーターと、各種データベース・ノードに使用できる値についての詳細は、個々の構成パラメーターの説明を参照してください。これらのパラメーターの値は、構成するデータベース・ノードの各タイプ (サーバー、クライアント、またはリモート・クライアントを持つサーバー) によって異なります。

すべてのパラメーターを更新できるわけではありません。

データベース構成ファイルへの変更の一部は、ファイルがメモリーにロードされた後にのみ有効になります。これを行う前にすべてのアプリケーションはデータベースから切断されている必要があります。オンラインで構成できるパラメーターと構成できないパラメーターについては、構成パラメーターの一覧をご覧ください。

## UPDATE DATABASE CONFIGURATION

たとえば、SALES データベースの *sortheap* データベース構成パラメーターをオンラインで変更するには、次のようなコマンドを入力します。

```
db2 connect to sales
db2 update db cfg using sortheap 1000
db2 connect reset
```

エラーが発生した場合、データベース構成ファイルは変更されません。チェックサムが無効な場合、データベース構成ファイルは更新できません。適当なコマンドを使用しないでデータベース構成ファイルを変更するとこれが発生することがあります。エラーが発生する場合、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:

- 「管理ガイド: 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:

- 422 ページの『GET DATABASE CONFIGURATION』
- 689 ページの『RESET DATABASE CONFIGURATION』
- 「管理ガイド: パフォーマンス」の『構成パラメーターのサマリー』



## UPDATE DATABASE MANAGER CONFIGURATION

データベース・マネージャー構成ファイルの中の個々の項目を修正します。

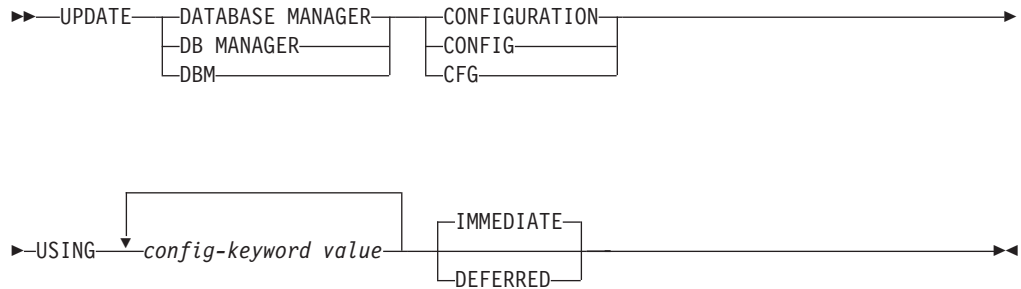
権限:

*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、データベースが再始動されるときに有効になります。

### 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。

たとえば、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**、**NNAME**、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** サポートが使用可能で、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:

- 「管理ガイド: 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:

- 428 ページの『GET DATABASE MANAGER CONFIGURATION』
- 691 ページの『RESET DATABASE MANAGER CONFIGURATION』
- 761 ページの『TERMINATE』
- 「管理ガイド: パフォーマンス」の『構成パラメーターのサマリー』

## UPDATE HEALTH NOTIFICATION CONTACT LIST

インスタンスが発行するヘルス・アラートについての通知に関して連絡先リストを更新します。

### 権限:

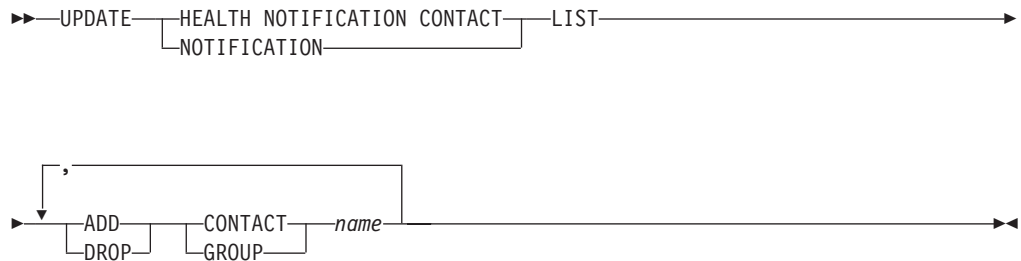
以下のどれかが必要です。

- sysadm
- sysctrl
- sysmaint

### 必要な接続:

インスタンス。明示的なアタッチは必要ありません。

### コマンド構文:



### コマンド・パラメーター:

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

インスタンスのヘルスの通知を受ける新しい連絡先グループを追加します。

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

インスタンスのヘルスの通知を受ける新しい連絡先を追加します。

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

インスタンスのヘルスの通知を受ける連絡先のリストから、連絡先グループを除去します。

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

インスタンスのヘルスの通知を受ける連絡先のリストから、連絡先を除去します。

# UPDATE HISTORY FILE

履歴ファイル項目にあるロケーション、装置タイプ、またはコメントを更新します。

**権限:**

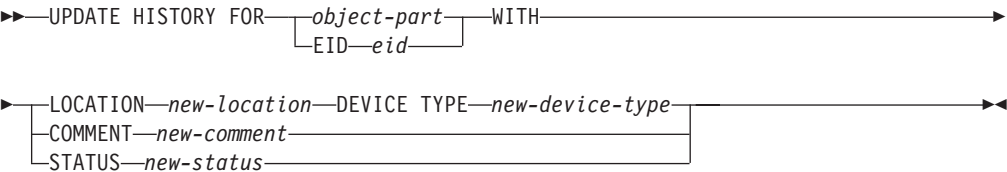
以下のいずれか。

- *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:**

**EID** *eid*

履歴項目 ID。

**FOR** *object-part*

イメージのバックアップまたはコピーの 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。

**LOCATION** *new-location*

バックアップ・イメージの新しい物理ロケーションを指定します。このパラメーターの解釈は装置タイプに依存します。

**DEVICE TYPE** *new-device-type*

バックアップ・イメージを保管する新しい装置タイプを指定します。有効な装置タイプは次のとおりです。

- |          |         |
|----------|---------|
| <b>D</b> | ディスク    |
| <b>K</b> | ディスケット  |
| <b>T</b> | テープ     |
| <b>A</b> | TSM     |
| <b>U</b> | ユーザー出口  |
| <b>P</b> | パイプ     |
| <b>N</b> | Null 装置 |
| <b>X</b> | 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。有効な値は以下のとおりです。

**A** 項目をアクティブとしてマークします。

**I** 項目を非アクティブとしてマークします。

**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 19970413100000001 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に使用されます。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から再配置されているなら、現在のロケーションを反映するよう、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。

**関連資料:**

- 630 ページの『PRUNE HISTORY/LOGFILE』

## 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。

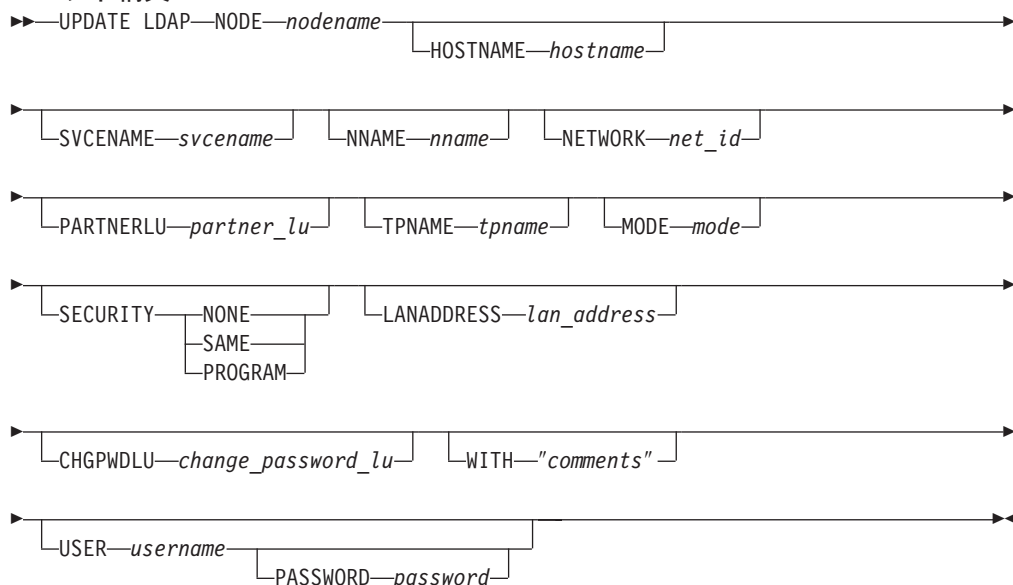
**権限:**

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*

TCP/IP ホスト名 (または IP アドレス) を指定します。

**SVCENAME** *svcename*

TCP/IP サービス名またはポート番号を指定します。

**NNAME** *nname*

NETBIOS ワークステーション名を指定します。

**注:** NETBIOS 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。ただし、このプロトコルは、Windows 以外のオペレーティング・システムを使用するワークステーションからでも更新できます。

**NETWORK** *net\_id*

APPN ネットワーク ID を指定します。

**PARTNERLU partner\_lu**

DB2 サーバー・マシン用の APPN パートナー LU 名を指定します。

**TPNAME tpname**

APPN トランザクション・プログラム名を指定します。

**MODE mode**

APPN モード名を指定します。

**SECURITY**

APPN セキュリティー・レベルを指定します。有効な値は以下のとおりです。

**NONE** サーバーに送信する割り振り要求に、セキュリティ情報が含まれないということを指定します。これは、DB2 UDB サーバーの場合のデフォルト・セキュリティです。

**SAME** サーバーに送信する割り振り要求に、ユーザー名が含まれないということを指定します。これは、ユーザー名が「すでに検査済み」という標識で指定されます。サーバーは、「すでに検査済み」という保証を受け入れられるように構成されていなければなりません。

**PROGRAM**

サーバーに送信する割り振り要求に、ユーザー名とパスワードの両方が含まれるということを指定します。これは、DB2 for OS/390 and z/OS、DB2 for iSeries などのホスト・データベース・サーバーの場合のデフォルト・セキュリティです。

**LANADDRESS lan\_address**

APPN ネットワーク・アダプター・アドレスを指定します。

**CHGPWDLU change\_password\_lu**

ホスト・データベース・サーバーのパスワード変更時に使用される、パートナー LU の名前を指定します。

**WITH "comments"**

DB2 サーバーを記述します。ネットワーク・ディレクトリーで登録されるサーバー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。注釈テキストは必ず二重引用符で囲ってください。

**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 ディレクトリー内でオブジェクトを作成したり更新したりするための権限が必要です。ユーザーの LDAP DN が指定されない場合、現行ログオン・ユーザーの認証が使用されます。

**PASSWORD password**

アカウント・パスワード。

**関連資料:**

- 661 ページの『REGISTER』
- 372 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。

**権限:**

以下のどれかが必要です。

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

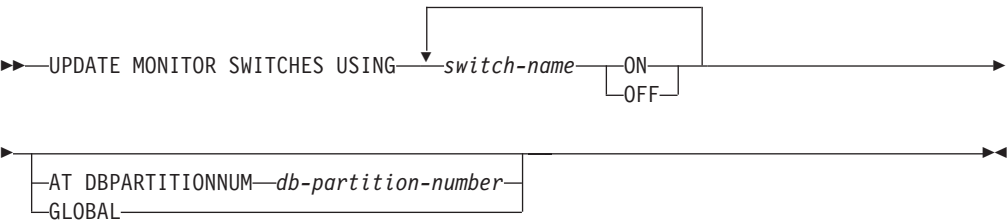
**必要な接続:**

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・アタッチが作成されます。
- 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・アタッチが使用されます。

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**

次のスイッチ名が使用可能です。

**BUFFERPOOL**

バッファ・プール活動情報

**LOCK**

ロック情報

**SORT**

ソート情報

**STATEMENT**

SQL ステートメント情報



|                  |                 |
|------------------|-----------------|
| <b>TABLE</b>     | 表活動情報           |
| <b>TIMESTAMP</b> | タイム・スタンプ情報のモニター |
| <b>UOW</b>       | 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。

**関連資料:**

- 452 ページの『**GET SNAPSHOT**』
- 443 ページの『**GET MONITOR SWITCHES**』

## UPDATE MONITOR SWITCHES

## 第 4 章 コマンド行 SQL 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、804 ページの表 23 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。

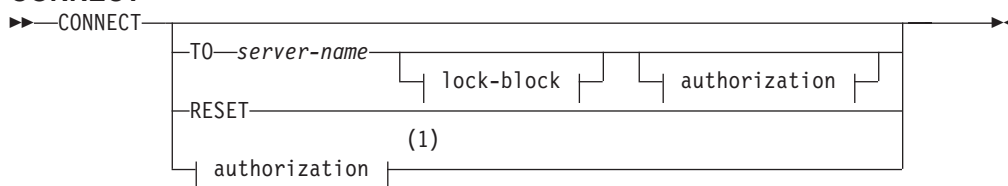
### CALL



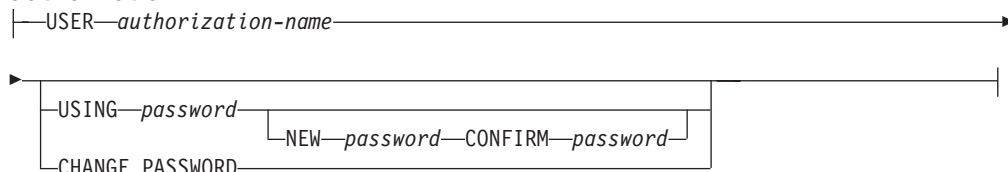
### CLOSE



### CONNECT

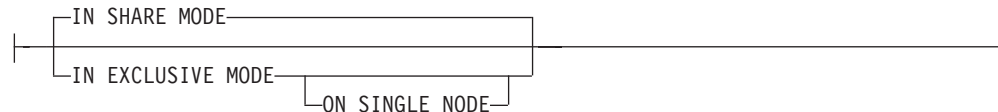


### authorization:



## コマンド行 SQL ステートメントの使用

### lock-block:



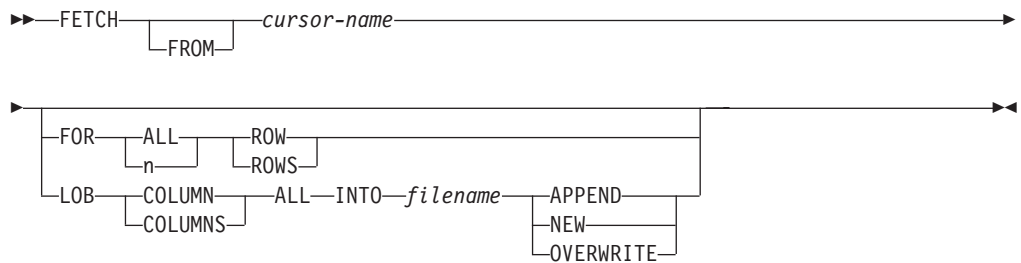
### 注:

- 1 この形式は、暗黙接続が使用可能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。
  - 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。
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. 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。
4. 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。

たとえば、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
```

5. 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。
  - 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 列の最高数は 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。
6. コマンド行プロセッサは、BLOB 列を 16 進表記で表示します。
7. 適切な変換関数を使用できない場合には、構造型列への参照を含む SQL ステートメントを発行することはできません。

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。たとえば、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。

## コマンド行 SQL ステートメントの使用

```
db2c1pcs.bnd +
db2c1prrr.bnd +
db2c1pur.bnd +
db2c1prs.bnd +
db2c1pns.bnd
```

- データベースに接続する。
- 次のコマンドを発行する。

```
db2 bind @c1p.lst collection nullid datetime iso
```

表 23. SQL ステートメント (DB2 Universal Database)

| SQL ステートメント                                                                                                                                                                                                                                                                                                                                                   | 動的 <sup>1</sup> | コマンド行<br>プロセッサ<br>ー (CLP) | コール・レベル・<br>インターフェース <sup>3</sup> (CLI)                | SQL<br>プロシ<br>ジャー |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|---------------------------|--------------------------------------------------------|-------------------|
| ALLOCATE CURSOR                                                                                                                                                                                                                                                                                                                                               |                 |                           |                                                        | X                 |
| 割り当てステートメント                                                                                                                                                                                                                                                                                                                                                   |                 |                           |                                                        | X                 |
| ASSOCIATE LOCATORS                                                                                                                                                                                                                                                                                                                                            |                 |                           |                                                        | X                 |
| ALTER { BUFFERPOOL,<br>NICKNAME, <sup>9</sup> NODEGROUP,<br>SERVER, <sup>9</sup> TABLE,<br>TABLESPACE, USER MAPPING, <sup>9</sup><br>TYPE, VIEW }                                                                                                                                                                                                             | X               | X                         | X                                                      |                   |
| BEGIN DECLARE SECTION <sup>2</sup>                                                                                                                                                                                                                                                                                                                            |                 |                           |                                                        |                   |
| CALL                                                                                                                                                                                                                                                                                                                                                          | X               | X                         | X                                                      | X                 |
| CASE ステートメント                                                                                                                                                                                                                                                                                                                                                  |                 |                           |                                                        | X                 |
| CLOSE                                                                                                                                                                                                                                                                                                                                                         |                 | X                         | SQLCloseCursor()、<br>SQLFreeStmt()                     | X                 |
| COMMENT ON                                                                                                                                                                                                                                                                                                                                                    | X               | X                         | X                                                      | X                 |
| COMMIT                                                                                                                                                                                                                                                                                                                                                        | X               | X                         | SQLEndTran()、SQLTransact()                             | X                 |
| コンパウンド SQL (組み込み)                                                                                                                                                                                                                                                                                                                                             |                 |                           | X <sup>4</sup>                                         |                   |
| コンパウンド・ステートメント                                                                                                                                                                                                                                                                                                                                                |                 |                           |                                                        | X                 |
| CONNECT (タイプ 1)                                                                                                                                                                                                                                                                                                                                               |                 | X                         | SQLBrowseConnect()、<br>SQLConnect()、SQLDriverConnect() |                   |
| CONNECT (タイプ 2)                                                                                                                                                                                                                                                                                                                                               |                 | X                         | SQLBrowseConnect()、<br>SQLConnect()、SQLDriverConnect() |                   |
| CREATE { ALIAS, BUFFERPOOL,<br>DISTINCT TYPE,<br>EVENT MONITOR, FUNCTION,<br>FUNCTION MAPPING, <sup>9</sup> INDEX,<br>INDEX EXTENSION, METHOD,<br>NICKNAME, <sup>9</sup> NODEGROUP,<br>PROCEDURE, SCHEMA, SERVER,<br>TABLE, TABLESPACE,<br>TRANSFORM, TYPE MAPPING, <sup>9</sup><br>TRIGGER, USER MAPPING, <sup>9</sup> TYPE,<br>VIEW, WRAPPER <sup>9</sup> } | X               | X                         | X                                                      | X <sup>10</sup>   |
| DECLARE CURSOR <sup>2</sup>                                                                                                                                                                                                                                                                                                                                   |                 | X                         | SQLAllocStmt()                                         | X                 |
| DECLARE GLOBAL TEMPORARY<br>TABLE                                                                                                                                                                                                                                                                                                                             | X               | X                         | X                                                      | X                 |

表 23. SQL ステートメント (DB2 Universal Database) (続き)

| SQL ステートメント                      | 動的 <sup>1</sup> | コマンド行<br>プロセッサ<br>ー (CLP) | コール・レベル・<br>インターフェース <sup>3</sup> (CLI)                                     | SQL<br>プロシージャ<br>ー |
|----------------------------------|-----------------|---------------------------|-----------------------------------------------------------------------------|--------------------|
| DELETE                           | X               | X                         | X                                                                           | X                  |
| DESCRIBE <sup>8</sup>            |                 | X                         | SQLColAttributes(),<br>SQLDescribeCol(),<br>SQLDescribeParam() <sup>6</sup> |                    |
| DISCONNECT                       |                 | X                         | SQLDisconnect()                                                             |                    |
| DROP                             | X               | X                         | X                                                                           | X <sup>10</sup>    |
| END DECLARE SECTION <sup>2</sup> |                 |                           |                                                                             |                    |
| EXECUTE                          |                 |                           | SQLExecute()                                                                | X                  |
| EXECUTE IMMEDIATE                |                 |                           | SQLExecDirect()                                                             | X                  |
| EXPLAIN                          | X               | X                         | X                                                                           | X                  |
| FETCH                            |                 | X                         | SQLExtendedFetch(), SQLFetch(),<br>SQLFetchScroll()                         | X                  |
| FLUSH EVENT MONITOR              | X               | X                         | X                                                                           |                    |
| FOR ステートメント                      |                 |                           |                                                                             | X                  |
| FREE LOCATOR                     |                 |                           | X <sup>4</sup>                                                              | X                  |
| GET DIAGNOSTICS                  |                 |                           |                                                                             | X                  |
| GOTO ステートメント                     |                 |                           |                                                                             | X                  |
| GRANT                            | X               | X                         | X                                                                           | X                  |
| IF ステートメント                       |                 |                           |                                                                             | X                  |
| INCLUDE <sup>2</sup>             |                 |                           |                                                                             |                    |
| INSERT                           | X               | X                         | X                                                                           | X                  |
| ITERATE                          |                 |                           |                                                                             | X                  |
| LEAVE ステートメント                    |                 |                           |                                                                             | X                  |
| LOCK TABLE                       | X               | X                         | X                                                                           | X                  |
| LOOP ステートメント                     |                 |                           |                                                                             | X                  |
| OPEN                             |                 | X                         | SQLExecute(), SQLExecDirect()                                               | X                  |
| PREPARE                          |                 |                           | SQLPrepare()                                                                | X                  |
| REFRESH TABLE                    | X               | X                         | X                                                                           |                    |
| RELEASE                          |                 | X                         |                                                                             | X                  |
| RELEASE SAVEPOINT                | X               | X                         | X                                                                           | X                  |
| RENAME TABLE                     | X               | X                         | X                                                                           |                    |
| RENAME TABLESPACE                | X               | X                         | X                                                                           |                    |
| REPEAT ステートメント                   |                 |                           |                                                                             | X                  |
| RESIGNAL ステートメント                 |                 |                           |                                                                             | X                  |
| RETURN ステートメント                   |                 |                           |                                                                             | X                  |
| REVOKE                           | X               | X                         | X                                                                           |                    |
| ROLLBACK                         | X               | X                         | SQLEndTran(), SQLTransact()                                                 | X                  |
| SAVEPOINT                        | X               | X                         | X                                                                           | X                  |
| select-statement                 | X               | X                         | X                                                                           | X                  |

## コマンド行 SQL ステートメントの使用

表 23. SQL ステートメント (DB2 Universal Database) (続き)

| SQL ステートメント                            | 動的 <sup>1</sup> | コマンド行<br>プロセッサ<br>ー (CLP) | コール・レベル・<br>インターフェース <sup>3</sup> (CLI) | SQL<br>プロシージャ<br>ー |
|----------------------------------------|-----------------|---------------------------|-----------------------------------------|--------------------|
| SELECT INTO                            |                 |                           |                                         | X                  |
| SET CONNECTION                         |                 | X                         | SQLSetConnection()                      |                    |
| SET CURRENT DEFAULT<br>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<br>SNAPSHOT        | X               | X                         | X、SQLSetConnectAttr()                   | X                  |
| SET CURRENT PACKAGESET                 |                 |                           |                                         |                    |
| SET CURRENT QUERY<br>OPTIMIZATION      | X               | X                         | X                                       | X                  |
| SET CURRENT REFRESH AGE                | X               | X                         | X                                       | X                  |
| SET EVENT MONITOR STATE                | X               | X                         | X                                       | X                  |
| SET INTEGRITY                          | X               | X                         | X                                       |                    |
| SET PASSTHRU <sup>9</sup>              | X               | X                         | X                                       | X                  |
| SET PATH                               | X               | X                         | X                                       | X                  |
| SET SCHEMA                             | X               | X                         | X                                       | X                  |
| SET SERVER OPTION <sup>9</sup>         | X               | X                         | X                                       | X                  |
| SET transition-variable <sup>5</sup>   | X               | X                         | X                                       | X                  |
| SIGNAL ステートメント                         |                 |                           |                                         | X                  |
| SIGNAL SQLSTATE <sup>5</sup>           | X               | X                         | X                                       |                    |
| UPDATE                                 | X               | X                         | X                                       | X                  |
| VALUES INTO                            |                 |                           |                                         | X                  |
| WHENEVER <sup>2</sup>                  |                 |                           |                                         |                    |
| WHILE ステートメント                          |                 |                           |                                         | X                  |

**注:**

- このリストのすべてのステートメントは静的 SQL としてコーディングできますが、動的 SQL としてコーディングできるのは X になっているステートメントだけです。
- このステートメントは実行できません。
- X は、該当するステートメントが SQLExecDirect() または SQLPrepare() と SQLExecute() のどちらによっても実行できるという意味です。同等の DB2 CLI 機能がある場合は、機能名がリストされています。
- このステートメントは動的ではないものの、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。
- CREATE TRIGGER ステートメント内だけで使用できます。
- SQL DESCRIBE ステートメントで出力の記述ができますが、DB2 CLI を使用すると、入力記述も可能となります (SQLDescribeParam() 関数を使用する場合)。
- SQL FETCH ステートメントで 1 つの行を一度に 1 方向に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。
- DESCRIBE SQL ステートメントの構文は、CLP DESCRIBE コマンドの構文と異なります。
- ステートメントは、フェデレーテッド・データベース・サーバーでのみサポートされます。
- SQL プロシージャは、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『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『CONNECT (タイプ 2) 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. 構文図の読み方

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。

構文図は、水平線（主線）に沿って左から右、上から下へ読みます。線が矢印で終わっている場合、コマンド構文は継続しており、次の線は矢印で始まります。垂直線は、コマンド構文の終わりを示します。

構文図から情報を入力する際には、引用符や等号などの句読点を必ず使用するようになしてください。

パラメーターは、キーワードと変数に分類されます。

- キーワードは定数を表し、大文字で示されます。実際のコマンド・プロンプトでは、キーワードを大文字、小文字、または大文字と小文字の混合で入力することができます。キーワードの例としてはコマンド名があります。
- 変数は、ユーザーによって提供される名前または値を表し、小文字で示されます。実際のコマンド・プロンプトでは、変数を大文字、小文字、または大文字と小文字の混合で入力することができます（大文字小文字の制限が明示されているのでない限り）。変数の例としては、ファイル名があります。

1 つのパラメーターがキーワードと変数の組み合わせである場合があります。

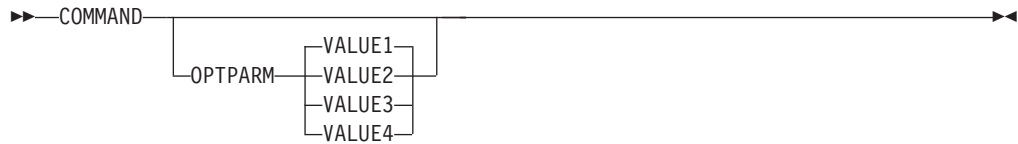
必須パラメーターは、主線と同じ高さに示されます。

▶▶—COMMAND—*required parameter*—▶▶

オプション・パラメーターは、主線の下側に示されます。

▶▶—COMMAND—  
          └*optional parameter*┘—▶▶

パラメーターのデフォルト値は、線の上側に示されます。



パラメーターのスタックで、最初のパラメーターが主線と同じ高さに示されている場合は、パラメーターの 1 つを必ず選択しなければならないことを示します。



パラメーターのスタックで、最初のパラメーターが主線の下側に示されている場合は、パラメーターの 1 つを選択可能であることを示します。

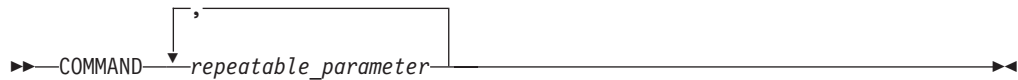


線の上側で左に戻る矢印がある場合は、項目が以下の規則に従って繰り返し可能であることを示します。

- 矢印が中断されていない場合、項目はブランク・スペースによって区切られたリストの形式で繰り返すことができます。

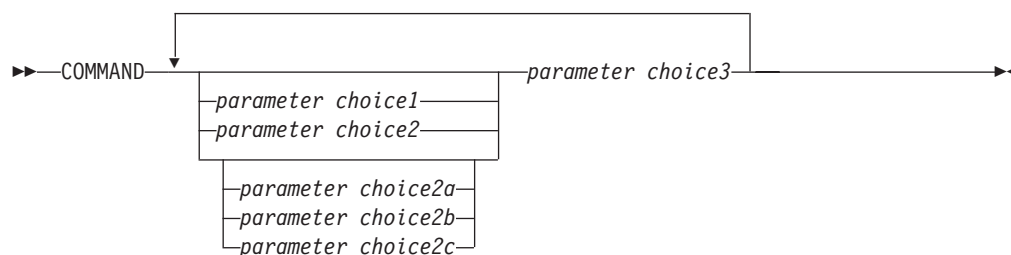


- 矢印にコンマが使用されている場合、項目はコンマによって区切られたリストの形式で繰り返すことができます。



パラメーター・スタックの項目は、前述の必須およびオプション・パラメーターのスタック規則に従って繰り返すことができます。

一部の構文図では、パラメーター・スタックが他のパラメーター・スタックの中に収められていることがあります。スタックの項目を繰り返すには、前述の規則に従わなければなりません。つまり、内側のスタックの上に繰り返し矢印がなく、外側のスタックの上にはある場合には、内側のスタックから 1 つのパラメーターのみを選択し、外側のスタックからの任意のパラメーターと組み合わせて、その組み合わせを繰り返すことができます。たとえば、次の図では、パラメーター *choice2a* をパラメーター *choice2* と組み合わせて、その組み合わせ (*choice2* と *choice2a*) を繰り返すことができます。

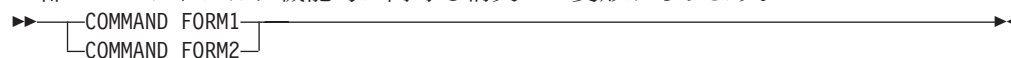


一部のコマンドの前には、オプションのパス・パラメーターが付けられています。



このパラメーターを指定しない場合、システムはコマンドを現行ディレクトリーから検索します。コマンドが見つからない場合、システムは、`.profile` にリストされているすべてのディレクトリーからコマンドの検索を続行します。

一部のコマンドには、機能的に同等な構文上の変形があります。





## 付録 B. 命名規則

データベースや表、認証 ID 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適用される規則について説明します。

- 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・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で入力するときに区切り文字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。
- 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ユニークな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 文字のユニーク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) はユニークなユーザー 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。

- 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。
- ノード・ディレクトリーでカタログ化されるリモート・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。





---

## 付録 C. DB2 Universal Database の技術情報

---

### DB2 資料とヘルプ

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
- ダウンロード可能な PDF ファイル、CD 上の PDF ファイル、および印刷された資料
  - ガイド
  - リファレンス・マニュアル
- コマンド行ヘルプ
  - コマンド・ヘルプ
  - メッセージ・ヘルプ
  - SQL 状態ヘルプ
- インストール済みソース・コード
  - サンプル・プログラム

ibm.com® にある技術資料、白書、Redbooks™ その他の DB2 Universal Database™ 技術情報にオンライン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 ([www.ibm.com/software/data/pubs/](http://www.ibm.com/software/data/pubs/)) にアクセスしてください。

### DB2 資料の更新

IBM® は、DB2 インフォメーション・センターの資料のフィックスパックやその他の資料更新を定期的に発行しています。DB2 インフォメーション・センター (<http://publib.boulder.ibm.com/infocenter/db2help/>) にアクセスすれば、常に最新の情報が掲載されます。DB2 インフォメーション・センターをローカル・インストールしている場合、更新記事を表示するには、まず手動で更新をインストールしてください。新しい情報が発表されたときに資料を更新することにより、DB2 インフォメーション・センター CD からインストールした情報を更新することができます。

インフォメーション・センターの方が、PDF 資料やハードコピー資料よりも頻繁に更新されます。DB2 の最新の技術情報を入手するには、資料更新が発行されたときにそれをインストールするか、または [www.ibm.com](http://www.ibm.com) サイトの DB2 インフォメーション・センターにアクセスしてください。

#### 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『CLI サンプル・プログラム』

- 「アプリケーション開発ガイド アプリケーションの構築および実行」の『Java サンプル・プログラム』
- 816 ページの『DB2 インフォメーション・センター』

#### 関連タスク:

- 836 ページの『DB2 ツールからコンテキスト・ヘルプを呼び出す』
- 827 ページの『コンピューターまたはイントラネット・サーバーへの DB2 インフォメーション・センターの更新インストール』
- 838 ページの『コマンド行プロセッサからメッセージ・ヘルプを呼び出す』
- 838 ページの『コマンド行プロセッサからコマンド・ヘルプを呼び出す』
- 839 ページの『コマンド行プロセッサから SQL 状態ヘルプを呼び出す』

#### 関連資料:

- 829 ページの『DB2 PDF 資料および印刷された資料』

---

## DB2 インフォメーション・センター

DB2<sup>®</sup> インフォメーション・センターを使用すると、DB2 Universal Database<sup>™</sup>、DB2 Connect<sup>™</sup>、DB2 Information Integrator および DB2 Query Patroller<sup>™</sup>などのDB2 ファミリー製品を最大限に活用するのに必要なすべての情報にアクセスできます。また、DB2 インフォメーション・センターは、DB2 の主な機能とコンポーネントに関する情報を提供します (レプリケーション、データウェアハウジング、および DB2 の種々の Extender など)。

Mozilla 1.0 以上または Microsoft<sup>®</sup> Internet Explorer 5.5 以上で表示する場合、DB2 インフォメーション・センターには以下の機能があります。以下のいくつかの機能では、JavaScript<sup>™</sup> のサポートを使用可能にする必要があります:

#### 柔軟なインストール・オプション

以下の中から、ご使用の環境に最も適したオプションを使って DB2 資料を表示できます。

- 最新の資料を常に自動的に利用できるようにするには、IBM<sup>®</sup> の Web サイト (<http://publib.boulder.ibm.com/infocenter/db2help/>) にある DB2 インフォメーション・センターからすべての資料に直接アクセスします。
- 更新処理を最小化し、イントラネット内のネットワーク・トラフィックだけに制限するには、イントラネット上の 1 つのサーバーに DB2 資料をインストールします。
- 柔軟性を改善し、ネットワーク接続への依存を軽減するには、個々のコンピューターに DB2 資料をインストールします。

**検索** 「検索」テキスト・フィールドに検索語を入力することにより、DB2 インフォメーション・センターのすべてのトピックを検索できます。複数の語句を引用符で囲めば、完全一致を検索できます。また、ワイルドカード演算子 (\*、?) とブール演算子 (AND、NOT、OR) を使用して検索を絞り込むことができます。

#### タスク指向の目次

単一の目次の中から、DB2 資料のトピックを見付けることができます。目

次は、主に実行するタスクの種類に従って編成されていますが、そのほかに製品概要、特定のゴール (目的) の情報、参照情報、索引、および用語集も含まれます。

- 製品概要では、DB2 ファミリーで使用可能な製品間の関係、そうした各製品で提供される機能、および各製品の最新リリース情報について説明されています。
- インストール、管理および開発などのゴール・カテゴリーには、タスクを迅速に完了し、そのための背景情報をよく理解できるようにするトピックが含まれています。
- 「参照」トピックでは、その対象に関する詳細な情報 (ステートメントとコマンドの構文、メッセージ・ヘルプ、構成パラメーターなど) が説明されています。

#### 現在のトピックを目次に表示する

現在のトピックが目次のどの部分に該当するかを表示するには、目次フレーム内の「リフレッシュ/現在のトピックの表示 (Refresh/Show Current Topic)」ボタンをクリックするか、コンテンツ・フレーム内の「目次に表示 (Show in Table of Contents)」ボタンをクリックします。幾つかのファイルで関連トピックへの複数のリンクをたどった場合、または検索結果からトピックにアクセスした場合には、この機能が役立ちます。

**索引** 索引から、すべての資料にアクセスすることができます。索引では、用語が 50 音順に編成されています。

**用語集** 用語集を見れば、DB2 資料で使われているさまざまな用語の定義を調べることができます。用語集では、用語が 50 音順に編成されています。

#### 組み込まれているローカライズ情報

DB2 インフォメーション・センターは、ブラウザで設定された言語でトピックを表示します。設定された言語のトピックが利用できない場合、DB2 インフォメーション・センターにはそのトピックの英語版が表示されます。

iSeries™ 技術情報については、IBM eServer™ iSeries Information Center ([www.ibm.com/eserver/iserics/infocenter/](http://www.ibm.com/eserver/iserics/infocenter/)) を参照してください。

#### 関連概念:

- 818 ページの『DB2 インフォメーション・センターのインストール・シナリオ』

#### 関連タスク:

- 827 ページの『コンピューターまたはイントラネット・サーバーへの DB2 インフォメーション・センターの更新インストール』
- 828 ページの『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』
- 826 ページの『DB2 インフォメーション・センターの呼び出し』
- 820 ページの『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 (UNIX)』
- 823 ページの『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 インフォメーション・センターのインストール・シナリオ

さまざまに異なる業務環境のもとでは、DB2<sup>®</sup> 情報にどのようにアクセスするかの要件もそれぞれ異なります。DB2 インフォメーション・センターにアクセスするには、IBM<sup>®</sup> の Web サイト、サーバーまたは組織のネットワーク、あるいはコンピューターへのインストールという 3 つの方法が可能です。この 3 つのケースのいずれも、資料は DB2 インフォメーション・センター内に置かれます。インフォメーション・センターは、ブラウザを使って表示できるように設計されたトピック・ベースの情報の Web サイトです。デフォルトでは、DB2 製品から、IBM Web サイト上の DB2 インフォメーション・センターにアクセスします。これに対して、イントラネット・サーバーまたはご自分のコンピューターから DB2 インフォメーション・センターにアクセスしたい場合、製品メディア・パック内にある DB2 インフォメーション・センター CD から DB2 インフォメーション・センターをインストールする必要があります。以下では、DB2 資料へのアクセス・オプションの要約、および 3 つのインストール・シナリオを示します。これを参考にして、お客様の業務環境で DB2 インフォメーション・センターにアクセスするにはどの方法が最適か、どのようなインストール上の問題に配慮する必要があるかを判別してください。

### DB2 資料にアクセスするオプションの要約:

以下の表は、お客様の実際の業務環境で、DB2 インフォメーション・センターの DB2 製品情報にアクセスする方法としてどんなオプションが推奨されるかを示します。

| インターネット・アクセス | イントラネット・アクセス | 推奨されるアクション                                                                                |
|--------------|--------------|-------------------------------------------------------------------------------------------|
| はい           | はい           | IBM Web サイト上の DB2 インフォメーション・センターへのアクセス、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・センターへのアクセス |
| はい           | いいえ          | IBM Web サイト上の DB2 インフォメーション・センターへのアクセス                                                    |
| いいえ          | はい           | 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・センターへのアクセス                                           |
| いいえ          | いいえ          | ローカル・コンピューター上の DB2 インフォメーション・センターへのアクセス                                                   |

### シナリオ: コンピューター上の DB2 インフォメーション・センターへのアクセス:

Tsu-Chen 氏は小さな町で工場を経営していますが、その町には、インターネット・アクセスを提供する地元のインターネット・サービス・プロバイダーがありません。彼は、在庫、製品オーダー、銀行口座情報、および営業経費を管理するために DB2 Universal Database<sup>™</sup> を購入しました。Tsu-Chen 氏は以前に DB2 製品を利用したことがないので、DB2 の使用方法を習得するために、DB2 製品資料を参照する必要があります。

Tsu-Chen 氏は 標準インストール・オプションを使って DB2 Universal Database を自分のコンピューターにインストールした後、DB2 資料にアクセスしようとしてみず。しかし、開こうとしているページが見つからないというエラー・メッセージがブラウザから通知されました。Tsu-Chen 氏は DB2 製品のインストール・マニュアルを調べた結果、DB2 資料を自分のコンピューター上で利用するには、DB2 インフォメーション・センターをインストールしなければならないことに気がきます。そしてメディア・パックの中にあった DB2 インフォメーション・センター CD を見つけ出して、インストールしました。

これで、Tsu-Chen 氏はオペレーティング・システムのアプリケーション・ランチャーから DB2 インフォメーション・センターにアクセスできるようになり、より良い業務成果をあげるために DB2 製品を利用する方法を習得できます。

#### シナリオ: IBM Web サイト上の DB2 インフォメーション・センターへのアクセス:

Colin は、あるセミナー企業に所属する情報技術コンサルタントです。彼の専門はデータベース・テクノロジーおよび SQL で、DB2 Universal Database を使って北米一帯の企業を対象にこれらの科目のセミナーを開催しています。Colin のセミナーでは、教材として DB2 資料も使用されます。たとえば、SQL の講習コースでは、データベース照会の基本構文と拡張構文を教えるために SQL に関する DB2 資料が使用されます。

Colin が教えている企業の大半はインターネット・アクセスを配備しています。このような状況から判断して、Colin は、最新バージョンの DB2 Universal Database を自分のモバイル・コンピューターにインストールしたとき、IBM Web サイト上の DB2 インフォメーション・センターにアクセスするよう構成しました。この構成によって、Colin はセミナーで教えるときに最新の DB2 資料にオンライン・アクセスすることができます。

しかし、時折、Colin は移動中にインターネット・アクセスを利用できないことがあります。これは問題となります。担任するセミナーの準備のために DB2 資料にアクセスする必要のある場合には、とくにそうです。このような事態が起きないようにするために、Colin は自分のモバイル・コンピューターに DB2 インフォメーション・センターのコピーをインストールしました。

こうして、Colin は常に DB2 資料のコピーを自在に活用できるようになりました。**db2set** コマンドを使って自分のモバイル・コンピューターのレジストリー変数を簡単に構成し、どこにいるかに応じて、IBM Web サイトまたは自分のモバイル・コンピューターから DB2 インフォメーション・センターにアクセスできます。

#### シナリオ: イン트라ネット・サーバー上の DB2 インフォメーション・センターへのアクセス:

Eva は、生命保険会社のデータベース上級管理者です。彼女は管理業務の一環として、会社の UNIX<sup>®</sup> データベース・サーバーに最新バージョンの DB2 Universal Database をインストールおよび構成します。彼女の会社は最近、セキュリティ上の理由から、インターネット・アクセスをもはや業務で利用できないようにすると社員に通知しました。同社はネットワーク環境を装備しているため、Eva は DB2 インフォメーション・センターのコピーをイントラネット・サーバー上にインストール

ールして、社内のデータウェアハウスを定期的に利用するすべての社員（営業担当者、営業部長、および業務分析担当者）から DB2 資料へのアクセスを可能にすることにしました。

Eva は、応答ファイルを使って全社員のコンピューター上に最新バージョンの DB2 Universal Database をインストールするようデータベース・チームに指示します。その際、イントラネット・サーバーのホスト名とポート番号を使って DB2 インフォメーション・センターにアクセスできるよう、確実に各コンピューターを構成します。

しかし、Eva のチームの下級データベース管理者である Migual の誤解によって、数人の社員のコンピューター上で、イントラネット・サーバーの DB2 インフォメーション・センターにアクセスするよう DB2 Universal Database を構成する代わりに、DB2 インフォメーション・センターのコピーをそれらのコンピューターにインストールしてしまいました。これを訂正するために、Eva は、**db2set** コマンドを使ってこれらのコンピューター上の DB2 インフォメーション・センターのレジストリー変数（ホスト名は DB2\_DOCHOST、ポート番号は DB2\_DOCPORT）を変更するよう Migual に指示しました。これで、ネットワーク上の適切なすべてのコンピューターが DB2 インフォメーション・センターにアクセスできるようになり、社員は DB2 に関する質問の答えを DB2 資料から見つけることができます。

#### 関連概念:

- 816 ページの『DB2 インフォメーション・センター』

#### 関連タスク:

- 827 ページの『コンピューターまたはイントラネット・サーバーへの DB2 インフォメーション・センターの更新インストール』
- 820 ページの『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 (UNIX)』
- 823 ページの『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 インフォメーション・センターへのアクセスのロケーションの設定: Common GUI help』

#### 関連資料:

- 216 ページの『db2set - DB2 プロファイル・レジストリー』

---

## 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 (UNIX)

DB2 製品資料にアクセスする方法として、IBM Web サイト、イントラネット・サーバー、またはコンピューターにインストールしたバージョンの 3 つがあります。デフォルトでは、DB2 製品は IBM Web サイト上の DB2 資料にアクセスします。イントラネット・サーバーまたはコンピューター上の DB2 資料にアクセスしたい場合には、DB2 インフォメーション・センター CD から資料をインストールする必要があります。DB2 セットアップ・ウィザードを使用すれば、インストール設定を定義し、UNIX オペレーティング・システムを使用するコンピューターに DB2 インフォメーション・センターをインストールできます。



## 前提条件:

このセクションでは、UNIX コンピューターに DB2 インフォメーション・センターをインストールするためのハードウェア、オペレーティング・システム、ソフトウェア、および通信の諸要件を一覧で示します。

### • ハードウェア要件

以下のいずれかのプロセッサが必要です。

- PowerPC (AIX)
- HP 9000 (HP-UX)
- Intel 32 ビット (Linux)
- Solaris UltraSPARC コンピューター (Solaris オペレーティング環境)

### • オペレーティング・システム要件

以下のいずれかのオペレーティング・システムが必要です。

- IBM AIX 5.1 (PowerPC 上)
- HP-UX 11i (HP 9000 上)
- Red Hat Linux 8.0 (Intel 32 ビット上)
- SuSE Linux 8.1 (Intel 32 ビット上)
- Sun Solaris バージョン 8 (Solaris オペレーティング環境の UltraSPARC コンピューター上)

**注:** DB2 インフォメーション・センターは、DB2 クライアントをサポートする UNIX オペレーティング・システム上で稼動します。このため、IBM Web サイトから DB2 インフォメーション・センターにアクセスするか、イントラネット・サーバーに DB2 インフォメーション・センターをインストールしてそれにアクセスすることをお勧めします。

### • ソフトウェア要件

– 以下のブラウザがサポートされています。

- Mozilla バージョン 1.0 以上

- DB2 セットアップ・ウィザードは、グラフィック・インストーラーです。ご使用のマシンで DB2 セットアップ・ウィザードのグラフィカル・ユーザー・インターフェースを表示可能にする X Window システム・ソフトウェアをインプリメントする必要があります。DB2 セットアップ・ウィザードを実行する前に、ディスプレイを正しくエクスポートしたことを確認してください。たとえば、コマンド・プロンプトで

```
export DISPLAY=9.26.163.144:0.
```

というコマンドを入力します。

### • 通信要件

- TCP/IP

## 手順:

DB2 セットアップ・ウィザードを使用して DB2 インフォメーション・センターをインストールするには、以下のようになります。

1. システムにログオンします。

2. DB2 インフォメーション・センター製品 CD を挿入してシステムにマウントします。
3. 次のコマンドを入力して、CD がマウントされているディレクトリーに移動します。

```
cd /cd
```

/cd は、CD のマウント・ポイントを表します。

4. **/db2setup** コマンドを入力して、DB2 セットアップ・ウィザードを開始します。
5. IBM DB2 セットアップ・ランチパッドが開きます。DB2 インフォメーション・センターのインストールに直接進むには、「**製品のインストール**」をクリックします。残りのステップについて説明しているオンライン・ヘルプを利用できます。オンライン・ヘルプを呼び出すには、「**ヘルプ**」をクリックします。「**キャンセル**」をクリックすれば、いつでもインストールを終了できます。
6. 「**インストールしたい製品を選択します**」ページでは、「**次へ**」をクリックします。
7. 「**DB2 セットアップ・ウィザードによるこそ (Welcome to the DB2 Setup wizard)**」ページで、「**次へ**」をクリックします。DB2 セットアップ・ウィザードは、プログラムのセットアップ操作を案内します。
8. インストールを続行するには、使用許諾条件に同意する必要があります。「**ご使用条件**」ページで、「**ご使用条件に同意します (I accept the terms in the license agreement)**」をクリックして、「**次へ**」をクリックします。
9. 「**インストール・アクションの選択**」で、「**このコンピューターに DB2 インフォメーション・センターをインストールする (Install DB2 Information Center on this computer)**」を選択します。応答ファイルを使用して、このコンピューターまたは他のコンピューターに DB2 インフォメーション・センターをあとでインストールしたい場合には、「**設定を応答ファイルに保管する**」を選択します。「**次へ**」をクリックします。
10. 「**インストールする言語の選択**」ページでは、DB2 インフォメーション・センターをインストールする言語を選択します。「**次へ**」をクリックします。
11. 「**DB2 インフォメーション・センター・ポートの指定**」ページでは、DB2 インフォメーション・センターへの着信通信を構成します。「**次へ**」をクリックしてインストールを続けます。
12. 「**ファイルのコピーの開始**」ページでは、インストールの選択項目を確認します。設定を変更するには、「**戻る**」をクリックします。「**インストール**」をクリックすると、DB2 インフォメーション・センターのファイルがコンピューターにコピーされます。

このほか、応答ファイルを使って DB2 インフォメーション・センターをインストールすることもできます。

インストール・ログ db2setup.his、db2setup.log、および db2setup.err は、デフォルトでは /tmp ディレクトリーに置かれます。

db2setup.log ファイルは、エラーも含めた DB2 製品のインストール情報をすべてキャプチャーします。db2setup.his ファイルは、コンピューター上の DB2 製品



インストール内容をすべて記録します。 DB2 は、db2setup.log ファイルを db2setup.his に付加します。 db2setup.err ファイルは、 Java から戻されるすべてのエラー出力 (例外やトラップの情報など) をキャプチャーします。

インストールが完了したら、ご使用の UNIX オペレーティング・システムに応じて、 DB2 は以下のいずれかのディレクトリーにインストールされます。

- AIX: /usr/opt/db2\_08\_01
- HP-UX: /opt/IBM/db2/V8.1
- Linux: /opt/IBM/db2/V8.1
- Solaris オペレーティング環境: /opt/IBM/db2/V8.1

#### 関連概念:

- 816 ページの『DB2 インフォメーション・センター』
- 818 ページの『DB2 インフォメーション・センターのインストール・シナリオ』

#### 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 のインストール (UNIX)』
- 827 ページの『コンピューターまたはイントラネット・サーバーへの DB2 インフォメーション・センターの更新インストール』
- 828 ページの『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』
- 826 ページの『DB2 インフォメーション・センターの呼び出し』
- 823 ページの『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 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 製品資料にアクセスする方法として、 IBM Web サイト、イントラネット・サーバー、またはコンピューターにインストールしたバージョンの 3 つがあります。デフォルトでは、DB2 製品は IBM Web サイト上の DB2 資料にアクセスします。イントラネット・サーバーまたはコンピューター上の DB2 資料にアクセスしたい場合には、 DB2 インフォメーション・センター CD から DB2 資料をインストールする必要があります。 DB2 セットアップ・ウィザードを使用すれば、インストール設定を定義し、 Windows オペレーティング・システムを使用するコンピューターに DB2 インフォメーション・センターをインストールできます。

#### 前提条件:

このセクションでは、Windows に DB2 インフォメーション・センターをインストールするためのハードウェア、オペレーティング・システム、ソフトウェア、および通信の諸要件を一覧で示します。

#### • ハードウェア要件

以下のいずれかのプロセッサが必要です。

- 32 ビット・コンピューター: Pentium または Pentium 互換の CPU

## • オペレーティング・システム要件

以下のいずれかのオペレーティング・システムが必要です。

- Windows 2000
- Windows XP

**注:** DB2 インフォメーション・センターは、DB2 クライアントをサポートする Windows オペレーティング・システム上で稼動します。このため、IBM Web サイトの DB2 インフォメーション・センターにアクセスするか、イントラ ネット・サーバーに DB2 インフォメーション・センターをインストールしてそれにアクセスすることをお勧めします。

## • ソフトウェア要件

- 以下のブラウザーがサポートされています。
  - Mozilla 1.0 以上
  - Internet Explorer バージョン 5.5 または 6.0 (Windows XP の場合はバージョン 6.0)

## • 通信要件

- TCP/IP

### 制約事項:

- DB2 インフォメーション・センターをインストールするには、管理権限をもつアカウントが必要です。

### 手順:

DB2 セットアップ・ウィザードを使用して DB2 インフォメーション・センターをインストールするには、以下のようになります。

1. DB2 インフォメーション・センターのインストールで定義したアカウントで、システムにログオンします。
2. CD をドライブに挿入します。自動実行機能が使用可能になっていれば、IBM DB2 セットアップ・ランチパッドが起動します。
3. DB2 セットアップ・ウィザードは、システム言語を判別して、その言語用のセットアップ・プログラムを立ち上げます。英語以外の言語でセットアップ・プログラムを実行したい場合、またはセットアップ・プログラムの自動始動が失敗した場合には、DB2 セットアップ・ウィザードを手動で開始できます。

次のようにして、DB2 セットアップ・ウィザードを手動で開始します。

- a. 「スタート」をクリックし、「ファイル名を指定して実行」を選択します。
- b. 「開く」フィールドで、以下のコマンドを入力します。

```
x:¥setup.exe /i 2-letter language identifier
```

ここで、*x:* は CD ドライブ、*2-letter language identifier* (2 文字の言語識別子) はセットアップ・プログラムを実行する言語を表します。

- c. 「OK」をクリックします。

4. IBM DB2 セットアップ・ランチパッドが開きます。DB2 インフォメーション・センターのインストールに直接進むには、「製品のインストール」をクリックします。残りのステップについて説明しているオンライン・ヘルプを利用

できます。オンライン・ヘルプを呼び出すには、「ヘルプ」をクリックします。「キャンセル」をクリックすれば、いつでもインストールを終了できます。

5. 「インストールしたい製品を選択します」ページでは、「次へ」をクリックします。
6. 「DB2 セットアップ・ウィザードによるこそ (Welcome to the DB2 Setup wizard)」ページで、「次へ」をクリックします。DB2 セットアップ・ウィザードは、プログラムのセットアップ操作を案内します。
7. インストールを続行するには、使用許諾条件に同意する必要があります。「ご使用条件」ページで、「ご使用条件に同意します (I accept the terms in the license agreement)」をクリックして、「次へ」をクリックします。
8. 「インストール・アクションの選択」で、「このコンピューターに DB2 インフォメーション・センターをインストールする (Install DB2 Information Center on this computer)」を選択します。応答ファイルを使用して、このコンピューターまたは他のコンピューターに DB2 インフォメーション・センターをあとでインストールしたい場合には、「設定を応答ファイルに保管する」を選択します。「次へ」をクリックします。
9. 「インストールする言語の選択」ページでは、DB2 インフォメーション・センターをインストールする言語を選択します。「次へ」をクリックします。
10. 「DB2 インフォメーション・センター・ポートの指定」ページでは、DB2 インフォメーション・センターへの着信通信を構成します。「次へ」をクリックしてインストールを続けます。
11. 「ファイルのコピーの開始」ページでは、インストールの選択項目を確認します。設定を変更するには、「戻る」をクリックします。「インストール」をクリックすると、DB2 インフォメーション・センターのファイルがコンピューターにコピーされます。

応答ファイルを使って DB2 インフォメーション・センターをインストールすることができます。また、**db2rspgn** コマンドを使って、既存のインストール内容に基づく応答ファイルを生成することもできます。

インストール時に検出されるエラーの詳細については、「マイ ドキュメント」¥DB2LOG¥ ディレクトリー内の db2.log ファイルと db2wi.log ファイルを参照してください。「マイ ドキュメント」ディレクトリーの場所は、ご使用のコンピューターの設定によって異なります。

db2wi.log ファイルは、DB2 の最新のインストール情報をキャプチャーします。db2.log は、DB2 製品のインストールの履歴をキャプチャーします。

#### 関連概念:

- 816 ページの『DB2 インフォメーション・センター』
- 818 ページの『DB2 インフォメーション・センターのインストール・シナリオ』

#### 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 製品のインストール (Windows)』

- 827 ページの『コンピューターまたはイントラネット・サーバーへの DB2 インフォメーション・センターの更新インストール』
- 828 ページの『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』
- 826 ページの『DB2 インフォメーション・センターの呼び出し』
- 820 ページの『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 (UNIX)』

#### 関連資料:

- 213 ページの『db2rspgn - 応答ファイル生成プログラム (Windows)』

## DB2 インフォメーション・センターの呼び出し

DB2 インフォメーション・センターは、Linux、UNIX、および Windows オペレーティング・システム用の DB2 製品 (DB2 Universal Database、 DB2 Connect、 DB2 Information Integrator、 DB2 Query Patroller など) を使用するために必要なすべての情報を提供します。

DB2 インフォメーション・センターは、以下の場所から呼び出すことができます。

- DB2 UDB クライアントまたはサーバーがインストールされているコンピューター
- DB2 インフォメーション・センターがインストールされているイントラネット・サーバーまたはローカル・コンピューター
- IBM の Web サイト

#### 前提条件:

DB2 インフォメーション・センターを呼び出すための要件は、以下のとおりです。

- オプショナル: 希望する言語でトピックを表示するようブラウザーを構成する
- オプショナル: 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・センターを使用するよう DB2 クライアントを構成する

#### 手順:

DB2 UDB クライアントまたはサーバーがインストールされているコンピューターから DB2 インフォメーション・センターを呼び出すには、以下のようになります。

- (Windows オペレーティング・システムの)「スタート」メニューから: 「スタート」→「プログラム」→「IBM DB2」→「情報」→「インフォメーション・センター」をクリックします。
- コマンド行プロンプトから:
  - Linux および UNIX オペレーティング・システムの場合、 **db2icdocs** コマンドを発行します。
  - Windows オペレーティング・システムの場合、 **db2icdocs.exe** コマンドを発行します。

イントラネット・サーバーまたはローカル・コンピュータ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 Web ブラウザーで開くには、以下のようになります。

- Web ページ <http://<host-name>:<port-number>/> を開きます (<host-name> はホスト名、<port-number> は DB2 インフォメーション・センターを利用可能なポート番号)。

IBM Web サイトにある DB2 インフォメーション・センターを Web ブラウザーで開くには、以下のようにします。

- Web ページ [publib.boulder.ibm.com/infocenter/db2help/](http://publib.boulder.ibm.com/infocenter/db2help/) を開きます。

#### 関連概念:

- 816 ページの『DB2 インフォメーション・センター』
- 818 ページの『DB2 インフォメーション・センターのインストール・シナリオ』

#### 関連タスク:

- 836 ページの『DB2 ツールからコンテキスト・ヘルプを呼び出す』
- 827 ページの『コンピューターまたはイントラネット・サーバーへの DB2 インフォメーション・センターの更新インストール』
- 838 ページの『コマンド行プロセッサからコマンド・ヘルプを呼び出す』
- 『DB2 インフォメーション・センターへのアクセスのロケーションの設定: Common GUI help』

#### 関連資料:

- 481 ページの『HELP』

---

## コンピューターまたはイントラネット・サーバーへの DB2 インフォメーション・センターの更新インストール

<http://publib.boulder.ibm.com/infocenter/db2help/> から利用できる DB2 インフォメーション・センターは、資料の新規追加または変更によって定期的に更新されます。さらに、更新された DB2 インフォメーション・センターをコンピューターまたはイントラネット・サーバーにダウンロードしてインストールできる場合もあります。DB2 インフォメーション・センターを更新しても、DB2 クライアント製品またはサーバー製品は更新されません。

#### 前提条件:

インターネットに接続されたコンピューターへのアクセスが必要です。

#### 手順:

DB2 インフォメーション・センターの更新を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するには、以下のようにします。

1. IBM の Web サイト (<http://publib.boulder.ibm.com/infocenter/db2help/>) にある DB2 インフォメーション・センターを開きます。
2. 「DB2 インフォメーション・センターによるサポート」ページの見出し「サービスおよびサポート」の「ダウンロード」セクションで、「**DB2 資料**」リンクをクリックします。
3. 最新のドキュメンテーション・イメージのレベルと、インストール済みのドキュメンテーション・レベルを比較して、DB2 インフォメーション・センターを更

新する必要があるかどうかを確認します。「DB2 インフォメーション・センターによろこそ」ページに、インストール済みのドキュメンテーションのレベルがリストされます。

4. より新しいバージョンの DB2 インフォメーション・センターが存在する場合、ご使用のオペレーティング・システムに対応する最新の DB2 インフォメーション・センター・イメージをダウンロードします。
5. 最新の DB2 インフォメーション・センター・イメージをインストールするには、Web ページの指示に従ってください。

**関連概念:**

- 818 ページの『DB2 インフォメーション・センターのインストール・シナリオ』

**関連タスク:**

- 826 ページの『DB2 インフォメーション・センターの呼び出し』
- 820 ページの『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 (UNIX)』
- 823 ページの『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 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「上へ」ボタンをクリックします。
3. 使いたい言語で DB2 インフォメーション・センターを表示するには、ページをリフレッシュします。

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. Mozilla の「編集」→「設定」→「言語」ボタンをクリックします。「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. 使いたい言語で DB2 インフォメーション・センターを表示するには、ページをリフレッシュします。

**関連概念:**

- 816 ページの『DB2 インフォメーション・センター』

## DB2 PDF 資料および印刷された資料

以下の表は、正式な資料名、資料番号、および PDF ファイル名を示しています。ハードコピー版の資料を注文するには、正式な資料名を知っておく必要があります。PDF ファイルを印刷するには、PDF ファイル名を知っておく必要があります。

DB2 資料は、以下のカテゴリーに分類されています。

- DB2 中核情報
- 管理情報
- アプリケーション開発情報
- ビジネス・インテリジェンス情報
- DB2 Connect 情報
- 入門情報
- チュートリアル情報
- オプション・コンポーネント情報
- リリース・ノート

以下の表は、DB2 ライブラリー内の各資料について、その資料のハードコピー版を注文したり、PDF 版を印刷または表示したりするのに必要な情報を示しています。

DB2 ライブラリー内の各資料に関する詳細な説明については、

[www.ibm.com/shop/publications/order](http://www.ibm.com/shop/publications/order) にある IBM Publications Center にアクセスしてください。

## DB2 の基本情報

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とって役立つ情報であるとともに、DB2 Connect、DB2 Warehouse Manager、または他の DB2 製品を使用するユーザーにとっても役立つ内容です。

表 24. DB2 の基本情報

| 資料名                                         | 資料番号      | PDF ファイル名 |
|---------------------------------------------|-----------|-----------|
| 「IBM DB2 Universal Database<br>コマンド・リファレンス」 | SC88-9140 | db2n0j81  |



表 24. DB2 の基本情報 (続き)

| 資料名                                             | 資料番号                 | PDF ファイル名 |
|-------------------------------------------------|----------------------|-----------|
| 「IBM DB2 Universal Database 用語集」                | 資料番号なし               | db2t0j81  |
| 「IBM DB2 Universal Database メッセージ・リファレンス 第 1 巻」 | GC88-9152 (ハードコピーなし) | db2m1j81  |
| 「IBM DB2 Universal Database メッセージ・リファレンス 第 2 巻」 | GC88-9153 (ハードコピーなし) | db2m2j81  |
| 「IBM DB2 Universal Database 新機能」                | SC88-9158            | db2q0j81  |

## 管理情報

これらの資料の情報は、DB2 データベース、データウェアハウス、およびフェデレーテッド・システムを効果的に設計し、インプリメントし、保守するために必要なトピックを扱っています。

表 25. 管理情報

| 資料名                                                     | 資料番号      | PDF ファイル名 |
|---------------------------------------------------------|-----------|-----------|
| 「IBM DB2 Universal Database 管理ガイド: プランニング」              | SC88-9135 | db2d1j81  |
| 「IBM DB2 Universal Database 管理ガイド: インプリメンテーション」         | SC88-9133 | db2d2j81  |
| 「IBM DB2 Universal Database 管理ガイド: パフォーマンス」             | SC88-9134 | db2d3j81  |
| 「IBM DB2 Universal Database 管理 API リファレンス」              | SC88-9136 | db2b0j81  |
| 「IBM DB2 Universal Database データ移動ユーティリティーガイドおよびリファレンス」  | SC88-9142 | db2dmj81  |
| 「IBM DB2 Universal 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」 | SC88-9143 | db2haj81  |
| 「IBM DB2 Universal Database データウェアハウス・センター管理ガイド」        | SC88-9165 | db2ddj81  |
| 「IBM DB2 Universal Database SQL リファレンス 第 1 巻」           | SC88-9155 | db2s1j81  |
| 「IBM DB2 Universal Database SQL リファレンス 第 2 巻」           | SC88-9156 | db2s2j81  |
| 「IBM DB2 Universal Database システム・モニター ガイドおよびリファレンス」     | SC88-9157 | db2f0j81  |



## アプリケーション開発情報

これらの資料の情報は、DB2 Universal Database (DB2 UDB) のアプリケーション開発者またはプログラマーが特に興味を持つ内容です。サポートされるさまざまなプログラミング・インターフェース (組み込み SQL、ODBC、JDBC、SQLJ、CLI など) を使用して DB2 UDB にアクセスするのに必要な資料とともに、サポートされる言語およびコンパイラーについても紹介されています。また、DB2 インフォメーション・センターをご使用の場合には、サンプル・プログラムのソース・コードの HTML バージョンにアクセスすることもできます。

表 26. アプリケーション開発情報

| 資料名                                                                   | 資料番号      | PDF ファイル名 |
|-----------------------------------------------------------------------|-----------|-----------|
| 「IBM DB2 Universal Database アプリケーション開発ガイド<br>アプリケーションの構築および実行」        | SC88-9137 | db2axj81  |
| 「IBM DB2 Universal Database アプリケーション開発ガイド<br>クライアント・アプリケーションのプログラミング」 | SC88-9138 | db2a1j81  |
| 「IBM DB2 Universal Database アプリケーション開発ガイド<br>サーバー・アプリケーションのプログラミング」   | SC88-9139 | db2a2j81  |
| 「IBM DB2 Universal Database コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻」      | SC88-9159 | db2l1j81  |
| 「IBM DB2 Universal Database コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻」      | SC88-9160 | db2l2j81  |
| 「IBM DB2 Universal Database データウェアハウス・センター<br>アプリケーション統合ガイド」          | SC88-9166 | db2adj81  |
| 「IBM DB2 Universal Database XML Extender 管理およびプログラミングのガイド」            | SC88-9172 | db2sxj81  |

## ビジネス・インテリジェンス情報

これらの資料の情報は、さまざまなコンポーネントを使用して、DB2 Universal Database のデータウェアハウジング機能および分析機能を拡張する方法を説明しています。

表 27. ビジネス・インテリジェンス情報

| 資料名                                                                                                   | 資料番号      | PDF ファイル名   |
|-------------------------------------------------------------------------------------------------------|-----------|-------------|
| 「IBM DB2 Warehouse Manager Standard Edition インフォメーション・カタログ・センター 管理ガイド」                                | SC88-9167 | db2dij81    |
| 「IBM DB2 Warehouse Manager Standard Edition インストール・ガイド」                                               | GC88-9164 | db2idj81    |
| 「IBM DB2 Warehouse Manager Standard Edition DB2 Warehouse Manager を使用時の ETI ソリューション・コンパージョン・プログラムの管理」 | SC88-9894 | iwhe1mstx80 |

## DB2 Connect 情報

このカテゴリーの情報は、DB2 Connect Enterprise Edition または DB2 Connect Personal Edition を使用して、メインフレーム・サーバーおよびミッドレンジ・サーバー上のデータにアクセスする方法を説明しています。

表 28. DB2 Connect 情報

| 資料名                                              | 資料番号      | PDF ファイル名 |
|--------------------------------------------------|-----------|-----------|
| 「IBM コネクティビティー 補足」                               | 資料番号なし    | db2h1j81  |
| 「IBM DB2 Connect Enterprise Edition 概説およびインストール」 | GC88-9145 | db2c6j81  |
| 「IBM DB2 Connect Personal Edition 概説およびインストール」   | GC88-9146 | db2c1j81  |
| 「IBM DB2 Connect ユーザーズ・ガイド」                      | SC88-9147 | db2c0j81  |

## 入門情報

このカテゴリーの情報は、サーバー、クライアント、および他の DB2 製品をインストールして構成する場合に役立ちます。

表 29. 入門情報

| 資料名                                                   | 資料番号                 | PDF ファイル名 |
|-------------------------------------------------------|----------------------|-----------|
| 「IBM DB2 Universal Database DB2 クライアント機能 概説およびインストール」 | GC88-9144 (ハードコピーなし) | db2itj81  |
| 「IBM DB2 Universal Database DB2 サーバー機能 概説およびインストール」   | GC88-9148            | db2isj81  |

表 29. 入門情報 (続き)

| 資料名                                                                    | 資料番号                 | PDF ファイル名 |
|------------------------------------------------------------------------|----------------------|-----------|
| 「IBM DB2 Universal Database<br>DB2 Personal Edition 概説およびインストール」       | GC88-9150            | db2i1j81  |
| 「IBM DB2 Universal Database<br>インストールおよび構成 補<br>足」                     | GC88-9149 (ハードコピーなし) | db2iyj81  |
| 「IBM DB2 Universal Database<br>DB2 Data Links Manager 概説<br>およびインストール」 | GC88-9141            | db2z6j81  |

## チュートリアル情報

チュートリアル情報は、DB2 機能を紹介し、さまざまなタスクを実行する方法を示します。

表 30. チュートリアル情報

| 資料名                                           | 資料番号   | PDF ファイル名 |
|-----------------------------------------------|--------|-----------|
| 「ビジネス・インテリジェンス・チュートリアル: データウェアハウス・センターの紹介」    | 資料番号なし | db2tuj81  |
| 「ビジネス・インテリジェンス・チュートリアル: データウェアハウジングの上級者向けガイド」 | 資料番号なし | db2taj81  |
| 「インフォメーション・カタログ・センター チュートリアル」                 | 資料番号なし | db2aij81  |
| 「Video Central for e-business チュートリアル」        | 資料番号なし | db2twj81  |
| 「Visual Explain チュートリアル」                      | 資料番号なし | db2tvj81  |

## オプション・コンポーネント情報

このカテゴリーの情報は、DB2 のオプション・コンポーネントを使用する方法について説明しています。

表 31. オプション・コンポーネント情報

| 資料名                                         | 資料番号      | PDF ファイル名 |
|---------------------------------------------|-----------|-----------|
| 「IBM DB2 Cube Views Guide and Reference」    | SC18-7298 | db2aax81  |
| 「IBM DB2 Query Patroller インストール、管理、使用法のガイド」 | GC88-9154 | db2dwj81  |

表 31. オptional・コンポーネント情報 (続き)

| 資料名                                                                 | 資料番号      | PDF ファイル名 |
|---------------------------------------------------------------------|-----------|-----------|
| 「IBM DB2 Spatial Extender and Geodetic Extender ユーザーズ・ガイドおよびリファレンス」 | SC88-9171 | db2sbj81  |
| 「IBM DB2 Universal Database Data Links Manager 管理ガイドおよびリファレンス」      | SC88-9169 | db2z0x82  |
| 「DB2 Net Search Extender 管理およびユーザーズ・ガイド」                            | SH88-8546 | N/A       |

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。

## リリース・ノート

リリース・ノートは、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。また、リリース・ノートには、各リリース、アップデート、およびフィックスパックで組み込まれた資料上の更新の要約も含まれています。

表 32. リリース・ノート

| 資料名            | 資料番号               | PDF ファイル名 |
|----------------|--------------------|-----------|
| 「DB2 リリース・ノート」 | 「注」を参照。            | 「注」を参照。   |
| 「DB2 インストール情報」 | 製品 CD-ROM でのみ参照可能。 | 使用できません。  |

注: リリース・ノートは以下の形式で入手できます。

- XHTML およびテキスト形式 (製品 CD 内)
- PDF 形式 (PDF ドキュメンテーション CD 内)

さらに、リリース・ノートの中で、『既知の問題と予備手段』および『リリース間の非互換性』に関する部分は DB2 インフォメーション・センターにも表示されます。

UNIX ベースのプラットフォームでテキスト形式でリリース・ノートを確認するには、Release.Notes ファイルを参照してください。このファイルは、DB2DIR/Readme/%L ディレクトリーに収録されています。%L はロケール名を表しています。DB2DIR は以下になります。

- AIX オペレーティング・システムの場合: /usr/opt/db2\_08\_01
- その他のすべての UNIX ベースのオペレーティング・システムの場合: /opt/IBM/db2/V8.1

関連概念:

- 815 ページの『DB2 資料とヘルプ』

**関連タスク:**

- 835 ページの『PDF ファイルからの DB2 資料の印刷方法』
- 836 ページの『DB2 の印刷資料の注文方法』
- 836 ページの『DB2 ツールからコンテキスト・ヘルプを呼び出す』

---

## PDF ファイルからの DB2 資料の印刷方法

DB2 PDF ドキュメンテーション CD に収録されている DB2 資料を印刷することができます。 Adobe Acrobat Reader を使用すれば、資料全体または特定のページを印刷できます。

**前提条件:**

Adobe Acrobat Reader がインストールされていることを確認してください。 Adobe Acrobat Reader をインストールする必要がある場合、 Adobe Web サイト ([www.adobe.com](http://www.adobe.com)) から入手できます。

**手順:**

PDF ファイルから DB2 資料を印刷するには以下のようにします。

1. DB2 PDF ドキュメンテーション CD をドライブに挿入します。 UNIX オペレーティング・システムの場合、 DB2 PDF ドキュメンテーション CD をマウントします。 UNIX オペレーティング・システムで CD をマウントする方法については、「概説およびインストール」を参照してください。
2. index.htm を開きます。ブラウザ・ウィンドウにファイルが開きます。
3. 参照したい PDF のタイトルをクリックします。 Acrobat Reader で PDF が開きます。
4. 「ファイル」 → 「印刷」を選択して、所要の資料の任意の部分印刷します。

**関連概念:**

- 816 ページの『DB2 インフォメーション・センター』

**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『CD-ROM のマウント (AIX)』
- 「DB2 Universal Database サーバー機能 概説およびインストール」の『HP-UX 上での CD-ROM のマウント』
- 「DB2 Universal Database サーバー機能 概説およびインストール」の『CD-ROM のマウント (Linux)』
- 836 ページの『DB2 の印刷資料の注文方法』
- 「DB2 Universal Database サーバー機能 概説およびインストール」の『CD-ROM のマウント (Solaris)』

**関連資料:**

- 829 ページの『DB2 PDF 資料および印刷された資料』

---

## DB2 の印刷資料の注文方法

ハードコピー版の資料を望む場合には、以下のいずれかの方法で注文できます。

### 印刷資料の注文方法:

一部の国または地域では、印刷された資料を注文することもできます。お客様がお住まいの国または地域でこのサービスが利用可能かどうかを確認するには、お住まいの国または地域の IBM Publications Web サイトをご覧ください。資料のご注文が可能な場合、以下のようになすことができます。

- 正規の IBM 製品販売業者または営業担当員に連絡してください。お客様がお住まいの地域の IBM 担当員の情報については、お手数ですが IBM の Web サイト ([www.ibm.com/planetwide](http://www.ibm.com/planetwide)) の IBM Worldwide Directory of Contacts で確認してください。
- IBM Publications Center (<http://www.ibm.com/shop/publications/order>) にアクセスしてください。なお、IBM Publications Center から資料を注文できない国もあります。

DB2 製品がご利用可能になった時点で、印刷された資料は DB2 PDF ドキュメンテーション CD にある PDF 形式の資料と同じものです。さらに、DB2 インフォメーション・センター CD に収録されている印刷された資料の内容もまた、これらと同じです。ただし、DB2 インフォメーション・センター CD には、PDF 資料にない追加情報も含まれます (たとえば、SQL 管理作業や HTML サンプル)。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、ハードコピーとしてご注文できない資料もあります。

**注:** DB2 インフォメーション・センターは、PDF またはハードコピーの資料よりも頻繁に更新されます。ドキュメンテーションの更新が入手可能になった時点でインストールするか、DB2 インフォメーション・センター (<http://publib.boulder.ibm.com/infocenter/db2help/>) を参照して最新の情報を入手してください。

### 関連タスク:

- 835 ページの『PDF ファイルからの DB2 資料の印刷方法』

### 関連資料:

- 829 ページの『DB2 PDF 資料および印刷された資料』

---

## DB2 ツールからコンテキスト・ヘルプを呼び出す

コンテキスト・ヘルプは、特定のウィンドウ、ノートブック、ウィザード、またはアドバイザーに関連したタスクまたはコントロールの情報を提供します。コンテキスト・ヘルプは、グラフィカル・ユーザー・インターフェースのある DB2 管理ツールおよび開発ツールから利用できます。コンテキスト・ヘルプには、以下の 2 種類があります。

- それぞれのウィンドウまたはノートブックにある「ヘルプ」ボタンからアクセス可能なヘルプ

- infopop (ポップアップ情報ウィンドウ)。これは、マウス・カーソルを特定のフィールドまたはコントロール上に置いたとき、またはウィンドウ、ノートブック、ウィザード、アドバイザー内でフィールドまたはコントロールを選択して F1 を押すと表示されます。

「ヘルプ」ボタンを押すと、概説、前提条件、およびタスク情報が表示されます。infopop は、それぞれのフィールドおよびコントロールについて説明します。

#### 手順:

コンテキスト・ヘルプを呼び出すには、以下のようにします。

- ウィンドウおよびノートブックのヘルプを表示するには、いずれかの DB2 ツールを開始して、任意のウィンドウまたはノートブックを開きます。ウィンドウまたはノートブックの右下隅にある「ヘルプ」ボタンをクリックして、コンテキスト・ヘルプを呼び出します。

また、それぞれの DB2 ツール・センターの上部にある「ヘルプ」メニュー項目からコンテキスト・ヘルプにアクセスすることもできます。

ウィザードおよびアドバイザーでは、最初のページの「タスクの概要」リンクをクリックすると、コンテキスト・ヘルプを表示できます。

- ウィンドウまたはノートブック上の各コントロールの infopop ヘルプを表示するには、コントロールをクリックしてから、**F1** を押します。コントロールの詳細情報を示すポップアップ情報が、黄色いウィンドウに表示されます。

**注:** フィールドまたはコントロールにマウス・カーソルを置いておくだけで infopops が表示されるようにするには、「ツール設定」ノートブックの「**文書 (Documentation)**」ページの「**infopops の自動表示**」チェック・ボックスを選択します。

infopop に似た別のコンテキスト・ヘルプに、診断ポップアップ情報があります。これにはデータ入力規則が示されます。診断ポップアップ情報は、無効または不十分なデータが入力されたとき、紫色のウィンドウに表示されます。診断ポップアップ情報は、以下に関して表示されます。

- 必須フィールド。
- 日付フィールドのように、正確なフォーマットを必要とするデータのフィールド。

#### 関連タスク:

- 826 ページの『DB2 インフォメーション・センターの呼び出し』
- 838 ページの『コマンド行プロセッサからメッセージ・ヘルプを呼び出す』
- 838 ページの『コマンド行プロセッサからコマンド・ヘルプを呼び出す』
- 839 ページの『コマンド行プロセッサから SQL 状態ヘルプを呼び出す』
- 『DB2 インフォメーション・センターへのアクセス: Concepts help』
- 『DB2 UDB ヘルプの使用法: Common GUI help』
- 『DB2 インフォメーション・センターへのアクセスのロケーションの設定: Common GUI help』
- 『DB2 コンテキスト・ヘルプと資料へのアクセスを設定する: Common GUI help』

---

## コマンド行プロセッサからメッセージ・ヘルプを呼び出す

メッセージ・ヘルプは、メッセージが出された原因と、エラーへの応答として実行すべきアクションを説明します。

### 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以下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。

たとえば、`? SQL30081` と入力すると、メッセージ `SQL30081` に関するヘルプを表示します。

### 関連概念:

- 「メッセージ・リファレンス 第 1 巻」の『メッセージの概要』

### 関連資料:

- 263 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コマンド・ヘルプを呼び出す

コマンド・ヘルプは、コマンド行プロセッサで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以下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。

たとえば、`? catalog` と入力すると、すべての `CATALOG` コマンドに関するヘルプが表示され、`? catalog database` と入力すると、`CATALOG DATABASE` コマンドのヘルプだけが表示されます。

### 関連タスク:

- 836 ページの『DB2 ツールからコンテキスト・ヘルプを呼び出す』
- 826 ページの『DB2 インフォメーション・センターの呼び出し』
- 838 ページの『コマンド行プロセッサからメッセージ・ヘルプを呼び出す』
- 839 ページの『コマンド行プロセッサから SQL 状態ヘルプを呼び出す』

### 関連資料:

- 263 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 SQL 状態ヘルプを呼び出す

DB2 Universal Database は、SQL ステートメントの結果の原因と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` または `? class code`

ここで、*sqlstate* は有効な 5 桁の SQL 状態を、*class code* は SQL 状態の最初の 2 桁を表します。

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:

- 826 ページの『DB2 インフォメーション・センターの呼び出し』
- 838 ページの『コマンド行プロセッサからメッセージ・ヘルプを呼び出す』
- 838 ページの『コマンド行プロセッサからコマンド・ヘルプを呼び出す』

---

## DB2 チュートリアル

DB2® チュートリアルは、DB2 Universal Database のさまざまな機能について学習するのを支援します。このチュートリアルでは、アプリケーションの開発、SQL 照会のパフォーマンス調整、データウェアハウスの処理、メタデータの管理、および DB2 を使用した Web サービスの開発の各分野で、段階的なレッスンが用意されています。

### 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。

チュートリアルの中で、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、それぞれのチュートリアルを参照してください。

### DB2 Universal Database チュートリアル:

以下に示すチュートリアルのタイトルをクリックすると、そのチュートリアルを表示できます。

ビジネス・インテリジェンス・チュートリアル: データウェアハウス・センターの紹介  
データウェアハウス・センターを使用して簡単なデータウェアハウジング・タスクを実行します。

ビジネス・インテリジェンス・チュートリアル: データウェアハウジングの上級者向けガイド

データウェアハウス・センターを使用して高度なデータウェアハウジング・タスクを実行します。

インフォメーション・カタログ・センター・チュートリアル

インフォメーション・カタログを作成および管理して、インフォメーション・カタログ・センターを使用してメタデータを配置し使用します。

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。

---

## 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 インフォメーション・センター、および DB2 ライブラリーに含まれる PDF 資料の中でご利用いただけます。DB2 インフォメーション・センターで、(ブラウザー・ウィンドウの左側の) ナビゲーション・ツリーの「サポートおよびトラブルシューティング (Support and troubleshooting)」ブランチを参照すると、DB2 トラブルシューティング・ドキュメンテーションの詳細なリストが見つかります。

### 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 出版物、TechNotes、プログラム診断依頼書 (APAR)、フィックスパック、DB2 内部エラー・コードの最新リスト、その他のリソース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/winosa2unix/support>) にアクセスしてください。

### DB2 Problem Determination Tutorial Series

DB2 製品で作業中に直面するかもしれない問題を素早く識別し、解決する方法に関する情報を見つけるには、DB2 Problem Determination Tutorial Series の Web サイトを参照してください。あるチュートリアルでは、使用可能な DB2 問題判別機能およびツールを紹介し、それらをいつ使用すべきかを判断する助けを与えます。別のチュートリアルは、『データベース・エンジン問題判別 (Database Engine Problem Determination)』、『パフォーマンス問題判別 (Performance Problem Determination)』、『アプリケーション問題判別 (Application Problem Determination)』などの関連トピックを扱っています。

DB2 Technical Support

(<http://www.ibm.com/software/data/support/pdm/db2tutorials.html>) には、DB2 問題判別チュートリアルがすべて揃っています。

#### 関連概念:

- 816 ページの『DB2 インフォメーション・センター』
- 「問題判別の手引き」の『Introduction to Problem Determination - DB2 テクニカル・サポートのチュートリアル』

---

## アクセス支援

アクセス支援機能は、身体に障害のある（身体動作が制限されている、視力が弱いなど）ユーザーがソフトウェア製品を十分活用できるように支援します。DB2®バージョン 8 製品に備わっている主なアクセス支援機能は、以下のとおりです。

- すべての DB2 機能は、マウスの代わりにキーボードを使ってナビゲーションできます。詳細については、『キーボードによる入力およびナビゲーション』を参照してください。
- DB2 ユーザー・インターフェースのフォント・サイズおよび色をカスタマイズすることができます。詳細については、842 ページの『アクセスしやすい表示』を参照してください。
- DB2 製品は、Java™ Accessibility API を使用するアクセス支援アプリケーションをサポートします。詳細については、842 ページの『支援テクノロジーとの互換性』を参照してください。
- DB2 資料は、アクセスしやすい形式で提供されています。詳細については、842 ページの『アクセスしやすい資料』を参照してください。

## キーボードによる入力およびナビゲーション

### キーボード入力

キーボードだけを使用して DB2 ツールを操作できます。マウスを使って実行できる操作は、キーまたはキーの組み合わせによっても実行できます。標準のオペレーティング・システム・キー・ストロークを使用して、標準のオペレーティング・システム操作を実行できます。

キーまたはキーの組み合わせによって操作を実行する方法について、詳しくは キーボード・ショートカットおよびアクセラレーター: Common GUI help を参照してください。

### キーボード・ナビゲーション

キーまたはキーの組み合わせを使用して、DB2 ツールのユーザー・インターフェースをナビゲートできます。

キーまたはキーの組み合わせによって DB2 ツールをナビゲートする方法の詳細については、キーボード・ショートカットおよびアクセラレーター: Common GUI help を参照してください。

### キーボード・フォーカス

UNIX® オペレーティング・システムでは、アクティブ・ウィンドウの中で、キー・ストロークによって操作できる領域が強調表示されます。

## アクセスしやすい表示

DB2 ツールには、視力の弱いユーザー、その他の視力障害をもつユーザーのためにアクセシビリティを向上させる機能が備わっています。これらのアクセシビリティ拡張機能には、フォント・プロパティのカスタマイズを可能にする機能も含まれています。

### フォントの設定

「ツール設定」ノートブックを使用して、メニューおよびダイアログ・ウィンドウに使用されるテキストの色、サイズ、およびフォントを選択できます。

フォント設定に関する詳細情報は、メニューおよびテキストのフォントを変更する: Common GUI help を参照してください。

### 色に依存しない

本製品のすべての機能を使用するために、ユーザーは必ずしも色を識別する必要はありません。

## 支援テクノロジーとの互換性

DB2 ツールのインターフェースは、Java Accessibility API をサポートします。これによって、スクリーン・リーダーその他の支援テクノロジーを DB2 製品で利用できるようになります。

## アクセスしやすい資料

DB2 形式は、ほとんどの Web ブラウザーで表示可能な XHTML 1.0 形式で提供されています。XHTML により、ご使用のブラウザーに設定されている表示設定に従って資料を表示できます。さらに、スクリーン・リーダーや他の支援テクノロジーを使用することもできます。

シンタックス・ダイアグラムはドット 10 進形式で提供されます。この形式は、スクリーン・リーダーを使用してオンライン・ドキュメンテーションにアクセスする場合にのみ使用できます。

#### 関連概念:

- 842 ページの『ドット 10 進シンタックス・ダイアグラム』

#### 関連タスク:

- 『キーボード・ショートカットおよびアクセラレーター: Common GUI help』
- 『メニューおよびテキストのフォントを変更する: Common GUI help』

---

## ドット 10 進シンタックス・ダイアグラム

スクリーン・リーダーを使用してインフォメーション・センターを利用するユーザーのために、シンタックス・ダイアグラムがドット 10 進形式で提供されます。

ドット 10 進形式では、各シンタックス・エレメントは別々の行に書き込まれます。複数のシンタックス・エレメントが常に同時に存在する (または常に同時に不在の) 場合、単一のコンパウンド・シンタックス・エレメントとみなせるので同一行に表示できます。

各行は、ドット 10 進数で開始します。たとえば、3 または 3.1 ないしは 3.1.1 です。こうした数を適切に聞き取るには、スクリーン・リーダーが句読点を読み取るように設定されていることを確認してください。同じドット 10 進数を持つすべてのシンタックス・エレメント (たとえば、3.1 という数値を持つすべてのシンタックス・エレメント) は、相互に排他的な代替エレメントです。3.1 USERID および 3.1 SYSTEMID という行を聞き取る場合、シンタックスには両方ではなく USERID または SYSTEMID のどちらかが含まれることが分かります。

ドット 10 進レベルは、ネストのレベルを表示します。たとえば、ドット 10 進数 3 のシンタックス・エレメントの後に、一連のドット 10 進数 3.1 のシンタックス・エレメントが続きます。3.1 の番号が付されたシンタックス・エレメントすべては、番号 3 の付されたシンタックス・エレメントに従属します。

シンタックス・エレメントに関する情報を追加するため、ドット 10 進数の次に特定のワードおよびシンボルが使用されます。時折、こうしたワードおよびシンボルはエレメントの最初に表示される場合もあります。簡単に識別するため、ワードやシンボルがシンタックス・エレメントの一部である場合には、円記号 (¥) 文字が先頭に付きます。\* シンボルはドット 10 進数の次に使用でき、シンタックス・エレメントが反復することを示します。たとえば、ドット 10 進数 3 のシンタックス・エレメント \*FILE は、3 ¥\* FILE という形式になります。3\* FILE という形式は、シンタックス・エレメント FILE が反復されることを示します。3\* ¥\* FILE という形式は、シンタックス・エレメント \* FILE が反復されることを示します。

シンタックス・エレメントのストリングを分離するのに使用されるコンマなどの文字は、シンタックス内の分離する項目の直前に表示されます。こうした文字は、それぞれの項目と同一行に表示するか、同じドット 10 進数を持つ関連する項目のある別の行に表示できます。またその行には、シンタックス・エレメントに関する情報を提供する別のシンボルを表示することも可能です。たとえば、複数の LASTRUN および DELETE シンタックス・エレメントを使用している場合には、5.1\*、5.1 LASTRUN、および 5.1 DELETE という行は、エレメントをコンマで区切る必要があります。区切り文字が指定されないと、各シンタックス・エレメントを区切るのにブランクが使用されると想定されます。

シンタックス・エレメントの前に % シンボルが付く場合、他の箇所で定義されている参照であることを示します。% シンボルの後のストリングは、リテラルではなくシンタックス・フラグメントの名前です。たとえば、2.1 %OP1 という行は別のシンタックス・フラグメント OP1 を参照すべきことを意味します。

以下のワードおよびシンボルが、ドット 10 進数の次に使用されます。

- ? は、オプションのシンタックス・エレメントであることを表します。? シンボルが後に続くドット 10 進数は、対応するドット 10 進数のシンタックス・エレメント、および任意の従属のシンタックス・エレメントがオプションであることを示します。ドット 10 進数の付いたシンタックス・エレメントが 1 つしかない場合、? シンボルはそのシンタックス・エレメントと同じ行に表示されます (たとえば、5? NOTIFY)。ドット 10 進数の付いたシンタックス・エレメントが複数ある場合、? シンボルだけで行に表示され、その後にオプションのシンタックス・エレメントが続きます。たとえば、「5 ?, 5 NOTIFY、および 5 UPDATE」という行を聞き取る場合、シンタックス・エレメント NOTIFY および UPDATE



がオプションである、つまりそのいずれかを選択でき、どちらも選択しないこともできることが分かります。 ? シンボルは、線路型ダイアグラムのバイパス線に相当します。

- ! は、デフォルトのシンタックス・エレメントであることを表します。 ! シンボルおよびシンタックス・エレメントが後に続くドット 10 進数は、そのシンタックス・エレメントが、同じドット 10 進数を共用するシンタックス・エレメントすべてのデフォルト・オプションであることを示します。同じドット 10 進数を共用するシンタックス・エレメントのうち 1 つだけに、 ! シンボルを指定できます。たとえば、「2? FILE、2.1! (KEEP)、および 2.1 (DELETE)」という行を聞き取る場合、 FILE キーワードのデフォルト・オプションは (KEEP) になります。この例では、FILE キーワードを含めてもオプションを指定しない場合には、デフォルト・オプション KEEP が適用されます。デフォルト・オプションは、次に高位のドット 10 進数にも適用されます。この例の場合、FILE キーワードが省略されると、デフォルトの FILE(KEEP) が使用されます。しかし、「2? FILE、2.1、2.1.1! (KEEP)、および 2.1.1 (DELETE)」という行を聞き取る場合、デフォルト・オプション KEEP は次に高位のドット 10 進数 2.1 (関連キーワードを持っていない) にのみ適用され、 2? FILE には適用されません。キーワード FILE が省略されると、どれも使用されません。
- \* は、0 回以上反復できるシンタックス・エレメントを示します。 \* シンボルが後に続くドット 10 進数は、このシンタックス・エレメントが 0 回以上使用できること、つまりオプションであり、なおかつ反復できることを表します。たとえば、5.1\* データ域という行を聞き取る場合、 1 つまたは複数のデータ域を含めるか、またはデータ域を全く含めないことが可能です。「3\*、3 HOST、および 3 STATE」という行を聞き取る場合、 HOST、STATE をどちらか一方または両方同時に含めるか、どちらも含めないことができます。

#### 注:

1. ドット 10 進数の後にアスタリスク (\*) が付き、ドット 10 進数の付いた項目が 1 つしかない場合には、同じ項目を複数回反復できます。
  2. ドット 10 進数の後にアスタリスクが付き、ドット 10 進数の付いた項目が複数ある場合、リストから複数の項目を使用できますが、各項目を複数回使用することはできません。前述の例では、HOST STATE と書くことはできませんが、HOST HOST とは書けません。
  3. \* シンボルは、線路型シンタックス・ダイアグラムのループバック線に相当します。
- + は、1 回以上含める必要のあるシンタックス・エレメントであることを示します。 + シンボルが後に続くドット 10 進数は、このシンタックス・エレメントを 1 回以上含める必要があること、つまり少なくとも 1 回は含める必要があり、反復できることを表します。たとえば、「6.1+ データ域」という行を聞き取る場合、データ域を少なくとも 1 回は含めなければなりません。「2+、2 HOST、および 2 STATE」という行を聞き取る場合には、 HOST、STATE、またはその両方を含める必要があります。 \* シンボルと同様に、+ シンボルは、ドット 10 進数の付いた項目が 1 つしかない場合に限り、その特定の項目のみを反復できます。 \* シンボルと同様、 + シンボルは線路型シンタックス・ダイアグラムのループバック線に相当します。

#### 関連概念:

- 841 ページの『アクセス支援』

| 関連タスク:

- | • 『キーボード・ショートカットおよびアクセラレーター: Common GUI help』

| 関連資料:

- | • 「*SQL* リファレンス 第 2 巻」の『構文図の見方』

---

| **DB2 Universal Database 製品の共通基準認証**

| DB2 Universal Database は、Common Criteria の評価検定レベル 4 (EAL4) で認証  
| の評価を受けています。Common Criteria の詳細については、以下の Common  
| Criteria の Web サイトを参照してください。 <http://niap.nist.gov/cc-scheme/>





---

## 付録 D. 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。

〒106-0032  
東京都港区六本木 3-2-31  
IBM World Trade Asia Corporation  
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を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

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

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム(本プログラムを含む)との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Canada Limited  
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が、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

以下は、IBM Corporation の商標です。

|                                                 |                  |
|-------------------------------------------------|------------------|
| ACF/VTAM                                        | iSeries          |
| AISPO                                           | LAN Distance     |
| AIX                                             | MVS              |
| AIXwindows                                      | MVS/ESA          |
| AnyNet                                          | MVS/XA           |
| APPN                                            | Net.Data         |
| AS/400                                          | NetView          |
| BookManager                                     | OS/390           |
| C Set++                                         | OS/400           |
| C/370                                           | PowerPC          |
| CICS                                            | pSeries          |
| Database 2                                      | QBIC             |
| DataHub                                         | QMF              |
| DataJoiner                                      | RACF             |
| DataPropagator                                  | RISC System/6000 |
| DataRefresher                                   | RS/6000          |
| DB2                                             | S/370            |
| DB2 Connect                                     | SP               |
| DB2 Extenders                                   | SQL/400          |
| DB2 OLAP Server                                 | SQL/DS           |
| DB2 Information Integrator                      | System/370       |
| DB2 Query Patroller                             | System/390       |
| DB2 Universal Database                          | SystemView       |
| Distributed Relational<br>Database Architecture | Tivoli           |
| DRDA                                            | VisualAge        |
| eServer                                         | VM/ESA           |
| Extended Services                               | VSE/ESA          |
| FFST                                            | VTAM             |
| First Failure Support Technology                | WebExplorer      |
| IBM                                             | WebSphere        |
| IMS                                             | WIN-OS/2         |
| IMS/ESA                                         | z/OS             |
|                                                 | zSeries          |

以下は、それぞれ各社の商標または登録商標です。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

Pentium 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。  
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 E. IBM と連絡をとる

技術上の問題がある場合は、お客様サポートにご連絡ください。

---

### 製品情報

DB2 Universal Database 製品に関する情報は、  
<http://www.ibm.com/software/data/db2/udb> から入手できます。

このサイトには、技術ライブラリー、資料の注文方法、製品のダウンロード、ニュースグループ、フィックスパック、ニュース、および Web リソースへのリンクに関する最新情報が掲載されています。

米国以外の国で IBM に連絡する方法については、IBM Worldwide ページ ([www.ibm.com/planetwide](http://www.ibm.com/planetwide)) にアクセスしてください。



# 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】

アクセス可能性  
機能 841  
小数点付き 10 進数構文図 842  
アクセス・パス  
最適化 720  
アドバイザー  
設計アドバイザー 17  
db2advis 17  
アンカatalog  
システム・データベース・ディレクトリー 762  
データベース項目 762  
ホスト DCS データベース項目 764  
暗黙接続 273  
異常終了  
再始動コマンド 695  
一時ファイル  
LOAD コマンド 558  
移動、データ  
データベース間 483  
イベント・アナライザー・コマンド 76  
イベント・モニター生産性向上ツール・コマンド 78  
イベント・モニターのターゲット表定義の生成コマンド 79  
印刷  
PDF ファイル 835  
印刷版の資料の注文 836  
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 163  
インスタンス作成インターフェースの開始コマンド 119  
インスタンスの移行コマンド 114  
インスタンスの更新コマンド 121  
インスタンスの作成コマンド 106  
インスタンスの除去コマンド 110  
インスタンスのリスト・コマンド 112  
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 160  
インストール  
インフォメーション・センター 818, 820, 823

インフォメーション・センター  
インストール 818, 820, 823  
インポート  
データ 483  
ウィザード  
設計アドバイザー 17  
db2advis 17  
エクスポート  
データベース表ファイル 393  
ファイル・タイプ修飾子 393  
DB2 Data Links Manager に関する考慮事項 393  
エラー・メッセージ  
データベース構成ファイル 422  
無効なチェックサム  
データベース構成ファイル 689, 788  
データベース・マネージャー構成ファイル 685  
リモート・データベースのドロップ 380  
オートローダー・コマンド 22  
応答ファイル  
生成プログラム  
db2rspgn 213  
大文字小文字の区別  
コマンド 273  
命名規則における 813  
オンライン  
ヘルプへのアクセス 836

## 【カ行】

カーソル固定 (CS)  
変更 355  
カーネル・パラメーター値のためのキューティリティー・コマンド 165  
開始  
DB2  
db2start コマンド 235  
カタログ作成  
データベース 330  
ホスト・データベース 334  
監査機能管理者ツール・コマンド 23  
キーボード・ショートカット  
サポート 841  
キーワード  
構文 809  
行継続文字  
コマンド行プロセッサ (CLP) 273

継続文字、コマンド行プロセッサ (CLP) 273  
検査結果のフォーマット・コマンド 118  
コード・ページ  
EXPORT コマンド 393  
IMPORT コマンド 483  
コード・ページ・ファイル・タイプ修飾子 558  
更新  
HTML 文書 827  
構成  
管理  
リセット、デフォルトに 685  
例 407  
データベース  
更新 788  
リセット、デフォルトに 689  
例 422  
データベース・マネージャー、例 428  
CLI、サンプル 416  
構文  
コマンド行プロセッサ SQL ステートメント 801  
構文図  
読み方 809  
コマンド  
バージョン 8 の現行フィックス・レベルへのデータベースの更新 252  
リダイレクト、出力の 273  
ACTIVATE DATABASE 283  
ADD CONTACT 285  
ADD CONTACTGROUP 286  
ADD DATALINKS MANAGER 287  
ADD DBPARTITIONNUM 289  
ARCHIVE LOG 291  
ATTACH 294  
AUTOCONFIGURE 296  
BACKUP DATABASE 299  
BIND 305  
CATALOG APPC NODE 324  
CATALOG APPN NODE 327  
CATALOG DATABASE 330  
CATALOG DCS DATABASE 334  
CATALOG LDAP DATABASE 337  
CATALOG LDAP NODE 341  
CATALOG LOCAL NODE 342  
CATALOG NAMED PIPE NODE 344  
CATALOG NETBIOS NODE 346  
CATALOG ODBC DATA SOURCE 348  
CATALOG TCP/IP NODE 349

## コマンド (続き)

CHANGE DATABASE  
   COMMENT 353  
 CHANGE ISOLATION LEVEL 355  
 CREATE DATABASE 357  
 CREATE TOOLS CATALOG 367  
 dasauto 3  
 dasprt 4  
 dasdrop 5  
 dasmigr 6  
 db2 263  
 DB2 JDBC パッケージ・バインド・プログラム・ユーティリティー 124  
 DB2 SQLJ プロファイル・バインド・プログラム 223  
 DB2 SQLJ 変換プログラム 260  
 db2admin 8  
 db2adutl 10  
 db2advis 17  
 db2atld 22  
 db2audit 23  
 db2batch 24  
 db2bfd 30  
 db2cap 31  
 db2cc 33  
 db2cfexp 35  
 db2cfimp 37  
 db2cidmg 38  
 db2ckbkb 39  
 db2ckmig 43  
 db2ckrst 45  
 db2cli 47  
 db2cmd 48  
 db2dclgn 53  
 db2drdat 71  
 db2empfa 75  
 db2eva 76  
 db2evmon 78  
 db2evtbl 79  
 db2flsn 89  
 db2fm 91  
 db2fs 93  
 db2gov 96  
 db2govlg 98  
 db2hc 100  
 db2iclus 102  
 db2icrt 106  
 db2idrop 110  
 db2ilist 112  
 db2imigr 114  
 db2inidb 116  
 db2inspf 118  
 db2isetup 119  
 db2iupdt 121  
 db2jdbcbind 124  
 db2ldcfg 126

## コマンド (続き)

db2level 127  
 db2licm 128  
 db2logsforrwd 130  
 db2look 131  
 db2move 141  
 db2mqlsn 147  
 db2mscs 151  
 db2mtrk 155  
 db2nchg 158  
 db2ncrt 160  
 db2ndrop 163  
 db2osconf 165  
 db2perfc 197  
 db2perfi 199  
 db2perfr 200  
 db2rbind 201  
 db2relocatedb 206  
 db2sampl 214  
 db2set 216  
 db2setup 219  
 db2sql92 220  
 db2sqljbind 223  
 db2sqljcustomize 227  
 db2sqljprint 234  
 db2start 235  
 db2stop 236  
 db2support 237  
 db2sync 240  
 db2tbtst 245  
 db2trc 246  
 db2uidl 250  
 db2undgp 249  
 db2untag 251  
 db2updv8 252  
 db2\_recon\_aid 203  
 DEACTIVATE DATABASE 370  
 DEREGISTER 372  
 DESCRIBE 373  
 DETACH 377  
 DROP CONTACT 378  
 DROP CONTACTGROUP 379  
 DROP DATABASE 380  
 DROP DATALINKS MANAGER 382  
 DROP DBPARTITIONNUM  
   VERIFY 386  
 DROP TOOLS CATALOG 388  
 ECHO 390  
 EDIT 391  
 EXPORT 393  
 FORCE APPLICATION 405  
 GET ADMIN CONFIGURATION 407  
 GET ALERT CONFIGURATION 409  
 GET AUTHORIZATIONS 415  
 GET CLI CONFIGURATION 416  
 GET CONNECTION STATE 418

## コマンド (続き)

GET CONTACTGROUP 419  
 GET CONTACTGROUPS 420  
 GET CONTACTS 421  
 GET DATABASE  
   CONFIGURATION 422  
 GET DATABASE MANAGER  
   CONFIGURATION 428  
 GET DATABASE MANAGER  
   MONITOR SWITCHES 433  
 GET DESCRIPTION FOR HEALTH  
   INDICATOR 436  
 GET HEALTH NOTIFICATION  
   CONTACT LIST 438  
 GET HEALTH SNAPSHOT 439  
 GET INSTANCE 442  
 GET MONITOR SWITCHES 443  
 GET RECOMMENDATIONS 446  
 GET ROUTINE 450  
 GET SNAPSHOT 452  
 HELP 481  
 HISTORY 482  
 IMPORT 483  
 INITIALIZE TAPE 509  
 INSPECT 510  
 LIST ACTIVE DATABASES 515  
 LIST APPLICATIONS 517  
 LIST COMMAND OPTIONS 519  
 LIST DATABASE DIRECTORY 520  
 LIST DATABASE PARTITION  
   GROUPS 524  
 LIST DATALINKS MANAGERS 526  
 LIST DBPARTITIONNUMS 527  
 LIST DCS APPLICATIONS 528  
 LIST DCS DIRECTORY 531  
 LIST DRDA INDOUBT  
   TRANSACTIONS 533  
 LIST HISTORY 535  
 LIST INDOUBT  
   TRANSACTIONS 538  
 LIST NODE DIRECTORY 542  
 LIST ODBC DATA SOURCES 545  
 LIST PACKAGES/TABLES 546  
 LIST TABLESPACE  
   CONTAINERS 549  
 LIST TABLESPACES 551  
 LOAD 558  
 LOAD QUERY 596  
 Microsoft Cluster Server 102  
 MIGRATE DATABASE 599  
 MQ Listener 147  
 PING 601  
 PRECOMPILE 603  
 PRUNE HISTORY/LOGFILE 630  
 PUT ROUTINE 632  
 QUERY CLIENT 634



## コマンド (続き)

QUIESCE 635  
QUIESCE TABLESPACES FOR  
TABLE 638  
QUIT 641  
REBIND 642  
RECONCILE 646  
REDISTRIBUTE DATABASE  
PARTITION GROUP 656  
REFRESH LDAP 660  
REGISTER 661  
REORG INDEXES/TABLE 666  
REORGCHK 674  
RESET ADMIN  
CONFIGURATION 685  
RESET ALERT  
CONFIGURATION 687  
RESET DATABASE  
CONFIGURATION 689  
RESET DATABASE MANAGER  
CONFIGURATION 691  
RESET MONITOR 693  
RESTART DATABASE 695  
RESTORE DATABASE 697  
REWIND TAPE 708  
ROLLFORWARD DATABASE 709  
RUNCMD 719  
RUNSTATS 720  
SET CLIENT 732  
SET RUNTIME DEGREE 735  
SET TABLESPACE  
CONTAINERS 737  
SET TAPE POSITION 739  
SET WRITE 742  
sqlj 260  
START DATABASE MANAGER 744  
STOP DATABASE MANAGER 752  
TERMINATE 761  
UNCATALOG DATABASE 762  
UNCATALOG DCS DATABASE 764  
UNCATALOG LDAP  
DATABASE 766  
UNCATALOG LDAP NODE 768  
UNCATALOG NODE 769  
UNCATALOG ODBC DATA  
SOURCE 770  
UNQUIESCE 771  
UPDATE ADMIN  
CONFIGURATION 773  
UPDATE ALERT  
CONFIGURATION 775  
UPDATE CLI CONFIGURATION 782  
UPDATE COMMAND OPTIONS 784  
UPDATE CONTACT 786  
UPDATE CONTACTGROUP 787

## コマンド (続き)

UPDATE DATABASE  
CONFIGURATION 788  
UPDATE DATABASE MANAGER  
CONFIGURATION 791  
UPDATE HEALTH NOTIFICATION  
CONTACT LIST 793  
UPDATE HISTORY FILE 794  
UPDATE LDAP NODE 796  
UPDATE MONITOR SWITCHES 798  
コマンド行プロセッサ (CLP)  
アクセス、データベースへの 263  
アクセス、ヘルプへの 263  
オプション 264  
行継続文字 273  
コマンド・モード 263  
シェル・コマンド 263  
終了 263, 641, 761  
使用 273  
説明 263  
対話式入力モード 263  
バッチ・モード 263  
戻りコード 272  
呼び出し 263  
SQL ステートメント 801  
コマンド行プロセッサ呼び出しコマンド  
263  
コマンド構文  
解釈 809  
コマンド構文、CLP コマンド 263  
コマンド・ヘルプ  
呼び出し 838  
コミットなし (NC) 355  
コンテナー・タグの解放コマンド 251  
コントロール・センター  
開始 33  
コントロール・センターの開始コマンド  
33

## [サ行]

最初のステップ 93  
最適化  
REORG INDEXES/TABLE コマンド  
666  
索引  
統計 720  
REORGCHK コマンド 674  
サンプル・データベースの作成コマンド  
214  
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atalog 762  
実行特権の取り消しコマンド 249

## 修飾子

ファイル・タイプ  
EXPORT コマンド 393  
IMPORT コマンド 483  
LOAD コマンド 558  
終了  
異常 695  
コマンド行プロセッサのバック・エ  
ンド処理 761  
正常 752  
終了コード、CLP 272  
小数点付き 10 進数構文図 842  
使用不可 841  
スキーマ  
新規データベースの 357  
ストレージ  
物理的 666  
すべてのパッケージの再バインド・コマン  
ド 201  
静止、ファントム 638  
設計アドバイザー 17, 446  
接続構成インポート・ツール・コマンド  
37  
接続構成エクスポート・ツール・コマンド  
35  
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 761  
宣言生成プログラム・コマンド 53  
ゾーン 10 進数ファイル・タイプ修飾子  
558  
増分リストア・イメージ順序の検査コマン  
ド 45

## [タ行]

ダンプ、トレースをファイルへ 246  
チュートリアル 839  
トラブルシューティングと問題判別  
840  
注文、DB2 資料の 836  
データ  
フラグメント化、消去、表の再編成に  
よる 666  
データベース  
移行 599  
インポート、ファイルを表へ 483  
エクスポート、表のファイルへの 393  
カタログ作成 330  
間接 (間接) ディレクトリー項目 520  
検査、許可の 415  
項目の除去 (アンカatalog) 762  
再始動 695  
再編成 674

## データベース (続き)

削除、ログ・ファイルによるリカバリーの確認 380  
情報 452  
除去、ホスト DCS 項目の 764  
統計 720  
ドロップ 380  
バックアップ履歴ファイル 630  
変更、ディレクトリー内の注釈 353  
ホーム (home) ディレクトリー項目 520  
モニター  
リセット 693  
リカバリー 709  
リストア (再構築) 697  
リモート (remote) ディレクトリー項目 520  
ロード、ファイルを表へ 558  
ロールフォワード・リカバリー 709  
データベース移動ツール・コマンド 141  
データベース構成  
更新 788  
ネットワーク・パラメーター値 788  
リセット、デフォルトに 689  
例 422  
データベース事前移行ツール 43  
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 764  
データベースの再配置コマンド 206  
データベース・システム・モニター  
GET DATABASE MANAGER  
MONITOR SWITCHES コマンド 433  
GET MONITOR SWITCHES コマンド 443  
GET SNAPSHOT 452  
RESET MONITOR コマンド 693  
UPDATE MONITOR SWITCHES コマンド 798  
データベース・ディレクトリー  
サンプル内容 520  
説明 520  
変更、注釈の 353  
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 158  
データベース・パフォーマンス値のリセット・コマンド 197  
データベース・マネージャー  
アクセス、コマンド・プロンプトからの 1  
インスタンス 442  
開始 744  
システム・コマンド 1  
停止 752  
統計 452

## データベース・マネージャー (続き)

モニター・スイッチ 433, 443  
データベース・マネージャー構成  
サンプル・ファイル 428  
GET DATABASE MANAGER  
CONFIGURATION コマンド 428  
データベース・モニター  
説明 798  
データ保全性  
維持、分離レベルを使用した 355  
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 656  
テーブル・バックアップ 299  
停止  
DB2  
db2stop コマンド 236  
ディレクトリー  
アンカタログ 762  
削除、項目の 769  
システム・データベース、除去 762  
データベース  
変更、注釈の 353  
データベース接続サービス (DCS)、項目のアンカタログ 764  
ノード  
項目の除去 769  
デフォルトの構成  
データベース、リセット 689  
ADMIN、リセット 685  
統計  
再編成、索引の 674  
データベース 720  
データベース・マネージャー 452  
REORGCHK 674  
特殊文字  
コマンド 273  
特権  
間接 415  
直接 415  
データベース  
作成時に付与される 357  
報告 415  
トラブルシューティング  
オンライン情報 840  
チュートリアル 840  
トレース  
活動化 246  
トレース・コマンド 246

## [ナ行]

ノード  
SOCKS 349  
ノード・ディレクトリー、項目の除去 769

## [ハ行]

バージョン 8 の現行フィックス・レベル  
へのデータベースの更新コマンド 252  
バイナリー・ファイル 246  
バインド  
暗黙的に作成されるスキーマ 305, 603  
エラー 357  
バインド・ファイル記述ツール・コマンド 30  
パスワード  
ATTACH コマンドを使って変更 294  
CONNECT による変更 802  
バックアップの検査コマンド 39  
バックアップ・サービス API (XBSA) 299  
パッケージ  
再作成 642  
パフォーマンス  
調整  
再編成表による 666  
REORGCHK コマンド 674  
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 199  
パフォーマンス・モニター登録ツール・コマンド 200  
パラメーター  
構文 809  
反復可能読み取り (RR)  
変更 355  
非コミット読み取り (UR)  
変更 355  
表  
再編成  
必要性の判別 674  
REORG INDEXES/TABLE コマンド 666  
統計  
説明 720  
ファイルのインポート 483  
ファイルのロード 558  
ファイルへのエクスポート 393  
表スペース状態の獲得コマンド 245  
ファイル形式  
インポート、ファイルを表へ 483  
ファイルへの表のエクスポート 393  
ファイル・タイプ修飾子  
エクスポート・ユーティリティー 393  
IMPORT コマンド 483  
LOAD コマンド 558  
ファントム静止 638  
複数の表の RECONCILE コマンド 203  
複数ページ・ファイル割り振りの使用可能化コマンド 75

文書  
表示 826  
分離レベル  
CHANGE ISOLATION LEVEL コマンド 355  
ヘルス・センターの開始コマンド 100  
ヘルプ  
コマンド用  
呼び出し 838  
表示 826, 828  
メッセージ用  
呼び出し 838  
SQL ステートメント用  
呼び出し 839  
変数  
構文 809  
ベンチマーク・ツール・コマンド 24  
ホスト・システム  
カタログ作成、データベースの 334  
除去、DCS カatalog項目の 764  
DB2 Connect がサポートする接続 334

## [マ行]

未確定トランザクション・フィールド 538  
ミラーリングされたデータベースの初期化コマンド 116  
命名規則  
データベース・マネージャー・オブジェクト 813  
メッセージ  
アクセス、ヘルプへの 263  
メッセージ・ヘルプ  
呼び出し 838  
メモリー・トラッカー・コマンド 155  
戻りコード  
コマンド行プロセッサ (CLP) 272  
モニター  
データベース 433, 443  
問題判別  
オンライン情報 840  
チュートリアル 840  
問題分析および環境収集ツール・コマンド 237

## [ヤ行]

ユーザー ID  
権限 415  
呼び出し  
コマンド・ヘルプ 838  
メッセージ・ヘルプ 838  
SQL ステートメント・ヘルプ 839

読み取り固定 (RS)  
変更 355

## [ラ行]

ライセンス管理ツール・コマンド 128  
リカバリー  
データベース 697  
ロールフォワードなし 697  
ロールフォワードを用いた 709  
リストア  
旧バージョンの DB2 データベース 697  
リモート・データベース移行コマンド 38  
レジストリー変数  
DB2OPTIONS 264  
ロード  
ファイルをデータベース表へ 558  
ファイル・タイプ修飾子 558  
ロード・ユーティリティ  
一時ファイル 558  
ロールフォワード・ペンディング状態のリセット・コマンド 211  
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 130  
ログ  
ロールフォワード中のリスト 709  
ログ・シーケンス番号の検出コマンド 89  
ロック  
リセット、最大値をデフォルトに 689

## [ワ行]

ワークステーション  
リモート  
アンカログ、ローカル・ワークステーションから 769  
カタログ作成、データベースの 330  
除去、データベースのカタログ項目の 762

## A

action プリコンパイル/ BIND オプション 305, 603  
ACTIVATE DATABASE コマンド 283  
ADD CONTACT コマンド 285  
ADD CONTACTGROUP コマンド 286  
ADD DATALINKS MANAGER コマンド 287  
ADD DBPARTITIONNUM コマンド 289  
ADMIN 構成  
ネットワーク・パラメーター値 773  
ファイル 407

ADMIN 構成 (続き)  
リセット、デフォルトに 685  
例 407  
Administration Server  
構成 407  
作成 8  
ドロップ 8  
anyorder ファイル・タイプ修飾子 558  
APPC (拡張プログラム間通信機能)  
ノード  
アンカログ 769  
ARCHIVE LOG コマンド 291  
ASC インポート・ファイル・タイプ 483  
ATTACH コマンド 294  
AUTOCONFIGURE コマンド 296

## B

BACKUP DATABASE コマンド 299  
binarynumerics ファイル・タイプ修飾子 558  
BIND コマンド  
構文 305  
bindfile プリコンパイル・オプション 603  
blocking プリコンパイル/ BIND オプション 305, 603

## C

CALL ステートメント  
実行、CLP による 801, 802  
CATALOG APPC NODE コマンド 324  
CATALOG APPN NODE コマンド 327  
CATALOG DATABASE コマンド  
構文 330  
CATALOG DCS DATABASE コマンド 334  
CATALOG LDAP DATABASE コマンド 337  
CATALOG LDAP NODE コマンド 341  
CATALOG LOCAL NODE コマンド 342  
CATALOG NAMED PIPE NODE コマンド 344  
CATALOG NETBIOS NODE コマンド 346  
CATALOG ODBC DATA SOURCE コマンド 348  
CATALOG TCP/IP NODE コマンド 349  
CCSIDG プリコンパイル/ BIND オプション 305, 603  
CCSIDM プリコンパイル/ BIND オプション 305, 603  
CCSIDS プリコンパイル/ BIND オプション 305, 603

CHANGE DATABASE COMMENT コマンド 353

CHANGE ISOLATION LEVEL コマンド 355

chardel ファイル・タイプ修飾子  
インポート 483  
エクスポート 393  
ロード 558

charsub プリコンパイル/ BIND オプション 305, 603

CLI (コール・レベル・インターフェース) 構成 416

CLIPKG プリコンパイル/ BIND オプション 305

CLI/ODBC 静的パッケージ・バインディング・ツール・コマンド 31

CLOSE ステートメント  
実行、CLP による 801

CLP (コマンド行プロセッサ)  
コマンド構文 263  
終了 641, 761

cnulreqd プリコンパイル/ BIND オプション 305, 603

coldel ファイル・タイプ修飾子  
インポート 483  
エクスポート 393  
ロード 558

collection プリコンパイル/ BIND オプション 305, 603

compound ファイル・タイプ修飾子 483

CONNECT ステートメント  
実行、CLP による 801

connect プリコンパイル・オプション 603

CREATE DATABASE コマンド  
説明 357

CREATE TOOLS CATALOG コマンド 367

## D

DAS (DB2 Administration Server)  
構成 407  
作成 8  
ドロップ 8

DAS の自動始動コマンド 3

dasauto コマンド 3

dasrt コマンド 4

dasdrop コマンド 5

dasmigr コマンド 6

dateformat ファイル・タイプ修飾子 483, 558

datesiso ファイル・タイプ修飾子 393, 483, 558

DATETIME プリコンパイル/ BIND オプション 305, 603

db2  
CMD 記述 263

DB2 Administration Server (DAS)  
作成 8  
ドロップ 8

DB2 Administration Server コマンド 8

DB2 Administration Server の移行コマンド 6

DB2 Administration Server の作成コマンド 4

DB2 Administration Server の除去コマンド 5

DB2 Connect  
サポートされる接続 334

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 124

DB2 SQLJ プロファイル・カスタマイザー・コマンド 227

DB2 SQLJ プロファイル・バインド・プログラム・コマンド 223

DB2 SQLJ プロファイル・プリンター・コマンド 234

DB2 SQLJ 変換プログラム・コマンド 260

DB2 インフォメーション・センター 816  
呼び出し 826

DB2 管理プログラム・コマンド 96

DB2 管理プログラム・ログ照会コマンド 98

db2 コマンド 263

DB2 コマンド・ウィンドウのオープン・コマンド 48

DB2 サービス・レベルの表示コマンド 127

DB2 索引アドバイザー 17

DB2 障害モニター・コマンド 91

DB2 シンクロナイザーの開始コマンド 240

DB2 対話機能 CLI コマンド 47

DB2 チュートリアル 839

DB2 統計および DDL 抽出ツール・コマンド 131

DB2 のインストール・コマンド 219

DB2 の開始コマンド 235

DB2 の停止コマンド 236

DB2 ブック  
PDF ファイルの印刷 835

DB2 プロファイル・レジストリー・コマンド 216

db2admin コマンド 8

db2adutl コマンド 10

db2advis 17, 446

db2atld コマンド 22

db2audit コマンド 23

db2batch コマンド 24

db2bfd コマンド 30

db2cap コマンド 31

db2cc コマンド 33

db2cfexp コマンド 35

db2cfimp コマンド 37

db2cidmg コマンド 38

db2ckbkp コマンド 39

db2ckmig コマンド 43

db2ckrst コマンド 45

db2cli コマンド 47

db2cmd コマンド 48

db2dclgn 宣言生成プログラム  
構文 53

db2drdat コマンド 71

db2empfa コマンド 75

db2eva コマンド 76

db2evmon コマンド 78

db2evtbl コマンド 79

db2exfmt ツール 81

db2expln ツール  
構文とパラメーター 83

db2flsn コマンド 89

db2fm コマンド 91

db2fs コマンド 93

db2gov コマンド 96

db2govlg コマンド 98

db2hc コマンド 100

db2iclus コマンド 102

db2icrt コマンド 106

db2idrop コマンド 110

db2ilist コマンド 112

db2imigr コマンド 114

db2inidb コマンド 116

db2inspf コマンド 118

db2isetup コマンド 119

db2iupdt コマンド 121

db2jdbcbind コマンド 124

db2ldcfg コマンド 126

db2level コマンド 127

db2licm コマンド 128

db2logsforrwd コマンド 130

db2look コマンド 131

db2move コマンド 141

db2mqlsn コマンド 147

db2mscs コマンド 151

db2mtrk コマンド 155

db2nchg コマンド 158

db2ncrt コマンド 160

db2ndrop コマンド 163

DB2OPTIONS レジストリー変数 264

db2osconf コマンド 165

db2perfco コマンド 197

db2perfi コマンド 199

db2perfr コマンド 200

db2profc コマンド 227

db2profp コマンド 234

db2rbind コマンド 201  
db2relocatedb コマンド 206  
db2rfpen コマンド 211  
db2rspgn 応答ファイル生成プログラム  
213  
db2sampl コマンド 214  
db2set コマンド 216  
db2setup コマンド 219  
db2sql92 コマンド 220  
db2sqljbind コマンド 223  
db2sqljcustomize コマンド 227  
db2sqljprint コマンド 234  
db2start コマンド 235, 744  
db2stop コマンド 236, 752  
db2support コマンド 237  
db2sync コマンド 240  
db2tbst コマンド 245  
db2trc コマンド 246  
db2uidl コマンド 250  
db2undgp コマンド 249  
db2untag コマンド 251  
db2updv8 コマンド 252  
db2\_recon\_aid コマンド 203  
DCLGEN コマンド、「db2dclgn 宣言生成  
プログラム」を参照 53  
DEACTIVATE DATABASE コマンド  
370  
dec プリコンパイル/ BIND オプション  
305, 603  
decdec1 プリコンパイル/ BIND オプション  
305, 603  
DECLARE CURSOR ステートメント  
実行、CLP による 802  
decplusblank ファイル・タイプ修飾子  
393, 483, 558  
decpt ファイル・タイプ修飾子 393, 483,  
558  
deferred\_prepare プリコンパイル・オプシ  
ョン 603  
degree プリコンパイル/ BIND オプション  
305, 603  
delprioritychar ファイル・タイプ修飾子  
483, 558  
DEREGISTER コマンド 372  
DESCRIBE コマンド 373  
DETACH コマンド 377  
disconnect プリコンパイル・オプション  
603  
dlldel ファイル・タイプ修飾子 393, 483,  
558  
DRDA トレース・コマンド 71  
DROP CONTACT コマンド 378  
DROP CONTACTGROUP コマンド 379  
DROP DATABASE コマンド  
構文 380

DROP DATALINKS MANAGER コマンド  
382  
DROP DBPARTITIONNUM VERIFY コマ  
ンド 386  
DROP TOOLS CATALOG コマンド 388  
dumpfile ファイル・タイプ修飾子 558  
DYNAMICRULES プリコンパイル/ BIND  
オプション  
BIND コマンド 305  
PRECOMPILE コマンド 603

## E

ECHO コマンド 390  
EDIT コマンド 391  
explain BIND オプション 305, 603  
Explain 表  
データ用のフォーマット・ツール 81  
explsnap プリコンパイル/ BIND オプショ  
ン 305, 603  
EXPORT コマンド 393

## F

fastparse ファイル・タイプ修飾子 558  
federated プリコンパイル/ BIND オプショ  
ン 305, 603  
FETCH ステートメント  
CLP による実行 802  
FORCE APPLICATION コマンド 405  
forcein ファイル・タイプ修飾子 483,  
558  
funcpath プリコンパイル/ BIND オプショ  
ン 305, 603

## G

generatedignore ファイル・タイプ修飾子  
483, 558  
generatedmissing ファイル・タイプ修飾子  
483, 558  
generatedoverride ファイル・タイプ修飾子  
558  
generic プリコンパイル/ BIND オプショ  
ン 305, 603  
GET ADMIN CONFIGURATION コマン  
ド 407  
GET ALERT CONFIGURATION コマンド  
409  
GET AUTHORIZATIONS コマンド 415  
GET CLI CONFIGURATION コマンド  
416  
GET CONNECTION STATE コマンド  
418  
GET CONTACTGROUP コマンド 419

GET CONTACTGROUPS コマンド 420  
GET CONTACTS コマンド 421  
GET DATABASE CONFIGURATION コマ  
ンド 422  
GET DATABASE MANAGER  
CONFIGURATION コマンド 428  
GET DATABASE MANAGER MONITOR  
SWITCHES コマンド 433  
GET DESCRIPTION FOR HEALTH  
INDICATOR コマンド 436  
GET HEALTH NOTIFICATION  
CONTACT LIST コマンド 438  
GET HEALTH SNAPSHOT コマンド  
439  
GET INSTANCE コマンド 442  
GET MONITOR SWITCHES コマンド  
443  
GET RECOMMENDATIONS  
コマンド 446  
GET ROUTINE コマンド 450  
GET SNAPSHOT コマンド 452  
UPDATE MONITOR SWITCHES への  
影響 798

grant BIND オプション 305  
grantgroup BIND オプション 305  
grantuser BIND オプション 305

## H

HELP コマンド 481  
HISTORY コマンド 482  
HTML 文書  
更新 827

## I

identityignore 483  
identityignore ファイル・タイプ修飾子  
558  
identitymissing ファイル・タイプ修飾子  
483, 558  
identityoverride ファイル・タイプ修飾子  
558  
implieddecimal ファイル・タイプ修飾子  
483, 558  
IMPORT コマンド 483  
indexfreespace ファイル・タイプ修飾子  
558  
indexixf ファイル・タイプ修飾子 483  
indexschema ファイル・タイプ修飾子  
483  
INITIALIZE TAPE コマンド 509  
insert プリコンパイル/ BIND オプション  
305, 603  
INSPECT コマンド 510



IPX/SPX ノード  
アンカタログ 769  
isolation プリコンパイル/ BIND オプション 305, 603

## J

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 124

## K

keepblanks ファイル・タイプ修飾子 483, 558

## L

LANGLEVEL プリコンパイル・オプション 603  
SQL92E 603  
LDAP 環境の構成コマンド 126  
level プリコンパイル・オプション 603  
LIST ACTIVE DATABASES コマンド 515  
LIST APPLICATIONS コマンド 517  
LIST COMMAND OPTIONS コマンド 519  
LIST DATABASE DIRECTORY コマンド 520  
LIST DATABASE PARTITION GROUPS コマンド 524  
LIST DATALINKS MANAGERS コマンド 526  
LIST DBPARTITIONNUMS コマンド 527  
LIST DCS APPLICATIONS コマンド 528  
LIST DCS DIRECTORY コマンド 531  
LIST DRDA INDOUBT TRANSACTIONS コマンド 533  
LIST HISTORY コマンド 535  
LIST INDOUBT TRANSACTIONS コマンド 538  
LIST NODE DIRECTORY コマンド 542  
LIST ODBC DATA SOURCES コマンド 545  
LIST PACKAGES コマンド 546  
LIST PACKAGES/TABLES コマンド 546  
LIST TABLES コマンド 546  
LIST TABLESPACE CONTAINERS コマンド 549  
LIST TABLESPACES コマンド 551  
LOAD QUERY コマンド 596  
LOAD コマンド 558

lobsinfile ファイル・タイプ修飾子 393, 483, 558  
longerror プリコンパイル・オプション 603

## M

messages プリコンパイル/ BIND オプション 305, 603  
Microsoft Cluster Server コマンド 102  
MIGRATE DATABASE コマンド 599  
MQ Listener コマンド 147

## N

NetBIOS  
ノード  
アンカタログ 769  
nochecklengths ファイル・タイプ修飾子 483, 558  
nodefaults ファイル・タイプ修飾子 483  
nodoublede1 ファイル・タイプ修飾子 393, 483, 558  
noeofchar ファイル・タイプ修飾子 483, 558  
noheader ファイル・タイプ修飾子 558  
NOLINEMACRO プリコンパイル・オプション 603  
norowwarnings ファイル・タイプ修飾子 558  
notypeid ファイル・タイプ修飾子 483  
NULL 値  
コマンド行プロセッサの表現 273  
NULL スtring 273  
nullindchar ファイル・タイプ修飾子 483, 558

## O

OPEN ステートメント  
実行、CLP による 802  
optlevel プリコンパイル・オプション 603  
output プリコンパイル・オプション 603  
owner プリコンパイル/ BIND オプション 305, 603

## P

packages プリコンパイル・オプション 603  
packeddecimal ファイル・タイプ修飾子 558  
pagefreespace ファイル・タイプ修飾子 558

PING コマンド 601  
PRECOMPILE コマンド 603  
PREP コマンド 603  
preprocessor プリコンパイル・オプション 603  
PRUNE HISTORY/LOGFILE コマンド 630  
PUT ROUTINE コマンド 632

## Q

qualifier プリコンパイル/ BIND オプション 305, 603  
QUERY CLIENT コマンド 634  
queryopt プリコンパイル/ BIND オプション  
BIND コマンド 305  
PRECOMPILE コマンド 603  
QUIESCE TABLESPACES FOR TABLE コマンド 638  
QUIESCE コマンド 635  
QUIT コマンド 641

## R

REBIND コマンド 642  
reclen ファイル・タイプ修飾子 483  
ロード 558  
RECONCILE コマンド  
構文 646  
REDISTRIBUTE DATABASE PARTITION GROUP コマンド 656  
REFRESH LDAP コマンド 660  
REGISTER コマンド 661  
release プリコンパイル/ BIND オプション 305, 603  
REORG TABLE コマンド 666  
REORGCHK コマンド 674  
RESET ADMIN CONFIGURATION コマンド 685  
RESET ALERT CONFIGURATION コマンド 687  
RESET DATABASE CONFIGURATION コマンド 689  
RESET DATABASE MANAGER CONFIGURATION コマンド 691  
RESET MONITOR コマンド 693  
RESTART DATABASE コマンド 695  
RESTORE DATABASE コマンド 697  
REWIND TAPE コマンド 708  
ROLLFORWARD DATABASE コマンド 709  
RUNCMD コマンド 719  
RUNSTATS コマンド  
構文 720

## S

SELECT ステートメント  
    EXPORT コマンド内の 393  
SET CLIENT コマンド 732  
SET RUNTIME DEGREE コマンド 735  
SET TABLESPACE CONTAINERS コマ  
    ンド 737  
SET TAPE POSITION コマンド 739  
SET WRITE コマンド 742  
SIGALRM 信号  
    開始、データベース・マネージャの  
        744  
SIGINT 信号  
    開始、データベース・マネージャの  
        744  
SOCKS ノード  
    パラメーター 349  
SQL ステートメント  
    アクセス、ヘルプへの 263  
    コマンド行の使用 801  
    CLP による実行 801  
SQL ステートメント・ヘルプ  
    呼び出し 839  
SQL92 準拠 SQL ステートメント・プロ  
    セッサ・コマンド 220  
sqlca プリコンパイル・オプション 603  
sqlerror プリコンパイル/ BIND オプショ  
    ン 305, 603  
sqlflag プリコンパイル・オプション 603  
sqlj コマンド 260  
SQLJ プロファイル・バインド・プログラ  
    ム・コマンド 223  
SQLJ 変換プログラム・コマンド 260  
sqlrules プリコンパイル・オプション  
    603  
sqlwarn プリコンパイル/ BIND オプショ  
    ン 305, 603  
START DATABASE MANAGER コマン  
    ド 744  
STOP DATABASE MANAGER コマンド  
    752  
strdel プリコンパイル/ BIND オプション  
    305, 603  
striptblanks ファイル・タイプ修飾子 483,  
    558  
striptnulls ファイル・タイプ修飾子 483,  
    558  
subtableconvert ファイル・タイプ修飾子  
    558  
syncpoint プリコンパイル・オプション  
    603

## T

target プリコンパイル・オプション 603

TCP/IP  
    ノード  
        アンカタログ 769  
TERMINATE コマンド 761  
text プリコンパイル/ BIND オプション  
    305, 603  
timeformat ファイル・タイプ修飾子 483,  
    558  
timestampformat ファイル・タイプ修飾子  
    483, 558  
totalreespace ファイル・タイプ修飾子  
    558  
transform group プリコンパイル/ BIND オ  
    プション 305, 603  
True Type フォント  
    コマンド行プロセッサの要件 273  
TSM アーカイブ・イメージ 10  
TSM アーカイブ・イメージによる作業コ  
    マンド 10

## U

UNCATALOG DATABASE コマンド  
    762  
UNCATALOG DCS DATABASE コマンド  
    764  
UNCATALOG LDAP DATABASE コマン  
    ド 766  
UNCATALOG LDAP NODE コマンド  
    768  
UNCATALOG NODE コマンド 769  
UNCATALOG ODBC DATA SOURCE コ  
    マンド 770  
UNQUIESCE コマンド 771  
UPDATE ADMIN CONFIGURATION コ  
    マンド 773  
UPDATE ALERT CONFIGURATION コマ  
    ンド 775  
UPDATE CLI CONFIGURATION コマン  
    ド 782  
UPDATE COMMAND OPTIONS コマンド  
    784  
UPDATE CONTACT コマンド 786  
UPDATE CONTACTGROUP コマンド  
    787  
UPDATE DATABASE CONFIGURATION  
    コマンド 788  
UPDATE DATABASE MANAGER  
    CONFIGURATION コマンド 791  
UPDATE HEALTH NOTIFICATION  
    CONTACT LIST コマンド 793  
UPDATE HISTORY FILE コマンド 794  
UPDATE LDAP NODE コマンド 796  
UPDATE MONITOR SWITCHES コマン  
    ド 798

usedefaults ファイル・タイプ修飾子 483,  
    558

## 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 250  
validate プリコンパイル/ BIND オプショ  
    ン 305, 603  
version プリコンパイル・オプション 603

## W

WCHARTYPE プリコンパイラー・オプシ  
    ョン  
        プリコンパイル・コマンド 603  
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 151

## X

XBSA (バックアップ・サービス  
    API) 299

## [特殊文字]

! シェル・コマンド 263









Printed in Japan

SC88-9140-01



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