

DB2® Universal Database™



リリース情報

バージョン 8

DB2® Universal Database™



リリース情報

バージョン 8

目次

リリース情報について	xiii
----------------------	------

第 1 部 1

既知の制限、問題および予備手段	3
製品および製品レベルのサポート	4
バックレベル・サーバーのサポート	4
DB2 UDB バージョン 7 サーバー・アクセス	7
クラシック・コネクトは使用不可	7
データウェアハウス・センターは中国語 (簡体字) では使用しない	7
データウェアハウス・センターの下位レベル・サーバーのサポート制限	7
DB2 Administration Server	8
DB2 Workgroup Server Edition 用 DB2 ライセンス・ポリシー	8
DB2 Web ツール	8
DB2 Warehouse Manager は中国語 (簡体字) では使用しない	9
デベロップメント・センターは 64 ビット・オペレーティング・システムをサポートしない	9
Intel 32 ビット Linux オペレーティング・システム上のデベロップメント・センター	9
デベロップメント・センター・ラージ・オブジェクト (LOB) サポート (z/OS、OS/390)	9
デベロップメント・センターは Windows 98 オペレーティング・システムをサポートする	9
連合システムの制限	9
DB2 Universal Database Workgroup Server Edition に含まれる高可用性機能	12
ライセンス・センターのバックレベル・バージョンの非サポート	12
64 ビット・オペレーティング・システムで Microsoft XP のフィックスが必要	13
MVS はサポートされない	13
Net Search Extender	13
XML Extender	13
Windows XP オペレーティング・システム	13
構成アシスタント	13

サポートされないバインド・オプション	13
DB2 バックアップおよびリストア	14
Linux 390 オペレーティング・システム上のバックアップおよびリストア	14
DB2 Data Links Manager	14
Data Links サーバーのバックアップは、Tivoli Storage Manager アーカイブ・サーバー (AIX、Solaris オペレーティング環境) を使用しない	14
DB2 レプリケーション	15
DB2 データ・レプリケーション用の Java 管理 API ドキュメンテーション	15
列マッピングの制限およびレプリケーション・センター	16
iSeries システムでのレプリケーション・センターの制限	16
データウェアハウス・センター	16
リモート・オブジェクトの日本語名	16
Clean Data トランスフォーマーの制限	16
レプリケーション用のウェアハウス・エージェントの使用および Client Connect ウェアハウス・ソースへのアクセス	17
ウェアハウス・プロセスの実行間隔をスケジュールする	17
デベロップメント・センター	18
デベロップメント・センター・サーバー接続 (AIX)	18
デベロップメント・センター・オンライン・ヘルプが、IBM DB2 アプリケーション・ドライバーを IBM DB2 Java Combined Client として誤って参照する	18
ドキュメンテーション	18
DB2 レプリケーションの手引きおよび解説書ドキュメンテーション	18
DB2、バージョン 8 HTML ドキュメンテーション・インストールの制限 (Windows)	18
Microsoft Internet Explorer でのドキュメンテーション検索の予備手段	19
インストール時のオプションにない言語の DB2 インフォメーション・センターのインストール	19

GUI ツール	20
コントロール・センターのサンプル・プラグイン	20
DB2 GUI ツールでのインド語文字の表示	20
Linux オペレーティング・システムが稼働中の zSeries サーバーで GUI ツールはサポートされない	21
列のロードおよびインポート・ページで IXF ファイル内の DBCS 文字がサポートされない	21
ロード操作失敗時に誤ったインディケーターが表示される	21
GUI ツール用の最低ディスプレイ設定	21
製品のセンターでの SQL1224N エラー	21
ヘルス・モニター	22
SQL アシスタント	22
Windows オペレーティング・システムで、使用不可で表示されたときに、メイン SQL アシスタント・ウィンドウを使用可能にする	22
SQL アシスタント・ボタンがコマンド・センターで使用できない	22
DB2 から起動される 2 つのバージョンの SQL アシスタント	22
追加情報	23
CLI の非同期実行	23
ユニコード・サーバー動作の変更	23
Java 共通クライアント	23
UNIX オペレーティング・システムでの Java 関数およびルーチン	23
翻訳版の MDAC ファイルが最初にインストールされていない場合、英語の Microsoft Data Access Components (MDAC) ファイルが DB2 V8.1 のすべての各国語版に使用される	24

第 2 部 ドキュメンテーションの更新 (サマリー) 25

新着情報	27
管理および開発の GUI ツールの手引き	29
インストール	31
概説およびインストール	31
Warehouse Manager インストールの手引き	35

インストールおよび構成補足	37
-------------------------	----

データベース管理 39	
管理ガイド	39
データ移動ユーティリティー ガイドおよびリファレンス	42
データ・リカバリーと高可用性のガイドおよびリファレンス	42
DB2 Connect ユーザーズ・ガイド	43
サテライト管理 ガイドおよびリファレンス	44
DB2 レプリケーションのガイドおよびリファレンス	44
システム・モニター ガイドおよびリファレンス	46
連合システムの手引き	47

アプリケーション開発 49	
管理 API 解説書	49
CLI のガイドおよびリファレンス	50
アプリケーション開発の手引き	52

DB2 機能の拡張 57	
インフォメーション・カタログ・センター 管理ガイド	57
データウェアハウス・センター 管理ガイド	57
データウェアハウス・センター・アプリケーション統合の手引き	58
Spatial Extender User's Guide and Reference	58
XML エクステンダー 管理およびプログラミング	59
Data Links Manager 管理ガイドおよび解説書	59

Cross-area 参照 61	
メッセージ解説書	61
コマンド解説書	62
SQL 解説書	63

追加のトピックの更新	67
----------------------	----

第 3 部 69

オンライン・ヘルプの修正および更新	71
SQL ストアード・プロシージャの C 環境をデベロップメント・センターで構成	71
MQSeries での作業のデベロップメント・センターの前提条件	71

Java ルーチンをデベロップメント・センターでコンパイル可能にするパスの設定	72
Runstats ダイアログ - 更新された到達情報	72
クレンジング・タイプ記述の訂正	72
Spatial Extender - 索引アドバイザー使用時の要件	73
Java ストアード・プロシージャのビルド・オプションをデベロップメント・センターで指定	73
Web ツール - ログイン中に表示される情報	73

第 4 部 複製トピック 75

新着情報	87
製品およびパッケージの変更点	87
ブラウザからアクセスする DB2 インフォメーション・センター	88
定期的に更新される DB2 資料	89
JDBC ドライバーの機能強化	90

インストール文書 93

概説およびインストール	93
コントロール・センターを使ったマルチサイト更新の使用可能化	93
構成アシスタント (CA) によるデータベース接続の構成	94
クライアントでの TCP/IP ノードのカタログ	95
コマンド行プロセッサを使用した、クライアントからサーバーへの接続の構成	97
構成アシスタント (CA) のエクスポート機能を使用したクライアント・プロファイルの作成	98
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する	99
DB2 Personal Edition	100
DB2 Personal Edition 上のインスタンスおよびデータベースの移行 (Linux)	101
DB2 Personal Edition 上のデータベースの移行 (Windows)	103
DB2 セットアップ・ウィザードを使用した DB2 オンライン・ドキュメンテーションのインストール (UNIX)	104
移行に関する推奨事項	106
DB2 クライアント	108

doc_install コマンドを使用した DB2 HTML ドキュメンテーションのインストール (UNIX)	109
db2setup の各国語表示	110
サポートされる DB2 インターフェース言語、ロケール、およびコード・ページ	111
構成アシスタント (CA) のインポート機能を使用したクライアント・プロファイルの構成	116
データベース接続のテスト	117
START または STOP DATABASE MANAGER 処理中のエラー	117
DB2 Connect のバインディング・データベース・ユーティリティ	119
ライセンス・センターを使用した DB2 Connect ライセンス・キーのインストール	121
db2licm コマンドを使用した DB2 Connect ライセンス・キーのインストール	122
ライセンス・センターを使用したライセンス・タイプの設定	123
DB2 Data Links Manager をインストールする前に (Solaris オペレーティング環境)	124
DB2 Data Links Manager をインストールする前に (AIX)	134
DB2 Data Links Manager をインストールする前に (Windows)	143
V8.1 の DATALINK 列に対するレジストリー変数のサイズ	153
インストールおよび構成補足	153
Tivoli Ready の使用可能性	153
ディスクバリーによるデータベース接続の構成	154
CLP によるクライアントでの名前付きパイプの構成	155
NetBIOS 接続に関するクライアントの論理アダプター番号の判別	156
ホストおよび AS/400 アプリケーションから DB2 Universal Database サーバーへのアクセス	157
ホストまたは OS/400 から DB2 Universal Database サーバーにアクセスするためにサポートされているプロトコル	158
手動インストール後の DB2 サーバーのセットアップ	159
DB2 製品の手動インストール	161
応答ファイルのキーワード	163

マシンにインストールされている HTML ドキュメンテーションをアップデートする	168	z/OS ウェアハウス・エージェントを使用 した、DataPropagator レプリケーション・ アプリ・ステップの自動化	211
WebSphere Application Server 上での DB2 Web ツールのデプロイ	169	ウェアハウス・エージェントのロギングの 開始	213
WebLogic アプリケーション・サーバー上 での DB2 Web ツールのデプロイ	174	iSeries ウェアハウス・エージェントのイ ンストール	214
その他のアプリケーション・サーバー上で の DB2 Web ツールのデプロイ	176	インフォメーション・カタログ管理ウィザ ードの実行	215
db2licm コマンドによる DB2 ライセン ス・ポリシーの設定	180	AIX Websphere IBM HTTP Web サーバ ーへの、Web 用インフォメーション・カタ ログ・センターのインストール	216
UNIX での Data Direct Technologies Driver Manager および DB2 UDB パージ ョン 8 ODBC ドライバー	181	Web 用インフォメーション・カタログ・ センターのインストール後の構成	218
応答ファイル・インストール・エラー・コ ード	183	AIX、Linux、および Solaris オペレーティ ング環境のウェアハウス・エージェントの インストール	223
DB2 製品ライセンス・ファイル	184	z/OS ウェアハウス・エージェントの概要 エージェント・デーモンを z/OS 開始済み タスクとして開始する	226
db2licm コマンドによるライセンス・タイ プの設定	185	DB2 チュートリアル	227
db2licm コマンドによる DB2 ライセン ス・キーのインストール	185	AIX、Linux および Solaris オペレーティ ング環境用サンプル IWH.environment フ ァイル	228
ライセンス・センターによる DB2 ライセ ンス・キーのインストール	187	ステップ完了の z/OS コンソール通知の設 定	232
Warehouse Manager インストールの手引き	188	データベース管理 233	
Windows NT Websphere IBM HTTP サー バーへの Web 用インフォメーション・カ タログ・センターのインストール	188	管理ガイド 233	
AIX、Linux、および Solaris オペレーティ ング環境のウェアハウス・エージェント環 境の構成	190	ユーロを使用可能なコード・ページ遷移表 ファイル	233
ウェアハウス・エージェントをインストー ルするための準備	194	サポートされているテリトリリー・コードお よびコード・ページ	242
z/OS ウェアハウス・エージェントのイン ストール	197	双方向特定の CCSID	270
z/OS ウェアハウス・エージェント・デー モンの開始	202	DB2 でのユニコードのインプリメンテー ション	273
DB2 Warehouse Manager ユーザー定義プ ログラム	203	データベース・ディレクトリーとファイル	276
トリガー・プログラムによるウェアハウ ス・ステップのスケジューリング (XTClient)	204	MDC 表を作成する際の考慮事項	279
z/OS 版 DB2 でのウェアハウス・トラン スフォーマーのセットアップ	206	バージョン 8 と以前のリリースとの非互 換性	282
z/OS ウェアハウス・トランスフォーマー 用の環境変数データ・セット内の文字数の 削減	208	コード・ページ 1394 とユニコード間の変 換のために以前の表をインストールする	304
		ストレージ管理ビュー	305
		DB2 Administration Server	305
		環境変数およびプロファイル・レジストリ ー	308

サンプル・プラグインのコンパイルおよび 実行	311	サテライト・コントロール・サーバーのイ ンストールのエラー・メッセージの場所	398
ロックとパフォーマンス	313	データ・リカバリーと高可用性の手引き	399
カタログ統計の表	318	バックアップの概要	399
データベースの作成	324	表スペースでの変更のロールフォワード	402
ユーザー定義の一時表の作成	326	Tivoli Storage Manager	407
ツール・カタログ・データベースおよび DAS スケジューラーのセットアップと構 成	327	ドロップされた表のリカバリー	410
ツリーに複数のオブジェクトを入れるため のフォルダーの追加	332	db2Backup - データベースのバックアップ	412
フォルダーの下へのサンプル・オブジェク トの追加	335	db2ReadLogNoConnTerm - データベース接 続なしのログ読み取りの終了	420
新しい表での ID 列の定義	336	データ移動ユーティリティ ガイドおよび リファレンス	421
Enterprise Server Edition (ESE) システムの DAS 構成	337	エクスポートを使用した DB2 Data Links Manager データの移動 - 概念	421
Alter アクションの追加	338	エクスポートを使用した DB2 Data Links Manager データの移動	424
コントロール・センター用プラグインの開 発者向けのガイドライン	339	インポートを使用した DB2 Data Links Manager データの移動	425
「アプリケーション・ヒープ・サイズ」構 成パラメーター - applheapsz	340	ロードを使用した DB2 Data Links Manager データの移動	426
「アプリケーション・コントロール・ヒー プ・サイズ」構成パラメーター - app_ctl_heap_sz	342	統合システムの手引き	427
システム環境変数	343	デフォルトの逆方向データ・タイプ・マッ ピング	427
パフォーマンス変数	346	統合システムのサーバー・オプション	437
再配布ストアド・プロシージャおよび 関数	355	サーバー定義を提供する	445
「照合情報」構成パラメーター - collate_info	363	データ・タイプ・マッピング	448
「データベース・システム・モニター・ヒ ープ・サイズ」構成パラメーター - mon_heap_sz	363	透過 DDL を使用したりモート表の作成お よび変更	450
「専用メモリーしきい値」構成パラメータ ー - priv_mem_thresh	365	デフォルトのデータ・タイプ・マッピング の変更	456
汎用レジストリー変数	366	Microsoft SQL Server データ・ソースを連 合サーバーに追加する	459
SQL コンパイラー変数	371	ラージ・オブジェクト (LOB) データ・タ イプの計画	467
ストレージ管理ツール用のストアド・プ ロシージャ	377	連合パススルーの考慮事項および制約事項	468
ストレージ管理ビュー表	377	Microsoft SQL Server データ・ソースにア クセスするサーバーのセットアップ	469
サテライト管理 ガイドおよびリファレンス	392	データ・ソース・オブジェクトからデータ を削除する	475
DB2 サーバーのインストールに必要なユ ーザー・アカウント (Windows)	392	パフォーマンスに影響するシステム・モニ ター要素	476
データベースの移行	394	DB2 レプリケーションのガイドおよびリファ レンス	478
DB2 サーバーのインストールの概要 (Windows)	396	アプライ・プログラムの操作	478
		ASN.IBMSNAP_SUBS_COLS	500
		ASN.IBMSNAP_SUBS_MEMBR	502

ASN.IBMSNAP_SUBS_SET	507	イベント・タイプ	737
キャプチャー・コントロール・サーバーで 使用される表のリスト	512	すべてのデータベース・ユーザーのための システム・モニター・データへのアクセス : DB2_SNAPSHOT_NOAUTH レジストリ 一変数	738
<i>schema</i> .IBMSNAP_CAPENQ (UNIX、Windows、z/OS)	517	アプリケーション開発	741
asnanalyze: アナライザーの操作 (UNIX お よび Windows).	517	アプリケーション開発の手引き	741
<i>schema</i> .IBMSNAP_CAPTRACE (DB2 のみ)	521	Java 環境のセットアップ	741
<i>schema</i> .IBMSNAP_SIGNAL	522	マルチスレッド・データベース・アクセス の目的	742
asnccmd: キャプチャーの操作 (UNIX、Windows、z/OS)	526	Java のソースおよび出力ファイル	744
asnapply: アプライの開始 (UNIX、Windows、z/OS)	533	SQLj 用の DB2 サポート	744
asnccap: キャプチャーの開始 (UNIX、Windows、z/OS)	542	SQLj に関する DB2 の制約事項	746
asntrc: レプリケーション・トレース機能 の操作 (UNIX、Windows、z/OS).	549	SQLj プログラムのコンパイルと実行の例	747
ADDDPRSUBM: DPR サブスクリプション・ セットのメンバーの追加 (OS/400)	557	SQLj カスタマイザー・オプション	749
ADDDPRREG: DPR 登録の追加 (OS/400)	570	SQLj ルーチン内の接続コンテキスト	749
ADDDPRSUB: DPR サブスクリプション・ セットの追加 (OS/400)	580	UNIX Java 環境のセットアップ	750
RVKDPRAUT: 権限の取り消し (OS/400)	600	HP-UX Java 環境のセットアップ	751
RMVDPREG: DPR 登録の除去 (OS/400)	602	Windows Java 環境設定値	753
特殊なデータ・タイプのレプリケーション レプリケーション・プログラムの操作 (z/OS).	603 613	SQL プロシージャのプリコンパイル・ オプションとバインド・オプションのカス タマイズ	754
レプリケーション環境の保守	617	Windows SQL プロシージャ環境のセッ トアップ	755
レプリケーション・オブジェクトの命名規 則	633	Application Development Client	757
Windows Service Control Manager を使用 してシステム・コマンドを発行する方法 (Windows)	634	アプリケーションの作成と実行用として AIX でサポートされるソフトウェア	759
表およびビューをレプリケーション・ソー スとして登録する	637	アプリケーションの作成と実行用として HP-UX でサポートされるソフトウェア	761
レプリケーションの設定	663	アプリケーションの作成と実行用として Linux でサポートされるソフトウェア	762
ソースのサブスクライブ	687	アプリケーションの作成と実行用として Solaris でサポートされるソフトウェア	765
レプリケーション環境におけるデータ操作 表の概観	721 724	アプリケーションの作成と実行用として Windows でサポートされるソフトウェア	766
ASN.IBMSNAP_MONTRAIL	729	ビルド・ファイル	769
整合変更データ (CCD) 表	731	Java アプレットに関する考慮事項	772
キャプチャー・プログラムがジャーナル・ エントリ・タイプを処理する方法 (iSeries)	734	UNIX 用の SQLj アプリケーション・オ プション	774
システム・モニター ガイドおよびリファレン ス	737	UNIX の SQLj ストアード・プロシージ ャー・オプション	775
		JDBC アプレットの作成	776
		SQLj アプレットの作成	778
		SQL プロシージャの作成	780
		Java クラス・ライブラリー	781

Java でサポートされている SQL データ・タイプ	782	CLI アプリケーションでの CLI LOAD ユーティリティによるデータのインポート	870
Windows の SQLj アプリケーション・オプション	783	PATCH1 CLI/ODBC 構成キーワード	872
Windows 用の SQLj ストアード・プロシージャ・オプション	784	PATCH2 CLI/ODBC 構成キーワード	873
Windows C/C++ ルーチンのコンパイルとリンクのオプション	785	CLIPKG CLI/ODBC 構成キーワード	874
SQLj プログラムの作成	787	SQLColAttribute 関数 (CLI) - 列属性を戻す	874
HP-UX C アプリケーションのコンパイルとリンクのオプション	788	SQLDriverConnect 関数 (CLI) - データ・ソースへの (拡張) 接続	888
HP-UX C ルーチンのコンパイルとリンクのオプション	789	SQLGetData 関数 (CLI) - 列からのデータの取得	895
HP-UX C++ アプリケーションのコンパイルとリンクのオプション	791	SQLGetDescField 関数 (CLI) - 記述子レコードの単一フィールド設定の取得	905
HP-UX C++ ルーチンのコンパイルとリンクのオプション	792	SQLSetDescField 関数 (CLI) - 記述子レコードの単一フィールドの設定	910
HP-UX Micro Focus COBOL アプリケーションのコンパイルとリンクのオプション	794	SQLExtendedBind 関数 (CLI) - 列の配列のバインド	917
HP-UX Micro Focus COBOL ルーチンのコンパイルとリンクのオプション	795	SQLNextResult 関数 (CLI) - 別のステートメント・ハンドルへの次の結果セットの関連付け	923
CLI のガイドおよびリファレンス	796	管理 API 解説書	926
CLI の紹介	796	sqlcrea - データベースの作成	926
DiagIdentifier 引き数 (CLI) のヘッダー・フィールドとレコード・フィールド	798	db2AutoConfig - 自動構成	934
接続属性 (CLI) リスト	802	db2Inspect - データベースの検査	938
記述子ヘッダーとレコード・フィールドの初期設定値 (CLI)	819	db2Load - ロード	945
CLI アプリケーション用のプログラミングのヒントと提案	827	db2Runstats - 統計の実行	985
CLI アプリケーションでのコンパウンド SQL ステートメントの実行	836	db2Restore - データベースのリストア	993
CLI アプリケーションからのストアード・プロシージャの呼び出し	839	db2ReadLogNoConnInit - データベース接続なしのログ読み取りの初期設定	1005
Windows CLI ルーチンのコンパイルおよびリンク・オプション	844	db2Rollforward - データベースのロールフォワード	1007
CLI での SQL から C へのデータ変換例	845	プリコンパイラーのカスタマイズ API	1019
CLI での C から SQL へのデータ変換例	853	sqlsact - 会計情報ストリングの設定	1019
CLI アプリケーションのストリングの処理	860	アプリケーションの移行時の考慮事項	1021
DB2 CLI のサンプル	863	並行アクセスを伴うスレッド化アプリケーション	1022
HP-UX CLI アプリケーションのコンパイルおよびリンク・オプション	865	db2ReadLogNoConn - データベース接続なしのログの読み取り	1024
HP-UX CLI ルーチンのコンパイルおよびリンク・オプション	866	変更された API およびデータ構造	1028
DB2 CLI のバインド・ファイルおよびパッケージ名	868	DB2 機能の拡張 1031	
		Data Links Manager 管理ガイドおよび解説書	1031
		ファイル・システム・ディレクトリー階層を現時点の状態まで戻す	1031

データ・リンク・ファイル・マネージャ のエラー・メッセージの解説	1033
dlfm_dump コマンド	1074
Data Links Manager バージョン 8 での DCE-DFS サポートの撤回.	1076
データウェアハウス・センター管理ガイド ウェアハウス・サーバーとロガーの開始 および停止 (AIX)	1076
ウェアハウス・サーバーとロガー・デー モンの動作確認 (AIX)	1077
サポートされる DB2 以外のデータ・ソ ース	1078
DATALINK のバックス正規形式 (BNF) 仕様	1078
データウェアハウス・センター・メタデ ータの出版用にアプリケーション・ヒー プ・サイズを増大	1080
ウェアハウス・ステップ	1081
ステップ・タスク・フロー	1084
外部トリガー・プログラムを使用して、 データウェアハウス・センターの外側で のステップの実行	1085
外部トリガー・クライアントを開始する ための構文.	1086
プロセス・タスク・フロー	1088
Clean Data トランスフォーマー.	1089
ウェアハウス・メタデータの発行	1094
データウェアハウス・センター管理イン ターフェースの開始 (AIX)	1095
インフォメーション・カタログ・センター 管理ガイド.	1096
コマンド行からのタグ言語ファイルのイン ポート	1096
オブジェクト・タイプの更新.	1097
インフォメーション・カタログ・センタ ーの説明データに対して有効なデータ・ タイプ	1102
ERwin からインフォメーション・カタロ グ・センターへのマッピング.	1104
Spatial Extender User's Guide and Reference インストール・サンプル・プログラムの トラブルシューティングのヒント	1106
索引アドバイザーの使用	1107
DB2 Spatial Extender サンプル・プログ ラム	1113

XML エクステンダー 管理およびプログラ ミング	1122
XML Extender の概要	1122
SQLSTATE コードおよび関連メッセージ 番号	1124
XML Extender をバージョン 7 からバー ジョン 8 へ移行する	1129
一般解説書	1131
コマンド解説書	1131
AUTOCONFIGURE	1131
EXPORT	1134
INSPECT	1144
LOAD	1149
PRECOMPILE.	1190
db2inspf - 検査結果のフォーマット	1218
QUIESCE	1219
REORGCHK	1222
IMPORT	1234
RUNSTATS	1258
UNQUIESCE	1266
RESTORE DATABASE.	1267
UPDATE DATABASE CONFIGURATION	1276
START DATABASE MANAGER	1279
db2advis - DB2 索引アドバイザー	1286
db2fs - 最初のステップ	1288
db2look - DB2 統計および DDL 抽出ツ ール	1289
db2sqljcustomize - DB2 SQLj プロファイ ル・カスタマイザー.	1298
db2sqljprint - DB2 SQLj プロファイル・ プリンター.	1304
db2relocatedb - データベースの再配置	1305
db2support - 問題分析および環境収集ツ ール	1310
db2sqljbind - DB2 SQLj プロファイル・ バインダー.	1312
db2inidb - ミラーリングされたデータベ ースの初期化.	1315
SQL 解説書	1317
Select-statement	1317
SQLDA (SQL 記述子域)	1322
SYSCAT.ROUTINEPARMS	1334
SYSCAT.ROUTINES.	1336
SYSCAT.INDEXES	1344
SYSTAT.ROUTINES	1349

SYSSTAT.INDEXES	1351	CREATE FUNCTION (ソースまたはテンプレート)	1662
SYSCAT.FUNCMAPPINGS	1356	CREATE PROCEDURE (外部)	1672
SYSCAT.CASTFUNCTIONS	1357	CREATE PROCEDURE (SQL)	1686
SYSCAT.TRANSFORMS	1357	メッセージ解説書	1692
SYSCAT.PREDICATESPECS	1358	DBI メッセージ	1692
SYSCAT.SEQUENCES	1358	SQL0200 - SQL0299	1757
ルーチンで使用可能な SQL ステートメント	1360	SQL0400 - SQL0499	1784
DATALINK 値	1363	SQL1000 - SQL1099	1814
割り当てと比較	1364	SQL1100 - SQL1199	1840
デフォルトの順方向データ・タイプ・マッピング	1381	SQL1200 - SQL1299	1859
データ・ソース	1398	SQL1700 - SQL1799	1884
REBIND_ROUTINE_PACKAGE	1400	SQL20100 - SQL20199	1893
SYSCAT.COLIDENTATTRIBUTES	1401	SQL20200 - SQL20299	1910
ALTER TABLE	1402	SQLSTATE メッセージ	1915
ALTER TABLESPACE	1434	追加のトピックの更新	1953
CREATE TYPE (構造化)	1441	コントロール・センター	1953
CREATE VIEW	1469	ツールの呼び出し	1955
CREATE TABLE	1484	ストレージ管理ビュー	1956
CREATE METHOD	1546	管理できるオブジェクト	1957
CREATE FUNCTION (SQL スカラー、表、または行)	1552	ブラウザーをオンライン・ヘルプおよびトピックの表示用に構成	1958
ALTER PROCEDURE	1560	DB2 インフォメーション・センターの呼び出し	1958
CREATE TABLESPACE	1562	付録. 特記事項	1961
DROP	1572	商標	1964
EXECUTE	1602	索引	1967
DECLARE GLOBAL TEMPORARY TABLE	1610		
CREATE FUNCTION (外部スカラー)	1618		
CREATE FUNCTION (外部表)	1644		

リリース情報について

内容:

リリース情報には、以下のバージョン 8 の製品が含まれています。

DB2® Universal Database Personal Edition (PE)
DB2 Universal Database™ Workgroup Server Edition (WSE)
DB2 Universal Database Workgroup Server Unlimited Edition (WSUE)
DB2 Universal Database Enterprise Server Edition (ESE)
DB2 Personal Developer's Edition (PDE)
DB2 Universal Developer's Edition (UDE)
DB2 Warehouse Manager (WHM)
DB2 Warehouse Manager Sourcing Agent for z/OS™
DB2 Data Links Manager (DLM)
DB2 Net Search Extender (NSE)
DB2 Spatial Extender (SE)
DB2 Intelligent Miner™ Scoring (IM)
DB2 Intelligent Miner Modeling (IMM)
DB2 Intelligent Miner Visualization (IMV)
DB2 Connect™ Personal Edition (CPE)
DB2 Connect Enterprise Edition (CEE)
DB2 Connect Unlimited Edition (CUE)

構造:

リリース情報は、2 つのセクションに分かれています。

第 1 部「既知の制約、問題および回避処置」には、このリリースに特定の情報が含まれています。この部をお読みになり、DB2 製品ファミリーの、このリリースの既知の未解決の問題についての理解を深めてください。

第 2 部「資料の更新」には、DB2 の PDF が印刷され、資料が完成した後のすべての更新のサマリーが含まれています。最新のトピックは、ブラウザーから DB2 インフォメーション・センターにアクセスすることにより見ることができます。ブラウザーを通じて DB2 インフォメーション・センターにアクセスして、この情報を表示した場合、リビジョン・バー (l) は、PDF および印刷資料が完了した後に追加または変更されたテキストを示します。第 3 部「オンライン・ヘルプの訂正と更新」には、製品のオンライン・ヘルプにある情報の訂正および更新が含まれています。ブラウザーを通じてアクセスする DB2 インフォメーション・センターに含まれているものに加えて、更新としてリストされたトピックは、この PDF の第 4 部「Reproduced topics」にもう一度含まれています。全トピックが、相互参照を最小化するために、繰り返されています。

追加リソース:

DB2 Life Science Data Connect 製品の資料は、IBM ソフトウェア・サイト <http://www.ibm.com/software/data/db2/lifesciencesdataconnect/library.html> からダウンロードできます。

DB2 ファミリー製品の最新情報については、購読無料の *DB2 Magazine* を申し込んでください。このマガジンのオンライン版は、<http://www.db2mag.com> から入手できます。このサイトに、購読方法に関する説明もあります。

第 1 部

既知の制限、問題および予備手段

製品および製品レベルのサポート	4
バックレベル・サーバーのサポート	4
DB2 UDB バージョン 7 サーバー・アクセス	7
クラシック・コネクトは使用不可	7
データウェアハウス・センターは中国語 (簡体字) では使用しない	7
データウェアハウス・センターの下位レベル・サーバーのサポート制限	7
DB2 Administration Server	8
DB2 Workgroup Server Edition 用 DB2 ライセンス・ポリシー	8
DB2 Web ツール	8
DB2 Warehouse Manager は中国語 (簡体字) では使用しない	9
デベロップメント・センターは 64 ビット・オペレーティング・システムをサポートしない	9
Intel 32 ビット Linux オペレーティング・システム上のデベロップメント・センター	9
デベロップメント・センター・ラージ・オブジェクト (LOB) サポート (z/OS、OS/390)	9
デベロップメント・センターは Windows 98 オペレーティング・システムをサポートする	9
連合システムの制限	9
DB2 Universal Database Workgroup Server Edition に含まれる高可用性機能	12
ライセンス・センターのバックレベル・バージョンの非サポート	12
64 ビット・オペレーティング・システムで Microsoft XP のフィックスが必要	13
MVS はサポートされない	13
Net Search Extender	13
XML Extender	13
Windows XP オペレーティング・システム	13
構成アシスタント	13
サポートされないバインド・オプション	13
DB2 バックアップおよびリストア	14
Linux 390 オペレーティング・システム上のバックアップおよびリストア	14
DB2 Data Links Manager	14
Data Links サーバーのバックアップは、Tivoli Storage Manager アーカイブ・サーバー (AIX、Solaris オペレーティング環境) を使用しない	14
DB2 レプリケーション	15
DB2 データ・レプリケーション用の Java 管理 API ドキュメンテーション	15
列マッピングの制限およびレプリケーション・センター	16
iSeries システムでのレプリケーション・センターの制限	16
データウェアハウス・センター	16
リモート・オブジェクトの日本語名	16
Clean Data トランスフォーマーの制限	16
レプリケーション用のウェアハウス・エージェントの使用および Client Connect ウェアハウス・ソースへのアクセス	17
ウェアハウス・プロセスの実行間隔をスケジュールする	17
デベロップメント・センター	18

デベロップメント・センター・サーバー接続 (AIX)	18
デベロップメント・センター・オンライン・ヘルプが、IBM DB2 アプリケーション・ドライバー を IBM DB2 Java Combined Client として誤って参照する	18
ドキュメンテーション	18
DB2 レプリケーションの手引きおよび解説書ドキュメンテーション	18
DB2、バージョン 8 HTML ドキュメンテーション・インストールの制限 (Windows).	18
Microsoft Internet Explorer でのドキュメンテーション検索の予備手段.	19
インストール時のオプションにない言語の DB2 インフォメーション・センターのインストール.	19
GUI ツール	20
コントロール・センターのサンプル・プラグイン	20
DB2 GUI ツールでのインド語文字の表示	20
Linux オペレーティング・システムが稼働中の zSeries サーバーで GUI ツールはサポートされない	21
列のロードおよびインポート・ページで IXF ファイル内の DBCS 文字がサポートされない	21
ロード操作失敗時に誤ったインディケーターが表示される	21
GUI ツール用の最低ディスプレイ設定	21
製品のセンターでの SQL1224N エラー.	21
ヘルス・モニター	22
SQL アシスタント.	22
Windows オペレーティング・システムで、使用不可で表示されたときに、メイン SQL アシスタン ト・ウィンドウを使用可能にする.	22
SQL アシスタント・ボタンがコマンド・センターで使用できない	22
DB2 から起動される 2 つのバージョンの SQL アシスタント	22
追加情報	23
CLI の非同期実行	23
ユニコード・サーバー動作の変更.	23
Java 共通クライアント	23
UNIX オペレーティング・システムでの Java 関数およびルーチン.	23
翻訳版の MDAC ファイルが最初にインストールされていない場合、英語の Microsoft Data Access Components (MDAC) ファイルが DB2 V8.1 のすべての各国語版に使用される.	24

以下は、DB2[®] UDB バージョン 8 の現在知られている制限、問題および予備手段です。このセクションの情報は、DB2 UDB のバージョン 8.1 およびそのサポートする製品にのみ適用されます。制限および制約事項は、製品の将来のリリースに適用されるとは限りません。

製品および製品レベルのサポート

バックレベル・サーバーのサポート

すべてのサーバーをバージョン 8 に移行する前に、クライアント・システムをバージョン 8 に移行する必要がある場合は、いくつかの制限や制約事項があります。これらの制約事項と制限事項は DB2 Connect[™] または zSeries、OS/390、または iSeries[™] データベース・サーバーには関連しません。

バージョン 8 クライアントをバージョン 7 サーバーで処理する場合は、バージョン 7 サーバー上で DRDA[®]アプリケーション・サーバー機能を構成して使用可能にする必要があります。これを行う方法については、バージョン 7 の「インストールおよび構成補足」を参照してください。

これらの制約事項を回避するために、クライアント・システムをバージョン 8 に移行する前に、すべてのサーバーをバージョン 8 に移行する必要があります。それが不可能な場合は、バージョン 8 クライアントからバージョン 7 サーバーにアクセスするとき、以下のものがサポートされないことに注意してください。

- 以下のデータ・タイプ:
 - ラージ・オブジェクト (LOB) データ・タイプ
 - ユーザー定義特殊タイプ
 - DATALINK データ・タイプ
DATALINK データ・タイプは、非リレーショナル・ストレージに存在する外部データの管理を可能にします。 DATALINK データ・タイプは、物理的に DB2 Universal Database 以外のファイル・システムに存在するファイルを参照します。
- 以下のセキュリティ機能:
 - 認証タイプ SERVER_ENCRYPT。
SERVER_ENCRYPT は、パスワード暗号化方式です。暗号化されたパスワードとユーザー ID を使用して、ユーザーが認証されます。
 - パスワードの変更。
バージョン 8 クライアントからバージョン 7 サーバー上のパスワードを変更することはできません。
- 以下の接続および通信プロトコル:
 - 接続ではなく ATTACH を必要とするインスタンス要求
バージョン 8 クライアントからバージョン 7 サーバーへの ATTACH はサポートされていません。
 - ネットワーク・プロトコルは TCP/IP 以外
(SNA、NetBIOS、IPX/SPX など)
- 以下のアプリケーション機能およびタスク:
 - ODBC/JDBC 以外のすべてのアプリケーション用の DESCRIBE INPUT ステートメント。
バージョン 7 サーバーにアクセスする ODBC/JDBC アプリケーションを実行するバージョン 8 クライアントをサポートするには、このタイプのアクセスが必要なすべてのバージョン 7 サーバーに、 DESCRIBE INPUT サポート用の修正プログラムを適用する必要があります。この修正は、APAR IY30655 に関連しています。いずれかの DB2 ドキュメンテーション・セット (PDF または HTML) の『IBM[®]への連絡方法』を参照して、APAR IY30655 に関連する修正プログラムの入手方法を確認してください。

DESCRIBE INPUT ステートメントを使用すると、アプリケーション・リクエスターは準備済みステートメントにおける入力パラメーター・マーカについての記述を入手でき、パフォーマンスおよびユーザビリティが改善されます。CALL ステートメントの場合、これには、ストアード・プロシージャの IN および INOUT パラメーターに関連したパラメーター・マーカが含まれます。

- 2 フェーズ・コミット

バージョン 7 サーバーは、バージョン 8 クライアントが含まれる整合トランザクションを使用するときに、トランザクション・マネージャー・データベースとして使用不可であり、またバージョン 8 サーバーがトランザクション・マネージャー・データベースである可能性がある場合、バージョン 7 サーバーが整合トランザクションを使用することもできません。

- XA 準拠のトランザクション・マネージャー

バージョン 8 クライアントを使用するアプリケーションは、バージョン 7 サーバーを XA リソースとして使用できません。これには、トランザクション管理の一部となっている WebSphere、Microsoft® COM+/MTS、BEA WebLogic などが含まれます。

- モニター

- ユーティリティ

クライアントからサーバーに向けて開始されるユーティリティ。

- サイズが 32 KB を超える SQL ステートメント

バージョン 7 サーバーと連動するバージョン 8 ツールには、同様の制限や制約事項があります。

以下のバージョン 8 ツール、製品、およびセンターは、バージョン 8 サーバーのみをサポートします。

- コントロール・センター
- ヘルス・センター (Web バージョンのセンターを含む)
- 未確定トランザクション・マネージャー
- インフォメーション・カタログ・センター (Web バージョンのセンターを含む)
- ジャーナル
- ライセンス・センター
- サテライト管理センター
- Spatial Extender
- タスク・センター
- ツール設定

以下のバージョン 8 ツールは、バージョン 7 サーバー (制約事項あり) をサポートします。

- 構成アシスタント・インポート / エクスポート構成ファイル
- データウェアハウス・センター
- レプリケーション・センター
- コマンド・センター (Web バージョンのセンターを含む)
スクリプトの保管、インポート、およびスケジューリングは、コマンド・センターではサポートされていません。
- SQL アシスタント
- デベロップメント・センター
- Visual Explain

一般的に言って、コントロール・センターのナビゲーション・ツリーからのみ立ち上げられるバージョン 8 ツール、およびこれらのツールに基づく詳細ビューは、バージョン 7 以前のサーバーからアクセスおよび利用できません。バージョン 7 のツールをバージョン 7 以前のサーバーとともに使用することを考慮してください。

DB2 UDB バージョン 7 サーバー・アクセス

DB2 UDB バージョン 7 クライアントから Linux、UNIX、または Windows オペレーティング・システム上の DB2 UDB バージョン 8 サーバーにアクセスするには、サーバーにバージョン 7 フィックスパック 8 以降をインストールし、**db2updv7** コマンドを実行する必要があります。フィックスパックのインストール方法については、フィックスパックの Readme およびリリース情報の資料を参照してください。

クラシック・コネクトは使用不可

クラシック・コネクト製品は、使用できません。データウェアハウスのドキュメンテーションやその他の場所で、クラシック・コネクト製品への参照がある可能性がありますが、もはや該当しないので無視してください。

データウェアハウス・センターは中国語 (簡体字) では使用しない

データウェアハウス・センターは中国語 (簡体字) では使用できません。

データウェアハウス・センターの下位レベル・サーバーのサポート制限

DB2 Enterprise Server Edition バージョン 8 データウェアハウス・センターの下位レベル・サーバーのサポートには、以下の制限があります。

LOB サポート

- ウェアハウス・コントロール・データベースを DB2 Enterprise Server Edition バージョン 8 より古いサーバーで使用している場合、LOB での作業はできません。ウェアハウス・コントロール・データベースを正しいレベルにアップグレードするか、または DB2 Enterprise Server Edition バージョン

8 ウェアハウス・サーバーがインストールされ、システムからローカルで使用されているシステムにコントロール・データベースを移動する必要があります。

- LOB をデータウェアハウス・センターと DB2 間で移動したい場合は、DB2 Enterprise Server Edition バージョン 8 にアップグレードする必要があります。

SNA サポート

SNA を使用してウェアハウス・ソースおよびターゲットに接続する場合は、構成を SNA 上の TCP/IP に変更するか、または Windows[®] NT ウェアハウス・エージェントを使用する必要があります。

EXPORT および LOAD ユーティリティのサポート

ウェアハウス・エージェントをアップグレードする場合は、ソース・ターゲット・データベースもアップグレードするか、またはウェアハウス・プロセス内の EXPORT および LOAD ユーティリティを SQL Select および Insert ステップで置き換えなければなりません。SQL Select および Insert ステップは、DELETE* コマンド、続けて SELECT および INSERT コマンドを使用します。SQL Select および Insert ステップには、すべてのトランザクションをログに記録するデータベースが必要です。結果として、SQL Select および Insert ステップのパフォーマンスは、EXPORT および LOAD ユーティリティ用のものほど効果的ではありません。

DB2 Administration Server

AIX[®] 5L で 64 ビットのインスタンスでツール・カタログ・データベースが作成されると、DAS 移行 (dasmigr) および下位レベルのスクリプト・センターとジャーナルはサポートされません。

DB2 Workgroup Server Edition 用 DB2 ライセンス・ポリシー

「DB2 サーバーの概説およびインストール」ブックおよびライセンス・センターのオンライン・ヘルプでは別の記述がありますが、インターネット・ライセンス・ポリシーは DB2 Universal Database Workgroup Server Edition では使用できません。インターネット・ユーザーのライセンスが必要な場合は、DB2 Universal Database[™] Unlimited Workgroup Server Edition を購入する必要があります。

DB2 Web ツール

日本語、韓国語、中国語 (簡体字)、中国語 (繁体字)、ロシア語、ポーランド語では、DB2 Web ツールでサポートされるアプリケーション・サーバーはサブレット 2.3 仕様に準拠している必要があります。

DB2 Warehouse Manager は中国語 (簡体字) では使用しない

DB2 Warehouse Manager は中国語 (簡体字) では使用できません。そのため、次の従属コンポーネントもこの環境では使用できません。

- インフォメーション・カタログ・センター (DB2 Warehouse Manager のインフォメーション・カタログの管理ウィザードに従属)。
- DB2 Warehouse Manager Connector for the Web および DB2 Warehouse Manager Connector for SAP (DB2 Warehouse Manager のインストールに従属)。

デベロップメント・センターは 64 ビット・オペレーティング・システムをサポートしない

デベロップメント・センターは、64 ビット・オペレーティング・システムでは実行できません。さらに、64 ビット・オペレーティング・システムへの接続やルーチンの展開もできません。

Intel 32 ビット Linux オペレーティング・システム上のデベロップメント・センター

Intel 32 ビット Linux オペレーティング・システムで実行中の Java™ ストアード・プロシージャーをデバッグするためにデベロップメント・センターを使用することはできません。

デベロップメント・センター・ラージ・オブジェクト (LOB) サポート (z/OS、OS/390)

デベロップメント・センターは DB2 バージョン 6 (z/OS™ または OS/390® のオペレーティング・システム) では、ラージ・オブジェクト・データ・タイプ (LOB) をサポートしません。しかし、デベロップメント・センターは DB2 バージョン 7 に APAR PQ46957 が適用されているシステムでは、LOB をサポートします。

デベロップメント・センターは Windows 98 オペレーティング・システムをサポートする

デベロップメント・センターは Windows 98 オペレーティング・システムでサポートされます。

連合システムの制限

DB2 Relational Connect および DB2 Life Sciences Data Connect:

IBM は情報統合に焦点を合わせる提案を再編し、拡張しています。これらのアクティビティーには、以前から DB2 Relational Connect および DB2 Life Science Data Connect で使用可能な連合機能を置き換え、拡張する新規の機能を紹介することを含みます。詳細については、後ほど発表されます。DB2 Relational Connect バージョン 7 または

DB2 Life Science Data Connect バージョン 7 を使用して連合データ・ソースにアクセスする場合は、この新規機能が使用可能になってから、DB2 Universal Database バージョン 8.1 にアップグレードしてください。DB2 Universal Database Version 8.1 には、リレーショナル・データを IBM ファミリーのデータベース (DB2 および Informix™ IDS を含む) 間で連合させる組み込み機能があります。DB2 および Informix IDS からの連合データのみを使用したいカスタマーは、バージョン 8.1 にアップグレードできません。

DB2 バージョン 7 (UNIX および Windows 版) および DataJoiner から連合システムを移行する:

IBM は情報統合に焦点を合わせる提案を再編し、拡張しています。これらのアクティビティには、以前から DB2 Relational Connect、DB2 Life Science Data Connect および DataJoiner で使用可能な連合機能を置き換え、拡張する新機能を紹介することが含まれます。詳細については、後ほど発表されます。

- DataJoiner® バージョン 2.1.1 を使用しているお客様は、この新機能が使用可能になるまで DB2 Universal Database バージョン 8.1 にアップグレードすることができません。
- DB2 ファミリー製品および Informixにアクセスするためのみに DB2 Universal Database バージョン 7 を使用しているお客様はバージョン 7 から 8.1 へのアップグレードが可能です。他のデータ・ソースにアクセスするために DB2 Universal Database バージョン 7 を使用している場合には、新しい情報の組み込み機能が使用可能になってからアップグレードすることをお勧めします。DB2 Universal Database バージョン 8.1 にアップグレードする決定をした場合、DB2 ファミリー製品および Informix のデータ・ソースにのみアクセスすることができます。他のデータ・ソースへのラッパーは動作しません。サポートされないデータ・ソースのリストを参照してください。

ご使用のバージョン 7 連合システムの DB2 for UNIX® および Windows バージョン 8.1 への移行に関する最新情報は、DB2 Relational Connect Web サイト:
www.ibm.com/software/data/db2/relconnect/ にあります。

DB2 バージョン 7.2 (UNIX および Windows 版) 連合データベースのユーザーについて:

DB2 バージョン 8 (for UNIX and Windows) の表およびビューに、ニックネームを正常に作成するには、DB2 バージョン 7.2 (UNIX および Windows 版) フィックスパック 8 をバージョン 7.2 連合データベースに適用する必要があります。フィックスパック 8 を DB2 バージョン 7.2 (UNIX および Windows 版) 連合データベースに適用しないと、ニックネームにアクセスするとエラーが発生します。

LONG VARCHAR および LONG VARGRAPHIC サポート:

連合ドキュメンテーションは、DB2 ファミリー製品で使用される LONG VARCHAR および LONG VARGRAPHIC データ・タイプはサポートされないことを示しています。これは完全に正確だというわけではありません。LONG VARCHAR および LONG VARGRAPHIC データ・タイプ列が含まれる DB2 for UNIX および Windows のデー

| タ・ソース・オブジェクトにはニックネームを作成することができます。これらのリモ
| ート列は DB2 for UNIX および Windows の LOB データ・タイプにマップされます。
| DB2 ファミリーの他の製品については、これらのデータ・タイプを省略または作成しな
| おすビューの作成が可能で、ビューの作成後、このビューに対するニックネームを作成
| することができます。

| **WITH HOLD** カーソル:

| ニックネームまたは PASSTHRU セッションで定義されたカーソルで WITH HOLD セ
| マンティクスを使用することができます。ただし、このセマンティクスを (COMMIT 指
| 定で) 使用しようとするとうエラーを受け取ることになり、データ・ソースは WITH
| HOLD セマンティクスをサポートしません。

| **サポートされないデータ・ソース:**

| 以下のデータ・ソースは、バージョン 8.1 ではサポートされていません。

- DB2 Server (VM および VSE 版)
- Microsoft SQL Server
- ODBC
- Oracle
- Sybase
- BLAST
- Documentum
- Microsoft Excel
- 表構造ファイル
- XML タグ・ファイル

| これらのデータ・ソースは連合システムのドキュメンテーションに記述されていま
| すが、今回はサポートされていません。

| **サポートされない製品:**

| 以下の製品は、バージョン 8.1 ではサポートされていません。

- DB2 Relational Connect
- DB2 Life Sciences Data Connect

| これらの製品は連合システムのドキュメンテーションに記述されていますが、今回はサ
| ポートされていません。

| **サポートされないオペレーティング・システム:**

| 連合システムは Windows ME オペレーティング・システムではサポートされていま
| せん。

データ・ソースにアクセスする連合サーバーのセットアップ:

コンパクト・インストール・オプションでは、DB2 ファミリーまたは Informix データ・ソースへの必要なアクセスはインストールされません。DB2 ファミリー・データ・ソースにアクセスするには、標準またはカスタム・インストールを使用する必要があります。カスタム・インストールのみで、DB2 ファミリーと Informix データ・ソースの両方へのアクセスをインストールできます。

64 ビット連合サーバー:

64 ビット連合サーバーは、バージョン 8.1 ではサポートされていません。

連合システム・データベース・ディレクトリーに DB2 ファミリー・データをカタログ作成:

リモート DB2 データベースの名前が 9 文字以上の場合、DCS ディレクトリー・エントリーを作成する必要があります。

CATALOG DCS DATABASE コマンドを使用して DCS ディレクトリー内のエントリーをデータベースにカタログ作成する例は、以下のとおりです。

```
CATALOG DCS DATABASE SALES_DB2DB400 AS SALES400
```

ここで:

```
SALES_DB2DB400
```

CATALOG DATABASE コマンドで入力したリモート・データベースの名前です。

```
AS SALES400
```

カタログ作成したいターゲット・ホスト・データベースの名前です。

DB2 Universal Database Workgroup Server Edition に含まれる高可用性機能

概説およびインストール・ドキュメンテーションには、明白には記述されていませんが、DB2 Universal Database Enterprise Server Edition の高可用性機能は、DB2 Universal Database Workgroup Server Edition に含まれています。

ライセンス・センターのバックレベル・バージョンの非サポート

バージョン 7 のライセンス・センターが、バージョン 8 サーバーに接続しようとする時、ライセンス・センターは「SQL1650 - 機能はサポートされていません」というエラー・メッセージを受け取ります。これは接続がサポートされていないことを示します。

64 ビット・オペレーティング・システムで Microsoft XP のフィックスが必要

Microsoft XP オペレーティング・システム (2600) を使用していて、DB2 ファミリー製品で NETBIOS プロトコルを使用するように構成されている場合、Microsoft からホット・フィックスを入手する必要があります。Q アーティクル番号 Q317437 について、Microsoft に問い合わせてください。

MVS はサポートされない

ドキュメンテーションには書かれていますが、MVS™ はもはや DB2 Universal Database ではサポートされていません。

Net Search Extender

現時点ではまだサポートされていませんが、Net Search Extender は後続リリースで Windows .NET Server 2003 でサポートされる予定です。

XML Extender

現時点では、XML Extender はバージョン 8 と 9 の Solaris オペレーティング環境でのみサポートされています。後続リリースで、XML Extender も Windows .NET Server 2003 でサポートされる予定です。

Windows XP オペレーティング・システム

Windows XP Professional および Home Edition オペレーティング・システムは、Personal Edition 製品でのみサポートされます。

構成アシスタント

サポートされないバインド・オプション

構成アシスタントは、以下のバインド・オプションをサポートしません。

- CALL RESOLUTION
- CLIPKG
- CNULREQD
- DBPROTOCOL
- ENCODING
- MESSAGES
- OPTHINT
- OS400NAMING
- GENERIC
- IMMEDIATE

- KEEP DYNAMIC
- PATH
- SORTSEQ
- TRANSFORM_GROUP
- VALIDATE
- VARS

DB2 バックアップおよびリストア

Linux 390 オペレーティング・システム上のバックアップおよびリストア

複数の磁気テープ装置を使用するバックアップおよびリストアは、Linux 390 オペレーティング・システムを使用している場合、動作しない可能性があります。

DB2 Data Links Manager

Data Links サーバーのバックアップは、Tivoli Storage Manager アーカイブ・サーバー (AIX、Solaris オペレーティング環境) を使用しない

問題: DB2 Data Links Manager バージョン 8 のインストールまたはこのバージョンへの移行中にデータ・リンク・ファイル・マネージャー (DLFM) で開始された Tivoli® Storage Manager アーカイブ・サーバーへのデータ・リンク・サーバー・データのバックアップはできません。以下のエラー・メッセージのいずれかが画面に表示されるか、インストール状況のレポートに書き出されます。

```
DLFM129I: Automatic backup of DLFM_DB database has been triggered.  
         Please wait for the backup to complete.
```

```
DLFM901E: A system error occurred. Return code = "-2062".  
         The current command cannot be processed.  
         Refer to the db2diag.log file for additional information.
```

— または —

```
DLFM811E: The current DLFM database could not be backed up.  
         SQL code = "-2062", Return code = "-2062"
```

```
DLFM901E: A system error occurred. Return code = "-2062".  
         The current command cannot be processed. Refer to the db2diag.log file for additional information.
```

原因: DB2 Data Links Manager インストーラー・プログラムが Tivoli Storage Manager を、データ・リンクのサーバー・マシンに対するアーカイブ (バックアップ) サーバーとして使用するために必要な変数を設定できませんでした。

ヒント: Tivoli Storage Manager をアーカイブ・サーバーとして使用する予定で、DB2 Data Links Manager バージョン 8.1 をまだインストールしていない、あるいはこのバー

ジョンに移行していない場合には、この問題の発生を回避することができます。まず、インストーラー・プログラムでは "Tivoli Storage Manager" バックアップ・オプションを使用しないでください。次に、下記のステップ 2 に記述しているように、該当する Tivoli Storage Manager の変数を組み込むように Tivoli Storage Manager の管理者プロファイルを手動で構成してください。このタスクを両方とも完了したら、インストールあるいは移行を続行することができます。

予備手段: 次のタスクをリストの順に実行します。

1. 次のコマンドで DLFM データベースをバックアップします。

```
db2 backup <d1fm_db><path>
```

- <d1fm_db> は DLFM データベースの名前です。ほとんどの場合、データベースは DLFM_DB という名前です。
 - <path> は選択項目のバックアップ・ストレージ・ロケーションへのディレクトリー・パスです。
2. 該当する Tivoli Storage Manager の変数を組み込むように Tivoli Storage Manager の管理者プロファイルを手動で構成してください。手動構成の手順および必要な変数は、次のドキュメンテーションのトピックに記述があります。
 - Tivoli Storage Manager をアーカイブ・サーバーとして使用する (AIX)
 - Tivoli Storage Manager をアーカイブ・サーバーとして使用する (Solaris オペレーティング環境)

これらのトピックについては、オンラインの DB2 インフォメーション・センターまたは「DB2 Data Links Manager 管理ガイドおよびリファレンス」の『システム管理オプション』の章で確認することができます。

- DB2 Data Links Manager バージョン 8.1 の新規インストールを完了している場合は、ここで終わりです。
 - DB2 Data Links Manager バージョン 8.1 への移行の場合は、次のステップに進んでください。
3. 移行ユーティリティー・プログラム **db2dlmmg** を再実行します。

DB2 レプリケーション

DB2 データ・レプリケーション用の Java 管理 API ドキュメンテーション

DB2 DataPropagator で使用可能な管理機能を使用して、アプリケーション開発をしている場合、IBM サポートから、関係のある管理 Java API のドキュメンテーションを入手できます。

列マッピングの制限およびレプリケーション・センター

ターゲット表で IBMSNAP_SUBS_MEMBR 表の TARGET_KEY_CHG 列が「Y」に設定されている場合に、ソース表内の式をターゲット表内のキー列にマップすることはできません。これは、サブスクリプション・セット・メンバーの作成にレプリケーション・センターを使用している場合、ターゲット表のキー列がソース表の式にマップされる場合には、オプション「ターゲット・キー列を更新するために、アプライ・プログラムに変更前イメージ値を使用させる」を選択してはなりません。

iSeries システムでのレプリケーション・センターの制限

レプリケーション・センターの使用中に、iSeries システムでは IASP で管理用タスクは実行できません。

データウェアハウス・センター

リモート・オブジェクトの日本語名

日本語のリモート・ソースのスキーマ、表、および列名には、含めることのできない文字があります。ユニコード・マッピングの差が原因で、名前が NULL になる場合があります。詳細については、<http://www.ingrid.org/java/i18n/encoding/ja-conv.html> を参照してください。

Clean Data トランスフォーマーの制限

リンクの制限:

表またはビューなどの OS/390 データ・リソースを、新規の Clean Data ステップにリンクすることはできません。OS/390 データ・リソースを、使用すべきでないプログラム Clean Data ステップにリンクすることはできます。

パラメーターの制限:

「検索および置換」パラメーター: 新規の Clean Data トランスフォーマー規則表に検索および置換列の異なるデータ・タイプが含まれている場合、トランスフォーマーをテスト・モードにプロモートする前に、ターゲット表データ・タイプをターゲット表プロパティ・ページおよび列マッピング・ページの両方で変更する必要があります。

「Discretize」パラメーター: 新規の Clean Data トランスフォーマー規則表にバインド済みおよび置換列の異なるデータ・タイプが含まれている場合、トランスフォーマーをテスト・モードにプロモートする前に、ターゲット列データ・タイプをターゲット表プロパティ・ページおよび列マッピング・ページの両方で変更する必要があります。

iSeries プラットフォームの制限:

iSeries プラットフォームでは、新規の Clean Data トランスフォーマーはエラー処理を行いません。「すべて合致」合致タイプは、iSeries プラットフォーム上でのみ生成できます。

レプリケーション用のウェアハウス・エージェントの使用および Client Connect ウェアハウス・ソースへのアクセス

レプリケーション用にウェアハウス・エージェントを使用する

ウェアハウス・コントロール・データベースがクライアント・システムに対してリモートである場合、コントロール・データベースはクライアントおよびウェアハウス・エージェント・システムの両方で、同じ名前、ユーザー ID、およびパスワードでカタログ作成する必要があります。両方のシステムでソースをカタログ作成した後で、コントロール・データベースに接続できることを確認してください。コントロール・データベースに接続できない場合は、リモート・システムの環境変数 DB2COMM が TCP/IP に設定されていて、ポート番号がクライアント・システムでカタログ作成されたノードのポート番号と合致することを確認してください。

リモート・システムのポート番号を確認するには、以下のコマンドを DB2 コマンド・プロンプトで入力します。

```
get dbm cfg | grep SVCENAME
```

ノードをカタログ作成するときに、クライアント・システムのポート番号を指定します。

ウェアハウス・エージェントを使用して **Client Connect** ウェアハウス・ソースにアクセスする

Client Connect を使用して、ウェアハウス・エージェントで定義されたウェアハウス・ソースにアクセスする場合、ソースはクライアント・システムおよびウェアハウス・エージェント・システムの両方で、同じ名前、ユーザー ID、およびパスワードでカタログ作成される必要があります。ODBC パージョンのウェアハウス・エージェントを使用している場合は、ウェアハウス・エージェント・サイトおよびクライアント・サイトの両方で、ソースを ODBC ソースとしてカタログ作成する必要があります。そうしないと、ウェアハウス・ソースにアクセスするウェアハウス・エージェントを必要とするアクションは失敗します。

ウェアハウス・プロセスの実行間隔をスケジュールする

ウェアハウス・プロセスを時間間隔で実行するようスケジュールすると、プロセスのすべての実動ステップを実行するために必要な最長の時間を判別し、その間隔を適宜にスケジュールする必要があります。プロセスがスケジュールされた時間間隔を超過した場合は、後続のすべてのプロセスが実行されないか、またはスケジュール変更されません。

デベロップメント・センター・サーバー接続 (AIX)

デベロップメント・センターを使用していて、AIX サーバーに接続している場合、デフォルト構成が並行アクセスを許可しないためにエラーが発生します。並行アクセスでのマルチスレッド・アプリケーションに関する情報については、管理 API 解説書ドキュメンテーションを参照してください。

デベロップメント・センター・オンライン・ヘルプが、IBM DB2 アプリケーション・ドライバーを IBM DB2 Java Combined Client として誤って参照する

デベロップメント・センター・オンライン・ヘルプが、IBM DB2 アプリケーション・ドライバーを IBM DB2 Java Combined Client として誤って参照します。

IBM DB2 アプリケーション・ドライバーを、分散およびローカルのクロス・プラットフォーム DB2 アクセスに使用できます。これは CLI (ODBC) の従属関係を除去するように設計されています。

ドキュメンテーション

DB2 レプリケーションの手引きおよび解説書ドキュメンテーション

「レプリケーションのガイドおよびリファレンス」のまえがきに参照されている <http://www.ibm.com/software/data/dbtools/datarepl.htm> のソリューション情報はもう存在しません。

DB2、バージョン 8 HTML ドキュメンテーション・インストールの制限 (Windows)

Windows では、DB2 バージョン 7 (またはそれ以前のバージョン) がすでにインストールされているワークステーションまたはサーバーには、DB2 バージョン 8 HTML ドキュメンテーションをインストールしないでください。インストーラーが以前のバージョンを検出し、以前の製品を除去します。

予備手段が存在します。古いバージョンの DB2 がインストールされているマシンに、DB2 バージョン 8 HTML ドキュメンテーションをインストールする必要がある場合は、インストーラーを使用しないで、DB2 バージョン 8 HTML ドキュメンテーション CD から、手動でファイルおよびディレクトリーをコピーできます。DB2 インフォメーション・センターおよび全テキスト検索が使用できますが、HTML ドキュメンテーション FixPak を適用することはできません。

Microsoft Internet Explorer でのドキュメンテーション検索の予備手段

DB2 ドキュメンテーション検索機能は、Java 2 では Internet Explorer で動作しない場合があります。Microsoft Internet Explorer を使用してドキュメンテーションを表示する場合に、Java 2 がインストールされていて、検索が動作しない場合は、Internet Explorer で Java 2 を使用不可にしてください。

1. 「ツール」 -> 「インターネット オプション...」を選択します。「インターネット オプション」ノートブックが表示されます。
2. 「詳細設定」タブをクリックします。
3. 「<アプレット> に Java 2 を使用する」チェック・ボックスのチェックをはずします。
4. 「OK」をクリックします。

インストール時のオプションにない言語の DB2 インフォメーション・センターのインストール

DB2 セットアップ・ウィザードは、ウィザードがサポートしない言語（製品がサポートする言語）の DB2 HTML ドキュメンテーションのインストールをサポートしません。インストール中の製品によってサポートされている以外の言語のインフォメーション・センターをインストールするには、以下のようにします。

1. DB2 HTML ドキュメンテーション CD を CD-ROM ドライブに挿入します。
2. 以下のディレクトリーをご使用のコンピューターにコピーします。

- Windows オペレーティング・システムの場合

`d:¥lang¥`

`d:` は CD-ROM ドライブで、`lang` は使用したい言語のコードです。

- UNIX オペレーティング・システムの場合

`/cdrom/language/`

`cdrom` は CD をマウントしている場所、`language` は使用したい言語のコードです。

フォルダーはどこに置いてかまいません。DB2 HTML ドキュメンテーションは、CD から直接表示することもできます。この方法については、概説およびインストールの『DB2 HTML ドキュメンテーション CD から直接技術情報を表示』トピックを参照してください。

注:

1. ドキュメンテーションを表示するには、Microsoft Internet Explorer 5.0 以上、または Netscape 6.1 以上のブラウザを使用する必要があります。
2. この方法でドキュメンテーションをインストールすると、documentation updater は使用できません。同様に、ドキュメンテーションを製品から立ち上げると、手動でコピ

一したドキュメンテーションではなく、製品のインストールの一部としてインストールされたドキュメンテーションになります。

GUI ツール

コントロール・センターのサンプル・プラグイン

データベース構成ウィンドウでデフォルト・ボタンを使用不可にする、コントロール・センター・プラグイン `example6.java` は動作しません。このプラグインを使用して、インスタンス構成ダイアログのデフォルト・ボタンを使用不可にすることもできません。`Example6.java` は、`sqllib\samples\java\plugin` ディレクトリにあります。

DB2 GUI ツールでのインド語文字の表示

DB2 GUI ツールの使用中に、インド語文字の表示で問題がある場合は、必要なフォントがシステムにインストールされていない可能性があります。

DB2 Universal Database には、以下の IBM TrueType および OpenType プロポーショナル・インド語言語フォントがパッケージされています。これらは、*IBM Developer Kit 64 ビット・システム (AIX 版)*、*Java Technology Edition* バージョン 1.3.1 CD の `fonts` ディレクトリにあります。これらのフォントは、DB2 と共にのみ使用されません。これらのフォントの一般または無制限の販売、または配布を行うことはできません。

表 1. DB2 Universal Database にパッケージされたインド語のフォント

書体	重み	フォント・ファイル名
Devanagari MT for IBM	Medium	devamt.ttf
Devanagari MT for IBM	Bold	devamtb.ttf
Tamil	Medium	TamilMT.ttf
Tamil	Bold	TamilMTB.ttf
Telugu	Medium	TeluguMT.ttf
Telugu	Bold	TeleguMTB.ttf

フォントのインストールおよび `font.properties` ファイルの変更方法についての詳細については、IBM development kit for Java のドキュメンテーションの国際化対応セクションを参照してください。

さらに、以下の Microsoft 製品が、GUI ツールで使用できるインド語フォントを含んでいます。

- Microsoft Windows 2000 オペレーティング・システム
- Microsoft Windows XP オペレーティング・システム
- Microsoft Publisher

- Microsoft Office

Linux オペレーティング・システムが稼働中の zSeries サーバーで GUI ツールはサポートされない

DB2 セットアップ・ウィザードを除いて、Linux オペレーティング・システムが稼働中の zSeries™ サーバーでは、GUI ツールは使用できません。これは、クイック・ツアーなどの、インストール・ランチパッドから通常起動されるアイテムを含む制限です。

これらのシステムで GUI ツールを使用したい場合は、クライアント・システムに別のシステム構成で管理ツールをインストールし、このクライアントを使用して zSeries サーバーに接続してください。

列のロードおよびインポート・ページで IXF ファイル内の DBCS 文字がサポートされない

ロード・ウィザードまたはインポート・ノートブックを使用して、DBCS 文字を含む IXF 入力ファイルからのロードまたはインポートを設定すると、列ページは、ファイル内の列名を正しく表示しません。

ロード操作失敗時に誤ったインディケータが表示される

ロードが失敗しても、警告のみ (エラーではない) が返され、タスク・センターにタスク・アイコンが緑のチェックマークで表示されたままになります。実行したロードの成功したものを、確かめてください。

GUI ツール用の最低ディスプレイ設定

コントロール・センターなどの GUI ツールが正常に動作するには、最低 800 x 600 dpi の画面解像度、および最低 32 色の表示パレットを使用する必要があります。

製品のセンターでの SQL1224N エラー

AIX オペレーティング・システムで製品センターを使用している場合、SQL1224N エラーを受け取る可能性があります。このエラーは、DB2 のメモリー処理によって起こります。以下の予備手段で、エラーを除去できます。

手順:

AIX オペレーティング・システムで SQL1224N エラーを止めるには、以下のようになります。

1. インスタンス所有者として、以下のコマンドを実行します。

```
export EXTSHM=ON
db2set DB2ENVLIST=EXTSHM
```

2. 以下のコマンドでインスタンスを再始動します。

```
db2stop
db2start
```

インスタンスが、新規の環境変数設定で再始動されると、SQL1224N エラーは止まります。

ヘルス・モニター

未確定トランザクション存在ヘルス・インディケーター (db.indoubt_trans_exist) は、今回は使用できません。

SQL アシスタント

Windows オペレーティング・システムで、使用不可で表示されたときに、メイン SQL アシスタント・ウィンドウを使用可能にする

Windows オペレーティング・システムでは、メイン SQL アシスタント・ダイアログ・ウィンドウは、2 次 SQL アシスタント・メッセージ・ウィンドウが、他の非 SQL アシスタント・ウィンドウによって隠され、またはカバーされたとき、使用不可の状態が表示されます。メイン SQL アシスタント・ウィンドウは、メインウィンドウが再びアクティブになる前に、隠れた 2 次ウィンドウにユーザーの入力が必要なため、使用不可になります。マウスを使用しては、隠れたウィンドウにアクセスできない可能性があります。

メイン SQL アシスタント・ウィンドウを使用可能にするには、以下のようにします。

1. Alt (前面) キーを押したままにします。
2. Alt (前面) キーを押したままで、Tab キーを押します。ユーザーが切り替えられるウィンドウを表示するウィンドウが表示されます。
3. Tab キーを繰り返し押しして、SQL アシスタントの 2 次ウィンドウを選択して、Alt (前面) キーを放します。
4. SQL アシスタントの 2 次ウィンドウに入力するか、またはクローズします。メイン SQL アシスタント・ウィンドウがアクティブになります。

SQL アシスタント・ボタンがコマンド・センターで使用できない

コマンド・センターで、SQL アシスタント・ボタンは、接続が確立した時にのみ使用可能になります。

DB2 から起動される 2 つのバージョンの SQL アシスタント

DB2 バージョン 8.1 では、バージョン 7 とバージョン 8 の両方の SQL アシスタントを呼び出すことができます。バージョン 7 は DB2 データウェアハウス・センターから

起動できます。その他のすべてのセンターは、最新のバージョン 8 を起動します。製品のオンライン・ヘルプには、SQL アシスタント・バージョン 7 についての追加情報があります。

追加情報

CLI の非同期実行

CLI の非同期実行は使用できません。

ユニコード・サーバー動作の変更

バージョン 7 では、ユニコード・サーバーは接続時にアプリケーションによってグラフィック・コード・ページを無視し、UCS2 ユニコード (コード・ページ 1200) の使用を想定しました。バージョン 8 ユニコード・サーバーは、クライアントによって送信されたコード・ページを使用します。

Java 共通クライアント

Java タイプ 4 共通クライアントは、HP のデフォルト文字セット Roman8 で作成されたデータベースには接続できません。すべての SQLj アプリケーションは、タイプ 4 ドライバーを使用する JDBC アプリケーションと同様に、別の文字セットで作成されたデータベースに接続する必要があります。ご使用の LANG が "C" または "roman8" ロケールに設定されている場合、これを対応する ISO ロケールに変更する必要があります。たとえば、ご使用の LANG が de_DE.roman8 であると、次のようにして de_DE.iso88591 に変更してください:

```
export LANG=de_DE.iso88591
```

DB2 SQLj および JDBC サンプル・プログラムをタイプ 4 ドライバーで実行するには、次のコマンドを使用してサンプル・データベースを作成することができます (この例では、米国英語の ISO ロケールを使用):

```
export LANG=en_US.iso88591
db2 terminate
db2samp1
```

サンプル・データベースがすでに存在している場合、このコマンドを実行する前にドロップしておく必要があることに注意してください。

UNIX オペレーティング・システムでの Java 関数およびルーチン

JVM と UNIX プラットフォームでの DB2 との間の非互換性により、すべての Java 関数とルーチンは NOT FENCED キーワードの使用のいかんに関わらず、FENCED THREADSAFE モードで稼働します。パラメーター・スタイルを一般で定義し、パラメーター定義で LOB ロケーターを使用する Java 関数は、NOT FENCED キーワードが

サポートされるまで、機能しません。LOB LOCATORS ではなく、LOB パラメーターを使用するためには、この関数を変更する必要があります。

翻訳版の MDAC ファイルが最初にインストールされていない場合、英語の Microsoft Data Access Components (MDAC) ファイルが DB2 V8.1 のすべての各国語版に使用される

各国語版の DB2 をインストールする前に、各国語版の MDAC 2.7 をインストールしていない場合、DB2 は英語の MDAC ファイルをデフォルトでインストールします。これによって、オペレーティング・システムが英語版でなくても、Windows ODBC Data Source Administrator パネルが英語で表示されることとなります。この問題を修正するには、『MDAC 2.7 RTM - Refresh』バンドルを Microsoft の Web サイト http://www.microsoft.com/data/download_270RTM.htm からインストールすることができます。インストールする言語を選択し、必要な実行プログラムをダウンロードして実行します。これは翻訳版の ODBC Data Source Administrator ファイルをインストールするものです。

第 2 部 ドキュメンテーションの更新 (サマリー)

新着情報

以下のトピックが、製品の PDF 情報の出版以降に更新されました。以下のトピックの最新バージョンは、DB2 インフォメーション・センターにあります。

トピック・タイトル	更新内容
DB2 チュートリアル 227	リンクの更新。
ブラウザからアクセスする DB2 インフォメーション・センター 88	日本語のインフォメーション・センターから英語のインフォメーション・センターへのリンク方法の更新。
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
製品およびパッケージの変更点 87	Data Links Manager の機能は、バージョン 8 では DCE-DFS 環境には提供されていません。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。
定期的に変更される DB2 資料 89	更新方式の誤情報の除去。
JDBC ドライバーの機能強化 90	汎用ドライバー情報の更新。

管理および開発の GUI ツールの手引き

以下のトピックが、製品の PDF 情報の出版以降に更新されました。以下のトピックの最新バージョンは、DB2 インフォメーション・センターにあります。

表 2.

トピック名	更新内容
コントロール・センター 1953	GUI ツールの実行の前提条件の更新。
DB2 チュートリアル 227	リンクの更新。
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。

インストール

概説およびインストール

以下のトピックが、製品の PDF 情報の出版以降に更新されました。以下のトピックの最新バージョンは、DB2 インフォメーション・センターにあります。

表 3. DB2 Personal Edition

トピック・タイトル	更新内容
クライアントでの TCP/IP ノードのカタログ 95	クライアントに TCP/IP ノードをカタログするプロシーチャーの訂正。
構成アシスタント (CA) によるデータベース接続の構成 94	タスクに追加された追加ステップ。
コマンド行プロセッサを使用した、クライアントからサーバーへの接続の構成 97	重複情報の除去。
構成アシスタント (CA) のエクスポート機能を使用したクライアント・プロファイルの作成 98	構成アシスタントのエクスポート機能を使用したクライアント・プロファイルの作成方法の訂正。
DB2 チュートリアル 227	リンクの更新。
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
DB2 オンライン資料のインストール (UNIX) 104	CD 名の訂正。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。
DB2 Personal Edition 上のインスタンスおよびデータベースの移行 (Linux) 101	maxappls 構成パラメーターの訂正。
DB2 Personal Edition 上のデータベースの移行 (Windows) 103	maxappls 構成パラメーターの訂正。

表 4. DB2 サーバー

トピック・タイトル	更新内容
コマンド行プロセッサを使用した、クライアントからサーバーへの接続の構成 97	重複情報の除去。

表 4. DB2 サーバー (続き)

トピック・タイトル	更新内容
構成アシスタント (CA) によるデータベース接続の構成 94	タスクに追加された追加ステップ。
クライアントでの TCP/IP ノードのカタログ 95	クライアントに TCP/IP ノードをカタログするプロシーチャーの訂正。
DB2 クライアント 108	クライアントの更新に関する追加情報。
DB2 チュートリアル 227	リンクの更新。
db2setup の各国語表示 110	新規トピック。
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
DB2 サーバーのインストールの概要 (Windows) 396	Windows® 2000 Terminal Server および MDAC コンポーネント情報の更新。
DB2 オンライン資料のインストール (UNIX) 104	CD 名の訂正。
移行に関する推奨事項 106	移行の考慮事項の更新。
データベースの移行 394	maxappls 構成パラメーターの訂正。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。
db2licm コマンドによる DB2 ライセンス・ポリシーの設定 180	ライセンス・ポリシーの更新。
サポートされる DB2 インターフェース言語、ロケール、およびコード・ページ 111	新規トピック。
doc_install コマンドを使用して DB2 ドキュメンテーションをインストール (UNIX) 109	新規トピック。
DB2 サーバーのインストールに必要なユーザー・アカウント (Windows) 392	Windows 2000 オペレーティング・システムのインストール・ディレクトリー情報の更新。

表 5. DB2 クライアント

トピック・タイトル	更新内容
クライアントでの TCP/IP ノードのカタログ 95	クライアントに TCP/IP ノードをカタログするプロシーチャーの訂正。
コマンド行プロセッサーを使用した、クライアントからサーバーへの接続の構成 97	重複情報の除去。

表 5. DB2 クライアント (続き)

トピック・タイトル	更新内容
構成アシスタント (CA) によるデータベース接続の構成 94	タスクに追加された追加ステップ。
CLP によるクライアントでの名前付きパイプの構成 155	追加の関連リンク。
構成アシスタント (CA) のエクスポート機能を使用したクライアント・プロファイルの作成 98	構成アシスタントのエクスポート機能を使用したクライアント・プロファイルの作成方法の訂正。
構成アシスタント (CA) のインポート機能を使用したクライアント・プロファイルの構成 116	タスク・ステップの訂正。
DB2 クライアント 108	クライアントの更新に関する追加情報。
DB2 チュートリアル 227	リンクの更新。
NetBIOS 接続に関するクライアントの論理アダプター番号の判別 156	タスク・ステップの訂正。
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。
データベース接続のテスト 117	タスク・ステップの訂正。

表 6. DB2 Connect Personal Edition

トピック名	更新内容
構成アシスタント (CA) によるデータベース接続の構成 94	タスクに追加された追加ステップ。
構成アシスタント (CA) のインポート機能を使用したクライアント・プロファイルの構成 116	タスク・ステップの訂正。
構成アシスタント (CA) のエクスポート機能を使用したクライアント・プロファイルの作成 98	構成アシスタントのエクスポート機能を使用したクライアント・プロファイルの作成方法の訂正。
DB2 チュートリアル 227	リンクの更新。
コントロール・センターを使ったマルチサイト更新の使用可能化 93	一般訂正および追加。

表 6. DB2 Connect Personal Edition (続き)

トピック名	更新内容
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
データベース接続のテスト 117	タスク・ステップの訂正。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。

表 7. DB2 Connect Enterprise Edition

トピック名	更新内容
DB2 Connect のバインディング・データベース・ユーティリティー 119	新規トピック。
構成アシスタント (CA) のインポート機能を使用したクライアント・プロファイルの構成 116	タスク・ステップの訂正。
構成アシスタント (CA) によるデータベース接続の構成 94	タスクに追加された追加ステップ。
構成アシスタント (CA) のエクスポート機能を使用したクライアント・プロファイルの作成 98	構成アシスタントのエクスポート機能を使用したクライアント・プロファイルの作成方法の訂正。
DB2 クライアント 108	クライアントの更新に関する追加情報。
DB2 チュートリアル 227	リンクの更新。
コントロール・センターを使ったマルチサイト更新の使用可能化 93	一般訂正および追加。
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
ライセンス・センターを使用した DB2 Connect ライセンス・キーのインストール 121	画面制御のカバレッジの訂正。
db2licm コマンドを使用した DB2 Connect ライセンス・キーのインストール 122	CASE 情報の追加。
ライセンス・センターを使用したライセンス・タイプの設定 123	制約ポリシーのカバレッジの訂正。
データベース接続のテスト 117	タスク・ステップの訂正。

表 7. DB2 Connect Enterprise Edition (続き)

トピック名	更新内容
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。

表 8. Data Links Manager

トピック・タイトル	更新内容
DB2 Data Links Manager をインストールする前に (AIX) 134	マイグレーション・ユーティリティーに関する追加情報。
DB2 Data Links Manager をインストールする前に (Solaris オペレーティング環境) 124	移行ユーティリティーに関する追加情報。
DB2 Data Links Manager をインストールする前に (Windows) 143	移行ユーティリティーに関する追加情報。
DB2 チュートリアル 227	リンクの更新。
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。
V8.1 の DATALINK 列に対するレジストリー変数のサイズ 153	新規トピック
Data Links Manager バージョン 8 での DCE-DFS サポートの中止 1076	撤回特記事項。

Warehouse Manager インストールの手引き

以下のトピックが、ブックの出版以降に更新されました。最新のドキュメンテーションは、DB2 インフォメーション・センターにあります。

トピック・タイトル	更新内容
AIX、Linux、および Solaris オペレーティング環境のウェアハウス・エージェント環境の構成 190	OLAP を実行するエージェントの構成に関する追加情報。
Windows NT Websphere IBM HTTP Web サーバーへの、Web 用インフォメーション・カタログ・センターの インストール 188	ICC のインストールの訂正。

トピック・タイトル	更新内容
Web 用インフォメーション・カタログ・センターのインストール後のカスタマイズ 218	ICC の構成の訂正。
DB2 チュートリアル 227	リンクの更新。
DB2 Warehouse Manager ユーザー定義プログラム 203	追加の sendWTO 情報。
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
AIX、Linux、および Solaris オペレーティング環境のウェアハウス・エージェントのインストール 223	バージョン 7 とバージョン 8 を並行実行する方法に関する追加情報。
AIX Websphere IBM HTTP Web サーバーへの、Web 用インフォメーション・カタログ・センターのインストール 216	一般訂正。
iSeries ウェアハウス・エージェントのインストール 214	サポートされるプラットフォーム情報の更新。
z/OS ウェアハウス・エージェントのインストール 197	サポートされるプラットフォーム情報の更新。
z/OS ウェアハウス・エージェントの概要 225	サポートされるプラットフォームの情報の更新。
ウェアハウス・エージェントをインストールするための準備 194	サポートされるプラットフォーム情報の更新
z/OS ウェアハウス・トランスフォーマー用の環境変数データ・セット内の文字数の削減 208	UNIX [®] パス情報の訂正。
インフォメーション・カタログ管理ウィザードの実行 215	OS/390 でのインフォメーション・カタログの初期化に関する情報の訂正。
z/OS 版 DB2 でのウェアハウス・トランスフォーマーのセットアップ 206	UNIX パス情報の訂正。
トリガー・プログラムによるウェアハウス・ステップのスケジューリング (XTClient) 204	UNIX パス情報の訂正
z/OS ウェアハウス・エージェント・デーモンの開始 202	UNIX パス情報の訂正。
ウェアハウス・エージェントのログインの開始 213	ウェアハウス・エージェント・デーモン・ログインの開始プロシーチャーの更新。

トピック・タイトル	更新内容
エージェント・デーモンを z/OS 開始済みタスクとして開始する 226	UNIX パス情報の訂正。
サンプル IWH.environment ファイル 228	IWH.environment ファイルの CLASSPATH および PATH 設定に関する追加情報。
z/OS コンソールのステップ完了通知の設定 232	新規トピック。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。
z/OS ウェアハウス・エージェントを使用した、DataPropagator レプリケーション・アプライ・ステップ の自動化 211	UNIX パス情報の訂正

インストールおよび構成補足

以下のトピックが、ブックの出版以降に更新されました。最新のドキュメンテーションは、DB2 インフォメーション・センターにあります。

トピック・タイトル	更新内容
ホストおよび AS/400 アプリケーションから DB2 Universal Database サーバーへのアクセス 157	APPC および MVS™ 情報の除去。
コマンド行プロセッサを使用した、クライアントからサーバーへの接続の構成 97	重複情報の除去。
ディスクバリーによるデータベース接続の構成 154	タスクに追加された追加ステップ。
構成アシスタント (CA) によるデータベース接続の構成 94	タスクに追加された追加ステップ。
CLP によるクライアントでの名前付きパイプの構成 155	追加の関連リンク。
クライアントでの TCP/IP ノードのカタログ 95	クライアントに TCP/IP ノードをカタログするプロシージャの訂正。
DB2 チュートリアル 227	リンクの更新。
db2setup の各国語表示 110	新規トピック。
NetBIOS 接続に関するクライアントの論理アダプター番号の判別 156	タスク・ステップの訂正。

トピック・タイトル	更新内容
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
DB2 製品の手動インストール 161	関連情報ポインターの訂正。
Merant Driver Manager および DB2 UDB バージョン 8 ODBC ドライバー (UNIX) 181	新規トピック。
START または STOP DATABASE MANAGER 処理中の問題 117	新規トピック。
応答ファイルのキーワード 163	応答ファイル・キーワード SATELLITE_ADMIN の訂正
db2licm コマンドによる DB2 ライセンス・ポリシーの設定 180	ライセンス・ポリシーの更新。
手動インストール後の DB2 サーバーのセットアップ 159	追加の関連リンク。
ホストまたは OS/400 から DB2 Universal Database サーバーにアクセスするためにサポートされているプロトコル 158	APPC および MVS の対象からの除去
データベース接続のテスト 117	タスク・ステップの訂正。
Tivoli Ready の使用可能性 153	新規トピック。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。
WebSphere Application Server 上での DB2 Web ツールのデプロイ 169	追加の制限情報。
WebLogic アプリケーション・サーバー上での DB2 Web ツールのデプロイ 174	追加の制限情報。
その他のアプリケーション・サーバー上での DB2 Web ツールのデプロイ 176	追加の制限情報。
DB2 製品ライセンス・ファイル 184	ライセンス情報の更新。
db2licm コマンドを使用したライセンス・タイプの設定 185	ライセンス情報の更新。
db2licm コマンドによる DB2 ライセンス・キーのインストール 185	ライセンス情報の更新。
ライセンス・センターによる DB2 ライセンス・キーのインストール 187	ライセンス情報の更新。

データベース管理

管理ガイド

以下のトピックが、ブックの出版以降に更新されました。最新のドキュメンテーションは、DB2 インフォメーション・センターにあります。

表9. プラン

トピック・タイトル	更新内容
ユーロを使用可能なコード・ページ遷移表ファイル 233	更新された変換表のリスト。
サポートされているテリトリリー・コードおよびコード・ページ 242	更新されたコード・ページ 950 の情報。
双方向特定の CCSID 270	新規にサポートされた CCSID の追加。
DB2 での Unicode のインプリメンテーション 273	コード・ページ変換に関する追加情報。
データベース・ディレクトリーとファイル 276	情報に対する一般訂正および追加。
MDC 表を作成する際の考慮事項 279	MDC 表の作成の構文訂正。
バージョン 8 のリリース間の非互換性 282	追加非互換性。
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。
DB2 チュートリアル 227	リンクの更新。
コード・ページ 1394 とユニコードの間の変換用に、直前の表をインストール 304	新規トピック。
ストレージ管理ビュー 305	新規トピック。
ストレージ管理ツール用ストアード・プロシージャ 377	新規トピック。
ストレージ管理ビュー表 377	新規トピック。

表 10. インプリメンテーション

トピック・タイトル	更新内容
DB2 Administration Server 305	今回サポートされていない、iSeries™ システムの DAS サポートの除去。
環境変数およびプロファイル・レジストリ ー 308	インスタンス・レベル・プロファイルに関する追加情報。
サンプル・プラグインのコンパイルおよび 実行 311	情報に対する一般訂正および追加。
データベース・ディレクトリーとファイル 276	情報に対する一般訂正および追加。
ロックとパフォーマンス 313	デッドロック・モニターに関する更新情報。
カタログ統計の表 318	統計の計算時の訂正。
データベースの作成 324	デッドロック・モニターに関する更新情報。
ユーザー定義の TEMPORARY 表の作成 326	追加の前提条件情報。
ツール・カタログ・データベース、DAS スケジューラーのセットアップ、および構 成 327	DAS Windows サーバー・セットアップお よびスケジューラー・プロセスに関する追 加情報。
ツリーに複数のオブジェクトを入れるため のフォルダーの追加 332	一般訂正および追加。
フォルダーの下へのサンプル・オブジェク トの追加 335	一般訂正および追加。
新しい表での ID 列の定義 336	IDENTITY 列のユニーク制約方法の説明。
Enterprise Server Edition (ESE) システムで の DAS の構成 337	DB2 UDB ESE 上の DAS 構成の記述の 訂正。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。
DB2 チュートリアル 227	リンクの更新。
Alter アクションの追加 338	一般訂正および追加。
コントロール・センター用プラグインの開 発者向けのガイドライン 339	コントロール・センター・ドキュメンテー ションの訂正。
管理ツールから DB2 インフォメーショ ン・センターにアクセスして、製品情報 を検索する 99	検索機能の情報の訂正。

表 11. パフォーマンス

トピック・タイトル	更新内容
「データベース・システム・モニター・ヒープ・サイズ」構成パラメーター - mon_heap_sz 363	一般追加および訂正。
「専用メモリーしきい値」構成パラメーター - priv_mem_thresh 365	デフォルト値の変更。
「アプリケーション・ヒープ・サイズ」構成パラメーター - applheapsz 340	デフォルト値の訂正。
「照合情報」構成パラメーター - collate_info 363	パラメーター値の取得方法の訂正。
「アプリケーション・コントロール・ヒープ・サイズ」構成パラメーター - app_ctl_heap_sz 342	デフォルト値の訂正。
カタログ統計の表 318	統計の計算時の訂正。
ロックとパフォーマンス 313	情報に対する一般訂正および追加。
システム環境変数 343	DB2ENVLIST および DB2LIBPATH の訂正。
汎用レジストリー変数 366	追加された新規変数 DB2GRAPHICUNICODESERVER。グラフィック・データを、クライアントのコード・ページの代わりにユニコードで送信するアプリケーションに使用されます。
パフォーマンス変数 346	DB2BPVARS の訂正。
データベース・ディレクトリーとファイル 276	情報に対する一般訂正および追加。
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
DB2 チュートリアル 227	リンクの更新。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。
ストアード・プロシージャーおよび機能の再配布 355	新規参照トピック
SQL コンパイラー変数 371	新規参照トピック

データ移動ユーティリティー ガイドおよびリファレンス

以下のトピックが、ブックの出版以降に更新されました。最新のドキュメンテーションは、DB2 インフォメーション・センターにあります。

トピック・タイトル	更新内容
DB2 チュートリアル 227	リンクの更新。
db2Load - ロード 945	一般訂正。
db2relocatedb - データベースの再配置コマンド 1305	追加情報。(コンテナ・パス宣言内のワイルドカードの使用を含む)
EXPORT コマンド 1134	LOB ファイル・エクスポート・パスおよび LOBSINFILE 修飾子の訂正。
IMPORT コマンド 1234	USEGRAPHICCODEPAGE および LOBSINFILE 修飾子に関する追加情報。
LOAD コマンド 1149	USEGRAPHICCODEPAGE および LOBSINFILE 修飾子、および COPY NO パラメーターに関する追加情報。
エクスポートを使用した DB2 Data Links Manager の移動 - 概念 421	今回サポートされていない、分散ファイル・システムへの参照の除去。
エクスポートを使用した DB2 Data Links Manager データの移動 424	今回サポートされていない、分散ファイル・システムへの参照の除去。
インポートを使用した DB2 Data Links Manager データの移動 425	今回サポートされていない、分散ファイル・システムへの参照の除去。
ロードを使用した DB2 Data Links Manager データの移動 426	今回サポートされていない、分散ファイル・システムへの参照の除去。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。

データ・リカバリーと高可用性のガイドおよびリファレンス

以下のトピックが、ブックの出版以降に更新されました。最新のドキュメンテーションは、DB2 インフォメーション・センターにあります。

トピック・タイトル	更新内容
バックアップの概要 399	並行バックアップ操作に関する追加情報。
DB2 チュートリアル 227	リンクの更新。
db2Backup - データベースのバックアップ 412	一般訂正。
db2Restore - データベースのリストア 993	一般訂正。

トピック・タイトル	更新内容
db2ReadLogNoConn - データベース接続なしのログの読み取り 1024	一般訂正。
db2ReadLogNoConnInit - データベース接続なしのログ読み取りの初期設定 1005	一般訂正。
db2ReadLogNoConnTerm - データベース接続なしのログ読み取りの終了 420	一般訂正。
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
ドロップされた表のリカバリー 410	ドロップされた表のリカバリーに関する追加の制限。
表スペースにおける変更のロールフォワード 402	表スペース・リカバリーに不要なログのスキップに関する新情報。
RESTORE DATABASE コマンド 1267	一般訂正。
Tivoli Storage Manager 407	64 ビット Windows システムに関する追加情報。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。

DB2 Connect ユーザーズ・ガイド

表 12.

トピック名	更新内容
DB2 チュートリアル 227	リンクの更新。
コントロール・センターを使ったマルチサイト更新の使用可能化 93	一般訂正および追加。
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。

サテライト管理 ガイドおよびリファレンス

表 13.

トピック名	更新内容
DB2 チュートリアル 227	リンクの更新。
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
DB2 サーバーのインストールの概要 (Windows) 396	Windows 2000 Terminal Server および MDAC コンポーネント情報の更新。
DB2 オンライン資料のインストール (UNIX) 104	CD 名の訂正。
Location of Error Messages for Satellite Control Server Installation 398	一般訂正。
データベースの移行 394	maxappls 構成パラメーターの訂正。
DB2 Personal Edition 上のデータベースの移行 (Windows) 103	maxappls 構成パラメーターの訂正。
移行に関する推奨事項 106	移行の考慮事項の更新。
応答ファイルのキーワード 163	応答ファイル・キーワード SATELLITE_ADMIN の訂正
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。
DB2 サーバーのインストールに必要なユーザー・アカウント (Windows) 392	Windows 2000 オペレーティング・システムのインストール・ディレクトリー情報の更新。

DB2 レプリケーションのガイドおよびリファレンス

表 14.

トピック名	更新内容
ADDDPRSUBM: DPR サブスクリプション・セットのメンバーの追加 (OS/400) 557	一般訂正
ADDDPRREG: DPR 登録の追加 (OS/400) 570	一般訂正
ADDDPRSUB: DPR サブスクリプション・セットの追加 (OS/400) 580	一般訂正

表 14. (続き)

トピック名	更新内容
ASN.IBMSNAP_SUBS_COLS 500	TARGET_NAME 列の記述の更新。
ASN.IBMSNAP_SUBS_MEMBR 502	一般更新。
ASN.IBMSNAP_MONTRAIL 729	一般訂正
ASN.IBMSNAP_SUBS_SET 507	SLEEP_MINUTES 列の NULL 値の重要度の更新。
asnanalyze: アナライザーの操作 (UNIX および Windows) 517	-la パラメーターの説明。
asnapply: アプライの開始 (UNIX、Windows、z/OS) 533	パラメーター名の訂正。
asncap: キャプチャーの開始 (UNIX、Windows、z/OS) 542	lag_limit パラメーターの説明。
asnccmd: キャプチャーの操作 (UNIX、Windows、z/OS) 526	パラメーター・リストの訂正。
asntrc: レプリケーション・トレース機能の操作 (UNIX、Windows、z/OS) 549	-db および -subsystem パラメーターの使用の説明。
整合変更データ (CCD) 表 731	IBMSNAP_APPLY_QUAL 列の訂正。
キャプチャー・プログラムがジャーナル・エントリー・タイプを処理する方法 (iSeries) 734	EJ ジャーナル・タイプの訂正。
キャプチャー・コントロール・サーバーで使用される表のリスト 512	IBMSNAP_PRUNE_LOCK 表の定義の更新。
レプリケーション環境の保守 617	一般訂正
レプリケーション環境におけるデータ操作 721	ニックネームの使用の追加制限。
レプリケーション・オブジェクトの命名規則 633	サブスクリプション・セットの命名上の規則に関する追加情報。
レプリケーション・プログラムの操作 (z/OS) 613	一般訂正
特殊なデータ・タイプのレプリケーション 603	定義済みプロシージャ制限およびユーザー定義タイプ情報の訂正。 Oracle LONG 列の追加制限。
RMVDPREG: DPR 登録の除去 (OS/400) 602	一般訂正
RVKDPRAUT: 権限の取り消し (OS/400) 600	サンプル・テキストの訂正。

表 14. (続き)

トピック名	更新内容
表およびビューをレプリケーション・ソースとして登録する 637	ASP 内ソースの追加 OS/400® 制限。変更前イメージ接頭部および Oracle LONG 列の追加制限。
.IBMSNAP_CAPENQ (UNIX、Windows、z/OS) 517	一般訂正
.IBMSNAP_CAPTRACE (DB2 のみ) 521	JOB_NAME 列の定義の更新。
.IBMSNAP_SIGNAL 522	SIGNAL_SUBTYPE 列の記述の更新。
レプリケーションの設定 663	IASP の追加 OS/400 制限。更新された z/OS USS パス情報。
ソースのサブスクライブ 687	サブスクリプション・セットのスケジューリングの追加プロシージャ。 target-key-change オプションの追加制限。
表の概観 724	キャプションの訂正。
Windows Service Control Manager を使用してシステム・コマンドを発行する方法 (Windows) 634	除去された、レプリケーション・サービスを一時停止および再開する Windows SCM の使用に関する情報。

システム・モニター ガイドおよびリファレンス

表 15.

トピック名	更新内容
すべてのデータベース・ユーザーの、システム・モニター・データへのアクセス: DB2_SNAPSHOT_NOAUTH レジストリー変数 738	新規トピック。
DB2 チュートリアル 227	リンクの更新。
イベント・タイプ 737	新規に作成されたデータベース用に定義された詳細デッドロック・イベント・モニターの名前の訂正。
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。

連合システムの手引き

以下のトピックが、ブックの出版以降に更新されました。最新のドキュメンテーションは、DB2 インフォメーション・センターにあります。

トピック・タイトル	更新内容
Microsoft SQL Server データ・ソースを連合サーバーに追加 459	一般更新
透過的 DDL を使用したリモート表の作成および変更 450	更新された記述および例。
データ・ソース 1398	除去された ALTER TYPE MAPPING ドキュメンテーション (サポート終了ステートメント)
データ・タイプ・マッピング 448	除去された ALTER TYPE MAPPING ドキュメンテーション (サポート終了ステートメント)
デフォルトの順方向データ・タイプ・マッピング 1381	更新されたデータ・タイプ・マッピング。
デフォルトの逆方向データ・タイプ・マッピング 427	更新されたデータ・タイプ・マッピング。
データ・ソース・オブジェクトからデータを削除する 475	DELETE ステートメントの訂正。
連合パススルーの考慮事項および制約事項 468	更新された制限情報。
デフォルトのデータ・タイプ・マッピングの変更 456	サポートされているデータ・タイプの訂正。
ラージ・オブジェクト (LOB) データ・タイプの計画 467	iSeries V5 に関する追加情報。
Microsoft SQL Server データ・ソースにアクセスするサーバーのセットアップ 469	一般更新。
サーバー定義を提供する 445	DBNAME の記述の訂正。
連合システムのサーバー・オプション 437	除去された ENCRYPTION オプション。
パフォーマンスに影響するシステム・モニター・エレメント 476	新規トピック。

アプリケーション開発

管理 API 解説書

表 16.

トピック名	更新内容
アプリケーション移行時の考慮事項 1021	一般訂正。
変更された API およびデータ構造 1028	正確性の更新。
DB2 チュートリアル 227	リンクの更新。
db2AutoConfig - 自動構成 934	コンポーネント名の変更を反映する更新。
db2Backup - データベースのバックアップ 412	一般訂正。
db2Inspect - データベースの検査 938	除去された iCatalogToTablespace パラメーター。
db2Load - ロード 945	一般訂正。
db2ReadLogNoConnInit - データベース接続なしのログ読み取りの初期設定 1005	一般訂正。
db2ReadLogNoConnTerm - データベース接続なしのログ読み取りの終了 420	一般訂正。
db2Restore - データベースのリストア 993	一般訂正。
db2Rollforward - データベースのロールフォワード 1007	API の変更を反映するための変更。
db2Runstats - 統計の実行 985	API が索引統計を収集する方法の変更。
db2ReadLogNoConn - データベース接続なしのログの読み取り 1024	一般訂正。
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
プリコンパイラーのカスタマイズ API 1019	一般訂正。
sqlcrea - データベースの作成 926	使用上の注意の追加制限。
sqlsact - 会計情報ストリングの設定 1019	古い情報の除去。
並行アクセスを伴うスレッド化アプリケーション 1022	マルチスレッド・アプリケーションの情報の訂正。

表 16. (続き)

トピック名	更新内容
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。

CLI のガイドおよびリファレンス

以下のトピックが、ブックの出版以降に更新されました。最新のドキュメンテーションは、DB2 インフォメーション・センターにあります。

表 17. ボリューム 1

トピック・タイトル	更新内容
CLI での C から SQL へのデータ変換例 853	"タイム・スタンプ C データから SQL データへの変換" 表の更新。
CLI アプリケーションでのストアード・プロシージャの呼び出し 839	追加されたストアード・プロシージャ・コミット動作の情報。
CLIPKG CLI/ODBC 構成キーワード 874	使用法情報の更新。
DB2 CLI のバインド・ファイルおよびパッケージ名 868	追加された新規バインド・ファイル。
DB2 CLI のサンプル 863	新規 CLI ロード・サンプル tload.c。
DB2 チュートリアル 227	リンクの更新。
複合 SQL を使用した CLI アプリケーションでの SQL ステートメントの実行 836	追加された制限および更新されたタスク情報。
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
HP-UX CLI アプリケーションのコンパイルおよびリンク・オプション 865	コンパイルおよびリンク・オプション表から +DAportable を除去。
HP-UX CLI ルーチンのコンパイルおよびリンク・オプション 866	HP-UX OS 情報およびコンパイラー・サポートの更新。
CLI アプリケーションでの CLI LOAD ユーティリティーによるデータのインポート 870	インライン・コード・サンプルを除去し、関連するサンプル・リンクで置き換え。
CLI の紹介 796	DB2 アプリケーション開発 Web ページへのリンクを追加。
PATCH1 CLI/ODBC 構成キーワード 872	現行の値リストへの Web リンクの追加。
PATCH2 CLI/ODBC 構成キーワード 873	現行の値リストへの Web リンクの追加。

表 17. ボリューム 1 (続き)

トピック・タイトル	更新内容
CLI アプリケーション用のプログラミングのヒントと提案 827	DB2 アプリケーション開発 Web ページへのリンクを追加。
CLI での SQL から C へのデータ変換例 845	"タイム・スタンプ SQL データから C データへの変換" 表の更新。
CLI アプリケーションのストリングの処理 860	"ストリングのブランク埋め込み" セクションの追加。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。
Windows CLI ルーチンのコンパイルおよびリンク・オプション 844	追加の -MD リンク・オプション
Windows C/C++ ルーチンのコンパイルとリンクのオプション 785	追加の -MD リンク・オプション

表 18. ボリューム 2

トピック・タイトル	更新内容
SQLColAttribute 関数 (CLI) - 列属性を戻す 874	HY091 SQLSTATE の記述の訂正。
SQLDriverConnect 関数 (CLI) - データ・ソースへの (拡張) 接続 888	WindowHandle 関数の引き数の説明の更新。
SQLExtendedBind 関数 (CLI) - 列の配列のバインド 917	使用法情報の更新。
SQLGetData 関数 (CLI) - 列からのデータの取得 895	LOB の取扱情報の更新。
SQLGetDescField 関数 (CLI) - 記述子レコードの単一フィールド設定の取得 905	HY091 SQLSTATE の記述の訂正。
SQLNextResult 関数 (CLI) - 別のステートメント・ハンドルへの次の結果セットの関連付け 923	関数の引き数名の訂正。
SQLSetDescField 関数 (CLI) - 記述子レコードの単一フィールドの設定 910	HY091 SQLSTATE の記述の訂正。
接続属性 (CLI) リスト 802	追加された SQL_ATTR_CURRENT_PACKAGE_SET 属性。
記述子ヘッダーとレコード・フィールドの初期設定値 (CLI) 819	HY091 SQLSTATE の記述の訂正。

表 18. ボリューム 2 (続き)

トピック・タイトル	更新内容
DiagIdentifier 引き数 (CLI) のヘッダー・フィールドとレコード・フィールド 798	追加された新規ヘッダー・フィールド SQL_DIAG_RELATIVE_COST_ESTIMATE。
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。
DB2 チュートリアル 227	リンクの更新。

アプリケーション開発の手引き

表 19. クライアント・アプリケーションのプログラミング

トピック名	更新内容
SQLj に関する DB2 の制約事項 746	デフォルト・パッケージ名の生成方法に関する新規の情報。
SQLj 用の DB2 サポート 744	新規の db2sqlj コマンドによる更新。
DB2 チュートリアル 227	リンクの更新。
SQLj プログラムのコンパイルと実行の例 747	新規の db2sqlj コマンドによる更新。
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
マルチスレッド・データベース・アクセスの目的 742	C/C++ アプリケーションは、デフォルトでマルチスレッドで実行され、sqlcSetTypeCtx API は使用すべきではありません。
Java のソースおよび出力ファイル 744	新規の db2sqlj コマンドによる更新。
SQLj 変換プログラム・オプション 749	新規の db2sqlj コマンドによる更新。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。

表 20. サーバー・アプリケーションのプログラミング

トピック名	更新内容
SQLj ルーチン内の接続コンテキスト 749	新規トピック。

表 20. サーバー・アプリケーションのプログラミング (続き)

トピック名	更新内容
DB2 チュートリアル 227	リンクの更新。
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。

表 21. ビルドおよび実行

トピック名	更新内容
アプリケーションの作成と実行用として AIX でサポートされるソフトウェア 759	AIX® OS 情報およびコンパイラー情報の更新。
ビルド・ファイル 769	ファイル名情報の訂正。
JDBC アプレットの作成 776	タイプ 4 ドライバー情報の訂正。
SQLJ アプレットの作成 778	タイプ 4 ドライバー情報の訂正。
SQLJ プログラムの作成 787	db2sqljcustomize およびタイプ 4 ドライバーに関する追加情報。
SQL プロシーチャーの作成 780	CALL ステートメントの訂正。
SQL プロシーチャーのプリコンパイル・オプションとバインド・オプションのカスタマイズ 754	SQL プロシーチャー・プリコンパイルおよびバインド・オプションの訂正。
Application Development Client 757	Web サイト・リンクの訂正。
DB2 CLI のサンプル 863	新規 CLI ロード・サンプル tload.c。
DB2 チュートリアル 227	リンクの更新。
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
HP-UX C アプリケーションのコンパイルとリンクのオプション 788	C アプリケーション・コンパイルおよびリンク・オプション表から +DAportable を除去。
HP-UX C ルーチンのコンパイルとリンクのオプション 789	C ルーチン・コンパイルおよびリンク・オプション表から +DAportable を除去。
HP-UX C++ アプリケーションのコンパイルとリンクのオプション 791	C アプリケーション・コンパイルおよびリンク・オプション表から +DAportable を除去。

表 21. ビルドおよび実行 (続き)

トピック名	更新内容
HP-UX Micro Focus COBOL アプリケーションのコンパイルとリンクのオプション 794	HP-UX M.F. COBOL アプリケーション・コンパイルおよびリンク・オプション表から +DAportable を除去。
HP-UX Micro Focus COBOL ルーチンのコンパイルとリンクのオプション 795	HP-UX M.F. COBOL ルーチン・コンパイルおよびリンク・オプション表から +DAportable を除去。
HP-UX C++ ルーチンのコンパイルとリンクのオプション 792	C ルーチン・コンパイルおよびリンク・オプション表から +DAportable を除去。
アプリケーションの作成と実行用として HP-UX でサポートされるソフトウェア 761	HP-UX OS 情報およびコンパイラー・サポートの更新。
Java アプレットに関する考慮事項 772	ファイル名情報の訂正。
Java クラス・ライブラリー 781	CLASSPATH に組み込む必要のあるファイルの訂正。
アプリケーションの作成と実行用として Linux でサポートされるソフトウェア 762	Linux g++ 3.0.2. fstream パラメーター制限の追加。
HP-UX Java 環境のセットアップ 751	HP-UX 64 ビット Java™ セットアップ要件の訂正。
Java 環境のセットアップ 741	DB2 SQLJ カスタマイザー、プロファイル・プリンターおよびプロファイル・バインダーに関する追加情報。
UNIX Java 環境のセットアップ 750	パスおよびファイル情報の訂正。
Windows SQL プロシージャ環境のセットアップ 755	追加の -MD リンク・オプション
アプリケーションの作成と実行用として Solaris でサポートされるソフトウェア 765	Solaris OS およびコンパイラー情報の更新。
UNIX の SQLJ アプリケーション・オプション 774	UNIX SQLK アプリケーションおよびアプレットのビルド・オプションに関する更新された情報。
Windows の SQLJ アプリケーション・オプション 783	Windows SQLJ アプリケーションおよびアプレットのビルド・オプションに関する更新された情報。
UNIX の SQLJ ストアード・プロシージャ・オプション 775	UNIX SQLJ ルーチンのビルド・オプションに関する更新された情報。

表 21. ビルドおよび実行 (続き)

トピック名	更新内容
Windows の SQLJ ストアド・プロシージャ・オプション 784	Windows SQLJ ルーチンのビルド・オプションに関する更新された情報。
Java でサポートされている SQL データ・タイプ 782	DB2 バージョン 7 サーバーから BIGINT 値を検索する getObject() 方式の使用に関する追加情報。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。
Windows Java 環境設定値 753	ファイル情報の訂正。
アプリケーションの作成と実行用として Windows でサポートされるソフトウェア 766	Windows OS およびコンパイラ情報の更新。

DB2 機能の拡張

インフォメーション・カタログ・センター 管理ガイド

以下のトピックが、ブックの出版以降に更新されました。

トピック・タイトル	更新内容
ERwin からインフォメーション・カタログ・センターへのマッピング 1104	新規トピック。
コマンド行からのタグ言語ファイルのインポート 1096	タグ言語ファイルをインポートするコマンドの訂正。
オブジェクト・タイプの更新 1097	更新中の制約名の除去に関する情報の訂正。
インフォメーション・カタログ・センターの説明データに対して有効なデータ・タイプ 1102	BLOB および CLOB プロパティの長さ制限の訂正。

データウェアハウス・センター 管理ガイド

表 22.

トピック名	更新内容
Clean Data トランスフォーマー 1089	合計機能に関する追加情報。
Setting up connectivity for an IMS (OS/390) data source (AIX) 1080	新規トピック。
プロセス・タスク・フロー 1088	マージに関する追加情報。
ウェアハウス・メタデータの発行 1094	リモート・コントロール・データベースおよび発行に関する追加情報。更新された前提条件。
データウェアハウス・センターの外側でのステップの実行 1085	前提条件情報の訂正。
ウェアハウス・サーバーとロガーの開始および停止 (AIX) 1076	EXTSHM に関する追加情報。
データウェアハウス・センター管理インターフェースの開始 (AIX) 1095	新規トピック。

表 22. (続き)

トピック名	更新内容
ステップ・タスク・フロー 1084	ステップ・タスク・フローに関する追加情報。
サポートされる DB2 以外のデータ・ソース 1078	Oracle 8.1.5、Microsoft® SQL Server 2000 および Oracle 9i に関する追加情報。 IMS の記述の除去。
外部トリガー・クライアントを開始するための構文 1086	ファイアウォール処理の追加パラメーター。
ウェアハウス・サーバーとロガー・デーモンの動作確認 (AIX) 1077	ウェアハウス・サーバーおよびロガー・デーモンの検証方法の訂正。
ウェアハウス・ステップ 1081	従属関係検査に関する追加情報。

データウェアハウス・センター・アプリケーション統合の手引き

表 23.

トピック名	更新内容
インフォメーション・カタログ・センターの説明データに対して有効なデータ・タイプ 1102	BLOB および CLOB プロパティの長さ制限の訂正。

Spatial Extender User's Guide and Reference

以下のトピックが、ブックの出版以降に更新されました。最新のドキュメンテーションは、DB2 インフォメーション・センターにあります。

トピック・タイトル	更新内容
DB2 Spatial Extender サンプル・プログラム 1113	Windows オペレーティング・システムの制限が追加されました。
インストール・サンプル・プログラムのトラブルシューティングのヒント 1106	Windows オペレーティング・システムの制限。
索引アドバイザーの使用 1107	ANALYZE 文節の使用に関する追加の前提条件。

XML エクステンダー 管理およびプログラミング

以下のトピックが、ブックの出版以降に更新されました。最新のドキュメンテーションは、DB2 インフォメーション・センターにあります。

トピック・タイトル	更新内容
XML Extender の概要 1122	DB2DXXTEMP 環境変数を使用した一時 XML Extender ファイルのロケーションの指定に関する更新。
SQLSTATE codes 1124	新規トピック。
XML Extender をバージョン 7 からバージョン 8 へ移行する 1129	新規トピック。

Data Links Manager 管理ガイドおよび解説書

以下のトピックが、ブックの出版以降に更新されました。最新のドキュメンテーションは、DB2 インフォメーション・センターにあります。

トピック・タイトル	更新内容
ファイル・システム・ディレクトリー階層を現時点の状態まで戻す 1031	fsysadm.log ファイルの訂正。
データ・リンク・ファイル・マネージャーのエラー・メッセージの解説 1033	有効なメッセージ数の更新。
dlfm_dump コマンド 1074	新規トピック。
Data Links Manager バージョン 8 での DCE-DFS サポートの中止 1076	撤回特記事項。

Cross-area 参照

メッセージ解説書

以下のトピックが、ブックの出版以降に更新されました。最新のドキュメンテーションは、DB2 インフォメーション・センターにあります。

表 24. ボリューム 1

トピック・タイトル	更新内容
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。
DB2 チュートリアル 227	リンクの更新。
DBI messages1692	変更された DBI1400、DBI1401 および DBI1406。

表 25. ボリューム 2

トピック・タイトル	更新内容
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。
DB2 チュートリアル 227	リンクの更新。
SQL0200 - SQL0299 1757	新規理由コード 66 を SQL0270N に追加。
SQL0400 - SQL0499 1784	SQL0443N の更新。
SQL1000 - SQL1099 1814	追加された SQLCODE/SQLSTATE マッピング: SQL1011N/08004。SQL1022N を変更。
SQL1100 - SQL1199 1840	SQL1187W の更新。
SQL1200 - SQL1299 1859	SQL1274N の更新。
SQL1700 - SQL1799 1884	SQL1704N の更新。

表 25. ボリューム 2 (続き)

トピック・タイトル	更新内容
sql20100 1893	SQL20136N の更新。
sql20200 1910	追加された新規メッセージ SQL20230N。
SQLSTATE messages 1915	SQLSTATE 55047 の更新。

コマンド解説書

以下のトピックが、ブックの出版以降に更新されました。最新のドキュメンテーションは、DB2 インフォメーション・センターにあります。

トピック・タイトル	更新内容
AUTOCONFIGURE コマンド 1131	パーティション・データベースに関する追加の使用上の注意および情報。
EXPORT コマンド 1134	LOB ファイル・エクスポート・パスおよび LOBSINFILE 修飾子の訂正。
IMPORT コマンド 1234	USEGRAPHICCODEPAGE および LOBSINFILE 修飾子に関する追加情報。
INSPECT コマンド 1144	除去されたサポートされない文節。
LOAD コマンド 1149	USEGRAPHICCODEPAGE および LOBSINFILE 修飾子、および COPY NO パラメーターに関する追加情報。
PRECOMPILE コマンド 1190	除去された FEDERATED バインド・オプション (非 UNIX/Windows サーバー)。(サポートされないオプション)
QUIESCE コマンド 1219	構文訂正。
REORGCHK コマンド 1222	ACTBLK フィールドに関する追加情報。例示の訂正。
RESTORE DATABASE コマンド 1267	一般訂正。
RUNSTATS コマンド 1258	制限に関する追加情報。
START DATABASE MANAGER コマンド 1279	追加のパラメーターおよび記述 : REMOTE INSTANCE および ADMIN MODE。
UNQUIESCE コマンド 1266	構文の訂正。
UPDATE DATABASE CONFIGURATION コマンド 1276	複数パーティション環境でのコマンドの使用に関する追加情報。

トピック・タイトル	更新内容
db2advis - DB2 索引アドバイザー・コマンド 1286	-g パラメーターに関する追加情報。
db2fs - First Steps 1288	追加情報。OS/2® 情報の除去。
db2inspf - 検査結果のフォーマット・コマンド 1218	追加された -w オプション。
db2look - DB2 統計および DDL 抽出ツール・コマンド 1289	追加コマンド・オプション。
db2relocatedb - データベースの再配置コマンド 1305	追加情報。(コンテナー・パス宣言内のワイルドカードの使用を含む)
db2sqljbind - DB2 SQLj プロファイル・バインダー・コマンド 1312	新規トピック。
db2profcc - DB2 SQLj プロファイル・カスタマイザー・コマンド 1298	db2sqljcustomize は、db2profcc を置き換える新規のコマンドです。
db2profp - DB2 SQLj プロファイル・プリンター・コマンド 1304	db2sqljprint は、db2profp を置き換える新規のコマンドです。
db2support - 問題分析および環境収集ツール・コマンド 1310	-m および -HTML パラメーターに関する追加情報。
DB2 チュートリアル 227	リンクの更新。
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。
db2inidb - ミラーリングされたデータベースの初期化コマンド 1315	新規トピック。

SQL 解説書

以下のトピックが、製品の PDF 情報の出版以降に更新されました。以下のトピックの最新バージョンは、DB2 インフォメーション・センターにあります。

表 26. ボリューム 1

トピック・タイトル	更新内容
割り当てと比較 1364	DFS™ スキーム名情報の訂正。
DATALINK 値 1363	DFS スキーム名情報の訂正。
DB2 チュートリアル 227	リンクの更新。

表 26. ボリューム 1 (続き)

トピック・タイトル	更新内容
デフォルトの順方向データ・タイプ・マッピング 1381	更新されたデータ・タイプ・マッピング。
デフォルトの逆方向データ・タイプ・マッピング 427	更新されたデータ・タイプ・マッピング。
データ・ソース 1398	除去された ALTER TYPE MAPPING ドキュメンテーション (サポート終了ステートメント)
データ・タイプ・マッピング 448	除去された ALTER TYPE MAPPING ドキュメンテーション (サポート終了ステートメント)
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
REBIND_ROUTINE_PACKAGE プロシージャ 1400	新規トピック
SELECT ステートメント 1317	読み取り専用文節の記述の訂正。
ルーチンで使用可能な SQL ステートメント 1360	「SQL ステートメント」および「データ・アクセス識別表」の訂正。
SQLDA (SQL 記述子域) 1322	sqltype フィールドの記述の訂正。
SYSCAT.CASTFUNCTIONS カタログ・ビュー 1357	SPECIFICNAME 列の訂正。
SYSCAT.COLIDENTATTRIBUTES カタログ・ビュー 1401	新規トピック
SYSCAT.FUNCMAPPINGS カタログ・ビュー 1356	SPECIFICNAME 列の訂正。
SYSCAT.INDEXES カタログ・ビュー 1344	AVERAGE_SEQUENCE_FETCH_PAGES、AVERAGE_SEQUENCE_FETCH_GAP、AND AVERAGE_RANDOM_FETCH_PAGES 列の訂正。
SYSCAT.ROUTINEPARMS カタログ・ビュー 1334	SPECIFICNAME 列の訂正。
SYSCAT.ROUTINES カタログ・ビュー 1336	SPECIFICNAME 列の訂正。FEDERATED 列の記述が、使用されないように変更されました。

表 26. ボリューム 1 (続き)

トピック・タイトル	更新内容
SYSCAT.TRANSFORMS カタログ・ビュー 1357	SYSCAT.TRANSFORMS 列の記述の訂正。
SYSSTAT.ROUTINES カタログ・ビュー 1349	SPECIFICNAME 列の訂正。
SYSSTAT.INDEXES カタログ・ビュー 1351	AVERAGE_SEQUENCE_FETCH_PAGES、 AVERAGE_SEQUENCE_FETCH_GAP、 AND AVERAGE_RANDOM_FETCH_PAGES 列の訂正。
SYSCAT.PREDICATESPECS カタログ・ビュー 1358	FUNCNAME および SPECIFICNAME 列の訂正。
SYSCAT.SEQUENCES カタログ・ビュー 1358	SYSCAT.SEQUENCES カタログ・ビューの概要の訂正。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。

表 27. ボリューム 2

トピック・タイトル	更新内容
ALTER PROCEDURE ステートメント 1560	除去された FEDERATED および NOT FEDERATED キーワード (非サポート)。
ALTER TABLE ステートメント 1402	VOLATILE キーワードの記述の訂正。
ALTER TABLESPACE ステートメント 1434	コンテナ名制限の訂正。
CREATE FUNCTION (外部スカラー) ステートメント 1618	absolute_path_id パラメーター情報の訂正。除去された NOT FEDERATED キーワード (非サポート)。連合オブジェクトの制限への追加注釈。
CREATE FUNCTION (外部表) ステートメント 1644	除去された NOT FEDERATED キーワード (非サポート)。連合オブジェクトの制限への追加注釈。
CREATE FUNCTION (ソースまたはテンプレート) ステートメント 1662	除去された FEDERATED および NOT FEDERATED キーワードへの参照 (非サポート)。
CREATE FUNCTION (SQL スカラー、表、または行) ステートメント 1552	除去された FEDERATED および NOT FEDERATED キーワード (非サポート)。

表 27. ボリューム 2 (続き)

トピック・タイトル	更新内容
CREATE METHOD ステートメント 1546	除去された FEDERATED および NOT FEDERATED キーワードへの参照 (非サポート)。連合オブジェクトの制限への追加注釈。
CREATE PROCEDURE (外部) ステートメント 1672	データ・タイプおよび absolute_path_id パラメーターの訂正。除去された FEDERATED および NOT FEDERATED キーワード (非サポート)。
CREATE PROCEDURE (SQL) ステートメント 1686	データ・タイプ・パラメーター情報の訂正。除去された FEDERATED および NOT FEDERATED キーワード (非サポート)。
CREATE TABLE ステートメント 1484	多次元クラスタリングの 3 つの例の追加。除去された FEDERATED および NOT FEDERATED キーワード (非サポート)。
CREATE TYPE (構造化) ステートメント 1441	除去された FEDERATED および NOT FEDERATED キーワード (非サポート)。
CREATE VIEW ステートメント 1469	除去された FEDERATED キーワード (不必要)。追加の互換性情報。
CREATE TABLESPACE ステートメント 1562	EXTENTSIZE パラメーター、およびコンテナ名の制限の訂正。
DECLARE GLOBAL TEMPORARY TABLE ステートメント 1610	ユーザー定義一時表の要件の訂正。
DB2 チュートリアル 227	リンクの更新。
DROP ステートメント 1572	透過 DDL によって作成されたリモート表のドロップに関する追加情報。
EXECUTE ステートメント 1602	INTO キーワードの記述の訂正。
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。

追加のトピックの更新

表 28.

トピック名	更新内容
管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する 99	検索機能の情報の訂正。
マシンにインストールされている HTML 資料をアップデートする 168	タスク・ステップの訂正。
DB2 チュートリアル 227	リンクの更新。
オンライン・ヘルプおよびトピックを表示するためのブラウザの構成 1958	新規トピック。
DB2 インフォメーション・センターの呼び出し 1958	新規トピック。

さらに、オンライン・ヘルプの更新は、71 ページの『オンライン・ヘルプの修正および更新』を参照してください。

第 3 部

オンライン・ヘルプの修正および更新

SQL ストアード・プロシージャの C 環境をデベロップメント・センターで構成

サーバー上の DB2[®] for Windows[®] で作業を行っており、Visual C++ コンパイラーを使用している場合は、SQL ビルド設定を構成する必要があります。SQL ビルド・オプションを構成するまで、SQL ストアード・プロシージャをビルドすることはできません。

デベロップメント・センターのデータベース接続プロパティ・ノートブックを使用して、SQL ビルド設定を構成します。

SQL ストアード・プロシージャ用に C コンパイラー環境を構成するには、以下のようになります。

1. ノートブックの SQL ビルド設定ページで、SQL オブジェクトのビルドに使用したいコンパイラー環境を指定します。
 - 「リフレッシュ」をクリックします。
 - 「コンパイラー環境」で、Windows サーバー上の VC98¥BIN¥VCVARS32.BAT ファイルのロケーションを入力します。
2. 「OK」をクリックして、ノートブックをクローズして変更を保管します。「適用」をクリックすると、変更は保管され、プロパティの変更を継続できます。

MQSeries での作業のデベロップメント・センターの前提条件

DB2 Universal Database のバージョン 8 では、デベロップメント・センターの「MQSeries[®] 表の作成機能 (Create MQSeries[®] Table Function)」ウィザードを使用してユーザー定義関数 (UDF) を作成できます。この UDF は、簡単な SQL 照会と一緒に DB2 表に挿入されるキュー・メッセージの内容を準備します。

MQSeries は、メッセージのキューイングを管理し、アプリケーションが分散の、異機種混合の環境で通信できるようにする、ミドルウェア・システムです。MQSeries には、UDF を通じて DB2 からアクセスできます。UDF は、SQL ステートメントの一部を形成しているため、キューイングとデータベースの操作が混在できるからです。

MQSeries での作業の前提条件は、以下のとおりです。

- MQSeries 5.2 以上のインストール。
- MQSeries Application Messaging Interface (AMI) のインストール。
- DB2 MQSeries Integration Functions の使用可能化 (DB2 Universal Database[™] バージョン 7.2 リリース情報 を参照してください)。

Java ルーチンをデベロップメント・センターでコンパイル可能にするパスの設定

デベロップメント・センターは、デベロッパー・キットのバージョンをインストールするロケーションがわからないと、Java™ ルーチンをコンパイルできません。デフォルトのロケーションは、デベロップメント・センターが最初に開始されたときに、`$HOME/IBM/DB2DC/DB2DC.settings` ファイルに書き込まれます。これらを、`$USER.settings` ファイルにコピーし、ユニコード・エディターで変更するか、あるいはデフォルト・ロケーションにあるデベロッパー・キットのディレクトリへのシンボリック・リンクを作成することができます。

Runstats ダイアログ - 更新された到達情報

Runstats ノートブックを開くには、以下のようにします。

1. コントロール・センターで、表フォルダーに達するまでオブジェクト・ツリーを拡張します。
2. 表フォルダーをクリックします。存在する表が、内容ペインに表示されます。
3. 統計を実行したい表をすべて右マウス・ボタン・クリックし、ポップアップ・メニューから「統計の実行」を選択します。Runstats ノートブックが開きます。

クレンジング・タイプ記述の訂正

「検索および置換」および「無効値のエンコード」の両方のクリーン・タイプは、ステップの一部として入力値を出力値としてマップするために規則表を使用します。

「検索および置換」クリーン・タイプは、2 つの列を持つ規則表を使用し、値間の同等なりレーションシップを定義します。たとえば、入力として使用した製品コード番号は、それぞれ出力としたい製品名と同じになります。ソース表のクレンジング中、「検索および置換」クリーン・タイプは、検出された製品コード番号を同等の製品名に置き換えます。全ストリング、およびサブストリングの両方が置き換えられます。入力レコードが規則表の値のどれとも一致しない場合、未変更のまま渡されるか（デフォルトの動作）、または「処理オプション (Processing Options)」ページで定義された、別のエラー表に書き込まれます。

「無効値のエンコード」クリーン・タイプもまた、規則表を使用しますが、この場合は有効な入力と考えられる値をリストします。入力レコードに、規則、表中にある値がある場合、レコードは未変更の出力で渡されます。しかし、レコードが規則表の値のどれとも一致しない場合、レコードは渡されますが、一致しない値は定数と置き換えられます。ユーザーは使用したい定数を GUI に指定します。たとえば、製品コードのリストがあり、関連するセールス・データが入力表だとします。有効なコードのリストを指定し、表をクリーンします。一致しない製品のレコードでは、リストの製品コード情報は“UNKNOWN PRODUCT”または、ユーザーが GUI で指定したストリングで置き換えられます。規則表に一致するすべての値は、未変更のまま渡されます。

Spatial Extender - 索引アドバイザー使用時の要件

ANALYZE 文節は、ユーザー一時表スペースの使用が必要です。ANALYZE 文節が必要な場合、索引アドバイザーを使用するために、表スペースに USE 特権がなければなりません。

Java ストアード・プロシージャのビルド・オプションをデベロップメント・センターで指定

ストアード・プロシージャ・プロパティ・ノートブックを使用して、Java ストアード・プロシージャのビルド時に使用されるコンパイル・オプションを指定します。

これらのステップは、ストアード・プロシージャ・プロパティの変更に関するタスクのうち、より大きな方のタスクの一部です。

ストアード・プロシージャのビルド・オプションを指定するには、以下のようになります。

1. ストアード・プロシージャ・プロパティ・ノートブックの「ビルド (Build)」ページで、ストアード・プロシージャのビルドのコンパイル・オプションを指定します。使用可能なオプションについての情報は、コンパイラーのドキュメンテーションを参照してください。
 - a. 「**プリコンパイル・オプション**」フィールドに、ストアード・プロシージャのビルド時に使用したい DB2 プリコンパイラー・オプションを入力します。パッケージ名は、7 文字以下でなければなりません。
 - b. 「**コンパイル・オプション**」フィールドに、ストアード・プロシージャのビルド時に使用したいコンパイラー・オプションを入力します。
2. 「**OK**」をクリックして、ノートブックをクローズして変更を保管します。「**適用**」をクリックすると、変更は保管され、プロパティの変更を継続できます。

Web ツール - ログイン中に表示される情報

DB2 Web ツールへのログイン時に、以下の情報が表示されます。

システム

DB2 Administration Server の名前、または TCP/IP ホスト名。

インスタンス

DB2 インスタンス名、または TCP/IP ホスト名。

データベース

DB2 データベース名、またはデータベース別名。

サブシステム

zSeries™ および OS/390® システムでは、DB2 サブシステムのデータベース別名。

第 4 部 複製トピック

新着情報	87	構成アシスタント (CA) のインポート機能 を使用したクライアント・プロファイルの 構成	116
製品およびパッケージの変更点	87	データベース接続のテスト	117
ブラウザからアクセスする DB2 インフォメ ーション・センター	88	START または STOP DATABASE MANAGER 処理中のエラー	117
定期的更新される DB2 資料	89	DB2 Connect のバインディング・データ ベース・ユーティリティ	119
I JDBC ドライバーの機能強化	90	ライセンス・センターを使用した DB2 Connect ライセンス・キーのインストール .	121
インストール文書	93	db2licm コマンドを使用した DB2 Connect ライセンス・キーのインストール	122
概説およびインストール	93	ライセンス・センターを使用したライセン ス・タイプの設定	123
コントロール・センターを使ったマルチサ イト更新の使用可能化	93	DB2 Data Links Manager をインストール する前に (Solaris オペレーティング環境) .	124
構成アシスタント (CA) によるデータベー ス接続の構成	94	DB2 Data Links Manager をインストール する前に (AIX)	134
クライアントでの TCP/IP ノードのカタロ グ	95	DB2 Data Links Manager をインストール する前に (Windows)	143
コマンド行プロセッサを使用した、クラ イアントからサーバーへの接続の構成 . . .	97	V8.1 の DATALINK 列に対するレジスト リー変数のサイズ	153
構成アシスタント (CA) のエクスポート機 能を使用したクライアント・プロファイル の作成	98	インストールおよび構成補足	153
管理ツールから DB2 インフォメーショ ン・センターにアクセスして、製品情報を 検索する	99	Tivoli Ready の使用可能性	153
DB2 Personal Edition	100	ディスクバリーによるデータベース接続の 構成	154
DB2 Personal Edition 上のインスタンスお よびデータベースの移行 (Linux)	101	CLP によるクライアントでの名前付きパ イプの構成	155
DB2 Personal Edition 上のデータベースの 移行 (Windows)	103	NetBIOS 接続に関するクライアントの論 理アダプター番号の判別	156
DB2 セットアップ・ウィザードを使用し た DB2 オンライン・ドキュメンテーショ ンのインストール (UNIX)	104	ホストおよび AS/400 アプリケーションか ら DB2 Universal Database サーバーへの アクセス	157
移行に関する推奨事項	106	ホストまたは OS/400 から DB2 Universal Database サーバーにアクセスするために サポートされているプロトコル	158
DB2 クライアント	108	手動インストール後の DB2 サーバーのセ ットアップ	159
doc_install コマンドを使用した DB2 HTML ドキュメンテーションのインスト ール (UNIX)	109	DB2 製品の手動インストール	161
db2setup の各国語表示	110	応答ファイルのキーワード	163
サポートされる DB2 インターフェース言 語、ロケール、およびコード・ページ . .	111		

マシンにインストールされている HTML ドキュメンテーションをアップデートする	168	z/OS ウェアハウス・エージェントを使用 した、DataPropagator レプリケーション・ アプリ・ステップの自動化	211
WebSphere Application Server 上での DB2 Web ツールのデプロイ	169	ウェアハウス・エージェントのロギングの 開始	213
WebLogic アプリケーション・サーバー上 での DB2 Web ツールのデプロイ	174	iSeries ウェアハウス・エージェントのイ ンストール	214
その他のアプリケーション・サーバー上で の DB2 Web ツールのデプロイ	176	インフォメーション・カタログ管理ウィザ ードの実行	215
db2licm コマンドによる DB2 ライセン ス・ポリシーの設定	180	AIX Websphere IBM HTTP Web サーバ ーへの、Web 用インフォメーション・カタ ログ・センターのインストール	216
UNIX での Data Direct Technologies Driver Manager および DB2 UDB パージ ョン 8 ODBC ドライバー	181	Web 用インフォメーション・カタログ・ センターのインストール後の構成	218
応答ファイル・インストール・エラー・コ ード	183	AIX、Linux、および Solaris オペレーティ ング環境のウェアハウス・エージェントの インストール	223
DB2 製品ライセンス・ファイル	184	z/OS ウェアハウス・エージェントの概要	225
db2licm コマンドによるライセンス・タイ プの設定	185	エージェント・デーモンを z/OS 開始済み タスクとして開始する	226
db2licm コマンドによる DB2 ライセン ス・キーのインストール	185	DB2 チュートリアル	227
ライセンス・センターによる DB2 ライセ ンス・キーのインストール	187	AIX、Linux および Solaris オペレーティ ング環境用サンプル IWH.environment フ ァイル	228
Warehouse Manager インストールの手引き	188	ステップ完了の z/OS コンソール通知の設 定	232
Windows NT Websphere IBM HTTP サー バーへの Web 用インフォメーション・カ タログ・センターのインストール	188	データベース管理	233
AIX、Linux、および Solaris オペレーティ ング環境のウェアハウス・エージェント環 境の構成	190	管理ガイド	233
ウェアハウス・エージェントをインスト ールするための準備	194	ユーロを使用可能なコード・ページ遷移表 ファイル	233
z/OS ウェアハウス・エージェントのイン ストール	197	サポートされているテリトリリー・コードお よびコード・ページ	242
z/OS ウェアハウス・エージェント・デー モンの開始	202	双方向特定の CCSID	270
DB2 Warehouse Manager ユーザー定義プ ログラム	203	DB2 でのユニコードのインプリメンテー ション	273
トリガー・プログラムによるウェアハウ ス・ステップのスケジューリング		コード・ページ/CCSID 番号	275
(XTClient)	204	データベース・ディレクトリーとファイル	276
z/OS 版 DB2 でのウェアハウス・トラン スフォーマーのセットアップ	206	MDC 表を作成する際の考慮事項	279
z/OS ウェアハウス・トランスフォーマ ー用の環境変数データ・セット内の文字数の 削減	208	バージョン 8 と以前のリリースとの非互 換性	282
		システム・カタログ情報	282
		アプリケーション・プログラミング	283
		SQL	290
		データベースのセキュリティと調整	295
		ユーティリティーとツール	296

接続性と共存	298		stepwise_redistribute_dbpg ストアード・	
メッセージ	302		プロシージャ	360
構成パラメーター	302		db_partitions UDF	361
コード・ページ 1394 とユニコード間の変			使用例	361
換のために以前の表をインストールする .	304		「照合情報」構成パラメーター -	
ストレージ管理ビュー	305		collate_info	363
DB2 Administration Server	305		「データベース・システム・モニター・ヒ	
環境変数およびプロファイル・レジストリ			ープ・サイズ」構成パラメーター -	
ー	308		mon_heap_sz.	363
サンプル・プラグインのコンパイルおよび			「専用メモリーしきい値」構成パラメータ	
実行	311		ー - priv_mem_thresh.	365
ロックとパフォーマンス	313		汎用レジストリー変数	366
カタログ統計の表	318		SQL コンパイラー変数	371
データベースの作成	324		ストレージ管理ツール用のストアード・ブ	
ユーザー定義の一時表の作成	326		ロシージャ	377
ツール・カタログ・データベースおよび			ストレージ管理ビュー表	377
DAS スケジューラーのセットアップと構			サテライト管理 ガイドおよびリファレンス	392
成	327		DB2 サーバーのインストールに必要なユ	
ツリーに複数のオブジェクトを入れるため			ーザー・アカウント (Windows)	392
のフォルダーの追加	332		データベースの移行	394
フォルダーの下へのサンプル・オブジェク			DB2 サーバーのインストールの概要	
トの追加	335		(Windows)	396
新しい表での ID 列の定義	336		サテライト・コントロール・サーバーのイ	
Enterprise Server Edition (ESE) システムの			ンストールのエラー・メッセージの場所 .	398
DAS 構成	337		データ・リカバリーと高可用性の手引き .	399
Alter アクションの追加	338		バックアップの概要	399
コントロール・センター用プラグインの開			バックアップ情報の表示	402
発者向けのガイドライン	339		表スペースでの変更のロールフォワード	402
「アプリケーション・ヒープ・サイズ」構			Tivoli Storage Manager	407
成パラメーター - applheapsz	340		Tivoli Storage Manager クライアントの	
「アプリケーション・コントロール・ヒー			構成	408
プ・サイズ」構成パラメーター -			ドロップされた表のリカバリー	410
app_ctl_heap_sz.	342		db2Backup - データベースのバックアップ	412
システム環境変数	343		db2ReadLogNoConnTerm - データベース接	
パフォーマンス変数	346		続なしのログ読み取りの終了	420
再配布ストアード・プロシージャおよび			データ移動ユーティリティー ガイドおよび	
関数	355		リファレンス	421
get_swrd_settings ストアード・プロシ			エクスポートを使用した DB2 Data Links	
ジャー	356		Manager データの移動 - 概念	421
set_swrd_settings ストアード・プロシ			エクスポートを使用した DB2 Data Links	
ジャー	357		Manager データの移動	424
analyze_log_space ストアード・プロシ			インポートを使用した DB2 Data Links	
ージャー	358		Manager データの移動	425
generate_Distfile ストアード・プロシ			ロードを使用した DB2 Data Links	
ジャー	359		Manager データの移動	426
			連合システムの手引き	427

デフォルトの逆方向データ・タイプ・マッピング	427	データ・ソース・オブジェクトからデータを削除する	475
DB2 (z/OS および OS/390 版) データ・ソース	428	パフォーマンスに影響するシステム・モニター要素	476
DB2 (iSeries 版) データ・ソース	429	DB2 レプリケーションのガイドおよびリファレンス	478
DB2 サーバー (VM および VSE 版) データ・ソース	430	アプライ・プログラムの操作	478
DB2 (UNIX および Windows 版) データ・ソース	431	アプライ・プログラムの始動 (UNIX、Windows、z/OS)	478
Informix データ・ソース	432	アプライ・プログラムの始動 (OS/400)	488
Oracle SQLNET データ・ソース	433	アプライ・プログラムの停止	489
Oracle NET8 データ・ソース	434	ASNDONE 出口ルーチンの変更 (UNIX、Windows、z/OS)	490
Microsoft SQL Server データ・ソース	435	ASNDONE 出口ルーチンの変更 (OS/400)	491
Sybase データ・ソース	436	ASNLOAD 出口ルーチンを使ったターゲット表のリフレッシュ	493
連合システムのサーバー・オプション	437	ASN.IBMSNAP_SUBS_COLS	500
サーバー定義を提供する	445	ASN.IBMSNAP_SUBS_MEMBR	502
追加のサーバー・オプション	447	ASN.IBMSNAP_SUBS_SET	507
データ・タイプ・マッピング	448	キャプチャー・コントロール・サーバーで使用される表のリスト	512
透過 DDL を使用したりリモート表の作成および変更	450	<i>schema</i> .IBMSNAP_CAPENQ (UNIX、Windows、z/OS)	517
透過 DDL を使用した新規リモート表の作成	452	asnanalyze: アナライザーの操作 (UNIX および Windows)	517
透過 DDL を使用して作成されたりリモート表の変更	454	<i>schema</i> .IBMSNAP_CAPTRACE (DB2 のみ)	521
透過 DDL を使用して作成されたりリモート表のドロップ	455	<i>schema</i> .IBMSNAP_SIGNAL	522
デフォルトのデータ・タイプ・マッピングの変更	456	asnccmd: キャプチャーの操作 (UNIX、Windows、z/OS)	526
Microsoft SQL Server データ・ソースを連合サーバーに追加する	459	asnapply: アプライの開始 (UNIX、Windows、z/OS)	533
ステップ 1: 連合サーバーとデータベースを準備する	460	asncap: キャプチャーの開始 (UNIX、Windows、z/OS)	542
ステップ 2: ラッパーを作成する	461	asntrc: レプリケーション・トレース機能の操作 (UNIX、Windows、z/OS)	549
ステップ 3: サーバー定義を作成する	462	ADDDPRSUBM: DPR サブスクリプション・セットのメンバーの追加 (OS/400)	557
ステップ 4: ユーザー・マッピングを作成する	464	ADDDPRREG: DPR 登録の追加 (OS/400)	570
ステップ 5: Microsoft SQL Server リモート・サーバーへの接続をテストする	465	ADDDPRSUB: DPR サブスクリプション・セットの追加 (OS/400)	580
ステップ 6: 表およびビューにニックネームを作成する	466	RVKDPRAUT: 権限の取り消し (OS/400)	600
ラージ・オブジェクト (LOB) データ・タイプの計画	467	RMVDPRREG: DPR 登録の除去 (OS/400)	602
連合パススルーの考慮事項および制約事項	468	特殊なデータ・タイプのレプリケーション	603
Microsoft SQL Server データ・ソースにアクセスするサーバーのセットアップ	469	レプリケーションにおける一般的なデータの制約事項	603

ラージ・オブジェクトのレプリケーション	604	レプリケーション・コントロール表の セットアップ	673
DATALINK 値のレプリケーション	605	レプリケーション・プログラムのセッ トアップ	676
レプリケーション・プログラムの操作 (z/OS)	613	ジャーナルのセットアップ (OS/400)	682
レプリケーション・プログラムの操作 に JCL またはシステム開始タスクを使 用する方法 (z/OS)	613	ソースのサブスクライブ	687
レプリケーション・プログラムを自動 的に再始動するために MVS 自動リス タート・マネージャー (ARM) を使用 する方法 (z/OS)	615	ソースおよびターゲットのグループ化 の計画	687
データ共有モードへのレプリケーショ ン環境の移行 (z/OS)	617	サブスクリプション・セットの作成	690
レプリケーション環境の保守	617	サブスクリプション・セットのオブシ ョン処理	693
ソース・システムの保守	618	サブスクリプション・セット内におけ るソース表およびビューのターゲット 表およびビューへのマッピング	701
コントロール表の保守	623	ターゲット・タイプの選択	704
ターゲット表の保守	632	すべてのターゲット表タイプに共通の プロパティ	716
レプリケーション・オブジェクトの命名規 則	633	レプリケーション環境におけるデータ操作 ストアード・プロシージャまたは SQL ステートメントを使用したデータ 拡張	721
Windows Service Control Manager を使用 してシステム・コマンドを発行する方法 (Windows)	634	名前が異なるソース列とターゲット列 のマッピング	723
レプリケーション・サービスの作成	635	算出列の作成	723
レプリケーション・サービスの操作	636	表の概観	724
レプリケーション・サービスのドロッ プ	637	ASN.IBMSNAP_MONTRAIL	729
表およびビューをレプリケーション・ソー スとして登録する	637	整合変更データ (CCD) 表	731
DB2 表をソースとして登録する	638	キャプチャー・プログラムがジャーナル・ エントリー・タイプを処理する方法 (iSeries)	734
非 DB2 リレーショナル表をソースと して登録する	640	システム・モニター ガイドおよびリファレ ンス	737
ソース表用の登録オプション	641	イベント・タイプ	737
レプリケーション・ソースとしてのビ ューの動作	658	すべてのデータベース・ユーザーのための システム・モニター・データへのアクセス : DB2_SNAPSHOT_NOAUTH レジストリ 一変数	738
表のビューをソースとして登録する	661	アプリケーション開発	741
CCD 表をソースとして保守する (IMS)	662	アプリケーション開発の手引き	741
レプリケーションの設定	663	Java 環境のセットアップ	741
レプリケーション・サーバーに対する アクセスのコントロール	664	マルチスレッド・データベース・アクセス の目的	742
レプリケーション用のユーザー ID の 許可	666	Java のソースおよび出力ファイル	744
レプリケーション用のユーザー ID お よびパスワードの保管 (UNIX、Windows)	672	SQLj 用の DB2 サポート	744
		SQLj に関する DB2 の制約事項	746

SQLj プログラムのコンパイルと実行の例	747	HP-UX C++ アプリケーションのコンパイルとリンクのオプション	791
SQLj カスタマイザー・オプション	749	HP-UX C++ ルーチンのコンパイルとリンクのオプション	792
SQLj ルーチン内の接続コンテキスト	749	HP-UX Micro Focus COBOL アプリケーションのコンパイルとリンクのオプション	794
UNIX Java 環境のセットアップ	750	HP-UX Micro Focus COBOL ルーチンのコンパイルとリンクのオプション	795
HP-UX Java 環境のセットアップ	751	CLI のガイドおよびリファレンス	796
Windows Java 環境設定値	753	CLI の紹介	796
SQL プロシージャのプリコンパイル・オプションとバインド・オプションのカスタマイズ	754	DiagIdentifier 引き数 (CLI) のヘッダー・フィールドとレコード・フィールド	798
Windows SQL プロシージャ環境のセットアップ	755	接続属性 (CLI) リスト	802
Application Development Client	757	記述子ヘッダーとレコード・フィールドの初期設定値 (CLI)	819
アプリケーションの作成と実行用として AIX でサポートされるソフトウェア	759	CLI アプリケーション用のプログラミングのヒントと提案	827
アプリケーションの作成と実行用として HP-UX でサポートされるソフトウェア	761	CLI アプリケーションでのコンパウンド SQL ステートメントの実行	836
Linux でサポートされるソフトウェア	762	CLI アプリケーションからのストアード・プロシージャの呼び出し	839
アプリケーションの作成と実行用として Solaris でサポートされるソフトウェア	765	Windows CLI ルーチンのコンパイルおよびリンク・オプション	844
Windows でサポートされるソフトウェア	766	CLI での SQL から C へのデータ変換例	845
ビルド・ファイル	769	CLI での C から SQL へのデータ変換例	853
Java アプレットに関する考慮事項	772	CLI アプリケーションのストリングの処理	860
UNIX 用の SQLj アプリケーション・オプション	774	DB2 CLI のサンプル	863
UNIX の SQLj ストアード・プロシージャ・オプション	775	HP-UX CLI アプリケーションのコンパイルおよびリンク・オプション	865
JDBC アプレットの作成	776	HP-UX CLI ルーチンのコンパイルおよびリンク・オプション	866
SQLj アプレットの作成	778	DB2 CLI のバインド・ファイルおよびパッケージ名	868
SQL プロシージャの作成	780	CLI アプリケーションでの CLI LOAD コーティリティーによるデータのインポート	870
Java クラス・ライブラリー	781	PATCH1 CLI/ODBC 構成キーワード	872
Java でサポートされている SQL データ・タイプ	782	PATCH2 CLI/ODBC 構成キーワード	873
Windows の SQLj アプリケーション・オプション	783	CLIPKG CLI/ODBC 構成キーワード	874
Windows 用の SQLj ストアード・プロシージャ・オプション	784	SQLColAttribute 関数 (CLI) - 列属性を戻す	874
Windows C/C++ ルーチンのコンパイルとリンクのオプション	785	SQLDriverConnect 関数 (CLI) - データ・ソースへの (拡張) 接続	888
SQLj プログラムの作成	787	SQLGetData 関数 (CLI) - 列からのデータの取得	895
HP-UX C アプリケーションのコンパイルとリンクのオプション	788	SQLGetDescField 関数 (CLI) - 記述子レコードの単一フィールド設定の取得	905
HP-UX C ルーチンのコンパイルとリンクのオプション	789		

SQLSetDescField 関数 (CLI) - 記述子レコードの単一フィールドの設定	910	DLFM128E	1040
SQLExtendedBind 関数 (CLI) - 列の配列のバインド	917	DLFM129I	1040
SQLNextResult 関数 (CLI) - 別のステートメント・ハンドルへの次の結果セットの関連付け	923	DLFM151I	1040
管理 API 解説書	926	DLFM201E	1040
sqlecrea - データベースの作成	926	DLFM202E	1041
db2AutoConfig - 自動構成	934	DLFM203E	1041
db2Inspect - データベースの検査	938	DLFM204E	1041
db2Load - ロード	945	DLFM205E	1041
db2Runstats - 統計の実行	985	DLFM206E	1042
db2Restore - データベースのリストア	993	DLFM207E	1042
db2ReadLogNoConnInit - データベース接続なしのログ読み取りの初期設定	1005	DLFM208I	1042
db2Rollforward - データベースのロールフォワード	1007	DLFM209E	1043
プリコンパイラのカスタマイズ API	1019	DLFM210E	1043
sqlesact - 会計情報ストリングの設定	1019	DLFM211E	1043
アプリケーションの移行時の考慮事項	1021	DLFM212I	1044
並行アクセスを伴うスレッド化アプリケーション	1022	DLFM215E	1044
db2ReadLogNoConn - データベース接続なしのログの読み取り	1024	DLFM217I	1044
変更された API およびデータ構造	1028	DLFM221E	1044
DB2 機能の拡張	1031	DLFM222I	1045
Data Links Manager 管理ガイドおよび解説書	1031	DLFM223E	1045
ファイル・システム・ディレクトリー階層を現時点の状態まで戻す	1031	DLFM224I	1045
データ・リンク・ファイル・マネージャーのエラー・メッセージの解説	1033	DLFM225I	1045
DLFM001I	1036	DLFM252E	1046
DLFM002I	1036	DLFM254I	1046
DLFM003I	1036	DLFM255E	1046
DLFM101E	1036	DLFM256E	1047
DLFM102E	1037	DLFM341E	1047
DLFM103W	1037	DLFM402E	1048
DLFM104W	1038	DLFM501E	1048
DLFM121E	1038	DLFM701E	1056
DLFM122E	1039	DLFM703E	1056
DLFM123E	1039	DLFM704E	1056
DLFM124E	1039	DLFM706W	1056
DLFM126E	1039	DLFM707I	1057
		DLFM721E	1057
		DLFM741E	1057
		DLFM742E	1057
		DLFM743E	1058
		DLFM744E	1058
		DLFM746E	1059
		DLFM747E	1059
		DLFM748I	1059
		DLFM749E	1059
		DLFM750I	1060
		DLFM751E	1060
		DLFM801I	1060

DLFM802I	1061	DATALINK のバックス正規形式 (BNF)	
DLFM803I	1061	仕様	1078
DLFM804I	1061	データウェアハウス・センター・メタデ	
DLFM805I	1061	ータの出版用にアプリケーション・ヒー	
DLFM806I	1062	プ・サイズを増大	1080
DLFM807I	1062	ウェアハウス・ステップ	1081
DLFM808I	1062	ステップ・タスク・フロー	1084
DLFM809I	1063	外部トリガー・プログラムを使用して、	
DLFM810I	1063	データウェアハウス・センターの外側で	
DLFM811E	1063	のステップの実行	1085
DLFM812I	1064	外部トリガー・クライアントを開始する	
DLFM813I	1064	ための構文.	1086
DLFM814I	1064	例.	1088
DLFM815E	1064	プロセス・タスク・フロー	1088
DLFM816W	1065	スケジューリング処理に関する情報	1089
DLFM817E	1066	Clean Data トランスフォーマー.	1089
DLFM818E	1066	Clean Data トランスフォーマーの規則	1090
DLFM819I	1067	追加の機能.	1091
DLFM820E	1067	制約事項	1093
DLFM821I	1068	ウェアハウス・メタデータの発行	1094
DLFM822E	1068	データウェアハウス・センター管理イン	
DLFM823I	1069	ターフェースの開始 (AIX)	1095
DLFM841E	1069	インフォメーション・カタログ・センター	
DLFM842I	1069	管理ガイド.	1096
DLFM843E	1070	コマンド行からのタグ言語ファイルのイ	
DLFM844I	1070	ンポート	1096
DLFM845I	1070	オブジェクト・タイプの更新.	1097
DLFM846I	1071	インフォメーション・カタログ・センタ	
DLFM847I	1071	ーの説明データに対して有効なデータ・	
DLFM848I	1071	タイプ	1102
DLFM853E	1071	ERwin からインフォメーション・カタロ	
DLFM900I	1072	グ・センターへのマッピング.	1104
DLFM901E	1072	Spatial Extender User's Guide and Reference	1106
DLFM908E	1072	インストール・サンプル・プログラムの	
DLFM909E	1073	トラブルシューティングのヒント	1106
dfm_dump コマンド	1074	索引アドバイザーの使用	1107
Data Links Manager バージョン 8 での		DB2 Spatial Extender サンプル・プログ	
DCE-DFS サポートの撤回.	1076	ラム.	1113
データウェアハウス・センター管理ガイド	1076	XML エクステンダー 管理およびプログラ	
ウェアハウス・サーバーとロガーの開始		ミング	1122
および停止 (AIX)	1076	XML Extender の概要	1122
ウェアハウス・サーバーとロガー・デー		SQLSTATE コードおよび関連メッセージ	
モンの動作確認 (AIX)	1077	番号.	1124
サポートされる DB2 以外のデータ・ソ		XML Extender をバージョン 7 からバー	
ース.	1078	ジョン 8 へ移行する	1129

一般解説書	1131	SYSSTAT.ROUTINES	1349
コマンド解説書	1131	SYSSTAT.INDEXES	1351
AUTOCONFIGURE	1131	SYSCAT.FUNCMAPPINGS	1356
EXPORT	1134	SYSCAT.CASTFUNCTIONS	1357
区切り文字の制限	1142	SYSCAT.TRANSFORMS	1357
INSPECT	1144	SYSCAT.PREDICATESPECS	1358
LOAD	1149	SYSCAT.SEQUENCES	1358
PRECOMPILE	1190	ルーチンで使用可能な SQL ステートメント	1360
db2inspf - 検査結果のフォーマット	1218	DATALINK 値	1363
QUIESCE	1219	割り当てと比較	1364
REORGCHK	1222	数値割り当て	1366
IMPORT	1234	ストリング割り当て	1367
RUNSTATS	1258	日時割り当て	1370
UNQUIESCE	1266	DATALINK 割り当て	1370
RESTORE DATABASE	1267	ユーザー定義タイプの割り当て	1374
UPDATE DATABASE CONFIGURATION	1276	参照タイプの割り当て	1375
START DATABASE MANAGER	1279	数値比較	1375
db2advis - DB2 索引アドバイザー	1286	ストリングの比較	1376
db2fs - 最初のステップ	1288	日時比較	1379
db2look - DB2 統計および DDL 抽出ツール	1289	ユーザー定義タイプの比較	1380
db2sqljcustomize - DB2 SQLj プロファイル・カスタマイザー	1298	参照タイプの比較	1381
db2sqljprint - DB2 SQLj プロファイル・プリンター	1304	デフォルトの順方向データ・タイプ・マッピング	1381
db2relocatedb - データベースの再配置	1305	DB2 (z/OS および OS/390 版) データ・ソース	1382
db2support - 問題分析および環境収集ツール	1310	DB2 (iSeries 版) データ・ソース	1384
db2sqljbind - DB2 SQLj プロファイル・バインダー	1312	DB2 サーバー (VM および VSE 版) データ・ソース	1385
db2inidb - ミラーリングされたデータベースの初期化	1315	DB2 (UNIX および Windows 版) データ・ソース	1386
SQL 解説書	1317	Informix データ・ソース	1388
Select-statement	1317	Oracle SQLNET データ・ソース	1390
common-table-expression	1317	Oracle NET8 データ・ソース	1391
update-clause	1320	Microsoft SQL Server データ・ソース	1392
read-only-clause	1320	ODBC データ・ソース	1395
optimize-for-clause	1321	Sybase データ・ソース	1397
select ステートメントの例	1321	データ・ソース	1398
SQLDA (SQL 記述子域)	1322	REBIND_ROUTINE_PACKAGE	1400
SQLDA フィールド記述	1323	SYSCAT.COLIDENTATTRIBUTES	1401
SQLDA に対する DESCRIBE の効果	1329	ALTER TABLE	1402
SQLTYPE と SQLLEN	1331	ALTER TABLESPACE	1434
SYSCAT.ROUTINEPARMS	1334	CREATE TYPE (構造化)	1441
SYSCAT.ROUTINES	1336	CREATE VIEW	1469
SYSCAT.INDEXES	1344	CREATE TABLE	1484
		CREATE METHOD	1546

CREATE FUNCTION (SQL スカラー、 表、または行)	1552	クラス・コード 24 無効なカーソル状 態.	1925
ALTER PROCEDURE	1560	クラス・コード 25 無効なトランザク ション状態.	1926
CREATE TABLESPACE	1562	クラス・コード 26 無効な SQL ステ ートメント ID	1926
DROP	1572	クラス・コード 28 無効な許可指定	1926
EXECUTE	1602	クラス・コード 2D 無効なトランザク ション終了.	1926
DECLARE GLOBAL TEMPORARY TABLE	1610	クラス・コード 2E 無効な接続.	1927
CREATE FUNCTION (外部スカラー)	1618	クラス・コード 34 無効なカーソル名	1927
CREATE FUNCTION (外部表)	1644	クラス・コード 36 無効なカーソル指 定.	1927
CREATE FUNCTION (ソースまたはテン プレート)	1662	クラス・コード 38 外部関数例外	1927
CREATE PROCEDURE (外部)	1672	クラス・コード 39 外部関数呼び出し 例外.	1929
CREATE PROCEDURE (SQL)	1686	クラス・コード 3B SAVEPOINT が無 効.	1929
メッセージ解説書	1692	クラス・コード 40 トランザクショ ン・ロールバック	1930
DBI メッセージ	1692	クラス・コード 42 構文エラーまたは アクセス規則違反	1930
SQL0200 - SQL0299	1757	クラス・コード 44 WITH CHECK OPTION 違反.	1943
SQL0400 - SQL0499	1784	クラス・コード 46 Java DDL	1943
SQL1000 - SQL1099	1814	クラス・コード 51 無効なアプリケー ション状態.	1943
SQL1100 - SQL1199	1840	クラス・コード 53 無効なオペランド または矛盾する指定.	1945
SQL1200 - SQL1299	1859	クラス・コード 54 SQL または製品 の限界の超過.	1945
SQL1700 - SQL1799	1884	クラス・コード 55 前提条件の状態に ないオブジェクト	1946
SQL20100 - SQL20199	1893	クラス・コード 56 その他の SQL ま たは製品エラー	1947
SQL20200 - SQL20299	1910	クラス・コード 57 リソースが使用不 能、またはオペレーターの介入.	1949
SQLSTATE メッセージ	1915	クラス・コード 58 システム・エラー	1950
クラス・コード 00 無条件正常終了	1916	追加のトピックの更新.	1953
クラス・コード 01 警告	1916	コントロール・センター	1953
クラス・コード 02 データなし.	1920	ツールの呼び出し	1955
クラス・コード 07 動的 SQL エラー	1920	ストレージ管理ビュー	1956
クラス・コード 08 接続例外.	1921	管理できるオブジェクト	1957
クラス・コード 09 トリガー・アクシ ョン.	1921	ブラウザーをオンライン・ヘルプおよびト ピックの表示用に構成.	1958
クラス・コード 0A サポートされてい ない機能	1921		
クラス・コード 0D ターゲット・タイ プ指定が無効.	1922		
クラス・コード 0F 無効なトークン	1922		
クラス・コード 0K RESIGNAL ステ ートメントが無効	1922		
クラス・コード 20 CASE ステートメ ントにケースが見つからない.	1922		
クラス・コード 21 カーディナリティ ー違反	1923		
クラス・コード 22 データ例外.	1923		
クラス・コード 23 制約違反.	1924		

DB2 インフォメーション・センターの呼び出し 1958

新着情報

製品およびパッケージの変更点

DB2[®] UDB Enterprise Edition (EE) および DB2 UDB Enterprise-Extended Edition (EEE) は、DB2 UDB Enterprise Server Edition (ESE) という 1 つの製品にまとめられました。複数のデータベース・パーティションを作成して管理する機能は、ESE 製品とともに出荷されます。単一 SMP サーバーに複数データベース・パーティションを作成したい場合、または複数の物理サーバーにまたがって複数のデータベース・パーティションを作成したい (つまりクラスター・ハードウェア構成) 場合、データベース区分化フィーチャー・ライセンスを別に取得する必要があります。

DB2 UDB バージョン 8.1 のデータベース区分化機能により、単一 SMP サーバー上、または複数サーバーにわたって、複数のデータベース・パーティションを作成することができます。この機能は、さらにスケラビリティを拡張し、データの単一データベース・イメージを保存しつつ、複数のサーバーにまたがるデータベース・パーティションの分散を可能にしています。

DB2 UDB Workgroup Edition は、現在は DB2 UDB Workgroup Server Edition という名称になっています。

DB2 UDB Workgroup Unlimited Edition は、現在は DB2 UDB Workgroup Server Unlimited Edition という名称になっています。

以下に示す機能は、DB2 UDB Runtime Client からは取り除かれています。

- クライアント構成アシスタント
- コマンド・センター

コマンド行プロセッサが組み込まれているので、これを使って DB2 UDB Runtime Client を管理することができます。

DB2 UDB Satellite Edition バージョン 6 の機能は、DB2 UDB Personal Edition バージョン 8 に組み入れられています。既存の DB2 UDB Satellite Edition をご使用の場合は、DB2 UDB Personal Edition バージョン 8 への移行をお勧めします。

DB2 Connect[™] Application Server Edition はバージョン 8 からの新規の製品です。DB2 Connect Enterprise Edition と同様に、この製品はコネクティビティ・サーバーと JDBC、ODBC、DB2 CLI、OLE DB 用ドライバー、SQLj および組み込み SQL API などのアプリケーション・イネープリング・コンポーネントの組み合わせです。メインフレームおよびミニコンピューター・データベースから Web およびアプリケーション・サーバーへのコネクティビティを提供するように、特に設計されたこの製品は、

DB2 Connect Enterprise Edition と同じすべての機能を提供します。ただし、Web テクノロジーおよびアプリケーション・サーバーを使用するアプリケーション特有の必要性に対応したライセンスご使用条件が付帯します。DB2 Connect Application Server Edition は柔軟なデプロイメントが可能であり、スタンドアロンのコネクティビティー・サーバーに配置することも、同じサーバー・マシン上に DB2 Connect とアプリケーション・サーバーを配置することも、追加のソフトウェア・ライセンス使用料なしで行うことができます。

DB2 OLAP Starter Kit は DB2 Universal Database™ バージョン 8 では利用できません。

Data Links Manager の機能は、バージョン 8 の DCE-DFS 環境では提供されません。Data Links Manager は引き続きバージョン 7 の DFS™ サポートは提供します。バージョン 8 製品、インストール・パッケージ、メッセージ、およびドキュメンテーション内に、Data Links Manager の DFS 環境への参照がある場合がありますが、これらはもはや該当しないので無視してください。Data Links バージョン 7 を使用していて、DCE-DFS サポートを必要としている場合は、Data Links Manager バージョン 8 の別のストレージ環境を使用する必要があります。

関連概念:

- 新機能 の『サテライト管理の変更点』

ブラウザーからアクセスする DB2 インフォメーション・センター

DB2® インフォメーション・センターを使用すると、DB2 Universal Database™ および DB2 Connect™ をビジネスで最大限に活用するのに必要な情報すべてにアクセスできます。また、DB2 インフォメーション・センターでは、DB2 の主な機能およびコンポーネントが説明されています (レプリケーション、データウェアハウジング、インフォメーション・カタログ・センター、Life Sciences Data Connect、および DB2 の種々の Extender など)。

Netscape Navigator 6.1 以上または Microsoft® Internet Explorer 5 以上で表示する場合、ブラウザーからアクセスする DB2 インフォメーション・センターには、以下の機能が備えられています。以下のいくつかの機能では、Java™ または JavaScript のサポートを使用可能にする必要があります。

資料の定期的な更新

更新された HTML をダウンロードすることによってトピックを最新の状態に保つことができます。

検索 ナビゲーション・ツールバーの「**検索**」をクリックすることによって、ワークステーションにインストールされているすべてのトピックを検索できます。

統合されたナビゲーション・ツリー

1 つのナビゲーション・ツリーから、DB2 ライブラリー内のすべてのトピックを位置指定できます。ナビゲーション・ツリーは、以下に示すように、情報のタイプに基づいて編成されています。

- 「タスク」では、目的を達成するための段階的な指示が紹介されています。
- 「概念」では、対象の概要が示されています。
- 「参照」では、その対象に関する詳細な情報（ステートメントおよびコマンドの構文、メッセージ・ヘルプ、要件など）が説明されています。

マスター索引

マスター索引から *DB2 HTML* ドキュメンテーション CD からインストールされた情報にアクセスします。索引では、用語が 50 音順に編成されています。

マスター用語集

マスター用語集は、DB2 インフォメーション・センターで使用されている用語を定義しています。用語集では、用語が 50 音順に編成されています。

組み込まれているローカライズ情報

トピックの翻訳版がご使用の言語で使用できない場合、新しいブラウザのウィンドウが開き、そのトピックの英語版が表示されます。オリジナルのブラウザ・ウィンドウには、使用可能な翻訳版のトピック表示を続行します。

関連タスク:

- *Data Links Manager* 概説およびインストールの『ブラウザから DB2 インフォメーション・センターにアクセスしてトピックを検索する』
- 99 ページの『管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する』
- 168 ページの『マシンにインストールされている HTML ドキュメンテーションをアップデートする』

定期的に更新される DB2 資料

DB2® 情報ライブラリーの HTML バージョンは、すべての DB2 フィックスパックとともに更新されます。ご使用の製品のレベルに関係なく、資料を最新のバージョンにアップグレードできます。資料フィックスパックを使用して、HTML CD 資料インストール・プログラムを使用してインストールされた資料のみをアップグレードすることができます。資料フィックスパックは、HTML CD から直接コピーされた HTML を更新しません。

DB2 コントロール・センターなどの DB2 ツールをインストールするとき、同時にこれらのツールのヘルプがインストールされます。ツール・ヘルプへの更新は、ツールが使用可能であれば、新規 DB2 フィックスパックをインストールするときに適用されます。

情報を更新するときにはいつでも、ご使用のコンピューター上で English 資料だけでなく、その他の各国語資料も更新しています。すべての言語における情報が同じレベルに保たれています。

指定されたフィックスパックに関するすべての新規および更新済み資料をリストしているトピックは、 Information Center ナビゲーション・ツリーから使用可能です。

PDF バージョンの資料は、DB2 のすべてのポイント・リリースで更新され、DB2 サポート・サイト (<http://www.ibm.com/software/data/db2/udb/win0s2unix/support>) から使用可能です。この情報のサブセットは、選択されたフィックスパックと共にリフレッシュされます。リリース情報の PDF バージョンには、最終ポイント・リリース以降のすべての更新のレコードが含まれます。

JDBC ドライバーの機能強化

DB2[®] バージョン 8 の JDBC ドライバーでは、数多くの新機能と機能強化が行われています。この変更の中で最も大きい変更は、IBM[®] DB2 JDBC 汎用ドライバーとして認識される新規 JDBC ドライバー・アーキテクチャーです。汎用ドライバーは、分散およびローカル DB2 アクセスに対するアーキテクチャー中立の JDBC ドライバーです。汎用ドライバーは、特殊な JDBC ドライバー・タイプ・コネクティビティーまたはターゲット・プラットフォームから独立し、DB2 プラットフォームに対するシングル・ドライバー・インスタンスで、タイプ 4 およびタイプ 2 コネクティビティーの両方で使用可能です。さらに、プラットフォーム固有の部分は最も低位の層に要約されているため、さまざまな DB2 プラットフォームの中のドライバーの差は最小化されます。汎用 JDBC ドライバーの最初のリリースである 1.0 は、タイプ 4 コネクティビティーのみをサポートします。これは DB2 へのプラットフォーム相互アクセスのためのオープン分散プロトコル (分散リレーショナル・データベース体系、または DRDA として認識されています) に基づいています。

DB2 バージョン 8 では次のようになります。

- 新しいタイプ 4 JDBC ドライバーは、2 層の純粋な Java[™] JDBC ドライバーです。このドライバーによって、Java クライアントは DRDA[®] プロトコルを介して DB2 サーバーと直接通信することができます。このドライバーは、タイプ 3 ドライバーに代わるものとして設計されました。タイプ 3 ドライバーのサポートがやがて終了するのに備えて、タイプ 3 JDBC ドライバーを使用するアプレットをタイプ 4 に移行してください。
- DB2 はタイプ 4 JDBC ドライバーを使用する新規 SQLj プロファイル・カスタマイザーを提供します。
- JDBC ドライバーは、64 ビット・プラットフォームで使用できます。
- メモリー管理が改善されたため、ドライバーの安定度とパフォーマンスも改善されました。
- JDBC 1.2 ドライバーは使用しないでください

- 新しい SQLj 変換プログラムは、 java.sql.Blob および java.sql.Clob タイプの JDBC 2.0 と、ホスト変数式をサポートします。

関連概念:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『アプリケーションの移行』
- アプリケーション開発ガイド クライアント・アプリケーションのプログラミング の『Type 4 ドライバーによる Java のアプリケーションおよびアプレット・サポート』
- アプリケーション開発ガイド クライアント・アプリケーションのプログラミング の『DB2 JDBC Type 4 ドライバーによるJDBC 2.1 コア API 制約事項』
- アプリケーション開発ガイド クライアント・アプリケーションのプログラミング の『DB2 JDBC Type 4 ドライバーでサポートされる JDBC 2.1 オプション・パッケージ API』

インストール文書

概説およびインストール

コントロール・センターを使ったマルチサイト更新の使用可能化

コントロール・センターを使ってマルチサイト更新を行えます。

手順:

マルチサイト更新を使用可能にするには、以下のようになります。

1. コントロール・センターを立ち上げます。
2. [+] 記号をクリックし、ツリー・ビューを展開します。
3. 構成するインスタンスを右マウス・ボタンで選択します。ポップアップ・メニューが開きます。
4. 「マルチサイト更新 (Multisite Update)」 → 「構成 (Configure)」 の順にメニュー項目を選択します。「マルチサイト更新ウィザード」がオープンします。
5. 「下の名前付き TP モニターの使用 (Use the TP monitor named below)」および「トランザクション・プロセッサ (TP) モニターの指定 (Specify a Transaction Processor (TP) monitor)」を選択します。このフィールドには、使用可能にした TP モニターのデフォルトが表示されます。TP モニターを使用しないときは、「TP モニターを使用しない (Do Not Use a TP Monitor)」を選択します。「次へ (Next)」をクリックします。
6. TP モニターを使用している場合は、同期点管理プログラムの設定値を指定します。TP モニターを使用していない場合は、トランザクション・マネージャー・データベースを指定します。
7. 「完了 (Finish)」をクリックします。

関連概念:

- *DB2 Connect Personal Edition 概説およびインストール* の『マルチサイト更新』

関連タスク:

- *DB2 Connect Personal Edition 概説およびインストール* の『コントロール・センターを使ったマルチサイト更新のテスト』

構成アシスタント (CA) によるデータベース接続の構成

接続したいデータベースについての情報がある場合、手動ですべての構成情報を入力することができます。この方法は、コマンド行プロセッサを介してコマンドを入力するのと類似していますが、パラメーターがグラフィカルに提示されます。

前提条件:

CA を使用してデータベースとの接続を構成する前に、

- 有効な DB2 ユーザー ID であることを確認してください。
- DB2 サーバーまたは DB2 Connect サーバー製品がインストールされているシステムにデータベースを追加する場合は、SYSADM または SYSCTRL 権限を付与されたユーザー ID であることを確認してください。

手順:

CA を使用して手動でシステムにデータベースを追加するには、

1. 有効な DB2 ユーザー ID を使用してシステムにログオンします。
2. CA を開始します。CA は、「スタート」メニューから (Windows の場合)、または **db2ca** コマンドを使用することによって (Windows および UNIX システムの場合) 開始できます。
3. CA メニュー・バーの「**選択済み (Selected)**」で、「**ウィザードを使用してデータベースを追加 (Add Database Using Wizard wizard)**」を選択します。
4. 「**データベースへの接続を手動で構成する (Manually configure a connection to a database)**」ラジオ・ボタンを選択して、「**次へ (Next)**」をクリックします。
5. Lightweight Directory Access Protocol (LDAP) を使用している場合には、DB2 ディレクトリーを保持したい場所に対応するラジオ・ボタンを選択します。「**次へ (Next)**」をクリックします。
6. 「**プロトコル (Protocol)**」リストから、使用したいプロトコルに対応するラジオ・ボタンを選択します。

使用しているマシンに DB2 Connect がインストールされており、TCP/IP または APPC を選択する場合には、「**データベースはホストまたは OS/400 システムに物理的に常駐 (The database physically resides on a host or OS/400 system)**」を選択できます。このチェック・ボックスを選択すると、ホストまたは OS/400 に確立したい接続のタイプを選択するオプションが表示されます。

- DB2 Connect ゲートウェイ経由の接続を確立するには、「**ゲートウェイ経由でサーバーに接続 (Connect to the server via the gateway)**」ラジオ・ボタンを選択します。
- 直接接続を確立するには、「**サーバーに直接接続 (Connect directly to the server)**」ラジオ・ボタンを選択します。

「**次へ (Next)**」をクリックします。

7. 必要な通信プロトコル・パラメーターを入力し、「次へ (Next)」をクリックします。
8. 追加したいリモート・データベースのデータベース別名を「データベース名 (Database name)」フィールドに入力し、ローカル・データベース別名を「データベース別名 (Database alias)」フィールドに入力します。
ホストまたは OS/400 データベースを追加している場合、OS/390 または z/OS データベースのロケーション名、OS/400 データベースの RDB 名、または VSE か VM データベースの DBNAME を、「データベース名 (Database name)」フィールドに入力します。さらにオプションとして、このデータベースについて記述するコメントを「注釈 (Comment)」フィールドに入力します。
「次へ (Next)」をクリックします。
9. ODBC を使用する計画がある場合には、このデータベースを ODBC データ・ソースとして登録します。この操作を実行するには、ODBC がインストールされていなければなりません。
10. 「ノード・オプションの指定 (Specify the node options)」パネルで、オペレーティング・システムを選択し、接続したいデータベース・システムのリモート・インスタンス名を入力します。
11. 「完了 (Finish)」をクリックします。これで、このデータベースを使用できます。メニューから「終了 (Exit)」を選択して、CA を閉じます。

関連タスク:

- 154 ページの『ディスカバリーによるデータベース接続の構成』
- インストールおよび構成 補足 の『プロファイルによるデータベース接続の構成』
- 117 ページの『データベース接続のテスト』

クライアントでの TCP/IP ノードのカタログ

これは、CLP を使用したクライアント・サーバー接続の構成 のメインタスクの一部です。

TCP/IP ノードのカタログでは、DB2 クライアントのノード・ディレクトリーに、リモート・ノードについて記述するエントリーが追加されます。このエントリーでは、選択された別名 (*node_name*)、*hostname* (または *ip_address*)、およびクライアントがリモート・ホストにアクセスするときに使う *svcname* (または *port_number*) を指定します。

手順:

TCP/IP ノードをカタログするには、以下のステップを実行します。

1. システム管理 (SYSADM) 権限またはシステム・コントローラー (SYSCTRL) 権限のあるユーザーとしてシステムにログオンします。 *catalog_noauth* オプションが ON に設定されている場合には、これらの権限がなくてもシステムにログオンすることができます。

2. UNIX クライアントを使用する場合は、インスタンス環境をセットアップしてから、DB2 コマンド行プロセッサを起動します。開始スクリプトを以下のように実行します。

```
. INSTHOME/sqllib/db2profile (bash、 Bourne、 または Korn シェルの場合)
source INSTHOME/sqllib/db2cshrc (C シェルの場合)
```

ここで、*INSTHOME* は、インスタンスのホーム・ディレクトリーです。

3. **db2** プロンプトから以下のコマンドを入力することによって、ノードをカタログします。

```
catalog tcpip node node_name remote hostname|ip_address\
server service_name|port_number\
[remote_instance instance_name] [system system_name]
[ostype os_type]
terminate
```

ここで、

- *system* は、リモート・サーバーのシステム名です。
- *ostype* は、リモート・サーバー・システムのオペレーティング・システムです。

remote_instance、*system*、および *ostype* はオプションですが、DB2 ツールを使用するユーザーの場合は指定することをお勧めします。クライアントで使用される *service_name* は、サーバーのものと同じである必要はありません。しかし、そのマップ先ポート番号は同じでなければなりません。

たとえば、サービス名 *server1* を使用して、*db2node* というノードでリモート・ホスト *myserver* をカタログするには、**db2** プロンプトに以下のように入力します。

```
catalog tcpip node db2node remote myserver server server1
terminate
```

たとえば、ポート番号 *3700* を使用して、*db2node* というノードに IP アドレス *9.21.15.235* でリモート・サーバーをカタログするには、**db2** プロンプトに以下のように入力します。

```
catalog tcpip node db2node remote 9.21.15.235 server 3700
terminate
```

注: **terminate** コマンドは、ディレクトリー・キャッシュをリフレッシュするために必要です。

以下のステップは、クライアントでのデータベースのカタログです。

関連タスク:

- インストールおよび構成 補足 の『CLP によるクライアントでの TCP/IP の構成』

関連資料:

- コマンド・リファレンス の『CATALOG TCP/IP NODE コマンド』

コマンド行プロセッサを使用した、クライアントからサーバーへの接続の構成

このタスクでは、コマンド行プロセッサ (CLP) を使用して、DB2 クライアントからリモート・データベースへの接続を構成する方法を説明します。

構成アシスタントを使用して、クライアントからサーバーへの接続を構成することもできます。

前提条件:

クライアントからサーバーへの接続を構成する前に、

- クライアント・コンピューター上で、通信を構成する必要があります。ご使用のオペレーティング・システムによって、通信は 名前付きパイプ、NetBIOS または TCP/IP のいずれかとなります。
- DB2 サーバー上で、通信を構成する必要があります。ご使用のオペレーティング・システムによって、通信は 名前付きパイプ、NetBIOS または TCP/IP のいずれかとなります。
- サポートされる、クライアントからサーバーへの接続シナリオの 1 つを使用する必要があります。接続シナリオは、どのオペレーティング・システムがどの通信方式またはプロトコルを使用できるかを概説しています。NetBIOS を使用して Windows クライアントから、UNIX ベースのシステム上で稼動しているサーバーに接続することはできません。

手順:

以下のようにして、コマンド行プロセッサを使用して、クライアントからサーバーへの接続を構成します。

1. DB2 クライアントで、以下のいずれかのメソッドを使用して、データベース・ノードをカタログします。
 - DB2 クライアント上での TCP/IP ノードのカタログ
 - DB2 クライアント上での NetBIOS ノードのカタログ
 - DB2 クライアント上での名前付きパイプのカタログ
2. DB2 クライアント上でのデータベースのカタログ
3. クライアントからサーバーへの接続のテスト

関連タスク:

- 95 ページの『クライアントでの TCP/IP ノードのカタログ』
- インストールおよび構成 補足の『DB2 クライアントでの NetBIOS ノードのカタログ』
- インストールおよび構成 補足の『クライアントでの名前付きパイプ・ノードのカタログ』

- インストールおよび構成 補足 の『CLP によるデータベースのカatalog』
- インストールおよび構成 補足 の『CLP によるクライアント・サーバー接続のテスト』
- インストールおよび構成 補足 の『リモート DB2 インスタンスの通信プロトコルの構成』
- インストールおよび構成 補足 の『ローカル DB2 インスタンスの通信プロトコルの構成』
- インストールおよび構成 補足 の『DB2 インスタンスの APPC 通信の構成』
- インストールおよび構成 補足 の『DB2 インスタンスの NetBIOS 通信の構成』
- インストールおよび構成 補足 の『DB2 インスタンスの TCP/IP 通信の構成』
- インストールおよび構成 補足 の『DB2 インスタンスの名前付きパイプ通信の構成』
- *DB2 Universal Database* サーバー機能 概説およびインストール の『構成アシスタント (CA) を使用した、クライアントからサーバーへの接続の構成』

関連資料:

- インストールおよび構成 補足 の『クライアント・サーバー通信のシナリオ』

構成アシスタント (CA) のエクスポート機能を使用したクライアント・プロファイルの作成

クライアント・プロファイルは DB2 クライアントとサーバーの間の接続を作成するのに使用します。クライアント・プロファイルに含まれている情報は、エクスポート処理中に判別されます。クライアント・プロファイル内の判別された情報は、インポート処理を使用して、他のクライアントを構成するのに使用できます。

手順:

CA のエクスポート機能を使用してクライアント・プロファイルを作成するには、以下のステップを実行します。

1. CA を開始します。
2. 「**構成 (Configure)**」メニュー項目から「**プロファイルのエクスポート (Export Profile)**」を選択します。
3. 以下のオプションの内の 1 つを選択してください。
 - システムにカatalogされているすべてのデータベース、およびこのクライアントのすべての構成情報を含むプロファイルを作成したい場合、「**すべて (All)**」オプションを選択します。
 - このクライアントの構成情報なしで、システムにカatalogされているすべてのデータベースを含むプロファイルを作成したい場合、「**データベース接続 (Database connection)**」オプションを選択します。

- ・システム上にカタログされているデータベースのサブセット、またはこのクライアントの構成情報のサブセットを選択したい場合、「**カスタマイズ (Customize)**」オプションを選択し、次のステップに進んでください。

4. プロファイルの名前を入力します。
5. 「**データベース接続 (Database Connections)**」チェック・ボックスを選択します。
6. 「**使用可能なデータベース別名 (Available database aliases)**」ボックスからエクスポートするデータベースを選択し、ボタンをクリックして、「**選択されたデータベース別名 (Selected database aliases)**」ボックスに追加します。使用可能なデータベースをすべて「**選択されたデータベース別名 (Selected database aliases)**」ボックスに追加するには、「**>>**」ボタンをクリックします。
7. ターゲット・クライアント用に設定したいオプションに対応するチェック・ボックスを選択します。
8. 「**エクスポート (Export)**」をクリックします。

一度この作業を行ってから、インポート機能を使って他のクライアントを構成する必要があります。

関連概念:

- ・ *DB2 Universal Database* クライアント機能 概説およびインストール の『クライアント・プロファイル』

関連タスク:

- ・ インストールおよび構成 補足 の『プロファイルによるデータベース接続の構成』
- ・ インストールおよび構成 補足 の『プロファイルのエクスポートとインポート』
- ・ 116 ページの『構成アシスタント (CA) のインポート機能を使用したクライアント・プロファイルの構成』

管理ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索する

DB2 インフォメーション・センターによってユーザーは DB2 製品情報にすばやくアクセスできます。これは、DB2 管理ツールを使用できるすべてのオペレーティング・システムで利用できます。

ツールから DB2 インフォメーション・センターにアクセスすると、6 つのタイプの情報を参照できます。

タスク DB2 を使用して実行できる重要なタスク。

概念 DB2 の重要な概念。

参照情報

キーワード、コマンド、API などの DB2 参照情報。

トラブルシューティング

DB2 共通の問題に関して役立つエラー・メッセージおよび情報。

サンプル

DB2 によって提供されるサンプル・プログラムの HTML リストへのリンク。

チュートリアル

DB2 機能を学習するのを支援するように設計されたインストラクション。

前提条件:

DB2 インフォメーション・センターのリンクの中には、インターネット上の Web サイトを指しているものがあります。これらのリンクのコンテンツを表示するには、その前にインターネットに接続している必要があります。

手順:

ツールから DB2 インフォメーション・センターにアクセスして、製品情報を検索するには以下のようにします。

1. 以下に示す方法のうちの 1 つを使って、DB2 インフォメーション・センターを開始します。
 - グラフィカル管理ツールから、ツールバーの「**インフォメーション・センター (Information Center)**」アイコンをクリックします。「**ヘルプ (Help)**」メニューからも選択できます。
 - コマンド行で、**db2ic** と入力します。
2. 検索する情報に関連した情報タイプのタグをクリックします。
3. ツリーをナビゲートして、内容を知りたいトピックをクリックします。インフォメーション・センターは Web ブラウザーを起動して、情報を表示します。

関連概念:

- *Data Links Manager 概説*およびインストールの『*アクセス支援*』
- 88 ページの『*ブラウザーからアクセスする DB2 インフォメーション・センター*』

関連タスク:

- *Data Links Manager 概説*およびインストールの『*ブラウザーから DB2 インフォメーション・センターにアクセスしてトピックを検索する*』
- *Data Links Manager 概説*およびインストールの『*DB2 文書の検索*』

DB2 Personal Edition

DB2[®] Personal Edition は、シングル・ユーザー・バージョンの DB2 です。これは、ローカル・データベースを作成および管理するのに使用することができますが、DB2 Workgroup Server Edition または Enterprise Server Edition データベース・サーバーに対するクライアントとして使用することもできます。また DB2 Personal Edition は、サテライトとして機能することもでき、DB2 Enterprise Server Edition データベース・サー

パーからリモート管理されます。サテライト環境での DB2 Personal Edition の使用に関する詳細は、サテライト管理のドキュメンテーションを参照してください。

DB2 Personal Edition 上のインスタンスおよびデータベースの移行 (Linux)

このトピックでは、Linux 上の旧バージョンの DB2 Personal Edition からの移行が完全にインストールした後にとる必要のあるステップを説明します。移行に関するさらに詳しい解説と、**db2imigr** および **migrate** コマンドに関する総合情報の詳細は、このトピックの巻末の参考資料の項を参照してください。

前提条件:

- インスタンスを移行するには、root 権限が必要です。
- データベースを移行するには、SYSADM 権限が必要です。

手順:

DB2 Personal Edition のインストールが完了したら、インスタンスとデータベースを移行して移行処理を完了する必要があります。インスタンスとデータベースを移行するには、以下のステップを実行します。

1. **db2imigr** コマンドを使用して、以下のようにインスタンスを移行します。
 - a. root 権限があるユーザーとしてログインします。
 - b. **db2imigr** コマンドを使用して、以下のようにインスタンスを移行します。

```
DB2DIR/instance/db2imigr [-u fencedID] InstName
```

ここで、

DB2DIR

Linux オペレーティング・システム上の /opt/IBM/db2/V8.1 です。

-u fencedID

分離ユーザー定義関数 (UDF) およびストアド・プロシージャを実行するユーザーです。このオプションを使用するのは、クライアント・インスタンスをサーバー・インスタンスに移行する場合のみです。サーバーからサーバーや、クライアントからクライアントへの移行ではこのオプションは必要ありません。

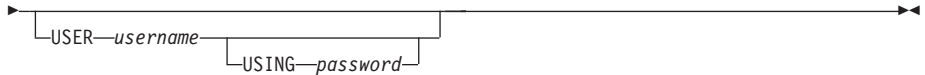
InstName

インスタンス所有者のログイン名です。

2. SYSADM 権限を持つユーザー・アカウントでログインし、**DB2 MIGRATE DATABASE** コマンドを使用して、データベースを移行します。

DB2 MIGRATE DATABASE コマンド

```
▶—MIGRATE—DATABASE—database-alias—▶  
          └──DB──┘
```



ここで、

DATABASE *database-alias*

現在インストールされているバージョンのデータベース・マネージャーに移行するデータベースの別名を指定します。

USER *username*

データベースの移行に使用するユーザー名を識別します。

USING *password*

ユーザー名を認証するのに使用するパスワード。パスワードを省略しても、ユーザー名が指定されている場合には、パスワードを入力するようプロンプトが出されます。

3. オプション: 統計を更新します。データベースの移行が完了したとき、照会のパフォーマンスを最適化するために使用されていた旧統計は、カタログ内に保存されています。ただし、DB2 バージョン 8 には、改良された統計や、DB2 バージョン 6 または DB2 バージョン 7 にはなかった統計もあります。これらの統計の利点を活用するには、表に対して **runstats** コマンドを実行すると良いでしょう。SQL 照会のパフォーマンスに対して重大な影響のある表の場合は特にそうです。
4. オプション: パッケージを再バインドします。データベースを移行すると、既存のパッケージはすべて無効になります。移行処理後に各パッケージは、DB2 バージョン 8 データベース・マネージャーによって最初に使用されるときに再作成されます。あるいは、**db2rbind** コマンドを実行し、データベースに格納されているすべてのパッケージを再作成することもできます。
5. オプション: PUBLIC からの SQL データ・アクセスを含む、外部ストアド・プロシージャ上で、EXECUTE 特権を取り消します。データベースの移行時に、すべての関数、メソッド、および外部ストアド・プロシージャの EXECUTE 特権が PUBLIC に対して認可されます。これは、SQL データ・アクセスの入った外部ストアド・プロシージャの機密漏れを引き起こします。この外部ストアド・プロシージャを使用すれば、ユーザーは本来なら特権の対象にはなりえない SQL オブジェクトにアクセスできるようになるからです。特権を取り消すには、**db2undgp - r** コマンドを入力します。

注: 移行中に、データベース構成パラメーター *maxappls* は自動的に設定されます。これを別の値に設定したい場合は、手動で更新する必要があります。

関連概念:

- 管理ガイド: インプリメンテーション の『システム管理権限 (SYSADM)』

関連資料:

- コマンド・リファレンス の『MIGRATE DATABASE コマンド』

- コマンド・リファレンス の『db2imigr - インスタンスの移行コマンド』

DB2 Personal Edition 上のデータベースの移行 (Windows)

このトピックでは、Windows 上の旧バージョンの DB2 Personal Edition からの移行の実行のためのインストール後にとる必要のあるステップを説明します。移行に関するさらに詳しい解説と、**移行**のためのコマンドに関する総合情報の詳細は、このトピックの巻末の参考資料の項を参照してください。

前提条件:

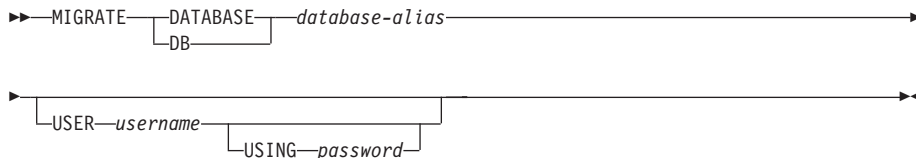
- データベースを移行するには、SYSADM 権限が必要です。

手順:

DB2 Personal Edition のインストールが完了したら、データベースを移行して、移行処理を完了する必要があります。データベースを移行するには、以下のステップを実行します。

1. SYSADM 権限を持つユーザー・アカウントでログインし、 **db2 migrate database** コマンドを使用して、データベースを移行します。

DB2 MIGRATE DATABASE コマンド



ここで、

DATABASE *database-alias*

現在インストールされているバージョンのデータベース・マネージャーに移行するデータベースの別名を指定します。

USER *username*

データベースの移行に使用するユーザー名を識別します。

USING *password*

ユーザー名を認証するのに使用するパスワード。パスワードを省略しても、ユーザー名が指定されている場合には、パスワードを入力するようプロンプトが出されます。

2. オプション: 統計を更新します。データベースの移行が完了したとき、照会のパフォーマンスを最適化するのに使用されていた旧統計は、カタログ内に保存されています。ただし、DB2 バージョン 8 には、改良された統計や、DB2 バージョン 6 または DB2 バージョン 7 にはなかった統計もあります。これらの統計の利点を活用するには、表に対して **runstats** コマンドを実行すると良いでしょう。SQL 照会のパフォーマンスに対して重大な影響のある表の場合は特にそうです。

3. オプション: パッケージを再バインドします。データベースを移行すると、既存のパッケージはすべて無効になります。移行処理後に各パッケージは、DB2 バージョン 8 データベース・マネージャーによって最初に使用されるときに再作成されます。あるいは、**db2rbind** コマンドを実行し、データベースに格納されているすべてのパッケージを再作成することもできます。
4. オプション: PUBLIC からの SQL データ・アクセスを含む、外部ストアド・プロシージャ上で、EXECUTE 特権を取り消します。データベースの移行時に、すべての関数、メソッド、および外部ストアド・プロシージャの EXECUTE 特権が PUBLIC に対して認可されます。これは、SQL データ・アクセスの入った外部ストアド・プロシージャの機密漏れを引き起こします。この外部ストアド・プロシージャを使用すれば、ユーザーは本来なら特権の対象にはなりえない SQL オブジェクトにアクセスできるようになるからです。特権を取り消すには、**db2undgp - r** コマンドを入力します。

注: 移行中に、データベース構成パラメーター *maxappls* は自動的に設定されます。これを別の値に設定したい場合は、手動で更新する必要があります。

関連概念:

- 管理ガイド: インプリメンテーション の『システム管理権限 (SYSADM)』

関連資料:

- コマンド・リファレンス の『MIGRATE DATABASE コマンド』

DB2 セットアップ・ウィザードを使用した DB2 オンライン・ドキュメンテーションのインストール (UNIX)

このタスクでは、UNIX 上で DB2 セットアップ・ウィザードを使用して、DB2 オンライン・ドキュメンテーションをインストールする方法を説明します。DB2 オンライン・ドキュメンテーションは、他の DB2 製品とは別個に、それ自体の CD-ROM からインストールします。

前提条件:

DB2 セットアップ・ウィザードを開始する前に、以下の事柄を行います。

- インストールを実行するために、root 権限が必要です。
- DB2 HTML ドキュメンテーション CD-ROM が、ご使用のシステムにマウントされていないなければなりません。
- DB2 セットアップ・ウィザードは、グラフィック・インストーラーです。ご使用のマシンでグラフィック・インストーラーが稼働するためには、グラフィカル・ユーザー・インターフェースが使用可能な X Window System ソフトウェアが必要です。
- Java Runtime Environment (JRE) がすでにインストール済みでなければなりません。

手順:

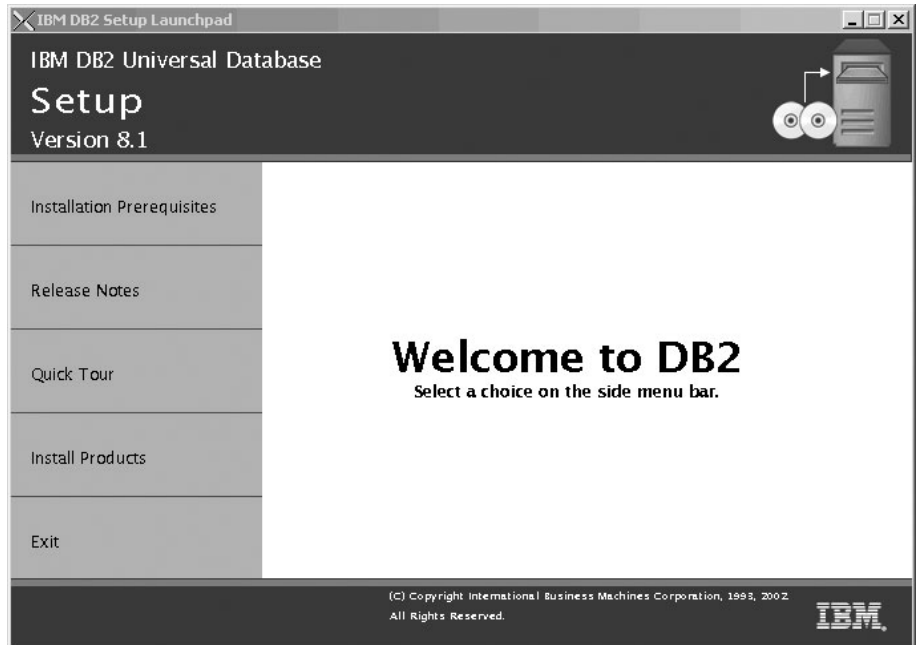
以下のようにして、DB2 セットアップ・ウィザードを使用して DB2 オンライン情報をインストールします。

1. root 権限を持つユーザーとしてシステムにログオンします。
2. 以下のコマンドを入力することによって、CD-ROM がマウントされているディレクトリに移動します。

```
cd /cdrom
```

ここで、`/cdrom` は、CD-ROM のマウント・ポイントを表しています。

3. `./db2setup` コマンドを入力し、DB2 セットアップ・ウィザードを開始します。数分後に、IBM DB2 セットアップのランチパッドが開きます。



このウィンドウで、インストールの前提条件およびリリース情報を表示できますし、クイック・ツアーを実行して DB2 Universal Database バージョン 8 の機能を知ることができます。また、そのままインストールを実行できます。後で追加されたインストール前提条件およびリリース情報も参照することができます。

インストールを開始したなら、DB2 セットアップ・ウィザードのインストール・パネルにしたがって、選択を行ってください。残りのステップについて説明しているインストール操作のヘルプを利用できます。インストール操作のヘルプを呼び出すには、「ヘルプ」をクリックするか「F1」キーを押します。「キャンセル (Cancel)」をクリックすれば、いつでもインストールを終了できます。

関連概念:

- *DB2 Universal Database* サーバー機能 概説およびインストール の『DB2 サーバーのインストールの概要 (UNIX)』
- *DB2 Universal Database* サーバー機能 概説およびインストール の『パーティション DB2 サーバーのインストールの概要 (UNIX)』
- *DB2 Universal Database Personal Edition* 概説およびインストール の『DB2 Personal Edition のインストールの概要 (Linux)』

移行に関する推奨事項

データベース移行を計画する際には、以下の推奨事項を考慮してください。

移行の前に `sqldddir.bak` ファイルを削除する

ファイル `sqldddir.bak` は、DB2[®] システム・データベース・ディレクトリーをバックアップするために、インスタンスの移行中に作成されます。このファイルは、この情報を保存し、失敗したときに移行を再開するために使用されません。ファイル `sqldddir.bak` は以下のロケーションにあります。

Windows の場合:

- `<instance path>\NODE<XXXX>\sqlbdir\sqldddir.bak`
`<instance path>` はインスタンスが作成されたディレクトリーで、`<XXXX>` は 0000 から 9999 までの値です。または、
- `<SQLLIB path>\<instance name>\sqlbdir\sqldddir.bak`
`<SQLLIB path>` は DB2 がインストールされているディレクトリーで、`<instance name>` はインスタンスの名前です。

UNIX の場合:

- `<instance path>/NODE<XXXX>/sqlbdir/sqldddir.bak`
`<instance path>` はインスタンスが作成されたディレクトリーで、`<XXXX>` は 0000 から 9999 までの値です。または、
- `<SQLLIB path>/<instance name>/sqlbdir/sqldddir.bak`
`<SQLLIB path>` は DB2 がインストールされているディレクトリーで、`<instance name>` はインスタンスの名前です。

以前に正常に移行されたインスタンスを移行している場合、このファイルが存在するか確認する必要があります。インスタンスの移行を開始する前に、このファイルを削除する必要があります。このファイルを削除しないと、前の移行以降のすべてのカタログ項目が失われます。

警告: 失敗した移行を再開しようとしている場合は、このファイルを削除しないでください。インスタンスの移行の開始前にのみ、削除してください、

DB2 がレプリケーションを使用する場合は、移行の前にログ・ファイルをバックアップする レプリケーションを使用する場合には、移行の前に、すべての DB2 ログ・ファイルをアーカイブする必要があります。

ハードウェアおよびオペレーティング・システムのアップグレードは、DB2 の移行とは別個に行う

ハードウェアおよびオペレーティング・システムのアップグレードを DB2 の移行とは別個に実行することで、移行時に問題が生じたときの問題判別をより簡単に行なうことができます。DB2 の移行に先立ってソフトウェアまたはハードウェアをアップグレードする場合は、DB2 の移行を試行する前に、ご使用のシステムが正常に作動することを確認してください。

下位レベル・サーバーのサポート

DB2 バージョン 7 での環境を DB2 バージョン 8 に移すとき、すべての DB2 サーバーをバージョン 8 に移行する前に DB2 クライアントをバージョン 8 に移行する場合は、制約と制限がいくつかあります。これらの制約事項と制限事項は DB2 Connect には関連しません。また、zSeries、OS/390、および iSeries™ データベース・サーバーにも関連しません。既知の制約事項と制限事項を回避するために、すべての DB2 サーバーをバージョン 8 に移行してから、DB2 クライアントをバージョン 8 に移行してください。

ベンチマーク DB2 パフォーマンス

DB2 を移行する前に、いくつかのテスト照会を実行し、照会が実行されたときの正確な環境条件を記録してください。また、それぞれのテスト照会ごとの **db2expln** コマンド出力の記録を保存しておき、移行前後の結果を比較してください。このことは、性能低下を識別し修正するのに役立ちます。

移行をバックアウトするプランの考案

移行を逆行させるためのユーティリティはありません。移行をバックアウトする必要がある場合には、システムから DB2 バージョン 8 コードを除去し、以前のバージョンの DB2 を再インストールして、バックレベルのインスタンスを再作成し、データベース・バックアップをリストアする必要があります。移行をバックアウトする必要がある場合には、現行のデータベース・バックアップおよびデータベースとデータベース構成設定値の明細レコードが不可欠です。

テスト環境での DB2 バージョン 8 の移行

実稼働環境で移行を行う前に、テスト環境で DB2 バージョン 8 に移行してください。これを行うことにより、実稼働環境を移行処理にコミットする前に、移行の障害を検出し、アプリケーションおよびツールが確実に適正に作動するようにしておくことができます。

DB2 DataPropagator™ レプリケーションでの移行

DB2 DataPropagator 用のキャプチャーまたはアプライ・プログラムを実行している DataJoiner、DB2 for UNIX、または DB2 for Windows® のインスタンスを移行する前に、DB2 DataPropagator バージョン 8 用の移行に関するドキュ

メンテーションをお読みください。DB2 または DataJoiner[®] インスタンスを移行する前に、レプリケーション環境の移行の準備を行う必要があります。DB2 または DataJoiner インスタンスの移行後に直ちに実行しなければならないステップもあります。DB2 DataPropagator バージョン 8 用の移行に関するドキュメンテーションは、<http://www.ibm.com/software/data/dpropr/library.html> の Web サイトにあります。

関連概念:

- 管理ガイド: パフォーマンス の『ベンチマーク・テスト』
- 管理ガイド: パフォーマンス の『Explain ツール』

関連タスク:

- *DB2 Universal Database* サーバー機能 概説およびインストール の『DB2 の移行 (Windows)』
- *DB2 Universal Database* サーバー機能 概説およびインストール の『DB2 の移行 (UNIX)』

関連資料:

- コマンド・リファレンス の『db2expln - DB2 SQL Explain ツール・コマンド』
- 管理ガイド: プランニング の『DB2 Universal Database で計画されている非互換性』
- 282 ページの『バージョン 8 と以前のリリースとの非互換性』
- 管理ガイド: プランニング の『バージョン 7 のリリース間の非互換性』

DB2 クライアント

DB2[®] クライアントには以下の 3 つのタイプがあります。

- Run-Time Client
- Administration Client
- Application Development Client

バージョン 8 クライアントは、バージョン 8 サーバーをサポートします。バージョン 8 クライアントは、バージョン 7 サーバーが DRDA-AS を実行している場合にのみ、バージョン 7 サーバーに接続できます。バージョン 7 クライアントは、機能限定でバージョン 8 に接続できます。

DB2 クライアント上にデータベースを作成することはできません。DB2 サーバー上にあるデータベースにアクセスする必要があります。

関連概念:

- *DB2 Universal Database* クライアント機能 概説およびインストール の『DB2 Run-Time Client』

- *DB2 Universal Database* クライアント機能 概説およびインストール の『DB2 Administration Client』
- *DB2 Universal Database* クライアント機能 概説およびインストール の『DB2 Application Development Client』
- インストールおよび構成 補足 の『応答ファイル』

関連タスク:

- *DB2 Universal Database* クライアント機能 概説およびインストール の『Windows オペレーティング・システムでの DB2 クライアントのインストール』
- *DB2 Universal Database* クライアント機能 概説およびインストール の『UNIX での DB2 クライアントのインストール』

doc_install コマンドを使用した DB2 HTML ドキュメンテーションのインストール (UNIX)

「DB2 セットアップ」ウィザードの代わりに、**doc_install** コマンドを使用して、DB2 HTML ドキュメンテーションを UNIX にインストールすることができます。

前提条件:

DB2 HTML ドキュメンテーションのインストールを行う前に、次を確認してください。

- インストールを実行するために、root 権限が必要です。
- DB2 HTML ドキュメンテーション CD-ROM が、ご使用のシステムにマウントされていなければなりません。

手順:

DB2 HTML ドキュメンテーションを **doc_install** コマンドを使用してインストールするには、以下のようにします。

1. コマンド行から、**doc_install** コマンドを以下の構文で実行します。

```
doc_install -l <language> -t <topic> [-p <path>] [-d]
```

<language> はサポートされている言語で、<topic> は以下のうちの 1 です。

core	コア DB2 情報
admin	管理情報
ad	アプリケーション開発情報
wareh	ビジネス・インテリジェンス情報
conn	DB2 Connect 情報
start	入門情報
tutr	チュートリアル情報

opt オプションのコンポーネント情報

たとえば、以下のように指定します。

```
doc_install -l en_US -t start -p /var/www/html/db2
```

DB2 HTML ドキュメンテーションの入門情報を /var/www/html/db2 ディレクトリにインストールします。

doc_install と入力することで、コマンドの使用法を表示できます。表示される言語のリストは、CD-ROM で使用可能な言語です。

関連タスク:

- *Data Links Manager 概説およびインストール* の『ブラウザから DB2 インフォメーション・センターにアクセスしてトピックを検索する』
- *Data Links Manager 概説およびインストール* の『DB2 HTML ドキュメンテーション CD から技術資料を直接参照する』
- *Data Links Manager 概説およびインストール* の『DB2 HTML ドキュメンテーション CD から Web サーバーヘッファイルをコピーする』

関連資料:

- *Data Links Manager 概説およびインストール* の『DB2 Universal Database の技術情報の概要』

db2setup の各国語表示

db2setup コマンドは、既存の言語設定を判別するために、オペレーティング・システムを照会します。ご使用のオペレーティング・システムの言語設定が **db2setup** でサポートされている場合は、その言語が使用されます。

ご使用のシステムで同じコード・ページを使用しているが、DB2 インターフェースでサポートされているものとは異なるロケール名を使用している場合でも、**LANG** 環境変数を、以下のコマンドを入力して適当な値に設定すれば、翻訳された **db2setup** を表示することができます。

export LANG= <locale>ここで *locale* は DB2 インターフェースでサポートされているロケールです。

たとえば、ご使用の Linux マシンのデフォルト・ロケールが ja_JP.ujis で、ja_JP.ujis のコード・ページが 954 の場合、以下のコマンド (Bourne シェル、またはその亜種を使用した場合) を実行して、**LANG** 環境変数を ja_JP.sjis に設定することができます。

```
export LANG=ja_JP.sjis
```

これで、**db2setup** は日本語 (コード・ページ 954) で表示されます。

注: **db2setup** は、インストールしている DB2 製品によってサポートされている言語のサブセットに翻訳される場合があります。

関連資料:

- 111 ページの『サポートされる DB2 インターフェース言語、ロケール、およびコード・ページ』
- コマンド・リファレンス の『db2setup - DB2 のインストール・コマンド』

サポートされる DB2 インターフェース言語、ロケール、およびコード・ページ

以下の表には、DB2 インターフェースによってサポートされる言語 (オペレーティング・システムごと)、言語ロケール、および各ロケールのコード・ページがリストされています。DB2 インターフェースには、メッセージ、ヘルプ、およびグラフィカル・ツール・インターフェースが含まれます。

DB2 インターフェースの DB2 言語サポートは、サーバー・グループ言語とクライアント・グループ言語に分類できます。サーバー・グループ言語は、メッセージ、ヘルプおよび DB2 グラフィカル・インターフェース要素のほとんどを翻訳します。クライアント・グループ言語は、メッセージのほとんどと特定のヘルプ・ドキュメンテーションを含む、DB2 Run-time Client コンポーネントを翻訳します。

サーバー・グループ言語には、フランス語、ドイツ語、スペイン語、イタリア語、ブラジル・ポルトガル語、デンマーク語、フィンランド語、ノルウェー語、スウェーデン語、日本語、韓国語、中国語 (簡体字)、中国語 (繁体字)、ロシア語、ポーランド語が含まれます。

クライアント・グループ言語には、チェコ語、ハンガリー語、ブルガリア語、スロベニア語、トルコ語、オランダ語、クロアチア語、ルーマニア語、スロバキア語が含まれます。

DB2 によってサポートされる言語と、DB2 インターフェースによってサポートされる言語とを混同しないでください。DB2 によってサポートされる言語とは、データの言語のことで、DB2 インターフェースによってサポートされる言語のスーパーセットです。

言語 DB2 インターフェースによってサポートされる言語 (メッセージ、ヘルプ、およびグラフィカル・ツール・インターフェース)

ロケール

ロケールは、UNIX システムの LANG 環境変数で、DB2 インターフェースの言語を設定するために使用されます。Windows システムで DB2 インター

フェース言語を設定する場合には、ロケールは必要ありません。Windows システムでは、DB2 インターフェース言語は、システムのデフォルト言語設定によって決まります。

コード・ページ

各ロケールに関連したコード・ページです。

表 29. AIX および HP-UX のロケールおよびコード・ページ

オペレーティング・システム	AIX		HPUX	
	ロケール	コード・ページ	ロケール	コード・ページ
フランス語	fr_FR.iso88591 fr_FR.ibm850 fr_FR.utf8	819 850 1208	fr_FR.iso88591 fr_FR.roman8 fr_FR.utf8	819 1051 1208
ドイツ語	de_DE.iso88591 de_DE.ibm850 de_DE.utf8	819 850 1208	de_DE.iso88591 de_DE.roman8 de_DE.utf8	819 1051 1208
イタリア語	it_IT.iso88591 it_IT.ibm850 it_IT.utf8	819 850 1208	it_IT.iso88591 it_IT.roman8 it_IT.utf8	819 1051 1208
スペイン語	es_ES.iso88591 es_ES.ibm850 es_ES.utf8	819 850 1208	es_ES.iso88591 es_ES.roman8 es_ES.utf8	819 1051 1208
ブラジル・ポルトガル語	pt_BR.iso88591 pt_BR.ibm850 pt_BR.utf8	819 850 1208	pt_BR.iso88591 pt_BR.utf8	819 1208
日本語	ja_JP.eucJP ja_JP.sjis ja_JP.utf8	954 932 1208	ja_JP.eucJP ja_JP.sjis ja_JP.utf8	954 932 1208
韓国語	ko_KR.eucKR ko_KR.utf8	970 1208	ko_KR.eucKR ko_KR.utf8	970 1208
中国語 (簡体字)	zh_CN.eucCN zh_CN.gbk zh_CN.utf8	1383 1386 1208	zh_CN.eucCN zh_CN.utf8	1383 1208
中国語 (繁体字)	zh_TW.eucTW zh_TW.big5 zh_TW.utf8	964 950 1208	zh_TW.eucTW zh_TW.big5 zh_TW.utf8	964 950 1208

表 29. AIX および HP-UX のロケールおよびコード・ページ (続き)

デンマーク語	da_DK.iso88591	819	da_DK.iso88591	819
	da_DK.ibm850	850	da_DK.ibm850	1051
	da_DK.utf8	1208	da_DK.utf8	1208
フィンランド語	fi_FL.iso88591	819	fi_FL.iso88591	819
	fi_FL.ibm850	850	fi_FL.roman8	1051
	fi_FL.utf8	1208	fi_FL.utf8	1208
ノルウェー語	no_NO.iso88591	819	no_NO.iso88591	819
	no_NO.ibm850	850	no_NO.roman8	1051
	no_NO.utf8	1208	no_NO.utf8	1208
スウェーデン語	sv_SE.iso88591	819	sv_SE.iso88591	819
	sv_SE.ibm850	850	sv_SE.roman8	1051
	sv_SE.utf8		sv_SE.utf8	
チェコ語	cs_CZ.iso88592	912	cs_CZ.iso88592	912
	cs_CZ.utf8	1208	cs_CZ.utf8	1208
ハンガリー語	hu_HU.iso88592	912	hu_HU.iso88592	912
	hu_HU.utf8	1208	hu_HU.utf8	1208
ポーランド語	pl_PL.iso88592	912	pl_PL.iso88592	912
	pl_PL.utf8	1208	pl_PL.utf8	1208
オランダ語	nl_NL.iso88591	912		
	nl_NL.ibm850	850		
	nl_NL.utf8	1208		
トルコ語	tr_TR.iso88599	920		
	tr_TR.utf8	1208		
ロシア語	ru_RU.iso88595	915	ru_RU.iso88595	915
	ru_RU.utf8	1208	ru_RU.utf8	1208
ブルガリア語	bg_BG.iso88595	915	bg_BG.iso88595	915
	bg_BG.utf8	1208	bg_BG.utf8	1208
スロベニア語	sl_SI.iso88592	912	sl_SI.iso88592	912
	sl_SI.utf8	1208	sl_SI.utf8	1208
クロアチア語	hr_HR.iso88592	912	hr_HR.iso88592	912
	hr_HR.utf8	1208	hr_HR.utf8	1208
ルーマニア語	ro_RO.iso88592	912	ro_RO.iso88592	912
	ro_RO.utf8	1208	ro_RO.utf8	1208
スロバキア語	sk_SK.iso88592	912	sk_SK.iso88592	912
	sk_SK.utf8	1208	sk_SK.utf8	1208

表 30. Solaris オペレーティング環境および Linux のロケールおよびコード・ページ

オペレーティング・システム	Solaris		Linux および Linux/390	
	ロケール	コード・ページ	ロケール	コード・ページ
フランス語	fr_FR.iso88591 fr_FR.utf8	819 1208	fr_FR.iso88591 fr_FR.utf8	819 1208
ドイツ語	de_DE.iso88591 de_DE.utf8	819 1208	de_DE.iso88591 de_DE.utf8	819 1208
イタリア語	it_IT.iso88591 it_IT.utf8	819 1208	it_IT.iso88591 it_IT.utf8	819 1208
スペイン語	es_ES.iso88591 es_ES.utf8	819 1208	es_ES.iso88591 es_ES.utf8	819 1208
ブラジル・ポルトガル語	pt_BR.iso88591 pt_BR.utf8	819 1208	pt_BR.iso88591 pt_BR.utf8	819 1208
日本語	ja_JP.eucJP ja_JP.sjis ja_JP.utf8	954 932 1208	ja_JP.eucJP ja_JP.sjis ja_JP.utf8	954 932 1208
韓国語	ko_KR.eucKR ko_KR.utf8	970 1208	ko_KR.eucKR ko_KR.utf8	970 1208
中国語 (簡体字)	zh_CN.eucCN zh_CN.utf8	1383 1208	zh_CN.eucCN zh_CN.gbk zh_CN.utf8	1383 1386 1208
中国語 (繁体字)	zh_TW.eucTW zh_TW.big5 zh_TW.utf8	964 950 1208	zh_TW.eucTW zh_TW.big5 zh_TW.utf8	964 950 1208
デンマーク語	da_DK.iso88591 da_DK.utf8	819 1208		
フィンランド語	fi_FI.iso88591 fi_FI.utf8	819 1208		
ノルウェー語	no_NO.iso88591 no_NO.utf8	819 1208		
スウェーデン語	sv_SE.iso88591 sv_SE.utf8	819 1208		
チェコ語	cs_CZ.iso88592 cs_CZ.utf8	912 1208	cs_CZ.iso88592 cs_CZ.utf8	912 1208
ハンガリー語	hu_HU.iso88592 hu_HU.utf8	912 1208	hu_HU.iso88592 hu_HU.utf8	912 1208

表 30. Solaris オペレーティング環境および Linux のロケールおよびコード・ページ
(続き)

ポーランド語	pl_PL.iso88592 pl_PL.utf8	912 1208	pl_PL.iso88592 pl_PL.utf8	912 1208
オランダ語				
トルコ語				
ロシア語	ru_RU.iso88595 ru_RU.utf8	915 1208	ru_RU.iso88595 ru_RU.utf8	915 1208
ブルガリア語	bg_BG.iso88595 bg_BG.utf8	915 1208	bg_BG.iso88595 bg_BG.utf8	915 1208
スロベニア語	sl_SI.iso88592 sl_SI.utf8	912 1208	sl_SI.iso88592 sl_SI.utf8	912 1208
クロアチア語	hr_HR.iso88592 hr_HR.utf8	912 1208	hr_HR.iso88592 hr_HR.utf8	912 1208
ルーマニア語	ro_RO.iso88592 ro_RO.utf8	912 1208	ro_RO.iso88592 ro_RO.utf8	912 1208
スロバキア語	sk_SK.iso88592 sk_SK.utf8	912 1208	sk_SK.iso88592 sk_SK.utf8	912 1208

注:

各プラットフォームで使用される共通ロケール名の場合、DB2 は上に示した対応するディレクトリーにマップします。たとえば、ご使用の AIX マシンのデフォルト・ロケールが ja_JP で、ja_JP のコード・ページが 954 の場合、DB2 は ja_JP.eucJP ディレクトリーからストリングを検索します。同様に、ご使用の Linux マシンのデフォルト・ロケールが Japanese で、Japanese が 954 の場合、DB2 はストリングを ja_JP.eucJP ディレクトリーから検索します。

関連タスク:

- DB2 Universal Database サーバー機能 概説およびインストール の『DB2 移行前の診断エラー・レベルの変更』
- DB2 Universal Database サーバー機能 概説およびインストール の『DB2 インターフェイス言語の変更 (Windows)』
- DB2 Universal Database サーバー機能 概説およびインストール の『DB2 インターフェイス言語の変更 (UNIX)』

関連資料:

- 管理ガイド: プランニング の『各国語バージョン』
- 242 ページの『サポートされているテリトリー・コードおよびコード・ページ』

構成アシスタント (CA) のインポート機能を使用したクライアント・プロファイルの構成

この作業は、CA のエクスポート機能を使用した後にクライアント・プロファイルを構成するために行います。クライアント・プロファイルは DB2 クライアントとサーバーの間の接続を作成するのに使用します。クライアント・プロファイルに含まれている情報は、エクスポート処理中に判別されます。クライアント・プロファイル内の判別された情報は、インポート処理を使用して、他のクライアントを構成するのに使用できません。

手順:

CA のインポート機能を使用してクライアント・プロファイルを構成するには、以下のタスクを実行します。

1. CA を開始します。
2. 「構成 (Configure)」メニューで「プロファイルのインポート (Import Profile)」を選択します。
3. クライアント・プロファイル中の情報すべてまたはサブセットのインポートを選択できます。以下のインポート・オプションから 1 つを選択してください。
 - クライアント・プロファイルにあるすべてをインポートするには、「すべて (All)」オプションを選択します。このオプションを選択した場合、この時点で DB2 製品の使用を開始する準備ができました。
 - クライアント・プロファイルに定義されている特定のデータベースまたは設定をインポートするには、「カスタマイズ (Customize)」オプションを選択してください。
 - a. クライアント・プロファイルを選択して、「ロード (Load)」をクリックします。
 - b. カスタマイズしたいオプションに対応するチェック・ボックスを選択します。
 - c. 表示されているデータベースのリストから、インポートしたいデータベースを選択して、「インポート (Import)」をクリックします。

関連概念:

- *DB2 Universal Database* クライアント機能 概説およびインストール の『クライアント・プロファイル』

関連タスク:

- インストールおよび構成 補足 の『プロファイルによるデータベース接続の構成』
- インストールおよび構成 補足 の『プロファイルのエクスポートとインポート』
- 98 ページの『構成アシスタント (CA) のエクスポート機能を使用したクライアント・プロファイルの作成』

データベース接続のテスト

データベースを構成した後、データベース接続をテストする必要があります。

手順:

データベース接続をテストするには、

1. 「**構成アシスタント (Configuration Assistant)**」を開始します。
2. 詳細ビューでデータベースを強調表示してから、「**テスト接続 (Test Connection)**」を「**選択済み (Selected)**」メニューから選択します。「テスト接続 (Test Connection)」ウィンドウが表示されます。
3. テストする接続の種類を選択します (デフォルトは **CLI**)。リモート・データベースの有効なユーザー ID およびパスワードを入力し、「**テスト接続 (Test Connection)**」をクリックします。接続が正常に確立された場合、接続が確立されたことを確認するメッセージが「**結果 (Results)**」ページに表示されます。

接続テストに失敗した場合には、ヘルプ・メッセージを受け取ります。誤って指定した設定を変更するには、詳細ビューでデータベースを選択してから、「**データベースの変更 (Change Database)**」を「**選択済み (Selected)**」メニューから選択します。

関連タスク:

- 154 ページの『ディスクバリーによるデータベース接続の構成』
- 94 ページの『構成アシスタント (CA) によるデータベース接続の構成』
- インストールおよび構成 補足 の『プロファイルによるデータベース接続の構成』

START または STOP DATABASE MANAGER 処理中のエラー

概説およびインストールのセットアップ手順にしたがって、複数パーティションのある DB2 UDB Enterprise Server Edition を UNIX システムにセットアップしている場合、START または STOP DATABASE MANAGER の処理中に、以下のエラーを検出する場合があります。

SQL6048N

START または STOP DATABASE MANAGER 処理中に、通信エラーが発生しました。

説明: START または STOP DATABASE MANAGER コマンドが、`sql1lib/db2nodes.cfg` ファイルに定義されているすべてのノードとの接続を確立しようとしているときに、TCP/IP 通信エラーが発生しました。

最初に複数パーティションを持つ DB2 UDB Enterprise Server Edition をセットアップする場合の一般的な問題は、ノード間の通信に使用される、特定の DB2 サービスがセットアップされないことです。この問題を避けるためには、以下のステップを実行します。

1. リモート・コマンドの実行には、以下のネットワーク・サービスが使用可能になっていることを確認してください。

- rlogind
- rshd

UNIX オペレーティング・システムの中には、セキュリティ上の理由で、`rlogind` および `rsh` (`rshd`) ネットワーク・サービスがデフォルトで使用不可になっているものがあります (`Turbolinux` がその一例)。`rlogind` および `rshd` ネットワーク・サービスを使用可能にするには、`/etc/inetd.conf` ファイル内のネットワーク・サービス項目のコメントをはずします。

注: `rshd` および `rlogind` サービス項目のロケーションは、オペレーティング・システムによって異なります。`Redhat` バージョン 7 などの `Linux` ディストリビューションでは、ネットワーク・サービスは `xinetd.conf` ファイルにあります。詳しくは、オペレーティング・システムのドキュメンテーションを参照してください。

`rshd` および `rlogind` ネットワーク・サービスが使用可能かどうかを確認するには、各 `DB2` ノードで `rsh` を使用してコマンドを実行します。たとえば、以下のコマンドをインスタンス所有者として実行します。

```
rsh db2node2 "hostname"
```

`db2node2` は他の `DB2` ノードのうちの 1 つの `TCP/IP` ホスト名です。`rsh` を使用して各 `DB2` ノードでコマンドを実行し、各 `DB2` ノードがシステム上の他の `DB2` ノードと通信できるようにします。

2. `DB2` 高速コミュニケーション・マネージャー、`DB2` 接続サービス、および `DB2` 割り込みサービスの項目が、インスタンス内の各ノード上の `/etc/services` ファイルに存在することを確認してください。

各ノードの `/etc/services` ファイル内の `DB2` 項目は、以下の例と類似している必要があります。

```
# This is a DB2 V8 instance that is running a maximum
# of 4 multiple logical nodes per physical node

DB2_db2inst1 60000/tcp # This is a EEE instance
DB2_db2inst1_1 60001/tcp # that has four multiple logical nodes
DB2_db2inst1_2 60002/tcp
DB2_db2inst1_END 60003/tcp

# Ports for connection service and interrupt service

db2cdbinst1 50000/tcp # DB2 connection service
db2idb2inst1 50001/tcp # DB2 interrupt service
```

注:

- a. デフォルトで、DB2 は 60000/tcp ~ 60003/tcp の範囲の 4 つのポートをすべて予約します。インストール中に、これらのデフォルトを受け入れるか、または異なるポート番号と範囲の開始点を指定できます。
- b. DB2 接続サービスに指定する名前 (上の例で db2cdbinst1) は、データベース・マネージャー構成ファイル内の SVCENAME パラメーターに指定したものと同じ値でなければなりません。

DB2 Connect のバインディング・データベース・ユーティリティー

データベース・ユーティリティー (インポート、エクスポート、再編成、コマンド行プロセッサ) および DB2 CLI バインド・ファイルをデータベースで使用するためには、各データベースに対してあらかじめバインドしておく必要があります。ネットワーク環境で、別々のオペレーティング・システムで稼働している複数のクライアントや、別々のバージョンまたはサービス・レベルにある複数のクライアントを使用する場合、オペレーティング・システムと DB2 バージョンの各組み合わせごとにユーティリティーを 1 回ずつバインドする必要があります。

ユーティリティーのバインドにより、パッケージ が作成されます。これは単一のソース・ファイルから特定の SQL ステートメントを処理するのに必要な情報がすべて入っているオブジェクトです。

バインド・ファイルは、インストール・ディレクトリー (Windows では通常、sqllib です) の bnd ディレクトリーの下に、別の .lst ファイルと一緒にグループ化されています。各ファイルは、サーバーに固有のものであります。

手順:

ホストまたは iSeries データベースへのバインド

ユーティリティーおよびアプリケーションをホストまたは iSeries データベース・サーバーにバインドするには、ホストまたは iSeries サーバーに接続してから、以下のような例をテンプレートとして使用します。

```
connect to dbalias user userid using password
bind path/bnd/@ddcsmvs.lst blocking all sqlerror continue
messages mvs.msg grant public
connect reset
```

path は DB2PATH レジストリー値に対応します。

DB2 Universal Database へのバインド

データベースへのデータベース・ユーティリティーのバインド方法は、ワークステーションのオペレーティング・システムによって異なります。

- 構成アシスタントを使って以下を行います。
 1. 構成アシスタント (CA) を開始します。

2. ユーティリティをバインドする先のデータベースを選びます。
 3. 右クリックして、「バインド (Bind)」を選択します。
 4. バインドするユーティリティまたはファイルを選択します。
 5. 必要なバインド・オプションを追加します。
 6. データベースに接続するためのユーザー ID とパスワードを入力します。そのユーザー ID には、データベースに対して新パッケージをバインドするための権限がなければなりません。「バインド (Bind)」をクリックします。
- コマンド行プロセッサを使って以下を行います。
 1. `x:%sqllib%bnd` と入力して、`bnd` ディレクトリに移動します。ただし `x:` は、DB2 をインストールしているドライブです。
 2. データベースに接続するには、コマンド・センターまたはコマンド行プロセッサに以下のようなコマンドを入力します。

```
connect to database_alias
```

ただし、`database_alias` は、接続先のデータベースの名前です。

3. コマンド・センターまたはコマンド行プロセッサに以下のようなコマンドを入力します。

```
"bind @db2ubind.lst messages bind.msg grant public"  
"bind @db2cli.lst messages clibind.msg grant public"
```

この例では、`bind.msg` および `clibind.msg` は出力メッセージ・ファイルであり、EXECUTE および BINDADD 特権が `public` に付与されます。

4. 以下のようなコマンドを入力して、データベースへの接続をリセットします。

```
connect reset
```

注:

1. `db2ubind.lst` ファイルには、データベース・ユーティリティ用のパッケージを作成するのに必要なバインド (`.bnd`) ファイルのリストが入っています。`db2cli.lst` ファイルには、DB2 CLI および DB2 ODBC ドライバー用のパッケージを作成するのに必要なバインド (`.bnd`) ファイルのリストが入っています。
2. バインドは、完了するまでに数分かかることがあります。
3. BINDADD 権限をもっている場合は、DB2 CLI または ODBC ドライバーを最初に使用するとき、DB2 CLI パッケージが自動的にバインドされます。使用しているアプリケーションで、データベースをバインドする必要がある場合、構成アシスタントのバインド機能を使うか、またはコマンド行プロセッサを使って、バインド処置を実行することができます。

関連資料:

- コマンド・リファレンス の『BIND コマンド』
- 868 ページの『DB2 CLI のバインド・ファイルおよびパッケージ名』

ライセンス・センターを使用した DB2 Connect ライセンス・キーのインストール

ライセンス・センターを使用して、DB2 Connect 製品 CD-ROM からライセンス・キーをインストールすることができます。

手順:

ライセンス・センターを使用して、DB2 Connect 製品 CD-ROM からライセンス・キーをインストールするには、以下の操作を行います。

1. DB2 コントロール・センターを開始してから、「ツール (**Tools**)」メニューから「ライセンス・センター (**License Center**)」を選択します。
2. ライセンスをインストールするシステムを選択します。「インストール済み製品 (**Installed products**)」フィールドに、インストール済み製品の名前が表示されます。
3. 「ライセンス (**License**)」メニューから「追加 (**Add**)」を選択します。
4. 「ライセンスの追加 (Add License)」ウィンドウで、「ファイルから (**From a file**)」または「手動 (**Manually**)」のいずれかの追加を選択できます。

ファイルから (**From a file**)

「ファイルから (**From a file**)」ラジオ・ボタンを選択して、DB2 Connect 製品 CD-ROM からライセンス・ファイルを選択します。

- Windows では、以下のコマンドを入力します。

```
x:\db2\license\license_filename
```

x: は DB2 Connect 製品 CD の入っている CD-ROM ドライブです。

- UNIX では、以下のコマンドを入力します。

```
/db2/license/license_filename
```

license_filename は以下を表します。

db2conee.lic

DB2 Connect Enterprise Edition

db2conpe.lic

DB2 Connect Personal Edition

db2conue.lic

DB2 Connect Unlimited Edition

db2conas.lic

DB2 Connect Application Server Edition

手動 (Manually)

「手動 (Manually)」ラジオ・ボタンを選択して、「製品名 (Product name)」、「製品バージョン (Product version)」および「製品パスワード (Product password)」を入力します。

5. 「適用 (Apply)」をクリックしてライセンス・キーを追加します。

関連タスク:

- 122 ページの『db2licm コマンドを使用した DB2 Connect ライセンス・キーのインストール』
- 185 ページの『db2licm コマンドによるライセンス・タイプの設定』
- 123 ページの『ライセンス・センターを使用したライセンス・タイプの設定』

db2licm コマンドを使用した DB2 Connect ライセンス・キーのインストール

ライセンス・センターを使用する代わりに、**db2licm** コマンドを使って、DB2 Connect の製品 CD-ROM からライセンス・キーを追加することができます。

手順:

db2licm コマンドを使用して DB2 Connect の製品 CD-ROM からライセンス・キーを追加するには、以下のようにします。

- Windows では、以下のコマンドを入力します。

```
db2licm -a x:\db2\license\license_filename
```

x: は DB2 Connect の製品 CD の入っている CD-ROM ドライブです。

- UNIX では、以下のコマンドを入力します。

```
db2licm -a db2/license/license_filename
```

license_filename は以下を表します。

db2conee.lic

DB2 Connect Enterprise Edition

db2conpe.lic

DB2 Connect Personal Edition

db2conue.lic

DB2 Connect Unlimited Edition

db2conas.lic

DB2 Connect Application Server Edition

関連タスク:

- 121 ページの『ライセンス・センターを使用した DB2 Connect ライセンス・キーのインストール』
- 185 ページの『db2licm コマンドによるライセンス・タイプの設定』
- 123 ページの『ライセンス・センターを使用したライセンス・タイプの設定』

ライセンス・センターを使用したライセンス・タイプの設定

ライセンス・センターを使用したライセンス・タイプを設定できます。

手順:

ライセンス・タイプを設定するには、以下のようになります。

1. ライセンス・センターの「**ライセンス (License)**」メニューで「**変更 (Change)**」を選択します。
2. 「**ライセンスの変更 (Change License)**」ウィンドウで、購入したライセンスのタイプを選択します。
 - 同時ユーザー・ライセンスを購入していた場合、「**同時 DB2 Connect ユーザー (Concurrent DB2 Connect users)**」を選択してから、購入したユーザー・ライセンスの数を入力します。

注: DB2 Connect Enterprise Edition のライセンスは、1 人のユーザーにのみ供与されます。それ以外の DB2 Connect ユーザー・ライセンスは別途購入する必要があります。

登録済みユーザー・ライセンスを購入した場合、「**登録済み DB2 Connect ユーザー (Registered DB2 Connect users)**」を選択してから、「**OK**」をクリックして「**ライセンスの変更 (Change License)**」ウィンドウをクローズし、ライセンス・センターに戻ります。「**ユーザー (Users)**」タブをクリックし、購入したライセンスの対象のすべてのユーザー ID を追加します。

3. 「**制約ポリシー (Enforcement policy)**」を選択します。「**ソフト停止 (Soft stop)**」がデフォルトです。

関連タスク:

- 122 ページの『db2licm コマンドを使用した DB2 Connect ライセンス・キーのインストール』
- 121 ページの『ライセンス・センターを使用した DB2 Connect ライセンス・キーのインストール』
- 185 ページの『db2licm コマンドによるライセンス・タイプの設定』

DB2 Data Links Manager をインストールする前に (Solaris オペレーティング環境)

DB2[®] Data Links Manager をインストールする前に、まずこの情報をお読みください。

サポートされるオペレーティング・システム、カーネル・レベルおよびカーネル・アーキテクチャー

32 ビット・カーネルの以下のバージョンのいずれかを実行していることを確認してください。

- Solaris バージョン 7
- Solaris バージョン 8

カーネル・アーキテクチャー sun4d および sun4m はサポートされていません。

オペレーティング・システムのレベルを確認するには、**uname -r** コマンドを入力します。このコマンドは、Solaris バージョン 7 以降でのみ有効です。カーネル・レベルを確認するには、**isainfo -v** コマンドを入力します。この **isainfo** コマンドが、32-bit sparc applications を戻さなければなりません。

システムを 32 ビット・カーネルで始動するには、**OK** プロンプトで、**setenv boot-file kernel/unix** コマンドを入力します。その後、**boot** コマンドを入力して、ブート・プロセスを開始します。

ディスク・スペース要件

/opt/IBM/db2 ディレクトリに 85 MB 以上の空きディスク・スペースがあることを確認してください。使用可能な空きディスク・スペースを確認するには、**df -k /opt/IBM/db2** コマンドを入力します。

Data Links Manager 管理者ユーザー ID

インストール中に、「**Data Links インストールのためのユーザー情報の設定 (Set user information for the Data Links installation)**」パネルにおいて、DB2 Data Links インスタンスの所有者を作成するオプションを選択する機会があります。ここに示すインストール手順は、そのオプションを選択したことを前提としたものです。

Data Links Manager 管理者ユーザー ID を作成すると、DB2 セットアップ・ウィザードは、ユーザー名 *dlfm* およびパスワード *ibmdb2* でこのユーザーを追加します。これらのデフォルト値を使用するか、既存のユーザー名を指定するか、またはデフォルト値を変更して別のユーザー名を作成することができます。デフォルトの値は、すべての DB2 Data Links のインストールで使用されるため、公に知られています。したがって、セキュリティ上の理由から、独自のユーザー名とパスワードを指定することをお勧めします。Data Links Manager 管理者ユーザー ID は、Data Links Manager インスタンスのためにも使用されます。

ご使用のマシンが NIS クライアントの場合、独自の既存のユーザー名を指定する必要があります。指定する既存のユーザー名には、以下の条件がありません。

- そのホーム・ディレクトリーが、データ・リンク・ファイル・システム・フィルターを使用するファイル・システムに常駐していない。
- ユーザー名が 8 文字以下である。
- root 権限を持つユーザーではない。

DB2 Data Links Manager の管理者のためのユーザー ID を作成するには、

1. root 権限があるユーザーとしてログインします。
2. DB2 Data Links 管理者のグループを作成します (たとえば `dlfmgrp`)。

```
groupadd dlfmgrp
```

3. 書き込みグループを作成します (たとえば `dlfmxgrp`)。

```
groupadd dlfmxgrp
```

WRITE PERMISSION ADMIN によって定義された DATALINK 列にファイルがリンクされた場合、そのファイルはこのグループによって所有されるように変更されます。

DLFMXGRP グループは、WRITE PERMISSION ADMIN によって定義された DATALINK 列にリンクされたファイルに関するインプレース更新機能で使用するために作成されます。DLFMGRP グループも作成されます。それらのグループには、ユーザー ID を追加しないでください。DLFM の所有するファイルを作成するユーザー ID は、DLFM だけでなければなりません。

4. Data Links 管理者のホーム・ディレクトリーとして `/home/dlfm` ディレクトリーを使用して、DB2 Data Links 管理者のユーザー ID を作成します (たとえば `dlfm`)。

```
useradd -g dlfmgrp -G dlfmxgrp -d /home/dlfm dlfm
```

5. `passwd user` コマンドを入力して、このユーザー名にパスワードを割り当てます (`user` は作成したアカウント)。

DB2 Data Links Manager 管理者のユーザー ID は、データ・リンク・ファイル・マネージャーを管理するためだけでなく、DLFM_DB およびリンク状態のファイルを所有するためのものでもあります。DB2 Data Links Manager 管理者は、ファイル・システム上にデータ・リンク・ファイル・システム・フィルター (DLFF) によって制御されるディレクトリーを所有するべきではありません。DB2 Data Links Manager 管理者のユーザー ID は、データ・リンク・ファイル・マネージャー (DLFM) を管理するためののみ使用されるべきです。DB2 Data Links Manager 管理者のユーザー ID をファイルの所有者にするこ

とは可能ですが、それは READ PERMISSION DB のリンク先ファイルについてだけです。DB2 Data Links Manager 管理者を所有者とするファイルを手動で作成しないようにしてください。

デフォルト値を変更して別のユーザー名を作成する場合は、必ず、指定するユーザー名が 8 文字以下であることを確認してください。

DB2 Data Links Manager 管理者ユーザー ID のディスク・スペース要件

DB2 Data Links Manager 管理者のホーム・ディレクトリーが常駐するホーム・ディレクトリーに、70 MB 以上の空きディスク・スペースがあることを確認してください。使用可能な空きディスク・スペースを確認するには、**df -k INSTHOME** のコマンドを入力します (INSTHOME は Data Links 管理者ユーザー ID のホーム・ディレクトリー)。

メモリー要件

システムが使用できるメモリーが 256 MB 以上あることを確認してください。使用可能なメモリーを確認するには、**/usr/bin/dmesg | grep -i "avail mem"** コマンドを入力します。

カーネル・パラメーターの更新

DB2 セットアップ・ウィザードを使用するか、または **db2_install** および **pkgadd** コマンドを使用して、DB2 (Solaris 版) 製品をインストールする前に、ご使用のシステムのカーネル構成パラメーターを更新する必要がある可能性があります。表 31 の値は、推奨される Solaris カーネル構成パラメーターです。

注: カーネル構成パラメーターを更新した後、マシンを再始動する必要があります。

表 31. Solaris バージョン 7 カーネル構成パラメーターの推奨値

カーネル・パラメーター	物理メモリー			
	64MB - 128MB	128MB - 256MB	256MB - 512MB	512MB+
msgsys:msginfo_msgmax	65535 (1)	65535 (1)	65535 (1)	65535 (1)
msgsys:msginfo_msgmnb	65535 (1)	65535 (1)	65535 (1)	65535 (1)
msgsys:msginfo_msgmap	130	258	258	258
msgsys:msginfo_msgmni	128	256	256	256
msgsys:msginfo_msgssz	16	16	16	16
msgsys:msginfo_msgtql	256	512	1024	1024
msgsys:msginfo_msgseg	8192	16384	32767 (2)	32767 (2)
shmsys:shminfo_shmmax	67108864	134217728 (2)	268435456 (3)	536870912 (3)
shmsys:shminfo_shmseg	50	50	50	50
shmsys:shminfo_shmmni	300	300	300	300
semsys:seminfo_semmni	128	256	512	1024
semsys:seminfo_semmap	130	258	514	1026
semsys:seminfo_semmns	256	512	1024	2048
semsys:seminfo_semmnu	256	512	1024	2048
semsys:seminfo_semtime	50	50	50	50

表 31. Solaris バージョン 7 カーネル構成パラメーターの推奨値 (続き)

カーネル・パラメーター	物理メモリー			
	64MB - 128MB	128MB - 256MB	256MB - 512MB	512MB+
dlfsdrv:glob_mod_pri ¹	0x100800	0x100800	0x100800	0x100800
dlfsdrv:glob_mesg_pri ¹	0xff	0xff	0xff	0xff
dlfsdrv:ConfigDlfsUid ¹	UID ²	UID ²	UID ²	UID ²
dlfsdrv:ConfigDlfsGid ¹	GID ³	GID ³	GID ³	GID ³
¹ これらの値は必須です。 ² UID は Data Links Manager 管理者のユーザー ID です。 ³ GID は書き込みグループのグループ ID です。 『Data Links Manager 管理者のユーザー ID』のセクションを参照してください。				

表 32. Solaris バージョン 8 カーネル構成パラメーターの推奨値

カーネル・パラメーター	物理メモリー
	512MB+
msgsys:msginfo_msgmax	65535
msgsys:msginfo_msgmnb	65535
msgsys:msginfo_msgmap	1026
msgsys:msginfo_msgmni	1024
msgsys:msginfo_msgssz	32
msgsys:msginfo_msgtql	2048
msgsys:msginfo_msgseg	32767 (2)
shmsys:shminfo_shmmax	0xe0000000
shmsys:shminfo_shmseg	500
shmsys:shminfo_shmmni	1024
semsys:seminfo_semmni	1024
semsys:seminfo_semmap	1026
semsys:seminfo_semmns	2048
semsys:seminfo_semmnu	2048
semsys:seminfo_semume	50
dlfsdrv:glob_mod_pri ¹	0x100800
dlfsdrv:glob_mesg_pri ¹	0xff
dlfsdrv:ConfigDlfsUid ¹	UID ²
dlfsdrv:ConfigDlfsGid ¹	GID ³
¹ これらの値は必須です。 ² UID は Data Links Manager 管理者のユーザー ID です。 ³ GID は書き込みグループのグループ ID です。 『Data Links Manager 管理者のユーザー ID』のセクションを参照してください。	

DB2 Data Links および DB2 Universal Database™ のバージョン・レベル

DB2 Universal Database は、バージョン 6.1、7.1、7.2、および 8.1 のいずれかです。Data Links Manager は、バージョン 7.1、7.2、および 8.1 のいずれかです。たとえば、バージョン 7.1 またはバージョン 8.1 の DB2 UDB と、バージョン 8.1 の Data Links Manager を組み合わせることができます。ワークステーション上の DB2 のバージョンを確認するには、**db2level** コマンドを入力します。DB2 UDB と Data Links Manager のそれらのバージョンは、任意の組み合わせで使用できます。

バージョン 8.1 など、もっと後のバージョンの DB2 でバージョン 7 の Data Links Manager に接続した場合、バージョン 8.1 から導入された Data Links Manager の機能は利用できません。バージョン 8 の機能を使用しようとすると、SQL エラーになります。

以前のバージョンからの移行

レベルの異なる DB2 Data Links Manager の複数のインスタンスを、同一のマシン上に置くことはできません。ご使用のシステムに 8.1 より前のバージョンの Data Links インスタンスがある場合、**db2imigr** コマンドを使用して、そのインスタンスをバージョン 8.1 形式に移行する必要があります。

レジストリー変数には、環境変数よりも高い機能と柔軟性が備わっていますが、逆にこのことが原因となって、移行が期待どおりに実行されない場合があります。移行後には、**db2set** コマンドを使用してレジストリー変数をチェックする必要があります。

Data Links サーバーのセキュリティー

DB2 Data Links バージョン 8.1 より前のリリースには、ファイルのリンクを制御するセキュリティー機能はありませんでした。旧リリースから移行した場合、DLFM 移行時に、既存の DLFM サーバーに対してセキュリティー制御を有効にするオプションが提供されます。

DB2 Data Links バージョン 7.1 または 7.2 から、DB2 Data Links Manager バージョン 8 への移行

DB2 Data Links バージョン 7.1 または 7.2 から、DB2 Data Links Manager バージョン 8.1 に移行するには、

1. DB2 および DLFM データベースで移行の準備をします。たとえば、既存のデータベースをバックアップするとよいでしょう。
2. DB2 サーバーおよび Data Links Manager サーバー・マシンに DB2 バージョン 8 をインストールします。これらの作業については、以下で詳細に説明します。
3. DLFM として、**db2dlimg** コマンドを実行します。

DB2 Data Links Manager バージョン 8 に移行すると、以下の環境変数が DB2 レジストリー変数に変換されます。

```
DLFM_INSTALL_PATH
DLFM_PORT
DLFM_LOG_LEVEL
DB2_RR_TO_RS
DLFM_BACKUP_DIR_NAME (1)
DLFM_BACKUP_TARGET (2)
DLFM_BACKUP_TARGET_LIBRARY (3)
```

注:

1. この変数は、ローカル・ファイル・システムがバックアップのターゲットである場合にのみ使用されます。

2. この変数は、使用されるバックアップ・ターゲットのタイプを示します。この変数の値には、LOCAL、TSM、または XBSA があります。
3. この変数は、DLFM_BACKUP_TARGET が XBSA に設定されている場合のみ、XBSA のサポート・コード・ライブラリーを示します。サポート・コード・ライブラリーは、共有オブジェクト名を含む完全修飾パス名で示されます (たとえば /opt/IBM/db2/v8.1/Legato/libxdb2.so(bsashr10.o))。共有オブジェクトの名前 (この例では bsashr10.o) は、XBSA 準拠の共有ライブラリーを提供しているベンダーから入手することができます。

DLFM データベース移行ユーティリティー

DLFM データベース移行ユーティリティー **db2dlmng** は、既存の DLFM データベースを DB2 バージョン 7.1、またはバージョン 7.2 フォーマットから、新規 DB2 バージョン 8.1 フォーマットに変換します。このユーティリティーを実行する必要がある DB2 データベースのみが、DLFM サーバーに常駐するものです。このデータベースは DLFM_DB と呼ばれ、デフォルトでは DLFM という Data Links Manager 管理者に所有される DB2 インスタンスに属しています。このデータベースの各オカレンスは、DLFM サーバーごとに、**db2dlmng** ユーティリティーを使用して移行されなければなりません。

db2dlmng ユーティリティーは INSTHOME/sql1lib/adm ディレクトリーにインストールされます。INSTHOME は Data Links Manager 管理者のホーム・ディレクトリーです。

db2dlmng ユーティリティーを実行する前に、DB2 バージョン 8.1 Data Links Manager がインストールされている必要があります。DLFM_DB データベースに対して、古いリリースの **db2dlmng** ユーティリティーを実行しようとしてはいけません。また、DB2 インスタンス移行 (**db2imigr**) ユーティリティーを、DLFM データベース・インスタンスに対して実行済みでなければなりません。

DLFM データベース移行ユーティリティーは、以下の基本的なステップにしたがって実行されます。

1. DLFM が実行中の場合は、停止します。
2. バージョン 8.1 Data Links Manager リンク・ファイル・セキュリティー管理をセットアップする方法を要求します。
3. 現行の DLFM_DB の内容が、有効なバージョン 8.1 以前のフォーマットかどうかを検証します。
4. 現行の DLFM_DB データベースが自動的にバックアップされます。
5. このデータベースに予約されているログ・スペースの量を増やします。
6. 新規のバッファー・プールおよび表スペースを作成します。
7. 表および索引を作成、変更します。
8. 最も大きい DLFM_DB 表を新規表スペースのうちの 1 つに移動します。

9. リンク・ファイル・セキュリティー管理を、ステップ 2 に指定されたようにセットアップします。
10. DLFM 実行可能ファイルを、変更されたデータベースに再バインドします。

移行ユーティリティーが処理中にエラーを検出したときは、エラー・メッセージおよび続行する方法の説明が示されます。問題を訂正して、**db2dlmmg** を再実行できる場合もあります。しかし多くの場合、まず初めに元の DLFM_DB (上のステップ 4 でバックアップしたもの) をリストアし、**db2dlmmg** を再実行するように指示されます。ステップ 4 のバックアップは、DLFM 構成変数 DLFM_BACKUP_TARGET および DLFM_BACKUP_DIR_NAME で指定されたロケーションに保管されています。元のデータベースをリストアするには、以下のように、DB2 リストア・ユーティリティーを実行します。

- db2start
 - db2 restore database dlfm_db from <backup-directory>[taken at <date-time>] without rolling forward
- <backup-directory> は、DLFM_BACKUP_DIR_NAME で指定された完全修飾パスを示し、<backup-directory> の下に複数のバックアップ・イメージがある場合は、taken at <date-time> を指定しなくてはなりません。

構成変数 DLFM_BACKUP_TARGET および DLFM_BACKUP_DIR_NAME の値を表示するには、以下のコマンドを入力します。

- db2set DLFM_BACKUP_TARGET
- db2set DLFM_BACKUP_DIR_NAME

DLFM_BACKUP_TARGET 変数の値が、「TSM」または「XBSA」の場合、上に示した必要なリストア・コマンドは異なります。TSM または XBSA アーカイブ・サーバーのバックアップへアクセスする RESTORE ユーティリティーの説明を、参照してください。

DLFM データベース移行ユーティリティーを実行するには、以下のステップを各 DLFM サーバーで実行します。

1. Data Links Manager 管理者 ID を使用して、DLFM サーバーにログインします。
2. DLFM_BACKUP_DIR_NAME 構成変数が指しているファイル・システムに、十分なフリー・スペースがあることを確認してください。少なくとも、DLFM_DB データベースの全バックアップに十分なスペースが必要です。(少なくとも 2 つの DLFM_DB バックアップ分は必要です。詳しくは下のステップ 7 を参照してください。)
3. DLFM インスタンス・ファイル・システムに、増加するトランザクション・ログ・ファイルに十分なフリー・スペースがあることを確認してください。

い。LOGPRIMARY 構成変数は、6 まで増加します。(それ以前が 6 以下の場合) LOGFILSIZ 構成変数は 2000 まで増加します (それ以前が 2000 以下の場合)。これらの変数の現行設定値を表示するには、**db2 get db cfg for dlfm_db** コマンドを入力します。

- DLFM インスタンス・ファイル・システムに、作業ファイル (**db2dlmmg** によって使用) に十分な追加フリー・スペースがあるかどうかを確認してください。現在 DLFM_DB によって使用されているスペースの量を、この追加のスペースに必要な量の見積もりとして使用できます。
- db2dlmmg** コマンドを入力して、移行ユーティリティを実行します。
- 移行ユーティリティが完了するまで中断することなく実行するようにしてください。DLFM_DB のサイズによって、数分から 1 時間ほどかかる場合があります。移行の進行は、さまざまな状況メッセージで、画面に報告されます。エラーが発生した場合は、エラー・メッセージの説明にしたがって処置を行い、その後で **db2dlmmg** コマンドを再実行してください。
- 一度移行ユーティリティが正常に完了したら、バージョン 8.1 で、DLFM サーバーのフル・リカバリーのために、手動で DLFM_DB のフル・バックアップを行う必要があります (移行ユーティリティは、これは自動的にはいけません。)

注意すべきいくつかの点:

- 一度上の手順を完了したら、バージョン 8.1 DLFM 操作をすべて実行できるようになります。 *dlfm start* コマンドを実行して、バージョン 8.1 Data Links File Manager が、新規の DLFM_DB で正常に開始されたかどうかを確認します。DLFM の実行を確認するには、**dlfm see** コマンドを使用して DLFM プロセスをチェックできます。
- Data Links Manager インストール・ドキュメンテーションにある、インストールの妥当性検査の説明に従うか、または既存の Data Links アプリケーションを使用して Data Links Manager の正しい操作を検証する必要があります。
- 移行の前に、DLFM_DB のいかなる バックアップも、リストアしようとしてください。バージョン 8.1 以前のバックアップは、バージョン 8.1 Data Links Manager では使用できません。
- 移行で問題がある場合は、IBM[®] サービスに連絡してください。IBM サービスの援助なしに、DLFM_DB の内容を手動で更新しようとは、決してしないでください。

DLFM ユーザー・アカウント

DB2 Data Links Manager 管理者のユーザー・アカウントに加えて、インストールの際には、データ・リンク・ファイル・マネージャーで使用する DLFM ユーザー・アカウントも作成されます。DLFM ユーザー・アカウントは、すべての READ PERMISSION DB ファイルの所有者となります。

TCP/IP ポート番号

データ・リンク・ファイル・マネージャーに使用できる TCP/IP ポートが必要です。デフォルトでは、DB2 セットアップ・ウィザードが値を生成しますが、その値を使用するか、または独自の値を指定できます。インストールを検証するには、このポート番号を知っていなければなりません。

独自のポート番号を指定したい場合は、`/etc/services` ファイルを開いて、すでにマシンで使用されている TCP/IP ポートを調べてください。インストールの際にはこのポートを指定する必要があります。

DLFM に使用する TCP/IP ポート番号は、いったん選択したら変更しないでください。

ホスト名の決定

DB2 サーバーおよび Data Links サーバーのそれぞれの名前を決定する必要があります。インストールを検証するには、これらのホスト名を知っていなければなりません。DB2 データ・リンク・ファイル・マネージャーに接続する際、DB2 UDB サーバーは内部で以下の情報を DLFM に送信します。

- データベース名
- インスタンス名
- ホスト名

次に DLFM は、この情報と内部の表とを比較して、接続が可能かどうかを判断します。接続が可能なのは、データベース名、インスタンス名、およびホスト名のこの組み合わせが、`dlfm add_db` コマンドを使用して登録されている場合だけです。`dlfm add_db` コマンドで使用されるホスト名は、DB2 UDB サーバーによって内部的に送信されるホスト名と正確に一致していなければなりません。

DLFM は、これらの情報を検証して、この特定の DB2 サーバーからの接続を許可するかどうかを決定します。DB2 サーバー上でホスト名の情報を取得するサブルーチン `gethostbyname` は、`/etc/resolv.conf` というファイルを探します。このファイルが存在すると、このサブルーチンはまず、ドメイン・ネーム・サーバーに照会します。DNS に対する要求がタイムアウトになると、`gethostbyname` ルーチンはローカルの `/etc/hosts` ファイルを調べます。DLFM に正しく接続するには、DLFM に登録されている名前と、`gethostbyname` ルーチンによって DB2 UDB サーバー上で取得される名前とが一致していなければなりません。

`dlfm add_db` コマンドを使用して DB2 UDB データベースを登録する際には、ホスト名として `uname -n` コマンドの出力を使用します。`dlfm add_db` コマンドにおいてその他の別名が使用されていると、DB2 サーバーと DLFM の内部接続は失敗します。

これを、DB2 Data Links サーバーと DB2 UDB サーバーのそれぞれについて繰り返してください。

Data Links サーバーを DB2 データベースに登録するには、DB2 の **add datalinks manager for database *database_alias* using node *hostname* port *port_number*** というコマンドを使用します。

hostname は、Data Links サーバーの名前です。このコマンドでは、Data Links サーバーの有効な別名を使用することもできます。この Data Links サーバーへの参照である DATALINK 値では、URL 値でホスト名を指定する必要があります。つまり、DATALINK 列に URL 値を代入する際には、**add datalinks manager** コマンドで使用された正確な名前を使用する必要があります。異なる別名を使用すると、SQL ステートメントが失敗します。

システム・クロックの同期化

Data Links サーバーとリモート DB2 サーバーでは、READ PERMISSION DB オプションによって、DATALINKS 列のリンク先ファイルのために、システム・クロックが常に同期化されている必要があります。クロックの同期化は、Data Links でトークンの有効期限を正しく機能させる上で不可欠です。トークン有効期限は、選択した DATALINK 値 (URL と埋め込みファイル許可トークンで構成される) を使用できる期間を制御するデータベース構成パラメーターです。システムの時刻と日付を確認するには、**date** コマンドを入力します。

システム・クロックの同期化に関する詳細については、*Solaris* 管理の手引きを参照してください。

Data Links Replication

Data Links Manager バージョン 8.1 では、DLFM_START_ASNCOPYD および DLFM_ASNCOPYD_PORT という 2 つの新しいレジストリー変数が導入されました。インストール中に、Data Links Manager Replication デーモンを有効にするように促されます。Data Links Replication を使用する予定であれば、インストール中に Replication デーモンを有効にしてください。それは、レジストリー変数 (DLFM_START_ASNCOPYD および DLFM_ASNCOPYD_PORT) を設定して DLFM を再開することによって、インストール後にも有効にできます。

関連概念:

- *Data Links Manager* 概説およびインストールの『DB2 Data Links Manager』
- 396 ページの『DB2 サーバーのインストールの概要 (Windows)』
- *DB2 Data Links Manager* 管理ガイドおよびリファレンスの『Data Links Manager のセキュリティの概要』

関連タスク:

- *Data Links Manager* 概説およびインストールの『DB2 セットアップ・ウィザードによる DB2 Data Links Manager のインストール (Solaris オペレーティング環境)』

- *Data Links Manager 概説*およびインストールの『db2_install コマンドによる DB2 Data Links Manager の手動インストール (Solaris オペレーティング環境)』
- 394 ページの『データベースの移行』
- *DB2 Universal Database サーバー機能 概説*およびインストールの『DB2 の移行 (UNIX)』
- *DB2 Data Links Manager 管理ガイド*およびリファレンスの『Data Links Manager レプリケーション・デーモンの使用可能化』

DB2 Data Links Manager をインストールする前に (AIX)

DB2[®] Data Links Manager をインストールする前に、まずこの情報をお読みください。このセクションの情報は、JFS に対する Tivoli[®] Space Manager FSM 拡張を含め、JFS 環境でのインストールと DCE-DFS 環境でのインストールの両方に共通です。JFS および DCE-DFS に固有の情報については、それぞれ別個のセクションで扱います。

DB2 セットアップ・ウィザードまたは SMIT による DB2 Data Links のインストール
DB2 Data Links Manager を JFS または DCE-DFS 環境でインストールする場合は、DB2 セットアップ・ウィザードを使用することを強くお勧めします。DB2 セットアップ・ウィザードは、Data Links Manager のセットアップと構成に関連したほとんどすべての作業を実行します。SMIT または db2_install を使用して DB2 Data Links Manager をインストールする場合は、Data Links Manager システムのセットアップと構成を手動で実行する必要があります。

AIX[®] 5L バージョン 5.1 のサポート

データ・リンク・ファイル・マネージャー (DLFM) およびデータ・リンク・ファイル・システム・フィルター (DLFF) の各コンポーネントは、AIX 5L[™] バージョン 5.1 において完全にサポートされるようになりました。AIX 5L バージョン 5.1 では、Data Links Manager に関連したツールや命令、AIX のそれ以前リリースでサポートされていたツールや命令がすべて完全にサポートされており、適用可能になっています。

ディスク・スペース要件

/usr/opt ディレクトリーに 85 MB 以上の空きディスク・スペースがあることを確認してください。使用可能な空きディスク・スペースを確認するには、**df -k /usr/opt** コマンドを入力します。

メモリー要件

システムが使用できるメモリーが 256 MB 以上あることを確認してください。使用可能なメモリーを確認するには、**lsattr -l sys0 -E -a realmem** コマンドを入力します。

以前のバージョンからの移行

レベルの異なる DB2 Data Links Manager の複数のインスタンスを、同一のマシン上に置くことはできません。ご使用のシステムに 8.1 より前のバージョン

の Data Links インスタンスがある場合、 **db2imigr** コマンドを使用して、そのインスタンスをバージョン 8.1 形式に移行する必要があります。

レジストリー変数には、環境変数よりも高い機能と柔軟性が備わっていますが、逆にこのことが原因となって、移行が期待どおりに実行されない場合があります。移行の後には、 **db2set** コマンドを使用してレジストリー変数を調べることができます。

Data Links サーバーのセキュリティー

DB2 Data Links バージョン 8.1 より前のリリースには、ファイルのリンクを制御するセキュリティー機能はありませんでした。旧リリースから移行した場合、DLFM 移行時に、既存の DLFM サーバーに対してセキュリティー制御を有効にするオプションが提供されます。

DB2 Data Links バージョン 6.1、7.1、または 7.2 から、DB2 Data Links Manager バージョン 8 への移行

DB2 Data Links Manager バージョン 6.1、7.1、および 7.2 から、DB2 Data Links Manager バージョン 8.1 に移行する場合、以下のステップを実行する必要があります。

1. DB2 および DLFM データベースで移行の準備をします。たとえば、既存のデータベースをバックアップするとよいでしょう。
2. DB2 サーバーおよび Data Links Manager サーバー・マシンに DB2 バージョン 8 をインストールします。
3. DLFM として、**db2dlimg** コマンドを実行します。

DB2 Data Links Manager バージョン 8 に移行すると、以下の環境変数が DB2 レジストリー変数に変換されます。

```
DLFM_INSTALL_PATH
DLFM_PORT
DLFM_LOG_LEVEL
DB2_RR_TO_RS
DLFM_BACKUP_DIR_NAME (1)
DLFM_BACKUP_TARGET (2)
DLFM_BACKUP_TARGET_LIBRARY (3)
```

注:

1. この変数は、ローカル・ファイル・システムがバックアップのターゲットである場合にのみ使用されます。
2. この変数は、使用されるバックアップ・ターゲットのタイプを示します。この変数の値には、LOCAL、TSM、または XBSA があります。
3. この変数は、DLFM_BACKUP_TARGET が XBSA に設定されている場合にのみ、XBSA のサポート・コード・ライブラリーを示します。サポート・コード・ライブラリーは、共有オブジェクト名を含む完全修飾パス名で示されます (たとえば /usr/lpp/Legato/libxdb2.a(bsashr10.o))。共有オブジ

エクトの名前 (この例では bsashr10.o) は、XBSA 準拠の共有ライブラリーを提供しているベンダーから入手することができます。

DLFM データベース移行ユーティリティー

DLFM データベース移行ユーティリティー **db2dlmng** は、既存の DLFM データベースを DB2 バージョン 6.1、バージョン 7.1、またはバージョン 7.2 フォーマットから、新規 DB2 バージョン 8.1 フォーマットに変換します。このユーティリティーを実行する必要がある DB2 データベースのみが、DLFM サーバーに常駐するものです。このデータベースは **DLFM_DB** と呼ばれ、デフォルトでは **DLFM** という Data Links Manager 管理者に所有される DB2 インスタンスに属しています。このデータベースの各オカレンスは、DLFM サーバーごとに、**db2dlmng** ユーティリティーを使用して移行されなければなりません。

db2dlmng ユーティリティーは `INSTHOME/sql1lib/adm` ディレクトリーにインストールされます。INSTHOME は Data Links Manager 管理者のホーム・ディレクトリーです。

db2dlmng ユーティリティーを実行する前に、DB2 バージョン 8.1 Data Links Manager がインストールされている必要があります。DLFM_DB データベースに対して、古いリリースの **db2dlmng** ユーティリティーを実行しようとしてはいけません。また、DB2 インスタンス移行 (**db2imigr**) ユーティリティーを、DLFM データベース・インスタンスに対して実行済みでなければなりません。

DLFM データベース移行ユーティリティーは、以下の基本的なステップにしたがって実行されます。

1. DLFM が実行中の場合は、停止します。
2. バージョン 8.1 Data Links Manager リンク・ファイル・セキュリティー管理をセットアップする方法を要求します。
3. 現行の DLFM_DB の内容が、有効なバージョン 8.1 以前のフォーマットかどうかを検証します。
4. 現行の DLFM_DB データベースが自動的にバックアップされます。
5. このデータベースに予約されているログ・スペースの量を増やします。
6. 新規のバッファー・プールおよび表スペースを作成します。
7. 表および索引を作成、変更します。
8. 最も大きい DLFM_DB 表を新規表スペースのうちの 1 つに移動します。
9. リンク・ファイル・セキュリティー管理を、ステップ 2 に指定されたようにセットアップします。
10. DLFM 実行可能ファイルを、変更されたデータベースに再バインドします。

移行ユーティリティーが処理中にエラーを検出したときは、エラー・メッセージおよび続行する方法の説明が示されます。問題を訂正して、**db2dlimg** を再実行できる場合もあります。しかし多くの場合、まず初めに元の DLFM_DB (上のステップ 4 でバックアップしたもの) をリストアし、**db2dlimg** を再実行するように指示されます。ステップ 4 のバックアップは、DLFM 構成変数 DLFM_BACKUP_TARGET および DLFM_BACKUP_DIR_NAME で指定されたロケーションに保管されています。元のデータベースをリストアするには、以下のように、DB2 リストア・ユーティリティーを実行します。

- db2start
 - db2 restore database dlfm_db from <backup-directory>[taken at <date-time>] without rolling forward
- <backup-directory> は、DLFM_BACKUP_DIR_NAME で指定された完全修飾パスを示し、<backup-directory> の下に複数のバックアップ・イメージがある場合は、taken at <date-time> を指定しなくてはなりません。

構成変数 DLFM_BACKUP_TARGET および DLFM_BACKUP_DIR_NAME の値を表示するには、以下のコマンドを入力します。

- db2set DLFM_BACKUP_TARGET
- db2set DLFM_BACKUP_DIR_NAME

DLFM_BACKUP_TARGET 変数の値が、「TSM」または「XBSA」の場合、上に示した必要なリストア・コマンドは異なります。TSM または XBSA アーカイブ・サーバーのバックアップへアクセスする RESTORE ユーティリティーの説明を、参照してください。

DLFM データベース移行ユーティリティーを実行するには、以下のステップを各 DLFM サーバーで実行します。

1. Data Links Manager 管理者 ID を使用して、DLFM サーバーにログインします。
2. DLFM_BACKUP_DIR_NAME 構成変数が指しているファイル・システムに、十分なフリー・スペースがあることを確認してください。少なくとも、DLFM_DB データベースの全バックアップに十分なスペースが必要です。(少なくとも 2 つの DLFM_DB バックアップ分は必要です。詳しくは下のステップ 7 を参照してください。)
3. DLFM インスタンス・ファイル・システムに、増加するトランザクション・ログ・ファイルに十分なフリー・スペースがあることを確認してください。LOGPRIMARY 構成変数は、6 まで増加します。(それ以前が 6 以下の場合) LOGFILSIZ 構成変数は 2000 まで増加します (それ以前が 2000 以下の場合)。これらの変数の現行設定値を表示するには、**db2 get db cfg for dlfm_db** コマンドを入力します。

4. DLFM インスタンス・ファイル・システムに、作業ファイル (**db2dlmmg** によって使用) に十分な追加フリー・スペースがあるかどうかを確認してください。現在 DLFM_DB によって使用されているスペースの量を、この追加のスペースに必要な量の見積もりとして使用できます。
5. **db2dlmmg** コマンドを入力して、移行ユーティリティを実行します。
6. 移行ユーティリティが完了するまで中断することなく実行するようにしてください。DLFM_DB のサイズによって、数分から 1 時間ほどかかる場合があります。移行の進行は、さまざまな状況メッセージで、画面に報告されます。エラーが発生した場合は、エラー・メッセージの説明にしたがって処置を行い、その後で **db2dlmmg** コマンドを再実行してください。
7. 一度移行ユーティリティが正常に完了したら、バージョン 8.1 で、DLFM サーバーのフル・リカバリーのために、手動で DLFM_DB のフル・バックアップを行う必要があります (移行ユーティリティは、これは自動的にはいけません。)

注意すべきいくつかの点:

- 一度上の手順を完了したら、バージョン 8.1 DLFM 操作をすべて実行できるようになります。 *dlfm start* コマンドを実行して、バージョン 8.1 Data Links File Manager が、新規の DLFM_DB で正常に開始されたかどうかを確認します。DLFM の実行を確認するには、 **dlfm see** コマンドを使用して DLFM プロセスをチェックできます。
- Data Links Manager インストール・ドキュメンテーションにある、インストールの妥当性検査の説明に従うか、または既存の Data Links アプリケーションを使用して Data Links Manager の正しい操作を検証する必要があります。
- 移行の前に、DLFM_DB のいかなる バックアップも、リストアしようとしないうでください。バージョン 8.1 以前のバックアップは、バージョン 8.1 Data Links Manager では使用できません。
- 移行で問題がある場合は、IBM® サービスに連絡してください。IBM サービスの援助なしに、DLFM_DB の内容を手動で更新しようとは、決してしないでください。

TCP/IP ポート番号

データ・リンク・ファイル・マネージャーに使用できる TCP/IP ポートが必要です。デフォルトでは、DB2 セットアップ・ウィザードが値を生成しますが、その値を使用するか、または独自の値を指定できます。インストールを検証するには、このポート番号を知っていなければなりません。

独自のポート番号を指定したい場合は、`/etc/services` ファイルを開いて、すでにマシンで使用されている TCP/IP ポートを調べてください。インストールの際にはこのポートを指定する必要があります。

DLFM に使用する TCP/IP ポート番号は、いったん選択したら変更しないでください。

システム・クロックの同期化

Data Links サーバーとリモート DB2 サーバーでは、READ PERMISSION DB オプションによって、DATALINKS 列のリンク先ファイルのために、システム・クロックが常に同期化されている必要があります。クロックの同期化は、Data Links でトークンの有効期限を正しく機能させる上で不可欠です。トークン有効期限は、選択した DATALINK 値 (URL と埋め込みファイル許可トークンで構成される) を使用できる期間を制御するデータベース構成パラメーターです。

システムの時刻と日付を確認するには、**date -u** コマンドを入力します。

システム・クロックの同期化に関する詳細については、*AIX 管理の手引き* を参照してください。

Data Links Manager 管理者ユーザー ID

インストール中に、「Data Links インストールのためのユーザー情報の設定 (Set user information for the Data Links installation)」パネルにおいて、DB2 Data Links インスタンスの所有者を作成するオプションを選択する機会があります。ここに示すインストール手順は、そのオプションを選択しており、それを既存のユーザー ID で置き換えたいと考えていることを前提としたものです。

既存の独自のユーザー ID を指定する場合、そのアカウントは以下の条件を満たしていなければなりません。

- ホーム・ディレクトリーが、データ・リンク・ファイル・システム・フィルターを使用しているファイル・システムに常駐していない。
- ユーザー名が 8 文字以下である。
- root 権限を付与されたユーザー ID であってはならない。

DB2 Data Links Manager の管理者として使用するユーザー ID を作成するには、

1. root 権限を付与されたユーザー ID としてログインします。
2. DB2 Data Links 管理者のグループを作成します (たとえば `dlfmggrp`)。

```
mkgroup dlfmggrp
```

3. 書き込みグループを作成します (たとえば `dlfmxgrp`)。

```
mkgroup dlfmxgrp
```

WRITE PERMISSION ADMIN によって定義された DATALINK 列にファイルがリンクされた場合、そのファイルはこのグループによって所有されるように変更されます。

DLFMXGRP グループは、WRITE PERMISSION ADMIN によって定義された DATALINK 列にリンクされたファイルに関するインプレース更新機能で使用するために作成されます。DLFMGRP グループも作成されます。それらのグループには、ユーザー ID を追加しないでください。DLFM の所有するファイルを作成するユーザー ID は、DLFM だけでなければなりません。

4. Data Links 管理者のホーム・ディレクトリーとして /home/dlfm ディレクトリーを使用して、DB2 Data Links 管理者のユーザー名を作成します (たとえば dlfm)。

```
mkuser pgrp='dlfmgrp' groups='dlfmgrp,dlfmxgrp' home='/home/dlfm' dlfm
```

5. **passwd user** コマンドを入力して、このユーザー名にパスワードを割り当てます (**user** は作成したユーザー名)。

DB2 Data Links Manager 管理者のユーザー ID は、データ・リンク・ファイル・マネージャーを管理するためだけでなく、DLFM_DB およびリンク状態のファイルを所有するためのものでもあります。DB2 Data Links Manager 管理者のユーザー ID は、ファイル・システム上にデータ・リンク・ファイル・システム・フィルター (DLFF) によって制御されるディレクトリーを所有するべきではありません。DB2 Data Links Manager 管理者のユーザー ID は、データ・リンク・ファイル・マネージャー (DLFM) を管理するためにのみ使用されるべきです。DB2 Data Links Manager 管理者のユーザー ID をファイルの所有者にすることは可能ですが、それは READ PERMISSION DB のリンク先ファイルについてだけです。DB2 Data Links Manager 管理者のユーザー ID を所有者とするファイルを手動で作成しないようにしてください。

デフォルト値を変更して別のユーザー名を作成する場合は、必ず、指定するユーザー名が 8 文字以下であることを確認してください。

ホスト名の決定

DB2 および Data Links サーバーのそれぞれの名前を決定する必要があります。インストールを検証するには、これらのホスト名を知っていなければなりません。データ・リンク・ファイル・マネージャー (DLFM) に接続する際、DB2 UDB サーバーは内部で以下の情報を DLFM に送信します。

- データベース名
- インスタンス名
- ホスト名

次に DLFM は、この情報と内部の表とを比較して、接続が可能かどうかを判断します。接続が可能なのは、データベース名、インスタンス名、およびホスト名のこの組み合わせが、`dlfm add_db` コマンドを使用して登録されている場合だけです。`dlfm add_db` コマンドで使用されるホスト名は、DB2 UDB サーバーによって内部的に送信されるホスト名と正確に一致していなければなりません。

以下のようにして得られる正確なホスト名を使用します。

1. DB2 サーバーで **hostname** コマンドを入力します。たとえば、このコマンドから *db2server* が戻されます。
2. **host** *db2server* のコマンドを入力します (*db2server* は前のステップで得た名前)。このコマンドによって、以下のような出力が戻されます。

```
db2server.services.com is 9.11.302.341, Aliases: db2server
```

`dlfm add_db` コマンドを使用して DB2 UDB データベースを登録するには、ホスト名として `db2server.services.com` を使用します。`dlfm add_db` コマンドにおいてその他の別名が使用されていると、DB2 サーバーと DLFM の内部接続は失敗します。

Data Links サーバーを DB2 データベースに登録するには、DB2 の **add datalinks manager for database** *database_alias* **using node** *hostname* **port** *port_number* というコマンドを使用します。

hostname は、Data Links サーバーの名前です。このコマンドでは、Data Links サーバーの有効な別名を使用することもできます。この Data Links サーバーへの参照である DATALINK 値では、URL 値でホスト名を指定する必要があります。つまり、DATALINK 列に URL 値を代入する際には、**add datalinks manager** コマンドで使用された正確な名前を使用する必要があります。異なる別名を使用すると、SQL ステートメントが失敗します。

Tivoli Space Manager と Data Links の統合

DB2 Data Links Manager では、Tivoli Space Manager の機能を利用できます。Tivoli Space Manager Hierarchical Storage Manager (HSM) クライアント・プログラムは、該当するファイルを 2 次ストレージに自動的に移行することによって、ローカル・ファイル・システム上に特定のレベルのフリー・スペースを常に確保します。移行したファイルがアクセスされると、それらのファイルは自動的に再呼び出しされ、それによってユーザー ID が特定のファイルを移行および再呼び出しできるようになります。

この機能の前提条件は、Tivoli Space Manager バージョン 4.2 です。

定期的に 2 次ストレージに移すことの必要な大規模なファイルを含んでいるため、スペースを定期的に管理することの必要なファイル・システムを使用しているカスタマーにとって、この機能は非常に便利です。多くのカスタマーにとって、Tivoli Space Manager は、2 次ストレージを管理するための手段を提供します。DB2 Data Links Manager による Tivoli Space Manager のサポートにより、DATALINK ファイルのためのスペースを柔軟に管理することができます。Tivoli Space Manager は、格納する可能性のあるすべてのファイルのために十分なストレージを DB2 Data Links Manager ファイル・システムの中に前もって確保する代わりに、一定の期間にわたって Data Links の管理するファ

イル・システムの割り振りを調整できるようにして、通常の使用でファイル・システムが不注意にスペース不足になることがないようにします。

Data Links Replication

Data Links Manager バージョン 8.1 では、DLFM_START_ASNCOPYD および DLFM_ASNCOPYD_PORT という 2 つの新しいレジストリー変数が導入されました。インストール中に、Data Links Manager Replication デーモンを有効にするように促されます。Data Links Replication を使用する予定であれば、インストール中に Replication デーモンを有効にしてください。それは、レジストリー変数 (DLFM_START_ASNCOPYD および DLFM_ASNCOPYD_PORT) を設定して DLFM を再開することによって、インストール後にも有効にできます。

これらの考慮事項に加えて、DB2 Data Links Manager ソリューションで使用するファイル・システム (JFS または DCE-DFS) に応じて、インストールに関する考慮事項があります。AIX システムで DB2 Data Links Manager のインストールを開始する前に、これらの点についても考慮する必要があります。

関連概念:

- *Data Links Manager 概説*およびインストールの『DB2 セットアップ・ウィザードによって実行されるアクション (AIX)』
- *Data Links Manager 概説*およびインストールの『DCE-DFS 環境でのインストールの注意点 (AIX)』
- *Data Links Manager 概説*およびインストールの『JFS 環境でのインストールの注意点 (AIX)』
- 396 ページの『DB2 サーバーのインストールの概要 (Windows)』
- *DB2 Data Links Manager 管理ガイド*およびリファレンスの『Data Links Manager のセキュリティの概要』

関連タスク:

- *Data Links Manager 概説*およびインストールの『DB2 Data Links Manager のインストールの検証 (AIX)』
- *Data Links Manager 概説*およびインストールの『db2_install コマンドによる DB2 Data Links の手動インストール (AIX)』
- *Data Links Manager 概説*およびインストールの『DB2 セットアップ・ウィザードによる DB2 Data Links Manager のインストール (AIX)』
- 394 ページの『データベースの移行』
- *DB2 Universal Database サーバー機能 概説*およびインストールの『DB2 の移行 (UNIX)』
- *DB2 Data Links Manager 管理ガイド*およびリファレンスの『Data Links Manager レプリケーション・デーモンの使用可能化』

DB2 Data Links Manager をインストールする前に (Windows)

DB2® Data Links Manager をインストールする前に、まずこの情報をお読みください。

システム要件

- **Windows® NT。** Service Pack 6 以降が適用された Windows NT® バージョン 4.0 を実行している必要があります。
- **Windows 2000。** Windows 2000 (5.00.2195) (Service Pack 1 以降、RC 1.1) を実行している必要があります。

オペレーティング・システムのレベルを確認するには、「マイ コンピュータ」をクリックして、メニュー・バーから「ヘルプ」→「バージョン情報」の順に選択します。

NTFS フォーマット済みドライブ

データ・リンク・ファイル・システム・フィルター (DLFF) が制御する、Windows のファイル・システム (NTFS) でフォーマットされたドライブが 1 つ以上必要です。DLFF が制御するドライブは、NTFS でフォーマットされている必要があります。インストール時に、DLFF が制御するドライブを選択することができます。NTFS でフォーマットされたドライブがない場合、インストールを実行することはできません。

リンク先のファイルのバックアップをローカル・ファイル・システムに保管する場合は、さらに別の NTFS ドライブが必要です。

システム上にある NTFS でフォーマットされたドライブのリストを表示するには、

Windows NT

「スタート」をクリックしてから、「プログラム」→「管理ツール (共通)」→「ディスク アドミニストレータ」を選択します。

Windows 2000

「スタート」をクリックし、「設定」→「コントロール パネル」を選択します。「管理ツール」をクリックします。「コンピュータの管理」→「ディスクの管理」をクリックします。

既存のファイル・システムを NTFS に変換することもできますし、新たに作成することもできます。既存のファイル・システムを NTFS に変換する場合、以下のコマンドを入力します。

```
convert x: /fs:ntfs
```

ここで、x: は NTFS に変換するドライブです。

このコマンドは、コマンド・プロンプトから、NTFS に変換するドライブとは別のドライブで実行する必要があります。

NTFS パーティションは、Windows NT ディスク・アドミニストレータ・ツール、または Windows 2000 のディスク管理を使用して作成できます。

以前のバージョンからの移行

ご使用のシステムに 8.1 より前のバージョンの Data Links インスタンスがある場合、**db2imigr** コマンドを使用して、そのインスタンスをバージョン 8.1 形式に移行する必要があります。

レジストリー変数には、環境変数よりも高い機能と柔軟性が備わっていますが、逆にこのことが原因となって、移行が期待どおりに実行されない場合があります。移行後には、**db2set** コマンドを使用してレジストリー変数をチェックする必要があります。

Data Links サーバーのセキュリティ

DB2 Data Links バージョン 8.1 より前のリリースには、ファイルのリンクを制御するセキュリティ機能はありませんでした。旧リリースから移行した場合、DLFM 移行時に、既存の DLFM サーバーに対してセキュリティ制御を有効にするオプションが提供されます。

DB2 Data Links バージョン 6.1、7.1、または 7.2 から、DB2 Data Links Manager バージョン 8 への移行

DB2 Data Links バージョン 6.1、7.1、または 7.2 から、DB2 Data Links Manager バージョン 8.1 に移行するには、

1. DB2 および DLFM データベースで移行の準備をします。たとえば、既存のデータベースをバックアップするとよいでしょう。
2. DB2 サーバーおよび Data Links Manager サーバー・マシンに DB2 バージョン 8 をインストールします。
3. Data Links Manager の管理者として、**db2dlmmg** コマンドを実行して、既存の `dlfm_db` データベース・スキーマを新しいバージョンに移行します。

DB2 Data Links Manager バージョン 8 に移行すると、以下の環境変数が DB2 レジストリー変数に変換されます。

```
DLFM_INSTALL_PATH
DLFM_PORT
DLFM_LOG_LEVEL
DB2_RR_TO_RS
DLFM_BACKUP_DIR_NAME (1)
DLFM_BACKUP_TARGET (2)
DLFM_BACKUP_TARGET_LIBRARY (3)
```

注:

1. この変数は、ローカル・ファイル・システムがバックアップのターゲットである場合のみ使用されます。
2. この変数は、使用されるバックアップ・ターゲットのタイプを示します。この変数の値には、LOCAL、TSM、または XBSA があります。

3. この変数は、DLFM_BACKUP_TARGET が XBSA に設定されている場合のみ、XBSA のサポート・コード・ライブラリーを示します。サポート・コード・ライブラリーは、共有オブジェクト名を含む完全修飾パス名で示されます。共有オブジェクトの名前は、XBSA 準拠の共有ライブラリーを提供しているベンダーから入手できます。

DLFM データベース移行ユーティリティ

DLFM データベース移行ユーティリティ **db2dlmmg** は、既存の DLFM データベースを DB2 バージョン 6.1、バージョン 7.1、またはバージョン 7.2 フォーマットから、新規 DB2 バージョン 8.1 フォーマットに変換します。このユーティリティを実行する必要がある DB2 データベースのみが、DLFM サーバーに常駐するものです。このデータベースは DLFM_DB と呼ばれ、デフォルトでは DLFM という Data Links Manager 管理者に所有される DB2 インスタンスに属しています。このデータベースの各オカレンスは、DLFM サーバーごとに、**db2dlmmg** ユーティリティを使用して移行されなければなりません。

db2dlmmg ユーティリティは、x:\sqllib\bin ディレクトリーにインストールされます。x: は Data Links Manager がインストールされているドライブです。

db2dlmmg ユーティリティを実行する前に、DB2 バージョン 8.1 Data Links Manager がインストールされている必要があります。DLFM_DB データベースに対して、古いリリースの **db2dlmmg** ユーティリティを実行しようとしてはいけません。また、DB2 インスタンス移行 (**db2imigr**) ユーティリティを、DLFM データベース・インスタンスに対して実行済みでなければなりません。

DLFM データベース移行ユーティリティは、以下の基本的なステップにしたがって実行されます。

1. DLFM が実行中の場合は、停止します。
2. バージョン 8.1 Data Links Manager リンク・ファイル・セキュリティー管理をセットアップする方法を要求します。
3. 現行の DLFM_DB の内容が、有効なバージョン 8.1 以前のフォーマットかどうかを検証します。
4. 現行の DLFM_DB データベースが自動的にバックアップされます。
5. このデータベースに予約されているログ・スペースの量を増やします。
6. 新規のバッファ・プールおよび表スペースを作成します。
7. 表および索引を作成、変更します。
8. リンク・ファイルの権限を変更し、必要に応じて DLFM_DB 内の記述子情報を更新します。
9. 最も大きい DLFM_DB 表を新規表スペースのうちの 1 つに移動します。

10. リンク・ファイル・セキュリティー管理を、ステップ 2 に指定されたようにセットアップします。
11. DLFM 実行可能ファイルを、変更されたデータベースに再バインドします。

移行ユーティリティーが処理中にエラーを検出したときは、エラー・メッセージおよび続行する方法の説明が示されます。問題を訂正して、**db2dlmmg** を再実行できる場合もあります。しかし多くの場合、まず初めに元の DLFM_DB (上のステップ 4 でバックアップしたもの) をリストアし、**db2dlmmg** を再実行するように指示されます。ステップ 4 のバックアップは、DLFM 構成変数 DLFM_BACKUP_TARGET および DLFM_BACKUP_DIR_NAME で指定されたロケーションに保管されています。元のデータベースをリストアするには、以下のように、DB2 リストア・ユーティリティーを実行します。

- db2start
 - db2 restore database dlfm_db from <backup-directory>[taken at <date-time>] without rolling forward
- <backup-directory> は、DLFM_BACKUP_DIR_NAME で指定された完全修飾パスを示し、<backup-directory> の下に複数のバックアップ・イメージがある場合は、taken at <date-time> を指定しなくてはなりません。

構成変数 DLFM_BACKUP_TARGET および DLFM_BACKUP_DIR_NAME の値を表示するには、以下のコマンドを入力します。

- db2set DLFM_BACKUP_TARGET
- db2set DLFM_BACKUP_DIR_NAME

DLFM_BACKUP_TARGET 変数の値が、「TSM」または「XBSA」の場合、上に示した必要なリストア・コマンドは異なります。TSM または XBSA アーカイブ・サーバーのバックアップへアクセスする RESTORE ユーティリティーの説明を、参照してください。

DLFM データベース移行ユーティリティーを実行するには、以下のステップを各 DLFM サーバーで実行します。

1. Data Links Manager 管理者 ID を使用して、DLFM サーバーにログインします。
2. DLFM_BACKUP_DIR_NAME 構成変数が指しているファイル・システムに、十分なフリー・スペースがあることを確認してください。少なくとも、DLFM_DB データベースの全バックアップに十分なスペースが必要です。(少なくとも 2 つの DLFM_DB バックアップ分は必要です。詳しくは下のステップ 7 を参照してください。)
3. DLFM インスタンス・ファイル・システムに、増加するトランザクション・ログ・ファイルに十分なフリー・スペースがあることを確認してください。

い。LOGPRIMARY 構成変数は、6 まで増加します。(それ以前が 6 以下の場合) LOGFILSIZ 構成変数は 2000 まで増加します (それ以前が 2000 以下の場合)。これらの変数の現行設定値を表示するには、**db2 get db cfg for dlfm_db** コマンドを入力します。

4. DLFM インスタンス・ファイル・システムに、作業ファイル (**db2dlmmg** によって使用) に十分な追加フリー・スペースがあるかどうかを確認してください。現在 DLFM_DB によって使用されているスペースの量を、この追加のスペースに必要な量の見積もりとして使用できます。
5. **db2dlmmg** コマンドを入力して、移行ユーティリティを実行します。
6. 移行ユーティリティが完了するまで中断することなく実行するようにしてください。DLFM_DB のサイズによって、数分から 1 時間ほどかかる場合があります。移行の進行は、さまざまな状況メッセージで、画面に報告されます。エラーが発生した場合は、エラー・メッセージの説明にしたがって処置を行い、その後で **db2dlmmg** コマンドを再実行してください。
7. 一度移行ユーティリティが正常に完了したら、バージョン 8.1 で、DLFM サーバーのフル・リカバリーのために、手動で DLFM_DB のフル・バックアップを行う必要があります (移行ユーティリティは、これは自動的にはいけません。)

注意すべきいくつかの点:

- 一度上の手順を完了したら、バージョン 8.1 DLFM 操作をすべて実行できるようにします。 *dlfm start* コマンドを実行して、バージョン 8.1 Data Links File Manager が、新規の DLFM_DB で正常に開始されたかどうかを確認します。DLFM の実行を確認するには、タスク・マネージャーの DLFM プロセスをチェックできます。
- Data Links Manager インストール・ドキュメンテーションにある、インストールの妥当性検査の説明に従うか、または既存の Data Links アプリケーションを使用して Data Links Manager の正しい操作を検証する必要があります。
- 移行の前に、DLFM_DB のいかなる バックアップも、リストアしようとしなくてください。バージョン 8.1 以前のバックアップは、バージョン 8.1 Data Links Manager では使用できません。
- 移行で問題がある場合は、IBM[®] サービスに連絡してください。IBM サービスの援助なしに、DLFM_DB の内容を手動で更新しようとは、決してしないでください。

ドライブを共用可能にする

データ・リンク・ファイル・システム・フィルター (DLFF) が制御する各ドライブは、共有可能でなければなりません。

ドライブを共有可能にするには、以下のようになります。

1. 「スタート」をクリックし、「プログラム」→「Windows エクスプローラ」を選択します。
2. 共有にしたいドライブを選択して右マウス・ボタン・クリックし、「共有」を選択します。DLFF が制御するドライブをすでに共有にしている場合、ステップ 5 に進んでください。
3. 「共有する」を選択します。
4. 「新規共有」をクリックします。
5. 「共有名」フィールドに、このドライブの共有名を入力し、「OK」をクリックします。
6. 「アクセス権」をクリックします。
7. 「全員」オプションを選択します。
8. 「アクセス権の種類」をクリックし、「フル コントロール」を選択します。
9. 「OK」をクリックして新しい共有名を登録します。

メモリー要件

システムに使用できる RAM が最低 64 MB が必要です。システムに使用できるメモリーの量を確認するには、「マイ コンピュータ」アイコンを右マウス・ボタンでクリックして、「プロパティ」オプションを選択します。

システム・クロックの同期化

Data Links サーバーと DB2 サーバーでは、READ PERMISSION DB オプションによって、DATALINKS 列のリンク先ファイルのために、システム・クロックが常に同期化されている必要があります。クロックの同期化は、Data Links でトークンの有効期限を正しく機能させる上で不可欠です。トークン有効期限は、選択した DATALINK 値 (URL と埋め込みファイル許可トークンで構成される) を使用できる期間を制御するデータベース構成パラメーターです。

マシンのシステム・クロックの時刻を設定するには、以下のようにします。

1. 「スタート」をクリックし、「設定」→「コントロール パネル」を選択します。
2. 「日付と時刻」アイコンをダブルクリックします。
3. 「時刻」ボックスにあるスピン・ボタンを使用して、ローカル・システム時刻を設定します。この時刻は、必ず DB2 サーバーのローカル・システム時刻に合わせて設定してください。
4. 「タイム ゾーン」タブをクリックして、ドロップダウン・ボックスから適切なタイム・ゾーンを選択します。すべてのサーバーで同じタイム・ゾーン設定を使用してください。
5. 「OK」をクリックします。

DB2 Data Links と DB2 UDB のバージョン・レベル

DB2 Data Links と DB2 Universal Database™ は、バージョン 6.1、7.1、7.2、および 8.1 の任意の組み合わせで使用できます。たとえば、バージョン 6.1 の DB2 と、バージョン 8.1 の Data Links Manager を組み合わせることができません。

ワークステーション上の DB2 のバージョンを確認するには、**db2level** コマンドを入力します。

バージョン 8.1 など、もっと後のバージョンの DB2 でバージョン 6 または 7 の Data Links Manager に接続した場合、バージョン 8.1 から導入された Data Links Manager の機能は利用できません。バージョン 8 の機能を使用しようとする、SQL エラーになります。

ユーザー・アカウント権

データ・リンク・ファイル・マネージャーのインストールを実行するには、下記に示す特権を付与されたユーザーとしてログインする必要があります。

1. ローカル・マシンで定義されている
2. *Local Administrators* (ローカル管理者) グループに属する
3. 下記に示す拡張ユーザー権限を付与されている
 - オペレーティング・システムの一部として機能
 - クォータの増加
 - プロセス・レベル・トークンの置き換え
 - トークン・オブジェクトの作成
 - サービスとしてログオン

Windows ユーザー権限の詳細については、Windows オンライン・ヘルプを参照してください。

dldadmin ユーザー名

インストール時に、DB2 Data Links Manager の管理者として使用されるユーザー・アカウントを指定するように求められます。

デフォルトでは、セットアップ・プログラムが、ユーザー名 *dldadmin* およびパスワード *dldadmin* でユーザー・アカウントを設定します。これらのデフォルト値を使用するか、既存のアカウントを指定するか、またはデフォルト値を変更して別のユーザー・アカウントを作成するかを選ぶことができます。このユーザー・アカウントは、DLFM を Windows のサービスとして実行するためにも使用されます。

デフォルトの *dldadmin* ユーザー・アカウントを使用する場合、「パスワード」フィールドと「パスワードの確認入力」フィールドを必ず変更する必要があります。すべての DB2 Data Links Manager のインストール時に使用する *dldadmin* ユーザー・アカウントのパスワードは *dldadmin* であり、このパスワ

ードは公開されています。このユーザー・アカウントのデフォルト設定を使用すると、ご使用のネットワークのセキュリティーが脅かされる可能性があります。

ご使用のシステムに *dlmadmin* ユーザー・アカウントがすでに設定されている場合、そのユーザー・アカウントに対して以前に設定されたパスワードを使用する必要があります。

既存のユーザー・アカウントを指定する場合、そのアカウントは以下の条件を満たしていなければなりません。

- ローカル・マシンで定義されている。
- *Local Administrators* グループに属している。
- 「ファイルとディレクトリのバックアップ」 および 「ファイルとディレクトリのリストア」 ユーザー権限を持っている。
- 以下の拡張ユーザー権限を持っている。
 - オペレーティング・システムの一部として機能
 - クォータの増加
 - プロセス・レベル・トークンの置き換え
 - トークン・オブジェクトの作成
 - サービスとしてログオン
- ユーザー名が 30 文字以下である。

セットアップ・プログラムを使用して新しいユーザー・アカウントを作成する場合、指定するユーザー名は必ず 30 文字以下でなければなりません。

Administrator (管理者) グループの特権

Windows において、Data Links を使用してリンクしたファイルに関する *dlmadmin* ユーザー・アカウントの特権は、多くの機能の点で UNIX® の root (ルート) ユーザーと同じです。

	名前変更	トークンなしの ファイル・ アクセス	削除	更新
UNIX (root)	Yes	Yes	Yes	Yes
Windows (<i>dlmadmin</i>)	Yes	Yes	No*	No*

注: * NTFS において、読み取り専用ファイルに対してこれらの操作を実行することはできません。 *dlmadmin* ユーザー・アカウントの場合は、そのファイルの書き込みアクセス権を得ることによってこれらの操作を正常に実行できます。

DLFM ユーザー・アカウント

DB2 Data Links Manager 管理者のユーザー・アカウント (*dlmadmin*) に加え

て、インストールの際には、データ・リンク・ファイル・マネージャーで使用する DLFM ユーザー・アカウントも作成されます。DLFM ユーザー・アカウントは、すべての READ PERMISSION DB ファイルの所有者となります。

DLFMXGRP グループは、WRITE PERMISSION ADMIN によって定義された DATALINK 列にリンクされたファイルに関するインプレース更新機能で使用するために作成されます。それらのグループには、その他のユーザーを追加しないようにしてください。それらのグループのユーザーは DLFM ユーザーだけでなければなりません。DLFM ユーザー・アカウントは、DLFF 制御ファイル・システム内のファイルを作成するためには使用しないでください。これは READ PERMISSION DB 列の参照のためのリンク先ファイルとして誤って解釈される可能性があります。

TCP/IP ポート番号

データ・リンク・ファイル・マネージャーには、DB2 サーバーとの通信のために TCP/IP ポートが 1 つ必要です。デフォルトでは、セットアップ・プログラムが値を生成しますが、その値を使用するか、または独自の値を指定できません。インストールを検証するには、このポート番号を知っていなければなりません。

マシンで使用されている TCP/IP ポートを参照するには、services ファイルをオープンします。services ファイルは、x:¥winnt¥system32¥drivers¥etc ディレクトリーにあります (x: は Windows のインストール先ドライブ)。

ホスト名の決定

DB2 サーバーおよび Data Links サーバーのそれぞれの名前を決定する必要があります。インストールを検証するには、これらのホスト名を知っていなければなりません。データ・リンク・ファイル・マネージャー (DLFM) に接続する際、DB2 UDB サーバーは内部で以下の情報を DLFM に送信します。

- データベース名
- インスタンス名
- ホスト名

次に DLFM は、この情報と内部の表とを比較して、接続が可能かどうかを判断します。接続が可能なのは、データベース名、インスタンス名、およびホスト名のこの組み合わせが、**dlfm add_db** コマンドを使用して登録されている場合だけです。**dlfm add_db** コマンドで使用されるホスト名は、DB2 UDB サーバーによって内部的に送信されるホスト名と正確に一致していなければなりません。

これらのホスト名を取得するには、DB2 Data Links サーバーと DB2 UDB サーバーで **hostname** コマンドを入力します。このコマンドを実行すると、*dlmsvrer* のような出力が Data Links サーバーに返されます。

次に、**nslookup** *dlmserver* コマンドを入力します。ここで、*dlmserver* は、使用されているホスト名です。このコマンドによって、以下のような出力が戻されます。

```
Server: dnsserv.ibm.com
Address: 9.21.14.135
Name: dlmserver.ibm.com
Address: 9.21.51.178
```

Name: dlmserver.ibm.com エントリーは、実際のホスト名です。

各 Data Links Manager および DB2 UDB サーバーに対してこれらのステップを繰り返します。

Data Links Replication

Data Links Manager バージョン 8.1 では、DLFM_START_ASCOPYD および DLFM_ASCOPYD_PORT という 2 つの新しいレジストリー変数が導入されました。インストール中に、Data Links Manager Replication デーモンを有効にするように促されます。Data Links Replication を使用する予定であれば、インストール中に Replication デーモンを有効にしてください。それは、レジストリー変数 (DLFM_START_ASCOPYD および DLFM_ASCOPYD_PORT) を設定して DLFM を再開することによって、インストール後にも有効にできます。

関連概念:

- *Data Links Manager* 概説およびインストールの『DB2 Data Links Manager』
- *Data Links Manager* 概説およびインストールの『DB2 セットアップ・ウィザードによって実行されるアクション (Windows)』
- 396 ページの『DB2 サーバーのインストールの概要 (Windows)』
- *DB2 Data Links Manager* 管理ガイドおよびリファレンスの『Data Links Manager のセキュリティの概要』

関連タスク:

- *Data Links Manager* 概説およびインストールの『DB2 Data Links Manager のインストール (Windows)』
- 394 ページの『データベースの移行』
- *DB2 Universal Database* サーバー機能 概説およびインストールの『DB2 の移行 (Windows)』
- *DB2 Data Links Manager* 管理ガイドおよびリファレンスの『Data Links Manager レプリケーション・デーモンの使用可能化』

V8.1 の DATALINK 列に対するレジストリー変数のサイズ

V8.1 で、少なくとも 1 つの DATALINK 列を使用する DB2® データベースでは、調整 (Reconcile) およびロードのようなユーティリティでの失敗を回避するために、APP_CTL_HEAP_SZ レジストリー変数で、サイズを増やす必要があります。

APP_CTL_HEAP_SZ のデフォルトは 128 (4KB ページ) です。DATALINK 列を使用する DB2 データベースでは、これを 256 に増やすことをお勧めします。次のコマンドを下記のように指定して、この値を増加することができます。

```
db2 update db cfg for <dbname> using APP_CTL_HEAP_SZ 256
```

別の理由でこの値をすでに増やしている場合、この値をさらに 128 増やすことをお勧めします。(つまり、すでに 256 を使用している場合は、DATALINK 列のうち、設定を $256+128=384$ にさらに増やすことを意味します)。

インストールおよび構成補足

Tivoli Ready の使用可能性

"Ready for Tivoli" ロゴのある IBM® ソフトウェア・プロダクトを購入すると、IBM ソフトウェア・プロダクトを、さまざまな Tivoli® オファリングを通じて管理することができます。これにより、複数の Ready for Tivoli アプリケーションを自動的に発見、モニター、および在庫管理できます。

"Ready for Tivoli" の IBM ソフトウェア・プロダクトは、IBM Tivoli Configuration Manager などの製品を通じて管理できます。IBM Tivoli Monitoring for Database のオファリングでは、DB2、Oracle、Informix などのすべての主要なデータベースをサポートしています。

このプログラムの追加情報については、ご使用のブラウザで http://www.tivoli.com/products/tivoli_ready/ の IBM Web サイトを参照してください。

Ready for Tivoli Instrumentation および DB2 V8.1:

この DB2® 製品用の Tivoli Ready シグニチャー・ファイルは、DB2 製品と一緒にインストールされます。ファイルは xxx.sys と呼ばれ、以下のディレクトリーにインストールされます。

```
%install DB2DIR%\tivready
```

(Windows® オペレーティング・システムの場合)

```
%install DB2DIR%/tivready
```

(UNIX® オペレーティング・システムの場合)

高度なデータベース管理は、IBM Tivoli Monitoring for Database を通じて達成されます。ITM for Database 製品では、新たに高度な Distributed Monitoring 製品 (IBM Tivoli Monitoring、ITM と呼ばれる) を使用しており、この新規モニター・エンジンを使用することでモニター能力が大幅に拡張されています。ITM for Database は、Proactive Analysis Component (PAC) を通じて DB2 サポートを提供します。この PAC は DB2 と緊密に統合され、インストールしてすぐに利用可能なモニターのセットが用意されているので、迅速なデプロイメントと活動化が可能になります。カスタム・モニター、しきい値、およびタスクは、データベース管理者によっても定義することができます。

追加情報は、<http://www.tivoli.com/products/index/monitor-db/> を参照してください。

他の Tivoli 製品の情報については、<http://www.tivoli.com/products/index/> を参照してください。

ディスカバリーによるデータベース接続の構成

構成アシスタントのディスカバリー機能を使うと、ネットワークでデータベースを検索できます。

前提条件:

ディスカバリーを使用してデータベースとの接続を構成する前に、

- 有効な DB2 ユーザー ID であることを確認してください。
- DB2 サーバーまたは DB2 Connect サーバー製品がインストールされているシステムにデータベースを追加する場合は、SYSADM または SYSCTRL 権限を付与されたユーザー ID であることを確認してください。

制限:

CA のディスカバリー機能が DB2 システムについての情報を戻すためには、DB2 Administration Server (DAS) が実行されていて有効になっている必要があります。

手順:

ディスカバリーを使用してシステムにデータベースを追加するには、

1. 有効な DB2 ユーザー ID を使用してシステムにログオンします。
2. CA を開始します。CA は、「スタート」メニューから (Windows の場合)、または **db2ca** コマンドを使用することによって (Windows および UNIX システムの場合) 開始できます。
3. CA メニュー・バーの「選択済み (Selected)」で、「ウィザードを使用してデータベースを追加 (Add Database Using Wizard wizard)」を選択します。「データベースの追加 (Add Database Wizard)」ウィザードが表示されます。
4. 「ネットワークの検索 (Search the network)」ラジオ・ボタンを使用して、「次へ (Next)」をクリックします。

5. 「既知のシステム (Known Systems)」の横のフォルダーをダブルクリックします。クライアントに認識されているすべてのシステムのリストが表示されます。「その他のシステム (Other Systems)」の横のフォルダーをダブルクリックします。ネットワーク上のすべてのシステムのリストが表示されます。
6. システムの横の **[+]** 符号をクリックして、インスタンスおよびそのデータベースのリストを表示します。追加するデータベースを選択し、「次へ (Next)」ボタンをクリックします。
7. ローカル・データベース別名を「データベース別名 (Database alias)」フィールドに入力し、このデータベースについて記述するコメントを「注釈 (Comment)」フィールドに入力します。
8. ODBC を使用する計画がある場合には、このデータベースを ODBC データ・ソースとして登録します。この操作を実行するには、ODBC がインストールされていなければなりません。
9. 「完了 (Finish)」をクリックします。これで、追加したデータベースを使用できるようになります。「クローズ (Close)」をクリックして、CA を終了します。

関連タスク:

- 94 ページの『構成アシスタント (CA) によるデータベース接続の構成』
- インストールおよび構成 補足の『プロファイルによるデータベース接続の構成』
- 117 ページの『データベース接続のテスト』

CLP によるクライアントでの名前付きパイプの構成

ここでは、コマンド行プロセッサ (CLP) を使用することによって、DB2 クライアント上で名前付きパイプを構成する方法について説明します。

前提条件:

クライアントで名前付きパイプを構成する前に、

- その DB2 クライアントで名前付きパイプを使用できることを確認してください。接続を確立するためには、DB2 サーバー上でも名前付きパイプが使用可能でなければなりません。
- 下記のパラメーター値を確認してください。
 - コンピューター名 (サーバーのコンピューター名)
 - インスタンス名 (接続先となるサーバー上のインスタンスの名前)
 - ノード名 (サーバー・ノードのローカル別名)

これらのパラメーター値については、『名前付きパイプ・パラメーター値ワークシート』を参照してください。

手順:

名前付きパイプを構成するには、以下の操作を行ってください。

1. DB2 クライアント上での名前付きパイプのカタログ
2. DB2 クライアント上でのデータベースのカタログ

関連タスク:

- インストールおよび構成 補足 の『クライアントでの名前付きパイプ・ノードのカタログ』
- インストールおよび構成 補足 の『CLP によるデータベースのカタログ』

関連資料:

- インストールおよび構成 補足 の『クライアントでの名前付きパイプ構成のための名前付きパイプ・パラメーター値ワークシート』

NetBIOS 接続に関するクライアントの論理アダプター番号の判別

これは、CLP を使用したクライアントでの NetBIOS の構成 のメインタスクの一部です。

CLP を使用してクライアントとサーバーの NetBIOS 接続を構成するには、クライアントの論理アダプター番号が必要です。

制限:

この手順が必要なのは、Windows オペレーティング・システムの場合だけです。

手順:

NetBIOS 接続の論理アダプター番号を判別するには、

1. コマンド・プロンプトから **regedit** コマンドを入力してレジストリー・エディターを起動します。
2. **HKEY_LOCAL_MACHINE** フォルダーを展開し、**Software/Microsoft/Rpc/NetBIOS** フォルダーから、NetBIOS アダプターの割り当てを見つけます。
3. **ncacn_nb_nb_x** で始まる項目をダブルクリックして、NetBIOS 接続に関連したアダプター番号を表示します。ここで、*x* には 0、1、2... です (通常は **nb0** アダプターを選択します)。「**DWORD 値の編集 (Edit DWORD Value)**」ウィンドウの「**データ値 (Data value)**」の設定を記録します。

注: 接続の両端で同じエミュレーションを使用している必要があります。

CLP を使用したクライアントでの NetBIOS の構成 の次のステップは、データベース・マネージャーの構成ファイルの更新です。

関連タスク:

- 97 ページの『コマンド行プロセッサーを使用した、クライアントからサーバーへの接続の構成』

ホストおよび AS/400 アプリケーションから DB2 Universal Database サーバーへのアクセス

ホストおよび AS/400 アプリケーションは、DB2 Enterprise Server Edition に保管されている DB2 Universal Database のデータにアクセスできます。ここでは、ホストまたは AS/400 クライアントからのインバウンド・クライアント要求を受け入れるように DB2 Universal Database サーバーを構成する手順について説明します。

前提条件:

- DB2 Universal Database サーバー上に TCP/IP が構成されていなければなりません。
- ホストまたは AS/400 システム上に、下記のデータベース製品のいずれかがインストールされていて、動作していなければなりません。
 - DB2 (OS/390 版) バージョン 5 以上
 - DB2 AS/400 用 バージョン 3.1 以上
 - DB2 (VSE & VM 版) バージョン 5 以上

DB2 Universal Database サーバーにアクセスするその他の IBM 製品と IBM 以外の製品の詳細については、その製品のソフトウェア・サポート担当にお問い合わせください。

- ホストまたは AS/400 環境に必要な下記の PTF がインストールされていることが必要です。
 - DB2 (OS/390 版) バージョン 5: PQ07537
 - DB2 (VM/ESA 版) バージョン 5: VM60922; VM61072
 - OS/400 バージョン 3 リリース 2: SF23270; SF23277; SF23271; SF23721; SF23985; SF23960
- 双方向コード化文字セット ID (CCSID) サポートが有効でなければなりません。

制限:

ユーザー ID の変換は、DB2 Universal Database ではサポートされません。

手順:

ホストおよび AS/400 アプリケーションから DB2 Universal Database サーバーにアクセスするには、

1. ホストまたは AS/400 システム上にアプリケーション・リクエスターを構成します。
 - DB2 (OS/390 版) 上にアプリケーション・リクエスターを構成します。
 - DB2 AS/400 用上にアプリケーション・リクエスターを構成します。
 - DB2 (VSE & VM 版) 上にアプリケーション・リクエスターを構成します。
2. DB2 (VM 版) を使用している場合には、『DB2 Universal Database サーバーにアクセスするための DB2 (VM 版) ユーティリティの構成』で説明されている手順を実行してください。

3. 接続をテストします。 DB2 for OS/390 の場合は、 TSO にログオンして DB2I/SPUFI を使用することによって実行できます。

ホストおよび AS/400 アプリケーションからのデータ・アクセスは、 DB2 Universal Database サーバーで、DB2 データベース・マネージャー構成パラメーターを使って調整されます。 DRDA ヒープ・サイズ (DRDA_HEAP_SZ) というパラメーターは、ホストおよび AS/400 データベース・クライアントの接続専用です。 DB2 Universal Database サーバーにはさらにリソースが必要になるため、いずれかのパラメーターの設定を変更しなければならないことがあります。

UNIX ワークステーションでは、DRDA ヒープ・サイズは、ページ数単位でメモリー容量を指定します。これは、 DB2 Universal Database サーバーがホストおよび AS/400 接続用に使用するために割り振られます。

Windows の場合、DRDA ヒープ・サイズは、セグメント数単位でメモリー容量を指定します。これは、 DB2 Universal Database サーバーがホストおよび AS/400 接続用に使用するために割り振られます。

関連タスク:

- インストールおよび構成 補足 の『DB2 Universal Database サーバーにアクセスするための DB2 (VSE & VM 版) ユーティリティの構成』

関連資料:

- インストールおよび構成 補足 の『APPC を使用してホストおよび AS/400 アプリケーションから DB2 Universal Database サーバーにアクセスする際の考慮事項』
- 158 ページの『ホストまたは OS/400 から DB2 Universal Database サーバーにアクセスするためにサポートされているプロトコル』
- インストールおよび構成 補足 の『ホストおよび AS/400 アプリケーションから DB2 UDB サーバーへのアクセス例』
- インストールおよび構成 補足 の『DB2 DRDA アプリケーション・サーバー (AS) でサポートされるバインド・オプション』
- インストールおよび構成 補足 の『DB2 UDB サーバーでサポートされる DRDA 機能』

ホストまたは OS/400 から DB2 Universal Database サーバーにアクセスするためにサポートされているプロトコル

ホストおよび OS/400 アプリケーションから DB2 Universal Database サーバーへのアクセスでは、下記のプロトコルがサポートされています。

表 33. DB2 Universal Database サーバーでサポートされているプロトコル

DB2 UDB サーバー・オペレーティング・システム	サポートされているプロトコル
AIX	TCP/IP
Linux	TCP/IP
Solaris	TCP/IP
Windows NT、Windows 2000、Windows XP、および Windows .NET	TCP/IP

表 34. ホストまたは OS/400 システムでサポートされているプロトコル

ホストまたは OS/400 システム	サポートされているプロトコル
DB2 for OS/390 バージョン 5 以上	TCP/IP
DB2 for AS/400 バージョン 4 リリース 2 以上	TCP/IP
DB2 for VSE & VM	TCP/IP
DB2 for z/OS バージョン 7 以上	TCP/IP

複数サイト更新サポート (2 フェーズ・コミット) を必要とするホストまたは OS/400 アプリケーションの場合、以下のことに注意してください。

TCP/IP 接続

どのホストまたは OS/400 データベース・クライアントでも、複数サイト更新はサポートされません。リモート作業単位 (RUW) はサポートされます (1 フェーズ・コミット)。TCP/IP では、すべてのセキュリティ情報が接続試行で送られます。

関連タスク:

- 157 ページの『ホストおよび AS/400 アプリケーションから DB2 Universal Database サーバーへのアクセス』

関連資料:

- インストールおよび構成 補足の『APPC を使用してホストおよび AS/400 アプリケーションから DB2 Universal Database サーバーにアクセスする際の考慮事項』

手動インストール後の DB2 サーバーのセットアップ

ここでは、UNIX オペレーティング・システムでの手動インストール後に DB2 サーバーをセットアップする方法について説明します。db2_install スクリプトまたはオペレーティング・システム付属のインストール・ユーティリティを使用した手動インストー

ルで実行されるのは、DB2 のコンポーネントのインストールだけです。下記に示すような構成およびセットアップの作業は、手動で実行する必要があります。

手順:

手動インストール後に DB2 サーバーをセットアップするには、

1. DB2 インストールのためのグループおよびユーザー ID を作成します。
2. DB2 Administration Server (DAS) を作成します。
3. db2icrt を使用してインスタンスを作成します。db2icrt を使用してインスタンスを作成します。
4. DB2 ファイルのリンクを作成します。(オプション)
5. インスタンスのための通信をセットアップします。
 - DB2 インスタンスのための APPC 通信を構成します。
 - DB2 インスタンスのための TCP/IP 通信を構成します。
6. 製品ライセンス・キーを更新します。

タスク・センターや DB2 Administration Server スケジューラーなどの DB2 ツールを使用する場合は、DB2 ツール・カタログをセットアップすることをお勧めします。DB2 ツール・カタログには、DB2 ツールや機能スケジューラーが機能するために必要なメタデータが含まれています。

関連タスク:

- インストールおよび構成 補足 の『DB2 インストールのためのグループおよびユーザー ID の作成』
- インストールおよび構成 補足 の『DB2 Administration Server (DAS) の作成』
- インストールおよび構成 補足 の『db2icrt によるインスタンスの作成』
- インストールおよび構成 補足 の『DB2 ファイルのためのリンクの作成』
- インストールおよび構成 補足 の『DB2 インスタンスの TCP/IP 通信の構成』
- インストールおよび構成 補足 の『DB2 クライアントでの APPC 通信の構成』
- インストールおよび構成 補足 の『DB2 製品のライセンス・キーの更新』
- インストールおよび構成 補足 の『SMIT を使用して AIX に DB2 製品をインストールする』
- インストールおよび構成 補足 の『pkgadd を使用して Solaris に DB2 製品をインストールする』
- インストールおよび構成 補足 の『rpm を使用して Linux に DB2 製品をインストールする』
- インストールおよび構成 補足 の『swinstall を使用して HP-UX に DB2 製品をインストールする』
- インストールおよび構成 補足 の『db2_install スクリプトによる DB2 製品のインストール』

- 161 ページの『DB2 製品の手動インストール』
- 327 ページの『ツール・カタログ・データベースおよび DAS スケジューラーのセットアップと構成』

DB2 製品の手動インストール

DB2 製品は、対話式のインストール方法 (DB2 セットアップ・ウィザード) または応答ファイルを使用してインストールすることが勧められています。DB2 セットアップ・ウィザードには、便利なグラフィカル・インターフェースとインストールのヘルプ、ユーザーおよびグループの作成、プロトコルの構成、インスタンスの作成など、さまざまな付加的な機能が含まれています。応答ファイルによるインストールも、グラフィカル・インターフェースがないだけで、それ以外の点では同じメリットがあります。それに加えて、応答ファイルを使用する場合は、個々の DBM 構成パラメーターやプロファイル・レジストリー変数を設定するなど、高度な構成機能を利用できます。それでも、UNIX ベースのオペレーティング・システムでそれらのインストール方法を使用したくない場合は、`db2_install` スクリプトやオペレーティング・システム付属のインストール・ユーティリティーを使用して、手動で DB2 製品をインストールすることができます。

前提条件:

インストールする個々の DB2 製品については、インストールのドキュメンテーションを参照してください。たとえば、DB2 Enterprise Server Edition をインストールする場合には、*DB2 サーバー機能 概説およびインストール* のマニュアルを参照して、インストールの前提条件などの重要なセットアップ情報を確認してください。

制限:

Windows オペレーティング・システムでは、DB2 製品を手動でインストールすることはできません。Windows オペレーティング・システムの場合、DB2 セットアップ・ウィザードまたは応答ファイルを使用する以外の方法で DB2 製品をインストールすることはできません。

手順:

下記のようにして、手動インストールを選択します。

- `db2_install` スクリプトを使用して DB2 をインストールする (サポートされるいずれかのオペレーティング・システムでのインストールで使用可能)
- SMIT を使用して AIX に DB2 をインストールする
- rpm を使用して Linux に DB2 をインストールする
- `swinstall` を使用して HP-UX に DB2 をインストールする
- `pkgadd` を使用して Solaris に DB2 製品をインストールする

関連概念:

- *DB2 Universal Database* サーバー機能 概説およびインストール の『DB2 のためのインストール・メソッド』

関連タスク:

- インストールおよび構成 補足 の『db2_install スクリプトによる DB2 製品のインストール』
- インストールおよび構成 補足 の『SMIT を使用して AIX に DB2 製品をインストールする』
- インストールおよび構成 補足 の『rpm を使用して Linux に DB2 製品をインストールする』
- インストールおよび構成 補足 の『swinstall を使用して HP-UX に DB2 製品をインストールする』
- インストールおよび構成 補足 の『pkgadd を使用して Solaris に DB2 製品をインストールする』
- *Data Links Manager* 概説およびインストール の『db2_install コマンドによる DB2 Data Links の手動インストール (AIX)』
- *Data Links Manager* 概説およびインストール の『db2_install コマンドによる DB2 Data Links Manager の手動インストール (Solaris オペレーティング環境)』
- *DB2 Universal Database Personal Edition* 概説およびインストール の『「DB2 セットアップ (DB2 Setup)」ウィザードを使用した DB2 Personal Edition のインストール (Linux)』
- インストールおよび構成 補足 の『UNIX での DB2 の応答ファイル・インストール』
- *DB2 Universal Database* クライアント機能 概説およびインストール の『UNIX での DB2 クライアントのインストール』
- *DB2 Life Sciences Data Connect* 計画、インストールおよび構成のガイド の『DB2 Life Science Data Connect を AIX、HP-UX、Linux、および Solaris オペレーティング環境のサーバーにインストールする』
- *DB2 Warehouse Manager* インストール・ガイド の『DB2 Warehouse Manager Connector for SAP R/3 のインストール』
- *DB2 Warehouse Manager* インストール・ガイド の『DB2 Warehouse Manager Connector for the Web のインストール』
- *DB2 Connect Personal Edition* 概説およびインストール の『DB2 Connect Personal Edition のインストール (Linux)』
- *DB2 Connect Personal Edition* 概説およびインストール の『DB2 Connect Enterprise Edition のインストール (Solaris)』
- *DB2 Connect Personal Edition* 概説およびインストール の『DB2 Connect Enterprise Edition のインストール (Linux)』

- *DB2 Connect Personal Edition 概説およびインストール* の『DB2 Connect Enterprise Edition のインストール (HP-UX)』
- *DB2 Connect Personal Edition 概説およびインストール* の『DB2 Connect Enterprise Edition のインストール (AIX)』
- 469 ページの『Microsoft SQL Server データ・ソースにアクセスするサーバーのセットアップ』
- *Spatial Extender ユーザーズ・ガイド* の『DB2 Spatial Extender for AIX のインストール』

応答ファイルのキーワード

ここでは、分散インストールを実行する際に指定するキーワードの一部について説明します。初回インストール後も、追加のコンポーネント/製品をインストールするのに応答ファイルを使用できます。

PROD インストールする製品を指定します。オプションは以下のとおりです。

- ADMINISTRATION_CLIENT (DB2 Administration Client)
- APPLICATION_DEVELOPMENT_CLIENT (DB2 Application Development Client)
- CONNECT_APPLICATION_SERVER_EDITION (DB2 Connect Application Server Edition)
- CONNECT_ENTERPRISE_EDITION (DB2 Connect Enterprise Edition)
- CONNECT_PERSONAL_EDITION (DB2 Connect Personal Edition)
- CONNECT_UNLIMITED_EDITION (DB2 Connect Unlimited Edition)
- DATA_LINKS_MANAGER (DB2 Data Links Manager)
- DB2_HTML_DOCUMENTATION (DB2 HTML ドキュメンテーション CD)
- ENTERPRISE_SERVER_EDITION (DB2 Enterprise Server Edition)
- LIFE_SCIENCES_DATA_CONNECT (DB2 Life Sciences Data Connect)
- PERSONAL_EDITION (DB2 Personal Edition)
- RELATIONAL_CONNECT (DB2 Relational Connect)
- RUNTIME_CLIENT (DB2 Run-Time Client)
- SPATIAL_EXTENDER (DB2 Spatial Extender Server)
- WAREHOUSE_MANAGER (DB2 Data Warehouse Manager)
- WAREHOUSE_MANAGER_CONNECTORS (DB2 Data Warehouse Manager Connector)
- WORKGROUP_SERVER_EDITION (DB2 Workgroup Server Edition)

注: 正常な応答ファイルのインストールであったとしても、いくつかのコンポーネントがインストールされていない場合があるので、**PROD** キーワードはコメント化するべきではありません。

FILE DB2 製品の宛先ディレクトリーを指定します。

注: **FILE** は Windows の場合だけです。

INSTALL_TYPE

インストールのタイプを指定します。

オプションは以下のとおりです。

- COMPACT
- TYPICAL
- CUSTOM

重要: 簡略または標準のインストールでは、カスタム・キーワード (COMP) はすべて無視されます。

TYPICAL_OPTION

製品のユーザーの大半に適用される機能を含む標準インストール。標準 (TYPICAL) オプションの場合、その機能に加え、データウェアハウジング環境またはサテライト環境をインストールするユーザーにとって標準的な付加機能がインストールされます。それらのオプションは、INSTALL_TYPE キーワードが TYPICAL の場合にのみ有効です。たとえば、以下の部分から * を削除します (コメント解除)。

```
*TYPICAL_OPTION = DATA_WAREHOUSE  
*TYPICAL_OPTION = SATELLITE_ADMIN
```

COMP インストールするコンポーネントを指定します。セットアップ・プログラムは製品の必須コンポーネントを自動的にインストールし、使用できないコンポーネントが要求されたら無視します。

カスタム・インストールの場合は、コンポーネントを個々に選択する必要があります。これは、インストールするコンポーネントの COMP キーワードをコメント解除することによって可能です (製品ごとに異なります)。たとえば、CA をインストールするには、以下の部分から * を削除します (コメント解除)。

```
*COMP = CONFIGURATION_ASSISTANT
```

注: **INSTALL_TYPE** が *CUSTOM* でない場合、このキーワードは無視されません。

LANG 言語選択キーワード。言語を追加インストールする場合は、それをコメント解除してください。英語は必須であり、常に選択されます。たとえば、フランス語をインストールするには、以下の部分から * を削除します (コメント解除)。

```
*LANG=FR
```

REBOOT

インストールが完了したらシステムを再始動するかどうかを指定します。

注: **REBOOT** は Windows の場合だけです。

KILL_PROCESSES

DB2 の既存のバージョンが動作中の場合、このキーワードが YES に設定されていると、その動作中の DB2 プロセスが通知なしで終了します。

注: **KILL_PROCESSES** は Windows の場合だけです。

DB2 Administration Server の設定値

下記の DAS 設定値を有効にするには、対応する * を削除してください (コメント解除)。この設定は、以下の Windows および UNIX 環境の両方に適用されます。

- UNIX の場合:

```
*DAS_USERNAME = dasuser
*DAS_PASSWORD = dasp
*DAS_GID = 100
*DAS_UID = 100
*DAS_GROUP_NAME = dasgroup
*DAS_SMTP_SERVER = jsmith.torolab.ibm.com
```

- Windows の場合:

```
*DAS_USERNAME = dasuser
*DAS_DOMAIN = domain
*DAS_PASSWORD = dasp
*DAS_SMTP_SERVER = jsmith.torolab.ibm.com
```

下記のオプションは、DAS 連絡先リストの場所を指定します。連絡先リストがリモート位置にある場合には、システムに連絡先を追加する権限の付与されているユーザー名とパスワードを指定する必要があります。

```
*DAS_CONTACT_LIST = LOCAL または REMOTE (DEFAULT = LOCAL)
*DAS_CONTACT_LIST_HOSTNAME = hostname
*DAS_CONTACT_LIST_USERNAME = username
*DAS_CONTACT_LIST_PASSWORD = password
```

特殊なインスタンスの指定

これらのすべては、インスタンス名ではなく、インスタンス・セクションを取ります。インスタンス・セクションは、応答ファイル内になければなりません。

- Windows の場合:

- **DEFAULT_INSTANCE** - これはデフォルト・インスタンスです。
- **CTLSRV_INSTANCE** - これはサテライト・コントロール・サーバーとして動作するように構成されるインスタンスです。

- UNIX の場合:

- **WAREHOUSE_INSTANCE** - これはインストール・システムに、どのインスタンスをデータウェアハウスを使用するようにセットアップするかを示します。IWH.environment ファイルは、セクションがここに表示されるインスタンスの名前で更新されます。

インスタンスの指定

応答ファイルを使用すると、インスタンスをいくつでも必要なだけ作成できます。新しいインスタンスを作成するには、`INSTANCE` キーワードを使用してインスタンス・セクションを指定する必要があります。その後は、`INSTANCE` の中に指定されている値を含むキーワードは、すべてそのインスタンスに属するものになります。以下に示すのは、Windows および UNIX 環境の両方に対するインスタンスの指定の例です。

- UNIX の場合:

```
*INSTANCE=DB2_INSTANCE
*DB2_INSTANCE.NAME = db2inst1
*DB2_INSTANCE.TYPE = ESE
*DB2_INSTANCE.PASSWORD = PASSWORD
*DB2_INSTANCE.UID = 100
*DB2_INSTANCE.GID = 100
*DB2_INSTANCE.GROUP_NAME = db2grp1
*DB2_INSTANCE.HOME_DIRECTORY = /home/db2inst1
*DB2_INSTANCE.SVCENAME = db2cdb2inst1
*DB2_INSTANCE.PORT_NUMBER = 50000
*DB2_INSTANCE.FCM_PORT_NUMBER = 60000
*DB2_INSTANCE.MAX_LOGICAL_NODES = 4
*DB2_INSTANCE.AUTOSTART = YES
*DB2_INSTANCE.DB2COMM = TCPIP
*DB2_INSTANCE.WORDWIDTH = 32
*DB2_INSTANCE.FENCED_USERNAME = USERNAME
*DB2_INSTANCE.FENCED_PASSWORD = PASSWORD
*DB2_INSTANCE.FENCED_UID = 100
*DB2_INSTANCE.FENCED_GID = 100
*DB2_INSTANCE.FENCED_GROUP_NAME = db2grp1
*DB2_INSTANCE.FENCED_HOME_DIRECTORY = /home/db2inst1
```

- Windows の場合:

```
*INSTANCE = DB2_INSTANCE
*DB2_INSTANCE.NAME = db2inst1
*DB2_INSTANCE.TYPE = ESE
*DB2_INSTANCE.PASSWORD = PASSWORD
*DB2_INSTANCE.USERNAME = db2admin
*DB2_INSTANCE.SVCENAME = db2cdb2inst1
*DB2_INSTANCE.PORT_NUMBER = 50000
*DB2_INSTANCE.FCM_PORT_NUMBER = 60000
*DB2_INSTANCE.MAX_LOGICAL_NODES = 4
*DB2_INSTANCE.AUTOSTART = YES
*DB2_INSTANCE.DB2COMM = TCPIP, NETBIOS, NPIPE
```

DATABASE (データベース) セクション

これらのキーワードは、インストール・システムに、インストール先のマシン上でデータベースを作成またはカタログさせるために使用できます。

```
DATABASE = DATABASE_SECTION
DATABASE_SECTION.INSTANCE = DB2_INSTANCE
DATABASE_SECTION.DATABASE_NAME = TOOLSDB
DATABASE_SECTION.LOCATION = LOCAL
DATABASE_SECTION.ALIAS = TOOLSDB
```

```
DATABASE_SECTION.USERNAME = username  
DATABASE_SECTION.PASSWORD = password
```

* these keywords are only used for REMOTE databases
that are being cataloged

```
DATABASE_SECTION.SYSTEM_NAME = hostname  
DATABASE_SECTION.SVCENAME = db2cdb2inst1
```

WAREHOUSE_CONTROL_DATABASE

このキーワードの値は、応答ファイルで指定されたデータベース・セクション・キーワードの 1 つでなければなりません。たとえば、

```
*WAREHOUSE_CONTROL_DATABASE = DATABASE_SECTION
```

. このキーワードで指定されたデータベース・セクションは、 USERNAME および PASSWORD キーワードを指定する必要があります。

WAREHOUSE_SCHEMA

たとえば、ウェアハウス・スキーマを設定するには、以下の部分から * を削除します (コメント解除)。

```
*WAREHOUSE_SCHEMA = wn_schema
```

ICM_DATABASE

インフォメーション・カタログの保存に使用するデータベース。このキーワードの値は、応答ファイルで指定されたデータベース・セクション・キーワードの 1 つでなければなりません。

```
*ICM_DATABASE = DATABASE_SECTION
```

ICM_SCHEMA

たとえば、インフォメーション・カタログ・スキーマを設定するには、以下の部分から * を削除します (コメント解除)。

```
*ICM_SCHEMA = icm_schema
```

TOOLS_CATALOG_DATABASE

ツール・カタログの保存に使用するデータベース。このキーワードの値は、応答ファイルで指定されたデータベース・セクション・キーワードの 1 つでなければなりません。

```
*TOOLS_CATALOG_DATABASE = DATABASE_SECTION
```

TOOLS_CATALOG_SCHEMA

たとえば、ツール・カタログ・スキーマを設定するには、以下の部分から * を削除します (コメント解除)。

```
*TOOLS_CATALOG_SCHEMA = toolscat_schema
```

SATELITE_CONTROL_DATABASE

ここには、サテライト・コントロール・サーバーとして使用するデータベースを指定します。このキーワードの値は、応答ファイルで指定されたデータベース・セクション・キーワードの 1 つでなければなりません。

*SATELITE_CONTROL_DATABASE = DATABASE_SECTION

CONTACT (連絡先) セクション

これは、インストール・プロセスによって作成される連絡先セクションを (それが存在しない場合に) 定義し、指定されたインスタンスのヘルス通知がこの連絡先に送信されることを定義します。

```
CONTACT = contact_section
contact_section.CONTACT_NAME = contact name
contact_section.INSTANCE = DB2_INSTANCE
contact_section.EMAIL = Email address
contact_section.PAGER = NO
```

関連概念:

- インストールおよび構成 補足 の『応答ファイル』

関連資料:

- インストールおよび構成 補足 の『使用可能なサンプル応答ファイル』

マシンにインストールされている HTML ドキュメンテーションをアップデートする

DB2 HTML ドキュメンテーション *CD* からインストールした *HTML* ドキュメンテーションの更新版が *IBM* から提供されるときに、それらをアップデートできるようになりました。

注: このアップデートでは、*DB2* コードの更新は行われません。 *DB2 HTML* ドキュメンテーション *CD* からインストールした *HTML* ドキュメンテーションだけが更新されます。

手順:

手操作でドキュメンテーション・フィックスパックをダウンロードして適用するには、以下のようにします。

1. ご使用のマシンがインターネットに接続されていることを確認します。
2. Web ブラウザーで *DB2* サポート・ページ (www.ibm.com/software/data/db2/udb/winos2unix/support) を開きます。
3. バージョン 8 に関するリンクをクリックして、「ドキュメンテーション・フィックスパック」リンクを探します。
4. ドキュメンテーション・フィックスパックのレベルと、マシンにインストールされている文書のレベルを比較して、インストールされている文書のバージョンが更新されているかどうかを確認します。現在マシンにインストールされている文書のレベルは **DB2 v8.1 GA** です。

5. さらに新しいバージョンの文書がある場合、ご使用のオペレーティング・システムに対応するフィックスパックをダウンロードします。すべての Windows プラットフォーム用に 1 つのフィックスパックがあり、すべての UNIX プラットフォーム用に 1 つのフィックスパックがあります。
6. フィックスパックを適用するには、ドキュメンテーションのフィックスパック Web ページで提供される指示に従います。

関連タスク:

- *Data Links Manager 概説*およびインストールの『DB2 HTML ドキュメンテーション CD から Web サーバーへファイルをコピーする』

関連資料:

- *Data Links Manager 概説*およびインストールの『DB2 Universal Database の技術情報の概要』

WebSphere Application Server 上での DB2 Web ツールのデプロイ

ここでは、WebSphere 4.0 で DB2 Web ツール (Web コマンド・センターおよび Web ヘルプ・センター) をデプロイおよび構成する方法について説明します。これらのツールは、Web サーバー上の Web サーバーとして実行され、Web ブラウザーを介して DB2 サーバーへのアクセスを提供します。

前提条件:

WebSphere に DB2 Web ツールをインストールする場合、以下が必要です。

- IBM WebSphere Application Server 4.0 以降。IBM WebSphere 4.0 は、`prereq.properties` ファイルを変更したあとで、DB2 バージョン 8 データベースを Administration Server データベース (WAS40) として使用してインストールできません。`prereq.properties` は、IBM WebSphere 4.0 のインストールを開始する `Setup.exe` と同じパスにあり、以下のように変更する必要があります。

```
[WAS]: prereq_checker=0
```

これは、変更前は以下のようにになっています。

```
[WAS]: prereq_checker=1
```

これにより、データベース・バージョン・チェックの検査を通過できます。

- IBM DB2 Administration Client バージョン 8。
- HTML 4.0 と互換性のある Web ブラウザー。

注: DB2 Web ツールは Netscape 4.x、Internet Explorer 5.x、Opera 6.x および Eudora Web (Palm OS) でテストされました。テストされていない PDA ブラウザーを使用する場合は、参照を明示的にサーブレット構成の中に追加する必要があります。

制限:

DB2 Web ツール・デプロイメントには、以下の制約事項が適用されます。

1. DB2 Web ツールのエンタープライズ・アプリケーションで使用するために、新しい仮想およびアプリケーション・サーバーを作成するようにお勧めします。
classpath を変更することにより変更できるデフォルト・サーバーおよび default_host が WebSphere に含まれている場合、新しい仮想ホストまたはアプリケーション・サーバーの作成は必須ではありません。
2. OS/390 では、サブシステム SQL およびコンソール・コマンドはサポートされていません。
3. 複数言語および中間層 (middle tier) と DB2 サーバー間のコード・ページ変換は、サポートされていません。サーバーの言語は表示されているとおりでありますが、文字が正しく表示されない場合があります。
4. DB2 表操作のコマンドおよびスクリプトは、サポートされていません。
5. Web ヘルス・センターおよび Web コマンド・センターの同時使用は、サポートされていません。片方を使用するには、もう片方からログオフする必要があります。同じユーザー・セッションで 2 つのセンターを使用すると、アプリケーション・サーバーが不安定にある場合があります。
6. データベース、表スペース、および Web ヘルス・センター内の表スペース・コンテナのヘルス・アラートを表示するには、データベースが Web アプリケーション・サーバー上でカタログされている必要があります。
7. Web ブラウザーのボタンの使用 (「中止」、「戻る」、「履歴」) は、DB2 Web ツールの使用中はサポートされていません。
8. Netscape Navigator 4 で DB2 Web ツールを使用している場合は、ブラウザー表示が正常にリフレッシュされない場合があります。この問題が起こった場合は、ウィンドウを最小化して、再び元にもどすことで表示をリフレッシュできます。他のウィンドウにブラウザー・ウィンドウを隠し、再び前面に戻すことでも、表示をリフレッシュできます。
9. DB2 システム、インスタンス・ノード、またはデータベースに独自の別名を割り当てるには、DB2 構成アシスタントまたは DB2 コントロール・センターを使用して、アプリケーション・サーバーでそれらを明示的にカタログする必要があります。
10. DB2 Web ツールの最初の始動には、その後の始動に比べて初期化に非常に長い時間が必要です。待ち時間のほとんどは、自動カタログ処理によるものです。自動カタログ機能を使用したくない場合は、サーブレット構成でカタログをオフにすることで、待ち時間を短くすることができます。

注: サーブレット構成パラメーターは、デプロイメント・ディスクリプター・ファイル web.xml にあります。パラメーター名およびデフォルト値は、リリースによって変更される場合があります。アプリケーション・サーバーの中には、イ

インターフェースを通じて、または直接 web.xml ファイルを編集することで、これらのパラメーターの変更を許可するものがあります。

11. 出力 (結果) バッファには、1MB の絶対最大サイズがあります。これ以上に構成されていた場合でも同じです。

手順:

WebSphere Application Server に DB2 Web ツールをインストールするために、以下を行います。

1. 以下のようにして、WebSphere 管理コンソールから DB2 Web ツールのアプリケーションを準備します。
 - a. WebSphere Application Server を開始して、WebSphere 管理コンソールをオープンします。
 - b. 「**WebSphere 管理可能ドメイン (WebSphere Administrative Domain)**」をクリックし、ウィンドウの左側のペインから「**仮想ホスト (Virtual Hosts)**」をクリックして、新しい仮想ホストを作成します。
 - c. 「**仮想ホスト (Virtual Hosts)**」を選択して右クリックし、「**新規 (New)**」を選択します。
 - d. 「**名前 (Name)**」フィールドに db2tools_host を入力し、「**追加 (Add)**」をクリックします。
 - e. 「**ホスト別名 (Host Aliases)**」見出しの下に、ポート 9090 が使用可能であるという前提で、値 <*:9090> を入力するか、またはその他の使用可能な TCPIP ポートを使用します。
 - f. 「**WebSphere 管理可能ドメイン (WebSphere Administrative Domain)**」をクリックし、ウィンドウの左側のペインから「**ノード**」を拡張して、新しいアプリケーション・サーバーを作成します。任意の名前を使用できますが、セットアップ手順のあとの方でこれを再び使用するのを忘れないでください。「**ノード (Nodes)**」ディレクトリーの下に適当なサーバー名を拡張します。「**アプリケーション・サーバー (Application Servers)**」を右クリックし、「**新規 (New)**」を選択します。値を以下のようにします。
 - 1) 「**一般 (General)**」タブで、WebSphere/AppServer/bin に、インストール・ディレクトリーを使用する作業ディレクトリー の値を入力します。たとえば [...] \WebSphere \AppServer \bin がインストール・ディレクトリーの場合は、「作業ディレクトリー」フィールドに入力します。
 - 2) 「**ファイル (File)**」:
 - ¥WebSphere¥AppServer/logs/DB2Tools_stdout.txt に、インストール・ディレクトリーを使用する標準出力 を入力します。
 - ¥WebSphere¥AppServer/logs/DB2Tools_stderr.txt に、インストール・ディレクトリーを使用する標準エラー を入力します。

注: そのほかのデフォルト値は、セットアップ手順で使用できます。ただし、正常にデプロイメントおよび実行が行われたあとで、必要に応じて値を変更することができます。

2. DB2 コマンド・ウィンドウから DB2 Web ツールの構成をインポートします。
 - a. 以下の DB2 Web ツールの構成ファイルから db2wa.war を抜き出します (unzip ユーティリティを使用して)。
 - importDB2WebTools.xml、および
 - importDB2WebTools.bat

そして、それらを WebSphere¥AppServer¥bin に置きます。

注: **ImportDB2WebTools.bat** は、その暗黙的なフォルダー・パスなしでアーカイブから抽出する必要があります (デフォルトでは、このファイルは web-inf ディレクトリの下にあります)。これは、WinZip GUI の「**Use folder names**」を選択解除するか、または後でファイルが \bin\web-inf から \bin に移動されることを確認することで、解冻できます。

- b. 「スタート」 → 「プログラム」 → 「IBM DB2」 → 「コマンド行ツール」 → 「コマンド・ウィンドウ」 をクリックして、DB2 コマンド・ウィンドウをオープンします。ディレクトリー WebSphere¥AppServer¥bin に移動します。
 - c. 以下のコマンドを実行します。

```
importDB2WebTools.bat [server_name][application_server_name]
```

server_name はサーバーのホスト名です。 application_server_name は以前に WebSphere Administrator のコンソールで作成されたアプリケーション・サーバーの名前で管理コンソールを使用する前に、このコマンドを実行してください。

注: すべてのスクリプト・パラメーターには大文字小文字の区別があります。一致しない場合、新しいアプリケーション・サーバーが作成されます。

- d. WebSphere 管理コンソールに戻ります。
 3. 以下のようにして、 WebSphere 管理コンソールから DB2 Web ツールのエンタープライズ・アプリケーションをインストールします。
 - a. 「WebSphere 管理可能ドメイン (WebSphere Administrative Domain)」で「エンタープライズ・アプリケーション (Enterprise Applications)」を右クリックし、「エンタープライズ・アプリケーションのインストール (Install Enterprise Application)」メニュー項目を選択します。「エンタープライズ・アプリケーションのインストール・ウィザード (Install Enterprise Application Wizard)」ウィンドウが開きます。
 - b. 「スタンドアロン・モジュール (*.war, *.jar) のインストール (Install stand-alone module (*.war, *.jar))」ラジオ・ボタンを選択します。

c. `SQLLIB¥tools¥web¥db2wa.war` ファイル (つまり、Web アプリケーションの DB2 Web ツールのセット) を見つけてブラウズします。アプリケーション名およびコンテキスト・ルートには、以下の値を入力する必要があります。

- アプリケーション名: DB2 Web Tools
- コンテキスト・ルート: /db2wa

注: /db2wa は必須名です。これを指定しないと、アプリケーションが失敗します。

- d. 「仮想ホストの選択 (Select Virtual Host)」が表示されるまで「次へ (Next)」をクリックし、すでにすでに作成または判別されている仮想ホストを選択します。
- e. 「サーバーの選択 (Select Server)」が表示されるまで「次へ (Next)」をクリックし、すでにすでに作成または判別されているアプリケーションを選択します。「完了 (Finish)」をクリックします。
- f. DB2 Web ツールのインストールで使ったアプリケーション・サーバーを開始します。イベント・メッセージで、仮想ホストに選択した適切な HTTP ポートが報告されているか確認します。たとえば、

```
Transport http is listening on port 9,090.
```

注: ポートが一致していなければ、アプリケーション・サーバーを停止して、仮想ホスト・ポートを、イベント・メッセージに報告した値に変更しなければならない場合があります。

4. 以下を入力して、ブラウザーから DB2 Web ツールのエンタープライズ・アプリケーションを呼び出します。

```
http://localhost:port number/db2wa
```

ここで、localhost は新しいアプリケーション・サーバーの作成に使ったノード名、port number はアプリケーション・サーバーの開始後にイベント・メッセージで報告されている値です。

関連概念:

- インストールおよび構成 補足の『DB2 Web コマンド・センター (DB2 Web ツール)』
- インストールおよび構成 補足の『DB2 Web ヘルス・センター (DB2 Web ツール)』

関連タスク:

- 174 ページの『WebLogic アプリケーション・サーバー上での DB2 Web ツールのデプロイ』
- 176 ページの『その他のアプリケーション・サーバー上での DB2 Web ツールのデプロイ』

- ・ インストールおよび構成 補足 の『DB2 Web ツールのデバッグ』

WebLogic アプリケーション・サーバー上での DB2 Web ツールのデプロイ

ここでは、BEA WebLogic 7.0 で DB2 Web ツール (Web コマンド・センターおよび Web ヘルス・センター) をデプロイおよび構成する方法について説明します。これらのツールは、Web サーバー上の Web サーバーとして実行され、Web ブラウザーを介して DB2 サーバーへのアクセスを提供します。

前提条件:

WebSphere に DB2 Web ツールをインストールする場合、以下が必要です。

- ・ BEA WebLogic 7.0 アプリケーション・サーバー。
- ・ IBM DB2 Administration Client バージョン 8。
- ・ HTML 4.0 と互換性のある Web ブラウザー。

注: DB2 Web ツールは Netscape 4.x、Internet Explorer 5.x、Opera 6.x および Eudora Web (Palm OS) でテストされました。テストされていない PDA ブラウザーを使用する場合は、参照を明示的にサーブレット構成の中に追加する必要があります。ある場合があります。

制限:

DB2 Web ツール・デプロイメントには、以下の制約事項が適用されます。

1. OS/390 では、サブシステム SQL およびコンソール・コマンドはサポートされていません。
2. 複数言語および中間層 (middle tier) と DB2 サーバー間のコード・ページ変換は、サポートされていません。サーバーの言語は表示されているとおりで、文字が正しく表示されない場合があります。
3. DB2 表操作のコマンドおよびスクリプトは、サポートされていません。
4. Web ヘルス・センターおよび Web コマンド・センターの同時使用は、サポートされていません。片方を使用するには、もう片方からログオフする必要があります。同じユーザー・セッションで 2 つのセンターを使用すると、アプリケーション・サーバーが不安定にある場合があります。
5. データベース、表スペース、および Web ヘルス・センター内の表スペース・コンテナのヘルス・アラートを表示するには、データベースが Web アプリケーション・サーバー上でカタログされている必要があります。
6. Web ブラウザーのボタンの使用 (「中止」、「戻る」、「履歴」) は、DB2 Web ツールの使用中はサポートされていません。
7. Netscape Navigator 4 で DB2 Web ツールを使用している場合は、ブラウザー表示が正常にリフレッシュされない場合があります。この問題が起こった場合は、ウィ

ンドウを最小化して、再び元にもどすことで表示をリフレッシュできます。他のウィンドウにブラウザ・ウィンドウを隠し、再び前面に戻すことで、表示をリフレッシュできます。

8. DB2 システム、インスタンス・ノード、またはデータベースに独自の別名を割り当てるには、DB2 構成アシスタントまたは DB2 コントロール・センターを使用して、アプリケーション・サーバーでそれらを明示的にカタログする必要があります。
9. DB2 Web ツールの最初の始動には、その後の始動に比べて初期化に非常に長い時間が必要です。待ち時間のほとんどは、自動カタログ処理によるものです。自動カタログ機能を使用したくない場合は、サーブレット構成でカタログをオフにすることで、待ち時間を短くすることができます。

注: サーブレット構成パラメーターは、デプロイメント・ディスクリプター・ファイル `web.xml` にあります。パラメーター名およびデフォルト値は、リリースによって変更される場合があります。アプリケーション・サーバーの中には、インターフェースを通じて、または直接 `web.xml` ファイルを編集することで、これらのパラメーターの変更を許可するものがあります。

10. 出力 (結果) バッファには、1MB の絶対最大サイズがあります。これ以上に構成されていた場合でも同じです。

手順:

WebLogic アプリケーション・サーバーに DB2 Web ツールをインストールするために、以下を行います。

1. 以下を完了させて、WebLogic アプリケーション・サーバーへの DB2 Web ツールの JVM classpath を構成します。
 - a. WebLogic インストール・パス `weblogic700¥server¥bin` 内の **startWLS.cmd** を探します。たとえば、

```
D:¥BEA¥weblogic700¥server¥bin¥
```
 - b. 以下の行を探します。

```
set CLASSPATH=%JAVA_HOME¥lib¥tools.jar;
    ¥WL_HOME¥server¥lib¥weblogic_sp.jar;
    ¥WL_HOME¥server¥lib¥weblogic.jar;%CLASSPATH%
```
 - c. その後、以下を挿入します。

```
set DB2PATH=DB2_install_path
set CLASSPATH=%CLASSPATH%;%DB2PATH¥tools¥web¥webtools.jar;
    ¥DB2PATH¥tools¥databean.jar;%DB2PATH¥tools¥xalan.jar;
    ¥DB2PATH¥tools¥xerces.jar;%DB2PATH¥tools¥db2das.jar;
    ¥DB2PATH¥tools¥db2cmn.jar;%DB2PATH¥tools¥db2ca.jar;
    ¥DB2PATH¥tools¥db2cc.jar;%DB2PATH¥tools¥db2hcapi.jar;
    ¥DB2PATH¥tools¥db2ssmonapis.jar;%DB2PATH¥java¥Common.jar;
    ¥DB2PATH¥java¥db2java.zip;%DB2PATH¥java¥db2jcc.jar
```
2. 以下を完了させて、WebLogic 管理コンソールを介して DB2 Web ツールをデプロイします。

- a. WebLogic 管理コンソールを開始します。
 - b. ウィンドウの左側のペインで、「ドメイン (domain)」 → 「デプロイメント (deployments)」 → 「Web アプリケーション (Web Applications)」をクリックします。
 - c. 「新しい Web アプリケーションの構成 (Configure a new Web Application)」リンクをクリックして、DB2 Web ツールの Web アプリケーションをインストールします。
 - d. ファイル・システムのリストをブラウズして、`Sql1lib¥tools¥web¥db2wa.war` を見つけます。
 - e. `db2wa.war` ファイル名の横にある 「選択 (select)」 を選択します。
 - f. 選択可能なサーバーのリストから DB2 Web ツールを入れるサーバーを選択し、矢印をクリックしてターゲット・サーバーに移動させます。
- 注: DB2 Web ツールでハードコーディングされているため、オリジナルの名前である `db2wa` はそのまま使用する必要があります。
- g. 「構成およびデプロイ (Configure and Deploy)」 ボタンをクリックします。
 - h. 選択したサーバー上の Web アプリケーションのデプロイメント状況を、アプリケーション・サーバーがリフレッシュするまで待ちます。正常に行われた場合、「`Deployed=true`」が表示されます。
3. 以下の場所にある DB2 Web ツールの Web アプリケーションを呼び出します。

`http://server_name:app_server_port_number/db2wa`

たとえば、`http://server_name:7001/db2wa` などです。

関連概念:

- インストールおよび構成 補足 の『DB2 Web コマンド・センター (DB2 Web ツール)』
- インストールおよび構成 補足 の『DB2 Web ヘルス・センター (DB2 Web ツール)』

関連タスク:

- 169 ページの『WebSphere Application Server 上での DB2 Web ツールのデプロイ』
- 176 ページの『その他のアプリケーション・サーバー上での DB2 Web ツールのデプロイ』
- インストールおよび構成 補足 の『DB2 Web ツールのデバッグ』

その他のアプリケーション・サーバー上での DB2 Web ツールのデプロイ

ここでは、Tomcat 4.0 や Macromedia JRun 4.0 などのほかのアプリケーション・サーバーで DB2 Web ツール (Web コマンド・センターおよび Web ヘルス・センター) を

デプロイおよび構成する方法について説明します。これらのツールは、Web サーバー上の Web サーバーとして実行され、Web ブラウザーを介して DB2 サーバーへのアクセスを提供します。

前提条件:

DB2 Web ツールをインストールする場合、以下が必要です。

- 以下のようなアプリケーション・サーバー。
 - Tomcat 4.0 Servlet/JSP Container (<http://jakarta.apache.org/tomcat/>)
 - Macromedia JRun 4.0
- IBM DB2 Administration Client バージョン 8。
- HTML 4.0 と互換性のある Web ブラウザー。

制限:

データベース、表スペース、および Web ヘルス・センター内の表スペース・コンテナのヘルス・アラートを表示するには、データベースが Web アプリケーション・サーバー上でカタログされている必要があります。

手順:

以下に、Tomcat 4.0 や Macromedia JRun 4.0 などのアプリケーション・サーバーを使用して、DB2 Web ツールをインストールする際の手順を示します。

Tomcat 4.0

1. 以下を完了して、Tomcat 4.0 構成ファイル (CLASSPATH) を準備します。
 - a. 新しい environment/system 変数 **CATALINA_HOME** を作成して、Tomcat 4.0 へのパス (ルート・ディレクトリ) を入れます。たとえば、D:\jakarta-tomcat-4.0.3 などです。

注: このステップは Windows オペレーティング・システムでは必要ありませんが、**ステップ c** は、この値の設定か、または使用するオリジナルのパスに応じて異なります。
 - b. 以下のように、Tomcat 4.0 Servlet/JSP Container が機能しているか確認します。
 - 1) Tomcat の bin ディレクトリから **startup.bat** を実行して、Tomcat を開始します。
 - 2) Web ブラウザーを使用してメインの Web ページ **<http://localhost:8080/>** にアクセスします。
 - 3) Tomcat の bin ディレクトリから **shutdown.bat** を実行するか、または Tomcat を開始したオリジナルのコマンド・ウィンドウをクローズして、Tomcat をシャットダウンします。

- c. bin ディレクトリー内にある **setclasspath.bat** 構成ファイルを変更します。たとえば、ファイルの最後に以下を入力した場合の D:\jakarta-tomcat-4.0.3\bin などです。

```
set CLASSPATH=%CLASSPATH%; %CATALINA_HOME%\common\lib\
servlet.jar;%DB2PATH%\tools\web\webttools.jar;
%DB2PATH%\tools\xalan.jar;%DB2PATH%\tools\xerces.jar;
%DB2PATH%\tools\db2cmn.jar;%DB2PATH%\tools\das.jar;
%DB2PATH%\tools\db2ca.jar;%DB2PATH%\tools\db2cc.jar;
%DB2PATH%\tools\db2hcapi.jar;%DB2PATH%\tools\databean.jar;
%DB2PATH%\tools\db2ssmonapis.jar;%DB2PATH%\java\Common.jar;
%DB2PATH%\java\db2java.zip
```

2. DB2 Web ツールのインストール・パス (つまり、Sql1lib\tools\web\db2wa.war) を見つけて、**db2wa.war** を Tomcat のデプロイメント・ディレクトリー (つまり、Tomcat の webapps ディレクトリー) にコピーし、DB2 Web ツールを Tomcat Servlet/JSP Container にデプロイします。
3. 以下を完了して、Tomcat Servlet/JSP Container で DB2 Web ツールを呼び出します。
 - a. DB2 コマンド・ウィンドウを開いて、ディレクトリーを Tomcat の bin ディレクトリーに移動します。
 - b. **startup.bat** を使用して Tomcat を開始し、新しいディレクトリー (**db2wa**) が webapps ディレクトリーに追加されているか確認します。

注: コマンド・ウィンドウからの **startup.bat** の実行では、DB2PATH は設定されません。それを有効にするには、%DB2PATH% 環境変数ではなく DB2 インストール・パスを明示的に参照するように、CLASSPATH 行を変更する必要があります。

- c. DB2 Web ツールのエンタープライズ・アプリケーションは **http://localhost:8080/db2wa** に置かれ、HTML 4.0 対応の Web ブラウザーでアクセスできます。

JRun

1. 以下を完了して、DB2 Web ツールに新しいアプリケーション・サーバーを準備します。

注: 新しいアプリケーション・サーバーを作成するようにお勧めしますが、必須ではありません。テスト目的の場合、デフォルト・サーバーを使用でき、JVM classpath およびデプロイメントの構成だけが必須です。

- a. JRun Management Console を開始して、アプリケーション・サーバーの管理者としてログインします。
- b. メイン・ページの右側にある「**新しいサーバーの作成 (Create New Server)**」を使用して、新しいアプリケーション・サーバーを作成します。ホスト名の選択は localhost から変更しないでください。

- c. 新しいサーバー名 (**DB2WebToolsServer**) を入力して、「JRun サーバー・ディレクトリー (JRun Server Directory)」をクリックします。値が自動的に入力されます。
 - d. 「サーバーの作成 (**Create Server**)」ボタンをクリックします。
 - e. 生成された値を記録するか、または以下に新しい値を入力します。
 - 「JNDI プロバイダーの URL (JNDI Provider URL)」
 - 「Web サーバーのポート番号 (Web Server Port Number)」。 DB2 Web ツールの URL に使用される値になります (つまり、`http://localhost:web_server_port_numer/db2wa`)
 - 「Web コネクターのプロキシ・ポート番号 (Web Connector Proxy Port Number)」
 - f. 必要に応じて「ポート番号の更新 (**update port numbers**)」をクリックし、ウィンドウを閉じます。
2. 以下を完了させて、アプリケーション・サーバーへの JVM classpath を構成します。
 - a. コンテンツの表 (左側のパネル) 中の新たに作成された「**DB2WebToolsServer**」を選択し、「**設定 (Settings)**」を選択して、「**JVM 設定 (JVM Settings)**」をクリックします。
 - b. `DB2_install_path` をシステム上の明示的な値で置換してから、以下の値を含む「**新しい classpath (New Classpath)**」に新しい項目を追加します。


```
DB2_install_path%tools%web%webtools.jar;
DB2_install_path%tools%databasean.jar;
DB2_install_path%tools%xalan.jar;
DB2_install_path%tools%xerces.jar;
DB2_install_path%tools%das.jar;
DB2_install_path%tools%db2cmn.jar;
DB2_install_path%tools%db2ca.jar;
DB2_install_path%tools%db2cc.jar;
DB2_install_path%tools%db2hcapi.jar;
DB2_install_path%tools%db2ssmonapis.jar;
DB2_install_path%java%Common.jar;
DB2_install_path%java%db2java.zip
```
3. 以下を完了させて、JRun アプリケーション・サーバーでの DB2 Web ツールをデプロイします。
 - a. DB2 Web ツールの Web アプリケーションを入れるように選択したアプリケーション・サーバー (DB2WebToolsServer、デフォルトまたは admin 以外のほかの任意のもの) を開始します。
 - b. 「**Web アプリケーション (Web Applications)**」をクリックしてから、「**追加 (Add)**」をクリックします。

- c. 「**デプロイメント・ファイル (Deployment File)**」 セクションをブラウザして、DB2 インストール・パスから `Sql1lib¥tools¥web¥db2wa.war` ファイルを選択します。
 - d. 「**デプロイ (Deploy)**」 をクリックし、コンテキスト・パスが `/db2wa` であることを確認します。
 - e. アプリケーション・サーバーを選択し、「**Web アプリケーション (Web Applications)**」 セクションに DB2 Web ツール・アプリケーションが表示されているか確認します。このページの「**適用 (Apply)**」はクリックしないでください。
 - f. メイン・ページの左側のパネルから「**ホーム (Home)**」リンクを選択します。
 - g. DB2 Web ツール (**DB2WebToolsServer**) を含む、「**ホーム (Home)**」ビューからアプリケーション・サーバーを再始動します。
4. DB2 Web ツールのエンタープライズ・アプリケーションは `http://localhost:web_server_port_numer/db2wa` に置かれ、HTML 4.0 対応の Web ブラウザーでアクセスできます。

関連概念:

- インストールおよび構成 補足 の『DB2 Web コマンド・センター (DB2 Web ツール)』
- インストールおよび構成 補足 の『DB2 Web ヘルス・センター (DB2 Web ツール)』

関連タスク:

- 169 ページの『WebSphere Application Server 上での DB2 Web ツールのデプロイ』
- 174 ページの『WebLogic アプリケーション・サーバー上での DB2 Web ツールのデプロイ』
- インストールおよび構成 補足 の『DB2 Web ツールのデバッグ』

db2licm コマンドによる DB2 ライセンス・ポリシーの設定

ライセンス・ポリシーは、ライセンス・センターではなく **db2licm** コマンドを使用しても追加できます。

手順:

db2licm コマンドを使用してライセンス・ポリシーを設定するには、購入したライセンスの種類に応じて、以下のうちのいずれかを実行します

- 並行ユーザー・ライセンスを購入した場合、下記のコマンドを入力します (この例は DB2 UDB Enterprise Server Edition の場合)。


```
db2licm -p db2ese concurrent
db2licm -u N
```

N は、購入した並行ユーザー・ライセンスの数です。

- 登録済みユーザー・ライセンスを購入した場合、下記のコマンドを入力します (この例は DB2 UDB Enterprise Server Edition の場合)。

```
db2licm -p db2ese registered
```

- 並行ユーザー・ライセンスと登録済みユーザー・ライセンスの両方 を購入した場合、下記のコマンドを入力します (この例は DB2 UDB Enterprise Server Edition の場合)。

```
db2licm -p db2ese concurrent registered
```

関連タスク:

- 185 ページの『db2licm コマンドによる DB2 ライセンス・キーのインストール』
- 187 ページの『ライセンス・センターによる DB2 ライセンス・キーのインストール』
- インストールおよび構成 補足の『ライセンス・センターによる DB2 ライセンス・ポリシーの設定』

UNIX での Data Direct Technologies Driver Manager および DB2 UDB バージョン 8 ODBC ドライバー

UNIX で Data Direct Technologies Driver Manager が DB2 の ODBC ドライバーにアクセスするときに、ユニコード・サポートとの非互換性が検出されます。これらの非互換性により、アプリケーションがその使用を要求したかどうかにかかわらず、Data Direct Technologies Driver Manager でユニコードが使用されるようになります。これは、非 IBM データ・ソースのプロダクトに Data Direct Technologies Driver Manager を必要とする、データウェアハウス・センター、インフォメーション・カタログ・マネージャー、および MQSI のような製品で、問題になる可能性があります。永続的なソリューションが使用可能になるまで、ユニコード・サポートのない代替 DB2 ODBC ドライバー・ライブラリーを使用できます。

ユニコード・サポートのない代替 DB2 ODBC ドライバー・ライブラリーは、DB2 UDB バージョン 8.1 (AIX および Solaris オペレーティング環境版) に含まれています。この代替ライブラリーを使用するには、コピーを作成し、コピーに元の DB2 ODBC ドライバーのライブラリー名を付ける必要があります。

注: 代替 (_36) ライブラリーには、DB2 JDBC ドライバーに必要なユニコード機能が含まれます。このライブラリーを使用しても、WebSphere Application Server を含む JDBC アプリケーションは、DB2 で正常に作業を行うことができます。

AIX または Solaris オペレーティング環境で、非ユニコード ODBC ライブラリーに切り替えるには、以下の説明にしたがってください。これは手動のプロセスなので、製品を更新、またはフィックス・パックを適用するたびに、実行する必要があります。

手順:

AIX

必要なライブラリーを AIX 上に作成するには、以下のようになります。

1. インスタンス所有者で、**db2stop force** を使用してすべてのデータベース・インスタンスをシャットダウンします。
2. 管理インスタンス ID で、**db2admin stop force** を使用して管理サーバー・インスタンスをシャットダウンします。
3. 元の db2.o を /usr/lpp/db2_81/lib の下にバックアップします。
4. root で、**slibclean** を実行します。
5. db2_36.o を db2.o にコピーします。所有権および権限が整合したままであることを確認します。以下のコマンドを使用します。

```
cp db2_36.o db2.o
-r--r--r-- bin:bin for db2.o
```

もとのオブジェクトに戻すには、db2_36.o の代わりにバックアップ・ファイルを使用して、同じ手順にしたがいます。

Solaris オペレーティング環境

必要なライブラリーを作成するには、以下のようになります。

1. インスタンス所有者で、**db2stop force** を使用してすべてのデータベース・インスタンスをシャットダウンします。
2. 管理インスタンス ID で、**db2admin stop force** を使用して管理サーバー・インスタンスをシャットダウンします。
3. 元の libdb2.so.1 を /opt/IBMDB2/V8.1/lib の下にバックアップします。
4. libdb2_36.so.1 を libdb2.so.1 にコピーします。所有権および権限が整合したままであることを確認します。以下のコマンドを使用します。

```
cp libdb2_36.so.1 libdb2.so.1
-r-xr-xr-x bin:bin libdb2.so.1
```

5. それぞれのデータベース・インスタンスに対して **db2iupdt <instance>** を実行し、DAS インスタンスに対して **dasupdt <das_instance>** を実行します。

もとのオブジェクトに戻すには、libdb2_36.so.1 の代わりにバックアップ・ファイルを使用して、同じ手順にしたがいます。

その他の UNIX オペレーティング・システム

その他の UNIX オペレーティング・システムで、DB2 および Data Direct Technologies Driver Manager に関する援助が必要な場合は、IBM サポートに連絡してください。

応答ファイル・インストール・エラー・コード

以下の表は、応答ファイルのインストール中に検出される可能性のあるエラー戻りコード (1 次および 2 次) を表しています。

表 35. 1 次応答ファイル・インストール・エラー・コード

エラー・コード値	説明
0	アクションは正常に完了しました。
1	アクションは警告を返しました。
1603	インストール中に致命的エラーが発生しました。

表 36. 2 次応答ファイル・インストール・エラー・コード

エラー・コード値	説明
3	パスが見つかりませんでした。
5	アクセスが拒否されました。
10	環境エラーが発生しました。
13	データが無効です。
87	パラメーターのうちのいずれかが無効です。
1602	インストールはユーザーによって取り消されました。
1610	構成データが壊れています。サポート担当者に連絡してください。
1612	この製品のインストール・ソースが利用不能です。ソースが存在し、アクセス可能であることを確認してください。
1618	別のインストールが既に進行中です。このインストールを続ける前に、前のインストールを完了してください。
1622	インストール・ログ・ファイルを開いているときにエラーが発生しました。指定されたログ・ファイルのロケーションが存在し、書き込み可能かどうかを確認してください。
1632	Temp フォルダがフルか、またはアクセス不可能です。 Temp フォルダが存在し、書き込み可能かどうかを確認してください。
1633	このインストール・パッケージは、このプラットフォームではサポートされていません。
1638	この製品の別のバージョンがすでにインストールされています。このバージョンのインストールを継続できません。
1639	コマンド行引き数が無効です。

Windows Installer の戻りコードについて詳しくは、Microsoft の Web サイトを参照してください。

DB2 製品ライセンス・ファイル

DB2 製品のライセンス・ファイルの名前は、下記のとおりです。ライセンス・ファイルは、DB2 を使用する前にインストールする必要があります。

db2conee.lic

DB2 Connect Enterprise Edition

db2conpe.lic

DB2 Connect Personal Edition

db2conas.lic

DB2 Connect Application Server Edition

db2dlm.lic

DB2 Data Links Manager

db2gse.lic

DB2 Spatial Extender

db2rcon.lic

DB2 Relational Connect

db2ese.lic

DB2 Enterprise Server Edition

db2pe.lic

DB2 Personal Edition

db2wm.lic

DB2 Warehouse Manager

db2wse.lic

DB2 Workgroup Server Edition

db2conue.lic

DB2 Connect Unlimited Edition

db2nse.lic

Net Search Extender

db2wsue.lic

DB2 Workgroup Server Unlimited Edition

関連タスク:

- インストールおよび構成 補足の『DB2 製品のライセンス・キーの更新』

db2licm コマンドによるライセンス・タイプの設定

ライセンス・センターを使用する代わりに **db2licm** コマンドを使って、ライセンス・キーを設定することができます。

手順:

db2licm コマンドを使用してライセンス・タイプを設定するには、次のようにします。

DB2 Connect Enterprise Edition の場合:

- 並行ユーザーのライセンスを購入した場合、次のコマンドを入力します。

```
db2licm -p db2cneee concurrent
db2licm -u N
```

N は、購入した並行ユーザー・ライセンスの数です。

- 登録済みユーザー・ライセンスを購入した場合、次のようなコマンドを入力します。

```
db2licm -p db2cneee registered
```

関連タスク:

- 122 ページの『db2licm コマンドを使用した DB2 Connect ライセンス・キーのインストール』
- 121 ページの『ライセンス・センターを使用した DB2 Connect ライセンス・キーのインストール』
- 123 ページの『ライセンス・センターを使用したライセンス・タイプの設定』

db2licm コマンドによる DB2 ライセンス・キーのインストール

ライセンス・キーは、ライセンス・センターではなく **db2licm** コマンドを使用しても追加できます。

制限:

DB2 ライセンス・キーは、DB2 がインストールされているコンピューターごとにインストールする必要があります。

手順:

db2licm コマンドを使用してライセンス・キーを追加するには、

1. ディレクトリーを、**db2licm** コマンドがインストールされているディレクトリーに変更します。
 - Windows サーバーでは、これは `SQLLIB/bin` ディレクトリーです。
 - UNIX サーバーでは、これは `INSTALLPATH/adm` ディレクトリーです。この `INSTALLPATH` は、DB2 がインストールされているディレクトリーです。

2. Windows サーバーで、下記のコマンドを入力します。

```
db2licm -a x:%db2¥license¥license_filename
```

x: は、DB2 製品 CD が含まれている CD-ROM ドライブです。

- UNIX サーバーで、下記のコマンドを入力します。

```
db2licm -a db2/license/license_filename
```

DB2 Universal Database 製品の場合の *license_filename* は、下記のとおりです。

db2conee.lic

DB2 Connect Enterprise Edition

db2conpe.lic

DB2 Connect Personal Edition

db2conas.lic

DB2 Connect Application Server Edition

db2dlm.lic

DB2 Data Links Manager

db2gse.lic

DB2 Spatial Extender

db2rcon.lic

DB2 Relational Connect

db2ese.lic

DB2 Enterprise Server Edition

db2pe.lic

DB2 Personal Edition

db2wm.lic

DB2 Warehouse Manager

db2wse.lic

DB2 Workgroup Server Edition

db2conue.lic

DB2 Connect Unlimited Edition

db2nse.lic

Net Search Extender

db2wsue.lic

DB2 Workgroup Server Unlimited Edition

関連タスク:

- 187 ページの『ライセンス・センターによる DB2 ライセンス・キーのインストール』
- 180 ページの『db2licm コマンドによる DB2 ライセンス・ポリシーの設定』
- インストールおよび構成 補足 の『ライセンス・センターによる DB2 ライセンス・ポリシーの設定』

関連資料:

- 184 ページの『DB2 製品ライセンス・ファイル』

ライセンス・センターによる DB2 ライセンス・キーのインストール

ライセンス・キーは、ライセンス・センターを使用してインストールできます。

手順:

ライセンス・センターを使用してライセンス・キーをインストールするには、

1. DB2 コントロール・センターを開始してから、「ツール (**Tools**)」メニューから「ライセンス・センター (**License Center**)」を選択します。
2. ライセンスをインストールするシステムを選択します。「インストール済み製品 (**Installed Products**)」フィールドに、インストール済み製品の名前が表示されます。
3. 「ライセンス (**License**)」メニューから「追加 (**Add**)」を選択します。
4. 「ライセンスの追加 (Add License)」ウィンドウで、「ファイルから (**From a file**)」ラジオ・ボタンを選択し、ライセンス・ファイルを選択します。
 - Windows サーバーの場合: `x:\db2\license\license_filename` (x: は DB2 製品 CD が入っている CD-ROM ドライブ)。
 - UNIX サーバーの場合: `/db2/license/license_filename`

DB2 Universal Database 製品の場合の `license_filename` は、下記のとおりです。

db2conee.lic

DB2 Connect Enterprise Edition

db2conpe.lic

DB2 Connect Personal Edition

db2conas.lic

DB2 Connect Application Server Edition

db2d1m.lic

DB2 Data Links Manager

db2gse.lic

DB2 Spatial Extender

db2rcon.lic

DB2 Relational Connect

db2ese.lic

DB2 Enterprise Server Edition

db2pe.lic

DB2 Personal Edition

db2wm.lic

DB2 Warehouse Manager

db2wse.lic

DB2 Workgroup Server Edition

db2conue.lic

DB2 Connect Unlimited Edition

db2nse.lic

Net Search Extender

db2wsue.lic

DB2 Workgroup Server Unlimited Edition

5. 「適用 (**Apply**)」をクリックしてライセンス・キーを追加します。

関連タスク:

- 185 ページの『db2licm コマンドによる DB2 ライセンス・キーのインストール』
- 180 ページの『db2licm コマンドによる DB2 ライセンス・ポリシーの設定』
- インストールおよび構成 補足の『ライセンス・センターによる DB2 ライセンス・ポリシーの設定』

関連資料:

- 184 ページの『DB2 製品ライセンス・ファイル』

Warehouse Manager インストールの手引き

Windows NT Websphere IBM HTTP サーバーへの Web 用インフォメーション・カタログ・センターのインストール

Web 用インフォメーション・カタログ・センターは、DB2 Universal Database への JDBC 接続をサポートし、IBM WebSphere を実行する、任意のオペレーティング・システムにインストールすることができます。 Web サーバー (IBM HTTP Server など) も必要です。

前提条件:

Web 用インフォメーション・カタログ・センターを Windows NT Websphere IBM HTTP サーバーにインストールする前に、次のことを行ってください。

- IBM HTTP サーバーがインストールされていることを確認します。
- そのサーバーに IBM Websphere Application Server がインストールされていることを確認します。
- Web サーバーのポート番号をチェックします。 Web サーバーのポート番号が 80 (通常のデフォルト) 以外である場合は、ドメイン・ネームの構成時に Web アドレスのホスト名にポート番号を付加します。たとえば、http://hostname:portnumber/

手順:

Web 用インフォメーション・カタログ・センターをインストールするには、次のようにします。

1. Web サーバーに、Web 用インフォメーション・カタログ・センターのファイルをインストールします。
 - a. DB2 Universal Database CD-ROM を Web サーバー・ワークステーションの CD-ROM ドライブに挿入します。ランチパッドがオープンします。
 - b. ランチパッドから「インストール (Install)」をクリックします。
 - c. 「製品の選択 (Select Products)」ウィンドウで、「DB2 管理クライアント (DB2 Administration Client)」チェック・ボックスを選択します。それから、「次へ (Next)」をクリックします。
 - d. 「インストールのタイプの選択 (Select Type of Install)」ページで、「カスタム (Custom)」を選択します。「次へ (Next)」をクリックします。
 - e. 「DB2 コンポーネントの選択 (Select DB2 Components)」ページで、「インフォメーション・カタログ・マネージャー・ツール (Information Catalog Manager tools)」以外のすべてのコンポーネントのチェック・ボックスをクリアします。
 - f. 「サブコンポーネント (Subcomponents)」をクリックします。
 - g. 「サブコンポーネントの選択 (Select Subcomponents)」ページで、「Web 用インフォメーション・カタログ・センター (Information Catalog Center for the Web)」のチェック・ボックスが選択されており、他のすべてのチェック・ボックスがクリアされていることを確認します。
 - h. 「継続 (Continue)」をクリックします。
 - i. 「インフォメーション・カタログ・マネージャー・ツール (Information Catalog Manager tools)」が選択されていることを確認します。
 - j. 「次へ (Next)」をクリックして継続し、最後のページで「終了 (Finish)」をクリックしてインストールを完了します。
2. Web 用インフォメーション・カタログ・センターを使用する前に、『Web 用インフォメーション・カタログ・センターのインストール後の構成』に示されたステップにしたがってください。

関連概念:

- *DB2 Warehouse Manager* インストール・ガイド の『インフォメーション・カタログ・マネージャー』

関連タスク:

- *DB2 Warehouse Manager* インストール・ガイド の『インフォメーション・カタログ・マネージャーのコンポーネントのインストール』
- *DB2 Warehouse Manager* インストール・ガイド の『インフォメーション・カタログ・センターのインストール』
- 216 ページの『AIX Websphere IBM HTTP Web サーバーへの、Web 用インフォメーション・カタログ・センターのインストール』
- 218 ページの『Web 用インフォメーション・カタログ・センターのインストール後の構成』

AIX、Linux、および Solaris オペレーティング環境のウェアハウス・エージェント環境の構成

ウェアハウス・エージェント環境を構成するには、IWH.environment ファイルに特定の環境変数を設定する必要があります。変更しなければならない環境変数は、すべて START コメント行と END コメント行の間にあります。

前提条件:

以下の規則は、ODBC と DB2 CLI の両方のユーザーに適用されます。

1. すべての状況で DB2INSTANCE 変数を設定する必要があります。
2. DB2 CLI アクセスの場合、設定する必要があるのは DB2 environment セクションの変数だけです。

DB2 データベース内のウェアハウス・ソースまたはターゲットにアクセスするには、DB2 environment セクションの変数を設定する必要があります。

3. 設定する変数については、コメント文字 (#) を除去する必要があります。

以下の規則は、ODBC ユーザーに適用されます。

1. ODBC アクセスの場合、IS_ODBC 変数を ODBC インストール・パスに設定する必要があります。この変数は、コメントを持つことはできません。

AIX では、デフォルト・パスは `/usr/opt/db2_08_01/odbc` です。Linux および Solaris Operating Environment では、デフォルト・パスは `/opt/IBM/db2/V8.1/odbc` です。

2. IWH.environment ファイルの該当するセクションのコメントをはずして、選択したすべてのデータベースについて、ODBC 環境変数を設定してください。

AIX サーバーからインフォメーション・カタログにメタデータを公開するには、以下の変数を IWH.environment ファイルに設定する必要があります。

```
#CLASSPATH=/home/db2inst/sqllib/java/db2java.zip
           :/home/db2inst/sqllib/java/sqlj.zip
           :/home/db2inst/sqllib/java/runtime.zip
           :/home/db2inst/sqllib/tools/ibmjsse.jar
           :/home/db2inst/sqllib/tools/db2_vw_web.jar
           :/home/db2inst/sqllib/java/Common.jar
           :/home/db2inst/sqllib/tools/db2_vw.jar
           :/home/db2inst/sqllib/tools/db2vwcom.jar
           :/home/db2inst/sqllib/tools/db2vwddd.jar
           :/home/db2inst/sqllib/tools/db2cmn.jar
           :/home/db2inst/sqllib/java/db2jcc.jar:
```

```
#export CLASSPATH
```

```
#PATH=./usr/bin:/usr/bins/etc:/usr/sbin:/usr/ucb
       :/usr/bin/ X11:/sbin:/home/db2inst/sqllib/bin:
```

```
#export PATH
```

IWH.environment ファイルの設定を変更したら、ウェアハウス・エージェント・デーモンを再始動する必要があります。

手順:

ウェアハウス・エージェント環境を構成するには、以下のようにします。

1. 接続タイプを設定します。
 - a. root としてログオンします。
 - b. 以下のいずれかのコマンドを入力することによって、インストール・ディレクトリー内の bin サブディレクトリーに移動します。
 - AIX では、cd /usr/opt/db2_08_01/bin
 - Linux および Solaris オペレーティング環境では、cd/opt/IBM/db2/V8.1/bin
 - c. IWH2AGNT への正しいソフト・リンクを確立します。これを行うには、IWH.agent.db.interface コマンドを実行します。Linux では、まず /usr/bin から /bin に変更して、ファイルを変更する必要があります。Linux では、chmod、dirname、ln および pwd コマンドのロケーションは AIX と違います。which コマンドを実行して、これらのコマンドのパスを見付けて、IWH.agent.db.interface 内のパスを変更します。このコマンドの構文は以下のとおりです。

```
IWH.agent.db.interface [ intersolv | db2cli ]
```

ウェアハウス・エージェント実行可能ファイルには、以下の 2 つのバージョンがあります。

- IWH2AGNT.db2cli ファイルは、DB2 CLI リンクです。DB2 ファミリーにアクセスし、DB2 Relational Connect を通じて異機種データ

(Oracle, Sybase, Microsoft SQL Server, Informix, VSAM, および IMS) にアクセスするには、このリンクを使用してください。

これはデフォルト・リンクです。

- IWH2AGNT.ivodbc ファイルは、ODBC リンクです。DB2 ファミリー、Oracle, Sybase, および Informix のデータベースとデータ・ファイルにアクセスするには、このリンクを使用してください。

2. DB2 や DB2 Relational Connect ソースまたはターゲット・データにアクセスするには、IWH.environment ファイルに以下の変数を設定します。AIX では、ファイルは `/usr/opt/db2_08_01/bin` ディレクトリーにあります。Linux および Solaris オペレーティング環境では、ファイルは `/opt/IBM/db2/V8.1/bin` ディレクトリーにあります。これらの変数は、2 バイト文字セット (DBCS) と 1 バイト文字セット (SBCS) のどちらを使用するかに関係なく、設定する必要があります。

DB2INSTANCE

ウェアハウス・エージェントを実行する予定の DB2 インスタンスの名前。複数の DB2 インスタンスがある場合は、マスター DB2 インスタンスとなる 1 つのインスタンスを選択する必要があります。

LANG

ユーザーのローカルの地域。

LC_ALL

ユーザーのローカルの地域。LANG および LC_ALL 環境変数は、ウェアハウス・エージェントに対して、ロケールをローカル・ロケールに設定するよう指示します。

DB2CODEPAGE

クライアント・データベースのコード・ページ。DB2CODEPAGE 変数は、システム・コード・ページをオーバーライドします。この変数は、クライアントのデフォルト・コード・ページを解釈できないサーバーとの通信に使用されます。

DB2CODEPAGE 変数により、デフォルト・コード・ページがサーバーによってサポートされるコード・ページに設定されます。

3. データウェアハウス・センター ODBC ドライバーを使用して ODBC データにアクセスするには、IWH.environment ファイルに以下の変数を設定します。

- IS_ODBC = データウェアハウス・センター ODBC ドライバーをインストールした基本ディレクトリー。AIX では、デフォルト・パスは `/usr/opt/db2_08_01/odbc` です。Linux および Solaris Operating Environment では、デフォルト・パスは `/opt/IBM/db2/V8.1/odbc` です。
- Oracle データにアクセスするには、以下の環境変数を設定します。
 - ORACLE_HOME
 - TNS_ADMIN

- ORACLE_SID
- Sybase データにアクセスするには、以下の環境変数を設定します。
 - SYBASE
 - DSQUERY
- Informix データにアクセスするには、以下の環境変数を設定します。
 - INFORMIXDIR
 - INFORMIXSERVER

4. Essbase または DB2 OLAP Server™ データにアクセスするには、IWH.environment ファイルで ARBORPATH 環境変数を設定します。データ・ウェアハウス OLAP プログラムを使用する場合は、Essbase または DB2 OLAP が必要とする実行可能コードが PATH または LIBPATH 環境変数に含まれるように更新します。

Linux および Solaris オペレーティング環境では、Essbase または DB2 OLAP が必要とする実行可能コードが LD_LIBRARY_PATH 環境変数に含まれるように更新します。

エージェント・インスタンスの HOME 環境変数は、Essbase.id ファイルのある場所を指し、エージェント・プロセスにこのファイルへの読み取り/書き込みアクセスがなければなりません。Essbase.id は、最初に OLAP クライアントが実行された時に作成されます。

5. オプション: IWH.environment ファイルで以下の変数を設定することによって、環境をさらにカスタマイズします。
- VWS_LOGGING = ウェアハウス・エージェントのトレース・ファイル、プログラム出力ログ、および環境出力ファイルが保管されるディレクトリー
デフォルト・ディレクトリーは /var/IWH です。
 - VWSPATH = ウェアハウス・エージェントの基本インストール・ディレクトリー
 - VWA_LOCALE_TABLE = ロケール変換表
各国語版の Solaris オペレーティング環境ウェアハウス・エージェントを使用する場合は、エージェントがストリング変換に使用できるユーザー定義の変換表を選択する必要があります。これを行うには、IWH.environment ファイルに環境変数 VWA_LOCALE_TABLE を追加し、その値を、使用したいコード・セット (たとえば、en_US 用の 8859-1) に設定します。

6. NFS または AFS 環境で共有 /etc/services ファイルを使用する場合、共有 /etc/services ファイルに以下のサービスを追加します。

これらのポート番号が操作環境ですでに使用されている場合は、システムの /etc/services ファイルでそれらを変更することができます。ただしその場合は、ウェアハウス・サーバーがインストールされているワークステーション、および関連するすべてのウェアハウス・エージェント・サイトでも、同じポート番号を変更する必要があります。

インストール処理では、ローカル `/etc/services` ファイルのみが更新されるため、これらのサービスを追加する必要があります。

ポート名	ポート番号
vwd	11001/tcp

7. これらの変更を `IWH.environment` ファイルで有効にするために、ウェアハウス・エージェント・デーモンをいったん停止して再始動します。

a. デーモン・プロセス ID を判別するには、以下のように入力します。

```
ps -e | grep vwdaemon
```

b. デーモンを停止するには、以下のように入力します。

```
kill pid
```

ここで、`pid` はデーモン・プロセス ID です。

デーモンを停止すると、自動的に再始動されます。AIX ウェアハウス・サーバーをインストールした場合は、コマンド `db2vwsvr` を使用して、`IWH.environment` ファイルを変更した後、サーバー・デーモンも再始動する必要があります。

関連タスク:

- *DB2 Warehouse Manager* インストール・ガイド の『AIX、Linux、または Solaris オペレーティング環境のウェアハウス・エージェントをインストールした後の DB2 ノードおよびデータベースのカatalog』
- 223 ページの『AIX、Linux、および Solaris オペレーティング環境のウェアハウス・エージェントのインストール』
- *DB2 Warehouse Manager* インストール・ガイド の『データウェアハウス・センター外部トリガーの使用』
- 1076 ページの『ウェアハウス・サーバーとロガーの開始および停止 (AIX)』

関連資料:

- *DB2 Warehouse Manager* インストール・ガイド の『AIX、Linux、および Solaris オペレーティング環境のウェアハウス・エージェント用のデータベース接続』
- 228 ページの『AIX、Linux および Solaris オペレーティング環境用サンプル `IWH.environment` ファイル』

ウェアハウス・エージェントをインストールするための準備

ウェアハウス・エージェント・サイトは、ウェアハウス・ソースとウェアハウス・ターゲットにアクセスできなければなりません。ソースが Client Connect ソースの場合、クライアントは DB2 サーバーもカatalogする必要があります。ウェアハウス・ソースおよびターゲットを、Windows、Linux、AIX、および Solaris オペレーティング環境の ODBC からアクセスする場合は、これらのソースとターゲットのデータベースをシステム DSN として ODBC に登録する必要があります (Linux、AIX および Solaris オペレ

ーティング環境では、.odbc.ini ファイルにソースを指定する必要があります)。
AIX、Linux、または Solaris オペレーティング環境では、ウェアハウス・ソースおよびターゲットを、その下でエージェントが実行されるユーザー ID のホーム・ディレクトリーにある .odbc.ini ファイルに指定する必要があります。

UNIX ウェアハウス・エージェントを使用していて、IWH.environment ファイルに HOME 環境変数を定義している場合、.odbc.ini ファイルを \$HOME に指定されたディレクトリー・パスに置かなければなりません。ウェアハウス・サーバーを AIX にインストールし、デフォルト・エージェントを使用してウェアハウス・ソースおよびターゲットをアクセスする場合は、デフォルト・エージェントがアクセスするデータ用のウェアハウス・ソースおよびターゲットもサーバー・ワークステーションにカタログする必要があります。ウェアハウス・ソースおよびターゲットは、ウェアハウス・サーバーおよびデフォルト・エージェントがインストールされている場所にカタログする必要があります。 /usr/opt/db2_08_01/bin/IWH2AGNT ソフト・リンクが IWH2AGNT.ivodbc を指している場合、デフォルト・エージェントは AIX エージェントの ODBC バージョンを使用し、デフォルト・エージェントに参照されるデータ・ソースおよびターゲットは、ODBC ソースとして .odbc.ini にカタログされている必要があります。

前提条件:

ウェアハウス・エージェントは、以下のオペレーティング・システムで使用できます。

- Microsoft Windows NT ワークステーションまたは Service Pack 6 以上を持つ Windows NT Server バージョン 4.0。
- Microsoft Windows 2000 または Windows XP。
- IBM AIX バージョン 4.3.3 FixPak 2 以上。
- IBM OS/390 バージョン 2.10 以上。
- Intel 32 ビット・プロセッサでの Linux。サポートされている Linux ディストリビューション: カーネル 2.4.7、glibc 2.2.4。
- IBM iSeries V5R1 以上 (前提ソフトウェアの FixPaks および PTF を含む)。現行の保守リストについては、README ファイルを参照してください。iSeries ワークステーションにインストールされているソフトウェアを判別するには、iSeries コマンド・プロンプトで DSPSPWRSC と入力してください。
- Solaris Operating Environment バージョン 7 以上。

注: Linux プラットフォームの将来のリリースについてのサポートは未定です。詳細は Linux 用 DB2 Universal Database の Web サイトを参照してください。

DB2 OLAP ステップを使用している場合は、DB2 OLAP Server (バージョン 7 以上) または Essbase (バージョン 6 以上) のいずれかがインストールされている必要があります。Solaris オペレーティング環境では、DB2 OLAP Server バージョン 7、FixPak 8 以上 (Essbase バージョン 6.2) または DB2 OLAP Server バージョン 8 (Essbase バージョン 6.5) が必要です。

iSeries の場合、ALLOBJ および JOBCTL 権限があるユーザー ID を使用する必要があります。このレベルの権限は、iSeries RSTLICPGM コマンドと、STRVWD/ENDVWD コマンドの両方のために必要です。さらに、ウェアハウス・エージェント・デーモンを開始するユーザー・プロファイルのユーザー・クラスが *PGMR (バインド能力を持つ) でなければなりません。

AIX、Linux、Windows、および Solaris オペレーティング環境のウェアハウス・エージェントには、それぞれ 50 MB のハード・ディスク・スペースが必要です。

z/OS 上のウェアハウス・エージェントには、およそ 95 MB の HFS スペースが必要です。

iSeries 上のウェアハウス・エージェントには、ハード・ディスク・スペースは適用されません。

手順:

ウェアハウス・エージェント・サイトでは、ウェアハウス・エージェント・サイトとウェアハウス・サーバーの間の接続を確立するために、TCP/IP をインストールしてください。

DB2 サーバーからリモート・クライアントにサービスを提供する場合は、DB2 サーバーで DB2COMM 環境変数を設定してください。

```
db2set -i instance DB2COMM=TCPIP
```

この設定により、サーバーは、リモート・クライアントからの TCP/IP 要求を受け入れることができるようになります。

関連概念:

- *DB2 Warehouse Manager* インストール・ガイド の『ウェアハウス・エージェント』

関連タスク:

- 94 ページの『構成アシスタント (CA) によるデータベース接続の構成』
- *DB2 Warehouse Manager* インストール・ガイド の『Windows ウェアハウス・エージェントのインストール』
- 197 ページの『z/OS ウェアハウス・エージェントのインストール』
- 214 ページの『iSeries ウェアハウス・エージェントのインストール』
- 223 ページの『AIX、Linux、および Solaris オペレーティング環境のウェアハウス・エージェントのインストール』

z/OS ウェアハウス・エージェントのインストール

z/OS ウェアハウス・エージェントには *DB2 Warehouse Manager* プログラム・ディレクトリーが含まれており、ここにはストレージ、プログラミング、および駆動システムとターゲット・システムの要件についての情報があります。

前提条件:

z/OS ウェアハウス・エージェントは以下のソフトウェアを必要とします。

- OS/390 バージョン 2.10 以上
- DB2 Universal Database for OS/390 バージョン 6 以上
- OS/390 UNIX システム・サービス

以下の表は、z/OS エージェントをインストールする前に DB2 サブシステムに適用する必要がある APAR を示しています。

表 37. 必須 APAR

DB2 Universal Database バージョン	必須 APAR
DB2 for OS/390 バージョン 6	PQ36585、PQ46261
DB2 (z/OS 版および OS/390 版) バージョン 7	PQ36585、PQ46261

制約事項:

z/OS ウェアハウス・エージェントは、以下のアクセス権を必要とします。

- 許可プログラム機能 (APF) 権限を持つユーザー ID (UNIX システム・サービスで作業するため)
- ODBC プランに対する EXECUTE 権限
- ログイングおよび ODBC トレース・ディレクトリーへの READ および WRITE 権限 (ログイングまたは ODBC トレースをオンで実行する場合)

手順:

z/OS ウェアハウス・エージェントをインストールするには、以下のようになります。

1. SMP/E テープからウェアハウス・エージェントのコードをアンロードします。製品と一緒にプログラム・ディレクトリーが提供されています。

SMP/E インストールの結果のターゲット・ライブラリーの 1 つとして UNIX システム・サービス・ディレクトリーがあり、ここにウェアハウス・エージェントの実行可能コードが含まれます。このディレクトリーのデフォルト名は /usr/lpp/DWC81 です。このディレクトリーは、ウェアハウス・エージェントの実行可能ファイルを含んでいます。

2. エージェント・デーモンを開始するそれぞれのユーザー ID ごとに、Unix システム・サービス内にホーム・ディレクトリーをセットアップします。エージェントは環境変数を必要とし、この環境変数は各種の DB2 ライブラリーおよび出力ディレクトリーを指します。これらの変数をセットする最善の方法は、エージェント・デーモンを開始するユーザーのホーム・ディレクトリーにある `.profile` ファイルにこれらの変数を置くことです。こうすれば、ユーザーがサインオンした時に `.profile` ファイルが自動的に実行され、環境変数がセットされます。

以下の例はサンプルの `.profile` ファイルの内容を示しています。

```
export VWS_LOGGING=/usr/lpp/DWC81/logs
export VWP_LOG=/usr/lpp/DWC81/vwp.log
export VWS_TEMPLATES=/usr/lpp/DWC81
export DSNAOINI=/u/userid/dsnaoini
export LIBPATH=/usr/lpp/DWC81/ :$LIBPATH
export PATH=/usr/lpp/DWC81/ :$PATH
export STEPLIB=DSN710.SDSNEXIT:DSN710.SDSNLOAD
```

3. カーネルとデーモンの接続をセットアップします。これらの接続をセットアップするには、`/etc/services` ファイルまたは `TCPIP.ETC.SERVICESFILE` に以下の行を追加します。

```
vwkernel 11000/tcp
vwd 11001/tcp
vwlogger 11002/tcp
```

z/OS ウェアハウス・エージェントとデータベース間の接続をセットアップするには、z/OS コミュニケーション・データベースに何らかのリモート・データベースを追加します。

以下の例は、Windows NT で実行されているデータベースを z/OS 上のデータベースに接続するためのコミュニケーション・データベースへの挿入の例です。

```
INSERT INTO SYSIBM.LOCATIONS
  (LOCATION, LINKNAME, PORT)
VALUES
  ('NTDB', 'VWNT704', '60002');
INSERT INTO SYSIBM.IPNAMES
  (LINKNAME, SECURITY_OUT, USERNAMES, IPADDR)
VALUES
  ('VWNT704', 'P', '0', 'VWNT704.STL.IBM.COM');
INSERT INTO SYSIBM.USERNAMES
  (TYPE, AUTHID, LINKNAME, NEWAUTHID, PASSWORD)
VALUES
  ('0', 'MVSUID', 'VWNT704', 'NTUID', 'NTPW');
```

4. ODBC をデータベースにローカルに、またはリモートでバインドします。z/OS ウェアハウス・エージェントは ODBC を使用して DB2 と通信するので、エージェントがアクセスするすべてのローカルおよびリモート・データベースに ODBC プランをバインドする必要があります。

以下の例は、z/OS データベース用のローカル DB2 のためのいくつかのバインド・パッケージ・ステートメントの例を示しています。この例では、ローカル・システムの名前は *DWC6*、およびそのサブシステム上の ODBC パッケージ名として名前 *DWC6CLI* を使用すると想定しています。

```
DSN SYSTEM(DWC6)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLICS) ISO(CS)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLINC) ISO(NC)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIRR) ISO(RR)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIRS) ISO(RS)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIUR) ISO(UR)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIQR) ISO(QR)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIMS)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIC1)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIC2)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIF4)
```

以下の例は、z/OS データベース用のリモート DB2 のためのいくつかのバインド・パッケージ・ステートメントの例を示しています。ここで *REMLOC* は、コミュニケーション・データベースに登録された z/OS システム用のリモート DB2 のロケーション名です。

```
DSN SYSTEM(DWC6)
BIND PACKAGE (REMLOC.DWC6CLI) MEMBER(DSNCLICS) ISO(CS)
BIND PACKAGE (REMLOC.DWC6CLI) MEMBER(DSNCLINC) ISO(NC)
BIND PACKAGE (REMLOC.DWC6CLI) MEMBER(DSNCLIRR) ISO(RR)
BIND PACKAGE (REMLOC.DWC6CLI) MEMBER(DSNCLIRS) ISO(RS)
BIND PACKAGE (REMLOC.DWC6CLI) MEMBER(DSNCLIUR) ISO(UR)
BIND PACKAGE (REMLOC.DWC6CLI) MEMBER(DSNCLIQR) ISO(QR)
BIND PACKAGE (REMLOC.DWC6CLI) MEMBER(DSNCLIMS)
BIND PACKAGE (REMLOC.DWC6CLI) MEMBER(DSNCLIC1)
BIND PACKAGE (REMLOC.DWC6CLI) MEMBER(DSNCLIC2)
BIND PACKAGE (REMLOC.DWC6CLI) MEMBER(DSNCLIF4)
```

以下の例は、Windows NT で実行される DB2 Universal Database のためのいくつかのバインド・パッケージ・ステートメントの例を示しています。

```
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLICS) ISO(CS)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLINC) ISO(NC)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIRR) ISO(RR)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIRS) ISO(RS)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIUR) ISO(UR)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIC1)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIC2)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIQR)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIF4)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIV1)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIV2)
```

次に示すのは、1 つのプラン内にすべてのロケーションの ODBC パッケージを一緒にバインドするバインド・ステートメントの例です。

```
DSN SYSTEM(DWC6)
BIND PLAN(DWC6CLI) PKLIST(*.DWC6CLI.* )
```

ODBC パッケージのバインディングを完了したならば、.profile ファイル内の DSNAOINI 環境変数が、たった今バインドした ODBC プランを使用する ODBC 初期設定ファイルを指していることを確認してください。

たとえば、ODBC プランの名前が DW6CLI であり、ローカル・システムの名前が DW6 の場合、ODBC 初期設定ファイルには以下の情報が含まれているはずですが。

```
;SUBSYSTEM stanza
[DW6]
MVSATTACH=CAF
PLANNAME=DW6CLI
```

5. ODBC 初期設定ファイルをセットアップします。サンプルの ODBC 初期設定ファイル INISAMP が usr/lpp/DW6CLI/ ディレクトリにあります。このファイルをご使用のシステム用に編集することも、独自のファイルを作成することもできます。ファイルが正しく働くことを確認するため、以下のように適切に構成されていることを検査してください。

- DSNAOINI 環境変数が初期設定ファイルを指している必要があります。
- ファイルのコモン・スタンザに CONNECTTYPE=2 を含める必要があります。
- サブシステム・スタンザ内の PLANNAME 値は、最後のステップでバインドした ODBC プランの名前でなければなりません。
- ファイルのサブシステム・スタンザに MVSATTACHTYPE=CAF を含める必要があります。

DB2 システム用のデータ・ソース・スタンザがあることを確認します。これはローカル DB2 システムのロケーション名を指定する必要があります。

次に示すのは DSNAOINI ファイルの例です。

```
[COMMON]
MVSDEFAULTSSID=DW6
CONNECTTYPE=2
;APPLTRACE=1
;APPLTRACEFILENAME=/usr/lpp/DW6CLI/logs/application.CLITRACE
;TRACEFLUSH=1
;Example SUBSYSTEM stanza for DW6 subsystem
[DW6]
MVSATTACHTYPE=CAF
PLANNAME=DW6CLI
;DATA SOURCE stanza for ISC710P1 data source
[LOCATION_DSN]
```

ODBC トレースをオンにするには、COMMON セクションの最初の欄にあるセミコロンを除去します。

odbctest プログラムを使用して、リモート・データベースへの接続性をテストできます。プログラムを実行するには、UNIX システム・サービス・シェルで以下のシステムを入力します。

```
odbcstest <dsn> <userid> <password> <<towner>> <<tname>> <<sqlstmt>>
```

dsn は上で入力された LOCATION 値、userid は DB2 ユーザー ID、password は DB2 ユーザー ID のパスワード、towner はこの表スキーマのすべての表が返されることを指定、tname は表 towner の列情報を指定 (tname が返される)、sqlstmt は実行される引用 SQL ステートメントを示します。

最初の 3 つのパラメーター、dsn、userid および password のみが必要です。プログラムは、接続している DB2 システムについての情報を返し、すべてのユーザー表のリストを提供します。パラメーター towner、tname および sqlstmt は、それぞれ入力される直前のパラメーターが必要です。

6. z/OS ウェアハウス・エージェントは、デーモン・プロセスです。エージェント・デーモンは _password() 関数を使用するので、これらのエージェント実行可能プログラムを RACF プログラム制御に定義します。

```
libt1s4d.dll  
iwhcomnt.dll  
vwd
```

実行可能プログラムを RACF プログラム制御に定義するには、データウェアハウス・センター実行可能ファイルが保管されているディレクトリーに変更し、以下のコマンドを実行します。

```
extattr +p libt1s4d.dll  
extattr +p iwhcomnt.dll  
extattr +p vwd
```

extattr コマンドで +p パラメーターを使用するには、BPX.FILEATTR.PROGCTL FACILITY クラスに対して少なくとも READ 権限が必要です。

以下の例は、ユーザー ID JOEUSER にこの許可を与える RACF コマンドを示しています。

```
RDEFINE FACILITY BPX.FILEATTR.PROGCTL UACC(NONE)  
PERMIT BPX.FILEATTR.PROGCTL CLASS(FACILITY) ID(JOEUSER)  
ACCESS(READ)  
SETROPTS RACLIST(FACILITY) REFRESH
```

7. エージェント・デーモンを開始します。z/OS および OS/390 ホスト名および USS ポートを使用して OS/390 UNIX システム・サービスに接続するには、Telnet を使用します。

関連概念:

- *DB2 Warehouse Manager* インストール・ガイド の『z/OS および OS/390 版 DB2 表 およびフラット・ファイルのサンプルの内容』
- *DB2 Warehouse Manager* インストール・ガイド の『ウェアハウス・エージェント』
- 225 ページの『z/OS ウェアハウス・エージェントの概要』

関連タスク:

- 202 ページの『z/OS ウェアハウス・エージェント・デーモンの開始』
- 226 ページの『エージェント・デーモンを z/OS 開始済みタスクとして開始する』

z/OS ウェアハウス・エージェント・デーモンの開始

通常、z/OS ウェアハウス・エージェント・デーモンを開始するユーザー ID には、BPX.DAEMON FACILITY クラス・プロファイルの使用を許可すべきではありません。エージェント・デーモンは `setuid()` コマンドを実行し、これは制御の対象となる機能です。あるユーザー ID がプログラムを実行し、このプログラムがコマンドを実行し、このコマンドが制御の対象となる機能である場合、UNIX システム・サービス・カーネルは、そのユーザー ID が BPX.DAEMON ファシリティー・クラス・プロファイルの使用を許可されているかどうかをチェックします。許可されている場合、次にカーネルは、アドレス・スペースにロードされたすべてのプログラムが、プログラム制御に定義されているかどうかをチェックします。制御対象でないプログラムがロードされている場合、そのアドレス・スペースはダーティーとしてマークされます。この場合、プログラムは、一切の制御機能 (`setuid()` など) を実行できません。この場合、EMVSERR 戻りコードが JRENVIRTY 理由コードと一緒に表示されます。メッセージ: "BPXP014I ENVIRONMENT MUST BE CONTROLLED FOR DAEMON (BPX.DAEMON) PROCESSING" は、以下のことを意味します。

- エージェント・デーモンを開始したユーザー ID は、BPX.DAEMON ファシリティー・クラス・プロファイルの使用を許可されていた。
- コマンドが実行され、それは制御機能であった。
- 制御対象でないプログラムが少なくとも 1 つロードされた。

手順:

フォアグラウンドでデーモンを開始するには、UNIX システム・サービス・シェルの下でコマンド行に `vwd` と入力します。

バックグラウンドでデーモンを開始するには、以下のように入力します。

```
vwd>/usr/lpp/DWC81/logs/vwd.log 2>&1 &
```

上のコマンドは、UNIX システム・サービス・シェルの下でコマンド行に入力します。ここで、`/usr/lpp/DWC81/logs/vwd.log` は、デーモン出力をパイピングしたいパスおよびファイルの名前です。

z/OS ウェアハウス・エージェントが実行されていることを検証するには、UNIX シェルのコマンド行に `ps -e | grep vwd` と入力します。

```
| または、z/OS コンソールで D OMVS,a=a11 と入力し、ストリング vwd を探します。また、z/OS または OS/390 コンソールに、D J,vwdproc と入力することもできます。vwdproc は上で作成したプロシージャの名前です。2 つのエージェント・デーモンが
```

同じ名前で行われているように見えます。実際は、ジョブの 1 つは開始されたタスク JCL であり、もう 1 つは UNIX システム・サービスで実行中のエージェント・デーモンです。

関連概念:

- *DB2 Warehouse Manager* インストール・ガイド の『ウェアハウス・エージェント』
- 225 ページの『z/OS ウェアハウス・エージェントの概要』

関連タスク:

- 197 ページの『z/OS ウェアハウス・エージェントのインストール』
- 226 ページの『エージェント・デーモンを z/OS 開始済みタスクとして開始する』

DB2 Warehouse Manager ユーザー定義プログラム

DB2[®] Warehouse Manager (z/OS[™] 版) のパッケージには、以下のユーザー定義プログラムが含まれています。

- VWPFTP プログラムは、FTP コマンド・ファイルを実行します。
- VWPMVS プログラムは、JCL ジョブ・ストリームをサブミットします。
- VWPRCPY プログラムは、FTP を使用してファイルをコピーします。
- XTClient は、クライアント・トリガー・プログラムです。
- sendWTO プログラムは、ステップ完了の通知を z/OS コンソールに提供します。

これに追加して、ユーザー定義プログラムおよびストアード・プロシージャをデータウェアハウス・センターに作成することができます。z/OS ウェアハウス・エージェントは、UNIX[®] システム・サービスの下で実行される実行可能プログラムならば何でもサポートします。

ユーザー定義プログラムは 1 つまたは複数のステップに割り当てられます。ユーザー定義プログラムを実行すると、以下のアクションが取られます。

- エージェントが、ユーザー定義プログラムを実行します。
- ユーザー定義プログラムは、戻りコードとフィードバック・ファイルをエージェントに戻します。
- エージェントは、その結果をカーネルに戻します。

VWP_LOG 環境変数を使用して、ユーザー定義プログラムが出力を書き込めるディレクトリーを定義してください。

ユーザー定義プログラムを使用して、FTP でジョブを転送する場合は、最初にサブミットしたい JCL とデータを作成する必要があります。JCL 内のジョブ名は、USERIDX でなければならず、ここで X は、1 文字の文字または数字です (例: JOEUSERA)。JCL に含まれる MSGCLASS および SYSOUT ファイルの出力クラスは、JES が保持する出力クラスを指定する必要があります。

サブミットされるジョブの最大 LRECL は、254 文字です。 JES は JCL の最初の 72 文字だけをスキャンします。

関連概念:

- *DB2 Warehouse Manager* インストール・ガイド の『z/OS トランスフォーマー』
- 225 ページの『z/OS ウェアハウス・エージェントの概要』

関連タスク:

- 197 ページの『z/OS ウェアハウス・エージェントのインストール』
- 208 ページの『z/OS ウェアハウス・トランスフォーマー用の環境変数データ・セット内の文字数の削減』

関連資料:

- *DB2 Warehouse Manager* インストール・ガイド の『Trillium ユーザー定義プログラム・ステップ用の z/OS ウェアハウス・エージェント・サポート』

トリガー・プログラムによるウェアハウス・ステップのスケジューリング (XTClient)

トリガー・プログラムを使用して、z/OS または OS/390 からウェアハウス・ステップをスケジュールします。データウェアハウス・センター内のステップを起動するジョブをサブミットできるのは、ユーザーまたは OS/390 ジョブ・スケジューラーです。ステップが成功すると、JCL 内のトリガー・ステップは、ゼロの戻りコードを戻します。

前提条件:

トリガー・プログラムを使用するには、Unix システム・サービスに Java Development Kit (JDK) 1.1.8 以上をインストールしている必要があります。

手順:

トリガー・プログラムを開始するには、ウェアハウス・サーバーが実行されているコンピュータで XTServer を開始します。

Windows で XTServer が開始したならば、Unix システム・サービスで、またはバッチ JCL を使用して、XTClient を開始します。

トリガーを開始するサンプル JCL を以下に示します。

```
| //DBA1A JOB 1,'XTCLIENT',CLASS=A,MSGCLASS=H,  
| // MSGLEVEL=(1,1),REGION=4M,NOTIFY=&SYSUID  
| //*****  
| /* submit iwhetrig  
| //*****  
| //BRADS EXEC PGM=BPXBATCH,  
| // PARM=('sh cd /usr/lpp/DWC81/; java XTClient 9.317.171.133 1100x  
| // 9 drummond pw bvmvs2nt 1 1 100')
```



```
| //STDOUT DD PATH='/tmp/xtclient.stdout',  
| // PATHOPTS=(OWRONLY,OCREAT),  
| // PATHMODE=SIRWXU  
| //STDERR DD PATH='/tmp/xtclient.stderr',  
| // PATHOPTS=(OWRONLY,OCREAT),  
| // PATHMODE=SIRWXU  
| //  
|
```

上記のサンプル JCL では、パラメーターが次の行に継続していることが示されています。パラメーターを列 71 まで入力し、列 72 に X を指定し、次の行の列 16 から続きを入力します。パラメーターの最初の部分 (cd /usr/lpp/DWC81/;) は、z/OS ウェアハウス・エージェントがインストールされているディレクトリーに変更するステートメントです。

パラメーターの 2 番目の部分は、XTClient を開始し、以下のパラメーターを渡しています。

- ご使用の DWC サーバー・ホスト名または IP アドレス
- XTServer が listen しているご使用の DWC サーバー・ポート番号 (例: 11009)
- DWC ユーザー ID
- DWC パスワード
- 実行するステップの名前
- XTserver コマンド。以下のとおりです。
 - 1 = ステップを充てんする
 - 2 = ステップをテスト・モードにプロモートする
 - 3 = ステップを実動モードにプロモートする
 - 4 = ステップをテスト・モードにデモートする
 - 5 = ステップを開発モードにデモートする
- ステップの完了を待つかどうかを示すオプションであり、1= yes (待つ)、または 0 = no (待たない)。
- 行の最大数 (すべての行を取り出す場合は 0 またはブランクを使用)

関連概念:

- *DB2 Warehouse Manager* インストール・ガイド の『z/OS トランスフォーマー』

関連タスク:

- 197 ページの『z/OS ウェアハウス・エージェントのインストール』

関連資料:

- *DB2 Warehouse Manager* インストール・ガイド の『Trillium ユーザー定義プログラム・ステップ用の z/OS ウェアハウス・エージェント・サポート』

z/OS 版 DB2 でのウェアハウス・トランスフォーマーのセットアップ

以下の手順は、z/OS および OS/390 版 DB2 でのウェアハウス・トランスフォーマーのセットアップ方法を説明しています。

手順:

z/OS 版 DB2 でウェアハウス・トランスフォーマーをセットアップするには、以下のようになります。

1. トランスフォーマーのソースをウェアハウスに定義します。
2. z/OS 版 DB2 にトランスフォーマーを定義するには、必要な SQL ステートメントを含むディレクトリーに行き、SQL ステートメントを適用します。SQL ステートメントのロケーションは、DB2 のバージョンにより異なります。

表 38. DB2 Universal Database バージョン別の SQL ステートメントのロケーション

DB2 Universal Database バージョン	SQL ステートメントのロケーション
DB2 (OS/390 版) バージョン 6	/usr/lpp/DWC81/createXfSQL ディレクトリー内の SQL ステートメントを使用する
DB2 (z/OS 版および OS/390 版) バージョン 7	/usr/lpp/DWC81/createXfSQLV7 ディレクトリー内の SQL ステートメントを使用する

Java ストアード・プロシージャーをセットアップする時に、WLM を使用して Java WLM 始動プロシージャーと WLM 環境名を関連付けます。環境名は、CREATE PROCEDURE ステートメントの WLM ENVIRONMENT オプションに指定します。DSNWLMJ は、トランスフォーマー定義に含まれる WLM 環境名です。

DSNWLMJ の WLM 関連名を追加するか、またはそれぞれのトランスフォーマー定義の WLM ENVIRONMENT オプションを、始動プロシージャーとすでに関連付けられている名前に変更します。

3. UNIX システム・サービスから IWH810.SIWHLOAD 内のトランスフォーマー・ロード・モジュールへのリンクを以下のようにセットアップします。
 - a. Telnet を使用して、OS/390 または z/OS ホスト・システムに UNIX システム・サービスを接続します。
 - b. z/OS ウェアハウス・エージェントをインストールしたディレクトリーに変更します。デフォルトのインストール・ディレクトリーは、/usr/lpp/DWC81 です。
 - c. DB2 V7 を使用している場合は、ステップ 3e (207 ページ) に行きます。DB2 バージョン 5 またはバージョン 6 を使用している場合は、インストール・ディレクトリーにある trlinks データ・セットを編集します。
 - d. 列 1 にポンド記号 (#) を指定してこの行をコメントにします。たとえば、#ln -e IWHXF xf.j11; とします。列 1 のポンド記号 (#) を除去して、この行からコメントを除去します。たとえば、ln -e IWHXFV7 xf.j11; とします。変更を保管します。

- e. `trlinks` と入力し、Enter キーを押します。 `/usr/lpp/DWC81/com/ibm/data` ディレクトリーに `xf.j11` リンクが作成されます。このリンクは、`IWHXF` または `IWHXFV7` モジュールのいずれかをロードすることをエージェントに指示します。
4. APF 許可 `IWH810.SIWHPDSE` を次に `DB2 Java` ストアード・プロシージャー始動プロシージャー内の `STEPLIB` 連結に追加します。
5. `xf.j11` リンクが存在するディレクトリー (デフォルトは `/usr/lpp/DWC81`) を、`WLM` 環境データ・セット内の `CLASSPATH` および `LIBPATH` 環境変数に追加します。`WLM` 環境データ・セットを見つけるには、`DB2 Java` ストアード・プロシージャー始動プロシージャーを探します。`WLM` 環境データ・セットは、`JAVAENV DD` カードが指しているデータ・セットです。
6. ストアード・プロシージャーを開始し、ウェアハウス・ステップを作成して実行します。

関連概念:

- *DB2 Warehouse Manager* インストール・ガイド の『ウェアハウス・トランスフォーマー用 `JDK` のインストール』
- *DB2 Warehouse Manager* インストール・ガイド の『`z/OS` トランスフォーマー』
- *DB2 Warehouse Manager* インストール・ガイド の『ウェアハウス・トランスフォーマー』

関連タスク:

- *DB2 Warehouse Manager* インストール・ガイド の『`Windows` 上のウェアハウス・トランスフォーマー用環境変数の更新』
- *DB2 Warehouse Manager* インストール・ガイド の『`AIX` 上のウェアハウス・トランスフォーマー用環境変数の更新』
- *DB2 Warehouse Manager* インストール・ガイド の『ターゲット・データベース用のデータベース構成の更新』
- *DB2 Warehouse Manager* インストール・ガイド の『ウェアハウス・トランスフォーマーのインストール』
- *DB2 Warehouse Manager* インストール・ガイド の『ウェアハウス・トランスフォーマーの使用可能化』
- *DB2 Warehouse Manager* インストール・ガイド の『ウェアハウス・トランスフォーマー用の環境の準備』
- *DB2 Warehouse Manager* インストール・ガイド の『`DB2` サブシステムでの `Java` ストアード・プロシージャーの、ウェアハウス・トランスフォーマー用セットアップ』
- *DB2 Warehouse Manager* インストール・ガイド の『`Solaris` オペレーティング環境の環境変数のウェアハウス・トランスフォーマー用の更新』

- *DB2 Warehouse Manager* インストール・ガイド の『Linux 上のウェアハウス・トランスフォーマー用環境変数の更新』

関連資料:

- *DB2 Warehouse Manager* インストール・ガイド の『ウェアハウス・トランスフォーマーの各国語サポート』

z/OS ウェアハウス・トランスフォーマー用の環境変数データ・セット内の文字数の削減

環境変数データ・セットに含めることができる文字数の最大は 244 文字です。

前提条件:

JDBC のインストール時に `installVAJDLL` コマンドを実行します。このコマンドは JDBC ドライバーに `sqlj.jll` リンクを作成します。この `sqlj.jll` リンクをインストールしたディレクトリーの名前を、環境変数データ・セット内の `CLASSPATH` および `LIBPATH` ステートメントに追加する必要があります。

`SQLj` は `/COM/ibm/db2os390/` という名前の Java パッケージ名に含まれます。たとえば、`sqlj.jll` リンクが `/usr/lpp/db2710/COM/ibm/db2os390/` ディレクトリーにある場合は、`CLASSPATH` および `LIBPATH` ステートメントに `/usr/lpp/db2710/` を追加する必要があります。

手順:

環境変数データ・セット内の文字数を減らすには、以下のようにします。

1. たとえば `/u/userid/links` のようなディレクトリーを作成します。この `/u/userid/links` ディレクトリーに、`JDBC` リンクおよびトランスフォーマー・リンクの両方を含めます。

```
/u/userid/links/com/ibm/data/xf.jll
/u/userid/links/COM/ibm/db2os390/sqlj.jll
```

2. `LIBPATH` ステートメントを `LIBPATH=/u/userid/links` に変更します。
3. `CLASSPATH` ステートメントを `/u/userid/links` に変更します。

リンクを移動する場合は、`.jll` ファイルに再度権限を実行し、拡張ビットをオンにする必要があります。

拡張ビットがオンかどうかを判別するには、以下のようにします。

1. リンクを含むディレクトリーに行き、コマンド `ls -lt` を入力します。
高位ビット位置にある許可ビットに `e` が含まれています。たとえば、`erwxrwxrwx` となります。

2. 高位ビットが 1 (たとえば、1wxrwxrwx) であり、かつ DB2 (OS/390 版) バージョン 5 またはバージョン 6 を実行している場合、以下のコマンドを使用してリンクを再度作成する必要があります。

```
In -e DSNAQJLL sqlj.j11
In -e IWHXF xf.j11
```

高位ビットが 1 であり、かつ DB2 (z/OS および OS/390 版) バージョン 7 を実行している場合、以下のコマンドを使用してリンクを再度作成します。

```
In -e IWHXV7 xf.j1
```

ストアード・プロシージャのシグニチャー内の Java オブジェクトは、DB2 (z/OS および OS/390 版) バージョン 7 でのみサポートされます。DB2 (OS/390 版) バージョン 5 およびバージョン 6 では、トランスフォーマーはパラメーター内の NULL 値をサポートしません。バージョン 5 または 6 では、NULL のパラメーターを渡すとゼロと見なされます。DB2 バージョン 5 およびバージョン 6 では、トランスフォーマーはゼロのパラメーターを NULL スtring のように扱います。

DB2 は、ストアード・プロシージャ内の COMMIT SQL ステートメントを DB2 (z/OS および OS/390 版) バージョン 7 でのみサポートします。INVERTDATA ストアード・プロシージャは、ストアード・プロシージャ内の表をドロップし、再作成するため、COMMIT ステートメントを必要とします。IWH.INVERTDATA は、DB2 (OS/390 版) バージョン 5 またはバージョン 6 ではサポートされません。

DB2 (z/OS および OS/390 版) は、Java ユーザー定義関数をサポートしないので、IWH.FORMATDATE は S/390 プラットフォームではサポートされません。

以下は、Java ストアード・プロシージャ用のサンプルの始動プロシージャです。

```
//DSNWLMJ PROC DB2SSN=DSN,NUMTCB=5,APPLENV=DSNWLMJ
//*****
//* THIS PROC IS USED TO START THE WLM-ESTABLISHED SPAS *
//* ADDRESS SPACE FOR THE DSNWLMJ APPLICATION ENVIRONMENT *
//* V WLM,APPLENV=DSNWLMJ,RESUME *
//*****
//DSNWLMJ EXEC PGM=DSNX9WLM,TIME=1440,REGION=0M,
// PARM='&DB2SSN, &NUMTCB, &APPLENV'
// DD DSN=IWH810.SIWHPDSE,DISP=SHR
// DD DSN=DSN.HPJSP.PDSE.JDBC,DISP=SHR
// DD DSN=SYS1.PP.PDSELINK,DISP=SHR
// DD DSN=DSN710.SDSNEXIT,DISP=SHR
// DD DSN=DSN710.SDSNLOAD,DISP=SHR
// DD DSN=SYS1.SCEERUN,DISP=SHR
// DD DSN=DSN.PDSE,DISP=SHR
//JAVAENV DD DSN=DSN.WLMENVJ.JSPENV,DISP=SHR
//CEEDUMP DD SYSOUT=A
//DSSPRINT DD SYSOUT=A
//JSPDEBUG DD SYSOUT=A
//SYSABEND DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
```

この例で、IWH810.SIWHPDSE ライブラリーには、トランスフォーマーのロード・モジュールが含まれています。DSN.HPJSP.PDSE.JDBC には、HPJ セットアップからの High Performance Java (HPJ) DLL が含まれています。DB2 (z/OS 版) バージョン 7 では、このライブラリーの名前は DSN710.SDSNLOD2 です。SYS1.PP.PDSELINK には、HPJ ランタイム・ライブラリーが含まれています。DSN.PDSE には HPJ セットアップ情報が含まれています。DSN.WLMENVJ.JSPENV には環境変数が含まれていません。

次に示すのは、環境変数データ・セットです。

```
ENVAR("TZ=PST07",  
"DB2SQLJPROPERTIES=/usr/lpp/db2/jdbc/db2710/classes/db2sqljjdbc.properties",  
"LIBPATH=/usr/lpp/DWC81",  
"VWSPATH=/usr/lpp/DWC81",  
"CLASSPATH=/usr/lpp/db2/jdbc/db2710/classes:/usr/lpp/DWC81:/usr/lpp/hpj/lib"),  
MSGFILE(JSPDEBUG)
```

関連概念:

- *DB2 Warehouse Manager* インストール・ガイド の『ウェアハウス・トランスフォーマー用 JDK のインストール』
- *DB2 Warehouse Manager* インストール・ガイド の『z/OS トランスフォーマー』
- *DB2 Warehouse Manager* インストール・ガイド の『ウェアハウス・トランスフォーマー』

関連タスク:

- *DB2 Warehouse Manager* インストール・ガイド の『Windows 上のウェアハウス・トランスフォーマー用環境変数の更新』
- *DB2 Warehouse Manager* インストール・ガイド の『AIX 上のウェアハウス・トランスフォーマー用環境変数の更新』
- *DB2 Warehouse Manager* インストール・ガイド の『ターゲット・データベース用のデータベース構成の更新』
- *DB2 Warehouse Manager* インストール・ガイド の『ウェアハウス・トランスフォーマーのインストール』
- *DB2 Warehouse Manager* インストール・ガイド の『ウェアハウス・トランスフォーマーの使用可能化』
- *DB2 Warehouse Manager* インストール・ガイド の『ウェアハウス・トランスフォーマー用の環境の準備』
- *DB2 Warehouse Manager* インストール・ガイド の『DB2 サブシステムでの Java スタートアップ・プロシーチャーの、ウェアハウス・トランスフォーマー用セットアップ』
- 206 ページの『z/OS 版 DB2 でのウェアハウス・トランスフォーマーのセットアップ』

- *DB2 Warehouse Manager* インストール・ガイド の『Solaris オペレーティング環境の環境変数のウェアハウス・トランスフォーマー用の更新』
- *DB2 Warehouse Manager* インストール・ガイド の『Linux 上のウェアハウス・トランスフォーマー用環境変数の更新』

関連資料:

- *DB2 Warehouse Manager* インストール・ガイド の『ウェアハウス・トランスフォーマーの各国語サポート』

z/OS ウェアハウス・エージェントを使用した、DataPropagator レプリケーション・アプライ・ステップの自動化

z/OS ウェアハウス・エージェントを使用して、DataPropagator レプリケーションのアプライ・ステップを自動化することができます。レプリケーションには、ソース・データベース、制御データベース、およびターゲット・データベースが必要です (これらのデータベースは、別にも同じにもできます)。これらのデータベースには、レプリケーション・コントロール表をインストールする必要があります。キャプチャー・ジョブは DB2 ログを読み、ソース・データベース内のどの行を追加、更新、または削除するかを判断します。このジョブは次に、変更を変更データ表 (CD 表とも呼ばれる) に書き込みます。次に、アプライ・ジョブが実行され、ターゲット・データベースに変更をアプライします。DB2 Warehouse Manager パッケージはレプリケーション・ステップを作成することにより、このアプライ・ジョブを自動化することができます。データウェアハウス・センターを使用して、実行するアプライ・ジョブのステップおよび、いつ実行するかを定義します。ウェアハウス・エージェントは、ウェアハウス・ステップが実行される時にアプライ・プログラムを実行します。

DataPropagator バージョン 8 を使用するには、STEPLIB および PATH 環境変数を更新し、レプリケーション・ライブラリーおよび実行可能コードのディレクトリーを指すようにする必要があります。DataPropagator バージョン 8 は、JCL テンプレートを必要としません (JCL テンプレートは DataPropagator バージョン 7 以前では必要でした)。

DataPropagator バージョン 7 (またはこれ以前) を使用する場合は、レプリケーション・サポートには JCL テンプレートを使用する必要があります (テンプレートは DB2 Warehouse Manager パッケージに含まれています)。z/OS ウェアハウス・エージェントを使用してアプライ・プログラムを実行する場合は、このテンプレート内のアカウント情報とデータ・セット情報を、ご使用の OS/390 または z/OS システム用に変更する必要があります。

手順:

z/OS ウェアハウス・エージェントを使用して、DataPropagator バージョン 7 (またはこれ以前) のステップを自動化するには、以下のようにします。

1. /usr/lpp/DWC81/ ディレクトリー内のファイルをコピーし、更新する権限を持つ ID を使用してログオンします。
2. apply.jcl ファイルを見つけ、これを systemname.apply.jcl としてコピーします (ここで systemname は、OS/390 MVS システムの名前です)。
たとえば、STLMVS1 で、STLMVS1.apply.jcl という名前のファイルのコピーを作成します。
3. 要件を満たすように JCL をカスタマイズするには、テキスト・エディターを使用してください。アカウント情報を標準のアカウント情報に合わせて変更し、STEPLIB DD および MSGS DD のデータ・セットをご使用の OS/390 MVS システム用に変更します。
4. 必要ならば、EXEC カード上のプログラム名を変更します。
大括弧に含まれているパラメーター (たとえば、[USERID] や [APPLY_PARMS] など) は、一切変更しないでください。大括弧は 16 進文字では、左側が X'AD'、右側が X'BD' です。ご使用の TSO 端末タイプが SPF オプション 0 で 3278A にセットされていない場合、これらの値は大括弧としてではなく、特殊文字として表示される場合があります。大括弧として表示されなくても、X'AD' または X'BD' またはこれらの文字の間にあるデータを変更しなければ、問題はありません。
5. 環境変数 VWS_TEMPLATES を、コピーしたテンプレート・ファイルのディレクトリーを指すように更新します。

次に示すのは、データウェアハウス・センターに含まれている JCL テンプレートです。

```
//[USERID]A JOB ,MSGCLASS=H,MSGLEVEL=(1,1),
// REGION=2M,TIME=1440,NOTIFY=&SYSUID
/** DON'T CHANGE THE FIRST LINE OF THIS TEMPLATE.
/** THE REMAINING JCL SHOULD BE MODIFIED FOR YOUR SITE.
/*****
/** RUN APPLY/MVS ON OS/390 DB2 6.1 *
/*****
//ASNARUN EXEC PGM=ASNAPV66,REGION=10M,
// [APPLY_PARMS]
//STEPLIB DD DISP=SHR,DSN=DPROPR.V6R1M0.SASNLINK
// DD DISP=SHR,DSN=DSN610.SDSNLOAD
//MSGS DD DSN=DPROPR.V2R1M0A.MSGS,DISP=SHR
//ASNASPL DD DSN=&ASNASPL
,DISP=(NEW,DELETE,DELETE),
// UNIT=SYSDA,SPACE=(CYL,(10,1)),
// DCB=(RECFM=VB,BLKSIZE=6404)
//SYSTEM DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSPPRINT DD SYSOUT=*
//
```

関連概念:

- 225 ページの『z/OS ウェアハウス・エージェントの概要』

関連タスク:

- 197 ページの『z/OS ウェアハウス・エージェントのインストール』
- 213 ページの『ウェアハウス・エージェントのログギングの開始』

ウェアハウス・エージェントのログギングの開始

DB2 Warehouse Manager コンポーネントの多く (サーバー、ロガー、エージェント、およびいくつかのデータウェアハウス・センター・プログラムなど) は、ログを VWS_LOGGING 環境変数で指定されたログギング・ディレクトリーに書き込みます。これらのログ・ファイルは、プレーン・テキストです。IBM ソフトウェア・サポートは、デバッグのためログギングをオンにするようお願いする場合があります。

手順:

ウェアハウス・エージェント・デーモン・ログギングを開始するには、vwd コマンドの入力時に -t オプションを追加します。より高水準のトレースには、-t x を追加します。x は 1 から 4 までの数です。

データウェアハウス・センターからエージェント・ログギングを開始するには、以下のようになります。

1. 左ペインから、「ウェアハウス (Warehouse)」を右クリックし、「プロパティ (Properties)」をクリックします。
2. 「トレース・レベル (Trace Level)」ページで、希望するトレース・レベルに設定を変更します。

エージェント・トレースはレベル 0 から 5 をサポートします。

- レベル 1 - 入り口/出口のトレース
- Level 2 - レベル 1 に加えて、デバッグ・トレース
- Level 3 - レベル 2 に加えて、データ・トレース
- Level 4 - 内部バッファー・トレース

トレースをレベル 1 よりも高く設定すると、パフォーマンスが低下します。トレースは、デバッグに必要な場合のみオンにしてください。トレース情報は、ファイル AGNTxxx.LOG に保管されます。環境情報は、ファイル AGNTxxx.SET に保管されます。

関連概念:

- データウェアハウス・センター 管理ガイド の『DB2 for iSeries Load ユーティリティー用のトレース・ファイル』
- DB2 Warehouse Manager インストール・ガイド の『ウェアハウス・エージェント』
- データウェアハウス・センター 管理ガイド の『コンポーネント・トレース・データ』
- データウェアハウス・センター 管理ガイド の『開始エラー・トレース・ファイル』

関連タスク:

- データウェアハウス・センター 管理ガイド の『DB2 for iSeries Load ユーティリティー用のトレース・ファイルの表示』
- データウェアハウス・センター 管理ガイド の『データウェアハウス・センターのコンポーネント・トレースの実行』
- データウェアハウス・センター 管理ガイド の『アプライ・プログラムによって作成されるエラーのトレース』

関連資料:

- *DB2 Warehouse Manager* インストール・ガイド の『ウェアハウス・エージェント環境の構造』

iSeries ウェアハウス・エージェントのインストール

ウェアハウス・エージェントを iSeries (AS/400) ワークステーションにインストールすることができます。

前提条件:

iSeries ウェアハウス・エージェントをインストールする前に、以下のことを行ってください。

- 以前にインストールした iSeries ウェアハウス・エージェントをすべて除去します。
- ウェアハウス用のセキュリティと特権を定義します。
- 以下のソフトウェアがインストール済みであることを確認します。
 - IBM DB2 Warehouse Manager バージョン 8
 - IBM AS/400 V5R1 以上
- 以下の権限レベルを持っていることを確認します。
 - *ALLOBJ (製品のインストールに RSTLICPGM コマンドを使用するため)
 - *JOBCTL (STRVWD および ENDVWD コマンドを使用するため)
 - *SECOFR (インストールには必須ではないが、インストールを検証するには必要)。 DSPSFWRSC コマンドも実行できる。
 - *USE オブジェクト権限 (すべてのコマンドをアクセスするため)。ライブラリーを作成するには追加の権限が必要です。

手順:

iSeries ウェアハウス・エージェントをインストールするには、以下のようになります。

1. DB2 Warehouse Manager CD-ROM を iSeries CD-ROM ドライブに挿入します。
2. iSeries システムにログオンします。
3. iSeries コマンド・プロンプトで、以下のように入力します。

```
RSTLICPGM LICPGM(5765F42) DEV(OPT01)
```

ここで、OPT01 は CD-ROM ドライブです。インストール処理により、
/QIBM/UserData/IWH および /QIBM/ProdData/IWH サブディレクトリーが作成されま
す。トレースは、/QIBM/UserData/IWH サブディレクトリーに書き込まれます。

4. 「ライセンス・プログラムの処理 (Work with Licensed Programs)」画面で、iSeries
ウェアハウス・エージェントが正常にインストールされたことを検証します。
iSeries コマンド・プロンプトで GO LICPGM を入力し、オプション 10 (ライセンス交
付を受けたインストール済みプログラムの表示 (Display Licensed Installed Programs))
を使用します。

関連概念:

- *DB2 Warehouse Manager* インストール・ガイド の『ウェアハウス・エージェント』

関連タスク:

- 194 ページの『ウェアハウス・エージェントをインストールするための準備』
- *DB2 Warehouse Manager* インストール・ガイド の『iSeries ウェアハウス・エージェ
ントの除去』

関連資料:

- *DB2 Warehouse Manager* インストール・ガイド の『iSeries ウェアハウス・エージェ
ントの米国英語以外のインストール』
- *DB2 Warehouse Manager* インストール・ガイド の『iSeries ウェアハウス・エージェ
ントのインストール後の考慮事項』

インフォメーション・カタログ管理ウィザードの実行

インフォメーション・カタログを作成する、または既存のインフォメーション・カタロ
グを移行するには、「インフォメーション・カタログ管理 (Manage Information
Catalog)」ウィザードを実行します。ウィザードは、インストール中にインフォメーショ
ン・カタログを準備または移行しなかった場合のみ必要です。

手順:

「インフォメーション・カタログ管理 (Manage Information Catalog)」ウィザードを実行
するには、以下のようになります。

1. DB2 Universal Database にデータベースを作成するか、またはインフォメーション・
カタログを含む既存のデータベースを見つけます。
2. AIX ユーザー: AIX オペレーティング・システムでインフォメーション・カタロ
グ・センターを使用している場合、SQL1224N エラーを受け取る可能性があります
。このエラーを受け取った場合、可能な原因のリストのメッセージ・ヘルプ・テキ
ストを参照してください。問題が共有メモリー・セグメントの数に関係している場合
は、管理 API 関連資料で、並行アクセスを使用するスレッド化アプリケーションに

関する情報を参照してください。この状態の予備手段として、環境変数 EXTSHM の設定、またはアクセスされているデータベースへのループバックの定義などがあります。

3. Windows ユーザー: 「スタート」->「プログラム」->「IBM DB2」->「ツールのセットアップ (Set-up tools)」->「インフォメーション・カタログ管理ウィザード (Manage Information Catalog wizard)」をクリックします。 AIXユーザー: db2iccwz を入力します。「インフォメーション・カタログ管理 (Manage Information Catalog)」ウィザードがオープンします。
4. 「インフォメーション・カタログ管理 (Manage Information Catalog)」ウィザードの各ページで必要な情報を入力し、入力した情報をサマリー・ページで見直し、「終了 (Finish)」をクリックします。「インフォメーション・カタログ管理 (Manage Information Catalog)」ウィザードがクローズし、インフォメーション・カタログが正常に初期化または移行されます。

関連概念:

- *DB2 Warehouse Manager* インストール・ガイド の『インフォメーション・カタログ・マネージャー』

関連タスク:

- *DB2 Warehouse Manager* インストール・ガイド の『インフォメーション・カタログ・マネージャーのコンポーネントのインストール』
- *DB2 Warehouse Manager* インストール・ガイド の『インフォメーション・カタログ・センターのインストール』

AIX Websphere IBM HTTP Web サーバーへの、Web 用インフォメーション・カタログ・センターのインストール

Web 用インフォメーション・カタログ・センターは、DB2 Universal Database への JDBC 接続をサポートし、IBM WebSphere を実行する、任意のオペレーティング・システムにインストールすることができます。Web サーバー (IBM HTTP Server など) も必要です。

前提条件:

Web 用インフォメーション・カタログ・センターを AIX Websphere IBM HTTP Web サーバーにインストールする前に、以下のことを行ってください。

- IBM HTTP Web サーバーがインストールされていることを確認します。
- そのサーバーに IBM Websphere Application Server がインストールされていることを確認します。
- Web サーバーのポート番号をチェックします。Web サーバーのポート番号が 80 (通常のデフォルト) 以外である場合は、ドメイン・ネームの構成時に Web アドレスのホスト名にポート番号を付加します。たとえば、http://hostname:portnumber/

手順:

Web 用インフォメーション・カタログ・センターをインストールするには、次のようにします。

1. Web サーバーに、Web 用インフォメーション・カタログ・センターのファイルをインストールします。
 - a. DB2 Universal Database CD-ROM を Web サーバー・ワークステーションの CD-ROM ドライブに挿入します。ランチパッドがオープンします。
 - b. ランチパッドから「インストール (Install)」をクリックします。
 - c. 「製品の選択 (Select Products)」ウィンドウで、「**DB2 管理クライアント (DB2 Administration Client)**」チェック・ボックスを選択します。それから、「次へ (Next)」をクリックします。
 - d. 「インストールのタイプの選択 (Select Type of Install)」ページで、「カスタム (Custom)」を選択します。「次へ (Next)」をクリックします。
 - e. 「**DB2 コンポーネントの選択 (Select DB2 Components)**」ページで、「インフォメーション・カタログ・マネージャー・ツール (Information Catalog Manager tools)」以外のすべてのコンポーネントのチェック・ボックスをクリアします。
 - f. 「サブコンポーネント (Subcomponents)」をクリックします。
 - g. 「サブコンポーネントの選択 (Select Subcomponents)」ページで、「**Web 用インフォメーション・カタログ・センター (Information Catalog Center for the Web)**」のチェック・ボックスが選択されており、他のすべてのチェック・ボックスがクリアされていることを確認します。
 - h. 「**継続 (Continue)**」をクリックします。
 - i. 「**インフォメーション・カタログ・マネージャー・ツール (Information Catalog Manager tools)**」が選択されていることを確認します。
 - j. 「**次へ (Next)**」をクリックして継続し、最後のページで「**終了 (Finish)**」をクリックしてインストールを完了します。
2. Web 用インフォメーション・カタログ・センターを使用する前に、『Web 用インフォメーション・カタログ・センターのインストール後の構成』に示されたステップにしたがってください。

関連概念:

- *DB2 Warehouse Manager* インストール・ガイド の『インフォメーション・カタログ・マネージャー』

関連タスク:

- *DB2 Warehouse Manager* インストール・ガイド の『インフォメーション・カタログ・マネージャーのコンポーネントのインストール』
- *DB2 Warehouse Manager* インストール・ガイド の『インフォメーション・カタログ・センターのインストール』

- 188 ページの『Windows NT Websphere IBM HTTP サーバーへの Web 用インフォメーション・カタログ・センターのインストール』

Web 用インフォメーション・カタログ・センターのインストール後の構成

Web 用インフォメーション・カタログ・センターを構成して、インフォメーション・カタログのユーザーがより簡単にアクセスできるようにすることができます。

前提条件:

Web 用インフォメーション・カタログ・センターを構成する前に、適切にインストールされていることを確認してください。

Web 用インフォメーション・カタログ・センターを使用してインフォメーション・カタログにアクセスする前に、データベース・サーバー上でデータベース・マネージャーが開始されており、Web サーバーが開始されていることを確認してください。

ユーザーに、各自のユーザー ID およびパスワードと、Web 用インフォメーション・カタログ・センターにアクセスするために使用できる Web アドレスを知らせてください。

IBM WebSphere Application Server バージョン 4.01 以上と IBM HTTP Server バージョン 1.3.19 以上が稼働していなければなりません。Web 用インフォメーション・カタログ・センターにアクセスするには、クライアント・ワークステーションに、Javascript および HTML バージョン 3.2 以上が使用可能な Web ブラウザーがインストールされている必要があります。

Windows ユーザーは、Windows 2000 Service Pack 1 以上を使用する必要があります。

AIX ユーザーは AIX バージョン 4.3.3 以上を使用する必要があります。

手順:

Web 用インフォメーション・カタログ・センターを構成するには、以下のようになります。

1. AIX ユーザー: AIX オペレーティング・システムでインフォメーション・カタログ・センターを使用している場合、SQL1224N エラーを受け取る可能性があります。このエラーを受け取った場合、可能な原因のリストのメッセージ・ヘルプ・テキストを参照してください。問題が共有メモリー・セグメントの数に関係している場合は、管理 API 関連資料で、並行アクセスを使用するスレッド化アプリケーションに関する情報を参照してください。この状態の予備手段として、環境変数 EXTSHM の設定、またはアクセスされているデータベースへのループバックの定義などがあります。
2. IBM WebSphere Administrative Server の開始

- Windows の場合:
「スタート」 -> 「プログラム」 -> 「IBM WebSphere」 -> 「Application Server V4.0 AE」 -> 「Admin Server の開始」をクリックします。
 - AIX の場合:
 - a. Websphere Application Server ホーム・ディレクトリーにナビゲートします。例
: cd /usr/WebSphere/AppServer/bin
 - b. startupServer.sh & と入力します。
3. IBM WebSphere Administrative Console の開始
- Windows の場合:
「スタート」 -> 「プログラム」 -> 「IBM WebSphere」 -> 「Application Server V4.0 AE」 -> 「Administrative Console」をクリックします。
 - AIX の場合:
 - a. Websphere Application Server ホーム・ディレクトリーにナビゲートします。例
: cd /usr/WebSphere/AppServer/bin
 - b. adminclient.sh & と入力します。
4. 「WebSphere 管理可能ドメイン (WebSphere Administrative Domain)」を拡張します。
5. 「エンタープライズ・アプリケーション (Enterprise Applications)」で右マウス・ボタン・クリックをして、「エンタープライズ・アプリケーションのインストール (Install Enterprise Application)」をクリックします。「エンタープライズ・アプリケーションのインストール・ウィザード (Install Enterprise Application Wizard)」がオープンします。
6. ノードの名前を「ノードのファイルをブラウズ (Browse for file on node)」フィールドに入力します。「アプリケーションのインストール (*ear) (Install Application (*ear))」が選択されていることを確認します。「パス (Path)」フィールドに、icweb.ear file ファイルの絶対パス名を入力します。icweb を「アプリケーション名 (Application name)」フィールドに入力します。「次へ (Next)」をクリックします。
7. ウィザードの残りのページは、デフォルト設定のままにして、「完了 (Finish)」をクリックして設定を保管します。
8. Web サーバー・プラグイン構成を再生成するには、以下のようになります。
- a. Administrative Console で「WebSphere 管理可能ドメイン (WebSphere Administrative Domain)」を拡張します。
 - b. 「ノード (Nodes)」を拡張します。
 - c. ノードを右マウス・ボタン・クリックして、「Webserver Plugin の再生成 (Regen Webserver Plugin)」を選択します。
9. アプリケーション・クラスパスを更新するには、以下のようになります。

- a. Administrative Console で「**WebSphere 管理可能ドメイン (WebSphere Administrative Domain)**」を拡張します。
- b. 「**ノード (Nodes)**」を拡張します。
- c. ノードを拡張します。
- d. 「**アプリケーション・サーバー (Application Servers)**」をクリックします。プロパティ・パネルが右に表示されます。
- e. 「**JVM 設定 (JVM Settings)**」タブをクリックします。
- f. Windows ユーザーは、次でクラスパスを更新します。
 - C:\Program Files\IBM\SQLLIB\java\db2java.zip
 - C:\Program Files\IBM\SQLLIB\java\Common.jar
 - C:\Program Files\IBM\SQLLIB\java\db2jcc.jar
 - C:\Program Files\IBM\SQLLIB\tools\db2cmn.jar

DB2 を別のディレクトリーにインストールした場合は、パスに必要な変更を加えます。

- g. AIX ユーザーは、次でクラスパスを更新します。
 - /home/db2admin/sqllib/java/db2java.zip
 - /home/db2admin/sqllib/java/Common.jar
 - /home/db2admin/sqllib/java/db2jcc.jar
 - /home/db2admin/sqllib/tools/db2cmn.jar

DB2 を別のディレクトリーにインストールした場合は、パスに必要な変更を加えます。

- h. 「**適用 (Apply)**」をクリックします。
10. ファイル `Webgui.conf` を編集します。Web 用インフォメーション・カタログ・センターからユーザーにアクセスさせたいそれぞれのインフォメーション・カタログごとに、リンクを追加します。それぞれの行を更新し、システムおよびカタログの設定を反映します。以下のフォーマット (例を含む) を使用して、Web 用インフォメーション・カタログ・センターの設定を更新します。
 - `html=/usr/WebSphere/AppServer/installedApps/icweb.ear/icweb.war/html`

Web 用インフォメーション・カタログ・センターの HTML ファイルが保管されているシステム・ディレクトリーです。

- `servlet=http://yournode/icweb/`

IBM WebSphere 内のアプリケーション・セットアップ用の URL です。URL の `yournode` は、ご使用の完全修飾サーバー名になります。

- `databasename=ICMSAMP`

インフォメーション・カタログ・センターのデータベース名です。

- `catalogname=<catalog name>:<description>`

カタログ名には、カタログへのリンクとして使用される説明テキストが続きます。たとえば、

```
catalogname=ICM:Sample 8.1 Information Catalog
```

以下のコマンドを DB2 コマンド・プロンプトに入力して、ヒープ・サイズを更新します。

```
db2 update db cfg for <databasename> using applheapsz 4096
```

11. データベースがリモートの場合は、必要な DB2 クライアント機能を使用することによって、サーバー・ノードおよびデータベースをカタログします。DB2 コマンド行プロセッサから、サーバーがデータベースに正常に接続できることを検証してください。
12. セキュリティーの構成
 - a. Administrative Console で「**コンソール (Console)**」メニュー項目をクリックします。「**セキュリティ・センター... (Security Center...)**」をクリックします。
 - b. 一般ページで「**セキュリティを使用可能にする (Enable Security)**」をチェックします。
 - c. 認証ページをクリックします。認証機構に「**ローカル・オペレーティング・システム (Local Operating System)**」を選択してセキュリティ・サーバー ID およびパスワードを入力します。ローカル・オペレーティング・システム認証は、ローカル・ユーザー・レジストリーを使用します。Web のインフォメーション・カタログ・センターにアクセスできるように、ローカル・オペレーティング・システム・ユーザーを作成する必要があります。もう 1 つのオプションは、LDAP 認証、つまり集中ユーザー・リポジトリーを使用することです。Web のインフォメーション・カタログ・センターにアクセスできるように、LDAP ディレクトリーをセットアップして、LDAP ユーザーを作成する必要があります。
 - d. 「**OK**」をクリックします。ユーザー ID およびパスワードが、ローカル・オペレーティング・システムに対して検証されます。
 - e. Administrative Console で「**エンタープライズ・アプリケーション (Enterprise Applications)**」を拡張します。
 - f. 「icweb」をクリックします。「**ユーザー/役割マッピング (User/Role mappings)**」タブを選択します。デフォルトの **icwebsec** 役割が表示されます。
 - g. 「**選択 (Select)**」をクリックします。「**ユーザー/グループの選択 (Select Users/Groups)**」ウィンドウがオープンします。
 - h. 「**ユーザー/グループの選択 (Select Users/Groups)**」ボックスをチェックします。検索ストリングに (*) を入力して、「**検索 (Search)**」をクリックします。

すべてのユーザーおよびグループが結果にリストされます。インフォメーション・カタログへアクセスさせたいユーザーおよびグループを追加します。「適用 (Apply)」をクリックします。

13. 管理サーバーを再始動して、アプリケーション・サーバーを開始するには、以下のようになります。

サーバーが開始済みの場合は、停止します。

- アプリケーションをインストールしたアプリケーション・サーバーを右マウス・ボタン・クリックして、「停止 (Stop)」をクリックします。

- a. AIX ユーザーのみ: HTTP サーバーを再始動します。

- HTTP サーバーがインストールされているディレクトリーから、以下のコマンドを入力します。

```
apachectl stop
apachectl start
```

- b. 「ノード (Nodes)」を拡張します。ノードを右マウス・ボタン・クリックして、「停止 (Stop)」を選択します。

- c. Administrative Console を開始します。

- Windows ユーザー: 「スタート」 -> 「プログラム」 -> 「IBM WebSphere」 -> 「Application Server V4.0 AE」 -> 「Administrative Console」

- AIX ユーザー: 以下のコマンドを入力します。

```
cd /usr/WebSphere/AppServer/bin
```

(または Websphere Application Server のホーム・ディレクトリー)

```
adminclient.sh &
```

- ユーザー ID およびパスワードを入力するプロンプトが表示されます。

- d. 「WebSphere 管理可能ドメイン (WebSphere Administrative Domain)」を拡張します。

- e. 「ノード (Nodes)」を拡張します。ノードを拡張します。

- f. 「アプリケーション・サーバー (Application Servers)」を拡張します。アプリケーションをインストールしたサーバーを右マウス・ボタン・クリックして、「開始 (Start)」をクリックします。「OK」をクリックして確認します。

- g. Web ブラウザーをオープンして、`http://yournode/icweb` を入力します。`yournode` はご使用のノードの名前です。

ユーザー ID およびパスワードを入力するプロンプトが表示されます。インフォメーション・カタログへの接続に使用したい DB2 ユーザー ID を入力します。

Web ユーザーは、どの Web ブラウザーからでも、`http://yournode/icweb` でインフォメーション・カタログにアクセスできます。`yournode` はご使用のノードの名前です。

Web ユーザーがインフォメーション・カタログに簡単にアクセスできるように、どの

Web ページからでも、この URL にリンクできます。ユーザーには、新規ブラウザをオープンするたびに、有効なユーザー ID およびパスワードを入力するプロンプトが出されます。

関連概念:

- *DB2 Warehouse Manager* インストール・ガイド の『インフォメーション・カタログ・マネージャー』

関連タスク:

- *DB2 Warehouse Manager* インストール・ガイド の『インフォメーション・カタログ・マネージャーのコンポーネントのインストール』
- 188 ページの『Windows NT Websphere IBM HTTP サーバーへの Web 用インフォメーション・カタログ・センターのインストール』
- 216 ページの『AIX Websphere IBM HTTP Web サーバーへの、Web 用インフォメーション・カタログ・センターのインストール』

AIX、Linux、および Solaris オペレーティング環境のウェアハウス・エージェントのインストール

AIX、Linux、Windows、または Solaris オペレーティング環境にウェアハウス・エージェントをインストールすることができます。

前提条件:

AIX または Solaris オペレーティング環境のウェアハウス・エージェントをインストールする前に、そのワークステーションに存在する、以前の AIX または Solaris オペレーティング環境のエージェントを除去する必要があります。

バージョン 7 とバージョン 8 の両方の *DB2 Warehouse Manager* がある場合、デフォルト・エージェントへのソフト・リンクは、使用したいエージェント実行可能プログラムを指している必要があります。バージョン 7 のデフォルト・パスは `/usr/bin/` です。バージョン 8 のデフォルト・パスは `/usr/opt/db2_08_01/bin` です。

AIX では、デフォルト・エージェントは `/usr/opt/db2_08_01/bin/IWH2AGNT` のソフト・リンクが指している、実行可能プログラムです。CLI エージェント `IWH2AGNT.db2cli` または ODBC エージェント `IWH2AGNT.ivodbc` を使用できます。AIX でのデフォルト・エージェントに ODBC エージェントを使用する場合、ODBC エージェントが必要とするドライバー用の *DB2 Warehouse Manager* もインストールする必要があります。

手順:

`db2setup` ユーティリティを使用して、AIX、Linux、または Solaris Operating Environment にウェアハウス・エージェントをインストールします。

ウェアハウス・エージェントをインストールする前に、少なくとも 1 つの DB2 インスタンスを作成および構成する必要があります。

db2setup ユーティリティーを使用して、AIX、Linux、および Solaris Operating Environment のウェアハウス・エージェントをインストールするには、以下のようになります。

1. root 権限があるユーザーとしてログインします。
2. DB2 Warehouse Manager CD-ROM を挿入し、マウントします。
3. `cd /cdrom` と入力することによって、CD-ROM がマウントされているディレクトリに移動します。ここで、`cdrom` は製品 CD-ROM のマウント・ポイントです。
4. 以下のディレクトリの 1 つに変更します。

AIX /cdrom/db2/aix

Linux /cdrom/db2/linux

Solaris /cdrom/unnamed_cdrom/db2/solaris

5. `./db2setup` コマンドを入力します。数分後、DB2 セットアップ (DB2 Setup) ウィザードがオープンします。
6. DB2 ランチパッドがオープンしたら、「製品のインストール (Install Products)」をクリックします。
7. **DB2 Warehouse Manager** が選択されていることを確認し、「次へ (Next)」をクリックします。
8. 「DB2 セットアップ (DB2 Setup)」ウィザードの指示にしたがって進みます。
9. フィーチャーのリストから、「ウェアハウス・エージェント (Warehouse agent)」を選択し、「次へ (Next)」をクリックします。
10. 「DB2 セットアップ (DB2 Setup)」ウィザードの指示にしたがって進みます。残りのステップについてガイドするオンライン・ヘルプが利用できます。
11. ログアウトします。

インストール処理が完了すると、ウェアハウス・エージェントは以下のディレクトリにインストールされます。

- AIX では、`/usr/opt/db2_08_01` ディレクトリ
- Linux および Solaris Operating Environment では、`/opt/IBM/db2/V8.1/` ディレクトリ

インストールが失敗した場合は、IBM ソフトウェア・サポートに連絡してください。

関連概念:

- *DB2 Warehouse Manager* インストール・ガイド の『ウェアハウス・ソースおよびターゲット用の接続製品』

関連タスク:

- 190 ページの『AIX、Linux、および Solaris オペレーティング環境のウェアハウス・エージェント環境の構成』
- *DB2 Warehouse Manager* インストール・ガイド の『DB2 バージョン 8 ウェアハウス・エージェントの除去』

関連資料:

- *DB2 Warehouse Manager* インストール・ガイド の『AIX、Linux、および Solaris オペレーティング環境のウェアハウス・エージェント用のデータベース接続』

z/OS ウェアハウス・エージェントの概要

DB2[®] Warehouse Manager (z/OS[™] 版) には z/OS ウェアハウス・エージェントが組み込まれています。このエージェントを使用して、DB2 Universal Database[™] for z/OS とその他のデータベース間の通信を行うことができます。通信できるデータベースには、他のオペレーティング・システム上の DB2 データベースおよび DB2 以外のデータベースが含まれます。ウェアハウス・エージェントは、Open Database Connectivity (ODBC) 接続を使用する、サポート対象のデータ・ソースと通信することができます。UNIX[®] システム・サービスで実行されるウェアハウス・エージェントは、OS/390[®] バージョン 2 リリース 10 以上を必要とし、DB2 (z/OS および OS/390 版) バージョン 6 および 7 と互換性があります。

z/OS ウェアハウス・エージェントは以下の作業をサポートします。

- データを DB2 データベース・ソースから DB2 データベース・ターゲットにコピーする
- 表またはファイルからサンプル内容を作成する
- ユーザー定義プログラムを実行する
- DB2 Relational Connect を使用して DB2 以外のデータベースをアクセスする
- DB2 Universal Database for z/OS ユーティリティを実行する
- IBM[®] Data Propagator 用のアプライ・ジョブを実行する
- ウェアハウス提供のトランスフォーマーを実行する

関連タスク:

- 197 ページの『z/OS ウェアハウス・エージェントのインストール』
- 202 ページの『z/OS ウェアハウス・エージェント・デーモンの開始』
- 226 ページの『エージェント・デーモンを z/OS 開始済みタスクとして開始する』
- *DB2 Warehouse Manager* インストール・ガイド の『1 つの z/OS サブシステム上で複数のウェアハウス・エージェント・デーモンを実行』

関連資料:

- *DB2 Warehouse Manager* インストール・ガイド の『Trillium ユーザー定義プログラム・ステップ用の z/OS ウェアハウス・エージェント・サポート』

エージェント・デーモンを z/OS 開始済みタスクとして開始する

エージェント・デーモンを開始済みタスクとしてセットアップすると、デーモンを z/OS コンソールから開始することができます。

前提条件:

- プロシージャーと関連付けるユーザー ID を定義する必要があります。このユーザー ID は、STDOUT および STDERR への書き込みと STDENV からの読み取りができる必要があります。
- ユーザー ID とそのグループには、OMVS セグメントを定義する必要があります。

手順:

エージェント・デーモンを z/OS 開始済みタスクとして開始するには、以下のようになります。

1. エージェント・デーモン用の環境変数を含むファイルを作成します。たとえば、環境変数を含む環境ファイルを作成します (export コマンドなしで)。以下に示すように、/bin を PATH 環境変数に追加し、環境変数_BPX_BATCH_SPAWN と _BPX_SHAREAS を追加します。環境変数ファイルには、たとえば /u/USERID/BPXprofile と名前を付け、以下の内容を含みます。

```
STEPLIB=DSN710.SDSNEXIT:DSN710.SDSNLOAD:$STEPLIB
LIBPATH=/usr/lpp/DWC81
PATH=/bin:/usr/lpp/DWC81
CXA_CONFIG=/usr/lpp/DWC81/cxa.ini
VWS_LOGGING=/u/USERID/logs
VWP_LOG=/u/USERID/logs/vwp.log
DSNAOINI=/u/USERID/dsnaoini
IBM_MIXED_MODE_THRESHOLD=0
_BPX_BATCH_SPAWN=YES
_BPX_SHAREAS=NO
```

2. プロシージャー・ライブラリーにプロシージャーを作成します。どのプロシージャー・ライブラリーを使用するかはシステム・プログラマーに相談してください。プロシージャーは BPXBATCH を使用してエージェント・デーモンを開始します。STDENV DD は、ステップ 1 で作成された環境変数ファイルを指します。以下に示すのは、エージェント・デーモンを開始するプロシージャーの例です。

```
/* Start the 390 agent daemon in batch
//*****
//VWDPROC EXEC PGM=BPXBATCH,PARM='pgm /usr/lpp/DWC81/vwd',
//          REGION=8M
//STDOUT DD PATH='/tmp/stdout',
//          PATHOPTS=(OWRONLY,OCREAT),
//          PATHMODE=(SIRWXU,SIRWXG,SIRWXO)
//STDERR DD PATH='/tmp/stderr',
```

```
//          PATHOPTS=(OWRONLY,OCREAT),  
//          PATHMODE=(SIRWXU,SIRWXG,SIRWXO)  
//STDENV   DD  PATH='/u/USERID/BPXprofile',  
//          PATHOPTS=ORDONLY
```

コンソールからエージェント・デーモンを開始するには、S vwdproc.vwd001 と入力します。ここで、vwdproc は作成したプロシージャの名前、.vwd001 は開始済みのタスクをキャンセルするのに使用できる ID です。ID は、8 文字までで、先頭文字は英文字でなければなりません。ID は、開始済みタスクとは別の名前ではなくてはなりません。

コンソールからエージェント・デーモンを停止するには、C vwd001 と入力します。ここで、vwd001 は使用した ID です。

関連概念:

- *DB2 Warehouse Manager* インストール・ガイド の『z/OS および OS/390 版 DB2 表およびフラット・ファイルのサンプルの内容』
- 225 ページの『z/OS ウェアハウス・エージェントの概要』

関連タスク:

- 197 ページの『z/OS ウェアハウス・エージェントのインストール』
- *DB2 Warehouse Manager* インストール・ガイド の『DB2 サブシステムでの Java スタートアップ・プロシージャの、ウェアハウス・トランスフォーマー用セットアップ』
- *DB2 Warehouse Manager* インストール・ガイド の『DB2 (z/OS 版) ユーティリティの実行』
- *DB2 Warehouse Manager* インストール・ガイド の『1 つの z/OS サブシステム上で複数のウェアハウス・エージェント・デーモンを実行』

DB2 チュートリアル

DB2[®] チュートリアルは、DB2 Universal Database のさまざまな機能について学習するのに支援します。これらのチュートリアルでは、アプリケーションの開発、SQL 照会のパフォーマンス調整、データウェアハウスの処理、メタデータの管理、および DB2 を使用した Web サービスの開発の各分野で、段階的なレッスンが用意されています。

はじめに:

以下に示すリンクを使用してチュートリアルにアクセスする前に、*DB2 HTML* ドキュメンテーション CD-ROM からチュートリアルをインストールする必要があります。

チュートリアルをインストールしたくない場合は、*DB2 HTML* ドキュメンテーション CD から直接 HTML 版のチュートリアルを参照することができます。PDF 版のチュートリアルは、*DB2 PDF* ドキュメンテーション CD から利用できます。

チュートリアルの中で、サンプル・データまたはサンプル・コードを使用する場合があります。個々のタスクの前提条件については、それぞれのチュートリアルを参照してください。

DB2 Universal Database チュートリアル:

DB2 HTML ドキュメンテーション CD-ROM からチュートリアルをインストールした場合、以下に示すチュートリアルのタイトルをクリックすると、チュートリアルを表示できます。

ビジネス・インテリジェンス・チュートリアル: データウェアハウス・センターの紹介
データウェアハウス・センターを使用して簡単なデータウェアハウジング・タスクを実行します。

ビジネス・インテリジェンス・チュートリアル: データウェアハウジングの上級者向けガイド
データウェアハウス・センターを使用して高度なデータウェアハウジング・タスクを実行します。(CD には収録されていません。このチュートリアルは、Business Intelligence Solutions Web サイト <http://www.ibm.com/software/data/bi/> のダウンロード・セクションからダウンロードできます。)

Microsoft® Visual Basic を使用しての *Video Online* 用チュートリアル

Development Center Add-in for Microsoft Visual Basic を使用して、アプリケーションのさまざまなコンポーネントを構築します。

インフォメーション・カタログ・センター チュートリアル

インフォメーション・カタログを作成および管理して、インフォメーション・カタログ・センターによってメタデータを位置指定し使用します。

Video Central for e-business チュートリアル

WebSphere® 製品を使用して、高性能な DB2 Web サービス・アプリケーションを開発してデプロイします。

Visual Explain チュートリアル

Visual Explain を使用して、パフォーマンスを向上させるために SQL ステートメントを分析し、最適化し、調整します。

AIX、Linux および Solaris オペレーティング環境用サンプル IWH.environment ファイル

IWH.environment ファイルには、ウェアハウス・エージェント環境を構成する時に設定する必要がある環境変数が含まれます。

```
#!/bin/bash
#####
#
# Licensed Materials - Property of IBM
#
# 5648-B90
# (C) COPYRIGHT International Business Machines Corp. 1993, 2002
```



```

#
# 5648-B91
# (C) COPYRIGHT International Business Machines Corp. 1993, 2002
#
# 5648-B95
# (C) COPYRIGHT International Business Machines Corp. 1993, 2002
#
# 5648-B97
# (C) COPYRIGHT International Business Machines Corp. 1993, 2002
#
# 5648-B99
# (C) COPYRIGHT International Business Machines Corp. 1993, 2002
#
# All Rights Reserved
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
#####
#
# NAME: IWH.environment
#
#####

DB2DIR="/usr/opt/db2_08_01"

# Data Warehouse environment setup script.
# Please Update this script to setup environment variables for other databases.

# NOTE :
# After adding new setup, the daemon process named : vwdaemon should
# be terminated using command: "kill <PID>" for changes to
# take effect.
# DO NOT EXECUTE vwdaemon. It will be re-executed immediately by init process.

# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! ATTENTION !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! ATTENTION !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# !      START OF MUST SET ENVIRONMENT VARIABLES SECTION      !
# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! ATTENTION !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! ATTENTION !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

# DB2 environment section. Please set DB2 instance id
DB2INSTANCE=db2inst1 # make sure this is correct!

# These are the defaults. To change any of them, uncomment it and make
# the change. The default values are defined in the IWH.startup file.
#LANG=en_US
#VWS_LOGGING=/var/IWH
#VWA_LOCALE_TABLE=8859-1
#LC_ALL=en_US
#DB2CODEPAGE=819
#VWSPATH=${DB2DIR?}
#VWS_TEMPLATES=${VWSPATH?}/templates

# If you are using the Data Warehouse Center publish to Information
# Catalog Manager function or the Connector for the Web, you must

```

```

# update the Classpath and Path variables. In this example,
# db2inst is the name of the DB2 instance.
#CLASSPATH=./home/db2inst/sqllib/java/db2java.zip:/home/db2inst/
sqllib/java/sqlj.zip:/home/db2inst/sqllib/java/runtime.zip:/home/
db2inst/sqllib/tools/ibmjsse.jar:/home/db2inst/sqllib/tools/
db2_vw_web.jar:/home/db2inst/sqllib/java/Common.jar:/home/db2inst
/sqllib/tools/db2_vw.jar:/home/db2inst/sqllib/tools/db2vwcom.jar:/
home/db2inst/sqllib/tools/db2vwddd.jar:/home/db2inst/sqllib/tools/
db2cmn.jar:/home/db2inst/sqllib/java/db2jcc.jar:
#export CLASSPATH

#PATH=./usr/bin:/usr/bins/etc:/usr/sbin:/usr/ucb:/usr/bin/
X11:/sbin:/home/db2inst/sqllib/bin:

#export PATH

# If you are running a Java application using JDBC and the AIX
# warehouse agent to access a DB2 AIX server, and you may get
# the SQL1224N error message when attempting to start another
# warehouse agent. If you receive this error, look at the message
# help text for a list of possible causes. If the problem involves
# the number of shared memory segments, see the Administrative API
# reference documentation for information regarding threaded
# applications with concurrent access, which presents workarounds
# for this situation including setting the environment variable
# EXTSHM or defining a loop-back to the database being accessed.

#EXTSHM=ON
#export EXTSHM

#The following setting is needed to run the warehouse agent on Linux.
#The .bashrc file is the configuration file for the "bash" shell.
#export LD_LIBRARY_PATH=./home/db2inst8/sqllib/lib, into .bashrc file

# If using InterSolv odbc drivers, set the IS_ODBC
# environment variable to the path for Intersolv odbc
# directory.
#
# IS_ODBC=<.../intersolv/.../odbc> # Should have path for Intersolv odbc dir

# For essbase udp, set the following path. HOME must be set to a directory
# to which the agent has write authority:

# ARBORPATH=<.../essbase>
# export ARBORPATH

# For Replication using the Data Warehouse agent on any Unix type
# system (AIX, Sun, and Linux) you need to set the HOME variable in
# the IWH.environment file in the /sqllib/bin directory using the
# current userid. This is a requirement of the Replication Apply program.

# HOME=/home/<userid>

# If using Oracle InterSolv driver, set the following paths:
#

```

```

# ORACLE_HOME=<... your Oracle client HOME dir ...>
# TNS_ADMIN=<... your TNS_ADMIN dir ...>
# ORACLE_SID=<... your ORACLE_SID ...>
# export ORACLE_HOME TNS_ADMIN ORACLE_SID

# If using Sybase InterSolv driver, set the following paths:
#
# SYBASE=<... your Sybase client dir ...>
# DSQUERY=<... your DSQUERY value ...>
# export SYBASE DSQUERY

# If using Informix InterSolv driver, set the following paths:
#
# INFORMIXDIR=<... your Informix client dir ...>
# INFORMIXSERVER=<... your Informix Server ...>
# export INFORMIXDIR INFORMIXSERVER

# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# !           END OF MUST SET ENVIRONMENT VARIABLES SECTION       !
# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

# For the following, uncomment out the appropriate section(s)

# if [ "${INFORMIXDIR}" != "" ]; then
#   PATH=${INFORMIXDIR?}/bin:${PATH?}
#   LIBPATH=${LIBPATH:-""}:${INFORMIXDIR?}/lib
# else
#   INFORMIXDIR=${IS_ODBC?}
#   export INFORMIXDIR
# fi
#
# if [ "${SYBASE}" != "" ]; then
#   PATH=${SYBASE?}/bin:${PATH?}
#   LIBPATH=${LIBPATH:-""}:${SYBASE?}/lib
# fi
#
# if [ "${ORACLE_HOME}" != "" ]; then
#   PATH=${ORACLE_HOME?}/bin:${PATH?}
#   LIBPATH=${LIBPATH:-""}:${ORACLE_HOME?}/lib
# fi
#
# if [ "${ARBORPATH}" != "" ]; then
#   PATH=${PATH}:${ARBORPATH?}/bin
#   LIBPATH=${LIBPATH:-""}:${ARBORPATH?}/api/lib
# fi
#
# if [ "${IS_ODBC}" != "" ]; then
#   # for intersolv odbc
#   LIBPATH=${LIBPATH:-""}:${IS_ODBC?}/lib
#   IV_GLS_LCDIR=${IS_ODBC?}/gls/lc11

```

```
| # IV_GLS_REGISTRY=${IS_ODBC?}/gls/cm3/registry  
| # export IV_GLS_LCDIR IV_GLS_REGISTRY  
| # fi  
|
```

| ステップ完了の z/OS コンソール通知の設定

| ステップの完了時に、z/OS コンソールへの通知が必要な場合は、sendWTO プログラム
| を使用します。このプログラムは、任意のストリングを入力として使用し、コンソール
| に WTO メッセージを発行します。このプログラムは、z/OS エージェントでのみ提供
| され、z/OS または OS/390 上の UNIX システム・サービス (USS) の下で実行されま
| す。どのプラットフォームで実行されているエージェントからでも、ユーザー定義のプ
| ログラム VWPMVS を使用して、sendWTO を指定した BPXBATCH を実行する JCL
| をサブミットすることにより、sendWTO を呼び出すことができます。ただし、
| sendWTO は z/OS または OS/390 上で実行する必要があります。

| 手順:

| USS シェルから sendWTO プログラムを実行するには、コマンド `sendWTO 'wto`
| `string'` を発行します。sendWTO プログラムは、その他のユーザー定義プログラムと
| 同じように、ユーザー定義プログラムとして定義することができます。

| 関連概念:

- | • 203 ページの『DB2 Warehouse Manager ユーザー定義プログラム』

| 関連タスク:

- | • 204 ページの『トリガー・プログラムによるウェアハウス・ステップのスケジューリ
| ング (XTClient)』

データベース管理

管理ガイド

ユーロを使用可能なコード・ページ遷移表ファイル

以下の表は、ユーロ通貨記号をサポートするように拡張された変換表をリストしています。ユーロ記号サポートを使用不可にしたい場合、「遷移表ファイル」欄に示された遷移表ファイルをダウンロードしてください。

アラビア語:

データベース・サーバー CCSID/CPGID	データベース・クライアント CCSID/CPGID	遷移表ファイル
420, 16804	1046, 9238	04201046.cnv, IBM00420.ucs
420, 16804	1256, 5352	04201256.cnv, IBM00420.ucs
420, 16804	1200, 1208, 13488, 17584	IBM00420.ucs
864, 17248	1046, 9238	08641046.cnv, 10460864.cnv, IBM00864.ucs
864, 17248	1256, 5352	08641256.cnv, 12560864.cnv, IBM00864.ucs
864, 17248	1200, 1208, 13488, 17584	IBM00864.ucs
1046, 9238	864, 17248	10460864.cnv, 08641046.cnv, IBM01046.ucs
1046, 9238	1089	10461089.cnv, 10891046.cnv, IBM01046.ucs
1046, 9238	1256, 5352	10461256.cnv, 12561046.cnv, IBM01046.ucs
1046, 9238	1200, 1208, 13488, 17584	IBM01046.ucs
1089	1046, 9238	10891046.cnv, 10461089.cnv
1256, 5352	864, 17248	12560864.cnv, 08641256.cnv, IBM01256.ucs
1256, 5352	1046, 9238	12561046.cnv, 10461256.cnv, IBM01256.ucs
1256, 5352	1200, 1208, 13488, 17584	IBM01256.ucs

バルト語:

データベース・サーバー CCSID/CPGID	データベース・クライアント CCSID/CPGID	遷移表ファイル
921、901	1257	09211257.cnv、12570921.cnv、 IBM00921.ucs
921、901	1200、1208、13488、17584	IBM00921.ucs
1112、1156	1257、5353	11121257.cnv
1257、5353	921、901	12570921.cnv、09211257.cnv、 IBM01257.ucs
1257、5353	922、902	12570922.cnv、09221257.cnv、 IBM01257.ucs
1257、5353	1200、1208、13488、17584	IBM01257.ucs

ベラルーシ:

データベース・サーバー CCSID/CPGID	データベース・クライアント CCSID/CPGID	遷移表ファイル
1131、849	1251、5347	11311251.cnv、12511131.cnv
1131、849	1283	11311283.cnv

キリル文字:

データベース・サーバー CCSID/CPGID	データベース・クライアント CCSID/CPGID	遷移表ファイル
855、872	866、808	08550866.cnv、08660855.cnv
855、872	1251、5347	08551251.cnv、12510855.cnv
866、808	855、872	08660855.cnv、08550866.cnv
866、808	1251、5347	08661251.cnv、12510866.cnv
1025、1154	855、872	10250855.cnv、IBM01025.ucs
1025、1154	866、808	10250866.cnv、IBM01025.ucs
1025、1154	1131、849	10251131.cnv、IBM01025.ucs
1025、1154	1251、5347	10251251.cnv、IBM01025.ucs
1025、1154	1200、1208、13488、17584	IBM01025.ucs
1251、5347	855、872	12510855.cnv、08551251.cnv、 IBM01251.ucs

データベース・サーバー CCSID/CPGID	データベース・クライアント CCSID/CPGID	遷移表ファイル
1251、5347	866、808	12510866.cnv、08661251.cnv、 IBM01251.ucs
1251、5347	1124	12511124.cnv、11241251.cnv、 IBM01251.ucs
1251、5347	1125、848	12511125.cnv、11251251.cnv、 IBM01251.ucs
1251、5347	1131、849	12511131.cnv、11311251.cnv、 IBM01251.ucs
1251、5347	1200、1208、13488、17584	IBM01251.ucs

エストニア語:

データベース・サーバー CCSID/CPGID	データベース・クライアント CCSID/CPGID	遷移表ファイル
922、902	1257	09221257.cnv、12570922.cnv、 IBM00922.ucs
922、902	1200、1208、13488、17584	IBM00922.ucs
1122、1157	1257、5353	11221257.cnv

ギリシャ語:

データベース・サーバー CCSID/CPGID	データベース・クライアント CCSID/CPGID	遷移表ファイル
423	869、9061	04230869.cnv
813、4909	869、9061	08130869.cnv、08690813.cnv、 IBM00813.ucs
813、4909	1253、5349	08131253.cnv、12530813.cnv、 IBM00813.ucs
813、4909	1200、1208、13488、17584	IBM00813.ucs
869、9061	813、4909	08690813.cnv、08130869.cnv
869、9061	1253、5349	08691253.cnv、12530869.cnv
875、4971	813、4909	08750813.cnv、IBM00875.ucs
875、4971	1253、5349	08751253.cnv、IBM00875.ucs
875、4971	1200、1208、13488、17584	IBM00875.ucs

データベース・サーバー CCSID/CPGID	データベース・クライアント CCSID/CPGID	遷移表ファイル
1253、5349	813、4909	12530813.cnv、08131253.cnv、 IBM01253.ucs
1253、5349	869、9061	12530869.cnv、08691253.cnv、 IBM01253.ucs
1253、5349	1200、1208、13488、17584	IBM01253.ucs

ヘブライ語:

データベース・サーバー CCSID/CPGID	データベース・クライアント CCSID/CPGID	遷移表ファイル
424、12712	856、9048	04240856.cnv、IBM00424.ucs
424、12712	862、867	04240862.cnv、IBM00424.ucs
424、12712	916	04240916.cnv、IBM00424.ucs
424、12712	1255、5351	04241255.cnv、IBM00424.ucs
424、12712	1200、1208、13488、17584	IBM00424.ucs
856、9048	862、867	08560862.cnv、08620856.cnv、 IBM0856.ucs
856、9048	916	08560916.cnv、09160856.cnv、 IBM0856.ucs
856、9048	1255、5351	08561255.cnv、12550856.cnv、 IBM0856.ucs
856、9048	1200、1208、13488、17584	IBM0856.ucs
862、867	856、9048	08620856.cnv、08560862.cnv、 IBM00862.ucs
862、867	916	08620916.cnv、09160862.cnv、 IBM00862.ucs
862、867	1255、5351	08621255.cnv、12550862.cnv、 IBM00862.ucs
862、867	1200、1208、13488、17584	IBM00862.ucs
916	856、9048	09160856.cnv、08560916.cnv
916	862、867	09160862.cnv、08620916.cnv
1255、5351	856、9048	12550856.cnv、08561255.cnv、 IBM01255.ucs
1255、5351	862、867	12550862.cnv、08621255.cnv、 IBM01255.ucs

データベース・サーバー CCSID/CPGID	データベース・クライアント CCSID/CPGID	遷移表ファイル
1255、5351	1200、1208、13488、17584	IBM01255.ucs

日本語:

データベース・サーバー CCSID/CPGID	データベース・クライアント CCSID/CPGID	遷移表ファイル
290、8482	850、858	02900850.cnv
1027、5123	850、858	10270850.cnv

Latin-1:

データベース・サーバー CCSID/CPGID	データベース・クライアント CCSID/CPGID	遷移表ファイル
37、1140	437	00370437.cnv、 IBM00037.ucs
37、1140	850、858	00370850.cnv、 IBM00037.ucs
37、1140	860	00370860.cnv、 IBM00037.ucs
37、1140	1051	00371051.cnv、 IBM00037.ucs
37、1140	1252、5348	00371252.cnv、 IBM00037.ucs
37、1140	1275	00371275.cnv、 IBM00037.ucs
37、1140	1200、1208、13488、17584	IBM00037.ucs
273、1141	437	02730437.cnv、 IBM00273.ucs
273、1141	850、858	02730850.cnv、 IBM00273.ucs
273、1141	1051	02731051.cnv、 IBM00273.ucs
273、1141	1252、5348	02731252.cnv、 IBM00273.ucs
273、1141	1275	02731275.cnv、 IBM00273.ucs
273、1141	1200、1208、13488、17584	IBM00273.ucs
277、1142	437	02770437.cnv、 IBM00277.ucs
277、1142	850、858	02770850.cnv、 IBM00277.ucs
277、1142	1051	02771051.cnv、 IBM00277.ucs
277、1142	1252、5348	02771252.cnv、 IBM00277.ucs
277、1142	1275	02771275.cnv、 IBM00277.ucs
277、1142	1200、1208、13488、17584	IBM00277.ucs
278、1143	437	02780437.cnv、 IBM00278.ucs
278、1143	850、858	02780850.cnv、 IBM00278.ucs

データベース・サーバー CCSID/CPGID	データベース・クライアント CCSID/CPGID	遷移表ファイル
278、1143	1051	02781051.cnv、IBM00278.ucs
278、1143	1252、5348	02781252.cnv、IBM00278.ucs
278、1143	1275	02781275.cnv、IBM00278.ucs
278、1143	1200、1208、13488、17584	IBM00278.ucs
280、1144	437	02800437.cnv、IBM00280.ucs
280、1144	850、858	02800850.cnv、IBM00280.ucs
280、1144	1051	02801051.cnv、IBM00280.ucs
280、1144	1252、5348	02801252.cnv、IBM00280.ucs
280、1144	1275	02801275.cnv、IBM00280.ucs
280、1144	1200、1208、13488、17584	IBM00280.ucs
284、1145	437	02840437.cnv、IBM00284.ucs
284、1145	850、858	02840850.cnv、IBM00284.ucs
284、1145	1051	02841051.cnv、IBM00284.ucs
284、1145	1252、5348	02841252.cnv、IBM00284.ucs
284、1145	1275	02841275.cnv、IBM00284.ucs
284、1145	1200、1208、13488、17584	IBM00284.ucs
285、1146	437	02850437.cnv、IBM00285.ucs
285、1146	850、858	02850850.cnv、IBM00285.ucs
285、1146	1051	02851051.cnv、IBM00285.ucs
285、1146	1252、5348	02851252.cnv、IBM00285.ucs
285、1146	1275	02851275.cnv、IBM00285.ucs
285、1146	1200、1208、13488、17584	IBM00285.ucs
297、1147	437	02970437.cnv、IBM00297.ucs
297、1147	850、858	02970850.cnv、IBM00297.ucs
297、1147	1051	02971051.cnv、IBM00297.ucs
297、1147	1252、5348	02971252.cnv、IBM00297.ucs
297、1147	1275	02971275.cnv、IBM00297.ucs
297、1147	1200、1208、13488、17584	IBM00297.ucs
437	850、858	04370850.cnv、08500437.cnv
500、1148	437	05000437.cnv、IBM00500.ucs
500、1148	850、858	05000850.cnv、IBM00500.ucs
500、1148	857、9049	05000857.cnv、IBM00500.ucs

データベース・サーバー CCSID/CPGID	データベース・クライアント CCSID/CPGID	遷移表ファイル
500, 1148	920	05000920.cnv, IBM00500.ucs
500, 1148	1051	05001051.cnv, IBM00500.ucs
500, 1148	1114, 5210	05001114.cnv, IBM00500.ucs
500, 1148	1252, 5348	05001252.cnv, IBM00500.ucs
500, 1148	1254, 5350	05001254.cnv, IBM00500.ucs
500, 1148	1275	05001275.cnv, IBM00500.ucs
500, 1148	1200, 1208, 13488, 17584	IBM00500.ucs
850, 858	437	08500437.cnv, 04370850.cnv
850, 858	860	08500860.cnv, 08600850.cnv
850, 858	1114, 5210	08501114.cnv, 11140850.cnv
850, 858	1275	08501275.cnv, 12750850.cnv
860	850, 858	08600850.cnv, 08500860.cnv
871, 1149	437	08710437.cnv, IBM00871.ucs
871, 1149	850, 858	08710850.cnv, IBM00871.ucs
871, 1149	1051	08711051.cnv, IBM00871.ucs
871, 1149	1252, 5348	08711252.cnv, IBM00871.ucs
871, 1149	1275	08711275.cnv, IBM00871.ucs
871, 1149	1200, 1208, 13488, 17584	IBM00871.ucs
1275	850, 858	12750850.cnv, 08501275.cnv

Latin-2:

データベース・サーバー CCSID/CPGID	データベース・クライアント CCSID/CPGID	遷移表ファイル
852, 9044	1250, 5346	08521250.cnv, 12500852.cnv
870, 1153	852, 9044	08700852.cnv, IBM00870.ucs
870, 1153	1250, 5346	08701250.cnv, IBM00870.ucs
870, 1153	1200, 1208, 13488, 17584	IBM00870.ucs
1250, 5346	852, 9044	12500852.cnv, 08521250.cnv, IBM01250.ucs
1250, 5346	1200, 1208, 13488, 17584	IBM01250.ucs

中国語 (簡体字):

データベース・サーバー CCSID/CPGID	データベース・クライアント CCSID/CPGID	遷移表ファイル
837、935、1388	1200、1208、13488、17584	1388ucs2.cnv
1386	1200、1208、13488、17584	1386ucs2.cnv、ucs21386.cnv

中国語 (繁体字):

データベース・サーバー CCSID/CPGID	データベース・クライアント CCSID/CPGID	遷移表ファイル
937、835、1371	950、1370	09370950.cnv、0937ucs2.cnv
937、835、1371	1200、1208、13488、17584	0937ucs2.cnv
1114、5210	850、858	11140850.cnv、08501114.cnv

タイ:

データベース・サーバー CCSID/CPGID	データベース・クライアント CCSID/CPGID	遷移表ファイル
838、1160	874、1161	08380874.cnv、IBM00838.ucs
838、1160	1200、1208、13488、17584	IBM00838.ucs
874、1161	1200、1208、13488、17584	IBM00874.ucs

トルコ語:

データベース・サーバー CCSID/CPGID	データベース・クライアント CCSID/CPGID	遷移表ファイル
857、9049	1254、5350	08571254.cnv、12540857.cnv
1026、1155	857、9049	10260857.cnv、IBM01026.ucs
1026、1155	1254、5350	10261254.cnv、IBM01026.ucs
1026、1155	1200、1208、13488、17584	IBM01026.ucs
1254、5350	857、9049	12540857.cnv、08571254.cnv、 IBM01254.ucs
1254、5350	1200、1208、13488、17584	IBM01254.ucs

ウクライナ:

データベース・サーバー CCSID/CPGID	データベース・クライアント CCSID/CPGID	遷移表ファイル
1123、1158	1124	11231124.cnv
1123、1158	1125、848	11231125.cnv
1123、1158	1251、5347	11231251.cnv
1124	1251、5347	11241251.cnv、12511124.cnv
1125、848	1251、5347	11251251.cnv、12511125.cnv

ユニコード:

データベース・サーバー CCSID/CPGID	データベース・クライアント CCSID/CPGID	遷移表ファイル
1200、1208、13488、 17584	813、4909	IBM00813.ucs
1200、1208、13488、 17584	862、867	IBM00862.ucs
1200、1208、13488、 17584	864、17248	IBM00864.ucs
1200、1208、13488、 17584	874、1161	IBM00874.ucs
1200、1208、13488、 17584	921、901	IBM00921.ucs
1200、1208、13488、 17584	922、902	IBM00922.ucs
1200、1208、13488、 17584	1046、9238	IBM01046.ucs
1200、1208、13488、 17584	1250、5346	IBM01250.ucs
1200、1208、13488、 17584	1251、5347	IBM01251.ucs
1200、1208、13488、 17584	1253、5349	IBM01253.ucs
1200、1208、13488、 17584	1254、5350	IBM01254.ucs
1200、1208、13488、 17584	1255、5351	IBM01255.ucs

データベース・サーバー CCSID/CPGID	データベース・クライアント CCSID/CPGID	遷移表ファイル
1200、1208、13488、 17584	1256、5352	IBM01256.ucs
1200、1208、13488、 17584	1386	ucs21386.cnv、1386ucs2.cnv

ベトナム語:

データベース・サーバー CCSID/CPGID	データベース・クライアント CCSID/CPGID	遷移表ファイル
1130、1164	1258、5354	11301258.cnv
1258、5354	1129、1163	12581129.cnv

関連概念:

- *SQL* リファレンス 第 1 巻 の『文字変換』

関連タスク:

- 管理ガイド: プランニング の『ユーロ記号サポートの使用可能化および使用不可』

サポートされているテリトリー・コードおよびコード・ページ

以下の表は、データベース・サーバーがサポートする言語とコード・セット、およびこれらの値がデータベース・マネージャーで使用されるテリトリー・コードとコード・ページ値にどのようにマップされるかを示したものです。

次に、この表の列を説明します。

- **コード・ページ**は、オペレーティング・システムのコード・セットからマップされる IBM 定義のコード・ページを示します。
- **グループ**は、コード・ページが単一バイト ("S")、2 バイト ("D")、またはニュートラル ("N") のいずれかを示します。「-n」は、文字番号の組み合わせを作成するのに使われる番号です。一致する組み合わせは、DB2 で接続と変換が可能なところを示しています。たとえば、「S-1」グループすべては、一緒に動作できます。ただし、グループがニュートラルの場合、以下にリストするほかのどのようなコード・ページを使用した接続および変換も可能です。
- **コード・セット**は、サポートされる言語に関連するコード・セットを示します。このコード・セットは、DB2 コード・ページにマップされます。
- **テリトリー・コード**は、データベース・マネージャーが地域固有のサポートを提供するために内部で使用するものです。
- **ロケール**は、データベース・マネージャーがサポートするロケール値を示します。

- **オペレーティング・システム** は、言語およびコード・セットをサポートするオペレーティング・システムを示します。

表 39. ユニコード

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
1200	N-1	16 ビット・ユニコード	任意	任意	任意
1208	N-1	ユニコードの UTF-8 エンコード	任意	任意	任意

表 40. アルバニア、テリトリー ID: AL

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
819	S-1	ISO8859-1	355	sq_AL	AIX
850	S-1	IBM-850	355	-	AIX
923	S-1	ISO8859-15	355	sq_AL.8859-15	AIX
1208	N-1	UTF-8	355	SQ_AL	AIX
37	S-1	IBM-37	355	-	ホスト
1140	S-1	IBM-1140	355	-	ホスト
819	S-1	iso88591	355	-	HP-UX
923	S-1	iso885915	355	-	HP-UX
1051	S-1	roman8	355	-	HP-UX
437	S-1	IBM-437	355	-	OS/2
850	S-1	IBM-850	355	-	OS/2
819	S-1	ISO8859-1	355	-	Solaris
923	S-1	ISO8859-15	355	-	Solaris
1252	S-1	1252	355	-	Windows

表 41. アラブ諸国 / 地域、テリトリー ID: AA

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
1046	S-6	IBM-1046	785	Ar_AA	AIX
1089	S-6	ISO8859-6	785	ar_AA	AIX
1208	N-1	UTF-8	785	AR_AA	AIX
420	S-6	IBM-420	785	-	ホスト
425	S-6	IBM-425	785	-	ホスト

表 41. アラブ諸国 / 地域、テリトリー ID: AA (続き)

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
1089	S-6	iso88596	785	ar_SA.iso88596	HP-UX
864	S-6	IBM-864	785	-	OS/2
1256	S-6	1256	785	-	Windows

表 42. オーストラリア、テリトリー ID: AU

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
819	S-1	ISO8859-1	61	en_AU	AIX
850	S-1	IBM-850	61	-	AIX
923	S-1	ISO8859-15	61	en_AU.8859-15	AIX
1208	N-1	UTF-8	61	EN_AU	AIX
37	S-1	IBM-37	61	-	ホスト
1140	S-1	IBM-1140	61	-	ホスト
819	S-1	iso88591	61	-	HP-UX
923	S-1	iso885915	61	-	HP-UX
1051	S-1	roman8	61	-	HP-UX
437	S-1	IBM-437	61	-	OS/2
850	S-1	IBM-850	61	-	OS/2
819	S-1	ISO8859-1	61	en_AU	SCO
819	S-1	ISO8859-1	61	en_AU	Solaris
923	S-1	ISO8859-15	61	-	Solaris
1252	S-1	1252	61	-	Windows

表 43. オーストリア、テリトリー ID: AT

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
819	S-1	ISO8859-1	43	-	AIX
850	S-1	IBM-850	43	-	AIX
923	S-1	ISO8859-15	43	-	AIX
1208	N-1	UTF-8	43	-	AIX
37	S-1	IBM-37	43	-	ホスト
1140	S-1	IBM-1140	43	-	ホスト
819	S-1	iso88591	43	-	HP-UX
923	S-1	iso885915	43	-	HP-UX
1051	S-1	roman8	43	-	HP-UX
819	S-1	ISO-8859-1	43	de_AT	Linux

表 43. オーストリア、テリトリー ID: AT (続き)

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
923	S-1	ISO-8859-15	43	de_AT@euro	Linux
437	S-1	IBM-437	43	-	OS/2
850	S-1	IBM-850	43	-	OS/2
819	S-1	ISO8859-1	43	de_AT	SCO
819	S-1	ISO8859-1	43	de_AT	Solaris
923	S-1	ISO8859-15	43	de_AT.ISO8859-15	Solaris
1252	S-1	1252	43	-	Windows

表 44. ベラルーシ、テリトリー ID: BY

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
915	S-5	ISO8859-5	375	be_BY	AIX
1208	N-1	UTF-8	375	BE_BY	AIX
1025	S-5	IBM-1025	375	-	ホスト
1154	S-5	IBM-1154	375	-	ホスト
915	S-5	ISO8859-5	375	-	OS/2
1131	S-5	IBM-1131	375	-	OS/2
1251	S-5	1251	375	-	Windows

表 45. ベルギー、テリトリー ID: BE

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
819	S-1	ISO8859-1	32	fr_BE	AIX
819	S-1	ISO8859-1	32	nl_BE	AIX
850	S-1	IBM-850	32	Fr_BE	AIX
850	S-1	IBM-850	32	Nl_BE	AIX
923	S-1	ISO8859-15	32	fr_BE.8859-15	AIX
923	S-1	ISO8859-15	32	nl_BE.8859-15	AIX
1208	N-1	UTF-8	32	FR_BE	AIX
1208	N-1	UTF-8	32	NL_BE	AIX
274	S-1	IBM-274	32	-	ホスト
500	S-1	IBM-500	32	-	ホスト
1148	S-1	IBM-1148	32	-	ホスト
819	S-1	iso88591	32	-	HP-UX
923	S-1	iso885915	32	-	HP-UX
819	S-1	ISO-8859-1	32	fr_BE	Linux

表 45. ベルギー、テリトリー ID: BE (続き)

コード・ ページ	グループ	コード・ セット	テリトリー・ コード	ロケール	オペレー ティング・ システム
819	S-1	ISO-8859-1	32	nl_BE	Linux
923	S-1	ISO-8859-15	32	fr_BE@euro	Linux
923	S-1	ISO-8859-15	32	nl_BE@euro	Linux
437	S-1	IBM-437	32	-	OS/2
850	S-1	IBM-850	32	-	OS/2
819	S-1	ISO8859-1	32	fr_BE	SCO
819	S-1	ISO8859-1	32	nl_BE	SCO
819	S-1	ISO8859-1	32	fr_BE	Solaris
819	S-1	ISO8859-1	32	nl_BE	Solaris
923	S-1	ISO8859-15	32	fr_BE.ISO8859-15	Solaris
923	S-1	ISO8859-15	32	nl_BE.ISO8859-15	Solaris
1252	S-1	1252	32	-	Windows

表 46. ブルガリア、テリトリー ID: BG

コード・ ページ	グループ	コード・ セット	テリトリー・ コード	ロケール	オペレー ティング・ システム
915	S-5	ISO8859-5	359	bg_BG	AIX
1208	N-1	UTF-8	359	BG_BG	AIX
1025	S-5	IBM-1025	359	-	ホスト
1154	S-5	IBM-1154	359	-	ホスト
915	S-5	iso88595	359	bg_BG.iso88595	HP-UX
855	S-5	IBM-855	359	-	OS/2
915	S-5	ISO8859-5	359	-	OS/2
1251	S-5	1251	359	-	Windows

表 47. ブラジル、テリトリー ID: BR

コード・ ページ	グループ	コード・ セット	テリトリー・ コード	ロケール	オペレー ティング・ システム
819	S-1	ISO8859-1	55	pt_BR	AIX
850	S-1	IBM-850	55	-	AIX
923	S-1	ISO8859-15	55	pt_BR.8859-15	AIX
1208	N-1	UTF-8	55	PT_BR	AIX
37	S-1	IBM-37	55	-	ホスト
1140	S-1	IBM-1140	55	-	ホスト
819	S-1	ISO8859-1	55	-	HP-UX
923	S-1	ISO8859-15	55	-	HP-UX

表 47. ブラジル、テリトリー ID: BR (続き)

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
819	S-1	ISO-8859-1	55	pt_BR	Linux
923	S-1	ISO-8859-15	55	-	Linux
850	S-1	IBM-850	55	-	OS/2
819	S-1	ISO8859-1	55	pt_BR	SCO
819	S-1	ISO8859-1	55	pt_BR	Solaris
923	S-1	ISO8859-15	55	-	Solaris
1252	S-1	1252	55	-	Windows

表 48. カナダ、テリトリー ID: CA

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
819	S-1	ISO8859-1	1	fr_CA	AIX
850	S-1	IBM-850	1	Fr_CA	AIX
923	S-1	ISO8859-15	1	fr_CA.8859-15	AIX
1208	N-1	UTF-8	1	FR_CA	AIX
37	S-1	IBM-37	1	-	ホスト
1140	S-1	IBM-1140	1	-	ホスト
819	S-1	iso88591	1	fr_CA.iso88591	HP-UX
923	S-1	iso885915	1	-	HP-UX
1051	S-1	roman8	1	fr_CA.roman8	HP-UX
819	S-1	ISO-8859-1	1	en_CA	Linux
923	S-1	ISO-8859-15	1	-	Linux
850	S-1	IBM-850	1	-	OS/2
819	S-1	ISO8859-1	1	en_CA	SCO
819	S-1	ISO8859-1	1	fr_CA	SCO
819	S-1	ISO8859-1	1	en_CA	Solaris
923	S-1	ISO8859-15	1	-	Solaris
1252	S-1	1252	1	-	Windows

表 49. カナダ (フランス語)、テリトリー ID: CA

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
863	S-1	IBM-863	2	-	OS/2

表 50. 中国 (PRC)、テリトリー ID: CN

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
1383	D-4	IBM-eucCN	86	zh_CN	AIX
1386	D-4	GBK	86	Zh_CN.GBK	AIX
1208	N-1	UTF-8	86	ZH_CN	AIX
935	D-4	IBM-935	86	-	ホスト
1388	D-4	IBM-1388	86	-	ホスト
1383	D-4	hp15CN	86	zh_CN.hp15CN	HP-UX
1383	D-4	GBK	86	zh_CN.GBK	Linux
1381	D-4	IBM-1381	86	-	OS/2
1386	D-4	GBK	86	-	OS/2
1383	D-4	eucCN	86	zh_CN	SCO
1383	D-4	eucCN	86	zh_CN.eucCN	SCO
1383	D-4	gb2312	86	zh	Solaris
1208	N-1	UTF-8	86	zh.UTF-8	Solaris
1381	D-4	IBM-1381	86	-	Windows
1386	D-4	GBK	86	-	Windows
1392/5488	D-4		86	-	

1 (268 ページ) を参照してください。

表 51. クロアチア、テリトリー ID: HR

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
912	S-2	ISO8859-2	385	hr_HR	AIX
1208	N-1	UTF-8	385	HR_HR	AIX
870	S-2	IBM-870	385	-	ホスト
1153	S-2	IBM-1153	385	-	ホスト
912	S-2	iso88592	385	hr_HR.iso88592	HP-UX
912	S-2	ISO-8859-2	385	hr_HR	Linux
852	S-2	IBM-852	385	-	OS/2
912	S-2	ISO8859-2	385	hr_HR.ISO8859-2	SCO
1250	S-2	1250	385	-	Windows

表 52. チェコ共和国、テリトリー ID: CZ

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
912	S-2	ISO8859-2	421	cs_CZ	AIX
1208	N-1	UTF-8	421	CS_CZ	AIX

表 52. チェコ共和国、テリトリー ID: CZ (続き)

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
870	S-2	IBM-870	421	-	ホスト
1153	S-2	IBM-1153	421	-	ホスト
912	S-2	iso88592	421	cs_CZ.iso88592	HP-UX
912	S-2	ISO-8859-2	421	cs_CZ	Linux
852	S-2	IBM-852	421	-	OS/2
912	S-2	ISO8859-2	421	cs_CZ.ISO8859-2	SCO
1250	S-2	1250	421	-	Windows

表 53. デンマーク、テリトリー ID: DK

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
819	S-1	ISO8859-1	45	da_DK	AIX
850	S-1	IBM-850	45	Da_DK	AIX
923	S-1	ISO8859-15	45	da_DK.8859-15	AIX
1208	N-1	UTF-8	45	DA_DK	AIX
277	S-1	IBM-277	45	-	ホスト
1142	S-1	IBM-1142	45	-	ホスト
819	S-1	iso88591	45	da_DK.iso88591	HP-UX
923	S-1	iso885915	45	-	HP-UX
1051	S-1	roman8	45	da_DK.roman8	HP-UX
819	S-1	ISO-8859-1	45	da_DK	Linux
923	S-1	ISO-8859-15	45	-	Linux
850	S-1	IBM-850	45	-	OS/2
819	S-1	ISO8859-1	45	da	SCO
819	S-1	ISO8859-1	45	da_DA	SCO
819	S-1	ISO8859-1	45	da_DK	SCO
819	S-1	ISO8859-1	45	da	Solaris
923	S-1	ISO8859-15	45	da.ISO8859-15	Solaris
1252	S-1	1252	45	-	Windows

表 54. エストニア、テリトリー ID: EE

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
922	S-10	IBM-922	372	Et_EE	AIX
1208	N-1	UTF-8	372	ET_EE	AIX
1122	S-10	IBM-1122	372	-	ホスト

表 54. エストニア、テリトリー ID: EE (続き)

コード・ ページ	グループ	コード・ セット	テリトリー・ コード	ローケール	オペレー ティング・ システム
1157	S-10	IBM-1157	372	-	ホスト
922	S-10	IBM-922	372	-	OS/2
1257	S-10	1257	372	-	Windows

表 55. フィンランド、テリトリー ID: FI

コード・ ページ	グループ	コード・ セット	テリトリー・ コード	ローケール	オペレー ティング・ システム
819	S-1	ISO8859-1	358	fi_FI	AIX
850	S-1	IBM-850	358	Fi_FI	AIX
923	S-1	ISO8859-15	358	fi_FI.8859-15	AIX
1208	N-1	UTF-8	358	FI_FI	AIX
278	S-1	IBM-278	358	-	ホスト
1143	S-1	IBM-1143	358	-	ホスト
819	S-1	iso88591	358	fi_FI.iso88591	HP-UX
923	S-1	iso885915	358	-	HP-UX
1051	S-1	roman8	358	fi-FI.roman8	HP-UX
819	S-1	ISO-8859-1	358	fi_FI	Linux
923	S-1	ISO-8859-15	358	fi_FI@euro	Linux
437	S-1	IBM-437	358	-	OS/2
850	S-1	IBM-850	358	-	OS/2
819	S-1	ISO8859-1	358	-	SCO
819	S-1	ISO8859-1	358	fi_FI	SCO
819	S-1	ISO8859-1	358	sv_FI	SCO
819	S-1	ISO8859-1	358	-	Solaris
923	S-1	ISO8859-15	358	fi.ISO8859-15	Solaris
1252	S-1	1252	358	-	Windows

表 56. FYR マケドニア、テリトリー ID: MK

コード・ ページ	グループ	コード・ セット	テリトリー・ コード	ローケール	オペレー ティング・ システム
915	S-5	ISO8859-5	389	mk_MK	AIX
1208	N-1	UTF-8	389	MK_MK	AIX
1025	S-5	IBM-1025	389	-	ホスト
1154	S-5	IBM-1154	389	-	ホスト
915	S-5	iso88595	389	-	HP-UX
855	S-5	IBM-855	389	-	OS/2

表 56. FYR マケドニア、テリトリー ID: MK (続き)

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
915	S-5	ISO8859-5	389	-	OS/2
1251	S-5	1251	389	-	Windows

表 57. フランス、テリトリー ID: FR

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
819	S-1	ISO8859-1	33	fr_FR	AIX
850	S-1	IBM-850	33	Fr_FR	AIX
923	S-1	ISO8859-15	33	fr_FR.8859-15	AIX
1208	N-1	UTF-8	33	FR_FR	AIX
297	S-1	IBM-297	33	-	ホスト
1147	S-1	IBM-1147	33	-	ホスト
819	S-1	iso88591	33	fr_FR.iso88591	HP-UX
923	S-1	iso885915	33	-	HP-UX
1051	S-1	roman8	33	fr_FR.roman8	HP-UX
819	S-1	ISO-8859-1	33	fr_FR	Linux
923	S-1	ISO-8859-15	33	fr_FR@euro	Linux
437	S-1	IBM-437	33	-	OS/2
850	S-1	IBM-850	33	-	OS/2
819	S-1	ISO8859-1	33	-	SCO
819	S-1	ISO8859-1	33	fr_FR	SCO
819	S-1	ISO8859-1	33	-	Solaris
923	S-1	ISO8859-15	33	fr.ISO8859-15	Solaris
1208	N-1	UTF-8	33	fr.UTF-8	Solaris
1252	S-1	1252	33	-	Windows

表 58. ドイツ、テリトリー ID: DE

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
819	S-1	ISO8859-1	49	de_DE	AIX
850	S-1	IBM-850	49	De_DE	AIX
923	S-1	ISO8859-15	49	de_DE.8859-15	AIX
1208	N-1	UTF-8	49	DE_DE	AIX
273	S-1	IBM-273	49	-	ホスト
1141	S-1	IBM-1141	49	-	ホスト
819	S-1	iso88591	49	de_DE.iso88591	HP-UX

表 58. ドイツ、テリトリー ID: DE (続き)

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
923	S-1	iso885915	49	_	HP-UX
1051	S-1	roman8	49	de_DE.roman8	HP-UX
819	S-1	ISO-8859-1	49	de_DE	Linux
923	S-1	ISO-8859-15	49	de_DE@euro	Linux
437	S-1	IBM-437	49	-	OS/2
850	S-1	IBM-850	49	-	OS/2
819	S-1	ISO8859-1	49		SCO
819	S-1	ISO8859-1	49	de_DE	SCO
819	S-1	ISO8859-1	49		Solaris
923	S-1	ISO8859-15	49	de.ISO8859-15	Solaris
1208	N-1	UTF-8	49	de.UTF-8	Solaris
1252	S-1	1252	49	-	Windows

表 59. ギリシャ、テリトリー ID: GR

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
813	S-7	ISO8859-7	30	el_GR	AIX
1208	N-1	UTF-8	30	EL_GR	AIX
423	S-7	IBM-423	30	-	ホスト
875	S-7	IBM-875	30	-	ホスト
813	S-7	iso88597	30	el_GR.iso88597	HP-UX
813	S-7	ISO-8859-7	30	el_GR	Linux
813	S-7	ISO8859-7	30	-	OS/2
869	S-7	IBM-869	30	-	OS/2
813	S-7	ISO8859-7	30	el_GR.ISO8859-7	SCO
737	S-7	737	30	-	Windows
1253	S-7	1253	30	-	Windows

表 60. ハンガリー、テリトリー ID: HU

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
912	S-2	ISO8859-2	36	hu_HU	AIX
1208	N-1	UTF-8	36	HU_HU	AIX
870	S-2	IBM-870	36	-	ホスト
1153	S-2	IBM-1153	36	-	ホスト
912	S-2	iso88592	36	hu_HU.iso88592	HP-UX

表 60. ハンガリー、テリトリー ID: HU (続き)

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
912	S-2	ISO-8859-2	36	hu_HU	Linux
852	S-2	IBM-852	36	-	OS/2
912	S-2	ISO8859-2	36	hu_HU.ISO8859-2	SCO
1250	S-2	1250	36	-	Windows

表 61. アイスランド、テリトリー ID: IS

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
819	S-1	ISO8859-1	354	is_IS	AIX
850	S-1	IBM-850	354	Is_IS	AIX
923	S-1	ISO8859-15	354	is_IS.8859-15	AIX
1208	N-1	UTF-8	354	IS_IS	AIX
871	S-1	IBM-871	354	-	ホスト
1149	S-1	IBM-1149	354	-	ホスト
819	S-1	iso88591	354	is_IS.iso88591	HP-UX
923	S-1	iso885915	354	-	HP-UX
1051	S-1	roman8	354	is_IS.roman8	HP-UX
819	S-1	ISO-8859-1	354	is_IS	Linux
923	S-1	ISO-8859-15	354	-	Linux
850	S-1	IBM-850	354	-	OS/2
819	S-1	ISO8859-1	354	-	SCO
819	S-1	ISO8859-1	354	is_IS	SCO
819	S-1	ISO8859-1	354	-	Solaris
923	S-1	ISO8859-15	354	-	Solaris
1252	S-1	1252	354	-	Windows

表 62. インド、テリトリー ID: IN

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
806	S-13	IBM-806	91	hi_IN	-
1137	S-13	IBM-1137	91	-	ホスト

表 63. インドネシア、テリトリー ID: ID

コード・ ページ	グループ	コード・ セット	テリトリー・ コード	ロケール	オペレー ティング・ システム
1252	S-1	1252	62	-	Windows

表 64. アイルランド、テリトリー ID: IE

コード・ ページ	グループ	コード・ セット	テリトリー・ コード	ロケール	オペレー ティング・ システム
819	S-1	ISO8859-1	353	-	AIX
850	S-1	IBM-850	353	-	AIX
923	S-1	ISO8859-15	353	-	AIX
1208	N-1	UTF-8	353	-	AIX
285	S-1	IBM-285	353	-	ホスト
1146	S-1	IBM-1146	353	-	ホスト
819	S-1	iso88591	353	-	HP-UX
923	S-1	iso885915	353	-	HP-UX
1051	S-1	roman8	353	-	HP-UX
819	S-1	ISO-8859-1	353	en_IE	Linux
923	S-1	ISO-8859-15	353	en_IE@euro	Linux
437	S-1	IBM-437	353	-	OS/2
850	S-1	IBM-850	353	-	OS/2
819	S-1	ISO8859-1	353	en_IE.ISO8859-1	SCO
819	S-1	ISO8859-1	353	en_IE	Solaris
923	S-1	ISO8859-15	353	en_IE.ISO8859-15	Solaris
1252	S-1	1252	353	-	Windows

表 65. イスラエル、テリトリー ID: IL

コード・ ページ	グループ	コード・ セット	テリトリー・ コード	ロケール	オペレー ティング・ システム
856	S-8	IBM-856	972	Iw_IL	AIX
916	S-8	ISO8859-8	972	iw_IL	AIX
1208	N-1	UTF-8	972	HE-IL	AIX
916	S-8	ISO-8859-8	972	iw_IL	Linux
424	S-8	IBM-424	972	-	ホスト
862	S-8	IBM-862	972	-	OS/2
1255	S-8	1255	972	-	Windows

表 66. イタリア、テリトリー ID: IT

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
819	S-1	ISO8859-1	39	it_IT	AIX
850	S-1	IBM-850	39	It_IT	AIX
923	S-1	ISO8859-15	39	it_IT.8859-15	AIX
1208	N-1	UTF-8	39	It_IT	AIX
280	S-1	IBM-280	39	-	ホスト
1144	S-1	IBM-1144	39	-	ホスト
819	S-1	iso88591	39	it_IT.iso88591	HP-UX
923	S-1	iso885915	39	-	HP-UX
1051	S-1	roman8	39	it_IT.roman8	HP-UX
819	S-1	ISO-8859-1	39	it_IT	Linux
923	S-1	ISO-8859-15	39	it_IT@euro	Linux
437	S-1	IBM-437	39	-	OS/2
850	S-1	IBM-850	39	-	OS/2
819	S-1	ISO8859-1	39	-	SCO
819	S-1	ISO8859-1	39	it_IT	SCO
819	S-1	ISO8859-1	39	-	Solaris
923	S-1	ISO8859-15	39	it.ISO8859-15	Solaris
1208	N-1	UTF-8	39	it.UTF-8	Solaris
1252	S-1	1252	39	-	Windows

表 67. 日本、テリトリー ID: JP

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
932	D-1	IBM-932	81	Ja_JP	AIX
943	D-1	IBM-943	81	Ja_JP	AIX
2 (268 ページ) を参照してください。					
954	D-1	IBM-eucJP	81	ja_JP	AIX
1208	N-1	UTF-8	81	JA_JP	AIX
930	D-1	IBM-930	81	-	ホスト
939	D-1	IBM-939	81	-	ホスト
5026	D-1	IBM-5026	81	-	ホスト
5035	D-1	IBM-5035	81	-	ホスト
1390	D-1		81	-	ホスト
1399	D-1		81	-	ホスト
954	D-1	eucJP	81	ja_JP.eucJP	HP-UX
5039	D-1	SJIS	81	ja_JP.SJIS	HP-UX
954	D-1	EUC-JP	81	ja_JP	Linux
932	D-1	IBM-932	81	-	OS/2

表 67. 日本、テリトリー ID: JP (続き)

コード・ ページ	グループ	コード・ セット	テリトリー・ コード	ロケール	オペレー ティング・ システム
942	D-1	IBM-942	81	-	OS/2
943	D-1	IBM-943	81	-	OS/2
954	D-1	eucJP	81	ja	SCO
954	D-1	eucJP	81	ja_JP	SCO
954	D-1	eucJP	81	ja_JP.EUC	SCO
954	D-1	eucJP	81	ja_JP.eucJP	SCO
943	D-1	IBM-943	81	ja_JP.PCK	Solaris
954	D-1	eucJP	81	ja	Solaris
954	D-1	eucJP	81	japanese	Solaris
1208	N-1	UTF-8	81	ja_JP.UTF-8	Solaris
943	D-1	IBM-943	81	-	Windows
1394	D-1		81	-	

3 (268 ページ) を参照してください。

表 68. カザフスタン、テリトリー ID: KZ

コード・ ページ	グループ	コード・ セット	テリトリー・ コード	ロケール	オペレー ティング・ システム
1251	S-5	1251	7	-	Windows

表 69. 韓国、テリトリー ID: KR

コード・ ページ	グループ	コード・ セット	テリトリー・ コード	ロケール	オペレー ティング・ システム
970	D-3	IBM-eucKR	82	ko_KR	AIX
1208	N-1	UTF-8	82	KO_KR	AIX
933	D-3	IBM-933	82	-	ホスト
1364	D-3	IBM-1364	82	-	ホスト
970	D-3	eucKR	82	ko_KR.eucKR	HP-UX
970	D-3	EUC-KR	82	ko_KR	Linux
949	D-3	IBM-949	82	-	OS/2
970	D-3	eucKR	82	ko_KR.eucKR	SGI
970	D-3	5601	82	ko	Solaris
1208	N-1	UTF-8	82	ko.UTF-8	Solaris
1363	D-3	1363	82	-	Windows

表 70. ラテンアメリカ、テリトリー ID: Lat

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
819	S-1	ISO8859-1	3	-	AIX
850	S-1	IBM-850	3	-	AIX
923	S-1	ISO8859-15	3	-	AIX
1208	N-1	UTF-8	3	-	AIX
284	S-1	IBM-284	3	-	ホスト
1145	S-1	IBM-1145	3	-	ホスト
819	S-1	iso88591	3	-	HP-UX
923	S-1	iso885915	3	-	HP-UX
1051	S-1	roman8	3	-	HP-UX
819	S-1	ISO-8859-1	3	-	Linux
923	S-1	ISO-8859-15	3	-	Linux
437	S-1	IBM-437	3	-	OS/2
850	S-1	IBM-850	3	-	OS/2
819	S-1	ISO8859-1	3	-	Solaris
923	S-1	ISO8859-15	3	-	Solaris
1252	S-1	1252	3	-	Windows

表 71. ラトビア、テリトリー ID: LV

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
921	S-10	IBM-921	371	Lv_LV	AIX
1208	N-1	UTF-8	371	LV_LV	AIX
1112	S-10	IBM-1112	371	-	ホスト
1156	S-10	IBM-1156	371	-	ホスト
921	S-10	IBM-921	371	-	OS/2
1257	S-10	1257	371	-	Windows

表 72. リトアニア、テリトリー ID: LT

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
921	S-10	IBM-921	370	Lt_LT	AIX
1208	N-1	UTF-8	370	LT_LT	AIX
1112	S-10	IBM-1112	370	-	ホスト
1156	S-10	IBM-1156	370	-	ホスト
921	S-10	IBM-921	370	-	OS/2
1257	S-10	1257	370	-	Windows

表 73. マレーシア、テリトリー ID: ID

コード・ ページ	グループ	コード・ セット	テリトリー・ コード	ロケール	オペレー ティング・ システム
1252	S-1	1252	60	-	Windows

表 74. オランダ、テリトリー ID: NL

コード・ ページ	グループ	コード・ セット	テリトリー・ コード	ロケール	オペレー ティング・ システム
819	S-1	ISO8859-1	31	nl_NL	AIX
850	S-1	IBM-850	31	Nl_NL	AIX
923	S-1	ISO8859-15	31	nl_NL.8859-15	AIX
1208	N-1	UTF-8	31	NL_NL	AIX
37	S-1	IBM-37	31	-	ホスト
1140	S-1	IBM-1140	31	-	ホスト
819	S-1	iso88591	31	nl_NL.iso88591	HP-UX
923	S-1	iso885915	31	-	HP-UX
1051	S-1	roman8	31	nl_NL.roman8	HP-UX
819	S-1	ISO-8859-1	31	nl_NL	Linux
923	S-1	ISO-8859-15	31	nl_NL@euro	Linux
437	S-1	IBM-437	31	-	OS/2
850	S-1	IBM-850	31	-	OS/2
819	S-1	ISO8859-1	31	nl	SCO
819	S-1	ISO8859-1	31	nl_NL	SCO
819	S-1	ISO8859-1	31	nl	Solaris
923	S-1	ISO8859-15	31	nl.ISO8859-15	Solaris
1252	S-1	1252	31	-	Windows

表 75. ニュージーランド、テリトリー ID: NZ

コード・ ページ	グループ	コード・ セット	テリトリー・ コード	ロケール	オペレー ティング・ システム
819	S-1	ISO8859-1	64	-	AIX
850	S-1	IBM-850	64	-	AIX
923	S-1	ISO8859-15	64	-	AIX
1208	N-1	UTF-8	64	-	AIX
37	S-1	IBM-37	64	-	ホスト
1140	S-1	IBM-1140	64	-	ホスト
819	S-1	ISO8859-1	64	-	HP-UX

表 75. ニュージーランド、テリトリー ID: NZ (続き)

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
923	S-1	ISO8859-15	64	-	HP-UX
850	S-1	IBM-850	64	-	OS/2
819	S-1	ISO8859-1	64	en_NZ	SCO
819	S-1	ISO8859-1	64	en_NZ	Solaris
923	S-1	ISO8859-15	64	-	Solaris
1252	S-1	1252	64	-	Windows

表 76. ノルウェー、テリトリー ID: NO

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
819	S-1	ISO8859-1	47	no_NO	AIX
850	S-1	IBM-850	47	No_NO	AIX
923	S-1	ISO8859-15	47	no_NO.8859-15	AIX
1208	N-1	UTF-8	47	NO_NO	AIX
277	S-1	IBM-277	47	-	ホスト
1142	S-1	IBM-1142	47	-	ホスト
819	S-1	iso88591	47	no_NO.iso88591	HP-UX
923	S-1	iso885915	47	-	HP-UX
1051	S-1	roman8	47	no_NO.roman8	HP-UX
819	S-1	ISO-8859-1	47	no_NO	Linux
923	S-1	ISO-8859-15	47	-	Linux
850	S-1	IBM-850	47	-	OS/2
819	S-1	ISO8859-1	47	no	SCO
819	S-1	ISO8859-1	47	no_NO	SCO
819	S-1	ISO8859-1	47	no	Solaris
923	S-1	ISO8859-15	47	-	Solaris
1252	S-1	1252	47	-	Windows

表 77. ポーランド、テリトリー ID: PL

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
912	S-2	ISO8859-2	48	pl_PL	AIX
1208	N-1	UTF-8	48	PL_PL	AIX
870	S-2	IBM-870	48	-	ホスト
1153	S-2	IBM-1153	48	-	ホスト
912	S-2	iso88592	48	pl_PL.iso88592	HP-UX

表 77. ポーランド、テリトリー ID: PL (続き)

コード・ ページ	グループ	コード・ セット	テリトリー・ コード	ロケール	オペレー ティング・ システム
912	S-2	ISO-8859-2	48	pl_PL	Linux
852	S-2	IBM-852	48	-	OS/2
912	S-2	ISO8859-2	48	pl_PL.ISO8859-2	SCO
1250	S-2	1250	48	-	Windows

表 78. ポルトガル、テリトリー ID: PT

コード・ ページ	グループ	コード・ セット	テリトリー・ コード	ロケール	オペレー ティング・ システム
819	S-1	ISO8859-1	351	pt_PT	AIX
850	S-1	IBM-850	351	Pt_PT	AIX
923	S-1	ISO8859-15	351	pt_PT.8859-15	AIX
1208	N-1	UTF-8	351	PT_PT	AIX
37	S-1	IBM-37	351	-	ホスト
1140	S-1	IBM-1140	351	-	ホスト
819	S-1	iso88591	351	pt_PT.iso88591	HP-UX
923	S-1	iso885915	351	-	HP-UX
1051	S-1	roman8	351	pt_PT.roman8	HP-UX
819	S-1	ISO-8859-1	351	pt_PT	Linux
923	S-1	ISO-8859-15	351	pt_PT@euro	Linux
850	S-1	IBM-850	351	-	OS/2
860	S-1	IBM-860	351	-	OS/2
819	S-1	ISO8859-1	351	pt	SCO
819	S-1	ISO8859-1	351	pt_PT	SCO
819	S-1	ISO8859-1	351	pt	Solaris
923	S-1	ISO8859-15	351	pt.ISO8859-15	Solaris
1252	S-1	1252	351	-	Windows

表 79. ルーマニア、テリトリー ID: RO

コード・ ページ	グループ	コード・ セット	テリトリー・ コード	ロケール	オペレー ティング・ システム
912	S-2	ISO8859-2	40	ro_RO	AIX
1208	N-1	UTF-8	40	RO_RO	AIX
870	S-2	IBM-870	40	-	ホスト
1153	S-2	IBM-1153	40	-	ホスト
912	S-2	iso88592	40	ro_RO.iso88592	HP-UX
912	S-2	ISO-8859-2	40	ro_RO	Linux

表 79. ルーマニア、テリトリー ID: RO (続き)

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
852	S-2	IBM-852	40	-	OS/2
912	S-2	ISO8859-2	40	ro_RO.ISO8859-2	SCO
1250	S-2	1250	40	-	Windows

表 80. ロシア、テリトリー ID: RU

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
915	S-5	ISO8859-5	7	ru_RU	AIX
1208	N-1	UTF-8	7	RU_RU	AIX
1025	S-5	IBM-1025	7	-	ホスト
1154	S-5	IBM-1154	7	-	ホスト
915	S-5	iso88595	7	ru_RU.iso88595	HP-UX
915	S-5	ISO-8859-5	7	ru_RU	Linux
866	S-5	IBM-866	7	-	OS/2
915	S-5	ISO8859-5	7	-	OS/2
915	S-5	ISO8859-5	7	ru_RU.ISO8859-5	SCO
1251	S-5	1251	7	-	Windows

表 81. セルビア/モンテネグロ、テリトリー ID: SP

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
915	S-5	ISO8859-5	381	sr_SP	AIX
1208	N-1	UTF-8	381	SR_SP	AIX
1025	S-5	IBM-1025	381	-	ホスト
1154	S-5	IBM-1154	381	-	ホスト
915	S-5	iso88595	381	-	HP-UX
855	S-5	IBM-855	381	-	OS/2
915	S-5	ISO8859-5	381	-	OS/2
1251	S-5	1251	381	-	Windows

表 82. スロバキア、テリトリー ID: SK

コード・ ページ	グループ	コード・ セット	テリトリー・ コード	ロケール	オペレー ティング・ システム
912	S-2	ISO8859-2	422	sk_SK	AIX
1208	N-1	UTF-8	422	SK_SK	AIX
870	S-2	IBM-870	422	-	ホスト
1153	S-2	IBM-1153	422	-	ホスト
912	S-2	iso88592	422	sk_SK.iso88592	HP-UX
852	S-2	IBM-852	422	-	OS/2
912	S-2	ISO8859-2	422	sk_SK.ISO8859-2	SCO
1250	S-2	1250	422	-	Windows

表 83. スロベニア、テリトリー ID: SI

コード・ ページ	グループ	コード・ セット	テリトリー・ コード	ロケール	オペレー ティング・ システム
912	S-2	ISO8859-2	386	sl_SI	AIX
1208	N-1	UTF-8	386	SL_SI	AIX
870	S-2	IBM-870	386	-	ホスト
1153	S-2	IBM-1153	386	-	ホスト
912	S-2	iso88592	386	sl_SI.iso88592	HP-UX
912	S-2	ISO-8859-2	386	sl_SI	Linux
852	S-2	IBM-852	386	-	OS/2
912	S-2	ISO8859-2	386	sl_SI.ISO8859-2	SCO
1250	S-2	1250	386	-	Windows

表 84. 南アフリカ、テリトリー ID: ZA

コード・ ページ	グループ	コード・ セット	テリトリー・ コード	ロケール	オペレー ティング・ システム
819	S-1	ISO8859-1	27	en_ZA	AIX
850	S-1	IBM-850	27	En_ZA	AIX
923	S-1	ISO8859-15	27	en_ZA.8859-15	AIX
1208	N-1	UTF-8	27	EN_ZA	AIX
285	S-1	IBM-285	27	-	ホスト
1146	S-1	IBM-1146	27	-	ホスト
819	S-1	iso88591	27	-	HP-UX
923	S-1	iso885915	27	-	HP-UX
1051	S-1	roman8	27	-	HP-UX
437	S-1	IBM-437	27	-	OS/2
850	S-1	IBM-850	27	-	OS/2

表 84. 南アフリカ、テリトリー ID: ZA (続き)

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
819	S-1	ISO8859-1	27	en_ZA.ISO8859-1	SCO
819	S-1	ISO8859-1	27	-	Solaris
923	S-1	ISO8859-15	27	-	Solaris
1252	S-1	1252	27	-	Windows

表 85. スペイン、テリトリー ID: ES

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
819	S-1	ISO8859-1	34	es_ES	AIX
850	S-1	IBM-850	34	Es_ES	AIX
923	S-1	ISO8859-15	34	es_ES.8859-15	AIX
1208	N-1	UTF-8	34	ES_ES	AIX
284	S-1	IBM-284	34	-	ホスト
1145	S-1	IBM-1145	34	-	ホスト
819	S-1	iso88591	34	es_ES.iso88591	HP-UX
923	S-1	iso885915	34	-	HP-UX
1051	S-1	roman8	34	es_ES.roman8	HP-UX
819	S-1	ISO-8859-1	34	es_ES	Linux
923	S-1	ISO-8859-15	34	es_ES@euro	Linux
437	S-1	IBM-437	34	-	OS/2
850	S-1	IBM-850	34	-	OS/2
819	S-1	ISO8859-1	34	es	SCO
819	S-1	ISO8859-1	34	es_ES	SCO
819	S-1	ISO8859-1	34	es	Solaris
923	S-1	ISO8859-15	34	es.ISO8859-15	Solaris
1208	N-1	UTF-8	34	es.UTF-8	Solaris
1252	S-1	1252	34	-	Windows

表 86. スペイン (カタロニア語)、テリトリー ID: ES

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
819	S-1	ISO8859-1	34	ca_ES	AIX
850	S-1	IBM-850	34	Ca_ES	AIX
923	S-1	ISO8859-15	34	ca_ES.8859-15	AIX
1208	N-1	UTF-8	34	CA_ES	AIX

表 87. スウェーデン、テリトリー ID: SE

コード・ ページ	グループ	コード・ セット	テリトリー・ コード	ロケール	オペレー ティング・ システム
819	S-1	ISO8859-1	46	sv_SE	AIX
850	S-1	IBM-850	46	Sv_SE	AIX
923	S-1	ISO8859-15	46	sv_SE.8859-15	AIX
1208	N-1	UTF-8	46	SV_SE	AIX
278	S-1	IBM-278	46	-	ホスト
1143	S-1	IBM-1143	46	-	ホスト
819	S-1	iso88591	46	sv_SE.iso88591	HP-UX
923	S-1	iso885915	46	-	HP-UX
1051	S-1	roman8	46	sv_SE.roman8	HP-UX
819	S-1	ISO-8859-1	46	sv_SE	Linux
923	S-1	ISO-8859-15	46	-	Linux
437	S-1	IBM-437	46	-	OS/2
850	S-1	IBM-850	46	-	OS/2
819	S-1	ISO8859-1	46	sv	SCO
819	S-1	ISO8859-1	46	sv_SE	SCO
819	S-1	ISO8859-1	46	sv	Solaris
923	S-1	ISO8859-15	46	sv.ISO8859-15	Solaris
1208	N-1	UTF-8	46	sv.UTF-8	Solaris
1252	S-1	1252	46	-	Windows

表 88. スイス、テリトリー ID: CH

コード・ ページ	グループ	コード・ セット	テリトリー・ コード	ロケール	オペレー ティング・ システム
819	S-1	ISO8859-1	41	de_CH	AIX
850	S-1	IBM-850	41	De_CH	AIX
923	S-1	ISO8859-15	41	de_CH.8859-15	AIX
1208	N-1	UTF-8	41	DE_CH	AIX
500	S-1	IBM-500	41	-	ホスト
1148	S-1	IBM-1148	41	-	ホスト
819	S-1	iso88591	41	-	HP-UX
923	S-1	iso885915	41	-	HP-UX
1051	S-1	roman8	41	-	HP-UX
819	S-1	ISO-8859-1	41	de_CH	Linux
923	S-1	ISO-8859-15	41	-	Linux
437	S-1	IBM-437	41	-	OS/2
850	S-1	IBM-850	41	-	OS/2
819	S-1	ISO8859-1	41	de_CH	SCO
819	S-1	ISO8859-1	41	fr_CH	SCO

表 88. スイス、テリトリー ID: CH (続き)

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
819	S-1	ISO8859-1	41	it_CH	SCO
819	S-1	ISO8859-1	41	de_CH	Solaris
923	S-1	ISO8859-15	41	-	Solaris
1252	S-1	1252	41	-	Windows

表 89. 台湾、テリトリー ID: TW

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
950	D-2	big5	88	Zh_TW	AIX
8 (269 ページ) を参照してください。					
964	D-2	IBM-eucTW	88	zh_TW	AIX
1208	N-1	UTF-8	88	ZH_TW	AIX
937	D-2	IBM-937	88	-	ホスト
1371	D-2	IBM-1371	88	-	ホスト
950	D-2	big5	88	zh_TW.big5	HP-UX
964	D-2	eucTW	88	zh_TW.eucTW	HP-UX
950	D-2	BIG5	88	zh_TW	Linux
938	D-2	IBM-938	88	-	OS/2
948	D-2	IBM-948	88	-	OS/2
950	D-2	big5	88	-	OS/2
950	D-2	big5	88	zh_TW.BIG5	Solaris
964	D-2	cns11643	88	zh_TW	Solaris
1208	N-1	UTF-8	88	zh_TW.UTF-8	Solaris
950	D-2	big5	88	-	Windows
8 (269 ページ) を参照してください。					

表 90. タイ、テリトリー ID: TH

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
874	S-20	TIS620-1	66	th_TH	AIX
1208	N-1	UTF-8	66	TH_TH	AIX
838	S-20	IBM-838	66	-	ホスト
1160	S-20	IBM-1160	66	-	ホスト
874	S-20	tis620	66	th_TH.tis620	HP-UX
874	S-20	TIS620-1	66	-	OS/2
874	S-20	TIS620-1	66	-	Windows

表 91. トルコ、テリトリ ID: TR

コード・ページ	グループ	コード・セット	テリトリ ー・コード	ロケール	オペレー ティング・ システム
920	S-9	ISO8859-9	90	tr_TR	AIX
1208	N-1	UTF-8	90	TR_TR	AIX
1026	S-9	IBM-1026	90	-	ホスト
1155	S-9	IBM-1155	90	-	ホスト
920	S-9	iso88599	90	tr_TR.iso88599	HP-UX
920	S-9	ISO-8859-9	90	tr_TR	Linux
857	S-9	IBM-857	90	-	OS/2
920	S-9	ISO8859-9	90	tr_TR.ISO8859-9	SCO
1254	S-9	1254	90	-	Windows

表 92. 英国、テリトリ ID: GB

コード・ページ	グループ	コード・セット	テリトリ ー・コード	ロケール	オペレー ティング・ システム
819	S-1	ISO8859-1	44	en_GB	AIX
850	S-1	IBM-850	44	En_GB	AIX
923	S-1	ISO8859-15	44	en_GB.8859-15	AIX
1208	N-1	UTF-8	44	EN_GB	AIX
285	S-1	IBM-285	44	-	ホスト
1146	S-1	IBM-1146	44	-	ホスト
819	S-1	iso88591	44	en_GB.iso88591	HP-UX
923	S-1	iso885915	44	-	HP-UX
1051	S-1	roman8	44	en_GB.roman8	HP-UX
819	S-1	ISO-8859-1	44	en_GB	Linux
923	S-1	ISO-8859-15	44	-	Linux
437	S-1	IBM-437	44	-	OS/2
850	S-1	IBM-850	44	-	OS/2
819	S-1	ISO8859-1	44	en_GB	SCO
819	S-1	ISO8859-1	44	en	SCO
819	S-1	ISO8859-1	44	en_GB	Solaris
923	S-1	ISO8859-15	44	en_GB.ISO8859-15	Solaris
1252	S-1	1252	44	-	Windows

表 93. ウクライナ、テリトリー ID: UA

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
1124	S-12	IBM-1124	380	Uk_UA	AIX
1208	N-1	UTF-8	380	UK_UA	AIX
1123	S-12	IBM-1123	380	-	ホスト
1158	S-12	IBM-1158	380	-	ホスト
1125	S-12	IBM-1125	380	-	OS/2
1251	S-12	1251	380	-	Windows

表 94. 米国、テリトリー ID: US

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
819	S-1	ISO8859-1	1	en_US	AIX
850	S-1	IBM-850	1	En_US	AIX
923	S-1	ISO8859-15	1	en_US.8859-15	AIX
1208	N-1	UTF-8	1	EN_US	AIX
37	S-1	IBM-37	1	-	ホスト
1140	S-1	IBM-1140	1	-	ホスト
819	S-1	iso88591	1	en_US.iso88591	HP-UX
923	S-1	iso885915	1	-	HP-UX
1051	S-1	roman8	1	en_US.roman8	HP-UX
819	S-1	ISO-8859-1	1	en_US	Linux
923	S-1	ISO-8859-15	1	-	Linux
437	S-1	IBM-437	1	-	OS/2
850	S-1	IBM-850	1	-	OS/2
819	S-1	ISO8859-1	1	en_US	SCO
819	S-1	ISO8859-1	1	en_US	SGI
819	S-1	ISO8859-1	1	en_US	Solaris
923	S-1	ISO8859-15	1	en_US.ISO8859-15	Solaris
1208	N-1	UTF-8	1	en_US.UTF-8	Solaris
1252	S-1	1252	1	-	Windows

表 95. ベトナム、テリトリー ID: VN

コード・ページ	グループ	コード・セット	テリトリー・コード	ロケール	オペレーティング・システム
1129	S-11	IBM-1129	84	Vi_VN	AIX
1208	N-1	UTF-8	84	VI_VN	AIX
1130	S-11	IBM-1130	84	-	ホスト

表 95. ベトナム、テリトリー ID: VN (続き)

コード・ ページ	グループ	コード・ セット	テリトリー・ コード	ロケール	オペレー ティング・ システム
1164	S-11	IBM-1164	84	-	ホスト
1129	S-11	IBM-1129	84	-	OS/2
1258	S-11	1258	84	-	Windows

注:

- ロードまたはインポート・ユーティリティーで CCSID 1392 および 5488 (GB 18030) を使用して、CCSID 1392 および 5488 から DB2 ユニコード・データベースにデータを移動したり、DB2 ユニコード・データベースから CCSID 1392 または 5488 にエクスポートすることができます。
- AIX 4.3 以降では、コード・ページは 943 です。AIX 4.2 以前をご使用の場合、コード・ページは 932 です。
- ロードまたはインポート・ユーティリティーでコード・ページ 1394 (Shift JIS X0213) を使用して、コード・ページ 1394 から DB2 ユニコード・データベースにデータを移動したり、DB2 ユニコード・データベースからコード・ページ 1394 にエクスポートすることができます。
- 以下は、アラビア語圏の国/地域にマップされます。
 - アラビア語 (サウジアラビア)
 - アラビア語 (イラク)
 - アラビア語 (エジプト)
 - アラビア語 (リビア)
 - アラビア語 (アルジェリア)
 - アラビア語 (モロッコ)
 - アラビア語 (チュニジア)
 - アラビア語 (オマーン)
 - アラビア語 (イエメン)
 - アラビア語 (シリア)
 - アラビア語 (ヨルダン)
 - アラビア語 (レバノン)
 - アラビア語 (クウェート)
 - アラビア語 (アラブ首長国連邦)
 - アラビア語 (バーレーン)
 - アラビア語 (カタール)
- 以下は、英語 (US) にマップされます。

- 英語 (ジャマイカ)
 - 英語 (カリブ海)
6. 以下は、ラテンアメリカ (Lat) にマップされます。
- スペイン語 (メキシコ)
 - スペイン語 (グアテマラ)
 - スペイン語 (コスタリカ)
 - スペイン語 (パナマ)
 - スペイン語 (ドミニカ共和国)
 - スペイン語 (ベネズエラ)
 - スペイン語 (コロンビア)
 - スペイン語 (ペルー)
 - スペイン語 (アルゼンチン)
 - スペイン語 (エクアドル)
 - スペイン語 (チリ)
 - スペイン語 (ウルグアイ)
 - スペイン語 (パラグアイ)
 - スペイン語 (ボリビア)
7. ヒンディ語、グジャラート語、カンナダ語、コンカニー語、マラーティー語、パンジャブ語、サンスクリット語、タミール語、およびテルグ語といったインド語派スクリプトは、ユニコードでサポートされています。
8. コード・ページ 950 は Big5 としても知られています。Microsoft コード・ページ 950 と IBM コード・ページ 950 は以下の点で異なります。

範囲	説明	IBM	Microsoft	違い
X'8140' から X'8DFE'	ユーザー定義文字	ユーザー定義エリア	ユーザー定義エリア	同じ
X'8E40' から X'A0FE'	ユーザー定義文字	ユーザー定義エリア	ユーザー定義エリア	同じ
X'A140' から X'A3BF'	特殊記号	システム文字	システム文字	同じ
X'A3C0' から X'A3E0'	制御記号	システム文字	空	違う
X'A3E1' から X'A3FE'	予約済み	空	空	同じ
X'A440' から X'C67E'	基本使用文字	システム文字	システム文字	同じ
X'C6A1' から X'C878'	Eten 追加記号	システム文字	ユーザー定義エリア	違う
X'C879' から X'C8CC'	Eten 追加記号	空	ユーザー定義エリア	違う

範囲	説明	IBM	Microsoft	違い
X'C8CD' から X'C8D3'	Eten 追加記号	システム文字	ユーザー定義エリア	違う
X'C8D4' から X'C8FD'	予約済み	システム文字	ユーザー定義エリア	違う
X'C8FE'	無効/未定義文字	システム文字	ユーザー定義エリア	違う
X'C940' から X'F9D5'	2 次使用文字	システム文字	システム文字	同じ
X'F9D6' から X'F9FE'	Big-5 用 Eten 拡張	ユーザー定義エリア	システム文字	違う
X'FA40' から X'FEFE'	ユーザー定義文字	ユーザー定義エリア	ユーザー定義エリア	同じ
X'8181' から X'8C82'	ユーザー定義文字	ユーザー定義エリア	空	違う
X'F286' から X'F9A0'	IBM 選択文字	システム文字	空	違う
合計文字数		14 060	13 502	
合計ユーザー定義文字数		6 204	6 217	
合計定義コード・ポイント数		20 264	19 719	

関連タスク:

- 304 ページの『コード・ページ 1394 とユニコード間の変換のために以前の表をインストールする』

双方向特定の CCSID

以下の双方向属性は、異なるプラットフォーム上で双方向データの正しい処理を行うために必要です。

- テキスト・タイプ
- 数値の成形
- 方向
- テキストの成形
- 対称スワッピング

プラットフォームごとにデフォルト値が異なるため、DB2 データをあるプラットフォームから別のプラットフォームに移すと問題が生じる可能性があります。たとえば、Windows オペレーティング・システムは LOGICAL UNSHAPED データを使用しますが、z/OS および OS/390 は通常 SHAPED VISUAL データを使用します。したがって、双方向属性がサポートされていない場合、DB2 Universal Database for OS/390 and

z/OS から Windows 32 ビット・オペレーティング・システム上の DB2 UDB に送られたデータは正しく表示されない可能性があります。

DB2 は、特別な双方向コード化文字セット ID (CCSID) を介して、双方向データ属性をサポートします。表 96 で示されているように、以下の双方向 CCSID はすでに定義されており、DB2 UDB で実装されます。CDRA スtring・タイプは、273 ページの表 97 で示されているように定義されます。

表 96. 双方向 CCSID

CCSID	コード・ページ	String・タイプ
420	420	4
424	424	4
856	856	5
862	862	4
864	864	5
867	862	4
916	916	5
1046	1046	5
1089	1089	5
1200	1200	10
1208	1208	10
1255	1255	5
1256	1256	5
5351	1255	5
5352	1256	5
8612	420	5
8616	424	10
9048	856	5
9238	1046	5
12712	424	4
13488	13488	10
16804	420	4
17248	864	5
62208	856	4
62209	862	10
62210	916	4

表 96. 双方向 CCSID (続き)

CCSID	コード・ページ	STRING・タイプ
62211	424	5
62213	862	5
62215	1255	4
62218	864	4
62220	856	6
62221	862	6
62222	916	6
62223	1255	6
62224	420	6
62225	864	6
62226	1046	6
62227	1089	6
62228	1256	6
62229	424	8
62230	856	8
62231	862	8
62232	916	8
62233	420	8
62234	420	9
62235	424	6
62236	856	10
62237	1255	8
62238	916	10
62239	1255	10
62240	424	11
62241	856	11
62242	862	11
62243	916	11
62244	1255	11
62245	424	10
62246	1046	8
62247	1046	9

表 96. 双方向 CCSID (続き)

CCSID	コード・ページ	ストリング・タイプ
62248	1046	4
62249	1046	12
62250	420	12

表 97. CDRA ストリング・タイプ

ストリング・タイプ	テキスト・タイプ	数値の成形	方向	テキストの成形	対称スワッピング
4	表示	パススルー	LTR	成形	オフ
5	暗黙	アラビア	LTR	非成形	オン
6	暗黙	アラビア	RTL	非成形	オン
7*	表示	パススルー	コンテキスト依存*	非成形リガチャー	オフ
8	表示	パススルー	RTL	成形	オフ
9	表示	パススルー	RTL	成形	オン
10	暗黙	アラビア	コンテキスト依存 LTR	非成形	オン
11	暗黙	アラビア	コンテキスト依存 RTL	非成形	オン
12	暗黙	アラビア	RTL	成形	オフ

注: * 最初のアルファベット文字がローマ字なら、ストリングの方向は左から右 (left-to-right (LTR)) になります。アラビア語またはヘブライ語文字の場合は、右から左 (right-to-left (RTL)) になります。文字が非成形になっていても、LamAlef リガチャーは保たれており、各構成部分に分けられてはいません。

関連概念:

- 管理ガイド: プランニング の『DB2 Connect による双方向サポート』

関連タスク:

- 管理ガイド: プランニング の『双方向サポートの使用可能化』

DB2 でのユニコードのインプリメンテーション

DB2[®] UDB は、UTF-8 および UCS-2、すなわちサロゲートのないユニコードをサポートしています。

ユニコード・データベースが作成される際、CHAR、VARCHAR、LONG VARCHAR、および CLOB データは UTF-8 で格納され、GRAPHIC、VARGRAPHIC、LONG VARGRAPHIC、および DBCLOB データは UCS-2 で格納されます。

バージョン 7.2 FixPak 4 以前の DB2 UDB では、サロゲート・ペアの 2 つの文字が 2 つの独立したユニコード文字として扱われていました。そのため、UTF-16/UCS-2 から UTF-8 にペアをトランスフォームすると、2 つの 3 バイト・シーケンスになります。DB2 UDB バージョン 7.2 FixPak 4 からは、UTF-16/UCS-2 と UTF-8 との間のトランスフォームでサロゲート・ペアが認識されるようになり、UTF-16 サロゲートのペアが 1 つの UTF-8 4 バイト・シーケンスになりました。ほかの使用法では、DB2 はサロゲート・ペアを、引き続き 2 つのそれぞれ別個の UCS-2 文字として扱います。補足文字を非補足文字と区別する方法を知っているのであれば、安全に補足文字を DB2 ユニコード・データベースに保管できます。

DB2 UDB では、揚音アクセントの結合文字 (U+0301) のような (字配りを行わない) 文字を含め、それぞれのユニコード文字が個々の文字として扱われます。したがって、DB2 UDB は、揚音付きのラテン小文字 A (U+00E1) とラテン小文字 A (U+0061) の後に揚音アクセントの結合文字 (U+0301) を続けたものが正規に同一であると認識しません。

UCS-2 ユニコード・データベースのデフォルトの照合シーケンスは IDENTITY です。このシーケンスでは、文字の順番はコード・ポイント順になります。したがって、デフォルトでは、すべてのユニコード文字が順番に並べられ、それぞれのコード・ポイントによって比較されます。非補足ユニコード文字の場合、UTF-8 でエンコードされたときのバイナリー照合の順番と UCS-2 でエンコードされたときのバイナリー照合の順番は同じになります。しかし、エンコードのためにサロゲート文字のペアを必要とする補足文字が含まれている場合は、UTF-8 での文字のエンコードは後ろに向かって照合され、同じ文字を UCS-2 でエンコードしたときは中間のどこかで照合が行われます。そして、この 2 つのサロゲート文字は別々に分けることができます。この理由は、外字を UTF-8 でエンコードしたときに 4 バイトのバイナリー・コード 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx ができ、これが UTF-8 エンコードの U+FFFF (つまり X'EFBFBF') よりも大きくなってしまふことにあります。しかし UCS-2 では、同じ補足文字が UCS-2 の高位のサロゲート文字と低位のサロゲート文字の組み合わせとしてエンコードされ、1101 1000 xxxx xxxx 1101 1100 xxxx xxxx というバイナリー・フォームができ、これは UCS-2 エンコードの U+FFFF よりも小さいです。

IDENTITY_16BIT 照合オプションを使用して、ユニコード・データベースを作成することもできます。IDENTITY_16BIT がデフォルトの IDENTITY 照合オプションの異なっているところは、ユニコード・データベース内の CHAR、VARCHAR、LONG VARCHAR、CLOB データの照合に、UTF-8 バイナリー順序でなく CESU-8 バイナリー順序が使用されることです。CESU-8 は *Compatibility Encoding Scheme for UTF-16: 8-Bit (UTF-16 互換の 8 ビット・エンコード・スキーム)* の略であり、本書の執筆時点において、その仕様は Unicode Technical Consortium Web サイト (www.unicode.org) で

入手可能な Draft Unicode Technical Report #26 に含まれています。CESU-8 は、ユニコード補足文字を除き、バイナリーが UTF-8 と同じです。つまり、ユニコード補足文字は 16 ビットの Basic Multilingual Plane (BMP または Plane 0) の外部で定義されています。UTF-8 エンコードでは、補足文字は 1 つの 4 バイト・シーケンスによって表されますが、CESU-8 では同じ文字に 2 つの 3 バイト・シーケンスが必要です。IDENTITY_16BIT 照合オプションを使用すれば、文字とグラフィック・データの両方で同じ照合順序に従うことができます。IDENTITY_16BIT オプションが指定されている場合でも、CHAR、VARCHAR、LONG VARCHAR、および CLOB データは、引き続き UTF-8 フォーマットで保管されるので注意してください。

日時形式、小数点など、地域に関するパラメーターはすべて、クライアントの現在の地域に基づいています。

ユニコード・データベースでは、DB2 UDB によってサポートされているすべてのコード・ページからの接続が可能です。クライアントのコード・ページと UTF-8 との間のコード・ページ文字変換は、データベース・マネージャーによって自動的に行われます。GRAPHIC ストリング・タイプのデータは常に UCS-2 内に存在し、コード・ページ変換は行われません。ただしコマンド行プロセッサ (CLP) 環境は例外です。CLP から GRAPHIC ストリング (UCS-2) データを意図的に選択すると、戻される GRAPHIC ストリング・データは、CLP によって UCS-2 からそれぞれのクライアント環境のコード・ページに変換されます。

各クライアントは、それぞれの環境でサポートされている文字レパートリー、入力方式、およびフォントによって制限を受けますが、UCS-2 データベース自体はすべての UCS-2 文字を受け入れて格納します。そのため、各クライアントでは、通常は UCS-2 文字のサブセットを処理しますが、データベース・マネージャーでは UCS-2 文字の全レパートリーを処理できます。

文字をローカル・コード・ページから UTF-8 に変換すると、バイト数が増大する可能性があります。ASCII 文字の場合はこのような増大はありませんが、他の UCS-2 文字は、いくつかの要因のために増大します。バージョン 8 以前は、SQL ステートメントのセマンティクスに基づいて、文字データはクライアントのコード・ページでエンコードされているとマークされていたため、データベース・サーバーは、クライアントのコード・ページで全ステートメントを操作していました。この操作では、データが拡張する可能性がありました。バージョン 8 以降では、一度 SQL ステートメントがデータベース・サーバーに入力されると、データベース・サーバーのコード・ページでのみ操作されます。この場合には、サイズの変更はありません。

コード・ページ/CCSID 番号

IBM では、UCS-2 コード・ページを、文字セットが増えていくコード・ページ 1200 として登録しています。つまり、あるコード・ページに新しい文字が追加されるときに、そのコード・ページ番号が変わってしまうことはありません。コード・ページ 1200 は、常に現行バージョンのユニコードを表します。

また、ユニコード 2.0 および ISO/IEC 10646-1 で定義されている特別なバージョンの UCS 規格が、IBM® 内部で CCSID 13488 として登録されています。この CCSID は、GRAPHIC ストリング・データを euc-Japan および euc-Taiwan データベースに格納するときに、DB2 UDB によって内部的に使用されます。CCSID 13488 およびコード・ページ 1200 はどちらも UCS-2 を参照し、値が「2 バイト」(DBCS) のスペースの場合を除き、同じ方法で処理されます。

CP/CCSID	Single-byte (SBCS) space	Double-byte (DBCS) space
1200	N/A	U+0020
13488	N/A	U+3000

NOTE: In a UCS-2 database, U+3000 has no special meaning.

変換表によると、コード・ページ 1200 は CCSID 13488 のスーパーセットであるため、どちらにも同じ (スーパーセット) 表が使われます。

IBM では、UTF-8 は、文字セットが増やされた CCSID 1208 として登録されています (コード・ページ 1208 と同じ)。この規格に新しい文字が追加されたとしても、この番号 (1208) は変わりません。

この MBCS コード・ページ番号は 1208 です。これは、データベースのコード・ページ番号、そしてそのデータベース内に含まれる文字ストリング・データのコード・ページ番号です。UCS-2 用の 2 バイトのコード・ページ番号は 1200 です。これは、データベース内の GRAPHIC ストリング・データのコード・ページ番号です。

関連概念:

- 管理ガイド: プランニング の『Unicode 文字のエンコード』
- 管理ガイド: プランニング の『データ・タイプの Unicode 処理』
- 管理ガイド: プランニング の『Unicode リテラル』

関連タスク:

- 管理ガイド: プランニング の『Unicode データベースの作成』

データベース・ディレクトリーとファイル

データベースを作成する際、デフォルト情報を含むデータベースについての情報がディレクトリー階層に保管されます。この階層ディレクトリー構造は、データベース作成 (CREATE DATABASE) コマンドで提供した情報が決定する位置に作成されます。データベースを作成する際にディレクトリー・パスまたはドライブの位置を指定しない場合には、デフォルト位置が使用されます。

データベースを作成する位置を明示的に指定するようにお勧めします。

CREATE DATABASE コマンドで指定するディレクトリーでは、インスタンスの名前を使用したサブディレクトリーが作成されます。このサブディレクトリーにより、同じデ

ディレクトリーの異なるインスタンスで作成されたデータベースが同じパスを使用しないことが保証されます。インスタンス名のサブディレクトリーの下に、`NODE0000` と名づけられたサブディレクトリーが作成されます。このサブディレクトリーは、論理パーティション・データベース環境のパーティションを区別します。ノード名ディレクトリーの下に、`SQL00001` と名づけられたサブディレクトリーが作成されます。サブディレクトリーの名前は、データベース・トークンを使用し、作成中のデータベースを表します。`SQL00001` には、作成された最初のデータベースに関連したオブジェクトが入りますが、その後のデータベースには、`SQL00002` などの後続番号が付けられます。これらのサブディレクトリーは、`CREATE DATABASE` コマンドで指定したディレクトリーで、このインスタンス内で作成されたデータベースと区別します。

ディレクトリー構造は、以下のとおりです。

```
<your_directory>/<your_instance>/NODE0000/SQL00001/
```

データベース・ディレクトリーには、データベース作成 (`CREATE DATABASE`) コマンドの一部として作成される、以下のファイルが組み込まれます。

- ファイル `SQLBP.1` および `SQLBP.2` には、バッファ・プール情報が入っています。各ファイルには、バックアップを提供するための複製コピーがあります。
- ファイル `SQLSPCS.1` および `SQLSPCS.2` には、表スペース情報が入っています。各ファイルには、バックアップを提供するための複製コピーがあります。
- `SQLDBCON` ファイルには、データベース構成情報が入っています。このファイルを編集しないでください。構成パラメーターを変更するには、コントロール・センターまたはコマンド行ステートメント `UPDATE DATABASE MANAGER CONFIGURATION` および `RESET DATABASE MANAGER CONFIGURATION` のどちらかを使用します。

- `DB2RHIST.ASC` ヒストリー・ファイルおよびそのバックアップ `DB2RHIST.BAK` には、バックアップ、リストア、表のロード、表の再編成、表スペースの変更、およびデータベースへの他の変更についてのヒストリー情報が含まれています。

`DB2TSCHNG.HIS` ファイルには、ログ・ファイル・レベルでの表スペース変更のヒストリーが入っています。各ログ・ファイルごとに、`DB2TSCHG.HIS` には、ログ・ファイルに影響される表スペースを識別するのに役立つ情報が入っています。表スペース・リカバリーは、このファイルからの情報を使用して、表スペース・リカバリー中に処理するログ・ファイルを判別します。テキスト・エディターで、両方のヒストリー・ファイルの内容を調べることができます。

- ログ制御ファイル `SQLOGCTL.LFH` および `SQLOGMIR.LFH` にはアクティブ・ログについての情報が入ります。

リカバリー処理では、このファイルの情報を使用して、リカバリーのために戻るログ内の位置を判別します。`SQLOGDIR` サブディレクトリーには、実際のログ・ファイルが入ります。

注: このログ・サブディレクトリーがご使用のデータに使用されているディスクとは異なるディスクにマップされていることを確認してください。こうすると、ディスクの問題が生じたときに、データとログの両方ではなく、データまたはログのいずれかに範囲を狭めることができます。ログ・ファイルおよびデータベース・コンテナは、同じディスク・ヘッドの移動で競合することはないため、パフォーマンスにも大きな利点となります。ログ・サブディレクトリーのロケーションを変更するには、`newlogpath` データベース構成パラメーターを変更します。

- `SQLINSLK` ファイルは、データベースが必ず 1 つのデータベース・マネージャー・インスタンスでしか使われないようにします。

データベースが作成されると同時に、詳細デッドロック・イベント・モニターも作成されます。詳細デッドロック・イベント・モニター・ファイルは、カタログ・ノードのデータベース・ディレクトリーに保管されています。イベント・モニターが、出力するファイルの最大数に達した場合、イベント・モニターは非活動化され、メッセージが通知ログに書き込まれます。これは、イベント・モニターがディスク・スペースを使用し過ぎるのを防ぎます。不要になった出力ファイルを除去すると、イベント・モニターは次のデータベースの活動化時にアクティブになります。

SMS データベース・ディレクトリーについての追加情報

`SQLT*` サブディレクトリーには、作動データベースに必要なデフォルトのシステム管理スペース (SMS) 表スペースが含まれます。デフォルトの表スペースは 3 つ作成されません。

- `SQLT0000.0` サブディレクトリーには、システム・カタログ表のカタログ表スペースが含まれます。
- `SQLT0001.0` サブディレクトリーには、デフォルト `TEMPORARY` 表スペースが含まれます。
- `SQLT0002.0` サブディレクトリーには、デフォルト・ユーザー・データ表スペースが含まれます。

各サブディレクトリーまたはコンテナには、`SQLTAG.NAM` というファイルが作成されています。このファイルは、サブディレクトリーに使用中のマークを付け、後続の表スペース作成で、これらのサブディレクトリーが使用されないようにします。

さらに、`SQL*.DAT` というファイルが、サブディレクトリーまたはコンテナに含まれる、各表についての情報を保管します。アスタリスク (*) は、各表を示す固有の数字の集合で置き換えられます。各 `SQL*.DAT` ファイルごとに、表タイプ、表の再編成状況、または索引、LOB、または `LONG` フィールドがその表に存在するかどうかによって、以下のファイルが 1 つ以上存在することがあります。

- `SQL*.BMP` (MDC 表である場合、ブロック割り振り情報を含む)
- `SQL*.LF` (`LONG VARCHAR` または `LONG VARGRAPHIC` データを含む)
- `SQL*.LB` (`BLOB`、`CLOB`、または `DBCLOB` データを含む)

- SQL*.LBA (SQL*.LB ファイルについての割り当ておよびフリー・スペース情報を含む)
- SQL*.INX (索引表データを含む)
- SQL*.DTR (SQL*.DAT ファイルの再編成についての一時データを含む)
- SQL*.LFR (SQL*.LF ファイルの再編成についての一時データを含む)
- SQL*.RLB (SQL*.LB ファイルの再編成についての一時データを含む)
- SQL*.RBA (SQL*.LBA ファイルの再編成についての一時データを含む)

関連概念:

- 管理ガイド: プランニング の『SMS 表スペースと DMS 表スペースの比較』
- 管理ガイド: パフォーマンス の『DMS 装置に関する考慮事項』
- 管理ガイド: パフォーマンス の『SMS 表スペース』
- 管理ガイド: パフォーマンス の『DMS 表スペース』
- 管理ガイド: パフォーマンス の『DMS 表スペース・アドレス・マップの図』
- データ・リカバリーと高可用性 ガイドおよびリファレンス の『リカバリー履歴ファイルについて』

関連資料:

- コマンド・リファレンス の『CREATE DATABASE コマンド』

MDC 表を作成する際の考慮事項

既存の表から MDC 表へのデータの移動:

既存の表から MDC 表へデータを移動するには、データをエクスポートし、元の表を削除した後、(ORGANIZE BY DIMENSIONS 文節を含む CREATE TABLE ステートメントを使用して) MDC 表を作成し、その MDC 表にデータをロードします。

SMS 表スペース内の MDC 表:

SMS 表スペースの場合、デフォルトでは一度に 1 つのページがファイルに拡張されます。多次元クラスタリングではデータがブロック単位でクラスタリングされ、ブロッキング因数は表スペースのエクステント・サイズと等しくなります。MDC 表が拡張されるとき、一度に複数ページからなるブロック (つまりエクステント) 単位で拡張されます。したがって、多次元表を SMS 表スペースに保管する場合、複数ページ・ファイル割り振りを使用可能にする必要があることに注意してください。複数ページのファイル割り振りが使用可能であれば、SMS ファイルは厳密に必要なサイズ単位で拡張され、パフォーマンスが大きく向上します。

複数ページのファイル割り振りを使用可能にするためには、`db2empfa` ユーティリティを実行してください。複数ページのファイル割り振りがいったん使用可能になると、それを使用不能にすることはできません。

3 つの次元を使用する MDC 表:

照会で一部の特定の述部が頻繁に使用されることがあらかじめわかっている場合、**ORGANIZE BY DIMENSIONS** を使って、関連する列に基づいて表をクラスター化することができます。

例 1:

```
CREATE TABLE T1 (c1 DATE, c2 INT, c3 INT, c4 DOUBLE)
  ORGANIZE BY DIMENSIONS (c1, c3, c4)
```

例 1 の表は、論理キューブ (つまり 3 次元) を形成する 3 つの固有な列における値に基づいてクラスター化されています。こうすれば、照会処理の際、1 つまたは複数の次元において表をスライスすることにより、適切なスライスまたはセルに含まれるブロックだけがリレーショナル演算子によって処理されるようにすることができます。ブロック・サイズ (つまり 1 ブロック内のページ数) が、表のエクステント・サイズになります。

複数の列からなる次元を使用する MDC 表:

それぞれの次元を、1 つまたは複数の列で構成することが可能です。したがって、2 つの列を含む次元に基づいてクラスター化される表を作成することができます。

例 2:

```
CREATE TABLE T1 (c1 DATE, c2 INT, c3 INT, c4 DOUBLE)
  ORGANIZE BY DIMENSIONS (c1, (c3, c4))
```

例 2 では、c1 および (c3, c4) の 2 つの次元に基づいて表がクラスター化されます。こうすると、照会処理においては、次元 c1 もしくは複合次元 (c3, c4) に基づいて表を論理的にスライスすることができます。この表のブロック数は例 1 の表と同じですが、次元ブロック索引の数は 1 つ少なくなります。例 1 では、列 c1、c3、c4 それぞれに関する 3 つの次元ブロック索引が作成されます。例 2 の場合は、列 c1 に関する次元ブロック索引と、列 c3 および c4 に関する次元ブロック索引の 2 つが作成されます。この 2 つの方法の主な違いは、例 1 の場合、c4 のみを扱う照会が c4 に関する次元ブロック索引を使用して、関連するデータ・ブロックにすばやく直接的にアクセスできることです。例 2 では c4 が次元ブロック索引の 2 番目のキー部分であるため、c4 のみを扱う照会はより多くの処理を行います。ただし、例 2 の場合、DB2® が保守および保管するブロック索引の数は 1 つ少なくなります。

列式を次元として使用する MDC 表:

次元をクラスタリングする際、列式を使用することもできます。列式に基づくクラスター化は、細分性を粗くして次元をロールアップする場合に便利です。たとえば、住所を地理上の場所または地域にロールアップする場合や、日付を週、月、または年にロールアップする場合などです。このような次元のロールアップを実装するには、生成された列を使用することができます。このタイプの列定義では、次元を表す式を使って列を作

成することができます。例 3 のステートメントによって作成される表は、基本となる 1 つの列、および 2 つの列式に基づいてクラスター化されます。

例 3:

```
CREATE TABLE T1(c1 DATE, c2 INT, c3 INT, c4 DOUBLE,  
                c5 DOUBLE GENERATED ALWAYS AS (c3 + c4),  
                c6 INT GENERATED ALWAYS AS (MONTH(C1)))  
    ORGANIZE BY DIMENSIONS (c2, c5, c6)
```

例 3 では、列 c5 が c3 および c4 に基づく式であり、列 c6 は列 c1 を時間的に粗い細分性にロールアップします。このステートメントによって、列 c2、c5、および c6 の値に基づいて表がクラスター化されます。

単調性:

列式には、ユーザー定義関数 (UDF) を含むどんな有効な式をも含めることができます。ただし、ある次元に対して範囲照会を適用するためには (つまり、背景にある次元ブロック索引に対する範囲走査を可能にするためには)、式の性質が単調 (monotonic) でなければなりません。非単調式の場合、その次元に対して適用できるのは等式述部だけです。非単調関数の 1 つの例として、例 3 の列 c6 の定義に出てきた MONTH() があります。列 c1 が日付、タイム・スタンプ、または日付やタイム・スタンプを表す有効なストリング表記である場合、この関数は 1 ~ 12 の範囲の整数値を戻します。この関数の出力は決まりきったものですが、実際には以下のようなステップ関数と同じような出力 (つまり循環パターン) を生成します。

```
MONTH(date('99/01/05')) = 1  
MONTH(date('99/02/08')) = 2  
MONTH(date('99/03/24')) = 3  
MONTH(date('99/04/30')) = 4  
...  
MONTH(date('99/12/09')) = 12  
MONTH(date('00/01/18')) = 1  
MONTH(date('00/02/24')) = 2  
...
```

この例の一連の日付は単調に大きくなっていますが、MONTH(date) はそうではありません。より厳密に言うと、date1 が date2 よりも大きいとき、MONTH(date1) が MONTH(date2) よりも常に大または等しくなるとは限りません。単調性を考慮する必要があるのは、このような場合です。このような非単調性は許可されていますが、基本となる列の範囲述部がその次元の範囲述部を生成できないという点で、次元を制限してしまいます。ただし、式の範囲述部 (たとえば where month(c1) between 4 and 6) は使用できます。こうすれば、開始キーを 4、終了キーを 6 として、次元に関する索引を通常の方法で使用できます。

この関数を単調にするには、月の上位部として年を使用する必要があります。DB2 の組み込み関数 INTEGER には、日付に関する単調式を定義するのに役立つ拡張機能があります。INTEGER(date) は日付の整数表記を戻します。この表記をさらに分割して、

年および月を表す表記を検出することができます。たとえば、
INTEGER(date('2000/05/24')) は 20000524 を返し、
INTEGER(date('2000/05/24'))/100 = 200005 となります。関数 INTEGER(date)/100 は
単調です。

同様に、組み込み関数 DECIMAL および BIGINT にもまた、単調関数を作成できる拡張機能があります。DECIMAL(timestamp) はタイム・スタンプの 10 進表記を戻します。この表記を単調式で使用して、月、日、時間、分などに関して単調増加する値を生成することができます。BIGINT(date) は INTEGER(date) と同様に、日付に関する BIGINT 表記を戻します。

表において生成された列を作成するとき、あるいは次元文節の式から次元を作成するとき、DB2 は可能な限り式の単調性を判別します。DATENUM()、DAYS()、YEAR() などの一部の関数は、単調性を保持する関数として認識されます。さらに、さまざまな数式 (たとえば列や定数の除算、乗算、加算) が単調性を保持します。式の単調性が保持されないと DB2 が判断した場合、あるいは判別が不可能な場合には、次元の基本列では等式述部のみを使用することができます。

関連概念:

- データ移動ユーティリティ 手引きおよび解説書の『エクスポートの概要』
- 管理ガイド: プランニング の『多次元クラスタリング』
- 管理ガイド: プランニング の『MDC 表の次元を選択する際の考慮事項』

関連タスク:

- 管理ガイド: インプリメンテーション の『表のディメンションの定義』

関連資料:

- 1484 ページの『CREATE TABLE』
- コマンド・リファレンス の『db2empfa - 複数ページ・ファイル割り振りの使用可能化コマンド』

バージョン 8 と以前のリリースとの非互換性

システム・カタログ情報

カタログ表の IMPLEMENTED 列:

Windows	UNIX
---------	------

変更点: 以前のバージョンでは、SYSIBM.SYSFUNCTIONS および SYSCAT.SYSFUNCTIONS 内の IMPLEMENTED 列の値は Y、M、H、および N でした。バージョン 8 では、値は Y と N です。

解決方法: 値 Y および N のみを使用するようアプリケーション・コードを変更します。

OBJCAT ビューから SYSCAT ビューへの名前変更:

Windows	UNIX
---------	------

変更点: 以下の OBJCAT ビューは SYSCAT ビューに名前変更されました。
TRANSFORMS、INDEXEXTENSIONS、INDEXEXTENSIONMETHODS、
INDEXEXTENSIONDEP、INDEXEXTENSIONPARMS、PREDICATESPECS、
INDEXEXPLOITRULES。

解決方法: SYSCAT ビューを使用するようアプリケーション・コードを変更します。

読み取り専用になった SYSCAT ビュー:

Windows	UNIX
---------	------

変更点: バージョン 8 では、SYSCAT ビューが読み取り専用になりました。

症状: SYSCAT スキーマ内のビューに対して UPDATE または INSERT 操作を行うと失敗します。

説明: システム・カタログ情報を更新するには、SYSSTAT ビューを使用することをお勧めします。SYSCAT ビューでも更新可能でしたが、これは意図されない動作であり、修正されました。

解決方法: 更新可能な SYSSTAT ビューを新たに参照するようアプリケーションを変更します。

アプリケーション・プログラミング

VERSION オプション使用時に SQL0818N エラーが戻されない:

Windows	UNIX
---------	------

変更点: PRECOMPILE、BIND、REBIND、および DROP PACKAGE コマンドで新しい VERSION オプションを使用すると、実行要求に対して SQL0818N エラーではなく、SQL0805N エラーが戻されるようになりました。

症状: SQL0818N エラーに対応するアプリケーション・コードは、以前のように動作しなくなる可能性があります。

解決方法: SQL0805N と SQL0818N の両方のエラーに対応するようアプリケーション・コードを変更します。

ホスト変数が未定義の場合、SQL0306N エラーがプリコンパイラーに戻されない:

Windows	UNIX
---------	------

変更点: BEGIN DECLARE セクションで宣言されないホスト変数が EXEC SQL セクションで使われた場合、プリコンパイラーによって SQL0306N が戻されなくなりました。アプリケーション内の他のどこかの場所に変数が宣言されている場合、アプリケーション・ランタイムは SQL0804N を戻します。変数がアプリケーションのどこにも宣言されていない場合、コンパイル時にコンパイラーがエラーを戻します。

症状: プリコンパイル時に SQL0306N エラーに対応するアプリケーション・コードは、以前のように動作しなくなる可能性があります。

解決方法: ホスト変数を BEGIN DECLARE セクションで宣言してください。BEGIN DECLARE 以外のセクションでホスト変数を宣言する場合、戻りコード SQL0804 を処理するようアプリケーション・コードを変更する必要があります。

両方向スクロール・カーソルとともに使用できないデータ・タイプ:

Windows	UNIX
---------	------

変更点: LONG VARCHAR、LONG VARGRAPHIC、DATALINK、LOB の各タイプと、これらのタイプの特殊タイプ、および構造型を使用する両方向スクロール・カーソルは、バージョン 8 ではサポートされません。バージョン 7 の両方向スクロール・カーソルでサポートされていたこれらのデータ・タイプは、すべてサポートされなくなりました。

症状: このいずれかのデータ・タイプの列が両方向スクロール・カーソルの選択リストで指定された場合、SQL0270N 理由コード 53 が戻されます。

解決方法: 両方向スクロール・カーソルの選択リストを変更して、これらのタイプの列が含まれないようにします。

ユーロ・バージョンのコード・ページ変換表:

Windows	UNIX
---------	------

変更点: バージョン 8 のコード・ページ変換表はユーロ記号をサポートしますが、以前のバージョンの DB2 の変換表とは少し異なっています。

解決方法: バージョン 8 より前のコード・ページ変換表を使用したい場合には、ディレクトリー `sqllib/conv/v7` にこれらの変換表があります。

LOB ロケータと LOB 値の間の切り替え:

Windows	UNIX
---------	------

変更点: カーソル・ステートメントのバインド・アウト時の、ラージ・オブジェクト (LOB) ロケータと LOB 値の間の切り替え機能が変更されました。アプリケーションが `SQLRULES DB2` でバインドされる時 (デフォルトの動作)、ユーザーは LOB ロケータと LOB 値の間で切り替えできません。

解決方法: カーソル・ステートメントのバインド・アウト時に LOB ロケータと LOB 値の間で切り替えたい場合には、アプリケーションを `SQLRULES STD` でプリコンパイルしてください。

UNIX プラットフォームでコミットされない作業単位:

	UNIX
--	------

変更点: バージョン 8 より前は、コンテキストを明示的にも暗黙にもサポートしない UNIX ベースのアプリケーションでは、`CONNECT RESET`、`COMMIT`、`OR ROLLBACK` ステートメントを直接呼び出さずにアプリケーションが正常終了した場合、未解決の作業単位はコミットされました。(コンテキストを暗黙的にサポートする `CLI`、`ODBC`、および `Java` ベースのアプリケーションの場合、およびアプリケーション・コンテキストを明示的に作成するアプリケーションの場合には、アプリケーション終了時に未解決の作業単位は常にロールバックされました。さらに、アプリケーションが異常終了したときにも、未解決の作業単位に対して暗黙的に `ROLLBACK` が実行されました。バージョン 8 では、アプリケーションのあらゆる終了時に、未解決の作業単位は暗黙的にロールバックされます。Windows ベースのアプリケーションは変更されていません。Windows ベースのアプリケーションの場合、アプリケーションが正常終了したか異常終了したかにかかわらず、これまでどおり暗黙的に `ROLLBACK` が実行されます。

解決方法: トランザクションを確実にコミットするためには、アプリケーションが終了する前に、明示的 `COMMIT` または `CONNECT RESET` を実行する必要があります。

保管点の命名に関する変更:

Windows	UNIX
---------	------

変更点: 保管点の名前の先頭に "SYS" を使用することはできなくなりました。

症状: 名前が "SYS" で始まる保管点を作成しようとする、エラー SQL0707N が発生して失敗します。

説明: 名前が "SYS" で始まる保管点は、システムで使用するために予約されています。

解決方法: 名前が "SYS" で始まるすべての保管点を名前変更して、"SYS" 以外で始まるようにしてください。

コード・ページ変換エラーとバイト置換:

Windows	UNIX
---------	------

変更点: コード・ページ変換が必要な場合、変換はバインド・イン段階で実行されるようになりました。

症状: コード・ページ変換エラーとバイト置換が、新しい状況で発生し得るようになりました。たとえば以下のステートメントは、ホスト変数 :hv 内のデータをアプリケーション・コード・ページからデータベース・コード・ページに変換するようになりました (これらのコード・ページが同じでない場合)。

```
SELECT :hv FROM table  
VALUES :hv
```

説明: エラーやバイト置換を伴わずに、提供されたデータをデータベース・コード・ページに変換することは不可能でした。以前はコード・ページ変換が行われませんでした、今後はコード・ページ変換が行われます。

解決方法: データ、アプリケーション・コード・ページ、またはデータベース・コード・ページを変更して、コード・ページ変換によってエラーやバイト置換が発生しないようにしてください。あるいは、コード・ページ変換エラーやバイト置換が発生した場合に対処できるよう、アプリケーション・コードを変更してください。

ホスト変数のコード・ページ変換:

Windows	UNIX
---------	------

変更点: コード・ページ変換が必要な場合、変換はバインド・イン段階で実行されるようになりました。

症状: 異なる結果。

説明: コード・ページ変換が必要な場合、常にホスト変数に関して変換されるようになったため、述部評価はアプリケーション・コード・ページではなく、常にデータベース・コード・ページで行われます。たとえば、

```
SELECT * FROM table WHERE :hv1 > :hv2
```

この処理は、アプリケーション・コード・ページではなく、データベース・コード・ページを使って行われます。使用される照合は、これまでどおりデータベース照合です。

解決方法: 以前のバージョンでの結果が、本当に望ましい結果であったかどうかを確認してください。望ましい結果であった場合には、述部を変更して、使用されるデータベース照合とコード・ページのもとで適切な結果が生成されるようにしてください。あるいは、アプリケーション・コード・ページまたはデータベース・コード・ページを変更してください。

ホスト変数内のデータの拡張と収縮:

Windows	UNIX
---------	------

変更点: コード・ページ変換が必要な場合、変換はバインド・イン段階で実行されるようになりました。

症状: ホスト変数のデータの長さが異なります。

説明: コード・ページ変換時に拡張または収縮が行われる可能性があるため、ホスト変数内のデータの長さに依存する操作の結果が異なったり、エラー状態が発生したりする場合があります。

解決方法: データ、アプリケーション・コード・ページ、またはデータベース・コード・ページを変更して、コード・ページ変換によって変換後のデータの長さが変更されないようにしてください。あるいは、コード・ページ変換によってデータの長さが変更された場合に対処できるよう、アプリケーション・コードを変更してください。

コード・ページ変換後のホスト変数の長さ:

Windows	UNIX
---------	------

変更点: コード・ページ変換の際、拡張のために、ホスト変数やパラメーター・マーカの結果の長さが増幅されることはなくなりました。

症状: データ切り捨てエラー。

説明: タイプなしパラメーター・マーカ用に決定される文字データ・タイプの長さは、もはやコード・ページ変換による拡張の可能性を考慮して増幅されません。結果の長さは、タイプなしパラメーター・マーカの長さを使って結果の長さを決定する操作のために、より短くなります。たとえば、C1 が CHAR(10) 列とすると、

```
VALUES CONCAT (?, C1)
```

この結果は、アプリケーション・コード・ページからデータベース・コード・ページへの変換で 3 倍に拡張される可能性のあるデータベースの場合、もはや CHAR(40) のデータ・タイプおよび長さにはならず、CHAR(20) の結果データ・タイプおよび長さになります。

解決方法: CAST を使用して、タイプなしパラメーター・マーカに適切なタイプを設定するか、タイプなしパラメーター・マーカのタイプを決定するオペランドを変更して、コード・ページ変換による拡張に対応できるようなデータ・タイプおよび長さにします。

DESCRIBE ステートメントの出力に関する変更:

Windows	UNIX
---------	------

変更点: コード・ページ変換の際、拡張のために、ホスト変数やパラメーター・マーカの結果の長さが増幅されることはなくなりました。

症状: DESCRIBE ステートメントの出力が変更されました。

説明: コード・ページ変換による拡張可能性を考慮して結果の長さが増幅されることはないため、そのような結果の長さを記述する DESCRIBE ステートメントの出力が異なります。

解決方法: 必要であれば、DESCRIBE ステートメントから戻される新しい値を扱えるようにアプリケーションを変更してください。

ホスト変数とともに SUBSTR 関数を使用する際のエラー:

Windows	UNIX
---------	------

変更点: コード・ページ変換の際、拡張のために、ホスト変数やパラメーター・マーカの結果の長さが増幅されることはなくなりました。

症状: SUBSTR からのエラー SQL0138N。

説明: コード・ページ変換による拡張の可能性を考慮して、ホスト変数用の長さが増幅されていました。これによって、たとえば SUBSTR (:hv,19,1) が、長さ 10 のホスト変数に関して正常に機能しました。今後は、これが機能しません。

解決方法: 変換後のデータの長さを考慮に入れるようホスト変数の長さを増幅するか、SUBSTR 呼び出しを変更して、ホスト変数の長さ以内の位置を指定します。

非スレッド・セーフのライブラリーはもはや Solaris でサポートされない:

	UNIX
--	------

変更点: 非スレッド・セーフ・ライブラリー `libdb2_noth.so` はもはや使用できません。

症状: `libdb2_noth.so` を必要とするツールやアプリケーションは機能しません。

説明: 非スレッド・セーフ・ライブラリーを引き続きサポートする必要がないため、`libdb2_noth.so` ライブラリーは DB2 UDB for Solaris に付属していません。

解決方法: 新たにスレッド・セーフの `libdb2.so` ライブラリーを使用するよう、ツールまたはアプリケーションを変更してください。 `-mt` パラメーターを使ってアプリケーションを再リンクしてください。

ユニコード・データベースへの接続時の DBCLOB のインポートまたはエクスポート:

Windows	UNIX
---------	------

変更点: バージョン 8 以前は、ユニコード・データベース (UTF-8) から DBCLOB を含むデータをエクスポートしたときに、`LOBSINFILE` ファイル・タイプ修飾子を使用した場合、DBCLOB はコード・ページ 1200 (ユニコード・グラフィック・コード・ページ) でエクスポートされました。 DBCLOB を含むデータをインポートしたときに、`LOBSINFILE` ファイル・タイプ修飾子を使用した場合、DBCLOB はコード・ページ 1200 (ユニコード・グラフィック・コード・ページ) でインポートされました。この動作は、`DB2GRAPHICUNICODESERVER` レジストリー変数を ON に設定している場合に、バージョン 8 でも維持されます。

バージョン 8 では、`DB2GRAPHICUNICODESERVER` レジストリー変数のデフォルト設定は OFF です。 DBCLOB を含むデータをエクスポートするときに、`LOBSINFILE` ファイル・タイプ修飾子を使用している場合、DBCLOB はアプリケーションのグラフィック・コード・ページでエクスポートされます。 DBCLOB を含むデータをインポートするときに、`LOBSINFILE` ファイル・タイプ修飾子を使用している場合、DBCLOB はアプリケーションのグラフィック・コード・ページでインポートされます。ご使用のアプリケーション・コードが EUC-JP (954) または IBMec-TW (964) で、DBCLOB を含むデータをエクスポートするときに、`LOBSINFILE` ファイル・タイプ修飾子を使用している場合、DBCLOB はアプリケーションの文字コード・ページでエクスポートされます。 DBCLOB を含むデータをインポートするときに、`LOBSINFILE` ファイル・タイプ修飾子を使用している場合、DBCLOB はアプリケーションの文字コード・ページでインポートされます。

症状: `LOBSINFILE` ファイル・タイプ修飾子を持つデータを、ユニコード・データベースにインポートしている場合、文字データは正しく変換されますが、DBCLOB データは壊れます。

解決方法: バージョン 8 データベースと古いデータベース間でデータを移動している場合は、DB2GRAPHICUNICODESERVER レジストリー変数を ON に設定して、以前の動作を保存します。

SQL

関数とプロシージャの同一の特定名は許可されない:

Windows	UNIX
---------	------

変更点: SPECIFICNAME 用のネーム・スペースが統一されました。以前のバージョンの DB2 では、関数とプロシージャの特定名を同じにすることができましたが、バージョン 8 ではこれは許可されません。

症状: データベースをバージョン 8 に移行する場合、db2ckmig ユーティリティーが関数とプロシージャの同じ特定名をチェックします。移行の際に重複名が検出されると、移行は失敗します。

解決方法: プロシージャを削除し、別の特定名を使って再作成してください。

関数およびプロシージャの EXECUTE 特権:

Windows	UNIX
---------	------

変更点: これまでは、他のユーザーがルーチンを使用できるようにするには、単にルーチンを作成する必要があるだけでした。今後は、ルーチンを作成した後、それに対して GRANT EXECUTE を実行すれば、他のユーザーはそれを使用できるようになります。

以前のバージョンではプロシージャに関する許可検査がありませんでしたが、呼び出し元では、プロシージャから呼び出されるすべてのパッケージに関する EXECUTE 特権が必要でした。バージョン 8 では、CALL_RESOLUTION IMMEDIATE を使ってプリコンパイルされる組み込みアプリケーションの場合、および CLI カタログ式プロシージャの場合には、呼び出し元にプロシージャに関する EXECUTE 特権が必要です。すべてのパッケージに関する EXECUTE 特権が必要なのは、プロシージャ定義者のみです。

症状:

1. アプリケーションが正しく動作しない可能性があります。
2. 複数のパッケージからなる既存のプロシージャで、定義者がすべてのパッケージへのアクセスを許可されているわけではない場合には、プロシージャが正しく動作しません。

解決方法:

1. 必要な GRANT EXECUTE ステートメントを発行します。すべてのルーチンが単一のスキーマに入っている場合、それぞれのタイプのルーチンごとに以下のような 1 つのステートメントを使用して、特権を付与します。

```
GRANT EXECUTE ON FUNCTION schema1.* TO PUBLIC
```

2. 1 つのパッケージはすべてのユーザーが使用でき、別のパッケージは少数の特権ユーザーにのみ制限されている場合、両方のパッケージを使用するストアード・プロシージャは、2 番目のパッケージにアクセスするとき、権限エラーを監視します。権限エラーが検出された場合には、そのユーザーが特権ユーザーではないことが判明したため、プロシージャの論理の一部がう回されます。

これを解決するには、以下のいくつかの方法があります。

- a. プログラムをプリコンパイルするとき、CALL_RESOLUTION DEFERRED を設定します。こうすれば、プリコンパイラーが CALL ステートメントでプロシージャの解決に失敗した場合、使用すべきでない sqlproc() API 呼び出しとしてプログラムが実行されます。
- b. プロシージャの呼び出し方法を制御するために、CLI キーワード UseOldStpCall を db2cli.ini ファイルに追加することができます。この値は 2 つあり、値 0 の場合は古い呼び出しメソッドを使ってプロシージャを呼び出しません。値 1 の場合、古い呼び出しメソッドを使ってプロシージャを呼び出します。
- c. パッケージを実行するすべてのユーザーに EXECUTE 特権を付与します。

表への外部キー制約の追加:

Windows	UNIX
---------	------

変更点: 以前のバージョンでは、チェック・ペンディング状態の表を参照する外部キー制約を作成した場合、従属表もまたチェック・ペンディング状態になりました。バージョン 8 では、チェック・ペンディング状態の表を参照する外部キー制約を作成した場合、結果は 2 通りの可能性があります。

1. 従属表の作成時に外部キー制約が追加された場合、表は空の状態で作成され、制約に反する行が存在しないため、表の作成と制約の追加はどちらも正常に行われます。
2. 既存の表に外部キーが追加された場合、エラー SQL0668N が発生します。

解決方法: 表を参照する外部キーを追加する前に、SET INTEGRITY ... IMMEDIATE CHECKED ステートメントを使用して、チェック・ペンディング状態にある表の保天性チェックをオンにします。

SET INTEGRITY ... IMMEDIATE CHECKED の変更:

Windows	UNIX
---------	------

変更点: 前のリリースでは、SET INTEGRITY ... UNCHECKED ステートメントが実行された表 (つまり、SYSCAT.TABLES の const_checked 列に 'U' バイトが含まれる表) は、デフォルトで、次の SET INTEGRITY ... IMMEDIATE CHECKED ステートメントで完全に処理されました (つまり、すべてのレコードの制約違反がチェックされました)。全体が処理されないようにするには、明示的に INCREMENTAL を指定する必要があります。

バージョン 8 では、SET INTEGRITY ... IMMEDIATE CHECKED ステートメントが発行された場合、デフォルトでは増分処理だけを行い、未チェック・データをそのままにします (つまり 'U' バイトを保持します)。(古いデータが検証されない旨の警告が戻されます。)

説明: このように変更された理由は、デフォルト動作で、(多くのリソースを消費する) すべてのレコードの制約チェックを行わないようにするためです。

解決方法: 全体の処理を強制するためには、明示的に NOT INCREMENTAL を指定する必要があります。

CHAR 関数の小数点:

Windows	UNIX
---------	------

変更点: 小数点としてコンマを使用するロケールがサーバーに設定されている場合、そこで実行される動的アプリケーションに、引き数タイプ REAL または DOUBLE を使った CHAR 関数の非修飾呼び出しが含まれていれば、CHAR(double) 関数の結果としてピリオドが区切り文字として戻されます。さらに、ビューやトリガーのようなオブジェクトをバージョン 8 で再作成した場合や、静的パッケージを明示的に再バインドした場合にも、こうした非互換性が見られます。

説明: これは、SYSFUN.CHAR(double) 関数シグニチャーではなく、SYSIBM.CHAR(double) 関数シグニチャーに解決されたためです。

解決方法: 以前のバージョンの DB2 の動作を保持するには、関数解決で SYSIBM.CHAR シグニチャーが選択されるのを防ぐために、アプリケーションで明示的に関数 SYSFUN.CHAR を呼び出す必要があります。

CALL ステートメントの変更:

Windows	UNIX
---------	------

変更点: バージョン 8 では、CALL_RESOLUTION IMMEDIATE および CLI カタログ式プロシーチャーを使ってプリコンパイルされたアプリケーションには、以前のバージョンと比べて以下のような大きな相違点があります。

- ホスト変数サポートに代わって、動的 CALL がサポートされます。

- 非カタログ式ストアード・プロシージャーを呼び出すアプリケーションのコンパイルは、サポートされなくなりました。非カタログ式ストアード・プロシージャーは、将来の DB2 バージョンでは完全にサポートされなくなる予定です。
- 変数引き数リスト・ストアード・プロシージャーはサポートされなくなりました。
- ストアード・プロシージャー・ライブラリーのロードに関する規則が異なります。

解決方法: バージョン 8 より前にサポートされていた CALL ステートメントを引き続き使用できます。PRECOMPILE PROGRAM コマンドで CALL_RESOLUTION DEFERRED オプションを使用して、これを利用することができます。

(バージョン 8 より前に作成された) 既存のアプリケーションは、引き続き動作します。CALL_RESOLUTION DEFERRED オプションを使わずにアプリケーションが再ブリコンパイルされる場合、ソース・コードを変更する必要があるかもしれません。

CALL_RESOLUTION DEFERRED ステートメントは、将来のバージョンではサポートされなくなる予定です。

固定長ストリングを戻す UDF からの出力:

Windows	UNIX
---------	------

変更点: UDF (スカラー関数または表関数) が固定長ストリング (CHAR(n) または GRAPHIC(n)) を戻すよう定義することができます。以前のバージョンでは、戻り値に組み込み NULL 文字が含まれる場合、NULL 文字および NULL 文字の右側のすべてのバイトを含めて、結果は単に n バイト (GRAPHIC データ・タイプの場合は 2n バイト) になりました。バージョン 8 では DB2 は NULL 文字を検索し、その地点 (NULL 文字) 以降、値の末尾までブランクを戻します。

解決方法: バージョン 8 より前の動作を保持したい場合には、戻り値の定義を CHAR(n) から CHAR(n) FOR BIT DATA に変更してください。GRAPHIC データに関しては、バージョン 8 より前の動作を保持する方法はありません。

データベース接続動作の変更点:

Windows	UNIX
---------	------

変更点: バージョン 7 では、組み込み SQL を使ってデータベースに接続した後、存在しないデータベースへの接続を試行した場合、SQL1013N が発生してその試行は失敗しますが、最初のデータベースへの接続は存続しました。バージョン 8 では、存在しないデータベースへの接続を試行すると、最初のデータベースへの接続が切断されます。これによって、アプリケーションはどこにも接続していない状態になります。

解決方法: 別のデータベースへの接続試行が失敗した後、最初のデータベースに再接続するよう、組み込み SQL をコーディングしてください。

パッケージに関する CONTROL の取り消し:

Windows	UNIX
---------	------

変更点: ユーザーは CONTROL 特権を使用して、パッケージに関する特権を付与できます。DB2 バージョン 8 では、あるユーザーがパッケージ使用特権を他のユーザーに付与できるかどうかの権限を決定するために、WITH GRANT OPTION を使用できます。このメカニズムは、ユーザーが他のユーザーに特権を付与できるかどうかを決定するうえで、CONTROL に代わるものです。CONTROL が取り消されても、ユーザーは引き続き他のユーザーに特権を付与できます。

症状: CONTROL 特権が取り消された後でも、ユーザーは引き続きパッケージに関する特権を付与できます。

解決方法: もはやパッケージに関する特権を付与させないようにするには、パッケージに関するすべての特権を取り消して、必要な特権のみを付与してください。

FOR BIT DATA 文字ストリングを CLOB にキャストする際のエラー:

Windows	UNIX
---------	------

変更点: CAST 指定または CLOB 関数を使用して、FOR BIT DATA として定義されている文字ストリングを CLOB にキャストすると、エラー (SQLSTATE 42846) が戻されるようになりました。

症状: CLOB にキャストすると、これまでとは異なり、エラーが戻されます。

説明: FOR BIT DATA は CLOB データ・タイプに関してサポートされていません。FOR BIT DATA ストリングを引数とする CAST 指定または CLOB 関数の使用結果は、定義されていません。この状況は、エラーとして検知されるようになりました。

解決方法: CAST 指定または CLOB 関数への引数を変更して、FOR BIT DATA ストリング以外にします。そうするには、CAST 指定を使用し、FOR BIT DATA ストリングを FOR SBCS DATA ストリングまたは FOR MIXED DATA ストリングにキャストします。たとえば、C1FBD が FOR BIT DATA と宣言された VARCHAR(20) 列であれば、非 DBCS データベースでは、以下のような引数が CLOB 関数への引数として有効です。

```
CAST (C1FBD AS VARCHAR(20) FOR SBCS DATA)
```

CHR 関数からの出力:

Windows	UNIX
---------	------

変更点: CHR(0) は、コード・ポイント X'00' の文字の代わりに、空白 (X'20') を返します。

症状: X'00' を引き数とした CHR 関数からの出力が、異なる結果を返します。

説明: ユーザー定義関数から呼び出され、返されるときにの文字列処理が、X'00' を文字列の終わりとして解釈します。

解決方法: 新しい出力値を処理するように、アプリケーション・コードを変更してください。または、SYSFUN CHR 関数にある CHAR(1) FOR BIT DATA を返すユーザー定義関数を定義して、この関数を SQL パス上で SYSFUN の前に置き換えます。

TABLE_NAME および TABLE_SCHEMA 関数は生成された列またはチェック制約で使用できません:

Windows	UNIX
---------	------

変更点: TABLE_NAME および TABLE_SCHEMA 関数の定義が訂正され、生成された列またはチェック制約では使用できなくなります。

症状: バインドは SQLCODE -548/SQLSTATE 42621 で、TABLE_NAME または TABLE_SCHEMA がチェック制約のコンテキストで無効であると述べられ、失敗します。

説明: TABLE_NAME および TABLE_SCHEMA 関数はカタログ・ビューからデータを検索します。これらはクラス READS SQL DATA に属します。クラス READS SQL DATA の機能は GENERATED COLUMN 式およびチェック制約では許可されません。これは DB2 が長期にわたる制約の訂正を強制できないためです。

解決方法: 生成列およびチェック制約が含まれる任意の列を更新して、TABLE_NAME と TABLE_SCHEMA の使用を除去してください。生成列を変更するには ALTER TABLE ステートメントを使用して、新しい式を設定します。チェック制約を除去するには、DROP CONSTRAINT 文節を指定して ALTER TABLE ステートメントを使用してください。これによって変更を反映した列が含まれる表のバインドとアクセスの続行が可能になります。

データベースのセキュリティと調整

CREATE FUNCTION、CREATE METHOD、および CREATE PROCEDURE ステートメントに関する権限:

Windows	UNIX
---------	------

変更点: CREATE_EXTERNAL_ROUTINE 権限がバージョン 8 で新たに導入されました。

症状: EXTERNAL オプションを使用する CREATE FUNCTION、CREATE METHOD、および CREATE PROCEDURE ステートメントは失敗します。

解決方法: EXTERNAL オプションを使用する CREATE FUNCTION、CREATE METHOD、および CREATE PROCEDURE ステートメントを発行するユーザーに対して、CREATE_EXTERNAL_ROUTINE 権限を付与してください。

ユーティリティとツール

下位レベルの CREATE DATABASE および DROP DATABASE はサポートされない:

Windows	UNIX
---------	------

変更点: バージョン 8 では、下位クライアントからの、または下位サーバーへの CREATE DATABASE コマンドと DROP DATABASE コマンドはサポートされません。

症状: これらのいずれかのコマンドを発行した場合、エラー SQL0901N が発生します。

説明: CREATE DATABASE コマンドおよび DROP DATABASE コマンドは、バージョン 8 のクライアントとサーバーの間でのみサポートされています。これらのコマンドを、バージョン 6 または 7 のクライアントからバージョン 8 サーバーに発行することはできません。これらのコマンドを、バージョン 8 クライアントからバージョン 7 サーバーに発行することはできません。

解決方法: バージョン 8 クライアントからは、バージョン 8 のデータベースを作成またはドロップしてください。バージョン 7 データベースの作成またはドロップは、バージョン 6 または 7 のクライアントから行ってください。

ロード後の表のモード変更:

Windows	UNIX
---------	------

変更点: 以前のバージョンでは、INSERT オプションを使ってロードされ、即時マテリアライズ照会表 (サマリー表ともいう) を含んでいる表は、SET INTEGRITY IMMEDIATE CHECKED ステートメントの実行後に 標準 (フル・アクセス) 状態になりました。バージョン 8 では、SET INTEGRITY IMMEDIATE CHECKED ステートメント実行後に表は非データ移動モードになります。

説明: 非データ移動モードの表へのアクセスは標準 (フル・アクセス) モードの表へのアクセスとほぼ同様ですが、その表内部でのデータ移動を行う一部のステートメントやユーティリティーは使用できません。

解決方法: 従属即時サマリー表を含んでいるロード済みの基本表が、非データ移動モードをう回して、フル・アクセス・モードに直接移行するように強制できます。そうするには、基本表に対して SET INTEGRITY ... IMMEDIATE CHECKED FULL ACCESS ステートメントを発行します。ただし、これによって従属即時マテリアライズ照会表 (サマリー表ともいう) のフル・リフレッシュが強制されるため、このオプションの使用は推奨されていません。

挿入モードまたは置換モードのロード・ユーティリティー:

Windows	UNIX
---------	------

変更点: 以前バージョンでは、ロード・ユーティリティーを挿入モードまたは置換モードで使用する時、健全性検査がオフの場合のデフォルト・オプションは CASCADE IMMEDIATE でした。表がチェック・ペンディング状態になると、そのすべての従属外部キー表および従属マテリアライズ照会表 (サマリー表ともいう) もまた、ただちにチェック・ペンディング状態になりました。

バージョン 8 の場合、ロード・ユーティリティーを挿入モードまたは置換モードで使用する時、健全性検査がオフの場合のデフォルトは CASCADE DEFERRED です。

解決方法: 従属外部キー表および従属マテリアライズ照会表を親表とともにチェック・ペンディング状態にするには、LOAD コマンドの CHECK PENDING CASCADE IMMEDIATE オプションを使用することができます。

DB2_LIKE_VARCHAR がサブエレメント統計の収集を制御しない:

Windows	UNIX
---------	------

変更点: バージョン 7 では、DB2_LIKE_VARCHAR レジストリー変数はサブエレメント統計の収集を、これらの統計の使用と同様に制御していました。バージョン 8 では、DB2_LIKE_VARCHAR はサブエレメント統計の収集を制御しません。その代わりに、サブエレメント統計の収集は、RUNSTATS コマンドの LIKE STATISTICS オプションまたは、db2Runstats API の iColumnflags パラメーターの DB2RUNSTATS_COLUMN_LIKE_STATS 値によって制御されます。

症状: RUNSTATS コマンドの呼び出し、または db2Runstats API の呼び出しの後で、サブエレメント統計はシステム・カタログ内で -1 (デフォルト) に設定されます。これは以下のような照会を実行することで確認できます。

```

SELECT SUBSTR(TABSCHEMA,1,18), SUBSTR(TABNAME,1,18), SUBSTR(COLNAME,1,18),
COLCARD, AVGCOLLEN, SUB_COUNT, SUB_DELIM_LENGTH
FROM SYSSTAT.COLUMNS
WHERE COLNAME IN ('P_TYPE', 'P_NAME')
ORDER BY 1,2,3

```

(適当な列名で P_TYPE および P_NAME を置換)

列の結果で、COLCARD および AVGCOLLEN に負でない値があり、SUB_COUNT および SUB_DELIM_LENGTH に -1 の値がある場合、基本統計は列から収集されますが、サブエレメント統計は収集されないことを示します。

解決方法: バージョン 7 で DB2_LIKE_VARCHAR=?,Y (? は任意の値) を指定した場合、該当する列に対するこれらの統計を収集するためには、RUNSTATS コマンドで LIKE STATISTICS オプションを指定するか、db2Runstats API で DB2RUNSTATS_COLUMN_LIKE_STATS オプションを指定する必要があります。

接続性と共存

下位レベル・サーバーのサポート:

Windows	UNIX
---------	------

変更点: バージョン 7 環境からバージョン 8 環境に移行するとき、すべてのサーバーをバージョン 8 に移行する前に、クライアント・マシンを 8 に移行する場合には、いくつかの制約事項および制限があります。これらの制約事項と制限は、DB2 Connect、zSeries、OS/390、iSeries データベース・サーバーのいずれにも関連していません。

解決方法: バージョン 8 クライアントがバージョン 7 サーバーと連動する場合、バージョン 7 サーバー上で DRDA Application Server 機能を構成して使用可能にする必要があります。これを行う方法については、バージョン 7 の インストールおよび構成補足を参照してください。

既知の制約事項と制限を回避するためには、いずれかのクライアント・マシンをバージョン 8 に移行する前に、すべてのサーバーをバージョン 8 に移行する必要があります。それが不可能な場合は、バージョン 8 クライアントからバージョン 7 サーバーにアクセスするとき、以下のものがサポートされないことに注意してください。

- 以下のデータ・タイプ:
 - ラージ・オブジェクト (LOB) データ・タイプ
 - ユーザー定義特殊タイプ (UDT)
 - DATALINK データ・タイプ。

DATALINK データ・タイプは、非リレーショナル・ストレージに存在する外部データの管理を可能にします。 DATALINK データ・タイプは、物理的に DB2 Universal Database 以外のファイル・システムに存在するファイルを参照します。

- 以下のセキュリティ機能:
 - 認証タイプ SERVER_ENCRYPT。

SERVER_ENCRYPT は、パスワード暗号化方式です。暗号化されたパスワードとユーザー ID を使用して、ユーザーが認証されます。
 - パスワードの変更。

バージョン 8 クライアントからバージョン 7 サーバー上のパスワードを変更することはできません。
- 以下の接続および通信プロトコル:
 - 接続ではなく ATTACH を必要とするインスタンス要求。

バージョン 8 クライアントからバージョン 7 サーバーへの ATTACH はサポートされていません。
 - サポートされている唯一のネットワーク・プロトコルは TCP/IP です。

その他のネットワーク・プロトコル (SNA、NetBIOS、IPX/SPX など) はサポートされません。
- 以下のアプリケーション機能およびタスク:
 - DESCRIBE INPUT ステートメントはサポートされていません。ただし、ODBC/JDBC アプリケーションの場合は例外です。

バージョン 7 サーバーにアクセスする ODBC/JDBC アプリケーションを実行するバージョン 8 クライアントをサポートするには、このタイプのアクセスが必要なすべてのバージョン 7 サーバーに、 DESCRIBE INPUT サポート用の修正プログラムを適用する必要があります。この修正プログラムは APAR IY30655 に関連付けられており、バージョン 8 の一般出荷可能日よりも前に入手できる予定です。いずれかの DB2 Universal Database ドキュメンテーションの『IBM への連絡方法』を参照して、APAR IY30655 に含まれる修正プログラムの入手方法を確認してください。

DESCRIBE INPUT ステートメントを使用すると、アプリケーション要求元は準備済みステートメントにおける入力パラメーター・マーカーについての記述を入手でき、パフォーマンスおよびユーザビリティが改善されます。CALL ステートメントの場合、これには、ストアード・プロシージャの IN および INOUT パラメーターに関連したパラメーター・マーカーが含まれます。
 - 2 フェーズ・コミット

バージョン 8 クライアントが含まれる整合トランザクションを使用するとき、バージョン 7 サーバーはトランザクション・マネージャー・データベースとして使用できません。また、バージョン 7 サーバーは、バージョン 8 サーバーがトランザクション・マネージャー・データベースになる可能性のある整合トランザクションには参加できません。

- XA 準拠のトランザクション・マネージャー。

バージョン 8 クライアントを使用するアプリケーションは、バージョン 7 サーバーを XA リソースとして使用できません。これには、トランザクション管理の一部となっている WebSphere、Microsoft COM+/MTS、BEA WebLogic などが含まれます。

- モニター

- ユーティリティ。

クライアントからサーバーに向けて開始されるユーティリティは、クライアントがバージョン 8、サーバーがバージョン 7 の場合にはサポートされません。

- サイズが 32 KB を超える SQL ステートメント

バージョン 7 サーバーと連動するバージョン 8 クライアントに関するこれらの制限および制約事項のほかに、バージョン 7 サーバーと連動するバージョン 8 ツールにも同様の制限や制約事項があります。

以下のバージョン 8 ツールは、バージョン 8 サーバーのみをサポートします。

- コントロール・センター
- タスク・センター
- ジャーナル
- サテライト管理センター
- インフォメーション・カタログ・センター (Web バージョンのセンターを含む)
- ヘルプ・センター (Web バージョンのセンターを含む)
- ライセンス・センター
- "Spatial Extender"
- ツール設定

以下のバージョン 8 ツールは、バージョン 7 サーバー (制約事項あり) とバージョン 8 サーバーの両方をサポートします。

- 構成アシスタント (このツールにはさまざまなコンポーネントがありますが、バージョン 7 サーバーとともに使用できるのはインポート/エクスポート構成ファイルのみです。すべてのコンポーネントは、バージョン 8 と連動します。)
- データウェアハウス・センター
- レプリケーション・センター
- コマンド・センター (Web バージョンのセンターを含む)
- SQL アシスタント
- デベロップメント・センター
- Visual Explain

一般的に言って、コントロール・センターのナビゲーション・ツリーからのみ立ち上げられるバージョン 8 ツール、およびこれらのツールに基づく詳細ビューは、バージョン 7 以前のサーバーからアクセスおよび利用できません。バージョン 7 のツールをバージョン 7 以前のサーバーとともに使用することを考慮してください。

両方向スクロール・カーソル・サポート:

Windows	UNIX
---------	------

変更点: バージョン 8 では、DB2 UDB for Unix および for Windows バージョン 8 クライアントから DB2 UDB for Unix および for Windows バージョン 7 サーバーへの両方向スクロール・カーソル機能はサポートされません。両方向スクロール・カーソル機能がサポートされるのは、DB2 UDB for Unix および for Windows バージョン 8 クライアントから DB2 UDB for Unix および for Windows バージョン 8 サーバーへ、または DB2 UDB for OS/390 and z/OS サーバー 7 へ、という場合のみです。DB2 UDB for Unix および for Windows バージョン 7 クライアントは、DB2 UDB for Unix および for Windows バージョン 8 サーバーへの既存の両方向スクロール・カーソル機能を引き続きサポートします。

解決方法: サーバーをバージョン 8 にアップグレードします。

DB2 Connect バージョン 8 サーバーを介したバージョン 7 サーバー・アクセス:

Windows	UNIX
---------	------

変更点: バージョン 8 では、DB2 UDB for Unix および for Windows クライアントからバージョン 7 DB2 UDB サーバーへのアクセスは、バージョン 8 サーバーを介してはサポートされません。バージョン 8 サーバーの機能は、DB2 Connect Enterprise Edition バージョン 8 または DB2 UDB Enterprise Server Edition バージョン 8 によって提供されます。

解決方法: サーバーをバージョン 8 にアップグレードします。

CLP または組み込み SQL を使用したタイプ 1 接続:

Windows	UNIX
---------	------

変更点: DB2 の以前のバージョンでは、コマンド行プロセッサ (CLP) または組み込み SQL を使用してタイプ 1 接続でデータベースに接続する場合、作業単位中に他のデータベースに接続しようとする、SQL0752N エラーで失敗しました。バージョン

8 では、作業単位がコミットされると接続はリセットされ、2 番目のデータベースへの接続が許可されます。作業単位はコミットされ、AUTOCOMMIT がオフの場合でも接続はリセットされます。

メッセージ

DB2 メッセージの代わりに戻される DB2 Connect メッセージ:

Windows	UNIX
---------	------

変更点: バージョン 8 では、以前のリリースで DB2 メッセージが戻されるような状況で、新たに DB2 Connect メッセージが戻される場合があります。

例:

- SQLCODE -1224 に代わって SQLCODE -30081 が戻されます
- SQLCODE -1403 に代わって SQLCODE -30082 が戻されます
- SQLCODE -4930 に代わって SQLCODE -30104 が戻されます

症状: DB2 メッセージに対応するアプリケーション・コードは、以前のように動作しなくなる可能性があります。

構成パラメーター

使用されなくなったデータベース・マネージャー構成パラメーター:

Windows	UNIX
---------	------

変更点: 以下のデータベース・マネージャー構成パラメーターは、使用されなくなりました。

- *backbufsz*: 以前のバージョンでは、デフォルト・バッファー・サイズを使ってバックアップ操作を実行でき、*backbufsz* の値がデフォルトと見なされました。バージョン 8 では、バックアップ・ユーティリティーの使用時に、バックアップ・バッファー・サイズを明示的に指定する必要があります。
- *dft_client_adprt*: DCE ディレクトリー・サービスはもはやサポートされません。
- *dft_client_comm*: DCE ディレクトリー・サービスはもはやサポートされません。
- *dir_obj_name*: DCE ディレクトリー・サービスはもはやサポートされません。
- *dir_path_name*: DCE ディレクトリー・サービスはもはやサポートされません。
- *dir_type*: DCE ディレクトリー・サービスはもはやサポートされません。
- *dos_rqrioblk*
- *drda_heap_sz*

- *fcm_num_anchors*、*fcm_num_connect*、および *fcm_num_rqb*: DB2 は新たにメッセージ・アンカー、接続項目、および要求ブロックを動的かつ自動的に調整するようになりました。ユーザーがこれらのパラメーターを調整する必要はありません。
- *fileserv*: IPX/SPX はもはやサポートされません。
- *initdari_jvm*: Java スタート・プロシージャは、デフォルトではマルチスレッドで実行されるようになりました。さらに、他の言語のルーチンとは別個のプロセスで実行されるようになったため、このパラメーターはもはやサポートされません。
- *ipx_socket*: IPX/SPX はもはやサポートされません。
- *jdk11_path*: これは、データベース・マネージャー構成パラメーター *jdk_path* に置き換われました。
- *keepdari*: これは、データベース・マネージャー構成パラメーター *keepfenced* に置き換われました。
- *max_logicagents*: これは、データベース・マネージャー構成パラメーター *max_connections* に置き換われました。
- *maxdari*: これは、データベース・マネージャー構成パラメーター *fenced_pool* に置き換われました。
- *num_initdaris*: これは、データベース・マネージャー構成パラメーター *num_initfenced* に置き換われました。
- *objectname*: IPX/SPX はもはやサポートされません。
- *restbufsz*: 以前のバージョンでは、デフォルト・バッファー・サイズを使ってリストア操作を実行でき、*restbufsz* の値がデフォルトと見なされました。バージョン 8 では、リストア・ユーティリティの使用時に、リストア・バッファー・サイズを明示的に指定する必要があります。
- *route_obj_name*: DCE ディレクトリー・サービスはもはやサポートされません。
- *ss_logon*: これは OS/2 パラメーターであり、OS/2 はもはやサポートされません。
- *udf_mem_sz*: UDF はもはや共有メモリーにデータを渡さないで、このパラメーターはサポートされません。

解決方法: アプリケーションから、これらのパラメーターの参照をすべて除去します。

使用されなくなったデータベース構成パラメーター:

Windows	UNIX
---------	------

変更点: 以下のデータベース構成パラメーターは、使用されなくなりました。

- *buffpage*: 以前のバージョンでは、デフォルト・サイズを使ってバッファー・プールを作成または変更でき、*buffpage* の値がデフォルトと見なされました。バージョン 8

では、ALTER BUFFERPOOL ステートメントまたは CREATE BUFFERPOOL ステートメントで SIZE キーワードを使用することによって、バッファー・プール・サイズを明示的に指定する必要があります。

- *copyprotect*
- *indexsort*

解決方法: アプリケーションから、これらのパラメーターの参照をすべて除去します。

コード・ページ 1394 とユニコード間の変換のために以前の表をインストールする

コード・ページ 1394 (シフト JIS X0213 と呼ばれる) とユニコード間の変換表は、拡張されています。日本語シフト JIS X0213 (1394) とユニコード間の変換は、JIS X0213 文字の最終 ISO/IEC 10646-1:2000 Amendment-1 に準拠します。以前のバージョンの変換表は、<ftp://ftp.software.ibm.com/ps/products/db2/info/vr8/conv/> から FTP で入手可能です。

手順:

シフト JIS X0213 とユニコード間の変換用の以前の定義をインストールするには、以下のようにします。

1. DB2 インスタンスを停止する。
2. Web ブラウザーに <ftp://ftp.software.ibm.com/ps/products/db2/info/vr8/conv/> を指定するか、または FTP を使用して [ftp.software.ibm.com](ftp://ftp.software.ibm.com) サイトに接続する。この FTP サーバーは匿名です。
3. コマンド行から接続している場合は、`anonymous` をユーザー ID、E メール・アドレスをパスワードとしてログインする。
4. ログイン後に、以下の変換表ディレクトリーに移動する。

```
cd ps/products/db2/info/vr8/conv
```
5. 2 つのファイル `1394ucs4.cnv` および `ucs41394.cnv` をバイナリー形式で、`sqllib/conv/` ディレクトリーにコピーする。
6. DB2 インスタンスを再始動する。

関連概念:

- 273 ページの『DB2 でのユニコードのインプリメンテーション』

関連資料:

- 242 ページの『サポートされているテリトリー・コードおよびコード・ページ』

ストレージ管理ビュー

「ストレージ管理」ビューは、パーティション・データベースのストレージ状態をモニターするために使用します。「ストレージ管理」ビューは、ストレージ管理表およびストアード・プロシージャのグラフィカル・インターフェースです。ストレージ管理ビューでは、データベース、データベース・パーティション・グループ、または表スペースのストレージ・スナップショットをとることができます。表スペースのスナップショットをとる場合、指定された表スペースの有効範囲内に定義された表、索引、およびコンテナに関して、システム・カタログおよびデータベース・モニターから統計情報が収集されます。データベースまたはデータベース・パーティション・グループのスナップショットをとると、指定されたデータベースまたはデータベース・パーティション・グループ内に定義されているすべての表スペースに関して、統計情報が収集されます。データベースのスナップショットをとると、そのデータベース内のすべてのデータベース・パーティション・グループに関して、統計情報が収集されます。以下のような異なるタイプのストレージ・スナップショットがあり、ストレージの異なった面をモニターするのに役立ちます。

- スペース使用量は、表スペースのスナップショットを使用してモニターできます。
- パーティション・データベースの場合のみ: データ・スキュー (データベース分散) は、データベース・パーティション・グループのスナップショットを介した場合に最適なモニターが可能です。
- 索引のクラスター化率は、データベース・パーティション・グループ・スナップショットと表スペース・スナップショットの両方でキャプチャーすることができます。索引のクラスター化率は、索引フォルダーの詳細ビューを介して提供されます。

「ストレージ管理」ビューによって、データ・スキュー、スペース使用量、および索引クラスター化率に関して、しきい値を設定することもできます。ターゲット・オブジェクトが指定されたしきい値を超えると、「ストレージ管理」ビューの中のそのオブジェクトとその親オブジェクトのアイコンに、警告フラグまたはアラーム・フラグのマークが付けられます。

注: パーティション・データベースのデータ・スキューしきい値のみをモニターできません。

関連資料:

- 377 ページの『ストレージ管理ツール用のストアード・プロシージャ』
- 377 ページの『ストレージ管理ビュー表』

DB2 Administration Server

DB2[®] Administration Server (DAS) は、DB2 サーバーでの作業を専門的に支援するためのコントロール・ポイントです。構成アシスタント、コントロール・センター、または

デベロップメント・センターなどのツールを使用するには、DAS を実行していなければなりません。DAS は、以下の管理タスクを処理するときにはコントロール・センターおよび構成アシスタントを援助します。

- DB2 サーバーをリモートに管理できる。
- DB2 とオペレーティング・システム・コマンド・スクリプトの両方の実行をスケジュールする機能も含めた、ジョブ管理用の機能を提供する。これらのコマンド・スクリプトはユーザーが定義します。
- タスク・センターを使用して、ジョブ・スケジュールの定義、完了したジョブの結果表示、および DAS にとってリモートまたはローカルに置かれているジョブに対する他の管理タスクの実行を行います。
- DB2 ディスカバリー・ユーティリティーと共に、DB2 インスタンス、データベースおよび他の DB2 Administration Server の構成についての情報を検出するための手段を提供する。この情報は、DB2 データベースへのクライアント接続の構成を単純化して自動化するために、構成アシスタントとコントロール・センターが使用します。

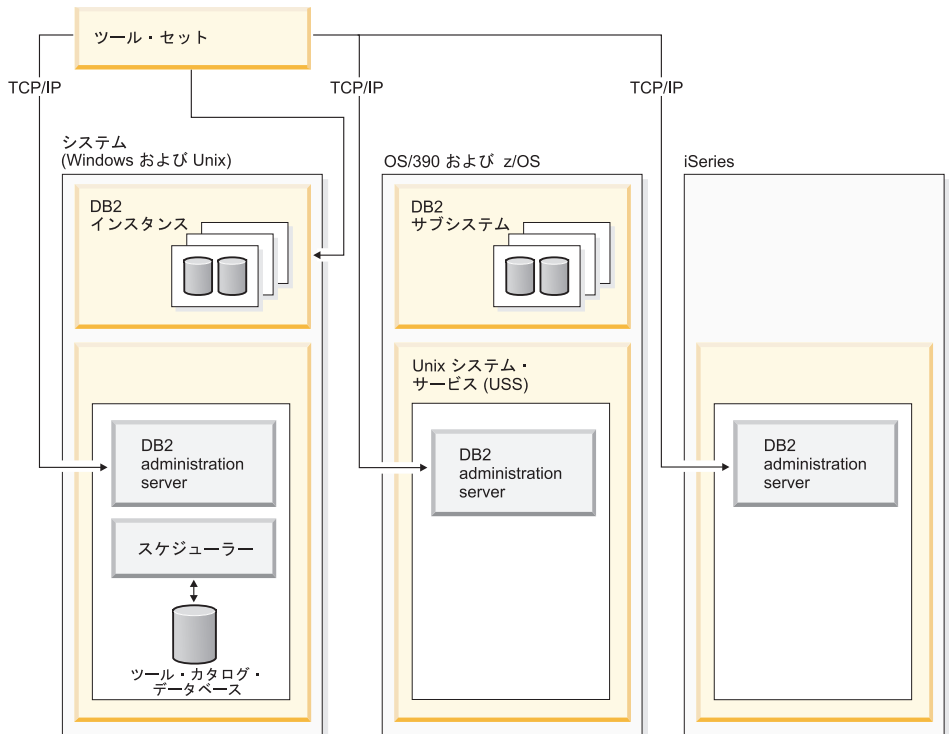


図 1. DAS が使用される場所

1 つのマシン上には 1 つの DAS しか持つことができません。オペレーティング・システムのブート時に、DAS を開始するために DAS の構成がインストール中に行われます。

DAS の用途は、コントロール・センター、構成アシスタント、その他のツールからのクライアント要求を扱うために、サーバー・システムおよびホスト・システムでリモート・タスクを実行することです。

DAS は、zSeries™ (OS/390 および z/OS™ のみ) プラットフォームだけではなく、サポートされるすべての Windows® および UNIX® で使用できます。zSeries 上で DAS を使用すれば、コントロール・センター、デベロップメント・センター、およびレプリケーション・センターの管理タスクがサポートされます。

zSeries (OS/390 および z/OS のみ) 用の DB2 Administration Server は、DB2 管理のクライアント機能の一部としてパッケージされ、提供されます。DAS を必要とする製品 (コントロール・センター、レプリケーション・センター、デベロップメント・センターなど) では、DAS 機能をインストールする必要があります。ご使用のオペレーティング・システムでの DAS の利用について、詳しくは IBM® 担当員にお問い合わせください。

Windows および UNIX 用の DAS には、タスク・センターを使って定義されるタスク (たとえば、DB2 およびオペレーティング・システムのコマンド・スクリプト) を実行するためのスケジューラーが含まれます。実行されるコマンドなどに関するタスク情報、すなわちタスクに関連したスケジュール、通知、および完了アクション、およびタスクの実行結果が、ツール・カタログと呼ばれる DB2 データベース内の表およびビューのセットに保管されます。ツール・カタログは、セットアップの一部として作成されます。さらに、コントロール・センターで、あるいは CLP で **CREATE TOOLS CATALOG** コマンドを使って、これを作成して活動化することもできます。

スケジューラーは zSeries (OS/390 および z/OS のみ) では提供されませんが、コントロール・センターの「JCL のビルド」および「JCL の作成」機能を使用して JCL を生成し、パーティション・データベースに保管して、システム・スケジューラーで実行することができます。

関連概念:

- 管理ガイド: インプリメンテーションの『Windows での DAS のセキュリティに関する考慮事項』
- 337 ページの『Enterprise Server Edition (ESE) システムの DAS 構成』
- 管理ガイド: インプリメンテーションの『DAS とのコントロール・センター通信: サービス・ポート』
- 管理ガイド: インプリメンテーションの『ノード間管理通信: Windows DB2 ESE』
- 管理ガイド: インプリメンテーションの『Administration Server、インスタンス、およびデータベースのディスカバリー』

関連タスク:

- 管理ガイド: インプリメンテーションの『DB2 Administration Server の作成』
- 管理ガイド: インプリメンテーションの『DAS の開始と停止』
- 管理ガイド: インプリメンテーションの『DAS のリスト』
- 管理ガイド: インプリメンテーションの『DAS の構成』
- 管理ガイド: インプリメンテーションの『UNIX での DAS の更新』
- 管理ガイド: インプリメンテーションの『DAS の除去』
- 管理ガイド: インプリメンテーションの『Enterprise Server Edition (ESE) システムでの DAS のセットアップ』
- 管理ガイド: インプリメンテーションの『ディスカバリーからサーバー・インスタンスおよびデータベースを隠す』
- 管理ガイド: インプリメンテーションの『ディスカバリー・パラメーターの設定』
- 管理ガイド: インプリメンテーションの『構成アシスタントおよびコントロール・センターを使用するための DAS の設定』
- 管理ガイド: インプリメンテーションの『ディスカバリー用の DAS 構成の更新』
- 327 ページの『ツール・カタログ・データベースおよび DAS スケジューラーのセットアップと構成』
- 管理ガイド: インプリメンテーションの『通知、連絡先リストのセットアップ、および構成』
- 管理ガイド: インプリメンテーションの『DAS Java Virtual Machine セットアップ』

環境変数およびプロファイル・レジストリー

環境変数およびレジストリー変数は、データベース環境を制御します。

構成アシスタント (**db2ca**) を使って、構成パラメーターとレジストリー変数を構成することができます。

DB2[®] プロファイル・レジストリーの導入前は、(たとえば) Windows[®] ワークステーションの環境変数を変更するためには環境変数を変更してリポートする必要があります。いくつかの例外はありますが、現在、環境は DB2 プロファイル・レジストリーに保管されているレジストリー変数によって制御されます。UNIX[®] オペレーティング・システムで特定のインスタンスに関するシステム管理 (SYSADM) 権限を持つユーザーは、そのインスタンスのレジストリー値を更新することができます。Windows ユーザーの場合、レジストリー変数を更新するために SYSADM 権限は必要ありません。リポートせずにレジストリー変数を更新するには、**db2set** コマンドを使用します。この情報は、即時にプロファイル・レジストリーの中に保管されます。DB2 レジストリーは、変更を行った後に開始した DB2 サーバー・インスタンスと DB2 アプリケーションに更新情報を適用します。

レジストリーを更新する場合、現在実行中の DB2 アプリケーションまたはユーザーに影響はありません。更新後に開始されたアプリケーションは新しい値を使用します。

注: DB2 環境変数 DB2INSTANCE および DB2NODE は、DB2 プロファイル・レジストリーに保管されない場合があります。一部のオペレーティング・システムでは、これらの環境変数を更新するために **set** コマンドを使用する必要があります。変更は、次回システムをリブートするまで有効です。UNIX プラットフォームでは、**set** コマンドの代わりに **export** コマンドを使用できます。

プロファイル・レジストリーを使用することによって、環境変数の中央制御が可能になります。さまざまなレベルのサポートが、さまざまなプロファイルを通して提供されるようになっていきます。環境変数のリモートでの管理も、DB2 Administration Server を使用すれば可能です。

以下の 4 つのプロファイル・レジストリーがあります。

- DB2 インスタンス・レベル・プロファイル・レジストリー。DB2 環境変数の大多数は、このレジストリーの中に置かれます。特定のインスタンスについての環境変数の設定値が、このレジストリーに保持されます。このレベルに定義された値は、グローバル・レベルの設定値をオーバーライドします。
- DB2 グローバル・レベル・プロファイル・レジストリー。特定のインスタンスごとに 1 つの環境変数が設定されるのではない場合、このレジストリーが使用されます。このレジストリーは、マシン全体にわたる環境変数設定値を持ちます。DB2 UDB ESE では、マシンごとに 1 つのグローバル・レベル・プロファイルが存在します。
- DB2 インスタンス・ノード・レベル・プロファイル・レジストリー。このレジストリー・レベルには、複数パーティション環境のパーティション (ノード) に関する変数設定値が含まれます。このレベルで定義された値は、インスタンス・レベルおよびグローバル・レベルの設定値をオーバーライドします。
- DB2 インスタンス・プロファイル・レジストリー。このレジストリーには、このシステムによって認識されるすべてのインスタンス名のリストが含まれます。db2ilist を実行すれば、システムで使用できるすべてのインスタンスがリストされます。

DB2 は、レジストリー値と環境変数を検査し、それらを以下の順序で解決することによって、操作環境を構成します。

1. **set** コマンドを使用して設定された環境変数。(UNIX プラットフォームでは **export** コマンド。)
2. インスタンス・ノード・レベル・プロファイルを使用して設定されたレジストリー値 (コマンド **db2set -i <instance name> <nodenum>** を使用)。
3. インスタンス・レベル・プロファイルを使用して設定されたレジストリー値 (コマンド **db2set -i** を使用)。
4. グローバル・レベル・プロファイルを使用して設定されたレジストリー値 (コマンド **db2set -g** を使用)。

インスタンス・レベル・プロファイル・レジストリー

パーティション・データベース環境で作業を行っている場合、UNIX と Windows では、いくつかの違いがあります。これらの違いについて、以下に例を示します。

「red」、「white」、および「blue」で識別される 3 つの物理ノードを持つ、パーティション・データベース環境があると想定します。UNIX プラットフォームでは、インスタンス所有者がいずれかのノードから次を実行した場合、

```
db2set -i FOO=BAR
```

または

```
db2set FOO=BAR      ('-i' is implied)
```

FOO の値は現行インスタンスのすべてのノード（「red」、「white」、および「blue」）に対して可視になります。

UNIX プラットフォームでは、インスタンス・レベル・プロファイル・レジストリーは、sqllib ディレクトリー内のテキスト・ファイルに保管されます。パーティション・データベース環境では、sqllib ディレクトリーはすべての物理ノードによって共有されているファイル・システムにあります。

Windows プラットフォームでは、ユーザーが「red」から同じコマンドを実行した場合、FOO の値は現行インスタンスの「red」でのみ可視になります。DB2 はインスタンス・レベル・プロファイル・レジストリーを Windows レジストリーに保管します。物理ノード間に共有はありません。レジストリー変数をすべての物理マシンに設定するには、以下のように「rah」コマンドを使用します。

```
rah db2set -i FOO=BAR
```

rah はリモートで db2set コマンドを「red」、「white」、および「blue」上で実行します。

DB2REMOTEPEG を使用して、インスタンスを所有していないマシン上のレジストリー変数を構成することにより、インスタンスを所有しているマシン上のレジストリー変数を参照することができます。これにより、インスタンスを所有しているマシン上のレジストリー変数が、インスタンス内のすべてのマシン間で共有される環境が効果的に作成されます。

上に示された例を使用し、「red」が所有マシンと想定すると、次を行うことで、「red」上のレジストリー変数を共有するために、DB2REMOTEPEG が「white」および「blue」マシンに設定されます。

```
(on red) do nothing
(on white and blue) db2set DB2REMOTEPEG=¥¥red
```

DB2REMOTEPEG の設定値は、設定後に変更できません。

以下に REMOTEPEG がどのように動作するかを示します。

DB2 が Windows でレジストリー変数を読み取る場合、最初に DB2REMOTEPREG 値を読み取ります。DB2REMOTEPREG が設定されている場合、DB2REMOTEPREG 変数に指定されているマシン名のリモート・マシン上のレジストリーをオープンします。その後のレジストリー変数の読み取りおよび更新は、指定されたりモート・マシンにリダイレクトされます。

リモート・レジストリーへのアクセスには、ターゲット・マシンで Remote Registry Service が実行されている必要があります。また、ユーザーのログオン・アカウントおよびすべての DB2 サービス・ログオン・アカウントに、リモート・レジストリーへの十分なアクセス権限が必要です。そのため、DB2REMOTEPREG を使用するには、Windows ドメイン環境で操作を行う必要があります。これによって、必要なレジストリー・アクセスが、ドメイン・アカウントに付与されます。

Microsoft® Cluster Server (MSCS) には考慮事項があります。DB2REMOTEPREG を MSCS 環境では使用できません。すべてのマシンが同じ MSCS クラスタに属する MSCS 構成で稼働している場合、レジストリー変数はクラスタ・レジストリー内に保持されます。そのため、それらはすでに同じ MSCS クラスタ内のすべてのマシン間で共用されており、この場合は DB2REMOTEPREG を使用する必要はありません。

パーティションが複数の MSCS サーバーに及ぶ、マルチパーティション・フェイルオーバー環境で稼働している場合、インスタンスを所有しているマシンのレジストリー変数はこのクラスタ・レジストリーにあるため、インスタンスを所有しているマシンを指すのに、DB2REMOTEPREG は使用できません。

関連概念:

- 管理ガイド: パフォーマンス の『DB2 レジストリー変数と環境変数』

関連タスク:

- 管理ガイド: インプリメンテーション の『レジストリーおよび環境変数の宣言』

サンプル・プラグインのコンパイルおよび実行

コントロール・センター用のプラグインの機能は、これ以降のセクションと、対応するプラグイン・サンプル・プログラム (Example1.java, Example2.java, Example3.java, Example3Folder.java, Example4.java, Example5.java, および Example6.java) に例示されています。これらのサンプルの Java ファイルは、DB2® Application Development Client と一緒にインストールされます。Windows® プラットフォームでは、これらのサンプル・プログラムは、DRIVE:%sqllib%samples%java%plugin にあります。ここで DRIVE: は、DB2 のインストール先のドライブを表します。UNIX® プラットフォームでは、これらのサンプルは、/u/db2inst1/sqllib/samples/java/plugin にあります。ここで /u/db2inst1 は、DB2 のインストール先のディレクトリーを表します。

注: プラグイン・サンプル・プログラムには、ここにはまだ反映されていない更新が含まれている場合があります。サンプル・コードおよび Java ドキュメンテーションが、ここに表示されているものと違いがある場合、最新の情報と考えられます。

サンプルのプラグインを実行するには、Java™ アーカイブ・ファイルの規則にしたがって、拡張クラス・ファイルを ZIP する必要があります。ZIP ファイル (db2plug.zip) は *classpath* 内に置いておかなければなりません。Windows オペレーティング・システムでは、db2plug.zip を `DRIVE:¥sqllib¥tools` ディレクトリーに置いてください。ここで `DRIVE:` は、DB2 のインストール先のドライブを表します。UNIX プラットフォームでは、db2plug.zip を `/u/db2inst1/sqllib/tools` ディレクトリーに置いてください。ここで `/u/db2inst1` は、DB2 のインストール先のディレクトリーを表します。

注: **db2cc** コマンドは、*classpath* を `tools` ディレクトリー内の `db2plug.zip` を指すように設定します。

サンプル (Example3、Example3Folder および Example3Child を除く) は、互いに競合する場合がありますので同じ `db2plug.zip` に ZIP してはなりません。

これらのサンプルの Java ファイルをコンパイルするには、以下を *classpath* に組み込む必要があります。

- Windows プラットフォームでは、以下を使用します。

- `DRIVE: ¥sqllib¥java¥Common.jar`
- `DRIVE: ¥sqllib¥tools¥db2cc.jar`

ここで `DRIVE` は、DB2 のインストール先のドライブを表します。

- UNIX プラットフォームでは、以下を使用します。

- `/u/db2inst1/sqllib/java/Common.jar`
- `/u/db2inst1/sqllib/tools/db2cc.jar`

ここで `/u/db2inst1` は、DB2 のインストール先のディレクトリーを表します。

`db2plug.zip` を作成して、サンプルの Java ファイルをコンパイルすることにより生成されるすべてのクラスを組み込みます。このファイルは圧縮しないようにします。たとえば、以下を発行します。

```
zip -r0 db2plug.zip *.class
```

このコマンドを発行すると、全クラス・ファイルが `db2plug.zip` ファイルに入れられ、相対パス情報が保存されます。

関連概念:

- 管理ガイド: インプリメンテーションの『コントロール・センターの拡張機能としてのプラグインの作成』
- 339 ページの『コントロール・センター用プラグインの開発者向けのガイドライン』

関連資料:

- コマンド・リファレンス の『db2cc - コントロール・センターの開始コマンド』

ロックとパフォーマンス

いくつかの関連要素が、ロックの使用およびアプリケーション・パフォーマンスに影響を与えます。この部分では、以下の点について説明します。

- 並行性と細分性
- ロックの互換性
- ロックの変換
- ロック・エスカレーション
- ロックの待機とタイムアウト
- デッドロック

並行性と細分性

1 つのアプリケーションがデータベース・オブジェクト上のロックを保持している場合、別のアプリケーションはそのオブジェクトにアクセスできません。そのため、行レベルのロックは、表レベルのロックよりも最大の並行性が得られます。ただし、ロックには、ストレージや処理時間が必要なので、単一の表ロックはロック・オーバーヘッドを最小化します。

ALTER TABLE ステートメントの LOCKSIZE 文節は、行または表レベルのいずれかでロックの有効範囲 (細分性) を指定します。デフォルトでは、行ロックが使用されます。S (共用) および X (排他) ロックのみがこれらの定義済みの表ロックによって要求されます。ALTER TABLE ステートメントの LOCKSIZE ROW 文節を使用しても、通常のロック・エスカレーションに支障はありません。

以下のような場合には、LOCK TABLE ステートメントを使用して単一トランザクション表をロックするよりも、ALTER TABLE を使用して永続表ロックをすることができます。

- 使用している表が読み取り専用であり、S ロックが必ず必要な場合。その他のユーザーは、表に S ロックを獲得することもできます。
- 表は、通常、読み取り専用アプリケーションによってアクセスされますが、時折、簡単な保守のために単一のユーザーによってアクセスされます。したがって、そのユーザーには X ロックが必要です。保守プログラムが実行されている間、読取専用アプリケーションはロックアウトされますが、その他の環境では、読み取り専用アプリケーションは、最小のロッキング・オーバーヘッドで並行して表にアクセスすることができます。

ALTER TABLE ステートメントは、グローバルにロックを指定するので、その表にアクセスするすべてのアプリケーションおよびユーザーが影響を受けます。個々のアプリケ

ーションは、その代わりにアプリケーション・レベルで表ロックを指定するために LOCK TABLE ステートメントを使用することができます。

ロックの互換性

ロックの互換性は、表の並行アクセスにおける別の重要な要素です。ロックの互換性は、オブジェクト上の現行ロックおよびそのオブジェクトで要求されているロックのタイプを表しており、要求を許可できるかどうかを判別します。

アプリケーション A がある表のロックを保持していて、その表へのアクセスをアプリケーション B も待っているとします。データベース・マネージャーはアプリケーション B のために、ある特定のモードのロックを要求します。A が保持しているロックのモードが、B の要求しているロックを許可するものである場合、2 つのロック (またはモード) は、互換性があると言います。

アプリケーション B が要求したロック・モードとアプリケーション A が保持しているロックとの間に互換性がない場合には、アプリケーション B はそれ以上継続できません。その代わりに、アプリケーション A がロックを解放し、さらに既存の非互換のロックがすべて解放されるまで待機する必要があります。

ロックの変換

すでに保持しているロック・モードの変更は、変換 と呼ばれます。ロックの変換が行われるのはあるプロセスがすでにロックを保持しているデータ・オブジェクトにアクセスする場合に、そのアクセスのモードが、すでに保持しているロックよりさらに制約の大きいロックを必要とするものである場合です。照会によって間接的に 1 プロセスの中で同じデータ・オブジェクトに、何度もロックを要求できますが、1 つのデータ・オブジェクトのロックは 1 回に 1 つしか保持できません。

一部のロック・モードは表にのみ適用され、その他のロック・モードは行またはブロックにのみ適用されます。行またはブロックの場合、通常、X が必要なときに S または U (更新) を保持している場合に、変換が行われます。

しかし、IX (意図排他) と S (共用) ロックは、ロック変換に関しては特殊なケースです。S と IX はどちらも他方よりも制約が大きいとは見なされないため、これらのロックの一方を保持しているときに他方が必要になった場合には、結果として SIX (意図排他共用) ロックに変換されることになります。他のすべての変換の結果は、要求されているモードの制限がより大きい場合は、要求されたロック・モードが、保持するロックのモードになります。

照会が行を更新するとき、二重変換が発生する場合があります。索引アクセスによって行が読み取られ、S としてロックされる場合、行を含む表には、目的ロックが含まれています。ただし、ロック・タイプが IX ではなく IS である場合、後で行が変更されると、表ロックは IX に変換され、行ロックは X に変換されることになります。

ロック変換は、通常、照会が実行されるときに暗黙的に行われます。各種の照会および表と索引の組み合わせの下で発行されるロックの種類について知っておくことは、アプリケーションの設計と調整に役立つでしょう。

ロック・エスカレーション

ロック・エスカレーションは、保持されるロックの数を減らすための内部機構です。単一表では、ロックは多数の行ロックから表ロックにエスカレーションされ、多次元クラスタリング (MDC) 表では、多数の行またはブロック・ロックから表ロックにエスカレーションされます。ロック・エスカレーションは、アプリケーションのロック・タイプの保持が多すぎるときに起こります。ロック・エスカレーションは、特定のデータベース・エージェントがロック・リストのそのエージェントに割り振られた分を超過した場合にも、起こることがあります。このようなエスカレーションは内部的に処理されるので、1 つまたは複数の表への並行アクセスが少なくなるのは、外部から検出できる唯一の結果です。

適正に構成されたデータベースでは、ロック・エスカレーションはほとんど行われません。ロック・エスカレーションが起きる場合の例としては、アプリケーション設計者が大きな表で索引を作成してパフォーマンスと並行性を向上させようとしたが、そのトランザクションはその表の大部分のレコードにアクセスするというような場合があげられます。この場合、データベース・マネージャーは、表の大部分がロックされることを予測できないため、S または X の表だけをロックするのではなく、各レコードを個別にロックします。このような場合の解決策として、データベース設計者は、アプリケーション設計者とも相談した上で、このトランザクションに LOCK TABLE ステートメントを使用するように推奨することができます。

時折、内部エスカレーション要求を受信するプロセスは、表にほとんど行ロックを保持していないか全く行ロックを保持していても、1 つ以上のプロセスが多数のロックを保持しているためにロックがエスカレーションされることがあります。そのプロセスは、トランザクション終了時を除いて、別のロックを要求したりデータベースにアクセスしたりしないかもしれませんが。そして、別のプロセスが、エスカレーション要求を引き起こすロックを要求する場合があります。

注: ロック・エスカレーションは、デッドロックを引き起こすこともあります。たとえば、読み取り専用アプリケーションと更新アプリケーションが両方とも同じ表にアクセスするとします。更新アプリケーションが表の多数の行に排他ロックを持っている場合、データベース・マネージャーはこの表のロックを排他表ロックにエスカレーションしようとしています。ただし、読み取り専用アプリケーションが保持している表ロックによって、排他ロックのエスカレーション要求が待機させられることがあります。更新アプリケーションがすでにロックしている行に、読み取り専用アプリケーションが行ロックを要求する場合、デッドロックが作成されます。この種類の問題を回避するには、開始時に、表を排他的にロックするように更新アプリケーションをコーディングするか、またはロック・リストのサイズを大きくしてください。

ロックの待機とタイムアウト

ロックのタイムアウト検出は、ロックが異常な状況で解放されるまで無限にアプリケーションが待機しなくて済むようにするデータベース・マネージャーの機能です。たとえば、あるトランザクションが別のユーザーのアプリケーションによって保持されているロックを待機しており、一方でそのユーザーがトランザクションをアプリケーションがコミットできるようにロックを解放するということをせずに作業場から席をはずしてしまっているかもしれません。こうしたケースでアプリケーションが停止しないようにするには、*locktimeout* 構成パラメーターを使用して、アプリケーションがロックを獲得するまで待機する最大待ち時間を設定します。

このパラメーターを設定すると、特に分散作業単位 (DUOW) アプリケーションにおいては、グローバル・デッドロックを避けることができます。ロック要求が保留にされている時間が *locktimeout* 値より長くなると、要求しているアプリケーションはエラーを受け取り、トランザクションがロールバックされます。たとえば、*program1* が、すでに *program2* によって保持されているロックを獲得しようとするときにタイムアウトになると、*program1* は *SQLCODE -911* と理由コード *68* を返します。*locktimeout* のデフォルト値は *-1* です。これにより、ロックのタイムアウト検出はオフになります。

管理通知ログ内にロック要求タイムアウトの詳細をログに記録するには、データベース・マネージャーの構成パラメーター *notifylevel* を *4* に設定します。ログに記録された情報には、オブジェクト、ロック・モード、およびロックを保持しているアプリケーションが含まれます。また、現行の動的 SQL ステートメントまたは静的パッケージ名もログに記録されている可能性があります。動的 SQL ステートメントは、*notifylevel* が *4* である場合にのみログに記録されます。

デッドロック

ロックの競合により、デッドロックが生じます。たとえば、プロセス *1* が *X* (排他) モードで表 *A* にロックをかけ、プロセス *2* が *X* モードで表 *B* にロックをかけるとします。次にプロセス *1* が *X* モードで表 *B* にロックをかけようとし、さらにプロセス *2* が *X* モードで表 *A* にロックをかけようとする、それらのプロセスはデッドロックに陥ります。デッドロック状態になると、プロセスは両方とも、それらの *2* 度目のロック要求が受け入れられるまで中断状態になり、しかもどちらの要求も、いずれかのプロセスがコミットまたはロールバックを実行するまで受け入れられることはありません。外部エージェントがいずれかのプロセスを活性化し、強制的にロールバックを実行させるまで、この状態が無限に続きます。

デッドロックを処理するには、データベース・マネージャーは、デッドロック検出機能と呼ばれる非同期システム背景プロセスを使用します。デッドロック検出機能は、*dlchktme* 構成パラメーターによって指定された周期でアクティブになります。デッドロック検出機能がアクティブになると、ロック・システムがデッドロック状態になっていないかどうかを調べます。パーティション・データベースでは、それぞれのパーティシ

ョンは、システム・カタログ・ビューを含むデータベース・パーティションにロック・グラフを送信します。グローバルなデッドロック検出は、このパーティション上で行われます。

デッドロックが見つかる場合、デッドロック検出機能は、1つのデッドロック・プロセスを、ロールバックする選択されたプロセスとして選択します。選択されたプロセスはアクティブにされ、呼び出し側アプリケーションに `SQLCODE -911 (SQLSTATE 40001)` 理由コード 2 で戻されます。データベース・マネージャーは選択されたプロセスを自動的にロールバックします。ロールバックが完了すると、選択されたプロセスに属するロックは解除され、それによってそのデッドロックにかかわっていた他のプロセスが先に進めるようになります。

デッドロック検出機能の適切な時間間隔を選択することは、適正なパフォーマンスを確保するうえで必要なことです。時間間隔が短すぎると不必要なオーバーヘッドが生じ、長すぎるとデッドロックによるプロセスの遅延が長くなってしまいます。たとえば、5分というウェイクアップ間隔を選択すると、デッドロックはほぼ5分間存在することができます。これは、短いトランザクション処理の場合には長い時間のように思われます。デッドロックを解決するための遅延と、デッドロックを検出するオーバーヘッドとの間でバランス[®]を保つ必要があります。

パーティション・データベースでは、`dlchktme` 構成パラメーター間隔は、カタログ・ノードにのみ適用されます。大量のデッドロックがパーティション・データベースで検出される場合には、`dlchktme` パラメーターの値を大きくして、ロック待機や通信待機を解決するようにしてください。

データベースにアクセスする独立したプロセスが複数個あるアプリケーションが、デッドロックが生じやすい構造になっている場合は、別の問題が起きる可能性があります。その一例は、いくつかのプロセスが同じ表にアクセスして、読み取りを行ってから書き込みを行うようになっているアプリケーションです。それらのプロセスが最初に読み取り専用 SQL 照会を行い、次に同じ表に対する SQL 更新を行うと、プロセス相互間で同じデータに対する競合が生じる可能性があるため、デッドロックの起きる可能性が大きくなります。たとえば、2つのプロセスが同じ表を読み取った後でその表の更新を行うと、プロセス A がある行に対する X ロックを入手しようとしているが、プロセス B がその行に対する S ロックを持っている（あるいは、その反対）という状況になります。こうしたデッドロックを避けるため、修正する目的でデータにアクセスするアプリケーションは、選択を実行するときに `FOR UPDATE OF` 文節を使用するようにしてください。それによって、プロセス A がデータを読み取ろうとすると、U ロックがかけられるようになります。

注: デッドロックが生じるときを記録するモニターを定義することを考慮する場合があります。SQL ステートメント `CREATE EVENT` を使用して、モニターを作成してください。

データベースが作成されると同時に、詳細デッドロック・イベント・モニターも作成されます。他のモニターと同様に、このイベント・モニターにも関連したオーバーヘッドがあります。詳細デッドロック・イベント・モニターを必要としない場合は、以下のコマンドを使用してイベント・モニターをドロップできます。

```
DROP EVENT MONITOR db2detaildeadlock
```

このイベント・モニターが消費するディスク・スペースの量を制限するために、出力ファイルの最大数に達すると、イベント・モニターが非アクティブになり、メッセージが管理通知ログに書き込まれます。必要のない出力ファイルを除去すると、イベント・モニターは次のデータベースの活動化時にアクティブになります。

アプリケーションがニックネームにアクセスする連合システム環境では、アプリケーションによって要求されたデータが、データ・ソースでデッドロックが起きているために使用できない可能性があります。このことが起こると、DB2® はデータ・ソースでのデッドロック処理機能により、ロックを解決します。複数のデータ・ソースにまたがるデッドロックが生じる場合は、DB2 はデータ・ソースのタイムアウト機構により、デッドロックを解除します。

デッドロックに関する詳細をログに記録するには、データベース・マネージャーの構成パラメーター *notifylevel* を 4 に設定します。管理通知ログには、オブジェクト、ロック・モード、およびこのオブジェクトへのロックを保持しているアプリケーションを含む情報が保管されます。また、現行の動的 SQL ステートメントまたは静的パッケージ名もログに記録されている可能性があります。動的 SQL ステートメントは、*notifylevel* が 4 である場合にのみログに記録されます。

関連概念:

- 管理ガイド: パフォーマンス の『ロックおよび並行性の制御』
- 管理ガイド: パフォーマンス の『アプリケーション間のデッドロック』

関連タスク:

- 管理ガイド: パフォーマンス の『ロック・エスカレーション問題の修正』

関連資料:

- 管理ガイド: パフォーマンス の『「診断エラー・キャプチャー・レベル」構成パラメーター - *diaglevel*』
- 管理ガイド: パフォーマンス の『「ロック・タイムアウト」構成パラメーター - *locktimeout*』

カタログ統計の表

以下の表では、カタログ統計を含むシステム・カタログ表と、特定の統計を収集する RUNSTATS オプションに関する情報が提供されています。

表 98. 表統計 (SYSCAT.TABLES と SYSSTAT.TABLES)

統計	説明	RUNSTATS オプション	
		表	索引
FPAGES	表が使用しているページの数	○	○
NPAGES	行が含まれているページの数	○	○
OVERFLOW	オーバフローしている行の数	○	×
CARD	表内の行の数 (カーディナリティー)	○	○ (注 1)
ACTIVE_BLOCKS	MDC 表の場合、占有されているブロックの合計数	○	×

注:

1. 表に定義された索引がないときに、索引の統計を要求した場合には、CARD 統計が新たに更新されることはありません。前の CARD 統計は引き続き保持されます。

表 99. 列統計 (SYSCAT.COLUMNS と SYSSTAT.COLUMNS)

統計	説明	RUNSTATS オプション	
		表	索引
COLCARD	列カーディナリティー	○	○ (注 1)
AVGCOLLEN	列の平均長	○	○ (注 1)
HIGH2KEY	列内で 2 番目に高い値	○	○ (注 1)
LOW2KEY	列内で 2 番目に低い値	○	○ (注 1)
NUMNULLS	列内の NULL の数	○	○ (注 1)
SUB_COUNT	サブエレメントの平均数	○	× (注 2)
SUB_DELIM_LENGTH	各サブエレメントを分ける各区切り文字の平均長	○	× (注 2)

注:

1. 列統計は、索引キーの中の最初の列に関して収集されます。
2. これらの統計では、空白で区切られている一連のサブフィールドまたはサブエレメントを含む列の、データに関する情報が提供されています。SUB_COUNT および SUB_DELIM_LENGTH 統計が収集されるのは、タイプ CHAR、VARCHAR、GRAPHIC、および VARGRAPHIC の、1 バイト文字セットの文字列の場合だけです。

表 100. 複数列統計 (SYSCAT.COLGROUPS および SYSSTAT.COLGROUPS)

統計	説明	RUNSTATS オプション	
		表	索引
COLGROUPCARD	列グループのカーディナリティー	○	×

注: 以下の 2 つの表でリストされている複数列の分布統計は、RUNSTATS では収集されません。ただし、手動で更新することはできます。

表 101. 複数列の分布統計 (SYSCAT.COLGROUPDIST および SYSSTAT.COLGROUPDIST)

統計	説明	RUNSTATS オプション	
		表	索引
TYPE	F = 頻出値 Q = 変位値	○	×
ORDINAL	グループ内の列の序数	○	×
SEQNO	n 番目の TYPE 値を表す、シーケンス番号 n 。	○	×
COLVALUE	文字リテラルまたは NULL 値としてのデータ値	○	×

表 102. 複数列の分布統計 2 (SYSCAT.COLGROUPDISTCOUNTS および SYSSTAT.COLGROUPDISTCOUNTS)

統計	説明	RUNSTATS オプション	
		表	索引
TYPE	F = 頻出値 Q = 変位値	○	×
SEQNO	n 番目の TYPE 値を表す、シーケンス番号 n 。	○	×

表 102. 複数列の分布統計 2 (SYSCAT.COLGROUPDISTS および SYSSTAT.COLGROUPDISTS) (続き)

統計	説明	RUNSTATS オプション	
		表	索引
VALCOUNT	<p>TYPE = F の場合、 VALCOUNT は、この SEQNO によって識別される列グループ での、COLVALUE のオカレン スの数です。</p> <p>TYPE = Q の場合、 VALCOUNT は、この SEQNO の列グループで、値が COLVALUE 以下である行の数 です。</p>	○	×
DISTCOUNT	<p>TYPE = Q の場合、この列に は、この SEQNO の列グループ での、COLVALUE 以下の個別 値の数が含まれています。利用 不能の場合は NULL です。</p>	○	×

表 103. 索引統計 (SYSCAT.INDEXES と SYSSTAT.INDEXES)

統計	説明	RUNSTATS オプション	
		表	索引
NLEAF	索引リーフ・ページの数	×	○
NLEVELS	索引レベルの数	×	○
CLUSTERRATIO	表データのクラスタリングの程 度	×	○ (注 2)
CLUSTERFACTOR	クラスタリングの詳細の程度	×	詳細説明 (注 1、2)
DENSITY	索引の対象となるページ範囲に あるページ数に対する SEQUENTIAL_PAGES の比率 (パーセンテージ) (注 3)	×	○
FIRSTKEYCARD	索引の最初の列の個別値の数	×	○
FIRST2KEYCARD	索引の最初の 2 つの列の個別値 の数	×	○
FIRST3KEYCARD	索引の最初の 3 つの列の個別値 の数	×	○

表 103. 索引統計 (SYSCAT.INDEXES と SYSSTAT.INDEXES) (続き)

統計	説明	RUNSTATS オプション	
		表	索引
FIRST4KEYCARD	索引の最初の 4 つの列の個別値の数	×	○
FULLKEYCARD	索引のすべての列の個別値の数 (ただし、すべての RID が削除対象としてマークされているタイプ 2 索引のキー値はすべて除く)	×	○
PAGE_FETCH_PAIRS	異なるバッファ・サイズでのページ取り出し見積もり	×	詳細説明 (注 1、2)
SEQUENTIAL_PAGES	索引キーの順序で、間をあまり空けずにディスクに位置づけられたリーフ・ページの数	×	○
AVERAGE_SEQUENCE_PAGES	順次にアクセスできる索引ページの平均数。これは、プリフェッチャーが順次として検出できる索引ページの数です。	×	○
AVERAGE_RANDOM_PAGES	順次ページ・アクセスの間のランダム索引ページの平均数	×	○
AVERAGE_SEQUENCE_GAP	シーケンス間のギャップ	×	○
AVERAGE_SEQUENCE_FETCH_PAGES	順次にアクセスできる表ページの平均数。これは、索引を使用して表の行をフェッチするとき、プリフェッチャーが順次として検出できる表ページの数です。	×	○ (注 4)
AVERAGE_RANDOM_FETCH_PAGES	索引を使用して表の行をフェッチするときの、順次ページ・アクセスの間のランダム表ページの平均数	×	○ (注 4)
AVERAGE_SEQUENCE_FETCH_GAP	索引を使用して表の行をフェッチするときの、シーケンス間のギャップ	×	○ (注 4)
NUMRIDS	索引内のレコード ID (RID) の数 (タイプ 2 索引の削除対象の RID を含む)。	×	○

表 103. 索引統計 (SYSCAT.INDEXES と SYSSTAT.INDEXES) (続き)

統計	説明	RUNSTATS オプション	
		表	索引
NUMRIDS_DELETED	索引内の削除対象としてマークされている RID の合計数 (すべてのレコード ID が削除対象としてマークされている、リーフ・ページの RID は除く)	×	○
NUM_EMPTY_LEAFS	すべてのレコード ID が削除対象としてマークされている、リーフ・ページの合計数	×	○
注: 1. 詳細索引統計は、RUNSTATS コマンドに DETAILED 文節を指定します。 2. 表のサイズが相当大きいものでない限り、DETAILED 文節を指定しても CLUSTERFACTOR および PAGE_FETCH_PAIRS は収集されません。表のページ数が約 25 より大きいと、CLUSTERFACTOR または PAGE_FETCH_PAIRS 統計が収集されます。この場合、CLUSTERRATIO は -1 です (収集されません)。表が比較的小さいと、RUNSTATS は CLUSTERRATIO だけを記入し、CLUSTERFACTOR および PAGE_FETCH_PAIRS は記入しません。DETAILED 文節を指定しないと、CLUSTERRATIO 統計だけが収集されます。 3. この統計は、その表に属する索引を含むページがファイルに対してどのくらいの比率 (パーセンテージ) を占めるかを測定します。表に定義された索引が 1 つしかない表の場合には、通常、DENSITY は 100 になります。DENSITY は、索引ページがプリフェッチされたときに、他の索引から不適切なページが平均してどのくらい読み取られたのかについて、オプティマイザーが見積もるのに使用されます。 4. これらの統計は、この表が DMS 表スペースにある場合は計算できません。			

表 104. 列分布統計 (SYSCAT.COLDIST と SYSSTAT.COLDIST)

統計	説明	RUNSTATS オプション	
		表	索引
DISTCOUNT	TYPE が Q の場合は、COLVALUE 統計以下の個別値の数	分布 (注 2)	×
TYPE	行の統計が頻出値統計または変位値統計かの標識	分布	×
SEQNO	表の行を固有に識別するのに役立つ順序番号の頻度のランク	分布	×
COLVALUE	頻出値統計または変位値統計を収集する際のデータ値	分布	×

表 104. 列分布統計 (SYSCAT.COLDIST と SYSSTAT.COLDIST) (続き)

統計	説明	RUNSTATS オプション	
		表	索引
VALCOUNT	列内でデータ値が発生する頻度、または変位数の場合は、データ値 (COLVALUE) 以下の数値	分布	×
注: 1. 列分布統計は、RUNSTATS コマンドに WITH DISTRIBUTION 文節を指定します。列の値が十分に不均一でないかぎり、分布統計は収集されません。 2. DISTCOUNT は、索引の最初のキー列である列でのみ収集されます。			

関連概念:

- 管理ガイド: パフォーマンス の『カタログ統計』
- 管理ガイド: パフォーマンス の『収集される統計情報』
- 管理ガイド: パフォーマンス の『ユーザー定義関数の統計』
- 管理ガイド: パフォーマンス の『実動データベースのモデル化の統計』

データベースの作成

前提条件:

作成前に、データベースの内容、レイアウト、潜在的な増大度、および使用方法の設計に十分な時間をかけてください。

手順:

データベースの作成時には、以下のタスクがそれぞれ実行されます。

- データベースで必要になるすべてのシステム・カタログ表を設定する。
- データベースのリカバリー・ログを割り当てる。
- データベースの構成ファイルを作成し、デフォルト値を設定する。
- データベースのユーティリティをデータベースにバインドする。

システム・カタログ・ビュー上の CREATETAB、BINDADD、CONNECT、IMPLICIT_SCHEMA、および SELECT の各データベース特権は、PUBLIC に付与されます。

コントロール・センターを使用してデータベースを作成するには、以下のようになります。

1. オブジェクト・ツリーを順に展開し、「データベース (Databases)」フォルダーを表示します。
2. 「データベース (Databases)」フォルダーを右マウス・ボタン・クリックし、ポップアップ・メニューから「作成 (Create)」→「ウィザードを使用したデータベース (Database Using Wizard)」を選択します。
3. ステップにしたがってこのタスクを完了します。

以下のコマンド行プロセッサ・コマンドでは、デフォルトの位置に person1 と呼ばれるデータベースを、「Personnel DB for BSchiefer Co」という関連するコメントを付けて作成します。

```
CREATE DATABASE person1
WITH "Personnel DB for BSchiefer Co"
```

データベースの作成時、構成パラメータすべてのデフォルト値を受け入れないで、構成アドバイザーを使用し、データベースの構成を援助するように要求することもできます。これは、**CREATE DATABASE** コマンドで **AUTOCONFIGURE** オプションを使用することによって実行できます。

```
CREATE DATABASE <database name>
AUTOCONFIGURE
```

AUTOCONFIGURE 文節にはいくつかのオプションがあります。パーティション環境でデータベースを作成する場合、**AUTOCONFIGURE** 文節は使用できません。

データベースが作成されると同時に、詳細デッドロック・イベント・モニターも作成されます。他のモニターと同様に、このイベント・モニターにも関連したオーバーヘッドがあります。詳細デッドロック・イベント・モニターを必要としない場合は、以下のコマンドを使用してイベント・モニターをドロップできます。

```
DROP EVENT MONITOR db2detaildeadlock
```

このイベント・モニターが消費するディスク・スペースの量を制限するために、出力ファイルの最大数に達すると、イベント・モニターが非アクティブになり、メッセージが管理通知ログに書き込まれます。必要のない出力ファイルを除去すると、イベント・モニターは次のデータベースの活動化時にアクティブになります。

データベースを別のインスタンス (通常はリモートの) データベース・マネージャ・インスタンスに作成することができます。このタイプの環境では、デフォルト・インスタンス以外のインスタンス (リモート・インスタンスを含む) に対してインスタンス・レベルの管理を実行することができます。

関連概念:

- 管理ガイド: プランニング の『データベースに記録される事柄』

- 管理ガイド: インプリメンテーション の『データベース・マネージャーの複数インスタンス』
- 管理ガイド: インプリメンテーション の『データベースの特権』
- 管理ガイド: プランニング の『追加のデータベース設計に関する考慮事項』

関連資料:

- コマンド・リファレンス の『CREATE DATABASE コマンド』

ユーザー定義の一時表の作成

ユーザー定義の一時表は、データベース中のデータを処理するために書き込みを行っているアプリケーションに必要です。データの操作の結果は、一時的に表に保管する必要があります。

前提条件:

ユーザー一時表スペースは、ユーザー定義の一時表が作成される前に、存在しなくてはなりません。

制限:

この表の記述は、システム・カタログには現れません。したがって、この表を他のアプリケーションのために保持したり、他のアプリケーションと共用したりすることはできません。

この表を使用するアプリケーションが終了したりデータベースから切断されたりすると、表の中のデータはすべて削除され、表は暗黙的にドロップされます。

ユーザー定義 TEMPORARY 表は、以下のものをサポートしません。

- LOB タイプの列 (または LOB に基づく特殊タイプ列)
- ユーザー定義タイプの列
- LONG VARCHAR 列
- DATALINK 列

手順:

TEMPORARY 表を定義するには、DECLARE GLOBAL TEMPORARY TABLE ステートメントを使用します。このステートメントは 1 つのアプリケーションの中から使用されます。ユーザー定義 TEMPORARY 表が保持されるのは、アプリケーションがデータベースから切断されるまでの間だけです。

TEMPORARY 表を定義する方法の例を以下に示します。

```
DECLARE GLOBAL TEMPORARY TABLE gbl_temp
  LIKE empltab1
  ON COMMIT DELETE ROWS
  NOT LOGGED
  IN usr_tbsp
```

このステートメントにより、gbl_temp というユーザー TEMPORARY 表が作成されます。このユーザー TEMPORARY 表の列は、名前と記述が empltab1 の列と完全に一致するように定義されています。暗黙定義には、列名、データ・タイプ、NULL 可能特性、および列のデフォルト値の属性だけが含まれます。他のすべての列属性 (ユニーク制約、外部キー制約、トリガー、索引を含む) は、定義されていません。COMMIT 操作を実行すると、表で WITH HOLD カーソルがオープンしていなければ、表の中のデータはすべて削除されます。ユーザー TEMPORARY 表に対する変更内容はログに記録されません。ユーザー TEMPORARY 表は、指定された USER TEMPORARY 表スペースに置かれます。この表スペースがないと、この表の宣言は失敗します。

ROLLBACK または ROLLBACK TO SAVEPOINT がこの表の作成時に指定されると、表中のすべての行を削除する (DELETE ROWS、これがデフォルトです) か、表の行を保存する (PRESERVE ROWS) ことを指定できます。

関連タスク:

- 管理ガイド: インプリメンテーション の『USER TEMPORARY 表スペースの作成』

関連資料:

- *SQL* リファレンス 第 2 巻 の『ROLLBACK ステートメント』
- *SQL* リファレンス 第 2 巻 の『SAVEPOINT ステートメント』
- 1610 ページの『DECLARE GLOBAL TEMPORARY TABLE』

ツール・カタログ・データベースおよび DAS スケジューラーのセットアップと構成

ツール・カタログ・データベースには、タスク・センターおよびコントロール・センターによって作成されたタスク情報が含まれます。これらのタスクは、DB2 Administration Server のスケジューラーによって実行されます。

前提条件:

DB2 Administration Server がインストール済みでなければなりません。

手順:

ツール・カタログ・データベースおよび DAS スケジューラーをセットアップして構成します。

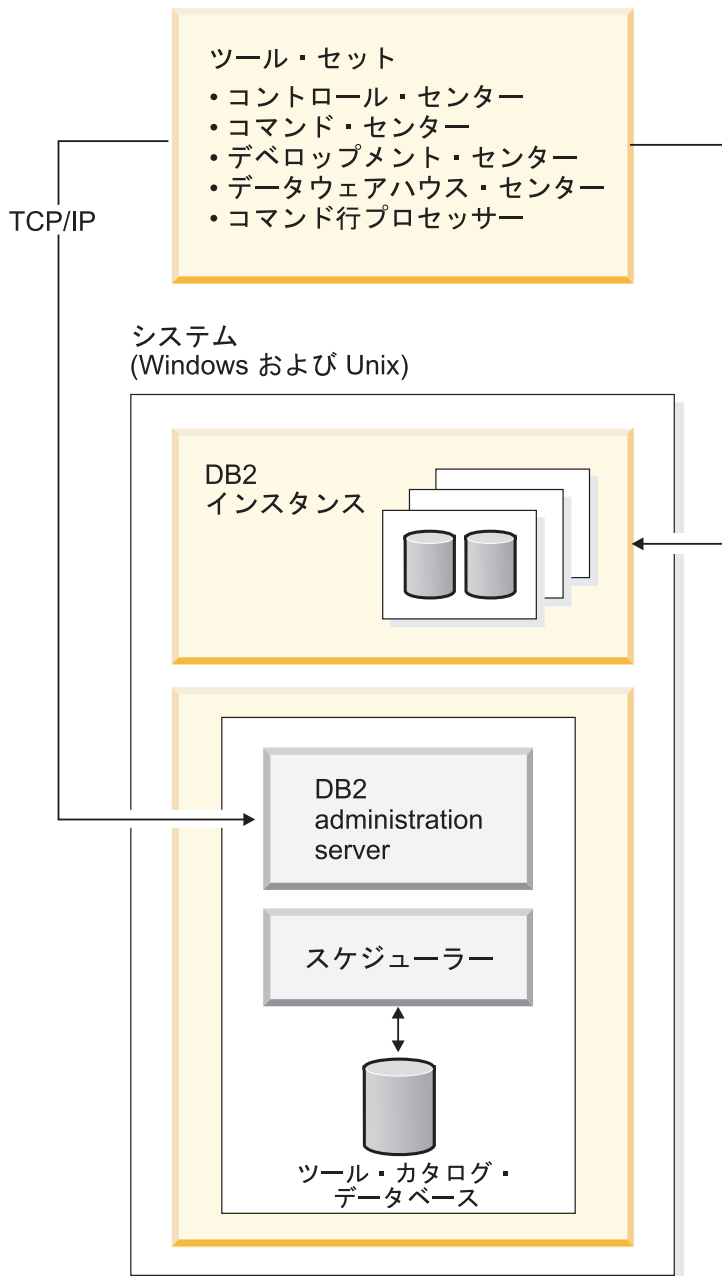


図2. DAS と DB2 の他の部分との関連

DAS スケジューラーは、ツール・カタログ情報にアクセスするために Java 仮想計算機 (JVM) を必要とします。JVM 情報を指定するには、DAS の DB2 Administration Server 構成パラメーター `jdk_path` を使用します。

コントロール・センターおよびタスク・センターは、ツール・カタログ・データベースにクライアントから直接アクセスします。したがって、コントロール・センターで使用する前に、ツール・カタログ・データベースをクライアントでカタログしておく必要があります。コントロール・センターは、ツール・カタログ・データベースに関する情報を自動的に検索し、ローカル・ノード・ディレクトリーおよびデータベース・ディレクトリー内に必要なディレクトリー項目を作成します。この自動カタログ用にサポートされる通信プロトコルは、TCP/IP のみです。

DAS パラメーターの 1 つは *exec_exp_task* です。このパラメーターは、すでにスケジュール済みの、まだ実行されていないタスクをスケジューラーが実行するかどうかを指定します。スケジューラーの開始時には、有効期限切れタスクを検出するだけです。

たとえば、毎週土曜日に実行するようにスケジュールされたジョブがあり、スケジューラーが金曜日にオフにされ、月曜日に再始動された場合、土曜日にスケジュールされていたジョブは、過去にスケジュールされていたジョブになります。*exec_exp_task* が「Yes」に設定されている場合、土曜日のジョブは、スケジューラーが再始動した時点で実行されます。

スケジューラーが必要とするその他の DAS 構成パラメーターは、ツール・カタログ・データベースの識別と、通知に使われる Simple Mail Transfer Protocol (SMTP) サーバーの識別で構成されます。

以下の例は、これらのパラメーターの使用法を示しています。

• Windows サーバー・セットアップの例

1. ツール・カタログ・データベースは任意の名前にすることができます。この例では、ツール・カタログ・データベースの名前は "CCMD" で、サーバー・マシン Host1 (tcp/ip hostname Host1) 上の DB2 インスタンスの下に作成されます。特定のデータベース内のツール・カタログを固有に識別するために、スキーマ名を使用します。この例では、スキーマ名を「CCADMIN」と想定します。
2. インスタンス DB2 は、ポート番号 50000 を使用して、以下のように TCP/IP 通信にセットアップされます。

```
db2set -i DB2 DB2COMM=TCPIP
db2 update dbm cfg using svccname db2cDB2
db2stop
db2start
```

3. db2cDB2 サービス名は、`%SystemRoot%\system32\drivers\etc\services` に定義されています。つまり、`services` の中に以下の 1 行があります。

```
db2cDB2 50000/tcp #connection port for the DB2 instance DB2
```
4. ツール・カタログは、`CREATE TOOLS CATALOG` コマンドを使用して作成されます。これにより、指定したデータベース内のカタログ名と対応するスキーマ名を持つ、ツール・カタログ表およびビューが作成されます。DB2 Administration server 構成パラメーターは自動的に更新され、スケジューラーが使用可能になり、開始されます。

5. e-mail 通知に使われる SMTP サーバーがマシン Host2 (tcp/ip hostname Host2) 上にあるとします。以下を使用して、この情報が DB2 Administration Server に対して指定されます。

```
db2 update admin cfg using smtp_server Host2
```

これは、インストール中に行われる場合があります。後で行う場合は、上記のような DB2 UDB バージョン 8 CLP コマンドを使用して、手動で DAS に対して指定する必要があります。

6. Windows での Java Development Kit (JDK) は、%DB2PATH%\java\jdk の下にインストールされます。これは、DAS に対してすでに指定済みでなければなりません。必要であれば、以下のようにして、このパラメーターを検査および設定できます。

```
db2 update admin cfg using jdk_path c:%SQLLIB%\java\jdk
```

ここでは、DB2 が C:%SQLLIB の下にインストールされていることを想定しています。

注: DAS を db2admin create で作成する場合、/USER および /PASSWORD オプションを必ず使用してください。USER アカウントは、スケジューラー・プロセスによって使用されます。これがないと、スケジューラーは正常に開始されません。USER アカウントには、ツール・カタログ・インスタンスで SYSADM 権限が必要です。

DAS が db2admin create で作成され、/USER および /PASSWORD オプションが同時に指定されていない場合は、後で USER 情報を更新できます。この更新は、以下のコマンドを実行することで、DAS 上で行えます。

```
db2admin stop
db2admin setid <user account ID> <password>
db2admin start
```

- Windows クライアント・セットアップの例

1. コントロール・センターがクライアント・マシン C1 (tcp/ip ホスト名 C1) で実行されているとします。
2. DAS は、ローカル・ノード・ディレクトリー内の Administration Server ノードとしてカタログされます。そうするには、構成アシスタントまたはコントロール・センターを使用するか、以下のコマンドを使用します。

```
db2 catalog admin tcpip node Host1 remote Host1 system Host1 ostype NT
```

3. タスク・センターが開始済みで、システム Host1 が選択されている場合、タスク・センターはローカル・ディレクトリー内にツール・カタログ・データベースを見つけようとしています。(タスク・センターの代わりにコントロール・センターを使用することもできます。) 見つからない場合、以下を使用してノードとデータベースをカタログしようとしています。

```

|         db2 catalog tcpip node <unique-node name>
|           remote Host1 server 50000
|           remote_instance DB2 system Host1 ostype NT
|         db2 catalog db CCMD as <unique-db alias> at node <unique-node name>

```

自動カタログが失敗した場合、構成アシスタントまたはコントロール・センターを使ってデータベースをカタログすることができます。その後、データベースはタスク・センターに認識され、使用できるようになります。

- AIX サーバー・セットアップの例

1. ツール・カタログ・データベースは任意の名前にすることができます。この例では、ツール・カタログ・データベースの名前は "CCMD" で、サーバー・マシン Host1 (tcp/ip hostname Host1) 上の db2inst1 インスタンスの下に作成されます。特定のデータベース内のツール・カタログを固有に識別するために、スキーマ名を使用します。この例では、スキーマ名を「CCADMIN」と想定します。
2. インスタンス db2inst1 は、ポート番号 50000 を使用して、以下のように TCP/IP 通信用にセットアップされます。

```

|         db2set -i DB2 DB2COMM=TCPIP
|         db2 update dbm cfg using svcename xdb2inst
|         db2stop
|         db2start

```

3. xdb2inst サービス名は、 /etc/services に定義されています。つまり、services の中に以下の 1 行があります。

```

|         xdb2inst1    50000/tcp    #connection port for the DB2 instance db2inst1

```

4. ツール・カタログは、CREATE TOOLS CATALOG コマンドを使用して作成されます。これにより、指定したデータベース内のカタログ名と対応するスキーマ名を持つ、ツール・カタログ表およびビューが作成されます。DB2 Administration server 構成パラメーターは自動的に更新され、スケジューラーが使用可能になり、開始されます。
5. e-mail 通知に使われる SMTP サーバーがマシン Host2 (tcp/ip hostname Host2) 上にあるとします。以下を使用して、この情報が DB2 Administration Server に対して指定されます。

```

|         db2 update admin cfg using smtp_server Host2

```

これは、インストール中に行われる場合があります。後で行う場合は、上記のような DB2 UDB バージョン 8 CLP コマンドを使用して、手動で DAS に対して指定する必要があります。

6. AIX 上の Java Development Kit (JDK) は、/usr/java131 の下にインストールされます。これは、DAS に対してすでに指定済みでなければなりません。必要であれば、以下のようにして、このパラメーターを検査および設定できます。

```

|         db2 update admin cfg using jdk_path /usr/java131

```

- AIX クライアント・セットアップの例

1. コントロール・センターがクライアント・マシン C1 (tcp/ip ホスト名 C1) で実行されているとします。
2. DAS は、ローカル・ノード・ディレクトリー内の Administration Server ノードとしてカタログされます。そのためには、構成アシスタントまたはコントロール・センターを使用するか、以下のコマンドを使用します。

```
db2 catalog admin tcpip node Host1 remote Host1 system Host1
ostype AIX
```

3. タスク・センターが開始済みで、システム Host1 が選択されている場合、タスク・センターはローカル・ディレクトリー内にツール・カタログ・データベースを見つけようとしています。(タスク・センターの代わりにコントロール・センターを使用することもできます。) 見つからない場合、以下を使用してノードとデータベースをカタログしようとしています。

```
db2 catalog tcpip node <unique-node name>
remote Host1 server 50000
remote_instance DB2 system Host1 ostype AIX
db2 catalog db CCMD as <unique-db alias> at node <unique-node name>
```

自動カタログが失敗した場合、構成アシスタントまたはコントロール・センターを使ってデータベースをカタログすることができます。その後、データベースはタスク・センターに認識され、使用できるようになります。

関連資料:

- 管理ガイド: パフォーマンス の『「TCP/IP サービス名」構成パラメーター - svcname』
- 管理ガイド: パフォーマンス の『「スケジューラー・モード」構成パラメーター - sched_enable』
- 管理ガイド: パフォーマンス の『「ツール・カタログ・データベース・インスタンス」構成パラメーター - toolscat_inst』
- 管理ガイド: パフォーマンス の『「ツール・カタログ・データベース」構成パラメーター - toolscat_db』
- 管理ガイド: パフォーマンス の『「ツール・カタログ・データベース・スキーマ」構成パラメーター - toolscat_schema』
- 管理ガイド: パフォーマンス の『「SMTP サーバー」構成パラメーター - smtp_server』
- 管理ガイド: パフォーマンス の『「Java Development Kit インストール・パス DAS」構成パラメーター - jdk_path』
- 管理ガイド: パフォーマンス の『「有効期限切れタスクの実行」構成パラメーター - exec_exp_task』

ツリーに複数のオブジェクトを入れるためのフォルダーの追加

手順:

この例では、プラグイン・オブジェクトを「コントロール・センター (Control Center)」ツリーの「データベース (Database)」の下に示すため、CCObject ではなく CCTreeObject をインプリメントします。まず、このオブジェクトのために CCTreeObject インプリメンテーションを作成する必要があります。ツリーに配置するオブジェクトが複数の場合は、「データベース (Database)」の下にすべてを直接配置するのではなく、フォルダーを作成することが定石です。以下は、初期バージョンのフォルダーです。

```
public class Example3Folder implements CCTreeObject {
    public boolean isEditable () { return false; }
    public boolean isConfigurable () { return false; }
    public CCTableObject getChildren () { return null; }
    public CCColumn[] getColumns () { return null; }
    public boolean isLeaf () { return false; }
    public CCMenuItem[] getMenuActions () { return null; }

    public String getName () { return "Example3 Folder"; }

    public void getData (Object[] data) {
        data[0] = this;
    }

    public int getType () { return CCTypeFactory.getTypeNumber
(this.getClass().getName()); }

    public Icon getIcon (int iconState) {
        switch (iconState) {
            case CLOSED_FOLDER:
                return CommonImageRepository.getScaledIcon(CommonImageRepository.NV_CLOSED_
FOLDER);
            case OPEN_FOLDER:
                return CommonImageRepository.getScaledIcon(CommonImageRepository.NV_OPEN_
FOLDER);
            default:
                return CommonImageRepository.getScaledIcon(CommonImageRepository.NV_CLOSED_
FOLDER);
        }
    }
}
```

getType がクラス CCTypeFactory を利用するようになったことに注意してください。CCTypeFactory の目的は、プラグインが固有のタイプを持つものとしてコントロール・センターに見なされるように、2つのオブジェクトが同じタイプ番号を使用することを防ぐことにあります。新しいフォルダーは、いずれかの組み込み CC オブジェクト・タイプではなく、新しいタイプですので、作成できる他の新しいタイプの番号と競合せず、なおかつ組み込みタイプの番号と競合しない、新しいタイプの番号が必要です。

getIcon メソッドは、オープンしたフォルダーかクローズしたフォルダーかを知らせる iconState のために、パラメーターを取ります。そして、上記のように、アイコンを状態に対応させることができます。

データベースが選択される場合で、ツリー内だけに存在するのではない場合に、フォルダーを詳細ビューで表示するためには、getData は、値がプラグイン・オブジェクト自体である 1 つの列を戻す必要があります。getData メソッドは、この参照を、データ配列の最初のエレメントに割り当てます。これにより、アイコンと名前の両方を、詳細ビューの同じ列に表示できます。コントロール・センターは、ユーザーが CCTableObject サブクラスを戻すことが分かると、Example3Folder に対して getIcon および getName を呼び出せると判別します。

次のステップは、以下のようにして、CCTreeObject をインプリメントする CCDatabase クラスを作成し、その getChildren メソッドから Example3Folder のインスタンスを含む CCTableObject 配列を戻すことです。

```
import java.util.*;

class CCDatabase implements CCTreeObject {
    private Vector childVector;

    public CCDatabase() {
        childVector = new Vector();
        childVector.addElement(new Example3Folder());
    }

    public CCTableObject[] getChildren() {
        CCTableObject[] children = new CCTableObject[childVector.size()];
        childVector.copyInto(children);
        return children;
    }

    public String getName () { return null; }
    public boolean isEditable () { return false; }
    public boolean isConfigurable () { return false; }
    public void getData (Object[] data) { }
    public CCColumn[] getColumns () { return null; }
    public boolean isLeaf () { return false; }
    public int getType () { return UDB_DATABASE; }
    public Icon getIcon (int iconState) { return null; }
    public CCMenuAction[] getMenuActions () { return null; }
}
```

関連概念:

- 311 ページの『サンプル・プラグインのコンパイルおよび実行』

関連タスク:

- 335 ページの『フォルダーの下へのサンプル・オブジェクトの追加』
- 管理ガイド: インプリメンテーション の『プラグイン・ツリー・オブジェクトの属性の設定』

- 管理ガイド: インプリメンテーション の『Create アクションの追加』
- 管理ガイド: インプリメンテーション の『複数選択サポートを持った Remove アクションの追加』
- 338 ページの『Alter アクションの追加』

フォルダーの下へのサンプル・オブジェクトの追加

手順:

最初のステップは、以下のようにして、子オブジェクト用の CCOBJECT インプリメンテーションを作成することです。

```

|         class Example3Child implements CCTableObject {
|             public String getName () { return null; }
|             public boolean isEditable () { return false; }
|             public boolean isConfigurable () { return false; }
|             public void getData (Object[] data) { }
|             public CCColumn[] getColumns () { return null; }
|             public Icon getIcon (int iconState) { return null; }
|             public CCMenuAction[] getMenuActions () { return null; }
|             public int getType () { return CCFactory.getTypeNumber
| (this.getClass().getName()); }
|         }

```

次に、これらの Exercise3Child オブジェクトの Vector を保持するよう、以下のようにして、Example3Folder を変更します。

```

public class Example3Folder implements CCOBJECT {
    private Vector childVector;

    public Example3Folder() {
        childVector = new Vector();
    }

    ...
    public CCTableObject[] getChildren () {
        CCTableObject[] children = new CCTableObject[childVector.size()];
        childVector.copyInto(children);
        return children;
    }
}

```

関連概念:

- 311 ページの『サンプル・プラグインのコンパイルおよび実行』

関連タスク:

- 332 ページの『ツリーに複数のオブジェクトを入れるためのフォルダーの追加』
- 管理ガイド: インプリメンテーション の『プラグイン・ツリー・オブジェクトの属性の設定』

- 管理ガイド: インプリメンテーションの『Create アクションの追加』
- 管理ガイド: インプリメンテーションの『複数選択サポートを持った Remove アクションの追加』
- 338 ページの『Alter アクションの追加』

新しい表での ID 列の定義

ID 列を使用すると、表に追加される個々の行に対し、固有の数値を DB2 が自動的に生成します。表に追加する個々の行を固有に識別する必要があることが分かっている場合、その表を作成する際に、ID 列を追加できます。表に追加する個々の行の固有な数値を保証するには、ID 列に固有索引を作成するか、または主キーとして宣言する必要があります。

制限:

いったん作成した後で、表の記述を変更して、ID 列を組み込むことはできません。

ID 列は単一パーティション・データベースでのみサポートされています。

明示的な ID 列値が指定されている表に行を挿入する場合、内部的に生成される次の値は更新されないため、表の中の既存の値との間で矛盾が発生することがあります。重複値は、ID 列の固有性が、ID 列で定義された主キーまたは固有索引によって強制されている場合に、エラー・メッセージを生成します。

手順:

CREATE TABLE ステートメントで AS IDENTITY 文節を使用すると、ID 列を指定できます。

以下に、CREATE TABLE ステートメントを使って ID 列を定義する方法の例を示します。

```
CREATE TABLE table (col1 INT,
                    col2 DOUBLE,
                    col3 INT NOT NULL GENERATED ALWAYS AS IDENTITY
                    (START WITH 100, INCREMENT BY 5))
```

この例では、3 番目の列が ID 列です。この列で使用される値を指定して、追加される個々の行を固有に識別することもできます。この例の場合、最初に入力される行については、値 "100" が ID 列に入れられます。この表に行が追加されるごとに、値は 5 ずつ増えます。

ID 列を使用する他の例としては、注文番号、従業員番号、在庫番号、事例番号などがあります。ALWAYS または BY DEFAULT を指定して、DB2 によって ID 列の値が生成されるようにすることができます。

GENERATED ALWAYS として定義された ID 列は、常に DB2 が生成する値に指定されます。アプリケーションが明示的に値を指定することはできません。ID 列を GENERATED BY DEFAULT として定義すると、アプリケーションが明示的に ID 列の値を指定できます。アプリケーションが値を指定しないと、DB2 が値を生成します。アプリケーションが値を制御するので、値が固有であることを DB2 が保証することはできません。GENERATED BY DEFAULT 文節は、既存の表の内容をコピーする目的でデータ伝搬を行う場合や、表のアンロードや再ロードを行う場合に使用します。

関連概念:

- 管理ガイド: インプリメンテーション の『IDENTITY 列とシーケンスの比較』

関連資料:

- 1484 ページの『CREATE TABLE』

Enterprise Server Edition (ESE) システムの DAS 構成

DAS は、ツールの代わりに特定のタスクを実行する管理コントロール・ポイントです。物理的なマシン 1 台につき設けることのできる DAS は多くて 1 つです。いくつかのマシンで構成される ESE インスタンスの場合は、コントロール・センターが ESE インスタンスを実行できるよう、すべてのマシンで DAS を実行している必要があります。この DAS (db2as) は、ターゲット DB2[®] インスタンス (db2inst) の親としてコントロール・センターのナビゲーター・ツリーに存在するシステムによって表されています。

たとえば、db2inst は 2 つの物理マシンまたはホストにわたって分散している 3 つのノードから構成されています。hostA および hostB ので **db2as** を実行することによって、最小要件を満たすことができます。

注:

1. hostA に存在するパーティションの数は、hostA で実行できる DAS の数とは何の関係もありません。ホストに複数の論理ノード (MLN) 構成が存在するとしても、hostA で実行できる DAS のコピーは 1 つだけです。
2. **dascrt** コマンドを使って個々に作成する必要がある、各マシン、または物理ノードに必要な DAS は 1 つです。各マシンまたは物理ノード上の DAS は、タスク・センターおよびコントロール・センターが正しく作動するように、稼働していなければなりません。ID db2as は、hostA および hostB に存在する必要があります。db2as ID のホーム・ディレクトリーは、2 つのシステム間でクロス・マウントしてはいけません。その代わりに、DAS を hostA と hostB に作成するために、異なるユーザー ID を、使用することができます。

DB2 UDB for Windows[®] Enterprise - Server Edition では、構成アシスタントまたはコントロール・センターを使って DB2 サーバーへの接続構成を自動化する場合、DAS と同一マシン上にあるデータベース・パーティション・サーバーはコーディネーター・

ノードになります。つまり、クライアントからデータベースへの物理接続はすべて、他のデータベース・パーティション・サーバーに経路指定される前に、コーディネーター・ノードに向けられます。

DB2 UDB for Windows Enterprise - Server Edition では、他のマシン上で DB2 Administration Server を追加すると、構成アシスタントまたはコントロール・センターで DB2 ディスカバリーを使って他のシステムをコーディネーター・ノードとして構成することができます。

関連タスク:

- 管理ガイド: インプリメンテーションの『DB2 Administration Server の作成』

関連資料:

- コマンド・リファレンスの『db2admin - DB2 Administration Server コマンド』

Alter アクションの追加

手順:

プラグインに関連してコントロール・センターが listen する最後のタイプのイベントは、OBJECT_ALTERED イベントです。この機能をこの例で実演できるように、前の例で "State" 列を作成しました。Alter アクションが呼び出されると、状態値を増分することになります。

最初のステップは、状態を変更するメソッドを作成することですが、今回は、フォルダーではなく Example3Child 上で行います。この場合、最初の引き数と 3 番目の引き数はどちらも Example3Child です。必ず Observable を拡張してください。

```
class Example3Child extends Observable implements CCTableObject {
    ...
    public void setState(String state) {
        this.state = state;
        setChanged();
        notifyObservers(new CCOBJECTCollectionEvent(this,
            CCOBJECTCollectionEvent.OBJECT_ALTERED, this));
    }
    ...
}
```

次に、Alter のメニュー・アクションを作成し、それを Example3Child の CCMenuAction 配列へ追加します。AlterAction クラスはまた、ユーザーがコントロール・センターで Example3Child オブジェクトをダブルクリックするときに呼び出されるデフォルトのアクションとして、Alter を定義するために、CCDefaultMenuAction インターフェースをインプリメントします。

```
class AlterAction implements CCMenuAction, CCDefaultMenuAction {
    private int stateNumber = 1;
    public String getMenuText () { return "Alter"; }
```

```

    public void actionPerformed (ActionEvent e) {
        ((Example3Child)((Vector)e.getSource()).elementAt(0)).setState("State "
            + ++stateNumber);
    }
}

class Example3Child implements CCTableObject {
    ...
    public Example3Child(Example3Folder folder, String name, String state) {
        ...
        menuActions = new CCMenuAction[] { new AlterAction(),
            new RemoveAction(folder) };
    }
    ...
}

```

関連概念:

- 311 ページの『サンプル・プラグインのコンパイルおよび実行』

関連タスク:

- 332 ページの『ツリーに複数のオブジェクトを入れるためのフォルダーの追加』
- 335 ページの『フォルダーの下へのサンプル・オブジェクトの追加』
- 管理ガイド: インプリメンテーション の『プラグイン・ツリー・オブジェクトの属性の設定』
- 管理ガイド: インプリメンテーション の『Create アクションの追加』
- 管理ガイド: インプリメンテーション の『複数選択サポートを持った Remove アクションの追加』

コントロール・センター用プラグインの開発者向けのガイドライン

db2plug.zip ファイルには複数のプラグインを入れることができるので、プラグインの開発者は、コントロール・センター用のプラグインを作成する場合は、以下のガイドラインに従う必要があります。

- Java™ パッケージを使用して、プラグイン・クラスが固有の名前を持つようにする。Java パッケージの命名規則にしたがってください。パッケージ名の接頭部には、インターネット・ドメインを逆にした名前 (たとえば、com.companyname) を付けます。パッケージ名の全部または少なくとも固有の接頭部は、小文字でなければなりません。
- db2plug.zip は、sqllib ディレクトリーの下 の tools ディレクトリーにインストールする必要があります。V8 以前では、db2plug.zip は、sqllib ディレクトリーの下 の cc ディレクトリーにインストールする必要がありました。
- コントロール・センター用のプラグインを作成する場合は、db2plug.zip ファイルがすでに存在する場合は、プラグイン・クラスを既存の db2plug.zip に追加する必要があります。

あります。既存の db2plug.zip ファイルを、独自の db2plug.zip ファイルで上書きしてはなりません。既存の db2plug.zip にプラグインを追加するには、以下の zip コマンドを使用してください。

```
zip -r0 db2plug.zip com%companyname%myplugin%*.class
```

プラグイン・パッケージ名は com.companyname.myplugin です。

- db2plug.zip 内のすべてのクラスは、コントロール・センターの開始時にロードされます。db2plug.zip ファイルには、com.ibm.db2.tools.cc.navigator パッケージ内のクラスを拡張またはインプリメントする、すべての CCEExtension クラス・ファイルおよびクラスが含まれていなければなりません。これらのクラスによって直接使用されないクラスは、db2plug.zip に含まれている必要はありません。これらのファイルは、コントロール・センターの開始時のパフォーマンス上の影響を最小限に抑えるために、別の jar ファイルに保管することができます。これは、余分のクラスが多数ある場合に役立ちます。jar ファイルは、sql1ib ディレクトリーの下 tools ディレクトリーに置く必要があります。db2cc コマンドを使用してコントロール・センターを開始すると、jar ファイルは自動的に classpath に組み込まれます。
- CCOBJECT をインプリメントするプラグイン・クラスは、コントロール・センターから Class.newInstance() への呼び出しが可能な、引き数なしのデフォルト・コンストラクターを提供する必要があります。
- 内部クラスの使用を可能な限り避けます。一般的に、コントロール・センターで新規プラグイン・オブジェクトを作成するために、CCTreeObject をインプリメントするプラグイン・クラスを、内部クラスとして宣言することはできません。これは、コントロール・センターがこれらのクラスをインスタンス化するのを防ぎます。
- db2cc -tf filename を使用して、プラグインが正常にロードされているかをテストします。これにより、コントロール・センターのトレース情報が指定したファイル名で書き込まれます。絶対パス名を指定しないと、トレース・ファイルは sql1ib のツール・ディレクトリーに書き込まれます。プラグイン関連のトレース・ステートメントには、「Plugin」という語が含まれます。行に「PluginLoader」というテキストが含まれているかどうかを探することで、クラスがロードされているかを確認できます。

関連概念:

- 311 ページの『サンプル・プラグインのコンパイルおよび実行』
- 管理ガイド: インプリメンテーション の『コントロール・センターの拡張機能としてのプラグインの作成』

関連資料:

- コマンド・リファレンス の『db2cc - コントロール・センターの開始コマンド』

「アプリケーション・ヒープ・サイズ」構成パラメーター - applheapsz

構成タイプ

データベース

パラメーター・タイプ

構成可能

デフォルト[範囲]

32 ビット ローカル・クライアントおよびリモート・クライアントを指定したデータベース・サーバー
256 [16 - 60 000]

64 ビット ローカル・クライアントおよびリモート・クライアントを指定したデータベース・サーバー
256 [16 - 60 000]

32 ビット ローカル・クライアントおよびリモート・クライアントを指定したパーティション・データベース・サーバー
64 [16 - 60 000]

64 ビット ローカル・クライアントおよびリモート・クライアントを指定したパーティション・データベース・サーバー
128 [16 - 60 000]

計測単位

ページ (4 KB)

割り振られるタイミング

エージェントが、アプリケーションの作業を行うように初期化されるとき

解放されるタイミング

エージェントがアプリケーションの作業を完了するとき

このパラメーターは、特定エージェントまたはサブエージェントの代わりに、データベース・マネージャーが使用可能な専用メモリー・ページの数を定義します。

エージェントまたはサブエージェントがアプリケーション用に初期化されると、ヒープが割り振られます。割り振られる量は、エージェントまたはサブエージェントに指定された要求を処理するのに最低限必要な量です。 エージェントまたはサブエージェントが、大きな SQL ステートメントを処理するためにさらにヒープ・スペースを要求した場合は、データベース・マネージャーが必要に応じて、このパラメーターで指定された最大値までメモリーを割り振ります。

注: パーティションのあるデータベース環境では、アプリケーション・コントロール・ヒープ (*app_ctl_heap_sz*) が、エージェントとサブエージェントの SQL ステートメントの実行セクションのコピーを保管するのに使用されます。 ただし、他のすべての環境でエージェントを実行するときは、SMP サブエージェントは、*applheapsz* を使用します。

推奨: アプリケーションが、アプリケーション・ヒープに十分なストレージがないことを示すエラーを受け取った場合は、このパラメーターの値を増やしてください。

アプリケーション・ヒープ (*applheapsz*) は、エージェント専用メモリーから割り振られます。

関連資料:

- 342 ページの『「アプリケーション・コントロール・ヒープ・サイズ」構成パラメーター - *app_ctl_heap_sz*』

「アプリケーション・コントロール・ヒープ・サイズ」構成パラメーター - *app_ctl_heap_sz*

構成タイプ	データベース
パラメーター・タイプ	構成可能
デフォルト[範囲]	ローカル・クライアントおよびリモート・クライアントを指定したデータベース・サーバー 128 [1-64 000]
	ローカル・クライアントを指定したデータベース・サーバー 64 [1-64 000] (非 UNIX プラットフォームの場合) 128 [1-64 000] (UNIX ベースのプラットフォームの場合)
	ローカル・クライアントおよびリモート・クライアントを指定したパーティション・データベース・サーバー 512 [1-64 000]
計測単位	ページ (4 KB)
割り振られるタイミング	アプリケーション開始時期
解放されるタイミング	アプリケーション完了時期

パーティション・データベースの場合、およびパーティション内並列処理が使用可能 (*intra_parallel=ON*) の非パーティション・データベースの場合は、このパラメーターは、アプリケーション用として割り振られる共有メモリー領域の平均サイズを指定します。パーティション内並列処理が使用不可 (*intra_parallel=OFF*) の非パーティション・データベースの場合は、これはヒープとして割り振られる最大専用メモリー・サイズです。それぞれのパーティションごとに 1 つの接続に 1 つずつアプリケーション・コントロール・ヒープがあります。

アプリケーション・コントロール・ヒープが必要なのは、主として、同一要求のために作動するエージェント間で情報を共用するためです。このヒープの使用量が最小になるのは、非パーティション・データベースで、1 に等しい並列処理の度合いで照会を実行しているときです。

このヒープは、宣言済み一時表の記述子情報を保管することにも使用されます。明示的にドロップされていないすべての宣言済み一時表の記述子情報はこのヒープのメモリーに保持され、宣言済み一時表がドロップされるまでドロップすることはできません。

推奨: 最初はデフォルト値で開始してください。複雑なアプリケーションを実行する場合、多数のデータベース・パーティションを含むシステムである場合、または宣言済み一時表を使用する場合は、値をより高く設定しなければならないことがあります。必要とされるメモリーの量は、並行してアクティブな宣言済み一時表の数とともに増加します。多くの列を持つ宣言済み一時表の表記述子サイズは、少ない列を持つ表のものよりも大きくなります。そのため、アプリケーションの宣言済み一時表が多数の列を持つと、アプリケーション・コントロール・ヒープへの要求も増えます。

関連資料:

- 管理ガイド: パフォーマンス の『「パーティション内並列処理の使用可能化」構成パラメーター - intra_parallel』
- 340 ページの『「アプリケーション・ヒープ・サイズ」構成パラメーター - applheapsz』
- 管理ガイド: パフォーマンス の『「アプリケーション・グループ・メモリー・セットの最大サイズ」構成パラメーター - appgroup_mem_sz』
- 管理ガイド: パフォーマンス の『「アプリケーション・グループ・ヒープ用メモリーのパーセント」構成パラメーター - groupheap_ratio』

システム環境変数

表 105. システム環境変数

変数名	オペレーティング・システム		値
DB2CONNECT_IN_APP_PROCESS	すべて		デフォルト = YES 値: YES または NO
この変数を NO に設定すると、DB2 Connect Enterprise Edition マシン上のローカル DB2 Connect クライアントはエージェント内で強制的に実行されます。 エージェント内で実行する利点としては、ローカル・クライアントをモニターできることと、ローカル・クライアントが SYSPLEX サポートを使用できることがあります。			
DB2DOMAINLIST	Windows NT サーバーのみ		デフォルト = null 値: コンマ (',') 区切りの Windows NT ドメイン・ネームのリスト

表 105. システム環境変数 (続き)

変数名	オペレーティング・システム	
説明		
1 つ以上の Windows NT ドメインを定義します。これらのドメインに属しているユーザーの接続要求のみが受け入れられます。		
このレジストリー変数は、DB2 Universal Database バージョン 7.1 (またはそれ以降) が稼働する DB2 サーバーおよびクライアントの純粋な Windows NT ドメイン環境下でのみ使用します。		
DB2ENVLIST	UNIX	デフォルト: null
<p>ストアド・プロシージャまたはユーザー定義関数の特定の変数名をリストします。デフォルトでは、db2start コマンドは、接頭部が DB2 または db2 になっているユーザー環境変数を除いて、すべてのユーザー環境変数をフィルターに掛けて除去します。特定の環境変数をストアド・プロシージャまたはユーザー定義関数のどちらかに渡さなければならない場合、DB2ENVLIST 環境変数にその変数名をリストできます。その場合、各変数を 1 つまたは複数のスペースで区切ります。</p>		
DB2INSTANCE	すべて	デフォルト = DB2INSTDEF (Windows 32 ビット・オペレーティング・システムの場合)
デフォルト解釈によりアクティブになるインスタンスを指定するために使用される環境変数。UNIX では、ユーザーは DB2INSTANCE に値を指定する必要があります。		
DB2INSTPROF	Windows オペレーティング・システム	デフォルト: null
Windows オペレーティング・システムにおいて、インスタンス・ディレクトリーが DB2PATH 以外の場合にある場合のその位置を示すために使用される環境変数。		
DB2LIBPATH	UNIX	デフォルト: null
<p>DB2 はその共有ライブラリー・パスを構成します。PATH をエンジンのライブラリー・パスに追加した場合は、(たとえば、AIX ではユーザー定義関数には LIBPATH 内に特定項目が必要) DB2LIBPATH を設定する必要があります。実際の DB2LIBPATH の値は、DB2 構成共有ライブラリー・パスの最後に付加されます。</p>		
DB2NODE	すべて	デフォルト: null 値: 1 ~ 999
<p>アタッチまたは接続する DB2 Enterprise Server Edition データベース・パーティション・サーバーのターゲット論理ノードを指定します。この変数を指定しない場合、ターゲット論理ノードはデフォルトとして、マシン上のポート 0 に定義された論理ノードに設定されます。</p>		
DB2_PARALLEL_IO	すべて	デフォルト: null 値: * (すべての表スペース) または定義された複数の表スペースのコンマ区切りリスト

表 105. システム環境変数 (続き)

変数名	オペレーティング・システム 値	
説明		
<p>表スペース・コンテナからの読み取り、または表スペース・コンテナへの書き込み中に、DB2 は指定する表スペースの値ごとに並列入出力を使用する場合があります。並列処理の度合いは、表スペース内のコンテナのプリフェッチ・サイズおよびエクステント・サイズによって決まります。たとえば、プリフェッチ・サイズがエクステント・サイズの 4 倍であるとき、4 つのエクステント・サイズのプリフェッチ要求が存在します。表スペース内のコンテナの数は、プリフェッチャーの数に影響しません。すべての表スペースの並列入出力を使用可能にするには、ワイルドカード文字「*」を使用します。すべての表スペースのサブセットの並列入出力を使用可能にするには、表スペースのリストを入力します。複数のコンテナが存在する場合、エクステント・サイズのフル・プリフェッチ要求は、プリフェッチャーの数に基づいて並列に実行されるようにより小さな要求に分割されます。</p> <p>この変数が使用可能ではない場合、作成されたプリフェッチャー要求の数は表スペース内のコンテナ数に基づいています。</p>		
DB2PATH	Windows オペレー	デフォルト: (オペレーティング・システムによって異なる)
この製品が Windows 32 ビットのオペレーティング・システム上に、インストールされるディレクトリを指定するために使用される環境変数。		
DB2_USE_PAGE_CONTAINER_TAG	すべて	デフォルト: null 値: ON、null
<p>デフォルトでは、DB2 はコンテナ・タグを各 DMS コンテナ (それがファイルでも装置でも) の最初のエクステントに保管します。コンテナ・タグは、コンテナのメタデータです。DB2 バージョン 8.1 より前には、コンテナ・タグは単一のページに保管されたので、コンテナ内でより少ないスペースだけを必要としました。継続してコンテナ・タグを単一のページに保管するには、DB2_USE_PAGE_CONTAINER_TAG を ON に設定します。</p> <p>しかし、コンテナに RAID 装置を使用する場合にこのレジストリー変数を ON に設定すると、入出力パフォーマンスは低下することがあります。RAID 装置ではエクステント・サイズが RAID ストライプ・サイズと等しいかその倍数の表スペースを作成するので、DB2_USE_PAGE_CONTAINER_TAG を ON に設定すると、エクステントが RAID ストライプとラインアップしなくなります。その結果、入出力要求は最適な場合よりも多くの物理ディスクにアクセスしなければならないことがあります。このレジストリー変数を使用可能にしないことを強くお勧めします。</p> <p>このレジストリー変数に対する変更を有効にするには、DB2STOP コマンドを出して DB2START コマンドを入力します。</p>		

関連概念:

- 管理ガイド: パフォーマンス の『DB2 レジストリー変数と環境変数』

パフォーマンス変数

表 106. パフォーマンス変数

変数名	オペレーティング・システム	
説明		
DB2_APM_PERFORMANCE	すべて	デフォルト = OFF 値: ON、OFF
<p>SQL キャッシュ (パッケージ・キャッシュ) の動作に影響するアクセス・プラン・マネージャー (APM) 内でパフォーマンス関連の変更を使用可能にするには、この変数を ON に設定します。これらの設定値は通常、実動システムには勧められません。これらはパッケージ外キャッシュ・エラー、メモリー使用量の増加、またはその両方など、いくらかの制限を生じさせます。</p> <p>DB2_APM_PERFORMANCE を ON に設定すると、「パッケージ・ロックなし」モードも使用可能になります。このモードは、グローバル SQL キャッシュがパッケージ・ロックを使用しないで操作できるようにします。パッケージ・ロックは、キャッシュされたパッケージ項目が除去されないように保護する内部システム・ロックです。「パッケージ・ロックなし」モードにより、パフォーマンスはいくらか改善されることがあるが、特定のデータベース操作は許可されなくなります。これらの禁止される操作には、パッケージを無効にする操作、パッケージを操作不能にする操作、PRECOMPILE、BIND、およびREBINDが含まれます。</p>		
DB2_AVOID_PREFETCH	すべて	デフォルト = OFF 値: ON または OFF
<p>クラッシュ・リカバリーにおいて、プリフェッチを使用するかを指定します。 DB2_AVOID_PREFETCH=ON の場合は、プリフェッチは使用されません。</p>		
DB2_AWE	Windows 2000	デフォルト = null 値: <entry>[:<entry>...] (<entry>=<バッファ・プール ID>,<物理ページの数>,<アドレス・ウィンドウの数>)

表 106. パフォーマンス変数 (続き)

変数名	オペレーティング・システム	
説明		
<p>32 ビットの Windows 2000 プラットフォームの DB2 UDB では、バッファ・プールに 64 GB までのメモリーを割り振ることができます。Windows 2000 は、Address Windowing Extensions (AWE) バッファ・プールをサポートするように正しく構成されている必要があります。これには、「メモリー内のページのロック」権利をユーザーに関連付け、物理ページおよびアドレス・ウィンドウ・ページを割り振り、このレジストリー変数を設定する作業が含まれる。この変数を設定するには、AWE サポートに使用するバッファ・プールのバッファ・プール ID を知っておく必要があります。バッファ・プールの ID は、SYSCAT.BUFFERPOOLS システム・カタログ・ビューの BUFFERPOOLID 列にあります。</p> <p>注:</p> <ul style="list-style-type: none"> AWE サポートが使用可能になっている場合には、拡張ストレージをデータベースのバッファ・プールに対して使用することはできません。 このレジストリー変数を使用して参照されるバッファ・プールは、すでに SYSCAT.SYSBUFFERPOOLS に存在していなければなりません。 AWE で使用可能なバッファ・プールは、ブロック・ベース I/O で使用可能なバッファ・プールより優先します。バッファ・プールが AWE とブロック・ベース I/O の両方に構成されている場合は、AWE がブロック・ベース I/O より優先します。 		
DB2_BINSORT	すべて	デフォルト = YES 値: YES または NO
<p>ソートの CPU 時間と経過時間が減少する新しいソート・アルゴリズムを使用可能にします。この新アルゴリズムにより、DB2 UDB の非常に効率的な整数ソート技法が、あらゆるソート・データ・タイプ (BIGINT、CHAR、VARCHAR、FLOAT、DECIMAL、およびそれらを組み合わせたデータ・タイプ) に拡張されます。この新アルゴリズムを使用可能にするには、以下のコマンドを使用します。</p> <pre>db2set DB2_BINSORT = yes</pre>		
DB2BPVARS	各パラメーターに指 定されたとおり デフォルト = パス	

表 106. パフォーマンス変数 (続き)

変数名	オペレーティング・値 システム
説明	
<p>2 種類のパラメーターを使用してバッファ・プールを調整できます。パラメーターの 1 つの種類は Windows だけで使用可能なもので、バッファ・プールが特定のタイプのコンテナの分散読み取りを使用することを指定します。他の種類のパラメーターは、すべてのプラットフォームで使用可能なもので、プリフェッチ動作に影響を与えます。</p>	
<p>複数のパラメーターは ASCII ファイル内で、各行に 1 つずつ <code>parameter=value</code> の形式で指定できます。例として、<code>bpvars.vars</code> という名前のファイルには以下の行が含まれます。</p>	
<pre>NO_NT_SCATTER = 1 NUMPREFETCHQUEUES = 2</pre>	
<p><code>bpvars.vars</code> が <code>F:%vars%</code> に保管されていると想定すると、これらの変数を設定するには以下のコマンドを実行します。</p>	
<pre>db2set DB2BPVARS=F:%vars%bpvars.vars</pre>	
<p>分散読み取りパラメーター</p>	
<p>分散読み取りパラメーターは、それぞれのコンテナ・タイプに対する順次プリフェッチが大量に行われ、<code>DB2NTNOCACHE</code> をすでに <code>ON</code> に設定したシステムで推奨されます。これらのパラメーターは、Windows プラットフォームだけで使用可能であり、<code>NT_SCATTER_DMSFILE</code>、<code>NT_SCATTER_DMSDEVICE</code> および <code>NT_SCATTER_SMS</code> です。<code>NO_NT_SCATTER</code> パラメーターを指定すると、すべてのコンテナで分散読み取りを明示的に禁止します。指定の種類すべてのコンテナで分散読み取りをオンに切り替えるには、特定のパラメーターを使用します。上記の個々のパラメーターのデフォルトはゼロ (OFF) で、可能な値はゼロ (OFF) および 1 (ON) です。</p>	
<p>注: 分散読み取りをオンに切り替えることができるのは、<code>DB2NTNOCACHE</code> を <code>ON</code> に設定して Windows ファイルのキャッシングをオフにした場合だけです。<code>DB2NTNOCACHE</code> の設定が <code>OFF</code> に設定されているか未設定の場合は、管理通知ログに警告メッセージが書き込まれ、分散読み取りは使用不可のままになります。</p>	

表 106. パフォーマンス変数 (続き)

変数名	オペレーティング・システム	
説明		
<p>プリフェッチ調整パラメーター</p> <p>プリフェッチ調整パラメーターは、NUMPREFETCHQUEUES および PREFETCHQUEUESIZE です。これらのパラメーターはすべてのプラットフォームで使用可能であり、バッファ・プール・データ・プリフェッチを改善するために使用できます。たとえば、目的の PREFETCHSIZE が複数の PREFETCHSIZE/EXTENTSIZE プリフェッチ要求に分割されている順次プリフェッチについて考えます。この場合、要求は入出力サーバーが非同期入出力を実行するプリフェッチ・キューに入れられます。デフォルトでは、DB2 はデータベース・パーティションごとにサイズ $\max(100, 2 * \text{NUM_IOSERVERS})$ の 1 つのキューを保守します。一部の環境では、キューの増加またはキューのサイズの変更、もしくはその両方により、パフォーマンスが改善されます。プリフェッチ・キューの数は最大で入出力サーバーの数の半分とします。これらのパラメーターを設定するとき、現行ユーザー数などのワークロード特性と同様に、PREFETCHSIZE、EXTENTSIZE、NUM_IOSERVERS などのパラメーター、およびバッファ・プール・サイズを検討してください。</p> <p>デフォルト値が環境に対して小さすぎると考えられる場合、最初に少しだけ値を増加します。たとえば、NUMPREFETCHQUEUES=4 および PREFETCHQUEUESIZE=200 と設定できます。これらのパラメーターの変更を制御された方法で行い、変更の効果を観察および評価できるようにします。</p> <p>NUMPREFETCHQUEUES では、デフォルトは 1 で、値の範囲は 1 から NUM_IOSERVERS となります。NUMPREFETCHQUEUES を 1 未満に設定した場合、それは 1 に調整されます。その値を NUM_IOSERVERS よりも大きく設定した場合、それは NUM_IOSERVERS に調整されます。</p> <p>PREFETCHQUEUESIZE では、デフォルト値は $\max(100, 2 * \text{NUM_IOSERVERS})$ です。値の範囲は 1 ~ 32767 です。PREFETCHQUEUESIZE を 1 未満に設定した場合、それはデフォルト値に調整されます。その値を 32767 より大きく設定した場合、それは 32767 に調整されます。</p>		
DB2CHKPTR	すべて	デフォルト = OFF 値: ON または OFF
入力のパインター検査が必要であるか否かを指定します。		
DB2_ENABLE_BUFDPD	すべて	デフォルト = OFF 値: ON または OFF
照会パフォーマンスを向上させるために DB2 で中間バッファリングを使用するかどうかを指定します。バッファリングによって、あらゆる環境で照会パフォーマンスが向上するわけではありません。それぞれの照会パフォーマンスが向上するかどうかを判別するために、テストを行う必要があります。		
DB2_EXTENDED_OPTIMIZATION	すべて	デフォルト = OFF 値: ON または OFF

表 106. パフォーマンス変数 (続き)

変数名	オペレーティング・システム	
説明		
照会パフォーマンスを向上させるために、照会オプティマイザーが最適化拡張を使用するかどうかを指定します。拡張によって、あらゆる環境で照会パフォーマンスが向上するわけではありません。それぞれの照会パフォーマンスが向上するかどうかを判別するために、テストを行う必要があります。		
DB2MAXFSCRSEARCH	すべて	デフォルト= 5 値: -1、1 ~ 33 554
表にレコードが追加された場合、検索のために、フリー・スペース制御レコードの数を指定します。デフォルトは、フリー・スペース制御レコードを 5 つ検索します。この値の変更は、スペース再利用で挿入速度の平衡を取れるようにします。スペース再利用の最適化のためには大きな値を使用します。挿入速度の最適化のためには小さな値を使用します。値を -1 に設定するとデータベース・マネージャーはすべてのフリー・スペース制御レコードを強制的に検索します。		
DB2MEMDISCLAIM	AIX	デフォルト = YES 値: YES または NO
AIX では、DB2 処理が使用するメモリーに、関連するページング・スペースを存在させることができません。このページング・スペースは、関連するメモリーが解放された後も予約されたままになる場合があります。これは、AIX システムの (調整可能な) 仮想メモリー管理割り振りポリシーによって異なります。DB2MEMDISCLAIM レジストリー変数は、解放されたメモリーと予約ページング・スペースとの関連付けを AIX が解除することを DB2 エージェントが明示的に要求するかどうかを制御します。		
DB2MEMDISCLAIM を YES に設定すると、ページング・スペースの所要量が少なくなり、ページングのディスク活動も減少します。DB2MEMDISCLAIM を NO に設定すると、ページング・スペースの所要量は多くなり、ページングのディスク活動も増加します。ページング・スペースが多い場合や、ページングが行われないほど実メモリーが十分にある場合などは、NO を設定してもパフォーマンスはわずかしか向上しません。		
DB2MEMMAXFREE	すべて	デフォルト= 8 388 608 バイト 値: 0 ~ 2 ³² -1 バイト
未使用のメモリーがオペレーティング・システムに戻される前に DB2 プロセスによって保存される、未使用の専用メモリーの最大バイト数を指定します。		
DB2_MMAP_READ	AIX	省略値 = ON 値: ON または OFF

表 106. パフォーマンス変数 (続き)

変数名	オペレーティング・システム	
説明		
<p>DB2_MMAP_WRITE と一緒に使用して、入出力の代替方法として DB2 が mmap を使用できるようにします。複数プロセスが同一ファイルの異なるセクションから読み取る場合は、ほとんどの環境において、オペレーティング・システムのロックを防ぐために mmap を使用すべきです。しかし、デフォルトが OFF の Parallel Edition V1.2 から移行した場合は、(バッファ・プールとは無関係に) JFS ファイル・システムからメモリーへ読み取る DB2 データの AIX によるキャッシングが可能になっていることがあります。DB2 UDB と同等のパフォーマンスにしたい場合、バッファ・プールのサイズを増やすか、DB2_MMAP_READ と DB2_MMAP_WRITE を OFF に変更することができます。</p>		
DB2_MMAP_WRITE	AIX	省略値 = ON 値: ON または OFF
<p>DB2_MMAP_READ と一緒に使用して、入出力の代替方法として DB2 が mmap を使用できるようにします。複数プロセスが同一ファイルの異なるセクションに書き出す場合は、ほとんどの環境において、オペレーティング・システムのロックを防ぐために mmap を使用すべきです。しかし、デフォルトが OFF の Parallel Edition V1.2 から移行した場合は、(バッファ・プールとは無関係に) JFS ファイル・システムからメモリーに読み取る DB2 データの AIX キッシングが可能になっていることがあります。DB2 UDB と同等のパフォーマンスにしたい場合、バッファ・プールのサイズを増やすか、DB2_MMAP_READ と DB2_MMAP_WRITE を OFF に変更することができます。</p>		
DB2NTMEMSIZE	Windows NT	デフォルト = (メモリー・セグメントにより異なる)
<p>Windows NT では、プロセス間でアドレスの一致を保証するために DLL 初期設定時にすべての共有メモリー・セグメントを予約する必要があります。必要に応じて DB2NTMEMSIZE により Windows NT の DB2 デフォルトをオーバーライドできます。ほとんどの状態では、デフォルト値で十分なはずですが。メモリー・セグメント、デフォルトのサイズ、およびオーバーライド・オプションは以下のとおりです。1) データベース・カーネル: デフォルトのサイズ 16777216 (16 MB); オーバーライド・オプションは DBMS:<number of bytes> 2) 並列 FCM バッファ: デフォルトのサイズ 22020096 (21 MB); オーバーライド・オプションは FCM:<number of bytes> 3) データベース Admin GUI: デフォルトのサイズ 33554432 (32 MB); オーバーライド・オプションは DBAT:<number of bytes> 4) 分離ストアード・プロシージャ: デフォルトのサイズ 16777216 (16 MB); オーバーライド・オプションは APLD:<number of bytes>。オーバーライド・オプションをセミコロン (;) で区切って、複数のセグメントをオーバーライドできます。たとえば、データベース・カーネルを 256K に制限し、FCM バッファを 64 MB に制限するには、以下のようにします。</p> <p>db2set DB2NTMEMSIZE=DBMS:256000;FCM:64000000</p>		
DB2NTNOCACHE	Windows NT	デフォルト = OFF 値: ON または OFF

表 106. パフォーマンス変数 (続き)

変数名	オペレーティング・システム 値	
説明		
<p>DB2 がデータベース・ファイルを NOCACHE オプションを指定してオープンするかどうかを指定します。 DB2NTNOCACHE=ON の場合は、ファイル・システムのキャッシュは除去されます。 DB2NTNOCACHE=OFF の場合、オペレーティング・システムは DB2 ファイルをキャッシュに入れます。これは、long fields または LOBs を含んでいるファイルを除くすべてのデータに適用されます。システム・キャッシュを除去すると、より多くのメモリーがデータベースに利用できるようになるため、バッファー・プールやソート・ヒープの量を増すことができます。</p> <p>Windows NT では、デフォルトの動作として、ファイルをオープンするときにキャッシュに入れられません。ファイル内の 1 GB ごとに 1 MB がシステム・プールから予約されます。このレジストリー変数を使用して、キャッシュに関する (文書化されていない) 192 MB 制限をオーバーライドします。キャッシュの限界に達すると、リソース不足を示すエラーが表示されます。</p>		
DB2NTPRCLASS	Windows NT	デフォルト = null 値: R、H、(任意の他の値)
<p>DB2 インスタンスの優先度クラスを設定します (プログラム DB2SYSCS.EXE)。以下の 3 つの優先度クラスがあります。</p> <ul style="list-style-type: none"> • NORMAL_PRIORITY_CLASS (デフォルトの優先度クラス) • REALTIME_PRIORITY_CLASS (「R」を使って設定) • HIGH_PRIORITY_CLASS (「H」を使って設定) <p>この変数は、個々のスレッド優先順位 (DB2PRIORITIES を使って設定) と一緒に使われ、システム中の別のスレッドに関する DB2 スレッドの絶対優先順位を決定します。</p> <p>注: この変数を使用する際には、注意する必要があります。誤用すると、システム・パフォーマンス全体に悪い影響を及ぼす可能性があります。</p> <p>詳細は、Win32 ドキュメンテーションの <i>SetPriorityClass()</i> API を参照してください。</p>		
DB2NTWORKSET	Windows NT	デフォルト = 1,1
<p>DB2 に利用できる最小および最大の実効ページ・セットを変更するために使用されます。デフォルトを使用して、ページングが行われていない場合は、プロセスの実効ページ・セットは必要なだけ大きくすることができます。ただし、ページングが発生しているときは、プロセスが持つことができる最大の実効ページ・セットは約 1 MB です。 DB2NTWORKSET を使えば、このデフォルトの動作をオーバーライドできます。</p> <p>DB2 に対する DB2NTWORKSET の指定は、DB2NTWORKSET=min,max の構文を使用します。ここで、min と max はメガバイト単位で表されます。</p>		

表 106. パフォーマンス変数 (続き)

変数名	オペレーティング・値 システム	
説明		
DB2_OVERRIDE_BPF	すべて	デフォルト = 設定しない 値: 正の数値ページの数または、 <entry>[;<entry>...] (<entry>=<バッファ ー・プール ID>,<ページの数>)
<p>データベース活動化時または初回接続時に作成されるバッファー・プールのサイズをページ単位で指定します。これが役立つのは、メモリー制約の結果としてデータベース活動化時または初回接続時に障害が発生する場合です。データベース・マネージャーによって最小のバッファー・プール (16 ページ) も起動しない場合、ユーザーはこの環境変数を使ってさらに小さいページ数を指定した後に再試行することができます。メモリー制約は、実メモリーの不足 (めったに起こらない) のために起こることもあれば、データベース・マネージャーが間違った構成の大きいバッファー・プールを割り当てようとしたために起こることもあります。設定時に、この値は現行バッファー・プールをオーバーライドします。</p>		
<p>バッファー・プールのすべてまたはサブセットのサイズを一時的に変更して始動できるように、以下も使用できます。 <entry>[;<entry>...] (<entry>=<バッファー・プール ID>,<ページの数>)</p>		
DB2_PINNED_BP	AIX、HP-UX	デフォルト = NO 値: YES または NO

表 106. パフォーマンス変数 (続き)

変数名	オペレーティング・値 システム	
説明		
<p>この変数は、一部の AIX オペレーティング・システムで、データベースに関連するデータベース・グローバル・メモリー (バッファ・プールを含む) をメイン・メモリーに指定するために使用されます。データベース・グローバル・メモリーをシステム・メイン・メモリーに保持することにより、データベース・パフォーマンスがより一貫性のあるものになります。</p> <p>たとえば、バッファ・プールがシステム・メイン・メモリーからスワップアウトされる場合、データベース・パフォーマンスは低下します。バッファ・プールをシステム・メモリーに保持することによってディスク入出力が減ると、データベース・パフォーマンスは改善されます。別のアプリケーションがより多くのメイン・メモリーを要求した場合、システム・メイン・メモリー所要量に応じて、データベース・グローバル・メモリーをメイン・メモリーからスワップアウトできます。</p> <p>64 ビット環境での HP-UX では、このレジストリー環境を変更する他に、DB2 インスタンス・グループに MLOCK 特権を与えなければなりません。これを行うには、ルート・アクセス権限を持つユーザーが以下の処置を実行します。</p> <ol style="list-style-type: none"> DB2 インスタンス・グループを /etc/privgroup ファイルに追加します。たとえば、DB2 インスタンス・グループが db2iadm1 グループに属している場合、以下の行を /etc/privgroup ファイルに追加します。 db2iadm1 MLOCK 以下のコマンドを発行します。 setprivgrp -f /etc/privgroup 		
DB2PRIORITIES	すべて	値の設定はプラットフォームにより異なる
DB2 プロセスとスレッドの優先順位を制御します。		
DB2_SORT_AFTER_TQ	すべて	デフォルト = NO 値: YES または NO
<p>受信終了時にデータをソートすることが必要で、受信ノード数が送信ノード数と等しい場合、オプティマイザーがパーティション・データベースの直接表キューを処理する方法を指定します。</p> <p>DB2_SORT_AFTER_TQ=NO の場合、オプティマイザーは送信終了時には行のソートを、受信終了時には行のマージを行う傾向があります。</p> <p>DB2_SORT_AFTER_TQ=YES の場合、オプティマイザーはソートをしないで行を送信し、すべての行を受信した後の受信終了時にもマージを行わない傾向があります。</p>		
DB2_STPROC_LOOKUP_FIRST	すべて	デフォルト = OFF 値: ON または OFF

表 106. パフォーマンス変数 (続き)

変数名	オペレーティング・値 システム
説明	
<p>以前の DB2_DARI_LOOKUP_ALL。この変数は、sqllib サブディレクトリーの function サブディレクトリー、および sqllib サブディレクトリーの function サブディレクトリーの unfenced サブディレクトリーを検索する前に、UDB サーバーがすべての DARI およびストアード・プロシージャのカタログ検索を実行するかどうかを指定します。</p> <p>注: 上記のディレクトリーにある PARAMETER TYPE DB2DARI のストアード・プロシージャの場合、この値を ON に設定するとパフォーマンスが低下します。これは、カタログ検索が、関数ディレクトリーを検索する前に ESE 構成内の別のノードで実行されることがあるからです。</p> <p>ストアード・プロシージャを呼び出す際の DB2 のデフォルトの振る舞いは、sqllib サブディレクトリーの function サブディレクトリー、および sqllib サブディレクトリーにある function サブディレクトリーの unfenced サブディレクトリーから、ストアード・プロシージャと同じ名前を持つ共有ライブラリーを検索することです。これは、システム・カタログ内のストアード・プロシージャの共有ライブラリー名を検索する前に行われます。共有ライブラリーと同じ名前を持つのは PARAMETER TYPE DB2DARI のストアード・プロシージャだけであるため、DB2DARI ストアード・プロシージャだけが DB2 のデフォルトの振る舞いから益を受けます。異なる PARAMETER TYPE を使用してカタログを作成したストアード・プロシージャを使用する場合、DB2 が上記のディレクトリーを検索するために費やす時間のため、それらのストアード・プロシージャのパフォーマンスは低下します。</p> <p>PARAMETER TYPE DB2DARI としてカタログを作成していないストアード・プロシージャのパフォーマンスを向上させるには、DB2_STPROC_LOOKUP_FIRST レジストリー変数を ON に設定します。このレジストリー変数は、DB2 が上記のディレクトリーを検索する前に、システム・カタログ内のストアード・プロシージャの共有ライブラリー名を検索するよう強制します。</p>	

関連概念:

- 管理ガイド: パフォーマンス の『DB2 レジストリー変数と環境変数』

再配布ストアード・プロシージャおよび関数

ステップ単位の再配布ストアード・プロシージャは、データベース・パーティション・グループを、いくつかのステップで安全に再配布するのに使用できます。

1. ログ・スペースの可用性およびデータ・スキューに関してデータベース・パーティション・グループを分析します。
2. 指定した表のデータ配分ファイルを作成します。
3. データベース・パーティション・グループのステップ単位再配布プランの内容を作成し、報告します。
4. プランにしたがって、データベース・パーティション・グループを再配布します。

以下のプロシージャーで、個別のパラメーターを扱っている場合、値を取得できないときは、パラメーターの出力値に「-1」が使用されます。

注: 再配布ストアード・プロシージャーおよび関数は、それぞれの表に区分化キーが定義されているパーティション・データベースでのみ使用可能です。

get_swrд_settings ストアード・プロシージャー

get_swrд_settings 関数は、指定したデータベース・パーティション・グループの既存の再配布レジストリー・レコードを読み取ります。

表 107. get_swrд_settings 入力パラメーター

名前	データ・タイプ	説明
dbpgName	VARCHAR(128)	再配布プロセスが実行される、データベース・パーティション・グループ名。
matchingSpec	SMALLINT	表 108 のビット単位フィールド ID で、表 109 の出力パラメーターによって返されるターゲット・フィールドを示します。これらの必要ない出力パラメーターは、NULL 設定できます。 たとえば、matchingSpec が (REDIST_STAGE_SIZE REDIST_NEXT_STEP) の整数値である 96 に設定されている場合、この関数の呼び出し元は、値を受け取るために stageSize および nextStep を提供する必要があるだけで、出力パラメーターの残りは NULL にできます。

表 108. ビット単位フィールド ID

フィールド名	16 進値	10 進値
REDIST_METHOD	0x0001<<0	1
REDIST_PMAP_FILE	0x0001<<1	2
REDIST_DIST_FILE	0x0001<<2	4
REDIST_STEP_SIZE	0x0001<<3	8
REDIST_NUM_STEPS	0x0001<<4	16
REDIST_STAGE_SIZE	0x0001<<5	32
REDIST_NEXT_STEP	0x0001<<6	64
REDIST_PROCESS_STATE	0x0001<<7	128
REDIST_PWEIGHT_START_NODE	0x0001<<8	256
REDIST_PWEIGHT	0x0001<<9	512

表 109. get_swrд_settings 出力パラメーター

名前	データ・タイプ	説明
redistMethod	SMALLINT	配布ファイルまたはターゲット区分化マップを使用して再配布が実行されることを示す数。

表 109. `get_swrd_settings` 出力パラメーター (続き)

名前	データ・タイプ	説明
<code>pMapFile</code>	VARCHAR(255)	ターゲット区分化マップの絶対パス名。
<code>distFile</code>	VARCHAR(255)	データ配布ファイルの絶対パス名。
<code>stepSize</code>	BIGINT	ログがフルになる状態を避けるために、コミットが呼び出される前に移動できる、行の最大数。数は、各配布ステップで移動できます。
<code>totalSteps</code>	SMALLINT	指定したデータベース・パーティション・グループを、完全に再配布するのに必要なステップの数。
<code>stageSize</code>	SMALLINT	連続して実行されるステップの数。
<code>nextStep</code>	SMALLINT	完了したステップおよび、実行する必要のあるステップを区切る索引。
<code>processState</code>	SMALLINT	<code>nextStep</code> で再配布ステージを停止するのにユーザーが設定できるフラグ。
<code>pNumber</code>	VARCHAR(6000)	パーティション重みに対応するデータベース・パーティション・グループ内のすべてのパーティション番号で移植される事前割り当ての文字列。文字列内のパーティション番号は「,」で区切られています。
<code>pWeight</code>	VARCHAR(6000)	<code>SET_SWRD_SETTINGS</code> ストアード・プロシージャで指定される各パーティション上のボリュームのすべての相対重みで移植される事前割り当ての文字列。文字列内のパーティション重みは「,」で区切られています。

`set_swrd_settings` ストアード・プロシージャ

`set_swrd_settings` 関数は、再配布レジストリーを作成または変更します。レジストリーが存在しない場合は作成され、レコードが追加されます。レジストリーがすでに存在する場合は、`overwriteSpec` を使用して上書きされる必要のあるフィールド値を識別します。`overwriteSpec` フィールドはこの関数を使用可能にして、更新される必要のないフィールドに `NULL` 入力を指定します。

このプロシージャでは、数が無制限であることを示すのに、`stepSize` および `totalSteps` に値「-2」を使用することができます。

表 110. `set_swrd_settings` 入力パラメーター

名前	データ・タイプ	説明
<code>dbpgName</code>	VARCHAR(128)	再配布プロセスが実行される、データベース・パーティション・グループ名。
<code>overwriteSpec</code>	SMALLINT	356 ページの表 108 のビット単位フィールド ID で、再配布上書きレジストリーに書き込み、または上書きされるターゲット・フィールドを示します。
<code>redistMethod</code>	SMALLINT	配布ファイルまたはターゲット区分化マップを使用して再配布が実行されることを示す数。

表 110. `set_swrd_settings` 入力パラメーター (続き)

名前	データ・タイプ	説明
<code>pMapFile</code>	VARCHAR(255)	ターゲット区分化マップの絶対パス名。
<code>distFile</code>	VARCHAR(255)	データ配布ファイルの絶対パス名。
<code>stepSize</code>	BIGINT	ログがフルになる状態を避けるために、コミットが呼び出される前に移動できる、行の最大数。数は、各配布ステップで移動できません。
<code>totalSteps</code>	SMALLINT	指定したデータベース・パーティション・グループを、完全に再配布するのに必要なステップの数。
<code>stageSize</code>	SMALLINT	連続して実行されるステップの数。
<code>nextStep</code>	SMALLINT	完了したステップおよび、実行する必要があるステップを区切る索引。
<code>processState</code>	SMALLINT	<code>nextStep</code> で再配布ステージを停止するのにユーザーが設定できるフラグ。
<code>pNumber</code>	VARCHAR(6000)	パーティション重みに対応したすべてのパーティション番号を含むストリング。各パーティション番号は 0 から 999 で、番号は「,」で区切られています。ストリング内にスペースは使用できません。
<code>pWeight</code>	VARCHAR(6000)	<code>pNumber</code> ストリング内のパーティション番号に対応する、ユーザーが指定したすべてのパーティション重みを含むストリング。各パーティション重みは 0 から <code>SQL_MAXSMALLVAL</code> の番号で、番号は「,」で区切られています。ストリング内にスペースは使用できません。

analyze_log_space ストアード・プロシージャ

`analyze_log_space` 関数は、ログ・スペース分析の結果セット (オープン・カーソル) を返します。この中には、指定されたデータベース・パーティション・グループの各データベース・パーティションについて、以下のフィールドが含まれます。

表 111. `analyze_log_space` フィールド

列名	列タイプ	説明
<code>PARTITION_NUM</code>	SMALLINT	ログ・スペース分析のパーティション番号
<code>TOTAL_LOG_SIZE</code>	BIGINT	割り振られたログ・スペースの合計 (バイト単位)。-1 はサイズが無制限であることを示します。
<code>AVAIL_LOG_SPACE</code>	BIGINT	空いていて、再配布プロセスに使用できるログ・スペースの量 (バイト単位)
<code>DATA_SKEW</code>	BIGINT	ターゲット・レベルからはずれたデータのサイズの絶対値 (バイト単位)
<code>REQ_LOG_SPACE</code>	BIGINT	所要のデータ配分に達するのに必要なスペースの量 (バイト単位)
<code>NUM_OF_STEPS</code>	SMALLINT	データ・スキューをゼロまで減らすのに必要なステップの数
<code>MAX_STEP_SIZE</code>	BIGINT	ログ・フル・エラーを起こさずに一度に移動できる、データの最大量 (バイト単位)

表 112. analyze_log_space 入力パラメーター

名前	データ・タイプ	説明
inDBPGroup	VARCHAR(128)	データベース・パーティション・グループ名
inMainTbSchema	VARCHAR(128)	メイン表のスキーマ
inMainTable	VARCHAR(128)	データベース・パーティション・グループ内のメイン表 (dbpg 内最大の表である場合が多い)
useTbType	SMALLINT	分析タイプのインディケーター: SWRD_USE_STMG_TABLE 1: パーティションごとの行数を検出する、ストレージ管理表の情報の使用を示す。 SWRD_USE_REALTIME_ANALYSIS 2: パーティションごとの行数を検出する、選択照会の使用を示す。
inStmgTime	VARCHAR(26)	ストレージ管理レコードのタイム・スタンプ。このパラメーターは、analysisType が SWRD_USE_REALTIME_ANALYSIS に設定されている場合は無視されます。
addDropOption	CHAR(1)	パーティションの追加またはドロップのインディケーター: 'A' パーティションの追加 'D' パーティションのドロップ 'N' 追加またはドロップなし
addDropList	VARCHAR(6000)	追加またはドロップされるパーティションのリスト (コンマで区切られた文字列)
pNumber	VARCHAR(6000)	パーティション重みに対応したすべてのパーティション番号を含む文字列。各パーティション番号は 0 から 999 で、番号は「,」で区切られています。文字列内にスペースは使用できません。
pWeight	VARCHAR(6000)	pNumber 文字列内のパーティション番号に対応する、ユーザーが指定したすべてのパーティション重みを含む文字列。各パーティション重みは 0 から SQL_MAXSMALLVAL の番号で、番号は「,」で区切られています。文字列内にスペースは使用できません。

generate_Distfile ストアド・プロシージャ

generate_Distfile 関数は、指定された表のデータ配分ファイルを生成し、指定された fileName の下に保管します。

表 113. generate_Distfile、入力パラメーター

名前	データ・タイプ	説明
inTbSchema	VARCHAR(128)	表スキーマ名
inTbName	VARCHAR(128)	表名

表 114. `generate_Distfile`、入出力パラメーター

名前	データ・タイプ	説明
<code>fileName</code>	VARCHAR(255)	データ配分ファイル名。指定されたファイル名が、通常のファイル名の場合、ファイルは <code>instance/tmp</code> ディレクトリーに保管され、完全ファイル・パス名がパラメーターによって返されます。

stepwise_redistribute_dbpg ストアード・プロシージャー

`stepwise_redistribute_dbpg` 関数は、入力および設定ファイルにしたがって、データベース・パーティション・グループの一部を再配布します。

ステップ 1. 設定レジストリーは、`inDbPGroup` 名を使用して検索されます。

- レジストリーが検出されなかった場合、エラーが返されます。
- レジストリーが検出された場合、以下の値が読み取られます。
 - 現行ステップ
 - ステップの数
 - 区分化マップ名
 - 最大データ・サイズ (各ステップで移動可能)
 - プロセスの状態
 - パーティションの重み

`inNumSteps` に `-1 (SWRD_UNLIMITED_STEPS)` を入力することで、データベース・パーティション・グループをすべて同時に再配布できます。

ステップ 2. レジストリー処理状態が `SWRD_STOP` の場合、処理は停止され、警告メッセージが返されます。処理状態が `SWRD_CONTINUE` の場合、処理は継続されます。

ステップ 3. `inStartingPoint` が `NULL` でなく有効な場合、対応するレジストリー値は上書きされます。そうでない場合、現行ステップのレジストリー値が読み取られ、このステップの開始点として使用されます。

ステップ 4. 次に現行ステップの区分化マップが生成され、パーティション重みは設定レジストリーから検索されます。既存の完全な区分化マップまたは `distfile` がある場合は、それに合わせて区分化マップが生成されます。

ステップ 5. 再配布 API は、区分化マップ・オプションを使用して呼び出されます。プロセスが完了すると、次のステップのレジストリー値が、増加します。

ステップ 1 から 5 は、レジストリーに指定されたステップ数繰り返されます。

このプロシージャーでは、数が無制限であることを示すのに、値「-2」を使用することができます。

表 115. *stepwise_redistribute_dbpg* 入力パラメーター

名前	データ・タイプ	説明
inDBPGroup	VARCHAR(128)	ターゲット・データベース・パーティション・グループの名前
inStartingPoint	SMALLINT	このパラメーターは、NULL にすることができます。NULL でなく、正数を指している場合は、swrd 設定レジストリーに指定された「nextStep」値を上書きします。これは、特定のステップから SWRD を再実行したい場合に便利なオプションです。
inNumSteps	SMALLINT	実行するステップの数。NULL でなく、正数を指している場合は、swrd 設定レジストリーに指定された「numSteps」値を上書きします。これは、SWRD を設定に指定されているのと違うステップ数で再実行したい場合に、便利なオプションです。たとえば、スケジュールされたステージに 5 つのステップがあり、SWRD プロセスがステップ 3 で失敗した場合、エラー状態を訂正したあとで、残りの 3 つのステップを実行するのに SWRD を呼び出すことができます。

db_partitions UDF

db_partitions ユーザー定義関数は db2nodes.cfg ファイルを構文解析し、検出されたパーティションごとに 1 行を返します。

入力パラメーター: なし

表 116. *db_partitions* 出力パラメーター

名前	データ・タイプ	説明
PARTITION_NUMBER	SMALLINT	パーティション番号
HOST_NAME	VARCHAR(128)	ホスト名 (Intel プラットフォームでは、これはマシン名です)
PORT_NUMBER	SMALLINT	論理ポート番号
SWITCH_NAME	VARCHAR(128)	ネット・スイッチ名 (NULL でも可)

使用例

以下に示すのは、AIX での CLP スクリプトの例です。

```
# -----
# Set the database you wish to connect to
# -----
dbName="SAMPLE"

# -----
# Set the target database partition group name
# -----
dbpgName="IBMDEFAULTGROUP"

# -----
# Specify the table name and schema
# -----
tbSchema="$USER"
```

```

| tbName="STAFF"
|
| # -----
| # Specify the name of the data distribution file
| # -----
| distFile="$HOME/sql1lib/function/$dbName.IBMDEFAULTGROUP_swrData.dst"
|
| export DB2INSTANCE=$USER
| export DB2COMM=TCPIP
|
| # -----
| # Invoke call statements in clp
| # -----
| db2start
| db2 -v "connect to $dbName"
|
| # -----
| # Analysing the effect of adding a partition without applying the changes - a 'what if'
| # hypothetical analysis
| #
| # - In the following case, the hypothesis is adding partition 40, 50 and 60 to the
| # database partition group, and for partitions 10,20,30,40,50,60, using a respective
| # target ratio of 1:2:1:2:1:2.
| #
| # NOTE: in this example only partitions 10, 20 and 30 actually exist in the database
| # partition group
| # -----
| db2 -v "call sysproc.analyze_log_space('$dbpgName', '$tbSchema', '$tbName', 2, ' ',
| 'A', '40,50,60', '10,20,30,40,50,60', '1,2,1,2,1,2')"
|
| # -----
| # Analysing the effect of dropping a partition without applying the changes
| #
| # - In the following case, the hypothesis is dropping partition 30 from the database
| # partition group, and redistributing the data in partitions 10 and 20 using a
| # respective target ratio of 1 : 1
| #
| # NOTE: In this example all partitions 10, 20 and 30 should exist in the database
| # partition group
| # -----
| db2 -v "call sysproc.analyze_log_space('$dbpgName', '$tbSchema', '$tbName', 2, ' ',
| 'D', '30', '10,20','1,1')"
|
| # -----
| # Generate a data distribution file to be used by the redistribute process
| # -----
| db2 -v "call sysproc.generate_distfile('$tbSchema', '$tbName', '$distFile')"
|
| # -----
| # Write a step wise redistribution plan into a registry
| #
| # Setting the 10th parameter to 1, may cause a currently running step wise redistribute
| # stored procedure to complete the current step and stop, until this parameter is reset
| # to 0, and the redistribute stored procedure is called again.
| # -----

```

```

| db2 -v "call sysproc.set_swrd_settings('$dbpgName', 255, 0, ' ', '$distFile', 1000,
| 12, 2, 1, 0, '10,20,30', '50,50,50')"
|
| # -----
| # Report the content of the step wise redistribution plan for the given database
| # partition group.
| # -----
| db2 -v "call sysproc.get_swrd_settings('$dbpgName', 255, ?, ?, ?, ?, ?, ?, ?, ?, ?)"
|
| # -----
| # Redistribute the database partition group "dbpgName" according to the redistribution
| # plan stored in the registry by set_swrd_settings. It starting with step 3 and
| # redistributes the data until 2 steps in the redistribution plan are completed.
| # -----
| db2 -v "call sysproc.stepwise_redistribute_dbpg('$dbpgName', 3, 2)"

```

「照合情報」構成パラメーター - collate_info

このパラメーターを表示できるのは、db2CfgGet API を使用した場合だけです。コマンド行プロセッサやコントロール・センターでは表示できません。

構成タイプ	データベース
パラメーター・タイプ	通知

このパラメーターは、260 バイトのデータベース照合情報を提供します。最初の 256 バイトでデータベース照合シーケンスを指定するのに対して、バイト「n」には、データベースのコード・ページで基本 10 進表記が「n」になっている、コード・ポイントのソート重みが入ります。

最後の 4 バイトには、照合シーケンスのタイプについての内部情報が入ります。これは、データベースのプラットフォームに適用できる整数として扱うことができます。次の 3 つの値があります。

- 0 - シーケンスに非ユニークの重みが含まれる
- 1 - シーケンスにユニークの重みすべてが含まれる
- 2 - シーケンスは ID シーケンスで、ストリングがバイトごとに比較される

この内部タイプ情報を使用する場合は、別のプラットフォームにあるデータベースに関する情報を検索するときに、バイト反転を考慮する必要があります。

照合シーケンスは、データベース作成時に指定できます。

「データベース・システム・モニター・ヒープ・サイズ」構成パラメーター - mon_heap_sz

構成タイプ	データベース・マネージャー
適用	

- ローカル・クライアントおよびリモート・クライアントを指定したデータベース・サーバー
- ローカル・クライアントを指定したデータベース・サーバー
- ローカル・クライアントおよびリモート・クライアントを指定したパーティション・データベース・サーバー

パラメーター・タイプ

構成可能

デフォルト[範囲]

UNIX 90 [0 - 60 000]

Windows ローカル・クライアントおよびリモート・クライアントを指定したデータベース・サーバー 66 [0 - 60 000]

Windows ローカル・クライアントを指定したデータベース・サーバー 46 [0 - 60 000]

計測単位

ページ (4 KB)

割り振られるタイミング

データベース・マネージャーが *db2start* コマンドで開始されるとき

解放されるタイミング

データベース・マネージャーが *db2stop* コマンドで停止されたとき

このパラメーターは、データベース・システム・モニター・データに割り振られるメモリーの大きさ (ページ数) を決定します。メモリーは、スナップショットの取得、モニター・スイッチのオン、モニターのリセット、あるいはイベント・モニターの活動化といった、データベースのモニター活動を実行するときにモニター・ヒープから割り振られます。

ゼロの値を指定すると、データベース・マネージャーはデータベース・システム・モニター・データを収集しません。

推奨: 活動のモニターに必要なメモリーの量は、スイッチが設定されるモニター・アプリケーション (スナップショットまたはイベント・モニターを取るアプリケーション) の数と、データベース活動のレベルに依存します。

以下の計算式では、モニター・ヒープに必要なおおよそのページ数が算出されます。

$$\frac{(\text{モニター・アプリケーションの数} + 1) * (\text{データベースの数} * (800 + (\text{アクセスされた表の数} * 20) + ((\text{接続されたアプリケーションの数} + 1) * (200 + (\text{表スペースの数} * 100))))}{4096}$$

このヒープ内の使用可能なメモリーが使い尽くされた場合、以下のいずれかが行われま
す。

- 最初のアプリケーションが、このイベント・モニターが定義されているデータベース
に接続すると、エラー・メッセージが管理通知ログに書き込まれます。
- SET EVENT MONITOR ステートメントを使用して動的に開始されているイベント・
モニターが失敗した場合は、エラー・コードがアプリケーションに戻されます。
- モニター・コマンドまたは API サブルーチンが失敗した場合は、エラー・コードが
アプリケーションに戻されます。

関連資料:

- 管理ガイド: パフォーマンス の『「デフォルトのデータベース・システム・モニタ
ー・スイッチ」構成パラメーター - dft_monswitches』

「専用メモリーしきい値」構成パラメーター - priv_mem_thresh

構成タイプ

データベース・マネージャー

適用

- ローカル・クライアントおよびリモート・クライ
アントを指定したデータベース・サーバー
- ローカル・クライアントを指定したデータバ
ース・サーバー
- ローカル・クライアントおよびリモート・クライ
アントを指定したパーティション・データバ
ース・サーバー

パラメーター・タイプ

構成可能

デフォルト[範囲]

20 000 [-1; 32 - 112 000]

計測単位

ページ (4 KB)

このパラメーターは、開始される新しいエージェントが使用できる状態にある、未使用
のエージェント専用メモリー量の判別に使用されます。このパラメーターは、UNIX ベ
ースのプラットフォームには適用されません。

-1 の値を指定すると、このパラメーターは、*min_priv_mem* パラメーターの値を使用し
ます。

推奨: このパラメーターを設定する場合、クライアント接続/切断パターンと、同じマシン上の他の複数のプロセスのメモリー要件を考慮してください。

多くのクライアントがデータベースに同時に接続する期間が短い場合、しきい値を高くすると未使用メモリーがコミット解除されなくなり、他の処理で使用できなくなります。この場合、十分なメモリー管理が行われず、メモリーを必要とする他の処理に影響を与えます。

並行クライアント数が一定で、この値が頻繁に変動する場合にしきい値を高くすると、クライアント・プロセスでメモリーが使用できるようになり、メモリーの割り振りと割り振り解除のオーバーヘッドが低減します。

関連資料:

- 管理ガイド: パフォーマンス の『「最小コミット済み専用メモリー」構成パラメーター - min_priv_mem』

汎用レジストリー変数

表 117. 汎用レジストリー変数

変数名	オペレーティング・システム 値	
説明		
DB2ACCOUNT	すべて	デフォルト = null
リモート・ホストに送信される会計情報ストリング。詳細は、「DB2 Connect ユーザーズ・ガイド」を参照してください。		
DB2BIDI	すべて	デフォルト = NO 値: YES または NO
この変数は双方向サポートを可能にします。 DB2CODEPAGE 変数は、使用するコード・ページを宣言する場合に使用します。双方向サポートについての追加詳細は、付録『各国語サポート』を参照してください。		
DB2CODEPAGE	すべて	デフォルト: オペレーティング・システムの指定どおりに言語 ID から得られません。
データベース・クライアント・アプリケーションのために DB2 に提示されるデータのコード・ページを指定します。(設定するように) DB2 文書に明確に記述されているか、または DB2 サービスで求められない限り、DB2CODEPAGE は設定しないでください。DB2CODEPAGE にオペレーティング・システムでサポートされていない値を設定すると、その結果は予測できなくなります。通常、DB2 はコード・ページ情報を自動的にオペレーティング・システムから得るので、DB2CODEPAGE を設定する必要はありません。		
DB2DBDFT	すべて	デフォルト = null

表 117. 汎用レジストリー変数 (続き)

変数名	オペレーティング・値 システム	
説明		
<p>暗黙接続に使用するデータベースのデータベース別名を指定します。アプリケーションがデータベースに接続していないが SQL ステートメントが発行されている場合、デフォルト・データベースで DB2DBDFT 環境変数が定義されていれば、暗黙接続が行われます。</p>		
DB2DBMSADDR	Windows 32 ビット・オペレーティング・システム	デフォルト = Windows NT では 0x20000000 値: 0x20000000 ~ 0xB0000000 (増分は 0x10000)
<p>デフォルトのデータベース・マネージャー共有メモリーのアドレスを 16 進形式で指定します。共有メモリーのアドレス衝突のために db2start が失敗する場合、このレジストリー変数は強制的にデータベース・マネージャー・インスタンスに変更され、別のアドレスでその共有メモリーに割り当てられます。</p>		
DB2_DISABLE_FLUSH_LOG	すべて	デフォルト = OFF 値: ON または OFF
<p>オンライン・バックアップの完了時にアクティブ・ログ・ファイルがクローズされないようにするかどうかを指定します。</p>		
<p>オンライン・バックアップが完了した時点で、最後のアクティブ・ログ・ファイルは切り捨てられ、クローズされて、アーカイブが可能になります。これにより、オンライン・バックアップは、リカバリーに使用可能なアーカイブ・ログの完全セットとなります。</p>		
<p>ログ順序番号 (LSN) スペースの部分が無駄に使用されることを懸念している場合は、最後のアクティブ・ログがクローズされないようにすることができます。アクティブ・ログ・ファイルが切り捨てられるたびに、LSN は切り捨てられたスペースと比例した量だけ増加します。毎日膨大な数のオンライン・バックアップが実行される場合、最後のアクティブ・ログ・ファイルがクローズされないようにすることができます。</p>		
<p>オンライン・バックアップが完了したすぐあとで、ログがいったいのメッセージを受け取っていることに気付く場合、最後のアクティブ・ログ・ファイルがクローズされないようにすることができます。ログ・ファイルが切り捨てられると、予約済みアクティブ・ログ・スペースが、切り捨てられたログのサイズと比例した量だけ増加します。切り捨てられたログ・ファイルがレクラメーション処理されると、アクティブ・ログ・スペースは解放されます。レクラメーションは、ログ・ファイルが非アクティブになったすぐあとに行われます。この処理は、ログ・フル・メッセージを受け取る間の短い時間に行われます。</p>		
DB2DISCOVERYTIME	Windows オペレーティング・システム	デフォルト = 40 秒。 最低 = 20 秒
<p>SEARCH ディスカバリーが DB2 システムを探索する時間を指定します。</p>		

表 117. 汎用レジストリー変数 (続き)

変数名	オペレーティング・システム 値	
説明		
DB2GRAPHICUNICODESERVER	すべて	デフォルト = OFF 値: ON または OFF
このレジストリー変数は、グラフィック・データをユニコード・データベースに挿入するために作成された既存のアプリケーションを収容するのに使用されます。このレジストリー変数を使用するのは、sqldbchar (グラフィック) データを、クライアントのコード・ページの代わりにユニコードで送信するアプリケーションにのみ必要です。(sqldbchar は、単一の 2 バイト文字を保持できる C および C++ の SQL データ・タイプでサポートされています。)「ON」に設定すると、データベースに対して、グラフィック・データがユニコードで送信されてくることを伝え、アプリケーションはグラフィック・データをユニコードで受信することを期待します。		
DB2INCLUDE	すべて	デフォルト = 現行ディレクトリー
DB2 PREP 処理において、SQL INCLUDE テキスト・ファイル・ステートメントの処理時に使用されるパスを指定します。これによって、INCLUDE ファイルが検出されるディレクトリーのリストが提供されます。いろいろなプリコンパイル済み言語において DB2INCLUDE が使用される方法については、「アプリケーション開発ガイド」を参照してください。		
DB2_INDEX_TYPE2	すべて	省略値 = ON 値: ON または OFF
このレジストリー変数は、ユーザー表で作成された索引のタイプを制御します。これが設定されていない場合、または ON に設定されている場合、ユーザー索引はタイプ 2 として作成されます。これが OFF に設定されている場合、ユーザー索引はタイプ 1 として作成されます (この変数は、索引をまだ持っていない表での索引作成にのみ影響することに注意してください。すでに索引を持つ表に作成された索引は、既存の索引と同じタイプになります。)		
タイプ 2 索引はバージョン 8 での新規タイプで、タイプ 1 索引以上の有効な機能および並行性の利点があります。将来のリリースでは、レジストリー変数は除去され、すべての索引はタイプ 2 で作成されるようになります。		
DB2INSTDEF	Windows オペレーティング・システム	デフォルト = DB2
DB2INSTANCE が定義されていない場合に使用される値を設定します。		
DB2INSTOWNER	Windows NT	デフォルト = null
インスタンスの初回作成時に DB2 プロファイル登録で作成されるレジストリー変数。この値は、インスタンスを所有するマシンの名前に設定されます。		
DB2_LIC_STAT_SIZE	すべて	デフォルト = null 範囲: 0 ~ 32 767

表 117. 汎用レジストリー変数 (続き)

変数名	オペレーティング・システム	
説明	このレジストリー変数は、システムのライセンス統計が入っているファイルの最大サイズ (MB 単位) を決定します。値がゼロの場合、ライセンスの統計収集はオフになります。認識または定義されていない場合、この変数はデフォルト (無制限) に設定されます。統計は、ライセンス・センターを使って表示されます。	
DB2LOCALE	すべて	デフォルト: NO 値: YES または NO
DB2 を呼び出した後にデフォルトの「C」ロケールがデフォルトの「C」ロケールにリストアされるかどうか、および DB2 機能呼び出した後にプロセス・ロケールをリストアして元の「C」に戻すかどうかを指定します。元のロケールが「C」ではない場合、このレジストリー変数は無視されます。		
DB2NBDISCOVERRCVBUFS	すべて	デフォルト = 16 バッファー 最小値 = 16 バッファー
この変数は NetBIOS 検索ディスクバリーで使用されます。変数は、クライアントが同時に受信可能なディスクバリー応答数を指定します。クライアントがこの値で指定した数よりも多くの応答を同時に受け取ると、超過した応答は NetBIOS 層によって廃棄されます。デフォルトは、16 の NetBIOS 受信バッファーです。デフォルト値よりも小さい値を選択すると、デフォルトが使用されます。		
DB2OPTIONS	全 (Windows 3.1 および Macintosh を除く) デフォルト = null	
コマンド行プロセッサ・オプションを設定します。		
DB2SLOGON	Windows 3.x	デフォルト = null 値: YES または NO
DB2 (Windows 3.x 版) での保護ログオンを使用可能にします。DB2SLOGON=YES ならば、DB2 はユーザー ID とパスワードをファイルに書かず、代わりにそれらを保守するためにメモリーのセグメントを使用します。DB2SLOGON が使用可能になっていると、ユーザーは Windows 3.x を開始するたびにログオンする必要があります。		
DB2TERRITORY	すべて	デフォルト: オペレーティング・システムの指定どおりに言語 ID から得られます。
クライアント・アプリケーションの領域コード、およびテリトリー・コードを指定します。これは、日付と時刻の形式に影響します。		
DB2TIMEOUT	Windows 3.x および Macintosh デフォルト = (設定しない)	

表 117. 汎用レジストリー変数 (続き)

変数名	オペレーティング・値 システム
説明	
<p>Windows 3.x および Macintosh のクライアントの場合、長い SQL 照会時のタイムアウト時間を制御するために使用されます。このタイムアウト時間が経過すると、ダイアログ・ボックスがポップアップして、その照会を中断するか続行するか尋ねます。この変数のための最小値は 30 秒です。DB2TIMEOUT の値を 1 から 30 の間の数に設定すると、デフォルトの最小値が使用されます。DB2TIMEOUT の値を 0 または負数に設定すると、タイムアウト機能は使用不能になります。この機能は、デフォルトでは使用不能になっています。</p>	
DB2TRACENAME	<p>Windows 3.x および デフォルト = DB2WIN.TRC (Windows Macintosh 3.x の場合)、 DB2MAC.TRC (Macintosh 場合)</p>
<p>Windows 3.x および Macintosh では、トレース情報が保管されているファイルの名前を指定します。各システムのデフォルトは、現行のインスタンス・ディレクトリー (たとえば ¥sql1lib¥db2) に保管されています。トレース・ファイルを命名するときは、絶対パス名を指定することを強くお勧めします。</p>	
DB2TRACEON	<p>Windows 3.x および デフォルト = NO Macintosh 値: YES または NO</p>
<p>Windows 3.x および Macintosh では、問題が発生したときに IBM に情報を提供するためにトレースをオンにします。(自分で解決できない問題に遭遇することがないのであれば、トレースをオンにしないようにお勧めします。) トラブルシューティング情報には、クライアントでトレース機能を使用する場合の情報も含まれます。</p>	
DB2TRCFLUSH	<p>Windows 3.x および デフォルト = NO Macintosh 値: YES または NO</p>
<p>Windows 3.x および Macintosh の場合、DB2TRACFLUSH を DB2TRACEON=YES と一緒に使用することができます。DB2TRACFLUSH=YES を指定すると、各トレース・レコードが即座にトレース・ファイルに書き込まれます。この指定により、DB2 システムはかなりスローダウンするので、デフォルトの設定は DB2TRACFLUSH=NO です。これを設定しておくこと、アプリケーションによってシステムがハングし、システムをリポートする必要が起きたときに役立ちます。このキーワードの設定によって、トレース・ファイルとトレース項目がリポート時に失われないことが保証されます。</p>	
DB2TRCSYSERR	<p>Windows 3.x および デフォルト = 1 Macintosh 値: 1 ~ 32767</p>
<p>システム・エラーがいくつになるとクライアントはトレースをオフにするか、その数を指定します。デフォルト値は 1 つのシステム・エラーをトレースし、この後、トレースはオフになります。</p>	
DB2YIELD	<p>Windows 3.x デフォルト = NO 値: YES または NO</p>

表 117. 汎用レジストリー変数 (続き)

変数名	オペレーティング・値 システム
説明	
リモート・サーバーと通信している間の Windows 3.x クライアントの動作を指定します。 NO に設定すると、クライアントは CPU を他の Windows 3.x アプリケーションに譲らず、そのクライアント・アプリケーションがリモート・サーバーと通信している間は Windows 環境は一時停止となります。他のタスクを再開するには、その通信操作が完了するのを待たなければなりません。 YES に設定すると、システム機能は通常どおりです。アプリケーションを、 DB2YIELD=YES で実行することを推奨します。システムが破損される場合は、 DB2YIELD=NO に設定する必要があります。アプリケーション開発を行う場合は、通信操作が完了するのを待機している間に、アプリケーションが Windows メッセージを受け入れ、処理できるようにアプリケーションを作成する必要があります。	

関連概念:

- 管理ガイド: パフォーマンス の『DB2 レジストリー変数と環境変数』

SQL コンパイラー変数

表 118. SQL コンパイラー変数

変数名	オペレーティング・値 システム	
説明		
DB2_ANTIJOIN	すべて	デフォルト=NO (ESE 環境) デフォルト=YES (ESE 環境以外) 値: YES または NO
DB2 Universal Database ESE 環境の場合に「Yes」が指定されていると、オブティマイザーは、機会があるたびに、「NOT EXISTS」副照会を DB2 がより効率的に処理できる非結合に変換します。非 ESE 環境で「No」が指定されていると、オブティマイザーは「NOT EXISTS」副照会を非結合にほとんど変換しません。		
DB2_CORRELATED_PREDICATES	すべて	デフォルト = Yes 値: Yes または No

表 118. SQL コンパイラ変数 (続き)

変数名	オペレーティング・値 システム	
説明		
<p>この変数のデフォルトは「Yes」です。結合内の相関列上にユニーク索引があり、このレジストリー変数が「Yes」の場合、このオプティマイザーは、結合述語の相関を検出し、補正しようとします。このレジストリー変数が「Yes」の場合、オプティマイザーは相関しているケースを検出するためユニーク索引統計の KEYCARD 情報を使用し、結合された、関連する述部の選び方を動的に調整します。こうすることで、結合するサイズやコスト面で、より正確な見積もりを立てることができます。単純な等価述部の相関で調整が行われます。たとえば、C1 および C2 に索引が存在する場合の WHERE C1=5 AND C2=10 など。索引は固有である必要はないが、等価述部列は索引内のすべての列を網羅していなければなりません。</p>		
DB2_HASH_JOIN	すべて	デフォルト = YES 値: YES または NO
アクセス・プランのコンパイル時に可能な結合方法としてハッシュ結合を指定します。		
DB2_INLIST_TO_NLJN	すべて	デフォルト = NO 値: YES または NO
<p>状況によっては、SQL コンパイラーは IN リスト述部を結合に書き換えることができます。たとえば、以下のような照会は、</p> <pre data-bbox="142 927 595 1005">SELECT * FROM EMPLOYEE WHERE DEPTNO IN ('D11', 'D21', 'E21')</pre> <p>以下のように書くことができます。</p> <pre data-bbox="142 1078 774 1156">SELECT * FROM EMPLOYEE, (VALUES 'D11', 'D21', 'E21) AS V(DNO) WHERE DEPTNO = V.DNO</pre> <p>DEPTNO に索引がある場合、この改訂はより良いパフォーマンスを提供することがあります。値のリストが最初にアクセスされて、結合述部に適用する索引を使用してネストされたループ結合のある EMPLOYEE に結合されます。</p> <p>場合によっては、オプティマイザーが照会の再書き込みバージョンに最適な結合メソッドを判別するための正確な情報を持っていないことがあります。IN リストにオプティマイザーがカタログ統計を使用して選択性を判別することを禁止するパラメーター・マーカーまたはホスト変数が含まれる場合に、このことが生じることがあります。このレジストリー変数はオプティマイザーが値のリストを結合するために、IN リストを結合内の内部表として与える表を使用して、ネストされたループ結合を優先的に使用するようにします。</p>		
DB2_LIKE_VARCHAR	すべて	デフォルト =Y,Y

表 118. SQL コンパイラー変数 (続き)

変数名	オペレーティング・値 システム	
説明		
サブエレメント統計の使用を制御します。これらの統計は、データにブランクで区切られた一連のサブフィールドまたはサブエレメント形式の構造がある場合、列内のデータ内容に関する統計です。サブエレメント統計の収集はオプションで、RUNSTATS コマンドまたは API 内のオプションによって制御されます。		
このレジストリー変数は、以下の形式の述部をオプティマイザーが処理する方法に影響します。		
<pre>COLUMN LIKE '%xxxxxx%'</pre>		
xxxxxx は文字のストリングです。		
このレジストリー変数の使用方法を示す構文は以下のとおりです。		
<pre>db2set DB2_LIKE_VARCHAR=[Y N S num1] [,Y N S num2]</pre>		
説明		
<ul style="list-style-type: none"> • コマの前の用語、または述部の右にある唯一の用語は、2 番目の用語が N に指定されているか、または列に正の値のサブエレメント統計がない場合にのみ、以下の意味になります。 <ul style="list-style-type: none"> - S - オプティマイザーは % 文字で囲まれたストリングの長さに基づいて、列を形成するために連結する一連のエレメントの各エレメントの長さを見積もる。 - Y - デフォルト。アルゴリズム・パラメーターのデフォルト値 1.9 を使用する。アルゴリズム・パラメーターで可変長サブエレメント・アルゴリズムを使用する。 - N - 固定長サブエレメント・アルゴリズムを使用する。 - num1 - 可変長サブエレメント・アルゴリズムにより、アルゴリズム・パラメーターとして num1 の値を使用する。 • コマの後の用語は以下のような意味がありますが、正の値のサブエレメント統計を持つ列に対してのみです。 <ul style="list-style-type: none"> - N - サブエレメント統計を使用しない。最初の用語が有効になります。 - Y - デフォルト。正の値のサブエレメント統計を持つ列の場合に、アルゴリズム・パラメーターのデフォルト値 1.9 と一緒にサブエレメント統計を使用する可変長サブエレメント・アルゴリズムを使用する。 - num2 - 正の値のサブエレメント統計を持つ列の場合に、アルゴリズム・パラメーターとして num2 の値と一緒にサブエレメント統計を使用する可変長サブエレメント・アルゴリズムを使用する。 		
DB2_MINIMIZE_LIST_PREFETCH	すべて	デフォルト = NO
		値: YES または NO

表 118. SQL コンパイラ変数 (続き)

変数名	オペレーティング・ 値 システム	
説明		
<p>リスト・プリフェッチは、修飾するための RID を索引から検索して、それらをページ番号でソートしてからデータ・ページをプリフェッチすることを必要とする、特殊な表アクセス方式です。場合によっては、オプティマイザーがリスト・プリフェッチが良いアクセス方式であるかどうかを判別するための正確な情報を持っていないことがあります。述部選択性にオプティマイザーがカタログ統計を使用いて選択性を判別することを禁止するパラメーター・マーカーまたはホスト変数が含まれる場合に、このことが生じることがあります。</p> <p>このレジストリー変数は、そのような状況でオプティマイザーがリスト・プリフェッチを検討することを禁止します。</p>		
DB2_SELECTIVITY	ALL	デフォルト = No 値: Yes または No
<p>このレジストリー変数は、SQL ステートメント内での検索条件に SELECTIVITY 文節を使用できるかどうかを制御します。</p> <p>このレジストリー変数が「Yes」に設定された場合、以下の述部に SELECTIVITY 文節を設定可能です。</p> <ul style="list-style-type: none"> • 少なくとも 1 つの式がホスト変数を含む基本述部 • MATCH 式、述部式、またはエスケープ式にホスト変数が含まれる LIKE 述部 		
DB2_NEW_CORR_SQ_FF	すべて	デフォルト = OFF 値: ON または OFF
<p>「ON」に設定すると、SQL オプティマイザーが特定の副照会述部について計算した選択値に影響します。このパラメーターを使用すると、副照会の SELECT リストで MIN または MAX 総計関数を使用する等価副照会述部の選択値の正確度を高めることができます。例:</p> <pre>SELECT * FROM T WHERE T.COL = (SELECT MIN(T.COL) FROM T WHERE ...)</pre>		
DB2_PRED_FACTORIZE	すべて	デフォルト = NO 値: YES または NO

表 118. SQL コンパイラー変数 (続き)

変数名	オペレーティング・値 システム	
説明		
<p>オブティマイザーが、論理和から追加の述部を取り出す機会を探るかどうかを指定します。状況によっては、追加の述部は中間の推定カーディナリティー、または結果セットを変更できます。以下の照会で、</p>		
<pre>SELECT n1.empno, n1.lastname FROM employee n1, employee n2 WHERE ((n1.lastname='SMITH' AND n2.lastname='JONES') OR (n1.lastname='JONES' AND n2.lastname='SMITH'))</pre>		
<p>オブティマイザーは、以下の追加述部を生成できます。</p>		
<pre>SELECT n1.empno, n1.lastname FROM employee n1, employee n2 WHERE n1.lastname IN ('SMITH', 'JONES') AND n2.lastname IN ('SMITH', 'JONES') AND ((n1.lastname='SMITH' AND n2.lastname='JONES') OR (n1.lastname='JONES' AND n2.lastname='SMITH'))</pre>		
DB2_REDUCED_OPTIMIZATION	すべて	デフォルト = NO
<p>値: NO、YES、任意の整数、DISABLE</p>		

表 118. SQL コンパイラ変数 (続き)

変数名	オペレーティング・値 システム
説明	
<p>このレジストリー変数によって、最適化機能を削減したり、最適化機能を指定した最適化レベルに固定して使用するよう要求することができます。使用される最適化手法の数を削減する場合、最適化の際に使用される時間およびリソースも削減されます。</p>	
<p>注: 最適化に使用される時間およびリソースは削減されることがあるが、最適ではないデータ・アクセス・プランが生成されるリスクは増加します。このレジストリー変数は、IBM またはそのパートナーから指示された場合にのみ使用します。</p>	
<ul style="list-style-type: none"> • NO に設定した場合 オプティマイザーは最適化手法を変更しません。 	
<ul style="list-style-type: none"> • YES に設定した場合 最適化レベルが 5 (デフォルト) 以下の場合、相当量の準備時間とリソースを消費するものの通常はより良いアクセス・プランを生成することのないいくつかの最適化手法をオプティマイザーは使用不可にします。 	
<p>最適化レベルがちょうど 5 の場合、いくつかのその他の手法を縮小または使用不可にします。その結果、オプティマイザーによる最適化に必要な時間とリソースがさらに削減されることがあるが、同時に最適ではないアクセス・プランが生成されるリスクがさらに増大します。最適化レベルが 5 より低い場合、これらの技法のいくつかは最初から無効であることがあります。しかしそれらが有効であれば、有効のままとなります。</p>	
<ul style="list-style-type: none"> • 任意の整数に設定した場合 YES と同じ効果があり、さらにレベル 5 で最適化された動的に準備された照会のために以下の追加の動作が伴います。いずれかの照会ブロック内にある結合の合計数が設定値を超える場合、上記のレベル 5 最適化レベルについての説明で示したような、追加の最適化手法を使用不可にする代わりに、オプティマイザーは貪欲型結合エミュレーションに切り替えます。これは、照会が最適化レベル 2 に類似したレベルで最適化されることを暗黙に示します。 	
<ul style="list-style-type: none"> • DISABLE に設定した場合 この DB2_REDUCED_OPTIMIZATION 変数によって拘束されないときのオプティマイザーの動作は、最適化レベル 5 での動的照会の最適化を動的に削減するものとなる場合があります。この設定値はこの動作を使用不可にして、オプティマイザーがレベル 5 の最適化を完全に実行することを要求します。 	
<p>最適化レベル 5 での動的な最適化の削減は、DB2_REDUCED_OPTIMIZATION を YES に設定したときの最適化レベルがちょうど 5 の場合について説明された動作、および整数の設定値について説明された動作よりも優先されることに注意してください。</p>	

関連概念:

- 管理ガイド: パフォーマンス の『最適化クラスのガイドライン』

- 管理ガイド: パフォーマンス の『最適の結合を選択する方法』
- 管理ガイド: パフォーマンス の『DB2 レジストリー変数と環境変数』

関連資料:

- 管理ガイド: パフォーマンス の『最適化クラス』

ストレージ管理ツール用のストアード・プロシージャ

以下の表には、ストレージ管理ツール用に作成されるストアード・プロシージャ機能が表示されています。ストアード・プロシージャは、データベースが作成された時に、自動的に作成されます。またそれぞれのパッケージは、必要に応じてバインドされます。

表 119. ストレージ管理ツール用のストアード・プロシージャ

完全修飾名	パラメーター	機能
SYSPROC.CREATE_STORAGEMGMT _TABLES	in_tbspace VARCHAR(128) input - 表スペース名	すべてのストレージ管理表を input で指定した表スペース内の固定の「SYSTOOLS」スキーマに作成します。
SYSPROC.DROP_STORAGEMGMT _TABLES	dropSpec SMALLINT input - 0 / 1	すべてのストレージ管理表のドロップを試みます。dropSpec=0 の場合、エラーが検出されると処理は停止します。dropSpec=1 の場合、検出されたエラーは無視して、処理を続行します。
SYSPROC.CAPTURE_STORAGEMGMT _INFO	in_rootType SMALLINT input - 有効なすべての値は STMG_OBJECT_TYPE 表に指定 in_rootSchema VARCHAR(128) input - ストレージ・スナップショット・ルート・オブジェクトのスキーマ名 in_rootName VARCHAR(128) input- ルート・オブジェクトの名前	システム・カタログおよびスナップショットのために、指定されたルート・オブジェクトのストレージ関連情報、およびその有効範囲内に定義されたストレージ・オブジェクトの収集を試みます。すべてのストレージ・オブジェクトは、STMG_OBJECT_TYPE 表に指定されています。

関連資料:

- 305 ページの『ストレージ管理ビュー』

ストレージ管理ビュー表

STMG_OBJECT_TYPE 表:

STMG_OBJECT_TYPE 表には、モニター可能なサポートされたストレージ・タイプごとに、1 つの行が含まれています。

表 120. STMG_OBJECT_TYPE 表

列名	データ・タイプ	NULL 可能	説明
OBJ_TYPE	INTEGER	N	ストレージ・オブジェクトのタイプに対応する整数値
TYPE_NAME	VARCHAR	N	ストレージ・オブジェクト・タイプの記述名

STMG_THRESHOLD_REGISTRY 表:

STMG_THRESHOLD_REGISTRY 表には、ストレージしきい値タイプごとに 1 つの行が含まれています。使用可能になったしきい値は、ストレージ・スナップショットが取られたときに、分析プロセスによって使用されます。

表 121. STMG_THRESHOLD_REGISTRY 表

列名	データ・タイプ	NULL 可能	説明
STMG_TH_TYPE	INTEGER	N	ストレージしきい値タイプに対応する整数値
ENABLED	CHARACTER	N	Y = しきい値は使用可能 N = しきい値は使用可能でないため、ストレージ分析中に比較されない
STMG_TH_NAME	VARCHAR	Y	ストレージしきい値の記述名

STMG_CURR_THRESHOLD 表:

STMG_CURR_THRESHOLD 表には、ストレージ・オブジェクトに対して明示的に設定されたしきい値タイプごとに、1 つの行が含まれています。

表 122. STMG_CURR_THRESHOLD 表

列名	データ・タイプ	NULL 可能	説明
STMG_TH_TYPE	INTEGER	N	ストレージしきい値タイプに対応する整数値
OBJ_TYPE	INTEGER	N	ストレージ・オブジェクトのタイプに対応する整数値
OBJ_NAME	VARCHAR	N	ストレージ・オブジェクトの名前

表 122. STMG_CURR_THRESHOLD 表 (続き)

列名	データ・タイプ	NULL 可能	説明
OBJ_SCHEMA	VARCHAR	N	ストレージ・オブジェクトのスキーマ。 スキーマがオブジェクトに適用できない場合、「-」が使用される
WARNING_THRESHOLD	SMALLINT	Y	ストレージ・オブジェクトに設定された警告しきい値の値
ALARM_THRESHOLD	SMALLINT	Y	ストレージ・オブジェクトに設定されたアラームしきい値の値

STMG_ROOT_OBJECT 表:

STMG_ROOT_OBJECT 表には、ストレージ・スナップショットごとのルート・オブジェクトに、1 つの行が含まれています。

表 123. STMG_ROOT_OBJECT 表

列名	データ・タイプ	NULL 可能	説明
STMG_TIMESTAMP	TIMESTAMP	N	ストレージ・スナップショットのタイム・スタンプ。これはデータ・キャプチャー・プロセスが開始された時間を示します。
OBJ_TYPE	INTEGER	N	ストレージ・オブジェクトのタイプに対応する整数値
ROOT_ID	VARCHAR	N	ルート・オブジェクトのID

STMG_OBJECT 表:

STMG_OBJECT 表には、ストレージ・スナップショットの取得によって分析されるストレージ・オブジェクトごとに、1 つの行があります。

表 124. STMG_OBJECT 表

列名	データ・タイプ	NULL 可能	説明
STMG_TIMESTAMP	TIMESTAMP	N	ストレージ・スナップショットのタイム・スタンプ。これはデータ・キャプチャー・プロセスが開始された時間を示します。
ROOT_ID	CHARACTER	N	ルート・オブジェクトのID

表 124. STMG_OBJECT 表 (続き)

列名	データ・タイプ	NULL 可能	説明
OBJ_TYPE	INTEGER	N	ストレージ・オブジェクトのタイプに対応する整数値
OBJ_SCHEMA	VARCHAR	N	ストレージ・オブジェクトのスキーマ。 スキーマがオブジェクトに適用できない場合、「-」が使用される
OBJ_NAME	VARCHAR	N	ストレージ・オブジェクトの名前
DBPG_NAME	VARCHAR	Y	オブジェクトが常駐するデータベース・パーティション・グループの名前。該当しない場合は NULL
TS_NAME	VARCHAR	Y	オブジェクトが常駐する表スペースの名前。該当しない場合は NULL
OBJ_ID	VARCHAR	N	指定されたストレージ・スナップショット・タイム・スタンプの下のストレージ・オブジェクトごとの固有 ID

STMG_HIST_THRESHOLD 表:

STMG_HIST_THRESHOLD 表には、ストレージ・スナップショットが取られるときに、ストレージ・オブジェクトを分析するために使用される、それぞれのしきい値ごとに 1 つの行を含みます。

表 125. STMG_HIST_THRESHOLD 表

列名	データ・タイプ	NULL 可能	説明
STMG_TIMESTAMP	TIMESTAMP	N	ストレージ・スナップショットのタイム・スタンプ。これはデータ・キャプチャー・プロセスが開始された時間を示します。
STMG_TH_TYPE	INTEGER	N	ストレージしきい値タイプに対応する整数値
OBJ_ID	VARCHAR	N	指定されたストレージ・スナップショット・タイム・スタンプの下のストレージ・オブジェクトごとの固有 ID

表 125. STMG_HIST_THRESHOLD 表 (続き)

列名	データ・タイプ	NULL 可能	説明
WARNING_THRESHOLD	SMALLINT	Y	ストレージ・スナップショットが 取られるときにストレージ・オブ ジェクトに設定された警告しきい 値の値。
ALARM_THRESHOLD	SMALLINT	Y	ストレージ・スナップショットが 取られるときにストレージ・オブ ジェクトに設定されたアラームし きい値の値

STMG_DATABASE 表:

STMG_DATABASE 表には、データベース・ストレージ・スナップショットのそれぞれの
詳細項目ごとに、1 つの行が含まれています。

表 126. STMG_DATABASE 表

列名	データ・タイプ	NULL 可能	説明
STMG_TIMESTAMP	TIMESTAMP	N	ストレージ・スナップショットの タイム・スタンプ。これはデー タ・キャプチャー・プロセスが開 始された時間を示します。
OBJ_ID	VARCHAR	N	指定されたストレージ・スナップ ショット・タイム・スタンプの下 のストレージ・オブジェクトごと の固有 ID
COMPLETE_TIMESTAMP	TIMESTAMP	Y	OBJ_ID 列によって識別されたデ ータベースに対して、データ・キ ャプチャー・プロセスが完了した ときのタイム・スタンプ。
SPACE_THRESHOLD_EXCEEDED	SMALLINT	Y	データベースのストレージ・ス ペース使用状態を示すフラグ。 0 = 正常作動状態 1 = しきい値超 過警告 2 = しきい値超過アラーム
SKEW_THRESHOLD_EXCEEDED	SMALLINT	Y	データベースのデータ配分状態を 示すフラグ。 0 = 正常作動状態 1 = しきい値超 過警告 2 = しきい値超過アラーム

表 126. STMG_DATABASE 表 (続き)

列名	データ・タイプ	NULL 可能	説明
CR_THRESHOLD_EXCEEDED	SMALLINT	Y	データベースの索引クラスタリング状態を示すフラグ。 0 = 正常作動状態 1 = しきい値超過警告 2 = しきい値超過アラーム

STMG_DBPGROUP 表:

STMG_DBPGROUP 表には、データベース・パーティション・グループ・ストレージ・スナップショットのそれぞれの詳細項目ごとに、1 つの行が含まれています。

表 127. STMG_DBPGROUP 表

列名	データ・タイプ	NULL 可能	説明
STMG_TIMESTAMP	TIMESTAMP	N	ストレージ・スナップショットのタイム・スタンプ。これはデータ・キャプチャー・プロセスが開始された時間を示します。
OBJ_ID	VARCHAR	N	指定されたストレージ・スナップショット・タイム・スタンプの下のストレージ・オブジェクトごとの固有 ID
COMPLETE_TIMESTAMP	TIMESTAMP	Y	OBJ_ID 列によって識別されたデータベース・パーティション・グループに対して、データ・キャプチャー・プロセスが完了したときのタイム・スタンプ。
SPACE_THRESHOLD_EXCEEDED	SMALLINT	Y	データベース・パーティション・グループのストレージ・スペース使用状態を示すフラグ。 0 = 正常作動状態 1 = しきい値超過警告 2 = しきい値超過アラーム
SKEW_THRESHOLD_EXCEEDED	SMALLINT	Y	データベース・パーティション・グループのデータ配分状態を示すフラグ。 0 = 正常作動状態 1 = しきい値超過警告 2 = しきい値超過アラーム

表 127. STMG_DBPGROUP 表 (続き)

列名	データ・タイプ	NULL 可能	説明
PARTITON_COUNT	SMALLINT	Y	データベース・パーティション・グループに含まれているパーティションの数。
TARGET_LEVEL	BIGINT	Y	データベース・パーティション・グループに含まれている、すべてのパーティションにわたる平均データ・サイズ (バイト単位) これは、均一なデータ配分のターゲット・レベルです。
DATA_SKEW	SMALLINT	Y	すべてのパーティション間の TARGET_LEVEL からの最大データ・サイズ偏差のパーセンテージ。この値は、データ・キャプチャおよび分析プロセス中に、STMG_CURR_THRESHOLD 表内のデータベース・パーティション・グループに設定されたデータ配分スキューと比較するために使用されます。
TOTAL_SIZE	BIGINT	Y	データベース・パーティション・グループに含まれている、すべてのパーティションにわたる合計サイズ (バイト単位) これは、データベース・パーティション・グループに定義されたすべての表スペースの合計サイズ (ページ・サイズで乗算されたページの数) の和です。DMS 表スペースでは、合計サイズは割り振られたサイズです。SMS 表スペースでは、合計サイズは現在表スペースによって使用されているサイズです。

表 127. STMG_DBPGROUP 表 (続き)

列名	データ・タイプ	NULL 可能	説明
DATA_SIZE	BIGINT	Y	データベース・パーティション・グループに含まれている、すべてのパーティションにわたるデータ・サイズ (バイト単位) これは、データベース・パーティション・グループに定義されたすべての表スペースのデータ・サイズ (ページ・サイズで乗算されたデータ・ページの数) の和です。
PERCENT_USED	SMALLINT	Y	合計サイズを超えるデータ・サイズのパーセンテージ値です。この値は、データ・キャプチャーおよび分析プロセス中に、スペース使用しきい値と比較されます。SMS 表スペースの場合、表スペースまたはその親データベース・パーティション・グループのスペース使用しきい値は、不必要なアラームを避けるために、100 に設定する必要があります。

STMG_DBPARTITION 表:

STMG_DBPARTITION 表には、データベース・パーティション・ストレージ・スナップショットのそれぞれの詳細項目ごとに、1 つの行が含まれています。これは STMG_DBPGROUP 表と一緒に使用されることを意図しています。

表 128. STMG_DBPARTITION 表

列名	データ・タイプ	NULL 可能	説明
STMG_TIMESTAMP	TIMESTAMP	N	ストレージ・スナップショットのタイム・スタンプ。これはデータ・キャプチャー・プロセスが開始された時間を示します。
OBJ_ID	VARCHAR	N	指定されたストレージ・スナップショット・タイム・スタンプの下のストレージ・オブジェクトごとの固有 ID

表 128. STMG_DBPARTITION 表 (続き)

列名	データ・タイプ	NULL 可能	説明
PARTITION_NUM	INTEGER	Y	データベース・パーティション番号。
COMPLETE_TIMESTAMP	TIMESTAMP	Y	OBJ_ID 列によって識別されたデータベース・パーティションに対して、データ・キャプチャー・プロセスが完了したときのタイム・スタンプ。
DBPG_NAME	CHARACTER	Y	データベース・パーティション・グループの名前
IN_USE	CHARACTER	Y	ストレージ・スナップショットを取ったときのパーティションの状況。SYSCAT.NODEGROUPDEF 内の IN_USE 列と同様。
HOST_NAME	VARCHAR	Y	データベース・パーティションのホスト名。
HOST_SYSTEM_SIZE	BIGINT	Y	使用不可
EST_DATA_SIZE	BIGINT	Y	データベース・パーティション・グループ有効範囲内のデータベース・パーティションの見積もりデータ・サイズ。この値は、指定されたパーティションの表パーティション・データ・サイズの合計として計算されます。

STMG_TABLESPACE 表:

STMG_TABLESPACE 表には、表スペース・ストレージ・スナップショットのそれぞれの詳細項目ごとに、1 つの行が含まれています。

表 129. STMG_TABLESPACE 表

列名	データ・タイプ	NULL 可能	説明
STMG_TIMESTAMP	TIMESTAMP	N	ストレージ・スナップショットのタイム・スタンプ。これはデータ・キャプチャー・プロセスが開始された時間を示します。

表 129. STMG_TABLESPACE 表 (続き)

列名	データ・タイプ	NULL 可能	説明
OBJ_ID	VARCHAR	N	指定されたストレージ・スナップショット・タイム・スタンプの下のストレージ・オブジェクトごとの固有 ID
COMPLETE_TIMESTAMP	TIMESTAMP	Y	OBJ_ID 列によって識別された表スペースに対して、データ・キャプチャー・プロセスが完了したときのタイム・スタンプ。
SPACE_THRESHOLD_EXCEEDED	SMALLINT	Y	表スペースのストレージ・スペース使用状態を示すフラグ。 0 = 正常作動状態 1 = しきい値超過警告 2 = しきい値超過アラーム
TYPE	CHARACTER	Y	SYSCAT.TABLESPACES に定義されているのと同じ
DATATYPE	CHARACTER	Y	SYSCAT.TABLESPACES に定義されているのと同じ
TOTAL_SIZE	BIGINT	Y	SYSCAT.TABLESPACES に定義されているのと同じ
PERCENT_USED	SMALLINT	Y	SYSCAT.TABLESPACES に定義されているのと同じ。この値は、データ・キャプチャーおよび分析プロセス中に、 STMG_CURR_THRESHOLD 表内のスペース使用しきい値と比較するために使用されます。
DATA_SIZE	BIGINT	Y	SYSCAT.TABLESPACES に定義されているのと同じ
DATA_PAGE	BIGINT	Y	SYSCAT.TABLESPACES に定義されているのと同じ
EXTENT_SIZE	INTEGER	Y	SYSCAT.TABLESPACES に定義されているのと同じ
PREFETCH_SIZE	INTEGER	Y	SYSCAT.TABLESPACES に定義されているのと同じ
OVERHEAD	DOUBLE	Y	SYSCAT.TABLESPACES に定義されているのと同じ

表 129. STMG_TABLESPACE 表 (続き)

列名	データ・タイプ	NULL 可能	説明
TRANSFER_RATE	DOUBLE	Y	SYSCAT.TABLESPACES に定義されているのと同じ
BUFFERPOOL_ID	INTEGER	Y	SYSCAT.TABLESPACES に定義されているのと同じ
PAGE_SIZE	INTEGER	Y	SYSCAT.TABLESPACES に定義されているのと同じ

STMG_CONTAINER 表:

STMG_CONTAINER 表には、コンテナ・ストレージ・スナップショットのそれぞれの詳細項目ごとに、1 つの行が含まれています。

表 130. STMG_CONTAINER 表

列名	データ・タイプ	NULL 可能	説明
STMG_TIMESTAMP	TIMESTAMP	N	ストレージ・スナップショットのタイム・スタンプ。これはデータ・キャプチャー・プロセスが開始された時間を示します。
OBJ_ID	VARCHAR	N	指定されたストレージ・スナップショット・タイム・スタンプの下のストレージ・オブジェクトごとの固有 ID
COMPLETE_TIMESTAMP	TIMESTAMP	Y	OBJ_ID 列によって識別されたコンテナに対して、データ・キャプチャー・プロセスが完了したときのタイム・スタンプ
TS_ID	INTEGER	Y	コンテナが割り当てられる表スペースの整数 ID
SPACE_THRESHOLD_EXCEEDED	SMALLINT	Y	コンテナのストレージ・スペース使用状態を示すフラグ。 0 = 正常作動状態 1 = しきい値超過警告 2 = しきい値超過アラーム
PARTITION_NUM	INTEGER	Y	コンテナが常駐するデータベース・パーティションのパーティション番号。

表 130. STMG_CONTAINER 表 (続き)

列名	データ・タイプ	NULL 可能	説明
TYPE	CHARACTER	Y	'P' = パス・コンテナ、'F' = ファイル・コンテナ、'D' = ロー・デバイス・コンテナ
TOTAL_PAGE	BIGINT	Y	コンテナに割り振られた合計ページ
USABLE_PAGES	BIGINT	Y	コンテナ内の使用可能なページの数
PERCENT_USED	SMALLINT	Y	使用不可。この値は、データ・キャプチャーおよび分析プロセス中に、STMG_CURR_THRESHOLD 表内のスペース使用しきい値と比較するために使用されます。
DATA_SIZE	BIGINT	Y	NOT AVAILABLE
DATA_PAGE	BIGINT	Y	NOT AVAILABLE

STMG_TABLE 表:

STMG_TABLE 表には、表ストレージ・スナップショットのそれぞれの詳細項目ごとに、1 つの行が含まれています。

表 131. STMG_TABLE 表

列名	データ・タイプ	NULL 可能	説明
STMG_TIMESTAMP	TIMESTAMP	N	ストレージ・スナップショットのタイム・スタンプ。これはデータ・キャプチャー・プロセスが開始された時間を示します。
OBJ_ID	VARCHAR	N	指定されたストレージ・スナップショット・タイム・スタンプの下のストレージ・オブジェクトごとの固有 ID
COMPLETE_TIMESTAMP	TIMESTAMP	Y	OBJ_ID 列によって識別された表に対して、データ・キャプチャー・プロセスが完了したときのタイム・スタンプ。
DBPG_NAME	VARCHAR	Y	表が常駐するデータベース・パーティション・グループの名前

表 131. STMG_TABLE 表 (続き)

列名	データ・タイプ	NULL 可能	説明
SKEW_THRESHOLD_EXCEEDED	SMALLINT	Y	表のデータ配分状態を示すフラグ。 0 = 正常作動状態 1 = しきい値超過警告 2 = しきい値超過アラーム
TOTAL_ROW_COUNT	BIGINT	Y	表の合計行数
AVG_ROW_COUNT	BIGINT	Y	すべての表パーティションにわたった平均行数
TARGET_LEVEL	BIGINT	Y	各パーティション上の平均データ・サイズ (バイト単位)
DATA_SKEW	SMALLINT	Y	指定された表の、すべての表パーティションにわたった TARGET_LEVEL からはずれた ROW_COUNT 値の最大パーセンテージ。この値は、データ・キャプチャーおよび分析プロセス中に、STMG_CURR_THRESHOLD 表内のデータ・スキューしきい値と比較するために使用されます。
AVG_ROW_LENGTH	BIGINT	Y	表の平均の行の長さ。この統計が収集されると、この表内のすべての列の平均列長の和になります。統計データがない場合、この値は変数列長のパーセンテージに、固定列長を加算して計算されます。
COLCOUNT	INTEGER	Y	SYSCAT.TABLES に定義されているのと同じ
ESTIMATED_SIZE	BIGINT	Y	SYSCAT.TABLES に定義されているのと同じ
NPAGES	INTEGER	Y	SYSCAT.TABLES に定義されているのと同じ
FPAGES	INTEGER	Y	SYSCAT.TABLES に定義されているのと同じ
OVERFLOW	INTEGER	Y	SYSCAT.TABLES に定義されているのと同じ
MAIN_TBSPACE	VARCHAR	Y	SYSCAT.TABLES に定義されているのと同じ

表 131. STMG_TABLE 表 (続き)

列名	データ・タイプ	NULL 可能	説明
INDEX_TBSPACE	VARCHAR	Y	SYSCAT.TABLES に定義されているのと同じ
LONG_TBSPACE	VARCHAR	Y	SYSCAT.TABLES に定義されているのと同じ

STMG_TBPARTITION 表:

STMG_TBPARTITION 表には、表パーティション・ストレージ・スナップショットのそれぞれの詳細項目ごとに、1 つの行が含まれています。

表 132. STMG_TBPARTITION 表

列名	データ・タイプ	NULL 可能	説明
STMG_TIMESTAMP	TIMESTAMP	N	ストレージ・スナップショットのタイム・スタンプ。これはデータ・キャプチャー・プロセスが開始された時間を示します。
OBJ_ID	VARCHAR	N	指定されたストレージ・スナップショット・タイム・スタンプの下のストレージ・オブジェクトごとの固有 ID
PARTITION_NUM	INTEGER	N	表パーティションが常駐するデータベース・パーティションのパーティション番号
COMPLETE_TIMESTAMP	TIMESTAMP	Y	OBJ_ID 列によって識別された表パーティションに対して、データ・キャプチャー・プロセスが完了したときのタイム・スタンプ
DBPG_NAME	VARCHAR	Y	表が常駐する場所のデータベース・パーティション・グループの名前
ROWCOUNT	BIGINT	Y	この表パーティション内の行数

STMG_INDEX 表:

STMG_INDEX 表には、索引ストレージ・スナップショットのそれぞれの詳細項目ごとに、1 つの行が含まれています。

表 133. STMG_INDEX 表

列名	データ・タイプ	NULL 可能	説明
STMG_TIMESTAMP	TIMESTAMP	N	ストレージ・スナップショットのタイム・スタンプ。これはデータ・キャプチャー・プロセスが開始された時間を示します。
OBJ_ID	VARCHAR	N	指定されたストレージ・スナップショット・タイム・スタンプの下のストレージ・オブジェクトごとの固有 ID
COMPLETE_TIMESTAMP	TIMESTAMP	Y	OBJ_ID 列によって識別された索引に対して、データ・キャプチャー・プロセスが完了したときのタイム・スタンプ
DBPG_NAME	VARCHAR	Y	索引が常駐する場所のデータベース・パーティション・グループの名前
TB_SCHEMA	VARCHAR	Y	SYSCAT.INDEXES に定義されている TABNAME と同じ
TB_NAME	VARCHAR	Y	SYSCAT.INDEXES に定義されている TABSCHEMA と同じ
CR_THRESHOLD_EXCEEDED	SMALLINT	Y	索引クラスタリング状態を示すフラグ。 0 = 正常作動状態 1 = しきい値超過警告 2 = しきい値超過アラーム
COLCOUNT	INTEGER	Y	SYSCAT.INDEXES に定義されているのと同じ
ESTIMATED_SIZE	BIGINT	Y	SYSCAT.INDEXES に定義されているのと同じ
NLEAF	INTEGER	Y	SYSCAT.INDEXES に定義されているのと同じ
NLEVELS	SMALLINT	Y	SYSCAT.INDEXES に定義されているのと同じ
FIRSTKEYCARD	BIGINT	Y	SYSCAT.INDEXES に定義されているのと同じ
FIRST2KEYCARD	BIGINT	Y	SYSCAT.INDEXES に定義されているのと同じ

表 133. STMG_INDEX 表 (続き)

列名	データ・タイプ	NULL 可能	説明
FIRST3KEYCARD	BIGINT	Y	SYSCAT.INDEXES に定義されているのと同じ
FIRST4KEYCARD	BIGINT	Y	SYSCAT.INDEXES に定義されているのと同じ
FULLKEYCARD	BIGINT	Y	SYSCAT.INDEXES に定義されているのと同じ
CLUSTERRATIO	SMALLINT	Y	SYSCAT.INDEXES に定義されているように、この値は、データ・キャプチャーおよび分析プロセス中に、指定された索引のしきい値セットと比較するために使用されます。
CLUSTERFACTOR	BIGINT	Y	SYSCAT.INDEXES に定義されているのと同じ
SEQUENTIAL_PAGES	INTEGER	Y	SYSCAT.INDEXES に定義されているのと同じ
DENSITY	INTEGER	Y	SYSCAT.INDEXES に定義されているのと同じ

関連資料:

- 305 ページの『ストレージ管理ビュー』

サテライト管理 ガイドおよびリファレンス

DB2 サーバーのインストールに必要なユーザー・アカウント (Windows)

Windows NT、Windows 2000、Windows XP、または Windows .NET にインストールする場合には、1 つのインストール・ユーザー・アカウントと、セットアップ用に 1 つ以上のユーザー・アカウントが必要です。インストール・ユーザー・アカウントは、DB2 セットアップ・ウィザードの実行に先立って定義する必要があります。セットアップ・ユーザー・アカウントは、インストールの前に定義することもできますし、DB2 セットアップ・プログラムに作成させることもできます。

すべてのユーザー・アカウント名は、ご使用のシステムの命名規則と DB2 命名規則に従ったものでなければなりません。

DB2 サーバー・ユーザー・アカウント:

インストール・ユーザー・アカウント

ローカルまたはドメイン・ユーザー・アカウントは、インストールを実行するために必要です。ユーザー・アカウントは、インストールを実行するマシンの管理者 グループに属していなければなりません。

DB2 Administration Server のユーザー・アカウント

ローカルまたはドメイン・ユーザー・アカウントは、DB2 Administration Server (DAS) に必要です。DAS ユーザー・アカウントは、DB2 をインストールする前に作成することもできますし、DB2 セットアップ・ウィザードに作成させることもできます。DB2 セットアップ・ウィザードに新規ドメイン・ユーザー・アカウントを作成させたい場合には、インストールを実行するために使用するユーザー・アカウントが、ドメイン・ユーザー・アカウントを作成する権限を持っている必要があります。ユーザー・アカウントは、インストールを実行するマシンの管理者 グループに属していなければなりません。このアカウントは、以下のユーザー権限を付与されます。

- オペレーティング・システムの一部として機能
- トークン・オブジェクトの作成
- サービスとしてログオン
- クォータの増加
- プロセス・レベル・トークンの置き換え

DB2 Administration Server (DAS) は、GUI ツールをサポートするために使用される特殊 DB2 管理サービスで、ローカルおよびリモート DB2 サーバー上の管理タスクを援助します。DAS には、DAS サービスの開始時にコンピューターに DAS サービスをログオンするために使用される、ユーザー・アカウントが割り当てられています。ご使用の環境内のそれぞれの DB2 システム上の DAS ユーザーに、SYSADM 権限を与えることをお勧めします。そうすれば、必要であれば、それが他のインスタンスを開始したり停止したりすることができます。デフォルトでは、管理者 グループに参加しているユーザーには SYSADM 権限があります。

DB2 インスタンス・ユーザー・アカウント

ローカルまたはドメイン・ユーザー・アカウントは、DB2 インスタンスに必要です。DB2 インスタンス・ユーザー・アカウントは、DB2 をインストールする前に作成することもできますし、DB2 セットアップ・ウィザードに作成させることもできます。DB2 セットアップ・ウィザードに新規ドメイン・ユーザー・アカウントを作成させたい場合には、インストールを実行するために使用するユーザー・アカウントが、ドメイン・ユーザー・アカウントを作成する権限を持っている必要があります。ユーザー・アカウントは、インストールを実行するマシンの管理者 グループに属していなければなりません。このアカウントは、以下のユーザー権限を付与されます。

- オペレーティング・システムの一部として機能
- トークン・オブジェクトの作成

- クォータの増加
- サービスとしてログオン
- プロセス・レベル・トークンの置き換え

すべての DB2 インスタンスには、インスタンスの作成時に割り当てられた 1 つのユーザーがあります。インスタンスの開始時に、DB2 はこのユーザー名でログオンします。

DB2 の Windows 2000 へのインストール時のディレクトリーの選択

DB2 は、すべてのユーザーが書き込みアクセス権限を持っているディレクトリーにインストールする必要があります。DB2 が、限られたユーザー (管理者など) のみが書き込みアクセス権限を持つディレクトリーにインストール済みされた場合、通常のユーザーは DB2 コマンド行プロセッサを使用しようとしたときに、SQL1035N エラーを受け取る可能性があります。

関連概念:

- *DB2 Universal Database サーバー機能 概説およびインストール* の『ユーザー、ユーザー ID、およびグループの命名規則』

関連タスク:

- *DB2 Universal Database サーバー機能 概説およびインストール* の『Windows 上での DB2 サーバーのインストール』

データベースの移行

このタスクは、メインタスクである『DB2 の移行』の一部です。

前提条件:

SYSADM 権限が必要です。

制限:

移行は以下のものからのみサポートされます。

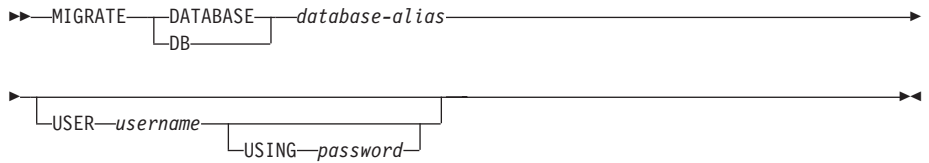
- DB2 バージョン 6.x またはバージョン 7.x。(バージョン 6.x およびバージョン 7.x でサポートされているすべてのプラットフォーム。)
- DB2 DataJoiner V2.1.1 (AIX、Windows NT、および Solaris オペレーティング環境)

手順:

以下のようにして、DB2 データベースを移行します。

1. **db2 migrate database** コマンドを使用し、データベースを移行します。

DB2 MIGRATE DATABASE コマンド



各パラメーターの意味は以下のとおりです。

DATABASE *database-alias*

現在インストールされているバージョンのデータベース・マネージャーに移行するデータベースの別名を指定します。

USER *username*

データベースの移行に使用するユーザー名を識別します。

USING *password*

ユーザー名を認証するのに使用するパスワード。パスワードを省略しても、ユーザー名が指定されている場合には、パスワードを入力するようプロンプトが出されます。

2. オプション: 統計を更新します。データベース移行の完了時に、照会のパフォーマンスを最適化するために使用されていた旧統計は、カタログ内に保存されています。ただし、DB2 バージョン 8 には、改良された統計や、DB2 バージョン 6 または DB2 バージョン 7 にはなかった統計もあります。これらの統計の利点を活用するには、表に対して **runstats** コマンドを実行すると良いでしょう。SQL 照会のパフォーマンスに対して重大な影響のある表の場合は特にそうです。
3. オプション: パッケージを再バインドします。データベースを移行すると、既存のパッケージはすべて無効になります。移行処理後に各パッケージは、DB2 バージョン 8 データベース・マネージャーによって最初に使用されるときに再作成されます。**db2rbind** コマンドを実行し、データベースに格納されているすべてのパッケージを再作成することができます。
4. オプション: PUBLIC からの SQL データ・アクセスを含む、外部ストアド・プロシージャ上で、EXECUTE 特権を取り消します。データベースの移行時に、すべての関数、メソッド、および外部ストアド・プロシージャの EXECUTE 特権が PUBLIC に対して認可されます。これは、SQL データ・アクセスの入った外部ストアド・プロシージャの機密漏れを引き起こします。この外部ストアド・プロシージャを使用すれば、ユーザーは本来なら特権の対象にはなりえない SQL オブジェクトにアクセスできるようになるからです。特権を取り消すには、**db2undgpp - r** コマンドを入力します。
5. オプション: DB2 Explain 表の移行
6. オプション: 移行前に構成設定を記録しておけば、移行前の構成設定を現在の構成設定と比較して、移行が成功したかを確認することができます。以下の事柄を確認します。
 - データベース構成パラメーターの設定

- データベース・マネージャー構成パラメーターの設定
- 表スペース・レコード
- パッケージ・レコード

注: 移行中に、データベース構成パラメーター *maxappls* は自動的に設定されます。これを別の値に設定したい場合は、手動で更新する必要があります。

関連タスク:

- *DB2 Universal Database* サーバー機能 概説およびインストール の『DB2 の移行前のシステム構成設定の記録』
- *DB2 Universal Database* サーバー機能 概説およびインストール の『Explain 表の移行』

関連資料:

- コマンド・リファレンス の『MIGRATE DATABASE コマンド』
- コマンド・リファレンス の『LIST DATABASE DIRECTORY コマンド』
- コマンド・リファレンス の『db2rbind - すべてのパッケージの再バインド・コマンド』

DB2 サーバーのインストールの概要 (Windows)

このトピックでは、Windows 上での DB2® Enterprise Server Edition (単一パーティション) および DB2 Workgroup Server Edition のインストールについての概要を説明します。

インストールの概要:

インストールのための環境の準備

インストールする前に、インストールのためコンピューターを準備する必要があります。以下のようにして、コンピューターを準備します。

1. ご使用のコンピューターが、必要なインストール要件に合っていることを確認します。
2. ご使用のシステムに、DB2 を実行するのに十分なメモリーがあることを確認します。
3. ご使用のシステムに、DB2 インストールのための十分なディスク・スペースがあることを確認します。
4. インストールおよびセットアップに必要なユーザー・アカウントを有していることを確認します。ユーザー・アカウントは、インストール用に 1 つ、セットアップ用に 2 つが必要です。セットアップに必要なユーザー・アカウントは、インストール前に作成することもできますし、DB2 セットアップ・ウィザードに作成させることもできます。

5. Windows® 2000 上にインストールし、 Lightweight Directory Access Protocol (LDAP) を使用して DB2 サーバーを LDAP に登録する場合は、Windows 2000 ディレクトリー・スキーマを拡張して、それに DB2 オブジェクト・クラスと属性定義を組み込めるようにします。

DB2 のインストール

環境を準備した後、DB2 セットアップ・ウィザードを使用して DB2 をインストールします。DB2 セットアップ・ウィザード機能には、以下のものが組み込まれています。

- DB2 セットアップ・ランチパッド。ここからインストール情報およびリリース情報を参照したり、DB2 バージョン 8 の機能について学習したりすることができます。
- 一般、コンパクト、およびカスタムのインストール・タイプ
- 複数言語サポートをインストールするオプション
- DB2 Administration Server のセットアップ (DAS ユーザー・セットアップを含む)
- 管理連絡先およびヘルス・モニター通知のセットアップ
- インスタンスのセットアップおよび構成 (インスタンス・ユーザーのセットアップを含む)
- DB2 ツール・メタデータおよびデータウェアハウス・コントロール・データベースのセットアップ
- 応答ファイルの作成

以下の作業のいくつかは、インストール後まで据え置き、DB2 セットアップ・ウィザードを使用しないで実行することができます。インストール後にこれらの作業を実行することについて詳しくは、下記の関連情報を参照してください。

推奨事項：最新の修正パッケージの適用

DB2 セットアップ・ウィザードを使用して DB2 をインストールした後、最新の DB2 バージョン 8 修正パッケージを適用することをお勧めします。DB2 修正パッケージは、IBM® サポート・サイトで入手できます。

インストールの検証

DB2 セットアップ・ウィザードを使用して DB2 をインストールし、最新の DB2 修正パッケージを適用した後、インストールを検証することをお勧めします。以下のようにして、インストールを検証します。

1. **db2sampl** コマンドを使用して、サンプル・データベースを作成します。ファースト・ステップ・ユーティリティのインストールを選択した場合には、それを使用してサンプル・データベースを作成することもできます。
2. サンプル・データベースが正常に作成されたならば、SQL コマンドを実行してサンプル・データを検索します。

関連概念:

- 管理ガイド: インプリメンテーション の『インスタンスの作成』

関連タスク:

- データウェアハウス・センター 管理ガイド の『インストール時のウェアハウス・コントロール・データベースの初期化』
- 327 ページの『ツール・カタログ・データベースおよび DAS スケジューラーのセットアップと構成』
- 管理ガイド: インプリメンテーション の『通知、連絡先リストのセットアップ、および構成』

関連資料:

- *DB2 Universal Database* サーバー機能 概説およびインストール の『DB2 サーバーのインストール要件 (Windows)』
- コマンド・リファレンス の『UPDATE HEALTH NOTIFICATION CONTACT LIST コマンド』

サテライト・コントロール・サーバーのインストールのエラー・メッセージの場所

DB2 Universal Database Enterprise Server Edition のサテライト・コントロール・サーバー・コンポーネントを選択して、サテライト・コントロール・サーバーをインストールします。インストール関連のメッセージは、以下のようにログに記録されます。

- Windows ベースのプラットフォームの場合
 - /L オプションを使って対話式インストールまたは応答ファイル・インストールを実行し、インストール・メッセージをファイルにリダイレクトしない場合、インストール・メッセージは以下のように db2.log ファイルに書き込まれます。
 - Windows[®] NT の場合:

```
x:\WINNT\Profiles\username\Personal\db2log\db2.log
```

ここで、x: は Windows がブートされるドライブ (ブート・ドライブ) です。
 - Windows 2000 の場合:

```
x:\Documents and Settings\username\My Documents\db2log\db2.log
```

ここで、x: は Windows がブートされるドライブ (ブート・ドライブ) です。
 - /L オプションを使って応答ファイル・インストールを実行する場合、インストール・メッセージは、指定したパスとファイルに書き込まれます。

インストール・プロセス中に作成されるファイルを調べる場合は、任意のエディターを使用できます。

インストール・プログラムがサテライト・コントロール・データベース SATCTLDB を作成する場合、この操作で作成されるメッセージは db2.log ファイルにも記録されます。

- AIX® プラットフォームでは、最新の対話式または応答ファイル・インストールのインストール・メッセージは、/tmp/db2setup.log ファイルに書き込まれます。コマンド行インストールを実行する場合 (**db2_install** コマンドを使用)、エラー・メッセージは /tmp/db2_install_log.pid に書き込まれます。これらのファイルは任意のエディターを使って調べることができます。サテライト・コントロール・サーバーをインストールしたら、AIX 上でサテライト・コントロール・データベース SATCTLDB を作成します。データベースの作成操作に関するメッセージは、指定されたログ・ファイルに書き込まれます。

データ・リカバリーと高可用性の手引き

バックアップの概要

DB2® BACKUP DATABASE コマンドの最も単純な形式の場合、必要なのは、バックアップしたいデータベースの別名を指定することだけです。たとえば、以下のようにします。

```
db2 backup db sample
```

コマンドが正常に完了すると、コマンドを出したパスまたはディレクトリーに新しいバックアップ・イメージが作成されます。このディレクトリーに入れられる理由は、この例のコマンドはバックアップ・イメージの宛先を明示的に指定していないからです。たとえば、Windows® オペレーティング・システムの場合、このコマンドは (ルート・ディレクトリーから出すと)、以下にリストされているディレクトリーにイメージを作成します。

```
Directory of D:\SAMPLE.0\DB2\NODE0000\CATN0000\20010320
```

```
03/20/2001 12:26p <DIR> .
03/20/2001 12:26p <DIR> ..
03/20/2001 12:27p      12,615,680 122644.001
```

注: DB2 クライアントおよびサーバーが同じシステム上にない場合、バックアップ・イメージのデフォルト・ターゲット・ディレクトリーは、コマンドが実行されたクライアント・システムの現行作業ディレクトリーです。このターゲット・ディレクトリーまたはデバイスは、サーバー・システムになければなりません。

バックアップ・ユーティリティーを起動する際にオプションを指定すると、その宛先にバックアップ・イメージが作成されます。指定できる場所は以下のとおりです。

- ディレクトリー (ディスクまたはディスクレットへのバックアップの場合)
- 装置 (テープへのバックアップの場合)

- Tivoli® Storage Manager (TSM) サーバー
- 他のベンダーのサーバー

データベースのバックアップ操作を起動すると、履歴ファイルが要約情報によって自動的に更新されます。このファイルは、データベース構成ファイルと同じディレクトリーに作成されます。

UNIX® ベースのシステムでは、ディスク上に作成されるバックアップ・イメージのファイル名は、複数のエレメントを連結してピリオドで区切ったものになります。

```
DB_alias.Type.Inst_name.NODEnnnn.CATNnnnn.timestamp.Seq_num
```

たとえば、以下のようにします。

```
STAFF.0.DB201.NODE0000.CATN0000.19950922120112.001
```

Windows オペレーティング・システムでは、4 レベルのサブディレクトリー・ツリーが使用されます。

```
DB_alias.Type\Inst_name\NODEnnnn\CATNnnnn\yyyymmdd\hhmms.Seq_num
```

たとえば、以下のようにします。

```
SAMPLE.0\DB2\NODE0000\CATN0000\20010320\122644.001
```

データベース別名	バックアップ・ユーティリティーを起動する際に指定した、1 ~ 8 文字のデータベース別名。
タイプ	バックアップ操作のタイプ。0 は全データベース・レベルのバックアップ、3 は表スペース・レベルのバックアップ、4 は LOAD..COPY TO コマンドによって生成されるバックアップ・イメージ
インスタンス名	DB2INSTANCE 環境変数から取られる 1 ~ 8 文字の現行インスタンス名。
ノード番号	ノード番号。非パーティション・データベース・システムでは、この値は常に NODE0000 です。パーティション・データベース・システムでは、この値は NODExxxx です。xxxx は、db2nodes.cfg ファイル中でノードに割り当てられている番号です。
カタログ・ノード番号	データベースのカタログ・ノードのノード番号。非パーティション・データベース・システムでは、この値は常に CATN0000 です。パーティション・データベース・システムでは、この値は CATNxxxx です。xxxx は、db2nodes.cfg ファイル中でノードに割り当てられている番号です。
タイム・スタンプ	バックアップ操作が実行された日付と時刻を 14 文

字で表記したものの。タイム・スタンプの形式は `yyyymmddhhnnss` です。ただし、

- `yyyy` は年 (1995 ~ 9999)
- `mm` は月 (01 ~ 12)
- `dd` は日 (01 ~ 31)
- `hh` は時 (00 ~ 23)
- `nn` は分 (00 ~ 59)
- `ss` は秒 (00 ~ 59)

シーケンス番号

ファイル拡張子として使用する 3 桁の番号。

バックアップ・イメージをテープに書き込む際には、以下のようになります。

- ファイル名は作成されません。ただし、検査のために上記の情報がバックアップ・ヘッダーに保管されます。
- テープ装置が標準オペレーティング・システム・インターフェースを介して使用可能でなければなりません。しかし、大規模なパーティション・データベース・システムでは、テープ装置を各データベース・パーティション・サーバーに専用接続することは実際的でないことがあります。この場合は、複数のテープ装置を 1 つまたは複数の TSM サーバーに接続することができます。これは、これらのテープ装置に対するアクセスを、各データベース・パーティション・サーバーから可能にするためです。
- パーティション・データベース・システムでは、REELibrarian 4.2 または CLIO/S などの仮想テープ装置機能を提供している製品を使用することもできます。これらの製品を使用し、疑似テープ装置を介して他のノード (データベース・パーティション・サーバー) に接続されているテープ装置にアクセスすることができます。リモート・テープ装置へのアクセスはユーザーが意識せずに実行され、標準オペレーティング・システム・インターフェースを介して疑似テープ装置にアクセスできます。

使用不能状態になっているデータベースのバックアップをとることはできません (バックアップ・ペンディング状態のデータベースを除く)。いずれかの表スペースが異常な状態になっている場合は、その表スペースがバックアップ・ペンディング状態でない限り、そのデータベースまたはその表スペースのバックアップをとることはできません。

| 同じ表での並行バックアップ操作は許可されていません。一度バックアップ操作が表スペースで開始されると、後に続く操作は失敗します (SQL2048)。
|

リストア操作中にシステム障害が起きたために、データベースまたは表スペースが部分的にリストアされた状態になっている場合、そのデータベースまたは表スペースを正常にリストアしてからでなければ、バックアップをとることはできません。

バックアップ操作は、バックアップする表スペースのリストに一時表スペースの名前が含まれていると、失敗します。

バックアップ・ユーティリティには、異なるデータベースのバックアップ・コピーを作成する複数のプロセスのための並行性を制御する機能が用意されています。この並行制御機能のために、すべてのバックアップ操作が終了するまでバックアップ先の装置はオープンしたままになります。バックアップ操作時にエラーが発生してオープン・コンテナがクローズできない場合は、同じドライブに対する他のバックアップ操作にはアクセス・エラーが発生することがあります。この種のアクセス・エラーを訂正するには、エラーが発生したバックアップ操作を終了し、バックアップ先装置との接続も切断する必要があります。バックアップ・ユーティリティを使用してテープへのバックアップの複数の並行操作を実行する場合は、それらのプロセスのバックアップ先を同じテープにしないようにしてください。

バックアップ情報の表示

db2ckbkp を使用して、既存のバックアップ・イメージに関する情報を表示できます。このユーティリティによって、以下のことが可能です。

- バックアップ・イメージの保全性のテスト、およびリストアできるかどうかの判別。
- バックアップ・ヘッダーに保管されている情報の表示。
- バックアップ・イメージのオブジェクトおよびログ・ファイル・ヘッダーに関する情報の表示。

関連概念:

- データ・リカバリーと高可用性 ガイドおよびリファレンス の『リカバリー履歴ファイルについて』

関連資料:

- コマンド・リファレンス の『db2ckbkp - バックアップの検査コマンド』
- 407 ページの『Tivoli Storage Manager』

表スペースでの変更のロールフォワード

データベースが順方向リカバリーであれば、データベース全体を使用する代わりに、表スペースをバックアップしたり、リストアしたり、ロールフォワードしたりできます。個別の表スペースについてリカバリー方針を決めておくこともでき、これにより時間が節約されます。つまり、データベース全体をリカバリーするよりは、データベースの一部をリカバリーした方が時間は短縮されるからです。たとえば、ディスクが不良で、そのディスクに表スペースが 1 つしか含まれていない場合、その表スペースのみをリストアおよびロールフォワードすることができます。その際データベース全体をリカバリーする必要はなく、データベースの残りの部分に対するユーザー・アクセスに影響を与えることもありません。ただし、損傷した表スペースにシステム・カタログ表が含まれている場合は別です。その状態ではデータベースに接続することができません。(システム・カタログ表を含む表スペース・レベルのバックアップ・イメージが使用可能である場合、システム・カタログ表スペースはそれだけでリストアできます。) 表スペース・

レベルのバックアップにより、データベースの重要な部分を他の部分より頻繁にバックアップすることも可能になり、これによりデータベース全体のバックアップより時間は短縮されます。

表スペースがリストアされた後は、常にロールフォワード・ペンディング状態になります。表スペースを使用可能にするためには、表スペースにロールフォワード・リカバリーを実行する必要があります。ログの最後までロールフォワードすることもでき、特定の時点までロールフォワードすることもできます。ただし、システム・カタログ表を含む表スペースを特定の時刻までロールフォワードすることはできません。必ずログの最後までロールフォワードし、データベース内のすべての表スペースに整合性があるようにしてください。

表スペースがロールフォワードされる時、DB2[®] は、その表スペースに影響するログ・レコードが含まれていなくても、すべてのログ・ファイル进行处理します。この表スペースに影響するいずれのログ・レコードも含まれていないログ・ファイルをスキップするには、DB2_COLLECT_TS_REC_INFO レジストリー変数を ON に設定してください。ログ・ファイルのスキップに必要な情報を集めたかを確認するには、ログ・ファイルが作成されて使用される前にレジストリー変数を設定しておく必要があります。

データベース・ディレクトリーにある、表スペース変更履歴ファイル (DB2TSCHG.HIS) は、それぞれの表スペースごとにどのログ进行处理すべきかを追跡します。

db2logsForRfwd ユーティリティーを使用してこのファイルの内容を表示することができ、そこから項目を削除するには、PRUNE HISTORY コマンドを使用します。データベース・リストア操作中、DB2TSCHG.HIS はバックアップ・イメージからリストアされ、その後データベース・ロールフォワード操作中に最新になります。ログ・ファイルの情報が使用できない場合には、ログ・ファイルは、すべての表スペースのリカバリーにそれが必要であるかのように扱われます。

それぞれのログ・ファイルの情報は、ログが非アクティブになった後にディスクにフラッシュされるので、破損の結果としてこの情報が消失する可能性があります。これを補うため、リカバリー操作がログ・ファイルの途中から開始した場合には、ログ全体は、それがシステムのすべての表スペースの変更を含んでいるかのように扱われます。その後、アクティブ・ログが処理され、それらのための情報は再作成されます。より古いログ・ファイルまたはアーカイブログ・ファイルの情報が破損状態で消失し、それらの情報がデータ・ファイル内にはない場合には、それらのログ・ファイルは、表スペース・リカバリー操作時に、それらがすべての表スペースの変更を含んでいるものとして扱われます。

表スペースのロールフォワードを実行する前に、LIST TABLESPACES SHOW DETAIL コマンドを起動してください。このコマンドは、表スペースがロールフォワードできる最早ポイントである 最小リカバリー時間を戻します。最小リカバリー時間は、データ定義言語 (DDL) ステートメントが表スペースまたは表スペース内の表に対し実行されると、更新されます。表スペースは、システム・カタログ表に含まれる情報と同期するように、少なくとも最小リカバリー時間までロールフォワードする必要があります。複数

の表スペースをリカバリーする場合、表スペースは少なくとも、リカバリーされるすべての表スペースの最小リカバリー時間のうちで最大の時間までロールフォワードする必要があります。パーティション・データベース環境では、`LIST TABLESPACES SHOW DETAIL` コマンドをすべてのパーティションで発行してください。表スペースは、少なくとも、すべてのパーティションにあるすべての表スペースの最小リカバリー時間のうちで最大の時間までロールフォワードする必要があります。

特定の時点にロールフォワードする場合で、表が複数の表スペースに含まれている場合には、すべての表スペースを同時にロールフォワードする必要があります。たとえば、表データがある表スペースに含まれていて、その表の索引が別の表スペースに含まれている場合は、両方の表スペースを同じ特定の時点まで同時にロールフォワードする必要があります。

表のデータおよび長形式オブジェクトが別々の表スペースに存在する場合に、その長形式オブジェクト・データが再編成された場合は、データと長形式のオブジェクトのための表スペースは同時にリストアし、ロールフォワードする必要があります。表再編成後に、関係する表スペースのバックアップを作成する必要があります。

表スペースを特定の時点までロールフォワードしたい場合で、その表スペースの中の表が次のいずれかである場合には、

- 別の表スペースにあるマテリアライズ照会表またはステージング表の基本表
- 別の表スペースにあるマテリアライズ照会表またはステージング表

両方の表スペースを同じ時点までロールフォワードする必要があります。そのようにしない場合は、チェック・ペンディング状態で、マテリアライズ照会表またはステージング表がロールフォワード操作の最後に置かれます。マテリアライズ照会表は完全に更新される必要があり、ステージング表は不完全とマークされます。

特定の時点まで表スペースをロールフォワードしようとするときに、その表スペースに含まれる表が別の表スペースに含まれる別の表との間で参照保全の関係にある場合は、同じ特定の時点まで両方の表スペースを同時にロールフォワードする必要があります。そのようにしない場合は、チェック・ペンディング状態で、参照保全の関係にある子表がロールフォワード操作の最後に置かれます。子表が後で制約違反のチェックをされた時に、表全体のチェックが必要になります。以下のいずれかの表が存在する場合には、それらも子表とともにチェック・ペンディング状態に置かれます。

- 子表の任意の下層マテリアライズ照会表
- 子表の任意の下層ステージング表
- 子表の任意の下層外部キー表

それらの表は、チェック・ペンディング状態から戻るために完全処理が必要です。両方の表スペースを同時にロールフォワードすると、ポイント・イン・タイム指定ロールフォワード操作の終了時に制約が活動状態のままになります。

時刻指定の表スペース・ロールフォワード操作を実行する際、トランザクションが、ある表スペースではロールバックされ、別の表スペースではコミットされるという状態にならないよう注意してください。この状態が発生するのは次の場合です。

- ポイント・イン・タイム指定ロールフォワード操作が、トランザクションにより更新された表スペースのサブセットに実行され、その特定のポイント・イン・タイムがトランザクションのコミットされたポイント・イン・タイムより前になっている。
- 特定の時点までロールフォワードされる表スペースに含まれている表にトリガーが関連付けられているか、ロールフォワードされる表スペース以外の表スペースに影響を与えるトリガーによりその表が更新されている。

解決方法として、この状態が発生しないような適切な指定時刻を見つけてください。

QUIESCE TABLESPACES FOR TABLE コマンドを発行して、トランザクションの整合性が保証された時点を確認し、表スペースのロールフォワードに使用することができます。静止要求（共用、更新意図、または排他モード）は、（ロックを使用して）それらの表スペースに対する実行中のトランザクションがすべて完了するのを待機し、新規の要求をブロックします。静止要求が認可されると、表スペースは整合性のある状態になります。ロールフォワードを停止するのに適切な時刻を決定するには、リカバリー履歴ファイルを調べて静止点を見つけ、それらが最小リカバリー時間よりも後に起こったかを確認することができます。

表スペースのポイント・イン・タイム指定ロールフォワード操作が完了した後、表スペースはバックアップ・ペンディング状態になります。ロールフォワードを実行した時点と現在の時間との間の表スペースに対するすべての更新は除去されてしまうため、その表スペースのバックアップを作成する必要があります。表スペースは、直前のデータベース・レベルや表スペース・レベルのバックアップ・イメージからロールフォワードできなくなります。以下の例では、表スペース・レベルのバックアップ・イメージが必要な理由とその使用方法を示しています。（表スペースを使用可能にするためには、データベース全体、バックアップ・ペンディング状態の表スペース、またはバックアップ・ペンディング状態の表スペースを含む表スペースのセットのいずれかをバックアップできます。）

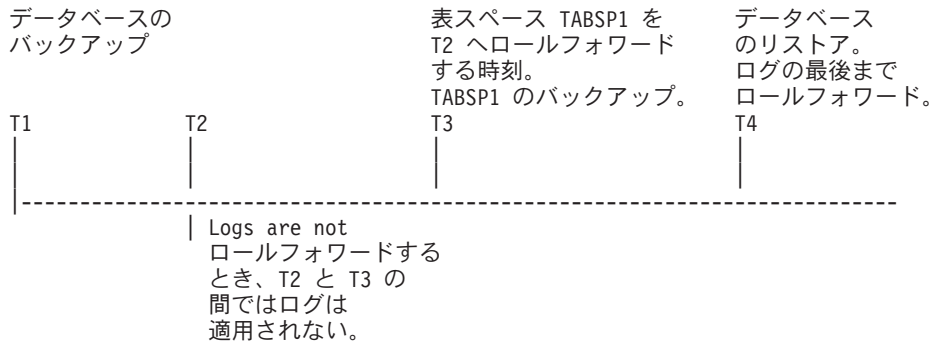


図3. 表スペースのバックアップ要件

上記の例では、データベースは時刻 T1 にバックアップされます。次に、時刻 T3 に、表スペース TABSP1 は特定の指定時刻 (T2) までロールフォワードされ、表スペースは時刻 T3 の後でバックアップされます。表スペースはバックアップ・ペンディング状態であるため、このバックアップ操作は必須です。表スペース・バックアップ・イメージのタイム・スタンプは時刻 T3 より後ですが、表スペースは時刻 T2 にあります。T2 と T3 の間では、ログ・レコードは TABSP1 には適用されません。時刻 T4 では、データベースが T1 で作成されたバックアップ・イメージを使用してリストアされ、ログの最後までロールフォワードされます。表スペース TABSP1 は時刻 T3 でリストア・ペンディング状態になります。これは、TABSP1 に対し T3 と T4 の間では T2 と T3 の間のログ変更項目を表スペースに適用しないで操作が行われたものとデータベース・マネージャーが想定するからです。これらのログ変更が、実際にはデータベースに対するロールフォワードの一部として適用されていた場合、これは誤った想定となります。表スペース・レベルのバックアップは、表スペースが指定ポイント・イン・タイムまでロールフォワードされた後で作成しなければなりません。このバックアップにより、直前のポイント・イン・タイム指定ロールフォワード操作 (例では T3) の後でもその表スペースをロールフォワードできます。

表スペース TABSP1 を T4 までリカバリーしたい場合は、T3 の後に作成したバックアップ・イメージ (必須バックアップまたはそれ以後のバックアップ) から表スペースをリストアし、ログの最後まで TABSP1 をロールフォワードします。

上記の例では、時刻 T4 までデータベースをリストアする最も効果的な方法は、必須ステップを以下の順序で実行する方法です。

1. データベースをリストアする。
2. 表スペースをリストアする。
3. データベースをロールフォワードする。
4. 表スペースをロールフォワードする。

データベースをロールフォワードする前に表スペースをリストアするので、データベースをロールフォワードする際にログ・レコードを表スペースに適用するのにリソースを使用することはありません。

時刻 T3 より後の時間の TABSP1 バックアップ・イメージが見つからないか、TABSP1 を T3 またはそれより前にリストアしたい場合は、以下の操作を実行できません。

- 表スペースを T3 までロールフォワードする。表スペースはデータベース・バックアップ・イメージからリストアされているため、表スペースを再びリストアする必要はありません。
- 時刻 T1 で作成したデータベース・バックアップを使用して表スペースを再びリストアし、次に表スペースを時刻 T3 より前の時刻までロールフォワードする。
- 表スペースをドロップする。

パーティション・データベース環境では、以下のようになります。

- 表スペースのすべての部分を同時に同じ指定時刻までロールフォワードする必要があります。これにより、表スペースはそれぞれのデータベース・パーティションで整合性が保証されます。
- 一部のデータベース・パーティションがロールフォワード・ペンディング状態で、さらに他のデータベース・パーティションで一部の表スペースがロールフォワード・ペンディング状態である (ただしデータベース・パーティション自体はその状態でない) 場合は、まずデータベース・パーティションをロールフォワードし、次に表スペースをロールフォワードします。
- 表スペースをログの最後までロールフォワードする場合は、データベース・パーティションごとにリストアする必要はありません。リカバリーが必要なデータベース・パーティションでのみリストアが必要です。ただし、特定の時刻まで表スペースをロールフォワードしたい場合は、各データベース・パーティションごとにリストアする必要があります。

関連概念:

- データ・リカバリーと高可用性 ガイドおよびリファレンス の『ロード・コピー・ロケーション・ファイルの使用』

関連資料:

- コマンド・リファレンス の『ROLLFORWARD DATABASE コマンド』

Tivoli Storage Manager

BACKUP DATABASE コマンドまたは RESTORE DATABASE コマンドを呼び出す時に、Tivoli Storage Manager (TSM) 製品を使用してデータベースまたは表スペースのバックアップの管理またはリストア操作の管理を行うことを指定できます。TSM クライアント API の最低の必須レベルは、バージョン 4.2.0 です。ただし、64 ビット Solaris

システムでは TSM クライアント API バージョン 4.2.1、64 ビット Windows NT システムでは TSM クライアント API バージョン 5.1 が必須です。

Tivoli Storage Manager クライアントの構成

データベース・マネージャーで TSM オプションを使用できるようにするには、TSM 環境を構成するために以下のステップを行うことが必要になるかもしれません。

1. 機能している TSM クライアントおよびサーバーをインストールし、構成する必要があります。加えて、TSM クライアント API をインストールする必要があります。
2. TSM API で使う環境変数を設定します。

DSMI_DIR API トラストッド・エージェント・ファイル (dsmtca) が存在しているユーザー定義のディレクトリー・パスを識別します。

DSMI_CONFIG dsm.opt ファイルへのユーザー定義ディレクトリー・パスを識別します。これには、TSM ユーザー・オプションが含まれています。他の 2 つの変数とは異なり、この変数には完全修飾パスおよびファイル名を含めなければなりません。

DSMI_LOG エラー・ログ (dserror.log) が作成されるユーザー定義のディレクトリー・パスを指定します。

注: 複数パーティション・データベース環境では、これらの設定は、`sqllib/userprofile` ディレクトリー内で指定する必要があります。

3. それらの環境変数に変更がなされ、データベース・マネージャーが稼働している場合は、以下のようにしてください。
 - **db2stop** コマンドを使用して、データベース・マネージャーを停止します。
 - **db2start** コマンドを使用して、データベース・マネージャーを開始します。
4. サーバーの構成によっては、Tivoli クライアントは TSM サーバーとインターフェースを持つためにパスワードが必要になるかもしれません。TSM 環境が `PASSWORDACCESS=generate` を使用するように構成されている場合には、Tivoli クライアントは、パスワードを設定する必要があります。

実行可能ファイル `dsmapiw` は、インスタンス所有者の `sqllib/adsm` ディレクトリーにインストールされています。この実行可能ファイルにより、TSM パスワードの確立と再設定が可能になります。

`dsmapiw` コマンドを実行するには、ローカル管理者または "root" ユーザーにログインする必要があります。このコマンドを実行すると、以下の情報を入力するよう求められます。

- **旧パスワード。** TSM サーバーで認識されている、TSM ノードの現在のパスワード。このコマンドの初回実行時には、このパスワードは、ご使用のノードが TSM サーバーに登録された時に TSM 管理者から提供されたものになります。

- **新規パスワード。** TSM サーバーに保管させる、TSM ノードの新しいパスワード。(新規パスワードは、入力エラーがないかどうかを調べるため、2 回入力するよう求められます。)

注: BACKUP DATABASE または RESTORE DATABASE コマンドを呼び出すユーザーは、このパスワードを知っている必要はありません。 dsmapiw コマンドを実行する必要があるのは、初期接続時のパスワードを確立する場合、および TSM サーバーのパスワードが初期化された場合のみです。

Tivoli Storage Manager を使用する際の考慮事項: TSM 内部の特定の機能を使用するには、その機能を使用するオブジェクトの完全修飾パス名を指定する必要があります。(Windows オペレーティング・システムでは、/ の代わりに ¥ が使用されることに注意してください。) 完全修飾パス名は、以下のようになります。

- データベース全体のバックアップ・オブジェクト
/<database>/NODEnnnn/FULL_BACKUP.timestamp.seq_no
- 増分データベース・バックアップ・オブジェクト
/<database>/NODEnnnn/DB_INCR_BACKUP.timestamp.seq_no
- 増分差分データベース・バックアップ・オブジェクト
/<database>/NODEnnnn/DB_DELTA_BACKUP.timestamp.seq_no
- 満杯表スペース・バックアップ・オブジェクト
/<database>/NODEnnnn/TSP_BACKUP.timestamp.seq_no
- 増分表スペース・バックアップ・オブジェクト
/<database>/NODEnnnn/TSP_INCR_BACKUP.timestamp.seq_no
- 増分差分表スペース・バックアップ・オブジェクト
/<database>/NODEnnnn/TSP_DELTA_BACKUP.timestamp.seq_no

ここで、<database> はデータベースの別名で、NODEnnnn はノード番号です。大文字で表記されている名前は大文字で入力する必要があります。

- 複数のバックアップ・イメージで同じデータベース別名が使用されている場合は、完全修飾名は、タイム・スタンプと順序番号で区別されます。使用するバックアップ・バージョンを判別するには、TSM を照会する必要があります。
- 個々のバックアップ・イメージは、TSM が管理するファイル・スペースにプールされます。個別のバックアップ・イメージは、TSM API を介して、またはこれらの API を使用する **db2aduti** を介してのみ、操作できます。
- サーバーの構成ファイルの COMMTIMEOUT パラメーターで指定されている一定時間以内に Tivoli クライアントからの応答がないと、セッションは TSM サーバーによりタイムアウトになります。タイムアウト問題には、以下の 3 つの要素が起因している可能性があります。
 - TSM サーバーの COMMTIMEOUT パラメーターの設定値が低すぎる可能性がある。たとえば、リストア操作中に大きな DMS 表スペースが作成されていると、タイムアウトが起こる可能性があります。このパラメーターの推奨値は 6000 秒です。

- DB2 バックアップまたはリストアのバッファーが大きすぎる可能性がある。
- オンライン・バックアップ操作中のデータベース活動が多すぎる可能性がある。
- 複数セッションを使用して、スループットを増加させます (TSM サーバー上で十分なハードウェアが使用可能な場合のみ)。

関連概念:

- データ・リカバリーと高可用性 ガイドおよびリファレンス の『ログ・ファイルの管理』
- *Data Links Manager 概説*およびインストール の『Tivoli Space Manager Hierarchical Storage Manager (AIX)』

関連資料:

- コマンド・リファレンス の『db2adutl - TSM アーカイブ・イメージによる作業コマンド』

ドロップされた表のリカバリー

ときに、まだ必要なデータのある表をドロップしてしまうことがあります。そのような場合は、表ドロップ操作の後でもリカバリー可能なクリティカル表の作成を考慮するとよいでしょう。

表データを、データベース・リストア操作によってリカバリーし、その後、表がドロップされる前の時刻までのデータベース・ロールフォワード操作を実行できます。この操作はデータベースが大きいと時間がかかり、そのリカバリーのあいだはデータは使用できなくなります。

ドロップされた表をリカバリーする機能により、表スペース・レベルのリストアおよびロールフォワード操作を使用して、ドロップされた表をリカバリーすることができます。これはデータベース・レベルのリカバリーより速く、ユーザーもデータベースをそのまま使用できます。

前提条件:

ドロップされた表がリカバリー可能になるには、表が常駐する表スペースで **DROPPED TABLE RECOVERY** オプションがオンになっていなければなりません。これは、表スペース作成の間に行うことができます。または、**ALTER TABLESPACE** ステートメントを起動することによっても行えます。**DROPPED TABLE RECOVERY** オプションは表スペースに固有で、**REGULAR** 表スペースに限定されます。ある表スペースで、ドロップされた表のリカバリーが可能かどうかを判別するには、**SYSCAT.TABLESPACES** カタログ表にある **DROP_RECOVERY** 列を照会することができます。ドロップした表のリカバリーは、新規に作成されたデータ表スペースにおいてデフォルトで使用可能で
す。

DROP TABLE ステートメントが、ドロップされた表のドロップが可能になっている表スペースを持つ表に対して実行されると、追加のエントリー（ドロップされた表を識別する）がログ・ファイル内に作成されます。エントリーはリカバリー履歴ファイルでも作成され、これには表を再作成するのに使用できる情報が含まれます。

制限:

ドロップされた表からリカバリー可能なデータのタイプについて、いくつかの制限事項があります。以下のものは、リカバリーすることはできません。

- ラージ・オブジェクト (LOB) または長形式フィールドのデータ。LARGE 表スペースには DROPPED TABLE RECOVERY オプションはサポートされていません。LOB または LONG VARCHAR 列を含むドロップされた表をリカバリーしようとすると、生成されるエクスポート・ファイルでこれらの列は NULL に設定されます。DROPPED TABLE RECOVERY オプションは REGULAR 表スペースにのみ使用できるもので、一時または LARGE 表スペースには使用できません。
- 行タイプと関連したメタデータ。(データはリカバリーされますが、メタデータはリカバリーされません。)型付き表の階層表にあるデータはリカバリーされます。このデータには、ドロップされた型付き表に現れたよりも詳細にわたる情報が含まれます。

リカバリーされたデータのデータ・タイプが GRAPHIC または VARGRAPHIC の場合、2 つ以上のコード・ページが含まれている可能性があります。このデータをリカバリーするには、IMPORT または LOAD コマンドに USEGRAPHICCODEPAGE ファイル・タイプ修飾子を指定する必要があります。この場合、データのリカバリーに LOAD コマンドを使用すると、リカバリー操作のパフォーマンスが向上します。

手順:

一度でリカバリーできるドロップされた表は 1 つだけです。以下のようにして、ドロップされた表をリカバリーできます。

1. LIST HISTORY DROPPED TABLE コマンドを起動して、ドロップされた表を識別します。ドロップされた表の ID は、Backup IDENTITY 列にリストされます。
2. 表がドロップされる前に作成されたデータベース・レベルまたは表スペース・レベルのバックアップ・イメージをリストアップします。
3. 表データが含まれているファイルを書き込むエクスポート・ディレクトリーを作成します。このディレクトリーは、すべてのデータベース・パーティションからアクセスできるものか、またはそれぞれのパーティションに存在するかのいずれかでなければなりません。このエクスポート・ディレクトリーの下サブディレクトリーは、各データベース・パーティションごとに自動的に作成されます。これらのサブディレクトリーの名前は NODEnnnn です。ここで、nnnn はデータベース・パーティションまたはノード番号を表します。それぞれのデータベース・パーティションにあったときと同じ、ドロップされた表データを含むデータ・ファイルは、data という下位のサブディレクトリーにエクスポートされます。たとえば
\`\export_directory\NODE0000\data` のようになります。

4. ROLLFORWARD DATABASE コマンドの RECOVER DROPPED TABLE オプションを使用して、表がドロップされた後の特定の時刻までロールフォワードします。または、ログの最後までロールフォワードします。こうすると、表スペースまたはデータベース内の他の表への更新は失われません。
5. リカバリー履歴ファイルから CREATE TABLE ステートメントを使用して表を再作成します。
6. ロールフォワード操作中にエクスポートされた表データを、表にインポートします。

DATALINK 列に関連したリンク・ファイルの名前はリカバリーできます。表データのインポートの後に、表を DB2 Data Links Manager に合わせて調整する必要があります。ファイルのバックアップは、ガーベッジ・コレクションですでに削除されたかどうかによって、DB2 Data Links Manager によりリカバリーされる場合とリカバリーされない場合があります。

関連資料:

- 1434 ページの『ALTER TABLESPACE』
- 1484 ページの『CREATE TABLE』
- コマンド・リファレンス の『ROLLFORWARD DATABASE コマンド』
- コマンド・リファレンス の『LIST HISTORY コマンド』

db2Backup - データベースのバックアップ

データベースまたは表スペースのバックアップ・コピーを作成します。

有効範囲:

この API は、それが実行されるデータベース・パーティションにのみ影響を与えません。

許可:

以下のいずれかが必要です。

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

必要な接続:

データベース。この API を呼び出せば、指定したデータベースへの接続が自動的に確立されます。

接続はバックアップの完了時に終了します。

API 組み込みファイル:

db2ApiDf.h

C API 構文:

```

/* File: db2ApiDf.h */
/* API: db2Backup */
/* ... */
SQL_API_RC SQL_API_FN
db2Backup (
    db2UInt32      versionNumber,
    void          *pDB2BackupStruct,
    struct sqlca *pSqlca);

typedef SQL_STRUCTURE db2BackupStruct
{
    char                *piDBAlias;
    char                oApplicationId[SQLU_APPLID_LEN+1];
    char                oTimestamp[SQLU_TIME_STAMP_LEN+1];
    struct db2TablespaceStruct *piTablespaceList;
    struct db2MediaListStruct *piMediaList;
    char                *piUsername;
    char                *piPassword;
    void                *piVendorOptions;
    db2UInt32           iVendorOptionsSize;
    db2UInt32           oBackupSize;
    db2UInt32           iCallerAction;
    db2UInt32           iBufferSize;
    db2UInt32           iNumBuffers;
    db2UInt32           iParallelism;
    db2UInt32           iOptions;
} db2BackupStruct;

typedef SQL_STRUCTURE db2TablespaceStruct
{
    char                **tablespaces;
    db2UInt32           numTablespaces;
} db2TablespaceStruct;

typedef SQL_STRUCTURE db2MediaListStruct
{
    char                **locations;
    db2UInt32           numLocations;
    char                locationType;
} db2MediaListStruct;
/* ... */

```

汎用 API 構文:

```

/* File: db2ApiDf.h */
/* API: db2Backup */
/* ... */
SQL_API_RC SQL_API_FN
db2gBackup (
    db2UInt32      versionNumber,
    void          *pDB2gBackupStruct,

```

db2Backup - データベースのバックアップ

```
    struct sqlca *pSqlca);

typedef SQL_STRUCTURE db2gBackupStruct
{
    char                *piDBAlias;
    db2Uint32           iDBAliasLen;
    char                *poApplicationId;
    db2Uint32           iApplicationIdLen;
    char                *poTimestamp;
    db2Uint32           iTimestampLen;
    struct db2gTablespaceStruct *piTablespaceList;
    struct db2gMediaListStruct *piMediaList;
    char                *piUsername;
    db2Uint32           iUsernameLen;
    char                *piPassword;
    db2Uint32           iPasswordLen;
    void                *piVendorOptions;
    db2Uint32           iVendorOptionsSize;
    db2Uint32           oBackupSize;
    db2Uint32           iCallerAction;
    db2Uint32           iBufferSize;
    db2Uint32           iNumBuffers;
    db2Uint32           iParallelism;
    db2Uint32           iOptions;
} db2gBackupStruct;

typedef SQL_STRUCTURE db2gTablespaceStruct
{
    struct db2Char      *tablespaces;
    db2Uint32           numTablespaces;
} db2gTablespaceStruct;

typedef SQL_STRUCTURE db2gMediaListStruct
{
    struct db2Char      *locations;
    db2Uint32           numLocations;
    char                locationType;
} db2gMediaListStruct;

typedef SQL_STRUCTURE db2Char
{
    char                *pioData;
    db2Uint32           iLength;
    db2Uint32           oLength;
} db2Char;
/* ... */
```

API パラメーター:

versionNumber

入力。 2 番目のパラメーター *pDB2BackupStruct* として渡される構造のバージョンとリリースのレベルを指定します。

pDB2BackupStruct

入力。 *db2BackupStruct* 構造を指すポインター。

pSqlca 出力。 *sqlca* 構造へのポインター。

piDBAlias

入力。バックアップをとるデータベースのデータベース別名 (システム・データベース・ディレクトリーにカタログされている) を含むストリングを指定します。

iDBAliasLen

入力。データベースの別名の長さを示す 4 バイトの符号なし整数 (バイト単位)。

oApplicationId

出力。API によって、アプリケーションにサービスを提供しているエージェントを識別するストリングが戻されます。データベース・モニターを使用するバックアップ操作の進行状況に関する情報を取得することもできます。

poApplicationId

出力。長さ `SQLU_APPLID_LEN+1` (`sqlutil.h` で定義) のバッファを提供します。API によって、アプリケーションにサービスを提供しているエージェントを識別するストリングが戻されます。データベース・モニターを使用するバックアップ操作の進行状況に関する情報を取得することもできます。

iApplicationIdLen

入力。 *poApplicationId* バッファの長さを示す 4 バイトの符号なし整数 (バイト単位)。 `SQLU_APPLID_LEN+1` (`sqlutil.h` で定義) と等しくする必要があります。

oTimestamp

出力。API によって、バックアップ・イメージのタイム・スタンプが戻されます。

poTimestamp

出力。長さ `SQLU_TIME_STAMP_LEN+1` (`sqlutil.h` で定義) のバッファを提供します。API によって、バックアップ・イメージのタイム・スタンプが戻されます。

iTimestampLen

入力。 *poTimestamp* バッファの長さを示す 4 バイトの符号なし整数 (バイト単位)。 `SQLU_TIME_STAMP_LEN+1` (`sqlutil.h` で定義) と等しくする必要があります。

piTablespaceList

入力。バックアップをとる表スペースのリストです。表スペース・レベルのバックアップの場合にのみ必要です。データベース・レベルのバックアップの場合は NULL に設定する必要があります。 `DB2TablespaceStruct` 構造を参照してください。

piMediaList

入力。この構造を使用することにより、呼び出し側はバックアップ操作の宛先を指定することができます。提供される情報は、*locationType* パラメーターの値によって異なります。*locationType* に有効な値 (*sqlutil.h* で定義) は、以下のとおりです。

SQLU_LOCAL_MEDIA

ローカル装置 (テープ、ディスクまたはディスクットの組み合わせ)。

SQLU_TSM_MEDIA

TSM。ロケーション・ポインターが NULL に設定されている場合、DB2 で提供される TSM 共有ライブラリーが使用されます。別のバージョンの TSM 共有ライブラリーが必要な場合には、SQLU_OTHER_MEDIA を使用し、共有ライブラリー名を入力してください。

SQLU_OTHER_MEDIA

ベンダー製品。ロケーション・フィールド内の共有ライブラリー名を提供します。

SQLU_USER_EXIT

ユーザー出口。追加の入力は必要ありません (サーバーが OS/2 上にある場合のみ使用可能です)。

詳細については、*db2MediaListStruct* 構造を参照してください。

piUsername

入力。接続の試行時に使用されるユーザー名を含むストリングを指定します。NULL にすることもできます。

iUsernameLen

入力。ユーザー名の長さを示す 4 バイトの符号なし整数 (バイト単位) です。ユーザー名が提供されていない場合は、ゼロに設定してください。

piPassword

入力。ユーザー名とともに使用されるパスワードを含むストリングを指定します。NULL にすることもできます。

iPasswordLen

入力。パスワードの長さを示す 4 バイトの符号なし整数 (バイト単位) です。パスワードが提供されていない場合は、ゼロに設定してください。

piVendorOptions

入力。情報をアプリケーションからベンダー関数へ渡すのに使用されます。このデータ構造はフラットでなければなりません。つまり、間接のレベルはサポートされません。このデータについては、バイト反転が行われず、また、コード・ページがチェックされないことに注意してください。

iVendorOptionsSize

入力。 *piVendorOptions* フィールドの長さです。65535 バイト以下でなければなりません。

oBackupSize

出力。バックアップ・イメージのサイズ (MB バイト単位) を示します。

iCallerAction

入力。実行するアクションを指定します。有効な値 (*db2ApiDf.h* で定義) は、以下のとおりです。

DB2BACKUP_BACKUP

バックアップを開始します。

DB2BACKUP_NOINTERRUPT

バックアップを開始します。バックアップを自動実行するよう指定します。通常ユーザーの介入を要求するシナリオは、呼び出し側への最初の戻りなしに試行されるか、エラーを生成します。この呼び出し側アクションは、たとえば、バックアップに必要なメディアがすべてマウントされていることが明らかで、ユーティリティーによるプロンプトが必要とされない場合に使用してください。

DB2BACKUP_CONTINUE

ユーザーがユーティリティーによって要求された何らかのアクション (たとえば、新しいテープのマウント) を実行した後で、バックアップを継続します。

DB2BACKUP_TERMINATE

ユーザーがユーティリティーによって要求された何らかのアクションの実行に失敗した場合、バックアップを終了します。

DB2BACKUP_DEVICE_TERMINATE

バックアップに使用される装置のリストから特定の装置を除外します。特定のメディアがいっぱいになると、バックアップは呼び出し側に警告を戻します (一方、残りの装置を使用して処理を継続します)。その場合、この呼び出し側アクションを指定してバックアップを再び呼び出すことによって、警告が生成される原因となった装置を使用装置のリストから除外してください。

DB2BACKUP_PARM_CHK

バックアップを実行することなく、パラメーターの妥当性を検査します。このオプションは、呼び出しが戻った後でデータベース接続を終了しません。この呼び出しが正常に戻った後、ユーザーが *SQLUB_CONTINUE* で呼び出しを発行し、処置を進めることが期待されます。

DB2BACKUP_PARM_CHK_ONLY

バックアップを実行することなく、パラメーターの妥当性を検査しま

db2Backup - データベースのバックアップ

す。この呼び出しが戻る前に、この呼び出しによって確立したデータベース接続は終了し、後続する呼び出しは必要なくなります。

iBufferSize

入力。バッファー・サイズを 4KB の割り振り単位 (ページ) でバックアップします。最小値は 8 単位です。デフォルトは 1024 単位です。

iNumBuffers

入力。使用するバックアップ・バッファーの数を指定します。最小値は 2 です。最大値はメモリーによって制限されます。デフォルト値 2 には 0 を指定できません。

iParallelism

入力。並列処理の度合い (バッファー・マネジュラーの数) を指定します。最小値は 1 です。最大値は 1024 です。デフォルトは 1 です。

iOptions

入力。バックアップ・プロパティのビットマップ。オプションは組み合わされて、ビット単位 OR 演算子を使用して *iOptions* の値を生成します。有効な値 (db2ApiDf.h で定義) は、以下のとおりです。

DB2BACKUP_OFFLINE

オフラインで、データベースへの排他的接続が確立されます。

DB2BACKUP_ONLINE

オンラインで、バックアップ操作の実行中に他のアプリケーションがデータベースにアクセスできるようになります。

注: ユーザーが SMS LOB データに対するロックを保持している場合は、オンライン・バックアップ操作は停止しているように見える場合があります。

DB2BACKUP_DB

データベースの全バックアップ。

DB2BACKUP_TABLESPACE

表スペース・レベルのバックアップ。表スペース・レベルのバックアップの場合は、*piTablespaceList* パラメーターに表スペースのリストを提供してください。

DB2BACKUP_INCREMENTAL

累積 (増分) バックアップ・イメージを指定します。増分バックアップ・イメージとは、最新の成功した全バックアップ操作以降に変更された、すべてのデータベース・データのコピーです。

DB2BACKUP_DELTA

非累積 (差分) バックアップ・イメージを指定します。差分バックアップ・イメージとは、最新の成功した任意のタイプのバックアップ操作以降に変更された、すべてのデータベース・データのコピーです。

tablespaces

バックアップを取る表スペースのリストを指すポインター。C の場合、リストは NULL 終了ストリングです。一般的には、*db2Char* 構造のリストです。

numTablespaces

tablespaces パラメーター内の項目数。

locations

メディア・ロケーションのリストを指すポインター。C の場合、リストは NULL 終了ストリングです。一般的には、*db2Char* 構造のリストです。

numLocations

locations パラメーター内の項目数。

locationType

メディア・タイプを示す文字。有効な値 (*sqlutil.h* で定義) は、以下のとおりです。

SQLU_LOCAL_MEDIA

ローカル装置 (テープ、ディスク、ディスクレット、または名前付きパイプ)

SQLU_TSM_MEDIA

Tivoli Storage Manager。

SQLU_OTHER_MEDIA

ベンダー・ライブラリー。

SQLU_USER_EXIT

ユーザー出口 (サーバーが OS/2 上にある場合のみ使用可能です)。

pioData

文字データ・バッファーを指すポインター。

iLength

入力。 *pioData* バッファーのサイズ。

oLength

出力。将来の使用のために予約されています。

関連資料:

- 管理 API リファレンス の『*sqlmgdb* - データベースの移行』
- 1007 ページの『*db2Rollforward* - データベースのロールフォワード』
- 管理 API リファレンス の『SQLCA』
- 993 ページの『*db2Restore* - データベースのリストア』

関連サンプル:

- 『*dbrecov.sqc* -- How to recover a database (C)』
- 『*dbrecov.sqC* -- How to recover a database (C++)』

db2ReadLogNoConnTerm - データベース接続なしのログ読み取りの終了

本来は db2ReadLogNoConnInit によって初期設定され、db2ReadLogNoConn によって使用されるメモリーを割り振り解除します。

許可:

なし

必要な接続:

なし

API 組み込みファイル:

db2ApiDf.h

C API 構文:

```
/* File: db2ApiDf.h */
/* API: db2ReadLogNoConnTerm */
/* ... */
SQL_API_RC SQL_API_FN
db2ReadLogNoConnTerm (
    db2UInt32      versionNumber,
    void * pDB2ReadLogNoConnTermStruct,
    struct sqlca * pSqlca);

typedef SQL_STRUCTURE db2ReadLogNoConnTermStruct
{
    char                **poReadLogMemPtr;
} db2ReadLogNoConnTermStruct;
/* ... */
```

API パラメーター:

バージョン番号

入力。 2 番目のパラメーター *pDB2ReadLogNoConnTermStruct* として渡される構造のバージョンとリリースのレベルを指定します。

pDB2ReadLogNoConnTermStruct

入力。 *db2ReadLogNoConnTermStruct* 構造を指すポインター。

pSqlca

出力。 *sqlca* 構造へのポインター。

poReadLogMemPtr

出力。 初期設定呼び出しで割り振られたメモリーのブロックを指すポインター。このポインターは解放され、NULL に設定されます。

関連資料:

- 管理 API リファレンス の『SQLCA』

- 1024 ページの『db2ReadLogNoConn - データベース接続なしのログの読み取り』
- 1005 ページの『db2ReadLogNoConnInit - データベース接続なしのログ読み取りの初期設定』

データ移動ユーティリティー ガイドおよびリファレンス

エクスポートを使用した DB2 Data Links Manager データの移動 - 概念

表データがデータベース内にあり、一方 DATALINK 列の参照するファイルが Data Links サーバー上にあるため、エクスポート・ユーティリティーは、データベース上のデータを移動することに加えて、対応する Data Links サーバー上のデータ・ファイルも移動する必要があります。そのためにエクスポート・ユーティリティーは、それぞれの Data Links サーバーごとに 1 つの制御ファイルを作成します。制御ファイルの名前は、Data Links サーバーの名前と同じです。コントロール・ファイルは、`d1fm/YYYYMMDD/HHMMSS` という名前の新しいディレクトリーの中に作成されます (YYYYMMDD は年月日、HHMMSS は時分秒 を表す)。このディレクトリーは、エクスポートされたデータ・ファイルが入る場所と同じディレクトリーの下に作成されます。制御ファイルには、エクスポートされる行の DATALINK 列によって参照される対応する DB2 Data Links Manager のファイル名のリストが入られます。

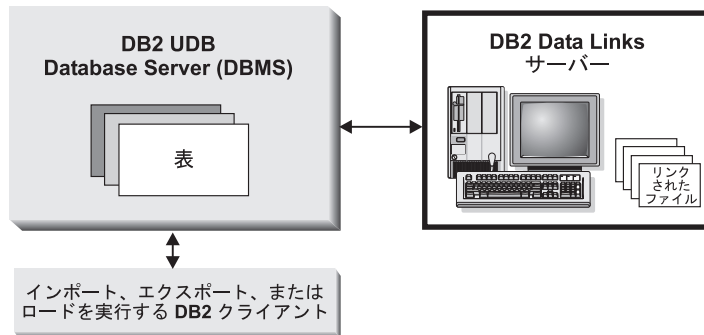


図4. DB2 Data Links Manager のデータの移動：表データがデータベース内にあり、一方 DATALINK 列の参照するファイルが Data Links サーバー上にあるため、エクスポート、インポート、およびロードの各ユーティリティーは、データベース上のデータを移動することに加えて、対応する Data Links サーバー上のデータ・ファイルも移動する必要があります。

Windows® オペレーティング・システムでは、エクスポート・ユーティリティーは、すべての Data Links サーバーに対して制御ファイルを 1 つだけ作成します。この制御ファイルの名前は `ctrlfile.lst` です。これは、`d1fmYYYYMMDDHHMMSS` という名前の新しいディレクトリーに作成されます。このディレクトリーは、エクスポートされたデータ・ファイルが入る場所と同じディレクトリーの下に作成されます。この制御ファイルには、エクスポートされる行の DATALINK 列によって参照されるすべての DB2 Data Links Manager の URL のリストが入られます。

NO LINK CONTROL プロパティの DATALINK 値は、制御ファイルには入れられません。

制御ファイルは、それぞれ対応する Data Links サーバーに移送する必要があります。Windows オペレーティング・システムでは、ただ 1 つのコントロール・ファイルを、参照されているすべての Data Links サーバーに移送する必要があります。各 Data Links サーバーごとに、コントロール・ファイル名を指定して **dlfm_export** ユーティリティを実行する必要があります。このユーティリティは、その Data Links サーバーのコントロール・ファイルに示されているファイルのアーカイブを作成します。**DLFM_FS_ENVIRONMENT** レジストリー変数を適切に設定してから、**dlfm_export** ユーティリティを実行してください。

エクスポート・ユーティリティが SQL アプリケーションとして実行されます。SELECT ステートメントの条件を満たす行と列がデータベースから抽出されます。DATALINK 列に対しては、SELECT ステートメントでスカラー関数を指定しないでください。

エクスポート・ユーティリティは、以下のようなファイルを生成します。

- エクスポート・データ・ファイル。このファイルの DATALINK 列の値の形式は、インポートおよびロード・ユーティリティで使用される形式と同じです。DATALINK 列の値が NULL の場合は、他の NULL 列と同じように扱われます。
- それぞれの Data Links サーバーの制御ファイル。制御ファイルには、その Data Links サーバーからエクスポートされるすべてのファイルの完全なパスと名前のリストが入れられます。Windows オペレーティング・システムの場合、DATALINK 列の値が参照しているすべての Data Links サーバーに対してただ 1 つのコントロール・ファイルが作成されます。

以下のようにして、**dlfm_export** ユーティリティを使って 1 つ以上の Data Links サーバーからファイルをエクスポートします。

```
dlfm_export control-file-name archive-file-name
```

control-file-name は DB2® クライアント上でエクスポート・ユーティリティを実行して生成されるコントロール・ファイルの名前、*archive-file-name* は生成されるアーカイブ・ファイルの名前です。*archive-file-name* のデフォルトは **export.tar** で、現行作業ディレクトリーに入れられます。

dlfm_export が生成したアーカイブからファイルを取り出してリストアするには、**dlfm_import** という補足ユーティリティを使用します。アーカイブ・ファイルと同じ Data Links サーバーにリストアするか別のサーバーにリストアするかに関係なく、このユーティリティを実行する必要があります。

以下のようにして、**dlfm_import** ユーティリティを使ってアーカイブからファイルを取り出します。

`dlfm_import archive-file-name [LISTFILES]`

archive-file-name はファイルのリストアに使うアーカイブ・ファイルの名前です。*archive-file-name* のデフォルトは `export.tar` です。LISTFILES はオプションのキーワードであり、これを指定すると、アーカイブに入っているファイルのリストが戻されません。各 Data Links サーバーごとに **dlfm_import** ユーティリティを root 権限で実行します。これは、アーカイブ・ファイルを別の Data Links サーバーにリストアする場合、そのサーバーのディレクトリ構造やユーザー ID が、**dlfm_export** を実行したサーバーとは異なっている場合があるからです。DLFM_FS_ENVIRONMENT レジストリ変数を適切に設定してから、**dlfm_import** ユーティリティを実行してください。

注: **dlfm_export** ユーティリティを実行した Data Links サーバーとは別のサーバーで **dlfm_import** ユーティリティを実行した場合、ファイルは正しいパスにリストアされます。インポートしているマシンにユーザー ID のうちのいくつかが存在しない場合は、root がファイルを所有します。これらのファイルをデータベースに挿入する前に、すべてのファイルに適切な許可があること、また適切なユーザー ID に属していることを確認してください。

以下の表は、データベース SystemA が参照している DB2 データとファイルを、データベース SystemB へエクスポートする方法を示しています。SystemA は、Data Links サーバー DLFM1 および DLFM2 を使用しています。SystemB は、Data Links サーバー DLFMX および DLFMY を使用しています。DLFM1 上のファイルを DLFMX にエクスポートし、DLFM2 上のファイルを DLFMY にエクスポートします。

データベース SystemA (Data Links サーバー DLFM1 および DLFM2 を使用)			ステップ
ファイル上の DB2 データ	File1 (DLFM1 でのファイル名)	File2 (DLFM2 でのファイル名)	1) 2 つの Data Links サーバーで <code>dlfm_export</code> コマンドを root として実行します。これによって、2 つの Data Links サーバーにアーカイブが作成されます。
データベース SystemB (Data Links サーバー DLFMX および DLFMY を使用)			
	DLFMX 上でアーカイブからリストア	DLFMY 上でアーカイブからリストア	2) 2 つの Data Links サーバーで <code>dlfm_import</code> コマンドを root として実行します。
			3) SystemB 上で <code>IMPORT</code> コマンドを実行します (DL_URL_REPLACE_PREFIX パラメーターを使用して、各エクスポート・ファイルに対する適切な Data Links サーバーを指定)。

データベース SystemA (Data Links サーバー DLFM1 および DLFM2 を使用)	ステップ
SystemB 上で IMPORT コマンドを実行すると、DATALINK 列が参照している SystemA のデータとすべてのファイルがインポートされます。	

関連タスク:

- データ移動ユーティリティ 手引きおよび解説書の『エクスポートの使用』
- 424 ページの『エクスポートを使用した DB2 Data Links Manager データの移動』

関連資料:

- データ移動ユーティリティ 手引きおよび解説書の『sqluxpr - エクスポート』

エクスポートを使用した DB2 Data Links Manager データの移動

手順:

DATALINK 列が参照する表および対応するファイルの一貫性のあるコピーを作成するため、以下のことを実行してください。

1. 以下のコマンドを出して、エクスポート操作中に更新トランザクションが実行されないようにします。

```
db2 quiesce tablespaces for table tablename share
```

2. エクスポート・ユーティリティを起動します。
3. 各 Data Links サーバーで、root 権限で **dlfm_export** ユーティリティを実行します。こうすると、Data Links File Manager の管理者からアクセスできないアーカイブ・ファイルが正常に作成されます。このユーティリティは、アーカイブされているファイルの ACL 情報は取り込みません。 **dlfm_export** への入力として、エクスポート・ユーティリティが生成した制御ファイルの名前を指定します。
4. 以下のコマンドを出して、表の更新を有効にします。

```
db2 quiesce tablespaces for table tablename reset
```

関連概念:

- データ移動ユーティリティ 手引きおよび解説書の『エクスポートの概要』
- 421 ページの『エクスポートを使用した DB2 Data Links Manager データの移動 - 概念』

関連タスク:

- 425 ページの『インポートを使用した DB2 Data Links Manager データの移動』
- 426 ページの『ロードを使用した DB2 Data Links Manager データの移動』

インポートを使用した DB2 Data Links Manager データの移動

表データがデータベース内にあり、一方 DATALINK 列の参照するファイルが Data Links サーバー上にあるため、インポート・ユーティリティーは、データベース上のデータを移動することに加えて、対応する Data Links サーバー上のデータ・ファイルも移動する必要があります。

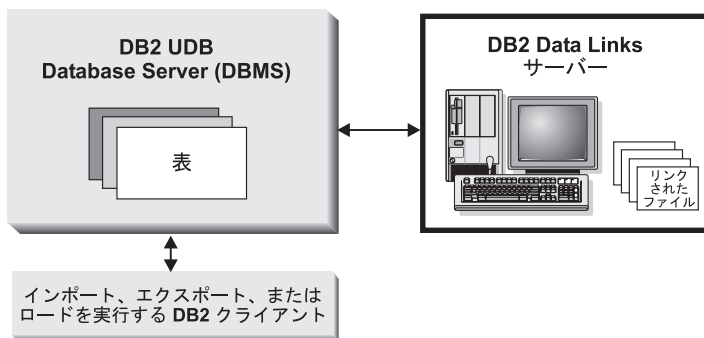


図5. DB2 Data Links Manager のデータの移動：表データがデータベース内にあり、一方 DATALINK 列の参照するファイルが Data Links サーバー上にあるため、エクスポート、インポート、およびロードの各ユーティリティーは、データベース上のデータを移動することに加えて、対応する Data Links サーバー上のデータ・ファイルも移動する必要があります。

手順:

ターゲット・データベースに対してインポート・ユーティリティーを実行する前に、以下のことを実行してください。

1. 参照されるファイルを、適切な Data Links サーバーにコピーします。 **dlfm_import** ユーティリティーを使用することによって、 **dlfm_export** ユーティリティーの生成したアーカイブからファイルを取り出すことができます。
2. Data Links サーバー上の Data Links File Manager に対して、1 つまたは複数の接頭部名を定義します。(データベース登録などのその他の管理作業も実行できます。)
3. 必要なら、DATALINK 列の URL の Data Links サーバー情報を、SQL 表のエクスポート・データから更新します。(元の構成の Data Links サーバーがターゲット・ロケーションのサーバーと同じなら、Data Links サーバー名を更新する必要はありません。)
4. DB2 Data Links Manager 構成ファイル内で、ターゲット構成に Data Links サーバーを定義します。

インポート・ユーティリティーがターゲット・データベースに対して実行される場合、DATALINK 列データの参照するファイルが、該当する Data Links サーバー上でリンクされます。

関連概念:

- データ移動ユーティリティー 手引きおよび解説書の『インポートの概要』

関連タスク:

- 424 ページの『エクスポートを使用した DB2 Data Links Manager データの移動』
- 426 ページの『ロードを使用した DB2 Data Links Manager データの移動』

ロードを使用した DB2 Data Links Manager データの移動

手順:

FILE LINK CONTROL で定義された DATALINK 列を含む表の中にデータをロードする場合は、ロード・ユーティリティーを起動する前に以下のステップを実行してください。(すべての DATALINK 列が NO LINK CONTROL として定義されている場合、これらのステップは必要ありません。)

1. DATALINK 列の値が参照する予定の Data Links サーバー上に DB2 Data Links Manager がインストールされていることを確認します。
2. データベースが DB2 Data Links Manager に登録されていることを確認します。
3. DATALINK 値として挿入されることになるファイルを、適切な Data Links サーバーにコピーします。
4. Data Links サーバー上の DB2 Data Links Manager に接頭部名 (複数可) を定義します。
5. ロードする DATALINK データが参照する Data Links サーバーを、DB2 Data Links Manager 構成ファイルの中で登録します。

ロード操作中に失敗したリンクはデータ保全性違反と見なされ、ユニーク索引違反と同じ方法で処理されます。そのため、1 つ以上の DATALINK 列を含む表をロードする場合の特別な例外が定義されています。

ロード・ユーティリティーの以下の機能は、DATALINK 列をもつ表のロード時にはサポートされません。

- CPU_PARALLELISM (値は強制的に 1 になります)
- LOAD REPLACE
- LOAD TERMINATE
- LOAD COPY

関連資料:

- 1149 ページの『LOAD』

デフォルトの逆方向データ・タイプ・マッピング

データ・ソースのデータ・タイプと連合データベースのデータ・タイプ間のマッピングには、順方向タイプのマッピングと逆方向タイプのマッピングの 2 種類があります。順方向タイプ・マッピングでは、マッピングはリモートのタイプから対応するローカル・タイプへのマッピングです。マッピングのもう一方のタイプは逆方向タイプ・マッピングであり、これはリモートの表を作成または変更するために、透過 DDL で使用されません。

ほとんどのデータ・ソースの場合、ラッパー内にデフォルトのタイプ・マッピングがあります。DB2 ファミリーのデータ・ソース用のデフォルトのタイプ・マッピングは、DRDA ラッパーにあります。Informix 用のデフォルトのタイプ・マッピングは INFORMIX ラッパーなどにあります。

DB2 連合データベースにリモートの表またはビューを定義する際、その定義には、逆方向タイプ・マッピングが含まれます。このマッピングは、それぞれの列のローカルの DB2 (UNIX および Windows 版) のデータ・タイプからのもの、および対応するリモート・データ・タイプからのものです。たとえば、ローカル・タイプ REAL が Informix タイプ SMALLFLOAT を指す、デフォルトの逆方向タイプ・マッピングがあります。

DB2 (UNIX および Windows 版) 連合サーバーは、ローカル・データ・タイプ LONG VARCHAR、LONG VARGRAPHIC、DATALINK、およびユーザー定義タイプのマッピングをサポートしません。

CREATE TABLE ステートメントを使用してリモート表を作成する時に、リモート表に含めたいローカル・データ・タイプを指定します。これらのデフォルトの逆方向タイプ・マッピングは、これらの列に対応するリモート・タイプを割り当てます。たとえば、CREATE TABLE ステートメントを使用して、列 C2 を持つ Informix の表を定義するとします。ステートメント内で C2 のデータ・タイプとして BIGINT を指定します。BIGINT のデフォルトの逆方向タイプ・マッピングは、どのバージョンの Informix で表を作成しているかにより異なります。Informix バージョン 7 では Informix 表の C2 のマッピングは DECIMAL となり、Informix バージョン 8 では INT8 になります。

デフォルトのタイプ・マッピングをオーバーライドすることも、CREATE TYPE MAPPING ステートメントを使用して新しいタイプ・マッピングを作成することもできます。

以下の表は、DB2 (UNIX および Windows 版) のローカル・データ・タイプとリモート・データ・ソースのデータ・タイプ間のデフォルトの逆方向マッピングを示しています。

これらのマッピングは、特に記述のないかぎり、サポート対象のすべてのバージョンで有効です。

DB2 (z/OS および OS/390 版) データ・ソース

表 134. DB2 (z/OS および OS/390 版) のデフォルトの逆方向データ・タイプ・マッピング (表示されていない列があります)

FEDERATED_TYPENAME	FEDERATED_LOWER_LEN	FEDERATED_UPPER_LEN	FEDERATED_LOWER_SCALE	FEDERATED_UPPER_SCALE	FEDERATED_BIT_DATA	FEDERATED_DATA_OPERATORS	REMOTE_TYPENAME	REMOTE_LENGTH	REMOTE_SCALE	REMOTE_BIT_DATA
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHAR	-	-	N
CHARACTER	-	-	-	-	Y	-	CHAR	-	-	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
I DOUBLE	-	8	-	-	-	-	DOUBLE	-	-	-
FLOAT	-	8	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
I REAL	-	4	-	-	-	-	REAL	-	-	-
I SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP	-	10	-	-	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	N
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	-	Y
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	-	N

DB2 (iSeries 版) データ・ソース

表 135. DB2 (iSeries 版) のデフォルトの逆方向データ・タイプ・マッピング (表示されていない列があります)

FEDERATED_TYPENAME	FEDERATED_LOWER_LEN	FEDERATED_UPPER_LEN	FEDERATED_LOWER_SCALE	FEDERATED_UPPER_SCALE	FEDERATED_BIT_DATA	FEDERATED_DATA_OPERATORS	REMOTE_TYPENAME	REMOTE_LENGTH	REMOTE_SCALE	REMOTE_BIT_DATA
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHARACTER	-	-	N
CHARACTER	-	-	-	-	Y	-	CHARACTER	-	-	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	NUMERIC	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	FLOAT	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
REAL	-	4	-	-	-	-	FLOAT	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP	-	10	-	-	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	N
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	-	Y
VARGRAPHIC	-	-	-	-	-	-	VARG	-	-	N

DB2 サーバー (VM および VSE 版) データ・ソース

表 136. DB2 サーバー (VM および VSE 版) のデフォルトの逆方向データ・タイプ・マッピング (表示されていない列があります)

FEDERATED_TYPENAME	FEDERATED_LOWER_LEN	FEDERATED_UPPER_LEN	FEDERATED_LOWER_SCALE	FEDERATED_UPPER_SCALE	FEDERATED_BIT_DATA	FEDERATED_DATA_OPERATORS	REMOTE_TYPENAME	REMOTE_LENGTH	REMOTE_SCALE	REMOTE_BIT_DATA
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHAR	-	-	-
CHARACTER	-	-	-	-	Y	-	CHAR	-	-	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	FLOAT	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP	-	10	-	-	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	-
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	-	Y
VARGRAPH	-	-	-	-	-	-	VARGRAPH	-	-	N

DB2 (UNIX および Windows 版) データ・ソース

表 137. DB2 (UNIX および Windows 版) のデフォルトの逆方向データ・タイプ・マッピング (表示されていない列があります)

FEDERATED_TYPENAME	FEDERATED_LOWER_LEN	FEDERATED_UPPER_LEN	FEDERATED_LOWER_SCALE	FEDERATED_UPPER_SCALE	FEDERATED_BIT_DATA	FEDERATED_DATA_OPERATORS	REMOTE_TYPENAME	REMOTE_LENGTH	REMOTE_SCALE	FEDERATED_BIT_DATA
BIGINT	-	8	-	-	-	-	BIGINT	-	-	-
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHAR	-	-	N
CHARACTER	-	-	-	-	Y	-	CHAR	-	-	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	DOUBLE	-	-	-
FLOAT	-	8	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
I REAL	-	-	-	-	-	-	REAL	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP	-	10	-	-	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	N
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	-	Y
VARGRAPH	-	-	-	-	-	-	VARGRAPHIC	-	-	N
I VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	-	-

Informix データ・ソース

表 138. Informix のデフォルトの逆方向データ・タイプ・マッピング (表示されていない列があります)

	FEDERATED_TYPENAME	FEDERATED_LOWER_LEN	FEDERATED_UPPER_LEN	FEDERATED_LOWER_SCALE	FEDERATED_UPPER_SCALE	FEDERATED_BIT_DATA	FEDERATED_DATA_OPERATORS	REMOTE_TYPENAME	REMOTE_LENGTH	REMOTE_SCALE	REMOTE_BIT_DATA
I	BIGINT ¹	-	-	-	-	-	DECIMAL		19	-	-
	BIGINT ²	-	-	-	-	-	INT8		-	-	-
	BLOB	1	2147483647	-	-	-	BYTE		-	-	-
	CHARACTER	-	-	-	-	N	CHAR		-	-	-
	CHARACTER	-	-	-	-	Y	BYTE		-	-	-
	CLOB	1	2147483647	-	-	-	TEXT		-	-	-
	DATE	-	4	-	-	-	DATE		-	-	-
	DECIMAL	-	-	-	-	-	DECIMAL		-	-	-
	DOUBLE	-	8	-	-	-	FLOAT		-	-	-
	INTEGER	-	4	-	-	-	INTEGER		-	-	-
	LONG VARCHAR	-	32700	-	-	N	TEXT		-	-	-
	LONG VARCHAR	-	32700	-	-	Y	BYTE		-	-	-
	REAL	-	4	-	-	-	SMALLFLOAT		-	-	-
	SMALLINT	-	2	-	-	-	INTEGER		-	-	-
	TIME	-	3	-	-	-	DATETIME		6	10	-
	TIMESTAMP	-	10	-	-	-	DATETIME		0	15	-
	VARCHAR	1	254	-	-	N	VARCHAR		-	-	-
	VARCHAR	255	32672	-	-	N	TEXT		-	-	-
	VARCHAR	-	-	-	-	Y	BYTE		-	-	-
I	VARCHAR ²	255	2048	-	-	N	LVARCHAR		-	-	-

注:

1. このタイプ・マッピングは Informix サーバーのバージョン 7 以下でのみ有効。
2. このタイプ・マッピングは Informix サーバーのバージョン 8 以上でのみ有効。

Oracle SQLNET データ・ソース

表 139. Oracle SQLNET のデフォルトの逆方向データ・タイプ・マッピング (表示されていない列があります)

FEDERATED_TYPENAME	FEDERATED_LOWER_LEN		FEDERATED_UPPER_LEN	FEDERATED_LOWER_SCALE	FEDERATED_UPPER_SCALE	FEDERATED_BIT_DATA	FEDERATED_DATA_OPERATORS	REMOTE_TYPENAME	REMOTE_LENGTH	REMOTE_SCALE	REMOTE_BIT_DATA
BLOB	0	2147483647	0	0	Y	\0		LONG RAW	0	0	Y
CHARACTER	1	254	0	0	N	\0		CHAR	0	0	N
CHARACTER	0	0	0	0	Y	\0		RAW	0	0	Y
CLOB	0	2147483647	0	0	N	\0		LONG	0	0	N
DATE	0	4	0	0	N	\0		DATE	0	0	N
DECIMAL	0	0	0	0	N	\0		NUMBER	0	0	N
DOUBLE	0	8	0	0	N	\0		FLOAT	126	0	N
INTEGER	0	4	0	0	N	\0		NUMBER	10	0	N
REAL	0	4	0	0	N	\0		FLOAT	63	0	N
SMALLINT	0	2	0	0	N	\0		NUMBER	5	0	N
TIME	0	3	0	0	N	\0		DATE	0	0	N
TIMESTAMP	0	10	0	0	N	\0		DATE	0	0	N
VARCHAR	1	2000	0	0	Y	\0		RAW	0	0	Y
VARCHAR	1	4000	0	0	N	\0		VARCHAR2	0	0	N

注: DB2 (UNIX および Windows 版) の BIGINT データ・タイプは透過 DDL では使用できません。リモート Oracle 表を作成する時に CREATE TABLE ステートメントで BIGINT データ・タイプを指定することはできません。

Oracle NET8 データ・ソース

表 140. Oracle NET8 のデフォルトの逆方向データ・タイプ・マッピング (表示されていない列があります)

FEDERATED_TYPENAME	FEDERATED_LOWER_LEN		FEDERATED_UPPER_LEN	FEDERATED_LOWER_SCALE	FEDERATED_UPPER_SCALE	FEDERATED_BIT_DATA	FEDERATED_DATA_OPERATORS	REMOTE_TYPENAME	REMOTE_LENGTH	REMOTE_SCALE	REMOTE_BIT_DATA
BLOB	0	2147483647	0	0	Y	\0	BLOB	0	0	Y	
CHARACTER	1	254	0	0	N	\0	CHAR	0	0	N	
CHARACTER	0	0	0	0	Y	\0	RAW	0	0	Y	
CLOB	0	2147483647	0	0	N	\0	CLOB	0	0	N	
DATE	0	4	0	0	N	\0	DATE	0	0	N	
DECIMAL	0	0	0	0	N	\0	NUMBER	0	0	N	
DOUBLE	0	8	0	0	N	\0	FLOAT	126	0	N	
INTEGER	0	4	0	0	N	\0	NUMBER	10	0	N	
REAL	0	4	0	0	N	\0	FLOAT	63	0	N	
SMALLINT	0	2	0	0	N	\0	NUMBER	5	0	N	
TIME	0	3	0	0	N	\0	DATE	0	0	N	
TIMESTAMP	0	10	0	0	N	\0	DATE	0	0	N	
VARCHAR	1	4000	0	0	N	\0	VARCHAR2	0	0	N	
VARCHAR	1	2000	0	0	Y	\0	RAW	0	0	Y	

注: DB2 (UNIX および Windows 版) の BIGINT データ・タイプは透過 DDL では使用できません。リモート Oracle 表を作成する時に CREATE TABLE ステートメントで BIGINT データ・タイプを指定することはできません。

Microsoft SQL Server データ・ソース

表 141. Microsoft SQL Server のデフォルトの逆方向データ・タイプ・マッピング (表示されていない列があります)

FEDERATED_TYPENAME	FEDERATED_LOWER_LEN	FEDERATED_UPPER_LEN	FEDERATED_LOWER_SCALE	FEDERATED_UPPER_SCALE	FEDERATED_BIT_DATA	FEDERATED_DATA_OPERATORS	REMOTE_TYPENAME	REMOTE_LENGTH	REMOTE_SCALE	REMOTE_BIT_DATA
BLOB	-	-	-	-	-	-	image	-	-	-
CHARACTER	-	-	-	-	Y	-	binary	-	-	-
CHARACTER	-	-	-	-	N	-	char	-	-	-
CLOB	-	-	-	-	-	-	text	-	-	-
DATE	-	4	-	-	-	-	datetime	-	-	-
DECIMAL	-	-	-	-	-	-	decimal	-	-	-
DOUBLE	-	8	-	-	-	-	float	-	-	-
INTEGER	-	-	-	-	-	-	int	-	-	-
SMALLINT	-	-	-	-	-	-	smallint	-	-	-
REAL	-	4	-	-	-	-	real	-	-	-
TIME	-	3	-	-	-	-	datetime	-	-	-
TIMESTAMP	-	10	-	-	-	-	datetime	-	-	-
VARCHAR	1	8000	-	-	N	-	varchar	-	-	-
VARCHAR	8001	32672	-	-	N	-	text	-	-	-
VARCHAR	1	8000	-	-	Y	-	varbinary	-	-	-
VARCHAR	8001	32672	-	-	Y	-	image	-	-	-

Sybase データ・ソース

表 142. Sybase CTLIB および DBLIB のデフォルトの逆方向データ・タイプ・マッピング (表示されていない列があります)

FEDERATED_TYPENAME	FEDERATED_LOWER_LEN	FEDERATED_UPPER_LEN	FEDERATED_LOWER_SCALE	FEDERATED_UPPER_SCALE	FEDERATED_BIT_DATA	FEDERATED_DATA_OPERATORS	REMOTE_TYPENAME	REMOTE_LENGTH	REMOTE_SCALE	REMOTE_BIT_DATA
BIGINT	-	-	-	-	-	-	decimal	19	0	-
BLOB	-	-	-	-	-	-	image	-	-	-
CHARACTER	-	-	-	-	N	-	char	-	-	-
CHARACTER	-	-	-	-	Y	-	binary	-	-	-
CLOB	-	-	-	-	-	-	text	-	-	-
DATE	-	-	-	-	-	-	datetime	-	-	-
DECIMAL	-	-	-	-	-	-	decimal	-	-	-
DOUBLE	-	-	-	-	-	-	float	-	-	-
GRAPHIC	-	-	-	-	-	-	nchar	-	-	-
INTEGER	-	-	-	-	-	-	integer	-	-	-
REAL	-	-	-	-	-	-	real	-	-	-
SMALLINT	-	-	-	-	-	-	smallint	-	-	-
TIME	-	-	-	-	-	-	datetime	-	-	-
TIMESTAMP	-	-	-	-	-	-	datetime	-	-	-
VARCHAR	1	255	-	-	N	-	varchar	-	-	-
VARCHAR	256	32672	-	-	N	-	text	-	-	-
VARCHAR	1	255	-	-	Y	-	varbinary	-	-	-
VARCHAR	256	32672	-	-	Y	-	image	-	-	-
VARGRAPHIC	1	255	-	-	-	-	nvarchar	-	-	-

連合システムのサーバー・オプション

サーバー・オプションは、CREATE SERVER ステートメントでデータ・ソース・サーバーを記述するために使用されます。サーバー・オプションは、データ保全性、ロケーション、セキュリティ、およびパフォーマンス情報を指定します。サーバー・オプションのいくつかはデータ・ソース特有であり、以下の表にそのことが注記されています。Life Sciences データ・ソースには、追加の非常に特異なサーバー・オプションがあります。

共通の連合サーバー・オプションは以下のとおりです。

- 互換性オプション。COLLATING_SEQUENCE、IGNORE_UDT
- データ保全性オプション。IUD_APP_SVPT_ENFORCE
- ロケーション・オプション。CONNECTSTRING、DBNAME、IFILE
- セキュリティ・オプション。FOLD_ID、FOLD_PW
- パフォーマンス・オプション。COMM_RATE、CPU_RATIO、IO_RATIO、LOGIN_TIMEOUT、PACKET_SIZE、PLAN_HINTS、PUSHDOWN、TIMEOUT、VARCHAR_NO_TRAILING_BLANKS

表 143. サーバー・オプションとその設定値

オプション	有効な設定	デフォルトの 設定値	適用
COLLATING_SEQUENCE	データ・ソースが連合データベースと同じデフォルト照合シーケンスを使用するかどうかを、NLS コード・セットと国別情報に基づいて指定する。	'N'	DB2 for iSeries
	'Y' データ・ソースは DB2 連合データベースと同じ照合シーケンスを持つ。		DB2 for z/OS and OS/390
	'N' データ・ソースは DB2 連合データベースの照合シーケンスとは別の照合シーケンスを持つ。		DB2 for UNIX および DB2 for Windows
	'I' データ・ソースは DB2 連合データベースの照合シーケンスとは別の照合シーケンスを持ち、データ・ソースの照合シーケンスは大文字小文字の区別をしない (たとえば、'STEWART' と 'StewART' は等しいと見なされる)。		Informix、 MS SQL Server、 ODBC、 Oracle、 Sybase

表 143. サーバー・オプションとその設定値 (続き)

オプション	有効な設定	デフォルトの 設定値	適用
COMM_RATE	<p>連合サーバーとデータ・ソース・サーバー間の通信レートを指定する。秒当たりの MB 単位で表されます。</p> <p>有効な値は 0 より大きく、2147483648 より小さい値です。値は整数でのみ指定できます (たとえば、12 など)。</p>	'2'	<p>DB2 for iSeries</p> <p>DB2 for z/OS and OS/390</p> <p>DB2 for UNIX および DB2 for Windows</p> <p>Informix、MS SQL Server、ODBC、Oracle、Sybase</p>
CONNECTSTRING	OLE DB Provider への接続に必要な初期化プロパティを指定する。	なし	OLE DB
CPU_RATIO	<p>データ・ソースの CPU が連合サーバーの CPU より、どれほど速いまたは遅いかを表す。</p> <p>有効な値は 0 より大きく、1×10^{23} より小さい値です。値は、任意の有効な倍精度表記 (たとえば、123E10、123、または 1.21E4 など) で表現できます。</p>	'1.0'	<p>DB2 for iSeries</p> <p>DB2 for z/OS and OS/390</p> <p>DB2 for UNIX および DB2 for Windows</p> <p>Informix、MS SQL Server、ODBC、Oracle、Sybase</p>

表 143. サーバー・オプションとその設定値 (続き)

オプション	有効な設定	デフォルトの 設定値	適用
DBNAME	連合サーバーにアクセスさせるデータ・ソース・データベースの名前。 DB2 の場合、この値は 1 つのインスタンス内の特定のデータベースに対応するか、または DB2 for z/OS and OS/390 では、データベースの LOCATION 値に対応します。 Oracle インスタンスに含まれるのは 1 つのデータベースだけなので、Oracle データ・ソースには該当しません。	なし	DB2 for iSeries DB2 for z/OS and OS/390 DB2 for UNIX および DB2 for Windows Informix、 MS SQL Server、 ODBC、 Sybase
FOLD_ID (この表の最後にある注 1 およ び 4 を参照。)	連合サーバーが認証のためにデータ・ソース・サーバーに送信するユーザー ID に適用される。 有効な値は以下のとおりです。 'U' 連合サーバーは、ユーザー ID をデータ・ソースに送信する前に、大文字に変換します。これは、DB2 ファミリーおよび Oracle データ・ソースについては当然の選択です (この表の最後にある注 2 を参照。) 'N' 連合サーバーは、ユーザー ID をデータ・ソースに送信する前に、ユーザー ID に対して何の処理も行いません。(この表の最後の注 2 を参照。) 'L' 連合サーバーは、ユーザー ID をデータ・ソースに送信する前に、小文字に変換します。 これらの設定値のいずれも使用しない場合は、連合サーバーはユーザー ID を大文字にしてデータ・ソースに送信しようとします。そのユーザー ID を正常に送信できない場合は、サーバーはユーザー ID を小文字で送信しようとします。	なし	DB2 for iSeries DB2 for z/OS and OS/390 DB2 for UNIX および DB2 for Windows Informix、 MS SQL Server、 ODBC、 Oracle、 Sybase

表 143. サーバー・オプションとその設定値 (続き)

オプション	有効な設定	デフォルトの 設定値	適用
FOLD_PW (この表の最後にある注 1、3 および 4 を参照。)	<p>連合サーバーが認証のためにデータ・ソースに送信するパスワードに適用される。有効な値は以下のとおりです。</p> <p>'U' 連合サーバーは、パスワードをデータ・ソースに送信する前に、大文字に変換します。これは、DB2 ファミリーおよび Oracle データ・ソースについては当然の選択です。</p> <p>'N' 連合サーバーは、パスワードをデータ・ソースに送信する前に、パスワードに対して何の処理も行いません。</p> <p>'L' 連合サーバーは、パスワードをデータ・ソースに送信する前に、小文字に変換します。</p> <p>これらの設定値のいずれも使用しない場合は、連合サーバーはパスワードを大文字にしてデータ・ソースに送信しようとします。そのパスワードを正常に送信できない場合は、サーバーはパスワードを小文字で送信しようとします。</p>	なし	<p>DB2 for iSeries</p> <p>DB2 for z/OS and OS/390</p> <p>DB2 for UNIX および DB2 for Windows</p> <p>Informix、MS SQL Server、ODBC、Oracle、Sybase</p>
IFILE	<p>Sybase Open Client インターフェース・ファイルのパスと名前を指定する。 Windows NT 連合サーバーでは、デフォルトは %DB2PATH%\interfaces です。 UNIX 連合サーバーでは、デフォルトのパスと名前の値は、 class="IBMfilepath\$DB2INSTANCE/sql1lib/interfaces です。</p>	なし	Sybase

表 143. サーバー・オプションとその設定値 (続き)

オプション	有効な設定	デフォルトの 設定値	適用
IGNORE_UDT	<p>連合サーバーが、「強い型定義」(strong typing) を持たない UDT の基礎にある組み込みタイプを判別すべきかどうかを指定する。 CTLIB および DBLIB プロトコルを使用してアクセスされるデータ・ソースにのみ適用されます。有効な値は以下のとおりです。</p> <p>‘Y’ UDT がユーザー定義であるという事実を無視し、その基礎にある組み込みタイプを判別する。</p> <p>‘N’ UDT のユーザー定義の指定を無視しない。</p>	‘N’	Sybase
	<p>DB2 はニックネームを作成する時に、ニックネームが指すオブジェクト (表、ビュー、ストアード・プロシージャ) についての情報を探し、カタログに入れます。情報を探しているときに、あるオブジェクトのデータ・タイプが認識されないものであることがわかる場合があります (つまり、連合データベースにある対応するものとマッピングできないデータ・タイプ)。このような認識できないタイプとして、次のようなものがあります。</p> <ul style="list-style-type: none"> • 新しい組み込みタイプ • 「強い型定義」を持つ UDT • 「強い型定義」を持たない UDT。これらは、ユーザーが単に名前変更した組み込みタイプです。これらのタイプは、ある種のデータ・ソース (たとえば Sybase や Microsoft SQL Server) でのみサポートされます。 		
	<p>連合サーバーが認識できないデータ・タイプを検出すると、エラー・メッセージ SQL3324N が戻されます。しかし、これには例外があります。 CTLIB または DBLIB プロトコルを使用してアクセス可能なデータ・ソースの場合、 IGNORE_UDT サーバー・オプションをセットすると、連合データベースは、「強い型定義」を持たない、認識不能な UDT を検出した場合に、UDT の基礎にある組み込みタイプが何かを判別します。そして、連合データベースはこの組み込みタイプを認識した場合、組み込みタイプについての情報をカタログに戻します。「強い型定義」を持たない UDT の基礎にある組み込みタイプを、連合データベースに判別させるには、 IGNORE_UDT を ‘Y’ にします。</p>		

表 143. サーバー・オプションとその設定値 (続き)

オプション	有効な設定	デフォルトの設定値	適用
IO_RATIO	<p>データ・ソースの入出力システムが連合サーバーの入出力システムより、どれほど速いまたは遅いかを表す。</p> <p>有効な値は 0 より大きく、1×10^{23} より小さい値です。値は、任意の有効な倍精度表記 (たとえば、123E10、123、または 1.21E4 など) で表現できます。</p>	'1.0'	<p>DB2 for iSeries</p> <p>DB2 for z/OS and OS/390</p> <p>DB2 for UNIX および DB2 for Windows</p> <p>Informix、MS SQL Server、ODBC、Oracle、Sybase</p>
IUD_APP_SVPT_ENFORCE	<p>DB2 連合システムがアプリケーションの保管点ステートメントの検出または作成を強制するかどうかを指定する。</p> <p>'Y' 連合サーバーは、データ・ソースがアプリケーション保管点ステートメントをサポートしない場合、ニックネームに対する INSERT、UPDATE、または DELETE ステートメントを許可しません。DB2 がアトミック INSERT、UPDATE、または DELETE を実行できない場合、SQL エラー・コード (SQL20190) が出されます。</p> <p>'N' 連合サーバーは、ニックネームに対する INSERT、UPDATE、または DELETE ステートメントを許可します。</p>	'Y'	<p>DB2 for iSeries</p> <p>DB2 for z/OS and OS/390</p> <p>DB2 for UNIX および DB2 for Windows</p> <p>Informix、MS SQL Server、ODBC、Oracle、Sybase</p>
LOGIN_TIMEOUT	<p>ログイン要求に対して、Sybase Open Client からの応答を DB2 連合サーバーが待つ秒数を指定する。デフォルト値は TIMEOUT と同じです。</p>	'0'	Sybase
NODE	<p>データ・ソースを RDBMS へのインスタンスとして定義するための名前。</p>	なし	<p>Informix、MS SQL Server、Oracle、Sybase</p>

表 143. サーバー・オプションとその設定値 (続き)

オプション	有効な設定	デフォルトの 設定値	適用
PACKET_SIZE	Sybase インターフェース・ファイルのパケット・サイズをバイト単位で指定する。指定したパケット・サイズをデータ・ソースがサポートしない場合、接続は失敗します。各レコードが非常に大きい場合 (たとえば、大きな表に行を挿入する場合など) にパケット・サイズを増やすと、パフォーマンスが非常によくなります。バイト・サイズは数値です。		Sybase
PLAN_HINTS	プラン・ヒント を使用可能にするかどうかを指定する。プラン・ヒントはステートメントの一部であり、データ・ソース・オプティマイザーに対しての追加情報を提供します。特定の照会タイプについてこの情報を利用すれば、照会パフォーマンスを改善することができます。プラン・ヒントは、データ・ソース・オプティマイザーが索引を使用するかどうか、どの索引を使用するか、またはどの表結合シーケンスを使うかを判別するのに役立ちます。	'N'	Informix、 MS SQL Server、 ODBC、 Oracle、 Sybase
	'Y' データ・ソースがプラン・ヒントをサポートしている場合は、プラン・ヒントを使用可能にします。		
	'N' プラン・ヒントはデータ・ソースで使用可能になりません。		
PUSHDOWN	'Y' DB2 はデータ・ソースに操作を評価させることを考慮します。	'Y'	DB2 for iSeries
	'N' DB2 はリモート・データ・ソースから列を検索するだけで、データ・ソースに結合などのその他の操作を評価させません。		DB2 for z/OS and OS/390
			DB2 for UNIX および DB2 for Windows
			Informix、 MS SQL Server、 ODBC、 Oracle、 Sybase

表 143. サーバー・オプションとその設定値 (続き)

オプション	有効な設定	デフォルトの 設定値	適用
TIMEOUT	何らかの SQL ステートメントに対する Sybase Open Client からの応答を、DB2 連合サーバーが待つ秒数を指定する。 <i>seconds</i> の値は、DB2 Universal Database の整数範囲内にある、正の整数です。指定するタイムアウト値は、使用するラッパーにより異なります。Sybase ラッパーの TIMEOUT オプションのデフォルトは 0 であり、これは DB2 が無期限に応答を待つことを意味します。	'0'	Sybase
VARCHAR_NO_ TRAILING_BLANKS	このオプションは、データ・ソースに可変長の文字データ・タイプがあり、その可変長データが末尾ブランクで埋められていない場合に適用されます。ある種のデータ・ソース (Oracle など) は、ブランク埋め込みなしの比較をするセマンティクスを持ち、これは DB2 for UNIX および DB2 for Windows の比較セマンティクスと同じ結果を戻します。このオプションは、指定されたサーバーからアクセスされるデータ・ソース・オブジェクト内の、すべての VARCHAR および VARCHAR2 列にこれを適用したい場合に設定してください。これにはビューも含まれます。 'Y' このデータ・ソースは、連合サーバーと類似のブランク埋め込みなしで比較するセマンティクスを持つ。 'N' このデータ・ソースは、連合サーバーとは異なる、可変長の文字比較セマンティクスを持つ。	'N'	DB2 for iSeries DB2 for z/OS and OS/390 DB2 for UNIX and Windows Informix、 MS SQL Server、 ODBC、 Oracle、 Sybase

この表に関する注

1. このフィールドは、認証に指定される値に関係なく適用されます。
2. DB2 はユーザー ID を大文字で保管するので、値 'N' と 'U' は論理的に互いに同等です。
3. パスワードの設定が 'N' の場合は、 FOLD_PW を設定しても効果はありません。パスワードが送信されないので、大文字小文字の区別は意味をなしません。
4. これらのいずれのオプションについても、NULL 値を設定することは避けてください。NULL 値を設定すると、ユーザー ID とパスワードを解決するために DB2 は複数回試行することになるので、便利のように思えますが、パフォーマンスが悪化する可能性があります (DB2 がユーザー ID とパスワードを 4 回送信した後で、ようやくデータ・ソース認証に成功するということもあり得ます)。

関連概念:

- SQL リファレンス 第 1 巻 の『サーバー定義およびサーバー・オプション』

- 連合システム・ガイド の『プッシュダウンの可能性に影響を与えるサーバー特性』
- 連合システム・ガイド の『グローバルな最適化に影響を与えるサーバー特性』

関連タスク:

- *DB2 Life Sciences Data Connect* 計画、インストールおよび構成のガイド の『表構造ファイルのサーバーの登録』
- *DB2 Life Sciences Data Connect* 計画、インストールおよび構成のガイド の『Documentum データ・ソースのサーバーの登録』
- *DB2 Life Sciences Data Connect* 計画、インストールおよび構成のガイド の『Excel データ・ソースのサーバーの登録』
- *DB2 Life Sciences Data Connect* 計画、インストールおよび構成のガイド の『BLAST データ・ソースのサーバーの登録』
- *DB2 Life Sciences Data Connect* 計画、インストールおよび構成のガイド の『XML データ・ソースのサーバーの登録』

関連資料:

- *SQL リファレンス* 第 2 巻 の『CREATE SERVER ステートメント』

サーバー定義を提供する

ラッパーを作成した後、アクセスしたいそれぞれのデータ・ソース・サーバーを指定する必要があります。サーバー定義を作成するには、DB2[®] コントロール・センターを使用します。DB2 コマンド・センター、またはコマンド行プロセッサ (CLP) から CREATE SERVER ステートメントを実行することもできます。CREATE SERVER ステートメントに必要なパラメーターとオプションは、アクセスするデータ・ソースによって異なります。

DB2 ファミリーのデータ・ソースの場合:

2 つの DB2 (OS/390[®] 版) データベースがあるとします。NEWYORKDB データベースは NEWYORK390 サーバーにあり、クライアント表と従業員表を含んでいます。LONDONDB データベースは LONDON390 サーバーにあり、売上表を含んでいます。ここでは 2 つのサーバー定義 (1 つは NEWYORK390 サーバー用、1 つは LONDON390 サーバー用) を作成する必要があります。次に、3 つのニックネーム (それぞれの表ごとに 1 つ) を作成する必要があります。

NEWYORK390 サーバー用のサーバー定義の例を以下に示します。

```

| CREATE SERVER NEWYORK TYPE DB2/390
| VERSION 5 WRAPPER DRDA®
| AUTHID 'STEWART' PASSWORD 'BONNIE',
| OPTIONS (DBNAME 'NYCLIENTS')

```

ここで、

NEWYORK

これは、データ・ソース・サーバーにユーザーが割り当てる名前です。この名前はユニークでなければなりません。

TYPE DB2/390

アクセスを構成しているデータ・ソースのタイプを指定します。

VERSION 5

これは、アクセスしたいデータ・ソース・サーバー・ソフトウェアのバージョンです。

WRAPPER DRDA

これは、CREATE WRAPPER ステートメントで指定したラッパー名です。

AUTHID 'STEWART'

これは、データ・ソース側の許可 ID です。この値は大文字小文字の区別があります。

PASSWORD 'BONNIE'

これは、データ・ソース側の許可 ID に関連したパスワードです。この値は大文字小文字の区別があります。

DBNAME 'NYCLIENTS'

これは、データ・ソース・サーバーが存在するデータベースの名前です。この値は大文字小文字の区別があります。

AUTHID パラメーター、PASSWORD パラメーター、および DBNAME オプションは必要です。

Informix、Sybase、および OLE DB データ・ソースの場合:

SY6500 サーバー上に SYBDB と呼ばれる Sybase データベースが 1 つあるとします。SYBDB データベースには 3 つのオブジェクト (2 つの表と 1 つのビュー) が含まれています。ここでは SY6500 サーバー用にサーバー定義を 1 つ作成する必要があります。次に、3 つのニックネーム (それぞれの表ごとに 1 つとビュー用に 1 つ) を作成する必要があります。

SY6500 サーバー用のサーバー定義の例を以下に示します。

```
CREATE SERVER SYBSERVER TYPE SYBASE
VERSION 12.0 WRAPPER CTLIB
OPTIONS (NODE 'sybnode' DBNAME 'sybDB')
```

ここで、

SYBSERVER

これは、データ・ソース・サーバーにユーザーが割り当てる名前です。この名前はユニークでなければなりません。

TYPE SYBASE

アクセスを構成しているデータ・ソースのタイプを指定します。

VERSION 12.0

これは、アクセスしたいデータ・ソース・サーバー・ソフトウェアのバージョンです。

WRAPPER CTLIB

これは、CREATE WRAPPER ステートメントで指定したラッパー名です。

NODE 'sybnode'

これは、データ・ソース・サーバーが存在するノードの名前です。この値は大文字小文字の区別があります。

DBNAME 'sybDB'

これは、アクセスしたいデータベースの名前です。この値は大文字小文字の区別があります。

NODE オプションと DBNAME オプションは必要です。さらに、Informix™ では IUD_APP_SVPT_ENFORCE サーバー・オプションを 'N' にする必要があります。

Oracle、ODBC、および Microsoft SQL Server データ・ソースの場合:

Oracle、ODBC、および Microsoft® SQL Server データ・ソース用のサーバー定義は、Informix、Sybase、および OLE DB データ・ソースの場合と似ています。唯一の相違は、DBNAME オプションが必要ではないことです。

NODE オプションは必要です。

ノードの概念はデータ・ソースによって異なります。リレーショナル・データ・ソースの場合、ノードはデータ・ソースのサーバー・インスタンスを反映します。DB2 では、ノードは DB2 の実行コピーであるインスタンスと同じです。

追加のサーバー・オプション

サーバー定義の作成時に、CREATE SERVER ステートメントに追加のサーバー・オプションを指定することができます。サーバー・オプションには、汎用のサーバー・オプションとデータ・ソースに特有のサーバー・オプションがあります。

たとえば、データ・ソースに接続する時に、連合サーバーは、ユーザー ID とパスワードについて大文字小文字の考えられるすべての組み合わせを使用して、接続を試みます。つまり、データ・ソース・サーバーとの接続に成功するまで、サーバーは最高で 9 回まで接続を試みる場合があることを意味します。このような試みは接続時間を長引かせることとなります。FOLD_ID および FOLD_PW サーバー・オプションに値を指定すれば、これを防ぐことができます。

関連タスク:

- 『サーバー』

- 『Selecting server options: Federated Systems help』
- 『Viewing server options: Federated Systems help』
- 『Creating a server: Federated Systems help』

関連資料:

- *SQL* リファレンス 第 2 巻 の『CREATE SERVER ステートメント』
- 連合システム・ガイド の『連合オブジェクトのネーミング規則』

データ・タイプ・マッピング

連合サーバーがデータ・ソースからデータを検索するには、データ・ソース側のデータ・タイプが、対応する DB2® のデータ・タイプと対応付けられて (マッピングされて) いなければなりません。ほとんどのデータ・ソースの場合、ラッパー内にデフォルトのタイプ・マッピングがあります。DB2 データ・ソース用のデフォルトのタイプ・マッピングは、DRDA® ラッパー内にあります。Informix™ 用のデフォルトのタイプ・マッピングは INFORMIX ラッパーにあり、他も同様です。

非リレーショナルのデータ・ソースの場合、CREATE NICKNAME ステートメントでデータ・タイプ情報を指定しなければならないものがあります。

ニックネームの作成時に、データ・ソース・オブジェクトの列ごとに、対応する DB2 (UNIX® および Windows® 版) データ・タイプを指定する必要があります。それぞれの列は、データ・ソース・オブジェクト内の特定のフィールドまたは列にマップされている必要があります。

たとえば、以下ようになります。

- Oracle タイプ FLOAT は、デフォルトで DB2 タイプ DOUBLE にマップされます。
- Oracle タイプ DATE は、DB2 タイプ DB2 TIMESTAMP にマップされます。
- DB2 (z/OS™ 版) のタイプ DATE は、デフォルトで DB2 タイプ DATE にマップされます。

データ・ソース列からの値が連合データベースに戻されると、その値は、データ・ソースの列が対応付けられた (マップされた) DB2 データ・タイプに完全に適合します。これがデフォルトのマッピングである場合、値はマッピング内のデータ・ソース・タイプにも完全に適合します。たとえば、FLOAT 列を持つ Oracle 表が連合データベースに定義されているとします。この列には、Oracle FLOAT から DB2 DOUBLE へのデフォルト・マッピングが自動的に適用されます。したがって、その列から戻される値は、FLOAT と DOUBLE の両方に完全に適合します。

いくつかのラッパーについては、戻される値のフォーマットまたは長さを変更することができます。これは、値が適合すべき DB2 データ・タイプを変更して行います。たとえば、Oracle データ・タイプ DATE はタイム・スタンプとして使用されます。Oracle DATE データ・タイプには、世紀、年、月、日、時、分、秒が含まれます。デフォルト

により、Oracle DATE データ・タイプは DB2 TIMESTAMP データ・タイプにマップします。ここで、いくつかの Oracle 表の列のデータ・タイプが DATE であるとし、これらの列の照会では、時、分、秒だけを戻したいとします。この場合、デフォルトのデータ・タイプ・マッピングをオーバーライドし、Oracle DATE データ・タイプが DB2 TIME データ・タイプにマップされるようにできます。Oracle DATE 列が照会されると、タイム・スタンプ値の時間の部分だけが DB2 に戻されます。

以下のものを作成するには、CREATE TYPE MAPPING ステートメントを使用します。

- デフォルトのデータ・タイプ・マッピングをオーバーライドするデータ・タイプ・マッピング
- 現在、マッピングが存在しないデータ・タイプ・マッピング。たとえば、データ・ソース側で新しい組み込みタイプが使用可能になった場合、またはデータ・ソース側にマップしたいユーザー定義タイプがある場合。

CREATE TYPE MAPPING ステートメントで、データ・ソースのすべてのサーバーにマッピングが適用されるのか、または特定のサーバーにマッピングが適用されるのかを指定することができます。

データ・ソース・ニックネームを既に作成している場合は、タイプ・マッピングの変更には、以下の 2 つの選択があります。

- マッピングを適用したい各ニックネームを変更できます。
- ニックネームをドロップして、必要なタイプ・マッピングを変更し、ニックネームを再作成します。

特定のデータ・ソース・オブジェクトの特定の列のデータ・タイプ・マッピングを変更する場合は、ALTER NICKNAME ステートメントの列オプション・パラメーターを使用します。このステートメントにより、個別の表、ビュー、またはその他のデータ・ソース・オブジェクト用のデータ・タイプ・マッピングを指定できます。

タイプ・マッピングを変更しても、タイプ・マッピングを変更する前に作成されたニックネームには新しいマッピングは反映されません。

サポートされないデータ・タイプ:

DB2 連合サーバーは以下のデータ・タイプをサポートしません。

- DATALINK
- データ・ソース側で作成されたユーザー定義のデータ・タイプ (UDT)

これらのデータ・タイプにユーザー定義のマッピングを作成することはできません。ただし、ユーザー定義データ・タイプを含む表と同一の、データ・ソースのビューにニックネームを作成することができます。このビューは、ユーザー定義タイプの列を、組み込みタイプまたはシステム・タイプに「キャスト」する必要があります。

ニックネームは、LONG VARCHAR 列を含むリモートの表に対して作成することができます。ただし、結果はローカルの DB2 データ・タイプ CLOB にマップされます。

関連概念:

- 連合システム・ガイド の『ラッパーの変更』

関連タスク:

- 456 ページの『デフォルトのデータ・タイプ・マッピングの変更』

関連資料:

- SQL リファレンス 第 2 巻 の『ALTER NICKNAME ステートメント』
- SQL リファレンス 第 2 巻 の『CREATE TYPE MAPPING ステートメント』
- 1381 ページの『デフォルトの順方向データ・タイプ・マッピング』

透過 DDL を使用したリモート表の作成および変更

透過 DDL は、PASSTHRU セッションを使用する必要なく、DB2 でリモート表を作成、変更できる機能を提供します。

透過 DDL でユーザーが使用する SQL ステートメントは、CREATE TABLE、ALTER TABLE、および DROP TABLE です。

透過 DDL の CREATE TABLE ステートメントは、同一のトランザクションで、データ・ソース側にリモート表を作成し、さらに連合サーバー側にその表のニックネームを作成します。このステートメントは、デフォルトの逆方向タイプ・マッピングを使用して、ユーザーが指定する DB2 のデータ・タイプをリモートのデータ・タイプにマップします。ほとんどのデータ・ソースの場合、ラッパー内にデフォルトのタイプ・マッピングがあります。DB2 ファミリーのデータ・ソース用のデフォルトのタイプ・マッピングは、DRDA ラッパーにあります。Informix 用のデフォルトのタイプ・マッピングは INFORMIX ラッパーなどにあります。たとえば、Oracle SQLNET ラッパーを使用しているときに、DB2 の CLOB データ・タイプを指定すると、そのデータ・タイプは、Oracle の LONG データ・タイプにマップされます。

透過 DDL を使用する利点は、DB2 データベース管理者が、慣れた手順にしたがって、ローカル表とリモート表の両方を作成できる点にあります。透過 DDL を使用すると、表管理を集中化でき、許可の付与が容易になります。

透過 DDL は、以下のデータ・ソースでサポートされています。

- DB2 for z/OS and OS/390
- DB2 for iSeries
- DB2 for UNIX and Windows
- DB2 Server for VM and VSE
- Informix IDS

- Microsoft SQL Server
- Oracle
- Sybase

データベース管理者は、DB2 コントロール・センターを使用するか、DB2 コマンド行プロセッサ (CLP) で DDL ステートメントを使用して、表を作成できます。透過 DDL を使用すると、各データ・ソースで必要とされる異なる DDL 構文を学ぶ必要がなくなります。

推奨事項: リモート表を作成するには、DB2 コントロール・センターのリモート表ウィザードを使用してください。

前提条件:

データ・ソースでリモート表を作成するには、その前に、以下のようにそのデータ・ソースに構成をセットアップする必要があります。

- そのデータ・ソース・タイプに対して、ラッパーを作成する。
- リモート表を置くサーバーに対して、サーバー定義を作成する。
- DB2 サーバーとデータ・ソース・サーバー間のユーザー・マッピングを作成する。

透過 DDL ステートメントの許可 ID が持つ特権には、少なくとも以下の 1 つが含まれている必要があります。

- SYSADM または DBADM 権限
- データベースに対する CREATETAB 権限および表スペースに対する USE 特権に加えて、以下のいずれか。
 - データベースに対する IMPLICIT_SCHEMA 権限 (表の暗黙的または明示的スキーマ名が存在しない場合)
 - スキーマに対する CREATEIN 特権 (表のスキーマ名が既存のスキーマを指す場合)

透過 DDL ステートメントを実行するには、許可 ID がニックネームに必要な特権 (連合データベースが要求を受け入れるための) およびリモート・データ・ソース・サーバーに同等の特権 (データ・ソースが要求を受け入れるため) を持っている必要があります。

制限:

透過 DDL には以下の制限があります。

- マテリアライズ照会表は、リモート・データ・ソースには作成できない。
- 表の定義に基本的な列情報を指定できるが、表オプションまたは列オプションを指定することはできない。たとえば、LOB オプション (LOGGED および COMPACT) はサポートされていません。
- 列にコメントを指定することはできない。

- 列の内容を生成することはできない。
- 主キーを指定することはできるが、外部キーやチェック制約を指定することはできない。主キーに使用される列は、NOT NULL でなければならず、LOB を含む列にはできません。
- データ・タイプやデータ長などの、既存の列のパラメーターを変更することはできない。
- もともとリモート・データ・ソースで作成された表を変更またはドロップすることはできない。
- CREATE TABLE および ALTER TABLE ステートメントでは DEFAULT 文節をサポートしません。

CREATE TABLE、ALTER TABLE、および DROP TABLE ステートメントの完全な構文については、「DB2 SQL リファレンス」を参照してください。

透過 DDL を使用した新規リモート表の作成

透過 DDL を使用してリモート表を作成するには、DB2 コントロール・センター・ウィザードか CREATE TABLE ステートメントのどちらかを使用できます。サポートされていないパラメーターやオプションの指定を避けるには、DB2 コントロール・センターのリモート表ウィザードを使用します。ウィザードを使用すると、定義済み列のリストから選択したり、新規列に属性を指定したりすることによって、列を指定できます。

リモート表の作成の前に、そのデータ・ソースにアクセスするように連合サーバーを構成する必要があります。この構成には、データ・ソースのラッパーの作成、リモート表を置くサーバーのサーバー定義の指定、DB2 とデータ・ソース・サーバー間のユーザー・マッピングの作成が含まれます。

リモート・データ・ソースは、CREATE TABLE ステートメント内の列データ・タイプおよび主キー・オプションをサポートする必要があります。たとえば、リモート・データ・ソースが主キーをサポートしないとします。データ・ソースの、サポートしていない要求への応答方法によって、エラーが返されるか、または要求が無視される可能性があります。

リモート・サーバーは、OPTIONS 文節に指定する必要があります。作成される表のリモート名またはリモート・スキーマをオーバーライドするには、OPTIONS 文節を使用できます。

Oracle サーバー上に表 EMPLOYEE を作成したいと仮定します。CREATE TABLE ステートメントで、各列を指定するときに DB2 データ・タイプを使用します。表を作成するために使用する CLP の構文は、以下のとおりです。

```
CREATE TABLE EMPLOYEE
( EMP_NO      CHAR(6) NOT NULL,
  FIRSTNAME   VARCHAR(12) NOT NULL,
  MIDINT      CHAR(1) NOT NULL,
  LASTNAME    VARCHAR(15) NOT NULL,
```



```
HIREDATE    DATE,  
JOB         CHAR(8),  
SALARY      DECIMAL(9,2),  
PRIMARY KEY (EMP_NO )  
OPTIONS (REMOTE_SERVER 'ORASERVER',  
REMOTE_SCHEMA 'J15USER1', REMOTE_TABNAME 'EMPLOY' )
```

ここで、

EMPLOYEE

ローカル表名です。この名前は、表に関連したニックネームにも使用されま
す。

REMOTE_SERVER 'ORASERVER'

CREATE SERVER ステートメントでサーバーに指定した名前です。この値は
大文字小文字の区別があります。

REMOTE_SCHEMA 'J15USER1'

リモート・スキーマ名です。このパラメーターはオプションですが、スキーマ
名を指定することをお勧めします。このパラメーターが指定されていないと、
英大文字のローカル AUTHID がリモート・スキーマ名に使用されます。この
値は大文字小文字の区別があります。

REMOTE_TABNAME 'EMPLOY'

リモート表名です。このパラメーターはオプションです。このパラメーターが
指定されていないと、ローカル表名がリモート表名に使用されます。この値
は、データ・ソースで有効な名前、既存の表名であってはなりません。この
値は大文字小文字の区別があります。

リモート表が、透過性 DDL を使用して DB2 で作成された場合、その他のアクション
がいくつか発生します。

- リモート表に対して、ニックネームが自動的に作成されます。ニックネームの名前
は、リモート表と同じです。リモート表のニックネームは、他のニックネームと同様
に使用できます。加えて、リモート表を ALTER および DROP できます (これは
CREATE NICKNAME で作成されたニックネームではできません)。
- SYSTABOPTIONS カタログ・ビューに、オプション名 TRANSPARENT と値 YES
が付いた行が追加されます。

上の例では、DB2 は DB2 データ・タイプから Oracle データ・タイプにマップするの
に、逆方向データ・タイプ・マッピングを使用します。リモート Oracle サーバーで
は、EMPLOY 表は Oracle データ・タイプを使用して作成されます。以下の表に、例
に指定された列の、DB2 データ・タイプから Oracle データ・タイプへのマッピングを
示します。

表 144. DB2 から Oracle への逆方向データ・タイプ・マッピングの例

列	CREATE TABLE ステートメント に指定された DB2 データ・タイプ	リモート表で 사용되는 Oracle データ・タイプ
EMP_NO	CHAR(6) NOT NULL	CHAR(6) NOT NULL
FIRST_NAME	VARCHAR(12) NOT NULL	VARCHAR2(12) NOT NULL
MID_INT	CHAR(1) NOT NULL	CHAR(1) NOT NULL
LAST_NAME	VARCHAR(15) NOT NULL	VARCHAR2(15) NOT NULL
HIRE_DATE	DATE	DATE
JOB	CHAR(8)	CHAR(8)
SALARY	DECIMAL(9,2)	NUMBER(9,2)

注: SQL_SUFFIX オプションは、CREATE TABLE ステートメントの末尾で使用できません。このオプションは主に、DB2 ファミリー・データ・ソースでリモート表を作成するときに、IN DATABASE 文節を指定するのに使用されます。

逆方向データ・タイプ・マッピングの完全なリストは、別のトピックで取り上げられています。

透過 DDL を使用して作成されたリモート表の変更

透過 DDL を使用して、DB2 で作成した表を変更するには、ALTER TABLE ステートメントを使用します。ALTER TABLE ステートメントを使用すると、以下のことができます。

- 新規列の追加
- 表主キーの追加

ただし、ALTER TABLE ステートメントを使用して行うことのできる変更に関して、透過 DDL には以下のようないくつかの制約があります。

- 既存の主キーは、リモート表内で変更またはドロップすることはできない。
- リモート表の変更を行うと、リモート表に関連したニックネームに依存しているパッケージはいずれも無効になる。
- リモート・データ・ソースは、ALTER TABLE ステートメント内の変更をサポートする必要があります。たとえば、リモート・データ・ソースが主キーをサポートしないとします。データ・ソースの、サポートしていない要求への応答方法によって、エラーが返されるか、または要求が無視される可能性があります。
- 透過 DDL を使用して作成されたローカル表およびリモート表のみ変更できる。

透過 DDL を使用してリモート表を変更するには、DB2 コントロール・センターか ALTER TABLE ステートメントのどちらかを使用できます。サポートされていないパラメーターやオプションの指定を避けるには、DB2 コントロール・センターを使用します。

列オプションを追加または変更するのに、ALTER TABLE ステートメントを使用しないでください。代わりに ALTER NICKNAME ステートメントを使用してください。

透過 DDL を使用して作成したリモート表 EMPLOYEE に主キーを追加したいと仮定します。CLP を使用して表を変更する際の構文は以下のとおりです。

```
ALTER TABLE EMPLOYEE  
ADD PRIMARY KEY (EMP_NO, WORK_DEPT)
```

注: 主キーに使用される列は、NOT NULL でなければならず、LOB を含む列にはできません。

透過 DDL を使用して作成したリモート表 SPALTEN に、列 ORDER_DATE と列 SHIP_DATE を追加したいと仮定します。CLP を使用して表を作成する際の構文は以下のとおりです。

```
ALTER TABLE SPALTEN  
ADD COLUMN ORDER_DATE DATE  
ADD COLUMN SHIP_DATE DATE
```

透過 DDL を使用して作成されたリモート表のドロップ

透過 DDL を使用して DB2 で作成されたリモート表をドロップするには、DB2 コントロール・センターまたは DROP ステートメントのどちらかを使用できます。透過 DDL を使用して作成されたリモート表のニックネームをドロップすると、単にその表のローカル・ニックネームがドロップされます。DROP NICKNAME ステートメントでは、リモート表はドロップしません。リモート表をドロップするには、DROP TABLE ステートメントを使用する必要があります。

リモート表を先にドロップすると、リモート表が削除され、それからリモート表のニックネームが削除されます。ニックネームを削除すると、そのニックネームを基にしたパッケージが無効になります。

表 SPALTEN をドロップするための構文は、以下のとおりです。

```
DROP TABLE SPALTEN
```

SPALTEN はリモート表のローカル名です。

関連概念:

- 連合システム・ガイド の『データ・ソースの構成の近道』

関連タスク:

- 『リモート表を変更する』
- 『リモート表の列の変更』
- 『リモート表を作成する』

関連資料:

- 1402 ページの『ALTER TABLE』

- 1484 ページの『CREATE TABLE』
- 1572 ページの『DROP』
- 427 ページの『デフォルトの逆方向データ・タイプ・マッピング』

デフォルトのデータ・タイプ・マッピングの変更

データ・ソース表にニックネームを作成すると、連合サーバーは、表についての情報をグローバル・カタログに入れます。この情報には、ニックネーム、データ・ソースの表名、表のそれぞれの列に定義された列名およびデータ・タイプが含まれます。

データ・ソースのデータ・タイプはリモート・データ・タイプと呼ばれ、連合データベースのデータ・タイプはローカル・データ・タイプと呼ばれます。

データ・ソースのデータ・タイプと連合データベースのデータ・タイプ間のマッピングには、順方向タイプのマッピングと逆方向タイプのマッピングの 2 種類があります。順方向タイプ・マッピングでは、マッピングはリモートのタイプから対応するローカル・タイプへのマッピングです。マッピングのもう一方のタイプは逆方向タイプ・マッピングで、これは透過 DDL で使用されます。逆方向タイプ・マッピングでは、マッピングはローカル・タイプから対応するリモート・タイプへのマッピングです。

DB2 はデータ・タイプ・マッピングを使用して、データ・ソース・オブジェクトの列に、DB2 でサポートされるどのようなデータ・タイプを定義すべきかを判別します。デフォルトのデータ・タイプ・マッピングは、データ・ソース・ラッパーに組み込まれています。

しかし、アプリケーションによっては、デフォルトのマッピングとは異なるデータ・タイプ・マッピングを必要とする場合があります。以下のようにデフォルトのマッピングをオーバーライドすることができます。

- 特定のサーバーにあるすべてのデータ・ソース・オブジェクトについて、タイプ・マッピングを変更する
- 特定のデータ・ソース・オブジェクトのタイプ・マッピングの変更
- 特定のデータ・ソース・タイプのタイプ・マッピングの変更
- 特定のデータ・ソース・タイプおよびバージョンのためのタイプ・マッピングの変更

新しいデータ・タイプ・マッピングを定義するには、CREATE TYPE MAPPING ステートメントを使用します。作成したマッピングは、連合データベースのグローバル・カタログ SYSSTAT.TYPEMAPPING ビューに保管されます。

データ・タイプ・マッピングの変更は、データ・ソース・オブジェクトのニックネームを作成する前に行います。データ・ソース・オブジェクトにニックネームを作成すると、連合サーバーは、列名および、それらの列のデータ・タイプをグローバル・カタログに入れます。マッピングが変更された後に作成されたニックネームだけが、新しいタイプ・マッピングを反映します。マッピングを変更する前に作成されたニックネーム

は、デフォルトのデータ・タイプ・マッピングを反映します。既存のニックネームを更新するには、新しいマッピングを反映するように各ニックネームを変更するか、またはニックネームをドロップして再作成する必要があります。

順方向タイプ・マッピング:

順方向タイプ・マッピングは、ローカル DB2 タイプからリモートの (データ・ソースの) データ・ソースへのマッピングです。順方向タイプ・マッピングは、ユーザーが連合データベースにサブミットする照会に、そのデータ・ソースのニックネームが含まれている場合に使用されます。

CREATE TYPE MAPPING ステートメントの構文は、DB2 (UNIX および Windows 版) バージョン 8 で更新されました。順方向タイプ・マッピングの作成には以下の 2 つのフォーマットを使用できます。

```
CREATE TYPE MAPPING type_mapping_name FROM LOCAL TYPE local_data_type
TO remote_server REMOTE TYPE data_source_data_type
```

```
CREATE TYPE MAPPING type_mapping_name TO remote_server
REMOTE TYPE data_source_data_type FROM LOCAL TYPE local_data_type
```

CREATE TYPE MAPPING ステートメントには、TO および FROM キーワードの両方が必要です。タイプが短い形式と長い形式 (たとえば、CHAR と CHARACTER) を持つ場合、短い形式を指定する必要があります。

逆方向タイプ・マッピング:

逆方向タイプ・マッピングは、透過 DDL でのみ使用されます。透過 DDL の一部として、データ・ソースのデータ・タイプから DB2 のタイプにマッピングを設定します。逆方向タイプ・マッピングは、透過 DDL を使用して、リモートの表およびそのリモート表のニックネームを作成する場合にのみ使用します。

CREATE TYPE MAPPING ステートメントの構文は、DB2 (UNIX および Windows 版) バージョン 8 で更新されました。逆方向タイプ・マッピングの作成には以下の 2 つのフォーマットを使用できます。

```
CREATE TYPE MAPPING type_mapping_name FROM remote_server
REMOTE TYPE data_source_data_type TO LOCAL TYPE local_data_type
```

```
CREATE TYPE MAPPING type_mapping_name TO LOCAL TYPE local_data_type
FROM remote_server REMOTE TYPE data_source_data_type
```

CREATE TYPE MAPPING ステートメントには、TO および FROM キーワードの両方が必要です。タイプが短い形式と長い形式 (たとえば、CHAR と CHARACTER) を持つ場合、短い形式を指定する必要があります。

サポートされないデータ・タイプ:

連合サーバーはすべての DB2 データ・タイプをサポートするわけではありません。以下のタイプは `local_data_type` に指定できません。

- DATALINK
- ユーザー定義データ・タイプ

`data_source_data-type` はユーザー定義タイプにはできません。

ただし、代わりに以下のものを使用できます。

- ユーザー定義タイプ。このデータ・タイプにユーザー定義タイプ・マッピングを作成することはできません。ただし、データ・ソースのビューがユーザー定義のデータ・タイプを含む表と同一の場合、そのビューにニックネームを作成することができます。このビューは、ユーザー定義タイプの列を、組み込みタイプまたはシステム・タイプに「キャスト」する必要があります。この代替方法の欠点は、そのビューに統計または索引を持たないことです。

注: ニックネームは、LONG VARCHAR 列を含むリモートの表に対して作成することができます。ただし、結果はローカルの DB2 データ・タイプ CLOB にマップされます。

関連概念:

- 連合システム・ガイド の『連合データベースに代替データ・タイプ・マッピングを定義する』
- 448 ページの『データ・タイプ・マッピング』

関連タスク:

- 連合システム・ガイド の『特定のサーバーにあるすべてのデータ・ソース・オブジェクトのタイプ・マッピングの変更』
- 連合システム・ガイド の『特定のデータ・ソース・オブジェクトのタイプ・マッピングの変更』
- 連合システム・ガイド の『特定のデータ・ソース・タイプのタイプ・マッピングの変更』
- 連合システム・ガイド の『特定のデータ・ソース・タイプおよびバージョンのためのタイプ・マッピングの変更』

関連資料:

- 1381 ページの『デフォルトの順方向データ・タイプ・マッピング』
- 427 ページの『デフォルトの逆方向データ・タイプ・マッピング』

Microsoft SQL Server データ・ソースを連合サーバーに追加する

Microsoft SQL Server データ・ソースにアクセスするように連合サーバーを構成するには、アクセスしたい Microsoft SQL Server データ・ソースとオブジェクトの情報を連合サーバーに提供します。Microsoft SQL Server データ・ソースへのアクセスを構成するには以下の 2 つの方法があります。

- DB2 コントロール・センターを使用
- DB2 コマンド・センターまたはコマンド行プロセッサ (CLP) を使用

DB2 コントロール・センターを使用した場合の利点は、各ステートメントやコマンドをキー入力する必要がないことです。これは、Microsoft SQL Server データ・ソースへのアクセスを素早く構成できる最も簡単な方法です。DB2 コントロール・センターでは実行できない以下のような構成作業も少しあります。

- サーバー定義とユーザー・マッピングを検証するための、Microsoft SQL Server サーバーへの接続をテストする
- 列オプションの追加またはドロップ。

このセクションのステップは、DB2 コマンド・センターまたは DB2 コマンド行プロセッサ (CLP) を使用して、Microsoft SQL Server データ・ソースへのアクセスを構成すると想定しています。

前提条件:

- Microsoft SQL Server データ・ソースにアクセスするようにセットアップされた連合サーバーおよびデータベース
- Microsoft SQL Server ODBC ドライバーが連合サーバー上にインストール済みで、構成済みであること。
- 適切な変数セットアップ。これには、システム環境変数、db2dj.ini 変数 (UNIX のみ)、および DB2 Profile Registry (db2set) 変数が含まれます。

これらの作業を行うためのステップは、に説明があります。連合サーバーおよびデータベースのセットアップに説明があります。

手順:

連合サーバーに Microsoft SQL Server データ・ソースを追加するには、以下のようになります。

1. 連合サーバーと連合データベースを準備する。
 - Windows では、ODBC システム DSN が適切にセットアップされていることを確認し、Microsoft SQL Server リモート・サーバーへの接続をテストする。
 - UNIX では、odbc.ini ファイルを更新または作成し、Microsoft SQL Server リモート・サーバーへの接続をテストする。
2. ラッパーを作成する。
3. サーバー定義を作成し、サーバー・オプションをセットする。

4. ユーザー・マッピングを作成する。
5. Microsoft SQL Server リモート・サーバーへの接続をテストする。
6. Microsoft SQL Server の表およびビューのニックネームを作成する。

これらのステップについては、このセクションで詳細に説明しています。オペレーティング・システムに特有の差異については、その都度、記しています。

ステップ 1: 連合サーバーとデータベースを準備する

Microsoft SQL Server データ・ソースにアクセスするための連合サーバーとデータベースの準備のステップは、連合サーバーで稼働しているオペレーティング・システムにより異なります。

Windows の場合:

ODBC システム DSN が Microsoft SQL Server データ・ソースに接続するように設定されているか、検証することが重要です。この設定は、「コントロール」パネルで以下のようにしてチェックできます。

1. 「スタート」メニューからコントロール パネルを開きます。
2. 「ODBC データ ソース」をダブルクリックし、「ODBC データ ソース アドミニストレータ」ウィンドウを表示します。
3. 「システム DSN」タブをクリックし、ODBC ドライバー用に定義したシステム DSN がリストに表示されることを確認します。Microsoft SQL Server データ・ソースのノード名が、システム DSN に定義されている必要があります。

接続をテストし、ODBC システム DSN が Microsoft SQL Server データ・ソースに接続できることを確認してください。「ODBC データ ソース アドミニストレータ」ウィンドウで「構成」を選択すると、接続をテストすることができます。あるいは、Microsoft SQL Server の照会ツールを使用して、接続をテストすることもできます。

UNIX の場合:

連合サーバー上で `odbc.ini` ファイルが更新されている (または必要な場合、作成されている) ことを確認してください。 `odbc.ini` ファイルまたはそのコピーを、DB2 インスタンス所有者のホーム・ディレクトリーに置くことをお勧めします。 `odbc.ini` へのパスが `ODBCINI` 環境変数にあることを確認します。

次にリストされたオペレーティング・システムでは、以下のシンボリック・リンクが作成されていていつことを確認してください。

HP-UX `/usr/exe/libodbcinst.sl` から `$DJX_ODBC_LIBRARY_PATH/libodbcinst.sl` へのシンボリック・リンク

Linux `/usr/local/locale` から `$DJX_ODBC_LIBRARY_PATH/./locale` へのシンボリック・リンク

Solaris オペレーティング環境

\$HOME/sql/lib/locale から \$DJX_ODBC_LIBRARY_PATH/./locale へのシンボリック・リンク。\$HOME は DB2 インスタンス所有者のホーム・ディレクトリーです。

接続をテストし、連合サーバーが Microsoft SQL Server データ・ソースに接続できることを確認してください。接続をテストするには、DataDirect Connect ODBC **demoodbc** ツールを使用します。このツールは、Connect ODBC ライブラリーの /demo サブディレクトリーにあります。

ステップ 2: ラッパーを作成する

Microsoft SQL Server データ・ソースにアクセスするために使用するラッパーを指定するには、CREATE WRAPPER ステートメントを使用します。DB2 Relational Connect には、Microsoft SQL Server 用に 2 つのラッパーが組み込まれています。

表 145. サポートされる ODBC ドライバーと、デフォルトのラッパー名

連合サーバーの オペレーティング・システム	ODBC ドライバー	ラッパー名
UNIX	DataDirect Connect ODBC 3.7 ドライバ	MSSQLODBC3
Windows	ODBC 3.0 以上のドライバ	DJXMSSQL3

以下の例は、Windows NT および Windows 2000 ラッパー用の CREATE WRAPPER ステートメントを示しています。

```
CREATE WRAPPER DJXMSSQL3
```

推奨事項: デフォルトのラッパー名 (DJXMSSQL3 および MSSQLODBC3) を使用することをお勧めします。デフォルト名の 1 つを使用してラッパーを作成すると、連合サーバーはそのラッパー名に関連するデフォルト・ライブラリー名を自動的に選択します。ラッパー名が連合データベース内の既存のラッパー名と競合した場合、デフォルトのラッパー名をユーザーが指定した名前でも置き換えることができます。デフォルト名とは異なる名前を使用する場合は、CREATE WRAPPER ステートメントに LIBRARY パラメーターを含める必要があります。仮に、AIX で稼働する連合サーバーがあり、デフォルト名とは異なるラッパー名を使用することに決めたとします。この場合、以下のような CREATE WRAPPER ステートメントを使用する必要があります。

```
CREATE WRAPPER mywrapper LIBRARY 'libdb2mssql3.a'
```

ここで *mywrapper* は、デフォルトのラッパー名の代わりに、ユーザーがラッパーに与えた名前です。

Microsoft SQL Server 用のラッパー・ライブラリー名は以下のとおりです。

表 146. Microsoft SQL Server ラッパーのライブラリー名

連合サーバーのオペレーティング・システム	ラッパーのライブラリー名
AIX	libdb2mssql3.a
HP-UX	libdb2mssql3.sl
Linux	libdb2mssql3.so
Solaris オペレーティング環境	libdb2mssql3.so
Windows	db2mssql3.dll

ステップ 3: サーバー定義を作成する

連合データベース内に、アクセスしたいそれぞれの Microsoft SQL Server リモート・サーバーを定義する必要があります。CREATE SERVER ステートメントを使用してサーバー定義を作成します。たとえば、以下のように指定します。

```
CREATE SERVER sqlserver TYPE MSSQLSERVER VERSION 7.0 WRAPPER djxmssql3  
OPTIONS (NODE 'sqlnode', DBNAME 'africa')
```

ここで、

sqlserver

は、Microsoft SQL Server リモート・サーバーに割り当てる名前です。この名前はユニークでなければなりません。重複するサーバー名は使用できません。

TYPE MSSQLSERVER

は、アクセスを構成しているデータ・ソースのタイプです。Microsoft SQL Server ラッパー用の TYPE パラメーターは MSSQLSERVER でなければなりません。

VERSION 7.0

は、アクセスしたい Microsoft SQL Server データベース・サーバーのソフトウェアのバージョンです。サポートされるバージョンは 6.5、7.0 および 2000 です。

WRAPPER djxmssql3

は、CREATE WRAPPER ステートメントで指定したラッパー名です。

NODE 'sqlnode'

は、Microsoft SQL Server リモート・サーバーが存在するノードの名前です。この値は大文字小文字の区別があります。

ノード名は CREATE SERVER ステートメントではオプションとなっていますが、Microsoft SQL Server データ・ソースの場合は必要です。

DBNAME 'africa'

は、アクセスしたいデータベースの名前です。この値は大文字小文字の区別があります。

データベース名は CREATE SERVER ステートメントではオプションとなっていますが、Microsoft SQL Server データ・ソースの場合は必要です。

ノード名の探索: 連合サーバーが Windows NT または Windows 2000 を使用する場合、NODE は、アクセスする Microsoft SQL Server リモート・サーバーに指定したシステム DSN 名です。

連合サーバーが AIX、HP-UX、Linux、または Solaris オペレーティング環境を使用する場合、NODE は .odbc.ini ファイルに定義します。以下の例は、AIX での .odbc.ini ファイルです。 .odbc.ini ファイルの最上部に [ODBC Data Sources] とラベルの付いたセクションがあり、そこにノードがリストされています。各ノードには、各ノードを記述するセクション [node_name] があります。

表 147. .odbc.ini ファイルにあるノード名を探す

オペレーティング・システム	サンプルの .odbc.ini ファイル
AIX	rawilson=MS SQL Server 7.0 medusa=MS SQL Server 7.0 [rawilson] Driver=/djxcclient/mssql/merant/3.7/lib/ivmsss16.so Description=MS SQL Server Driver for AIX Address=9.112.30.39,1433 [medusa] Driver=/djxcclient/mssql/merant/3.7/lib/ivmsss16.so Description=MS SQL Server Driver for AIX Address=9.112.98.123,1433

オプション: 追加のサーバー・オプションの設定: サーバー定義の作成時に、CREATE SERVER ステートメントに追加のサーバー・オプションを指定することができます。サーバー・オプションには、汎用のサーバー・オプションとデータ・ソースに特有のサーバー・オプションがあります。

collating_sequence サーバー・オプションは、データ・ソースが連合サーバーと同じ照合シーケンスを使用しているかどうかを指定します。Windows NT または Windows 2000 で稼働する Microsoft SQL Server データベース・サーバーでは、デフォルトの照合シーケンスは大文字小文字の区別をしません (たとえば、'STEWART' と 'Stewart' は等しいと見なされます)。連合サーバーからの正しい結果を保証するには、COLLATING_SEQUENCE サーバー・オプションを 'I' にしてください。

注: COLLATING_SEQUENCE サーバー・オプションを 'I' にすると、連合サーバーは以下のものをプッシュダウンしません。

- 文字の列に対して ORDER BY を含む照会。
- 文字の列に対して GROUP BY を含む照会。
- 文字の列に対して DISTINCT を含む照会。
- 文字の列に対して WHERE= を含む照会。
- 文字の列に対して WHERE< を含む照会。
- 文字の列に対して WHERE> を含む照会。

これらのサーバー・オプションを指定した CREATE SERVER ステートメントの例を示します。

```
CREATE SERVER sqlserver TYPE MSSQLSERVER VERSION 7.0 WRAPPER djxmssql3
OPTIONS (NODE 'sqlnode', DBNAME 'africa', COLLATING_SEQUENCE 'I')
```

サーバー定義を作成した後、サーバー・オプションを追加またはドロップするには、ALTER SERVER ステートメントを使用します。

ステップ 4: ユーザー・マッピングを作成する

Microsoft SQL Server データ・ソースにアクセスする場合、連合サーバーは、まずデータ・ソースへの接続を確立する必要があります。連合サーバーは、そのデータ・ソースに対して有効なユーザー ID とパスワードを使用して接続を行います。連合サーバーのユーザー ID とパスワードおよびデータ・ソースのユーザー ID とパスワードの間の関連付けを定義しておく必要があります。分散要求を送信するために連合システムを使用するユーザー ID には、この関連付けを必ず作成しておかなければなりません。この関連付けは、ユーザー・マッピングと呼ばれます。

CREATE USER MAPPING ステートメントを使用して、ローカルのユーザー ID を Microsoft SQL Server サーバーのユーザー ID およびパスワードにマップします。たとえば、以下のように指定します。

```
CREATE USER MAPPING FOR elizabeth SERVER sqlserver
OPTIONS (REMOTE_AUTHID 'liz', REMOTE_PASSWORD 'abc123')
```

ここで、

elizabeth

は、Microsoft SQL Server リモート・サーバーで定義されたユーザー ID にマップする、ローカルのユーザー ID です。

SERVER *sqlserver*

は、CREATE SERVER ステートメントで定義した Microsoft SQL Server リモート・サーバーの名前です。

REMOTE_AUTHID *'liz'*

は、*elizabeth* をマップする、Microsoft SQL Server リモート・サーバー側のユ

ーザー ID です。この値は、CREATE SERVER ステートメントで FOLD_ID サーバー・オプションを 'U' または 'L' にしている場合を除き、大文字小文字の区別があります。

REMOTE_PASSWORD 'abc123'

は、'liz' に関連付けられたパスワードです。この値は、CREATE SERVER ステートメントで FOLD_PW サーバー・オプションを 'U' または 'L' にしている場合を除き、大文字小文字の区別があります。

注: DB2 特殊レジスター **USER** を使用して、CREATE USER MAPPING ステートメントを実行した人の許可 ID を、**REMOTE_AUTHID** ユーザー・オプションに指定されたデータ・ソースの許可 ID にマップすることができます。以下の例は、**USER** 特殊レジスターを含む CREATE USER MAPPING ステートメントです。

```
CREATE USER MAPPING FOR USER SERVER sqlserver
OPTIONS (REMOTE_AUTHID 'liz', REMOTE_PASSWORD 'abc123')
```

ステップ 5: Microsoft SQL Server リモート・サーバーへの接続をテストする

サーバー定義および定義したユーザー・マッピングを使用して接続を確立できることを確認するため、Microsoft SQL Server リモート・サーバーへの接続をテストします。パススルー・セッションを開き、Microsoft SQL Server のシステム表に対して SELECT ステートメントを実行します。たとえば、

```
SET PASSTHRU server_name
SELECT count(*) FROM dbo.sysobjects
SET PASSTHRU RESET
```

SELECT がカウントを戻せば、サーバー定義とユーザー・マッピングは適切にセットアップされています。SELECT がエラーを戻した場合、以下のチェックをしてください。

- Microsoft SQL Server リモート・サーバーが開始されていることを確認します。
- Microsoft SQL Server リモート・サーバーを調べ、入ってくる接続に備えた構成がされていることを確認します。
- ユーザー・マッピングを調べ、remote_authid と remote_password のオプションが Microsoft SQL Server リモート・サーバーとの接続に有効な設定になっているかを確認します。
- DB2 連合サーバー上の ODBC ドライバーを調べ、これがインストールされ、Microsoft SQL Server リモート・サーバーに接続するように正しく構成されていることを確認します。
- DB2 連合変数を調べ、Microsoft SQL Server サーバーで作業するための正しい値になっているかを確認します。これには、システム環境変数、db2dj.ini 変数、および DB2 Profile Registry (db2set) 変数が含まれます。
- サーバー定義を調べ、問題があればドロップして再作成してください。

- ユーザー・マッピングを調べ、必要に応じて変更、または再作成してください。

ステップ 6: 表およびビューにニックネームを作成する

連合データベースは、ニックネームが付けられたオブジェクトのカatalog統計に基づいて、照会処理を最適化します。これらの統計は、CREATE NICKNAME ステートメントを使用してデータ・ソース・オブジェクトにニックネームを作成した時に収集されます。連合データベースはデータ・ソース側にオブジェクトが存在するかチェックし、次に既存のデータ・ソース統計データを収集します。オプティマイザーにとって役に立つ情報は、データ・ソース・カatalogから読み取られ、連合サーバー上のグローバル・カatalogに置かれます。オプティマイザーは、データ・ソースのカatalog情報の一部、またはすべてを使用するので、ニックネームを作成する前に、データ・ソース側で (RUNSTATS と同等のデータ・ソース・コマンドを使用して) 統計を更新しておくことをお勧めします。

定義したそれぞれの Microsoft SQL Server リモート・サーバーごとに、これらのサーバー上でアクセスするそれぞれの表またはビューにニックネームを割り当てます。Microsoft SQL Server データ・ソースを照会する時には、データ・ソース・オブジェクトの名前ではなく、これらのニックネームを使用します。ニックネームの長さは 128 文字までです。

連合サーバーは、Microsoft SQL Server サーバー、スキーマ、および表名が二重引用符 (") で囲まれていない場合、これを大文字に変換します。以下の例は、CREATE NICKNAME ステートメントの例です。

```
CREATE NICKNAME cust_africa FOR sqlserver.customers.egypt
```

ここで、

cust_egypt

は、Microsoft SQL Server の表またはビューのユニークのニックネームです。

注: ニックネームは、2 つの部分 (スキーマとニックネーム) からなる名前です。ニックネームの作成時にスキーマを省略すると、そのニックネームのスキーマはニックネームを作成したユーザーの許可 ID になります。

sqlserver.customers.egypt

は、リモート・オブジェクトを表す、3 つの部分からなる ID です。

- *sqlserver* は、CREATE SERVER ステートメントで Microsoft SQL Server データベース・サーバーに割り当てた名前です。
- *customers* は、表またはビューが属するリモート・スキーマの名前です。
- *egypt* は、アクセスしたいリモートの表またはビューの名前です。

ニックネームを作成したい Microsoft SQL Server の表またはビューごとに、このステップを繰り返します。ニックネームを作成すると、DB2 は接続を使用してデータ・ソースのカatalog表を照会します (Microsoft SQL Server では、これらをシステム表と呼びま

す)。この照会は、ニックネームを使用してデータ・ソースへの接続をテストします。接続が働かない場合、エラー・メッセージが出ます。

関連概念:

- *SQL* リファレンス 第 1 巻 の『ユーザー・マッピングおよびユーザー・オプション』
- *SQL* リファレンス 第 1 巻 の『ニックネームとデータ・ソース・オブジェクト』
- *SQL* リファレンス 第 1 巻 の『索引の指定』

関連タスク:

- 連合システム・ガイド の『サーバーとデータベースをセットアップする近道』
- 469 ページの『Microsoft SQL Server データ・ソースにアクセスするサーバーのセットアップ』

関連資料:

- 437 ページの『連合システムのサーバー・オプション』
- *SQL* リファレンス 第 1 巻 の『連合システムのユーザー・オプション』
- *SQL* リファレンス 第 1 巻 の『連合システムのラッパー・オプション』

ラージ・オブジェクト (LOB) データ・タイプの計画

DB2 連合システムは、以下のデータ・ソースの LOB に対する、SELECT 操作をサポートします。

- DB2 (z/OS および OS/390 版)
- DB2 (iSeries 版)
- DB2 (UNIX および Windows 版)
- Informix
- Microsoft SQL Server
- Sybase

DB2 連合システムは、下記のデータ・ソースの LOB に対する、SELECT、INSERT、UPDATE、および DELETE 操作をサポートします。

- Oracle バージョン 8 以上で、NET8 ラッパーを使用

LOB には、以下のようないくつかの制限があります。

- DB2 for iSeries データ・ソースは、LOB を扱う場合は、バージョン 5 以降でなければなりません。
- 連合データベースは、リモート LOB をファイル参照変数にバインドすることはできません。
- LOB は、パススルー・セッションではサポートされません。

アクション: LOB データ・タイプを含むデータ・ソース・オブジェクトを洗い出します。プランニング・チェックリストにオブジェクトと列をリストします。

関連概念:

- 連合システム・ガイド の『連合 LOB サポート』
- 連合システム・ガイド の『LOB ロケーター』
- 連合システム・ガイド の『LOB と LOB 以外のデータ・タイプ間のマッピングの作成』

関連資料:

- 連合システム・ガイド の『連合システムの計画のためのチェックリスト』

連合パススルーの考慮事項および制約事項

パススルー・セッションを使用する場合に、注意すべきいくつかの考慮事項と制約事項があります。以下の考慮事項および制約事項はすべてのデータ・ソースに適用されます。

- パススルー・セッション内で準備されたステートメントは、同じパススルー・セッション内で実行する必要があります。パススルー・セッション内で準備したステートメントを、そのパススルー・セッション外で実行すると、そのステートメントは失敗し、SQLSTATE 56098 エラーになります。
- アプリケーションは複数の SET PASSTHRU ステートメントを出すことができますが、最後のセッションだけがアクティブになります。新しい SET PASSTHRU ステートメントを呼び出すと、直前の SET PASSTHRU ステートメントは終了します。同じパススルー・セッションで複数のデータ・ソースにパススルーすることはできません。
- アプリケーションで複数のパススルー・セッションを使用する場合、別のパススルー・セッションを開く前に、必ず COMMIT を実行してください。これにより、現行セッションの作業単位が終了します。
- パラメーター・マーカーは、パススルー・セッションではサポートされません。パラメーター・マーカーの代わりにホスト変数を使用してください。
- パススルー・セッション内の SQL ステートメントに定義されたホスト変数は、:Hn (H は大文字で、n はユニークの整数) の形式にする必要があります。n の値はゼロから始まり、順番に番号を振る必要があります。
- パススルーは LOB をサポートしません。
- パススルーはストアド・プロシージャ呼び出しをサポートしません。
- パススルーは SELECT INTO ステートメントをサポートしません。

関連概念:

- SQL リファレンス 第 1 巻 の『パススルー・セッション』
- 連合システム・ガイド の『パススルーによりデータ・ソースを直接照会する』

関連タスク:

- 連合システム・ガイド の『Oracle データ・ソースへのパススルー・セッション』
- 連合システム・ガイド の『PASSTHRU を使用してデータ・ソースにアクセスする』

関連資料:

- *SQL* リファレンス 第 2 巻 の『SET PASSTHRU ステートメント』

Microsoft SQL Server データ・ソースにアクセスするサーバーのセットアップ

Microsoft SQL Server データ・ソースにアクセスしたい場合は、連合サーバーとして働くサーバー上に、以下のソフトウェアをインストールする必要があります。

- 適切な ODBC ドライバーとして、
 - UNIX では、DataDirect Technologies Connect ODBC バージョン 3.7 または 4.0 ドライバー
 - Windows では、Microsoft SQL Server Client バージョン 2000.8 ドライバー
- DB2 Universal Database Enterprise Server Edition、バージョン 8
- DB2 Relational Connect、バージョン 8

前提条件:

セットアップを開始する前に、システムが、インストール、メモリー、およびディスクの要件を満たすことを確認してください。さらに、以下のことも確認してください。

- UNIX では、DB2 製品 CD-ROM をシステムにマウントする必要があります。CD をマウントするステップについては、「*DB2 Universal Database* サーバー機能 概説 およびインストール」を参照してください。
- Windows の場合、Windows 2000 または Windows .NET 上で LDAP を使用して、DB2 サーバーを Active Directory に登録する予定であれば、DB2 サーバー・ソフトウェアをインストールする前にディレクトリー・スキーマを拡張する必要があります。

制限:

UNIX では、インストールを実行するには root 権限が必要です。

Windows では、インストールを実行するには、推奨されたユーザー権限を持つ、ローカル管理者ユーザー・アカウントを持っている必要があります。

手順:

Microsoft SQL Server データ・ソース用の連合サーバーをセットアップするステップは、以下のとおりです。

1. システムにログオンします。

- UNIX では、root 権限を持つユーザー ID を使用してログオンします。
 - Windows では、DB2 インストール用に定義した管理者アカウントを使用してログオンします。
2. DB2 インストール・プログラムが必要に応じてファイルを更新できるように、オープンされているプログラムをすべてクローズします。
 3. DB2 連合サーバーとして働くサーバー上に、ODBC ドライバーをインストールし、構成します。
 - UNIX では、Connect ODBC 3.7 ドライバーをインストールし、Microsoft SQL Server への接続をテストします。
 - Connect ライブラリー・ディレクトリーを、LIBPATH 内の最初のエントリーとして指定します。
 - Connect ODBC ライブラリーの許可をチェックし、Connect ODBC ライブラリーを他のユーザーが使用できるようにします。
 - Connect ODBC **demoodbc** テスト・ツールを使用して、.odbc.ini の構成および Microsoft SQL Server データ・ソースへの接続をテストします。
demoodbc テスト・ツールは、Connect ODBC の /demo サブディレクトリーにあります。 **demoodbc** テスト・ツールは、要求された SQL Server データ・ソースに接続し、EMP 表の照会を試みます。 Microsoft SQL Server データ・ソースに EMP 表があるはずがないため、エラー・メッセージが戻されるはずです。以下のようなになれば、テストは成功です。
 - EMP 表はないというメッセージが示される。
 - EMP 表からのレコードが戻される。
 - EMP 表はあるが、要求された列がないというメッセージが示される。

demoodbc テスト・ツールは、UNIX システムで root 権限以外のユーザーから実行する必要があります。システムに他のユーザーがない場合、root 権限を持つユーザーは DB2 インスタンス用のグループとユーザー ID を作成することができます。そのユーザー ID を使用して、**demoodbc** テスト・ツールを実行します。たとえば、ルートはグループ *db2admin1* およびユーザー *db2inst1* を作成することができます。これはインスタンスを作成するのではなく、インスタンス所有者となる新しいユーザー ID を追加します。 **demoodbc** を実行するには、*db2inst1* ユーザーは以下の作業をする必要があります。

- Connect ODBC /lib サブディレクトリーを LIBPATH システム環境変数値に追加します。通常、ディレクトリーは /opt/odbc/lib であり、以下のコマンドにより設定することができます。

```
export LIBPATH=/opt/odbc/lib:$LIBPATH
```

- SQL サーバー・データ・ソース用の ODBC 接続情報を持つ、odbc.ini ファイルのロケーションを指すように ODBCINI システム環境変数を設定します。 ODBCINI システム環境変数を設定するには、エクスポート・コマンドを使用します。たとえば、odbc.ini ファイルのロケーションが、DB2 イン

スタンス所有者 *db2inst1* のホーム・ディレクトリーであり、連合サーバーのオペレーティング・システムが AIX の場合、コマンドは以下のようになります。

```
export ODBCINI=/home/db2inst1/.odbc.ini
```

- Windows の場合、Windows をインストールする時に、Microsoft SQL Server Client バージョン 2000.8 ドライバーをインストールする必要があります。このドライバーがインストールされていることを確認し、Microsoft SQL Server データ・ソースにアクセスするようにドライバーを構成する必要があります。
 - ドライバーがインストールされていることを確認します。Windows のコントロール パネルから、「Microsoft ODBC データ ソース アドミニストレータ」にアクセスします。「**ドライバ**」タブをクリックし、2000.8 ドライバーがインストールされていることを確認します。
 - Microsoft SQL Server データ・ソースを System DSN として登録します。「**構成**」ボタンを使用して、Microsoft SQL Server データ・ソースへの接続をテストします。

注: Microsoft SQL Server 2000 Personal Edition を使用する場合は、SQL Server Client Network Utility を使用して、新しい SQL Server ODBC データ・ソースを ODBC System DSN リストに追加する必要があります。

ドライバーをインストールし、構成する詳細な方法については、ODBC ドライバーと一緒に配布されるドキュメンテーションにある、インストール手順を参照してください。

4. DB2 CD を挿入し、セットアップ・プログラム (DB2 セットアップ・ウィザード) を開始して、DB2 サーバー・ソフトウェアをインストールします。
 - UNIX で、DB2 CD を CD-ROM に挿入し、マウントします。CD-ROM がマウントされたディレクトリーに変更します。**./db2setup** コマンドを入力し、DB2 セットアップ・ウィザードを開始します。
 - Windows では、CD-ROM をドライブに挿入します。自動実行機能が DB2 セットアップ・ウィザードを自動的に開始します。セットアップ・プログラムが自動開始に失敗した場合は、手動で DB2 セットアップ・ウィザードを開始することができます。

DB2 セットアップ・ウィザードを手動で開始するには、「**スタート**」をクリックし、「**ファイル名を指定して実行**」オプションを選択します。「**名前**」フィールドに、*x:\setup* (*x*: は CD-ROM ドライブ) と入力します。それから、「**OK**」をクリックします。
5. 「DB2 セットアップ・ランチパッド (DB2 Setup Launchpad)」が開きます。このウィンドウで、インストールの前提条件と最新のリリース情報を確認してください。
6. DB2 セットアップ・ウィザードのインストール・パネルに沿って進み、必要な選択を行います。

注: インストールの一部として、DB2 インスタンスを作成しないでください。インスタンスは、DB2 Relational Connect をインストールする時に作成します。

インストール操作のヘルプを使用すれば、ステップに沿ってガイドしてくれます。インストール操作のヘルプを呼び出すには、「ヘルプ」をクリックするか「F1」キーを押します。「キャンセル (Cancel)」をクリックすれば、いつでもインストールを終了できます。

7. DB2 セットアップ・ウィザードの最後のインストール・パネルで「終了 (Finish)」をクリックし、DB2 ファイルをシステムにコピーします。

インストールを完了すると、DB2 は、以下のディレクトリーのいずれかに (オペレーティング・システムによって異なる) インストールされます。

`/usr/opt/db2_08_01 (AIX)`

`/opt/IBM/db2/V8.1 (HP-UX、Linux、Solaris オペレーティング環境)`

`\Program Files\IBM\SQLLIB (Windows)`

クライアント・ソフトウェアと DB2 サーバー・ソフトウェアをインストールした後、DB2 Relational Connect バージョン 8 を DB2 サーバーにインストールする必要があります。DB2 Relational Connect には、Microsoft SQL Server データ・ソースにアクセスするために必要なソフトウェアが含まれています。

1. システムにログオンします。
 - UNIX では、root 権限を持つユーザー ID を使用してログオンします。
 - Windows では、DB2 インストール用に定義した管理者アカウントを使用してログオンします。
2. インストール・プログラムが必要に応じてファイルを更新できるように、オープンされているプログラムをすべてクローズします。
3. DB2 Relational Connect CD を挿入し、セットアップ・プログラムを開始して DB2 Relational Connect をインストールします。
 - UNIX では、DB2 Relational Connect CD を CD-ROM に挿入し、マウントします。CD-ROM がマウントされたディレクトリーに変更します。`./db2setup` コマンドを入力し、セットアップ・プログラムを開始します。
 - Windows では、DB2 Relational Connect CD を CD-ROM ドライブに挿入します。自動実行機能がセットアップ・プログラムを自動的に開始します。セットアップ・プログラムの自動開始に失敗した場合は、セットアップ・プログラムを手動で開始することができます。
セットアップ・プログラムを手動で開始するには、「スタート」をクリックし、「ファイル名を指定して実行」オプションを選択します。「名前」フィールドに、`x:\setup` (`x:` は CD-ROM ドライブ) と入力します。それから、「OK」をクリックします。
4. 「DB2 Relational Connect セットアップ・ランチパッド」が開きます。このウィンドウで、インストールの前提条件と最新のリリース情報を確認してください。

5. セットアップ・プログラムの「インストールする機能の選択 (Select the features to install)」パネルから、「SQL Server データ・ソース用 Relational Connect (Relational Connect for SQL Server Data Sources)」を選択します。このセットアップでは、以下のものを指定する必要があります。

- ODBC ドライバーがインストールされているローカル・パス。
- ODBC ドライバー・マネージャー・ディレクトリーのローカル・パス。
- ODBC トレース・ディレクトリーのローカル・パス。
- ODBC ライブラリーのローカル・パス。

Relational Connect のインストールは、sqllib/cfg/db2dj.ini ファイルを更新し、いくつかの環境変数 (ODBCINI、DJXODBCTRACE、および DJX_ODBC_LIBRARY_PATH 7) を設定します。その他の環境変数を設定する場合は、手作業で設定する必要があります。このステップについては、「データ・ソース環境変数のチェック」というトピックで説明しています。

UNIX では、インストールで DB2 を ODBC ドライバーにリンクします。

注意: DB2 Relational Connect のインストールを実行する前に、Microsoft SQL Server ドライバーをインストールしない場合は、手作業で環境変数を設定し、DB2 をクライアント・ソフトウェアにリンクする必要があります。このステップについては、「連合サーバーのセットアップのチェック」に記載されています。

インストール操作のヘルプを使用すれば、ステップに沿ってガイドしてくれます。インストール操作のヘルプを呼び出すには、「ヘルプ」をクリックするか「F1」キーを押します。「キャンセル (Cancel)」をクリックすれば、いつでもインストールを終了できます。

6. インストールの一部で、以下のことを行います。
 - 連合サーバーに DB2 インスタンスを作成する。これにより、DB2 データベース・マネージャーの FEDERATED パラメーターは YES に設定され、これにより DB2 サーバーはデータ・ソースにアクセスできるようになります。
 - インスタンスに対するユーザー権限情報を指定する。
7. 最後のインストール・セットアップ・パネルで「終了 (Finish)」をクリックし、DB2 Relational Connect ファイルをシステムにコピーします。

インストールを完了すると、DB2 Relational Connect は、DB2 サーバー・ソフトウェアと同じディレクトリーにインストールされています。
8. .odbc.ini ファイルを DB2 インスタンス所有者のホーム・ディレクトリーにリンクまたはコピーします。ODBC ドライバーと一緒に配布される .odbc.ini ファイルは、現行ディレクトリーに置かれます。

オペレーティング・システムの中には、以下のシンボリック・リンクを作成しなければならないものもあります。

HP-UX /usr/exe/libodbcinst.sl から \$DJX_ODBC_LIBRARY_PATH/libodbcinst.sl へのシンボリック・リンクを作成

Linux /usr/local/locale から \$DJX_ODBC_LIBRARY_PATH/./locale へのシンボリック・リンクを作成

Solaris オペレーティング環境

\$HOME/sql1lib/locale から \$DJX_ODBC_LIBRARY_PATH/./locale へのシンボリック・リンクを作成

\$HOME は DB2 インスタンス所有者のホーム・ディレクトリーです。

ソフトウェアをインストールした後、SYSADM 権限を持つユーザーがセットアップをチェックし、**連合データベース**を作成する必要があります。その後、DB2 インスタンス所有者は、Microsoft SQL Server データ・ソースにアクセスするようにサーバーを構成します。

関連概念:

- *管理ガイド: インプリメンテーション* の『インスタンスの作成』
- *DB2 Universal Database サーバー機能 概説およびインストール* の『DB2 サーバーのインストールの概要 (UNIX)』
- 396 ページの『DB2 サーバーのインストールの概要 (Windows)』
- *DB2 Universal Database サーバー機能 概説およびインストール* の『パーティション DB2 サーバーのインストールの概要 (UNIX)』
- *DB2 Universal Database サーバー機能 概説およびインストール* の『パーティション DB2 サーバーのインストールの概要 (Windows)』
- *連合システム・ガイド* の『データ・ソースの構成の近道』

関連タスク:

- *DB2 Universal Database サーバー機能 概説およびインストール* の『DB2 CD-ROM のマウント (HP-UX)』
- *DB2 Universal Database サーバー機能 概説およびインストール* の『DB2 CD-ROM のマウント (AIX)』
- *DB2 Universal Database サーバー機能 概説およびインストール* の『CD-ROM のマウント (Solaris)』
- *連合システム・ガイド* の『サーバーとデータベースをセットアップする近道』
- *連合システム・ガイド* の『連合サーバーのセットアップをチェックする』
- *連合システム・ガイド* の『データ・ソース環境変数をチェックする』
- *連合システム・ガイド* の『連合データベースを作成する』

データ・ソース・オブジェクトからデータを削除する

データ・ソースからデータを削除するには、DELETE ステートメントでデータ・ソース・オブジェクトのニックネームを使用します。

前提条件:

ニックネームを使用してデータを削除する場合、ステートメントの許可 ID が持つ特権には、(連合データベースが要求を受け入れるための) ニックネームに対する DELETE 特権、および (データ・ソースが要求を受け入れるための) 基になる表オブジェクトに対する DELETE 特権が含まれている必要があります。

制限:

Sybase DBLIB ラッパーを使用するニックネームに対して DELETE 操作を実行することはできません。 Sybase DBLIB ラッパーは、読み取り専用だからです。

ODBC ラッパーを使用するニックネームに対して DELETE 操作を実行することはできません。 ODBC ラッパーは、読み取り専用だからです。

DB2 Life Sciences Data Connect ラッパーを使用するニックネームに対して DELETE 操作を実行することはできません。 DB2 Life Sciences Data Connect のラッパーは、読み取り専用だからです。

手順:

連合データベースはローカル・データ・ソースです。連合データベースにある表とビューは、ローカル・オブジェクトです。これらのオブジェクトにはニックネームを作成せず、DELETE ステートメントでは実際のオブジェクト名を使用します。

リモート・データ・ソースには、連合サーバー上の別の DB2 (UNIX および Windows 版) データベース・インスタンス、別のサーバー上の DB2 (UNIX および Windows 版) データベース・インスタンス、および DB2 (UNIX および Windows 版) 以外のデータ・ソースが含まれます。リモートのデータ・ソースにあるオブジェクトは、リモート・オブジェクトです。

Informix 表が 2 つの列から構成されているとします。1 番目の列には、INTEGER データが、2 番目の列には (最大 20 桁の) VARCHAR データが入っています。表名は、**infx_table** です。連合サーバーが **infx_table** にアクセスできるように構成するためのステップと SQL ステートメントは、以下ようになります。

- ラッパーを作成するためのステートメントは、以下ようになります。

```
CREATE WRAPPER informix
```

- サーバー定義を作成するためのステートメントは、以下ようになります。


```
CREATE SERVER infx_server TYPE informix  
VERSION 9.3 WRAPPER informix  
OPTIONS(ADD NODE 'inf93', ADD DBNAME 'inf_db',  
ADD IUD_APP_SVPT_ENFORCE 'N')
```

- ユーザー・マッピングを作成するためのステートメントは、以下のようになります。

```
CREATE USER MAPPING FOR USER SERVER infx_server  
OPTIONS(ADD REMOTE_AUTHID 'infx_authid', ADD REMOTE_PASSWORD 'infx_pswd')
```

- 表 **infx_table** にニックネームを作成するためのステートメントは、以下のようになります。

```
CREATE NICKNAME infx_table_nn FOR infx_server."infx_authid".infx_table
```

infx_table_nn ニックネームを使用して、Informix 表に対して INSERT、UPDATE および DELETE ステートメントを実行することができます。たとえば、Informix 表の 1 行を削除するためのステートメントは、以下のようになります。

```
DELETE FROM infx_table_nn WHERE c1=3
```

関連タスク:

- 連合システム・ガイド の『連合システム内のデータの選択』
- 連合システム・ガイド の『データ・ソース・オブジェクトにデータを挿入する』
- 連合システム・ガイド の『データ・ソース・オブジェクト内のデータを更新する』

関連資料:

- SQL リファレンス 第 2 巻 の『DELETE ステートメント』

パフォーマンスに影響するシステム・モニター要素

DB2 データベース・システム・モニターは、データベース・マネージャーの現在の状態に関する統計情報、およびデータベース処理のカウンターやその他の測定などの、アクティビティー情報を収集します。

連合システムでは、DB2 データベース・システム・モニターを使用して、データベース・システム・モニター、システム・パフォーマンス、およびアプリケーション・パフォーマンスについての情報を収集できます。

タイム・スタンプ・モニター・スイッチは、連合データベースとデータ・ソースとの対話の応答時間をトラッキングするのに使用されます。タイム・スタンプ・スイッチによってトラッキングされる連合データ・エレメントは、以下のとおりです。

- ニックネーム作成応答時間
- 削除応答時間
- 挿入応答時間
- パススルー時間
- 照会応答時間

- リモート・ロック時間
- 更新応答時間

タイム・スタンプ・モニター・スイッチのデフォルトの設定値は ON です。

推奨: タイム・スタンプ・モニター・スイッチの設定値を、すべてのアプリケーションに対して OFF にすることで、パフォーマンスを向上させることができます。1 つのアプリケーションでタイム・スタンプ・スイッチが ON に設定されていると、システムは応答時間の収集を継続します。そのため、一部のアプリケーションのタイム・スタンプ・スイッチをオフにただけでは、パフォーマンスを向上させることはできません。

スイッチをオフにすることには、その他の意味があります。

- すべてのアプリケーションでタイム・スタンプ・モニター・スイッチをオフにする場合、変更をインプリメントするためには、DB2 インスタンスを停止し、再始動する必要があります。
- タイム・スタンプ・モニター・スイッチをオフにすると、連合および非連合アプリケーションの両方のタイム・スタンプ情報の収集が使用不可になります。ローカル・データベースも、タイム・スタンプ情報を受け取りません。

ローカルの非連合アプリケーションのタイム・スタンプ情報が必要な場合は、タイム・スタンプ・モニター・スイッチをオフにはいけません。

このコマンドを使用して、すべてのアプリケーションに対してタイム・スタンプ・スイッチを OFF にできます。

```
update dbm cfg using dft_mon_timestamp off
```

その後で、次のコマンドを実行します。

```
db2stop  
db2start
```

DB2 の停止と開始を実行することによって、スイッチがすべてのアプリケーションに対してオフになっていることが確認できます。

タイム・スタンプ・スイッチによってトラッキングされる各要素についての具体的な情報は、別のトピックで説明します。

関連資料:

- システム・モニター ガイドおよびリファレンス の『連合データベース・システムのデータ・エレメント』

アプライ・プログラムの操作

この章では、アプライ・プログラムの始動および停止の方法を説明します。また、ASNDONE および ASNLOAD 出口ルーチンの使用方法も説明します。

この章は、以下のセクションを含んでいます。

- 『アプライ・プログラムの始動 (UNIX、Windows、z/OS)』
- 488 ページの『アプライ・プログラムの始動 (OS/400)』
- 489 ページの『アプライ・プログラムの停止』
- 490 ページの『ASNDONE 出口ルーチンの変更 (UNIX、Windows、z/OS)』
- 491 ページの『ASNDONE 出口ルーチンの変更 (OS/400)』
- 493 ページの『ASNLOAD 出口ルーチンを使ったターゲット表のリフレッシュ』

アプライ・プログラムの始動 (UNIX、Windows、z/OS)

アプライ・プログラムのインスタンスを始動して、ターゲットへのデータのアプライを開始できます。

アプライ・プログラムは始動された後、以下のイベントのいずれかが発生するまで、連続的に実行されます (**copyonce** 始動パラメーターが使用されない場合)。

- ユーザーがレプリケーション・センターまたはコマンドを使用してアプライ・プログラムを停止する。
- アプライ・プログラムがアプライ・コントロール・サーバーに接続できない。
- アプライ・プログラムは処理用のメモリーを割り振ることができない。

前提条件:

アプライ・プログラムを始動する前に、663 ページの『レプリケーションの設定』の説明にしたがって、システムが正しくセットアップされていることを確認してください。

- 必要なレプリケーション・サーバーのすべてへの接続が構成されている。
- 正しい許可を受けている。
- 必要なアプライ修飾子に対してソースおよびコントロール・データを含むコントロール表が作成されている。
- レプリケーション・プログラムが構成済みである。
- z/OS の場合、必要なすべてのサーバーにアプライ・プログラムが手動でバインドされている。
- UNIX および Windows 上で実行されるリモート・サーバーに対するエンド・ユーザー認証用のパスワード・ファイルが存在している。

また、以下の条件が満たされていることを確認してください。

- アプライ修飾子に対して少なくとも 1 つのアクティブなサブスクリプション・セットが存在し、サブスクリプション・セットが、1 つ、または複数の以下の項目を含んでいる必要があります。
 - サブスクリプション・セット・メンバー
 - SQL ステートメント
 - 手順
- コンデンスされているターゲット表はすべてターゲット・キーを必要とします。ターゲット・キーは、アプライ・プログラムが各アプライ・サイクル中に複製する変更のトラッキングに使用する主キーまたはユニーク索引のいずれかである、ユニーク列のセットです。(非コンデンス CCD 表には主キーまたはユニーク索引はありません。)

手順:

以下の方法のいずれかを使用して、アプライ・プログラムを始動します。

レプリケーション・センター

「アプライの開始」ウィンドウを使用します。詳細については、レプリケーション・センターのヘルプを参照してください。

asnapply システム・コマンド (UNIX、Windows、z/OS)

詳細については、533 ページの『asnapply: アプライの開始 (UNIX、Windows、z/OS)』を参照してください。

Windows サービス (Windows)

詳細については、634 ページの『Windows Service Control Manager を使用してシステム・コマンドを発行する方法 (Windows)』を参照してください。

MVS コンソールまたは TSO (z/OS)

詳細については、613 ページの『レプリケーション・プログラムの操作 (z/OS)』を参照してください。

どのプロシージャーを使用してアプライ・プログラムを始動した場合にも始動パラメーターを設定する必要があります。以下のセクションでは、始動パラメーターについて説明し、各パラメーターで選択するべき値を推奨します。

- 480 ページの『apply_path (UNIX、Windows、z/OS)』
- 480 ページの『apply_qual (UNIX、Windows、z/OS)』
- 481 ページの『control_server (UNIX、Windows、z/OS)』
- 481 ページの『copyonce (UNIX、Windows、z/OS)』
- 482 ページの『db2_subsystem (z/OS)』
- 482 ページの『delay (UNIX、Windows、z/OS)』
- 482 ページの『errwait (UNIX、Windows、z/OS)』
- 483 ページの『inamsg (UNIX、Windows、z/OS)』
- 483 ページの『loadxit (UNIX、Windows、z/OS)』

- 483 ページの『logreuse (UNIX、Windows、z/OS)』
- 484 ページの『logstdout (UNIX、Windows、z/OS)』
- 484 ページの『notify (UNIX、Windows、z/OS)』
- 484 ページの『opt4one (UNIX、Windows、z/OS)』
- 485 ページの『pwdfile (UNIX、Windows)』
- 485 ページの『sleep (UNIX、Windows、z/OS)』
- 485 ページの『spillfile (UNIX、Windows、z/OS)』
- 486 ページの『sqlerrcontinue (UNIX、Windows)』
- 487 ページの『term (UNIX、Windows、z/OS)』
- 487 ページの『trlreuse (UNIX、Windows、z/OS)』

apply_path (UNIX、Windows、z/OS): デフォルト (UNIX、Windows、z/OS):
apply_path=current_directory

デフォルト (Windows NT 上のサービス): **apply_path=sqllib¥bin**

アプライ・パスは、アプライ・プログラムがログおよび作業ファイルを保管するディレクトリーです。デフォルトのアプライ・パスはプログラムを始動するディレクトリーです。ログおよび作業ファイルを他の場所に保管するには、アプライ・パスを変更します (たとえば、AIX システムの場合は /home/db2inst/apply_files)。アプライ・ログ・ファイルにアクセスするには、このディレクトリーに移動する必要があるため、どのディレクトリーを選択したか覚えておいてください。

重要: 選択したディレクトリーに、アプライ・プログラムから使用される一時ファイルに十分なスペースがあることを確認してください。

Windows NT システムでのアプライ・プログラムのインスタンスの始動: レプリケーション・センターまたは **asnapply** コマンドを使用してアプライ・プログラムを始動する場合、もし 大文字と小文字の区別以外は同じ複数のアプライ修飾子が使用されるのであれば、アプライ・パスを指定する必要があります。Windows NT システムのファイル名は大文字小文字が区別されません。たとえば、APPLYQUAL1、ApplyQual1、applyqual1 という 3 つのアプライ修飾子があるとします。アプライ・プログラムの各インスタンスのログ・ファイルのファイル名が競合しないように、これらのアプライ・インスタンスはそれぞれ異なる **apply_path** を使用して開始する必要があります。

apply_qual (UNIX、Windows、z/OS): 処理対象のサブスクリプション・セットのアプライ修飾子を指定する必要があります。(アプライ修飾子は、サブスクリプション・セットを作成するときに定義します。) 1 つの始動コマンドで指定できるアプライ修飾子は 1 つだけです。

重要: アプライ修飾子の`大文字小文字`は区別されます。入力された値が、サブスクリプション・セット (IBMSNAP_SUBS_SET) 表の `APPLY_QUAL` 列の値と一致する必要があります。

複数のアプライ修飾子を定義してあれば、アプライ・プログラムの別のインスタンスを始動できます。始動されたアプライ・プログラムの各インスタンスは、同じアプライ・コントロール・サーバーで表される異なるサブスクリプション・セットを処理します。たとえば、2 つのサブスクリプション・セットが定義され、それぞれのセットにユニークの修飾子、`APPLY1` および `APPLY2` があるとします。アプライ・プログラムの 2 つのインスタンス (各修飾子に対して 1 つずつ) を始動できます。各インスタンスは、`CNTRLSVR` と呼ばれるアプライ・コントロール・サーバーにあるコントロール表を使用します。アプライ・プログラムの各インスタンスは、各自のサブスクリプション・セットを別個に処理するため、アプライ・プログラムの 1 つのインスタンスですべてのセットを処理する場合よりもパフォーマンスが向上します。

control_server (UNIX, Windows, z/OS): デフォルト (UNIX, Windows): `DB2DBDFT` 環境変数の値 (使用可能な場合)

デフォルト (z/OS): なし

アプライ・コントロール・サーバーは、サブスクリプション定義およびアプライ・コントロール表が置かれているサーバーです。1 つのアプライ修飾子に対して、コントロール・サーバーを 1 つだけ指定してください。値が指定されない場合、アプライ・プログラムはデフォルト・サーバー上で始動されます。デフォルトはオペレーティング・システムによって異なります。

アプライ・プログラムがコントロール・サーバーに接続できない場合、アプライ・プログラムは終了します。アプライ・プログラムは他のサーバーに接続できなくても終了しません。この場合には、エラー・メッセージを発行して処理を続けます。

copyonce (UNIX, Windows, z/OS): デフォルト: `copyonce=n`

copyonce パラメーターは、アプライ・プログラムのコピー・サイクルを決定します。

copyonce=y を指定して開始されたアプライ・プログラムは、適格なサブスクリプション・セットをそれぞれ一度だけ処理した後で終了します。この場合、以下の条件のいずれかが満たされた場合、サブスクリプション・セットは処理に適格となります。

- サブスクリプション・セットは相対タイミングを使用し、時間が経過し、サブスクリプション・セットはアクティブである。
- サブスクリプション・セットはイベント・ベースのタイミングを使用し、アクティブであり、イベントが発生したが、アプライ・プログラムはまだサブスクリプション・セットを処理していない。

copyonce=n を使用してアプライ・プログラムを始動する典型的な状況は、アプライ・プログラムの実行を継続し、適格なサブスクリプションの処理を続ける必要があるときです。

時々ネットワークに接続するようなダイヤルイン環境でアプライ・プログラムを実行するときには、**copyonce=n** ではなく、**copyonce=y** を使用してください。また、テスト環境でアプライ・プログラムを実行するときにも、**copyonce=y** を使用することが考えられます。

ヒント: サブスクリプション・セットが適格であり、レプリケーションできるデータがあるかぎり、アプライ・プログラムがサブスクリプション・セットの処理を何回も行うようにしたい場合は、**copyonce=y** ではなく、**sleep=n** を使用してください。

Copyonce=y は、複製すべきデータが残っている場合も、各セットを一度しか処理しません。

db2_subsystem (z/OS): **db2_subsystem** パラメーターは、**control_server** が z/OS 上にある場合に、DB2 サブシステムの名前を指定します。入力する DB2 サブシステム名は最大 4 文字です。このパラメーターにはデフォルトはありません。このパラメーターは必須です。

delay (UNIX, Windows, z/OS): デフォルト: **delay=6** 秒

delay パラメーターは、アプライ・プログラムが各アプライ・サイクルの最後に待機する秒数を設定します。

デフォルトでは、連続してレプリケーションを行う場合 (サブスクリプション・セットで **sleep=0** 分を使用する場合)、アプライ・プログラムはサブスクリプション・セットが正常に処理されてから、6 秒間待った後にサブスクリプション・セットを再試行します。複製すべきデータベース・アクティビティーがない場合は、ゼロ以外の値を使用して CPU サイクルを節約してください。待ち時間を少なくするには遅延値を小さくします。

errwait (UNIX, Windows, z/OS): デフォルト: **errwait=300** 秒 (5 分)

errwait パラメーターは、サブスクリプション・サイクルが失敗した後に、アプライ・プログラムがサブスクリプション・セットを再試行するまでに待つ秒数を指定します。

デフォルトでは、アプライ・プログラムはサブスクリプション・サイクルが失敗した後 300 秒待機してから、サブスクリプション・セットを再試行します。テスト環境ではより小さい値を使用することが考えられます。最小値は 1 秒です。実稼働環境では、このパラメーターのデフォルトを変更する前に、トレードオフを考慮してください。

- 小さい値を使用すると、アプライ・プログラムがハード・エラーを繰り返し再試行することにより、CPU サイクルが無駄になります。たとえば、ターゲット表に問題があるときにアプライ・プログラムがサブスクリプション・セットの処理を繰り返し再試

行した場合は、CPU サイクルを不必要に消費することになります。ログ・ファイルの中に、あるいはアプライ・プログラムが z/OS 上で実行されている場合はオペレーター・コンソールに、大量のメッセージが送られることもあります。

- 大きな値を使用すると、アプライ・プログラムが一時的エラー条件を再試行するまで待つ必要がある場合の待ち時間が増えることになります。たとえば、迅速に解決されるネットワーク・エラーを検出したときにもアプライ・プログラムは不必要に待つことになるため、**errwait** パラメーターに大きな値を使用すると、待ち時間が増加します。

inamsg (UNIX, Windows, z/OS): デフォルト: **inamsg=y**

inamsg パラメーターは、アプライ・プログラムが非アクティブになったときにメッセージを発行するかどうかを指定します。

デフォルトでは、アプライ・プログラムは非アクティブになるとメッセージを発行します。サブスクリプション・セットの処理間のアプライ・プログラムの待ち時間が長くない場合は特にそうですが、メッセージはアプライ・ログ・ファイルのスペースを多量に消費します。このため、アプライ・プログラムが非アクティブになった場合のメッセージを発行させたくないことも考えられます。これらのメッセージをオフにするには、**inamsg=n** を使用します。

loadxit (UNIX, Windows, z/OS): デフォルト: **loadxit=n**

loadxit パラメーターは、アプライ・プログラムが ASNLOAD 出口ルーチンを使用してターゲット表をリフレッシュするかどうかを指定します。

デフォルトでは、アプライ・プログラムはターゲット表をリフレッシュするために ASNLOAD 出口ルーチンを使用しません (**loadxit=n**)。アプライ・プログラムが ASNLOAD 出口ルーチンを呼び出してターゲット表をリフレッシュするようにしたい場合は、**loadxit=y** を使用します。フル・リフレッシュ時にターゲット表に大量のデータをコピーするときには、ASNLOAD 出口を使用することを考えてみてください。アプライ・プログラムで ASNLOAD を使用するべき状況と、ASNLOAD の使用方法については、493 ページの『ASNLOAD 出口ルーチンを使ったターゲット表のリフレッシュ』を参照してください。

logreuse (UNIX, Windows, z/OS): デフォルト: **logreuse=n**

アプライ・プログラムは、操作情報をログ・ファイルに保管します。UNIX および Windows の場合、ログ・ファイルの名前は **db2instance.control_server.apply_qualifier.APP.log** になります。z/OS オペレーティング・システムの場合のファイル名は DB2 インスタンス名が含まれないことを除いては同じです。

このパラメーターは、ログ・ファイルに追加するか、上書きするかを指定します。

デフォルトでは、アプライ・プログラムは開始されるたびに、ログ・ファイルにメッセージを追加します (**logreuse=n**)。アプライ・プログラムから発行されたメッセージの履歴を保存する場合は、デフォルトのままにします。以下のような状況では、**logreuse=y** を使用して、アプライ・プログラムが開始時にログを削除し、再作成するようにできます。

- ログが大きくなったため、ログを消去してスペースを節約したい場合。
- ログに保管されている履歴が不要な場合。

logstdout (UNIX, Windows, z/OS): デフォルト: **logstdout=n**

logstdout パラメーターは、**asnapply** コマンドを使用した場合にしか使用できません。このパラメーターはレプリケーション・センターからは使用できません。

logstdout パラメーターは、アプライ・プログラムが、ログ・ファイルにのみメッセージを送信するか、ログ・ファイルと標準出力にメッセージを送信するかを指定します。

デフォルトでは、アプライ・プログラムはログ・ファイルにのみメッセージを送信します。トラブルシューティングの場合や、アプライ・プログラムの稼働状態をモニターする場合は、メッセージを標準出力に送信するように選択できます (**logstdout=y**)。

notify (UNIX, Windows, z/OS): デフォルト: **notify=n**

notify パラメーターは、アプライ・プログラムがサブスクリプションを処理した後に、ASNDONE 出口ルーチンに通知するかどうかを指定します。

デフォルトでは、アプライ・プログラムはサブスクリプションの処理が完了した後で ASNDONE 出口ルーチンに通知しません。**notify=y** が指定されると、アプライ・プログラムはサブスクリプション・サイクルが完了した後、アプライ・コントロール表の検査や、E メール・メッセージの送信など、追加処理を実行するために、ASNDONE を呼び出します。ASNDONE の詳細については、490 ページの『ASNDONE 出口ルーチンの変更 (UNIX, Windows, z/OS)』を参照してください。

opt4one (UNIX, Windows, z/OS): デフォルト: **opt4one=n**

opt4one パラメーターは、アプライ・プログラムの処理が、1 つのサブスクリプション・セット用に最適化されるかどうかを指定します。

デフォルトでは、アプライ・プログラムは複数のサブスクリプション・セット用に最適化されています。アプライ・プログラムは各コピー・サイクルの最初にレプリケーション・コントロール表から情報を読み取ります。アプライ修飾子に対してサブスクリプション・セットが 1 つである場合は、アプライ・プログラムがメモリー・キャッシュ内にサブスクリプション・セットのメンバーおよび列の情報を入れてそれを再利用するように、**opt4one=y** を使用してアプライ・プログラムを始動してください。アプライ・プロ

グラムを 1 つのサブスクリプション・セット用に最適化すると、アプライ・プログラムから使用される CPU が少なくなるため、スルーット率が向上します。

重要: `opt4one=y` を使用した場合には、セットにメンバーを追加するか、その他の方法でセットを変更したときには、アプライ・プログラムがコントロール表の中の変更を入手できるように、アプライ・プログラムを停止してから再度開始する必要があります。

pwdfile (UNIX, Windows): デフォルト: `pwdfile=asnpwd.aut`

データを複数のサーバーに分散する場合は、アプライ・プログラムがリモート・サーバー上のデータにアクセスできるように、暗号化されたパスワード・ファイルにユーザー ID とパスワードを保管できます。詳細については、672 ページの『レプリケーション用のユーザー ID およびパスワードの保管 (UNIX, Windows)』を参照してください。

sleep (UNIX, Windows, z/OS): デフォルト: `sleep=y`

sleep パラメーターは、アプライ・プログラムが適格なサブスクリプション・セットを処理した後で、スリープ・モードで実行を続けるか、終了するかを指定します。

デフォルトでは、アプライ・プログラムは `sleep=y` で始動されます。アプライ・プログラムは適格なサブスクリプション・セットがあるかどうかをチェックします。適格なサブスクリプション・セットが検出されると、アプライ・プログラムはセットを処理し、別の適格なセットがないか探します。アプライ・プログラムは適格なセットが検出されると、その処理を続けます。適格なセットが検出されない場合、アプライ・プログラムはスリープ・モードで実行を続け、定期的に「ウェイクアップ」して、適格なサブスクリプション・セットがあるかどうかをチェックします。通常の場合は、長期間にわたって更新を適用するために、アプライ・プログラムをアクティブな状態で実行しておきたいため、この方法でアプライ・プログラムを始動します。

`sleep=n` を指定して始動されたアプライ・プログラムは、適格なサブスクリプション・セットがあるかどうかをチェックし、それを処理します。アプライ・プログラムは適格なサブスクリプション・セットが見つからなくなるまでセットの処理を続け、複製すべきデータがなくなるまで適格なセットの処理を繰り返した後、終了します。`sleep=n` を使用する典型的な状況は、適格なサブスクリプション・セットが検出されたときのみアプライ・プログラムが実行され、その後プログラムが終了される、モバイル環境またはテスト環境です。この場合は、アプライ・プログラムをスリープ・モードで待機させ、適格なセットがないかどうかをチェックするために定期的にウェイクアップさせることはしません。このような環境では、アプライ・プログラムを無期限に実行するのではなく、アプライ・プログラムをいつ実行するかをユーザーがコントロールできるようにします。

ヒント: 各サブスクリプション・セットを一度だけ処理する場合は、`sleep=n` ではなく、`copyonce=y` を使用してください。

spillfile (UNIX, Windows, z/OS): デフォルト (UNIX, Windows): `spillfile=disk`

デフォルト (z/OS): spillfile=MEM

アプライ・プログラムはソース表からデータをリトリブし、アプライ・プログラムが実行されているシステム上の予備ファイルに入れます。

UNIX および Windows オペレーティング・システムの場合、spillfile の唯一有効な設定は disk です。予備ファイルは、必ず **apply_path** で指定されたロケーションのディスクに置かれるからです。

USS を含む z/OS オペレーティング・システムの場合は、予備ファイルはデフォルトでメモリー内に保管されます。アプライ・プログラムが ASNAPLDD カード上の指定を使用して予備ファイルの割り振りを行うように、予備ファイルをディスクに保管するように指定できます。ASNAPLDD カードが指定されていない場合は、VIO が使用されます。レプリケーション・サイクルが短い場合は、レプリケーションするデータが少量であれば、メモリー内に予備ファイルを作成するのが妥当です。

sqlerrcontinue (UNIX, Windows): デフォルト: sqlerrcontinue=n

sqlerrcontinue パラメーターは、アプライ・プログラムが特定の SQL エラーに対処する方法を指定します。

デフォルトでは、アプライ・プログラムは SQL エラーを検出すると、そのサブスクリプション・セットの処理を停止し、エラー・メッセージを生成します。通常の場合、実稼働環境ではデフォルトを使用します。

テスト環境の場合は、ターゲット表にデータを挿入するときに、特定の SQL エラーが発生することが予想されます。これらのエラーはユーザーにとっては許容できる場合もありますが、エラーにより現行のサブスクリプション・サイクルは停止します。このような状況では、アプライ・プログラムがエラーを無視し、そのサイクルで複製したデータをロールバックしないように、**sqlerrcontinue=y** を使用してアプライ・プログラムを始動できます。アプライ・プログラムはターゲット表へのデータ挿入時に SQL エラーを受け取ると、**apply_qualifier.sqs** ファイルの中の値をチェックします。一致するものがあると、エラーに関する詳細をエラー・ファイル、**apply_qualifier.err** に書き込み、処理を続行します。アプライ・プログラムは **apply_qualifier.sqs** ファイルにリストされていない SQL エラーを検出すると、セットの処理を停止し、次のセットに進みます。

sqlerrcontinue=y オプションを使用してアプライ・プログラムを始動する前に、**apply_qualifier.sqs** ファイルを作成し、アプライ・プログラムの呼び出しに使用したディレクトリーにこのファイルを保管する必要があります。このファイルには、最大 20 の 5 バイトの値を続けてリストしてください。アプライ・プログラムの実行中にこのファイルの内容を変更したときには、アプライ・プログラムが新しい値を認識できるように、アプライ・プログラムを停止してから再度開始してください。

例: ターゲット表で次のエラーを受け取ったときにアプライ・プログラムがサブスクリプション・セットの処理を続けるようにするとします (sqlstate/code)。

42704/-803

重複索引違反

以下の SQL 状態を含む SQL 状態ファイルを作成します。

42704

ターゲット表の更新時にこの SQL 状態が戻された場合、アプライ・プログラムはセット内のその他のターゲット表に変更を適用し、エラーおよび拒否された行の両方を示すエラー・ファイルを作成します。

ヒント: アプライ・トレール (IBMSNAP_APPLYTRAIL) 表の STATUS 列をチェックします。16 という値は、アプライ・プログラムがサブスクリプション・セットを正常に処理したが、*apply_qualifier.sqs* ファイルで定義された、許容されるなんらかのエラーが発生したことを意味します。

term (UNIX, Windows, z/OS): デフォルト: term=y

term パラメーターは、DB2 の状況が、アプライ・プログラムの稼働にどのような影響を与えるかを決定します。

デフォルトでは、アプライ・プログラムは DB2 が終了すると終了します。

DB2 がアクティブでないときに、アプライ・プログラムが DB2 の始動を待つようにするには、**term=n** を使用します。z/OS プラットフォームの場合、DB2 が静止したときにアプライ・プログラムがアクティブであると、アプライ・プログラムはアクティブなまま残り、DB2 が開始されるまで再接続しません。UNIX および Windows プラットフォームの場合、DB2 が静止したときにアプライ・プログラムがアクティブであると、アプライ・プログラムはアクティブなまま残り、DB2 が静止モードを終了するまで再接続しません。

trlreuse (UNIX, Windows, z/OS): デフォルト: trlruse=n

trlreuse パラメーターは、アプライ・プログラムの始動時に、アプライ・トレール (IBMSNAP_APPLYTRAIL) 表を再利用する (追加する) か、上書きするかを指定します。

デフォルトでは、アプライ・プログラムは開始されると、アプライ・トレール表にエントリーを追加します。この表は、アプライ・コントロール・サーバーにあるすべてのアプライ・インスタンスの操作の履歴を保持します。この表は、診断およびパフォーマンス統計のリポジトリです。更新の履歴を保持する場合は、デフォルトを使用します。以下のような状況では、アプライ・プログラムの始動時に、アプライ・トレール表に追加するのではなく、表を空にすることができます (**trlreuse=y**)。

- アプライ・トレール表が大きくなったため、消去してスペースを節約したい場合。
- 表に保管されている履歴が不必要な場合。

ヒント: `trlreuse=y` を使用する代わりに、アプライ・プログラムがサブスクリプション・セットを正常に処理した後で (`status=0`)、SQL 処理を使用してアプライ・トレール表から行を削除できます。

アプライ・プログラムの始動 (OS/400)

アプライ・プログラムのインスタンスを始動して、ターゲットへのデータのアプライを開始できます。

アプライ・プログラムは始動された後、以下の条件のいずれかが発生するまで、連続的に実行されます。

- ユーザーが `COPYONCE(*YES)` 始動パラメーターを使用してプログラムを始動する。
- ユーザーが `ALWINACT(*NO)` を指定し、処理すべきデータがない。
- ユーザーがレプリケーション・センターまたはコマンドを使用してアプライ・プログラムを停止する。
- アプライ・プログラムがアプライ・コントロール・サーバーに接続できない。
- アプライ・プログラムは処理用のメモリーを割り振ることができない。

前提条件:

アプライ・プログラムを始動する前に、システムが正しくセットアップされていることを確認してください。

- すべてのレプリケーション・サーバーへの接続が構成されている。
- 正しい許可を受けている。
- コントロール表が作成されている。
- レプリケーション・プログラムが構成済みである。

また、以下の条件が満たされていることを確認してください。

- アプライ修飾子に対して少なくとも 1 つのアクティブなサブスクリプション・セットが存在し、サブスクリプション・セットが、1 つ、または複数の以下の項目を含んでいる必要があります。
 - サブスクリプション・セット・メンバー
 - SQL ステートメント
 - 手順
- コンデンスされているターゲット表はすべてターゲット・キーを必要とします。ターゲット・キーは、アプライ・プログラムが各アプライ・サイクル中に複製する変更のトラッキングに使用する主キーまたはユニーク索引のいずれかである、ユニーク列のセットです。(非コンデンス CCD 表には主キーまたはユニーク索引はありません。)

手順:

以下の方法のいずれかを使用して、アプライ・プログラムを始動します。

レプリケーション・センター

「アプライの開始」ウィンドウを使用します。詳細については、レプリケーション・センターのヘルプを参照してください。

STRDPRAPY システム・コマンド

アプライ・プログラムを始動するときには、稼働パラメーターにこれらのデフォルト設定を使用できます。

表 148. アプライ・プログラムのデフォルト設定 (OS/400)

稼働パラメーター	(*value) の記述
USER (*CURRENT)	システムにサインオンしたユーザー。
JOBID (*LIBL/QZSNDPR)	製品ライブラリー名 / ジョブ記述。
APYQUAL (*USER)	現行ユーザー名 (上記から)。
CTLSVR (*LOCAL)	ローカル RDB サーバー名。
TRACE (*NONE)	トレースを生成しない。
FULLREFPGM (*NONE)	ASNLOAD 出口ルーチンを実行しない。
SUBNFYPGM (*NONE)	ASNDONE 出口ルーチンを実行しない。
INACTMSG (*YES)	アプライ・プログラムは非アクティブ期間を開始すると、プログラムが非アクティブになる期間を示すメッセージ ASN1044 を生成する。
ALWINACT (*YES)	処理するものがない場合はスリープする。
DELAY (6)	再度処理する前に、アプライ・サイクルの後に 6 秒待機する。
RTYWAIT (300)	失敗した操作を再始動する前に 300 秒待機する。
COPYONCE (*NO)	1 つのコピー・サイクルを完了した後で終了せずに処理を続行する。
TRLREUSE (*NO)	アプライ・プログラムの始動時にアプライ・トレール (IBMSNAP_APPLYTRAIL) 表を空にしない。
OPTSNGSET (*NO)	アプライ・プログラムのパフォーマンスを 1 つのサブスクリプション・セットの処理用に最適化しない。

アプライ・プログラムの停止

ユーザーはアプライ・プログラムのインスタンスを停止できます。ユーザーがアプライ・プログラムを停止すると、アプライ・プログラムはターゲット表にデータをコピーしなくなり、次にプログラムが開始されたときに正しく開始されるように、コントロール表の中の情報を更新します。

前提条件:

アプライ・プログラムのインスタンスが開始されている必要があります。

手順:

アプライ・プログラムのインスタンスを停止するには、以下の方法のいずれかを使用します。

レプリケーション・センター

「アプライの停止」ウィンドウを使用します。詳細については、レプリケーション・センターのヘルプを参照してください。

ASNDONE 出口ルーチンの変更 (UNIX、Windows、z/OS)

このセクションでは、UNIX、Windows、および z/OS オペレーティング・システムでの ASNDONE 出口ルーチンのカスタマイズ方法を説明します。

notify=y パラメーターを使用してアプライ・プログラムを始動すると、アプライ・プログラムはサブスクリプションの処理が成功したかどうかに関係なく、サブスクリプションの処理が終了した後、ASNDONE 出口ルーチン呼び出します。以下のリストは、レプリケーション環境で使用するために ASNDONE 出口ルーチンをどのように変更できるかの例を示しています。

- トランザクションが拒否されたことが判明した場合は、この出口ルーチンを使用して、リジェクトされたトランザクションを UOW 表で確認し、その後のアクション (たとえば、E メールをレプリケーション・オペレーターに自動的に送信する、メッセージを発行する、またはアラートを生成する) を開始します。
- この出口ルーチンを使用して、失敗したサブスクリプション・セットが訂正されるまでアプライ・プログラムが再試行を続けないように、失敗したサブスクリプション・セットを非活動化します。失敗したサブスクリプション・セットを検出するには、アプライ・トレール (IBMSNAP_APPLYTRAIL) 表で STATUS= -1 を探すように出口ルーチンを変更します。サブスクリプション・セットを非活動化するには、サブスクリプション・セット (IBMSNAP_SUBS_SET) 表に ACTIVATE=0 を設定するように出口ルーチンを構成します。
- データが各 サブスクリプション・セットに適用された後で出口ルーチンを使用してデータを操作します。(この代わりに、SQL ステートメントまたはストアド・プロシージャを使用して、アプライ・プログラムによる特定の サブスクリプション・セットの処理の前または後で実行される実行時処理ステートメントを定義することもできます)

手順:

サンプルの ASNDONE 出口ルーチンの変更したバージョンを使用するには、以下のようになります。

1. 要件に合うように、ASNDONE ルーチンを調整します。

UNIX、Windows の場合: この出口ルーチンの変更方法については、サンプル・プログラム (%sqllib%samples%repl%asndone.smp) の PROLOG セクションを参照してください。

z/OS の場合: サンプル・プログラム SASNAJCL(ASNDONE) の PROLOG セクションを参照してください。

2. プログラムをコンパイル、リンク、バインドし、実行可能ファイルを適切なディレクトリに入れます。
3. **notify=y** パラメーターを使用してアプライ・プログラムを始動し、ASNDONE 出口ルーチンを呼び出します。

ASNDONE 出口ルーチンの変更 (OS/400)

このセクションでは、OS/400 環境での ASNDONE 出口ルーチンのカスタマイズ方法を説明します。

SUBNFYPGM パラメーターを ASNDONE 出口ルーチンの名前に設定してアプライ・プログラムを始動すると、アプライ・プログラムは、サブスクリプションの処理が成功したかどうかに関係なく、サブスクリプションの処理が終了した後、ASNDONE 出口ルーチンを呼び出します。以下のリストは、レプリケーション環境で使用するために ASNDONE 出口ルーチンをどのように変更できるかの例を示しています。

- トランザクションが拒否されたことが判明した場合は、この出口ルーチンを使用して、リジェクトされたトランザクションを UOW 表で確認し、その後のアクション (たとえば、E メールをレプリケーション・オペレーターに自動的に送信する、メッセージを発行する、またはアラートを生成する) を開始します。
- この出口ルーチンを使用して、失敗したサブスクリプション・セットが訂正されるまでアプライ・プログラムが再試行を続けないように、失敗したサブスクリプション・セットを非活性化します。失敗したサブスクリプション・セットを検出するには、アプライ・トレール (IBMSNAP_APPLYTRAIL) 表で STATUS= -1 を探すように出口ルーチンを変更します。サブスクリプション・セットを非活性化するには、サブスクリプション・セット (IBMSNAP_SUBS_SET) 表に ACTIVATE=0 を設定するように出口ルーチンを構成します。
- データが各 サブスクリプション・セットに適用された後で出口ルーチンを使用してデータを操作します。(この代わりに、SQL ステートメントまたはストアード・プロシージャを使用して、アプライ・プログラムによる特定の サブスクリプション・セットの処理の前または後で実行される実行時処理ステートメントを定義することもできます。詳細については、722 ページの『ストアード・プロシージャまたは SQL ステートメントを使用したデータ拡張』を参照してください。)

手順:

サンプルの ASNDONE 出口ルーチンの変更したバージョンを使用するには、以下のようになります。

1. サイトの要件に合うように、ASNDONE 出口ルーチンを調整します。

以下の表は、C、COBOL、および RPG 言語のこのルーチンのソース・コードがどこにあるかを示しています。

コンパイラ言語	ライブラリー名	ソース・ファイル名	メンバー名
C	QDP4	QCSRC	ASNDONE
COBOL	QDP4	QCBLLSRC	ASNDONE
RPG	QDP4	QRPGLSRC	ASNDONE

プログラムの変更時に、以下の活動化グループに関連することを考慮してください。

新しい活動化グループを使用して実行するようにプログラムが作成されている場合:
アプライ・プログラムおよび ASNLOAD プログラムは、リレーショナル・データベースの接続およびオープン・カーソルなどの SQL リソースを共有しません。
OS/400 オペレーティング・システムにある活動化処理コードは、コントロールがアプライ・プログラムに戻される前に、ASNLOAD プログラムによって割り振られたリソースをすべて解放します。アプライ・プログラムが ASNLOAD プログラムを呼び出すたびに、追加のリソースが使用されます。

このプログラムが呼び出し側の活動化グループで実行するために作成される場合: アプライ・プログラムと SQL リソースを共有します。アプライ・プログラムへの影響が最小限で済むようプログラムを設計します。たとえば、現行のリレーショナル・データベース接続を変更すると、予期しないアプライ・プログラムの処理を引き起こす可能性があります。

このプログラムが名前付き活動化グループで実行するために作成される場合: アプライ・プログラムとの間でリソースを共有しません。名前付き活動化グループを使用すると、ASNLOAD プログラムが呼び出されるたびに、活動化グループのオーバーヘッドは回避されます。実行時のデータ構造および SQL リソースは、呼び出し間で共有されます。アプライ・プログラムが終了するまで、アプリケーションの終結処置は実行されません。したがって、サブスクリプション通知プログラムは、コントロールがアプライ・プログラムに戻される時、ソース表、ターゲット表、またはコントロール表をロックすることにより、アプライ・プログラムでロックの競合が生じることがないように設計します。

2. プログラムをコンパイル、リンク、バインドし、実行可能ファイルを適切なディレクトリに入れます。
3. アプライ・プログラムを始動し、**STRDPRAPY** コマンドのパラメーター **SUBNFYPGM** を使用して、ASNLOAD プログラムの名前を指定します。たとえば、プログラムの名前が **ASNDONE_1** でライブラリー **APPLIB** にある場合には、以下のコマンドを使用します。

```
SUBNFYPGM(APPLIB/ASNDONE_1)
```


ASNLOAD 出口ルーチンを使ったターゲット表のリフレッシュ

デフォルトでは、アプライ・プログラムはサブスクリプション・セット内の各ターゲット表のリフレッシュを実行するときに ASNLOAD 出口ルーチンを使用しません。アプライ・プログラムはソース表に対して全選択を行い、アプライ・プログラムが実行されているサーバー上の予備ファイルにデータを持ち込み、INSERT ステートメントを使用してターゲット表にデータを取り込みます。ソース表が大きい場合には、フル・リフレッシュ時にターゲットにデータを効率的にコピーするために、代わりに ASNLOAD 出口ルーチンを使用できます。

ASNLOAD 出口ルーチンは、ソース・フォーマットとコンパイル済みフォーマットの両方で、サンプルの出口プログラムとして提供されます。サンプルの出口プログラムは DB2 プラットフォームによってそれぞれ異なり、そのプラットフォームで提供されるユーティリティ・オプションを利用します。

アプライ・プログラムが ASNLOAD 出口ルーチンを呼び出したときにエラーが発生するか、出口ルーチンがゼロ以外の戻りコードを戻した場合、アプライ・プログラムはメッセージを発行し、現行のサブスクリプション・セットの処理を停止し、次のサブスクリプション・セットを処理します。

前提条件:

ASNLOAD 出口ルーチンを使用する前に、以下の前提条件が満たされていることを確認してください。

- ターゲット表の列は、ソース表の順序とデータ・タイプと一致する。
- ターゲット表は、レプリケーション・マッピングの一部である表のみを含む。

制約事項:

- UNIX および Windows システムでは、ASNLOAD 出口ルーチン用のソースおよびターゲット・データベースの両方もがバージョン 8 の DB2 データベースでなければなりません。しかし ASNLOAD 出口ルーチンは、すべてのサポートされているバージョンの DB2 for z/OS および DB2 for iSeries で使用できます。
- ASNLOAD 出口ルーチンはターゲット・ニックネームには使用できません。
- ASNLOAD 出口ルーチンを (DB2 Utilities Suite で使用可能なクロス・ローダー機能と一緒に)、以下の場合に、ソース・ニックネームに対してのみ使用できます。

- ソース表のニックネームがターゲット・データベースに存在する
- ターゲット・データベースがローカル (ソースと同じマシン上にある)

ソース表は、DB2 表または非 DB2 リレーショナル表です。ソース表が DB2 表の場合、ASN.IBMSNAP_SUBS_MEMBR 表の LOADX_SRC_N_OWNER および LOADX_SRC_N_TABLE 列を、ソース表を含むサブスクリプション・セット・メンバー用に設定する必要があります。

以下のセクションでは、さまざまなプラットフォーム上で ASNLOAD 出口ルーチンを使用する方法を説明します。

- 『ASNLOAD 出口ルーチンを使ったターゲット表のリフレッシュ (UNIX、Windows)』
- 495 ページの 『ASNLOAD 出口ルーチンを使ったターゲット表のリフレッシュ (z/OS)』
- 498 ページの 『ASNLOAD 出口ルーチンを使ったターゲット表のリフレッシュ (OS/400)』

ASNLOAD 出口ルーチンを使ったターゲット表のリフレッシュ

(UNIX、Windows): ASNLOAD 出口ルーチンは、DB2 の IMPORT ユーティリティーまたは DB2 の LOAD ユーティリティーと一緒に DB2 の EXPORT ユーティリティーを使用する、または新しい LOAD FROM CURSOR ユーティリティーを使用するなど、多数のユーティリティー・オプションを提供します。ユーザーがサンプルの出口ルーチンを呼び出すと、デフォルトでは、ソース・サーバー、ターゲット・サーバー、およびランタイム環境に基づいて、使用されるユーティリティーが選択されます。

コンパイル済みの出口ルーチンを使用したり、レプリケーション構成をカスタマイズして動作を構成したり、出口コード自体を変更することができます。レプリケーション構成は、サブスクリプション・メンバー (IBMSNAP_SUBS_MEMBER) 表の列を更新するか、サンプルの構成ファイル (asnload.ini) を更新することによりカスタマイズできません。

手順:

提供されたままの ASNLOAD ルーチンを使用する場合は、**loadxit=y** パラメーターを使用してアプライ・プログラムを始動します。

サンプルの ASNLOAD 出口ルーチンの変更したバージョンを使用するには、以下のようになります。

1. サイトの要件に合うように、ASNLOAD ルーチンを調整します。この出口ルーチンの変更方法については、サンプル・プログラム (`¥sqllib¥samples¥rep1¥asnload.smp`) の PROLOG セクションを参照してください。

重要: サンプルのソースは、asnload.ini ファイルからユーザー ID とパスワードの組み合わせを使用します。asnload.ini ファイルに特定のサーバーのユーザー ID とパスワードがない場合、または asnload.ini ファイルを使用できない場合は、user/using 句を使用しないで接続が行われます。

2. プログラムをコンパイル、リンク、バインドし、実行可能ファイルを適切なディレクトリに入れます。

3. ユーザーが指定したコードを使用して取り込みが行われるメンバーについては、LOADX_TYPE を 2 に設定してください。詳細については、497 ページの『ASNLOAD 出口の動作のカスタマイズ (UNIX、Windows、z/OS)』を参照してください。
4. **loadxit=y** パラメーターを使用してアプライ・プログラムを始動し、ASNLOAD 出口ルーチン呼び出します。

ASNLOAD 出口ルーチンへの入力を構成するには、498 ページの『ASNLOAD に構成ファイルを使用』を参照してください。

ASNLOAD 出口ルーチンから生成されるファイル:

これらのファイルは、ASNLOAD 出口ルーチン呼び出したアプライ・インスタンスの **apply_path** ディレクトリーに保管されます。

- **asnloadllapply_qualifier.trc**
トレースがオンの場合、このファイルはトレース情報を保持します。ASNLOAD 出口ルーチンがこのファイルを作成します。ファイルが存在する場合は、情報がファイルに追加されます。
- **asnloadllapply_qualifier.msg**
このファイルは、ロード統計を含めて、一般出口障害、警告メッセージおよび情報メッセージを保持します。ASNLOAD 出口ルーチンがこのファイルを作成します。ファイルが存在する場合は、情報がファイルに追加されます。
- **asnaEXPTllapply_qualifier.msg**
このファイルは、DB2 の EXPORT ユーティリティーが発行したエラー・メッセージ、警告メッセージ、または通知メッセージを保持します。ASNLOAD 出口ルーチンがこのファイルを作成します。ファイルが存在する場合は、情報がファイルに追加されます。
- **asnaIMPTllapply_qualifier.msg**
このファイルは、DB2 の IMPORT ユーティリティーが発行したエラー・メッセージ、警告メッセージ、または通知メッセージを保持します。ASNLOAD 出口ルーチンがこのファイルを作成します。ファイルが存在する場合は、情報がファイルに追加されます。
- **asnaLOADllapply_qualifier.msg**
このファイルは、DB2 の LOAD ユーティリティーが発行したエラー・メッセージ、警告メッセージ、または通知メッセージを保持します。ASNLOAD 出口ルーチンがこのファイルを作成します。ファイルが存在する場合は、情報がファイルに追加されません。

ASNLOAD 出口ルーチンを使ったターゲット表のリフレッシュ (z/OS):

ASNLOAD 出口ルーチンは、カーソルに基づくフェッチを行い、ソースからデータを入力し、ターゲットにデータをロードする、LOAD ユーティリティーを呼び出します。

ASNLOAD 出口ルーチンは LOG NO を指定して LOAD を使用し、表スペースの

COPYPEND 状況のリセットします。サンプルの ASNLOAD ソース・コードを CHANGE して、ロード・オプションを CHANGE できます。ソースは、2 つのヘッダー・ファイルと 3 つの C++ プログラムから構成されています。

手順:

提供されたままの ASNLOAD ルーチンを使用する場合は、**loadxit=y** パラメーターを使用してアプライ・プログラムを始動します。

サンプルの ASNLOAD ルーチンの変更したバージョンを使用するには、以下のようにします。

1. サイトの要件に合うように、ASNLOAD ルーチンを調整します。この出口ルーチンの変更方法については、サンプル・プログラム SASNAJCL(ASNLOAD) の PROLOG セクションを参照してください。
2. プログラムをコンパイル、リンク、バインドし、実行可能ファイルを適切なディレクトリに入れます。アプライ・プランに ASNLOAD パッケージを追加します。
 - a. 以下の条件が満たされていることを確認してください。
 - ユーティリティ・サポートを含む DB2 Universal Database for z/OS および DB2 Universal Database for OS/390 バージョン 7 以上がインストールされている。
 - DSNUTILS ストアード・プロシージャが実行されている。DSNUTILS は WLM 環境で実行する必要があります。DSNUTILS の使用方法の詳細は、『DB2 (OS/390 & z/OS) ユーティリティ・ガイド』(SC88-8773) を参照してください。
 - b. **connect(1)** プリコンパイラー・オプションを指定して、2 つのヘッダー・ファイルと 3 つのプログラムをプリコンパイルします。プリコンパイル、コンパイル、リンク、およびバインド用のサンプルの JCL がサンプルの ASNLOAD プログラムから提供されています。
 - c. ASNLOAD 出口ルーチンをリンク・エディットします。アプライ・プログラムが APF 許可ライブラリーから実行される場合に備えて、**SETCODE AC(1)** を指定してコンパイルおよびリンクする必要があります。
 - d. ASNLOAD 出口ルーチンを DSNUTILS およびアプライ・パッケージとバインドします。

サンプルの ASNLOAD は **LOG NO** を指定してロードを実行した後、表スペースを修理して **nocopypend** を設定します。表スペースのバックアップは行いません。ASNLOAD は、アプライ・プログラムのインスタンスを実行しているユーザー ID の元に一時ファイルを作成します。また、ロードに関するすべての情報を含むファイルも作成します。
3. ユーザーが指定したコードを使用して取り込みが行われるメンバーについては、**loadx_type = 2** を設定します。

4. **loadxit=y** パラメーターを使用してアプライ・プログラムを始動し、ASNLOAD 出口ルーチン呼び出しします。

ASNLOAD 出口ルーチンへの入力を構成するには、498 ページの『ASNLOAD に構成ファイルを使用』を参照してください。

ASNLOAD 出口ルーチンから生成されるファイル: これらのファイルは、ASNLOAD 出口ルーチン呼び出ししたアプライ・インスタンスの **apply_path** ディレクトリーまたは HLQ に保管されます。

- *userid.apply_qual.LOADMSG*

このファイルは、ロード統計を含めて、障害、警告メッセージおよび情報メッセージを保持します。ASNLOAD 出口ルーチンがこのファイルを作成します。ファイルが存在する場合は、情報がファイルに追加されます。

- *userid.apply_qual.LOADTRC*

トレースがオンの場合、このファイルはトレース情報を保持します。ASNLOAD 出口ルーチンがこのファイルを作成します。ファイルが存在する場合は、情報がファイルに追加されます。

ASNLOAD 出口の動作のカスタマイズ (UNIX、Windows、z/OS): レプリケーション構成をカスタマイズして ASNLOAD 出口ルーチンの動作を構成したり、出口コード自体をカスタマイズすることができます。レプリケーション構成は、サブスクリプション・メンバー (IBMSNAP_SUBS_MEMBR) 表の列を更新するか、構成ファイルを更新することによりカスタマイズできます。

サブスクリプション・メンバー表の使用: サブスクリプション・メンバー (IBMSNAP_SUBS_MEMBR) 表の列を使用して、ASNLOAD 出口ルーチンの動作をカスタマイズできます。LOADX_TYPE を使用してロード・オプションを選択します。LOADX_TYPE の有効な値は以下のとおりです。

NULL (デフォルト)

z/OS の場合: クロスローダー・ユーティリティーを使用します。

UNIX、Windows の場合: ASNLOAD 出口ルーチンが適切なユーティリティーを決定します (オプション 3、4、または 5)。

- 1 このメンバーについては ASNLOAD 出口ルーチン呼び出ししません。
このメンバーについて ASNLOAD 出口ルーチン呼び出したくない場合は、LOADX_TYPE を 1 に設定します。
- 2 ユーザー自身の出口ロジックを提供します。
ユーザー自身のロジックを ASNLOAD 出口ルーチンで提供する場合は、ASNLOAD 出口ルーチンにより取り込みを行うサブスクリプション・セットのメンバーについて LOADX_TYPE を 2 に設定してください。LOADX_TYPE を 2 に設定しているのに出口ロジックを提供しないと、出口は失敗します。

3 クロスローダー・ユーティリティを使用します。

UNIX および Windows プラットフォームの場合、クロスローダー・ユーティリティはターゲットにロードするデータをフェッチするために SELECT ステートメントを必要とします。このステートメントは、ロード対象のターゲット表に対してローカルである DB2 表を参照できます。また、ニックネームを使用して、リモートの表を参照できます。レプリケーション・ソースがターゲット表から見てリモートの DB2 表である場合は、ターゲット・サーバー・データベースに DB2 ソース表のニックネームを作成する必要があります。また、サブスクリプション・メンバー (IBMSNAP_SUBS_MEMBR) 表の LOADX_SRC_N_OWNER および LOADX_SRC_N_TABLE 列にニックネーム所有者および表を指定する必要があります。DB2 以外のレプリケーション・ソースを使用する場合は、レプリケーション・コントロール表にユーティリティで利用できるニックネームがすでに含まれているため、LOADX_SRC_N_OWNER および LOADX_SRC_N_TABLE に情報を追加する必要はありません。

4 (UNIX および Windows のみ)

EXPORT ユーティリティと LOAD ユーティリティの組み合わせを使用します。

5 (UNIX および Windows のみ)

EXPORT ユーティリティと IMPORT ユーティリティの組み合わせを使用します。

ASNLOAD に構成ファイルを使用: ASNLOAD 出口ルーチンへの入力を構成するために、オプションの構成ファイルを使用できます。このファイルは ASNLOAD の実行に必須ではありません。

UNIX および Windows プラットフォームの場合、構成ファイルのファイル名は `asnload.ini` にする必要があります。ASNLOAD 出口ルーチンは、**apply_path** ディレクトリでこのオプションの構成ファイルを探します。サンプル・ファイル `sqllib/samples/repl/asnload.ini` を編集し、ASNLOAD 出口ルーチンを呼び出したアプリ・プログラムのインスタンスの **apply_path** ディレクトリに保管してください。

z/OS プラットフォームの場合、構成ファイルは `apply_qualifer.ASNLOAD.INI` という名前のカタログ順次ファイルにする必要があります。ASNLOAD 出口ルーチンへの入力を構成するには、このオプション・ファイルを作成するために提供されているサンプルの JCL を使用してください。

ASNLOAD 出口ルーチンを使ったターゲット表のリフレッシュ (OS/400): アプリ・プログラムの代わりに出口ルーチンを使用して、もっと効率よくフル・リフレッシュを行います。たとえば、ソース表の各列および各行をターゲット表にコピーしている場合、分散データ管理 (DDM) ファイルおよびファイルのコピー (CPYF) CL コマ

ンドを使用する出口ルーチンのフル・リフレッシュを、ソース表からターゲット表へとファイル全体をコピーするように設計することができます。

手順:

ASNLOAD 出口ルーチンを使用してターゲット表をリフレッシュするには、FULLREFPGM パラメーターを使用してアプライ・プログラムを始動します。

サンプルの ASNLOAD ルーチンの変更したバージョンを使用するには、以下のようになります。

1. サイトの要件に合うように、ASNLOAD 出口ルーチンを調整します。この出口ルーチンの変更方法については、サンプル・プログラムの PROLOG セクションを参照してください。C、COBOL、および RPG 言語のソースを使用することができます。

コンパイラ言語	ライブラリー名	ソース・ファイル名	メンバー名
C	QDP4	QCSRC	ASNLOAD
COBOL	QDP4	QCBLLSRC	ASNLOAD
RPG	QDP4	QRPGLESRC	ASNLOAD

2. プログラムをコンパイル、リンク、バインドし、実行可能ファイルを適切なディレクトリに入れます。

アプライ・プログラムを妨害しないようにするため、出口ルーチンが(呼び出し側の活動化グループではなく)新しい活動化グループを使うようにコンパイルします。

名前付き活動化グループまたは新しい活動化グループを使って出口ルーチンをコンパイルすることができます。パフォーマンスを向上させるには、名前付き活動化グループを使用します。名前付き活動化グループを使用すると、出口ルーチンは必要に応じて変更をコミットまたはロールバックする必要があります。アプライ・プログラムは、(終了するまで) 変更をコミットまたはロールバックすることはありません。出口ルーチンは、完了時に変更を明示的にコミットするか、または変更を暗黙的にコミットするためにコンパイルされる必要があります。出口ルーチンの完了時にコミットされていない変更はすべて、以下のいずれかが行われるまでコミットされることはありません。

- アプライ・プログラムが、同じ活動化グループを使って別の出口ルーチンを呼び出す。
- アプライ・プログラムに応じて開始したジョブが終了する。

3. FULLREFPGM パラメーターを ASNLOAD プログラムの名前に設定してアプライ・プログラムを始動します。

始動されたアプライ・プログラムは、ユーザーから指定された ASNLOAD 出口ルーチンを使用します。別の ASNLOAD 出口ルーチンを使用させたい場合は、アプライ・プログラムを終了してから再始動します。

ASNLOAD 出口ルーチンを実行すると、すべてのターゲット表が 1 つずつリフレッシュされます。

関連タスク:

- 613 ページの『レプリケーション・プログラムの操作 (z/OS)』
- 634 ページの『Windows Service Control Manager を使用してシステム・コマンドを発行する方法 (Windows)』

関連資料:

- レプリケーションのガイドおよびリファレンス の『asnsrct: キャプチャー、アプライ、またはレプリケーション・アラート・モニターを開始する DB2 レプリケーション・サービスの作成 (Windows のみ)』
- レプリケーションのガイドおよびリファレンス の『asnacmd: アプライの操作 (UNIX、Windows、z/OS)』
- 533 ページの『asnapply: アプライの開始 (UNIX、Windows、z/OS)』
- レプリケーションのガイドおよびリファレンス の『ENDDPRAPY: アプライ・プログラムの停止 (OS/400)』
- レプリケーションのガイドおよびリファレンス の『STRDPRAPY: アプライ・プログラムの始動 (OS/400)』

ASN.IBMSNAP_SUBS_COLS

サーバー: アプライ・コントロール・サーバー

重要: SQL を使用してこの表を更新するときには、注意してください。変更の方法が不適切だと、予期せぬ結果が生じたり、データが失われたりします。

サブスクリプション列表には、サブスクリプション・セット内にコピーされるサブスクリプション・セット・メンバーの列に関する情報が含まれます。一対のソース表とターゲット表で、1 つまたは複数の列の情報が変更されると、この表で自動的に行が挿入または削除されます。

この表は、サブスクリプション・セット・メンバーの特定の列に関する情報が必要な場合に使用してください。

表 149 では、サブスクリプション列表の列の要旨を示します。

表 149. サブスクリプション列表の列

列名	説明
APPLY_QUAL	このサブスクリプション・セット・メンバーを処理するアプライ・プログラムを一意的に識別します。
SET_NAME	このメンバーが所属するサブスクリプション・セットの名前。

表 149. サブスクリプション列表の列 (続き)

列名	説明
WHOS_ON_FIRST	<p>update-anywhere レプリケーション・シナリオでは、処理順序をコントロールするために以下の値を使用します。</p> <p>F (first の略) ソース表がレプリカであり、ターゲット表がマスターです。レプリカ表とマスター表の間で更新の矛盾が生じた場合、レプリカの側の矛盾するトランザクションは拒否されません。F は読み取り専用のサブスクリプションでは使用されません。update-anywhere で使用されるものです。</p> <p>S (second の略) ソース表はマスター表またはその他のソースであり、ターゲット表はレプリカまたはその他のコピーです。マスター表とレプリカ表の間で更新の矛盾が生じた場合、レプリカの側の矛盾するトランザクションは拒否されます。S は、すべての読み取り専用サブスクリプションについて使用されます。</p>
TARGET_OWNER	ターゲット表またはビューの高位修飾子。
TARGET_TABLE	データがアプライされる表またはビュー。
COL_TYPE	<p>列のタイプを示すフラグ。</p> <p>A 変更後イメージ列。</p> <p>B 変更前イメージ列。</p> <p>C スカラー関数を使用する算出列または SQL 式。</p> <p>D DATALINK 列。</p> <p>F 列関数を使用する算出列。</p> <p>L LOB 標識値。</p> <p>P 変更前イメージ述部列。</p> <p>R システムから提供され、主キー列として使用される相対レコード番号列。DB2 DataPropagator for iSeries でのみ使用されません。</p>
TARGET_NAME	<p>ターゲット表またはビューの列の名前。ソース列名と一致する必要はありません。</p> <p>内部の CCD 列名は変更できません。それらの名前はソース表の列名と一致している必要があります。</p>

IBMSNAP_SUBS_COLS

表 149. サブスクリプション列表の列 (続き)

列名	説明
IS_KEY	列がターゲット・キーの一部であるかどうかを示すフラグ。ターゲット・キーは、コンデンス・ターゲット表のユニーク索引、または主キーのいずれかです。 Y 列はターゲット・キー全体であるか、ターゲット・キーの一部です。 N 列は、ターゲット・キーの一部ではありません。
COLNO	オリジナル・ソースにおける列の数値ロケーション。これは、表示およびサブスクリプションで他のユーザー列に対して相対的に保たれます。
EXPRESSION	ターゲット列の内容を作成するために使用された SQL 式、またはソース列名。

ASN.IBMSNAP_SUBS_MEMBR

サーバー: アプライ・コントロール・サーバー

重要: SQL を使用してこの表を更新するときには、注意してください。変更の方法が不適切だと、予期せぬ結果が生じたり、データが失われたりします。

サブスクリプション・メンバー表には、サブスクリプション・セットに対して定義された、ソース表とターゲット表の個々のペアに関する情報が含まれています。ユーザーがサブスクリプション・セット・メンバーを追加すると、この表に 1 つの行が自動的に挿入されます。

この表は、サブスクリプション・セット内で特定のソース表とターゲット表の対を指定するために使用します。

表 150 では、サブスクリプション・メンバー表の列の要旨を示します。

表 150. サブスクリプション・メンバー表の列

列名	説明
APPLY_QUAL	このサブスクリプション・セット・メンバーを処理するアプライ・プログラムを一意的に識別します。
SET_NAME	このメンバーが所属するサブスクリプション・セットの名前。

表 150. サブスクリプション・メンバー表の列 (続き)

列名	説明
WHOS_ON_FIRST	<p>update-anywhere レプリケーション・シナリオでは、処理順序をコントロールするために以下の値を使用します。</p> <p>F (first の略) ソース表がレプリカであり、ターゲット表がマスターです。レプリカ表とマスター表の間で更新の矛盾が生じた場合、レプリカの側の矛盾するトランザクションは拒否されません。F は読み取り専用のサブスクリプションでは使用されません。update-anywhere で使用されるものです。</p> <p>S (second の略) ソース表はマスター表またはその他のソースであり、ターゲット表はレプリカまたはその他のコピーです。マスター表とレプリカ表の間で更新の矛盾が生じた場合、レプリカの側の矛盾するトランザクションは拒否されます。S は、すべての読み取り専用サブスクリプションについて使用されます。</p>
SOURCE_OWNER	このメンバーのソース表またはビューの高位修飾子。
SOURCE_TABLE	このメンバーのソース表またはビューの名前。
SOURCE_VIEW_QUAL	登録表内の類似した例を突き合わせることによって、物理表のビューをサポートします。この値は、ソースとして定義されている物理表の場合は 0 に、ソースとして定義されているビューの場合は 0 より大きい値に設定されます。同じ SOURCE_OWNER および SOURCE_TABLE 列値を持つ別々のソース・ビューについて複数のサブスクリプションをサポートするためにこの列を使用します。
TARGET_OWNER	このメンバーのターゲット表またはビューの高位修飾子。
TARGET_TABLE	このメンバーのターゲット表またはビューの名前。
TARGET_CONDENSED	<p>以下のことを示すフラグ。</p> <p>Y 特定の主キー値に関して、ターゲット表に表示される行は 1 つだけです。</p> <p>N 完全な更新履歴を保持して、すべての変更を残す必要があります。</p> <p>A ターゲット表は、基礎集約表または変更集約表です。</p>
TARGET_COMPLETE	<p>以下のことを示すフラグ。</p> <p>Y ターゲット表は、関係するそれぞれの主キー値につき 1 つの行を保持します。</p> <p>N ターゲット表は、主キー値の行のサブセットを保持します。</p>

IBMSNAP_SUBS_MEMBR

表 150. サブスクリプション・メンバー表の列 (続き)

列名	説明
TARGET_STRUCTURE	ターゲット表の構造。 1 ユーザー表 3 CCD 表 4 ポイント・イン・タイム表 5 基礎集約表 6 変更集約表 7 レプリカ 8 ユーザー・コピー
PREDICATES	TARGET_TABLE 列内の表用の WHERE 文節に入れられる述部をリストします。この WHERE 文節は、ソース表の行サブセットを作成します。述部が認識されるのは、WHOS_ON_FIRST が S に設定されている場合だけです。アプライ・プログラムは ORDER BY 文節を生成できないため、述部に ORDER BY 文節を含めることはできません。集約表は、ダミー述部とその後に続く GROUP BY 文節を必要としています。 アプライ・プログラムはフル・リフレッシュおよび変更キャプチャー・レプリケーションの両方でこれらの述部を使用するため、この列に、CD 表または UOW 表の列に関係する述部を含めることはできません。CD 表または UOW 表の参照を含む述部は、UOW_CD_PREDICATES 列に保管されます。
MEMBER_STATE	メンバーの状態を示すフラグ。 N (New の略) メンバーはこのサブスクリプション・セットの新規メンバーです。 L (Loaded の略) このサブスクリプション・セットのメンバーがロードされましたが、変更キャプチャー・サイクルはまだ発生していません。 S (Synchronized の略) このサブスクリプション・セットのメンバーは、整合した同期点値に同期化されています。

表 150. サブスクリプション・メンバー表の列 (続き)

列名	説明
TARGET_KEY_CHG	<p>ユーザーがソース表で、ターゲット表のターゲット・キー列のソース列を変更したときに、アプライ・プログラムが更新をどのように処理するかを示すフラグ。</p> <p>Y アプライ・プログラムは、ターゲット・キー列の変更前イメージに基づいてターゲット表を更新します。つまり、アプライ・プログラムは述部を新しい値ではなく、古い値に変更します。ターゲット・キーの変更前イメージが CD 表の中に存在するように、変更前イメージの各列を登録してください。登録表の中の対応する登録エントリーについて、CHG_UPD_TO_DEL_INS 列の値が N に設定されていることを確認してください。</p> <p>N アプライ・プログラムは、ターゲット・キーを構成する列が更新されていないという仮定に基づくロジックを使用して、更新および削除を処理します。</p>
JOIN_UOW_CD	<p>アプライ・プログラムがユーザー・コピー・ターゲット表を処理するときに、CD 表および UOW 表の結合を行うかどうかを示すフラグ。このフラグは、CD 表の中にある、UOW 表の列を使用する述部を持つサブスクリプション・セット・メンバーをユーザーが定義したときに必要になります。ターゲット表のタイプがユーザー・コピー以外のものである場合には、アプライ・プログラムはメンバーの処理時に CD 表および UOW 表の結合を使用するため、メンバーの処理時にはこの列は無視されます。</p> <p>Y アプライ・プログラムはメンバーの処理時に CD 表および UOW 表の結合を使用します。</p> <p>N アプライ・プログラムはメンバーの処理時に CD 表および UOW 表の結合を使用しません。CD 表からのみ変更が読み取られます。</p> <p>NULL アプライ・プログラムはメンバーの処理時にこの列を無視します。ターゲット表がユーザー・コピーであり、この列の値が NULL である場合には、アプライ・プログラムはメンバーの処理時に CD 表および UOW 表を結合しません。</p>
UOW_CD_PREDICATES	<p>アプライ・プログラムが変更キャプチャー・レプリケーションにのみ必要とし、フル・リフレッシュには必要としない、CD 表または UOW 表からの列を含む述部が入っています。アプライ・プログラムは変更キャプチャー・レプリケーション時には、この列の述部と、PREDICATES 列の述部を処理します。アプライ・プログラムはフル・リフレッシュ時には、PREDICATES 列の述部のみを処理します。</p>

IBMSNAP_SUBS_MEMBER

表 150. サブスクリプション・メンバー表の列 (続き)

列名	説明
LOADX_TYPE	<p>このメンバーのロードのタイプ。この列の値は、デフォルトのオーバーライドに使用されます。</p> <p>NULL</p> <p>z/OS の場合: クロス・ローダー機能 (DB2 Utilities Suite で使用可能) が、このメンバーに対して使用されます。</p> <p>UNIX および Windows の場合: ASNLOAD 出口が、このメンバーに最適なユーティリティを決定します。(オプション 3、4 または 5)</p> <p>1 このメンバーには ASNLOAD は使用されません。これにより、ユーザーが始動時に LOADX を指定した場合にでも、特定のサブスクリプション・セット・メンバーについて ASNLOAD オプションがオフになります。</p> <p>2 ユーザー定義の、またはユーザーが変更した ASNLOAD 終了コードが使用されます。</p> <p>3 クロス・ローダー機能 (DB2 Utilities Suite で使用可能) が、このメンバーに対して使用されます。</p> <p>4 UNIX および Windows の場合のみ: このメンバーには EXPORT/LOAD が使用されます。</p> <p>5 UNIX および Windows の場合のみ: このメンバーには EXPORT/IMPORT が使用されます。</p>
LOAD_SRC_N_OWNER	<p>ユーザーが作成したニックネームの所有者。以下の場合はこの値が必須です。</p> <ul style="list-style-type: none">クロス・ローダー機能 (DB2 Utilities Suite で使用可能) が、このメンバーに対して使用されます。(LOADX_TYPE は 3)ターゲット・サーバーは UNIX または Windows である。ソースにまだニックネームがない。
LOAD_SRC_N_TABLE	<p>ユーザーが作成したニックネーム表。以下の場合はこの値が必須です。</p> <ul style="list-style-type: none">クロス・ローダー機能 (DB2 Utilities Suite で使用可能) が、このメンバーに対して使用されます。(LOADX_TYPE は 3)ターゲット・サーバーは UNIX または Windows である。ソースにまだニックネームがない。

ASN.IBMSNAP_SUBS_SET

サーバー: アプライ・コントロール・サーバー

重要: SQL を使用してこの表を更新するときには、注意してください。変更の方法が不適切だと、予期せぬ結果が生じたり、データが失われたりします。

サブスクリプション・セット表には、アプライ・コントロール・サーバーで定義されたすべてのサブスクリプション・セットが記載され、これらのセットのレプリケーションの進行状況が文書化されます。行は、サブスクリプション・セット定義を作成したときに自動的にこの表に挿入されます。

表 151 では、サブスクリプション・セット表の列の要旨を示します。

表 151. サブスクリプション・セット表の列

列名	説明
APPLY_QUAL	このサブスクリプション・セットを処理するアプライ・プログラムを一意的に識別します。
SET_NAME	サブスクリプション・セットの名前。
SET_TYPE	セットが読み取り専用か読み取り / 書き込みかを示すフラグ。 R セットは読み取り専用です。 U セットは update-anywhere 構成であるため、読み取り / 書き込みです。
WHOS_ON_FIRST	update-anywhere レプリケーション・シナリオでは、処理順序をコントロールするために以下の値を使用します。 F (first の略) ソース表がレプリカであり、ターゲット表がマスターです。レプリカ表とマスター表の間で更新の矛盾が生じた場合、レプリカの側の矛盾するトランザクションは拒否されます。F は読み取り専用のサブスクリプションでは使用されません。update-anywhere で使用されるものです。 S (second の略) ソース表はマスター表またはその他のソースであり、ターゲット表はレプリカまたはその他のコピーです。マスター表とレプリカ表の間で更新の矛盾が生じた場合、レプリカの側の矛盾するトランザクションは拒否されます。S は、すべての読み取り専用サブスクリプションについて使用されます。

IBMSNAP_SUBS_SET

表 151. サブスクリプション・セット表の列 (続き)

列名	説明
ACTIVATE	アプライ・プログラムが次のサイクルでこのセットを処理するかどうかを示すフラグ。 0 サブスクリプション・セットは非活動化されています。アプライ・プログラムはこのセットを処理しません。 1 サブスクリプション・セットは無期限にアクティブです。アプライ・プログラムは、ユーザーがセットを非活動化するか、アプライ・プログラムが処理を行えなくなるまで、各アプライ・サイクルでセットを処理します。 2 サブスクリプション・セットは、1つのアプライ・サイクルでのみアクティブです。アプライ・プログラムはセットを一度処理した後、セットを非活動化します。
SOURCE_SERVER	ソース表およびビューが定義されている、キャプチャー・コントロール・サーバーのデータベース名。
SOURCE_ALIAS	SOURCE_SERVER 列で指定されているキャプチャー・コントロール・サーバーに対応する DB2 Universal Database 別名。
TARGET_SERVER	ターゲットの表またはビューが保管されているサーバーのデータベース名。
TARGET_ALIAS	TARGET_SERVER 列で指定されているターゲット・サーバーの DB2 Universal Database 別名。

表 151. サブスクリプション・セット表の列 (続き)

列名	説明
STATUS	<p>特定のサイクル後のアプライ・プログラムの作業状況を表す値。</p> <p>-1 レプリケーションは失敗しました。アプライ・プログラムは適用済みの行のセット全体をバックアウトし、データはコミットされません。始動パラメーターが <code>SQLERRCONTINUE = Y</code> の場合、最後のサイクル中にアプライ・プログラムに戻される <code>SQLSTATE</code> は、<code>SQLERRCONTINUE (apply_qualifier.SQS)</code> の入力ファイルでユーザーが指定した許容エラーの 1 つではありません。</p> <p>0 アプライ・プログラムはサブスクリプション・セットを正常に処理しました。始動パラメーターが <code>SQLERRCONTINUE = Y</code> の場合、アプライ・プログラムは、ユーザーから <code>SQLERRCONTINUE</code> 始動パラメーター (<code>apply_qualifier.SQS</code> で) で指定された <code>SQL</code> エラーを検出して、行を拒否していません。</p> <p>2 アプライ・プログラムはサブスクリプション・セットを複数のサイクルで処理しています。アプライ・プログラムは、<code>MAX_SYNCH_MINUTES</code> コントロール列にしたがって分割された 1 つの論理サブスクリプションを正常に処理しました。</p> <p>16 アプライ・プログラムはサブスクリプション・セットを正常に処理し、<code>0</code> という状況に戻しました。しかしアプライ・プログラムは、ユーザーが <code>SQLERRCONTINUE</code> 始動パラメーター (<code>apply_qualifier.SQS</code> で) で指定したいくつかの <code>SQL</code> エラーを検出したため、いくつかの行を拒否しました。失敗した行の詳細は、<code>apply_qualifier.ERR</code> ファイルで確認してください。</p> <p>例: ユーザーは <code>SQLERRCONTINUE = Y</code> と設定し、<code>SQL</code> の許容される状態を <code>23502 (SQL コード -407)</code> と指定します。<code>23502</code> エラーが発生しますが、他のエラーは発生していません。アプライ・プログラムはサブスクリプション・セットの処理を終了し、状態を <code>16</code> に設定します。次の実行時に、<code>23502</code> エラーが発生した後、<code>07006 (SQL コード -301)</code> が発生します。アプライ・プログラムは今回は、サブスクリプション・セットの処理を停止し、適用済みの行のセット全体をバックアウトし、状況を <code>-1</code> に設定します (データはコミットされていないため)。</p> <p>18 アプライ・プログラムは複数のサイクルでサブスクリプション・セットを処理し、<code>2</code> という状況に戻しています。これは、<code>MAX_SYNCH_MINUTES</code> コントロール列にしたがって分割された 1 つの論理サブスクリプションが正常に処理されたことを意味します。しかし、ユーザーが <code>SQLERRCONTINUE</code> 始動パラメーター (<code>apply_qualifier.SQS</code> で) で指定した <code>SQL</code> エラーのいくつかを検出されたため、いくつかの行が拒否されています。失敗した行の詳細は、<code>apply_qualifier.ERR</code> ファイルで確認してください。</p>

IBMSNAP_SUBS_SET

表 151. サブスクリプション・セット表の列 (続き)

列名	説明
LASTRUN	サブスクリプション・セットが最後に開始された概算の時刻。アプライ・プログラムは、サブスクリプション・セットを処理するたびに LASTRUN 値を設定します。これは、アプライ・プログラムがサブスクリプション・セットの処理を開始する、アプライ・コントロール・サーバーにおけるおおよその時刻です。
REFRESH_TYPE	アプライ・プログラムにこのサブスクリプション・セットの処理を促すために使用されるスケジューリングのタイプ。 R アプライ・プログラムは時間ベースのスケジューリングを使用します。アプライ・プログラムは、 SLEEP_MINUTES の中の値を使用して、サブスクリプション・セットの処理を開始する時間を判断します。 E アプライ・プログラムはイベント・ベースのスケジューリングを使用します。アプライ・プログラムはサブスクリプション・イベント (IBMSNAP_SUBS_EVENT) 表の中の時刻値を確認して、サブスクリプション・セットの処理を開始する時間を判断します。イベントが生じないと、レプリケーション (変更のキャプチャーまたはフル・リフレッシュ) を何も開始できません。 B アプライ・プログラムは時間ベースのスケジューリングとイベント・ベースのスケジューリングの両方を使用します。このため、アプライ・プログラムは時間またはイベントの基準に基づいてサブスクリプション・セットを処理します。
SLEEP_MINUTES	サブスクリプション・セット処理の間の非活動時間 (分単位) を指定します。処理時間は、 REFRESH_TYPE が R または B の場合にのみ使用されます。値が NULL の場合は、アプライ・プログラムが継続してセットを処理します。アプライ・プログラムは、セットを可能な限り頻繁に処理しますが、また同じアプライ修飾子で、その他すべてのアクティブなサブスクリプション・セットも処理します。
EVENT_NAME	イベントの名前を表すユニークの文字ストリング。この ID は、サブスクリプション・セットに対してレプリケーションを起動したい場合にサブスクリプション・イベント表を更新するのに使用します。イベント名は、 REFRESH_TYPE が E または B の場合にのみ使用されます。
LASTSUCCESS	サブスクリプション・セットが最後に正常に処理されたときの、処理開始時点のアプライ・コントロール・サーバーのタイム・スタンプ。
SYNCHPOINT	アプライ・プログラムはこの列を使用して進行状況を記録します。サブスクリプション・セットのデータの処理が、この同期点の値まで終了していることを示します。
SYNCHTIME	アプライ・プログラムはこの列を使用して進行状況を記録します。サブスクリプション・セットのデータの処理が、このタイム・スタンプまで終了していることを示します。
CAPTURE_SCHEMA	このサブスクリプション・セットのソースを処理するキャプチャー・コントロール表のスキーマ名。

表 151. サブスクリプション・セット表の列 (続き)

列名	説明
TGT_CAPTURE_SCHEMA	ターゲット表が別のサブスクリプション・セットのソースでもある場合 (multi-tier 構成中の外部の CCD 表、または update-anywhere 構成中のレプリカ表など) は、この列には、表がソースとして機能するとき使用されるキャプチャー・スキーマが含まれます。
FEDERATED_SRC_SRVR	DB2 以外のリレーショナル・ソースの場合にのみアプライされる、サブスクリプション・セットのソースである、連合リモート・サーバーの名前。
FEDERATED_TGT_SRVR	DB2 以外のリレーショナル・ターゲットの場合にのみアプライされる、サブスクリプション・セットのターゲットである、連合リモート・サーバーの名前。
JRN_LIB	OS/400 キャプチャー・サーバーにのみアプライされるこの列は、ソース表が使用するジャーナルのライブラリー名です。
JRN_NAME	OS/400 キャプチャー・サーバーにのみアプライされるこの列は、ソース表が使用するジャーナルの名前です。この列で、アスタリスクの後ろに 9 つのブランクが続くときには、ソース表が現在ジャーナルの中に入っていないことを意味します。この場合は、このソース表のデータをキャプチャーすることはできません。
OPTION_FLAGS	DB2 レプリケーションの将来のオプション用に予約済み。この列には現在、デフォルト値の NNNN が入っています。
COMMIT_COUNT	<p>アプライ・プログラムがサブスクリプション・セットに対して実行する処理のタイプを示すフラグ。</p> <p>NULL これは、読み取り専用のサブスクリプション・セットの場合のデフォルト設定です。アプライ・プログラムは、すべてのデータが処理されるまで、<i>n</i> のサブスクリプション・セット・メンバーのメンバー 1 つずつについて、フェッチした応答セットを処理していき、セット全体のデータ処理が終了すると最後に、1 つのコミットを発行します。この COMMIT_COUNT 設定値を使用する利点は、処理が早く終了する可能性があることです。</p> <p>非 NULL の整数</p> <p>アプライ・プログラムはトランザクションのモードでサブスクリプション・セットを処理します。すべての応答セットがフェッチされると、各トランザクションが IBMSNAP_INTENTSEQ 値の順に並べられ、コミット・シーケンスの順番で予備ファイルの内容が適用されます。このタイプの処理では、すべての予備ファイルを同時に開いて処理できます。この列で指定された数のトランザクションの後に、コミットが発行されます。たとえば、1 は各トランザクションの後にコミットすることを意味し、2 は、トランザクション 2 つごとにコミットすることを意味します。0 という整数は、すべてのフェッチ・データがアプライされた後に 1 つのコミットを発行することを意味します。トランザクションのモードの処理を使用する利点は、この処理ではターゲットでの参照保全制約が可能であり、暫定のコミットを発行できることです。</p>

IBMSNAP_SUBS_SET

表 151. サブスクリプション・セット表の列 (続き)

列名	説明
MAX_SYNCH_MINUTES	サブスクリプション・サイクルでフェッチおよびアプライする変更データの量を規制するための時間しきい値限度。アプライ・プログラムはキャプチャー・サーバーの UOW 表または CCD 表の中の IBMSNAP_LOGMARKER に基づいてサブスクリプション・セットの処理をミニサイクルに分割し、ミニサイクルが正常に終了するたびに、ターゲット・サーバーで COMMIT を発行します。設定された限度が不適切になるようなリソース制約をアプライ・プログラムが検出した場合、この限度は自動的に再計算されます。1 より小さい MAX_SYNCH_MINUTES 値は、NULL の MAX_SYNCH_MINUTES 値と同じものとして処理されます。
AUX_STMTS	サブスクリプション・ステートメント (IBMSNAP_SUBS_STMTS) 表で定義する SQL ステートメントの数。これらのステートメントは、アプライ・プログラムがサブスクリプション・セットを処理する前または後に実行可能です。
ARCH_LEVEL	行に含まれる定義の構造レベル。この列は、行を作成する基礎となった規則を識別します。このレベルは、IBM で定義されていて、バージョン 8 では 0801 です。

キャプチャー・コントロール・サーバーで使用される表のリスト

キャプチャー・コントロール・サーバーに保管される表には、ユーザーの登録済みソースに関する情報と、キャプチャー・プログラムまたはトリガーがソースを処理する方法に関する情報が入っています。UNIX、Windows、および z/OS の場合は、レプリケーション・センターを使用して、ユーザーの指定に合わせてこれらのコントロール表を作成します。OS/400 の場合は、DataPropagator for iSeries のインストール時に、ASN ライブラリーの中にこれらのコントロール表が自動的に作成されます。OS/400 上でレプリケーションのシステム・コマンドを使用して、代替のキャプチャー・スキーマ内にキャプチャー・コントロール表を作成できます。

表 152. キャプチャー・コントロール・サーバーで使用される表のクイック・リファレンス

表名	説明	トピックを参照
ASN.IBMSNAP_CAPSCHEMAS	キャプチャー・スキーマ表 すべてのキャプチャー・スキーマの名前を保持します。	ASN.IBMSNAP_CAPSCHEMAS
schema.IBMSNAP_AUTHTKN (OS/400)	アプライ修飾子相互参照表 update-anywhere をサポートするための情報が入っています。	.IBMSNAP_AUTHTKN (OS/400)

表 152. キャプチャー・コントロール・サーバーで使用される表のクイック・リファレンス (続き)

表名	説明	トピックを参照
<i>schema</i> .IBMSNAP_CAPENQ (UNIX、Windows、 z/OS)	<p>キャプチャー・エンキュー表</p> <p>この表は、各キャプチャー・スキーマについて、以下のことを保証するために使用されます。</p> <ul style="list-style-type: none"> • DB2 for UNIX および DB2 for Windows の場合は、1 つのデータベースに対して 1 つのキャプチャー・プログラムのみが実行される。 • データを共有しない DB2 for z/OS の場合は、1 つのサブシステムに対して 1 つのキャプチャー・プログラムのみが実行される。 • データを共有する DB2 for z/OS の場合は、1 つのデータ共有グループに対して 1 つのキャプチャー・プログラムのみが実行される。 	.IBMSNAP_CAPENQ (UNIX、Windows、 z/OS)
<i>schema</i> .CD_table	<p>変更データ (CD) 表</p> <p>ソースに発生する変更に関する情報を保持します。この表は、レプリケーション・ソースが登録されるまでは作成されません。</p>	.CD_table
<i>schema</i> .CCD_table	<p>整合変更データ (CCD) 表</p> <p>ソースに発生する変更に関する情報と、これらの変更の順序を識別するための追加の列を保持します。</p>	.CCD_table (DB2 以外)
<i>schema</i> .IBMSNAP_CAPMON	<p>キャプチャー・モニター表</p> <p>キャプチャー・プログラムの進行状況をモニターする助けとなる操作統計を保持します。</p>	.IBMSNAP_CAPMON

表 152. キャプチャー・コントロール・サーバーで使用される表のクイック・リファレンス (続き)

表名	説明	トピックを参照
<i>schema</i> .IBMSNAP_CAPPARMS	<p>キャプチャー・パラメーター表</p> <p>キャプチャー・プログラムの操作をコントロールするためにユーザーが指定できるパラメーターを保持します。</p>	.IBMSNAP_CAPPARMS
<i>schema</i> .IBMSNAP_CAPTRACE	<p>キャプチャー・トレース表</p> <p>キャプチャー・プログラムからの重要なメッセージを保持します。</p>	.IBMSNAP_CAPTRACE (DB2 のみ)
<i>schema</i> .IBMSNAP_PRUNE_LOCK	<p>プルーニング・ロック表</p> <p>コールド・スタート、または保存限度プルーニング (保存限度に達したか、超えたときのプルーニング) 時に、キャプチャー・プログラムの CD 表へのアクセスをシリアライズするために使用されま</p>	.IBMSNAP_PRUNE_LOCK
<i>schema</i> .IBMSNAP_PRUNE_SET	<p>プルーニング・セット表</p> <p>CD 表のプルーニングを調整しま</p>	.IBMSNAP_PRUNE_SET
<i>schema</i> .IBMSNAP_PRUNCNTL	<p>プルーニング・コントロール表</p> <p>キャプチャー・プログラムとアプリ・プログラムの間で同期点更新を調整します。</p>	.IBMSNAP_PRUNCNTL
<i>schema</i> .IBMSNAP_REG_EXT (OS/400)	<p>登録拡張表</p> <p>登録表を拡張したものです。ジャーナル名やリモート・ソース表のデータベース・エントリー名など、レプリケーション・ソースに関する追加情報が含まれます。</p>	.IBMSNAP_REG_EXT (OS/400)

表 152. キャプチャー・コントロール・サーバーで使用される表のクイック・リファレンス (続き)

表名	説明	トピックを参照
<i>schema</i> .IBMSNAP_REGISTER	登録表 レプリケーション・ソース表の名前、その属性、および対応する CD 表および CCD 表の名前など、レプリケーション・ソースに関する情報が入ります。	.IBMSNAP_REGISTER
<i>schema</i> .IBMSNAP_REG_SYNCH (DB2 以外のリレーショナル)	登録同期表 DB2 以外のリレーショナル・データ・ソースから複製するとき 사용됩니다。この表の更新トリガーは、アプライ・プログラムが登録表から情報を読み込む前に、登録表のすべての行で SYNCHPOINT 値の更新を始めることにより、キャプチャー・プログラムをシミュレートします。	.IBMSNAP_REG_SYNCH (DB2 以外のリレーショナル)
<i>schema</i> .IBMSNAP_RESTART	再始動表 キャプチャー・プログラムがログまたはジャーナル内の正しい時点からキャプチャーを再開できるようにするための情報が入っています。OS/400 環境では、この表は RCVJRNE (ジャーナル・エントリーの受信) コマンドの開始時刻を判別するためにも使用されます。	.IBMSNAP_RESTART
<i>schema</i> .IBMSNAP_SEQTABLE (DB2 以外のリレーショナル)	順序付け表 Informix 表のログ・シーケンス番号と同等のものとして DB2 レプリケーションから使用される一連のユニーク番号を保持します。	.IBMSNAP_SEQTABLE (Informix)

表 152. キャプチャー・コントロール・サーバーで使用される表のクイック・リファレンス (続き)

表名	説明	トピックを参照
<i>schema</i> .IBMSNAP_SIGNAL	シグナル表 キャプチャー・プログラムにプロンプトを出すために使用されるすべてのシグナルを保持します。これらのシグナルは手動で、またはアプライ・プログラムから送信できます。	.IBMSNAP_SIGNAL
<i>schema</i> .IBMSNAP_UOW	作業単位 (UOW) 表 ソース表にコミットされたトランザクションに関する追加情報を提供します。	.IBMSNAP_UOW

関連資料:

- レプリケーションのガイドおよびリファレンス の『ASN.IBMSNAP_CAPSCHEMAS』
- レプリケーションのガイドおよびリファレンス の『.IBMSNAP_AUTHTKN (OS/400)』
- 517 ページの『*schema*.IBMSNAP_CAPENQ (UNIX、Windows、z/OS)』
- レプリケーションのガイドおよびリファレンス の『.IBMSNAP_CAPMON』
- レプリケーションのガイドおよびリファレンス の『.IBMSNAP_CAPPARMS』
- 521 ページの『*schema*.IBMSNAP_CAPTRACE (DB2 のみ)』
- レプリケーションのガイドおよびリファレンス の『.CCD_table (DB2 以外)』
- レプリケーションのガイドおよびリファレンス の『.CD_table』
- レプリケーションのガイドおよびリファレンス の『.IBMSNAP_PRUNCNTL』
- レプリケーションのガイドおよびリファレンス の『.IBMSNAP_PRUNE_LOCK』
- レプリケーションのガイドおよびリファレンス の『.IBMSNAP_PRUNE_SET』
- レプリケーションのガイドおよびリファレンス の『.IBMSNAP_REG_EXT (OS/400)』
- レプリケーションのガイドおよびリファレンス の『.IBMSNAP_REGISTER』
- レプリケーションのガイドおよびリファレンス の『.IBMSNAP_REG_SYNCH (DB2 以外のリレーショナル)』
- レプリケーションのガイドおよびリファレンス の『.IBMSNAP_RESTART』
- レプリケーションのガイドおよびリファレンス の『.IBMSNAP_SEQTABLE (Informix)』
- 522 ページの『*schema*.IBMSNAP_SIGNAL』
- レプリケーションのガイドおよびリファレンス の『.IBMSNAP_UOW』

schema.IBMSNAP_CAPENQ (UNIX、Windows、z/OS)

サーバー: キャプチャー・コントロール・サーバー

デフォルト・スキーマ: ASN

重要: SQL を使用してこの表を更新するときには、注意してください。変更の方法が不適切だと、予期せぬ結果が生じたり、データが失われたりします。

キャプチャー・エンキュー表は、DB2 以外のリレーショナル・サーバー、または OS/400 サーバーでは使用されません。

1 つのキャプチャー・スキーマについて、キャプチャー・エンキュー表は以下のことを保証します。

- DB2 for UNIX および DB2 for Windows の場合は、1 つのデータベースに対して 1 つのキャプチャー・プログラムのみが実行される。
- データを共有しない DB2 for z/OS の場合は、1 つのサブシステムに対して 1 つのキャプチャー・プログラムのみが実行される。
- データを共有する DB2 for z/OS の場合は、1 つのデータ共有グループに対して 1 つのキャプチャー・プログラムのみが実行される。

キャプチャー・プログラムは実行中、この表を排他的にロックします。

表 153 では、キャプチャー・エンキュー表の要旨を示します。

表 153. キャプチャー・エンキュー表の列

列名	説明
LOCKNAME	この列にはデータは含まれません。

asnanalyze: アナライザーの操作 (UNIX および Windows)

レプリケーション・コントロール表の状態についてのレポートを生成するには、**asnanalyze** コマンドを使用します。このコマンドは、任意のオペレーティング・システムにあるレプリケーション・コントロール表を分析します。OS/400 オペレーティング・システムでも可能ですが、コマンドは UNIX または Windows から呼び出す必要があります。

コマンドを呼び出す場合、**asnanalyze** コマンドと最初のパラメーターの間にはスペースを 1 つ入れる必要があります。パラメーターなしでコマンドを実行すると、画面にコマンド・ヘルプが表示されます。

asnanalyze コマンドを使用してアナライザーを操作するための構文

asnanalyze

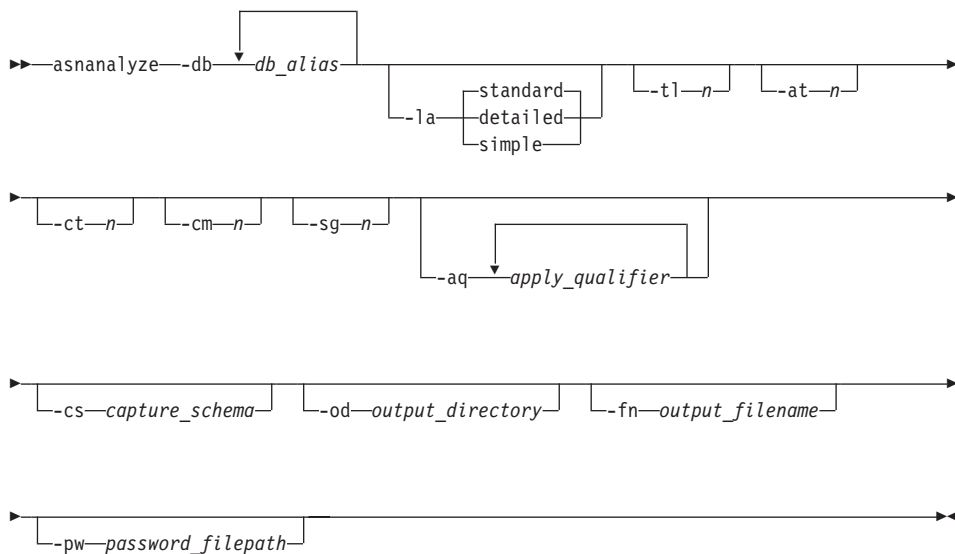


表 154 は、呼び出しパラメーターを定義します。

表 154. UNIX および Windows オペレーティング・システム用 `asnanalyze` 呼び出しパラメーター定義

パラメーター	定義
-db <code>db_alias</code>	キャプチャー・コントロール・サーバー、ターゲット・サーバー、およびアプライ・コントロール・サーバーを指定します。 データベース別名を少なくとも 1 つ指定する必要があります。複数のデータベース別名がある場合は、ブランク・スペースを使用して値を区切ります。

表 154. UNIX および Windows オペレーティング・システム用 asnanalyze 呼び出しパラメーター定義 (続き)

パラメーター	定義
-la <i>level_of_analysis</i>	<p>報告される分析のレベルを指定します。</p> <p>standard (デフォルト) コントロール表の内容および、キャプチャー・プログラムおよびアプライ・からの状況情報を含むレポートを生成します。</p> <p>detailed 標準レポートの情報に加えて以下の情報を生成します。</p> <ul style="list-style-type: none"> 変更データ (CD) 表および作業単位 (UOW) 表のプルーニング情報。 DB2 for z/OS 表スペースのパーティション化および圧縮情報 サブスクリプション・キーのためのターゲット索引の分析 <p>simple 標準レポートで情報を生成しますが、サブスクリプション列 (IBMSNAP_SUBS_COLS) 表からの詳細は含まれません。</p>
-tl <i>n</i>	アプライ・トレール (IBMSNAP_APPLYTRAIL) 表から検索するエントリーの日付範囲 (0 から 30 日) を指定します。デフォルトは 3 日です。
-at <i>n</i>	アプライ・トレース (IBMSNAP_APPLYTRACE) 表から検索するエントリーの日付範囲 (0 から 30 日) を指定します。デフォルトは 3 日です。
-ct <i>n</i>	キャプチャー・トレース (IBMSNAP_CAPTRACE) 表から検索するエントリーの日付範囲 (0 から 30 日) を指定します。デフォルトは 3 日です。
-cm <i>n</i>	キャプチャー・モニター (IBMSNAP_CAPMON) 表から検索するエントリーの日付範囲 (0 から 30 日) を指定します。デフォルトは 3 日です。
-sg <i>n</i>	シグナル (IBMSNAP_SIGNAL) 表から検索するエントリーの日付範囲 (0 から 30 日) を指定します。デフォルトは 3 日です。

表 154. UNIX および Windows オペレーティング・システム用 *asnanalyze* 呼び出しパラメーター定義 (続き)

パラメーター	定義
-aq <i>apply_qualifier</i>	分析する特定のサブスクリプション・セットを識別するアプライ修飾子を指定します。 複数のアプライ修飾子を指定することができます。複数のアプライ修飾子がある場合は、ブランク・スペースを使用して値を区切ります。アプライ修飾子を指定しないと、指定されたデータベース別名のすべてのサブスクリプション・セットが分析されます。
-cs <i>capture_schema</i>	分析するキャプチャー・スキーマの名前を指定します。 このパラメーターを使用する場合、指定できるキャプチャー・スキーマは 1 つだけです。
-od <i>output_directory</i>	アナライザー・レポートを保管するディレクトリーを指定します。デフォルトは、現行ディレクトリーです。
-fn <i>output_filename</i>	アナライザー・レポート出力を含むファイルの名前を指定します。 アナライザーを実行するオペレーティング・システムのファイル名規則を使用します。ファイル名がすでに存在する場合、ファイルは上書きされます。デフォルトのファイル名は、 <i>asnanalyze.htm</i> です。
-pw <i>password_filepath</i>	パスワード・ファイルの名前とパスを指定します。このパラメーターを指定しないと、アナライザーは現行ディレクトリーで <i>asnpwd.aut</i> ファイルを探します。

asnanalyze の例

以下の例は、**asnanalyze** コマンドの使用法を示しています。

例 1: *proddb1* という名前のデータベース上のレプリケーション・コントロール表を分析する例です。

```
asnanalyze -db proddb1
```

例 2: *proddb1* と *proddb2* のデータベース上のレプリケーション・コントロール表について、詳細レベルの分析を入手する例です。

```
asnanalyze -db proddb1 proddb2 -la detailed
```

例 3: *proddb1* と *proddb2* のデータベース上の、*IBMSNAP_APPLYTRAIL*、*IBMSNAP_APPLYTRACE*、*IBMSNAP_CAPTRACE*、*IBMSNAP_CAPMON*、および *IBMSNAP_SIGNAL* 表から、最新の 2 日の情報を分析する例です。

```
asnanalyze -db proddb1 proddb2 -t1 2 -at 2 -ct 2 -cm 2 -sg 2
```

例 4: proddb1 および proddb2 データベース上の、IBMSNAP_APPLYTRAIL、IBMSNAP_APPLYTRACE、IBMSNAP_CAPTRACE、IBMSNAP_CAPMON、および IBMSNAP_SIGNAL 表から、qual1 および qual2 のアプライ修飾子についてのみ、最新の 2 日の情報を simple レベルで分析するには、以下のように指定します。

```
asnanalyze -db proddb1 proddb2 -la simple -t1 2 -at 2 -ct 2 -cm 2 -sg 2
-aq qual1 qual2 -od c:%mydir -fn anzout -pw c:%SQLLIB
```

このコマンド例は、アナライザーの出力を c:%mydir ディレクトリーの下に anzout ファイルに書き込み、c:%SQLLIB ディレクトリーからパスワード情報を使用します。

例 5: 特定のキャプチャー・スキーマを分析する例です。

```
asnanalyze -db proddb1 proddb2 -cs BSN
```

例 6: コマンド・ヘルプを表示するには、以下のように入力します。

```
asnanalyze
```

関連資料:

- レプリケーションのガイドおよびリファレンス の『ANZDPR: アナライザーの操作 (OS/400)』

schema.IBMSNAP_CAPTRACE (DB2 のみ)

サーバー: キャプチャー・コントロール・サーバー

デフォルト・スキーマ: ASN

キャプチャー・トレース表は、キャプチャー・プログラムから重要なメッセージを保持します。

以下の 2 つの表は、オペレーティング・システム別のキャプチャー・トレース表のレイアウトを示しています。

表 155. UNIX、Windows、および z/OS のキャプチャー・トレース表の列

列名	説明
OPERATION	キャプチャー・プログラムの操作のタイプ (たとえば、初期化、キャプチャー、またはエラー条件)。
TRACE_TIME	キャプチャー・トレース表に行が挿入されたときの、キャプチャー・コントロール・サーバーにおける時刻。
説明	メッセージ ID とメッセージ・テキスト。エラー・メッセージ、警告メッセージ、または情報メッセージです。この列に入れられるテキストは英語のみです。

IBMSNAP_CAPTRACE

表 156. OS/400 のキャプチャー・トレース表の列

列名	説明								
OPERATION	初期化、キャプチャー、またはエラー条件など、キャプチャー・プログラムで実行された操作のタイプ。								
TRACE_TIME	キャプチャー・トレース表に行が挿入された時刻。トレース限度ブルーニングの対象となる TRACE_TIME 行は、キャプチャー・プログラムが CD 表および UOW 表のブルーニングを行うときに削除されます。								
JOB_NAME	このトレース・エントリーを書き込んだジョブの完全修飾名。 <table border="1"><thead><tr><th>位置</th><th>説明</th></tr></thead><tbody><tr><td>1-10</td><td>キャプチャー・スキーマ名またはジャーナル・ジョブ名</td></tr><tr><td>11-20</td><td>キャプチャー・プログラム を始動したユーザーの ID</td></tr><tr><td>21-26</td><td>ジョブ番号</td></tr></tbody></table>	位置	説明	1-10	キャプチャー・スキーマ名またはジャーナル・ジョブ名	11-20	キャプチャー・プログラム を始動したユーザーの ID	21-26	ジョブ番号
位置	説明								
1-10	キャプチャー・スキーマ名またはジャーナル・ジョブ名								
11-20	キャプチャー・プログラム を始動したユーザーの ID								
21-26	ジョブ番号								
JOB_STR_TIME	JOB_NAME 列で指定されたジョブの開始時刻。								
説明	メッセージ ID とメッセージ・テキスト。メッセージ ID は DESCRIPTION 列の最初の 7 文字です。メッセージ・テキストは、DESCRIPTION 列の位置 9 から始まります。								

schema.IBMSNAP_SIGNAL

サーバー: キャプチャー・コントロール・サーバー

デフォルト・スキーマ: ASN

この表の情報は、SQL を使って更新できます。

シグナル表には、キャプチャー・プログラムに特定のアクションを実行するように促すシグナルが保管されます。シグナルは、ユーザーまたはアプライ・プログラムから入力されます。

シグナル表は DATA CAPTURE CHANGES 属性を指定して作成されます。つまり、この表に対して実行されたすべての挿入、更新、および削除操作は、DB2 リカバリーログから読み取られたログ・レコードのように、キャプチャー・プログラムから見るすることができます。キャプチャー・プログラムは、シグナル表の更新および削除ログ・レコードはすべて無視しますが、シグナル挿入の、有効に作成およびコミットされたログ・レコードはすべて、注意が必要な「シグナル」として認識します。シグナル挿入によるログ・レコードに対してキャプチャー・プログラムが実行するアクションは、その挿入に関してシグナル表がどのように指定されているかによって異なります。シグナル表の中の値は、取るべきアクションをキャプチャー・プログラムに指示します。

この表の中で、完全を表す、`SIGNAL_STATE` 値が `C` のレコード、またはまたは保持制限プルーニングの対象となるタイム・スタンプを持つレコードは、キャプチャー・プログラムによるプルーニングが行われると削除されます。

表 157 では、シグナル表の列の要旨を示します。

表 157. シグナル表の列

列名	説明
<code>SIGNAL_TIME</code>	行を一意的に識別するために使用されるタイム・スタンプ。キャプチャー・プログラムはこのユニーク値を使用して、キャプチャー・シグナルの処理が終了した時刻を示す、シグナル表の中の正しい行を検出します。タイム・スタンプ列は <code>NOT NULL WITH DEFAULT</code> として作成されるため、キャプチャー・シグナルは一般的に、 <code>DB2</code> で現行タイム・スタンプが <code>SIGNAL_TIME</code> 値として提供されるのと同じ方法で挿入できます。
<code>SIGNAL_TYPE</code>	通知されたシグナルのタイプを示すフラグ。 <p>CMD ユーザー、アプライ・プログラム、またはよく知られたシステム・コマンドまたはシグナルである、別のアプリケーションから通知されたシグナル。使用可能なシグナルのサブタイプのリストについては、この表の <code>SIGNAL_SUBTYPE</code> 列を参照してください。</p> <p>USER ユーザーから通知されたシグナル。キャプチャー・プログラムは、<code>SIGNAL_LSN</code> 列の値を、シグナルが挿入されたときのログの <code>LSN</code> で更新し、<code>SIGNAL_STATE</code> 列の値を、<code>P</code> (保留) から <code>R</code> (受信) に更新します。</p>

表 157. シグナル表の列 (続き)

列名	説明
SIGNAL_SUBTYPE	<p>システム・コマンドからのシグナルが発生したときに (SIGNAL_TYPE = CMD)、キャプチャー・プログラムが実行するアクション。</p> <p>CAPSTART</p> <p>キャプチャー・プログラムは、SIGNAL_INPUT_IN 列の中の MAP_ID (IBMSNAP_PRUNCNTL 表から) で識別される、特定のサブスクリプション・セット・メンバーの登録済みソースでの変更のキャプチャーを開始します。たとえば、アプライ・プログラムはセット内のすべてのターゲット表に対してフル・リフレッシュを実行する前にこのシグナルを発行して、このセットで変更キャプチャー・レプリケーションを行う準備ができていることをキャプチャー・プログラムに知らせます。アプライ・プログラムがこのシグナルを通知します。</p> <p>STOP</p> <p>キャプチャー・プログラムは変更のキャプチャーを停止して終了します。このコマンドは、ユーザーからのみ発行できます。アプライ・プログラムからは発行できません。</p> <p>CAPSTOP</p> <p>キャプチャー・プログラムは、SIGNAL_INPUT_IN 列の中の <i>source_owner.source_table</i> で識別される特定の登録済みソースの変更のキャプチャーを停止します。このコマンドは、ユーザーからのみ発行できます。アプライ・プログラムからは発行できません。</p> <p>UPDANY</p> <p>アプライ・プログラム (SIGNAL_INPUT_IN 列の中のアプライ修飾子で識別される) は、update-anywhere 構成で、2 つのキャプチャー・プログラムを使用していることを、キャプチャー・プログラムに知らせます。アプライ・プログラムがこのシグナルを通知します。</p> <p>シグナル・タイプが USER の場合、シグナル・サブタイプは使用されな いか、キャプチャー・プログラムから認識されないため、これは必要フ ィールドではありません。どのような値に設定してもかまいません。</p>

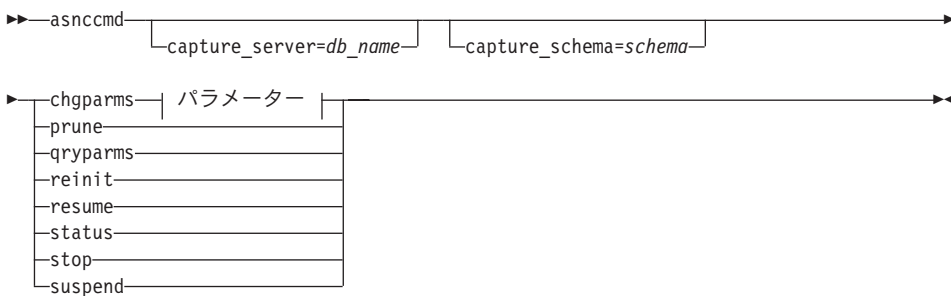
表 157. シグナル表の列 (続き)

列名	説明
SIGNAL_INPUT_IN	<p>SIGNAL_TYPE = USER の場合、この列は、ユーザー定義の入力を保持します。SIGNAL_TYPE = CMD の場合は、この値の意味は、このシグナルの SIGNAL_SUBTYPE によって異なります。</p> <p>CMD + CAPSTART マッピング ID。DB2 以外のリレーショナル・ソースは、キャプチャー・プログラムではなく、キャプチャー・トリガーによって処理されるため、シーケンス内の次の値で枝取り制御 (IBMSNAP_PRUNCNTL) 表を更新する、IBMSNAP_SIGNAL 表の更新後に起動される、SIGNAL_TRIGGER というトリガーがあります。</p> <p>CMD + UPDANY update-anywhere 構成でアプライ・プログラムを識別するためのアプライ修飾子。</p> <p>CMD + CAPSTOP キャプチャー・プログラムによる変更キャプチャーを停止する必要のある、ソース所有者およびソース表の名前。 (<i>source_owner.source_table</i>)</p>
SIGNAL_STATE	<p>シグナルの状況を示すフラグ。</p> <p>P シグナルは保留。キャプチャー・プログラムはまだシグナルを受け取っていません。ユーザーがシグナルを通知するときには、SIGNAL_STATE を P に設定してください。</p> <p>R キャプチャー・プログラムはシグナルを受け取りました。キャプチャー・プログラムは、SIGNAL_TYPE = USER であるか、SIGNAL_TYPE = CMD および SIGNAL_SUBTYPE = STOP であるシグナルを受け取ると、SIGNAL_STATE を R に設定します (完全を示す C に変更するのではなく)。</p> <p>C キャプチャー・プログラムはシグナルの処理を完了しました。キャプチャー・プログラムは、SIGNAL_TYPE = CMD の場合、STOP を除くすべての SIGNAL_SUBTYPE について、この値を C に設定します。</p>
SIGNAL_LSN	<p>コミット・レコードのログ・シーケンス番号。この値は、キャプチャー・プログラムからのみ設定されます。</p>

asnccmd: キャプチャーの操作 (UNIX、Windows、z/OS)

UNIX、Windows、および z/OS 上の UNIX System Services (USS) でキャプチャー・プログラムを操作するには、 **asnccmd** コマンドを使用します。このコマンドは、オペレーティング・システムのプロンプトまたはシェル・スクリプト内で実行します。

asnccmd コマンドを使用してキャプチャー・プログラムを操作するための構文



パラメーター:

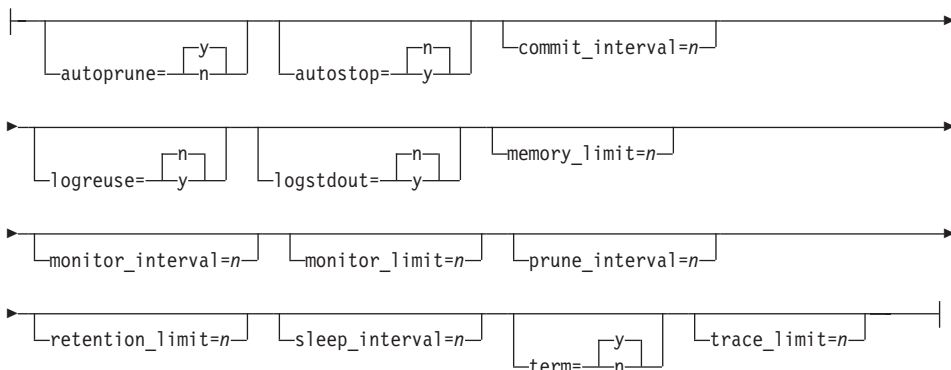


表 158 は、呼び出しパラメーターを定義します。

表 158. UNIX、Windows、および z/OS オペレーティング・システム用 *asnccmd* 呼び出しパラメーター定義

パラメーター	定義
capture_server=db_name	<p>キャプチャー・コントロール・サーバーの名前を指定します。</p> <p>UNIX および Windows の場合: キャプチャー・コントロール・サーバーを指定しないと、このパラメーターはデフォルトで DB2DBDFT 環境変数の値になります。</p> <p>z/OS の場合: コントロール・サーバーに接続するデータベース・サーバーの名前です。データ共有の場合、グループ・アタッチ名を使用しないでください。その代わりに、メンバー・サブシステム名を指定してください。</p>
capture_schema=schema	<p>特定のキャプチャー・プログラムを識別するために使用するキャプチャー・スキーマの名前を指定します。入力するスキーマ名の長さは 1 から 30 文字でなければなりません。デフォルトは ASN です。</p>
chgparms	<p>キャプチャー・プログラムの実行中にそのパラメーター値を変更することを指定します。新しいパラメーター値を指定するかまたは、キャプチャー・プログラムの始動時に渡された値をオーバーライドすることができます。どのパラメーターをオーバーライドできるかについては、529 ページの表 159 を参照してください。</p>
prune	<p>変更データ (CD) 表、作業単位 (UOW) 表、キャプチャー・モニター (IBMSNAP_CAPMON) 表、キャプチャー・トレース (IBMSNAP_CAPTRACE) 表、およびシグナル (IBMSNAP_SIGNAL) 表の整理を一度行う場合にこのパラメーターを指定します。このコマンドが正常にキューに入ると、キャプチャー・プログラムはメッセージを出します。</p>
qryparms	<p>現行の稼働パラメーター値を標準出力 (stdout) に書き込む場合に指定します。</p>

表 158. UNIX、Windows、および z/OS オペレーティング・システム用 `asnccmd` 呼び出しパラメーター定義 (続き)

パラメーター	定義
reinit	<p>キャプチャー・プログラムに、新しく追加されたレプリケーション・ソースを登録 (IBMSNAP_REGISTER) 表から入手させる場合に指定します。たとえば、新しいレプリケーション・ソースを追加した場合、またはキャプチャー・プログラムの実行中に、ALTER ADD を使用してレプリケーション・ソースと変更データ (CD) 表に列を追加した場合に、このパラメーターを使用します。</p> <p>重要: キャプチャー・プログラムの実行中に、レプリケーション・ソースの取り消しまたはレプリケーション・ソース表のドロップを行った後で、キャプチャー・プログラムを再初期化するために、reinit パラメーターを使用しないでください。この場合は、キャプチャー・プログラムを停止してから、startmode パラメーターに warmsa、warmns、または warmsi を指定した asnccap コマンドを使用して、再始動してください。</p>
resume	<p>中断したキャプチャー・プログラムにデータのキャプチャーを再開させることを指定します。</p>
status	<p>それぞれのキャプチャー・スレッド (管理、プルーニング、シリアライゼーション、およびワーカー) の状態を示すメッセージを受け取ることを指定します。</p>
stop	<p>キャプチャー・プログラムを通常どおりに停止して、その時点までに処理したログ・レコードをコミットすることを指定します。</p>
suspend	<p>ピーク時に、キャプチャー・プログラムの環境を損なわずにすむ範囲内で、運用トランザクションにオペレーティング・システムのリソースを解放することを指定します。</p> <p>重要: レプリケーション・ソースを取り消すためにキャプチャーを中断しないでください。中断するのではなく、キャプチャー・プログラムを停止してください。</p>

表 159 は、chgparms 呼び出しパラメーターを定義しています。

表 159. UNIX、Windows、および z/OS オペレーティング・システム用 *asnccmd* *chgparms* パラメーター定義

パラメーター	定義
autoprun =y/n	<p>変更データ (CD) 表、作業単位 (UOW) 表、キャプチャー・モニター (IBMSNAP_CAPMON) 表、キャプチャー・トレース (IBMSNAP_CAPTRACE) 表、およびシグナル (IBMSNAP_SIGNAL) 表の中の行の自動プルーニングを可能にするかどうかを指定します。</p> <p>y (デフォルト) キャプチャー・プログラムは、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表で指定されたインターバルで、適格な行の自動プルーニングを行います。キャプチャー・プログラムは、行が複製されたものか否かに関係なく、保持制限より古い CD、UOW、および IBMSNAP_SIGNAL の行を削除します。</p> <p>n 自動プルーニングは使用不可になります。</p>
autostop =y/n	<p>キャプチャー・プログラムの始動前にログに記録されたすべてのトランザクションを検索した後、キャプチャー・プログラムを終了させるかどうかを指定します。</p> <p>n (デフォルト) キャプチャー・プログラムは、トランザクションを検索した後、終了しません。</p> <p>y キャプチャー・プログラムは、トランザクションを検索した後、終了します。</p>
commit_interval =n	<p>キャプチャー・プログラムが、何秒待ってから、作業単位 (UOW) 表および変更データ (CD) 表に行をコミットするかを示す秒数を指定します。デフォルトは 30 秒です。</p>

表 159. UNIX、Windows、および z/OS オペレーティング・システム用 *asnccmd* *chgpargs* パラメーター定義 (続き)

パラメーター	定義
logreuse =y/n	<p>キャプチャー・プログラムがログ・ファイルを再利用するか、またはログ・ファイルにメッセージを付加するかを指定します (ログ・ファイル名は <i>db2instance.capture_server.capture_schema.CAP.log</i>)。</p> <p>n (デフォルト) キャプチャー・プログラムは、キャプチャー・プログラムの再始動後であってもログ・ファイルにメッセージを付加します。</p> <p>y キャプチャー・プログラムは、まず現行のログ・ファイルを切り捨て、次にキャプチャー・プログラムの再始動時に新しいログを開始して、ログ・ファイルを再利用します。</p> <p>chgpargs パラメーターを使用してこのパラメーターを <i>y</i> に変更すると、ログは即時に切り捨てられ、再利用されます。この場合の logreuse パラメーターの変更は、キャプチャー・プログラムの次の始動には影響しません。</p> <p>z/OS の場合: ログ・ファイル名には DB2 インスタンス名は含まれません (<i>capture_server.capture_schema.CAP.log</i>)。</p>
logstdout =y/n	<p>キャプチャー・プログラムがメッセージをどこに送信するかを指定します。</p> <p>n (デフォルト) キャプチャー・プログラムはログ・ファイルにのみメッセージを送信します。</p> <p>y キャプチャー・プログラムは、ログ・ファイルと標準出力 (stdout) の両方にメッセージを送信します。</p>
memory_limit =n	<p>トランザクションを作成するためにキャプチャー・プログラムが使用できるメモリーの最大サイズ (MB 単位) を指定します。このメモリー限度に達すると、キャプチャー・プログラムはトランザクションをファイルに書き出します。デフォルトは 32 MB です。</p>
monitor_interval =n	<p>キャプチャー・プログラムが、キャプチャー・モニター (IBMSNAP_CAPMON) 表に行を挿入する頻度 (秒単位) を指定します。デフォルトは 300 秒 (5 分) です。</p>

表 159. UNIX、Windows、および z/OS オペレーティング・システム用 *asnccmd* *chgpargs* パラメーター定義 (続き)

パラメーター	定義
monitor_limit=n	キャプチャー (IBMSNAP_CAPMON) 表内の行は、何分たったら、プルーニングの対象として適格になるかを示す分数を指定します。 monitor_limit パラメーターの値よりも古い、すべての IBMSNAP_CAPMON 行は、次のプルーニング・サイクルで削除されます。デフォルトは 10,080 分 (7 日) です。
prune_interval=n	変更データ (CD) 表、作業単位 (UOW) 表、キャプチャー・モニター (IBMSNAP_CAPMON) 表、キャプチャー・トレース (IBMSNAP_CAPTRACE) 表、およびシグナル (IBMSNAP_SIGNAL) 表の整理の頻度 (秒数) を指定します。 <i>autoprune</i> パラメーターを n に設定した場合は、このパラメーターは無視されます。デフォルトは 300 秒 (5 分) です。
retention_limit=n	変更データ (CD) 表、作業単位 (UOW) 表、またはシグナル (IBMSNAP_SIGNAL) 表の行は、何分たったら、プルーニングの対象として適格になるかを示す分数を指定します。 retention_limit パラメーターの値よりも古い行は、次のプルーニング・サイクルで削除されます。デフォルトは 10,080 分 (7 日) です。
sleep_interval=n	<p>キャプチャー・プログラムが、アクティブ・ログの処理を終了し、バッファが空であると判断するまで、何秒スリープするかを示す秒数を指定します。デフォルトは 5 秒です。</p> <p>z/OS の場合: バッファが半分以上空いていることを戻した後、キャプチャー・プログラムが何秒スリープするかを指定します。</p>

表 159. UNIX、Windows、および z/OS オペレーティング・システム用 *asnccmd* *chgpargs* パラメーター定義 (続き)

パラメーター	定義
term=y/n	<p>DB2 が終了した場合にキャプチャー・プログラムは終了するかどうかを指定します。</p> <p>y (デフォルト) DB2 が終了した場合、キャプチャー・プログラムは終了します。</p> <p>n DB2 が MODE(QUIESCE) で終了した場合、キャプチャー・プログラムは実行を継続します。DB2 が始動されると、キャプチャー・プログラムは warm モードで始動し、DB2 が終了した時点からのキャプチャーを開始します。</p> <p>DB2 が FORCE または異常終了で終了した場合は、このパラメーターを n にしてもキャプチャー・プログラムは終了します。</p> <p>このパラメーターを n にし、制限付きアクセス (ACCESS MAINT) を使用して DB2 を始動すると、キャプチャー・プログラムは接続できないので、結果として終了します。</p>
trace_limit=n	<p>キャプチャー・トレース (IBMSNAP_CAPTRACE) 表の行は、何分たったら、プルーニングの対象として適格になるかを示す分数を指定します。 trace_limit パラメーターの値よりも古い、すべての IBMSNAP_CAPTRACE 行が、次のプルーニング・サイクルで削除されます。デフォルトは 10,080 分 (7 日) です。</p>

asnccmd の例

以下の例は、**asnccmd** コマンドの使用法を示しています。

例 1: 新しく追加したレプリケーション・ソースを、実行中のキャプチャー・プログラムに認識させるには、以下のように入力します。

```
asnccmd capture_server=db capture_schema=ASN reinitt
```

例 2: CD、UOW、IBMSNAP_CAPMON、IBMSNAP_CAPTRACE、および IBMSNAP_SIGNAL 表の整理を 1 回行う例です。

```
asnccmd capture_server=db capture_schema=ASN prune
```

例 3: それぞれのキャプチャー・スレッドの状態についてメッセージを受け取る場合の例です。


```
asnccmd capture_server=db capture_schema=ASN status
```

例 4: キャプチャー・プログラムの現行操作値を標準出力に送信する例です。

```
asnccmd capture_server=db capture_schema=ASN qryparms
```

例 5: キャプチャー・プログラムの実行中に自動ブルーニングを行わないようにするには、以下のように入力します。

```
asnccmd capture_server=db capture_schema=ASN chgparms autoprune=n
```

例 6: キャプチャー・プログラムの実行を停止するには、以下のように入力します。

```
asnccmd capture_server=db capture_schema=ASN stop
```

関連タスク:

- 613 ページの『レプリケーション・プログラムの操作 (z/OS)』

関連資料:

- レプリケーションのガイドおよびリファレンス の『OVRDPRCAPA: DPR キャプチャー属性のオーバーライド (OS/400)』

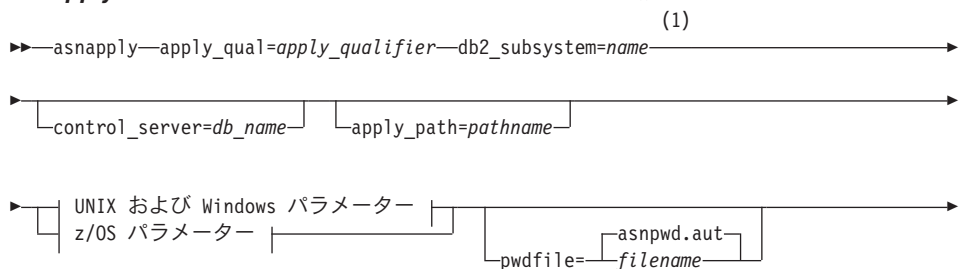
asnapply: アプライの開始 (UNIX、Windows、z/OS)

UNIX、Windows、および z/OS 上の UNIX System Services (USS) でアプライ・プログラムを始動するには、**asnapply** コマンドを使用します。このコマンドは、オペレーティング・システムのプロンプトまたはシェル・スクリプト内で実行します。

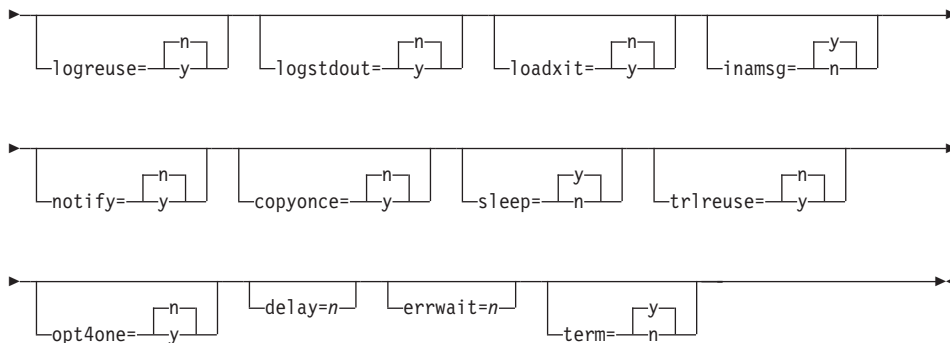
アプライ・プログラムを始動すると、以下のいずれかが生じるまで連続して稼働します。

- 通常の方法で停止する。
- オペレーターが取り消す。
- 予期しないエラーまたは障害が発生する。

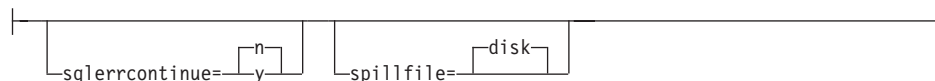
asnapply コマンドを使用してアプライ・プログラムを始動するための構文



asnapply



UNIX および Windows パラメーター:



z/OS パラメーター:



注:

- 1 db2_subsystem パラメーターは z/OS オペレーティング・システムでのみ使用しません。

表 160 は、呼び出しパラメーターを定義します。

表 160. UNIX、Windows、および z/OS オペレーティング・システム用 *asnapply* 呼び出しパラメーター定義

パラメーター	定義
apply_qual=apply_qualifier	アプライ・プログラムが、処理されるサブスクリプション・セットの識別に使用するアプライ修飾子を指定します。このパラメーターは必須です。 入力する値は、サブスクリプション・セット (IBMSNAP_SUBS_SET) 表の APPLY_QUAL 列の値と一致する必要があります。アプライ修飾子名には大文字小文字の区別があり、最大 18 文字です。

表 160. UNIX、Windows、および z/OS オペレーティング・システム用 *asnapply* 呼び出しパラメーター定義 (続き)

パラメーター	定義
db2_subsystem=name	z/OS の場合のみ: DB2 サブシステムの名前を指定します。入力する DB2 サブシステム名は最大 4 文字です。このパラメーターにはデフォルトはありません。このパラメーターは必須です。
control_server=db_name	サブスクリプション定義とアプライ・プログラム・コントロール表が存在する、アプライ・コントロール・サーバーの名前を指定します。 UNIX および Windows の場合: アプライ・コントロール・サーバーを指定しないと、このパラメーターはデフォルトで DB2DBDFT 環境変数の値になります。 z/OS の場合: コントロール・サーバー・パラメーターは、コントロール・サーバーに接続するデータベース・サーバーの名前です。
apply_path=pathname	アプライ・プログラムが使用する作業ファイルのロケーションを指定します。デフォルトは、 asnapply コマンドが呼び出されたディレクトリーです。
pwdfile=filename	パスワード・ファイルの名前を指定します。パスワード・ファイルを指定しない場合、デフォルトは asnpwd.aut です。 このコマンドは、 apply_path パラメーターで指定されたディレクトリー内でパスワード・ファイルを探します。 apply_path パラメーターを指定しないと、このコマンドは、コマンドを呼び出したディレクトリー内でパスワード・ファイルを探します。

表 160. UNIX、Windows、および z/OS オペレーティング・システム用 *asnapply* 呼び出しパラメーター定義 (続き)

パラメーター	定義
logreuse=y/n	<p>アプライ・プログラムが、ログ・ファイル (<i>db2instance.control_server.apply_qualifier.APP.log</i>) を再利用するか、またはメッセージを付加するかを指定します。</p> <p>n (デフォルト) アプライ・プログラムは、アプライ・プログラムの再始動後であっても、ログ・ファイルにメッセージを付加しません。</p> <p>y アプライ・プログラムは、ログ・ファイルを削除し、アプライ・プログラムの再始動時にそれを再作成することにより、ログ・ファイルを再利用します。</p> <p>z/OS の場合: ログ・ファイルには DB2 インスタンス名は含まれません (<i>control_server.apply_qualifier.APP.log</i>)。</p>
logstdout=y/n	<p>アプライ・プログラムがメッセージをどこに送信するかを指定します。</p> <p>n (デフォルト) アプライ・プログラムはログ・ファイルにのみメッセージを送信します。</p> <p>y アプライ・プログラムは、メッセージをログ・ファイルと標準出力 (stdout) の両方に送信します。</p>
loadxit=y/n	<p>アプライ・プログラムが ASNLOAD を呼び出すかどうかを指定します。ASNLOAD は IBM 提供の出力ルーチンであり、エクスポートおよびロード・ユーティリティーを使用して、ターゲット表をリフレッシュします。</p> <p>n (デフォルト) アプライ・プログラムは ASNLOAD を呼び出しません。</p> <p>y アプライ・プログラムは ASNLOAD を呼び出します。</p>

表 160. UNIX、Windows、および z/OS オペレーティング・システム用 *asnapply* 呼び出しパラメーター定義 (続き)

パラメーター	定義
inamsg=y/n	<p>アプライ・プログラムを非アクティブにしたとき、アプライ・プログラムからメッセージを出すかどうかを指定します。</p> <p>y (デフォルト) アプライ・プログラムは非アクティブ時にメッセージを出します。</p> <p>n アプライ・プログラムは非アクティブ時にメッセージを出しません。</p>
notify=y/n	<p>アプライ・プログラムが ASNDONE を呼び出すかどうかを指定します。ASNDONE は、アプライ・プログラムがサブスクリプション・セットのコピーを終了した時に、ユーザーにコントロールを戻すための出口ルーチンです。</p> <p>n (デフォルト) アプライ・プログラムは ASNDONE を呼び出しません。</p> <p>y アプライ・プログラムは ASNDONE を呼び出します。</p>

表 160. UNIX、Windows、および z/OS オペレーティング・システム用 *asnapply* 呼び出しパラメーター定義 (続き)

パラメーター	定義
copyonce =y/n	<p>アプライ・プログラムが呼びだされた時点で適格と見なされたサブスクリプション・セットごとに、アプライ・プログラムがコピー・サイクルを 1 回実行するかどうかを指定します。その後、アプライ・プログラムは終了します。適格と見なされるサブスクリプション・セットとは、以下の基準を満たすものです。</p> <ul style="list-style-type: none"> • サブスクリプション・セット (IBMSNAP_SUBS_SET) 表の中で (ACTIVATE > 0)。ACTIVATE 列の値がゼロより大きい場合、そのサブスクリプション・セットは無期限にアクティブであるか、または 1 回のみのサブスクリプション処理に使用されています。 • (REFRESH_TYPE = R または B) または (REFRESH_TYPE = E であり、指定されたイベントが発生)。REFRESH_TYPE 列の値は IBMSNAP_SUBS_SET 表に保管されます。 <p>サブスクリプション・セット表の MAX_SYNCH_MINUTES 限度および、サブスクリプション・イベント (IBMSNAP_SUBS_EVENT) 表の END_OF_PERIOD タイム・スタンプが指定されている場合は、これに従います。</p> <p>n (デフォルト) アプライ・プログラムは、適格なサブスクリプション・セットごとにコピー・サイクルを 1 回実行しません。</p> <p>y アプライ・プログラムは、適格なサブスクリプション・セットごとにコピー・サイクルを 1 回実行します。</p>
sleep =y/n	<p>処理の対象として適格となる新しいサブスクリプションがない場合に、アプライ・プログラムがどうするかを指定します。</p> <p>y (デフォルト) アプライ・プログラムはスリープ状態に入ります。</p> <p>n アプライ・プログラムは停止します。</p>

表 160. UNIX、Windows、および z/OS オペレーティング・システム用 *asnapply* 呼び出しパラメーター定義 (続き)

パラメーター	定義
trlreuse=y/n	<p>アプライ・プログラムの始動時に、アプライ・プログラムがアプライ・トレール (IBMSNAP_APPLYTRAIL) 表を空にするかどうかを指定します。</p> <p>n (デフォルト) アプライ・プログラムは IBMSNAP_APPLYTRAIL 表にエントリーを付加します。アプライ・プログラムは表を空にしません。</p> <p>y アプライ・プログラムはプログラム始動時に IBMSNAP_APPLYTRAIL 表を空にします。</p>
opt4one=y/n	<p>アプライ・プログラムに定義されているサブスクリプション・セットが 1 つだけの場合、アプライ・プログラムのパフォーマンスを最適化するかどうかを指定します。</p> <p>n (デフォルト) サブスクリプション・セットが 1 つの場合、アプライ・プログラムのパフォーマンスを最適化しません。</p> <p>y サブスクリプション・セットが 1 つの場合、アプライ・プログラムのパフォーマンスを最適化します。</p> <p>最適化を y に設定すると、アプライ・プログラムはサブスクリプション・セット・メンバーの情報をキャッシュに入れて再利用します。このようにサブスクリプション・セット・メンバーの情報を再利用すると、CPU 使用率が減り、スループットが向上します。</p>
delay=n	<p>連続レプリケーションを使用する場合に、それぞれのアプライ・サイクルが終了した後、何秒待つかを示す遅延時間 (秒単位) を指定します。 <i>n</i> は、0、1、2、3、4、5、または 6 です。デフォルトは 6 です。</p>

表 160. UNIX、Windows、および z/OS オペレーティング・システム用 *asnapply* 呼び出しパラメーター定義 (続き)

パラメーター	定義
errwait=<i>n</i>	<p>アプライ・プログラムがエラー状態になった後、何秒待ってから再試行するかを示す秒数 (1 から 300) を指定します。デフォルト値は 300 秒 (5 分) です。</p> <p>重要: アプライ・プログラムはほとんど切れ目なく稼動しており、アプライ・トレール (IBMSNAP_APPLYTRAIL) 表に多くの行を生成するので、ここにあまり小さい数を指定しないでください。</p>
term=<i>y/n</i>	<p>DB2 の状況がアプライ・プログラムの動作にどのように影響するかを指定します。</p> <p>y (デフォルト) DB2 が終了すると、アプライ・プログラムは終了します。</p> <p>n DB2 がアクティブでない場合、アプライ・プログラムは DB2 の始動を待ちます。</p> <p>UNIX および Windows の場合: DB2 が静止し、アプライ・プログラムがアクティブの場合、アプライ・プログラムはアクティブのままであり、DB2 が静止モードでなくなるまで再接続しません。</p> <p>z/OS の場合: DB2 が静止し、アプライ・プログラムがアクティブの場合、アプライ・プログラムはアクティブのままであり、DB2 が再度開始されるまで再接続しません。</p>

表 160. UNIX、Windows、および z/OS オペレーティング・システム用 `asnapply` 呼び出しパラメーター定義 (続き)

パラメーター	定義
<code>sqlerrcontinue=y/n</code>	<p>UNIX および Windows の場合のみ: アプライ・プログラムがある種の SQL エラーを検出した場合、アプライ・プログラムが処理を継続するかどうかを指定します。</p> <p>アプライ・プログラムは、失敗した SQLSTATE を SQLSTATE ファイルに指定された値に照らしてチェックします。この SQLSTATE ファイルは、アプライ・プログラムの実行前にユーザーが作成します。ファイルの内容と一致すれば、アプライ・プログラムは失敗した行についての情報をエラー・ファイル (<code>apply_qualifier.ERR</code>) に書き込み、処理を継続します。SQLSTATE ファイルには 5 バイト値を 20 個まで含めることができます。</p> <p>n (デフォルト) アプライ・プログラムは SQLSTATE ファイルをチェックしません。</p> <p>y アプライ・プログラムは処理中に SQLSTATE ファイルをチェックします。</p>
<code>spillfile=filetype</code>	<p>フェッチした応答セットをどこに保管するかを指定します。</p> <p>UNIX および Windows の場合、有効な値は以下のとおりです。</p> <p>disk (デフォルト) ディスク・ファイル。</p> <p>z/OS の場合、有効な値は以下のとおりです。</p> <p>mem (デフォルト) メモリー・ファイル。応答セット用の十分なメモリーがない場合、アプライ・プログラムは失敗します。</p> <p>disk ディスク・ファイル。</p>

asnapply の例

以下の例は、`asnapply` コマンドの使用法を示しています。

例 1: アプライ修飾子 AQ1 を使用し、コントロール・サーバー名は dbx で、/home/files/apply/ ディレクトリーにある作業ファイルを使用して、アプライ・プログラムを始動する例です。

```
asnapply apply_qual=AQ1 control_server=dbx apply_path=/home/files/apply/  
pwdfile=pass1.txt
```

アプライ・プログラムは /home/files/apply/ ディレクトリーで pass1.txt という名前のパスワード・ファイルを探します。

例 2: ASNLOAD 出口ルーチンを呼び出すアプライ・プログラムを始動します。

```
asnapply apply_qual=AQ1 control_server=dbx pwdfile=pass1.txt loadxit=y
```

この例では、アプライ・プログラムは pass1.txt という名前のパスワード・ファイルを現行ディレクトリーで探します。

例 3: 適格なサブスクリプション・セットごとにコピー・サイクルを 1 回実行するアプライ・プログラムを始動します。

```
asnapply apply_qual=AQ1 control_server=dbx apply_path=/home/files/apply/  
copyonce=y
```

この例では、アプライ・プログラムは /home/files/apply/ ディレクトリーで、デフォルトのパスワード・ファイル (asnpwd.aut) を探します。

関連タスク:

- 613 ページの『レプリケーション・プログラムの操作 (z/OS)』

関連資料:

- レプリケーションのガイドおよびリファレンス の『STRDPRAPY: アプライ・プログラムの始動 (OS/400)』

asncap: キャプチャーの開始 (UNIX、Windows、z/OS)

UNIX、Windows、および z/OS 上の UNIX System Services (USS) でキャプチャー・プログラムを始動するには、**asncap** コマンドを使用します。このコマンドは、レプリケーション・センターではなく、オペレーティング・システムのプロンプトまたはシェル・スクリプト内で実行します。

キャプチャー・プログラムを始動すると、停止されるかまたはリカバリー不能エラーが検出されるまで実行を続けます。

asncap コマンドを使用してキャプチャー・プログラムを始動するための構文

```
→ asncap [capture_server=db_name] [capture_schema=schema]
```

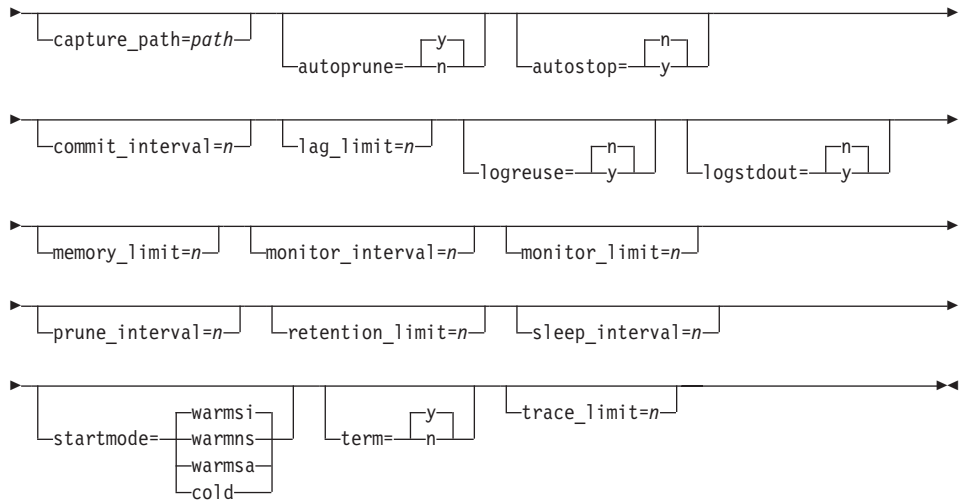


表 161 は、呼び出しパラメーターを定義します。

表 161. UNIX、Windows、および z/OS オペレーティング・システム用 asncap 呼び出しパラメーター定義

パラメーター	定義
capture_server = <i>db_name</i>	<p>キャプチャー・コントロール・サーバーの名前を指定します。</p> <p>UNIX および Windows の場合: キャプチャー・コントロール・サーバーを指定しないと、このパラメーターはデフォルトで DB2DBDFT 環境変数の値になります。</p> <p>z/OS の場合: capture_server パラメーターは、コントロール・サーバーに接続するデータベース・サーバーの名前です。データ共有の場合、グループ・アタッチ名を使用しないでください。その代わりに、メンバー・サブシステム名を指定してください。</p>
capture_schema = <i>schema</i>	<p>特定のキャプチャー・プログラムを識別するために使用するキャプチャー・スキーマの名前を指定します。入力するスキーマ名の長さは 1 から 30 文字でなければなりません。デフォルトは ASN です。</p>
capture_path = <i>path</i>	<p>キャプチャー・プログラムが使用する作業ファイルのロケーションを指定します。デフォルトは、asncap コマンドが呼び出されたディレクトリーです。</p>

表 161. UNIX、Windows、および z/OS オペレーティング・システム用 asncap 呼び出しパラメーター定義 (続き)

パラメーター	定義
autoprun =y/n	<p>変更データ (CD) 表、作業単位 (UOW) 表、キャプチャー・モニター (IBMSNAP_CAPMON) 表、キャプチャー・トレース (IBMSNAP_CAPTRACE) 表、およびシグナル (IBMSNAP_SIGNAL) 表の中の行の自動ブルーニングを可能にするかどうかを指定します。</p> <p>y (デフォルト) キャプチャー・プログラムは、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表で指定されたインターバルで、適格な行の自動ブルーニングを行います。キャプチャー・プログラムは、行が複製されたものか否かに関係なく、保持制限より古い CD、UOW、および IBMSNAP_SIGNAL の行を削除します。</p> <p>n 自動ブルーニングは使用不可になります。</p>
autostop =y/n	<p>キャプチャー・プログラムの始動前にログに記録されたすべてのトランザクションを検索した後、キャプチャー・プログラムを終了するかどうかを指定します。</p> <p>n (デフォルト) キャプチャー・プログラムは、トランザクションを検索した後、終了しません。</p> <p>y キャプチャー・プログラムは、トランザクションを検索した後、終了します。</p>
commit_interval =n	<p>キャプチャー・プログラムが、何秒待ってから、作業単位 (UOW) 表および変更データ (CD) 表に行をコミットするかを示す秒数を指定します。デフォルトは 30 秒です。</p>
lag_limit =n	<p>キャプチャー・プログラムがログ・レコードを処理するときに、許される遅れの分数を指定します。デフォルトは 10,080 分 (7 日) です。キャプチャー・プログラムは、ウォーム・スタートの場合にのみ、この値をチェックします。この限界を超えると、キャプチャー・プログラムは開始されません。</p>

表 161. UNIX、Windows、および z/OS オペレーティング・システム用 asncap 呼び出しパラメーター定義 (続き)

パラメーター	定義
logreuse=y/n	<p>キャプチャー・プログラムがログ・ファイルを再利用するか、またはログ・ファイルにメッセージを付加するかを指定します (ログ・ファイル名は <code>db2instance.capture_server.capture_schema.CAP.log</code>)。</p> <p>n (デフォルト) キャプチャー・プログラムは、キャプチャー・プログラムの再始動後であってもログ・ファイルにメッセージを付加します。</p> <p>y キャプチャー・プログラムは、まず現行のログ・ファイルを切り捨て、次にキャプチャー・プログラムの再始動時に新しいログを開始して、ログ・ファイルを再利用します。</p> <p>z/OS の場合: ログ・ファイル名には DB2 インスタンス名は含まれません (<code>capture_server.capture_schema.CAP.log</code>)。</p>
logstdout=y/n	<p>キャプチャー・プログラムがメッセージをどこに送信するかを指定します。</p> <p>n (デフォルト) キャプチャー・プログラムはログ・ファイルにのみメッセージを送信します。</p> <p>y キャプチャー・プログラムは、ログ・ファイルと標準出力 (stdout) の両方にメッセージを送信します。</p>
memory_limit=n	<p>トランザクションを作成するためにキャプチャー・プログラムが使用できるメモリの最大サイズ (MB 単位) を指定します。このメモリ限度に達すると、キャプチャー・プログラムはトランザクションをファイルに書き出します。デフォルトは 32 MB です。</p>
monitor_interval=n	<p>キャプチャー・プログラムが、キャプチャー・モニター (IBMSNAP_CAPMON) 表に行を挿入する頻度 (秒単位) を指定します。デフォルトは 300 秒 (5 分) です。</p>
monitor_limit=n	<p>キャプチャー (IBMSNAP_CAPMON) 表内の行は、何分たったら、ブルーニングの対象として適格になるかを示す分数を指定します。 monitor_limit パラメーターの値よりも古い、すべての IBMSNAP_CAPMON 行は、次のブルーニング・サイクルで削除されます。デフォルトは 10,080 分 (7 日) です。</p>

表 161. UNIX、Windows、および z/OS オペレーティング・システム用 asncap 呼び出しパラメーター定義 (続き)

パラメーター	定義
prune_interval = <i>n</i>	変更データ (CD) 表、作業単位 (UOW) 表、キャプチャー・モニター (IBMSNAP_CAPMON) 表、キャプチャー・トレース (IBMSNAP_CAPTRACE) 表、およびシグナル (IBMSNAP_SIGNAL) 表の整理の頻度 (秒数) を指定します。 autoprune パラメーターを <i>n</i> に設定した場合は、このパラメーターは無視されます。デフォルトは 300 秒 (5 分) です。
retention_limit = <i>n</i>	変更データ (CD) 表、作業単位 (UOW) 表、またはシグナル (IBMSNAP_SIGNAL) 表の行は、何分たったら、ブルーニングの対象として適格になるかを示す分数を指定します。 retention_limit パラメーターの値よりも古い行は、次のブルーニング・サイクルで削除されます。デフォルトは 10,080 分 (7 日) です。
sleep_interval = <i>n</i>	キャプチャー・プログラムが、アクティブ・ログの処理を終了し、バッファーが空であると判断するまで、何秒スリープするかを示す秒数を指定します。デフォルトは 5 秒です。 z/OS の場合: バッファーが半分以上空いていることを戻した後、キャプチャー・プログラムが何秒スリープするかを指定します。

表 161. UNIX、Windows、および z/OS オペレーティング・システム用 asncap 呼び出しパラメーター定義 (続き)

パラメーター	定義
startmode=mode	<p>キャプチャー始動時にキャプチャー・プログラムが使用する、処理プロシージャを指定します。</p> <p>warmsi (デフォルト)</p> <p>ウォーム・スタート情報を入手できる場合、キャプチャー・プログラムは、直前の実行の終了時点から処理を再開します。これがキャプチャー・プログラムの最初の始動である場合は、自動的にコールド・スタートに切り替えます。</p> <p>ウォーム・スタート中は、キャプチャー・プログラムは、キャプチャー・トレース (IBMSNAP_CAPTRACE) 表、変更データ (CD) 表、作業単位 (UOW) 表、および再始動 (IBMSNAP_RESTART) 表に手を付けずそのままにしておきます。キャプチャー・プログラムの始動後にエラーが起こった場合、キャプチャー・プログラムは終了します。</p> <p>warmms ウォーム・スタート情報を入手できる場合、キャプチャー・プログラムは、直前の実行の終了時点から処理を再開します。キャプチャー・プログラムの始動後にエラーが起こった場合、キャプチャー・プログラムは終了します。キャプチャー・プログラムは、ウォーム・スタートできない場合、コールド・スタートに切り替わりません。</p> <p>warmsa ウォーム・スタート情報を入手できる場合、キャプチャー・プログラムは、直前の実行の終了時点から処理を再開します。キャプチャー・プログラムは、ウォーム・スタートできない場合、コールド・スタートに切り替わります。</p> <p>cold キャプチャー・プログラムを始動すると、初期化中に CD、UOW、および IBMSNAP_CAPTRACE 表内のすべての列を削除します。レプリケーション・ソースに対するすべてのサブスクリプションは、次のアプライ処理のサイクルですべてリフレッシュされます。ターゲットが非コンプリート整合変更データ (CCD) 表の場合、フル・リフレッシュは行われません。</p>

表 161. UNIX、Windows、および z/OS オペレーティング・システム用 *asncap* 呼び出しパラメーター定義 (続き)

パラメーター	定義
term=y/n	<p>DB2 が終了した場合にキャプチャー・プログラムは終了するかどうかを指定します。</p> <p>y (デフォルト) DB2 が終了した場合、キャプチャー・プログラムは終了します。</p> <p>n DB2 が MODE(QUIESCE) で終了した場合、キャプチャー・プログラムは実行を継続します。DB2 が始動されると、キャプチャー・プログラムは warm モードで始動し、DB2 が終了した時点からのキャプチャーを開始します。</p> <p>DB2 が FORCE または異常終了で終了した場合は、このパラメーターを n にしてもキャプチャー・プログラムは終了します。</p> <p>このパラメーターを n にし、制限付きアクセス (ACCESS MAINT) を使用して DB2 を始動すると、キャプチャー・プログラムは接続できないので、結果として終了します。</p>
trace_limit=n	<p>キャプチャー・トレース (IBMSNAP_CAPTRACE) 表の行は、何分たったら、プルーニングの対象として適格になるかを示す分数を指定します。 trace_limit パラメーターの値よりも古い、すべての IBMSNAP_CAPTRACE 行が、次のプルーニング・サイクルで削除されます。デフォルトは 10,080 分 (7 日) です。</p>

asncap の例

以下の例は、**asncap** コマンドの使用法を示しています。

例 1: db という名前のキャプチャー・コントロール・サーバーおよび、キャプチャー・スキーマ ASN を使用し、 /home/files/capture/logs/ ディレクトリーにある作業ファイルを使用して、初めてキャプチャー・プログラムを始動する例です。

```
asncap capture_server=db capture_schema=ASN
capture_path=/home/files/capture/logs/ startmode=cold
```

例 2: キャプチャー・プログラムが停止した後、プルーニングを行わずにキャプチャー・プログラムを再始動する例です。

```
asncap capture_server=db autoprune=n sleep_interval=10 startmode=warmsa
```

この例では、キャプチャー・プログラムは、対応するコントロール表内のすべての行を保存し、アクティブ・ログの処理を終了し、バッファが空であると判断した後、10 秒

スリープします。ウォーム・スタート情報を入手できない場合、キャプチャー・プログラムは、直前の実行の終了時点から処理を再開し、コールド・スタートに切り替えます。

例 3: `warmns startmode` で、変更されたパラメーター設定値を使用して、キャプチャー・プログラムを再始動します。

```
asncap capture_server=db autoprun=y prune_interval=60 retention_limit=1440
startmode=warmns
```

このコマンドは、キャプチャー・プログラムを再始動し、新しいパラメーター設定値を使用するようにします。新しいパラメーター設定値は、CD、UOW、および `IBMSNAP_SIGNAL` 表がプルーニングの対象として適格になるまでの時間を減らし、プルーニングの頻度をデフォルトのパラメーター設定値よりも増やしています。ウォーム・スタート情報を入手できない場合、キャプチャー・プログラムは、直前の実行の終了時点から処理を再開しますが、自動的にコールド・スタートに切り替えることはしません。

例 4: その作業ファイルをすべて、`capture_files` という新しいサブディレクトリーに送信するキャプチャー・プログラムを始動します。

1. 該当のディレクトリーに行き、`capture_files` という新しいサブディレクトリーを作成します。

```
cd /home/db2inst
mkdir capture_files
```

2. キャプチャー・プログラムを始動し、たった今作成した新しいサブディレクトリーにあるキャプチャー・パスを指定します。

```
asncap capture_server=db capture_schema=ASN
capture_path=/home/db2inst/capture_files startmode=warmns
```

関連タスク:

- 613 ページの『レプリケーション・プログラムの操作 (z/OS)』

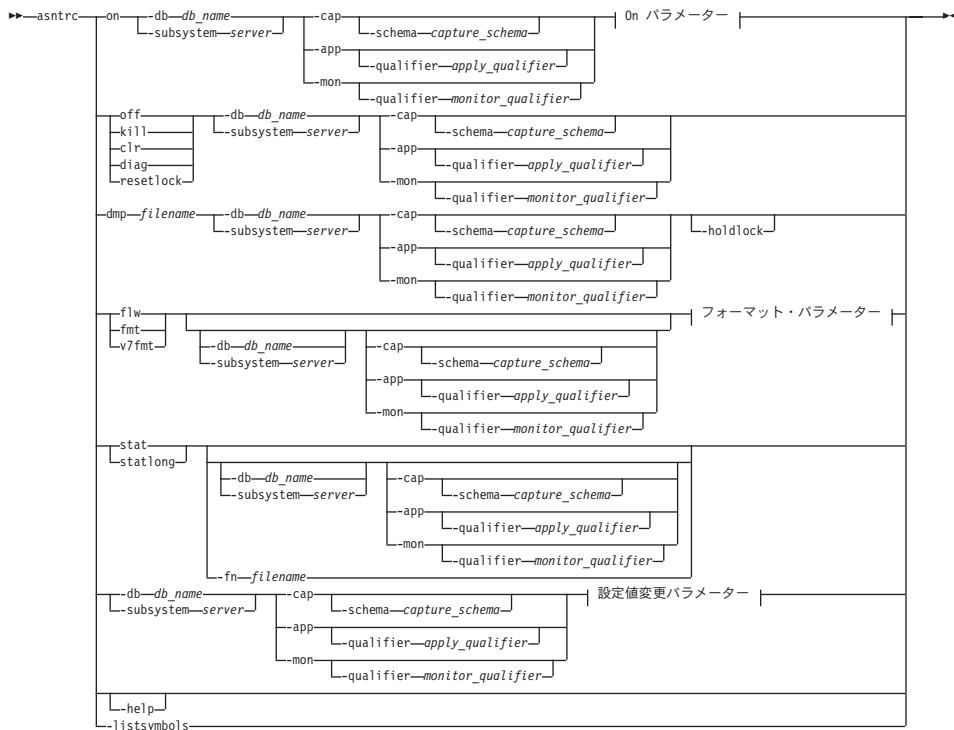
関連資料:

- レプリケーションのガイドおよびリファレンス の『STRDPRCAP: キャプチャー・プログラムの始動 (OS/400)』

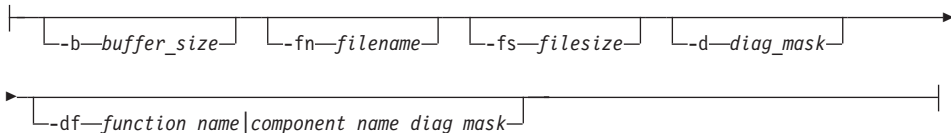
asntrc: レプリケーション・トレース機能の操作 (UNIX、Windows、z/OS)

UNIX、Windows、および z/OS 上の UNIX System Services (USS) でトレース機能を実行するには、**asntrc** コマンドを使用します。トレース機能は、キャプチャー、アプライおよびレプリケーション・アラート・モニター・プログラムから、プログラム・フロー情報をログに記録します。このトレース情報を IBM ソフトウェア・サポートに提供して、トラブルシューティングに役立てることができます。このコマンドは、オペレーティング・システムのプロンプトまたはシェル・スクリプト内で実行します。

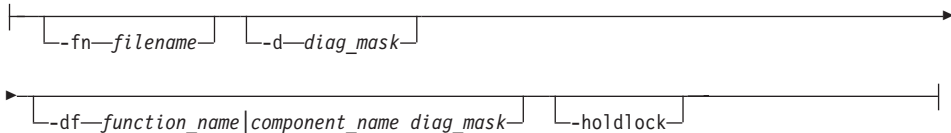
asntrc コマンドを使用してトレース機能を実行するための構文



On パラメーター:



フォーマット・パラメーター:



設定値変更パラメーター:



551 ページの表 162 は、呼び出しパラメーターを定義します。

表 162. UNIX、Windows、および z/OS オペレーティング・システム用 asnlrc 呼び出しパラメーター定義

パラメーター	定義
on	特定のキャプチャー・プログラム、アプライ・プログラム、またはレプリケーション・アラート・モニター・プログラムについて、トレース機能をオンにすることを指定します。トレース機能は、トレース処理中に使用する共有メモリー・セグメントを作成します。
-db db_name	<p>UNIX および Windows のみ: トレースするデータベースの名前を指定します。</p> <ul style="list-style-type: none"> • トレースするキャプチャー・プログラム用のキャプチャー・コントロール・サーバーの名前を指定します。 • トレースするアプライ・プログラム用のアプライ・コントロール・サーバーの名前を指定します。 • トレースするレプリケーション・アラート・モニター・プログラム用のモニター・コントロール・サーバーの名前を指定します。
-subsystem server	<p>z/OS のみ: トレースするサブシステムの名前を指定します。</p> <ul style="list-style-type: none"> • トレースするキャプチャー・プログラム用のキャプチャー・コントロール・サーバーの名前を指定します。 • トレースするアプライ・プログラム用のアプライ・コントロール・サーバーの名前を指定します。 • トレースするレプリケーション・アラート・モニター・プログラム用のモニター・コントロール・サーバーの名前を指定します。
-cap	キャプチャー・プログラムをトレースすることを指定します。キャプチャー・プログラムは -schema パラメーターで識別されます。
-schema capture_schema	トレースするキャプチャー・プログラムの名前を指定します。キャプチャー・プログラムは指定したキャプチャー・スキーマにより識別されます。このパラメーターは -cap パラメーターと一緒に使用します。
-app	アプライ・プログラムをトレースすることを指定します。アプライ・プログラムは -qualifier パラメーターで識別されます。

表 162. UNIX、Windows、および z/OS オペレーティング・システム用 *asnlrc* 呼び出しパラメーター定義 (続き)

パラメーター	定義
-qualifier <i>apply_qualifier</i>	トレースするアプライ・プログラムの名前を指定します。このアプライ・プログラムは、指定したアプライ修飾子により識別されます。このパラメーターは -app パラメーターと一緒に使用します。
-mon	レプリケーション・アラート・モニター・プログラムをトレースすることを指定します。レプリケーション・アラート・モニター・プログラムは -qualifier パラメーターで識別されます。
-qualifier <i>monitor_qualifier</i>	トレースするレプリケーション・アラート・モニター・プログラムの名前を指定します。このレプリケーション・アラート・モニター・プログラムは、指定したモニター修飾子により識別されます。このパラメーターは -mon パラメーターと一緒に使用します。
off	特定のキャプチャー・プログラム、アプライ・プログラム、またはレプリケーション・アラート・モニター・プログラムについて、トレース機能をオフにし、使用中の共有メモリー・セグメントを解放することを指定します。
kill	トレース機能を強制的に異常終了させることを指定します。 このパラメーターは、何らかの問題により、トレース機能を off パラメーターでオフにできない場合のみ使用してください。
clr	トレース・バッファーをクリアすることを指定します。このパラメーターは、トレース・バッファーの内容を消去しますが、バッファーはアクティブのままにします。
diag	トレース機能の実行中に、フィルター設定を表示することを指定します。
resetlock	トレース機能のバッファー・ラッチを解放することを指定します。このパラメーターは、エラー状態が起これ、トレース・プログラムがバッファー・ラッチを保留したまま終了した場合に、バッファー・ラッチをエラー状態からリカバーできるようにします。
dmp <i>filename</i>	トレース・バッファーの現在の内容をファイルに書き込むことを指定します。

表 162. UNIX、Windows、および z/OS オペレーティング・システム用 asnlrc 呼び出しパラメーター定義 (続き)

パラメーター	定義
-holdlock	トレース機能がバッファをコピーするためのメモリーが不足している場合でも、ロックを保留している間に、トレース機能がファイルのダンプまたはコマンドの出力を完了できることを指定します。
flw	トレース機能が作成し、共有メモリーまたはファイルに保管したサマリー情報を表示することを指定します。この情報には、プログラム・フローが含まれ、それぞれの処理およびスレッドごとに、関数と呼び出しのスタック構造がわかるように字下げして表示されます。
fmt	トレース機能が作成し、共有メモリーまたはファイルに保管した詳細情報を表示することを指定します。このパラメーターは、トレースしたデータ構造の内容全体を発生順に表示します。
v7fmt	トレース機能が作成し、共有メモリーまたはファイルに保管した情報を表示することを指定します。このトレース情報はバージョン 7 のフォーマットで表示されます。
stat	トレース機能の状況を表示することを指定します。この状況情報には、トレース・バージョン、アプリケーション・バージョン、エンタリー数、バッファ・サイズ、使用中のバッファ量、状況コード、およびプログラム・タイム・スタンプが含まれます。
statlong	トレース機能の状況に z/OS バージョン・レベル情報を追加して表示することを指定します。この追加情報には、アプリケーション内の各モジュールのサービス・レベルが含まれ、長ストリングのテキストとして表示されます。
-fn filename	ミラーリングされたトレース情報を含むファイル名を指定します。ここには、トレース機能からのすべての出力が含まれます。
-help	有効なコマンド・パラメーターを記述と一緒に表示します。
-listsymbols	-df パラメーターで使用できる有効な関数およびコンポーネント ID を表示します。

表 162. UNIX、Windows、および z/OS オペレーティング・システム用 *asntrc* 呼び出しパラメーター定義 (続き)

パラメーター	定義
-b <i>buffer_size</i>	トレース・バッファのサイズをバイト単位で指定します。数値の後に、キロバイトなら K を、メガバイトなら M を指定できます。これらの文字には大文字小文字の区別はありません。
-fs <i>filesize</i>	ミラーリングされたトレース情報ファイルのサイズ制限をバイト単位で指定します。
-d <i>diag_mask</i>	<p>トレース機能により記録されるトレース・レコードのタイプを指定します。トレース・レコードは、以下の診断マスク番号により分類されます。</p> <ol style="list-style-type: none"> 1 フロー・データ。関数の入力点と出口点が含まれます。 2 基本データ。トレース機能が検出したすべての主要なイベントが含まれます。 3 詳細データ。主要なイベントとその記述が含まれます。 4 パフォーマンス・データ。 <p>重要: 診断マスク番号の大きいものは、診断マスク番号の小さいものを包含していません。</p> <p>これらの番号を 1 つまたは複数入力し、必要なトレース・レコードだけを含む診断マスクを作成することができます。たとえば、-d 4 を指定すると、パフォーマンス・データだけが記録されます。フローとパフォーマンスのデータだけを記録するには -d 1,4 と指定し、すべてのトレース・レコードを記録するには -d 1,2,3,4 (デフォルト) と指定します。番号はコマンドで区切ります。</p> <p>トレース機能がグローバル・トレース・レコードを記録しないようにするには、診断マスク番号 0 (ゼロ) を入力します。トレース機能に新しい診断マスク番号を指定する前に、診断レベルをリセットするには、-d 0 を入力します。</p>

表 162. UNIX、Windows、および z/OS オペレーティング・システム用 asnlrc 呼び出しパラメーター定義 (続き)

パラメーター	定義
-df	特定の関数またはコンポーネント ID をトレースすることを指定します。
<i>function_name component_name</i>	
<i>diag_mask</i>	関数またはコンポーネント ID 名の後に診断マスク番号 (1、2、3、4) を入力します。1 つまたは複数の番号を入力できます。番号はコンマで区切ります。

asnlrc の例

以下の例は、**asnlrc** コマンドの使用法を示しています。

例 1: UNIX または Windows の下で実行中のキャプチャー・プログラムをトレースするには、以下のようにします。

1. 最大のバッファ・サイズとファイル・サイズを持つトレース・ファイル名を指定して、トレース機能を開始します。

```
asnlrc on -db mydb -cap -schema myschema -b 256k -fn myfile.trc -fs 500m
```

2. キャプチャー・プログラムを始動し、適切な長さの時間実行させます。
3. トレース機能がオンになっている間に、共有メモリーからデータを直接表示します。トレース機能から処理とスレッドのサマリー情報を表示するには、以下のように指定します。

```
asnlrc flw -db mydb -cap -schema myschema
```

キャプチャー・ログ・リーダーからのみ、フロー、基本、詳細、およびパフォーマンスのデータ・レコードを表示するには、以下のように指定します。

```
asnlrc fmt -db mydb -cap -schema myschema -d 0
-df "Capture Log Read" 1,2,3,4
```

4. トレース機能を停止します。

```
asnlrc off -db mydb -cap -schema myschema
```

トレース・ファイルには、キャプチャー・プログラムの始動時点から、トレース機能をオフにした時点までに生成された、すべてのキャプチャー・プログラム・トレース・データが含まれます。

5. トレース機能を停止した後、生成されたバイナリー・ファイルからのデータをフォーマットします。

```
asnlrc flw -fn myfile.trc
```

および

```
asnlrc fmt -fn myfile.trc -d 0 -df "Capture Log Read" 1,2,3,4
```

例 2: UNIX または Windows の下で実行中のレプリケーション・アラート・モニター・プログラムのトレース機能を開始するには、以下のように指定します。

```
asnlrc on -db mydb -mon -qualifier monq
```

例 3: UNIX または Windows で実行中のアプライ・プログラムのパフォーマンス・データだけをトレースするには、以下のように指定します。

```
asnlrc on -db mydb -app -qualifier aq1 -b 256k -fn myfile.trc -d 4
```

例 4: z/OS の下で実行中のキャプチャー・プログラムのすべてのフローおよびパフォーマンス・データをトレースするには、以下のように指定します。

```
asnlrc on -subsystem dbsevl -cap -schema myschema -b 256k  
-fn myfile.trc -d 1,4
```

例 5: すべてのグローバルなパフォーマンス・データおよび、UNIX または Windows の下で実行中のキャプチャー・プログラムの、特定のキャプチャー・ログ・リーダーのフロー・データをトレースするには、以下のように指定します。

```
asnlrc on -db mydb -cap -schema myschema -b 256k -fn myfile.trc -d 4  
-df "Capture Log Read" 1
```

例 6: UNIX または Windows の下で実行中のキャプチャー・プログラムをトレースし、トレース機能の特定の時点のイメージを表示して保管するには、以下のようにします。

1. 最新のレコードを保留するために十分なバッファー・サイズを指定して、トレース・コマンドを始動します。

```
asnlrc on -db mydb -cap -schema myschema -b 4m
```

2. キャプチャー・プログラムを始動し、適切な長さの時間実行させます。
3. 共有メモリーに保管された、特定の時点の詳細なトレース情報を表示します。

```
asnlrc fmt -db mydb -cap -schema myschema
```

4. 特定の時点のトレース情報をファイルに保管します。

```
asnlrc dmp myfile.trc -db mydb -cap -schema myschema
```

5. トレース機能を停止します。

```
asnlrc off -db mydb -cap -schema myschema
```

関連資料:

- レプリケーションのガイドおよびリファレンスの『WRKDPTRC: DPR トレース機能の使用法 (OS/400)』

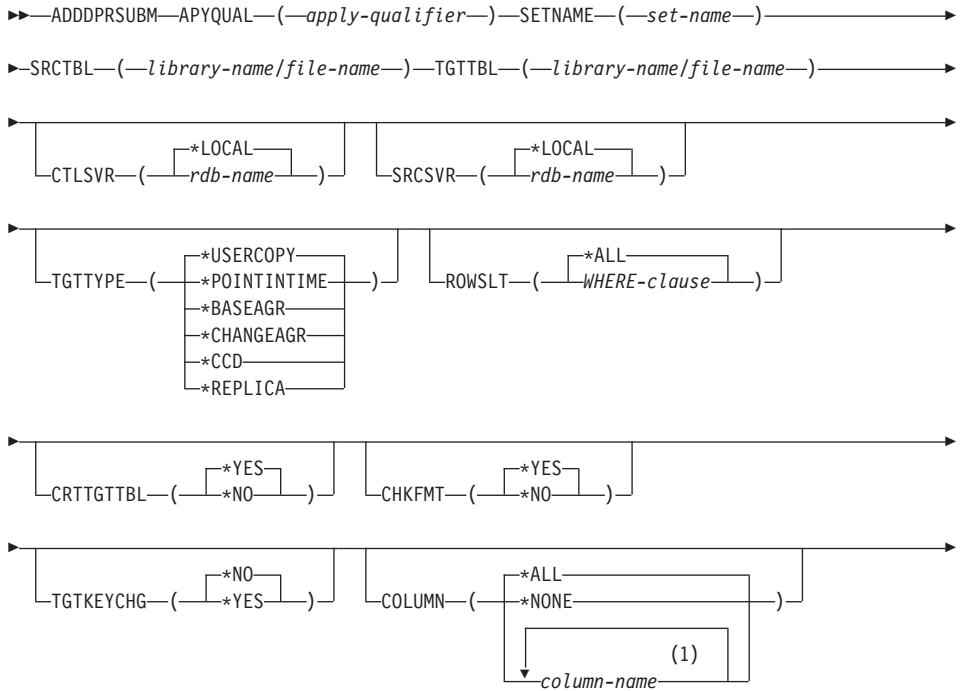
ADDDPRSUBM: DPR サブスクリプション・セットのメンバーの追加 (OS/400)

既存のサブスクリプション・セットにメンバーを追加するには、DPR サブスクリプション・セット・メンバーの追加 (ADDDPRSUBM) コマンドを使用します。サブスクリプション・セットは、ADDDPRSUB コマンドを使用して、UNIX、Windows、または z/OS 上のシステム・コマンドを使用して、またはレプリケーション・センターから作成することができます。サブスクリプション・セット内のすべてのソース表は、すでにジャーナルに記録済みであり、登録済みでなければならず、その後でないこのコマンドは使用できません。

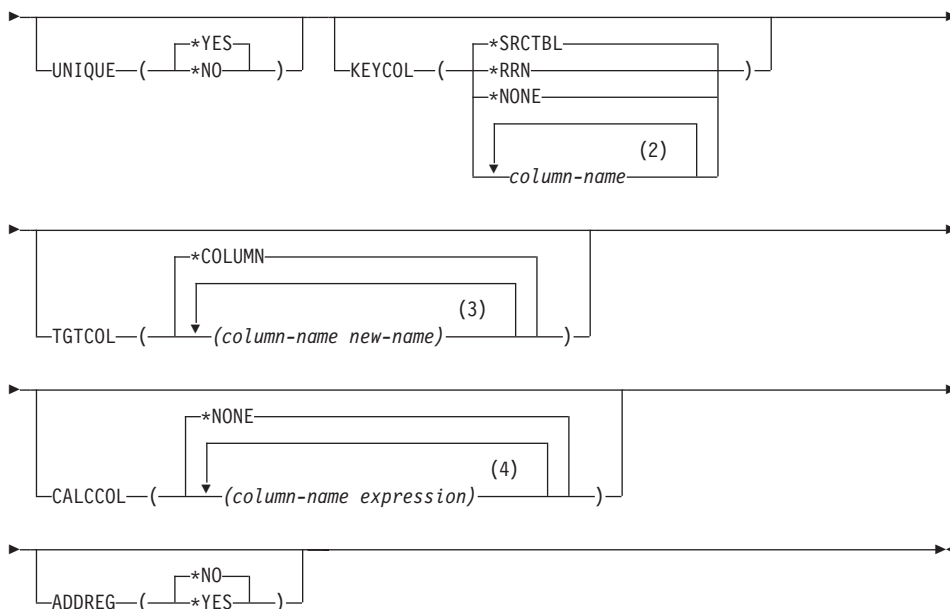
コマンド行にコマンド名を入力したら、F4 キーを押してコマンド構文を表示することができます。

このコマンド、およびこのコマンドのすべてのパラメーターの詳しい記述を表示するには、画面の一番上のコマンドにカーソルを移動し、F1 キーを押します。特定のパラメーターの記述を表示するには、そのパラメーター上にカーソルを移動し、F1 キーを押します。

ADDDPRSUBM コマンドを使用してサブスクリプション・セットにメンバーを追加するための構文



ADDDPRSUBM



注:

- 1 列名は 300 個まで指定できます。
- 2 列名は 120 個まで指定できます。
- 3 列名は 300 個まで指定できます。
- 4 100 個までの列名および式を指定できます。

表 163. ADDDPRSUBM コマンド・パラメーター定義 (OS/400 用)

パラメーター	定義およびプロンプト
APYQUAL	<p>どのアプライ・プログラムがこのサブスクリプション・セットを処理するかを示すアプライ修飾子を指定します。アプライ修飾子の下のサブスクリプション・セットは別のジョブで実行されます。このパラメーターは必須です。</p> <p><i>apply-qualifier</i> アプライ修飾子の名前。</p>

表 163. ADDDPRSUBM コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
SETNAME	<p>サブスクリプション・セットの名前を指定します。このパラメーターは必須です。</p> <p><i>set-name</i></p> <p>サブスクリプション・セットの名前。入力するサブスクリプション・セット名は、指定されたアプライ修飾子についてユニークでなければなりません。ユニークでない場合、ADDDPRSUBM コマンドはエラーになります。アプライ・プログラムはターゲット表のセットをグループとして扱うので、何らかの理由で 1 つのターゲット表が失敗すると、そのセット全体が失敗します。</p>
SRCTBL	<p>このサブスクリプション・セット・メンバーのソースである表の名前を指定します。この表をサブスクリプション・セットのメンバーにするには、この表をキャプチャー・コントロール・サーバーにあらかじめ登録しておく必要があります。このパラメーターは必須です。</p> <p><i>library-name/file-name</i></p> <p>ソース表の修飾名。</p>
TGTTBL	<p>このサブスクリプション・セット・メンバーのターゲット表の名前を指定します。CRRTGTTBL パラメーターを *YES に設定し、ターゲット表が存在しないと、ターゲット表は自動的に作成されます。このパラメーターは必須です。</p> <p><i>library-name/file-name</i></p> <p>ターゲット表の修飾名。</p>
CTLSVR	<p>アプライ・コントロール表を含むシステムのリレーショナル・データベース名を指定します。</p> <p>*LOCAL (デフォルト)</p> <p>アプライ・コントロール表はローカルに存在します (ADDDPRSUBM コマンドを実行するマシン上)。</p> <p><i>rdb-name</i></p> <p>アプライ・コントロール表が置かれているリレーショナル・データベースの名前。RDB ディレクトリー・エントリーの作業 (WRKRDBDIRE) コマンドを使用して、この名前を検出することができます。</p>

表 163. ADDDPRSUBM コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
SRCSVR	<p>キャプチャー・コントロール表を含むシステムのリレーショナル・データベース名を指定します。</p> <p>*LOCAL (デフォルト)</p> <p>ソース表はローカル・マシンに登録されています (ADDDPRSUBM コマンドを実行するマシン)。</p> <p><i>rdB-name</i></p> <p>キャプチャー・コントロール表が存在するリレーショナル・データベースの名前。 RDB ディレクトリー・エントリーの作業 (WRKRDBDIRE) コマンドを使用して、この名前を検出することができます。</p>
TGTTYPE	<p>ターゲット表のタイプを指定します。これらは、ターゲット表の内容を記述する DB2 レプリケーション用語です。これらのいずれかのタイプとしてターゲット表を作成した後、DPR 登録の追加 (ADDDPRREG) コマンドの SRCTBL パラメーターにこのパラメーター値を使用し、このターゲット表をソース表として登録することができます。</p> <p>*USERCOPY (デフォルト)</p> <p>ターゲット表はユーザー・コピーであり、これはソース表の内容のすべてまたは一部と一致する内容を持つ、ターゲット表です。ユーザー・コピーはポイント・イン・タイムの表のように扱われますが、ポイント・イン・タイムのターゲット表に存在する、DB2 DataPropagator for iSeries システム列は 1 つも含まれていません。</p> <p>この値は、KEYCOL パラメーターに *RRN の値が指定されている場合は無効です。</p> <p>SRCTBL パラメーターで指定した表は、ユーザー・データベース、ポイント・イン・タイム表、または整合変更データ (CCD) のいずれかでなければなりません。</p> <p>重要: ターゲット表がすでに存在する場合、DB2 DataPropagator for iSeries は、これに対する変更を自動的にジャーナルに記録しません。ジャーナリングは、DB2 DataPropagator for iSeries の外側で開始する必要があります。</p>

表 163. ADDDPRSUBM コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター 定義およびプロンプト

TGTTYPE (続き)***POINTINTIME**

ターゲット表はポイント・イン・タイム表です。ポイント・イン・タイム表は、ソース表の内容の一部またはすべてと一致する内容を持つターゲット表であり、DB2 DataPropagator for iSeries システム列 (IBMSNAP_LOGMARKER) を保持します。この列は、特定の行がキャプチャー・コントロール・サーバーでいつ挿入または変更されたかを示します。

***BASEAGR**

ターゲット表は基本集約表であり、これはソース表から集約された (算出された) データを含む、ターゲット表です。基本集約ターゲットのソース表は、ユーザー表またはポイント・イン・タイム表のどちらかでなければなりません。このターゲット表には、システム・タイム・スタンプ列の IBMSNAP_HLOGMARKER および IBMSNAP_LLOGMARKER が含まれます。

***CHANGEAGR**

表は変更集約表であり、これは、変更データ (CD) 表の内容を基に集約された (算出された) データを含むターゲット表です。このターゲット表は、システム・タイム・スタンプ列 IBMSNAP_HLOGMARKER および IBMSNAP_LLOGMARKER を使用して作成されます。

***CCD**

表は整合変更データ (CCD) 表であり、これは、変更データ (CD) 表と作業単位 (UOW) 表内のデータを結合したものから作成されたターゲット表です。CCD 表は、アプライ・プログラムのためのトランザクション整合データを提供し、以下の列が入っている必要があります。

- IBMSNAP_INTENTSEQ
- IBMSNAP_OPERATION
- IBMSNAP_COMMITSEQ
- IBMSNAP_LOGMARKER

***REPLICA**

ターゲット表はレプリカ表であり、これは update-anywhere レプリケーションにのみ使用されます。レプリカ・ターゲット表はマスター・ソース表から変更を受信し、またレプリカ・ターゲット表への変更は、マスター・ソース表に戻して伝搬されます。レプリカ表はソース表として自動的に登録されます。

表 163. ADDDPRSUBM コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
ROWSLT	<p>SQL の WHERE 文節に入れる述部を指定します。アプライ・プログラムはこれらの述部を使用して、ソースの変更データ (CD) 表のどの行をターゲット表に適用するかを決めます。ソースの変更のサブセットだけをターゲット表に複製したい場合に、このパラメーターを使用します。</p>
	<p>*ALL (デフォルト) アプライ・プログラムは、CD 表内のすべての変更をターゲット表に適用します。</p>
	<p><i>WHERE-clause</i> アプライ・プログラムが CD 表からどの行をターゲット表に適用するかを指定する SQL の WHERE 文節です。WHERE キーワードはこのパラメーターで暗黙に想定されているので、WHERE キーワードを含めないでください。この WHERE 文節は、この文節を実行するデータ・サーバー上で有効なものでなければなりません。</p>
	<p>注: このパラメーターの WHERE 文節は、SQLBEFORE または SQLAFTER パラメーターに指定された WHERE 文節とはまったく関係ありません。</p>

表 163. ADDDPRSUBM コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
CRTTGTTBL	<p>ターゲット表 (またはビュー) を作成するかどうかを指定します。</p> <p>*YES (デフォルト) ターゲット表 (またはビュー) が存在しないければ、作成します。存在する場合は、既存の表またはビューがターゲットになり、この既存の表またはビューのフォーマットの CHKFMT パラメーターが *YES に設定されているかどうかをチェックします。UNIQUE および KEYCOL パラメーターに指定された値を使用して、(そのような索引がまだ存在しなければ) ターゲット表に追加の索引が作成されます。既存のターゲット表に、追加索引の条件に違反するような行が含まれていると、コマンドは失敗します。</p> <p>*NO ターゲット表またはビューを作成しません。アプライ・プログラムを始動する前に、正しい属性を使用して表またはビューを作成する必要があります。</p> <p>表またはビューが存在する場合に CHKFMT を *YES に設定すると、ADDDPRSUBM コマンドは、既存の表のフォーマットが、設定されたサブスクリプション・セット定義と一致することを確認します。CHKFMT を *NO にする場合は、ユーザーは既存の表のフォーマットがサブスクリプション・セット定義と一致することを確認しておく必要があります。</p> <p>重要: 表またはビューがすでに存在する場合、DB2 DataPropagator for iSeries は、既存のオブジェクトへの変更を自動的にジャーナルに記録しません。ジャーナリングは、DB2 DataPropagator for iSeries の外側で開始する必要があります。</p>

表 163. ADDDPRSUBM コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
CHKFMT	<p>DB2 DataPropagator for iSeries が、サブスクリプション・セット・メンバーの定義を既存のターゲット表に照らし合わせ、列が一致することをチェックするかどうかを指定します。このパラメーターは、CRITGTBL パラメーターが *YES の場合は無視され、また CRITGTBL パラメーターが *NO でターゲット表が存在しない場合も無視されます。</p> <p>*YES (デフォルト) DB2 DataPropagator for iSeries は、このサブスクリプション・セット・メンバーに定義された列が、ターゲット表内の列と一致するか検証します。両者が一致しないと、このコマンドは失敗します。</p> <p>*NO DB2 DataPropagator for iSeries は、サブスクリプション・セット・メンバーと既存のターゲット表間の相違を無視します。ユーザーはターゲット表がサブスクリプション・セット・メンバーと互換性があることを確認する必要があります。</p>
TGTKEYCHG	<p>ターゲット表のターゲット・キー列の一部であるソース列に変更があった場合、アプライ・プログラムがその更新をどのように扱うかを指定します。このパラメーターは、ADDDPRREG コマンドの USEDELINS パラメーターと組み合わせて働きます。</p> <ul style="list-style-type: none"> • USEDELINS が YES で TGTKEYCHG が YES の場合、更新はできません。 • USEDELINS が YES で TGTKEYCHG が NO の場合、更新は削除と挿入の対になります。 • USEDELINS が NO で TGTKEYCHG が YES の場合、アプライ・プログラムは特別な論理を使用してこの条件を扱います。 • USEDELINS が NO で TGTKEYCHG が NO の場合、アプライ・プログラムは変更を通常の更新として処理します。 <p>*NO (デフォルト) ソース表に対する更新は、キャプチャー・プログラムによりステージ化され、アプライ・プログラムによりターゲット表に処理が行われます。</p> <p>*YES アプライ・プログラムは、ターゲット・キー列の変更前イメージに基づいてターゲット表を更新します。つまり、アプライ・プログラムは述部を新しい値ではなく、古い値に変更します。</p>

表 163. ADDDPRSUBM コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
COLUMN	<p>ターゲット表に含める列を指定します。列名は修飾できません。列名は、ソース表の登録時に CAPCOL パラメーターに指定した列名のリストから選択してください。</p> <p>この表の登録時に IMAGE パラメーターを *BOTH に設定した場合は、変更前イメージ列名を指定することができます。変更前イメージ列名は、接頭部を持つオリジナルの列名です。この接頭部は、ADDDPRREG コマンドの PREFIX パラメーターに指定した文字です。</p> <p>*ALL (デフォルト) ソースに登録した列のすべてがターゲット表に含まれます。</p> <p>*NONE ソース表からの列は 1 つもターゲット表に含まれません。 *NONE は、算出列だけをターゲット表に含めたい場合に使用します。この値は、CALCCOL パラメーターに合計関数が含まれているが、グループ化が実行されない、という場合に必要です。</p> <p><i>column-name</i> ターゲット表に含めたいソース列の名前を 300 個まで指定できます。列名はスペースで区切ります。</p>

表 163. ADDDPRSUBM コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
UNIQUE	<p>ターゲット表が KEYCOL パラメーターで示されたユニーク・キーを持つかどうかを指定します。</p>
	<p>*YES (デフォルト)</p> <p>ターゲット表はキーごとに正味 1 つの変更をサポートします。つまり、キーに対していかに多くの変更がなされたとしても、そのキーについてはターゲット表に 1 つの行しか存在しないということです。</p>
	<p>この値は、表がデータの変更の履歴ではなく、現行のデータを含むことを指定します。コンデンス表には、1 つの主キー値に対して複数の行が含まれることはなく、リフレッシュ用の最新情報を提供するために使用できます。</p>
	<p>*NO</p> <p>ターゲット表はキーごとに複数の変更をサポートします。変更はターゲット表に付加されます。</p>
	<p>この値は、表が現行のデータではなく、変更の履歴を含むことを指定します。非コンデンス表には、それぞれのキー値に対して複数の行が含まれ、データの変更履歴を提供するために使用できます。ただし、非コンデンス表はリフレッシュ用の最新データは提供できません。</p>

表 163. ADDDPRSUBM コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
KEYCOL	<p>ターゲット表のキーを記述する列を指定します。列名は修飾できません。*POINTINTIME、*REPLICA、および *USERCOPY ターゲット表 (TGTTYPE パラメーターで指定されている) の場合、ターゲット表に 1 つまたは複数の列をターゲット・キーとして指定する必要があります。アプライ・プログラムはこのターゲット・キーを使用して、変更キャプチャー・レプリケーション中に、変更された個々のユニークな行を識別します。</p> <p>*SRCTBL (デフォルト)</p> <p>ターゲット表のキー列は、ソース表のキー列と同じです。ADDDPRREG コマンドは、ソース表にキーがある場合、ソース表に指定されたキーを使用します。以下のキー列が使用されます。</p> <ul style="list-style-type: none"> • 物理ファイル作成コマンド (CRTPF) を使用して表を作成した時に、DDS を使用して定義したキー列 • CREATE TABLE および ALTER TABLE SQL ステートメントを使用して定義した、主キーおよびユニーク・キー • CREATE INDEX SQL ステートメントを使用して定義したユニーク・キー <p>1 つの列を、キーとして、異なる順序付けで複数回使用すると、ターゲット表のキーは昇順で定義されます。</p> <p>*RRN</p> <p>ターゲット表のキー列は IBMQSQ_RRN 列です。ターゲット表は IBMQSQ_RRN 列を使用して作成され、この列がキーとして使用されます。アプライ・プログラムの実行時に、ソース表がユーザー表であり、ターゲット表がポイント・イン・タイムまたはユーザー・コピーの場合、ターゲット表の IBMQSQ_RRN 列が、ソース表内の関連するレコードの RRN (相対レコード番号) で更新されます。それ以外では、ターゲット表の IBMQSQ_RRN 列は、ソース表内の IBMQSQ_RRN 列の値で更新されます。</p> <p>*NONE</p> <p>ターゲット・コピーはターゲット・キーを含みません。ターゲット表のタイプが *POINTINTIME、*REPLICA、または *USERCOPY の場合、*NONE は指定できません。</p> <p><i>column-name</i></p> <p>ターゲット・キー列として使用したいターゲット列の名前。列名は 120 個まで指定できます。列名はスペースで区切ります。</p>

表 163. ADDDPRSUBM コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
TGTCOL	<p>アプライ・プログラムがターゲット表内で更新するすべての列の新しい名前を指定します。これらの名前は、ソース表から取られた列名をオーバーライドします。列名は修飾できません。COLUMN パラメーターに *NONE を指定した場合は、TGTCOL パラメーターを使用しないでください。</p> <p>このパラメーターを使用して、ターゲット表の列により分かりやすい名前を付けることができます。それぞれのソース列の名前と、ターゲット表の対応する列の名前を指定します。</p> <p>*COLUMN (デフォルト) ターゲット列は、COLUMN パラメーターに指定した列と同じです。</p> <p><i>column-name</i> ターゲットではその名前を変更したい、ソース表からの列名。列名は 300 個まで指定できます。</p> <p><i>new-name</i> ターゲット列の新しい名前。新しい列名を 300 個まで指定できます。このパラメーターを使用しないと、ターゲット表の列名はソースの列名と同じになります。</p>
CALCCOL	<p>ターゲット表のユーザー定義の列または算出された列のリストを指定します。列名は修飾できません。それぞれの列名と式の対を括弧で囲みます。</p> <p>各 SQL 式には列名を指定する必要があります。GROUP BY 文節のない SQL 式として列を定義する場合は、COLUMN パラメーターを *NONE にする必要があります。</p> <p>*NONE (デフォルト) ターゲット表はユーザー定義の列または算出された列を含みません。</p> <p><i>column-name</i> ターゲット表のユーザー定義の列または算出された列の列名。列名は 100 個まで指定できます。</p> <p><i>expression</i> ターゲット表のユーザー定義の列または算出された列の式。SQL 列の式は 100 個まで指定できます。</p>

表 163. ADDDPRSUBM コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
ADDREG	<p>ターゲット表をソース表として自動的に登録するかどうかを指定します。このパラメーターは CCD ターゲット・タイプの表を登録する場合に使用します。</p> <p>*NO (デフォルト) ターゲット表はソース表として登録されません。 DB2 DataPropagator for iSeries は、ターゲット・タイプが *REPLICA の場合、このパラメーター値を無視します。レプリカ・ターゲット表は必ず、ソース表として自動的に登録されます。</p> <p>*YES ターゲット表はソース表として登録されます。ターゲット表をすでにユーザーが登録していると、このコマンドは失敗します。</p> <p>ターゲット表のタイプが *USERCOPY、*POINTINTIME、*BASEAGR、または *CHANGEAGR の場合、このパラメーターを *YES にしないでください。</p> <p>CRTTGTTBL パラメーターを *NO にした場合、これをソースとして登録する前に、ターゲット表を作成する必要があります。</p>

ADDDPRSUBM の例

以下の例は、ADDDPRSUBM コマンドの使用法を示しています。

例 1: サブスクリプション・セット・メンバーを AQHR アプライ修飾子の下の SETHR という名前のサブスクリプション・セットに追加します。

```
ADDDPRSUBM APYQUAL(AQHR) SETNAME(SETHR) SRCTBL(HR/YDXTAX) TGTTBL(TGTHR/TGTTAX)
```

例 2: 2 つだけの列 (AMOUNT および NAME) を持つサブスクリプション・セット・メンバーを、YDXTAX という名前の登録済みソース表から追加し、これらの列を TGTTAX という名前の既存のターゲット表に複製します。

```
ADDDPRSUBM APYQUAL(AQHR) SETNAME(SETHR) SRCTBL(HR/YDXTAX) TGTTBL(TGTLIB/TGTTAX)
  CRTTGTTBL(*NO) COLUMN(AMOUNT NAME) CHKFMT(*YES)
```

このコマンドは、このサブスクリプション・セット・メンバーに定義された AMOUNT 列と NAME 列が、ターゲット表内の列と一致するか検証します。

例 3: サブスクリプション・セット・メンバーを SETHR という名前のサブスクリプション・セットに追加し、このデータを TGTYPD という名前の整合変更データ・ターゲット表に複製します。

```
ADDDPRSUBM APYQUAL(AQHR) SETNAME(SETHR) SRCTBL(HR/YDXTAX) TGTTBL(TGTLIB/TGTYPD)
  TGTTYPE(*CCD) ADDREG (*YES)
```

ADDDPRSUBM

このコマンドは、ターゲット表を DB2 DataPropagator for iSeries のソース表として登録します。

関連タスク:

- 687 ページの『ソースのサブスクライブ』

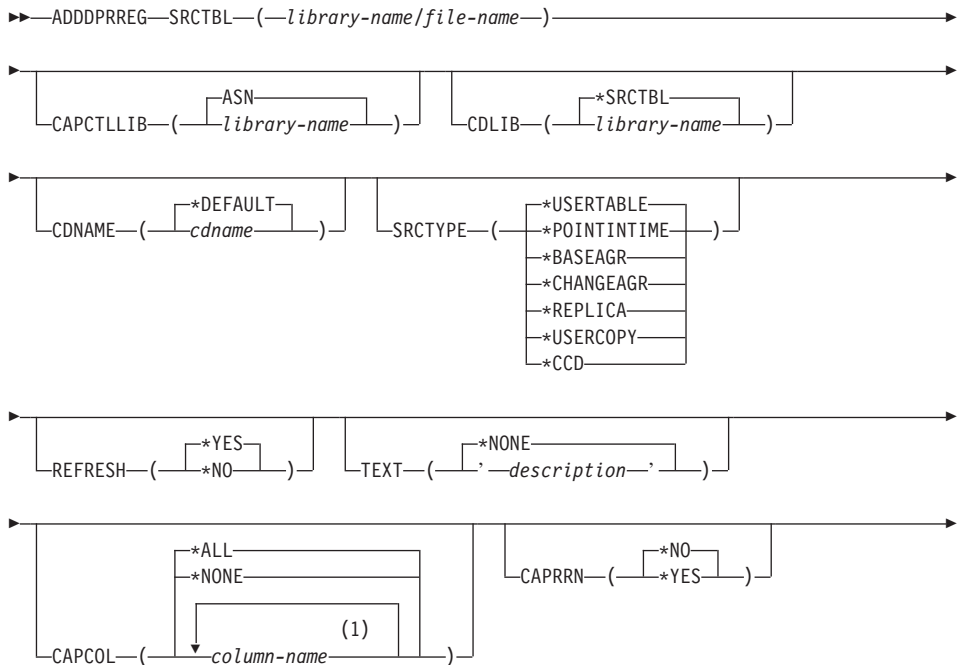
ADDDPRREG: DPR 登録の追加 (OS/400)

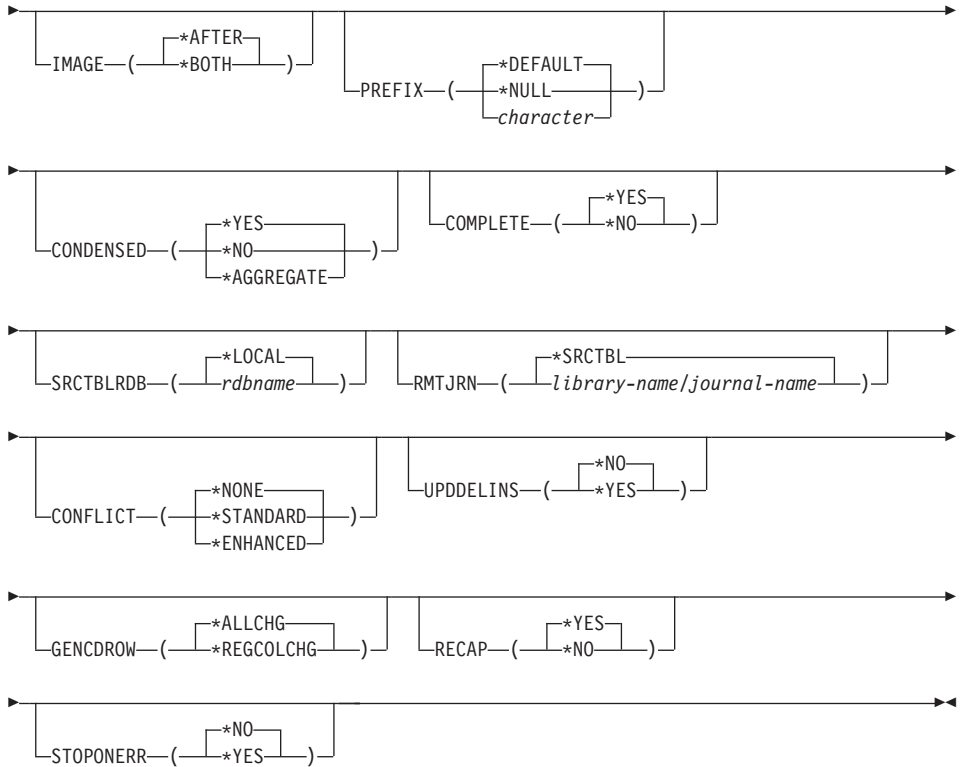
表を DB2 DataPropagator for iSeries のソース表として登録するには、DPR 登録の追加 (ADDDPRREG) コマンドを使用します。

コマンド行にコマンド名を入力したら、F4 キーを押してコマンド構文を表示することができます。

このコマンド、およびこのコマンドのすべてのパラメーターの詳しい記述を表示するには、画面の一番上のコマンドにカーソルを移動し、F1 キーを押します。特定のパラメーターの記述を表示するには、そのパラメーター上にカーソルを移動し、F1 キーを押します。

ADDDPRREG コマンドを使用して表を登録するための構文





注:

- 1 列名は 300 個まで指定できます。

表 164. ADDDPRREG コマンド・パラメーター定義 (OS/400 用)

パラメーター	定義およびプロンプト
SRCTBL	<p>ソース表として登録する表を指定します。キャプチャー・プログラムは、OS/400 ライブラリー内の任意の物理ファイルまたは、外部で定義され、1 つのフォーマットを持つ集合をサポートします。このパラメーターは必須です。</p> <p><i>library-name/file-name</i> 登録する表の修飾名を表します。</p>

表 164. ADDDPRREG コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
CAPCTLLIB	<p>キャプチャー・スキーマ (キャプチャー・コントロール表が存在するライブラリーの名前) を指定します。</p> <p>ASN (デフォルト) キャプチャー・コントロール表は ASN ライブラリー内にあります。</p> <p><i>library-name</i> キャプチャー・コントロール表を含むライブラリーの名前。 CRTDPRBL コマンドに CAPCTLLIB パラメーターを指定して、このライブラリーを作成することができます。</p>
CDLIB	<p>この登録されたソースの変更データ (CD) 表を作成するライブラリーを指定します。</p> <p>*SRCTBL (デフォルト) ソース表が存在するライブラリー内に CD 表を作成します。</p> <p><i>library-name</i> 指定したライブラリー名に CD 表を作成します。</p>
CDNAME	<p>変更データ (CD) 表の名前を指定します。</p> <p>*DEFAULT (デフォルト) デフォルト名 (現行タイム・スタンプを基にしたもの) を使用して CD 表を作成します。たとえば、現行のタイム・スタンプが 2002 年 1 月 23 日 9 時 58 分 26 秒の場合、デフォルト名は ASN020123095826CD となります。</p> <p><i>cdname</i> ここに指定した名前 で CD 表を作成します。</p>
SRCTYPE	<p>登録するソース表のタイプを指定します。ご使用のレプリケーション構成に基づいてソース・タイプを選択します。</p> <ul style="list-style-type: none"> 基本データ配分またはデータ統合構成の場合は、デフォルトの USERTABLE を使用します。 update-anywhere 構成の場合は REPLICA を使用します。 複数の階層からなる構成を持ち、ターゲット表を、レプリケーション構成内の下位の層のためのソースにしたい場合は、POINTINTIME、BASEAGR、CHANGEAGR、USERCOPY、または CCD を使用します。 <p>既存のターゲット表をソースとして登録する場合、指定されたソース・タイプに示された IBMSNAP 表の列がターゲット表に含まれていないと、登録は失敗します。</p>

表 164. ADDDPRREG コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
SRCTYPE (続き)	<p>*USERTABLE (デフォルト) ユーザー・データベース表。登録された表の最もよくあるタイプです。表には、DB2 DataPropagator for iSeries の列 ID である、IBMSNAP または IBMQSQ で始まる列を含めることはできません。</p> <p>*POINTINTIME ポイント・イン・タイム・コピー表。これは、ソース表の内容の一部またはすべてと一致する内容を含み、さらに、DB2 DataPropagator for iSeries システム列を保持します。このシステム列は、特定の行がソース・システムで最後に挿入または更新されたポイント・イン・タイムを示します。表には IBMSNAP_LOGMARKER タイム・スタンプ列を含める必要があり、オプションとして IBMQSQ_RRN と呼ばれる INTEGER 列を含めることができます。</p> <p>*BASEAGR 基本集約コピー。ユーザー表またはポイント・イン・タイム表からインターバルごとに集約されたデータが入ります。基礎集約表には、IBMSNAP_HLOGMARKER および IBMSNAP_LLOGMARKER タイム・スタンプ列を含める必要があります。</p> <p>*CHANGEAGR 変更を集約したコピー表。ソース表に記録された変更に基づくデータの集約が入ります。この表には、IBMSNAP_HLOGMARKER および IBMSNAP_LLOGMARKER タイム・スタンプ列を含める必要があります。</p> <p>*REPLICA レプリカ・サブスクリプションのターゲット表。ターゲット表からの変更が複製されて、元のソース表に戻せるようにするには、このタイプの表を登録します。この表には、DB2 DataPropagator for iSeries のシステム列または、DataPropagator for iSeries の列 ID である、IBMSNAP または IBMQSQ で始まる列を含めることはできません。この表は、元のソース表からのすべての列を保持します。</p> <p>*USERCOPY ソース表の内容のすべてまたは一部と一致する内容を持つ、ターゲット表です。ユーザー・コピー表はユーザー・データ列のみ保持します。</p>

表 164. ADDDPRREG コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
SRCTYPE (続き)	<p>*CCD</p> <p>整合変更データ (CCD) 表。ソース表からのトランザクション整合性のあるデータが入ります。表には、以下のように定義された列を含める必要があります。</p> <ul style="list-style-type: none"> • IBMSNAP_INTENTSEQ CHAR(10) FOR BIT DATA NOT NULL • IBMSNAP_OPERATION CHAR(1) NOT NULL • IBMSNAP_COMMITSEQ CHAR(10) FOR BIT DATA NOT NULL • IBMSNAP_LOGMARKER TIMESTAMP NOT NULL
REFRESH	<p>フル・リフレッシュ機能を使用可能にするかどうかを指定します。この値を使用して、ソース・データベースからのフル・リフレッシュを実行するアプライ・プログラムの機能をオフにすることができます。</p> <p>*YES (デフォルト)</p> <p>フル・リフレッシュを使用可能にします。</p> <p>*NO</p> <p>フル・リフレッシュは使用不可です。</p> <p>ターゲット表が基礎集約表または変更集約表の場合、このパラメーターは *NO にする必要があります。</p>
TEXT	<p>この登録に関する記述テキストを指定します。</p> <p>*NONE (デフォルト)</p> <p>エントリーには記述がありません。</p> <p>説明</p> <p>この登録を記述するテキスト。最大 50 文字を入力でき、テキストは単一引用符で囲む必要があります。</p>

表 164. ADDDPRREG コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
CAPCOL	<p>この登録済み表について、どの列の変更をキャプチャーするかを指定します。</p> <p>*ALL (デフォルト) すべての列の変更をキャプチャーします。</p> <p>*NONE この表の変更をキャプチャーしません。この値は、この表をフル・リフレッシュ用としてのみ登録する場合に指定します。登録されたこの表については変更データ (CD) 表は作成されず、キャプチャー・プログラムは表の変更をキャプチャーしません。</p> <p><i>column-name</i> その列の変更をキャプチャーする列名。列名は 300 個まで指定できます。列名はスペースで区切ります。</p>
CAPRRN	<p>変更されたレコードの相対レコード番号 (RRN) をキャプチャーするかどうかを指定します。</p> <p>*NO (デフォルト) 相対レコード番号をキャプチャーしません。</p> <p>*YES 相対レコード番号をキャプチャーします。変更データ (CD) 表に IBMQSQ_RRN と呼ばれる追加の列が作成されます。 ソース表にユニーク・キーがない場合のみ、このパラメーターを *YES にします。</p>
IMAGE	<p>変更データ (CD) 表に、ソース表の変更前イメージと変更後イメージの両方を含めるかどうかを指定します。この値は、キャプチャー列パラメーター (CAPCOL) に指定したすべての列にグローバルに適用されます。</p> <p>この IMAGE パラメーターは、CAPCOL パラメーターが *NONE の場合は無効です。</p> <p>このパラメーターに *AFTER を指定した場合でも、ソース表は *BOTH イメージでジャーナルに記録する必要があります。</p> <p>*AFTER (デフォルト) キャプチャー・プログラムはソース表の変更後イメージのみ CD 表に記録します。</p> <p>*BOTH キャプチャー・プログラムはソース表の変更前イメージと変更後イメージの両方を CD 表に記録します。</p>

表 164. ADDDPREG コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
PREFIX	<p>変更データ (CD) 表の変更前イメージの列を識別する接頭部文字を指定します。ソース表の登録された列名に、この接頭部文字で始まる列名がないことを確認してください。</p> <p>*DEFAULT (デフォルト) デフォルトの接頭部 (@) が使用されます。</p> <p>*NULL 変更前イメージは取り込まれません。この値は、IMAGE パラメーターを *BOTH にした場合は無効です。</p> <p><i>character</i> オブジェクト名に使用できる任意の 1 つの英字。</p>
CONDENSED	<p>ソース表が圧縮されているかどうかを指定します。コンデンス表には、現行データが含まれ、1 つの主キーの値に対して複数の行が含まれることはありません。</p> <p>*YES (デフォルト) ソース表は圧縮されています。</p> <p>*NO ソース表は圧縮されていません。</p> <p>*AGGREGATE ソース表のタイプは *BASEAGR (基本集約) または *CHANGEAGR (変更集約) のいずれかです。この値を使用する場合、COMPLETE パラメーターを *NO にする必要があります。</p>
COMPLETE	<p>ソース表が完全であるかどうかを指定します。これは、表に、対象となる主キー値ごとに 1 行が含まれていることを意味します。</p> <p>*YES (デフォルト) ソース表は完全です。</p> <p>*NO ソース表は完全ではありません。</p>

表 164. ADDDPRREG コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
SRCTBLRDB	<p>リモート・ジャーナリングを使用するかどうかを指定します。リモート・ジャーナリングでは、ソース表とリモート・ジャーナルは別のシステムにあります。このパラメーターを使用して、ソース表のロケーションを指定します。</p> <p>*LOCAL (デフォルト) ソース表はローカルにあります (ADDDPRREG コマンドを実行するマシン上に)。</p> <p><i>rdbname</i> ソース表が存在するリレーショナル・データベースの名前。 RDB ディレクトリー・エントリーの作業 (WRKRDBDIRE) コマンドを使用して、このリレーショナル・データベース名を見つけることができます。</p>
RMTJRN	<p>このジャーナルの名前とソース・システム上のジャーナルの名前が異なる場合に、リモート・ジャーナルの名前を指定します。このコマンドは、リモート・ジャーナルが存在するシステムから実行する必要があります。</p> <p>*SRCTBL (デフォルト) リモート・ジャーナル名は、ソース表のジャーナル名と同じです。</p> <p><i>library-name/journal-name</i> このシステムにあり、リモート・ソース表のジャーナリングに使用される、修飾されたライブラリー名とジャーナル名。</p> <p>リモート・ジャーナル名は、SRCTBLRDB パラメーターでリモート・ソース表のロケーションを指定した場合のみ指定できます。</p>
CONFLICT	<p>レプリカ・サブスクリプション内の競合を検出するときにアプライ・プログラムが使用する、競合レベルを指定します。</p> <p>*NONE (デフォルト) 競合検出を行いません。</p> <p>*STANDARD 適度な競合検出。アプライ・プログラムは、レプリカ変更データ (CD) 表内のすでに取り込まれた行について、競合を探します。</p> <p>*ENHANCED 拡張競合検出を行います。このオプションは、すべてのレプリカとソース表の間に最良のデータ保全性を提供します。</p>

表 164. ADDDPRREG コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
UPDDELINS	<p>キャプチャー・プログラムが、更新されたソース・データを、変更データ (CD) 表にどのように保管するかを決めます。</p> <p>*NO (デフォルト) キャプチャー・プログラムは、それぞれのソース変更を CD 表内の 1 つの行に保管します。</p> <p>*YES キャプチャー・プログラムは、それぞれのソース変更を CD 表内に 2 行を使用して保管し、その 1 行は削除用、もう 1 行は挿入用です。アプライ・プログラムは最初に削除用の行を処理し、2 番目に挿入用の行を処理します。</p>
GENCDROW	<p>キャプチャー・プログラムがソース表のすべての行から変更をキャプチャーするかどうかを指定します。</p> <p>*ALLCHG (デフォルト) キャプチャー・プログラムはソース表のすべての行から変更をキャプチャー (登録されていない列の変更も含む)、これらの変更を変更データ (CD) 表に追加します。</p> <p>*REGCOLCHG キャプチャー・プログラムは、登録された列に変更があった場合のみ変更をキャプチャーします。その後、キャプチャー・プログラムはこれらの行を CD 表に追加します。</p> <p>*REGCOLCHG は、CAPCOL パラメーターが *ALL または *NONE の場合には指定できません。</p>
RECAP	<p>アプライ・プログラムが行った変更を、キャプチャー・プログラムが再度キャプチャーするかどうかを指定します。</p> <p>*YES (デフォルト) アプライ・プログラムがソース表を変更した場合に、その変更をキャプチャーし、変更データ (CD) 表に入れます。</p> <p>*NO アプライ・プログラムがソース表を変更した場合に、その変更は取り込まず、したがって、変更データ (CD) 表には入れません。このオプションは、REPLICA タイプの表を登録する場合に使用してください。</p>

表 164. ADDDPRREG コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
STOPONERR	<p>キャプチャー・プログラムがエラーを検出した場合に、キャプチャー・プログラムを停止するかどうかを指定します。</p> <p>*NO (デフォルト) キャプチャー・プログラムはエラーを検出しても停止しません。 キャプチャー・プログラムはメッセージを出し、エラーの原因となった登録を非活動化してから、処理を継続します。</p> <p>*YES キャプチャー・プログラムはエラーを検出した場合に、メッセージを出してから停止します。</p>

ADDDPRREG の例

以下の例は、**ADDDPRREG** コマンドの使用法を示しています。

例 1: デフォルトのキャプチャー・スキーマの下に、HR ライブラリーから EMPLOYEE という名前のソース表を登録します。

```
ADDDPRREG SRCTBL(HR/EMPLOYEE)
```

例 2: BSN キャプチャー・スキーマの下に、HR ライブラリーから EMPLOYEE という名前のソース表を登録し、HRCDLIB ライブラリーの下に CDEMPLOYEE という名前の CD 表を作成します。

```
ADDDPRREG SRCTBL(HR/EMPLOYEE) CAPCTLLIB(BSN) CDLIB(HRCDLIB) CDNAME(CDEMPLOYEE)
```

例 3: BSN キャプチャー・スキーマの下に、DEPT ライブラリーから SALES という名前のポイント・イン・タイムのソース・タイプ付きソース表を登録します。

```
ADDDPRREG SRCTBL(DEPT/SALES) CAPCTLLIB(BSN) SRCTYPE(*POINTINTIME)
```

例 4: DEPT ライブラリーから SALES という名前のソース表を登録し、CD 表には、ソース表の変更について変更前イメージと変更後イメージの両方を含めます。

```
ADDDPRREG SRCTBL(DEPT/SALES) IMAGE(*BOTH)
```

例 5: DEPT ライブラリーから SALES という名前のソース表 (RMTRDB1 という名前のリレーショナル・データベースで、リモート・ジャーナルを使用する) を登録します。

```
ADDDPRREG SRCTBL(DEPT/SALES) SRCTBLRDB(RMTRDB1) RMTJRN(RMTJRNLIB/RMTJRN)
```

例 6: HR ライブラリーから EMPLOYEE ソース表を登録し、EMPNO、NAME、DEPT、および NETPAY の列についてのみ、変更をキャプチャーします。

```
ADDDPRREG SRCTBL(HR/EMPLOYEE) CAPCOL(EMPNO NAME DEPT NETPAY)
```

関連タスク:

- 637 ページの『表およびビューをレプリケーション・ソースとして登録する』

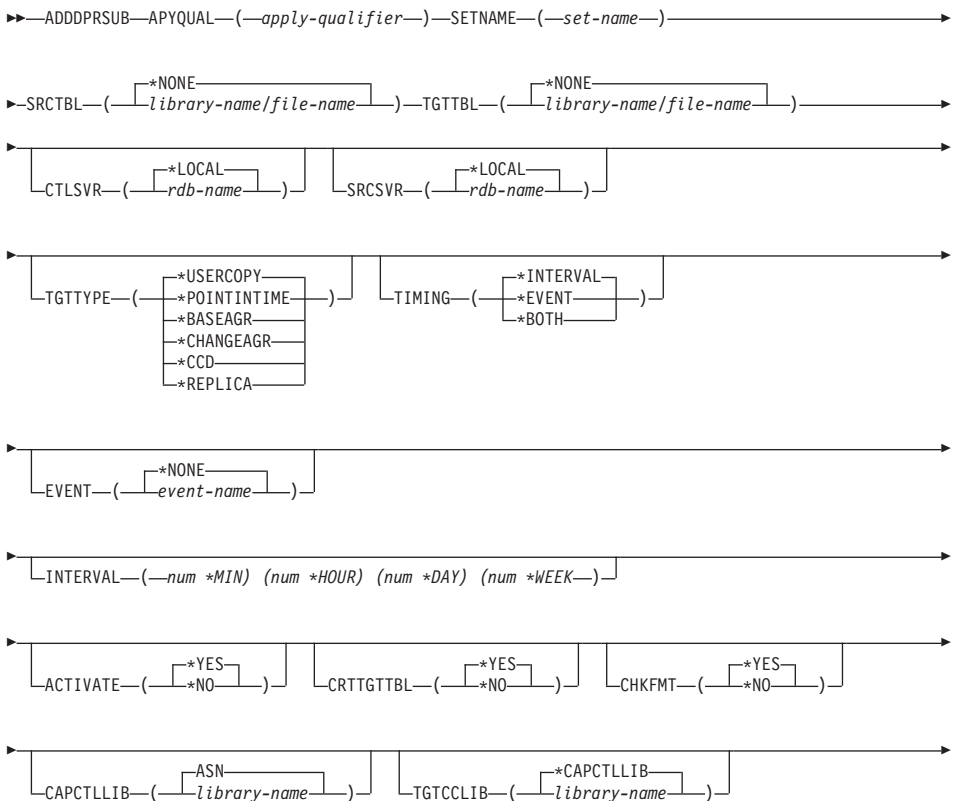
ADDDPRSUB: DPR サブスクリプション・セットの追加 (OS/400)

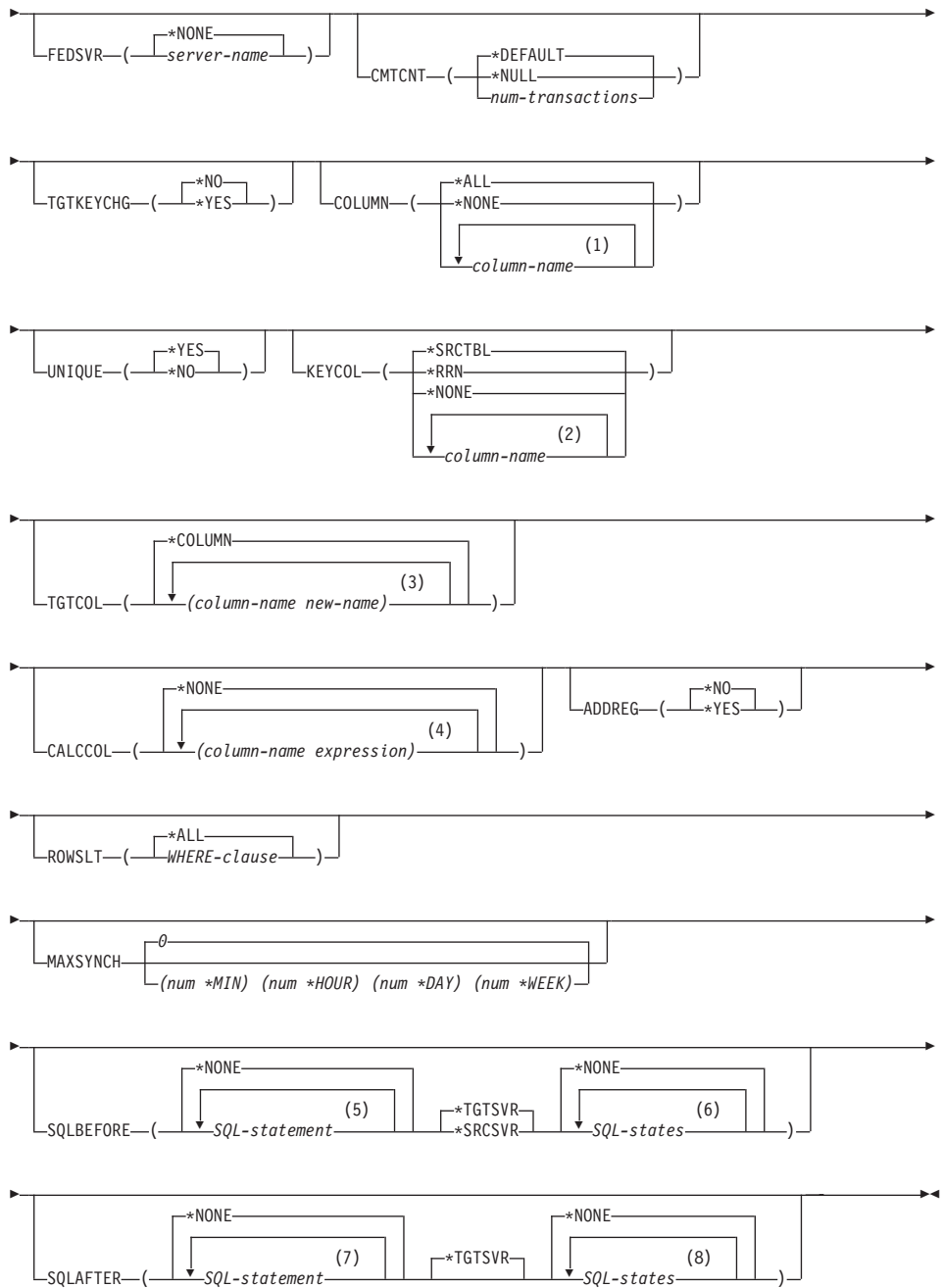
1 つのメンバーを持つ、またはメンバーを持たないサブスクリプション・セットを作成するには、DPR サブスクリプション・セットの追加 (**ADDDPRSUB**) コマンドを使用します。

コマンド行にコマンド名を入力したら、F4 キーを押してコマンド構文を表示することができます。

このコマンド、およびこのコマンドのすべてのパラメーターの詳しい記述を表示するには、画面の一番上のコマンドにカーソルを移動し、F1 キーを押します。特定のパラメーターの記述を表示するには、そのパラメーター上にカーソルを移動し、F1 キーを押します。

ADDDPRSUB コマンドを使用してサブスクリプション・セットを作成するための構文





注:

1 列名は 300 個まで指定できます。

ADDDPRSUB

- 2 列名は 120 個まで指定できます。
- 3 列名は 300 個まで指定できます。
- 4 100 個までの列名および式を指定できます。
- 5 SQL ステートメントは 3 個まで指定できます。
- 6 10 個までの SQLSTATES を指定できます。
- 7 SQL ステートメントは 3 個まで指定できます。
- 8 10 個までの SQLSTATES を指定できます。

表 165. ADDDPRSUB コマンド・パラメーター定義 (OS/400 用)

パラメーター	定義およびプロンプト
APYQUAL	<p>どのアプライ・プログラムがこのサブスクリプション・セットを処理するかを示すアプライ修飾子を指定します。アプライ修飾子の下のサブスクリプション・セットは別のジョブで実行されます。このパラメーターは必須です。</p> <p><i>apply-qualifier</i> アプライ修飾子の名前。</p>
SETNAME	<p>サブスクリプション・セットの名前を指定します。このパラメーターは必須です。</p> <p><i>set-name</i> サブスクリプション・セットの名前。入力するサブスクリプション・セット名は、指定されたアプライ修飾子についてユニークでなければなりません。ユニークでない場合、ADDDPRSUB コマンドはエラーになります。アプライ・プログラムはターゲット表のセットをグループとして扱うので、何らかの理由で 1 つのターゲット表が失敗すると、サブスクリプション・セット全体が失敗します。</p>
SRCTBL	<p>情報をサブスクリプション・セットにコピーするために使用する、ソース表の名前を指定します。この表をサブスクリプション・セットのメンバーにするには、この表をキャプチャー・コントロール・サーバーにあらかじめ登録しておく必要があります。このパラメーターは必須です。</p> <p>*NONE (デフォルト) このサブスクリプション・セットはソース・メンバーを持ちません。メンバーのないサブスクリプション・セットを作成する場合に使用します。</p> <p><i>library-name/file-name</i> ソース表の修飾名。1 つのメンバーを持つサブスクリプション・セットを作成する場合に使用します。</p>

表 165. ADDDPRSUB コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
TGTTBL	<p>ターゲット表の名前を指定します。 CRRTGTTBL パラメーターを *YES に設定し、ターゲット表が存在しないと、ターゲット表は自動的に作成されます。このパラメーターは必須です。</p> <p>*NONE (デフォルト) このサブスクリプション・セットはターゲット・メンバーを持ちません。メンバーのないサブスクリプション・セットを作成する場合に使用します。</p> <p><i>library-name/file-name</i> ターゲット表の修飾名。1 つのメンバーを持つサブスクリプション・セットを作成する場合に使用します。</p>
CTLSVR	<p>アプライ・コントロール表を含むシステムのリレーショナル・データベース名を指定します。</p> <p>*LOCAL (デフォルト) アプライ・コントロール表はローカルに存在します (ADDDPRSUB コマンドを実行するマシン上)。</p> <p><i>rdb-name</i> アプライ・コントロール表が置かれているリレーショナル・データベースの名前。 RDB ディレクトリー・エントリーの作業 (WRKRDBDIRE) コマンドを使用して、この名前を検出することができます。</p>
SRCSVR	<p>キャプチャー・コントロール表を含むシステムのリレーショナル・データベース名を指定します。</p> <p>*LOCAL (デフォルト) ソース表はローカル・マシンに登録されています (ADDDPRSUB コマンドを実行するマシン)。</p> <p><i>rdb-name</i> キャプチャー・コントロール表が存在するリレーショナル・データベースの名前。 RDB ディレクトリー・エントリーの作業 (WRKRDBDIRE) コマンドを使用して、この名前を検出することができます。</p>

表 165. ADDDPRSUB コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
TGTTYPE	<p>ターゲット表のタイプを指定します。これらのいずれかのタイプとしてターゲット表を作成した後、DPR 登録の追加 (ADDDPRREG) コマンドの SRCTBL パラメーターにこのパラメーター値を使用し、このターゲット表を multi-tier レプリケーションのソース表として登録することができます。</p> <p>*USERCOPY (デフォルト)</p> <p>ターゲット表はユーザー・コピーであり、これはソース表の内容のすべてまたは一部と一致する内容を持つ、ターゲット表です。ユーザー・コピーはポイント・イン・タイムのコピーのように扱われますが、ポイント・イン・タイムのターゲット表に存在する、DB2 DataPropagator for iSeries システム列は 1 つも含まれていません。</p> <p>この値は、KEYCOL パラメーターに *RRN の値が指定されている場合は無効です。</p> <p>SRCTBL パラメーターで指定した表は、ユーザー・データベース、ポイント・イン・タイム・コピー、または整合変更データ (CCD) のいずれかでなければなりません。</p> <p>重要: ターゲット表がすでに存在する場合、DB2 DataPropagator for iSeries は、これに対する変更を自動的にジャーナルに記録しません。ジャーナリングは、DB2 DataPropagator for iSeries の外側で開始する必要があります。</p>

表 165. ADDDPRSUB コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
TGTYPE (続き)	<p>*POINTINTIME</p> <p>ターゲット表はポイント・イン・タイム・コピーです。ポイント・イン・タイム・コピーは、ソース表の内容の一部またはすべてと一致する内容を持つターゲット表であり、 DB2 DataPropagator for iSeries システム列 (IBMSNAP_LOGMARKER) を保持します。この列は、特定の行がキャプチャー・コントロール・サーバーでいつ挿入または変更されたかを示します。</p> <p>*BASEAGR</p> <p>ターゲット表は基本集約コピーであり、これはソース表から集約された (算出された) データを含む、ターゲット表です。基本集約ターゲットのソース表は、ユーザー表またはポイント・イン・タイム表のどちらかでなければなりません。このターゲット表には、システム・タイム・スタンプ列の IBMSNAP_HLOGMARKER および IBMSNAP_LLOGMARKER が含まれます。</p> <p>*CHANGEAGR</p> <p>表は変更集約コピーであり、これは、変更データ (CD) 表の内容を基に集約された (算出された) データを含むターゲット表です。このターゲット表は、システム・タイム・スタンプ列 IBMSNAP_HLOGMARKER および IBMSNAP_LLOGMARKER を使用して作成されます。</p> <p>*CCD</p> <p>表は整合変更データ (CCD) 表であり、これは、変更データ (CD) 表と作業単位 (UOW) 表内のデータを結合したものから作成されたターゲット表です。 CCD 表は、アプライ・プログラムのためのトランザクション整合データを提供し、以下の列が入っている必要があります。</p> <ul style="list-style-type: none"> • IBMSNAP_INTENTSEQ • IBMSNAP_OPERATION • IBMSNAP_COMMITSEQ • IBMSNAP_LOGMARKER <p>*REPLICA</p> <p>ターゲット表はレプリカ表であり、これは update-anywhere レプリケーションにのみ使用されます。レプリカ・ターゲット表はマスター・ソース表から変更を受信し、またレプリカ・ターゲット表への変更は、マスター・ソース表に戻して伝搬されます。レプリカ表はソース表として自動的に登録されます。</p>

表 165. ADDDPRSUB コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
TIMING	<p>アプライ・プログラムがサブスクリプション・セットの処理に使用するタイミング (スケジューリング) のタイプを指定します。</p> <p>*INTERVAL (デフォルト) アプライ・プログラムは、サブスクリプション・セットを特定の時間インターバル (たとえば、1 日に 1 度) で処理します。</p> <p>*EVENT アプライ・プログラムは、特定のイベントが起こった時にサブスクリプション・セットを処理します。</p> <p>*BOTH アプライ・プログラムは、特定の時間インターバル、またはイベントが起こった時のどちらでも、最初に起こった時に処理します。</p>
EVENT	<p>イベントを指定します。入力するイベントは、サブスクリプション・イベント表 (IBMSNAP_SUBS_EVENT) のイベント名と一致する必要があります。</p> <p>*NONE (デフォルト) イベントは使用しません。</p> <p><i>event-name</i> IBMSNAP_SUBS_EVENT 表に記述されているイベントを表す、ユニークの文字ストリング。</p>

表 165. ADDDPRSUB コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
INTERVAL	<p>ターゲット・コピーのリフレッシュを行う時間インターバルを指定します。これは開始時刻から開始時刻までの時間インターバルであり、週、日、時間、および分で指定します。これは 2 つの部分からなる値です。最初の部分は数値、2 番目の部分は時間の単位です。</p> <p>*MIN 分</p> <p>*HOUR 時間</p> <p>*DAY 日</p> <p>*WEEK 週</p> <p>時間の単位と一緒に、数値の組み合わせを指定することができます。たとえば、((2 *WEEK) (3 *DAY) (35 *MIN)) は、2 週、3 日、および 35 分の時間インターバルを指定します。同じ時間単位で複数を指定すると、最後の指定が使用されます。</p>
ACTIVATE	<p>サブスクリプション・セットがアクティブかどうかを指定します。アプライ・プログラムは、このパラメーターが *YES でない場合、このサブスクリプション・セットを処理しません。</p> <p>*YES (デフォルト) サブスクリプション・セットはアクティブです。</p> <p>*NO サブスクリプション・セットはアクティブではありません。</p>

表 165. ADDDPRSUB コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
CRTTGTTBL	ターゲット表 (またはビュー) を作成するかどうかを指定します。

***YES** (デフォルト)

ターゲット表 (またはビュー) が存在しないければ、作成します。存在する場合は、既存の表またはビューがターゲットになり、この既存の表またはビューのフォーマットの **CHKFMT** パラメーターが ***YES** に設定されているかどうかをチェックします。 **UNIQUE** および **KEYCOL** パラメーターに指定された値を使用して、(そのような索引がまだ存在しなければ) ターゲット表に追加の索引が作成されます。既存のターゲット表に、追加索引の条件に違反するような行が含まれていると、コマンドは失敗します。

***NO**

ターゲット表またはビューを作成しません。アプライ・プログラムを始動する前に、正しい属性を使用して表またはビューを作成する必要があります。

表またはビューが存在する場合に **CHKFMT** を ***YES** にすると、**ADDDPRSUB** コマンドは、既存の表のフォーマットが、設定されたサブスクリプション・セット定義と一致することを確認します。**CHKFMT** を ***NO** にする場合は、ユーザーは既存の表のフォーマットがサブスクリプション・セット定義と一致することを確認しておく必要があります。

重要: 表またはビューがすでに存在する場合、DB2 DataPropagator for iSeries は、既存のオブジェクトへの変更を自動的にジャーナルに記録しません。ジャーナリングは、DB2 DataPropagator for iSeries の外側で開始する必要があります。

表 165. ADDDPRSUB コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
CHKFMT	<p>DB2 DataPropagator for iSeries が、サブスクリプション・セットとターゲット表をチェックし、列が一致することを確認するかどうかを指定します。このパラメーターは、 CRTTGTTBL パラメーターが *YES の場合は無視され、また CRTTGTTBL パラメーターが *NO でターゲット表が存在しない場合も無視されます。</p> <p>*YES (デフォルト) DB2 DataPropagator for iSeries は、このサブスクリプション・セットに定義された列がターゲット表内の列と一致するか検証します。両者が一致しないと、このコマンドは失敗します。</p> <p>*NO DB2 DataPropagator for iSeries は、サブスクリプション・セットと既存のターゲット表間の相違を無視します。ユーザーはターゲット表がサブスクリプション・セットと互換性があることを確認する必要があります。</p>
CAPCTLLIB	<p>キャプチャー・スキーマ (キャプチャー・コントロール表が存在するライブラリーの名前) を指定します。これらのキャプチャー・コントロール表は、このサブスクリプション・セットのソースを処理します。</p> <p>ASN (デフォルト) キャプチャー・コントロール表は ASN ライブラリー内にあります。</p> <p><i>library-name</i> キャプチャー・コントロール表を含むライブラリーの名前。これは、ソース表が登録されたライブラリーです。</p>
TGTCCLIB	<p>ターゲット・コントロール・ライブラリーを指定します。</p> <p>*CAPCTLLIB (デフォルト) ターゲット・コントロール・ライブラリーは、キャプチャー・コントロール表が存在するライブラリーと同じです。</p> <p><i>library-name</i> ターゲット・コントロール表を含むライブラリーの名前。</p> <p>ターゲット表を別のサブスクリプション・セット (たとえば外部 CCD 表など) のソースとして使用する場合、このパラメーター値は、この表がソースとして使用される時のキャプチャー・スキーマです。</p>

表 165. ADDDPRSUB コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
FEDSVR	<p>このサブスクリプション・セットのソースが連合データベース・システムであるかどうかを指定します。</p> <p>*NONE (デフォルト) ソース・サーバーは連合データベース・システムではありません。</p> <p><i>server-name</i> このサブスクリプション・セットの連合データベース・システムの 名前 (DB2 以外のリレーショナル・ソースの場合)。</p>
CMTCNT	<p>コミットメント・カウントを指定します。これは、アプライ・プログラムがトランザクションをいくつ処理したらコミットするかを示す数です。</p> <p>*DEFAULT (デフォルト) 使用する値をコマンドが決めます。TGTTYE が *REPLICA の場合、CMTCNT はゼロ (0) です。TGTTYE が *REPLICA 以外の場合、CMTCNT は NULL です。</p> <p>*NULL サブスクリプション・セットは読み取り専用です。アプライ・プログラムは、サブスクリプション・セット・メンバーの応答セットを 1 度に 1 メンバーずつフェッチし、すべてのデータの処理を終了した後、サブスクリプション・セット全体について 1 つのコミットを出します。</p> <p><i>num-transactions</i> いくつのトランザクション処理を処理したらアプライ・プログラムが変更をコミットするかを示す数を指定します。このパラメーターは、TGTTYE パラメーターが *REPLICA の場合のみ有効です。</p>

表 165. ADDDPRSUB コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
TGTKEYCHG	<p>ターゲット表のターゲット・キー列の一部であるソース列に変更があった場合、アプライ・プログラムがその更新をどのように扱うかを指定します。このパラメーターは、ADDDPRREG コマンドの USEDELINS パラメーターと組み合わせて働きます。</p> <ul style="list-style-type: none"> • USEDELINS が YES で TGTKEYCHG が YES の場合、更新はできません。 • USEDELINS が YES で TGTKEYCHG が NO の場合、更新は削除と挿入の対になります。 • USEDELINS が NO で TGTKEYCHG が YES の場合、アプライ・プログラムは特別な論理を使用してこの条件を扱います。 • USEDELINS が NO で TGTKEYCHG が NO の場合、アプライ・プログラムは変更を通常の更新として処理します。 <p>*NO (デフォルト) ソース表に対する更新は、キャプチャー・プログラムによりステージ化され、アプライ・プログラムによりターゲット表に処理が行われます。</p> <p>*YES アプライ・プログラムは、ターゲット・キー列の変更前イメージに基づいてターゲット表を更新します。つまり、アプライ・プログラムは述部を新しい値ではなく、古い値に変更します。</p>

表 165. ADDDPRSUB コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
COLUMN	<p>ターゲット表に含める列を指定します。列名は修飾できません。列名は、ソース表の登録時に CAPCOL パラメーターに指定した列名のリストから選択してください。</p> <p>この表の登録時に IMAGE パラメーターを *BOTH に設定した場合は、変更前イメージ列名を指定することができます。変更前イメージ列名は、接頭部を持つオリジナルの列名です。この接頭部は、ADDDPRREG コマンドの PREFIX パラメーターに指定した文字です。</p> <p>*ALL (デフォルト) ソースに登録した列のすべてがターゲット表に含まれます。</p> <p>*NONE ソース表からの列は 1 つもターゲット表に含まれません。 *NONE は、算出列だけをターゲット表に含めたい場合に使用します。この値は、CALCCOL パラメーターに合計関数が含まれているが、GROUP BY が実行されない、という場合に必要です。</p> <p><i>column-name</i> ターゲット表に含めたいソース列の名前を 300 個まで指定できます。列名はスペースで区切ります。</p>

表 165. ADDDPRSUB コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
UNIQUE	<p>ターゲット表が KEYCOL パラメーターで示されたユニーク・キーを持つかどうかを指定します。</p> <p>*YES (デフォルト)</p> <p>ターゲット表はキーごとに正味 1 つの変更をサポートします。つまり、キーに対していかに多くの変更がなされたとしても、そのキーについてはターゲット表に 1 つの行しか存在しないということです。</p> <p>この値は、表がデータの変更の履歴ではなく、現行のデータを含むことを指定します。コンデンス表には、1 つの主キー値に対して複数の行が含まれることはなく、リフレッシュ用の最新情報を提供するために使用できます。</p> <p>*NO</p> <p>ターゲット表はキーごとに複数の変更をサポートします。変更はターゲット表に付加されます。</p> <p>この値は、表が現行のデータではなく、変更の履歴を含むことを指定します。非コンデンス表には、それぞれのキー値に対して複数の行が含まれ、データの変更履歴を提供するために使用できます。ただし、非コンデンス表はリフレッシュ用の最新データは提供できません。</p>

表 165. ADDDPRSUB コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
KEYCOL	<p>ターゲット表のキーを記述する列を指定します。列名は修飾できません。*POINTINTIME、*REPLICA、および *USERCOPY ターゲット表 (TGTTYE パラメーターで指定されている) の場合、ターゲット表に 1 つまたは複数の列をターゲット・キーとして指定する必要があります。アプライ・プログラムはこのターゲット・キーを使用して、変更キャプチャー・レプリケーション中に、変更された個々のユニークな行を識別します。</p>
*SRCTBL (デフォルト)	<p>ターゲット表のキー列は、ソース表のキー列と同じです。 ADDDPRREG コマンドは、ソース表がキー付きの場合、ソース表に指定されたキーを使用します。以下のキー列が使用されます。</p> <ul style="list-style-type: none"> • 物理ファイル作成コマンド (CRTPF) を使用して表を作成した時に、DDS を使用して定義したキー列 • CREATE TABLE および ALTER TABLE SQL ステートメントを使用して定義した、主キーおよびユニーク・キー • CREATE INDEX SQL ステートメントを使用して定義したユニーク・キー <p>1 つの列を、キーとして、異なる順序付けで複数回使用すると、ターゲット表のキーは昇順で定義されます。</p>
*RRN	<p>ターゲット表のキー列は IBMQSQ_RRN 列です。ターゲット表は IBMQSQ_RRN 列を使用して作成され、この列がキーとして使用されます。アプライ・プログラムの実行時に、ソース表がユーザー表であり、ターゲット表がポイント・イン・タイムまたはユーザー・コピーの場合、ターゲット表の IBMQSQ_RRN 列が、ソース表内の関連するレコードの RRN (相対レコード番号) で更新されます。それ以外では、ターゲット表の IBMQSQ_RRN 列は、ソース表内の IBMQSQ_RRN 列の値で更新されます。</p>
*NONE	<p>ターゲット・コピーはターゲット・キーを含みません。ターゲット表のタイプが *POINTINTIME、*REPLICA、または *USERCOPY の場合、*NONE は指定できません。</p>
<i>column-name</i>	<p>ターゲット・キー列として使用したいターゲット列の名前。列名は 120 個まで指定できます。列名はスペースで区切ります。</p>

表 165. ADDDPRSUB コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
TGTCOL	<p>アプライ・プログラムがターゲット表内で更新するすべての列の新しい名前を指定します。これらの名前は、ソース表から取られた列名をオーバーライドします。列名は修飾できません。 COLUMN パラメーターに *NONE を指定した場合はこのパラメーターを使用しないでください。</p> <p>このパラメーターを使用して、ターゲット表の列により分かりやすい名前を付けることができます。それぞれのソース列の名前と、ターゲット表の対応する列の名前を指定します。</p> <p>*COLUMN (デフォルト) ターゲット列は、COLUMN パラメーターに指定した列と同じです。</p> <p><i>column-name</i> ターゲットではその名前を変更したい、ソース表からの列名。列名は 300 個まで指定できます。</p> <p><i>new-name</i> ターゲット列の新しい名前。新しい列名を 300 個まで指定できます。このパラメーターを使用しないと、ターゲット表の列名はソースの列名と同じになります。</p>
CALCCOL	<p>ターゲット表のユーザー定義の列または算出された列のリストを指定します。列名は修飾できません。それぞれの列名と式の対を括弧で囲みます。</p> <p>各 SQL 式には列名を指定する必要があります。 GROUP BY ステートメントのない SQL 式として列を定義する場合は、COLUMN パラメーターを *NONE にする必要があります。</p> <p>*NONE (デフォルト) ターゲット表はユーザー定義の列または算出された列を含みません。</p> <p><i>column-name</i> ターゲット表のユーザー定義の列または算出された列の列名。列名は 100 個まで指定できます。</p> <p><i>expression</i> ターゲット表のユーザー定義の列または算出された列の式。 SQL 列の式は 100 個まで指定できます。</p>

表 165. ADDDPRSUB コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
ADDREG	<p>ターゲット表をソース表として自動的に登録するかどうかを指定します。このパラメーターは CCD ターゲット・タイプの表を登録する場合に使用します。</p> <p>*NO (デフォルト) ターゲット表はソース表として登録されません。 DB2 DataPropagator for iSeries は、ターゲット・タイプが *REPLICA の場合、このパラメーター値を無視します。レプリカ・ターゲット表は必ず、ソース表として自動的に登録されます。</p> <p>*YES ターゲット表はソース表として登録されます。ターゲット表をすでにユーザーが登録していると、このコマンドは失敗します。</p> <p>ターゲット表のタイプが *USERCOPY、*POINTINTIME、*BASEAGR、または *CHANGEAGR の場合、このパラメーターを *YES にしないでください。</p> <p>CRTTGTTBL パラメーターを *NO にした場合、これをソースとして登録する前に、ターゲット表を作成する必要があります。</p>
ROWSLT	<p>SQL の WHERE 文節に入れる述部を指定します。アプライ・プログラムはこれらの述部を使用して、ソースの変更データ (CD) 表のどの行をターゲット表に適用するかを決めます。ソースの変更のサブセットだけをターゲット表に複製したい場合に、このパラメーターを使用します。</p> <p>*ALL (デフォルト) アプライ・プログラムは、CD 表内のすべての変更をターゲット表に適用します。</p> <p><i>WHERE-clause</i> アプライ・プログラムが CD 表からどの行をターゲット表に適用するかを指定する SQL の WHERE 文節です。WHERE キーワードはこのパラメーターで暗黙に想定されているので、WHERE キーワードを含めないでください。この WHERE 文節は、この文節を実行するデータ・サーバー上で有効なものでなければなりません。</p> <p>注: このパラメーターの WHERE 文節は、SQLBEFORE または SQLAFTER パラメーターに指定された WHERE 文節とはまったく関係ありません。</p>

表 165. ADDDPRSUB コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
MAXSYNCH	<p>最大の同期化の分数を指定します。このパラメーターは、時間しきい値の制限を指定し、サブスクリプションのサイクル中に、キャプチャー・プログラムおよびアプライ・プログラムが処理する変更データの量を規制するために使用されます。時間しきい値の制限は 2 つの部分からなる値で指定します。最初の部分は数値、2 番目の部分は時間の単位です。</p> <p>*MIN 分</p> <p>*HOUR 時間</p> <p>*DAY 日</p> <p>*WEEK 週</p> <p>時間の単位と一緒に、数値の組み合わせを指定することができます。たとえば、((1 *WEEK) (2 *DAY) (35 *MIN)) は、1 週、2 日、および 35 分の時間インターバルを指定します。同じ時間単位で複数指定すると、最後の指定が使用されます。</p> <p>デフォルトはゼロ (0) であり、変更データのすべてを適用することを示します。</p>

表 165. ADDDPRSUB コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
SQLBEFORE	<p>アプライ・プログラムがターゲット表をリフレッシュする前に実行する、SQL ステートメントを指定します。このパラメーターは以下の 3 つのエレメントからなります。</p> <p>エレメント 1: SQL コード</p> <p>*NONE (デフォルト) SQL ステートメントを指定しません。</p> <p><i>SQL-statement</i></p> <p>実行したい SQL ステートメント。SQL ステートメントの構文が正しいことを確認してください。DB2 DataPropagator for iSeries は構文の妥当性を検査しません。また、適切な SQL 命名規則を使用する必要があります。SQL ファイル参照は、システムの命名規則 (LIBRARY/FILE) ではなく、LIBRARY.FILE の形式でなければなりません。3 つまでの SQL ステートメントを指定できます。</p> <p>エレメント 2: 実行するサーバー</p> <p>*TGTSVR (デフォルト) SQL ステートメントは、ターゲット表が存在するターゲット・サーバーで実行されます。</p> <p>*SRCSVR SQL ステートメントは、ソース表が存在するキャプチャー・コントロール・サーバーで実行されます。</p> <p>エレメント 3: 許される SQLSTATE 値</p> <p>*NONE (デフォルト) SQLSTATE 値 00000 のみが正常と見なされます。</p> <p><i>SQL-states</i></p> <p>1 から 10 個までの許容される SQLSTATE 値のリスト。 SQLSTATE 値はスペースで区切ります。SQLSTATE 値は、00000 から FFFFF の範囲の 5 桁の 16 進数です。</p> <p>SQL ステートメントは、SQLSTATE 値 00000 または、リストされた許容可能な SQLSTATE 値の 1 つで完了した場合に、正常終了となります。</p>

表 165. ADDDPRSUB コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
SQLAFTER	<p>アプライ・プログラムがターゲット表をリフレッシュした後に実行する、SQL ステートメントを指定します。このパラメーターは以下の 3 つのエレメントからなります。</p> <p>エレメント 1: SQL コード</p> <p>*NONE (デフォルト) SQL ステートメントを指定しません。</p> <p><i>SQL-statement</i></p> <p>実行したい SQL ステートメント。SQL ステートメントの構文が正しいことを確認してください。DB2 DataPropagator for iSeries は構文の妥当性を検査しません。また、適切な SQL 命名規則を使用する必要があります。SQL ファイル参照は、システムの命名規則 (LIBRARY/FILE) ではなく、LIBRARY.FILE の形式でなければなりません。3 つまでの SQL ステートメントを指定できます。</p> <p>エレメント 2: 実行するサーバー</p> <p>*TGTSVR (デフォルト) SQL ステートメントは、ターゲット表が存在するターゲット・サーバーで実行されます。</p> <p>エレメント 3: 許される SQLSTATE 値</p> <p>*NONE (デフォルト) SQLSTATE 値 00000 のみが正常と見なされます。</p> <p><i>SQL-states</i></p> <p>1 から 10 個までの許容される SQLSTATE 値のリスト。SQLSTATE 値はスペースで区切ります。SQLSTATE 値は、00000 から FFFFF の範囲の 5 桁の 16 進数です。</p> <p>SQL ステートメントは、SQLSTATE 値 00000 または、リストされた許容可能な SQLSTATE 値の 1 つで完了した場合に、正常終了となります。</p>

ADDDPRSUB の例

以下の例は、ADDDPRSUB コマンドの使用法を示しています。

例 1: AQHR アプライ修飾子の下に SETHR という名前のサブスクリプション・セットを作成します。

ADDDPRSUB

```
ADDDPRSUB APYQUAL(AQHR) SETNAME(SETHR) SRCTBL(HR/EMPLOYEE)
TGTTBL(TGTLIB/TGTEMPL)
```

このサブスクリプション・セットには 1 つのサブスクリプション・セット・メンバーが含まれ、HR ライブラリー下の EMPLOYEE という名前の登録済みソース表から、TGTLIB ライブラリー下の TGTEMPL という名前のターゲット表にデータを複製します。

例 2: 2 つだけの列 (EMPNO (キー) および NAME) を持つ、SETHR という名前のサブスクリプション・セットを、EMPLOYEE という名前の登録済みソース表から作成し、これらの列を TGTEMPL という名前の既存のターゲット表に複製します。

```
ADDDPRSUB APYQUAL(AQHR) SETNAME(SETHR) SRCTBL(HR/EMPLOYEE)
TGTTBL(TGTLIB/TGTEMPL) CRTTGTTBL(*NO) COLUMN(EMPNO NAME) KEYCOL(EMPNO)
```

例 3: SETHR という名前のサブスクリプション・セットを、EMPLOYEE という名前の登録済みソース表からのデータを使用して作成し、このデータを TGTREPL という名前のレプリカ・タイプのターゲット表に複製します。

```
ADDDPRSUB APYQUAL(AQHR) SETNAME(SETHR) SRCTBL(HR/EMPLOYEE)
TGTTBL(TGTLIB/TGTREPL) TGTTYPE(*REPLICA)
```

例 4: NOMEM という名前のサブスクリプション・セットを、サブスクリプション・セット・メンバーなしで作成します。

```
ADDDPRSUB APYQUAL(AQHR) SETNAME(NOMEM) SRCTBL(*NONE) TGTTBL(*NONE)
```

関連タスク:

- 687 ページの『ソースのサブスクリプト』

RVKDPRAUT: 権限の取り消し (OS/400)

DPR 権限の取り消し (RVKDPRAUT) コマンドは、ユーザーがレプリケーション・ソースおよびサブスクリプション・セットを定義または変更を行うことができないようにするため、レプリケーション・コントロール表に対する権限を取り消します。

コマンド行にコマンド名を入力したら、F4 キーを押してコマンド構文を表示することができます。

このコマンド、およびこのコマンドのすべてのパラメーターの詳しい記述を表示するには、画面の一番上のコマンドにカーソルを移動し、F1 キーを押します。特定のパラメーターの記述を表示するには、そのパラメーター上にカーソルを移動し、F1 キーを押します。

RVKDPRAUT コマンドを使用してレプリケーション・コントロール表の権限を取り消す方法



表 166. RVKDPRAUT コマンド・パラメーター定義 (OS/400 用)

パラメーター	定義およびプロンプト
CAPCTLLIB	ユーザー権限が取り消されるライブラリーの名前である、キャプチャー・スキーマを指定します。
ASN (デフォルト)	キャプチャー・コントロール表は ASN ライブラリー内にあります。
<i>library-name</i>	レプリケーション・コントロール表を含むライブラリーの名前。
USER	権限が取り消されるユーザーを指定します。このパラメーターは必須です。
<i>user-name</i>	権限が取り消されるユーザーの名前を最大 50 まで指定します。
*PUBLIC	特定の許可を所有していないすべてのユーザー、許可リストにないすべてのユーザー、およびグループ・プロファイルが許可を所有していないすべてのユーザーから、許可を取り消すように指定します。

使用上の注意

以下のいずれかの状態が生じると、このコマンドはエラー・メッセージを戻します。

- 指定したユーザーが存在しない。
- コマンドを実行しているユーザーが、指定したユーザー・プロファイルへの権限を所有していない。
- このコマンドを実行しているユーザーには、DB2 DataPropagator for iSeries のコントロール表に対する権限を取り消す許可がない。
- DB2 DataPropagator for iSeries のコントロール表が存在しない。
- キャプチャー・プログラムまたはアプライ・プログラムが実行中。

RVKDPRAUT の例

以下の例は、RVKDPRAUT コマンドの使用方法を示しています。

例 1: ASN ライブラリー下のコントロール表に対する権限を、HJONES という名前のユーザーから取り消すには、以下のようにします。

RVKDPAUT CAPCTLLIB(ASN) USER(HJONES)

例 2: **GRTDPRAUT** コマンドに指定されていないすべてのユーザーの権限を取り消し、ASN ライブラリー内のコントロール表にアクセスできないようにするには、以下のようにします。

RVKDPAUT CAPCTLLIB(ASN) USER(*PUBLIC)

関連タスク:

- 663 ページの『レプリケーションの設定』

関連資料:

- レプリケーションのガイドおよびリファレンス の『GRTDPRAUT: ユーザーの許可 (OS/400)』

RMVDPRREG: DPR 登録の除去 (OS/400)

ソース表がレプリケーションの目的で使用されないようにするために、登録 (IBMSNAP_REGISTER) 表から 1 つのソース表を除去するには、DPR 登録の除去 (**RMVDPRREG**) コマンドを使用します。

コマンド行にコマンド名を入力したら、F4 キーを押してコマンド構文を表示することができます。

このコマンド、およびこのコマンドのすべてのパラメーターの詳しい記述を表示するには、画面の一番上のコマンドにカーソルを移動し、F1 キーを押します。特定のパラメーターの記述を表示するには、そのパラメーター上にカーソルを移動し、F1 キーを押します。

RMVDPRREG コマンドを使用して DPR 登録を除去する方法

▶—RMVDPRREG—SRCTBL(—*library-name/file-name*—)—————▶

▶—————▶
 ┌──────────┬──────────┬──────────┐
 │CAPCTLLIB(—┬──ASN──┬──*library-name*──┬──)──┐
 └──────────┴──────────┴──────────┘

表 167. **RMVDPRREG** コマンド・パラメーター定義 (OS/400 用)

パラメーター	定義およびプロンプト
SRCTBL	除去する登録を識別します。このパラメーターは必須です。 <i>library-name/file-name</i> 登録済みファイルの修飾名。

表 167. RMVDPREG コマンド・パラメーター定義 (OS/400 用) (続き)

パラメーター	定義およびプロンプト
CAPCTLLIB	キャプチャー・スキーマ (キャプチャー・コントロール表が存在するライブラリーの名前) を指定します。 ASN (デフォルト) キャプチャー・コントロール表は ASN ライブラリー内にありません。 <i>library-name</i> キャプチャー・コントロール表を含むライブラリーの名前。

RMVDPREG の例

以下の例は、**RMVDPREG** コマンドの使用方法を示しています。

例 1: デフォルト ASN キャプチャー・スキーマ内の HR ライブラリーの、EMPLOYEE という名前のソース表の登録を除去するには、以下のようにします。

```
RMVDPREG SRCTBL(HR/EMPLOYEE)
```

例 2: BSN という名前のキャプチャー・スキーマ内の DEPT ライブラリーの、SALES という名前のソース表の登録を除去するには、以下のようにします。

```
RMVDPREG SRCTBL(DEPT/SALES) CAPCTLLIB(BSN)
```

関連タスク:

- レプリケーションのガイドおよびリファレンス の『レプリケーション環境の変更』

特殊なデータ・タイプのレプリケーション

LOB、DATALINK、ROWID、または DB2 以外のデータ・タイプなど、特殊なデータ・タイプを複製するときには、一定の条件および制約事項がありますので注意してください。場合によっては、DB2 レプリケーションでこれらのデータ・タイプを処理できるように追加のセットアップ・ステップを実行する必要があります。この章では、これらの条件および制約事項について説明します。この章は以下のセクションから構成されています。

- 『レプリケーションにおける一般的なデータの制約事項』
- 604 ページの『ラージ・オブジェクトのレプリケーション』
- 605 ページの『DATALINK 値のレプリケーション』

レプリケーションにおける一般的なデータの制約事項

DB2 レプリケーションでは現在のところ、特定のデータ・タイプに関して制約事項があります。

- データ暗号化の制約事項

DB2 レプリケーションでは、暗号化されたデータをレプリケーションすることはできません。

• データ・タイプの制約事項

DB2 レプリケーションでは、いかなる状況でも以下のデータ・タイプをレプリケーションすることはできません。

- DB2 以外のリレーショナル・ソースからの LOB 列
- 以下のいずれかのプロシージャが定義されている場所の列
 - EDITPROC
 - FIELDPROC
 - VALIDPROC

DB2 レプリケーションは、ソース表およびターゲット表が DB2 for z/OS にある場合は、長い可変グラフィック (LONG VARGRAPHIC) データをレプリケーションできません。

DB2 レプリケーションでは、要約データ・タイプを含む表をレプリケーションすることはできません。

DB2 では、地理情報データ・タイプ列を含む表をレプリケーションすることはできませんが、実際の地理情報データ・タイプ列をレプリケーションすることはできません。

ユーザー定義のデータ・タイプ (DB2 Universal Database の特殊データ・タイプ) は、レプリケーションされる前に変更データ (CD) 表で基本データ・タイプに変換されます。さらに、DB2 レプリケーションがターゲット表をサブスクリプション・セット・メンバー定義として作成した場合、ユーザー定義タイプは CD 表内と同様に、ターゲット表内で基本データ・タイプに変換されます。

DB2 レプリケーションは、Oracle ソース表からのタイプ LONG の列は複製できません。これは、Oracle のトリガーが LONG 列をサポートしていないからです。

ラージ・オブジェクトのレプリケーション

DB2 Universal Database はラージ・オブジェクト (LOB) のデータ・タイプをサポートします。このデータ・タイプには、バイナリー LOB (BLOB)、文字 LOB (CLOB)、2 バイト文字 LOB (DBCLOB) が含まれます。ここでは、ここに示したタイプの LOB データをすべて取り上げます。

キャプチャー・プログラムはログ・レコード内の LOB 記述子を読み、変更されているために複製する必要があるデータが LOB 列にあるかどうかを判別しますが、変更データ (CD) 表には LOB データをコピーしません。LOB 列が変更されると、キャプチャー・プログラムは CD 表に標識を設定します。アプライン・プログラムはこの標識を読み取ると、次に LOB 列全体 (LOB 列の変更部分だけではない) をソース表からターゲット表に直接コピーします。

LOB 列には最大 2 GB のデータを入れることができるので、アプライ・プログラムには十分なネットワーク帯域幅を指定できるようにしてください。同じように、ターゲット表にも LOB データが入るだけのディスク・スペースを指定する必要があります。

制約事項:

- アプライ・プログラムは、常に最新バージョンの LOB 列をソース表 (CD 表ではない) から直接コピーします。これはその列が CD 表の他の列よりも新しい場合にも当てはまります。このため、ターゲット行の LOB 列が変更された場合この LOB 列は、そのターゲット行の残りのデータと矛盾するものになる可能性があります。ターゲット行でデータの矛盾が発生する可能性を最小化するために、アプライ・サイクルのインターバルは、アプリケーションで実用上の問題が起きない範囲で、できるだけ短くしてください。
- 1 つの表で複製できる LOB 列は 10 以下です。10 よりも多い LOB 列を含む表を登録すると、アプライ・プログラムからエラー・メッセージが戻されます。1 つの表に 10 よりも多くの LOB 列を登録しようとする、レプリケーション・センターからエラー・メッセージが戻されます。
- LOB データは読み取り専用表にのみコピーできます。したがって、LOB データをレプリカ表に複製することはできません。
- DB2 for OS/390 バージョン 6 (またはそれ以降) と DB2 Universal Database (その他のオペレーティング・システム用) の間で LOB データをコピーするには、DB2 Connect 7 以降が必要です。
- ニックネームを使って LOB データを参照することはできません。
- LOB、DATALINK、または ROWID 列の変更前イメージ値はサポートされていません。
- エクステンダーの LOB 列データに関連付けられた追加のコントロール・ファイルがデータベースの外部で保守されている、テキスト、音声、ビデオ、イメージ、その他のエクステンダーの場合、DB2 エクステンダー™ におけるレプリケーションはサポートしていません。
- DB2 は、LOB 全体のレプリケーションしかできません。LOB の部分的なレプリケーションはできません。
- OS/400 上のレプリケーション環境でリモート・ジャーナル・セットアップを使用する場合は、LOB 列はレプリケーションできません。

DATALINK 値のレプリケーション

リモート・ネットワークを介して大きなファイル (マルチメディア・データなど) にアクセスするのは、非効率的で費用がかかります。外部ファイル・システムに保管されるデータを表す DATALINK データ・タイプを使用すると、構造化されていないファイルに、より迅速にアクセスし、ファイルを複製できます。

DB2 Universal Database は、アクセス・コントロール、参照保全、および構造化されていない大きなファイルのリカバーをデータベースで管理できるようにする、

DATALINK データ・タイプをサポートしています。DB2 Universal Database は、以下のオペレーティング・システムで DATALINK の値をサポートします。

- AIX
- Solaris™ 実行環境版
- Windows NT
- Windows 2000
- OS/400

DATALINK 列の値には、外部ファイルのロケーションを示す URL が入っています。DB2 レプリケーションは、DATALINK 列の値、およびここから参照されるファイルをレプリケーションするときに、以下のコンポーネントを使用します。

ASNDLCOPY 出口ルーチン

ソース・ファイル・システム上の URL をターゲット・ファイル・システム上の URL にマップした後、適切なファイル・コピー・デーモンに接続して、URL がポイントする外部ファイルを複製します。

Data Links Manager レプリケーション・デーモン (DLFM_ASNCOPYD)

ASNDLCOPY 出口ルーチンと共同で、DATALINK 列の値から参照されるファイルをコピーします。DLFM_ASNCOPYD デーモンは DB2 Data Links Manager バージョン 8 の一部です。このデーモンは、AIX、Solaris™ 実行環境版、および Windows オペレーティング・システムで使用できます。

ASNDLCOPYD デーモン

ASNDLCOPY 出口ルーチンと共同で、DATALINK 列の値から参照されるファイルをコピーします。このデーモンは DB2 for iSeries で提供されます。ASNDLCOPYD は OS/400 で、またオプションとして、他のオペレーティング・システムで使用してください。

アプライ・プログラムはデータ・タイプが DATALINK のデータを読み取ると、参照データを予備ファイルに入れるとともに、更新されたファイルの URL を入力ファイルに入れます。

アプライ・プログラムは続いて ASNDLCOPY 出口ルーチンを呼び出します。この ASNDLCOPY 出口ルーチンは、ソース・ファイル・システム上に物理ファイルが存在することを確認し、ターゲット・ファイル・システム上の対応するファイルに URL をマップし、このターゲット・ファイルのロケーションを結果ファイルに保管し、適切なファイル・コピー・デーモン (DLFM_ASNCOPYD、ASNDLCOPYD、または FTP) に接続し、ソース・ファイル・システムからターゲット・ファイル・システムへ外部ファイルをコピーします。

推奨: アプライ・プログラムは ASNDLCOPY ルーチンの処理が完了するのを待ってからサブスクリプション・セットのレプリケーションを完了させるため、DATALINK 列に対しては、別個のサブスクリプション・セットを使用してください。外部ファイルの

コピー時にエラーが生じると、サブスクリプション・セット全体のレプリケーションが失敗します。サブスクリプション・セットが失敗した場合、アプライ・プログラムはサブスクリプション・セットを非活性化せずに、次のアプライ・サイクルでサブスクリプション・セットを再度処理します。

UNIX および Windows の場合: `loadxit` パラメーターを `y` に設定して `ASNLOAD` 出口ルーチン呼び出しして、アプライ・プログラムを始動してください。`ASNLOAD` 出口ルーチンはフル・リフレッシュ時に外部ファイル (`DATALINK` の値が指し示す) をコピーします。詳細については、493 ページの『`ASNLOAD` 出口ルーチンを使ったターゲット表のリフレッシュ』を参照してください。

OS/400 の場合: アプライ・プログラムがフル・リフレッシュ時に外部ファイルをコピーできるように、`ASNDLCP` 出口ルーチン呼び出すように `ASNLOAD` 出口ルーチンを変更してください。詳細については、493 ページの『`ASNLOAD` 出口ルーチンを使ったターゲット表のリフレッシュ』を参照してください。

重要: 外部ファイルは非常に大きい場合があるため、アプライ・プログラムとファイル転送メカニズムの両方について、これらのファイルをコピーするために使用する十分なネットワーク帯域幅を確保してください。同じように、ターゲット・システムにもこれらのファイルが入るだけのディスク・スペースを確保する必要があります。

制約事項:

- OS/400 上の DB2 データベースと他のオペレーティング・システム上の DB2 データベースの間で `DATALINK` 列を複製することはできません。
- OS/400 オペレーティング・システムの場合、`DATALINK` の値の "comment" 属性に対するサポートはありません。
- `DATALINK` 列で `update-anywhere` レプリケーションを使用する場合は、競合検出レベルに **None** を指定して、`DATALINK` 列、および同じサブスクリプション・セット内のその他の列の両方について、競合検出をオフにする必要があります。DB2 レプリケーションは、`DATALINK` 列が参照する外部ファイルの更新競合はチェックしません。
- `DATALINK` 列における変更前イメージ値はサポートしていません。
- ターゲット表が基礎集約表または変更集約表である場合は、`DATALINK` 列がサポートできません。
- 整合変更データ (CCD) 表の中のデータを複製するときには、以下の制約事項が適用されます。
 - 内部 CCD 表は、`DATALINK` 標識 (関連付けられた URL に関する情報を含む `VARCHAR` タイプの文字ストリング) を含むことはできますが、`DATALINK` の値を含むことはできません。アプライ・プログラムは、これらの表タイプでデータを複製するときには、`ASNDLCOPY` 出口ルーチン呼び出しません。
 - コンデンスされた外部 CCD 表は `DATALINK` 列を含むことができます。

- 非コンデンス CCD ターゲット表は DATALINK 列を含むことができません。

以下のセクションでは、DATALINK の値および URL がポイントする外部ファイルの両方をターゲット・システムに複製するためにアプライ・プログラムが使用する、ユーザー出力ルーチンおよびファイル・コピー・デーモン (オペレーティング・システムによって異なる) について説明します。

- 『ASNDLCOPY 出力ルーチンの設定と使用方法』
- 610 ページの『DLFM_ASNCOPYD の設定と使用方法 (UNIX、Windows)』
- 611 ページの『ASNDLCOPYD の設定と使用方法 (OS/400)』

ASNDLCOPY 出力ルーチンの設定と使用方法:

サブスクリプション・セットのレプリケーションの準備ができると、アプライ・プログラムは変更データ (CD) 表の中の対象の行を識別します。DATALINK 列の値が検出された場合、アプライ・プログラムは更新されたファイルの URL を入力ファイルに入れます。アプライ・プログラムは続いて ASNDLCOPY 出力ルーチンと呼び出します。この出力ルーチンはこの入力ファイルを読み取り、DATALINK ソース・ファイルの各ロケーションを対応するターゲット・ファイル・ロケーションにマップします。次に、ASNDLCOPY 出力ルーチンはファイル・コピー・デーモンに接続し、URL がポイントする外部ファイルを、ソース・ファイル・システムから新しくマップされたターゲット・ファイル・システム・ロケーションにマップします。

ASNDLCOPY ルーチンが完了した時点で、このルーチンはアプライ・プログラムに戻りコードを渡します。ゼロ以外の戻りコードは、1 つまたは複数のファイルでレプリケーションが失敗したことをアプライ・プログラムに知らせます。このような場合、アプライ・プログラムはメッセージを出し、そのサブスクリプションをスキップして次のサブスクリプションを処理します。ゼロの戻りコードは、レプリケーションが正常に行われたことをアプライ・プログラムに知らせます。

ASNDLCOPY 出力ルーチンのソース・コードを使用して、システムの要件に合わせてサンプル・プログラム (ASNDLCOPY.smp と呼ばれ、¥sqllib¥samples¥rep1 ディレクトリーに置かれている) を変更できます。サンプル・プログラムには、以下の構成ファイルが含まれます。

ASNDLSRVMAP

ソース URL をターゲット URL にマップします。

例: `http://source.com/file` to `http://target.com/file`

ASNDLUSER

ソースおよびターゲットのファイル・システムに接続するときを使用される、ログオンおよびアドレス・ロケーション情報が入っています。

ASNDLPARM

ASNDLCOPY 出口ルーチンの関数をコントロールするために使用される稼働パラメーターが入っています。これらのパラメーターには、ソース・ファイルを異なるターゲット・ファイル・ロケーションに複製するために使用される **REPLACE_FILE** パラメーター、そして複製するファイルの最後の変更時刻を保存するために使用される **PRESERVE_MODTIME** パラメーターが含まれます。ASNDLPARM は、UNIX および Windows オペレーティング・システムでのみ使用される、オプションの構成ファイルです。

外部ファイルを複製するために、ユーザー自身の出口ルーチンを構成することができますが、プログラムの名前は ASNDLCOPY にする必要があります。構成ファイルは、アプライ・プログラムの現行実行パスに入れてください。

構成ファイルのセットアップ方法、およびこの出口ルーチンの変更方法については、`¥sqllib¥samples¥rep1` ディレクトリーにあるサンプル・プログラムの PROLOG セクションを参照してください。

手順:

ASNDLCOPY 出口ルーチンを使用するには:

1. サイトの要件に合わせて、必要であれば ASNDLCOPY ルーチンを変更します。
アプライ・プログラムのトレース・オプションをオンにすると、ASNDLCOPY ルーチンはログ・ファイルとトレース・ファイルという 2 つのファイルを作成します。このログ・ファイルの名前は以下のとおりです。

`ASNDLApplyQualSetNameSrcSrvrTgtSrvr.LOG`

`ApplyQual` はアプライ修飾子、`SetName` はサブスクリプション・セットの名前、`SrcSrvr` はソース・サーバーの名前、そして `TgtSrvr` はターゲット・サーバーの名前です。ログ・ファイルには、ASNDLCOPY ルーチンが生成するすべてのメッセージが入れられます。トレース・ファイルの名前は以下のとおりです。

`ASNDLApplyQualSetNameSrcSrvrTgtSrvr.TRC`

トレース・ファイルには、ASNDLCOPY ルーチンが生成するトレース情報がすべて入れられます。

2. 必要に応じて ASNDLUSER、ASNDLSRVMAP、および ASNDLPARM 構成ファイルを構成します。

UNIX および Windows の場合: ASNDLPARM ファイルの中で **REPLACE_FILE** パラメーターを YES (デフォルト) に設定した場合、ターゲット・ディレクトリーにターゲット・ファイルがすでに存在するときには、ASNDLCOPY 出口ルーチンはソース・ファイルの内容を別のターゲット・システム・ファイルに複製します。ASNDLCOPY 出口ルーチンは、ソース・ファイルの内容を一時ファイルに直接コピーします。この一時ファイルの名前は、ソース・ファイルの名前と同じですが、“new ” という接尾部が付けられます。(この接尾部は ASNDLPARM ファイルの中で変更できます。)アプライ・プログラムは次に、結果ファイルからオリジナルのター

ゲット・ファイルの URL と一時ファイルの URL を受け取ります。アプライ・プログラムが変更をターゲット表に伝搬すると、DB2 は、レプリケーション・トランザクションをコミットするときに、一時ファイルの名前を、オリジナルのターゲットの URL 中のファイル名に変更します。

3. ASNDLCOPY 出口ルーチンを変更した場合には、プログラムをコンパイルし、実行可能プログラムを適切なディレクトリーに入れてください。

アプライ・プログラムは ASNDONE 出口ルーチンを、正常に処理されたかエラーが発生したかに関係なく、サブスクリプションの処理が完了した後に呼び出します。そのため、ASNDLCOPY ルーチンが外部ファイルのレプリケーションに失敗した場合は、ASNDONE 出口ルーチンを使って必要なクリーンアップを任意に実行できます。

DLFM_ASNCOPYD の設定と使用方法 (UNIX、Windows): DB2 Data Links Manager バージョン 8 をインストールしてある場合は、Data Links Manager レプリケーション・デーモン (DLFM_ASNCOPYD) を使用して、DATALINK データ・タイプから参照されるファイルをコピーできます。

ASNDLCOPY 出口ルーチンはソースおよびターゲットの URL をマップした後、ファイルをコピーするためのデーモンに接続します。使用するファイル・コピー・デーモンに接続するようにアドレスとポート番号を指定して、ASNDLUSER 構成ファイルを構成できます。任意の FTP デーモン、または DLFM_ASNCOPYD ファイル・コピー・デーモンを使用できます。

FTP デーモンも DLFM_ASNCOPYD デーモンもソース・ファイル・システムからターゲット・ファイル・システムへ外部ファイルをコピーします。しかし、DLFM_ASNCOPYD ファイル・コピー・デーモンには追加の関数があります。

- RECOVERY YES と定義された DATALINK 列から参照される特定のバージョンのファイルを検索する機能。
- ユーザーのアクセス権に基づいて、READ PERMISSION DB と定義された DATALINK 列から参照されるファイルを検索する機能。
- 複製されたファイルの最終変更時刻を保存する機能。

DLFM_ASNCOPYD の制約事項:

複製したファイルを DLFM_ASNCOPYD を使用してコピーするには、DB2 Universal Database バージョン 8 で、DB2 Data Links Manager バージョン 8 を使用する必要があります。

DLFM_ASNCOPYD ファイル・コピー・デーモンは、以下のオペレーティング・システムでしか使用できません。AIX、Solaris™ 実行環境版、Windows NT、および Windows 2000。

FTP の場合の Solaris™ 実行環境版の制約事項:

Solaris™ 実行環境版で DATALINK 列の値を複製する場合、FTP デーモンを使用してファイルをコピーするには、MDTM (modtime) コマンドをサポートする FTP デーモンを使用する必要があります。ソースおよびターゲットのファイル・システムで実行される FTP デーモンは、特定のファイルの最終変更時刻を表示する、MDTM をサポートする必要があります。Solaris™ 実行環境版のバージョン 2.6、または FTP による MDTM サポートを含まないその他のバージョンを使用する場合は、WU-FTPD などの追加ソフトウェアが必要です。

手順:

DLFM_ASNCOPYD ファイル・コピー・デーモンを設定するには:

1. このファイル・コピー・デーモンへの接続を必要とするユーザーを識別します。
2. これらのファイルが置かれているディレクトリーに基づいて、ファイルへのアクセス権をユーザーに付与します。
3. DLFM_ASNCOPYD デーモンが使用可能であり、正しいポート番号が指定されていることを確認します。

このポート番号は、ASNDLUSER 構成ファイルで指定されているポート番号と一致する必要があります。

詳細については、*DB2 Data Links Manager Quick Beginnings* および *DB2 Data Links Manager Administration Guide and Reference* を参照してください。

データ・リンク・ファイル・マネージャーは、標準の SQL 操作によってアプリケーションがファイルにリンクするたびに、RECOVERY YES として定義された DATALINK 列で、ソース・ファイルの新しいバージョンをアーカイブします。キャプチャー・プログラムは、RECOVERY YES として定義された DATALINK 列で行の変更をキャプチャーすると、ファイルのバージョン番号を記録し、このバージョン番号を CD 表に入れます。アプライ・プログラムは、CD 表でデータの変更とバージョン番号を読み取り、新しい DATALINK 列の値の URL とバージョン番号を ASNDLCOPY 出口ルーチンに渡します。ASNDLCOPY 出口ルーチンが DLFM_ASNCOPYD デーモンに接続すると、このファイル・コピー・デーモンは外部ファイルの一貫性のあるバージョンをリトリブします。

ソース・システム上にファイルのより新しいバージョンが存在している場合も、データ・リンク・ファイル・マネージャーは、CD 表に取り込まれているバージョンと整合するバージョンのファイルを提供します。このため、ターゲット・サーバーは、キャプチャー・プログラムがまだログ内に取り込んでいないバージョンを受け取ることはできません。

ASNDLCOPYD の設定と使用方法 (OS/400): ASNDLCOPYD は、ASNDLCOPY 出口ルーチンがソースとターゲットの URL をマップした後で、許可ユーザーが OS/400 ソース・サーバーから OS/400 ターゲット・サーバーにファイルをリトリブできるようにするデーモンです。ASNDLCOPY 出口ルーチンはソースおよびターゲットの URL

をマップした後、ファイルをリトリートするために ASNDLCOPYD デーモンに接続します。ASNDLCOPYD ファイル・コピー・デーモンは FTP デーモンと似ていますが、DATALINK の値を複製するために以下の関数を提供します。

- ファイル情報 (ファイル・サイズや最終変更時刻など) を抽出するためのコマンド
- 特定ファイルの中身を取り出すコマンド

READ PERMISSION DB として定義されている DATALINK 列を複製するために、ASNDLCOPYD ファイル・コピー・デーモンに接続するように ASNDLCOPY 出口ルーチンを構成できます。

ASNDLCOPYD のサンプル・ファイルはライブラリー QDP4、ソース・ファイル QCSRC、メンバー ASNDLCPD に入っています。このサンプル・ファイルは 3 つのプログラムを構築します。

ASNDLCOPYD

メインの親プログラムとファイル・コピー・デーモン。

ASNCHILD

クライアントから ASNDLCOPYD デーモンへの接続を調整するプログラム。

ASNCHILD は ASNDLCOPYD デーモンの一部です。クライアントからの要求のたびに、新しい ASNCHILD プロセスが作成されます。

ASNDLCFG

ユーザー ID の追加と除去、そしてユーザー ID のパスワードの変更のための構成プログラム。

注: OS/400 またはその他のオペレーティング・システムで、現在 DB2 バージョン 7 上で ASNDLCOPYD ファイル・コピー・デーモンを使用している場合は、DB2 バージョン 8 で引き続きこのデーモンを使用できます。

前提条件:

ASNDLCOPYD デーモンを実行するにはルート (管理者) 権限が必要です。

手順:

ASNDLCOPYD ファイル・コピー・デーモンを使用するには:

1. ライブラリー QDP4、ソース・ファイル QCSRC、メンバー ASNDLCPD の ASNDLCOPYD サンプル・プログラムにアクセスします。
2. サイトの要件に合わせてサンプル・プログラムを変更します。
3. プログラム・デーモンを構築します。
 - a. 基本モジュールを構築します。

```
CRTCMOD MODULE(libraryname/ASNDLCPD) SRCFILE(QDP4/QCSRC)
                DBGVIEW(*SOURCE) SYSIFCOPT(*ALL)
```


- b. 子プログラム (ASNCHILD) を構築します。

```
CRTPGM PGM(libraryname/ASNCHILD) MODULE(libraryname/ASNDLCPD)
```

- c. 親プログラム (ASNDLCOPYD) を構築します。

```
CRTPGM PGM(libraryname/ASNDLCOPYD) MODULE(libraryname/ASNDLCPD)
```

- d. 構成プログラム (ASNDLCFG) を構築します。

```
CRTPGM PGM(libraryname/ASNDLCFG) MODULE(libraryname/ASNDLCPD)
```

この *libraryname* は既存のライブラリー名です。詳細については、サンプル・プログラムの PROLOG セクションを参照してください。

4. 実行可能プログラムを QDP4 ライブラリーに入れます。
5. サイトの要件に合わせて構成ファイルを変更します。
6. 管理者権限およびスーパーユーザー・アクセスを使用して ASNDLCOPYD デーモンを始動します。ポート番号、および構成ファイルを含むディレクトリーの両方を指定してください。

ASNDLCOPYD ファイル・コピー・デーモンは、ASNDLCOPYD プログラムが生成するすべてのメッセージのためのログ・ファイルを作成します。このログ・ファイルの名前は ASNDLCOPYDYYYYMMDDHHMMSS.LOG (YYYYMMDDHHMMSS は、デーモンの実行開始時刻) になります。

OS/400 の場合、DB2 レプリケーションは必ず、DATALINK 列の値から参照される、最新のバージョンの外部ファイルをレプリケーションします。

関連タスク:

- 478 ページの『アプライ・プログラムの操作』

レプリケーション・プログラムの操作 (z/OS)

この章は、以下のセクションで構成されています。

- 『レプリケーション・プログラムの操作に JCL またはシステム開始タスクを使用する方法 (z/OS)』
- 615 ページの『レプリケーション・プログラムを自動的に再始動するために MVS 自動リスタート・マネージャー (ARM) を使用する方法 (z/OS)』
- 617 ページの『データ共有モードへのレプリケーション環境の移行 (z/OS)』

レプリケーション・プログラムの操作に JCL またはシステム開始タスクを使用する方法 (z/OS)

z/OS では、JCL を使用するか、システム開始タスクとして、キャプチャー・プログラム、アプライ・プログラムおよびレプリケーション・アラート・モニターを操作できます。

JCL を使用したレプリケーション・プログラムの操作: このセクションでは、JCL を使用して、キャプチャー・プログラム、アプライ・プログラムおよびレプリケーション・アラート・モニターを操作する方法を説明します。

サンプルの JCL はプログラム・ディレクトリーにあります。

JCL を使用して z/OS でキャプチャー・プログラムを始動する方法

1. ライブラリー SASNLJCL(ASNL2RN#) にある JCL をカスタマイズします。この # は、DB2 for z/OS のレベルです (たとえば、バージョン 7 の場合は、# を 7 という数表示で置き換えます)。キャプチャー・ジョブの PARM フィールドに、適切なオプションの呼び出しパラメーターを指定することにより、z/OS 用の JCL を準備します。バッチ・モードで実行する場合は、JCL で時間帯および言語環境変数も設定できます。

次に示す呼び出し JCL の行の例では、TZ および LANG 変数を設定しています。

```
//CAPJFA EXEC PGM=ASNCAP, PARM='ENVAR('TZ=PST8PDT','LANG=en_US')/  
DSN6 cold capture_schema=JFA autostop'
```

2. TSO または MVS コンソールから JCL をサブミットします。

JCL を使用して z/OS 上でアプライ・プログラムを始動する方法

アプライ・ジョブの PARM フィールドに適切な呼び出しパラメーターを指定することにより、z/OS 用の JCL を準備します。サイトの要件に合うように、JCL を調整します。ライブラリー SASNAJCL(ASNA2RN#) にある呼び出し JCL は、Apply for z/OS 製品に含まれています。この # は、DB2 for z/OS のレベルです (たとえば、バージョン 7 の場合は、# を 7 という数表示で置き換えます)。

呼び出し JCL でのこの行の例を次に示します。

```
//apyasn EXEC PGM=ASNAPPLY,PARM='control_server=CTLDB1  
apply_qual=myqual spillfile=disk'
```

JCL を使用して z/OS 上でレプリケーション・アラート・モニターを開始する方法

レプリケーション・アラート・モニター・ジョブの PARM フィールドで適切な呼び出しパラメーターを指定して、z/OS 用の JCL を準備します。サイトの要件に合うように、JCL を調整します。ライブラリー SASNMJCL(ASNMON#) にある呼び出し JCL は、レプリケーション・アラート・モニター (z/OS 版) 製品に含まれています。

呼び出し JCL でのこの行の例を次に示します。

```
//monasn EXEC PGM=ASNMON,PARM='monitor_server=MONDB1  
monitor_qualifier=monqual'
```

JCL を使用して z/OS 上の開始済みプログラムを変更する方法

キャプチャー・プログラム、アプライ・プログラム、またはレプリケーション・アラート・モニターを開始した後で、MODIFY コマンドを使用して、プログラムを停止したり、関連するタスクを実行することができます。MODIFY コマンドは MVS コンソールから実行する必要があります。以下の構文例に示すように、F という省略形を使用できます。

▶—F—*jobname*—, —| パラメーター |————▶

基本的に、F *jobname* , は、 **asnacmd**、**asnccmd**、または **asnmcmd** など、実際のコマンド名を置き換えます。たとえば、キャプチャー・プログラムを停止するには、以下のコマンドを使用します。

```
F capjfa,stop
```

MODIFY の情報については、「*z/OS MVS System Commands*」(SA22-7627) を参照してください。

システム開始タスクを使用してレプリケーション・プログラムを操作する方法

： このセクションでは、システム開始タスクを使用して、キャプチャー・プログラム、アプライ・プログラムおよびレプリケーション・アラート・モニターを操作する方法を説明します。

Capture for z/OS プログラムをシステム開始タスクとして開始するためのセットアップ

1. PROCLIB にプロシージャ (*procname*) を作成します。このプロシージャには、キャプチャー・プログラムを実行するのに必要な JCL (たとえば、SASNLJCL ライブラリーの ASNL2RN6、ASNL2RN7、または ASNL2RN8) が入っています。
2. *procname* について、RACF STARTED クラス内にエントリーを作成します。このエントリーは、*procname* を、キャプチャー・プログラムを始動するために使用される RACF ユーザー ID と関連付けます。キャプチャー・プログラムを始動する前に、このユーザー ID に必要な DB2 許可が付与されるようにしてください。
3. MVS システム・コンソールから、**start *procname*** を実行します。

レプリケーション・プログラムを自動的に再始動するために MVS 自動リスタート・マネージャー (ARM) を使用する方法 (z/OS)

キャプチャー・プログラム、アプライ・プログラムおよびレプリケーション・アラート・モニターと一緒に、自動リスタート・マネージャー (ARM) を使用できます。ARM は、特定のバッチ・ジョブまたは開始タスクの可用性を改善するための MVS リカバー関数です。ジョブまたはタスクが失敗するか、ジョブやタスクを実行しているシステムに障害が発生した場合、ARM はオペレーターの介入なしに、ジョブまたはタスクを再始動できます。ARM は、処理対象のアプリケーションを識別するためにエレメント名を使用します。MVS ARM が使用可能な各アプリケーションは、自分自身についてユニークなエレメント名を生成し、ARM とのすべての連絡にこの名前を使用します。ARM はエレメント名をトラッキングし、エレメント名に対して再始動ポリシーを定義しま

す。ARM の設定についての詳細は、「z/OS MVS プログラミング: シスプレックス・サービス・ガイド」(SA88-8583) を参照してください。

前提条件:

ARM がインストール済みであり、レプリケーション・プログラムが正しく設定されていることを確認してください。レプリケーション・プログラムと一緒に ARM を使用する場合は、レプリケーション・プログラムが APF 許可であることを確認してください。たとえば、アプライ・プログラムまたはレプリケーション・アラート・モニターで ARM を使用する場合は、適切なロード・モジュールを APF 許可ライブラリーにコピーする必要があります。(キャプチャー・プログラムは、ARM を使用するかしないかに関係なく、必ず APF 許可にする必要があります。)

ARM を構成するときには、レプリケーション・プログラムに以下のエレメント名を使用してください。

キャプチャー・プログラム

ASNTCxxxxyyyy

アプライ・プログラム

ASNTAxxxxyyyy

レプリケーション・アラート・モニター

ASNAMxxxxyyyy

この xxxx は DB2 サブシステム名であり、yyyy はデータ共有メンバー名です (後者はデータ共有構成の場合にのみ必要)。エレメント名の長さは常に 16 文字であり、ブランクが埋め込まれます。エレメント名は、シスプレックス全体を通じてユニークのものにする必要があります。このため、ARM を使用するには、1 つのサブシステムにつき、特定のプログラムのインスタンスは 1 つしか実行できません。

レプリケーション・プログラムは初期化時にエレメント名を使用して ARM に登録されます。登録時にイベント出口を ARM に提供することはありません。(レプリケーション・プログラムは MVS サブシステムとして実行されるのではないため、イベント出口は必要ありません。) 登録済みプログラムが異常終了した場合 (たとえばセグメント違反の発生)、ARM はユーザーの代わりに登録済みプログラムを再始動します。登録されたレプリケーション・プログラムは、通常終了した場合 (たとえば、STOP コマンドによる終了)、または無効な登録を検出した場合は、登録解除されます。

ヒント: パラメーター NOTERM=Y を指定してキャプチャー・プログラムまたはアプライ・プログラムを始動すると、DB2 が静止してもプログラムは停止しません。この場合、プログラムは ARM から登録解除されません。プログラムは実行を続けますが、DB2 が再始動されるまでは、データをキャプチャーしません。

データ共有モードへのレプリケーション環境の移行 (z/OS)

キャプチャー・プログラムが非データ共有モードで実行されているときに、システムをデータ共有モードに移行するときには、**ASNPLXFY** を実行して、システムが Sysplex で実行されるように、一度準備する必要があります。キャプチャー・プログラムが正しい LRSN から開始するように、キャプチャー・プログラムをウォーム・スタートする前に、このユーティリティを実行してください。このユーティリティは、再始動 (IBMSNAP_RESTART) 表の中のデータを移行します。非データ共有のログ・シーケンス番号 (RBA) は、データ共有環境での同等のシーケンス番号 (LRSN) に変換されます。

前提条件:

キャプチャー・プログラムを実行するために使用したのと同じユーザー ID を使用するか、同じ特権を持つものを使用します。**ASNPLXFY** ユーティリティが APF 許可であることを確認してください。ASNPLXFY プランは、サブシステムにバインドする必要があります。また、サブシステムはデータ共有モードで実行されている必要があります。このユーティリティのバインディングの詳細は、プログラム・ディレクトリーを参照してください。

手順:

USS データ共有環境で **ASNPLXFY** ユーティリティを実行するには、以下のようになります。

1. キャプチャー・プログラムを停止してください。
2. コマンド行から以下のコマンドを入力します。

```
ASNPLXFY yoursystem captureschema
```

サブシステムの名前が必要です。キャプチャー・スキーマはオプションです。デフォルトのキャプチャー・スキーマは ASN です。

3. キャプチャー・プログラムをウォーム・スタートします。

レプリケーション環境の保守

この章では、データベース内にあり、DB2 レプリケーションに使用される、ソース・システム、コントロール表、およびターゲット表の保守の方法を説明します。

DB2 レプリケーションはデータベース・システムと共同で処理を行うため、既存のデータベース・アクティビティーの変更は最小限ですみます。しかし、システム全体の円滑な実行を保証し、潜在的な問題を回避するためには、レプリケーション環境の処理要件を判別し、これらの要件がデータベース・システムに影響を与える可能性を判別する必要があります。この章では、DB2 レプリケーションの 3 つの関数コンポーネントの保守の要件について説明します。

- 618 ページの『ソース・システムの保守』
- 623 ページの『コントロール表の保守』

- 632 ページの『ターゲット表の保守』

ソース・システムの保守

レプリケーション・ソース・システムは、変更キャプチャー・メカニズム、レプリケーションするソース表 (OS/400 システムで使用されるリモート・ジャーナルを含む)、キャプチャー・プログラムから使用されるログ・データ、および DB2 以外のリレーショナル・データベース・ソースで使用されるキャプチャー・トリガーで構成されます。このセクションでは、ソース表とログ・ファイルを正しく保守する方法、そしてこれらの表およびファイルが常に DB2 レプリケーションからアクセス可能であるようにする方法を説明します。

ソース・オブジェクトの保守: レプリケーション・ソース・オブジェクトは、システム上の他のデータベース表およびビューと同じ保守を必要とする、データベース表およびビューです。これらのオブジェクトに関して、既存のユーティリティおよび保守ルーチンを引き続き実行してください。

キャプチャー・プログラムおよびアプライ・プログラムが常に処理を進められるように、DB2 レプリケーションでのこれらのソース表の可用性を考慮する必要があります。DB2 レプリケーションは、ほとんどのレプリケーション処理においては、ソース表に直接アクセスする必要はありません。しかし、DB2 レプリケーションは、以下の 2 つのアクションのいずれかが発生したときには、ソース表または表スペースにアクセスする必要があります。

- アプライ・プログラムがフル・リフレッシュを実行したとき。
- ロギング・マネージャーが圧縮されたログ・レコードを読み取ろうとしたとき (z/OS のみ)。

フル・リフレッシュ時にレプリケーション・アプライ・プログラムの処理が阻害されないように、ソース表に対して読み取りアクセスが可能であることを確認してください。また、z/OS の場合は、ソース表が圧縮されている場合に、DB2 が圧縮されたログ・レコードの表スペースに対してラッチを入手できるように、ユーティリティがオンライン・モードで実行されることを確認してください。ユーティリティおよび保守ルーチンが、データベース (または z/OS の場合は圧縮された表) をオフラインにする必要のある排他モードで実行される場合は、レプリケーションでソース・オブジェクトを使用することはできません。

ソース・ログおよびジャーナル・レシーバーの保守および保存: DB2 リカバリー・ログには、DB2 リカバリー機能の提供と、実行中のキャプチャー・プログラムへの情報の提供という 2 つの目的があります。DB2 リカバリー、および DB2 レプリケーションの両方についてログ・データを保存する必要があります。また、このデータを削除する前に、キャプチャー・プログラムおよび DB2 が、ログまたはジャーナル・レシーバーのセットの処理を完全に終了していることを確実に確認する必要があります。

注: DB2 レプリケーションは、DB2 以外のリレーショナル・データベースからのログ・データは使用しません。

ログ・データの保存 (UNIX、Windows、z/OS): ログ・データはログ・バッファ、アクティブ・ログ、またはアーカイブ・ログに入っています。キャプチャー・プログラムはウォーム・スタートのたびに、プログラムの停止後に作成されたすべての DB2 ログと、処理が完全に終了していないすべての DB2 ログを要求します。

UNIX および Windows の場合: キャプチャー・プログラムがアーカイブ・ログからデータをリトリブできるように、ユーザー出口アーカイブを使用するようにデータベースを構成する必要があります。

DB2 の実行時は常にキャプチャー・プログラムを実行するようしておけば、キャプチャー・プログラムは一般的に DB2 のリカバー・ログに合わせた最新の状態になります。DB2 がアクティブになっている間は常にキャプチャー・プログラムを実行するか、ログ・レコードを 1 週間以上保存するのであれば、既存のログ保存手順を使用し続けることができます。しかし、以下の場合には、DB2 レプリケーションに合わせて、ログ保存手順を変更する必要があります。

- DB2 がバックアップを完了したら即時にログ・レコードを削除する場合、そしてこれらのログ・レコードが順方向リカバーに必要とされない場合。
- ストレージの制約があるため、アーカイブしたりカバー・ログを頻繁に削除する必要がある場合。

手順:

キャプチャー・プログラムで使用するために保存する必要のあるログ・レコードと、削除できるログ・レコードを判別するには、以下のようになります。

UNIX および Windows の場合:

1. 以下の SQL ステートメントを実行し、再始動 (IBMSNAP_RESTART) 表から MIN_INFLIGHTSEQ 値を入手します。

```
SELECT MIN_INFLIGHTSEQ  
FROM ASN.IBMSNAP_RESTART  
WITH UR;
```

MIN_INFLIGHTSEQ 値が表示されます。(IBMSNAP_RESTART 表には行が 1 つしかありません。) MIN_INFLIGHTSEQ 値は、ビット・データ列の場合は char(10) であり、16 進の 20 文字のように見えます。たとえば、以下のようになります。

```
00000000123456123456
```

MIN_INFLIGHTSEQ 値の最後の 12 文字を見てください。この例では以下のようになっています。

```
123456123456
```


2. コマンド行から **db2 get db cfg** コマンドを入力し、アクティブ・ログ・ファイルのパスを入手します。たとえば、以下のようになります。

```
db2 get db cfg for yourdbname
```

この *yourdbname* はデータベース名です。画面に表示された出力からアクティブ・ログ・ファイルのパスを確認します。たとえば、以下のようになります。

```
Path to log files    =C:¥DB2¥NODE0000¥SQL00001¥SQLLOGDIR¥
```

3. DB2 コマンド行から **db2flsn** コマンドを入力し、MIN_INFLIGHTSEQ 値の最後の 12 文字を入力します。たとえば、以下のようになります。

```
C:¥DB2¥NODE0000¥SQL00001¥>db2flsn 123456123456
```

db2flsn コマンドを実行するには、SQLLOGCTL.LFH ファイルにアクセスできる必要があります。このファイルは、アクティブ・ログ・ファイルのパスの 1 つ上のディレクトリー (C:¥DB2¥NODE0000¥SQL00001¥) にあります。

システムは、ログ・シーケンス番号により識別されるログ・レコードを含むファイルの名前をリトリブして表示します。たとえば、以下のようになります。

```
Given LSN is contained in the log file S000123.LOG
```

4. リトリブされたログ・ファイルの経過日数を確認してください。
キャプチャー・プログラムは、任意の時点から再始動を実行するために、このログ・ファイル、およびこれより新しいログ・ファイルを必要とします。キャプチャー・プログラムの連続稼働を保証するために、このログ・ファイル、およびこれより新しいログ・ファイルは保存する必要がありますが、これより古いログは削除できます。

z/OS の場合、

1. 以下の SQL ステートメントを実行し、再始動 (IBMSNAP_RESTART) 表から MIN_INFLIGHTSEQ 値を入手します。

```
SELECT MIN_INFLIGHTSEQ  
FROM ASN.IBMSNAP_RESTART  
WITH UR;
```

MIN_INFLIGHTSEQ 値が表示されます。(IBMSNAP_RESTART 表には行が 1 つしかありません。) たとえば、以下のようになります。

```
0000555551F031230000
```


常に 0000 である、最初の 4 文字は無視してください。以下の 12 文字は、アクティブ・ログ・シーケンス番号に相当します。(この 12 文字の値は、非データ共有環境では相対バイト・アドレス [RBA] であり、データ共有環境ではログ・レコード・シーケンス番号 [LRSN] です。)最後の 4 文字は、非データ共有環境では 0000 になります。これら最後の 4 文字は、データ共有環境ではメンバー ID に相当します。

2. DSNJU004 ユーティリティを使用してログ・マップ印刷ユーティリティを呼び出します。このユーティリティは、ブートストラップ・データ・セット (BSDS) に関する情報を表示します。

たとえば、以下のようになります。

```
# ACTIVE LOG COPY 1 DATA SETS
# START RBA/TIME      END RBA/TIME          DATE   LTIME   DATA SET INFORMATION
#-----
# 555551F03000        555551F05FFF         1998.321 12:48 DSN=DSNC710.LOGCOPY1.DS02
#2001.57 15:46:32.2  2001.057 15:47:03.9  PASSWORD=(NULL)STATUS=TRUNCATED,REUSABLE
# 555551F06000        555551F09FFF         1998.321 12:49 DSN=DSNC710.LOGCOPY1.DS03
#2001.57 15:47:32.2  2001.057 15:48:12.9  PASSWORD=(NULL)STATUS=TRUNCATED,REUSABLE
```

3. MIN_INFLIGHTSEQ 値の 12 文字のアクティブ・ログ番号を、表示された各行の開始 RBA および対応する終了 RBA の範囲と比較します。
4. 自分の 12 文字のアクティブ・ログ番号の値が当てはまる行を探します。この例では以下のようになっています。

```
# 555551F03000          000001F05FFF         1998.321 12:48 DSN=DSNC710.LOGCOPY1.DS02
#2001.57 15:46:32.2  2001.057 15:47:03.9  PASSWORD=(NULL)STATUS=TRUNCATED,REUSABLE
```

5. このアクティブ・ログ番号に対応するデータ・セット情報を確認します。この例では以下のようになっています。

```
DSNC710.LOGCOPY1.DS02
```

6. このデータ・セットの日時を確認します。

キャプチャー・プログラムは、任意の時点から再始動を実行するために、このデータ・セット、およびこれより新しいデータ・セットを必要とします。

このログ・ファイルまたはデータ・セットの経過日数はベンチマークと考えてください。キャプチャー・プログラムの連続稼働を保証するために、このファイル、およびこれより新しいログ・ファイルは保存する必要がありますが、これより古いログは削除できます。

推奨: キャプチャー・プログラムはログ・バッファから直接ログ・レコードを読み取るため、最適なパフォーマンスを実現するために、DB2 がアクティブになっている間は常にキャプチャー・プログラムを実行してください。

ジャーナル・レシーバーの保存 (OS/400): キャプチャー・プログラムから必要とされるすべてのジャーナル・レシーバーを保存することが重要です。RESTART(*YES) パラメーターを指定してキャプチャー・プログラムを再始動すると、キャプチャー・プログラムは以前に終了した場所から処理を続行し、1 つまたは複数のソース表により使用されるすべてのジャーナル・レシーバーを必要とします。

キャプチャー・プログラムが必要なジャーナル・レシーバーのすべてにアクセスできるように、DB2 DataPropagator for iSeries のインストール時に自動的に登録された、ジャーナル・レシーバー削除出口プログラムを使用してください。この出口プログラムは、ユーザー、またはユーザーのアプリケーション・プログラムの 1 つがジャーナル・レシーバーの削除を試みるたびに呼び出されます。この出口プログラムは、ジャーナル・レシーバーを削除できるかどうかを決定します。

推奨: CHGJRN または CRTJRN コマンドで DLTRCV(*YES) および MNGRCV(*SYSTEM) を指定し、ジャーナル・レシーバー削除出口ルーチンを使用し、ジャーナル管理をシステムに任せるようにします。

ジャーナル・レシーバーが 1 つまたは複数のソース表で使用されている場合、ジャーナル・レシーバー削除出口ルーチンは、キャプチャー・プログラムによってまだ処理されていないエントリが、削除対象のレシーバーに含まれていないことを確認します。キャプチャー・プログラムがそのレシーバーのエントリをさらに処理する必要がある場合には、出口プログラムはレシーバーの削除を承認しません。詳細については、684 ページの『ジャーナルおよびジャーナル・レシーバーの管理 (OS/400)』を参照してください。

コンプレッション・ディクショナリーの使用方法 (z/OS): DB2 のコンプレッション・ディクショナリーユーティリティを使用する場合は、キャプチャー・プログラムとの間でこれらのユーティリティの使用を調整する必要があります。

DB2 コンプレッション・ディクショナリーの更新 (z/OS)

キャプチャー・プログラムがログ・レコードを要求した場合、DB2 は圧縮された表スペースに保管されている表のログ・レコードを圧縮解除する必要があります。DB2 は現行のコンプレッション・ディクショナリーを使用して圧縮を解除します。コンプレッション・ディクショナリーが一時的に使用できない場合、DB2 はキャプチャー・プログラムにエラーを戻します。キャプチャー・プログラムは何度か処理の続行を試みます。しかし、辞書が引き続き使用できない状態である場合は、キャプチャー・プログラムは ASN0011E メッセージを発行して終了します。また、コンプレッション・ディクショナリーがもはや使用できない場合には、キャプチャー・プログラムは登録を非活動化します。これらの状況を防止するために、表のコンプレッション・ディクショナリーに影響を与えるアクティビティを実行する前に、キャプチャー・プログラムが表のすべてのログ・レコードを処理するようにしてください。これらのアクティビティには以下のものが含まれます。

- 圧縮設定を変更する表スペースの変更
- データ共有環境から非データ共有環境へのコピーを含めて、DSN1COPY を使用したサブシステム間での圧縮した表スペースのコピー
- 表スペースに対する REORG ユーティリティの実行

推奨: 再編成時にコンプレッション・ディクショナリーの現行バージョンを維持するには、KEEPDICTIONARY=YES オプションを使用します。KEEPDICTIONARY=YES オプションを使用すると、既存のログ・レコードに対して辞書の互換性を維持できます。

しかし、新しいコンプレッション・ディクショナリーを生成する場合は、現在実行中のアプリケーション、およびキャプチャー・プログラムに対して、REORG ユーティリティを以下のように同期させてください。

1. 表を更新するすべてのアプリケーション・プログラムを静止させる。
2. キャプチャー・プログラムで、ログに保管されている表の更新をすべてキャプチャーする。
3. 圧縮表に対して REORG ユーティリティを使用し、新しいコンプレッション・ディクショナリーを作成する。
4. アプリケーション・プログラムを再始動する。

DB2 コンプレッション・ディクショナリーのラッチ (z/OS)

コンプレッション・ディクショナリーの可用性についても考慮する必要があります。キャプチャー・プログラムが圧縮されたログ・レコードを読み取るときには、DB2 は辞書にアクセスするために、ソースの圧縮された表のラッチを取ります。DB2 のログ読み取りインターフェースがこのラッチを必要としたときに、ソース・システム上の圧縮された表スペースが STOPPED 状態にある場合は、キャプチャー・プログラムは停止します。その逆に、ソースの表スペースへの完全アクセスを必要とする、または表スペースが STOPPED 状態であることを必要とするユーティリティは、キャプチャー・プログラムが辞書の読み取り中にラッチを保持しているために締め出されることがあります。

ラッチが使用できないために一時なロックアウトが発生しないように、ソースの圧縮された表スペースを DB2 (またはベンダー) のユーティリティで排他的に使用する必要があるときには、キャプチャー・プログラムを中断してください。

コントロール表の保守

DB2 レプリケーションは、ソース定義、サブスクリプション・セット定義、およびその他のレプリケーション・ユニークのコントロール情報を保管するためにコントロール表を使用します。コントロール表のサイズは静的なものもあれば、データベースおよびレプリケーションのサイズの要件にしたがって動的に拡大 (または縮小) するものもあります。

以下のコントロール表のサイズは、通常処理時に頻繁に変更されます。

- アプライ・ジョブ (IBMSNAP_APPLY_JOB) (OS/400 のみ)
- アプライ修飾子相互参照 (IBMSNAP_AUTHTKN) (OS/400 のみ)
- アプライ・トレース (IBMSNAP_APPLYTRACE)
- アプライ・トレール (IBMSNAP_APPLYTRAIL)
- キャプチャー・モニター (IBMSNAP_CAPMON)
- キャプチャー・トレース (IBMSNAP_CAPTRACE)
- 変更データ (*schema.CD_table*)
- 整合変更データ (*schema.target_table*)
- レプリケーション・アラート・モニター・アラート (IBMSNAP_ALERTS)
- レプリケーション・アラート・モニター・トレース (IBMSNAP_MONTRACE)
- レプリケーション・アラート・モニター・トレール (IBMSNAP_MONTRAIL)
- シグナル (IBMSNAP_SIGNAL)
- サブスクリプション・イベント (IBMSNAP_SUBS_EVENT)
- 作業単位 (IBMSNAP_UOW)

これらの動的なコントロール表のサイズおよび拡大により、システムのパフォーマンスが影響を受ける可能性があります。

このセクションでは、コントロール表に対して実行する必要がある保守のアクティビティについて説明します。

***RUNSTATS* ユーティリティーの使用法 (UNIX、Windows、z/OS):**

RUNSTATS ユーティリティーは、表および関連する索引の物理的特性に関する統計を更新します。以前に DB2 レプリケーションで使用していたのと同じ頻度で、既存の表に対して引き続き *RUNSTATS* ユーティリティーを実行する必要があります。しかし、表に含まれるデータの量が大量である場合には、変更データ (CD)、作業単位 (IBMSNAP_UOW)、およびその他の動的なコントロール表に対して *RUNSTATS* ユーティリティーを一度だけ実行してください。*RUNSTATS* が動的な表に関して有用な情報を報告するのは、これらの表が実動レベルで最大のサイズにある場合です。最適化プログラムはデータへのアクセスの最良のストラテジーを判断します。

パッケージおよびプランの再バインド (UNIX、Windows、z/OS): DB2 レプリケーションのパッケージおよびプランの多くは、分離 UR (非コミット読み取り) を使用してバインドされます。パッケージおよびプランの自動再バインドに使用される内部の保守プログラムは、キャプチャー・プログラムおよびアプライ・プログラムがカーソル固定などの標準オプションを使用してレプリケーション・パッケージを再バインドした場合に、プログラム間で競合の問題を発生させる可能性がありますので、パッケージお

よびプランの再バインドが必要な場合は注意してください。DB2 レプリケーション・パッケージは、最適なシステム・パフォーマンスを維持するために、分離 UR にバインドしておく必要があります。

詳細については、676 ページの『レプリケーション・プログラムのセットアップ』を参照してください。

コントロール表の再編成: 頻繁に更新される動的なコントロール表は定期的に再編成する必要があります。変更データ (CD) 表および作業単位 (IBMSNAP_UOW) 表は、変更キャプチャー時には多数の INSERTS を受け取り、プルーニング時には多数の DELETES を受け取ります。キャプチャー・モニター (IBMSNAP_CAPMON) 表、キャプチャー・トレース (IBMSNAP_CAPTRACE) 表、およびアプライ・トレール (IBMSNAP_APPLYTRAIL) 表のサイズは、レプリケーションのソース表の更新率によって、大きく変化する可能性があります。

手順:

データのフラグメント化を防止し、スペースをレクラメーション処理するために、以下の表の再編成方式のいずれかを使用してください。

REORG コマンド (UNIX、Windows)

PREFORMAT オプションを指定した REORG ユーティリティ (z/OS)

このユーティリティの PREFORMAT オプションは、キャプチャー・プログラムの挿入処理を迅速化します。

RGZPFM (物理ファイル・メンバーの再編成) コマンド (OS/400)

ENDDPRCAP コマンドで **RGZCTLTBL(*YES)** パラメーターを指定すると、キャプチャー・プログラムが終了したときに UOW 表およびアクティブな CD 表を再編成できます。

推奨: 以下の動的なコントロール表は週に一度は再編成してください。

- CD 表
- IBMSNAP_ALERTS
- IBMSNAP_APPLYTRACE
- IBMSNAP_APPLYTRAIL
- IBMSNAP_CAPMON
- IBMSNAP_CAPTRACE
- IBMSNAP_MONTRAIL
- IBMSNAP_MONTRACE
- IBMSNAP_UOW

静的なコントロール表に対しては、未使用のスペースをレクラメーション処理するためのユーティリティを実行したり、頻繁に更新される最適化プログラムの統計を生成する必要はありません。

- アブライ・エンキュー (IBMSNAP_APPENQ)
- キャプチャー・エンキュー (IBMSNAP_CAPENQ) (UNIX、Windows、z/OS)
- キャプチャー・パラメーター (IBMSNAP_CAPPARMS)
- キャプチャー・スキーマ (IBMSNAP_CAPSCHEMAS)
- プルーニング・ロック (IBMSNAP_PRUNE_LOCK)
- プルーニング・セット (IBMSNAP_PRUNE_SET)
- プルーニング・コントロール (IBMSNAP_PRUNCNTL)
- レジスター (IBMSNAP_REGISTER)
- 登録拡張 (IBMSNAP_REG_EXT) (OS/400 のみ)
- 登録同期 (IBMSNAP_REG_SYNCH)
- レプリケーション・アラート・モニター条件 (IBMSNAP_CONDITIONS)
- レプリケーション・アラート・モニター連絡先 (IBMSNAP_CONTACTS)
- レプリケーション・アラート・モニター連絡先グループ (IBMSNAP_CONTACTGRP)
- レプリケーション・アラート・モニター・エンキュー (IBMSNAP_MONENQ)
- レプリケーション・アラート・モニター・グループ (IBMSNAP_GROUPS)
- レプリケーション・アラート・モニター・サーバー (IBMSNAP_MONSERVERS)
- 再始動 (IBMSNAP_RESTART)
- 順序付け (IBMSNAP_SEQTABLE)
- サブスクリプション列 (IBMSNAP_SUBS_COLS)
- サブスクリプション・メンバー (IBMSNAP_SUBS_MEMBR)
- サブスクリプション・セット (IBMSNAP_SUBS_SET)
- サブスクリプション・ステートメント (IBMSNAP_SUBS_STMTS)

コントロール表のプルーニング: 古いデータを除去し、システム・パフォーマンスを向上させるために、レプリケーション・コントロール表の定期的な整理が必要です。このセクションでは、コントロール表の整理のさまざまな方式と、これらの方式がシステムのパフォーマンスにどのような影響を与えるかについて説明します。

キャプチャー・プログラムによって保守される動的なコントロール表のプルーニング: 以下の動的なコントロール表の拡張をモニターし、使用可能な各種のプルーニング方式を考慮する必要があります。

- CD 表
- IBMSNAP_UOW
- IBMSNAP_CAPMON

- IBMSNAP_CAPTRACE
- IBMSNAP_SIGNAL
- IBMSNAP_AUTHTKN (OS/400 のみ)

これらの表でブルーニングが一定インターバルで自動的に行われるように、キャプチャー・プログラムを設定できます。また、ブルーニング・プロセスを立ち上げることにより、要求時に一度だけブルーニングを行うこともできます。ユーザーが次にブルーニング・コマンドを入力するまで、キャプチャー・プログラムはブルーニングを行いません。

推奨: これらのコントロール表の拡張を管理するために、自動ブルーニングを使用することを考えてみてください。自動ブルーニングにより、ストレージ・コストは最小化され、アプライ・プログラムの効率は向上し、また、これらの表の古いデータが定期的には除去されるため、ストレージのオーバーフローによるシステム障害のリスクが一般的に減少します。自動ブルーニングを呼び出すには以下のようにします。

- キャプチャー・プログラムの **autoprun** パラメーターを **y** に設定します (UNIX、Windows、z/OS)。
- キャプチャー・プログラムのパラメーター設定、**CLNUPITV(*IMMED)** または **CLNUPITV(*DELAYED)** を使用します (OS/400)。

自動ブルーニングでは、**prune_interval** 稼働パラメーター (UNIX、Windows、および z/OS の場合)、または **RETAIN** パラメーター (OS/400 の場合) を設定して、自動ブルーニング処理を発生させる頻度を指定します。

手順:

以下の方法のいずれかを使用してブルーニングを開始します。

レプリケーション・センター

「キャプチャー・コントロール表の整理」ウィンドウを使用して、表の整理を一度行います。詳細については、レプリケーション・センターのヘルプを参照してください。

asncap システム・コマンドで **autoprun=y** を指定 (UNIX、Windows、z/OS)

自動ブルーニングを指定してキャプチャー・プログラムを始動するには、このコマンドを使用します。コマンド構文およびパラメーター記述の詳細については、542 ページの『asncap: キャプチャーの開始 (UNIX、Windows、z/OS)』を参照してください。

asnccmd システム・コマンドで **chgparms autoprun=y** を指定 (UNIX、Windows、z/OS)

実行中のキャプチャー・プログラムで自動ブルーニングを使用可能にするには、このコマンドを使用します。コマンド構文およびパラメーター記述の詳細については、526 ページの『asnccmd: キャプチャーの操作 (UNIX、Windows、z/OS)』を参照してください。

**asnccmd システム・コマンドでブルーニング・パラメーターを使用
(UNIX、Windows、z/OS)**

実行中のキャプチャー・プログラムでブルーニングを一度開始するには、このコマンドを使用します。コマンド構文およびパラメーター記述の詳細については、526 ページの『asnccmd: キャプチャーの操作 (UNIX、Windows、z/OS)』を参照してください。

STRDPRCAP CLNUPITV(*IMMED) または STRDPRCAP CLNUPITV(*DELAYED) システム・コマンド (OS/400)

キャプチャー・プログラムを始動した後で、指定したインターバルで古い行を削除するには、これらのコマンドを使用します。

OVRDPRCAPA PRUNE(*IMMED) または OVRDPRCAPA PRUNE(*DELAYED) システム・コマンド (OS/400)

実行中のキャプチャー・プログラムによるコントロール表の整理の方式を変更するには、このコマンドを使用します。

CD 表および UOW 表のブルーニング: キャプチャー・プログラムは、自動的に呼び出された場合も、要求時に実行する場合も、アプライ・プログラムから報告された進行に基づいて、各ブルーニング・サイクルで CD 表および UOW 表のブルーニングを行います。進行は、ブルーニング・セット (IBMSNAP_PRUNE_SET) 表の SYNCHPOINT 列で示されます。この通常 のブルーニングは、各 CD 表をサブスクライプするすべてのアプライ・プログラムを通じて最小の同期点値に基づいて、また UOW 表の場合は全体を通じて最小の同期点値に基づいて行われます。

しかし通常のブルーニングでは、関連するサブスクリプション・セットが非常にまれにしか実行されない場合は、CD 表および UOW 表の効果的なブルーニングを行うことはできません。関連するアプライ・プログラムの実行頻度を決定するとき、これらのアプライ・プログラムを停止するとき、そしてサブスクリプション・セットを比較的長い間非活動化するときには、ブルーニングの効率性を考慮してください。

サブスクリプション・セットを非常にまれにしか実行しないか、またはアプライ・プログラムを停止すると、使用している CD 表と UOW 表が非常に大きくなって、保持制限ブルーニングの対象として適格となる可能性があります。保持制限は、キャプチャー・プログラムの稼働パラメーターであり、このデフォルト値は 1 週間です。この値は、表の中の古いデータがどれくらいたつと保持制限ブルーニングに適格になるかを決定します。

サブスクリプション・セットが非アクティブになるか、まれにしか実行されないために、通常のブルーニング処理が使用禁止になると、データが非常に長い時間表の中に溜まる可能性があります。このデータが DB2 の現行タイム・スタンプから保持制限値を引いた値よりも古くなると、保持制限ブルーニング処理は、このデータを表からブルーニングします。

保持制限ブルーニングを必要とするような条件の発生は回避してください。古いデータが累積されることにより、ストレージのオーバーフローが発生し、性能低下を招く可能性があります。

推奨: すべてのサブスクリプション・セットに対して、アプライ・プログラムを少なくとも週に一度実行してください。

ソース・サーバーが多様なターゲット・システムに変更データを提供している場合、そして、各ターゲットの要件が大きく異なるものであり、また、ターゲットによっては、少数の登録済みソースに対してアプライ・プログラムがまれにしか実行されない場合は、複数のキャプチャー・プログラムを使用することを考えてみてください。複数のキャプチャー・プログラムを実行することにより、異なるキャプチャー・スキーマを使用して、多様な処理要件を管理できます。1 つのキャプチャー・スキーマを使用して、サブスクリプション・セットのユニークのタイミング要件によりまれにしか整理されない表を分離し、残りのソース表に対しては、別のキャプチャー・スキーマを使用できます。

キャプチャー・モニター表とキャプチャー・トレース表のブルーニング: キャプチャー・プログラムは、キャプチャー・プログラムの以下の稼働パラメーターの値に基づいて、各ブルーニング・サイクルでキャプチャー・モニター (IBMSNAP_CAPMON) 表およびキャプチャー・トレース (IBMSNAP_CAPTRACE) 表のブルーニングを行います。

- 行が IBMSNAP_CAPMON 表の中に維持される時間の長さを示す **monitor_limit** パラメーター (UNIX、 Windows、 z/OS の場合) および **MONLMT** パラメーター (OS/400 の場合)
- 行が IBMSNAP_CAPTRACE 表の中に維持される時間の長さを示す **trace_limit** パラメーター (UNIX、 Windows、 z/OS の場合) および **TRCLMT** パラメーター (OS/400 の場合)

モニター限度パラメーターおよびトレース限度パラメーターのデフォルト値は両方ともに 1 週間です。IBMSNAP_CAPMON 表の中にキャプチャー・プログラムの待ち時間およびスループットの履歴情報をどのくらい長く保持したいか、そして IBMSNAP_CAPTRACE 表の中に監査およびトラブルシューティングの情報をどのくらい長く保持したいかによって、これらの値は変更できます。

シグナル表のブルーニング: シグナル (IBMSNAP_SIGNAL) 表もまた、各ブルーニング・サイクル中にブルーニングされます。シグナル行は、**SIGNAL_STATE** 列の値が C になるとブルーニングに適格になります。C という値は、シグナル情報が完成しており、キャプチャー・プログラム、またはその他のユーザー処理から必要とされることはないため、ブルーニングに適格であることを意味します。**SIGNAL_TIME** 列の値が、DB2 の現行タイム・スタンプから保持制限パラメーターの値を引いたものよりも古いシグナル行は、保持制限ブルーニングに適格です。

その他の動的コントロール表のプルーニング:

キャプチャー・プログラムは、自分が保守している表に対してのみプルーニング操作を実行します。整合変更データ (CCD) 表はアプライ・プログラムによって保守されているため、キャプチャー・プログラムはこれらの表を自動的に整理しません。CCD 表のタイプによっては、プルーニングの必要のないものもあります。完全なコンデンス CCD 表は、同じ場所で更新されます。

完全なコンデンス CCD 表からユーザーが除去できる唯一のレコードは、従属するターゲット表にすでに複製済みで、IBMSNAP_OPERATION 列の値が D (削除) のものです。非コンデンス CCD 表は、履歴データを含むものであり、非常に大きくなる可能性があります。このデータは監査のために保存しておく必要があるため、非コンデンス CCD 表に対してはプルーニング操作を実行しないでください。

しかし、内部 CCD 表のプルーニングは考慮に値します。これらの表は、システムの更新アクティビティが多い場合は、すぐに大きくなる可能性があります。内部 CCD 表からは最新の変更だけがフェッチされるため、古い行を保存しておく必要はありません。

内部 CCD 表のプルーニングを使用可能にするために、従属するすべてのターゲットへのアプライがすでに完了している変更データをプルーニングする、事後に実行される SQL ステートメントを、関連するサブスクリプション・セットに追加することを考えてみてください。また、自動スケジューリング機能に必要な SQL DELETE ステートメントを追加して、これらの表から行を削除することもできます。

アプライ・トレール (IBMSNAP_APPLYTRAIL) 表およびアプライ・トレース (IBMSNAP_APPLYTRACE) 表も手で整理する必要があります。頻繁に実行されるアプライ・プログラムで複数のサブスクリプション・セットを定義および使用する場合、IBMSNAP_APPLYTRAIL 表は急速に拡張するため、頻繁なプルーニングが必要です。これらの表の拡張を管理する最良の方法は、サブスクリプション・セットの 1 つに、事後に実行される SQL ステートメントまたはプロシーチャーを追加することです。また、自動スケジューリング機能に SQL DELETE ステートメントを追加することもできます。

レプリケーションの失敗の防止およびエラーからのリカバー: このセクションでは、コントロール表およびレプリケーション・データに影響を与える可能性のある、レプリケーションの失敗を防止し、失敗からリカバーする方法について説明します。

- キャプチャー・プログラムのコールド・スタートの防止
- コントロール表の入出力エラーおよび接続障害からのリカバー
- 脱落したソース・データのリトリート

キャプチャー・プログラムのコールド・スタートの防止: キャプチャー・プログラムのコールド・スタートは、プログラムを初めて開始するとき、またはコントロール

表およびターゲット表のリフレッシュが必要な場合にだけ実行してください。キャプチャー・プログラムをコールド・スタートすると、レプリケーション環境内のすべてのターゲット表がリフレッシュされます。

UNIX、Windows または z/OS 上で、warmns、warmsa、または warmsi オプションを指定してキャプチャー・プログラムを始動した場合、プログラムは再始動 (IBMSNAP_RESTART) 表の中の再始動点に基づいてログ・レコードのリトリブを試みます。キャプチャー・プログラムがログを検出できないと、キャプチャー・プログラムのウォーム・スタートは失敗します。ユーザーが warmns または warmsi オプションを指定してキャプチャー・プログラムを始動すると、再始動プロセスは終了し、エラー・メッセージが発行されます。ユーザーが warmsa オプションを指定してキャプチャー・プログラムを始動すると、再始動プロセスは停止し、キャプチャー・プログラムはコールド・スタートを実行し、CD 表および UOW 表の中のすべてのレコードを削除します。

キャプチャー・プログラムのコールド・スタートを防止するために、以下の推奨事項を考慮してください。

- UNIX、Windows および z/OS オペレーティング・システムの場合は warmsa を使用せずに、できるだけ warmns または warmsi 始動モードを使用してキャプチャー・プログラムを再始動してください。warmns および warmsi オプションを使用すると、再始動プロセスが失敗した場合の、キャプチャー・プログラムの自動コールド・スタートを防止できます。詳細については、542 ページの『asncap: キャプチャーの開始 (UNIX、Windows、z/OS)』を参照してください。
- OS/400 オペレーティング・システムの場合は、**RESTART(*YES)** パラメーターを指定してキャプチャー・プログラムを始動してください。キャプチャー・プログラムは、以前に終了したときに停止したポイントから処理を続けます。
- レプリケーション・アラート・モニター、またはその他のメカニズムを使用して、キャプチャー・プログラムからの履歴データの状況を確認してください。次にこの情報を使用して、DB2 がアクティブのときには必ずキャプチャー・プログラムが実行されていることを検証できます。
- システム上に十分な DB2 ログ・データおよびジャーナル・レシーバーが保存され、このデータが DB2 から使用可能であることを確認してください。ログ保存の詳細は、618 ページの『ソース・ログおよびジャーナル・レシーバーの保守および保存』を参照してください。

コントロール表の入出力エラーおよび接続障害からのリカバー: コントロール表で入出力エラーまたは接続障害が発生した場合には、DB2 の標準リカバー手順を使用して、表の順方向リカバーを行います。表のデータが失われることはありません。

キャプチャー・プログラムは入出力エラーまたは接続障害を検出すると、適切なエラー・メッセージを発行してシャットダウンします。ユーザーはエラーを訂正してから、障害発生時点からキャプチャー・プログラムを再開できます。

アプライ・プログラムは、コントロール表に重大なエラーを検出するとシャットダウンします。アプライ・プログラムはターゲット表のエラー、またはネットワーク接続のエラーを検出すると、アプライ・トレール (IBMSNAP_APPLYTRAIL) 表にエラーを書き込み、処理を続行します。

脱落したソース・データのリトリブ: ソース表で障害時点まで順方向リカバーが行われると、DB2 レプリケーションは正常に進行します。表がリカバーされると、キャプチャー・プログラムは表のデータ変更の収集を続行します。

しかし、キャプチャー・プログラムおよびアプライ・プログラムは、読み取り専用ターゲット表のポイント・イン・タイム・リカバリーは検出しません。ユーザーがソース表をリカバーした場合、アプライ・プログラムが、ソース表にもはや存在していない変更をターゲット表に複製した可能性があるため、ユーザーがターゲット表を同じ論理ポイント・イン・タイムまで戻せない場合は、ソース表とターゲット表の間で不整合が生じる可能性があります。

複数レベルでレプリケーションが行われる場合は、この状況がより複雑なものになります。各レベル間のリカバー点を照合するためのメカニズムを提供するか、リカバー方式としてフル・リフレッシュを使用する必要があります。

ターゲット表の保守

ターゲット・サーバー上の表は、データベース・システムの他の表を保守するのと同じ方法で保守してください。ターゲット表が既存のデータベース表である場合も、DB2 レプリケーションにより自動的に生成されるように指定された表である場合も、これらの表に対して現在のバックアップおよび保守のルーチンを使用してください。

重要: ユーティリティーを実行するためにターゲット表をオフラインにする前に、アプライ・プログラムを非活動化してください。

関連概念:

- レプリケーションのガイドおよびリファレンス の『DB2 レプリケーション・コンポーネントの通信方法』

関連タスク:

- レプリケーションのガイドおよびリファレンス の『レプリケーションの計画』
- 663 ページの『レプリケーションの設定』

レプリケーション・オブジェクトの命名規則

以下の表は、レプリケーション・オブジェクトの名前の制限を示しています。

表 168. レプリケーション・オブジェクトの名前の制限

オブジェクト	名前の制限
ソース表とターゲット表	データベース管理システムの命名規則にしたがってください。
ソース列とターゲット列	データベース管理システムの命名規則にしたがってください。(すべての変更前イメージ列には、1 文字の接頭部が追加されます。変更前イメージ列の名前が未確定にならないように、ソース列名が 29 文字のユニークなものになるようにしてください。)
サブスクリプション・セット	サブスクリプション・セット名には、DB2 で可変文字 (VARCHAR) 列に許可されているすべての文字を使用できません。 推奨: DB2 表および列名の命名規則にしたがってください。DB2 レプリケーションは、サブスクリプション・セット名を各レプリケーション・コントロール・サーバーに保管するため、名前を 3 つのすべてのサーバーのコード・ページと互換性のあるものにしてください。
キャプチャー・スキーマ	UNIX、Windows、z/OS の場合: キャプチャー・スキーマには、30 文字以下のストリングを使用できます (ただし、DB2 for z/OS および DB2 for OS/390 の場合の制限は 18 文字)。 OS/400 の場合: キャプチャー・スキーマ (CAPCTLLIB) は 10 文字以下のストリングにできます。
アプライ修飾子	UNIX、Windows、z/OS の場合: アプライ修飾子は、18 文字以下のストリングにできます。 OS/400 の場合: アプライ修飾子には 18 文字以下のストリングを使用できますが、アプライ・ジョブの長さは 10 文字までにしかできないため、特定のアプライ修飾子について、最初の 10 文字はユニークでなければなりません。
モニター修飾子	UNIX、Windows、z/OS の場合: モニター修飾子には 18 文字以下のストリングを使用できます。

また、キャプチャー・スキーマ、アプライ修飾子、およびモニター修飾子では、これらのオブジェクトの名前に、以下の有効文字のみを使用してください。

- A から Z (上段シフト文字)

- a から z (小文字)
- 数表示 (0 から 9)
- 下線文字 "_"

ブランクは使用できません。また、コロン ":" および正符号 "+" などの特殊文字も使用できません。

レプリケーション・システム・コマンドおよびレプリケーション・センターは、デフォルトとして、ユーザーから指定されたすべての名前を大文字に変換します。入力された名前のおりに正確に大文字と小文字を維持するには、大文字小文字混合文字の名前を二重引用符 (またはターゲット・システムで使用できるように構成されている他の文字) で囲んでください。たとえば、myqual または MyQual または MYQUAL と入力すると、名前は MYQUAL として保管されます。これらの同じ名前を二重引用符で囲んで入力すると、myqual または MyQual または MYQUAL としてそれぞれ保管されます。オペレーティング・システムによっては二重引用符が認識されないことがあります。その場合は、エスケープ文字としてバックスラッシュまたは円記号 (¥) を使用する必要があります。

Windows オペレーティング・システムの場合は、同じ名前を区別するためにユニーク・パスを使用する必要があります。たとえば、3 つのアプライ修飾子、myqual、MyQual、および MYQUAL を使用するとします。3 つの名前は同じ文字を使用していますが、大文字と小文字が異なります。これら 3 つの修飾子が同じアプライ・パスにあると、名前の競合が発生します。

重要: キャプチャー・プログラム、アプライ・プログラム、またはレプリケーション・アラート・モニター用に Windows サービスをセットアップするときには、キャプチャー・スキーマ、アプライ修飾子、およびモニター修飾子にユニークの名前を使用する必要があります。大文字と小文字を使用してこれらの名前を区別することはできません。

Windows Service Control Manager を使用してシステム・コマンドを発行する方法 (Windows)

このセクションでは、Windows オペレーティング・システムの場合に、レプリケーション・プログラムを始動するサービスを作成する方法を説明します。キャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、およびモニター・コントロール・サーバーごとにサービスを作成できます。サービスは他の DB2 サービスとグループ化されます。サービスを開始した後で、プログラムのパラメーターを変更する場合は、サービスをドロップして、新しいサービスを作成する必要があります。

- 635 ページの『レプリケーション・サービスの作成』
- 636 ページの『レプリケーション・サービスの操作』
- 637 ページの『レプリケーション・サービスのドロップ』

レプリケーション・サービスの作成

レプリケーション・サービスを作成する前に、DB2 インスタンス・サービスが実行されていることを確認してください。レプリケーション・サービスを作成するときに DB2 インスタンス・サービスが実行されていないと、レプリケーション・サービスは作成されませんが、自動的に開始されなくなります。

以下の方法のいずれかを使用して、レプリケーション・サービスを作成します。

- レプリケーション・センターで始動するプログラムの始動パラメーターを指定し、「**Windows サービスとして *program_name* プログラムを開始**」を選択します。詳細については、レプリケーション・センターのヘルプを参照してください。
- **asnscri** コマンドを使用します。

ヒント: レプリケーション・サービスが正しくセットアップされた場合は、サービスが正常に開始された後、サービス名が **STDOUT** に送信されます。サービスが開始されない場合は、開始しようとしているプログラムのログ・ファイルを調べてください。デフォルトでは、ログ・ファイルは **DB2PATH** 環境変数で指定されたディレクトリ内にあります。サービスとして開始するプログラムのパス・パラメーターを指定して (**capture_path**、**apply_path**、**monitor_path**)、このデフォルトをオーバーライドできません。また、Windows Service Control Manager (SCM) を使用して、サービスの状況を表示できます。

サービスを作成するときには、Windows にログオンするために使用するアカウント名と、そのアカウント名のパスワードを指定する必要があります。

システムには複数のレプリケーション・サービスを追加できます。すべてのキャプチャー・サーバーのスキーマごとにサービスを追加し、すべてのアプライ・コントロール・サーバーおよびモニター・コントロール・サーバーの修飾子ごとにサービスを追加できます。たとえば、5 つのデータベースがあり、各データベースがアプライ・コントロール・サーバー、キャプチャー・コントロール・サーバー、およびモニター・コントロール・サーバーである場合は、15 のレプリケーション・サービスを作成できます。各サーバーに複数のスキーマまたは修飾子がある場合は、さらに多くのサービスを作成できます。

ユーザーが作成したレプリケーション・サービスは、自動モードで SCM に追加され、サービスが開始されます。Windows はユニークのサービス名および表示名を使用してサービスを登録します。

レプリケーション・サービス名

レプリケーション・サービス名は、各サービスを一意的に識別します。サービスを停止または開始するときにはこれを使用します。名前は以下のようなフォーマットになります。

`DB2.instance.alias.program.qualifier_or_schema`

ここで、

- *instance* は DB2 インスタンスの名前です。
- *alias* は、キャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、またはモニター・コントロール・サーバーのデータベース別名です。
- *program* は以下の値のいずれかです。CAP (キャプチャー・プログラム)、APP (アプライ・プログラム)、または MON (レプリケーション・アラート・モニター・プログラム)
- *qualifier_or_schema* は、アプライ修飾子、モニター修飾子、またはキャプチャー・スキーマという ID のいずれかです。

インスタンス: 以下のサービス名は、スキーマ ASN を持ち、INST1 という名前のインスタンスの下でデータベース DB1 を処理しているキャプチャー・プログラムのものです。

DB2.INST1.DB1.CAP.ASN

レプリケーション・サービスの表示名

表示名は、「サービス」ウィンドウで表示されるテキスト・ストリングで、読みやすい形式のサービス名です。たとえば、以下のようになります。

DB2 - INST1 DB1 CAPTURE ASN

サービスの記述を追加する場合は、レプリケーション・サービスを作成した後で、Service Control Manager (SCM) を使用してください。また、SCM を使用し、サービスのユーザー名およびパスワードを指定することもできます。

レプリケーション・サービスの操作

レプリケーション・サービスを作成したら、サービスを停止してから、再度開始できません。

以下の方法のいずれかを使用して、サービスを停止します。

- SCM
- **net stop** コマンド

重要: レプリケーション・サービスを停止すると、サービスに関連するプログラムは自動的に停止します。しかし、レプリケーション・システム・コマンド (**asnacmd**、**asnccmd**、または **asnmcmd**) を使用してプログラムを停止すると、プログラムを始動するために使用されたサービスは、ユーザーが明示的に停止するまで、実行を続けます。

レプリケーション・コマンドのサービスを開始するには、以下の方法のいずれかを使用します。

- SCM

- **net start** コマンド
- レプリケーション・センター

重要: サービスからレプリケーション・プログラムを始動した場合、同じスキーマまたは修飾子を使用してプログラムを始動しようとすると、エラーが発生します。

レプリケーション・サービスのドロップ

レプリケーション・サービスが必要なくなった場合は、SCM からドロップされるように、サービスをドロップできます。また、サービスによって開始されるプログラムの始動パラメーターを変更する場合は、サービスをドロップして、新しい始動パラメーターを使用する新しいサービスを作成する必要があります。

レプリケーション・コマンドのサービスをドロップするには、**asnsdrop** コマンドを使用します。

関連資料:

- レプリケーションのガイドおよびリファレンス の『**asnsrct**: キャプチャー、アプライ、またはレプリケーション・アラート・モニターを開始する DB2 レプリケーション・サービスの作成 (Windows のみ)』
- レプリケーションのガイドおよびリファレンス の『**asnsdrop**: DB2 レプリケーション・サービスのドロップ (Windows のみ)』

表およびビューをレプリケーション・ソースとして登録する

DB2 レプリケーションの場合、レプリケーション・ソースとして使用する表とビューは、それらを登録するという方法で示します。レプリケーション用として特定の表またはビューを登録する際には、後でさまざまな目的のために異なるターゲットに対して使用できる使用可能なデータのソースを作成します。この章で説明されている管理タスクは、レプリケーション・ゴールに基づいてデータが各ソースからどのように取り込まれるのかを定義するコントロール情報をセットアップするのに役立ちます。

ソースの登録時には、レプリケーション・ソースとして使用する表またはビュー、表内でレプリケーションのために使用可能にする列、および DB2 レプリケーションがソースからデータと変更をキャプチャーする方法に関するプロパティーを示してください。

DB2 レプリケーションでは、以下のオブジェクトをソースとして登録できます。

- DB2 表
- 非 DB2 リレーショナル表 (ニックネームを使用)
- 表内のデータのサブセット (DB2 または非 DB2 リレーショナル)
- 単一表のビュー (DB2)
- 複数の表の内部結合を表すビュー (DB2)

この章は、以下のセクションを含んでいます。

- 『DB2 表をソースとして登録する』
- 640 ページの『非 DB2 リレーショナル表をソースとして登録する』
- 641 ページの『ソース表用の登録オプション』
- 658 ページの『レプリケーション・ソースとしてのビューの動作』
- 661 ページの『表のビューをソースとして登録する』
- 662 ページの『CCD 表をソースとして保守する (IMS)』

DB2 表をソースとして登録する

このセクションでは、DB2 表をレプリケーション・ソースとして登録する方法を説明します。DB2 レプリケーションでは、以下のタイプの DB2 表がソースとしてサポートされています。

UNIX および Windows の場合

- ユーザー・アプリケーションが保守する DB2 表
- カタログ表 (フル・リフレッシュのみのレプリケーション)
- 自動サマリー表
- 外部 CCD 表

z/OS の場合

- ユーザー・アプリケーションが保守する DB2 表
- カタログ表
- 外部 CCD 表

OS/400 の場合

- ユーザー・アプリケーションが保守する DB2 表 (ローカルまたはリモート側でジャーナリングされる)
- 外部 CCD 表

OS/400 の場合を除くすべての DB2 ソースについては、ソース表 DDL は DATA CAPTURE CHANGES オプションを必要とします。このオプションをソースから除去しないでください。

表をソースとして登録すると、CD (変更データ) 表が作成されます。登録された表に関連付けられているキャプチャー・プログラムは、そのソース用のログを読み取って、登録された列に対して発生した未完了の変更を、トランザクションがコミットまたはロールバックされるまでメモリーに保管します。ロールバックの場合は、それらの変更はメモリーから削除されます。コミットの場合は、それらの変更はキャプチャー・プログラムがコミット・ログ・レコードを読み取ると同時に CD 表に挿入されます。これらの変更は、キャプチャー・プログラムが各キャプチャー・サイクルの後でコミットするまで

メモリー内に残されます。キャプチャー・プログラムは、CAPSTART シグナルがユーザーまたはアプライ・プログラムによって発行されるまで、DB2 ソース表のデータのキャプチャーを開始しません。

非リレーショナル・ソース表の場合の注: IMS などの非リレーショナル・データベース管理システムからのデータを含む DB2 表を登録できます。これを行うには、IMS DataPropagator や Data Refresher などといった、非リレーショナル・データベースからのデータを含む CCD 表を移植するためのアプリケーションが必要です。このアプリケーションは、IMS データベース内の非リレーショナル・セグメントへの変更をキャプチャーし、CCD 表を移植します。この CCD 表はコンプリートでなければなりませんが、コンデンスされていてもコンデンスされていなくても構いません。他の CCD ソースと同様に、CCD ソース表には関連付けられているキャプチャー・プログラムがありません。なぜなら、この表には非リレーショナル・ソース表からの変更済みデータがすでに保管されているからです。IMS DataPropagator および Data Refresher 製品は、登録 (IBMSNAP_REGISTER) 表内の値を保守して、アプライ・プログラムがこのソース表からの読み取りを正しく行えるようにします。上記の製品のいずれかを使用してこれらのタイプの CCD 表を保守するのではなく、自分自身でそれらの表を保守する場合は、662 ページの『CCD 表をソースとして保守する (IMS)』を参照してください。

前提条件:

ソースとして登録する表を処理させるキャプチャー・コントロール・サーバー上に、キャプチャー・コントロール表がすでに存在していなければなりません。キャプチャー・コントロール表を作成する必要がある場合は、673 ページの『レプリケーション・コントロール表のセットアップ』を参照してください。

制約事項 (OS/400):

- SQL ステートメントは長さが 32,000 文字までに制限されているため、1 つの表当たりおよそ 2000 列までしか登録できない。正確な列数は、列名の長さによって決まります。
- 単一キャプチャー・スキーマに対し、同じジャーナルを使用するソース表を 300 個より多く登録してはならない。
- ソース表、CD 表およびソース表のジャーナルは、すべてこれらのソース表の登録情報を含むキャプチャー・コントロール表と同じ補助記憶域プール (ASP) になければなりません。

手順:

以下の方法のいずれかを使用して、DB2 表を登録してください。

レプリケーション・センター

「表の登録」ウィンドウを使用します。詳細については、レプリケーション・センターのヘルプを参照してください。

ヒント: 登録の際の時間を節約するために、キャプチャー・コントロール・サーバーに対してあらかじめソース・オブジェクト・プロファイルを設定アップしておけます。そのようにした場合は、表の登録時に、レプリケーション・センターはレプリケーション・センターのデフォルトではなく、そのソース・オブジェクト・プロファイルに定義されたデフォルトを使用します。これによって登録の際の時間を節約できます。なぜなら、それぞれの表を一度に 1 つずつ選択してデフォルト設定を手動で変更する代わりに、デフォルトを一度に上書きできるからです。

レプリケーション用のシステム・コマンド (OS/400)

ADDDPRREG システム・コマンドを使用します。このコマンドの構文およびパラメーターについての記述は、570 ページの『ADDDPRREG: DPR 登録の追加 (OS/400)』を参照してください。

DB2 表の登録時には、ソース・サーバー、ソース表の名前、およびキャプチャー・スキーマを指定して、どの表を登録するのかを示してください。別々のキャプチャー・スキーマを使用して、同じ表を複数回登録できます。登録用のデフォルト設定を使用するか、あるいはレプリケーションに関するニーズに合わせて登録オプションを変更するかは任意です。登録オプションとそれらのデフォルトの完全なリスト、およびこれらのデフォルトを使用した方がよい場合と変更した方がよい場合についての説明は、641 ページの『ソース表用の登録オプション』を参照してください。

非 DB2 リレーショナル表をソースとして登録する

このセクションでは、非 DB2 リレーショナル表をレプリケーション・ソースとして登録する方法を説明します。DB2 レプリケーションは、ニックネームを使用して非 DB2 リレーショナル表にアクセスします。

非 DB2 リレーショナル表をソースとして登録すると、CCD (整合変更データ) 表が作成されます。登録済みの非 DB2 リレーショナル表に対する変更が発生すると、キャプチャー・トリガーがキャプチャー・プログラムをシミュレートして、その変更を CCD 表に挿入します。キャプチャー・トリガーは、非 DB2 リレーショナル・ソース表への変更のキャプチャーを、そのソースが登録された時点から開始します。

デフォルトでは、CCD 所有者はソース表のスキーマ名から導き出されます。CCD 所有者をスキーマ名と一致しないように変更する場合は、ソース表の所有者が CCD 表への書き込みを許可されていることを確認してください。ソース表の所有者が CCD 表を更新できない場合は、ソース表のトリガーは変更を CCD 表に書き込めません。

前提条件:

このソースを処理させるキャプチャー・コントロール・サーバー上に、キャプチャー・コントロール表がすでに存在していなければなりません。キャプチャー・コントロール表を作成する必要がある場合は、674 ページの『非 DB2 リレーショナル・ソース用のコントロール表の作成』を参照してください。

制約事項:

- 複数の非 DB2 リレーショナル・ソース・サーバーにアクセスするのに単一の連合 DB2 データベースを使用している場合は、その単一連合データベースではそれぞれの非 DB2 リレーショナル・ソース・サーバーごとに異なるキャプチャー・スキーマを使用する必要があります。どれも同じであってはなりません。各非 DB2 リレーショナル表は、それぞれ 1 つのキャプチャー・スキーマの下にしか登録できません。
- データ・タイプ LOB または DATALINK を持つ非 DB2 リレーショナル表には列を登録できない。これらのデータ・タイプを含む表を登録する場合は、列のサブセットを登録する必要があります。列のサブセットだけを登録する方法についての詳細は、642 ページの『列のサブセットの登録 (垂直方向のサブセット化)』を参照してください。

手順:

以下に、非 DB2 リレーショナル表を登録する方法を示します。

レプリケーション・センター

「ニックネームの登録」ウィンドウを使用します。詳細については、レプリケーション・センターのヘルプを参照してください。

ヒント: 登録の際の時間を節約するために、キャプチャー・コントロール・サーバーに対してあらかじめソース・オブジェクト・プロファイルを設定アップしておけます。そのようにした場合は、表の登録時に、レプリケーション・センターはレプリケーション・センターのデフォルトではなく、CCD 表用のソース・オブジェクト・プロファイルに定義されたデフォルトと CCD 表のニックネームを使用します。これによって登録の際の時間を節約できます。なぜなら、それぞれの表を一度に 1 つずつ選択してデフォルト設定を手動で変更する代わりに、デフォルトを一度に上書きできるからです。

非 DB2 リレーショナル表の登録時には、ソース表のニックネームを指定して、どの表を登録するのかを示してください。登録用のデフォルト設定を使用するか、あるいはレプリケーションに関するニーズに合わせて登録オプションを変更するかは任意です。登録オプションとそれらのデフォルトの完全なリスト、およびこれらのデフォルトを使用した方がよい場合と変更した方がよい場合についての説明は、『ソース表用の登録オプション』を参照してください。

ソース表用の登録オプション

このセクションでは、表をレプリケーション・ソースとして登録する際に使用できるさまざまなオプションについて説明します。これらのオプションは、表の登録に関するタスクのうち、より大きな方のタスクの一部です。各表の登録方法の参照先を以下に示します。

- DB2 表、638 ページの『DB2 表をソースとして登録する』を参照してください。
- 非 DB2 リレーショナル表、640 ページの『非 DB2 リレーショナル表をソースとして登録する』を参照してください。

表のビューを作成してそれらのビューをソースとして登録するときには、ビューの登録オプションは基本表の登録定義によって決定されます。ビューが基本表からどの特性を継承するのか、およびビューが基礎登録定義にしたがってどのように動作するのかについての詳細は、658 ページの『レプリケーション・ソースとしてのビューの動作』を参照してください。

登録する表の選択を済ませたら、レプリケーションのためにどの列を使用可能にするのかを示して、このソースから登録されたデータの処理方法と保管方法を決定するプロパティを定義できます。また、キャプチャー・プログラムによる CD 表へのソース・データの保管方法（またはキャプチャー・トリガーによる CCD 表へのデータの保管方法）などといった、他の登録オプションも指定できます。このセクションでは、表をソースとして登録する際に指定可能な以下のオプションについて説明します。

- 『列のサブセットの登録 (垂直方向のサブセット化)』
- 643 ページの『フル・リフレッシュ・コピーと変更キャプチャー・レプリケーション』
- 645 ページの『変更後イメージ列と変更前イメージ列』
- 648 ページの『変更前イメージ接頭部』
- 648 ページの『エラー発生時におけるキャプチャー・プログラムの停止』
- 649 ページの『キャプチャー・プログラムが更新を保存する方法』
- 650 ページの『変更の再キャプチャーの防止 (update-anywhere レプリケーション)』
- 655 ページの『競合検出の設定 (update-anywhere レプリケーション)』
- 658 ページの『主キーの代わりに相対レコード番号 (RRN) を使用する (OS/400)』

列のサブセットの登録 (垂直方向のサブセット化): デフォルト: すべての列がレプリケーションのために登録されます

レプリケーション用のソース表を定義するときには、その表内のすべての列をレプリケーションのために登録しなければならないわけではなく、ソース表の列のサブセットを登録できます。この垂直サブセットは、サブスクライブするターゲットからすべての列を使用可能にするのは望ましくない場合に有用です。また、このオプションは、このソースのターゲット表ではソース表に定義されているデータ・タイプのすべてはサポートされていない場合にも選択するとよいでしょう。

列のサブセットを登録するには、ターゲット表へのレプリケーションのために使用可能にしたい列だけを選択します。選択しない列は、どのターゲット表へのレプリケーションにも使用できません。CD (および CCD) 表には一部のタイプのターゲット表にとって十分なキー・データ (ポイント・イン・タイムなど) が含まれていないため、使用するサブセットにターゲットに対してキー列 (主キーまたはユニーク索引) として機能する列が含まれていることを確認してください。

ヒント: ソース表内の列のサブセットを登録するのは、未登録の列は決して複製しないことを確信している場合だけにしてください。ソースの列のサブセットを登録し、後に

なって登録しなかった列を複製する必要がある場合は、登録を変更して未登録の列を追加しなければなりません (非 DB2 リレーショナル・ソースの場合は、1 つの登録に新しい列を追加するのにすべての登録をまとめて再定義する必要があります。) このソースに関連した内部 CCD を持つ予定である場合は、後で列を追加するのはさらに困難になる可能性があります。なぜなら、新しい列を登録すると、それらは内部 CCD ではなく、CD 表に追加されるからです。これらの問題を回避するためには、ソースのすべての列を登録し、代わりにアプライ・プログラムを使用して、ターゲットに複製する列をサブセット化するという方法もあります。ソースでなくターゲットでサブセット化を行う方法についての詳細は、716 ページの『ターゲットに適用するソース列』を参照してください。

フル・リフレッシュ・コピーと変更キャプチャー・レプリケーション: デフォルト: 変更キャプチャー・レプリケーション

レプリケーション・サイクルで毎回ソース表内のすべてのデータをターゲットに複製する (フル・リフレッシュのみのレプリケーション) か、ターゲットが最後にリフレッシュされた時点以降に発生した変更のみを複製 (変更キャプチャー・レプリケーション) するかを選択できます。

制約事項: 複数ノード DB2 Enterprise Server Edition (ESE) 構成で変更を取り込めるのは、ソース表がパーティションに分割されていない状態でカタログ・ノードに常駐している場合だけです。

フル・リフレッシュのみのレプリケーション: ターゲットがフル・リフレッシュのみのレプリケーション用に登録されているソースに対してサブスクライブしている場合、アプライ・プログラムは毎回のレプリケーション・サイクルで、ターゲット表からすべてのデータを削除し、ソースの登録済み列に含まれているデータをコピーして、ソース・データをターゲットに移植します。キャプチャー・プログラムは関与せず、CD 表もありません。アプライ・プログラムは直接ソース表からデータを読み取ります。

表が小さい場合のヒント: コピーするのに時間もリソースもあまり要しない非常に小さなソース表を使用している場合は、フル・リフレッシュのみのレプリケーションを選択するとよいでしょう。

表が大きい場合のヒント: 表が比較的大きく、かつフル・リフレッシュのみのレプリケーションを使用したい場合は、表をより迅速にロードするために ASNLOAD 出口ルーチンを使用するとよいでしょう。詳細については、493 ページの『ASNLOAD 出口ルーチンを使ったターゲット表のリフレッシュ』を参照してください。

制約事項: このソースにサブスクライブするのをコンデンスされたターゲット表にする予定であり、さらにそのターゲット表のユニーク索引は用意できない場合は、ソースをフル・リフレッシュのみのレプリケーション用として登録する必要があります。

変更キャプチャー・レプリケーション: デフォルト: すべての行に対する変更が取り込まれます

変更キャプチャー・レプリケーション時には、変更されたデータだけがターゲット表にレプリケーションされます。このソース用に選択したターゲット表のタイプによっては、この表の初期ロードを実行する必要があります。ほとんどの場合、アプライ・プログラムは初期フル・リフレッシュを実行してから、継続して変更キャプチャー・レプリケーションを実行をします。

ターゲット表に対するフル・リフレッシュを許可しないことを選択すると、ソース表とターゲット表の再同期化が必要になった場合にターゲット表を手動で再ロードする必要があります。ターゲットが初期ソース・データを含んでいる状態でロードされた後、キャプチャー・プログラムはソースで発生した変更をキャプチャー CD 表に保管します。非 DB2 リレーショナル・ソースの場合の変更キャプチャー・レプリケーションでは、ソースで発生した変更をキャプチャー・トリガーがキャプチャー CCD 表に保管します。アプライ・プログラムは、CD 表または CCD 表から変更を読み取って、その登録されているソースにサブスクライブしているターゲットにそれらの変更を適用します。

DB2 ソース表を変更キャプチャー・レプリケーション用として定義する場合、ソースで発生した変更をすべて CD 表に保管するのは望ましくないことがあります。行 (水平方向) サブセットを登録して変更をフィルター操作し、実際にソースで発生している変更より少ない変更が CD 表に取り込まれるようにできます。以下の 2 つの行キャプチャー規則のいずれかを選択して、キャプチャー・プログラムがソース表からのどの変更済み行を CD 表に記録するのかを決定できます。

- すべての行に対する変更をキャプチャーする
- 変更が登録済みの列に発生した場合にのみ変更をキャプチャーする (DB2 のみ)

デフォルトでは、ソース表で行のいずれかの列 (登録済みでも未登録でも) が更新されるたびに、必ず変更が取り込まれます。列のサブセットだけを登録した場合は、ソース表に対して変更が発生するたびに、キャプチャー・プログラムは登録済み列の値を行単位で CD 表に記録します。これは、変更された列が登録済みの列とは異なる列であった場合でも実行されます。このデフォルト・オプションは、ソース表へのすべての変更の履歴を保持したい場合に使用してください。非 DB2 リレーショナル・ソースの場合に使用可能なオプションはこれだけです。キャプチャー・トリガーは、変更が未登録の列に対して発生した場合であっても、ソースで変更のあったすべての行をキャプチャーします。

例: 表には 100 個の列があり、それらの列うち 50 個をレプリケーション用に登録しているとします。デフォルトでは、表内の 100 個の列のいずれか 1 つにでも変更が加えられると、いつでもキャプチャー・プログラムによって行が CD 表に書き込まれます (あるいは、キャプチャー・トリガーによって行が CCD 表に書き込まれます)。

DB2 ソースがある場合は、登録済みの列への変更だけをキャプチャー・プログラムに取り込ませたいことがあります。このケースでは、キャプチャー・プログラムは登録済みの列に対する変更が発生したときだけ行を CD 表に書き込みます。

提案: 監査を目的とする情報が必要な場合、または表ではほとんどいつも登録済みの列に対してのみ変更が発生している場合は、すべての行への変更のキャプチャーを選択してください。未登録の列にのみ影響する変更が頻繁に発生している場合は、登録済みの列だけに対する変更のキャプチャーを選択してください。このオプションは、ソース表へのすべての変更の履歴を保持するのは望ましくない場合に使用してください。

変更後イメージ列と変更前イメージ列: デフォルト: 変更後イメージ列のみ

ソースを変更キャプチャー・レプリケーション用に登録する際には、キャプチャー・プログラムに変更後イメージ値 (変更後の列内の値) だけを取り込ませるのか、あるいは変更後イメージ値と変更前イメージ値 (変更前に列内に入っていた値) の両方を取り込ませるのかを選択できます。UNIX、Windows、および z/OS の場合は、変更前イメージ値を取り組むかどうかを表内の列ごとに選択できます。OS/400 の場合は、変更前イメージのキャプチャーを表内のすべての列について行うか、あるいはどの列に対しても行わないかのどちらかを選択できます。このオプションは、個々の列ごとには選択できません。以下の各セクションでは、どのような場合に各オプションを選択する必要があるのかについて説明します。

以下に示すように、いくつかの非 DB2 リレーショナル・ソース表の場合、CCD 表には変更後イメージ値だけを含めるようにする必要があります。

- Sybase または Microsoft SQL Server の表には、タイプが **TIMESTAMP** の列は 1 つしか含められない。データ・ソースが Sybase または Microsoft SQL Server であり、かつソース表にタイプが **TIMESTAMP** の列があるときは、この列をレプリケーション・ソースの一部として定義する場合、この列については変更後イメージのみを選択してください。

以下の特定のデータ・タイプを指定された列の場合は、CD 表に変更前イメージ値を含められません。

- LOB データ・タイプの列
- DATALINK データ・タイプの列

変更後イメージ値のみのキャプチャー: 変更キャプチャー・レプリケーション用に登録するそれぞれの列ごとに、変更が行われるたびにキャプチャー・プログラムまたはキャプチャー・トリガーに対して、変更後イメージ値だけを記録させることを選択することができます。変更後イメージ値だけを取り組むことを選択した場合は、CD (または CCD) 表にはそれぞれの変更された値ごとに 1 つの列が含まれ、その列には変更発生後のソース列の値が保管されます。

このソースに対しては基礎集約ターゲット表タイプおよび変更集約ターゲット表タイプだけを使用する予定である場合は、変更前イメージは必要ありません。ターゲット表を計算済みの値のために使用する予定である場合は、変更前イメージ列は無意味です。なぜなら、計算された列には変更前イメージがないからです。他のすべてのターゲット表タイプでは、変更前イメージ列を利用することができます。集約ターゲット表についての詳細は、707 ページの『ソースにおけるデータまたは変更の算出サマリー』を参照してください。

変更前イメージ値と変更後イメージ値のキャプチャー: 変更キャプチャー・レプリケーション用に登録するそれぞれの列ごとに、変更が行われるたびにキャプチャー・プログラムまたはキャプチャー・トリガーに対して、変更前イメージ値と変更後イメージ値の両方を記録させることを選択することができます。変更前イメージ値と変更後イメージ値をキャプチャーすることを選択した場合は、CD (または CCD) 表にはそれぞれの変更された値ごとに 2 つの列が含まれます。一方の列は変更が発生する前にソース列に入っていた値用で、もう一方の列は変更発生後の値用です。

変更前イメージと変更後イメージの両方を CD (または CCD) 表に保管することを選択した場合は、以下に示すように、変更前イメージ列と変更後イメージ列には、ソース表に対して実行されたそれぞれ異なるアクションの種類に応じて異なる値が含まれます。

処置	列値
挿入	変更前イメージ列には NULL 値が含まれます。変更後イメージ列には挿入された値が含まれます。
更新	変更前イメージ列には、変更が発生する前の列値が含まれます。変更後イメージ列には、変更が発生した後の列値が含まれます。 更新を削除と挿入の対としてキャプチャーすることを選択した場合は、削除行ではその行の変更前イメージ列と変更後イメージ列の両方に更新の変更前イメージが含まれ、挿入行では変更前イメージ列には NULL 値が、変更後イメージ列には変更後イメージが含まれます。このオプションの詳細については、649 ページの『キャプチャー・プログラムが更新を保存する方法』を参照してください。
削除	変更前イメージ列と変更後イメージ列に、変更が発生する前の列値が含まれます。

UNIX、Windows、および OS/400 の場合の重要事項: 定義によって変更前イメージが含まれている列の場合、DB2 レプリケーションでは列名が 29 文字までに制限されます。なぜなら、列名全体で許容される文字数が 30 文字だけだからです。列名がそれより長いと、左方から切り捨てるようにプロファイルを設定していないかぎり、デフォルトでは余分な文字が DB2 レプリケーションによって右方から切り捨てられます。DB2 レプリケーションによってターゲット列に変更前イメージ列 ID (通常は X) が追加される上に、各列名は必ずユニークでなければならぬため、29 文字より長い列名は使用で

きません。複製するつもりがない表にはさらに長い列名を使用できますが、その列を将来複製する可能性がある場合は 29 文字の名前を使用することを検討してください。

z/OS の場合の重要事項: DB2 for z/OS の表であれば 18 文字の列名を使用できますが、DB2 DataPropagator は 18 番目の文字をターゲット表では変更前イメージ列 ID に置き換えるため、列名の最初の 17 文字が必ずユニークになるようにする必要があります。

以下のセクションでは、変更前イメージ値のキャプチャーが必要となる可能性のあるケースについて説明しています。

- 『ソース・データの履歴を保持する場合』
- 『競合検出を使用する update-anywhere 構成の場合』
- 『ターゲットでキー列が更新の対象となっている場合』

ソース・データの履歴を保持する場合: 監査目的でデータを保持する必要がある場合は、ある期間内にデータがどのように変更されたかについてのレコードを持てるように、変更前イメージと変更後イメージの両方を選択します。監査やアプリケーション・ロールバック機能を必要とする業界では、変更前イメージと変更後イメージのコピーのセットが役立ちます。

競合検出を使用する update-anywhere 構成の場合: レプリカ表 (競合検出が None 以外に設定されているもの) の間で競合が起こり得る update-anywhere 構成では、レプリカの CD 表に変更後イメージ列と変更前イメージ列の両方を登録して、競合が発生した場合には変更をロールバックできるようにする必要があります。

ターゲットでキー列が更新の対象となっている場合: ソースを登録する際には、この表をソースとして使用して定義する可能性のあるターゲット表があるかどうかを検討してください。通常ターゲット表はコンデンスされており、そのターゲット表内の各行をユニークにする列または列のセットを必要とします。それらのユニーク列によっていわゆるターゲット・キーが構成されます。これらのターゲット・キー列のいずれかがソースで更新される可能性がある場合は、DB2 レプリケーションではターゲット表で必ず正しい行が更新されるようにするための特殊な処理が必要となります。DB2 レプリケーションがターゲット表で必ず正しい行を新しいキー値で更新するようにするために、ターゲット・キーを構成する列については変更後イメージと変更前イメージの両方をキャプチャーすることを選択できます。これらの登録済み列の変更前イメージ値は、アプライ・プログラムが非キー・ソース列の変更をターゲット表のターゲット・キー列に適用するときには必要となります。変更を適用するときには、アプライ・プログラムはこの行をターゲット表でソースの CD (または CCD) 表内の変更前イメージ値と一致するターゲット・キー値を探すという方法で検索し、次にそのターゲット行をソースの CD (または CCD) 表内の変更後イメージ値で更新します。

これらの変更前イメージ値をソース表またはビューの登録時に登録しても、DB2 レプリケーションはユーザー・アプリケーションがターゲット・キーに対して更新を行うこと

を知りません。後で (サブスクリプション・セットを作成して) どのターゲットをこのソースにサブスクライブするのかを定義するときに、アプライ・プログラムに対し、変更をソースの非キー列からターゲットのキー列に適用する際に特殊な更新を実行するよう指定できます。詳細については、720 ページの『アプライ・プログラムがターゲット・キー変更オプションを使用してターゲット・キー列を更新する方法』を参照してください。

変更前イメージ接頭部: デフォルト (レプリケーション・センター): X

デフォルト (OS/400 システム・コマンド): @

CD (または CCD) 表の変更後イメージ列と変更前イメージ列の両方をキャプチャーすることを選択した場合は、変更後イメージ列名はソース表での列名となり、変更前イメージ列名は先頭に 1 文字の接頭部を追加されたソース表での列名となります。変更前イメージ列名用のデフォルトの 1 文字接頭部は変更可能です。変更前イメージ接頭部と CD (または CCD) 列名の組み合わせは、あいまいさのないものでなければなりません。これは、接頭部列名は CD (または CCD) 表内の現行列名または潜在的な列名と同じであってはならないという意味です。

制限: 変更前イメージ接頭部に、ブランク文字は使用できません。

例: 変更前イメージ接頭部として X を使用していて、さらに COL という名前のソース列を登録している場合は、XCOL という名前の列は登録できません。なぜなら、XCOL が別のソース列の実際の列名なのか、それとも COL という列名と変更前イメージ接頭部 X を持つ変更前イメージ列の名前なのかがはっきりしないからです。

表の変更前イメージ列を一切複製していない場合は、変更前イメージ接頭部を持たないことを選択して、このプロパティを NULL に設定できます。

エラー発生時におけるキャプチャー・プログラムの停止: デフォルト: キャプチャー・プログラムは特定のエラーを検出したときに停止します

登録の処理中に特定の問題を検出すると、キャプチャー・プログラムは停止するか、または実行を継続します。以下のオプションのいずれかを選択して、登録されたソースの処理中に特定のエラーを検出した場合のキャプチャー・プログラムの反応を決定できます。

エラー発生時にキャプチャーを停止

キャプチャー・プログラムは、キャプチャー・トレース (IBMSNAP_CAPTRACE) 表にエラー・メッセージを書き込んで終了します。

エラー発生時にキャプチャーを停止しない

特定のエラーが発生した場合、キャプチャー・プログラムは実行を継続します。ソース処理の初回の試行中にエラーを検出した場合は、キャプチャー・プログラムは登録を活動化しません。登録されたソースがすでに活動化されていた場合は、登録の処理を停止します。

このオプションにより、以下の致命的でないエラーが発生した場合にキャプチャー・プログラムが稼働し続けるかどうかが決まります。

- 登録が正しく定義されていません。
- キャプチャー・プログラムが変更済みデータの行の挿入を試行したときに、CD 表が検出されなかった。
- キャプチャー・プログラムが始動または再初期化されたときに、(非 OS/400) ソース表の DATA CAPTURE CHANGES オプションがオフにされた状態で検出された。

致命的エラーの場合は、キャプチャー・プログラムは実行を継続しません。

キャプチャー・プログラムが更新を保存する方法: デフォルト: 更新は CD 表の単一行に保管されます

ソースの更新を CD (または CCD) 表に保管する方法を選択できます。ソース表に対する更新を取り込んだ場合、キャプチャー・トリガーまたはキャプチャー・プログラムは、更新された値を CD 表の単一行に保管するか、あるいは CD (または CCD) 表の 2 つの行を使用して、削除を一方の行に、挿入をもう一方の行に保管するかのいずれかを実行できます。デフォルトでは、更新は単一行に保管されます。このデフォルトを使用すると、ストレージが削減され、パフォーマンスが向上します。なぜなら、CD (または CCD) 表には 1 行だけが保管され、さらにその行は変更が行われるたびにアプライ・プログラムによって読み取られるからです。ただし、キャプチャー・プログラムまたはキャプチャー・トリガーに対してソース表への更新を DELETE と INSERT の対としてキャプチャーするよう指示する必要があるシナリオがいくつかあります。

ソース・アプリケーションがサブスクリプション・セット・メンバーの述部で参照される列を 1 つ以上更新する場合は、更新を DELETE ステートメントと INSERT ステートメントとしてキャプチャーする必要があります。ソース・データだけにサブスクライブするターゲットを、特定の列値を基にした述部を使用して定義する (たとえば、WHERE DEPT = 'J35') 予定であるとしてします。その列を変更すると (たとえば、DEPT='FFK' に変更)、取り込まれた変更は、ターゲットにレプリケーションする対象として選択されません。なぜなら、その列は述部の基準に合致していないからです。つまり、新しい FFK 部門は、サブスクリプション・セットが部門 J35 に基づいているため複製されません。更新を DELETE および INSERT の対に変換すると、そのターゲット表列は確実に削除されます。

取り込まれたそれぞれの更新は、CD (または CCD) 表の 2 つの行に変換されます。この変換は、すべての列について実行されます。この取り込んだデータの増加に合わせて CD (または CCD) 表のスペース割り振りの調節が必要になる場合もあります。

DATALINK 値の場合の重要事項: DATALINK 列が ON UNLINK DELETE として定義されている場合、DELETE と INSERT の対が同じトランザクション内で処理されるため、このリンク解除は無視されます。外部ファイルは削除されませんが、更新は行われます。

変更の再キャプチャーの防止 (update-anywhere レプリケーション): 新しいソース表の場合のデフォルト: 変更が再キャプチャーされます

新しいレプリカ表の場合のデフォルト: 変更は再キャプチャーされません

制約事項: 非 DB2 リレーショナル・データベースからの表は、update-anywhere にかかわられません。したがって、このオプションは DB2 ソース専用です。

update-anywhere レプリケーションの場合は、変更はマスター表または関連したレプリカ表で発生する可能性があります。update-anywhere レプリケーションで使用する予定の表を登録する場合、DB2 レプリケーションはその表がユーザーの構成の中でマスター表となることを前提とします。再キャプチャー・オプションを使用すると、あるサイトで発生して 2 番目のサイトに複製された変更を、その 2 番目のサイトで再キャプチャーし、それによってその変更を追加の各サイトへのレプリケーションに使用できるようにするかどうかをコントロールできます。登録時に、このオプションをマスター表に対して設定してください。その後、そのマスター・ソース表をレプリカ・ターゲットにマップする際に、レプリカの変更を再キャプチャーして他の表に転送するかどうかを設定できます。(マスターをレプリカにマップする方法についての詳細は、712 ページの『読み取り / 書き込みターゲットの定義 (update-anywhere)』を参照してください。)

update-anywhere 構成でマスターとして機能するソース表を登録する場合は、以下の 2 つのオプションのうち、どちらかを選択できます。

マスターで変更を再キャプチャーする

あるレプリカで発生したマスターに対する更新は、マスターで再キャプチャーされて、他のレプリカに転送されます。

マスターで変更を再キャプチャーしない

あるレプリカで発生してマスターに加えられた更新は、マスターで再キャプチャーされず、他のレプリカに転送されません。

update-anywhere 構成のレプリカ表を登録する場合は、以下の 2 つのオプションのうち、どちらかを選択できます。

レプリカで変更を再キャプチャーする

マスターで発生してレプリカに加えられた更新は、そのレプリカで再キャプチャーされて、このレプリカにサブスクライブしている他のレプリカに転送されます。

レプリカで変更を再キャプチャーしない

マスターで発生してレプリカに加えられた更新は、そのレプリカで再キャプチャーされず、このレプリカにサブスクライブしている他のレプリカに転送されません。

変更の再キャプチャーを防止すると、パフォーマンスを向上させ、さらにストレージ・コストを削減できます。なぜなら、キャプチャー・プログラムが同じ変更をそれぞれのレプリカごとに再キャプチャーしないからです。

以下のセクションでは、update-anywhere 構成に基づいて変更を再キャプチャーするかどうかを決定する方法について説明しています。

- 『レプリカを 1 つだけ持つマスターの場合』
- 『マスターの相互に排他的なパーティションである複数のレプリカの場合』
- 652 ページの『変更を複数のレプリカに複製するマスターの場合』
- 653 ページの『他のレプリカに変更を複製するレプリカの場合 (multi-tier)』

レプリカを 1 つだけ持つマスターの場合: マスター: 変更をマスターで再キャプチャーしません

単一レプリカ: 変更をレプリカで再キャプチャーしません

update-anywhere 構成内にレプリカを 1 つしか持たない予定である場合は、マスター表でもレプリカ表でも変更が再キャプチャーされないようにしたいことがあります。マスター表が他のレプリカ表にとってのソースではなく、レプリカが他のレプリカにとってのソースではない場合 (multi-tier 構成において) は、これが最良の設定です。関与するのがこの 2 つの表だけである場合は、レプリカで発生した変更をマスターで再キャプチャーする必要は無く、またマスターで発生したいかなる変更も単一レプリカで再キャプチャーする必要がありません。

マスターの相互に排他的なパーティションである複数のレプリカの場合: マスター: 変更をマスターで再キャプチャーしません

レプリカ: 変更をレプリカで再キャプチャーしません

マスター表のパーティションである複数のレプリカを持つ予定である場合は、マスター表および各レプリカ表の両方で変更が再キャプチャーされないようにすることができます。どのレプリカも他のレプリカ表にとってのソースではない場合は、これが最良の設定です。レプリカがマスターのパーティションである場合は、複数のレプリカがマスターの同じデータにサブスクライブすることがあってはなりません。したがって、どのレプリカで発生したどの変更もマスターで再キャプチャーする必要は無く、他のレプリカに転送する必要もありません。なぜなら、そのソース・データにサブスクライブしているのはその変更が発生したレプリカだけだからです。

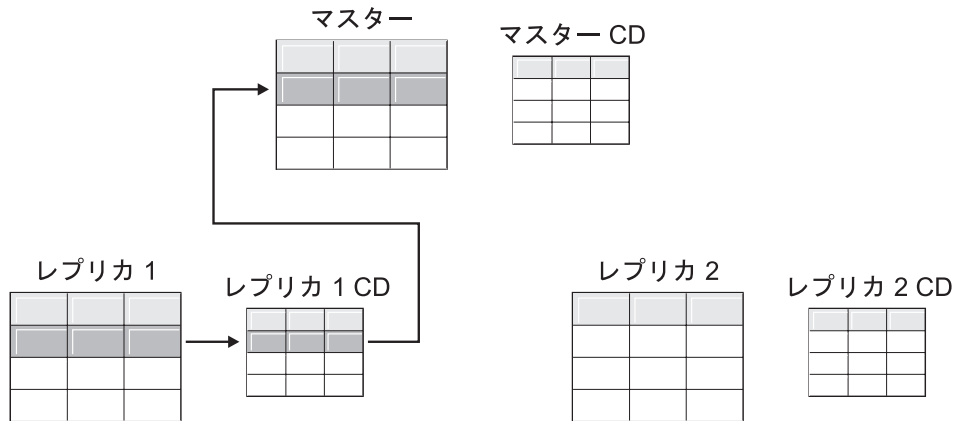


図6. マスターの相互に排他的なパーティションであるレプリカ用の再キャプチャー・オプション：マスターの同じデータにサブスクライブしていない複数のレプリカを持っている場合は、どの表に対してもこの再キャプチャー・オプションを使用する必要はありません。

変更を複数のレプリカに複製するマスターの場合： マスター：変更をマスターで再キャプチャーします

レプリカ：変更をレプリカで再キャプチャーしません

マスター表の同じデータにサブスクライブする複数のレプリカを持つ予定である場合は、キャプチャー・プログラムに変更をマスターで再キャプチャーさせる必要がある場合があります。そのようにすると、レプリカで発生した変更はマスターで再キャプチャーされて、その更新されたマスター・データにサブスクライブしている他のレプリカへと複製されます。

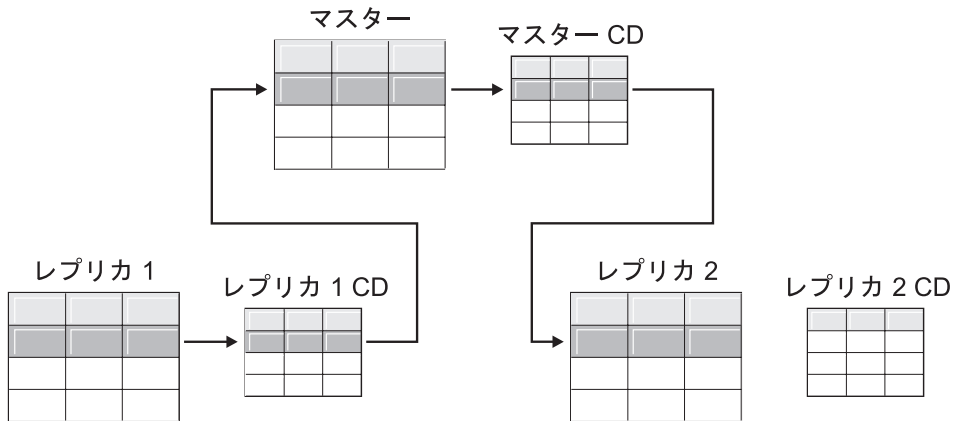


図7. 変更を複数のレプリカに複製するマスター用の再キャプチャー・オプション：マスターの同じデータにサブスクライブしている複数のレプリカを持っている場合は、マスターで再キャプチャー・オプションを使用して、あるレプリカで発生した変更がマスターで再キャプチャーされて他のレプリカ表に転送されるようになります。

他のレプリカに変更を複製するレプリカの場合 (multi-tier): マスター：変更をマスターで再キャプチャーしません

レプリカ： 変更をレプリカで再キャプチャーします

マスター (層 1) があるレプリカ (層 2) に対してソースとして機能し、次にそのレプリカが同様に別のレプリカ (層 3) に対してソースとして機能する multi-tier 構成を持てます。このタイプの構成を持つ予定である場合は、キャプチャー・プログラムに中間のレプリカ (層 2) で変更を再キャプチャーさせて、マスターで発生した変更がその次のレプリカ (層 3) に転送されるようにしたい場合があります。

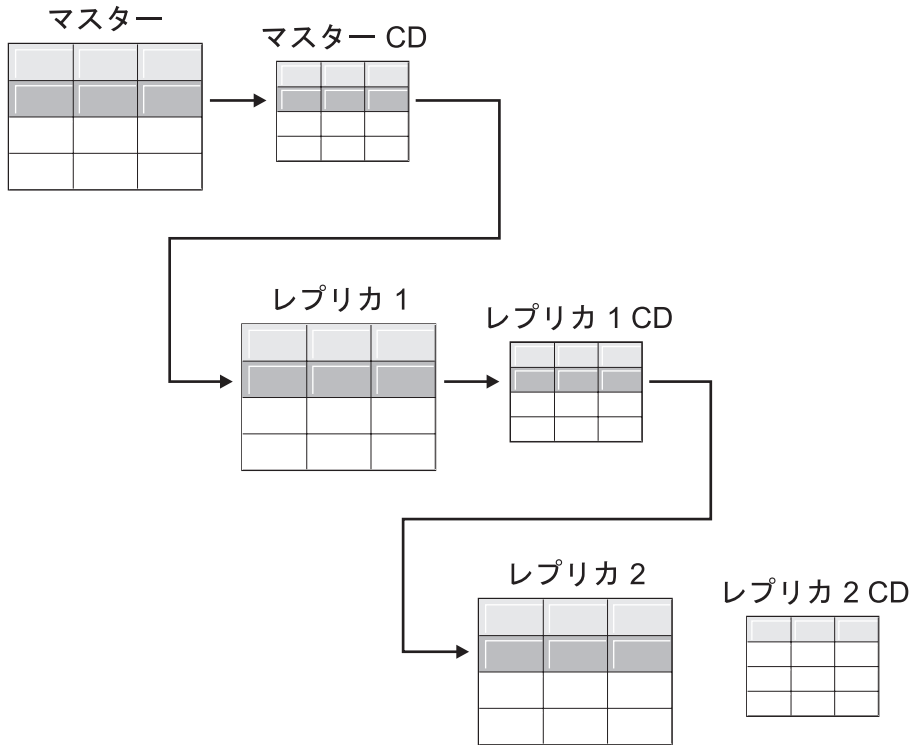


図8. 層 2 で再キャプチャー・オプションを使用すると、層 1 での変更が層 3 にまで複製されるようにできます： multi-tier 構成で中間層として機能するレプリカ表がある場合は、そのレプリカで再キャプチャー・オプションを使用して、マスターで発生した変更が中間層のレプリカで再キャプチャーされて下層のレプリカに転送されるようにできます。

また、中間のレプリカ (層 2) に再キャプチャーを設定してある場合は、最終のレプリカ (層 3) で発生した変更が中間のレプリカ (層 2) で再キャプチャーされてマスター (層 1) に転送されます。

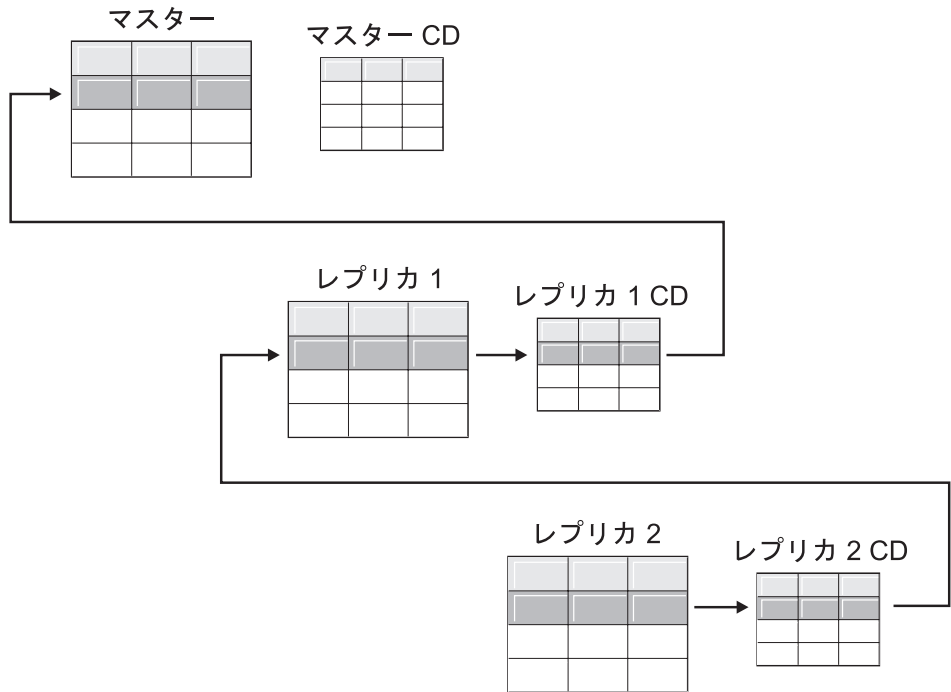


図9. 層 2 で再キャプチャー・オプションを使用すると、層 3 での変更が層 1 にまで複製されるようにできます： multi-tier 構成で中間層として機能するレプリカ表がある場合は、そのレプリカで再キャプチャー・オプションを使用して、下層のレプリカで発生した変更が中間層のレプリカで再キャプチャーされてマスターに転送されるようにできます。

競合検出の設定 (update-anywhere レプリケーション): デフォルト: 競合検出なし

制約事項:

- 非 DB2 リレーショナル・データベースからの表は update-anywhere にかかわれない。したがって、非 DB2 リレーショナル・ソースでは競合検出は行われません。
- DATALINK 列を含む update-anywhere 構成がある場合は、競合検出レベルに None を指定する必要がある。DB2 は DATALINK 列が参照する外部ファイルの更新競合はチェックしません。
- LOB 列を含む update-anywhere 構成がある場合は、競合検出レベルに None を指定する必要がある。LOB データ・タイプの列は、update-anywhere レプリケーションの対象になりません。

update-anywhere 構成では、マスターとそのレプリカの間で時々競合が発生する可能性があります。以下に、競合が起こる可能性のある場合を示します。

- マスター表である行が更新されたが、1 つ以上のレプリカ表で同じ行にそれとは異なる更新が加えられ、さらにアプライ・プログラムがそれらの競合する変更を同一サイクルの間に処理した。
- 制約に違反した。

競合検出レベルを個々のレプリケーション・ソースに設定した場合でも、アプライ・プログラムはサブスクリプション・セットのすべてのメンバーのレベルとして、サブスクリプション・セット・メンバーの競合検出レベルのうち最も高いものを使用します。

DB2 レプリケーションでは、3 つのレベルの競合検出 (検出なし、標準検出、および拡張検出) が提供されます。逸失またはリジェクトされたトランザクションの許容度とパフォーマンス要件に基づいて、どのタイプの検出を使用するかを決定できます。

update-anywhere レプリケーションに使用する予定のソースの登録時に、以下の競合検出レベルの中からいずれかを選択できます。

なし 競合検出なし。マスター表とレプリカ表の間で競合している更新は検出されません。このオプションは、update-anywhere レプリケーションではお勧めしません。

標準 適度な競合検出。

各アプライ・サイクルの間に、アプライ・プログラムはマスターの CD 表のキー値とレプリカの CD 表のキー値を比較します。両方の CD 表に同じキー値が存在している場合、それは競合です。競合が発生すると、アプライ・プログラムは直前にレプリカでコミットされたトランザクションを取り消します。これは、レプリカの CD 表からの読み取りを実行し、マスターで発生した変更だけを保持するという方法で行われます。

拡張 マスターとそのレプリカとの間での最良のデータ保全性を実現する競合検出。

標準検出の場合と同様に、アプライ・プログラムは各アプライ・サイクルの間にマスターの CD 表のキー値とレプリカの CD 表のキー値を比較します。両方の CD 表に同じキー値が存在している場合、それは競合です。ただし拡張検出では、競合をチェックする前に、アプライ・プログラムはすべての未完了トランザクションがコミットされるまで待ちます。すべての未完了トランザクションを確実に捕らえられるように、アプライ・プログラムはこれから処理されるトランザクションに対してサブスクリプション・セット内のすべてのターゲット表をロックし、CD 表内のすべての変更が取り込まれた後で競合検出を開始します。競合が発生すると、アプライ・プログラムは直前にレプリカでコミットされたトランザクションを取り消します。これは、レプリカの CD からの読み取りを実行し、マスターで発生した変更だけを保持するという方法で行われます。

制約事項: 拡張競合検出を指定しても、アプライ・プログラムが不定期接続環境で実行されている (COPYONCE キーワードを使用して始動された) 場合は、アプライ・プログラムは標準競合検出を使用します。

アプライ・プログラムは、読み取り従属関係を検出することはできません。たとえば、後で (DELETE ステートメントによって、またはロールバック・トランザクションによって) 除去されることになる情報をアプリケーションが読み取る場合、アプライ・プログラムは従属関係を検出することはできません。

競合が発生し得るレプリケーション構成を (検出なしまたは標準検出のいずれかを選択して) セットアップする場合は、発生したあらゆる競合を識別して処理するための方法を組み込む必要があります。競合のあったトランザクション更新はレプリケーションのインフラストラクチャーによってすでに検出されてバックアウトされているとしても、アプリケーション・デザイナーは、いったんコミットされたが現在はバックアウト済みのトランザクションをどのように処理するのかを決定する必要があります。ASNDONE 出口ルーチンは各サブスクリプション・サイクルの最後で実行されるため、アプリケーション・デザイナーはこのルーチンを上記のようなアプリケーション・ユニークのロジックのためのランチ点として使用できます。バックアウト済みの競合する更新に関する情報は、保持制限プルーニングの対象として適格となるまで CD 表および UOW 表に残されます。

リモート・ジャーナリングを使用する表の登録 (OS/400): デフォルト: リモート・ジャーナルはソースとして使用されません

リモート・ジャーナリングを使用する OS/400 表を登録する場合、DB2 レプリケーションではレプリケーション・ソースとしてローカル・ジャーナルの代わりにリモート・ジャーナルを使用することを定義できます。レプリケーション用のリモート・ジャーナリング・オプションを選択して、CD 表、キャプチャー・プログラム、およびキャプチャー・コントロール表を、ソース表が置かれている OS/400 サーバーとは別個の OS/400 データベース・サーバーに移動してください。

OS/400 で表をソースとして登録する場合、デフォルトではリモート・ジャーナリングを使用しないことが前提とされます。

推奨: ある OS/400 表から別の OS/400 表にデータを複製する場合、セットアップ済みのリモート・ジャーナルがあるときは常に、登録時にリモート・ジャーナリング関数を使用することを強く推奨します。レプリケーションにリモート・ジャーナリングを使用すると、パフォーマンスが大幅に向上します。リモート・ジャーナル関数を使用すると、登録、キャプチャー・プログラム、およびキャプチャー・コントロール表をソース表が置かれているシステムから遠く離れたところに移動できるため、そのシステムにはより多くのリソースが使用可能な状態で残されます。これにより、プロセッサ使用量が削減され、ディスク・スペースが節約されます。また、ターゲット・サーバーにあるリモート・ジャーナルを使用する場合は CD 表がターゲット表と同じシステム上に存在するため、アプライ・プログラムは予備ファイルを使用せずに直接 CD 表からターゲット表に変更を適用できます。予備ファイルを使用しないことで、アプライ・プログラムによって使用されるリソースの量が削減されます。

提案: リモート・ジャーナルを使用する表をソースとして登録するのは、登録がレプリケーション・ターゲットと同じ OS/400 システム上にある場合だけにしてください。DB2 レプリケーションを使用すると、登録がターゲットと同じ OS/400 システム上になくてもリモート・ジャーナルをソースとして登録できますが、その場合にはジャーナルをターゲット・システムに置くことで得られるパフォーマンス上の利点を得られません。

リモート・ジャーナリングを使用する OS/400 表を登録する前に、使用するリモート・ジャーナルがアクティブ状態になっていることを確認してください。

リモート・ジャーナル関数についての詳細は、「*Backup and Recovery*」(SC41-5304) および「*OS/400 Remote Journal Function for High Availability and Data Replication*」(SG24-5189) を参照してください。

主キーの代わりに相対レコード番号 (RRN) を使用する (OS/400):

通常、あるソース用のターゲット表はそのソース内の主キー列と同じキー列を使用します。アプライ・プログラムはこのキー値を使用して、自分がどのデータをソースの CD 表からターゲットに複製したのかをトラッキングします。主キー、ユニーク索引、またはユニーク索引として使用できる列の組み合わせを含んでいない OS/400 表を登録する場合は、その表を相対レコード番号 (RRN) を使用して登録する必要があります。RRN を使用して複製することを選択すると、CD 表とターゲット表の両方に、それぞれの行ごとユニークの値を含む INTEGER 型の列 (IBMQSQ_RRN) が追加されます。この列に含まれているのは、ソース表の各行に対応する RRN です。

ソース表が再編成されないかぎり、RRN はソース表の行に対する主キーとして使用されます。ソース表が再編成されると、ソース表の各行の RRN が変更されます。したがって、CD 表およびターゲット表の行に含まれている RRN の値は、当該の行のソース表内での新しい位置を反映した正しい値ではなくなります。ソース表を (たとえば、削除された行を圧縮するために) 再編成すると、必ず DB2 DataPropagator for iSeries はそのソース表のセットになっているすべてのターゲット表に対してフル・リフレッシュを実行します。この理由から、RRN を主キーとして使用するターゲット表をやはり RRN を使用する他のターゲットと一緒にサブスクリプション・セットに入れて、RRN 以外で一意性を表すならんらかの因子を使用する表とはセットにしないでください。

レプリケーション・ソースとしてのビューの動作

レプリケーション用のビューを登録すると、それらは登録オプションを基本表の登録定義から継承します。最も重要なことは、ビューが変更キャプチャー・レプリケーション用として登録されるか、あるいはフル・リフレッシュのみとして登録されるかが、ビューの基本表によって決定されることです。以下のセクションでは、登録されたビューがさまざまなシナリオのレプリケーションにおいてどのように動作するのが説明されています。

- 659 ページの『単一の表に対するビュー』

• 『複数の表の結合に対するビュー』

単一の表に対するビュー: 基本表がレプリケーション用に登録されている場合は、単一表に対するビューを登録できます。単一の登録済みの表に対するビューを登録すると、ビューはその基本表が持っているレプリケーション・タイプを継承します。基本表がフル・リフレッシュのみのレプリケーション用に登録されている場合は、ビューはフル・リフレッシュのみのレプリケーションを持ちます。そのビューを変更キャプチャー・レプリケーションに登録することはできません。なぜなら、その基本表は変更をトラッキングするための関連した CD 表を持っていないからです。基本表が変更キャプチャー・レプリケーション用に登録されている場合は、ビューは変更キャプチャー・レプリケーションを持つため、フル・リフレッシュのみのレプリケーション用に登録することはできません。

変更キャプチャー・レプリケーション用に登録されている表に対するビューを登録すると、基本表の CD 表に対するビューが作成されます。この CD ビューには、登録したビューから参照される列だけが含まれています。

ビュー内の列のサブセットは登録できません。なぜなら、ビュー内のすべての列が自動的に登録されるからです。

複数の表の結合に対するビュー: 複数の表の結合に対するビューを登録する場合は、その結合内の表が 1 つでも登録されているかぎり、基本表は登録済みの表であっても未登録の表であっても構いません。また、ソースとして登録されている CCD 表の内部結合も持てます。

結合をレプリケーション・ソースとして登録すると、DB2 レプリケーションは同一の SOURCE_OWNER 値と SOURCE_TABLE 値が含まれている複数の行を登録 (IBMSNAP_REGISTER) 表に追加します。これらの行は、各自の SOURCE_VIEW_QUAL 値によって区別されます。これらの各エントリーによって結合のコンポーネントが識別されます。

制約事項: CCD 表が組み込まれた結合を定義する場合は、その結合内の他のすべての表が CCD 表である必要があります。

ある結合ビューがレプリケーション・ソースとして存続できるようにするためには、その結合ビューを相関 ID を使用して作成する必要があります (単一表に対するビューの場合には相関 ID は不要です。)

例:

```
create view REGRES1.VW000 (c000,c1001,c2001,c2002,c1003) as
  select a.c000,a.c001,b.c001,b.c002,a.c003
  from REGRES1.SRC001 a, REGRES1.SRC005 b
  where a.c000=b.c000;
```

ここで、VW000 はビューの名前です。 SRC001 および SRC005 は、ビューの一部である表です。C000、C001、C002、および C003 は、両方の表 (SRC001 と SRC005) の C000 列が等しいという条件の下でビューの一部となる列です。

ビューが継承するレプリケーションのタイプは、そのビューの基本表の組み合わせによって決まります。各基本表は以下のいずれかです。

- 変更キャプチャー・レプリケーション用に登録されているもの
- フル・リフレッシュのみのレプリケーション用に登録されているもの
- 未登録のもの

表 169 には、基本表のさまざまな組み合わせと、それぞれの組み合わせからソース・ビューと CD ビューが結果的にどのタイプになるのかが示されています。

表 169. ビューの場合の基本表の組み合わせ

表 1	表 2	結合ビューおよび CD ビューの記述
変更キャプチャー用に登録されているもの	変更キャプチャー用に登録されているもの	このビューは、変更キャプチャー・レプリケーション用に登録されたものである。この CD ビューには、表 1 の CD 表と表 2 の CD 表から参照される列が含まれている。
変更キャプチャー用に登録されているもの	フル・リフレッシュのみ用に登録されているもの	このビューは、変更キャプチャー・レプリケーション用に登録されたものである。この CD ビューには、表 1 の CD 表から参照される列と、表 2 から参照される列が含まれている。各レプリケーション・サイクルでは、表 1 に入っている列に対する変更だけが登録済みビューのターゲットにレプリケーションされる。
フル・リフレッシュのみ用に登録されているもの	フル・リフレッシュのみ用に登録されているもの	このビューは、フル・リフレッシュのみのレプリケーション用に登録されたものである。CD ビューはない。
フル・リフレッシュのみ用に登録されているもの	未登録のもの	このビューは、フル・リフレッシュのみのレプリケーション用に登録されたものである。CD ビューはない。
変更キャプチャー用に登録されているもの	未登録のもの	このビューは、変更キャプチャー・レプリケーション用に登録されたものである。この CD ビューには、表 1 の CD 表から参照される列と、表 2 から参照される列が含まれている。各レプリケーション・サイクルでは、表 1 に入っている列に対する変更だけが登録済みビューのターゲットにレプリケーションされる。
未登録のもの	未登録のもの	このビューは有効なレプリケーション・ソースではないため登録できない。

複数のソース表がレプリケーション・ソースとして組み込まれているビューを定義する場合は、二重削除を防ぐための配慮が必要です。二重削除は、同一レプリケーション・サイクルの間に両方とも 1 つのビューの一部である表から行を削除した場合に発生します。たとえば、CUSTOMERS 表と CONTRACTS 表を含むビューを作成したとします。二重削除は、同一のレプリケーション・サイクルで CUSTOMERS 表から 1 つの行を削除し、同様に (ビューの結合点から) それに対応する行を CONTRACTS 表からも削除した場合に発生します。ここで問題となるのは、その行が結合の 2 つのソース表から削除されているために、その行はビューに (基本ビューにも CD 表ビューにも) 表示されず、したがってこの二重削除がターゲットに複製できないことです。

二重削除を防ぐためには、結合内のソース表のいずれか 1 つに CCD 表を定義する必要があります。この CCD 表は、コンデンスされた非コンプリートの表でなければならず、さらにターゲット・サーバー上になければなりません。結合内のいずれかのソース表に対し、コンデンスされた非コンプリートの CCD 表を定義すれば、ほとんどの場合の二重削除問題は解決されます。なぜなら、この CCD 表の IBMSNAP_OPERATION 列を使用すれば削除を検出できるからです。サブスクリプション・サイクルの後 に実行するサブスクリプション・セットの定義に、単に SQL ステートメントを追加してくださいこの SQL ステートメントは、CCD 表内の IBMSNAP_OPERATION が “D” と等しくなるターゲット表からすべての行を除去します。

同一アプライ・サイクルで、CCD を持つソース表のある行が更新されたが、結合内のもう一方の表ではそれに対応する行が削除されたという場合には、更新と削除に関する問題がまだ発生する可能性があります。その結果、アプライ・プログラムは結合された表でその対応する行を検出できず、更新された値を複製できません。

表のビューをソースとして登録する

このセクションでは、DB2 表のビューをレプリケーション・ソースとして登録する方法を説明します。

前提条件:

- ソースとして登録するビューを処理させるキャプチャー・コントロール・サーバー上に、キャプチャー・コントロール表がすでに存在していなければならない。キャプチャー・コントロール表を作成する必要がある場合は、673 ページの『レプリケーション・コントロール表のセットアップ』を参照してください。
- ソース・ビューの名前は DB2 表の命名規則に従う必要がある。

制約事項:

- 非 DB2 リレーショナル表のビューは登録できない。
- 別のビューの上層のビューは登録できない。
- OS/400 では、SQL ステートメントは長さが 32,000 文字までに制限されているため、1 つのビュー当たりおよそ 2000 列までしか登録できない。正確な列数は、列名の長さによって決まります。

- ビューを定義されているすべての CCD 表は、レプリケーション・ソースとして登録するためにはコンプリートでコンデンスされたものでなければならない。

推奨: ビューをソースとして登録する前に、基本表のうち少なくとも 1 つをソースとして登録してください。表の登録方法は、638 ページの『DB2 表をソースとして登録する』を参照してください。

手順:

以下の方法のいずれかを使用して、ビューを登録してください。

レプリケーション・センター

「ビューの登録」ウィンドウを使用します。詳細については、レプリケーション・センターのヘルプを参照してください。

レプリケーション用のシステム・コマンド (OS/400)

ADDDPRREG システム・コマンドを使用します。このコマンドの構文およびパラメーターについての記述は、570 ページの『ADDDPRREG: DPR 登録の追加 (OS/400)』を参照してください。

ビューの登録時には、ビューの名前、ソース・サーバー、およびキャプチャー・スキーマを指定して、どのビューを登録するのかを示してください。ビュー用の登録オプションは、それらのビューが定義されているソース表の登録定義から導き出されます。登録オプションとそれらのデフォルトの完全なリスト、およびこれらのデフォルトを使用した方がよい場合と変更した方がよい場合についての説明は、641 ページの『ソース表用の登録オプション』を参照してください。基本表に基づいてビューがどのレプリケーション・タイプ (変更キャプチャーまたはフル・リフレッシュのみ) を継承するのかについては、658 ページの『レプリケーション・ソースとしてのビューの動作』を参照してください。

CCD 表をソースとして保守する (IMS)

アプライ・プログラムによって移植されたのではなく外部的に移植された CCD 表、または IMS DataPropagator や DataRefresher などのプログラムによって保守されている CCD 表がある場合は、アプライ・プログラムがそれらの CCD 表をソースとして読み取れるように、また正しく機能できるように、それらの表を保守する必要があります。このセクションでは、CCD 表をレプリケーション・ソースとして保守する方法を説明します。

外部ツールによって移植された CCD 表を保守するためには、登録 (IBMSNAP_REGISTER) 表の 3 つの列 (CCD_OLD_SYNCHPOINT、SYNCHPOINT、および SYNCHTIME) を更新する必要があります。これらの 3 つの列は、以下のタイプのイベントが発生するたびに更新する必要があります。

- はじめてフル・リフレッシュまたは CCD 表のロードが行われたとき
 - CCD_OLD_SYNCHPOINT を CCD 表の IBMSNAP_COMMITSEQ の最小値を表す値に設定する。

- SYNCHPOINT を CCD 表の IBMSNAP_COMMITSEQ の最大値を表す値に設定する。SYNCHPOINT を 0 に設定しないでください。順序付け用に独自の値を作成する場合は、初回用の SYNCHPOINT 値は 1 にしてください。
- SYNCHTIME を CCD 表の IBMSNAP_LOGMARKER の最大タイム・スタンプ値を表す値に設定する。
- フル・リフレッシュ後またはロード後に CCD 表に対してリフレッシュが行われたとき
 - CCD_OLD_SYNCHPOINT 値を変更してはならない。
 - SYNCHPOINT を CCD 表の IBMSNAP_COMMITSEQ の新しい最大値を表す値に設定する。
 - SYNCHTIME を CCD 表の IBMSNAP_LOGMARKER の新しい最大タイム・スタンプ値を表す値に設定する。
- 2 回目以降のフル・リフレッシュまたは CCD 表のロードが行われたとき
 - CCD_OLD_SYNCHPOINT を CCD 表の IBMSNAP_COMMITSEQ の最小値を表す値に設定する。
 - SYNCHPOINT を CCD 表の IBMSNAP_COMMITSEQ の最大値を表す値に設定する。
 - SYNCHTIME を CCD 表の IBMSNAP_LOGMARKER の最大タイム・スタンプ値を表す値に設定する。

重要: 上記では、CCD 表で IBMSNAP_COMMITSEQ および IBMSNAP_LOGMARKER に使用されている値は常に増えてゆく値であることを前提としています。アプライ・プログラムは、CCD_OLD_SYNCHPOINT 値が最新に適用された SYNCHPOINT 値より大きくならないかぎり、ソース CCD 表でフル・リフレッシュが実行されたことを検出しません。

関連概念:

- レプリケーションのガイドおよびリファレンスの『DB2 レプリケーション・センターの使用』

関連タスク:

- 687 ページの『ソースのサブスクライブ』

関連資料:

- 570 ページの『ADDDPRREG: DPR 登録の追加 (OS/400)』

レプリケーションの設定

データを複製するためには、その前に環境をセットアップする必要があります。

この章は、以下のセクションを含んでいます。

- 664 ページの『レプリケーション・サーバーに対するアクセスのコントロール』

- 666 ページの『レプリケーション用のユーザー ID の許可』
- 672 ページの『レプリケーション用のユーザー ID およびパスワードの保管 (UNIX、Windows)』
- 673 ページの『レプリケーション・コントロール表のセットアップ』
- 676 ページの『レプリケーション・プログラムのセットアップ』
- 682 ページの『ジャーナルのセットアップ (OS/400)』

レプリケーション・サーバーに対するアクセスのコントロール

ほとんどのレプリケーション環境では、データは複数のサーバーに分散されています。このような環境を使用している場合は、レプリケーション・プログラムがすべてのサーバーに確実に接続できるようにしておく必要があります。サーバー間の接続を実現するには正しいソフトウェアがインストールされていなければならない、さらにサーバー間の接続を構成する必要があります。非 DB2 リレーショナル・データベースへのレプリケーションを行う場合は、連合サーバーおよび関連付けられた接続も構成する必要があります。

レプリケーションのための接続要件: アプライ・プログラム、レプリケーション・センター、またはレプリケーション・コマンドを実行するワークステーションは、必ずソース・サーバー、キャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、およびターゲット・サーバーのデータベースに接続できなければなりません。

レプリケーション・アラート・モニターを使用する場合、これを実行するワークステーションは、モニター・コントロール・サーバーとこのモニターがモニターするあらゆるサーバーに接続できなければなりません。レプリケーション・センターを使用してモニターをセットアップする場合は、レプリケーション・センターがモニター・コントロール・サーバーに接続できることを確認してください。

レプリケーションの設計で、ソース・データベースとは異なるサーバーでデータのステージングが関係しているなら、さまざまなサーバー間の通信について慎重に考慮する必要があります。エミュレーション層の数、LAN ブリッジの数、およびルーター・リンクの数については、どれもレプリケーションのパフォーマンスに影響を与えるものなので、必ず制限してください。

データベースがネットワークに接続される場合のコネクティビティーは、接続されるプラットフォームによって異なります。

非 DB2 リレーショナル・サーバーへの接続: 非 DB2 リレーショナル・サーバーとの間で相互にデータのレプリケーションを行う場合は、その非 DB2 リレーショナル・サーバーにアクセスして接続できなければなりません。

非 DB2 リレーショナル・ソース・サーバー からのレプリケーションを試行する前に、使用する連合サーバーと連合データベースをセットアップする必要があります。セットアップには以下の 3 つのメイン・ステップがあります。

1. ラッパーを定義して、DB2 データベースが他の非 DB2 リレーショナル・データベースにアクセスできるようにする。
2. サーバー・マッピングを使用して非 DB2 リレーショナル・データベースを定義する。
3. DB2 データベースへの接続に使用されるユーザー ID とパスワードの組み合わせが非 DB2 リレーショナル・データベースへのアクセスに使用されるものとは異なる場合は、ユーザー・マッピングを作成する必要がある。

「DB2 Federated Systems Guide」(GC27-1224) の説明にしたがって、使用する環境が正しく構成されていることを確認してください。

UNIX または Windows サーバーからの z/OS または iSeries サーバーへの接続

: すべてのリモート・サーバーに接続できることを確認してください。z/OS または OS/400 システムと Windows または UNIX システムとの間の接続を構成する方法は、「DB2 Connect Quick Beginnings」を参照してください。

手順 (iSeries 用):

DB2 for Windows ワークステーションから iSeries サーバーに接続する場合

1. DB2 for Windows クライアントから iSeries サーバーに接続する前に、使用するワークステーションが正しくセットアップされていることを確認する。
 - DB2 Universal Database または DB2 クライアント・アプリケーション・イネーブラー (CAE) クライアントがワークステーションにインストールされていなければならない。
 - TCP/IP がワークステーションにセットアップされている必要があること。
2. iSeries サーバーにログオンして、リレーショナル・データベースを探します。
 - a. 接続先となる iSeries サーバーにログオンします。
 - b. **dsprdbdire** コマンドをサブミットしてから、*LOCAL に local と指定します。
 - c. 出力の中から、リレーショナル・データベースの名前を探します。たとえば、以下の出力では、データベースの名前は DB2400E です。

MYDBOS2	9.112.14.67
RCHASDPD	RCHASDPD
DB2400E	*LOCAL
RCHASLJN	RCHASLJN

3. OS/400 データベースを DB2 for Windows のカタログに入れる。

- a. Windows NT ワークステーションで、「スタート」 → 「プログラム」 → 「IBM DB2」 → 「コマンド・ウィンドウ (Command Window)」 をクリックします。DB2 CLP コマンド・ウィンドウが開きます。
- b. コマンド・ウィンドウで、以下の 3 つのコマンドを正確な順序で入力します。


```
db2 catalog tcpip node server_name remote server_name server 446 system
server_name ostype OS400

db2 catalog dcs database rdb_name AS rdb_name

db2 catalog database rdb_name AS rdb_name at node server_name
authentication dcs
```

ここで、*server_name* は iSeries システムの TCP/IP ホスト名で、*rdb_name* は 2 (665 ページ) で説明されている iSeries リレーショナル・データベースの名前です。
4. コマンド・ウィンドウで、以下のコマンドを実行します。


```
db2 terminate
```
5. iSeries システムにログオンするために使用する iSeries ユーザー・プロファイルが CCSID37 を使用していることを確認します。
 - a. iSeries システムにログオンします。
 - b. 以下のコマンドを入力します。ここで、*user* はユーザー・プロファイルです。


```
CHGUSRPRF USRPRF (user) CCSID(37)
```
 - c. 以下のように入力して、DDM サーバーが iSeries システムで始動していることを確認します。


```
STRTCPSVR SERVER(*DDM)
```
6. DB2 for Windows NT および DB2 for iSeries が接続されていることを確認する。


```
db2 connect to rdb_name user user_name using password
```

レプリケーション用のユーザー ID の許可

DB2 および非 DB2 のリレーショナル・サーバーに置かれているデータにアクセスする必要がある場合は、以下の許可要件が満たされていることを確認してください。

- 『管理の許可要件』
- 669 ページの『キャプチャー・プログラムの許可要件』
- 670 ページの『非 DB2 リレーショナル・データベースでのキャプチャー・トリガーの場合の許可要件』
- 670 ページの『アプライ・プログラムの許可要件』
- 672 ページの『レプリケーション・アラート・モニターの場合の許可要件』

管理の許可要件: レプリケーション・センターを使用してレプリケーションを管理します。レプリケーション環境が OS/400 オペレーティング・システム上にのみある場合は、OS/400 システム・コマンドを使用してレプリケーションを管理できます。レプリケ

ーションを管理するためには、レプリケーション構成に関与するすべてのデータベースに対するユーザー ID を少なくとも 1 つは持っている必要があります、さらにそのユーザー ID にはレプリケーションをセットアップする権限がなければなりません。使用するユーザー ID は、すべてのシステムで同じである必要はありません。ただし、同じであった方が楽です。レプリケーションのセットアップには、オブジェクト (コントロール表や表スペースなど) の作成、プランのバインディング (UNIX、Windows、および z/OS の場合)、SQL パッケージの作成 (OS/400 の場合)、ならびに表、登録、およびサブスクリプション・セットを作成するための生成済み SQL の実行が含まれます。使用するレプリケーション環境内のすべてのサーバーに対して同一の許可ユーザー ID を使用するか、あるいはそれぞれのサーバーごとに別々の許可ユーザー ID を使用するかは任意です。

UNIX、Windows、および z/OS の場合の要件

レプリケーションのセットアップに使用するユーザー ID で以下のタスクを実行できることを確認してください。

- すべてのサーバー (ソース・サーバー、キャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、モニター・コントロール・サーバー、およびターゲット・サーバー) への接続
- ソース・サーバー、キャプチャー・コントロール・サーバー、モニター・コントロール・サーバー、およびターゲット・サーバーに置かれているカタログ表からの選択
- ソース・サーバー、モニター・コントロール・サーバー、キャプチャー・コントロール・サーバー、およびアプライ・コントロール・サーバーでの表 (レプリケーション・コントロール表も含む)、表スペース、およびビューの作成
- ターゲット・サーバーでの表および表スペースの作成 (新しいターゲット表の作成に DB2 レプリケーション・プログラムを使用する場合) (既存の表をターゲットとして使用する場合は必須ではありません)
- レプリケーションに関与するそれぞれの DB2 データベース (ソース・サーバー、ターゲット・サーバー、モニター・コントロール・サーバー、およびアプライ・コントロール・サーバーも含む) でのプランのバインドまたはパッケージの作成
- 共有ライブラリーを使用したストアド・プロシージャの作成およびストアド・プロシージャの呼び出し (UNIX と Windows のみ)

非 DB2 リレーショナル・データベースの場合は、使用するユーザー ID で以下のアクションを実行できなければなりません。

- 表を作成します。
- ソース表およびコントロール表に対するキャプチャー・トリガーの作成
- プロシージャの作成
- DB2 連合データベースでのニックネームの作成

- シーケンスの作成 (Oracle データベースの場合のみ)
- カタログ表からの選択

ほとんどのレプリケーション管理者は DBADM 特権または SYSADM 特権を持っています。DB2 for z/OS の場合、レプリケーション管理者は、少なくともカタログからの選択を許可されている必要があり、さらに索引作成特権も含め、ASN スキーマを使用した表の作成、およびソース表の特性を持つ CD 表とターゲット表の作成を行うために必要なすべての特権を持っている必要があります。

OS/400 の場合の要件

レプリケーションのセットアップに使用するユーザー ID で以下のタスクを実行できることを確認してください。

- すべてのサーバー (ソース・サーバー、キャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、モニター・コントロール・サーバー、およびターゲット・サーバー) への接続
- ソース・サーバー、キャプチャー・コントロール・サーバー、モニター・コントロール・サーバー、およびターゲット・サーバーに置かれているカタログ表からの選択
- ソース・サーバー、モニター・コントロール・サーバー、キャプチャー・コントロール・サーバー、およびアプライ・コントロール・サーバーでの表 (レプリケーション・コントロール表も含む) とビューの作成
- ターゲット・サーバーでの表の作成 (新しいターゲット表の作成に DB2 レプリケーション・プログラムを使用する場合) (既存の表をターゲットとして使用する場合は必須ではありません)
- レプリケーションに関与するそれぞれの DB2 データベース (ソース・サーバー、ターゲット・サーバー、モニター・コントロール・サーバー、およびアプライ・コントロール・サーバーも含む) でのプランのバインドまたはパッケージの作成

ほとんどのレプリケーション管理者は DBADM 特権または SYSADM 特権を持っています。

DPR 権限付与 (**GRTDPRAUT**) コマンドは、ユーザーがソースの登録、それらのソースに対するサブスクライブ、およびコントロール表の作成を行うのを許可するために使用します。OS/400 システム間でのみレプリケーションを行う場合は、すべてのサーバーに対して同じユーザー ID を使用する必要があります。

DPR 権限付与 (**GRTDPRAUT**) コマンドがマシンにインストールされていない場合は、オブジェクト権限付与 (**GRTOBJAUT**) コマンドを使用する必要があります。

キャプチャー・プログラムの許可要件: キャプチャー・プログラムを実行するユーザー ID は、DB2 システム・カタログにアクセスできなければなりません。また、キャプチャー・コントロール・サーバー上のすべてのレプリケーション・コントロール表へのアクセスと更新が可能で、さらにキャプチャー・プログラム・パッケージを実行できなければなりません。レプリケーション管理者のユーザー ID を使用してキャプチャー・プログラムを実行できますが、これは要件ではありません。

UNIX および Windows の場合の要件

キャプチャー・プログラムを実行するユーザー ID が以下の権限および特権を持っていることを確認してください。

- DBADM または SYSADM 権限。
- キャプチャー・パス・ディレクトリーに対する WRITE 特権。これは、キャプチャー・プログラムはキャプチャー・プログラムの始動時に指定された **capture_path** ディレクトリーに診断ファイルを作成するためです。

z/OS の場合の要件

キャプチャー・プログラムを実行するために使用するユーザー ID は、USS にアクセスできるものとして登録する必要があります。

また、キャプチャー・ロード・ライブラリーに APF 許可が与えられていること、およびキャプチャー・プログラムを実行するユーザー ID が以下の権限および特権を持っていることも確認してください。

- キャプチャー CONTROL サーバー上のすべてのレプリケーション表に対する SELECT、UPDATE、INSERT、および DELETE 特権。(これらの表のリストについては、512 ページの『キャプチャー・コントロール・サーバーで使用される表のリスト』を参照。)
- DB2 カタログ (SYSIBM.SYSTABLES および SYSIBM.SYSCOLUMNS) に対する SELECT 特権。
- TRACE 特権。
- MONITOR1 および MONITOR2 特権。
- キャプチャー・プログラム・パッケージに対する EXECUTE 特権。

また、そのユーザー ID にキャプチャー・パス・ディレクトリーに対する WRITE アクセス権があること (USS)、または高位修飾子が付けられていること (z/OS) を確認してください。キャプチャー・プログラムを USS シェルで実行するためには、STEPLIB システム変数が設定されていて、さらにこの変数にキャプチャー・ロード・ライブラリーが組み込まれている必要があります。PATH には HFS パス (/usr/lpp/db2rep1_08_01/bin) が含まれていなければなりません。

OS/400 の場合の要件

DPR 権限付与 (GRTPRAUT) コマンドは、ユーザーにローカル・システムでのキャプチャー・プログラムの実行を許可するために使用します。OS/400 シ

システム間でのみレプリケーションを行う場合は、すべてのサーバーに対して同じユーザー ID を使用する必要があります。GRTDPRAUT コマンドがマシンにインストールされていない場合は、オブジェクト権限付与 (GRTOBJAUT) コマンドを使用する必要があります。

非 DB2 リレーショナル・データベースでのキャプチャー・トリガーの場合の

許可要件: 非 DB2 RDBMS から複製する場合は、ソースからの変更のキャプチャーにはキャプチャー・トリガーが使用されます。リモート・ソース表を変更するリモート・ユーザー ID (たとえば、ユーザー・アプリケーションからの) には、CCD 表への挿入を行う権限が必要です。ほとんどの場合、INSERT、UPDATE、または DELETE トリガーを実行するのに明示的な権限は必要ありません。なぜなら、表に対してトリガーが定義されると、それ以降トリガーは INSERT、UPDATE、または DELETE を実行中のアプリケーションにとって透過的に実行されるからです。Informix データベースの場合は、登録済みのソース表に対して INSERT、UPDATE、および DELETE アクションを実行するリモート・ユーザー ID には EXECUTE PROCEDURE 特権が必要です。

アプライ・プログラムの許可要件: アプライ・プログラムを始動および操作するユーザー ID でプログラムの実行とパスワード・ファイルの読み取りができ、さらにそのユーザー ID がアプライ・パス・ディレクトリーに対する WRITE 特権を持っていることを確認してください。また、パスワード・ファイル内のユーザー ID は他のサーバーへの接続に使用されるため、それらのユーザー ID には以下のタスクを実行できるだけの十分な特権が与えられていなければなりません。

z/OS の場合の要件

アプライ・プログラムを実行するユーザー ID が以下の権限および特権を持っていることを確認してください。

- アプライ・プログラムを実行するために使用するユーザー ID は、USS にアクセスできるものとして登録する必要があります。ロード・ライブラリーに APF 許可が必要となるのは、アプライ・プログラムが ARM の指定付きで登録される場合だけです。アプライ・プログラムを USS シェルで実行するためには、STEPLIB システム変数が設定されていて、さらにこの変数にアプライ・ロード・ライブラリーが組み込まれている必要があります。PATH には HFS パス (/usr/lpp/db2repl_08_01/bin) が含まれていなければなりません。
- キャプチャー・コントロール・サーバー、ターゲット・サーバー、およびアプライ・コントロール・サーバーでのアプライ・プログラム・プランの実行

UNIX および Windows の場合の要件

アプライ・プログラムを実行するユーザー ID が以下の権限および特権を持っていることを確認してください。

- レプリケーション・ソース表 (関連した CD 表および CCD 表も含む) へのアクセス
 使用するソース表が非 DB2 リレーショナル・データベース管理システム上にある場合は、ユーザー ID は DB2 連合データベースおよび非 DB2 リレーショナル・データベースの両方において、連合データベースで定義されているニックネームを使用してソース表にアクセスできるだけの十分な特権を持っている必要があります。
- レプリケーション・ターゲット表に対するアクセスおよび更新
 アプライ・ユーザー ID は、ターゲット表に対する UPDATE 特権を持っている必要があります。
 ターゲット表が非 DB2 RDBMS 上にある場合は、ユーザー ID は DB2 連合データベースおよび非 DB2 リレーショナル・データベースにおいて、連合データベースで定義されているニックネームを使用してターゲット表にアクセスし、それらを更新できるだけの十分な特権を持っている必要があります。
- DB2 レプリケーション・プログラムによって生成されて、キャプチャー・コントロール・サーバーおよびアプライ・コントロール・サーバーでビルドされたすべてのコントロール表に対するアクセスと更新
- アプライ・プログラムによって使用される任意のパスワード・ファイルの読み取り

非 DB2 リレーショナル・データベース管理システムの場合の要件

コントロール表が非 DB2 リレーショナル・データベース管理システム上にある場合は、変更されたデータを非 DB2 リレーショナル・ターゲットにプッシュしたりそこからデータをプルするユーザー ID は、DB2 連合データベースおよび非 DB2 リレーショナル・データベースにおいて十分な特権を持っている必要があります。

非 DB2 リレーショナル・ターゲットの場合、アプライ・プログラムを実行中のユーザー ID には、DB2 連合データベース上のニックネームに対して WRITE を実行できる特権と、ユーザー・マッピングを介して実際の非 DB2 ターゲットへの WRITE を実行できる特権が必要です。

非 DB2 リレーショナル・ソースの場合、アプライ・プログラムを実行中の ID には以下の特権が必要です。

- DB2 連合データベース上のニックネームに対して READ と WRITE を実行できる特権、およびユーザー・マッピングを介してキャプチャー CONTROL 表に対し READ と WRITE を実行できる特権
- DB2 連合データベース上のニックネームからの READ を実行できる特権、およびユーザー・マッピングを介して非 DB2 サーバー上の実際の CCD 表からの READ を実行できる特権

- DB2 連合データベース上のニックネームからの READ を実行できる特権、およびユーザー・マッピングを介して非 DB2 サーバー上の実際のソース表からの READ を実行できる特権

OS/400 の場合の要件

DPR 権限付与 (**GRTDPRAUT**) コマンドは、ユーザーにローカル・システムでのアプライ・プログラムの実行を許可するために使用します。OS/400 システム間でのみレプリケーションを行う場合は、すべてのサーバーに対して同じユーザー ID を使用する必要があります。**GRTDPRAUT** コマンドがマシンにインストールされていない場合は、オブジェクト権限付与 (**GRTOBJAUT**) コマンドを使用する必要があります。

使用するレプリケーション環境内のそれぞれのサーバーごとに異なるユーザー ID を使用できます。

レプリケーション・アラート・モニターの場合の許可要件:

UNIX および Windows の場合の要件

レプリケーション・アラート・モニターを始動するユーザー ID が、モニター・コントロール表が置かれているモニター・コントロール・サーバー、およびモニターしているコントロール表を含んでいるサーバーでの有効なログオン ID であることを確認してください。また、レプリケーション・アラート・モニターを実行するユーザー ID が以下の権限を持っていることも確認してください。

- モニター CONTROL サーバー上のモニター CONTROL 表に対する SELECT、UPDATE、INSERT、および DELETE 特権
- モニターするサーバー上にあるキャプチャー・コントロール表およびアプライ・コントロール表に対する SELECT 権限
- BINDADD 権限 (モニター・パッケージに対して自動バインド機能を使用する場合にのみ必要)

UNIX および Windows では、レプリケーション・アラート・モニターが診断ファイルを保管する **monitor_path** ディレクトリーに対する WRITE 特権と、レプリケーション・アラート・モニターによって使用されるパスワード・ファイルに対する読み取りアクセスが必要です。

レプリケーション用のユーザー ID およびパスワードの保管 (UNIX、Windows)

レプリケーション環境が複数のサーバーに分散されていない場合は、ユーザー ID とパスワードを保管する必要はありません。しかし、ほとんどのレプリケーション環境では、データは複数のサーバーに分散されています。そのような環境がある場合は、データベースに接続しようとする時に有効なユーザー ID とパスワードを入力して、DB2 が

ユーザーの ID を検証できるようにする必要があります。レプリケーション・センターの場合とそれ以外のレプリケーション・プログラムの場合とは、パスワード情報の保管方法が異なります。

asnpwd コマンドは、アプライ・プログラム、レプリケーション・アラート・モニター、およびレプリケーション・アナライザーがリモート・サーバー上のデータにアクセスできるように、パスワード・ファイルを作成して保守するために使用します (キャプチャー・プログラムはパスワード・ファイルを必要としません。) パスワード・ファイル内の情報は、機密性を確保するために暗号化されます。

レプリケーション・センターの場合のパスワード要件は、レプリケーション・センターのヘルプを参照してください。

レプリケーション・コントロール表のセットアップ

レプリケーションに使用されるコントロール表を作成できます。

- 『コントロール表の作成 (UNIX、Windows)』
- 『コントロール表の作成 (z/OS)』
- 674 ページの『コントロール表の作成 (OS/400)』
- 674 ページの『非 DB2 リレーショナル・ソース用のコントロール表の作成』
- 675 ページの『キャプチャー・コントロール表のセットを複数作成する』

コントロール表の作成 (UNIX、Windows): レプリケーション・センターを使用して、UNIX および Windows でのキャプチャー・プログラム用とアプライ・プログラム用のレプリケーション・コントロール表を作成します。レプリケーション・コントロール表の作成時にコントロール表の作成方法をカスタマイズしない場合は、2 つの表スペース (1 つは UOW 表用、もう 1 つはその他のコントロール表用) が作成されます。それらのデフォルトのレプリケーション表スペースを使用したくない場合は、既存の表スペースを指定するか、新しい表スペースを作成するか、または現行の DB2 デフォルト表スペースを使用できます。

UNIX オペレーティング・システム用のプロファイルと Windows オペレーティング・システム用のプロファイルを別々に作成して、このタイプのシステム用のコントロール表を作成するときに、当該のタイプで使用すべきデフォルトが識別されるようにできます。それらのコントロール表に対してプロファイルを設定してしまえば、それ以降はコントロール表のセットを作成するたびにプロファイルを設定する必要がなくなります。ただし、それらのデフォルトはコントロール表の作成時にオーバーライドできます。また、プロファイルもいつでも変更できます。ただし、その変更の影響を受けるのは、プロファイルの変更後に作成したコントロール表だけです。

コントロール表の作成 (z/OS): レプリケーション・センターを使用して、z/OS でのレプリケーション・コントロール表を作成します。z/OS プラットフォーム用のプロファイルを作成して、このタイプのシステム用のコントロール表を作成するときに使用すべきデフォルトが識別されるようにできます。それらのコントロール表に対してプロ

ファイルを設定してしまえば、それ以降はコントロール表のセットを作成するたびにプロファイルを設定する必要がなくなります。ただし、それらのデフォルトはコントロール表の作成時にオーバーライドできます。また、プロファイルもいつでも変更できます。ただし、その変更の影響を受けるのは、プロファイルの変更後に作成したコントロール表だけです。

コントロール表の作成 (OS/400): レプリケーション・コントロール表は、DB2 DataPropagator for iSeries のインストール時に自動的に作成されます。それらの表は、DataPropagator (DPROPR) のデフォルト・スキーマ (ASN と呼ばれる) 内に作成されず (まだ存在していない場合)。

新しいキャプチャー・スキーマを使用してキャプチャー・コントロール表の新しいセットを作成できます。スキーマは、最大 25 個まで作成できます。675 ページの『キャプチャー・コントロール表のセットを複数作成する』の説明にしたがって、DPR 表の作成 (**CRTDPRTBL**) コマンドを使用してください。また、**CRTDPRTBL** コマンドは、レプリケーション・コントロール表が誤って削除または破壊された場合にも使用できます。

重要: OS/400 では、コントロール表は必ず **CRTDPRTBL** コマンドを使用して作成するようにしてください。レプリケーション・センターは OS/400 用のコントロール表の作成をサポートしていません。

ユーザー定義のファイル・システムでは、レプリケーション・コントロール表を基本補助記憶域プール (ASP) または独立補助記憶域プール (IASP) グループに作成できますが、両方に作成することはできません。コントロール表を IASP グループに作成する場合は、まず最初にすべてのキャプチャーおよびアプライ・コントロール表を基本 ASP から除去する必要があります。キャプチャーまたはアプライ・プログラムを開始する前に、ASN ライブラリー (またはキャプチャー・スキーマ用のライブラリー) を含む ASP グループに対して **SETASPGRP** コマンドを発行します。

制限: レプリケーション・センターは、レプリケーション・コントロール表が IASP に作成されるレプリケーション環境用の管理タスクをサポートしません。

非 DB2 リレーショナル・ソース用のコントロール表の作成: Informix のような非 DB2 RDBMS から のレプリケーションを行う場合は、DB2 からのレプリケーションの場合とまったく同じように、レプリケーション・センターを使用してコントロール表を作成する必要があります。このようなタイプのソースの場合、レプリケーション・センターは非 DB2 リレーショナル・データベースに以下のキャプチャー・コントロール表を作成します。

- プルーニング・コントロール表 (IBMSNAP_PRUNCNTL)
- プルーニング・セット表 (IBMSNAP_PRUNE_SET)
- 登録同期表 (IBMSNAP_REG_SYNCH)
- 登録表 (IBMSNAP_REGISTER)
- 順序付け表 (IBMSNAP_SEQTABLE) (Informix の場合のみ)

- シグナル表 (IBMSNAP_SIGNAL)

連合データベース内に順序付け表 (IBMSNAP_SEQTABLE) 以外のすべての表のニックネームが作成されます。(この順序付け表は、Informix トリガーだけが使用します。アプライ・プログラムはこれを使用しません。) シグナル表 (IBMSNAP_SIGNAL) および登録同期表 (IBMSNAP_REG_SYNCH) にトリガーが自動的に作成されます。

重要: IBMSNAP_SIGNAL 表および IBMSNAP_REG_SYNCH 表に作成されたトリガーは、除去または変更をしないでください。

キャプチャー・コントロール表のセットを複数作成する: 1 台のサーバー上で複数のキャプチャー・プログラムを使用する場合は、キャプチャー・コントロール表のセットを複数作成して、それぞれの表セットが必ずユニークのキャプチャー・スキーマを持つようにする必要があります。このスキーマにより、ある表のセットを使用するキャプチャー・プログラムが識別されます。複数のキャプチャー・スキーマを使用すると、複数のキャプチャー・プログラムを並行実行できます。

以下の場合には、複数のキャプチャー・プログラムを実行しなければならない可能性があります。

- 待ち時間が短い表は他の表とは別に処理するという方法でパフォーマンスを最適化する場合。待ち時間が短い表がある場合は、それらの表の複製はそれら専用のキャプチャー・プログラムを使用して行うのが望ましいことがあります。この方法を使用すると、それらの表には別のランタイム優先順位が与えられます。また、キャプチャー・プログラムのパラメーター (ブルーニング・インターバルやモニター・インターバルなど) を、それらの表の短い待ち時間に合うように設定することも可能です。
- 可能なときにはキャプチャー・スループットを上げる場合。これは、複数の CPU を持つソース環境では大いに益になる可能性があります。スループット増加のトレードオフは、ログ・リーダーが複数存在することに関連した CPU オーバーヘッドの増加です。

同一の連合データベース内で複数の非 DB2 ソース・データベースからのレプリケーションを実行する場合は、キャプチャー・コントロール表のセットを複数作成して、各セットに独自のスキーマを持たせる必要があります。あるいはその代わりに、複数の連合データベースを別個に使用しても構いません。そのようにすると、各サーバーのキャプチャー・コントロール表でデフォルトの ASN スキーマを使用できます。

z/OS システムでは、UNICODE コード化スキームと EBCDIC コード化スキームを別々に使用して作業したい場合、または 1 つのサブシステムでキャプチャー・プログラムのインスタンスを複数実行したい場合は、複数のキャプチャー・スキーマを使用できます。コントロール表の作成については、673 ページの『コントロール表の作成 (z/OS)』を参照してください。

OS/400 システムでは、DPR 表の作成 (**CRTPDPR**TBL) コマンドを使用してキャプチャー・コントロール表の追加のセットを作成します。その際には、**CAPCTLLIB** パラメーターを使用してスキーマ名を指定します。

レプリケーション・プログラムのセットアップ

以下のセクションでは、環境内のサーバーに対してレプリケーション・プログラムをセットアップするステップについて説明しています。

- 『レプリケーション・プログラムのセットアップ (UNIX、Windows)』
- 680 ページの『キャプチャー・プログラムおよびアプライ・プログラムのセットアップ (OS/400)』
- 682 ページの『レプリケーション・プログラムのセットアップ (z/OS)』

レプリケーション・プログラムのセットアップ (UNIX、Windows): 以下の説明を読んで、レプリケーション・プログラムをセットアップしてください。

- 『レプリケーション・プログラムのための環境変数の設定 (UNIX、Windows)』
- 677 ページの『キャプチャー・プログラムを実行するための DB2 データベースの準備 (UNIX、Windows)』
- 677 ページの『オプション : キャプチャー・プログラム・パッケージのバインディング (UNIX、Windows)』
- 678 ページの『オプション : アプライ・プログラム・パッケージのバインディング (UNIX、Windows)』
- 679 ページの『オプション : レプリケーション・アラート・モニター・プログラム・パッケージのバインディング (UNIX、Windows)』

レプリケーション・プログラムのための環境変数の設定 (UNIX、Windows):

キャプチャー・プログラム、アプライ・プログラム、またはレプリケーション・アラート・モニター・プログラムを始動する前と停止する前、およびレプリケーション・センターまたはレプリケーション・システム・コマンドを使用する前に、環境変数を設定する必要があります。

手順:

環境変数を設定するには、以下のようにします。

1. 以下のように入力して、DB2 インスタンス名 (DB2INSTANCE) の環境変数を設定します。

Windows の場合

```
SET DB2INSTANCE=db2_instance_name
```

UNIX の場合

```
export DB2INSTANCE=db2_instance_name
```


2. デフォルトのコード・ページ値以外のコード・ページを使用してソース・データベースを作成した場合は、DB2CODEPAGE 環境変数をそのコード・ページに設定する。

1

3. オプション: 環境変数 DB2DBDFT をソース・サーバーに設定します。
4. **UNIX の場合:** 使用しているシステムにユニークのライブラリー・パス・システム変数および実行可能モジュール・パス・システム変数に、レプリケーション用のライブラリーと実行可能モジュールがインストールされているディレクトリーが組み込まれていることを確認する。

キャプチャー・プログラムを実行するための DB2 データベースの準備 (UNIX、Windows): 手順:

キャプチャー・プログラムを実行するために DB2 データベースを準備するには、以下のようになります。

1. 以下のように入力して、キャプチャー・コントロール・サーバーのデータベースに接続する。

```
db2 connect to database
```

ここで、*database* はキャプチャー・コントロール・サーバーのデータベースです。

2. キャプチャー・コントロール・サーバーのデータベースを、ロールフォワード・リカバーのために準備する。これを行うには、**update database configuration** (データベース構成の更新) コマンド (ログ保存リカバー) と **backup database** (データベースのバックアップ) コマンドを発行します。インストール要件に基づいて構成値を大きくする必要がある場合があります。多数の行または非常に大きな行を使用するトランザクションの場合は、CAPPARMS メモリ限度パラメーターの値を大きくすることが推奨されます。多くの大規模ワークステーション・シナリオの場合、適切なデータベース構成値は以下のとおりです: APPLHEAPSZ 1000、LOGFILSIZ 4000、LOGPRIMARY 8、LOGSECOND 40、DBHEAP 1000、LOGBUFSZ 16、MAXAPPLS 200。

オプション: キャプチャー・プログラム・パッケージのバインディング

(UNIX、Windows): 以下のステップはオプションです。なぜなら、UNIX および Windows では、キャプチャー・プログラムは実行時に自動的にバインドされるからです。

手順:

-
1. キャプチャーは、データを取り込んでいるデータベースのコード・ページと同じコード・ページで実行する必要があります。DB2 は、キャプチャーが実行されているアクティブ環境からキャプチャー・コード・ページを導き出します。DB2CODEPAGE が設定されていない場合は、DB2 はコード・ページ値をオペレーティング・システムから取得します。データベースの作成時にデフォルトのコード・ページを使用したのであれば、オペレーティング・システムから取得した値はキャプチャーにとって正しいものです。

キャプチャー・プログラム・パッケージをバインドするには、以下のようにします。

1. 以下のように入力して、キャプチャー・コントロール・サーバーのデータベースに接続する。

```
db2 connect to database
```

ここで、*database* はキャプチャー・コントロール・サーバーのデータベースです。

2. キャプチャー・プログラムのバインド・ファイルがあるディレクトリーに変更します。

Windows:

```
drive:¥sql11ib¥bnd
```

UNIX: *db2homedir/sql11ib/bnd*

ここで、*db2homedir* は DB2 インスタンスのホーム・ディレクトリーです。

3. 以下のようなコマンドを入力して、キャプチャー・プログラム・パッケージを作成してソース・サーバー・データベースにバインドします。

```
db2 bind @capture.lst isolation ur blocking all
```

ただし *ur* は、パフォーマンスを向上するための非コミット読み取り形式のリストを指定します。

これらのコマンドは、パッケージを作成します。その名前は、*capture.lst* ファイル内で見つけることができます。

オプション：アプライ・プログラム・パッケージのバインディング

(UNIX、Windows): UNIX および Windows では、アプライ・プログラムは実行時に自動的にバインドされます。したがって、これらのオペレーティング・システムでは、以下のステップはオプションです。

手順:

アプライ・プログラム・パッケージをバインドするには、以下のようにします。

1. アプライ・プログラムのバインド・ファイルがあるディレクトリーに変更します。

Windows:

```
drive:¥sql11ib¥bnd
```

UNIX: *db2homedir/sql11ib/bnd*

ここで、*db2homedir* は DB2 インスタンスのホーム・ディレクトリーです。

2. アプライ・プログラムが接続するそれぞれのソース・サーバー、ターゲット・サーバー、キャプチャー・コントロール・サーバー、およびアプライ・コントロール・サーバーごとに、以下のステップを実行する。
 - a. 以下のように入力して、データベースに接続します。

db2 connect to *database*

ここで、*database* はソース・サーバー、ターゲット・サーバー、キャプチャー・コントロール・サーバー、またはアプライ・コントロール・サーバーです。データベースがリモート・データベースとしてカタログされている場合は、**db2 connect to** コマンドでユーザー ID とパスワードを指定する必要があります。たとえば、以下のように指定します。

db2 connect to *database* user *userid* using *password*

- b. 以下の 2 つのコマンドを両方入力して、Applyプログラム・パッケージを作成してデータベースにバインドします。

```
db2 bind @applycs.lst isolation cs blocking all grant public
```

```
db2 bind @applyur.lst isolation ur blocking all grant public
```

ただし、*cs* はカーソル固定形式でリストを指定するのに対し、*ur* は非コミット読み取り形式でリストを指定します。

これらのコマンドは、パッケージを作成します。その名前は、*applycs.lst* および *applyur.lst* ファイル内で見つけることができます。

オプション：レプリケーション・アラート・モニター・プログラム・パッケージのバインディング (UNIX、Windows): パッケージを BIND するための以下のステップはオプションにすぎません。レプリケーション・アラート・モニター・パッケージは、実行時に自動的にバインドされます。オプションを指定したい場合、またはすべての BIND 処理が正常に完了したことをチェックしたい場合は、以下のステップを実行してください。

手順:

レプリケーション・アラート・モニター・プログラム・パッケージをバインドするには、以下のようにします。

1. レプリケーション・アラート・モニター・プログラムのバインド・ファイルが配置されているディレクトリーに変更する。

Windows:

```
drive:¥sqllib¥bnd
```

UNIX: *db2homedir/sql/lib/bnd*

ここで、*db2homedir* は DB2 インスタンスのホーム・ディレクトリーです。

2. それぞれのモニター・コントロール・サーバーごとに以下のステップを実行する。
 - a. 以下のように入力して、モニター・コントロール・サーバーのデータベースに接続する。

```
db2 connect to database
```

ここで、*database* はモニター・コントロール・サーバーです。データベースがリモート・データベースとしてカタログされている場合は、**db2 connect to** コマンドでユーザー ID とパスワードを指定する必要があります。たとえば、以下のよう指定します。

```
db2 connect to database user userid using password
```

- b. 以下のコマンドを入力し、レプリケーション・アラート・モニター・プログラム・パッケージを作成してデータベースにバインドする。

```
db2 bind @asnmoncs.lst isolation cs blocking all grant public
```

```
db2 bind @asnmonur.lst isolation ur blocking all grant public
```

ただし、*cs* はカーソル固定形式でリストを指定するのに対し、*ur* は非コミット読み取り形式でリストを指定します。

これらのコマンドは、パッケージを作成します。その名前は、*asnmoncs.lst* および *asnmonur.lst* ファイル内で見つけることができます。

3. 現在モニターの対象としており、レプリケーション・アラート・モニター・プログラムの接続先となっているそれぞれのサーバーごとに、以下のステップを実行する。
 - a. 以下のように入力して、データベースに接続します。

```
db2 connect to database
```

ここで、*database* はモニターされているサーバーです。データベースがリモート・データベースとしてカタログされている場合は、**db2 connect to** コマンドでユーザー ID とパスワードを指定する必要があります。たとえば、以下のよう指定します。

```
db2 connect to database user userid using password
```

- b. 以下のコマンドを入力し、レプリケーション・アラート・モニター・プログラム・パッケージを作成してデータベースにバインドする。

```
db2 bind @asnmonit.lst isolation ur blocking all grant public
```

ただし *ur* は、非コミット読み取り形式のリストを指定します。

これらのコマンドは、パッケージを作成します。その名前は、*asnmonit.lst* ファイル内で見つけることができます。

キャプチャー・プログラムおよびアプライ・プログラムのセットアップ

(OS/400): アプライ・プログラムを他の非 OS/400 オペレーティング・システムのリモート・システムで使用する場合は、環境をセットアップする必要があります。以下のセクションでは、レプリケーション環境をセットアップするステップについて説明しています。

- 681 ページの『リモート・システムで使用するための SQL パッケージの作成 (OS/400)』

- 682 ページの『SQL パッケージへの特権の付与』

リモート・システムで使用するための SQL パッケージの作成 (OS/400): 以下のケースでは、**CRTSQPKG** コマンドを使用してパッケージを作成する必要があります。

- リモート・ジャーナリングを使用する場合。キャプチャー・プログラムを実行中のシステムで **CRTSQPKG** コマンドを実行し、ソース表が配置されているシステムを指示してください。
- **ADDDPRSUB** または **ADDDPRSUBM** コマンドを使用してサブスクリプション・セットまたはサブスクリプション・セットのメンバーを追加する前。以下のようにして、ターゲット・サーバーで **CRTSQPKG** コマンドを実行してください。
 - ソース表が別のマシン上にある場合は、そのソース表が配置されているシステムを指示する。
 - アプライ・コントロール・サーバーが別のマシン上にある場合は、そのアプライ・コントロール・サーバーを指示する。

SQL パッケージを使用すると、レプリケーション・プログラムが分散レプリケーション環境で稼働できるようになります。その場合、レプリケーションが OS/400 システム間で行われる環境であっても、OS/400 システムと他のいずれかのオペレーティング・システム (UNIX や Windows など) との間で行われる環境であっても構いません。

CRTSQPKG コマンドの使用方法は、「*DB2 Universal Database for iSeries SQL Programming*」を参照してください。

パッケージは ASN 修飾子を使って作成されます。OS/400 では、パッケージは ASN ライブラリーに作成されます。他のプラットフォームでは、パッケージは ASN スキーマに作成されます。

アプライ・プログラム用の SQL パッケージの作成: SQL パッケージを作成して、接続する必要があるすべてのリモート・サーバーとアプライ・プログラムが相互作用できるようにする必要があります。たとえば、アプライ・プログラムがリモート・システムに接続できるようにするためには、アプライ・プログラムが実行されているシステムで以下のコマンドを実行してください。

```
CRTSQPKG PGM(QDP4/QZSNAPV2) RDB(remote_system)
```

remote_system は、アプライ・プログラムが接続する必要があるリモート・システムでのリレーショナル・データベース入り口名です。

レプリケーション・アナライザー用の SQL パッケージの作成: SQL パッケージを作成して、キャプチャー・コントロール・サーバーやターゲット・サーバーなどといった、分析の対象としているサーバーとレプリケーション・アナライザーが相互作用できるようにする必要があります。レプリケーション・アナライザーが実行されているシステムで以下のコマンドを実行してください。

```
CRTSQLPKG PGM(QDP4/QZSNANZR) RDB(remote_system)
```

ここで、*remote_system* は分析の対象としているシステムの名前です。

レプリケーション管理コマンド用の SQL パッケージの作成: OS/400 システム間のレプリケーションでは、リモート・ジャーナルを使用する場合、このコマンドを使用してキャプチャー・プログラム用およびレプリケーション管理コマンド用のパッケージを作成する必要があります。キャプチャーが実行されているシステムで以下のコマンドを実行してください。

```
CRTSQLPKG PGM(QDP4/QZSNSQLF) RDB(source_system) OBJTYPE(*SRVPGM)
```

ここで、*source_system* はソース表が実際に存在するシステムの名前です。

SQL パッケージへの特権の付与: パッケージの作成後には、ソース・データベースに登録されているファイルにサブスクライブするすべてのユーザーに対して *EXECUTE 特権を付与する必要があります。ソース・データベースが常駐している OS/400 システムにログオンし、以下の方法のいずれかを使用してください。

- 以下のオブジェクト権限付与 (**GRTOBJAUT**) コマンドを使用する。

```
GRTOBJAUT OBJ(ASN/package_name) OBJTYPE(*SQLPKG)
USER(subscriber_name) AUT(*OBJOPR *EXECUTE)
```

- SQL を使用してソース・データベースに接続し、以下の GRANT SQL ステートメントを実行する。

```
CONNECT TO data_server_RDB_name
GRANT EXECUTE ON PACKAGE ASN/package_name TO subscriber_name
```

- **GRTDPRAUT** コマンドを使用する (このコマンドがローカル・システムにインストールされている場合)。

レプリケーション・プログラムのセットアップ (z/OS): IBM DB2

DataPropagator for z/OS のインストール時にレプリケーション・プログラムをセットアップしてカスタマイズする必要があります。「*Program Directory for IBM DB2 DataPropagator for z/OS*」に記載されている説明を参照してください。

ジャーナルのセットアップ (OS/400)

DB2 DataPropagator for iSeries は、データに対する変更についてジャーナルから受け取る情報を使用することによって、レプリケーションのための CD および UOW 表にデータを入れます。

DB2 DataPropagator for iSeries は、大部分の操作のコミットメント・コントロール下で実行するため、コントロール表のジャーナリングを必要とします。(QSQJRN ジャーナルは、**CRTPRTBL** コマンドが集合を作成するときに作成されます。)

管理者は、ソース表、CD 表、およびターゲット表が含まれているライブラリーにジャーナルが含まれていることを確認する必要があります。すべてのソース表のジャーナル処理が正しく行われるのを見届ける責任もあります。

OS/400 でレプリケーション用の表を登録する前に、その表が変更前イメージと変更後イメージの両方についてジャーナリングされている必要があります。

以下のセクションでは、レプリケーションに必要なジャーナル・セットアップについて説明しています。

- 『ソース表用ジャーナルの作成 (OS/400)』
- 684 ページの『ジャーナルおよびジャーナル・レシーバーの管理 (OS/400)』

ソース表用ジャーナルの作成 (OS/400): ソース表のジャーナルをセットアップするには、ソース表のジャーナルおよびジャーナル・レシーバーを定義するために作成する権限を所有している必要があります。(使用するソース表がすでにジャーナリングされている場合は、このセクションをスキップしてください。)

重要: ソース表のために使用するジャーナルは、DB2 DataPropagator for iSeries によって ASN (またはこれ以外のキャプチャー・スキーマ) ライブラリーに作成されたジャーナル以外のものにしてください。

手順:

ソース表のジャーナルを作成するには:

1. ジャーナル・レシーバーの作成 (**CRTJRNRCV**) コマンドを使用して、選択したライブラリーにジャーナル・レシーバーを作成します。定期的に保管されるライブラリーにジャーナル・レシーバーを置く。将来ジャーナル・レシーバーの命名規則を作成するのに使用できるようなジャーナル・レシーバー名 (**RCV0001** など) を選択する。***GEN** オプションを使用して、ジャーナル・レシーバーを変更するときに命名規則を継続することができます。このタイプの命名規則は、ジャーナル・レシーバーの変更をシステム管理に許可する場合にも役立ちます。以下の例では、ジャーナル・レシーバー用に **JRNLIB** という名前のライブラリーを使用します。

```
CRTJRNRCV JRNRCV(JRNLIB/RCV0001)
          THRESHOLD(100000)
          TEXT('DataPropagator Journal Receiver')
```

2. ジャーナルの作成 (**CRTJRN**) コマンドを使用して、ジャーナルを作成します。

```
CRTJRN JRN(JRNLIB/DJRN1)
       JRNRCV(JRNLIB/RCV0001)
       MNGRCV(*SYSTEM) DLTRCV(*YES)
       TEXT('DataPropagator Journal')
```

- ステップ 1 で作成したジャーナル・レシーバーの名前を指定する。
- アタッチされたレシーバーが大きくなりすぎる場合、システムがジャーナル・レシーバーを変更して、新しいレシーバーをアタッチするために、レシーバーの管理 (**MNGRCV**) パラメーターを使用します。このオプションを選択すると、**CRTJRN** コマンドを使って、手動でレシーバーを切り離したり、新しいレシーバーを作成およびアタッチしたりする必要がなくなります。

- デフォルトの属性 MINENTDTA(*NONE) を使用する。このキーワードの場合は、他の値は無効です。
- オーバーライドする理由がある場合 (たとえば、リカバーさせる目的でこれらのジャーナル・レシーバーを保管する必要がある場合) にのみ、DLTRCV(*NO) を指定します。DLTRCV(*YES) を指定すると、これらのレシーバーは、保管する機会もなく削除されてしまう可能性があります。

CRTJRN コマンドの RCVSIZOPT パラメーターに 2 つの値 (*RMVINTENT および *MINFIXLEN) を使用して、ストレージの可用性やシステム・パフォーマンスを最適化することができます。詳細については、「*OS/400 Programming: Performance Tools Guide*」を参照してください。

3. 以下の例で示す方法で、物理ファイルのジャーナル開始 (**STRJRNP**) コマンドを使って、ソース表のジャーナリングを開始します。

```
STRJRNP FILE(library/file)
        JRN(JRNLIB/DJRN1)
        OMTJRNE(*OPNCLO)
        IMAGES(*BOTH)
```

ステップ 2 (683 ページ) で作成したジャーナルの名前を指定します。キャプチャー・プログラムには、IMAGES パラメーターの *BOTH という値が必要です。

4. ソース表のジャーナリングのセットアップを変更する。
 - a. IMAGES(*BOTH) を使用して、ソース表に対するジャーナリングが確実に変更前イメージと変更後イメージの両方について行われるようにする。
 - b. そのジャーナルに属性 MNGRCV(*YES) および DLTRCV(*YES) が指定されていることを確認する。
 - c. そのジャーナルに MINENTDTA(*NONE) が指定されていることを確認する。

ジャーナルおよびジャーナル・レシーバーの管理 (OS/400): キャプチャー・プログラムは、ジャーナル・エントリーの受信 (**RCVJRNE**) コマンドを使って、ジャーナルを受信します。以下のセクションでは、レプリケーション環境でジャーナルおよびジャーナル・レシーバーを管理する方法が説明されています。

- 『ジャーナル・レシーバーのシステム管理の指定 (OS/400)』
- 685 ページの『実行管理機能オブジェクトの定義の変更 (OS/400)』
- 685 ページの『ジャーナル・レシーバーのユーザー管理の指定』
- 686 ページの『ジャーナル・レシーバー削除出口ルーチンの使用』

ジャーナル・レシーバーのシステム管理の指定 (OS/400): 推奨: OS/400 システムにジャーナル・レシーバーの変更を管理させてください。これは、システム変更ジャーナル管理と呼ばれています。ジャーナルを作成するとき、またはジャーナルをその値に変更するとき、MNGRCV(*SYSTEM) を指定してください。システム変更ジャーナル管理を使用する場合、システムに変更させるジャーナル・レシーバーのしきい値を指定するジャーナル・レシーバーを作成する必要があります。このしきい値は、最低でも

5,000 KB にし、システムでのトランザクションの数に基づいて決定する必要があります。システムは、レシーバーがしきい値のサイズに達して、新しいジャーナル・レシーバーが作成およびアタッチされると (可能な場合)、そのレシーバーを自動的に切り離します。

実行管理機能オブジェクトの定義の変更 (OS/400): DB2 DataPropagator for iSeries のインストール時に、このライブラリー用の SQL ジャーナルと SQL ジャーナル・レシーバー、および実行管理機能オブジェクトがインストール・プログラムによって作成されます。表 170 で、作成される実行管理機能オブジェクトをリストします。

表 170. 実行管理機能オブジェクト

説明	オブジェクト・タイプ	名前
サブシステム記述	*SBSD	QDP4/QZSNDPR
ジョブ・キュー	*JOBQ	QDP4/QZSNDPR
ジョブ記述	*JOBDB	QDP4/QZSNDPR

上記の 3 つのタイプの実行管理機能オブジェクトについて、そのデフォルトの定義を変更しても構いませんが、独自の定義も提供できます。独自のサブシステム記述を作成する場合は、サブシステムの名前として QZSNDPR を指定して、そのサブシステムを QDP4 以外のライブラリーに作成する必要があります。これらの定義の変更についての詳細は、「*iSeries Work Management*」(SC41-5306) を参照してください。

ジャーナル・レシーバーのユーザー管理の指定: ジャーナルの作成時に MNGRCV(*USER) を指定すると (独自にジャーナル・レシーバーの変更管理を行いたい場合)、ジャーナル・レシーバーがストレージのしきい値 (レシーバーに指定した場合) に達すると、ジャーナルのメッセージ・キューにメッセージが送信されます。

CHGJRN コマンドを使用して、古いジャーナル・レシーバーを切り離し、新しいレシーバーをアタッチします。このコマンドを使用すると、ジャーナルされないエントリーエラー状態を避け、ジャーナルが使用するストレージの量を制限することができます。パフォーマンスに影響がでないようにするため、システムが最大使用回数に達していないときにこのことを行ってください。

CHGJRN MNGRCV(*SYSTEM) を指定すると、ジャーナル・レシーバーの管理をシステムが行うよう切り替えることができます。

以下の 2 つの理由で、現行のジャーナル・レシーバーを定期的に切り離し、新しいレシーバーをアタッチする必要があります。

- 各ジャーナル・レシーバーに特定の管理可能な期間が含まれていると、ジャーナル・エントリーのアナライズがより簡単になるから。

- 大きなジャーナル・レシーバーは、システム・パフォーマンスに影響を及ぼし、補助記憶装置の貴重なスペースをふさいでしまう場合があるから。

ジャーナルのデフォルトのメッセージ・キューは QSYSOPR です。QSYSOPR メッセージ・キューに大きなボリュームのメッセージがあると、別のメッセージ・キュー (DPRUSRMSG など) とジャーナルを関連付けることができます。メッセージ処理プログラムを使って、DPRUSRMSG メッセージ・キューをモニターすることができます。ジャーナル・メッセージ・キューに送信できるメッセージに関する説明は、OS/400 *Backup and Recovery* を参照してください。

ジャーナル・レシーバー削除出口ルーチンの使用: DB2 DataPropagator for iSeries をインストールすると、ジャーナル・レシーバー削除 出口ルーチン (**DLTJRNRCV**) が自動的に登録されます。この出口ルーチンは、ジャーナル・レシーバーが削除されると、そのレシーバーがソース表のジャーナリングに使用されているかどうかにかかわらず、常に呼び出されます。この出口ルーチンは、ジャーナル・レシーバーを削除できるかどうかを決定します。

ジャーナル・レシーバー削除出口ルーチンを使用して、ジャーナル管理をシステムに任せるには、**CHGJRN** または **CRTJRN** コマンドに対して **DLTRCV(*YES)** および **MNGRCV(*SYSTEM)** を指定します。

重要: ジャーナル・レシーバー削除出口ルーチン用の登録を除去する場合は、ソース表用に使用されるすべてのジャーナルを変更して、それらが **DLTRCV(*NO)** 属性を持つようにする必要があります。

レシーバーが関連付けられているジャーナルがどのソース表にも関連付けられていない場合、この出口ルーチンはそのレシーバーの削除を承認します。

ジャーナル・レシーバーが 1 つかそれ以上のソース表で使われている場合、この出口ルーチンは、キャプチャー・プログラムによって処理されなかったエントリーが、削除されているレシーバーに含まれていないことを確認します。キャプチャー・プログラムがそのレシーバーのエントリーをさらに処理する必要がある場合には、出口ルーチンはレシーバーの削除を承認しません。

ジャーナル・レシーバーを削除する必要があるのに、ジャーナル・レシーバー出口ルーチンが削除を承認しない場合には、**DLTJRNRCV DLTOPT(*IGNEXITPGM)** を指定して、その出口ルーチンをオーバーライドします。

ジャーナル・レシーバー削除出口ルーチンの削除: ジャーナル・レシーバーの削除を手動で処理したい場合、以下のコマンドを使用して、ジャーナル・レシーバー削除出口ルーチンを削除することができます。

```
RMVEXITPGM EXITPNT (QIBM_QJO_DLT_JRNRCV)
                FORMAT(DRCV0100)
                PGMNBR(value)
```

手順:

RMVEXITPGM コマンドの **PGMNBR** 値を決定するには:

1. **WRKREGINF** コマンドを実行します。
2. 「登録情報の処理」ウィンドウで、出口点のエントリ **QIBM_QJO_DLT_JRNRCV** を検索します。「**Opt**」フィールドに **8** と入力します。
3. 「出口プログラムの処理」ウィンドウで、ライブラリー **QDP4** にある出口プログラムのエントリ **QZSNDREP** を検索します。必要な数値は、出口プログラム番号の見出しの下にあります。

ジャーナル・レシーバー削除出口ルーチンの登録: 出口点を除去した後でそれを元に戻す必要が生じた場合は **ADDEXITPGM** コマンドを使用してください。その出口ルーチンは以下のコマンドを使用して登録する必要があります。

```
ADDEXITPGM EXITPNT(QIBM_QJO_DLT_JRNRCV)
              FORMAT(DRCV0100)
              PGM(QDP4/QZSNDREP)
              PGMNBR(*LOW)
              CRTEXITPNT(*NO)
              PGMDTA(65535 10 QSYS)
```

ソースのサブスクライブ

レプリケーション・ソースとして使用する表とビューを登録後、ターゲット表またはビューがソースからのソース・データと変更内容を受け取れるように、ターゲット表またはビューのサブスクリプションを定義できます。本章で説明する管理タスクは、キャプチャー・プログラムとアプライ・プログラムが、ソース・データをコピーするため、あるいは変更データを取り込んでそれをターゲット表に適切なインターバルで複製するために使用するコントロール情報のセットアップを支援します。

本章は、以下のセクションで構成されています。

- 『ソースおよびターゲットのグループ化の計画』
- 690 ページの『サブスクリプション・セットの作成』
- 693 ページの『サブスクリプション・セットのオプション処理』
- 701 ページの『サブスクリプション・セット内におけるソース表およびビューのターゲット表およびビューへのマッピング』
- 704 ページの『ターゲット・タイプの選択』
- 716 ページの『すべてのターゲット表タイプに共通のプロパティ』

ソースおよびターゲットのグループ化の計画

どのターゲットがどのソースをサブスクライブするかを定義する前に、ソースとターゲットをどのようにグループ化するかを計画する必要があります。DB2 レプリケーションは、グループ単位でソースからターゲットへのマッピングを処理します。これらのグ

ループは、同一のキャプチャー・プログラムによって処理される 1 つ以上のソースと、ソース・データのすべてまたは一部をサブスクライブし、同一のアプライ・プログラムによって処理される 1 つ以上のターゲットで構成されています。これらのグループを、サブスクリプション・セットと呼び、ソースからターゲットへのマッピングをサブスクリプション・セット・メンバーと呼びます。

サブスクリプション・セットを計画するときは、以下の規則と制約に注意してください。

- サブスクリプション・セットは、ソース・サーバーをターゲット・サーバーにマップする。サブスクリプション・セット・メンバーは、ソース表またはビューを、ターゲット表またはビューにマップします。サブスクリプション・セットとそのメンバーは、アプライ・コントロール・サーバーに保管されます。
- アプライ・プログラムは、サブスクリプション・セットのすべてのメンバーを単一グループとして処理する。このため、サブスクリプション・セットのいずれかのメンバーで、何らかの理由でフル・リフレッシュ・コピーが必要な場合、セット全体のすべてのメンバーがリフレッシュされます。
- 1 つのセットのメンバーにおいて、すべてのソース表およびビューのキャプチャー・スキーマは同じでなければならない。
- OS/400 システムでは、1 つのサブスクリプション・セットのメンバーにおいて、すべてのソース表は同じジャーナルに記録されなければならない。
- IMS DataPropagator で作成され、サブスクリプション・セットのメンバーである外部 CCD 表すべてのキャプチャー・スキーマは同じでなければならない。

ユニークのアプライ修飾子を持つ単一のアプライ・プログラムは、サブスクリプション・セットを 1 つでもあるいは多数でも処理できます。単一のサブスクリプション・セットには、サブスクリプション・セット・メンバーを 1 つでもあるいは多数でも入れることができます。以下のセクションでは、アプライ・プログラムごとに少数または多数のセットを保有する場合と、サブスクリプション・セットごとに少数または多数のサブスクリプション・セット・メンバーを保有する場合のトレードオフについて説明します。

サブスクリプション・セット・メンバー数の計画: サブスクリプション・セットにメンバーを追加するとき、ソースとターゲットのすべてのペア (サブスクリプション・セット・メンバー) を 1 つのサブスクリプション・セットにまとめるか、それぞれのペアごとに別々のサブスクリプション・セットを作成するか、あるいはそれぞれにかなりの数のペアが含まれた、少数のサブスクリプション・セットを作成するかを決定する必要があります。

アプライ・プログラムは、サブスクリプション・セットのメンバーを 1 つの (論理) トランザクションで複製するため、以下のどちらの状態においても、複数のメンバーを 1 つのサブスクリプション・セットにまとめる必要があります。

- ソース表に相互に論理的な関連がある場合。

- ターゲット表に参照保全制約がある場合。

複数のメンバーを 1 つのサブスクリプション・セットにまとめると、すべてのメンバーのレプリケーションを必ず同時に開始することができます。さらに、サブスクリプション・セットを処理するのに必要なデータベース接続の数と、レプリケーション環境を保守するための管理オーバーヘッドが削減されます。サブスクリプション・セットに SQL ステートメントまたはストアド・プロシージャが含まれている場合、これらのステートメントまたはプロシージャを使用して、そのサブスクリプション・セットのすべてのメンバーを処理できます。

サブスクリプション・セットの表の間に、論理的または参照保全のリレーションシップがない場合、その表を 1 つまたは幾つかのサブスクリプション・セットにまとめることができます。サブスクリプション・セットの数を制限する主な理由は、レプリケーション環境の管理を簡単にすることです。ただし、サブスクリプション・セットの数を増やすと、レプリケーションの失敗による影響は最小になります。

アプライ・プログラムが失敗する原因となるエラーをより簡単に突き止められるようにするには、少数のメンバーしかサブスクリプション・セットに追加しないようにします。メンバーが少数なら、多数のメンバーがセットに含まれている場合より迅速に問題のソースを見付けることができます。サブスクリプション・セットの 1 つのメンバーで障害が起こった場合、そのセットの他のメンバーに適用されたデータはすべてロールバックされます。そのため、すべてのメンバーがサイクルを完了しない限り、どのメンバーも正常にサイクルを完了できません。アプライ・プログラムは、失敗したサブスクリプション・セットをその最後の正常なコミット・ポイントまでロールバックします。このコミット・ポイントは、アプライ・プログラムを始動したときに **commit_count** キーワードを指定していれば、現行のアプライ・サイクル内にある可能性があります。

アプライ修飾子あたりのサブスクリプション・セット数の計画: サブスクリプション・セットを定義するとき、そのサブスクリプション・セット用のアプライ修飾子を指定します。アプライ修飾子は、アプライ・プログラムのインスタンスを 1 つ以上のサブスクリプション・セットに関連付けます。それぞれのサブスクリプション・セットを処理するのは、ただ 1 つのアプライ・プログラムですが、アプライ・プログラムはそれぞれ 1 つ以上のサブスクリプション・セットを、1 つ 1 つのアプライ・サイクル内で処理できます。

アプライ・プログラムのインスタンス (それぞれに独自のアプライ修飾子があります) は必要な数だけ実行でき、アプライ・プログラムはそれぞれサブスクリプション・セットに必要な数だけ処理できます。基本オプションが 2 つあります。

- アプライ修飾子をそれぞれ 1 つのサブスクリプション・セットに関連付ける (個々のアプライ・プログラムが、厳密に 1 つのサブスクリプション・セットを処理する)。速度が重要な場合は、幾つかのアプライ修飾子間にセットを分散でき、こうすることで、アプライ・プログラムの幾つかのインスタンスを同時に実行できます。1 つのアプライ・プログラム・インスタンス・プロセスで 1 つのサブスクリプション・セ

ットを処理するように決めていると、このアプライ・プログラムの OPT4ONE 始動オプションを使用できます。このオプションは、サブスクリプション・セットのコントロール表情報をメモリーにロードするものです。このオプションを使用すると、アプライ・プログラムは、サブスクリプション・セット情報のコントロール表をアプライ・サイクルごとには読み取りません。したがって、アプライ・プログラムのパフォーマンスが向上します。ただし、実行するアプライ・プログラム・インスタンスが多くなれば、それらが使用するシステム・リソースも多くなり、全体としてのパフォーマンスは低下する可能性があります。

- アプライ修飾子をそれぞれ複数のサブスクリプション・セットに関連付ける (個々のアプライ・プログラムが、多数のサブスクリプション・セットを処理する)。

複数のアプライ修飾子を使用すれば、シングル・ユーザー ID からアプライ・プログラムの複数のインスタンスを実行できます。

アプライ・プログラムでは、指定されたアプライ修飾子に対応するセットすべてを、可能な限り現状のまま保持しようとしています。アプライ・サイクルが開始すると、アプライ・プログラムは含まれる現行データが最も少ないサブスクリプション・セットを判別し、まずそのセットの処理を開始します。

速度が主要なゴールでない場合は、1つのアプライ修飾子が指定された膨大な数のサブスクリプション・セットを複製することがあります。たとえば、営業時間後まで待ってから複製する場合に、これは大変適切なオプションということもあります。

1つのアプライ・プログラムで複数のサブスクリプション・セットを処理する欠点は、そのアプライ・プログラムがサブスクリプション・セットを順次処理することです。したがって、全体のレプリケーション待ち時間は増加する可能性があります。

一部のサブスクリプション・セットに対して特定の要件がある場合、この2つのオプションを組み合わせることができます。たとえば、1つのアプライ・プログラムにほとんどのサブスクリプション・セットを処理させることができるので、関連するサブスクリプション・セットをまとめて処理すれば、アプライ・プログラムを有効に利用できるでしょう。そして、別のアプライ・プログラムに単一のサブスクリプション・セットを処理させることができるので、そのサブスクリプション・セットのレプリケーション待ち時間を確実に最小にできます。さらに、アプライ・プログラムの2つのインスタンスを使用すれば、サブスクリプション・セットの並行処理全体が強化されます。

サブスクリプション・セットの作成

登録済みのソースからデータを複製する前に、アプライ・プログラムが1つの集合として処理するサブスクリプション・セット・メンバー (ソースからターゲットへのマッピング) の集合である、サブスクリプション・セットを作成する必要があります。このセクションでは、それぞれのサブスクリプション・セットごとに定義するプロパティーについて説明します。セットに追加するメンバーごとに適用されるこれらのプロパティーは、データのレプリケーション先とレプリケーション元のサーバーを定義し、使用するキャプチャー・プログラム (登録済みソースのキャプチャー・スキーマを基にしています) とアプライ・プログラム、およびアプライ・プログラムがセットを処理する時期と方法も定義します。

サブスクリプション・セット・メンバーをサブスクリプション・セットに追加する必要はありません。代わりに、ソースからターゲットへのマッピングを含まない空のセットを作成できます。以下のような理由から、空のセットの作成が必要になることがあります。

- 後でメンバーをセットに追加する計画があり、メンバーを追加するまではサブスクリプション・セットをアクティブにする計画はない。
- 空のサブスクリプション・セットが処理に適切であるとき、いつでも SQL ステートメントまたはストアード・プロシージャを呼び出すために、アプライ・プログラムでそのセットを処理する。

前提条件:

1. サブスクリプション・セット用に、アプライ・コントロール表をアプライ・コントロール・サーバーで作成しなければならない。
2. サブスクリプション・セットにサブスクリプション・セット・メンバーを追加する前に、ソースとして使用する表またはビューを登録しなければならない。レプリケーションのためにソースを登録する必要がある場合は、637 ページの『表およびビューをレプリケーション・ソースとして登録する』の説明を読んでください。セットの分類方法について検討する必要もあります。セットについて計画する必要がある場合は、詳細は、687 ページの『ソースおよびターゲットのグループ化の計画』を参照してください。

手順:

サブスクリプション・セットを作成するには、以下の 2 つの方法のどちらかを使用できます。

レプリケーション・センター

「サブスクリプション・セットの作成 (Create subscription sets)」ノートブックを使用します。詳細については、レプリケーション・センターのヘルプを参照してください。

レプリケーション用のシステム・コマンド (OS/400)

ADDDPRSUB システム・コマンドを使用します。このコマンドの構文およびパラメーターについての記述は、580 ページの『ADDDPRSUB: DPR サブスクリプション・セットの追加 (OS/400)』を参照してください。

サブスクリプション・セットを作成するには、以下の基本的な特性を指定します。

アプライ・コントロール・サーバー別名

サブスクリプション・セットを処理するアプライ・プログラム用のコントロール表を含むサーバーのローカルの別名。どのデータベースにあるアプライ・コントロール・サーバーにも、同一の別名を定義します。このサーバーからレプリケーション・センターまたはアプライ・プログラムを実行して、レプリケー

ション・センターがアプライ・コントロール表を適切に移植するようにし、さらにどのアプライ・プログラムも標準の別名を使用して適切なサーバーに接続するようにします。

サブスクリプション・セット名

サブスクリプション・セットの名前。サブスクリプション・セットを処理するアプライ・コントロール・サーバーにおいて、セット名は指定されたアプライ修飾子に対してユニークでなければなりません。名前の長さは 18 文字まで可能です。

アプライ修飾子

新規または既存のアプライ修飾子の名前。サブスクリプション・セットを処理するアプライ・プログラムを示します。同じアプライ修飾子を使用して、複数のサブスクリプション・セットを処理できます。同じアプライ修飾子を持つサブスクリプション・セットは、同じアプライ・コントロール・サーバーで定義する必要があります。新規のアプライ修飾子を作成する場合は、アプライ修飾子の命名方法の規則について、633 ページの『レプリケーション・オブジェクトの命名規則』を参照してください。

キャプチャー・コントロール・サーバー別名

サブスクリプション・セットに登録済みのソースを処理するキャプチャー・プログラム用のコントロール表を含むサーバーの別名。どのデータベースにあるキャプチャー・コントロール・サーバーにも、同一の別名を定義します。このサーバーからレプリケーション・センターまたはアプライ・プログラムを実行して、レプリケーション・センターがキャプチャー・コントロール表とアプライ・コントロール表を適切に移植するようにし、さらにどのアプライ・プログラムも標準の別名を使用して適切なサーバーに接続するようにします。

キャプチャー・スキーマ

キャプチャー・スキーマの名前。サブスクリプション・セットに登録済みのソースを定義するキャプチャー・コントロール表のセットを示します。サブスクリプション・セットのすべてのソース表は、同じサーバーに常駐していなければならない、1 つのキャプチャー・プログラムだけがソース表への変更をキャプチャーすることができます。

ターゲット・サーバー別名

アプライ・プログラムがソースからの変更を複製する先の表またはビューが含まれるターゲット・サーバーの名前。どのデータベースにあるターゲット・サーバーにも、同一の別名を定義します。このサーバーからレプリケーション・センターまたはアプライ・プログラムを実行して、レプリケーション・センターがアプライ・コントロール表を適切に移植するようにし、さらにどのアプライ・プログラムも標準の別名を使用して適切なサーバーに接続するようにします。

サブスクリプション・セットを作成するときは、アプライ・プログラムがセットを処理する方法について、デフォルトの設定値を使用できます。あるいは、サブスクリプションのプロパティを、レプリケーションの要求を満たすように変更できます。サブスクリプション・セット用の処理オプションとそのデフォルト、およびデフォルトの使用または変更が必要な場合についての説明の完全なリストは、『サブスクリプション・セットのオプション処理』を参照してください。

サブスクリプション・セットのオプション処理

このセクションでは、アプライ・プログラムがサブスクリプション・セットを処理する方法を指定するために定義可能なプロパティについて説明します。さらに、レプリケーションの要求に基づいて選択すべき設定値を決定する際にも、このセクションは役立ちます。

- 『セットがアクティブかどうかの指定』
- 694 ページの『アプライ・プログラムが取り出すデータに相当する分数の指定』
- 696 ページの『アプライ・プログラムがセットのメンバーへの変更を複製する方法の指定』
- 698 ページの『サブスクリプション・セット用の SQL ステートメントまたはストアド・プロシージャの定義』
- 698 ページの『サブスクリプション・セットのレプリケーションのスケジューリング』

セットがアクティブかどうかの指定: デフォルト: 非アクティブ

アプライ・プログラムがサブスクリプション・セットの処理を開始するかどうかを指定できます。サブスクリプション・セットを活動化すると、アプライ・プログラムは、そのセットのフル・リフレッシュを開始します。活動化レベルは以下の 3 つから選択します。

アクティブ

アプライ・プログラムは、その次のサイクル内でセットを処理します。アプライ・プログラムに次の実行時でセットを処理させる場合は、セットを活動化します。後になっても、メンバーをセットに追加できます。セットを活動化すると、ユーザーがセットを非活動化するまで、セットはアクティブのまま、アプライ・プログラムはその処理を続けます。

非アクティブ

アプライ・プログラムはセットを処理しません。アプライ・プログラムでセットを処理する準備ができていない場合は、セットを非アクティブにしておきます。

1 回のみアクティブ

アプライ・プログラムは次のサイクル内でセットを処理し、その後セットを非活動化します。セットを 1 回だけ実行したい場合は、このオプションを指定します。このオプションを選択する前に、必ずすべてのサブスクリプション・セ

ット・メンバーを追加してください。その理由は、サブスクリプション・セットを再活動化しない限り、アプライ・プログラムは後で追加したメンバーを処理しないためです。

アプライ・プログラムが取り出すデータに相当する分数の指定: デフォルト: 20分

アプライ・プログラムが各アプライ・サイクル内でレプリケーション・ソースから取り出すデータを、それに相当するおおよその分数で指定できます。この指定が役に立つ状態が幾つかあります。

- 1 回のサブスクリプション・セット・サイクル内で処理されるデータ量が大量の場合。
1 回のアプライ・サイクルで大量の変更ブロックを複製するサブスクリプション・セットは、予備ファイルまたはログ (ターゲット・データベースの場合) でのオーバーフローの原因となります。たとえば、アプライ・プログラムをバッチ処理するシナリオでは、レプリケーションを必要とするエンキューされたトランザクションのバックログが大量に生成される可能性があります。
- ネットワークの停止が長引くと、大量のデータ・ブロックが CD 表に累積され、アプライ・プログラムの予備ファイルとターゲットのログがオーバーフローする可能性がある。

指定する分数を、データ・ブロックと呼びます。指定するデータ・ブロッキングの値は、サブスクリプション・セット (IBMSNAP_SUBS_SET) の表で MAX_SYNCH_MINUTES 列に保管されます。データの累積がデータ・ブロックのサイズより大きいと、アプライ・プログラムは 1 つのアプライ・サイクルを幾つかのミニサイクルに変換します。リソースが、指定されたブロッキング因数をハンドルするにはまだ不十分な場合、アプライ・プログラムはデータ・ブロックのサイズを削減し、使用可能なシステム・リソースに一致させます。より小さいデータ集合を取り出すことにより、アプライ・プログラムはネットワーク負荷と取り出されたデータ用に一時的に必要なスペースの両方を減少させます。

例: アプライ・プログラムがミニサイクル当たりせいぜい 10 分相当のデータしか取り出さないように指定すると、最後のミニサイクルの約 10 分間以内で、アプライ・プログラムはかなりの量のコミット済みデータをソースの CD 表から取り出すようになります。

ログ・ファイルと予備ファイルをオーバーフローさせない点に加えて、ミニサイクルには、他にも幾つかの利点があります。レプリケーション・サイクルでエラーが発生している場合、アプライ・プログラムは、失敗したミニサイクルで行った変更のみをロールバックする必要があります。ミニサイクルでレプリケーションが失敗した場合、アプライ・プログラムは、最後に正常に実行されたミニサイクルからサブスクリプション・セットを処理しようとはしますが、これで大量の変更データを処理できる場合は、かなりの

時間を節約できます。図 10 で、変更データが変更のサブセットに分割される仕組みを示します。

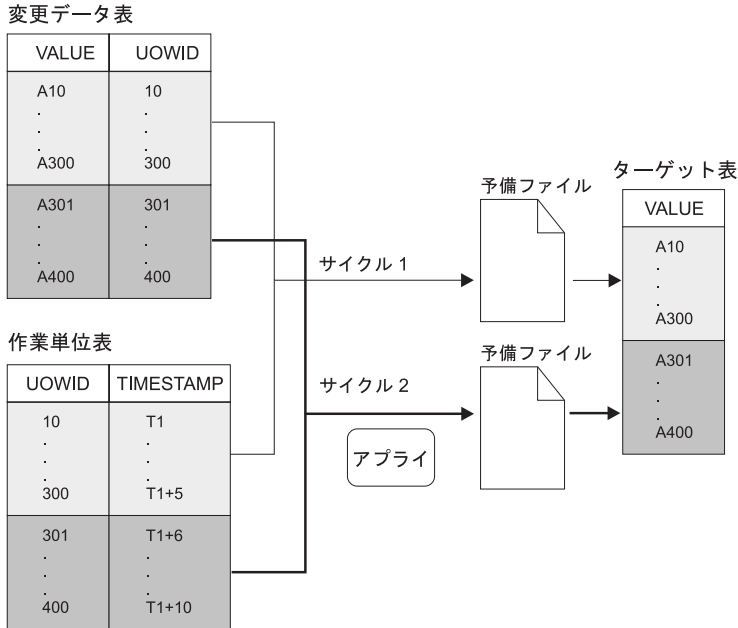


図 10. データ・ブロッキング： ネットワークのトラフィック量を少なくするには、データ・ブロック値を指定します。

設定する分数が小さければ、インターバル内に生じるサブスクリプション・セットのトランザクションをすべてコピーし、ミニサイクル内での予備ファイルやログのオーバーフローを防ぐことができます。

重要 (OS/400 の場合): 各アプライ・サイクル内で複製されるデータの総量が 400 万行を超えないようにしてください。

データを処理するとき、アプライ・プログラムは以下のどのアクションも行いません。

- 作業単位を分割する（すなわち、コミットされないで長時間実行中のバッチ・ジョブは、データ・ブロッキング因数で分割できない）。
- 以前コミットされたミニサブスクリプション・サイクルをロールバックする。
- フル・リフレッシュ中にデータ・ブロッキング因数を使用する。

アプライ・プログラムが参照保全性のあるターゲット表をロードする方法の決定

決定: セットのターゲット表の間で参照保全が要求される場合、ターゲット表の初期ロード時にアプライ・プログラムがソース・データを処理する方法を選択する必要があります。デフォルトでは、アプライ・プログラムはソースのすべての行を読み取り、それをメモリーに保管し、その後それらの行をターゲット表に挿入することによって、ター

ゲットのフル・リフレッシュを実行します。しかし、ターゲット表を初期ロードする方法には他のオプションがあります。サブスクリプション・セットを作成するとき、あるいはソースからターゲットへのマッピング (メンバー) をセットごとに定義するとき、これらを決定することはありません。すなわち、ターゲットのロード方法は、アプライ・プログラムの始動パラメーターを設定するときに決定します。サブスクリプション・セット・メンバーを定義するとき、それぞれのメンバーごとにレプリケーション要件を考慮する必要があり、それぞれのメンバーを処理するアプライ・プログラムで使用する始動オプションを決定できます。

これらの参照保全リレーションシップを作成する時期について、以下の 2 つの選択肢のどちらかを検討してください。

- ターゲット表が移植される前。

この場合は、ターゲット表の抽出およびロードの全段階を通じて、ソース表への変更が行われないことが必要です。また、この初期の移植中にロード処理の速度を確保し、参照制約チェックをバイパスするため、LOADX 始動オプションを使用してアプライ・プログラムを始動する必要があります。アプライ始動オプション LOADX を使用しない場合、ターゲット表への挿入が失敗する可能性があります。

- アプライ・プログラムがターゲット表を完全に移植し、それらの表の該当するセットに変更を適用する 1 つのサイクルを正常に完了した後。

参照保全制約をこれらの表に追加する時期を待つ場合、ターゲット表のロード中に、ソース表でさらに変更を行うことができるという利点があります。バイパスする必要があるという制約がないため、始動オプション LOADX を使用してもしなくても、アプライ・プログラムを始動できます。始動オプション LOADX を使用するよりも、一般的にフル・リフレッシュはかなり速度があります。ターゲット表の初期の移植中、ターゲットは参照保全リレーションシップの点で相互に同期がとれていないことがあります。しかし、ロード中に、セットの変更はすべて取り込まれます。アプライ・プログラムが変更の入った最初のセットを複製した後、すべてのターゲット表には同じトランザクションが入り、参照保全性を持つようになります。この時点でセットを非活動化して参照保全制約を追加し、その後セットを再び活動化できます。

ターゲット表の初期ロード方法のために提供されている始動オプションについての詳細は、493 ページの『ASNLOAD 出口ルーチンを使ったターゲット表のリフレッシュ』を参照してください。

アプライ・プログラムがセットのメンバーへの変更を複製する方法の指定: サブスクリプション・セットに変更キャプチャー・レプリケーションが含まれるとき、アプライ・プログラムがセットのソースからターゲットへのすべてのマッピングへの変更をレプリケーションする方法を決定できます。ターゲット表の初期ロード後、アプライ・プログラムは CD (または CCD) 表の読み取りを開始し、変更を予備ファイルに収集します。それぞれの CD (または CCD) 表ごとに、アプライ・プログラムは別々の予

備ファイルを作成します。そこでアプライ・プログラムは予備ファイルから変更を読み取り、変更をターゲット表にアプライします。以下の 3 つの方法のいずれかで、これを実行します。

- 表モード処理の使用。
- トランザクション・モード処理の使用。
- 表モードとトランザクション・モードが混合した処理の使用。サブスクリプション・セットのターゲット表タイプに依存します。

サブスクリプション・セットの処理タイプを指定すると、ターゲット表またはビューへの変更をアプライ・プログラムがコミットする頻度をコントロールできます。アプライ・プログラムは、それぞれのサブスクリプション・セット・メンバーごとに 1 回、あるいは相当数のトランザクションをアプライした後にコミットできます。コミットが 1 回の場合、サブスクリプション・セットの待ち時間を削減できますが、コミットが複数の場合は、アプライ・プログラムがオリジナルのコミット・シーケンスのデータをアプライできます。

表モード: アプライ・プログラムは CD (または CCD) 表の予備ファイルからすべての変更を読み取り、対応するターゲット表に変更をアプライしてから、次の CD (または CCD) 表の処理を開始します。セット内のすべての CD (または CCD) 表からの変更の読み取りとアプライが完了したら、アプライ・プログラムは DB2 コミットを実行して、サブスクリプション・セット内のすべてのターゲット表への変更をすべてコミットします。

トランザクション・モード: アプライ・プログラムは直ちにすべての予備ファイルを開き、予備ファイルからの変更を同時に処理します。変更がターゲット表に適用される順序は、トランザクションがソース表で実行された順序です。アプライ・プログラムは、始動時に指定されたインターバルで DB2 コミットを実行します。サブスクリプション・セットのターゲット表に参照保全制約がある場合は、このタイプの処理を使用してください。

どのサブスクリプション・セットの場合でもアプライ・プログラムがトランザクション・モード処理を使用するように指定できます。しかし、その指定で変更されるのは、セットにユーザー・コピー・ターゲット表とポイント・イン・タイム・ターゲット表が含まれる場合のアプライ・プログラムの動作のみで、以下のタイプのターゲット表がセットに含まれる場合の動作は変更されません。

- CCD ターゲット表。CCD 表がソースとして含まれるセットは、常に表モードで処理されます。
- ソース表が CCD 表である場合、それに対応するターゲット表。CCD 表が含まれるセットは、常に表モードで処理されます。
- レプリカ・ターゲット表。レプリカ表が含まれるセットは、常にトランザクション・モードで処理されます。

サブスクリプション・セット用の SQL ステートメントまたはストアド・プロシージャの定義: アプライ・プログラムがサブスクリプション・セットを処理するたびに実行される SQL ステートメントまたはストアド・プロシージャを定義できます。これらのステートメントは、CCD 表のブルーニング、またはターゲットにアプライされる前のソース・データの取り扱いに役立ちます。 SQL ステートメントまたはストアド・プロシージャの実行時期と実行場所を指定できます。

- アプライ・プログラムがデータをアプライする前にキャプチャー・コントロール・サーバーで。
- アプライ・プログラムがデータをアプライする前にターゲット・サーバーで。
- アプライ・プログラムがデータをアプライした後にターゲット・サーバーで。

アプライ・プログラムは、ステートメントまたはプロシージャを上記にリストした順序で処理します。

レプリケーション・センターを使用して SQL ステートメントをサブスクリプション・セットに追加するときは、「SQL ステートメントの追加」ウィンドウまたは「プロシージャ呼び出し」ウィンドウで **ステートメントの準備 (Prepare statement)** をクリックし、ステートメントの構文を検証します。

サブスクリプション・セットのレプリケーションのスケジューリング: アプライ・プログラムがサブスクリプション・セットを処理する頻度をコントロールできるので、その結果、ターゲット表のデータをどのくらいで最新にするかをコントロールできるようになります。時間に基づくスケジューリングまたはイベントに基づくスケジューリングを使用して、どのくらいの頻度でサブスクリプション・セットを処理対象として適格とするかをコントロールでき、あるいはこれらのスケジューリング・オプションと一緒に使用することもできます。アプライ・プログラムは、サブスクリプション・セットが処理対象として適格になると、その処理を開始します。たとえば、ある 1 日のインターバルを設定する一方で、あるイベントによってサブスクリプション・サイクルを起動するよう指定することもできます。これらのスケジューリング・オプションの両方を使用する場合、サブスクリプション・セットはスケジュールされた時刻とイベント発生時刻の両方で処理対象として適格になります。

アプライ・プログラムが処理するサブスクリプション・セットのインターバル内またはイベントとイベントの間で、複製されるデータ量が膨大な場合、特定のサブスクリプション・セットを処理対象として適格にすることは可能です。しかし、アプライ・プログラムがそれを処理できるのは、先行するインターバルまたはイベントで、すべてのサブスクリプション・セットへのデータのアプライが終了してからです。サブスクリプション・セットの処理が終了すると、アプライ・プログラムは直ちに次の処理に適格なサブスクリプション・セットの処理を開始します。このケースでは、レプリケーション待ち時間が予想とは異なるかもしれませんが、失われるデータはありません。

手順:

サブスクリプション・セットを指定するには、以下の 2 つの方法のどちらかを使用できます。

レプリケーション・センター

以下のノートブックのいずれかを使用します。

- サブスクリプション・セットの作成。「スケジュール」ページを使用して、スケジューリング・オプションを選択します。
- サブスクリプション・セット・プロパティ。すでにサブスクリプション・セットを作成済みで、サブスクリプション・セットのスケジューリングを変更したい場合は、このノートブックを使用します。

詳細については、レプリケーション・センターのオンライン・ヘルプを参照してください。

レプリケーション用のシステム・コマンド (OS/400)

ADDDPRSUB システム・コマンドを使用します。このコマンドの構文およびパラメーターについての記述は、580 ページの『ADDDPRSUB: DPR サブスクリプション・セットの追加 (OS/400)』を参照してください。

時間に基づくスケジューリング: セットを処理する時期をコントロールする最も単純な方法は、時間に基づくスケジューリング (相対タイミグまたはインターバル・タイミグとしても知られています) を使用することです。特定の開始日付、時間、およびインターバルを決定します。インターバルは、特定の値 (1 分 ~ 1 年) または連続した値にすることができますが、時間インターバルはおおよそその値になります。アプライ・プログラムは、作業負荷とリソースの可用性に基づいて、可能な限りすみやかにサブスクリプション・セットの処理を開始します。ある時間インターバルを選択しても、レプリケーションの頻度が正確にそのインターバルになるとは限りません。連続したタイミグを指定すると、アプライ・プログラムは可能な限り頻繁にデータを複製します。

イベントに基づくスケジューリング: イベントに基づくスケジューリング (イベント・タイミグとしても知られています) を使用してデータを複製するために、サブスクリプション・セットを定義するときにイベント名を指定できます。イベントの発生時に、アプライ・プログラムにそのイベントを認識させるには、イベント名に対応するタイム・スタンプを持つサブスクリプション・イベント (IBMSNAP_SUBS_EVENT) 表も移植する必要があります。アプライ・プログラムはイベントを検出するとレプリケーションを開始します。

サブスクリプション・イベント表には、表 171 で示されているように、4 つの列があります。

表 171. サブスクリプション・イベント表

EVENT_NAME	EVENT_TIME	END_OF_PERIOD	END_SYNCHPOINT
END_OF_DAY	2002-05-01-17.00.00.000000	2002-05-01-15.00.00.000000	

EVENT_NAME は、サブスクリプション・セットの定義時に指定するイベントです。EVENT_TIME は、アプライ・プログラムがセットの処理を開始する時刻を示すタイム・スタンプです。END_OF_PERIOD は、該当時刻より後の更新が後のイベントまたは時刻まで据え置かれることを指定するオプション値です。END_SYNCHPOINT は、該当するログ・シーケンス番号より後の更新が後のイベントまたは時刻まで据え置かれることを指定するオプション値です。END_OF_PERIOD と END_SYNCHPOINT の両方の値を指定すると、END_SYNCHPOINT の値が優先されます。EVENT_TIME の値はアプライ・コントロール・サーバーのクロックにより設定されますが、END_OF_PERIOD の値は ソース・サーバーのクロックにより設定されます。2 つのサーバーが別の時間帯にある場合、この区別は重要です。

699 ページの表 171 によると、END_OF_DAY というイベントの場合、EVENT_TIME のタイム・スタンプ値 (2002-05-01-17.00.00.000000) は、アプライ・プログラムがサブスクリプション・セットの処理を開始する時刻です。END_OF_PERIOD のタイム・スタンプ値 (2000-05-01-17.00.00.000000) は、更新が複製されなかった後、次の日のサイクルで複製される時刻です。つまり、イベントは 3 時前に作成されたすべての未確定の更新を複製して、それに続くすべての更新を延期します。

ユーザーまたはアプリケーションは、行を表に INSERT する SQL の INSERT ステートメントを使用してイベントをサブスクリプション・イベント (IBMSNAP_SUBS_EVENT) 表に通知して、イベントを活動化する必要があります。たとえば、現行のタイム・スタンプに 1 分加算した値を使用して、EVENT_NAME に指定したイベントを起動します。このイベントに結び付けられたサブスクリプション・セットはすべて、1 分内で実行するのに適したものとなります。フル・リフレッシュと変更キャプチャー・レプリケーション両方の目的で、イベントを手動で通知する必要があります。

イベントは、翌週、翌年、または毎週土曜日のように、前もって通知することができません。アプライ・プログラムが実行されている場合、アプライ・プログラムは指定されたおおよその時刻に処理を開始します。アプライ・プログラムは、指定された時刻に停止しており、後で再始動されると、サブスクリプション・イベント表をチェックして、通知されたイベントのサブスクリプション・セットの処理を開始します。

アプライ・プログラムはこの表を整理しません。ユーザーはこの表を移植して保守する必要があります。さらに、レプリケーション・センターを使用してサブスクリプション・イベント表を更新することはできません。この表にイベントを追加するには、SQL ステートメントを実行するか自動手順を定義する必要があります。

例:

```
INSERT INTO ASN.IBMSNAP_SUBS_EVENT
    (EVENT_NAME, EVENT_TIME)
VALUES ('EVENT01', CURRENT_TIMESTAMP + 1 MINUTES)
```


アプライ・プログラムがサブスクリプション・セットを処理した最新の時点 (サブスクリプション・セットのコントロール表の LASTRUN 列にある値で指定) より前に発生するイベントは、期限切れのイベントと見なされて無視されます。そのため、アプライ・プログラムが実行中である場合は、期限切れのイベントを通知することを避けるため、時間的にわずかに先のイベントを通知する必要があります。

サブスクリプション・セット内におけるソース表およびビューのターゲット表およびビューへのマッピング

サブスクリプション・セット内で、アプライ・プログラムがそのセットの処理中にグループとして処理するソースからターゲットへのマッピングを追加できます。ソースからターゲットへのこれらのマッピングを、サブスクリプション・セット・メンバーと呼びます。サブスクリプション・セット・メンバーを定義するとき、ソース・データをサブスクライブするターゲット表またはビューを指定し、複製されたデータのターゲットでの表示方法を定義できます。

前提条件:

ソースでの変更をサブスクライブするターゲットをセットアップする前に、ソースとして使用する表またはビューを登録する必要があります。レプリケーション用のソースをまだ登録していない場合は、637 ページの『表およびビューをレプリケーション・ソースとして登録する』の説明を読んでそれにしたがってください。さらに、サブスクリプション・セットを作成したり、セットに追加するメンバーの数を計画する必要もあります。サブスクリプション・セットの作成が必要な場合は、690 ページの『サブスクリプション・セットの作成』を参照してください。サブスクリプション・セット・メンバーの計画が必要な場合は、688 ページの『サブスクリプション・セット・メンバー数の計画』を参照してください。

制限:

- DB2 レプリケーションは非 DB2 リレーショナル表のビューをソースとしてサポートする。
- ターゲット・ビューを定義する場合、そのビューは挿入可能なビューでなければならない。すなわち、ビューのすべての列は更新可能でなければならない、ビューの全選択にキーワード UNION ALL を組み込むことができません。
- レプリケーション・センターを使用する場合、該当する列がターゲット表にまだ存在していなければ、サブスクリプション・セット・メンバーにその列を追加できない。
- **Windows、UNIX、z/OS の場合:** それぞれのサブスクリプション・セットごとに、最大 200 のメンバーを定義できます。
- **OS/400 の場合:** それぞれのサブスクリプション・セットごとに、最大 78 のメンバーを定義できます。

手順:

サブスクリプション・セット・メンバーを追加するのに、以下の 2 つの方法のどちらかを使用できます。

レプリケーション・センター

以下のノートブックのいずれかを使用します。

- サブスクリプション・セットの作成。サブスクリプション・セットを作成するときは、このノートブックを使用します。
- サブスクリプション・セット・プロパティ。すでにサブスクリプション・セットを作成済みで、このセットに 1 つ以上のサブスクリプション・セット・メンバーを追加する場合は、このノートブックを使用します。
- サブスクリプション・セットにメンバーを追加する。1 つのメンバーを複数のサブスクリプション・セットに追加するには、このノートブックを使用します。たとえば、このノートブックを開いて 4 つのサブスクリプション・セットを選択した場合、1 つのメンバーをそれぞれのセットに追加できます。各メンバーは同じソースを使用する必要があります。

詳細については、レプリケーション・センターのオンライン・ヘルプを参照してください。

レプリケーション用のシステム・コマンド (OS/400)

ADDDPRSUBM システム・コマンドを使用します。このコマンドの構文およびパラメーターについての記述は、557 ページの『ADDDPRSUBM: DPR サブスクリプション・セットのメンバーの追加 (OS/400)』を参照してください。

ソースをターゲットにマップするには、ソースとして使用する登録済みの表またはビューについて、以下の情報を指定します。

- ソース表またはビュー、およびターゲット表またはビュー (ターゲット表の表スペースと索引も含まれる)。
- ターゲット表のタイプ。
- ターゲット表に複製するソース表の登録済みの列。

レプリケーション・センターを使用してソースをターゲットにマップするとき、LOB 列と DATALINK 列は列マッピングに自動的に組み込まれません。これらの列は明示的に選択する必要があります。

- ターゲット表に複製するソース表の行 (行を指定するには WHERE 文節を組み込む)。

選択したソースを DB2 ターゲットにマップするには、ターゲット表またはビューについて、以下の情報を指定します。

- ターゲット表またはビューのスキーマ。
- ターゲットとして使用する表またはビューの名前。

デフォルト: デフォルト名は、ターゲット・サーバーのターゲット・オブジェクト・プロファイルがあれば、それからもたらされます。このプロファイルを設定していな

いと、デフォルトは TG にソース表またはビューの名前が続いたものになります。(たとえば、ソース表の名前が EMPLOYEE なら、ターゲット表のデフォルト名は TGEMPLOYEE になります)

- ターゲット表のタイプ

デフォルト: ユーザー・コピー

指定したターゲット表が存在しない場合、レプリケーション・センターまたは **ADDDPRSUBM** システム・コマンドを使用して作成します。

選択したソースを非 DB2 リレーショナル・ターゲットにマップするには、ターゲット表について、以下の情報を指定します。

- ターゲット表のニックネームのスキーマ
- ターゲット表のニックネーム
- リモート・スキーマ
- リモート表の名前

デフォルト: デフォルト名は、ターゲット・サーバーのターゲット・オブジェクト・プロファイルがあれば、それからもたらされます。このプロファイルを設定していないと、デフォルトは TG にソース表またはビューの名前が続いたものになります。(たとえば、ソース表の名前が EMPLOYEE なら、ターゲット表のデフォルト名は TGEMPLOYEE になります)

- ターゲット表のタイプ

デフォルト: ユーザー・コピー

サブスクリプション・セット・メンバーを追加するとき、ターゲット表のデフォルト・タイプであるユーザー・コピーが使用できます。あるいはレプリケーションの要求を満たす別のターゲット表タイプを選択することができます。

まだ存在しないターゲット表のサブスクリプション・セット・メンバーを追加するとき、デフォルトの設定値を使用することができます。あるいは、メンバーのプロパティを、レプリケーションの要求を満たすように変更できます。まず、使用するターゲット表のタイプを選出してから、アプライ・プログラムがそのターゲットにデータを複製する方法に合うようにプロパティを設定することができます。多様なレプリケーションのシナリオの記述と、それぞれのケースで使用するターゲット表のタイプについては、704 ページの『ターゲット・タイプの選択』を参照してください。このセクションは、レプリケーションのゴールに基づく設定値選択全般を通じてのガイドとしても役立ちます。選択するターゲット・タイプにかかわらず、すべてのメンバーが共有するプロパティの共通セットを変更できます。サブスクリプション・セット・メンバー用のオプションとそのデフォルト、およびデフォルトの使用または変更が必要な場合についての説明の完全なリストは、716 ページの『すべてのターゲット表タイプに共通のプロパティ』を参照してください。

ターゲット・タイプの選択

このセクションでは、選択可能なターゲット表の各タイプについて記述し、選択するターゲット表のタイプと、レプリケーションのゴールに基づいたターゲット表のプロパティの定義方法の決定を支援します。また、既存の表をターゲットとして使用する場合に必要なことについても説明します。必要なターゲット表のタイプは、ターゲットでのデータ表示方法と、使用しているレプリケーション構成に左右されます。既存の表をターゲットとして使用できますが、新規の表も作成できます。

すべての非 DB2 のリレーショナル・ターゲット表と索引の名前は、DB2 の表と索引の命名規則に従わなければなりません。

制限:

- 変更後イメージ・ターゲットの列の NULL 属性は、ソース表またはビューのその列の NULL 属性と互換性がなければならない。既存の列に互換性を持たせるには、SQL の COALESCE 式を使用します。
- 非 DB2 リレーショナル・データベースのソース表の場合、ターゲット表は以下のタイプのみ定義できる。
 - ユーザー・コピー表
 - ポイント・イン・タイム表
 - 外部 CCD 表
- RRN 列をキー列として使用する OS/400 システムのソース表の場合、ターゲット表は以下のタイプのみ定義できる。
 - ポイント・イン・タイム表
 - 外部 CCD 表
- z/OS サブシステムのソース表については、ユーザー・コピー表用のサブスクリプション・セットの where 文節を満たすために、アプライ・プログラムが CD 表および UOW 表を結合する場合、CD 表および UOW 表のコード化スキームは同じでなければなりません。

ターゲット表のタイプは以下から選択できます。

ユーザー・コピー

サブスクリプション・セット・メンバーで定義された列のみが組み込まれる読み取り専用ターゲット表です。ユーザー・コピー表の構造はソース表と同じにすることができ、そうでなければ、ソース列のサブセットを組み込むことができます。変更前イメージまたは算出列が含まれる場合と含まれない場合があります。

ポイント・イン・タイム

サブスクリプション・セット・メンバーで定義された列とタイム・スタンプ列が組み込まれる読み取り専用ターゲット表です。ポイント・イン・タイム表の

構造はソース表と同じにすることができ、そうでなければ、ソース列のサブセットを組み込むことができます。変更前イメージまたは算出列が含まれる場合と含まれない場合があります。

基礎集約

SQL 列関数 (SUM、AVG など) を使用して、ソース表の内容全体のサマリーを計算する読み取り専用ターゲット表です。

基礎集約表は、ソース表の内容を要約します。基礎集約表には、アプライ・プログラムが集約を実行したタイム・スタンプも組み込まれます。基礎集約表は、ソース表の状態を定期的にトラッキングする場合に使用します。

変更集約

SQL 列関数 (SUM、AVG など) を使用して、ソース表に加えられた最近の変更の内容 (CD 表または内部 CCD 表に保管されています) 全体に関するサマリーを計算する読み取り専用ターゲット表です。

変更集約表は、ソース表ではなく、CD 表または内部 CCD 表の内容を要約します。変更集約表には、変更が取り込まれた (CD 表または CCD 表に書き込まれた) 時間インターバルにマークを付けるための 2 つのタイム・スタンプも組み込まれます。レプリケーション・サイクルの合間に行われた変更 (UPDATE、INSERT、および DELETE 操作) をトラッキングする場合は、変更集約表を使用します。

CCD (整合変更データ)

レプリケーション・コントロール情報用の列が追加された読み取り専用ターゲット表です。これらの列に組み込まれるのは、ログ・レコード番号 (またはジャーナル・レコード番号)、ソース表が SQL の INSERT、DELETE、または UPDATE ステートメントを使用して変更されたかどうかの標識、および挿入、削除、更新に関連したコミット・ステートメントのログ・レコード番号とタイム・スタンプです。オプションで、変更前イメージ列と UOW 表からの列を組み込むこともできます。

レプリカ

`update-anywhere` レプリケーション用の読み取り / 書き込みターゲット表です。ユーザー・アプリケーション・プログラムまたはユーザーが直接更新できるタイプのターゲット表は、レプリカ表のみです。したがって、レプリカ表は、マスター表から、そしてローカル・アプリケーション・プログラムまたはユーザーから変更を受け取ります。レプリカ表の構造はソース表と同じにすることができ、そうでなければ、ソース列のサブセットを組み込むことができます。しかし、追加のレプリケーション・コントロール列 (タイム・スタンプなど) は組み込みません。レプリカ表は DB2 データベースでのみサポートされています。

以下のセクションで、各ターゲット・タイプごとに可能な使用法について説明します。各セクションでは、使用できるターゲット表のタイプを通じてのガイドと、レプリケーションの要求を満たすターゲット表のプロパティの設定方法のガイドを行います。

- 『読み取り専用ターゲット表の定義』
- 708 ページの『行の最終的な変更をターゲット表に複製する』
- 710 ページの『multi-tier 構成内の中間層の定義』
- 712 ページの『読み取り / 書き込みターゲットの定義 (update-anywhere)』
- 715 ページの『既存の表をターゲット表として使用する』

ターゲット表のタイプを選択したら、そのターゲット表のデフォルトの設定値を使用できます。あるいは、ターゲット表のプロパティを、レプリケーションの要求を満たすように変更できます。共通のターゲット表オプションとそのデフォルト、およびデフォルトの使用または変更が必要な場合についての説明の完全なリストは、716 ページの『すべてのターゲット表タイプに共通のプロパティ』を参照してください。

読み取り専用ターゲット表の定義: ターゲット表タイプ: ユーザー・コピー、ポイント・イン・タイム、基礎集約、変更集約、CCD

ソース・データをターゲットでどのように表示するかによって、読み取り専用ターゲット表に以下のものが含まれるように定義できます。

- 『ソース表またはビューのコピー』
- 708 ページの『変更の履歴または監査情報』
- 707 ページの『ソースにおけるデータまたは変更の算出サマリー』

ソース表またはビューのコピー: ターゲット表タイプ: ユーザー・コピー、ポイント・イン・タイム

ソース表のコピー: デフォルトでは、サブスクリプション・セット・メンバーを定義するときに、ユーザー・コピー表がターゲット・タイプとして作成されます。コピー時にターゲット表をソース表に突き合わせる場合は、このデフォルト・タイプを使用します。ユーザー・コピー表に追加のレプリケーション・コントロール列は含まれませんが、ソース表の行または列のサブセット、あるいはレプリケーションされない追加の列を入れることができます。

タイム・スタンプが含まれたソース表のコピー: ターゲットに変更が適用されたポイント・イン・タイムをトラッキングする場合は、ポイント・イン・タイムをターゲット・タイプとして選択します。ポイント・イン・タイム・ターゲットにはソース表と同じデータが含まれ、アプライ・プログラムがそれぞれの行をいつターゲットにコミットしたかを知らせるために追加されたタイム・スタンプ列もあります。タイム・スタンプ列は、当初は NULL です。ポイント・イン・タイム表には、ソース表の行または列のサブセット、あるいは複製されない追加の列を入れることができます。

ソースにおけるデータまたは変更の算出サマリー: ターゲット表タイプ: 基礎集約、変更集約

制約事項: 非 DB2 リレーショナル・ターゲットを、集約ターゲット表タイプにすることはできません。非 DB2 リレーショナル・ソースのタイプを、集約ターゲット表にすることはできません。

ソース表の内容全体またはソース表データに加えられた最新の変更のサマリーを含むターゲット表を作成することができます。集約ターゲット表タイプでは、COUNT、SUM、MIN、MAX、AVG などの集約 SQL 列関数を使用して、ターゲットの列を定義できます。これらの列には、オリジナルのソース・データは含まれません。含まれるのは、定義した SQL 関数の算出値です。アプライ・プログラムはフル・リフレッシュの間、集約を作成しません。行は、アプライ・プログラムがセットを処理する経過において付加されます。集約表を使用すると、DB2 レプリケーションが 1 行 1 行ではなくサマリー情報だけをレプリケーションできるため、ネットワーク帯域幅とターゲット表内のスペース両方を節約できるという利点があります。

ソース表の内容の要約:

基礎集約ターゲット表を使用して、各レプリケーション・サイクル内でソース表の状態をトラッキングします。基礎集約ターゲット表では、アプライ・プログラムがソース表から集約します (読み取って計算を実行します)。基礎集約表には、アプライ・プログラムが集約を実行したタイム・スタンプも組み込まれます。

基礎集約表のみが登録済みソース表のターゲットである場合、ソース表への変更をキャプチャーする必要はありません。

例: 週ごとの平均カスタマー数を知りたいとします。ソース表にカスタマーごとの行があれば、アプライ・プログラムが週ごとにソース表の行数を合計し、結果を基礎集約表に保管します。集約を毎週実行すると、ターゲット表にはその年の週ごとのカスタマー数を示す 52 のエントリーが含まれるようになります。

CD 表または CCD 表の内容の要約:

レプリケーション・サイクルの合間にソース表で行われた変更 (UPDATE、INSERT、および DELETE 操作) をトラッキングする場合は、変更集約表を使用します。変更集約ターゲット表では、アプライ・プログラムが CD 表または内部 CCD 表から集約します (読み取って計算を実行します)。変更集約表には、キャプチャー・プログラムが CD 表または CCD 表に変更を挿入した時間インターバルにマークを付けるための 2 つのタイム・スタンプも組み込まれます。

例: 毎週新たに獲得したカスタマー数 (INSERT) と、失った既存のカスタマー数 (DELETE) を知りたいとします。CD 表の挿入された行と削除された行の数を週ごとに数え、その数を変更集約表に保管します。

重要: サブスクリプション・セット・メンバーのソース表がフル・リフレッシュのみのレプリケーション用に登録済みの場合、ソースで CD 表または CCD 表が必須である変更集約ターゲット表を持つことはできません。

変更の履歴または監査情報: ターゲット表タイプ: CCD

ソース・データの監査、あるいはデータの使用状況の履歴を保持することが必要な場合があります。ターゲット・タイプとして CCD 表を使用すると、CCD 表をどのように定義するかによって決まる多様な方法で、ソースの変更履歴をトラッキングできます。たとえば、データの変更が発生したときの変更前後の比較や、ソース表への更新を行ったユーザー ID をトラッキングできます。

ソース表の履歴を保持する読み取り専用ターゲット表を定義するには、以下の属性を持つようにターゲット CCD 表を定義します。

非コンデンス

ソースの変更すべてに関するレコードを保持するには、CCD 表を非コンデンスに定義し、発生した変更ごとに 1 行が保管されるようにします。非コンデンスの表には同じキー値を持つ複数の行が含まれるため、ユニーク索引は定義しないでください。非コンデンス CCD 表は、UPDATE、INSERT、または DELETE 操作のたびに 1 つの行を保留するので、ソース表で実行された操作の履歴を保守することができます。UPDATE 操作を INSERT および DELETE 操作 (パーティション化キー列用) としてキャプチャーする場合、CCD 表は、それぞれの更新ごとに 2 つの行、すなわち DELETE に 1 行、INSERT に 1 行を指定します。

コンプリートまたは非コンプリート

CCD 表をコンプリートにするか、あるいは非コンプリートにするかを選択できます。不完全な CCD 表には当初、ソース行の完全セットは含まれていないため、不完全な CCD 表を作成してソース表への更新 (アプライ・プログラムが CCD 表への入力を開始してからの更新) の履歴を記録します。

UOW (作業単位) 列の組み込み

監査機能を向上させるには、UOW 表からの付加的な列を含めます。ユーザー指向の識別がさらに必要であれば、UOW 表で、DB2 for z/OS の関連 ID に関する列、1 次許可 ID、または OS/400 のジョブ名およびユーザー・プロファイルを使用することができます。CCD 表に組み込み可能な UOW 列についての詳細は、731 ページの『整合変更データ (CCD) 表』を参照してください。

行の最終的な変更をターゲット表に複製する: ターゲット表タイプ: 内部 CCD

ソース表で変更が頻繁に発生する場合、最後のアプライ・サイクル以降にソースで発生したコミット済みの変更を要約するために、内部 CCD 表を作成できます。キャプチャー・プログラムがログからの変更を追加するとき、CD 表は常に流動的であるため、CCD のソース変更のローカル・キャッシュが、ターゲット用のより継続的なソースとして機能します。

オリジナルのソース表が更新されるとき、キャプチャー・プログラムはソースのログでの頻繁な変更を読み取り、その変更をソースの CD 表に追加します。その CD 表から、アプライ・プログラムは CD 表での変更を読み取り、内部 CCD 表を移植します。内部 CCD 表を、最終のサイクル内で発生した CD 表のそれぞれの行ごとの最新の変更のみが含まれるように定義できます。したがって、CCD 表はアプライ・サイクルの間では静的であるため (CD 表から CCD 表に複製するアプライ・プログラムの場合)、ターゲット用のより継続的なソースになります。ソースからのコンデンス変更を行うことで、同じ行の多数の変更をターゲット表にレプリケーションしなくなるため、レプリケーションのパフォーマンス全体が改善されます。

キャプチャー・プログラムは新しい変更を常に CD 表に追加しているため、2 番目のアプライ・プログラムは CD 表の代わりに内部 CCD 表から変更を読み取ります。そのため、様々な変更を様々なターゲットに複製することはなく、ターゲット相互の同期がとれた状態を保つことができます。2 番目のアプライ・プログラムは、オリジナルのソース表をフル・リフレッシュのために使用し、内部 CCD 表を変更キャプチャー・レプリケーションのために使用します。

推奨事項:

- ソース表と内部 CCD 表との間でサブスクリプション・セット・メンバーを定義するのは、ソース表と他のターゲット表との間で他のサブスクリプション・セット・メンバーを定義する前に行う。そうするとアプライ・プログラムは、ソース表から変更を複製するために CD 表ではなく 内部 CCD 表を使用するようになります。ソース表に内部 CCD 表を定義する前に他のサブスクリプション・セット・メンバーを定義し、それらのメンバーを使用してレプリケーションを開始すると、ソース表のすべてのターゲットに対してフル・リフレッシュの実行が必要になる可能性があります。
- すべての内部 CCD 表を 1 つのサブスクリプション・セットに結合して、ソース・データベースのすべてのターゲット表相互の同期が必ずとれるようにする。
- 頻繁に変化するソース列のサブセットを他のターゲットに適用するだけの場合でも、すべての登録済みソース列が内部 CCD に複製されるというデフォルトを使用する。そうすると、オリジナルのソース表の他の登録済み列のデータを今後必要とする可能性のあるターゲット表のために、内部 CCD 表をソースとして使用できます。今後のどのようなターゲットへの変更キャプチャー・レプリケーションにおいても使用可能なのは、内部 CCD 表の列のみです。

内部 CCD 表は、レプリケーションにおける暗黙的なソースとして使用します。レプリケーション・ソースとして明示的に定義することはできません。サブスクリプション・セット・メンバーを追加する場合、オリジナルのソース表 (内部 CCD 表ではありません) をターゲット表にマップします。内部 CCD 表には以下の属性があります。

内部 CCD 表はソースの CD 表の代わりとして機能します。内部 CCD 表についての情報は、登録 (IBMSNAP_REGISTER) 表で、この内部 CCD 表のソース表と同じ列に保管されます。内部 CCD 表独自の行は、登録表内にはありません。アプライ・プログラムは、CD 表からではなく、内部 CCD 表 (存在する場合)

から変更を自動的に複製します。それぞれのレプリケーション・ソースごとに、内部 CCD 表は 1 つだけ存在できます。

ローカル

CCD 表はソース表と同じデータベース内にあります。

非コンプリート

アプライ・プログラムは、フル・リフレッシュのために内部 CCD ではなくオリジナルのソース表を使用するため、後続ターゲットにすべてのソース行の初期コピーがすでに含まれるという理由から、CCD は非コンプリートです。

コンデンス

内部 CCD はコンデンスされています。すなわち、その表にはキー値ごとに 1 行が含まれ、アプライ・プログラムは変更ごとに 1 行をアプライする代わりに、CCD 表のそれぞれの行ごとに最新の変更をアプライするようになります。

UOW 列なし

内部 CCD 表では、UOW 表の列の追加はサポートされません。UOW 列を組み込んでいるターゲット CCD 表をすでに定義している場合は、内部 CCD 表を使用できません。

重要 (update-anywhere の場合): 内部 CCD 表を定義すると、アプライ・プログラムはレプリカをターゲットとしてサブスクリプション・セットを処理するときにそれを無視します。マスター・ソースの CCD 表からレプリカへの変更をアプライします。

multi-tier 構成内の中間層の定義: ターゲット表タイプ: CCD

基本のレプリケーション・モデルは、ソースが 1 つでターゲットが 1 つ以上の 2 層モデルですが、3 (またはそれより多い) 層の構成もセットアップできます。multi-tier 構成にはソース表とターゲット表が含まれ、そのターゲット表は他のターゲット表にはソースとして機能します。

multi-tier レプリケーション環境をセットアップする 1 つの理由は、第 3 層のターゲットに継続的なソースを提供することです。変更を層 1 から層 2 の CCD 表に収集できるため、各層への変更を複製する頻度をコントロールでき、ターゲット (層 3) に複製される変更の数を削減できます。しかも、ソース・システムへのデータベース接続の多くを回避できるので、接続コストを第 2 層に移動できます。

たとえば、3 層モデルで、第 1 層 (層 1) はソース・データベース、第 2 層 (層 2) は層 1 のターゲットです。しかも層 2 はターゲットを含む第 3 層 (層 3) のソースであり、変更を 1 つまたは多数の層 3 のデータベースに配布できます。レプリケーション構成内に 2 つより多い層が含まれる場合、ソースとターゲットの両方として機能する中間層は CCD 表です。

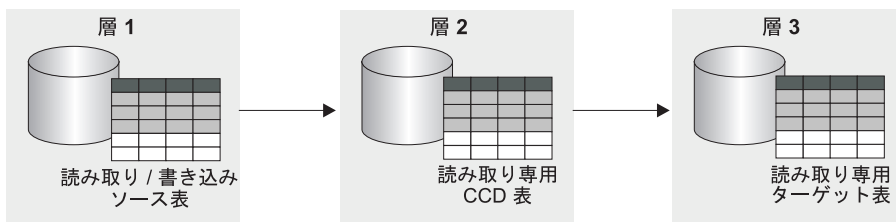


図 11. 3 層レプリケーションのモデル：データを、ソース表からターゲット表に、さらにその表から別のターゲット表に複製できます。

制約事項:

非 DB2 リレーショナル表または非 DB2 リレーショナル・データベースの CCD 表は、multi-tier 構成の中間層としては使用できません。

手順:

multi-tier レプリケーションをセットアップするには、ターゲット表が後続のターゲットへのソースとして機能するようにします。

1. ソース表 (層 1) をレプリケーション用に登録する。表をレプリケーションのために登録する方法については、638 ページの『DB2 表をソースとして登録する』を参照してください。

このソースにおいて、キャプチャー・プログラムは層 1 で発生する変更をキャプチャーし、それを層 1 の CCD 表に保管します。

2. ソース・サーバーと (層 2 の) ターゲット・サーバーとの間でサブスクリプション・セットを作成する。サブスクリプション・セットの作成方法については、690 ページの『サブスクリプション・セットの作成』を参照してください。

このサブスクリプション・セットにおいてアプライ・プログラムは、層 1 から層 2 の CCD 表に変更を適用します。

3. ソース表 (層 1) と CCD ターゲット表 (層 2) をマッピングするサブスクリプション・セット・メンバーを定義する。サブスクリプション・セット・メンバーの定義方法については、701 ページの『サブスクリプション・セット内におけるソース表およびビューのターゲット表およびビューへのマッピング』を参照してください。

このメンバーにターゲット表を定義するとき、以下の属性を持つ CCD 表になるようなターゲット表を選択します。

外部登録済みソース

外部ターゲット表として定義しなければならず、この表が後続の層のためのソースとして機能できるように登録する必要があります。他の登録済みソースと同様に、外部 CCD 表にも独自の行が登録 (IBMSNAP_REGISTER) 表内にあります。ソースとしても機能可能な外部 CCD 表は、ただ 1 つのソース表によってのみ移植されます。

すべての外部 CCD 表は、同じキャプチャー・スキーマを使用してサブスクリプション・セットに登録する必要があります。

コンプリート

後続の層に対して、アプライ・プログラムがこの表を使用してフル・リフレッシュと変更キャプチャー・レプリケーションの両方を実行するため、完全な CCD 表を使用する必要があります。

コンデンス

コンデンスされた CCD を使用します。すなわち、この表にはキー値ごとに 1 行が含まれ、必ず最新の変更のみが後続の層に複製されます。アプライ・プログラムは変更ごとに 1 行を適用する代わりに、CCD 表のそれぞれの行ごとに最新の変更を適用します。コンデンスされた表はそれぞれの行ごとにユニーク・キー値が必須であるため、ユニーク索引を定義しなければなりません。

4. CCD 表が登録されるため、中間層のデータベースにキャプチャー・コントロール表がまだない場合は作成する。
5. 登録済み CCD 表が含まれる層 2 のサーバーと (層 3 の) 後続のターゲット・サーバーとの間で、サブスクリプション・セットを作成する。サブスクリプション・セットの作成方法については、690 ページの『サブスクリプション・セットの作成』を参照してください。

このセットにおいてアプライ・プログラムは、CCD 表から後続の層のターゲット表に変更をアプライします。アプライ・プログラムは、CCD 表をフル・リフレッシュと変更キャプチャー・レプリケーションの両方に使用します。アプライ修飾子は通常、CCD にデータを入れるのに使った修飾子とは別のものを使用しますが、同一のものを使用することも可能です。

6. CCD ソース表 (層 2) と後続のターゲット表 (層 3) をマッピングするサブスクリプション・セット・メンバーを定義する。サブスクリプション・セット・メンバーの定義方法については、701 ページの『サブスクリプション・セット内におけるソース表およびビューのターゲット表およびビューへのマッピング』を参照してください。

この CCD ソース表をサブスクライブするターゲット表を組み込んで、複数のメンバーをセットアップできます。これが multi-tier 構成の最終層であれば、ターゲット表のタイプは何でも可能です。ただし、計画している層が 3 つより多い場合は、層 3 のターゲット表をステップ 3 で示されているように定義し、ステップ 4 から 5 を繰り返して後続の層を追加します。

重要: フル・リフレッシュが外部 CCD (中間層) で発生した場合、ソースとしてその外部 CCD を使用する、すべての後続の層におけるアプライ・プログラムはフル・リフレッシュを実行します。これをカスケード・フル・リフレッシュ と呼びます。

読み取り / 書き込みターゲットの定義 (update-anywhere): ターゲット表タイプ: レプリカ

update-anywhere レプリケーションでは、マスター・ソース表における変更はレプリカ・タイプの従属ターゲット表にレプリケーションされ、さらにそのレプリカ表における変更はマスター・ソース表に逆にレプリケーションされることが可能です。

update-anywhere レプリケーションで、マスター表とそのレプリカのすべては、ソースとターゲットの両方として機能する読み取り / 書き込み表です。

前提条件:

update-anywhere レプリケーションに必要な条件は以下のとおりです。

- 単一アプリケーション・プログラムはマスター表とレプリカ表の両方を更新しないため、参照保全制約を宣言して使用しなければならない。参照保全違反はアプリケーションのロジックでは検出できません。
- 参照保全違反を防止するため、複数のマスター表を対象に存在するすべての参照制約をレプリカ表に含めなければならない。いくつかの参照制約を省略すると、レプリカ表に加えられた更新がマスター表に複製される時、参照保全違反を引き起こす可能性があります。管理ツールは、ソース表からターゲット表への参照制約定義のコピーを行わず、新しい制約を生成することもできません。
- フル・リフレッシュ中の参照保全チェックをバイパスするには、ASNLOAD 出口ルーチンを使用しなければならない。

制限:

- update-anywhere レプリケーションで、CCD 表をソースまたはターゲットとして使用することはできない。
- LOB データ・タイプの列は、update-anywhere レプリケーションの対象になりません。
- DATALINK データ・タイプの列は、ソース表を競合検出しないように登録する場合を除いて、update-anywhere レプリケーションの対象にならない。
- 非 DB2 データベースのタイプをレプリカ・ターゲット表にすることはできない。そのため、update-anywhere レプリケーションの対象にできません。

手順:

マスター表と 1 つ以上のレプリカ表との間で (それぞれのレプリカ表が別々のデータベースにある場合) update-anywhere 構成をセットアップするには、以下のようになります。

1. キャプチャー・プログラムはそれぞれのレプリカ表ごとに変更をキャプチャーするため、レプリカ表を入れるデータベースそれぞれにキャプチャー・コントロール表がまだない場合は作成する。
2. ソース表 (マスター表) をレプリケーションのために登録する。表をレプリケーションのために登録する方法については、638 ページの『DB2 表をソースとして登録する』を参照してください。

このソースにおいて、キャプチャー・プログラムはマスター表での変更をキャプチャーし、それをマスターの CD 表に保管します。

3. 1 つ以上のレプリカを入れるマスター・データベースとターゲット・データベースとの間に、サブスクリプション・セットを作成する。サブスクリプション・セットの作成方法については、690 ページの『サブスクリプション・セットの作成』を参照してください。

すべてのレプリカ表が同じデータベースに、そしてすべてのマスター表が別のデータベースに含まれている場合、サブスクリプション・セットは 1 つだけ必要です。レプリカ表が複数のデータベースに含まれている場合、保有するレプリカ・データベースと同数のサブスクリプション・セットが必要です。

4. マスター表とそのおのおのに関連したレプリカ表との間のマッピングごとに、サブスクリプション・セット・メンバーを定義する。サブスクリプション・セット・メンバーの定義方法については、701 ページの『サブスクリプション・セット内におけるソース表およびビューのターゲット表およびビューへのマッピング』を参照してください。

この構成では、レプリカ表が含まれたサーバーで通常実行されるアプライ・プログラムは 1 つしかありません。このセットにおけるアプライ・プログラムは、マスターの CD 表から変更をプルして、それをレプリカ表に適用します。また、アプライ・プログラムはレプリカ表の CD 表から変更をプッシュして、それをマスター表に適用します。

重要: update-anywhere 構成において、マスター表とレプリカ表はデータを往復させて互いに複製するため、レプリカ・ターゲット表にはソース表と同じ列が含まれている必要があります。脱落した列がマスター・サイトで NULL 可能、または NOT NULL WITH DEFAULT として定義されている場合に限り、マスター表の列のサブセットが含まれるレプリカ・ターゲットを作成できますが、レプリカで新しい列の追加または列の名前変更は行わないでください。

5. レプリカ表にソースのプロパティを定義する。

レプリカ表を組み込んでサブスクリプション・セット・メンバーを作成するとき、DB2 レプリケーションはそのレプリカ表をレプリケーション・ソースとして自動的に登録します。レプリカ・ターゲット表はソースとして機能するため、ターゲット表に共通するプロパティとは別に設定可能なプロパティがあります。これらのプロパティにより、キャプチャー・プログラムがレプリカへの変更をハンドルする方法が決定されます。ただし、マスター表から継承されても、レプリカ表に合うように変更できないプロパティが 2 つあります。競合検出レベルとフル・リフレッシュの可否のプロパティです。このソースにおいて、キャプチャー・プログラムはレプリカ表での変更をキャプチャーし、それをレプリカの CD 表に保管します。登録オプションとそれらのデフォルトの完全なリスト、およびこれらのデフォルトを使用した方がよい場合と変更した方がよい場合についての説明は、641 ページの『ソース表用の登録オプション』を参照してください。

重要: マスターとレプリカがソースとターゲットの両方として機能するとしても、フル・リフレッシュ・コピーは、マスターからレプリカへのみ行われ、レプリカからマスターへは行われません。

競合を防止するため、レプリカ表のターゲット・キーを、マスター・ソース表の主キーまたはユニーク索引と同じにする必要があります。マスター表はレプリカを更新でき、レプリカはマスターを更新できるため、アプライ・サイクルの合間にマスター表のある行が更新され、1 つ以上のレプリカ表の同じ行にそれとは異なる更新が加えられた場合 (変更がマスター CD 表とレプリカ CD 表で行われます)、競合が発生する可能性があります。レプリカ表は競合検出レベルをマスター・ソース表またはビューから継承します。最善なのは、マスターからすべてのレプリカ表へデータを複製するときに、競合が決して発生しないようにアプリケーションを設計することです。マスター・ソースを登録したとき、競合検出のレベルを 3 つの選択肢から選びました。競合検出レベルの選択と、(標準または拡張競合検出を選択した場合に) 発生する競合への対応方法についての詳細は、655 ページの『競合検出の設定 (update-anywhere レプリケーション)』を参照してください。

ソース表における参照保全制約を定義した場合、保全違反を防止するために、同じ参照保全制約をレプリカ表にも定義する必要があります。参照保全違反が発生した場合、サブスクリプション・サイクルは自動的に再試行されます。

既存の表をターゲット表として使用する: 事前定義した DB2 表を、サブスクリプション・セットでのターゲット表として使用できます。すなわち、DB2 レプリケーションの外部で定義されているターゲット表を組み込むサブスクリプション・セット・メンバーを定義することができます。このようなユーザー定義のターゲット表は、表の構造が有効である限り、レプリケーションで有効なターゲット表タイプ (ユーザー・コピー、ポイント・イン・タイム、基礎または変更集約、CCD、あるいはレプリカ) のいずれかになります。たとえば、ユーザー定義のポイント・イン・タイム表には、IBMSNAP_LOGMARKER という TIMESTAMP タイプの列が組み込まなければなりません。

要件:

- サブスクリプション・セット・メンバーの定義に含まれる列が既存のターゲット表より少ない場合、レプリケーションに関連するターゲット表の列は NULL が許可されているか、あるいは NOT NULL WITH DEFAULT として定義されていなければならない。
- ポイント・イン・タイム CCD 表、ユーザー・コピー CCD 表、レプリカ CCD 表、およびコンデンス CCD 表にはユニーク索引が必要である。既存のターゲット表を使用してサブスクリプション・セット・メンバーを定義するときは、既存のユニーク索引を使用するか、または新規のユニーク索引を指定することができます。

制限:

- サブスクリプション・セット・メンバーの定義に、既存のターゲット表より多くの列を含めることはできない。
- レプリケーション・センターを使用する場合、該当する列がターゲット表にまだ存在していなければ、サブスクリプション・セット・メンバーにその列を追加できない。

DB2 レプリケーションは、既存のターゲット表とサブスクリプション・セット・メンバー定義との間の矛盾をチェックします。

重要 (multi-tier の場合): ソース表が層 1、CCD 表が層 2、そして既存の表が層 3 であるような multi-tier 構成をセットアップする場合、層 1 と層 2 との間でサブスクリプション・セット・メンバーを定義するときに、 CCD 表を既存のターゲット表に指定されている属性に一致するように定義します。そして、この CCD 表をソース表とする既存のターゲット表のサブスクリプション・セット・メンバーを定義します。

すべてのターゲット表タイプに共通のプロパティ

このセクションでは、ターゲット表を作成するときにタイプに関係なく設定できる共通のプロパティについて説明します。必要なレプリケーションのタイプを基にして、使用するターゲット表またはビューに合うようにプロパティを変更することができます。以下のセクションでは、ソース・データをターゲット表にマップする方法に応じて定義できる共通な特性について説明します。

- 『ターゲットに適用するソース列』
- 『ターゲットに適用するソース行』
- 718 ページの『ソース列からターゲット列にマップする方法』
- 718 ページの『ターゲット・キー』
- 720 ページの『アプライ・プログラムがターゲット・キー変更オプションを使用してターゲット・キー列を更新する方法』

ターゲットに適用するソース列: デフォルト: すべての登録済みソース列はターゲットに複製されます。

レプリケーションのシナリオによっては、すべての列をターゲット表にレプリケーションしたいわけではない場合があります。また、ソース表に定義したすべてのデータ・タイプでターゲット表がサポートできないものがある場合もあります。ソース表より列数が少ない列 (垂直) のサブセットを定義することができます。

デフォルトでは、ターゲット表にはソース表の LOB 列と DATALINK 列を除くすべての登録済みの列が含まれます。ソース表に存在する列で、ターゲット表に入れたくないものがある場合は、ターゲット表に複製したいソース列のみを選択します。選択しないソース表の登録済みの列は、他のサブスクリプション・セット・メンバーに引き続き使用できますが、現行のソースからターゲットへのマッピングには組み込まれません。

算出列をターゲット表に追加することもできます。これらの列は、SUBSTR のような SQL スカラー関数で定義するか、あるいは列 A の値を列 B の値で除算する (colA/colB) ような派生列にすることができます。これらの算出列は、ソース表のすべての列を参照します。

ターゲットに適用するソース行: デフォルト: すべてのソース行はターゲットに複製されます。

デフォルトでは、ターゲット表にはソース表のすべての行が含まれます。レプリケーションのシナリオによっては、すべての行をソース表からターゲット表にレプリケーションしたいわけではなかったり、あるいは異なる種類のデータを含むソース行を異なるターゲット表にレプリケーションしたいという場合があります。一定の条件 (SQL WHERE 文節) と一致する行が含まれる行 (水平方向) サブセットを定義できます。SQL の述部には、通常 ID または区切り ID を入れることができます。WHERE 文節についての詳細は、「DB2 SQL リファレンス」を参照してください。

例:

- ターゲット表が、社内のある業務部門の業務データを格納したものであると想定します。WHERE 文節をサブスクリプション・セット・メンバーで定義すると、その部門 (またはその部門内のすべての部) のすべての行を、ソース表からターゲット表に複製できます。
- 同じデータベース内に幾つかのターゲット表があると想定する。WHERE 文節を 1 つのサブスクリプション・セット・メンバーで定義すると、すべての LOB 列 (主キー列も加えて) を 1 つのターゲット表に複製でき、WHERE 文節を別のサブスクリプション・セット・メンバーで定義すると、他の列すべてを別々のターゲット表に複製できます。したがって、ターゲット・データベースはソース表からのデータすべてを保有できますが、ターゲット・データベースのソース表を非正規化して、データウェアハウスへの照会パフォーマンスを調整します。

行の述部の制約:

- この文節に WHERE を入力しない。これは暗黙で指定されます。副選択ステートメントの文節にのみ WHERE を入力してください。
- 文節をセミコロン (;) で終わらせない。
- WHERE 文節にブール式 OR を含める場合、たとえば、(COL1=X OR COL2=Y) のように、述部を括弧で囲む。
- ターゲット表が変更集約表であって、変更前イメージ列を含んでいる場合、変更前イメージ列を GROUP BY 文節に含めなければならない。

次に示す例には、ターゲット表の行をフィルターに掛けるために使用できる WHERE 文節が含まれます。これらの例は一般的であり、モデルとして使用するためのものです。

- 特定の値を持つ行を指定する WHERE 文節
特定の値 (たとえば、管理職にある社員を表す MGR) をもつ行だけをコピーするには、以下のような WHERE 文節を使用します。
EMPLOYEE = 'MGR'
- ある範囲の値を持つ行を指定する WHERE 文節
ある範囲の値 (たとえば従業員番号 5000 ~ 7000) の行をターゲット表にコピーするには、以下のような WHERE 文節を使用します。
EMPID BETWEEN 5000 AND 7000

ソース列からターゲット列にマップする方法: デフォルト: ソース列名は同じターゲット列名にマップします (ターゲット表がまだ存在しない場合)。

デフォルトでは、ターゲット表 (存在していない場合) の列名はソース表の列名と一致し、ソース列のデータ値は同じ名前のターゲット列に複製されます。レプリケーション・コントロール列 (IBMSNAP または IBMQSQ で始まる) を除くターゲット表のすべての列の名前を変更できます。ターゲット表が存在していれば、それぞれの列を明示的にマップする必要があります。

DB2 表を、既存のニックネームを持つ非 DB2 リレーショナル表にマップする場合、一部の列のデータ・タイプに互換性がないことがあります。ソース列のデータ・タイプにターゲット列のデータ・タイプとの互換性がない場合、以下のようにターゲットでデータ・タイプを変更して、ソースとの互換性を持たせることができます。

- ソースのデータ・タイプがターゲットで必要とするデータ・タイプに一致するように調整するための算出列を追加できる。
- 非 DB2 リレーショナル・ターゲット表のニックネームを変更して、データ・タイプ変換を変更できる。

例: データ・タイプが DATE の DB2 列を含む DB2 ソース表から、データ・タイプが DATE の Oracle 列を含む Oracle ターゲット表にデータを複製するとします。

表 172. DB2 DATE 列の Oracle DATE 列へのマッピング

DB2 列	ニックネーム・データの マッピング	Oracle 列
A_DATE DATE	A_DATE TIMESTAMP A_DATE DATE	A_DATE DATE

Oracle データ・タイプ DATE を含む Oracle ターゲット表が作成されます (日付データとタイム・スタンプ・データの両方が入ります)。連合データベースにある Oracle の DATE データ・タイプの初期のニックネームは、DB2 データ・タイプを TIMESTAMP としてマップします。DB2 レプリケーション・センターとレプリケーション用の OS/400 システム・コマンドは、ニックネームのデータ・タイプを DATE に変更するので、DATE は Oracle にレプリケーションされますが、TIMESTAMP はされません。

ターゲット・キー: デフォルト索引名: デフォルト名は、ターゲット・サーバーのターゲット・オブジェクト・プロファイルがあれば、それからもたらされます。このプロファイルを設定していないと、デフォルトは「IX + ターゲット表の名前」となります。たとえば、ターゲット表の名前が TGEEMPLOYEE なら、ターゲット表の索引のデフォルト名は IXTGEEMPLOYEE になります。

コンデンス・ターゲット表が変更キャプチャー・レプリケーションに関係している場合、アプライ・プログラムは、そのターゲット表にターゲット・キー と呼ばれる主キー

またはユニーク索引があることを必要とします。ターゲット表のユニーク索引として使用する列は選択できます。以下のタイプのターゲット表はコンデンスされており、ターゲット・キーが必須です。

- ユーザー・コピー
- ポイント・イン・タイム
- レプリカ
- コンデンス CCD

ターゲット表を新規に作成する場合は、デフォルトの索引名とスキーマを使用できますが、デフォルトをユーザーの命名規則に一致するように変更することもできます。

新しいターゲット表のユニーク索引を作成するためのオプションが 2 つあります。

- ターゲット表のユニーク索引として使用する列を指定する。
- DB2 レプリケーションでユーザー向けのユニーク索引を選択する。

ユニーク索引用の列を選択しないと、DB2 レプリケーションではソース表に以下の定義のいずれかがあるかどうかをこの順序でチェックします。

1. 主キー
2. ユニーク制約
3. ユニーク索引

DB2 レプリケーションで、ソース表のこれらの定義のいずれかが検出され、関連した列が登録されていてターゲット表の一部であることが検出された場合、DB2 レプリケーションはソース表の主キー（またはユニーク索引か RRN）をターゲット・キーとして使用します。ユニーク制約の場合、DB2 レプリケーションは制約列を使用して、ターゲット表のユニーク索引を作成します。

主キーまたはユニーク索引を持たない OS/400 ソース表の場合は、相対レコード番号 (RRN) を一意性の因子として使用するよう表の登録を変更します。サブスクリプション・セット・メンバーを定義するときに、RRN 列をターゲット表のユニーク索引として指定します。OS/400 ソース表における RRN の定義についての詳細は、658 ページの『主キーの代わりに相対レコード番号 (RRN) を使用する (OS/400)』を参照してください。

ターゲット・キーとして RRN を使用する OS/400 システムのターゲット表の場合、これらのターゲット表を複製するには、アプライ・プログラムを OS/400 システムで実行します。

既存のターゲット表の場合、ユニーク索引を選択する必要があります。以下のオプションのいずれかを選択することができます。

- ターゲット表の既存の索引を使用する。

既存の索引を使用するには、レプリケーション・センターで、索引を表す列を選択します。レプリケーション・センターが厳密に一致するものを検出すれば、アプライ・

プログラムが使用するようにターゲット・キーを設定するだけで、検出しなかった場合は、ユニーク索引を作成してから、ターゲット・キーをアプライ・プログラムが使用するように設定します。

- ターゲット表の別の索引を作成する。

ユニーク索引がまだない場合は作成され、アプライ・プログラムが使用するようにターゲット・キーが設定されます。

重要: ソース表で更新される可能性のある列が組み込まれたターゲット表のキーを選択する場合、ターゲット・キー列に特殊な更新を加えるようアプライ・プログラムに指示する必要があります。詳細については、『アプライ・プログラムがターゲット・キー変更オプションを使用してターゲット・キー列を更新する方法』を参照してください。

アプライ・プログラムがターゲット・キー変更オプションを使用してターゲット・キー列を更新する方法:

制約事項:

- 更新を削除および挿入のペアとしてキャプチャーするために登録されたソース表において、ターゲット・キー変更オプションを使用することはできません。
- アプライ・プログラムが、ターゲット・キー列の変更前イメージを基にして、ターゲット表を更新する場合、ソース表内の式を、ターゲット表内のキー列にマップすることはできません。(すなわち、IBMSNAP_SUBS_MEMBR 表の TARGET_KEY_CHG 列にそのターゲット表に対する値 Y がある場合です。)

サブスクリプション・セット・メンバーを定義するときにターゲット・キー変更オプションを選択すると、ターゲット・キーの変更時、アプライ・プログラムはターゲット・キー列に特殊な更新を加えます。アプライ・プログラムでこれらの特殊な更新を行うためには、ターゲット表のターゲット・キー列の一部になっているソース表の列を、CD (または CCD) 表の変更前イメージ列に登録する必要があります。ターゲット・キーを構成する列の変更前イメージ値をキャプチャーするためのソース登録を定義していない場合は、異なるキーを持つターゲット表をサブスクライブする前に、登録を変更して変更前イメージ値を組み込まなければなりません。

ターゲット・キー列の変更前イメージ値が CD (または CCD) 表にあることを確認後、ターゲット・キー列の更新時にアプライ・プログラムが変更前イメージ値を使用するように、サブスクリプション・セット・メンバーのオプションを選択します。

ターゲット・キー列の更新時にアプライ・プログラムが変更前イメージ値を使用するように指定しないと、ターゲット・キーの一部となっているソース表の列を更新しても、DB2 レプリケーションはデータを正しくレプリケーションしません。アプライ・プログラムは、新しい値を使用してターゲット表の行の更新を試みますが、更新のための新しいキー値がターゲット表で検出されません。そこでアプライ・プログラムはこの更新を INSERT に変換して、新しいキー値をターゲット表に挿入します。この場合、古いキー値をもつ古い行はターゲット表内に残ります。これは不要な行です。 ターゲット・キー

列が変更前イメージ値を使用して処理されるような変更を指定すると、アプライ・プログラムは古いキー値を検出でき、それを削除して新しい値を挿入します。

関連概念:

- レプリケーションのガイドおよびリファレンス の『DB2 レプリケーション・センターの使用』

関連タスク:

- 637 ページの『表およびビューをレプリケーション・ソースとして登録する』
- レプリケーションのガイドおよびリファレンス の『レプリケーション環境におけるデータのサブセット化』
- レプリケーションのガイドおよびリファレンス の『z/OS におけるユニコードおよび ASCII のコード化スキーム』

関連資料:

- 557 ページの『ADDDPRSUBM: DPR サブスクリプション・セットのメンバーの追加 (OS/400)』
- 580 ページの『ADDDPRSUB: DPR サブスクリプション・セットの追加 (OS/400)』
- 731 ページの『整合変更データ (CCD) 表』

レプリケーション環境におけるデータ操作

ターゲット表のデータは、ソース表における表示ほど正確に表示される必要はありません。ソース・データは、ターゲット表に複製される前にトランスフォームまたは拡張することができます。たとえば、データ整理の実行、データ集約の実行、あるいはソースにないターゲット表の列の移植のようなデータ操作を行う場合があります。

本章では、データをトランスフォームするために使用できる幾つかの先進的な技法について説明します。

データ操作は、登録済みソースからのキャプチャー前、キャプチャー後のいずれかで行うことができます。データを 1 回操作して、トランスフォームされたデータを多数のターゲット表に複製する場合は、サブスクリプションではなく登録で操作します。すべてのソース・データをキャプチャーし、トランスフォームされたデータを選択して個々のターゲットに適用する場合は、登録ではなくサブスクリプション中にデータを操作します。

レプリケーションのシナリオによっては、CD 表に保管されたソース・データの内容を操作することになります。トリガー、サブスクリプション中の式、あるいはソース・ビューのどれを使用しても、同じジョブを実行できます。それぞれの方式に、良い点と悪い点があります。トリガーは、使用される CPU の点からコストがかかりすぎる可能性があります。関数をセットアップするのに、サブスクリプションは複数回必要ですが、ビューを使用すると 1 回でセットアップできます。

たとえば、ソース表で特定の値が脱落している場合、キャプチャー・プログラムで NULL 値をキャプチャーする必要のない場合があります。

CD 表に対してトリガーを使用して、データを CD 表に挿入するときにキャプチャー・プログラムがデータを拡張する条件を指定できます。この場合、キャプチャー・プログラムがソースで NULL 値を発見したら、デフォルト値を CD 表に挿入するように指定できます。データがソース表の更新から脱落している場合、以下のコードを使用して、確定したデフォルトを提供するトリガーを作成できます。

```
CREATE TRIGGER ENHANCECD
NO CASCADE BEFORE INSERT ON CD_TABLE
REFERENCING NEW AS CD
FOR EACH ROW MODE DB2SQL
WHEN (CD.COL1 IS NULL)
SET CD.COL1 = 'MISSING DATA'
END
```

トリガーの代わりに、登録済みソース・ビューまたはサブスクリプションの式で DB2 の COALESCE スカラー関数を使用できます。登録済みビューで、この合体関数が最初の非 NULL 値を戻します。

ソース・ビューの使用例の一部:

```
CREATE VIEW SAMPLE.SRCVIEW (columns) AS SELECT
... COALESCE(A.COL1, 'MISSING DATA') ...
FROM SAMPLE.TABLE A
```

式の使用例の一部:

```
COALESCE(CD.COL1, 'MISSING DATA')
```

アプライ・プログラムは、データをターゲットに適用する前または後どちらかに、以下の方法でデータを操作できます。

- 『ストアド・プロシージャまたは SQL ステートメントを使用したデータ拡張』
- 723 ページの『名前が異なるソース列とターゲット列のマッピング』
- 723 ページの『算出列の作成』

ストアド・プロシージャまたは SQL ステートメントを使用したデータ拡張

サブスクリプション・セット情報を定義するとき、特定のセットを処理するたびにアプライ・プログラムに実行させる実行時処理ステートメントを、SQL ステートメントまたはストアド・プロシージャを使って定義できます。これらの実行時処理により、レプリケーション中にデータを操作できるようになります。このようなステートメントは、CCD 表のプルーニングやサブスクリプション・セット処理でのシーケンス・コントロールにも役立ちます。実行時処理ステートメントは、サブスクリプション・セットの処理前ならキャプチャー・コントロール・サーバーで、あるいはサブスクリプショ

ン・セットの処理前または後であればターゲット・サーバーで実行できます。たとえば、データを取り出す前、またはターゲット表にデータを複製した後、あるいはその両方の場合に SQL ステートメントを実行することができます。

ニックネームの制限: 通常、連合 DB2 表 (ニックネームを使用) は、単一の作業単位で更新されるため、アプライ・プログラムがすべてのデータをターゲットに適用した後、SQL ステートメントをサブスクリプション・セットに追加する場合、以下の 2 つの状況のいずれかで、SQL COMMIT ステートメントをその SQL ステートメントより先行させる必要があります。

- この SQL ステートメントは、サブスクリプション・セットのターゲット表またはターゲット・ニックネームがあるサーバー以外のサーバー上のニックネームに対して挿入、更新、または削除を行う。
- この SQL ステートメントは、アプライ・コントロール・サーバーに対してローカルの表に対して、挿入、更新、または削除を行うが、サブスクリプション・セットのターゲット・ニックネームがリモート・サーバーにある。

追加の COMMIT ステートメントは、追加された SQL ステートメントを処理する前に、アプライ・プログラムの作業をコミットします。

ストアド・プロシージャは、パラメーターなしの SQL CALL ステートメントを使用します。プロシージャ名の長さは、18 文字以下でなければなりません (OS/400 の場合は、最大 128 文字です)。ソースまたはターゲット表が非 DB2 リレーショナル・データベースにある場合、SQL ステートメントは連合 DB2 データベースを対象として実行されます。SQL ステートメントが非 DB2 データベースを対象として実行されることは決してありません。各タイプの実行時プロシージャは単一トランザクションとして一緒に実行されます。さらに、それぞれのステートメントごとに受け入れ可能 SQLSTATE を定義することもできます。

(特定のセットの完了後ではなく) それぞれのセットの完了後にデータを操作する場合は、ASNDONE 出口ルーチンを使用します。

名前が異なるソース列とターゲット列のマッピング

レプリケーション・センターを使用してターゲット表を作成する際に、ターゲット表のタイプに関係なく、ターゲット表で列の名前を変更できます。さらに、列の属性 (データ・タイプ、長さ、位取り、精度、NULL 可能かどうか) を互換性があれば変更もできます。レプリケーション・センターを使用して、既存のターゲット表の列を名前変更することはできません。ソース列とターゲット列が一致しない場合は、レプリケーション・センターを使用してソースからターゲットに列をマップするか、あるいはソースの列名に一致する名前を含むターゲット表のビューを作成するかのどちらかが可能です。

算出列の作成

既存のターゲット表の列名を変更することはできませんが、ソース列の式を変更して、既存のターゲット表の列に正しくマップするようしたり、あるいはそれとの互換性を持たせることができます。SQL 式を使用すれば、既存のソース列から新しい列を派生

させることもできます。集約ターゲット表タイプでは、COUNT や SUM といった集約関数を使って新しい列を定義できます。ターゲット表の他のタイプでは、式でスカラー関数を使用して新しい列を定義できます。ソース表とターゲット表の列の名前だけが違って、他は互換性がある場合、レプリケーション・センターを使用すれば列をもう一方の表の列にマップできます。

たとえば、既存のソース表 (SRC.TABLE) とターゲット表 (TGT.TABLE) があるとします。

```
CREATE TABLE SRC.TABLE (SRC_COL1 CHAR(12) NOT NULL, SRC_COL2 INTEGER,  
SRC_COL3 DATE, SRC_COL4 TIME, SRC_COL5 VARCHAR(25))
```

```
CREATE TABLE TGT.TABLE (TGT_COL1 CHAR(12) NOT NULL,  
TGT_COL2 INTEGER NOT NULL, TGT_COL3 TIMESTAMP, TGT_COL4 CHAR(5))
```

以下のステップで、サブスクリプション中に算出列を使用して必要なターゲット表をマップします。

1. レプリケーション・センターを使用して、ソース表の SRC_COL1 をターゲット表の TGT_COL1 にマップする。これらの列には互換性があるので、一方から他方にマップするのに式を使う必要はありません。
2. 式 COALESCE(SRC_COL2, 0) を使用して列値を計算し、TGT_COL2 に提供するためにマップする。SRC_COL2 は NULL 可能で TGT_COL2 は NOT NULL のため、TGT_COL2 の値が必ず非 NULL になるようにこのステップを実行する必要があります。
3. 式 TIMESTAMP(CHAR(SRC_COL3) CONCAT CHAR(SRC_COL4)) を使用して列値の計算をし、TGT_COL3 に提供するためにマップする。この列式はターゲット・データベースのタイム・スタンプ列にマップするデータを提供します。
4. 式 SUBSTR(SRC_COL5, 1,5) を使用して列値の計算をし、TGT_COL4 に提供するためにマップする。

表の概観

725 ページの図 12 および 726 ページの図 13 は、キャプチャー・コントロール・サーバー上の表、各表の列、そして各表の索引を示しています。728 ページの図 15 および 727 ページの図 14 は、アプライ・コントロール・サーバー上の表、各表の列、そして各表の索引を示しています。729 ページの図 16 は、モニター・コントロール・サーバー上の表、各表の列、そして各表の索引を示しています。

Control tables used at the Capture control server (image 1 of 2)

<p><i>OS/400 only</i></p> <p>schema.IBMSNAP_AUTHTKN (JRN_LIB, JRN_NAME)</p> <p>APPLY_QUAL CHAR(18) NOT NULL IBMSNAP_AUTHTKN CHAR(26) NOT NULL JRN_LIB CHAR(10) NOT NULL JRN_NAME CHAR(10) NOT NULL IBMSNAP_LOGMARKER TIMESTAMP NOT NULL</p>	<p>ASN.IBMSNAP_CAPSCHEMAS (CAP_SCHEMA_NAME)</p> <p>CAP_SCHEMA_NAME VARCHAR(30)</p> <p><i>OS/400</i></p> <p>ASN.IBMSNAP_CAPSCHEMAS (CAP_SCHEMA_NAME)</p> <p>CAP_SCHEMA_NAME VARCHAR(30) STATUS CHAR(1)</p>
<p><i>UNIX, Windows, and z/OS only</i></p> <p>schema.IBMSNAP_CAPENQ (no index)</p> <p>LOCKNAME CHAR(9)</p>	<p>schema.IBMSNAP_CAPTRACE (TRACE_TIME)</p> <p>OPERATION CHAR(8) NOT NULL TRACE_TIME TIMESTAMP NOT NULL DESCRIPTION VARCHAR(1024) NOT NULL</p> <p><i>OS/400</i></p> <p>schema.IBMSNAP_CAPTRACE (TRACE_TIME)</p> <p>OPERATION CHAR(8) NOT NULL TRACE_TIME TIMESTAMP NOT NULL JOB_NAME CHAR(26) NOT NULL JOB_STR_TIME TIMESTAMP NOT NULL DESCRIPTION VARCHAR(298) NOT NULL</p>
<p>schema.IBMSNAP_CAPMON (MONITOR_TIME)</p> <p>MONITOR_TIME TIMESTAMP NOT NULL RESTART_TIME TIMESTAMP NOT NULL CURRENT_MEMORY INT NOT NULL CD_ROWS_INSERTED INT NOT NULL RECAP_ROWS_SKIPPED INT NOT NULL TRIGR_ROWS_SKIPPED INT NOT NULL CHG_ROWS_SKIPPED INT NOT NULL TRANS_PROCESSED INT NOT NULL TRANS_SPILLED INT NOT NULL MAX_TRAN_SIZE INT NOT NULL LOCKING_RETRIES INT NOT NULL JRN_LIB CHAR(10) JRN_NAME CHAR(10) LOGREADLIMIT INT NOT NULL CAPTURE_IDLE INT NOT NULL SYNCHTIME TIMESTAMP NOT NULL</p>	<p>schema.IBMSNAP_PRUNCNTL (SOURCE_OWNER, SOURCE_TABLE, SOURCE_VIEW_QUAL, APPLY_QUAL, SET_NAME, TARGET_SERVER, TARGET_TABLE, TARGET_OWNER)</p> <p>TARGET_SERVER CHAR(18) NOT NULL TARGET_OWNER VARCHAR(30) NOT NULL TARGET_TABLE VARCHAR(128) NOT NULL SYNCHTIME TIMESTAMP SYNCHPOINT CHAR(10) FOR BIT DATA SOURCE_OWNER VARCHAR(30) NOT NULL SOURCE_TABLE VARCHAR(128) NOT NULL SOURCE_VIEW_QUAL SMALLINT NOT NULL APPLY_QUAL CHAR(18) NOT NULL SET_NAME CHAR(18) NOT NULL CNTL_SERVER CHAR(18) NOT NULL TARGET_STRUCTURE SMALLINT NOT NULL CNTL_ALIAS CHAR(8) PHYS_CHANGE_OWNER VARCHAR(30) PHYS_CHANGE_TABLE VARCHAR(128) MAP_ID VARCHAR(10) NOT NULL</p>
<p>schema.IBMSNAP_CAPPARMS (no index)</p> <p>RETENTION_LIMIT INT LAG_LIMIT INT COMMIT_INTERVAL INT PRUNE_INTERVAL INT TRACE_LIMIT INT MONITOR_LIMIT INT MONITOR_INTERVAL INT MEMORY_LIMIT SMALLINT REMOTE_SRC_SERVER CHAR(18) AUTOPRUNE CHAR(1) TERM CHAR(1) AUTOSTOP CHAR(1) LOGREUSE CHAR(1) LOGSTDOUT CHAR(1) SLEEPINTERVAL SMALLINT CAPTURE_PATH VARCHAR(1040) STARTMODE VARCHAR(10)</p>	<p>schema.IBMSNAP_PRUNE_LOCK (no index)</p> <p>DUMMY CHAR(1)</p>

図 12. キャプチャー・コントロール・サーバーで使用される表：キャプチャー・コントロール・サーバーで、キャプチャー・プログラム、アプライ・プログラム、およびキャプチャー・トリガーから使用される表。各表の主索引を構成する列は、表名の下に括弧内に示されています。

Control tables used at the Capture control server (image 2 of 2)

<p>schema.IBMSNAP_PRUNE_SET (TARGET_SERVER, APPLY_QUAL, SET_NAME)</p> <p>TARGET_SERVER CHAR(18) NOT NULL APPLY_QUAL CHAR(18) NOT NULL SET_NAME CHAR(18) NOT NULL SYNCHTIME TIMESTAMP SYNCHPOINT CHAR(10) FOR BIT DATA NOT NULL</p>	<p>schema.IBMSNAP_REG_SYNCH (TRIGGER_ME)</p> <p>TRIGGER_ME CHAR(1) NOT NULL</p>
<p>schema.IBMSNAP_REGISTER (SOURCE_OWNER, SOURCE_TABLE, SOURCE_VIEW_QUAL)</p> <p>SOURCE_OWNER VARCHAR(30) NOT NULL SOURCE_TABLE VARCHAR(128) NOT NULL SOURCE_VIEW_QUAL SMALLINT NOT NULL GLOBAL_RECORD CHAR(1) NOT NULL SOURCE_STRUCTURE SMALLINT NOT NULL SOURCE_CONDENSED CHAR(1) NOT NULL SOURCE_COMPLETE CHAR(1) NOT NULL CD_OWNER VARCHAR(30) CD_TABLE VARCHAR(128) PHYS_CHANGE_OWNER VARCHAR(30) PHYS_CHANGE_TABLE VARCHAR(128) CD_OLD_SYNCHPOINT CHAR(10) FOR BIT DATA CD_NEW_SYNCHPOINT CHAR(10) FOR BIT DATA DISABLE_REFRESH SMALLINT NOT NULL CCD_OWNER VARCHAR(30) CCD_TABLE VARCHAR(128) CCD_OLD_SYNCHPOINT CHAR(10) FOR BIT DATA SYNCHPOINT CHAR(10) FOR BIT DATA SYNCHTIME TIMESTAMP CCD_CONDENSED CHAR(1) CCD_COMPLETE CHAR(1) ARCH_LEVEL CHAR(4) NOT NULL DESCRIPTION CHAR(254) BEFORE_IMG_PREFIX VARCHAR(4) CONFLICT_LEVEL CHAR(1) CHG_UPD_TO_DEL_INS CHAR(1) CHGONLY CHAR(1) RECAPTURE CHAR(1) OPTION_FLAGS CHAR(4) NOT NULL STOP_ON_ERROR CHAR(1) STATE CHAR(1) STATE_INFO CHAR(8)</p>	<p>schema.IBMSNAP_RESTART (no index)</p> <p>MAX_COMMITSEQ CHAR(10) FOR BIT DATA NOT NULL MAX_COMMIT_TIME TIMESTAMP NOT NULL MIN_INFLIGHTSEQ CHAR(10) FOR BIT DATA NOT NULL CURR_COMMIT_TIME TIMESTAMP NOT NULL CAPTURE_FIRST_SEQ CHAR(10) FOR BIT DATA NOT NULL</p> <p>OS/400</p> <p>schema.IBMSNAP_RESTART (JRN_LIB, JRN_NAME)</p> <p>MAX_COMMITSEQ CHAR(10) FOR BIT DATA NOT NULL MAX_COMMIT_TIME TIMESTAMP NOT NULL MIN_INFLIGHTSEQ CHAR(10) FOR BIT DATA NOT NULL CURR_COMMIT_TIME TIMESTAMP NOT NULL CAPTURE_FIRST_SEQ CHAR(10) FOR BIT DATA NOT NULL</p> <p>UID INTEGER NOT NULL SEQNBR BIGINT NOT NULL JRN_LIB CHAR(10) NOT NULL JRN_NAME CHAR(10) NOT NULL STATUS CHAR(1)</p>
<p>OS/400</p> <p>schema.IBMSNAP_REG_EXT (VERSION, SOURCE_OWNER, SOURCE_TABLE, SOURCE_VIEW_QUAL)</p> <p>VERSION INT NOT NULL SOURCE_OWNER VARCHAR(30) NOT NULL SOURCE_TABLE VARCHAR(128) NOT NULL SOURCE_NAME CHAR(10) SOURCE_MBR CHAR(10) SOURCE_TABLE_RDB CHAR(18) JRN_LIB CHAR(10) JRN_NAME CHAR(10) FR_START_TIME TIMESTAMP SOURCE_VIEW_QUAL SMALLINT NOT NULL CMT_BEHAVIOR_CASE SMALLINT NOT NULL WITH DEFAULT MAX_ROWS_BTWN_CMTS SMALLINT NOT NULL WITH DEFAULT</p>	<p>schema.IBMSNAP_SEQTABLE (SEQ)</p> <p>SEQ INTEGER NOT NULL</p>
<p>OS/400</p> <p>schema.IBMSNAP_SIGNAL (SIGNAL_TIME)</p> <p>SIGNAL_TIME TIMESTAMP NOT NULL WITH DEFAULT SIGNAL_TYPE VARCHAR(30) NOT NULL SIGNAL_SUBTYPE VARCHAR(30) SIGNAL_INPUT_IN VARCHAR(500) SIGNAL_STATE CHAR(1) NOT NULL SIGNAL_LSN CHAR(10) FOR BIT DATA</p>	<p>schema.IBMSNAP_UOW (IBMSNAP_COMMITSEQ, IBMSNAP_LOGMARKER)</p> <p>IBMSNAP_UOWID CHAR(10) FOR BIT DATA NOT NULL IBMSNAP_COMMITSEQ CHAR(10) FOR BIT DATA NOT NULL IBMSNAP_LOGMARKER TIMESTAMP NOT NULL IBMSNAP_AUTHTKN VARCHAR(30) NOT NULL IBMSNAP_AUTHID VARCHAR(30) NOT NULL IBMSNAP_REJ_CODE CHAR(1) NOT NULL WITH DEFAULT IBMSNAP_APPLY_QUAL CHAR(18) NOT NULL WITH DEFAULT</p>

図 13. キャプチャー・コントロール・サーバーで使用される表 (続き): キャプチャー・コントロール・サーバーで、キャプチャー・プログラム、アプライ・プログラム、およびキャプチャー・トリガーから使用される表。各表の主索引を構成する列は、表名の下に括弧内に示されています。

Control tables used at the Apply control server (image 1 of 2)

ASNJBMSNAP_APPLYTRAIL (LASTRUN, APPLY_QUAL)		ASNJBMSNAP_APPENQ (APPLY_QUAL)	
APPLY_QUAL	CHAR(18) NOT NULL	APPLY_QUAL	CHAR(18)
SET_NAME	CHAR(18) NOT NULL		
SET_TYPE	CHAR(1) NOT NULL		
WHOS_ON_FIRST	CHAR(1) NOT NULL		
ASNLOAD	CHAR(1)		
FULL_REFRESH	CHAR(1)		
EFFECTIVE_MEMBERS	INT		
SET_INSERTED	INT NOT NULL		
SET_DELETED	INT NOT NULL		
SET_UPDATED	INT NOT NULL		
SET_REWORKED	INT NOT NULL		
SET_REJECTED_TRXS	INT NOT NULL		
STATUS	SMALLINT NOT NULL		
LASTRUN	TIMESTAMP NOT NULL		
LASTSUCCESS	TIMESTAMP		
SYNCHPOINT	CHAR(10) FOR BIT DATA		
SYNCHTIME	TIMESTAMP		
SOURCE_SERVER	CHAR (18) NOT NULL		
SOURCE_ALIAS	CHAR(8)		
SOURCE_OWNER	VARCHAR(30)		
SOURCE_TABLE	VARCHAR(128)		
SOURCE_VIEW_QUAL	SMALLINT		
TARGET_SERVER	CHAR(18) NOT NULL		
TARGET_ALIAS	CHAR(8)		
TARGET_OWNER	VARCHAR(30) NOT NULL		
TARGET_TABLE	VARCHAR(128) NOT NULL		
CAPTURE_SCHEMA	VARCHAR(30) NOT NULL		
TGT_CAPTURE_SCHEMA	VARCHAR(30)		
FEDERATED_SRC_SRVR	VARCHAR(18)		
FEDERATED_TGT_SRVR	VARCHAR(18)		
JRN_LIB	CHAR(10)		
JRN_NAME	CHAR(10)		
COMMIT_COUNT	SMALLINT		
OPTION_FLAGS	CHAR(4) NOT NULL		
EVENT_NAME	CHAR(18)		
ENDTIME	TIMESTAMP NOT NULL WITH DEFAULT		
SOURCE_CONN_TIME	TIMESTAMP		
SQLSTATE	CHAR(5)		
SQLCODE	INT		
SQLERRP	CHAR(8)		
SQLERRM	VARCHAR(70)		
APPERRM	VARCHAR(760)		
		<i>OS/400 only</i>	
		ASNJBMSNAP_APPLY_JOB (no index)	
APPLY_QUAL	CHAR(18) NOT NULL	APPLY_QUAL	CHAR(18)
CONTROL_SERVER	CHAR(18) NOT NULL	JOB_NAME	CHAR(10) NOT NULL
JOB_NAME	CHAR(10) NOT NULL	USER_NAME	CHAR(10) NOT NULL
USER_NAME	CHAR(10) NOT NULL	JOB_NUMBER	CHAR(6) NOT NULL
JOB_NUMBER	CHAR(6) NOT NULL		
		ASNJBMSNAP_APPLYTRACE (APPLY_QUAL, TRACE_TIME)	
APPLY_QUAL	CHAR(18) NOT NULL	APPLY_QUAL	CHAR(18) NOT NULL
TRACE_TIME	TIMESTAMP NOT NULL	TRACE_TIME	TIMESTAMP NOT NULL
OPERATION	CHAR(8) NOT NULL	OPERATION	CHAR(8) NOT NULL
DESCRIPTION	VARCHAR(1024) NOT NULL	DESCRIPTION	VARCHAR(1024) NOT NULL

図 14. アプライ・コントロール・サーバーで使用される表：アプライ・コントロール・サーバーでアプライ・プログラムから使用される表。各表の主索引を構成する列は、表名の下の括弧内に示されています。

Control tables used at the Apply control server (image 2 of 2)

ASN.IBMSNAP_SUBS_COLS
(APPLY_QUAL, SET_NAME, WHOS_ON_FIRST, TARGET_OWNER, TARGET_TABLE, TARGET_NAME)

APPLY_QUAL	CHAR(18) NOT NULL
SET_NAME	CHAR(18) NOT NULL
WHOS_ON_FIRST	CHAR(1) NOT NULL
TARGET_OWNER	VARCHAR(30) NOT NULL
TARGET_TABLE	VARCHAR(128) NOT NULL
COL_TYPE	CHAR(1) NOT NULL
TARGET_NAME	VARCHAR(30) NOT NULL
IS_KEY	CHAR(1) NOT NULL
COLNO	SMALLINT NOT NULL
EXPRESSION	VARCHAR(254) NOT NULL

ASN.IBMSNAP_SUBS_STMTS
(APPLY_QUAL, SET_NAME, WHOS_ON_FIRST, BEFORE_OR_AFTER, STMT_NUMBER)

APPLY_QUAL	CHAR(18) NOT NULL
SET_NAME	CHAR(18) NOT NULL
WHOS_ON_FIRST	CHAR(1) NOT NULL
BEFORE_OR_AFTER	CHAR(1) NOT NULL
STMT_NUMBER	SMALLINT NOT NULL
EL_OR_CALL	CHAR(1) NOT NULL
SQL_STMT	VARCHAR(1024)
ACCEPT_SQLSTATES	VARCHAR(50)

ASN.IBMSNAP_SUBS_EVENT
(EVENT_NAME, EVENT_TIME)

EVENT_NAME	CHAR(18) NOT NULL
EVENT_TIME	TIMESTAMP NOT NULL
END_SYNCHPOINT	CHAR(10) FOR BIT DATA
END_OF_PERIOD	TIMESTAMP

ASN.IBMSNAP_SUBS_SET
(APPLY_QUAL, SET_NAME, WHOS_ON_FIRST)

APPLY_QUAL	CHAR(18) NOT NULL
SET_NAME	CHAR(18) NOT NULL
SET_TYPE	CHAR(1) NOT NULL
WHOS_ON_FIRST	CHAR(1) NOT NULL
ACTIVATE	SMALLINT NOT NULL
SOURCE_SERVER	CHAR(18) NOT NULL
SOURCE_ALIAS	CHAR(8)
TARGET_SERVER	CHAR(18) NOT NULL
TARGET_ALIAS	CHAR(8)
STATUS	SMALLINT NOT NULL
LASTRUN	TIMESTAMP NOT NULL
REFRESH_TYPE	CHAR(1) NOT NULL
SLEEP_MINUTES	INT
EVENT_NAME	CHAR(18)
LASTSUCCESS	TIMESTAMP
SYNCHPOINT	CHAR(10) FOR BIT DATA
SYNCHTIME	TIMESTAMP
CAPTURE_SCHEMA	VARCHAR(30) NOT NULL
TGT_CAPTURE_SCHEMA	VARCHAR(30)
FEDERATED_SRC_SRVR	VARCHAR(18)
FEDERATED_TGT_SRVR	VARCHAR(18)
JRN_LIB	CHAR(10)
JRN_NAME	CHAR(10)
OPTION_FLAGS	CHAR(4) NOT NULL
COMMIT_COUNT	SMALLINT
MAX_SYNCH_MINUTES	SMALLINT
AUX_STMTS	SMALLINT NOT NULL
ARCH_LEVEL	CHAR(4) NOT NULL

ASN.IBMSNAP_SUBS_MEMBR

(APPLY_QUAL, SET_NAME, WHOS_ON_FIRST, SOURCE_OWNER, SOURCE_TABLE, SOURCE_VIEW_QUAL, TARGET_OWNER, TARGET_TABLE)

APPLY_QUAL	CHAR(18) NOT NULL
SET_NAME	CHAR(18) NOT NULL
WHOS_ON_FIRST	CHAR(1) NOT NULL
SOURCE_OWNER	VARCHAR(30) NOT NULL
SOURCE_TABLE	VARCHAR(128) NOT NULL
SOURCE_VIEW_QUAL	SMALLINT NOT NULL
TARGET_OWNER	VARCHAR(30) NOT NULL
TARGET_TABLE	VARCHAR(128) NOT NULL
TARGET_CONDENSED	CHAR(1) NOT NULL
TARGET_COMPLETE	CHAR(1) NOT NULL
TARGET_STRUCTURE	SMALLINT NOT NULL
PREDICATES	VARCHAR(1024)
MEMBER_STATE	CHAR(1)
TARGET_KEY_CHG	CHAR(1) NOT NULL
UOW_CD_PREDICATES	VARCHAR(1024)
JOIN_UOW_CD	CHAR(1)
LOADX_TYPE	SMALLINT
LOADX_SRC_N_OWNER	VARCHAR(30)
LOADX_SRC_N_TABLE	VARCHAR(128)

図 15. アプライ・コントロール・サーバーで使用される表 (続き): アプライ・コントロール・サーバーでアプライ・プログラムから使用される表。各表の主索引を構成する列は、表名の下の括弧内に示されています。

Control tables used at the Monitor control server

ASN.IBMSNAP_ALERTS (MONITOR_QUAL, COMPONENT, SERVER_NAME, SCHEMA_OR_QUAL, SET_NAME, CONDITION_NAME, ALERT_CODE)	
MONITOR_QUAL	CHAR(18) NOT NULL
ALERT_TIME	TIMESTAMP NOT NULL
COMPONENT	CHAR(1) NOT NULL
SERVER_NAME	CHAR(18) NOT NULL
SERVER_ALIAS	CHAR(8)
SCHEMA_OR_QUAL	VARCHAR(30) NOT NULL
SET_NAME	CHAR(18) NOT NULL WITH DEFAULT
CONDITION_NAME	CHAR(18) NOT NULL
OCCURRED_TIME	TIMESTAMP NOT NULL
ALERT_COUNTER	SMALLINT NOT NULL
ALERT_CODE	CHAR(10) NOT NULL
RETURN_CODE	INT NOT NULL
NOTIFICATION_SENT	CHAR(1) NOT NULL
ALERT_MESSAGE	VARCHAR(1024) NOT NULL

ASN.IBMSNAP_GROUPS (GROUP_NAME)	
GROUP_NAME	VARCHAR(127) NOT NULL
DESCRIPTION	VARCHAR(1024)

ASN.IBMSNAP_MONENQ (MONITOR_QUAL)	
MONITOR_QUAL	CHAR(18) NOT NULL

ASN.IBMSNAP_MONSERVERS (MONITOR_QUAL, SERVER_NAME)	
MONITOR_QUAL	CHAR(18) NOT NULL
SERVER_NAME	CHAR(18) NOT NULL
SERVER_ALIAS	CHAR(8)
LAST_MONITOR_TIME	TIMESTAMP NOT NULL
START_MONITOR_TIME	TIMESTAMP
END_MONITOR_TIME	TIMESTAMP
LASTRUN	TIMESTAMP NOT NULL
LASTSUCCESS	TIMESTAMP
STATUS	SMALLINT NOT NULL

ASN.IBMSNAP_CONDITIONS (MONITOR_QUAL, SERVER_NAME, COMPONENT, SCHEMA_OR_QUAL, SET_NAME, CONDITION_NAME)	
SERVER_NAME	CHAR(18) NOT NULL
COMPONENT	CHAR(1) NOT NULL
SCHEMA_OR_QUAL	VARCHAR(30) NOT NULL
SET_NAME	CHAR(18) NOT NULL WITH DEFAULT
MONITOR_QUAL	CHAR(18) NOT NULL
SERVER_ALIAS	CHAR(8)
ENABLED	CHAR(1) NOT NULL
CONDITION_NAME	CHAR(18) NOT NULL
PARAM_INT	INT
PARAM_CHAR	VARCHAR(128)
CONTACT_TYPE	CHAR(1) NOT NULL
CONTACT	VARCHAR(127) NOT NULL

ASN.IBMSNAP_MONTRACE (MONITOR_QUAL, TRACE_TIME)	
MONITOR_QUAL	CHAR(18) NOT NULL
TRACE_TIME	TIMESTAMP NOT NULL
OPERATION	CHAR(8) NOT NULL
DESCRIPTION	VARCHAR(1024) NOT NULL

ASN.IBMSNAP_CONTACTGRP (GROUP_NAME, CONTACT_NAME)	
GROUP_NAME	VARCHAR(127) NOT NULL
CONTACT_NAME	VARCHAR(127) NOT NULL

ASN.IBMSNAP_MONTRAIL (no index)	
MONITOR_QUAL	CHAR(18) NOT NULL
SERVER_NAME	CHAR(18) NOT NULL
SERVER_ALIAS	CHAR(8)
STATUS	SMALLINT NOT NULL
LASTRUN	TIMESTAMP NOT NULL
LASTSUCCESS	TIMESTAMP
ENDTIME	TIMESTAMP NOT NULL WITH DEFAULT
LAST_MONITOR_TIME	TIMESTAMP NOT NULL
START_MONITOR_TIME	TIMESTAMP
END_MONITOR_TIME	TIMESTAMP
SQLCODE	INT
SQLSTATE	CHAR(5)
NUM_ALERTS	INT NOT NULL
NUM_NOTIFICATIONS	INT NOT NULL

ASN.IBMSNAP_CONTACTS (CONTACT_NAME)	
CONTACT_NAME	VARCHAR(127) NOT NULL
EMAIL_ADDRESS	VARCHAR(128) NOT NULL
ADDRESS_TYPE	CHAR(1) NOT NULL
DELEGATE	VARCHAR(127)
DELEGATE_START	DATE
DELEGATE_END	DATE
DESCRIPTION	VARCHAR(1024)

図 16. モニター・コントロール・サーバーで使用される表： モニター・コントロール・サーバーで、レプリケーション・アラート・モニター・プログラムから使用される表。各表の主索引を構成する列は、表名の下に括弧内に示されています。

ASN.IBMSNAP_MONTRAIL

サーバー: モニター・コントロール・サーバー

モニター・トレール表には、各モニター・サイクルに関する情報が入ります。レプリケーション・アラート・モニターは、モニター対象のキャプチャー・コントロール・サーバー、またはアプライ・コントロール・サーバーごとに行を 1 つ挿入します。

730 ページの表 173 では、モニター・トレール表の列の要旨を示します。

IBMSNAP_MONTRAIL

表 173. モニター・トレール表の列

列名	説明
MONITOR_QUAL	キャプチャー・コントロール・サーバーまたはアプライ・コントロール・サーバーをモニターしているレプリケーション・アラート・モニターを識別するモニター修飾子。
SERVER_NAME	レプリケーション・アラート・モニターによるモニター対象のキャプチャー・コントロール・サーバーまたはアプライ・コントロール・サーバーの名前。
SERVER_ALIAS	レプリケーション・アラート・モニターによるモニター対象のキャプチャー・コントロール・サーバー、またはアプライ・コントロール・サーバーの DB2 Universal Database 別名。
STATUS	モニター・サイクルの状況を示すフラグ。 -1 レプリケーション・アラート・モニターは、このサーバーを正常に処理できませんでした。 0 レプリケーション・アラート・モニターは、このサーバーを正常に処理しました。 1 レプリケーション・アラート・モニターは、このサーバーを現在処理中です。
LASTRUN	レプリケーション・アラート・モニター・プログラムが、キャプチャー・コントロール・サーバー、またはアプライ・コントロール・サーバーの処理を最後に開始したときの時刻 (モニター・コントロール・サーバーの)。
LASTSUCCESS	レプリケーション・アラート・モニターが、キャプチャー・コントロール・サーバー、またはアプライ・コントロール・サーバーの処理を最後に正常に完了したときの時刻 (モニター・コントロール・サーバーの)。
ENDTIME	この表にこの行が挿入された時刻。
LAST_MONITOR_TIME	レプリケーション・アラート・モニターが、キャプチャー・コントロール・サーバー、またはアプライ・コントロール・サーバーに最後に接続したときの時刻 (キャプチャー・コントロール・サーバーまたはアプライ・コントロール・サーバーの)。この値は、コントロール表からメッセージをフェッチする下限値として使用され、直前の成功したモニター・サイクルの START_MONITOR_TIME の値と同じ値になります。
START_MONITOR_TIME	レプリケーション・アラート・モニターが、キャプチャー・コントロール・サーバー、またはアプライ・コントロール・サーバーのモニターを最後に開始した時刻。
END_MONITOR_TIME	レプリケーション・アラート・モニターが、キャプチャー・コントロール・サーバー、またはアプライ・コントロール・サーバーのモニターを最後に終了した時刻。

表 173. モニター・トレール表の列 (続き)

列名	説明
SQLCODE	このモニター・サイクル中に発生したエラーの SQLCODE。
SQLSTATE	このモニター・サイクル中に発生したエラーの SQLSTATE。
NUM_ALERTS	このモニター・サイクル中に発生したアラート条件の数。
NUM_NOTIFICATIONS	このモニター・サイクル中に送信された通知の数。

整合変更データ (CCD) 表

schema.CCD_table

この表の情報は、SQL を使って更新できます。

サーバー: ターゲット・サーバー

重要: SQL を使用してこの表を更新した場合は、アプライ・プログラムによってフル・リフレッシュが実行されたときにユーザーが行った更新が失われる危険性があります。

整合変更データ (CCD) 表は、ソースで発生した変更に関する情報と、これらの変更の順序を識別するための追加の列を含む、サブスクリプション・セット・メンバー内のターゲットです。列の値は、CD 表と UOW 表の結合により得られます。ターゲット・サーバーである CCD 表としては以下のものが考えられます。

- CD 表の代替として機能する内部 CCD 表。
変更キャプチャー・レプリケーションの場合、アプライ・プログラムはこの表からターゲットへ直接変更を適用します。このタイプの CCD 表の名前は、登録 (IBMSNAP_REGISTER) 表の中で、変更元のレプリケーション・ソースと同じ行に保管されます。
- 読み取り専用のターゲット表である外部 CCD。
このタイプの CCD は、ターゲット・サーバーのソース・データの監査証跡を保持します。
- multi-tier のレプリケーション構成の中間層である外部 CCD。
このタイプの CCD は、層 1 に対してはターゲット表であり、層 3 に対してはソース表です。このタイプの CCD 表の名前は、登録 (IBMSNAP_REGISTER) 表の中のもの自身に保管されます。

CCD 表およびターゲットの使用方法的詳細は、704 ページの『ターゲット・タイプの選択』を参照してください。

キャプチャー・プログラムは、CCD 表にデータを挿入せず、この表の整理も行いません。アプリケーションの要件に応じて、CCD 表の履歴の保存期間を決定してください。

整合変更データ表

い。したがって、デフォルトでは CCD 表のブルーニングは自動ではありませんが、サブスクリプション・サイクルの後に処理される SQL ステートメントを使えば簡単に自動化することができます。

外部 CCD の場合、UOW 表から以下のいくつかの列を含めるように選択できます。APPLY_QUAL、IBMSNAP_AUTHID、IBMSNAP_AUTHTKN、IBMSNAP_REJ_CODE、および IBMSNAP_UOWID。

IBMSNAP_OPERATION 列で最初に組み込まれた操作コードと、シーケンス番号 IBMSNAP_INTENTSEQ および IBMSNAP_COMMITSEQ は、CCD 表に含まれます。コンデンス CCD 表の場合、それぞれの列ごとに最新の値のみが保持されます。

表 174 では、CCD 表の列の要旨を示します。

表 174. CCD 表の列

列名	説明
IBMSNAP_INTENTSEQ	1 つの変更をユニークに識別するログまたはジャーナル・レコードのシーケンス番号この値はグローバルに昇順です。
IBMSNAP_OPERATION	レコードの操作のタイプを示すフラグ。 I 挿入 U 更新 D 削除
IBMSNAP_COMMITSEQ	取り込まれたコミット・ステートメントのログ・レコード・シーケンス番号。この値は、ソース表の元のトランザクションによる挿入、更新、および削除をグループ化します。
IBMSNAP_LOGMARKER	キャプチャー・コントロール・サーバーにおけるコミット時刻。
<i>user key columns</i>	CCD 表が圧縮されている場合、この列には、ターゲット・キーを構成する列が含まれます。
<i>user nonkey columns</i>	ソース表からの非キー・データ列。このターゲット表の中の列名はソース表の中の列名と一致している必要はありませんが、データ・タイプの互換性は必要です。
<i>user computed columns</i>	SQL 式から派生したユーザー定義の列。ソース・データ・タイプを別のターゲット・データ・タイプに変換するために、SQL 関数で算出列を使用することができます。
IBMSNAP_APPLY_QUAL (オプション)	この CCD 表を処理するアプライ・プログラムを一意的に識別します。

表 174. CCD 表の列 (続き)

列名	説明
IBMSNAP_AUTHID (オプション)	このトランザクションに関連する許可 ID。これはデータベースの監査に役立ちます。AUTHID の長さは 18 文字です。それを超えると切り捨てられます。DB2 Universal Database for z/OS の場合、この列は 1 次許可 ID です。DB2 Universal Database for iSeries の場合、この列は、トランザクションを発生させたアプリケーションを実行しているユーザー・プロファイル ID の名前になります。この列は、ブランクを埋め込んだ 10 文字の ID を保留します。この列は自動的に他の表にコピーされません。ユーザー・データ列として選択してコピーする必要があります。この列は、非コンプライト CCD ターゲット表のユーザー・データ列として選択できます。
IBMSNAP_AUTHTKN (オプション)	このトランザクションに関連する許可トークン。この ID は、データベースの監査に役立ちます。DB2 Universal Database for z/OS の場合、この列は関連 ID です。DB2 Universal Database for iSeries の場合、この列は、トランザクションを発生させたジョブのジョブ名です。この列は自動的に他の表にコピーされません。ユーザー・データ列として選択してコピーする必要があります。この列は、非コンプライト CCD ターゲット表のユーザー・データ列として選択できます。
IBMSNAP_REJ_CODE (オプション)	レプリケーション・ソースの定義時に競合検出が標準または詳細と指定されている場合、この値は update-anywhere レプリケーションでのみ設定されます。update-anywhere 構成に参加できないため、DB2 以外のリレーショナル・ターゲットの場合は使用できません。 <ul style="list-style-type: none"> 0 認識された競合のないトランザクション。 1 ソース表とレプリカ表の同じ行に、複製されなかった変更による競合があるトランザクション。競合が生じると、トランザクションはレプリカ表でロールバックされます。 2 最低でも 1 つの同一行競合がある、以前のトランザクションに從属するトランザクションのカスケード拒否。競合が生じると、トランザクションはレプリカ表でロールバックされます。 3 最低でも 1 つの参照保全制約違反があるトランザクション。このトランザクションはソース表で定義された参照制約に違反しているため、アプライ・プログラムはこのサブスクリプション・セットに失敗というマークを付けます。参照保全定義が訂正されるまで、更新はコピーできません。 4 最低でも 1 つの制約競合がある、以前のトランザクションに從属するトランザクションのカスケード拒否。
IBMSNAP_UOWID (オプション)	この作業単位についての、ログ・レコード・ヘッダーからの作業単位 ID。

関連資料:

- レプリケーションのガイドおよびリファレンス の『.CCD_table (DB2 以外)』

キャプチャー・プログラムがジャーナル・エントリー・タイプを処理する方法 (iSeries)

以下の表では、キャプチャー・プログラムが様々なジャーナル・エントリー・タイプを処理する方法について説明します。

表 175. ジャーナル・エントリーによるキャプチャー・プログラムの処理

ジャーナル ・コード ¹	エントリー ・タイプ	説明	処理
C	CM	コミット済みのレコード変更のセット	UOW 表にレコードを挿入します。
C	RB	ロールバック	挿入された UOW 行はありません。
F	AY	物理ファイル・メンバーに適用されたジャーナル済みの変更	ASN2004 メッセージを発行し、ファイルのフル・リフレッシュを行います。
F	CE	物理ファイルのデータの終わりの変更	ASN2004 メッセージを発行し、ファイルのフル・リフレッシュを行います。
F	CR	消去された物理ファイル・メンバー	ASN2004 メッセージを発行し、ファイルのフル・リフレッシュを行います。
F	EJ	終了した物理ファイル・メンバーのジャーナリング	ASN2004 メッセージを発行し、ファイルのフル・リフレッシュを行います。フル・リフレッシュは、ユーザーまたはシステムがジャーナリングを終了させたかにかかわらず、キャプチャー・プログラムが EJ ジャーナル・エントリーを読み取るたびに起こります。ファイルの暗黙的なジャーナル終了イベントについての情報は、OS/400 のドキュメンテーションを参照してください。

表 175. ジャーナル・エントリーによるキャプチャー・プログラムの処理 (続き)

ジャーナル ・コード ¹	エントリー ・タイプ	説明	処理
F	IZ	初期化された物理ファイル・メンバー	ASN2004 メッセージを発行し、ファイルのフル・リフレッシュを行います。
F	MD	物理ファイルから削除されたメンバー (DLTLIB、DLTF、または RMVM)	ASN2004 メッセージを発行し、フル・リフレッシュを試行します。
F	MF	解放された物理ファイル・メンバーのストレージ	ASN2004 メッセージを発行し、ファイルのフル・リフレッシュを行います。
F	MM	移動されたメンバーを含む物理ファイル (ライブラリーのオブジェクトの名前変更 (RNMOBJ)、ファイルのオブジェクトの移動 (MOVOBJ))	ASN200A メッセージを発行し、フル・リフレッシュを試行します。
F	MN	名前変更されたメンバー (ファイルの RNMOBJ、メンバーの名前変更 (RNMM)) を含む物理ファイル	ASN200A メッセージを発行し、フル・リフレッシュを試行します。
F	MR	リストアされた物理ファイル・メンバー	ASN2004 メッセージを発行し、ファイルのフル・リフレッシュを行います。
F	RC	物理ファイル・メンバーから削除されたジャーナル済みの変更	ASN2004 メッセージを発行し、ファイルのフル・リフレッシュを行います。
F	RG	再編成された物理ファイル・メンバー	レプリケーション・キーとしてソース表の RRN が使用されている場合には、ASN2004 メッセージを発行して、ファイルのフル・リフレッシュを行います。
J	NR	次のジャーナル・レシーバーの ID	キャプチャー・プログラムをリセットします。
J	PR	直前のジャーナル・レシーバーの ID	ユニークのシーケンス番号カウンターを増分します。

表 175. ジャーナル・エントリーによるキャプチャー・プログラムの処理 (続き)

ジャーナル ・コード ¹	エントリー ・タイプ	説明	処理
R	DL	物理ファイル・メンバーから削除されたレコード	DLT レコードを CD 表に挿入します。
R	DR	ロールバックで削除されたレコード	DLT レコードを CD 表に挿入します。
R	PT	物理ファイル・メンバーに追加されたレコード	ADD レコードを CD 表に挿入します。
R	PX	物理ファイル・メンバーに直接追加されたレコード	ADD レコードを CD 表に挿入します。
R	UB	物理ファイル・メンバーで更新されたレコードの変更前イメージ	注 2 を参照してください。
R	UP	物理ファイル・メンバーで更新されたレコードの変更後イメージ	注 2 を参照してください。
R	BR	ロールバックで更新されたレコードの変更前イメージ	注 3 を参照してください。
R	UR	ロールバックで更新されたレコードの変更後イメージ	注 3 を参照してください。

注:

- 以下の値が、ジャーナル・コードに使用されます。
 - C** コミットメント・コントロール操作
 - F** データベース・ファイル操作
 - J** ジャーナルまたはジャーナル・レシーバー操作
 - R** 特定のレコードに対する操作
- 登録表の PARTITION_KEYS_CHG 列が N の場合、R-UP イメージおよび R-UB イメージが単一の UPD レコードを CD 表に形成します。そうでない場合には、R-UB イメージが DLT レコードを CD 表に挿入し、R-UP イメージが ADD 表を CD 表に挿入します。
- 登録表の PARTITION_KEYS_CHG 列が N の場合、R-UR イメージおよび R-BR イメージが単一の UPD レコードを CD 表に形成します。そうでない場合には、R-BR イメージが DLT レコードを CD 表に挿入し、R-UR イメージが ADD 表を CD 表に挿入します。

他のジャーナル・エントリー・タイプはすべて、キャプチャー・プログラムによって無視されます。

イベント・タイプ

イベント・モニターは、CREATE EVENT MONITOR ステートメントで指定されたイベント・タイプに関する情報を戻します。それぞれのイベント・タイプごとに、特定の時点でモニター情報が収集されます。以下の表は、使用可能なイベント・タイプ、モニター・データが収集される時点、およびそれぞれのイベント・タイプについて入手できる情報をリストしています。最初の列にある使用可能なイベント・タイプは、CREATE EVENT MONITOR ステートメントで使用されるキーワードに対応しています。ここでイベント・タイプが定義されます。

データが発生するように定義されたイベントに加えて、FLUSH EVENT MONITOR SQL ステートメントを使用してイベントを生成することもできます。この方式によって生成されたイベントは、フラッシュされたイベント・モニターに関連したすべてのモニター・タイプ (DEADLOCKS および DEADLOCKS WITH DETAILS を除く) に関する現行のデータベース・モニター値とともに書き込まれます。

表 176. イベント・タイプ

イベント・タイプ	データが収集される時点	入手できる情報
DEADLOCKS	デッドロック検出時	関係しているアプリケーション、および競合しているロック。
DEADLOCKS WITH DETAILS	デッドロック検出時	関係するアプリケーションについての広範囲の情報。関係するステートメント (およびステートメント・テキスト) の識別や保持されているロックなど。DEADLOCKS イベント・モニターではなく DEADLOCKS WITH DETAILS イベント・モニターを使用すると追加の情報が収集されるため、デッドロックが発生したときにパフォーマンス・コストの負担になります。
STATEMENTS	SQL ステートメント終了時	ステートメントの開始 / 停止時刻、使用されている CPU、動的 SQL のテキスト、SQLCA (SQL ステートメントの戻りコード)、 およびその他のメトリック (取り出しカウントなど)。
	サブセクション終了時	パーティション・データベースの場合、使用されている CPU、実行時間、表、および表キューに関する情報。
TRANSACTIONS	作業単位の終了時	作業単位の作業開始 / 停止時刻、直前の UOW 時刻、使用されている CPU、ロック、およびログ記録のメトリック。XA で実行している場合は、トランザクション・レコードは生成されない。

表 176. イベント・タイプ (続き)

イベント・タイプ	データが収集される時点	入手できる情報
CONNECTIONS	接続終了時	すべてのアプリケーション・レベルのカウンター。
DATABASE	データベース非活動化時	すべてのデータベース・レベルのカウンター。
BUFFERPOOLS	データベース非活動化時	バッファ・プールのカウンター、プリフェッチ機能、ページ・クリーナー、および個々のバッファ・プールの直接 I/O。
TABLESPACES	データベース非活動化時	バッファ・プールのカウンター、プリフェッチ機能、ページ・クリーナー、および個々の表スペースの直接 I/O。
TABLES	データベース非活動化時	個々の表の、読み取り / 書き込みが行われる行。

注: 「デッドロックの詳細」 イベント・モニターが、新規に作成されるデータベースごとに作成されます。このイベント・モニターは DB2DETAILDEADLOCK という名前前で、データベースが活動化されると開始され、データベース・ディレクトリー内のファイルに書き込みます。このイベント・モニターによるオーバーヘッドは、これをドロップすることによって回避することができます。

関連概念:

- SQL リファレンス 第 1 巻 の『イベント・モニター』
- システム・モニター ガイドおよびリファレンス の『カウンターの状況および可視性』
- システム・モニター ガイドおよびリファレンス の『イベント・タイプの論理データ・グループへのマッピング』

関連タスク:

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

関連資料:

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

すべてのデータベース・ユーザーのためのシステム・モニター・データへのアクセス: DB2_SNAPSHOT_NOAUTH レジストリー変数

レジストリー変数 DB2_SNAPSHOT_NOAUTH は、スナップショット・モニター API、CLP コマンド、または SQL 表機能を使用して、すべてのユーザーがデータベース・システム・モニター・データにアクセスできるようにします。このレジストリー変数がいずれかの値に設定されていると、すべてのユーザーがスナップショット・モニターを使用して、システム・モニター・データにアクセスすることができます。このレジ

ストリー変数が設定されていない場合は、システム管理またはシステム管理権限を持つユーザーのみが、スナップショット・モニターを使用してシステム・モニター・データにアクセスすることができます。

システム管理またはシステム管理権限を持たないユーザーには、以下の制限があります。

- システム・モニター・スイッチの状態を変更できません。
- RESET MONITOR コマンドを実行できません。

DB2_SNAPSHOT_NOAUTH レジストリー変数をオンにして、すべてのユーザーがシステム・モニター・データにアクセスできるようにすると、以下のセキュリティー上のリスクが発生します。

- スナップショット・モニター API、CLP コマンド、または SQL 表機能へのおびただしいアクセスは、高い接続 / 切断率で、ワークロードに深刻な性能低下を引き起こします。(特に集線装置環境において) したがって、ユーザーが過度のラッチ競合を引き起こすことによって、故意にデータベース・システムの速度を低下させることがあります。
- 他の接続ユーザーのリストおよび、データベースに対してそれらのユーザーがサブミットした SQL ステートメントなどの機密データが、すべてのユーザーから使用可能になります。注: データベースからの実際のデータは、スナップショット・モニター API、CLP コマンド、または SQL 表機能を使用して漏れることはありません。

DB2_SNAPSHOT_NOAUTH レジストリー変数は、システム管理またはシステム管理権限を持たないユーザーが、データベース・システム・モニター・データにアクセスできるようにする暫定的な手段です。将来のリリースで、このレジストリー変数は **GRANT/REVOKE** ステートメントで管理できるデータベース・マネージャー・レベルの権限で置き換えられます。

アプリケーション開発

アプリケーション開発の手引き

Java 環境のセットアップ

DB2 データベースにアクセスする Java プログラムは、ご使用のプラットフォームに対応した Java Development Kit (JDK) を使用して開発できます。JDK には、Java 用の動的 SQL API である JDBC (Java Database Connectivity) が含まれています。

DB2 JDBC サポートは、DB2 クライアントおよびサーバー上の Java Enablement オプションの一部として提供されます。このサポートによって、JDBC アプリケーションとアプレットを構築し、実行できます。これらには動的 SQL だけが含まれ、Java 呼び出しインターフェースを使用して SQL ステートメントを DB2 に渡します。

DB2 Java 組み込み SQL (SQLj) サポートは、DB2 AD クライアントの一部として提供されます。DB2 JDBC サポートとともに DB2 SQLj サポートを利用することで、SQLj アプレットおよびアプリケーションの作成と実行が可能になります。これらには、静的 SQL が含まれ、DB2 データベースにバインドされた組み込み SQL ステートメントを使用します。

DB2 AD クライアントによって提供される SQLj サポートには、以下のものが含まれません。

- DB2 SQLj 変換プログラム、**sqlj**。SQLj プログラムにある組み込み SQL ステートメントを Java ソース・ステートメントで置換し、SQLj プログラムで見つかった SQL 操作についての情報を含むシリアルライズされたプロファイルを生成します。
- DB2 SQLj プロファイル・カスタマイザー **db2sqljcustomize**。これはシリアルライズされたプロファイルに保管された SQL ステートメントをプリコンパイルし、それらを実行時関数呼び出しにカスタマイズし、そして DB2 データベース内にパッケージを生成します。
- DB2 SQLj プロファイル・プリンター **db2sqljprint**。これはカスタマイズしたバージョンの DB2 プロファイルの内容をプレーン・テキスト形式で印刷します。
- DB2 SQLj プロファイル・バインダー **db2sqljbind**。既にカスタマイズされた SQLj プログラムからパッケージを生成します。

手順:

DB2 Java アプリケーションを実行するには、ネイティブ・スレッド・サポートを提供する Java 仮想マシン (JVM) をインストールして呼び出さなければなりません。ネイティブ・スレッドを使用して Java アプリケーションを実行するには、コマンド内の

-native オプションを使用することができます。たとえば、Java サンプル・アプリケーション DbInfo.class を実行するには、以下のコマンドを使うことができます。

```
java -native DbInfo
```

THREADS_FLAG 環境変数を native に設定すれば、Java 仮想マシンによっては、デフォルトのスレッド・サポートとしてネイティブ・スレッドを指定することができます。本書では、ネイティブ・スレッド・サポートがデフォルトであることを前提としています。ご使用のシステムでネイティブ・スレッドをデフォルトにする方法については、JVM の資料を参照してください。

DB2 Java アプレットを実行するには、ネイティブ・スレッド・サポートまたはグリーン・スレッド・サポートのいずれかを提供する Java 仮想マシンを呼び出すことができます。

上記がすべてインストール済みで稼働状態になったら、以下に示すいずれかのステップを行って、各自のオペレーティング・システムの Java 環境をセットアップすることができます。

- UNIX Java 環境のセットアップ
- Windows Java 環境のセットアップ

DB2 Java アプリケーション開発の最新の更新事項については、以下の Web ページを参照してください。

<http://www.ibm.com/software/data/db2/java>

関連タスク:

- 750 ページの『UNIX Java 環境のセットアップ』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『Windows Java 環境のセットアップ』

関連資料:

- 1298 ページの『db2sqljcustomize - DB2 SQLj プロファイル・カスタマイザー』
- 1304 ページの『db2sqljprint - DB2 SQLj プロファイル・プリンター』
- 1312 ページの『db2sqljbind - DB2 SQLj プロファイル・バインダー』

マルチスレッド・データベース・アクセスの目的

いくつかのオペレーティング・システムに共通する特徴は、1 つのプロセスで実行プログラムの複数のスレッドを実行できることです。複数のスレッドにより、アプリケーションが非同期のイベントを処理することができ、ポーリング機能がなくても容易にイベント・ドリブン・アプリケーションを作成できます。以下の情報では、データベース・マネージャーが複数のスレッドを処理する方法を解説し、留意すべき設計の指針を示します。

マルチスレッドのアプリケーション開発に関する用語 (クリティカル・セクションおよびセマフォなど) について詳しくない方は、ご使用のオペレーティング・システムのプログラミングに関する資料をご覧ください。

DB2 アプリケーションは、コンテキスト を使用して複数のスレッドから SQL ステートメントを実行することができます。コンテキストとは、アプリケーションがすべての SQL ステートメントおよび API 呼び出しを実行する環境のことです。すべての接続、作業単位、および他のデータベース・リソースは、特定のコンテキストに関連付けられています。各コンテキストは、アプリケーション内の 1 つ以上のスレッドに関連付けられています。

各実行可能 SQL ステートメントでは、最初のランタイム・サービス呼び出しは常にラッチを取得しようとします。成功すると処理を続行しますが、(他のスレッドの SQL ステートメントがすでにラッチを取得しているために) 失敗すると、呼び出しは信号セマフォでこれがポストされるまでブロックされ、それからラッチを取得し処理を続行します。ラッチは SQL ステートメントが処理を終了するまで保持され、その SQL ステートメントに対して生成された最後のランタイム・サービス呼び出しにより解放されます。

最終的な結果として、他のスレッドが SQL ステートメントを同時に実行しようとしても各 SQL ステートメントはアトミック単位で実行されます。これにより内部データ構造は、異なるスレッドによって同時に変更されることがなくなります。API もランタイム・サービスを使用したラッチを使用します。したがって、API には、各コンテキスト内のランタイム・サービス・ルーチンと同じ制限が課されます。

DB2[®] バージョン 8 では、すべてのバージョン 8 アプリケーションは、デフォルトでマルチスレッドになり、複数のコンテキストを使用できます。(バージョン 8 以前のアプリケーションの動作は、変更ありません。) 必要な場合は、以下の DB2 API を使用して、複数コンテキストを使用できます。具体的に言えば、アプリケーションはスレッド用のコンテキストを作成でき、各スレッド用の別個のコンテキストにアタッチ、またはデタッチができ、スレッド間でコンテキストを渡すことができます。アプリケーションがこれらの API のいずれも呼び出さない場合は、DB2 が自動的にアプリケーション用に複数コンテキストを管理します。

- `sqlcBeginCtx()`
- `sqlcEndCtx()`
- `sqlcAttachToCtx()`
- `sqlcDetachFromCtx()`
- `sqlcGetCurrentCtx()`
- `sqlcInterruptCtx()`

コンテキストはプロセス内のスレッド間で交換できますが、プロセス間では交換できません。複数のコンテキストの使用法の 1 つは、並行トランザクションのサポートです。

関連概念:

- アプリケーション開発ガイド クライアント・アプリケーションのプログラミングの『並行トランザクション』

関連資料:

- 管理 API リファレンスの『`sqlAttachToCtx` - コンテキストへの接続』
- 管理 API リファレンスの『`sqlBeginCtx` - アプリケーション・コンテキストの作成および接続』
- 管理 API リファレンスの『`sqlDetachFromCtx` - コンテキストからの切り離し』
- 管理 API リファレンスの『`sqlEndCtx` - アプリケーション・コンテキストの切り離しおよび破棄』
- 管理 API リファレンスの『`sqlGetCurrentCtx` - 現行コンテキストの入手』
- 管理 API リファレンスの『`sqlInterruptCtx` - コンテキストへの割り込み』

関連サンプル:

- 『`dbthrds.sqc -- How to use multiple context APIs on UNIX (C)`』
- 『`dbthrds.sqc -- How to use multiple context APIs on UNIX (C++)`』

Java のソースおよび出力ファイル

ソース・ファイルは、以下のような拡張子を持ちます。

- **.java** プリコンパイルの必要なしの Java™ ソース・ファイル。これらのファイルは、Java 開発環境に組み込まれている `javac` Java コンパイラーを使用してコンパイルできます。
- **.sqlj** `sqlj` 変換プログラムでの変換が必要な SQLj ソース・ファイル。変換プログラムは、以下のものを作成します。
 - 1 つ以上の `.class` バイトコード・ファイル
 - 接続コンテキストごとに 1 つの `.ser` プロファイル・ファイル

対応する出力ファイルは、以下のような拡張子を持ちます。

- **.class** JDBC および SQLj バイトコード・コンパイル済みファイル。
- **.ser** SQLj 順次化プロファイル・ファイル。それぞれのプロファイル・ファイルについて、`db2sqljcustomize` および `db2sqljbind` ユーティリティを使用してデータベースにパッケージを作成します。

SQLj 用の DB2 サポート

DB2® SQLj サポートは、DB2 Application Development Client で提供されます。DB2 クライアントが提供する JDBC サポートとともに DB2 SQLj サポートを使用すると、Java™ 用組み込み SQL アプリケーション、アプレット、ストアード・プロシージャ

一、およびユーザー定義関数 (UDF) を作成、構築、および実行できるようにします。これらには、静的 SQL が含まれ、DB2 データベースにバインドされた組み込み SQL ステートメントを使用します。

DB2 Application Development Client が提供する SQLj サポートには、以下のものが含まれます。

- SQLj 変換プログラム、**sqlj**。SQLj プログラムにある組み込み SQL ステートメントを Java ソース・ステートメントで置換し、SQLj プログラムで見つかった SQL 操作についての情報を含むシリアルライズされたプロファイルを生成します。SQLj 変換プログラムは、`sqllib/java/sqlj.zip` ファイルを使用します。
- SQLj 実行時クラス。`sqllib/java/db2jcc.jar` にある。
- DB2 SQLj プロファイル・カスタマイザー `db2sqljcustomize`。これは生成されたプロファイルに保管された SQL ステートメントをプリコンパイルし、DB2 データベース内にパッケージを生成します。
- DB2 SQLj プロファイル・プリンター `db2sqljprint`。カスタマイズ済みの DB2 プロファイルの内容をプレーン・テキストで印刷します。
- DB2 SQLj プロファイル・バインダー `db2sqljbind`。既にカスタマイズされた SQLj プログラムからパッケージを生成します。
- SQLj プロファイル監査プログラム・インストーラー `profdb`。デバッグ・クラス監査プログラムを バイナリー・プロファイルの既存のセットにインストール (またはアンインストール) します。いったんインストールされると、アプリケーションの実行時に行われる `RTStatement` と `ResultSet` の呼び出しはすべて、ファイル (または標準出力) のログに記録されます。その後、このログを調べて、期待した動作をしているかどうか検証したり、エラーのトレースを行うことができます。実行時に基礎となる `RTStatement` および `ResultSet` 呼び出しインターフェースに対して行われた呼び出しだけが監査される点に注意してください。
- SQLj プロファイル変換ツール `profconv`。シリアルライズされたプロファイル・インスタンスをクラス・バイトコード形式に変換する。一部のブラウザーは、アプレットに関連付けられたリソース・ファイルからのシリアルライズ・オブジェクトのロードをサポートしていません。対処方法として、このユーティリティを実行して変換を実行する必要があります。

SQLj ランタイム・クラスの詳細については、DB2 Java Web サイトを参照してください。

関連概念:

- 746 ページの『SQLj に関する DB2 の制約事項』

関連資料:

- 1298 ページの『`db2sqljcustomize` - DB2 SQLj プロファイル・カスタマイザー』
- 1304 ページの『`db2sqljprint` - DB2 SQLj プロファイル・プリンター』

- 1312 ページの『db2sqljbind - DB2 SQLj プロファイル・バインダー』

SQLj に関する DB2 の制約事項

SQLj を使用して DB2 アプリケーションを作成する際には、以下の制約事項に注意してください。

- DB2® SQLj サポートは、SQL ステートメントの発行に関する標準 DB2 Universal Database の制約事項に準拠します。
- 位置指定された UPDATE および DELETE ステートメントは、コンバウンド SQL ステートメントでは有効なサブステートメントではありません。
- プリコンパイル・オプション "DATETIME" はサポートされていません。国際標準化機構規格の日付および時刻形式しかサポートされていません。
- **db2sqljcustomize** コマンドに `-rootpkgname` オプションを指定しないと、パッケージ名は以下のように決定されます。
 1. ルート・パッケージ名はプログラム名にプロファイル・ファイルの索引を付加し、その結果として得られた名前を英大文字に変換して生成されます。ルート・パッケージ名は合計で 7 文字に制限されているため、ルート・パッケージ名のプログラム名部分が切り捨てられる場合があります。
 2. 4 つのパッケージの名前は、ルート・パッケージ名に接尾部 1、2、3、および 4 を追加して生成されます。

たとえば、プログラム名が `SampleSqlj_SJProfile0.ser` で、ルート・パッケージ名が `SAMPLE0` の場合、パッケージは `SAMPLE01`、`SAMPLE02`、`SAMPLE03`、および `SAMPLE04` になります。

パッケージ名が `A_SJProfile10.ser` で、ルート・パッケージ名が `A10` の場合、パッケージは `A101`、`A102`、`A103`、および `A104` になります。

- `java.math.BigDecimal` ホスト変数が使用される場合、そのホスト変数の精度およびスケールはアプリケーションの変換中は使用できません。10 進数ホスト変数の精度およびスケールが、これを使用しているステートメントの内容からはっきり判別できない場合には、`CAST` を使用して精度およびスケールを指定することができます。
- タイプ `java.math.BigInteger` の Java™ 変数は、SQL ステートメントではホスト変数として使用できません。

一部のブラウザは、アプレットに関連付けられたリソース・ファイルからのシリアライズ・オブジェクトのロードをサポートしていません。そのようなブラウザでアプレット `Applt` をロードしようとする、以下のようなエラー・メッセージが出されます。

```
java.lang.ClassNotFoundException: Applt_SJProfile0
```

これを回避する方法としては、シリアライズ・プロファイルを Java クラス形式で保管するプロファイルに変換するユーティリティを使用することができます。このユーテ

イリティーは、`sqlj.runtime.profile.util.SerProfileToClass` という名前の Java クラスです。これはシリアルライズ・プロファイルのリソース・ファイルを入力として取り込み、そのプロファイルを含んだ Java クラスを出力として生成します。プロファイルを変換するには、以下のコマンドを使用します。

```
profconv Applt_SJProfile0.ser
```

または

```
java sqlj.runtime.profile.util.SerProfileToClass Applt_SJProfile0.ser
```

このコマンドの結果、クラス `Applt_SJProfile0.class` が作成されます。アプレットが使用している `.ser` 形式のすべてのプロファイルを `.class` 形式のプロファイルに置き換えます。

SQLj アプレットには、`db2java.zip` および `runtime.zip` ファイルの両方が必要です。すべてのアプレット・クラス (`db2java.zip` および `runtime.zip` 中のクラス) を、単一の Jar ファイルにパッケージしないように選択した場合には、`db2java.zip` と `runtime.zip` (コンマで区切る) の両方を "applet" タグのアーカイブ・パラメーターに入れます。アーカイブ・タグで複数の zip ファイルをサポートしないブラウザの場合は、そのタグで `db2java.zip` を指定し、ご使用の Web ブラウザーにアクセスできる作業ディレクトリで、現在のアプレット・クラスを指定して `runtime.zip` を `unzip` します。

SQLj プログラムのコンパイルと実行の例

MyClass という SQLj プログラムがあるとします。このプログラムを実行するには、以下のようにします。

1. SQLj (Java 用組み込み SQL) ソース・ファイル `MyClass.sqlj` を SQLj 変換プログラムにより変換して、Java™ ソース・ファイル `MyClass.java` を生成します。変換プログラムは、プロファイル `MyClass_SJProfile0.ser`、`MyClass_SJProfile1.ser`、... (それぞれの接続コンテキストごとに 1 つのプロファイル) も生成します。

```
sqlj MyClass.sqlj
```

`sqlj.properties` ファイルを指定せずに `sqlj` 変換プログラムを使用すると、変換プログラムは以下の値を使用します。

```
sqlj.url=jdbc:db2:sample
sqlj.driver=COM.ibm.db2.jdbc.app.DB2Driver
sqlj.online=sqlj.semantics.JdbcChecker
sqlj.offline=sqlj.semantics.OfflineChecker
```

`sqlj.properties` ファイルを指定する場合は、以下のオプションが設定されていることを確認してください。


```
sqlj.url=jdbc:db2:dbname
sqlj.driver=COM.ibm.db2.jdbc.app.DB2Driver
sqlj.online=sqlj.semantics.JdbcChecker
sqlj.offline=sqlj.semantics.OfflineChecker
```

ここで *dbname* は、データベースの名前です。また、コマンド行でこれらのオプションを指定することもできます。たとえば、MyClass の変換時にデータベース *mydata* を指定するには、以下のコマンドを実行できます。

```
sqlj -url=jdbc:db2:mydata MyClass.sqlj
```

-compile=false 文節でコンパイル・オプションの設定を明示的にオフにしないかぎり、SQLj 変換プログラムは変換されたソース・コードをクラス・ファイルに自動的にコンパイルします。

2. 生成されたプロファイルに DB2 SQLj カスタマイザーをインストールし、DB2 データベース *dbname* に DB2[®] パッケージを作成します。

```
db2sqljcustomize -user user-name -password user-password
                 -url jdbc:db2://server:port/dbname MyClass_SJProfile0.ser
db2sqljcustomize -user user-name -password user-password
                 -url jdbc:db2://server:port/dbname MyClass_SJProfile1.ser
...
```

3. SQLj プログラムを実行します。

```
java MyClass
```

変換プログラムは、SQLj プロファイルがカスタマイズされるデータベースの SQL 構文を生成します。たとえば、以下のようになります。

```
i = { VALUES ( F(:x) ) };
```

が SQLj 変換プログラムによって変換され、

```
? = VALUES ( F (?) )
```

として、生成されるプロファイルに保管されます。DB2 Universal Database データベースに接続すると、DB2 は、VALUE ステートメントを以下のようにカスタマイズします。

```
VALUES(F(?)) INTO ?
```

一方、DB2 Universal Database for OS/390 and z/OS データベースに接続すると、DB2 は VALUE ステートメントを以下のようにカスタマイズします。

```
SELECT F(?) INTO ? FROM SYSIBM.SYSDUMMY1
```

関連タスク:

- 778 ページの『SQLj アプレットの作成』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『SQLj アプリケーションの作成』

- アプリケーション開発ガイド アプリケーションの構築および実行 の『SQLJ ルーチンの作成』
- 787 ページの『SQLj プログラムの作成』

SQLj カスタマイザー・オプション

SQLj カスタマイザーは、いくつかの例外はありますが、DB2® の BIND コマンドと同じバインド・オプションのいくつかをサポートします。サポートされているバインド・オプションについては、db2sqljcustomize コマンドを参照してください。

SQLj 変換プログラムによって生成されたプロファイルの内容を、プレーン・テキストで印刷するには、以下のように profp ユーティリティを使用します。

```
profp MyClass_SJProfile0.ser
profp MyClass_SJProfile1.ser
...
```

DB2 のカスタマイズされたプロファイルの内容をプレーン・テキストで印刷するには、以下のように db2sqljprint ユーティリティを使用します。

```
db2sqljprint MyClass_SJProfile0.ser
db2sqljprint MyClass_SJProfile1.ser
...
```

関連資料:

- 1298 ページの『db2sqljcustomize - DB2 SQLj プロファイル・カスタマイザー』

SQLj ルーチン内の接続コンテキスト

DB2® Universal Database バージョン 8 でマルチスレッド・ルーチンが採り入れられたため、SQLj ルーチンでデフォルトの接続コンテキストの使用を避けることが重要です。つまり、各 SQL ステートメントは ConnectionContext オブジェクトを明示的に指定し、コンテキストは Java™ 方式で明示的にインスタンス化する必要があります。たとえば、前のリリースの DB2 では、SQLj ルーチンを以下のように書くことができました。

```
class myClass
{
    public static void myRoutine( short myInput )
    {
        DefaultContext ctx = DefaultContext.getDefaultContext();
        #sql { some SQL statement };
    }
}
```

このデフォルト・コンテキストの使用は、マルチスレッド環境のすべてのスレッドが同じ接続コンテキストを使用することになり、予期しない障害が発生します。

上記の SQLj ルーチンを、以下のように変更する必要があります。

```
|  
|  
| #context MyContext;  
|  
|  
| class myClass  
| {  
|     public static void myRoutine( short myInput )  
|     {  
|         MyContext ctx = new MyContext( "jdbc:default:connection", false );  
|         #sql [ctx] { some SQL statement };  
|         ctx.close();  
|     }  
| }  
|
```

|
| このように、ルーチンのそれぞれの呼び出しで、並行スレッドによる予期しない干渉を
| 避ける、独自でユニークの ConnectionContext (および基礎となる JDBC 接続) を作成し
| ます。
|

UNIX Java 環境のセットアップ

DB2 JDBC サポートを利用して UNIX 上で JDBC プログラムや SQLj プログラムを実行するために、Java 環境を更新するコマンドが、データベース・マネージャー・ファイル db2profile と db2cshrc に組み込まれています。DB2 インスタンスの作成時に、.bashrc、.profile、および .cshrc、あるいは、いずれかが以下のように変更されます。

1. THREADS_FLAG を "native" に設定します。(HP-UX、Linux、および Solaris のみ)
2. CLASSPATH に以下のものを含めます。
 - "." (現行ディレクトリー)
 - ファイル sqllib/java/db2java.zip
 - ファイル sqllib/java/db2jcc.jar

SQLj プログラムを作成するには、以下のファイルを組み込むように CLASSPATH を更新します。

```
sqllib/java/sqlj.zip
```

|
| **注:** 将来の DB2 のバージョンでは、このディレクトリーはなくなるため、
| sqllib/java12 ディレクトリーへの直接の参照は、すべて除去する必要があります。
| す。代わりに、sqllib/java ディレクトリーを参照してください。
|

手順:

DB2 Java ルーチン (ストアード・プロシージャと UDF) を実行するには、そのマシンの JDK のインストール先のパスを組み込むように、サーバー上の DB2 データベース・マネージャー構成を更新する必要があります。そのためは、サーバーのコマンド行で以下のように入力します。

```
db2 update dbm cfg using JDK_PATH /home/db2inst/jdk13
```

`/home/db2inst/jdk13` は、JDK がインストールされているパスです。

以下のコマンドをサーバーで入力して、DB2 データベース・マネージャー構成をチェックし、`JDK_PATH` フィールドの値が正しいことを確認できます。

```
db2 get dbm cfg
```

出力をファイルにリダイレクトすれば、一層容易に表示できます。`JDK_PATH` フィールドは、出力の先頭近くに表示されます。

上記のものがインストール済みで稼働状態になったら、以下に示すいずれかのステップを行って、各自の UNIX オペレーティング・システム環境をセットアップすることができます。

- AIX Java 環境のセットアップ
- HP-UX Java 環境のセットアップ
- Linux Java 環境のセットアップ
- Solaris Java 環境のセットアップ

関連タスク:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『AIX Java 環境のセットアップ』
- 751 ページの『HP-UX Java 環境のセットアップ』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『Linux Java 環境のセットアップ』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『Solaris Java 環境のセットアップ』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『データベース・マネージャー構成ファイルの更新』

関連資料:

- コマンド・リファレンス の『GET DATABASE MANAGER CONFIGURATION コマンド』
- コマンド・リファレンス の『RESET DATABASE MANAGER CONFIGURATION コマンド』

HP-UX Java 環境のセットアップ

手順:

HP-UX 上で DB2 JDBC サポートを利用して Java アプリケーションを作成するには、開発マシンで以下をインストールして構成する必要があります。

1. HP-UX 32 ビット: Software Developer's Kit および Runtime Environment 1.3.1 for HP-UX 11.0 および Hewlett-Packard の 11i PA-RISC。

HP-UX 64 ビット: Software Developer's Kit および Runtime Environment 1.4 for HP-UX 11.0 および Hewlett-Packard の 11i PA-RISC。

2. DB2 Java Enablement。これは HP-UX クライアントおよびサーバーに対応した DB2 Universal Database バージョン 8 に装備されています。

HP-UX での Java ルーチン (ストアード・プロシージャとユーザー定義関数) では、最小 JAVA_HEAP_SZ は 1024 です。

HP-UX 32 ビットで Java ルーチンを実行するには、共有ライブラリー・パスを必ず以下のように入力してください。

```
export SHLIB_PATH=$JAVADIR/jre/lib/PA_RISC:¥
                $JAVADIR/jre/lib/PA_RISC/classic:¥
                $HOME/sql11ib/lib:¥
                /usr/lib:$SHLIB_PATH
```

\$JAVADIR は、通常、/opt/java1.3 (Java SDK for HP-UX 32 ビットのデフォルト・ロケーション) に設定されます。

HP-UX 64 ビットで Java ルーチンを実行するには、以下のコマンドをコマンド行で実行して、db2hplv ツールを使用可能にします。

```
db2hplv -e
```

次のコマンドは、このサポートを使用不可にします。

```
db2hplv -d
```

変更を有効にするために、db2hplv -e または db2hplv -d の実行後に、db2stop および db2start を実行する必要があります。Java ルーチン・サポートは、デフォルトで使用不可になっています。

注: DB2 for HP-UX は、Java ルーチン・サポートが db2hplv -e で使用可能にされている場合は実行されませんが、Java はシステムでアンインストールされます。

HP-UX 64 ビットでは、以下のライブラリーへのシンボリック・リンクは、/usr/lib/pa20_64 に作成する必要があります。そうでない場合は、SQL4301N エラーとなる場合があります。

```
/opt/java1.4/jre/lib/PA_RISC2.0W/libnet.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libzip.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/librmi.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libnio.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libverify.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libmlib_image.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libhprof.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libjaas_unix.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libawt.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libcmm.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libdcpr.sl
```

```
| /opt/java1.4/jre/lib/PA_RISC2.0W/libdt_socket.sl  
| /opt/java1.4/jre/lib/PA_RISC2.0W/libfontmanager.sl  
| /opt/java1.4/jre/lib/PA_RISC2.0W/libioser12.sl  
| /opt/java1.4/jre/lib/PA_RISC2.0W/libmawt.sl  
| /opt/java1.4/jre/lib/PA_RISC2.0W/libjsound.sl  
| /opt/java1.4/jre/lib/PA_RISC2.0W/libjava.sl  
| /opt/java1.4/jre/lib/PA_RISC2.0W/libjawt.sl  
| /opt/java1.4/jre/lib/PA_RISC2.0W/libjcov.sl  
| /opt/java1.4/jre/lib/PA_RISC2.0W/libjcpm.sl  
| /opt/java1.4/jre/lib/PA_RISC2.0W/libjdwps.sl  
| /opt/java1.4/jre/lib/PA_RISC2.0W/libjpeg.sl  
| /opt/java1.4/jre/lib/PA_RISC2.0W/hotspot/libjsig.sl  
| /opt/java1.4/jre/lib/PA_RISC2.0W/hotspot/libjvm.sl
```

リンクが存在しない場合は、以下のコマンドで作成できます。(実行にはルート権限が必要です)

```
| ln -s /opt/java1.4/jre/lib/PA_RISC2.0W/*.sl /usr/lib/pa20_64  
| ln -s /opt/java1.4/jre/lib/PA_RISC2.0W/hotspot/*.sl /usr/lib/pa20_64
```

関連概念:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『Java サンプル・プログラム』
- 772 ページの『Java アプレットに関する考慮事項』

関連タスク:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『サンプル・データベースのセットアップ』

関連資料:

- 761 ページの『アプリケーションの作成と実行用として HP-UX でサポートされるソフトウェア』

Windows Java 環境設定値

サポートされる Windows® プラットフォーム上で JDBC プログラムや SQLj プログラムを DB2® JDBC サポートを利用して実行するために、CLASSPATH は以下のものを組み込むように DB2 のインストール時に自動的に更新されます。

- "." (現行ディレクトリー)
- ファイル sqllib¥java¥db2java.zip
- ファイル sqllib¥java¥db2jcc.jar

SQLj プログラムを作成するには、以下のファイルを組み込むように CLASSPATH を更新します。

```
sqllib¥java¥sqlj.zip
```

注: Microsoft® Software Developer's Kit for Java™ は DB2 バージョン 8 ではサポートされません。これは、SQLj のカスタマイズでもタイプ 2 の JDBC アプリケーションでも使用できません。

関連タスク:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『Windows Java 環境のセットアップ』

SQL プロシージャのプリコンパイル・オプションとバインド・オプションのカスタマイズ

プリコンパイルおよびバインド・オプションは、DB2_SQLROUTINE_PREPOPTS DB2 レジストリー変数を設定することによってカスタマイズできます。これらのオプションを、プロシージャ・レベルでカスタマイズすることはできません。

手順:

SQL プロシージャに、カスタマイズしたプリコンパイル・オプションを指定するには、以下のコマンドを使用して、DB2 プリコンパイラーで使用するプリコンパイル・オプションのリストを DB2 レジストリーに追加します。

```
db2set DB2_SQLROUTINE_PREPOPTS=options
```

ここで、*options* は、DB2 プリコンパイラーで使用するプリコンパイル・オプションのリストを示します。使用できるオプションは、以下のものだけです。

```
| BLOCKING {UNAMBIG | ALL | NO}  
| DATETIME {DEF | USA | EUR | ISO | JIS | LOC}  
| DEGREE {1 | degree-of-parallelism | ANY}  
| DYNAMICRULES {BIND | RUN}  
| EXPLAIN {NO | YES | ALL}  
| EXPLSNAP {NO | YES | ALL}  
| FEDERATED {NO | YES}  
| INSERT {DEF | BUF}  
| ISOLATION {CS |RR |UR |RS |NC}  
| QUERYOPT optimization-level
```

例。日付には ISO フォーマットを、分離レベルには反復可能読み取りをパッケージで使用するには、以下のようにオプションを指定します。

```
db2set DB2_SQLROUTINE_PREPOPTS="DATETIME ISO ISOLATION RR"
```

次に DB2 をいったん停止してから再始動して、変更を有効化します。

関連タスク:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『SQL プロシージャ用の中間ファイルの保存』

- アプリケーション開発ガイド アプリケーションの構築および実行 の『SQL プロシージャのバックアップとリストア』
- 780 ページの『SQL プロシージャの作成』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『CALL ステートメントによるストアド・プロシージャの呼び出し』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『UNIX でのクライアント・アプリケーションによる SQL プロシージャの呼び出し』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『Windows でのクライアント・アプリケーションによる SQL プロシージャの呼び出し』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『コンパイル済み SQL プロシージャの配布』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『SQL プロシージャの再バインド』

Windows SQL プロシージャ環境のセットアップ

SQL プロシージャは、Windows NT、Windows 2000、Windows XP、および Windows .NET Server の各 Windows オペレーティング・システムでサポートされます。

SQL プロシージャでは、サーバー上のコンパイラ構成を使用して DB2_SQLROUTINE_COMPILER_PATH と DB2_SQLROUTINE_COMPILE_COMMAND の 2 つの環境変数を設定する必要があります。コンパイラの環境変数が SYSTEM 変数として設定されていると、構成は必要ありません。

制限:

64 ビットの Windows では、変数 DB2_SQLROUTINE_COMPILER_PATH を設定しようとする問題が生じることがあります。この変数の場合、ファイルの絶対パスが必要であり、しかも引き数 / スイッチは使用できないからです。したがって、もしユーザーが以下のように指定すると、作動しません。

```
db2set DB2_SQLROUTINE_COMPILER_PATH="C:¥MsSdk64¥SetEnv.bat /XP64 /RETAIL"
```

その対処法としては、たとえば以下のように、適切なフラグを使用して Microsoft のセットアップ・バッチ・ファイルを呼び出す別のバッチ・ファイルを作成します。

```
db2set DB2_SQLROUTINE_COMPILER_PATH="C:\MsSdk64\SetEnvXP64.bat"
```

C:¥MsSdk64¥SetEnvXP64.bat の内容は以下のようになります。

```
call C:¥MsSdk64¥SetEnv.bat /XP64 /RETAIL
```

Windows の 32 ビット環境ではこの問題は起きません。vcvars32.bat ではパラメータは必要もなく、受け入れられることもないからです。

手順:

C++ コンパイラーが C: ドライブにインストールされていると仮定して、DB2_SQLROUTINE_COMPILER_PATH DB2 レジストリー変数を以下のように設定します。

Microsoft Visual C++ バージョン 5.0 の場合:

```
db2set DB2_SQLROUTINE_COMPILER_PATH="c:%devstudio%vc%bin%vcvars32.bat"
```

Microsoft Visual C++ バージョン 6.0 の場合:

```
db2set DB2_SQLROUTINE_COMPILER_PATH="c:%Micros~1%vc98%bin%vcvars32.bat"
```

必要であれば、ドライブまたはパスを変更して、ご使用のシステムの C++ コンパイラーのロケーションを反映させてください。

コンパイル・コマンドの場合、キーワード SQLROUTINE_FILENAME を使用して、生成された SQC、C、PDB、DEF、EXP、メッセージ・ログ、および共有ライブラリー・ファイルのファイル名を置き換えます。Microsoft Visual C++ バージョン 5.0 および 6.0 を使用するには:

```
| db2set DB2_SQLROUTINE_COMPILE_COMMAND="cl -Od -W2 /TC -D_X86_=1 -MD  
| -I%DB2PATH%¥include SQLROUTINE_FILENAME.c /link -d11  
| -def:SQLROUTINE_FILENAME.def /out:SQLROUTINE_FILENAME.d11  
| %DB2PATH%\lib\db2api.lib"
```

DB2_SQLROUTINE_COMPILE_COMMAND DB2 レジストリー変数が設定されていない場合、これがデフォルトのコンパイル・コマンドになります。

以下に、Microsoft Visual C++ バージョン 6.0 コンパイラー用の環境セットアップ・コマンドの例を示してあります。下記のコマンドを実行するには、コマンドを切り取ってバッチ・ファイルに貼り付けてから、DB2 コマンド・ウィンドウでそのファイルを実行します。ご使用の特定の環境に合わせて、パスの設定を含めて、必要な変更をすべて加えてください。

```
| @echo on  
| rem Setting the SQL PROCEDURE environment:  
| db2set DB2_SQLROUTINE_COMPILER_PATH="c:%Micros~1%vc98%bin%vcvars32.bat"  
| db2set DB2_SQLROUTINE_COMPILE_COMMAND="cl -Od -W2 /TC -D_X86_=1 -MD  
| -I%DB2PATH%¥include SQLROUTINE_FILENAME.c /link -d11  
| -def:SQLROUTINE_FILENAME.def /out:SQLROUTINE_FILENAME.d11  
| %DB2PATH%¥lib¥db2api.lib"  
| @echo off
```

関連タスク:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『SQL プロシージャ用の中間ファイルの保存』

- 754 ページの『SQL プロシージャのプリコンパイル・オプションとバインド・オプションのカスタマイズ』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『SQL プロシージャのバックアップとリストア』
- 780 ページの『SQL プロシージャの作成』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『CALL ステートメントによるストアード・プロシージャの呼び出し』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『Windows でのクライアント・アプリケーションによる SQL プロシージャの呼び出し』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『コンパイル済み SQL プロシージャの配布』

関連資料:

- 766 ページの『アプリケーションの作成と実行用として Windows でサポートされるソフトウェア』

Application Development Client

DB2 Application Development (DB2 AD) Client は、分散リレーショナル・データベース体系 (DRDA) を実現する DB2 サーバーおよびアプリケーション・サーバーにアクセスするアプリケーションを開発するのに必要なツールおよび環境を提供します。

DB2 AD クライアントをインストールすれば、DB2 アプリケーションを構築し、実行することができます。以下の DB2 クライアントで DB2 アプリケーションを実行することもできます。

- DB2 Run-Time Client
- DB2 Administration Client

本書で記述しているプラットフォーム用の DB2 AD クライアントには、以下のものが含まれます。

- **C/C++、COBOL、および Fortran プリコンパイラー** (ただし、該当するプラットフォームでこの言語がサポートされている場合)。
- プログラミング・ライブラリー、組み込みファイル、およびコード・サンプルを含む、**組み込み SQL アプリケーション・サポート**。
- ODBC SDK への移植や ODBC SDK とのコンパイルが容易に行えるアプリケーションを開発するためのプログラミング・ライブラリー、組み込みファイル、およびコード例を含む、**DB2 コール・レベル・インターフェース (DB2 CLI) アプリケーション・サポート**。ODBC SDK は、Windows オペレーティング・システムの場合は Microsoft、および他のサポートされるプラットフォームの場合はさまざまなベンダーから入手可能です。Windows オペレーティング・システムの場合、DB2 クライアントには Microsoft ODBC Software Developer's Kit で開発されたアプリケーション

をサポートする、ODBC ドライバーが含まれています。他のすべてのプラットフォームについては、そのプラットフォーム用の ODBC SDK があれば、それを使用して開発されたアプリケーションをサポートする、任意でインストールされた ODBC ドライバーが DB2 クライアントに含まれます。Windows オペレーティング・システム用の DB2 クライアントだけに ODBC ドライバー・マネージャーが含まれています。

- Java アプリケーションおよびアプレットを開発するための DB2 Java Database Connectivity (DB2 JDBC) サポート、および Java 組み込み SQL アプリケーションとアプレットを開発するための DB2 embedded SQL for Java (DB2 SQLj) サポートを含む、**DB2 Java Enablement**。
- **Java Development Kit (JDK)**。IBM for AIX の JDK 1.3.1 および Java Runtime Environment (JRE) 1.3.1、IBM Developer Kit および Runtime Environment (Linux 版 および Windows 版) バージョン 1.3.1、HP-UX Software Developer's Kit および Runtime Environment 1.3.1 for HP-UX、および Sun Microsystems の Java Development Kit 1.3.1 for Solaris。Solaris を除いて、Java の実行を必要とするいずれかのコンポーネントをインストールすることが決まった場合、該当するオペレーティング・システム用の JDK をインストールします。いずれのコンポーネントも選択しなくても、JDK をインストールすることはできます。Solaris では、DB2 に付属している JDK 1.3.1 がユーザーによってインストールされなければなりません。
- **ActiveX Data Objects (ADO) およびオブジェクトのリンクと埋め込み (OLE) オートメーション・ルーチン (UDF とストアード・プロシージャ)**。これには、Microsoft Visual Basic および Microsoft Visual C++ のコード・サンプルのインプリメンテーションも含まれます。また、Microsoft Visual Basic でインプリメントされた Remote Data Object (RDO) を持つコード・サンプル。
- Windows オペレーティング・システムでは、**オブジェクトのリンクと埋め込みデータベース (OLE DB) 表関数**。
- **DB2 デベロップメント・センター**。これは、ルーチン (ストアード・プロシージャとユーザー定義関数) の迅速な開発と構造型をサポートするグラフィカル・アプリケーションです。デベロップメント・センターは、ワークステーションから z/OS までの DB2 ファミリー全体をサポートする単一の開発環境として機能します。デベロップメント・センターは、スタンドアロンのアプリケーションとして立ち上げることができますが、コントロール・センター、コマンド・センター、またはタスク・センターなどの DB2 Universal Database のセンターから立ち上げることもできます。デベロップメント・センターは、Java を使用してインプリメントするので、データベース接続の管理には Java Database Connectivity (JDBC) API を使用します。また、デベロップメント・センターには、以下の各開発環境用の DB2 Development Add-In も備えられています。
 - Microsoft Visual C++ バージョン 6
 - Microsoft Visual Basic バージョン 6
 - Microsoft Visual InterDev バージョン 6

- SQL ステートメントのプロトタイピングまたはデータベースの随時照会の実行のための、コマンド・センターまたはコマンド行プロセッサ (CLP) を介する**対話式 SQL**。
- 他のアプリケーション開発ツールの製品が、それらの製品内で DB2 用のプリコンパイラー・サポートを直接実現するための**文書化された API のセット**。たとえば AIX 上の IBM COBOL は、このインターフェースを使用します。プリコンパイラー・サービス API のセットに関する情報は、以下の DB2 アプリケーション開発 Web サイトにある PDF ファイル `prepapi.pdf` に記載されています。

`http://www.ibm.com/software/data/db2/udb/ad`
- **SQL92 および MVS 適合 flagger**。これは、ISO/ANSI SQL92 Entry Level 基準に適合しないアプリケーションや、DB2 UDB (OS/390 および z/OS 版) がサポートしないアプリケーション内の組み込み SQL ステートメントを識別します。ワークステーション上で開発したアプリケーションを他のプラットフォームに移行する場合には、Flagger によって構文の非互換性が示されるため、時間が節約できます。

関連資料:

- 1190 ページの『PRECOMPILE』
- 759 ページの『アプリケーションの作成と実行用として AIX でサポートされるソフトウェア』
- 761 ページの『アプリケーションの作成と実行用として HP-UX でサポートされるソフトウェア』
- 762 ページの『アプリケーションの作成と実行用として Linux でサポートされるソフトウェア』
- 765 ページの『アプリケーションの作成と実行用として Solaris でサポートされるソフトウェア』
- 766 ページの『アプリケーションの作成と実行用として Windows でサポートされるソフトウェア』

アプリケーションの作成と実行用として AIX でサポートされるソフトウェア

DB2 (AIX 版) は、以下のオペレーティング・システムをサポートします。

AIX/6000

バージョン 4.3.3 (保守レベル 9 以上)、および APAR IY22308、Y32690 と IY33024 (32 ビットのみ)

バージョン 5.1.0 (保守レベル 2 以上)、および APAR IY31254、IY32217、IY32905 と IY33023 (32 ビット)

バージョン 5.1.0 (保守レベル 2 以上)、および APAR IY31254、IY32217、IY32466、IY32905 と IY33023 (64 ビット)

バージョン 5.2.0 (32 ビットおよび 64 ビット)

注: 以下のコマンドを使用して、特定の APAR がシステムにインストールされているかどうかを照会できます。

```
instfix -v -i -k <APAR>
```

たとえば、`instfix -v -i -k IY31254`

DB2 (AIX 版) は、以下のプログラム言語およびコンパイラーをサポートします。

C IBM C for AIX バージョン 5.0.2.3

C++ IBM VisualAge C++ バージョン 5.0.2.3

COBOL

IBM COBOL Set for AIX バージョン 1.1

Micro Focus COBOL Server Express バージョン 2.0.10

Fortran

IBM XL Fortran for AIX バージョン 4.1 (32 ビットの場合のみ) および 5.1.0 (32 ビットおよび 64 ビットの場合)

Java Java Development Kit (JDK) バージョン 1.3.1 および IBM 提供の Java Runtime Environment (JRE) (AIX 版) バージョン 1.3.1 (DB2 の必要に応じてインストールされます)。

注: Java の実行を必要とするいずれかのコンポーネントをインストールすることが決められた場合、DB2 はこの JDK をインストールします。いずれのコンポーネントも選択しなくても、JDK をインストールすることはできます。

Perl Perl 5.004_04 以上、DBI 0.93 以上

REXX IBM AIX REXX/6000 AISPO 製品番号: 5764-057

IBM Object REXX (AIX 版) バージョン 1.1

REXXSAA 4.00

注: REXX サポートは、32 ビット用のみです。

DB2 (AIX 版) のソフトウェア・サポートの更新の詳細は、以下の DB2 アプリケーション開発の Web サイトを参照してください。

<http://www.ibm.com/software/data/db2/udb/ad>

関連資料:

- *アプリケーション開発ガイド アプリケーションの構築および実行* の『アプリケーションの作成と実行用として DB2 でサポートされるソフトウェア』
- *DB2 Universal Database サーバー機能 概説およびインストール* の『パーティション DB2 サーバーのインストール要件 (AIX)』

- アプリケーション開発ガイド アプリケーションの構築および実行 の『AIX C アプリケーションのコンパイルとリンクのオプション』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『AIX C ルーチンのコンパイルとリンクのオプション』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『AIX C++ アプリケーションのコンパイルとリンクのオプション』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『AIX C++ ルーチンのコンパイルとリンクのオプション』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『AIX IBM COBOL アプリケーションのコンパイルとリンクのオプション』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『AIX IBM COBOL ルーチンのコンパイルとリンクのオプション』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『AIX Micro Focus COBOL アプリケーションのコンパイルとリンクのオプション』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『AIX Micro Focus COBOL ルーチンのコンパイルとリンクのオプション』
- DB2 Universal Database サーバー機能 概説およびインストール の『DB2 サーバーのインストール要件 (AIX)』

アプリケーションの作成と実行用として HP-UX でサポートされるソフトウェア

DB2 (HP-UX 版) は、以下のオペレーティング・システムをサポートします。

HP-UX HP-UX バージョン 11i(11.11) for PA-RISC 2 (以下が適用されたもの)

2001 年 12 月 GOLDBASE11i および 2001 年 12 月 GOLDAPPS11i バンドル
およびパッチ PHSS_26263

DB2 (HP-UX 版) は、以下のプログラム言語およびコンパイラーをサポートします。

C HP C コンパイラーのバージョン B.11.11.02

C++ HP aC++ バージョン A.03.31

COBOL

Micro Focus COBOL バージョン 4.1

Fortran

HP Fortran/9000 バージョン 10.0

HP-UX F77 B.11.00.01

Java HP-UX 32 ビット: Software Developer's Kit および Runtime Environment 1.3.1 for HP-UX 11.0 および Hewlett-Packard の 11i PA-RISC (DB2 で必要に応じてインストールされる)。

HP-UX 64 ビット: Software Developer's Kit および Runtime Environment 1.4 for HP-UX 11.0 および Hewlett-Packard の 11i PA-RISC (DB2 for HP-UX CD-ROM で提供)。

注: Java の実行を必要とするいずれかのコンポーネントのインストールが選択された場合、DB2 は 1.3.1 SDK をインストールします。いずれのコンポーネントも選択しなくても、SDK をインストールすることはできます。1.4 SDK は、DB2 for HP-UX CD-ROM から手作業でインストールできます。

Perl Perl 5.004_04 以上、DBI 0.93 以上

DB2 (HP-UX 版) のソフトウェア・サポートの更新の詳細は、以下の DB2 アプリケーション開発の Web サイトを参照してください。

<http://www.ibm.com/software/data/db2/udb/ad>

関連資料:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『アプリケーションの作成と実行用として DB2 でサポートされるソフトウェア』
- *DB2 Universal Database* サーバー機能 概説およびインストール の『パーティション DB2 サーバーのインストール要件 (HP-UX)』
- 788 ページの『HP-UX C アプリケーションのコンパイルとリンクのオプション』
- 789 ページの『HP-UX C ルーチンのコンパイルとリンクのオプション』
- 791 ページの『HP-UX C++ アプリケーションのコンパイルとリンクのオプション』
- 792 ページの『HP-UX C++ ルーチンのコンパイルとリンクのオプション』
- 794 ページの『HP-UX Micro Focus COBOL アプリケーションのコンパイルとリンクのオプション』
- 795 ページの『HP-UX Micro Focus COBOL ルーチンのコンパイルとリンクのオプション』
- *DB2 Universal Database* サーバー機能 概説およびインストール の『DB2 サーバーのインストール要件 (HP-UX)』

アプリケーションの作成と実行用として Linux でサポートされるソフトウェア

DB2 for Linux for Intel x86 (32 ビット・アーキテクチャー) は、以下のオペレーティング・システム環境をサポートします。

Linux カーネルのバージョン 2.4.18、glibc バージョン 2.2.4、および rpm (インストールの必要あり)。

| **注:** 必要な最低限のカーネル・レベルは、変更される可能性があります。サポートされ
| ている配布レベルおよびカーネル・レベルの最新情報については、以下のサイトを
| 参照してください。

| <http://www.ibm.com/db2/linux/validate>

DB2 for Linux on S/390 は、以下のオペレーティング・システム環境をサポートします。

以下のいずれかになります。

- RedHat v7.2
- SuSE SLES-7 for Linux on S/390

DB2 for Linux for Intel x86 および S/390 は、以下のプログラム言語とコンパイラーをサポートします。

C/C++ Linux on Intel の場合: GNU/Linux gcc および g++ バージョン 2.95.3 および 2.96

Linux/390 の場合: GNU/Linux gcc および g++ バージョン 2.95.3

Linux on Intel の gcc および g++ バージョン 3.0 サポートおよび Linux/390 の今後のバージョン・サポートについては、DB2 アプリケーション開発の Web サイトにアクセスしてください。

<http://www.ibm.com/software/data/db2/udb/ad>

Java Linux on Intel の場合: IBM Developer Kit および Runtime Environment (Linux 版)、Java 2 Technology Edition バージョン 1.3.1 の 32 ビット・バージョン (DB2 での必要に応じてインストールされます)。

Linux/390 の場合: IBM zSeries Developer Kit for Linux, Java 2 Technology Edition (Sun 1.3.1 SDK レベル、DB2 での必要に応じてインストールされます)。

注: Java の実行を必要とするいずれかのコンポーネントのインストールが選択された場合、DB2 は該当する Developer Kit をインストールします。コンポーネントを選択しなくても、Developer Kit をインストールすることはできます。

Perl Perl 5.004_04 以上、DBI 0.93 以上

REXX Linux on Intel の場合: Object REXX Interpreter for Linux バージョン 2.1

Linux/390 の場合: Object REXX 2.2.0 for Linux/390

DB2 for Linux on IA64 は、以下のオペレーティング・システム環境をサポートします。

以下のいずれかになります。

- Red Hat 7.2
- SuSE SLES-7 for Linux on IA64

DB2 (IA64 版) は、以下のプログラム言語およびコンパイラーをサポートします。

C GNU/Linux gcc バージョン 3.0.2

C++ GNU/Linux g++ バージョン 3.0.2

注: このバージョンのコンパイラーは、`fstream` 関数で、整数パラメーターを受け入れないものがあります。詳しくは、コンパイラーの資料を参照してください。

Java IBM Developer Kit および Runtime Environment (Linux 版)、Java 2 Technology Edition バージョン 1.3.1 の 64 ビット・バージョン (DB2 での必要に応じてインストールされます)。この JDK を使用するには、gcc 3.0.2 と gcc3 libstdc++ ランタイム・ライブラリーもインストールする必要があります。

注: Java の実行を必要とするいずれかのコンポーネントのインストールが選択された場合、DB2 はこの Developer Kit をインストールします。コンポーネントを選択しなくても、Developer Kit をインストールすることはできます。

Perl Perl 5.6

注: DB2 の 32 ビット・アプリケーションまたはルーチン (ストアード・プロシージャおよびユーザー定義関数) の実行は、Linux IA64 ではサポートされていません。

DB2 (Linux 版) のソフトウェア・サポートの更新の詳細は、以下の DB2 アプリケーション開発の Web サイトを参照してください。

<http://www.ibm.com/software/data/db2/udb/ad>

関連資料:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『アプリケーションの作成と実行用として DB2 でサポートされるソフトウェア』
- DB2 Universal Database サーバー機能 概説およびインストール の『パーティション DB2 サーバーのインストール要件 (Linux)』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『Linux C アプリケーションのコンパイルとリンクのオプション』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『Linux C ルーチンのコンパイルとリンクのオプション』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『Linux C++ アプリケーションのコンパイルとリンクのオプション』

- アプリケーション開発ガイド アプリケーションの構築および実行 の『Linux C++ ルーチンのコンパイルとリンクのオプション』
- DB2 Universal Database Personal Edition 概説およびインストール の『DB2 Personal Edition のインストール要件 (Linux)』
- DB2 Universal Database サーバー機能 概説およびインストール の『DB2 サーバーのインストール要件 (Linux)』

アプリケーションの作成と実行用として Solaris でサポートされるソフトウェア

DB2 (Solaris 版) は、以下のオペレーティング・システムをサポートします。

Solaris Solaris 7 (32 ビット) パッチ 106327-10

Solaris 7 (64 ビット) パッチ 106300-11

Solaris 8 (32 ビット) パッチ 108434-03 および 108528-12

Solaris 8 (64 ビット) パッチ 108435-03 および 108528-12

Solaris 9 (32 ビット)

Solaris 9 (64 ビット)

Fujitsu PRIMEPOWER には、以下のパッチも必要です (64 ビット)。

Solaris 8 KU15 パッチ 912040-01

Solaris 9 FCS パッチ 912041-01

DB2 を Solaris にインストールするには、ソフトウェア、SUNWlibC が必要です。

Java をサポートするには、以下のパッチも必要です。

Solaris 7 「推奨 & セキュリティー・パッチ」 + 107226-17 + 107153-01

Solaris 8 「推奨 & セキュリティー・パッチ」 + 108921-12 + 108940-24

「推奨 & セキュリティー・パッチ」は、次のところから入手できます。

<http://sunsolve.sun.com>

DB2 (Solaris 版) は、以下のプログラム言語およびコンパイラーをサポートします。

C Forte C バージョン 5.0、6、および 6.1

注: これらのコンパイラーの旧称は SPARCompiler です。

C++ Forte C++ バージョン 5.0、6、および 6.1

注: これらのコンパイラーの旧称は SPARCompiler です。

COBOL

Micro Focus COBOL Server Express バージョン 2.0.10

Fortran

SPARCCompiler Fortran バージョン 4.2 および 5.0

Java Solaris 32 ビット: Sun Microsystems の Java Development Kit (JDK) (Solaris 版) バージョン 1.3.1 (DB2 と一緒に出荷)。

Solaris 64 ビット: Sun Microsystems の Java Development Kit (JDK) (Solaris 版) バージョン 1.4 (DB2 と一緒に出荷)。

Perl Perl 5.004_04 以上、DBI 0.93 以上

DB2 (Solaris 版) のソフトウェア・サポートの更新の詳細は、以下の DB2 アプリケーション開発の Web サイトを参照してください。

<http://www.ibm.com/software/data/db2/udb/ad>

関連資料:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『アプリケーションの作成と実行用として DB2 でサポートされるソフトウェア』
- *DB2 Universal Database* サーバー機能 概説およびインストール の『パーティション DB2 サーバーのインストール要件 (Solaris オペレーティング環境)』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『Solaris C アプリケーションのコンパイルとリンクのオプション』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『Solaris C ルーチンのコンパイルとリンクのオプション』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『Solaris C++ アプリケーションのコンパイルとリンクのオプション』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『Solaris C++ ルーチンのコンパイルとリンクのオプション』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『Solaris Micro Focus COBOL アプリケーションのコンパイルとリンクのオプション』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『Solaris Micro Focus COBOL ルーチンのコンパイルとリンクのオプション』
- *DB2 Universal Database* サーバー機能 概説およびインストール の『DB2 サーバーのインストール要件 (Solaris)』

アプリケーションの作成と実行用として Windows でサポートされるソフトウェア

DB2 (Windows 32 ビット・オペレーティング・システム版) は、以下をサポートします。

Microsoft Windows XP

Microsoft Windows .NET

Microsoft Windows 2000

Windows Terminal Server には Service Pack 2 が必要です。

Microsoft Windows NT

Service Pack 6a 付きのバージョン 4.0 以降。

Microsoft Windows ME

Microsoft Windows 98

DB2 (Windows 32 ビット・オペレーティング・システム版) は、以下のプログラム言語をサポートします。

Basic Microsoft Visual Basic バージョン 4.2 およびバージョン 5.0

Microsoft Visual Basic 6.0 Professional Edition

C/C++ Microsoft Visual C++ バージョン 5.0 および 6.0

32 ビット・アプリケーション用の Intel C++ Compiler バージョン 5 以上

COBOL

Micro Focus COBOL バージョン 4.0.20

Micro Focus COBOL Net Express バージョン 3.1.0

IBM VisualAge COBOL バージョン 3.0.4 以上

REXX IBM Object REXX for Windows NT/95 バージョン 1.1

IBM Object REXX for Windows の入手法の詳細は、以下を参照してください。

<http://www.ibm.com/software/ad/obj-rexx/>

Java IBM Developer Kit および Runtime Environment (Windows 版)、Java 2 Technology Edition バージョン 1.3.1 の 32 ビット・バージョン (DB2 での必要に応じてインストールします)。

注: Java の実行を必要とするいずれかのコンポーネントのインストールが選択された場合、DB2 はこの Developer Kit をインストールします。コンポーネントを選択しなくても、Developer Kit をインストールすることはできます。

Java Development Kit (JDK) 1.3.1 (Win32 版) (Sun Microsystems 社提供)

Perl Perl 5.004_04, DBI 0.93

Microsoft Windows Scripting Host

バージョン 5.1

DB2 (Windows 64 ビット・オペレーティング・システム版) は、以下をサポートしません。

Microsoft Windows XP 64 ビット版

Microsoft Windows .NET Server 64 ビット版

DB2 (Windows 64 ビット・オペレーティング・システム版) は、以下のプログラム言語をサポートします。

C/C++ Intel C++ Compiler for Itanium バージョン 6.0

Intel Itanium アーキテクチャー用の Microsoft の C/C++ コンパイラー

Java IBM Developer Kit および Runtime Environment (Windows 版)、 Java 2 Technology Edition バージョン 1.3.1 の 64 ビット・バージョン (DB2 での必要に応じてインストールされます)。

注: Java の実行を必要とするいずれかのコンポーネントのインストールが選択された場合、 DB2 はこの Developer Kit をインストールします。コンポーネントを選択しなくても、Developer Kit をインストールすることはできません。

Microsoft Windows Scripting Host

バージョン 5.1

注: Windows .NET Server には、以下がすべて組み込まれています。

- Windows .NET Web Server
- Windows .NET Standard Server
- Windows .NET Enterprise Server
- Windows .NET Datacenter Server

DB2 for Windows のソフトウェア・サポートの更新の詳細は、以下の DB2 アプリケーション開発の Web サイトを参照してください。

<http://www.ibm.com/software/data/db2/udb/ad>

関連資料:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『アプリケーションの作成と実行用として DB2 でサポートされるソフトウェア』
- *DB2 Universal Database* サーバー機能 概説およびインストール の『DB2 サーバーのインストール要件 (Windows)』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『Windows C/C++ アプリケーションのコンパイルとリンクのオプション』
- 785 ページの『Windows C/C++ ルーチンのコンパイルとリンクのオプション』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『Windows IBM COBOL アプリケーションのコンパイルとリンクのオプション』

- アプリケーション開発ガイド アプリケーションの構築および実行 の『Windows IBM COBOL ルーチンのコンパイルとリンクのオプション』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『Windows Micro Focus COBOL アプリケーションのコンパイルとリンクのオプション』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『Windows Micro Focus COBOL ルーチンのコンパイルとリンクのオプション』
- DB2 Universal Database Personal Edition 概説およびインストール の『DB2 Personal Edition のインストール要件 (Windows)』
- DB2 Universal Database サーバー機能 概説およびインストール の『パーティション DB2 サーバーのインストール要件 (Windows)』

ビルド・ファイル

サンプル・プログラムの作成の例を示すのに使用するファイルは、UNIX[®] ではスクリプト・ファイル、Windows ではバッチ・ファイルと呼ばれます。本書では、これらのファイルを総称してビルド・ファイルと呼ぶことにします。このファイルには、サポートされているプラットフォーム・コンパイラ用に推奨されるコンパイルとリンクのコマンドが入っています。

サポートされるプラットフォームでの各言語に対するビルド・ファイルが DB2[®] によって提供されています。そこには、各言語のサンプル・プログラムと同じディレクトリー内に、作成されるタイプのプログラムが入っています。以下の表に、さまざまなタイプのプログラムを作成するためのさまざまなタイプのビルド・ファイルを一覧で示してあります。他に明記されていない限り、これらのビルド・ファイルは、サポートされているすべてのプラットフォーム上のサポートされている言語用のファイルです。Windows ではこのビルド・ファイルには .bat (バッチ) の拡張子が付いていますが、この表では付けられていません。UNIX プラットフォームの場合には、拡張子はありません。

表 177. DB2 ビルド・ファイル

ビルド・ファイル	構築されるプログラムのタイプ
bldapp	アプリケーション・プログラム
bldrtn	ルーチン (ストアード・プロシージャと UDF)
bldsqlj	Java [™] SQLj アプリケーション
bldsqljs	Java SQLj ルーチン (ストアード・プロシージャおよび UDF)
bldmc	C/C++ 複数接続アプリケーション
bldmt	UNIX C/C++ マルチスレッド・アプリケーション
bldcli	sqlproc サンプル・サブディレクトリー内の SQL プロシージャ用の CLI クライアント・アプリケーション
bldevm	AIX [®] および Windows [®] イベント・モニター・プログラム evm (C 言語のみ)

注: bldcli ファイルは、 samples/cli ディレクトリー内の bldapp ファイルと同じです。これに別の名前が付いているのは、組み込み C の bldapp ファイルは samples/sqlproc ディレクトリーにも入っているからです。

以下の表は、プラットフォーム別およびプログラム言語別のビルド・ファイルと、それが置かれているディレクトリーを一覧で示しています。オンライン文書では、ビルド・ファイルの名前が HTML のソース・ファイルにホット・リンクされています。該当するサンプル・ディレクトリー内のテキスト・ファイルにアクセスすることもできます。

表 178. 言語別およびプラットフォーム別のビルド・ファイル

プラットフォーム → 言語	AIX	HP-UX	Linux	Solaris	Windows
C samples/c	bldapp bldrtn bldmt bldmc bldevm	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp.bat bldrtn.batbldrtn.bat bldmc.batbldmc.bat bldevm.batbldevm.bat
C++ samples/cpp	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp.bat bldrtn.batbldrtn.bat bldmc.batbldmc.bat
CLI samples/cli	bldapp bldrtn bldmc	bldapp bldrtn bldmc	bldapp bldrtn bldmc	bldapp bldrtn bldmc	bldapp.bat bldrtn.batbldrtn.bat bldmc.batbldmc.bat
SQLj samples/java/sqlj	bldsqlj bldsqljs	bldsqlj bldsqljs	bldsqlj bldsqljs	bldsqlj bldsqljs	bldsqlj.bat bldsqljs.bat
IBM® COBOL samples/cobol	bldapp bldrtn	n/a	n/a	n/a	bldapp.bat bldrtn.batbldrtn.bat
Micro Focus COBOL samples/cobol_mf	bldapp bldrtnbldrtn	bldapp bldrtnbldrtn	n/a	bldapp bldrtnbldrtn	bldapp.bat bldrtn.bat

本書でアプリケーションとルーチンの作成にビルド・ファイルが使われているのは、DB2 がサポートされているコンパイラーに対しお勧めする、コンパイルおよびリンクのオプションが、それによって明らかに実体として示されるからです。通常はこれらの他にも使用できるコンパイルとリンクのオプションは多数あり、ユーザーはそれらを自由に試すことができます。用意されているコンパイルとリンクのすべてのオプションについて知りたい場合は、ご使用のコンパイラーのマニュアルを参照してください。開発者はそれらのビルド・ファイルを使用してサンプル・プログラムを構築するだけでなく、自分のプログラムを構築することも可能です。サンプル・プログラムをユーザーが変更できるテンプレートとして利用することにより、プログラム開発に役立てることができます。

都合のよいことに、コンパイラーで許容される任意のファイル名でソース・ファイルを構築できるようにビルド・ファイルは設計されています。これは、プログラム名がファ

イル中にハードコーディングされる `makefile` とは異なります。 `makefile` は、作成したプログラムのコンパイルとリンクのためにビルド・ファイルにアクセスします。ビルド・ファイルは、UNIX の場合には `$1` 変数、Windows オペレーティング・システムの場合には `%1` 変数を使用して、プログラム名を内部的に置き換えます。このような変数名中の数字は、順に大きくなって、他の引き数が必要となるごとに入れ換わります。

たとえばスタンドアロン・アプリケーション、ルーチン (ストアド・プロシージャおよび UDF)、または複数接続あるいはマルチスレッド・プログラムなどのもっと特殊なプログラム・タイプといった、個々の種類のプログラム作成に対して個々のビルド・ファイルがそれぞれ適応しているため、迅速かつ簡単にビルド・ファイルを試してみることができます。ビルド・ファイルの設計目的に沿った種類のプログラムがコンパイラでサポートされてさえいれば、どこでもすべてのタイプのビルド・ファイルを利用することができます。

ビルド・ファイルが作成するオブジェクト・ファイルや実行可能ファイルは、ソース・ファイルが修正されない場合でさえ、プログラムが構築されるたびに自動的に上書きされます。それは、`makefile` を使用する場合には当てはまりません。つまり、開発者は以前のオブジェクト・ファイルや実行可能ファイルを削除したり、またはソースを修正したりすることなく、既存のプログラムを再構築することができます。

ビルド・ファイルには、サンプル・データベース用のデフォルト設定が組み込まれています。ユーザーが別のデータベースにアクセスする場合は、別のパラメーターを指定してデフォルトの指定を変更するだけで済みます。その別のデータベースを一貫して使用する予定であれば、ビルド・ファイルの中にある `sample` を置き換えて、このデータベースの名前をハードコーディングすることができます。

組み込み SQL プログラムに使用されるビルド・ファイルは、組み込み SQL プログラムのためのプリコンパイルとバインドのステップが入っている、別のファイル `embprep` を呼び出します。これらのステップでは、組み込み SQL プログラムをどこに作成するかによって、ユーザー ID とパスワード用のオプションのパラメーターが必要になる場合があります。

SQLj の場合を除いて、データベースが置かれているサーバー・インスタンス上で開発者がプログラムを作成する場合は、ユーザー ID とパスワードはどちらにも共通であるため、指定する必要はありません。それに対して、開発者が別のインスタンスで作業する場合、たとえばサーバー・データベースにリモートからアクセスするクライアント・マシン上で作業する場合などは、これらのパラメーターを指定する必要があります。

SQLj ビルド・ファイルの場合、ローカル・データベースへのアクセスであっても、`db2sqljcustomize` カスタマイザー用のユーザー ID とパスワードが必要です。これは、タイプ 4 ドライバーの規則に準拠しています。

最後の点として、ビルド・ファイルは開発者が自分の都合に合わせて修正することが可能です。開発者は (前述のように) ビルド・ファイル中のデータベース名を変更できる

だけでなく、他のパラメーターをファイル内にハードコーディングしたり、コンパイルとリンクのオプションを変更したり、デフォルトの DB2 インスタンス・パスを変更したりすることが簡単に行えます。ビルド・ファイルはその性質上、簡単で分かりやすく、具体的であるため、自分の必要に応じてそれらのファイルに手を加えるのが容易です。

関連概念:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『makefile』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『エラー・チェック・ユーティリティー』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『サンプル・ファイル』

関連資料:

- 759 ページの『アプリケーションの作成と実行用として AIX でサポートされるソフトウェア』
- 761 ページの『アプリケーションの作成と実行用として HP-UX でサポートされるソフトウェア』
- 762 ページの『アプリケーションの作成と実行用として Linux でサポートされるソフトウェア』
- 765 ページの『アプリケーションの作成と実行用として Solaris でサポートされるソフトウェア』
- 766 ページの『アプリケーションの作成と実行用として Windows でサポートされるソフトウェア』

Java アプレットに関する考慮事項

DB2[®] には、Java[™] アプレットを使用してアクセスすることができます。これを使用する際は、以下の点に注意してください。

1. すでに使用するべきでなくなったタイプ 3 ドライバー ('net' ドライバーとも呼びます) を使用している場合、Java アプレットで使用される db2java.zip ファイルが、JDBC アプレット・サーバーと同じフィックスパック・レベルにあることが不可欠です。通常の場合では、db2java.zip は、JDBC アプレット・サーバーが実行されている Web サーバーからロードされます。それによってレベルは必ず同一になります。ただし、構成が別のロケーションから db2java.zip をロードする Java アプレットを持っている場合は、不一致が起きることがあります。2 つのファイル間でのフィックスパック・レベルの一致は、接続時に厳しく強制されます。不一致が見つかったら、接続はリジェクトされ、クライアントは以下の例外の 1 つを受け取ります。
 - db2java.zip が DB2 バージョン 7 フィックスパック 2 以上の場合:


```
COM.ibm.db2.jdbc.DB2Exception: [IBM][JDBC Driver]
CLI0621E Unsupported JDBC server configuration.
```

- db2java.zip がフィックスパック 2 以前の場合:

```
COM.ibm.db2.jdbc.DB2Exception: [IBM][JDBC Driver]
CLI0601E Invalid statement handle or statement is closed.
SQLSTATE=S1000
```

不一致が起きた場合、JDBC アプレット・サーバーは以下のメッセージの 1 つを jdbcerr.log ファイルに記録します。

- JDBC アプレット・サーバーが DB2 バージョン 7 フィックスパック 2 以降の場合:

```
jdbcFSQLConnect: JDBC Applet Server and client (db2java.zip)
versions do not match. Unable to proceed with connection., einfo= -111
```

- JDBC アプレット・サーバーがフィックスパック 2 以前の場合:

```
jdbcServiceConnection(): Invalid Request Received., einfo= 0
```

JDBC 環境をテストするには、sqllib¥samples¥java (Windows の場合) または sqllib/samples/java (UNIX の場合) にあるサンプル・ファイル db2JDBCVersion.java を使用します。db2JDBCVersion プログラムは、現在使用されている DB2 JDBC ドライバーのバージョンを調べ、それに対して JDBC 環境が正しく設定されているかどうかをチェックします。

アプレットをタイプ 4 ドライバーに移行することを強くお勧めします。

2. いくつかの Java クラスから成る大型の JDBC または SQLj アプレットの場合は、そのクラスすべてを単一の JAR ファイルにパッケージすることができます。また、SQLj アプレットでは、そのクラスと一緒に、シリアライズされたプロファイルのパッケージする必要があります。これを選択する場合、JAR ファイルを "applet" タグの archive パラメーターに追加します。詳細については、JDK バージョン 1.3 の資料を参照してください。

SQLj アプレットの場合、ブラウザーによっては、アプレットに関連付けられたリソース・ファイルからのシリアライズされたオブジェクトのロードをまだサポートしていません。たとえば、そのようなブラウザーで付属のサンプル・アプレット Applet をロードしようとする、以下のようなエラー・メッセージが出されます。

```
java.lang.ClassNotFoundException: Applet_SJProfile0
```

これを回避する方法としては、シリアライズ・プロファイルを Java クラス形式で保管するプロファイルに変換するユーティリティを使用することができます。このユーティリティは、sqlj.runtime.profile.util.SerProfileToClass という名前の

Java クラスです。これはシリアルライズ・プロファイルのリソース・ファイルを入力として取り込み、そのプロファイルを含んだ Java クラスを出力として生成します。プロファイルは以下のいずれかのコマンドを使用して変換できます。

```
profconv Applt_SJProfile0.ser
```

または

```
java sqlj.runtime.profile.util.SerProfileToClass Applt_SJProfile0.ser
```

このコマンドの結果、クラス `Applt_SJProfile0.class` が作成されます。アプレットが使用している `.ser` 形式のすべてのプロファイルを `.class` 形式のプロファイルに置き換えれば、問題はなくなるはずで

3. ファイル `db2java.zip` (JDBC タイプ 3 ドライバーを使用するアプレット用) または `db2jcc.jar` (JDBC タイプ 4 ドライバーを使用したアプレットまたは任意の SQLj アプレット用)、あるいはその両方を、Web サイトからロードすることのできるいくつかのアプレットによって共有されるディレクトリーに置くことができます。これらのファイルは、Windows® オペレーティング・システムの場合には `sql1lib\java` ディレクトリーに、また UNIX の場合には `sql1lib/java` ディレクトリーにあります。この場合、`codebase` パラメーターを、そのディレクトリーを識別する HTML ファイルの "applet" タグに追加する必要があることがあります。詳細については、JDK バージョン 1.3 の資料を参照してください。
4. DB2 バージョン 5.2 以降、JDBC アプレット・サーバー (listener) である `db2jd` の機能を強化するために信号処理機能が追加されています。その結果、`db2jd` は Ctrl-C で強制終了することができません。したがって、「kill -9」(Unix の場合) または「タスク マネージャ」(Windows の場合) を使用してプロセスを強制終了するのが、リスナーを終了させる唯一の方法です。
5. Web サーバー、特に Domino™ Go Webserver での DB2 Java アプレットの実行についての詳細は、以下のサイトを参照してください。

<http://www.ibm.com/software/data/db2/db2lotus/gojava.htm>

関連タスク:

- 741 ページの『Java 環境のセットアップ』
- 776 ページの『JDBC アプレットの作成』
- 778 ページの『SQLj アプレットの作成』

UNIX 用の SQLj アプリケーション・オプション

以下の表は、UNIX 上の `b1dsq1j` ビルド・スクリプトで使用する SQLj 変換プログラム・オプションとプリコンパイル・オプションを示しています。これらは、UNIX プラットフォームで SQLj アプリケーションおよびアプレットの作成時に使用するように DB2 で推奨するオプションです。

bldsqlj の変換プログラム・オプションとプリコンパイル・オプション

sqlj SQLj 変換プログラム (プログラムのコンパイルも行います)。

\$1.sqlj

SQLj ソース・ファイル。

db2sqljcustomize

Java プロファイル・カスタマイザーが使用する DB2。

-url データベース接続を確立するための JDBC URL を
jdbc:db2://servername:50000/sample のように指定します。

-user ユーザー ID を指定します。

-password
パスワードを指定します。

\$1_SJProfile0

プログラムのシリアル化プロファイルを指定します。

関連タスク:

- 778 ページの『SQLj アプレットの作成』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『SQLJ アプリケーションの作成』

関連資料:

- 775 ページの『UNIX の SQLj ストアード・プロシージャー・オプション』

関連サンプル:

- 『bldsqlj -- Builds Java embedded SQL (SQLJ) applications and applets on UNIX』

UNIX の SQLj ストアード・プロシージャー・オプション

以下の表は、UNIX 上の bldsqljs ビルド・スクリプトで使用する SQLj 変換プログラム・オプションとプリコンパイル・オプションを示しています。これらは、UNIX プラットフォームで SQLj ルーチン (ストアード・プロシージャーとユーザー定義関数) を作成する時に使用するように DB2 で推奨されているオプションです。

bldsqljs の変換プログラムとプリコンパイル・オプション

sqlj SQLj 変換プログラム (プログラムのコンパイルも行います)。

\$1.sqlj

SQLj ソース・ファイル。

db2sqljcustomize

Java プロファイル・カスタマイザーが使用する DB2。

-url データベース接続を確立するための JDBC URL を
jdbc:db2://servername:50000/sample のように指定します。

-user ユーザー ID を指定します。

-password

パスワードを指定します。

\$1_SJProfile0

プログラムのシリアル化プロファイルを指定します。

関連タスク:

- ・ アプリケーション開発ガイド アプリケーションの構築および実行 の『SQLJ ルーチンの作成』

関連資料:

- ・ 774 ページの『UNIX 用の SQLj アプリケーション・オプション』

関連サンプル:

- ・ 『bldsqljs -- Builds Java embedded SQL (SQLJ) stored procedures on UNIX』

JDBC アプレットの作成

App1t は、DB2 データベースにアクセスする動的 SQL Java アプレットの例を示します。

手順:

すでに使用すべきでなくなったタイプ 3 ドライバー (net ドライバーとも呼びます) またはタイプ 4 ドライバーのどちらを使用してもかまいません。この 2 種類のドライバーでの接続については、この後の項に説明されています。アプレットをタイプ 4 ドライバーに移行することを強くお勧めします。

JDBC アプレット App1t を作成して、コマンド行にコマンドを入力して実行する場合、DB2 マシン (サーバーまたはクライアント) に Web サーバーがインストール済みで実

行中であることを確認するか、または以下のコマンドをクライアント・マシンの作業ディレクトリーで入力して、Java Development Kit に付属の アプレット・ビューアーを使用します。

```
appletviewer Applt.html
```

タイプ 3 (net) ドライバーとの接続

タイプ 3 ドライバーに接続するには、まず Applt.html ファイルを、ファイル内の指示にしたがって修正します。次に、Applt.html に指定されている TCP/IP ポートで JDBC アプレット・サーバーを開始します。たとえば Applt.html では、param name=port value='6789' と指定してから以下のように入力できます。

```
db2jstrt 6789
```

接続ストリング内の JDBC ポート番号は、推奨デフォルトの "6789" にしてください。これは、番号が他のポート番号と競合しないと断定できる場合にだけ変更してください。データベースのポート番号 "50000" は使用しないでください。

タイプ 4 ドライバーとの接続

タイプ 4 ドライバーに接続するには、Applt.html ファイルを、ファイル内の指示にしたがって修正します。TCP/IP ポート番号には、データベース・ポート番号 "50000" を使用できます。

アプレットの作成

1. 以下に示すコマンドで Applt.java をコンパイルし、ファイル Applt.class を生成します。

```
javac Applt.java
```

2. Web ブラウザーが、作業ディレクトリーにアクセス可能であることを確認してください。アクセス可能ではない場合は、Applt.class と Applt.html をアクセス可能なディレクトリーにコピーします。
3. タイプ 3 ドライバーを使用する場合、Windows では sqllib¥java¥db2java.zip を、UNIX では sqllib/java/db2java.zip を、Applt.class および Applt.html と同じディレクトリーにコピーします。

タイプ 4 ドライバーを使用する場合、Windows では sqllib¥java¥db2jcc.jar を、UNIX では sqllib/java/db2jcc.jar を、Applt.class および Applt.html と同じディレクトリーにコピーします。

4. クライアント・マシンで、Web ブラウザー (Java 1.3 をサポートしていなければなりません) を開始し、Applt.html をロードします。

このプログラムは、Java の makefile を使用して作成することもできます。

関連概念:

- 772 ページの『Java アプレットに関する考慮事項』

関連タスク:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『JDBC アプリケーションの構築』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『JDBC ルーチンの作成』
- 778 ページの『SQLj アプレットの作成』

関連資料:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『JDBC のサンプル』

関連サンプル:

- 『Applt.java -- A Java applet that use JDBC applet driver to access a database (JDBC)』

SQLj アプレットの作成

以下のステップでは、DB2 データベースにアクセスする SQLj アプレットの例を示す Applt サンプルの作成方法が示されています。これらのステップでは、ビルド・ファイル bldsqlj (UNIX)、または bldsqlj.bat (Windows) が使用されていますが、これには、SQLj アプレットまたはアプリケーションを作成するためのコマンドが入っていません。

ビルド・ファイルは、UNIX では \$1、\$2、\$3、\$4、\$5、および \$6、Windows では %1、%2、%3、%4、%5、および %6 の、最大 6 個のパラメーターをとります。最初のパラメーターは、プログラムの名前を指定します。2 番目のパラメーターはデータベース・インスタンスのユーザー ID を、3 番目のパラメーターはパスワードを指定します。4 番目のパラメーターは、サーバー名を指定します。5 番目のパラメーターはポート番号を指定します。そして 6 番目のパラメーターはデータベース名を指定します。最初のパラメーター (プログラム名) 以外のどのパラメーターでも、デフォルト値を使用してもかまいません。デフォルトのパラメーター値の使用に関する詳細は、ビルド・ファイルを参照してください。

手順:

すでに使用すべきでなくなったタイプ 3 ドライバー (net ドライバーとも呼びます) またはタイプ 4 ドライバーのどちらを使用してもかまいません。この 2 種類のドライバーでの接続については、この後の項に説明されています。アプレットをタイプ 4 ドライバーに移行することを強くお勧めします。

このアプレットを実行する場合、DB2 マシン (サーバーまたはクライアント) に Web サーバーがインストール済みで実行中であることを確認するか、または以下のコマンド

をクライアント・マシンの作業ディレクトリーで入力して、Java Development Kit に付属のアプレット・ビューアーを使用することができます。

```
appletviewer Applt.html
```

タイプ 3 (net) ドライバーとの接続

タイプ 3 ドライバーに接続するには、まず Applt.html ファイルを、ファイル内の指示にしたがって修正します。次に、Applt.html に指定されている TCP/IP ポートで JDBC アプレット・サーバーを開始します。たとえば Applt.html では、param name=port value='6789' と指定してから以下のように入力できます。

```
db2jstrt 6789
```

接続ストリング内の JDBC ポート番号は、推奨デフォルトの "6789" にしてください。これは、番号が他のポート番号と競合しないと断定できる場合にだけ変更してください。データベースのポート番号 "50000" は使用しないでください。

タイプ 4 ドライバーとの接続

タイプ 4 ドライバーに接続するには、Applt.html ファイルを、ファイル内の指示にしたがって修正します。TCP/IP ポート番号には、データベース・ポート番号 "50000" を使用できます。

アプレットの作成

1. アプレットを以下のコマンドで作成します。

```
bldsq1j Applt <userid> <password> <server_name> <port_number> <db_name>
```

ただし、ビルド・ファイルに説明されているとおり、プログラム名以外のどのパラメーターにもデフォルト値を使用することができます。

2. Web ブラウザーからか、またはアプレット・ビューアー (使用している場合) から、作業ディレクトリーにアクセスできることを確認してください。ディレクトリーにアクセスできない場合、アクセス可能なディレクトリーに以下のファイルをコピーします。

```
Applt.html                Applt.class
Applt_Cursor1.class       Applt_Cursor2.class
Applt_SJProfileKeys.class Applt_SJProfile0.ser
```

3. タイプ 3 ドライバーを使用する場合、Windows では sqllib\java\db2jcc.jar および sqllib\java\db2java.zip を、UNIX では sqllib/java/db2jcc.jar および sqllib/java/db2java.zip を、Applt.class および Applt.html と同じディレクトリーにコピーします。

タイプ 4 ドライバーを使用する場合、Windows では sqllib¥java¥db2jcc.jar を、UNIX では sqllib/java/db2jcc.jar を、Applt.class および Applt.html と同じディレクトリーにコピーします。

4. クライアント・マシンで、Web ブラウザー (JDK 1.3 をサポートしていなければなりません) を開始し、App1t.html をロードします。

このプログラムは、Java の makefile を使用して作成することもできます。

関連概念:

- 772 ページの『Java アプレットに関する考慮事項』

関連タスク:

- 776 ページの『JDBC アプレットの作成』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『SQL アプリケーションの作成』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『SQL ルーチンの作成』

関連資料:

- 774 ページの『UNIX 用の SQLj アプリケーション・オプション』
- 783 ページの『Windows の SQLj アプリケーション・オプション』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『SQL のサンプル』

関連サンプル:

- 『Applt.sqlj -- An SQLJ applet that uses a JDBC applet driver to access a database (SQLj)』
- 『bldsqlj.bat -- Builds a Java embedded SQL (SQLJ) application or applet on Windows』
- 『bldsqlj -- Builds Java embedded SQL (SQLJ) applications and applets on UNIX』

SQL プロシージャの作成

UNIX では sqllib/samples/sqlproc ディレクトリーに、Windows では sqllib\samples\sqlproc ディレクトリーにある DB2 コマンド行プロセッサ・スクリプト (.db2 拡張子で終わるもの) は、CREATE PROCEDURE ステートメントを実行してサーバー上にストアード・プロシージャを作成します。どの CLP スクリプトにも、対応する同一名のクライアント・アプリケーション・ファイルがあります。なおその名前には、.sqc (C 組み込み SQL の場合)、.c (DB2 CLI の場合)、または .java (JDBC の場合) といった、言語とアプリケーション・インターフェースを表す拡張子が付いています。

注: CALL は DB2 バージョン 8 の SQL ステートメントです。これは、いかなる順序でも、プロシージャを作成できなくなったことを意味します。コンパイラーは、呼び出されるプロシージャがあるかどうかをコンパイル時にチェックし、プロシージャが見つからない場合は、SQLCODE -440 を返します。

手順:

CREATE PROCEDURE CLP スクリプトを実行するには、以下のコマンドでサンプル・データベースに接続してください。

```
db2 connect to sample user userid using password
```

ここで、*userid* と *password* は、sample データベースが置かれているインスタンスのユーザー ID とパスワードを表します。

resultset.db2 スクリプト・ファイルの CREATE PROCEDURE を実行するには、以下のコマンドを入力します。

```
db2 -td@ -vf resultset.db2
```

これで、SQL プロシージャを呼び出す準備ができました。

関連タスク:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『CALL ステートメントによるストアド・プロシージャの呼び出し』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『UNIX でのクライアント・アプリケーションによる SQL プロシージャの呼び出し』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『Windows でのクライアント・アプリケーションによる SQL プロシージャの呼び出し』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『コンパイル済み SQL プロシージャの配布』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『SQL プロシージャの再バインド』

関連サンプル:

- 『resultset.db2 -- To create the MEDIAN_RESULT_SET SQL procedure 』

Java クラス・ライブラリー

DB2 Universal Database は、JDBC および SQLj サポートにクラス・ライブラリーを提供します。これは、CLASSPATH に指定するか、アプレットに組み込まなければなりません。以下のものがあります。

db2jcc.jar

JDBC タイプ 4 ドライバーおよび SQLj へのランタイム・サポート。

db2java.zip

JDBC タイプ 2 および 3 ドライバー、およびサポート・クラス。ストアド・プロシージャおよび UDF サポートを含む。

sqlj.zip SQLj 変換プログラム・クラス・ファイル。

Java でサポートされている SQL データ・タイプ

JDBC 仕様のデータ・タイプ・マッピングに基づいた、各 SQL データ・タイプに等しい Java の値を以下の表に示します。 JDBC ドライバーは、アプリケーションとデータベースとの間で交換されるデータを、以下のマッピング・スキーマを使って変換します。これらのマッピングは、Java アプリケーションと PARAMETER STYLE JAVA プロシージャと UDF で使用します。

注: DB2 がサポートするどのプログラム言語にも、 DATALINK データ・タイプに対するホスト変数サポートはありません。

表 179. Java 宣言にマップされる SQL データ・タイプ

SQL 列名	Java データ・タイプ	SQL 列タイプ記述
SMALLINT (500 または 501)	short	16 ビットの符号付き整数
INTEGER (496 または 497)	int	32 ビットの符号付き整数
BIGINT ¹ (492 または 493)	long	64 ビットの符号付き整数
REAL (480 または 481)	float	単精度浮動小数点
DOUBLE (480 または 481)	double	倍精度浮動小数点
DECIMAL(<i>p,s</i>) (484 または 485)	java.math.BigDecimal	パック 10 進数
CHAR(<i>n</i>) (452 または 453)	java.lang.String	長さが <i>n</i> の固定長文字ストリング。 <i>n</i> は 1 から 254 まで
CHAR(<i>n</i>) FOR BIT DATA	バイト[]	長さが <i>n</i> の固定長文字ストリング。 <i>n</i> は 1 から 254 まで
VARCHAR(<i>n</i>) (448 または 449)	java.lang.String	可変長文字ストリング
VARCHAR(<i>n</i>) FOR BIT DATA	バイト[]	可変長文字ストリング
LONG VARCHAR (456 または 457)	java.lang.String	long 可変長文字ストリング
LONG VARCHAR FOR BIT DATA	バイト[]	long 可変長文字ストリング
BLOB(<i>n</i>) (404 または 405)	java.sql.Blob	ラージ・オブジェクト可変長 バイナリー・ストリング

表 179. Java 宣言にマップされる SQL データ・タイプ (続き)

SQL 列名	Java データ・タイプ	SQL 列タイプ記述
CLOB(<i>n</i>) (408 または 409)	java.sql.Clob	ラージ・オブジェクト可変長文字スト リング
DBCLOB(<i>n</i>) (412 または 413)	java.sql.Clob	ラージ・オブジェクト可変長 2 バイト文字 ストリング
DATE (384 または 385)	java.sql.Date	10 バイトの文字ストリング
TIME (388 または 389)	java.sql.Time	8 バイトの文字ストリング
TIMESTAMP (392 または 393)	java.sql.Timestamp	26 バイトの文字ストリング

注:

1. DB2 UDB バージョン 8.1 クライアントから DB2 UDB バージョン 7.1 (または 7.2) サーバーに接続している Java アプリケーションでは、以下のことに注意してください。BIGINT 値の検索に getObject() 方式が使用された場合、java.math.BigDecimal オブジェクトが返されます。

Windows の SQLj アプリケーション・オプション

以下の表は、Windows オペレーティング・システム上の bldsqlj.bat バッチ・ファイルで使用する SQLj 変換プログラム・オプションとプリコンパイル・オプションを示しています。これらは、SQLj ルーチン (ストアード・プロシージャとユーザー定義関数) の作成時に使用するように DB2 で推奨されているオプションです。

bldsqlj.bat の変換プログラムとプリコンパイル・オプション	
sqlj	SQLj 変換プログラム (プログラムのコンパイルも行います)。
%1.sqlj	SQLj ソース・ファイル。
db2sqljcustomize	Java プロファイル・カスタマイザーが使用する DB2。
-url	データベース接続を確立するための JDBC URL を jdbc:db2://servername:50000/sample のように指定します。
-user	ユーザー ID を指定します。
-password	パスワードを指定します。
%1_SJProfile0	プログラムのシリアル化プロファイルを指定します。

関連タスク:

- 778 ページの『SQLj アプレットの作成』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『SQLJ アプリケーションの作成』

関連資料:

- 784 ページの『Windows 用の SQLj ストアード・プロシージャ・オプション』

関連サンプル:

- 『bldsqlj.bat -- Builds a Java embedded SQL (SQLJ) application or applet on Windows』

Windows 用の SQLj ストアード・プロシージャ・オプション

以下の表は、Windows オペレーティング・システム上の bldsqljs.bat バッチ・ファイルで使用する SQLj 変換プログラム・オプションとプリコンパイル・オプションを示しています。これらは、SQLj ルーチン (ストアード・プロシージャとユーザー定義関数) の作成時に使用するように DB2 で推奨されているオプションです。

bldsqljs.bat の変換プログラムとプリコンパイル・オプション

sqlj SQLj 変換プログラム (プログラムのコンパイルも行います)。

%1.sqlj

SQLj ソース・ファイル。

db2sqljcustomize

Java プロファイル・カスタマイザーが使用する DB2。

-url データベース接続を確立するための JDBC URL を
jdbc:db2://servername:50000/sample のように指定します。

-user ユーザー ID を指定します。

-password

パスワードを指定します。

%1_SJProfile0

プログラムのシリアル化プロファイルを指定します。

関連タスク:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『SQLJ ルーチンの作成』

関連資料:

- 783 ページの『Windows の SQLj アプリケーション・オプション』

関連サンプル:

- 『bldsqljs.bat -- Builds a Java embedded SQL (SQLJ) stored procedure on Windows』

Windows C/C++ ルーチンのコンパイルとリンクのオプション

以下は、bldrtn.bat バッチ・ファイルに示されているように、Windows 上で Microsoft Visual C++ コンパイラーを使用して、C/C++ ルーチン (ストアード・プロシージャとユーザー定義関数) を作成するのにお勧めするコンパイルとリンクのオプションです。

bldrtn のコンパイルおよびリンク・オプション

コンパイル・オプション

%BLDCOMP%

コンパイラー用の変数。デフォルトは、c1 (Microsoft Visual C++ コンパイラー) です。またこれは、ic1 (32 ビット・アプリケーション用の Intel C++ コンパイラー)、または ec1 (Itanium 64 ビット・アプリケーション用の Intel C++ コンパイラー) に設定することもできます。

-Zi デバッグ情報を使用可能にします。

-Od 最適化なし。

-c コンパイルのみを実行し、リンクは実行しません。本書では、コンパイルとリンクが別個のステップであることを前提としています。

-W2 警告、エラー、重大、およびリカバリー不能エラー・メッセージを出力します。

-DWIN32

Windows オペレーティング・システムに必要なコンパイラー・オプション。

-MDd MSVCRT.LIB を使用してマルチスレッド DLL を作成します。

注: -debug オプションを使用しない場合は、代わりに -MD を使用してください。

リンク・オプション

link リンクにリンカーを使用します。

-debug デバッグ情報を組み込みます。

-out:%1.dll

.DLL ファイルを作成します。

%1.obj オブジェクト・ファイルを組み込みます。

db2api.lib

DB2 ライブラリーとリンクします。

-def:%1.def

モジュール定義ファイル。

他のコンパイラー・オプションについては、コンパイラーの資料をご覧ください。

関連タスク:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『Windows での C/C++ ルーチンの作成』

関連サンプル:

- 『bldrtn.bat -- Builds C routines (stored procedures and UDFs) on Windows』

- 『bldrtn.bat -- Builds C++ routines (stored procedures and UDFs) on Windows』

SQLj プログラムの作成

DB2 には、SQLj サンプル・プログラムを作成するためのビルド・ファイルが備えられています。アプレットとアプリケーションの場合、UNIX では bldsqlj スクリプトを使用し、Windows では bldsqlj.bat バッチ・ファイルを使用することができます。ルーチン (ストアード・プロシージャとユーザー定義関数) の場合、UNIX 上では bldsqljs スクリプトを使用し、Windows 上では bldsqljs.bat バッチ・ファイルを使用することができます。

DB2 に付属している SQLj 変換プログラムは、変換後の .java ファイルを .class ファイルにコンパイルします。そのため、ビルド・ファイルは java コンパイラーを使用しません。

注: 旧バージョンの DB2 では db2profrc コマンドは -url=jdbc:db2:dbname のフォームの URL を使用していました。ただし dbname は、ローカル・カタログされたデータベース別名です。新しい db2sqljcustomize コマンドは、DB2 のタイプ 4 JDBC ドライバーの規則 -url jdbc:db2://hostname:portnumber/dbname に準じています。ここで hostname は DB2 サーバーの名前、portnumber は DB2 サーバーの TCP/IP Listener のポート番号、そして dbname は DB2 サーバーにカタログされたデータベースの別名です。つまり、DB2 サーバーは、TCP/IP 接続用に構成する必要があるということです。

手順:

IBM Java Development Kit (UNIX および Windows オペレーティング・システム版) で SQLj プログラムを作成して実行するには、以下に示す、オペレーティング・システム別のコマンドを使用して、JDK のジャストインタイトム・コンパイラーをオフにする必要があります。

UNIX: bash または Korn シェルの場合

```
export JAVA_COMPILER=NONE
```

C シェルの場合

```
setenv JAVA_COMPILER NONE
```

Windows:

```
SET JAVA_COMPILER=NONE
```

別のタイプの DB2 SQLj プログラムを作成する場合は、以下を参照してください。

- SQLj アプレットの作成
- SQLj アプリケーションの作成SQLj アプリケーションの作成
- SQLj ルーチンの作成SQLj ルーチンの作成

関連概念:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『Java サンプル・プログラム』
- 772 ページの『Java アプレットに関する考慮事項』

関連タスク:

- 778 ページの『SQLj アプレットの作成』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『SQLJ アプリケーションの作成』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『SQLJ ルーチンの作成』

関連資料:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『SQLJ のサンプル』

HP-UX C アプリケーションのコンパイルとリンクのオプション

以下は、bldapp ビルド・スクリプトに示されているように、HP-UX C コンパイラーを使用して、C 組み込み SQL および DB2 API アプリケーションを作成するのにお勧めするコンパイルとリンクのオプションです。

bldapp のコンパイルおよびリンク・オプション	
コンパイル・オプション	
cc	C コンパイラー。
\$EXTRA_CFLAG	
	64 ビット・サポートが使用可能な場合、これには値 +DA2.0W が含まれます。 32 ビット・サポートでは、これに値は含まれません。
+DA2.0W (64 ビットのみ)	
	64 ビット・コードを生成する場合に使用する必要があります。
-Ae	HP ANSI 拡張モードを使用可能にします。
-I\$DB2PATH/include	
	DB2 組み込みファイルのロケーションを指定します。
-c	コンパイルのみを実行し、リンクは実行しません。コンパイルとリンクは別個のステップです。

bldapp のコンパイルおよびリンク・オプション

リンク・オプション

cc コンパイラーをリンカーのフロントエンドとして使用します。

\$EXTRA_CFLAG

64 ビット・サポートが使用可能な場合、これには値 **+DA2.0W** が含まれます。

32 ビット・サポートでは、これに値は含まれません。

+DA2.0W (64 ビットのみ)

64 ビット・コードを生成する場合に使用する必要があります。

-o \$1 実行可能ファイルを指定します。

\$1.o プログラム・オブジェクト・ファイルを指定します。

utilemb.o

組み込み SQL プログラムの場合に、エラー・チェックを行う組み込み SQL ユーティリティ・オブジェクト・ファイルを組み込みます。

utilapi.o

非組み込み SQL プログラムの場合に、エラー・チェックを行う DB2 API ユーティリティ・オブジェクト・ファイルを組み込みます。

-L\$DB2PATH/lib

DB2 ランタイム共有ライブラリーのロケーションを指定します。たとえば、**-L\$DB2PATH/lib**。 **-L** オプションを指定しないと、コンパイラーはパスとして **/usr/lib:/lib** を想定します。

-ldb2 DB2 ライブラリーとリンクします。

他のコンパイラー・オプションについては、コンパイラーの資料をご覧ください。

関連タスク:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『HP-UX での C アプリケーションの作成』

関連サンプル:

- 『bldapp -- Builds HP-UX C applications』

HP-UX C ルーチンのコンパイルとリンクのオプション

以下は、**bldrtn** ビルド・スクリプトに示されているように、**HP-UX C** コンパイラーを使用して、**C** ルーチン (ストアード・プロシージャとユーザー定義関数) を作成するのに勧めするコンパイルとリンクのオプションです。

bldrtn のコンパイルおよびリンク・オプション

コンパイル・オプション

cc C コンパイラー。

\$EXTRA_CFLAG

64 ビット・サポートが使用可能な場合、これには値 **+DA2.0W** が含まれます。
32 ビット・サポートでは、これに値は含まれません。

+DA2.0W (64 ビットのみ)

64 ビット・コードを生成する場合に使用する必要があります。

+u1 位置合わせしないデータ・アクセスを認めます。アプリケーションが位置合わせしないデータを使用する場合にのみ使用します。

+z 位置に依存しないコードを生成します。

-Ae HP ANSI 拡張モードを使用可能にします。

-I\$DB2PATH/include

DB2 組み込みファイルのロケーションを指定します。たとえば、
-I\$DB2PATH/include。

-D_POSIX_C_SOURCE=199506L

ルーチンを他のルーチンと同じプロセスで実行する (THREADSAFE) か、またはエンジンそのもので実行する (NOT FENCED) ときに必要な、
_REENTRANT が必ず定義されるようにするための POSIX スレッド・ライブラリー・オプション。

-c コンパイルのみを実行し、リンクは実行しません。本書では、コンパイルとリンクが別個のステップであることを前提としています。

リンク・オプション

ld リンクにリンカーを使用します。

-b 通常の実行可能ファイルではなく、共有ライブラリーを作成します。

-o \$1 出力を、共有ライブラリー・ファイルとして指定します。

\$1.o プログラム・オブジェクト・ファイルを指定します。

-L\$DB2PATH/lib

DB2 ランタイム共有ライブラリーのロケーションを指定します。たとえば、
\$HOME/sql1lib/lib。 **-L** オプションを指定しないと、コンパイラーはパスとして **/usr/lib:/lib** を想定します。

-ldb2 DB2 ライブラリーとリンクします。

-lpthread

POSIX スレッド・ライブラリーとリンクします。

他のコンパイラー・オプションについては、コンパイラーの資料をご覧ください。

関連タスク:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『HP-UX での C ルーチンの作成』

関連サンプル:

- 『bldrtn -- Builds HP-UX C routines (stored procedures and UDFs)』

HP-UX C++ アプリケーションのコンパイルとリンクのオプション

以下は、bldapp ビルド・スクリプトに示されているように、HP-UX C++ コンパイラーを使用して、C++ 組み込み SQL および DB2 API アプリケーションを作成するのに
お勧めするコンパイルとリンクのオプションです。

bldapp のコンパイルおよびリンク・オプション	
コンパイル・オプション	
aCC	HP aC++ コンパイラー。
\$EXTRA_CFLAG	
	64 ビット・サポートが使用可能な場合、これには値 +DA2.0W が含まれます。 32 ビット・サポートでは、これに値は含まれません。
+DA2.0W (64 ビットのみ)	64 ビット・コードを生成する場合に使用する必要があります。
-ext	"long long" サポートを含むさまざまな C++ 拡張子を許可します。
-I\$DB2PATH/include	DB2 組み込みファイルのロケーションを指定します。たとえば、 \$HOME/sql1lib/include。
-c	コンパイルのみを実行し、リンクは実行しません。本書では、コンパイルと リンクが別個のステップであることを前提としています。

bldapp のコンパイルおよびリンク・オプション

リンク・オプション

aCC HP aC++ コンパイラーをリンカーのフロントエンドとして使用します。

\$EXTRA_CFLAG

64 ビット・サポートが使用可能な場合、これには値 **+DA2.0W** が含まれます。

32 ビット・サポートでは、これに値は含まれません。

+DA2.0W (64 ビットのみ)

64 ビット・コードを生成する場合に使用する必要があります。

-o \$1 実行可能ファイルを指定します。

\$1.o プログラム・オブジェクト・ファイルを指定します。

utilemb.o

組み込み SQL プログラムの場合に、エラー・チェックを行う組み込み SQL ユーティリティー・オブジェクト・ファイルを組み込みます。

utilapi.o

非組み込み SQL プログラムの場合に、エラー・チェックを行う DB2 API ユーティリティー・オブジェクト・ファイルを組み込みます。

-L\$DB2PATH/lib

DB2 ランタイム共有ライブラリーのロケーションを指定します。たとえば、**\$HOME/sql1lib/lib**。 **-L** オプションを指定しないと、コンパイラーはパスとして **/usr/lib:/lib** を想定します。

-ldb2 DB2 ライブラリーとリンクします。

他のコンパイラー・オプションについては、コンパイラーの資料をご覧ください。

関連タスク:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『HP-UX での C++ アプリケーションの作成』

関連サンプル:

- 『bldapp -- Builds HP-UX C++ applications』

HP-UX C++ ルーチンのコンパイルとリンクのオプション

以下は、bldrtn ビルド・スクリプトに示されているように、HP-UX C++ コンパイラーを使用して、C++ ルーチン (ストアード・プロシージャとユーザー定義関数) を作成するのに勧めするコンパイルとリンクのオプションです。

bldrtn のコンパイルおよびリンク・オプション

コンパイル・オプション

aCC HP aC++ コンパイラー。

\$EXTRA_CFLAG

64 ビット・サポートが使用可能な場合、これには値 **+DA2.0W** が含まれます。
32 ビット・サポートでは、これに値は含まれません。

+DA2.0W (64 ビットのみ)

64 ビット・コードを生成する場合に使用する必要があります。

+u1 位置合わせしないデータ・アクセスを認めます。

+z 位置に依存しないコードを生成します。

-ext "long long" サポートを含むさまざまな C++ 拡張子を許可します。

-mt ルーチンを他のルーチンと同じプロセスで実行する (THREADSAFE) か、またはエンジンそのもので実行する (NOT FENCED) かに応じて、HP aC++ コンパイラーのマルチスレッド・サポートを使用可能にします。

-I\$DB2PATH/include

DB2 組み込みファイルのロケーションを指定します。たとえば、**\$DB2PATH/include**。

-c コンパイルのみを実行し、リンクは実行しません。本書では、コンパイルとリンクが別個のステップであることを前提としています。

リンク・オプション

aCC HP aC++ コンパイラーをリンカーのフロントエンドとして使用します。

\$EXTRA_CFLAG

64 ビット・サポートが使用可能な場合、これには値 **+DA2.0W** が含まれます。
32 ビット・サポートでは、これに値は含まれません。

+DA2.0W (64 ビットのみ)

64 ビット・コードを生成する場合に使用する必要があります。

-mt ルーチンを他のルーチンと同じプロセスで実行する (THREADSAFE) か、またはエンジンそのもので実行する (NOT FENCED) かに応じて、HP aC++ コンパイラーのマルチスレッド・サポートを使用可能にします。

-b 通常の実行可能ファイルではなく、共有ライブラリーを作成します。

-o \$1 実行可能ファイルを指定します。

\$1.o プログラム・オブジェクト・ファイルを指定します。

-L\$DB2PATH/lib

DB2 ランタイム共有ライブラリーのロケーションを指定します。たとえば、**-L\$DB2PATH/lib**。 **-L** オプションを指定しないと、コンパイラーはパスとして **/usr/lib/lib** を想定します。

-ldb2 DB2 ライブラリーとリンクします。

他のコンパイラー・オプションについては、コンパイラーの資料をご覧ください。

関連タスク:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『HP-UX での C++ ルーチンの作成』

関連サンプル:

- 『bldrtn -- Builds HP-UX C++ routines (stored procedures and UDFs)』

HP-UX Micro Focus COBOL アプリケーションのコンパイルとリンクのオプション

以下は、bldapp ビルド・スクリプトに示されているように、HP-UX 上で Micro Focus COBOL コンパイラーを使用して、COBOL 組み込み SQL および DB2 API アプリケーションを作成するのにお勧めするコンパイルとリンクのオプションです。

bldapp のコンパイルおよびリンク・オプション	
コンパイル・オプション	
cob	Micro Focus COBOL コンパイラー。
-cx	オブジェクト・モジュールにコンパイルします。
リンク・オプション	
cob	コンパイラーをリンカーのフロントエンドとして使用します。
-x	実行可能プログラムを指定します。
\$1.o	プログラムのオブジェクト・ファイルを組み込みます。
checkerr.o	エラー・チェック用のユーティリティ・オブジェクト・ファイルを組み込みます。
-L\$DB2PATH/lib	DB2 ランタイム共有ライブラリーのロケーションを指定します。
-ldb2	DB2 ライブラリーとリンクします。
-ldb2gmf	Micro Focus COBOL 用 DB2 例外ハンドラー・ライブラリーとリンクします。
他のコンパイラー・オプションについては、コンパイラーの資料をご覧ください。	

関連タスク:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『HP-UX での Micro Focus COBOL アプリケーションの作成』

関連サンプル:

- 『bldapp -- Builds HP-UX Micro Focus COBOL applications』

HP-UX Micro Focus COBOL ルーチンのコンパイルとリンクのオプション

以下は、bldrtn ビルド・スクリプトに示されているように、HP-UX 上で Micro Focus COBOL コンパイラーを使用して、COBOL ルーチン (ストアード・プロシージャー) を作成するのにお勧めするコンパイルとリンクのオプションです。

bldrtn のコンパイルおよびリンク・オプション	
コンパイル・オプション	
cob	COBOL コンパイラー。
+z	位置に依存しないコードを生成します。
-cx	オブジェクト・モジュールにコンパイルします。
リンク・オプション	
ld	リンクにリンカーを使用します。
-b	通常の実行可能ファイルではなく、共有ライブラリーを作成します。
-o \$1	実行可能ファイルを指定します。
\$1.o	プログラムのオブジェクト・ファイルを組み込みます。
-L\$DB2PATH/lib	DB2 ランタイム共有ライブラリーのロケーションを指定します。
-ldb2	DB2 共有ライブラリーにリンクします。
-ldb2gmf	Micro Focus COBOL 用 DB2 例外ハンドラー・ライブラリーとリンクします。
-L\$COBDIR/coblib	COBOL ランタイム・ライブラリーのロケーションを指定します。
-lcobol	COBOL ライブラリーにリンクします。
-lcrtm	crtm ライブラリーにリンクします。
他のコンパイラー・オプションについては、コンパイラーの資料をご覧ください。	

関連タスク:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『HP-UX での Micro Focus COBOL ルーチンの作成』

関連サンプル:

- 『bldrtn -- Builds HP-UX Micro Focus COBOL routines (stored procedures)』

CLI の紹介

DB2 コール・レベル・インターフェース (DB2 CLI) は、データベース・サーバーの DB2 ファミリーに対する IBM の呼び出し可能な SQL インターフェースです。これは、関係データベース・アクセス用の 'C' および 'C++' アプリケーション・プログラミング・インターフェースで、関数呼び出しを使用して、動的 SQL ステートメントを関数の引き数として渡します。これは組み込み動的 SQL の代替方法ですが、組み込み SQL とは違って、DB2 CLI はホスト変数またはプリコンパイラーを必要としません。

DB2 CLI は、Microsoft** オープン・データベース・コネクティビティ (Open Database Connectivity** (ODBC)) 仕様、および SQL/CLI 用国際規格 (International Standard for SQL/CLI) に基づいています。業界の標準に従う努力の一環として、これらの仕様が DB2 コール・レベル・インターフェースの基盤として採用されました。これは、上記のデータベース・インターフェースのいずれかについてすでに精通しているアプリケーション・プログラマーが短期間で学習できるようにするためです。さらに、複数の DB2 特定の拡張が追加されており、アプリケーション・プログラマーが DB2 機能を特に活用するのに役立ちます。

DB2 CLI ドライバーは、ODBC ドライバー・マネージャーによってロードされる際、ODBC ドライバーとしても働きます。これは ODBC 3.51 に準拠しています。

DB2 CLI の背景情報:

DB2 CLI または呼び出し可能 SQL インターフェースを理解するには、それが何に基づいているのかを理解し、それを既存のインターフェースと比較するとわかりやすくなります。

X/Open Company と SQL アクセス・グループは共同で、X/Open コール・レベル・インターフェース と呼ばれる呼び出し可能 SQL インターフェースの仕様を開発しました。このインターフェースの目標は、アプリケーションがいずれか 1 つのデータベース・ベンダーのプログラミング・インターフェースから独立できるようにすることによって、アプリケーションの可搬性を高めることです。X/Open コール・レベル・インターフェース仕様のほとんどは、ISO コール・レベル・インターフェース国際規格 (ISO/IEC 9075-3:1995 SQL/CLI) の一部として受け入れられています。

Microsoft® 社は、X/Open CLI の準備草案に基づいて、Microsoft オペレーティング・システム用のオープン・データベース・コネクティビティ (ODBC) と呼ばれる呼び出し可能 SQL インターフェースを開発しました。

また、ODBC 仕様には、接続要求時に与えられるデータ・ソース (データベース名) に基づいて、ドライバー・マネージャーによってデータベース特定の ODBC ドライバーが実行時に動的にロードされる操作環境が含まれています。アプリケーションは、各

DBMS のライブラリーではなく、単一のドライバー・マネージャーのライブラリーに直接リンクされます。ドライバー・マネージャーは、アプリケーションの関数呼び出しを実行時に仲介して、それが該当する DBMS 特定の ODBC ドライバーに確実に仕向けられるようにします。ODBC ドライバー・マネージャーは、ODBC 特定の関数だけを認識しているので、DBMS 特定の関数は ODBC 環境ではアクセスできません。DBMS 特定の動的 SQL ステートメントは、エスケープ文節と呼ばれるメカニズムによってサポートされます。

ODBC は、Microsoft オペレーティング・システムに限られるものではなく、他のインプリメンテーションをさまざまなプラットフォームで利用できます。

DB2 CLI ロード・ライブラリーは、ODBC ドライバーとして ODBC ドライバー・マネージャーによってロードできます。ODBC アプリケーションの開発の際には、ODBC ソフトウェア開発キットを入手してください。Windows® プラットフォームの場合、ODBC SDK は Microsoft Data Access Components (MDAC) SDK の一部として入手でき、<http://www.microsoft.com/data/> からダウンロードして使用できます。Windows 以外のプラットフォームの場合、ODBC SDK は他のベンダーによって提供されます。DB2 サーバーに接続する可能性のある ODBC アプリケーションを開発する場合、本書 (DB2 特定の拡張についての情報および診断情報) と、Microsoft 社から入手できる ODBC Programmer's Reference and SDK Guide を併用してください。

DB2 CLI に対して直接記述されたアプリケーションは、DB2 CLI ロード・ライブラリーに直接リンクします。DB2 CLI では、DB2 特定の関数はもとより、複数の ODBC および ISO SQL/CLI 関数のサポートが含まれています。

以下の DB2 機能は、ODBC と DB2 CLI の両方のアプリケーションで利用可能です。

- 2 バイトの (図形) データ・タイプ
- ストアード・プロシージャ
- 分散作業単位 (DUOW)、2 フェーズ・コミット
- コンパウンド SQL
- ユーザー定義タイプ (UDT)
- ユーザー定義関数 (UDF)

DB2 CLI の更新の詳細は、以下の DB2® アプリケーション開発の Web サイトを参照してください。

<http://www.ibm.com/software/data/db2/udb/ad>

関連概念:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『DB2 CLI と Microsoft ODBC の比較』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『DB2 コール・レベル・インターフェース (CLI) と組み込み動的 SQL の比較』

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『組み込み SQL と比較した DB2 CLI の利点』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『DB2 CLI または組み込み SQL をいつ使用するか』

DiagIdentifier 引き数 (CLI) のヘッダー・フィールドとレコード・フィールド

ヘッダー・フィールド

以下のヘッダー・フィールドを、*DiagIdentifier* 引き数に組み込むことができます。記述子フィールドに定義されている唯一の診断ヘッダー・フィールドは、SQL_DIAG_NUMBER および SQL_DIAG_RETURNCODE です。

表 180. *DiagIdentifier* 引き数のヘッダー・フィールド

SQL_DIAG_CURSOR_ROW_COUNT (戻りタイプ SQLINTEGER)

このフィールドは、カーソルにある行のカウントを含みます。そのセマンティクスは、SQLGetInfo() 情報タイプに依存しています。

- SQL_DYNAMIC_CURSOR_ATTRIBUTES2
- SQL_FORWARD_ONLY_CURSOR_ATTRIBUTES2
- SQL_KEYSET_CURSOR_ATTRIBUTES2
- SQL_STATIC_CURSOR_ATTRIBUTES2

これらは、それぞれのカーソル・タイプごとにどの行カウントが使用可能かを示しています (SQL_CA2_CRC_EXACT および SQL_CA2_CRC_APPROXIMATE ビットにおいて)。

このフィールドの内容が定義されるのは、ステートメント・ハンドルに対してのみであり、しかも SQLExecute()、SQLExecDirect()、SQLMoreResults() が呼び出された後でのみです。ステートメント・ハンドル以外のハンドルで、SQL_DIAG_CURSOR_ROW_COUNT の *DiagIdentifier* を指定して SQLGetDiagField() を呼び出すと、SQL_ERROR が返されます。

SQL_DIAG_DYNAMIC_FUNCTION (戻りタイプ CHAR *)

これは基礎となる関数が実行した SQL ステートメントを記述する文字列です (DB2 CLI がサポートしている値については、801 ページの『動的関数フィールド』を参照してください)。このフィールドの内容が定義されるのは、ステートメント・ハンドルに対してのみであり、しかも SQLExecute() または SQLExecDirect() の呼び出しの完了後のみです。このフィールドの値は、SQLExecute() または SQLExecDirect() への呼び出し前には定義されていません。

SQL_DIAG_DYNAMIC_FUNCTION_CODE (戻りタイプ SQLINTEGER)

表 180. *DiagIdentifier* 引き数のヘッダー・フィールド (続き)

これは基礎となる関数が実行した SQL ステートメントを記述する数字コードです (DB2 CLI がサポートしている値については、801 ページの『動的関数フィールド』を参照してください)。このフィールドの内容が定義されるのは、ステートメント・ハンドルに対してのみであり、しかも `SQLExecute()` または `SQLExecDirect()` の呼び出しの完了後のみです。このフィールドの値は、`SQLExecute()`、`SQLExecDirect()`、または `SQLMoreResults()` への呼び出し前には定義されていません。ステートメント・ハンドル以外のハンドルで、`SQL_DIAG_DYNAMIC_FUNCTION_CODE` の *DiagIdentifier* を指定して `SQLGetDiagField()` を呼び出すと、`SQL_ERROR` が返されます。このフィールドの値は、`SQLExecute()` または `SQLExecDirect()` への呼び出し前には定義されていません。

`SQL_DIAG_NUMBER` (戻りタイプ `SQLINTEGER`)

指定されたハンドルで使用可能な状況レコードの数です。

`SQL_DIAG_RELATIVE_COST_ESTIMATE` (戻りタイプ `SQLINTEGER`)

`SQLPrepare()` が正常に呼び出された場合、ステートメントを処理するのに必要なリソースの相対コスト見積もりが含まれます。据え置き準備が使用可能な場合、このフィールドの値は、ステートメントが実行されるまで、0 になります。

`SQL_DIAG_RETURNCODE` (戻りタイプ `RETCODE`)

指定されたハンドルに関連した、最後に実行された関数により返される戻りコード。 *Handle* でまだ何の関数も呼び出されていないければ、`SQL_DIAG_RETURNCODE` には `SQL_SUCCESS` が返されます。

`SQL_DIAG_ROW_COUNT` (戻りタイプ `SQLINTEGER`)

`SQLExecute()`、`SQLExecDirect()`、または `SQLSetPos()` により実行される、挿入、削除、または更新で影響を受ける行数。これはカーソル指定が実行された後で定義されます。このフィールドの内容は、ステートメント・ハンドルでのみ定義されます。フィールド内のデータは、`SQLRowCount()` の *RowCountPtr* 引き数に返されます。フィールド内のデータは毎回の関数呼び出し後にリセットされますが、`SQLRowCount()` で返される行カウントは、ステートメントが準備済みまたは割り当てられた状態に戻されるまでは同じ状態に保たれます。

レコード・フィールド

以下のレコード・フィールドを、*DiagIdentifier* 引き数に組み込むことができます。

表 181. *DiagIdentifier* 引き数のレコード・フィールド

SQL_DIAG_CLASS_ORIGIN (戻りタイプ CHAR *)

このレコードにある SQLSTATE 値のクラスおよびサブクラス部分を定義する文書を示すストリング。

DB2 CLI は常に SQL_DIAG_CLASS_ORIGIN に空ストリングを返します。

SQL_DIAG_COLUMN_NUMBER (戻りタイプ SQLINTEGER)

SQL_DIAG_ROW_NUMBER フィールドで、行セットまたはパラメーター・セットにある行数が有効である場合、そのフィールドには結果セット内の列番号を示す値が入ります。結果セットの列番号は常に 1 で始まります。この状況レコードがブックマーク列に関するものであれば、フィールドはゼロになる可能性があります。状況レコードが列番号に関連していない場合には、その値は SQL_NO_COLUMN_NUMBER となります。DB2 CLI がこのレコードに関連している列番号を判別できなければ、フィールド値は SQL_COLUMN_NUMBER_UNKNOWN となります。このフィールドの内容は、ステートメント・ハンドルでのみ定義されます。

SQL_DIAG_CONNECTION_NAME (戻りタイプ CHAR *)

診断レコードが関連する接続の名前を示すストリング。

DB2 CLI は常に SQL_DIAG_CONNECTION_NAME に空ストリングを返します。

SQL_DIAG_MESSAGE_TEXT (戻りタイプ CHAR *)

エラーまたは警告に関する通知メッセージ。

SQL_DIAG_NATIVE (戻りタイプ SQLINTEGER)

ドライバー / データ・ソース指定のユニーク・エラー・コード。ユニークのエラー・コードがなければ、ドライバーは 0 を返します。

SQL_DIAG_ROW_NUMBER (戻りタイプ SQLINTEGER)

表 181. *DiagIdentifier* 引き数のレコード・フィールド (続き)

このフィールドには、状況レコードに関連している、行セットにある行番号 (または、パラメーター・セットにあるパラメーター番号) が入ります。この状況レコードが行番号に関連していない場合には、フィールド値は `SQL_NO_ROW_NUMBER` となります。DB2 CLI がこのレコードに関連している行番号を判別できなければ、フィールド値は `SQL_ROW_NUMBER_UNKNOWN` となります。このフィールドの内容は、ステートメント・ハンドルでのみ定義されます。

SQL_DIAG_SERVER_NAME (戻りタイプ CHAR *)

診断レコードが関連するサーバー名を示す文字列。これは、*InfoType* に `SQL_DATA_SOURCE_NAME` を指定した `SQLGetInfo()` 呼び出しで返される値と同じです。環境ハンドルに関連する診断データ構造の場合、およびどのサーバーにも関連しない診断の場合、このフィールドはゼロ長文字列です。

SQL_DIAG_SQLSTATE (戻りタイプ CHAR *)

5 文字の SQLSTATE 診断コード。

SQL_DIAG_SUBCLASS_ORIGIN (戻りタイプ CHAR *)

`SQL_DIAG_CLASS_ORIGIN` と同じフォーマットおよび有効値の文字列。これは、SQLSTATE コードのサブクラスの定義部分を識別します。

DB2 CLI は常に `SQL_DIAG_SUBCLASS_ORIGIN` に空文字列を返します。

動的関数フィールドの値

以下の表は、`SQL_DIAG_DYNAMIC_FUNCTION` と `SQL_DIAG_DYNAMIC_FUNCTION_CODE` の値を示しています。これらは、`SQLExecute()` または `SQLExecDirect()` への呼び出しで実行される各タイプの SQL ステートメントに適用されます。これは DB2 CLI が使用するリストであり、ODBC では、その他の値も指定します。

表 182. 動的関数フィールドの値

実行される SQL ステートメント	<code>SQL_DIAG_DYNAMIC_FUNCTION</code> の値	<code>SQL_DIAG_DYNAMIC_FUNCTION_CODE</code> の値
alter-table-statement	“ALTER TABLE”	<code>SQL_DIAG_ALTER_TABLE</code>
create-index-statement	“CREATE INDEX”	<code>SQL_DIAG_CREATE_INDEX</code>
create-table-statement	“CREATE TABLE”	<code>SQL_DIAG_CREATE_TABLE</code>
create-view-statement	“CREATE VIEW”	<code>SQL_DIAG_CREATE_VIEW</code>

表 182. 動的関数フィールドの値 (続き)

実行される SQL ステートメント	SQL_DIAG_ DYNAMIC_FUNCTION の値	SQL_DIAG_DYNAMIC_ FUNCTION_CODE の値
cursor-specification	“SELECT CURSOR”	SQL_DIAG_SELECT_CURSOR
delete-statement-positioned	“DYNAMIC DELETE CURSOR”	SQL_DIAG_DYNAMIC_DELETE_ CURSOR
delete-statement-searched	“DELETE WHERE”	SQL_DIAG_DELETE_WHERE
drop-index-statement	“DROP INDEX”	SQL_DIAG_DROP_INDEX
drop-table-statement	“DROP TABLE”	SQL_DIAG_DROP_TABLE
drop-view-statement	“DROP VIEW”	SQL_DIAG_DROP_VIEW
grant-statement	“GRANT”	SQL_DIAG_GRANT
insert-statement	“INSERT”	SQL_DIAG_INSERT
ODBC-procedure-extension	“CALL”	SQL_DIAG_PROCEDURE_CALL
revoke-statement	“REVOKE”	SQL_DIAG_REVOKE
update-statement-positioned	“DYNAMIC UPDATE CURSOR”	SQL_DIAG_DYNAMIC_UPDATE_ CURSOR
update-statement-searched	“UPDATE WHERE”	SQL_DIAG_UPDATE_WHERE
不明	空ストリング	SQL_DIAG_UNKNOWN_STATEMENT

関連概念:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『CLI アプリケーションでの据え置き準備』

関連資料:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『SQLGetDiagField 関数 (CLI) - 診断データ・フィールドの取得』

接続属性 (CLI) リスト

以下の表は、いつそれぞれの CLI 接続属性を設定できるかを示しています。「ステートメントを割り当てた後」列内に「可」があれば、ステートメントの割り振りの前でも後でも接続属性を設定できることを意味します。

表 183. 接続属性をいつ設定するか

属性	接続前	接続後	ステートメントを 割り当てた後
SQL_ATTR_ACCESS_MODE	可	可	可 ^a
SQL_ATTR_ANSI_APP	可	不可	不可
SQL_ATTR_AUTO_IPD (読み取り専用)	不可	可	可

表 183. 接続属性をいつ設定するか (続き)

属性	接続前	接続後	ステートメントを割り当てた後
SQL_ATTR_AUTOCOMMIT	可	可	可 ^b
SQL_ATTR_CLISHEMA	可	可	可
SQL_ATTR_CONN_CONTEXT	可	不可	不可
SQL_ATTR_CONNECT_NODE	可	不可	不可
SQL_ATTR_CONNECTTYPE	可	不可	不可
SQL_ATTR_CURRENT_PACKAGE_SET	可	可 ^a	不可 [*]
SQL_ATTR_CURRENT_SCHEMA	可	可	可
SQL_ATTR_DB2_SQLERRP (読み取り専用)	不可	可	可
SQL_ATTR_DB2ESTIMATE	不可	可	可
SQL_ATTR_DB2EXPLAIN	不可	可	可
SQL_ATTR_ENLIST_IN_DTC	不可	可	可
SQL_ATTR_INFO_ACCTSTR	不可	可	可
SQL_ATTR_INFO_APPLNAME	不可	可	可
SQL_ATTR_INFO_USERID	不可	可	可
SQL_ATTR_INFO_WRKSTNNAME	不可	可	可
SQL_ATTR_LOGIN_TIMEOUT	可	不可	不可
SQL_ATTR_LONGDATA_COMPAT	可	可	可
SQL_ATTR_MAXCONN	可	不可	不可
SQL_ATTR_QUIET_MODE	可	可	可
SQL_ATTR_SYNC_POINT	可	不可	不可
SQL_ATTR_TXN_ISOLATION	不可	可 ^b	可 ^a
SQL_ATTR_WCHARTYPE	可	可 ^b	可 ^b

^a 結果的に割り当てられたステートメントにのみ影響を与えます。

^b 属性を設定できるのは、その接続上にオープン・トランザクションがない場合だけです。

^{*} ステートメントが割り振られた後でこの属性を設定してもエラーにはなりません、どのパッケージがどのステートメントで使用されるかの判断があいまいで、予期しない動作になる可能性があります。この属性を、ステートメントの割り振り後に設定することは、お勧めできません。

属性 ValuePtr の内容

SQL_ATTR_ACCESS_MODE (DB2 CLI v2)

以下のいずれかである 32 ビット整数値。

- SQL_MODE_READ_ONLY: アプリケーションは、この時点からデータに関する更新を行わないことを示します。したがって、非コミット読み取り (SQL_TXN_READ_UNCOMMITTED) といった、制限の少ない分離レベルおよびロックをトランザクションで使えるようになります。

DB2 CLI は、データベースに対する要求が読取専用であることを確認しません。更新要求を出すと、DB2 CLI は SQL_MODE_READ_ONLY 設定値の結果として選択されたトランザクション分離レベルを使用して、その要求を処理します。

- **SQL_MODE_READ_WRITE:** アプリケーションは、この時点からデータに関する更新を行うことを示します。DB2 CLI は、この接続に関するデフォルト・トランザクション分離レベルを使用する状態に戻ります。

SQL_MODE_READ_WRITE がデフォルト値です。

この接続に未解決のトランザクションがあってはなりません。

SQL_ATTR_ANSI_APP (DB2 CLI v7)

アプリケーションを ANSI またはユニコード・アプリケーションとして識別する 32 ビットの符号なし整数。この属性の値は以下のうちの 1 つとなります。

- **SQL_AA_TRUE:** アプリケーションは ANSI アプリケーションです。すべての文字データは、ANSI バージョンの CLI/ODBC 関数を使用して、ネイティブ・アプリケーション (クライアント) のコード・ページでアプリケーションとやり取りされます。
- **SQL_AA_FALSE:** アプリケーションはユニコード・アプリケーションです。ユニコード (W) バージョンの CLI/ODBC 関数が呼び出されると、すべての文字データはユニコードでアプリケーションとやり取りされます。

SQL_ATTR_AUTO_IPD (DB2 CLI v5)

SQLPrepare() 呼び出しの後で IPD の自動移植をサポートするかどうかを指定する 32 ビットの符号なしの読み取り専用整数値。

- **SQL_TRUE = SQLPrepare() 呼び出し後の IPD の自動移植がサーバーによりサポートされています。**
- **SQL_FALSE = SQLPrepare() 呼び出し後の IPD の自動移植はサーバーによりサポートされていません。準備状態のステートメントをサポートしないサーバーでは、IPD を自動的に移植することはできません。**

SQL_ATTR_AUTO_IPD 接続属性に SQL_TRUE が戻される場合は、接続属性 SQL_ATTR_ENABLE_AUTO_IPD を設定して、IPD 自動移植のオン / オフを切り換えることができます。SQL_ATTR_AUTO_IPD が SQL_FALSE の場合、SQL_ATTR_ENABLE_AUTO_IPD を SQL_TRUE に設定することはできません。

SQL_ATTR_ENABLE_AUTO_IPD のデフォルト値は、SQL_ATTR_AUTO_IPD の値と等しくなります。

この接続属性は、SQLGetConnectAttr() によって戻せませんが、SQLSetConnectAttr() で設定することはできません。

SQL_ATTR_AUTOCOMMIT (DB2 CLI v2)

自動コミット・モードまたは手動コミット・モードのどちらを使用するかを指定する 32 ビットの符号なしの整数値。

- **SQL_AUTOCOMMIT_OFF:** アプリケーションは、SQLEndTran() 呼び出しでトランザクションを手動で明示的にコミットまたはロールバックします。

- **SQL_AUTOCOMMIT_ON**: デフォルトでは、DB2 CLI は自動コミット・モードで動作します。各ステートメントは、暗黙コミットされます。照会ではないそれぞれのステートメントは、実行されるとすぐコミットされます。各照会は、関連付けられているカーソルがクローズするとすぐにコミットされます。

SQL_AUTOCOMMIT_ON はデフォルト値です。

注: この値が調整された分散作業接続単位の場合、デフォルト値は **SQL_AUTOCOMMIT_OFF** になります。

多くの DB2 環境では、SQL ステートメントの実行およびコミットは別個にデータベース・サーバーへ流される場合があるので、自動コミットは費用がかかることがあります。アプリケーション開発者が自動コミット・モードを選択するときに、このことを考慮に入れることをお勧めします。

注: 手動のコミット・モードから自動コミット・モードへ変更すると、接続上のオープン・トランザクションをコミットします。

SQL_ATTR_CLISHEMA (DB2 CLI v6)

使用するホスト DBMS 上に保管されている DB2 ODBC カタログ・ビューの名前が入っている NULL 終了文字ストリングを指すポインター。

DB2 Connect を介してホスト DBMS に接続する ODBC アプリケーションでの表リストのスキーマ呼び出しのパフォーマンスが向上するよう、DB2 ODBC カタログは設計されています。

DB2 ODBC カタログは、ホスト DBMS 上で作成および保守されます。これには、実際の DB2 カタログに定義されているオブジェクトを表す行が入っていますが、各行に入っているのは、ODBC 操作をサポートするのに必要な列だけです。DB2 ODBC カタログ内の表は事前に結合されていて、ODBC アプリケーションへの高速なアクセスをサポートするように特に索引付けされています。

システム管理者は、複数の DB2 ODBC カタログ・ビューを作成し、そのおのおのに個々のユーザー・グループが必要とする行だけを取り込むことができます。各エンド・ユーザーは、自分が使用する DB2 ODBC カタログを (この属性を設定することによって) 選択できます。

この属性には SYSSHEMA キーワードと似た効果がありますが、可能な場合には SQL_ATTR_CLISHEMA の方を使用するようにしてください。

SQL_ATTR_CLISHEMA を使用すると、データ・アクセスの効率が向上しますが、SYSSHEMA で使用するユーザー定義の表は DB2 カタログ表のミラー・イメージであり、ODBC スキーマ関数に必要な情報を作成するためには、DB2 CLI ドライバーはやはり複数の表の行を結合する必要があります。また、SQL_ATTR_CLISHEMA を使用すると、カタログ表の競合も減少します。

DB2 CLI/ODBC ドライバーの構成キーワード CLISCHEMA を使うと、この属性のデフォルト値を指定することができます。

SQL_ATTR_CONN_CONTEXT (DB2 CLI v5)

接続でどのコンテキストを使用するかを示します。SQLPOINTER は以下のどちらかになります。

- コンテキストを設定する、有効なコンテキスト (sqlBeginCtx() DB2 API によって割り当てられているもの)。
- コンテキストをリセットする NULL ポインター。

この属性を使用できるのは、アプリケーションが DB2 コンテキスト API を使ってマルチスレッド・アプリケーションを管理している場合だけです。デフォルトでは、DB2 CLI は接続ハンドルごとに 1 つのコンテキストを割り当て、実行スレッドが確実に正しいコンテキストにアタッチされるようにすることにより、コンテキストを管理します。

コンテキストの詳細については、sqlBeginCtx() API を参照してください。

SQL_ATTR_CONNECT_NODE (DB2 CLI v6)

接続先の DB2 Enterprise Server Edition データベース・パーティション・サーバーのターゲット論理ノードを指定する 32 ビットの整数。この設定は、環境変数 DB2NODE の値をオーバーライドします。以下の値に設定できます。

- 0 ~ 999 の整数
- SQL_CONN_CATALOG_NODE

この変数を指定しない場合、ターゲット論理ノードはデフォルトとして、マシン上のポート 0 に定義された論理ノードに設定されます。

これに対応する CONNECTNODE という DB2 CLI/ODBC 構成キーワードもあります。

SQL_ATTR_CONNECTION_DEAD (DB2 CLI v6)

接続が依然としてアクティブであるかどうかを示す、読み取り専用 32 ビット整数値。DB2 CLI は、以下のいずれかの値を戻します。

- SQL_CD_FALSE - 接続はアクティブのままです。
- SQL_CD_TRUE - エラーが起きたので、サーバーへの接続が終了されました。この場合でもアプリケーションは切断を実行して、すべての DB2 CLI リソースの終結処理を行う必要があります。

この属性が主に使用されるのは、接続をプールする前の Microsoft ODBC ドライバー・マネージャー 3.5x においてです。

SQL_ATTR_CONNECTION_TIMEOUT (DB2 CLI v5)

この接続属性は ODBC により定義されますが、DB2 CLI ではサポートされません。その結果、この属性を設定または取得しようとする、SQLSTATE が HYC00 (ドライバーが機能しない) になります。

アプリケーションに戻る前に、要求が完了するまで待機する秒数に対応する 32 ビット整数値。

DB2 CLI は、常に *ValuePtr* が 0 (デフォルト値) に設定されたかのように動作します。タイムアウトはありません。

SQL_ATTR_CONNECTTYPE (DB2 CLI v2)

このアプリケーションを整合分散環境で実行するか、または非整合分散環境で実行するかを指定する 32 ビット整数値。処理を調整する必要がある場合、SQL_ATTR_SYNC_POINT 接続オプションとの組み合わせにおいて、このオプションを考慮に入れる必要があります。以下の値を指定することができます。

- **SQL_CONCURRENT_TRANS:** アプリケーションを使用して、1 つのデータベースまたは複数のデータベースへの並行複数接続を行うことができます。各接続には、それぞれのコミット範囲があります。トランザクションの調整の実施は試みられません。あるアプリケーションが `SQLEndTran()` 上の環境ハンドルを使用してコミットを発行したが、すべての接続コミットが成功したわけではない場合、そのアプリケーションはリカバリーを行う必要があります。

SQL_ATTR_SYNC_POINT オプションの現行設定は無視されます。

これがデフォルトです。

- **SQL_COORDINATED_TRANS:** アプリケーションはコミットを行い、複数のデータベース接続で調整をロールバックします。このオプション設定は、組み込み SQL のタイプ 2 CONNECT の指定に対応しており、この設定を SQL_ATTR_SYNC_POINT 接続オプションと合わせて考慮する必要があります。上記の SQL_CONCURRENT_TRANS 設定とは対照的に、アプリケーションは 1 つのデータベースにつき 1 つのオープン接続のみを許可されます。

注: この接続タイプでは、SQL_ATTR_AUTOCOMMIT 接続オプションのデフォルト値である SQL_AUTOCOMMIT_OFF の設定になります。

このオプションは、接続要求を行う前に設定する必要があります。そうしない場合は、`SQLSetConnectAttr()` 呼び出しは拒否されます。

アプリケーション内のすべての接続の SQL_ATTR_CONNECTTYPE 値と SQL_ATTR_SYNC_POINT 値は、同じ値でなければなりません。最初の接続で、以後の接続のために受け入れ可能な属性を判別します。アプリケーションが SQL_ATTR_CONNECTTYPE 属性を設定するときに、接続レベルではなく、環境レベルで設定することをお勧めします。ドライバー・ユニークの環境属性の設定のための `SQLSetEnvAttr()` の呼び出しは ODBC でサポートされて

いないので、調整 DB2 トランザクションを利用するために書かれた ODBC アプリケーションは、各接続ごとに接続レベルでこれらの属性を設定する必要があります。

CONNECTTYPE DB2 CLI/ODBC 構成キーワードを使用して、このデフォルト接続タイプを設定することもできます。

注: これは、IBM 定義の拡張機能です。

SQL_ATTR_CURRENT_CATALOG (ODBC 2.0)

この接続属性は ODBC により定義されますが、DB2 CLI ではサポートされません。その結果、この属性を設定または取得しようとすると、SQLSTATE が HYC00 (ドライバが機能しない) になります。

データ・ソースで使用するカタログの名前が入っている NULL 終了文字ストリング。

SQL_ATTR_CURRENT_PACKAGE_SET (DB2 CLI v5)

後続の SQL ステートメント用のパッケージの選択に使用されるスキーマ名 (集合 ID) を示す NULL 終了文字ストリング。この属性を設定すると、"SET CURRENT PACKAGESET" SQL ステートメントが実行されます。この属性が接続以前に設定された場合、接続時に "SET CURRENT PACKAGESET" SQL ステートメントが実行されます。

CLI/ODBC アプリケーションは動的 SQL ステートメントを実行します。この接続属性を使用すると、これらのステートメントの実行に使用される特権をコントロールできます。

- CLI/ODBC アプリケーションから SQL ステートメントを実行するときに使用するスキーマを選択します。
- スキーマ内のオブジェクトに必要な特権があることを確認し、それからそれにしたがって再バインドします。これは特に、COLLECTION <collid> オプションを使用して、CLI パッケージ (sqllib/bnd/db2cli.lst) をバインドすることを意味します。詳細については、BIND コマンドを参照してください。
- CURRENTPACKAGESET オプションをこのスキーマに設定します。

これで CLI/ODBC アプリケーションからの SQL ステートメントが、指定したスキーマの下で実行され、そこに定義されている特権を使用します。

CLI/ODBC 構成キーワード CURRENTPACKAGESET の設定は、スキーマ名の指定に代わる、代替の方法です。

SQL_ATTR_CURRENT_SCHEMA (DB2 CLI v2)

注: これは、IBM 定義の拡張機能です。

szSchemaName ポインタを NULL に設定する場合に、`SQLColumns()` 呼び出しで DB2 CLI が使用するスキーマの名前の入った NULL 終了文字ストリング。

このオプションをリセットするには、*ValuePtr* 引き数で長さゼロのストリングまたは NULL ポインタを使ってこのオプションを指定します。

このオプションは、アプリケーション開発者が、`SQLColumns()` への一般呼び出しを以下のようにコーディングしてある場合に便利です。つまり、スキーマ名で結果セットを制限せず、結果セットをコード内で孤立させて制約をかける必要がある場合です。

このオプションはいつでも設定することができますが、*szSchemaName* ポインタが NULL である次の `SQLColumns()` 呼び出しで有効になります。

SQL_ATTR_DB2_SQLERRP (DB2 CLI v6)

sqlca の *sqlerrp* フィールドを収めた NULL 終了ストリングを指す *sqlpointer*。

製品を示す 3 文字の ID と、その後に製品のバージョン、リリース、修正レベルを示す 5 桁の数字が続きます。たとえば SQL08010 は DB2 Universal Database バージョン 8 リリース 1 修正レベル 0 を意味します。

SQLCODE がエラー条件を示している場合は、このフィールドはエラーを戻したモジュールを識別します。

接続が正しく完了したときにも、このフィールドが使用されます。

SQL_ATTR_DB2ESTIMATE (DB2 CLI v2)

DB2 CLI が、最適化プログラムから戻された見積もりを報告するため、SQL 照会の準備の終わりにダイアログ・ウィンドウを表示するかどうかを指定する 32 ビット整数。

- **0:** 見積もりは戻されません。

これがデフォルトです。

- 非常に大きい正の整数: DB2 CLI がウィンドウをポップアップして見積もりを報告するしきい値。この正の整数値は、`PREPARE` と関連付けられている *SQLCA* の `SQLERRD(4)` フィールドに対して比較されます。この値が `DB2ESTIMATE` より大きい場合、見積ウィンドウが表示されます。

グラフィカル・ウィンドウには、最適化プログラムの評価と、この照会を続行するか取り消すかを選択するためのボタンが表示されます。

このオプションの推奨値は 60000 です。

このオプションは、`SQL_ATTR_QUIET_MODE` との組み合わせで使用され、グラフィカル・ユーザー・インターフェースのあるアプリケーションのみ適用されます。アプリケーションは、このオプションを使用しないで、`SQLPrepare()`

の後に `SQLGetSQLCA()` を呼び出して照会を出し、次いで適切な情報を表示することによってこの機能を直接インプリメントすれば、より統合化された包括的なインターフェースを実現できます。

`SQL_ATTR_DB2ESTIMATE` 設定は、この接続の次のステートメント準備で有効になります。

注: これは、IBM 定義の拡張機能です。

SQL_ATTR_DB2EXPLAIN (DB2 CLI v2)

サーバーで Explain スナップショットまたは Explain モード情報 (あるいは両方) を生成するかどうかを指定する 32 ビット整数。指定できる値は以下のとおりです。

- `SQL_DB2EXPLAIN_OFF`: Explain スナップショット機能も Explain 表オプション機能も使用不可になっています (`SET CURRENT EXPLAIN SNAPSHOT=NO` と `SET CURRENT EXPLAIN MODE=NO` がサーバーに送信されます)。
- `SQL_DB2EXPLAIN_SNAPSHOT_ON`: Explain スナップショット機能が使用可能、 Explain 表オプション機能が使用不可になっています (`SET CURRENT EXPLAIN SNAPSHOT=YES` と `SET CURRENT EXPLAIN MODE=NO` がサーバーに送信されます)。
- `SQL_DB2EXPLAIN_MODE_ON`: Explain スナップショット機能が使用不可、 Explain 表オプション機能が使用可能になっています (`SET CURRENT EXPLAIN SNAPSHOT=NO` と `SET CURRENT EXPLAIN MODE=YES` がサーバーに送信されます)。
- `SQL_DB2EXPLAIN_SNAPSHOT_MODE_ON`: Explain スナップショット機能も Explain 表オプション機能も使用可能になっています (`SET CURRENT EXPLAIN SNAPSHOT=YES` と `SET CURRENT EXPLAIN MODE=YES` がサーバーに送信されます)。

Explain 情報を生成する前に、Explain 表を作成する必要があります。

このステートメントはトランザクション・コントロールの影響を受けず、`ROLLBACK` の影響も受けません。新規の `SQL_ATTR_DB2EXPLAIN` 設定は、この接続の次のステートメント準備で有効になります。

現行の許可 ID は、Explain 表に対して `INSERT` 特権を持っていないければなりません。

`DB2EXPLAIN DB2 CLI/ODBC` 構成キーワードを使用してデフォルト値を設定することもできます。

注: これは、IBM 定義の拡張機能です。

SQL_ATTR_ENLIST_IN_DTC (DB2 CLI v5.2)

以下のいずれかの `SQLPOINTER`。

- 非 NULL トランザクション・ポインター:

アプリケーションは、接続の状態を非分散トランザクション状態から分散トランザクション状態に変更するよう、DB2 CLI/ODBC ドライバーに求めています。接続の参加は、分散トランザクション・コーディネーター (DTC) によって行われます。

- NULL:

アプリケーションは、接続の状態を分散トランザクション状態から非分散トランザクション状態に変更するよう、DB2 CLI/ODBC ドライバーに求めています。

この属性は、Microsoft Transaction Server (MTS) との接続の参加、または参加の取り止めを行うために、MTS の環境でのみ使用されます。

この属性に非 NULL トランザクション・ポインターを指定して使用すると、直前のトランザクションは終了して、新しいトランザクションが開始されたものと想定されます。非 NULL ポインターを指定してこの API を呼び出す前に、アプリケーションは ITransaction メンバー関数 Endtransaction を呼び出す必要があります。そうしないと、直前のトランザクションが打ち切られてしまいます。アプリケーションは、同じトランザクション・ポインターを使用して複数の接続に参加させることができます。

注: この接続属性は、トランザクションごとに MTS が自動的に指定するものであって、ユーザー・アプリケーションがコーディングするものではありません。

CLI/ODBC アプリケーションは、同一のトランザクションに参加する 2 つの異なる接続上で、同一のデータベースに対して複数の SQL ステートメントを並列に実行することはできません。

SQL_ATTR_INFO_ACCTSTR (DB2 CLI v6)

DB2 Connect の使用時に、ホスト・データベース・サーバーに送信されるクライアント会計情報ストリングを識別するのに使用される、NULL 文字で終了する文字ストリングを指すポインター。以下の点に注意してください。

- 値の設定中、サーバーによっては、指定した長さ全体を処理せず、値を切り捨てる場合があります。
- DB2 for OS/390 サーバーがサポートするのは、最大 200 文字までの長さです。
- ホスト・システムへの送信時にデータが正確に変換されるようにするには、A ~ Z、0 ~ 9、および下線 (_) またはピリオド (.) の文字だけを使用するようにしてください。

注: これは、IBM 定義の拡張機能です。

SQL_ATTR_INFO_APPLNAME (DB2 CLI v6)

DB2 Connect の使用時に、ホスト・データベース・サーバーに送信されるクライアント・アプリケーション名を識別するのに使用される、 NULL 終了文字ストリングを指すポインター。以下の点に注意してください。

- 値の設定中、サーバーによっては、指定した長さ全体を処理せず、値を切り捨てる場合があります。
- DB2 for OS/390 サーバーがサポートするのは、最大 32 文字までの長さです。
- ホスト・システムへの送信時にデータが正確に変換されるようにするには、A ~ Z、0 ~ 9、および下線 (_) またはピリオド (.) の文字だけを使用するようにしてください。

注: これは、IBM 定義の拡張機能です。

SQL_ATTR_INFO_USERID (DB2 CLI v6)

DB2 Connect の使用時に、ホスト・データベース・サーバーに送信されるクライアント・ユーザー ID を識別するのに使用される、 NULL 終了文字ストリングを指すポインター。以下の点に注意してください。

- 値の設定中、サーバーによっては、指定した長さ全体を処理せず、値を切り捨てる場合があります。
- DB2 for OS/390 サーバーがサポートするのは、最大 16 文字までの長さです。
- このユーザー ID を認証ユーザー ID と混同しないでください。このユーザー ID は識別のためだけに使用され、許可にはまったく使用されません。
- ホスト・システムへの送信時にデータが正確に変換されるようにするには、A ~ Z、0 ~ 9、および下線 (_) またはピリオド (.) の文字だけを使用するようにしてください。

注: これは、IBM 定義の拡張機能です。

SQL_ATTR_INFO_WRKSTNNAME (DB2 CLI v6)

DB2 Connect の使用時に、ホスト・データベース・サーバーに送信されるクライアント・ワークステーション名を識別するのに使用される、 NULL 終了文字ストリングを指すポインター。以下の点に注意してください。

- 値の設定中、サーバーによっては、指定した長さ全体を処理せず、値を切り捨てる場合があります。
- DB2 for OS/390 サーバーがサポートするのは、最大 18 文字までの長さです。
- ホスト・システムへの送信時にデータが正確に変換されるようにするには、A ~ Z、0 ~ 9、および下線 (_) またはピリオド (.) の文字だけを使用するようにしてください。

注: これは、IBM 定義の拡張機能です。

SQL_ATTR_LOGIN_TIMEOUT (DB2 CLI v2)

この接続属性は ODBC で定義されていますが、DB2 CLI ではサポートされません。デフォルト値以外の値でこの属性を設定しようとすると、01S02 の SQLSTATE (オプション値が変更されました) が出力されます。

ログイン要求の完了を待機し始めてからアプリケーションに戻るまでの秒数に対応する 32 ビット整数値。ValuePtr 引き数で許可されている値は 0 だけです。つまり、接続を試行すると、接続が確立されるか、基礎となる通信層がタイムアウトになるまで待機します。

SQL_ATTR_LONGDATA_COMPAT (DB2 CLI v2)

既存のアプリケーションでシームレスにラージ・オブジェクト・データ・タイプにアクセスできるようにするため、文字データ・タイプ、2 バイト文字データ・タイプ、およびバイナリー・ラージ・オブジェクト・データ・タイプを、それぞれ SQL_LONGVARCHAR、SQL_LONGVARGRAPHIC、または SQL_LONGBINARY として報告するかどうかを指定する 32 ビット整数。オプションの値は、以下のとおりです。

- **SQL_LD_COMPAT_NO:** ラージ・オブジェクト・データ・タイプは、それぞれの IBM 定義のタイプ (SQL_BLOB、SQL_CLOB、SQL_DBCLOB) で報告されます。これがデフォルトです。
- **SQL_LD_COMPAT_YES:** IBM ラージ・オブジェクト・データ・タイプ (SQL_BLOB、SQL_CLOB、および SQL_DBCLOB) は、SQL_LONGVARIABLE、SQL_LONGVARCHAR、および SQL_LONGVARGRAPHIC にマップされます。SQLGetTypeInfo() は、SQL_LONGVARIABLE SQL_LONGVARCHAR、および SQL_LONGVARGRAPHIC にそれぞれ 1 つの項目を戻します。

注: これは、IBM 定義の拡張機能です。

SQL_ATTR_MAXCONN (DB2 CLI v2)

アプリケーションが設定する同時接続の最大数に対応する 32 ビット整数値。デフォルト値は 0 で、これは最大値がないことを意味します。アプリケーションはシステム・リソースで許可される個数の接続を設定することができます。整数値は 0 または正の数でなければなりません。

これは、アプリケーション・ベースでの接続の最大数のための管理プログラムとして使用できます。

Windows NT および Windows 2000 では、NetBIOS プロトコルを使用している場合、この値は、アプリケーションによって並行してセットアップされる接続 (NetBIOS セッション) の数に対応します。0 (デフォルト値) を指定すると、5 つの予約済み接続が行われます。他のアプリケーションは、予約済み NetBIOS セッションを使用できません。このパラメーターで指定された接続数

は、DB2 NetBIOS プロトコルがリモート・サーバーに接続するのに使用するアダプターに適用されます (アダプター番号は NetBIOS ノードのノード・ディレクトリーで指定します)。

最初の接続が確立される時点で有効な値は、使用しようとしている値です。最初の接続がすでに確立されていると、この値を変更しようとしても拒否されず。アプリケーションが `SQL_ATTR_MAXCONN` を設定するときに、接続レベルではなく環境レベルで設定することをお勧めしてきました。ドライバー・ユニークの環境属性の設定のための `SQLSetEnvAttr()` の呼び出しは ODBC でサポートされていないので、ODBC アプリケーションは、接続レベルでその属性を設定する必要があります。

`MAXCONN DB2 CLI/ODBC` 構成キーワードを使用して、同時接続の最大数を設定することもできます。

注: これは、IBM 定義の拡張機能です。

SQL_ATTR_METADATA_ID (ODBC 3.0)

この接続属性は ODBC により定義されますが、DB2 CLI ではサポートされません。その結果、この属性を設定または取得しようとすると、`SQLSTATE` が `HYC00` (ドライバーが機能しない) になります。

カタログ関数のストリング引き数を処理する方法を決定する `SQLINTEGER` 値。

SQL_ATTR_ODBC_CURSORS (ODBC 2.0)

この接続属性は ODBC により定義されますが、DB2 CLI ではサポートされません。その結果、この属性を設定または取得しようとすると、`SQLSTATE` が `HYC00` (ドライバーが機能しない) になります。

ドライバー・マネージャーが ODBC カーソル・ライブラリーを使用する方法を指定する 32 ビット・オプション。

SQL_ATTR_PACKET_SIZE (ODBC 2.0)

この接続属性は ODBC により定義されますが、DB2 CLI ではサポートされません。その結果、この属性を設定または取得しようとすると、`SQLSTATE` が `HYC00` (ドライバーが機能しない) になります。

ネットワーク・パケット・サイズをバイト単位で指定する 32 ビット整数値。

SQL_ATTR_QUIET_MODE (DB2 CLI v2)

32 ビット・プラットフォーム特定のウィンドウ・ハンドル。

アプリケーションがこのオプションを指定して `SQLSetConnectAttr()` を呼び出したことがない場合、DB2 CLI はこのオプションに、`SQLGetConnectAttr()` に対する `NULL` の親ウィンドウ・ハンドルを戻し、`NULL` の親ウィンドウ・ハンドルを使用してダイアログ・ボックスを表示します。たとえば、エンド・ユーザーが (DB2 CLI 初期設定ファイルの項目を介して) 最適化プログラム情

報を表示するよう求めると、DB2 CLI は NULL のウィンドウ・ハンドルを使用してこの情報が入っているダイアログ・ボックスを表示します。(プラットフォームによっては、ダイアログ・ボックスが画面の中央に表示されます。)

`ValuePtr` を NULL に設定すると、DB2 CLI はダイアログ・ボックスを表示しません。エンド・ユーザーが最適化プログラム見積もりを表示するよう求める上記の例では、アプリケーションが明示的にそのようなダイアログ・ボックスをすべて抑止するので、DB2 CLI はこれらの見積もりを表示しません。

`ValuePtr` が NULL ではない場合、アプリケーションの親ウィンドウ・ハンドルになるはずですが、DB2 CLI は、このハンドルを使ってダイアログ・ボックスを表示します。(プラットフォームによっては、ダイアログ・ボックスがアプリケーションのアクティブ・ウィンドウとの関係において、中央に表示されます。)

注: この接続オプションを、`SQLDriverConnect()` ダイアログ・ボックスを隠すために使用することはできません。(隠すためには、`fDriverCompletion` 引き数を `SQL_DRIVER_NOPROMPT` に設定します。)

SQL_ATTR_SYNC_POINT (DB2 CLI v2)

アプリケーションが 1 フェーズ調整トランザクションと 2 フェーズ調整トランザクションのどちらかを選択するのに使える 32 ビット整数値。以下の値を指定することができます。

- **SQL_ONEPHASE:** 各データベースが複数のデータベース・トランザクションで行う作業をコミットするために、1 フェーズ・コミットが使用されます。データ保全性を確かにするには、1 つのトランザクションで 2 つ以上のデータベースを更新することがないようにします。1 つのトランザクションで実行された更新のうち最初のデータベースだけが、そのトランザクションでのただ 1 つの更新側になり、それ以外のすべてのデータベースはアクセスされても読み取り専用になります。このトランザクション内でこれらの読み取り専用のデータベースを更新しようとしても拒否されます。これがデフォルトです。
- **SQL_TWOPHASE:** 複数のデータベース・トランザクションで、各データベースが行った作業をコミットする場合には、2 フェーズ・コミットが用いられます。このとき、このプロトコルをサポートする複数のデータベース間で 2 フェーズ・コミットを調整するために、トランザクション・マネージャーを使用する必要があります。1 つのトランザクション内で、複数の読み取り側および複数の更新側があっても許可されます。

アプリケーション内のすべての接続の `SQL_ATTR_CONNECTTYPE` 値と `SQL_ATTR_SYNCPOINT` 値は、同じ値でなければなりません。最初の接続で、以後の接続のために受け入れ可能な属性を判別します。アプリケーションが `SQL_ATTR_CONNECTTYPE` 属性を設定するときに、接続レベルではなく、環境レベルで設定することをお勧めします。ドライバ・ユニークの環境属性の設定のための `SQLSetEnvAttr()` の呼び出しは ODBC でサポートされて

いないので、調整 DB2 トランザクションを利用するために書かれた ODBC アプリケーションは、接続レベルでこれらの属性を設定する必要があります。

SYNCPOINT DB2 CLI/ODBC 構成キーワードを使用して、調整トランザクションのタイプを設定することもできます。

注: これは、IBM 拡張機能です。組み込み SQL では、SYNCPOINT NONE という追加の同期点設定があります。SYNCPOINT NONE は、同じデータベースへの複数の接続を許可していないので、SQL_ATTR_CONNECTTYPE オプションの SQL_CONCURRENT_TRANS 設定よりも限定された設定です。結果的に、DB2 CLI は SYNCPOINT NONE をサポートする必要はありません。

SQL_ATTR_TRACE (ODBC 1.0)

この接続属性は ODBC により定義されますが、DB2 CLI ではサポートされません。その結果、この属性を設定または取得しようとすると、SQLSTATE が HYC00 (ドライバーが機能しない) になります。

トレースを実行するかどうか DB2 CLI に指示する 32 ビット整数値。

この属性を使用する代わりに、TRACE DB2 CLI/ODBC 構成キーワードを使用して DB2 CLI トレース機能を設定することができます。

SQL_ATTR_TRACEFILE (ODBC 1.0)

この接続属性は ODBC により定義されますが、DB2 CLI ではサポートされません。その結果、この属性を設定または取得しようとすると、SQLSTATE が HYC00 (ドライバーが機能しない) になります。

トレース・ファイルの名前が入っている NULL で終了する文字ストリング。

この属性を使用する代わりに TRACEFILENAME DB2 CLI/ODBC 構成キーワードを使用して、DB2 CLI トレース・ファイル名を設定します。

SQL_ATTR_TRANSLATE_LIB (DB2 CLI v5)

この接続属性は ODBC で定義されていますが、DB2 CLI ではサポートされません。他のプラットフォームでこの属性を設定または取得しようとすると、SQLSTATE は HYC00 (ドライバーが機能しない) になります。

これは、変換 .DLL に渡される 32 ビット・フラグ値です。

DB2 Client Application Enabler for Windows または Application Development Client for Windows をインストールしたディレクトリを指示します。DB2TRANS.DLL は、コード・ページ・マッピング表が入っている DLL です。

SQL_ATTR_TRANSLATE_OPTION (ODBC 1.0)

この接続属性は ODBC で定義されていますが、DB2 CLI ではサポートされません。他のプラットフォームでこの属性を設定または取得しようとする、SQLSTATE は HYC00 (ドライバーが機能しない) になります。

DB2 バージョン 1 でのデータベースのコード・ページ番号 (データベース構成パラメーターを照会すると取得できます) を定義します。TRANSLATEDLL および TRANSLATEOPTION を指定すると、コード・ページ番号 *database codepage number* から Windows 1004 コード・ページに文字を変換できるようになります。

database codepage number でサポートされている値には、437 と 850 の 2 つがあります。それ以外の値を指定すると、変換が不可能であるということを示す警告が、接続要求時に戻されます。

注: このオプションは、Windows アプリケーションが接続するサーバーが低位バージョンのサーバー (たとえば DB2 バージョン 1) で、そのサーバーでは非同等コード・ページ変換がサポートされていない場合には便利です。

SQL_ATTR_TXN_ISOLATION (DB2 CLI v2)

ConnectionHandle で参照される現行接続のトランザクション分離レベルを設定する 32 ビットのビット・マスク。 *ValuePtr* の有効値は、実行時に *fInfoType* を SQL_TXN_ISOLATION_OPTIONS に設定して SQLGetInfo() を呼び出せば判別できます。以下の値は DB2 CLI では受け入れられますが、各サーバーではこれらの分離レベルのうちの 1 つのサブセットしかサポートできないことがあります。

- SQL_TXN_READ_UNCOMMITTED - ダーティー読み取り、反復不能読み取り、幻像読み取りが可能です。
- **SQL_TXN_READ_COMMITTED** - ダーティー読み取りが不可能です。反復不能読み取りと幻像読み取りが可能です。これがデフォルトです。
- SQL_TXN_REPEATABLE_READ - ダーティー読み取りと反復不能読み取りが不可能です。幻像読み取りが可能です。
- SQL_TXN_SERIALIZABLE - トランザクションがシリアライズ可能です。ダーティー読み取り、反復不能読み取り、幻像読み取りが不可能です。
- SQL_TXN_NOCOMMIT - 操作が正常に終了したときに、変更内容が有効にコミットされます。明示コミットやロールバックはできません。これは、自動コミットに似ています。これは SQL92 分離レベルではありませんが、DB2 UDB for AS/400 のみがサポートする IBM 定義の拡張機能です。

IBM の用語では、

- SQL_TXN_READ_UNCOMMITTED は、非コミット読み取りです。
- SQL_TXN_READ_COMMITTED は、カーソル固定です。
- SQL_TXN_REPEATABLE_READ は、読み取り固定です。
- SQL_TXN_SERIALIZABLE は、反復可能読み取りです。

このオプションは、いずれかのステートメント・ハンドル上にオープン・カーソルがある場合や、この接続に未解決のトランザクションがある場合は指定できません。それ以外の場合は、関数呼び出し時に `SQL_ERROR (SQLSTATE S1011)` が戻されます。

この属性 (または対応するキーワード) を使用できるのは、デフォルトの分離レベルが使用される場合だけです。アプリケーションが特定の分離レベルを設定する場合は、この属性を設定しても効果はありません。

注: ステートメント・ハンドルでトランザクション分離レベルの設定を許可する IBM 拡張機能があります。詳細は、`SQL_ATTR_STMTTXN_ISOLATION` ステートメント属性の項を参照してください。

SQL_ATTR_WCHARTYPE (DB2 CLI v2)

アプリケーションで使用したい `wchar_t (SQLDBCHAR)` 文字フォーマットを、2 バイトの環境で指定する 32 ビット整数。このオプションには、`wchar_t` データを複数バイト・フォーマットまたはワイド文字フォーマットのどちらにするかの選択の融通性があります。このオプションに使用可能な 2 つの値は、以下のとおりです。

- **SQL_WCHARTYPE_CONVERT:** データベースにあるグラフィック SQL データとアプリケーション変数間で、文字コードが変換されます。この変換により、アプリケーションで広幅文字ストリング (C リテラル、`'wc'` ストリング関数など) を処理する ANSI C メカニズムを最大限に活用できるようになります。データベースと通信する前に、データをマルチバイト・フォーマットに明示的に変換する必要はありません。不利な点としては、暗黙的な変換により、アプリケーションの実行時パフォーマンスが影響を受け、メモリー所要量が増える可能性があります。WCHARTYPE CONVERT を行う必要がある場合、コンパイル時に C プリプロセッサ・マクロ `SQL_WCHART_CONVERT` を定義してください。これにより、DB2 ヘッダー・ファイルにある特定の定義で、データ・タイプ `sqldbchar` の代わりに `wchar_t` を使用することになります。
- **SQL_WCHARTYPE_NOCONVERT:** アプリケーションとデータベースとの間で、暗黙的な文字コード変換は行われません。アプリケーション変数のデータは、非代替 DBCS 文字としてデータベースへ送信され、かつデータベースから受信します。この送受信の場合、アプリケーションのパフォーマンスは向上しますが、アプリケーションが `wchar_t (SQLDBCHAR)` アプリケーション変数で広幅文字データを使用しなくなったり、`wcstombs()` および `mbstowcs()` ANSI C 関数を明示的に呼び出して、データをデータベースとやりとりするときに、そのデータをマルチバイト・フォーマットに変換したり、このフォーマットから変換したりする、という不利な点があります。これがデフォルトです。

注: これは、IBM 定義の拡張機能です。

関連概念:

- *SQL* リファレンス 第 1 巻 の『分離レベル』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『Unicode 関数 (CLI)』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『CLI アプリケーションのカーソル』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『マルチスレッド CLI アプリケーション』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『Unicode CLI アプリケーション』

関連資料:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『SQLGetConnectAttr 関数 (CLI) - 現在の属性設定の取得』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『SQLGetStmtAttr 関数 (CLI) - ステートメント属性の現行設定値の取得』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『SQLSetConnectAttr 関数 (CLI) - 接続属性の設定』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『SQLSetStmtAttr 関数 (CLI) - ステートメントに関連したオプションの設定』
- コマンド・リファレンス の『BIND コマンド』
- *SQL* リファレンス 第 1 巻 の『SQLCA (SQL 連絡域)』
- 管理 API リファレンス の『sqlBeginCtx - アプリケーション・コンテキストの作成および接続』
- アプリケーション開発ガイド クライアント・アプリケーションのプログラミングの『DBCS 文字セット』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『CLI/ODBC 構成キーワード (カテゴリー別)』

記述子ヘッダーとレコード・フィールドの初期設定値 (CLI)

以下の表は、各記述子タイプごとの各フィールドの初期設定を示しています。なお“D”は、そのフィールドがデフォルト値を使って初期設定され、“ND”は、そのフィールドがデフォルト値を使わないで初期設定されることを意味します。数字が示されている場合は、その数字がフィールドのデフォルト値です。この表にはまた、フィールドが読み取り / 書き込み (R/W) であるか、読み取り専用 (R) であるかも示されています。

ヘッダー・フィールドの初期設定は以下のとおりです。

表 184. ヘッダー・フィールドの初期設定

SQL_DESC_ALLOC_TYPE (SQLSMALLINT)		
R/W:	ARD: R APD: R IRD: R IPD: R	デフォルト値: ARD: SQL_DESC_ALLOC_AUTO (明示指定の場合) または SQL_DESC_ALLOC_USER (暗黙指定の場合) APD: SQL_DESC_ALLOC_AUTO (明示指定の場合) または SQL_DESC_ALLOC_USER (暗黙指定の場合) IRD: SQL_DESC_ALLOC_AUTO IPD: SQL_DESC_ALLOC_AUTO
SQL_DESC_ARRAY_SIZE (SQLINTEGER)		
R/W:	ARD: R/W APD: R/W IRD: 未使用 IPD: 未使用	デフォルト値: ARD: ^a APD: ^a IRD: 未使用 IPD: 未使用
SQL_DESC_ARRAY_STATUS_PTR (SQLUSMALLINT *)		
R/W:	ARD: R/W APD: R/W IRD: R/W IPD: R/W	デフォルト値: ARD: NULL ポインター APD: NULL ポインター IRD: NULL ポインター IPD: NULL ポインター
SQL_DESC_BIND_OFFSET_PTR (SQLINTEGER *)		
R/W:	ARD: R/W APD: R/W IRD: 未使用 IPD: 未使用	デフォルト値: ARD: NULL ポインター APD: NULL ポインター IRD: 未使用 IPD: 未使用
SQL_DESC_BIND_TYPE (SQLINTEGER)		
R/W:	ARD: R/W APD: R/W IRD: 未使用 IPD: 未使用	デフォルト値: ARD: SQL_BIND_BY_COLUMN APD: SQL_BIND_BY_COLUMN IRD: 未使用 IPD: 未使用
SQL_DESC_COUNT (SQLSMALLINT)		
R/W:	ARD: R/W APD: R/W IRD: R IPD: R/W	デフォルト値: ARD: 0 APD: 0 IRD: D IPD: 0
SQL_DESC_ROWS_PROCESSED_PTR (SQLINTEGER *)		

表 184. ヘッダー・フィールドの初期設定 (続き)

R/W:	ARD: 未使用	デフォルト値:	ARD: 未使用
	APD: 未使用		APD: 未使用
	IRD: R/W		IRD: NULL ポインター
	IPD: R/W		IPD: NULL ポインター

- a** これらのフィールドは、IPD が DB2 CLI によって自動的に移植される場合にのみ定義されます。フィールドが自動的に移植されない場合には定義されません。アプリケーションがこれらのフィールドを設定しようとすると、SQLSTATE HY091 (記述子のフィールド ID が無効) が返されます。

レコード・フィールドの初期設定は以下のとおりです。

表 185. レコード・フィールドの初期設定

SQL_DESC_AUTO_UNIQUE_VALUE (SQLINTEGER)

R/W:	ARD: 未使用	デフォルト	ARD: 未使用
	APD: 未使用	値:	APD: 未使用
	IRD: R		IRD: D
	IPD: 未使用		IPD: 未使用

SQL_DESC_BASE_COLUMN_NAME (SQLCHAR *)

R/W:	ARD: 未使用	デフォルト	ARD: 未使用
	APD: 未使用	値:	APD: 未使用
	IRD: R		IRD: D
	IPD: 未使用		IPD: 未使用

SQL_DESC_BASE_TABLE_NAME (SQLCHAR *)

R/W:	ARD: 未使用	デフォルト	ARD: 未使用
	APD: 未使用	値:	APD: 未使用
	IRD: R		IRD: D
	IPD: 未使用		IPD: 未使用

SQL_DESC_CASE_SENSITIVE (SQLINTEGER)

R/W:	ARD: 未使用	デフォルト	ARD: 未使用
	APD: 未使用	値:	APD: 未使用
	IRD: R		IRD: D
	IPD: R		IPD: D ^a

SQL_DESC_CATALOG_NAME (SQLCHAR *)

表 185. レコード・フィールドの初期設定 (続き)

R/W:	ARD: 未使用	デフォルト	ARD: 未使用
	APD: 未使用	値:	APD: 未使用
	IRD: R		IRD: D
	IPD: 未使用		IPD: 未使用
SQL_DESC_CONCISE_TYPE (SQLSMALLINT)			
R/W:	ARD: R/W	デフォルト	ARD: SQL_C_DEFAULT
	APD: R/W	値:	APD: SQL_C_DEFAULT
	IRD: R		IRD: D
	IPD: R/W		IPD: ND
SQL_DESC_DATA_PTR (SQLPOINTER)			
R/W:	ARD: R/W	デフォルト	ARD: NULL ポインター
	APD: R/W	値:	APD: NULL ポインター
	IRD: 未使用		IRD: 未使用
	IPD: 未使用		IPD: 未使用 ^b
SQL_DESC_DATETIME_INTERVAL_CODE (SQLSMALLINT)			
R/W:	ARD: R/W	デフォルト	ARD: ND
	APD: R/W	値:	APD: ND
	IRD: R		IRD: D
	IPD: R/W		IPD: ND
SQL_DESC_DATETIME_INTERVAL_PRECISION (SQLINTEGER)			
R/W:	ARD: R/W	デフォルト	ARD: ND
	APD: R/W	値:	APD: ND
	IRD: R		IRD: D
	IPD: R/W		IPD: ND
SQL_DESC_DISPLAY_SIZE (SQLINTEGER)			
R/W:	ARD: 未使用	デフォルト	ARD: 未使用
	APD: 未使用	値:	APD: 未使用
	IRD: R		IRD: D
	IPD: 未使用		IPD: 未使用
SQL_DESC_FIXED_PREC_SCALE (SQLSMALLINT)			

表 185. レコード・フィールドの初期設定 (続き)

R/W:	ARD: 未使用	デフォルト	ARD: 未使用
	APD: 未使用	値:	APD: 未使用
	IRD: R		IRD: D
	IPD: R		IPD: D ^a
SQL_DESC_INDICATOR_PTR (SQLINTEGER *)			
R/W:	ARD: R/W	デフォルト	ARD: NULL ポインター
	APD: R/W	値:	APD: NULL ポインター
	IRD: 未使用		IRD: 未使用
	IPD: 未使用		IPD: 未使用
SQL_DESC_LABEL (SQLCHAR *)			
R/W:	ARD: 未使用	デフォルト	ARD: 未使用
	APD: 未使用	値:	APD: 未使用
	IRD: R		IRD: D
	IPD: 未使用		IPD: 未使用
SQL_DESC_LENGTH (SQLINTEGER)			
R/W:	ARD: R/W	デフォルト	ARD: ND
	APD: R/W	値:	APD: ND
	IRD: R		IRD: D
	IPD: R/W		IPD: ND
SQL_DESC_LITERAL_PREFIX (SQLCHAR *)			
R/W:	ARD: 未使用	デフォルト	ARD: 未使用
	APD: 未使用	値:	APD: 未使用
	IRD: R		IRD: D
	IPD: 未使用		IPD: 未使用
SQL_DESC_LITERAL_SUFFIX (SQLCHAR *)			
R/W:	ARD: 未使用	デフォルト	ARD: 未使用
	APD: 未使用	値:	APD: 未使用
	IRD: R		IRD: D
	IPD: 未使用		IPD: 未使用
SQL_DESC_LOCAL_TYPE_NAME (SQLCHAR *)			

表 185. レコード・フィールドの初期設定 (続き)

R/W:	ARD: 未使用	デフォルト	ARD: 未使用
	APD: 未使用	値:	APD: 未使用
	IRD: R		IRD: D
	IPD: R		IPD: D ^a
SQL_DESC_NAME (SQLCHAR *)			
R/W:	ARD: 未使用	デフォルト	ARD: ND
	APD: 未使用	値:	APD: ND
	IRD: R		IRD: D
	IPD: R/W		IPD: ND
SQL_DESC_NULLABLE (SQLSMALLINT)			
R/W:	ARD: 未使用	デフォルト	ARD: ND
	APD: 未使用	値:	APD: ND
	IRD: R		IRD: N
	IPD: R		IPD: ND
SQL_DESC_NUM_PREC_RADIX (SQLINTEGER)			
R/W:	ARD: R/W	デフォルト	ARD: ND
	APD: R/W	値:	APD: ND
	IRD: R		IRD: D
	IPD: R/W		IPD: ND
SQL_DESC_OCTET_LENGTH (SQLINTEGER)			
R/W:	ARD: R/W	デフォルト	ARD: ND
	APD: R/W	値:	APD: ND
	IRD: R		IRD: D
	IPD: R/W		IPD: ND
SQL_DESC_OCTET_LENGTH_PTR (SQLINTEGER *)			
R/W:	ARD: R/W	デフォルト	ARD: NULL ポインター
	APD: R/W	値:	APD: NULL ポインター
	IRD: 未使用		IRD: 未使用
	IPD: 未使用		IPD: 未使用
SQL_DESC_PARAMETER_TYPE (SQLSMALLINT)			

表 185. レコード・フィールドの初期設定 (続き)

R/W:	ARD: 未使用	デフォルト	ARD: 未使用
	APD: 未使用	値:	APD: 未使用
	IPD: 未使用		IPD: 未使用
	IRD: R/W		IRD: D=SQL_PARAM_INPUT
SQL_DESC_PRECISION (SQLSMALLINT)			
R/W:	ARD: R/W	デフォルト	ARD: ND
	APD: R/W	値:	APD: ND
	IRD: R		IRD: D
	IPD: R/W		IPD: ND
SQL_DESC_SCALE (SQLSMALLINT)			
R/W:	ARD: R/W	デフォルト	ARD: ND
	APD: R/W	値:	APD: ND
	IRD: R		IRD: D
	IPD: R/W		IPD: ND
SQL_DESC_SCHEMA_NAME (SQLCHAR *)			
R/W:	ARD: 未使用	デフォルト	ARD: 未使用
	APD: 未使用	値:	APD: 未使用
	IRD: R		IRD: D
	IPD: 未使用		IPD: 未使用
SQL_DESC_SEARCHABLE (SQLSMALLINT)			
R/W:	ARD: 未使用	デフォルト	ARD: 未使用
	APD: 未使用	値:	APD: 未使用
	IRD: R		IRD: D
	IPD: 未使用		IPD: 未使用
SQL_DESC_TABLE_NAME (SQLCHAR *)			
R/W:	ARD: 未使用	デフォルト	ARD: 未使用
	APD: 未使用	値:	APD: 未使用
	IRD: R		IRD: D
	IPD: 未使用		IPD: 未使用
SQL_DESC_TYPE (SQLSMALLINT)			

表 185. レコード・フィールドの初期設定 (続き)

R/W:	ARD: R/W	デフォルト	ARD: SQL_C_DEFAULT
	APD: R/W	値:	APD: SQL_C_DEFAULT
	IRD: R		IRD: D
	IPD: R/W		IPD: ND

SQL_DESC_TYPE_NAME (SQLCHAR *)

R/W:	ARD: 未使用	デフォルト	ARD: 未使用
	APD: 未使用	値:	APD: 未使用
	IRD: R		IRD: D
	IPD: R		IPD: D ^a

SQL_DESC_UNNAMED (SQLSMALLINT)

R/W:	ARD: 未使用	デフォルト	ARD: ND
	APD: 未使用	値:	APD: ND
	IRD: R		IRD: D
	IPD: R/W		IPD: ND

SQL_DESC_UNSIGNED (SQLSMALLINT)

R/W:	ARD: 未使用	デフォルト	ARD: 未使用
	APD: 未使用	値:	APD: 未使用
	IRD: R		IRD: D
	IPD: R		IPD: D ^a

SQL_DESC_UPDATABLE (SQLSMALLINT)

R/W:	ARD: 未使用	デフォルト	ARD: 未使用
	APD: 未使用	値:	APD: 未使用
	IRD: R		IRD: D
	IPD: 未使用		IPD: 未使用

a これらのフィールドは、IPD が DB2 CLI によって自動的に移植される場合のみ定義されます。フィールドが自動的に移植されない場合には定義されません。アプリケーションがこれらのフィールドを設定しようとすると、SQLSTATE HY091 (記述子のフィールド ID が無効) が返されます。

b IPD 内の SQL_DESC_DATA_PTR フィールドは、整合性検査を強制的に行わせるように設定できます。その後の SQLGetDescField() または SQLGetDescRec() への呼び出しにおいて、DB2 CLI は SQL_DESC_DATA_PTR が設定された値を返す必要はありません。

関連概念:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『CLI アプリケーションの記述子』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『CLI アプリケーションの記述子の整合性検査』

関連資料:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『CLI アプリケーション用の C データ・タイプ』
- 905 ページの『SQLGetDescField 関数 (CLI) - 記述子レコードの単一フィールド設定の取得』
- 910 ページの『SQLSetDescField 関数 (CLI) - 記述子レコードの単一フィールドの設定』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『記述子 FieldIdentifier 引き数の値 (CLI)』

CLI アプリケーション用のプログラミングのヒントと提案

このトピックでは、以下の対象について説明します。

- 828 ページの『共通接続属性』
- 828 ページの『共通ステートメント属性』
- 829 ページの『ステートメント・ハンドルの再利用』
- 829 ページの『バインドおよび SQLGetData()』
- 829 ページの『カタログ関数の使用を制限する』
- 830 ページの『関数生成による結果セットの列名』
- 830 ページの『ODBC アプリケーションからロードした DB2 CLI ユニークの関数』
- 830 ページの『グローバル動的ステートメント・キャッシュ』
- 831 ページの『データ追加および検索の最適化』
- 831 ページの『ラージ・オブジェクト・データの最適化』
- 831 ページの『オブジェクト ID の大文字小文字の区別』
- 832 ページの『SQLDriverConnect() と SQLConnect()』
- 832 ページの『SQL ガバナーのインプリメンテーション』
- 833 ページの『ステートメント走査をオフにする』
- 833 ページの『複数のロールバックでカーソルを保留する』
- 834 ページの『コンパウンド SQL サブステートメントの作成』
- 834 ページの『ユーザー定義タイプおよびキャスト』
- 835 ページの『ネットワーク・フローを減らすための据え置き準備』

DB2 CLI の更新の詳細は、以下の DB2[®] アプリケーション開発の Web サイトを参照してください。

<http://www.ibm.com/software/data/db2/udb/ad>

共通接続属性:

以下の接続属性は、DB2 CLI アプリケーションによって設定することが必要な場合があります。

- **SQL_ATTR_AUTOCOMMIT** - 各コミット要求が余分なネットワーク・フローを生成できるので、通常この属性は **SQL_AUTOCOMMIT_OFF** に設定します。特に必要な場合に限って、**SQL_AUTOCOMMIT_ON** をオンにしておきます。

注: デフォルト値は **SQL_AUTOCOMMIT_ON** です。

- **SQL_ATTR_TXN_ISOLATION** - この接続属性は、接続またはステートメント操作時の分離レベルを判別します。分離レベルとは、可能な並行性のレベル、およびステートメントを実行するのに必要なロックのレベルを決めるものです。アプリケーションにしてみれば、並行性を最大にし、一方でデータの一貫性が保証される分離レベルを選択することが必要になります。

共通ステートメント属性:

以下のステートメント属性は、DB2 CLI アプリケーションによって設定することが必要な場合があります。

- **SQL_ATTR_MAX_ROWS** - このオプションを設定することにより、照会操作によってアプリケーションに戻される行数を制限することができます。アプリケーション（とりわけ、メモリー・リソースが制限されているクライアント上のアプリケーション）で非常に大きな結果セットが不用意に生成されて処理できなくなるといった状況を避けるために、このオプションを使用することができます。

DB2 (z/OS[™] および OS/390 版) バージョン 7 以降への接続中に

SQL_ATTR_MAX_ROWS を設定すると、ステートメントに "OPTIMIZE FOR n ROWS" および "FETCH n ROWS ONLY" 文節が追加されます。バージョン 7 より前のバージョンの DB2 (OS/390 版) や、"FETCH n ROWS ONLY" 文節をサポートしない DBMS の場合、サーバー側で "OPTIMIZE FOR n ROWS" 文節を使用して完全な結果セットが生成されますが、DB2 CLI は、クライアント上で行をカウントし、"SQL_ATTR_MAX_ROWS" 行までしかフェッチしません。

- **SQL_ATTR_CURSOR_HOLD** - このステートメント属性は、DB2 CLI がこのステートメントのカーソルを **WITH HOLD** 文節を使用して宣言するかどうかを決めます。カーソル保留動作を必要としないステートメントがこの属性を **SQL_CURSOR_HOLD_OFF** に設定する場合、ステートメント・ハンドルに関連するリソースはサーバーによってよりよく活用されます。この属性を適切に使用することによって得られる効率上の利点は、OS/390 および z/OS 上で著しいものです。

注: ODBC アプリケーションの多くは、コミット後にカーソル位置が保持されている場合に、デフォルトの動作を予期します。

- `SQL_ATTR_TXN_ISOLATION` - `DB2 CLI` では分離レベルをステートメント・レベルで設定することが可能です。ただし、分離レベルは接続レベルで設定することをお勧めします。分離レベルとは、可能な並行性のレベル、およびステートメントを実行するのに必要なロックのレベルを決めるものです。

すべてのステートメントがデフォルトの分離レベルのままにされているのではなく、要求された分離レベルに設定されていれば、ステートメント・ハンドルに関連するリソースは `DB2 CLI` によってよりよく活用されます。以上のことは、接続している `DBMS` のロックおよび分離レベルについて完全に理解している場合にのみ行ってください。

アプリケーションは、並行性を最大にするように、最小の分離レベルを使用すべきです。

ステートメント・ハンドルの再利用:

`CLI` アプリケーションがステートメント・ハンドルを宣言するたびに、`DB2 CLI` ドライバーは、そのハンドルの基礎となるデータ構造を割り振って初期設定します。パフォーマンスを向上させるために、`CLI` アプリケーションは、別のステートメントでステートメント・ハンドルを再利用することにより、ステートメント・ハンドルの割り振りと初期設定に関連したコストを削減できます。

注: ステートメント・ハンドルを再利用する前に、以前のステートメントで使用されたメモリー・バッファおよび他のリソースを、`SQLFreeStmt()` 関数を呼び出すことによって解放しなければならない場合があります。また、ステートメント・ハンドルに以前に設定されたステートメント属性 (たとえば、`SQL_ATTR_PARAMSET_SIZE`) を明示的にリセットする必要もあります。そうしない場合、ステートメント・ハンドルを使用する今後のすべてのステートメントに継承される可能性があります。

バインドおよび `SQLGetData()`:

通常は、`SQLGetData()` の使用に比べて、アプリケーション変数または結果セットへのファイル参照をバインドするほうが効率的です。データが `LOB` 列にある場合、`SQLGetData()` よりも `LOB` 関数のほうが望ましいです (詳細は、831 ページの『ラーズ・オブジェクト・データの最適化』を参照)。データ値が大きい可変長データであるために、状況が以下のような場合には、`SQLGetData()` を使用してください。

- データを分割して受け取らなければならない。または、
- データを検索する必要がない。

カタログ関数の使用を制限する:

`SQLTables()` のようなカタログ関数により、`DB2 CLI` ドライバーは、情報を取り出すために `DBMS` カタログ表を照会します。発行される照会は複雑で、`DBMS` カタログ表は

非常に大きくなる可能性があります。一般論として、カタログ関数を呼び出せる回数の制限、および戻される行数の制限について論じます。

一度関数を呼び出してから、そのデータをアプリケーションに保管させる (キャッシュに入れる) ことによって、カタログ関数呼び出しの回数を減らすことが可能です。

戻される行数は、以下を指定することによって制限することができます。

- すべてのカタログ関数のスキーマ名またはパターン
- `SQLTables()` 以外のすべてのカタログ関数の表名またはパターン
- 詳細な列情報を戻すカタログ関数の列名またはパターン

ただ、アプリケーションの開発やテストを何百もの表を持つデータ・ソースに対して行ったとしても、アプリケーションの実行は何千もの表を持つデータベースに対して行われることがある、という点を覚えておいてください。アプリケーションを開発する際には、この類似性を考慮に入れてください。

カタログ照会に使用されたステートメント・ハンドルのカーソルでオープンしているものをクローズし (`SQL_CLOSE Option` を指定して `SQLCloseCursor()` または `SQLFreeStmt()` を呼び出す)、カタログ表へのロックをすべて解除します。カタログ表に未解決のロックがあると、`CREATE`、`DROP` または `ALTER` ステートメントを実行できなくなることがあります。

関数生成による結果セットの列名:

カタログおよび情報関数により生成される結果セットの列名は、`ODBC` および `CLI` 標準が変化するにつれて変更されることがあります。ただし、列の位置が変更されることはありません。

アプリケーション依存性は列の位置 (`SQLBindCol()`、`SQLGetData()`、および `SQLDescribeCol()` で使用される `iCol` パラメーター) に基づいており、列の名前には基づいていません。

ODBC アプリケーションからロードした DB2 CLI ユニークの関数:

ODBC ドライバー・マネージャーは、自分のステートメント・ハンドルのセットを保持しつつ、それを各呼び出しの `CLI` ステートメント・ハンドルにマッピングします。DB2 CLI 関数が直接に呼び出される場合、CLI ドライバーには ODBC マッピングへのアクセス権がないため、CLI ドライバーのステートメント・ハンドルに渡す必要があります。

DB2 CLI ステートメント・ハンドル (HSTMT) を入手するには、`SQLGetInfo()` に `SQL_DRIVER_HSTMT` オプションを指定して呼び出します。DB2 CLI 関数は、必要な箇所では HSTMT 引き数を渡すことによって、共有ライブラリーまたは DLL から直接呼び出すことができます。

グローバル動的ステートメント・キャッシュ:

UNIX® または Windows® 用のバージョン 5 以降の DB2 Universal Database サーバーには、グローバル動的ステートメント・キャッシュが備えられています。このキャッシュは、準備状態の動的 SQL ステートメントに対する最も一般的なアクセス・プランを保管するのに使用します。

各ステートメントが準備される前に、サーバーは自動的にこのキャッシュを検索して、(このアプリケーションか別のアプリケーション、またはクライアントによって) この SQL ステートメント用のアクセス・プランが作成済みであるかどうかを調べます。作成済みであれば、サーバーが新たにアクセス・プランを生成する必要はなく、代わりに、キャッシュの中にあるものを使用します。現在では、グローバル動的ステートメント・キャッシュのないサーバーへの接続でなければ、アプリケーションがクライアントで接続をキャッシュする必要はありません。

データ追加および検索の最適化:

パラメーターをバインドしてデータを検索するために配列を使用することを記述する方式では、コンパウンド SQL を使用してネットワーク・フローを最適化します。可能な限りこれらの方法を用いるようにしてください。

ラージ・オブジェクト・データの最適化:

長形式ストリングには、可能な限り LOB データ・タイプおよびそれをサポートする関数を使用してください。LONG VARCHAR、LONG VARBINARY、および LONG VARCHARGRAPHIC タイプとは異なり、LOB データ値は LOB ロケーターおよび SQLGetPosition() や SQLGetSubString() などの関数を使用して、サーバーの大きなデータ値を操作することができます。

LOB 値をファイルに直接取り出すこともできますし、LOB パラメーター値をファイルから直接読み取ることができます。このようにすると、アプリケーション・バッファーを經由してデータを転送するアプリケーションのオーバーヘッドを節約することができます。

オブジェクト ID の大文字小文字の区別:

表名、ビュー名、および列名など、データベース・オブジェクト ID はすべて、その ID が区切られていなければ、カタログ表には大文字で保管されます。区切り名を用いて ID が作成された場合には、名前前の記述に用いられた文字がそのままカタログ表に保管されます。

ID が SQL ステートメント内で参照されると、ID が区切られていなければ、大小文字は区別なしとして処理されます。

たとえば、以下の 2 つの表が作成された場合、

```
CREATE TABLE MyTable (id INTEGER)
CREATE TABLE "YourTable" (id INTEGER)
```

2 つの表 MYTABLE と YourTable が存在することになります。

以下の 2 つのステートメントは同等です。

```
SELECT * FROM MyTable (id INTEGER)
SELECT * FROM MYTABLE (id INTEGER)
```

以下の 2 番目のステートメントは、YOURTABLE と命名された表がないため、TABLE NOT FOUND というエラーが出されて失敗します。

```
SELECT * FROM "YourTable" (id INTEGER) // executes without error
SELECT * FROM YourTable (id INTEGER) // error, table not found
```

すべての DB2 CLI カタログ関数の引き数は、オブジェクトの名前を大文字小文字の区別あり、すなわち各名前が区切られているものとして処理します。

SQLDriverConnect() と SQLConnect():

SQLDriverConnect() を使用することにより、アプリケーションはユーザーへの接続情報の入力要求を DB2 CLI によって提供されるダイアログ・ボックスに任せることができます。

あるアプリケーションが接続情報の照会にアプリケーション自身のダイアログ・ボックスを使用している場合、ユーザーは接続ストリングに追加の接続オプションを指定できます。このストリングも保管され、それ以後の接続ではデフォルト値として使用されません。

SQL ガバナーのインプリメンテーション:

SQL ステートメントが作成されるたびに、サーバーはステートメントのコストを見積もります。次いで、アプリケーションはそのステートメントの実行を継続するかどうかを判断します。

この見積もりを SQLCA (SQLERRD(4)) から入手し、アプリケーションが直接使用することができますし、または SQL_ATTR_DB2ESTIMATE 接続属性をしきい値に設定することができます。見積もられたステートメントのコストがどれもしきい値を超えている場合には、DB2 CLI が警告のダイアログ・ボックスを表示し、ステートメントの継続または取り消しを入力要求します。

提案されているしきい値は 60000 ですが、通常アプリケーションではエンド・ユーザーがしきい値を設定できるようになっています。

注: この見積もりは、サーバーがステートメントを実行するために使用しているリソースの合計の見積もりだけであって、ステートメントを実行するために必要な時間を示してはいません。

結果の行数の見積もりは SQLCA (SQLERRD(3)) から利用可能で、大きい照会を制限するためにアプリケーションが使用することも可能です。

注: SQLERRD(3) および SQLERRD(4) フィールドに戻される情報の正確度は、パラメーター・マーカーの使用のようなさまざまな要素、およびステートメント内のさまざまな式によって異なります。カタログ表のデータベース統計が最新であれば、より正確な情報が示されます。コマンド行プロセッサ・セッションから RUNSTATS コマンドを発行することにより、DB2 Universal Database 上でデータベース統計を更新することができます。

ステートメント走査をオフにする:

DB2 CLI はデフォルト設定で、ベンダー・エスケープ文節順序列を探索して、SQL ステートメントを一つ走査します。

アプリケーションがベンダー・エスケープ文節順序列を含む SQL ステートメントを生成しない場合は、SQL_ATTR_NOSCAN ステートメント属性を接続レベルで SQL_NOSCAN_ON に設定することによって、DB2 CLI がベンダー・エスケープ文節を探索して文節スキャンを実行することがないようにすることができます。

複数のロールバックでカーソルを保留する:

トランザクション管理上の複雑な問題を処理することが必要なアプリケーションでは、同一のデータベースに複数の並行接続を確立するとよい場合があります。DB2 CLI 内の各接続にはそれぞれトランザクション有効範囲があり、1 つの接続で実行されるアクションが他の接続のトランザクションに影響を与えることはありません。

たとえば、ある 1 つのトランザクション内でオープンされているすべてのカーソルは、問題が起ってそのトランザクションがロールバックされるとクローズしてしまいます。カーソルは一つのトランザクション内に個別にあり、あるステートメントでロールバックがなされても他のステートメントのカーソルには影響しないので、アプリケーションは同一のデータベースに複数の接続を使用して、オープンしているカーソルを持つ複数のステートメントを分離しておきます。

しかし、複数の接続を使用するという事は、ある接続でクライアントにデータを渡してから、別の接続でサーバーにそのデータを戻すということを意味します。例:

- 接続 #1 で、ラージ・オブジェクト列にアクセスしており、かつラージ・オブジェクト値の一部にマッピングする LOB ロケーターを作成していると仮定します。
- 接続 #2 で、LOB ロケーターにより表される LOB 値の一部を使用 (挿入) する場合、まず接続 #1 の LOB 値をアプリケーションに移動し、それから接続 #2 で作業中の表に渡す必要があります。そうする理由は、接続 #2 が接続 #1 の LOB ロケーターを認識しないためです。

- 接続が 1 つしかなければ、LOB ロケータを直接使用することができます。ただし、トランザクションをロールバックするとすぐに、LOB ロケータは失われてしまいます。

注: あるアプリケーションによって 1 つのデータベースに対する複数の接続が使用される場合、そのアプリケーションでは、データベース・オブジェクトに対するアクセスを注意深く同期化する必要があります。そのようにしないと、データベース・ロックはトランザクション間で共有されるものではないため、さまざまなロック競合問題が生じる可能性があります。ある接続によって更新が行われると、最初の接続が (COMMIT または ROLLBACK によって) ロックを解放するまで、他の接続も容易にロック待機状態になってしまう場合があります。

コンパウンド SQL サブステートメントの作成:

コンパウンド・ステートメントの効率を最大にするには、BEGIN COMPOUND ステートメントの前にサブステートメントを作成し、次いでコンパウンド・ステートメント内でそのサブステートメントを実行します。

このようにすることによっても、準備エラーがコンパウンド・ステートメントの外側で処理されるので、エラー処理が単純化されます。

ユーザー定義タイプおよびキャスト:

照会ステートメントの述部にパラメーター・マーカが使用されており、かつ、そのパラメーターがユーザー定義タイプである場合には、ステートメントに CAST 関数を使用して、パラメーター・マーカまたは UDT のいずれかをキャストする必要があります。

たとえば、以下のようにタイプおよび表が定義されているとします。

```
CREATE DISTINCT TYPE CNUM AS INTEGER WITH COMPARISONS

CREATE TABLE CUSTOMER (
    Cust_Num      CNUM NOT NULL,
    First_Name    CHAR(30) NOT NULL,
    Last_Name     CHAR(30) NOT NULL,
    Phone_Num     CHAR(20) WITH DEFAULT,
    PRIMARY KEY  (Cust_Num) )
```

さらに、その後で以下の SQL ステートメントが発行されたとします。

```
SELECT first_name, last_name, phone_num from customer
WHERE cust_num = ?
```

このステートメントはパラメーター・マーカがタイプ CNUM にはならないために失敗し、したがってタイプに互換性がないことから比較が失敗して、以下のようになりません。

列を整数 (その基本 SQL タイプ) にキャストすると、パラメーターには整数のタイプが与えられるので、比較を実行することができます。

```
SELECT first_name, last_name, phone_num from customer
where cast( cust_num as integer ) = ?
```

あるいは、パラメーター・マーカーを INTEGER にキャストすることによって、サーバーは INTEGER に CNUM 変換を適用することができます。

```
SELECT first_name, last_name, phone_num FROM customer
where cust_num = cast( ? as integer )
```

ネットワーク・フローを減らすための据え置き準備:

DB2 CLI では、デフォルトで据え置き準備がオンになります。対応する実行要求が発行されるまで、PREPARE 要求はサーバーに送られません。2 つの要求が 2 つではなく 1 つのコマンド / 応答の流れに結合され、ネットワークの流れを最小化して、パフォーマンスを改善します。これが最大の利点となるのは、アプリケーションが照会を生成し、その応答のセットが非常に少ない場合や、別々の要求と回答のオーバーヘッドが照会データの複数ブロックに広がっていない場合です。DB2 コネクトや DDCS ゲートウェイを使用する環境では、要求と回答の 4 つの組み合わせが 2 つに減るため、コスト削減の機会ともなります。

注: SQLDescribeParam(), SQLDescribeCol(), SQLNumParams(), および SQLNumResultCols() のような関数では、ステートメントを準備しておく必要があります。ステートメントが準備されていない場合、これらの関数は、サーバーに対して即時 PREPARE 要求を生成するため、据え置き準備の効果は表れません。

関連概念:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI アプリケーションでのシステム・カタログ情報の照会のためのカタログ関数』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI アプリケーションでのベンダー・エスケープ文節』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI でのハンドル』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI アプリケーションのカーソル』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI アプリケーションでのラージ・オブジェクトの使用法』

関連タスク:

- 836 ページの『CLI アプリケーションでのコンパウンド SQL ステートメントの実行』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI アプリケーションでの列方向バインドを使用した配列データの取り出し』

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『CLI アプリケーションでの行方向バインドを使用した配列データの取り出し』

関連資料:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『CLI と ODBC 関数のサマリー』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『SQLConnect 関数 (CLI) - データ・ソースへの接続』
- 888 ページの『SQLDriverConnect 関数 (CLI) - データ・ソースへの (拡張) 接続』
- 895 ページの『SQLGetData 関数 (CLI) - 列からのデータの取得』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『ステートメント属性 (CLI) のリスト』
- 802 ページの『接続属性 (CLI) リスト』

CLI アプリケーションでのコンパウンド SQL ステートメントの実行

コンパウンド SQL を使用すると、複数の SQL ステートメントをグループ化して単一の実行可能ブロックにすることができます。このステートメントのブロックを入力パラメーター値と共に使って、1 つの連続ストリームで実行することができ、これにより実行時間およびネットワーク通信量を少なくすることができます。

制約事項:

- コンパウンド SQL は、サブステートメントが実行される順序を保証しないので、サブステートメント間に依存性があってはなりません。
- コンパウンド SQL ステートメントはネストすることはできません。
- BEGIN COMPOUND および END COMPOUND ステートメントは、同じステートメント・ハンドルで実行する必要があります。
- BEGIN COMPOUND SQL ステートメントの STOP AFTER FIRST ? STATEMENTS 文節で指定される値は、タイプ SQL_INTEGER でなければならず、この値に対して、タイプ SQL_C_INTEGER または SQL_C_SMALLINT のアプリケーション・バッファーだけをバインドできます。
- 個々のサブステートメントには独自のステートメント・ハンドルが必要です。
- すべてのステートメント・ハンドルは同じ接続に属し、同じ分離レベルでなければなりません。
- アトミック配列入力は、SQL ステートメントの BEGIN COMPOUND および END COMPOUND ブロック内ではサポートされていません。アトミック配列入力とは、挿入が 1 回でも失敗すると、すべての挿入を取り消す動作を指します。
- END COMPOUND ステートメントが実行されるまで、ステートメント・ハンドルはすべて割り振られている状態でなければなりません。

- `SQLEndTran()` は、同一接続で、または `BEGIN COMPOUND` および `END COMPOUND` 間の接続要求で呼び出すことはできません。
- コンパウンド・サブステートメント用に割り振られたステートメント・ハンドルを使用して呼び出せるのは、以下の関数だけです。

```

|         - SQLAllocHandle()
|         - SQLBindParameter()
|         - SQLBindFileToParam()
|         - SQLExecute()
|         - SQLParamData()
|         - SQLPrepare()
|         - SQLPutData()

```

手順:

```

| CLI アプリケーションでコンパウンド SQL ステートメントを実行するには、以下のよ
| うにします。

```

1. 親ステートメント・ハンドルを割り振ります。以下に例を示します。

```
SQLAllocHandle (SQL_HANDLE_STMT, hdbc, &hstmtparent);
```

2. コンパウンド・サブステートメントごとにステートメント・ハンドルを割り振ります。以下に例を示します。

```
SQLAllocHandle (SQL_HANDLE_STMT, hdbc, &hstmtsub1);
SQLAllocHandle (SQL_HANDLE_STMT, hdbc, &hstmtsub2);
SQLAllocHandle (SQL_HANDLE_STMT, hdbc, &hstmtsub3);
```

3. サブステートメントを準備します。以下に例を示します。

```
SQLPrepare (hstmtsub1, stmt1, SQL_NTS);
SQLPrepare (hstmtsub2, stmt2, SQL_NTS);
SQLPrepare (hstmtsub3, stmt3, SQL_NTS);
```

4. 親ステートメント・ハンドルを使用して `BEGIN COMPOUND` ステートメントを実行します。以下に例を示します。

```
SQLExecDirect (hstmtparent, (SQLCHAR *) "BEGIN COMPOUND NOT ATOMIC STATIC",
SQL_NTS);
```

5. これがアトミック・コンパウンド SQL 操作の場合は、`SQLExecute()` 関数を使用し
てのみ、サブステートメントを実行してください。以下に例を示します。

```

| SQLExecute (hstmtsub1);
| SQLExecute (hstmtsub2);
| SQLExecute (hstmtsub3);

```

注: アトミック・コンパウンド・ブロック内で実行されるすべてのステートメントを、最初に準備する必要があります。アトミック・コンパウンド・ブロック内で `SQLExecDirect()` 関数を使用しようとすると、エラーになります。

6. 親ステートメント・ハンドルを使用して END COMPOUND ステートメントを実行します。以下に例を示します。

```
SQLExecDirect (hstmtparent, (SQLCHAR *) "END COMPOUND NOT ATOMIC STATIC",  
                SQL_NTS);
```

7. オプション: 入力パラメーター値の配列を使用した場合、親ステートメント・ハンドルを指定した SQLRowCount () を呼び出して、入力配列のすべてのエレメントに影響を受ける行数をまとめて検索します。以下に例を示します。

```
SQLRowCount (hstmtparent, &numRows);
```

8. サブステートメントのハンドルを解放します。以下に例を示します。

```
SQLFreeHandle (SQL_HANDLE_STMT, hstmtsub1);  
SQLFreeHandle (SQL_HANDLE_STMT, hstmtsub2);  
SQLFreeHandle (SQL_HANDLE_STMT, hstmtsub3);
```

9. 親ステートメント・ハンドルの使用が終了したら、その親ステートメント・ハンドルを解放します。以下に例を示します。

```
SQLFreeHandle (SQL_HANDLE_STMT, hstmtparent);
```

アプリケーションが自動コミット・モードで動作しておらず、COMMIT オプションが指定されていない場合、サブステートメントはコミットされません。しかし、アプリケーションが自動コミット・モードで動作している場合には、COMMIT オプションが指定されていなくても、サブステートメントは END COMPOUND 時にコミットされます。

関連タスク:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『CLI アプリケーションでのステートメント・ハンドルの割り振り』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『CLI アプリケーションでの SQL ステートメントの準備と実行』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『列方向配列の入力を使用した CLI アプリケーションでのパラメーター・マーカのバインド』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『行方向配列の入力を使用した CLI アプリケーションでのパラメーター・マーカのバインド』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『CLI アプリケーションでのステートメント・リソースの解放』

関連資料:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『SQLAllocHandle 関数 (CLI) - ハンドルの割り振り』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『SQLExecDirect 関数 (CLI) - ステートメントの直接実行』

- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻の『SQLExecute 関数 (CLI) - ステートメントの実行』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻の『SQLFreeHandle 関数 (CLI) - ハンドル・リソースの解放』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻の『SQLPrepare 関数 (CLI) - ステートメントの準備』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻の『SQLRowCount 関数 (CLI) - 行カウントの取得』
- SQL リファレンス 第 2 巻の『COMMIT ステートメント』
- SQL リファレンス 第 2 巻の『ROLLBACK ステートメント』
- SQL リファレンス 第 2 巻の『コンパウンド SQL (動的) ステートメント』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI アプリケーションでの複合 SQL の戻りコード』

関連サンプル:

- 『dbuse.c -- How to use a database (CLI)』

CLI アプリケーションからのストアード・プロシージャの呼び出し

CLI アプリケーションは、CALL プロシージャ SQL ステートメントを実行することにより、ストアード・プロシージャを呼び出します。このトピックでは、CLI アプリケーションからストアード・プロシージャを呼び出す方法を説明します。

前提条件:

ストアード・プロシージャを呼び出す前に、CLI アプリケーションを初期設定しておくようにします。

手順:

ストアード・プロシージャを呼び出すには、以下のようになります。

1. ストアード・プロシージャの IN、INOUT、および OUT パラメーターにそれぞれ対応するアプリケーション・ホスト変数を宣言します。アプリケーションの変数データのタイプと長さが、ストアード・プロシージャのシグニチャーのデータ・タイプと引き数の長さに一致することを確認します。DB2 CLIは、すべての SQL タイプをパラメーター・マーカースとして使用して、ストアード・プロシージャを呼び出すことをサポートしています。
2. IN、INOUT、および OUT パラメーターのアプリケーション変数を初期設定します。
3. CALL SQL ステートメントを発行します。以下に例を示します。

```
SQLCHAR *stmt = (SQLCHAR *)"CALL OUT_LANGUAGE (?);"
```

パフォーマンスを最高にするために、アプリケーションでは、CALL プロシージャ・ストリングの中でストアード・プロシージャ引き数のパラメーター・マーカースを使用して、ホスト変数をこれらのパラメーター・マーカースにバインドする必要があります。ただし、インバウンド・ストアード・プロシージャ引き数を、パラメーター・マーカースではなく、ストリング・リテラルとして指定しなければならない場合、CALL プロシージャ・ステートメントに、ODBC 呼び出しエスケープ文節の区切り文字 { } を含めます。以下に例を示します。

```
SQLCHAR *stmt = (SQLCHAR *)"{CALL IN_PARAM (123, 'Hello World!')}";
```

CALL プロシージャ・ステートメントでストリング・リテラルおよび ODBC エスケープ文節が使用される場合、IN モード・ストアード・プロシージャ引き数として、ストリング・リテラルだけを指定できます。INOUT および OUT モード・ストアード・プロシージャ引き数は、引き続きパラメーター・マーカースを使用して指定する必要があります。

- オプション: SQLPrepare() を呼び出して CALL ステートメントを準備します。
- SQLBindParameter() を呼び出して、CALL プロシージャ・ステートメントの各パラメーターをバインドします。

注: 各パラメーターが (SQL_PARAM_INPUT、SQL_PARAM_OUTPUT、または SQL_PARAM_INPUT_OUTPUT に対して) 正しくバインドされたことを確認します。正しくバインドされていないと、CALL プロシージャ・ステートメントが実行される時に、予期しない結果が生じる可能性があります。たとえば、入力パラメーターが、SQL_PARAM_OUTPUT の *InputOutputType* を使用して、不正確にバインドされる場合に、このことが生じます。

- SQLExecDirect() を使用して CALL プロシージャ・ステートメントを実行するか、ステップ 4 で CALL プロシージャ・ステートメントを準備済みの場合には、SQLExecute() を使用して実行します。

注: ストアード・プロシージャを呼び出したアプリケーションかスレッドが、そのストアード・プロシージャの完了前に終了する場合、ストアード・プロシージャの実行も終了します。ストアード・プロシージャが早めに終了してしまう場合にも、データベースは一貫した状態と望ましい状態を保つようなロジックを、そのストアード・プロシージャに含めることは大切です。

- 関数が戻される時に SQLExecDirect() または SQLExecute() の戻りコードを調べ、CALL プロシージャ・ステートメントまたはストアード・プロシージャのいずれかの実行時に、何らかのエラーが発生していないかを判別します。戻りコードが SQL_SUCCESS_WITH_INFO か SQL_ERROR である場合、CLI 診断関数 SQLGetDiagRec() および SQLGetDiagField() を使用して、エラーが発生した理由を判別します。

ストアード・プロシージャを正常に実行した場合、OUT パラメーターとしてバインドされた変数には、そのストアード・プロシージャが CLI アプリケーションに戻したデータが含まれる可能性があります。該当する場合には、ストアード・プロシ

ージャーは、スクロール不可カーソルを使用して、1 つ以上の結果セットを戻す場合もあります。CLI アプリケーションでは、SELECT ステートメントの実行によって生成された結果セットを処理するときに、ストアード・プロシージャの結果セットを処理する必要があります。

注: CLI アプリケーションが、ストアード・プロシージャによって戻された結果セットに示された、パラメーターの番号またはタイプが分からない場合、その結果セットに対して、SQLNumResultCols()、SQLDescribeCol()、および SQLColAttribute() 関数を (この順序で) 呼び出して、この情報を判別することができます。

CALL ステートメントを実行したら、該当する場合には、ストアード・プロシージャから結果セットを検索できます。

注:

CLI/ODBC ドライバーは、CLI/ODBC アプリケーションが、データベースに対して SQL を初めて実行した時に、ユーザーに適切な特権または許可がある場合、CLI パッケージを自動バインドします。CLI パッケージの自動バインドは、ストアード・プロシージャ内から実行できません。そのため、アプリケーションが最初に行うことが CLI ストアード・プロシージャの呼び出しである場合、この自動バインドは起きません。新しい DB2 データベースに対して CLI ストアード・プロシージャを呼び出す CLI アプリケーションを実行する前に、以下のコマンドで、一度 CLI パッケージをバインドする必要があります。

UNIX

```
db2 bind <BNDPATH>/@db2cli.lst blocking all
```

Windows

```
db2bind "%DB2PATH%\%bnd%\@db2cli.lst" blocking
```

お勧めするアプローチとしては、データベースが、実行時に自動バインドするのを回避するために作成される時に、このパッケージを常にバインドする方法があります。自動バインドは、ユーザーに特権がない場合、または別のアプリケーションが同時に自動バインドを行っている場合には失敗します。

コミット動作:

DB2 サーバーで実行されている DB2 CLI クライアント・アプリケーションとコールされたストアード・プロシージャの両方での SQL ステートメントのコミット動作は、そのアプリケーションおよびストアード・プロシージャで適用されるコミットの組み合わせによります。可能な組み合わせおよび、その結果のコミット動作が、以下の表に説明されています。

表 186. DB2 CLI ストアード・プロシージャー・コミット動作

CLI クライアント	ストアード・プロシージャー	コミット動作
自動コミット ON	自動コミット ON	ストアード・プロシージャー内の正常に実行されたすべての SQL ステートメントは、ストアード・プロシージャー内の他の SQL ステートメントが失敗し、CALL ステートメントにエラーまたは警告の SQLCODE が返された場合でも、コミットされます。
自動コミット ON	自動コミット OFF	ストアード・プロシージャーが SQLCODE 0 を返した場合、ストアード・プロシージャー内のすべての正常に実行された SQL ステートメントはコミットされます。そうでない場合、ストアード・プロシージャー内のすべての SQL ステートメントはロールバックされます。
自動コミット ON	マニュアル・コミット	<p>手動でコミットされた、ストアード・プロシージャー内の正常に実行されたすべての SQL ステートメントは、CALL ステートメントにエラーまたは警告の SQLCODE が戻された場合でも、ロールバックされません。</p> <p>注: ストアード・プロシージャーが SQLCODE 0 を戻した場合、最後の手動コミットの後に発生したストアード・プロシージャー内のすべての正常に実行された SQL ステートメントは、コミットされます。そうでない場合は、手動コミット時点までロールバックされます。</p>
自動コミット OFF	自動コミット ON	<p>ストアード・プロシージャー内の正常に実行されたすべての SQL ステートメントは、CALL ステートメントにエラーまたは警告の SQLCODE が戻された場合でもコミットされ、ロールバックはされません。さらに、CALL ステートメントを含む、それまでの CLI クライアント・アプリケーション内の、正常に実行されコミットされていないすべての SQL ステートメントはコミットされます。</p> <p>注: CALL ステートメントの実行後は、トランザクションを完全にロールバックすることはできないため、このコミットの組み合わせをマルチ SQL ステートメント・クライアント・サイド・トランザクションで使用する場合は、注意してください。</p>
自動コミット OFF	自動コミット OFF	ストアード・プロシージャーが SQLCODE 0 を戻した場合、ストアード・プロシージャー内のすべての正常に実行された SQL ステートメントは、CALL ステートメントを含むトランザクションがコミットされるとコミットされます。そうでない場合、ストアード・プロシージャー内のすべての SQL ステートメントは、CALL ステートメントを含むトランザクションがロールバックされたときにロールバックされます。

表 186. DB2 CLI ストアード・プロシージャー・コミット動作 (続き)

CLI クライアント	ストアード・ プロシージャー	コミット動作
自動コミット OFF	マニュアル・コ ミット	<p>手動でコミットされた、ストアード・プロシージャー内の正常に実行されたすべての SQL ステートメントは、CALL ステートメントにエラーまたは警告の SQLCODE が戻された場合でも、ロールバックされません。さらに、CALL ステートメントまでの、CLI クライアント・アプリケーション内のすべての正常に実行された、コミットされていない SQL ステートメントはコミットされます。</p> <p>注: ストアード・プロシージャーが SQLCODE 0 を返した場合、最後の手動コミットの後に発生したストアード・プロシージャー内のすべての正常に実行された SQL ステートメントは、コミットされます。そうでない場合は、手動コミット時点でロールバックされます。</p> <p>注: CALL ステートメントの実行後は、トランザクションを完全にロールバックすることはできないため、このコミットの組み合わせをマルチ SQL ステートメント・クライアント・サイド・トランザクションで使用する場合は、注意してください。</p>

関連概念:

- アプリケーション開発ガイド クライアント・アプリケーションのプログラミングの『DB2 ストアード・プロシージャー』

関連タスク:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI アプリケーションの初期設定』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI 環境のセットアップ』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI アプリケーションでの SQL ステートメントの準備と実行』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI アプリケーションでのパラメーター・マーカのバインド』

関連資料:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻の『SQLExecDirect 関数 (CLI) - ステートメントの直接実行』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻の『SQLExecute 関数 (CLI) - ステートメントの実行』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻の『SQLGetDiagField 関数 (CLI) - 診断データ・フィールドの取得』

- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『SQLGetDiagRec 関数 (CLI) - 診断レコードの複数のフィールド設定の取得』
- SQL リファレンス 第 2 巻 の『CALL ステートメント』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『SQLBindParameter 関数 (CLI) - バッファまたは LOB ロケータへのパラメーター・マーカのバインド』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『CLI 関数戻りコード』

関連サンプル:

- 『spcall.c -- Call individual stored procedures (CLI)』
- 『spclient.c -- Call various stored procedures (CLI)』
- 『spclires.c -- Contrast stored procedure multiple result set handling methods (CLI)』
- 『spserver.c -- Definition of various types of stored procedures (CLI)』

Windows CLI ルーチンのコンパイルおよびリンク・オプション

Microsoft Visual C++ コンパイラーを使用して CLI ルーチンを構築する場合は、DB2 ではこの種のコンパイルおよびリンク・オプションが推奨されています。これらのオプションは、`sqllib\samples\cli\bldrtn.bat` バッチ・ファイル中に例示されています。

bldrtn のコンパイルおよびリンク・オプション	
コンパイル・オプション	
%BLDCOMP%	コンパイラー用の変数。デフォルトは、 <code>cl</code> (Microsoft Visual C++ コンパイラー) です。またこれは、 <code>icl</code> (32 ビット・アプリケーション用の Intel C++ コンパイラー)、または <code>ec1</code> (Itanium 64 ビット・アプリケーション用の Intel C++ コンパイラー) に設定することもできます。
-Zi	デバッグ情報を使用可能にします。
-Od	最適化なし。最適化をオフにしてデバッガーを使用する方が簡単です。
-c	コンパイルのみを実行し、リンクは実行しません。このバッチ・ファイルでは、コンパイルとリンクは別個のステップです。
-W2	警告レベルを設定します。
-DWIN32	Windows オペレーティング・システムに必要なコンパイラー・オプション。
-MDd	MSVCRT.LIB を使用してマルチスレッド DLL を作成します。 注: <code>-debug</code> オプションを使用しない場合は、代わりに <code>-MD</code> を使用してください。

bldrtn のコンパイルおよびリンク・オプション

リンク・オプション

link 32 ビットのリンカーを使用します。

-debug デバッグ情報を組み込みます。

-OUT:%1.dll

.DLL ファイルを作成します。

%1.obj オブジェクト・ファイルを組み込みます。

utilcli.obj

エラー・チェック用のユーティリティ・オブジェクト・ファイルを組み込みます。

db2cli.lib

DB2 CLI ライブラリーとリンクします。

-def:%1.def

モジュール定義ファイルを使用します。

他のコンパイラー・オプションについては、コンパイラーの資料をご覧ください。

関連タスク:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『Windows での CLI アプリケーションの作成』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『Windows での CLI ルーチンの作成』

関連サンプル:

- 『bldrtn.bat -- Builds C routines (stored procedures and UDFs) on Windows』

CLI での SQL から C へのデータ変換例

指定の SQL データ・タイプについて、以下の内容がリストされています。

- 表の最初の列には、SQLBindCol() および SQLGetData() の *fCType* 引き数の有効な入力値をリストします。
- 2 番目の列では、テストの出力をリストします。このテストでは SQLBindCol() または SQLGetData() に指定されている *cbValueMax* 引き数が頻繁に使用されます。ドライバーはこのテストを実行して、データを変換できるかどうかを判別します。
- 3 番目と 4 番目の列では、ドライバーがデータ変換を試行した後の、SQLBindCol() または SQLGetData() に指定されている *rgbValue* 引き数と *pcbValue* 引き数の値を(出力ごとに) リストします。

- 最後の列では、SQLFetch()、SQLExtendedFetch()、SQLGetData()、または SQLGetSubString() によって各出力用に戻される SQLSTATE をリストします。

表では、指定の SQL データ・タイプにとって有効になるよう ODBC で定義された変換をリストします。

SQLBindCol() または SQLGetData() の *fCType* 引数に、所定の SQL データ・タイプについて表にない値が含まれていると、SQLFetch() または SQLGetData() は、SQLSTATE 07006 (データ・タイプ属性制約違反) を戻します。

fCType 引数に、表にはあっても、ドライバーでサポートされていない変換を指定する値が含まれていると、SQLFetch() または SQLGetData() は、SQLSTATE HYC00 (ドライバー不可) を戻します。

表には示されていませんが、*pcbValue* 引数には、SQL データ値が NULL のときの SQL_NULL_DATA が含まれます。データを取り出す場合に複数呼び出しが行われる際の *pcbValue* の使用方法に関する説明は、SQLGetData() を参照してください。

SQL データが文字 C データに変換される際に、*pcbValue* に戻される文字カウントには NULL 終了バイトは含まれません。*rgbValue* が NULL ポインターの場合、SQLBindCol() または SQLGetData() は、SQLSTATE HY009 (無効な引数値) を戻します。

表では、以下の用語が使われています。

データの長さ

指定の C データ・タイプに変換された後のデータの全長 (データがストリングに変換された場合の NULL 終了バイトを除く)。これは、アプリケーションに戻される前にデータが切り捨てられる場合にも当てはまります。

有効桁 負符号 (必要な場合) および小数点の左の桁。

表示サイズ

文字形式でデータを表示するのに必要なバイトの合計数。

文字 SQL データから C データへの変換:

文字 SQL データ・タイプは以下のとおりです。

SQL_CHAR
SQL_VARCHAR
SQL_LONGVARCHAR
SQL_CLOB

表 187. 文字 SQL データから C データへの変換

fCType	テスト	rgbValue	pcbValue	SQLSTATE
SQL_C_CHAR	データ長 < cbValueMax	データ	データの長さ	00000
	データ長 >= cbValueMax	切り捨てデータ	データの長さ	01004
SQL_C_BINARY	データ長 <= cbValueMax	データ	データの長さ	00000
	データ長 > cbValueMax	切り捨てデータ	データの長さ	01004
SQL_C_SHORT	切り捨てられずに変換されたデータ ^a	データ	C データ・タイプのサイズ	00000
SQL_C_LONG		データ	C データ・タイプのサイズ	01004
SQL_C_FLOAT	変換され切り捨てられたデータ、ただし有効桁は失われていない ^a	データ	C データ・タイプのサイズ	01004
SQL_C_TINYINT		データ	C データ・タイプのサイズ	22003
SQL_C_BIT	データの変換で、有効桁が失われる ^a	影響なし	C データ・タイプのサイズ	22003
SQL_C_UBIGINT		影響なし	C データ・タイプのサイズ	22005
SQL_C_SBIGINT	データは数値でない ^a	影響なし	C データ・タイプのサイズ	22005
SQL_C_NUMERIC ^c		影響なし	C データ・タイプのサイズ	22005
SQL_C_DATE	データ値は有効な時刻 ^a	データ	6 ^b	00000
	データ値は有効な時刻ではない ^a	影響なし	6 ^b	22007
SQL_C_TIME	データ値は有効な時刻 ^a	データ	6 ^b	00000
	データ値は有効な時刻ではない ^a	影響なし	6 ^b	22007
SQL_C_TIMESTAMP	データ値は有効なタイムスタンプ ^a	データ	16 ^b	00000
	データ値は有効なタイムスタンプではない ^a	影響なし	16 ^b	22007

注:

- ^a *cbValueMax* の値はこの変換では無視されます。ドライバーは、*rgbValue* のサイズは C データ・タイプのサイズであると想定します。
- ^b これは、対応する C データ・タイプのサイズです。
- ^c SQL_C_NUMERIC は Windows プラットフォーム上でのみサポートされます。

SQLSTATE 00000 は SQLError() では戻されません。これは、関数が SQL_SUCCESS を戻すときに示されます。

グラフィック SQL データから C データへの変換:

グラフィック SQL データ・タイプは、以下のとおりです。

SQL_GRAPHIC
 SQL_VARGRAPHIC
 SQL_LONGVARGRAPHIC
 SQL_DBCLOB

表 188. グラフィック SQL データから C データへの変換

fCType	テスト	rgbValue	pcbValue	SQLSTATE
SQL_C_CHAR	2 バイト文字数 * 2 ≤ cbValueMax	データ	データの長さ (オクテット)	00000
	2 バイト文字数 * 2 > cbValueMax	cbValueMax 未満の最大 偶数バイト に切り捨て られたデータ	データの長さ (オクテット)	01004
SQL_C_DBCHAR	2 バイト文字数 * 2 < cbValueMax	データ	データの長さ (オクテット)	00000
	2 バイト文字数 * 2 >= cbValueMax	cbValueMax 未満の最大 偶数バイト に切り捨て られたデータ	データの長さ (オクテット)	01004

注: SQLSTATE 00000 は `SQLError()` では戻されません。これは、関数が `SQL_SUCCESS` を戻すときに示されます。

浮動小数点値への変換時に、結果値の非有効桁が失われていると、SQLSTATE 22003 は戻されません。

数値 SQL データから C データへの変換:

SQL 数値データ・タイプは、以下のとおりです。

SQL_DECIMAL
 SQL_NUMERIC
 SQL_SMALLINT

SQL_INTEGER
 SQL_BIGINT
 SQL_REAL
 SQL_FLOAT
 SQL_DOUBLE

表 189. 数値 SQL データから C データへの変換

fCType	テスト	rgbValue	pcbValue	SQLSTATE
SQL_C_CHAR	表示サイズ < cbValueMax	データ	データの長さ	00000
	有効桁数 < cbValueMax	切り捨てデータ	データの長さ	01004
	有効桁数 >= cbValueMax	影響なし	データの長さ	22003
SQL_C_SHORT	切り捨てられずに変換されたデータ ^a	データ	C データ・タイプのサイズ	00000
SQL_C_LONG				
SQL_C_FLOAT				
SQL_C_DOUBLE	変換され切り捨てられたデータ、ただし有効桁は失われていない ^a	切り捨てデータ	C データ・タイプのサイズ	01004
SQL_C_TINYINT				
SQL_C_BIT				
SQL_C_UBIGINT	データの変換で、有効桁が失われる ^a	影響なし	C データ・タイプのサイズ	22003
SQL_C_SBIGINT				
SQL_C_NUMERIC ^b				

注:

- ^a *cbValueMax* の値はこの変換では無視されます。ドライバーは、*rgbValue* のサイズは C データ・タイプのサイズであると想定します。
- ^b SQL_C_NUMERIC は Windows プラットフォーム上でのみサポートされます。

SQLSTATE 00000 は SQLError() では戻されません。これは、関数が SQL_SUCCESS を返すときに示されます。

バイナリー SQL データから C データへの変換:

バイナリー SQL データ・タイプは以下のとおりです。

SQL_BINARY
 SQL_VARBINARY
 SQL_LONGVARBINARY
 SQL_BLOB

表 190. バイナリー SQL データから C データへの変換

fCType	テスト	rgbValue	pcbValue	SQLSTATE
SQL_C_CHAR	(データ長) < cbValueMax	データ	データの 長さ	なし
	(データ長) >= cbValueMax	切り捨てデ ータ	データの 長さ	01004
SQL_C_BINARY	データ長 <= cbValueMax	データ	データの 長さ	なし
	データ長 > cbValueMax	切り捨てデ ータ	データの 長さ	01004

日付 SQL データから C データへの変換:

日付 SQL データ・タイプは以下のとおりです。

SQL_DATE

表 191. 日付 SQL データから C データへの変換

fCType	テスト	rgbValue	pcbValue	SQLSTATE
SQL_C_CHAR	cbValueMax >= 11	データ	10	00000
	cbValueMax < 11	影響なし	10	22003
SQL_C_DATE	なし ^a	データ	6 ^b	00000
SQL_C_TIMESTAMP	なし ^a	データ ^c	16 ^b	00000

注:

- ^a *cbValueMax* の値はこの変換では無視されます。ドライバーは、*rgbValue* のサイズは C データ・タイプのサイズであると想定します。これは、対応する C データ・タイプのサイズです。
- ^b これは、対応する C データ・タイプのサイズです。
- ^c `TIMESTAMP_STRUCT` 構造の時刻フィールドはゼロに設定されます。

SQLSTATE **00000** は `SQLError()` では戻されません。これは、関数が `SQL_SUCCESS` を戻すときに示されます。

日付 SQL データ・タイプが文字の C データ・タイプに変換される場合、ストリングは「yyyy-mm-dd」形式になります。

時刻 SQL データから C データへの変換:

時刻 SQL データ・タイプは以下のとおりです。

SQL_TIME

表 192. 時刻 SQL データから C データへの変換

fCType	テスト	rgbValue	pcbValue	SQLSTATE
SQL_C_CHAR	cbValueMax >= 9	データ	8	00000
	cbValueMax < 9	影響なし	8	22003
SQL_C_TIME	なし ^a	データ	6 ^b	00000
SQL_C_TIMESTAMP	なし ^a	データ ^c	16 ^b	00000

注:

- ^a *cbValueMax* の値はこの変換では無視されます。ドライバーは、*rgbValue* のサイズは C データ・タイプのサイズであると想定します。
- ^b これは、対応する C データ・タイプのサイズです。
- ^c `TIMESTAMP_STRUCT` 構造の日付フィールドは、アプリケーションが実行しているマシンの現行システム日付に設定され、時刻小数部はゼロに設定されます。

SQLSTATE `00000` は `SQLError()` では戻されません。これは、関数が `SQL_SUCCESS` を戻すときに示されます。

時刻 SQL データ・タイプが文字の C データ・タイプに変換される場合、ストリングは「hh:mm:ss」形式になります。

タイム・スタンプ SQL データから C データへの変換:

タイム・スタンプ SQL データ・タイプは以下のとおりです。

`SQL_TIMESTAMP`

表 193. タイム・スタンプ SQL データから C データへの変換

fCType	テスト	rgbValue	pcbValue	SQLSTATE
SQL_C_CHAR	表示サイズ < cbValueMax	データ	データの長さ	00000
	19 <= cbValueMax <= 表示サイズ	切り捨てデータ ^b	データの長さ	01004
	cbValueMax < 19	影響なし	データの長さ	22003
SQL_C_DATE	なし ^a	切り捨てデータ ^c	6 ^e	01004
SQL_C_TIME	なし ^a	切り捨てデータ ^d	6 ^e	01004
SQL_C_TIMESTAMP	なし ^a	データ	16 ^e	00000

注:

- ^a cbValueMax の値はこの変換では無視されます。ドライバーは、rgbValue のサイズは C データ・タイプのサイズであると想定します。
- ^b タイム・スタンプの小数秒は切り捨てられます。
- ^c タイム・スタンプの時刻部分は削除されます。
- ^d タイム・スタンプの日付部分は削除されます。
- ^e これは、対応する C データ・タイプのサイズです。

SQLSTATE 00000 は SQL_Error() では戻されません。これは、関数が SQL_SUCCESS を戻すときに示されます。

タイム・スタンプ SQL データ・タイプが文字の C データ・タイプに変換される場合、
 スtringは「yyyy-mm-dd hh:mm:ss.ffffff」形式になります。アプリケーションに ISO
 フォーマットが必要な場合、CLI/ODBC 構成キーワード PATCH2=33 に設定します。

SQL から C へのデータ変換例:

表 194. SQL から C へのデータ変換例

SQL データ・タイプ	SQL データ値	C データ・タイプ	cbValue Max	rgbValue	SQL STATE
SQL_CHAR	abcdef	SQL_C_CHAR	7	abcde¥0 ^a	00000
SQL_CHAR	abcdef	SQL_C_CHAR	6	abcde¥0 ^a	01004
SQL_DECIMAL	1234.56	SQL_C_CHAR	8	1234.56¥0 ^a	00000
SQL_DECIMAL	1234.56	SQL_C_CHAR	5	1234¥0 ^a	01004
SQL_DECIMAL	1234.56	SQL_C_CHAR	4	---	22003

表 194. SQL から C へのデータ変換例 (続き)

SQL データ・ タイプ	SQL データ値	C データ・ タイプ	cbValue Max	rgbValue	SQL STATE
SQL_DECIMAL	1234.56	SQL_C_FLOAT	無視され ます	1234.56	00000
SQL_DECIMAL	1234.56	SQL_C_SHORT	無視され ます	1234	01004
SQL_DATE	1992-12-31	SQL_C_CHAR	11	1992-12-31¥0 ^a	00000
SQL_DATE	1992-12-31	SQL_C_CHAR	10	---	22003
SQL_DATE	1992-12-31	SQL_C_ TIMESTAMP	無視され ます	1992,12,31, 0,0,0,0 ^b	00000
SQL_TIMESTAMP	1992-12-31 23:45:55.12	SQL_C_CHAR	23	1992-12-31 23:45:55.12¥0 ^a	00000
SQL_TIMESTAMP	1992-12-31 23:45:55.12	SQL_C_CHAR	22	1992-12-31 23:45:55.1¥0 ^a	01004
SQL_TIMESTAMP	1992-12-31 23:45:55.12	SQL_C_CHAR	18	---	22003

注:

- ^a 「¥0」は NULL 終了文字を表します。
- ^b このリストの数値は、TIMESTAMP_STRUCT 構造体のフィールドに保管される数値です。

SQLSTATE 00000 は SQLError() では戻されません。これは、関数が SQL_SUCCESS を戻すときに示されます。

関連資料:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI アプリケーション用の SQL 記号データ・タイプおよびデフォルト・データ・タイプ』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI アプリケーション用の C データ・タイプ』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI でサポートされているデータ変換』
- 853 ページの『CLI での C から SQL へのデータ変換例』
- 873 ページの『PATCH2 CLI/ODBC 構成キーワード』

CLI での C から SQL へのデータ変換例

指定の C データ・タイプについて、以下の内容がリストされています。

- 表の最初の列には、SQLBindParameter() または SQLSetParam() の *fSqlType* 引き数の有効な入力値をリストします。
- 2 番目の列では、テストの出力をリストします。このテストでは SQLBindParameter() または SQLSetParam の *pcbValue* 引き数で指定されたパラメータ・データの長さが頻繁に使用されます。ドライバーはこのテストを実行して、データを変換できるかどうかを判別します。
- 3 番目の列では、SQLExecDirect() または SQLExecute() によって各出力に返される SQLSTATE をリストします。

表では、指定の SQL データ・タイプにとって有効になるよう ODBC で定義された変換をリストします。

SQLBindParameter() または SQLSetParam() の *fSqlType* 引き数に、指定の C データ・タイプについて表にない値が含まれている場合、SQLSTATE 07006 (データ・タイプ属性制約違反) が戻されます。

fSqlType 引き数に、表にはあっても、ドライバーでサポートされていない変換を指定する値が含まれていると、SQLBindParameter() または SQLSetParam() は、SQLSTATE HYC00 (ドライバー不可) を戻します。

SQLBindParameter() または SQLSetParam() に指定された *rgbValue* および *pcbValue* 引き数が両方とも NULL ポインタである場合、その関数は SQLSTATE HY009 (引き数値が無効です) を戻します。

データの長さ

指定された SQL データ・タイプに変換された後のデータの全長 (データがストリングに変換された場合の NULL 終了バイトを除く)。これは、データ・ソースに送られる前にデータが切り捨てられる場合にも当てはまります。

列の長さ

データがデフォルト C データ・タイプへ転送されるときにアプリケーションに戻されるバイトの最大数。文字データの場合、長さには NULL 終了バイトは含まれません。

表示サイズ

データを文字書式で表示するために必要な最大バイト数。

有効桁 負符号 (必要な場合) および小数点の左の桁。

文字 C データから SQL データへの変換:

文字 C データ・タイプは以下のとおりです。

SQL_C_CHAR

表 195. 文字 C データから SQL データへの変換

fSQLType	テスト	SQLSTATE
SQL_CHAR	データ長 <= 列長	N/A
SQL_VARCHAR		
SQL_LONGVARCHAR	データ長 > 列長	22001
SQL_CLOB		
SQL_DECIMAL	切り捨てられずに変換されたデータ	なし
SQL_NUMERIC		
SQL_SMALLINT	変換され切り捨てられたデータ、ただし有効桁は失われていない	22001
SQL_INTEGER		
SQL_REAL	データの変換で有効桁が失われる	22003
SQL_FLOAT		
SQL_DOUBLE	データ値は数値ではない	22005
SQL_BINARY	(データ長) < 列長	なし
SQL_VARBINARY		
SQL_LONGVARBINARY	(データ長) >= 列長	22001
SQL_BLOB	データ値は 16 進値ではない	22005
SQL_DATE	データ値は有効な日付	なし
	データ値は有効な日付ではない	22007
SQL_TIME	データ値は有効な時刻	なし
	データ値は有効な時刻ではない	22007
SQL_TIMESTAMP	データ値は有効なタイム・スタンプ	なし
	データ値は有効なタイム・スタンプではない	22007
SQL_GRAPHIC	データ長 / 2 <= 列長	なし
SQL_VARGRAPHIC		
SQL_LONGVARGRAPHIC	データ長 / 2 < 列長	22001
SQL_DBCLOB		

注: SQLSTATE 00000 は SQLError() では戻されません。これは、関数が SQL_SUCCESS を戻すときに示されます。

数値 C データから SQL データへの変換:

数値 C データ・タイプは以下のとおりです。

- SQL_C_SHORT
- SQL_C_LONG
- SQL_C_FLOAT
- SQL_C_DOUBLE
- SQL_C_TINYINT

SQL_C_BIT

表 196. 数値 C データから SQL データへの変換

fSQLType	テスト	SQLSTATE
SQL_DECIMAL	切り捨てられずに変換されたデータ	なし
SQL_NUMERIC		
SQL_SMALLINT	変換され切り捨てられたデータ、ただし有効桁は失われていない	22001
SQL_INTEGER		
SQL_REAL		
SQL_FLOAT	データの変換で有効桁が失われる	22003
SQL_DOUBLE		
SQL_CHAR	切り捨てられずに変換されたデータ	なし
SQL_VARCHAR	データの変換で有効桁が失われる	22003

注: SQLSTATE 00000 は `SQLError()` では戻されません。これは、関数が `SQL_SUCCESS` を戻すときに示されます。

浮動小数点値への変換時に、結果値の非有効桁が失われていると、SQLSTATE 22003 は戻されません。

バイナリー C データから SQL データへの変換:

バイナリー C データ・タイプは以下のとおりです。

SQL_C_BINARY

表 197. バイナリー C データから SQL データへの変換

fSQLType	テスト	SQLSTATE
SQL_CHAR	データ長 <= 列長	なし
SQL_VARCHAR		
SQL_LONGVARCHAR	データ長 > 列長	22001
SQL_CLOB		
SQL_BINARY	データ長 <= 列長	なし
SQL_VARBINARY		
SQL_LONGVARBINARY	データ長 > 列長	22001
SQL_BLOB		

DBCHAR C データから SQL データへの変換:

2 バイト C データ・タイプは以下のとおりです。

SQL_C_DBCHAR

表 198. DBCHAR C データから SQL データへの変換

fSQLType	テスト	SQLSTATE
SQL_CHAR	データ長 <= 列長 x 2	なし
SQL_VARCHAR		
SQL_LONGVARCHAR	データ長 > 列長 x 2	22001
SQL_CLOB		
SQL_BINARY	データ長 <= 列長 x 2	なし
SQL_VARBINARY		
SQL_LONGVARBINARY	データ長 > 列長 x 2	22001
SQL_BLOB		

日付 C データから SQL データへの変換:

日付 C データ・タイプは以下のとおりです。

SQL_C_DATE

表 199. 日付 C データから SQL データへの変換

fSQLType	テスト	SQLSTATE
SQL_CHAR	列長 >= 10	なし
SQL_VARCHAR	列長 < 10	22003
SQL_DATE	データ値は有効な日付	なし
	データ値は有効な日付ではない	22007
SQL_TIMESTAMP ^a	データ値は有効な日付	なし
	データ値は有効な日付ではない	22007

注: SQLSTATE 00000 は SQLError() では戻されません。これは、関数が SQL_SUCCESS を戻すときに示されます。

注: a、TIMESTAMP の時刻コンポーネントはゼロに設定されます。

時刻 C データから SQL データへの変換:

時刻 C データ・タイプは以下のとおりです。

SQL_C_TIME

表 200. 時刻 C データから SQL データへの変換

fSQLType	テスト	SQLSTATE
SQL_CHAR	列長 >= 8	なし
SQL_VARCHAR	列長 < 8	22003
SQL_TIME	データ値は有効な時刻	なし
	データ値は有効な時刻ではない	22007
SQL_TIMESTAMP ^a	データ値は有効な時刻	なし
	データ値は有効な時刻ではない	22007

注: SQLSTATE **00000** は `SQLERROR()` では戻されません。これは、関数が `SQL_SUCCESS` を戻すときに示されます。

注: **a**、TIMESTAMP の日付コンポーネントは、アプリケーションが実行しているマシンのシステム日付に設定されます。

タイム・スタンプ C データから SQL データへの変換:

タイム・スタンプ C データ・タイプは以下のとおりです。

SQL_C_TIMESTAMP

表 201. タイム・スタンプ C データから SQL データへの変換

fSQLType	テスト	SQLSTATE
SQL_CHAR	列長 >= 表示サイズ	なし
SQL_VARCHAR	19 <= 列長 < 表示サイズ ^a	22001
	列長 < 19	22003
SQL_DATE	時刻フィールドがゼロ	なし
	時刻フィールドがゼロ以外	22008
	データ値に有効な日付が含まれていない ^b	22007
SQL_TIME	小数秒フィールドがゼロ	なし
	小数秒フィールドがゼロ以外	22008
	データ値に有効な時刻が含まれていない	22007
SQL_TIMESTAMP	データ値は有効なタイム・スタンプ	なし
	データ値は有効なタイム・スタンプではない	22007

注:

^a タイム・スタンプの小数秒は切り捨てられます。

^b timestamp_struct は、時間、分、秒および小数部を 0 にリセットする必要があります。そうでない場合は、SQLSTATE 22007 が返されます。

SQLSTATE 00000 は SQLError() では戻されません。これは、関数が SQL_SUCCESS を戻すときに示されます。

C から SQL へのデータ変換例:

表 202. C から SQL へのデータ変換例

C データ・ タイプ	C データ 値	SQL データ・ タイプ	列の長さ	SQL データ値	SQL STATE
SQL_C_CHAR	abcdef¥0	SQL_CHAR	6	abcdef	N/A
SQL_C_CHAR	abcdef¥0	SQL_CHAR	5	abcde	22001
SQL_C_CHAR	1234.56¥0	SQL_DECIMAL	6	1234.56	なし
SQL_C_CHAR	1234.56¥0	SQL_DECIMAL	5	1234.5	22001
SQL_C_CHAR	1234.56¥0	SQL_DECIMAL	3	---	22003
SQL_C_CHAR	4.46.32	SQL_TIME	6	4.46.32	なし
SQL_C_CHAR	4-46-32	SQL_TIME	6	該当せず	22007
SQL_C_DOUBLE	123.45	SQL_CHAR	22	1.23450000 000000e+02	なし
SQL_C_FLOAT	1234.56	SQL_FLOAT	該当せず	1234.56	N/A
SQL_C_FLOAT	1234.56	SQL_INTEGER	該当せず	1234	22001

表 202. C から SQL へのデータ変換例 (続き)

C データ・ タイプ	C データ 値	SQL データ・ タイプ	列の長さ	SQL データ値	SQL STATE
SQL_C_ TIMESTAMP	1992-12-31 23:45:55. 123456	SQL_DATE	6	1992-12-31	01004

注: SQLSTATE 00000 は SQLError() では戻されません。これは、関数が SQL_SUCCESS を戻すときに示されます。

関連資料:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI アプリケーション用の SQL 記号データ・タイプおよびデフォルト・データ・タイプ』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI アプリケーション用の C データ・タイプ』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI でサポートされているデータ変換』
- 845 ページの『CLI での SQL から C へのデータ変換例』

CLI アプリケーションのストリングの処理

以下に示す規則によって、DB2 CLI 関数のストリング引き数のさまざまな面を取り扱います。

ストリング引き数の長さ:

入力ストリングは、関連した長さ引き数を保持できます。この引き数は、ストリングの正確な長さ (NULL 終止符を除く)、 NULL 終了ストリングを示す特殊値 SQL_NTS、または NULL 値を渡す SQL_NULL_DATA のうちのいずれかを示します。長さを SQL_NTS に設定すると、DB2 CLI は NULL 終止符を見つけてストリングの長さを判別します。

出力ストリングには、2 つの関連した長さ引き数があります。1 つは割り振られる出力バッファの長さを指定する入力長さ引き数で、もう 1 つは DB2 CLI が返したストリングの実際の長さを返す出力長さ引き数です。戻される長さの値は、戻りに使用できるストリングの全長です。それがバッファに適合するかどうかとは関係ありません。

SQL 列データの場合、出力が NULL であれば、SQL_NULL_DATA が長さ引き数に戻され、出力バッファは考慮されません。列の値が NULL 値の場合、記述子フィールド SQL_DESC_INDICATOR_PTR は SQL_NULL_DATA にセットされます。その他のフィールド設定を含む詳細については、記述子 FieldIdentifier の引き数値を参照してください。

出力長さ引き数に NULL ポインターを指定して関数が呼び出される場合、DB2 CLI は長さを戻しません。出力データが NULL 値であっても、DB2 CLI はその値が NULL 値であることを示すことはできません。結果セットの列に NULL 値が入る可能性があるときは、出力長さ引き数を指す有効なポインターを必ず指定する必要があります。有効な出力長さ引き数を必ず使用することを強くお勧めします。

パフォーマンスのヒント:

長さ引き数 (*StrLen_or_IndPtr*) と出力バッファ (*TargetValuePtr*) がメモリー内で隣接していると、DB2 CLI は両方の値をさらに効果的に返すことができ、アプリケーションのパフォーマンスは向上します。たとえば、以下の構造が定義されているとします。

```
struct
{
    SQLINTEGER pcbValue;
    SQLCHAR    rgbValue [BUFFER_SIZE];
} buffer;
```

さらに &buffer.pcbValue および buffer.rgbValue が SQLBindCol() に渡されると、DB2 CLI は 1 回の操作で両方の値を更新します。

ストリングの NULL 終了:

デフォルトでは、DB2 CLI が戻すすべての文字ストリングが NULL 終了文字 (16 進数 00) で終わります。ただし、図形および DBCLOB データ・タイプから SQL_C_CHAR アプリケーション変数へ戻されるストリングは除きます。SQL_C_DBCHAR アプリケーション変数に取り出される図形および DBCLOB データ・タイプは、2 バイトの NULL 終了文字で NULL 終了します。また、SQL_C_WCHAR に取り出されるストリング・データは、ユニコード NULL 終了文字 0x0000 で終了します。このためすべてのバッファが、予期される最大バイト数に NULL 終了文字を加えた値が入る大きさのスペースを割り振る必要があります。

また、SQLSetEnvAttr() を使用し、環境属性を設定して、可変長出力 (文字ストリング) データの NULL 終了を使用不能にすることもできます。この場合には、アプリケーションが予期される最長のストリングと同じ長さにバッファを正確に割り振ります。アプリケーションは、出力長さ引き数のストレージを指す有効なポインターを与えなければならず、これにより DB2 CLI は戻されるデータの実際の長さを示すことができます。こうしないと、アプリケーションにはこの長さを判別する方法が何もないことになります。DB2 CLI のデフォルトは、常に NULL 終了文字を書き込むことです。

PATCH1 CLI/ODBC 構成キーワードを使用すると、DB2 CLI に NULL 終了の図形および DBCLOB ストリングを挿入することが可能です。

ストリングの切り捨て:

出力ストリングがバッファに入りきらない場合、DB2 CLI はバッファのサイズにストリングを切り捨て、NULL 終了文字を書き込みます。切り捨てが行われると、関数

は `SQL_SUCCESS_WITH_INFO` と、切り捨てを示す `SQLSTATE 01004` を戻します。それからアプリケーションはバッファ長と出力長を比較して、どのストリングが切り捨てられたかを判別することができます。

たとえば、`SQLFetch()` が、`SQL_SUCCESS_WITH_INFO` と `SQLSTATE 01004` を戻す場合、列にバインドされたバッファのうち少なくとも 1 つが小さすぎてデータを保持できないということになります。列にバインドされたバッファごとに、アプリケーションはバッファ長と出力長を比較してどの列が切り捨てられたかを判別できます。また `SQLGetDiagField()` を呼び出して、どの列が失敗したかを検出することもできます。

ストリングの解釈:

通常、DB2 CLI はストリング引き数を大文字と小文字の区別をして解釈し、値からスペースをトリムすることはありません。1 つの例外は、`SQLSetCursorName()` 関数のカーソル名の入力引き数です。カーソル名が区切られ (二重引用符で囲まれ) ないと、前書きおよび後書きブランクが除去され、大文字小文字は無視されます。

ストリングのブランク埋め込み:

CHAR 列に格納されるデータは、固定列サイズになるまでブランクが埋め込まれます。データが CHAR 列から取り出された時、これらのブランクはデータの一部として戻されます。これまでは、DB2[®] バージョン 7 の CLI/ODBC ユニコード・アプリケーションは、コード・ページ変換のため、一貫性のある固定長を持たないストリングを返すことがありました。しかし DB2 バージョン 8 では、戻されるストリングはブランクで埋め込まれ、CHAR 列で定義されたとおりの固定長になります。

関連概念:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『DB2 CLI 用の SQLSTATES』

関連資料:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI アプリケーション用の SQL 記号データ・タイプおよびデフォルト・データ・タイプ』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI アプリケーション用の C データ・タイプ』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻の『SQLFetch 関数 (CLI) - 次の行の取り出し』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻の『SQLGetDiagField 関数 (CLI) - 診断データ・フィールドの取得』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻の『SQLSetEnvAttr 関数 (CLI) - 環境属性の設定』
- 819 ページの『記述子ヘッダーとレコード・フィールドの初期設定値 (CLI)』

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『CLI 関数戻りコード』
- 872 ページの『PATCH1 CLI/ODBC 構成キーワード』

DB2 CLI のサンプル

表 203. サンプル CLI プログラム・ファイル

サンプル・プログラム名	プログラムの説明
チュートリアル・サンプル - データベースの基本操作の例を示すプログラム。	
tut_mod.c	表データの変更方法。
tut_read.c	表の読み方。
tut_use.c	データベースの使用方法。
インストール・イメージ・レベル - DB2 と CLI のインストール・イメージ・レベルを扱うサンプル。	
ilinfo.c	インストール・レベル情報 (CLI ドライバーのバージョンなど) の入手および設定方法。
クライアント・レベル - クライアント・レベルの DB2 を扱うサンプル。	
cli_info.c	クライアント・レベル情報の入手および設定方法。
clihandl.c	ハンドルの割り当ておよび解放方法。
clisqlca.c	SQLCA データの処理方法。
インスタンス・レベル - DB2 のインスタンス・レベルを扱うサンプル。	
ininfo.c	インスタンス・レベル情報の入手および設定方法。
データベース・レベル - DB2 内のデータベース・オブジェクトを扱うサンプル。	
dbcongui.c	グラフィカル・ユーザー・インターフェース (GUI) にデータベースを接続する方法。
dbconn.c	データベースからの接続および切断方法。
dbinfo.c	データベース・レベルの情報の入手および設定方法。
dbmcon.c	データベースからの接続および切断方法。
dbmconx.c	組み込み SQL をもった複数のデータベースへの接続および切断方法。
dbmconx1.h	dbmconx1.sqc 用のヘッダー・ファイル。
dbmconx1.sqc	dbmconx プログラム用の組み込み SQL ファイル。
dbmconx2.h	dbmconx2.sqc 用のヘッダー・ファイル。
dbmconx2.sqc	dbmconx プログラム用の組み込み SQL ファイル。
dbnative.c	ODBC エスケープ文節の入ったステートメントを、データ・ソース特有の形式に変換する方法。
dbuse.c	データベース・オブジェクトの使用方法。

表 203. サンプル CLI プログラム・ファイル (続き)

サンプル・プログラム名	プログラムの説明
dbusemx.sqc	組み込み SQL を備えたデータベース・オブジェクトの使用方法。
表レベル - DB2 内の表オブジェクトを扱うサンプル。	
tbconstr.c	表の制約の処理方法。
tbcreate.c	表の作成、変更、およびドロップ方法。
tbinfo.c	表レベルの情報の入手および設定方法。
tbload.c	CLI LOAD ユーティリティを使用してデータを挿入する方法です。
tbmod.c	表内の情報の修正方法。
tbread.c	表内の情報の読み取り方法。
データ・タイプ・レベル - データ・タイプを扱うサンプル。	
dtinfo.c	データ・タイプに関する情報の入手方法。
dtlob.c	LOB データの読み取りおよび書き込み方法。
dtudt.c	ユーザー定義特殊タイプの作成、使用、およびドロップ方法。
ストアド・プロシージャ・レベル - ストアド・プロシージャを示すサンプル。	
spcat	spserver プログラムのストアド・プロシージャ・カタログ・スクリプト。このスクリプトは、spdrop.db2 と spcreate.db2 を呼び出します。
spcreate.db2	CREATE PROCEDURE ステートメントを発行するための CLP スクリプト。
spdrop.db2	カタログからストアド・プロシージャをドロップするための CLP スクリプト。
spclient.c	spserver.c 内で宣言されるサーバー関数を呼び出すために使用されるクライアント・プログラム。
spserver.c	サーバー上で構築および実行されるストアド・プロシージャ関数。
spclires.c	複数の結果セットの SQLMoreResults と SQLNextResults の相違を示すクライアント・アプリケーション。
spcall.c	任意のストアド・プロシージャを呼び出すためのクライアント・プログラム。
UDF レベル - ユーザー定義関数を示すサンプル。	
udfcli.c	udfsrv.c 内のユーザー定義関数を呼び出すクライアント・アプリケーション。
udfsrv.c	udfcli.c によって呼び出されるユーザー定義関数 ScalarUDF。
共通ユーティリティ・ファイル	
utilcli.c	CLI サンプルで使用されるユーティリティ関数。
utilcli.h	CLI サンプルで使用されるユーティリティ関数用のヘッダー・ファイル。

関連概念:

- アプリケーション開発ガイド アプリケーションの構築および実行 の『サンプル・ファイル』
- アプリケーション開発ガイド アプリケーションの構築および実行 の『サンプル・プログラム: 構造と設計』

HP-UX CLI アプリケーションのコンパイルおよびリンク・オプション

HP-UX C コンパイラーを使用して CLI アプリケーションを構築する場合は、DB2 ではこの種のコンパイルおよびリンク・オプションが推奨されています。これらは、`sqllib/samples/cli/blldapp` ビルド・スクリプトで例示されます。

blldapp のコンパイルおよびリンク・オプション	
コンパイル・オプション	
cc	C コンパイラーを使用します。
\$EXTRA_CFLAG	
	64 ビット・サポートが使用可能な場合、これには値 <code>+DA2.0W</code> が含まれます。 32 ビット・サポートでは、これに値は含まれません。
+DA2.0W (64 ビットのみ)	
	64 ビット・コードを生成する場合に使用する必要があります。
-Ae	HP ANSI 拡張モードを使用可能にします。
-I\$DB2PATH/include	
	DB2 組み込みファイルのロケーションを指定します。
-I\$DB2PATH/include	
	DB2 組み込みファイルのロケーションを指定します。たとえば、 <code>\$HOME/sqllib/include</code> 。
-c	コンパイルのみを実行し、リンクは実行しません。コンパイルとリンクは別個のステップです。

bldapp のコンパイルおよびリンク・オプション

リンク・オプション

cc コンパイラーをリンカーのフロントエンドとして使用します。

\$EXTRA_CFLAG

64 ビット・サポートが使用可能な場合、これには値 **+DA2.0W** が含まれます。
32 ビット・サポートでは、これに値は含まれません。

+DA2.0W (64 ビットのみ)

64 ビット・コードを生成する場合に使用する必要があります。

-o \$1 実行可能プログラムを指定します。

-o \$1.o

オブジェクト・ファイルを指定します。

utilcli.o

エラー・チェック用のユーティリティ・オブジェクト・ファイルを組み込みます。

-L\$DB2PATH/lib

DB2 ランタイム共有ライブラリーのロケーションを指定します。たとえば、
\$HOME/sqlllib/lib。

-ldb2 データベース・マネージャー・ライブラリーとリンクします。

他のコンパイラー・オプションについては、コンパイラーの資料をご覧ください。

関連タスク:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『UNIX での CLI アプリケーションの作成』

関連資料:

- 866 ページの『HP-UX CLI ルーチンのコンパイルおよびリンク・オプション』

関連サンプル:

- 『bldapp -- Builds HP-UX C applications』

HP-UX CLI ルーチンのコンパイルおよびリンク・オプション

HP-UX C コンパイラーを使用して CLI ルーチンを構築する場合は、DB2 ではこの種のコンパイルおよびリンク・オプションが推奨されています。これらは、
sqlllib/samples/cli/bldrtn ビルド・スクリプトで例示されます。

bldrtn のコンパイルおよびリンク・オプション

コンパイル・オプション

cc C コンパイラー。

\$EXTRA_CFLAG

64 ビット・サポートが使用可能な場合、これには値 **+DA2.0W** が含まれます。
32 ビット・サポートでは、これに値は含まれません。

+DA2.0W (64 ビットのみ)

64 ビット・コードを生成する場合に使用する必要があります。

+u1 位置合わせしないデータ・アクセスを認めます。アプリケーションが位置合わせしないデータを使用する場合にのみ使用します。

+z 位置に依存しないコードを生成します。

-Ae HP ANSI 拡張モードを使用可能にします。

-I\$DB2PATH/include

DB2 組み込みファイルのロケーションを指定します。たとえば、
\$HOME/sql1lib/include。

-D_POSIX_C_SOURCE=199506L

ルーチンを他のルーチンと同じプロセスで実行する (THREADSAFE) か、またはエンジンそのもので実行する (NOT FENCED) ときに必要な、
_REENTRANT が必ず定義されるようにするための POSIX スレッド・ライブラリー・オプション。

-c コンパイルのみを実行し、リンクは実行しません。コンパイルとリンクは別個のステップです。

リンク・オプション

ld リンクにリンカーを使用します。

-b 通常の実行可能ファイルではなく、共有ライブラリーを作成します。

-o \$1 実行可能ファイルを指定します。

\$1.o オブジェクト・ファイルを指定します。

utilcli.o

エラー・チェック・ユーティリティ・オブジェクト・ファイル中にリンクします。

-L\$DB2PATH/lib

DB2 ランタイム共有ライブラリーのロケーションを指定します。たとえば、
-L\$HOME/sql1lib/lib。-L オプションを指定しないと、コンパイラーはパスとして /usr/lib:/lib を想定します。

-ldb2 DB2 ライブラリーとリンクします。

-lpthread

POSIX スレッド・ライブラリーとリンクします。

他のコンパイラー・オプションについては、コンパイラーの資料をご覧ください。

関連タスク:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『UNIX での CLI ルーチンの作成』

関連資料:

- 865 ページの『HP-UX CLI アプリケーションのコンパイルおよびリンク・オプション』

関連サンプル:

- 『bldrtn -- Builds HP-UX C routines (stored procedures and UDFs)』

DB2 CLI のバインド・ファイルおよびパッケージ名

適切な特権または許可のある場合に、CLI/ODBC ドライバーは、変更されたりアンバインドされたりした CLI パッケージを必要な時点で自動的にバインドします。一般的に自動バインドは、データベースの作成後に最初に接続したときか、新しい DB2 クライアント・バージョンによってデータベースに最初に接続したときに起こります。管理者は、自動バインド・プロセス (一般的には CLI プログラムの初回実行時) を監視したり、必要なファイルを明示的にバインドできます。

表 204. DB2 CLI バインド・ファイルおよびパッケージ名

バインド・ファイル名	パッケージ名	DB2 Universal Database で必要?	ホスト・サーバーで必要?	説明
db2clipk.bnd	SYSSHxyy	はい	はい	動的プレースホルダー - スモール・パッケージ WITH HOLD
	SYSSNxyy	はい	はい	動的プレースホルダー - スモール・パッケージ NO WITH HOLD
	SYSLHxyy	はい	はい	動的プレースホルダー - ラージ・パッケージ WITH HOLD
	SYSLNxyy	はい	はい	動的プレースホルダー - ラージ・パッケージ NO WITH HOLD

表 204. DB2 CLI バインド・ファイルおよびパッケージ名 (続き)

バインド・ファイル名	パッケージ名	DB2 Universal Database で必要?	ホスト・サーバーで必要?	説明
db2clist.bnd	SYSSTAT	はい	はい	共通静的 CLI 関数
db2schema.bnd	SYSSCwww	はい	はい	カタログ関数サポート
db2cliws.bnd	SQLL65zz	サーバー・バージョン 2 ~ 7	いいえ	DB2 for Intel/UNIX カタログ関数サポート
db2cliv2.bnd	SQLL95zz	サーバー・バージョン 2 ~ 7	いいえ	共通静的 CLI 関数
<p>注:</p> <ul style="list-style-type: none"> • 'www' は 000 ~ ZZZ のパッケージ反復。つまり、個々の桁は 36 進数で、000、001、...、009、00A、00B、...、00Z、010、...、01Z、020、... です。 • 'x' 分離レベル: 0=NC、1=UR、2=CS、4=RS、8=RR。 • 'yy' は、パッケージ反復 00 ~ FF。 • 'zz' は、それぞれのプラットフォームでユニークの値。 <p>たとえば、動的パッケージの場合は、以下のようになります。</p> <ul style="list-style-type: none"> • SYSSN100: スモール・パッケージ (64 セクション)、カーソル宣言はすべて非保留カーソルが対象。分離レベル UR でバインドされます。これは、このパッケージの最初の反復です。 • SYSLH401: ラージ・パッケージ (384 セクション)、カーソル宣言はすべて保留カーソルが対象。分離レベル RS でバインドされます。これは、このパッケージの 2 番目の反復です。 <p>以前のバージョンの DB2 サーバーでは、すべてのバインド・ファイルが必要なわけではなく、バインド時にエラーが返されます。バインド・オプション SQLERROR(CONTINUE) を使用することにより、すべてのプラットフォーム上で同一のパッケージをバインドでき、そこでサポートされていないステートメントに関するエラーが無視されるようにしてください。</p>				

db2cli.lst ファイルには、DB2 CLI が DB2 バージョン 8 サーバーに接続するのに必要なバインド・ファイルの名前 (db2clipk.bnd, db2clist.bnd, and db2schema.bnd) が含まれています。

ホストおよび iSeries サーバーの場合は、ddcsvm.lst、ddcsmvs.lst、ddcsvse.lst、または ddc400.lst の各バインド・リスト・ファイルのうちいずれか 1 つを使用してください。

関連タスク:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI 環境のセットアップ』

CLI アプリケーションでの CLI LOAD ユーティリティによるデータのインポート

CLI LOAD 機能は、CLI から IBM DB2 LOAD ユーティリティへのインターフェースを設けます。この機能を使用すると、配列を挿入する代わりに、LOAD を使用して CLI 中のデータを挿入できます。大量のデータを挿入する必要がある場合に、このオプションを使用するとパフォーマンスの面で大きな利点が生じます。このインターフェースは LOAD を呼び出すので、LOAD を使用する際の考慮事項が、CLI LOAD インターフェースを使用する際にも考慮される必要があります。

前提条件:

CLI LOAD ユーティリティを使用してデータをインポートする場合は、その前に CLI アプリケーションを初期設定してあることを確認してください。

制約事項:

- IBM DB2 LOAD ユーティリティとは違って、CLI LOAD ユーティリティは入力ファイルから直接データをロードしない。その代わりに、必要に応じて、アプリケーションはデータを入力ファイルから取り出して、準備したステートメント中のパラメーター・マーカーに対応する当該アプリケーション・パラメーター中に挿入する必要があります。
- ロード・ユーティリティはアトミシティを排除するので、データの追加は非アトミックである。LOAD は、渡された行をすべて正常に挿入できる訳ではありません。たとえば、行を挿入するとユニーク・キーの制約に対する違反が生じる場合は、LOAD はこの行を挿入しませんが、残りの行のロードを続行します。
- COMMIT が LOAD によって発行される。したがって、データの挿入が正常に完了したら、LOAD やトランザクション中の他のステートメントをロールバックできません。
- CLI LOAD インターフェースに関して報告されるエラーは、配列を挿入する際のエラーとは違う。特定の行に関するエラーなどの重大でないエラーや警告は、LOAD メッセージ・ファイルだけに示されます。

手順:

CLI LOAD ユーティリティを使用してデータをインポートするには、以下のようになります。

1. 以下のサポートされている値のいずれかを指定して、`SQLSetStmtAttr()` 中にステートメント属性 `SQL_ATTR_USE_LOAD_API` を指定します。

SQL_USE_LOAD_INSERT

LOAD ユーティリティを使用して、表中の既存のデータに追加します。

SQL_USE_LOAD_REPLACE

LOAD ユーティリティを使用して、表中の既存のデータを置き換えます。

たとえば、以下の呼び出しは、CLI LOAD ユーティリティを使用して表中の既存のデータに追加することを指示します。

```
SQLSetStmtAttr (hStmt, SQL_ATTR_USE_LOAD_API,  
                (SQLPOINTER) SQL_USE_LOAD_INSERT, 0);
```

注: `SQL_USE_LOAD_INSERT` または `SQL_USE_LOAD_REPLACE` を設定し、`SQL_USE_LOAD_OFF` を設定しないと、以下を除く CLI 関数は呼び出せません (下記のステップ 3 を参照)。

- `SQLBindParameter()`
 - `SQLExecute()`
 - `SQLExtendedBind()`
 - `SQLParamOptions()`
 - `SQLSetStmtAttr()`
2. タイプ `db2LoadStruct` の構造体を作成し、この構造体を使用してご希望のロード・オプションを指定します。 `SQL_ATTR_LOAD_INFO` ステートメントをこの構造体を指すポインターに設定します。
 3. 挿入するデータのために準備した SQL ステートメントに対して、`SQLExecute()` を発行します。
 4. `SQL_USE_LOAD_OFF` を指定して `SQLSetStmtAttr()` を呼び出します。呼び出すと、LOAD ユーティリティを使用したデータの処理が終了します。以後 `SQL_ATTR_USE_LOAD_API` を再設定しない限り、正規の CLI 配列の挿入が有効になります (ステップ 1 を参照)。

関連タスク:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『CLI アプリケーションの初期設定』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『列方向配列の入力を使用した CLI アプリケーションでのパラメーター・マーカのバインド』

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『行方向配列の入力を使用した CLI アプリケーションでのパラメーター・マーカのバインド』

関連資料:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『ステートメント属性 (CLI) のリスト』
- 1149 ページの『LOAD』
- 945 ページの『db2Load - ロード』

関連サンプル:

- 『tload.c -- How to insert data using the CLI LOAD utility (CLI)』

PATCH1 CLI/ODBC 構成キーワード

キーワードの説明:

認識されている CLI/ODBC アプリケーション問題について、予備手段を使用します。

db2cli.ini キーワード構文:

```
PATCH1 = { 0 | 1 | 2 | 4 | 8 | 16 | ... }
```

デフォルト設定:

予備手段を使用しません。

使用上の注意:

このキーワードは、ODBC アプリケーションの認識されている問題について予備手段を指定するために使用されます。値は、予備手段なしを指定するか、1 つまたは複数の予備手段について指定できます。ここに指定されたパッチ値は、一緒に設定できる PATCH2 値とともに使用されます。

「DB2 CLI/ODBC 設定」ノートブックを使用して、使用する 1 つ以上のパッチを選択できます。db2cli.ini ファイル自体に値を設定して複数のパッチ値を使用する場合は、単純に値を足してキーワード値を作成します。たとえば、パッチ 1、4、および 8 を使用する場合は、PATCH1=13 と指定してください。

- 0 = 予備手段なし (デフォルト)

PATCH1 値の現行のリストについては、以下の DB2 アプリケーション開発 Web サイトを参照してください。

<http://www.ibm.com/software/data/db2/udb/ad>

関連概念:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『db2cli.ini 初期設定ファイル』

関連資料:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI/ODBC 構成キーワード (カテゴリー別)』
- 873 ページの『PATCH2 CLI/ODBC 構成キーワード』

PATCH2 CLI/ODBC 構成キーワード

キーワードの説明:

認識されている CLI/ODBC アプリケーション問題について、予備手段を使用します。

db2cli.ini キーワード構文:

PATCH2 = "パッチ値 1, パッチ値 2, パッチ値 3, ..."

デフォルト設定:

予備手段を使用しません

使用上の注意:

このキーワードは、CLI/ODBC アプリケーションの認識されている問題について予備手段を指定するために使用されます。値は、予備手段なしを指定するか、1 つまたは複数の予備手段について指定できます。ここに指定されたバッチ値は、一緒に設定できる PATCH1 値とともに使用される場合があります。

複数のパッチを指定するときには、コンマで区切られたストリングとして値を指定します (値が足されて合計が使用される PATCH1 オプションとは異なります)。

- 0 = 予備手段なし (デフォルト)

PATCH2 値 3、4、および 8 を設定するには、次を指定してください。

```
PATCH2="3, 4, 8"
```

PATCH2 値の現行のリストについては、以下の DB2 アプリケーション開発 Web サイトを参照してください。

```
http://www.ibm.com/software/data/db2/udb/ad
```

関連概念:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『db2cli.ini 初期設定ファイル』

関連資料:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI/ODBC 構成キーワード (カテゴリー別)』
- 872 ページの『PATCH1 CLI/ODBC 構成キーワード』

CLIPKG CLI/ODBC 構成キーワード

キーワードの説明:

db2cli.ini キーワード構文:

CLIPKG = 3 | 4 | ... | 30

デフォルト設定:

3 つの大きなパッケージが生成されます。

使用上の注意:

このキーワードは、CLI/ODBC アプリケーションでの SQL セクションの数を増やすのに使用されます。これが使用される場合には、管理者は CLIPKG バインド・オプションを使用して、必要なバインド・ファイルを明示的にバインドする必要があります。クライアント・アプリケーションの場合、クライアント上の db2cli.ini ファイルは、この CLIPKG の値で更新する必要があります。CLI/JDBC ストアード・プロシージャの場合、(UNIX または Intel プラットフォームでの DB2 UDB バージョン 6.1 以降の) サーバー上の db2cli.ini ファイルは、CLIPKG の同じ値で更新する必要があります。

値が 3 ~ 30 の整数に指定されていない場合には、デフォルトが使用されます。このとき、エラーまたは警告は出されません。

この設定は、ラージ・パッケージ (384 個のセクションを含む) にのみ適用されます。スモール・パッケージ (64 個のセクションを含む) の数は 3 個であり、変更できません。

パッケージはデータベースでスペースをとるため、増やすセクションの数は、ご使用のアプリケーションを実行できるだけの数にとどめるようお勧めします。

関連概念:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLIでのハンドル』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『db2cli.ini 初期設定ファイル』

関連資料:

- コマンド・リファレンスの『BIND コマンド』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI/ODBC 構成キーワード (カテゴリー別)』

SQLColAttribute 関数 (CLI) - 列属性を戻す

目的:

仕様:	DB2 CLI 5.0	ODBC 3.0	ISO CLI
-----	-------------	----------	---------

SQLColAttribute() は、結果セット内の列について記述子情報を返します。記述子情報は、文字ストリング、32 ビットの記述子従属値、または整数値として返されます。

同等のユニコード: この関数はユニコード文字セットとともに使用することもできます。これに対応するユニコード関数は SQLColAttributeW() です。ANSI からユニコード関数へのマッピングの詳細は、Unicode 関数 (CLI) を参照してください。

構文:

```
SQLRETURN SQLColAttribute (
    SQLHSTMT          StatementHandle,      /* hstmt */
    SQLSMALLINT       ColumnNumber,         /* icol */
    SQLSMALLINT       FieldIdentifier,      /* fDescType */
    SQLPOINTER        CharacterAttributePtr, /* rgbDesc */
    SQLSMALLINT       BufferLength,         /* cbDescMax */
    SQLSMALLINT       *StringLengthPtr,    /* pcbDesc */
    SQLPOINTER        NumericAttributePtr); /* pfDesc */
```

関数引き数:

表 205. SQLColAttribute 引き数

データ・タイプ	引き数	使用法	説明
SQLHSTMT	<i>StatementHandle</i>	入力	ステートメント・ハンドル
SQLUSMALLINT	<i>ColumnNumber</i>	入力	フィールド値を検索する IRD 中のレコードの番号。この引き数は、結果データの列番号に対応し、その番号は 1 で始まり、左から右へ連続で順序付けられています。列は任意の順序で記述できます。 この引き数内に列 0 を指定できますが、SQL_DESC_TYPE と SQL_DESC_OCTET_LENGTH を除くすべての値が、未定義の値を返すこととなります。
SQLSMALLINT	<i>FieldIdentifier</i>	入力	返されることになっている IRD の行 <i>ColumnNumber</i> にあるフィールド (877 ページの表 206 を参照)。
SQLPOINTER	<i>CharacterAttributePtr</i>	出力	フィールドが文字ストリングの場合、IRD の <i>ColumnNumber</i> 行の <i>FieldIdentifier</i> フィールド内の値を返すバッファを指すポインター。それ以外の場合は、フィールドは未使用になります。

SQLColAttribute

表 205. *SQLColAttribute* 引き数 (続き)

データ・タイプ	引き数	使用法	説明
SQLINTEGER	<i>BufferLength</i>	入力	フィールドが文字ストリングの場合、 <i>CharacterAttributePtr</i> バッファの長さ。それ以外の場合は、このフィールドは無視されます。
SQLSMALLINT *	<i>StringLengthPtr</i>	出力	<p>*<i>CharacterAttributePtr</i> に戻すのに使える総バイト数 (文字データの場合の NULL 終了文字のバイト・カウントを除く) を戻すバッファを指すポインター。</p> <p>文字データの場合、戻りに使用できるバイト数が <i>BufferLength</i> 以上であれば、*<i>CharacterAttributePtr</i> 内の記述子情報は <i>BufferLength</i> バイトから NULL 終了文字分の長さを差し引いたバイトに切り捨てられ、DB2 CLI. によって NULL 終了されます。</p> <p>その他のすべてのデータのタイプについては、<i>BufferLength</i> の値は無視されて、DB2 CLI は、*<i>CharacterAttributePtr</i> のサイズを 32 ビットと想定します。</p>
SQLPOINTER	<i>NumericAttributePtr</i>	出力	フィールドが <code>SQL_DESC_COLUMN_LENGTH</code> のような文字ストリングの場合、IRD の <i>ColumnNumber</i> 行の <i>FieldIdentifier</i> フィールド内の値を返す整数バッファを指すポインター。それ以外の場合は、フィールドは未使用になります。

使用法:

`SQLColAttribute()` は、情報を **NumericAttributePtr* または **CharacterAttributePtr* に返します。整数情報は、32 ビットの符号付き値として、**NumericAttributePtr* に返されます。その他のすべてのフォーマットの情報は、**CharacterAttributePtr* に返されます。情報が **NumericAttributePtr* に返されるとき、DB2 CLI は、*CharacterAttributePtr*、*BufferLength*、および *StringLengthPtr* を無視します。情報が **CharacterAttributePtr* に返されるとき、DB2 CLI は、*NumericAttributePtr* を無視します。

`SQLColAttribute()` は、IRD の記述子フィールドからの値を返します。関数は、記述子ハンドルではなくステートメント・ハンドルを使用して呼び出されます。下記にリスト

される *FieldIdentifier* 値について SQLColAttribute() によって返される値は、適当な IRD ハンドルを使用して SQLGetDescField() を呼び出すことにより取り出すことができます。

現在定義されている記述子タイプ、そのタイプが (おそらく別の名前で) 導入されている DB2 CLI のバージョン、およびそれについて情報が返される引き数を、以下に示します。さまざまなデータ・ソースを利用するために、より多くの記述子タイプが今後定義される見込みです。

DB2 CLI は、記述子タイプのおおのについて値を返さなければなりません。記述子タイプがデータ・ソースに適用されない場合、他に断り書きがない限り、DB2 CLI は、*StringLengthPtr に 0 を返すか、または *CharacterAttributePtr に空ストリングを返します。

以下の表には、SQLColAttribute() によって返される記述子タイプがリストされています。

表 206. SQLColAttribute 引き数

<i>FieldIdentifier</i>	情報の戻り先	説明
SQL_DESC_AUTO_UNIQUE_VALUE (DB2 CLI v2)	Numeric AttributePtr	列データ・タイプが自動増分データ・タイプであるかどうかを示します。 DB2 SQL データ・タイプの場合はすべて、SQL_FALSE が <i>NumericAttributePtr</i> に返されます。現在、列が ID 列かどうかを DB2 CLI は確かめられないので、常に SQL_FALSE が戻されます。このような制限事項は、ODBC 仕様に全面的に準じているわけではありません。UNIX、および Windows サーバー用の将来のバージョンの DB2 CLI では、auto-unique のサポートが設けられる予定です。
SQL_DESC_BASE_COLUMN_NAME (DB2 CLI v5)	Character AttributePtr	セット列用の基本列名。基本列名が存在しない場合 (列が式になっている場合など) は、この変数には空ストリングが入ります。 この情報は、読取専用フィールドである IRD の SQL_DESC_BASE_COLUMN_NAME レコード・フィールドから返されます。
SQL_DESC_BASE_TABLE_NAME (DB2 CLI v5)	Character AttributePtr	列を含む基本表の名前。基本表名が定義できないか適用不能である場合、この変数には空ストリングが入ります。

SQLColAttribute

表 206. *SQLColAttribute* 引き数 (続き)

<i>FieldIdentifier</i>	情報の戻り先	説明
SQL_DESC_CASE_SENSITIVE (DB2 CLI v2)	Numeric AttributePtr	列データ・タイプが大文字小文字の区別があるタイプであるかどうかを示します。 SQL_TRUE または SQL_FALSE のどちらが <i>NumericAttributePtr</i> に返されるかは、データ・タイプに依存します。 大文字小文字の区別はグラフィック・データ・タイプには適用されず、SQL_FALSE が返されます。 非文字データ・タイプには SQL_FALSE が返されます。
SQL_DESC_CATALOG_NAME (DB2 CLI v2)	Character AttributePtr	DB2 CLI は 1 つの表につき 2 つの部分からなる命名しかサポートしないため、空ストリングが返されます。
SQL_DESC_CONCISE_TYPE (DB2 CLI v5)	Numeric AttributePtr	コンサイス・データ・タイプ 日時データ・タイプの場合、このフィールドはコンサイス・データ・タイプ、たとえば、SQL_TYPE_TIME を返します。 この情報は、IRD の SQL_DESC_CONCISE_TYPE レコード・フィールドから返されます。
SQL_DESC_COUNT (DB2 CLI v2)	Numeric AttributePtr	結果セット内の列数が、 <i>NumericAttributePtr</i> に返されます。
SQL_DESC_DISPLAY_SIZE (DB2 CLI v2)	Numeric AttributePtr	文字フォーマットでデータを表示するのに必要な最大バイト数が、 <i>NumericAttributePtr</i> に返されます。 おのこの列タイプの表示サイズについては、『データ・タイプ表示サイズ』の表を参照してください。

表 206. SQLColAttribute 引き数 (続き)

FieldIdentifier	情報の戻り先	説明
SQL_DESC_DISTINCT_TYPE (DB2 CLI v2)	Character AttributePtr	列のユーザー定義の特殊タイプ名が、 <i>CharacterAttributePtr</i> に返されます。列が組み込み SQL タイプであってユーザー定義の特殊タイプではない場合、空ストリングが返されます。 注: これは、ODBC によって定義された記述子属性のリストに対する IBM 定義の拡張機能です。
SQL_DESC_FIXED_PREC_SCALE (DB2 CLI v2)	Numeric AttributePtr	SQL_TRUE は、列がデータ・ソース・ユニークの固定精度および非ゼロのスケールを持っている場合です。 SQL_FALSE は、列がデータ・ソース・ユニークの固定精度および非ゼロのスケールを持っていない場合です。 DB2 SQL データ・タイプの場合にはすべて、SQL_FALSE が <i>NumericAttributePtr</i> に返されます。
SQL_DESC_LABEL (DB2 CLI v2)	Character AttributePtr	列ラベルが、 <i>CharacterAttributePtr</i> に返されます。列にラベルがない場合、列名または列式が返されます。列にラベルがなく、名前もない場合は、空ストリングが返されます。
SQL_DESC_LENGTH (DB2 CLI v2)	Numeric AttributePtr	文字ストリングまたはバイナリー・データ・タイプの最大文字長または実際の文字長のどちらかである数値。これは、固定長データ・タイプの場合は最大文字長、可変長データ・タイプの場合は実際の文字長となります。その値からは常に、文字ストリングの終わりを示す NULL 終了バイトが除かれています。 この情報は、IRD の SQL_DESC_LENGTH レコード・フィールドから返されます。
SQL_DESC_LITERAL_PREFIX (DB2 CLI v5)	Character AttributePtr	この VARCHAR(128) レコード・フィールドには、DB2 CLI がこのデータ・タイプのリテラル用の接頭部として認識する文字 (複数を含む) が入っています。リテラルの接頭部が適用不能であるデータ・タイプに対しては、このフィールドに空ストリングが入れられます。

SQLColAttribute

表 206. *SQLColAttribute* 引き数 (続き)

<i>FieldIdentifier</i>	情報の戻り先	説明
SQL_DESC_LITERAL_SUFFIX (DB2 CLI v5)	Character AttributePtr	この VARCHAR(128) レコード・フィールドには、DB2 CLI がこのデータ・タイプのリテラル用の接尾部として認識する文字 (複数を含む) が入っています。リテラルの接尾部が適用不能であるデータ・タイプに対しては、このフィールドに空ストリングが入られます。
SQL_DESC_LOCAL_TYPE_NAME (DB2 CLI v5)	Character AttributePtr	この列には、データ・タイプの正規名とは異なる、データ・タイプ用のローカライズされた (ネイティブ言語の) 名前が入ります。ローカライズされた名前がない場合は、空ストリングが返されます。このフィールドは、表示の目的においてのみ使用されます。ストリングの文字セットはロケールに依存しており、通常はサーバーのデフォルト文字セットです。
SQL_DESC_NAME (DB2 CLI v2)	Character AttributePtr	列 <i>ColumnNumber</i> の名前が、 <i>CharacterAttributePtr</i> に返されます。列が式である場合は、列番号が返されます。 いずれの場合にも、SQL_DESC_UNNAMED が SQL_NAMED に設定されます。列名または列別名がない場合は、空ストリングが返されて、SQL_DESC_UNNAMED が SQL_UNNAMED に設定されます。 この情報は、IRD の SQL_DESC_NAME レコード・フィールドから返されます。
SQL_DESC_NULLABLE (DB2 CLI v2)	Numeric AttributePtr	<i>ColumnNumber</i> によって識別される列に NULL を入れることができる場合、SQL_NULLABLE が <i>NumericAttributePtr</i> に返されます。 列が NULL を受け入れないように制約されている場合、SQL_NO_NULLS が <i>NumericAttributePtr</i> に返されます。 この情報は、IRD の SQL_DESC_NULLABLE レコード・フィールドから返されます。

表 206. SQLColAttribute 引き数 (続き)

FieldIdentifier	情報の戻り先	説明
SQL_DESC_NUM_PREX_RADIX (DB2 CLI v5)	Numeric AttributePtr	<ul style="list-style-type: none"> • SQL_DESC_TYPE フィールド内のデータ・タイプが近似的なデータ・タイプである場合、この SQLINTEGER フィールドには 2 の値が入ります。SQL_DESC_PRECISION フィールドにビット数が入っているからです。 • SQL_DESC_TYPE フィールド内のデータ・タイプが正確な数値データ・タイプである場合は、このフィールドの値は 10 になります。SQL_DESC_PRECISION フィールドは 10 進数を含むからです。 • 数値以外のすべてのデータ・タイプに対しては、このフィールドは 0 に設定されます。
SQL_DESC_OCTET_LENGTH (DB2 CLI v2)	Numeric AttributePtr	<p>列に関連したデータのバイト数が、<i>NumericAttributePtr</i> に返されます。これは、SQL_C_DEFAULT が C データ・タイプに指定されているときにこの列の取り出し時または SQLGetData() 実行時に転送されたデータの長さをバイト数で表したものです。個々の SQL データ・タイプの長さについては、『データ・タイプ長』の表を参照してください。</p> <p><i>ColumnNumber</i> 内で識別される列が、固定長の文字ストリングまたはバイナリー・ストリング (たとえば、SQL_CHAR または SQL_BINARY) である場合、実際の長さが返されます。</p> <p><i>ColumnNumber</i> 内で識別される列が、可変長の文字ストリングまたはバイナリー・ストリング (たとえば、SQL_VARCHAR または SQL_BLOB) である場合、最大長が返されません。</p>

SQLColAttribute

表 206. *SQLColAttribute* 引き数 (続き)

<i>FieldIdentifier</i>	情報の戻り先	説明
SQL_DESC_PRECISION (DB2 CLI v2)	Numeric AttributePtr	<p>列が SQL_DECIMAL、SQL_NUMERIC、SQL_DOUBLE、SQL_FLOAT、SQL_INTEGER、SQL_REAL、または SQL_SMALLINT の場合、数値の精度 (有効桁数) が、<i>NumericAttributePtr</i> に返されます。</p> <p>列が文字 SQL データ・タイプである場合、<i>NumericAttributePtr</i> に返される精度は、列が保持できる文字 の最大数を示します。</p> <p>列がグラフィック SQL データ・タイプの場合、<i>NumericAttributePtr</i> に返される精度は、列が保持できる 2 バイト文字 の最大数を指定します。</p> <p>個々の SQL データ・タイプの精度については、『データ・タイプ精度』の表を参照してください。</p> <p>この情報は、IRD の SQL_DESC_PRECISION レコード・フィールドから返されます。</p>
SQL_DESC_SCALE (DB2 CLI v2)	Numeric AttributePtr	<p>列のスケールの属性が返されます。個々の SQL データ・タイプのスケールについては、『データ・タイプ・スケール』を参照してください。</p> <p>この情報は、IRD の SCALE レコード・フィールドから返されます。</p>
SQL_DESC_SCHEMA_NAME (DB2 CLI v2)	Character AttributePtr	<p>列を含む表のスキーマが、<i>CharacterAttributePtr</i> に返されます。DB2 CLI がこの属性を判別できないと、空ストリングが返されます。</p>

表 206. SQLColAttribute 引き数 (続き)

FieldIdentifier	情報の戻り先	説明
SQL_DESC_SEARCHABLE (DB2 CLI v2)	Numeric AttributePtr	<p>列データ・タイプが検索可能かどうかを示します。</p> <ul style="list-style-type: none"> • SQL_PRED_NONE (DB2 CLI v2 での SQL_UNSEARCHABLE): 列を WHERE 文節に使用できない場合。 • SQL_PRED_CHAR (DB2 CLI v2 での SQL_LIKE_ONLY): LIKE 述部を用いてのみ、列を WHERE 文節に使用できる場合。 • SQL_PRED_BASIC (DB2 CLI v2 での SQL_ALL_EXCEPT_LIKE): LIKE を除くすべての比較演算子を用いて、列を WHERE 文節に使用できる場合。 • SQL_SEARCHABLE: どの述部を指定したときでも WHERE 文節で列を使用できる場合。
SQL_DESC_TABLE_NAME (DB2 CLI v2)	Character AttributePtr	DB2 CLI がこの属性を判別できないと、空ストリングが返されます。
SQL_DESC_TYPE (DB2 CLI v2)	Numeric AttributePtr	<p><i>ColumnNumber</i> で識別される列の SQL データ・タイプが、<i>NumericAttributePtr</i> に返されます。返される可能性のある値は、『CLI 用のシンボリックおよびデフォルトのデータ・タイプ』にリストされています。</p> <p><i>ColumnNumber</i> が 0 に等しければ、可変長ブックマークの場合は SQL_BINARY が返され、固定長ブックマークの場合は SQL_INTEGER が返されます。</p> <p>日時データ・タイプの場合、このフィールドは冗長データ・タイプ、つまり、SQL_DATETIME を返します。</p> <p>この情報は、IRD の SQL_DESC_TYPE レコード・フィールドから返されます。</p>

SQLColAttribute

表 206. *SQLColAttribute* 引き数 (続き)

<i>FieldIdentifier</i>	情報の戻り先	説明
SQL_DESC_TYPE_NAME (DB2 CLI v2)	Character AttributePtr	列のタイプ (SQL ステートメントに入力したとおりのもの) が、 <i>CharacterAttributePtr</i> に返されます。 各データ・タイプの詳細は、『CLI 用のシンボリックおよびデフォルトのデータ・タイプ』を参照してください。
SQL_DESC_UNNAMED (DB2 CLI v5)	Numeric AttributePtr	SQL_NAMED または SQL_UNNAMED。IRD の SQL_DESC_NAME フィールドに列別名または列名が入っている場合、SQL_NAMED が返されます。列名も列別名も入っていない場合は、SQL_UNNAMED が返されます。 この情報は、IRD の SQL_DESC_UNNAMED レコード・フィールドから返されます。
SQL_DESC_UNSIGNED (DB2 CLI v2)	Numeric AttributePtr	列データ・タイプが無符号タイプであるかどうかを示します。 すべての非数値データ・タイプについては、SQL_TRUE が <i>NumericAttributePtr</i> に返され、すべての数値データ・タイプについては、SQL_FALSE が返されます。

表 206. SQLColAttribute 引き数 (続き)

FieldIdentifier	情報の戻り先	説明
SQL_DESC_UPDATABLE (DB2 CLI v2)	Numeric AttributePtr	<p>列のデータ・タイプが更新可能なデータ・タイプであるかどうかを指定します。</p> <ul style="list-style-type: none"> DB2 SQL データ・タイプの場合はずべて、SQL_ATTR_READWRITE_UNKNOWN が <i>NumericAttributePtr</i> に返されます。これが戻されるのは、列が更新可能かどうかを現在 DB2 CLI で確かめることはできないからです。UNIX および Windows サーバー用の将来のバージョンの DB2 CLI では、列が更新可能かどうかを判別できるようになる予定です。 列をカタログ関数呼び出しから入手した場合には、SQL_ATTR_READONLY が返されません。 <p>ODBC は以下の値 (DB2 CLI からは返されません) も定義しています。</p> <ul style="list-style-type: none"> SQL_ATTR_WRITE

この関数は、SQLDescribeCol() の代替として拡張性のあるものです。

SQLDescribeCol() は、ANSI-89 SQL に基づく記述子情報の固定セットを返します。

SQLColAttribute() は、ANSI SQL-92 および DBMS ベンダーの拡張機能で使用可能な、記述子情報のより広範なセットにアクセスできるようになっています。

戻りコード:

- SQL_SUCCESS
- SQL_SUCCESS_WITH_INFO
- SQL_STILL_EXECUTING
- SQL_ERROR
- SQL_INVALID_HANDLE

診断:

表 207. SQLColAttribute SQLSTATE

SQLSTATE	説明	解説
01000	警告 !	通知メッセージ。(関数は、SQL_SUCCESS_WITH_INFO を返します。)

SQLColAttribute

表 207. *SQLColAttribute* *SQLSTATE* (続き)

SQLSTATE	説明	解説
01004	データが切り捨てられました。	バッファ <i>*CharacterAttributePtr</i> は、文字列値を全部返すのに十分な大きさではなかったため、文字列値が切り捨てられました。 <i>*StringLengthPtr</i> には、切り捨て前の文字列値の長さが戻されます。(関数は、 <i>SQL_SUCCESS_WITH_INFO</i> を返します。)
07005	ステートメントが結果セットを返しませんでした。	<i>StatementHandle</i> に関連したステートメントが結果セットを返しませんでした。記述する列がありませんでした。
07009	記述子索引が無効です。	<i>ColumnNumber</i> に指定された値が 0 と同等であり、 <i>SQL_ATTR_USE_BOOKMARKS</i> ステートメント属性が <i>SQL_UB_OFF</i> でした。引き数 <i>ColumnNumber</i> に指定された値は、結果セット内の列数より大きい値でした。
HY000	General error.	特定の <i>SQLSTATE</i> がなかった場合のエラーが発生しました。 <i>SQLGetDiagRec()</i> から <i>*MessageText</i> バッファ内に戻されたエラー・メッセージに、エラーとその原因が説明されています。
HY001	メモリの割り振りが失敗しました。	DB2 CLI は、この関数の実行または完了をサポートするのに必要なメモリを割り当てられません。プロセス・レベルのメモリがアプリケーション・プロセスに使い尽くされた可能性があります。プロセス・レベルのメモリ制限については、オペレーティング・システムの構成を調べてください。
HY008	操作が取り消されました。	<i>StatementHandle</i> で非同期処理が使用可能になりました。関数が呼び出され、その実行が完了する前に、 <i>SQLCancel()</i> がマルチスレッドのアプリケーション内の別のスレッドから、 <i>StatementHandle</i> で呼び出されました。その関数が再び <i>StatementHandle</i> で呼び出されました。
HY010	関数のシーケンス・エラーです。	<i>SQLPrepare()</i> または <i>SQLExecDirect()</i> を <i>StatementHandle</i> 用に呼び出す前に、この関数が呼び出されました。 非同期で実行中の関数 (この関数ではない) が、 <i>StatementHandle</i> で呼び出されましたが、この関数の呼び出し時にはまだ実行中でした。 <i>StatementHandle</i> で <i>SQLExecute()</i> または <i>SQLExecDirect()</i> が呼び出され、 <i>SQL_NEED_DATA</i> が戻されました。すべての実行時データ・パラメータまたは列用のデータの送信前に、この関数が呼び出されました。
HY090	文字列またはバッファの長さが無効です。	引き数 <i>BufferLength</i> に指定された値は、0 より小さい値でした。

表 207. SQLColAttribute SQLSTATE (続き)

SQLSTATE	説明	解説
HY091	記述子フィールド ID が無効です。	引き数 <i>FieldIdentifier</i> に指定された値は、定義されているものの 1 つではなく、インプリメンテーション定義の値でもありませんでした。
HYC00	ドライバーが使用できません。	引き数 <i>FieldIdentifier</i> に指定された値は、DB2 CLI でサポートされていませんでした。

StatementHandle に関連した SQL ステートメントをデータ・ソースが評価する時期に応じて、SQLPrepare() の後から SQLExecute() の前までの間に呼び出された SQLColAttribute() は、SQLPrepare() または SQLExecute() によって返される可能性のある任意の SQLSTATE を返すことができます。

パフォーマンス上の理由から、アプリケーションは、ステートメントの実行前に SQLColAttribute() を呼び出さないようにする必要があります。

制約事項:

なし。

例:

```
/* get display size for column */
cliRC = SQLColAttribute(hstmt,
                        (SQLSMALLINT)(i + 1),
                        SQL_DESC_DISPLAY_SIZE,
                        NULL,
                        0,
                        NULL,
                        &colDataDisplaySize)
```

関連概念:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『Unicode 関数 (CLI)』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『DB2 CLI 用の SQLSTATES』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI アプリケーションの記述子』

関連資料:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI アプリケーション用の SQL 記号データ・タイプおよびデフォルト・データ・タイプ』

SQLColAttribute

- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻の『SQLBindCol 関数 (CLI) - アプリケーション変数または LOB ロケーターへの列のバインド』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻の『SQLCancel 関数 (CLI) - ステートメントの取り消し』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻の『SQLDescribeCol 関数 (CLI) - 列の属性のセットを戻す』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻の『SQLFetch 関数 (CLI) - 次の行の取り出し』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻の『SQLFetchScroll 関数 (CLI) - すべてのバインド列の行セットの取り出しとデータの戻り』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻の『データ・タイプ精度 (CLI) 表』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻の『データ・タイプ・スケール (CLI) 表』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻の『データ・タイプ長 (CLI) 表』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻の『データ・タイプ表示 (CLI) 表』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『CLI 関数戻りコード』

関連サンプル:

- 『tbread.c -- How to read data from tables (CLI)』
- 『utilcli.c -- Utility functions used by DB2 CLI samples (CLI)』

SQLDriverConnect 関数 (CLI) - データ・ソースへの (拡張) 接続

目的:

仕様:	DB2 CLI 2.1	ODBC 1.0	
-----	-------------	----------	--

SQLDriverConnect() は、SQLConnect() の代替りの関数です。両方の関数ともターゲット・データベースに対する接続を確立しますが、SQLDriverConnect() は追加の接続パラメーターと、接続情報をユーザーに入力要求する機能をサポートします。

SQLConnect() でサポートされる 3 つの入力引き数 (データ・ソース名、ユーザー ID、およびパスワード) 以外のパラメーターがデータ・ソースに必要な場合、または DB2 CLI のグラフィカル・ユーザー・インターフェースを使用してユーザーに必須接続情報を入力要求したい場合に、SQLDriverConnect() を使用します。

接続が確立されると、完全な接続ストリングが返されます。アプリケーションは、以後の接続要求のためにこのストリングを保管することができます。

同等のユニコード: この関数はユニコード文字セットとともに使用することもできます。これに対応するユニコード関数は `SQLDriverConnectW()` です。ANSI からユニコード関数へのマッピングの詳細は、Unicode 関数 (CLI) を参照してください。

構文:

総称

```
SQLRETURN SQLDriverConnect (
    SQLHDBC      ConnectionHandle,          /* hdbc */
    SQLHWND      WindowHandle,             /* hwnd */
    SQLCHAR      *InConnectionString,     /* szConnStrIn */
    SQLSMALLINT  InConnectionStringLength, /* cbConnStrIn */
    SQLCHAR      *OutConnectionString,     /* szConnStrOut */
    SQLSMALLINT  OutConnectionStringCapacity, /* cbConnStrOutMax */
    SQLSMALLINT  *OutConnectionStringLengthPtr, /* pcbConnStrOut */
    SQLUSMALLINT DriverCompletion);        /* fDriverCompletion */
```

関数引き数:

表 208. `SQLDriverConnect` 引き数

データ・タイプ	引き数	使用法	説明
SQLHDBC	<i>ConnectionHandle</i>	入力	接続ハンドル
SQLHWND	<i>WindowHandle</i>	入力	ウィンドウ・ハンドル。Windows プラットフォームでは、これは親ウィンドウ・ハンドルです。現在ウィンドウ・ハンドルは、Windows のみサポートされています。 NULL が渡されると、ダイアログは表示されません。
SQLCHAR *	<i>InConnectionString</i>	入力	完全、一部、または空 (NULL ポインター) の接続ストリング (以下の構文と説明を参照)。
SQLSMALLINT	<i>InConnectionStringLength</i>	入力	<i>InConnectionString</i> の長さ。
SQLCHAR *	<i>OutConnectionString</i>	出力	完全な接続ストリングのバッファーを指すポインター。 接続が正常に確立されると、このバッファーには完全な接続ストリングが入れます。アプリケーションは、このバッファー用に少なくとも <code>SQL_MAX_OPTION_STRING_LENGTH</code> バイトを割り振る必要があります。

SQLDriverConnect

表 208. *SQLDriverConnect* 引き数 (続き)

データ・タイプ	引き数	使用法	説明
SQLSMALLINT	<i>OutConnectionStringCapacity</i>	入力	<i>OutConnectionString</i> で指定されたバッファ어의最大サイズ。
SQLCHAR *	<i>OutConnectionStringLengthPtr</i>	出力	<i>OutConnectionString</i> バッファ어に返すために使用できる文字数を指すポインター。 * <i>OutConnectionStringLengthPtr</i> の値が <i>OutConnectionStringCapacity</i> 以上である場合、 <i>OutConnectionString</i> 内の完全接続ストリングは <i>OutConnectionStringCapacity</i> - 1 文字に切り捨てられます。
SQLUSMALLINT	<i>DriverCompletion</i>	入力	DB2 CLI がいつ詳細情報をユーザーに要求すべきかを示します。 有効値は以下のとおりです。 <ul style="list-style-type: none">• SQL_DRIVER_PROMPT• SQL_DRIVER_COMPLETE• SQL_DRIVER_COMPLETE_REQUIRED• SQL_DRIVER_NOPROMPT

使用法:

InConnectionString 引き数

要求の接続ストリングは、以下の構文になります。

```
connection-string ::= attribute[;] | attribute; connection-string
```

```
attribute ::= attribute-keyword=attribute-value  
| DRIVER=[{]attribute-value[}]
```

```
attribute-keyword ::= DSN | UID | PWD | NEWPWD  
| driver-defined-attribute-keyword
```

```
attribute-value ::= character-string  
driver-defined-attribute-keyword ::= identifier
```

ここで、

- 文字ストリングの文字数は 0 以上です。

- ID の文字数は 1 以上です。
- 属性キーワードは大文字小文字の区別がありません。
- 属性値は大文字小文字を区別することがあります。
- **DSN** キーワードの値はブランクのみでは成立しません。
- **NEWPWD** は、パスワード変更要求の一部として使用されます。アプリケーションは、**NEWPWD=anewpass;** などで、使用する新しいストリングを指定するか、または **NEWPWD=;** を指定して **DB2 CLI** ドライバーによって生成されるダイアログ・ボックスが新しいパスワードの入力を要求するようにすることができます。

接続ストリングと初期設定ファイルの文法上の理由から、`[]{}0,;*=!@` 文字の入っているキーワードおよび属性値は避ける必要があります。システム情報の文法上、キーワードとデータ・ソース名には、円記号 (¥) を入れることができません。 **DB2 CLI** パージョン 2 の場合、**DRIVER** キーワードの前後に中括弧が必要です。

あるキーワードがブラウズ要求の接続ストリングの中で繰り返される場合、 **DB2 CLI** は、そのキーワードの最初のオカレンスに関連した値を使用します。 **DSN** および **DRIVER** キーワードが同じブラウズ要求の接続ストリング内にある場合、 **DB2 CLI** は、最初に現れたキーワードの方を使用します。

OutConnectionString 引き数

結果の接続ストリングは、接続属性のリストになっています。接続属性は、属性キーワードとそれに対応する属性値から成っています。ブラウズ結果の接続ストリングは、以下の構文になります。

```
connection-string ::= attribute[;] | attribute; connection-string
```

```
attribute ::= [*]attribute-keyword=attribute-value
```

```
attribute-keyword ::= ODBC-attribute-keyword
```

```
| driver-defined-attribute-keyword
```

```
ODBC-attribute-keyword = {UID | PWD}[[:localized-identifier]
```

```
driver-defined-attribute-keyword ::= identifier[:localized-identifier]
```

```
attribute-value ::= {attribute-value-list} | ?
```

(中括弧はリテラルであり、**DB2 CLI** によって返されます。)

```
attribute-value-list ::= character-string [:localized-character
```

```
string] | character-string [:localized-character string], attribute-value-list
```

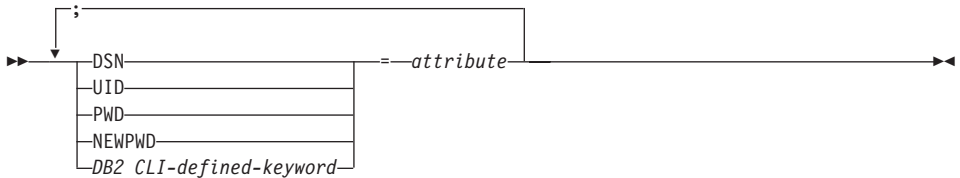
ここで、

- 文字ストリングおよびローカライズ文字ストリングの文字数は 0 以上です。
- **ID** およびローカライズ **ID** の文字数は、1 以上です。属性キーワードは、大文字小文字の区別がありません。
- 属性値は大文字小文字を区別することがあります。

SQLDriverConnect

接続ストリングと初期化ファイルの文法上の理由から、 `{ } ; ? * ! @` 文字の入っているキーワード、ローカライズ ID、および属性値は避ける必要があります。システム情報の文法上、キーワードとデータ・ソース名には、円記号 (¥) を入れることができません。

接続ストリングは、接続を完了するのに必要な 1 つ以上の値を渡すのに使用します。接続ストリングの内容と *DriverCompletion* の値で、DB2 CLI がユーザーとダイアログを確立する必要があるかどうかを判別します。



上記の各キーワードには、以下のような属性があります。

DSN データ・ソース名。データベースの名前または別名。 *DriverCompletion* が `SQL_DRIVER_NOPROMPT` と等しいときに必要です。

UID 許可名 (ユーザー ID)。

PWD 許可名に対応するパスワード。ユーザー ID のパスワードがないと、空の値が指定されます (`PWD=;`)。

NEWPWD

パスワード変更要求の一部として使用する新規パスワード。アプリケーションは、`NEWPWD=anewpass;` などで、使用する新しいストリングを指定するか、または `NEWPWD=;` を指定して DB2 CLI ドライバーによって生成されるダイアログ・ボックスが新しいパスワードの入力を要求するようにすることができます。 (*DriverCompletion* 引き数には `SQL_DRIVER_NOPROMPT` 以外の値を指定。)

CLI キーワードの任意の 1 つを接続ストリング上に指定することができます。キーワードが接続ストリング内で繰り返し指定されると、キーワードの最初のオカレンスに関連した値が使用されます。

CLI 初期設定ファイルにキーワードがある場合、それらのキーワードとそれらに対応する値は、接続ストリング内の DB2 CLI に渡される情報を追加するのに使用されます。CLI 初期設定ファイル内の情報が接続ストリング内の情報と矛盾するときは、接続ストリング内の値が優先されます。

表示されたダイアログ・ボックスをエンド・ユーザーが取り消すと、`SQL_NO_DATA_FOUND` が返されます。

次に示す *DriverCompletion* の値で、ダイアログがいつオープンするかが決まります。

SQL_DRIVER_PROMPT:

ダイアログは常に開始されます。接続ストリングと CLI 初期設定ファイルからの情報は初期値として使用され、ダイアログ・ボックスで入力したデータによって補足されます。

SQL_DRIVER_COMPLETE:

ダイアログは、接続ストリング内の情報が不足しているときだけ開始されます。接続ストリングからの情報は初期値として使用され、ダイアログ・ボックスで入力したデータによって補足されます。

SQL_DRIVER_COMPLETE_REQUIRED:

ダイアログは、接続ストリング内の情報が不足しているときだけ開始されます。接続ストリングからの情報は、初期値として使用されます。必須情報しか要求されません。ユーザーは、必要な情報だけを要求されます。

SQL_DRIVER_NOPROMPT:

ユーザーは、情報を要求されません。接続ストリングに含まれている情報を使用して、接続が試行されます。情報が足りない場合、SQL_ERROR が返されません。

接続が確立されると、完全な接続ストリングが返されます。特定のユーザー ID で 1 つのデータベースに複数の接続をセットアップする必要があるアプリケーションでは、この出力接続ストリングを保管する必要があります。次いで、このストリングを将来の SQLDriverConnect() 呼び出しの際の入力接続ストリング値として使用することができます。

戻りコード:

- SQL_SUCCESS
- SQL_SUCCESS_WITH_INFO
- SQL_NO_DATA_FOUND
- SQL_INVALID_HANDLE
- SQL_ERROR

診断:

SQLConnect() で生成されるすべての診断を、ここでも返すことができます。以下の表は、返すことのできる追加の診断を示したものです。

表 209. SQLDriverConnect SQLSTATE

SQLSTATE	説明	解説
01004	データが切り捨てられました。	バッファー <i>szConnstrOut</i> は、接続ストリング全体を保留できるほど大きくありませんでした。引き数 <i>*OutConnectionStringLengthPtr</i> には、戻りに使用できる接続ストリングの実際の長さが入っています。(関数は、SQL_SUCCESS_WITH_INFO を返します。)

SQLDriverConnect

表 209. *SQLDriverConnect SQLSTATE* (続き)

SQLSTATE	説明	解説
01S00	接続ストリング属性が無効です。	入力接続ストリングに無効なキーワードまたは属性値を指定し、以下のうちの 1 つが発生しましたが、データ・ソースへの接続は成功しました。 <ul style="list-style-type: none">認識されないキーワードが無視されました。無効な属性値が無視され、その代わりにデフォルト値が使用されました。 (関数は、 <code>SQL_SUCCESS_WITH_INFO</code> を返します。)
HY000	General error. ダイアログの失敗	接続ストリングに指定された情報は接続要求を行うには不十分でしたが、 <code>fCompletion</code> を <code>SQL_DRIVER_NOPROMPT</code> に設定してダイアログを禁止していました。 ダイアログを表示する試行が失敗しました。
HY090	ストリングまたはバッファアの長さが無効です。	<code>InConnectionStringLength</code> に指定された値は 0 より小さい値でしたが、 <code>SQL_NTS</code> と等しくありませんでした。 <code>OutConnectionStringCapacity</code> に指定された値は、0 より小さい値でした。
HY110	ドライバーの完了が無効です。	引き数 <code>fCompletion</code> に指定された値は、有効値のいずれとも等しくありませんでした。

制約事項:

なし。

例:

```
rc = SQLDriverConnect(hdbc,  
                      (SQLHWND)sqlHWND,  
                      InConnectionString,  
                      InConnectionStringLength,  
                      OutConnectionString,  
                      OutConnectionStringCapacity,  
                      StrLength2,  
                      DriveCompletion);
```

関連概念:

- ・ コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『Unicode 関数 (CLI)』
- ・ コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『DB2 CLI 用の SQLSTATES』

関連タスク:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『CLI アプリケーションの初期設定』

関連資料:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『SQLAllocHandle 関数 (CLI) - ハンドルの割り振り』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『SQLConnect 関数 (CLI) - データ・ソースへの接続』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『SQLSetConnectAttr 関数 (CLI) - 接続属性の設定』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『CLI 関数戻りコード』

関連サンプル:

- 『dbcongui.c -- How to connect to a database with a graphical user interface (GUI) (CLI)』
- 『dbconn.c -- How to connect to and disconnect from a database (CLI)』

SQLGetData 関数 (CLI) - 列からのデータの取得

目的:

仕様:	DB2 CLI 1.1	ODBC 1.0	ISO CLI
-----	-------------	----------	---------

SQLGetData() は、結果セットの現在行で 1 つの列のデータを取り出します。これは SQLBindCol() の代わりになるものであり、SQLFetch() または SQLFetchScroll() 呼び出しでアプリケーション変数または LOB ロケータヘデータを直接転送するのに使用されます。アプリケーションは、LOB を SQLBindCol() でバインド、または SQLGetData() を使用して LOB を検索できますが、両方の方法を同時に使用することはできません。SQLGetData() を使用して、大きなデータ値を分割して取り出すこともできます。

SQLFetch() または SQLFetchScroll() は、SQLGetData() の前に呼び出す必要があります。

列ごとに SQLGetData() を呼び出した後で、SQLFetch() または SQLFetchScroll() を呼び出して以下の行を取り出します。

構文:

```
SQLRETURN SQLGetData (
    SQLHSTMT          StatementHandle, /* hstmt */
    SQLUSMALLINT      ColumnNumber,    /* icol */
    SQLSMALLINT       TargetType,      /* fctype */
```

SQLGetData

```
SQLPOINTER TargetValuePtr, /* rgbValue */
SQLINTEGER BufferLength, /* cbValueMax */
SQLINTEGER *StrLen_or_IndPtr); /* pcbValue */
```

関数引き数:

表 210. *SQLGetData* 引き数

データ・タイプ	引き数	使用法	説明
SQLHSTMT	<i>StatementHandle</i>	入力	ステートメント・ハンドル
SQLUSMALLINT	<i>ColumnNumber</i>	入力	データ取り出しが要求されている列番号。結果セット列は、左から右へ順番に番号が付けられています。 <ul style="list-style-type: none">ブックマークを使用していない (SQL_ATTR_USE_BOOKMARKS ステートメント属性を SQL_UB_OFF に設定します) 場合は、列番号は 1 から始まります。ブックマークを使用している (ステートメント属性を SQL_UB_ON または SQL_UB_VARIABLE に設定します) 場合は、列番号は 0 から始まります。

表 210. SQLGetData 引き数 (続き)

データ・タイプ	引き数	使用法	説明
SQLSMALLINT	<i>TargetType</i>	入力	<p><i>ColumnNumber</i> による列 ID の C データ・タイプ。以下のタイプがサポートされます。</p> <ul style="list-style-type: none"> • SQL_C_BINARY • SQL_C_BIT • SQL_C_BLOB_LOCATOR • SQL_C_CHAR • SQL_C_CLOB_LOCATOR • SQL_C_DBCHAR • SQL_C_DBCLOB_LOCATOR • SQL_C_DECIMAL_IBM • SQL_C_DOUBLE • SQL_C_FLOAT • SQL_C_LONG • SQL_C_NUMERIC^a • SQL_C_SBIGINT • SQL_C_SHORT • SQL_C_TYPE_DATE • SQL_C_TYPE_TIME • SQL_C_TYPE_TIMESTAMP • SQL_C_TINYINT • SQL_C_UBIGINT • SQL_C_UTINYINT • SQL_C_WCHAR <p>SQL_ARD_TYPE を指定すると、データは、ARD の SQL_DESC_CONCISE_TYPE フィールドに指定されているデータ・タイプに変換されることとなります。</p> <p>SQL_C_DEFAULT を指定するとデータは、デフォルトの C データ・タイプに変換されます。</p>
SQLPOINTER	<i>TargetValuePtr</i>	出力	検索された列データを格納するバッファを指すポインター。
SQLINTEGER	<i>BufferLength</i>	入力	<i>TargetValuePtr</i> で指し示されたバッファの最大サイズ。ドライバが固定長データを戻すと、この値は無視されます。

SQLGetData

表 210. *SQLGetData* 引き数 (続き)

データ・タイプ	引き数	使用法	説明
SQLINTEGER *	<i>StrLen_or_IndPtr</i>	出力	<p>DB2 CLI が <i>TargetValuePtr</i> バッファ内に戻すのに使用できるバイト数を示す値を指すポインター。データが分割して検索されている場合、これにはまだ残っているバイト数が入ります。</p> <p>列のデータ値が NULL の場合、値は <code>SQL_NULL_DATA</code> です。このポインターが NULL で、<code>SQLFetch()</code> で NULL データを含む列を取得した場合、この関数はこのことを報告する手段がないので失敗します。</p> <p><code>SQLFetch()</code> がバイナリー・データを含む列を取り出した場合、<i>StrLen_or_IndPtr</i> を指すポインターが NULL であってはなりません。</p> <p>NULL の場合、この関数は <i>TargetValuePtr</i> バッファに取り出されたデータの長さについてアプリケーションに通知する他の手段がないので失敗します。</p>

注: *TargetValuePtr* がメモリー内の *StrLen_or_IndPtr* の後に連続して設定されると、DB2 CLI のパフォーマンスは多少強化されます。

使用法:

DB2 データ・ソースが異なれば、`SQLGetData()` の使用法に関する制限も異なります。アプリケーションは、この関数の機能を確認するには、以下の

`SQL_GETDATA_EXTENSIONS` オプションのいずれかを指定して `SQLGetInfo()` を呼び出す必要があります。

- `SQL_GD_ANY_COLUMN`: このオプションが戻された場合、最後にバインドされた列より前のものも含め、すべてのアンバインド済み列に対して `SQLGetData()` を呼び出すことができます。すべての DB2 データ・ソースがこの機能をサポートします。
- `SQL_GD_ANY_ORDER`: このオプションが戻された場合、任意の順序でアンバインド済み列に対して `SQLGetData()` を呼び出すことができます。すべての DB2 データ・ソースがこの機能をサポートします。
- `SQL_GD_BLOCK`: `SQL_GETDATA_EXTENSIONS InfoType` 引き数の場合に `SQLGetInfo()` でこのオプションが戻された場合、行セットのサイズが 1 より大きければ、ドライバーは `SQLGetData()` の呼び出しをサポートします。アプリケーションは `SQL_POSITION` オプションを指定した `SQLSetPos()` を呼び出して、カーソルを正

しい行上に置いてから SQLGetData() を呼び出すこともできます。少なくとも DB2 UDB (Unix および Windows 版) のデータ・ソースは、この機能をサポートします。

- **SQL_GD_BOUND**: このオプションが戻された場合、バインド済み列ならびにアンバインド済み列に対して SQLGetData() を呼び出すことができます。DB2 UDB は現在この機能をサポートしていません。

C データ・タイプ (*TargetType*) が SQL_C_CHAR、SQL_C_BINARY、SQL_C_DBCHAR、SQL_C_WCHAR であるか、または *TargetType* が SQL_C_DEFAULT で列タイプがバイナリーまたは文字ストリングを示している場合、SQLGetData() を使用して長い列を取り出すこともできます。

SQLGetData() の呼び出しごとに、戻りに使用できるデータが *BufferLength* 以上の場合には、切り捨てが行われます。切り捨ては、関数戻りコード SQL_SUCCESS_WITH_INFO とデータ切り捨てを示す SQLSTATE で示されます。アプリケーションは、同じ *ColumnNumber* 値を指定して SQLGetData() を再度呼び出し、アンバインドされた同じ列から切り捨て時以降のデータを取得することができます。列全体を取得するには、関数が SQL_SUCCESS を返すまでアプリケーションがこのような呼び出しを繰り返します。次の SQLGetData() 呼び出しは、SQL_NO_DATA_FOUND を返します。

SQLGetData() は LOB 列データの順次検索に使用できますが、LOB データのごく一部または LOB 列データの少数のセクションが必要な場合には、DB2 CLI LOB 関数を使用してください。

1. LOB ロケーターに列をバインドします。
2. 行を取り出します。
3. SQLGetSubString() 呼び出しにロケーターを使用して、データを分割して検索してください (一部の引き数の値を判別するために、SQLGetLength() および SQLGetPosition() が必要になることがあります)。
4. ステップ 2 を繰り返します。

切り捨ては、SQL_ATTR_MAX_LENGTH ステートメント属性によっても影響されません。アプリケーションは、SQL_ATTR_MAX_LENGTH および列ごとに返される最大長の値を指定して SQLSetStmtAttr() を呼び出し、同サイズ (に NULL 終止符を加えたもの) の *TargetValuePtr* バッファを割り振って、切り捨てが報告されないように指定することができます。列データが設定された最大長より大きい場合、SQL_SUCCESS が返され、実際の長さではなく最大長が *StrLen_or_IndPtr* に返されます。

取り出しによって列データの部分を廃棄するために、アプリケーションは *ColumnNumber* を次の対象列の位置に設定して SQLGetData() を呼び出すことができます。行全体として取り出されなかったデータを廃棄するには、アプリケーションで SQLFetch() を呼び出してカーソルを次の行に進めなければなりません。結果セットか

らのデータがこれ以上必要ない場合は、`SQL_CLOSE` または `SQL_DROP` を指定した `SQLCloseCursor()` または `SQLFreeStmt()` を呼び出してカーソルをクローズしてもかまいません。

TargetType 入力引き数は、*TargetValuePtr* で指示されたストレージに列データを入れる前に、必要なデータ変換 (存在する場合) のタイプを判別します。

SQL グラフィック列データの場合、以下ようになります。

- *TargetValuePtr* バッファの長さ (*BufferLength*) は、2 の倍数にします。アプリケーションは、最初に `SQLDescribeCol()` または `SQLColAttribute()` を呼び出して、列の SQL データ・タイプを判別することができます。
- DB2 CLI は *StrLen_or_IndPtr* 内に保管されているオクテット数を保管するので、*TargetValuePtr* を指すポインタを NULL にすることはできません。
- データを分割して取り出す場合、DB2 CLI は *TargetValuePtr* の値以下の最も大きい 2 オクテットの倍数まで *BufferLength* を埋め込もうとします。このことは、*BufferLength* が 2 の倍数でない場合にそのバッファ内の最後のバイトには処理を行わないことを意味します。DB2 CLI は、2 バイト文字を分割しません。

検索する列データがバイナリーでないか、または列の SQL データ・タイプがグラフィック (DBCS) であって C バッファ・タイプが `SQL_C_CHAR` であると、*TargetValuePtr* に返される内容は常に NULL 終了です。アプリケーションが複数の部分に分けてデータを検索している場合は、適切な調整を行う必要があります (たとえば、NULL 終了環境属性が有効であると想定して、各部分を連結し直す前に NULL 終了文字を除去します)。

切り捨てが小数点の右側の桁数に行われている場合、数値データ・タイプの切り捨ては警告として報告されます。切り捨てが小数点の左側で行われると、エラーが返されます (診断のセクションを参照)。

両方向スクロール・カーソルは例外ですが、データを検索するのに `SQLFetchScroll()` を使用するアプリケーションが `SQLGetData()` を呼び出すべきなのは、行セット・サイズが 1 (`SQLFetch()` を発行するのと同じ) のときだけです。`SQLGetData()` は、カーソルが現在置かれている行の列データだけを取り出せます。

両方向スクロール・カーソルでの `SQLGetData()` の使用

`SQLGetData()` は両方向スクロール・カーソルとともに使用することもできます。結果セットにある任意の行へのポインタを、ブックマークを付けて保管することができます。アプリケーションは、そのブックマークを相対位置として使用して、情報の行セットを検索します。

`SQLSetPos()` を使用して、行セット内の行へカーソルをいったん位置決めすれば、`SQLGetData()` を使用して列 0 からブックマーク値を得ることができます。多くの場

合、列 0 をバインドして行ごとのブックマーク値を検索する必要はありませんが、SQLGetData() を使用すると必要な特定行のブックマーク値を検索することができます。

戻りコード:

- SQL_SUCCESS
- SQL_SUCCESS_WITH_INFO
- SQL_STILL_EXECUTING
- SQL_ERROR
- SQL_INVALID_HANDLE
- SQL_NO_DATA_FOUND

先行の SQLGetData() 呼び出しでこの列のすべてのデータが検索されると、SQL_NO_DATA_FOUND が返されます。

SQLGetData() で長さゼロの字符串が取り出されると、SQL_SUCCESS が返されます。この場合、StrLen_or_IndPtr に 0 が入れられ、TargetValuePtr に NULL 終止符が入られます。

直前の SQLFetch() 呼び出しが失敗した場合、結果は定義されないので SQLGetData() を呼び出すことはできません。

診断:

表 211. SQLGetData SQLSTATE

SQLSTATE	説明	解説
01004	データが切り捨てられました。	指定された列 (ColumnNumber) に返されたデータは切り捨てられました。字符串または数値は右方切り捨てされます。SQL_SUCCESS_WITH_INFO が返されます。
07006	無効な変換です。	引き数 TargetType で指定された C データ・タイプにデータ値を変換することはできません。 この関数は以前と同じ ColumnNumber 値に対して呼び出されましたが、TargetType 値が異なります。
07009	記述子索引が無効です。	ColumnNumber に指定された値が 0 と同等であり、SQL_ATTR_USE_BOOKMARKS ステートメント属性が SQL_UB_OFF でした。引き数 ColumnNumber に指定された値は、結果セット内の列数より大きい値でした。
22002	無効な出力または標識バッファが指定されました。	引き数 StrLen_or_IndPtr に指定されたポインター値は NULL ポインターで、列の値は NULL です。SQL_NULL_DATA を報告する手段はありません。
22003	数値が範囲外です。	列の数値を (数値または字符串として) 返したため、数値の整数部分が切り捨てられたと考えられます。

SQLGetData

表 211. *SQLGetData* *SQLSTATE* (続き)

SQLSTATE	説明	解説
22005	割り当てにエラーがありました。	返された値は、引き数 <i>TargetType</i> で指示されたデータ・タイプと互換性がありませんでした。
22007	無効な日付時刻形式です。	文字ストリングから日時フォーマットへの変換が指定されましたが、無効なストリング表示または値が指定されたか、あるいは値が無効な日付になっています。
22008	日時フィールドがオーバーフローしました。	日時フィールドのオーバーフローが発生しました。たとえば、日付またはタイム・スタンプに関する算術演算で有効な日付の範囲内にはない結果になるか、あるいは、バインドされた変数が小さすぎて、その変数に日時値を割り当てることができません。
24000	カーソル状態が無効です。	直前の <i>SQLFetch()</i> の結果が <i>SQL_ERROR</i> であったか、または <i>SQL_NO_DATA</i> が検出されました。その結果、カーソルは行に置かれていません。
40003 08S01	通信リンクに障害が起きました。	アプリケーションとデータ・ソースとの間の通信リンクが、関数の完了する前に失敗しました。
58004	予期しないシステム障害です。	回復不能システム・エラー。
HY001	メモリーの割り振りが失敗しました。	DB2 CLI は、この関数の実行または完了をサポートするのに必要なメモリーを割り当てられません。プロセス・レベルのメモリーがアプリケーション・プロセスに使い尽くされた可能性があります。プロセス・レベルのメモリー制限については、オペレーティング・システムの構成を調べてください。
HY003	プログラム・タイプが範囲外です。	<i>TargetType</i> は、有効なデータ・タイプまたは <i>SQL_C_DEFAULT</i> ではありませんでした。

表 211. SQLGetData SQLSTATE (続き)

SQLSTATE	説明	解説
HY010	関数のシーケンス・エラーです。	<p>指定された <i>StatementHandle</i> がカーソル定位置状態にありませんでした。最初に <i>SQLFetch</i> を呼び出さずに、この関数を呼び出しました。実行時データ (<i>SQLParamData()</i>、<i>SQLPutData()</i>) 操作中に、関数が呼び出されました。</p> <p>BEGIN COMPOUND と END COMPOUND の SQL 操作中に、関数が呼び出されました。</p> <p>非同期で実行中の関数 (この関数ではない) が、<i>StatementHandle</i> で呼び出されましたが、この関数の呼び出し時にはまだ実行中でした。</p> <p>ステートメント・ハンドル上のステートメントが準備される前にこの関数が呼び出されました。</p>
HY013	予期しない、メモリーのハンドル・エラーです。	DB2 CLI は、この関数の実行または完了をサポートするために必要なメモリーにアクセスできませんでした。
HY090	文字列またはバッファの長さが無効です。	引き数 <i>BufferLength</i> の値は 0 より小さく、引き数 <i>TargetType</i> は SQL_C_CHAR、SQL_C_BINARY、SQL_C_DBCHAR (または SQL_C_DEFAULT で、デフォルト・タイプ SQL_C_CHAR、SQL_C_BINARY、または SQL_C_DBCHAR のいずれか) です。
HYC00	ドライバが使用できません。	<p>指定されたデータ・タイプの SQL データ・タイプは認識されますが、DB2 CLI ではサポートされません。</p> <p>SQL データ・タイプからアプリケーション・データ <i>TargetType</i> への要求された変換を、DB2 CLI またはデータ・ソースで行うことはできません。</p> <p>列は <i>SQLBindFileToCol()</i> を使用してバインドされました。</p>
HYT00	Timeout expired.	データ・ソースが結果セットを返す前に、タイムアウト期間が満了しました。タイムアウト期間は、 <i>SQLSetStmtAttr()</i> の SQL_ATTR_QUERY_TIMEOUT 属性を使用して設定することができます。

制約事項:

なし。

例:

```
/* use SQLGetData to get the results */
/* get data from column 1 */
cliRC = SQLGetData(hstmt,
                  1,
                  SQL_C_SHORT,
                  &deptnumb.val,
                  0,
                  &deptnumb.ind);
STMT_HANDLE_CHECK(hstmt, hdbc, cliRC);

/* get data from column 2 */
cliRC = SQLGetData(hstmt,
                  2,
                  SQL_C_CHAR,
                  location.val,
                  15,
                  &location.ind);
```

関連概念:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『DB2 CLI 用の SQLSTATES』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『CLI アプリケーションのカーソル』

関連資料:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『SQLBindCol 関数 (CLI) - アプリケーション変数または LOB ロケーターへの列のバインド』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『SQLFetch 関数 (CLI) - 次の行の取り出し』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『SQLFetchScroll 関数 (CLI) - すべてのバインド列の行セットの取り出しとデータの戻り』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『SQLSetStmtAttr 関数 (CLI) - ステートメントに関連したオプションの設定』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『CLI 関数戻りコード』

関連サンプル:

- 『dtlob.c -- How to read and write LOB data (CLI)』
- 『tbread.c -- How to read data from tables (CLI)』
- 『tut_read.c -- How to read data from tables (CLI)』

SQLGetDescField 関数 (CLI) - 記述子レコードの単一フィールド設定の取得

目的:

仕様:	DB2 CLI 5.0	ODBC 3.0	ISO CLI
-----	-------------	----------	---------

SQLGetDescField() は、記述子レコードの単一フィールドの現行設定値を返します。

同等のユニコード: この関数はユニコード文字セットとともに使用することもできます。これに対応するユニコード関数は SQLGetDescFieldW() です。ANSI からユニコード関数へのマッピングの詳細は、Unicode 関数 (CLI) を参照してください。

構文:

```
SQLRETURN SQLGetDescField (
    SQLHDESC          DescriptorHandle,
    SQLSMALLINT       RecNumber,
    SQLSMALLINT       FieldIdentifier,
    SQLPOINTER        ValuePtr,           /* Value */
    SQLINTEGER        BufferLength,
    SQLINTEGER        *StringLengthPtr); /* *StringLength */
```

関数引き数:

表 212. SQLGetDescField 引き数

データ・タイプ	引き数	使用法	説明
SQLHDESC	<i>DescriptorHandle</i>	入力	記述子ハンドル。
SQLSMALLINT	<i>RecNumber</i>	入力	アプリケーションが探索する情報を収めた記述子レコードを指定します。記述子レコードは 0 から数え、レコード番号 0 がブックマーク・レコードになります。 <i>FieldIdentifier</i> 引き数が記述子ヘッダー・レコードのフィールドを指定する場合、 <i>RecNumber</i> は 0 でなければなりません。 <i>RecNumber</i> が SQL_DESC_COUNT より小さく、行に列またはパラメーターのデータが含まれない場合、SQLGetDescField() の呼び出しはフィールドのデフォルト値を返します。
SQLSMALLINT	<i>FieldIdentifier</i>	入力	値を返す記述子のフィールドを指定します。
SQLPOINTER	<i>ValuePtr</i>	出力	記述子情報を返す先のバッファを指すポインタです。データ・タイプは、 <i>FieldIdentifier</i> の値により異なります。

SQLGetDescField

表 212. *SQLGetDescField* 引き数 (続き)

データ・タイプ	引き数	使用法	説明
SQLINTEGER	<i>BufferLength</i>	入力	<ul style="list-style-type: none">• <i>ValuePtr</i> が文字ストリングを指す場合、この引き数の長さは *<i>ValuePtr</i> になります。• <i>ValuePtr</i> がポインターであってもストリングを指していない場合、 <i>BufferLength</i> の値は SQL_IS_POINTER になります。• *<i>ValuePtr</i> の値がユニコード・データ・タイプである場合、 <i>BufferLength</i> 引き数は偶数でなければなりません。
SQLSMALLINT *	<i>StringLengthPtr</i>	出力	* <i>ValuePtr</i> 内に戻すために使用できる総バイト数 (NULL 終了文字に必要なバイト数を除く) を指すポインター。

使用法:

アプリケーションは、SQLGetDescField() を呼び出して、記述子レコードの単一フィールドの値を返すことができます。SQLGetDescField() への呼び出しでは、ヘッダー・フィールド、レコード・フィールド、およびブックマーク・フィールドを含むすべての記述子タイプのどのフィールドの設定でも返すことができます。SQLGetDescField() を繰り返し呼び出すなら、アプリケーションは同じ記述子または異なる記述子にある複数のフィールドの設定を、任意の順序で獲得することができます。SQLGetDescField() を呼び出して DB2 CLI 定義済み記述子フィールドを返すこともできます。

パフォーマンス上の理由で、ステートメントを実行する前にアプリケーションは IRD の SQLGetDescField() を呼び出してはなりません。そのような時点で SQLGetDescField() を呼び出すと、CLI ドライバーがステートメントを記述するので、さらに余計なネットワーク・フローを生じることになります。据え置き準備がオンになっているときに SQLGetDescField() を呼び出すと、記述情報を得るためにサーバーでステートメントを準備する必要が生じるため、据え置き準備の利点が失われます。

名前、データ・タイプ、および列またはパラメーター・データのストレージを記述できる複数のフィールドの設定を、SQLGetDescRec() への単一呼び出しで検索することもできます。SQLGetStmtAttr() を呼び出して、ステートメント属性が関連付けられた記述子ヘッダー内の単一のフィールドの値を返すことができます。

特定の記述子タイプが未定義になっているフィールドの値を検索するためにアプリケーションが SQLGetDescField() を呼び出すと、この関数は SQLSTATE HY091 (無効な記述子フィールド ID) を返します。特定の記述子タイプが定義されているのに、デフォルト値をもっておらず、しかもまだ設定されていないフィールドの値を検索するためにアプリケーションが SQLGetDescField() を呼び出すと、この関数は SQL_SUCCESS を

返しますがフィールドに返される値は未定義になります。存在するデフォルト値を確認するには、記述子フィールドの初期化値のリストを参照してください。

SQL_DESC_ALLOC_TYPE ヘッダー・フィールドは、読み取り専用として使用可能です。このフィールドは、すべてのタイプの記述子に定義されます。

これらのフィールドはそれぞれ、IRD 専用か、または IRD と IPD の両方のどちらかに定義されます。

SQL_DESC_AUTO_UNIQUE_VALUE	SQL_DESC_LITERAL_SUFFIX
SQL_DESC_BASE_COLUMN_NAME	SQL_DESC_LOCAL_TYPE_NAME
SQL_DESC_CASE_SENSITIVE	SQL_DESC_SCHEMA_NAME
SQL_DESC_CATALOG_NAME	SQL_DESC_SEARCHABLE
SQL_DESC_DISPLAY_SIZE	SQL_DESC_TABLE_NAME
SQL_DESC_FIXED_PREC_SCALE	SQL_DESC_TYPE_NAME
SQL_DESC_LABEL	SQL_DESC_UNSIGNED
SQL_DESC_LITERAL_PREFIX	SQL_DESC_UPDATABLE

上記のフィールドの詳細は、記述子 *FieldIdentifier* の値のリストを参照してください。

戻りコード:

- SQL_SUCCESS
- SQL_SUCCESS_WITH_INFO
- SQL_ERROR
- SQL_NO_DATA
- SQL_INVALID_HANDLE

RecNumber が記述子レコードの数よりも大きい場合は、SQL_NO_DATA が返されます。

DescriptorHandle が IRD ハンドルであり、ステートメントが準備済みまたは実行済み状態にあるが、それと関連するオープン・カーソルがない場合、SQL_NO_DATA が返されます。

診断:

表 213. SQLGetDescField SQLSTATE

SQLSTATE	説明	解説
01000	警告 !	通知メッセージ。(関数は、SQL_SUCCESS_WITH_INFO を返します。)
01004	データが切り捨てられました。	バッファー <i>*ValuePtr</i> には記述子フィールド全体を返すのに十分な大きさがなかったため、フィールドが切り捨てられました。切り捨てられていない記述子フィールドの長さが、 <i>StringLengthPtr</i> に返されます。(関数は、SQL_SUCCESS_WITH_INFO を返します。)

SQLGetDescField

表 213. *SQLGetDescField* *SQLSTATE* (続き)

SQLSTATE	説明	解説
07009	記述子索引が無効です。	<p><i>RecNumber</i> 引き数に指定された値は 1 より小さく、<i>SQL_ATTR_USE_BOOKMARKS</i> ステートメント属性は <i>SQL_UB_OFF</i> であり、フィールドはヘッダー・フィールドまたは DB2 CLI 定義済みフィールドではありませんでした。</p> <p><i>FieldIdentifier</i> 引き数はレコード・フィールドであり、<i>RecNumber</i> 引き数は 0 でした。</p> <p><i>RecNumber</i> 引き数は 0 より小さく、フィールドはヘッダー・フィールドまたは DB2 CLI 定義済みフィールドではありませんでした。</p>
08S01	通信リンクに障害が起きました。	関数が処理を完了する前に、DB2 CLI とその接続先データ・ソースとの間の通信リンクが失敗しました。
HY000	General error.	特定の <i>SQLSTATE</i> がなかった場合のエラーが発生しました。 <i>SQLGetDiagRec()</i> から <i>*MessageText</i> バッファ内に戻されたエラー・メッセージに、エラーとその原因が説明されています。
HY001	メモリーの割り振りが失敗しました。	DB2 CLI は、この関数の実行または完了をサポートするのに必要なメモリーを割り当てられません。プロセス・レベルのメモリーがアプリケーション・プロセスに使い尽くされた可能性があります。プロセス・レベルのメモリー制限については、オペレーティング・システムの構成を調べてください。
HY007	関連ステートメントが準備されていません。	<i>DescriptorHandle</i> は IRD と関連付けられており、関連付けられているステートメント・ハンドルが準備済みまたは実行済みの状態にはありません。
HY010	関数のシーケンス・エラーです。	<p><i>DescriptorHandle</i> が関連付けられていた <i>StatementHandle</i> で、非同期的に実行されるある関数 (この関数ではない) が呼び出され、この関数が呼び出された時点でまだ実行中でした。</p> <p><i>DescriptorHandle</i> に関連する <i>StatementHandle</i> で <i>SQLExecute()</i> または <i>SQLExecDirect()</i> が呼び出され、<i>SQL_NEED_DATA</i> が返されました。すべての実行時データ・パラメーターまたは列用のデータの送信前に、この関数が呼び出されました。</p>
HY013	予期しない、メモリーのハンドル・エラーです。	DB2 CLI は、この関数の実行または完了をサポートするために必要なメモリーにアクセスできませんでした。

表 213. SQLGetDescField SQLSTATE (続き)

SQLSTATE	説明	解説
HY021	記述子情報が矛盾します。	整合性検査時にチェックされた記述子情報は、整合性がとれていませんでした。
HY090	文字列またはバッファの長さが無効です。	名前引きの長さのうち 1 つの値は 0 より小さい値でしたが、SQL_NTS と等しくありませんでした。
HY091	記述子フィールド ID が無効です。	<i>DescriptorHandle</i> には <i>FieldIdentifier</i> が定義されていません。 SQL_DESC_COUNT フィールド内の値よりも大きい値が <i>RecNumber</i> 引き数に指定されました。

制約事項:

なし。

例:

```

/* see how the field SQL_DESC_PARAMETER_TYPE is set */
cliRC = SQLGetDescField(hIPD,
                        1, /* look at the parameter */
                        SQL_DESC_PARAMETER_TYPE,
                        &descFieldParameterType, /* result */
                        SQL_IS_SMALLINT,
                        NULL);

/* ... */

/* see how the descriptor record field SQL_DESC_TYPE_NAME is set */
rc = SQLGetDescField(hIRD,
                    (SQLSMALLINT)colCount,
                    SQL_DESC_TYPE_NAME, /* record field */
                    &descFieldTypeName, /* result */
                    25,
                    NULL);

/* ... */

/* see how the descriptor record field SQL_DESC_LABEL is set */
rc = SQLGetDescField(hIRD,
                    (SQLSMALLINT)colCount,
                    SQL_DESC_LABEL, /* record field */
                    &descFieldLabel, /* result */
                    25,
                    NULL);

```

関連概念:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『Unicode 関数 (CLI)』

SQLGetDescField

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『DB2 CLI 用の SQLSTATES』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『CLI アプリケーションの記述子』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『CLI アプリケーションの記述子の整合性検査』

関連資料:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『SQLGetDescRec 関数 (CLI) - 記述子レコードの複数フィールド設定の取得』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『SQLGetStmtAttr 関数 (CLI) - ステートメント属性の現行設定値の取得』
- 819 ページの『記述子ヘッダーとレコード・フィールドの初期設定値 (CLI)』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『記述子 FieldIdentifier 引き数の値 (CLI)』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『CLI 関数戻りコード』

関連サンプル:

- 『dbuse.c -- How to use a database (CLI)』

SQLSetDescField 関数 (CLI) - 記述子レコードの単一フィールドの設定

目的:

仕様:	DB2 CLI 5.0	ODBC 3.0	ISO CLI
-----	-------------	----------	---------

SQLSetDescField() は、記述子レコードの単一フィールドの値を設定します。

同等のユニコード: この関数はユニコード文字セットとともに使用することもできます。これに対応するユニコード関数は SQLSetDescFieldW() です。ANSI からユニコード関数へのマッピングの詳細は、Unicode 関数 (CLI) を参照してください。

構文:

```
SQLRETURN SQLSetDescField (SQLHDESC  
                            SQLSMALLINT  
                            SQLSMALLINT  
                            SQLPOINTER  
                            SQLINTEGER  
                            DescriptorHandle,  
                            RecNumber,  
                            FieldIdentifier,  
                            ValuePtr,  
                            BufferLength);
```

関数引き数:

表 214. *SQLSetDescField* 引き数

データ・タイプ	引き数	使用法	説明
SQLHDESC	<i>DescriptorHandle</i>	入力	記述子ハンドル。
SQLSMALLINT	<i>RecNumber</i>	入力	アプリケーションが設定しようとしているフィールドを含んでいる記述子レコードを示します。記述子レコードは 0 から数え、レコード番号 0 がブックマーク・レコードになります。 <i>RecNumber</i> 引き数は、ヘッダー・フィールドの場合には無視されます。
SQLSMALLINT	<i>FieldIdentifier</i>	入力	値を設定しようとしている記述子のフィールドを示します。詳細は、記述子の <i>FieldIdentifier</i> 引き数の値リストを参照してください。
SQLPOINTER	<i>ValuePtr</i>	入力	記述子情報が入っているバッファへのポインター、または 4 バイトからなる値。データ・タイプは、 <i>FieldIdentifier</i> の値により異なります。 <i>ValuePtr</i> が 4 バイトからなる値である場合、 <i>FieldIdentifier</i> 引き数の値に応じて、その 4 バイトすべてが使われるか、あるいは 4 バイトのうち 2 つだけが使われます。

SQLSetDescField

表 214. *SQLSetDescField* 引き数 (続き)

データ・タイプ	引き数	使用法	説明
SQLINTEGER	<i>BufferLength</i>	入力	<p><i>FieldIdentifier</i> が ODBC で定義されたフィールドであり、かつ <i>ValuePtr</i> が文字ストリングかバイナリー・バッファーを指している場合、この引き数は <i>*ValuePtr</i> の長さでなければなりません。また、<i>FieldIdentifier</i> が ODBC で定義されたフィールドであり、かつ <i>ValuePtr</i> が整数である場合には、<i>BufferLength</i> は無視されます。</p> <p><i>FieldIdentifier</i> がドライバーで定義されたフィールドである場合には、アプリケーションは <i>BufferLength</i> 引き数を設定してそのフィールドの性質を示します。 <i>BufferLength</i> には以下の値が有効です。</p> <ul style="list-style-type: none"> • <i>ValuePtr</i> が文字ストリングを指すポインターの場合、 <i>BufferLength</i> はストリングまたは SQL_NTS の長さです。 • <i>ValuePtr</i> がバイナリー・バッファーを指すポインターの場合、アプリケーションは SQL_LEN_BINARY_ATTR(length) マクロの結果を <i>BufferLength</i> に入れます。それによって、<i>BufferLength</i> には負の値が入ります。 • <i>ValuePtr</i> が文字ストリングやバイナリー・ストリング以外の値へのポインターであれば、<i>BufferLength</i> には必ず値 SQL_IS_POINTER が入ります。 • <i>ValuePtr</i> に固定長の値が入っている場合、<i>BufferLength</i> は状況に応じて SQL_IS_INTEGER、SQL_IS_UNSIGNED_INTEGER、SQL_IS_SMALLINT、または SQL_IS_USMALLINT のいずれかとなります。

使用法:

アプリケーションは *SQLSetDescField()* を呼び出すことにより、任意の記述子フィールドを一度に 1 つずつ設定できます。 *SQLSetDescField()* への 1 回の呼び出しで、単一の記述子にある単一のフィールドを設定します。その対象のフィールドが設定可能なも

のであれば、この関数を呼び出して、任意の記述子タイプの任意のフィールドを設定できます。詳しくは、記述子ヘッダーとレコード・フィールド初期設定の値を参照してください。

注: SQLSetDescField() への呼び出しが失敗した場合、*RecNumber* 引き数に示された記述子レコードの内容は定義されません。

この関数の 1 回の呼び出しで、他の関数を呼び出して複数の記述子フィールドを設定することができます。SQLSetDescRec() 関数は、列やパラメーターにバインドされたデータ・タイプおよびバッファーに影響する様々なフィールド (SQL_DESC_TYPE、SQL_DESC_DATETIME_INTERVAL_CODE、SQL_DESC_OCTET_LENGTH、SQL_DESC_PRECISION、SQL_DESC_SCALE、SQL_DESC_DATA_PTR、SQL_DESC_OCTET_LENGTH_PTR、および SQL_DESC_INDICATOR_PTR フィールド) を設定します。また、SQLBindCol() や SQLBindParameter() を使用すれば、列やパラメーターをバインドするための完全な仕様を作成できます。これらの関数はそれぞれ、1 回の関数呼び出しで、特定のグループの記述子フィールドを設定します。

SQLSetDescField() を呼び出してから、バインド用ポインター (SQL_DESC_DATA_PTR、SQL_DESC_INDICATOR_PTR、または SQL_DESC_OCTET_LENGTH_PTR) にオフセットを追加すれば、バインド用バッファーを変更することができます。こうすれば、SQLBindCol() や SQLBindParameter() を呼び出さなくてもバインド用バッファーが変更されます。これでアプリケーションは、たとえば SQL_DESC_DATA_TYPE などの他のフィールドを変更するという手間をかけずに済むので、SQL_DESC_DATA_PTR を迅速に変更できるようになります。

記述子のヘッダー・フィールドの設定は、*RecNumber* を 0 にして SQLSetDescField() を呼び出し、さらに適切な *FieldIdentifier* を呼び出して行います。多くのヘッダー・フィールドにはステートメント属性が入っていますが、それらも SQLSetStmtAttr() への呼び出しで設定できます。これにより、アプリケーションは最初に記述子ハンドルを取得することなく、ステートメント属性を設定することが可能です。*RecNumber* の 0 は、ブックマーク・フィールドの設定にも使用します。

注: ステートメント属性 SQL_ATTR_USE_BOOKMARKS は常に、SQLSetDescField() を呼び出してブックマーク・フィールドを設定する前に設定する必要があります。これは必須ではありませんが、強くお勧めします。

記述子フィールドの設定順序

SQLSetDescField() を呼び出して記述子フィールドを設定する場合、アプリケーションは以下に示す特定の順序に従う必要があります。

- アプリケーションはまず最初に SQL_DESC_TYPE、SQL_DESC_CONCISE_TYPE、または SQL_DESC_DATETIME_INTERVAL_CODE フィールドを設定する必要があります。

注: SQL_DESC_DATETIME_INTERVAL_CODE は、ODBC では定義されていますが、DB2 CLI ではサポートされていません。

- これらのフィールドのいずれかを設定したなら、アプリケーションはデータ・タイプの属性を設定することができ、ドライバーはデータ・タイプの属性フィールドをそのデータ・タイプの適切なデフォルト値に設定します。タイプ属性フィールドのデフォルト値が自動的に設定されることにより、アプリケーションがデータ・タイプを指定すると、記述子が常に使用できるようになっています。アプリケーションが明示的にデータ・タイプ属性を設定すると、デフォルトの属性はオーバーライドされます。
- ステップ 1 に示されているいずれかのフィールドが設定され、データ・タイプ属性も設定された後は、アプリケーションは SQL_DESC_DATA_PTR を設定できます。これにより、記述子フィールドの整合性検査に対する指示が出ます。アプリケーションが SQL_DESC_DATA_PTR フィールドの設定後にデータ・タイプや属性を変更すると、ドライバーは SQL_DESC_DATA_PTR を NULL ポインターに設定して、そのレコードをアンバインドします。こうなると、アプリケーションは適切なステップを順番どおりに実行しないと記述子レコードが使用できません。

記述子フィールドの初期設定

記述子を割り当てる時点で、その記述子内のフィールドはデフォルト値に初期設定したり、デフォルト値なしで初期設定したり、あるいは記述子のタイプを定義しないでおくことができます。詳細は、記述子ヘッダーとレコード・フィールド初期設定の値を参照してください。

IRD のフィールドにデフォルト値があるのは、ステートメントが作成または実行され、その IRD が移植された後だけであり、ステートメントのハンドルまたは記述子が割り当てられた時点ではありません。IRD が移植されてしまうまでは、IRD のフィールドにアクセスしようとするとエラーが返されます。

一部の記述子フィールドは、1 つまたは複数 (ただし全部ではない) の記述子タイプに対して定義されます (ARD と IRD、および APD と IPD)。フィールドを記述子のタイプに対して定義していないと、どの関数もその記述子を使用する必要がなくなります。記述子は実際のデータ構造ではなく、データの論理ビューであるため、これらの余分のフィールドは定義済みフィールドに対して何の影響もありません。

SQLGetDescField() がアクセスできるフィールドは、必ずしも SQLSetDescField() によって設定されるとはかぎりません。SQLSetDescField() で設定できるフィールドは、記述子ヘッダーおよびレコード・フィールド初期設定の値のリストで説明されています。

戻りコード:

- SQL_SUCCESS
- SQL_SUCCESS_WITH_INFO
- SQL_ERROR
- SQL_INVALID_HANDLE

診断:

表 215. SQLSetDescField SQLSTATE

SQLSTATE	説明	解説
01000	通常の警告	通知メッセージ。(関数は、SQL_SUCCESS_WITH_INFO を返します。)
01S02	オプション値が変更されました。	SQL の制約もしくは要件が原因で、*ValuePtr に指定された値 (ValuePtr がポインターだった場合) または ValuePtr 内の値 (ValuePtr が 4 バイトからなる値だった場合) を DB2 CLI がサポートしていなかったか、*ValuePtr が無効だったため、DB2 CLI がそれに近い値で置き換えました。(関数は、SQL_SUCCESS_WITH_INFO を返します。)
07009	記述子索引が無効です。	FieldIdentifier 引き数はヘッダー・フィールドでしたが、RecNumber 引き数が 0 ではありませんでした。 RecNumber 引き数が 0 で、DescriptorHandle は IPD でした。 RecNumber 引き数は 0 未満でした。
08S01	通信リンクに障害が起きました。	関数が処理を完了する前に、DB2 CLI とその接続先データ・ソースとの間の通信リンクが失敗しました。
HY000	General error.	特定の SQLSTATE がなかった場合のエラーが発生しました。SQLGetDiagRec() から *MessageText バッファ内に戻されたエラー・メッセージに、エラーとその原因が説明されています。
HY001	メモリーの割り振りが失敗しました。	DB2 CLI は、この関数の実行または完了をサポートするのに必要なメモリーを割り当てられません。プロセス・レベルのメモリーがアプリケーション・プロセスに使い尽くされた可能性があります。プロセス・レベルのメモリー制限については、オペレーティング・システムの構成を調べてください。
HY010	関数のシーケンス・エラーです。	DescriptorHandle が関連付けられていた StatementHandle で、非同期的に実行されるある関数 (この関数ではない) が呼び出され、この関数が呼び出された時点でまだ実行中でした。 DescriptorHandle と関連する StatementHandle で SQLExecute() または SQLExecDirect() が呼び出され、SQL_NEED_DATA が戻されました。すべての実行時データ・パラメーターまたは列用のデータの送信前に、この関数が呼び出されました。

SQLSetDescField

表 215. *SQLSetDescField* *SQLSTATE* (続き)

SQLSTATE	説明	解説
HY016	インプリメンテーション行の記述子を修正できません。	<i>DescriptorHandle</i> 引き数は IRD に関連付けられていたが、 <i>FieldIdentifier</i> 引き数が <i>SQL_DESC_ARRAY_STATUS_PTR</i> ではありませんでした。
HY021	記述子情報が矛盾します。	<i>TYPE</i> フィールド、あるいは記述子内で <i>TYPE</i> フィールドに関連付けられた他のフィールドが有効ではないか整合性がありません。 <i>TYPE</i> フィールドが有効な DB2 CLI C タイプではありませんでした。 整合性検査時にチェックされた記述子情報は、整合性がとれていませんでした。
HY091	記述子フィールド ID が無効です。	<i>FieldIdentifier</i> 引き数に指定された値が、DB2 CLI の定義済みフィールドではなく、定義済み値でもありませんでした。 <i>SQL_DESC_COUNT</i> フィールド内の値よりも大きい値が <i>RecNumber</i> 引き数に指定されました。 <i>FieldIdentifier</i> 引き数が <i>SQL_DESC_ALLOC_TYPE</i> でした。
HY092	オプション・タイプが範囲外です。	<i>Attribute</i> 引き数に指定された値が有効ではありませんでした。
HY105	パラメーター・タイプが無効です。	<i>SQL_DESC_PARAMETER_TYPE</i> に指定された値が無効でした。(詳しくは、 <i>SQLBindParameter()</i> の『 <i>InputOutputType</i> 引き数』の項を参照してください。)

制約事項:

なし。

例:

```
/* set a single field of a descriptor record */
rc = SQLSetDescField(hARD,
                    1,
                    SQL_DESC_TYPE,
                    (SQLPOINTER)SQL_SMALLINT,
                    SQL_IS_SMALLINT);
```

関連概念:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『Unicode 関数 (CLI)』

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『DB2 CLI 用の SQLSTATES』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『CLI アプリケーションの記述子』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『CLI アプリケーションの記述子の整合性検査』

関連資料:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『SQLBindCol 関数 (CLI) - アプリケーション変数または LOB ロケーターへの列のバインド』
- 905 ページの『SQLGetDescField 関数 (CLI) - 記述子レコードの単一フィールド設定の取得』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『SQLGetDescRec 関数 (CLI) - 記述子レコードの複数フィールド設定の取得』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『SQLSetDescRec 関数 (CLI) - 列またはパラメーター・データ用の複数の記述子フィールドの設定』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『SQLBindParameter 関数 (CLI) - バッファーまたは LOB ロケーターへのパラメーター・マーカートのバインド』
- 819 ページの『記述子ヘッダーとレコード・フィールドの初期設定値 (CLI)』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『記述子 FieldIdentifier 引き数の値 (CLI)』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『CLI 関数戻りコード』

関連サンプル:

- 『dbuse.c -- How to use a database (CLI)』

SQLExtendedBind 関数 (CLI) - 列の配列のバインド

目的:

仕様:	DB2 CLI 6		
-----	-----------	--	--

SQLExtendedBind() を使用して、SQLBindCol() または SQLBindParameter() を繰り返し呼び出さずに、列またはパラメーターの配列をバインドします。

構文:

SQLExtendedBind

```

SQLRETURN      SQLExtendedBind (
                SQLHSTMT          StatementHandle, /* hstmt */
                SQLSMALLINT       fBindCol,
                SQLSMALLINT       cRecords,
                SQLSMALLINT *     pfCType,
                SQLPOINTER *      rgbValue,
                SQLINTEGER *      cbValueMax,
                SQLUIINTEGER *    puiPrecisionCType,
                SQLSMALLINT *     psScaleCType,
                SQLINTEGER **     pcbValue,
                SQLINTEGER **     piIndicator,
                SQLSMALLINT *     pfParamType,
                SQLSMALLINT *     pfSQLType,
                SQLUIINTEGER *    pcbColDef,
                SQLSMALLINT *     pibScale ) ;
    
```

関数引き数:

表 216. *SQLExtendedBind()* 引き数

データ・タイプ	引き数	使用法	説明
SQLHSTMT	<i>StatementHandle</i>	入力	ステートメント・ハンドル
SQLSMALLINT	<i>fBindCol</i>	入力	SQL_TRUE の場合、結果は <i>SQLBindCol()</i> に類似したものになります。そうでない場合は、 <i>SQLBindParameter()</i> に類似したものになります。
SQLSMALLINT	<i>cRecords</i>	入力	バインドする列またはパラメーターの数。
SQLSMALLINT *	<i>pfCType</i>	入力	アプリケーション・データ・タイプの値の配列。
SQLPOINTER *	<i>rgbValue</i>	入力	アプリケーション・データ域を指すポインタの配列。
SQLINTEGER *	<i>cbValueMax</i>	入力	<i>rgbValue</i> の最大サイズの配列。
SQLUIINTEGER *	<i>puiPrecisionCType</i>	入力	小数点精度値の配列。いずれの値も、対応するレコードのアプリケーション・データ・タイプが <i>SQL_C_DECIMAL_IBM</i> の場合のみ使用します。
SQLSMALLINT *	<i>psScaleCType</i>	入力	10 進数値の配列。いずれの値も、対応するレコードのアプリケーション・データ・タイプが <i>SQL_C_DECIMAL_IBM</i> の場合のみ使用します。
SQLINTEGER **	<i>pcbValue</i>	入力	長さの値を指すポインタの配列。
SQLINTEGER **	<i>piIndicator</i>	入力	標識の値を指すポインタの配列。

表 216. SQLExtendedBind() 引き数 (続き)

データ・タイプ	引き数	使用法	説明
SQLSMALLINT *	<i>pfParamType</i>	入力	<p>パラメーター・タイプの配列。使用されるのは、<i>fBindCol</i> が FALSE の場合だけです。</p> <p>この配列の各行は、SQLBindParameter() の引き数 <i>InputOutputType</i> と同じ目的を果たします。以下の値に設定できます。</p> <ul style="list-style-type: none"> • SQL_PARAM_INPUT • SQL_PARAM_INPUT_OUTPUT • SQL_PARAM_OUTPUT
SQLSMALLINT *	<i>pfSQLType</i>	入力	<p>SQL データ・タイプの配列。使用されるのは、<i>fBindCol</i> が FALSE の場合だけです。</p> <p>この配列の各行は、SQLBindParameter() の引き数 <i>ParameterType</i> と同じ目的を果たします。</p>
SQLINTEGER *	<i>pcbColDef</i>	入力	<p>SQL 精度値の配列。使用されるのは、<i>fBindCol</i> が FALSE の場合だけです。</p> <p>この配列の各行は、SQLBindParameter() の引き数 <i>ColumnSize</i> と同じ目的を果たします。</p>
SQLSMALLINT *	<i>pibScale</i>	入力	<p>SQL スケール値の配列。使用されるのは、<i>fBindCol</i> が FALSE の場合だけです。</p> <p>この配列の各行は、SQLBindParameter() の引き数 <i>DecimalDigits</i> と同じ目的を果たします。</p>

使用法:

引き数 *fBindCol* は、この関数呼び出しを以下のどちらの関連付け (バインド) に使用するかを決定します。

- SQL ステートメントのパラメーター・マーカ (SQLBindParameter() と同様) - *fBindCol* = SQL_FALSE
- 結果セットの列 (SQLBindCol() と同様) - *fBindCol* = SQL_TRUE

この関数は、SQLBindCol() または SQLBindParameter() に対する複数の呼び出しを置き換えるために使用できますが、重要な違いに注意してください。 *fBindCol* パラメーターの設定方法によって、SQLExtendedBind() への入力は、以下の例外を除いて SQLBindCol() または SQLBindParameter() に類似しています。

- SQLExtendedBind() が SQLBindCol() モードに設定されている場合。
 - *targetValuePtr* は、戻り列にあるデータの最大長をバイトで示す、正の整数である必要があります。

SQLExtendedBind

- SQLExtendedBind() が SQLBindParameter() モードに設定されている場合。
 - *ColumnSize* は、該当する場合、ターゲット列の最大長をバイトで示す、正の整数である必要があります。
 - *DecimalDigits* は、該当する場合、ターゲット列のための正しいスケールに設定する必要があります。
 - SQL_C_DEFAULT を *ValueType* に使用することはできません。
 - *ValueType* がロケーター・タイプの場合、対応する *ParameterType* はマッチングするロケーター・タイプでなければなりません。
 - すべての *ValueType* から *ParameterType* へのマッピングは、DB2 CLI が実行しなければならない変換を最小化するために、可能なかぎり一致する必要があります。

SQLExtendedBind() に渡すどの配列参照にも、少なくとも *cRecords* で指示されている数のエレメントが入っていなければなりません。呼び出し側のアプリケーションが十分な大きさの配列を渡さなかった場合、DB2 CLI は配列の末尾を超えて読み取りを試みることがあり、その場合はデータが壊れたり重要なアプリケーションに障害が起きたりすることになります。

SQLExtendedBind() に渡される各配列は、据え置き引き数と見なされます。つまり、配列内の値は実行時にリトリブされ、調べられます。結果として、各配列が有効な状態にあり、DB2 CLI が配列内の値を使用して実行する場合に、有効な値が含まれていることを確認します。正常な実行に続いて、ステートメントを再実行する必要がある場合、元の呼び出しから SQLExtendedBind() へ渡されたハンドルが、まだ有効な配列を参照している場合、2 度目に SQLExtendedBind() を呼び出す必要はありません。

戻りコード:

- SQL_SUCCESS
- SQL_SUCCESS_WITH_INFO
- SQL_ERROR
- SQL_INVALID_HANDLE

診断:

表 217. SQLExtendedBind() SQLSTATE

SQLSTATE	説明	解説
07006	無効な変換です。	<i>pfCType</i> 引き数の行によって識別されるデータ値から <i>pfParamType</i> 引き数によって識別されるデータ・タイプへの変換は、意味のある変換ではありません。(たとえば、SQL_C_DATE から SQL_DOUBLE への変換は無意味です。)
07009	無効な記述子索引	引き数 <i>cRecords</i> に指定された値が、結果セット内の列の最大数を超えました。

表 217. *SQLExtendedBind()* *SQLSTATE* (続き)

SQLSTATE	説明	解説
40003 08S01	通信リンクに障害が起きました。	アプリケーションとデータ・ソースとの間の通信リンクが、関数の完了する前に失敗しました。
58004	予期しないシステム障害です。	回復不能システム・エラー。
HY001	メモリーの割り振りが失敗しました。	DB2 CLI は、この関数の実行または完了をサポートするのに必要なメモリーを割り当てられません。プロセス・レベルのメモリーがアプリケーション・プロセスに使い尽くされた可能性があります。プロセス・レベルのメモリー制限については、オペレーティング・システムの構成を調べてください。
HY003	プログラム・タイプが範囲外です。	<i>pfParamType</i> または <i>pfSQLType</i> の行は、有効なデータ・タイプまたは SQL_C_DEFAULT ではありませんでした。
HY004	SQL データ・タイプが範囲外です	引き数 <i>pfParamType</i> に指定された値が、有効な SQL データ・タイプではありません。
HY009	引き数の値が無効です。	引き数 <i>rgbValue</i> は NULL ポインターで、引き数 <i>cbValueMax</i> も NULL ポインターですが、 <i>pfParamType</i> が SQL_PARAM_OUTPUT ではありません。
HY010	関数のシーケンス・エラーです。	実行時データ (SQLParamData(), SQLPutData()) 操作中に、関数が呼び出されました。 BEGIN COMPOUND と END COMPOUND の SQL 操作中に、関数が呼び出されました。
HY013	予期しない、メモリーのハンドル・エラーです。	DB2 CLI は、この関数の実行または完了をサポートするために必要なメモリーにアクセスできませんでした。
HY021	不整合な記述子情報	整合性検査時にチェックされた記述子情報は、整合性がとれていませんでした。
HY090	ストリングまたはバッファの長さが無効です。	引き数 <i>cbValueMax</i> に指定された値は 1 より小さく、 <i>pfParamType</i> または <i>pfSQLType</i> の対応する行は、SQL_C_CHAR、SQL_C_BINARY、または SQL_C_DEFAULT のいずれかです。
HY093	無効なパラメーター数です。	引き数 <i>pfCType</i> の行に指定された値が、1 より小さいか、サーバーでサポートされる最大数より大きい値でした。

SQLExtendedBind

表 217. *SQLExtendedBind()* *SQLSTATE* (続き)

SQLSTATE	説明	解説
HY094	位取りの値が無効です。	<p><i>pfParamType</i> に指定された値が <code>SQL_DECIMAL</code> または <code>SQL_NUMERIC</code> であり、<i>DecimalDigits</i> に指定された値が 0 より小さいかまたは引き数 <i>pcbColDef</i> (精度) の値より大きい値でした。</p> <p><i>pfParamType</i> に指定された値が <code>SQL_C_TIMESTAMP</code> で、<i>pfParamType</i> に指定された値が <code>SQL_CHAR</code> または <code>SQL_VARCHAR</code> のどちらかであり、<i>DecimalDigits</i> に指定された値が 0 より小さいかまたは 6 より大きい値でした。</p>
HY104	精度の値が無効です。	<p><i>pfParamType</i> に指定された値が <code>SQL_DECIMAL</code> または <code>SQL_NUMERIC</code> のどちらかで、<i>pcbColDef</i> によって指定された値が 1 より小さい値でした。</p>
HY105	パラメーター・タイプが無効です。	<p><i>pfParamType</i> が <code>SQL_PARAM_INPUT</code>、<code>SQL_PARAM_OUTPUT</code>、または <code>SQL_PARAM_INPUT_OUTPUT</code> のいずれでもありません。</p>
HYC00	ドライバーが使用できません。	<p>DB2 CLI は、<i>pfParamType</i> または <i>pfSQLType</i> の行に指定されているデータ・タイプを認識はしますが、サポートしません。</p> <p>LOB ロケーター C データ・タイプが指定されましたが、接続されているサーバーは LOB データ・タイプをサポートしていません。</p>

制約事項:

なし。

関連概念:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『DB2 CLI 用の SQLSTATES』

関連資料:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『SQLBindCol 関数 (CLI) - アプリケーション変数または LOB ロケーターへの列のバインド』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻 の『SQLBindParameter 関数 (CLI) - バッファーまたは LOB ロケーターへのパラメーター・マーカのバインド』

- ・ コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『CLI 関数戻りコード』

SQLNextResult 関数 (CLI) - 別のステートメント・ハンドルへの次の結果セットの関連付け

目的:

仕様:	DB2 CLI 7.x		
-----	-------------	--	--

SQLNextResult() を使うと、ストアード・プロシージャから戻された複数の結果セットに順不同でアクセスすることができます。

構文:

```
SQLRETURN SQLNextResult (SQLHSTMT StatementHandle1
                          SQLHSTMT StatementHandle2);
```

関数引き数:

表 218. SQLNextResult 引き数

データ・タイプ	引き数	使用法	説明
SQLHSTMT	StatementHandle1	入力	ステートメント・ハンドル
SQLHSTMT	StatementHandle2	入力	ステートメント・ハンドル

使用法:

ストアード・プロシージャは、終了後も 1 つ以上のカーソルをオープンしたままにすることで、複数の結果セットを戻します。最初の結果セットにアクセスするには、そのストアード・プロシージャを呼び出したステートメント・ハンドルを常に使います。複数の結果セットが戻された場合、SQLMoreResults() または SQLNextResult() を使って、結果セットの記述と取り出しを行うことができます。

SQLMoreResults() は、最初の結果セットのカーソルのクローズに使われるとともに、同じステートメント・ハンドルで次の結果セットを処理する手段にもなるのに対して、SQLNextResult() は、StatementHandle1 上のカーソルをクローズしないまま、次の結果セットを StatementHandle2 に移動します。取り出す結果セットがない場合、どちらの関数も SQL_NO_DATA_FOUND を戻します。

SQLNextResult() を使用した場合、他のステートメント・ハンドルに転送された後の結果セットを任意の順序で処理することができます。StatementHandle1 上にもうカーソル (オープンされた結果セット) がなくなるまで、SQLMoreResults() と SQLNextResult() を混合して呼び出すことができます。

SQLNextResult

SQLNextResult() が SQL_SUCCESS を戻すと、次の結果セットはもう StatementHandle1 には関連付けられていません。つまり、SQLExecDirect() が StatementHandle2 に対する照会を正常に完了したばかりであるかのように、次の結果セットは StatementHandle2 に関連付けられています。したがって、SQLNumResultCols()、SQLDescribeCol()、または SQLColAttribute() を使ってカーソルを記述できるということです。

SQLNextResult() の呼び出しが完了すると、それまで StatementHandle2 に関連付けられていた結果セットは、残りの結果セットのチェーンから除去されるので、SQLNextResult() または SQLMoreResults() で再使用することはできません。すなわち、n 個の結果セットがある場合、最大 n-1 回だけ SQLNextResult() を正常に呼び出せるということです。

SQLCloseCursor() が呼び出された場合や、SQL_CLOSE オプションを指定して SQLFreeStmt() が呼び出された場合、あるいは HandleType を SQL_HANDLE_STMT に設定して SQLFreeHandle() が呼び出された場合、このステートメント・ハンドル上のすべてのペンディング結果セットは廃棄されます。

StatementHandle2 にオープン・カーソルがある場合や、StatementHandle1 と StatementHandle2 が同じ接続上でない場合、SQLNextResult() は SQL_ERROR を戻します。エラーまたは警告が戻された場合は常に、StatementHandle1 で SQLGetDiagRec() を呼び出さなければなりません。

注: SQLMoreResults() は、SQL_ATTR_ROW_ARRAY_SIZE ステートメント属性および SQLBindParameter() を使って指定された入力パラメーター値の配列が指定されているパラメーター化照会とも連動して稼働します。ただし SQLNextResult() はこれをサポートしていません。

戻りコード:

- SQL_SUCCESS
- SQL_SUCCESS_WITH_INFO
- SQL_STILL_EXECUTING
- SQL_ERROR
- SQL_INVALID_HANDLE
- SQL_NO_DATA_FOUND

診断:

表 219. SQLNextResult SQLSTATE

SQLSTATE	説明	解説
40003 08S01	通信リンクに障害が起きた。	アプリケーションとデータ・ソースとの間の通信リンクが、関数の完了する前に失敗しました。

表 219. SQLNextResult SQLSTATE (続き)

SQLSTATE	説明	解説
58004	想定外のシステム障害です。	回復不能システム・エラー。
HY001	メモリーの割り振りが失敗しました。	DB2 CLI は、この関数の実行または完了をサポートするのに必要なメモリーを割り当てられません。
HY010	関数のシーケンス・エラーです。	実行時データ (SQLParamData(), SQLPutData()) 操作中に、関数が呼び出されました。 <i>StatementHandle2</i> には、関連したオープン・カーソルがあります。 BEGIN COMPOUND と END COMPOUND の SQL 操作中に、関数が呼び出されました。
HY013	想定外のメモリー処理エラーです。	DB2 CLI は、この関数の実行または完了をサポートするのに必要なメモリーを割り当てられません。
HYT00	タイムアウトの有効期限切れです。	データ・ソースが結果セットを戻す前に、タイムアウト期間が満了しました。タイムアウト期間は、SQLSetStmtAttr() の SQL_ATTR_QUERY_TIMEOUT 属性を使用して設定することができます。

制約事項:

パラメーター化照会およびバッチ SQL には 1 つの SQLMoreResults() だけを使用することができます。

例:

```
/* use SQLNextResult to push Result Set 2 onto the second statement handle */
cliRC = SQLNextResult(hstmt, hstmt2); /* open second cursor */
```

関連概念:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻の『DB2 CLI 用の SQLSTATES』

関連タスク:

- 839 ページの『CLI アプリケーションからのストアード・プロシージャの呼び出し』

関連資料:

- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻の『SQLMoreResults 関数 (CLI) - さらに結果セットがあるかどうかの判別』
- コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻の『ステートメント属性 (CLI) のリスト』

SQLNextResult

- コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻 の『CLI 関数戻りコード』

関連サンプル:

- 『spcli.c -- Contrast stored procedure multiple result set handling methods (CLI)』

管理 API 解説書

sqlcrea - データベースの作成

オプションでユーザー定義の照合シーケンスを持つ新規データベースを初期設定し、3つの初期表スペースやシステム表を作成し、さらにはリカバリー・ログを割り当てます。

有効範囲:

パーティション・データベース環境では、この API は `db2nodes.cfg` ファイルにリストされているすべてのデータベース・パーティション・サーバーに影響を与えます。

この API が呼び出されるデータベース・パーティション・サーバーは、新規のデータベース用のカタログ・パーティションになります。

許可:

以下のいずれかです。

- `sysadm`
- `sysctrl`

必要な接続:

インスタンス。データベースを別の (リモート) ノードに作成するには、まずそのノードに接続する必要があります。データベース接続は、この API によって処理中に一時的に確立されます。

API 組み込みファイル:

`sqlenv.h`

C API 構文:

```
/* File: sqlenv.h */
/* API: sqlcrea */
/* ... */
SQL_API_RC SQL_API_FN
sqlcrea (
    char *pDbName,
    char *pLocalDbAlias,
    char *pPath,
    struct sqlcldbdesc *pDbDescriptor,
```

```

    struct sqldbterritoryinfo *pTerritoryInfo,
    char Reserved2,
    void *pReserved1,
    struct sqlca *pSqlca);
/* ... */

```

汎用 API 構文:

```

/* File: sqlenv.h */
/* API: sqlgcrea */
/* ... */
SQL_API_RC SQL_API_FN
sqlgcrea (
    unsigned short PathLen,
    unsigned short LocalDbAliasLen,
    unsigned short DbNameLen,
    struct sqlca *pSqlca,
    void *pReserved1,
    unsigned short Reserved2,
    struct sqldbterritoryinfo *pTerritoryInfo,
    struct sqldbdesc *pDbDescriptor,
    char *pPath,
    char *pLocalDbAlias,
    char *pDbName);
/* ... */

```

API パラメーター:

PathLen

入力。パスの長さを示す 2 バイトの符号なし整数 (バイト単位) です。パスが提供されていない場合は、ゼロに設定してください。

LocalDbAliasLen

入力。ローカル・データベース別名の長さを示す 2 バイトの符号なし整数 (バイト単位) です。ローカル別名が提供されていない場合は、ゼロに設定してください。

DbNameLen

入力。データベース名の長さを示す 2 バイトの符号なし整数 (バイト単位) です。

pSqlca 出力。sqlca 構造へのポインター。

pReserved1

入力。NULL に設定されたスペア・ポインター、またはゼロを指すスペア・ポインター。

Reserved2

入力。将来の使用のために予約されています。

pTerritoryInfo

入力。sqldbterritoryinfo 構造を指すポインター。データベースのロケールおよびコード・セットが含まれます。NULL にすることもできます。

pDbDescriptor

入力。データベースの作成時に使用されるデータベース記述ブロックを指すポインター。データベース記述ブロックは、照合シーケンスのように、永久にデータベースの構成ファイルに保管される値を提供するために使用することができます。NULL にすることもできます。

pPath 入力。UNIX ベースのシステムの場合は、データベースを作成するパスを指定します。パスが指定されない場合、データベースは、データベース・マネージャー構成ファイルに指定されているデフォルトのデータベース・パス (*dftdbpath* パラメーター) に作成されます。Windows オペレーティング・システムの場合は、データベースの作成先のドライブ名を指定します。NULL にすることもできます。

注: パーティション・データベース環境の場合は、NFS がマウントされたディレクトリーにはデータベースを作成しないでください。パスが指定されていない場合、NFS がマウントされたパスに *dftdbpath* データベース・マネージャー構成パラメーターが指定されないようにしてください。(たとえば、UNIX ベースのシステムの場合には、インスタンスの所有者の \$HOME ディレクトリーを指定しないでください。) パーティション・データベースでこの API に指定されたパスを相対パスにすることはできません。

pLocalDbAlias

入力。クライアントのシステム・データベース・ディレクトリーに置かれる別名を含むストリングを指定します。NULL にすることもできます。ローカル別名が指定されない場合、データベース名はデフォルトのものになります。

pDbName

入力。データベース名を含むストリングを指定します。これはシステム・データベース・ディレクトリーにカタログされるデータベース名です。データベースは、サーバーのシステム・データベース・ディレクトリーに正常に作成されると、データベース名と同じデータベース別名を持つシステム・データベース・ディレクトリーに自動的にカタログされます。NULL にはしないでください。

REXX API 構文:

```
CREATE DATABASE dbname [ON path] [ALIAS dbalias]
[USING CODESET codeset TERRITORY territory]
[COLLATE USING {SYSTEM | IDENTITY | USER :udcs}]
[NUMSEGS numsegs] [DFT_EXTENT_SZ dft_extentsize]
[CATALOG TABLESPACE <tablespace_definition>]
[USER TABLESPACE <tablespace_definition>]
[TEMPORARY TABLESPACE <tablespace_definition>]
[WITH comment]
```

<tablespace_definition> は以下を表します。
MANAGED BY {

```

SYSTEM USING :SMS_string |
DATABASE USING :DMS_string }
[ EXTENTSIZE number_of_pages ]
[ PREFETCHSIZE number_of_pages ]
[ OVERHEAD number_of_milliseconds ]
[ TRANSFERRATE number_of_milliseconds ]

```

REXX API パラメーター:**dbname**

データベースの名前。

dbalias

データベースの別名。

path

データベースを作成するパス。

パスが指定されない場合、データベースは、データベース・マネージャー構成ファイルに指定されているデフォルトのデータベース・パス (*dfidbpath* 構成パラメーター) に作成されます。

注: パーティション・データベース環境の場合は、NFS がマウントされたディレクトリにはデータベースを作成しないでください。パスが指定されていない場合、NFS がマウントされたパスに *dfidbpath* データベース・マネージャー構成パラメーターが指定されないようにしてください。(たとえば、UNIX ベースのシステムの場合には、インスタンスの所有者の \$HOME ディレクトリを指定しないでください。) パーティション・データベースでこの API に指定されたパスを相対パスにすることはできません。

codeset

データベースに入力されたデータに使用するコード・セットを表します。

territory

データベースに入力されたデータに使用する地域コード (ロケール) を表します。

SYSTEM

現行の国地域コードに基づくオペレーティング・システムの照合シーケンスを使用します。

IDENTITY

照合シーケンスは ID シーケンスで、ストリングはバイトごとに比較されます。左端のバイトから比較されていきます。

USER udcs

照合シーケンスは、アプリケーションを呼び出すことにより、照合シーケンスを定義する 256 バイトのストリングを含むホスト変数に指定されます。

numsegs

DAT、IDX、および LF ファイルを保管するために作成および使用されるセグメント・ディレクトリーの数を表します。

dft_extentsize

データベースの表スペースのデフォルトの *extent size* を指定します。

SMS_string

表スペースに属する予定の 1 つまたは複数のコンテナを識別する複合 REXX ホスト変数を表します。ここに表スペース・データが格納されます。以下の項目において、XXX はホスト変数名を表しています。おのおののディレクトリー名は 254 バイトを超えることはできないことに注意してください。

XXX.0 指定されたディレクトリーの数。

XXX.1 SMS 表スペースの最初のディレクトリー名。

XXX.2 SMS 表スペースの 2 番目のディレクトリー名。

XXX.3 以降、3 番目、4 番目 ... と続きます。

DMS_string

表スペースに属する予定の 1 つまたは複数のコンテナを識別する複合 REXX ホスト変数を表します。ここに表スペース・データが格納されます。コンテナ・サイズ (4KB ページの数で指定) およびタイプ (ファイルまたは装置) が格納されます。指定された装置 (ファイルではない) は、前もって存在していなければなりません。以下の項目において、XXX はホスト変数名を表しています。おのおののコンテナ名は 254 バイトを超えることはできないことに注意してください。

XXX.0 REXX ホスト変数内のストリングの数 (最初のレベル・エレメントの数)。

XXX.1.1 最初のコンテナのタイプ (file または device)。

XXX.1.2 最初のファイル名または装置名。

XXX.1.3 最初のコンテナのサイズ (ページ単位)。

XXX.2.1 2 番目のコンテナのタイプ (file または device)。

XXX.2.2 2 番目のファイル名または装置名。

XXX.2.3 2 番目のコンテナのサイズ (ページ単位)。

XXX.3.1

以降、3 番目、4 番目 ... と続きます。

EXTENTSIZ *number_of_pages*

次のコンテナースキップする前に、コンテナース書き込まれることになる 4KB ページの数。

PREFETCHSIZE *number_of_pages*

データのプリフェッチを実行する際に、表スペースから読み取られることになる 4KB ページの数。

OVERHEAD *number_of_milliseconds*

入出力制御装置のオーバーヘッド、ディスク・シーク、および待ち時間を指定する数 (ミリ秒単位)。

TRANSFERRATE *number_of_milliseconds*

1 つの 4KB ページをメモリーに読み取る時間を指定する数 (ミリ秒単位)。

comment

システム・ディレクトリー内のデータベースまたはデータベース項目のコメント。コメント中に復帰文字または改行文字を使用しないでください。コメント文は必ず二重引用符で囲んでください。コメントは最大 30 文字まで記述できます。

使用上の注意:

CREATE DATABASE は以下の事柄を行います。

- 指定されたサブディレクトリーにデータベースを作成します。パーティション・データベース環境では、`db2nodes.cfg` にリストされたすべてのデータベース・パーティション・サーバー上にデータベースを作成し、各データベース・パーティション・サーバーで指定されたサブディレクトリーの下に `$DB2INSTANCE/NODExxxx` ディレクトリーを作成します。 `xxxx` はローカル・データベース・パーティション・サーバー番号を表します。単一パーティション環境では、指定されたサブディレクトリーの下に `$DB2INSTANCE/NODE0000` ディレクトリーを作成します。
- システム・カタログ表およびリカバリー・ログを作成します。
- 以下のディレクトリーのデータベースをカタログします。
 - `pPath` によって示されたパスにあるサーバーのローカル・データベース・ディレクトリー。または、パスが指定されていない場合は、データベース・マネージャーのシステム構成ファイルで定義されたデフォルトのデータベース・パスが使用されます。ローカル・データベース・ディレクトリーは、データベースを含む各ファイル・システムに存在します。
 - アタッチされたインスタンス用のサーバーのシステム・データベース・ディレクトリー。作成されるディレクトリー項目にはデータベース名やデータベース別名が含まれます。

sqlcrea - データベースの作成

API がリモート・クライアントから呼び出された場合は、クライアントのシステム・データベース・ディレクトリーもデータベース名やデータベース別名で更新されます。

システム・データベース・ディレクトリーとローカル・データベース・ディレクトリーがどちらも存在しない場合、そのどちらかを作成します。指定されていれば、コメントおよびコード設定値は両方のディレクトリーに入れられます。

- 指定されたコード設定、区域、および照合シーケンスを保管します。照合シーケンスがユニークの重みから成っているか、または ID 順序である場合、フラグがデータベース構成ファイルで設定されます。
- SYSCAT、SYSFUN、SYSIBM、および SYSIBM を所有者とする SYSSTAT という各スキーマを作成します。この API が呼び出されるデータベース・パーティション・サーバーは、新規のデータベース用のカタログ・パーティションになります。2 つのデータベース・パーティション・グループ (IBMDEFAULTGROUP および IBMCATGROUP) が自動的に作成されます。
- 事前に定義されたデータベース・マネージャー・バインド・ファイルのデータベースにバインドします (これらは db2ubind.lst にリストされています)。これらの 1 つまたは複数のファイルのバインドが正常に行わなかった場合、sqlcrea は SQLCA に警告を戻し、失敗したバインドに関する情報を提供します。バインドが失敗した場合、ユーザーは訂正のアクションをとり、手動で失敗したファイルをバインドできません。データベースはどのような場合にでも作成されます。PUBLIC を認可した CREATEIN 特権を伴った、バインドを行った際 NULLID と呼ばれているスキーマが、暗黙的に作成されます。
- SYSCATSPACE、TEMPSPACE1、および USERSPACE1 表スペースを作成します。SYSCATSPACE 表スペースはカタログ・パーティションにのみ作成されます。すべてのデータベース・パーティションには同じ表スペース定義があります。
- 以下の権限を付与します。
 - データベース作成者に対する DBADM 権限と、CONNECT、CREATETAB、BINDADD、CREATE_NOT_FENCED、IMPLICIT_SCHEMA、および LOAD の各種特権。
 - PUBLIC に対する CONNECT、CREATETAB、BINDADD、IMPLICIT_SCHEMA の各種特権。
 - PUBLIC に対する USERSPACE1 表スペース上の USE 特権。
 - PUBLIC に対する各システム・カタログ上の SELECT 特権。
 - 正常実行したバインド・ユーティリティごとに、PUBLIC に対する BIND および EXECUTE 特権。
 - SYSFUN スキーマのすべての関数について、PUBLIC に対する EXECUTE WITH GRANT 特権。
 - SYSIBM スキーマのすべてのプロシージャについて、PUBLIC に対する EXECUTE 特権。

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

パーティション・データベース環境では、データベース・マネージャーはすべてのデータベース・パーティション・サーバーの指定されたパスまたはデフォルト・パスの下に、`$DB2INSTANCE/NODExxxx` を作成します。 *xxxx* は `db2nodes.cfg` ファイルで定義されたノード番号です (つまり、ノード 0 は `NODE0000` になります)。 `SQL00001` から `SQLnnnnn` のサブディレクトリーはこのパスにあります。これにより、別々のデータベース・パーティション・サーバーと関連付けられたデータベース・オブジェクトが、それぞれ別々のディレクトリーに保管されるようになります (指定されたパスまたはデフォルト・パスの下にあるサブディレクトリー `$DB2INSTANCE` をすべてのデータベース・パーティション・サーバーが共有している場合でも、そのようになります)。

Windows および AIX では、コード・セット名の長さは最大 9 文字に制限されています。たとえば、`IS08859-15` というコード・セット名の代わりに、`IS0885915` のようにします。

アプリケーションがすでにデータベースに接続されている場合、`CREATE DATABASE` は失敗します。

データベース記述ブロック構造が正しく設定されていない場合、エラー・メッセージが戻されます。

データベース記述ブロックの「目印」は、シンボリック値 `SQLE_DBDESC_2` (`sqlenv` で定義) に設定する必要があります。以下のユーザー定義の照合シーケンスの例は、ホスト言語組み込みファイルで使用できます。

sqlc819a データベースのコード・ページが 819 (ISO Latin/1) である場合、このシーケンスにより、ホスト CCSID 500 (EBCDIC 国際標準) にしたがつてソートが行われます。

sqlc819b データベースのコード・ページが 819 (ISO Latin/1) である場合、このシーケンスにより、ホスト CCSID 037 (EBCDIC 米国英語) にしたがつてソートが行われます。

sqlc850a データベースのコード・ページが 850 (ASCII Latin/1) である場合、このシーケンスにより、ホスト CCSID 500 (EBCDIC 国際標準) にしたがつてソートが行われます。

sqlc850b データベースのコード・ページが 850 (ASCII Latin/1) である場合、このシーケンスにより、ホスト CCSID 037 (EBCDIC 米国英語) にしたがつてソートが行われます。

sqlcrea - データベースの作成

- sql932a** データベースのコード・ページが 932 (ASCII 日本語) である場合、このシーケンスにより、ホスト CCSID 5035 (EBCDIC 日本語) にしたがってソートが行われます。
- sql932b** データベースのコード・ページが 932 (ASCII 日本語) である場合、このシーケンスにより、ホスト CCSID 5026 (EBCDIC 日本語) にしたがってソートが行われます。

CREATE DATABASE 中に指定された照合シーケンスは後で変更することはできません。データベースのすべての文字比較は指定された照合シーケンスを使用します。これは索引の構造と照会の結果に影響します。

新規のデータベースに異なる別名を定義するには、**sqlcadb** を使用してください。

関連資料:

- 管理 API リファレンス の『sqlabndx - バインド』
- 管理 API リファレンス の『sqlcadb - データベースのカタログ』
- 管理 API リファレンス の『sqledrpd - データベースのドロップ』
- 管理 API リファレンス の『sqlcran - ノードでのデータベースの作成』
- 管理 API リファレンス の『sqledpan - ノードでのデータベースのドロップ』
- 管理 API リファレンス の『SQLEDBTERRITORYINFO』
- 管理 API リファレンス の『SQLCA』
- 管理 API リファレンス の『SQLEDBDESC』

関連サンプル:

- 『db_udcs.cbl -- How to use user-defined collating sequence (IBM COBOL)』
- 『dbconf.cbl -- Update database configuration (IBM COBOL)』
- 『ebcdicdb.cbl -- Create a database with EBCDIC 037 standard collating sequence (IBM COBOL)』
- 『dbcreate.c -- Create and drop databases (C)』
- 『dbrecov.sqc -- How to recover a database (C)』
- 『dbsample.sqc -- Creates a sample database (C)』
- 『dbcreate.C -- Create and drop databases (C++)』
- 『dbrecov.sqc -- How to recover a database (C++)』

db2AutoConfig - 自動構成

アプリケーション・プログラムが、コントロール・センターで構成アドバイザーにアクセスできるようにします。このアドバイザーに関する詳細は、コントロール・センター内のオンライン・ヘルプ機能によって提供されます。

許可:

sysadm

必要な接続:

データベース

API 組み込みファイル:

db2AuCfg.h

C API 構文:

```

/* File: db2AuCfg.h */
/* API: db2AutoConfig */
/* ... */
SQL_API_RC SQL_API_FN
db2AutoConfig(
    db2UInt32 db2VersionNumber,
    void *pAutoConfigInterface,
    struct sqlca *pSqlca);

typedef struct {
    db2int32 iProductID;
    char iProductVersion[DB2_SG_PROD_VERSION_SIZE];
    char iDbAlias[SQL_ALIAS_SZ];
    db2int32 iApply;
    db2AutoConfigInput iParams;
    db2AutoConfigOutput oResult;
} db2AutoConfigInterface;

typedef struct {
    db2int32 token;
    db2int32 value;
} db2AutoConfigElement;

typedef struct {
    db2UInt32 numElements;
    db2AutoConfigElement *pElements;
} db2AutoConfigArray;

typedef db2AutoConfigArray db2AutoConfigInput;
typedef db2AutoConfigArray db2AutoConfigDiags;

typedef struct {
    db2UInt32 numElements;
    struct sqlfupd *pConfigs;
    void *pDataArea;
} db2ConfigValues;

typedef struct {
    char *pName;
    db2int32 value;

```

db2AutoConfig - 自動構成

```
} db2AutoConfigNameElement;

typedef struct {
    db2UInt32    numElements;
    db2AutoConfigElement *pElements;
} db2AutoConfigNameArray;

typedef db2AutoConfigNameArray db2BpValues;

typedef struct {
    db2ConfigValues    o01dDbValues;
    db2ConfigValues    o01dDbmValues;
    db2ConfigValues    oNewDbValues;
    db2ConfigValues    oNewDbmValues;
    db2AutoConfigDiags oDiagnostics;
    db2BpValues         o01dBpValues;
    db2BpValues         oNewBpValues;
} db2AutoConfigOutput;
/* ... */
```

API パラメーター:

db2VersionNumber

入力。 2 番目のパラメーター *pAutoConfigInterface* として渡される構造のバージョンとリリースのレベルを指定します。

pAutoConfigInterface

入力。 *db2AutoConfigInterface* 構造を指すポインター。

pSqlca 出力。 *sqlca* 構造へのポインター。

iProductID

入力。ユニークの製品 ID を指定します。有効な製品 ID の値については、API 組み込みファイル *db2AuCfg.h* を参照してください。

iProductVersion

入力。製品のバージョンを指定する 16 バイトの文字列です。

iDbAlias

入力。データベース別名を指定する文字列です。

iApply

入力。構成を自動的に更新します。有効な値については、API 組み込みファイル *db2AuCfg.h* を参照してください。

iParams

入力。 アドバイザーにパラメーターを渡します。

oResult

出力。 アドバイザーからの結果がすべて含まれます。

token 入力パラメーターと出力診断の両方に関する構成値を指定します。

value トークンによって指定されたデータを保持します。

numElements

配列エレメントの数を示します。

pElements

エレメント配列を指すポインター。

db2AutoConfigDiags

診断と問題判別を行うためのトークンと値を返します。トークンは問題を識別し、値は適切な勧告 (ある場合) を示します。トークンと値のリストについては、API 組み込みファイル `db2AuCfg.h` を参照してください。

pConfigs

SQLFUPD 構造を指すポインター。

pDataArea

構成の値が含まれるデータ域を指すポインター。

pName 出力。出力バッファ・プールの名前。

value 名前で指定されたバッファ・プールのサイズ (ページ単位) を保持します。

oOldDbValues

出力。 *iApply* の値がデータベース構成またはすべての構成を更新するように設定されている場合、この値は、アドバイザー が使用される前のデータベースの構成値を表します。そうでなければ、これは現行値です。

oOldDbmValues

出力。 *iApply* の値がすべての構成を更新するように設定されている場合、この値は、アドバイザーが使用される前のデータベース・マネージャーの構成値を表します。そうでなければ、これは現行値です。

oNewDbValues

出力。 *iApply* の値がデータベース構成またはすべての構成を更新するように設定されている場合、この値は現行データベースの構成値を表します。そうでなければ、これはアドバイザーに対する推奨値です。

oNewDbmValues

出力。 *iApply* の値がすべての構成を更新するように設定されている場合、この値は現行のデータベース・マネージャーの構成値を表します。そうでなければ、これはアドバイザーに対する推奨値です。

oDiagnostics

出力。アドバイザーからの診断が含まれます。

oOldBpValues

出力。 *iApply* の値がデータベース構成またはすべての構成を更新するように設定されている場合、この値は、アドバイザーが使用される前のバッファ・プールのサイズ (ページ単位) を表します。そうでなければ、これは現行値です。

db2AutoConfig - 自動構成

oNewBpValues

出力。 *iApply* の値がデータベース構成またはすべての構成を更新するよう設定されている場合、この値は現行のバッファ・プールのサイズ (ページ単位) を表します。そうでなければ、これはアドバイザーに対する推奨値です。

使用上の注意:

db2AutoConfig によって割り振られたメモリーを解放するには、db2AutoConfigFreeMemory を呼び出します。

関連資料:

- 管理 API リファレンス の『SQLCA』
- 管理 API リファレンス の『SQLFUPD』
- 管理 API リファレンス の『db2AutoConfigFreeMemory - 自動構成メモリーの解放』
- 管理 API リファレンス の『db2CfgSet - 構成パラメーターの設定』

関連サンプル:

- 『dbcfg.sqc -- Configure database and database manager configuration parameters (C)』
- 『dbcfg.sqC -- Configure database and database manager configuration parameters (C++)』

db2Inspect - データベースの検査

データベースの構造上の保全性を検査し、ページの整合性についてデータベースのページをチェックします。

有効範囲:

単一パーティション・データベースでは、有効範囲は単一パーティションのみです。パーティション・データベース環境では、これは db2nodes.cfg に定義されたすべての論理パーティションの集合です。

許可:

以下のいずれかです。

- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*
- 表に対する CONTROL 特権

必要な接続:

データベース

API 組み込みファイル:*db2ApiDf.h***C API 構文:**

```

/* File: db2ApiDf.h */
/* API: db2Inspect */
/* ... */
SQL_API_RC SQL_API_FN
db2Inspect (
    db2UInt32 versionNumber,
    void *pParmStruct,
    struct sqlca *pSqlca);

typedef SQL_STRUCTURE db2InspectStruct
{
    char                *piTablespaceName;
    char                *piTableName;
    char                *piSchemaName;
    char                *piResultsName;
    char                *piDataFileName;
    SQL_PDB_NODE_TYPE *piNodeList;
    db2UInt32           iAction;
    db2int32            iTablespaceID;
    db2int32            iObjectID;
    db2UInt32           iBeginCheckOption;
    db2int32            iLimitErrorReported;
    db2UInt16           iObjectErrorState;
    db2UInt16           iKeepResultfile;
    db2UInt16           iAllNodeFlag;
    db2UInt16           iNumNodes;
    db2UInt16           iLevelObjectData;
    db2UInt16           iLevelObjectIndex;
    db2UInt16           iLevelObjectLong;
    db2UInt16           iLevelObjectLOB;
    db2UInt16           iLevelObjectBlkMap;
    db2UInt16           iLevelExtentMap;
} db2InspectStruct;
/* ... */

```

汎用 API 構文:

```

/* File: db2ApiDf.h */
/* API: db2gInspect */
/* ... */
SQL_API_RC SQL_API_FN
db2gInspect (
    db2UInt32 versionNumber,
    void *pParmStruct,
    struct sqlca *pSqlca);

typedef SQL_STRUCTURE db2gInspectStruct
{
    char                *piTablespaceName;
    char                *piTableName;

```

db2Inspect - データベースの検査

```
char                *piSchemaName;
char                *piResultsName;
char                *piDataFileName;
SQL_PDB_NODE_TYPE  *piNodeList;
db2UInt32          iResultsNameLength;
db2UInt32          iDataFileNameLength;
db2UInt32          iTablespaceNameLength;
db2UInt32          iTableNameLength;
db2UInt32          iSchemaNameLength;
db2UInt32          iAction;
db2int32           iTablespaceID;
db2int32           iObjectID;
db2UInt32          iBeginCheckOption;
db2int32           iLimitErrorReported;
db2UInt16          iObjectErrorState;
db2UInt16          iKeepResultfile;
db2UInt16          iAllNodeFlag;
db2UInt16          iNumNodes;
db2UInt16          iLevelObjectData;
db2UInt16          iLevelObjectIndex;
db2UInt16          iLevelObjectLong;
db2UInt16          iLevelObjectLOB;
db2UInt16          iLevelObjectBlkMap;
db2UInt16          iLevelExtentMap;
} db2gInspectStruct;
/* ... */
```

API パラメーター:

versionNumber

入力。 2 番目のパラメーター *pParmStruct* として渡される構造のバージョンとリリースのレベルを指定します。

pParmStruct

入力。 *db2InspectStruct* 構造を指すポインター。

pSqlca 出力。 *sqlca* 構造を指すポインター。

piTablespaceName

入力。表スペース名を含むストリング。表スペースは、表スペースでの操作で識別される必要があります。ポインターが NULL の場合、表スペース ID の値が入力として使用されます。

piTableName

入力。表名を含むストリング。表は、表または表オブジェクトでの操作で識別される必要があります。ポインターが NULL の場合、表スペース ID および表オブジェクト ID の値が入力として使用されます。

piSchemaName

入力。スキーマ名を含むストリング。

piResultsName

入力。結果出力ファイルの名前を含むストリング。この入力は提供する必要があります。ファイルは診断データ・ディレクトリー・パスに書き込まれます。

piDataFileName

入力。将来の使用のために予約されています。 NULL に設定する必要があります。

piNodeList

入力。操作を実行するパーティション番号の配列を指すポインター。

iResultsNameLength

入力。結果ファイル名のストリングの長さ。

iDataFileNameLength

入力。データ出力ファイル名のストリングの長さ。

iTablesapceNameLength

入力。表スペース名のストリングの長さ。

iTableNameLength

入力。表名のストリングの長さ。

iSchemaNameLength

入力。スキーマ名のストリングの長さ。

iAction 入力。検査アクションを指定します。有効な値は以下のとおりです。

DB2INSPECT_ACT_CHECK_DB

データベース全体を検査します。

DB2INSPECT_ACT_CHECK_TABSPACE

表スペースを検査します。

DB2INSPECT_ACT_CHECK_TABLE

表を検査します。

iTablespaceID

入力。表スペース ID を指定します。表スペースを識別しなければならない場合、表スペース名へのポインターが NULL であれば、表スペース ID の値が入力として使用されます。

iObjectID

入力。オブジェクト ID を指定します。表を識別しなければならない場合、表名へのポインターが NULL であれば、オブジェクト ID の値が入力として使用されます。

iBeginCheckOption

入力。データベースのチェック、または操作の開始地点を示す表スペース操作をチェックするオプション。通常の開始地点から開始するには、ゼロに設定する必要があります。値は以下のとおりです。

DB2INSPECT_BEGIN_TSPID

この値をデータベースのチェックに使用し、表スペース ID フィールドによって指定された表スペースから開始します。表スペース ID が設定されている必要があります。

DB2INSPECT_BEGIN_TSPID_OBJID

この値をデータベースのチェックに使用し、表スペース ID およびオブジェクト ID フィールドによって指定された表から開始します。このオプションを使用するには、表スペース ID およびオブジェクト ID が設定されている必要があります。

DB2INSPECT_BEGIN_OBJID

この値を表スペースのチェックに使用し、オブジェクト ID フィールドによって指定された表から開始します。オブジェクト ID が設定されている必要があります。

iLimitErrorReported

入力。オブジェクトのエラーとなったページ数の報告限度を指定します。限度値として使用する数、または以下のいずれかの値を指定します。

DB2INSPECT_LIMIT_ERROR_DEFAULT

この値を使用して、報告されるエラーとなったページの最大数が、オブジェクトのエクステント・サイズであることを指定します。

DB2INSPECT_LIMIT_ERROR_ALL

この値を使用して、エラーとなったすべてのページを報告します。

iObjectErrorState

入力。エラー状態のオブジェクトをスキャンするかどうかを指定します。有効な値は以下のとおりです。

DB2INSPECT_ERROR_STATE_NORMAL

正常な状態にあるオブジェクトのみ処理します。

DB2INSPECT_ERROR_STATE_ALL

エラー状態のオブジェクトを含め、すべてのオブジェクトを処理します。

iKeepResultfile

入力。結果ファイルの保存を指定します。有効な値は以下のとおりです。

DB2INSPECT_RESFILE_CLEANUP

エラーが報告された場合、結果出力ファイルが保存されます。そうでない場合、結果ファイルは操作終了時に除去されます。

DB2INSPECT_RESFILE_KEEP_ALWAYS

結果出力ファイルが保存されます。

iAllNodeFlag

入力。操作が、db2nodes.cfg に定義されているすべてのノードに適用されるかどうかを示します。有効な値は以下のとおりです。

DB2_NODE_LIST

pNodeList で渡されたノード・リスト内のすべてのノードに適用されます。

DB2_ALL_NODES

すべてのノードに適用されます。 *pNodeList* は NULL でなければなりません。これがデフォルト値です。

DB2_ALL_EXCEPT

pNodeList で渡されたノード・リスト内のノードを除く、すべてのノードに適用されます。

iNumNodes

入力。 *pNodeList* 配列内のノードの数を指定します。

iLevelObjectData

入力。データ・オブジェクトの処理レベルを指定します。有効な値は以下のとおりです。

DB2INSPECT_LEVEL_NORMAL

通常レベルです。

DB2INSPECT_LEVEL_LOW

低いレベルです。

DB2INSPECT_LEVEL_NONE

レベルはありません。

iLevelObjectIndex

入力。索引オブジェクトの処理レベルを指定します。有効な値は以下のとおりです。

DB2INSPECT_LEVEL_NORMAL

通常レベルです。

DB2INSPECT_LEVEL_LOW

低いレベルです。

DB2INSPECT_LEVEL_NONE

レベルはありません。

iLevelObjectLong

入力。ロング・オブジェクトの処理レベルを指定します。有効な値は以下のとおりです。

DB2INSPECT_LEVEL_NORMAL

通常レベルです。

DB2INSPECT_LEVEL_LOW

低いレベルです。

DB2INSPECT_LEVEL_NONE

レベルはありません。

iLevelObjectLOB

入力。LOB オブジェクトの処理レベルを指定します。有効な値は以下のとおりです。

DB2INSPECT_LEVEL_NORMAL

通常レベルです。

DB2INSPECT_LEVEL_LOW

低いレベルです。

DB2INSPECT_LEVEL_NONE

レベルはありません。

iLevelObjectBlkMap

入力。ブロック・マップ・オブジェクトの処理レベルを指定します。有効な値は以下のとおりです。

DB2INSPECT_LEVEL_NORMAL

通常レベルです。

DB2INSPECT_LEVEL_LOW

低いレベルです。

DB2INSPECT_LEVEL_NONE

レベルはありません。

iLevelExtentMap

入力。エクステント・マップの処理レベルを指定します。有効な値は以下のとおりです。

DB2INSPECT_LEVEL_NORMAL

通常レベルです。

DB2INSPECT_LEVEL_LOW

低いレベルです。

DB2INSPECT_LEVEL_NONE

レベルはありません。

使用上の注意:

オンライン検査処理では、分離レベルを非コミット読み取りに指定してデータベース・オブジェクトにアクセスします。コミット処理は、検査処理時に行われます。検査操作を開始する前に、COMMIT または ROLLBACK を発行して作業単位を終了することをお勧めします。

検査チェック処理では、不定形式の検査データ結果を結果ファイルに書き込みます。ファイルは診断データ・ディレクトリー・パスに書き込まれます。チェック処理でエラーが検出されない場合、結果出力ファイルは検査操作の終了時に消去されます。チェック処理でエラーが検出された場合、結果出力ファイルは検査操作の終了時に消去されません。検査の詳細について調べるには、検査結果出力ファイルを db2inspf ユーティリティーでフォーマットしてください。

パーティション・データベース環境では、結果出力ファイルの拡張部分はデータベース・パーティション番号に対応します。ファイルは、データベース・マネージャーの診断データ・ディレクトリー・パスに置かれます。

ユニークの結果出力ファイル名を指定する必要があります。結果出力ファイルがすでに存在する場合は、操作は処理されません。

表スペースの処理では、その表スペースにあるオブジェクトだけを処理します。

関連資料:

- 管理 API リファレンス の『SQLCA』

db2Load - ロード

データを DB2 表にロードします。サーバー上にあるデータは、ファイル、カーソル、テープ、または Named PIPE の形式です。リモートで接続しているクライアント上にあるデータは、完全修飾ファイル、カーソル、または Named PIPE の形式とすることができます。ロード・ユーティリティーは、階層レベルでのデータのロードをサポートしていません。

許可:

以下のいずれかです。

- *sysadm*
- *dbadm*
- データベースに対するロード権限と以下のもの
 - ロード・ユーティリティーが INSERT モード、 TERMINATE モード (それまでのロード挿入操作を終了する)、または RESTART モード (以前のロード挿入操作を再開する) で呼び出された場合には、その表に対する INSERT 特権。
 - ロード・ユーティリティーが REPLACE モード、 TERMINATE モード (それまでのロード置換操作を終了する)、または RESTART モード (以前のロード置換操作を再開する) で呼び出された場合には、その表に対する INSERT および DELETE 特権。
 - 例外表がロード操作の一部として使用される場合、その例外表に対する INSERT 特権。

db2Load - ロード

注: 一般的に、すべてのロード処理、およびすべての DB2 サーバー処理は、インスタンス所有者に所有されています。これらのすべての処理では、インスタンス所有者の ID を使用して、必要なファイルにアクセスします。そのため、インスタンス所有者は、誰がコマンドを呼び出すかに関係なく、入力ファイルへの読み取りアクセスを持っている必要があります。

必要な接続:

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

インスタンス。明示的なアタッチは必要ありません。データベースへの接続が確立している場合には、ローカル・インスタンスへの暗黙的なアタッチが試行されます。

API 組み込みファイル:

db2ApiDf.h

C API 構文:

```
/* File: db2ApiDf.h */
/* API: Load */
/* ... */
SQL_API_RC SQL_API_FN
db2Load (
    db2Uint32 versionNumber,
    void * pParmStruct,
    struct sqlca * pSqlca);

typedef SQL_STRUCTURE db2LoadStruct
{
    struct sqlu_media_list *piSourceList;
    struct sqlu_media_list *piLobPathList;
    struct sqldcol *piDataDescriptor;
    struct sqlchar *piActionString;
    char *piFileType;
    struct sqlchar *piFileTypeMod;
    char *piLocalMsgFileName;
    char *piTempFilesPath;
    struct sqlu_media_list *piVendorSortWorkPaths;
    struct sqlu_media_list *piCopyTargetList;
    db2int32 *piNullIndicators;
    struct db2LoadIn *piLoadInfoIn;
    struct db2LoadOut *poLoadInfoOut;
    struct db2PartLoadIn *piPartLoadInfoIn;
    struct db2PartLoadOut *poPartLoadInfoOut;
    db2int16 iCallerAction;
} db2LoadStruct;

typedef SQL_STRUCTURE db2LoadIn
{
    db2Uint64                                iRowcount;
```

```

    db2UInt64
    char
    db2UInt32
    db2UInt32
    db2UInt32
    db2UInt32
    db2UInt16
    db2UInt16
    db2UInt16
    db2UInt16
    db2UInt16
    db2UInt16
    db2UInt16
    db2UInt16
    char
    char
} db2LoadIn;

typedef SQL_STRUCTURE db2LoadOut
{
    db2UInt64
    db2UInt64
    db2UInt64
    db2UInt64
    db2UInt64
    db2UInt64
} db2LoadOut;

typedef SQL_STRUCTURE db2PartLoadIn
{
    char
    char
    char
    struct db2LoadNodeList
    struct db2LoadNodeList
    db2UInt16
    db2UInt16
    db2UInt16
    db2UInt16
    struct db2LoadPortRange
    db2UInt16
    char
    char
    db2UInt16
    db2UInt16
    char
    db2UInt16
    SQL_PDB_NODE_TYPE
} db2PartLoadIn;

typedef SQL_STRUCTURE db2LoadNodeList
{
    SQL_PDB_NODE_TYPE
    db2UInt16
} db2LoadNodeList;

    iRestartcount;
    *piUseTablespace;
    iSavecount;
    iDataBufferSize;
    iSortBufferSize;
    iWarningcount;
    iHoldQuiesce;
    iCpuParallelism;
    iDiskParallelism;
    iNonrecoverable;
    iIndexingMode;
    iAccessLevel;
    iLockWithForce;
    iCheckPending;
    iRestartphase;
    iStatsOpt;

    oRowsRead;
    oRowsSkipped;
    oRowsLoaded;
    oRowsRejected;
    oRowsDeleted;
    oRowsCommitted;

    *piHostname;
    *piFileTransferCmd;
    *piPartFileLocation;
    *piOutputNodes;
    *piPartitioningNodes;
    *piMode;
    *piMaxNumPartAgents;
    *piIsolatePartErrs;
    *piStatusInterval;
    *piPortRange;
    *piCheckTruncation;
    *piMapFileInput;
    *piMapFileOutput;
    *piTrace;
    *piNewline;
    *piDistfile;
    *piOmitHeader;
    *piRunStatDBPartNum;

```

```

typedef SQL_STRUCTURE db2LoadPortRange
{
    db2UInt16                iPortMin;
    db2UInt16                iPortMax;
} db2LoadPortRange;

typedef SQL_STRUCTURE db2PartLoadOut
{
    db2UInt64                oRowsRdPartAgents;
    db2UInt64                oRowsRejPartAgents;
    db2UInt64                oRowsPartitioned;
    struct db2LoadAgentInfo  *poAgentInfoList;
    db2UInt32                iMaxAgentInfoEntries;
    db2UInt32                oNumAgentInfoEntries;
} db2PartLoadOut;

typedef SQL_STRUCTURE db2LoadAgentInfo
{
    db2int32                 oSqlcode;
    db2UInt32                oTableState;
    SQL_PDB_NODE_TYPE        oNodeNum;
    db2UInt16                oAgentType;
} db2LoadAgentInfo;
/* ... */

```

汎用 API 構文:

```

/* File: db2ApiDf.h */
/* API: Load */
/* ... */
SQL_API_RC SQL_API_FN
db2gLoad (
    db2UInt32 versionNumber,
    void * pParmStruct,
    struct sqlca * pSqlca);

typedef SQL_STRUCTURE db2gLoadStruct
{
    struct sqlu_media_list *piSourceList;
    struct sqlu_media_list *piLobPathList;
    struct sqldcol *piDataDescriptor;
    struct sqlchar *piActionString;
    char *piFileType;
    struct sqlchar *piFileTypeMod;
    char *piLocalMsgFileName;
    char *piTempFilesPath;
    struct sqlu_media_list *piVendorSortWorkPaths;
    struct sqlu_media_list *piCopyTargetList;
    db2int32 *piNullIndicators;
    struct db2gLoadIn *piLoadInfoIn;
    struct db2LoadOut *poLoadInfoOut;
    struct db2gPartLoadIn *piPartLoadInfoIn;
    struct db2PartLoadOut *poPartLoadInfoOut;
    db2int16 iCallerAction;
}

```



```

    db2UInt16 iFileTypeLen;
    db2UInt16 iLocalMsgFileLen;
    db2UInt16 iTempFilesPathLen;
} db2gLoadStruct;

typedef SQL_STRUCTURE db2gLoadIn
{
    db2UInt64                iRowCount;
    db2UInt64                iRestartcount;
    char                    *piUseTablespace;
    db2UInt32                iSavecount;
    db2UInt32                iDataBufferSize;
    db2UInt32                iSortBufferSize;
    db2UInt32                iWarningcount;
    db2UInt16                iHoldQuiesce;
    db2UInt16                iCpuParallelism;
    db2UInt16                iDiskParallelism;
    db2UInt16                iNonrecoverable;
    db2UInt16                iIndexingMode;
    db2UInt16                iAccessLevel;
    db2UInt16                iLockWithForce;
    db2UInt16                iCheckPending;
    char                    iRestartphase;
    char                    iStatsOpt;
    db2UInt16                iUseTablespaceLen;
} db2gLoadIn;

typedef SQL_STRUCTURE db2LoadOut
{
    db2UInt64                oRowsRead;
    db2UInt64                oRowsSkipped;
    db2UInt64                oRowsLoaded;
    db2UInt64                oRowsRejected;
    db2UInt64                oRowsDeleted;
    db2UInt64                oRowsCommitted;
} db2LoadOut;

typedef SQL_STRUCTURE db2gPartLoadIn
{
    char                    *piHostname;
    char                    *piFileTransferCmd;
    char                    *piPartFileLocation;
    struct db2LoadNodeList *piOutputNodes;
    struct db2LoadNodeList *piPartitioningNodes;
    db2UInt16                *piMode;
    db2UInt16                *piMaxNumPartAgents;
    db2UInt16                *piIsolatePartErrs;
    db2UInt16                *piStatusInterval;
    struct db2LoadPortRange *piPortRange;
    db2UInt16                *piCheckTruncation;
    char                    *piMapFileInput;
    char                    *piMapFileOutput;
    db2UInt16                *piTrace;
    db2UInt16                *piNewline;
    char                    *piDistfile;
}

```

db2Load - コード

```
    db2UInt16                *piOmitHeader;
    SQL_PDB_NODE_TYPE        *piRunStatDBPartNum;
    db2UInt16                iHostnameLen;
    db2UInt16                iFileTransferLen;
    db2UInt16                iPartFileLocLen;
    db2UInt16                iMapFileInputLen;
    db2UInt16                iMapFileOutputLen;
    db2UInt16                iDistfileLen;
} db2gPartLoadIn;

typedef SQL_STRUCTURE db2LoadNodeList
{
    SQL_PDB_NODE_TYPE        *piNodeList;
    db2UInt16                iNumNodes;
} db2LoadNodeList;

typedef SQL_STRUCTURE db2LoadPortRange
{
    db2UInt16                iPortMin;
    db2UInt16                iPortMax;
} db2LoadPortRange;

typedef SQL_STRUCTURE db2PartLoadOut
{
    db2UInt64                oRowsRdPartAgents;
    db2UInt64                oRowsRejPartAgents;
    db2UInt64                oRowsPartitioned;
    struct db2LoadAgentInfo   *poAgentInfoList;
    db2UInt32                iMaxAgentInfoEntries;
    db2UInt32                oNumAgentInfoEntries;
} db2PartLoadOut;

typedef SQL_STRUCTURE db2LoadAgentInfo
{
    db2int32                 oSqlcode;
    db2UInt32                oTableState;
    SQL_PDB_NODE_TYPE        oNodeNum;
    db2UInt16                oAgentType;
} db2LoadAgentInfo;
/* ... */
```

API パラメーター:

versionNumber

入力。 2 番目のパラメーター *pParmStruct* として渡される構造のバージョンとリリースのレベルを指定します。

pParmStruct

入力。 *db2LoadStruct* 構造を指すポインター。

pSqlca 出力。 *sqlca* 構造へのポインター。

piSourceList

入力。ソース・ファイル、装置、ベンダー、パイプ、または SQL ステートメントを提供するのに使用される、*sqlu_media_list* 構造を指すポインター。

この構造に提供される情報は、*media_type* フィールドの値によって異なります。有効な値 (*sqlutil* で定義) は、以下のとおりです。

SQLU_SQL_STMT

media_type フィールドがこの値に設定されている場合、呼び出し側は、ターゲット・フィールドの *pStatement* フィールドで SQL 照会を提供します。*pStatement* フィールドは、*sqlu_statement_entry* のタイプです。セッション・フィールドは値を 1 に設定していなければなりません。これは、ロード・ユーティリティーはロードごとに 1 つの SQL 照会だけを受け取るからです。

SQLU_SERVER_LOCATION

media_type フィールドがこの値に設定されている場合、呼び出し側から *sqlu_location_entry* 構造によって情報が提供されます。*sessions* フィールドは、提供される *sqlu_location_entry* 構造の数を示します。これは、ファイル、装置、および Named PIPE に使用されます。

SQLU_CLIENT_LOCATION

media_type フィールドがこの値に設定されている場合、呼び出し側から *sqlu_location_entry* 構造によって情報が提供されます。*sessions* フィールドは、提供される *sqlu_location_entry* 構造の数を示します。これは、完全修飾ファイル、および Named PIPE に使用されます。この *media_type* が有効なのは、リモートで接続されているクライアントを使用して API を呼び出している場合だけであることを注意してください。

SQLU_TSM_MEDIA

media_type フィールドがこの値に設定されている場合、*sqlu_vendor* 構造が使用されます。*filename* には、ロードされるデータにユニークな ID が入ります。*sessions* の値がいくつであっても、*sqlu_vendor* 項目の数は 1 つだけにする必要があります。*sessions* フィールドは、開始される TSM セッションの数を示します。ロード・ユーティリティーは、異なるシーケンス番号を持つセッションを開始しますが、ロードされるデータは、1 つの *sqlu_vendor* 項目にあるものと同じです。

SQLU_OTHER_MEDIA

media_type フィールドがこの値に設定されている場合、*sqlu_vendor* 構造が使用されます。*shr_lib* には共有ライブラリー名、*filename* にはロードされるデータにユニークな ID が入ります。*sessions* の値がいくつであっても、*sqlu_vendor* 項目の数は 1 つだけにする必要があります。*sessions* フィールドは、開始されるその他のベンダー・セッ

ションの数を示します。ロード・ユーティリティーは、異なるシーケンス番号を持つセッションを開始しますが、ロードされるデータは、1つの *sqlu_vendor* 項目にあるものと同じです。

piLobPathList

入力。 *sqlu_media_list* 構造を指すポインター。ファイル・タイプが IXF、ASC、および DEL の場合は、ロードされる個々の LOB ファイルのロケーションを識別する、完全修飾パスまたは装置のリスト。ファイル名は、IXF、ASC、または DEL ファイルで検索され、提供されたパスに追加されません。

この構造に提供される情報は、*media_type* フィールドの値によって異なります。有効な値 (sqlutil で定義) は、以下のとおりです。

SQLU_LOCAL_MEDIA

この値に設定されている場合、呼び出し側から *sqlu_media_entry* 構造によって情報が提供されます。 *sessions* フィールドは、提供される *sqlu_media_entry* 構造の数を示します。

SQLU_TSM_MEDIA

この値に設定されている場合、*sqlu_vendor* 構造が使用されます。 *filename* には、ロードされるデータにユニークな ID が入ります。 *sessions* の値がいくつであっても、*sqlu_vendor* 項目の数は 1 つだけにする必要があります。 *sessions* フィールドは、開始される TSM セッションの数を示します。ロード・ユーティリティーは、異なるシーケンス番号を持つセッションを開始しますが、ロードされるデータは、1つの *sqlu_vendor* 項目にあるものと同じです。

SQLU_OTHER_MEDIA

この値に設定されている場合、*sqlu_vendor* 構造が使用されます。 *shr_lib* には共有ライブラリー名、*filename* にはロードされるデータにユニークな ID が入ります。 *sessions* の値がいくつであっても、*sqlu_vendor* 項目の数は 1 つだけにする必要があります。 *sessions* フィールドは、開始されるその他のベンダー・セッションの数を示します。ロード・ユーティリティーは、異なるシーケンス番号を持つセッションを開始しますが、ロードされるデータは、1つの *sqlu_vendor* 項目にあるものと同じです。

piDataDescriptor

入力。外部ファイルからロードするよう選択された列に関する情報を含む *sqldcol* 構造を指すポインター。

pFileType パラメーターが SQL_ASC に設定されている場合、この構造の *dcolmeth* フィールドは、SQL_METH_L または SQL_METH_D に設定し、ファイル名は、開始と終了の対および NULL 標識の位置を含む POSITIONSFIL 構造の *pFileTypeMod* 修飾子とともに指定する必要があります。ユーザーは、ロードされる列ごとに開始ロケーションと終了ロケーションを指定します。

ファイル・タイプ `SQL_DEL` の場合、`dcolmeth` は `SQL_METH_P` または `SQL_METH_D` のどちらかにすることができます。 `SQL_METH_P` の場合、ソース列の位置を提供する必要があります。 `SQL_METH_D` の場合は、ファイル内の最初の列が表の最初の列にロードされ、以下同様に続きます。

ファイル・タイプが `SQL_IXF` の場合、`dcolmeth` は `SQL_METH_P`、`SQL_METH_D`、または `SQL_METH_N` のいずれかにすることができます。この場合は、`SQL_METH_N` が `sqldcol` 構造でファイル列名が提供されるべきであることを示す点を除き、`DEL` ファイルに関する規則が適用されます。

piActionString

入力。 `sqlchar` 構造を指すポインターと、それに続いて表に影響するアクションを指定する文字の配列。

文字配列の形式は、以下のようになります。

```
"INSERT|REPLACE|RESTART|TERMINATE
INTO tname [(column_list)]
[ATALINK SPECIFICATION datalink-spec]
[FOR EXCEPTION e_tname]"
```

INSERT

既存の表データを変更することなく、ロードされたデータを表に追加します。

REPLACE

表から既存データをすべて削除し、ロードされたデータを挿入します。表定義および索引定義は変更されません。

RESTART

以前に割り込みを受けたロード操作を再開します。ロード操作は、ロード、作成、または削除フェーズの最後の整合点から自動的に続行されます。

TERMINATE

以前に割り込みを受けたロード操作を終了し、ロード操作が開始された時点まで操作をロールバックします。途中で整合点があっても通過します。その操作に関係する表スペースの状態は通常に戻され、すべての表オブジェクトの整合性が保たれます (索引オブジェクトが無効とマークされる場合がありますが、そのような場合には、次のアクセス時に索引の再作成が自動的に行われます)。表の存在する表スペースがロード・ペンディング状態でなければ、このオプションは表スペースの状態に影響しません。

ロード終了オプションでは、表スペースのバックアップ・ペンディング状態は解除されません。

tname データのロード先の表の名前。システム表または宣言された一時表を指定することはできません。別名、完全修飾、または未修飾の表名を

指定できます。修飾された表名は、*schema.tablename* の形式になります。未修飾の表名を指定すると、その表は CURRENT SCHEMA で修飾されます。

(*column_list*)

データの挿入先の表の列名のリスト。列名は、コンマで区切らなければなりません。名前にスペースまたは小文字が含まれている場合には、それを引用符で囲まなければなりません。

DATALINK SPECIFICATION *datalink-spec*

DB2 データ・リンクに関連するパラメーターを指定します。これらのパラメーターは、LOAD コマンドと同じ構文を使って指定できます。

FOR EXCEPTION *e_tbname*

エラーが発生した行のコピー先となる例外表を指定します。ユニーク索引または主キー索引に違反した行がすべてコピーされます。DATALINK 例外も例外表に取り込まれます。

piFileType

入力。入力データ・ソースの形式を示すストリング。サポートされている外部の形式 (sqlutil で定義) は、以下のとおりです。

SQL_ASC

区切り文字なし ASCII。

SQL_DEL

区切り文字付き ASCII。これは dBase プログラム、BASIC プログラム、IBM Personal Decision Series プログラム、およびその他の多数のデータベース・マネージャー / ファイル・マネージャーとの交換のための形式です。

SQL_IXF

IXF (統合交換フォーマットの PC バージョン)。表からデータをエクスポートする場合の推奨方式で、同じ表または別のデータベース・マネージャー表にそれをロードすることが可能です。

SQL_CURSOR

SQL 照会。 *piSourceList* パラメーターによって渡された *sqlu_media_list* 構造のタイプは SQLU_SQL_STMT で、実際の SQL 照会を参照し、それに対して宣言されているカーソルは参照しません。

piFileTypeMod

入力。 *sqlchar* 構造を指すポインターと、それに続いて 1 つ以上の処理オプションを指定する文字の配列。このポインターが NULL であるか、このポインターが指す構造に 1 文字も入っていない場合、このアクションはデフォルトの指定が選択されたものとして解釈されます。

サポートされるすべてのファイル・タイプに、すべてのオプションを使用できるわけではありません。

piLocalMsgFileName

入力。出力メッセージの書き込み先となるローカル・ファイルの名前を含むストリング。

piTempFilePath

入力。一時ファイル用のサーバー上で使用されるパス名を含むストリング。一時ファイルは、メッセージや整合点を格納したり、フェーズ情報を削除したりするために作成されます。

piVendorSortWorkPaths

入力。ベンダー・ソート作業ディレクトリーを指定する *sqlu_media_list* 構造を指すポインター。

piCopyTargetList

入力。 *sqlu_media_list* 構造へのポインター。これは、(コピー・イメージを作成する予定の場合) コピー・イメージの書き込み先となるターゲット・パス、装置、または共有ライブラリーのリストを提供するときに使用します。

この構造に入力する値は、 *media_type* フィールドの値によって異なります。このフィールドに有効な値 (*sqlutil* で定義) は、以下のとおりです。

SQLU_LOCAL_MEDIA

コピーをローカル・メディアに書き込む予定の場合、 *media_type* をこの値に設定し、ターゲットに関する情報を *sqlu_media_entry* 構造に提供してください。 *sessions* フィールドは、提供される *sqlu_media_entry* 構造の数を示します。

SQLU_TSM_MEDIA

コピーを TSM に書き込む予定の場合、この値を使用してください。それ以外の情報は特に必要ありません。

SQLU_OTHER_MEDIA

ベンダー製品を使用する予定の場合、この値を使用し、 *sqlu_vendor* 構造を介して追加の情報を提供してください。この構造の *shr_lib* フィールドをベンダー製品の共有ライブラリー名に設定してください。 *sessions* の値に関係なく、1 つの *sqlu_vendor* 項目だけを提供してください。 *sessions* フィールドは、提供される *sqlu_media_entry* 構造の数を示します。ロード・ユーティリティーは、異なるシーケンス番号を持つセッションを開始しますが、ロードされるデータは、1 つの *sqlu_vendor* 項目で提供されているものと同じです。

piNullIndicators

入力。ASC ファイルの場合にのみ使用します。列データが NULL 可能であるかどうかを示す整数の配列です。この配列のエレメントと、データ・ファイルからロードされる列との間には、1 対 1 の順序付けられた対応関係があり

ます。要するに、エレメントの数は、*pDataDescriptor* パラメーターの *dcolnum* フィールドと同じでなければなりません。配列の各エレメントには、NULL 標識フィールドとして使用される、データ・ファイル内のロケーションを識別する数値、または表列が NULL 可能ではないことを示すゼロが含まれます。エレメントがゼロでない場合には、データ・ファイル内の識別されたロケーションに Y または N が入っていないければなりません。Y は表列のデータが NULL であることを示し、N は表列のデータが NULL ではないことを示します。

piLoadInfoIn

入力。 *db2LoadIn* 構造を指すポインター。

poLoadInfoOut

入力。 *db2LoadOut* 構造を指すポインター。

piPartLoadInfoIn

入力。 *db2PartLoadIn* 構造を指すポインター。

poPartLoadInfoOut

出力。 *db2PartLoadOut* 構造を指すポインター。

iCallerAction

入力。呼び出し側が要求するアクションを示します。有効な値 (*sqlutil* で定義) は、以下のとおりです。

SQLU_INITIAL

最初の呼び出し。この値 (または *SQLU_NOINTERRUPT*) は、API への最初の呼び出しの際には必ず使用してください。

SQLU_NOINTERRUPT

最初の呼び出し。処理を中断しません。この値 (または *SQLU_INITIAL*) は、API への最初の呼び出しの際には必ず使用してください。

最初の呼び出しまたは後続の呼び出しのいずれかが戻され、要求されたロード操作が完了する前に呼び出し側のアプリケーションが何らかのアクションを行うことが必要な場合、呼び出し側のアクションを以下のどちらかに設定する必要があります。

SQLU_CONTINUE

処理の継続。この値を使用できるのは、最初の呼び出しが戻されたときにユーティリティーがユーザー入力 (たとえば、テープの終わり条件への応答) を要求した後で、API への後続呼び出しを出す場合だけです。この値は、ユーティリティーが要求したユーザー・アクションが完了したら、ユーティリティーが最初の要求の処理を続行するよう指定するものです。

SQLU_TERMINATE

処理の終了。ロード中の表スペースを LOAD_PENDING 状態にしたまま、ロード・ユーティリティを早期に終了させます。このオプションは、これ以上データの処理が行われない場合に指定します。

SQLU_ABORT

処理の終了。ロード中の表スペースを LOAD_PENDING 状態にしたまま、ロード・ユーティリティを早期に終了させます。このオプションは、これ以上データの処理が行われない場合に指定します。

SQLU_RESTART

処理の再開。

SQLU_DEVICE_TERMINATE

単一の装置の終了。このオプションは、ユーティリティが装置からの読み取りを停止しても、データの処理をさらに続ける場合に指定します。

iFileTypeLen

入力。 *iFileType* の長さ (バイト単位) を指定します。

iLocalMsgFileLen

入力。 *iLocalMsgFileName* の長さ (バイト単位) を指定します。

iTempFilesPathLen

入力。 *iTempFilesPath* の長さ (バイト単位) を指定します。

iRowcount

入力。ロードされる物理レコードの数。これを使用すると、ファイル内の最初の *rowcnt* 個の行だけをロードすることができます。

iRestartcount

入力。将来の使用のために予約されています。

piUseTablespace

入力。索引が再作成されている場合、索引のシャドー・コピーが表スペース *iUseTablespaceName* 内に作成され、ロード終了時に元の表スペースにコピーされます。システム TEMPORARY 表スペースのみ、このオプションを使用できます。指定されない場合、シャドー索引が、索引オブジェクトと同じ表スペース内に作成されます。

シャドー・コピーが索引オブジェクトと同じ表スペース内に作成される場合、古い索引オブジェクトを介したシャドー索引オブジェクトのコピーは瞬時に終了します。シャドー・コピーが索引オブジェクトとは異なる表スペースにある場合、物理コピーが実行されます。これにはかなりの入出力および時間を要します。コピーは、表がロード終了時にオフラインであるときに行われます。

iAccessLevel が SQLU_ALLOW_NO_ACCESS である場合、このフィールドは無視されます。

ユーザーが INDEXING MODE REBUILD または INDEXING MODE AUTOSELECT を指定しない場合、このオプションは無視されます。このオプションは INDEXING MODE AUTOSELECT が選択され、ロードが索引を徐々に更新することを選択した場合にも無視されます。

iSavecount

整合点を確立する前にロードするレコードの数。この値はページ・カウントに変換され、エクステント・サイズのインターバルに切り上げられます。それぞれの整合点でメッセージが発行されるため、*db2LoadQuery* - 照会のロード を用いてロード操作をモニターする場合には、このオプションを選択する必要があります。*savecnt* の値を大きく設定しておかないと、それぞれの整合点で実行される活動の同期がとられるときにパフォーマンスに影響が及びます。

デフォルト値は 0 であり、これは、必要のない限り整合点が確立されないことを意味します。

iDataBufferSize

ユーティリティ内でデータ転送用のバッファ・スペースとして使用される 4KB ページの数 (並行処理の度合いとは無関係)。指定された値がアルゴリズムの最小値よりも小さい場合には、必要最低限のページが使用され、警告は戻されません。

このメモリーは、ユーティリティ・ヒープから直接に割り振られます (ユーティリティ・ヒープのサイズは、*util_heap_sz* データベース構成パラメーターを用いて修正できます)。

値を指定しないと、実行時にユーティリティによって適切なデフォルトが計算されます。デフォルトは、表の特性だけでなく、ローダーのインスタンス生成時にユーティリティ・ヒープ中で使用可能なフリー・スペースの割合に基づいています。

iSortBufferSize

入力。このオプションは、ロード操作時に SORTHEAP データベース構成パラメーターをオーバーライドする値を指定します。これは表を索引とともにロードする場合、および *iIndexingMode* パラメーターが *SQLU_INX_DEFERRED* として指定されない場合のみ関係があります。指定される値は、SORTHEAP の値を超えることはできません。このパラメーターは、一般的な照会処理にも影響を与える SORTHEAP の値を変更せずに、LOAD によって使用されるソート・メモリーをスロットルするために役立ちます。

iWarningcount

入力。*warningcnt* 個の警告後に、ロード操作を停止します。このパラメーターは、警告は予期されないが、正しいファイルと表が使用されていることの確認が望まれる場合に設定してください。*warningcnt* が 0 であるか、またはこのオプションを指定していない場合には、ロード操作は、発行された警告の数に関係なく続行されます。

警告のしきい値を超過したためにロード操作が停止された場合には、RESTART モードでもう一度ロード操作を開始することができます。ロード操作は最後の整合点から自動的に続行されます。または、入力ファイルの先頭から REPLACE モードであらためてロード操作を開始できます。

iHoldQuiesce

入力。ユーティリティによって、ロード後に表を排他静止状態のままにする場合は TRUE、それ以外の場合は FALSE に値が設定されるフラグ。

iCpuParallelism

入力。ユーティリティが表オブジェクトの作成時にレコードを解析、変換、および形式化するために作成するプロセスつまりスレッドの数。このパラメーターは、パーティション内並列処理を活用するために設計されています。これは、事前にソートされたデータをロードする際に役立ちます (ソース・データのレコード順序が保持されるため)。このパラメーターの値がゼロである場合には、ロード・ユーティリティは実行時に適切なデフォルト値を使用します。注: このパラメーターが LOB または LONG VARCHAR フィールドを含む表について使用されると、システム CPU の数やユーザーによって指定された値に関係なく、値は 1 になります。

iDiskParallelism

入力。ユーティリティがデータを表スペース・コンテナに書き込むために作成するプロセスつまりスレッドの数。値を指定しないと、ユーティリティは表スペース・コンテナの数と表の特性に基づいて適切なデフォルトを選択します。

iNonrecoverable

入力。ロード・トランザクションがリカバリー不能としてマークされ、後続のロールフォワード・アクションによってリカバリーできない場合には、`SQLU_NON_RECOVERABLE_LOAD` に設定します。ロールフォワード・ユーティリティは、このトランザクションをスキップし、データがロードされようとしていた表を「無効」としてマークします。さらに、ユーティリティは、その表に対する後続のすべてのトランザクションを無視します。ロールフォワードが完了したら、そのような表はドロップするしかありません。このオプションを使用すると、表スペースはロード操作後にバックアップ・ペンディング状態になりません。また、ロード操作中にロードされたデータのコピーが作成される必要もなくなります。ロード・トランザクションがリカバリー可能としてマークされる場合には、`SQLU_RECOVERABLE_LOAD` に設定します。

iIndexingMode

入力。索引付けモードを指定します。有効な値 (`sqlutil` で定義) は、以下のとおりです。

SQLU_INX_AUTOSELECT

LOAD は REBUILD と INCREMENTAL 索引モードの間で選択します。

SQLU_INX_REBUILD

表索引を再作成します。

SQLU_INX_INCREMENTAL

既存の索引を拡張します。

SQLU_INX_DEFERRED

表索引を更新しません。

iAccessLevel

入力。アクセス・レベルを指定します。有効な値は以下のとおりです。

SQLU_ALLOW_NO_ACCESS

ロードが表を排他ロックするように指定します。

SQLU_ALLOW_READ_ACCESS

表の元データ (非差分部分) が、ロードが進行中の間、リーダーに対して可視のままであるように指定します。このオプションは、ロードの付加 (たとえば、ロードの挿入など) に対してのみ有効です。ロード置換に対しては無視されます。

iLockWithForce

入力。ブール・フラグ。 TRUE に設定された場合、ロードは必要に応じて他のアプリケーションに対し、必ず即時に表ロックを得るように強制します。このオプションは、FORCE APPLICATIONS コマンド (SYSADM または SYSCTRL) と同じ権限を必要とします。

SQLU_ALLOW_NO_ACCESS ロードは、ロード操作の開始時に、アプリケーションの競合を強制終了させることができます。ロードの開始時に、ユーティリティーは、表の照会または変更のいずれかを試みるアプリケーションを強制する場合があります。

SQLU_ALLOW_READ_ACCESS ロードは、ロード操作の開始時または終了時に、アプリケーションの競合を強制終了させることができます。ロードの開始時に、ロード・ユーティリティーは、表の変更を試みるアプリケーションを強制する場合があります。ロードの終了時に、ロード・ユーティリティーは、表の照会または変更のいずれかを試みるアプリケーションを強制する場合があります。

iCheckPending

入力。表をチェック・ペンディング状態にするように指定します。

SQLU_CP_IMMEDIATE が指定されている場合、チェック・ペンディング状態は即時にすべての従属表および下層表にカスケードされます。

SQLU_CP_DEFERRED が指定されている場合、チェック・ペンディング状態の従属表へのカスケードは、ターゲット表の保水性違反がチェックされるまで据え置かれます。このオプションが指定されていない場合、SQLU_CP_DEFERRED がデフォルトとなります。

iRestartphase

入力。再始動フェーズ。

iStatsOpt

入力。収集する統計の細分性。有効な値は以下のとおりです。

SQLU_STATS_NONE

統計は収集されません。

SQL_STATS_INDEX

表統計なしの基本索引統計。

SQL_STATS_TABLE

索引統計なしの基本表統計。

SQL_STATS_BOTH

表および索引の基本統計。

SQL_STATS_EXTTABLE_ONLY

表および分散統計。

SQL_STATS_EXTTABLE_INDEX

表および分散統計と基本索引統計。

SQL_STATS_EXTINDEX_ONLY

索引の拡張統計のみ。

SQL_STATS_EXTINDEX_TABLE

索引の拡張統計と表の基本統計。

SQL_STATS_ALL

表および分散統計を持つ拡張索引統計。

iUseTablespaceLen

入力。 *piUseTablespace* の長さ (バイト単位)。

oRowsRead

出力。ロード操作中に読み取られたレコードの数。

oRowsSkipped

出力。ロード操作が開始される前にスキップされたレコードの数。

oRowsLoaded

出力。ターゲット表にロードされた行の数。

oRowsRejected

出力。ロードできなかったレコードの数。

oRowsDeleted

出力。削除された重複行の数。

oRowsCommitted

出力。処理されたレコードの合計数。正常にロードされ、データベースにコミットされたレコードの数と、スキップまたはリジェクトされたレコードの数の合計。

piHostname

入力。 *iFileTransferCmd* パラメーターのホスト名。 NULL の場合、ホスト名のデフォルトは「nohost」です。

piFileTransferCmd

入力。ファイル転送コマンドのパラメーター。必要ない場合、NULL に設定する必要があります。このパラメーターの詳細については、「Data Movement Guide」を参照してください。

piPartFileLocation

入力。 PARTITION_ONLY、LOAD_ONLY、および LOAD_ONLY_VERIFY_PART モードでは、このパラメーターは、パーティション・ファイルの完全修飾されたロケーションを指定するために使用できません。このロケーションは、*piOutputNodes* オプションで指定された各パーティションに存在している必要があります。

SQL_CURSOR 以外のファイル・タイプの場合、以下の規則が適用されます。 PARTITION_ONLY モードでは、このパラメーターの値が NULL の場合、ロードがパーティション・ファイルを作成するロケーションとして現行ディレクトリーが使用されます。 LOAD_ONLY モードおよび

LOAD_ONLY_VERIFY_PART モードでは、このパラメーターが NULL の場合、 *piSourceList* パラメーターで指定された入力ファイル名のパス接頭部として、その名前が完全修飾されている限り、入力ファイルのロケーションがとられます。そうでない場合は、入力ファイルのロケーションとして現行ディレクトリーが使用されます。

SQL_CURSOR ファイル・タイプの場合、このパラメーターは NULL にすることはできません。ロケーションはパスを参照しませんが、完全修飾されたファイル名を参照します。これは、PARTITION_ONLY モードの場合は、各出力パーティションで作成されたパーティション・ファイルの完全修飾された基本ファイル名、または LOAD_ONLY モードの場合は、各パーティションから読み取られるファイルのロケーションです。

PARTITION_ONLY モードの場合、ターゲット表に LOB 列が存在するならば、指定された基本名のファイルが複数作成されることがあります。

piOutputNodes

入力。ロード出力パーティションのリスト。 NULL は、ターゲット表が定義されたすべてのノードを示します。

piPartitioningNodes

入力。区分化ノードのリスト。 NULL はデフォルトを示します。デフォルト

を決定する方法については、データ移動ユーティリティー ガイドおよびリファレンスの『Load コマンド』を参照してください。

piMode

入力。パーティション・データベースのロード・モードを指定します。有効な値 (db2ApiDf で定義) は、以下のとおりです。

PARTITION_AND_LOAD

データはパーティション分割され (おそらく並列で)、対応するデータベース・パーティション上に同時にロードされます。

PARTITION_ONLY

データはパーティション分割され (おそらく並列で)、出力は各ロード・パーティション上の、指定のロケーションのファイルに書き込まれます。出力ファイルのロケーションを指定する方法については、*piPartFileLocation* パラメーターの説明を参照してください。

LOAD_ONLY

データはすでにパーティション分割されていると想定されます。パーティション分割プロセスはスキップされ、データは対応するデータベース・パーティションに同時にロードされます。入力ファイルのロケーションを指定する方法については、*piPartFileLocation* パラメーターの説明を参照してください。

LOAD_ONLY_VERIFY_PART

データはすでにパーティション分割されていると想定されます。しかし、データ・ファイルにはパーティション・ヘッダーがありません。区分化プロセスはスキップされ、データは対応するデータベース・パーティションに同時にロードされます。ロード操作時に、各行が正しいパーティション上にあるかがチェックされます。パーティション違反のある行は、ダンプ・ファイル修飾子が指定されている場合、ダンプ・ファイルに置かれます。指定されていない場合、その行は廃棄されます。パーティション違反が特定のロード・パーティションに存在する場合、1 つの警告が、パーティションのロード・メッセージ・ファイルに書き込まれます。入力ファイルのロケーションを指定する方法については、*piPartFileLocation* パラメーターの説明を参照してください。

ANALYZE

すべてのデータベース・パーティション間で均等に分散される最適な区分化マップが生成されます。

piMaxNumPartAgents

入力。パーティション・エージェントの最大数。NULL 値はデフォルトを示します。デフォルトは 25 です。

piIsolatePartErrs

入力。ロード操作が、個々のパーティションで発生するエラーに対応する方法を示します。有効な値 (db2ApiDf で定義) は、以下のとおりです。

DB2LOAD_SETUP_ERRS_ONLY

このモードでは、セットアップ時にパーティションで生じるエラー (たとえば、パーティションへのアクセスに関する問題や、パーティションの表スペースまたは表へのアクセスに関する問題) によって、失敗したパーティションではロード操作が停止してしましますが、残りのパーティションでは操作が継続されます。データのロード中にパーティションで生じるエラーによって、全操作が失敗し、各パーティションの最後の整合点にロールバックされます。

DB2LOAD_LOAD_ERRS_ONLY

このモードでは、セットアップ時にパーティションで生じるエラーによって、ロード操作全体が失敗します。データのロード中にエラーが生じた場合、エラーのあるパーティションは最後の整合点にロールバックされます。ロード操作は、失敗が生じるまで、またはすべてのデータがロードされるまで、残りのパーティションで続行します。すべてのデータがロードされたパーティションでは、ロード操作後は、データは可視ではありません。他のパーティションで生じたエラーのため、トランザクションは打ち切られます。すべてのパーティション上のデータは、ロードの再開操作が実行されるまで、不可視のままです。これにより、新たにロードされたデータはパーティション上で可視になります。パーティションでは、ロード操作が完了し、エラーが発生したパーティションでのロード操作が再開されます。

注: *iAccessLevel* が `SQLU_ALLOW_READ_ACCESS` に設定されている場合、またはコピー・ターゲットが指定されている場合、このモードは使用できません。

DB2LOAD_SETUP_AND_LOAD_ERRS

このモードでは、セットアップまたはデータのロード時に生じるパーティション・レベルのエラーによって、影響を受けたパーティション上でのみ、処理が停止します。 `DB2LOAD_LOAD_ERRS_ONLY` モードと同様、データ・ロード中にパーティション・エラーが生じた場合、すべてのパーティション上のデータは、ロードの再開操作が実行されるまで不可視のままです。

注: *iAccessLevel* が `SQLU_ALLOW_READ_ACCESS` に設定されている場合、またはコピー・ターゲットが指定されている場合、このモードは使用できません。

DB2LOAD_NO_ISOLATION

ロード操作時にエラーが生じると、トランザクションは打ち切られます。

パラメーターが NULL の場合、*iAccessLevel* が `SQLU_ALLOW_READ_ACCESS` に設定されない限り、またはコピー・ターゲットが指定されない限り、デフォルトは `DB2LOAD_LOAD_ERRS_ONLY` になります。設定または指定されている場合、デフォルトは `DB2LOAD_NO_ISOLATION` です。

piStatusInterval

入力。進行メッセージを生成する前に、ロードするデータの MB 数を指定します。有効な値は、1 ~ 400 の範囲の整数です。NULL が指定される場合、デフォルト値の 100 が使用されます。

piPortRange

入力。内部通信用の TCP ポート範囲。NULL の場合、使用されるポート範囲は 6000 ~ 6063 です。

piCheckTruncation

入力。ロードで入出力時にレコードの切り捨てをチェックします。有効な値は TRUE および FALSE です。NULL の場合、デフォルトは FALSE です。

piMapFileInput

入力。パーティション・マップの入力ファイル名。モードが ANALYZE ではない場合、このパラメーターは NULL に設定する必要があります。モードが ANALYZE の場合、このパラメーターは指定する必要があります。

piMapFileOutput

入力。パーティション・マップの出力ファイル名。piMapFileInput に対する規則は、ここでも同じく適用されます。

piTrace

入力。すべてのデータ変換プロセスのダンプ、およびハッシュ値の出力を検討する必要がある場合、トレースするレコードの数を指定します。NULL の場合、レコード数のデフォルトは 0 です。

piNewline

入力。RECLLEN ファイル・タイプ修飾子も指定されている場合、ロードで ASC データ・レコードの終端で改行文字をチェックするように強制します。指定可能な値は TRUE および FALSE です。NULL の場合、値のデフォルトは FALSE です。

piDistfile

入力。パーティション配布ファイル名。NULL が指定された場合、値のデフォルトは「DISTFILE」です。

piOmitHeader

入力。 DB2LOAD_PARTITION_ONLY モードを使用する場合に、区分化マップのヘッダーをパーティション・ファイルに組み込まないことを示します。指定可能な値は TRUE および FALSE です。 NULL の場合、デフォルトは FALSE です。

piRunStatDBPartNum

統計を収集するデータベース・パーティションを指定します。デフォルト値は、出力パーティション・リスト内の最初のデータベース・パーティションです。

iHostnameLen

入力。 *piHostname* の長さ (バイト単位)。

iFileTransferLen

入力。 *piFileTransferCmd* の長さ (バイト単位)。

iPartFileLocLen

入力。 *piPartFileLocation* の長さ (バイト単位)。

iMapFileInputLen

入力。 *piMapFileInput* の長さ (バイト単位)。

iMapFileOutputLen

入力。 *piMapFileOutput* の長さ (バイト単位)。

iDistfileLen

入力。 *piDistfile* の長さ (バイト単位)。

piNodeList

入力。 ノード番号の配列。

iNumNodes

入力。 *piNodeList* 配列内のノードの数。 0 がデフォルトで、これはターゲット表が定義されているすべてのノードです。

iPortMin

入力。 小さいポート番号。

iPortMax

入力。 大きいポート番号。

oRowsRdPartAgents

出力。 すべてのパーティション・エージェントによって読み取られる行の総数。

oRowsRejPartAgents

出力。 すべてのパーティション・エージェントによってリジェクトされる行の総数。

oRowsPartitioned

出力。すべてのパーティション・エージェントによってパーティション分割される行の総数。

poAgentInfoList

出力。パーティション・データベースへのロード操作時には、ロード・エージェント、パーティション・エージェント、事前パーティション・エージェント、ファイル転送コマンド・エージェント、およびファイルへのロード・エージェントといった、ロード処理項目が関係してくる場合があります（これらは「Data Movement Guide」で説明されています）。 *poAgentInfoList* 出力パラメーターの目的は、呼び出し側に、ロード操作に関係した各ロード・エージェントに関する情報を戻すことです。リスト内の各項目には、以下の情報が含まれます。

- *oAgentType*。項目が記述するロード・エージェントの種類を示すタグ。
- *oNodeNum*。エージェントが実行されたパーティションの数。
- *oSqlcode*。エージェントの処理の結果の最終 *sqlcode*。
- *oTableState*。エージェントが実行されるパーティション上の表の最終状況（ロード・エージェントに関係するもののみ）。

API を呼び出す前に、このリストにメモリーを割り振るのは、API の呼び出し側の責任です。呼び出し側は、*iMaxAgentInfoEntries* パラメーターにメモリーを割り振った項目の数も示す必要があります。呼び出し側が *poAgentInfoList* を NULL に設定する場合、または *iMaxAgentInfoEntries* を 0 に設定する場合、ロード・エージェントに関する情報は戻されません。

iMaxAgentInfoEntries

入力。 *poAgentInfoList* 用にユーザーが割り振ったエージェント情報の項目の最大数。一般に、このパラメーターは、ロード操作に関係したパーティション数の 3 倍の数に設定すれば十分です。

oNumAgentInfoEntries

出力。ロード操作によって生成されたエージェント情報の項目の実際の数。*iMaxAgentInfoEntries* が *oNumAgentInfoEntries* の値以上である場合に限り、この項目数は *poAgentInfoList* パラメーターでユーザーに戻されます。*iMaxAgentInfoEntries* が *oNumAgentInfoEntries* より小さい場合、*poAgentInfoList* に戻される項目数は *iMaxAgentInfoEntries* と等しくなります。

oSqlcode

出力。エージェントの処理の結果の最終 *sqlcode*。

oTableState

出力。この出力パラメーターの目的は、ロード操作後に、表のいかなる状況も報告しないことです。その目的は、ロード処理中に表に何が起きたかについての一般情報を呼び出し側に提供するために、発生し得る表の状況の、小さなサ

ブセットだけを報告することです。この値は、ロード・エージェントにのみ関係があります。以下の値を指定することができます。

DB2LOADQUERY_NORMAL

ロードがパーティションで正常に完了し、表が **LOAD IN PROGRESS** (または **LOAD PENDING**) 状態ではなくなったことを示します。この場合、制約事項をさらに処理する必要があるために、表が引き続きチェック・ペンディング状態であることがありますが、これは正常な状態なので報告はされません。

DB2LOADQUERY_UNCHANGED

エラーが原因でロード・ジョブが処理を打ち切ったが、db2Load を呼び出す前の状態がどのようなものであっても、パーティション上の表の状態はまだ変更されていないことを示します。ロードの再始動、またはそのようなパーティション上での操作の終了を実行する必要はありません。

DB2LOADQUERY_LOADPENDING

処理中にロード・ジョブが打ち切られたが、パーティション上の表は **LOAD PENDING** 状態のままであることを示します。これは、パーティションでのロード・ジョブを、終了または再始動する必要があることを意味しています。

oNodeNum

出力。エージェントが実行されたパーティションの数。

oAgentType

出力。エージェント・タイプ。有効な値 (db2ApiDf で定義) は、以下のとおりです。

DB2LOAD_LOAD_AGENT

DB2LOAD_PARTITIONING_AGENT

DB2LOAD_PRE_PARTITIONING_AGENT

DB2LOAD_FILE_TRANSFER_AGENT

DB2LOAD_LOAD_TO_FILE_AGENT

使用上の注意:

データは、入力ファイル内に並んでいる順序でロードされます。特定の順序にしたいなら、ロードの前にデータをソートしておく必要があります。

ロード・ユーティリティーは、既存の定義に基づいて索引を作成します。ユニーク・キーの重複を処理するのに、例外表が使用されます。ユーティリティーは、参照保全を強制したり、制約検査を実行したり、ロードする表に従属するサマリー表を更新したりすることはありません。参照制約またはチェック制約を含む表は、チェック・ペンディン

グ状態になります。REFRESH IMMEDIATE として定義されているサマリー表、およびロードする表に依存するサマリー表は、チェック・ペンディング状態になります。表のチェック・ペンディング状態を解除するには、SET INTEGRITY ステートメントを発行してください。ロード操作は、複製されたサマリー表では実行できません。

クラスタリング索引の場合、ロードする前に、データをクラスタリング索引でソートする必要があります。多次元クラスターされた (MDC) 表にロードする場合は、データをソートする必要はありません。

DB2 Data Links Managerについての考慮事項

各 DATALINK 列ごとに、括弧内にそれぞれ 1 つの列を指定できます。それぞれの列の指定は、1 つ以上の DL_LINKTYPE、接頭部、および DL_URL_SUFFIX 指定で構成されます。接頭部 情報は、DL_URL_REPLACE_PREFIX、または DL_URL_DEFAULT_PREFIX の指定のいずれかになります。

DATALINK 列指定の数は、表で定義されている DATALINK 列の数と同じだけ指定できます。指定の順序は、挿入列リストの中での DATALINK 列の順序 (挿入列リストが INSERT INTO (insert-column, ...) で指定されている場合) か、または表定義内での順序 (insert-column が指定されていない場合) に従います。

たとえば、表に列 C1、C2、C3、C4、および C5 があり、そのうち C2 と C5 だけが DATALINK 型であり、挿入列 (insert-column) リストが (C1、C5、C3、C2) である場合、2 つの DATALINK 列を指定する必要があります。最初の列の指定は C5 用であり、2 番目の列の指定は C2 用です。挿入列リストを指定しない場合、最初の列の指定は C2 用になり、2 番目の列の指定は C5 用になります。

複数の DATALINK 列があり、一部の列では特別な指定が必要ではない場合、列の指定には、指定の順序をはっきりと示すために、少なくとも括弧を含める必要があります。どの列にも指定が行われない場合は、空の括弧のリスト全体を除くことができます。したがって、デフォルトで十分な場合には、DATALINK を指定する必要はありません。

FILE LINK CONTROL で定義されている DATALINK 列を含む表にデータをロードする場合は、ロード・ユーティリティを呼び出す前に、以下のステップを実行してください。(すべての DATALINK 列が NO LINK CONTROL として定義されている場合、これらのステップは必要ありません。)

1. DATALINK 列の値によって参照される Data Links サーバーに、DB2 Data Links Manager がインストールされていることを確認する。
2. データベースが DB2 Data Links Manager に登録されていることを確認する。
3. DATALINK 値として挿入されるすべてのファイルを、適切な Data Links サーバーにコピーする。
4. Data Links サーバー上の DB2 Data Links Manager に接頭部名 (複数可) を定義する。

5. (ロードする) DATALINK データによって参照される Data Links サーバーを、DB2 Data Links Manager 構成ファイルに登録します。

ロード・ユーティリティーの実行中に、DB2 と Data Links サーバー間の接続が失敗し、ロード操作も失敗してしまう場合があります。その場合には、以下のようにしてください。

1. Data Links サーバーおよび DB2 Data Links Manager を開始する。
2. ロード再開操作を起動する。

ロード操作中に失敗したリンクはデータ保全性違反と見なされ、ユニーク索引違反と同じ方法で処理されます。そのため、1 つ以上の DATALINK 列を含む表をロードする場合の特別な例外が定義されています。

入カファイル内での DATALINK 情報の表示

LINKTYPE (現在のところ URL のみサポート) は、DATALINK 情報の一部として指定されていません。LINKTYPE は、LOAD または IMPORT コマンドで指定され、PC/IXF タイプの入カファイルの場合は、適切な列記述子レコードで指定されます。

URL LINKTYPE の DATALINK 情報の構文は、以下のとおりです。



urlname と *comment* はいずれもオプションです。どちらも省略した場合、NULL 値が代入されます。

urlname

この URL 名は有効な URL 構文に適合していなければなりません。

注:

1. 現在のところ、「http」、「file」、および「unc」がスキーマ名として許可されています。
2. URL 名の接頭部 (スキーマ、ホスト、およびポート) はオプションです。接頭部がない場合は、ロード・ユーティリティーまたはインポート・ユーティリティーの DL_URL_DEFAULT_PREFIX または DL_URL_REPLACE_PREFIX 指定の接頭部が使われます。そのどちらも指定されていない場合、デフォルトの接頭部として "file://localhost" が使われます。したがって、ローカル・ファイルの場合、LOAD または IMPORT コマンド内で DATALINK 列を指定せずに、絶対パス名で指定したファイル名を URL 名として入力することができます。
3. 接頭部が URL 名に付けられている場合も、ロードまたはインポート操作時には、DL_URL_REPLACE_PREFIX で指定した異なる接頭部名によってオーバーライドされます。

4. (DL_URL_SUFFIX を指定した場合、それを追加した後の) 「path」は、リモート・サーバーにあるリモート・ファイルの絶対パス名です。相対パス名は使用できません。HTTP サーバーのデフォルトのパス接頭部は使用されません。

dl_delimiter

区切り付き ASCII (DEL) ファイル形式の場合、dlde1 修飾子で指定した文字、あるいは LOAD または IMPORT コマンドのデフォルトの文字。区切りなし ASCII (ASC) ファイル形式の場合、これを文字順序 ¥; (円記号とそれに続くセミコロン) に対応させる必要があります。空白文字 (ブランクやタブなど) は、このパラメーターに指定した値の前後に置くことができます。

comment

DATALINK 値のコメント部分。区切り付き ASCII (DEL) ファイル形式で指定する場合、comment テキストは、文字ストリング区切り文字で囲む必要があります。文字ストリング区切り文字は、デフォルトでは二重引用符 (") です。この文字ストリング区切り文字は、LOAD または IMPORT コマンドで MODIFIED BY *filetype-mod* を指定することによりオーバーライドできます。

コメントを指定しない場合、このコメントはデフォルトでは長さがゼロのストリングになります。

次に示すのは、区切り付き ASCII (DEL) ファイル形式での DATALINK データの例です。

- `http://www.almaden.ibm.com:80/mrep/intro.mpeg; "Intro Movie"`
これは、以下のような部分から構成されています。
 - スキーム = http
 - サーバー = www.almaden.ibm.com
 - パス = /mrep/intro.mpeg
 - 注釈 = "Intro Movie"
- `file://narang/u/narang; "InderPal's Home Page"`
これは、以下のような部分から構成されています。
 - スキーム = file
 - サーバー = narang
 - パス = /u/narang
 - 注釈 = "InderPal's Home Page"

次に示すのは、区切りなし ASCII (ASC) ファイル形式での DATALINK データの例です。

- `http://www.almaden.ibm.com:80/mrep/intro.mpeg¥;Intro Movie`
これは、以下のような部分から構成されています。

db2Load - ロード

- スキーム = http
- サーバー = www.almaden.ibm.com
- パス = /mrep/intro.mpeg
- 注釈 = "Intro Movie"
- file://narang/u/narang%; InderPal's Home Page
これは、以下のような部分から構成されています。
 - スキーム = file
 - サーバー = narang
 - パス = /u/narang
 - 注釈 = "InderPal's Home Page"

以下に、DATALINK データの例を示します。列のロードまたはインポート指定が DL_URL_REPLACE_PREFIX ("http://qso") であるとしています。

- http://www.almaden.ibm.com/mrep/intro.mpeg
これは、以下のような部分から構成されています。
 - スキーム = http
 - サーバー = qso
 - パス = /mrep/intro.mpeg
 - 注釈 = NULL スtring
- /u/me/myfile.ps
これは、以下のような部分から構成されています。
 - スキーム = http
 - サーバー = qso
 - パス = /u/me/myfile.ps
 - 注釈 = NULL スtring

表 220. 有効なファイル・タイプ修飾子 (LOAD)

修飾子	説明
	すべてのファイル形式
anyorder	この修飾子は、 <code>cpu_parallelism</code> パラメーターとともに使用され、ソース・データ順序を保つことが必要でないことを指定し、それによって SMP システムでさらにパフォーマンスを高めます。 <code>cpu_parallelism</code> の値が 1 の場合、このオプションは無視されます。 <code>SAVECOUNT > 0</code> の場合、整合点後のクラッシュ・リカバリーでは、データを順番にロードする必要があるため、このオプションはサポートされません。

表 220. 有効なファイル・タイプ修飾子 (LOAD) (続き)

修飾子	説明
fastparse	<p>ユーザー提供の列値に対して簡略化された構文チェックが実行され、パフォーマンスが向上します。このオプションの下でロードした表は、体系的に正確であることが保証され、ユーティリティーは、セグメント化違反またはトラップを防ぐための十分なデータ・チェックを実行することが保証されます。正しい形式のデータが正しくロードされます。</p> <p>たとえば、ASC ファイル内の整数列のフィールド項目として値 123qwr4 が検出された場合、この値は有効な数値を表すものではないので、ロード・ユーティリティーは通常構文エラーのフラグを付けます。fastparse を指定した場合、構文エラーは検出されず、整数フィールドに任意の数値がロードされます。この修飾子を使用する場合は、正しい形式のデータだけを使用するように注意してください。ASCII データでこのオプションを使用するとパフォーマンスはかなり向上しますが、PC/IXF データでこのオプションを使用しても fastparse のパフォーマンスはそれほど向上しません。これは、IXF がバイナリ形式であり、fastparse が ASCII から内部形式への構文解析および変換に影響するためです。</p> <p>ファイル・タイプが CURSOR の場合、このオプションはサポートされていません。</p>
generatedignore	<p>この修飾子はロード・ユーティリティーに対して、すべての生成列のデータがデータ・ファイル内に存在するが、それらのデータは無視するべきものであることを通知します。この結果、すべての生成列の値はユーティリティーによって生成されます。この修飾子は、generatedmissing または generatedoverride 修飾子とともに使用することはできません。</p>
generatedmissing	<p>この修飾子が指定されている場合、ユーティリティーは、生成列のデータが入力データ・ファイルに入っていない (NULL も入っていない) のものと見なします。この結果、すべての生成列の値はユーティリティーによって生成されます。この修飾子は、generatedoverride または generatedoverride 修飾子とともに使用することはできません。</p>
generatedoverride	<p>この修飾子は、ロード・ユーティリティーに対し、表内のすべての生成列に関して、(これらのタイプの列に関する通常の規則に反して) 明示的なユーザー提供の非 NULL データを受け入れるように指示します。これが役立つのは、別のデータベース・システムからデータを移行する場合や、ROLLFORWARD DATABASE コマンドで RECOVER DROPPED TABLE オプションを使用してリカバリーしたデータから表をロードする場合です。この修飾子を使用すると、データが入っていない行や NULL 不可の生成列に対する NULL データはすべてリジェクトされます (SQL3116W)。</p> <p>注: この修飾子が使用される場合、表はチェック・ペンディング状態になります。ユーザーが提供する値をチェックせずに、表をチェック・ペンディング状態から解放するには、ロード操作後に以下のコマンドを発行します。</p> <pre>SET INTEGRITY FOR <table-name> GENERATED COLUMN IMMEDIATED UNCHECKED</pre> <p>表の CHECK PENDING 状態を解除し、ユーザー定義の値のチェックを強制するには、ロード操作の後以下のコマンドを発行してください。</p> <pre>SET INTEGRITY FOR <table-name> IMMEDIATE CHECKED</pre>

表 220. 有効なファイル・タイプ修飾子 (LOAD) (続き)

修飾子	説明
identityignore	この修飾子はロード・ユーティリティに対して、IDENTITY 列のデータがデータ・ファイル内に存在するが、それらのデータは無視すべきものであることを通知します。この結果、すべて ID 値はユーティリティによって生成されます。この動作は、GENERATED ALWAYS および GENERATED BY DEFAULT IDENTITY 列のどちらの場合も同じです。つまり、GENERATED ALWAYS 列の場合、リジェクトされる行はないという意味です。この修飾子は、identitymissing または identityoverride 修飾子とともに使用することはできません。
identitymissing	この修飾子を指定すると、ユーティリティは、IDENTITY 列のデータが入力データ・ファイルに入っていない (NULL も入っていない) のものと見なし、行ごとに値を生成します。この動作は、GENERATED ALWAYS および GENERATED BY DEFAULT IDENTITY 列のどちらの場合も同じです。この修飾子は、identityignore または identityoverride 修飾子とともに使用することはできません。
identityoverride	この修飾子を使用するのは、GENERATED ALWAYS として定義されている IDENTITY 列が、ロードされる表にある場合だけです。この修飾子はユーティリティに対し、そのような列に関して、明示的な非 NULL データを受け入れる (これらのタイプの IDENTITY 列に関する通常の規則に反する) ように指示します。これは、表を GENERATED ALWAYS と定義することが必要なときに他のデータベース・システムからデータを移行する場合や、DROPPED TABLE RECOVERY オプションを指定した ROLLFORWARD DATABASE コマンドを使用してリストアしたデータから表をロードする場合に役立ちます。この修飾子を使用した場合、IDENTITY 列でデータまたは NULL データの入っていない行はリジェクトされます (SQL3116W)。この修飾子は、identitymissing または identityignore 修飾子とともに使用することはできません。 注: このオプションが使用されていると、ロード・ユーティリティは、表の IDENTITY 列内の値の固有性の保守または検証を行いません。
indexfreespace= <i>x</i>	<i>x</i> は 0 ~ 99 の整数です。この値は各索引ページのうち、索引のロード時にフリー・スペースとして残される部分のパーセンテージとして解釈されます。ページの最初の項目は制限なしで追加されます。残りの項目は、フリー・スペースのしきい値のパーセントを維持できる限り追加されます。デフォルト値は、CREATE INDEX の実行時に使用した値です。 この値は、CREATE INDEX ステートメントに指定された PCTFREE 値よりも優先して使用され、レジストリー変数 DB2 INDEX FREE は、索引のフリー・スペースよりも優先して使用されます。索引のフリー・スペース・オプションは、索引のリーフ・ページにのみ影響を与えます。

表 220. 有効なファイル・タイプ修飾子 (LOAD) (続き)

修飾子	説明
lobsinfile	<p><i>lob-path</i> は、LOB 値を含むファイルへのパスを指定します。ASC、DEL、または IXF ロード入力ファイルには、LOB 列に LOB データを含むファイルの名前が入っています。</p> <p>ファイル・タイプが CURSOR の場合、このオプションはサポートされていません。</p> <p>各パスには、データ・ファイル内で LOB ロケーション指定子 (LLS) によって示される 1 つ以上の LOB を含む、少なくとも 1 つのファイルが含まれます。LLS は、LOB ファイル・パスに保管されるファイル内の LOB のロケーションのストリング表記です。LLS の形式は、<i>filename.ext.nnn.mmm/</i> です。<i>filename.ext</i> は LOB を含むファイルの名前、<i>nnn</i> はファイル内の LOB のオフセット (バイト単位)、<i>mmm</i> は LOB の長さ (バイト単位) を表します。たとえば、ストリング <i>db2exp.001.123.456/</i> がデータ・ファイルに保存される場合、LOB はファイル <i>db2exp.001</i> のオフセット 123 に位置し、456 バイト長です。</p> <p>NULL LOB を指定するには、サイズに -1 と入力します。サイズを 0 と指定すると、長さが 0 の LOB として扱われます。長さが -1 の NULL LOB の場合、オフセットとファイル名は無視されます。たとえば、NULL LOB の LLS は <i>db2exp.001.7.-1/</i> です。</p>
noheader	<p>ヘッダー検査コードをスキップします (単一パーティション・データベースのパーティション・グループに存在する表へのロード操作にのみ適用します)。</p> <p>オートローダー・ユーティリティーは、複数パーティションのデータベース・パーティション・グループの表にデータを提供している各ファイルに、ヘッダーを書き込みます。このヘッダーには、データベース・パーティション番号、区分化マップ、および区分化キーの指定が含まれています。ロード・ユーティリティーには、データが正しいデータベース・パーティションでロードされたことを検証するために、この情報が必要です。単一パーティションのデータベース・パーティション・グループに存在する表にファイルをロードする場合、ヘッダーは存在しないため、このオプションを指定すると、ロード・ユーティリティーはヘッダー検査コードをスキップします。</p>
norowwarnings	<p>リジェクトされた行についての警告をすべて抑制します。</p>
pagefreespace= <i>x</i>	<p><i>x</i> は 0 ~ 100 の整数です。その値は、各データ・ページ内でフリー・スペースとして残される部分のパーセンテージとして解釈されます。最小行サイズのために、指定した値が無効である場合 (たとえば、最も小さい行の大きさが 3000 バイトで、<i>x</i> の値が 50 である場合)、その行は新しいページに置かれます。値 100 が指定された場合には、各行がそれぞれ新しいページに置かれます。</p> <p>注: 表の PCTFREE 値は、ページごとに指定されたフリー・スペースの量を決定します。ロード操作の pagefreespace 値、または表の PCTFREE 値が設定されていない場合、ユーティリティーは、それぞれのページにできるだけ大きなスペースを割り当てます。pagefreespace によって設定された値は、その表について指定された PCTFREE 値をオーバーライドします。</p>

表 220. 有効なファイル・タイプ修飾子 (LOAD) (続き)

修飾子	説明
totalfreespace=x	<p>x は 0 以上の整数です。この値は表内の合計ページのうち、表の終わりにフリー・スペースとして追加される部分のパーセンテージとして解釈されます。たとえば、x が 20 で、データのロード後に、表に 100 のデータ・ページがある場合、20 の追加の空ページが付加されます。この表のデータ・ページの合計数は、120 になります。データ・ページの総数は、表の索引ページの数には影響を与えません。このオプションは、索引オブジェクトには影響を与えません。</p> <p>注: このオプションを指定して 2 つのロードが実行された場合、2 番目のロードは、最初のロードで末尾に付加された余分なスペースを再使用しません。</p>
usedefaults	<p>ターゲット表の列に対応するソース列が指定されているが、1 つ以上の行インスタンスのデータが入っていない場合、デフォルト値がロードされます。欠落データの例は、以下のとおりです。</p> <ul style="list-style-type: none"> • DEL ファイルの場合、列に「.,」が指定されています。 • DEL/ASC/WSF ファイルの場合、列が不足している行、または元の指定では十分な長さのない行。 <p>このオプションが指定されていない場合、行インスタンスのソース列にデータがないと、以下のいずれかの処理が行われます。</p> <ul style="list-style-type: none"> • 列が NULL 可能な場合、NULL がロードされます。 • 列が NULL 不可能の場合、ユーティリティはその行をリジェクトします。
ASCII ファイル形式 (ASC/DEL)	
codepage=x	<p>x は ASCII 文字ストリングです。その値は、入力データ・セット内のデータのコード・ページとして解釈されます。ロード操作時に、文字データ (および文字に指定された数値データ) は、このコード・ページからデータベースのコード・ページへ変換されます。</p> <p>以下の規則が適用されます。</p> <ul style="list-style-type: none"> • 純粋な DBCS (グラフィック)、混合 DBCS、および EUC の場合、区切り文字は $x00 \sim x3F$ の範囲に制限されます。 • EBCDIC コード・ページで指定された DEL データの場合、区切り文字はシフトインおよびシフトアウト DBCS 文字と一致しない場合があります。 • nullindchar には、コード・ポイントが $x20 \sim x7F$ の標準 ASCII セットに含まれる記号を指定する必要があります。これは、ASCII シンボルおよびコード・ポイントを示します。EBCDIC データでは、コード・ポイントが異なるとしても、対応するシンボルを使用できます。 <p>ファイル・タイプが CURSOR の場合、このオプションはサポートされていません。</p>

表 220. 有効なファイル・タイプ修飾子 (LOAD) (続き)

修飾子	説明
dateformat="x"	<p>x はソース・ファイルの日付の形式です。 ^a 有効な日付エレメントは以下のとおりです。</p> <p>YYYY - 年 (0000 ~ 9999 の範囲の 4 桁の数字) M - 月 (1 ~ 12 の範囲の 1 桁または 2 桁の数) MM - 月 (1 ~ 12 の範囲の 2 桁の数。 M とは相互排他的) D - 日 (1 ~ 31 の範囲の 1 桁または 2 桁の数) DD - 日 (1 ~ 31 の範囲の 2 桁の数。 D とは相互排他的) DDD - 日 (年間) (001 ~ 366 の範囲の 3 桁の数。他の日または月エレメントとは相互排他的)</p> <p>デフォルト値の 1 が、指定されない各エレメントに割り当てられます。日付形式の例を以下に示します。</p> <p>"D-M-YYYY" "MM.DD.YYYY" "YYYYDDD"</p>
dumpfile = x	<p>x は、リジェクトされた行が書き込まれる先の例外ファイルの (サーバー・ノードによる) 完全修飾名です。 1 レコード当たり、最大で 32KB のデータが書き込まれます。以下に、ダンプ・ファイルの指定方法の例を示します。</p> <pre>db2 load from data of del modified by dumpfile = /u/user/filename insert into table_name</pre> <p>注:</p> <ol style="list-style-type: none"> パーティション・データベース環境では、並行して実行されるロード操作が同じファイルへ書き込まないように、パスはロード元のノードに対してローカルでなければなりません。 ファイルの内容は、非同期バッファ・モードでディスクに書き込まれます。ロード操作が失敗したり割り込まれる場合には、ディスクにコミットされるレコード数が正確に知られておらず、LOAD RESTART 後の整合性が保証されていない可能性があります。ファイルが完全であるとされるのは、1 回のパスの中で開始して完了するロード操作の場合だけです。 この修飾子では、ファイル拡張子が複数のファイル名はサポートされません。たとえば、以下のようにできます。 <pre>dumpfile = /home/svtdbm6/DUMP.FILE</pre> <p>ロード・ユーティリティーでは、上のファイル名は受け入れられますが、</p> <pre>dumpfile = /home/svtdbm6/DUMP.LOAD.FILE</pre> <p>このファイル名は受け入れられません。</p>
implieddecimal	<p>暗黙指定されている小数点のロケーションが列定義によって決定され、値の終わりにあるとは見なされなくなります。たとえば、値 12345 は、12345.00 ではなく、123.45 として DECIMAL(8,2) 列にロードされます。</p>

表 220. 有効なファイル・タイプ修飾子 (LOAD) (続き)

修飾子	説明
timeformat="x"	<p>x はソース・ファイルの時刻の形式です。 ^a 有効な時刻エレメントは以下のとおりです。</p> <p>H - 時 (12 時間制の場合は 0 ~ 12 の範囲の 1 桁または 2 桁の数。 24 時間制の場合は 0 ~ 24 の範囲の 1 桁または 2 桁の数。 1 桁または 2 桁の数)</p> <p>HH - 時 (12 時間制の場合は 0 ~ 12 の範囲の 2 桁の数。 24 時間制の場合は 0 ~ 24 の範囲の 1 桁または 2 桁の数。 2 桁の数。 H とは相互排他的)</p> <p>M - 分 (0 ~ 59 の範囲の 1 桁または 2 桁の数)</p> <p>MM - 分 (0 ~ 59 の範囲の 2 桁の数。 M とは相互排他的)</p> <p>S - 秒 (0 ~ 59 の範囲の 1 桁または 2 桁の数)</p> <p>SS - 秒 (0 ~ 59 の範囲の 2 桁の数。 S とは相互排他的)</p> <p>SSSSS - 夜の 12 時から数えた秒数 (00000 ~ 86399 の 5 桁の数。 他の時刻エレメントとは相互排他的)</p> <p>TT - 午前午後の標識 (AM または PM)</p> <p>デフォルト値の 0 が、指定されない各エレメントに割り当てられます。時刻形式の例を以下に示します。</p> <p>"HH:MM:SS" "HH.MM TT" "SSSSS"</p>

表 220. 有効なファイル・タイプ修飾子 (LOAD) (続き)

修飾子	説明
timestampformat="x"	<p>x はソース・ファイルのタイム・スタンプの形式です。 ^a 有効なタイム・スタンプ・エレメントは以下のとおりです。</p> <p>YYYY - 年 (0000 ~ 9999 の範囲の 4 桁の数字) M - 月 (1 ~ 12 の範囲の 1 桁または 2 桁の数) MM - 月 (1 ~ 12 の範囲の 2 桁の数。 M (月) とは相互排他的) D - 日 (1 ~ 31 の範囲の 1 桁または 2 桁の数) DD - 日 (1 ~ 31 の範囲の 2 桁の数。 D とは相互排他的) DDD - 日 (年間) (001 ~ 366 の範囲の 3 桁の数。 他の日または月エレメント とは相互排他的) H - 時 (12 時間制の場合は 0 ~ 12 の範囲の 1 桁または 2 桁の数。 24 時間制の場合は 0 ~ 24 の範囲の 1 桁または 2 桁の数。 1 桁または 2 桁の数) HH - 時 (12 時間制の場合は 0 ~ 12 の範囲の 2 桁の数。 24 時間制の場合は 0 ~ 24 の範囲の 1 桁または 2 桁の数。 2 桁の数。 H とは相互排他的) M - 分 (0 ~ 59 の範囲の 1 桁または 2 桁の数) MM - 分 (0 ~ 59 の範囲の 2 桁の数。 M (分) とは相互排他的) S - 秒 (0 ~ 59 の範囲の 1 桁または 2 桁の数) SS - 秒 (0 ~ 59 の範囲の 2 桁の数。 S とは相互排他的) SSSS - 夜の 12 時から数えた秒数 (00000 ~ 86399 の 5 桁の数。 他の時刻エレメントとは相互排他的) UUUUUU - マイクロ秒 (000000 ~ 9999996 の範囲の 6 桁の数) TT - 午前午後の標識 (AM または PM)</p> <p>YYYY、M、MM、D、DD、または DDD エレメントが指定されていない場合、デフォルト値の 1 が割り当てられます。他のエレメントが指定されていない場合には、デフォルト値の 0 が割り当てられます。以下に、タイム・スタンプ形式の例を示します。</p> <p>"YYYY/MM/DD HH:MM:SS.UUUUUU"</p> <p>以下の例では、ユーザー定義の日時形式を含むデータを、schedule という表にインポートする方法を示しています。</p> <pre>db2 import from delfile2 of del modified by timestampformat="yyyy.mm.dd hh:mm tt" insert into schedule</pre>
noeofchar	<p>オプションのファイル終了文字 x'1A' は、ファイルの終わりとして認識されません。それが普通の文字であるかのように処理は継続されます。</p>
ASC (区切りなし ASCII) ファイル形式	

表 220. 有効なファイル・タイプ修飾子 (LOAD) (続き)

修飾子	説明
binarynumerics	<p>数値 (DECIMAL 以外) データは、文字表記ではなく、バイナリー形式でなければなりません。これにより、コストのかかる変換を避けられます。</p> <p>このオプションがサポートされるのは、定位置 ASC において、<code>reclen</code> オプションによって固定長レコードが指定されている場合だけです。<code>noeofchar</code> オプションが前提です。</p> <p>以下の規則が適用されます。</p> <ul style="list-style-type: none"> • <code>BIGINT</code>、<code>INTEGER</code>、および <code>SMALLINT</code> を除き、データ・タイプ間の変換は実行されません。 • データ長は、それぞれのターゲット列定義と一致している必要があります。 • <code>FLOAT</code> は、IEEE 浮動小数点形式でなければなりません。 • ロード・ソース・ファイルに含まれるバイナリー・データは、ロード操作を実行するプラットフォームに関係なく、ビッグ・エンディアンであると見なされます。 <p>注: この修飾子の影響を受ける列のデータに、<code>NULL</code> を指定することはできません。この修飾子を使用すると、ブランク (通常は <code>NULL</code> と解釈される) は、バイナリー値であると解釈されます。</p>
nochecklengths	<p><code>nochecklengths</code> が指定した場合されていると、ソース・データの列定義がターゲット表の列のサイズを超えるものであっても、各行のロードが試行されます。このような行が正常にロードされるのは、コード・ページ変換でソース・データが縮小する場合です。たとえば、ソースにある 4 バイトの EUC データがターゲットで 2 バイトの DBCS データに縮小すれば、必要スペースは半分になります。このオプションが特に役立つのは、列の定義は不一致であるがソース・データが常に適合することが分かっている場合です。</p>
nullindchar=x	<p><code>x</code> は単一文字です。 <code>NULL</code> 値を示す文字を <code>x</code> に変更します。 <code>x</code> のデフォルト値は <code>Y</code> です。^b</p> <p>文字が 1 つの英字である場合を除いて、この修飾子は、EBCDIC データ・ファイルで大文字小文字を区別します。たとえば、<code>NULL</code> 標識文字が文字 <code>N</code> として指定されている場合、<code>n</code> も <code>NULL</code> 標識として認識されます。</p>

表 220. 有効なファイル・タイプ修飾子 (LOAD) (続き)

修飾子	説明
packeddecimal	<p>binarynumerics 修飾子は DECIMAL フィールド・タイプを含まないため、バック 10 進数データを直接ロードします。</p> <p>このオプションがサポートされるのは、定位置 ASC において、reclen オプションによって固定長レコードが指定されている場合だけです。noeofchar オプションが前提です。</p> <p>ニブル符号用にサポートされる値は以下のとおりです。</p> <pre> + = 0xC 0xA 0xE 0xF - = 0xD 0xB </pre> <p>注: この修飾子の影響を受ける列のデータに、NULL を指定することはできません。この修飾子を使用すると、ブランク (通常は NULL と解釈される) は、バイナリー値であると解釈されます。</p> <p>サーバー・プラットフォームに関係なく、ロード・ソース・ファイルの バイナリー・データのバイト配列はビッグ・エンディアンであると見なされます。つまり、Windows オペレーティング・システムでこの修飾子を使用する場合、バイト順序を逆転させてはなりません。</p>
reclen=x	<p>x は 32 767 以下の整数です。各行ごとに x 個の文字が読み取られ、行の終わりを示すのに改行文字は使用されません。</p>
striptblanks	<p>データを可変長フィールドにロードする際に、後書きブランク・スペースを切り捨てます。このオプションを指定しない場合、ブランク・スペースはそのまま保持されます。</p> <p>このオプションは、striptnulls と一緒には指定できません。これらのオプションは、相互に排他的です。</p> <p>注: このオプションは以前の t オプションに代わるもので、バックレベルの互換性のためにのみサポートされています。</p>
striptnulls	<p>データを可変長フィールドにロードする際に、後書き NULL (0x00 文字) を切り捨てます。このオプションを指定しない場合、NULL はそのまま保持されます。</p> <p>このオプションは、striptblanks と一緒には指定できません。これらのオプションは、相互に排他的です。</p> <p>注: このオプションは以前の padwithzero オプションに代わるもので、バックレベルの互換性のためにのみサポートされています。</p>
zoneddecimal	<p>BINARYNUMERICS 修飾子は DECIMAL フィールド・タイプを含まないため、ゾーン 10 進数データをロードします。このオプションがサポートされるのは、定位置 ASC において、RECLEN オプションによって固定長レコードが指定されている場合だけです。NOEOFCHAR オプションが前提です。</p> <p>ハーフバイト符号値は、以下のいずれかです。</p> <pre> + = 0xC 0xA 0xE 0xF - = 0xD 0xB </pre> <p>サポートされている数値は、0x0 ~ 0x9 です。</p> <p>サポートされているゾーン値は、0x3 および 0xF です。</p>
DEL (区切り付き ASCII) ファイル形式	

表 220. 有効なファイル・タイプ修飾子 (LOAD) (続き)

修飾子	説明
chardelx	<p><i>x</i> は単一文字の文字列区切り文字です。デフォルト値は二重引用符 (") です。指定した文字は、二重引用符の代わりに、文字列を囲む文字として使用されます。 ^{bc}</p> <p>単一引用符 (') も、以下のように文字列の区切り文字として指定できます。</p> <pre>modified by chardel''</pre> <p>二重引用符 (") を、文字列区切り文字として明示的に指定したい場合、以下のように指定してください。</p> <pre>modified by chardel""</pre>
coldelx	<p><i>x</i> は単一文字の列区切り文字です。デフォルト値はコンマ (,) です。指定した文字は、列の終わりを表すために、コンマの代わりに使用されます。 ^{bc}</p>
datesiso	日付形式。すべての日付データ値を ISO 形式でロードします。
decplusblank	正符号文字。正の 10 進値の先頭に正符号 (+) ではなく、ブランク・スペースが置かれます。デフォルトのアクションでは、10 進値に正符号の接頭部が付けられます。
decptx	<p><i>x</i> は小数点としてピリオドと置換される単一文字です。デフォルト値はピリオド (.) です。指定した文字は、小数点文字としてピリオドの代わりに使用されます。 ^{bc}</p>
delprioritychar	<p>区切り文字の現在のデフォルト優先順位は、(1) レコード区切り文字、(2) 文字区切り文字、(3) 列区切り文字です。この修飾子を使用すると、区切り文字の優先順位が (1) 文字区切り文字、(2) レコード区切り文字、(3) 列区切り文字に戻り、以前の優先順位に依存している既存のアプリケーションが保護されます。構文は以下のとおりです。</p> <pre>db2 load ... modified by delprioritychar ...</pre> <p>たとえば、以下のような DEL データ・ファイルがあるとします。</p> <pre>"Smith, Joshua",4000,34.98<row delimiter> "Vincent,<row delimiter>, is a manager", 4005,44.37<row delimiter></pre> <p>delprioritychar 修飾子が指定されている場合、このデータ・ファイルには 2 行しかありません。2 番目の <row delimiter> は 2 番目の行の最初のデータ列の一部と解釈されますが、1 番目と 3 番目の <row delimiter> は実レコードの区切り文字と解釈されます。この修飾子が指定されていないと、このデータ・ファイルは 3 行になり、各行は <row delimiter> によって区切られます。</p>
dldelx	<p><i>x</i> は単一文字の DATALINK 区切り文字です。デフォルト値はセミコロン (;) です。指定した文字はセミコロン (;) の代わりに、DATALINK 値のフィールド間区切り文字として使用されます。DATALINK 値には副値が複数個含まれる場合があるため、これが必要になります。 ^{bcd}</p> <p>注: <i>x</i> は、行、列、または文字列区切り文字とは異なる文字にしてください。</p>

表 220. 有効なファイル・タイプ修飾子 (LOAD) (続き)

修飾子	説明
keepblanks	<p>タイプが CHAR、VARCHAR、LONG VARCHAR、または CLOB の各フィールドの前後のブランクを保持します。このオプションが指定されていない場合、文字区切り文字の外側にある前後のブランクはすべて除去され、表のすべてのブランク・フィールドに NULL が挿入されます。</p> <p>以下の例では、データ・ファイル内の前後のスペースをすべて保持しながら、TABLE1 という表にデータをロードする方法を示しています。</p> <pre>db2 load from delfile3 of del modified by keepblanks insert into table1</pre>
nodoubledel	<p>二重になっている区切り文字の認識を抑止します。詳しくは、『API のインポート』で説明されている『区切り文字についての制約事項』を参照してください。</p>
IXF ファイル形式	
forcein	<p>コード・ページが不一致でもデータを受け入れ、コード・ページ間の変換を抑止するようにユーティリティに指示します。</p> <p>固定長ターゲット・フィールドは、データが入るだけの十分な大きさがあるかどうかチェックされます。nochecklengths が指定されていると、チェックはされず、各行をロードしようとします。</p>
nochecklengths	<p>nochecklengths が指定した場合されていると、ソース・データの列定義がターゲット表の列のサイズを超えるものであっても、各行のロードが試行されます。このような行が正常にロードされるのは、コード・ページ変換でソース・データが縮小する場合です。たとえば、ソースにある 4 バイトの EUC データがターゲットで 2 バイトの DBCS データに縮小すれば、必要スペースは半分になります。このオプションが特に役立つのは、列の定義は不一致であるがソース・データが常に適合することが分かっている場合です。</p>

表 220. 有効なファイル・タイプ修飾子 (LOAD) (続き)

修飾子	説明
	<p>注:</p> <ol style="list-style-type: none"> サポートされていないファイル・タイプを MODIFIED BY オプションで使用しようとしても、ロード・ユーティリティは警告を出しません。この場合、ロード操作が失敗し、エラー・コードが戻されます。 ^a 日付形式ストリングは必ず二重引用符で囲まなければなりません。フィールド区切り文字には、a ~ z、A ~ Z、および 0 ~ 9 を含めることはできません。フィールド区切り文字は、DEL ファイル形式の文字区切り文字またはフィールド区切り文字と同じにすることはできません。フィールド区切り文字は、エレメントの開始および終了位置が明確な場合は任意指定です。(修飾子によって) D、H、M、または S などのエレメントが使用される場合、項目が可変長であるためにあいまいさが存在することがあります。 タイム・スタンプ形式の場合、月の記述子と分の記述子のどちらも文字 M を使用するため、区別が明白であるように注意する必要があります。月のフィールドは、他の日付フィールドと隣接していなければなりません。分フィールドは、他の時刻フィールドに隣接していなければなりません。以下に、いくつかのあいまいなタイム・スタンプ形式を示します。 " M " (月または分のどちらにもとれる) " M : M " (月と分が区別がつかない) " M : Y Y Y Y : M " (両方とも月と解釈される) " S : M : Y Y Y Y " (時刻値と日付値の両方に隣接している) あいまいな形式になっている場合、ユーティリティはエラー・メッセージを報告し、操作は失敗します。 以下に、明確なタイム・スタンプ形式を示します。 " M : Y Y Y Y " (M (月)) " S : M " (M (分)) " M : Y Y Y Y : S : M " (M (月) M (分)) " M : H : Y Y Y Y : M : D " (M (分) M (月)) <p>注: 二重引用符や円記号などの文字の前には、エスケープ文字 (たとえば、¥) を付けなければなりません。</p> ^b この文字は、ソース・データのコード・ページで指定してください。 文字コード・ポイント (文字記号ではない) は、 xJJ または 0xJJ という構文で指定することができます (JJ はコード・ポイントの 16 進表記)。たとえば、列区切りとして # 文字を指定するには、以下のいずれかを使用します。 . . . modified by colde1# modified by colde10x23 modified by colde1X23 . . . ^c 『API のインポート』で説明されている『区切り文字についての制約事項』には、代替区切り文字として使用できる文字に適用される制限事項がリストされています。 ^d DATALINK 区切り文字が URL 構文内で有効な文字である場合でも、ロード操作の有効範囲内ではその特別な意味はなくなります。

関連資料:

- 管理 API リファレンスの『sqluvqdp - 表の表スペースの静止』
- データ移動ユーティリティ 手引きおよび解説書の『db2LoadQuery - ロードの照会』
- 管理 API リファレンスの『SQLU-MEDIA-LIST』
- データ移動ユーティリティ 手引きおよび解説書の『sqluxpr - エクスポート』

- データ移動ユーティリティー 手引きおよび解説書の『sqlimpr - インポート』
- 管理 API リファレンスの『db2DatabaseQuiesce - データベースの静止』
- 管理 API リファレンスの『db2InstanceQuiesce - インスタンスの静止』

関連サンプル:

- 『dtformat.sqc -- Load and import data format extensions (C)』
- 『tbload.sqc -- How to load into a partitioned database (C)』
- 『tbmove.sqc -- How to move table data (C)』
- 『tbmove.sqC -- How to move table data (C++)』

db2Runstats - 統計の実行

表または関連する索引の両方または一方の特性についての統計を更新します。これらの特性には、レコード数、ページ数、レコードの平均長などがあります。最適化マイザーは、データへのアクセス・パスを判別する際にこれらの統計を使用します。

表に数多くの更新が加えられたときや、表を再編成した後、あるいは新規の索引を作成した後などに、このユーティリティーを呼び出してください。

統計は、API が実行されるデータベース・パーティションに存在する表パーティションに基づいて収集されます。グローバル表統計は、あるデータベース・パーティションで取得された値に、表が完全に保管されているデータベース・パーティションの数を掛けることによって導出されます。グローバル統計は、カタログ表に保管されます。

API が呼び出されるデータベース・パーティションは、表のパーティションを含んでいない必要はありません。

- 表のパーティションを含むデータベース・パーティションから API が呼び出されると、ユーティリティーはこのデータベース・パーティションで実行されます。
- 表パーティションを含まないデータベース・パーティションから API が呼び出されると、要求は、表パーティションを保持しているデータベース・パーティション・グループ内の最初のデータベース・パーティションに送られます。その後、このデータベース・パーティションでユーティリティーが実行されます。

有効範囲:

この API は、db2nodes.cfg ファイル内の任意のデータベース・パーティション・サーバーから呼び出すことができます。カタログ・データベース・パーティションのカタログを更新するのに使用します。

許可:

以下のいずれかです。

- *sysadm*
- *sysctrl*

db2Runstats - 統計の実行

- *sysmaint*
- 表に対する CONTROL 特権
- LOAD

必要な接続:

データベース

API 組み込みファイル:

db2ApiDf.h

C API 構文:

```
/* File: db2ApiDf.h */
/* API: db2Runstats */
/* ... */
SQL_API_RC SQL_API_FN
db2Runstats (
    db2Uint32 versionNumber,
    db2RunstatsData *data,
    struct sqlca *sqlca);

typedef SQL_STRUCTURE db2RunstatsData
{
    double                iSamplingOption;
    unsigned char        *piTablename;
    db2ColumnData        **piColumnList;
    db2ColumnDistData    **piColumnDistributionList;
    db2ColumnGrpData     **piColumnGroupList;
    unsigned char        **piIndexList;
    db2Uint32            iRunstatsFlags;
    db2int16             iNumColumns;
    db2int16             iNumColDist;
    db2int16             iNumColGroups;
    db2int16             iNumIndexes;
    db2int16             iParallelismOption;
    db2int16             iTableDefaultFreqValues;
    db2int16             iTableDefaultQuantiles;
} db2RunstatsData;

typedef SQL_STRUCTURE db2ColumnData
{
    unsigned char        *piColumnName;
    db2int16             iColumnFlags;
} db2ColumnData;

typedef SQL_STRUCTURE db2ColumnDistData
{
    unsigned char        *piColumnName;
    db2int16             iNumFreqValues;
    db2int16             iNumQuantiles;
} db2ColumnDistData;

typedef SQL_STRUCTURE db2ColumnGrpData
```

```

{
    unsigned char          **piGroupColumnNames;
    db2int16              iGroupSize;
    db2int16              iNumFreqValues;
    db2int16              iNumQuantiles;
} db2ColumnGrpData;
/* ... */

```

汎用 API 構文:

```

/* File: db2ApiDf.h */
/* API: db2gRunstats */
/* ... */
SQL_API_RC SQL_API_FN
db2gRunstats (
    db2UInt32 versionNumber,
    db2gRunstatsData *data,
    struct sqlca *sqlca);

```

```

typedef SQL_STRUCTURE db2gRunstatsData
{
    double                iSamplingOption;
    unsigned char         *piTablename;
    db2gColumnData        **piColumnList;
    db2gColumnDistData    **piColumnDistributionList;
    db2gColumnGrpData     **piColumnGroupList;
    unsigned char         **piIndexList;
    db2UInt16             *piIndexNamesLen;
    db2UInt32             iRunstatsFlags;
    db2UInt16             iTablenameLen;
    db2int16              iNumColumns;
    db2int16              iNumColDist;
    db2int16              iNumColGroups;
    db2int16              iNumIndexes;
    db2int16              iParallelismOption;
    db2int16              iTableDefaultFreqValues;
    db2int16              iTableDefaultQuantiles;
} db2gRunstatsData;

```

```

typedef SQL_STRUCTURE db2gColumnData
{
    unsigned char         *piColumnName;
    db2UInt16             iColumnNameLen;
    db2int16              iColumnFlags;
} db2gColumnData;

```

```

typedef SQL_STRUCTURE db2gColumnDistData
{
    unsigned char         *piColumnName;
    db2UInt16             iColumnNameLen;
    db2int16              iNumFreqValues;
    db2int16              iNumQuantiles;
} db2gColumnDistData;

```

```

typedef SQL_STRUCTURE db2gColumnGrpData

```

db2Runstats - 統計の実行

```
{
    unsigned char          **piGroupColumnNames;
    db2Uint16             *piGroupColumnNamesLen;
    db2int16              iGroupSize;
    db2int16              iNumFreqValues;
    db2int16              iNumQuantiles;
} db2gColumnGrpData;
/* ... */
```

API パラメーター:

versionNumber

入力。 2 番目のパラメーター *data* として渡される構造のバージョンとリリースのレベルを指定します。

data 入力。 *db2RunstatsData* 構造を指すポインター。

sqlca 出力。 *sqlca* 構造を指すポインター。

iSamplingOption

入力。将来の使用のために予約されています。有効な値は 0 または 100 です。

piTablename

入力。統計が収集される表の完全修飾名を指すポインター。名前は別名にすることができます。行タイプの場合、*piTablename* は階層のルート表の名前でなければなりません。

piColumnList

入力。 *db2ColumnData* エレメントの配列。この配列の各エレメントは、以下の 2 つのサブエレメントで構成されています。

- 統計が収集される列の名前を表すストリング。
- 列の統計オプションを示すフラグ・フィールド。

iNumColumns がゼロの場合、*piColumnList* は無視されます (提供されている場合)。

piColumnDistributionList

入力。 *db2ColumnDistData* エレメントの配列。これらのエレメントは、特定の列 (複数の場合もある) の分散統計の収集が必要な場合に提供されます。この配列の各エレメントは、以下の 3 つのサブエレメントで構成されています。

- 分散統計が収集される列の名前を表すストリング。
- 収集する頻度 (数値)。
- 収集する変位値。

piColumnList に現れない *piColumnDistributionList* 内に現れる列には、列に収集された基本的な列統計があります。これは、最初から *piColumnList* 内にこれらの列が組み込まれているのと同じ効果があります。 *iNumColdist* がゼロの場合、*piColumnDistributionList* は無視されます。

piColumnGroupList

入力。 db2ColumnGrpData エレメントの配列。これらのエレメントは、列のグループで列統計を収集する場合に提供されます。つまり、各行に対するグループの各列の値は連結され、単一の値として処理されます。

各 db2ColumnGrpData は 3 つの整数フィールドとストリングの配列で構成されます。最初の整数フィールドは、ストリング piGroupColumns の配列内のストリング数を表します。この配列内の各ストリングには、1 つの列名が入っています。たとえば、列の組み合わせ統計が列グループ (c1、c2) および (c3、c4、c5) で収集される場合、 piGroupColumns には 2 つの db2ColumnGrpData エレメントが存在します。

最初の db2ColumnGrpData エレメントでは、 piGroupSize = 2 およびストリングの配列には、2 つのエレメントとして c1 と c2 が入っています。

2 番目の db2ColumnGrpData エレメントでは、 piGroupSize = 3 およびストリングの配列には、3 つのエレメントとして c3、c4、および c5 が入っています。

2 番目および 3 番目の整数フィールドは、列グループに関して分散統計を収集するときの、頻度および変位値の数値をそれぞれ表します。これは、現在サポートされていません。

piColumnList に現れない piColumnGroupList 内に現れる列には、列に収集された基本的な列統計があります。これは、最初から piColumnList 内にこれらの列が組み込まれているのと同じ効果があります。 iNumColGroups がゼロの場合、 piColumnGroupList は無視されます。

piIndexList

入力。ストリングの配列。それぞれのストリングには、完全修飾された索引名が 1 つ含まれます。 NumIndexes がゼロの場合、 piIndexList は無視されます。

piIndexNamesLen

入力。索引リストにある索引名のそれぞれの長さを示す値の配列 (バイト単位)。 NumIndexes がゼロの場合、 piIndexNamesLen は無視されます。

iRunstatsFlags

入力。統計オプションを指定するために使用されるビット・マスク・フィールド。有効な値は以下のとおりです。

DB2RUNSTATS_ALL_COLUMNS

表のすべての列で統計を収集します。このオプションは、列、列分布、列グループ、または索引構造リストの組み合わせで指定することができます。これは、表のすべての列で統計を収集したいが、特定の列に対して統計オプションを提供したい場合に役立ちます。

DB2RUNSTATS_KEY_COLUMNS

表で定義されたすべての索引を構成する列でのみ統計を収集します。

このオプションは、列、列分布、列グループ、または索引構造リストの組み合わせで指定することができます。これは、表のすべてのキー列で統計を収集したいが、非キー列でも統計を収集したい場合、または特定のキー列の統計オプションを提供したい場合に役立ちます。

DB2RUNSTATS_DISTRIBUTION

分散統計を収集します。このオプションは、DB2RUNSTATS_ALL_COLUMNS および DB2RUNSTATS_KEY_COLUMNS とともにのみ使用できます。DB2RUNSTATS_ALL_COLUMNS とともに使用される場合、分散統計は表のすべての列に対して収集されます。DB2RUNSTATS_KEY_COLUMNS とともに使用される場合、分散統計は表に定義されたすべての索引を構成するすべての列に対して収集されます。DB2RUNSTATS_ALL_COLUMNS および DB2RUNSTATS_KEY_COLUMNS で使用される場合、基本統計は表のすべて列について収集され、分散統計は表で定義されたすべての索引を構成する列についてのみ収集されます。

DB2RUNSTATS_ALL_INDEXES

表で定義されたすべての索引で統計を収集します。

DB2RUNSTATS_EXT_INDEX

詳細な索引統計を収集します。このオプションは、DB2RUNSTATS_ALL_INDEXES または索引名 (*piIndexList* および *iNumIndexes* > 0) の明示的なリストとともに指定する必要があります。

DB2RUNSTATS_EXT_INDEX_SAMPLED

抽出方式を使用して詳細な索引統計を収集します。このオプションは、DB2RUNSTATS_ALL_INDEXES または索引名 (*piIndexList* および *iNumIndexes* > 0) の明示的なリストとともに指定する必要があります。DB2RUNSTATS_EXT_INDEX は、同時に指定された場合は無視されます。

DB2RUNSTATS_ALLOW_READ

統計の収集中に、他のユーザーが読み取り専用アクセスを行えるようになります。デフォルトでは、読み取りアクセスおよび書き込みアクセスが許可されます。

iTablenameLen

入力。表名の長さを示す値 (バイト単位)。

iNumColumns

入力。 *piColumnList* リストで指定された項目数。

iNumColdist

入力。 *piColumnDistributionList* リストで指定された項目数。

iNumColGroups

入力。 *piColumnGroupList* リストで指定された項目数。

iNumIndexes

入力。 *piIndexList* リストで指定された項目数。

iParallelismOption

入力。将来の使用のために予約されています。有効な値は 0 です。

iTableDefaultFreqValues

入力。表について収集する頻度のデフォルトの回数を指定します。有効な値は以下のとおりです。

- n** 列レベルで他の値が指定されていない場合、n の頻度で収集されます。
- 0** 列レベルで他の値が指定されていない場合、収集の頻度はありません。
- 1** 収集する頻度に、デフォルトのデータベース構成パラメーター NUM_FREQVALUES を使用します。

iTableDefaultQuantiles

入力。表について収集する変位値のデフォルトの数値を指定します。有効な値は以下のとおりです。

- n** 列レベルで他の値が指定されていない場合、n 個の変位値が収集されます。
- 0** 列レベルで他の値が指定されない場合、変位値は収集されません。
- 1** 収集する変位値の数について、デフォルトのデータベース構成パラメーター NUM_QUANTILES を使用します。

piColumnName

入力。列名を表すストリングを指すポインター。

iColumnNameLen

入力。列名の長さを示す値 (バイト単位)。

iColumnFlags

入力。列の統計オプションを指定するために使用されるビット・マスク・フィールド。有効な値は以下のとおりです。

DB2RUNSTATS_COLUMN_LIKE_STATS

列で LIKE 統計を収集します。

iNumFreqValues

入力。列で収集する頻度。有効な値は以下のとおりです。

- n** 列で n の頻度で収集します。
- 1** 表の頻度のデフォルト値を使用します。たとえばこれには、設定され

db2Runstats - 統計の実行

ている場合は *iTableDefaultFreqValues* や、またはデータベース構成パラメーター *NUM_FREQVALUES* があります。

iNumQuantiles

入力。列に関して収集する変位値の数。有効な値は以下のとおりです。

- n** 列に関して **n** 個の変位値を収集します。
- 1** 変位値の表デフォルト数を使用します。たとえばこれには、設定されている場合は *iTableDefaultQuantiles* や、またはデータベース構成パラメーター *NUM_QUANTILES* があります。

piGroupColumnNames

入力。stringの配列。各stringは、統計が収集される列グループの一部である列名を表します。

piGroupColumnNamesLen

入力。列名リストにある列名のそれぞれの長さを示す値の配列 (バイト単位)。

iGroupSize

入力。列グループ内の列の数。有効な値は以下のとおりです。

- n** 列グループは **n** 列で構成されています。

iNumFreqValues

入力。将来の使用のために予約されています。

iNumQuantiles

入力。将来の使用のために予約されています。

使用上の注意:

RUNSTATS は、以下のような場合に統計を更新するために使用してください。

- 表が何回も修正されている場合 (たとえば、数多くの更新が行われている場合や、大量のデータが挿入または削除されている場合など)
- 表が再編成されている場合
- 新しい索引が作成された場合

統計が更新された後、*sqlabndx* - バインドを使用してパッケージを再バインドすることによって、表への新しいアクセス・パスを作成することができます。

索引統計を要求したときに、索引を含む表についての統計がそれまで実行されていなかった場合、表と索引の両方に関する統計が計算されます。

db2Runstats API が索引でのみ統計を収集している場合は、以前に収集された分散統計は保存されます。そうでない場合は、API は以前に収集された分散統計をドロップします。

この API を呼び出した後、アプリケーションは COMMIT を発行して、ロックを解除する必要があります。

新しいアクセス・プランが生成されるようにするには、この API を呼び出した後、ターゲット表を参照するパッケージを再バインドする必要があります。

表に対してのみこの API を実行すると、結果として表レベルの統計が、既存の索引レベルの統計と不整合な状況になる場合があります。たとえば、索引レベルの統計が特定の表で収集され、後にかんりの行数がこの表から削除された場合、表に対してのみこの API を発行すると、不整合状態の FIRSTKEYCARD よりも小さい表カーディナリティーという結果に終わる場合があります。それと同様に、索引に対してのみこの API を発行する場合、既存の表レベル統計が整合性がない状態のままになることがあります。たとえば、表レベルの統計が特定の表で収集され、後でかんりの行数がこの表から削除された場合、索引に対してのみ db2Runstats API を発行すると、いくつかの列が表カーディナリティーより大きい COLCARD を持つという結果に終わる場合があります。そのような不整合が検出された場合、警告が戻されます。

関連資料:

- 管理 API リファレンス の『sqlabndx - バインド』
- 管理 API リファレンス の『SQLCA』
- 1222 ページの『REORGCHK』
- 管理 API リファレンス の『db2CfgGet - 構成パラメーターの入手』
- 管理 API リファレンス の『db2Reorg - 再編成』

関連サンプル:

- 『dbstat.sqb -- Reorganize table and run statistics (MF COBOL)』
- 『tbreorg.sqc -- How to reorganize a table and update its statistics (C)』
- 『tbreorg.sqC -- How to reorganize a table and update its statistics (C++)』

db2Restore - データベースのリストア

db2Backup (データベースのバックアップ) を使用して、バックアップが取られていたデータベースが損傷または破壊された場合に、そのデータベースを再作成します。リストアされたデータベースは、バックアップ・コピーの作成時と同じ状態になります。このユーティリティーでは、(新しいデータベースへのリストアが可能であることに加えて) バックアップ・イメージ内のデータベース名とは異なる名前データベースをリストアすることが可能です。

このユーティリティーは、以前の 2 つのリリースで作成した DB2 データベースをリストアするためにも使用できます。

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

db2Restore - データベースのリストア

有効範囲:

この API は、それが呼び出されたデータベース・パーティションにのみ影響を与えません。

許可:

既存のデータベースにリストアするには、以下のいずれかが必要です。

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

新規データベースにリストアするには、以下のいずれかが必要です。

- *sysadm*
- *sysctrl*

必要な接続:

データベース (既存のデータベースにリストアする場合)。この API を呼び出せば、指定したデータベースへの接続が自動的に確立され、リストア操作が終了すると接続が解放されます。

新規のデータベースにリストアする場合、インスタンスおよびデータベース。データベースを作成するには、インスタンス・アタッチが必要です。

現行のインスタンス (DB2INSTANCE 環境変数の値で定義) とは異なるインスタンスにある新規のデータベースにリストアを行うには、まず、新規のデータベースを置くインスタンスにアタッチする必要があります。

API 組み込みファイル:

db2ApiDf.h

C API 構文:

```
/* File: db2ApiDf.h */
/* API: db2Restore */
/* ... */
SQL_API_RC SQL_API_FN
db2Restore (
    db2UInt32    versionNumber,
    void         *pDB2RestoreStruct,
    struct sqlca *pSqlca);
/* ... */

typedef SQL_STRUCTURE db2RestoreStruct
{
    char                *piSourceDBAlias;
```

```

char                *piTargetDBAlias;
char                oApplicationId[SQLU_APPLID_LEN+1];
char                *piTimestamp;
char                *piTargetDBPath;
char                *piReportFile;
struct db2TablespaceStruct *piTablespaceList;
struct db2MediaListStruct *piMediaList;
char                *piUsername;
char                *piPassword;
char                *piNewLogPath;
void                *piVendorOptions;
db2UInt32           iVendorOptionsSize;
db2UInt32           iParallelism;
db2UInt32           iBufferSize;
db2UInt32           iNumBuffers;
db2UInt32           iCallerAction;
db2UInt32           iOptions;
} db2RestoreStruct;

```

```

typedef SQL_STRUCTURE db2TablespaceStruct
{
    char                **tablespaces;
    db2UInt32           numTablespaces;
} db2TablespaceStruct;

```

```

typedef SQL_STRUCTURE db2MediaListStruct
{
    char                **locations;
    db2UInt32           numLocations;
    char                locationType;
} db2MediaListStruct;
/* ... */

```

汎用 API 構文:

```

/* File: db2ApiDf.h */
/* API: db2gRestore */
/* ... */
SQL_API_RC SQL_API_FN
db2gRestore (
    db2UInt32         versionNumber,
    void                *pDB2gRestoreStruct,
    struct sqlca *pSqlca);

typedef SQL_STRUCTURE db2gRestoreStruct
{
    char                *piSourceDBAlias;
    db2UInt32           iSourceDBAliasLen;
    char                *piTargetDBAlias;
    db2UInt32           iTargetDBAliasLen;
    char                *poApplicationId;
    db2UInt32           iApplicationIdLen;
    char                *piTimestamp;
    db2UInt32           iTimestampLen;
    char                *piTargetDBPath;
}

```

db2Restore - データベースのリストア

```
    db2UInt32          iTargetDBPathLen;
    char              *piReportFile;
    db2UInt32          iReportFileLen;
    struct db2gTablespaceStruct *piTablespaceList;
    struct db2gMediaListStruct *piMediaList;
    char              *piUsername;
    db2UInt32          iUsernameLen;
    char              *piPassword;
    db2UInt32          iPasswordLen;
    char              *piNewLogPath;
    db2UInt32          iNewLogPathLen;
    void              *piVendorOptions;
    db2UInt32          iVendorOptionsSize;
    db2UInt32          iParallelism;
    db2UInt32          iBufferSize;
    db2UInt32          iNumBuffers;
    db2UInt32          iCallerAction;
    db2UInt32          iOptions;
} db2gRestoreStruct;

typedef SQL_STRUCTURE db2gTablespaceStruct
{
    struct db2Char          *tablespaces;
    db2UInt32              numTablespaces;
} db2gTablespaceStruct;

typedef SQL_STRUCTURE db2gMediaListStruct
{
    struct db2Char          *locations;
    db2UInt32              numLocations;
    char                    locationType;
} db2gMediaListStruct;

typedef SQL_STRUCTURE db2Char
{
    char                    *pioData;
    db2UInt32              iLength;
    db2UInt32              oLength;
} db2Char;
/* ... */
```

API パラメーター:

versionNumber

入力。 2 番目のパラメーター *pDB2RestoreStruct* として渡される構造のバージョンとリリースのレベルを指定します。

pDB2RestoreStruct

入力。 *db2RestoreStruct* 構造を指すポインター。

pSqlca 出力。 *sqlca* 構造へのポインター。

piSourceDBAlias

入力。ソース・データベース・バックアップ・イメージのデータベース別名を含むストリング。

iSourceDBAliasLen

入力。ソース・データベースの別名の長さを示す 4 バイトの符号なし整数 (バイト単位)。

piTargetDBAlias

入力。ターゲット・データベースの別名を含むストリング。このパラメーターが NULL の場合、*piSourceDBAlias* が使用されます。

iTargetDBAliasLen

入力。ターゲット・データベースの別名の長さを示す 4 バイトの符号なし整数 (バイト単位)。

oApplicationId

出力。API によって、アプリケーションにサービスを提供しているエージェントを識別するストリングが戻されます。データベース・モニターを使用するバックアップ操作の進行状況に関する情報を取得することもできます。

poApplicationId

出力。長さ `SQLU_APPLID_LEN+1` (`sqlutil` で定義) のバッファーを提供します。API によって、アプリケーションにサービスを提供しているエージェントを識別するストリングが戻されます。データベース・モニターを使用するバックアップ操作の進行状況に関する情報を取得することもできます。

iApplicationIdLen

入力。 `poApplicationId` バッファーの長さを示す 4 バイトの符号なし整数 (バイト単位)。 `SQLU_APPLID_LEN+1` (`sqlutil` に定義) と等しくなければなりません。

piTimestamp

入力。バックアップ・イメージのタイム・スタンプを示すストリング。指定したソース内にバックアップ・イメージが 1 つしかない場合、このフィールドはオプションです。

iTimestampLen

入力。 `piTimestamp` バッファーの長さを示す 4 バイトの符号なし整数 (バイト単位)。

piTargetDBPath

入力。サーバー上のターゲット・データベースのディレクトリーの相対または完全修飾名を含むストリング。リストアされるバックアップのために新規のデータベースを作成する場合に使用します。そうでない場合は使用しません。

db2Restore - データベースのリストア

piReportFile

入力。ファイル名が指定されている場合、完全修飾名にする必要があります。リストア時にリンク解除されるデータ・リンク・ファイル (高速調整の結果) が報告されます。

iReportFileLen

入力。 piReportFile バッファの長さを示す 4 バイトの符号なし整数 (バイト単位) です。

piTablespaceList

入力。リストアされる表スペースのリストです。データベースまたは表スペース・バックアップ・イメージから、表スペースのサブセットをリストアする場合に使用されます。詳細については、 *DB2TablespaceStruct* 構造を参照してください。以下の制限が適用されます。

- データベースはリカバー可能でなければなりません。つまり、ログ保存またはユーザー出口が使用可能になっていなければなりません。
- リストアされるデータベースは、バックアップ・イメージの作成に使用されたのと同じデータベースでなければなりません。つまり、表スペース・リストア機能を使用して、データベースに表スペースを追加することはできません。
- ロールフォワード・ユーティリティーは、パーティション・データベース環境でリストアされる表スペースが、同じ表スペースを含む他のデータベース・パーティションと必ず同期化されるようにします。表スペースのリストア操作が要求され、 *piTablespaceList* が NULL の場合、リストア・ユーティリティーはバックアップ・イメージ内のすべての表スペースのリストアを試みます。

バックアップ後に名前が変更された表スペースをリストアする場合、リストア・コマンドでは新しい表スペース名を使用する必要があります。古い表スペース名を使用すると、その表スペースを見つけることができません。

piMediaList

入力。バックアップ・イメージのソース・メディア。提供される情報は、 *locationType* フィールドの値によって異なります。 *locationType* に有効な値 (*sqlutil* で定義) は、以下のとおりです。

SQLU_LOCAL_MEDIA

ローカル装置 (テープ、ディスクまたはディスクットの組み合わせ)。

SQLU_TSM_MEDIA

TSM。ロケーション・ポインターが NULL に設定されている場合、DB2 で提供される TSM 共有ライブラリーが使用されます。別のパーティションの TSM 共有ライブラリーが必要な場合には、SQLU_OTHER_MEDIA を使用し、共有ライブラリー名を入力してください。

SQLU_OTHER_MEDIA

ベンダー製品。ロケーション・フィールド内の共有ライブラリー名を提供します。

SQLU_USER_EXIT

ユーザー出口。追加の入力は必要ありません (サーバーが OS/2 上にある場合のみ使用可能です)。

piUsername

入力。接続の試行時に使用されるユーザー名を含むストリングを指定します。NULL にすることもできます。

iUsernameLen

入力。piUsername の長さを示す 4 バイトの符号なし整数 (バイト単位)。ユーザー名が提供されていない場合は、ゼロに設定してください。

piPassword

入力。ユーザー名とともに使用されるパスワードを含むストリングを指定します。NULL にすることもできます。

iPasswordLen

入力。piPassword の長さを示す 4 バイトの符号なし整数 (バイト単位)。パスワードが提供されていない場合は、ゼロに設定してください。

piNewLogPath

入力。リストア完了後、ロギングに使用されるパスを表すストリング。このフィールドが NULL の場合、デフォルトのログ・パスが使用されます。

iNewLogPathLen

入力。piNewLogPath の長さを示す 4 バイトの符号なし整数 (バイト単位)。

piVendorOptions

入力。情報をアプリケーションからベンダー関数へ渡すのに使用されます。このデータ構造はフラットでなければなりません。つまり、間接のレベルはサポートされません。このデータについては、バイト反転が行われず、また、コード・ページがチェックされないことに注意してください。

iVendorOptionsSize

入力。piVendorOptions の長さです。65535 バイト以下でなければなりません。

iParallelism

入力。並列処理の度合い (バッファー・マニピュレーターの数) を指定します。最小値は 1 です。最大値は 1024 です。デフォルトは 1 です。

iBufferSize

入力。バッファー・サイズを 4KB の割り振り単位 (ページ) でバックアップします。最小値は 8 単位です。デフォルトは 1024 単位です。リストア用に入力

db2Restore - データベースのリストア

するサイズは、バックアップ・イメージを作成するのに使用するバッファー・サイズと同じか、または整数倍でなければなりません。

iNumBuffers

入力。使用するリストア・バッファーの数を指定します。

iCallerAction

入力。実行するアクションを指定します。有効な値 (db2ApiDf で定義) は、以下のとおりです。

DB2RESTORE_RESTORE

リストア操作を開始します。

DB2RESTORE_NOINTERRUPT

リストアを開始します。リストアを自動実行するよう指定します。通常ユーザーの介入を要求するシナリオは、呼び出し側への最初の戻りなしに試行されるか、エラーを生成します。この呼び出し側アクションは、たとえば、リストアに必要なメディアがすべてマウントされていることが明らかで、ユーティリティーによるプロンプトが必要とされない場合に使用してください。

DB2RESTORE_CONTINUE

ユーザーがユーティリティーによって要求された何らかのアクション (たとえば、新しいテープのマウント) を実行した後で、リストアを継続します。

DB2RESTORE_TERMINATE

ユーザーがユーティリティーによって要求された何らかのアクションの実行に失敗した場合、リストアを終了します。

DB2RESTORE_DEVICE_TERMINATE

リストアによって使用される装置のリストから特定の装置を除外します。特定の装置が入力でいっぱいになると、リストア・ユーティリティーは呼び出し側に警告を戻します。その場合、この呼び出し側アクションを指定してリストアを再び呼び出すことによって、警告が生成される原因となった装置を使用装置のリストから除外してください。

DB2RESTORE_PARM_CHK

リストアを実行することなく、パラメーターの妥当性を検査します。このオプションは、呼び出しが戻った後でデータベース接続を終了しません。この呼び出しが正常に戻った後、ユーザーが DB2RESTORE_CONTINUE で呼び出しを発行し、アクションを進めることが期待されます。

DB2RESTORE_PARM_CHK_ONLY

リストアを実行することなく、パラメーターの妥当性を検査します。この呼び出しが戻る前に、この呼び出しによって確立したデータベース接続は終了し、後続する呼び出しは必要なくなります。

DB2RESTORE_TERMINATE_INCRE

完了前に増分リストア操作を終了します。

DB2RESTORE_RESTORE_STORDEF

最初の呼び出し。表スペース・コンテナの再定義が要求されます。

DB2RESTORE_STORDEF_NOINTERRUPT

最初の呼び出し。リストアは割り込まれずに実行されます。表スペース・コンテナの再定義が要求されます。

iOptions

入力。リストア・プロパティのビットマップ。オプションは組み合わせられて、ビット単位 OR 演算子を使用して *iOptions* の値を生成します。有効な値 (db2ApiDf で定義) は、以下のとおりです。

DB2RESTORE_OFFLINE

オフライン・リストア操作を実行します。

DB2RESTORE_ONLINE

オンライン・リストア操作を実行します。

DB2RESTORE_DB

データベースにあるすべての表スペースをリストアします。これはオフラインで実行する必要があります。

DB2RESTORE_TABLESPACE

バックアップ・イメージから、*piTablespaceList* パラメーターにリストされた表スペースのみリストアします。これはオンラインまたはオフラインで実行できます。

DB2RESTORE_HISTORY

履歴ファイルだけをリストアします。

DB2RESTORE_INCREMENTAL

手動累積リストア操作を実行します。

DB2RESTORE_AUTOMATIC

自動累積 (増分) リストア操作を実行します。

DB2RESTORE_INCREMENTAL とともに指定する必要があります。

DB2RESTORE_DATALINK

調整操作を実行します。定義された DATALINK 列を含む表では、RECOVERY YES オプションを指定する必要があります。

DB2RESTORE_NODATALINK

調整操作を実行しません。DATALINK 列を持つ表は、

DataLink_Roconcile_pending (DRP) 状態に入られます。定義された DATALINK 列を含む表では、RECOVERY YES オプションを指定する必要があります。

db2Restore - データベースのリストア

DB2RESTORE_ROLLFWD

データベースのリストアが成功した後、データベースをロールフォワード・ペンディング状態にします。

DB2RESTORE_NOROLLFWD

データベースのリストアが成功した後、データベースをロールフォワード・ペンディング状態にしません。バックアップがオンラインで実行される場合、または表スペース・レベルのリストアの場合は、この値は指定できません。リストアが成功した後で、データベースがロールフォワード保留状態にある場合は、`db2Rollforward` (データベースのロールフォワード) を実行してからでなければ、データベースを使用できません。

tablespaces

バックアップを取る表スペースのリストを指すポインター。C の場合、リストは NULL 終了ストリングです。一般的には、`db2Char` 構造のリストです。

numTablespaces

tablespaces パラメーター内の項目数。

locations

メディア・ロケーションのリストを指すポインター。C の場合、リストは NULL 終了ストリングです。一般的には、`db2Char` 構造のリストです。

numLocations

locations パラメーター内の項目数。

locationType

メディア・タイプを示す文字。有効な値 (`sqlutil` で定義) は、以下のとおりです。

SQLU_LOCAL_MEDIA

ローカル装置 (テープ、ディスク、ディスクレット、または Named Pipe)

SQLU_TSM_MEDIA

Tivoli Storage Manager。

SQLU_OTHER_MEDIA

ベンダー・ライブラリー。

SQLU_USER_EXIT

ユーザー出口 (サーバーが OS/2 上にある場合のみ使用可能です)。

pioData

文字データ・バッファーを指すポインター。

iLength

入力。 *pioData* バッファーのサイズ

oLength

出力。将来の使用のために予約されています。

使用上の注意:

オフラインのリストアの場合、このユーティリティーは、排他モードでデータベースに接続します。リストアされるデータベースにアプリケーション (呼び出し側のアプリケーションを含む) がすでに接続している場合、このユーティリティーは失敗します。さらに、リストア・ユーティリティーが、リストアの実行に使用されており、アプリケーション (呼び出し側のアプリケーションを含む) が同じワークステーション上の任意のデータベースにすでに接続されている場合、要求は失敗します。接続が成功すると、API はリストアが完了するまで他のアプリケーションを締め出します。

現行のデータベース構成ファイルは、それが使用不能でない限り、バックアップ・コピーによって置換されません。ファイルが置換される場合には、警告メッセージが戻されます。

db2Backup (データベースのバックアップ) を使用して、データベースまたは表スペースをバックアップする必要があります。

呼び出し側アクションが DB2RESTORE_NOINTERRUPT の場合、リストアはアプリケーションにプロンプトを出すことなく継続されます。呼び出し側のアクションが DB2RESTORE_RESTORE で、ユーティリティーが既存のデータベースにリストアしようとしている場合、ユーティリティーは、何らかのユーザー介入を要求するメッセージとともに、アプリケーションに制御を戻します。ユーザーとの対話の処理が終了した後、後続の呼び出しにおいて処理が継続される (DB2RESTORE_CONTINUE) か、終了 (DB2RESTORE_TERMINATE) されるかを示す呼び出し側アクションを設定して、RESTORE DATABASE を再び呼び出します。このユーティリティーは処理を終了させ、*sqlca* で SQLCODE を戻します。

終了時に装置をクローズするには、呼び出し側アクションを DB2RESTORE_DEVICE_TERMINATE に設定してください。たとえば、2 つのテープ装置を使用して 3 つのテープ・ボリュームからリストアを行おうとする場合、テープの 1 つがリストアされると、アプリケーションは、API からテープの終わりを示す SQLCODE とともに制御を受け取ります。ここで、アプリケーションはユーザーに別のテープをマウントするよう要求しますが、テープが「もうない」ことをユーザーが示すと、メディア装置の終了を通知する呼び出し側アクション SQLUD_DEVICE_TERMINATE を指定して API に戻ります。これで、デバイス・ドライバは終了しますが、リストアに関連する残りの装置は、リストア・セット内の全セグメントがリストアされるまで処理済みの入力を保持し続けます (バックアップ処理中に、リストア・セット内のセグメントの数が最後のメディア装置に置かれます)。この呼び出し側アクションは、テープ以外の装置 (バンダーによってサポートされる装置) でも使用できます。

db2Restore - データベースのリストア

アプリケーションに戻る前にパラメーター・チェックを実行するには、呼び出し側アクションを `DB2RESTORE_PARM_CHK` に設定してください。

リダイレクトしたリストアを実行する場合には、呼び出し側アクションを `DB2RESTORE_RESTORE_STORDEF` に設定し、「`sqlbstsc` - 表スペース・コンテナーの設定」との組み合わせで使用してください。

データベース・リストアの重要な段階でシステム障害が発生した場合、正常なリストアが実行されるまで、ユーザーはデータベースに正常に接続することができません。接続を試みたときにエラー・メッセージが戻されることによって、この状態であることが検出されます。バックアップされたデータベースがロールフォワード・リカバリーに使用できるよう構成されておらず、さらにログ保存パラメーターとユーザー出口パラメーターのいずれかが使用可能になっている、使用可能な現行の構成ファイルがある場合、ユーザーは、リストアに続いて、データベースの新しいバックアップをとるか、またはこれらのパラメーターを使用不能にしてから、データベースに接続することが必要です。

リストアが失敗すると、リストアされたデータベースはドロップされませんが (既存のデータベース以外へのリストアの場合を除き)、使用不能になります。

バックアップ上の履歴ファイルをリストアするようにリストア・タイプで指定されている場合、それはデータベースの既存の履歴ファイル上にリストアされます。事実上、リストアされるバックアップの後に履歴ファイルに加えられた変更は、すべて消去されることとなります。このことが望ましくない場合は、履歴ファイルを新規またはテスト・データベースにリストアさせることにより、実行された更新を破棄することなく、履歴ファイルの内容を表示できるようにしてください。

バックアップ操作時にデータベースのロールフォワード・リカバリーが使用可能だった場合には、`db2Restore` の実行が成功した後に `db2Rollforward` を発行することによって、データベースを損傷または破壊の発生前の状態に戻すことができます。データベースがリカバリー可能な場合、リストアの完了後に、デフォルトでペンディング状態がロールフォワードされます。

データベース・バックアップ・イメージがオフラインで作成されており、呼び出し側がリストア後のデータベースのロールフォワードを必要としない場合、リストア用に `DB2RESTORE_NOROLLFWD` オプションを使用できます。これにより、リストア後にデータベースがすぐに使用可能になります。バックアップ・イメージがオンラインで作成されている場合は、呼び出し元はリストアが完了した時点で、対応するログ・レコードを使用してロールフォワードを行わなければなりません。

関連資料:

- 管理 API リファレンス の『`sqlcmgdb` - データベースの移行』
- 1007 ページの『`db2Rollforward` - データベースのロールフォワード』
- 管理 API リファレンス の『SQLCA』

- 412 ページの『db2Backup - データベースのバックアップ』
- 管理 API リファレンス の『db2CfgGet - 構成パラメーターの入手』

関連サンプル:

- 『dbrecov.sql -- How to recover a database (C)』
- 『dbrecov.sqlC -- How to recover a database (C++)』

db2ReadLogNoConnInit - データベース接続なしのログ読み取りの初期設定

db2ReadLogNoConn によって使用されるメモリーを割り振り、ログ・レコードを DB2 UDB データベース・ログから抽出します。また、ログ・マネージャーを照会して現行のログ状態に関する情報を取得できるようにします。

許可:

なし

必要な接続:

なし

API 組み込みファイル:

db2ApiDf.h

C API 構文:

```

/* File: db2ApiDf.h */
/* API: db2ReadLogNoConnInit */
/* ... */
SQL_API_RC SQL_API_FN
db2ReadLogNoConnInit (
    db2UInt32 versionNumber,
    void * pDB2ReadLogNoConnInitStruct,
    struct sqlca * pSqlca);

typedef SQL_STRUCTURE db2ReadLogNoConnInitStruct
{
    db2UInt32                iFilterOption;
    char                    *piLogFilepath;
    char                    *piOverflowLogPath;
    db2UInt32                iRetrieveLogs;
    char                    *piDatabaseName;
    char                    *piNodeName;
    db2UInt32                iReadLogMemoryLimit;
    char                    **poReadLogMemPtr;
} db2ReadLogNoConnInitStruct;
/* ... */

```

API パラメーター:

db2ReadLogNoConnInit - データベース接続なしのログ読み取りの初期設定

バージョン番号

入力。 2 番目のパラメーター *pDB2ReadLogNoConnInitStruct* として渡される構造のバージョンとリリースのレベルを指定します。

pDB2ReadLogNoConnInitStruct

入力。 *db2ReadLogNoConnInitStruct* 構造を指すポインター。

pSqlca 出力。sqlca 構造へのポインター。

iFilterOption

入力。ログ・レコードを読み取るときに使用されるログ・レコード・フィルターのレベルを指定します。有効な値は以下のとおりです。

DB2READLOG_FILTER_OFF

指定された LSN 範囲内ですべてのログ・レコードを読み取ります。

DB2READLOG_FILTER_ON

伝搬可能とマークされた LSN 範囲内でログ・レコードのみを読み取ります。これは非同期ログ読み取り API の基本的な動作です。

piLogFilePath

入力。読み取られるログ・ファイルがある場所のパス。

piOverflowLogPath

入力。読み取られるログ・ファイルがある場所の代替パス。

iRetrieveLogs

入力。ログ・ファイル・パス、またはオーバーフロー・ログ・パスのどちらでも検索できないログ・ファイルを検索するために、ユーザー出口を呼び出すかどうかを指定するオプション。有効な値は以下のとおりです。

DB2READLOG_RETRIEVE_OFF

欠落したログ・ファイルを検索するために、ユーザー出口は呼び出しません。

DB2READLOG_RETRIEVE_LOGPATH

欠落したログ・ファイルを検索するために、指定されたログ・ファイル・パスにユーザー出口を呼び出します。

DB2READLOG_RETRIEVE_OVERFLOW

欠落したログ・ファイルを検索するために、指定されたオーバーフロー・ログ・パスにユーザー出口を呼び出します。

piDatabaseName

入力。読み取り中のリカバリー・ログを所有するデータベースの名前。これは、上記の検索オプションが指定された場合に必要です。

piNodeName

入力。読み取り中のリカバリー・ログを所有するノードの名前。これは、上記の検索オプションが指定された場合に必要です。

db2ReadLogNoConnInit - データベース接続なしのログ読み取りの初期設定

iReadLogMemoryLimit

入力。 API が内部に割り振る最大バイト数。

poReadLogMemPtr

出力。 API が割り振った、サイズ *iReadLogMemoryLimit* のメモリーのブロック。このメモリーには、呼び出しのたびに API が必要とする永続データが含まれています。このメモリー・ブロックは、どのような方法であっても呼び出し側は再割り振りまたは変更してはなりません。

使用上の注意:

db2ReadLogNoConnInit によって初期設定されたメモリーは変更してはなりません。

db2ReadLogNoConn がもう使用されない場合、db2ReadLogNoConnTerm を呼び出し、db2ReadLogNoConnInit によって初期設定されたメモリーを割り振り解除します。

関連資料:

- 管理 API リファレンス の『SQLCA』
- 1024 ページの『db2ReadLogNoConn - データベース接続なしのログの読み取り』
- 420 ページの『db2ReadLogNoConnTerm - データベース接続なしのログ読み取りの終了』

db2Rollforward - データベースのロールフォワード

データベース・ログ・ファイルに記録されたトランザクションを適用することによって、データベースをリカバーします。この API は、データベースまたは表スペースのバックアップがリストアされた後、あるいはメディア・エラーが原因で表スペースがデータベースによってオフラインにされた場合に呼び出されます。ロールフォワード・リカバリーを用いてデータベースをリカバーするには、前もってデータベースがリカバリー可能でなければなりません (すなわち、データベース構成パラメーター *logretain*、*userexit*、あるいはこの両方がオンに設定されていなければなりません)。

有効範囲:

パーティション・データベース環境では、この API はカタログ・パーティションからのみ呼び出すことができます。データベースまたは表スペースのポイント・イン・タイムを指定したロールフォワード呼び出しは、*db2nodes.cfg* ファイルにリストされているすべてのデータベース・パーティション・サーバーに影響を与えます。ログの終わりを指定したデータベースまたは表スペース・ロールフォワード呼び出しは、指定されたデータベース・パーティション・サーバーに影響を与えます。データベース・パーティション・サーバーが 1 つも指定されていない場合には、*db2nodes.cfg* ファイルにリストされているすべてのデータベース・パーティション・サーバーに影響を与えます。特定のデータベース・パーティション・サーバーでロールフォワードが必要ない場合、そのデータベース・パーティション・サーバーは無視されます。

db2Rollforward - データベースのロールフォワード

許可:

以下のいずれかです。

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

必要な接続:

なし。この API によってデータベース接続が確立されます。

API 組み込みファイル:

db2ApiDf.h

C API 構文:

```
/* File: db2ApiDf.h */
/* API: db2Rollforward */
/* ... */
SQL_API_RC SQL_API_FN
db2Rollforward (
    db2UInt32 versionNumber,
    void *pDB2RollforwardStruct,
    struct sqlca *pSqlca);

typedef SQL_STRUCTURE db2RollforwardStruct
{
    struct db2RfwdInputStruct *piRfwdInput;
    struct db2RfwdOutputStruct *poRfwdOutput;
} db2RollforwardStruct;

typedef SQL_STRUCTURE db2RfwdInputStruct
{
    sqluint32                iVersion;
    char                    *piDbAlias;
    db2UInt32                iCallerAction;
    char                    *piStopTime;
    char                    *piUserName;
    char                    *piPassword;
    char                    *piOverflowLogPath;
    db2UInt32                iNumChngLgOvrflw;
    struct sqlurf_newlogpath *piChngLogOvrflw;
    db2UInt32                iConnectMode;
    struct sqlu_tablespace_bkrst_list *piTablespaceList;
    db2int32                iAllNodeFlag;
    db2int32                iNumNodes;
    SQL_PDB_NODE_TYPE       *piNodeList;
    db2int32                iNumNodeInfo;
    char                    *piDroppedTblID;
    char                    *piExportDir;
    db2UInt32                iRollforwardFlags;
} db2RfwdInputStruct;
```

```

typedef SQL_STRUCTURE db2RfwdOutputStruct
{
    char                *poApplicationId;
    sqlint32            *poNumReplies;
    struct sqlurf_info  *poNodeInfo;
} db2RfwdOutputStruct;

typedef SQL_STRUCTURE sqlurf_newlogpath
{
    SQL_PDB_NODE_TYPE nodenum;
    unsigned short     pathlen;
    char                logpath[SQL_LOGPATH_SZ+SQL_LOGFILE_NAME_SZ+1];
} sqlurf_newlogpath;

typedef SQL_STRUCTURE sqlu_tablespace_bkrst_list
{
    long                num_entry;
    struct sqlu_tablespace_entry *tablespace;
} sqlu_tablespace_bkrst_list;

typedef SQL_STRUCTURE sqlu_tablespace_entry
{
    sqluint32          reserve_len;
    char               tablespace_entry[SQLU_MAX_TBS_NAME_LEN+1];
    char               filler[1];
} sqlu_tablespace_entry;

typedef SQL_STRUCTURE sqlurf_info
{
    SQL_PDB_NODE_TYPE nodenum;
    sqlint32           state;
    unsigned char      nextarclog[SQLUM_ARCHIVE_FILE_LEN+1];
    unsigned char      firstarcdel[SQLUM_ARCHIVE_FILE_LEN+1];
    unsigned char      lastarcdel[SQLUM_ARCHIVE_FILE_LEN+1];
    unsigned char      lastcommit[SQLUM_TIMESTAMP_LEN+1];
} sqlurf_info;
/* ... */

```

汎用 API 構文:

```

/* File: db2ApiDf.h */
/* API: db2Rollforward */
/* ... */
SQL_API_RC SQL_API_FN
db2gRollforward (
    db2Uint32 versionNumber,
    void *pDB2gRollforwardStruct,
    struct sqlca *pSqlca);

typedef SQL_STRUCTURE db2gRollforwardStruct
{
    struct db2gRfwdInputStruct *piRfwdInput;
    struct db2RfwdOutputStruct *poRfwdOutput;
} db2gRollforwardStruct;

```

db2Rollforward - データベースのロールフォワード

```
SQL_STRUCTURE db2gRfwdInputStruct
{
    db2UInt32          iDbAliasLen;
    db2UInt32          iStopTimeLen;
    db2UInt32          iUserNameLen;
    db2UInt32          iPasswordLen;
    db2UInt32          iOvrflwLogPathLen;
    db2UInt32          iDroppedTblIDLen;
    db2UInt32          iExportDirLen;
    sqluint32          iVersion;
    char               *piDbAlias;
    db2UInt32          iCallerAction;
    char               *piStopTime;
    char               *piUserName;
    char               *piPassword;
    char               *piOverflowLogPath;
    db2UInt32          iNumChngLgOvrflw;
    struct sqlurf_newlogpath *piChngLogOvrflw;
    db2UInt32          iConnectMode;
    struct sqlu_tablespace_bkrst_list *piTablespaceList;
    db2int32           iAllNodeFlag;
    db2int32           iNumNodes;
    SQL_PDB_NODE_TYPE *piNodeList;
    db2int32           iNumNodeInfo;
    char               *piDroppedTblID;
    char               *piExportDir;
    db2UInt32          iRollforwardFlags;
} db2gRfwdInputStruct;

typedef SQL_STRUCTURE db2RfwdOutputStruct
{
    char               *poApplicationId;
    sqlint32           *poNumReplies;
    struct sqlurf_info *poNodeInfo;
} db2RfwdOutputStruct;

typedef SQL_STRUCTURE sqlurf_newlogpath
{
    SQL_PDB_NODE_TYPE nodenum;
    unsigned short     pathlen;
    char               logpath[SQL_LOGPATH_SZ+SQL_LOGFILE_NAME_SZ+1];
} sqlurf_newlogpath;

typedef SQL_STRUCTURE sqlu_tablespace_bkrst_list
{
    long               num_entry;
    struct sqlu_tablespace_entry *tablespace;
} sqlu_tablespace_bkrst_list;

typedef SQL_STRUCTURE sqlu_tablespace_entry
{
    sqluint32          reserve_len;
    char               tablespace_entry[SQLU_MAX_TBS_NAME_LEN+1];
    char               filler[1];
} sqlu_tablespace_entry;
```

```

typedef SQL_STRUCTURE sqlurf_info
{
    SQL_PDB_NODE_TYPE nodenum;
    sqlint32         state;
    unsigned char   nextarclog[SQLUM_ARCHIVE_FILE_LEN+1];
    unsigned char   firstarcdel[SQLUM_ARCHIVE_FILE_LEN+1];
    unsigned char   lastarcdel[SQLUM_ARCHIVE_FILE_LEN+1];
    unsigned char   lastcommit[SQLUM_TIMESTAMP_LEN+1];
} sqlurf_info;
/* ... */

```

API パラメーター:**versionNumber**

入力。 2 番目のパラメーターとして渡される構造のバージョンとリリースのレベルを指定します。

pDB2RollforwardStruct

入力。 *db2RollforwardStruct* 構造を指すポインター。

pSqlca 出力。 *sqlca* 構造へのポインター。

piRfwdInput

入力。 *db2RfwdInputStruct* 構造を指すポインター。

poRfwdOutput

出力。 *db2RfwdOutputStruct* 構造を指すポインター。

iDbAliasLen

入力。 データベース別名の長さ (バイト単位) を指定します。

iStopTimeLen

入力。 停止時刻パラメーターの長さ (バイト単位) を指定します。 停止時刻が提供されていない場合は、ゼロに設定してください。

iUserNameLen

入力。 ユーザー名の長さ (バイト単位) を指定します。 ユーザー名が提供されていない場合は、ゼロに設定してください。

iPasswordLen

入力。 パスワードの長さ (バイト単位) を指定します。 パスワードが提供されていない場合は、ゼロに設定してください。

iOverflowLogPathLen

入力。 オーバーフロー・ログ・パスの長さ (バイト単位) を指定します。 オーバーフロー・ログ・パスが提供されていない場合は、ゼロに設定してください。

iVersion

入力。 ロールフォワード・パラメーターのバージョン ID。
SQLUM_RFWD_VERSION として定義されています。

db2Rollforward - データベースのロールフォワード

piDbAlias

入力。データベース別名を含むストリング。これは、システム・データベース・ディレクトリーにカタログされている別名です。

iCallerAction

入力。実行するアクションを指定します。有効な値 (db2ApiDf.h で定義) は、以下のとおりです。

DB2ROLLFORWARD_ROLLFWD

piStopTime で指定された時点までロールフォワードします。データベースのロールフォワードの場合、データベースはロールフォワード保留 状態のままになります。表スペースのロールフォワードの場合、表スペースはロールフォワード処理 状態のままになります。

DB2ROLLFORWARD_STOP

ロールフォワード・リカバリーを終了します。新しいログ・レコードは処理されず、コミットされていないトランザクションはバックアウトされます。データベースまたは表スペースのロールフォワード保留 状態はオフになります。同義語は DB2ROLLFORWARD_RFWD_COMPLETE です。

DB2ROLLFORWARD_RFWD_STOP

piStopTime で指定された時点までロールフォワードし、ロールフォワード・リカバリーを終了します。データベースまたは表スペースのロールフォワード保留 状態はオフになります。同義語は DB2ROLLFORWARD_RFWD_COMPLETE です。

DB2ROLLFORWARD_QUERY

nextarclog、*firstarcdel*、*lastarcdel* および *lastcommit* の照会値。データベース状況とノード番号を戻します。

DB2ROLLFORWARD_PARM_CHECK

ロールフォワードを実行することなく、パラメーターの妥当性を検査します。

DB2ROLLFORWARD_CANCEL

現在実行中のロールフォワード操作を取り消します。データベースまたは表スペースは、リカバリー・ペンディング状態に置かれます。

注: このオプションは、ロールフォワードが実際に実行中であるときには使用できません。ロールフォワードが休止されている (つまり、STOP を待っている) 場合、あるいはロールフォワード中にシステム障害が発生した場合に使用できます。このオプションの使用に際しては、注意が必要です。

データベースをロールフォワードするとき、テープ装置を使用するロード・リカバリーが必要な場合もあります。装置に関してユーザーの介入が必要な場

db2Rollforward - データベースのロールフォワード

合、ロールフォワード API は警告メッセージを戻します。以下の 3 つのアクションのいずれかを指定して、再び API を呼び出すことができます。

DB2ROLLFORWARD_LOADREC_CONT

警告メッセージを生成した装置を使用し続けます (たとえば、新しいテープがマウントされた場合)。

DB2ROLLFORWARD_DEVICE_TERM

警告メッセージを生成した装置の使用を停止します (たとえば、それ以上テープがない場合)。

DB2ROLLFORWARD_LOAD_REC_TERM

ロード・リカバリーに使用されているすべての装置を終了させます。

piStopTime

入力。ISO 形式のタイム・スタンプを含む文字ストリング。このタイム・スタンプで設定された時刻を過ぎると、データベース・リカバリーは停止します。可能な限りロールフォワードしたい場合には、SQLUM_INFINITY_TIMESTAMP を指定してください。

DB2ROLLFORWARD_QUERY、DB2ROLLFORWARD_PARAM_CHECK、およびいずれかのロード・リカバリー (DB2ROLLFORWARD_LOADREC_xxx) 呼び出し側アクションの場合には、NULL にすることができます。

piUserName

入力。アプリケーションのユーザー名を含むストリング。NULL にすることもできます。

piPassword

入力。提供されたユーザー名 (ある場合) のパスワードを含むストリング。NULL にすることもできます。

piOverflowLogPath

入力。使用される代替ログ・パスを指定します。このユーティリティを使用する前に、アクティブ・ログ・ファイルの他に、アーカイブ・ログ・ファイルをユーザーが *logpath* に移動させることが必要です。このことは、*logpath* に十分なスペースがない場合に問題になる可能性があります。その問題を解決するために、オーバーフロー・ログ・パスが備えられています。ロールフォワード・リカバリー中に、必要なログ・ファイルは、まず *logpath* で探索され、次にオーバーフロー・ログ・パスで探索されます。表スペースのロールフォワード・リカバリーに必要なログ・ファイルは、*logpath* またはオーバーフロー・ログ・パスのいずれかに置かれる可能性があります。呼び出し側がオーバーフロー・ログ・パスを指定しない場合、デフォルト値は *logpath* です。パーティション・データベース環境では、オーバーフロー・ログ・パスは有効な完全修飾パスでなければなりません。デフォルトのパスは、各ノードのデフォルトのオーバーフロー・ログ・パスです。単一パーティション・データベース環境では、サーバーがローカルであれば、オーバーフロー・ログ・パスは相対パスにすることもできます。

db2Rollforward - データベースのロールフォワード

iNumChngLgOvrflw

入力。パーティション・データベース環境のみ。変更されるオーバーフロー・ログ・パスの数。新しいログ・パスは、指定されたデータベース・パーティション・サーバーのデフォルトのオーバーフロー・ログ・パスだけをオーバーライドします。

piChngLogOvrflw

入力。パーティション・データベース環境のみ。変更されるオーバーフロー・ログ・パスの完全修飾名が入っている構造を指すポインター。新しいログ・パスは、指定されたデータベース・パーティション・サーバーのデフォルトのオーバーフロー・ログ・パスだけをオーバーライドします。

iConnectMode

入力。有効な値 (db2ApiDf.h で定義) は、以下のとおりです。

DB2ROLLFORWARD_OFFLINE

オフライン・ロールフォワード。データベースのロールフォワード・リカバリーの場合には、必ずこの値を指定してください。

DB2ROLLFORWARD_ONLINE

オンライン・ロールフォワード。

piTablespaceList

入力。ログの終わりまで、または指定された時点までロールフォワードされる表スペースの名前が入っている構造を指すポインター。指定されない場合には、ロールフォワードを必要とする表スペースが選択されます。

iAllNodeFlag

入力。パーティション・データベース環境のみ。ロールフォワード操作が、db2nodes.cfg で定義されているすべてのデータベース・パーティション・サーバーに適用されるかどうかを示します。有効な値は以下のとおりです。

DB2_NODE_LIST

piNodeList で渡されたリスト内のデータベース・パーティション・サーバーに適用されます。

DB2_ALL_NODES

すべてのデータベース・パーティション・サーバーに適用されます。*piNodeList* は NULL でなければなりません。これがデフォルト値です。

DB2_ALL_EXCEPT

piNodeList で渡されたリスト内のデータベース・パーティション・サーバーを除いた、すべてのデータベース・パーティション・サーバーに適用されます。

DB2_CAT_NODE_ONLY

カタログ・パーティションにのみ適用されます。*piNodeList* は NULL でなければなりません。

iNumNodes

入力。 *piNodeList* 配列内のデータベース・パーティション・サーバーの数を指定します。

piNodeList

入力。ロールフォワード操作を実行する対象のデータベース・パーティション・サーバー番号の配列を指すポインター。

iNumNodeInfo

入力。出力パラメーター *poNodeInfo* のサイズを定義します。これは、ロールフォワードされるそれぞれのデータベース・パーティションからの状況情報を保持するのに十分な大きさでなければなりません。単一パーティション・データベース環境では、このパラメーターは 1 に設定する必要があります。このパラメーターの値は、この API が呼び出されるデータベース・パーティション・サーバーの数と同じにする必要があります。

piDroppedTblID

入力。リカバリーが実行されているドロップ済み表の ID を含むストリング。

piExportDir

入力。ドロップした表データのエクスポート先のディレクトリー。

RollforwardFlags

入力。ロールフォワード・フラグを指定します。有効な値 (db2ApiDf.h で定義) は、以下のとおりです。

DB2ROLLFORWARD_EMPTY_FLAG

フラグが指定されていません。

DB2ROLLFORWARD_LOCAL_TIME

GMT 時刻ではなく、ユーザーのローカル時刻での時点にロールフォワードすることができます。これによって、ユーザーがローカル・マシンで特定の時点にロールフォワードすることが容易になり、現地時間を GMT 時刻に変換することによって生じる潜在的なユーザー・エラーの発生を除去します。

DB2ROLLFORWARD_NO_RETRIEVE

ユーザーがアーカイブ・ログの検索を使用不可にすることを許可することによって、待機マシン上でどのログ・ファイルがロールフォワードされるべきかをコントロールします。ログ・ファイルのロールフォワードをコントロールすることによって、待機マシンが実動マシンより X 時間遅れていることを確認でき、ユーザーが両システムに影響を与えることを防ぐことができます。待機システムがアーカイブにアクセスせず、たとえば TSM がアーカイブの場合に、元のマシンがファイルを検索することだけを許可する場合に、このオプションは役立ちます。また、実動システムがファイルをアーカイブし、待機システム

db2Rollforward - データベースのロールフォワード

が同じファイルを検索している間、待機システムが不完全なログ・ファイルを検索するという可能性も除去します。

poApplicationId

出力。アプリケーション ID。

poNumReplies

出力。受信した応答の数。

poNodeInfo

出力。データベース・パーティション応答情報。

nodenum

ノード番号。

pathlen

新規 logpath の長さ。

logpath

新規オーバーフロー・ログ・パス。

num_entry

tablespace フィールドによって指摘されるリスト内のエントリーの数。

tablespace

sqlu_tablespace_entry 構造を指すポインター。

reserve_len

tablespace_entry フィールドに提供されている文字ストリングの長さ。 C 以外の言語用。

tablespace_entry

表スペースの名前。

state 状態の情報。

nextarclog

戻された、次に必要なアーカイブ・ログ・ファイルの名前を保持するバッファ。 DB2ROLLFORWARD_QUERY 以外の呼び出し側アクションが提供された場合、このフィールドに戻される値は、ファイルへのアクセス中にエラーが発生したことを示します。以下の理由が考えられます。

- ファイルがデータベース・ログ・ディレクトリー、またはオーバーフロー・ログ・パス・パラメーターで指定されたパスのどちらにも見つかりませんでした。
- ユーザー出口プログラムが、アーカイブ・ファイルを戻すのに失敗しました。

firstarcdel

戻された、リカバリー用に必要のなくなった最初のアーカイブ・ログ・ファイ

ルの名前を保持するバッファー。このファイル、および `lastarcdel` を含む、それまでのすべてのファイルは、ディスクの容量を増やすために移動できます。

たとえば、`firstarcdel` および `lastarcdel` に返された値が `S0000001.LOG` および `S0000005.LOG` である場合、以下のログ・ファイルを移動できます。

- `S0000001.LOG`
- `S0000002.LOG`
- `S0000003.LOG`
- `S0000004.LOG`
- `S0000005.LOG`

lastarcdel

戻された、データベース・ログ・ディレクトリーから除去可能な最後のアーカイブ・ログ・ファイルの名前を保持するバッファー。

lastcommit

ISO 形式のタイム・スタンプを含むストリング。この値は、ロールフォワード操作の終了後の、最後にコミットされたトランザクションのタイム・スタンプを示します。

使用上の注意:

データベース・マネージャーは、アーカイブおよびログ・ファイルに格納された情報を使用して、最後のバックアップのときにデータベースで実行されたトランザクションを再構築します。

この API の呼び出し時に実行されるアクションは、呼び出し前のデータベースの `rollforward_pending` フラグによって異なります。これは `db2CfgGet` (構成パラメーターの入手) を使用して照会できます。データベースがロールフォワード保留状態にある場合、`rollforward_pending` フラグは `DATABASE` に設定されています。1 つまたは複数の表スペースが `SQLB_ROLLFORWARD_PENDING` または `SQLB_ROLLFORWARD_IN_PROGRESS` 状態にある場合、フラグは `TABLESPACE` に設定されています。データベースも表スペースもロールフォワードする必要がない場合は、`rollforward_pending` フラグが `NO` に設定されています。

この API の呼び出し時にデータベースがロールフォワード保留状態にある場合は、データベースがロールフォワードされます。表スペースは、異常状態によって 1 つまたは複数の表スペースがオフラインにならない限り、データベースのロールフォワードが正常に終了すると正常の状態に戻ります。`rollforward_pending` フラグが `TABLESPACE` に設定されている場合には、ロールフォワード保留状態にある表スペース、あるいは名前によって要求された表スペースだけがロールフォワードされます。

注: 表スペースのロールフォワードが異常終了してしまった場合、ロールフォワード中だった表スペースは、`SQLB_ROLLFORWARD_IN_PROGRESS` 状態に置かれます。次に

db2Rollforward - データベースのロールフォワード

ROLLFORWARD DATABASE を呼び出したときには、SQLB_ROLLFORWARD_IN_PROGRESS 状態にあるそれらの表スペースだけが処理されます。選択された表スペース名のセットに SQLB_ROLLFORWARD_IN_PROGRESS 状態のすべての表スペースが含まれているのではない場合には、要求されていない表スペースが SQLB_RESTORE_PENDING 状態に置かれます。

データベースがロールフォワード保留状態になく、特定の時点が指定されない場合には、ロールフォワード進行中状態にある表スペースがログの終わりまでロールフォワードされます。ロールフォワード進行中状態の表スペースがない場合には、ロールフォワード・ペンディング状態にある表スペースがログの終わりまでロールフォワードされません。

この API は、ログ・ファイルの読み取りを、バックアップ・イメージに一致するログ・ファイルから始めます。ログ・ファイルをロールフォワードする前に、DB2ROLLFORWARD_QUERY 呼び出し側アクションを指定してこの API を呼び出すと、このログ・ファイルの名前を判別することができます。

ログ・ファイル内のトランザクションは、データベースに再適用されます。ログは、情報が使用可能である限り、あるいは停止時刻パラメーターで指定された時刻まで、順方向に処理されます。

以下のイベントが生じると、リカバリーが停止します。

- ログ・ファイルがこれ以上見つからない
- ログ・ファイル内のタイム・スタンプが、停止時刻パラメーターで指定された完了タイム・スタンプを超えた。
- ログ・ファイルの読み取り中に、エラーが発生した。

一部のトランザクションは、リカバリーされない可能性があります。 *lascommit* で戻された値は、最後にコミットされ、データベースに適用されたトランザクションのタイム・スタンプを示します。

アプリケーションまたは人為エラーが原因でデータベースのリカバリーが必要となった場合、エラーが発生する前の時点でリカバリーを停止することを指示するために、*piStopTime* にタイム・スタンプ値を指定することができます。これは、データベースの全ロールフォワード・リカバリーと、表スペースの特定の時点までのロールフォワードに適用されます。また、このことにより、前回失敗したりカバリーの試みで判別された、ログ読み取りエラーが発生する前の時点でリカバリーを停止させることも可能になります。

rollforward_recovery フラグが DATABASE に設定されている場合、ロールフォワード・リカバリーが終了するまで、データベースは使用できません。 DB2ROLLFORWARD_STOP または DB2ROLLFORWARD_RFWRD_STOP の呼び出し側アクションを指定してこの API を呼び出すことにより、データベースのロールフォワード保留状態をオフにすれば、リカバリー

db2Rollforward - データベースのロールフォワード

を終了させることができます。 `rollforward_recovery` フラグが `TABLESPACE` になれば、データベースを使用できるようになります。ただし、`SQLB_ROLLFORWARD_PENDING` および `SQLB_ROLLFORWARD_IN_PROGRESS` 状態の表スペースは、表スペースのロールフォワード・リカバリーを実行するための API が呼び出されるまで使用不能になります。表スペースをある時点までロールフォワードすると、表スペースは、正常なロールフォワード後にバックアップ・ペンディング状態に置かれます。

`RollforwardFlags` オプションが `DB2ROLLFORWARD_LOCAL_TIME` に設定されている場合、ユーザーに戻されるすべてのメッセージは現地時間で示されます。パーティション・データベース環境の場合、すべての時間はサーバー、またはカタログ・パーティション上で変換されます。タイム・スタンプ・ストリングはサーバー上で GMT に変換されます。そのため、時間はクライアントの時間帯ではなく、サーバーの時間帯にローカルです。クライアントがある時間帯に属し、サーバーが別の時間帯に属している場合、サーバーの現地時間を使用する必要があります。これはクライアントにとってローカルなコントロール・センターの現地時間オプションとは異なります。タイム・スタンプ・ストリングが夏時間調整による時計の時間変更に近い場合、停止時刻が時計の変更の前か後かを知り、それを正確に指定することが重要です。

関連資料:

- 管理 API リファレンス の『SQLCA』
- 993 ページの『db2Restore - データベースのリストア』

関連サンプル:

- 『dbrecov.sqc -- How to recover a database (C)』
- 『dbrecov.sqC -- How to recover a database (C++)』

プリコンパイラーのカスタマイズ API

他のアプリケーション開発ツールの製品が、それらの製品内で DB2 用のプリコンパイラー・サポートを直接実現するための文書化された API のセット。たとえば AIX 上の IBM COBOL は、このインターフェースを使用します。プリコンパイラー・サービス API のセットに関する情報は、以下の DB2 アプリケーション開発 Web サイトにある PDF ファイル `prepapi.pdf` に記載されています。

<http://www.ibm.com/software/data/db2/udb/ad>

関連資料:

- *Data Links Manager* 概説およびインストール の『IBM と連絡をとる』

sqlsact - 会計情報ストリングの設定

アプリケーションの次の接続要求とともに、DRDA サーバーに送られる会計情報を提供します。

sqlsact - 会計情報ストリングの設定

許可:

なし

必要な接続:

なし

API 組み込みファイル:

sqlenv.h

C API 構文:

```
/* File: sqlenv.h */
/* API: sqlsact */
/* ... */
SQL_API_RC SQL_API_FN
sqlsact (
    char *pAccountingString,
    struct sqlca *pSqlca);
/* ... */
```

汎用 API 構文:

```
/* File: sqlenv.h */
/* API: sqlgsact */
/* ... */
SQL_API_RC SQL_API_FN
sqlgsact (
    unsigned short AccountingStringLength,
    char *pAccountingString,
    struct sqlca *pSqlca);
/* ... */
```

API パラメーター:

AccountingStringLength

入力。会計情報ストリングの長さを示す 2 バイトの符号なし整数 (バイト単位) を指定します。

pAccountingString

入力。会計情報を含むストリングを指定します。

pSqlca 出力。 *sqlca* 構造へのポインター。

使用上の注意:

会計情報を接続要求とともに送りたい場合には、データベースに接続する前に、アプリケーションからこの API を呼び出す必要があります。API を再び呼び出して別のデータベースに接続するまでは、会計情報ストリングに変更を加えることができます。接続しない場合には、アプリケーションが終了するまで、現行の値が有効のままになりま

す。会計情報ストリングは、最大 SQL_ACCOUNT_STR_SZ (sqlenv で定義) で指定されたバイト数の長さにまですることができます。それよりも長い場合は、切り捨てられます。DRDA サーバーへの伝送時に、会計情報ストリングが正しく変換されるようにするため、文字 A ~ Z、0 ~ 9、および下線記号 (_) だけを使用するようにしてください。

関連資料:

- 管理 API リファレンス の『sqleseti - クライアント情報の設定』
- 管理 API リファレンス の『SQLCA』

関連サンプル:

- 『setact.cbl -- How to set accounting string (IBM COBOL)』

アプリケーションの移行時の考慮事項

このセクションでは、アプリケーションをバージョン 8 に移行する前に考慮すべき問題について説明します。

考えられる操作のシナリオが 4 つあります。

1. 移行されていないデータベースに対して、バージョン 8 以前のアプリケーションを実行する
2. 移行済みデータベースに対して、バージョン 8 以前のアプリケーションを実行する
3. バージョン 8 の API でアプリケーションを更新する
4. 移行済みデータベースに対して、バージョン 8 のアプリケーションを実行する

1 番目と 4 番目は、修正を必要としない、一貫した操作環境です。

データベースのみが移行済みの 2 番目の操作環境は、バックレベルのアプリケーションがサポートされているため、アプリケーションに変更を加えなくても機能するはずですが、ただし、新しいバージョンについては、いくらかの非互換性が生じる可能性があります。

アプリケーションがバージョン 8 の API で更新される 3 番目のシナリオでは、以下の点を考慮する必要があります。

- バージョン 8 で廃止されている、バージョン 8 以前のすべての API は、バージョン 8 のヘッダー・ファイルで引き続き定義されています。これにより、古いアプリケーションがバージョン 8 のヘッダーを用いてコンパイルおよびリンクすることができます。
- 廃止された API はできるだけ早くアプリケーションから除去して、アプリケーションがバージョン 8 で利用可能な新しい関数を十分に活用できるようにし、将来の拡張に備える必要があります。

- アプリケーション移行時の考慮事項にリストされている API の名前は、バージョン 8 の新しい関数用に変更されています。ユーザーはアプリケーションのソース・コード内でこれらの名前をスキャンして、アプリケーションのバージョン 8 への移行後に必要になる変更を明らかにする必要があります。

ここにリストされていない API には、アプリケーションの移行後の変更は必要ありません。

アプリケーションには、使用中のアプリケーション・プログラミング言語によって、API 呼び出しの汎用バージョンが含まれることがあります。すべてのケースで、汎用バージョンの API 名は、4 番目の文字が常に **g** である点を除き、C バージョンの名前と同一です。

関連資料:

- 1028 ページの『変更された API およびデータ構造』

並行アクセスを伴うスレッド化アプリケーション

DB2 データベースに対するスレッド化アプリケーションのデフォルトのインプリメンテーションでは、データベースへのアクセスのシリアライゼーションはデータベース API によって実施されます。あるスレッドがデータベース呼び出しを実行する場合、最初の呼び出しが完了するまでは、続く呼び出しが最初の呼び出しに関連しないデータベース・オブジェクトにアクセスするとしても、他のスレッドによって行われた呼び出しはブロックされます。さらに、プロセス内のすべてのスレッドがコミット有効範囲を共有します。本当の意味でのデータベースへの並行アクセスは、独立したプロセスによって、あるいはこのセクションで説明されている API を用いることによってのみ達成できます。

このセクションでは、データベース API および組み込み SQL を使用するための独立した環境 (コンテキスト) を割り振り、操作するのに使用できる API について説明します。各コンテキストは独立したエンティティであり、あるコンテキストを使用した接続またはアタッチは、それ以外のすべてのコンテキスト (したがって、プロセス内の他のすべての接続あるいはアタッチ) から独立しています。あるコンテキストで作業が行われるためには、まず、そのコンテキストがスレッドに関連付けられなければなりません。スレッドは、データベース API 呼び出しを行うとき、または組み込み SQL を使用する際には、必ずコンテキストを所有していなければなりません。

DB2 バージョン 8 では、すべてのバージョン 8 アプリケーションはデフォルトでマルチスレッドになり、複数コンテキストの使用が可能です。(バージョン 8 以前のアプリケーションの動作は、変更ありません。) 必要な場合は、以下の DB2 API を使用して、複数コンテキストを使用できます。具体的に言えば、アプリケーションはスレッド用のコンテキストを作成でき、各スレッド用の別個のコンテキストにアタッチ、またはデタッチができ、スレッド間でコンテキストを渡すことができます。アプリケーションがこれらの API のいずれも呼び出さない場合は、DB2 が自動的にアプリケーション用に複数コンテキストを管理します。

- `sqlAttachToCtx` - コンテキストへの接続
- `sqlBeginCtx` - アプリケーション・コンテキストの作成および接続
- `sqlDetachFromCtx` - コンテキストからの切り離し
- `sqlEndCtx` - アプリケーション・コンテキストの切り離しおよび破棄
- `sqlGetCurrentCtx` - 現行コンテキストの入手
- `sqlInterruptCtx` - コンテキストへの割り込み

接続またはアタッチの期間中、コンテキストはある特定のスレッドに関連付けられていなくても構いません。あるスレッドがコンテキストにアタッチし、データベースに接続し、そのコンテキストからデタッチされたら、2 番目のスレッドがそのコンテキストにアタッチし、既存のデータベース接続を用いて作業を継続することが可能です。コンテキストは 1 つのプロセス内のスレッド間で受け渡しすることはできますが、複数のプロセス間で受け渡しすることはできません。

新しい API を使用する場合でも、以下の API は引き続きシリアライズされます。

- `sqlabndx` - バインド
- `sqlaprep` - プログラムのプリコンパイル
- `sqluexpr` - エクスポート
- `sqluimpr` - インポート

以下の API は、アプリケーションのスレッド化をサポートしないプラットフォームでは無効です (つまり、操作を行いません)。

注:

1. DB2 CLI は自動的に複数のコンテキストを使用して、マルチスレッドをサポートするプラットフォームで、スレッド・セーフの並行データベース・アクセスを実現します。DB2 では推奨されていませんが、必要な場合には、この機能を明示的に使用不可にできます。
2. デフォルトでは、AIX では 32 ビットのアプリケーションがプロセスあたり 11 を超える共有メモリー・セグメントにアタッチすることは許可されていません。その内、最大で 10 の共有メモリー・セグメントが DB2 接続に使用できます。しかし、Java アプリケーションでは、多くのアプリケーションはプロセスごとに 1 つの共有メモリー・セグメントに制限されています。

この制限に到達すると、DB2 は `SQL CONNECT` で `SQLCODE -1224` を戻します。DB2 Connect では、ローカル・ユーザーが SNA を介して 2 フェーズ・コミットを実行している場合や、TP モニター (SNA または TCP/IP) との 2 フェーズ・コミットを実行している場合にも、接続数は 10 に制限されます。

AIX 環境変数 **EXTSHM** を使用すれば、プロセスをアタッチする共有メモリー・セグメントの最大数を増やすことができます。

EXTSHM を DB2 で使用するには、以下を実行してください。

クライアント・セッションの場合:

```
export EXTSHM=ON
```

DB2 サーバーを始動する場合:

```
export EXTSHM=ON
db2set DB2ENVLIST=EXTSHM
db2start
```

ESE では、sqllib/db2profile に以下の行を追加します。

```
EXTSHM=ON
export EXTSHM
```

ローカル・データベースまたは DB2 Connect を別のマシンに移動してそれにリモートからアクセスするか、TCP/IP ループバックによってローカル・データベースまたは DB2 Connect データベースにアクセスできるようにするため、ローカル・マシンの TCP/IP アドレスを持つリモート・ノードとしてそのデータベースをカタログするのが代替の手段です。

関連資料:

- 1021 ページの『アプリケーションの移行時の考慮事項』
- 1028 ページの『変更された API およびデータ構造』

関連サンプル:

- 『dbthdrs.sqc -- How to use multiple context APIs on UNIX (C)』
- 『dbthdrs.sqc -- How to use multiple context APIs on UNIX (C++)』

db2ReadLogNoConn - データベース接続なしのログの読み取り

ログ・レコードを DB2 UDB データベース・ログから抽出し、現在のログ状態の情報をログ・マネージャーに照会します。この API の使用に先立ち、db2ReadLogNoConnInit を使用して、この API に入力パラメーターとして渡されるメモリーを割り振ります。この API の使用後は、db2ReadLogNoConnTerm を使用して、メモリーを割り振り解除します。

許可:

なし

必要な接続:

なし

API 組み込みファイル:

db2ApiDf.h

C API 構文:

```

/* File: db2ApiDf.h */
/* API: db2ReadLogNoConn */
/* ... */
SQL_API_RC SQL_API_FN
db2ReadLogNoConn (
    db2UInt32 versionNumber,
    void *pDB2ReadLogNoConnStruct,
    struct sqlca *pSqlca);

typedef SQL_STRUCTURE db2ReadLogNoConnStruct
{
    db2UInt32                iCallerAction;
    SQLU_LSN                 *piStartLSN;
    SQLU_LSN                 *piEndLSN;
    char                     *poLogBuffer;
    db2UInt32                iLogBufferSize;
    char                     *piReadLogMemPtr;
    db2ReadLogNoConnInfoStruct *poReadLogInfo;
} db2ReadLogNoConnStruct;

typedef SQL_STRUCTURE db2ReadLogNoConnInfoStruct
{
    SQLU_LSN                 firstAvailableLSN;
    SQLU_LSN                 firstReadLSN;
    SQLU_LSN                 nextStartLSN;
    db2UInt32                logRecsWritten;
    db2UInt32                logBytesWritten;
    db2UInt32                lastLogFullyRead;
    db2TimeOfLog             currentTimeValue;
} db2ReadLogNoConnInfoStruct;

/* ... */

```

API パラメーター:**バージョン番号**

入力。 2 番目のパラメーター *pDB2ReadLogNoConnStruct* として渡される構造のバージョンとリリースのレベルを指定します。

pDB2ReadLogNoConnStruct

入力。 *db2ReadLogNoConnStruct* 構造を指すポインター。

pSqlca 出力。 *sqlca* 構造へのポインター。

iCallerAction

入力。実行するアクションを指定します。有効な値は以下のとおりです。

DB2READLOG_READ

開始ログ・シーケンス番号から終了ログ・シーケンス番号までのデータベース・ログを読み取り、この範囲内にあるログ・レコードを戻します。

db2ReadLogNoConn - データベース接続なしのログの読み取り

DB2READLOG_READ_SINGLE

開始ログ・シーケンス番号によって識別される単一のログ・レコード (伝搬可能または伝搬不可のいずれでも) を読み取ります。

DB2READLOG_QUERY

データベース・ログを照会します。照会した結果は、`db2ReadLogNoConnInfoStruct` 構造を介して戻されます。

piStartLSN

入力。開始ログ・シーケンス番号は、ログの読み取りを開始する相対バイト・アドレスを指定します。この値は、実際のログ・レコードの始まりでなければなりません。

piEndLSN

入力。終了ログ・シーケンス番号は、ログの読み取りを終了する相対バイト・アドレスを指定します。この値は、`piStartLsn` の値より大きくなければなりません。実際のログ・レコードの終わりである必要はありません。

poLogBuffer

出力。指定した範囲内で読み取られた、伝搬可能なすべてのログ・レコードが順番に格納されるバッファ。このバッファは、単一のログ・レコードを保持するのに十分な大きさでなければなりません。目安として、このバッファは最低限 32 バイトでなければなりません。最大サイズは、要求された範囲のサイズによって異なってきます。バッファ内の各ログ・レコードには、接頭部として 6 バイトのログ・シーケンス番号 (LSN) が付けられます。これは、以下のログ・レコードの LSN を示します。

iLogBufferSize

入力。バイト単位でログ・バッファのサイズを指定します。

piReadLogMemPtr

入力。初期設定呼び出しで割り振られた、サイズ `iReadLogMemoryLimit` のメモリー・ブロック。このメモリーには、呼び出しのたびに API が必要とする永続データが含まれています。このメモリー・ブロックは、どのような方法であっても呼び出し側は再割り振りまたは変更してはなりません。

poReadLogInfo

出力。 `db2ReadLogNoConnInfoStruct` 構造を指すポインター。

firstAvailableLSN

使用可能なログ内で最初の使用可能な LSN。

firstReadLSN

この呼び出しでの最初の LSN 読み取り。

nextStartLSN

次の読み取り可能な LSN。

logRecsWritten

ログ・バッファ・フィールド *poLogBuffer* に書き込まれるログ・レコードの数。

logBytesWritten

ログ・バッファ・フィールド *poLogBuffer* に書き込まれるログ・バッファ・フィールドのバイト数。

lastLogFullyRead

読み取られて完了する最後のログ・ファイルを示す数。

使用上の注意:

db2ReadLogNoConn API は、db2ReadLogNoConnInit API を使用して割り振られるメモリー・ブロックを必要とします。メモリー・ブロックは、入力パラメーターとして、続くすべての db2ReadLogNoConn API に渡されなければならず、変更してはなりません。

ログの順次読み取りを要求する場合、API はログ・シーケンス番号 (LSN) の範囲および割り振られたメモリーを必要とします。API は、初期設定されたときに指定されたフィルター・オプションおよび LSN 範囲に基づいて、ログ・レコードの順序を戻します。照会を要求する場合、ログ情報の読み取り構造に、読み取りの呼び出しで使用される有効な開始 LSN が入ります。読み取りでの終了 LSN として使用される値は、以下のうちの 1 つになります。

- 呼び出し側が指定した startLSN の値より大きい値。
- 非同期ログ読み取りプログラムで、使用可能なログの終わりとして解釈される FFFF FFFF。

開始および終了 LSN の範囲内で読み取られた伝搬可能なログ・レコードは、ログ・バッファに戻されます。ログ・レコードには、その LSN は含まれません。LSN は、バッファの中で、実際のログ・レコードの前に付けられます。

db2ReadLogNoConn によって戻されるさまざまな DB2 UDB ログ・レコードの詳細については、『DB2 UDB ログ・レコード』セクションで説明しています。

最初の読み取りの後、次の順番のログ・レコードを読み取るには、db2ReadLogNoConnInfoStruct で戻された nextStartLSN 値を使用します。この新しい開始 LSN と有効な終了 LSN を使用して呼び出しをもう一度実行依頼すると、次のレコード・ブロックが読み取られます。SQLU_RLOG_READ_TO_CURRENT の sqlca コードは、使用可能なログ・ファイルが最後まで読み取られたことを示します。

この API がもう使用されない場合、db2ReadLogNoConnTerm を使用してメモリーを終了します。

関連資料:

- 管理 API リファレンスの『SQLCA』

db2ReadLogNoConn - データベース接続なしのログの読み取り

- 1005 ページの『db2ReadLogNoConnInit - データベース接続なしのログ読み取りの初期設定』
- 420 ページの『db2ReadLogNoConnTerm - データベース接続なしのログ読み取りの終了』

変更された API およびデータ構造

表 221. バックレベルがサポートされた API およびデータ構造

API またはデータ構造 (バージョン)	記述名	新しい API またはデータ構造 (バージョン)
sqlbfstsq (V2)	表スペース照会の取り出し	sqlbfstpq (V5)
sqlbstsq (V2)	単一の表スペース照会	sqlbstpq (V5)
sqlbstsq (V2)	表スペース照会	sqlbmtsq (V5)
sqlcctdd (V2)	データベースのカタログ	sqlcctadb (V5)
sqlcstart (V5)	データベース・マネージャーの始動	db2InstanceStart (V8)
sqlcstp (V5)	データベース・マネージャーの開始	db2InstanceStop (V8)
sqlcstpr (V2)	データベース・マネージャーの開始 (DB2 パラレル・エディション バージョン 1.2)	db2InstanceStart (V8)
sqlcstar (V2)	データベース・マネージャーの開始 (DB2 バージョン 2)	db2InstanceStart (V8)
sqlcstop (V2)	データベース・マネージャーの開始	db2InstanceStop (V8)
sqlcsetTypeCtx (V7)	アプリケーション・コンテキスト・タイプの設定	N/A すべてのアプリケーションは、デフォルトでマルチスレッドになり、複数コンテキストを使用できません (v8)。
sqlcstnd (V5)	データベースの再始動	db2DatabaseRestart (V6)
sqlcfddb (V7)	データベース構成のデフォルトの入手	db2CfgGet (V8)
sqlcfdsys (V7)	データベース・マネージャー構成のデフォルトの入手	db2CfgGet (V8)
sqlcfrdb (V7)	データベース構成のリセット	db2CfgSet (V8)
sqlcfrsys (V7)	データベース・マネージャー構成のリセット	db2CfgSet (V8)
sqlcfudb (V7)	データベース構成の更新	db2CfgSet (V8)
sqlcfusys (V7)	データベース・マネージャー構成の更新	db2CfgSet (V8)
sqlcfxdb (V7)	データベース構成の入手	db2CfgGet (V8)
sqlcfxsys (V7)	データベース構成の入手	db2CfgGet (V8)
sqlcmon (V6)	モニター・スイッチの入手 / 更新	db2MonitorSwitches (V7)
sqlcmonss (V5)	スナップショットの入手	db2GetSnapshot (V6)

db2ReadLogNoConn - データベース接続なしのログの読み取り

表 221. バックレベルがサポートされた API およびデータ構造 (続き)

API またはデータ構造 (バージョン)	記述名	新しい API またはデータ構造 (バージョン)
sqlmonsz (V6)	sqlmonss() 出力バッファに必要なサイズの見積もり	db2GetSnapshotSize (V7)
sqlmrset (V6)	モニターのリセット	db2ResetMonitor (V7)
sqlubkp (V5)	データベースのバックアップ	db2Backup (V8)
sqlubkup (V2)	データベースのバックアップ	db2Backup (V8)
sqlugrpi (V2)	行のパーティション情報の入手 (DB2 パラレル・エディション バージョン 1.x)	sqlugrpn (V5)
sqluhcls (V5)	リカバリー履歴ファイルのスキャンのクローズ	db2HistoryCloseScan (V6)
sqluhget (V5)	履歴ファイルからの DDL 情報の検索	db2HistoryGetEntry (V6)
sqluhgne (V5)	リカバリー履歴ファイルの次項目の入手	db2HistoryGetEntry (V6)
sqluhops (V5)	リカバリー履歴ファイルのスキャンのオープン	db2HistoryOpenScan (V6)
sqluhprn (V5)	リカバリー履歴ファイルの整理	db2Prune (V6)
sqluhupd (V5)	リカバリー履歴ファイルの更新	db2HistoryUpdate (V6)
sqluload (V7)	ロード	db2Load (V8)
sqluqry (V5)	ロードの照会	db2LoadQuery (V6)
sqlureot (V7)	表の再編成	db2Reorg (V8)
sqlurestore (V7)	RESTORE DATABASE	db2Restore (V8)
sqlurlog (V7)	ログの非同期読み取り	db2ReadLog (V8)
sqluroll (V7)	データベースのロールフォワード	db2Rollforward (V8)
sqlursto (V2)	RESTORE DATABASE	sqlurst (V5)
sqlustat (V7)	統計の実行	db2Runstats (V8)
sqlxhcom (V2)	未確定トランザクションのコミット	sqlxphcm (V5)
sqlxhqry (V2)	未確定トランザクションのリスト	sqlxphqr (V5)
sqlxhrol (V2)	未確定トランザクションのロールバック	sqlxphrl (V5)
sqlxphqr (V7)	未確定トランザクションのリスト	db2XaListIndTrans (V8)
SQLB-TBSQRY-DATA (V2)	表スペース・データ構造	SQLB-TBSPQRY-DATA (V5)
SQLE-START-OPTIONS (V7)	データベース・マネージャーのデータ構成の始動	db2StartOptionsStruct (V8)
SQLEDBSTOPOPT (V7)	データベース・マネージャーのデータ構成の始動	db2StopOptionsStruct (V8)
SQLEDBSTRTOPT (V2)	データベース・マネージャーの開始データ構造 (DB2 パラレル・エディション バージョン 1.2)	db2StartOptionsStruct (V8)

db2ReadLogNoConn - データベース接続なしのログの読み取り

表 221. バックレベルがサポートされた API およびデータ構造 (続き)

API またはデータ構造 (バージョン)	記述名	新しい API またはデータ構造 (バージョン)
SQLUHINFO and SQLUHADM (V5)	履歴ファイルのデータ構造	db2HistData (V6)
SQLULOAD-IN (V7)	入力構成のロード	db2LoadIn (V8)
SQLULOAD-OUT (V7)	出力構成のロード	db2LoadOut (V8)
SQLXA-RECOVER (V7)	トランザクション API 構成	db2XaRecoverStruct

表 222. バックレベル・サポートが存在しない API

名前	記述名	V8 でサポートされている API
sqlufrol/sqlgfrol	データベースのロールフォワード (DB2 バージョン 1.1)	db2Rollforward
sqluprfl	データベースのロールフォワード (DB2 パラレル・エディション バージョン 1.x)	db2Rollforward
sqlurflf/sqlgrflf	データベースのロールフォワード (DB2 バージョン 1.2)	db2Rollforward
sqlurllf/sqlgrllf	データベースのロールフォワード (DB2 バージョン 2)	db2Rollforward

関連資料:

- 1021 ページの『アプリケーションの移行時の考慮事項』

DB2 機能の拡張

Data Links Manager 管理ガイドおよび解説書

ファイル・システム・ディレクトリー階層を現時点の状態まで戻す

ファイル・システムのリストア操作の後、必要に応じて、ディレクトリーを再作成することによって、手作業でファイル・システム・ディレクトリー階層を現時点の状態にする必要があります。

再作成する必要があるディレクトリーを特定するには、Data Links Manager の `fsysadm.log` ファイルを使用します。データは常に `fsysadm.log` に追加されます。

AIX® と **Solaris** オペレーティング環境では、ディレクトリーの変更は `INSTHOME/sqllib/fsysadm.log` ファイルにログされます。ここで、`INSTHOME` は、Data Links Manager 管理者のホーム・ディレクトリーです。イベントごとに 1 項目ずつ追加されていきます。また、ファイルの属性の設定も記録されます。`fsysadm.log` ファイルの項目のフォーマットは、以下のとおりです。

```
Time = <timestamp> EUID = <integer> UID = <integer> GID = <integer> Mode = <octal>  
Action = <CREATE/REMOVE/SETATTR/RENAME> Object type = <DIR/FILE> Path = <fully qualified  
source name, destination name>
```

各パラメーターの意味は以下のとおりです。

- *Time* は、その活動のローカルでの時刻です。
- *EUID* は、その処理を実行したユーザーの有効ユーザー ID です。
- *UID* は、作成された、あるいは属性が変更されたファイルまたはディレクトリーのユーザー ID 属性です。
- *GID* は、作成された、あるいは属性が変更されたファイルまたはディレクトリーのグループ ID 属性です。
- *Mode* は、ファイルまたはディレクトリーのモードの 8 進表示です。

上記の Action は以下のいずれかになります。

- *CREATE* は、ディレクトリーが作成されたことを示しています。
- *REMOVE* は、ファイルまたはディレクトリーが除去されたことを示しています。
- *SETATTR* は、ファイルまたはディレクトリーのモードがユーザーによって変更されたことを示しています。
- *RENAME* は、ファイルが名前変更されたことを示しています。

上記の Object type は以下のいずれかになります。

- *DIR* は、ディレクトリーを示します。
- *FILE* は、ファイルを示します。

加えて、上記の Path はファイルまたはディレクトリーの完全修飾パスです。アクションが RENAME であった場合、宛先名はパス情報の後に表示されます。

Windows® システムでは、ディレクトリー変更は `x:\%sqllib%\dlfm\fsysadm.log` ファイルに記録されます。ここで、*x:* は、DB2® Data Links Manager をインストールしたドライブを表します。指定したファイルまたはディレクトリーについて、アクセス・コントロール・リストをもつユーザーまたはユーザー・グループの数によって、単一イベントが複数の項目をもつ場合があります。fsysadm.log ファイルの最初の項目のフォーマットは、以下のとおりです。

```
Time = <timestamp> User = <string> Action = <CREATE/REMOVE/SETATTR/RENAME>
Object type = <DIR/FILE> Path = <fully qualified source name, destination name>
```

最初の項目に関連した追加項目のフォーマットは、以下のとおりです。

```
ACE User = <string> Access = <Hex integer> ACE Type = <Hex integer>
ACE Flags = <Hex integer>
```

各パラメーターの意味は以下のとおりです。

- *Time* は、その活動のローカルでの時刻です。
- *User* は、その処置を実行したユーザーの名前です。
- *Owner* は、そのファイルまたはディレクトリーの所有者の名前です。
- *Path* は、ファイルまたはディレクトリーの完全修飾パスです。
- *ACE User* は、このファイルまたはディレクトリーの ACL 項目を持つユーザーの名前です。
- *Access** は、ユーザーが持つアクセス・タイプを示すフラグのセットです。
- *ACE Type** は、ACE のタイプです (たとえば、allow/deny など)。
- *ACE Flags** は、ACE タイプに固有のコントロール・フラグ・セットです。

上記の Action は以下のいずれかになります。

- *CREATE* は、ディレクトリーが作成されたことを示しています。
- *REMOVE* は、ファイルまたはディレクトリーが除去されたことを示しています。
- *SETATTR* は、ファイルまたはディレクトリーのモードがユーザーによって変更されたことを示しています。
- *RENAME* は、ファイルが名前変更されたことを示しています。

上記の Object type は以下のいずれかになります。

- *DIR* は、ディレクトリーを示します。

- *FILE* は、ファイルを示します。

加えて、上記の *Path* はファイルまたはディレクトリーの完全修飾パスです。アクションが *RENAME* であった場合、宛先名はパス情報の後に表示されます。

(*) これらの 16 進値の定義については、Windows NT® と Windows 2000 の Microsoft® SDK に関するドキュメンテーションのアクセス・コントロール項目の構造を参照してください。

関連概念:

- *DB2 Data Links Manager* 管理ガイドおよびリファレンス の『障害とリカバリーの概要』

データ・リンク・ファイル・マネージャーのエラー・メッセージの解説

ここでは、Windows NT、Windows 2000、AIX、Solaris オペレーティング環境でデータ・リンク・ファイル・マネージャーを使用するときに表示されるメッセージとエラーについて説明します。各メッセージ・コードやエラー・コードには、対応するメッセージ、考えられる原因、推奨アクションが付いています。

Windows NT および Windows 2000 のシステム上では、エラー・メッセージは、*x:\sql\lib\instance\db2diag.log* ファイルに送られます。ここで、*x:* は、DB2 Data Links Manager をインストールしたディレクトリーで、*instance* は、Data Links Manager 管理者が所有するインスタンスです (デフォルトでは *dlfm*)。エラー・メッセージは、**dlfm** コマンドを実行した人にも表示されます。

AIX および Solaris オペレーティング環境システムでは、エラー・メッセージは、*INSTHOME/sql/lib/db2dump/db2diag.log* ファイルに送られます。ここで、*INSTHOME* は、インスタンス所有者のホーム・ディレクトリーです。エラー・メッセージは、**dlfm** コマンドを実行した人にも表示されます。

ここでは、以下のメッセージについて説明します。

- 1036 ページの『DLFM001I』
- 1036 ページの『DLFM002I』
- 1036 ページの『DLFM003I』
- 1036 ページの『DLFM101E』
- 1037 ページの『DLFM102E』
- 1037 ページの『DLFM103W』
- 1038 ページの『DLFM104W』
- 1038 ページの『DLFM121E』
- 1039 ページの『DLFM122E』

- 1039 ページの『DLFM123E』
- 1039 ページの『DLFM124E』
- 1039 ページの『DLFM126E』
- 1040 ページの『DLFM128E』
- 1040 ページの『DLFM129I』
- 1040 ページの『DLFM151I』
- 1040 ページの『DLFM201E』
- 1041 ページの『DLFM202E』
- 1041 ページの『DLFM203E』
- 1041 ページの『DLFM204E』
- 1041 ページの『DLFM205E』
- 1042 ページの『DLFM206E』
- 1042 ページの『DLFM207E』
- 1042 ページの『DLFM208I』
- 1043 ページの『DLFM209E』
- 1043 ページの『DLFM210E』
- 1043 ページの『DLFM211E』
- 1044 ページの『DLFM212I』
- 1044 ページの『DLFM215E』
- 1044 ページの『DLFM217I』
- 1044 ページの『DLFM221E』
- 1045 ページの『DLFM222I』
- 1045 ページの『DLFM223E』
- 1045 ページの『DLFM224I』
- 1045 ページの『DLFM225I』
- 1046 ページの『DLFM252E』
- 1046 ページの『DLFM254I』
- 1046 ページの『DLFM255E』
- 1047 ページの『DLFM256E』
- 1047 ページの『DLFM341E』
- 1048 ページの『DLFM402E』
- 1048 ページの『DLFM501E』
- 1056 ページの『DLFM701E』
- 1056 ページの『DLFM703E』
- 1056 ページの『DLFM704E』

- 1056 ページの『DLFM706W』
- 1057 ページの『DLFM707I』
- 1057 ページの『DLFM721E』
- 1057 ページの『DLFM741E』
- 1057 ページの『DLFM742E』
- 1058 ページの『DLFM743E』
- 1058 ページの『DLFM744E』
- 1059 ページの『DLFM746E』
- 1059 ページの『DLFM747E』
- 1059 ページの『DLFM748I』
- 1059 ページの『DLFM749E』
- 1060 ページの『DLFM750I』
- 1060 ページの『DLFM751E』
- 1060 ページの『DLFM801I』
- 1061 ページの『DLFM802I』
- 1061 ページの『DLFM803I』
- 1061 ページの『DLFM804I』
- 1061 ページの『DLFM805I』
- 1062 ページの『DLFM806I』
- 1062 ページの『DLFM807I』
- 1062 ページの『DLFM808I』
- 1063 ページの『DLFM809I』
- 1063 ページの『DLFM810I』
- 1063 ページの『DLFM811E』
- 1064 ページの『DLFM812I』
- 1064 ページの『DLFM813I』
- 1064 ページの『DLFM814I』
- 1064 ページの『DLFM815E』
- 1065 ページの『DLFM816W』
- 1066 ページの『DLFM817E』
- 1066 ページの『DLFM818E』
- 1067 ページの『DLFM819I』
- 1067 ページの『DLFM820E』
- 1068 ページの『DLFM821I』
- 1068 ページの『DLFM822E』

- 1069 ページの『DLFM823I』
- 1069 ページの『DLFM841E』
- 1069 ページの『DLFM842I』
- 1070 ページの『DLFM843E』
- 1070 ページの『DLFM844I』
- 1070 ページの『DLFM845I』
- 1071 ページの『DLFM846I』
- 1071 ページの『DLFM847I』
- 1071 ページの『DLFM848I』
- 1071 ページの『DLFM853E』
- 1072 ページの『DLFM900I』
- 1072 ページの『DLFM901E』
- 1072 ページの『DLFM908E』
- 1073 ページの『DLFM909E』

DLFM001I

DLFM サーバーが開始されました。

原因 : データ・リンク・ファイル・マネージャーが正常に開始されました。

処置 : 必要ありません。

DLFM002I

DB2 Data Links Manager が開始中です ...

原因 : DLFM サーバーが、初期化プロセスを開始中です。

処置 : DLFM サーバーの開始が完了することを示すメッセージが表示されるまで、お待ちください。

DLFM003I

DB2 Data Links Manager が停止中です ...

原因 : DLFM サーバーが、終了プロセスを開始中です。

処置 : DLFM サーバーの終了が完了することを示すメッセージが表示されるまで、お待ちください。

DLFM101E

DLFM サーバーの開始でエラーが発生しました。理由コード = "<reason-code>"

理由コード -3:

データ・リンク・ファイル・マネージャーの初期化が失敗しました。

- 原因 :** 以下の理由が考えられます。
1. トレースの初期化ではエラーは発生しませんでした。
 2. 共通サブシステムの初期化ではエラーは発生しませんでした。
 3. DB2 ログ・マネージャーの初期化ではエラーは発生しませんでした。
 4. 共用リソースの作成ではエラーは発生しませんでした。
 5. モニター・スレッドの作成ではエラーは発生しませんでした。
 6. sqlwait コードまたは sqlopost コードにエラーはありませんでした。
 7. DLFM が正常に開始される前に停止要求がありました。
 8. Initdone 状態に移る際にエラーが発生しました。

- 処置 :** 以下のステップを実行してください。
1. DB2 Data Links Manager 管理者としてシステムにログオンします。
 2. **dlfm stop** コマンドを入力して、データ・リンク・ファイル・マネージャーを停止します。
 3. **dlfm shutdown** コマンドを入力して、Data Links サーバーをシャットダウンします。
 4. **dlfm start** コマンドを入力して、Data Links Manager を開始します。
 5. 問題が継続する場合は、db2diag.log のエラー情報を参照してください。
 6. システム管理者に連絡してください。

DLFM102E

DLFM サーバーのシャットダウンでエラーが起きました。理由コード = "`<reason-code>`"

理由コード -3:

原因 : DLFM を正常に終了できませんでした。

処置 : 追加の詳細および説明については、db2diag.log ファイルを参照してください。

DLFM103W

DLFM サーバーは既に開始されています。

原因 : Data Links Manager 管理者が DLFM サーバーを開始しようとしたが、DLFM サーバーは既に稼働中です。

処置 :

1. DLFM プロセスの完全セットが実行中かどうかを確認してください。
 - UNIX では、**dlfm see** コマンドを使用します。
 - Windows では、Windows タスク マネージャを使用します。
2. DLFM プロセスの完全セットがリストされていない場合は、**dlfm restart** コマンドで DLFM を再始動してください。
3. 問題が続く場合は、追加の詳細および説明について db2diag.log ファイルを参照してください。

DLFM104W

DLFM がまだ開始されていません。

原因 : Data Links Manager 管理者が DLFM サーバーを停止しようとしたが、DLFM サーバーが実行されていません。

処置 :

1. 1、2 分待つてから、DLFM サーバーが完全に初期化または終了されたことを確認してください。
2. DLFM プロセスの完全セットが実行中かどうかを確認してください。
 - UNIX では、**dlfm see** コマンドを使用します。
 - Windows では、Windows タスク マネージャを使用します。
3. DLFM プロセスがリストにない場合は、DLFM サーバーは終了しているため、アクションは必要ありません。
4. DLFM プロセスが実行中であることを確認した場合は、**dlfm stop** コマンドを使用して、もう一度 DLFM サーバーの停止を試行してください。
5. **dlfm stop** コマンドが失敗した場合は、**dlfm shutdown** コマンドを実行して、追加のクリーンアップ処理を行ってください。
6. 問題が続く場合は、追加の詳細および説明について db2diag.log ファイルを参照してください。

DLFM121E

接頭部名 "<prefix-name>" は既に登録されています。

原因 : 指定した *prefix-name* は、既にこのデータ・リンク・ファイル・マネージャで定義されています。

処置 : *prefix-name* パラメーターの正しい値を指定してください。現在登録されている接頭部のリストを表示するには、**dlfm list registered prefixes** コマンドを使用します。

DLFM122E

接頭部名 "<prefix-name>" は登録することができません。指定した共有ドライブは、既に別の接頭部名で登録されています。

このエラーは Windows 環境でのみ発生します。

原因 : 指定した *prefix-name* で指定された共用名は、既に別の接頭部名で、このデータ・リンク・ファイル・マネージャーに登録されています。

処置 : 異なる共有ドライブ用の共用名を、登録されている *prefix-name* で指定してください。現在登録されている接頭部のリストを表示するには、**dlfm list registered prefixes** コマンドを使用します。

DLFM123E

接頭部名 "<prefix-name>" ("*<directory-name>*") が、DLFS マウント・ポイントまたはボリューム名に対応していません。

原因 : 新規接頭部の追加の過程で、接頭部パスについての問題が検出されました。<directory-name> は、問題のあるディレクトリーを示していません。

- UNIX では、基本マウント・ポイントまたはボリュームが、データ・リンク・ファイル・システム (DLFS) として定義されていません。
- Windows では、接頭部 (共有名) が DLFS として定義されたシステム・ドライブにマップしていません。

処置 : DLFS ドライブまたはマウント・ポイントにマップする接頭部を指定してください。

DLFM124E

接頭部名 "<prefix-name>" は共有ドライブとして登録されていません。

このエラーは Windows 環境でのみ発生します。

原因 : 指定した *prefix-name* で指定された共用名は、このシステムで定義された共有ドライブに対応していません。

処置 : 異なる共有ドライブ用の共用名を、登録されている *prefix-name* で指定してください。接頭部を登録したい共有ドライブについて、Windows に定義されている共用名をチェックしてください。

DLFM126E

指定した接頭部名 "<prefix-name>" は、正しくないか、登録されていません。戻りコード = "<return-code>"

原因 : 指定した *prefix-name* は、このデータ・リンク・ファイル・マネージャーに登録されていません。

処置 : 正しい *prefix-name* パラメーター値を指定してください。現在登録されている接頭部のリストを表示するには、**dlfm list registered prefixes** コマンドを使用します。

DLFM128E

要求された管理機能を実行できません。戻りコード = "<return-code>"

原因 : 要求された管理機能は失敗しました。*return-code* は、失敗に関連した内部 SQL コードを表します。

処置 : 以下のステップを実行してください。

1. db2diag.log にあるエラー情報を参照してください。
2. そこに説明されている修正処置にしたがってください。
3. 問題が継続する場合は、IBM サービスに報告してください。

DLFM129I

DLFM_DB <db-scope> の自動バックアップが起動されました。バックアップ完了までお待ちください。

原因 : データ・リンク・ファイル・マネージャーは、そこに含まれる DLFM_DB のデータベースまたは表スペースのバックアップを開始しました。バックアップが行われるのは、最後に入力した **dlfm** コマンドによって記録された重要な情報を保存するためです。

処置 : バックアップを中断することなく完了させてください。データ・リンク・ファイル・マネージャーが保守するファイルの情報量によって異なりますが、この処理が完了するには、数分、またはそれ以上かかる場合があります。

DLFM151I

以下のファイルが更新されています。

原因 : WRITE PERMISSION ADMIN 属性をもった DATALINK 列にリンクされた、更新中のファイルのセットが、このメッセージに続いてリストされます。書き込みトークンを使用して更新のために開いているファイルのみが表示されます。

処置 : アクションは必要ありません。ただし、ファイル更新の継続や完了に必要な書き込みトークンの置き場所を誤った場合は、ファイル名と一緒にリストされるトークンをメモにとっておくことができます。

DLFM201E

DLFM 登録サービスでエラーが発生しました。理由コード = "<reason-code>"

理由コード	意味
-2	データベース登録でエラーが起きました。

-4	接頭部登録でエラーが起きました。
-6	レプリケーション権限登録でエラーが発生しました。
-8	権限登録の書き込み、リンクのいずれか、または両方でエラーが発生しました。

原因 : データ・リンク・ファイル・マネージャーのログ・マネージャーでエラーが起きました。

処置 : 以下のステップを実行してください。

1. db2diag.log にあるエラー情報を参照してください。
2. 問題を IBM サービスに報告してください。

ファイル・システムの登録の詳細については、このトピックの終わりにある関連リンクを参照してください。

DLFM202E

指定したデータベース・サーバーの登録が無効です。

原因 : コマンドで指定した、*database_name*、*instance_name*、または *node_name* (あるいはその全部) のパラメーターが無効です。

処置 : *database_name*、*instance_name*、および *node_name* の各パラメーターの正しい値を指定してください。現在登録されているデータベース・サーバーのリストを表示するには、**dlfm list registered databases** コマンドを使用します。

DLFM203E

指定したディレクトリーが無効です。

原因 : コマンドで指定した *directory* パラメーターが無効です。

処置 : 有効なディレクトリー名を指定してください。

DLFM204E

指定したディレクトリーは、登録済み接頭部にはありません。

原因 : コマンドで指定した *directory* パラメーターは、登録済み接頭部にはありません。

処置 : *directory* パラメーターの正しい値を指定してください。現在登録されているデータベース接頭部のリストを表示するには、**dlfm list registered prefixes** コマンドを使用します。

DLFM205E

指定したディレクトリー・アクセス・コントロールは既に定義されています。

原因 : 指定した *directory*、*db2_authorization_id*、および *database_server* の各パラメーターについてのアクセス・コントロールは、既に定義されています。

処置 : コマンドを再実行する前に、指定したこれらのパラメーターの既存のアクセス・コントロール項目を表示するために、**dlfm list registered directories** コマンドを使用してください。

DLFM206E

指定したディレクトリー・アクセス・コントロールが見つかりません。

原因 : 指定した *directory*、*db2_authorization_id*、および *database_server* の各パラメーターについてのアクセス・コントロールが登録されていません。

処置 : パラメーターの正しい値を指定してください。

DLFM207E

指定した接頭部が無効です。

原因 : コマンドで指定した *prefix* パラメーターは、登録済み接頭部ではありません。

処置 : 正しい *prefix* パラメーター値を指定してください。現在登録されている接頭部を表示するには、**dlfm list registered prefixes** コマンドを使用します。必要な場合は、**dlfm add prefix** コマンドを使用して、必要な接頭部を登録します。

DLFM208I

"<user-set>" に属するすべての既存の DB2 サーバー接続が、データベース (データベース名 "<db-name>"、インスタンス "<instance>"、ノード "<node>") から切断されるまでは、この変更内容は有効になりません。

user-set には、個々のユーザー、ユーザー・グループ、またはすべてのユーザー (つまり、PUBLIC) など、このコマンドによって影響を受けるユーザー (DB2 authids) が記述されています。

原因 : 指定した DB2 サーバーとデータ・リンク・ファイル・マネージャー間の切断を要求するコマンドが、既存の接続が切断されていない間に出された可能性があります。

処置 : 指定したコマンドの変更点をすぐに有効にする必要がある場合は、指定したデータベースに属する指定したユーザーが開始した、このデータ・リンク・ファイル・マネージャーと DB2 サーバー間のすべての接続を終了してください。

DLFM209E

指定したユーザー ID またはグループ ID が無効です。

原因：

- 指定した ID がシステムの有効なユーザー ID またはグループ ID ではありません。
- ユーザーまたはグループが、Data Links Manager 管理者のユーザー ID を参照できません。
- グループ ID は 8 文字以下でなければなりません。

処置：

- システムで ID を作成するか、既に存在する ID を指定してください。
- Data Links Manager 管理者のユーザー ID を参照しないユーザーまたはグループを指定してください。
- グループを指定する場合、名前が 8 文字以下であることを確認してください。

DLFM210E

このユーザーまたはグループについて指定したアクセス・コントロールは既に定義されています。

原因： 指定したユーザー ID またはグループ ID のアクセス・コントロールが既に定義されています。

処置： 必要なアクセス・コントロールが既に定義されていることを確認するためには、**dlfm list registered replication access control** コマンド、または **dlfm list registered users** コマンドのいずれかを使用してください。定義されていない場合は、ユーザー ID、グループ ID、その他の ID を必要なパラメーター値と置き換え、もう一度コマンドを出してください。

DLFM211E

指定したアクセス・コントロールが、このユーザーまたはグループに定義されていません。

原因： 指定したユーザー ID またはグループ ID についてのアクセス・コントロールが見付かりません。

処置： 正しいアクセス・コントロール値を指定してください。登録されているアクセス・コントロール値を表示するには、**dlfm list registered users** コマンド、**dlfm list registered directories** コマンド、または **dlfm list registered replication access control** コマンドを使用することができます。

DLFM212I

この変更は、DLFM サーバーが再始動されるまで有効になりません。

原因： Data Links Manager 管理者が、**dlfm set link security** コマンドを実行しました。

処置： **dlfm restart** コマンドを実行して、DLFM ファイル・リンク・セキュリティ・フィーチャーを、即時に指定されたように変更します。そうしないと、指定された変更は、DLFM サーバーが次に開始された時にのみ有効になります。

DLFM215E

接頭部名 "<prefix-name>" は、有効な既存の完全修飾パスではありません。

原因： 指定されたパスが完全修飾パス (スラッシュで始まるパス) でないか、または入力されたパスが存在しません。

処置： DLFS 内の既存の、完全修飾パスにマップする接頭部名を指定してください。

DLFM217I

この変更は、Data Links Manager レプリケーション・デーモンへの新規の接続用にのみ有効になります。

原因： 指定されたコマンドで行われた変更 (たとえば許可など) は、Data Links Manager レプリケーション・デーモンへの既存の接続には影響しません。

処置： 指定されたコマンド内の変更要求が即時に有効になる必要がある場合、Data Links Manager レプリケーション・デーモンへの接続を終了し、新規の接続を開始してください。そうでない場合は、アクションは必要ありません。

DLFM221E

DROP_DLM の処理が失敗しました。詳細については、db2diag.log ファイルを参照してください。

原因： Data Links Manager が、指定したデータベースに対して、リンク解除やガーベッジ・コレクションの処理を開始できませんでした。以下の理由が考えられます。

- Data Links Manager が実行されていない。
- *database_name*、*instance_name*、または *host_name* (あるいはその全部) のパラメーターの値が誤って指定されている。
- Data Links Manager のいずれかのコンポーネント・サービスが失敗した。

処置： 以下のステップを実行してください。

1. Data Links Manager が実行されていることを確認する。実行されていない場合は、**dlfm start** コマンドを入力して、DLFM を再始動する。
2. *database_name*、*instance_name*、および *host_name* の各パラメータの正しい値を指定する。現在登録されているデータベース・サーバーのリストを表示するには、**dlfm list registered databases** コマンドを使用します。

問題が継続する場合は、失敗したコンポーネント・サービス (接続管理サービス、トランザクション管理サービスなど) があるかどうかを確認するために、db2diag.log の情報を参照してください。db2diag.log のエラー・コードを見て、そのエラー・コードの下に指示されている適切な処置を実行してください。

DLFM222I

"<request-type>" 要求は成功しました。

原因 : 指定した要求 (またはコマンド) は正常に完了しました。

処置 : 必要ありません。

DLFM223E

"<request-type>" 要求は失敗しました。

原因 : 指定した要求またはコマンドは正常に完了しませんでした。

処置 : このメッセージに付いているその他のメッセージとコードを記録して、db2diag.log ファイルの追加エラー情報を参照してください。

DLFM224I

"<request-type>" 要求はキャンセルされました。

原因 : Data Links Manager 管理者が指定されたコマンドまたは要求の続行を確認しなかったため、要求は打ち切られました。

処置 : まだ要求を続けたい場合は、コマンドを再入力してください。そうでない場合は、アクションは必要ありません。

DLFM225I

ホスト名 "<host-name>" 上のインスタンス "<instance-name>" のデータベース "<database-name>" のリンク解除およびガーベッジ・コレクション処理の開始中です。

原因 : Data Links Manager 管理者が呼び出され、**dlfm drop_dlm** コマンドの処理を確認しました。

処置 : アクションは必要ありません。指定されたデータベースへのすべての参照は、DLFM サーバーから除去されます。

DLFM252E

DLFM レジストリー変数 DLFM_INSTALL_PATH ("*<variable-value>*") が、必要な "sqllib" ディレクトリーを指していません。

原因 : DLFM_INSTALL_PATH レジストリー変数は *variable-value* によって示される値をもっていますが、必要な "sqllib" サブディレクトリーを指していません。

処置 : **db2set** コマンドを使用して、DLFM_INSTALL_PATH レジストリー変数にデータ・リンク・ファイル・マネージャーがインストールされている "sqllib" ディレクトリーを含むパスを割り当てます。

DLFM254I

DLFS 構成ファイル "*<file-name>*" が、バックアップ・ファイル "*<backup-file-name>*" からリストアされています。

原因 : DLFS 構成ファイル *file-name* は見付かりませんでしたが、バックアップ・コピー *backup-file-name* は見付かりました。このバックアップ・コピーが、*file-name* 構成ファイルを置き換えるために使用されています。

処置 : 必要ありません。

DLFM255E

"*<module-name>*" : db2diag.log ファイルを初期化できません。戻りコード = "*<return-code>*" DB2INSTANCE および DLFM_INSTANCE_NAME 構成変数が正しく設定されていることを確認してください。

原因 : ファイル db2diag.log を DB2 のキャプチャーおよび DLFM 診断用に開けませんでした。

<module-name> は、このエラーを検出したプログラムの名前です。

処置 :

- データベース構成変数 DB2INSTANCE および DLFM_INSTANCE_NAME が正しく設定されていることを確認してください。
- DB2 データベース・マネージャー構成変数 DIAGPATH が正しく設定されていることを確認してください。
- DFLM データベース・インスタンスが常駐するファイル・システム、または DIAGPATH が示すファイル・システムに、db2diag.log ファイル用に十分なフリー・スペースがあることを確認してください。

問題が継続する場合は、IBM サービスに報告してください。

DLFM256E

DFLM データベース "DLFM_DB" はインストール済みの現行バージョンの Data Links Manager に対応しません。"<version>"

原因 : DLFM データベース DLFM_DB の内容は、実行中の Data Links Manager のバージョンが、DLFM_DB のバージョンと共に使用できないことを示しています。指定コマンドは、この不整合のために実行できませんでした。

処置 :

- 既存の DLFM_DB データベースを、現在インストールされているバージョンの Data Links Manager に一致するように変換する必要があります。この変換を行うには、DLFM データベース移行プログラムを実行してください。
- DLFM データベース移行プログラムを実行し、それが正常に終了しなかった場合は、元の DLFM_DB データベースをリストアし、移行プログラムを正常終了させる必要があります。
- DLFM データベース移行プログラムが正常終了しなかった場合は、以下のステップを実行してください。
 1. db2diag.log ファイルで追加のエラー情報を確認し、そこにある修正アクションに従います。
 2. Data Links Manager 操作 (たとえばファイルのリンク、dlfm コマンド、または DLFM サーバーが DLFM_DB データベースに保管される操作) を実行していない場合は、コマンド **dlfm drop_db**、および **dlfm setup** を実行して、DLFM_DB データベースを再作成できます。

重要: これらのコマンドは、以前のサーバー・アクティビティの記録を、完全に削除します。
 3. 問題が継続する場合は、IBM サービスに報告してください。

注: DLFM データベース移行プログラムは、UNIX プラットフォームでは db2dlmng、Windows では db2dlmng.exe という名前です。

DLFM341E

Data Links Manager が前に異常終了しています。"DLFM SHUTDOWN" コマンドがおそらく必要です。アクティブ・プロセスの数は <num-processes> です。戻りコード = "<return-code>"。

原因 : Data Links Manager が、前回の実行から残っているシステム・リソースを検出しました。これは、Data Links Manager が異常終了し、残っているシステム・リソースをクリーンアップするまでは、Data Links Manager を再始動できないことを意味します。

- 処置 :** 以下のステップを実行してください。
1. db2diag.log にあるエラー情報を参照してください。
 2. そこに説明されている修正処置にしたがってください。
 3. 残っているリソースをクリーンアップするために、**dlfm shutdown** コマンドを入力します。
 4. 問題が継続する場合は、IBM サービスに報告してください。

DLFM402E

DLFM は現在保守モードで動作しています。再始動リカバリー処理しか許可されていません。この時の DLFM に対しては、その他の処理を実行することはできません。

原因 : データ・リンク・ファイル・マネージャーに重大エラーが発生しました。

- 処置 :** 以下のステップを実行してください。
1. db2diag.log にあるエラー情報を参照してください。
 2. 修正処置を実行します。
 3. データ・リンク・ファイル・マネージャーを再始動します。
 4. 問題を解決できない場合は、その問題を IBM サービスに報告してください。

DLFM501E

<manager-type> 管理サービス "<service-type>" が失敗しました。戻りコード = "<return-code>"

接続、ファイル、グループ、接頭部、トランザクションの *manager-type* 値については、下記の対応する *service-type* を参照してください。

AbortTxn

- 原因 :** 以下の理由が考えられます。
1. データ・リンク・ファイル・マネージャーのログ・マネージャーでエラーが起きました。
 2. システム共用リソースにアクセスするときにエラーが発生しました。
 3. オペレーティング・システムにエラーが発生しました。
 4. データ・リンク・ファイル・マネージャーのログ・ファイルは破壊されています。

- 処置 :** 以下のステップを実行してください。
1. db2diag.log にあるエラー情報を参照してください。
 2. データ・リンク・ファイル・マネージャーを再始動します (必要な場合)。

3. 問題を IBM サービスに報告してください。

BeginTxn

原因 : 以下の理由が考えられます。

1. データ・リンク・ファイル・マネージャーのログ・マネージャーでエラーが起きました。
2. システム共用リソースにアクセスするときにエラーが発生しました。
3. オペレーティング・システムにエラーが発生しました。
4. データ・リンク・ファイル・マネージャーのログ・ファイルは破壊されています。

処置 : 以下のステップを実行してください。

1. db2diag.log にあるエラー情報を参照してください。
2. データ・リンク・ファイル・マネージャーを再始動します (必要な場合)。
3. 問題を IBM サービスに報告してください。

CommitTxn

原因 : 以下の理由が考えられます。

1. データ・リンク・ファイル・マネージャーのログ・マネージャーでエラーが起きました。
2. システム共用リソースにアクセスするときにエラーが発生しました。
3. オペレーティング・システムにエラーが発生しました。
4. データ・リンク・ファイル・マネージャーのログ・ファイルは破壊されています。

処置 : 以下のステップを実行してください。

1. db2diag.log にあるエラー情報を参照してください。
2. データ・リンク・ファイル・マネージャーを再始動します (必要な場合)。
3. 問題を IBM サービスに報告してください。

Connect

原因 : データベースがデータ・リンク・ファイル・マネージャーに接続するのに失敗しました。以下の理由が考えられます。

1. データベースがデータ・リンク・ファイル・マネージャーに登録されていません。

2. システム共用リソースにアクセスするときにエラーが発生しました。
3. オペレーティング・システムにエラーが発生しました。

処置 : 以下のステップを実行してください。

1. db2diag.log にあるエラー情報を参照してください。
2. DB2 Data Links Manager 管理者は、データベースをデータ・リンク・ファイル・マネージャーに登録する必要があります。
3. 問題が継続する場合は、IBM サービスに報告してください。

DefineGroup

原因 : 以下の理由が考えられます。

1. データ・リンク・ファイル・マネージャーのログ・マネージャーでエラーが起きました。
2. システム共用リソースにアクセスするときにエラーが発生しました。
3. オペレーティング・システムにエラーが発生しました。
4. データ・リンク・ファイル・マネージャーのログ・ファイルは破壊されています。

処置 : 以下のステップを実行してください。

1. db2diag.log にあるエラー情報を参照してください。
2. データ・リンク・ファイル・マネージャーを再始動します (必要な場合)。
3. 問題を IBM サービスに報告してください。

DeleteDatabase

原因 : 以下の理由が考えられます。

1. データ・リンク・ファイル・マネージャーのログ・マネージャーでエラーが起きました。
2. システム共用リソースにアクセスするときにエラーが発生しました。
3. オペレーティング・システムにエラーが発生しました。
4. データ・リンク・ファイル・マネージャーのログ・ファイルは破壊されています。

処置 : 以下のステップを実行してください。

1. db2diag.log にあるエラー情報を参照してください。
2. データ・リンク・ファイル・マネージャーを再始動します (必要な場合)。

3. 問題を IBM サービスに報告してください。

DeleteGroup

原因 : 以下の理由が考えられます。

1. データ・リンク・ファイル・マネージャーのログ・マネージャーでエラーが起きました。
2. システム共用リソースにアクセスするときにエラーが発生しました。
3. オペレーティング・システムにエラーが発生しました。
4. データ・リンク・ファイル・マネージャーのログ・ファイルは破壊されています。

処置 : 以下のステップを実行してください。

1. db2diag.log にあるエラー情報を参照してください。
2. データ・リンク・ファイル・マネージャーを再始動します (必要な場合)。
3. 問題を IBM サービスに報告してください。

Disconnect

原因 : 以下の理由が考えられます。

1. データ・リンク・ファイル・マネージャーのログ・マネージャーでエラーが起きました。
2. システム共用リソースにアクセスするときにエラーが発生しました。
3. オペレーティング・システムにエラーが発生しました。
4. データ・リンク・ファイル・マネージャーのログ・ファイルは破壊されています。

処置 : 以下のステップを実行してください。

1. db2diag.log にあるエラー情報を参照してください。
2. データ・リンク・ファイル・マネージャーを再始動します (必要な場合)。
3. 問題を IBM サービスに報告してください。

LinkFiles

原因 : 以下の理由が考えられます。

1. データ・リンク・ファイル・マネージャーのログ・マネージャーでエラーが起きました。
2. システム共用リソースにアクセスするときにエラーが発生しました。

3. オペレーティング・システムにエラーが発生しました。
4. データ・リンク・ファイル・マネージャーのログ・ファイルは破壊されています。

処置 : 以下のステップを実行してください。

1. db2diag.log にあるエラー情報を参照してください。
2. データ・リンク・ファイル・マネージャーを再始動します (必要な場合)。
3. 問題を IBM サービスに報告してください。

PrepareTxn

原因 : 以下の理由が考えられます。

1. データ・リンク・ファイル・マネージャーのログ・マネージャーでエラーが起きました。
2. システム共用リソースにアクセスするときにエラーが発生しました。
3. オペレーティング・システムにエラーが発生しました。
4. データ・リンク・ファイル・マネージャーのログ・ファイルは破壊されています。

処置 : 以下のステップを実行してください。

1. db2diag.log にあるエラー情報を参照してください。
2. データ・リンク・ファイル・マネージャーを再始動します (必要な場合)。
3. 問題を IBM サービスに報告してください。

QueryARTxns

原因 : 以下の理由が考えられます。

1. データ・リンク・ファイル・マネージャーのログ・マネージャーでエラーが起きました。
2. システム共用リソースにアクセスするときにエラーが発生しました。
3. オペレーティング・システムにエラーが発生しました。
4. データ・リンク・ファイル・マネージャーのログ・ファイルは破壊されています。

処置 : 以下のステップを実行してください。

1. db2diag.log にあるエラー情報を参照してください。
2. データ・リンク・ファイル・マネージャーを再始動します (必要な場合)。

3. 問題を IBM サービスに報告してください。

QueryGroups

原因 : 以下の理由が考えられます。

1. データ・リンク・ファイル・マネージャーのログ・マネージャーでエラーが起きました。
2. システム共用リソースにアクセスするときにエラーが発生しました。
3. オペレーティング・システムにエラーが発生しました。
4. データ・リンク・ファイル・マネージャーのログ・ファイルは破壊されています。

処置 : 以下のステップを実行してください。

1. db2diag.log にあるエラー情報を参照してください。
2. データ・リンク・ファイル・マネージャーを再始動します (必要な場合)。
3. 問題を IBM サービスに報告してください。

QueryGroupsExt

原因 : 以下の理由が考えられます。

1. データ・リンク・ファイル・マネージャーのログ・マネージャーでエラーが起きました。
2. システム共用リソースにアクセスするときにエラーが発生しました。
3. オペレーティング・システムにエラーが発生しました。
4. データ・リンク・ファイル・マネージャーのログ・ファイルは破壊されています。

処置 : 以下のステップを実行してください。

1. db2diag.log にあるエラー情報を参照してください。
2. データ・リンク・ファイル・マネージャーを再始動します (必要な場合)。
3. 問題を IBM サービスに報告してください。

QueryGroupsExt2

原因 : 以下の理由が考えられます。

1. データ・リンク・ファイル・マネージャーのログ・マネージャーでエラーが起きました。
2. システム共用リソースにアクセスするときにエラーが発生しました。

3. オペレーティング・システムにエラーが発生しました。
4. データ・リンク・ファイル・マネージャーのログ・ファイルは破壊されています。

処置 : 以下のステップを実行してください。

1. db2diag.log にあるエラー情報を参照してください。
2. データ・リンク・ファイル・マネージャーを再始動します (必要な場合)。
3. 問題を IBM サービスに報告してください。

QueryPreparedTxns

原因 : 以下の理由が考えられます。

1. データ・リンク・ファイル・マネージャーのログ・マネージャーでエラーが起きました。
2. システム共用リソースにアクセスするときにエラーが発生しました。
3. オペレーティング・システムにエラーが発生しました。
4. データ・リンク・ファイル・マネージャーのログ・ファイルは破壊されています。

処置 : 以下のステップを実行してください。

1. db2diag.log にあるエラー情報を参照してください。
2. データ・リンク・ファイル・マネージャーを再始動します (必要な場合)。
3. 問題を IBM サービスに報告してください。

ReleaseDelete

原因 : データ・リンク・ファイル・マネージャーは、現在の状態のファイルを削除することができません。ファイルは、データ・リンク・ファイル・マネージャーによって管理されていたときに、管理ユーザーによって削除または変更されました。

処置 : このファイルの状況をチェックしてから、必要に応じて削除してください。

ReleaseRestore

原因 : データ・リンク・ファイル・マネージャーは、このファイルの元の所有者および許可をリストアできませんでした。ファイルは、データ・リンク・ファイル・マネージャーによって管理されていたときに、管理ユーザーによって削除または変更されました。

処置 : このファイルの状況をチェックして、必要に応じて、所有権や許可の属性を変更するか、ファイルを削除してください。

ResolvePrefixId

原因 : 以下の理由が考えられます。

1. データ・リンク・ファイル・マネージャーのログ・マネージャーでエラーが起きました。
2. システム共用リソースにアクセスするときにエラーが発生しました。
3. オペレーティング・システムにエラーが発生しました。
4. データ・リンク・ファイル・マネージャーのログ・ファイルは破壊されています。

処置 : 以下のステップを実行してください。

1. db2diag.log にあるエラー情報を参照してください。
2. データ・リンク・ファイル・マネージャーを再始動します (必要な場合)。
3. 問題を IBM サービスに報告してください。

ResolvePrefixName

原因 : 所定のファイルの接頭部がデータ・リンク・ファイル・マネージャーに登録されていません。

処置 : DB2 Data Links Manager 管理者は、指定したファイルの接頭部を、Data Links サーバーに登録する必要があります。

TakeOver

原因 : データ・リンク・ファイル・マネージャーはファイルの管理を始めることができません。データ・リンク・ファイル・マネージャーがこのファイルをコントロールする前にファイルが削除または変更されました。

処置 : このファイルの状況をチェックして、必要に応じて、所有権や許可の属性を変更してください。

UnlinkFile

原因 : 以下の理由が考えられます。

1. データ・リンク・ファイル・マネージャーのログ・マネージャーでエラーが起きました。
2. システム共用リソースにアクセスするときにエラーが発生しました。

3. オペレーティング・システムにエラーが発生しました。
4. データ・リンク・ファイル・マネージャーのログ・ファイルは破壊されています。

処置 : 以下のステップを実行してください。

1. db2diag.log にあるエラー情報を参照してください。
2. データ・リンク・ファイル・マネージャーを再始動します (必要な場合)。
3. 問題を IBM サービスに報告してください。

DLFM701E

"<command>" に無効な数の引き数 (<num-args>) が指定されました。理由コード = "<reason-code>"

原因 : 指定した *command* で指定した引き数が多すぎるか、あるいは少なすぎます。 *num-args* は、実際に指定された引き数の数です。

処置 : 指定したコマンドで指定する引き数の数を修正してください。
command の正しい構文の詳細については、**dlfm help** を使用してください。

DLFM703E

指定したサーバー名 "<server-name>" を使用して、ネットワーク・ホスト項目を入力できませんでした。戻りコード = "<return-code>"

原因 : "gethostbyname" システム機能は指定した *server-name* を解決することができず、特定の *return-code* で失敗しました。

処置 : 指定した *server-name* が正しく指定されており、ローカル・ドメイン・ネーム・サーバーまたはシステム *hosts* ファイルで定義されていることを確認してください。

DLFM704E

指定したコントロール・ファイル名はこの DLFM サーバーに関連していません。

原因 : "dlfm_export" ユーティリティでは、最初のパラメーターとしてそれに渡された "control file name" は、現行 DLFM サーバーのホスト名と一致している必要があります。

処置 : 指定した "control file name" は正しく指定されており、DLFM サーバーのホスト名 (または IP アドレス) に一致することを確認してください。

DLFM706W

この DLFM サーバーからアーカイブされるファイルはありません。

原因 : "dlfm_export" ユーティリティは、"control file" の入力の内容を読

み取りましたが、その URL 項目を 1 つも処理できませんでした。この理由は、通常、ファイルに現行 DLFM サーバーを参照する URL が 1 つも存在していないためです。この結果、出力アーカイブ・ファイルに書き込まれるファイルはありません。

処置 : コントロール・ファイルの少なくとも 1 つの URL が、現行 DLFM サーバーのホスト名と一致するホスト名を持つことを確認してください。このような URL がない場合は、この DLFM サーバーのアーカイブに書き込む必要のあるファイルは実際に存在しません。また、このサーバーにこれ以上の処置は必要ありません。

DLFM707I

"<command>" は正常に完了しました。

原因 : 指定した *command* はその処理を正常に完了しました。

処置 : 必要ありません。

DLFM721E

コマンド "<command>" に無効な引き数 ("<arg>") が指定されています。

原因 : 指定した *arg* 値は、指定した *command* の有効な引き数ではありません。

処置 : 指定した *command* の正しい構文を確認し、それにしたがって入力した引き数を修正してください。

DLFM741E

DLFM_BACKUP_DIR_NAME レジストリー変数に無効なバックアップ・ディレクトリー名 "<dir-name>" が指定されています。

原因 : DLFM_BACKUP_DIR_NAME レジストリー変数の値 *dir-name* が無効です。この変数で指定するディレクトリーは、絶対パス (つまり、スラッシュ文字で始まる) でなければなりません。

処置 : *dir-name* 値が、現行システム上に存在する、正しい形式の絶対パスであることを確認してください。(このディレクトリーは、ファイル・バックアップのためにデータ・リンク・ファイル・マネージャーによって使用されます。) **db2set** コマンドを使用して、DLFM_BACKUP_DIR_NAME レジストリー変数の値が正しいものになるように修正してください。

DLFM742E

DLFM_BACKUP_DIR_NAME レジストリー変数に指定されたバックアップ・ディレクトリー名 "<dir_name>" にアクセスすることができません。 registry variable.)

原因 : DLFM_BACKUP_DIR_NAME レジストリー変数の値 *dir-name* が無効です。

この変数で指定するディレクトリーは、既存のディレクトリーの絶対パスでなければなりません。指定した *dir-name* が見付かりませんでした。

処置 : *dir-name* 値が、現行システム上に存在する絶対パスであることを確認してください。(このディレクトリーは、ファイル・バックアップのためにデータ・リンク・ファイル・マネージャーによって使用されます。) **db2set** コマンドを使用して、DLFM_BACKUP_DIR_NAME レジストリー変数の値が正しいものになるように修正してください。

DLFM743E

ローカル・ディスク・バックアップに保管されているファイル "<file-name>" が見付かりませんでした。

原因 : 指定した *file-name* はローカル・ディスク・バックアップにあるはずですが、そこにはありませんでした。

処置 : 以下のステップを実行してください。

1. db2diag.log にあるエラー情報を参照してください。
2. そこに説明されている修正処置にしたがってください。
3. **db2set DLFM_BACKUP_DIR_NAME** コマンドを使用して、DLFM_BACKUP_DIR_NAME レジストリー変数が、ファイル・バックアップのためにこのデータ・リンク・ファイル・マネージャーが使用するディレクトリーを指していることを確認してください。
4. 正しく設定されていない場合は、**db2set** コマンドを使用して、DLFM_BACKUP_DIR_NAME レジストリー変数の値が正しいものになるように修正してください。
5. 指定した *file-name* がバックアップ・ディレクトリーに存在することを確認してください。

DLFM744E

ローカル・ディスク・バックアップのファイル "<file-name>" にアクセスできませんでした。

原因 : 指定した *file-name* はローカル・ディスク・バックアップにありましたが、それにアクセスできませんでした。

処置 : 以下のステップを実行してください。

1. db2diag.log にあるエラー情報を参照してください。
2. そこに説明されている修正処置にしたがってください。
3. **db2set DLFM_BACKUP_DIR_NAME** コマンドを使用して、DLFM_BACKUP_DIR_NAME レジストリー変数が、ファイル・バックアップのためにこのデータ・リンク・ファイル・マネージャーが使用するディレクトリーを指していることを確認してください。

4. 正しく設定されていない場合は、**db2set** コマンドを使用して、`DLFM_BACKUP_DIR_NAME` レジストリー変数の値が正しいものになるように修正してください。
5. 指定した *file-name* がバックアップ・ディレクトリーに存在し、少なくとも、ファイル所有者が `READ` 許可をもっていることを確認してください。

DLFM746E

要求した `<item-type>` リストに対応するレコードはありませんでした。

原因 : データ・リンク・ファイル・マネージャー・データベースに、既に指定済みの基準に一致する *item-type* のタイプ項目がありません。

処置 : 探し出したいバックアップ・ファイル情報を検索できるかどうかを確認するために、別の引き数を指定するか、別の選択項目セットを指定することによって、このコマンドを再実行することをお勧めします。

DLFM747E

`<item-type>` リストを取り出しているときに `SQL` エラーが発生しました。戻りコード = "`<return-code>`"

原因 : 既に指定された基準にしたがってタイプ項目 *item-type* のリストを検索しているときに、予期しないデータベース・エラーが発生しました。*return-code* 値は、データベース処理に関連した `SQL` エラー・コードを示しています。

処置 : 以下のステップを実行してください。

1. `db2diag.log` にあるエラー情報を参照してください。
2. そこに説明されている修正処置にしたがってください。
3. 問題が継続する場合は、`IBM` サービスに報告してください。

DLFM748I

選択したい `<item-type>` に対応する番号を入力してください。

原因 : このメッセージに続いて、有効な *item-type* 選択のリストがあります。コマンドの処理を継続するには、このリストのいずれかの項目を選択する必要があります。

処置 : 選択したい項目に対応する番号を入力してください。

DLFM749E

`<item-type>` リストを処理しているときに `SQL` エラーが発生しました。戻りコード = "`<return-code>`"

原因 : 既に指定された基準にしたがってタイプ項目 *item-type* のリストを検

索しているときに、予期しないデータベース・エラーが発生しました。*return-code* 値は、データベース処理に関連した SQL エラー・コードを示しています。

処置： 以下のステップを実行してください。

1. db2diag.log にあるエラー情報を参照してください。
2. そこに説明されている修正処置にしたがってください。
3. 問題が継続する場合は、IBM サービスに報告してください。

DLFM750I

DLFM データベース "<dbname>" を使用しています。

原因： データ・リンク・ファイル・マネージャー環境は、*dbname* がそのプライベート DB2 データベースの名前であることを示しています。このデータベースは、DLFM バックアップ・ファイルの情報を検索するために使用されるデータベースです。

処置： 必要ありません。

DLFM751E

データベース "<dbname>" に接続できませんでした。戻りコード = "<return-code>"

原因： *dbname* データベースに接続しようとしているときに、予期しないデータベース・エラーが発生しました。*return-code* 値は、DB2 CONNECT ステートメントに関連する SQL エラー・コードを示しています。

処置： 以下のステップを実行してください。

1. db2diag.log にあるエラー情報を参照してください。
2. そこに説明されている修正処置にしたがってください。
3. *dbname* が、DLFM データベースの正しい名前であることを確認してください。(このデータベースのデフォルト名は "DLFM_DB" です。)
4. 問題が継続する場合は、IBM サービスに報告してください。

DLFM801I

DLFM 移行を開始しています。処理が完了するのをお待ちください。

原因： データ・リンク・ファイル・マネージャーの移行プログラムが開始されました。このプログラムは、前のバージョンの DLFM 環境を現在インストールされているバージョンの DLFM 環境に変換します。

処置： 移行を中断することなく完了させてください。データ・リンク・ファイル・マネージャーが保持するファイルの情報量によって異なりますが、この処置が完了するには、数分、またはそれよりかなりの時間がかかる場合もあります。

DLFM802I

移行パッケージをバインドしています。

原因 : データ・リンク・ファイル・マネージャー移行プログラムは、現在インストールされているバージョンの DLFM への移行を進めるために、それ自身を既存の DLFM_DB データベースにバインドしています。

処置 : 移行を中断することなく完了させてください。データ・リンク・ファイル・マネージャーが保持するファイルの情報量によって異なりますが、この処置が完了するには、数分、またはそれよりかなりの時間がかかる場合もあります。

DLFM803I

DLFM データベースをバックアップしています。

原因 : データ・リンク・ファイル・マネージャー移行プログラムは、現在インストールされているバージョンの DLFM への移行を進めるために、既存の DLFM_DB データベースをバックアップしています。

処置 : 移行を中断することなく完了させてください。データ・リンク・ファイル・マネージャーが保持するファイルの情報量によって異なりますが、この処置が完了するには、数分、またはそれよりかなりの時間がかかる場合もあります。

DLFM804I

以下のデータベース処理が実行されています。

原因 : データ・リンク・ファイル・マネージャー移行プログラムは、現在インストールされているバージョンの DLFM への移行を進めるために、既存の DLFM_DB データベースに対して数多くのデータベース処理を実行しています。このメッセージの次に、データベースの処理とその結果のリストが続いています。

処置 : 移行を中断することなく完了させてください。データ・リンク・ファイル・マネージャーが保持するファイルの情報量によって異なりますが、この処置が完了するには、数分、またはそれよりかなりの時間がかかる場合もあります。

DLFM805I

"<database-action>": SQL コード = "<sql-code>"、戻りコード = "<return-code>"。

原因 : データ・リンク・ファイル・マネージャー移行プログラムは、現在インストールされているバージョンの DLFM に移行するために、既存の DLFM_DB データベースに対して、指示された *database-action* を実行しました。*sql-code* および *return-code* は、指定した *database-action* の結果を示します。

処置 : 指定したデータベース処理と移行プログラム全体を、中断することな

く完了させてください。データ・リンク・ファイル・マネージャーが保持するファイルの情報量によって異なりますが、この処置が完了するには、数分、またはそれよりかなりの時間がかかる場合もあります。

DLFM806I

"<database-action>": SQL コード = "<sql-code>", 戻りコード = "<return-code>", 理由コード = "<reason-code>"。

原因 : データ・リンク・ファイル・マネージャー移行プログラムは、現在インストールされているバージョンの DLFM に移行するために、既存の DLFM_DB データベースに対して、指示された *database-action* を実行しました。*sql-code*、*return-code*、*reason-code* は、指定した *database-action* の結果を示しています。

処置 : 指定したデータベース処理と移行プログラム全体を、中断することなく完了させてください。データ・リンク・ファイル・マネージャーが保持するファイルの情報量によって異なりますが、この処置が完了するには、数分、またはそれよりかなりの時間がかかる場合もあります。

DLFM807I

データ "<data-name>" をファイル "<file-name>" にエクスポートしています。

原因 : データ・リンク・ファイル・マネージャー移行プログラムは、指示された *data-name* を作業ファイル *file-name* にエクスポートしています。このエクスポートは、現在インストールされているバージョンの DLFM に移行するために、既存の DLFM_DB データベースを使用して実行されています。

処置 : エクスポート処理と移行プログラム全体を、中断することなく完了させてください。データ・リンク・ファイル・マネージャーが保持するファイルの情報量によって異なりますが、この処置が完了するには、数分、またはそれよりかなりの時間がかかる場合もあります。

DLFM808I

エクスポートされたデータ "<data-name>" の行数 = <num-rows>。

原因 : データ・リンク・ファイル・マネージャー移行プログラムは、指示された *data-name* の *num-rows* を作業ファイルにエクスポートしました。このエクスポートは、現在インストールされているバージョンの DLFM に移行するために、既存の DLFM_DB データベースを使用して実行されています。

処置 : エクスポート処理と移行プログラム全体を、中断することなく完了させてください。データ・リンク・ファイル・マネージャーが保持する

ファイルの情報量によって異なりますが、この処置が完了するには、数分、またはそれよりかなりの時間がかかる場合もあります。

DLFM809I

ファイル "<file-name>" からデータ "<data-name>" をロードしています。

原因： データ・リンク・ファイル・マネージャー移行プログラムは、作業ファイル *file-name* から指示された *data-name* をロードしています。現在インストールされているバージョンの DLFM に移行するために、既存の DLFM_DB データベースの中へロードされています。

処置： ロード処理と移行プログラム全体を、中断することなく完了させてください。データ・リンク・ファイル・マネージャーが保持するファイルの情報量によって異なりますが、この処置が完了するには、数分、またはそれよりかなりの時間がかかる場合もあります。

DLFM810I

ロードされたデータ "<data-name>" の行数 = <num-rows1>、コミットされた行数 = <num-rows2>。

原因： データ・リンク・ファイル・マネージャー移行プログラムは、作業ファイルから、指示された *data-name* の *num-rows1* をロードし、これらの行の *num-rows2* をコミットしました。現在インストールされているバージョンの DLFM に移行するために、既存の DLFM_DB データベースの中へロードされています。

処置： ロード処理と移行プログラム全体を、中断することなく完了させてください。データ・リンク・ファイル・マネージャーが保持するファイルの情報量によって異なりますが、この処置が完了するには、数分、またはそれよりかなりの時間がかかる場合もあります。

DLFM811E

現行 DLFM データベースをバックアップすることができませんでした。戻りコード = "<return-code>" DLFM 移行は停止しています。

原因： データ・リンク・ファイル・マネージャー移行プログラムは、既存 DLFM_DB データベースのバックアップに失敗しました。*return-code* は、DB2 バックアップ・ユーティリティー・プログラムからの戻りコードです。

処置： 以下のステップを実行してください。

1. db2diag.log にあるエラー情報を参照してください。
2. そこに説明されている修正処置にしたがってください。
3. 問題が継続する場合は、IBM サービスに報告してください。

DLFM812I

主要な DLFM パッケージをバインドしています。

原因： データ・リンク・ファイル・マネージャー移行プログラムは、現在インストールされているバージョンの DLFM を、移行された DLFM_DB データベースにバインドしています。

処置： バインド処理と移行プログラム全体を、中断することなく完了させてください。データ・リンク・ファイル・マネージャーが保持するファイルの情報量によって異なりますが、この処置が完了するには、数分、またはそれよりかなりの時間がかかる場合もあります。

DLFM813I

DLFM 移行のすべてのステップが正常に完了しました。

原因： データ・リンク・ファイル・マネージャー移行プログラムは、現在インストールされているバージョンの DLFM に一致させるために、既存の DLFM_DB データベースの変換を完了しました。すべてのステップが正常に完了しました。

処置： 必要ありません。

DLFM814I

DLFM 移行が終了しました。

原因： データ・リンク・ファイル・マネージャーの移行プログラムが終了しました。

処置： DLFM 移行プログラムの成功や失敗を示す先行するメッセージを記録しておいてください。成功した場合は、追加処置は必要ありません。失敗した場合は、移行プログラムを再実行する前に、エラー・メッセージに関連した指示にしたがってください。

DLFM815E

DLFM データベース "DLFM_DB" は不明な状態です。元の DLFM データベースをリストアして、"<dlfm-migration-program>" 移行プログラムを再実行してください。

原因： DLFM データベース DLFM_DB の内容は、このバージョンのデータベースのフォーマットに対応していません。指定した Data Links Manager 移行プログラム <dlfm-migration-program > は、この矛盾のため、実行できませんでした。この問題は、前のこのプログラムの実行が正常に完了しなかったことが原因になっている可能性があります。

処置：

1. 移行される、元の DLFM_DB データベースをリストアします。移行中にバックアップされる DLFM_DB は、DLFM_BACKUP_DIR_NAME 変数に指定されたディレクトリーに保管されます。

2. Data Links Manager 移行を再実行してください。プログラムを中断することなく完了させてください。
3. 問題が継続する場合は、IBM サービスに報告してください。

注: Data Links Manager 移行プログラムは、UNIX プラットフォームでは db2d1mmg、Windows では db2d1mmg.exe です。

DLFM816W

既存の DLFM データベースは、既に現行の Data Links Manager リリース用の正しいフォーマットになっており、移行の必要はありません。

原因 : DLFM データベース DLFM_DB の内容は、既に Data Links Manager の現在インストールされているバージョン用のフォーマットになっています。内容がこのフォーマットになっているのには、以下のいずれかの理由があります。

- データベースが、Data Links Manager の現行リリースを使用して作成された (たとえば **dlfm create** または **dlfm setup** コマンドを使用)。
- データベースは以前のリリースの Data Links Manager を使用して作成され、既に移行済み。

処置 :

- DLFM_DB データベースが現在インストールされている Data Links Manager を使用して作成された場合は、追加のアクションは必要ありません。DLFM データベース移行プログラムは、新規に作成された DLFM_DB データベースで実行する必要はありません。
- DLFM_DB データベースが以前のリリースの Data Links Manager で作成され、DLFM データベース移行プログラムが正常に実行された場合は、追加のアクションは必要ありません。DLFM データベース移行プログラムの再実行はしないでください。
- DLFM_DB データベースが以前のリリースの Data Links Manager で作成され、DLFM データベース移行プログラムが正常に実行されなかった場合は、以下のステップを実行してください。
 1. 移行される、元の DLFM_DB データベースをリストアします。移行中にバックアップされる DLFM_DB は、DLFM_BACKUP_DIR_NAME 構成変数で指定されたディレクトリーに保管されます。
 2. DLFM データベース移行プログラムを再実行してください。**重要:**プログラムを中断することなく完了させてください。
 3. DLFM データベース移行プログラムが正常に完了した場合は、追加アクションはありません。

4. 移行プログラムが正常に完了しなかった場合は、報告されたエラーを記録して、示された説明にしたがってください。
5. 問題が継続する場合は、IBM サービスに報告してください。

注: Data Links Manager 移行プログラムは、UNIX プラットフォームでは db2d1mmg、Windows では db2d1mmg.exe です。

DLFM817E

表 "<table-name>" は空ではありません。

原因 : DLFM データベース移行プログラムは、"<table-name>" という名前の表を作成しようとしたのですが、この名前の空でない表が、DLFM データベース DLFM_DB に既に存在します。

注: Data Links Manager 移行プログラムは、UNIX プラットフォームでは db2d1mmg、Windows では db2d1mmg.exe です。

処置 :

- 以前に表を作成していて、その中にデータを挿入済みの場合、必要がない場合は表をドロップ、または 'dfm' が表名の最初の 3 文字にならないように、表を名前変更する必要があります。
- 指定した表が DLFM_DB データベースから実際に除去された後で、以下のステップを実行してください。
 1. DLFM データベース移行プログラムを再実行してください。重要: プログラムを中断することなく完了させてください。
 2. 移行プログラムが正常に完了した場合は、追加アクションはありません。
 3. 移行プログラムが正常に完了しなかった場合は、報告されたエラーを記録して、示された説明にしたがってください。
 4. 問題が継続する場合は、IBM サービスに報告してください。

DLFM818E

表 "<table-name>" には列 "<column-name>" が既に存在します。

原因 : DLFM データベース移行プログラムは、<column-name> という名前の列を <table-name> という名前に表に追加しようとしたのですが、この名前の列は、指定した表に既に存在します。

注: DLFM データベース移行プログラムは、UNIX プラットフォームでは db2d1mmg、Windows では db2d1mmg.exe です。

処置 : 以下のステップを実行してください。

1. 移行される、元の DLFM_DB データベースをリストアします。移行中にバックアップされる DLFM_DB は、DLFM_BACKUP_DIR_NAME 構成変数で指定されたディレクトリに保管されます。
2. 以前にこの列をこの表に追加している場合は、表から列をドロップしてください。Data Links Manager 以外では、DLFM_DB データベースの表は変更できません。
3. 以前にこの列をこの表に追加していなかった場合は、おそらく直前の不完全な DLFM データベース移行プログラムの実行によって追加されたと思われます。この列は、上の最初のステップで元の DLFM_DB をリストアしたときに、DLFM_DB データベースで指定した表から実際に除去されます。
4. DLFM データベース移行プログラムを再実行してください。**重要:** プログラムを中断することなく完了させてください。
5. 移行プログラムが正常に完了した場合は、追加アクションはありません。
6. 移行プログラムが正常に完了しなかった場合は、報告されたエラーを記録して、示された説明にしたがってください。
7. 問題が継続する場合は、IBM サービスに報告してください。

DLFM819I

バッファー・プール "<bufferpool-name>" は既に存在します。

原因 : DLFM データベース移行プログラムは、<bufferpool-name> という名前のバッファー・プールを作成しようとしたのですが、正しい必要な特性 (pagesize など) を持つこの名前のバッファー・プールは、DLFM データベース DLFM_DB に既に存在します。

注: DLFM データベース移行プログラムは、UNIX プラットフォームでは db2d1mng、Windows では db2d1mng.exe です。

処置 : アクションは必要ありません。既存のバッファー・プールには必要な特性があるため、それは DLFM_DB の内容に使用されます。

DLFM820E

バッファー・プール "<bufferpool-name>" は既に定義されていますが、ページ・サイズが間違っています。

原因 : DLFM データベース移行プログラムは、<bufferpool-name> という名前のバッファー・プールを作成しようとしたのですが、正しくない特性 (the pagesize) を持つこの名前のバッファー・プールは、DLFM データベース DLFM_DB に既に存在します。

注: DLFM データベース移行プログラムは、UNIX プラットフォームでは db2d1mmg、Windows では db2d1mmg.exe です。

処置 : 以下のステップを実行してください。

1. 以前にこのバッファ・プールを作成していた場合は、ドロップしてください。Data Links Manager プログラム以外では、DLFM_DB データベースの内容は変更できません。
2. DLFM データベース移行プログラムを再実行してください。重要: プログラムを中断することなく完了させてください。
3. 移行プログラムが正常に完了した場合は、追加アクションはありません。
4. 移行プログラムが正常に完了しなかった場合は、報告されたエラーを記録して、示された説明にしたがってください。
5. 問題が継続する場合は、IBM サービスに報告してください。

DLFM821I

表スペース "<tablespace-name>" は既に存在します。

原因 : DLFM データベース移行プログラムは、<tablespace-name> という名前の表スペースを作成しようとしたのですが、正しい必要な特性 (pagesize など) を持つこの名前の表スペースは、DLFM データベース DLFM_DB に既に存在します。

処置 : アクションは必要ありません。既存の表スペースには必要な特性があるため、それは DLFM_DB データベースの内容に使用されます。

DLFM822E

表スペース "<tablespace-name>" は既に定義されていますが、ページ・サイズが間違っています。

原因 : DLFM データベース移行プログラムは、<tablespace-name> という名前の表スペースを作成しようとしたのですが、正しくないページ・サイズを持つこの名前の表スペースは、DLFM データベース DLFM_DB に既に存在します。

注: DLFM データベース移行プログラムは、UNIX プラットフォームでは db2d1mmg、Windows では db2d1mmg.exe です。

処置 : 以下のステップを実行してください。

1. 以前にこの表スペースを作成していた場合は、ドロップしてください。Data Links Manager プログラム以外では、DLFM_DB データベースの内容は変更できません。
2. DLFM データベース移行プログラムを再実行してください。重要: プログラムを中断することなく完了させてください。

3. 移行プログラムが正常に完了した場合は、追加アクションはありません。
4. 移行プログラムが正常に完了しなかった場合は、報告されたエラーを記録して、示された説明にしたがってください。
5. 問題が継続する場合は、IBM サービスに報告してください。

DLFM823I

DLFM データベース移行の実行時間は、<minutes>:<seconds> (分:秒) でした。

原因 : DLFM データベース DLFM_DB の内容が、正常に移行されました。移行の経過時間は、分および秒で表示されます。

処置 : 必要ありません。 DLFM データベース移行プログラムは、中断されることなく最終処理を完了する必要があります。

注: DLFM データベース移行プログラムは、 UNIX プラットフォームでは db2d1mng、 Windows では db2d1mng.exe という名前です。

DLFM841E

プログラム "<program-name>" は UNIX 環境でのみ実行可能です。

原因 : 指定されたプログラムが、Windows などの、AIX または Solaris オペレーティング環境以外の環境で実行されました。

処置 : AIX または Solaris オペレーティング環境で必要とされるプログラムを実行します。

DLFM842I

"<migration-type>" 移行を開始しています。処理が完了するのをお待ちください。

移行タイプは、プログラム dlfm_migrate_fsid に "file system"、またはプログラム dlfm_migrate_prefix に "prefix" のいずれかになります。

原因 : データ・リンク・ファイル・マネージャー <migration-type> 移行プログラムが処理を開始しました。このプログラムは、論理的にファイル・システムまたは接頭部を、既存のロケーションから新規ロケーションに、プログラムの引き数に指定されたとおりに移行します。

処置 : 移行を中断することなく完了させてください。データ・リンク・ファイル・マネージャーが保守するファイルの情報量によって異なりますが、移行処理が完了するには、数分またはそれ以上かかる場合があります。

データ・リンク・ファイル・システムの、他のハード・ディスクへの移行へのヘルプは、以下を参照してください。

- AIX でデータ・リンク・ファイル・システムを他のハード・ディスクに移行する

- Solaris オペレーティング環境でデータ・リンク・ファイル・システムを他のハード・ディスクに移行する
- Windows でデータ・リンク・ファイル・システムを他のハード・ディスクに移行する
- DLFF 使用可能ファイル・システムを他のハード・ディスクに移行する

DLFM843E

"<file-system>" がマウントされていません。

原因 : 指定されたファイル・システムがマウントされていません。

処置 : 指定されたファイル・システムをマウントしてください。

ファイル・システムのマウントについてのヘルプは、「ファイル・システムの使用可能化と DLFFM への登録 (AIX、Solaris オペレーティング環境)」を参照してください。

DLFM844I

ファイル・システム ID "<fsid>" の "<prefix-name>" の処理中です ...

原因 : 対応するファイル・システム ID を持つ、指定した DLFFM 接頭部のファイル・システム移行が処理中です。

処置 : アクションは必要ありません。プログラムを中断することなく継続させてください。

データ・リンク・ファイル・システムの、他のハード・ディスクへの移行へのヘルプは、以下を参照してください。

- AIX でデータ・リンク・ファイル・システムを他のハード・ディスクに移行する
- Solaris オペレーティング環境でデータ・リンク・ファイル・システムを他のハード・ディスクに移行する
- Windows でデータ・リンク・ファイル・システムを他のハード・ディスクに移行する
- DLFF 使用可能ファイル・システムを他のハード・ディスクに移行する

DLFM845I

<file-count> ファイル・レコードが DLFFM データベースで更新されました...

原因 : DLFFM データベース DLFFM_DB に記録する <file-count> リンク・ファイル情報が、ファイル・システムの移行中に処理されました。この状況メッセージは定期的に表示され、データベース処理の進行を示します。

処置 : アクションは必要ありません。ファイル・システム移行プログラムが、中断されることなく、継続するようにしてください。

DLFM846I

DLFM データベースで更新が必要なファイル・レコードの合計数は <file-count> でした。

原因 : DLFM データベース DLFM_DB に記録する <file-count> リンク・ファイル情報の合計が、ファイル・システムの移行のファイル情報更新フェーズ中に処理されました。表示されたファイルの数が 0 の場合、更新が必要な移行済みファイル・システムのファイルに、システムの情報がありません (たとえば、元のファイル・システムと新規のファイル・システム間で、ファイル・システム ID が変更されなかった場合)。

処置 : アクションは必要ありません。ファイル・システム移行プログラムが、中断されることなく、残りの移行フェーズの処理を完了するようにしてください。

DLFM847I

ファイル・システム移行のすべてのステップが正常に完了しました。

原因 : DLFM ファイル・システム移行プログラムは、指定したファイル・システムの変更に一致するように、DLFM データベース DLFM_DB への必要な更新を正常に完了しました。

処置 : アクションは必要ありません。

DLFM848I

接頭部移行のすべてのステップが正常に完了しました。接頭部パスが、"<prefix1>" から "<prefix2>" に変更されました。

原因 : DLFM 接頭部移行プログラムは、<prefix1> から <prefix2> への、指定した接頭部の変更に一致するように、DLFM データベース DLFM_DB への必要な更新を正常に完了しました。

処置 : アクションは必要ありません。

DLFM853E

DLFM データベース・トランザクション・ログがフルです。ログのスペース割り振りを増加して、このプログラムを再実行してください。

原因 : DLFM データベース DLFM_DB のトランザクション・ログがフルです。DLFM データベースでのトランザクション処理のロギングに必要な記憶域が、これ以上ありません。

処置 : DLFM サーバー上の DLFM_DB データベースの、トランザクション・

ログ・データベース構成パラメーターを増加してください。ログ・スペース割り振りを増加した後で、このプログラムを再実行できます。

ログのスペース割り振りの増加のヘルプは、「DLFM_DB 用の十分な DB2 ログ・スペースの確認」を参照してください。

DLFM900I

DLFM サーバーが停止しました。

原因 : データ・リンク・ファイル・マネージャーが正常または異常に停止されました。

処置 : DLFM が異常終了した場合は、**dlfm shutdown** コマンドを出してから、**dlfm start** コマンドを出して、DLFM を再始動してください。

DLFM901E

システム・エラーが発生しました。戻りコード = "<return-code>"。現在のコマンドは処理できませんでした。追加情報については、db2diag.log ファイルを参照してください。

原因 : 予期しないシステム環境エラーまたは処理エラーが発生しました。以下の原因が考えられます。

1. DLFM_DB データベースの自動バックアップを実行できない。
2. ファイル・システムからファイル情報を取得できない。
3. 子プロセスを作成できない。
4. システム・コマンドを実行できない。
5. データベース・インスタンス情報を取得できない。
6. 予期しない SQL エラーが発生した。
7. ディレクトリーの変更のような、ファイル・システム操作を実行できない。
8. システム時刻を取得できない。
9. 内部処理エラーが発生した。
10. db2diag.log ファイルを初期化できない。
11. 固有の内部接頭部 ID を生成できない。

処置 : 以下のステップを実行してください。

1. db2diag.log にあるエラー情報を参照してください。
2. そこに説明されている修正処置にしたがってください。
3. 問題が継続する場合は、IBM サービスに報告してください。

DLFM908E

ファイル "<file-name>" に処理 "<file-action>" を実行しているときに、繰り返し起きる失敗が発生しました。DLFM は停止されています。

原因 : Data Links Manager は、ファイル・システムからのファイルの引き継ぎ、またはファイル・システムへのファイルのリリースに失敗しました。アーカイブにバックアップ中に、引き継がれているファイルが 1 つ以上失敗した可能性があります。これらは、DATALINK 列に挿入されたファイルか、DATALINK 列から削除されたファイルです。コミット処理が完了する前に、以下のいずれかの状況が発生した可能性があります。

- ファイルを含んでいるファイル・システムがオフラインにされた。
- 管理ユーザーによってファイルが削除された。
- アーカイブ・サーバーまたはアーカイブ・ストレージが一時的に使用できなくなっている。

処置 : 以下のステップを実行してください。

1. ファイル・システムが使用可能であることを確認し、そのファイルがあるかどうかをチェックします。
2. ファイルは引き継がれているが、バックアップができなかった場合は、アーカイブ・サーバーおよびアーカイブ・ストレージが使用可能であることを確認します。
3. ファイルが削除されておらず、現在使用可能である場合は、Data Links Manager を再始動してください。これにより、ファイルが解放されて Data Links Manager は使用可能になります。

ファイルが削除された場合、あるいは問題が継続する場合は、その問題を IBM サービスに報告してください。

DLFM909E

ファイル "<existing-file>" をファイル "<replacement-file>" と置き換えているときに、繰り返し起きる失敗が発生しました。DLFM は停止されています。

原因 : Data Links Manager は、既存ファイルを置き換えファイルと置き換えるのに失敗しました。コミット処理が完了する前に、以下のいずれかの状況が発生した可能性があります。

- ファイルを含んでいるファイル・システムがオフラインにされた。
- 置き換えファイルが管理ユーザーによって削除された。

処置 : 以下のステップを実行してください。

1. ファイル・システムが使用可能であることを確認し、置き換えファイルがあるかどうかをチェックします。
2. 置き換えファイルが削除されておらず、現在使用可能である場合は、Data Links Manager を再始動してください。再始動すると、ファイルがリリースされ、Data Links Manager が使用可能になります。

ファイルが削除された場合、あるいは問題が継続する場合は、その問題を IBM サービスに報告してください。

dlfm_dump コマンド

要件： このコマンドは、Data Links Manager の管理者ユーザー ID を使用して、Data Links サーバーから実行する必要があります。

```
▶▶—dlfm_dump—-all—┬───┴───▶▶  
                        └─o—output_filename┘
```

-all すべての有効な DLFM 定義およびデータが出力ファイルに置かれることを指定する、必要パラメーター。

-o output_filename

データが保管される出力ファイルの名前を指定するオプション・パラメーター。デフォルト出力ファイル名は `dlfm_dump.log` です。

`dlfm_dump` ユーティリティでは、DLFM_DB データベースに保管された有効データの "スナップショット" を入手できます。このデータがあると、さまざまなデータ・リンク・ファイル・マネージャー (DLFM) のシステム構成詳細、および Data Links サーバー上の他の DLFM 関連データを検査できます。出力は、`dlfm_dump` ユーティリティが呼び出されるディレクトリー内のファイルに保管されます。出力ファイルに保管されるデータには、以下のものが含まれます。

- 現行 DLFM_DB のバージョン。
- アクセス・トークン生成で現在使用中のキー。
- セキュリティー管理情報。
- 登録済みデータベースおよび接頭部。
- データ・リンク・ファイル・システム (DLFS) ツリー構造。
- この Data Links サーバーを参照する DATALINK 列。
- この Data Links サーバー上のリンク・ファイルを含む DB2 バックアップ。
- この Data Links サーバー上のすべてのリンクおよびリンク解除のファイルのリスト。

`dlfm_dump` ユーティリティの出力には、トランザクション処理中に保管されるデータなどの、DLFM_DB が保守している可能性のある一時データは含まれていません。

重要：

- DLFM が多くのファイルを管理している場合、現在リンクされているものと同様に、以前にリンクされていたものの両方で、出力ファイルは非常に大きくなります。出力ファイルが書き込まれるファイル・システムに、十分なスペースがあることを確認し

てください。DLFM_DB が占める、または DLFM_DB のバックアップが占めるスペースの量を、 dlfm_dump 出力のファイルに必要なスペースを見積もるのに使用できます。

- dlfm_dump ユーティリティの出力には重要なセキュリティ情報が含まれているため、出力ファイルがセキュア・ディレクトリーにあることを確認してください。

IBM サービスに問題を報告する場合、Data Links Manager 環境の詳細をサービス担当者に提供するために、dlfm_dump ユーティリティを実行するように依頼される場合があります。

注: dlfm_dump ユーティリティの出力は、最小限にフォーマットされており、一般的な報告が目的ではなく、問題の診断を支援することを第一にしています。Data Links サーバー・データの読み取り可能な出力を作成するには、以下のコマンドのいずれか、またはすべてを使用します。

- **dlfm retrieve**
- **dlfm list registered databases**
- **dlfm list registered directories**
- **dlfm list registered prefixes**
- **dlfm list registered replication access control**
- **dlfm list registered users**
- **dlfm list upd_in_progress files for db**
- **dlfm list upd_in_progress files for prefix**

関連資料:

- *DB2 Data Links Manager 管理ガイドおよびリファレンス* の『dlfm list registered databases コマンド』
- *DB2 Data Links Manager 管理ガイドおよびリファレンス* の『dlfm list registered prefixes コマンド』
- *DB2 Data Links Manager 管理ガイドおよびリファレンス* の『dlfm list registered replication access control コマンド』
- *DB2 Data Links Manager 管理ガイドおよびリファレンス* の『dlfm list registered users コマンド』
- *DB2 Data Links Manager 管理ガイドおよびリファレンス* の『dlfm list upd_in_progress files for db コマンド』
- *DB2 Data Links Manager 管理ガイドおよびリファレンス* の『dlfm list upd_in_progress files for prefix コマンド』
- *DB2 Data Links Manager 管理ガイドおよびリファレンス* の『dlfm list registered directories コマンド』
- *DB2 Data Links Manager 管理ガイドおよびリファレンス* の『dlfm retrieve コマンド』

Data Links Manager バージョン 8 での DCE-DFS サポートの撤回

Data Links Manager の機能は、バージョン 8 の DCE-DFS 環境では提供されません。Data Links Manager は引き続きバージョン 7 の DFS™ サポートは提供します。V8 製品、インストール・パッケージ、メッセージ、およびドキュメンテーションに、Data Links Manager の DFS 環境への参照がある場合がありますが、これらは無視されるべきで、適用できません。

Data Links バージョン 7 を使用していて、DCE-DFS サポートに従属している場合は、Data Links Manager バージョン 8 の別のストレージ環境を使用するか、または DFS での従属関係を解決するまで、Data Links Manager バージョン 8 に移動するべきではありません。

データウェアハウス・センター管理ガイド

ウェアハウス・サーバーとロガーの開始および停止 (AIX)

AIX では、root ユーザーは、ウェアハウス・サーバー (iwh2serv) デーモンとロガー (iwh2log) デーモンを、シェル・スクリプト db2vwsvr を使用して手動で開始または停止できます。これは、なんらかの理由で、iwh2serv デーモンと iwh2log デーモンが実行していない場合に実行できます。デーモンを開始すると、コマンドによって、ID が db2vwsvr である inittab ファイルに、1 つの項目が追加されます。結果として、ウェアハウス・サーバー・デーモンが停止した場合に、このコマンドが自動的に再始動されません。

手順:

ウェアハウス・サーバー・デーモンとロガー・デーモンを手動で開始するには、AIX コマンド・プロンプトで以下のコマンドを入力してください。

```
db2vwsvr start
```

ウェアハウス・サーバー・デーモンとロガー・デーモンを停止するには、AIX コマンド・プロンプトで以下のコマンドを入力してください。

```
db2vwsvr stop
```

デーモンを停止すると、iwh2serv 処理と iwhlog 処理が再始動され、ID が db2vwsvr である inittab 内の項目が更新されます。デーモンを停止すると、コマンドは inittab 項目も除去します。結果として、デーモンは、「開始 (start)」オプションを使用してコマンドを再実行するまでは、再始動されません。

注: db2vwsvr は最初に IWH.environment スクリプトを実行して、環境を初期化します。スクリプトは、同じディレクトリー内にあることが想定されます。db2vwsvr と IWH.environment スクリプト・ファイルは、DB2 インストール・ディレクトリ

一の bin サブディレクトリーにインストールされます。IWH.environment を変更した場合には、コマンド db2vwsvr start を使用して、デーモンを再始動する必要があります。

AIX オペレーティング・システムでデータウェアハウス・センターまたはインフォメーション・カタログ・センターを使用している場合、SQL1224N エラーを受け取る可能性があります。このエラーを受け取った場合、可能な原因のリストのメッセージ・ヘルプ・テキストを参照してください。問題が共有メモリー・セグメントの数に関係している場合は、管理 API 関連資料で、並行アクセスを使用するスレッド化アプリケーションに関する情報を参照してください。この状態の予備手段として、環境変数 EXTSHM の設定、またはアクセスされているデータベースへのループバックの定義などがあります。

関連資料:

- 1022 ページの『並行アクセスを伴うスレッド化アプリケーション』

ウェアハウス・サーバーとロガー・デーモンの動作確認 (AIX)

AIX ウェアハウス・サーバー・デーモンとウェアハウス・ロガー・デーモンが実行中であるかどうかを確認できます。

手順:

ウェアハウス・サーバーとロガー・デーモンが AIX 上で実行中であることを確認するには、以下のようにします。

1. 以下のコマンドを、DB2 インスタンス所有者として AIX コマンド・プロンプトで入力します。

```
db2 list applications
```

ウェアハウス・サーバーとウェアハウス・ロガー・デーモンが実行中である場合には、このコマンドによって、処理 iwh2serv (サーバー) と iwh2log (ロガー) が表示されます。

Auth Id	Application Name	Appl. Handle	Application Id	DB Name	# of Agents
DB2IV81	iwh2serv	4	*LOCAL.db2iv81.0A0871221727	DWCTRLDB	1
DB2IV81	iwh2log	3	*LOCAL.db2iv81.07B631221725	DWCTRLDB	1

2. DB2 インスタンス所有者として、以下のコマンドを AIX コマンド・プロンプトで入力します。

```
$ netstat -a|grep vwkernel
```

ウェアハウス・サーバーとウェアハウス・ロガーが実行中である場合には、このコマンドによって、listen 状態の vwkernel プロセスが表示されます。

```
tcp4      0      0 *.vwkernel          *.*          LISTEN
```

3. DB2 インスタンス所有者として、以下のコマンドを AIX コマンド・プロンプトで入力します。

```
$ netstat -a|grep vwlogger
```

ウェアハウス・サーバーとウェアハウス・ロガーが実行中である場合には、このコマンドによって、listen 状態の vwlogger プロセスが表示されます。

```
tcp4      0      0 *.vwlogger          *.*          LISTEN
```

デーモンが開始されていない場合、ロギング・ディレクトリー内の DB2VWSVR.LOG、IWH2LOG および IWH2SERV.LOG ファイルにエラーがないかどうかをチェックすることもできます。ロギング・ディレクトリーは、環境変数 VWS_LOGGING によって指定されます。デフォルト値は、/var/IWH です。

サポートされる DB2 以外のデータ・ソース

以下の表は、データウェアハウス・センターによってサポートされる DB2 以外のデータ・ソースをリストしています。

ソース・データベース	Windows NT、 Windows	AIX、 Solaris 操作環境、 Linux エージェント
	2000、 Windows XP エージェント	
Informix (クライアント使用 および不使用)	9.x	9.x
Oracle	8.1.5, 8.1.7, 9i	8.1.5, 8.1.7, 9i
Microsoft® SQL Server	7.0, 2000	7.0, 2000
Microsoft Excel	97	サポートされません
Microsoft Access	97	サポートされません
Sybase Adaptive Server (ク ライアント不使用)	12.0	11.9.2 - 12.0

データウェアハウス・センターには、DB2 以外のデータにアクセスするための、複数の DataDirect Technologies, Inc. ODBC ドライバーが組み込まれています。

DATALINK のバックス正規形式 (BNF) 仕様

DATALINK 値はカプセル化された値で、データベース以外のロケーションに保管されているファイルへのデータベースからの論理参照を含んでいます。

このカプセル化された値のデータ位置属性は、URL の形式でのファイルへの論理参照です。この属性の値は、以下の BNF によって指定される、URL の構文に準拠していません。これは、RFC 1738 : Uniform Resource Locators (URL), T. Berners-Lee, L. Masinter,

M. McCahill, December 1994 に基づいています。(BNF は "バックス正規形式" (特定の言語の構文を記述するための正式な表記) です。)

BNF 指定では、以下の規則が使用されます。

- 代替を指定するのに "|" を使用する
- オプションまたは繰り返されるエレメントの周りに大括弧 [] を使用する
- リテラルは "" で囲む
- エレメントの前に [n] * を付ければ、その直後のエレメントを n 回以上反復させることができる。n が指定されていない場合、デフォルトは 0 です。

DATALINK での BNF 指定を以下に示します。

URL

```
url          =  httpurl | fileurl | uncurl | emptyurl
```

HTTP

```
httpurl     =  "http://" hostport [ "/" hpath ]
hpath       =  hsegment *[ "/" hsegment ]
hsegment    =  *[ uchar | ";" | ":" | "@" | "&" | "=" ]
```

RFC1738 の元の BNF での検索エレメントは除去されている点に注意してください。これは、そのエレメントがファイル参照の本質的な部分ではなく、DATALINK のコンテキストでは意味をなさないためです。

FILE

```
fileurl     =  "file://" host "/" fpath
fpath       =  fsegment *[ "/" fsegment ]
fsegment    =  *[ uchar | "?" | ":" | "@" | "&" | "=" ]
```

RFC1738 とは異なり、host がオプションではなく、"localhost" スtringが特別な意味を持たない点に注意してください。これにより、"localhost" の解釈で、クライアント/サーバー構成とパーティション・データベース構成で混乱が生じるのを避けることができます。

UNC

```
uncurl      =  "unc:¥¥" hostname "¥¥" sharename "¥¥" uncpath
sharename   =  *uchar
uncpath     =  fsegment *[ "¥¥" fsegment ]
```

Windows では、一般的に使用されている UNC 命名規則をサポートしています。これは RFC1738 での標準方式ではありません。

EMPTYURL

```
emptyurl    =  ""
hostport    =  host [ ":" port ]
host        =  hostname | hostnumber
hostname    =  *[ domainlabel "." ] toplabel
domainlabel =  alphadigit | alphadigit *[ alphadigit | "-" ] alphadigit
```

DATALINK のバックス正規形式 (BNF) 仕様

```
|          toplabel    =  alpha | alpha *[ alphadigit | "-" ] alphadigit
|          alphadigit  =  alpha | digit
|          hostnumber  =  digits "." digits "." digits "." digits
|          port        =  digits
```

DATALINK 値では、空の (長さゼロ) URL もサポートされています。これらは、調整例外が報告されて NULL 不可の DATALINK 列が呼び出される場合に、DATALINK 列を更新するのに便利です。長さゼロの URL を使用すれば、その列が更新され、リンクが解除されます。

各種の定義

```
lowalpha  =  "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" |
             "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" |
             "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" |
             "y" | "z"
hialpha   =  "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" |
             "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P" |
             "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" |
             "Y" | "Z"
alpha     =  lowalpha | hialpha
digit     =  "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" |
             "8" | "9"
safe      =  "$" | "-" | "_" | "." | "+"
extra     =  "!" | "*" | "~" | "(" | ")" | ","
hex       =  digit | "A" | "B" | "C" | "D" | "E" | "F" |
             "a" | "b" | "c" | "d" | "e" | "f"
escape    =  "%" hex hex
unreserved = alpha | digit | safe | extra
uchar     =  unreserved | escape
digits    =  1*digit
```

前後の空白文字は、構文解析中に DB2 によって切り詰められます。また、スキーム名 ('HTTP'、'FILE'、'UNC') および host では大文字小文字が区別されず、いつでも大文字でデータベースに保管されます。

データウェアハウス・センター・メタデータの出版用にアプリケーション・ヒープ・サイズを増大

大量のデータウェアハウス・センター・メタデータを出版する場合、インフォメーション・カタログ・データベースのヒープ・サイズを超える場合があります。この問題を訂正するには、アプリケーション・ヒープ・サイズを増やす必要があります。

手順:

アプリケーション・ヒープ・サイズを増やすには、以下のコマンドを DB2 コマンド・プロンプトで入力します。

```
update db cfg for dbname using applheapsz 1024
```

ここで、*dbname* はインフォメーション・カタログ・データベースの名前です。

ウェアハウス・ステップ

ターゲット・データへのソース・データの移動およびトランスフォーム方法を定義するステップを作成する必要があります。主なステップ・タイプとウェアハウス・コネクタのリストを以下に示します。

SQL ステップ

データウェアハウス・センターには、2 種類の SQL ステップがあります。

「SQL 選択および挿入 (SQL Select and Insert)」ステップは、SQL SELECT ステートメントを使用してウェアハウス・ソースからデータを抽出し、そのデータをウェアハウス・ターゲット表に挿入するための INSERT ステートメントを生成します。「SQL 選択および更新 (SQL Select and Update)」SQL ステップは、SQL UPDATE ステートメントを使用して、ウェアハウス・ターゲット表内のデータを更新します。

ウェアハウス・プログラム・ステップ

ウェアハウス・プログラム・ステップは、事前定義されたプログラムとユーティリティを実行します。ウェアハウス・プログラム・ステップにはいくつかのタイプがあります。

- ファイル・プログラム
- OLAP プログラム
- ウェアハウスの使用すべきでないプログラム
- 特定の DB2 Universal Database プラットフォーム用に設計されたプログラム

特定のオペレーティング・システムのウェアハウス・プログラムは、オペレーティング・システムのエージェントと共にパッケージされています。ウェアハウス・プログラムは、エージェント・コードのインストール時にインストールしてください。

DSNUTILS は、WLM および RRS 環境で実行される DB2[®] for z/OS[™] ストアード・プロシージャです。ユーザー定義のストアード・プロシージャ・インターフェースを使用することによって、DSNUTILS を使用して、インストールされている任意の DB2 ユーティリティを実行できます。

各ウェアハウス・プログラムの定義は、初期化時に追加されます。定義のリストを表示するには、データウェアハウス・センターのメイン・ウィンドウで「プログラムおよびトランスフォーマー (Programs and Transformers)」フォルダーを展開します。

トランスフォーマー・ステップ

トランスフォーマー・ステップは、ストアード・プロシージャとユーザー定義関数であり、データをトランスフォームするために使用できる統計またはウェアハウス・トランスフォーマーを指定します。トランスフォーマーを使用し

て、データのクリーニング、逆転、およびピボット、1 次キーおよび期間表の生成、および各種統計の計算を行うことができます。

トランスフォーマー・ステップでは、統計またはウェアハウス・トランスフォーマーの 1 つを指定します。このプロセスを実行すると、トランスフォーマー・ステップは 1 つまたは複数のウェアハウス・ターゲットにデータを書き込みます。

トランスフォーマーには、以下の 2 つのタイプがあります。

- 統計トランスフォーマー
- ウェアハウス・トランスフォーマー

トランスフォーマー・ステップに加えて、ユーザー定義関数である別のトランスフォーマーがあります。このトランスフォーマーは SQL ステップで使用できます。

特定のオペレーティング・システムのトランスフォーマー・プログラムは、オペレーティング・システムのエージェントと共にパッケージされています。ウェアハウス・プログラムは、エージェント・コードのインストール時にインストールしてください。

制限: データウェアハウス・センターのトランスフォーマーは、DB2 Relational Connect ターゲット・データベースではサポートされません。

各トランスフォーマーの定義は、初期化時に追加されます。定義のリストを表示するには、データウェアハウス・センターのメイン・ウィンドウで「**プログラムおよびトランスフォーマー (Programs and Transformers)**」フォルダーを展開します。

トランスフォーマーを使用する前に、以下のことを実行してください。

- ターゲット・データベースにトランスフォーマーをインストールします。
- 「ウェアハウス・ターゲット (Warehouse Target)」ノートブックの、「データベース (Database)」ページから、トランスフォーマーを作成して登録します。
- 「ウェアハウス・ターゲット (Warehouse Target)」ノートブックの「データベース (Database)」ページから、トランスフォーマー用にターゲット・データベースを使用可能にします。
- 各トランスフォーマーでプログラム定義を変更し、エージェントとターゲット・リソースを指定します。
- ソース表とターゲット表は同じデータベース内に常駐するようにします。

レプリケーション・ステップ

レプリケーション・ステップは、任意の DB2 リレーショナル・データベース内の指定された変更を、あるロケーション (ソース) から別のロケーション (ターゲット) にコピーし、両方のロケーションでデータを同期化します。ソース

とターゲットは、同じマシン、または分散ネットワーク内の別々のマシンにある論理サーバー (DB2 データベース、DB2 for z/OS サブシステム、またはデータ共有グループなど) に存在することができます。

レプリケーション・ステップを使用すると、運用表が更新されるたびに表全体をロードしなくても、ウェアハウス表を運用表と同期化することができます。レプリケーションでは、インクリメンタル更新を使用して、データを現行の状態に保つことができます。

SAP ステップ

SAP Data Extract ステップを使用して、SAP ビジネス・オブジェクトを SAP R/3 システムから抽出し、それらをデータウェアハウス・センターにインポートします。

Web トラフィック・ステップ

Web Traffic Polling を使用して、WebSphere® Site Analyzer データ・インポートの状態を照会し、現行の Web トラフィック・データが、そのデータ・インポートから WebSphere Site Analyzer Webmart データベース内の表にコピーされているかどうかを判別します。

ユーザー定義プログラム・ステップ

ユーザー定義プログラム・ステップは、データウェアハウス・センターにある論理エンティティであり、データウェアハウス・センターが開始するビジネス固有のトランスフォーメーションを表します。いかなるビジネスにも固有のデータ・トランスフォーメーション要件があるので、ビジネスでは、独自のプログラム・ステップを作成するか、または ETI や Vality などが提供するツールを使用するかを選択できます。

たとえば、以下の関数を実行するユーザー定義プログラムを作成できます。

1. 表からデータをエクスポートする。
2. そのデータを操作する。
3. データを一時的な出力リソースまたはウェアハウス・ターゲットに書き込む。

これらのタイプのステップで提供されていない機能が必要な場合、独自のウェアハウス・プログラムまたはトランスフォーマーを作成してから、そのプログラムまたはトランスフォーマーを使用するステップを定義できます。

ステップの各グループには、ステップのサブタイプがあります。すべての場合に、データを移動したりトランスフォームしたりするための特定のステップ・サブタイプを選択します。たとえば、ANOVA トランスフォーマーは「統計トランスフォーマー (Statistical transformer)」グループのサブタイプです。

ウェアハウス・ステップの従属関係検査

従属関係検査は、ウェアハウス内のデータの保全性を、壊れたデータの発生時にステップの実行を避けることによって、保証します。たとえば、表にリンク

された 2 つのステップ、ステップ 1 およびステップ 2 があるとします。表はステップ 2 のソース表およびステップ 1 のターゲット表です。表は一時表ではありません。従属関係検査がステップ 2 に対して使用可能になっていて、ステップ 1 が失敗、または実行されなかった場合、ステップ 2 の実行時にエラー・メッセージを受け取ります。

従属関係検査は、ウェアハウスで他のステップによって移植されたソースを持つステップに対して使用可能です。従属関係検査は、ステップを作成したときに、デフォルトで使用可能になりますが、使用不可にする必要がある場合があります。たとえば、前のステップの成功、失敗にかかわらずステップを実行したい場合、従属関係検査を使用不可にする必要がある場合があります。

従属関係検査が使用可能な状態で、プロセス・モデル・ウィンドウからステップの順序をテストする場合、ステップの再試行プロパティを 0 に設定します。再試行プロパティを 0 に設定すると、従属関係検査に関連したエラーが、直ちにプロセス・モデル・ウィンドウに返されます。再試行プロパティが 0 より大きい値に設定されており、従属関係検査が使用可能な場合、すべての再試行の完了後に、進行状況ウィンドウに従属関係検査に関連したエラーが返されます。

関連タスク:

- *DB2 Warehouse Manager* インストール・ガイド の『ウェアハウス・トランスフォーマーの使用可能化』
- *DB2 Warehouse Manager* インストール・ガイド の『DB2 (z/OS 版) ユーティリティの実行』

ステップ・タスク・フロー

データウェアハウス・センターでは、ステップを開始する方法は 2 つあります。ステップが、他のステップの実行後に開始するように指定できます。または、指定された日時にステップを開始するようスケジュールすることができます。これらの方法を組み合わせて、プロセス内のステップを実行することができます。指定された日時に最初のステップを実行するようスケジュールし、このようなスケジュールおよびカスケードは、ステップが実行中にアクティブになります。最初のステップをスケジュールした後、別のステップが最初のステップの実行後に開始するように指定し、さらに 3 番目のステップが 2 番目のステップの実行後に開始するように指定し、これを繰り返すことができます。

2 つのステップ間で、タスク・フローの定義に複数の条件を使用する場合、その条件は、プロセス・モデル・ウィンドウを閉じて、再開した時に完了 (On completion) になります。たとえば、ステップ A および ステップ B の以下のタスク・フローを、スケジュール・ノートブックのステップ・タスク・フロー・ページに定義します。

```
Step A -> On success -> Step B  
Step A -> On failure -> Step B
```


そしてステップを保管し、プロセス・モデル・ウィンドウを閉じて、再開します。スケジュール・ノートブックのステップ・タスク・フロー・ページには、プロセス A および プロセス B の以下のタスク・フローが表示されます。

Step A -> On completion -> Step B

外部トリガー・プログラムを使用して、データウェアハウス・センターの外側でのステップの実行

外部トリガー・プログラムを使用することにより、ステップをデータウェアハウス・センター・コントロール・インターフェースに依存しないで開始できます。外部トリガー・プログラムは、データウェアハウス・センターを呼び出すウェアハウス・プログラムです。

外部トリガー・プログラムは、2 つのコンポーネント XTServer と XTClient から構成されます。XTServer はウェアハウス・サーバーと一緒にインストールされます。XTClient はすべてのエージェント・タイプのウェアハウス・エージェントとともにインストールされます。

前提条件:

- 外部トリガー・プログラムによってステップをトリガー処理する前に、ステップに関する「プロパティ (Properties)」ノートブックの「処理オプション (Processing Options)」ページで「要求時実行 (Run on demand)」を選択する必要があります。
- 使用している外部トリガー・プログラムが、DB2 Enterprise Server Edition バージョン 8 より古い DB2 Universal Database のバージョンである場合、CLASSPATH 変数を変更して、正しい Java クラス・ファイルをポイントする必要がある必要があります。

Windows 2000、Windows NT または Windows XP 上のクライアント・トリガーまたはサーバー・トリガーを実行するには、CLASSPATH が sqllib/tools/db2XTrigger.jar と sqllib/java/common.jar をポイントする必要があります。

クライアント・トリガーまたはサーバー・トリガーを iSeries で実行するには、CLASSPATH が /qibm/userdata/os400/sqllib/Function/JET をポイントする必要があります。

クライアント・トリガーまたはサーバー・トリガーを z/OS で実行するには、CLASSPATH がデータウェアハウス・センターのインストールされているディレクトリをポイントする必要があります。たとえば /user/lpp/dwc81 です。

- 外部トリガー・プログラムを使用してステップを開始するには、ウェアハウス・サーバー・ワークステーションおよびエージェント・サイトに、JDK 1.3 をインストールしておく必要があります。データウェアハウス・センターとコントロール・センターによってインストールされた JDK を使用することもできます。

手順:

外部トリガー・プログラムを開始するには、以下のようにします。

1. 外部トリガー・サーバーを開始する。
2. 外部トリガー・クライアントを開始する。

関連資料:

- データウェアハウス・センター 管理ガイド の『外部トリガー・サーバーを開始するための構文』
- 1086 ページの『外部トリガー・クライアントを開始するための構文』

外部トリガー・クライアントを開始するための構文

外部トリガー・クライアントを始動するための構文は、以下のとおりです。

```
▶ java -DDWC_MIN=min_port_number -DDWC_MAX=max_port_number  
▶ -db2_vw_xt.XTClient -TriggerServerHostName -TriggerServerPort -DWUserID  
▶ -DWUserPassword -StepName -Command -WaitForStepCompletion -RowLimit
```

-DDWC_MIN=min_port_number

オプション。ファイアウォールが XTClient の XTServer との通信を許可する最小ポート番号です。ファイアウォールの XTClient のサポートを使用可能にするために、このパラメーターの値を指定する必要があります。

-DDWC_MAX=max_port_number

オプション。ファイアウォールが XTClient の XTServer との通信を許可する最大ポート番号です。ファイアウォールの XTClient のサポートを使用可能にするために、このパラメーターの値を指定する必要があります。

TriggerServerHostName

外部トリガー・サーバー (XTServer) が実行中であるワークステーションの TCP/IP ホスト名。

完全修飾ホスト名を指定します。

TriggerServerPort

XTServer サーバーに割り当てられている TCP/IP ポート。外部トリガー・クライアントが使用するポートは、外部トリガー・サーバーが使用するものと同じである必要があります。

この値は、外部トリガー・サーバー (XTServer) が実行中であるポートを示す必要があります。

DWUserID

データウェアハウス・センター操作特権を持つユーザー ID。

DWUserPassword

ユーザー ID のパスワード。

StepName

開始するステップの名前。

名前では大文字小文字の区別があります。名前の中に空白が入っている場合は、名前を二重引用符 ("") で囲みます ("Corporate Profit" など)。

Command

以下のいずれかの値です。

- 1 データの取り込み (またはステップの実行)
外部トリガー・プログラムを実行するユーザー ID は、ステップを含むプロセスと同じウェアハウス・グループにある必要があります。
- 2 テスト・モードへのステップのプロモート
外部トリガー・プログラムを実行するユーザー ID は、ステップを含むプロセスと同じウェアハウス・グループにある必要があります。
- 3 実動モードへのステップのプロモート
外部トリガー・プログラムを実行するユーザー ID は、ステップを含むプロセスと同じウェアハウス・グループにある必要があります。
- 4 テスト・モードへのステップのデモート
外部トリガー・プログラムを実行するユーザー ID は、ステップを含むプロセスと同じウェアハウス・グループにある必要があります。
- 5 開発モードへのステップのデモート
外部トリガー・プログラムを実行するユーザー ID は、ステップを含むプロセスと同じウェアハウス・グループにある必要があります。

WaitForStepCompletion

オプション。このパラメーターは、外部トリガー・プログラムがステップ処理の結果を戻すかどうかを示します。以下のいずれかの値を選択します。

- 1 ステップの完了まで待機し、ステップが正常に終了した場合 0 を、失敗した場合エラーを戻します。
- 0 または空白
ステップの完了を待機しません。

RowLimit

オプション。このパラメーターは、外部トリガー・プログラムがソース表のすべての行を抽出するか、または行のサブセットを抽出するかを示します。このパラメーターを使用すると、大規模データベースからデータを抽出するステップを、す早くテストすることができます。以下のいずれかの値を選択します。

- 0 または空白
すべての行を取り出す
- n n 行を取り出す

このパラメーターは、ステップがテスト・モードである場合のみ有効です。

CLASSPATH 変数は、コマンドの実行前に正しく設定する必要があります。

例

たとえば、ユーザー ID db2admin およびパスワード db2admin を使用して、「会社収益 (Corporate Profit)」ステップを開始するとします。外部トリガー・プログラムは、dwserver ホスト上にあります。以下のコマンドを実行します。

```
java XTClient dwserver 11004 db2admin db2admin "Corporate Profit" 1
```

関連タスク:

- 1085 ページの『外部トリガー・プログラムを使用して、データウェアハウス・センターの外側でのステップの実行』

関連資料:

- データウェアハウス・センター 管理ガイド の『外部トリガー・サーバーを開始するための構文』

プロセス・タスク・フロー

データウェアハウス・センターでは、ステップと同様に処理もスケジュールできます。処理が、他の処理の実行後に開始するように指定できます。複数のステップを意味のある処理に注意深くまとめて、プロセス・タスクの流れを適切にスケジュールし、指定できるようにする必要があります。「スケジューラー (Scheduler)」ノートブックの「プロセス・タスク・フロー (Process Task Flow)」ページを使用すれば、別の処理の完了に基づいて、特定の処理を開始できます。処理の完了は、以下の条件に基づいて判別されません。

成功 (On success)

処理の成功時にカスケードがある場合、カスケードを実行するには、処理内のすべての終端ステップが、正常戻りコードまたは警告戻りコードで完了している必要があります。処理の成功または失敗を判別する場合、処理の内側のステップによってトリガー処理された、処理の外側のステップは考慮されません。

失敗 (On failure)

先行する処理内の終端ステップが失敗した場合に、処理が実行されます。処理の成功または失敗を判別する場合、処理の内側のステップによってトリガー処理された、処理の外側のステップは考慮されません。

完了 (On completion)

先行する処理のステップの実行が完了すると、先行する処理のステップの成功または失敗に関係なく、以下の処理が実行されます。

2 つのプロセス間で、タスク・フローの定義に複数の条件を使用する場合、その条件は、プロセス・モデル・ウィンドウを閉じて、再開した時に完了 (On completion) にな

ります。たとえば、スケジュール・ノートブックのプロセス・タスク・フロー・ページに定義されているプロセス A および プロセス B の次のタスク・フローを定義します。

```
Process A -> On success -> Process B
Process A -> On failure -> Process B
```

そしてプロセスを保管し、プロセス・モデル・ウィンドウを閉じて、再開します。プロセス・タスク・フロー・ページには、プロセス A および プロセス B の次のタスク・フローが表示されます。

```
Process A -> On completion -> Process B
```

カスケード処理では、実動モードのステップのみが実行されます。

スケジューリング処理に関する情報

処理タスクの流れを定義できるだけでなく、特定の日時に、または日時にに関する特定のスケジュールで実行するように、処理をスケジュールできます。「スケジューラー (Scheduler)」ノートブックの「スケジュール (Schedule)」ページを使用して、処理をスケジュールしてください。

カスケード処理の使用によって柔軟性が向上しますが、処理を組み立てる場合には、以下の事項について注意してください。

- 処理内の各ステップは、処理のスケジュールとは異なるスケジュールを持つことができます。各ステップは、処理の一部としてすでに実行中に、実行をスケジュールされている場合は、実行されません。ステップが、処理の一部として実行される前または後に実行するようにスケジュールされている場合、このステップは、処理が完了していない場合でも、実行されます。処理スケジュールが使用可能であるか、使用不可であるかに関係なく、処理内のステップを変更できます。
- 複数のステップを 1 つの処理にまとめる場合には、互いに参照するステップは使用しないように注意してください。自己参照ステップは、処理が無限ループに入る原因になります。
- カスケード処理でショートカットが実行されますが、ショートカットは、処理の完了状況に影響を与えません。

Clean Data トランスフォーマー

Clean Data トランスフォーマーは、ソース・データに対する基本的な修正、置換、およびマッピング操作を実行するために使用できます。Clean Data トランスフォーマーは、ユーザーのステップがアクセスするソース表の指定されたデータ列を操作します。さらに、トランスフォーマーは、ステップで作成したターゲット表に新しい行を挿入します。選択した処理オプションにしたがって、クリーニングできなかったデータがターゲット・エラー表に書き込まれます。Clean Data トランスフォーマーを使用すれば、処理の一部として、ロードまたはインポートの後にデータ値をクリーニングおよび標準化することもできます。

Clean Data トランスフォーマーの規則

Clean Data トランスフォーマーを使用してリレーショナル・ソース・データに適用できる規則を次に示します。

検索 / 置換 (Find and replace)

規則表の「検索 (Find)」列で、選択したソース列値を見つけた後、規則表に指定されている対応する置換値を使用して、ターゲット表の値を置換します。このクリーニング・タイプには規則表が必要です。規則表では、検索 / 置換プロセスで Clean Data トランスフォーマーによって使用される値を指定します。以下の「検索および置換 (Find and Replace)」タイプが使用可能です。

- 正確な一致を検索 (Find exact matches)
- 全ストリングを検索、全ストリングを置換 (Find whole string, replace whole string)¹
- サブストリングを検索、全ストリングを置換 (Find substring, replace whole string)¹
- サブストリングを検索、サブストリングを置換 (Find substring, replace substring)¹

¹ 「文字 (CHARACTER)」データ・タイプにのみ有効です。

「Match オプション (Matching Options)」ウィンドウを使用して、一致を処理する方法を指定します。

一致が検出されると、対応する置換値がターゲット表に書き込まれます。規則表とソース表の両方に差異化列を指定した場合、一致が正常に実施されるには、差異化列の値が正確に一致する必要があります。規則表に順序列を指定した場合、検索および置換操作の一致順序は、順序列の値の昇順に従います。

Multiple match オプションの「エラー表へ書き込み (Write to error table)」を選択するか、またはエラー処理を使用可能化して、エラー処理オプションをオンにしている場合に一致が検出されないと、入力行全体が、実行に関する RUN_ID と共にエラー表に書き込まれます。

このクリーニング・タイプに NULL を許可している場合には、規則表の「検索 (Find)」列に NULL 値を入れる必要があります。

数値クリップ (Numeric clip)

数値のみ。このクリーニング・タイプは、指定した範囲内でない数値入力値をクリップします。範囲内にある入力値は、そのまま出力に書き込まれます。範囲外にある入力値は、下限置換値または上限置換値によって置換されます。このクリーニング・タイプには規則表が必要です。「Match オプション (Matching Options)」ウィンドウを使用して、一致を処理する方法を指定します。

範囲への離散化 (Discretize into ranges)

規則表内の範囲に基づいて入力値の離散化を実行します。このクリーニング・タイプには規則表が必要です。「Match オプション (Matching Options)」ウィンドウを使用して、一致を処理する方法を指定します。

このクリーニング・タイプに NULL を許可している場合には、規則表の「限界 (Bound)」列に NULL 値を入れる必要があります。

NULL 処理による持ち越し (Carry over with null handling)

出力表にコピーされる入力表の列を指定します。入力表から複数の列を選択し、出力表に移すこともできます。このクリーニング・タイプには、規則表は不要です。持ち越しによって、NULL 値を指定値で置換できます。NULL をリジェクトし、リジェクトした行をエラー表に書き込むこともできます。

大文字小文字変換 (Convert case)

ソース列の文字を大文字から小文字へ、または小文字から大文字へ変換してから、ターゲット列へ挿入します。デフォルトでは、ソース列の文字が大文字に変換されます。このクリーニング・タイプには、規則表は不要です。

無効値のエンコード (Encode invalid values)

使用中の規則表の有効値列に含まれていないすべての値を、指定値によって置換します。Clean Data トランスフォーマーの「プロパティ (Properties)」ノートブックに置換値を指定します。ソース列と同じデータ・タイプの置換値を指定する必要があります。たとえば、ソース列が数値タイプの場合、数値置換値を指定する必要があります。有効値は、ターゲット表への書き込み時に変更されません。このクリーニング・タイプには規則表が必要です。

追加の機能

Clean Data トランスフォーマーは、基本的な ETML 機能を提供し、SQL を作成するだけでなく、クリーニング・データ規則と共に使用される以下の機能も提供します。

データ・タイプのサポート: 持ち越し列の規則は、持ち越しが必要なすべての SQL データ・タイプをサポートします。以前のバージョンと異なり、その他の規則は、さまざまな置換値データ・タイプを指定できる柔軟性を持つようになっています。たとえば、1 を 'A' で置換するように、整数値を、同じデータ・タイプではない文字値で置換できます。

以下の表は、各クリーニング・タイプでサポートされるデータ・タイプを示しています。

クリーニング・タイプ	文字 (Character)	数値 (Numeric)	日時 (Date/Time)	図形 (Graphic)	LOB
持ち越し (Carry over)	✔	✔	✔	✔ ¹	✔ ¹

検索 / 置換 (Find and replace)	✓	✓	✓		
離散化 (Discretize)	✓	✓	✓		
クリップ (Clip)		✓			
大文字小文字変換 (Convert case)	✓				
無効値のエンコード (Encode invalid values)	✓	✓	✓		

¹ 「持ち越し (Carry over)」が「グラフィック (Graphic)」および LOB データ・タイプをサポートするのは、このタイプの列を選択からの挿入 (Insert from Select) SQL ステートメントが許可している場合のみです。

NULL のサポート: クレンジングで NULL 値を許可するかどうかを明示的に指定できます。Clean Data トランスフォーマーの「プロパティ (Properties)」ノートブックの「パラメーター (Parameters)」ページの「NULL を許可 (Allow Nulls)」チェック・ボックスを選択して、NULL 値を他のソース値と同様に処理します。「NULL 置換値 (Null Replace Value)」フィールドで置換値を指定すれば、NULL 値を置換できます。ソース列のデータ・タイプとの互換性がある置換値を指定する必要があります。規則表を持つクリーニング・タイプの置換値には、規則置換列のデータ・タイプとの互換性が必要です。置換値は、規則表の NULL 値処理に関する規則をオーバーライドします。

トランスフォームで NULL を許可しない場合、規則が NULL 値を処理するかどうかに関係なく、NULL レコードはエラー表に書き込まれます。

一致:

最初および最後の一致 (First and last matching)

この規則は、指定にしたがって、最初または最後の一致基準を実現します。最初の一致 (first match) オプションは、関係のある規則表で最初の一致を探します。これは、規則表の検索列での重複、または離散化クリーニング・タイプでの範囲の重複を回避するためです。順序列を指定すれば、規則を順序付けできます。順序列を指定しない場合、デフォルトの順序が想定されます。たとえば、検索および置換でのデフォルトの順序は、正確な一致が指定されている場合には、最小の置換値であり、それ以外の場合は、サブストリングの一致に関する最小の検索および置換の値の組です。最大の置換値が使用される「最後の一致 (last match)」の場合にも、同じことが言えます。

複数一致のエラー (Error on multiple matches)

「検索および置換 (find and replace)」と「範囲への離散化 (discretize into ranges)」クリーニング・タイプの場合、入力値が複数の規則に一致する場合、

エラー表にその行を書き込むことができます。これは、どの規則にも一致しないレコードもエラー表に書き込まれることも意味します。

すべての一致を作成 (Create all matches)

デフォルト。すべての一致する値をターゲット表に作成できます。これは、入力行とその対応する出力行の間に 1:n のマッピングが存在することを意味します。たとえば、これは、規則表に重複する検索値または重複する範囲がある場合に発生します。

差異化列: ユーザーは、入力表と規則表の差異化列をオプションで指定できます。ソース表の入力と規則表の規則の全体的一致を満足するには、差異化列が正確に一致する必要があります。

自動サマリー表: オプション: 最初および最後の一致を効率的にサポートするために、トランスフォーマーによって自動的に作成されるサマリー表を使用できます。表に、固有の名前を提供することもできます。この機能は、デフォルトではオフになっています。

照会パフォーマンスの拡張に、サマリー表が DB2® によって使用されたことをチェックするには、Clean Data トランスフォーマーが IWH.LOGTABLE に生成する SQL 照会を参照してください。Visual Explain などのツールを使用して、アクセス・プランを表示することができます。検索および置換クリーン・タイプが置換サブstring・オプションで使用される場合、SQL 照会を編集して、REPLACE SQL 関数が完全にログに書き込まれるようにする必要があります。

トランスフォーマーが作成しようとしているサマリー表名を持つ表がすでに存在している場合、サマリー表は作成されませんが、エラーが IWH.LOGTABLE に書き込まれますが、サマリー表はトランスフォーマーの正常実行に必要なため、トランスフォーマーの実行は停止されません。

エラー処理: 規則に一致しないレコードは、それぞれエラー表に書き込まれます。レコードは変更されずに、RUN_ID と共にエラー表に書き込まれます。

制約事項

Clean Data トランスフォーマーには、以下の制約事項が適用されます。

- ソース表とターゲット表は同じデータベースに存在する必要があります。
- ソース表は、単一のウェアハウス表である必要があります。
- ターゲット表は、デフォルトのターゲット表、または同じターゲット・データベース内の既存の表です。
- ステップに変更を加えることができるのは、ステップが開発モードにあるときだけです。
- NULL 値を許可するソース列を、NULL 値を許可しないターゲット列にマップすることはできません。

- 「NULL 置換値 (Null Replace Value)」フィールドは、「無効値のエンコード (Encode invalid values)」クリーニング・タイプには使用できません。 NULL を、指定した有効値で置換するには、「無効値のエンコード (Encode invalid)」の規則表に NULL 値があってはなりません。
- 「数値クリップ (Numeric Clip)」クリーニング・タイプの最小置換値を使用 (Use minimum replacement value) オプションは、z/OS の場合は使用できません。
- 一部の一致オプションは、z/OS™ または iSeries™ では、各プラットフォーム上のデータベース・サーバーの制約事項が理由で機能しません。
- サマリー表とターゲット・エラー表は、不要になった場合に、ターゲット・ウェアハウス・データベースから手動でドロップする必要があります。不要になる例は、ステップを開発モードにデモートする場合です。

ウェアハウス・メタデータの発行

メタデータの発行 とは、メタデータをデータウェアハウス・センターからインフォメーション・カタログ・センターに転送する処理です。

ウェアハウス・サーバーおよびウェアハウス・エージェントがリモート・システムで実行されている場合、ウェアハウス・コントロール・データベースを、ウェアハウス・デフォルト・エージェント・システムおよびウェアハウス管理クライアント・システムのものと同じユーザー ID、パスワード、および名前でカタログ作成する必要があります。コントロール・データベースが、両方のシステムと同じ名前で作成されていない場合、ウェアハウス・エージェントはウェアハウス・コントロール・データベースと通信できません。

前提条件:

インフォメーション・カタログ・センターへの出版にデータウェアハウス・センターを使用している場合、IWH.environment ファイルの Classpath および Path 変数を更新する必要があります。例:

```
#CLASSPATH=./home/db2inst/sqllib/java/db2java.zip
:/home/db2inst/sqllib/java/sqlj.zip
:/home/db2inst/sqllib/java/runtime.zip
:/home/db2inst/sqllib/java/Common.jar
:/home/db2inst/sqllib/tools/db2_vw.jar
:/home/db2inst/sqllib/tools/db2vwcom.jar
:/home/db2inst/sqllib/tools/db2vwddd.jar
:/home/db2inst/sqllib/tools/db2cmn.jar
:/home/db2inst/sqllib/java/db2jcc.jar:

#PATH=./usr/bin:/usr/bins/etc
:/usr/sbin:/usr/ucb:/usr/bin/X11
:/sbin:/home/db2inst/sqllib/bin
:/home/db2inst/sqllib/bin:/home/db2inst/sqllib/bin
:/home/db2inst/sqllib/bin:
```

手順:

ウェアハウス・メタデータを発行するには、以下のようにします。

1. 環境を準備する。
2. 発行するメタデータを選択する。
3. 発行したメタデータを更新する。
4. 発行したメタデータの定期的な更新をスケジューリングする。

関連概念:

- インフォメーション・カタログ・センター 管理ガイド の『データウェアハウス・センター・メタデータはインフォメーション・カタログにどのように表示されるか』
- インフォメーション・カタログ・センター 管理ガイド の『データウェアハウス・センターでの発行オブジェクトのメンテナンス』
- インフォメーション・カタログ・センター 管理ガイド の『データウェアハウス・センターのメタデータに対する定期的な更新』

関連タスク:

- インフォメーション・カタログ・センター 管理ガイド の『データウェアハウス・センター・メタデータの発行の準備』

関連資料:

- インフォメーション・カタログ・センター 管理ガイド の『インフォメーション・カタログ・センターとデータウェアハウス・センターの間でのメタデータのマッピング』

データウェアハウス・センター管理インターフェースの開始 (AIX)

AIX では、データウェアハウス・センター管理インターフェースは、コマンド行から開始します。

前提条件:

データウェアハウス・センター管理インターフェースの開始前に、ウェアハウス・サーバーとロガーが実行中であることを確認します。

手順:

データウェアハウス・センター管理インターフェースを AIX 上で開始するには、AIX コマンド行で以下のコマンドを入力します。

```
db2dwc
```

AIX オペレーティング・システムでデータウェアハウス・センターまたはインフォメーション・カタログ・センターを使用している場合、SQL1224N エラーを受け取る可能性があります。このエラーを受け取った場合、可能な原因のリストのメッセージ・ヘル

ブ・テキストを参照してください。問題が共有メモリー・セグメントの数に関係している場合は、管理 API 関連資料で、並行アクセスを使用するスレッド化アプリケーションに関する情報を参照してください。この状態の予備手段として、環境変数 EXTSHM の設定、またはアクセスされているデータベースへのループバックの定義などがあります。

関連タスク:

- 『データウェアハウス・センターをオープンする』

関連資料:

- 1022 ページの『並行アクセスを伴うスレッド化アプリケーション』

インフォメーション・カタログ・センター管理ガイド

コマンド行からのタグ言語ファイルのインポート

タグ言語ファイルをインフォメーション・カタログにインポートする場合は、コマンド・プロンプトから `db2icmimport` コマンドを使用します。

前提条件:

タグ言語ファイルをインポートする前に以下の作業が必要です。

- インフォメーション・カタログが作成されていること。
- タグ言語ファイルが作成されていること。

制限:

この作業を行うには、DB2 Universal Database のデータベース管理者でなければなりません。

手順:

コマンド行からタグ言語ファイルをインポートする場合は、コマンド・プロンプトに対して以下のコマンドを入力します。

```
db2icmimport userid userid password password database database catalog catalog  
tagfile tagfile iconpath:iconpath logfile:logfile  
restart:checkpoint trace:YES|NO
```

コマンドの最初の 5 つの値は定位置であり、最後の 4 つはオプションで定位置ではありません。完全修飾パス名は、ファイルへのすべての参照に対して必要です。

database

インフォメーション・カタログが入っているデータベースの名前。

catalog

インフォメーション・カタログが入っているスキーマの名前。

userid

インフォメーション・カタログを保管するデータベースに必要なユーザー ID を指定します。

password

userid に関連するパスワード。

tagfile

インポートするタグ言語ファイルのディレクトリー・パスとファイル名。

logfile

メッセージ、エラー、およびトレース情報がログに記録されるファイルの、ディレクトリー・パスとファイル名。

iconpath

オプション。アイコンをインポートするときのみ使用します。インポート・ファンクションが使用するアイコン・パスを指定します。

checkpoint

インポート・ファンクションがタグ・ファイルの先頭から再始動することを指示します。このパラメーターが指定されなかった場合、インポート・ファンクションは、正常に行われた最後のデータベース・コミットから再始動します。

Trace

インフォメーション・カタログ・センターのトレース・ファイルに送られるトレース情報のレベル。このキーワードが指定された場合にトレースはオンになります。このキーワードが指定されなかった場合は、トレースはオフとなります。トレース・ファイルは、ロギング・ディレクトリー内にあります。

関連タスク:

- インフォメーション・カタログ・センター 管理ガイド の『タグ言語ファイルのインポート』

関連資料:

- データウェアハウス・センター アプリケーション統合ガイド の『タグ言語ファイルからのメタデータのインポート』
- データウェアハウス・センター アプリケーション統合ガイド の『インフォメーション・カタログ・マネージャーからのメタデータのエクスポート』
- インフォメーション・カタログ・センター 管理ガイド の『タグ言語』

オブジェクト・タイプの更新

オブジェクト・タイプは、「インフォメーション・カタログ・センター」ウィンドウまたはタグ言語を使用して更新することができます。オブジェクト・タイプの名前、アイコン、およびプロパティーの値リストを変更できます。また、オブジェクト・タイプにプロパティーを追加することもできますが、追加したプロパティーはオプションにしか

できません。必須にすることはできません。「オブジェクト・タイプの更新 (Update Object Type)」ウィンドウを使用すると、プログラムと関連付けたり、関連を解除したりすることもできます。

オブジェクトによってプロパティの値として使用されている場合、リストから制約エントリを除去することはできません。まず最初に、制約の変更を使用可能にするために、その値のプロパティを持つすべてのオブジェクトを変更する必要があります。変更が必要なオブジェクトを検出するために、その値のオブジェクトのオブジェクト・タイプを検索します。

前提条件:

インフォメーション・カタログで、いくつかのオブジェクト・タイプを定義してあることが必要です。

インフォメーション・カタログに対する管理者特権を持っていることが必要です。

手順:

「インフォメーション・カタログ・センター」ウィンドウを使用してオブジェクト・タイプを更新する場合は、以下のことを実行します。

1. インフォメーション・カタログのメイン・ウィンドウで「**管理 (Administration)**」フォルダーを展開する。
2. 「**オブジェクト・タイプ (Object types)**」フォルダーをクリックして、オブジェクト・タイプをリストする。
3. 更新したいオブジェクト・タイプで右マウス・ボタンをクリックする。
4. 「**プロパティ (Properties)**」をクリックします。

「オブジェクト・タイプの更新 (Update Object Type)」ウィンドウがオープンされません。

インフォメーション・カタログ・センターのタグ言語を使用してオブジェクト・タイプを更新する場合は、以下のことを実行します。

1. タグ言語ファイルに以下の行を入力する。

```
:ACTION.OBJTYPE(UPDATE)
:OBJECT.TYPE(object_type_short_name)
```
2. オブジェクト・タイプ名を変更する場合は、以下の行を追加する。

```
EXTNAME(new_object_type_name)
```
3. オブジェクト・タイプの記述を変更する場合は、以下の行を追加する。

```
DESCRIPTION(new_object_type_description)
```
4. オブジェクト・タイプのアイコンを変更するには、以下の行を追加する。

```
ICWFILE(new_icon_filename)
```

各キーワードの後の括弧内に適切な値を入力します。

キーワード	値
TYPE	更新するオブジェクト・タイプの短縮名。
EXTNAME	オブジェクト・タイプの新規名。名前についての規則は以下のとおりです。 <ul style="list-style-type: none">• 最大 200 文字。• NULL 文字を含んではならない。• すべてを空白文字にすることはできない。
DESCRIPTION	オブジェクト・タイプの新規記述。記述についての規則は以下のとおりです。 <ul style="list-style-type: none">• 最大 254 文字。• NULL 文字を含んではならない。• すべてを空白文字にすることはできない。
ICWFILE	拡張子を含めた、新しい .gif アイコン・ファイルの名前。タグ言語ファイルをインポートするときに、アイコン・ファイルがあるドライブとパスの情報を IMPORT コマンドの一部として指定します。

5. オプションのプロパティを追加するには、ユーザーのタグ言語ファイルに以下の行を入力する。

```
:ACTION.OBJTYPE(APPEND)
:OBJECT.TYPE(object_type_short_name)
:PROPERTY.SHRTNAME(new_property_short_name) DT(data_type) DL(size)
      UISEQ(0) NULLS(y) EXTNAME(new_property_name)
```

各キーワードの後の括弧内に適切な値を入力します。

オブジェクト・タイプを作成したあとで追加するすべてのプロパティは、オプションのプロパティでなければならないので、UISEQ は 0 で NULLS は Y でなければなりません。

キーワード	値
TYPE	更新するオブジェクト・タイプの短縮名。
SHRTNAME	プロパティの短縮名。プロパティの短縮名の規則は以下のとおりです。 <ul style="list-style-type: none">• 大文字小文字を区別する。• 最大 18 文字 (SBCS)。

- 最初の文字は大文字または小文字の英字、@、# でなければならない。
- 続く文字は大文字または小文字の英数字、@、#、または _ でなければならない。\$、先頭の空白、または組み込み空白は不可。
- SQL 予約語は使用できない。
- 固有でなければならない。入力した名前がこのオブジェクト・タイプにすでにあると、インフォメーション・カタログ・センターは別の名前を要求します。

DT

データ・タイプ :

I (INTEGER)

4 バイト

S (SMALLINT)

2 バイト

G (BIGINT)

8 バイト

E (DECIMAL)

16 バイト

U (DOUBLE)

8 バイト

R (REAL)

4 バイト

B (BLOB)

0 バイトから 2 GB

O (CLOB)

0 バイトから 2 GB の文字

C (CHAR)

最大 254 文字

V (VARCHAR)

最大 4,000 文字

L (LONG VARCHAR)

最大 32,700 文字

T (TIMESTAMP)

次の形式で 26 文字

yyyy-mm-dd-hh.mm.ss.nnnnnn

M (TIME)

次のフォーマットの 15 文字の時刻

hh.mm.ss.nnnnnn

D (DATE)

次のフォーマットの 10 文字の日付

yyyy-mm-dd

DL

プロパティのサイズ

UISEQ

UI 中でのこのプロパティの位置 : **1— 16**。
プロパティを UI の一部にしたい場合にのみ、
このキーワードを組み込みます。

NULLS

必要な項目かどうか。

N 必要な項目です。

Y 必要な項目ではありません。

EXTNAME

プロパティの拡張された記述名。名前についての
規則は以下のとおりです。

- 最大 200 文字。
- NULL 文字を含んではならない。
- すべてを空白文字にすることはできない。

関連タスク:

- インフォメーション・カタログ・センター 管理ガイド の『オブジェクト・タイプの定義』
- インフォメーション・カタログ・センター 管理ガイド の『オブジェクト・タイプの削除』

関連資料:

- インフォメーション・カタログ・センター 管理ガイド の『OBJECT』
- インフォメーション・カタログ・センター 管理ガイド の『インフォメーション・カタログ・センターの定義済みオブジェクト・タイプ』
- インフォメーション・カタログ・センター 管理ガイド の『ACTION.OBJTYPE』
- インフォメーション・カタログ・センター 管理ガイド の『PROPERTY』

インフォメーション・カタログ・センターの説明データに対して有効なデータ・タイプ

以下の表は、インフォメーション・カタログ・センターの説明データに対して有効なデータ・タイプを示しています。

表 223. インフォメーション・カタログ・センターの説明データに対して有効なデータ・タイプ

データ・タイプ	説明
INTEGER (I)	整数は、精度が 10 桁の 4 バイト整数です。整数の範囲は -2,147,483,648 ~ +2,147,483,647 です。
SMALLINT (S)	短整数は、精度が 5 桁の 2 バイト整数です。短整数の範囲は -32,768 ~ 32,767 です。
BIGINT (G)	長整数は、精度が 19 桁の 8 バイト整数です。長整数の範囲は、-9,223,372,036,854,775,808 ~ +9,223,372,036,854,775,807 です。
DECIMAL (E)	10 進数値は、暗黙的な小数点を持つバックス 10 進数です。小数点の位置は、その数値の精度および位取りによって決まります。位取り (数値の小数部分の桁数) を、負の数、または精度より大きい数にすることはできません。最大の精度は 31 桁です。
DOUBLE (U)	倍精度の浮動小数点数は、64 ビットの実数の近似値です。この数値としては、ゼロ、-1.79769E+308 ~ -2.225E-307 の範囲内、あるいは 2.225E-307 ~ 1.79769E+308 の範囲内が可能です。
REAL (R)	単精度の浮動小数点数は、32 ビットの実数の近似値です。この数値としては、ゼロ、-3.402E+38 ~ -1.175E-37 の範囲内、あるいは 1.175E-37 ~ 3.402E+38 の範囲内が可能です。
BLOB (B)	バイナリー・ラージ・オブジェクト。サイズが 0 バイト ~ 2 ギガバイト - 1 バイトの範囲内である一連のバイト。 データ・タイプが BLOB のプロパティを、固有 ID プロパティとして指定することはできません。
CLOB (O)	文字ラージ・オブジェクト。サイズが 0 バイト ~ 2 ギガバイト - 1 バイトの範囲内である一連の文字 (単一バイト、マルチバイト、あるいはその両方)。 データ・タイプが CLOB のプロパティを、固有 ID プロパティとして指定することはできません。
CHAR (C)	固定長文字ストリング (1 ~ 254 バイト)。 値がプロパティについて定義されているデータ長よりも短い場合は、値の右側を末尾空白で埋めてください。

表 223. インフォメーション・カタログ・センターの説明データに対して有効なデータ・タイプ (続き)

データ・タイプ	説明
TIMESTAMP (T)	次の形式の 26 文字のタイム・スタンプ: yyyy-mm-dd-hh.mm.ss.nnnnnn
TIME (M)	次のフォーマットの 15 文字の時刻 : hh.mm.ss.nnnnnn
DATE (D)	次のフォーマットの 10 文字の日付 : yyyy-mm-dd
LONG VARCHAR (L)	長い可変長文字ストリング (1 ~ 32,700 バイト)。 データ・タイプが LONG VARCHAR のプロパティを、固有 ID プロパティとして指定することはできません。
VARCHAR (V)	可変長文字ストリング (1 ~ 32,672 バイト)。

LOB プロパティ (BLOB または CLOB) は、カタログが保管されている場所のデータベース・サーバーでサポートされている限界までの長さを持つことができます。BLOB または CLOB プロパティに情報を書き込んでいるときに、情報は、データベース・クライアントによってメモリーに入れられます。LOB プロパティに 200-300 K バイト以上保管するには、クライアント・プラットフォームでインフォメーション・カタログ・センターを実行している Java 仮想計算機 (JVM) に、すべての BLOB または CLOB 情報を入れるのに十分なスペースが必要です。クライアント・プラットフォームの物理メモリーおよびページング・スペースもまた、大きな JVM を実行するのに十分でなければなりません。JVM のサイズは、インフォメーション・カタログ・センターを開始するスクリプトで調整できます。物理メモリーおよびページング・スペースの限度は、クライアント・オペレーティング・システムで構成されます。物理メモリーおよびページング・スペースの限度の変更については、オペレーティング・システムのドキュメンテーションを参照してください。

タグ言語ファイルのインポート中に、インフォメーション・カタログ・センターは、要求の妥当性検査を行い、受け入れる前に、自動的に可変値から末尾ブランクを除去し、適宜その長さを調整します。

必要な値は指定する必要があります。指定しないとエラーとなります。

関連概念:

- データウェアハウス・センター アプリケーション統合ガイド の『データウェアハウス・センターのためのオブジェクト定義』

関連資料:

- インフォメーション・カタログ・センター 管理ガイド の『タグ言語』

ERwin からインフォメーション・カタログ・センターへのマッピング

表 224 から 1105 ページの表 229 には、メイン ERwin オブジェクト属性がインフォメーション・カタログ・センター・タグにどのように対応するかが示されています。

表 224. データベース - Database.tag

ERwin	コマンド行タグ	インフォメーション・カタログ・センター・インターフェース
ダイアグラム名	NAME	データベース名
ダイアグラム作成者	RESPNSBL	データベース所有者
データベース名	DBNAME	データベース名
データベース・バージョン	DBTYPE	RELATIONAL
データベース拡張タイプ	DBETYPE	キーワード
ダイアグラムの記述	SHRTDESC	簡略説明

表 225. 表 - TableOrView.tag

ERwin	コマンド行タグ	インフォメーション・カタログ・センター・インターフェース
表名	NAME	表名
表名	TABLES	表名
データベース名	DBNAME	データベース名
表の所有者	OWNER	表の所有者
表のコメント	SHRTDESC	簡略説明
ERwin API	TABLVIEW	定義がビューを表す

表 226. 列 - ColumnOrField.tag

ERwin	コマンド行タグ	インフォメーション・カタログ・センター・インターフェース
列名	NAME	列名
Datatype	DATATYPE	列のデータ・タイプ
長さ	LENGTH	列の長さ
スケール	SCALE	列のスケール
NULL オプション	NULL	列が NULL 可能 (?)
位置	POSNO	列位置
主キー	KEYPOSNO	主キーの列の位置

表 226. 列 - *ColumnOrField.tag* (続き)

ERwin	コマンド行タグ	インフォメーション・カタログ・ センター・インターフェース
ERwin API	ISKEY	列がキーの一部か (?)
ERwin API	UNIKEY	列がユニーク・キーか (?)
データベース名	DBNAME	データベース名
表の所有者	OWNER	表の所有者
表名	TABLES	表名
列のコメント	SHRTDESC	簡略説明
ERwin	ISTEXT	データがテキストか (?)
ERwin API	IDSRES	データの解像度

表 227. モデル - *Model.tag*

ERwin	コマンド行タグ	インフォメーション・カタログ・ センター・インターフェース
ER1 ファイル名	NAME	モデル名
ダイアグラム作成者	RESPNSBL	詳細情報...
ダイアグラムの記述	SHRTDESC	簡略説明

表 228. エンティティ - *Entity.tag*

ERwin	コマンド行タグ	インフォメーション・カタログ・ センター・インターフェース
エンティティ名	NAME	エンティティ名
注	SHRTDESC	簡略説明
定義	LONGDESC	長い記述
エンティティ所有者	RESPNSBL	詳細情報...

表 229. 属性 - *Attribute.tag*

ERwin	コマンド行タグ	インフォメーション・カタログ・ センター・インターフェース
属性名	NAME	属性名
注	SHRTDESC	簡略説明
定義	LONGDESC	長い記述
Datatype	DATATYPE	メンバーのデータ・タイプ
長さ	LENGTH	メンバーの長さ

インストール・サンプル・プログラムのトラブルシューティングのヒント

runGseDemo サンプル・プログラムは、インストールでの問題を明らかにするように設計されています。インストール検査中に、特定のシステム的問題の診断に役立つエラー・メッセージを受け取ることがあります。ほとんどのエラー・メッセージは、少数の一般的な問題が原因で出されます。これらのエラーを避けるには、インストール検査プログラムを実行するたびに、以下の作業を行ってください。

- DB2 Spatial Extender 製品を適切な環境にインストールしたことを確認する。
- 地理情報操作が関連付けられていない、新規のデータベースを使用する。
- アプリケーション・ヒープ・サイズ用のデータベース構成パラメーター値を増やす。

Windows 上のサンプル・プログラムの注: Windows システムでは、製品 CD 内のサンプル・プログラムのバージョンが完全ではなく、エラーを生成する場合があります。サンプル・プログラムの使用に関する更新および詳細については、Spatial Extender の Web サイト: www.ibm.com/software/data/spatial/ を参照してください。

このセクションの残りの部分では、トラブルシューティングを行うときに、よく問題となる点をリストしています。

データベースがすでに地理情報処理可能になっている

インストールを検証するデータベースが新しいものであり、地理情報操作が関連付けられていないことをチェックしてください。これが該当する場合、サンプル・プログラムは失敗します。

サンプル・プログラムを実行するデータベースがすでに地理情報処理可能になっている場合は、以下のエラー・メッセージを受け取ります。

```
Enabling database logtst...
Returning from ENABLE_DB:
Return code = -14
Return message text =
GSE0014E The database has already been enabled for spatial operations.
```

この問題を修正するには、データベースをドロップして、「Spatial Extender のインストールの検査」にあるステップを繰り返します。

アプリケーション・ヒープ・サイズ用のデータベース・マネージャー構成パラメーター値

APPLHEAPSZ が適切な値に設定されていない場合は、地理情報操作用にデータベースを使用可能にするようにしている間に、以下のエラー・メッセージを受け取ります。

```
GSE0213N A bind operation failed.  
SQLERRR = "SQL0001N Binding or precompilation  
did not complete successfully.  
SQLSTATE=00000".SQLSTATE=57011
```

アプリケーション・ヒープ・サイズ用のデータベース構成パラメーター値を増やすには、以下のように入力します。

```
db2 update db cfg for database_name using APPLHEAPSZ 2048
```

2048 で十分でない場合は、APPLHEAPSZ パラメーターを 256 ずつ増やしてください。

索引アドバイザーの使用

DB2 Spatial Extender には、索引アドバイザー と呼ばれるユーティリティーが準備されています。これを使用すると、以下のことが行えます。

- シミュレートされた格子索引を作成し、実際の格子索引のために満足のいくモデルとなるまでその索引を調整する。
- 既存の格子索引を維持すべきか、置き換えるべきかを決定する。

このセクションでは、索引アドバイザーの使用方法を説明し、索引アドバイザーを呼び出すコマンドの構文とパラメーターについて解説します。

前提条件:

索引を作成するデータにアクセスするには、以下の要件があります。

- 行のサブセットのみを分析するために ANALYZE 文節を使用する場合は、USER TEMPORARY 表スペースを利用可能にしておくこと。この表スペースのページ・サイズは、少なくとも 8 KB に設定し、USE 特権を持っていること。
- ユーザー ID にこの表に対する SELECT 特権があること。

シミュレートされた格子索引を作成および調整するための手順:

シミュレートされた索引を作成し、調整するためには、以下のようになります。

1. 索引アドバイザーを呼び出すコマンドを開始する準備を行う。
2. 作成する索引に対する推奨格子セル・サイズを、索引アドバイザーを使って求める。
3. 推奨格子セル・サイズに基づく統計を索引アドバイザーに集めさせる。
4. 推奨格子セル・サイズが検索でいかに機能しているか調べる。
5. 統計結果に満足できない場合は、ステップ 2、3、および 4 を繰り返して、統計を改善してください。

以下に、詳しく説明します。

1. 索引アドバイザーを呼び出すコマンドを開始する準備を行う。

- a. オペレーティング・システムのプロンプトから `gseidx CONNECT TO mydb<` と入力する。ここで、`mydb` は、索引を作成する列を持つ表が入っているデータベースの名前です。
 - b. 接続に別のユーザー ID を使用したい場合は、その ID とそのパスワードを指定する。ID の前にはキーワード `USER` を、パスワードの前にはキーワード `USING` を付けます。たとえば、以下のように指定します。

```
gseidx CONNECT TO mydb USER user ID USING ppassword
```
2. 作成する索引に対する推奨格子セル・サイズを、索引アドバイザーを使って求める。
 - a. 索引アドバイザーを呼び出すコマンドを入力する。以下のいずれかが可能です。
 - 単に格子セル・サイズを要求する。このためには、`ADVISE` キーワードを指定します。
 - 通常の照会でアクセスするデータに基づく格子セル・サイズを要求する。このためには、索引を作成する列内のデータ・エクステント全体を決めます。次に、このエクステントのどの程度 (%) が通常の照会でアクセスされるかを判断します。次に、`ADVISE QUERY BOX` キーワードを指定し、`query-box-size` パラメーターをこのパーセンテージに設定します。
 - b. コマンドを実行する。索引アドバイザーは、推奨格子セル・サイズを戻します。
 3. 推奨格子セル・サイズに基づく統計を索引アドバイザーに集めさせる。
 - a. もう一度コマンドを入力する。今回は、索引アドバイザーが推奨する格子セル・サイズに基づく統計を要求するキーワードを指定します。索引が付けられたデータすべてについての統計、あるいは、そのデータのサブセットについての統計を要求できます。
 - 索引が付いたすべてのデータに関する統計を入手するには、`simulated-index` 文節を指定する。この文節で、ステップ 2b で戻された格子セル・サイズ (複数も可) を指定します。
 - 行のサブセットにある、索引が付いたデータに関する統計を入手するには、以下のことを指定する。
 - `simulated-index` 文節。この文節で、ステップ 2b で戻された格子セル・サイズ (複数も可) を指定します。
 - `ANALYZE` キーワードとそのパラメーター。このコマンド・エレメントによって、統計を得るためにアドバイザーが分析すべき行数または行のパーセンテージを指定できます。
 - b. コマンドを実行する。索引アドバイザーは、シミュレートされた索引についての統計を戻します。

ヒント: 表に含まれる行が 1000 行を超えていて、そのデータがおおよそ均等に分散している場合は、1000 行の分析を要求してください。データが均等に分散していない場合は、1000 行を超える分析を要求します。

4. 推奨格子セル・サイズが検索でいかに機能しているか調べる。これは、ステップ 3b で戻された統計を調べることによって行えます。

ヒント:

- 統計の「索引項目/図形の比率 (Index Entry/Geometry ratio)」は 1 ~ 4 でなければなりません。
 - 統計の「索引項目数 (Number of Index Entries)」は「図形数 (Number of Geometries)」に等しいかそれに近い数になるようにしてください。
5. 統計結果に満足できない場合は、ステップ 2、3、および 4 を繰り返して、統計を改善してください。

ヒント: 索引アドバイザーで行のサブセットを分析し、その統計が満足のいくものであるときは、分析する行数を毎回増やしながら、ステップ 2、3、および 4 を複数回繰り返します。各反復で、統計が満足いくようになるまで、これを続けます。

既存の格子索引を維持するかどうかを判断するための手順:

既存の格子索引に関する統計を見れば、その格子索引が効率よいものであるか、あるいは、もっと効率のよい索引に置き換える必要があるかが分かります。これらの統計を入手する手順、および必要であれば索引を置き換える手順は、以下のとおりです。

1. 索引アドバイザーを呼び出すコマンドを開始する準備を行う。
2. 既存の索引の格子セル・サイズに基づく統計を索引アドバイザーに集めさせる。
3. これらの格子セル・サイズが検索でどのようにうまく機能しているか判断する。
4. 統計が満足のいくものでなければ、
 - a. シミュレートされた格子索引を作成し、実際の格子索引のために満足のいくモデルとなるまでその索引を調整する。
 - b. 既存の索引をドロップして、シミュレートされた索引に合う索引と置き換える。

以下に、詳しく説明します。

1. 索引アドバイザーを呼び出すコマンドを開始する準備を行う。
 - a. オペレーティング・システムのプロンプトから `gseidx CONNECT TO mydb` と入力する。ここで、`mydb` は、索引を提供またはテストする列を持つ表が入っているデータベースの名前です。
 - b. 別のユーザー ID を使用してデータベースに接続したい場合は、その ID とそのパスワードを指定する。ID の前にはキーワード `USER` を、パスワードの前にはキーワード `USING` を付けます。たとえば、以下のように指定します。

```
gseidx CONNECT TO mydb USER user ID USING password
```
2. 既存の索引の格子セル・サイズに基づく統計を索引アドバイザーに集めさせる。索引が付けられたデータすべてについての統計、あるいは、そのデータのサブセットについての統計を要求できます。

- 索引が付いたすべてのデータに関する統計を入手するには、**GET GEOMETRY STATISTICS | INFORMATION** コマンドを入力し、existing-index 文節を指定する。DETAIL キーワードを含めます。
- 行のサブセットにある、索引が付いたデータに関する統計を入手するには、**GET GEOMETRY STATISTICS | INFORMATION** コマンドを入力し、以下のことを指定する。
 - existing-index 文節。DETAIL キーワードを含めます。
 - ANALYZE キーワードとそのパラメーター。これらのコマンド・エレメントによって、統計を得るために索引アドバイザーが分析する行の数またはパーセンテージを指定できます。

ヒント: 表に含まれる行が 1000 行を超えていて、そのデータがおおよそ均等に分散している場合は、1000 行の分析を要求してください。データが均等に分散していない場合は、1000 行を超える分析を要求します。

コマンドを実行する。索引アドバイザーは、既存索引についての統計を戻します。

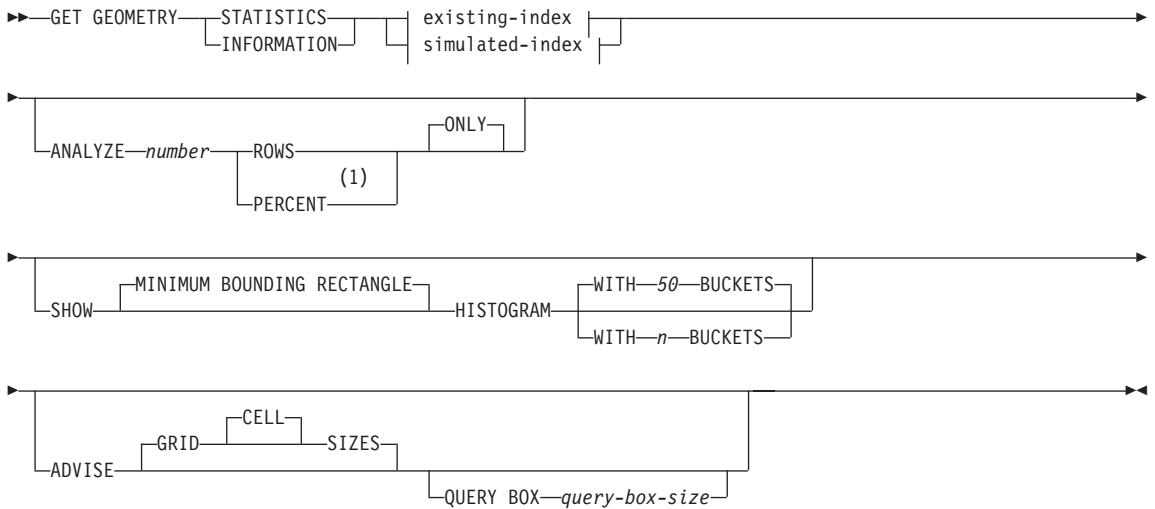
3. 既存の索引の格子セル・サイズが、検索でどの程度うまく機能しているかを判断する。これは、ステップ 2 で戻された統計を調べることによって行えます。

ヒント:

- 統計の「索引項目/図形の比率 (Index Entry/Geometry ratio)」は 1 ~ 4 でなければなりません。
 - 統計の「索引項目数 (Number of Index Entries)」は「図形数 (Number of Geometries)」に等しいかそれに近い数になるようにしてください。
4. 統計が満足のいくものでなければ、
 - a. 既存の索引が定義されている列について、シミュレートされた格子索引を作成する。実際の格子索引のための満足のいくモデルとなるまで、このシミュレートされた索引を調整します。
 - b. 既存の索引をドロップして、シミュレートされた索引に合う索引と置き換える。

GET GEOMETRY STATISTICS | INFORMATION コマンドの構文とパラメーター:

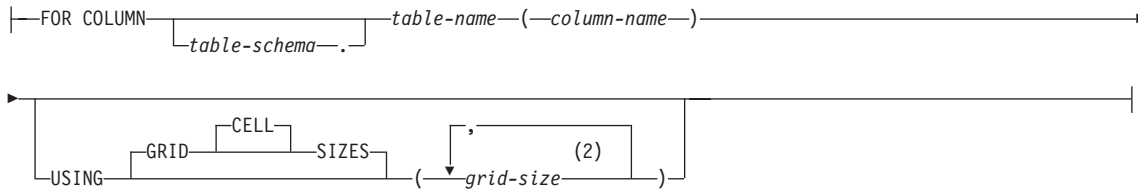
このセクションでは、**GET GEOMETRY STATISTICS | INFORMATION** コマンドの構文とパラメーターを説明します。1111 ページの図 17 は、構文を示しています。



existing-index:



simulated-index:



注:

- 1 PERCENT キーワードの代わりに、パーセント記号 (%) を指定できます。
- 2 1、2、または 3 レベルの格子レベルにセル・サイズを指定できます。

図 17. GET GEOMETRY STATISTICS | INFORMATION コマンドの構文

以下に、GET GEOMETRY STATISTICS | INFORMATION コマンドのキーワードとパラメーターを説明します。

existing-index

どの既存索引について統計を集めるかを指定する既存索引名。

index-schema

既存の索引を含んでいるスキーマの名前。

index-name

既存索引の非修飾名。

DETAIL

各格子レベルについて以下の情報を示します。

- 格子セルのサイズ
- 索引が付いた図形の数
- 索引項目数
- 図形を含んでいる格子セルの数
- 図形当たりの平均索引項目数
- 格子セル当たりの平均図形数
- 最も多く図形を含んでいるセル内の図形の数
- 含まれている図形が最も少ないセル内の図形の数
- 図形のセル間相対頻度を示すヒストグラム

simulated-index

表の列と、その列用にシミュレートされた索引を指します。

table-schema

シミュレートされた索引の対象となっている列を持つ表が入っているスキーマの名前

table-name

シミュレート索引の対象となっている列を持つ表の非修飾名。

column-name

シミュレート索引の対象となっている表列の非修飾名。

grid-size

シミュレートされた索引の各格子レベル (最も密、中程度、最も粗い) のセルのサイズ。少なくとも 1 つのレベルのセル・サイズを指定する必要があります。レベルを含めたくない場合は、レベルに対して格子セル・サイズを指定しないか、またはレベルに格子セル・サイズとして 0 を指定します。

grid-size パラメーターを指定すると、索引アドバイザーは、*existing-index* 文節に **DETAIL** キーワードが指定されている場合と同じ種類の統計を戻します。

ANALYZE *number* **ROWS | PERCENT ONLY**

表の行のサブセット内のデータに関する統計を集めるための指示。このサブセットに含めるおよその行数またはおよそのパーセンテージを指定します。

SHOW MINIMUM BOUNDING RECTANGLE HISTOGRAM

図形の最小外接長方形 (MBR) のサイズ、および MBR が同じサイズである図形の数を示すグラフを表示させます。

WITH *n* **BUCKETS**

すべての分析された図形の MBR に関するグループ化の数を参照。小さい MBR は

その他の小さい図形とまとめてグループ化されます。大きい MBR は、他の大きい図形とともにグループ化されます。デフォルトでは 50 バケットを使用します。

ADVISE GRID CELL SIZES

最適化サイズに近い格子セル・サイズを計算します。

QUERY BOX *query-box-size*

最適化サイズに近い照会ボックス・サイズを計算します。*query box* は、通常の照会でアクセスされる列のデータ・エクステント全体のパーセンテージです。その列のデータの通常の照会アクセスが 1% ならば、0.01 と指定します。アクセスが 0.03% ならば、0.0003 と指定します。

例:

以下の例は、完全修飾名が TOOLBOX.GRIDIRON である既存の格子索引に関する詳細情報を戻すための要求です。

```
GET GEOMETRY STATISTICS
FOR INDEX TOOLBOX.GRIDIRON
DETAIL
```

以下の例は、表 NEWTON.EXPPOINT8 内の列 POINT に対してシミュレートされた索引についての詳細情報を戻すための要求です。この索引には、2 つの格子が定義されています。1 つはセル・サイズが .5 のもの、もう 1 つはセル・サイズが 10 のものです。索引アドバイザーは、列 POINT の 1000 行の分析を基にした情報を戻します。

```
GET GEOMETRY STATISTICS
FOR COLUMN NEWTON.EXPPOINT8(POINT)
USING GRID SIZES (0.5, 10)
ANALYZE 1000 ROWS ONLY
```

DB2 Spatial Extender サンプル・プログラム

DB2[®] Spatial Extender サンプル・プログラムの runGseDemo には、目的が 2 つあります。このプログラムは、DB2 Spatial Extender 用のアプリケーション・プログラミングに慣れるため、および、DB2 Spatial Extender のインストールが正しく終了したことを検証するために使用できます。DB2 Spatial Extender のインストールを検証するステップについては、別のトピックで説明します。

- UNIX では、runGseDemo プログラムは以下のパスで見つけることができます。

```
$HOME/sqllib/samples/spatial
```

\$HOME は、インスタンス所有者のホーム・ディレクトリーです。

- Windows では、runGseDemo プログラムを以下のパスに配置できる。

```
c:\Program Files\IBM\sqllib\samples\spatial
```

c:\Program Files\IBM\sqllib は、DB2 Spatial Extender をインストールしたディレクトリーです。

Windows 上のサンプル・プログラムの注: Windows® システムでは、製品 CD 内のサンプル・プログラムのバージョンが完全ではなく、エラーを生成する場合があります。サンプル・プログラムの使用に関する更新および詳細については、Spatial Extender の Web サイト: www.ibm.com/software/data/spatial/ を参照してください。

DB2 Spatial Extender の runGseDemo サンプル・プログラムを使用すると、アプリケーション・プログラミングが簡単になります。このサンプル・プログラムを使用すると、データベースを地理情報操作で使用可能にできるとともに、そのデータベース内のデータに対する地理情報分析が行えます。このデータベースには、顧客情報と洪水地帯の情報の入った表が入ります。この情報から、どの顧客が洪水被害に遭う危険があるかを判断することができます。

このサンプル・プログラムでは、以下のことが行えます。

- 地理情報が使用可能なデータベースの作成と維持に通常必要なステップを知ること
- アプリケーション・プログラムから地理情報ストアード・プロシージャーを呼び出す方法を理解すること
- 独自のアプリケーションにサンプル・コードをカット・アンド・ペーストすること

DB2 Spatial Extender Web サイト www.ibm.com/software/data/spatial/s から他のサンプル・プログラムを利用することもできます。

DB2 Spatial Extender 用のタスクをコーディングするには、以下のサンプル・プログラムを使用してください。たとえば、DB2 Spatial Extender ストアード・プロシージャーを呼び出すためのデータベース・インターフェースを使用するアプリケーションを作成する場合を考えましょう。このサンプル・プログラムからコードをコピーして、アプリケーションをカスタマイズすることができます。DB2 Spatial Extender のプログラミング・ステップに慣れていない場合は、このサンプル・プログラムを実行すれば、各ステップを詳細に知ることができます。サンプル・プログラムを実行するための説明については、このセクションの最後にある関連リンクを参照してください。

以下の表には、サンプル・プログラムのステップ、関連するストアード・プロシージャー、各ステップの説明がまとめてあります。各ステップで、アクションを実行し、さらに、多くの場合、そのアクションの反対のアクション、すなわち取り消しを行います。たとえば、最初のステップでは、地理情報データベースを使用可能にし、次に、地理情報データベースを使用不可にします。このようにして、多くの Spatial Extender ストアード・プロシージャーに慣れていきます。

ストアード・プロシージャーを呼び出す C 関数を括弧内に示してあります。ストアード・プロシージャーについての詳細は、このセクションの最後にある関連リンクを参照してください。

表 230. DB2 Spatial Extender サンプル・プログラム・ステップ

ステップ	アクションと説明
地理情報データベースを使用可能または使用不可にする	<ul style="list-style-type: none"> <li data-bbox="585 213 1231 430">• 地理情報データベースを使用可能にする (stEnableDB) これは、DB2 Spatial Extender を使用するために必要な最初のステップです。地理情報操作用に使用可能にされているデータベースには、地理情報タイプのセット、地理情報関数セット、地理情報述部セット、新規の索引タイプと 1 組の地理情報のカタログ表とビューが入っています。 <li data-bbox="585 439 1231 751">• 地理情報データベースを使用不可にする (stDisableDB) このステップは、通常、間違ったデータベースに対して地理情報機能を使用可能にした場合、あるいは、そのデータベースで地理情報操作を実行する必要がなくなった場合に、実行します。地理情報データベースを使用不可にするときは、地理情報タイプのセット、地理情報関数セット、地理情報述部セット、新規の索引タイプ、およびそのデータベースに関連付けられている地理インフォメーション・カタログ表とビューを除去します。 <li data-bbox="585 760 1231 838">• 地理情報データベースを使用可能にする (stEnableDB) 上記に同じです。
座標系を作成またはドロップする	<ul style="list-style-type: none"> <li data-bbox="585 855 1231 1003">• NORTH_AMERICAN_TEST という名前の座標系を作成する (stCreateCS) このステップで、データベースに新しい座標系を作成します。 <li data-bbox="585 1012 1231 1159">• NORTH_AMERICAN_TEST という名前の座標系をドロップする (stDropCS) このステップで、データベースから座標系 NORTH_AMERICAN_TEST をドロップします。 <li data-bbox="585 1168 1231 1345">• NORTH_AMERICAN という名前の座標系を作成する (stCreateCS) このステップは、新しい座標系 NORTH_AMERICAN を作成します。これは、以下のステップで作成される地理情報基準システムで使用されます。

表 230. DB2 Spatial Extender サンプル・プログラム・ステップ (続き)

ステップ	アクションと説明
地理情報基準システムを作成またはドロップする	<ul style="list-style-type: none"> <li data-bbox="561 213 1220 534"> <p>• SRSDEMOTEST という名前の地理情報基準システムを作成する (stCreateSRS)</p> <p>このステップは、地理情報基準システム (SRS) を解釈するために使用する新しい SRS を定義します。この SRS には、地理情報使用可能データベースの列に保管できるフォーマットで、図形データが収められます。SRS が特定の地理情報列に対して登録されると、その地理情報列に適用される座標が、対応する CUSTOMERS 表列に保管できるようになります。</p> <li data-bbox="561 543 1220 725"> <p>• SRSDEMOTEST という名前の SRS をドロップする (stDropSRS)</p> <p>このステップは、データベースで SRS を必要としなくなった場合に実行します。SRS をドロップするときは、データベースから SRS 定義を取り除きます。</p> <li data-bbox="561 734 1220 765"> <p>• SRSDEMO1 という名前の SRS を作成する (stCreateSRS)</p>
地理情報表を作成し、データを入れる	<ul style="list-style-type: none"> <li data-bbox="561 777 1220 808"> <p>• CUSTOMERS 表を作成する (sqlSetupTables)</p> <li data-bbox="561 817 1220 930"> <p>• CUSTOMERS 表にデータを入れる (sqlSetupTables)</p> <p>CUSTOMERS 表には、数年にわたってデータベースに保管されているビジネス・データが入っています。</p> <li data-bbox="561 939 1220 1156"> <p>• LOCATION 列を追加して CUSTOMER 表を変更する (sqlSetupTables)</p> <p>ALTER TABLE ステートメントによってタイプが ST_Point の新しい列 (LOCATION) を追加します。この列には、以下のステップでアドレス列をジオコーディングすることによって、データが取り込まれます。</p> <li data-bbox="561 1164 1220 1414"> <p>• OFFICES 表を作成する (sqlSetupTables)</p> <p>OFFICES 表には、保険会社の各オフィスのセールス・ゾーンが他のデータとともに入っています。この表全体には、後のステップで、非 DB2 データベースから属性データが入れられます。その後続のステップには、SHAPE ファイルから OFFICES 表に属性データをインポートする作業も入っています。</p>

表 230. DB2 Spatial Extender サンプル・プログラム・ステップ (続き)

ステップ	アクションと説明
列にデータを入れる	<ul style="list-style-type: none"> <li data-bbox="585 215 1244 319">• DB2SE_USA_GEOCODER という名前のジオコーダーによって、CUSTOMERS 表の LOCATION 列のためのアドレス・データをジオコードする (stRunGC) このステップでは、ジオコーダー・ユーティリティーを呼び出して、地理情報のバッチ・ジオコーディングを行います。バッチ・ジオコーディングは、通常、表の大部分をジオコードする、または再ジオコードする必要があるときに行われます。 <li data-bbox="585 517 1244 725">• SRS SRSDEMO1 を使用して、先に作成した OFFICES 表に SHAPE ファイルからロードする (stImportShape) このステップは、既存の地理情報データを SHAPE ファイルの形式で OFFICES 表にロードします。OFFICES 表が存在するので、LOAD ユーティリティーは、既存の表に新しいレコードを追加します。 <li data-bbox="585 743 1244 951">• SRS SRSDEMO1 を使用して、FLOODZONES 表を作成し、SHAPE ファイルからロードする (stImportShape) このステップは、SHAPE ファイルの形式で FLOODZONES 表に既存データをロードします。表はまだ存在しないので、LOAD ユーティリティーは、ロードする前に表を作成します。 <li data-bbox="585 968 1244 1024">• SRS SRSDEMO1 を使用して、REGIONS 表を作成して SHAPE ファイルからロードする (stImportShape)
ジオコーダーを登録または登録抹消する	<ul style="list-style-type: none"> <li data-bbox="585 1083 1244 1152">• SAMPLEGC2 という名前のジオコーダーを登録する (stRegisterGC) <li data-bbox="585 1170 1244 1239">• SAMPLEGC2 という名前のジオコーダーの登録を抹消する (stUnRegisterGC) <li data-bbox="585 1256 1244 1281">• ジオコーダー SAMPLEGC を登録する (stRegisterGC) <p data-bbox="585 1308 1244 1433">以上のステップで、SAMPLEGC2 という名前のジオコーダーを登録および登録抹消して、次に新しいジオコーダー SAMPLEGC を作成し、サンプル・プログラムで使用できるようにします。</p>

表 230. DB2 Spatial Extender サンプル・プログラム・ステップ (続き)

ステップ	アクションと説明
地理情報索引を作成する	<ul style="list-style-type: none"> • CUSTOMERS 表の LOCATION 列の地理情報格子索引を作成する (sqlCreateIdx) • CUSTOMERS 表の LOCATION 列の地理情報格子索引をドロップする (sqlDropIdx) • CUSTOMERS 表の LOCATION 列の地理情報格子索引を作成する (sqlCreateIdx) • OFFICES 表の LOCATION 列の地理情報格子索引を作成する (sqlCreateIdx) • FLOODZONE 表の LOCATION 列の地理情報格子索引を作成する (sqlCreateIdx) • REGIONS 表の LOCATION 列の地理情報格子索引を作成する (sqlCreateIdx) <p>以上のステップでは、CUSTOMERS、OFFICES、FLOODZONES および REGIONS 表の地理情報格子索引を作成します。</p>
自動ジオコーディングを使用可能にする	<ul style="list-style-type: none"> • ジオコーダー DB2SE_USA_GEOCODER を使用して、CUSTOMERS 表の LOCATION 列のジオコーディングをセットアップする (stSetupGeocoding) <p>このステップは、CUSTOMERS 表の LOCATION 列をジオコーダー DB2SE_USA_GEOCODER に関連付け、対応するジオコーディング・パラメーター値をセットアップします。</p> <ul style="list-style-type: none"> • CUSTOMERS 表の LOCATION 列に対して自動ジオコーディングを使用可能にする (stEnableAutoGC) <p>このステップは、ジオコーダーの自動呼び出しをオンにします。自動ジオコーディングを使用すると、CUSTOMERS 表の LOCATION、ADDRESS、CITY、STATE および ZIP の各列が、後続の挿入および更新操作で互いに同期がとられます。</p>

表 230. DB2 Spatial Extender サンプル・プログラム・ステップ (続き)

ステップ	アクションと説明
CUSTOMERS 表に対する挿入、更新、削除操作	<ul style="list-style-type: none"> • 異なる番地 (street) を持つレコードをいくつか挿入する (sqlInsDelUpd) • 新しい住所でレコードをいくつか更新する (sqlInsDelUpd) • 表からすべてのレコードを削除する (sqlInsDelUpd) <p>これらのステップは、CUSTOMERS 表の ADDRESS、CITY、STATE および ZIP 列に対する挿入、更新、削除の操作を示すものです。自動ジオコーディングが使用可能にされると、これらの列に挿入または更新されたデータは、自動的に LOCATION 列にもジオコードされます。このプロセスは、前のステップで使用可能にされています。</p>
自動ジオコーディングを使用不可にする	<ul style="list-style-type: none"> • CUSTOMERS 表の LOCATION 列に対する自動ジオコーディングを使用不可にする (stDisableAutoGC) • CUSTOMERS 表の LOCATION 列に対するジオコーディングのセットアップを除去する (stRemoveGeocodingSetup) • CUSTOMERS 表の LOCATION 列の地理情報索引をドロップする (sqlDropIdx) <p>これらのステップは、以下のステップの準備として、ジオコーダーの自動呼び出しと地理情報索引を使用不可にします。以下のステップでは、CUSTOMERS 表全体の再ジオコーディングが行われます。</p> <p>推奨： 大量のジオデータ (地理データ) をロードする場合は、データをロードする前に地理情報索引をドロップし、データのロードが終わった後でもう一度索引を作成してください。</p>

表 230. DB2 Spatial Extender サンプル・プログラム・ステップ (続き)

ステップ	アクションと説明
CUSTOMERS 表を再ジオコードする	<ul style="list-style-type: none"> • より低いレベルの精度 (100% ではなく 90%) で、CUSTOMERS 表の LOCATION 列を再ジオコードする (stRunGC) • CUSTOMERS 表の LOCATION 列の地理情報索引を再作成する (sqlCreateIdx) • 精度レベルを 100% ではなく 90% に下げて、自動ジオコーディングを再び使用可能にする (stEnableAutoGC) <p>これらのステップは、バッチ・モードでジオコーダーを実行し、地理情報索引を再作成し、新しい精度レベルで自動ジオコーディングを再び使用可能にします。地理情報管理者がジオコーディング処理での高い障害発生率に気付いたときは、このアクションをとることをお勧めします。精度レベルが 100% に設定されていると、基準データの中に一致アドレスを検出できないためにアドレスのジオコーディングが失敗することがあります。精度レベルを下げると、ジオコーダーは、一致データを検出できやすくなります。バッチ・モードで表を再ジオコードした後、自動ジオコーディングがもう一度使用可能にされ、地理情報索引が再作成されます。これによって、これ以降の挿入と更新操作のために、地理情報索引と地理情報列の漸進的な保守が可能になります。</p>
ビューを作成し、ビューに地理情報列を登録する	<ul style="list-style-type: none"> • CUSTOMERS 表と FLOODZONE 表の結合に基づいて、ビュー HIGHRISKCUSTOMERS を作成する (sqlCreateView) • ビューの地理情報列を登録する (stRegisterSpatialColumn) <p>このステップでは、ビューを作成し、その地理情報列を登録します。</p>

表 230. DB2 Spatial Extender サンプル・プログラム・ステップ (続き)

ステップ	アクションと説明
地理情報を分析する	<ul style="list-style-type: none"> • 各地域でサービスする顧客数を調べる (ST_Within) • 同一地域のオフィスと顧客について、各オフィスから一定の範囲内に住んでいる顧客数を調べる (ST_Within、ST_Distance) • 各地域について、各顧客の平均収入と保険料を調べる (ST_Within) • 各オフィス・ゾーンに重なる洪水ゾーン数を調べる (ST_Overlaps) • オフィスがオフィス・ゾーンの中心に位置していると仮定して、特定の顧客のロケーションから最も近いオフィスを見つける (ST_Distance) • 特定の洪水ゾーンの境界に近いロケーションにある顧客を見つける (ST_Buffer、ST_Intersects) • 特定のオフィスから指定した範囲内のリスクの高い顧客を調べる (ST_Within)
	<p>以上のすべてのステップで、sqlRunSpatialQueries ストアド・プロシージャが使用されます。</p>
	<p>これらのステップは、DB2 SQL の関数と地理情報述部を使用して、地理情報を分析します。DB2 照会オプティマイザは、地理情報列に関する地理情報索引を活用して、可能なかぎり、照会パフォーマンスを向上させます。</p>
地理情報データを SHAPE ファイルにエクスポートする	<ul style="list-style-type: none"> • HIGHRISKCUSTOMERS ビューを SHAPE ファイルにエクスポートする (stExportShape) <p>このステップでは、HIGHRISKCUSTOMERS ビューを SHAPE ファイルにエクスポートする例を示しています。データベース・フォーマットのデータを別のファイル・フォーマットにエクスポートすることによって、他のツール (ArcExplorer など) でその情報が使用できるようになります。</p> <p>このステップは、runGseDemo.c プログラムに入っていますが、参照用としてコメント化されています。サンプル・プログラムを変更して、エクスポート先の SHAPE ファイルを指定し、サンプル・プログラムを実行し直すことができます。</p>

表 230. DB2 Spatial Extender サンプル・プログラム・ステップ (続き)

ステップ	アクションと説明
SDE ファイルをエクスポートおよびインポートする	<ul style="list-style-type: none"> • CUSTOMERS 表を SDE 転送ファイルにエクスポートする (gseExportSDE) • 新しくエクスポートした SDE 転送ファイルからデータをインポートする (gseImportSDE) <p>これらのステップは、SDE 転送ファイルのエクスポートとインポートの例を示しています。</p> <p>これらのステップは、runGseDemo.c プログラムに入っていますが、参照用としてコメント化されています。サンプル・プログラムを変更して、エクスポート先の SHAPE ファイルを指定し、サンプル・プログラムを実行し直すことができます。</p>

関連タスク:

- *Spatial Extender* ユーザーズ・ガイド の『Spatial Extender のインストールの検査』
- 1106 ページの『インストール・サンプル・プログラムのトラブルシューティングのヒント』
- *Spatial Extender* ユーザーズ・ガイド の『DB2 Spatial Extender 用のアプリケーションを作成する』
- *Spatial Extender* ユーザーズ・ガイド の『アプリケーションから DB2 Spatial Extender のストアード・プロシージャを呼び出す』
- *Spatial Extender* ユーザーズ・ガイド の『DB2 Spatial Extender のヘッダー・ファイルを地理情報アプリケーションに組み込む』

XML エクステンダー 管理およびプログラミング

XML Extender の概要

IBM® DB2® エクステンダー・ファミリーは、従来からあるデータ・タイプおよび新しい、すなわち従来型ではないデータ・タイプの両方を処理するための、データおよびメタデータの管理ソリューションを提供します。DB2 XML Extender は、IBM DB2 Universal Database™ (DB2 UDB) の能力と XML (Extensible Markup Language) の柔軟性を統合するのに役立ちます。

DB2 の XML Extender には、XML 文書の保管とアクセスを行ったり、既存のリレーショナル・データから XML 文書を生成したり、XML 文書からリレーショナル表に行を

挿入したりする機能が備わっています。XML Extender には、また、DB2 で XML を管理するための新規のデータ・タイプ、関数、ストアード・プロシージャも備わっています。

XML Extender は、以下のオペレーティング・システムで使用できます。

- Windows[®] NT
- Windows 2000
- AIX[®]
- Sun Solaris
- Linux
- OS/390 および z/OS
- iSeries

DB@ XML Extender での作業を開始する前に、一時ファイルが保管されるロケーションの設定が必要な場合があります。DB2DXXTEMP 環境変数は、XML Extender の一時ファイルのロケーションをコントロールします。変数が設定されていない場合、TMP の値が一時ファイルのロケーションを決定します。DB2DXXTEMP の値を Windows 2000 で設定するには、以下のようになります。

1. DB2 で使用するユーザー ID でログオンしていることを確認します。
2. 「スタート」 -> 「設定」 -> 「コントロール パネル」をクリックします。
3. 「システム」アイコンをダブルクリックします。
4. 「システムのプロパティ」ノートブックの「詳細」ページの「環境変数...」をクリックします。
5. 「システム変数」セクションで「新規」をクリックします。DB2DXXTEMP を変数名に入力し、変数の値を入力します。
6. すべてのウィンドウをクローズして、システムを再始動します。

関連概念:

- *XML Extender 管理およびプログラミングのガイド* の『XML 文書』
- *XML Extender 管理およびプログラミングのガイド* の『DB2 と XML Extender の連携使用の方法』
- *XML Extender 管理およびプログラミングのガイド* の『演習: XML 列への XML 文書の保管』
- *XML Extender 管理およびプログラミングのガイド* の『演習: XML 文書の合成』
- *XML Extender 管理およびプログラミングのガイド* の『XML Extender の開始』

SQLSTATE コードおよび関連メッセージ番号

表 231. SQLSTATE コードおよび関連メッセージ番号

SQLSTATE	メッセージ番号	説明
00000	DXXnnnnI	エラーは発生していません。
01HX0	DXXD003W	パス式で指定されたエレメントまたは属性が XML 文書にありません。
38X00	DXXC000E	XML Extender は、指定されたファイルを開くことができません。
38X01	DXXA072E	XML Extender は、データベースを使用可能にする前に自動的にバインドを試みましたが、バインド・ファイルを検出できませんでした。
	DXXC001E	XML Extender は、指定されたファイルを検出できませんでした。
38X02	DXXC002E	XML Extender は、指定されたファイルからデータを読み取ることができません。
38X03	DXXC003E	XML Extender は、データをファイルに書き込むことができません。
	DXXC011E	XML Extender は、データをトレース制御ファイルに書き込むことができません。
38X04	DXXC004E	XML Extender は、指定されたロケータを操作できませんでした。
38X05	DXXC005E	ファイル・サイズが XMLVarchar サイズより大きいため、XML Extender がファイルからインポートできなかったデータがあります。
38X06	DXXC006E	ファイル・サイズが XMLCLOB のサイズより大きいため、XML Extender がファイルからインポートできなかったデータがあります。

表 231. SQLSTATE コードおよび関連メッセージ番号 (続き)

SQLSTATE	メッセージ番号	説明
38X07	DXXC007E	LOB ロケーターのバイト数がファイル・サイズと等しくありません。
38X08	DXXD001E	スカラー抽出関数が、複数回出現するロケーション・パスを使用しました。スカラー関数は、複数回出現がないロケーション・パスのみを使用することができます。
38X09	DXXD002E	パス式の構文が正しくありません。
38X10	DXXG002E	XML Extender は、オペレーティング・システムからメモリーを割り振ることができませんでした。
38X11	DXXA009E	このストアード・プロシージャは XML 列専用です。
38X12	DXXA010E	XML Extender が列を使用可能にしようとした時、DTD ID を検出できませんでした (DTD ID は、文書アクセス定義 (DAD) ファイルの中で DTD 用に指定された ID)。
38X14	DXXD000E	無効な文書を表の中に保管しようとしてしました。妥当性検査に失敗しました。
38X15	DXXA056E	文書アクセス定義 (DAD) ファイル内の妥当性検査エレメントが正しくないか、またはエレメントがありません。
	DXXA057E	文書アクセス定義 (DAD) ファイル内のサイド表の名前属性が正しくないか、または属性がありません。
	DXXA058E	文書アクセス定義 (DAD) ファイル内の列の名前属性が正しくないか、または属性がありません。

表 231. *SQLSTATE* コードおよび関連メッセージ番号 (続き)

SQLSTATE	メッセージ番号	説明
	DXXA059E	文書アクセス定義 (DAD) ファイル内の列のタイプ属性が正しくないか、または属性がありません。
	DXXA060E	文書アクセス定義 (DAD) ファイル内の列のパス属性が正しくないか、または属性がありません。
	DXXA061E	文書アクセス定義 (DAD) ファイル内の列の <code>multi_occurrence</code> 属性が正しくないか、または属性がありません。
	DXXQ000E	文書アクセス定義 (DAD) ファイル内に必須エレメントがありません。
38X16	DXXG004E	必要パラメーターの NULL 値が XML ストアード・プロシージャに渡されました。
38X17	DXXQ001E	文書アクセス定義 (DAD)、またはこれをオーバーライドするファイルの中の SQL ステートメントが無効です。XML 文書を生成するには SELECT ステートメントが必要です。
38X18	DXXG001E	XML Extender は内部エラーを検出しました。
	DXXG006E	CLI を使用中に XML Extender で内部エラーが発生しました。
38X19	DXXQ002E	システムのメモリーまたはディスク・スペースが不足しています。生成される XML 文書を保管する容量がありません。
38X20	DXXQ003W	ユーザー定義の SQL 照会によって、指定した最大数以上の XML 文書が生成されます。指定された数の文書のみが戻されます。
38X21	DXXQ004E	指定された列は、SQL 照会の結果には含まれません。

表 231. *SQLSTATE* コードおよび関連メッセージ番号 (続き)

SQLSTATE	メッセージ番号	説明
38X22	DXXQ005E	SQL 照会の XML へのマッピングが正しくありません。
38X23	DXXQ006E	文書アクセス定義 (DAD) ファイル内の <code>attribute_node</code> エレメントに名前属性がありません。
38X24	DXXQ007E	文書アクセス定義 (DAD) ファイル内の <code>attribute_node</code> エレメントに、列エレメントまたは <code>RDB_node</code> がありません。
38X25	DXXQ008E	文書アクセス定義 (DAD) ファイル内の <code>text_node</code> エレメントに、列エレメントがありません。
38X26	DXXQ009E	指定された結果表が、システム・カタログ内にありません。
38X27	DXXQ010E	<code>attribute_node</code> または <code>text_node</code> の <code>RDB_node</code> には表が必要です。
	DXXQ040E	
	DXXQ011E	<code>attribute_node</code> または <code>text_node</code> の <code>RDB_node</code> には列が必要です。
	DXXQ017E	XML Extender の生成した XML 文書が大きすぎて、結果表の列の中に入りません。
	DXXQ040E	文書アクセス定義 (DAD) ファイル内の指定されたエレメント名が正しくありません。
38X28	DXXQ012E	DAD の処理中に XML Extender は予期したエレメントを検出できませんでした。
	DXXQ016E	すべての表は、文書アクセス定義 (DAD) ファイル内の先頭エレメントの <code>RDB_node</code> で定義しなければなりません。サブエレメントの表は、先頭エレメントで定義された表と一致しなければなりません。この <code>RDB_node</code> 内の表名は、先頭エレメントの中にはありません。

表 231. *SQLSTATE* コードおよび関連メッセージ番号 (続き)

SQLSTATE	メッセージ番号	説明
38X29	DXXQ013E	エレメントの表または列の名前が、文書アクセス定義 (DAD) ファイル内で指定されていなければなりません。
	DXXQ015E	文書アクセス定義 (DAD) の条件エレメントの条件が、無効な形式です。
38X30	DXXQ014E	文書アクセス定義 (DAD) ファイル内の <code>element_node</code> エレメントに名前属性がありません。
	DXXQ018E	SQL を XML にマップする文書アクセス定義 (DAD) ファイル内の SQL ステートメントに、 ORDER BY 文節がありません。
38X31	DXXQ019E	エレメント <code>objids</code> は、SQL を XML にマップする文書アクセス定義 (DAD) ファイルに列エレメントを持ちません。
38X36	DXXA073E	データベースがバインドされていません。使用可能にする前にデータベースをバインドしてください。
38X37	DXXG007E	サーバーのオペレーティング・システムのロケールが、DB2 コード・ページと矛盾します。
38X38	DXXG008E	サーバーのオペレーティング・システムのロケール設定が、コード・ページ表にありません。
38X41	DXXQ048E	スタイルシートのプロセッサが内部エラーを戻しました。XML 文書またはスタイルシートが有効ではない可能性があります。
38X42	DXXQ049E	指定された出力ファイルは、このディレクトリーにすでに存在しています。

表 231. SQLSTATE コードおよび関連メッセージ番号 (続き)

SQLSTATE	メッセージ番号	説明
38X43	DXXQ050E	UDF にアクセス権がないか、生成可能なすべてのファイル名が使用されているか、またはディレクトリーが存在しないため、UDF は指定のディレクトリーに出力文書のためのユニークなファイル名を作成できませんでした。
38X44	DXXQ051E	入力または出力パラメーターの中に有効な値を持っていないものがあります。
38x33	DXXG005E	このパラメーターはこのリリースではサポートされません。将来のリリースでサポートされます。
38x34	DXXG000E	無効なファイル名が指定されました。

XML Extender をバージョン 7 からバージョン 8 へ移行する

XML Extender バージョン 7.2 を使用している場合は、XML Extender バージョン 8 で既存の XML 使用可能データベースを使用する前に、XML Extender で使用可能な各データベースを移行する必要があります。

移行プログラムは、すでにインストール済みの XML Extender のベース・レベルにしたがって、いろいろなステップを実行します。移行プログラムが実行するステップには、以下のようなステップがあります。

- ユニコードと DBCS データベースで使用する XMLDBCLOB ユーザー定義タイプ (UDT) とユーザー定義関数 (UDF) の作成
- 一時表の使用をサポートするための dxxGenXML および dxxRetrieveXML 用のストアード・プロシージャの追加作成
- パラメーター・スタイル SQL を、引き渡しパラメーターのリンケージ規則として使用するストアード・プロシージャの変更
- スキーマ /DTD 妥当性検査および XSLT 関数の妥当性検査のための新しいユーザー定義関数の作成
- CLOB を戻す新しいストアード・プロシージャ (dxxGenXMLCLOB および dxxRetrieveXMLCLOB) の作成
- スカラー UDF の並列機能の使用を可能にするユーザー定義関数 UDF をドロップして再作成

注: 列を使用可能にしている場合、UDF はドロップも再作成もされず、移行プログラムは警告メッセージを出します。

注: `dxxEnableColl` は `dxxEnableCollection` に名前変更され、`dxxDisableColl` は `dxxDisableCollection` に名前変更されます。ストアード・プロシージャ `db2xml.dxxDisableDB` も追加されます。

ストアード・プロシージャを呼び出す場合、プロシージャ名には、感嘆符 (!) ではなく、ピリオド (.) を使用します。たとえば、`db2xml!dxxEnableColumn` ではなく、`db2xml.dxxEnableColumn` を使用します。

手順:

データベースを移行するには、

1. DB2 XML Extender バージョン 7.2 をインストールします。
2. DB2 コマンド行から以下のように入力します。

```
db2 connect to database name  
db2 bind dxxMigv.lst  
dxxMigv database name
```

3. 列を使用不可にする前に、列を使用可能にするために使用する DAD ファイルを `DB2XML.XML_USAGE` 表から検索します。
4. 以前に XML Extender 用に使用可能にした XML 列を含むすべてのデータベースを再度、使用可能にします。
 - a. `SAVOBJ` を使用してデータをサイド表に保管します。
 - b. XML 列を使用不可にします。
 - c. XML 列を使用可能にします。
 - d. `RSTOBJ` を使用してデータをサイド表にリストアします。サイド表にロング・ネームが付いている場合、リストアの前にサイド表をドロップする必要があります。
 - e. 索引を再作成します。

一般解説書

コマンド解説書

AUTOCONFIGURE

| バッファ・プール・サイズ、データベース構成およびデータベース・マネージャーの
| 構成パラメーターの初期値を計算し、これらの推奨値を適用するオプションと共に表示
| します。

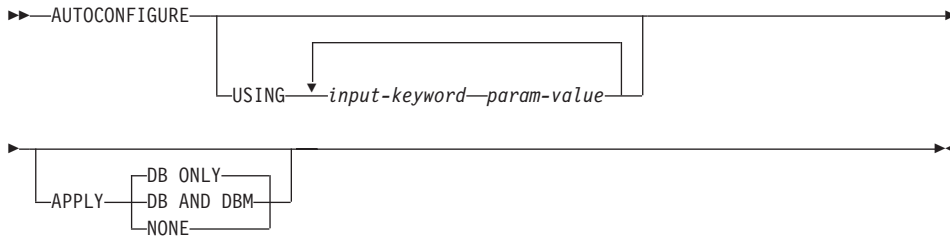
許可:

sysadm。

必要な接続:

データベース。

コマンド構文:



コマンド・パラメーター:

USING input-keyword param-value

表 232. 有効な入力キーワードおよびパラメーター値

キーワード	有効値	デフォルト値	説明
mem_percent	1-100	80	専用にするメモリーのパーセンテージ。他のアプリケーション (オペレーティング・システム以外) がこのサーバーで実行している場合、この値は 100 未満に設定してください。
workload_type	simple、mixed、complex	mixed	単純 (simple) ワークロードは入出力集約の傾向があり大部分がトランザクションであるのに対し、複雑 (complex) ワークロードは CPU 集約の傾向があり大部分が照会です。
num_stmts	1-1 000 000	10	作業単位ごとのステートメント数
tpm	1-50 000	60	1 分ごとのトランザクション
admin_priority	performance、recovery、both	both	より良いパフォーマンス (分あたりのより多いトランザクション数) またはより良いリカバリー時間のための最適化
is_populated	yes、no	yes	データベースがデータで移植されるかどうか
num_local_apps	0-5 000	0	接続されたローカル・アプリケーションの数

表 232. 有効な入力キーワードおよびパラメーター値 (続き)

キーワード	有効値	デフォルト値	説明
num_remote_apps	0-5 000	10	接続されたりリモート・アプリケーションの数
isolation	RR、RS、CS、UR	RR	このデータベースに接続するアプリケーションの分離レベル (反復可能読み取り (RR)、読み取り固定 (RS)、カーソル固定 (CS)、非コミット読み取り (UR))
bp_resizeable	yes、no	yes	バッファ・プールのサイズが変更可能かどうか

APPLY

DB ONLY

すべての推奨される変更を表示し、それらの変更をデータベース構成およびバッファ・プール設定にのみ適用します。APPLY オプションが指定されていない場合、これがデフォルト設定です。

DB AND DBM

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

NONE 推奨される変更を表示しますが、適用はしません。

使用上の注意:

入力キーワードのいずれかが指定されないと、そのパラメーターにはデフォルト値が使用されます。

パーティション・データベース環境では、このコマンドは現行パーティションにしか適用しません。

複数のロジカル・パーティションのあるシステムでは、*mem_percent* パラメーターは、すべてのロジカル・パーティションによって使用されるメモリーのパーセンテージを参照します。たとえば、DB2 がシステムの 80% のメモリーを使用している場合は、ロジカル・パーティションの数に関係なく、80% を指定します。データベースの推奨構成が作成されますが、これは 1 つのロジカル・パーティション用に調整されます。

AUTOCONFIGURE

このコマンドは、現在接続されているデータベース（このデータベースがシステム上で唯一のアクティブ・データベースであると仮定）のための、推奨構成を作成します。複数のデータベースがシステムでアクティブな場合は、`mem_percent` パラメーターを調整して、現行データベースのメモリーの配分を反映します。たとえば、DB2 がシステム・メモリーの 80% を使用し、リソースを平等に共有する 2 つのアクティブ・データベースがシステムにある場合、40% (80% を 2 データベースで割る) をパラメーター `mem_percent` に指定します。

EXPORT

データベースから、いくつかある外部ファイル形式のいずれかにデータをエクスポートします。ユーザーは、SQL SELECT ステートメントを提供するか、タイプ表の階層情報を提供して、エクスポートするデータを指定します。

許可:

以下のいずれかが必要です。

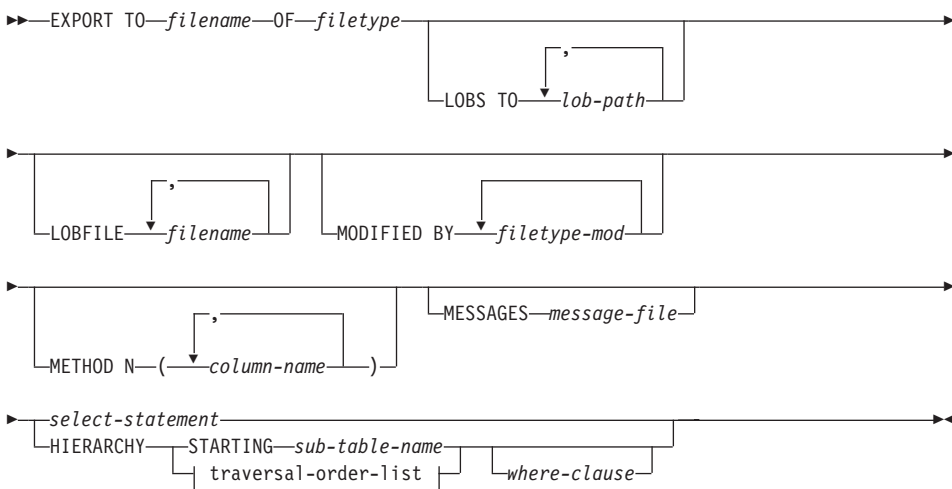
- `sysadm`
- `dbadm`

または、関係する各表またはビューに対する CONTROL または SELECT 特権

必要な接続:

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

コマンド構文:



traversal-order-list:**コマンド・パラメーター:****HIERARCHY traversal-order-list**

指定した走査順序を使用して副階層をエクスポートします。すべての副表は、PRE-ORDER 方式でリストされていなければなりません。最初の副表名が、SELECT ステートメントのターゲット表名として使用されます。

HIERARCHY STARTING sub-table-name

デフォルトの走査順序 (ASC、DEL、または WSF ファイルの OUTER 順序、または PC/IXF データ・ファイルに保管されている順序) を使用して、*sub-table-name* から始まる副階層をエクスポートします。

LOBFILE filename

LOB ファイルに 1 つ以上の基本ファイル名を指定します。最初の名前の名前スペースがいっぱいになると、2 番目の名前が使用され、以下 3 番目、4 番目と続きます。

エクスポート操作中に LOB ファイルを作成するときに、まずこのリストから現行パス (*lob-path* で指定されたパス) に現行の基本ファイル名を追加し、それに 3 桁の順序番号を追加したファイル名が構成されます。たとえば、現行 LOB パスがディレクトリー /u/foo/lob/path/ で、現行 LOB ファイル名が bar の場合、LOB ファイルは、
/u/foo/lob/path/bar.001、/u/foo/lob/path/bar.002 (以下 003、004 と続く) などのように作成されます。

LOBS TO lob-path

LOB ファイルが保管される、ディレクトリーへの 1 つ以上のパスを指定します。LOB パスごとに少なくとも 1 つのファイルが存在し、各ファイルには少なくとも 1 つの LOB が含まれます。

MESSAGES message-file

エクスポート操作中に生じ得る警告およびエラー・メッセージの宛先を指定します。宛先ファイルがすでに存在している場合、エクスポート・ユーティリティーは情報を追加します。 *message-file* を省略すると、メッセージは標準出力に書き込まれます。

METHOD N column-name

出力ファイルで使用される 1 つ以上の列名を指定します。このパラメーターが指定されない場合、表の列名が使用されます。このパラメーターは WSF および IXF ファイルでのみ有効ですが、階層データをエクスポートするときは無効です。

MODIFIED BY filetype-mod

追加オプションを指定します (1139 ページの表 233 を参照)。

OF filetype

以下のような出力ファイルのデータ形式を指定します。

- DEL (区切り文字付き ASCII 形式)。さまざまなデータベース・マネージャーやファイル・マネージャーで使用します。
- WSF (ワークシート形式)。以下のプログラムで使用します。
 - ロータス 1-2-3
 - Lotus Symphony

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

- SELECT ステートメントで列が指定してある場合を除き、ほとんどの表属性である IXF (統合交換フォーマット、PC バージョン) と、既存の索引が IXF ファイルに保管されます。このフォーマットを使うと、表は再作成されます。一方、他のファイル形式を使用する場合、データをそこにインポートするには表が存在していなければなりません。

select-statement

エクスポートされるデータを戻す SELECT ステートメントを指定します。

SELECT ステートメントによってエラーが発生する場合、メッセージ・ファイル (または標準出力) にメッセージが書き込まれます。エラー・コードが SQL0012W、SQL0347W、SQL0360W、SQL0437W、または SQL1824W である場合、エクスポート操作は続行します。これ以外のエラー・コードの場合、操作は停止します。

TO filename

データのエクスポート先のファイルの名前を指定します。このファイルへの完全パスが指定されていない場合、エクスポート・ユーティリティーは現行のディレクトリーおよびデフォルトのドライブを宛先として使用します。

すでに存在するファイルの名前を指定した場合、エクスポート・ユーティリティーはファイルの内容を上書きします。情報の追加は行いません。

例:

次に示すのは、SAMPLE データベースにある STAFF 表から、ファイル myfile.ixf に情報をエクスポートする方法の一例です。これは、IXF 形式で出力されます。コマンドを発行する前に、SAMPLE データベースと接続していなければなりません。データベース接続が DB2 Connect を介して確立されていない場合、索引定義 (もしあれば) は出力ファイルに保管されます。

```
db2 export to myfile.ixf of ixf messages msgs.txt select * from staff
```

次に示すのは、SAMPLE データベースの STAFF 表から部門 20 の従業員に関する情報を、エクスポートする方法の一例です。これは IXF 形式で出力され、awards.ixf ファイルに入ります。コマンドを発行する前に、まず SAMPLE データベースと接続する必要があります。また、表の中の実際の列名は、'department' ではなく 'dept' であることにも注意してください。

```
db2 export to awards.ixf of ixf messages msgs.txt select * from staff
where dept = 20
```

以下の例は LOB を DEL ファイルにエクスポートする方法を示しています。

```
db2 export to myfile.del of del lobs to mylobs/
lobfile lobs1, lobs2 modified by lobsinfile
select * from emp_photo
```

以下の例は LOB を DEL ファイルにエクスポートする方法を示しています。ここでは、最初のディレクトリーにファイルを入れることができない場合のために 2 番目のディレクトリーを指定しています。

```
db2 export to myfile.del of del
lobs to /db2exp1/, /db2exp2/ modified by lobsinfile
select * from emp_photo
```

以下の例はデータを DEL ファイルにエクスポートする方法を示しています。ここでは、単一引用符をストリング区切り文字として使用し、セミコロンを列の区切り文字として使用し、コンマを小数点として使用します。データを再びデータベースにインポートする場合、これと同じ規則を使用する必要があります。

```
db2 export to myfile.del of del
modified by charde1'' coldel; decpt,
select * from staff
```

使用上の注意:

エクスポート操作を開始する前に、すべての表操作が完了し、すべてのロックがペンディング解除になっていることを確認してください。これは、WITH HOLD でオープンされた、すべてのカーソルをクローズした後で COMMIT または ROLLBACK を発行することによって行われます。

SELECT ステートメントでは表の別名を使用できます。

メッセージ・ファイルに置かれるメッセージには、メッセージ検索サービスから戻される情報が含まれています。各メッセージは改行してから始まります。

DEL 形式ファイルへエクスポートするために 254 よりも長い文字データの列が選択されると、エクスポート・ユーティリティーは警告メッセージを生成します。

EXPORT

PC/IXF インポートはデータベース間でデータを移動する場合に使用します。行区切り文字を含む文字データを区切り文字付き ASCII (DEL) ファイルにエクスポートし、テキスト転送プログラムにより処理を行うと、行区切り文字を含むフィールドは長さが伸縮します。

ソースとターゲットのデータベースが両方とも同じクライアントからアクセス可能である場合、ファイルのコピーというステップは必要ありません。

DB2 Connect は、DB2 for OS/390、DB2 for VM and VSE、および DB2 for OS/400 などの DRDA サーバーから表をエクスポートするために使用できます。PC/IXF エクスポートだけがサポートされています。

エクスポート・ユーティリティーは、AIX システムから呼び出される場合、複数部分からなる PC/IXF ファイルを作成しません。

エクスポート・ユーティリティーは、提供される SELECT ステートメントが、SELECT * FROM tablename という形式である場合、IXF ファイルの表の NOT NULL WITH DEFAULT 属性を保管します。

タイプ表をエクスポートする場合、副選択ステートメントは、ターゲット表名と WHERE 文節を指定することによってのみ表現することができます。階層をエクスポートするとき、全選択と選択ステートメント は指定できません。

IXF 以外のファイル形式の場合は、走査順序リストを指定するようお勧めします。このリストは、階層を走査する方法やエクスポートする副表を DB2 に指示します。このリストを指定しない場合、階層内のすべての表がエクスポートされ、デフォルトの順序は OUTER 順序になります。または、OUTER 関数によって指定される順序である、デフォルトの順序を使用することができます。

注: インポート操作の間も同じ走査順序を使用します。ロード・ユーティリティーは、階層や副階層のロードをサポートしていません。

DB2 Data Links Managerについての考慮事項

表の整合性のとれたコピーと、DATALINK 列が参照する対応ファイルが、エクスポートでコピーされるよう保証するには、以下のようにします。

1. コマンド QUIESCE TABLESPACES FOR TABLE tablename SHARE を発行します。
これにより、EXPORT の実行時に進行中の更新トランザクションがないことが保証されます。
2. EXPORT コマンドを実行します。

3. 各 Data Links サーバーで **dlfm_export** ユーティリティを実行します。
dlfm_export への入力、は、エクスポート・ユーティリティで生成される制御ファイル名です。これにより、制御ファイル内にリストされるファイルの tar (または同等の) アーカイブが作成されます。
4. コマンド QUIESCE TABLESPACES FOR TABLE tablename RESET を発行します。
 これにより、表は更新に使用できるようになります。

EXPORT は SQL アプリケーションとして実行されます。SELECT ステートメントを満たす行と列がデータベースから抽出されます。DATALINK 列の場合、SELECT ステートメントはスカラー関数を指定できません。

EXPORT が正常に実行されると、以下のファイルが生成されます。

- EXPORT コマンドで指定したエクスポート・データ・ファイル。このファイルの DATALINK 列値は、インポートおよびロード・ユーティリティによって使用される形式と同じです。DATALINK 列の値が SQL NULL 値である場合、他のデータ・タイプと同じ処理が行われます。
- 各 Data Links サーバー用に生成される制御ファイル *server_name*。Windows オペレーティング・システムでは、単一制御ファイル、*ctrlfile.lst* がすべての Data Links サーバーによって使用されます。これらの制御ファイルは、ディレクトリー <data-file path>%dlfm %YYYYMMDD%HHMMSS に入れられます (Windows NT オペレーティング・システムの場合、*ctrlfile.lst* はディレクトリー <data-file path>%dlfm%YYYYMMDD%HHMMSS に入れられます)。YYYYMMDD は日付 (年月日) を、HHMMSS は時刻 (時、分、秒) を表します。

ファイルを Data Links サーバーからエクスポートするため、**dlfm_export** ユーティリティが提供されています。このユーティリティは、ターゲット Data Links サーバーにファイルをリストアするのに使用できる、アーカイブ・ファイルを生成します。

表 233. 有効なファイル・タイプ修飾子 (エクスポート)

修飾子	説明
	すべてのファイル形式

表 233. 有効なファイル・タイプ修飾子 (エクスポート) (続き)

修飾子	説明
lobsinfile	<p><i>lob-path</i> には、LOB データを含むファイルへのパスを指定します。</p> <p>各パスには、データ・ファイル内で LOB ロケーション指定子 (LLS) によって示される 1 つ以上の LOB を含む、少なくとも 1 つのファイルが含まれます。LLS は、LOB ファイル・パスに保管されるファイル内の LOB のロケーションのストリング表記です。LLS の形式は、<i>filename.ext.nnn.mmm/</i> です。<i>filename.ext</i> は LOB を含むファイルの名前、<i>nnn</i> はファイル内の LOB のオフセット (バイト単位)、<i>mmm</i> は LOB の長さ (バイト単位) を表します。たとえば、ストリング <i>db2exp.001.123.456/</i> がデータ・ファイルに保存される場合、LOB はファイル <i>db2exp.001</i> のオフセット 123 に位置し、456 バイト長です。</p> <p>EXPORT の使用時に "lobsinfile" 修飾子を指定した場合、LOB データは LOBS TO 文節に指定されたロケーションに置かれます。指定しない場合は、LOB データはデータ・ファイル内のインラインに置かれたままになります。LOBS TO 文節は、LOB ファイルが保管されるディレクトリーに、1 つまたは複数のパスを指定します。LOB パスごとに少なくとも 1 つのファイルが存在し、各ファイルには少なくとも 1 つの LOB が含まれます。</p> <p>NULL LOB を指定するには、サイズに -1 と入力します。サイズを 0 と指定すると、長さが 0 の LOB として扱われます。長さが -1 の NULL LOB の場合、オフセットとファイル名は無視されます。たとえば、NULL LOB の LLS は <i>db2exp.001.7.-1/</i> です。</p>
DEL (区切り付き ASCII) ファイル形式	
chardelx	<p><i>x</i> は単一文字のストリング区切り文字です。デフォルト値は二重引用符 (") です。指定した文字は二重引用符の代わりに、文字ストリングを囲むために使用されます。^a</p> <p style="text-align: center;">modified by chardel""</p> <p>単一引用符 (') も、以下のように文字ストリングの区切り文字として指定できます。</p> <p style="text-align: center;">modified by chardel''</p>

表 233. 有効なファイル・タイプ修飾子 (エクスポート) (続き)

修飾子	説明
codepage=x	<p>x は ASCII 文字ストリングです。その値は、出力データ・セット中のデータのコード・ページとして解釈されます。エクスポート操作中に、文字データをアプリケーション・コード・ページからこのコード・ページに変換します。</p> <p>純 DBCS (グラフィック)、混合 DBCS、および EUC では、区切り文字は $x00 \sim x3F$ の範囲に制限されます。 注: CODEPAGE 修飾子は、LOBSINFILE 修飾子と共に使用することはできません。</p>
coldelx	<p>x は単一文字の列区切り文字です。デフォルト値はコンマ (,) です。指定した文字はコンマの代わりに、列の終わりを表すために使用されます。^a</p> <p>以下の例では、coldel; によってエクスポート・ユーティリティーは、すべてのセミコロン (;) を列区切り文字として解釈します。</p> <pre>db2 "export to temp of del modified by coldel; select * from staff where dept = 20"</pre>
datesiso	<p>日付形式。これによって、すべての日付データ値は ISO 形式 ("YYYY-MM-DD") でエクスポートされます。^b</p>
decplusblank	<p>正符号文字。正の 10 進値の先頭に正符号 (+) ではなく、ブランク・スペースが置かれます。デフォルトのアクションでは、10 進値に正符号の接頭部が付けられます。</p>
decptx	<p>x は小数点としてピリオドと置換される単一文字です。デフォルト値はピリオド (.) です。指定した文字はピリオドの代わりに、小数点文字として使用されます。^a</p>
dlldelx	<p>x は単一文字の DATALINK 区切り文字です。デフォルト値はセミコロン (;) です。指定した文字はセミコロン (,) の代わりに、DATALINK 値のフィールド間区切り文字として使用されます。DATALINK 値には副値が複数個含まれる場合があるため、これが必要になります。^a</p> <p>注: x は、行、列、または文字ストリング区切り文字とは異なる文字にしてください。</p>
nodoubledel	<p>二重になっている区切り文字の認識を抑制します。詳細については、1142 ページの『区切り文字の制限』を参照してください。</p>
WSF ファイル形式	

表 233. 有効なファイル・タイプ修飾子 (エクスポート) (続き)

修飾子	説明
1	ロータス 1-2-3 リリース 1、または ロータス 1-2-3 リリース 1a と互換性のある WSF ファイルを作成します。 ^c これがデフォルトです。
2	Lotus Symphony リリース 1.0 と互換性のある WSF ファイルを作成します。 ^c
3	ロータス 1-2-3 バージョン 2、または Lotus Symphony リリース 1.1 と互換性のある WSF ファイルを作成します。 ^c
4	DBCS 文字を含む WSF ファイルを作成します。
<p>注:</p> <p>1. MODIFIED BY オプションでサポートされていないファイル・タイプが指定されても、エクスポート・ユーティリティーが警告を発することはありません。サポートされていないファイル・タイプを使おうとすると、エクスポート操作は失敗し、エラー・コードが戻されます。</p> <p>2. ^a 『区切り文字の制限』に、区切り文字のオーバーライドとして使用できる文字に適用される制限のリストが示されています。</p> <p>3. ^b エクスポート・ユーティリティーは、通常は以下を書き出します。</p> <ul style="list-style-type: none"> • YYYYMMDD 形式の日付データ • "YYYY-MM-DD" 形式の char(date) データ • "HH.MM.SS" 形式の時間データ • "YYYY-MM-DD-HH.MM.SS.uuuuuu" 形式のタイム・スタンプ・データ <p>SELECT ステートメントでエクスポート操作のために指定される日時列に含まれるデータも、これらの形式になります。</p> <p>4. ^c <i>filetype-mod</i> パラメーター・ストリングの中で、ロータス 1-2-3 の場合は L、Symphony の場合は S を指定することにより、これらのファイルを特定の製品を指すようにできます。1 つの値または製品指定文字だけを指定できます。</p>	

区切り文字の制限

選択した区切り文字が移動されるデータの一部になっていないことを確認するのは、ユーザーの責任において行ってください。区切り文字がデータの一部になっている場合、予期しないエラーが発生する場合があります。データを移動する際は、以下の制限が列、ストリング、DATALINK、および小数点区切り文字に適用されます。

- 区切り文字は相互に排他的です。
- 2 進ゼロ、改行文字、改行、またはブランク・スペースを区切り文字にすることはできません。
- デフォルトの小数点 (.) は、ストリング区切り文字にすることはできません。

- 以下の文字は、ASCII ファミリー・コード・ページと EBCDIC ファミリー・コード・ページで、仕様が異なります。
 - シフトイン (0x0F) およびシフトアウト (0x0E) 文字は、EBCDIC MBCS データ・ファイルの場合に、区切り文字にすることができません。
 - MBCS、EUC、または DBCS コード・ページの区切り文字は、0x40 より大きくすることはできません。ただし、EBCDIC MBCS データのデフォルトの小数点 0x4b は例外です。
 - ASCII コード・ページまたは EBCDIC MBCS コード・ページにおけるデータ・ファイルのデフォルト区切り文字は、以下のとおりです。
 - " (0x22、二重引用符; スtring区切り文字)
 - , (0x2c, comma; column delimiter)
 - EBCDIC SBCS コード・ページにおけるデータ・ファイルのデフォルト区切り文字は、以下のとおりです。
 - " (0x7F、二重引用符; スtring区切り文字)
 - , (0x6B, comma; column delimiter)
 - ASCII データ・ファイルのデフォルトの小数点は 0x2e (ピリオド) です。
 - EBCDIC データ・ファイルのデフォルトの小数点は 0x4B (ピリオド) です。
 - サーバーのコード・ページがクライアントのコード・ページと異なっている場合は、非デフォルトの区切り文字を 16 進表示で指定するようお勧めします。たとえば、以下のように指定します。

```
db2 load from ... modified by charde10x0C colde1X1e ...
```

DEL ファイルでの二重文字区切り文字の認識サポートに関する以下の情報は、エクスポート、インポート、およびロード・ユーティリティーに適用されます。

- 文字区切り文字を、DEL ファイルの文字ベースのフィールド内で使用することができます。これは、タイプが CHAR、VARCHAR、LONG VARCHAR、または CLOB (lobsinfile が指定されている場合を除く) のフィールドに適用されます。文字区切り文字で囲まれている文字区切り文字の対は、データベースにインポートまたはロードされます。たとえば、以下のように指定します。

```
"What a "nice" day!"
```

これは、以下のようにインポートされます。

```
What a "nice" day!
```

エクスポートの場合は、逆の規則が適用されます。たとえば、以下のように指定します。

```
I am 6" tall.
```

これは、以下のように DEL ファイルにエクスポートされます。

```
"I am 6" tall."
```

EXPORT

- DBCS 環境では、パイプ (|) 桁区切り文字はサポートされません。

関連概念:

- データ移動ユーティリティー 手引きおよび解説書の『エクスポートの概要』

INSPECT

データベースのページの整合性がとれているかどうかを調べることにより、データベースの構造上の健全性を検査します。この検査では、表オブジェクトの構造および表スペースの構造が有効かどうか調べられます。

有効範囲:

単一パーティション・システムでは、有効範囲はその単一パーティションに限定されません。パーティション・データベース・システムでは、db2nodes.cfg に定義されている論理パーティションすべてのコレクションです。

許可:

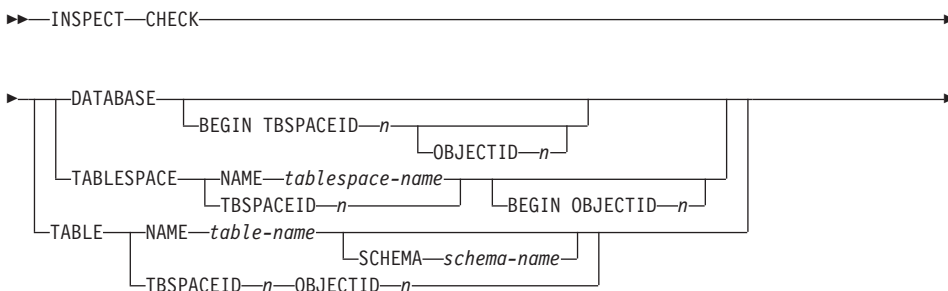
INSPECT CHECK の場合、以下のいずれかになります。

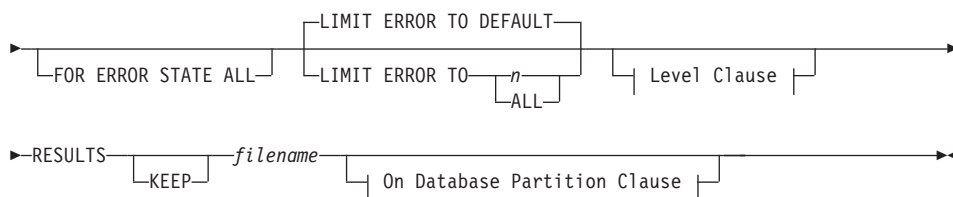
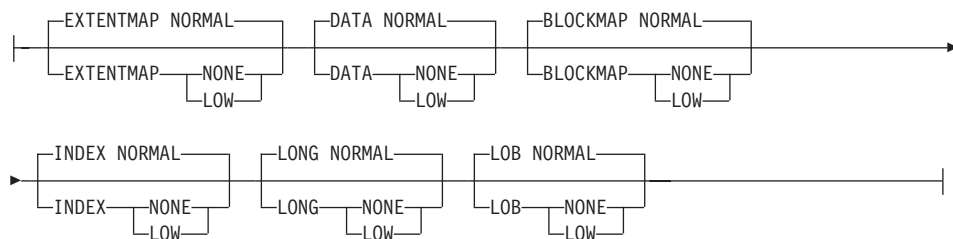
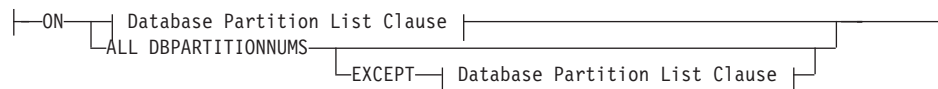
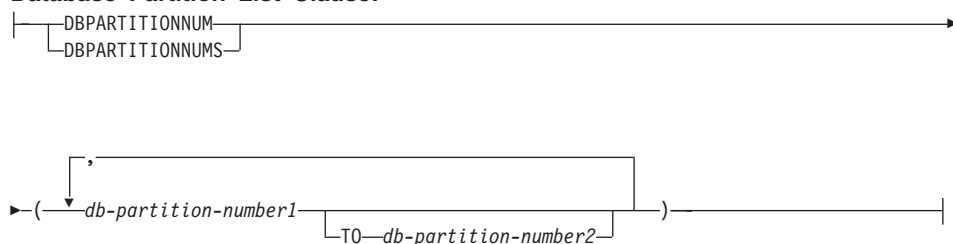
- *sysadm*
- *dbadm*
- *sysctrl*
- *sysmaint*
- 単一表の場合 CONTROL 特権。

必要な接続:

データベース

コマンド構文:



**Level Clause:****On Database Partition Clause:****Database Partition List Clause:****コマンド・パラメーター:****CHECK**

チェック処理を指定します。

DATABASE

全データベースを指定します。

BEGIN TBSPACEID *n*

指定された表スペース ID 番号を持つ表スペースから開始する処理を指定します。

BEGIN TBSpaceID n OBJECTID n

指定された表スペース ID 番号およびオブジェクト ID 番号を持つ表から開始する処理を指定します。

TABLESPACE**NAME table-space-name**

指定された表スペース名を持つ単一の表スペースを指定します。

TBSpaceID n

指定した表スペース ID 番号を持つ単一表スペースを指定します。

BEGIN OBJECTID n

指定されたオブジェクト ID 番号から開始する処理を指定します。

TABLE**NAME table-name**

指定された表名を持つ表を指定します。

SCHEMA schema-name

単一の表操作に対して指定された表名のスキーマ名を指定します。

TBSpaceID n OBJECTID n

指定された表スペース ID 番号およびオブジェクト ID 番号を持つ表を指定します。

FOR ERROR STATE ALL

その内部状態がすでにエラー状態を示している表オブジェクトに関してチェックを実行する場合、その状況だけを報告し、オブジェクトのスキャンは行いません。このオプションを指定すると、内部状態がすでにエラー状態を示している場合でも、オブジェクトのスキャンを行います。

LIMIT ERROR TO n

オブジェクトのエラー・ページ数を *n* まで報告できます。オブジェクトのエラー・ページ数がこの限界に達すると、残りのオブジェクトのチェック処理は中止されます。

LIMIT ERROR TO DEFAULT

オブジェクトのデフォルト・エラー・ページ数まで報告できます。この値は、オブジェクトのエクステント・サイズです。このパラメーターがデフォルトです。

LIMIT ERROR TO ALL

報告されるエラー・ページ数の限界はありません。

EXTENTMAP**NORMAL**

エクステント・マップの処理レベルを通常に指定します。デフォルト。

NONE エクステント・マップの処理レベルをなしに指定します。

LOW エクステント・マップの処理レベルを低に指定します。

DATA

NORMAL

データ・オブジェクトの処理レベルを通常に指定します。デフォルト。

NONE データ・オブジェクトの処理レベルをなしに指定します。

LOW データ・オブジェクトの処理レベルを低に指定します。

BLOCKMAP

NORMAL

ブロック・マップ・オブジェクトの処理レベルを通常に指定します。デフォルト。

NONE ブロック・マップ・オブジェクトの処理レベルをなしに指定します。

LOW ブロック・マップ・オブジェクトの処理レベルを低に指定します。

INDEX

NORMAL

索引オブジェクトの処理レベルを通常に指定します。デフォルト。

NONE 索引オブジェクトの処理レベルをなしに指定します。

LOW 索引オブジェクトの処理レベルを低に指定します。

LONG

NORMAL

長オブジェクトの処理レベルを通常に指定します。デフォルト。

NONE 長オブジェクトの処理レベルをなしに指定します。

LOW 長オブジェクトの処理レベルを低に指定します。

LOB

NORMAL

LOB オブジェクトの処理レベルを通常に指定します。デフォルト。

NONE LOB オブジェクトの処理レベルをなしに指定します。

LOW LOB オブジェクトの処理レベルを低に指定します。

RESULTS

結果出力ファイルを指定します。ファイルは診断データ・ディレクトリー・パスに書き込まれます。チェック処理によってエラーが検出されない場合、この

結果出力ファイルは INSPECT 操作の終了時に消去されます。チェック処理によってエラーが検出される場合、この結果出力ファイルは INSPECT 操作の終了時に消去されません。

KEEP 結果出力ファイルを常に保持することを指定します。

file-name

結果出力ファイルの名前を指定します。

ALL DBPARTITIONNUMS

db2nodes.cfg ファイルに指定されているすべてのデータベース・パーティションで操作が実行されることを指定します。ノード文節が指定されていない場合、これがデフォルトです。

EXCEPT

ノード・リストに指定されているデータベース・パーティションを除き、db2nodes.cfg ファイルに指定されているすべてのデータベース・パーティションで操作が実行されることを指定します。

ON DBPARTITIONNUM / ON DBPARTITIONNUMS

データベース・パーティションのセットに対して操作を実行します。

db-partition-number1

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

db-partition-number2

2 番目のデータベース・パーティション番号を指定し、db-partition-number1 から db-partition-number2 までのすべてのデータベース・パーティションがデータベース・パーティション・リストに含まれるようにします。

使用上の注意:

1. 表オブジェクトでのチェック操作に関して、処理レベルはオブジェクトに対して指定できます。デフォルトは NORMAL レベルです。オブジェクトに NONE を指定すると、そのオブジェクトは除外されます。LOW を指定すると、NORMAL で行われるチェックのサブセットのチェック操作を行います。
2. 表スペースまたは表を識別する ID 値を指定することにより、データベースのチェックを特定の表スペースまたは表から開始するように指定できます。
3. 表を識別する ID 値を指定することにより、表スペースのチェックを特定の表から開始するように指定できます。
4. 表スペースの処理は、表スペース内にあるオブジェクトにのみ影響を与えます。
5. オンライン検査処理では、分離レベルを非コミット読み取りに指定してデータベース・オブジェクトにアクセスします。COMMIT 処理は、INSPECT 処理の際に行われます。INSPECT を呼び出す前に、COMMIT または ROLLBACK を発行して作業単位を終了することをお勧めします。

6. オンライン検査処理により、フォーマットされていない検査データが、指定された結果ファイルに書き出されます。ファイルは診断データ・ディレクトリー・パスに書き込まれます。チェック処理によってエラーが検出されない場合、この結果出力ファイルは INSPECT 操作の終了時に消去されます。チェック処理によってエラーが検出される場合、この結果出力ファイルは INSPECT 操作の終了時に消去されません。チェック処理が完了した後、検査の詳細を表示するには、ユーティリティー `db2inspf` を使って検査結果データをフォーマットする必要があります。結果ファイルには、データベース・パーティション番号の拡張子が付きます。パーティション・データベース環境では、各データベース・パーティションで、そのデータベース・パーティション番号と一致した拡張子を持つ独自の結果出力ファイルが生成されます。結果出力ファイルは、データベース・マネージャーの診断データ・ディレクトリー・パスに出力されます。すでに存在するファイル名を指定すると、操作は処理されません。ですから、ファイル名を指定する前にそのファイルを除去する必要があります。

LOAD

データを DB2 表にロードします。サーバー上に存在するデータは、ファイル、テープ、または Named Pipe の形式にすることができます。リモートに接続されたクライアント上に存在するデータは、完全修飾ファイルまたは Named Pipe の形式にすることができます。ユーザー定義カーソルからデータをロードすることも可能です。ロード・ユーティリティーは、階層レベルでのデータのロードをサポートしていません。

有効範囲:

このコマンドは、1 度の要求で複数のデータベース・パーティションに対して発行できます。

許可:

以下のいずれかが必要です。

- `sysadm`
- `dbadm`
- データベースに対するロード権限と以下のもの
 - ロード・ユーティリティーが INSERT モード、 TERMINATE モード (それまでのロード挿入操作を終了する)、または RESTART モード (以前のロード挿入操作を再開する) で呼び出された場合には、その表に対する INSERT 特権。
 - ロード・ユーティリティーが REPLACE モード、 TERMINATE モード (それまでのロード置換操作を終了する)、または RESTART モード (以前のロード置換操作を再開する) で呼び出された場合には、その表に対する INSERT および DELETE 特権。
 - 例外表がロード操作の一部として使用される場合、その例外表に対する INSERT 特権。

LOAD

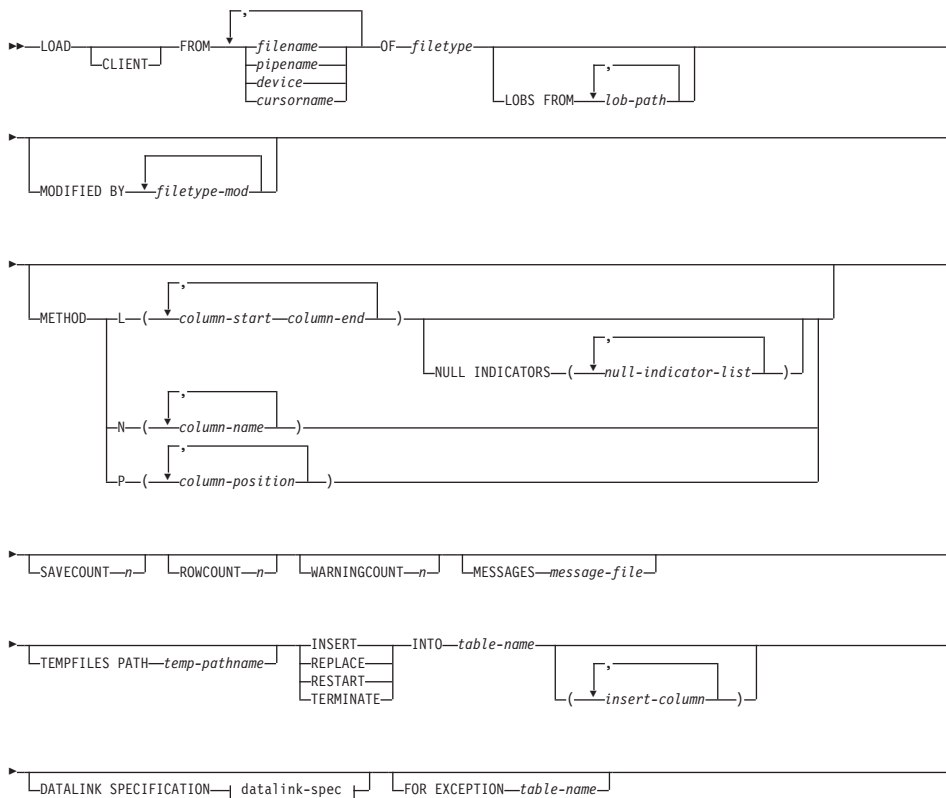
注: すべてのロード・プロセス (および一般にすべての DB2 サーバー・プロセス) はインスタンス所有者によって所有されており、それらのプロセスすべてにおいて、必要なファイルにアクセスするためにそのインスタンス所有者の ID を使うため、インスタンス所有者には入力データ・ファイルに対する読み取りアクセス権が必要です。このコマンドをだれが呼び出すかには関係なく、それらの入力データ・ファイルをインスタンス所有者から読むことができなければなりません。

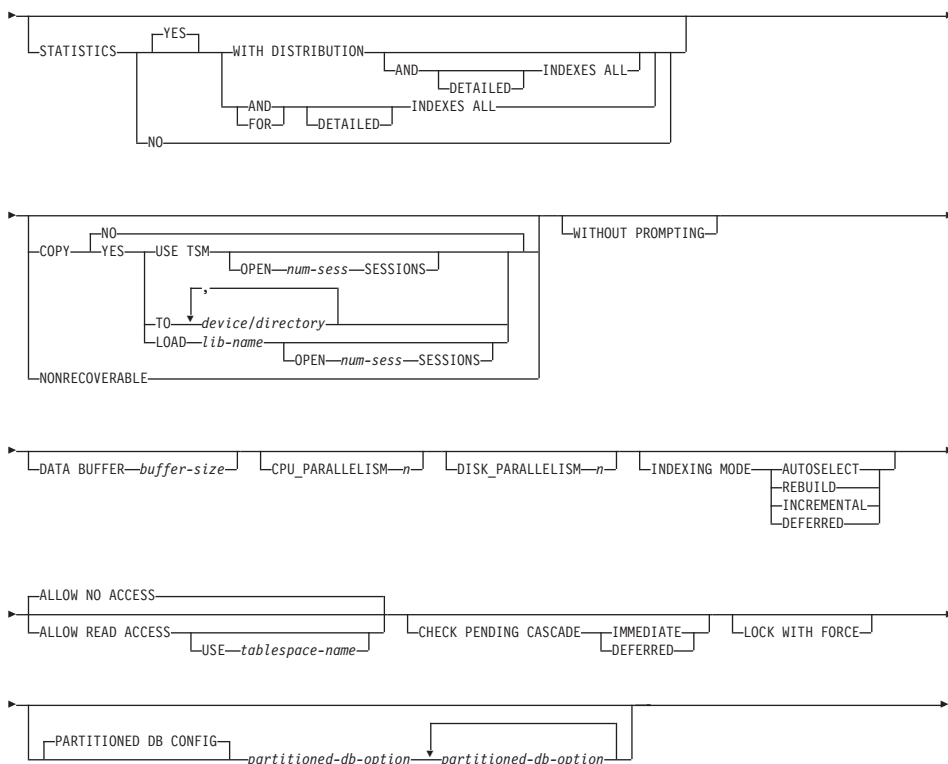
必要な接続:

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

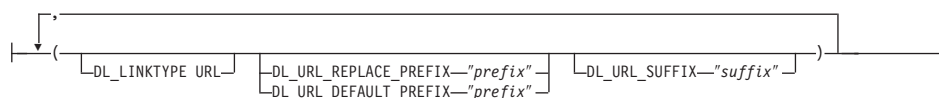
インスタンス。明示的なアタッチは必要ありません。データベースへの接続が確立している場合には、ローカル・インスタンスへの暗黙的なアタッチが試行されます。

コマンド構文:





datalink-spec:



コマンド・パラメーター:

ALLOW NO ACCESS

ロードを使用すると、ロード中に、排他的アクセスのターゲット表がロックされます。ロード中、表の状態は **LOAD IN PROGRESS** に設定されます。**ALLOW NO ACCESS** はデフォルトの動作です。これは、**LOAD REPLACE** で唯一有効なオプションです。

表に制約があると、表の状態は、**LOAD IN PROGRESS** の他に、**CHECK PENDING** に設定されます。**CHECK PENDING** を解除するには、**SET INTEGRITY** コマンドを使用する必要があります。

ALLOW READ ACCESS

ロードを使用すると、ターゲット表は共用モードでロックされます。表の状態は、**LOAD IN PROGRESS** および **READ ACCESS** の両方に設定されます。表

のロード中、データの非デルタ部分にアクセスすることができます。つまり、表を読み取る側はロードの開始前に存在していたデータにはアクセスができ、ロード中のデータはロードが完了するまで利用できない、ということです。

ALLOW READ ACCESS ロードの **LOAD TERMINATE** または **LOAD RESTART** はこのオプションを使用できますが、**ALLOW NO ACCESS** ロードの **LOAD TERMINATE** または **LOAD RESTART** はこのオプションを使用できません。また、ターゲット表上の索引が要再作成のマークが付けられると、このオプションは無効になります。

表に制約があると、表の状態は、**LOAD IN PROGRESS**、**READ ACCESS** の他に、**CHECK PENDING** に設定されます。ロードの終了時に、表の状態 **LOAD IN PROGRESS** は解除されますが、**CHECK PENDING** と **READ ACCESS** はそのまま残ります。**CHECK PENDING** を解除するには、**SET INTEGRITY** コマンドを使用する必要があります。表が **CHECK PENDING** および **READ ACCESS** の状態にある間、データの非デルタ部分は引き続きアクセス可能で、データの新しい (デルタ) 部分は、**SET INTEGRITY** コマンドが完了するまでアクセスできません。ユーザーは、**SET INTEGRITY** コマンドを発行せずに、同じ表上で複数のロードを実行できます。ただし、元の (チェック済み) データは、**SET INTEGRITY** コマンドが発行されるまで可視のままです。

ALLOW READ ACCESS は、以下の修飾子もサポートします。

USE tablespace-name

索引が再作成される場合、表スペース *tablespace-name* に索引のシャドー・コピーが作成され、**INDEX COPY PHASE** のロード終了時に、元の表スペース上にコピーされます。システム **TEMPORARY** 表スペースのみ、このオプションを使用できます。指定されない場合、索引オブジェクトと同じ表スペースにシャドー索引が作成されます。索引オブジェクトと同じ表スペースにシャドー・コピーが作成される場合、シャドー索引オブジェクトは瞬間的に古い索引オブジェクトの上にコピーされます。シャドー・コピーが索引オブジェクトと別の表スペースにある場合、物理コピーが実行されます。これにはかなりの入出力および時間を要します。コピーは、表がオフラインの間、**INDEX COPY PHASE** のロード終了時に行われます。

このオプションをしないと、シャドー索引は元の索引と同じ表スペースに作成されます。デフォルトでは、元の索引とシャドー索引の両方が同時に同じ表スペースに常駐するため、1つの表スペース内に両方の索引を保留するためのスペースが不足する場合があります。このオプションを使用すれば、索引用の十分な表スペースを確保できます。

ユーザーが **INDEXING MODE REBUILD** または **INDEXING MODE AUTOSELECT** を指定しない場合、このオプションは無視されます。このオプションは **INDEXING MODE AUTOSELECT** が選択され、ロードが索引を徐々に更新することを選択した場合にも無視されます。

CHECK PENDING CASCADE

LOAD によって表がチェック・ペンディング状態になる場合、CHECK PENDING CASCADE オプションを使用することによってユーザーはロードされる表を即時にすべての下層(下層外部キー表、下層即時マテリアライズ照会表、および下層即時ステージング表を含む)にカスケードするかどうか指定することができます。

IMMEDIATE

外部キー制約のチェック・ペンディング状態(読み取りまたは非アクセス・モード)が即時にすべての下層外部キー表に拡張されることを示します。表に下層即時マテリアライズ照会表または下層即時ステージング表がある場合、チェック・ペンディング状態は即時にマテリアライズ照会表およびステージング表に拡張されます。LOAD INSERT 操作の場合、IMMEDIATE オプションが指定されている場合でも、チェック・ペンディング状態は下層外部キー表に拡張されないことに注意してください。

後で (SET INTEGRITY ステートメントの IMMEDIATE CHECKED オプションを使用して) ロードされる表の制約違反をチェックする際、チェック・ペンディング読み取り状態だった下層外部キー表は、チェック・ペンディング非アクセス状態になります。

DEFERRED

ロードされる表だけがチェック・ペンディング状態(読み取りまたは非アクセス・モード)になることを示します。下層外部キー表、下層即時マテリアライズ照会表、および下層即時ステージング表は、未変更のままになります。

下層外部キー表は、(SET INTEGRITY コマンドの IMMEDIATE CHECKED オプションを使用して) その親表の制約違反がチェックされる時、後で暗黙的にチェック・ペンディング非アクセス状態になる場合があります。下層即時マテリアライズ照会表および下層即時ステージング表は、その基本表のいずれかの健全性違反がチェックされる際、暗黙的にチェック・ペンディング非アクセス状態になります。従属表がチェック・ペンディング状態になったことを示す警告 (SQLSTATE 01586) が出されます。この従属表がいつチェック・ペンディング状態になるかについては、SQL リファレンスにある SET INTEGRITY ステートメントの「注」のセクションを参照してください。

CHECK PENDING CASCADE オプションが指定されない場合、以下のようになります。

- ロードされる表だけが、チェック・ペンディング状態になります。下層外部キー表、下層即時マテリアライズ照会表、および下層即時ステージング表

LOAD

は、未変更のままになり、後にロードされた表の制約違反がチェックされる際に、暗黙的にチェック・ペンディング状態になる場合があります。

LOAD によってターゲット表がチェック・ペンディング状態にならない場合、CHECK PENDING CASCADE オプションは無視されます。

CLIENT

ロードするデータが、リモートに接続するクライアントにあることを指定します。ロード操作がリモート・クライアントから呼び出されない場合、このオプションは無視されます。ファイル・タイプが CURSOR の場合、このオプションはサポートされていません。

注:

1. DUMPFIL および LOBSINFIL 修飾子は、CLIENT キーワードが指定されている場合でも、サーバー上のファイルを参照します。
2. コード・ページ変換は、リモートのロード操作時には実行されません。データのコード・ページがサーバーのコード・ページとは異なる場合、データのコード・ページは CODEPAGE 修飾子を使用して指定する必要があります。

以下の例では、リモートに接続されたクライアント上に存在するデータ・ファイル (/u/user/data.del) は、サーバー・データベース上の MYTABLE にロードされます。

```
db2 load client from /u/user/data.del of del
      modified by codepage=850 insert into mytable
```

COPY NO

順方向リカバリーが使用可能 (つまり、logretain または userexit がオン) になっていれば、表が存在している表スペースをバックアップ・ペンディング状態にするよう指定します。COPY NO を使用する場合も、表スペース状態は LOAD IN PROGRESS になります。これは、一時的な状態であり、ロードが完了するか打ち切られると解除されます。表スペースのバックアップまたはデータベースの完全バックアップを実行しない限り、表スペースのどの表のデータも更新または削除できません。ただし、SELECT ステートメントを使用すれば、どの表のデータにもアクセス可能です。

リカバリー可能データベースでの COPY NO を指定した LOAD は、表スペースをバックアップ・ペンディング状態のままにし、索引のリフレッシュが必要な状態のままにします。たとえば、COPY NO を指定した LOAD および INDEXING MODE DEFERRED を実行すると、索引はリフレッシュが必要な状態になります。表での照会には、索引スキャンが必要なものがあり、索引がリフレッシュされるまで、成功しません。バックアップ・ペンディング状態にある表スペース内に常駐する場合、索引はリフレッシュできません。この場合、表へのアクセスは、バックアップが行われるまで許可されません。

注: 索引リフレッシュは、索引が照会によってアクセスされたときに、自動的に行われます。

COPY YES

ロードするデータのコピーを保存することを指定します。順方向リカバリーが使用禁止 (つまり *logretain* と *userexit* が両方ともオフ) であれば、このオプションは無効です。このオプションは DATALINK 列を含む表ではサポートされません。

USE TSM

Tivoli Storage Manager (TSM) を使ってコピーを保管することを指定します。

OPEN num-sess SESSIONS

TSM またはベンダー製品とともに使用する入出力セッションの数です。デフォルト値は 1 です。

TO device/directory

コピー・イメージを作成する先の装置またはディレクトリーを指定します。

LOAD lib-name

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

CPU_PARALLELISM n

表オブジェクトの作成時に、レコードの解析、変換、および形式設定のためにロード・ユーティリティーが spawn するプロセスまたはスレッドの数を指定します。このパラメーターは、パーティション内並列処理を活用するために設計されています。これは、事前にソートされたデータをロードする際に役立ちます (ソース・データのレコード順序が保持されるため)。このパラメーターの値が 0 の場合や、このパラメーターを指定しなかった場合、ロード・ユーティリティーは、ランタイムにインテリジェントなデフォルト (通常は使用可能な CPU の数に基づく) を使用します。

注:

1. LOB または LONG VARCHAR フィールドのどちらかを含む表でこのパラメーターを使用する場合、システムの CPU の数またはユーザーが指定した値には関係なく、値は 1 になります。
2. SAVECOUNT パラメーターに指定する値が小さいと、データと表のメタデータの両方をフラッシュするために、ローダーがさらに多くの入出力操作を実行することになります。CPU_PARALLELISM が 1 より大きいなら、フラッシュ操作は非同期的になり、ローダーは CPU を活用できます。CPU_PARALLELISM が 1 に設定されているなら、ローダーは整合点にお

LOAD

いて IO を待ちます。CPU_PARALLELISM を 2、SAVECOUNT を 10 000 に設定したロード操作は、CPU が 1 つしかなくても、同じ操作で CPU_PARALLELISM を 1 に設定した場合より速く完了します。

DATA BUFFER buffer-size

ユーティリティー内でデータを転送するためのバッファー・スペースとして使用する 4KB ページ数を設定します (並列処理の度合いには依存しません)。指定する値がアルゴリズム上の最小値より小さい場合、最小限必要なリソースが使用され、警告は戻されません。

このメモリーは、ユーティリティー・ヒープから直接に割り当てられ、そのサイズは `util_heap_sz` データベース構成パラメーターで修正可能です。

値を指定しないと、実行時にユーティリティーによって適切なデフォルトが計算されます。デフォルトは、表の特性だけでなく、ローダーのインスタンス生成時にユーティリティー・ヒープ中で使用可能なフリー・スペースの割合に基づいています。

DATALINK SPECIFICATION

各 DATALINK 列ごとに、それぞれ 1 つの列指定を括弧で囲んで指定できます。各列指定は、1 つ以上の DL_LINKTYPE、接頭部、および DL_URL_SUFFIX 指定で構成されます。接頭部指定は、DL_URL_REPLACE_PREFIX または DL_URL_DEFAULT_PREFIX のどちらかになります。

DATALINK 列指定の数は、表で定義されている DATALINK 列の数と同じだけ指定できます。指定の順序は、挿入列 リストの中での DATALINK 列の順序、または表定義内での順序 (挿入列 リストが指定されていない場合) に従います。

DISK_PARALLELISM n

表スペース・コンテナにデータを書き込むためにロード・ユーティリティーが生成するプロセスまたはスレッドの数を指定します。値を指定しないと、ユーティリティーは表スペース・コンテナの数と表の特性に基づいて適切なデフォルトを選択します。

DL_LINKTYPE

指定した場合は、列定義の LINKTYPE に一致していなければなりません。そうすることによって、列定義に LINKTYPE URL が指定されている場合に DL_LINKTYPE URL が受け入れ可能になります。

DL_URL_DEFAULT_PREFIX "prefix"

これを指定すると、同じ列内のすべての DATALINK 値のデフォルト接頭部になります。ここでいう接頭部とは、URL 指定の「スキーム・ホスト・ポート」部分のことです。(分散ファイル・システム (DFS) の場合、接頭部とは URL 指定の「スキーム・セル名とファイル・スペースの接合」部分のことです。)

接頭部の例


```
"http://server"
"file://server"
"file:"
"http://server:80"
"dfs://.../cellname/fs"
```

列データの中に接頭部がない場合、DL_URL_DEFAULT_PREFIX でデフォルトの接頭部が指定されているなら、列の値の接頭部としてそのデフォルト接頭部が付けられます (NULL でない場合)。

たとえば、DL_URL_DEFAULT_PREFIX でデフォルト接頭部が "http://toronto" として指定されている場合、

- 列入力値 "/x/y/z" は "http://toronto/x/y/z" として保管されます。
- 列入力値 "http://coyote/a/b/c" は "http://coyote/a/b/c" として保管されます。
- 列入力値 NULL は NULL として保管されます。

DL_URL_REPLACE_PREFIX "prefix"

この文節は、それ以前にエクスポート・ユーティリティーによって生成されたデータをロードまたはインポートする際に、ユーザーがデータに含まれるホスト名を別のホスト名に一括置換したい場合に便利です。指定する場合には、それがすべての非 NULL 列値の接頭部になります。列値にすでに接頭部があるなら、それは置き換えられます。列値に接頭部がないなら、DL_URL_REPLACE_PREFIX で指定される接頭部がその列値の接頭部になります。分散ファイル・システム (DFS) の場合、接頭部とは URL 指定の「スキーム・セル名とファイル・スペースの接合」部分のことです。

たとえば、DL_URL_REPLACE_PREFIX で接頭部が "http://toronto" として指定されている場合、

- 列入力値 "/x/y/z" は "http://toronto/x/y/z" として保管されます。
- 列入力値 "http://coyote/a/b/c" は "http://toronto/a/b/c" として保管されます。
"coyote" は "toronto" に置き換えられます。
- 列入力値 NULL は NULL として保管されます。

DL_URL_SUFFIX "suffix"

これを指定すると、それはその列のすべての非 NULL 列値に付加されます。これは実際には、DATALINK 値のデータ・ロケーション部分の『パス』コンポーネントに付加されます。

FOR EXCEPTION table-name

エラーが発生した行のコピー先となる例外表を指定します。ユニーク索引または主キー索引に違反した行がすべてコピーされます。DATALINK 例外も例外表に取り込まれます。未修飾の表名を指定すると、その表は CURRENT SCHEMA で修飾されます。

例外表に書き込まれる情報は、ダンプ・ファイルには書き込まれません。パーティション・データベース環境では、例外表は、ロードする表が定義されたノ

LOAD

ードについて定義する必要があります。一方ダンプ・ファイルは、無効であるか構文エラーであるためにロードできない行が含まれます。

FROM filename/pipename/device/cursorname

ロードするデータが含まれている SQL ステートメントを参照するファイル、パイプ、装置、またはカーソルを指定します。入力ソースがファイル、パイプ、または装置の場合、CLIENT オプションが指定されていなければ、データベースが存在するデータベース・パーティションになければなりません。複数の名前を指定すると、それらは順番に処理されます。最後に指定した項目がテープ装置の場合は、別のテープを使うようユーザーに対してプロンプトが出ます。有効な応答オプションは、以下のとおりです。

- c** 続行。警告メッセージを生成した装置を使用し続けます (たとえば、新しいテープがマウントされた場合)。
- d** 装置の終了。警告メッセージを生成した装置の使用を停止します (たとえば、それ以上テープがない場合)。
- t** 終了。すべての装置を終了します。

注:

1. 可能なかぎり完全修飾ファイル名を使用してください。リモート・サーバーの場合は、常に完全修飾ファイル名を使う必要があります。呼び出し側と同じデータベース・パーティションにデータベースが存在する場合には、相対パスを使用することもできます。
2. ファイルが物理的には分割されているが論理的には 1 つのファイルである場合には、複数の IXF ファイルからのデータのロードがサポートされています。ファイルが論理的にも物理的にも分割されている場合は、サポートされていません。(複数の物理ファイルがすべて 1 度の EXPORT コマンドの呼び出しで作成された場合、それらは論理的には 1 つであると見なされます。)
3. クライアント・マシン上に存在するデータをロードする場合、そのデータは、完全修飾ファイルまたは Named Pipe のいずれかの形式でなければなりません。

INDEXING MODE

ロード・ユーティリティーが索引を再作成するのか、それとも索引を増分で拡張するのかを指定します。有効な値は以下のとおりです。

AUTOSELECT

REBUILD モードと INCREMENTAL モードのどちらにするかを、ロード・ユーティリティーが自動的に決定します。

REBUILD

すべての索引が再作成されます。古い表データの索引キー部分も、追

加される新しい表データの索引キー部分もすべてソートできるようにするため、ロード・ユーティリティーには十分なリソースが必要となります。

INCREMENTAL

索引に新しいデータが取り込まれて拡張します。このアプローチでは、索引のフリー・スペースが消費されます。このアプローチでは、新たに挿入されるレコードの索引キーを追加するためのソート・スペースだけがあれば十分です。この方式がサポートされるのは、索引オブジェクトが有効で、かつロード操作の開始時にアクセス可能な場合だけです (たとえば、DEFERRED モードが指定されたロード操作の直後では、この方式は無効です)。このモードを指定したものの、索引の状態などの理由でサポートされない場合は、警告が戻され、REBUILD モードでロード操作が続行されます。同様に、ロード作成フェーズでロード再開操作を開始した場合も、INCREMENTAL モードはサポートされません。

以下の条件がすべて真の場合、増分索引の作成はサポートされません。

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

この制限をう回するため、索引は別々の表スペースに置くようお勧めします。

DEFERRED

このモードが指定されている場合、ロード・ユーティリティーは索引の作成を試みません。最新表示が必要であることを示すマークが索引に付けられます。ロード操作とは関係のないこのような索引に最初にアクセスするときは、再作成が強制的に実行されたり、データベースの再始動時に索引が再作成されたりする場合があります。このアプローチでは、最も大きい索引のキー部分をすべて処理できるだけのソート・スペースが必要です。索引を作成するためにその後かかる合計時間は、REBUILD モードの場合よりも長くなります。したがって、この索引作成据え置きモードで複数のロード操作を実行する場合、最初の非ロード・アクセス時に索引を再作成できるようにしておくよりも、順序列内の最後のロード操作で索引の再作成を実行できるようにした方が (パフォーマンスの観点から) 賢明であるといえます。

据え置き索引作成がサポートされるのは、非固有の索引がある表だけです。そのため、ロード・フェーズで挿入される複写キーがロード操作後は永続的ではなくなります。

LOAD

注: 据え置き索引作成は、DATALINK 列がある表ではサポートされません。

INSERT

ロード・ユーティリティを実行できる 4 つのモードのうちの 1 つ。既存の表データを変更することなく、ロードされたデータを表に追加します。

insert-column

データの挿入先となる表列を指定します。

ロード・ユーティリティは、1 つ以上のスペースを含んだ名前の列を解析できません。たとえば、

```
db2 load from delfile1 of del modified by noeofchar noheader
method P (1, 2, 3, 4, 5, 6, 7, 8, 9)
insert into table1 (BLOB1, S2, I3, Int 4, I5, I6, DT7, I8, TM9)
```

は、Int 4 列があるためエラーになります。これは、以下のようにして二重引用符で列名を囲むことによって解決できます。

```
db2 load from delfile1 of del modified by noeofchar noheader
method P (1, 2, 3, 4, 5, 6, 7, 8, 9)
insert into table1 (BLOB1, S2, I3, "Int 4", I5, I6, DT7, I8, TM9)
```

INTO table-name

データのロード先となるデータベース表を指定します。この表として、システム表または宣言一時表は指定できません。別名、完全修飾、または未修飾の表名を指定できます。修飾子付き表名は、`schema.tablename` の形式です。未修飾の表名を指定すると、その表は CURRENT SCHEMA で修飾されます。

LOBS FROM lob-path

ロードする LOB 値が含まれているデータ・ファイルへのパス。パスの最後は斜線 (/) でなければなりません。CLIENT オプションを指定した場合、パスは完全修飾する必要があります。LOB データ・ファイルの名前は、メイン・データ・ファイル (ASC、DEL、または IXF) の、LOB 列にロードされる列内に保管されます。`filetype-mod` スtring内に `lobsinfile` が指定されていない場合、このオプションは無視されます。

ファイル・タイプが CURSOR の場合、このオプションはサポートされていません。

LOCK WITH FORCE

ユーティリティはロード・プロセス中に、表ロックなどの様々なロックを獲得します。ロックを獲得する際、このオプションを使用すると、ロードは待機することなく、またタイムアウトになることなく、競合するロックを持つ他のアプリケーションを強制的にオフにします。強制されたアプリケーションは、ロールバックし、ロード・ユーティリティが必要とするロックをリリースします。その後、ロード・ユーティリティを続行できます。このオプションは、FORCE APPLICATIONS コマンド (SYSADM または SYSCTRL) と同じ権限を必要とします。

ALLOW NO ACCESS は、ロード操作の開始時に競合するロックを持つアプリケーションを強制する場合があります。ロードの開始時に、ユーティリティーは、表の照会または変更のいずれかを試みるアプリケーションを強制する場合があります。

ALLOW READ ACCESS は、ロード操作の開始時および終了時に競合するロックを持つアプリケーションを強制する場合があります。ロードの開始時に、ロード・ユーティリティーは、表の変更を試みるアプリケーションを強制する場合があります。ロードの終了時に、ロード・ユーティリティーは、表の照会または変更のいずれかを試みるアプリケーションを強制する場合があります。

MESSAGES message-file

ロード操作中に生じ得る警告およびエラー・メッセージの宛先を指定します。メッセージ・ファイルを指定しなかった場合、メッセージは標準出力に書き込まれます。このファイルへの完全パスが指定されていない場合、ロード・ユーティリティーは現行のディレクトリーおよびデフォルトのドライブを宛先として使用します。すでに存在するファイル名を指定すると、ロード時に情報が追加されます。

通常、メッセージ・ファイルには、ロード操作の終了時にメッセージが入れますが、それ自体は操作の進行状況のモニターには適していません。

METHOD

L データのロードを開始する列および終了する列の番号を指定します。列の番号は、データの行の先頭からのバイト単位のオフセットです。この番号は 1 から始まります。

注: このメソッドは、ASC ファイルの場合にのみ使用することができ、そのファイル・タイプに対してのみ有効なメソッドです。

N ロードするデータ・ファイルの中の列の名前を指定します。それらの列名の大きい文字小文字は、システム・カタログ中の対応する名前の大きい文字小文字と一致している必要があります。NULL 可能ではない各表列には、METHOD N リスト内に対応する項目が必要です。たとえば、データ・フィールドが F1、F2、F3、F4、F5、および F6 であり、表列が C1 INT、C2 INT NOT NULL、C3 INT NOT NULL、および C4 INT の場合、method N (F2, F1, F4, F3) は有効な要求ですが、method N (F2, F1) は無効です。

注: この方式は、ファイル・タイプ IXF または CURSOR の場合にのみ使用することができます。

P ロードする入力データ・フィールドのフィールド番号 (1 から始まる) を指定します。NULL 可能ではない各表列には、METHOD P リスト内に対応する項目が必要です。たとえば、データ・フィールドが F1、F2、F3、F4、F5、および F6 であり、表列が C1 INT、C2

LOAD

INT NOT NULL、C3 INT NOT NULL、および C4 INT の場合、method P (2, 1, 4, 3) は有効な要求ですが、method P (2, 1) は無効です。

注: この方式は、ファイル・タイプ IXF、DEL、または CURSOR の場合にのみ使用でき、DEL ファイル・タイプに対してのみ有効な方式です。

MODIFIED BY filetype-mod

追加オプションを指定します。 1177 ページの表 234 を参照してください。

NONRECOVERABLE

ロード・トランザクションがリカバリー不能としてマークされており、それ以降のロールフォワード・アクションによってそれをリカバリーさせることは不可能であることを指定します。ロールフォワード・ユーティリティは、そのトランザクションをスキップし、データのロード先の表に "invalid" (無効) としてマークします。さらに、ユーティリティは、その表に対する後続のすべてのトランザクションを無視します。ロールフォワード操作が完了すると、そのような表は、ドロップするか、またはリカバリー不能なロード操作完了後のコミット・ポイントの後に取られたバックアップ (全バックアップまたは表スペースのバックアップ) からリストアすることしかできません。

このオプションを使用すると、表スペースはロード操作後にバックアップ・ペンディング状態になりません。また、ロード操作中にロードされたデータのコピーが作成される必要もなくなります。

FILE LINK CONTROL が指定された DATALINK 列が表に存在している場合や、そのような列を表に追加しようとしている場合には、このオプションを使用しないでください。

NULL INDICATORS null-indicator-list

このオプションは、METHOD L パラメーターを指定した場合だけ使用できます (つまり、入力ファイルが ASC ファイルの場合)。NULL 標識リストは、コンマで区切られた正の整数のリストで、各 NULL 標識フィールドの列の番号を指定します。列の番号は、データの行の先頭からのバイト単位の、各 NULL 標識フィールドのオフセットです。NULL 標識リストには、METHOD L パラメーターで定義された各データ・フィールドに対する 1 つの項目がなければなりません。列の番号がゼロであることは、対応するデータ・フィールドが必ずデータを含んでいることを示します。

NULL 標識列中の Y の値は、その列データが NULL であることを指定します。NULL 標識列に Y 以外の文字を指定した場合は、列データが NULL ではなく、METHOD L オプションで指定された列データがロードされることを指定することになります。

NULL 標識文字は MODIFIED BY オプションを使用して変更できます。

OF filetype

データの形式を指定します。

- ASC (区切りなし ASCII 形式)
- DEL (区切り付き ASCII 形式)
- IXF (統合交換フォーマット、PC バージョン)。同一のあるいは別の DB2 表からエクスポートされたことを意味します
- CURSOR (SELECT または VALUES ステートメントに対して宣言されたカーソル)。

PARTITIONED DB CONFIG

パーティション表へのロードの実行を可能にします。PARTITIONED DB CONFIG パラメーターを使用すると、パーティション・データベース固有の構成オプションを指定することができます。partitioned-db-option の値は以下のいずれかになります。

```

HOSTNAME x
FILE_TRANSFER_CMD x
PART_FILE_LOCATION x
OUTPUT_DBPARTNUMS x
PARTITIONING_DBPARTNUMS x
MODE x
MAX_NUM_PART_AGENTS x
ISOLATE_PART_ERRS x
STATUS_INTERVAL x
PORT_RANGE x
CHECK_TRUNCATION
MAP_FILE_INPUT x
MAP_FILE_OUTPUT x
TRACE x
NEWLINE
DISTFILE x
OMIT_HEADER

```

- 非パーティション環境内からロードが実行される場合、これは通常どおり動作します。パーティション・データベース構成オプションが指定される場合、SQL エラー 27959、理由コード 1 が戻されます。
- それ以上の修飾はしないで、パーティション・データベース環境では、MODE オプションは DB2_PARTITIONEDLOAD_DEFAULT レジストリー変数が NO に設定されている場合を除いて、PARTITION_AND_LOAD に対してデフォルトになります。この場合、以下のデフォルトが適用されます。MODE は LOAD_ONLY になり、OUTPUT_DBPARTNUMS はユーザーが現在接続している 1 つのデータベース・パーティションを含むリストになり、PART_FILE_LOCATION に関しては、ロード入力ファイル名が完全修飾されていなくて、クライアントとサーバーが同一の物理マシン上にある場合、現在のクライアントの作業パスになり、または、ファイル名が完全修飾の場合はロード入力ファイルのパス接頭部になります。このレジストリー変

数の目的は、バージョン 8 以前のロード・ユーティリティーのパーティション・データベース環境における動作を保存することにあります。

- パーティション・データベース環境で `MODE` オプションに `PARTITION_ONLY` を設定してロードを使用する場合、入力ファイルはパーティションに分割され、オプション・マップ・ヘッダーと、そのパーティション独自のデータを含むそれぞれの出力パーティションにファイルが作成されます。 `CURSOR` を除くすべてのファイル・タイプに関して、各出力パーティション上に作成されるファイル名は、 `< filename >. < xxx >` となります。 `< filename >` はロード・コマンドで指定された入力ファイルの名前で、 `< xxx >` はファイルが常駐するパーティションの番号です。また、各出力パーティション上のファイルのロケーションは、 `PART_FILE_LOCATION` オプションによって示されます (指定される場合)。このオプションが指定されない場合、入力ファイルのロケーションが、現在の作業ディレクトリーとなります。ファイル・タイプが `CURSOR` の場合、 `PART_FILE_LOCATION` オプションは必須となり、完全修飾ベース名を指定する必要があります。この場合、各パーティションに作成されるファイル名に適切なパーティション番号が付加されたものが、このベース名になります。
- パーティション・データベース環境で `MODE` オプションに `LOAD_ONLY` を設定してロードを使用する場合、ロードされるファイルはそれぞれの出力パーティション上に存在し、有効なパーティション・マップ・ヘッダーが含まれていると想定されます、 `CURSOR` を除くすべてのファイル・タイプに関して、各パーティション上のファイル名は `< filename >. < xxx >` になると想定されます。 `< filename >` はロード・コマンドで指定された入力ファイルの名前で、 `< xxx >` はファイルが常駐するパーティションの番号です。また、各パーティション上のファイルのロケーションは、 `PART_FILE_LOCATION` オプションによって示されます (指定される場合)。このオプションが指定されない場合、入力ファイル名のパス接頭部によって示されるロケーションから (入力ファイル名が完全修飾の場合)、または現行作業ディレクトリーから (入力ファイル名が完全修飾ではない場合) 読み取られます。ファイル・タイプが `CURSOR` の場合、 `PART_FILE_LOCATION` オプションは必須となり、完全修飾ベース名を指定する必要があります。この場合、各パーティション上のファイル名に適切なパーティション番号が付加されたものが、このベース名になります。
- パーティション・データベース環境で `MODE` オプションに `LOAD_ONLY_VERIFY_PART` を設定してロードを使用する場合、ロードされるファイルはそれぞれの出力パーティションに存在し、パーティション・マップ・ヘッダーは含まれていないと想定されます。ロードを使用すると、各ファイル内のデータが適切なパーティションにあるか検証します。適切なパーティションにない行はリジェクトされ、ダンプ・ファイルに送信されます。

す (これが指定されている場合)。各出力パーティション上のファイルの名前とロケーションは、LOAD_ONLY モードのファイル名と同じ規則に従います。

注: ファイル・タイプが CURSOR の場合、LOAD_ONLY_VERIFY_PART モードはサポートされません。

- ロードの CLIENT キーワードが指定される場合、リモート・ロードが許可されます。CLIENT が指定されたロードでは、PARTITION_AND_LOAD および PARTITION_ONLY モードのみサポートされます。

REPLACE

ロード・ユーティリティーを実行できる 4 つのモードのうちの 1 つ。表から既存データをすべて削除し、ロードされたデータを挿入します。表定義および索引定義は変更されません。階層間でデータを移動する際にこのオプションを使用する場合は、階層全体に関係したデータだけが置き換えられます。副表は置き換えられません。

このオプションは DATALINK 列を含む表ではサポートされません。

RESTART

ロード・ユーティリティーを実行できる 4 つのモードのうちの 1 つ。以前に割り込みを受けたロード操作を再開します。ロード操作は、ロード、作成、または削除フェーズの最後の整合点から自動的に続行されます。

RESTARTCOUNT

予約済み。

ROWCOUNT *n*

ロードするファイル内の物理レコードの数 *n* を指定します。ユーザーはファイル内の最初の *n* 個の行だけをロードできます。

SAVECOUNT *n*

ロード・ユーティリティーが *n* 行ごとに整合点を取ることを指定します。この値はページ・カウントに変換され、エクステント・サイズのインターバルに切り上げられます。メッセージは整合点において発行されるので、LOAD QUERY を使用してロード操作をモニターする場合には、このオプションを選択する必要があります。 *n* の値が十分な大きさをでないなら、各整合点で実行される活動の同期化によってパフォーマンスに影響してしまいます。

デフォルトはゼロですが、それは、必要がなければ整合点は確立されないことを意味します。

ファイル・タイプが CURSOR の場合、このオプションはサポートされていません。

SORT BUFFER *buffer-size*

このオプションは、ロード操作時に SORTHEAP データベース構成パラメータをオーバーライドする値を指定します。これは、索引とともに表をロードす

LOAD

る場合、また INDEXING MODE パラメーターが DEFERRED として指定されていない場合にのみ関係があります。指定される値は、SORTHEAP の値を超えることはできません。このパラメーターは、SORTHEAP の値を変更せずに多くの索引を持つ表をロードする際に使用されるソート・メモリーのスロットルで役に立ちます。これは、一般的な照会処理にも影響を与えます。

STATISTICS NO

統計データを収集せず、したがってカタログ内の統計データも変更しないことを指定します。これがデフォルトです。

STATISTICS YES

表およびすべての既存索引の統計データを収集するよう指定します。このオプションがサポートされるのは、ロード操作が REPLACE モードの場合だけです。

WITH DISTRIBUTION

分散統計を収集するよう指定します。

AND INDEXES ALL

表統計と索引統計の両方を収集するよう指定します。

FOR INDEXES ALL

索引統計だけを収集するよう指定します。

DETAILED

拡張された索引統計を収集するよう指定します。

TEMPFILES PATH temp-pathname

ロード操作時に一時ファイルを作成する場合に使うパスの名前を指定します。これはサーバー・データベース・パーティションにしたがって完全に修飾する必要があります。

一時ファイルは、ファイル・システムのスペースを使用します。場合によっては、このスペースが相当必要になります。以下に示すのは、すべての一時ファイルにどの程度のファイル・システム・スペースを割り振るべきかの見積もりです。

- DATALINK 値を含む重複行またはリジェクト行ごとに 4 バイト
- ロード・ユーティリティーが生成するメッセージごとに 136 バイト
- データ・ファイルに長フィールド・データまたは LOB が含まれている場合は、15KB のオーバーヘッド。INSERT オプションを指定した場合で、表の中に多量の長フィールドまたは LOB データが既に含まれている場合には、この数値はこれよりもかなり大きくなる場合があります。

TERMINATE

ロード・ユーティリティーを実行できる 4 つのモードのうちの 1 つ。以前に割り込みを受けたロード操作を終了し、ロード操作が開始された時点まで操作をロールバックします。途中で整合点があっても通過します。その操作に関係

する表スペースの状態は通常に戻され、すべての表オブジェクトの整合性が保たれます (索引オブジェクトが無効とマークされる場合がありますが、そのような場合には、次のアクセス時に索引の再作成が自動的に行われます)。終了するロード操作がロード REPLACE の場合、その表はロード TERMINATE 操作完了後に空の表まで切り捨てられます。終了するロード操作がロード INSERT の場合、その表はロード TERMINATE 操作完了後も元のレコードをすべて保持します。

ロード終了オプションでは、表スペースのバックアップ・ペンディング状態は解除されません。

注: このオプションは DATALINK 列を含む表ではサポートされません。

USING directory

予約済み。

WARNINGCOUNT *n*

n 個の警告後に、ロード操作を停止します。このパラメーターは、警告は予期されないが、正しいファイルと表が使用されていることの確認が望まれる場合に設定してください。 *n* がゼロの場合、またはこのオプションが指定されていない場合、何度警告が出されてもロード操作は続行します。警告のしきい値に達したためにロード操作が停止された場合でも、あらためて RESTART モードでロード操作を開始できます。ロード操作は最後の整合点から自動的に続行されます。または、入力ファイルの先頭から REPLACE モードであらためてロード操作を開始できます。

WITHOUT PROMPTING

データ・ファイルのリストにロードするすべてのファイルを含め、しかもリストに含まれる装置またはディレクトリーがロード操作全体で十分であるということ指定します。続きの入力ファイルが見つからなかったり、ロード操作が終了する前にコピー先がいっぱいになるとロード操作は失敗し、表はロード・ペンディング状態のままになります。

このオプションを指定しない場合に、テープ装置がコピー・イメージ用のテープの終わりに達した場合、またはリスト中の最後の項目がテープ装置であった場合は、ユーザーに対してその装置に新しいテープを装着するよう求めるプロンプトが出されます。

例:

例 1

TABLE1 に以下の 5 つの列があるとします。

- COL1 VARCHAR 20 NOT NULL WITH DEFAULT
- COL2 SMALLINT
- COL3 CHAR 4

LOAD

- COL4 CHAR 2 NOT NULL WITH DEFAULT
- COL5 CHAR 2 NOT NULL

ASCFILE1 に以下の 6 つのエレメントがあるとします。

- ELE1 positions 01 to 20
- ELE2 positions 21 to 22
- ELE5 positions 23 to 23
- ELE3 positions 24 to 27
- ELE4 positions 28 to 31
- ELE6 positions 32 to 32
- ELE6 positions 33 to 40

データ・レコードは以下のとおりです。

```
1...5....10...15...20...25...30...35...40
Test data 1      XXN 123abcdN
Test data 2 and 3  QQY  wxyzN
Test data 4,5 and 6 WVN6789  Y
```

以下のコマンドは、ファイルから表をロードします。

```
db2 load from ascfile1 of asc modified by striptblanks recLEN=40
method L (1 20, 21 22, 24 27, 28 31)
null indicators (0,0,23,32)
insert into table1 (col1, col5, col2, col3)
```

注:

1. **MODIFIED BY** パラメーターで **striptblanks** を指定すると、**VARCHAR** 列の中のブランクが切り捨てられるようになります (たとえば行 1、2、および 3 の長さがそれぞれ 11、17、および 19 バイトである COL1)。
2. **MODIFIED BY** パラメーターで **recLEN=40** を指定すると、各入力レコードの最後が改行文字でなく、各レコードが 40 バイト長であることを指定することになります。最後の 8 バイトは、表のロードには使われません。
3. COL4 は入力ファイルにはないので、そのデフォルト (**NOT NULL WITH DEFAULT** と定義されています) を使って TABLE1 に挿入されます。
4. 位置 23 と 32 は、特定の行で TABLE1 の COL2 と COL3 が NULL としてロードされるかどうかを指示するために使われます。ある特定のレコードの、その列の NULL 標識位置が Y である場合、その列は NULL になります。N なら、入力レコード中のその列のデータ位置のデータ値 (L(.....)) で定義される) は、その行の列データのソースとして使用されます。この例では、行 1 のどの列も NULL ではなく、行 2 の COL2 は NULL であり、行 3 の COL3 は NULL です。
5. この例では、COL1 と COL5 の NULL INDICATORS は 0 (ゼロ) として指定されますが、それはそのデータを NULL 不可能であることを示しています。

6. 特定の列の NULL INDICATOR は、入力レコード内のどこにでも指定できますが、その位置は指定する必要があり、かつ Y または N の値を指定する必要があります。

例 2 (ファイルから LOB をロードする)

TABLE1 に以下の 3 つの列があるとします。

- COL1 CHAR 4 NOT NULL WITH DEFAULT
- LOB1 LOB
- LOB2 LOB

ASCFILE1 には以下の 3 つのエLEMENTがあるとします。

- ELE1 positions 01 to 04
- ELE2 positions 06 to 13
- ELE3 positions 15 to 22

次に示すファイルは、 /u/user1 または /u/user1/bin のどちらかにあります。

- ASCFILE2 has LOB data
- ASCFILE3 has LOB data
- ASCFILE4 has LOB data
- ASCFILE5 has LOB data
- ASCFILE6 has LOB data
- ASCFILE7 has LOB data

ASCFILE1 内のデータ・レコード

```
1...5...10...15...20...25...30.
REC1 ASCFILE2 ASCFILE3
REC2 ASCFILE4 ASCFILE5
REC3 ASCFILE6 ASCFILE7
```

以下のコマンドは、ファイルから表をロードします。

```
db2 load from ascfile1 of asc
      lobs from /u/user1, /u/user1/bin
      modified by lobsinfile reclen=22
      method L (1 4, 6 13, 15 22)
      insert into table1
```

注:

1. MODIFIED BY パラメーターの中で lobsinfile を指定すると、ファイルからすべての LOB データをロードすることをローダーに対して指定することになります。
2. MODIFIED BY パラメーターで reclen=22 を指定すると、各入力レコードの最後が改行文字でなく、各レコードが 22 バイト長であることを指定することになります。

LOAD

- LOB データは、ASCFILE2 から ASCFILE7 までの 6 つのファイルに入っています。各ファイルには、特定の行の LOB 列をロードするのに使用されるデータが入れられています。LOB と他のデータのリレーションシップは、ASCFILE1 に指定します。このファイルの最初のレコードは、REC1 を行 1 の COL1 にするようローダーに指示します。行 1 の LOB1 をロードするには ASCFILE2 の内容が使われ、ASCFILE3 の内容は行 1 の LOB2 をロードするのに使われます。同じように、行 2 の LOB1 および LOB2 をロードするには ASCFILE4 と ASCFILE5 が使われ、行 3 の LOB をロードするには ASCFILE6 と ASCFILE7 が使われます。
- これらのファイルがローダーで必要になった場合には、名前の指定された LOB ファイルを探索するのに使われる 2 つのパスが、LOBS FROM パラメーターに含まれています。
- lobsinfile 修飾子を指定しないで ASCFILE1 (区切りなしの ASCII ファイル) から直接 LOB をロードする場合は、以下の規則を守ってください。
 - LOB を含めたレコードの全長は 32KB 以下でなければなりません。
 - 入力レコード内の LOB フィールドは固定長でなければならず、必要なら LOB データにブランクを埋め込まなければなりません。
 - LOB をデータベースに挿入する際に、LOB の埋め込みに使われる後続ブランクを除去できるよう、striptblanks 修飾子を指定する必要があります。

例 3 (ダンプ・ファイルの使用)

表 FRIENDS は、以下のように定義されています。

```
table friends "( c1 INT NOT NULL, c2 INT, c3 CHAR(8) )"
```

以下のデータ・レコードをこの表にロードしようとする、

```
23, 24, bobby
, 45, john
4,, mary
```

最初の INT が NULL で、列定義に NOT NULL が指定されているため、第 2 行はリジェクトされます。DEL 形式と互換でない開始文字を含む列は、エラーを生成し、レコードはリジェクトされます。そのようなレコードは、ダンプ・ファイルに書き込むことができます。

文字区切り文字の外側にある列の DEL データは無視されますが、警告が生成されません。例:

```
22,34,"bob"
24,55,"sam" sdf
```

ユーティリティーは、表の第 3 列に "sam" をロードし、警告の中で文字 "sdf" にフラグが付けられます。このレコードはリジェクトされません。別の例を考えましょう。

```
22 3, 34,"bob"
```

ユーティリティは 22,34,"bob" をロードし、列 1 の中で 22 の後のデータは無視されたという警告を生成します。このレコードはリジェクトされません。

例 4 (DATALINK データのロード)

下記のコマンドは、DEL 形式のデータが含まれている入力ファイル delfile1 から、表 MOVIE TABLE をロードします。

```
db2 load from delfile1 of del
      modified by dldel|
      insert into movietable (actorname, description, url_making_of,
      url_movie) datalink specification (dl_url_default_prefix
      "http://narang"), (dl_url_replace_prefix "http://bomdel"
      dl_url_suffix ".mpeg") for exception excptab
```

注:

1. この表には下記の 4 つの列が含まれています。

actorname	VARCHAR(n)
description	VARCHAR(m)
url_making_of	DATALINK (with LINKTYPE URL)
url_movie	DATALINK (with LINKTYPE URL)

2. 入力ファイルの中の DATALINK データのサブフィールド区切り文字は、縦線 (|) 文字です。
3. url_making_of の列値に接頭部文字シーケンスが含まれていないなら、“http://narang” が使用されます。
4. url_movie の非 NULL 列値には、接頭部として “http://bomdel” が付けられます。既存の値は置き換えられます。
5. url_movie の非 NULL 列値のパスには、“.mpeg” が付加されます。たとえば、url_movie の列値が “http://server1/x/y/z” なら、それは “http://bomdel/x/y/z.mpeg” として保管されます。その値が “/x/y/z” なら “http://bomdel/x/y/z.mpeg” として保管されます。
6. 表のロード中にユニーク索引または DATALINK 例外が発生したなら、関係するレコードが表から削除され、例外表 excptab に入れられます。

例 5 (IDENTITY 列がある表へのロード)

TABLE1 には以下の 4 つの列があります。

- C1 VARCHAR(30)
- C2 INT GENERATED BY DEFAULT AS IDENTITY
- C3 DECIMAL(7,2)
- C4 CHAR(1)

TABLE2 は TABLE1 と同じですが、C2 が GENERATED ALWAYS IDENTITY 列である点が異なります。

LOAD

DATAFILE1 のデータ・レコード (DEL 形式):

```
"Liszt"  
"Hummel",,187.43, H  
"Grieg",100, 66.34, G  
"Satie",101, 818.23, I
```

DATAFILE2 のデータ・レコード (DEL 形式):

```
"Liszt", 74.49, A  
"Hummel", 0.01, H  
"Grieg", 66.34, G  
"Satie", 818.23, I
```

注:

1. 以下のコマンドは、DATAFILE1 で行 1 および 2 への ID 値の提供がないので、それらの行のための ID 値を生成します。ただし、行 3 および 4 は、それぞれユーザー提供の ID 値 100 と 101 が割り当てられます。

```
db2 load from datafile1.del of del replace into table1
```

2. DATAFILE1 を TABLE1 にロードしてすべての行に対する ID 値を生成するには、以下のコマンドのいずれかを発行します。

```
db2 load from datafile1.del of del method P(1, 3, 4)  
replace into table1 (c1, c3, c4)  
db2load from datafile1.del of del modified by identityignore  
replace into table1
```

3. DATAFILE2 を TABLE1 にロードして各行に対する ID 値を生成するには、以下のコマンドのいずれかを発行します。

```
db2 load from datafile2.del of del replace into table1 (c1, c3, c4)  
db2 load from datafile2.del of del modified by identitymissing  
replace into table1
```

4. ID 値 100 と 101 を行 3 および 4 に割り当てるために DATAFILE1 を TABLE2 にロードするには、以下のコマンドを実行します。

```
db2 load from datafile1.del of del modified by identityoverride  
replace into table2
```

この場合、ユーティリティには、ユーザー提供の値を優先して、システム生成の ID 値に上書きするように指示しているため、行 1 および 2 はリジェクトされません。ユーザー提供の値が存在しない場合でも、IDENTITY 列が暗黙的に非 NULL であるため、この行はリジェクトする必要があります。

5. DATAFILE1 を TABLE2 に、IDENTITY 関連のファイル・タイプ修飾子を使用せずにロードした場合、行 1 と 2 はロードされますが、行 3 と 4 はリジェクトされません。その理由は、それらが固有の非 NULL 値を提供し、IDENTITY 列が GENERATED ALWAYS であるからです。

例 6 (CURSOR ファイル・タイプを使用したロード)

表 ABC.TABLE1 には以下の 3 つの列があります。

```
ONE INT
TWO CHAR(10)
THREE DATE
```

表 ABC.TABLE2 には以下の 3 つの列があります。

```
ONE VARCHAR
TWO INT
THREE DATE
```

以下のコマンドを実行すると、すべてのデータが ABC.TABLE1 から ABC.TABLE2 にロードされます。

```
db2 declare mycurs cursor for select two,one,three from abc.table1
db2 load from mycurs of cursor insert into abc.table2
```

使用上の注意:

データは、入力ファイル内に並んでいる順序でロードされます。特定の順序にしたいなら、ロードの前にデータをソートしておく必要があります。

ロード・ユーティリティーは、既存の定義に基づいて索引を作成します。ユニーク・キーの重複を処理するのに、例外表が使用されます。ユーティリティーは、参照保全を強制したり、制約検査を実行したり、ロードする表に従属するサマリー表を更新したりすることはありません。参照制約またはチェック制約を含む表は、チェック・ペンディング状態になります。REFRESH IMMEDIATE として定義されているサマリー表、およびロードする表に依存するサマリー表は、チェック・ペンディング状態になります。表のチェック・ペンディング状態を解除するには、SET INTEGRITY ステートメントを発行してください。ロード操作は、複製されたサマリー表では実行できません。

クラスタリング索引が表に存在する場合、ロード前にクラスタリング索引でデータをソートしてください。ただし、データは多次元クラスタリング (MDC) 表にロードする前にソートする必要はありません。

DB2 Data Links Manager についての考慮事項

各 DATALINK 列ごとに、括弧内にそれぞれ 1 つの列を指定できます。それぞれの列の指定は、1 つ以上の DL_LINKTYPE、接頭部、および DL_URL_SUFFIX 指定で構成されます。接頭部 情報は、DL_URL_REPLACE_PREFIX、または DL_URL_DEFAULT_PREFIX の指定のいずれかになります。

DATALINK 列指定の数は、表で定義されている DATALINK 列の数と同じだけ指定できます。指定の順序は、挿入列リストの中での DATALINK 列の順序 (挿入列リストが INSERT INTO (insert-column, ...) で指定されている場合) か、または表定義内の順序 (insert-column が指定されていない場合) に従います。

LOAD

たとえば、表に列 C1、C2、C3、C4、および C5 があり、そのうち C2 と C5 だけが DATALINK 型であり、挿入列 (insert-column) リストが (C1、C5、C3、C2) である場合、2 つの DATALINK 列を指定する必要があります。最初の列の指定は C5 用であり、2 番目の列の指定は C2 用です。挿入列リストを指定しない場合、最初の列の指定は C2 用になり、2 番目の列の指定は C5 用になります。

複数の DATALINK 列があり、一部の列では特別な指定が必要ではない場合、列の指定には、指定の順序をはっきりと示すために、少なくとも括弧を含める必要があります。どの列にも指定が行われない場合は、空の括弧のリスト全体を除くことができます。したがって、デフォルトで十分な場合には、DATALINK を指定する必要はありません。

FILE LINK CONTROL で定義されている DATALINK 列を含む表にデータをロードする場合は、ロード・ユーティリティを呼び出す前に、以下のステップを実行してください。(すべての DATALINK 列が NO LINK CONTROL として定義されている場合、これらのステップは必要ありません。)

1. DATALINK 列の値によって参照される Data Links サーバーに、DB2 Data Links Manager がインストールされていることを確認する。分散ファイル・システム (DFS) の場合、DB2 Data Links Manager がターゲット・セル内で登録されていることを確認してください。
2. データベースが DB2 Data Links Manager に登録されていることを確認します。
3. DATALINK 値として挿入されるすべてのファイルを、適切な Data Links サーバーにコピーします。
4. Data Links サーバー上の DB2 Data Links Manager に接頭部名 (複数可) を定義します。
5. (ロードする) DATALINK データによって参照される Data Links サーバーを、DB2 Data Links Manager 構成ファイルに登録します。DFS の場合、DB2 Data Links Manager 構成ファイルの (ロードされる) DATALINK データによって参照されているターゲット構成で、セルを登録してください。

ロード・ユーティリティの実行中に、DB2 と Data Links サーバー間の接続が失敗し、ロード操作も失敗してしまう場合があります。その場合には、以下のようしてください。

1. Data Links サーバーおよび DB2 Data Links Manager を開始します。
2. ロード再開操作を起動します。

ロード操作中に失敗したリンクはデータ保全性違反と見なされ、ユニーク索引違反と同じ方法で処理されます。そのため、1 つ以上の DATALINK 列を含む表をロードする場合の特別な例外が定義されています。

入力ファイル内での DATALINK 情報の表示

LINKTYPE (現在のところ URL のみサポート) は、DATALINK 情報の一部として指定されていません。LINKTYPE は、LOAD または IMPORT コマンドで指定され、PC/IXF タイプの入力ファイルの場合は、適切な列記述子レコードで指定されます。

URL LINKTYPE の DATALINK 情報の構文は、以下のとおりです。



urlname と *comment* はいずれもオプションです。どちらも省略した場合、NULL 値が代入されます。

urlname

この URL 名は有効な URL 構文に適合していなければなりません。

注:

1. 現在 "http"、"file"、"unc"、および "dfs" などはスキーマ名としてのみ許可されています。
2. URL 名の接頭部 (スキーマ、ホスト、およびポート) はオプションです。DFS の場合、接頭部は、スキーマ・セル名、ファイル・スペース接合部分のことです。接頭部がない場合は、ロード・ユーティリティーまたはインポート・ユーティリティーの DL_URL_DEFAULT_PREFIX または DL_URL_REPLACE_PREFIX 指定の接頭部が使われます。そのどちらも指定されていない場合、デフォルトの接頭部として "file://localhost" が使われます。したがって、ローカル・ファイルの場合、LOAD または IMPORT コマンド内で DATALINK 列を指定せずに、絶対パス名で指定したファイル名を URL 名として入力することができます。
3. 接頭部が URL 名に付けられている場合も、ロードまたはインポート操作時には、DL_URL_REPLACE_PREFIX で指定した異なる接頭部名によってオーバーライドされます。
4. (DL_URL_SUFFIX を指定した場合、それを追加した後の) 「path」は、リモート・サーバーにあるリモート・ファイルの絶対パス名です。相対パス名は使用できません。HTTP サーバーのデフォルトのパス接頭部は使用されません。

dl_delimiter

区切り付き ASCII (DEL) ファイル形式の場合、dl_{del} 修飾子で指定した文字、あるいは LOAD または IMPORT コマンドのデフォルトの文字。区切りなし ASCII (ASC) ファイル形式の場合、これを文字順序 ¥; (円記号とそれに続くセミコロン) に対応させる必要があります。空白文字 (ブランクやタブなど) は、このパラメーターに指定した値の前後に置くことができます。

comment

DATALINK 値のコメント部分。区切り付き ASCII (DEL) ファイル形式で指定する場合、*comment* テキストは、文字ストリング区切り文字で囲む必要があ

LOAD

ります。文字ストリング区切り文字は、デフォルトでは二重引用符 (") です。
この文字ストリング区切り文字は、LOAD または IMPORT コマンドで
MODIFIED BY *filetype-mod* を指定することによりオーバーライドできます。
コメントを指定しない場合、このコメントはデフォルトでは長さがゼロのスト
リングになります。

次に示すのは、区切り付き ASCII (DEL) ファイル形式での DATALINK データの例で
す。

- `http://www.almaden.ibm.com:80/mrep/intro.mpeg; "Intro Movie"`
これは、以下のような部分から構成されています。
 - スキーム = `http`
 - サーバー = `www.almaden.ibm.com`
 - パス = `/mrep/intro.mpeg`
 - 注釈 = `"Intro Movie"`
- `file://narang/u/narang; "InderPal's Home Page"`
これは、以下のような部分から構成されています。
 - スキーム = `file`
 - サーバー = `narang`
 - パス = `/u/narang`
 - 注釈 = `"InderPal's Home Page"`

次に示すのは、区切りなし ASCII (ASC) ファイル形式での DATALINK データの例で
す。

- `http://www.almaden.ibm.com:80/mrep/intro.mpeg%;Intro Movie`
これは、以下のような部分から構成されています。
 - スキーム = `http`
 - サーバー = `www.almaden.ibm.com`
 - パス = `/mrep/intro.mpeg`
 - 注釈 = `"Intro Movie"`
- `file://narang/u/narang%; InderPal's Home Page`
これは、以下のような部分から構成されています。
 - スキーム = `file`
 - サーバー = `narang`
 - パス = `/u/narang`
 - 注釈 = `"InderPal's Home Page"`

以下に、DATALINK データの例を示します。列のロードまたはインポート指定が DL_URL_REPLACE_PREFIX ("http://qso") であるとしています。

- http://www.almaden.ibm.com/mrep/intro.mpeg
これは、以下のような部分から構成されています。
 - スキーム = http
 - サーバー = qso
 - パス = /mrep/intro.mpeg
 - 注釈 = NULL ストリング
- /u/me/myfile.ps
これは、以下のような部分から構成されています。
 - スキーム = http
 - サーバー = qso
 - パス = /u/me/myfile.ps
 - 注釈 = NULL ストリング

表 234. 有効なファイル・タイプ修飾子 (LOAD)

修飾子	説明
すべてのファイル形式	
anyorder	この修飾子は、 <i>cpu_parallelism</i> パラメーターとともに使用され、ソース・データ順序を保つことが必要でないことを指定し、それによって SMP システムでさらにパフォーマンスを高めます。 <i>cpu_parallelism</i> の値が 1 の場合、このオプションは無視されます。SAVECOUNT > 0 の場合、整合点後のクラッシュ・リカバリーでは、データを順番にロードする必要があるため、このオプションはサポートされません。
fastparse	<p>ユーザー提供の列値に対して簡略化された構文チェックが実行され、パフォーマンスが向上します。このオプションの下でロードした表は、体系的に正確であることが保証され、ユーティリティは、セグメント化違反またはトラップを防ぐための十分なデータ・チェックを実行することが保証されます。正しい形式のデータが正しくロードされます。</p> <p>たとえば、ASC ファイル内の整数列のフィールド項目として値 123qwr4 が検出された場合、この値は有効な数値を表すものではないので、ロード・ユーティリティは通常構文エラーのフラグを付けます。fastparse を指定した場合、構文エラーは検出されず、整数フィールドに任意の数値がロードされます。この修飾子を使用する場合は、正しい形式のデータだけを使用するように注意してください。ASCII データでこのオプションを使用するとパフォーマンスはかなり向上しますが、PC/IXF データでこのオプションを使用しても fastparse のパフォーマンスはそれほど向上しません。これは、IXF がバイナリー形式であり、fastparse が ASCII から内部形式への構文解析および変換に影響するためです。</p> <p>ファイル・タイプが CURSOR の場合、このオプションはサポートされていません。</p>

表 234. 有効なファイル・タイプ修飾子 (LOAD) (続き)

修飾子	説明
generatedignore	この修飾子はロード・ユーティリティに対して、すべての生成列のデータがデータ・ファイル内に存在するが、それらのデータは無視するべきものであることを通知します。この結果、すべての生成列の値はユーティリティによって生成されます。この修飾子は、generatedmissing または generatedoverride 修飾子とともに使用することはできません。
generatedmissing	この修飾子を指定すると、ユーティリティは、入力データ・ファイルには生成列のデータが全く入っていない (NULL さえない) と見なし、その列に NULL をロードします。この結果、すべての生成列の値はユーティリティによって生成されます。この修飾子は、generatedignore または generatedoverride 修飾子とともに使用することはできません。
generatedoverride	<p>この修飾子は、(こうした列のタイプの通常の規則に反して) 表内のすべての生成列で、ユーザーのデータを受け入れるようにロード・ユーティリティに指示します。これが役立つのは、別のデータベース・システムからデータを移行する場合や、ROLLFORWARD DATABASE コマンドで RECOVER DROPPED TABLE オプションを使用してリカバリーしたデータから表をロードする場合です。この修飾子を使用すると、データが入っていない行や NULL 不可の生成列に対する NULL データはすべてリジェクトされます (SQL3116W)。</p> <p>注: この修飾子が使用される場合、表はチェック・ペンディング状態になります。ユーザーが提供する値をチェックせずに、表をチェック・ペンディング状態から解放するには、ロード操作後に以下のコマンドを発行します。</p> <pre>SET INTEGRITY FOR < table-name > GENERATED COLUMN IMMEDIATED UNCHECKED</pre> <p>表の CHECK PENDING 状態を解除し、ユーザー定義の値のチェックを強制するには、ロード操作の後以下のコマンドを発行してください。</p> <pre>SET INTEGRITY FOR < table-name > IMMEDIATE CHECKED.</pre> <p>この修飾子は、generatedmissing または generatedignore 修飾子と共に使用することはできません。</p>
identityignore	この修飾子はロード・ユーティリティに対して、IDENTITY 列のデータがデータ・ファイル内に存在するが、それらのデータは無視するべきものであることを通知します。この結果、すべて ID 値はユーティリティによって生成されます。この動作は、GENERATED ALWAYS および GENERATED BY DEFAULT IDENTITY 列のどちらの場合も同じです。つまり、GENERATED ALWAYS 列の場合、リジェクトされる行はないという意味です。この修飾子は、identitymissing または identityoverride 修飾子とともに使用することはできません。
identitymissing	この修飾子を指定すると、ユーティリティは、IDENTITY 列のデータが入力データ・ファイルに入っていない (NULL も入っていない) ものと見なし、行ごとに値を生成します。この動作は、GENERATED ALWAYS および GENERATED BY DEFAULT IDENTITY 列のどちらの場合も同じです。この修飾子は、identityignore または identityoverride 修飾子とともに使用することはできません。

表 234. 有効なファイル・タイプ修飾子 (LOAD) (続き)

修飾子	説明
identityoverride	<p>この修飾子を使用するのは、GENERATED ALWAYS として定義されている IDENTITY 列が、ロードされる表にある場合だけです。この修飾子はユーティリティーに対し、そのような列に関して、明示的な非 NULL データを受け入れる (これらのタイプの IDENTITY 列に関する通常の規則に反する) ように指示します。これは、表を GENERATED ALWAYS と定義することが必要になるときに他のデータベース・システムからデータを移行する場合や、DROPPED TABLE RECOVERY オプションを指定した ROLLFORWARD DATABASE コマンドを使用してリストアしたデータから表をロードする場合に役立ちます。この修飾子を使用した場合、IDENTITY 列でデータまたは NULL データの入っていない行はリジェクトされます (SQL3116W)。この修飾子は、identitymissing または identityignore 修飾子とともに使用することはできません。</p> <p>注: このオプションが使用されていると、ロード・ユーティリティーは、表の IDENTITY 列内の値の固有性の保守または検証を行いません。</p>
indexfreespace=x	<p>x は 0 ~ 99 の整数です。その値は、各索引ページの中で索引再作成ロード時のフリー・スペースとして残しておく部分の割合を示すパーセントとして解釈されます。ロードに INDEXING MODE INCREMENTAL を指定すると、このオプションは無視されます。ページの最初の項目は、制限なしで追加されます。それより後の項目は、フリー・スペースのパーセントしきい値内である場合に追加されます。デフォルト値は、CREATE INDEX の実行時に使用した値です。</p> <p>この値は、CREATE INDEX ステートメントに指定された PCTFREE 値よりも優先して使用され、レジストリー変数 DB2 INDEX FREE は、索引のフリー・スペースよりも優先して使用されます。索引のフリー・スペース・オプションは、索引のリーフ・ページにのみ影響を与えます。</p>

表 234. 有効なファイル・タイプ修飾子 (LOAD) (続き)

修飾子	説明
lobsinfile	<p><i>lob-path</i> には、LOB データを含むファイルへのパスを指定します。ASC、DEL、または IXF ロード入力ファイルには、LOB 列に LOB データを含むファイルの名前が入っています。</p> <p>ファイル・タイプが CURSOR の場合、このオプションはサポートされていません。</p> <p>LOBS FROM 文節は、“lobsinfile” 修飾子が使用されているときの、LOB ファイルの場所を指定します。LOBS FROM 文節には、“lobsinfile” 修飾子のコンテキスト以外の意味はありません。LOBS FROM 文節は、データのロード中に、LOAD コーティリティーに LOB ファイルを検索するためのパスのリストを送ります。</p> <p>各パスには、データ・ファイル内で LOB ロケーション指定子 (LLS) によって示される 1 つ以上の LOB を含む、少なくとも 1 つのファイルが含まれます。LLS は、LOB ファイル・パスに保管されるファイル内の LOB のロケーションのストリング表記です。LLS の形式は、<i>filename.ext.nnn.mmm/</i> です。<i>filename.ext</i> は LOB を含むファイルの名前、<i>nnn</i> はファイル内の LOB のオフセット (バイト単位)、<i>mmm</i> は LOB の長さ (バイト単位) を表します。たとえば、ストリング <i>db2exp.001.123.456/</i> がデータ・ファイルに保存される場合、LOB はファイル <i>db2exp.001</i> のオフセット 123 に位置し、456 バイト長です。</p> <p>NULL LOB を指定するには、サイズに -1 と入力します。サイズを 0 と指定すると、長さが 0 の LOB として扱われます。長さが -1 の NULL LOB の場合、オフセットとファイル名は無視されます。たとえば、NULL LOB の LLS は <i>db2exp.001.7.-1/</i> です。</p>
noheader	<p>ヘッダー検査コードをスキップします (単一パーティション・データベースのパーティション・グループに存在する表へのロード操作にのみ適用します)。</p> <p>オートローダー・コーティリティーは、複数パーティションのデータベース・パーティション・グループの表にデータを提供している各ファイルに、ヘッダーを書き込みます。1 つのノード・グループに存在する表に対してデフォルトの MPP ロード (モード PARTITION_AND_LOAD) が使用される場合、ファイルにはヘッダーが含まれないと想定されます。よって、noheader 修飾子は必要ありません。LOAD_ONLY モードが使用される場合、ファイルにはヘッダーが含まれると想定されます。ヘッダーが含まれないファイルを使って操作を実行したい場合のみ、noheader 修飾子を使用する必要があります。</p>
norowwarnings	リジェクトされた行についての警告をすべて抑制します。
pagefreospace= <i>x</i>	<p><i>x</i> は 0 ~ 100 の整数です。その値は、各データ・ページ内でフリー・スペースとして残される部分のパーセンテージとして解釈されます。最小行サイズのため、指定した値が無効である場合 (たとえば、最も小さい行の大きさが 3000 バイトで、<i>x</i> の値が 50 である場合)、その行は新しいページに置かれます。値 100 が指定された場合には、各行がそれぞれ新しいページに置かれます。</p> <p>注: 表の PCTFREE 値は、ページごとに指定されたフリー・スペースの量を決定します。ロード操作の pagefreospace 値、または表の PCTFREE 値が設定されていない場合、コーティリティーは、それぞれのページにできるだけ大きなスペースを割り当てます。pagefreospace によって設定された値は、その表について指定された PCTFREE 値をオーバーライドします。</p>

表 234. 有効なファイル・タイプ修飾子 (LOAD) (続き)

修飾子	説明
subtableconvert	単一副表へのロードの場合のみ有効。これを使う場合として典型的な例は、正規の表からデータをエクスポートした後、この修飾子を使ってロード操作を呼び出してそのデータを単一の副表に変換する場合です。
totalreespace= x	<p>x は 0 以上の整数です。この値は表内の合計ページのうち、表の終わりにフリー・スペースとして追加される部分のパーセンテージとして解釈されます。たとえば、x が 20 で、データのロード後に、表に 100 のデータ・ページがある場合、20 の追加の空ページが付加されます。この表のデータ・ページの合計数は、120 になります。データ・ページの総数は、表の索引ページの数には影響を与えません。このオプションは、索引オブジェクトには影響を与えません。</p> <p>注: このオプションを指定して 2 つのロードが行われる場合、2 番目のロードは、最初のロードによって最後に付加された余分のスペースを再利用しません。</p>
usedefaults	<p>ターゲット表の列に対応するソース列が指定されているが、1 つ以上の行インスタンスのデータが入っていない場合、デフォルト値がロードされます。欠落データの例は、以下のとおりです。</p> <ul style="list-style-type: none"> • DEL ファイルの場合、列に「.,」が指定された場合 • DEL/ASC/WSF ファイルの場合、列が不足している行、または元の指定では十分な長さのない行。 <p>このオプションが指定されていない場合、行インスタンスのソース列にデータがないと、以下のいずれかの処理が行われます。</p> <ul style="list-style-type: none"> • 列が NULL 可能な場合、NULL がロードされます。 • 列が NULL 不可能の場合、ユーティリティはその行をリジェクトします。
ASCII ファイル形式 (ASC/DEL)	
codepage= x	<p>x は ASCII 文字ストリングです。その値は、入力データ・セット内のデータのコード・ページとして解釈されます。ロード操作時に、文字データ (および文字に指定された数値データ) は、このコード・ページからデータベースのコード・ページへ変換されます。</p> <p>以下の規則が適用されます。</p> <ul style="list-style-type: none"> • 純粋な DBCS (グラフィック)、混合 DBCS、および EUC の場合、区切り文字は x00 ~ x3F の範囲に制限されます。 • EBCDIC コード・ページで指定された DEL データの場合、区切り文字はシフトインおよびシフトアウト DBCS 文字と一致しない場合があります。 • nullindchar には、コード・ポイントが x20 ~ x7F の標準 ASCII セットに含まれる記号を指定する必要があります。これは、ASCII シンボルおよびコード・ポイントを示します。EBCDIC データでは、コード・ポイントが異なるとしても、対応するシンボルを使用できます。 <p>ファイル・タイプが CURSOR の場合、このオプションはサポートされていません。</p>

表 234. 有効なファイル・タイプ修飾子 (LOAD) (続き)

修飾子	説明
dateformat="x"	<p>x はソース・ファイルの日付の形式です。 ^a 有効な日付エレメントは以下のとおりです。</p> <p>YYYY - 年 (0000 ~ 9999 の範囲の 4 桁の数字) M - 月 (1 ~ 12 の範囲の 1 桁または 2 桁の数) MM - 月 (1 ~ 12 の範囲の 2 桁の数。 M とは相互排他的) D - 日 (1 ~ 31 の範囲の 1 桁または 2 桁の数) DD - 日 (1 ~ 31 の範囲の 2 桁の数。 D とは相互排他的) DDD - 日 (年間) (001 ~ 366 の範囲の 3 桁の数。他の日または月エレメントとは 相互排他的)</p> <p>デフォルト値の 1 が、指定されない各エレメントに割り当てられます。日付形式の例を以下に示します。</p> <p>"D-M-YYYY" "MM.DD.YYYY" "YYYYDDD"</p>
dumpfile = x	<p>x は、リジェクトされた行を書き込む例外ファイルの (サーバー・データベース・パーティションによる) 完全修飾名です。 1 レコード当たり、最大で 32KB のデータが書き込まれます。以下に、ダンプ・ファイルの指定方法の例を示します。</p> <pre>db2 load from data of del modified by dumpfile = /u/user/filename insert into table_name</pre> <p>注:</p> <ol style="list-style-type: none"> パーティション・データベース環境の場合、パスはロードを実行するデータベース・パーティションにローカルなものでなければなりません。それによって、並行して実行される複数のロード操作が同じファイルに書き込むことを防ぐことができます。 ファイルの内容は、非同期バッファ・モードでディスクに書き込まれます。ロード操作が失敗したり割り込まれる場合には、ディスクにコミットされるレコード数が正確に知らされておらず、LOAD RESTART 後の整合性が保証されていない可能性があります。ファイルが完全であるとされるのは、1 回のパスの中で開始して完了するロード操作の場合だけです。 この修飾子では、ファイル拡張子が複数のファイル名はサポートされません。たとえば、 <pre>dumpfile = /home/svtdbm6/DUMP.FILE</pre> ロード・ユーティリティでは、上のファイル名は受け入れられますが、 <pre>dumpfile = /home/svtdbm6/DUMP.LOAD.FILE</pre> このファイル名は受け入れられません。
implieddecimal	<p>暗黙指定されている小数点のロケーションが列定義によって決定され、値の終わりにあるとは見なされなくなります。たとえば、値 12345 は、12345.00 ではなく、123.45 として DECIMAL(8,2) 列にロードされます。</p> <p>この修飾子は、packeddecimal 修飾子と共に使用することはできません。</p>

表 234. 有効なファイル・タイプ修飾子 (LOAD) (続き)

修飾子	説明
timeformat="x"	<p>x はソース・ファイルの時刻の形式です。 ^a 有効な時刻エレメントは以下のとおりです。</p> <ul style="list-style-type: none"> H - 時 (12 時間制の場合は 0 ~ 12、 24 時間制の場合は 0 ~ 24 の範囲の 1 桁または 2 桁の数) HH - 時 (12 時間制の場合は 0 ~ 12、 24 時間制の場合は 0 ~ 24 の範囲の 2 桁の数。 H とは相互排他的) M - 分 (0 ~ 59 の範囲の 1 桁または 2 桁の数) MM - 分 (0 ~ 59 の範囲の 2 桁の数。 M とは相互排他的) S - 秒 (0 ~ 59 の範囲の 1 桁または 2 桁の数) SS - 秒 (0 ~ 59 の範囲の 2 桁の数。 S とは相互排他的) SSSSS - 夜の 12 時から数えた秒数 (00000 ~ 86399の範囲の 5 桁の数。 他の時刻エレメントとは相互排他的) TT - 午前午後の標識 (AM または PM) <p>デフォルト値の 0 が、指定されない各エレメントに割り当てられます。時刻形式の例を以下に示します。</p> <pre> "HH:MM:SS" "HH.MM TT" "SSSSS" </pre>

表 234. 有効なファイル・タイプ修飾子 (LOAD) (続き)

修飾子	説明
timestampformat="x"	<p>x はソース・ファイルのタイム・スタンプの形式です。 ^a 有効なタイム・スタンプ・エレメントは以下のとおりです。</p> <p>YYYY - 年 (0000 ~ 9999 の範囲の 4 桁の数字) M - 月 (1 ~ 12 の範囲の 1 桁または 2 桁の数) MM - 月 (1 ~ 12 の範囲の 2 桁の数。 M (月) とは相互排他的) D - 日 (1 ~ 31 の範囲の 1 桁または 2 桁の数) DD - 日 (1 ~ 31 の範囲の 2 桁の数。 D とは相互排他的) DDD - 日 (年間) (001 ~ 366 の範囲の 3 桁の数。 他の日または月エレメントとは相互排他的) H - 時 (12 時間制の場合は 0 ~ 12、 24 時間制の場合は 0 ~ 24 の範囲の 1 桁または 2 桁の数) HH - 時 (12 時間制の場合は 0 ~ 12、 24 時間制の場合は 0 ~ 24 の範囲の 2 桁の数。 H とは相互排他的) M - 分 (0 ~ 59 の範囲の 1 桁または 2 桁の数) MM - 分 (0 ~ 59 の範囲の 2 桁の数。 M (分) とは相互排他的) S - 秒 (0 ~ 59 の範囲の 1 桁または 2 桁の数) SS - 秒 (0 ~ 59 の範囲の 2 桁の数。 S とは相互排他的) SSSSS - 夜の 12 時から数えた秒数 (00000 ~ 86399 の範囲の 5 桁の数。 他の時刻エレメントとは相互排他的) UUUUUU - マイクロ秒 (000000 ~ 999996 の範囲の 6 桁の数) TT - 午前午後の標識 (AM または PM)</p> <p>YYYY、M、MM、D、DD、または DDD エレメントが指定されていない場合、デフォルト値の 1 が割り当てられます。他のエレメントが指定されていない場合には、デフォルト値の 0 が割り当てられます。以下に、タイム・スタンプ形式の例を示します。</p> <p>"YYYY/MM/DD HH:MM:SS.UUUUUU"</p> <p>以下の例では、ユーザー定義の日時形式を含むデータを、schedule という表にインポートする方法を示しています。</p> <pre>db2 import from delfile2 of del modified by timestampformat="yyyy.mm.dd hh:mm tt" insert into schedule</pre>
noeofchar	<p>オプションのファイル終了文字 x'1A' は、ファイルの終わりとして認識されません。それが普通の文字であるかのように処理は継続されます。</p>

表 234. 有効なファイル・タイプ修飾子 (LOAD) (続き)

修飾子	説明
usegraphiccodepage	<p>usegraphiccodepage が指定された場合、グラフィックまたは 2 バイト文字ラージ・オブジェクト (DBCLOB) データ・フィールドにロードされるデータは、グラフィック・コード・ページであると見なされます。データの残りは、文字コード・ページであると見なされます。グラフィック・コード・ページは、文字コード・ページと関連付けられます。LOAD は、指定されている場合の codepage 修飾子、または codepage 修飾子が指定されていない場合は、データベースのコード・ページを通じて、文字コード・ページを決定します。</p> <p>この修飾子は、リカバリーされている表にグラフィック・データが含まれている場合にのみ、表リカバリーのドロップによって生成された区切りデータ・ファイルとともに使用される必要があります。</p> <p>制限</p> <p>usegraphiccodepage 修飾子は、EXPORT ユーティリティで作成された DEL または ASC ファイルで指定することはできません。</p> <p>usegraphiccodepage 修飾子はまた、ファイル内の 2 バイト文字ラージ・オブジェクト (DBCLOB) には無視されます。</p>
ASC (区切りなし ASCII) ファイル形式	
binarynumerics	<p>数値 (DECIMAL 以外) データは、文字表記ではなく、バイナリー形式でなければなりません。これにより、コストのかかる変換を避けられます。</p> <p>このオプションがサポートされるのは、定位置 ASC において、recflen オプションによって固定長レコードが指定されている場合だけです。noeofchar オプションが前提です。</p> <p>以下の規則が適用されます。</p> <ul style="list-style-type: none"> • BIGINT、INTEGER、および SMALLINT を除き、データ・タイプ間の変換は実行されません。 • データ長は、それぞれのターゲット列定義と一致している必要があります。 • FLOAT は、IEEE 浮動小数点形式でなければなりません。 • ロード・ソース・ファイルに含まれるバイナリー・データは、ロード操作を実行するプラットフォームに関係なく、ビッグ・エンディアンであると見なされます。 <p>注: この修飾子の影響を受ける列のデータに、NULL を指定することはできません。この修飾子を使用すると、ブランク (通常は NULL と解釈される) は、バイナリー値であると解釈されます。</p>
nochecklengths	<p>nochecklengths が指定した場合されていると、ソース・データの列定義がターゲット表の列のサイズを超えるものであっても、各行のロードが試行されます。このような行が正常にロードされるのは、コード・ページ変換でソース・データが縮小する場合です。たとえば、ソースにある 4 バイトの EUC データがターゲットで 2 バイトの DBCS データに縮小すれば、必要スペースは半分になります。このオプションが特に役立つのは、列の定義は不一致であるがソース・データが常に適合することが分かっている場合です。</p>

表 234. 有効なファイル・タイプ修飾子 (LOAD) (続き)

修飾子	説明
nullindchar=x	<p>x は単一文字です。 NULL 値を示す文字を x に変更します。 x のデフォルト値は Y です。^b</p> <p>文字が 1 つの英字である場合を除いて、この修飾子は、EBCDIC データ・ファイルで大文字小文字を区別します。たとえば、NULL 標識文字が文字 N として指定されている場合、n も NULL 標識として認識されます。</p>
packeddecimal	<p>binarynumerics 修飾子は DECIMAL フィールド・タイプを含まないため、パック 10 進数データを直接ロードします。</p> <p>このオプションがサポートされるのは、定位置 ASC において、reclen オプションによって固定長レコードが指定されている場合だけです。noeofchar オプションが前提です。</p> <p>ニブル符号用にサポートされる値は以下のとおりです。</p> <pre> + = 0xC 0xA 0xE 0xF - = 0xD 0xB </pre> <p>この修飾子の影響を受ける列のデータに、NULL を指定することはできません。この修飾子を使用すると、ブランク (通常は NULL と解釈される) は、バイナリー値であると解釈されます。</p> <p>サーバーのプラットフォームには関係なく、ロードのソース・ファイルに含まれるバイナリー・データのバイト順はビッグ・エンディアンであることが前提となっています。つまり、この修飾子を Windows オペレーティング・システムで使用する場合も、バイト順を逆にしてはなりません。</p> <p>この修飾子は、implieddecimal 修飾子と共に使用することはできません。</p>
reclen=x	<p>x は 32 767 以下の整数です。各行ごとに x 個の文字が読み取られ、行の終わりを示すのに改行文字は使用されません。</p>
striptblanks	<p>データを可変長フィールドにロードする際に、後書きブランク・スペースを切り捨てます。このオプションを指定しない場合、ブランク・スペースはそのまま保持されます。</p> <p>このオプションは、striptnulls と一緒には指定できません。これらのオプションは、相互に排他的です。</p> <p>注: このオプションは以前の t オプションに代わるもので、バックレベルの互換性のためにのみサポートされています。</p>
striptnulls	<p>データを可変長フィールドにロードする際に、後書き NULL (0x00 文字) を切り捨てます。このオプションを指定しない場合、NULL はそのまま保持されます。</p> <p>このオプションは、striptblanks と一緒には指定できません。これらのオプションは、相互に排他的です。</p> <p>注: このオプションは以前の padwithzero オプションに代わるもので、バックレベルの互換性のためにのみサポートされています。</p>

表 234. 有効なファイル・タイプ修飾子 (LOAD) (続き)

修飾子	説明
zoneddecimal	<p>BINARYNUMERICS 修飾子は DECIMAL フィールド・タイプを含まないため、ゾーン 10 進数データをロードします。このオプションがサポートされるのは、定位置 ASC において、RECLEN オプションによって固定長レコードが指定されている場合だけです。NOEOFCHAR オプションが前提です。</p> <p>ハーフバイト符号値は、以下のいずれかです。</p> <p>+ = 0xC 0xA 0xE 0xF - = 0xD 0xB</p> <p>サポートされている数値は、0x0 ~ 0x9 です。</p> <p>サポートされているゾーン値は、0x3 および 0xF です。</p>
DEL (区切り付き ASCII) ファイル形式	
chardelx	<p>x は単一文字のストリング区切り文字です。デフォルト値は二重引用符 (") です。指定した文字は、文字ストリングを囲むために、二重引用符の代わりに使用されます。^{bc} 文字ストリング区切り文字として明示的に二重引用符 (") を指定したい場合、以下のように指定します。</p> <p style="text-align: center;">modified by charde1"</p> <p>単一引用符 (') も、以下のように文字ストリングの区切り文字として指定できます。</p> <p style="text-align: center;">modified by charde1'</p>
coldelx	<p>x は単一文字の列区切り文字です。デフォルト値はコンマ (,) です。指定した文字は、列の終わりを表すために、コンマの代わりに使用されます。^{bc}</p>
datesiso	<p>日付形式。すべての日付データ値を ISO 形式でロードします。</p>
decplusblank	<p>正符号文字。正の 10 進値の先頭に正符号 (+) ではなく、ブランク・スペースが置かれます。デフォルトのアクションでは、10 進値に正符号の接頭部が付けられます。</p>
decptx	<p>x は小数点としてピリオドと置換される単一文字です。デフォルト値はピリオド (.) です。指定した文字は、小数点文字としてピリオドの代わりに使用されます。^{bc}</p>

表 234. 有効なファイル・タイプ修飾子 (LOAD) (続き)

修飾子	説明
delprioritychar	<p>区切り文字の現在のデフォルト優先順位は、(1) レコード区切り文字、(2) 文字区切り文字、(3) 列区切り文字です。この修飾子を使用すると、区切り文字の優先順位が (1) 文字区切り文字、(2) レコード区切り文字、(3) 列区切り文字に戻り、以前の優先順位に依存している既存のアプリケーションが保護されます。構文:</p> <pre>db2 load ... modified by delprioritychar ...</pre> <p>たとえば、以下のような DEL データ・ファイルがあるとします。</p> <pre>"Smith, Joshua",4000,34.98<row delimiter> "Vincent,<row delimiter>, is a manager", 4005,44.37<row delimiter></pre> <p>delprioritychar 修飾子が指定されている場合、このデータ・ファイルには 2 行しかありません。2 番目の <row delimiter> は 2 番目の行の最初のデータ列の一部と解釈されますが、1 番目と 3 番目の <row delimiter> は実レコードの区切り文字と解釈されます。この修飾子が指定されていないと、このデータ・ファイルは 3 行になり、各行は <row delimiter> によって区切られます。</p>
dldelx	<p>x は単一文字の DATALINK 区切り文字です。デフォルト値はセミコロン (;) です。指定した文字はセミコロンの代わりに、DATALINK 値のフィールド間区切り文字として使用されます。DATALINK 値には副値が複数個含まれる場合があるため、これが必要になります。^{bcd}</p> <p>注: x は、行、列、または文字ストリング区切り文字とは異なる文字にしてください。</p>
keepblanks	<p>タイプが CHAR、VARCHAR、LONG VARCHAR、または CLOB の各フィールドの前後の空白を保持します。このオプションが指定されていない場合、文字区切り文字の外側にある前後の空白はすべて除去され、表のすべての空白・フィールドに NULL が挿入されます。</p> <p>以下の例では、データ・ファイル内の前後のスペースをすべて保持しながら、TABLE1 という表にデータをロードする方法を示しています。</p> <pre>db2 load from delfile3 of del modified by keepblanks insert into table1</pre>
nodoubledel	二重になっている区切り文字の認識を抑制します。
IXF ファイル形式	
forcein	<p>コード・ページが不一致でもデータを受け入れ、コード・ページ間の変換を抑制するようにユーティリティに指示します。</p> <p>固定長ターゲット・フィールドは、データが入るだけの十分な大きさがあるかどうかチェックされます。nochecklengths が指定されていると、チェックはされず、各行をロードしようとします。</p>
nochecklengths	nochecklengths が指定した場合されていると、ソース・データの列定義がターゲット表の列のサイズを超えるものであっても、各行のロードが試行されます。このような行が正常にロードされるのは、コード・ページ変換でソース・データが縮小する場合です。たとえば、ソースにある 4 バイトの EUC データがターゲットで 2 バイトの DBCS データに縮小すれば、必要スペースは半分になります。このオプションが特に役立つのは、列の定義は不一致であるがソース・データが常に適合することが分かっている場合です。

表 234. 有効なファイル・タイプ修飾子 (LOAD) (続き)

修飾子	説明
注:	
<p>^a 日付形式ストリングは必ず二重引用符で囲まなければなりません。フィールド区切り文字には、a ~ z, A ~ Z, および 0 ~ 9 を含めることはできません。フィールド区切り文字は、DEL ファイル形式の文字区切り文字またはフィールド区切り文字と同じにすることはできません。フィールド区切り文字は、エレメントの開始および終了位置が明確な場合は任意指定です。(修飾子によって) D, H, M, または S などのエレメントが使用される場合、項目が可変長であるためにあいまいさが存在することがあります。</p> <p>タイム・スタンプ形式の場合、月の記述子と分の記述子のどちらも文字 M を使用するため、区別が明白であるように注意する必要があります。月のフィールドは、他の日付フィールドと隣接していなければなりません。分フィールドは、他の時刻フィールドに隣接していなければなりません。以下に、いくつかのあいまいなタイム・スタンプ形式を示します。</p> <p>"M" (月または分のどちらにもとれる) "M:M" (月と分が区別がつかない) "M:YYYY:M" (両方とも月と解釈される) "S:M:YYYY" (時刻値と日付値の両方に隣接している)</p> <p>あいまいな形式になっている場合、ユーティリティはエラー・メッセージを報告し、操作は失敗します。</p> <p>以下に、明確なタイム・スタンプ形式を示します。</p> <p>"M:YYYY" (M (月)) "S:M" (M (分)) "M:YYYY:S:M" (M (月)...M (分)) "M:H:YYYY:M:D" (M (分)...M (月))</p> <p>注: 二重引用符や円記号などの文字の前には、エスケープ文字 (たとえば、¥) を付けなければなりません。</p> <p>^b この文字は、ソース・データのコード・ページで指定してください。</p> <p>文字コード・ポイント (文字記号ではない) は、xJJ または 0xJJ という構文で指定することができます (JJ はコード・ポイントの 16 進表記)。たとえば、列区切りとして # 文字を指定するには、以下のいずれかを使用します。</p> <pre>... modified by coldel# modified by colde10x23 modified by colde1X23 ...</pre> <p>^c 区切り文字の制限に、区切り文字のオーバーライドとして使用できる文字に適用される制限のリストが示されています。</p> <p>^d DATALINK 区切り文字が URL 構文内で有効な文字である場合でも、ロード操作の有効範囲内ではその特別な意味はなくなります。</p> <p>^e MODIFIED BY オプションでサポートされていないファイル・タイプが指定されても、ロード・ユーティリティが警告を発することはありません。この場合、ロード操作が失敗し、エラー・コードが戻されます。</p>	

表 235. codepage および usegraphiccodepage 使用時の LOAD 動作

codepage=N	usegraphiccodepage	LOAD 動作
なし	なし	ファイル内のすべてのデータは、データベース・コード・ページであると見なされます。

表 235. `codepage` および `usegraphiccodepage` 使用時の `LOAD` 動作 (続き)

<code>codepage=N</code>	<code>usegraphiccodepage</code>	LOAD 動作
あり	なし	<p>ファイル内のすべてのデータは、コード・ページ N であると見なされます。</p> <p>警告: N が単一バイト・コード・ページの場合、グラフィック・データをデータベースにロードすると壊れます。</p>
なし	あり	<p>ファイル内の文字データは、データベース・コード・ページであると見なされます。グラフィック・データは、データベース・グラフィック・データのコード・ページであると見なされます。</p> <p>データベース・コード・ページが単一バイトの場合は、すべてのデータはデータベース・コード・ページであると見なされます。</p> <p>警告: 単一バイト・データベースにグラフィック・データをロードすると、壊れます。</p>
あり	あり	<p>文字データは、コード・ページ N であると見なされます。グラフィック・データは、N のグラフィック・コード・ページであると見なされます。</p> <p>N が単一バイトまたは 2 バイト・コード・ページの場合は、すべてのデータは、コード・ページ N であると見なされます。</p> <p>警告: N が単一バイト・コード・ページの場合、グラフィック・データをデータベースにロードすると壊れます。</p>

関連資料:

- コマンド・リファレンス の『`QUIESCE TABLESPACES FOR TABLE` コマンド』
- データ移動ユーティリティー 手引きおよび解説書 の『パーティション・データベース・ロード構成オプション』

PRECOMPILE

組み込み SQL ステートメントの含まれているアプリケーション・プログラム・ソース・ファイルを処理します。SQL ステートメントに対するホスト言語呼び出しを含む変更後のソース・ファイルが作成されます。また、デフォルトとして、データベース内にパッケージが作成されます。

有効範囲:

このコマンドは、`db2nodes.cfg` 中のどのデータベース・パーティションからでも発行できます。パーティション・データベース環境では、これは `db2nodes.cfg` ファイル中のどのデータベース・パーティション・サーバーからでも出すことができます。実行する

と、カタログ・データベース・パーティションのデータベース・カタログが更新されます。その影響はすべてのデータベース・パーティションから見えます。

許可:

以下のいずれかが必要です。

- *sysadm* または *dbadm* 権限
- パッケージが存在しない場合は、**BINDADD** 特権および以下のどちらかが必要です。
 - パッケージのスキーマ名が存在しない場合は、データベースに対する **IMPLICIT_SCHEMA** 権限
 - パッケージのスキーマ名が存在する場合は、スキーマに対する **CREATEIN** 特権
- パッケージが存在する場合は、スキーマに対する **ALTERIN** 特権
- パッケージに対する **BIND** 特権 (パッケージが存在する場合)

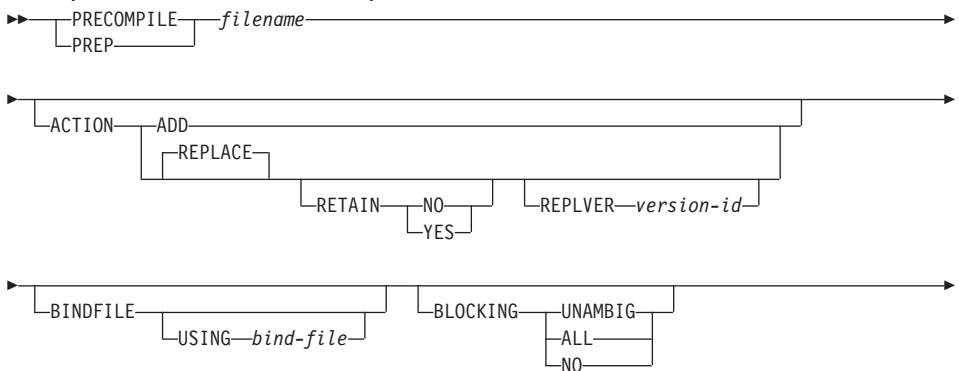
ユーザーには、アプリケーション内の静的 SQL ステートメントをコンパイルするのに必要な特権もすべて必要になります。グループに付与された特権は、静的ステートメントの許可検査では使用されません。ユーザーに *sysadm* 権限があつてバインドを完成させる明示特権がない場合、データベース・マネージャは、明示的な *dbadm* 権限を自動的に付与します。

必要な接続:

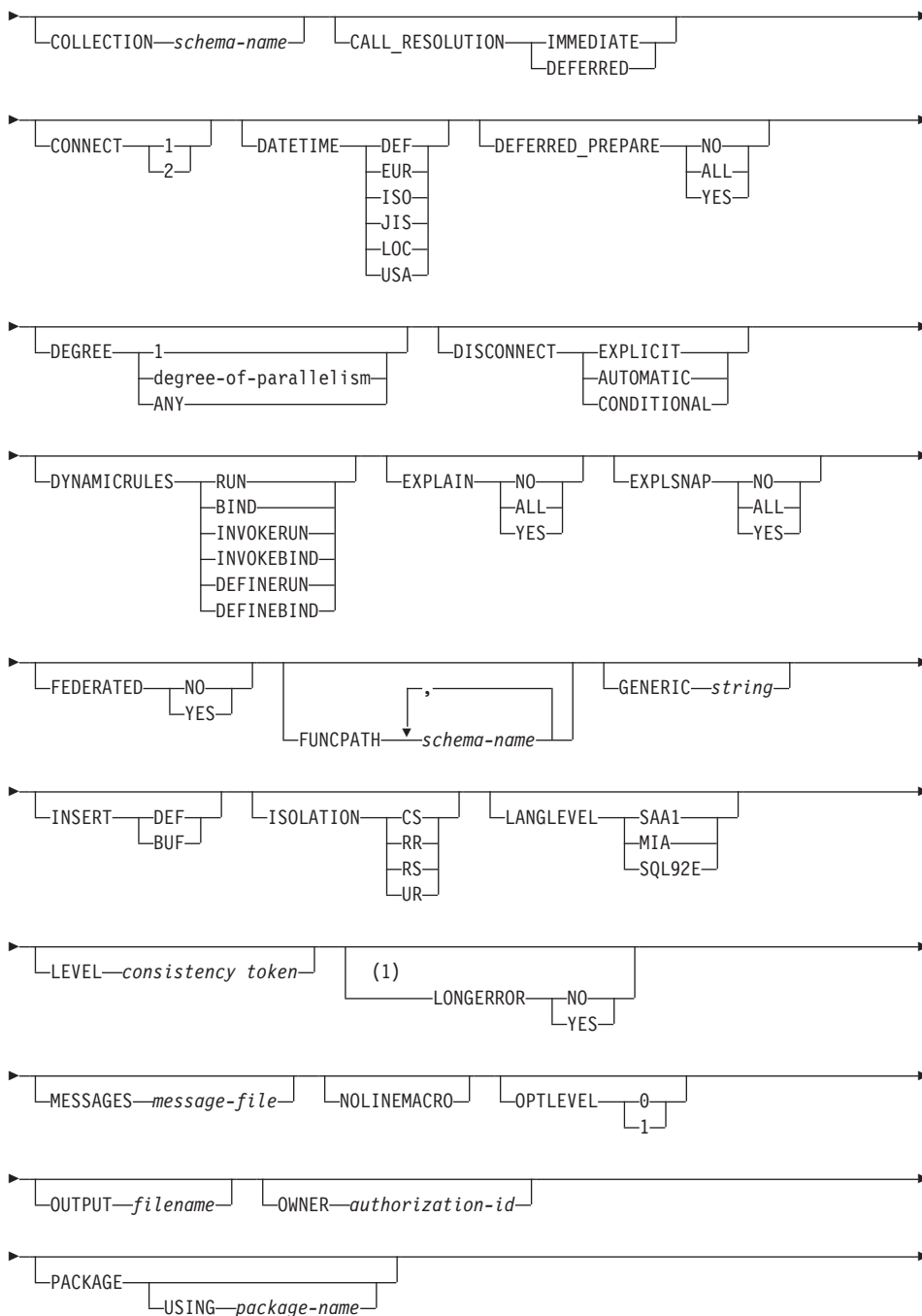
データベース。暗黙接続が可能な場合には、デフォルト・データベースへの接続が確立されます。

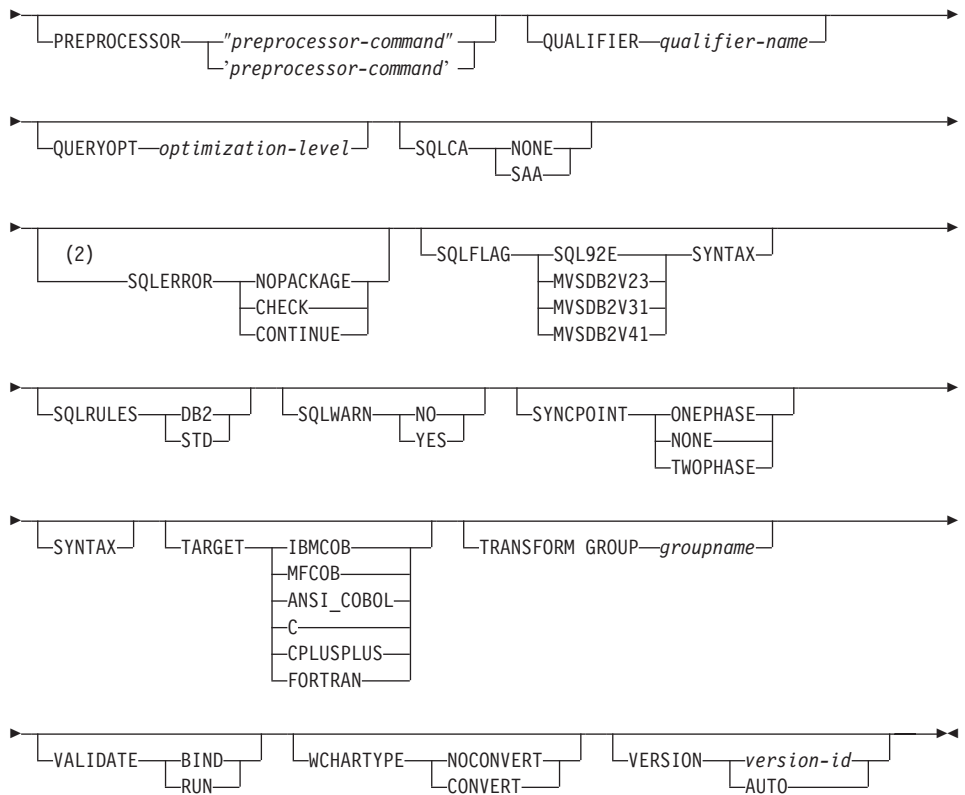
コマンド構文:

DB2 (Windows および UNIX 版) では



PRECOMPILE

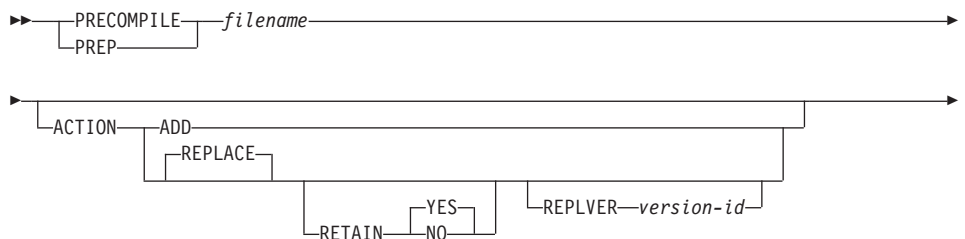




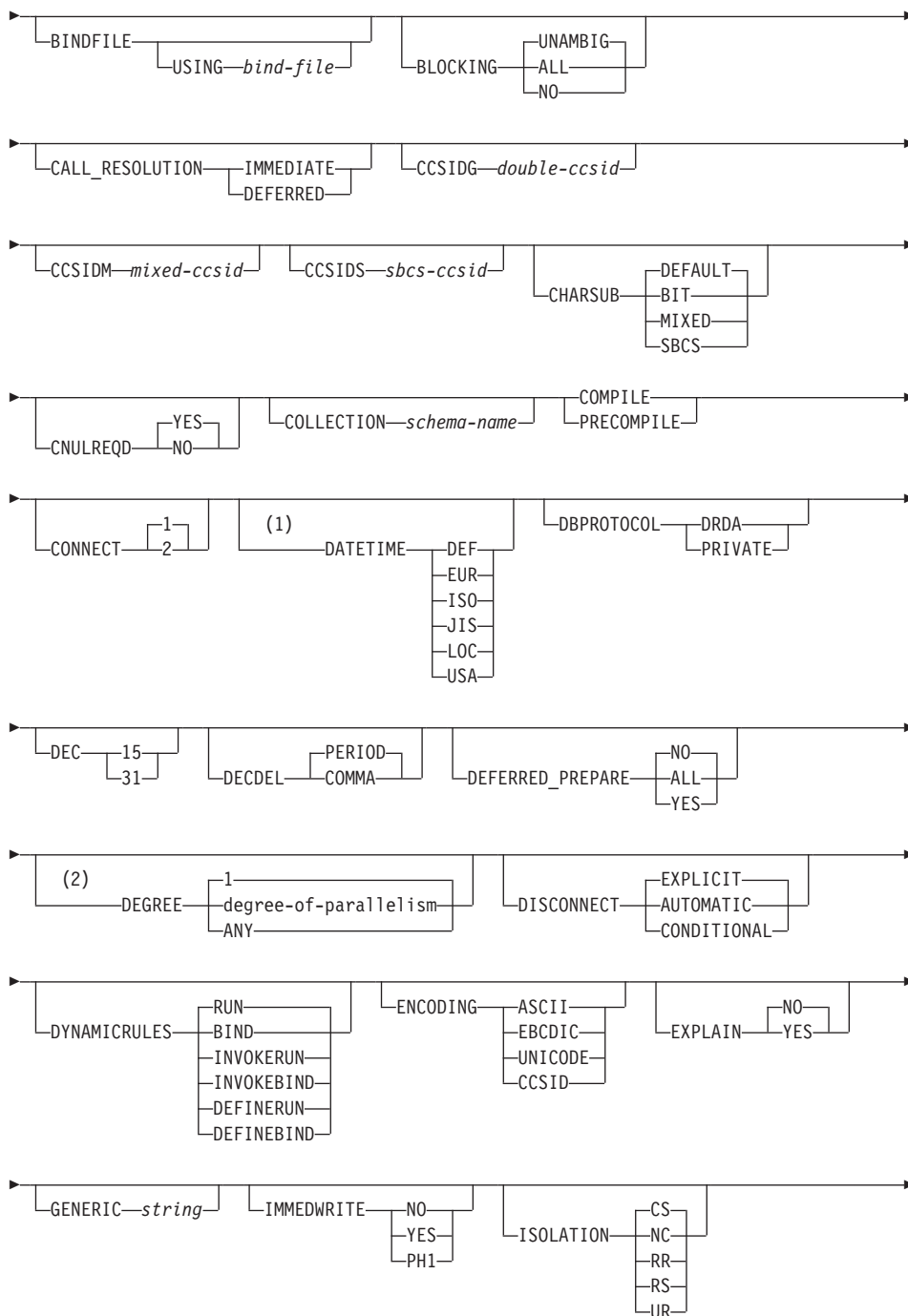
注:

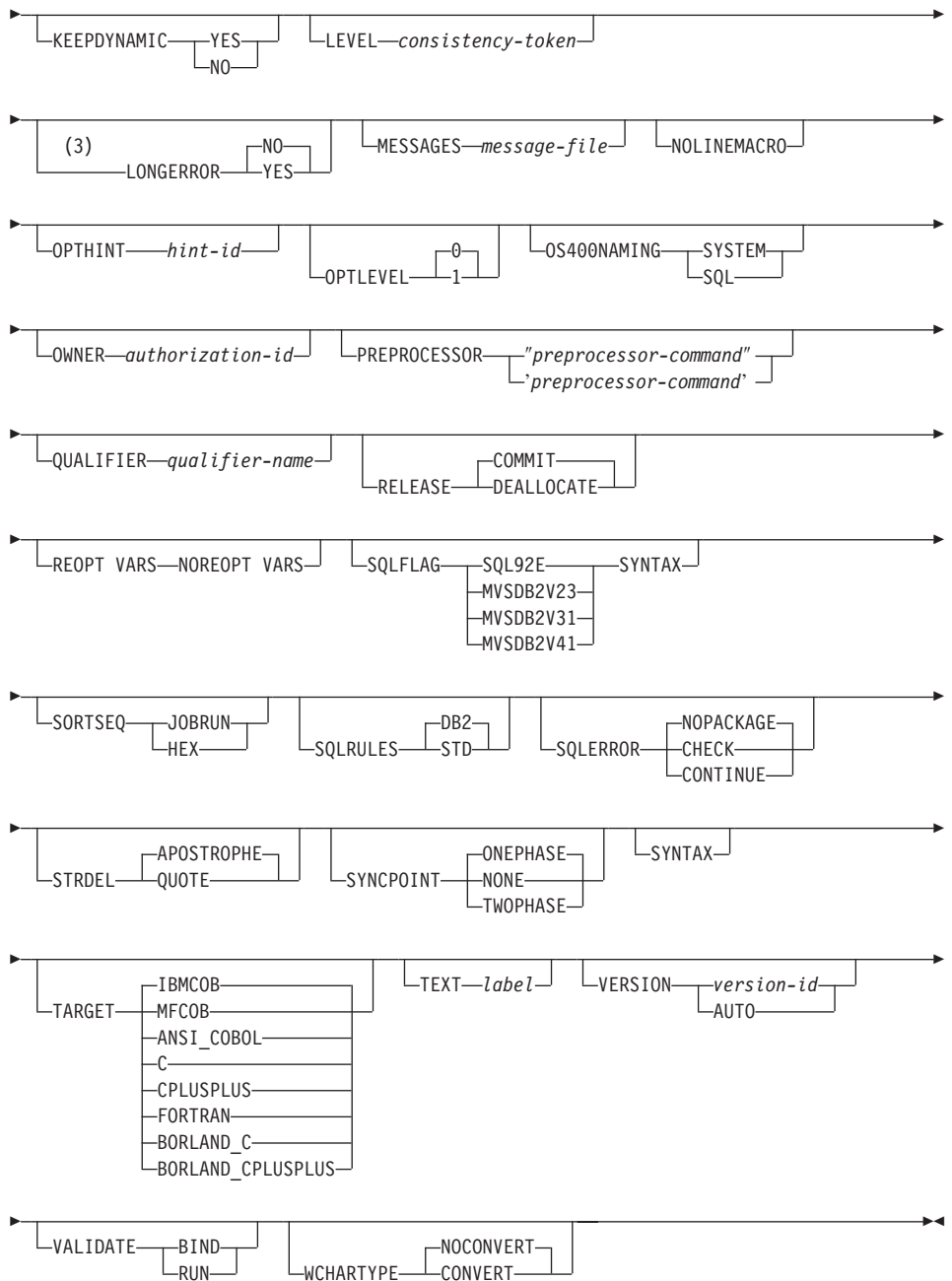
- 1 NO は、32 ビット・システムと 64 ビット NT システムのデフォルトです。それらのシステムでは、長いホスト変数を INTEGER 列の宣言として使用できません。YES は、64 ビット UNIX システムのデフォルトです。
- 2 SYNTAX は SQLERROR(CHECK) の同義語です。

Windows および UNIX 以外のサーバー上の DB2 では



PRECOMPILE





注:

- 1 サーバーが DATETIME DEF オプションをサポートしない場合、それは DATETIME ISO にマップされます。

PRECOMPILE

- 2 DEGREE オプションは DRDA レベル 2 のアプリケーション・サーバーでしかサポートされていません。
- 3 NO は、32 ビット・システムと 64 ビット NT システムのデフォルトです。これらのシステムでは、長いホスト変数を INTEGER 列の宣言として使用できません。YES は、64 ビット UNIX システムのデフォルトです。

コマンド・パラメーター:

filename

プリコンパイルするソース・ファイルを指定します。拡張子の指定は、以下のようになります。

- C アプリケーションの場合、.sqc を指定します (.c ファイルが生成されます)。
- C++ アプリケーションの場合、.sqx (Windows オペレーティング・システム)、または .sqC (UNIX 系システム) を指定します (Windows オペレーティング・システムの場合は .cxx ファイル、UNIX 系システムの場合は .C ファイルが作成されます)。
- COBOL アプリケーションの場合、.sqb を指定します (.cbl ファイルが生成されます)。
- FORTRAN アプリケーションの場合、.sqf を指定します (Windows オペレーティング・システムの場合は .for ファイル、UNIX 系システムの場合は .f ファイルが生成されます)。

UNIX 系システムにおいて、組み込み SQL を含む C++ アプリケーションの場合に望ましい拡張子は sqC です。しかし、UNIX 系システムでは、大文字小文字を区別しないシステムのための sqx 規則も通用します。

ACTION

パッケージを追加または置換できるかどうかを示します。

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

REPLACE

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

RETAIN

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

- NO** パッケージを置き換えたとき、EXECUTE 権限を保持しません。この値は DB2 ではサポートされていません。
- YES** パッケージを置き換えたとき、EXECUTE 権限を保持します。これがデフォルト値です。

REPLVER version-id

特定のバージョンのパッケージを置き換えます。バージョン ID は、どのバージョンのパッケージを置き換えるのかを指定するものです。指定されたバージョンが存在しない場合には、エラーが戻されます。REPLACE の REPLVER オプションが指定されていない場合、プリコンパイルされるパッケージのパッケージ名、およびバージョンと一致するパッケージがすでに存在すれば、そのパッケージは置換されます。存在しなければ、新規のパッケージが追加されます。

BINDFILE

バインド・ファイルが作成されます。 **package** オプションが共に指定されていない場合、パッケージは作成されません。次に示す例のように、バインド・ファイルを要求したのにパッケージが作成されないなら、

```
db2 prep sample.sqc bindfile
```

オブジェクトの存在と認証 SQLCODE はエラーではなく警告として扱われます。それで、プリコンパイルに使用されるデータベースに、アプリケーション内の静的 SQL ステートメントで参照されているすべてのオブジェクトが入っているわけではない場合でも、バインド・ファイルは正常に作成されます。必要なオブジェクトが作成されたなら、バインド・ファイルは正常にバインドされ、パッケージが作成されます。

USING bind-file

プリコンパイラーが生成するバインド・ファイルの名前。ファイル名には、.bnd 拡張子が付いていなければなりません。ファイル名を入力しないなら、プリコンパイラーは、 *filename* パラメーターとして入力されているプログラムの名前を使用し、それに .bnd 拡張子を付けてファイル名とします。パスを指定しないと、バインド・ファイルは現行ディレクトリーに作成されます。

BLOCKING

行のブロッキングについては、管理の手引きを参照してください。

ALL 以下のカーソルをブロック化することを指定します。

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

未確定のカーソルは、読み取り専用として扱われます。

PRECOMPILE

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

UNAMBIG

以下のカーソルをブロック化することを指定します。

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

未確定のカーソルは、更新可能として扱われます。

CALL_RESOLUTION

設定されている場合、CALL_RESOLUTION DEFERRED オプションは使用すべきでない `sqlproc()` API の起動として、CALL ステートメントが実行されることを示します。設定されていないか、または IMMEDIATE が設定されている場合、CALL ステートメントは通常の SQL ステートメントとして実行されます。プリコンパイラーが CALL_RESOLUTION IMMEDIATE を指定した CALL ステートメントのプロシーチャーを解決できなかった場合に、SQL0204 が出されることに注意してください。

CCSIDG double-ccsid

CREATE および ALTER TABLE SQL ステートメントの文字カラム定義で、2 バイト文字用のコード化文字セット ID (CCSID) (特定の CCSID 文節は使用しない) を指定する整数。なお、DB2 はこの DRDA プリコンパイル/バインド・オプションをサポートしません。このオプションを指定しないと、DRDA サーバーは、システムが定義したデフォルトを使用します。

CCSIDM mixed-ccsid

CREATE および ALTER TABLE SQL ステートメントの文字カラム定義で、混合バイト文字用のコード化文字セット ID (CCSID) (特定の CCSID 文節は使用しない) を指定する整数。なお、DB2 はこの DRDA プリコンパイル/バインド・オプションをサポートしません。このオプションを指定しないと、DRDA サーバーは、システムが定義したデフォルトを使用します。

CCSIDS sbcs-ccsid

CREATE および ALTER TABLE SQL ステートメントの文字カラム定義で、1 バイト文字用のコード化文字セット ID (CCSID) (特定の CCSID 文節は使用しない) を指定する整数。なお、DB2 はこの DRDA プリコンパイル/バインド・オプションをサポートしません。このオプションを指定しないと、DRDA サーバーは、システムが定義したデフォルトを使用します。

CHARSUB

CREATE および ALTER TABLE SQL ステートメントの列定義に使用する、デフォルトの文字サブタイプを指定します。なお、DB2 はこの DRDA プリコンパイル/バインド・オプションをサポートしません。

BIT 明示的にサブタイプを指定しなかった場合、すべての新規文字カラムに FOR BIT DATA SQL 文字サブタイプが使用されます。

DEFAULT

明示的にサブタイプを指定しなかった場合、すべての新規文字カラムにターゲット・システムが定義したデフォルト・サブタイプが使用されます。

MIXED 明示的にサブタイプを指定しなかった場合、すべての新規文字カラムに FOR MIXED DATA SQL 文字サブタイプが使用されます。

SBCS 明示的にサブタイプを指定しなかった場合、すべての新規文字カラムに FOR SBCS DATA SQL 文字サブタイプが使用されます。

CNULREQD

このオプションは DRDA でサポートされていない **langlevel** プリコンパイル・オプションと関連します。これは、C または C++ アプリケーションで作成されたバインド・ファイルの場合のみ有効です。この DRDA バインド・オプションは、DB2 ではサポートされていません。

NO C ストリング・ホスト変数中の NULL 終止符に関して、**langlevel SAA1** プリコンパイル・オプションに基づいてアプリケーションがコード化された場合です。

YES C ストリング・ホスト変数中の NULL 終止符に関して、**langlevel MIA** プリコンパイル・オプションに基づいてアプリケーションがコード化された場合です。

COLLECTION schema-name

パッケージ用の 30 文字の収集 ID を指定します。これを指定しなかった場合、パッケージを処理する際には、ユーザーの許可 ID が使用されます。

CONNECT

1 CONNECT ステートメントをタイプ 1 の CONNECT として処理するよう指定します。

2 CONNECT ステートメントをタイプ 2 の CONNECT として処理するよう指定します。

DATETIME

使用する日時形式を指定します。

DEF データベースのテリトリー・コードと対応する日時形式を使用します。

EUR IBM 欧州規格の日時形式を使用します。

ISO 国際標準化機構規格の日時形式を使用します。

JIS 日本工業規格の日時形式を使用します。

LOC データベースのテリトリー・コードと対応する地域別日時形式を使用します。

PRECOMPILE

USA IBM 米国規格の日時形式を使用します。

DBPROTOCOL

3 パート名ステートメントによって識別されるリモート・サイトに接続するとき使用するプロトコルを指定します。サポートしているのは DB2 for OS/390 だけです。サポートされているオプション値のリストについては、DB2 for OS/390 のドキュメンテーションを参照してください。

DEC 10 進算術演算に使用する最大精度を指定します。なお、DB2 はこの DRDA プリコンパイル/バインド・オプションをサポートしません。このオプションを指定しないと、DRDA サーバーは、システムが定義したデフォルトを使用します。

15 10 進算術演算に 15 桁精度が使用されます。

31 10 進算術演算に 31 桁精度が使用されます。

DECDEL

10 進数および浮動小数点リテラル中で小数点標識としてピリオド (.) またはコンマ (,) のどちらかを指定します。なお、DB2 はこの DRDA プリコンパイル/バインド・オプションをサポートしません。このオプションを指定しないと、DRDA サーバーは、システムが定義したデフォルトを使用します。

COMMA

小数点標識としてコンマ (,) を使用します。

PERIOD

小数点標識としてピリオド (.) を使用します。

DEFERRED_PREPARE

DB2 共通サーバー・データベースまたは DRDA データベースにアクセスする際のパフォーマンス機能を強化します。このオプションは、SQL PREPARE ステートメントを、それに関連した OPEN、 DESCRIBE、または EXECUTE ステートメント・フローと結合して、プロセス間またはネットワーク・フローを最小にします。

NO このコマンドが実行されると同時に PREPARE ステートメントが実行されます。

YES PREPARE ステートメントの実行は、対応する OPEN、 DESCRIBE、または EXECUTE オープンが発行されるまで据え置かれます。

SQLDA を即時に戻す必要がある INTO 文節を使用している場合、PREPARE ステートメントは据え置かれません。しかし、パラメーター・マーカを使用しないカーソルに対して PREPARE INTO ステートメントが発行される場合、その処理は PREPARE の実行時にカーソルを事前 OPEN することによって最適化されます。

ALL PREPARE INTO ステートメントも据え置かれること以外は、YES と同じです。PREPARE ステートメントで SQLDA を戻すために

INTO 文節を使用している場合、OPEN、DESCRIBE、または EXECUTE ステートメントが発行されて戻されるまで、アプリケーションでこの SQLDA の内容を参照してはなりません。

DEGREE

SMP システムで静的 SQL ステートメントを実行するための並列処理の度合いを指定します。このオプションは、CREATE INDEX 並列処理には影響を与えません。

1 ステートメントの実行に並列処理を使用しません。

degree-of-parallelism

ステートメントを実行する際の並列処理の度合いを指定します。値の範囲は 2 ~ 32767 です。

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

DISCONNECT

AUTOMATIC

コミット時にすべてのデータベース接続を切断するよう指定します。

CONDITIONAL

RELEASE をマークしたか、またはオープン状態の WITH HOLD カーソルをもたないデータベース接続を、コミット時に切断するよう指定します。

EXPLICIT

RELEASE ステートメントで明示的に解放をマークしたデータベース接続だけを、コミット時に切断するよう指定します。

DYNAMICRULES

許可 ID に使用される値の初期設定、および未修飾オブジェクト参照の暗黙的な修飾の、ランタイムの動的 SQL に適用される規則を定義します。

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

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

DEFINERUN

パッケージがルーチン・コンテキスト内で使用される場合、ルーチン

定義者の許可 ID が許可検査およびルーチン内の動的 SQL ステートメント内の未修飾オブジェクト参照の暗黙的な修飾に使用されます。

パッケージがスタンドアロン・アプリケーションとして使用される場合、動的 SQL ステートメントはパッケージが DYNAMICRULES RUN とバインドしたかのように処理されます。

DEFINEBIND

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

パッケージがスタンドアロン・アプリケーションとして使用される場合、動的 SQL ステートメントはパッケージが DYNAMICRULES BIND とバインドしたかのように処理されます。

INVOKERUN

パッケージがルーチン・コンテキスト内で使用される場合、ルーチン起動時に有効だった現行のステートメント許可 ID が、動的 SQL ステートメントの許可検査およびそのルーチン内の動的 SQL ステートメント内の未修飾オブジェクト参照の暗黙的な修飾に使用されます。

パッケージがスタンドアロン・アプリケーションとして使用される場合、動的 SQL ステートメントはパッケージが DYNAMICRULES RUN とバインドしたかのように処理されます。

INVOKEBIND

パッケージがルーチン・コンテキスト内で使用される場合、ルーチン起動時に有効だった現行のステートメント許可 ID が、動的 SQL ステートメントの許可検査およびそのルーチン内の動的 SQL ステートメント内の未修飾オブジェクト参照の暗黙的な修飾に使用されます。

パッケージがスタンドアロン・アプリケーションとして使用される場合、動的 SQL ステートメントはパッケージが DYNAMICRULES BIND とバインドしたかのように処理されます。

注: 動的 SQL ステートメントは、バインド動作を公開しているパッケージ内のパッケージ所有者の許可 ID を使用します。したがって、パッケージのユーザーが受け取るべきでない権限を、パッケージのバインダーに付与してはなりません。同様に、定義動作を公開するルーチンを定義するとき、パッケージのユーザーが受け取るべきでない権限を、ルーチンの定義者に付与してはなりません。動的ステートメントがルーチンの定義者の許可 ID を使用するためです。パッケージ動作についての詳細は、アプリケーション開発ガイドの「How DYNAMICRULES affects the behavior of dynamic SQL statements」のセクションを参照してください。

以下の動的な準備済み SQL ステートメントは、DYNAMICRULES RUN にバインドされなかったパッケージ内では使用できません。GRANT、REVOKE、ALTER、CREATE、DROP、COMMENT ON、RENAME、SET INTEGRITY、および SET EVENT MONITOR STATE です。

ENCODING

プランまたはパッケージ内の静的ステートメント内にあるすべてのホスト変数のエンコード方式を指定します。サポートしているのは DB2 for OS/390 だけです。サポートされているオプション値のリストについては、DB2 for OS/390 のドキュメンテーションを参照してください。

EXPLAIN

各 SQL ステートメント用に選択したアクセス・プランに関する Explain 表の情報を、パッケージに保管します。DRDA では、このオプションの ALL 値がサポートされていません。

NO Explain 情報はキャプチャーされません。

YES Explain 表には、静的ステートメントの場合は prep/bind 時間で、増分バインド・ステートメントの場合はランタイムで、選択されたアクセス・プランについての情報が取り込まれます。

パッケージがルーチンに使用されるもので、パッケージに追加バインド・ステートメントが含まれる場合、そのルーチンは MODIFIES SQL DATA として定義されなければなりません。これが行われない場合、パッケージ内の追加バインド・ステートメントはランタイム・エラーを生じます (SQLSTATE 42985)。

ALL 適格な静的 SQL ステートメントの Explain 情報が、prep/bind 時間で各 Explain 表に入れられます。適格な増分バインド SQL ステートメントの Explain 情報が、ランタイムで各 Explain 表に入れられます。さらに、CURRENT EXPLAIN SNAPSHOT レジスターが NO に設定されていても、Explain 情報はランタイムに適格な動的 SQL ステートメント用に集められます。

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

注: DRDA では、EXPLAIN のこの値 (ALL) はサポートされていません。

EXPLSNAP

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

NO Explain スナップショットはキャプチャーされません。

PRECOMPILE

YES Explain 表には、静的ステートメントの場合は prep/bind 時間で、増分バインド・ステートメントの場合はランタイムで、適格な各静的 SQL ステートメントの Explain スナップショットが、Explain 表内に入れます。

パッケージがルーチンに使用されるもので、パッケージに追加バインド・ステートメントが含まれる場合、そのルーチンは MODIFIES SQL DATA として定義されなければなりません。これが行われない場合、パッケージ内の追加バインド・ステートメントはランタイム・エラーを生じます (SQLSTATE 42985)。

ALL 適格な各静的 SQL ステートメントの Explain スナップショットが、prep/bind 時間で Explain 表内に入れます。適格な増分バインド SQL ステートメントの Explain スナップショット情報が、ランタイムで各 Explain 表に入れます。さらに、CURRENT EXPLAIN SNAPSHOT レジスターが NO に設定されていても、Explain スナップショット情報はランタイムに適格な動的 SQL ステートメント用に集められます。

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

FEDERATED

パッケージ内の静的 SQL ステートメントがニックネームまたは統合されたビューを参照するかどうかを指定します。このオプションが指定されず、パッケージ内の静的 SQL ステートメントがニックネームまたは統合されたビューを参照する場合は、警告が返され、パッケージは作成されません。

注: このオプションは DRDA サーバーではサポートされていません。

NO ニックネームまたは統合されたビューは、パッケージ内の静的 SQL ステートメントで参照されません。ニックネームまたは統合されたビューがこのパッケージの準備またはバインド・フェーズ中に静的 SQL ステートメントで見つかった場合、エラーが返され、パッケージは作成されません。

YES ニックネームまたは統合されたビューは、パッケージ内の静的 SQL ステートメントで参照が可能です。ニックネームまたは統合されたビューがこのパッケージの準備またはバインド中に静的 SQL ステートメントで見つからなかった場合、エラーまたは警告は返されず、パッケージは作成されます。

FUNCPATH

静的 SQL で、ユーザー定義の個別タイプおよび機能を解析する際に使用する機能パスを指定します。このオプションを指定しなかった場合、デフォルトの

機能パスは "SYSIBM"、"SYSFUN"、または USER になります。この DB2 プリコンパイル/バインド・オプションは、DRDA ではサポートされていません。

schema-name

SQL ID (通常または区切り)。これは、アプリケーション・サーバーに存在するスキーマを識別します。スキーマが存在する場合、プリコンパイル時やバインド時に妥当性検査は行われません。同一スキーマは、機能パス内に一度しか存在できません。指定できるスキーマ数は、処理結果の機能パスの長さによって限定され、254 バイトを超えることはできません。スキーマ SYSIBM は、明示的に指定する必要がありません。機能パス内に含まれていなければ、最初のスキーマに暗黙的に想定されます。

INSERT

DB2 Enterprise - Extended Edition サーバーへプリコンパイルまたはバインドされているプログラムが、パフォーマンス向上のために挿入データをバッファリングすることを要求できるようにします。

注: このオプションは DRDA サーバーではサポートされていません。

BUF アプリケーションからの挿入データをバッファリングすることを指定します。

DEF アプリケーションからの挿入データをバッファリングしないことを指定します。

GENERIC string

ターゲット・データベースで定義されていても、DRDA でサポートされていない新規バインド・オプションをサポートします。BINDまたはPRECOMPILE で定義されているバインド・オプションを渡すようにするには、このオプションを使用しないでください。このオプションは、動的 SQL のパフォーマンスをかなり向上させることができます。構文は以下のとおりです。

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

各オプションと値は、1 つ以上のブランク・スペースで区切らなければなりません。たとえば、ターゲット DRDA データベースが DB2 Universal Database バージョン 8 の場合、以下のようにします。

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

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

ストリングの最大長は 1023 バイトです。

PRECOMPILE

IMMEDWRITE

グループ・バッファ・プールに依存するページセットまたはパーティションに対する更新について、即時書き込みを行うかどうかを示します。サポートしているのは DB2 for OS/390 だけです。サポートされているオプション値のリストについては、DB2 for OS/390 のドキュメンテーションを参照してください。

ISOLATION

このパッケージにバインドされるプログラムを、他の実行プログラムの影響からどの程度分離できるかを指定します。

CS カーソル固定を分離レベルとして指定します。

NC コミットしません。コミット制御が使用されないということを指定します。なお、DB2 はこの分離レベルをサポートしません。

RR 反復可能読み取りを分離レベルとして指定します。

RS 読み取り固定を分離レベルとして指定します。読み取り固定は、パッケージ内での SQL ステートメントの実行を、他のアプリケーションが読み取りおよび変更を行った行に対する処理から分離させます。

UR 非コミット読み取りを分離レベルとして指定します。

LANGLEVEL

アプリケーション内の静的および動的 SQL の構文およびセマンティクスに適用される SQL 規則を指定します。このオプションは DRDA サーバーではサポートされていません。

MIA 以下のように ISO/ANS SQL92 規則を選択します。

- エラー SQLCODE または SQLSTATE のチェックをサポートするには、アプリケーション・コードの中で SQLCA が宣言されていなければなりません。
- C の NULL 終了ストリングにはブランクが埋め込まれ、切り捨てが実行された場合でも常に NULL 終了文字が含まれます。
- FOR UPDATE 文節は、定位置 UPDATE で更新されるすべての列において任意指定です。
- UPDATE または DELETE ステートメントの対象となる表の列が、検索条件の中や代入文節の右辺で参照されているなら、検索 UPDATE または DELETE に、対象となる表に対する SELECT 特権が必要です。
- 索引を使って解決可能な列関数 (たとえば MIN または MAX) は、NULL もチェックし、NULL があったなら警告 SQLSTATE 01003 を戻します。
- CREATE または ALTER TABLE ステートメントの中に重複したユニーク制約が含まれているなら、エラーが戻されます。

- 特権が何も付与されていない場合、付与者にそのオブジェクトに対する権限がないなら、エラーが戻されます (その権限があるなら警告が戻されます)。

SAA1 以下のように共通 IBM DB2 規則を選択します。

- エラー `SQLCODE` または `SQLSTATE` のチェックをサポートするには、アプリケーション・コードの中で `SQLCA` が宣言されていないければなりません。
- `C` の `NULL` 終了ストリング、切り捨てが実行された場合には、`NULL` 文字が最後に付けられません。
- `FOR UPDATE` 文節は、定位置 `UPDATE` で更新されるすべての列において必須です。
- `UPDATE` または `DELETE` ステートメントの対象となる表が、そのステートメントの全選択から参照されるのであれば、検索 `UPDATE` または `DELETE` に、対象となる表に対する `SELECT` 特権は不要です。
- 索引を使って解決可能な列関数 (たとえば `MIN` または `MAX`) は、`NULL` をチェックせず、警告 `SQLSTATE 01003` は戻されません。
- 警告が戻され、ユニーク制約が重複していても無視されます。
- 特権が付与されていないなら、エラーが戻されます。

SQL92E

以下のように ISO/ANS SQL92 規則を定義します。

- `SQLCODE` または `SQLSTATE` のチェックをサポートするには、その名前の変数をホスト変数の宣言セクションで宣言できます (どちらも宣言されていないなら、プリコンパイル中には `SQLCODE` が指定されているものとされます)。
- `C` の `NULL` 終了ストリングにはブランクが埋め込まれ、切り捨てが実行された場合でも常に `NULL` 終了文字が含まれます。
- `FOR UPDATE` 文節は、定位置 `UPDATE` で更新されるすべての列において任意指定です。
- `UPDATE` または `DELETE` ステートメントの対象となる表の列が、検索条件の中や代入文節の右辺で参照されているなら、検索 `UPDATE` または `DELETE` に、対象となる表に対する `SELECT` 特権が必要です。
- 索引を使って解決可能な列関数 (たとえば `MIN` または `MAX`) は、`NULL` もチェックし、`NULL` があったなら警告 `SQLSTATE 01003` を戻します。
- `CREATE` または `ALTER TABLE` ステートメントの中に重複したユニーク制約が含まれているなら、エラーが戻されます。

PRECOMPILE

- 特権が何も付与されていない場合、付与者にそのオブジェクトに対する権限がないなら、エラーが戻されます (その権限があるなら警告が戻されます)。

KEEPDYNAMIC

コミット・ポイントの後で動的 SQL ステートメントを保持するかどうかを指定します。サポートしているのは DB2 for OS/390 だけです。サポートされているオプション値のリストについては、DB2 for OS/390 のドキュメンテーションを参照してください。

LEVEL consistency-token

一貫性トークンを使用するモジュールのレベルを定義します。一貫性トークンとは、8 文字までの長さの任意の英数字値のことです。RDB パッケージの一貫性トークンは、リクエストのアプリケーションとリレーショナル・データベース・パッケージが同期化されているかどうかを検証します。

注: このオプションは、通常は使用しないでください。

LONGERROR

長いホスト変数宣言をエラーとして扱うかどうかを示します。移植性のために、sqlint32 は、プリコンパイル C および C++ コードで INTEGER 列の宣言として使用できます。

NO 長いホスト変数宣言の使用に対してエラーを生成しません。これが 32 ビット・システムと 64 ビット NT システムのデフォルトです。それらのシステムでは、長いホスト変数を INTEGER 列の宣言として使用できます。このオプションを 64 ビット UNIX プラットフォームで使用すると、長いホスト変数を、BIGINT 列の宣言として使用することができます。

YES 長いホスト変数宣言の使用に対してエラーを生成します。これが 64 ビット UNIX システムのデフォルトです。

MESSAGES message-file

警告メッセージ、エラー・メッセージ、および完了状況メッセージの宛先を指定します。メッセージ・ファイルは、バインドが正常であるかどうかによって作成されます。メッセージ・ファイル名を指定しなかった場合、メッセージは標準出力に書き込まれます。ファイルへの完全パスを指定しなかった場合は、現行ディレクトリが使用されます。なお、既存ファイルの名前を指定すると、そのファイルの内容は上書きされます。

NOLINEMACRO

出力 .c ファイルでの # 行マクロの生成を抑制します。これは、プロファイル、相互参照ユーティリティ、およびデバッガーのようなソース行情報を必要とする作成ツールでファイルを使用する場合に役に立ちます。

注: このプリコンパイル・オプションは、C/C++ プログラム言語でのみ使用します。

OPTHINT

照会最適化ヒントを静的 SQL に使用するかどうかを制御します。サポートしているのは DB2 for OS/390 だけです。サポートされているオプション値のリストについては、DB2 for OS/390 のドキュメンテーションを参照してください。

OPTLEVEL

SQL ステートメントの中でホスト変数が使用されている場合に、内部 SQLDA の初期化を C/C++ プリコンパイラーで最適化するかどうかを指示します。そのように最適化すれば、密なループ内で単一の SQL ステートメント (FETCH など) を使う場合のパフォーマンスが向上します。

- 0 SQLDA の初期化を最適化しないようプリコンパイラーに指示します。
- 1 SQLDA の初期化を最適化するようプリコンパイラーに指示します。アプリケーションで以下のものを使っている場合は、この値を指定しないでください。

- 以下の例に示すようなポインター・ホスト変数

```
exec sql begin declare section;
char (*name)[20];
short *id;
exec sql end declare section;
```

- 直接 SQL ステートメントに指定される C++ データ・メンバー

OUTPUT filename

コンパイラーが生成した修正済みソース・ファイルのデフォルト名をオーバーライドします。この名前にはパスを含めることもできます。

OS400NAMING

DB2 UDB for iSeries データにアクセスする際に使用する命名オプションを指定します。DB2 UDB for iSeries だけによってサポートされています。サポートされているオプション値のリストについては、DB2 for iSeries のドキュメンテーションを参照してください。

DB2 ユーティリティーが OS400NAMING SYSTEM オプションを指定してプリコンパイルまたはバインドされていても、区切り記号としてスラッシュが使用されているために、ユーティリティーは iSeries システムの命名規則を使用する特定の SQL ステートメントに関して実行時に構文エラーを報告することがありますので注意してください。たとえば、OS400NAMING SYSTEM オプションを指定してプリコンパイルまたはバインドされているかどうかには関係なく、iSeries システムの命名規則が使用されている場合、コマンド行プロセッサは SQL CALL ステートメントに関して構文エラーを報告します。

PRECOMPILE

OWNER authorization-id

パッケージ所有者の 30 文字の許可 ID を指定します。その所有者には、パッケージに含まれる SQL ステートメントを実行するための特権が必要です。

SYSADM または DBADM 許可を持つユーザーのみが、ユーザー ID 以外の許可 ID を指定できます。デフォルトは、プリコンパイル/バインド処理の 1 次許可 ID です。SYSIBM、SYSCAT、および SYSSTAT はこのオプションには無効な値です。

PACKAGE

パッケージを作成します。**package**、**bindfile**、または **syntax** のどれも指定されていない場合は、デフォルトで、データベースの中にパッケージが作成されます。

USING package-name

プリコンパイラーが生成するパッケージの名前。名前を入力しないなら、アプリケーション・プログラムのソース・ファイルの名前 (拡張子を取り去って大文字に変換したもの) が使われます。最大長は 8 文字です。

PREPROCESSOR "preprocessor-command"

組み込み SQL ステートメントを処理する前に、プリコンパイラーが実行できるプリプロセッサ・コマンドを指定します。プリプロセッサ・コマンド・ストリング (最大 1024 バイト) は、二重引用符または単一引用符で囲む必要があります。

このオプションは、宣言セクションでマクロを使用できるようにします。有効なプリプロセッサ・コマンドとは、コマンド行から発行でき、ソース・ファイルを指定しなくてもプリプロセッサを呼び出せるコマンドです。たとえば、

```
x1c -P -DMYMACRO=0
```

QUALIFIER qualifier-name

パッケージに含まれる未修飾オブジェクトの 30 文字の暗黙修飾子を指定します。**owner** が明示的に指定されていれば、その所有者の許可 ID がデフォルト ID になります。

QUERYOPT optimization-level

パッケージに含まれるすべての静的 SQL ステートメントに必要な最適化レベルを指示します。デフォルト値は 5 です。使用可能な最適化レベルの範囲の詳細については、*SQL* リファレンスの SET CURRENT QUERY OPTIMIZATION ステートメントを参照してください。この DB2 プリコンパイル/バインド・オプションは、DRDA ではサポートされていません。

RELEASE

リソースを、各 COMMIT ポイントで解放するか、アプリケーションの終了時

に解放するかどうかを指示します。なお、DB2 はこの DRDA プリコンパイル/バインド・オプションをサポートしません。

COMMIT

各コミット点でリソースを解放します。これは、動的 SQL ステートメントに使用されます。

DEALLOCATE

アプリケーションの終了時にだけリソースを解放します。

SQLCA

FORTRAN アプリケーションでのみ使用します。その他の言語で使用しても、このオプションは無視されます。

NONE 修正されたソース・コードは SAA 定義と整合性がないことを指定します。

SAA 修正されたソース・コードは SAA 定義と整合性があることを指定します。

SQLERROR

エラーを検出した場合に、パッケージまたはバインド・ファイルを作成するかどうかを指示します。

CHECK

ターゲット・システムが、バインドされている SQL ステートメントのすべての構文、およびセマンティックのチェックを行うことを指定します。この処理の一部としてパッケージが作成されることはありません。バインド中に、名前とバージョンが同じ既存パッケージを検出した場合、その既存パッケージはドロップも置換 (**ACTION REPLACE** を指定した場合) もされません。

CONTINUE

SQL ステートメントのバインド時にエラーが発生しても、パッケージを作成します。許可または存在などの理由でバインドに失敗したこれらのステートメントは、**VALIDATE RUN** も指定されている場合は、実行時に増分でバインドすることができます。ランタイムでこれらのステートメントを実行しようとする、エラー (SQLCODE -525, SQLSTATE 51015) が生成されます。

NOPACKAGE

エラーを検出した場合、パッケージもバインド・ファイルも作成しません。

REOPT / NOREOPT VARS

DB2 がホスト変数、パラメーター・マーカ、および特殊レジスターの値を使用して実行時にアクセス・パスを判別するようにするかどうかを指定します。

PRECOMPILE

サポートしているのは DB2 for OS/390 だけです。サポートされているオプション値のリストについては、DB2 for OS/390 のドキュメンテーションを参照してください。

SQLFLAG

このオプションに指定された SQL 言語構文との相違点を識別して報告します。

バインド・ファイルまたはパッケージが作成されるのは、**sqlflag** オプションに加えて **bindfile** または **package** オプションが指定されている場合だけです。

ローカル構文検査が実行されるのは、次に示すオプションのうちの 1 つが指定された場合だけです。

- **bindfile**
- **package**
- **sqlerror check**
- **syntax**

sqlflag が指定されていない場合、**flagger** は呼び出されず、バインド・ファイルやパッケージが影響を受けることはありません。

SQL92E SYNTAX

SQL ステートメントは、データベース・カタログにアクセスするのに必要な構文規則を除いて、ANSI または ISO SQL92 のエンタリー・レベル SQL 言語の書式および構文と比較して調べられます。相違があれば、プリコンパイラー・リストに報告されます。

MVSDB2V23 SYNTAX

SQL ステートメントは、MVS DB2 バージョン 2.3 SQL 言語構文と比較して調べられます。構文の相違があれば、プリコンパイラー・リストに報告されます。

MVSDB2V31 SYNTAX

SQL ステートメントは、MVS DB2 バージョン 3.1 SQL 言語構文と比較して調べられます。構文の相違があれば、プリコンパイラー・リストに報告されます。

MVSDB2V41 SYNTAX

SQL ステートメントは、MVS DB2 バージョン 4.1 SQL 言語構文と比較して調べられます。構文の相違があれば、プリコンパイラー・リストに報告されます。

SORTSEQ

iSeries システムで使用するソート・シーケンス表を指定します。DB2 UDB

for iSeries だけによってサポートされています。サポートされているオプション値のリストについては、DB2 for iSeries のドキュメンテーションを参照してください。

SQLRULES

下記のことを指定します。

- タイプ 2 の CONNECT を DB2 規則にしたがって処理するか、それとも ISO/ANS SQL92 の標準 (STD) の規則にしたがって処理するか。
- ユーザーまたはアプリケーションが LOB 応答セット列の形式をどのように指定するか。

DB2

- SQL CONNECT ステートメントで、現在の接続と、確立されている (休止 状態の) 別の接続との間で切り換えることができるようにします。
- ユーザーまたはアプリケーションは、LOB 列の形式を最初のフェッチ要求においてしか指定できません。

STD

- SQL CONNECT ステートメントでは、新しい 接続を確立することしかできないようにします。休止接続へ切り替えるには、SQL SET CONNECTION ステートメントを使います。
- ユーザーまたはアプリケーションは、LOB 列の形式を各フェッチ要求ごとに指定できます。

SQLWARN

動的 SQL ステートメントの完了時 (PREPARE または EXECUTE IMMEDIATE を通して)、または記述処理時 (PREPARE...INTO または DESCRIBE を通して) に警告を戻すかどうかを指示します。この DB2 プリコンパイル/バインド・オプションは、DRDA ではサポートされていません。

NO SQL コンパイラーから警告を戻しません。

YES SQL コンパイラーから警告を戻します。

注: SQLCODE +238 は例外です。これは、**SQLWARN** オプションの値が何であらうと戻されます。

STRDEL

SQL ステートメントで使用するストリング区切り文字として、アポストロフィ (') または二重引用符 (") のどちらを使用するか指定します。なお、DB2 はこの DRDA プリコンパイル/バインド・オプションをサポートしません。このオプションを指定しないと、DRDA サーバーは、システムが定義したデフォルトを使用します。

PRECOMPILE

APOSTROPHE

ストリング区切り文字として、アポストロフィ (') を使用します。

QUOTE

ストリング区切り文字として、二重引用符 (") を使用します。

SYNCPOINT

複数のデータベース接続にまたがってコミットまたはロールバックを調整する仕方を指定します。

NONE 2 フェーズ・コミットを実行するのにトランザクション管理機能 (TM) を使用しないことを指定し、単一更新機構、多重読み取り機構を適用しません。コミットは、関連する各データベースに送られます。コミットが失敗したときのリカバリーは、アプリケーションが行います。

ONEPHASE

2 フェーズ・コミットを実行するのに TM を使用しないことを指定します。複数のデータベース・トランザクションの各データベースが行う作業をコミットするときは、1 フェーズ・コミットが使用されません。

TWOPHASE

このプロトコルをサポートする複数のデータベースにまたがって 2 フェーズ・コミットを調整するのに TM が必要であることを指定します。

SYNTAX

プリコンパイル時に、パッケージまたはバインド・ファイルの作成を抑制します。このオプションを使用すれば、既存のパッケージまたはバインド・ファイルを修正したり変更したりしないで、ソース・ファイルの妥当性をチェックできます。**Syntax** は **sqlerror check** の同義語です。

syntax を **package** オプションと一緒に使うと、**package** は無視されます。

TARGET

現行のプラットフォームでサポートされているコンパイラーの 1 つに合わせて調整した修正コードを生成するように、プリコンパイラーに指示します。

IBMCOB

AIX では、IBM COBOL Set (AIX 版) のコンパイラーのためのコードが生成されます。

MFCOB

Micro Focus COBOL コンパイラー用のコードが生成されます。すべての UNIX プラットフォームおよび Windows NT の COBOL プリコンパイラーにおいて、**target** 値が指定されないなら、この値がデフォルトになります。

ANSI_COBOL

ANS X3.23-1985 標準規格と互換性のあるコードが生成されます。

C 現行プラットフォーム上の DB2 によりサポートされる C コンパイラと互換性のあるコードが生成されます。

CPLUSPLUS

現行プラットフォーム上の DB2 によりサポートされる C++ コンパイラと互換性のあるコードが生成されます。

FORTRAN

現行プラットフォーム上の DB2 によりサポートされる FORTRAN コンパイラと互換性のあるコードが生成されます。

TEXT label

パッケージの記述。最大長は 255 文字です。また、デフォルトは空白です。なお、DB2 はこの DRDA プリコンパイル/バインド・オプションをサポートしません。

TRANSFORM GROUP

静的 SQL ステートメントが、ユーザー定義の構造型の値をホスト・プログラムと交換するために使用する、変換グループ名を指定します。この変換グループは、動的 SQL ステートメントには使用されません。また、パラメーターの交換や外部関数またはメソッドの結果にも使用されません。このオプションは DRDA サーバーではサポートされていません。

groupname

SQL ID。長さは最大で 18 文字です。グループ名には、修飾子接頭部を含めることはできません。また、接頭部 SYS はデータベースで使用するために予約されているので、その接頭部は使用できません。ホスト変数とやりとりする静的 SQL ステートメントでは、構造型の値の交換に使用する変換グループの名前は以下のようになります。

- TRANSFORM GROUP バインド・オプション内のグループ名 (もしあれば)
- TRANSFORM GROUP 準備オプション内のグループ名。最初のプリコンパイル時に指定したとおりのもの (もしあれば)
- DB2_PROGRAM グループ。グループ名が DB2_PROGRAM の、特定のタイプに対する変換がある場合。
- 上記のいずれの条件もない場合には、変換グループは使用されません。

静的 SQL ステートメントのバインド時には、以下のエラーが発生する可能性があります。

- SQLCODE yyy, SQLSTATE xxxxx: 変換が必要ですが、静的変換グループが選択されていません。

PRECOMPILE

- **SQLCODE yyy, SQLSTATE xxxxx:** 選択された変換グループには、交換するデータ・タイプに必要な変換が含まれていません (入力変数用の **TO SQL**、出力変数用の **FROM SQL**)。
- **SQLCODE yyy, SQLSTATE xxxxx:** **FROM SQL** 変換の結果タイプは、出力変数のタイプと互換性がありません。または、**TO SQL** 変換のパラメーター・タイプは、入力変数のタイプと互換性がありません。

これらのエラー・メッセージで、yyyyy は **SQL** エラー・メッセージによって置き換えられ、xxxxx は **SQL** 状況コードによって置き換えられます。

VALIDATE

データベース・マネージャーが、許可エラーとエラー未検出のオブジェクトをいつチェックするかを判別します。この妥当性検査には、パッケージ所有者の許可 ID が使用されます。

BIND プリコンパイル/バインド時に妥当性検査が実行されます。オブジェクトが 1 つもない場合、または権限がまったく保持されていない場合、エラー・メッセージが作成されます。 **SQLERROR CONTINUE** を指定した場合、エラー・メッセージにかかわらずパッケージ/バインド・ファイルは作成されますが、エラーのあるステートメントは実行できません。

RUN バインド時に妥当性検査が行われます。すべてのオブジェクトが存在し、全権限が保持されていれば、それ以上実行してもチェックは行われません。

プリコンパイル/バインド時にオブジェクトが 1 つもない場合、または権限がまったく保持されていない場合、 **SQLERROR CONTINUE** オプションの設定とは無関係に警告メッセージが作成されて、パッケージは正常にバインドされます。ただし、プリコンパイル/バインド処理時に **SQL** ステートメントの権限検査と存在検査に障害が生じた場合、実行時に再実行される可能性があります。

VERSION

パッケージのバージョン ID を定義します。このオプションが指定されていない場合、パッケージ・バージョンは "" (空ストリング) です。

version-id

任意の英数字値、\$、#、@、_、-、または . で、長さは 64 バイト以下のバージョン ID を指定します。

AUTO バージョン ID は、整合性トークンから生成されます。整合性トークンがタイム・スタンプの場合 (**LEVEL** オプションが指定されていないければそうなります)、そのタイム・スタンプは **ISO** 文字フォーマットに変換されて、バージョン ID として使用されます。

WCHARTYPE

wchartype の使用と適用についての詳細および制約事項は、アプリケーション・プログラミングの手引き を参照してください。

CONVERT

wchar_t 基本タイプを使って宣言されたホスト変数には、**wchar_t** 形式のデータが入られるものとして扱われます。この形式は、データベースに格納されるグラフィック・データの形式 (DBCS 形式) と直接に互換性はないので、**wchar_t** ホスト変数の入力データは、ANSI C 関数 **wcstombs()** を使って DBCS 形式に暗黙のうちに変換されます。同じように出力 DBCS データは、ホスト変数を保管する前に、**mbstowcs()** を使用して **wchar_t** 形式に暗黙のうちに変換されます。

NOCONVERT

wchar_t 基本タイプを使って宣言されたホスト変数には、DBCS 形式のデータが入られるものとして扱われます。これは、データベースの中でグラフィック・データ用に使われる形式ですが、C 言語で採用されている固有の **wchar_t** 形式とは違うものです。**noconvert** を使用する場合、グラフィック・データはアプリケーションとデータベースの間で変換されないことになり、それによって効率が改善されます。しかし、アプリケーション側では、データベース・マネージャーに **wchar_t** 形式のデータが渡されることがないようにする必要があります。このオプションを使用する場合は、**wchar_t** ホスト変数を C ワイド文字列関数で処理しないようにし、ワイド文字リテラル (*L-literals*) で初期化しないようにする必要があります。

使用上の注意:

修正されたソース・ファイルが作成されます。それには SQL ステートメントに相当するホスト言語ステートメントが入っています。パッケージは、デフォルトでは、接続が確立されているデータベース内に作成されます。パッケージの名前は、ファイル名と同じ (拡張子を取り去って大文字に変換したもの) であり、最大 8 文字までです。

データベースへの接続が終わると、開始されているトランザクションの下で **PREP** が実行されます。次に **PREP** は **COMMIT** または **ROLLBACK** を実行し、現行トランザクションを終了して別のトランザクションを開始します。

すでに存在していないスキーマ名を指定してパッケージを作成すると、そのスキーマが暗黙のうちに作成されます。そのスキーマの所有者は **SYSIBM** になります。スキーマに対する **CREATEIN** 特権が **PUBLIC** に付与されます。

プリコンパイル時には、パッケージが作成されて **explsnap** が指定されているのでない限り、**Explain** スナップショットは取られません。スナップショットは、パッケージを作成するユーザーの **Explain** 表に入れられます。同じように、**Explain** 表情報が取得されるのは、**explain** が指定されていて、パッケージが作成される場合だけです。

PRECOMPILE

致命的エラーが発生するか、100 以上のエラーが発生すると、プリコンパイルは停止します。致命的エラーが発生すると、ユーティリティはプリコンパイルを停止し、すべてのファイルをクローズしてからパッケージを廃棄します。

パッケージがバインド動作を公開するとき、以下のとおりとなります。

1. BIND オプション OWNER の暗黙的または明示的な値は、動的 SQL ステートメントの許可検査に使用されます。
2. BIND オプション QUALIFIER の暗黙的または明示的な値は、動的 SQL ステートメント内の修飾オブジェクトを修飾するための暗黙的修飾子として使用されます。
3. 特殊レジスター CURRENT SCHEMA の値は、修飾には影響しません。

単一の接続で複数のパッケージが参照される場合、これらのパッケージによって準備されたすべての動的 SQL ステートメントはその特定のパッケージおよびそれらが使用される環境について DYNAMICRULES オプションで指定された動作を公開します。

SQL ステートメントがエラーであることが検出され、PRECOMPILE オプション SQLERROR CONTINUE が指定されている場合、このステートメントは無効とマークされます。この SQL ステートメントの状態を変えるには、さらに別の PRECOMPILE を発行する必要があります。暗黙的および明示的な再バインドでは、VALIDATE RUN でバインドされたパッケージ内の無効なステートメントの状態は変わりません。ステートメントは、再バインド時にオブジェクトが存在するかまたは権限の問題があるかどうかに応じて、暗黙的と明示的な再バインドとの両方で、静的バインドから増分バインドに変更したり、増分バインドを静的バインドに変更することができます。

関連概念:

- アプリケーション開発ガイド クライアント・アプリケーションのプログラミングの『動的 SQL における許可に関する考慮事項』
- アプリケーション開発ガイド サーバー・アプリケーションのプログラミングの『動的 SQL における DYNAMICRULES の影響』

関連資料:

- コマンド・リファレンスの『BIND コマンド』

db2inspf - 検査結果のフォーマット

このユーティリティは、INSPECT CHECK 結果からのデータを ASCII 形式にフォーマットします。このユーティリティを使用して、検査の詳細を表示します。db2inspf ユーティリティによるフォーマットには、表のみのフォーマット、表スペースのみのフォーマット、エラーのみのフォーマット、警告のみのフォーマット、またはサマリーのみのフォーマットがあります。

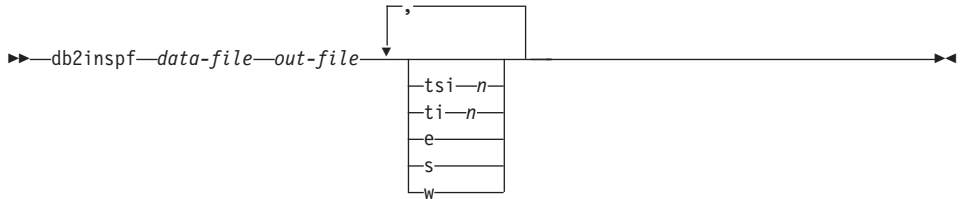
許可:

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

必要な接続:

なし

コマンド構文:



コマンド・パラメーター:

data-file

フォーマットを必要とする不定形式の検査結果ファイル。

out-file フォーマット済み出力の出力ファイル。

-tsi n 表スペース ID。この表スペースの表のみをフォーマットします。

-ti n 表 ID。この ID を持つ表のみをフォーマットします。表スペース ID も指定する必要があります。

-e エラーのみをフォーマットします。

-s サマリーのみ。

-w 警告のみ。

QUIESCE

指定したインスタンスおよびデータベースをすべて強制的にオフにして、静止モードにします。静止モード中は、ユーザーはデータベース・エンジンの外部から接続できません。データベース・インスタンスまたはデータベースが静止モードにある間、それに対して管理タスクを実行できます。管理タスクの完了後、UNQUIESCE コマンドを使用してインスタンスおよびデータベースを活動化して、他のユーザーがデータベースに接続できるようにします。ただし、シャットダウンしたり、他のデータベースの開始を実行したりしないようにする必要があります。

このモードでは、この制限モード中に権限を持つユーザーだけがインスタンス/データベースにアタッチまたは接続することができます。 `sysadm`、`sysmaint`、および `sysctrl` 権

QUIESCE

限を持つユーザーは、インスタンスの静止中に常にそのインスタンスにアクセスでき、*sysadm* および *dbadm* 権限を持つユーザーは、データベースの静止中に常にそのデータベースにアクセスできます。

有効範囲:

QUIESCE DATABASE を使用すると、データベース内のすべてのオブジェクトは静止モードに入ります。許可が与えられたユーザー / グループ、および *sysadm*、*sysmaint*、*dbadm*、または *sysctrl* だけが、データベースまたはそのオブジェクトにアクセスできません。

QUIESCE INSTANCE *instance-name* は、インスタンス *instance-name* 内のインスタンスおよびデータベースが静止モードに入ることを意味します。インスタンスにアクセスできるのは、*sysadm*、*sysmaint*、および *sysctrl* と、許可が与えられているユーザー / グループだけです。

インスタンスが静止モードにある場合、インスタンス内のデータベースを静止モードにすることはできません。

許可:

以下のいずれかが必要です。

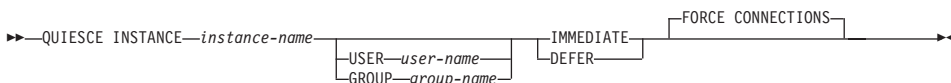
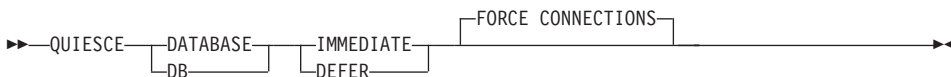
データベース・レベルの静止の場合:

- *sysadm*
- *dbadm*

インスタンス・レベルの静止の場合:

- *sysadm*
- *sysctrl*

コマンド構文:



必要な接続:

データベース

(インスタンスの静止ではデータベース接続は必須ではありません。)

コマンド・パラメーター:

DEFER アプリケーションが現行の作業単位をコミットするまでアプリケーションを待ちます。このパラメーターは現在機能しません。

IMMEDIATE

トランザクションがコミットされるのを待たず、即時にトランザクションをロールバックします。

FORCE CONNECTIONS

接続を強制的にオフにします。

DATABASE

データベースを静止します。データベース内のすべてのオブジェクトを静止モードにします。指定したグループのユーザーと、*sysadm*、*sysmaint*、および *sysctrl* 権限を持つユーザーだけが、データベースまたはそのオブジェクトにアクセスすることができます。

INSTANCE *instance-name*

インスタンス *instance-name*、およびインスタンス内のデータベースを静止モードにします。インスタンスにアクセスできるのは、*sysadm*、*sysmaint*、および *sysctrl* 権限を持つユーザーと、指定したグループのユーザーだけです。

USER *user-name*

インスタンスの静止中にそのインスタンスへのアクセスが許可されているユーザーの名前を指定します。

GROUP *group-name*

インスタンスの静止中にそのインスタンスへのアクセスが許可されているグループの名前を指定します。

例:

以下に示すのは、接続の強制がデフォルトの動作であるため、明示的な説明は必要なく、この例から省くことができる例です。

```
db2 quiesce instance crankarm user frank immediate force connections
```

以下の例は、データベースに接続しているすべてのユーザーを、強制的に切断にします。

```
db2 quiesce db immediate
```

- 最初の例は、インスタンス *crankarm* を静止し、ユーザー *frank* は引き続きデータベースを使用できるようにします。
- 2番目の例は、アタッチしているデータベースを静止し、以下のいずれかの権限をもつユーザー以外のユーザーは、すべてアクセスできないようにします。*sysadm*、*sysmaint*、*sysctrl* または *dbadm*

QUIESCE

- このコマンドとともに **FORCE CONNECTION** オプションを指定すると、データベースまたはインスタンスをすべて強制的にオフにします。 **FORCE CONNECTION** はデフォルトの動作です。コマンドのパラメーターは、互換性の理由により許容されています。
- コマンドは **FORCE** と同期され、**FORCE** が完了しないと完了しません。

使用上の注意:

- **QUIESCE INSTANCE** の後、 *sysadm*、 *sysmaint*、または *sysctrl* 権限を持つユーザー、またはコマンドにパラメーターとして指定するユーザー名およびグループ名だけが、インスタンスに接続できます。
- **QUIESCE DATABASE** の後、 *sysadm*、 *sysmaint*、 *sysctrl*、または *dbadm* 権限、および **GRANT/REVOKE** 特権を持つユーザーは、接続可能なユーザーを指定できます。この情報は永続的にデータベース・カタログ表に保管されます。
たとえば、以下のようになります。

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

REORGCHK

データベースでの統計を計算し、表または索引もしくはその両方が、再編成またはクリーンアップされる必要があるか判別します。

有効範囲:

このコマンドは、*db2nodes.cfg* ファイル中のどのデータベース・パーティションからでも発行できます。これを使用して、カタログ中の表および索引統計を更新できます。

許可:

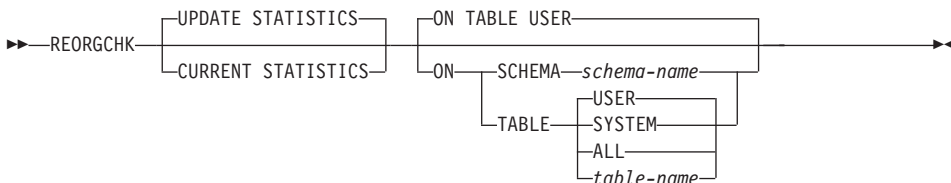
以下のいずれかが必要です。

- *sysadm* または *dbadm* 権限
- 表に対する **CONTROL** 特権

必要な接続:

データベース

コマンド構文:



コマンド・パラメーター:

UPDATE STATISTICS

RUNSTATS ルーチン呼び出して、表の統計を更新してから、更新された統計を使用して、表の再編成が必要であるか判別します。

REORGCHK を発行したノードに表パーティションが存在する場合、このノードで RUNSTATS を実行します。このノードに表パーティションが存在しない場合、その要求は表のパーティションを保留しているデータベース・パーティション・グループ中の最初のノードに送信されます。次にそのノードで RUNSTATS を実行します。

CURRENT STATISTICS

現在の表の統計を使用して、表の再編成が必要であるか判別します。

ON SCHEMA *schema-name*

指定のスキーマの下で作成されたすべての表をチェックします。

ON TABLE

USER ランタイム許可 ID が所有する表をチェックします。

SYSTEM

システムの表をチェックします。

ALL すべてのユーザーおよびシステムの表をチェックします。

table-name

チェックする表を指定します。名前は、*schema.table-name* 形式の完全修飾名または別名でなければなりません。 *schema* には、表作成時のユーザー名が入ります。指定された表がシステム・カタログの表である場合には、 *schema* は SYSIBM です。

注: 型付き表の場合、指定する表名は階層のルート表の名前でなければなりません。

例:

以下に示すのは、次のコマンドの出力例です。 *Index Statistics* 出力の一部だけが示されています。

```
db2 reorgchk update statistics on table system
```

SAMPLE データベースに対して実行すると、

REORGCHK

|
| Doing RUNSTATS
|

| Table statistics:
|

| F1: $100 * \text{OVERFLOW} / \text{CARD} < 5$

| F2: $100 * (\text{Effective Space Utilization of Data Pages}) > 70$

| F3: $100 * (\text{Required Pages} / \text{Total Pages}) > 80$
|
|

SCHEMA	NAME	CARD	OV	NP	FP	ACTBLK	TSIZE	F1	F2	F3	REORG
SYSIBM	SYSATTRIBUTES	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYSBUFFERPOOLNODES	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYSBUFFERPOOLS	1	0	1	1	-	52	0	-	100	---
SYSIBM	SYSCHECKS	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYSCODEPROPERTIES	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYSCOLAUTH	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYSCOLCHECKS	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYSCOLDIST	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYSCOLGROUPDIST	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYSCOLGROUPDISTCO>	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYSCOLGROUPS	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYSCOLGROUPSCOLS	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYSOLOPTIONS	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYSCOLPROPERTIES	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYSCOLUMNS	2812	34	141	143	-	553964	1	95	98	---
SYSIBM	SYSCOLUSE	4	0	1	1	-	156	0	-	100	---
SYSIBM	SYS COMMENTS	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYS CONSTDEP	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYS DATATYPES	17	0	3	3	-	14399	0	100	100	---
SYSIBM	SYS DBAUTH	2	0	1	1	-	92	0	-	100	---
SYSIBM	SYS DEPENDENCIES	6	0	1	1	-	468	0	-	100	---
SYSIBM	SYS EVENTMONITORS	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYS EVENTS	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYS EVENTTABLES	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYS FUNCMAPOPTIONS	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYS FUNCMAPPARMOPT>	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYS FUNC MAPPINGS	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYS HIERARCHIES	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYS INDEXAUTH	6	0	1	1	-	408	0	-	100	---
SYSIBM	SYS INDEXCOLUSE	506	0	7	7	-	23782	0	97	100	---
SYSIBM	SYS INDEXES	188	89	17	31	-	159988	47	100	54	*--
SYSIBM	SYS INDEXEXPLOITRU>	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYS INDEXEXTENSION>	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYS INDEXEXTENSION>	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYS INDEXEXTENSIONS	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYS INDEXOPTIONS	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYS JARCONTENTS	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYS JAROBJECTS	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYS KEYCOLUSE	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYS LIBRARIES	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYS LIBRARYAUTH	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYS LIBRARYBINDFIL>	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYS LIBRARYVERSIONS	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYS NAME MAPPINGS	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYS NODEGROUPDEF	2	0	1	1	-	60	0	-	100	---
SYSIBM	SYS NODEGROUPS	3	0	1	1	-	174	0	-	100	---
SYSIBM	SYS PARTITIONMAPS	4	0	1	1	-	160	0	-	100	---
SYSIBM	SYS PASSTRU AUTH	-	-	-	-	-	-	-	-	-	---
SYSIBM	SYS PLAN	81	0	7	7	-	66987	0	100	100	---
SYSIBM	SYS PLAN AUTH	159	0	3	3	-	9858	0	100	100	---

REORGCHK

SYSIBM	SYSPLANDEP	112	0	3	3	-	20720	0	100	100	---
SYSIBM	SYSPREDICATESPECS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSPROCOPTIONS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSPROCPARMOPTIONS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSRELS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSROUTINEAUTH	58	0	2	2	-	5046	0	100	100	---
SYSIBM	SYSROUTINEPARMS	1077	0	42	42	-	161550	0	96	100	---
SYSIBM	SYSROUTINEPROPERT>	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSROUTINES	195	0	21	21	-	134160	0	100	100	---
SYSIBM	SYSSCHEMAAUTH	2	0	1	1	-	100	0	-	100	---
SYSIBM	SYSSCHEMATA	7	0	1	1	-	427	0	-	100	---
SYSIBM	SYSSECTION	133	0	5	5	-	55195	0	100	100	---
SYSIBM	SYSSEQUENCEAUTH	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSSEQUENCES	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSSEVEROPTIONS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSSEVERERS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSSTMT	133	0	5	5	-	51205	0	100	100	---
SYSIBM	SYSTABAUTH	242	0	5	5	-	17666	0	100	100	---
SYSIBM	SYSTABCONST	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSTABLES	243	0	27	27	-	388071	0	100	100	---
SYSIBM	SYSTABLESPACES	3	0	1	1	-	321	0	-	100	---
SYSIBM	SYSTABOPTIONS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSTBSPACEAUTH	1	0	1	1	-	54	0	-	100	---
SYSIBM	SYSTRANSFORMS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSTRIGGERS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSTYPEMAPPINGS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSUSERAUTH	242	0	8	8	-	52998	0	100	100	---
SYSIBM	SYSUSEROPTIONS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSVERSIONS	1	0	1	1	-	36	0	-	100	---
SYSIBM	SYSVIEWDEP	214	0	5	5	-	18404	0	100	100	---
SYSIBM	SYSVIEWS	144	0	7	7	-	37872	0	100	100	---
SYSIBM	SYSWRAPOPTIONS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSWRAPPERS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSXMLOBJECTAUTH	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSXMLOBJECTAUTHP>	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSXMLOBJECTPROPE>	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSXMLOBJECTRELDEP	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSXMLOBJECTS	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSXMLOBJECTXMLDEP	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSXMLPHYSICALCOL>	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSXMLQUERIES	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSXMLRELATIONSHI>	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSXMLRSPROPERTIES	-	-	-	-	-	-	-	-	-	----
SYSIBM	SYSXMLSTATS	-	-	-	-	-	-	-	-	-	----

Index statistics:

F4: CLUSTERRATIO or normalized CLUSTERFACTOR > 80

F5: $100 * (KEYS * (ISIZE + 9) + (CARD - KEYS) * 5) / ((NLEAF - NUM_EMPTY_LEAFS) * INDEXPAGESIZE) > 50$

F6: $(100 - PCTFREE) * ((INDEXPAGESIZE - 96) / (ISIZE + 12)) ** (NLEVELS - 2) * (INDEXPAGESIZE - 96) / (KEYS * (ISIZE + 9) + (CARD - KEYS) * 5) < 100$

F7: $100 * (NUMRIDS DELETED / (NUMRIDS DELETED + CARD)) < 20$

F8: $100 * (NUM EMPTY LEAFS / NLEAF) < 20$

SCHEMA	NAME	CARD	LEAF	ELEAF	LVLS	ISIZE	NDEL	KEYS	F4	F5	F6	F7	F8	REORG

Table: SYSIBM.SYSATTRIBUTES														
SYSIBM	IBM83	-	-	-	-	-	-	-	-	-	-	-	-	-----
SYSIBM	IBM84	-	-	-	-	-	-	-	-	-	-	-	-	-----
SYSIBM	IBM85	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSBUFFERPOOLNODES														
SYSIBM	IBM69	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSBUFFERPOOLS														
SYSIBM	IBM67	1	1	0	1	22	0	1	100	-	-	0	0	-----
SYSIBM	IBM68	1	1	0	1	10	0	1	100	-	-	0	0	-----
Table: SYSIBM.SYSCHECKS														
SYSIBM	IBM37	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSCODEPROPERTIES														
SYSIBM	IBM161	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSCOLAUTH														
SYSIBM	IBM42	-	-	-	-	-	-	-	-	-	-	-	-	-----
SYSIBM	IBM43	-	-	-	-	-	-	-	-	-	-	-	-	-----
SYSIBM	IBM64	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSCOLCHECKS														
SYSIBM	IBM38	-	-	-	-	-	-	-	-	-	-	-	-	-----
SYSIBM	IBM39	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSCOLDIST														
SYSIBM	IBM46	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSCOLGROUPDIST														
SYSIBM	IBM157	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSCOLGROUPDISTCOUNTS														
SYSIBM	IBM158	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSCOLGROUPS														
SYSIBM	IBM154	-	-	-	-	-	-	-	-	-	-	-	-	-----
SYSIBM	IBM155	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSCOLGROUPSCOLS														
SYSIBM	IBM156	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSCOLOPTIONS														
SYSIBM	IBM89	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSCOLPROPERTIES														
SYSIBM	IBM79	-	-	-	-	-	-	-	-	-	-	-	-	-----
SYSIBM	IBM80	-	-	-	-	-	-	-	-	-	-	-	-	-----
SYSIBM	IBM82	-	-	-	-	-	-	-	-	-	-	-	-	-----
Table: SYSIBM.SYSCOLUMNS														
SYSIBM	IBM01	2812	59	0	2	41	12	2812	95	58	2	0	0	-----
SYSIBM	IBM24	2812	7	0	2	23	3	11	82	50	25	0	0	-*---
Table: SYSIBM.SYSCOLUSE														
SYSIBM	IBM146	4	1	0	1	22	1	4	100	-	-	20	0	----*
SYSIBM	IBM147	4	1	0	1	23	1	4	100	-	-	20	0	----*
Table: SYSIBM.SYSCOMMENTS														
SYSIBM	IBM73	-	-	-	-	-	-	-	-	-	-	-	-	-----

REORGCHK

```

Table: SYSIBM.SYSCONSTDEP
SYSIBM  IBM44          -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -
SYSIBM  IBM45          -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -
Table: SYSIBM.SYSDATATYPES
SYSIBM  IBM40          17   1   0   1  23   0   17 100  -   -   0   0  -   -   -
SYSIBM  IBM41          17   1   0   1   2   0   17 100  -   -   0   0  -   -   -
SYSIBM  IBM56          17   1   0   1  23   0   17 100  -   -   0   0  -   -   -
Table: SYSIBM.SYSDBAUTH
SYSIBM  IBM12           2   1   0   1  25   0    2 100  -   -   0   0  -   -   -
Table: SYSIBM.SYSDEPENDENCIES
SYSIBM  IBM51           6   1   0   1  66   0    6 100  -   -   0   0  -   -   -
SYSIBM  IBM52           6   1   0   1  25   0    5 100  -   -   0   0  -   -   -
Table: SYSIBM.SYSEVENTMONITORS
SYSIBM  IBM47           -   -   -   -   -   -   -   -   -   -   -   -   -   -   -
Table: SYSIBM.SYSEVENTS
SYSIBM  IBM48           -   -   -   -   -   -   -   -   -   -   -   -   -   -   -
Table: SYSIBM.SYSEVENTTABLES
SYSIBM  IBM165          -   -   -   -   -   -   -   -   -   -   -   -   -   -   -
SYSIBM  IBM167          -   -   -   -   -   -   -   -   -   -   -   -   -   -   -
Table: SYSIBM.SYSFUNCMAPOPTIONS
SYSIBM  IBM90           -   -   -   -   -   -   -   -   -   -   -   -   -   -   -
Table: SYSIBM.SYSFUNCMAPPARMOPTIONS
SYSIBM  IBM91           -   -   -   -   -   -   -   -   -   -   -   -   -   -   -
Table: SYSIBM.SYSFUNCMAPPINGS
SYSIBM  IBM92           -   -   -   -   -   -   -   -   -   -   -   -   -   -   -
SYSIBM  IBM93           -   -   -   -   -   -   -   -   -   -   -   -   -   -   -
SYSIBM  IBM94           -   -   -   -   -   -   -   -   -   -   -   -   -   -   -
SYSIBM  IBM95           -   -   -   -   -   -   -   -   -   -   -   -   -   -   -
SYSIBM  IBM96           -   -   -   -   -   -   -   -   -   -   -   -   -   -   -
Table: SYSIBM.SYSHIERARCHIES
SYSIBM  IBM86           -   -   -   -   -   -   -   -   -   -   -   -   -   -   -
SYSIBM  IBM87           -   -   -   -   -   -   -   -   -   -   -   -   -   -   -
Table: SYSIBM.SYSINDEXAUTH
SYSIBM  IBM17           6   2   0   2  57   96    6 100  4 909 100  0  -   -   -
SYSIBM  IBM18           6   2   0   2  32  151    6 100  3 +++ 100  0  -   -   -
Table: SYSIBM.SYSINDEXCOLUSE
SYSIBM  IBM139          506  9   0   2  36  114   506 96  61  15  20  0  -   -   -
Table: SYSIBM.SYSINDEXES
SYSIBM  IBM02           188  4   1   2  22   40   188 80  47  61  20  0  **--
SYSIBM  IBM03           188  3   0   2  31   8   188 77  61  47  4   0  *---
SYSIBM  IBM126          188  1   0   1  12   4    1 100  -   -   2   0  -   -   -
Table: SYSIBM.SYSINDEXEXPLOITRULES
SYSIBM  IBM97           -   -   -   -   -   -   -   -   -   -   -   -   -   -   -
Table: SYSIBM.SYSINDEXEXTENSIONMETHODS
SYSIBM  IBM100          -   -   -   -   -   -   -   -   -   -   -   -   -   -   -
SYSIBM  IBM101          -   -   -   -   -   -   -   -   -   -   -   -   -   -   -
Table: SYSIBM.SYSINDEXEXTENSIONPARMS
SYSIBM  IBM98           -   -   -   -   -   -   -   -   -   -   -   -   -   -   -
Table: SYSIBM.SYSINDEXEXTENSIONS
SYSIBM  IBM99           -   -   -   -   -   -   -   -   -   -   -   -   -   -   -
...

```

CLUSTERRATIO or normalized CLUSTERFACTOR (F4) will indicate REORG is necessary for indexes that are not in the same sequence as the base table. When multiple indexes are defined on a table, one or more indexes may be flagged as needing REORG. Specify the most important index for REORG sequencing.

Tables defined using the DIMENSION clause and the corresponding dimension indexes have a '*' suffix to their names. The cardinality of a dimension index is equal to the Active blocks statistic of the table.

表統計の用語 (式 1 ~ 3) の意味は以下のとおりです。

CARD 基本表の行数。

OV (OVERFLOW) オーバーフローした行数。

NP (NPAGES) データを含むページ数。

FP (FPAGES) ページの合計数。

ACTBLK

多次元クラスタリング (MDC) 表のアクティブ・ブロックの合計数。このフィールドは、ORGANIZE BY 文節を使用して定義された表に対してのみ適用できます。これは、データを含む表のブロック数を示します。

TSIZE 表サイズ (バイト数)。表 (CARD) 内の行数と行の長さの平均を基にして計算されます。行の長さの平均は、列の長さの平均 (SYSCOLUMNS の AVGCOLLEN) の合計に、行のオーバーヘッドとして 10 バイトを加えたものを基にして計算されます。長フィールドと LOB の場合には、記述子のおおよその長さだけが使用されます。実際の長フィールドまたは LOB データは、TSIZE にはカウントされません。

TABLEPAGESIZE

表データが存在する表スペースのページ・サイズ。

F1 式 1 の結果。

F2 式 2 の結果。

F3 式 3 の結果。

REORG

この列に表示されている各ハイフン (-) は、計算結果が、対応する式の設定範囲内であったことを示しています。各アスタリスク (*) は、計算結果が、その対応する式の設定範囲を超えたことを示しています。

- 列の左側の - または * は F1 (公式 1) に対応しています。
- 列の中央の - または * は F2 (公式 2) に対応しています。
- 列の右側の - または * は F3 (公式 3) に対応しています。

表の再編成は、その計算結果が式によって設定された範囲を超える場合に、提案されます。

たとえば、--- は、F1、F2、および F3 の式の結果がその式の設定範囲内であるために、再編成が提案される表はないことを示しています。表記 *-*

は、F2 の結果がその設定範囲内であっても、F1 と F3 の結果が表の再編成を提案していることを示しています。表記 *-- は、F1 の式のみが、その範囲を超えていることを示しています。

注: 表名は 30 文字で切り捨てられ、31 列目の ">" 記号が表名の切り捨て位置を表します。表名への "*" 接尾部は、これが MDC 表であることを示します。索引名への "*" 接尾部は、これが MDC ディメンション索引であることを示します。

索引統計の用語 (式 4 ~ 8) の意味は、以下のとおりです。

CARD 基本表の行数。

LEAF 索引リーフ・ページ (NLEAF) の合計数。

ELEAF 疑似空白リーフ・ページ (NUM_EMPTY_LEAFS) の数

疑似空白索引リーフ・ページは、すべての RID に削除済みのマークが付いていますが、それらが物理的には削除されていないページです。

NDEL 疑似削除された RID の数 (NUMRIDS_DELETED)

疑似削除された RID とは、削除済みのマークが付いた RID のことです。この統計は、疑似空白ではないリーフ・ページ上の疑似削除された RID に関して報告します。すべての RID に削除済みのマークが付いたリーフ・ページ上の、削除マークの付いた RID は含まれません。

LVLS 索引レベルの数 (NLEVELS)

ISIZE 索引サイズ、索引に関係するすべての列による列の平均の長さから計算される。

KEYS 削除マークの付いていないユニーク索引項目の数 (FULLKEYCARD)

INDEXPAGESIZE

表索引が存在する表スペースのページ・サイズ。表の作成時に指定します。指定しなかった場合、INDEXPAGESIZE の値は TABLEPAGESIZE の値と同じになります。

PCTFREE

各索引ページでフリー・スペースのままにしておくパーセントを指定します。値は索引の定義時に割り当てられます。値の範囲は 0 ~ 99 です。デフォルトは 10 です。

F4 式 4 の結果。

F5 式 5 の結果。表記法 +++ は、結果が 999 を超えて無効であることを示しています。UPDATE STATISTICS オプションを指定して REORGCHK を戻すか、REORGCHK コマンドに続けて RUNSTATS を発行してください。

F6 式 6 の結果。表記法 +++ は、結果が 9999 を超えて無効である可能性があることを示しています。UPDATE STATISTICS オプションを指定して

REORGCHK を戻すか、REORGCHK コマンドに続けて RUNSTATS を発行してください。統計が現行のものであり、有効であれば、再編成してください。

F7 式 7 の結果。

F8 式 8 の結果。

REORG

この列に表示されている各ハイフン (-) は、計算結果が、対応する式の設定範囲内であったことを示しています。各アスタリスク (*) は、計算結果が、その対応する式の設定範囲を超えたことを示しています。

- 左列の - または * は F4 (式 4) に対応しています。
- 左から 2 番目の列の - または * は F5 (式 5) に対応しています。
- 列の中央の - または * は F6 (式 6) に対応しています。
- 右から 2 番目の列の - または * は F7 (式 7) に対応しています。
- 右列の - または * は F8 (式 8) に対応しています。

索引を再編成する際の提案を以下に示します。

- 式 1、2、および 3 の計算結果がその式によって設定された境界を超えないで、式 4、5、または 6 の計算結果が設定された境界を超える場合、索引を再編成することをお勧めします。
- 式 7 の計算結果だけが設定された境界を超えて、式 1、2、3、4、5、および 6 の結果は設定された境界内にある場合、REORG INDEXES の CLEANUP ONLY オプションを使用して索引をクリーンアップすることをお勧めします。
- 式 8 の計算結果だけが設定された境界を超える場合、REORG INDEXES の CLEANUP ONLY PAGES オプションを使用して索引の疑似空白ページをクリーンアップすることをお勧めします。

使用上の注意:

このコマンドは、宣言一時表の統計情報は表示しません。

このユーティリティーでは、通称の使用はサポートされません。

CURRENT STATISTICS オプションを指定していなければ、REORGCHK はデフォルトのオプションだけを使用してすべての列から統計を収集します。特に、列グループは収集されません。さらに、LIKE 統計が以前に収集されている場合、それらは REORGCHK によっては収集されません。

収集される統計は、カタログ表に現在保管されている統計の種類によって異なります。

- いずれかの索引のカタログ内に詳細索引統計が存在する場合、すべての索引の表統計および詳細索引統計が収集されます (サンプリングは行われません)。

REORGCHK

- 詳細索引統計が収集されない場合、すべての索引の表統計および通常の索引統計が収集されます。
- 分散統計が検出された場合、その表についての分散統計が収集されます。分散統計が収集された場合、頻度および変位値の数はデータベース構成パラメーターの設定値によって異なります。

REORGCHK は、8 つの異なる公式から得た統計を計算し、表またはその索引の再編成によってパフォーマンスが低下するか、または改善できるのかを判別します。

重要: 空の表 (TSIZE=0) が再編成を必要としている場合には、これらの統計を参考にせず、次の規準を当てはめてください。つまり、TSIZE=0 で FPAGE>0 の場合は、表の再編成が必要です。TSIZE=0 で FPAGE=0 の場合は、再編成は必要ありません。

REORGCHK は、以下の式を使用して、行の物理的なロケーションおよび表のサイズを分析します。

- 式 F1:

$$100*\text{OVERFLOW}/\text{CARD} < 5$$

表のオーバーフロー行の合計数は、行の合計数の 5% 以下でなければなりません。オーバーフロー行は、行が更新されて、新しい行のバイト数が古い行 (VARCHAR フィールド) のそれより大きくなる場合、または列が既存の表に追加される場合に作成されます。

- 式 F2:

通常表では、以下のようになります。

$$100*\text{TSIZE} / ((\text{FPAGES}-1) * (\text{TABLEPAGESIZE}-76)) > 68$$

バイトで表された表のサイズ (TSIZE) は、表に割り当てられた合計スペースの 68 を超えていなければなりません (フリー・スペースを 32 以下の値にします)。表に割り当てられる合計スペースは、表データが存在する表スペースのページ・サイズによって決まります (オーバーヘッド分の 76 バイトを差し引きます)。割り当てられている最終ページは、通常埋められていないため、FPAGES から 1 を引きます。

MDC 表では、以下のようになります。

$$100*\text{TSIZE} / ((\text{ACTBLK}-\text{FULLKEYCARD}) * \text{EXTENTSIZ} * (\text{TABLEPAGESIZE}-76)) > 68$$

FULLKEYCARD は、MDC 表の複合ディメンション索引のカーディナリティーを示します。エクステント・サイズは、ブロックごとのページ数です。公式は、バイト単位の表サイズが、必要最低限のブロック数を差し引いた後の表のブロックの 68% 以上かどうかをチェックします。

- 式 F3:

$$100*\text{NPAGES}/\text{FPAGES} > 80$$

全く行を含まないページ数は、ページ合計数の 20% より少ない値にします (行が削除された後では、ページは空になります)。

REORGCHK は、以下の式を使用して、索引および表データに対する索引のリレーションシップを分析します。

- 式 F4:

$$\text{CLUSTERRATIO or normalized CLUSTERFACTOR} > 80$$

クラスタリング索引比率は、80% より大きくします。複数の索引が 1 つの表に定義される場合は、これらの索引のいくつかは、低いクラスター比率を持っています (索引順序は、表の順序と同じではありません)。これを避けることはできません。表を再編成する際に、必ず最も重要な索引を指定してください。そのクラスター比率は、通常、数の多い複写キーおよび数の多い項目を含む索引には最適ではありません。

- 式 F5:

$$100 * (\text{KEYS} * (\text{ISIZE} + 9) + (\text{CARD} - \text{KEYS}) * 5) / \\ ((\text{NLEAF} - \text{NUM_EMPTY_LEAFS}) * \text{INDEXPAGESIZE}) > 50$$

索引項目に予約されたスペースの 50% 以下は空でなければなりません (NLEAF>1 の場合のみチェックされます)。

- 式 F6:

$$(100 - \text{PCTFREE}) * ((\text{INDEXPAGESIZE} - 96) / (\text{ISIZE} + 12)) ** \\ (\text{NLEVELS} - 2) * (\text{INDEXPAGESIZE} - 96) / (\text{KEYS} * (\text{ISIZE} + 9) + (\text{CARD} - \text{KEYS}) * 5) \\ < 100$$

索引の再作成がツリーのレベル数を減少させるかどうかを判別するために、この式は現行のツリーよりも 1 つ低いレベルの索引ツリー内にあるスペースの量と必要なスペースの量との比率をチェックします。1 つ低いレベルのツリーを作成しても PCTFREE が使用可能であれば、再編成をお勧めします。索引項目の実際数は、NLEVELS-1 索引ツリーが取り扱うことができる項目の数の 90% 以上 (または 100-PCTFREE) でなければなりません (NLEVELS>1 の場合のみチェックされます)。

- 式 F7:

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

疑似空白ではないページ上の疑似削除された RID の数は 20% 未満でなければなりません。

- 式 F8:

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

疑似空白リーフ・ページ数は、リーフ・ページの合計数の 20% 未満でなければなりません。

注: 多数の表で統計を実行すると、表が大きい場合には特に時間がかかります。

関連資料:

- コマンド・リファレンス の『REORG INDEXES/TABLE コマンド』
- 1258 ページの『RUNSTATS』

IMPORT

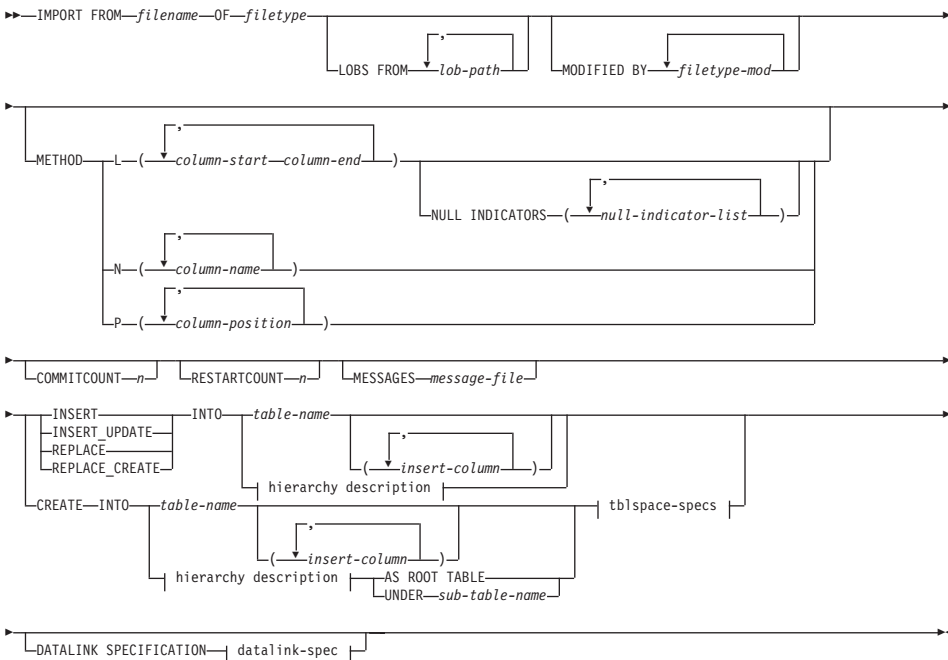
外部ファイルのデータを、サポートされているファイル形式で表、階層、またはビューに挿入します。LOAD はより高速な代替方法です。しかしロード・ユーティリティでは、階層レベルのデータのロードはサポートされていません。

許可:

- INSERT オプションを使用して IMPORT コマンドを実行する場合、以下のいずれかが必要です。
 - *sysadm*
 - *dbadm*
 - 関係する表またはビューのそれぞれに対する CONTROL 特権
 - 関係する表またはビューのそれぞれに対する INSERT および SELECT 特権
- INSERT_UPDATE、REPLACE、または REPLACE_CREATE オプションを使用して既存の表に IMPORT コマンドを実行する場合は、以下のいずれかが必要です。
 - *sysadm*
 - *dbadm*
 - 表またはビューに対する CONTROL 特権
- CREATE または REPLACE_CREATE オプションを使って、存在しない表または階層に IMPORT するには、以下のいずれかが必要です。
 - *sysadm*
 - *dbadm*
 - データベースに対する CREATETAB 権限、および以下のどちらか。
 - 表のスキーマ名が存在しない場合は、データベースに対する IMPLICIT_SCHEMA 権限
 - 表のスキーマ名が存在する場合は、スキーマに対する CREATEIN 特権
 - 階層全体に対して REPLACE_CREATE オプションが使用されている場合は、階層内のすべての副表に対する CONTROL 特権
- REPLACE オプションを使用して既存の階層に IMPORT するには、以下のいずれかが必要です。
 - *sysadm*
 - *dbadm*
 - 階層内のすべての副表に対する CONTROL 特権

必要な接続:

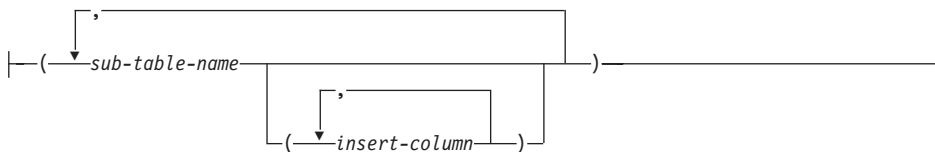
データベース。暗黙接続が可能な場合には、デフォルト・データベースへの接続が確立されます。



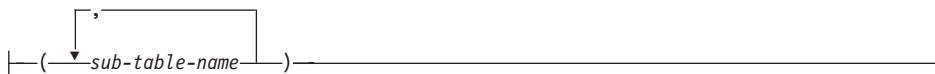
hierarchy description:



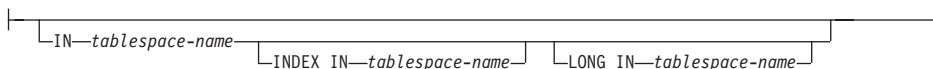
sub-table-list:



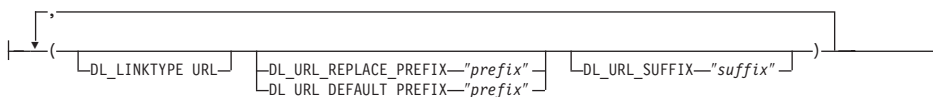
traversal-order-list:



tblspace-specs:



datalink-spec:



コマンド・パラメーター:

ALL TABLES

暗黙のキーワード (階層のみ)。階層をインポートする場合、走査順序で指定されるすべての表をインポートすることがデフォルトです。

AS ROOT TABLE

1 つ以上の副表を、独立した表階層として作成します。

COMMITCOUNT *n*

n 個のレコードがインポートされるたびに COMMIT を実行します。

CREATE

表定義と行の内容を作成します。DB2 の表、副表、または階層からエクスポートされたデータの場合、索引も作成されます。このオプションが階層に対するものである場合で、DB2 からデータがエクスポートされた場合には、タイプ階層も作成されます。このオプションは、IXF ファイルの場合にのみ使用することができます。

注: データが MVS ホスト・データベースからエクスポートされたもので、ページ・サイズで計算した長さが 254 より少ない LONGVAR フィールドを含んでいる場合、CREATE は行が長過ぎるために失敗します。この場合、その表は手動で作成します。そして、IMPORT に INSERT を指定して呼び出すか、または LOAD コマンドを使用してください。

DATALINK SPECIFICATION

各 DATALINK 列ごとに、それぞれ 1 つの列指定を括弧で囲んで指定できます。各列指定は、1 つ以上の DL_LINKTYPE、接頭部、および DL_URL_SUFFIX 指定で構成されます。接頭部指定は、DL_URL_REPLACE_PREFIX または DL_URL_DEFAULT_PREFIX のどちらかになります。

DATALINK 列指定の数は、表で定義されている DATALINK 列の数と同じだけ指定できます。指定の順序は、挿入列 リストの中での DATALINK 列の順序、または表定義内での順序 (挿入列 リストが指定されていない場合) に従います。

DL_LINKTYPE

指定した場合は、列定義の LINKTYPE に一致していなければなりません。そうすることによって、列定義に LINKTYPE URL が指定されている場合に DL_LINKTYPE URL が受け入れ可能になります。

DL_URL_DEFAULT_PREFIX "prefix"

これを指定すると、同じ列内のすべての DATALINK 値のデフォルト接頭部になります。ここでいう接頭部とは、URL 指定の「スキーム・ホスト・ポート」部分のことです。(分散ファイル・システム (DFS) の場合、接頭部とは URL 指定の「スキーム・セル名とファイル・スペースの接合」部分のことです。)

接頭部の例

```
"http://server"
"file://server"
"file:"
"http://server:80"
"dfs://.../cellname/fs"
```

列のデータの中に接頭部がない場合、DL_URL_DEFAULT_PREFIX でデフォルトの接頭部が指定されているなら、列の値の接頭部としてそのデフォルト接頭部が付けられます (NULL でない場合)。

たとえば、DL_URL_DEFAULT_PREFIX でデフォルト接頭部が "http://toronto" として指定されている場合、

- 列入力値 "/x/y/z" は "http://toronto/x/y/z" として保管されます。
- 列入力値 "http://coyote/a/b/c" は "http://coyote/a/b/c" として保管されます。
- 列入力値 NULL は NULL として保管されます。

DL_URL_REPLACE_PREFIX "prefix"

この文節は、それ以前にエクスポート・ユーティリティによって生成されたデータをロードまたはインポートする際に、ユーザーがデータに含まれるホスト名を別のホスト名に一括置換したい場合に便利です。指定する場合には、それがすべての非 NULL 列値の接頭部になります。列値にすでに接頭部があるなら、それは置き換えられます。列値に接頭部がないなら、DL_URL_REPLACE_PREFIX で指定される接頭部がその列値の接頭部になります。分散ファイル・システム (DFS) の場合、接頭部とは URL 指定の「スキーム・セル名とファイル・スペースの接合」部分のことです。

たとえば、DL_URL_REPLACE_PREFIX で接頭部が "http://toronto" として指定されている場合、

- 列入力値 "/x/y/z" は "http://toronto/x/y/z" として保管されます。
- 列入力値 "http://coyote/a/b/c" は "http://toronto/a/b/c" として保管されます。
"coyote" は "toronto" に置き換えられます。
- 列入力値 NULL は NULL として保管されます。

DL_URL_SUFFIX "suffix"

これを指定すると、それはその列のすべての非 NULL 列値に付加されます。これは実際には、DATALINK 値の URL 部分の「パス」コンポーネントに付加されます。

FROM filename

インポートするデータを含むファイルを指定します。パスを省略すると、現行の作業ディレクトリーが使用されます。

HIERARCHY

インポートする階層データを指定します。

IN tablespace-name

表を作成する表スペースを指定します。表スペースは存在している必要があります、REGULAR 表スペースでなければなりません。他の表スペースを指定しない場合、すべての表パーツはこの表スペースに保管されます。この文節を指定しない場合、表は許可 ID によって作成された表スペース中に作成されます。何も検出されない場合、その表はデフォルト表スペースの USERSPACE1 に入れます。USERSPACE1 がドロップされていた場合、表作成は失敗します。

INDEX IN tablespace-name

表に索引を作成する場合の表スペースを指定します。このオプションは、IN 文節に指定された 1 次表スペースが DMS 表スペースである場合にのみ使用できます。指定した表スペースは存在している必要があります、かつ REGULAR または LARGE DMS 表スペースでなければなりません。

注: どの表スペースに索引を含めるかは、表を作成するときのみ指定できます。

insert-column

データの挿入先となる表またはビュー内の列名を指定します。

INSERT

既存の表データを変更することなく、インポートしたデータを表に追加します。

INSERT_UPDATE

インポートしたデータ行をターゲット表に追加するか、または 主キーが一致するものがあればターゲット表の既存行を更新します。

INTO table-name

データのインポート先となるデータベース表を指定します。この表として、システム表、宣言一時表、またはサマリー表は指定できません。

下位のサーバーの場合を除き、INSERT、INSERT_UPDATE、および REPLACE オプションには、完全修飾または非修飾の表名を使わなければならないような

ときでも、別名を指定することができます。修飾子付き表名は、*schema.tablename* の形式です。 *schema* には、表作成時のユーザー名が入りません。

LOBS FROM lob-path

LOB ファイルを保管するパス (複数可) を指定します。LOB データ・ファイルの名前は、メイン・データ・ファイル (ASC、DEL、または IXF) の、LOB 列にロードされる列内に保管されます。lobsinfile 修飾子が指定されていない場合、このオプションは無視されます。

LONG IN tablespace-name

ロング列の値 (LONG VARCHAR、LONG VARGRAPHIC、LOB データ・タイプ、またはソース・タイプとしてこれらが指定されている特殊タイプ) を保管する表スペースを指定します。このオプションは、IN 文節で指定した 1 次表スペースが DMS 表スペースである場合のみ使用できます。指定した表スペースは存在している必要があり、LARGE DMS 表スペースでなければなりません。

MESSAGES message-file

インポート操作中に生じ得る警告およびエラー・メッセージの宛先を指定します。宛先ファイルがすでに存在している場合、インポート・ユーティリティーは情報を追加します。このファイルへの完全パスが指定されていない場合、このユーティリティーは現行のディレクトリーおよびデフォルトのドライブを宛先として使用します。 *message-file* を省略すると、メッセージは標準出力に書き込まれます。

METHOD

L データのインポートを開始する列および終了する列の番号を指定します。列の番号は、データの行の先頭からのバイト単位のオフセットです。この番号は 1 から始まります。

注: このメソッドは、ASC ファイルの場合にのみ使用することができます、そのファイル・タイプに対してのみ有効なオプションです。

N インポートする列の名前を指定します。

注: この方式は、IXF ファイルの場合にのみ使用することができます。

P インポートする入力データ・フィールドの索引 (1 から始まる) を指定します。

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

MODIFIED BY filetype-mod

追加オプションを指定します (1247 ページの表 236 を参照)。

NULL INDICATORS null-indicator-list

このオプションは、METHOD L パラメーターを指定した場合だけ使用できます。つまり、入力ファイルが ASC ファイルの場合です。NULL 標識リストは、コンマで区切られた正の整数のリストで、各 NULL 標識フィールドの列の番号を指定します。列の番号は、データの行の先頭からのバイト単位の、各 NULL 標識フィールドのオフセットです。NULL 標識リストには、METHOD L パラメーターで定義された各データ・フィールドに対する 1 つの項目がなければなりません。列の番号がゼロであることは、対応するデータ・フィールドが必ずデータを含んでいることを示します。

NULL 標識列中の Y の値は、その列データが NULL であることを指定します。NULL 標識列に Y 以外の文字を指定した場合は、列データが NULL ではなく、METHOD L オプションで指定された列データがインポートされることを指定することになります。

NULL 標識文字は MODIFIED BY オプションを使用して変更できます (1247 ページの表 236 の nullindchar 修飾子の説明を参照)。

OF filetype

入力ファイル内のデータの形式を指定します。

- ASC (区切りなし ASCII 形式)
- DEL (区切り文字付き ASCII 形式)。さまざまなデータベース・マネージャーやファイル・マネージャーで使用します。
- WSF (ワークシート形式)。以下のプログラムで使用します。
 - ロータス 1-2-3
 - Lotus Symphony
- IXF (統合交換フォーマット、PC バージョン)。同一のあるいは別の DB2 表からエクスポートされたことを意味します。IXF ファイルには、表定義および既存の索引定義も含まれます。ただし、SELECT ステートメントに列が指定されている場合は除きます。

REPLACE

データ・オブジェクトを切り捨てることによって表内の既存のデータすべてを削除してから、インポートしたデータを挿入します。表定義および索引定義は変更されません。表がない場合は、このオプションを使用できません。

DATALINK 列を含む表では無効です。階層間でデータを移動する際にこのオプションを使用する場合は、階層全体に関係したデータだけが置き換えられます。副表は置き換えられません。

REPLACE_CREATE

表がすでにある場合には、データ・オブジェクトを切り捨てることによって表内の既存のデータすべてを削除し、表定義や索引定義は変えることなく、インポートしたデータを挿入します。

表がまだない場合には、行の内容だけでなく表定義と索引定義も作成します。

このオプションは、IXF ファイルの場合にのみ使用することができます。
 DATALINK 列を含む表では無効です。階層間でデータを移動する際にこのオプションを使用する場合は、階層全体に関係したデータだけが置き換えられません。副表は置き換えられません。

RESTARTCOUNT *n*

$n + 1$ の位置のレコードからインポート操作を開始することを指定します。最初の n レコードはスキップされます。

STARTING *sub-table-name*

階層専用キーワード。 *sub-table-name* から始まるデフォルト順を要求します。PC/IXF ファイルの場合、デフォルト順は入力ファイルに保管されている順です。PC/IXF ファイル形式の場合、デフォルト順は有効な唯一の順序です。

sub-table-list

タイプ表で INSERT または INSERT_UPDATE オプションを指定した場合、データのインポート先副表を指定するために副表名のリストが使われます。

traversal-order-list

タイプ表で INSERT、INSERT_UPDATE、または REPLACE オプションを指定した場合、インポートする階層内の副表の走査順序を指定するために副表名のリストを使います。

UNDER *sub-table-name*

1 つ以上の副表を作成する場合に親表を指定します。

例:

例 1

次に示すのは myfile.ixf から STAFF 表に情報をインポートする方法の例です。

```
db2 import from myfile.ixf of ixf messages msg.txt insert into staff
```

SQL3150N PC/IXF 形式ファイルの H レコードには、製品 "DB2 01.00"、日付 "19970220"、および時刻 "140848" が入っています。

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

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

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

SQL3221W ...COMMIT WORK が開始されました。入力レコード数 = "58"。

SQL3222W ...すべてのデータベース変更のコミットが成功しました。

IMPORT

SQL3149N "58" 行が、入力ファイルから処理されました。"58" 行が表に挿入され、成功しました。"0" 行が拒否されました。

例 2

下記に示す例は、DEL 形式のデータが含まれている入力ファイル delfile1 から、表 MOVIE TABLE をインポートする方法を示す例です。

```
db2 import from delfile1 of del
modified by dldel |
insert into movietable (actorname, description, url_making_of,
url_movie) datalink specification (dl_url_default_prefix
"http://narang"), (dl_url_replace_prefix "http://bomdel"
dl_url_suffix ".mpeg")
```

注:

1. この表には下記の 4 つの列が含まれています。

actorname	VARCHAR(n)
description	VARCHAR(m)
url_making_of	DATALINK (with LINKTYPE URL)
url_movie	DATALINK (with LINKTYPE URL)

2. 入力ファイルの中の DATALINK データのサブフィールド区切り文字は、縦線 (|) 文字です。
3. url_making_of の列値に接頭部文字シーケンスが含まれていないなら、"http://narang" が使用されます。
4. url_movie の非 NULL 列値には、接頭部として "http://bomdel" が付けられます。既存の値は置き換えられます。
5. url_movie の非 NULL 列値のパスには、".mpeg" が付加されます。たとえば、url_movie の列値が "http://server1/x/y/z" なら、それは "http://bomdel/x/y/z.mpeg" として保管されます。その値が "/x/y/z" なら "http://bomdel/x/y/z.mpeg" として保管されます。

例 3 (IDENTITY 列がある表へのインポート)

TABLE1 には以下の 4 つの列があります。

- C1 VARCHAR(30)
- C2 INT GENERATED BY DEFAULT AS IDENTITY
- C3 DECIMAL(7,2)
- C4 CHAR(1)

TABLE2 は TABLE1 と同じですが、C2 が GENERATED ALWAYS IDENTITY 列である点が異なります。

DATAFILE1 のデータ・レコード (DEL 形式):

```
"Liszt"
"Hummel",,187.43, H
"Grieg",100, 66.34, G
"Satie",101, 818.23, I
```

DATAFILE2 のデータ・レコード (DEL 形式):

```
"Liszt", 74.49, A
"Hummel", 0.01, H
"Grieg", 66.34, G
"Satie", 818.23, I
```

以下のコマンドは、DATAFILE1 で行 1 および 2 への ID 値の提供がないので、それらの行のための ID 値を生成します。ただし、行 3 および 4 は、それぞれユーザー提供の ID 値 100 と 101 が割り当てられます。

```
db2 import from datafile1.del of del replace into table1
```

DATAFILE1 を TABLE1 にインポートしてすべての行に対する ID 値を生成するには、以下のコマンドのいずれかを発行します。

```
db2 import from datafile1.del of del method P(1, 3, 4)
replace into table1 (c1, c3, c4)
db2 import from datafile1.del of del modified by identityignore
replace into table1
```

DATAFILE2 を TABLE1 にインポートして各行に対する ID 値を生成するには、以下のコマンドのいずれかを発行します。

```
db2 import from datafile2.del of del replace into table1 (c1, c3, c4)
db2 import from datafile2.del of del modified by identitymissing
replace into table1
```

DATAFILE1 を TABLE2 に、ID 関連のファイル・タイプ修飾子を使用せずにインポートした場合、行 1 と 2 は挿入されますが、行 3 と 4 はリジェクトされます。その理由は、それらが固有の非 NULL を提供し、IDENTITY 列が GENERATED ALWAYS であるからです。

使用上の注意:

インポート操作を開始する前に、すべての表操作が完了し、すべてのロックがペンディング解除になっていることを確認してください。これは、WITH HOLD でオープンされた、すべてのカーソルをクローズした後で COMMIT または ROLLBACK を発行することによって行われます。

インポート・ユーティリティーは、SQL INSERT ステートメントを使ってターゲット表に行を追加します。ユーティリティーは入力ファイル中のデータの行ごとに 1 つの INSERT ステートメントを発行します。INSERT ステートメントが失敗すると、以下のどちらかのアクションが起きます。

IMPORT

- 後続の INSERT ステートメントを正常に実行できると思われる場合、メッセージ・ファイルに警告メッセージが書き込まれ、処理は続行します。
- 後続の INSERT ステートメントが失敗しそうで、データベースの損傷の可能性がある場合、メッセージ・ファイルにエラー・メッセージが書き込まれ、処理は停止します。

ユーティリティは、REPLACE または REPLACE_CREATE 操作中に、古い行が削除された後、自動 COMMIT を実行します。したがって、システムが失敗するか、またはアプリケーションが表オブジェクトの切り捨て後にデータベース・マネージャーに割り込む場合、古いデータはすべて失われます。これらのオプションを使用する前に、古いデータがもう必要ないことを必ず確認してください。

ログが CREATE、REPLACE、または REPLACE_CREATE 操作中にいっぱいになった場合、ユーティリティは挿入されたレコード上で自動 COMMIT を実行します。システムが失敗した場合、またはアプリケーションが自動 COMMIT の後にデータベース・マネージャーに割り込んだ場合、部分データを持つ表がデータベースに残されます。REPLACE または REPLACE_CREATE オプションを使用してインポート操作全体をやり直すか、または正常にインポートされる行の数に設定した RESTARTCOUNT パラメーターを指定して INSERT を使用してください。

デフォルトでは、自動 COMMIT は INSERT または INSERT_UPDATE オプションには実行されません。しかし、COMMITCOUNT パラメーターがゼロでない場合は実行されます。フル・ログの結果は ROLLBACK です。

インポート・ユーティリティが COMMIT を実行するときにはいつでも、2つのメッセージがメッセージ・ファイルに書き込まれます。1つはコミットされるレコードの数を示し、もう1つは正常に終了した COMMIT の後に書き込まれます。失敗の後にインポート操作を再始動する場合、最後に正常に終了した COMMIT から決定されたとおり、スキップするレコードの数を指定してください。

インポート・ユーティリティは、小さい非互換性問題 (たとえば、文字データは埋め込みまたは切り捨てを使ってインポートでき、数値データは異なる数値データ型を使ってインポートできる) は受け入れますが、大きな非互換性に関する問題は受け入れません。

それ自体以外への依存があるオブジェクト表や、基本表に何らかの依存 (それ自体も含めて) があるオブジェクト・ビューを、REPLACE または REPLACE_CREATE することはできません。このような表またはビューを置換するには、以下のようになります。

1. 表が親であるすべての外部キーをドロップします。
2. インポート・ユーティリティを実行します。
3. 表を変更して外部キーを再作成します。

外部キーの再作成中にエラーが発生する場合、参照保全を保守するためにデータを変更してください。

参照制約および外部キー定義は、PC/IXF ファイルから表を作成する場合は保存されません。(主キー定義は、データが前に SELECT * を使ってエクスポートされた場合、保存されます。)

リモート・データベースへのインポートには、入力データ・ファイルのコピー、出力メッセージ・ファイル、およびデータベースがサイズが大きくなる可能性に備えて、十分なディスク・スペースをサーバー上に確保することが必要です。

インポート操作がリモート・データベースに対して実行され、出力メッセージ・ファイルが非常に長い (60KB より長い) 場合、クライアント上でユーザーに戻されるメッセージ・ファイルがインポート操作中に欠落することがあります。メッセージ情報の最初の 30KB と最後の 30KB は、常に保存されます。

リモート・データベースへの PC/IXF ファイルのインポートは、PC/IXF ファイルがディスクセットではなくハード・ディスク上にある場合、より早く行えます。

ASC、DEL、または WSF のファイル形式をインポートするためには、それ以前にデータベース表または階層がすでに存在していなければなりません。しかし、表がまだ存在していない場合でも、IMPORT CREATE または IMPORT REPLACE_CREATE を使えば、PC/IXF ファイルからデータをインポートする際に表が作成されます。タイプ表の場合、IMPORT CREATE によってタイプ階層と表階層も作成されます。

データ (階層データを含む) を別のデータベースに移動するには、PC/IXF インポートを使う必要があります。行区切り文字を含む文字データを区切り文字付き ASCII (DEL) ファイルにエクスポートし、テキスト転送プログラムにより処理を行うと、行区切り文字を含むフィールドは長さが伸縮します。ソースとターゲットのデータベースが両方とも同じクライアントからアクセス可能である場合、ファイルのコピーというステップは必要ありません。

ここでは、ASC ファイルおよび DEL ファイル内のデータは、インポートを実行するクライアント・アプリケーションのコード・ページのデータであると想定します。別々のコード・ページにデータをインポートする場合、PC/IXF ファイル (異なるコード・ページへのインポートが考慮されたファイル) を使用することが推奨されています。

PC/IXF ファイルとインポート・ユーティリティーが同一のコード・ページの場合、処理は通常のアプリケーションと同じようになります。両者が異なるコード・ページであっても、FORCEIN オプションが指定されている場合には、インポート・ユーティリティーにおいて、PC/IXF ファイル内のデータが、インポートを実行するアプリケーションと同一のコード・ページであると見なされます。両者のコード・ページに共通の遷移表がある場合にも、同じ処置になります。両者が異なるコード・ページであり、FORCEIN オプションも指定されていないが、遷移表はある、という場合には、PC/IXF ファイル内のすべてのデータが、ファイル・コード・ページからアプリケーション・コ

ード・ページへと変換されます。両者が異なるコード・ページにあり、FORCEIN オプションも指定されておらず、さらに遷移表もない場合には、インポート操作は失敗してしまいます。これは、DB2 for AIX クライアントの PC/IXF ファイルにのみ適用されません。

列が 1012 個の限界に近づいている 8KB ページ上の表オブジェクトの場合、PC/IXF データ・ファイルをインポートすると、SQL ステートメントの最大サイズを超えているため、DB2 からエラーが戻されることがあります。このようになるのは、CHAR、VARCHAR、または CLOB 型の列の場合だけです。DEL または ASC ファイルのインポートでは、この制限は当てはまりません。PC/IXF ファイルを使って新しい表を作成している場合、別の方法として、*db2look* を使って表を作成した DDL ステートメントをダンプしてから、そのステートメント CLP から発行する、という方法があります。

DB2 Connect を使うと、DB2 for OS/390、DB2 for VM and VSE、および DB2 for OS/400 などの DRDA サーバーにデータをインポートできます。PC/IXF インポート (INSERT オプション) だけがサポートされています。RESTARTCOUNT パラメーターもサポートされていますが、COMMITCOUNT パラメーターはサポートされていません。

タイプ表に対して CREATE オプションを使うと、PC/IXF ファイルの中で定義されているすべての副表が作成されます。副表定義は変更されません。タイプ表に対して CREATE 以外のオプションを使うと、走査順序リストによって、走査順序を指定できません。その場合、走査順序リストはエクスポート操作で使用されたものと一致していなければなりません。PC/IXF 形式の場合は、ターゲット副表名を指定して、ファイルに保管されている走査順序を使うだけです。

インポート・ユーティリティーを使って、前に PC/IXF ファイルにエクスポートされた表をリカバリーできます。表は、エクスポート時の状態に戻ります。

データは、システム表、宣言一時表、またはサマリー表にはインポートできません。

インポート・ユーティリティーを介してビューを作成することはできません。

Windows NT オペレーティング・システムの場合、

- 論理分割 PC/IXF ファイルのインポートはサポートされていません。
- 正しくない形式の PC/IXF または WSF ファイルのインポートはサポートされていません。

DB2 Data Links Managerについての考慮事項

DB2 インポート・ユーティリティーを実行する場合、その前に下記のことを実行しておいてください。

1. 参照されるファイルを、適切な Data Links サーバーにコピーする。 **dlfm_import** ユーティリティを使用することによって、 **dlfm_export** ユーティリティの生成したアーカイブからファイルを取り出すことができます。
2. 必要な接頭部名を DB2 Data Links Manager に登録する。さらに、必要ならデータベースを登録するなど、その他の管理作業も必要になることがあります。
3. 必要なら、DATALINK 列の URL の Data Links サーバー情報を、SQL 表のエクスポート・データから更新します。(元の構成の Data Links サーバーがターゲットにおいても同じなら、その Data Links サーバー名は更新する必要はありません。)分散ファイル・システム (DFS) の場合、必要に応じて、(DATALINK 列の) URL に含まれているセル名の情報を SQL 表のエクスポート・データから更新します。
4. DB2 Data Links Manager 構成ファイルのターゲット構成で、Data Links サーバーを定義する。DFS の場合、DB2 Data Links Manager 構成ファイルのターゲット構成でセルを定義します。

インポート・ユーティリティがターゲット・データベースに対して実行される場合、DATALINK 列データの参照するファイルが、該当する Data Links サーバー上でリンクされます。

挿入操作の間、DATALINK 列の処理では、ターゲット・データベースでの列指定にしたがって、該当する Data Links サーバー中のファイルがリンクされます。

入力ファイル内での DATALINK 情報の表示

表 236. 有効なファイル・タイプ修飾子 (インポート)

修飾子	説明
すべてのファイル形式	
compound=x	<p>x は 1 ~ 100 の数字です。データの挿入に非アトミック複合 SQL を使用し、1 回につき x ステートメントずつ挿入が試みられます。</p> <p>この修飾子が指定され、トランザクション・ログに十分な大きさがいない場合、インポート操作は失敗します。トランザクション・ログは、COMMITCOUNT によって指定された行数か、または COMMITCOUNT が指定されていない場合はデータ・ファイルの行数を入れる十分な大きさが必要です。したがって、トランザクション・ログのオーバーフローを避けるために、COMMITCOUNT オプションを指定することをお勧めします。</p> <p>この修飾子は、INSERT_UPDATE モード、階層表、および修飾子 <code>usedefaults</code>、<code>identitymissing</code>、<code>identityignore</code>、<code>generatedmissing</code>、<code>generatedignore</code> とは互換性がありません。</p>
generatedignore	<p>この修飾子は、インポート・ユーティリティに、すべての生成列のデータはデータ・ファイルに存在するが、それらを見捨てるべきことを知らせます。この結果として、生成列のすべての値は、このユーティリティによって生成されます。この修飾子は、<code>generatedmissing</code> 修飾子と共に使用することはできません。</p>

表 236. 有効なファイル・タイプ修飾子 (インポート) (続き)

修飾子	説明
generatedmissing	この修飾子を指定すると、ユーティリティーは、入力データ・ファイルには生成列のデータが (NULL さえも) なく、したがって各行の値が生成されると想定します。この修飾子は、generatedignore 修飾子と共に使用することはできません。
identityignore	この修飾子は、インポート・ユーティリティーに、IDENTITY 列のデータはデータ・ファイルに存在するが、それらを見捨てるべきことを知らせます。この結果、すべて ID 値はユーティリティーによって生成されます。この動作は、GENERATED ALWAYS および GENERATED BY DEFAULT IDENTITY 列のどちらの場合も同じです。つまり、GENERATED ALWAYS 列の場合、リジェクトされる行はないという意味です。この修飾子は、identitymissing 修飾子と共に使用することはできません。
identitymissing	この修飾子を指定すると、ユーティリティーは、IDENTITY 列のデータが入力データ・ファイルに入っていない (NULL も入っていない) ものと見なし、行ごとに値を生成します。この動作は、GENERATED ALWAYS および GENERATED BY DEFAULT IDENTITY 列のどちらの場合も同じです。この修飾子は、identityignore 修飾子と共に使用することはできません。
lobsinfile	<p><i>lob-path</i> には、LOB データを含むファイルへのパスを指定します。</p> <p>各パスには、データ・ファイル内で LOB ロケーション指定子 (LLS) によって示される 1 つ以上の LOB を含む、少なくとも 1 つのファイルが含まれます。LLS は、LOB ファイル・パスに保管されるファイル内の LOB のロケーションのストリング表記です。LLS の形式は、<i>filename.ext.nnn.mmm/</i> です。ここで、<i>filename.ext</i> は LOB を含むファイルの名前、<i>nnn</i> はファイル内の LOB のオフセット (バイト単位)、<i>mmm</i> は LOB の長さ (バイト単位) です。たとえば、ストリング <i>db2exp.001.123.456/</i> がデータ・ファイルに保存される場合、LOB はファイル <i>db2exp.001</i> のオフセット 123 に位置し、456 バイト長です。</p> <p>LOBS FROM 文節は、“lobsinfile” 修飾子が使用されているときの、LOB ファイルの場所を指定します。LOBS FROM 文節には、“lobsinfile” 修飾子のコンテキスト以外の意味はありません。LOBS FROM 文節は、データのインポート中に、IMPORT ユティリティーに LOB ファイルを検索するためのパスのリストを送ります。</p> <p>NULL LOB を指定するには、サイズに -1 と入力します。サイズを 0 と指定すると、長さが 0 の LOB として扱われます。長さが -1 の NULL LOB の場合、オフセットとファイル名は無視されます。たとえば、NULL LOB の LLS は <i>db2exp.001.7.-1/</i> です。</p>
no_type_id	単一副表へのインポートの場合のみ有効。これを使う場合として典型的な例は、正規の表からデータをエクスポートした後、この修飾子を使ってインポート操作を呼び出してそのデータを単一の副表に変換する場合です。

表 236. 有効なファイル・タイプ修飾子 (インポート) (続き)

修飾子	説明
nodefaults	<p>ターゲット表の列に対応するソース列が明示的に指定されていない場合、その表列が NULL 不可能なら、デフォルトはロードされません。このオプションを指定せず、あるターゲット表列のためのソース列が明示的に指定されていない場合、以下のいずれかになります。</p> <ul style="list-style-type: none"> • 列にデフォルトを指定できる場合、そのデフォルトがロードされます。 • 列が NULL 可能で、デフォルトがその列に指定できない場合、NULL がロードされます。 • 列が NULL 不可能で、デフォルトがその列に指定できない場合、エラーが戻され、ユーティリティは処理を停止します。
usedefaults	<p>ターゲット表の列に対応するソース列が指定されているが、1 つ以上の行インスタンスのデータが入っていない場合、デフォルト値がロードされます。欠落データの例は、以下のとおりです。</p> <ul style="list-style-type: none"> • DEL ファイルの場合、列に「.,」が指定された場合 • ASC ファイルの場合、列の NULL 標識が yes として設定された場合 • DEL/ASC/WSF ファイルの場合、列が不足している行、または元の指定では十分な長さのない行。 <p>このオプションが指定されていない場合、行インスタンスのソース列にデータがないと、以下のいずれかの処理が行われます。</p> <ul style="list-style-type: none"> • 列が NULL 可能な場合、NULL がロードされます。 • 列が NULL 不可能の場合、ユーティリティはその行をリジェクトします。
ASCII ファイル形式 (ASC/DEL)	
codepage= <i>x</i>	<p><i>x</i> は ASCII 文字ストリングです。その値は、出力データ・セット中のデータのコード・ページとして解釈されます。インポート操作中に、文字データをアプリケーション・コード・ページからこのコード・ページに変換します。</p> <p>以下の規則が適用されます。</p> <ul style="list-style-type: none"> • 純 DBCS (グラフィック)、混合 DBCS、および EUC では、区切り文字は x00 ~ x3F の範囲に制限されます。 • nullindchar には、コード・ポイントが x20 ~ x7F の標準 ASCII セットに含まれる記号を指定する必要があります。これは、ASCII シンボルおよびコード・ポイントを示します。 <p>注:</p> <ol style="list-style-type: none"> 1. CODEPAGE 修飾子は、LOBSINFILE 修飾子と共に使用することはできません。 2. コード・ページをアプリケーション・コード・ページからデータベース・コード・ページに変換している途中でデータ拡張が起こると、データが切り捨てられてデータ損失が起こる場合があります。

表 236. 有効なファイル・タイプ修飾子 (インポート) (続き)

修飾子	説明
dateformat="x"	<p>x はソース・ファイルの日付の形式です。 ^a 有効な日付要素は以下のとおりです。</p> <p>YYYY - 年 (0000 ~ 9999 の範囲の 4 桁の数字) M - 月 (1 ~ 12 の範囲の 1 桁または 2 桁の数) MM - 月 (1 ~ 12 の範囲の 2 桁の数。 M とは相互排他的) D - 日 (1 ~ 31 の範囲の 1 桁または 2 桁の数) DD - 日 (1 ~ 31 の範囲の 2 桁の数。 D とは相互排他的) DDD - 日 (年間) (001 ~ 366 の範囲の 3 桁の数。他の日または月要素とは 相互排他的)</p> <p>デフォルト値の 1 が、指定されない各要素に割り当てられます。日付形式の例を以下に示します。</p> <p>"D-M-YYYY" "MM.DD.YYYY" "YYYYDDD"</p>
implieddecimal	<p>暗黙指定されている小数点のロケーションが列定義によって決定され、値の終わりにあるとは見なされなくなります。たとえば、値 12345 は、 12345.00 ではなく、 123.45 として DECIMAL(8,2) 列にロードされます。</p>
noeofchar	<p>オプションのファイル終了文字 'x'1A' は、ファイルの終わりとして認識されません。それが普通の文字であるかのように処理は継続されます。</p>
timeformat="x"	<p>x はソース・ファイルの時刻の形式です。 ^a 有効な時刻要素は以下のとおりです。</p> <p>H - 時 (12 時間制の場合は 0 ~ 12、 24 時間制の場合は 0 ~ 24 の範囲の 1 桁または 2 桁の数) HH - 時 (12 時間制の場合は 0 ~ 12、 24 時間制の場合は 0 ~ 24 の範囲の 2 桁の数。 H とは相互排他的) M - 分 (0 ~ 59 の範囲の 1 桁または 2 桁の数) MM - 分 (0 ~ 59 の範囲の 2 桁の数。 M とは相互排他的) S - 秒 (0 ~ 59 の範囲の 1 桁または 2 桁の数) SS - 秒 (0 ~ 59 の範囲の 2 桁の数。 S とは相互排他的) SSSSS - 夜の 12 時から数えた秒数 (00000 ~ 86399 の範囲の 5 桁の数。 他の時刻要素とは相互排他的) TT - 午前午後の標識 (AM または PM)</p> <p>デフォルト値の 0 が、指定されない各要素に割り当てられます。時刻形式の例を以下に示します。</p> <p>"HH:MM:SS" "HH.MM TT" "SSSSS"</p>

表 236. 有効なファイル・タイプ修飾子 (インポート) (続き)

修飾子	説明
timestampformat="x"	<p>x はソース・ファイルのタイム・スタンプの形式です。 ^a 有効なタイム・スタンプ・エレメントは以下のとおりです。</p> <p>YYYY - 年 (0000 ~ 9999 の範囲の 4 桁の数字)</p> <p>M - 月 (1 ~ 12 の範囲の 1 桁または 2 桁の数)</p> <p>MM - 月 (1 ~ 12 の範囲の 2 桁の数。M (月) とは相互排他的)</p> <p>D - 日 (1 ~ 31 の範囲の 1 桁または 2 桁の数)</p> <p>DD - 日 (1 ~ 31 の範囲の 2 桁の数。D とは相互排他的)</p> <p>DDD - 日 (年間) (001 ~ 366 の範囲の 3 桁の数。他の日または月エレメントとは相互排他的)</p> <p>H - 時 (12 時間制の場合は 0 ~ 12、24 時間制の場合は 0 ~ 24 の範囲の 1 桁または 2 桁の数)</p> <p>HH - 時 (12 時間制の場合は 0 ~ 12、24 時間制の場合は 0 ~ 24 の範囲の 2 桁の数。H とは相互排他的)</p> <p>M - 分 (0 ~ 59 の範囲の 1 桁または 2 桁の数)</p> <p>MM - 分 (0 ~ 59 の範囲の 2 桁の数。M (分) とは相互排他的)</p> <p>S - 秒 (0 ~ 59 の範囲の 1 桁または 2 桁の数)</p> <p>SS - 秒 (0 ~ 59 の範囲の 2 桁の数。S とは相互排他的)</p> <p>SSSSS - 夜の 12 時から数えた秒数 (00000 ~ 86399 の範囲の 5 桁の数。他の時刻エレメントとは相互排他的)</p> <p>UUUUUU - マイクロ秒 (000000 ~ 9999996 の範囲の 6 桁の数)</p> <p>TT - 午前午後の標識 (AM または PM)</p> <p>YYYY、M、MM、D、DD、または DDD エレメントが指定されていない場合、デフォルト値の 1 が割り当てられます。他のエレメントが指定されていない場合には、デフォルト値の 0 が割り当てられます。以下に、タイム・スタンプ形式の例を示します。</p> <p>"YYYY/MM/DD HH:MM:SS.UUUUUU"</p> <p>以下の例では、ユーザー定義の日時形式を含むデータを、schedule という表にインポートする方法を示しています。</p> <pre>db2 import from delfile2 of del modified by timestampformat="yyyy.mm.dd hh:mm tt" insert into schedule</pre>

表 236. 有効なファイル・タイプ修飾子 (インポート) (続き)

修飾子	説明
usegraphiccodepage	<p>usegraphiccodepage が指定された場合、グラフィックまたは 2 バイト文字ラージ・オブジェクト (DBCLOB) データ・フィールドにインポートされるデータは、グラフィック・コード・ページであると見なされます。データの残りは、文字コード・ページであると見なされます。グラフィック・コード・ページは、文字コード・ページと関連付けられます。IMPORT は、指定されている場合の codepage 修飾子、または codepage 修飾子が指定されていない場合は、アプリケーションのコード・ページを通じて、文字コード・ページを決定します。</p> <p>この修飾子は、リカバーされている表にグラフィック・データが含まれている場合にのみ、表リカバリーのドロップによって生成された区切りデータ・ファイルとともに使用される必要があります。</p> <p>制限</p> <p>usegraphiccodepage 修飾子は、EXPORT ユーティリティで作成された DEL または ASC ファイルで指定することはできません。</p> <p>usegraphiccodepage 修飾子はまた、ファイル内の 2 バイト文字ラージ・オブジェクト (DBCLOB) には無視されます。</p>
ASC (区切りなし ASCII) ファイル形式	
nochecklengths	<p>nochecklengths を指定した場合は、ソース・データの中にターゲット表列のサイズを超える列定義が含まれている場合であっても、各行のインポートが試みられます。コード・ページ変換によってソース・データが縮小されれば、そのような行であったとしても正常にインポートすることができます。たとえば、ソースに 4 バイトの EUC データがある場合、それがターゲットで 2 バイトの DBCS データに縮小されれば、必要なスペースは半分で済みます。このオプションが特に役立つのは、列の定義は不一致であるがソース・データが常に適合することが分かっている場合です。</p>
nullindchar= <i>x</i>	<p><i>x</i> は単一文字です。NULL 値を示す文字を <i>x</i> に変更します。 <i>x</i> のデフォルト値は Y です。^b</p> <p>文字が 1 つの英字である場合を除いて、この修飾子は、EBCDIC データ・ファイルで大文字小文字を区別します。たとえば、NULL 標識文字が文字 N として指定されている場合、n も NULL 標識として認識されます。</p>
reclen= <i>x</i>	<p><i>x</i> は 32 767 以下の整数です。各行ごとに <i>x</i> 個の文字が読み取られ、行の終わりを示すのに改行文字は使用されません。</p>

表 236. 有効なファイル・タイプ修飾子 (インポート) (続き)

修飾子	説明
striptblanks	<p>データを可変長フィールドにロードする際に、後書きブランク・スペースを切り捨てます。このオプションを指定しない場合、ブランク・スペースはそのまま保持されます。</p> <p>以下の例の場合、インポート・ユーティリティは、striptblanks によって後書きブランク・スペースを切り捨てます。</p> <pre>db2 import from myfile.asc of asc modified by striptblanks method 1 (1 10, 12 15) messages msgs.txt insert into staff</pre> <p>このオプションは、striptnulls と一緒には指定できません。これらのオプションは、相互に排他的です。</p> <p>注: このオプションは以前の t オプションに代わるもので、バックレベルの互換性のためにのみサポートされています。</p>
striptnulls	<p>データを可変長フィールドにロードする際に、後書き NULL (0x00 文字) を切り捨てます。このオプションを指定しない場合、NULL はそのまま保持されます。</p> <p>このオプションは、striptblanks と一緒には指定できません。これらのオプションは、相互に排他的です。</p> <p>注: このオプションは以前の padwithzero オプションに代わるもので、バックレベルの互換性のためにのみサポートされています。</p>
DEL (区切り付き ASCII) ファイル形式	
chardelx	<p>x は単一文字のストリング区切り文字です。デフォルト値は二重引用符 (") です。指定した文字は、文字ストリングを囲むために、二重引用符の代わりに使用されます。^{bc} 文字ストリング区切り文字として明示的に二重引用符を指定したい場合、以下のように指定します。</p> <pre>modified by chardel'"</pre> <p>単一引用符 (') も文字ストリング区切り文字として指定できます。以下の例では、chardel'' の指定によって、すべての単一引用符 (') が文字ストリング区切り文字として解釈されます。</p> <pre>db2 "import from myfile.del of del modified by chardel'' method p (1, 4) insert into staff (id, years)"</pre>
coldelx	<p>x は単一文字の列区切り文字です。デフォルト値はコンマ (,) です。指定した文字は、列の終わりを表すために、コンマの代わりに使用されます。^{bc}</p> <p>次の例では、coldel; の指定によって、すべてのセミコロン (;) が列区切りとして解釈されます。</p> <pre>db2 import from myfile.del of del modified by coldel; messages msgs.txt insert into staff</pre>
datesiso	日付形式。すべての日付データ値を ISO 形式でインポートします。
decplusblank	正符号文字。正の 10 進値の先頭に正符号 (+) ではなく、ブランク・スペースが置かれます。デフォルトのアクションでは、10 進値に正符号の接頭部が付けられます。

表 236. 有効なファイル・タイプ修飾子 (インポート) (続き)

修飾子	説明
decptx	<p>x は小数点としてピリオドと置換される単一文字です。デフォルト値はピリオド (.) です。指定した文字は、小数点文字としてピリオドの代わりに使用されます。^{bc}</p> <p>以下の例では、decpt; の指定によって、すべてのセミコロン (;) が小数点として解釈されます。</p> <pre>db2 "import from myfile.del of del modified by chardel' decpt; messages msgs.txt insert into staff"</pre>
delprioritychar	<p>区切り文字の現在のデフォルト優先順位は、(1) レコード区切り文字、(2) 文字区切り文字、(3) 列区切り文字です。この修飾子を使用すると、区切り文字の優先順位が (1) 文字区切り文字、(2) レコード区切り文字、(3) 列区切り文字に戻り、以前の優先順位に依存している既存のアプリケーションが保護されます。構文は以下のとおりです。</p> <pre>db2 import ... modified by delprioritychar ...</pre> <p>たとえば、以下のような DEL データ・ファイルがあるとします。</p> <pre>"Smith, Joshua",4000,34.98<row delimiter> "Vincent,<row delimiter>, is a manager", 4005,44.37<row delimiter></pre> <p>delprioritychar 修飾子が指定されている場合、このデータ・ファイルには 2 行しかありません。2 番目の <row delimiter> は 2 番目の行の最初のデータ列の一部と解釈されますが、1 番目と 3 番目の <row delimiter> は実レコードの区切り文字と解釈されます。この修飾子が指定されていないと、このデータ・ファイルは 3 行になり、各行は <row delimiter> によって区切られます。</p>
dldelx	<p>x は単一文字の DATALINK 区切り文字です。デフォルト値はセミコロン (;) です。指定した文字はセミコロンの代わりに、DATALINK 値のフィールド間区切り文字として使用されます。DATALINK 値には副値が複数個含まれる場合があるため、これが必要になります。^{bc}</p> <p>注: x は、行、列、または文字ストリング区切り文字とは異なる文字にしてください。</p>
keepblanks	<p>タイプが CHAR、VARCHAR、LONG VARCHAR、または CLOB の各フィールドの前後の空白を保持します。このオプションを指定しないと、文字区切り文字で囲まれていないすべての前後の空白は除去され、表のすべての空白・フィールドに NULL が挿入されます。</p>
nodoubledel	<p>二重になっている区切り文字の認識を抑制します。</p>
IXF ファイル形式	
forcein	<p>コード・ページが不一致でもデータを受け入れ、コード・ページ間の変換を抑制するようにユーティリティに指示します。</p> <p>固定長ターゲット・フィールドは、データが入るだけの十分な大きさがあるかどうかチェックされます。nochecklengths を指定した場合、そのような検査は実行されず、各行のインポートが試みられます。</p>

表 236. 有効なファイル・タイプ修飾子 (インポート) (続き)

修飾子	説明
indexif	既存の表に現在定義されている索引をすべてドロップし、 PC/IXF ファイルの索引定義に基づいて新しい索引を作成するようにユーティリティに指示します。このオプションを使用できるのは、表の内容を置換する場合だけです。ビューでは使用できません。また、 <i>insert-column</i> を指定した場合には使用できません。
indexschema= <i>schema</i>	指定した <i>schema</i> を、索引作成時の索引名として使用します。 <i>schema</i> を指定しなかった場合 (しかしキーワード <i>indexschema</i> は指定した 場合) には、接続ユーザー ID が使用されます。このキーワードを指定しない場合、IXF ファイルのスキーマが使用されます。
nochecklengths	<i>nochecklengths</i> を指定した場合は、ソース・データの中にターゲット表列のサイズを超える列定義が含まれている場合であっても、各行のインポートが試みられます。コード・ページ変換によってソース・データが縮小されれば、そのような行であったとしても正常にインポートすることができます。たとえば、ソースに 4 バイトの EUC データがある場合、それがターゲットで 2 バイトの DBCS データに縮小されれば、必要なスペースは半分で済みます。このオプションが特に役立つのは、列の定義は不一致であるがソース・データが常に適合することが分かっている場合です。

表 236. 有効なファイル・タイプ修飾子 (インポート) (続き)

修飾子	説明
注:	
<p>1. MODIFIED BY オプションでサポートされていないファイル・タイプが指定されても、インポート・ユーティリティが警告を発することはありません。サポートされていないファイル・タイプを使おうとすると、インポート操作は失敗し、エラー・コードが戻されます。</p>	
<p>^a 日付形式ストリングは必ず二重引用符で囲まなければなりません。フィールド区切り文字には、a ~ z、A ~ Z、および 0 ~ 9 を含めることはできません。フィールド区切り文字は、DEL ファイル形式の文字区切り文字またはフィールド区切り文字と同じにすることはできません。フィールド区切り文字は、エレメントの開始および終了位置が明確な場合は任意指定です。(修飾子によって) D、H、M、または S などのエレメントが使用される場合、項目が可変長であるためにあいまいさが存在することがあります。</p>	
<p>タイム・スタンプ形式の場合、月の記述子と分の記述子のどちらも文字 M を使用するため、区別が明白であるように注意する必要があります。月のフィールドは、他の日付フィールドと隣接していなければなりません。分フィールドは、他の時刻フィールドに隣接していなければなりません。以下に、いくつかのあいまいなタイム・スタンプ形式を示します。</p>	
<p>"M" (月または分のどちらにもとれる) "M:M" (月と分が区別がつかない) "M:YYYY:M" (両方とも月と解釈される) "S:M:YYYY" (時刻値と日付値の両方に隣接している)</p>	
<p>あいまいな形式になっている場合、ユーティリティはエラー・メッセージを報告し、操作は失敗します。</p>	
<p>以下に、明確なタイム・スタンプ形式を示します。</p>	
<p>"M:YYYY" (M (月)) "S:M" (M (分)) "M:YYYY:S:M" (M (月)...M (分)) "M:H:YYYY:M:D" (M (分)...M (月))</p>	
<p>注: 二重引用符や円記号などの文字の前には、エスケープ文字 (たとえば、¥) を付けなければなりません。</p>	
<p>^b この文字は、ソース・データのコード・ページで指定してください。</p>	
<p>文字コード・ポイント (文字記号ではない) は、xJJ または 0xJJ という構文で指定することができます (JJ はコード・ポイントの 16 進表記)。たとえば、列区切りとして # 文字を指定するには、以下のいずれかを使用します。</p>	
<pre>... modified by coldel# modified by coldel0x23 modified by coldelX23 ...</pre>	
<p>^c 区切り文字の制限に、区切り文字のオーバーライドとして使用できる文字に適用される制限のリストが示されています。</p>	

表 237. codepage および usegraphiccodepage 使用時の IMPORT 動作

codepage=N	usegraphiccodepage	IMPORT 動作
なし	なし	ファイル内のすべてのデータは、アプリケーション・コード・ページであると見なされます。

表 237. `codepage` および `usegraphiccodepage` 使用時の `IMPORT` 動作 (続き)

<code>codepage=N</code>	<code>usegraphiccodepage</code>	<code>IMPORT</code> 動作
あり	なし	<p>ファイル内のすべてのデータは、コード・ページ <code>N</code> であると見なされます。</p> <p>警告: <code>N</code> が単一バイト・コード・ページの場合、グラフィック・データをデータベースにインポートすると、壊れます。</p>
なし	あり	<p>ファイル内の文字データは、アプリケーション・コード・ページであると見なされます。グラフィック・データは、アプリケーション・グラフィック・データのコード・ページであると見なされます。</p> <p>アプリケーション・コード・ページが単一バイトの場合は、すべてのデータはアプリケーション・コード・ページであると見なされます。</p> <p>警告:アプリケーション・コード・ページが単一バイトの場合、グラフィック・データは、データベースにグラフィック列が含まれている場合でも、データベースにインポートされると壊れます。</p>
あり	あり	<p>文字データは、コード・ページ <code>N</code> であると見なされます。グラフィック・データは、<code>N</code> のグラフィック・コード・ページであると見なされます。</p> <p><code>N</code> が単一バイトまたは 2 バイト・コード・ページの場合は、すべてのデータは、コード・ページ <code>N</code> であると見なされます。</p> <p>警告: <code>N</code> が単一バイト・コード・ページの場合、グラフィック・データをデータベースにインポートすると、壊れます。</p>

関連資料:

- 1289 ページの『db2look - DB2 統計および DDL 抽出ツール』
- 1149 ページの『LOAD』

RUNSTATS

表およびそれに関連した索引の物理的特性についての統計を更新します。これらの特性には、レコード数、ページ数、および平均レコード長が含まれます。オプティマイザーは、データへのアクセス・パスを判別する際にこれらの統計を使用します。

このユーティリティーは、表が数多く更新される時、または表を再編成した後で、呼び出してください。

有効範囲:

このコマンドは、`db2nodes.cfg` ファイル中のどのデータベース・パーティションからでも発行できます。カタログ・データベース・パーティションのカタログを更新するのに使用します。

このコマンドは、表スペースの呼び出し元のデータベース・パーティションの統計を収集します。表がそのデータベース・パーティションに存在しない場合、データベース・パーティション・グループの最初のデータベース・パーティションが選択されます。

許可:

以下のいずれかが必要です。

- `sysadm`
- `sysctrl`
- `sysmaint`
- `dbadm`
- 表に対する CONTROL 特権
- LOAD 権限

このコマンドを使用するうえでは、接続内に存在する宣言されたグローバル一時表のいずれにおいても明示特権は必要ありません。

必要な接続:

データベース

コマンド構文:


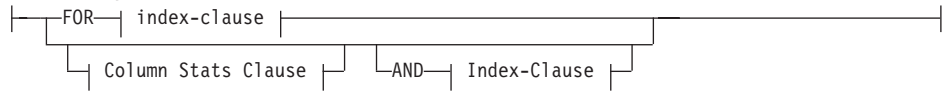
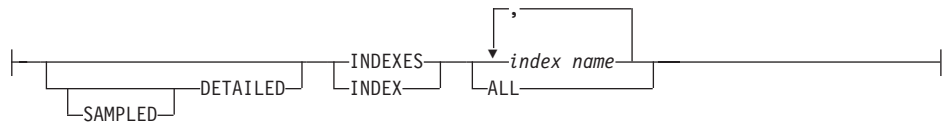
►►—RUNSTATS—ON TABLE—*table name*—



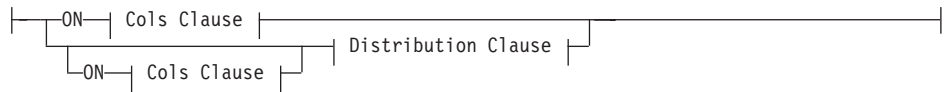
Table Object Options:



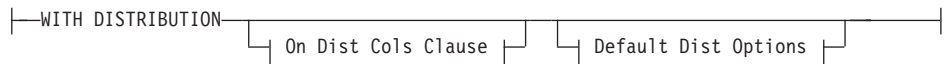
Index Clause:



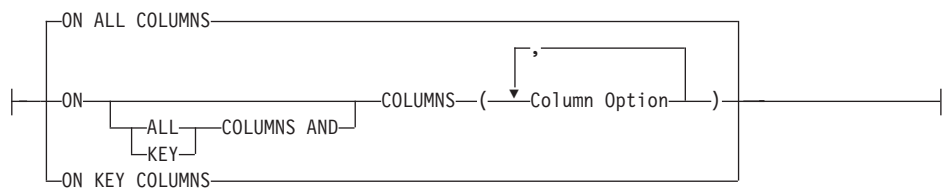
Column Stats Clause:



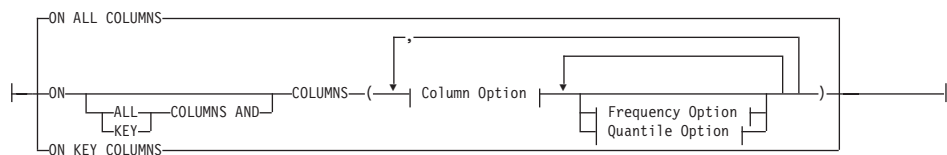
Distribution Clause:



On CoIs Clause:

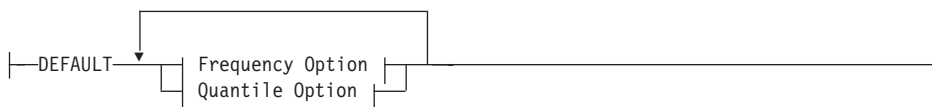


On Dist CoIs Clause:



Default Dist Option:

RUNSTATS



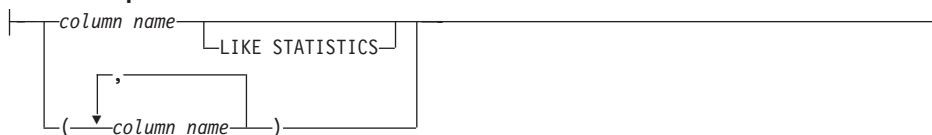
Frequency Option:

┌──NUM_FREQVALUES──integer──┐ ┌──┐

Quantile Option:

┌──NUM_QUANTILES──integer──┐ ┌──┐

Column Option:



コマンド・パラメーター:

table-name

統計が収集される表を識別します。これは、カタログで記述されている表でなければならず、ビューや階層表であってはなりません。型既定表の場合は、`table-name` を表階層のルート表の名前にしてください。名前は、`schema.table-name` 形式の完全修飾名または別名でなければなりません。`schema` には、表作成時のユーザー名が入ります。

index-name

表で定義されている既存の索引を識別します。名前は、`schema.index-name` 形式の完全修飾名を使用してください。

FOR INDEXES

索引のみの統計を収集および更新します。表に、以前に収集された表統計がない場合は、基本表統計も収集されます。これらの基本統計には、分布統計は一切含まれません。

AND INDEXES

表と索引両方を収集および更新します。

DETAILED

拡張された索引統計を計算します。これは `CLUSTERFACTOR` および `PAGE_FETCH_PAIRS` 統計で、比較的大規模な索引の場合に収集されます。

SAMPLED

このオプションは、`DETAILED` オプションと合わせて使用することにより、拡張された索引統計をコンパイルする際に `RUNSTATS` で CPU のサンプリング

技術を使用できるようにします。オプションが指定されていないときは、索引内のすべての項目で、拡張された索引統計を計算するかどうか調べられます。

ON ALL COLUMNS

一部の列で統計の収集を行えますが、収集を行えない列もあります。LONG VARCHAR 列や CLOB 列では統計を収集できません。収集が可能なすべての列で統計を収集することが希望の場合は、ON ALL COLUMNS 文節を使用できます。列は、基本統計の収集に指定する (on-cols-clause) こともできますし、WITH DISTRIBUTION 文節と組み合わせて指定する (on-dist-cols-clause) ことも可能です。これら列指定の文節がどちらも指定されない場合は、デフォルトで ON ALL COLUMNS が指定されます。

なお、on-cols-clause でこれが指定された場合は、特定の列が WITH DISTRIBUTION 文節の一部として選択されない限り、すべての列では基本列統計だけが収集されます。WITH DISTRIBUTION 文節の一部として指定された列では、基本統計も分布統計も収集されます。

WITH DISTRIBUTION ON ALL COLUMNS が指定されている場合は、収集が行えるすべての列で基本統計と分布統計の両方が収集されます。on-cols-clause での指定はすべて重複になるため、必要ありません。

ON COLUMNS

この文節では、統計の収集を行う列のリストを任意に指定できます。列のグループを指定した場合は、そのグループの特色となるいくつかの値が収集されます。リストしない列の統計は消去されます。この文節は、on-cols-clause と on-dist-cols-clause の中で使用できます。

注: 列のグループに関する分布統計の収集は、現在サポートされていません。

ON KEY COLUMNS

特定の列をリストする代わりに、表で定義されたすべての索引を構成する列の統計を収集することもできます。ここでは、照会に含まれる重要な列が、表での索引の作成にも使用されることが前提となっています。表に索引がない場合は、列がリストされず、列統計が収集されない場合と同様になります。この文節は、on-cols-clause または on-dist-cols-clause で使用できます。ただし、WITH DISTRIBUTION 文節で基本統計と分布統計の両方の収集が指定されているため、on-cols-clause と on-dist-cols-clause の両方で指定を行うと、on-cols-clause で重複が生じます。

column-name

column-name は、表内の列の名前でなければなりません。存在しない列が指定された場合や列名の入力を誤った場合など、統計収集を行えない列の名前が指定された場合は、エラー (-205) が戻されます。一方は分布なし、一方は分布ありで、2つの列のリストを指定できます。WITH DISTRIBUTION 文節が関連付けられていないリストで列を指定する場合は、基本列統計だけが収集されま

す。列が両方のリストに含まれている場合は、分布統計が収集されます (NUM_FREQVALUES および NUM_QUANTILES がゼロに設定されていない限り)。

NUM_FREQVALUES

収集の頻度を示す値の最大値を定義します。これは、ON COLUMNS 文節の中で、個々の列ごとに指定できます。個々の列に対して値が指定されない場合は、DEFAULT 文節で指定されている頻度のしきい値が選出されます。値が個々の列でも DEFAULT 文節でも指定されていない場合は、NUM_FREQVALUES データベース構成パラメーターで設定されている値が、収集の頻度を指定する値の最大値になります。

NUM_QUANTILES

収集する分布変位値の最大値を定義します。これは、ON COLUMNS 文節の中で、個々の列ごとに指定できます。個々の列に対して値が指定されない場合は、DEFAULT 文節で指定されている変位値のしきい値が選出されます。値が個々の列でも DEFAULT 文節でも指定されていない場合は、NUM_QUANTILES データベース構成パラメーターで設定されている値が、収集する変位値の最大値になります。

WITH DISTRIBUTION

この文節は、指定された列で基本統計と分布統計の両方を収集することを指定します。ON COLUMNS 文節が指定されていない場合は、表内のすべての列 (CLOB や LONG VARCHAR といった、収集が不可能な列を除く) で分布統計が収集されます。一方 ON COLUMNS 文節が指定されている場合は、指定されたリストにある列でのみ (統計収集を行えない列を除く) 分布統計が収集されます。なお、文節が指定されなければ、基本統計だけが収集されます。

注: 列のグループに関する分布統計の収集は、現在サポートされていません。

WITH DISTRIBUTION ON COLUMNS 文節で列のグループが指定された場合は、分布統計は収集されません。

DEFAULT

DEFAULT NUM_FREQVALUES または NUM_QUANTILES、あるいはその両方が指定された場合、これらの文節は、ON COLUMNS 文節で個々の列に対する指定が行われないうちに、頻度や収集の統計収集を試行する回数を決定します。DEFAULT 文節が指定されない場合は、対応するデータベース構成パラメーターにある値が使用されます。

LIKE STATISTICS

このオプションを指定すると、付加的な列統計が収集されます。収集されるのは、SYSSTAT.COLUMNS の SUB_COUNT および SUB_DELIM_LENGTH 統計です。これらの統計はストリング列に関してのみ収集され、タイプ "column LIKE '%xyz'" および "column LIKE '%xyz%'" の述部に関する選択性の評価を上げるために、照会オプティマイザーで使用されます。

ALLOW WRITE ACCESS

統計が計算される間に、他のユーザーがその表から読み込んだりそこに書き込んだりできることを指定します。

ALLOW READ ACCESS

統計が計算される間に、他のユーザーがその表に対して、読み取り専用のアクセスを行うことができることを指定します。

注: パーティション・データベースでは、RUNSTATS コマンドは、1 つのノードでしか統計を収集できません。RUNSTATS コマンドが実行されたデータベース・パーティションに表のパーティションがある場合、コマンドは、そのデータベース・パーティションで実行されます。それ以外の場合は、表がパーティションに分けられているデータベース・パーティション・グループの最初のデータベース・パーティションで実行されます。

使用上の注意:

注:

1. RUNSTATS コマンドは、以下のような場合に実行することが勧められています。
 - 表が大幅に変更されている場合 (たとえば、多くの変更が行われている場合や、大量のデータが挿入または削除されている場合、あるいは、LOAD 時に統計オプションを指定しないで LOAD が行われた場合など)
 - 表が再編成されている場合 (REORG、REDISTRIBUTE DATABASE PARTITION GROUP を使用)
 - 新しい索引が作成された場合
 - パフォーマンスが重要な意味を持つアプリケーションのバインドの前
 - プリフェッチ数量が変更されたとき
2. オプションの選択は、特定の表やアプリケーションに合わせて行う必要があります。一般的なヒントとして、以下の点を考慮してください。
 - 重要な照会に使用される非常に重要な表、比較的小規模な表、またはあまり変化がなく、システムそのものでの活動があまりない表には、可能な限り詳細に統計を収集する努力を費やす価値があります。
 - 統計を収集する時間が限られており、比較的表が大規模であったりたくさん変更されたりする場合は、述部で使用される一連の列に限って RUNSTATS を実行するのも良い方法かもしれません。このような方法を使用する場合には、より頻繁に RUNSTATS コマンドを実行できるでしょう。
 - 統計を収集する時間が極めて限られており、表ごとに表の RUNSTATS コマンドを調整するのが時間の面で大きな問題となっている場合は、“KEY” 列だけの統計を収集することも考慮してください。索引に含まれている一連の列は、表にとって重要で、述部に使用される確立が最も高いと考えられます。
 - 表に多くの索引があり、それらの索引に含まれる DETAILED (拡張) 情報がアクセス・プランを向上させる可能性がある場合は、統計の収集にかかる時間を減ら

すために、SAMPLED オプションを考慮してください。SAMPLED オプションを使用するかどうかに関係なく、索引での詳細な統計の収集には時間がかかります。それで、これらの統計は、それが照会にとって有益であることが確かな場合以外、収集しないでください。

- 特定の列にスキューがあり、述部のタイプが "column = constant" である場合、その列にはより大きな NUM_FREQVALUES 値を指定するほうが良い可能性があります。
 - 等式の述部で使用される列や、値の分布がスキューされる可能性のある列では、必ず分布統計を収集してください。
 - 範囲の述部を持つ列 (たとえば、"column >= constant"、"column BETWEEN constant1 AND constant2" など) や、タイプ "column LIKE '%xyz'" の列では、より大きな NUM_QUANTILES 値を指定したほうが有益な場合があります。
 - ストレージ・スペースが関係している場合で、統計の収集にあまり時間をかけられない場合は、述部で使用されない列の NUM_FREQVALUES 値や NUM_QUANTILES 値をあまり高くしないでください。
 - 索引の統計が要求され、索引を含む表に対して統計が実行されたことがない場合は、表と索引の両方に関する統計が計算されることに注意してください。
3. コマンドを実行した後は、以下の点に注意してください。
- ロックは、COMMIT を実行すると解除されます。
 - 新規のアクセス・プランを生成できるようにするには、ターゲット表を参照するパッケージを再バインドする必要があります。
 - 表で部分的にコマンドを実行すると、コマンドが最後に実行されてからの表での活動の結果として、不整合が生じる可能性があります。このような場合には、警告メッセージが戻されます。表でだけ RUNSTATS が実行されると、表レベルの統計と索引レベルの統計に不整合が生じます。たとえば、ある表に関して索引レベルの統計を収集した後で、その表からかなりの数の行を削除してしまったとします。このような場合に、その表でだけ RUNSTATS を実行すると、表のカーディナリティーが FIRSTKEYCARD よりも小さくなってしまいます。これが不整合です。これと同様に、作成した新しい索引で統計を収集した場合にも、表レベルの統計に不整合が生じることがあります。
4. RUNSTATS コマンドは、表統計が要求したときに、以前に収集された分散統計をドロップします。たとえば、"RUNSTATS ON TABLE" または "RUNSTATS ON TABLE ... AND INDEXES ALL" は、以前に収集された分散統計がドロップされる原因になります。コマンドが索引でのみ実行される場合、以前に収集された分散保存は保存されます。たとえば、"RUNSTATS ON TABLE ... FOR INDEXES ALL" は以前に収集された分散統計が保存される原因になります。
5. コマンド構文の "On Dist Cols" 文節では、列のグループに対する "Frequency Option" または "Quantile Option" パラメーターの使用は、現在サポートされていません。これらのオプションは、単一の列でのみサポートされています。

6. DMS モードでの作業中に、計算できない 3 つのプリフェッチ統計があります。索引カタログ内の索引統計で、以下の統計に "-1" 値があります。

- AVERAGE_SEQUENCE_FETCH_PAGES
- AVERAGE_SEQUENCE_FETCH_GAP
- AVERAGE_RANDOM_FETCH_PAGES

例:

表でのみ統計を収集し、どの列でも分布統計は収集しません。

```
RUNSTATS ON TABLE db2user.employee
```

表でのみ統計を収集し、列 empid と empname で分布統計を収集します。

```
RUNSTATS ON TABLE db2user.employee
  WITH DISTRIBUTION ON COLUMNS (empid, empname)
```

表でのみ統計を収集し、構成の設定から NUM_QUANTILES を取得した場合に表に指定される頻度のしきい値を使用して、すべての列で分布統計を収集します。

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

一連の索引で統計を収集します。

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

すべての索引のみの基本統計を収集します。

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

詳細な索引統計収集のサンプリングを使用して、表とすべての索引に関する基本統計を収集します。

```
RUNSTATS ON TABLE db2user.employee AND SAMPLED DETAILED INDEXES ALL
```

表の統計を収集し、列 empid、empname、empdept、および索引 Xempid および Xempname では分布統計を収集します。empdept に関しては個別に分布統計のしきい値を設定し、その他の 2 つの列には共通のデフォルトを使用します。

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

索引で使用されるすべての列と、すべての索引の統計を収集します。

```
RUNSTATS ON TABLE db2user.employee ON KEY COLUMNS AND INDEXES ALL
```

すべての索引とすべての列で統計を収集します。分布統計は、1 つの列を除いて収集しません。T1 には列 c1、c2、... c8 が含まれていると考えてください。

RUNSTATS

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

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

表 T1 で、個別の列 c1 および c5 と、列の組み合わせ (c2, c3) および (c2, c4) に関する統計を収集します。複数列のカーディナリティーは、照会オプティマイザーが、データの相関がある列の述部のフィルター係数を見積もるのに大変便利です。

```
RUNSTATS ON TABLE db2user.T1 ON COLUMNS (c1, (c2, c3), (c2, c4), c5)
```

表 T1 で、個別の列 c1 および c2 に関する統計を収集します。列 c1 に関しては、LIKE 述部の統計も収集します。

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

UNQUIESCE

保守またはその他の理由で静止状態になっていたインスタンスまたはデータベースに対するユーザー・アクセスをリストアします。UNQUIESCE は、シャットダウンしたりデータベースを再開したりせずにユーザー・アクセスをリストアします。

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

有効範囲:

UNQUIESCE DB は、静止データベース内のすべてのオブジェクトに対するユーザー・アクセスをリストアします。

UNQUIESCE INSTANCE *instance-name* は、インスタンス *instance-name* 内のインスタンスおよびデータベースに対するユーザー・アクセスをリストアします。

インスタンスを停止した後、そのインスタンスとそのすべてのデータベースの静止を解除するには、*db2stop* コマンドを実行します。DB2 を停止し、再開すると、すべてのインスタンスとデータベースの静止が解除されます。

許可:

以下のいずれかが必要です。

データベース・レベルの静止解除の場合:

- *sysadm*
- *dbadm*

インスタンス・レベルの静止解除の場合:

- *sysadm*
- *sysctrl*

コマンド構文:

```
▶—UNQUIESCE—DB—┬──────────────────────────────────────────────────────────────────────────────────────────▶
                  │
                  └─INSTANCE—instance-name─┘
```

必要な接続:

データベース

(インスタンスの静止解除にはデータベース接続は必須ではありません。)

コマンド・パラメーター:

DB データベースの静止解除。データベース内のすべてのオブジェクトに対するユーザー・アクセスがリストアされます。

INSTANCE *instance-name*

インスタンス *instance-name* およびインスタンス内のデータベースに対するアクセスがリストアされます。

例:

データベースの静止解除

```
db2 unquiesce db
```

このコマンドは、以前に静止されていたデータベースの静止を解除します。

RESTORE DATABASE

DB2 バックアップ・ユーティリティを使用してバックアップされた損傷のある、または破壊されたデータベースを再作成します。リストアされたデータベースは、バックアップ・コピーの作成時と同じ状態になります。このユーティリティはまた、別のイメージを持つデータベースも上書きできます。(さらに、新規データベースのリストアも可能です。)

異なるワークステーション・プラットフォームに作成されたデータベースのコピーを作成するには、`db2move` ユーティリティを使用します。異なるプラットフォームとは、異なるオペレーティング・システムまたはワード・サイズ (つまり、32 ビットまたは 64 ビット) のいずれか、または両方を意味します。32 ビット Windows で作成されたデータベースを、64 ビット Windows プラットフォームにリストアできますが、その逆も可能です。それぞれ 32 ビットまたは 64 ビットの AIX、HP、および Sun オペレー

RESTORE DATABASE

ティング・システムに作成されたデータベース・バックアップは、それぞれ 32 ビットまたは 64 ビットの AIX、HP、および Sun オペレーティング・システムにリストアできます。32 ビット Linux で作成されたデータベースを、64 ビット Linux プラットフォームにリストアすることができますが、その逆も可能です。

このユーティリティーは、DB2 の前の 2 つのバージョンによって生成されたバックアップ・イメージをリストアするためにも使用できます。移行が必要な場合、これはリストア操作の終了時に自動的に起動されます。

バックアップ操作のときに、データベースがすべてのロールフォワード・リカバリーに対して使用可能である場合、リストア操作が正常に完了した後に、ロールフォワード・ユーティリティーを起動することによって、データベースを損傷または破壊が起きる前の状態に戻すことができます。

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

オペレーティング・システムまたはワード・サイズ (32 ビットまたは 64 ビット) に差がある場合は、増分および差分イメージはリストアできません。

1 つの環境から異なる環境への成功したリストアに続いては、非増分バックアップが取られるまで、増分または差分バックアップはできません。(これは、同じ環境でのリストア後の制限ではありません。)

1 つの環境から異なる環境への成功したリストアでも、いくつかの考慮事項があります。すなわち、使用する前に、パッケージを再バインドする必要があります (BIND コマンド、REBIND コマンド、または db2rbind ユーティリティーを使用)。SQL プロシージャをドロップし、再作成する必要があります。そしてすべての外部ライブラリーは、新規プラットフォーム上で再ビルドされる必要があります。(これらは、同じ環境にリストアする場合の考慮事項ではありません。)

有効範囲:

このコマンドは、それが実行されたノードに対してだけ影響を与えます。

許可:

既存のデータベースにリストアするには、以下のいずれかが必要です。

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

新規データベースにリストアするには、以下のいずれかが必要です。

- *sysadm*

- *sysctrl*

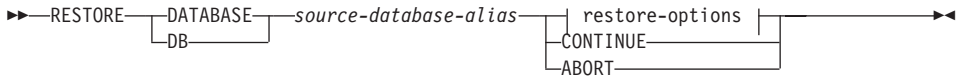
必要な接続:

必要な接続は、実行したいリストア・アクションのタイプによって異なります。

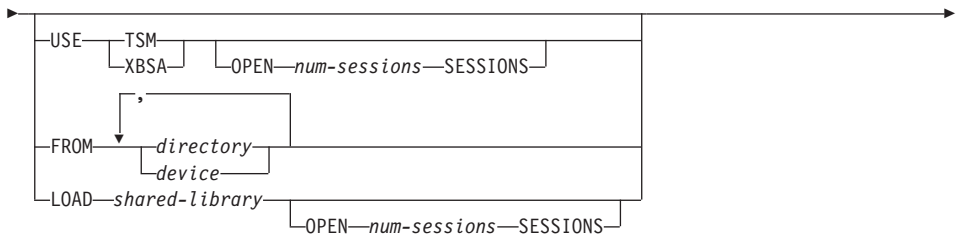
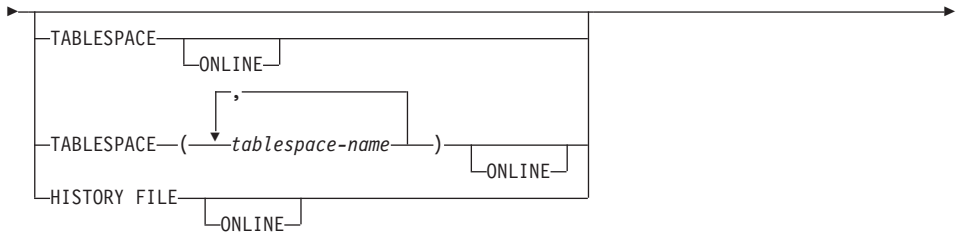
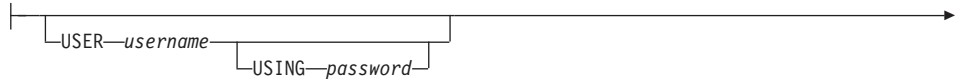
- データベース (既存のデータベースにリストアする場合)。このコマンドは、指定されたデータベースへの排他的接続を自動的に確立します。
- インスタンスおよびデータベース (新規データベースにリストアする場合)。データベースを作成するには、インスタンス・アタッチが必要です。

現行のインスタンスとは異なるインスタンスで新規のデータベースへのリストアを行うには、まず、新規のデータベースを存在させるインスタンスにアタッチする必要があります。新規インスタンスはローカルまたはリモートの可能性があります。現行インスタンスは **DB2INSTANCE** 環境変数の値によって定義されます。

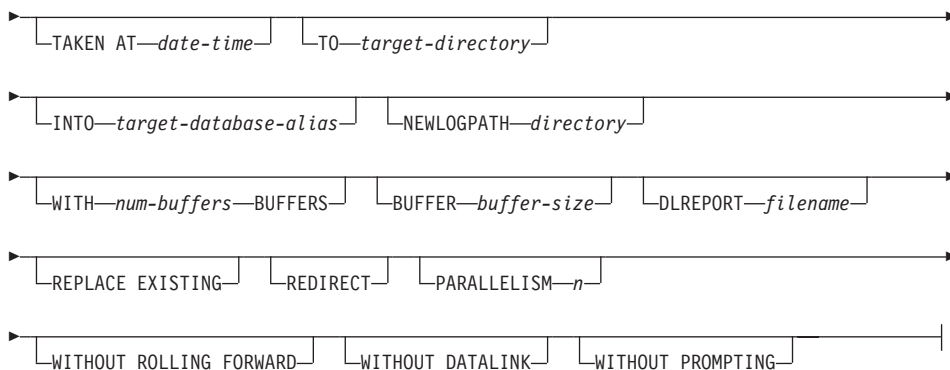
コマンド構文:



restore-options:



RESTORE DATABASE



コマンド・パラメーター:

DATABASE *source-database-alias*

バックアップが取得されるソース・データベースの別名です。

CONTINUE

コンテナが再定義されていること、およびリダイレクトしたりストア操作の最終ステップを実行する必要があることを指定します。

ABORT

このパラメーターは以下を指定します。

- リダイレクトしたりストア操作を停止します。これは、1 つ以上のステップを繰り返す必要があるエラーが発生したときに便利です。ABORT オプションを指定して RESTORE DATABASE を発行した後、REDIRECT オプションを指定した RESTORE DATABASE を含む、リダイレクトしたりストア操作の各ステップを繰り返す必要があります。
- 完了する前に増分リストア操作を終了します。

USER *username*

データベースがリストアされる際のユーザー名を識別します。

USING *password*

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

TABLESPACE *tablespace-name*

リストアされる表スペースを指定するとき使用する名前のレストランです。

ONLINE

このキーワードは、表スペース・レベルのリストア操作を行う場合のみ適用でき、これを指定するとオンラインでバックアップ・イメージがリストアできません。これは、他のエージェントが、バックアップ・イメージのリストア中にデータベースに接続できることや、指定された表スペースのリストア中に他の表スペースのデータを使用できることを意味します。

HISTORY FILE

このキーワードは、バックアップ・イメージから履歴ファイルのみをリストアするのに指定されます。

INCREMENTAL

INCREMENTAL は、追加のパラメーターを使用しないで手動累積リストア操作を指定します。手動リストアの際、ユーザーはリストアに含まれるイメージごとに各リストア・コマンドを手動で実行する必要があります。以下の順序でこれを行ってください。最後、1 番目、2 番目、以下同様に最後のイメージまで。

INCREMENTAL AUTOMATIC/AUTO

自動累積リストア操作を指定します。

INCREMENTAL ABORT

手動累積リストア操作を指定します。

USE TSM

データベースが TSM 管理の出力からリストアされるように指定します。

OPEN num-sessions SESSIONS

TSM またはベンダー製品とともに使用する入出力セッションの数を指定します。

USE XBSA

XBSA インターフェースを使用するように指定します。バックアップ・サービス API (XBSA) は、バックアップやアーカイブの目的でデータ・ストレージ管理を必要とするアプリケーションまたは機能のための、オープン・アプリケーション・プログラミング・インターフェースです。Legato NetWorker は、現在 XBSA インターフェースをサポートしているストレージ・マネージャーです。

FROM directory/device

バックアップ・イメージがあるディレクトリーまたは装置の完全修飾パス名。USE TSM、FROM、および LOAD が省略される場合には、デフォルト値はクライアント・マシン上の現行作業ディレクトリーです。このターゲット・ディレクトリーまたは装置は、データベース・サーバー上に存在している必要があります。

Windows オペレーティング・システムでは、DB2 が生成するディレクトリーを指定してはなりません。たとえば、以下のようなコマンドを実行するとします。

```
db2 backup database sample to c:%backup
db2 restore database sample from c:%backup
```

これらのコマンドを使用して、DB2 は、c:\backup ディレクトリーの下にサブディレクトリーを生成します。これによって、1 つ以上のバックアップを、指定された最上位ディレクトリーに置けるようになります。DB2 生成のサブデ

RESTORE DATABASE

イレクトリーは無視される必要があります。リストアするバックアップ・イメージを正確に指定するためには、`TAKEN AT` パラメーターを使用します。複数のバックアップ・イメージを同じパスに保管することもできます。

複数の項目が指定され、項目の最後がテープ装置である場合には、他のテープが要求されます。有効な応答オプションは、以下のとおりです。

- c** 続行。警告メッセージを生成した装置の使用を続けます (たとえば、新しいテープをマウントしたときなど)。
- d** 装置の終了。警告メッセージの原因となった装置の使用だけを停止します (たとえば、これ以上テープがない場合など)。
- t** 終了。ユーザーが、ユーティリティーによって要求された何らかのアクションを実行しなかった場合、リストア操作を異常終了します。

LOAD shared-library

使用するバックアップおよびリストア I/O 関数を含む共有ライブラリー (Windows オペレーティング・システムでは DLL) の名前。名前には絶対パスを含めることができます。絶対パスを指定しない場合、ユーザー出口プログラムが置かれているパスがデフォルト値として使われます。

TAKEN AT date-time

データベース・バックアップ・イメージのタイム・スタンプです。タイム・スタンプはバックアップ操作が正常に終了した後に表示され、バックアップ・イメージのパス名の一部になっています。 `yyymmddhhmmss` という形式で指定されます。タイム・スタンプを部分的に指定することもできます。たとえば、`20021001010101` と `20021002010101` というタイム・スタンプの 2 つの異なるバックアップ・イメージが存在する場合は、`20021002` を指定すると、タイム・スタンプが `20021002010101` のバックアップ・イメージが使用されます。このパラメーターに値を指定しない場合は、ソース・メディア上のバックアップ・イメージは 1 つだけでなければなりません。

TO target-directory

ターゲット・データベース・ディレクトリーです。ユーティリティーが存在するデータベースへリストアしている場合には、このパラメーターは無視されません。指定するドライブおよびディレクトリーは、ローカルのものでなければなりません。

注: Windows オペレーティング・システムでは、このパラメーターを使用するときにドライブ名を指定してください。たとえば、特定のパスをリストアする場合には `x:%path_name`、またはパスを指定する必要がない場合には `x:` を指定できます。パス名が長すぎる場合、エラーが戻されます。

INTO target-database-alias

ターゲット・データベースの別名です。ターゲット・データベースが存在しない場合には、作成されます。

データベース・バックアップを既存のデータベースにリストアするとき、リストアされたデータベースは既存のデータベースの別名およびデータベース名を継承します。データベース・バックアップを存在していないデータベースにリストアするとき、新規のデータベースが指定した別名およびデータベース名を使用して作成されます。新規のデータベース名は、リストア先のシステムで固有のものでなければなりません。

NEWLOGPATH directory

リストア操作後にアクティブ・ログに使用されるディレクトリーの絶対パス名。このパラメーターの機能は、データベース構成パラメーター *newlogpath* と同じです。ただし、*newlogpath* の影響は、それが指定されたリストア操作に限定されます。このパラメーターは、バックアップ・イメージのログ・パスが、リストア後の使用に適していない場合に使用することができます。たとえば、パスがもはや有効でない、または別のデータベースによって使用されている、という場合などです。

WITH num-buffers BUFFERS

使用するバッファの数です。デフォルト値は 2 です。複数のソースが読み取られる場合や、PARALLELISM の値が増えた場合は、パフォーマンスを向上させるために多数のバッファを使用することができます。

BUFFER buffer-size

リストア操作に使用するバッファのサイズ (ページ数)。このパラメーターの最小値は 8 ページです。デフォルトは 1024 ページです。

リストア・バッファ・サイズは、バックアップ操作中に指定したバックアップ・バッファ・サイズに正の整数を乗算したサイズでなければなりません。誤ったバッファ・サイズを指定すると、許容可能な最小のサイズで割り振られます。

DLREPORT filename

ファイル名を指定する場合は、絶対パスとして指定する必要があります。リストア操作中に高速調整が行われたためにリンク解除されたファイルを報告します。このオプションが使用されるのは、リストアする表に DATALINK 列タイプとリンク・ファイルが含まれている場合だけです。

REPLACE EXISTING

ターゲット・データベースの別名と同じ別名を持つデータベースがすでに存在している場合、このパラメーターは、リストア・ユーティリティーが既存のデータベースをリストアしたデータベースに置換するように指定します。これはリストア・ユーティリティーを起動するスクリプトで便利です。コマンド行プロセッサは、ユーザーに既存のデータベースの削除を検証するよう求めるプロンプトを出さないためです。WITHOUT PROMPTING パラメーターが指定された場合、REPLACE EXISTING を指定する必要はありませんが、その場合、ユーザー介入を標準的に必要とするイベントが起こるとこの操作は失敗します。

RESTORE DATABASE

REDIRECT

リダイレクトしたリストア操作を指定します。リダイレクトしたリストア操作を完了するには、このコマンドの後に 1 つまたは複数の SET TABLESPACE CONTAINERS コマンドを続け、次に CONTINUE オプションを指定して RESTORE DATABASE コマンドを続ける必要があります。

注: 同一のリダイレクトしたリストア操作に関連したコマンドはすべて、同じウィンドウまたは CLP セッションから起動する必要があります。

WITHOUT ROLLING FORWARD

データベースを、正常にリストアされた後ロールフォワード保留状態にしないように指定します。

正常なリストアに続いて、データベースがロールフォワード・ペンディング状態にある場合には、データベースが使用できるようになる前に、ROLLFORWARD コマンドを起動する必要があります。

WITHOUT DATALINK

DATALINK 列を持つ任意の表を DataLink_Reconcile_Pending (DRP) 状態にし、リンクされたファイルの調整を実行しないように指定します。

PARALLELISM n

リストア操作中に作成されるバッファー・マネジュラーの数を指定します。デフォルト値は 1 です。

WITHOUT PROMPTING

リストア操作を無人で実行するように指定します。通常はユーザー介入を必要とするアクションでは、エラー・メッセージが戻されます。テープやディスクなどの取り外し可能メディア装置を使用している場合、このオプションを指定していても、その装置が終わるとプロンプトが出されます。

例:

以下の例で、データベース WSDB は 0 ~ 3 の番号が付けられた 4 つのパーティションすべてに定義されています。パス /dev3/backup はすべてのパーティションからアクセスできます。以下のオフライン・バックアップ・イメージは、/dev3/backup から入手可能です。

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

最初にカタログ・パーティションをリストアしてから WSDB データベースの他のすべてのデータベース・パーティションを the /dev3/backup ディレクトリーからリストアするには、データベース・パーティションの 1 つから以下のコマンドを出します。

```

db2_all '<<+0< db2 RESTORE DATABASE wsdb FROM /dev3/backup
TAKEN AT 20020331234149
      INTO wsdb REPLACE EXISTING'
db2_all '<<+1< db2 RESTORE DATABASE wsdb FROM /dev3/backup
TAKEN AT 20020331234427
      INTO wsdb REPLACE EXISTING'
db2_all '<<+2< db2 RESTORE DATABASE wsdb FROM /dev3/backup
TAKEN AT 20020331234828
      INTO wsdb REPLACE EXISTING'
db2_all '<<+3< db2 RESTORE DATABASE wsdb FROM /dev3/backup
TAKEN AT 20020331235235
      INTO wsdb REPLACE EXISTING'

```

db2_all ユーティリティーは、指定された各データベース・パーティションにリストア・コマンドを出します。

以下は、別名が MYDB であるデータベースの典型的なリダイレクト・リストアのシナリオです。

1. 以下のように、REDIRECT オプションを指定して RESTORE DATABASE コマンドを実行する。

```
db2 restore db mydb replace existing redirect
```

ステップ 1 が正常終了した後でステップ 3 が完了する前に、次を発行してリストア操作を打ち切ることができる。

```
db2 restore db mydb abort
```

2. 再定義する必要があるコンテナを持つ表スペースごとに、SET TABLESPACE CONTAINERS コマンドを実行する。たとえば、

```
db2 set tablespace containers for 5 using
(file 'f:¥ts3con1' 20000, file 'f:¥ts3con2' 20000)
```

リストアしたデータベースのコンテナが、このステップで指定したものであることを検査するために、LIST TABLESPACE CONTAINERS コマンドを実行する。

3. ステップ 1 および 2 が正常終了した後、次を発行する。

```
db2 restore db mydb continue
```

これはリダイレクト・リストア操作の最終ステップです。

4. ステップ 3 が失敗した場合、またはリストア操作を打ち切った場合、リダイレクト・リストアはステップ 1 から再始動できる。

以下は、リカバリー可能データベース用の増分バックアップの週間予定のサンプルです。週 1 回の全データベース・バックアップ操作、1 日 1 回の非累積 (差分) バックアップ操作、および週 2 回の累積 (増分) バックアップ操作が含まれています。

```

(Sun) backup db mydb use tsm
(Mon) backup db mydb online incremental delta use tsm
(Tue) backup db mydb online incremental delta use tsm

```

RESTORE DATABASE

```
(Wed) backup db mydb online incremental use tsm
(Thu) backup db mydb online incremental delta use tsm
(Fri) backup db mydb online incremental delta use tsm
(Sat) backup db mydb online incremental use tsm
```

金曜日の午前中に作成されたイメージを自動データベース・リストアするには、次のようにします。

```
restore db mydb incremental automatic taken at (Fri)
```

金曜日の午前中に作成されたイメージを手動データベース・リストアするには、次のようにします。

```
restore db mydb incremental taken at (Fri)
restore db mydb incremental taken at (Sun)
restore db mydb incremental taken at (Wed)
restore db mydb incremental taken at (Thu)
restore db mydb incremental taken at (Fri)
```

使用上の注意:

db2 restore db <name> という形式のすべての RESTORE DATABASE コマンドは、リストアされるイメージがデータベース・イメージか表スペース・イメージかにかかわらずなく、全データベース・リストアを実行します。 db2 restore db <name> tablespace という形式のすべての RESTORE DATABASE コマンドは、イメージ内で検出される表スペースの表スペース・リストアを実行します。表スペースのリストが提供されるすべての RESTORE DATABASE コマンドは、明示的にリストされているすべての表スペースのリストアを実行します。

オンライン・バックアップのリストア方法にしたがって、すべてのロールフォワード・リカバリーを実行する必要があります。

関連資料:

- コマンド・リファレンス の『BACKUP DATABASE コマンド』
- コマンド・リファレンス の『ROLLFORWARD DATABASE コマンド』
- コマンド・リファレンス の『db2move - データベース移動ツール・コマンド』

UPDATE DATABASE CONFIGURATION

特定のデータベース構成ファイルの中の個々の項目を修正します。

データベース構成ファイルは、データベースが作成されたノードすべてに常駐していません。

有効範囲:

このコマンドは、それが実行されたノードに対してだけ影響を与えます。

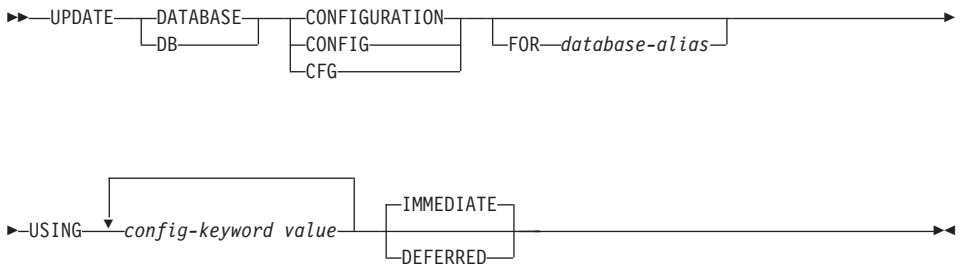
許可:

以下のいずれかが必要です。

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

必要な接続:

インスタンス。明示的なアタッチは必要ありませんが、データベースがアクティブになっているときは、データベース接続が勧められています。データベースがリモートとして示されている場合、リモート・ノードへのインスタンス・アタッチはコマンドの持続期間の間、ずっと確立されたままになります。パラメーターをオンラインにするには、データベースへの接続が必要です。

コマンド構文:**コマンド・パラメーター:****DEFERRED**

構成ファイルでのみ変更を行います。したがって、加えられた変更は、次にデータベースが再活動化されるときに有効になります。

FOR database-alias

構成を更新するデータベースの別名を指定します。データベース接続がすでに確立されている場合は、データベース別名を指定する必要はありません。

IMMEDIATE

データベースが稼動している場合に、即時に変更を行います。IMMEDIATEはデフォルトのアクションですが、このアクションを実行するためにはデータベース接続が有効でなければなりません。

USING config-keyword value

更新するデータベース構成パラメーターを指定します。

使用上の注意:

データベース構成パラメーターのリストを表示または印刷するには、GET DATABASE CONFIGURATION コマンドを使用してください。

UPDATE DATABASE CONFIGURATION

データベース構成パラメーターを推奨されているデータベース・マネージャーのデフォルトにリセットするには、`RESET DATABASE CONFIGURATION` コマンドを使用してください。

データベース構成パラメーターを変更するには、`UPDATE DATABASE CONFIGURATION` コマンドを使用してください。

たとえば、ロギング・モードを `"archival logging"` に、`"zellmart"` というデータベースを含む単一パーティション・データベース環境で変更する場合は、次を使用します。

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

`logretain` 構成パラメーターが変更されたことをチェックするには、次を使用します。

```
db2 get db cfg for zellmart
```

複数パーティション・データベース環境で構成パラメーターを変更するには、`db2_all` コマンドを使用してください。`db2_all` コマンドを使用すると、すべてのパーティションに対して更新が発行されます。

たとえば、ロギング・モードを `"archival logging"` に、`"zellmart"` というデータベースを含む複数パーティション・データベース環境で変更する場合は、次を使用します。

```
db2_all ";db2 update db cfg for zellmart using logretain recovery"
```

`logretain` 構成パラメーターが、すべてのデータベース・パーティションで変更されたことをチェックするには、次を使用します。

```
db2_all ";db2 get db cfg for zellmart"
```

UNIX オペレーティング・システムで作業を行っており、`"grep"` コマンドがある場合は、以下のコマンドを使用して、`logretain` 値のみを表示できます。

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

DB2 構成パラメーターと、各種データベース・ノードに使用できる値についての詳細は、個々の構成パラメーターの説明を参照してください。これらのパラメーターの値は、構成するデータベース・ノードの各タイプ (サーバー、クライアント、またはリモート・クライアントを持つサーバー) によって異なります。

すべてのパラメーターを更新できるわけではありません。

データベース構成ファイルへの変更の一部は、ファイルがメモリーにロードされた後のみ有効になります。これを行う前にすべてのアプリケーションはデータベースから切断されている必要があります。オンラインで構成できるパラメーターと構成できないパラメーターについては、構成パラメーターの一覧をご覧ください。

たとえば、`sales` データベースの `sortheap` データベース構成パラメーターをオンラインで変更するには、以下のようなコマンドを入力します。

```
db2 connect to sales
db2 update db cfg using sortheap 1000
db2 connect reset
```

エラーが発生した場合、データベース構成ファイルは変更されません。チェックサムが無効な場合、データベース構成ファイルは更新できません。適当なコマンドを使用しないでデータベース構成ファイルを変更するとエラーが発生します。エラーが発生する場合、データベースをリストアしてデータベース構成ファイルをリセットする必要があります。

関連タスク:

- 管理ガイド: パフォーマンス の『構成パラメーターによる DB2 の構成』

関連資料:

- コマンド・リファレンス の『GET DATABASE CONFIGURATION コマンド』
- コマンド・リファレンス の『RESET DATABASE CONFIGURATION コマンド』
- 管理ガイド: パフォーマンス の『構成パラメーターの要約』

START DATABASE MANAGER

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

有効範囲:

複数パーティション・データベース環境では、このコマンドは、`$HOME/sql1lib/db2nodes.cfg` ファイルにリストされているすべてのデータベース・パーティションに影響を与えます。ただし、`dbpartitionnum` パラメーターが使用されていない場合に限りです。

許可:

以下のいずれかが必要です。

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

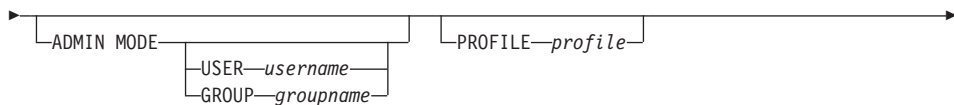
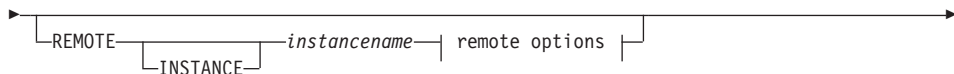
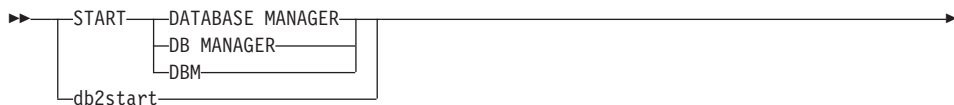
注: `ADD DBPARTITIONNUM` 開始オプションは、`sysadm` または `sysctrl` 権限のどちらかを必要とします。

必要な接続:

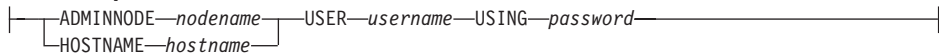
なし

START DATABASE MANAGER

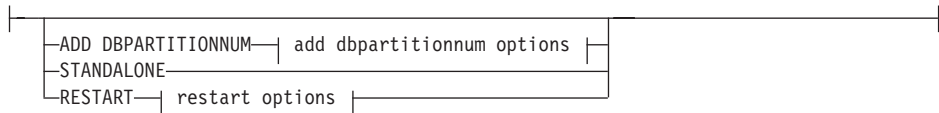
コマンド構文:



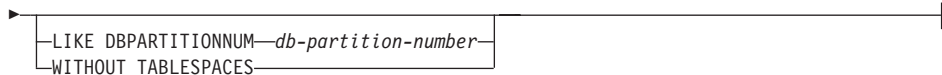
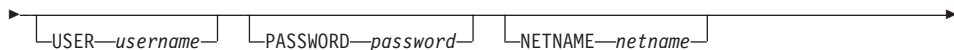
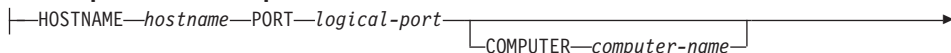
remote options:



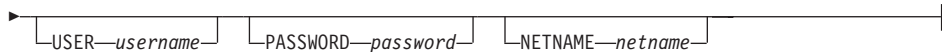
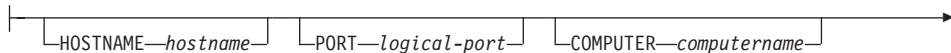
start options:



add dbpartitionnum options:



restart options:



コマンド・パラメーター:

REMOTE [INSTANCE] instancename

開始したいリモート・インスタンスの名前を指定します。

ADMINNODE nodename

REMOTE または REMOTE INSTANCE では、管理ノードの名前を指定します。

HOSTNAME hostname

REMOTE または REMOTE INSTANCE では、ホスト・ノードの名前を指定します。

USER username

REMOTE または REMOTE INSTANCE では、ユーザーの名前を指定します。

USING password

REMOTE または REMOTE INSTANCE、および USER では、ユーザーのパスワードを指定します。

ADMIN MODE

管理の目的のために、インスタンスを静止モードで開始します。これは、インスタンスがすでに『活動中』でない場合以外は、QUIESCE INSTANCE コマンドと同じで、接続を強制クローズする必要はありません。

USER username

ADMIN MODE では、ユーザーの名前を指定します。

GROUP groupname

ADMIN MODE では、グループの名前を指定します。

注: 以下のすべてのパラメーターは、Enterprise Server Edition (ESE) 環境でのみ有効です。

PROFILE profile

DB2 環境を定義するために、各データベース・パーティションで実行しなければならぬプロファイル・ファイル名を指定します。このファイルは、データベース・パーティションの開始前に実行されます。プロファイル・ファイルはインスタンス所有者の sqllib ディレクトリーに常駐していなければなりません。

注: プロファイル・ファイルの環境変数をすべてユーザー・セッションで定義する必要はありません。

DBPARTITIONNUM db-partition-number

開始するデータベース・パーティションを指定します。他のオプションが指定されていない場合、このデータベース・パーティションで通常の開始が行われます。

START DATABASE MANAGER

有効な値は、0 ~ 999 (0 および 999 を含む) です。ADD DBPARTITIONNUM が指定されていない場合、値はインスタンス所有者の db2nodes.cfg ファイルにすでに存在していなければなりません。データベース・パーティション番号が指定されていない場合、構成ファイルで定義されているすべてのデータベース・パーティションが開始されます。

ADD DBPARTITIONNUM

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

hostname および *logical-port* の組み合わせが固有であることを確認してください。

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

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

HOSTNAME *hostname*

ADD DBPARTITIONNUM を使用して、db2nodes.cfg ファイルに追加するホスト名を指定します。

PORT *logical-port*

ADD DBPARTITIONNUM を使用して、db2nodes.cfg ファイルに追加する論理ポートを指定します。有効な値は 0 ~ 999 です。

COMPUTER *computername*

新しいデータベース・パーティションが作成されるマシンのコンピューター名。このパラメーターは、Windows NT では必須ですが、その他のオペレーティング・システムでは無視されます。

USER *username*

新しいデータベース・パーティション上のアカウントのユーザー名。このパラメーターは、Windows NT では必須ですが、その他のオペレーティング・システムでは無視されます。

PASSWORD *password*

新しいデータベース・パーティション上のアカウントのパスワード。このパラメーターは、Windows NT では必須ですが、その他のオペレーティング・システムでは無視されます。

NETNAME netname

db2nodes.cfg ファイルに追加する *netname* を指定します。指定されていない場合、*hostname* で指定された値がデフォルトとなります。

LIKE DBPARTITIONNUM db-partition-number

システム TEMPORARY 表スペース用のコンテナが、インスタンス内の各データベース用に指定した *db-partition-number* のコンテナと同一になるように指定します。指定するデータベース・パーティションは、db2nodes.cfg ファイル中にすでに指定してあるデータベース・パーティションでなければなりません。

WITHOUT TABLESPACES

システム TEMPORARY 表スペースのコンテナがどのデータベースに対しても作成されないことを指定します。データベースを使用する前に、ALTER TABLESPACE ステートメントを使用して、SYSTEM TEMPORARY 表スペース・コンテナを各データベースに追加する必要があります。

STANDALONE

データベース・パーティションが独立方式で開始されることを指定します。FCM は他のデータベース・パーティションとの接続を確立しようとはしません。このオプションはデータベース・パーティションの追加の時に使用します。

RESTART

障害発生後、データベース・マネージャーを起動します。他のデータベース・パーティションの操作は続いており、このデータベース・パーティションは他のデータベース・パーティションとの接続を試みます。*hostname* と *logical-port* のどちらのパラメーターも指定されないと、データベース・マネージャーは、db2nodes.cfg で指定された *hostname* および *logical-port* 値を用いて再始動されます。どちらかのパラメーターが指定されている場合は、接続の確立時に新しい値が他のデータベース・パーティションに送信されます。db2nodes.cfg ファイルは、この情報に基づいて更新されます。

HOSTNAME hostname

RESTART を使用して、データベース・パーティション構成ファイルにあるホスト名を上書きするために使用するホスト名を指定します。

PORT logical-port

RESTART を使用して、データベース・パーティション構成ファイルにある論理ポート番号を上書きするために使用する論理ポート番号を指定します。指定されていない場合、db2nodes.cfg ファイルの *num* の値に対応する、*logical-port* の値がデフォルトとなります。有効な値は 0 ～ 999 です。

COMPUTER computername

新しいデータベース・パーティションが作成されるマシンのコンピュ

START DATABASE MANAGER

ーター名。このパラメーターは、Windows NT では必須ですが、その他のオペレーティング・システムでは無視されます。

USER username

新しいデータベース・パーティション上のアカウントのユーザー名。このパラメーターは、Windows NT では必須ですが、その他のオペレーティング・システムでは無視されます。

PASSWORD password

新しいデータベース・パーティション上のアカウントのパスワード。このパラメーターは、Windows NT では必須ですが、その他のオペレーティング・システムでは無視されます。

NETNAME netname

db2nodes.cfg ファイルで指定されたネット名を上書きするために使用される *netname* を指定します。指定されていない場合、db2nodes.cfg ファイルの *db-partition-number* の値に対応する、*netname* の値がデフォルトとなります。

例:

次に示すのは、データベース・パーティション 10、20、および 30 を使用する 3 データベース・パーティション・システムで発行された **db2start** からの出力例です。

```
04-07-1997 10:33:05    10    0    SQL1063N  DB2START processing was successful.
04-07-1997 10:33:07    20    0    SQL1063N  DB2START processing was successful.
04-07-1997 10:33:07    30    0    SQL1063N  DB2START processing was successful.
SQL1063N  DB2START processing was successful.
```

使用上の注意:

このコマンドをクライアント・ノードで発行しない場合もあります。旧クライアントとの互換性が提供されていますが、データベース・マネージャーには何も影響ありません。

一度開始されると、データベース・マネージャーのインスタンスは、たとえそれを使用していたすべてのアプリケーション・プログラムが終了したとしても、ユーザーが停止しない限り稼働しています。

データベース・マネージャーが正常に開始されると、正常終了メッセージが標準出力装置に送られます。エラーが発生すると、処理は停止され、エラー・メッセージが標準出力装置に送られます。パーティション・データベース環境では、メッセージは **START DATABASE MANAGER** コマンドを実行したデータベース・パーティションに戻されません。

パーティション・データベース環境でパラメーターが指定されていない場合、データベース・パーティション構成ファイルで指定されたパラメーターを使用してすべて並列ノードでデータベース・マネージャーが開始されます。

START DATABASE MANAGER コマンド実行中の場合、データベースへの要求を発行する前に、適用可能なデータベース・パーティションが開始されていることを確認してください。

db2cshrc ファイルはサポートされておらず、環境の定義付けに使用できません。

インスタンスを静止状態で開始できます。これは、以下のいずれかを使用して行うことができます。

```
db2start admin mode
```

または

```
db2start admin mode user username
```

または

```
db2start admin mode groupname
```

UNIX プラットフォームでは、START DATABASE MANAGER コマンドは SIGINT および SIGALRM 信号をサポートしています。CTRL+C を押すと、SIGINT 信号が発行されます。データベース・マネージャー構成パラメーターの *start_stop_time* の指定値が着信すると、SIGALRM 信号が発行されます。いずれかの信号が発行されると、データベース・パーティションの開始操作に割り込みが生じ、メッセージ (SIGINT の場合は SQL1044N、SIGALRM の場合は SQL6037N) が割り込みが生じたデータベース・パーティションから、`$HOME/sqllib/log/db2start.timestamp.log` エラー・ログ・ファイルに戻されます。すでに開始済みのデータベース・パーティションには影響がありません。開始しているデータベース・パーティションに対して CTRL+C が押された場合、そのデータベース・パーティションを再始動する前に、そのデータベース・パーティションに対して **db2stop** が発行されなければなりません。

Windows NT オペレーティング・システムの場合、開始することを失敗した通信サブシステムがあったとしても、**db2start** コマンドも **NET START** コマンドも警告を戻しません。Windows NT 環境のデータベース・マネージャーは NT サービスとしてインプリメントされます。サービスが正常に開始された場合にはエラーを戻しません。NT イベント・ログまたは DB2DIAG.LOG ファイルを調べて、**db2start** の実行中にエラーが発生しなかったか確認してください。

互換性:

バージョン 8 より前のバージョンとの互換性:

- キーワード LIKE NODE は、LIKE DBPARTITIONNUM に置き換えられます。

START DATABASE MANAGER

- キーワード `ADDNODE` は、`ADD DBPARTITIONNUM` に置き換えられます。
- キーワード `NODENUM` は、`DBPARTITIONNUM` に置き換えられます。

関連資料:

- コマンド・リファレンスの『STOP DATABASE MANAGER コマンド』
- コマンド・リファレンスの『ADD DBPARTITIONNUM コマンド』

db2advis - DB2 索引アドバイザー

1 つ以上の SQL ステートメントに作成する索引についてユーザーにアドバイスします。関連 SQL ステートメントのグループは、ワークロードと呼ばれます。ユーザーは、ワークロード中の各ステートメントの重要性をランク付けし、ワークロード中の各ステートメントが実行される頻度を指定することができます。各表ごとに推奨される索引、それらに応じて派生する統計、およびそれぞれを作成する DDL は、ユーザー作成の表 `ADVISE_INDEX` に書き込まれます。

注: 構造型列は、このコマンドの実行時には考慮されません。

許可:

データベースへの読み取りアクセス。 `Explain` 表への読み取りおよび書き込みアクセス。

必要な接続:

なし。このコマンドは、データベース接続を確立します。

コマンド構文:

```
db2advis -d database-name [-w workload-name] [-s "statement"] [-i filename] [-g] [-a userid [/passwd]] [-l disk-limit] [-t max-advise-time] [-h] [-p] [-o outfile]
```

コマンド・パラメーター:

-d database-name

接続の確立先のデータベースの名前を指定します。

-w workload-name

索引がアドバイスされるワークロードの名前を指定します。この名前は `ADVISE_WORKLOAD` 表で使用されます。

-s "statement"

索引がアドバイスされる単一の SQL ステートメントのテキストを指定します。ステートメントは必ず二重引用符で囲んでください。

-i filename

1 つ以上の SQL ステートメントが入っている入力ファイルの名前を指定します。デフォルトは標準入力です。注釈テキストは、各行の先頭に 2 つのハイフンを付けて -- <注釈> で表します。ステートメントは必ずセミコロンで区切ってください。

ワークロード中の各ステートメントが実行される頻度は、以下の行を入力ファイルに挿入することによって変更できます。

```
--#SET FREQUENCY <x>
```

頻度は、ファイル中何回でも更新できます。

-g

動的 SQL スナップショットからの SQL ステートメントの検索を指定します。-p コマンド・パラメーターと結合されている場合は、SQL ステートメントは ADVISE_WORKLOAD 表に保持されます。

-a userid/passwd

データベースへの接続に使用する名前およびパスワード。パスワードが指定される場合、斜線 (/) を含めなければなりません。

-l disk-limit

既存のスキーマですべての索引に使用可能な最大 MB を指定します。デフォルトは、パーティション (64 GB) ごとの索引の最大サイズにおける、データベース・マネージャー限度です。

-t max-advise-time

最大許可時間 (分) を指定し、操作を完了します。デフォルトは 10 です。無制限の時間は、ゼロの値によって指定されます。

-h

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

-p

Explain 表でツールを実行した際に生成されたプランを保持します。

-o outfile

推奨されたオブジェクトを作成するためのスクリプトを *outfile* に保管します。

例:

以下の例では、ユーティリティは PROTOTYPE データベースに接続し、解決策において制限なしで ADDRESSES 表に索引を推奨します。

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

db2advis - DB2 索引アドバイザー

以下の例では、ユーティリティーは PROTOTYPE データベースに接続し、ワークロード名が "production" に等しい ADVISE_WORKLOAD 表の照会で、53 MB を超えない索引を推奨します。 解決策を見つけるための最大許可時間は 20 分です。

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

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

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

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

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

使用上の注意:

動的 SQL ステートメントの場合、ステートメントが実行される頻度は、以下のようにモニターから獲得されます。

1. 以下のように発行します。

```
db2 reset monitor for database <database-alias>
```

適切な時間間隔で待機します。

2. 以下のように発行します。

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

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

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

db2fs - 最初のステップ

「最初のステップ」の GUI を立ち上げて、ユーザーの学習と DB2 の使用を開始するのに必要な機能を開始します。

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

許可:

sysadm

コマンド構文:

▶—db2fs—

コマンド・パラメーター:

なし

db2look - DB2 統計および DDL 抽出ツール

要求された DDL ステートメントを抽出して、テスト・データベース上の実動データベースのデータベース・オブジェクトを再生成します。このツールを使用すると、テスト・データベース内のオブジェクトの統計を複製するために使用される必須 UPDATE ステートメントを生成するだけでなく、更新データベース構成パラメーターと更新データベース・マネージャー構成パラメーター、および **db2set** ステートメントも生成して、テスト・データベースのレジストリー変数および構成パラメーター設定を、実動データベースの設定に適合させることができます。

テスト・システムに実動システムのデータのサブセットを含めておくと、便利なことが多くあります。しかし、そのようなテスト・システム用に選択したアクセス・プランが、必ずしも実動システム用に選択したアクセス・プランと同じであるとは限りません。テスト・システム用のカタログ統計と構成パラメーターの両方が、実動システムのものと一致するように更新されていなければなりません。このツールを使用すると、アクセス・プランが、実動システムで使用するものと類似しているテスト・データベースを作成することが可能になります。

許可:

システム・カタログに対する SELECT 特権

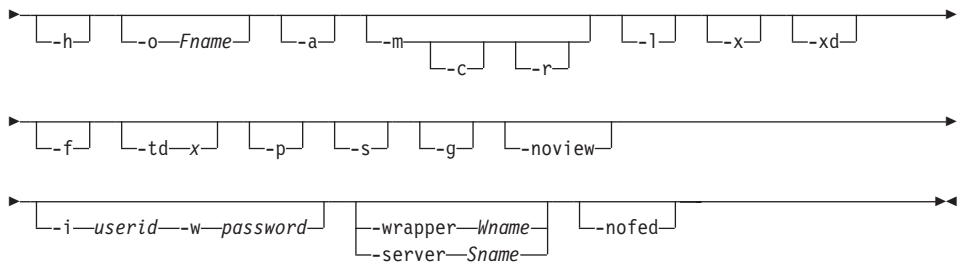
必要な接続:

なし。このコマンドは、データベース接続を確立します。

コマンド構文:

▶—db2look—d—DBname—
 —e— —u—Creator— —z—schema—
 —t— Tname—

db2look - DB2 統計および DDL 抽出ツール



コマンド・パラメーター:

-d *DBname*

照会する実動データベースの別名。 *DBname* は、DB2 UDB for UNIX、Windows、または DB2 Universal Database for OS/390 and z/OS データベースの名前にすることができます。 *DBname* が DB2 Universal Database for OS/390 and z/OS データベースの場合には、 **db2look** ユーティリティーは、OS/390 オブジェクト用の DDL および UPDATE 統計ステートメントを抽出します。これらの DDL および UPDATE 統計ステートメントは、DB2 UDB データベースには適用できますが、DB2 for OS/390 データベースには適用できません。これは、OS/390 オブジェクトを抽出して、それらを DB2 UDB データベースで再作成しようとするユーザーに役立ちます。

DBname が OS/390 データベースの場合、 **db2look** 出力は以下のものに制限されます。

- 表、索引、ビュー、およびユーザー定義特殊タイプ用の DDL の生成
- 表、列、列配布および索引用の UPDATE 統計ステートメントの生成

-e データベース・オブジェクト用の DDL ステートメントを抽出します。 **-e** オプションを使用する場合には、以下のデータベース・オブジェクト用の DDL を抽出します。

- 表
- ビュー
- 自動サマリー表 (AST)
- 別名
- 索引
- トリガー
- シーケンス
- ユーザー定義特殊タイプ
- 主キー、RI、および CHECK 制約
- ユーザー定義構造タイプ
- ユーザー定義関数
- ユーザー定義方式

- ユーザー定義変換
- ラッパー
- サーバー
- ユーザー・マッピング
- ニックネーム
- タイプ・マッピング
- 関数テンプレート
- 関数マッピング
- 索引の指定

注: **db2look** によって生成される DDL を使用して、ユーザー定義関数を正常に再作成することができます。ただし、ユーザー定義関数が使用可能であるためには、特定のユーザー定義関数 (**EXTERNAL NAME** 文節など) が参照するユーザー・ソース・コードが使用可能でなければなりません。

-u *Creator*

作成者 ID。出力をこの作成者 ID があるオブジェクトだけに制限します。オプション **-a** を指定した場合、このパラメーターは無視されます。 **-u** と **-a** のどちらも指定しない場合には、環境変数 **USER** が使用されます。

-z *schema*

スキーマ名を指定します。出力をこのスキーマ名のオブジェクトに制限します。オプション **-a** を指定した場合、このパラメーターは無視されます。このパラメーターが指定されない場合は、すべてのスキーマ名のオブジェクトが抽出されます。このオプションは、OS/390 データベースには適用されません。このオプションは、連合 DDL では無視されます。

-t *Tname1 Tname2 ... TnameN*

表名のリストです。表のリストにある特定の表への出力を制限します。表の最大数は 30 です。表名はブランク・スペースで区切られます。大文字と小文字を区別する表名や、名前の中にスペースを含む表名は、`¥" My Table ¥"` のように、円記号と二重引用符で囲む必要があります。

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

-o *Fname*

LaTeX 形式を使用する場合、*filename.tex* に出力を書き込みます。プレーン・テキスト形式を使用する場合、*filename.txt* に出力を書き込みます。そうでない場合は、出力を *filename.sql* に書き込みます。このオプションを指定しない場合、出力は標準出力に書き込まれます。ファイル名が拡張子付きで指定されている場合、出力はそのファイルに書き込まれます。

-a このオプションが指定されている場合には、特定の作成者 ID で作成されたオブジェクトだけに出力が制限されることはありません。すべてのユーザーによ

って作成されたすべてのオブジェクトが対象になります。たとえば、このオプションと `-e` オプションが共に指定される場合、データベース内のすべてのオブジェクト用の DDL ステートメントが抽出されます。このオプションと `-m` オプションが共に指定される場合、データベース内のすべてのユーザー作成表および索引用の UPDATE 統計ステートメントが抽出されます。

注: `-u` と `-a` のどちらも指定しない場合には、環境変数 **USER** が使用されます。UNIX ベースのシステムでは、この変数を明示的に設定する必要はありません。しかし Windows NT の場合、**USER** 環境変数にデフォルトがありません。このプラットフォームでは、**SYSTEM** 変数の中のユーザー変数を設定するか、または `set USER=<username>` をセッションに発行する必要があります。

-m 必要な UPDATE ステートメントを生成して、表、列、および索引についての統計を複製します。 `-p`、`-g`、および `-s` オプションは、`-m` オプションが指定された場合は無視されます。 `-c` オプションおよび `-r` オプションは、オプションで `-m` と共に使用されます。

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

-r このオプションと `-m` オプションを共に指定する場合には、**db2look** は RUNSTATS コマンドを生成しません。デフォルト・アクションでは、RUNSTATS コマンドを生成します。 `-m` オプションを指定しない場合、`-r` オプションは無視されます。

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

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

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

db2look がサポートする許可には、以下のものが含まれます。

- 表: ALTER、SELECT、INSERT、DELETE、UPDATE、INDEX、REFERENCE、CONTROL
- ビュー: SELECT、INSERT、DELETE、UPDATE、CONTROL
- 索引: CONTROL

- スキーマ: CREATEIN、DROPIN、ALTERIN
- データベース: CREATEDB、BINDADD、CONNECT、CREWATE_NOT_FENCED、IMPLICIT_SCHEMA
- ユーザー定義関数 (UDF): EXECUTE
- ユーザー定義メソッド: EXECUTE
- ストアド・プロシージャ: EXECUTE
- パッケージ: CONTROL、BIND、EXECUTE

-xd このオプションを指定すると、オブジェクトの作成時に SYSIBM によって権限を付与されたオブジェクトの権限 DDL を含むすべての権限 DDL が、db2look ユーティリティーによって生成されます。

-f このオプションは、構成パラメーターおよびレジストリー変数を抽出するために使用します。

注: DB2 照会オプティマイザーに影響を与える構成パラメーターおよびレジストリー変数だけが抽出されます。

db2look ユーティリティーは、以下の構成パラメーター用の更新ステートメントを生成します。

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

db2look ユーティリティーは、以下の DB2 レジストリー変数用の *db2set* コマンドを生成します。

- DB2_PRED_FACTORIZE
- DB2_CORRELATED_PREDICATES

db2look - DB2 統計および DDL 抽出ツール

- DB2_LIKE_VARCHAR
- DB2_SORT_AFTER_TQ
- DB2_HASH_JOIN
- DB2_ORDERED_NLJN
- DB2_NEW_CORR_SQ_FF
- DB2_PART_INNER_JOIN
- DB2_INTERESTING_KEYS

-td x **db2look** によって生成される SQL ステートメントのステートメント区切り文字を指定します。このオプションが指定されていない場合のデフォルトはセミコロン (;) です。このオプションは、**-e** オプションを指定した場合に使用することをお勧めします。この場合、抽出されたオブジェクトにはトリガーまたは SQL ルーチンが含まれる可能性があります。

-p プレーン・テキスト形式を使用します。

-s ポストスクリプト・ファイルを生成します。

注:

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

-g 索引の取り出しページ・ペアを示すためにグラフを使用します。

注:

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

-noview

このオプションを指定すると、CREATE VIEW DDL ステートメントが抽出されません。

-i userid

リモート・データベースで作業する場合には、このオプションを使用してください。

-w password

-i オプションと共にこのパラメーターを使用すると、リモート・システムに常駐するデータベースに対して **db2look** が実行可能になります。 **db2look** では、リモート・システムにログオンするために、ユーザー ID およびパスワードが使用されます。

注: リモート・データベースで作業を行っている場合は、リモート・データベースは、ローカル・データベースと同じバージョンである必要があります。 **db2look** ユーティリティーには、下位レベルまたは上位レベル・サポートはありません。

-wrapper *Wname*

このラッパーに適用する連合オブジェクト用の DDL を生成します。生成される可能性のある連合 DDL には、以下のものが含まれます: CREATE WRAPPER、CREATE SERVER、CREATE USER MAPPING、CREATE NICKNAME、CREATE TYPE MAPPING、CREATE FUNCTION ... AS TEMPLATE、CREATE FUNCTION MAPPING、CREATE INDEX SPECIFICATION、および GRANT (ニックネーム、サーバー、索引への特権) 1 つのラッパー名のみがサポートされています。1 より少ない、または複数指定された場合は、エラーが返されます。このオプションは、**-server** オプションが使用されている場合は使用できません。

-server *Sname*

このサーバーに適用する連合オブジェクト用の DDL を生成します。生成される可能性のある連合 DDL には、以下のものが含まれます: CREATE WRAPPER、CREATE SERVER、CREATE USER MAPPING、CREATE NICKNAME、CREATE TYPE MAPPING、CREATE FUNCTION ... AS TEMPLATE、CREATE FUNCTION MAPPING、CREATE INDEX SPECIFICATION、および GRANT (ニックネーム、サーバー、索引への特権) 1 つのサーバー名のみがサポートされています。1 より少ない、または複数指定された場合は、エラーが返されます。このオプションは、**-wrapper** オプションが使用されている場合は使用できません。

-nofed 連合 DDL が生成されないことを指定します。このオプションが指定された場合、**-wrapper Wname** および **server Sname** オプションは無視されます。

例:

データベース DEPARTMENT でユーザー walid によって作成されたオブジェクト用の DDL ステートメントを生成します。 **db2look** の出力は、以下のようにしてファイル db2look.sql に送信します。

```
db2look -d department -u walid -e -o db2look.sql
```

データベース DEPARTMENT でユーザー walid によって作成された、ianhe というスキーマ名のオブジェクト用に DDL ステートメントを生成します。 **db2look** の出力は、以下のようにしてファイル db2look.sql に送信します。

```
db2look -d department -u walid -z ianhe -e -o db2look.sql
```

UPDATE ステートメントを生成して、データベース DEPARTMENT でユーザー walid によって作成された表および索引の統計を複製します。出力は、以下のようにしてファイル db2look.sql に送信します。

db2look - DB2 統計および DDL 抽出ツール

```
db2look -d department -u walid -m -o db2look.sql
```

ユーザー walid および UPDATE ステートメントによって作成されたオブジェクト用の DDL ステートメントの両方を生成して、同じユーザーによって作成された表および索引についての統計を複製します。 **db2look** の出力は、以下のようにしてファイル db2look.sql に送信します。

```
db2look -d department -u walid -e -m -o db2look.sql
```

データベース DEPARTMENT ですべてのユーザーによって作成されたオブジェクトの DDL ステートメントを生成します。 **db2look** の出力は、以下のようにしてファイル db2look.sql に送信します。

```
db2look -d department -a -e -o db2look.sql
```

すべてのユーザー定義のデータベース・パーティション・グループ、バッファー・プール、および表スペース用の DDL ステートメントを生成します。 **db2look** の出力は、以下のようにしてファイル db2look.sql に送信します。

```
db2look -d department -l -o db2look.sql
```

データベースおよびデータベース・マネージャー構成パラメーター用の UPDATE ステートメント、およびデータベース DEPARTMENT にあるレジストリー変数用の **db2set** ステートメントを生成します。 **db2look** の出力は、以下のようにしてファイル db2look.sql に送信します。

```
db2look -d department -f -o db2look.sql
```

データベース DEPARTMENT にあるすべてのオブジェクト用の DDL、データベース DEPARTMENT にあるすべての表および索引についての統計を複製するための UPDATE ステートメント、 GRANT 許可ステートメント、データベースおよびデータベース・マネージャー構成パラメーター用の UPDATE ステートメント、レジストリー変数用の **db2set** ステートメント、およびデータベース DEPARTMENT にあるすべてのユーザー定義のデータベース・パーティション・グループ、バッファー・プール、および表スペース用の DDL を生成します。出力は、以下のようにしてファイル db2look.sql に送信します。

```
db2look -d department -a -e -m -l -x -f -o db2look.sql
```

オリジナルの作成者によって作成されたオブジェクトも含む、データベース DEPARTMENT 内のすべてのオブジェクトのすべての許可 DDL を生成します。(この場合には、オブジェクトの作成時に SYSIBM によって許可が付与されました。) **db2look** の出力は、以下のようにしてファイル db2look.sql に送信します。

```
db2look -d department -xd -o db2look.sql
```

データベース DEPARTMENT ですべてのユーザーによって作成されたオブジェクトの DDL ステートメントを生成します。 **db2look** の出力は、以下のようにしてファイル db2look.sql に送信します。

```
db2look -d department -a -e -td % -o db2look.sql
db2 -td% -f db2look.sql
```

データベース DEPARTMENT 内のオブジェクト用に、CREATE VIEW ステートメントを除く DDL ステートメントを生成します。db2look の出力は、以下のようにしてファイル db2look.sql に送信します。

```
db2look -d department -e -noview -o db2look.sql
```

指定した表に関連するデータベース DEPARTMENT 内のオブジェクト用に、DDL ステートメントを生成します。db2look の出力は、以下のようにしてファイル db2look.sql に送信します。

```
db2look -d department -e -t tab1 \"My TaB1E2\" -o db2look.sql
```

連合データベース FEDDEPART にすべてのオブジェクト (連合および非連合) 用の DDL ステートメントを生成します。連合 DDL ステートメントでは、指定されたラッパー FEDWRAP に適用されるもののみが生成されます。db2look 出力が標準出力に送信されます。

```
db2look -d feddepart -e -wrapper fedwrap
```

非連合 DDL のみを含むスクリプト・ファイルを生成します。以下のシステム・コマンドは、連合データベース (FEDDEPART) に対して実行でき、連合されていないデータベースの実行時に検出されたような出力を生成するだけです。db2look 出力がファイル out.sql に送信されます。

```
db2look -d feddepart -e -nofed -o out
```

使用上の注意:

Windows システムでは、db2look は DB2 コマンド・ウィンドウから実行される必要があります。

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

既存のオプションのいくつかは、連合環境をサポートします。以下の db2look コマンド行オプションが連合環境で使用されます。

- -e
使用されると、連合 DDL が生成されます。
- -g
使用されると、連合オブジェクトへの特権を付与するために、GRANT ステートメントが生成されます。
- -xd

db2look - DB2 統計および DDL 抽出ツール

使用されると、システム付与の特権を連合オブジェクトに追加するために、連合 DDL が生成されます。

- -f

使用されると、連合関連情報がデータベース・マネージャー構成から抽出されます。

- -m

使用されると、ニックネームの統計が抽出されます。

連合システムを使用する能力は、連合 DDL を作成するために、データベース・マネージャー構成でオンにされる必要があります。一度 **db2look** コマンドがスクリプト・ファイルを生成すると、このスクリプトを実行する前に、“federated” をオンにするのはユーザーの責任です。

出力スクリプトを変更して、CREATE USER MAPPING ステートメントのリモート・パスワードを追加する必要があります。

DB2 ファミリー・インスタンスをデータ・ソースとして定義するのに使用される、これらの CREATE SERVER ステートメントに、AUTHORIZATION および PASSWORD を追加して、**db2look** コマンド出力スクリプトを変更する必要があります。

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

組み込み SQL ステートメントを含む SQLj プロファイルを処理します。デフォルトでは、各分離レベルごとに 1 つずつ、4 つの DB2 パッケージがデータベースに作成されます。このユーティリティは、実行時に使用するために、DB2 特定情報を持つプロファイルを増加し、SQLj アプリケーションが変換された後の、アプリケーションが実行される前に実行する必要があります。

許可:

以下のいずれかが必要です。

- *sysadm* または *dbadm* 権限
- パッケージが存在しない場合は、BINDADD 特権および以下のどちらかが必要です。
 - パッケージのスキーマ名が存在しない場合は、データベースに対する IMPLICIT_SCHEMA 権限
 - パッケージのスキーマ名が存在する場合は、スキーマに対する CREATEIN 特権
- パッケージが存在する場合は、スキーマに対する ALTERIN 特権
- パッケージに対する BIND 特権 (パッケージが存在する場合)

ユーザーには、アプリケーション内の静的 SQL ステートメントをコンパイルするのに必要な特権もすべて必要になります。グループに付与された特権は、静的ステートメン

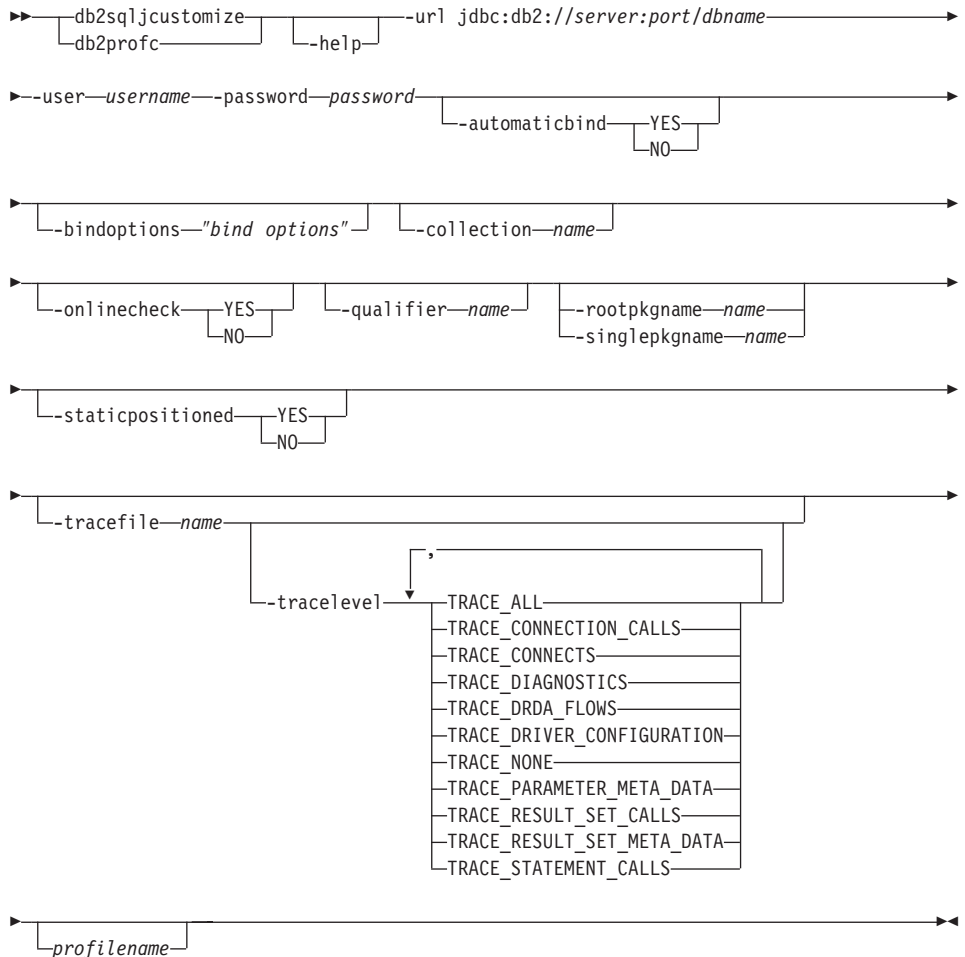
db2sqljcustomize - DB2 SQLj プロファイル・カスタマイザー

トの許可検査では使用されません。ユーザーに *sysadm* 権限があつてバインドを完成させる明示特権がない場合、データベース・マネージャーは、明示的な *dbadm* 権限を自動的に付与します。

必要な接続:

このコマンドは *-url* が指定されている場合、データベース接続を確立します。

コマンド構文:



コマンド・パラメーター:

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

-url jdbc:db2://server:port/dbname

データベース接続の設定に使用する JDBC URL を指定します。接続の確立に

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

は DB2 JDBC タイプ 4 ドライバーが使用されます。オンライン検査または自動バインドが使用可能な場合に必要です。

-user *username*

データベースに接続するときに使用する名前を指定します。

-password *password*

ユーザー名のパスワードを指定します。

-automaticbind

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

-bindoptions "*bind options*"

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

- DB2 (Windows および UNIX 版) の場合:
 - ACTION (ACTION RETAIN 以外)
 - BLOCKING
 - COLLECTION
 - DEGREE
 - EXPLAIN
 - EXPLSNAP
 - FEDERATED
 - FUNCPATH
 - INSERT
 - ISOLATION (下記の -singlepkgname オプションを参照してください)
 - OWNER
 - QUALIFIER
 - QUERYOPT
 - SQLERROR (SQLERROR CHECK 以外)
 - SQLWARN
 - VALIDATE
 - VERSION
- Windows および UNIX 以外のサーバー上の DB2 の場合:
 - ACTION (ACTION RETAIN 以外)
 - BLOCKING
 - COLLECTION
 - DBPROTOCOL
 - DEGREE
 - EXPLAIN
 - IMMEDIATEWRITE

- | - ISOLATION (db2sqljcustomize コマンドの -singlepkgname オプションを参
- | 照してください)
- | - NOREOPT
- | - OPTHINT
- | - OWNER
- | - PATH
- | - QUALIFIER
- | - RELEASE
- | - REOPT
- | - SQLERROR
- | - VALIDATE
- | - VERSION

-collection *name*

| デフォルトのコレクション IDを指定します。指定されていない場合は、
| NULLID が使用されます。コレクションが SET CURRENT PACKAGESET 経
| 由で明示的に設定されていない場合は、実行時にデフォルト・コレクションが
| 使用されます。

-onlinecheck

| オンライン検査が、-url オプションで指定されたデータベースを使用して実行
| されるかどうかを決定します。有効な値は YES または NO です。デフォルト
| は -url が指定されている場合は YES です。そうでない場合はデフォルトは
| NO です。

-qualifier *name*

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

-rootpkgname *name*

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

-singlepkgname *name*

SQLj バインダーによって生成されるパッケージ名を指定します。最大長は 8 文字です。このオプションには、`-bindoptions` フラグで指定される `ISOLATION` バインド・フラグが必要です。これは、単一トランザクション分離レベルを使用するアプリケーションにのみ使用してください。

-staticpositioned

`iterator` が更新ステートメントと同じプログラムで宣言されている場合、このオプションで、位置決めされた更新 (`UPDATE`) が、動的に準備されたステートメントではなく、静的にバインドされたステートメントを通じて発生するようになります。値は `YES` および `NO` です。デフォルト値は `YES` です。詳しくは使用上の注意を参照してください。

-tracefile *name*

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

-tracelevel

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

profilename

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

例:

```
db2sqljcustomize -user richler -password mordecai
  -url jdbc:db2:/server:50000/sample -collection duddy
  -bindoptions "EXPLAIN YES" pgmname_SJProfile0.ser
```

使用上の注意:

`-staticPositioned YES` オプションの使用の意味

SQLj は、メソッド間で `iterators` を変数として渡すことを許可します。変数として渡され、位置決めされた更新 (`UPDATE` または `DELETE`) 用を使用される `iterator` は、実行時にのみ識別されます。また、同じ SQLj 位置決めされた更新ステートメントは、実行時に異なる `iterator` で使用できます。静的に実行するために、SQLj カスタマイザーが、位置決めされた更新ステートメントを準備する場合、どの照会がどの位置決めされた更新ステートメントに属するかを判別する必要があります。SQLj カスタマイザーは、照会ステートメントと位置決めされた `UPDATE` ステートメントとの間をマップする `iterator` のクラスを使用して、これを行います。 `iterator` クラスが照会ステートメントと

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

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

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

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

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

上記のようなプログラムで、バインド時のエラーを避けるには、DB2 BIND オプション `SQLERROR(CONTINUE)` を指定します。しかし、プログラムに書き込む技法のほうが良く、iterator クラス、照会、および位置決めされた UPDATE または DELETE 間の固有マッピングがあるからです。下記の例には、これを行う方法が表示されています。このコーディングのメソッドでは、各 iterator クラスは 1 つの iterator インスタンスにのみ関連付けられています。そのため、DB2 バインド処理は常に位置決めされた UPDATE ステートメントを、正しい照会と関連付けます。

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

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

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

```
    ...
    updateTable1( iter1 );
}

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

関連資料:

- コマンド・リファレンス の『BIND コマンド』
- 1304 ページの『db2sqljprint - DB2 SQLj プロファイル・プリンター』
- 1312 ページの『db2sqljbind - DB2 SQLj プロファイル・バインダー』

db2sqljprint - DB2 SQLj プロファイル・プリンター

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

許可:

なし

必要な接続:

なし

コマンド構文:

```
db2sqljprint <profilename>
  db2profp
```

コマンド・パラメーター:

profilename

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

例:

```
db2sqljprint pgmname_SJProfile0.ser
```

関連資料:

- 1298 ページの『db2sqljcustomize - DB2 SQLj プロファイル・カスタマイザー』

- 1312 ページの『db2sqljbind - DB2 SQLj プロファイル・バインダー』

db2relocatedb - データベースの再配置

ユーザー提供の構成ファイルに指定されたように、データベースを名前変更、またはデータベース、またはデータベースの一部（たとえばコンテナおよびログ・ディレクトリ）を再配置します。このツールは、DB2 インスタンスおよびデータベース・サポート・ファイルに、必要な変更を行います。

許可:

なし

コマンド構文:

```
▶▶ db2relocatedb -f configFilename ▶▶
```

コマンド・パラメーター:

-f configFilename

データベースを再配置するために必要な構成情報を含むファイルの名前を指定します。これは、相対ファイル名でも絶対ファイル名でも構いません。構成ファイルのフォーマットは以下のとおりです。

```
DB_NAME=oldName,newName
DB_PATH=oldPath,newPath
INSTANCE=oldInst,newInst
NODENUM=nodeNumber
LOG_DIR=oldDirPath,newDirPath
CONT_PATH=oldContPath1,newContPath1
CONT_PATH=oldContPath2,newContPath2
...
```

ここで、

DB_NAME

再配置されるデータベースの名前を指定します。データベース名が変更する場合は、古い名前と新規の名前の両方を指定する必要があります。このフィールドは必須です。

DB_PATH

再配置されるデータベースのパスを指定します。このパスは、データベースが初めに作成された場所です。データベース・パスが変更される場合、古いパスと新規のパスの両方を指定する必要があります。このフィールドは必須です。

INSTANCE

データベースが存在する場所のインスタンスを指定します。データベ

ースが新規のインスタンスに移動される場合、古いインスタンスと新規のインスタンスの両方を指定する必要があります。このフィールドは必須です。

NODENUM

変更されるデータベース・ノードのノード番号を指定します。デフォルトは 0 です。

LOG_DIR

ログ・パスのロケーション内の変更を指定します。ログ・パスが変更されている場合、古いパスと新規のパスの両方を指定する必要があります。ログ・パスがデータベース・パスの下にある場合、パスは自動的に更新されるので、この指定はオプションです。

CONT_PATH

表スペース・コンテナのロケーション内の変更を指定します。古いコンテナと新規のコンテナの両方を指定する必要があります。複数コンテナ・パスが変更される場合、複数 **CONT_PATH** 行が提供されます。コンテナ・パスがデータベース・パスの下にある場合、パスは自動的に更新されるので、この指定はオプションです。同じ古いパスが共通の新規パスで置換される場所で、複数のコンテナに変更を行う場合、単一の **CONT_PATH** 項目が使用されます。このような場合、古いパス、新規パスの両方にアスタリスク (*) をワイルドカードとして使用できます。

注: ブランク行またはコメント文字 (#) で開始される行は無視されます。

使用上の注意:

データベースが属するインスタンスが変更されている場合、インスタンスおよびデータベース・サポート・ファイルへの変更が行われるのを確認するために、このコマンドの実行前に以下を行う必要があります。

- データベースが他のインスタンスに移動されている場合は、新規のインスタンスを作成します。
- 新規インスタンスが常駐するシステムにコピーされているデータベースに属するファイル / デバイスをコピーします。パス名を変更する必要があります。
- インスタンス所有者に所有されるように、コピーされたファイル / デバイスのアクセス権を変更します。

インスタンスが変更されている場合、ツールは新規のインスタンス所有者によって実行される必要があります。

パーティション・データベース環境では、このツールは変更が必要なすべてのパーティションに対して実行される必要があります。それぞれのパーティションには、変更されているパーティションの **NODENUM** 値を含む、別々の構成ファイルが提供されなければ

ばなりません。たとえば、データベースの名前が変更されている場合、すべてのパーティションは影響を受け、それぞれのパーティションの、別々の構成ファイルで **db2relocatedb** コマンドを実行する必要があります。単一データベース・パーティションに属するコンテナが移動されている場合、 **db2relocatedb** コマンドは、そのパーティションで一度だけ実行される必要があります。

例:

例 1

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

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

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

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

例 2

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

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

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

例 3

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

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

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

db2relocatedb - データベースの再配置

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

例 4

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

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

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

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

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

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

例 5

データベース TESTDB には、データベース・パーティション・サーバー 10 および 20 に 2 つのパーティションがあります。このインスタンスは servinst で、データベース・パスは両方のデータベース・パーティション・サーバーで /home/servinst です。データベースの名前は SERVDB に変更され、データベース・パスは両方のデータベー

ス・パーティション・サーバーで /databases に変更されます。さらに、ログ・ディレクトリはデータベース・パーティション・サーバー 20 で、 /testdb_logdir から /servdb_logdir に変更されます。

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

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

```
DB_NAME=TESTDB,SERVDB
DB_PATH=/home/servinst,/databases
INSTANCE=servinst
NODE_NUM=10
```

データベース・パーティション・サーバー 20 では、以下の構成ファイルが使用されます。

```
DB_NAME=TESTDB,SERVDB
DB_PATH=/home/servinst,/databases
INSTANCE=servinst
NODE_NUM=20
LOG_DIR=/testdb_logdir,/servdb_logdir
```

例 6

パス /home/maininst のインスタンス maininst 内に存在するデータベース MAINDB です。4 つの表スペース・コンテナのロケーションを、以下のように変更する必要があります。

```
/maininst_files/allconts/C0 needs to be moved to /MAINDB/C0
/maininst_files/allconts/C1 needs to be moved to /MAINDB/C1
/maininst_files/allconts/C2 needs to be moved to /MAINDB/C2
/maininst_files/allconts/C3 needs to be moved to /MAINDB/C3
```

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

注: 同様の変更が、すべてのコンテナに対して行われました。すなわち、

/maininst_files/allconts/ が /MAINDB/ で置換され、ワイルドカード文字のある単一項目が使用できるようになります。

```
DB_NAME=MAINDB
DB_PATH=/home/maininst
INSTANCE=maininst
NODE_NUM=0
CONT_PATH=/maininst_files/allconts/*, /MAINDB/*
```

db2relocatedb - データベースの再配置

関連資料:

- 1315 ページの『db2inidb - ミラーリングされたデータベースの初期化』

db2support - 問題分析および環境収集ツール

クライアント・マシンまたはサーバー・マシンについての環境データを収集して、システム・データを含むファイルを圧縮ファイル・アーカイブに置きます。

このツールは、ユーザーとの対話式の質問と答えによって、問題の性質についての基本データを収集することもできます。

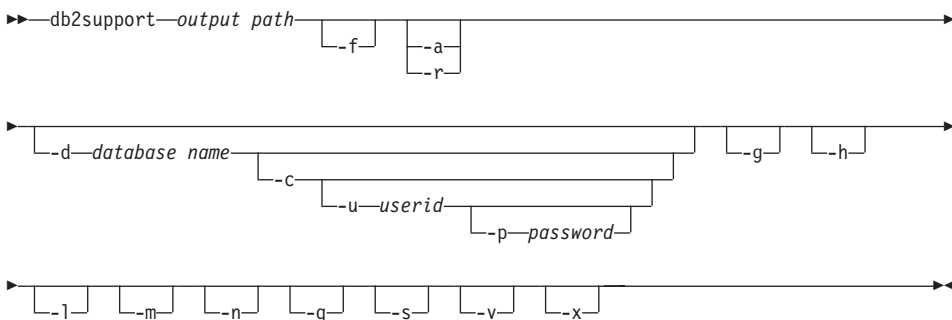
許可:

ほとんどの完全出力の場合、このユーティリティーはインスタンス所有者が起動する必要があります。システムに対するより限定された特権を持つユーザーはこのツールを実行できますが、データ収集アクションによっては、報告が少なくなったり出力が少なくなったりするものもあります。

必要な接続:

なし

コマンド構文:



コマンド・パラメーター:

output path

アーカイブ・ライブラリーを作成するパスを指定します。これは、アーカイブに組み込むために、ユーザー作成ファイルが置かれるディレクトリーです。

-f または -flow

ユーザーに対して、`Press <Enter> key to continue` という要求が出されたときに、休止を無視します。このオプションは、自動実行が要求される場合に、スクリプトまたはその他の自動プロシージャを経由して **db2support** ツールを実行したり、呼び出したりする際に役に立ちます。

- a または -all_core**
すべてのコア・ファイルを取り込むことを指定します。
- r または -recent_core**
最新のコア・ファイルを取り込むことを指定します。このオプションは、-a オプションを指定すると無視されます。
- d database_name または -database database_name**
データが収集されているデータベースの名前を指定します。
- c または -connect**
指定のデータベースに接続するための試行を行うことを指定します。
- u userid または -user userid**
データベースに接続するためのユーザー ID を指定します。
- p password または -password password**
ユーザー ID のパスワードを指定します。
- g または -get_dump**
ダンプ・ディレクトリー内のすべてのファイル (コア・ファイルを除く) を取り込むことを指定します。
- h または -help**
ヘルプ情報を表示します。このオプションを指定すると、他のすべてのオプションは無視され、ヘルプ情報だけが表示されます。
- l または -logs**
アクティブ・ログを取り込むことを指定します。
- m または -html**
すべてのシステム出力が、HTML 形式のファイルにダンプされることを指定します。デフォルトでは、すべてのシステム関連情報は、このパラメーターが使用されない場合はフラット・テキスト・ファイルにダンプされます。
- n または -number**
現在の問題の問題管理レポート (PMR) 番号または ID を指定します。
- q または -question_response**
対話式問題分析モードを使用することを指定します。
- s または -system_detail**
ハードウェアおよびオペレーティング・システムの詳細情報を収集することを指定します。
- v または -verbose**
このツールの実行中に、冗長出力を使用することを指定します。
- x または -xml_generate**
対話式問題分析モード (-q モード) 中に使用される判断ツリー論理全体を含む XML 文書を生成することを指定します。

db2support - 問題分析および環境収集ツール

使用上の注意:

ビジネス・データのセキュリティを保護するために、このツールは、表データ、スキーマ (DDL)、またはログを収集しません。オプションによっては、スキーマおよびデータ (アーカイブ・ログなど) のいくつかの性質を組み込むことができるものもあります。データベースのスキーマまたはデータを公開するオプションは、注意して使用する必要があります。このツールが起動されると、機密データを扱う方法を示すメッセージが表示されます。

db2support ツールから収集されるデータは、ツールが実行されているマシンから取得できます。クライアント/サーバー環境では、データベースに関連した情報は、インスタンス・アタッチまたはデータベース接続を経由して、データベースが常駐するマシンから得られます。たとえば、オペレーティング・システム情報またはハードウェア情報 (-s オプション)、および診断ディレクトリーからのファイル (DIAGPATH) は、db2support ツールを実行しているローカル・マシンから、そしてバッファー・プール、データベース構成、および表スペース情報などのデータは、データベースが物理的に常駐しているマシンから得られます。

db2sqljbind - DB2 SQLj プロファイル・バインダー

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

許可:

以下のいずれかが必要です。

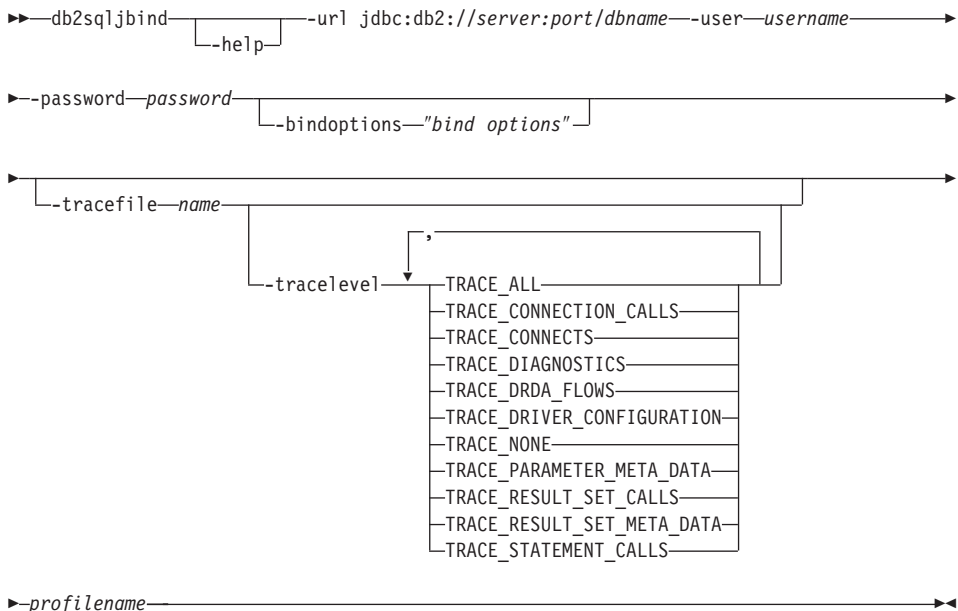
- *sysadm* または *dbadm* 権限
- パッケージが存在しない場合は、BINDADD 特権および以下のどちらかが必要です。
 - パッケージのスキーマ名が存在しない場合は、データベースに対する IMPLICIT_SCHEMA 権限
 - パッケージのスキーマ名が存在する場合は、スキーマに対する CREATEIN 特権
- パッケージが存在する場合は、スキーマに対する ALTERIN 特権
- パッケージに対する BIND 特権 (パッケージが存在する場合)

ユーザーには、アプリケーション内の静的 SQL ステートメントをコンパイルするのに必要な特権もすべて必要になります。グループに付与された特権は、静的ステートメントの許可検査では使用されません。ユーザーに *sysadm* 権限があつてバインドを完成させる明示特権がない場合、データベース・マネージャーは、明示的な *dbadm* 権限を自動的に付与します。

必要な接続:

このコマンドは、データベース接続を確立します。

コマンド構文:



コマンド・パラメーター:

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

-url jdbc:db2://server:port/dbname

データベース接続の設定に使用する JDBC URL を指定します。接続の確立には DB2 JDBC タイプ 4 ドライバーが使用されます。

-user username

データベースに接続するとき使用する名前を指定します。

-password password

ユーザー名のパスワードを指定します。

-bindoptions "bind options"

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

- DB2 (Windows および UNIX 版) の場合:
 - ACTION (ACTION RETAIN 以外)
 - BLOCKING
 - COLLECTION
 - DEGREE

db2sqljbind - DB2 SQLj プロファイル・バインダー

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

-tracefile *name*

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

-tracelevel

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

profilename

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

例:

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

使用上の注意:

関連資料:

- コマンド・リファレンス の『BIND コマンド』
- 1298 ページの『db2sqljcustomize - DB2 SQLj プロファイル・カスタマイザー』
- 1304 ページの『db2sqljprint - DB2 SQLj プロファイル・プリンター』

db2inidb - ミラーリングされたデータベースの初期化

分割ミラー環境のミラーリングされたデータベースを初期化します。ミラーリングされたデータベースは、ロールフォワード・ペンディング状態にある 1 次データベースの複製として初期化したり、1 次データベースをリストアするためのバックアップ・イメージとして使用できます。

許可:

以下のいずれかが必要です。

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

必要な接続:

なし

コマンド構文:

```
▶—db2inidb—database_alias—AS—  

   |——SNAPSHOT——|  

   |——STANDBY——|  

   |——MIRROR——|  

   |——RELOCATE USING——configFile——|
```

コマンド・パラメーター:

db2inidb - ミラーリングされたデータベースの初期化

database_alias

初期設定するデータベースの別名を指定します。

SNAPSHOT

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

STANDBY

データベースをロールフォワード・ペンディング状態にすることを指定します。

注: 1 次データベースからの新しいログは、フェッチおよびスタンバイ・データベースに適用することが可能です。スタンバイ・データベースは、1 次データベースがダウンした場合に、その代わりに使用できます。

MIRROR

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

RELOCATE USING configFile

データベースをスナップショット、スタンバイ、またはミラーとして初期化する前に、構成ファイルにリストされている情報に基づいて、データベース・ファイルを再配置することを指定します。

使用上の注意:

RELOCATE USING *configfile* パラメーターが指定され、データベースが正常に再配置された場合、構成ファイルはデータベース・ディレクトリーにコピーされ、db2path.cfg に名前変更されます。後続のクラッシュ・リカバリーまたはロールフォワード・リカバリー中に、この構成ファイルはコンテナ・パスを、ログ・ファイルが処理されるように名前変更するために使用されます。

スナップショットまたはミラーリングされたデータベースが初期化されると、構成ファイルはクラッシュ・リカバリーの完了後に自動的にデータベース・ディレクトリーから除去されます。

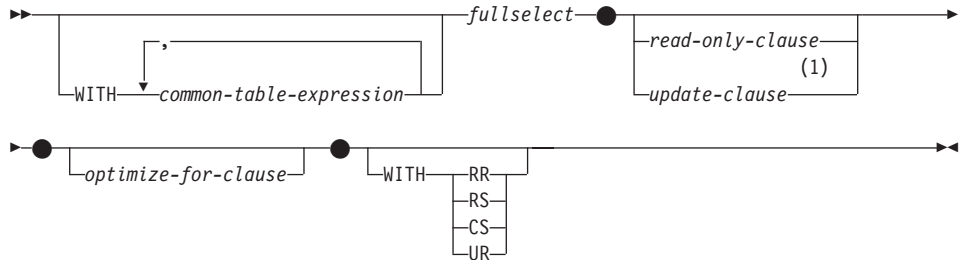
スタンバイ・データベースが初期化されると、構成ファイルはロールフォワード・リカバリーの完了または取り消し後に、自動的にデータベース・ディレクトリーから除去されます。db2inidb の実行後に、db2path.cfg ファイルに新規コンテナ・パスを追加できます。これは、CREATE または ALTER TABLESPACE 操作が元のデータベース上で行われ、スタンバイ・データベースに別のパスを使用しなければならない場合に必要になります。

関連資料:

- 1305 ページの『db2relocatedb - データベースの再配置』

SQL 解説書

Select-statement



注:

- 1 全選択に *order-by-clause* が含まれている場合、*update-clause* を指定することはできません。

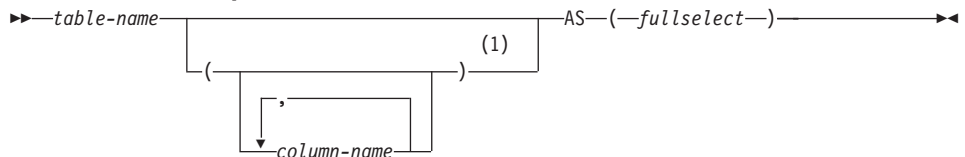
select ステートメントは、*DECLARE CURSOR* ステートメントに直接指定することや、準備した後で *DECLARE CURSOR* ステートメントで参照することができる照会の形式です。また、*select* ステートメントはコマンド行プロセッサ（または同種のツール）を使って動的 SQL ステートメントを使用することにより発行でき、それにより、結果表を画面に表示することもできます。いずれの場合も、*select* ステートメントによって指定される表は、全選択 (*fullselect*) の結果です。

オプションの *WITH* 文節は、*select* ステートメントが実行される分離レベルを指定しません。

- RR - 反復可能読み取り
- RS - 読み取り固定
- CS - カーソル固定
- UR - 非コミット読み取り

ステートメントのデフォルト分離レベルは、ステートメントがバインドされているパッケージの分離レベルです。

common-table-expression



注:

- 1 共通表式 (common table expression) が再帰的である場合、あるいは全選択の結果として列名が重複する場合は、列名を指定する必要があります。

共通表式を使用すると、結果表を *table-name* (表名) によって定義して、それをその後続く全選択の任意の FROM 文節に指定できるようにすることができます。単一の WITH キーワードの後に、複数の共通表式を指定することができます。指定する各共通表式は、それ以降の共通表式の FROM 文節の中でも名前によって参照することができます。

列のリストを指定する場合、その中の列の名前の数は、全選択の結果表内の列数と同じ数でなければなりません。各 *column-name* (列名) は、固有で、しかも非修飾でなければなりません。これらの列名を指定しない場合、共通表式の定義に使用された全選択の選択リストから名前が取られます。

共通表式の *table-name* は、同じステートメントの他の共通表式の *table-name* すべてと異なるものでなければなりません (SQLSTATE 42726)。共通表式が INSERT ステートメントに指定されている場合、*table-name* を、その挿入の対象である表またはビューの名前にすることはできません (SQLSTATE 42726)。共通表式の *table-name* は、その全選択を通じて、どの FROM 文節の中でも表名として指定することができます。共通表式の *table-name* は、(カタログの中で) 同じ修飾名の既存の表、ビュー、または別名をオーバーライドするものとなります。

同じステートメントの中に複数の共通表式が定義されている場合、共通表式相互間の循環参照があってはなりません (SQLSTATE 42835)。循環参照は、2 つの共通表式 *dt1* および *dt2* (*dt2* を参照する *dt1* および *dt1* を参照する *dt2* など) が作成されたときに発生します。

共通表式は、CREATE VIEW および INSERT の各ステートメントの前でもオプションとして使用できます。

共通表式は、以下のような場合に使用することができます。

- ビューの代わりに使用して、ビューが作成されないようにするため (ビューを一般的に使用する必要がなく、定位置の更新や削除を使わない場合)
- 決定的でなく外部処理を伴うスカラー副選択または関数から得られる列によりグループ化できるようにする場合
- 必要な結果表がホスト変数に基づいたものである場合
- 同じ結果表を全選択で共用する必要がある場合
- 結果表を再帰的に派生させる必要がある場合

共通表式的全選択の FROM 文節の中にそれ自体への参照が含まれる場合、その共通表式は、再帰的共通表式です。再帰処理を使用した照会は、部品構成表 (BOM)、予約システム、およびネットワーク・プランなどのアプリケーションをサポートする上で役立ちます。

再帰的共通表式では、以下のことが成り立っていなければなりません。

- 再帰サイクルの一部をなす各全選択は、SELECT または SELECT ALL で始まっていなければなりません。SELECT DISTINCT は使用できません (SQLSTATE 42925)。また、集合の和を求める場合には UNION ALL を使用する必要があります (SQLSTATE 42925)。
- 共通表式の *table-name* (表名) の後には、必ず列名を指定する必要があります (SQLSTATE 42908)。
- 最初の UNION の最初の全選択 (初期化全選択) には、どの FROM 文節の共通表式のどの列に対する参照も含まれてはなりません (SQLSTATE 42836)。
- 共通表式の列名が反復全選択において参照される場合、その列のデータ・タイプ、長さ、およびコード・ページは、初期化全選択に基づいて決められます。反復全選択の中の対応する列のデータ・タイプと長さは、初期化全選択に基づいて決められたデータ・タイプと長さと同じでなければならず、コード・ページは一致していなければなりません (SQLSTATE 42825)。ただし、文字ストリング・タイプの場合は、2 つのデータ・タイプの長さが違って構いません。この場合、反復全選択の列の長さは、初期化全選択から決められた長さに常に割り当て可能な長さでなければなりません。
- 再帰サイクルの一部をなす各全選択には、列関数、GROUP BY 文節、または HAVING 文節が含まれてはなりません (SQLSTATE 42836)。

これらの全選択の FROM 文節には、再帰サイクルの一部である共通表式に対する参照を多くても 1 つまで含めることができます (SQLSTATE 42836)。
- 反復全選択および全体再帰的全選択には、ORDER BY 文節を含めることはできません (SQLSTATE 42836)。
- 副照会 (スカラーまたは多値) が再帰サイクルの一部であってはなりません (SQLSTATE 42836)。

再帰的共通表式を開発するときには、無限再帰サイクル (ループ) が作成される恐れについて常に注意してください。再帰サイクルは、必ず終了するようにしてください。これは、関係しているデータが循環している場合に特に重要です。再帰的共通表式には、無限ループを防止する述部を含めるようにしてください。再帰的共通表式には、以下のものを含めるようにしてください。

- 反復全選択の中に、定数ずつ増分される整数列。
- "counter_col < constant" または "counter_col < :hostvar" の形式の反復全選択の WHERE 文節の述部。

この構文が再帰的共通表式に見つからないなら、警告が出されます (SQLSTATE 01605)。

update-clause

update-clause

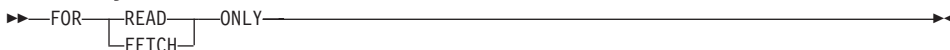


FOR UPDATE 文節は、それ以降の定位置 UPDATE ステートメントで更新可能となる列を指定します。各 *column-name* (列名) は非修飾でなければならず、全選択の最初の FROM 文節で指定された表またはビューの列を指定するものでなければなりません。列名のない FOR UPDATE 文節を指定すると、全選択の最初の FROM 文節に指定された表またはビューの列のうち更新可能な列すべてが含まれます。

以下のいずれかに該当する場合、FOR UPDATE 文節は使用できません。

- select ステートメントに関連付けられているカーソルが削除不能である。
- 選択された列のいずれかがカタログ表の更新不能な列であり、FOR UPDATE 列がその列を除外するのに使用されていない。

read-only-clause



FOR READ ONLY 文節は、結果表が読み取り専用であり、カーソルを定位置 UPDATE ステートメントおよび定位置 DELETE ステートメントで参照できないことを指定します。FOR FETCH ONLY も同じ意味です。

結果表の中には、最初から読み取り専用のものがあります。(読み取り専用ビューに基づく表など。) FOR READ ONLY は、このような表にも指定できますが、指定しても効果はありません。

更新と削除ができない結果表の場合、FOR READ ONLY (または FOR FETCH ONLY) を指定すると、データベース・マネージャが、ブロッキングを行うことができるため、FETCH 操作のパフォーマンスが向上する可能性があります。たとえば、FOR READ ONLY 文節または ORDER BY 文節のない動的 SQL ステートメントを含むプログラムでは、FOR UPDATE 文節が指定されたかのようにして、データベース・マネージャがカーソルをオープンする場合があります。したがって、定位置 UPDATE または DELETE ステートメントで照会を使用する場合以外は、パフォーマンスを向上させるために FOR READ ONLY 文節を使用するようにしてください。

読み取り専用結果表は、それが最初から読み取り専用であるか、それとも FOR READ ONLY (FOR FETCH ONLY) として指定されたのかには関係なく、定位置 UPDATE または DELETE ステートメントで参照することはできません。

optimize-for-clause



OPTIMIZE FOR 文節は、*select* ステートメント の特殊な処理を要求します。この文節が省略されると、結果表のすべての行が検索されることが想定されます。指定されている場合には、検索される行数はおそらく n を超えないことを前提としています。ここで、 n は *integer* の値です。 n の値は、正の整数でなければなりません。OPTIMIZE FOR 文節を使用すると、 n 個の行が検索されることを前提とする照会の最適化に影響を与えます。さらに、ブロックされているカーソルの場合、この文節は、各ブロックで戻される行の数に影響を与えます（すなわち、各ブロックで戻される行の数は n 行以下になります）。*fetch-first-clause* と *optimize-for-clause* の両方が指定されている場合には、これらの文節の *integer* 値のうちの小さい方を使用して通信バッファ・サイズが決定されます。これらの値は最適化処理専用です。

この文節を指定しても、取り出される行の数が制限されることはなく、パフォーマンス以外ではどんな点でも結果に影響を与えることはありません。OPTIMIZE FOR n ROWS を使用した場合、 n 個以下の行を取り出す場合にはパフォーマンスが向上することがありますが、 n 個を超える行を取り出す場合にはパフォーマンスが低下する可能性があります。

n の値に行のサイズを乗算した値が、通信バッファのサイズを超える場合、OPTIMIZE FOR 文節はデータ・バッファに影響を与えません。通信バッファのサイズは、RQRIOBLK または ASLHEAPSZ 構成パラメーターによって定義されます。

select ステートメントの例

例 1: EMPLOYEE 表からすべての列と行を選択します。

```
SELECT * FROM EMPLOYEE
```

例 2: PROJECT 表からプロジェクト名 (PROJNAME)、開始日 (PRSTDATE)、および終了日 (PRENDATE) を選択します。その日付が最新の終了日から順に結果表を配列します。

```
SELECT PROJNAME, PRSTDATE, PRENDATE
FROM PROJECT
ORDER BY PRENDATE DESC
```

例 3: EMPLOYEE 表のすべての部門の部門番号 (WORKDEPT) と部門別給与 (SALARY) の平均額を選択します。結果表は、部門別給与の平均額の昇順に配列します。

```
SELECT WORKDEPT, AVG(SALARY)
FROM EMPLOYEE
GROUP BY WORKDEPT
ORDER BY 2
```

select ステートメントの例

例 4: C プログラムで使用する UP_CUR という名前のカーソルを宣言して、PROJECT 表の開始日 (PRSTDATE) と終了日 (PRENDATE) の列を更新します。プログラムは、各行のこれらの 2 つの値と、プロジェクト番号 (PROJNO) とを受け取る必要があります。

```
EXEC SQL DECLARE UP_CUR CURSOR FOR
        SELECT PROJNO, PRSTDATE, PRENDATE
        FROM PROJECT
        FOR UPDATE OF PRSTDATE, PRENDATE;
```

例 5: この例では、SAL+BONUS+COMM に TOTAL_PAY という名前を付けます。

```
SELECT SALARY+BONUS+COMM AS TOTAL_PAY
        FROM EMPLOYEE
        ORDER BY TOTAL_PAY
```

例 6: セールス担当員の従業員番号と給与、およびその部門の給与平均額と人数とを調べます。また、部門別給与平均額と、平均額の最高値も調べます。

ここでは、共通表式を使用することによって、DINFO ビューを正規のビューとして作成したときのオーバーヘッドを軽減します。ステートメントの作成中に、ビューのカタログにはアクセスされません。これは、全選択の残りの部分のコンテキストにより、ビューによって考慮する必要があるのはセールス担当の部門の行だけだからです。

```
WITH
    DINFO (DEPTNO, AVGSALARY, EMPCOUNT) AS
        (SELECT OTHERS.WORKDEPT, AVG(OTHERS.SALARY), COUNT(*)
         FROM EMPLOYEE OTHERS
         GROUP BY OTHERS.WORKDEPT
        ),
    DINFOMAX AS
        (SELECT MAX(AVGSALARY) AS AVGMAX FROM DINFO)
SELECT THIS_EMP.EMPNO, THIS_EMP.SALARY,
       DINFO.AVGSALARY, DINFO.EMPCOUNT, DINFOMAX.AVGMAX
FROM EMPLOYEE THIS_EMP, DINFO, DINFOMAX
WHERE THIS_EMP.JOB = 'SALESREP'
AND THIS_EMP.WORKDEPT = DINFO.DEPTNO
```

関連資料:

- SQL リファレンス 第 1 巻 の『副選択』
- SQL リファレンス 第 2 巻 の『DECLARE CURSOR ステートメント』
- SQL リファレンス 第 1 巻 の『再帰の例: 部品構成表』

SQLDA (SQL 記述子域)

SQLDA は、SQL DESCRIBE ステートメントの実行に必要な変数の集まりです。SQLDA 変数は、PREPARE、OPEN、FETCH、および EXECUTE ステートメントでは、オプションとして使用できます。SQLDA は、動的 SQL との間で情報を伝えます。こ

これは、DESCRIBE ステートメントで使用され、ホスト変数のアドレスで変更して、FETCH または EXECUTE ステートメントで再び使用することができます。

SQLDA は、すべての言語でサポートされていますが、事前定義の宣言を使用できるのは、C、REXX、FORTRAN、および COBOL に対してだけです。

SQLDA の情報の意味は、その使用法によって異なります。PREPARE と DESCRIBE の場合、SQLDA は準備されるステートメントに関する情報をアプリケーション・プログラムに提供します。OPEN、EXECUTE、および FETCH の場合、SQLDA はホスト変数を記述します。

DESCRIBE および PREPARE において、記述する列のいずれかが LOB タイプ (LOB ロケーターおよびファイル参照変数では、2 倍の SQLDA は必要ありません)、参照タイプ、またはユーザー定義タイプの場合、SQLDA 全体の SQLVAR 項目の数を 2 倍にする必要があります。たとえば:

- 3 つの VARCHAR の列と 1 つの INTEGER の列の含まれている表を記述する場合、SQLVAR 項目は 4 つになります。
- 2 つの VARCHAR の列と 1 つの CLOB の列、および 1 つの INTEGER の列の含まれている表を記述する場合には、SQLVAR 項目は 8 つになります。

EXECUTE、FETCH、および OPEN において、記述する変数のいずれかが LOB タイプ (LOB ロケーターおよびファイル参照変数では、2 倍の SQLDA は必要ありません) または構造タイプの場合、SQLDA 全体の SQLVAR 項目の数を 2 倍にする必要があります。特殊タイプと参照タイプは、これらの場合には関係ありません。これは、それらのタイプの場合、データベースが 2 倍の数の項目の追加情報を必要としないためです。)

SQLDA フィールド記述

SQLDA は、4 つの変数と、その後に SQLVAR と総称して呼ばれる変数の任意の数のオカレンスによって構成されています。OPEN、FETCH、および EXECUTE では、SQLVAR の各オカレンスによってホスト変数が記述されます。DESCRIBE と PREPARE では、SQLVAR の各オカレンスによって結果表またはパラメーター・マーカの列が記述されます。SQLVAR の項目には、以下の 2 つのタイプがあります。

- **基本 SQLVAR:** これらの項目は常に存在します。これらの項目には、データ・タイプのコード、長さ属性、列名、ホスト変数のアドレス、および標識変数アドレスなどの列、パラメーター・マーカ、またはホスト変数に関する基本的な情報が入れられます。
- **副次 SQLVAR:** これらの項目は、上記で概説した規則にしたがって SQLVAR 項目の数が 2 倍になった場合にのみ存在します。ユーザー定義タイプ (特殊または構造) の場合は、ユーザー定義タイプ名が入ります。参照タイプの場合は、参照のターゲット・タイプが入れられます。LOB の場合は、ホスト変数の長さ属性と、実際の長さの入っているバッファーを指すポインターが入れられます。(特殊タイプと LOB の情報が重なり合うことはないので、DESCRIBE においては、SQLVAR 項目を 3 倍

SQLDA フィールド記述

にしなくても、LOB に基づく特殊タイプを使用できます。) ロケータまたはファイル参照変数を使用して LOB を示す場合、これらの項目は必要ありません。

SQLDA に上記 2 つのタイプの項目が両方とも含まれる場合、基本 SQLVAR は副次 SQLVAR のブロックの前のブロックに入れられます。それぞれの場合において、項目の数は SQLD の値で示されます (副次 SQLVAR 項目の多くは使用されない場合があります)。

DESCRIBE によって SQLVAR 項目が設定される環境については、1329 ページの『SQLDA に対する DESCRIBE の効果』に示されています。

SQLDA ヘッダーのフィールド:

表 238. SQLDA ヘッダーのフィールド

C での名前	SQL データ・ タイプ	DESCRIBE および PREPARE で 使用する場合 (SQLN を除き、 データベース・マネージャーにより 設定)	FETCH、OPEN、および EXECUTE で使用する場合 (ステートメントの 実行前にアプリケーションにより 設定)
sqldaid	CHAR(8)	このフィールドの 7 番目のバイトは、SQLDOUBLED という名前のフラグ・バイトです。データベース・マネージャーは、それぞれの列に対して 2 つの SQLVAR 項目が作成された場合は SQLDOUBLED を文字 '2' に設定し、その他の場合はブランク (ASCII では X'20'、EBCDIC では X'40' に設定します。 SQLDOUBLED がいつ設定されるかについては、1329 ページの『SQLDA に対する DESCRIBE の効果』を参照してください。	このフィールドの 7 番目のバイトは、SQLVAR の数が 2 倍になった場合に使用されます。これは SQLDOUBLED という名前のフィールドです。記述されるホスト変数が構造タイプ、BLOB、CLOB、または DBCLOB の場合、この 7 番目のバイトは文字 '2' に設定され、それ以外の場合は任意の文字に設定できます (ブランクの使用をお勧めします)。
sqldbcb	INTEGER	32 ビットの場合、SQLDA の長さ = SQLN*44+16。 64 ビットの場合、SQLDA の長さ = SQLN*56+16。	32 ビットの場合、SQLDA の長さ >= SQLN*44+16。 64 ビットの場合、SQLDA の長さ >= SQLN*56+16。
sqln	SMALLINT	データベース・マネージャーはこれを変更しません。 DESCRIBE ステートメントを実行する前に、ゼロまたはゼロより大きい値を設定する必要があります。これは、SQLVAR のオカレンスの合計数を示します。	SQLDA の SQLVAR のオカレンスの合計。 SQLN には、ゼロまたはゼロより大きい値を設定する必要があります。
sqld	SMALLINT	データベース・マネージャーによって、結果表の列の数またはパラメーター・マーカ―の数に設定されます。	SQLVAR のオカレンスにより記述されるホスト変数の数

基本 SQLVAR のオカレンスのフィールド:

表 239. 基本 SQLVAR のフィールド

資料名	データ・ タイプ	DESCRIBE および PREPARE で使用する 場合	FETCH、OPEN、および EXECUTE で使用する 場合
sqltype	SMALLINT	列のデータ・タイプと、その列または パラメーター・マーカが NULL 値 可能かどうかを示します。(パラメー ター・マーカは常に NULL 可能と 見なされます。) 1331 ページの表 241 は、許される値とその意味をリストし ています。 特殊タイプまたは参照タイプの場合 は、その基本タイプのデータ・タイプ がこのフィールドに入れられます。構 造タイプの場合は、そのタイプの変換 グループ (CURRENT DEFAULT TRANSFORM GROUP 特殊レジスタ ーに基づく) の FROM SQL 変換関数 が入れられます。基本 SQLVAR に は、それがユーザー定義タイプまたは 参照タイプの記述の一部であるかどう かを示す標識はありません。	ホスト変数の場合と同じ。日付/時刻 の値のホスト変数は、文字ストリン グ変数でなければなりません。FETCH の場合、日付/時刻のタイプ・コード は、固定長文字ストリングを意味しま す。sqltype が偶数値の場合、sqlind フィールドは無視されます。
sqllen	SMALLINT	列またはパラメーター・マーカ の長さ属性。日付/時刻の列または パラメーター・マーカの場合は、 値のストリンク表記の長さ。1331 ページの表 241 を参照してください。 ラージ・オブジェクト・ストリンク の場合、この値は 0 に設定されま す。その長さ属性が 2 バイト整数 に入る小さいものであっても、 設定値はやはり 0 になります。	ホスト変数の長さ属性。1331 ペ ージの表 241 を参照してください。 CLOB、DBCLOB、および BLOB の 列の場合、データベース・マネー ジャーはこの値を無視します。代 わりに、副次 SQLVAR の len.sql longlen フィールドが使用されま す。

基本 SQLVAR のオカレンスのフィールド

表 239. 基本 SQLVAR のフィールド (続き)

資料名	データ・タイプ	DESCRIBE および PREPARE で使用する場合	FETCH、OPEN、および EXECUTE で使用する場合
sqldata	ポインター	<p>文字列 SQLVARS の場合、sqldata にはコード・ページが含まれます。文字列 SQLVAR の場合、FOR BIT DATA 属性で列を定義すると、sqldata は 0 になります。ほかの文字列 SQLVARS の場合、sqldata には、SBCS データの SBCS コード・ページか、MBCS データの複合 MBCS コード・ページに関連づけられた SBCS コード・ページのいずれかが含まれています。日本語の EUC、中国語 (繁体字) の EUC、およびユニコードの UTF-8 文字列 SQLVARS では、sqldata にそれぞれ 954、964、および 1208 が含まれます。</p> <p>他のすべてのタイプの列の場合、sqldata は未定義です。</p>	<p>ホスト変数のアドレスを含みます (取り出したデータを保管するロケーション)。</p>
sqlind	ポインター	<p>文字列 SQLVARS の場合、sqlind は 0 になります。ただし、sqlind が複合 DBCS コード・ページに関連付けられた DBCS コード・ページの場合、MBCS データは例外です。</p> <p>他のすべてのタイプの列の場合、sqlind は未定義です。</p>	<p>関連する標識変数があれば、そのアドレスが入ります。それ以外の場合は、使用されません。sqltype が偶数値の場合、sqlind フィールドは無視されます。</p>
sqlname	VARCHAR (30)	<p>非修飾の列名またはパラメーター・マーカ名。</p> <p>システム生成の名前を持つ列およびパラメーター・マーカの場合、30 番目のバイトが X'FF' に設定されます。AS 文節によって列名が指定された場合は、このバイトは X'00' になります。</p>	<p>DB2 Connect を使用してサーバーにアクセスする場合は、FOR BIT DATA 文字列を指定するには、sqlname を以下のように設定します。</p> <ul style="list-style-type: none"> • sqlname の長さは 8 • sqlname の最初の 4 バイトは X'00000000' • sqlname の残りのバイトは予約済み (現在は無視される)

副次 SQLVAR のオカレンスのフィールド:

表 240. 副次 SQLVAR のフィールド

資料名	データ・タイプ	DESCRIBE および PREPARE で使用する場合	FETCH、OPEN、および EXECUTE で使用する場合
len.sqllonglen	INTEGER	BLOB、CLOB、または DBCLOB の列またはパラメーター・マーカ ーの長さ属性。	BLOB、CLOB、または DBCLOB ホスト変数の長さ属性。データベ ース・マネージャーは、それらの データ・タイプに対しては基本 SQLVAR の SQLLEN フィール ドを無視します。長さ属性は、 BLOB または CLOB ではバイト 数、DBCLOB では文字数になり ます。
reserve2	32 ビットの場合 は CHAR(3)、64 ビットの場合は CHAR(11)。	使用されません。	使用されません。
sqlflag4	CHAR(1)	SQLVAR の表している参照タイ プが sqldatatype_name に指定さ れたターゲット・タイプに関連付 けられたものである場合、この値 は X'01' になります。SQLVAR の表している構造タイプで、 sqldatatype_name にユーザー定義 タイプ名が指定されている場合、 値は X'12' になります。それ以 外の場合は、値は X'00' です。	SQLVAR の表している参照タイ プが sqldatatype_name に指定さ れたターゲット・タイプに関連付 けられたものである場合、X'01' に設定されます。SQLVAR の表 している構造タイプで、 sqldatatype_name にユーザー定義 タイプ名が指定されている場合、 X'12' に設定されます。それ以外 の場合は、値は X'00' です。

副次 SQLVAR のオカレンスのフィールド

表 240. 副次 SQLVAR のフィールド (続き)

資料名	データ・タイプ	DESCRIBE および PREPARE で使用する場合	FETCH、OPEN、および EXECUTE で使用する場合
sqldata1en	ポインタ	使用されません。	<p>BLOB、CLOB、および DBCLOB ホスト変数でのみ使用されます。</p> <p>このフィールドが NULL (NULL 値) の場合は、データの直前に実 際の長さ (文字単位) を 4 バイト で保管し、SQLDATA はフィー ルド長の最初のバイトを指すよう にする必要があります。</p> <p>このフィールドが NULL (NULL 値) でない場合は、対応する基本 SQLVAR 内の SQLDATA フィー ルドの指すバッファ内データの 実際の長さ (バイト単位、 DBCLOB の場合も含む) の入っ ている 4 バイト長のバッファ を指すポインタが入れられま す。</p> <p>このフィールドを使用するか否か に関係なく、len.sqllonglen フィ ールドは設定する必要があります。</p>
sqldatatype_name	VARCHAR(27)	<p>ユーザー定義タイプの場合、デー タベース・マネージャはこれを 完全修飾ユーザー定義タイプ名に 設定します。¹ 参照タイプの場合 は、データベース・マネージャ はこれを参照のターゲット・タイ プの完全修飾タイプ名に設定しま す。</p>	<p>構造タイプの場合、表の注¹ で示 されている形式の完全修飾ユーザ ー定義タイプ名に設定されます。</p>
予約済み	CHAR(3)	使用されません。	使用されません。

表 240. 副次 SQLVAR のフィールド (続き)

資料名	データ・タイプ	DESCRIBE および PREPARE で使用する場合	FETCH、OPEN、および EXECUTE で使用する場合
-----	---------	---------------------------------	-----------------------------------

¹ 最初の 8 バイトには、タイプのスキーマ名が入られます (必要に応じて右側にスペースが入られません)。バイト 9 はドット文字 (.) です。バイト 10 ~ 27 には、タイプ名のうちの下位部分が入られません。それは、右側にスペースを入れて拡張することはできません。

このフィールドの主な目的は、タイプの名前を入れることですが、IBM の定義済みデータ・タイプ用に設定することもできます。この場合、スキーマ名は SYSIBM、名前の下位部分は DATATYPES カタログ・ビューの TYPENAME 列に保管されている名前になります。たとえば:

type name	length	sqldatatype_name	
-----	-----	-----	-----
A.B	10	A	.B
INTEGER	16	SYSIBM	.INTEGER
"Frank's".SMINT	13	Frank's	.SMINT
MY."type "	15	MY	.type

SQLDA に対する DESCRIBE の効果

DESCRIBE OUTPUT または PREPARE OUTPUT INTO ステートメントの場合、データベース・マネージャーは、常に SQLD を結果セットの列の数、または出力パラメーター・マーカ―の数に設定します。DESCRIBE INPUT または PREPARE INPUT INTO ステートメントの場合、データベース・マネージャーは、常に SQLD をステートメント内の入力パラメーター・マーカ―の数に設定します。CALL ステートメント内の INOUT パラメーターに対応するパラメーター・マーカ―は、入力記述子と出力記述子の両方で記述されるので注意してください。

SQLDA の SQLVAR は、以下の場合に設定されます。

- $SQLN \geq SQLD$ で、しかも LOB、ユーザー定義タイプ、または参照タイプの項目がない
最初の SQLD SQLVAR 項目が設定され、SQLDOUBLED はブランクに設定されません。
- $SQLN \geq 2 * SQLD$ で、しかも少なくとも 1 つの項目が LOB、ユーザー定義タイプ、または参照タイプである
2 倍の数の SQLD SQLVAR 項目が設定され、SQLDOUBLED は '2' に設定されます。
- $SQLD \leq SQLN < 2 * SQLD$ で、しかも少なくとも 1 つの項目が特殊タイプまたは参照タイプで、LOB の項目または構造タイプの項目はない
最初の SQLD SQLVAR 項目が設定され、SQLDOUBLED はブランクに設定されます。SQLWARN バインド・オプションが YES の場合は、警告 SQLCODE +237 (SQLSTATE 01594) が出されます。

SQLDA に対する DESCRIBE の効果

SQLDA の SQLVAR は、以下の場合には設定されません (さらに多くのスペースの割り振りと別の DESCRIBE が必要)。

- SQLN < SQLD で、しかも LOB、ユーザー定義タイプ、または参照タイプの項目がない

SQLVAR 項目は設定されず、SQLDOUBLED はブランクに設定されます。

SQLWARN バインド・オプションが YES の場合は、警告 SQLCODE +236 (SQLSTATE 01005) が出されます。

DESCRIBE が正常に実行される場合には、SQLD 個の SQLVAR が割り振られます。

- SQLN < SQLD で、しかも少なくとも 1 つの項目が特殊タイプまたは参照タイプで、LOB の項目または構造タイプの項目はない

SQLVAR 項目は設定されず、SQLDOUBLED はブランクに設定されます。

SQLWARN バインド・オプションが YES の場合は、警告 SQLCODE +239 (SQLSTATE 01005) が出されます。

特殊タイプ名および参照タイプのターゲット・タイプを含む DESCRIBE が正常に実行されると、2*SQLD 個の SQLVAR が割り振られます。

- SQLN < 2*SQLD で、しかも少なくとも 1 つの項目が LOB または構造タイプである

SQLVAR 項目は設定されず、SQLDOUBLED はブランクに設定されます。

(SQLWARN バインド・オプションの設定に関係なく) 警告 SQLCODE +238 (SQLSTATE 01005) が出されます。

DESCRIBE が正常に実行される場合には、2*SQLD 個の SQLVAR が割り振られません。

上記リストでの「LOB 項目」には、ソース・タイプが LOB タイプである特殊タイプの項目も含まれます。

DESCRIBE (または PREPARE INTO) から警告 SQLCODE +236、+237、+239 を戻すかどうかを制御するには、BIND または PREP コマンドの SQLWARN オプションを使用します。使用するアプリケーション・コードでは、これらの SQLCODE がいつ戻されてもよいようにしておいてください。選択リストに LOB または構造タイプの項目が含まれていて、SQLDA 中の SQLVAR が不足している場合には、常に警告 SQLCODE +238 が戻されます。これは、結果セット内に LOB または構造タイプの項目があるために SQLVAR 数を 2 倍にする必要があることをアプリケーションに認識させる唯一の方法です。

構造タイプの項目を記述しようとしているものの、FROM SQL トランスフォームが定義されていない場合 (CURRENT DEFAULT TRANSFORM GROUP 特殊レジスターを使用した TRANSFORM GROUP の指定が行われていない (SQLSTATE 42741) か、またはその名前グループが FROM SQL トランスフォーム関数を定義していない (SQLSTATE 42744) ため)、DESCRIBE はエラーを戻します。このエラーは、構造タイプの項目がある表の DESCRIBE で戻されるエラーと同じです。

SQLTYPE と SQLLEN

表 241 に、SQLDA の SQLTYPE フィールドと SQLLEN フィールドに現れる値を示します。DESCRIBE と PREPARE INTO においては、SQLTYPE の値が偶数ならその列では NULL 値が使えないこと、また奇数ならその列で NULL 値が可能であることを意味しています。FETCH、OPEN、および EXECUTE において、SQLTYPE の値が偶数の場合には標識変数がないこと、奇数の場合には SQLIND に標識変数のアドレスが入れられていることを意味しています。

表 241. SQLTYPE と SQLLEN の値 (DESCRIBE、FETCH、OPEN、および EXECUTE の場合)

SQLTYPE	DESCRIBE および PREPARE INTO の場合		FETCH、OPEN、および EXECUTE の場合	
	列のデータ・ タイプ	SQLLEN	ホスト変数の データ・タイプ	SQLLEN
384/385	日付	10	日付の固定長文字 ストリング表示	ホスト変数の長さ 属性
388/389	time	8	時刻の固定長文字 ストリング表示	ホスト変数の長さ 属性
392/393	timestamp	26	タイム・スタンプ の固定長文字スト リング表示	ホスト変数の長さ 属性
396/397	DATALINK	列の長さ属性	DATALINK	ホスト変数の長さ 属性
400/401	N/A	N/A	NUL 文字で終了す る GRAPHIC スト リング	ホスト変数の長さ 属性
404/405	BLOB	0 *	BLOB	使用されません。*
408/409	CLOB	0 *	CLOB	使用されません。*
412/413	DBCLOB	0 *	DBCLOB	使用されません。*
448/449	可変長文字ストリ ング	列の長さ属性	可変長文字ストリ ング	ホスト変数の長さ 属性
452/453	固定長文字ストリ ング	列の長さ属性	固定長文字ストリ ング	ホスト変数の長さ 属性
456/457	長形式可変長文字 ストリング	列の長さ属性	長形式可変長文字 ストリング	ホスト変数の長さ 属性
460/461	N/A	N/A	NUL 文字で終了す る文字ストリング	ホスト変数の長さ 属性
464/465	可変長 GRAPHIC ストリング	列の長さ属性	可変長 GRAPHIC ストリング	ホスト変数の長さ 属性

SQLTYPE と SQLLEN

表 241. SQLTYPE と SQLLEN の値 (DESCRIBE、FETCH、OPEN、および EXECUTE の場合) (続き)

SQLTYPE	DESCRIBE および PREPARE INTO の場合		FETCH、OPEN、および EXECUTE の場合	
	列のデータ・ タイプ	SQLLEN	ホスト変数の データ・タイプ	SQLLEN
468/469	固定長 GRAPHIC ストリング	列の長さ属性	固定長 GRAPHIC ストリング	ホスト変数の長さ 属性
472/473	長形式可変長 GRAPHIC ストリン グ	列の長さ属性	長形式 GRAPHIC ストリング	ホスト変数の長さ 属性
480/481	浮動小数点数	倍精度の場合は 8、単精度の場合は 4	浮動小数点数	倍精度の場合は 8、単精度の場合は 4
484/485	パック 10 進数	バイト 1 は精度、 バイト 2 は位取り	パック 10 進数	バイト 1 は精度、 バイト 2 は位取り
492/493	64 ビット整数	8	64 ビット整数	8
496/497	64 ビット整数	4	64 ビット整数	4
500/501	短整数	2	短整数	2
916/917	該当なし	該当なし	BLOB ファイル参 照変数	267
920/921	該当なし	該当なし	CLOB ファイル参 照変数	267
924/925	該当なし	該当なし	DBCLOB ファイル 参照変数	267
960/961	該当なし	該当なし	BLOB ロケータ ー	4
964/965	該当なし	該当なし	CLOB ロケータ ー	4
968/969	該当なし	該当なし	DBCLOB ロケ ー	4

注:

- 副次 SQLVAR の len.sqllonglen フィールドに、列の長さ属性が入れます。
- SQLTYPE は、DB2 での移植性のために旧バージョンから変更されました。旧バージョンの値 (旧バージョンの SQL リファレンスを参照) は、引き続きサポートされています。

認識されない非サポート SQLTYPE: SQLDA の SQLTYPE フィールドに表示される値は、データの送信側および受信側で使用可能なデータ・タイプ・サポートのレベルによって異なります。これは、新しいデータ・タイプが製品に追加される場合に特に重要です。

新しいデータ・タイプは、データの送信側または受信側にサポートされることもあれば、サポートされないこともあり、データの送信側や受信側に認識されないことさえあります。状況に応じて、新しいデータ・タイプが戻されたり、送信側と受信側の両方が認めた互換データ・タイプが戻されたり、あるいは結果としてエラーが発生したりします。

送信側と受信局が互換データ・タイプの使用に同意する場合、以下に示すマッピングが実行されます。このマッピングは、送信側または受信側の少なくともどちらかが指定データ・タイプをサポートしない場合に実行されます。非サポート・データ・タイプは、アプリケーションまたはデータベース・マネージャーのどちらかによって提供されません。

データ・タイプ	互換データ・タイプ
BIGINT	DECIMAL(19, 0)
ROWID ¹	VARCHAR(40) FOR BIT DATA

¹ ROWID は、DB2 Universal Database (z/OS および OS/390 版) パージョン 6 によってサポートされています。

SQLDA では、データ・タイプが置換されたことは示されないので注意してください。

パック 10 進数: パック 10 進数は、一種の 2 進コードによる 10 進数 (BCD) 表記で保管されます。BCD においては、1 ニブル (4 ビット) で 10 進数の 1 桁が表されます。たとえば、0001 0111 1001 は 179 を表します。したがって、パック 10 進数の値はニブルごとに読む必要があります。値の保管はバイト単位で行い、16 進数表記としてそれらのバイトを読み、それを 10 進数に戻します。たとえば、0001 0111 1001 は、2 進表記では 00000001 01111001 となります。この数値を 16 進数として読むと、0179 になります。

小数点は、位取りによって決まります。たとえば、DEC(12,5) の列の場合、小数点より右側に 5 桁あることとなります。

符号は、桁数を表すニブルの右側のニブルで示します。正記号または負記号は、以下のように示します。

表 242. パック 10 進数の符号標識の値

符号	表記		
	2 進	10 進	16 進
正符号 (+)	1100	12	C
負符号 (-)	1101	13	D

まとめ:

- 値を保管するためには、 $p/2+1$ バイトを割り振ります。 p は精度です。

バック 10 進数

- 値を表すために、ニブルを左から右へ割り当てます。数値の精度が偶数の場合は、最初にニブルを追加します。この割り当てには、先行 (無効な) ゼロと後続 (有効な) ゼロの桁が含まれます。
- 符号ニブルは、最後のバイトの第 2 ニブルになります。

たとえば、以下のようになります。

列	値	バイトごとにグループにした 16 進のニブル
DEC(8,3)	6574.23	00 65 74 23 0C
DEC(6,2)	-334.02	00 33 40 2D
DEC(7,5)	5.2323	05 23 23 0C
DEC(5,2)	-23.5	02 35 0D

10 進数の SQLLEN フィールド: SQLLEN フィールドには、10 進数の列の精度 (第 1 バイト) と位取り (第 2 バイト) が入れられます。アプリケーションを移植可能にするには、精度のバイトと位取りのバイトを短整数として一度に設定するのではなく、個々に設定するようにしてください。これによって、整数のバイト反転の問題が回避されます。

たとえば、C の場合には以下のようにします。

```
((char *)&(sqlda->sqlvar[i].sqlllen))[0] = precision;  
((char *)&(sqlda->sqlvar[i].sqlllen))[1] = scale;
```

関連資料:

- *SQL* リファレンス 第 1 巻 の『CHAR スカラー関数』

SYSCAT.ROUTINEPARMS

SYSCAT.ROUTINES に定義されているルーチンの、パラメーターまたは結果ごとに、1 つの行が入れられます。(このカタログ・ビューは、SYSCAT.FUNCPARMS および SYSCAT.PROCPARMS に代わるものです。ほかのビューが存在するものの、DB2 バージョン 7.1 のものとして残っています。)

表 243. SYSCAT.ROUTINEPARMS カタログ・ビュー

列名	データ・タイプ	NULL 可能	説明
ROUTINESCHEMA	VARCHAR(128)		ルーチンの修飾名。
ROUTINENAME	VARCHAR(128)		
SPECIFICNAME	VARCHAR(128)		ルーチン・インスタンスの名前 (システム生成の場合もある)。

表 243. SYSCAT.ROUTINEPARMS カタログ・ビュー (続き)

列名	データ・タイプ	NULL 可能	説明
PARMNAME	VARCHAR(128)	Yes	パラメーターまたは結果の列の名前、あるいは名前が存在しない場合は NULL 値。
ROWTYPE	CHAR(1)		B = 入力と出力の両方のパラメーター C = キャスト後の結果 O = 出力パラメーター P = 入力パラメーター R = キャスト前の結果
ORDINAL	SMALLINT		ROWTYPE = B、O、または P の場合、ルーチン・シングニチャー内のパラメーターの位置番号。ROWTYPE= R であり、なおかつルーチンが表関数である場合、その結果表内の列の位置番号。それ以外の場合は 0。
TYPESHEMA	VARCHAR(128)		パラメーターまたは結果のデータ・タイプの修飾名。
TYPENAME	VARCHAR(128)		修飾名。
LOCATOR	CHAR(1)		Y = パラメーターまたは結果は、ロケータの形式で渡される N = パラメーターまたは結果は、ロケータの形式で渡されない
LENGTH	INTEGER		パラメーターまたは結果の長さ。パラメーターまたは結果が特殊タイプの場合は 0。注 1 を参照。
SCALE	SMALLINT		パラメーターまたは結果の位取り。パラメーターまたは結果が特殊タイプの場合は 0。注 1 を参照。
CODEPAGE	SMALLINT		パラメーターまたは結果のコード・ページ。該当しない場合、または FOR BIT DATA 属性を指定して宣言された文字データのパラメーターか結果の場合は 0。
CAST_FUNCSCHEMA	VARCHAR(128)	Yes	引き数また結果のキャストに使用される関数の修飾名。ソース関数および外部関数に適用され、それ以外の場合は NULL。
CAST_FUNCSPECIFIC	VARCHAR(128)	Yes	引き数また結果のキャストに使用される関数の修飾名。ソース関数および外部関数に適用され、それ以外の場合は NULL。
TARGET_TYPESHEMA	VARCHAR(128)	Yes	パラメーターまたは結果のタイプが REFERENCE の場合、ターゲット・タイプの修飾名。パラメーターまたは結果のタイプが REFERENCE でない場合は、NULL 値。
TARGET_TYPENAME	VARCHAR(128)	Yes	パラメーターまたは結果のタイプが REFERENCE でない場合は、NULL 値。

SYSCAT.ROUTINEPARMS

表 243. SYSCAT.ROUTINEPARMS カタログ・ビュー (続き)

列名	データ・タイプ	NULL 可能	説明
SCOPE_TABSCHEMA	VARCHAR(128)	Yes	パラメーターまたは結果のタイプが REFERENCE の場合、有効範囲 (ターゲット表) の修飾名。パラメーターまたは結果のタイプが REFERENCE でない場合や、有効範囲が定義されていない場合は、NULL 値。
SCOPE_TABNAME	VARCHAR(128)	Yes	
TRANSFORMGRPNAME	VARCHAR(128)	Yes	構造タイプの関数または結果の場合は、トランスフォーム・グループの名前。
REMARKS	VARCHAR(254)	Yes	パラメーターの注釈。

注:

1. ソース関数からの派生関数 (他の関数を参照して定義された関数) はソースのパラメーターの長さと同じ取りを継承するので、そのような関数の LENGTH と SCALE は 0 に設定されます。

SYSCAT.ROUTINES

ユーザー定義関数 (スカラー、表、またはソース)、システム生成メソッド、ユーザー定義メソッド、またはプロシージャごとに、1 つの行が含まれます。組み込み関数は含まれません。(このカタログ・ビューは、SYSCAT.FUNCTIONS および SYSCAT.PROCEDURES に代わるものです。ほかのビューが存在するものの、DB2 バージョン 7.1 のものとして残っています。)

表 244. SYSCAT.ROUTINES カタログ・ビュー

列名	データ・タイプ	NULL 可能	説明
ROUTINESCHEMA	VARCHAR(128)		ルーチンの修飾名。
ROUTINENAME	VARCHAR(128)		
ROUTINETYPE	CHAR(1)		F = 関数 M = メソッド P = プロシージャ
DEFINER	VARCHAR(128)		ルーチンの定義者の許可 ID。
SPECIFICNAME	VARCHAR(128)		ルーチン・インスタンスの名前 (システム生成の場合もある)。
ROUTINEID	INTEGER		内部割り当てによるルーチン ID。
RETURN_TYPESCHEMA	VARCHAR(128)	Yes	スカラー関数またはメソッドの場合、戻りタイプの修飾名。
RETURN_TYPENAME	VARCHAR(128)	Yes	

表 244. SYSCAT.ROUTINES カタログ・ビュー (続き)

列名	データ・タイプ	NULL 可能	説明
ORIGIN	CHAR(1)		B = 組み込み E = ユーザー定義、外部 M = テンプレート Q = SQL 本体 U = ユーザー定義、ソース関数に基づく S = システム生成 T = システム生成トランスフォーム
FUNCTIONTYPE	CHAR(1)		C = 列関数 R = 行関数 S = スカラー関数またはメソッド T = 表関数 ブランク = プロシージャ
PARAM_COUNT	SMALLINT		パラメーター数。
LANGUAGE	CHAR(8)		ルーチン本体のインプリメンテーション言語。可能な値は C、COBOL、JAVA、OLE、OLEDB、または SQL。ORIGIN が E または Q でない場合はブランク。
SOURCESCHEMA	VARCHAR(128)	Yes	ORIGIN=U かつルーチンがユーザー定義関数の場合、ソース関数の修飾名。ORIGIN=U で、しかもソース関数が組み込み関数の場合、SOURCESCHEMA は 'SYSIBM'、SOURCESPECIFIC は '組み込み関数の場合は N/A'。ORIGIN が U でない場合は、NULL 値。
SOURCESPECIFIC	VARCHAR(128)	Yes	
DETERMINISTIC	CHAR(1)		Y = 決定論 (結果は一貫している) N = 非決定論 (結果は毎回同じとは限らない) ORIGIN が E または Q でない場合はブランク。

SYSCAT.ROUTINES

表 244. SYSCAT.ROUTINES カタログ・ビュー (続き)

列名	データ・タイプ	NULL 可能	説明
EXTERNAL_ACTION	CHAR(1)		E = 関数に外部副次作用がある (呼び出しの数が重要) N = 副次作用がない ORIGIN が E または Q でない場合はブランク。
NULLCALL	CHAR(1)		Y = CALLED ON NULL INPUT N = RETURNS NULL ON NULL INPUT (オペランドに NULL 値が含まれる場合、暗黙のうちに結果は NULL 値) ORIGIN が E または Q でない場合はブランク。
CAST_FUNCTION	CHAR(1)		Y = cast 関数 N = cast 関数ではない
ASSIGN_FUNCTION	CHAR(1)		Y = 暗黙的な割り当て関数 N = 割り当て関数ではない
SCRATCHPAD	CHAR(1)		Y = スクラッチ・パッドあり N = スクラッチ・パッドなし ORIGIN が E または ROUTINETYPE が P でない場合はブランク。
SCRATCHPAD_LENGTH	SMALLINT		n = スクラッチパッドの長さ (バイト単位) 0 = SCRATCHPAD が N -1 = LANGUAGE が OLEDB
FINALCALL	CHAR(1)		Y = ステートメント終了の実行時に、この関数に対して最終呼び出しが行われる N = 最終呼び出しを行わない ORIGIN が E でない場合はブランク
PARALLEL	CHAR(1)		Y = 関数を並列して実行できる N = 関数を並列して実行できない ORIGIN が E でない場合はブランク

表 244. SYSCAT.ROUTINES カタログ・ビュー (続き)

列名	データ・タイプ	NULL 可能	説明
PARAMETER_STYLE	CHAR(8)		ルーチンの作成時に宣言されたパラメーターのスタイル。値: DB2SQL SQL DB2GENRL GENERAL JAVA DB2DARI GNRLNULL ORIGIN が E でない場合はブランク
FENCED	CHAR(1)		Y = 分離 N = 非分離 ORIGIN が E でない場合はブランク
SQL_DATA_ACCESS	CHAR(1)		C = CONTAINS SQL: SQL データの読み取りまたは変更を行わない SQL に限り許可されている。 M = MODIFIES SQL DATA: ルーチンで許可されているすべての SQL が許可されている。 N = NO SQL: SQL は許可されていない。 R = READS SQL DATA: SQL データを読み取る SQL に限り許可されている。
DBINFO	CHAR(1)		Y = DBINFO は渡される N = DBINFO は渡されない
PROGRAMTYPE	CHAR(1)		M = メイン S = サブルーチン
COMMIT_ON_RETURN	CHAR(1)		N = プロシージャラーの完了後に変更がコミットされない ROUTINETYPE が P でない場合はブランク
RESULT_SETS	SMALLINT		戻される結果セットの上限の見積もり。

SYSCAT.ROUTINES

表 244. SYSCAT.ROUTINES カタログ・ビュー (続き)

列名	データ・タイプ	NULL 可能	説明
SPEC_REG	CHAR(1)		I = INHERIT SPECIAL REGISTERS: ステートメントの呼び出しから、特殊レジスタをそれらの値を使用して開始する。 ORIGIN が E または Q でない場合はブランク。
FEDERATED	CHAR(1)		使用されません。
THREADSAFE	CHAR(1)		Y = ルーチンをほかルーチンと同じプロセスで実行できる。 N = ルーチンをほかのルーチンとは別のプロセスで実行しなければならない。 ORIGIN が E でない場合はブランク
VALID	CHAR(1)		Y = SQL プロシージャが有効。 N = SQL プロシージャが無効。 X = SQL プロシージャが必要とする何らかのオブジェクトがドロップされたので、その SQL プロシージャは作動不能。その SQL プロシージャを明示的にドロップし、再作成しなければならない。 ORIGIN が Q でない場合はブランク
METHODIMPLEMENTED	CHAR(1)		Y = メソッドがインプリメントされている。 N = インプリメントは行われずにメソッドが指定されている。 ROUTINETYPE が M でない場合はブランク
METHODEFFECT	CHAR(2)		MU = Mutator メソッド OB = オブザーバー・メソッド CN = コンストラクター・メソッド FUNCTIONTYPE が T ではない場合、ブランク。

表 244. SYSCAT.ROUTINES カタログ・ビュー (続き)

列名	データ・タイプ	NULL 可能	説明
TYPE_PRESERVING	CHAR(1)		Y = 戻りタイプは "type-preserving" パラメーターで管理される。システム生成の mutator メソッドはすべてタイプ保持です。 N = 戻りタイプはメソッドの宣言された戻りタイプ。 ROUTINETYPE が M でない場合はブランク
WITH_FUNC_ACCESS	CHAR(1)		Y = このメソッドは関数表記を使用して呼び出すことができます。 N = このメソッドは関数表記を使用して呼び出すことはできません。 ROUTINETYPE が M でない場合はブランク
OVERRIDEN_METHODID	INTEGER	Yes	将来の使用のために予約されています。
SUBJECT_TYPESHEMA	VARCHAR(128)	Yes	メソッドの対象となるタイプ。
SUBJECT_TYPENAME	VARCHAR(128)	Yes	
CLASS	VARCHAR(128)	Yes	LANGUAGE=JAVA の場合、このルーチンをインプリメントするクラス。それ以外の場合、NULL 値です。
JAR_ID	VARCHAR(128)	Yes	LANGUAGE=JAVA の場合、このルーチンをインプリメントする jar ファイル。それ以外の場合、NULL 値です。
JARSHEMA	VARCHAR(128)	Yes	LANGUAGE=JAVA の場合、このルーチンをインプリメントする jar ファイルのスキーマ。それ以外の場合、NULL 値です。
JAR_SIGNATURE	VARCHAR(1024)	Yes	LANGUAGE=JAVA の場合、このルーチンをインプリメントする Java メソッドのシグニチャー。それ以外の場合、NULL 値です。
CREATE_TIME	TIMESTAMP		ルーチン作成時のタイム・スタンプ。バージョン 1 の関数の場合は 0 に設定されます。
ALTER_TIME	TIMESTAMP		最新のルーチン変更のタイム・スタンプ。ルーチンが変更されていない場合、CREATE_TIME に設定します。
FUNC_PATH	VARCHAR(254)	Yes	ルーチンの定義された時点の SQL パス。

SYSCAT.ROUTINES

表 244. SYSCAT.ROUTINES カタログ・ビュー (続き)

列名	データ・タイプ	NULL 可能	説明
QUALIFIER	VARCHAR(128)		オブジェクト定義時のデフォルト・スキーマの値。
IOS_PER_INVOC	DOUBLE		呼び出しごとの入出力回数を見積もり。不明の場合は -1 (デフォルト値は 0)。
INSTS_PER_INVOC	DOUBLE		呼び出しごとの命令の数の見積もり。不明の場合は -1 (デフォルト値は 450)。
IOS_PER_ARGBYTE	DOUBLE		入力引き数 1 バイトごとの入出力回数を見積もり。不明の場合は -1 (デフォルト値は 0)。
INSTS_PER_ARGBYTE	DOUBLE		入力引き数 1 バイトごとの命令数の見積もり。不明の場合は -1 (デフォルト値は 0)。
PERCENT_ARGBYTES	SMALLINT		ルーチンが実際に読み取る入力引き数バイトの平均パーセント値の見積もり。不明の場合は -1 (デフォルト値は 100)。
INITIAL_IOS	DOUBLE		ルーチンが最初/最後に呼び出されたときに実行される入出力回数を見積もり。不明の場合は -1 (デフォルト値は 0)。
INITIAL_INSTS	DOUBLE		最初/最後のルーチン呼び出し時に実行される命令の数の見積もり。不明の場合は -1 (デフォルト値は 0)。
CARDINALITY	BIGINT		表関数の予測されるカーディナリティー。不明の場合、またはルーチンが表関数でない場合は -1。
SELECTIVITY	DOUBLE		ユーザー定義述部で使用。ユーザー定義述部がない場合は -1。注 1 を参照。
RESULT_COLS	SMALLINT		表関数 (ROUTINETYPE = F および TYPE = T) の場合は結果表の列数。ほかの関数またはメソッド (ROUTINETYPE = F または M) の場合は 1。プロシージャー (ROUTINETYPE = P) の場合は 0。
IMPLEMENTATION	VARCHAR(254)	Yes	ORIGIN=E の場合、この関数を実現するためのパス/モジュール/関数。ORIGIN=U、かつソース関数が組み込み関数の場合、ソース関数の名前とシングニチャー。それ以外の場合、NULL 値です。
LIB_ID	INTEGER	Yes	将来の使用のために予約されています。

表 244. SYSCAT.ROUTINES カタログ・ビュー (続き)

列名	データ・タイプ	NULL 可能	説明
TEXT_BODY_OFFSET	INTEGER		LANGUAGE = SQL の場合、CREATE ステートメントのテキスト全体の中の、SQL プロシージャ本体の開始位置に対するオフセット。LANGUAGE が SQL でない場合は 0。
TEXT	CLOB(2M)	Yes	LANGUAGE = SQL の場合、CREATE FUNCTION、CREATE METHOD、または CREATE PROCEDURE ステートメントのテキスト。
NEWSAVEPOINTLEVEL	CHAR(1)		呼び出し時にルーチンが新しい保管点レベルを開始するかどうかを指示します。 Y = ルーチンの呼び出し時に新しい保管点レベルが開始される。 N = ルーチンの呼び出し時に新しい保管点レベルが開始されない。ルーチンは既存の保管点レベルを使用します。 ブランク - 適用できない。
DEBUG_MODE	CHAR(3)		0 = このルーチンでデバッグをオフにする。 1 = このルーチンでデバッグをオンにする。
TRACE_LEVEL	CHAR(1)		将来の使用のために予約されています。
DIAGNOSTIC_LEVEL	CHAR(1)		将来の使用のために予約されています。
CHECKOUT_USERID	VARCHAR(128)	Yes	オブジェクトのチェックアウトを実行したユーザーのユーザー ID。チェックアウトが行われていない場合は NULL。
PRECOMPILE_OPTIONS	VARCHAR(1024)	Yes	ルーチンに指定されたプリコンパイル・オプション。
COMPILE_OPTIONS	VARCHAR(1024)	Yes	ルーチンに指定されたコンパイル・オプション。
REMARKS	VARCHAR(254)	Yes	ユーザー提供のコメントまたは NULL 値。

注:

1. どのようなユーザー定義ルーチンでも、バック記述子およびシステム・カタログでの移行中は、この列は -1 に設定されます。ユーザー定義述部の場合、システム・カタログでの選択性は -1 になります。この場合、オプティマイザーが使用する選択性の値は 0.01 です。

SYSCAT.INDEXES

表に定義されている索引 (継承された列の中で適用できるものを含む) ごとに 1 行が含まれます。

表 245. SYSCAT.INDEXES カタログ・ビュー

列名	データ・タイプ	NULL 可能	説明
INDSCHEMA	VARCHAR(128)		索引の名前。
INDNAME	VARCHAR(18)		
DEFINER	VARCHAR(128)		索引を作成したユーザー。
TABSCHEMA	VARCHAR(128)		その索引の定義されている表またはニックネームの修飾名。
TABNAME	VARCHAR(128)		
COLNAMES	VARCHAR(640)		列名の先頭に、昇順か降順かを示す + または - を付けたもののリスト。警告: この列は将来除去されます。詳しくは、SYSCAT.INDEXCOLUSE を参照してください。
UNIQUERULE	CHAR(1)		固有値に関する規則: D = 重複可 P = 1 次索引 U = 固有項目のみ可
MADE_UNIQUE	CHAR(1)		Y = 索引は元は非固有だったが、ユニーク・キー制約または主キー制約をサポートするために、ユニーク索引に変換された。制約がドロップされると、この索引は非固有に戻る。 N = 索引は作成時のまま。
COLCOUNT	SMALLINT		キー内の列数と組み込み列 (存在する場合) の数の合計。
UNIQUE_COLCOUNT	SMALLINT		ユニーク・キーに必要な列の数。常に <=COLCOUNT。組み込み列がある場合にのみ < COLCOUNT。索引にユニーク・キーがない場合は -1 (重複可能)。

表 245. SYSCAT.INDEXES カタログ・ビュー (続き)

列名	データ・タイプ	NULL 可能	説明
INDEXTYPE	CHAR(4)		索引のタイプ。 CLUS = クラスタリング REG = 通常 DIM = ディメンション・ブロック索引 BLOK = ブロック索引
ENTRYTYPE	CHAR(1)		H = 階層表の索引 L = 型付き表の論理索引 非型付き表の索引の場合は、ブランク
PCTFREE	SMALLINT		索引を最初に作成する際に予約する索引リーフ・ページのパーセンテージ。このスペースは、索引の作成後に行う挿入用に使用可能です。
IID	SMALLINT		索引の内部 ID。
NLEAF	INTEGER		リーフ・ページの数。統計が収集されていない場合は -1。
NLEVELS	SMALLINT		索引レベルの数。統計が収集されていない場合は -1。
FIRSTKEYCARD	BIGINT		最初のキーの値の種類数。統計が収集されていない場合は -1。
FIRST2KEYCARD	BIGINT		索引の最初の 2 つの列を使用するキーの種類数。統計がない場合、または適用されない場合は -1。
FIRST3KEYCARD	BIGINT		索引の最初の 3 つの列を使用するキーの種類数。統計がない場合、または適用されない場合は -1。
FIRST4KEYCARD	BIGINT		索引の最初の 4 つの列を使用するキーの種類数。統計がない場合、または適用されない場合は -1。
FULLKEYCARD	BIGINT		キー全体の値の種類数。統計が収集されていない場合は -1。

SYSCAT.INDEXES

表 245. SYSCAT.INDEXES カタログ・ビュー (続き)

列名	データ・タイプ	NULL 可能	説明
CLUSTERRATIO	SMALLINT		索引によるデータ・クラスタリングの程度。統計が収集されていない場合、または詳細な索引統計が収集されている場合は -1 (それらの場合は CLUSTERFACTOR のほうが使用されます)。
CLUSTERFACTOR	DOUBLE		クラスタリングの程度の詳細測定値。詳細索引統計が収集されていない場合、またはニックネームの索引が定義されていない場合は -1。
SEQUENTIAL_PAGES	INTEGER		索引キーの順序でディスクに存在し、それらの上に大きなギャップがなく、わずかなギャップしかないリーフ・ページの数。統計が使用できない場合は -1。
DENSITY	INTEGER		索引によって占有されているページの範囲内の、ページ数に対する SEQUENTIAL_PAGES の比率。パーセントで表現される (0 ~ 100 の整数。統計が入手できない場合は -1。)
USER_DEFINED	SMALLINT		この索引がユーザー定義であってドロップされていない場合は 1、それ以外の場合は 0。

表 245. SYSCAT.INDEXES カタログ・ビュー (続き)

列名	データ・タイプ	NULL 可能	説明
SYSTEM_REQUIRED	SMALLINT		<p>有効な値は以下のとおりです。</p> <p>以下の条件のいずれかまたはほかのものが当てはまる場合は 1。</p> <ul style="list-style-type: none"> - この索引が主キー制約またはユニーク・キー制約が必要である、またはこの索引が多次元クラスタリング (MDC) 表のディメンション・ブロック索引または複合ブロック索引である。 - これが型付き表の (OID) 列上の索引である。 <p>以下の条件の両方が当てはまる場合は 2。</p> <ul style="list-style-type: none"> - この索引が主キー制約またはユニーク・キー制約が必要である、またはこの索引が MDC 表のディメンション・ブロック索引または複合ブロック索引である。 - これが型付き表の (OID) 列上の索引である。 <p>それ以外の場合は 0。</p>
CREATE_TIME	TIMESTAMP		索引の作成された時刻。
STATS_TIME	TIMESTAMP	Yes	この索引について記録されている統計値が最後に変更された時刻。統計が使用可能でない場合は、NULL 値。
PAGE_FETCH_PAIRS	VARCHAR(254)		文字形式で表される、整数の対のリスト。それぞれの対は、仮のバッファ内のページ数と、その仮のバッファを使用した表のスキャンに必要なページ取り出しの回数を表しています。(データが入手できない場合は、長さ 0 のストリング。)
MINPCTUSED	SMALLINT		ゼロでない場合は、オンライン索引デフラグが使用可能になり、その値は、ページをマージをする前に使用される最小スペースのしきい値です。
REVERSE_SCANS	CHAR(1)		<p>Y = 索引は逆走査をサポートする</p> <p>N = 索引は逆走査をサポートしない</p>

SYSCAT.INDEXES

表 245. SYSCAT.INDEXES カタログ・ビュー (続き)

列名	データ・タイプ	NULL 可能	説明
INTERNAL_FORMAT	SMALLINT		有効な値は以下のとおりです。 索引に逆方向ポインターがない場合は 1。 索引に逆方向ポインターがある場合は >= 2。 索引が複合ブロック索引の場合は 6。
REMARKS	VARCHAR(254)	Yes	ユーザー提供のコメントまたは NULL 値。
IESHEMA	VARCHAR(128)	Yes	索引拡張の修飾名。通常の索引の場合は NULL。
IENAME	VARCHAR(18)	Yes	
IEARGUMENTS	CLOB(32K)	Yes	索引の作成時に指定されるパラメーターの外部情報。通常の索引の場合は NULL。
INDEX_OBJECTID	INTEGER		表の索引オブジェクト ID
NUMRIDS	BIGINT		索引内の行 ID (RID) の合計数。
NUMRIDS_DELETED	BIGINT		削除対象としてマークされている、索引内の行 ID の合計数 (すべての行 ID が削除対象としてマークされている、リーフ・ページ上の行 ID は除く)。
NUM_EMPTY_LEAFS	BIGINT		すべての行 ID が削除対象としてマークされている、索引リーフ・ページの合計数。
AVERAGE_RANDOM_FETCH_PAGES	DOUBLE		索引を使用してフェッチする際の、順次ページ・アクセス間のランダム表ページの平均数。不明の場合は -1。 注: DMS 表スペースの使用時には、この統計は計算されません。
AVERAGE_RANDOM_PAGES	DOUBLE		順次索引ページ・アクセス間のランダム索引ページの平均数。不明の場合は -1。
AVERAGE_SEQUENCE_GAP	DOUBLE		索引ページ・シーケンス間のキャップ。各ギャップは索引リーフ・ページのスキャンにより検出され、一連の索引ページの間でランダムにフェッチしなければならない索引ページの平均数を表します。不明の場合は -1。

表 245. SYSCAT.INDEXES カタログ・ビュー (続き)

列名	データ・タイプ	NULL 可能	説明
AVERAGE_SEQUENCE_ FETCH_GAP	DOUBLE		索引を使用してフェッチする際の、表ページ・シーケンス間のギャップ。各ギャップは索引リーフ・ページのスキャンにより検出され、一連の表ページの間でランダムにフェッチしなければならない表ページの平均数を表します。不明の場合は -1。 注: DMS 表スペースの使用時には、この統計は計算されません。
AVERAGE_SEQUENCE_ PAGES	DOUBLE		順次にアクセス可能な索引ページの平均数 (つまり、プリフェッチャーが順次として検出する索引ページの数)。不明の場合は -1。
AVERAGE_SEQUENCE_ FETCH_PAGES	DOUBLE		索引を使用してフェッチする際の、順次にアクセス可能な表ページの平均数 (つまり、プリフェッチャーが順次として検出する表ページの数)。不明の場合は -1。 注: DMS 表スペースの使用時には、この統計は計算されません。
TBSPACEID	INTEGER		索引表スペースの内部 ID。

関連資料:

- *SQL* リファレンス 第 1 巻 の『SYSCAT.INDEXCOLUSE カタログ・ビュー』

SYSSTAT.ROUTINES

ユーザー定義関数 (スカラー、表、またはソース)、システム生成メソッド、ユーザー定義メソッド、またはプロシージャごとに、1 つの行が含まれます。組み込み関数は含まれません。(SYSSTAT.FUNCTIONS はこのカタログ・ビューによって置き換えられています。他のビューも存在しますが、DB2 バージョン 7.1 におけるのと同じ状態のままです。)

表 246. SYSSTAT.ROUTINES カタログ・ビュー

列名	データ・ タイプ	NULL 可能	説明	更新可能
ROUTINESCHEMA	VARCHAR(128)		ルーチンの修飾名。	
ROUTINENAME	VARCHAR(128)			

SYSSTAT.ROUTINES

表 246. SYSSTAT.ROUTINES カタログ・ビュー (続き)

列名	データ・ タイプ	NULL 可能	説明	更新可能
ROUTINETYPE	CHAR(1)		F = 関数 M = メソッド P = プロシージャー	
SPECIFICNAME	VARCHAR(128)		ルーチン・インスタンスの名前 (システム生成の場合もある)。	
IOS_PER_INVOC	DOUBLE		呼び出しごとの入出力回数の見積もり。不明の場合は -1 (デフォルト値は 0)。この列は、-1 または ≥ 0 (ゼロ) でのみ更新できます。	Yes
INSTS_PER_INVOC	DOUBLE		呼び出しごとの命令の数の見積もり。不明の場合は -1 (デフォルト値は 450)。この列は、-1 または ≥ 0 (ゼロ) でのみ更新できます。	Yes
IOS_PER_ARGBYTE	DOUBLE		入力引き数 1 バイトごとの入出力回数 の見積もり。不明の場合は -1 (デフォルト値は 0)。この列は、-1 または ≥ 0 (ゼロ) でのみ更新できます。	Yes
INSTS_PER_ARGBYTE	DOUBLE		入力引き数 1 バイトごとの命令数の見積もり。不明の場合は -1 (デフォルト値は 0)。この列は、-1 または ≥ 0 (ゼロ) でのみ更新できます。	Yes
PERCENT_ARGBYTES	SMALLINT		ルーチンが実際に読み取る入力引き数バ イトの平均パーセント値の見積もり。不明の場合は -1 (デフォルト値は 100)。この列は、-1 か、あるいは 0 (ゼロ) と 100 の間の数でのみ更新できます。	Yes
INITIAL_IOS	DOUBLE		ルーチンが最初/最後に呼び出されたと きに実行される入出力回数の見積もり。不明の場合は -1 (デフォルト値は 0)。この列は、-1 または ≥ 0 (ゼロ) でのみ更新できます。	Yes
INITIAL_INSTS	DOUBLE		最初/最後のルーチン呼び出し時に実行 される命令の数の見積もり。不明の場合 は -1 (デフォルト値は 0)。この列 は、-1 または ≥ 0 (ゼロ) でのみ更新 できます。	Yes

表 246. SYSSTAT.ROUTINES カタログ・ビュー (続き)

列名	データ・ タイプ	NULL 可能	説明	更新可能
CARDINALITY	BIGINT		表関数の予測されるカーディナリティ。不明の場合、またはルーチンが表関数でない場合は -1。この列は、-1 または ≥ 0 (ゼロ) でのみ更新できます。	Yes
SELECTIVITY	DOUBLE		ユーザー定義述部に使用される。ユーザー定義述部がない場合は -1 です。注 1 を参照。	

注:

1. どのようなユーザー定義関数でも、システム・カタログでの DB2 バージョン 5.2 から 8.1 への移行中は、この列は -1 に設定されます。ユーザー定義述部の場合、システム・カタログでの選択性は -1 になります。この場合、オプティマイザが使用する選択性の値は 0.01 です。

SYSSTAT.INDEXES

表に定義されている各索引ごとに 1 つの行が含まれます。

表 247. SYSSTAT.INDEXES カタログ・ビュー

列名	データ・ タイプ	NULL 可能	説明	更新可能
INDSHEMA	VARCHAR(128)		索引の修飾名。	
INDNAME	VARCHAR(18)			
TABSHEMA	VARCHAR(128)		表名の修飾子。	
TABNAME	VARCHAR(128)		その索引の定義されている表またはニックネームの名前。	
COLNAMES	CLOB(1M)		+ または - の接頭部が付いた列名のリスト。	
NLEAF	INTEGER		リーフ・ページの数。統計が収集されていない場合は -1。 この列は、以下の値でのみ更新可能です。 • -1 または > 0 (ゼロ)	Yes

SYSSTAT.INDEXES

表 247. SYSSTAT.INDEXES カタログ・ビュー (続き)

列名	データ・ タイプ	NULL 可能	説明	更新可能
NLEVELS	SMALLINT		索引レベルの数。統計が収集されていない場合は -1。 この列は、以下の値でのみ更新可能です。 • -1 または > 0 (ゼロ)	Yes
FIRSTKEYCARD	BIGINT		最初のキーの値の種類数。統計が収集されていない場合は -1。 この列は、以下の値でのみ更新可能です。 • -1 または >= 0 (ゼロ)	Yes
FIRST2KEYCARD	BIGINT		索引の最初の 2 つの列を使用するキーの種類数 (統計がない場合、または適用されない場合は -1)。 この列は、以下の値でのみ更新可能です。 • -1 または >= 0 (ゼロ)	Yes
FIRST3KEYCARD	BIGINT		索引の最初の 3 つの列を使用するキーの種類数 (統計がない場合、または適用されない場合は -1)。 この列は、以下の値でのみ更新可能です。 • -1 または >= 0 (ゼロ)	Yes
FIRST4KEYCARD	BIGINT		索引の最初の 4 つの列を使用するキーの種類数 (統計がない場合、または適用されない場合は -1)。 この列は、以下の値でのみ更新可能です。 • -1 または >= 0 (ゼロ)	Yes

表 247. SYSSTAT.INDEXES カタログ・ビュー (続き)

列名	データ・ タイプ	NULL 可能	説明	更新可能
FULLKEYCARD	BIGINT		<p>キー全体の値の種類数。統計が収集されていない場合は -1。</p> <p>この列は、以下の値でのみ更新可能です。</p> <ul style="list-style-type: none"> • -1 または ≥ 0 (ゼロ) 	Yes
CLUSTERRATIO	SMALLINT		<p>この列は、オプティマイザーにより使用されます。索引のデータ・クラスタリングの程度を示し、統計が収集されていない場合、または詳細な索引統計が収集されている場合は -1。</p> <p>この列は、以下の値でのみ更新可能です。</p> <ul style="list-style-type: none"> • -1 または $0 \sim 100$ 	Yes
CLUSTERFACTOR	DOUBLE		<p>この列は、オプティマイザーにより使用されます。クラスタリングの程度の詳細測定値。詳細索引統計が収集されていない場合は -1。</p> <p>この列は、以下の値でのみ更新可能です。</p> <ul style="list-style-type: none"> • -1 または $0 \sim 1$ 	Yes
SEQUENTIAL_PAGES	INTEGER		<p>索引キーの順序でディスクに存在し、それらの間に大きなギャップがなく、わずかなギャップしかないリーフ・ページの数。統計が使用できない場合は -1。</p> <p>この列は、以下の値でのみ更新可能です。</p> <ul style="list-style-type: none"> • -1 または ≥ 0 (ゼロ) 	Yes

SYSSTAT.INDEXES

表 247. SYSSTAT.INDEXES カタログ・ビュー (続き)

列名	データ・ タイプ	NULL 可能	説明	更新可能
DENSITY	INTEGER		<p>索引によって占有されているページの範囲内の、ページ数に対する SEQUENTIAL_PAGES の比率。パーセントで表現される (0 ~ 100 の整数。統計が入手できない場合は -1。)</p> <p>この列は、以下の値でのみ更新可能です。</p> <ul style="list-style-type: none"> • -1 または 0 ~ 100 	Yes
PAGE_FETCH_PAIRS	VARCHAR(254)		<p>文字形式で表される、整数の対のリスト。それぞれの対は、仮のバッファのページ数と、その仮のバッファを使用した索引のスキャンに必要なページ取り出しの回数を表しています。(データが入手できない場合は、長さ 0 のストリング。)</p> <p>この列は、以下の入力値でのみ更新できます。</p> <ul style="list-style-type: none"> • 対の区切り文字および対の区切り記号は、数字以外の文字だけが受け入れられます。 • ブランクは、対の区切り文字および対の区切り記号として認識される唯一の文字です。 • 各数値項目には、対応するパートナーの数値項目がなければならず、それら 2 つは対区切り記号で区切る必要があります。 • 対と対の間は、対区切り文字で区切る必要があります。 • 予期される数値項目は、0 ~ 9 の正の数でなければなりません。 	Yes
NUMRIDS	BIGINT		<p>索引の中の RID の数。統計が収集されていない場合は -1。</p>	Yes

表 247. SYSSTAT.INDEXES カタログ・ビュー (続き)

列名	データ・ タイプ	NULL 可能	説明	更新可能
NUMRIDS_DELETED	BIGINT		削除済みとしてマークされた索引の中の RID の数 (ただし、すべての RID が削除済みとしてマークされたページの RID は除く)。統計が収集されていない場合は -1。	Yes
NUM_EMPTY_LEAFS	BIGINT		すべての RID が削除済みとしてマークされた索引内のリーフ・ページの数。統計が収集されていない場合は -1。	Yes
AVERAGE_RANDOM_ FETCH_PAGES	DOUBLE		索引を使用してフェッチする際の、順次ページ・アクセス間のランダム表ページの平均数。不明の場合は -1。 注: DMS 表スペースの使用時には、この統計は計算されません。	
AVERAGE_RANDOM_ PAGES	DOUBLE		順次索引ページ・アクセス間のランダム索引ページの平均数。不明の場合は -1。	
AVERAGE_SEQUENCE_ GAP	DOUBLE		索引ページ・シーケンス間のギャップ。各ギャップは索引リーフ・ページのスキャンにより検出され、一連の索引ページの間でランダムにフェッチしなければならない索引ページの平均数を表します。不明の場合は -1。	
AVERAGE_SEQUENCE_ FETCH_GAP	DOUBLE		索引を使用してフェッチする際の、表ページ・シーケンス間のギャップ。各ギャップは索引リーフ・ページのスキャンにより検出され、一連の表ページの間でランダムにフェッチしなければならない表ページの平均数を表します。不明の場合は -1。 注: DMS 表スペースの使用時には、この統計は計算されません。	
AVERAGE_SEQUENCE_ PAGES	DOUBLE		順次にアクセス可能な索引ページの平均数 (つまり、プリフェッチャーが順次として検出する索引ページの数)。不明の場合は -1。	

SYSSTAT.INDEXES

表 247. SYSSTAT.INDEXES カタログ・ビュー (続き)

列名	データ・ タイプ	NULL 可能	説明	更新可能
AVERAGE_SEQUENCE_ FETCH_PAGES	DOUBLE		索引を使用してフェッチする際の、順次にアクセス可能な表ページの平均数 (つまり、プリフェッチャーが順次として検出する表ページの数)。不明の場合は -1。 注: DMS 表スペースの使用時には、この統計は計算されません。	

SYSCAT.FUNCMAPPINGS

各行には関数マッピングが含まれています。

表 248. SYSCAT.FUNCMAPPINGS カタログ・ビュー

列名	データ・タイプ	NULL 可能	説明
FUNCTION_ MAPPING	VARCHAR(128)		関数マッピングの名前 (システム生成の場合もある)。
FUNCSHEMA	VARCHAR(128)	Yes	関数のスキーマ。システム組み込み関数の場合は NULL 値。
FUNCNAME	VARCHAR(1024)	Yes	ローカル関数 (組み込みまたはユーザー定義) の名前。
FUNCID	INTEGER	Yes	内部的に割り当てられた ID。
SPECIFICNAME	VARCHAR(128)	Yes	ローカル関数インスタンスの名前。
DEFINER	VARCHAR(128)		このマッピングの作成に使用された許可 ID。
WRAPNAME	VARCHAR(128)	Yes	マッピングが適用されるラッパー名。
SERVERNAME	VARCHAR(128)	Yes	データ・ソースの名前。
SERVERTYPE	VARCHAR(30)	Yes	マッピングが適用されるデータ・ソースのタイプ。
SERVERVERSION	VARCHAR(18)	Yes	マッピングが適用されるサーバー・タイプのバージョン。
CREATE_TIME	TIMESTAMP	Yes	マッピングが作成された時刻。
REMARKS	VARCHAR(254)	Yes	ユーザー提供のコメントまたは NULL 値。

SYSCAT.CASTFUNCTIONS

cast 関数ごとに 1 つの行が含まれています。組み込み cast 関数は含まれません。

表 249. SYSCAT.CASTFUNCTIONS カタログ・ビュー

列名	データ・ タイプ	NULL 可能	説明
FROM_TYPESHEMA	VARCHAR(128)		パラメーターのデータ・タイプの修飾名。
FROM_TYPENAME	VARCHAR(128)		
TO_TYPESHEMA	VARCHAR(128)		キャスト後の結果のデータ・タイプを示す修飾名。
TO_TYPENAME	VARCHAR(128)		
FUNCSHEMA	VARCHAR(128)		関数の修飾名。
FUNCNAME	VARCHAR(128)		
SPECIFICNAME	VARCHAR(128)		関数インスタンスの名前。
ASSIGN_FUNCTION	CHAR(1)		Y = 暗黙的な割り当て関数 N = 割り当て関数ではない

SYSCAT.TRANSFORMS

指名されたトランスフォーム・グループに含まれているユーザー定義タイプ内のトランスフォーム関数タイプごとに、1 つの行が含まれます。

表 250. SYSCAT.TRANSFORMS カタログ・ビュー

列名	データ・タイプ	NULL 可能	説明
TYPEID	SMALLINT		SYSCAT.DATATYPES で定義されている内部タイプ ID。
TYPESHEMA	VARCHAR(128)		特定のユーザー定義構造タイプの修飾名。
TYPENAME	VARCHAR(128)		
GROUPNAME	VARCHAR(128)		トランスフォーム・グループ名。
FUNCID	INTEGER	Yes	SYSCAT.ROUTINES で定義されている、関連するトランスフォーム関数の内部ルーチン ID。NULL になるのは、内部システム関数の場合だけです。
FUNCSHEMA	VARCHAR(128)		関連するトランスフォーム関数の修飾名。
FUNCNAME	VARCHAR(128)		
SPECIFICNAME	VARCHAR(128)		関数の特定名 (インスタンス名)。

SYSCAT.TRANSFORMS

表 250. SYSCAT.TRANSFORMS カタログ・ビュー (続き)

列名	データ・タイプ	NULL 可能	説明
TRANSFORMTYPE	VARCHAR(8)		'FROM SQL' = トランスフォーム関数は SQL から構造タイプをトランスフォームする 'TO SQL' = トランスフォーム関数は SQL に構造タイプをトランスフォームする
FORMAT	CHAR(1)		'U' = ユーザー定義
MAXLENGTH	INTEGER	Yes	FROM SQL トランスフォームからの出力の最大長 (バイト単位)。 TO SQL トランスフォームの場合は NULL。
ORIGIN	CHAR(1)		'O' = オリジナル・トランスフォーム・グループ (ユーザー定義またはシステム定義) 'R' = 再定義
REMARKS	VARCHAR(254)	Yes	ユーザー提供コメントまたは NULL。

SYSCAT.PREDICATESPECS

各行は述部指定を表します。

表 251. SYSCAT.PREDICATESPECS カタログ・ビュー

列名	データ・タイプ	NULL 可能	説明
FUNCSHEMA	VARCHAR(128)		関数の修飾名。
FUNCNAME	VARCHAR(128)		
SPECIFICNAME	VARCHAR(128)		関数インスタンスの名前。
FUNCID	INTEGER		関数 ID。
SPECID	SMALLINT		この述部指定の ID。
CONTEXTOP	CHAR(8)		比較演算子は組み込み関係演算子 (=、<、>= など) のいずれか。
CONTEXTEXP	CLOB(32K)		定数、または SQL 式。
FILTERTEXT	CLOB(32K)	Yes	データ・フィルター式のテキスト。

SYSCAT.SEQUENCES

データベースで定義された各シーケンスまたは識別列の行が含まれています。

表 252. SYSCAT.SEQUENCES カタログ・ビューの列

列名	データ・タイプ	NULL 可能	説明
SEQSCHEMA	VARCHAR(128)		シーケンスの修飾名 (ID 列に DB2 が生成した名前)。
SEQNAME	VARCHAR(128)		
DEFINER	VARCHAR(128)		シーケンスの定義者。
OWNER	VARCHAR(128)		シーケンスの所有者。
SEQID	INTEGER		シーケンスの内部 ID。
SEQTYPE	CHAR(1)		シーケンス・タイプ S = 通常のシーケンス I = ID シーケンス
INCREMENT	DECIMAL(31,0)		増分値
START	DECIMAL(31,0)		開始値
MAXVALUE	DECIMAL(31,0)		最大値
MINVALUE	DECIMAL(31,0)		最小値
CYCLE	CHAR(1)		境界に達したときにサイクリングを行うかどうか。 Y - サイクリングを行う N - サイクリングを行わない
CACHE	INTEGER		アクセスを高速化するために、メモリーに事前割り当てするシーケンス値の数。0 は値が事前割り当てされないことを示します。
ORDER	CHAR(1)		シーケンス番号を要求の順序で生成するかどうか。 Y - 要求の順序でシーケンス番号を生成します。 N - 要求の順序でシーケンス番号を生成しません。
DATATYPEID	INTEGER		組み込みタイプの場合は、組み込みタイプの内部 ID。特殊タイプの場合は、特殊タイプの内部 ID。
SOURCETYPEID	INTEGER		組み込みタイプの場合、この値は 0 になります。特殊タイプの場合は、特殊タイプのソース・タイプである組み込みタイプの内部 ID です。
CREATE_TIME	TIMESTAMP		シーケンスが作成された時刻。

SYSCAT.SEQUENCES

表 252. SYSCAT.SEQUENCES カタログ・ビューの列 (続き)

列名	データ・タイプ	NULL 可能	説明
ALTER_TIME	TIMESTAMP		最後の ALTER SEQUENCE ステートメントがこのシーケンスについて実行された時刻。
PRECISION	SMALLINT		シーケンスのデータ・タイプの精度。 SMALLINT では 5、INTEGER では 10、 BIGINT では 19 になります。DECIMAL では、指定した DECIMAL データ・タイプの精度です。
ORIGIN	CHAR(1)		シーケンス原点 U - ユーザー生成シーケンス S - システム生成シーケンス
REMARKS	VARCHAR(254)	Yes	ユーザー提供のコメントまたは NULL 値。

ルーチンで使用可能な SQL ステートメント

下記の表は、SQL ステートメント (第 1 列で指定されている) を、指定された SQL データ・アクセス指示を使ってルーチンで実行できるかどうかを示します。NO SQL と定義されたルーチン内に実行可能な SQL ステートメントが出現すると、SQLSTATE 38001 が戻されます。その他の実行コンテキストの場合、どのコンテキストでもサポートされていない SQL ステートメントは SQLSTATE 38003 を戻します。CONTAINS SQL コンテキスト内で使えないその他の SQL ステートメントの場合は SQLSTATE 38004 が戻されます。READS SQL DATA コンテキストの場合は SQLSTATE 38002 が戻されます。SQL ルーチンの作成中、SQL データ・アクセス指示と一致しないステートメントによって、SQLSTATE 42985 が戻されます。

ステートメントがルーチンを呼び出す際、ステートメントの有効 SQL データ・アクセス指示は、以下の SQL データ・アクセス指示よりも優先されます。

- 下記の表のステートメントの SQL データ・アクセス指示。
- ルーチンの作成時に指定されたルーチンの SQL データ・アクセス指示。

たとえば、CALL ステートメントには CONTAINS SQL の SQL データ・アクセス指示があります。ただし、READS SQL DATA として定義されたストアド・プロシージャが呼び出されると、CALL ステートメントの有効 SQL データ・アクセス指示は READS SQL DATA になります。

ルーチンが SQL ステートメントを呼び出す際、ステートメントの有効 SQL データ・アクセス指示は、ルーチンに対して宣言された SQL データ・アクセス指示を超えてはなりません。たとえば、READS SQL DATA として定義された関数は、MODIFIES SQL DATA として定義されたストアド・プロシージャを呼び出せません。

表 253. SQL ステートメントと SQL データ・アクセス指示

SQL ステートメント	NO SQL	CONTAINS SQL	READS SQL DATA	MODIFIES SQL DATA
ALTER...	N	N	N	Y
BEGIN DECLARE SECTION	Y(1)	Y	Y	Y
CALL	N	Y	Y	Y
CLOSE CURSOR	N	N	Y	Y
COMMENT ON	N	N	N	Y
COMMIT	N	N(4)	N(4)	N(4)
COMPOUND SQL	N	Y	Y	Y
CONNECT(2)	N	N	N	N
CREATE	N	N	N	Y
DECLARE CURSOR	Y(1)	Y	Y	Y
DECLARE GLOBAL TEMPORARY TABLE	N	N	N	Y
DELETE	N	N	N	Y
DESCRIBE	N	Y	Y	Y
DISCONNECT(2)	N	N	N	N
DROP ...	N	N	N	Y
END DECLARE SECTION	Y(1)	Y	Y	Y
EXECUTE	N	Y(3)	Y(3)	Y
EXECUTE IMMEDIATE	N	Y(3)	Y(3)	Y
EXPLAIN	N	N	N	Y
FETCH	N	N	Y	Y
FREE LOCATOR	N	Y	Y	Y
FLUSH EVENT MONITOR	N	N	N	Y
GRANT ...	N	N	N	Y
INCLUDE	Y(1)	Y	Y	Y
INSERT	N	N	N	Y
LOCK TABLE	N	Y	Y	Y
OPEN CURSOR	N	N	Y	Y
PREPARE	N	Y	Y	Y
REFRESH TABLE	N	N	N	Y

ルーチンで使用可能な SQL ステートメント

表 253. SQL ステートメントと SQL データ・アクセス指示 (続き)

SQL ステートメント	NO SQL	CONTAINS SQL	READS SQL DATA	MODIFIES SQL DATA
RELEASE CONNECTION(2)	N	N	N	N
RELEASE SAVEPOINT	N	N	N	Y
RENAME TABLE	N	N	N	Y
REVOKE ...	N	N	N	Y
ROLLBACK	N	N(4)	N(4)	N(4)
ROLLBACK TO SAVEPOINT	N	N	N	Y
SAVEPOINT	N	N	N	Y
SELECT INTO	N	N	Y	Y
SET CONNECTION(2)	N	N	N	N
SET INTEGRITY	N	N	N	Y
SET 特殊レジスター	N	Y	Y	Y
UPDATE	N	N	N	Y
VALUES INTO	N	N	Y	Y
WHENEVER	Y(1)	Y	Y	Y

注:

1. NO SQL オプションは SQL ステートメントを指定できないことを暗黙指定しますが、実行不能ステートメントに対する制限はありません。
2. どのルーチン実行コンテキストでも、接続管理ステートメントは使えません。
3. これは、実行しようとするステートメントによって異なります。EXECUTE ステートメントで指定するステートメントは、有効な個々の SQL アクセス・レベルのコンテキストで使えるものでなければなりません。たとえば、有効な SQL アクセス・レベルが READS SQL DATA である場合、ステートメントは INSERT、UPDATE、または DELETE であってはなりません。
4. TO SAVEPOINT 文節以外の COMMIT ステートメントおよび ROLLBACK ステートメントは、ストアード・プロシージャで使用できます。ただし、ストアード・プロシージャがアプリケーションから直接、またはアプリケーションからネスト・ストアード・プロシージャ呼び出しを介して間接的に呼び出される場合に限りです。(トリガー、関数、メソッド、またはアトミック・コンパウンド・ステートメントがストアード・プロシージャのいずれかが呼び出しチェーンにある場合、作業単位の COMMIT または ROLLBACK は使用できません。)

DATALINK 値

DATALINK 値はカプセル化された値で、データベース以外のロケーションに保管されているファイルへのデータベースからの論理参照を含んでいます。このカプセル化された値の属性は以下のとおりです。

link type

現在サポートされているリンクのタイプは 'URL' です。

data location

DB2 内でリンクで参照されているファイルのロケーション。これは、URL 形式になります。この URL に許可されているスキーム名は、以下のとおりです。

- HTTP
- FILE
- UNC

URL の他の部分は、以下のとおりです。

- HTTP、FILE、および UNC スキームのファイル・サーバー名
- ファイル・サーバー内の絶対ファイル・パス名

comment

データ位置属性を含む、最大 200 バイトの記述情報。これは、アプリケーションの特定の使用 (データのロケーションを詳細に識別する、または別の方法で識別するなど) のためのものです。

URL によるデータ・ロケーション属性を解析しているときに、前後の空白文字は切り詰められます。さらに、スキーム名 ('http'、'file'、'unc') およびホストは、大文字小文字を区別しないで、常に英大文字でデータベース内に格納されます。データベースから DATALINK 値を取り出すと、DATALINK 列が READ PERMISSION DB または WRITE PERMISSION ADMIN で定義される場合には、URL 属性内にアクセス・トークンが組み込まれます。トークンは動的に生成され、データベースに保管される DATALINK 値に永続的に組み込まれているわけではありません。

DATALINK 値に指定されているのがコメント属性だけで、データ・ロケーション属性が空である場合もあります。そのような値が列に保管されていても、当然そのような列にリンクされているファイルはありません。DATALINK 値のコメントおよびデータ・ロケーション属性の全長は、今のところ、200 バイトまでに制限されています。

ファイルへの DATALINK 参照と、LOB ファイル参照変数との違いを理解することは大切です。似ている点は、それらが両方ともファイルの表記を含んでいるということです。しかし、以下の点で異なります。

- DATALINK はデータベースに保存されるので、リンクおよびそのリンクされたファイルにあるデータは両方とも、データベース内のデータの自然な拡張と見なすことができます。

DATALINK 値

- ファイル参照変数はクライアント上に一時的に存在するものなので、ホスト・プログラムのバッファの代わりと見なすことができます。

組み込みスカラー関数は、DATALINK 値 (DLVALUE、DLNEWCOPY、DLPREVIOUSCOPY、および DLREPLACECONTENT) の作成と、DATALINK 値 (DLCOMMENT、DLINKTYPE、DLURLCOMPLETE、DLURLPATH、DLURLPATHONLY、DLURLSCHEME、DLURLSERVER、DLURLCOMPLETEONLY、DLURLCOMPLETEWRITE、および DLURLPATHWRITE) からのカプセル化された値の抽出を目的として提供されています。

関連資料:

- *SQL* リファレンス 第 1 巻 の『ID』
- 1078 ページの『DATALINK のバックス正規形式 (BNF) 仕様』

割り当てと比較

SQL の基本的な演算は、割り当てと比較です。割り当て演算は、INSERT、UPDATE、FETCH、SELECT INTO、VALUES INTO および SET 遷移変数ステートメントの実行時に行われます。関数の引き数も、関数の呼び出し時に割り当てられます。比較演算は、MAX、MIN、DISTINCT、GROUP BY、ORDER BY のような述部およびその他の言語エレメントを含むステートメントの実行時に行われます。

両方の演算に適用される 1 つの基本的な規則は、関係するオペランドのデータ・タイプは互換でなければならないということです。この互換性規則は集合演算にも適用されます。

割り当て操作の別の基本的な規則は、NULL 値を入れることができない列や、関連する標識変数がないホスト変数に、NULL 値を割り当てることができないという規則です。

文字とグラフィック・データの両方を含む割り当ておよび比較は、ストリングの 1 つがリテラルであるときにのみサポートされます。

以下は、割り当ておよび比較演算のためのデータ・タイプの互換性を示す互換性マトリックスです。

表 254. 割り当てと比較におけるデータ・タイプの互換性

オペランド	2 進 整数	10 進 数	浮動小数 点数	文字 ストリング	GRAPHIC ストリング	日付	時刻	タイム・ スタンプ	2 進 ストリング	UDT
2 進整数	Yes	Yes	Yes	No	No	No	No	No	No	2
10 進数	Yes	Yes	Yes	No	No	No	No	No	No	2
浮動小数 点 数	Yes	Yes	Yes	No	No	No	No	No	No	2

表 254. 割り当てと比較におけるデータ・タイプの互換性 (続き)

オペランド	2進 整数	10進 数	浮動小数 点数	文字 string	GRAPHIC string	日付	時刻	タイム・ スタンプ	2進 string	UDT
文字string	No	No	No	Yes	Yes ^{6,7}	1	1	1	No ³	²
GRAPHIC string	No	No	No	Yes ^{6,7}	Yes	1	1	1	No	²
日付	No	No	No	1	1	Yes	No	No	No	²
時刻	No	No	No	1	1	No	Yes	No	No	²
タイム・ スタンプ	No	No	No	1	1	No	No	Yes	No	²
2進string	No	No	No	No ³	No	No	No	No	Yes	²
UDT	²	²	²	²	²	²	²	²	²	Yes

割り当てと比較

表 254. 割り当てと比較におけるデータ・タイプの互換性 (続き)

オペランド	2 進 整数	10 進 数	浮動小数 点数	文字 ストリング	GRAPHIC ストリング	日付 時刻	タイム・2 進 スタンプ	ストリング	UDT
-------	-----------	-----------	------------	-------------	------------------	----------	-----------------	-------	-----

¹ 日付/時刻値とストリングの互換性は、割り当てと比較に限定されます。

- 日付/時刻値は、ストリング列とストリング変数に割り当てることができます。
- 日付の有効なストリング表記は、日付列に割り当てるか、または日付と比較できます。
- 時刻の有効なストリング表記は、時刻列に割り当てるか、または時刻と比較できます。
- タイム・スタンプの有効なストリング表記は、タイム・スタンプ列に割り当てるか、またはタイム・スタンプと比較できます。

(GRAPHIC ストリング・サポートは、ユニコード・データベースの場合にのみ使用可能です。)

² ユーザー定義特殊タイプの値は、同じユーザー定義特殊タイプで定義された値とのみ比較できます。一般に、特殊タイプの値とそのソース・データ・タイプの間では割り当てがサポートされます。ユーザー定義構造タイプは比較することができません。また、同じ構造タイプまたはそのスーパータイプのいずれかのおペラントにのみ、割り当てることができます。さらに詳しい情報については、1374 ページの『ユーザー定義タイプの割り当て』を参照してください。

³ これは、FOR BIT DATA 属性で定義された文字ストリングも 2 進ストリングと互換性がないことを意味します。

⁴ DATALINK オペラントは、別の DATALINK オペラントにのみ割り当てることができます。列が NO LINK CONTROL と定義されている場合、またはファイルは存在しているがまだファイル・リンク制御されていない場合、DATALINK 値はその列にしか割り当てることができません。

⁵ 参照タイプの割り当ておよび比較については、1375 ページの『参照タイプの割り当て』および 1381 ページの『参照タイプの比較』を参照してください。

⁶ ユニコード・データベースの場合にのみサポートされます。

⁷ ビット・データと GRAPHIC ストリングは互換性がありません。

数値割り当て

数値割り当ての基本的な規則は、10 進数または整数の整数部分は決して切り捨てられないことがないということです。宛先数値の位取りが割り当て元の数値の位取りより小さい場合は、10 進数の小数部の超過桁が切り捨てられます。

10 進数または整数から浮動小数点への割り当て: 浮動小数点数は、実数の近似値です。したがって、10 進数または整数が浮動小数点の列または変数に割り当てられた場合、その結果が元の数値と異なってくる可能性があります。

浮動小数点または 10 進数の整数への割り当て: 浮動小数点または 10 進数が、整数の列または整数変数に割り当てられた場合、その数の小数部分が失われます。

10 進数から 10 進数への割り当て: 10 進数が 10 進数の列または変数に割り当てられる場合、その数値は、必要に応じて宛先の精度および位取りに変換されます。必要な数だけ先行ゼロが追加または除去されます。また、数値の小数部分では、必要な数の後続ゼロが追加されるか、または必要な数だけ後続ブランクが除去されます。

整数から 10 進数への割り当て: 整数を 10 進数の列または変数に割り当てるときは、まず数値が一時的な 10 進数へ変換された後、必要であれば、宛先の精度と位取りに変換されます。一時的な 10 進数の精度と位取りは、短整数では 5,0、長精度整数では 11,0、64 ビット整数では 19,0 です。

浮動小数点数から 10 進数への割り当て: 浮動小数点数を 10 進数に割り当てるときは、まず数値が精度 31 の一時的な 10 進数へ変換された後、必要なら、宛先の精度と位取りまで切り捨てられます。この変換では、数値は精度 31 の 10 進数に (浮動小数点数算術演算を使って) 丸められます。その結果、 0.5×10^{-31} 未満の数値は 0 になります。位取りは、有効数字を消失させることなく数値全体を表現できるような最大可能値になります。

ストリング割り当て

割り当てには以下の 2 つのタイプがあります。

- 値を列またはルーチン・パラメーターに割り当てての場合のストレージ割り当て
- 値をホスト変数に割り当てての場合の検索割り当て

ストリング割り当てに関する規則は、割り当てのタイプによって異なります。

ストレージ割り当て: 基本的な規則は、列またはルーチン・パラメーターに割り当てられるストリングの長さが、列またはルーチン・パラメーターの長さ属性を超えてはならないということです。ストリングの長さが列またはルーチン・パラメーターの長さ属性を超えた場合は、以下の処置が取られます。

- ストリングは、宛先の列またはルーチン・パラメーターの長さ属性に適合するように、(長ストリングを除くすべてのストリング・タイプから) 後続ブランクが切り捨てられた上で割り当てられます。
- 以下の場合にはエラー (SQLSTATE 22001) になります。
 - LONG ストリング以外のストリングからブランク以外の文字が切り捨てられるとき
 - LONG ストリングから任意の文字 (またはバイト) が切り捨てられるとき

ストリングが固定長列に割り当てられる際に、ストリングの長さがターゲットの長さ属性よりも短い場合、ストリングの右端に必要な数の 1 バイト、2 バイト、または UCS-2 のブランクが埋め込まれます。埋め込み文字は、FOR BIT DATA 属性で定義されている列の場合も含めて、常にブランクです。(UCS-2 は、いくつかの SPACE 文字を異なるプロパティで定義します。ユニコード・データベースの場合、データベース・マネージャーは、常に、UCS-2 ブランクとして位置 x'0020' にある ASCII

ストレージ割り当て

SPACE を使用します。EUC データベースの場合、位置 x'3000' にある IDEOGRAPHIC SPACE は、埋め込み GRAPHIC ストリングに使用されます。)

検索割り当て: ホスト変数に割り当てられるストリングの長さは、ホスト変数の長さ属性より長くてもかまいません。ストリングがホスト変数に割り当てられるときに、ストリングの長さが変数の長さ属性より長ければ、ストリングの右側から文字 (またはバイト) が必要な数だけ切り捨てられます。この場合には警告 (SQLSTATE 01004) が戻され、SQLCA の SQLWARN1 フィールドに値 'W' が割り当てられます。

さらに、標識変数があってその値のソースが LOB でない場合は、その標識変数はストリングの元の長さに設定されます。

文字ストリングが固定長変数に割り当てられる際に、ストリングの長さがターゲットの長さ属性よりも短い場合、ストリングの右端に必要な数の 1 バイト、2 バイト、または UCS-2 の空白が埋め込まれます。埋め込み文字は、FOR BIT DATA 属性で定義されているストリングの場合も含めて、常に空白です。(UCS-2 は、いくつかの SPACE 文字を異なるプロパティで定義します。ユニコード・データベースの場合、データベース・マネージャーは、常に、UCS-2 空白として位置 x'0020' にある ASCII SPACE を使用します。EUC データベースの場合、位置 x'3000' にある IDEOGRAPHIC SPACE は、埋め込み GRAPHIC ストリングに使用されます。)

C の NUL 終止符ホスト変数の検索割り当ては、PREP または BIND コマンドに指定されたオプションに基づいて処理されます。

ストリング割り当てに関する変換規則: 列またはホスト変数に割り当てられる文字ストリングまたは GRAPHIC ストリングは、必要であれば、まず割り当て先のコード・ページに変換されます。文字変換が必要になるのは、以下の条件がすべて真の場合だけです。

- コード・ページが異なる。
- ストリングが NULL でも空でもない。
- どちらのストリングのコード・ページ値も 0 (FOR BIT DATA) でない。

ユニコード・データベースの場合、グラフィック列に文字ストリングを割り当てることができ、文字カラムに GRAPHIC ストリングを割り当てることができます。

文字ストリング割り当てに関する MBCS の考慮事項: 1 バイト文字とマルチバイト文字の両方を入れることのできる文字ストリングを割り当てる場合には、いくつかの考慮事項があります。このような考慮事項は、FOR BIT DATA として定義されているものを含めて、すべての文字ストリングに適用されます。

- 空白の埋め込みは、常に単一バイトの空白文字 (X'20') を使用して行われます。

文字ストリング割り当てに関する MBCS の考慮事項

- ブランクの切り捨ては、常に単一バイトのブランク文字 (X'20') に基づいて行われます。切り捨てに関しては、2 バイトのブランク文字はその他の文字と同様に扱われません。
- 文字ストリングをホスト変数に割り当てる場合に、割り当て先のホスト変数にソース・ストリング全体を収めるだけの長さがなければ、MBCS 文字のフラグメント化が発生します。MBCS 文字がこのようにフラグメント化される場合は、MBCS 文字フラグメントの各バイトがターゲットで単一バイトのブランク文字 (X'20') に設定されます。それ以外のバイトに関してはソースからの移動は行われず、SQLWARN1 が 'W' に設定されて切り捨ての発生を示します。MBCS 文字のフラグメント化に関するこの処理は、文字ストリングが FOR BIT DATA として定義されている場合にも同じであることに注意してください。

GRAPHIC ストリング割り当てに関する DBCS の考慮事項: GRAPHIC ストリング割り当ては、文字ストリングに似た方法で処理されます。非ユニコード・データベースの場合、GRAPHIC ストリング・データ・タイプが互換であるのは他の GRAPHIC ストリング・データ・タイプとだけであり、数値、文字ストリング、日付/時刻データ・タイプとは互換ではありません。ユニコード・データベースの場合、GRAPHIC ストリングのデータ・タイプは、文字ストリングのデータ・タイプと互換性があります。

GRAPHIC ストリング値が GRAPHIC ストリング列に割り当てられる場合、その値の長さがその列の長さを超えてはなりません。

GRAPHIC ストリング値 (「ソース」・ストリング) を固定長 GRAPHIC ストリング・データ・タイプ

(「割り当て先」、列またはホスト変数) に割り当てる場合に、ソース・ストリングの長さが割り当て先より短いなら、割り当て先には、ソース・ストリングのコピーの右端に、値の長さが割り当て先の長さに等しくなるために必要な数の 2 バイト・ブランク文字を埋め込んだものが入れます。

GRAPHIC ストリング値を GRAPHIC ストリングのホスト変数に割り当てる場合に、ソース・ストリングの長さがホスト変数の長さよりも長いなら、ホスト変数には、ソース・ストリングのコピーの右端から、値の長さがホスト変数の長さと同じになるために必要な数の 2 バイト・ブランク文字を切り捨てたものが入れます。(このシナリオでは、切り捨てにおいて 2 バイト文字の二分化を考慮する必要はありません。二分化が発生したなら、それはソース値または割り当て先のホスト変数のどちらかで GRAPHIC ストリング・データ・タイプの定義に異常があるということです。) SQLCA の警告フラグ SQLWARN1 が 'W' に設定されます。標識変数が指定されていれば、標識変数にはソース・ストリングの元の長さ (2 バイト文字の文字数) が入れます。しかし、DBCLOB の場合は、標識変数に元の長さは入られません。

C の NUL 終止符ホスト変数 (wchar_t を使って宣言されたもの) の検索割り当ては、PREP または BIND コマンドに指定されたオプションに基づいて処理されます。

日時割り当て

日付/時刻の割り当てに関する基本的な規則は、DATE、TIME、または TIMESTAMP 値は、データ・タイプの一致する列 (DATE、TIME、TIMESTAMP のいずれか)、または、固定長または可変長のストリング列変数かストリング列にしか割り当てできないということです。LONG VARCHAR、CLOB、LONG VARGRAPHIC、DBCLOB または BLOB の変数または列に割り当てすることはできません。

日付/時刻値をストリング変数またはストリング列に割り当てるときは、ストリング表記に自動的に変換されます。日付、時刻、タイム・スタンプのどの部分からも先行ゼロが省略されることはありません。割り当て先で必要な長さは、ストリング表記の形式によって異なります。割り当て先の長さが必要よりも長く、割り当て先が固定長ストリングである場合は、割り当て先の右端にブランクが埋め込まれます。割り当て先の長さが必要よりも短い場合は、関係する日付/時刻値のタイプと割り当て先のタイプによって結果が異なります。

宛先がホスト変数である場合、以下の規則が適用されます。

- **DATE の場合:** 変数の長さが 10 文字未満であればエラーが起こります。
- **TIME の場合:** USA 形式では 8 文字未満の長さの変数を使用できません。その他の形式では 5 文字未満にすることはできません。
ISO または JIS 形式を使用しホスト変数の長さが 8 文字未満の場合、時刻の 2 番目の部分は結果から省略され、標識変数があれば、その変数に割り当てられます。SQLCA の SQLWARN1 フィールドに、省略処理を示す値が設定されます。
- **TIMESTAMP の場合:** ホスト変数が 19 文字未満であればエラーになります。長さが 19 文字以上 26 文字未満の場合、値のマイクロ秒部分の末尾桁が省略されます。SQLCA の SQLWARN1 フィールドに、省略処理を示す値が設定されます。

DATALINK 割り当て

値を DATALINK 列に割り当てると、値のリンケージ属性が指定されていないか、あるいはその列に NO LINK CONTROL が定義されていない限り、ファイルへのリンクが確立されます。リンクされている値がその列にすでにある場合、そのファイルへのリンクは解除されます。リンクされている値がすでにある場合に、NULL 値を割り当てることによっても、その元の値に関連付けられているファイルへのリンクを解除することができます。

すでに列にあるデータの位置と同じ位置をアプリケーションが指定している場合、そのリンクが保存されます。このようになる理由には、以下のいくつかがあります。

- コメントが変更されるため。
- 表がデータ・リンク調整不能 (DRNP) 状態の場合、その表でのリンクは列と同じリンケージ属性を指定することにより、元に戻すことができます。
- 列が WRITE PERMISSION ADMIN で定義され、ファイル内容が変更される場合、同じデータ位置を持つ DLURLNEWCOPY 関数を使用して構成された DATALINK 値を提供することにより、リンクの新規バージョンを確立できます。

- 列が `WRITE PERMISSION ADMIN` で定義され、ファイル内容が変更されますが、変更をバックアウトする必要がある場合、同じデータ位置を持つ `DLURLPREVIOUSCOPY` 関数を使用して作成された `DATALINK` 値を提供することにより、リンクの既存バージョンを元に戻すことができます。
- 参照ファイルの内容は、`DLURLREPLACECONTENT` スカラー関数で指定された別のファイルで置き換えられます。

`DATALINK` 値は、以下のいずれかの方法で列に割り当てることができます。

- `DLVALUE` スカラー関数を使用して、新しい `DATALINK` 値を作成し、その値を列に割り当てることができます。値に含まれているのがコメントだけか、あるいは `URL` がまったく同じでない場合には、この割り当てを行うことによりファイルがリンクされます。
- `CLI` 関数 `SQLBuildDataLink` を使用して、`CLI` パラメーターで `DATALINK` 値を構成することができます。その後、この値を列に割り当てることができます。値に含まれているのがコメントだけか、あるいは `URL` がまったく同じでない場合には、この割り当てを行うことによりファイルがリンクされます。
- `DLURLNEWCOPY` スカラー関数を使用して、`DATALINK` 値を作成し、その値を列に割り当てることができます。作成された `DATALINK` 値によって参照されるデータ位置は、列内にすでに存在するデータ位置と同じでなければなりません。 `UPDATE` ステートメントを使用して割り当てを行うと、ファイルへのリンクが再確立されます。列が `RECOVERY YES` に定義される場合、ファイルのバックアップがとられます。このタイプの割り当ては、ファイルが更新されたことをデータベースに通知するために使用されます。したがって、データベースは、新規ファイルに気付き、ファイルへの新規リンクを再確立します。
- `DLURLPREVIOUSCOPY` スカラー関数を使用して、`DATALINK` 値を作成し、その値を列に割り当てることができます。作成された `DATALINK` 値によって参照されるデータ位置は、列内にすでに存在するデータ位置と同じでなければなりません。 `UPDATE` ステートメントを使用して割り当てを行うと、リンクが元に戻されます。また、列が `RECOVERY YES` に定義される場合、アーカイブから以前のバージョンにファイルをリストアします。このタイプの割り当ては、以前にコミットされたバージョン以降に行われたファイルへの変更をバックアウトするために使用されます。
- `DLURLREPLACECONTENT` スカラー関数を使用して、新しい `DATALINK` 値を作成し、その値を列に割り当てることができます。割り当てを行うと、ファイルのみにリンクされるだけでなく、 `DLURLREPLACECONTENT` スカラー関数に指定された別のファイルの内容に置き換えられます。

値を `DATALINK` 列に割り当てると、以下のエラー条件では `SQLSTATE 428D1` が戻されます。

- データ位置 (`URL`) 形式が無効です (理由コード 21)
- ファイル・サーバーがこのデータベースに登録されていません (理由コード 22)
- 無効なリンク・タイプが指定されています (理由コード 23)

DATALINK 割り当て

- コメントまたは URL の長さが無効です (理由コード 27)

URL パラメーターまたは関数結果のサイズは入力と出力のいずれでも同じで、DATALINK 列の長さに束縛されていることに注意してください。ただし、URL 値にアクセス・トークンが付加されて返される場合があります。このことが起こりうる状態では、アクセス・トークンと、DATALINK 列の長さのための十分な記憶スペースが出力位置になければなりません。したがって、入力で提供される (完全に展開された書式の) 注釈と URL の実際の長さには、出力ストレージ・スペースに合わせるための制限が加えられます。制限された長さを超えると、このエラーが生じます。

- 入力データ位置には、有効な書き込みトークンが含まれていません (理由コード 32)。

割り当てでは、有効な書き込みトークンがデータ位置に組み込まれている必要があります。これは、列が WRITE PERMISSION ADMIN REQUIRING TOKEN FOR UPDATE に定義されているときにのみ適用され、DLURLNEWCOPY または DLURLPREVIOUSCOPY スカラー関数によって DATALINK 値が作成されます。一方、ユーザーは、DATALINK 列が WRITE PERMISSION ADMIN NOT REQUIRING TOKEN FOR UPDATE に定義されている書き込みトークンを提供するためのオプションを持っています。ただし、トークンが有効でない場合、同じエラーが起こります。

このエラーは、値 '1' が 2 番目の引き数に指定されている DLURLNEWCOPY または DLURLPREVIOUSCOPY スカラー関数を使用して DATALINK 値を作成しますが、値に有効な書き込みトークンが含まれていないときにも起こることがあります。

- DLURLPREVIOUSCOPY スカラー関数によって作成された DATALINK 値は、WRITE PERMISSION ADMIN および RECOVERY YES に定義された DATALINK 列にのみ割り当てることができます (理由コード 33)。
- DLURLNEWCOPY または DLURLPREVIOUSCOPY スカラー関数によって作成された DATALINK 値が、列内にすでに存在する値と一致しません (理由コード 34)。
- DLURLNEWCOPY または DLURLPREVIOUSCOPY スカラー関数によって作成された DATALINK 値は、新規値を割り当てるために INSERT ステートメント内で使用できません (理由コード 35)。
- DLURLNEWCOPY スカラー関数によって作成された DATALINK 値は、WRITE PERMISSION BLOCKED で定義された DATALINK 列に割り当てることはできません (理由コード 39)。
- DLURLNEWCOPY または DLURLPREVIOUSCOPY スカラー関数によって作成された同じ DATALINK 値は、同じトランザクション内で何度も割り当てることはできません (理由コード 41)。
- DLURLREPLACECONTENT スカラー関数によって作成された DATALINK 値は、2 番目の引き数が NULL スtring または NULL 値である場合にのみ、NO LINK CONTROL で定義された DATALINK 列に割り当てることができます (理由コード 42)。

- DLREPLACECONTENT スカラー関数に指定された置換ファイルのリンク解除操作がコミットされていません (理由コード 43)。
- DLREPLACECONTENT スカラー関数に指定された置換ファイルは、別の置換プロセスで使用されています (理由コード 44)。
- DATALINK 参照ファイルは、別の操作で置換ファイルとして使用されています (理由コード 45)。
- DLREPLACECONTENT スカラー関数に指定された置換ファイルの形式が有効ではありません (理由コード 46)。
- DLREPLACECONTENT スカラー関数に指定された置換ファイル値が、ディレクトリ一またはシンボリック・リンクになることができません (理由コード 47)。
- DLREPLACECONTENT スカラー関数に指定された置換ファイルは、データベースにリンクされています (理由コード 48)。
- DLREPLACECONTENT スカラー関数に指定された置換ファイルを、データ・リンク・ファイル・マネージャーが見つけることができません (理由コード 49)。
- 書き込みトークンがデータ位置値に含まれている DLURLNEWCOPY スカラー関数によって作成された DATALINK 値は、WRITE PERMISSION ADMIN を含む DATALINK 列にのみ割り当てることができます (理由コード 50)。

割り当てによりリンクの作成も行う場合、以下のエラーが発生する場合があります。

- ファイル・サーバーは現在使用できません (SQLSTATE 57050)。
- ファイルがありません (SQLSTATE 428D1、理由コード 24)。
- ファイルはすでにほかの列にリンクされています (SQLSTATE 428D1、理由コード 25)。
ほかのデータベースへのリンクでも、このエラーは発生することに注意してください。
- 参照されているファイルにアクセスできず、リンクすることができません (理由コード 26)。
- データ位置に組み込まれている書き込みトークンが、ファイルをオープンするために使用された書き込みトークンと一致していません (SQLSTATE 428D1、理由コード 36)。
- DATALINK 値の参照ファイルが更新進行中の状態にあります (SQLSTATE 428D1、理由コード 37)。
- DATALINK 値の参照ファイルの以前のアーカイブ・コピーが使用不能です (SQLSTATE 428D1、理由コード 40)。

さらに、割り当てにより既存のリンクを除去する場合、以下のエラーが発生する場合があります。

- ファイル・サーバーは現在使用できません (SQLSTATE 57050)。

DATALINK 割り当て

- 参照保全制御が指定されているファイルが、データ・リンク・ファイル・マネージャーに従った正しい状態ではありません (SQLSTATE 58004)。
- DATALINK 値の参照ファイルが更新進行中の状態にあります (SQLSTATE 428D1、理由コード 37)。

(DLURLCOMPLETEWRITE または DLURLPATHWRITE スカラー関数を使用して) 書き込みアクセス用の DATALINK 値を検索しているときに、DATALINK 列が WRITE PERMISSION ADMIN で定義される場合、ディレクトリー・アクセス特権がファイル・サーバー (データ・リンク・ファイル・マネージャー) でチェックされます。照会を発行するユーザーは、書き込みトークンが生成され、戻り DATALINK 値に組み込まれる前に、指定されたディレクトリーの下のファイルへの書き込み権限を持っていない限りなりません。ユーザーが書き込み権限を持っていない場合、書き込みトークンは生成されず、SELECT ステートメントは失敗します (SQLSTATE 42511、理由コード 1)。

DATALINK 値の一部分を、スカラー関数 (DLINKTYPE または DLURLPATH など) のアプリケーションの後のホスト変数に割り当てることができます。

通常、検索時にファイル・サーバーへのアクセスは試みられないことに注意してください。そのため、それ以降にファイル・システム・コマンドを使ってファイル・サーバーにアクセスを試みても、失敗する可能性があります。パスに関連した接頭部名を決定するのに、ファイル・サーバーにアクセスすることが必要になる場合があります。これはファイル・システムのマウント・ポイントを移動する場合に、ファイル・サーバーで変更することができます。サーバー上のファイルに始めてアクセスしたときに、必要な値がファイル・サーバーから検索されてデータベース・サーバーのキャッシュに置かれます。そして、それ以降の DATALINK 値の検索はそのファイル・サーバーで行われます。ファイル・サーバーにアクセスできない場合には、エラーが戻されます (SQLSTATE 57050)。

DATALINK 値を検索するためにスカラー関数 DLURLCOMPLETEWRITE または DLURLPATHWRITE を使用する場合、ユーザーのパス上にディレクトリー・アクセス特権を決定するために、ファイル・サーバーにアクセスすることが必要になる場合があります。ファイル・サーバーにアクセスできない場合には、エラーが戻されます (SQLSTATE 57050)。

DATALINK の値を検索する場合、データベース・サーバーでのファイル・サーバーの登録がチェックされ、そのファイル・サーバーがまだそのデータベース・サーバーに登録されていることが確認されます (SQLSTATE 55022)。さらに、DATALINK 値を検索する場合に警告が戻される場合がありますが、それはその表が調整保留状態または調整不能状態のためです (SQLSTATE 01627)。

ユーザー定義タイプの割り当て

ユーザー定義タイプをホスト変数へ割り当てる場合には、他の割り当てで使用される規則とは異なる規則が適用されます。

特殊タイプ: ホスト変数への割り当ては、特殊タイプのソース・タイプに基づいて行われます。つまり、以下の規則に従います。

- 割り当ての右辺に指定する特殊タイプの値は、その特殊タイプのソース・タイプが割り当ての左辺に指定するホスト変数に割り当て可能な場合にのみ、ホスト変数へ割り当てられます。

割り当ての宛先が、特殊タイプに基づく列である場合、ソース・データ・タイプは、ターゲット・データ・タイプへキャスト可能でなければなりません。

構造タイプ: ホスト変数に対する割り当ては、ホスト変数の宣言済みタイプに基づきます。つまり、以下の規則に従います。

割り当ての右辺に指定する構造タイプの値は、宣言済みタイプのホスト変数が構造タイプ、または構造タイプのスーパータイプである場合にのみ、左辺のホスト変数へ割り当てられます。

割り当ての宛先が、構造タイプの列である場合、ソース・データ・タイプは、ターゲット・データ・タイプ、またはターゲット・データ・タイプのサブタイプでなければなりません。

参照タイプの割り当て

ターゲット・タイプ T を指定している参照タイプは、ターゲット・タイプ S を指定している参照タイプでもある参照タイプ列に割り当てることができます (S は T のスーパータイプ)。有効範囲が指定されている参照列または変数に割り当てが行われる場合、割り当てられている実際の値が、有効範囲で定義されているターゲット表またはターゲット・ビューに存在することを確認するためのチェックは行われません。

ホスト変数への割り当ては、参照タイプの表示タイプに基づいて行われます。つまり、以下の規則に従います。

- 割り当ての右側に指定する参照タイプの値は、ホスト参照変数の左側にも割り当て可能ですが、それはこの参照タイプの表示タイプがこのホスト変数に割り当て可能な場合だけです。

割り当てのターゲットが列で、その割り当ての右側にホスト変数が指定されている場合、そのホスト変数はそのターゲット列の参照タイプに明示的にキャストされなければなりません。

数値比較

数値は代数的に、つまり符号を考慮して比較されます。たとえば、 -2 は $+1$ より小さい値として扱われます。

一方が整数で、もう一方が 10 進数の場合、10 進数に変換された整数の一時コピーが比較に使用されます。

数値比較

位取りの異なる 10 進数を比較する場合、比較は、一方の数値の小数部分が、他方の数値の小数部分と同じ桁数になるように、後続ゼロを使って拡張されたその数値の一時コピーを使用して行われます。

一方が浮動小数点数で、他方が整数か 10 進数の場合、この後者の数値を倍精度浮動小数点数に変換したものの一時コピーが比較に使用されます。

一方が浮動小数点でもう一方が整数または 10 進数の場合、浮動小数点に変換された一時コピーが比較に使用されます。

ストリングの比較

文字ストリングは、データベースの作成時に指定された照合順序にしたがって比較されます。ただし、FOR BIT DATA 属性の文字ストリングは例外で、そのような文字ストリングは常にビット値にしたがって比較されます。

長さの異なる文字ストリングを比較する場合、短い方のストリングの右端に、長い方のストリングの長さになるまで 1 バイト・ブランクを埋め込んだものの論理コピーが比較に使用されます。この論理的な拡張は、FOR BIT DATA のものも含め、すべての文字ストリングに対して行われます。

文字ストリング (FOR BIT DATA のタグが付けられた文字ストリングを除く) は、データベースの作成時に指定された照合順序にしたがって比較されます。たとえば、データベース・マネージャーによって指定されるデフォルトの照合順序は、同じ文字の小文字と大文字に同じ重みを与えています。データベース・マネージャーは、完全に同一のストリングだけが相互に等しいものとして扱われるようにするために、2 つの比較を実行します。まず、ストリングがデータベースの照合順序にしたがって比較されます。ストリングの文字の重みが等しい場合、次の判断基準として、実際のコード・ポイント値に基づいてストリングを比較します。

2 つのストリングは、両方が空であるか、または対応するすべてのバイト数が等しい場合には、等しくなります。どちらかのオペランドが NULL 値の場合の結果は不明です。

基本比較演算子 (=、<>、<、>、<=、および >=) を使用する比較演算では、LONG ストリングおよび LOB ストリングはサポートされません。このようなストリングの比較は、LIKE 述部と POSSTR 関数を使用した比較でサポートされています。

LONG ストリングおよび LOB ストリングのうち 4 000 バイト以下の部分は、SUBSTR と VARCHAR のスカラー関数を使用して比較できます。たとえば、以下のような列を考えてみます。

```
MY_SHORT_CLOB  CLOB(300)
MY_LONG_VAR    LONG VARCHAR
```

この場合、以下の演算は有効です。

```
WHERE VARCHAR(MY_SHORT_CLOB) > VARCHAR(SUBSTR(MY_LONG_VAR,1,300))
```

例:

以下の例で、'A'、'Á'、'a'、および 'á' のコード・ポイント値はそれぞれ、X'41'、X'C1'、X'61'、および X'E1' です。

'A'、'Á'、'a'、'á' という文字の重みが 136、139、135、138 である照合順序を考えてみます。このような場合は以下ようになります。

'a' < 'A' < 'á' < 'Á'

今度は D1、D2、D3、および D4 という 4 つの DBCS 文字を例にとって考えてみます。これらの文字はそれぞれ 0xC141、0xC161、0xE141、および 0xE161 というコード・ポイントを持っています。これらの DBCS 文字が CHAR 列に含まれている場合、各文字のバイトが持っている照合重みに従った順序でソートされます。最初の 2 つのバイトの重みは 138 と 139 であるため、D3 と D4 は D2 と D1 よりも前に来ます。続く 2 つのバイトの重みは 135 と 136 であるため、順序は以下ようになります。

D4 < D3 < D2 < D1

ただし、比較する値に FOR BIT DATA 属性がある場合や、これらの DBCS 文字が GRAPHIC 列に格納された場合は、照合順序は無視され、これらの文字が持っているコード・ポイントにしたがって文字が比較されます。以下ようになります。

'A' < 'a' < 'Á' < 'á'

DBCS 文字はコード・ポイントの順序でソートされます。以下ようになります。

D1 < D2 < D3 < D4

次に 'A'、'Á'、'a'、'á' という文字が、74、75、74、および 75 の (固有でない) 重みを持つ照合順序を考えてみます。照合重みだけに注目すると (第 1 パス)、'a' は 'A' に等しく、'á' は 'Á' に等しいです。決着を付けるために文字のコード・ポイントを使用すると (第 2 のパス)、以下ようになります。

'A' < 'a' < 'Á' < 'á'

CHAR 列に含まれている DBCS 文字は、最初は重みに従ったバイトの順序 (第 1 パス) でソートされます。それでも決着がつかない場合は、コード・ポイントに従ったバイトの順序 (第 2 パス) でソートされます。最初の 2 つのバイトは重みが同じであるため、コード・ポイント (0xC1 と 0xE1) で決着を付けることとなります。結果として、文字 D1 と D2 は文字 D3 と D4 の前にソートされます。続く 2 つのバイトもこれと同じように比較されます。最終的な結果は以下ようになります。

D1 < D2 < D3 < D4

ストリングの比較

ここでも、比較する値に FOR BIT DATA 属性がある場合や、これらの DBCS 文字が GRAPHIC 列に格納された場合は、照合順序は無視され、これらの文字が持っているコード・ポイントにしたがって文字が比較されます。以下ようになります。

D1 < D2 < D3 < D4

この例では、照合重みが使用されたときと同じ結果が戻されていますが、実際の場面でいつもそのようになるとは限りません。

比較の際の変換規則: 2 つのストリングを比較する場合、必要なら、一方のストリングがまずもう一方のストリングのコード化スキームおよびコード・ページに変換されます。

結果の順序付け: 結果のソートが必要な場合、1376 ページの『ストリングの比較』で説明されているストリング比較規則に基づいて順序付けが行われます。比較はデータベース・サーバー側で実行されます。クライアント・アプリケーションに結果が戻される時点で、コード・ページ変換が実行されることがあります。後から行われるこのようなコード・ページ変換は、サーバーの決定した結果セットの順序には影響しません。

ストリング比較に関する MBCS の考慮事項: SBCS/MBCS 混合文字ストリングは、データベースの作成時に指定された照合順序にしたがって比較されます。デフォルト (SYSTEM) 照合順序で作成されたデータベースの場合、1 バイトの ASCII 文字はすべて正しい順序で保管されますが、2 バイト文字は必ずしもコード・ポイントの順序になっているとは限りません。IDENTITY 順序で作成されたデータベースの場合、2 バイト文字はすべてコード・ポイントの順序で保管され、1 バイトの ASCII 文字も同様にコード・ポイントの順序で保管されます。COMPATIBILITY 順序で作成されたデータベースの場合、ほとんどの 2 バイト文字について正しくソートを行い、ASCII についてもほぼ正しい、中間的な順序が使用されます。これは、DB2 バージョン 2 ではデフォルトの照合表でした。

混合文字ストリングはバイトごとに比較されます。混合ストリングに含まれるマルチバイト文字について通常と異なる結果になる場合がありますが、これは個々のバイトが個別に扱われるためです。

例:

この例で、'A'、'B'、'a'、および 'b' の 2 バイト文字のコード・ポイント値はそれぞれ、X'8260'、X'8261'、X'8281'、および X'8282' です。

コード・ポイント X'8260'、X'8261'、X'8281'、および X'8282' の重みがそれぞれ 96、65、193、および 194 である照合順序を考えてみます。この場合は以下ようになります。

'B' < 'A' < 'a' < 'b'

および

```
'AB' < 'AA' < 'Aa' < 'Ab' < 'aB' < 'aA' < 'aa' < 'ab'
```

GRAPHIC ストリングの比較は、文字ストリングの場合と同じように処理されます。

GRAPHIC ストリングの比較は、LONG VARGRAPHIC を除くすべての GRAPHIC ストリング・データ・タイプの間で有効です。LONG VARGRAPHIC および DBCLOB データ・タイプは、比較演算では使用できません。

GRAPHIC ストリングに対しては、データベースの照合順序は使用されません。その代わりに、GRAPHIC ストリングは、常に対応するバイトの数値 (2 進値) に基づいて比較されます。

前の例で、リテラルが GRAPHIC ストリングの場合、以下のような結果になります。

```
'A' < 'B' < 'a' < 'b'
```

および

```
'AA' < 'AB' < 'Aa' < 'Ab' < 'aA' < 'aB' < 'aa' < 'ab'
```

長さの異なる GRAPHIC ストリングを比較する場合、短い方のストリングの右端に長い方のストリングの長さになるまで、2 バイト・ブランクを埋め込んだものの論理コピーが比較に使用されます。

2 つの GRAPHIC ストリングの値が等しくなるのは、両方が空であるか、または対応するグラフィックがすべて等しい場合です。どちらかのオペランドが NULL 値の場合の結果は不明です。2 つの値が等しくない場合は、両者の関係は単純な 2 進ストリング比較によって決定されます。

この節で説明してきたとおり、バイトに基づくストリングの比較は誤った結果をもたらす場合があります。つまり、文字比較で得られる文字とは異なる結果が生じる場合があります。ここで示した一連の例は、同じ MBCS コード・ページであることを前提にしていますが、実際には、同じ言語を使用していても異なるマルチバイトのコード・ページを使用することがあるので、状況はもっと複雑であるといえます。たとえば、日本語 DBCS コード・ページと日本語 EUC コード・ページからのストリングを比較するというような場合が考えられます。

日時比較

DATE、TIME、または TIMESTAMP 値は、同じデータ・タイプの別の値か、そのデータ・タイプのストリング表記と比較することができます。すべての比較は日時順に行われます。つまり、0001 年 1 月 1 日からの時間の経過の大きい方が値が大きいということです。

TIME 値と、時刻値のストリング表記とが関係する比較では、常に秒数が含まれます。ストリング表記で秒数を省略しているときは、暗黙のうちにゼロ秒が補われます。

日時比較

TIMESTAMP 値に関する比較は、等しいと見なしてもよいような表示の考慮はしません。日時順に行われます。

例:

```
TIMESTAMP('1990-02-23-00.00.00') > '1990-02-22-24.00.00'
```

ユーザー定義タイプの比較

ユーザー定義特殊タイプの値は、完全に同じユーザー定義特殊タイプの値とのみ比較することができます。ユーザー定義特殊タイプは、WITH COMPARISONS 文節を使用して定義されていなければなりません。

例:

以下の YOUTH 特殊タイプおよび CAMP_DB2_ROSTER 表を想定します。

```
CREATE DISTINCT TYPE YOUTH AS INTEGER WITH COMPARISONS

CREATE TABLE CAMP_DB2_ROSTER
  ( NAME          VARCHAR(20),
    ATTENDEE_NUMBER INTEGER NOT NULL,
    AGE           YOUTH,
    HIGH_SCHOOL_LEVEL YOUTH)
```

以下の比較は有効です。

```
SELECT * FROM CAMP_DB2_ROSTER
WHERE AGE > HIGH_SCHOOL_LEVEL
```

以下の比較は無効です。

```
SELECT * FROM CAMP_DB2_ROSTER
WHERE AGE > ATTENDEE_NUMBER
```

ただし、特殊タイプとそのソース・タイプとの間では、キャストのための関数または CAST 指定を使用することによって、AGE と ATTENDEE_NUMBER とを比較することができます。以下の比較はすべて有効です。

```
SELECT * FROM CAMP_DB2_ROSTER
WHERE INTEGER(AGE) > ATTENDEE_NUMBER
```

```
SELECT * FROM CAMP_DB2_ROSTER
WHERE CAST( AGE AS INTEGER) > ATTENDEE_NUMBER
```

```
SELECT * FROM CAMP_DB2_ROSTER
WHERE AGE > YOUTH(ATTENDEE_NUMBER)
```

```
SELECT * FROM CAMP_DB2_ROSTER
WHERE AGE > CAST(ATTENDEE_NUMBER AS YOUTH)
```

ユーザー定義構造タイプの値を、他の値と比較することはできません (NULL 述部および TYPE 述部が使えます)。

参照タイプの比較

参照タイプ値を比較できるのは、それらのターゲット・タイプが共通のスーパータイプを持っている場合だけです。その共通のスーパータイプのスキーマ名が関数パスに組み込まれている場合、適切な比較関数は検索されるだけです。比較は参照タイプの表示タイプを使用して行われます。参照の有効範囲は、比較では考慮されません。

関連資料:

- *SQL* リファレンス 第 1 巻 の『ID』
- *SQL* リファレンス 第 1 巻 の『LIKE 述部』
- *SQL* リファレンス 第 1 巻 の『POSSTR スカラー関数』
- *SQL* リファレンス 第 1 巻 の『日付/時刻の値』
- *SQL* リファレンス 第 1 巻 の『データ・タイプ間のキャスト』
- *SQL* リファレンス 第 1 巻 の『結果データ・タイプの規則』
- *SQL* リファレンス 第 1 巻 の『string変換の規則』

デフォルトの順方向データ・タイプ・マッピング

データ・ソース・オブジェクトにニックネームを作成すると、DB2 (UNIX および Windows 版) は、表についての情報をグローバル・カタログに入れます。

この情報は、それぞれの列のリモート のデータ・タイプおよび、対応する DB2 (UNIX および Windows 版) のデータ・タイプを含んでいます。DB2 (UNIX および Windows 版) データ・タイプは、ローカル・データ・タイプと呼ばれます。

連合データベースは、データ・タイプ・マッピングを使用して、データ・ソース・オブジェクトの列にどのような DB2 (UNIX および Windows 版) データ・タイプを定義すべきかを判別します。

連合サーバーがデータ・ソースからデータを検索するには、データ・ソース側のデータ・タイプが、対応する DB2 (UNIX および Windows 版) のデータ・タイプと対応付けられて (マッピングされて) いなければなりません。ほとんどのデータ・ソースの場合、ラッパー内にデフォルトのタイプ・マッピングがあります。DB2 ファミリーのデータ・ソース用のデフォルトのタイプ・マッピングは、DRDA ラッパーにあります。Informix 用のデフォルトのタイプ・マッピングは INFORMIX ラッパーなどにあります。

DB2 (UNIX および Windows 版) 連合サーバーは、以下のローカル・データ・タイプのマッピングをサポートしません。

- DATALINK
- ユーザー定義タイプ

参照タイプの比較

データ・ソースのデータ・タイプと連合データベースのデータ・タイプ間のマッピングには、順方向タイプのマッピングと逆方向タイプのマッピングの 2 種類があります。順方向タイプ・マッピングでは、マッピングはリモートのタイプから対応するローカル・タイプへのマッピングです。

デフォルトのタイプ・マッピングをオーバーライドすることも、CREATE TYPE MAPPING ステートメントを使用して新しいタイプ・マッピングを作成することもできます。

以下の表は、DB2 (UNIX および Windows 版) のデータ・タイプとデータ・ソースのデータ・タイプ間のデフォルトの順方向マッピングを示しています。

これらのマッピングは、特に記述のないかぎり、サポート対象のすべてのバージョンで有効です。

注: データ・ソースから DB2 (UNIX および Windows 版) へのデフォルトの順方向データ・タイプ・マッピングのすべてについて、DB2 連合スキーマは SYSIBM です。

DB2 (z/OS および OS/390 版) データ・ソース

表 255. DB2 (z/OS および OS/390 版) のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	1	254	-	-	-	-	CHAR	-	0	N
CHAR	255	32672	-	-	-	-	VARCHAR	-	0	N
CHAR	1	254	-	-	Y	-	CHAR	-	0	Y
CHAR	255	32672	-	-	Y	-	VARCHAR	-	0	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-

表 255. DB2 (z/OS および OS/390 版) のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります) (続き)

REMOTE_TYPENAME		REMOTE_LOWER_LEN	REMOTE_UPPER_LEN REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH FEDERATED_SCALE	FEDERATED_BIT_DATA
FLOAT	4	-	-	-	-	-	REAL	-	-
FLOAT	8	-	-	-	-	-	DOUBLE	-	-
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	0 N
INTEGER	-	-	-	-	-	-	INTEGER	-	0 -
ROWID	-	-	-	-	Y	-	VARCHAR	40	- Y
SMALLINT	-	-	-	-	-	-	SMALLINT	-	0 -
TIME	-	-	-	-	-	-	TIME	-	0 -
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP	-	0 -
TIMESTMP	-	-	-	-	-	-	TIMESTAMP	-	0 -
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	0 N
VARCHAR	1	32672	-	-	Y	-	VARCHAR	-	0 Y
VARG	1	16336	-	-	-	-	VARGGRAPHIC	-	0 N
VARGGRAPHIC	1	16336	-	-	-	-	VARGGRAPHIC	-	0 N

参照タイプの比較

DB2 (iSeries 版) データ・ソース

表 256. DB2 (iSeries 版) のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	1	254	-	-	-	-	CHAR	-	0	N
CHAR	255	32672	-	-	-	-	VARCHAR	-	0	N
CHAR	1	254	-	-	Y	-	CHAR	-	0	Y
CHAR	255	32672	-	-	Y	-	VARCHAR	-	0	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
I FLOAT	4	-	-	-	-	-	REAL	-	-	-
I FLOAT	8	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	0	N
GRAPHIC	128	16336	-	-	-	-	VARGRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	0	-
NUMERIC	-	-	-	-	-	-	DECIMAL	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	0	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP	-	0	-
TIMESTMP	-	-	-	-	-	-	TIMESTAMP	-	0	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	0	N
VARCHAR	1	32672	-	-	Y	-	VARCHAR	-	0	Y
VARG	1	16336	-	-	-	-	VARGRAPHIC	-	0	N

表 256. DB2 (iSeries 版) のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります) (続き)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
VARGRAPHIC	1	16336	-	-	-	-	VARGRAPHIC	-	0	N

DB2 サーバー (VM および VSE 版) データ・ソース

表 257. DB2 サーバー (VM および VSE 版) のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	1	254	-	-	-	-	CHAR	-	0	N
CHAR	1	254	-	-	Y	-	CHAR	-	0	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DBAHW	-	-	-	-	-	-	SMALLINT	-	0	-
DBAINT	-	-	-	-	-	-	INTEGER	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
FLOAT	-	-	-	-	-	-	DOUBLE	-	-	-

参照タイプの比較

表 257. DB2 サーバー (VM および VSE 版) のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります) (続き)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	-	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP	-	0	-
TIMESTMP	-	-	-	-	-	-	TIMESTAMP	-	0	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	0	N
VARCHAR	1	32672	-	-	Y	-	VARCHAR	-	0	Y
VARGRAPHIC	1	16336	-	-	-	-	VARGRAPHIC	-	0	N
VARGRAPH	1	16336	-	-	-	-	VARGRAPHIC	-	0	N

DB2 (UNIX および Windows 版) データ・ソース

表 258. DB2 (UNIX および Windows 版) のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
BIGINT	-	-	-	-	-	-	BIGINT	-	0	-

表 258. DB2 (UNIX および Windows 版) のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります) (続き)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	-	-	-	-	-	-	CHAR	-	0	N
CHAR	-	-	-	-	Y	-	CHAR	-	0	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	-	-	-	-	-	DOUBLE	-	-	-
FLOAT	-	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	0	-
LONGVAR	-	-	-	-	N	-	CLOB	-	-	-
LONGVAR	-	-	-	-	Y	-	BLOB	-	-	-
LONGVARG	-	-	-	-	-	-	DBCLOB	-	-	-
REAL	-	-	-	-	-	-	REAL	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	0	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP	-	0	-
TIMESTMP	-	-	-	-	-	-	TIMESTAMP	-	0	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	0	N
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	0	Y
VARGRAPH	-	-	-	-	-	-	VARGRAPHIC	-	0	N
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	0	N

参照タイプの比較

Informix データ・ソース

表 259. Informix のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
BLOB	-	-	-	-	-	-	BLOB	2147483647	-	-
BOOLEAN	-	-	-	-	-	-	SMALLINT	2	-	-
BYTE	-	-	-	-	-	-	BLOB	2147483647	-	-
CHAR	1	254	-	-	-	-	CHARACTER	-	-	-
CHAR	255	32672	-	-	-	-	VARCHAR	-	-	-
CLOB	-	-	-	-	-	-	CLOB	2147483647	-	-
DATE	-	-	-	-	-	-	DATE	4	-	-
DATETIME	0	4	0	4	-	-	DATE	4	-	-
DATETIME	6	10	6	10	-	-	TIME	3	-	-
DATETIME	0	4	6	15	-	-	TIMESTAMP	10	-	-
DATETIME	6	10	11	15	-	-	TIMESTAMP	10	-	-
DECIMAL	1	31	0	31	-	-	DECIMAL	-	-	-
DECIMAL	32	32	-	-	-	-	DOUBLE	8	-	-
FLOAT	-	-	-	-	-	-	DOUBLE	8	-	-
INTEGER	-	-	-	-	-	-	INTEGER	4	-	-
INTERVAL	-	-	-	-	-	-	DECIMAL	19	5	-
INT8	-	-	-	-	-	-	BIGINT	19	0	-
LVARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	-
MONEY	1	31	0	31	-	-	DECIMAL	-	-	-
MONEY	32	32	-	-	-	-	DOUBLE	8	-	-
NCHAR	1	254	-	-	-	-	CHARACTER	-	-	-
NCHAR	255	32672	-	-	-	-	VARCHAR	-	-	-
NVARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	-

表 259. Informix のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります) (続き)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
REAL	-	-	-	-	-	-	REAL	4	-	-
SERIAL	-	-	-	-	-	-	INTEGER	4	-	-
SERIAL8	-	-	-	-	-	-	BIGINT	-	-	-
SMALLFLOAT	-	-	-	-	-	-	REAL	4	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	2	-	-
TEXT	-	-	-	-	-	-	CLOB	2147483647	-	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	-

注:

Informix DATETIME データ・タイプでは、DB2 UNIX および Windows 連合サーバーは、Informix 高水準修飾子を、REMOTE_LENGTH として、Informix 低水準修飾子を REMOTE_SCALE として使用します。

Informix 修飾子は、Informix クライアント SDK datatype.h ファイルに定義されている "TU_" 定数です。定数は次のとおりです。

0 = YEAR	8 = MINUTE	13 = FRACTION(3)
2 = MONTH	10 = SECOND	14 = FRACTION(4)
4 = DAY	11 = FRACTION(1)	15 = FRACTION(5)
6 = HOUR	12 = FRACTION(2)	

参照タイプの比較

Oracle SQLNET データ・ソース

表 260. Oracle SQLNET のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
CHAR	1	254	0	0	-	¥0	CHAR	0	0	N
CHAR	255	32672	0	0	-	¥0	VARCHAR	0	0	N
DATE	0	0	0	0	-	¥0	TIMESTAMP	0	0	N
FLOAT	1	63	0	0	-	¥0	REAL	0	0	N
FLOAT	64	126	0	0	-	¥0	DOUBLE	0	0	N
LONG	0	0	0	0	-	¥0	CLOB	2147483647	0	N
LONG RAW	0	0	0	0	-	¥0	BLOB	2147483647	0	Y
MLSLABEL	0	0	0	0	-	¥0	VARCHAR	255	0	N
NUMBER	1	38	-84	127	-	¥0	DOUBLE	0	0	N
NUMBER	1	31	0	31	-	>=	DECIMAL	0	0	N
NUMBER	1	5	0	0	-	¥0	SMALLINT	0	0	N
NUMBER	6	10	0	0	-	¥0	INTEGER	0	0	N
RAW	1	254	0	0	-	¥0	CHAR	0	0	Y
RAW	255	32672	0	0	-	¥0	VARCHAR	0	0	Y
ROWID	0	0	0	NULL	-	¥0	CHAR	18	0	N
VARCHAR2	1	32672	0	0	-	¥0	VARCHAR	0	0	N

Oracle NET8 データ・ソース

表 261. Oracle NET8 のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
BLOB	0	0	0	0	-	¥0	BLOB	2147483647	0	Y
CHAR	1	254	0	0	-	¥0	CHAR	0	0	N
CHAR	255	32672	0	0	-	¥0	VARCHAR	0	0	N
CLOB	0	0	0	0	-	¥0	CLOB	2147483647	0	N
DATE	0	0	0	0	-	¥0	TIMESTAMP	0	0	N
FLOAT	1	63	0	0	-	¥0	REAL	0	0	N
FLOAT	64	126	0	0	-	¥0	DOUBLE	0	0	N
MLSLABEL	0	0	0	0	-	¥0	VARCHAR	255	0	N
NUMBER	1	38	-84	127	-	¥0	DOUBLE	0	0	N
NUMBER	1	31	0	31	-	>=	DECIMAL	0	0	N
NUMBER	1	5	0	0	-	¥0	SMALLINT	0	0	N
NUMBER	6	10	0	0	-	¥0	INTEGER	0	0	N
RAW	1	254	0	0	-	¥0	CHAR	0	0	Y
RAW	255	32672	0	0	-	¥0	VARCHAR	0	0	Y
ROWID	0	0	0	NULL	-	¥0	CHAR	18	0	N
VARCHAR2	1	32672	0	0	-	¥0	VARCHAR	0	0	N

参照タイプの比較

Microsoft SQL Server データ・ソース

表 262. Microsoft SQL Server のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
バイナリー	1	254	-	-	-	-	CHARACTER	-	-	Y
バイナリー	255	8000	-	-	-	-	VARCHAR	-	-	Y
bit	-	-	-	-	-	-	SMALLINT	2	-	-
char	1	254	-	-	-	-	CHAR	-	-	N
char	255	8000	-	-	-	-	VARCHAR	-	-	N
datetime	-	-	-	-	-	-	TIMESTAMP	10	-	-
datetimen	-	-	-	-	-	-	TIMESTAMP	10	-	-
decimal	1	31	0	31	-	-	DECIMAL	-	-	-
decimal	32	38	0	38	-	-	DOUBLE	-	-	-
decimaln	1	31	0	31	-	-	DECIMAL	-	-	-
decimaln	32	38	0	38	-	-	DOUBLE	-	-	-
DUMMY65 ¹	1	38	-84	127	-	-	DOUBLE	-	-	-
DUMMY2000 ³	1	38	-84	127	-	-	DOUBLE	-	-	-
float	-	8	-	-	-	-	DOUBLE	8	-	-
floatn	-	8	-	-	-	-	DOUBLE	8	-	-
float	-	4	-	-	-	-	REAL	4	-	-
floatn	-	4	-	-	-	-	REAL	4	-	-
image	-	-	-	-	-	-	BLOB	2147483647	-	Y
int	-	-	-	-	-	-	INTEGER	4	-	-
intn	-	-	-	-	-	-	INTEGER	4	-	-
money	-	-	-	-	-	-	DECIMAL	19	4	-
moneyn	-	-	-	-	-	-	DECIMAL	19	4	-

表 262. Microsoft SQL Server のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります) (続き)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
nchar	1	127	-	-	-	-	CHAR	-	-	N
nchar	128	4000	-	-	-	-	VARCHAR	-	-	N
numeric	1	31	0	31	-	-	DECIMAL	-	-	-
numeric	32	38	0	38	-	-	DOUBLE	8	-	-
numericn	32	38	0	38	-	-	DOUBLE	-	-	-
numericn	1	31	0	31	-	-	DECIMAL	-	-	-
ntext ²	-	-	-	-	-	-	CLOB	2147483647	-	Y
nvarchar	1	4000	-	-	-	-	VARCHAR	-	-	N
real	-	-	-	-	-	-	REAL	4	-	-
smallint	-	-	-	-	-	-	SMALLINT	2	-	-
smalldatetime	-	-	-	-	-	-	TIMESTAMP	10	-	-
smallmoney	-	-	-	-	-	-	DECIMAL	10	4	-
smallmoneyn	-	-	-	-	-	-	DECIMAL	10	4	-
SQL_BIGINT	-	-	-	-	-	-	DECIMAL	-	-	-
SQL_BINARY	1	254	-	-	-	-	CHARACTER	-	-	Y
SQL_BINARY	255	8000	-	-	-	-	VARCHAR	-	-	Y
SQL_BIT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_CHAR	1	254	-	-	-	-	CHAR	-	-	N
SQL_CHAR	255	8000	-	-	-	-	VARCHAR	-	-	N
SQL_DATE	-	-	-	-	-	-	DATE	4	-	-
SQL_DECIMAL	1	31	0	31	-	-	DECIMAL	-	-	-
SQL_DECIMAL	32	38	0	38	-	-	DOUBLE	8	-	-
SQL_DECIMAL	32	32	0	31	-	-	DOUBLE	8	-	-
SQL_DOUBLE	-	-	-	-	-	-	DOUBLE	8	-	-

参照タイプの比較

表 262. Microsoft SQL Server のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります) (続き)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
SQL_FLOAT	-	-	-	-	-	-	DOUBLE	8	-	-
SQL_GUID ²	1	4000	-	-	Y	-	VARCHAR	16	-	Y
SQL_INTEGER	-	-	-	-	-	-	INTEGER	4	-	-
SQL_LONGVARCHAR	-	-	-	-	-	-	CLOB	2147483647	-	N
SQL_LONGVARBINARY	-	-	-	-	-	-	BLOB	-	-	Y
SQL_NUMERIC	1	31	0	31	-	-	DECIMAL	-	-	-
SQL_REAL	-	-	-	-	-	-	DOUBLE	8	-	-
SQL_SMALLINT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_TIME	-	-	-	-	-	-	TIME	3	-	-
SQL_TIMESTAMP	-	-	-	-	-	-	TIMESTAMP	10	-	-
SQL_TINYINT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_VARBINARY	1	8000	-	-	-	-	VARCHAR	-	-	Y
SQL_VARCHAR	1	8000	-	-	-	-	VARCHAR	-	-	N
sysname	-	-	-	-	-	-	VARCHAR	30	-	Y
sysname	1	254	-	-	-	-	CHAR	-	-	N
text	-	-	-	-	-	-	CLOB	-	-	N
timestamp	-	-	-	-	-	-	VARCHAR	8	-	Y
tinyint	-	-	-	-	-	-	SMALLINT	2	-	-
ユニーク ID ²	1	4000	-	-	Y	-	VARCHAR	16	-	Y
varbinary	1	8000	-	-	-	-	VARCHAR	-	-	Y
varchar	1	8000	-	-	-	-	VARCHAR	-	-	N

表 262. Microsoft SQL Server のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります) (続き)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH FEDERATED_SCALE	FEDERATED_BIT_DATA
-----------------	------------------	----------------------------------------	--------------------	------------------------------------------	--------------------	-------------------------------------	--------------------

注:

1. このタイプ・マッピングは Microsoft SQL Server バージョン 6.5 でのみ有効です。
2. このタイプ・マッピングは Microsoft SQL Server バージョン 7 およびバージョン 2000 でのみ有効です。
3. このタイプ・マッピングは Windows 2000 オペレーティング・システムでのみ有効です。

ODBC データ・ソース

表 263. ODBC のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH FEDERATED_SCALE	FEDERATED_BIT_DATA
SQL_BIGINT	-	-	-	-	DECIMAL	-	-
SQL_BINARY	1	254	-	-	CHARACTER	-	Y
SQL_BINARY	255	32672	-	-	VARCHAR	-	Y
SQL_BIT	-	-	-	-	SMALLINT	2	-
SQL_CHAR	1	254	-	-	CHAR	-	N
SQL_CHAR	255	32672	-	-	VARCHAR	-	N

参照タイプの比較

表 263. ODBC のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります) (続き)

REMOTE_TYPENAME	REMOTE_LOWER_LEN		REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
SQL_DECIMAL	1	31	0	31	-	-	DECIMAL	-	-	-	
SQL_DECIMAL	32	38	0	38	-	-	DOUBLE	8	-	-	
SQL_DOUBLE	-	-	-	-	-	-	DOUBLE	8	-	-	
SQL_FLOAT	-	-	-	-	-	-	DOUBLE	8	-	-	
SQL_INTEGER	-	-	-	-	-	-	INTEGER	4	-	-	
SQL_LONGVARCHAR	-	-	-	-	-	-	CLOB	2147483647	-	N	
SQL_LONGVARBINARY	-	-	-	-	-	-	BLOB	-	-	Y	
SQL_NUMERIC	1	31	0	31	-	-	DECIMAL	-	-	-	
SQL_NUMERIC	32	32	0	31	-	-	DOUBLE	8	-	-	
SQL_REAL	-	-	-	-	-	-	DOUBLE	8	-	-	
SQL_SMALLINT	-	-	-	-	-	-	SMALLINT	2	-	-	
I SQL_TYPE_DATE	-	-	-	-	-	-	DATE	4	-	Y	
I SQL_TYPE_TIME	-	-	-	-	-	-	TIME	3	-	Y	
I SQL_TYPE_TIMESTAMP	-	-	-	-	-	-	TIMESTAMP	10	-	Y	
SQL_TINYINT	-	-	-	-	-	-	SMALLINT	2	-	-	
SQL_VARBINARY	1	32672	-	-	-	-	VARCHAR	-	-	Y	
SQL_VARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	N	
I SQL_WCHAR	1	127	-	-	-	-	CHAR	-	-	N	
I SQL_WCHAR	128	16336	-	-	-	-	VARCHAR	-	-	N	
I SQL_WVARCHAR	1	16336	-	-	-	-	VARCHAR	-	-	N	
I SQL_WLONGVARCHAR	-	1073741823	-	-	-	-	CLOB	2147483647	-	N	

Sybase データ・ソース

表 264. Sybase CTLIB および DBLIB のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
バイナリー	1	254	-	-	-	-	CHAR	-	-	Y
バイナリー	255	255	-	-	-	-	VARCHAR	-	-	Y
bit	-	-	-	-	-	-	SMALLINT	-	-	-
char	1	254	-	-	-	-	CHAR	-	-	N
char	255	255	-	-	-	-	VARCHAR	-	-	N
datetime	-	-	-	-	-	-	TIMESTAMP	-	-	-
datetimen	-	-	-	-	-	-	TIMESTAMP	-	-	-
decimal	1	31	0	31	-	-	DECIMAL	-	-	-
decimal	32	-	-	-	-	-	DOUBLE	-	-	-
decimaln	1	31	0	31	-	-	DECIMAL	-	-	-
decimaln	32	-	-	-	-	-	DOUBLE	-	-	-
float	-	4	-	-	-	-	REAL	-	-	-
float	-	8	-	-	-	-	DOUBLE	-	-	-
floatn	-	4	-	-	-	-	REAL	-	-	-
floatn	-	8	-	-	-	-	DOUBLE	-	-	-
image	-	-	-	-	-	-	BLOB	-	-	-
int	-	-	-	-	-	-	INTEGER	-	-	-
intn	-	-	-	-	-	-	INTEGER	-	-	-
money	-	-	-	-	-	-	DECIMAL	19	4	-
moneyn	-	-	-	-	-	-	DECIMAL	19	4	-
nchar	1	254	-	-	-	-	CHAR	-	-	N
nchar	255	255	-	-	-	-	VARCHAR	-	-	N

参照タイプの比較

表 264. Sybase CTLIB および DBLIB のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります) (続き)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
numeric	1	31	0	31	-	-	DECIMAL	-	-	-
numeric	32	-	-	-	-	-	DOUBLE	-	-	-
numericn	1	31	0	31	-	-	DECIMAL	-	-	-
numericn	32	-	-	-	-	-	DOUBLE	-	-	-
nvarchar	1	255	-	-	-	-	VARCHAR	-	-	N
real	-	-	-	-	-	-	REAL	-	-	-
smalldatetime	-	-	-	-	-	-	TIMESTAMP	-	-	-
smallint	-	-	-	-	-	-	SMALLINT	-	-	-
smallmoney	-	-	-	-	-	-	DECIMAL	10	4	-
sysname	1	254	-	-	-	-	CHAR	-	-	N
text	-	-	-	-	-	-	CLOB	-	-	-
timestamp	-	-	-	-	-	-	VARCHAR	8	-	Y
tinyint	-	-	-	-	-	-	SMALLINT	-	-	-
varbinary	1	255	-	-	-	-	VARCHAR	-	-	Y
varchar	1	255	-	-	-	-	VARCHAR	-	-	N

データ・ソース

通常、連合システムのデータ・ソースは、リレーショナル DBMS インスタンス (Oracle や Sybase など) および、そのインスタンスによりサポートされる 1 つまたは複数のデータベースです。しかし、連合システムには、以下に示すような他のタイプのデータ・ソース (たとえば life sciences データ・ソースや検索アルゴリズム) を含めることもできます。

- スプレッドシート。Microsoft® Excel など。
- 検索アルゴリズム。BLAST など。

- 表構造ファイル。このタイプのファイルは、一連のレコードからなる規則的な構造を持ちます。各レコードには同じ数のフィールドがあり、それらは特定の区切り文字で区切られています。2つの区切り文字を続けると NULL 値を表します。
- Documentum 文書管理ソフトウェア。これには、文書の内容、属性、リレーションシップ、バージョン、翻訳、形式、ワークフロー、およびセキュリティーを保管するリポジトリが含まれます。
- XML タグ・ファイル。

DB2[®] Universal Database (UNIX[®] および Windows 版) では、サポートされるデータ・ソースは以下のとおりです。

表 265. サポート対象データ・ソースのバージョンおよびアクセス方式

データ・ソース	サポートされるデータ・ソース・バージョン	アクセス方式	注
DB2 Universal Database [™] (UNIX および Windows 版) [®]	6.1、7.1、7.2、8.1	DRDA [®]	DB2 バージョン 8 に直接統合
DB2 Universal Database (z/OS [™] および OS/390 版) [®]	5 (PTF PQ07537 を適用) 以上	DRDA	DB2 バージョン 8 に直接統合
DB2 Universal Database (iSeries [™] 版)	4.4 以上	DRDA	DB2 バージョン 8 に直接統合
DB2 Server (VM および VSE 版)	3.3 以上	DRDA	DB2 バージョン 8 に直接統合
Informix [™]	7、8、9	Informix Client SDK	DB2 バージョン 8 に直接統合
ODBC		ODBC 3.0 ドライバ	DB2 Relational Connect が必要
OLE DB		OLE DB 2.0 以上	DB2 バージョン 8 に直接統合
Oracle	7.x、8.x、9.x	SQL*Net または Net8 クライアント・ソフトウェア	DB2 Relational Connect が必要

参照タイプの比較

表 265. サポート対象データ・ソースのバージョンおよびアクセス方式 (続き)

データ・ソース	サポートされる データ・ソース・ バージョン	アクセス方式	注
Microsoft SQL Server	6.5、7.0、2000	Windows では、Microsoft SQL Server Client ODBC 3.0 以上のドライバー。UNIX では、Data Direct Technologies (以前の MERANT) Connect ODBC 3.6 ドライバー	DB2 Relational Connect が必要
Sybase	10.0、11.0、11.1、11.5、11.9、12.0	Sybase Open Client	DB2 Relational Connect が必要
BLAST	2.1.2	BLAST デーモン (ラッパーと一緒に提供される)	DB2 Life Sciences Data Connect が必要
Documentum	Documentum サーバー: EDMS 98 (バージョン 3 と呼ばれる) および 4i。	Documentum Client API/Library	DB2 Life Sciences Data Connect が必要
Microsoft Excel	97、2000	なし	DB2 Life Sciences Data Connect が必要
表構造ファイル		なし	DB2 Life Sciences Data Connect が必要
XML	1.0 仕様	なし	DB2 Life Sciences Data Connect が必要

データ・ソースは準自立的です。たとえば、連合サーバーは、Oracle アプリケーションがデータ・ソースにアクセスしている同じ時に、それらの Oracle データ・ソースに照会を送信することができます。DB2 連合システムは、保水性およびロッキング制約を超えて、他のデータ・ソースへのアクセスを独占したり、制限することはありません。

REBIND_ROUTINE_PACKAGE

►►—REBIND_ROUTINE_PACKAGE—(—type—, —routine_name_string—, —resolve—)————►►

そのスキーマは SYSPROC です。

REBIND_ROUTINE_PACKAGE プロシージャは、SQL プロシージャに関連付けられたパッケージを再バインドします。これは機能的には REBIND コマンドと同じですが、これは、パッケージ名の代わりにプロシージャ名を引き数として使用します。REBIND_ROUTINE_PACKAGE プロシージャは、コマンド行またはアプリケーションから呼び出すことができます。

type

以下のいずれかの値を使用してルーチンのタイプを指定する、タイプ CHAR(2) の入力引き数。

- P - プロシージャ
- SP - プロシージャの特定名

routine_name_string

ルーチンの修飾名を指定する、タイプ VARCHAR(257) の入力引き数。スキーマ名が指定されていない場合、ルーチンが処理されるときに、デフォルトは CURRENT SCHEMA 特殊レジスタの値です。 *routine_name_string* に二重引用符 (") を組み込むことはできません。

resolve

どのバインディング・セマンティクスを使用するかを指定する、タイプ VARCHAR(12) の入力引き数。 ANY 値は、SQL パスのどの関数およびタイプでも、関数およびタイプの解決に考慮されることを示します。 CONSERVATIVE 値は、最後の明示的なバインドのタイム・スタンプより前に定義された SQL パスの関数およびタイプだけが、関数およびタイプの解決に考慮されることを示します。

ルーチンの修飾名は、検索するルーチンを決定するために使用されます。検出されるルーチンは、SQL ルーチンでなければなりません。それ以外は、エラーが返されます。(SQLSTATE 428F7) 特定名が使用されない場合は、複数のルーチンが検出される可能性があります、エラーが返されます。(SQLSTATE 42725) このような場合は、必ず、検索したいルーチンの固有の名前を使用してください。

SYSCAT.COLIDENTATTRIBUTES

表に定義されている各 IDENTITY 列ごとに 1 つの行が含まれます。

表 266. SYSCAT.COLIDENTATTRIBUTES カタログ・ビューの列

列名	データ・タイプ	NULL 可能	説明
TABSCHEMA	VARCHAR(128)		この列を含む表またはビューの修飾名。
TABNAME	VARCHAR(128)		
COLNAME	VARCHAR(128)		列名。
START	DECIMAL(31,0)		開始値
INCREMENT	DECIMAL(31,0)		増分値

SYSCAT.COLIDENTATTRIBUTES

表 266. SYSCAT.COLIDENTATTRIBUTES カタログ・ビューの列 (続き)

列名	データ・タイプ	NULL 可能	説明
MINVALUE	DECIMAL(31,0)		最小値
MAXVALUE	DECIMAL(31,0)		最大値。
CYCLE	CHAR(1)		境界に達したときにサイクリングを行うかどうか。 Y - サイクリングを行う N - サイクリングを行わない
CACHE	INTEGER		アクセスを高速化するために、メモリーに事前割り当てするシーケンス値の数。0 は値が事前割り当てされないことを示します。
SEQID	INTEGER		シーケンスの内部 ID。

ALTER TABLE

ALTER TABLE ステートメントは、以下のことを行うことにより、既存の表を変更します。

- 1 つまたは複数の列を表に追加する
- 主キーの追加、あるいはドロップを行う
- 1 つまたは複数のユニーク制約、または参照制約の追加、あるいはドロップを行う
- 1 つまたは複数のチェック制約定義の追加、あるいはドロップを行う
- 表除去制約の追加、あるいはドロップを行う
- VARCHAR 列の長さを変更する
- 参照タイプ列を変更して、有効範囲を追加する
- 生成される列の生成式を変更する
- 1 つまたは複数のチェック制約属性、あるいは参照制約属性を変更する
- 区分化キーの追加、あるいはドロップを行う
- 表属性 (データ・キャプチャー・オプション、pctfree、ロック・サイズ、追加モードなど) を変更する
- 表を記録されていない初期状態 (NOT LOGGED INITIALLY) に設定する

呼び出し:

このステートメントはアプリケーション・プログラムに組み込むことができ、また動的 SQL ステートメントを使用して出すことができます。 DYNAMICRULES の実行動作がパッケージに効力を持つ場合にのみ、動的に準備できる実行可能ステートメントです (SQLSTATE 42509)。

許可:

ステートメントの許可 ID によって保持されている特権には、少なくとも以下のいずれかが含まれていなければなりません。

- 変更する表に対する ALTER 特権
- 変更する表に対する CONTROL 特権
- 表のスキーマに対する ALTERIN 特権
- SYSADM または DBADM 権限

外部キーの作成/ドロップの場合、このステートメントの許可 ID には、親表に対する以下の特権が少なくとも 1 つ含まれている必要があります。

- その表に対する REFERENCES 特権
- 指定の親キーのそれぞれの列に対する REFERENCES 特権
- 表に対する CONTROL 特権
- SYSADM または DBADM 権限

表 T の主キーまたはユニーク制約をドロップするには、この親キー T に従属しているすべての表において、ステートメントの許可 ID に以下の特権が少なくとも 1 つ含まれている必要があります。

- その表に対する ALTER 特権
- 表に対する CONTROL 特権
- 表のスキーマに対する ALTERIN 特権
- SYSADM または DBADM 権限

(全選択を使用して) 表をマテリアライズ照会表に変更するには、このステートメントの許可 ID に、以下のうち少なくとも 1 つが含まれている必要があります。

- その表に対する CONTROL
- SYSADM または DBADM 権限

なおかつ、全選択で識別された個々の表またはビューに対する以下の特権が少なくとも 1 つ含まれている必要があります。

- その表またはビューに対する SELECT および ALTER 特権
- 表またはビューに対する CONTROL 特権
- その表またはビューに対する SELECT 特権と、その表またはビューのスキーマに対する ALTERIN 特権
- SYSADM または DBADM 権限

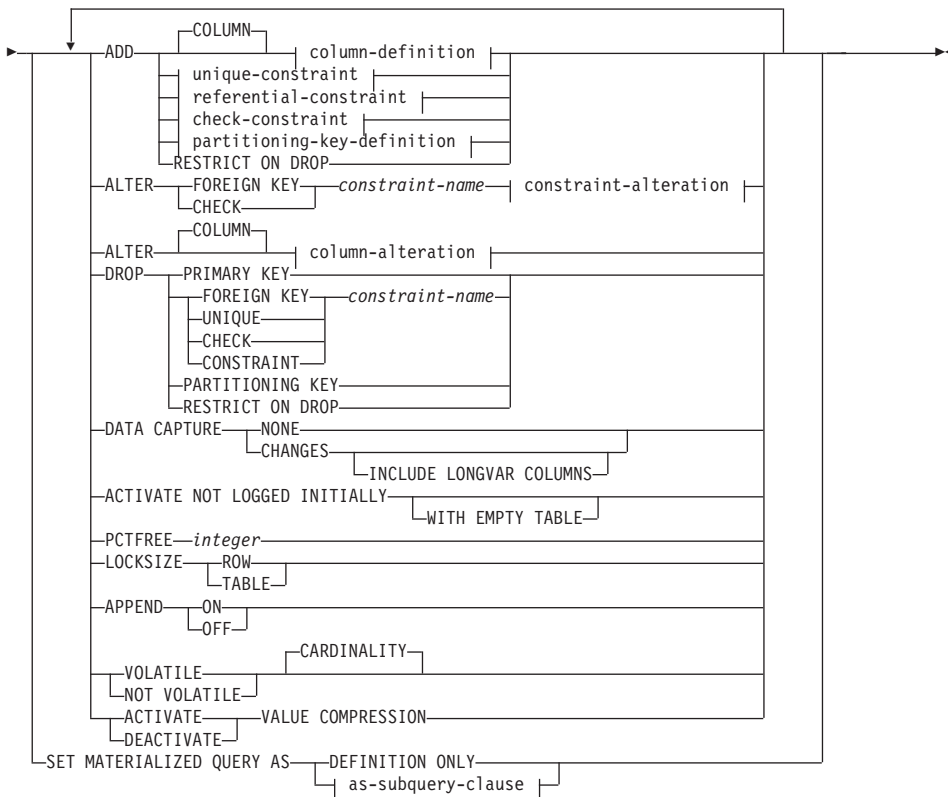
表を変更してマテリアライズ照会表でなくなるようにするには、ステートメントの許可 ID が持っている特権に、このマテリアライズ照会表を定義するのに使用する全選択で識別される各表またはビューに対して、少なくとも以下の 1 つが含まれている必要があります。

ALTER TABLE

- その表またはビューに対する ALTER 特権
- 表またはビューに対する CONTROL 特権
- その表またはビューのスキーマに対する ALTERIN 特権
- SYSADM または DBADM 権限

構文:

→ ALTER TABLE *table-name* →

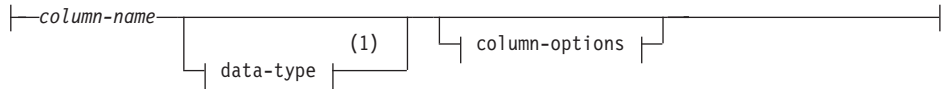
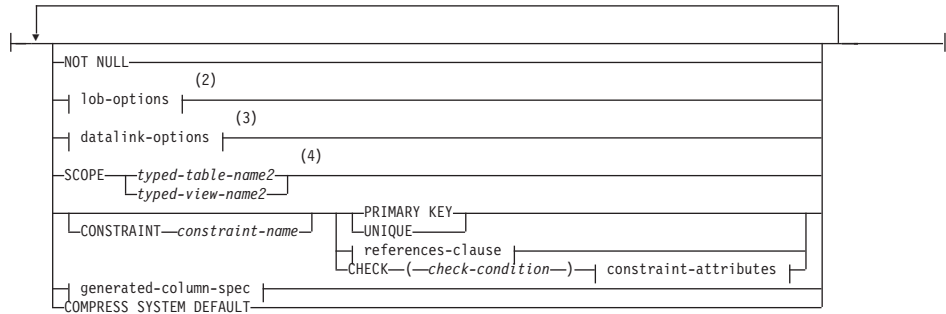


as-subquery-clause:

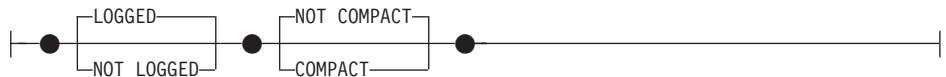
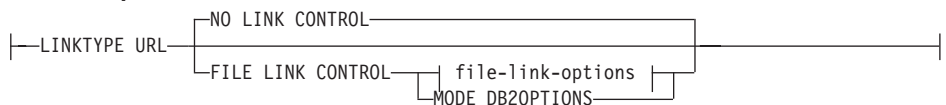
(-fullselect-) materialized-query-table-options

materialized-query-table-options:

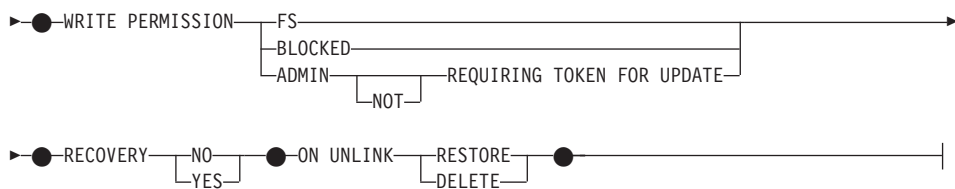
DATA INITIALLY DEFERRED REFRESH DEFERRED IMMEDIATE ENABLE QUERY OPTIMIZATION DISABLE QUERY OPTIMIZATION

column-definition:**column-options:****注:**

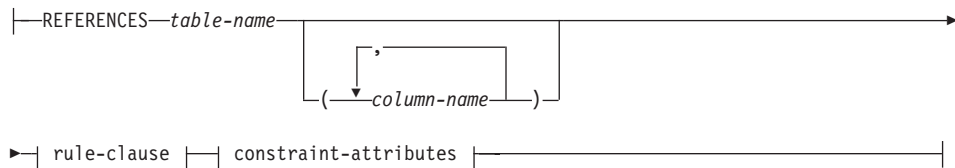
- 1 最初の column-option で generated-column-spec が選択された場合、data-type を省略して、generation-expression によって計算するようにできます。
- 2 lob-options (LOB オプション) 文節は、ラージ・オブジェクト・タイプ (BLOB、CLOB、および DBCLOB) と、ラージ・オブジェクト・タイプに基づく特殊タイプに対してのみ適用されます。
- 3 datalink-options 文節は、DATALINK タイプと、DATALINK タイプに基づく特殊タイプに対してのみ適用されます。
- 4 SCOPE 文節は REF タイプに対してのみ適用されます。

lob-options:**datalink-options:****file-link-options:**

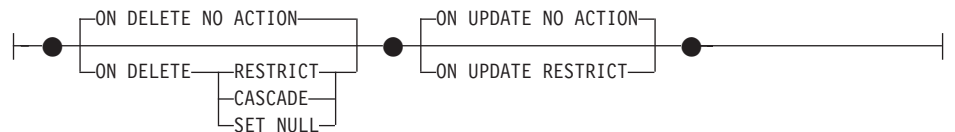
ALTER TABLE



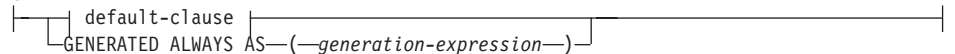
references-clause:



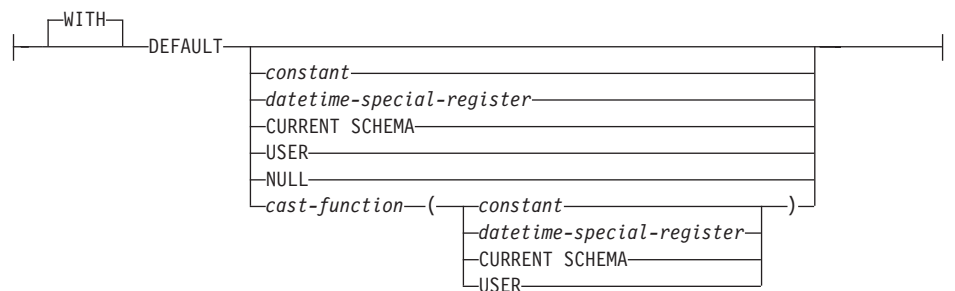
rule-clause:



generated-column-spec:

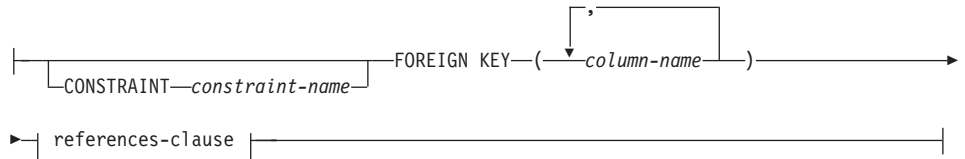
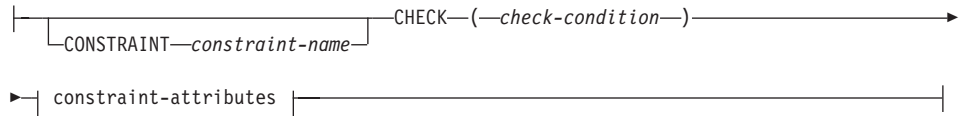
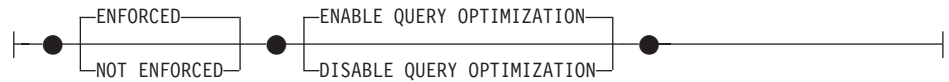
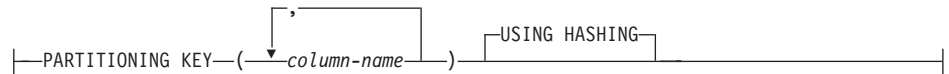
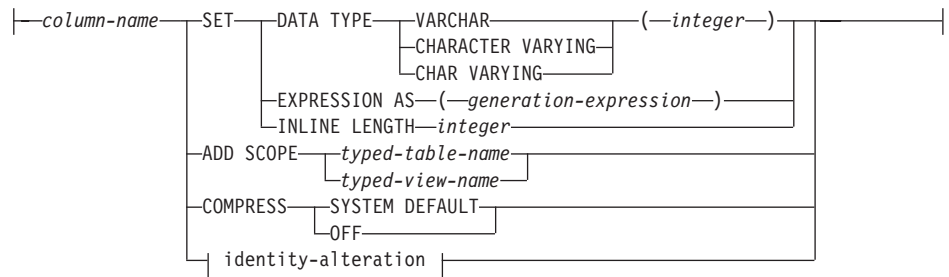


default-clause:

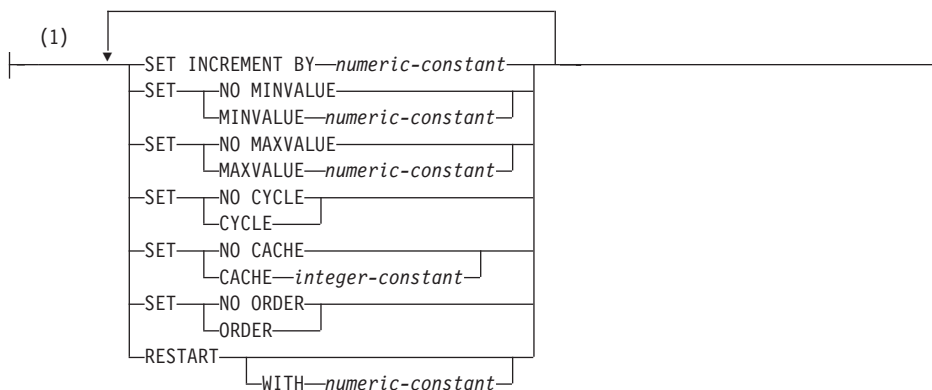


unique-constraint:

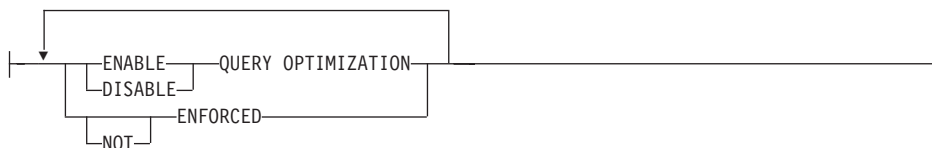


referential-constraint:**check-constraint:****constraint-attributes:****partitioning-key-definition:****column-alteration:****identity-alteration:**

ALTER TABLE



constraint-alteration:



注:

- 1 同じ文節を複数回指定することはできません。

説明:

table-name

変更する表を指定します。これは、カタログに記述されている表でなければならず、ビュー、またはカタログ表を対象にすることはできません。 *table-name* でマテリアライズ照会表を指定している場合、変更は、マテリアライズ照会表を DEFINITION ONLY に設定、NOT LOGGED INITIALLY の活動化、pctfree、locksize、append、または volatile の変更のみを行うことができます。 *table-name* をニックネーム (SQLSTATE 42809) または宣言済み一時表 (SQLSTATE 42995) にすることはできません。

SET MATERIALIZED QUERY AS

マテリアライズ照会表のプロパティの変更を許可します。

DEFINITION ONLY

表がマテリアライズ照会表と見なされなくなるように、マテリアライズ照会表を変更します。 *table-name* で指定される表は、複製ではないマテリアライズ照会表として定義されていなければなりません (SQLSTATE 428EW)。

table-name の列の定義は変更されませんが、照会の最適化にこの表を使用することはできなくなり、REFRESH TABLE ステートメントも使用できなくなります。

as-subquery-clause

照会の最適化で使用するために、正規表をマテリアライズ照会表に変更します。 *table-name* で指定する表には、以下の条件があります。

- マテリアライズ照会表として以前に定義されてはなりません。
- 型付き表であってはなりません。
- 何らかの定数、ユニーク索引、またはトリガーが定義されてはなりません。
- 他のマテリアライズ照会表の定義に参照されてはなりません。

表名が基準に適合しない場合、エラーが戻されます (SQLSTATE 428EW)。

fullselect

表の基礎となる照会を定義します。既存の表の列は、*fullselect* の結果列と以下のような関係になければなりません (SQLSTATE 428EW)。

- 列の数が同数でなければなりません。
- 全く同じデータ・タイプでなければなりません。
- 同じ順序を示す位置に同じ列名がなければなりません。

マテリアライズ照会表に *fullselect* を指定することについての詳細は、『CREATE TABLE』を参照してください。追加の制限事項としては、*table-name* は全選択で直接的にも間接的にも参照できません。

materialized-query-table-options

マテリアライズ照会表を変更するためのリフレッシュ可能オプションを指定します。

DATA INITIALLY DEFERRED

REFRESH TABLE または SET INTEGRITY ステートメントを使用して、表のデータを妥当性検査する必要があります。

REFRESH

表のデータを保守する方法を示します。

DEFERRED

REFRESH TABLE ステートメントを使っていつでも表のデータをリフレッシュできます。表のデータには、REFRESH TABLE ステートメント処理時のスナップショットである照会結果が反映されるにすぎません。この属性を定義したマテリアライズ照会表には、INSERT、UPDATE、または DELETE ステートメントを使用できません (SQLSTATE 42807)。

IMMEDIATE

DELETE、INSERT、または UPDATE の一部として基礎表に加えられた変更は、マテリアライズ照会表にカスケードされます。その場合、表の内容は、どのポイント・イン・タイム指定でも、指

ALTER TABLE

定した `subselect` (副選択) を処理する場合と同じ内容になります。この属性を定義したマテリアライズ照会表には、`INSERT`、`UPDATE`、または `DELETE` ステートメントを使用できません (SQLSTATE 42807)。

ENABLE QUERY OPTIMIZATION

マテリアライズ照会表を照会の最適化に使用できるようにします。

DISABLE QUERY OPTIMIZATION

マテリアライズ照会表を照会の最適化に使用しません。それでもその表を直接照会することはできます。

ADD *column-definition*

列を表に追加します。型付き表を使用することはできません (SQLSTATE 428DH)。表の既存の行に関して、新しい列の値はデフォルト値に設定されます。新しい列はその表の最後の列になります。つまり、当初 n 個の列があった場合、追加された列は列 $n+1$ になります。

新しい列を追加する場合、すべての列のバイト・カウントの合計が、最大レコード・サイズを超えてはなりません。

column-name

表に追加する列の名前です。名前は非修飾でなければなりません。表にすでにある列名は使用できません (SQLSTATE 42711)。

data-type

『CREATE TABLE』の項に示されるデータ・タイプのいずれかです。

NOT NULL

列に NULL 値が入るのを防止します。 *default-clause* (DEFAULT 文節) も指定する必要があります (SQLSTATE 42601)。

lob-options

LOB データ・タイプのオプションを指定します。『CREATE TABLE』の *lob-options* を参照してください。

datalink-options

DATALINK データ・タイプのオプションを指定します。『CREATE TABLE』の *datalink-options* を参照してください。

SCOPE

参照タイプ列の有効範囲を指定します。

typed-table-name2

型付き表の名前。 *column-name* のデータ・タイプは REF(S) でなければなりません。 S は *typed-table-name2* のタイプを表します (SQLSTATE 428DM)。値が *typed-table-name2* の既存行を実際に参照していることを確認するための、 *column-name* のデフォルト値の検査は行われません。

typed-view-name2

型付きビューの名前。 *column-name* のデータ・タイプは REF(S) でなければなりません。 S は *typed-view-name2* のタイプを表します (SQLSTATE 428DM)。値が *typed-view-name2* の既存行を実際に参照していることを確認するための、 *column-name* のデフォルト値の検査は行われません。

CONSTRAINT *constraint-name*

制約の名前を指定します。制約名 (*constraint-name*) は、同じ ALTER TABLE ステートメントにすでに指定されている制約、あるいは表に既存の他の制約の名前であってはなりません (SQLSTATE 42710)。

ユーザーが制約名を指定しない場合は、表に定義されている既存の制約の ID の中で固有な 18 文字の ID がシステムによって生成されます。(ID は、"SQL" と、タイム・スタンプに基づいて生成される 15 の数字から構成されます。)

PRIMARY KEY 制約またはユニーク制約とともに使用した場合、この *constraint-name* は、制約をサポートするために作成される索引の名前として使用されます。ユニーク制約に関連した索引名の詳細については、1428 ページの『注』を参照してください。

PRIMARY KEY

これは、1 つの列からなる主キーを定義する簡単な方法として用意されています。つまり、PRIMARY KEY が列 C の定義で指定されている場合、その効果は、PRIMARY KEY(C) 文節が独立した文節として指定された場合と同じです。列に NULL 値を含めることはできないので、NOT NULL 属性も指定する必要があります (SQLSTATE 42831)。

この後の *unique-constraint* の説明の中の PRIMARY KEY を参照してください。

UNIQUE

これは、1 つの列からなるユニーク・キーを定義する簡単な方法です。すなわち、UNIQUE を列 C の定義に指定すると、UNIQUE(C) 文節を独立した文節として指定した場合と同じ結果になります。

この後の *unique-constraint* の説明の中の UNIQUE を参照してください。

references-clause

これは、1 つの列からなる外部キーを定義する簡単な方法として用意されています。つまり、*references-clause* が列 C の定義に指定されている場合、その効果は、列として C しか指定されていない FOREIGN KEY 文節の一部として *references-clause* が指定された場合と同じになります。

『CREATE TABLE』の *references-clause* を参照してください。

ALTER TABLE

CHECK (*check-condition*)

これは、1 つの列に適用されるチェック制約を定義する簡単な方法として用意されています。『CREATE TABLE』の *check-condition* を参照してください。

generated-column-spec

column-generation の詳細については、『CREATE TABLE』を参照してください。

default-clause

列のデフォルト値を指定します。

WITH

オプション・キーワード。

DEFAULT

INSERT で値が提供されなかった場合、もしくは INSERT や UPDATE で DEFAULT が指定されている場合に、デフォルト値を提供します。

DEFAULT キーワードの後に特定のデフォルト値の指定がない場合のデフォルト値は、列のデータ・タイプによって異なります。表 267 を参照してください。列が DATALINK または構造タイプとして定義されている場合、DEFAULT 文節を指定することはできません。

列が特殊タイプを使用して定義される場合、列のデフォルト値は、特殊タイプにキャストされたソース・データ・タイプのデフォルト値になります。

表 267. デフォルト値 (値が指定されない場合)

データ・タイプ	デフォルト値
数値	0
固定長文字ストリング	ブランク
可変長文字ストリング	長さ 0 のストリング
固定長 GRAPHIC ストリング	2 バイトのブランク
可変長 GRAPHIC ストリング	長さ 0 のストリング
日付	既存の行の場合、0001 年 1 月 1 日に対応する日付。追加行の場合には、現在の日付。
時刻	既存の行の場合、0 時間 0 分 0 秒に対応する時刻。追加行の場合には、現在の時刻。
タイム・スタンプ	既存の行の場合、0001 年 1 月 1 日 0 時 0 分 0 秒 0 マイクロ秒に対応する日付。追加行の場合には、現在のタイム・スタンプ。

表 267. デフォルト値 (値が指定されない場合) (続き)

データ・タイプ	デフォルト値
2 進ストリング (blob)	長さ 0 のストリング

column-definition から DEFAULT を省略すると、その列のデフォルト値として NULL 値が使用されます。

DEFAULT キーワードに指定できる値のタイプは、以下のとおりです。

constant

列のデフォルト値として定数を指定します。指定する定数は、以下の条件を満たしていなければなりません。

- 第 3 章に示されている割り当ての規則にしたがって、その列に割り当てることができる値でなければなりません。
- その列が浮動小数点データ・タイプとして定義されている場合を除き、浮動小数点の定数を指定してはなりません。
- 定数が 10 進定数の場合、その列のデータ・タイプの位取りを超えるゼロ以外の数字を含めてはなりません (たとえば、DECIMAL(5,2) の列のデフォルト値として 1.234 を指定することはできません)。
- 指定する定数が 254 文字を超えてはなりません。この制約には、引用符文字や 16 進定数の X などの先行文字も含まれます。さらに、定数が *cast-function* の引き数の場合には、完全修飾された関数名から取った文字や括弧も含めて、この制限を超えてはなりません。

datetime-special-register

INSERT、UPDATE、または LOAD の実行時における日時特殊レジスタの値 (CURRENT DATE、CURRENT TIME、または CURRENT TIMESTAMP) を、その列のデフォルト値として指定します。その列のデータ・タイプは、指定した特殊レジスターに対応するデータ・タイプでなければなりません (たとえば、CURRENT DATE を指定した場合、データ・タイプは DATE でなければなりません)。既存の行の場合は、ALTER TABLE ステートメントが処理される時点の現行日付、現行時刻、または現行タイム・スタンプが値として使用されます。

CURRENT SCHEMA

INSERT、UPDATE、または LOAD の実行時における CURRENT SCHEMA 特殊レジスタの値を、その列のデフォルト値として指定します。CURRENT SCHEMA を指定した場合、その列のデータ・タイプは、CURRENT SCHEMA の長さ属性よりも長い等しい文字ス

ALTER TABLE

トリングでなければなりません。既存の行の場合は、ALTER TABLE ステートメントが処理される時点における CURRENT SCHEMA 特殊レジスターの値です。

USER

INSERT、UPDATE、または LOAD の実行時における USER 特殊レジスターの値を、その列のデフォルト値として指定します。USER を指定した場合、その列のデータ・タイプは、USER の長さ属性よりも長いか等しい文字ストリングでなければなりません。既存の行の場合、値は ALTER TABLE ステートメントの許可 ID になります。

NULL

その列のデフォルト値として NULL を指定します。NOT NULL の指定がある場合には、DEFAULT NULL を同じ列定義に指定してはなりません。

cast-function

この形式のデフォルト値は、特殊タイプ (distinct type)、BLOB、または日時 (DATE、TIME、または TIMESTAMP) データ・タイプとして定義された列に対してのみ使用することができます。特殊タイプの場合、BLOB や日時タイプに基づく例外があり、関数名が列の特殊タイプの名前に一致していなければなりません。スキーマ名で修飾されている場合には、その特殊タイプのスキーマ名と同じでなければなりません。修飾されていない場合には、関数の解決に用いるスキーマ名は特殊タイプのスキーマ名と同じでなければなりません。日時タイプに基づく特殊タイプで、デフォルト値が定数の場合、必ず関数を使用する必要があります。さらに、その関数名は、暗黙または明示のスキーマ名 SYSIBM を持つ特殊タイプのソース・タイプ名に一致していなければなりません。他の日時列の場合は、対応する日時関数も使用できます。BLOB に基づく BLOB または特殊タイプの場合も、関数を使用する必要があります。その関数名は、暗黙または明示のスキーマ名 SYSIBM を持つ BLOB でなければなりません。

constant

引き数として定数を指定します。指定する定数は、特殊タイプのソース・タイプに関する定数の規則 (特殊タイプでない場合は、データ・タイプに関する定数の規則) にしたがっていないければなりません。cast-function が BLOB の場合には、定数としてストリング定数を指定する必要があります。

datetime-special-register

CURRENT DATE、CURRENT TIME、または CURRENT TIMESTAMP を指定します。列の特殊タイプのソース・タイプは、指定した特殊レジスターに対応するデータ・タイプでなければなりません。

CURRENT SCHEMA

CURRENT SCHEMA 特殊レジスタの値を指定します。列の特殊タイプのソース・タイプのデータ・タイプは、CURRENT SCHEMA 特殊レジスタの長さ属性よりも長い等しい文字ストリングでなければなりません。cast-function が BLOB の場合には、長さ属性が 8 バイト以上でなければなりません。

USER

USER 特殊レジスタを指定します。列の特殊タイプのソース・タイプのデータ・タイプは、少なくとも 8 バイトの長さのストリング・データ・タイプでなければなりません。cast-function が BLOB の場合には、長さ属性が 8 バイト以上でなければなりません。

指定した値が無効な場合、エラー (SQLSTATE 42894) が発生します。

COMPRESS SYSTEM DEFAULT

システム・デフォルト値 (つまり、特定の値が指定されない場合にデータ・タイプとして使用されるデフォルト値) が最小限のスペースを使用して保管されるように指定します。VALUE COMPRESSION 文節が指定されていない場合には警告が出され (SQLSTATE 01648)、システム・デフォルト値は最小限のスペースを使用しては保管されません。

システム・デフォルト値がこのような方法で保管されると、列上での許可や更新操作の際に余分なチェックが行われるために、若干パフォーマンスが低下します。

基本データ・タイプは、DATE、TIME、または TIMESTAMP であってはなりません (SQLSTATE 42842)。基本データ・タイプが可変長ストリングの場合には、この文節は無視されます。表が VALUE COMPRESSION に設定されている場合は、長さ 0 のストリング値は自動的に圧縮されます。

ADD unique-constraint

ユニーク制約または主キー制約を定義します。主キー制約またはユニーク制約を、副表に追加することはできません (SQLSTATE 429B3)。階層最上部のスーパー表の場合、制約はその表および関連する副表すべてに適用されます。

CONSTRAINT constraint-name

主キー制約、またはユニーク制約の名前を指定します。詳細については、『CREATE TABLE』で *constraint-name* を参照してください。

UNIQUE (column-name...,)

指定した列で構成されるユニーク・キーを定義します。指定する列は NOT NULL として定義されていなければなりません。各 *column-name* (列名) は、表の列を指定するものでなければなりません。また、同じ列を複数回指定することはできません。名前は非修飾でなければなりません。指定する列の数は 16 を超えてはならず、保管されるそれらの長さの合計は 1024 を超えてはなりません。列の長さ属性が 1024 バイト以内に収まる場合でも、LOB、LONG

VARCHAR、LONG VARGRAPHIC、DATALINK、これらのタイプのうちのいずれかに基づく特殊タイプ、または構造タイプは、ユニーク・キーの一部として使用できません (SQLSTATE 54008)。ユニーク・キーにある一連の列は、主キーまたは他のユニーク・キーの一連の列と同じにすることはできません (SQLSTATE 01543)。LANGLEVEL が SQL92E または MIA の場合には、エラーが戻されます (SQLSTATE 42891)。指定した一連の列に存在する値は、固有である必要があります (SQLSTATE 23515)。

既存の索引がユニーク・キー定義に一致するかどうかを判別するチェックが行われます (索引の INCLUDE 列は無視されます)。列の順序や方向 (ASC/DESC) の指定に関係なく、同じ一連の列を指定していると、索引定義は一致します。一致する索引定義が見つかり、その索引の記述は、システムによりその索引が必要であることを示すように変更され、索引が固有でない場合はユニーク索引に変更されます (固有性を確実にした後)。その表に一致する索引が複数ある場合、既存のユニーク索引が選択されます (選択は任意に行われます)。一致する索引が見つからない場合は、CREATE TABLE で説明するように、その列に対してユニーク索引が自動的に作成されます。ユニーク制約に関連した索引名の詳細については、1428 ページの『注』を参照してください。

PRIMARY KEY *...(column-name,)*

指定された列で構成される主キーを定義します。各 *column-name* (列名) は、表の列を指定していなければなりません。また、同じ列を複数回指定することはできません。名前は非修飾でなければなりません。指定する列の数は 16 を超えてはならず、保管されるそれらの長さの合計は 1024 を超えてはなりません。表には主キーがあってはならず、指定する列は NOT NULL として定義されているものでなければなりません。列の長さ属性が 1024 バイト以内に収まる場合でも、LOB、LONG VARCHAR、LONG VARGRAPHIC、DATALINK、これらのタイプのうちのいずれかに基づく特殊タイプ、または構造タイプは、主キーの一部として使用できません (SQLSTATE 54008)。主キーの一連の列は、ユニーク・キーの一連の列と同じであってはなりません (SQLSTATE 01543)。LANGLEVEL が SQL92E または MIA の場合には、エラーが戻されます (SQLSTATE 42891)。指定した一連の列に存在する値は、固有である必要があります (SQLSTATE 23515)。

既存の索引が主キー定義に一致するかどうかを判別するチェックが行われます (索引の INCLUDE 列は無視されます)。列の順序や方向 (ASC/DESC) の指定に関係なく、同じ一連の列を指定していると、索引定義は一致します。一致する索引定義が見つかり、その索引の記述は、その索引が 1 次索引である (システムが必要としている) ことを示すように変更され、索引が固有でない場合はユニーク索引に変更されます (固有性を確実にした後)。その表に一致する索引が複数ある場合、既存のユニーク索引が選択されます (選択は任意に行われます)。一致する索引が見つからない場合は、CREATE TABLE で説明するように、その列に対してユニーク索引が自動的に作成されます。ユニーク制約に関連した索引名の詳細については、1428 ページの『注』を参照してください。

1 つの表には、主キーを 1 つだけ定義することができます。

ADD referential-constraint

参照制約を定義します。『CREATE TABLE』の *referential-constraint* を参照してください。

ADD check-constraint

チェック制約を定義します。『CREATE TABLE』の *check-constraint* を参照してください。

ADD partitioning-key-definition

区分化キーを定義します。表は、単一パーティションのデータベース・パーティション・グループにある表スペースに定義する必要があり、すでに区分化キーを持ってはなりません。区分化キーが表にすでに存在している場合には、新しい区分化キーを追加する前に既存のキーをドロップする必要があります。

区分化キーを、副表に追加することはできません (SQLSTATE 428DH)。

PARTITIONING KEY (column-name...)

指定した列を使用して、区分化キーを定義します。各 *column-name* (列名) は、表の列を指定していなければなりません。また、同じ列を複数回指定することはできません。名前は非修飾でなければなりません。列のデータ・タイプが LONG VARCHAR、LONG VARGRAPHIC、BLOB、CLOB、DBCLOB、DATALINK、これらのいずれかのタイプの特殊タイプ、または構造タイプである場合、区分化キーの一部として列を使用することはできません。

USING HASHING

データ配分の区分化方式として、ハッシュ関数を使用することを指定します。これは、サポートされる唯一の区分化方式です。

ADD RESTRICT ON DROP

表をドロップできないように、また、表を含む表スペースをドロップできないように指定します。

ALTER FOREIGN KEY constraint-name

制約名が *constraint-name* の参照制約を変更します。 *constraint-name* は既存の参照制約を指定する必要があります (SQLSTATE 42704)。

ALTER CHECK constraint-name

制約名が *constraint-name* のチェック制約を変更します。 *constraint-name* は既存の参照制約を指定する必要があります (SQLSTATE 42704)。

constraint-alteration

参照制約またはチェック制約に関連付けられた属性の変更のオプションです。

ENFORCED

制約を ENFORCED に変更します。この制約は、挿入、更新、削除などの通常の操作中にデータベース・マネージャーによって課せられます。

ALTER TABLE

NOT ENFORCED

制約を NOT ENFORCED に変更します。この制約は、挿入、更新、削除などの通常の操作中にデータベース・マネージャーによって課せられません。この制約に適合することが分かっている表データだけに、制約を指定してください。

ENABLE QUERY OPTIMIZATION

適切な状況下では、制約を照会最適化に使用することができます。

DISABLE QUERY OPTIMIZATION

制約を照会の最適化に使用できません。

ALTER *column-alteration*

列の特性を変更します。

column-name

表で変更する列の名前です。 *column-name* は、表の既存列を指定するものでなければなりません (SQLSTATE 42703)。名前は非修飾でなければなりません。

SET DATA TYPE VARCHAR (*integer*)

既存の VARCHAR 列の長さを増やします。 CHARACTER VARYING または CHAR VARYING を、 VARCHAR キーワードの同義語として使用することができます。 *column-name* のデータ・タイプは VARCHAR でなければならず、また現行の列の最大長は *integer* の値以下でなければなりません (SQLSTATE 42837)。 *integer* の値の上限は、32 672 です。型付き表を使用することはできません (SQLSTATE 428DH)。

列を変更する場合、すべての列のバイト・カウントの合計が、最大レコード・サイズを超えてはなりません (SQLSTATE 54010)。ユニーク制約または索引で列を使用する場合、新しい長さはユニーク制約または索引の列の保管長の合計が、1024 を超えないようにしなければなりません (SQLSTATE 54008)。

レジストリー変数 DB2_INDEX_2BYTEVARLEN が ON に設定された際に定義された、主キー、およびユニーク・キーを含む、索引の一部である可変長列の長さは、255 バイトより大きい長さに変更が可能です。可変長列が外部キーと関係があっても、レジストリー変数の設定にかかわらず、その列の長さを255 バイトより大きい長さに変更できます。しかし255 バイトより大きい長さのデータは、対応する主キーの列が255 バイトより大きくないと表に挿入できません。レジストリー変数が ON に設定されて主キーが作成される場合にだけ、挿入が可能です。

SET EXPRESSION AS (*generation-expression*)

列の式を、指定された *generation-expression* に変更します。 SET EXPRESSION AS では、 SET INTEGRITY ステートメントを使用して表をチェック・ペンディング状態にする必要があります。 ALTER TABLE ステートメントの後、 SET INTEGRITY ステートメントを使用して、新しい式に対してこの列にあるすべての値を更新およびチェックする必要があります。列は、式

に基づいて生成される列として定義されていなければなりません (SQLSTATE 42837)。また表の DIMENSIONS 文節に現れないようにする必要があります (SQLSTATE 42997)。 *generation-expression* は、生成される列を定義する際に適用されるのと同じ規則に適合する必要があります。 *generation-expression* の結果データ・タイプは、列のデータ・タイプに割り当て可能でなければなりません (SQLSTATE 42821)。

SET INLINE LENGTH *integer*

既存の構造タイプ列のインライン長を変更します。インライン長は、行内の残りの値とともにインラインで保管する構造タイプのインスタンスの最大バイト・サイズを指示します。インラインで保管できない構造タイプのインスタンスは、LOB 値が処理されるのに似た方法で、基本表の行とは別に保管されます。

column-name のデータ・タイプは、構造タイプでなければなりません (SQLSTATE 42842)。

構造タイプ列のデフォルトの INLINE LENGTH は、このタイプのインライン長になります (明示的に指定するか、または CREATE TYPE ステートメント内のデフォルトとして)。構造タイプのインライン長が 292 未満の場合、列のインライン長には値 292 が使われます。

明示的なインライン長の値は増やすことのみ可能で (SQLSTATE -1)、少なくとも 292 でなければならず、32672 を超えてはなりません (SQLSTATE 54010)。

列を変更する場合、すべての列のバイト・カウントの合計が、最大レコード・サイズを超えてはなりません (SQLSTATE 54010)。

残りの行とは別々に既に保管されたデータは、このステートメントによりインラインに移動されません。構造タイプ列のインライン長の変更の利点を生かすには、列のインライン長を変更した後に指定された表に対して REORG コマンドを呼び出します。

ADD SCOPE

有効範囲が未定義である既存の参照タイプ列に、有効範囲を追加します (SQLSTATE 428DK)。変更する表が型付き表である場合、列をスーパー表から継承することはできません (SQLSTATE 428DJ)。

typed-table-name

型付き表の名前。 *column-name* のデータ・タイプは REF(*S*) でなければなりません。 *S* は *typed-table-name* のタイプを表します (SQLSTATE 428DM)。値が *typed-table-name* の既存行を実際に参照していることを確認するための、 *column-name* の既存値の検査は行われません。

typed-view-name

型付きビューの名前。 *column-name* のデータ・タイプは REF(*S*) でなければなりません。 *S* は *typed-view-name* のタイプを表します (SQLSTATE

ALTER TABLE

428DM)。値が *typed-view-name* の既存行を実際に参照していることを確認するための、*column-name* の既存値の検査は行われません。

COMPRESS

この列のデフォルト値をさらに効率よく保管するかどうかを指定します。

SYSTEM DEFAULT

システム・デフォルト値 (つまり、特定の値が指定されない場合にデータ・タイプとして使用されるデフォルト値) が最小限のスペースを使用して保管されるように指定します。活動化された VALUE COMPRESSION 属性に表が設定されていない場合には、警告が出され (SQLSTATE 01648)、システム・デフォルト値は最小限のスペースを使用しては保管されません。

システム・デフォルト値がこのような方法で保管されると、列上での許可や更新操作の際に余分なチェックが行われるために、若干パフォーマンスが低下します。

列の既存のデータは変更されません。既存のデータを使用可能にして、最小限のスペースを使用してシステム・デフォルト値を保管することの利点を生かすため、オフラインでの表再編成を考慮してください。

OFF

システム・デフォルト値が列の正規値として保管されるように指定します。列の既存のデータは変更されません。既存のデータの変更のために、オフラインでの再編成をお勧めします。

基本データ・タイプは、DATE、TIME、または TIMESTAMP であってはなりません (SQLSTATE 42842)。基本データ・タイプが可変長ストリングの場合には、この文節は無視されます。表が VALUE COMPRESSION に設定されている場合は、長さ 0 のストリング値は自動的に圧縮されます。

変更する表が型付き表である場合、列をスーパー表から継承することはできません (SQLSTATE 428DJ)。

SET INCREMENT BY *numeric-constant*

連続した ID 列値の間隔を指定します。次に生成される IDENTITY 列の値は、最後に割り当てられた値に増分を適用することによって決まります。列はあらかじめ IDENTITY 属性で定義されていなければなりません (SQLSTATE 42837)。

この値は、小数点の右側に非ゼロの数字がない (SQLSTATE 428FA) かぎり、この列に割り当てることができる正または負の値にすることができ (SQLSTATE 42815)、長精度整数定数の値を超えることはありません (SQLSTATE 42820)。

この値が負の場合、ALTER ステートメント以降は降順になります。この値が 0 の場合、または正の場合は、ALTER ステートメント以降は昇順になります。

SET NO MINVALUE または **MINVALUE** *numeric-constant*

降順 IDENTITY 列が値の生成を循環または停止する最小値、あるいは最大値に達した後、昇順 IDENTITY 列が循環する最小値を指定します。列は、指定した表の中に存在していなければならず (SQLSTATE 42703)、あらかじめ IDENTITY 属性で定義されていなければなりません (SQLSTATE 42837)。

NO MINVALUE

昇順シーケンスの場合、値は元の開始値になります。降順シーケンスの場合、列のデータ・タイプの最小値です。

MINVALUE *numeric-constant*

最小値である数値定数を指定します。この値は、小数点の右側に非ゼロの数字がない (SQLSTATE 428FA、SQLCODE -336) かぎり、この列に割り当てることができる正または負の値にすることができます (SQLSTATE 42815、SQLCODE -846) が、最大値以下でなければなりません (SQLSTATE 42815、SQLCODE -846)。

SET NO MAXVALUE または **MAXVALUE** *numeric-constant*

昇順 IDENTITY 列が値の生成を循環または停止する最大値、あるいは最小値に達した後、降順 IDENTITY 列が循環する最大値を指定します。列は、指定した表の中に存在していなければならず (SQLSTATE 42703)、あらかじめ IDENTITY 属性で定義されていなければなりません (SQLSTATE 42837)。

NO MAXVALUE

昇順シーケンスの場合、値は列のデータ・タイプの最大値です。降順シーケンスの場合、値は元の開始値です。

MAXVALUE *numeric-constant*

最大値である数値定数を指定します。この値は、小数点の右側に非ゼロの数字がない (SQLSTATE 428FA) かぎり、この列に割り当てることができる正または負の値にすることができます (SQLSTATE 42815) が、最小値以下でなければなりません (SQLSTATE 42815)。

SET NO CYCLE または **CYCLE**

その最大値または最小値が生成された後、この ID 列が値の生成を続行するかどうかを指定します。列は、指定した表の中に存在していなければならず (SQLSTATE 42703)、あらかじめ IDENTITY 属性で定義されていなければなりません (SQLSTATE 42837)。

NO CYCLE

最大値または最小値に達した後、ID 列について値が生成されないことを指定します。

CYCLE

最大値または最小値に達した後、この列について値の生成が続行されることを指定します。このオプションが使用されると、昇順 IDENTITY 列が最大値に達した後は、その最小値が生成されます。降順 IDENTITY 列が最小値に達した後は、その最大値が生成されます。ID 列の最大値および最小値は、循環に使用される範囲を決定します。

CYCLE が有効な場合、IDENTITY 列について重複する値が生成される可能性があります。固有値が必要であれば (実際には必要ありません)、IDENTITY 列を使用して 1 列のユニーク索引を定義することによって、固有性を確実にしてください。このような IDENTITY 列にユニーク索引が存在し、固有ではない値が生成されると、エラーが起きます (SQLSTATE 23505)。

SET NO CACHE または CACHE *integer-constant*

特定の事前割り振り値を、高速アクセスできるようにメモリーに保存するかどうかを指定します。これはパフォーマンスおよびチューニング・オプションです。列はあらかじめ IDENTITY 属性で定義されていなければなりません (SQLSTATE 42837)。

NO CACHE

ID 列の値を事前割り振りしないことを指定します。データ共用環境では、IDENTITY 値は要求の順序で生成されなければならず、NO CACHE オプションを使用する必要があります。

このオプションが指定されると、ID 列の値はキャッシュに保管されません。この場合、新しい ID 値が要求されるたびに、ログに対して非同期入出力が行われます。

CACHE *integer-constant*

事前割り振りされ、メモリーに保管される ID シーケンスの値の数を指定します。ID 列について値が生成される場合、値を事前割り振りしてキャッシュに保管しておく、ログへの非同期入出力が少なくなります。

IDENTITY 列に新しい値が必要でも未使用の値がキャッシュにない場合、値の割り振りはログへの入出力を待機する必要があります。ただし、ID 列に新しい値が必要で、未使用の値がキャッシュにあれば、その ID 値の割り振りが、ログへの入出力なしで素早く行われます。

システム障害に起因するものであっても通常のものであっても、データベース非活動化が起こると、コミットされたステートメントで使用されていないキャッシュ済みシーケンス値はすべて失われます (使用されなくなります)。CACHE オプションに指定する値は、システム障害の際に失われても構わない IDENTITY 列の値の最大数です。

最小値は 2 です (SQLSTATE 42815)。

SET NO ORDER または ORDER

要求の順序で IDENTITY 列の値が生成されるかどうかを指定します。列は、指定した表の中に存在していなければならず (SQLSTATE 42703)、あらかじめ IDENTITY 属性で定義されていなければなりません (SQLSTATE 42837)。

NO ORDER

要求の順序で IDENTITY 列の値を生成する必要がないことを指定します。

ORDER

要求の順序で IDENTITY 列の値が生成されることを指定します。

RESTART または RESTART WITH *numeric-constant*

IDENTITY 列に関連付けられたシーケンスの状態をリセットします。WITH *numeric-constant* が指定されていないと、IDENTITY 列のシーケンスは、作成されたときに開始値として (暗黙的または明示的のいずれかで) 定義された値で再始動されます。

列は、指定した表の中に存在していなければならず (SQLSTATE 42703)、あらかじめ IDENTITY 属性で定義されていなければなりません (SQLSTATE 42837)。RESTART は、START WITH の元の値を変更することはありません。

numeric-constant は数値定数で、小数点の右側に非ゼロの数字がない (SQLSTATE 42815) かぎり、この列に割り当てることができる正または負の値にすることができます。 *numeric-constant* が列の次の値として使用されます。

DROP PRIMARY KEY

主キーの定義、およびその主キーに従属するすべての参照制約をドロップします。表には主キーがなければなりません。

DROP FOREIGN KEY *constraint-name*

制約名が *constraint-name* の参照制約をドロップします。 *constraint-name* (制約名) は、参照制約を指定していなければなりません。参照制約のドロップにより起こることについては、1428 ページの『注』を参照してください。

DROP UNIQUE *constraint-name*

ユニーク制約 *constraint-name* の定義、およびこのユニーク制約に従属するすべての参照制約をドロップします。 *constraint-name* は、既存のユニーク制約を指定していなければなりません。ユニーク制約のドロップにより起こることについては、1428 ページの『注』を参照してください。

DROP CHECK *constraint-name*

制約名が *constraint-name* のチェック制約をドロップします。 *constraint-name* は、表に定義されている既存のチェック制約を指定していなければなりません。

DROP CONSTRAINT *constraint-name*

制約名が *constraint-name* の制約をドロップします。 *constraint-name* は、表に定義されている既存のチェック制約、参照制約、主キー、またはユニーク制約のいずれ

ALTER TABLE

かを指定していなければなりません。制約のドロップにより起こることについては、1428 ページの『注』を参照してください。

DROP PARTITIONING KEY

区分化キーをドロップします。表には区分化キーがある必要があり、表は単一パーティションのデータベース・パーティション・グループで定義されている表スペースに入っている必要があります。

DROP RESTRICT ON DROP

表、および表を含む表スペースのドロップに関する制約事項を削除します。

DATA CAPTURE

データの複製に関する追加情報をログに記録するか否かを指定します。

表が型付き表である場合、このオプションはサポートされません (ルート表の場合は SQLSTATE 428DH で、他の副表の場合は 428DR)。

NONE

追加情報をログに記録しないことを指定します。

CHANGES

この表に対する SQL 変更についての追加情報をログに書き込むことを指定します。このオプションは、表を複製する場合で、キャプチャー・プログラムを使用してログからこの表に対する変更内容を取り込む場合に必須です。

カタログ・パーティション以外のパーティションにデータが置かれるように表が定義されている場合 (複数パーティションのデータベース・パーティション・グループ、またはカタログ・パーティション以外のパーティションを持つデータベース・パーティション・グループ)、このオプションはサポートされません (SQLSTATE 42997)。

表のスキーマ名 (暗黙または明示名) が 18 バイトより長い場合、このオプションはサポートされません (SQLSTATE 42997)。

INCLUDE LONGVAR COLUMNS

データ複製ユーティリティーが、LONG VARCHAR または LONG VARGRAPHIC 列に対する変更を取り込むようにします。この文節は、LONG VARCHAR または LONG VARGRAPHIC 列のない表に指定することもできます。これは、LONG VARCHAR または LONG VARGRAPHIC 列を含むよう、表を ALTER することができるためです。

ACTIVATE NOT LOGGED INITIALLY

現行の作業単位の表の NOT LOGGED INITIALLY 属性を活動化します。

このステートメントにより表を変更した後に、同一の作業単位の INSERT、DELETE、UPDATE、CREATE INDEX、DROP INDEX、または ALTER TABLE によって表に対して行われた変更は、ログ記録されません。NOT LOGGED INITIALLY 属性が活動状態にあるときに、ALTER ステートメントによってシス

テム・カタログに対して行われた変更は、ログ記録されます。同一の作業単位内でシステム・カタログ情報に対して行われる一連の変更は、ログ記録されます。

現行の作業単位が完了すると、NOT LOGGED INITIALLY 属性は非活動化され、それ以降の作業単位の表で行われるすべての操作はログ記録されます。

カタログ表へのデータの挿入中にロックを避けるためにこの機能を使用する場合、ALTER TABLE ステートメントにはこの文節だけを指定してください。ALTER TABLE ステートメントでこの文節以外のものを指定すると、カタログはロックされてしまいます。ALTER TABLE ステートメントでこの文節のみが指定されている場合、SHARE ロックのみがシステム・カタログ表で獲得されます。これにより、このステートメントが実行される時と、このステートメントが実行される作業単位が終了する時の、所要時間の競合が生じるのを、可能な限り抑えることができます。

表が型付き表である場合、このオプションがサポートされるのは、型付き表階層のルート表だけです (SQLSTATE 428DR)。

NOT LOGGED INITIALLY 属性の詳細については、『CREATE TABLE』にあるこの属性に関する記述を参照してください。

注: 活動化された NOT LOGGED INITIALLY 属性を持つ表に対してログに記録されない活動が生じ、ステートメントに障害が起こる (ロールバックが発生する)、または ROLLBACK TO SAVEPOINT が実行される場合には、その作業単位全体がロールバックされます (SQL1476N)。さらに、NOT LOGGED INITIALLY 属性が活動化されている表は、ロールバックされた後にアクセス不能としてマークされ、ドロップしかできなくなります。したがって、NOT LOGGED INITIALLY 属性が活動化されている作業単位内のエラーは、最小限に抑えられます。

WITH EMPTY TABLE

現在表にあるすべてのデータを除去します。一度データが除去されると、RESTORE 機能を使用しなければ、そのデータの回復を行うことができません。この ALTER ステートメントを発行した作業単位をロールバックしても、表データは元の状態には回復できません。

この処置が必要な場合、修復したい表に定義された DELETE トリガーは行われません。その表にある索引もすべて空になります。

PCTFREE *integer*

ロードまたは再編成時に、各ページでフリー・スペースとして残しておくスペースの割合を指定します。 *integer* の値は 0 ~ 99 です。各ページの最初の行は、制約なしに追加されます。行をさらに追加する場合、各ページに少なくとも *integer* パーセントをフリー・スペースとして残します。PCTFREE 値は、LOAD または REORGANIZE TABLE ユーティリティでのみ有効です。表が型付き表である場合、このオプションがサポートされるのは、型付き表階層のルート表だけです (SQLSTATE 428DR)。

ALTER TABLE

LOCKSIZE

表へのアクセス時に使用されるロックのサイズ (細分性) を指定します。表定義でこのオプションを使用しても、通常のロック・エスカレーションが行われます。表が型付き表である場合、このオプションがサポートされるのは、型付き表階層のルート表だけです (SQLSTATE 428DR)。

ROW

行ロックの使用を指定します。これは、表の作成時のデフォルトのロック・サイズです。

TABLE

表ロックの使用を指定します。これは、適切な共用ロックまたは排他ロックが表で獲得されており、意図ロック ("意図なし" は除く) が使用されないことを意味します。この値を使用すると、獲得すべきロック数が限定されるため、照会のパフォーマンスが向上します。しかし、すべてのロックが表全体に対して獲得されるので、並行性も限定されます。

APPEND

データを表データの終わりに追加するか、またはデータ・ページの使用可能なフリー・スペースを位置に追加するかを指定します。表が型付き表である場合、このオプションがサポートされるのは、型付き表階層のルート表だけです (SQLSTATE 428DR)。

ON

表データが追加され、各ページのフリー・スペース情報は保持されません。表にはクラスター索引があってはなりません (SQLSTATE 428CA)。

OFF

表データは使用可能なスペースに入れられます。これは、表の作成時のデフォルト値です。

APPEND OFF を設定した後に表の再編成が必要となります。これは、使用可能なフリー・スペース情報が不正確となるため、データ挿入時のパフォーマンスの低下につながるからです。

VOLATILE

これを指定することにより、オプティマイザーに対し、表 *table-name* のカーディナリティーが、空から非常に大きなものに至るまで、実行時に変化し得ることを知らせます。 *table-name* にアクセスする場合、オプティマイザーは、その統計に関係なく、表のスキャンではなく索引のスキャンを使います。ただしその場合、その索引は索引専用である (参照されるすべての列がその索引内にある) か、索引のスキャンで述部を適用できることが条件になります。リスト・プリフェッチ・アクセス・メソッドは、この表へのアクセスには使用されません。表が型付き表である場合、このオプションがサポートされるのは、型付き表階層のルート表だけです (SQLSTATE 428DR)。

NOT VOLATILE

これを指定することにより、オブティマイザーに対して、*table-name* のカーディナリティーが揮発ではないことを知らせます。この表に対するアクセス・プランは、既存の統計と、所定の最適化レベルに基づいて続けられます。

CARDINALITY

揮発するのが表内の行数であり、表そのものではないことを示すオプション・キーワード。

VALUE COMPRESSION

たいていのデータ・タイプで NULL および長さ 0 のデータ値を、さらに効率的に保管するかどうかを指定します。また、使用される行形式を判別します。表が型付き表である場合、このオプションがサポートされるのは、型付き表階層のルート表だけです (SQLSTATE 428DR)。

ACTIVATE

データ・タイプが BLOB、CLOB、DBCLOB、LONG VARCHAR、または LONG VARCHARIC の列の長さ 0 のデータ値を最小限のスペースを使用して保管するよう指定します。NULL 値は、追加バイトを使用することなく、それぞれ保管されます。これをサポートする行形式は、各データ・タイプのバイト・カウントを識別し、更新の際にデータ・フラグメントの原因となる傾向があります。新しい行形式 (COMPRESS SYSTEM DEFAULT オプションによって列を指定する) を使用しても、列のシステム・デフォルト値をより効率的に保管できます。

DEACTIVATE

今後の更新のためのスペースを確保するように NULL 値を指定します。このスペースは、可変長の列のためには確保されません。使用される行形式は、各データ・タイプのバイト・カウントを識別します。またそのようにすると、列のシステム・デフォルト値のストレージを効率的にサポートしません。列が COMPRESS SYSTEM DEFAULT 属性を既に有していると、警告が出されます (SQLSTATE 01648)。

更新操作により、既存の行を新しい行形式に変更します。既存の行の更新操作のパフォーマンスを向上させるには、オフラインでの表再編成をお勧めします。

規則:

- 表に対して定義されたユニーク・キー制約または主キー制約は、区分化キー (存在する場合) のスーパーセットである必要があります (SQLSTATE 42997)。
- 主キーまたはユニーク・キーは、ディメンションのサブセットにはなりません (SQLSTATE 429BE)。
- 1 つの列の参照は、1 つの ALTER TABLE ステートメント内の 1 つの ADD または ALTER COLUMN 文節でのみ可能です (SQLSTATE 42711)。

ALTER TABLE

- 表にマテリアライズ照会表があり、その表に従属している場合、列の長さを変更することはできません (SQLSTATE 42997)。
- 生成された列を追加する前に、SET INTEGRITY ステートメントを使用して、表をチェック・ペンディング状態に設定しなければなりません (SQLSTATE 55019)。

注:

- 表をマテリアライズ照会表に変更すると、この表はチェック・ペンディング状態になります。表が REFRESH IMMEDIATE として定義されている場合、この表はチェック・ペンディング状態から出されなければ、全選択で参照されている表で INSERT、DELETE、または UPDATE コマンドを発行することはできません。IMMEDIATE CHECKED オプションを指定して REFRESH TABLE または SET INTEGRITY を使用すると、表のチェック・ペンディング状態を解除し、全選択に基づいて表内のデータを完全にリフレッシュできます。表にあるデータが完全に全選択の結果を反映する場合、SET INTEGRITY の IMMEDIATE UNCHECKED オプションを使用して、表のチェック・ペンディング状態を解除できます。
- 表を変更して REFRESH IMMEDIATE マテリアライズ照会表にすると、全選択により参照される表に対して INSERT、DELETE、または UPDATE を使用するパッケージはどれも無効になります。
- 表をマテリアライズ照会表から正規表 (DEFINITION ONLY) に変更すると、表に関連するパッケージはどれも無効になります。
- 据え置かれたマテリアライズ照会表がステージング表に関連付けられる場合、マテリアライズ照会表が正規表に変更されると、ステージング表はドロップされます。
- ADD 列文節は、他のいずれの文節よりも先に処理されます。他の文節は、指定された順序で処理されます。
- ALTER TABLE によって追加される列は、表の既存のビューに自動的に追加されることはありません。
- ユニーク・キー制約または主キー制約に関して索引が自動的に作成される場合、データベース・マネージャーは、指定された制約名を表のスキーマ名と一致するスキーマ名を伴う索引名として使用することを試みます。この名前が既存の索引名と一致する場合、または制約の名前が指定されなかった場合、索引は SYSIBM スキーマに作成され、"SQL" とタイム・スタンプに基づいて生成される 15 個の数字からなるシステム生成の名前が付けられます。
- 表 T での DELETE 操作に関係する可能性のある表は、T に連結削除 されている、と言われます。つまり、ある表が T の従属表であるか、または T からの削除のカスケード先の表の従属表である場合、この表は T に対して連結削除されることになります。
- パッケージの中に挿入 (更新/削除) の使用があるといわれるのは、パッケージ内のステートメントによって直接に、あるいは、そのいずれかのステートメントの代わりにパッケージによって実行される制約やトリガーによって間接的に、レコードが T に挿入 (更新または削除) される場合です。同様に、パッケージの中に更新の使用があるといわれるのは、パッケージ内のステートメントによって直接に、あるいは、その

いずれかのステートメントの代わりにパッケージによって実行される制約やトリガーによって間接的に、列が変更される場合です。

- 連合システムでは、透過 DDL を使用して、DB2 を通じて作成されたりリモート表は変更できます。ただし、可能な変更に関して、透過 DDL には以下のようないくつかの制限があります。
 - 新規列および主キーのみがリモート表に追加可能です。
 - リモート表内の既存の主キーは、変更またはドロップできません。
 - リモート表の変更を行うと、リモート表に関連したニックネームに從属するパッケージはいずれも無効になります。
 - リモート・データ・ソースは、ALTER TABLE ステートメントを通じて要求された変更をサポートする必要があります。データ・ソースの、サポートしていない要求への応答方法によって、エラーが返されるか、または要求が無視される可能性があります。
 - 元から作成されていたリモート表 (透過 DDL を使用して、DB2 で作成されていない表) を変更しようとする、エラーが返されます。
- 主キー、ユニーク・キー、または外部キーに対する変更は、パッケージ、索引、およびその他の外部キーに以下の影響を与えます。
 - 主キーまたはユニーク・キーが追加された場合、
 - パッケージ、索引、外部キー、または既存のユニーク・キーには影響を与えません。(主キーまたはユニーク・キーが、前のバージョンで作成された既存のユニーク索引を使用しており、固有性の据え置きをサポートするように変換されていない場合、索引は変換され、関連した表の更新を行うパッケージは無効になります。)
 - 主キーまたはユニーク・キーがドロップされた場合、
 - 制約に関してその索引が自動的に作成されていた場合には、その索引はドロップされます。索引に從属しているパッケージはすべて無効になります。
 - 索引が制約に関して固有であるように変換されており、現在システムが索引に関して固有であることを必要としない場合、索引は非固有に戻されます。索引に從属しているパッケージはすべて無効になります。
 - 索引が制約のために使用された既存のユニーク索引だった場合、索引はシステムが必要としないことを示すよう設定されます。パッケージに影響はありません。
 - すべての從属外部キーはドロップされます。次の項目に示すように、各從属外部キーごとに、さらにアクションが取られます。
 - 外部キーが追加される、ドロップされる、または NOT ENFORCED から ENFORCED に (または ENFORCED から NOT ENFORCED に) 変更される場合:
 - オブジェクト表に対して挿入を行うパッケージは、すべて無効になります。
 - 外部キー内の少なくとも 1 つの列に対して更新を行うパッケージは、すべて無効になります。

ALTER TABLE

- 親表に削除使用のあるパッケージはすべて無効になります。
- 親キーの少なくとも 1 つの列に対して更新使用の指定があるパッケージは、すべて無効になります。
- 外部キーが、ENABLE QUERY OPTIMIZATION から DISABLE QUERY OPTIMIZATION に変更される場合:
 - 最適化のための制約と従属関係にあるパッケージすべては、無効です。
- 表に列を追加すると、変更された表に対して挿入を行うパッケージはすべて無効になります。追加された列が、表内の最初のユーザー定義構造タイプ列である場合、変更された表で DELETE 行おうとするパッケージは無効になります。
- チェック・ペンディング状態ではない既存の表に対してチェック制約または参照制約を追加するか、またはチェック・ペンディング状態にない既存の表の既存のチェック制約か参照制約を NOT ENFORCED から ENFORCED に変更すると、その表の既存の行は、制約に関して直ちに評価されます。検証に失敗すると、エラー (SQLSTATE 23512) になります。表がチェック・ペンディング状態の場合は、チェック制約または参照制約を追加しても、または制約を NOT ENFORCED から ENFORCED に変更しても、制約が直ちに適用されるわけではありません。その場合には、チェック・ペンディング操作で使用された制約タイプ・フラグのうち対応するものが更新されます。制約の適用を開始するには、SET INTEGRITY ステートメントを発行します。
- チェック制約の追加、変更、またはドロップを行うと、対象の表に対する挿入、制約に関係している少なくとも 1 つの列に対する更新、またはパフォーマンスを向上させるための制約の選択使用のいずれかを含むすべてのパッケージが無効になります。
- 区分化キーを追加すると、区分化キーの少なくとも 1 つの列に対して更新を行うパッケージは、すべて無効になります。
- デフォルトで主キーの最初の列を使用して定義された区分化キーは、主キーのドロップや異なる主キーの追加によっては影響を受けません。
- 列の長さを変更して長くすると、変更された列を含む表を (参照制約またはトリガーによって直接または間接的に) 参照するパッケージはすべて無効になります。
- 列の長さを変更して長くすると、表に従属するビュー (型付きビューを除く) が再生成されます。ビューの再生成時にエラーが生じると、エラーが戻されます (SQLSTATE 56098)。表に従属する型付きビューは、作動不能としてマークされます。
- 列の長さを長く変更すると、トリガーを含むステートメントを準備中またはバインド中に、トリガー処理でエラー (SQLSTATE 54010) が発生する可能性があります。このことは、遷移変数および遷移表列の長さの合計に基づく行の長さが長すぎる場合に生じます。このようなトリガーがドロップされると、それ以降にトリガーを作成しようとしてもエラー (SQLSTATE 54040) となります。

- それぞれ 4000 および 2000 より大きい数値に変更された VARCHAR および VARGRAPHIC 列は、SYSFUN スキーマの関数での入力パラメーターとして使用しないでください (SQLSTATE 22001)。
- 構造タイプ列を変更してインライン長を長くすると、参照制約またはトリガーによって直接または間接的に表を参照するパッケージはすべて無効になります。
- 構造タイプ列を変更してインライン長を長くすると、表に従属するビューは再生成されます。
- 表の LOCKSIZE を変更すると、変更された表に従属するすべてのパッケージは無効になります。
- ACTIVATE NOT LOGGED INITIALLY 文節は、FILE LINK CONTROL 属性のある DATALINK 列を表に追加するときには使えません (SQLSTATE 42613)。
- VOLATILE または NOT VOLATILE CARDINALITY を変更すると、変更された表に従属するすべてのパッケージは無効になります。
- 複製を行うユーザーは、VARCHAR 列の長さが長くなるときに、特に注意する必要があります。アプリケーション表と関連付けられた変更データ表は、すでに DB2 行サイズの限界近くに設定されている可能性があります。変更データ表をアプリケーション表よりも前に変更するか、これら 2 つを同じ作業単位内で変更するようにして、両方の表で変更が完了できるようにしてください。コピーについても考慮すべき点があります。これも、行サイズの限界近くに設定されていたり、既存の列の長さを長くする機能のないプラットフォームに存在している可能性があります。

VARCHAR 列の長さを長くしたログ・レコードをキャプチャー・プログラムが処理する前に、変更データ表を変更していなければ、キャプチャー・プログラムが失敗する場合があります。コピーを保持しているサブスクリプションを実行する前に、VARCHAR 列が含まれるコピーを変更していなければ、そのサブスクリプションは失敗する可能性があります。

- 互換性

- 以前のバージョンの DB2 との互換性:
 - 以下についての ADD キーワードはオプションです。
 - 名前のない PRIMARY KEY 制約
 - 名前のない参照制約
 - FOREIGN KEY 句の後に名前を指定した参照制約
 - CONSTRAINT キーワードは、参照文節を定義する *column-definition* から省略できます。
 - *constraint-name* (制約名) を FOREIGN KEY に続けて (CONSTRAINT キーワードなし) 指定することができます。
 - SET MATERIALIZED QUERY AS の代わりに SET SUMMARY AS を指定できます。
- 以前のバージョンの DB2 との互換性と整合性:

ALTER TABLE

- コンマは、*identity-alteration* 文節では複数のオプションを分離するのに使用できます。
- 以下の構文もサポートされています。
 - NOMINVALUE、NOMAXVALUE、NOCYCLE、NOCACHE、および NOORDER

例:

例 1: 1 文字の長さの RATING という名前の新しい列を、DEPARTMENT 表に追加します。

```
ALTER TABLE DEPARTMENT
ADD RATING CHAR(1)
```

例 2: SITE_NOTES という名前の新しい列を PROJECT 表に追加します。SITE_NOTES は、最大 1000 文字の長さの可変長列として作成します。この列の値には関連する文字セットがなく、変換されません。

```
ALTER TABLE PROJECT
ADD SITE_NOTES VARCHAR(1000) FOR BIT DATA
```

例 3: 以下の列が定義された EQUIPMENT という表が存在するものと想定します。

列名	データ・タイプ
EQUIP_NO	INT
EQUIP_DESC	VARCHAR(50)
LOCATION	VARCHAR(50)
EQUIP_OWNER	CHAR(3)

EQUIPMENT 表に、所有者 (EQUIP_OWNER) は DEPARTMENT 表に存在する部門番号 (DEPTNO) でなければならない、という参照制約を追加します。DEPTNO は、DEPARTMENT 表の主キーです。DEPARTMENT 表からある部門を削除する場合は、その部門の所有するすべての備品の所有者 (EQUIP_OWNER) の値を割り当て解除する必要があります (つまり NULL 値に設定する必要があります)。制約の名前は、DEPTQUIP です。

```
ALTER TABLE EQUIPMENT
ADD CONSTRAINT DEPTQUIP
FOREIGN KEY (EQUIP_OWNER)
REFERENCES DEPARTMENT
ON DELETE SET NULL
```

さらに、備品レコードに関係した数量を記録できるようにするため、追加の列が必要になります。特に指定されない限り、EQUIP_QTY 列には値 1 を入れます。NULL 値にしてはなりません。

```
ALTER TABLE EQUIPMENT
ADD COLUMN EQUIP_QTY
SMALLINT NOT NULL DEFAULT 1
```

例 4: 表 EMPLOYEE を変更します。各従業員の給与と歩合の合計が \$30,000 を超えていなければならない、という定義済みの REVENUE という名前のチェック制約を追加します。

```
ALTER TABLE EMPLOYEE
  ADD CONSTRAINT REVENUE
  CHECK (SALARY + COMM > 30000)
```

例 5: 表 EMPLOYEE を変更します。前に定義した制約 REVENUE をドロップします。

```
ALTER TABLE EMPLOYEE
  DROP CONSTRAINT REVENUE
```

例 6: SQL の変更内容をデフォルトのフォーマットでログに記録するように表を変更します。

```
ALTER TABLE SALARY1
  DATA CAPTURE NONE
```

例 7: SQL の変更内容を拡張フォーマットでログに記録するように表を変更します。

```
ALTER TABLE SALARY2
  DATA CAPTURE CHANGES
```

例 8: EMPLOYEE 表を変更して、デフォルト値を指定して 4 つの新しい列を追加します。

```
ALTER TABLE EMPLOYEE
  ADD COLUMN HEIGHT MEASURE DEFAULT MEASURE(1)
  ADD COLUMN BIRTHDAY BIRTHDATE DEFAULT DATE('01-01-1850')
  ADD COLUMN FLAGS BLOB(1M) DEFAULT BLOB(X'01')
  ADD COLUMN PHOTO PICTURE DEFAULT BLOB(X'00')
```

デフォルト値の指定時に、これらのデフォルト値はさまざまな関数名を使用します。MEASURE は INTEGER に基づく特殊タイプなので、MEASURE 関数を使用しています。ちなみに、HEIGHT 列のデフォルト値は、関数を使用しなくても指定することができたはずですが、MEASURE のソース・タイプは、BLOB または日時データ・タイプではないからです。BIRTHDATE は DATE に基づく特殊タイプなので、DATE 関数を使用しています（この場合、BIRTHDATE は使用できません）。FLAGS 列と PHOTO 列では、PHOTO が特殊名であるにもかかわらず、BLOB 関数を使用してデフォルト値が指定されています。BIRTHDAY、FLAGS、および PHOTO 列のデフォルト値を指定するためには、関数を使用する必要があります。タイプが、BLOB や日時データ・タイプのソースに基づく BLOB や特殊タイプだからです。

例 9: 以下の列のある CUSTOMERS という表が定義されます。

列名	データ・タイプ
BRANCH_NO	SMALLINT
CUSTOMER_NO	DECIMAL(7)
CUSTOMER_NAME	VARCHAR(50)

ALTER TABLE

この表では、主キーは `BRANCH_NO` 列と `CUSTOMER_NO` 列からなります。表を区分化するには、表に対して区分化キーを作成する必要があります。表は単一ノードのデータベース・パーティション・グループにある表スペースに定義する必要があります。主キーは、区分化列のスーパーセットである必要があり、主キーの少なくとも 1 つの列が区分化キーとして使用されている必要があります。以下のようにして、`BRANCH_NO` を区分化キーとして定義します。

```
ALTER TABLE CUSTOMERS
  ADD PARTITIONING KEY (BRANCH_NO)
```

例 10: リモート表 `EMPLOYEE` が、連合システムに透過 DDL を使用して作成されました。リモート表 `EMPLOYEE` を変更して、列 `PHONE_NO` および `WORK_DEPT` を追加します。また、主キーを既存の列 `EMP_NO` および新規列 `WORK_DEPT` に追加します。

```
ALTER TABLE EMPLOYEE
  ADD COLUMN PHONE_NO CHAR(4) NOT NULL
  ADD COLUMN WORK_DEPT CHAR(3)
  ADD PRIMARY KEY (EMP_NO, WORK_DEPT)
```

関連タスク:

- 450 ページの『透過 DDL を使用したリモート表の作成および変更』

関連資料:

- *SQL* リファレンス 第 2 巻 の『ALTER TYPE (構造化) ステートメント』
- 1484 ページの『CREATE TABLE』

関連サンプル:

- 『dbrecov.sql -- How to recover a database (C)』
- 『tbconstr.sql -- How to create, use, and drop constraints (C)』
- 『dbrecov.sqlC -- How to recover a database (C++)』
- 『dtstruct.sqlC -- Create, use, drop a hierarchy of structured types and typed tables (C++)』
- 『tbconstr.sqlC -- How to create, use, and drop constraints (C++)』
- 『TbGenCol.java -- How to use generated columns (JDBC)』

ALTER TABLESPACE

ALTER TABLESPACE ステートメントは、以下の方法で既存の表スペースを変更する場合に使用されます。

- DMS 表スペース (つまり `MANAGED BY DATABASE` オプションによって作成される表スペース) にコンテナを追加する、または DMS 表スペースからコンテナをドロップする。
- DMS 表スペースのコンテナのサイズを変更する。

- コンテナのないパーティション上の SMS 表スペースにコンテナを追加する。
- 表スペースの PREFETCHSIZE 設定値を変更する。
- 表スペースの表に対して使用する BUFFERPOOL を変更する。
- 表スペースの OVERHEAD 設定値を変更する。
- 表スペースの TRANSFERRATE 設定値を変更する。

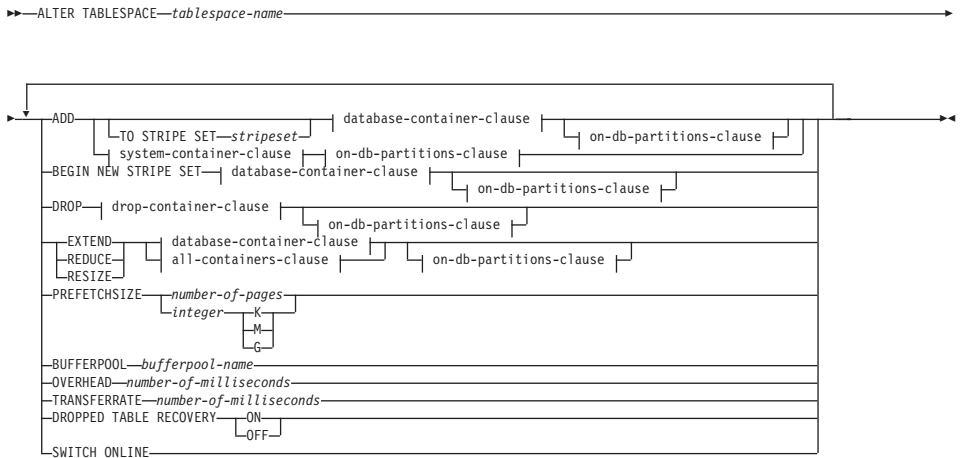
呼び出し:

このステートメントは、アプリケーション・プログラムに組み込むか、あるいは対話式に発行することができます。 DYNAMICRULES の実行動作がパッケージに効力を持つ場合にのみ、動的に準備できる実行可能ステートメントです (SQLSTATE 42509)。

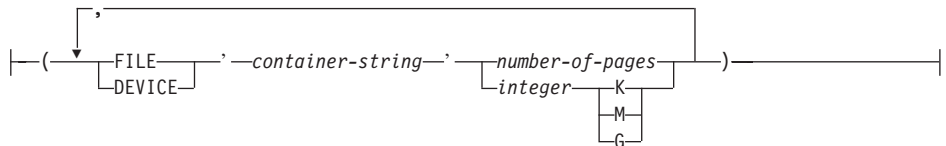
許可:

このステートメントの許可 ID には、SYSCTRL 権限または SYSADM 権限がなければなりません。

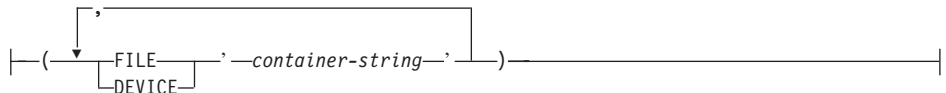
構文:



database-container-clause:



drop-container-clause:

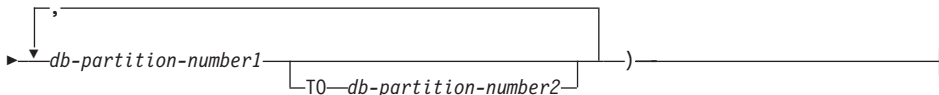


ALTER TABLESPACE

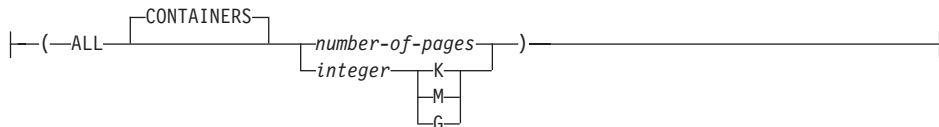
system-container-clause:



on-db-partitions-clause:



all-containers-clause:



説明:

tablespace-name

表スペースの名前を指定します。これは、1 つの部分からなる名前です。これは、長形式 SQL ID です (通常 ID または区切り ID のいずれか)。

ADD

表スペースに 1 つか複数の新しいコンテナを追加するように指定します。

TO STRIPE SET *stripeset*

1 つか複数の新しいコンテナを表スペースに追加し、指定されたストライプ・セットに配置するように指定します。

BEGIN NEW STRIPE SET

表スペースに新しいストライプ・セットを作成するように、およびその新しいストライプ・セットに 1 つか複数のコンテナを追加するように指定します。ADD オプションを使用して後に追加されるコンテナは、TO STRIPE SET が指定されない場合にはこの新しいストライプ・セットに追加されます。

DROP

1 つか複数のコンテナを表スペースからドロップするように指定します。

EXTEND

既存のコンテナのサイズを増やすように指示します。指定するサイズは、既存のコンテナに追加されるサイズです。all-containers-clause が指定されると、表スペースにあるすべてのコンテナがこのサイズで拡張されます。

REDUCE

既存のコンテナのサイズを減らすように指示します。指定するサイズは、既存のコンテナから減らすサイズです。 *all-containers-clause* が指定されると、表スペースにあるすべてのコンテナがこのサイズだけ縮小されます。

RESIZE

既存のコンテナのサイズを変更するよう指定します。指定されるサイズが、コンテナの新しいサイズになります。 *all-containers-clause* が指定されると、表スペースにあるすべてのコンテナがこのサイズに変更されます。この操作が複数のコンテナに影響を与える場合には、そうしたコンテナすべてのサイズは増やすか減らすかのどちらかにする必要があります。一部を増やし、その他を減らすことはできません (SQLSTATE 429BC)。

database-container-clause

DMS 表スペースに 1 つまたは複数のコンテナを追加します。表スペースは、すでにアプリケーション・サーバーに存在する DMS 表スペースを指定するものでなければなりません。

drop-container-clause

1 つまたは複数のコンテナを DMS 表スペースからドロップします。表スペースは、すでにアプリケーション・サーバーに存在する DMS 表スペースを指定するものでなければなりません。

system-container-clause

指定のパーティションにある SMS 表スペースに、1 つまたは複数のコンテナを追加します。表スペースは、すでにアプリケーション・サーバーに存在する SMS 表スペースを指定するものでなければなりません。表スペースに対して指定するパーティションにコンテナがあってはなりません (SQLSTATE 42921)。

on-db-partitions-clause

1 つまたは複数のパーティションに対応するコンテナ操作を指定します。

all-containers-clause

DMS 表スペースにあるコンテナすべてを拡張、縮小またはサイズ変更します。表スペースは、すでにアプリケーション・サーバーに存在する DMS 表スペースを指定するものでなければなりません。

PREFETCHSIZE *number-of-pages*

データのプリフェッチの実行中に、表スペースから読み取られる PAGESIZE ページの数を指定します。このプリフェッチ・サイズ値は、後に K (K バイトの場合)、M (M バイトの場合)、または G (G バイトの場合) を付けた整数値としても指定できます。このように指定した場合、ページ・サイズで分割されたバイト数のフロアが、プリフェッチ・サイズのページ値の数を判別するために使用されます。プリフェッチでは、照会に必要なデータがその照会で参照される前に読み取られるため、照会では入出力の実行を待たずに済みます。

ALTER TABLESPACE

BUFFERPOOL *bufferpool-name*

この表スペースの表に対して使用するバッファ・プールの名前を指定します。バッファ・プールは、現在データベースに存在する必要があります (SQLSTATE 42704)。 バッファ・プールに対して、この表スペースのデータベース・パーティション・グループを定義する必要があります (SQLSTATE 42735)。

OVERHEAD *number-of-milliseconds*

入出力制御装置のオーバーヘッドとディスク・シーク待ち時間をミリ秒単位で指定する数値リテラルです (整数、10 進数、または浮動小数点数)。この数値がすべてのコンテナで同一でない場合、それは表スペースに属するすべてのコンテナの平均でなければなりません。この値は、照会の最適化の過程で入出力コストを判別するのに使用されます。

TRANSFERRATE *number-of-milliseconds*

1 ページ (4K または 8K) をメモリーに読み込むための時間をミリ秒単位で指定する数値リテラルです (整数、10 進数、または浮動小数点数)。この数値がすべてのコンテナで同一でない場合、それは表スペースに属するすべてのコンテナの平均でなければなりません。この値は、照会の最適化の過程で入出力コストを判別するのに使用されます。

DROPPED TABLE RECOVERY

指定された表スペースからドロップされた表は、ROLLFORWARD コマンドの RECOVER DROPPED TABLE ON オプションを使用して回復させることができます。

SWITCH ONLINE

OFFLINE 状態の表スペースは、コンテナがアクセス可能であれば、オンラインになります。コンテナがアクセス可能でなければ、エラーが戻されます (SQLSTATE 57048)。

注:

- **互換性**

バージョン 8 より前のバージョンのキーワードとの互換性:

- DBPARTITIONNUM を NODE に置換できます。
- DBPARTITIONNUMS を NODES に置換できます。

- 各コンテナ定義には、53 バイトに加えて、コンテナ名を保管するのに必要なバイト数が必要です。表スペースのすべてのコンテナ名を結合した長さは、20 480 バイトを超えることはできません (SQLSTATE 54034)。
- デフォルトのコンテナ操作は、ALTER TABLESPACE ステートメントで指定されるコンテナ操作ですが、この操作は特定のデータベース・パーティションに明示的に向けられません。こうしたコンテナ操作は、ステートメントにリストされていない任意のデータベース・パーティションに向けられます。デフォルトのコンテナ操

作がどのデータベース・パーティションにも向けられない場合には、コンテナ操作ではすべてのデータベース・パーティションに明示的に言及するので、警告が出されます (SQLSTATE 1758W)。

- スペースが表スペースに追加または、表スペースから削除され、トランザクションがコミットされると、表スペースの内容はコンテナ間でバランスの再調整がなされるかもしれませんが。バランス再調整中も、表スペースへのアクセスは制限されません。
- 表スペースが **OFFLINE** 状態で、コンテナがアクセス可能である場合、すべてのアプリケーションを切断してから、もう一度データベースへ接続すれば、表スペースは **OFFLINE** 状態から脱することができます。別の方法として、**SWITCH ONLINE** オプションを使用すると、残りのデータベースは稼働状態で使用中のまま、表スペースは **OFFLINE** から脱する (稼働状態になる) ことができます。
- 表スペースに複数のコンテナを追加する場合は、バランスの再調整のコストが一度だけで済むように、これらのコンテナを同じステートメントで追加することをお勧めします。単一トランザクションで別々の **ALTER TABLESPACE** ステートメントを使用して、同じ表スペースにコンテナを追加するとエラーになります (SQLSTATE 55041)。
- 存在しないコンテナについて拡張、縮小、またはサイズ変更、またはドロップをしようとする、エラーが発生します (SQLSTATE 428B2)。
- コンテナを拡張、軽減またはサイズ変更する場合、このコンテナ・タイプは、コンテナが作成されたときに使用されたタイプと適合しなければなりません (SQLSTATE 428B2)。
- 1 つのトランザクションで、同じ表スペースに対して別個の **ALTER TABLESPACE** ステートメントを使用して、複数のコンテナ・サイズを変更しようとする、エラーが発生します (SQLSTATE 55041)。
- パーティション・データベースで、複数のデータベース・パーティションが同じ物理ノードに存在する場合、このようなデータベース・パーティションに同じ装置または特定のパスを指定することはできません (SQLSTATE 42730)。この環境の場合、それぞれのデータベース・パーティションごとに固有の *container-string* を指定するか、または相対パス名を使用してください。
- 表スペース定義はトランザクションであり、表スペース定義に対する変更はコミット時にカタログ表に反映されますが、新しい定義のバッファ・プールは、データベースの次回始動時まで使用することはできません。 **ALTER TABLESPACE** ステートメントが出されたときに使用中のバッファ・プールは、それまで引き続き使用されます。

規則:

- **BEGIN NEW STRIPE SET** 文節は、**ADD**、**DROP**、**EXTEND**、**REDUCE**、および **RESIZE** 文節が別のパーティションに向けられない限りは、それらの文節と同じステートメントでは指定できません (SQLSTATE 429BC)。
- **TO STRIPE SET** 文節を使用して指定されたストライプ・セットは、変更される表スペースの有効な範囲内になければなりません (SQLSTATE 42615)。

ALTER TABLESPACE

- 表スペースにスペースを追加、または表スペースからスペースを削除する場合、以下の規則に従います。
 - EXTEND および RESIZE は、各コンテナのサイズを拡張する、同じステートメントで使用できます (SQLSTATE 429BC)。
 - REDUCE および RESIZE は、各コンテナのサイズを軽減する、同じステートメントで使用できます (SQLSTATE 429BC)。
 - EXTEND および REDUCE は、異なるパーティションに向けられない限りは、同じステートメントでは使用できません (SQLSTATE 429BC)。
 - ADD は、REDUCE または DROP が異なるパーティションに向けられない限りは、それらと共に同じステートメントでは使用できません (SQLSTATE 429BC)。
 - DROP は、EXTEND または ADD が異なるパーティションに向けられない限りは、それらと共に同じステートメントでは使用できません (SQLSTATE 429BC)。

例:

例 1: PAYROLL 表スペースに装置を追加します。

```
ALTER TABLESPACE PAYROLL
ADD (DEVICE '/dev/rhdisk9' 10000)
```

例 2: ACCOUNTING 表スペースのプリフェッチ・サイズと入出力オーバーヘッドを変更します。

```
ALTER TABLESPACE ACCOUNTING
PREFETCHSIZE 64
OVERHEAD 19.3
```

例 3: 表スペース TS1 を作成した後、コンテナをサイズ変更して、すべてのコンテナのサイズが 2000 ページになるようにします (3 つの異なる ALTER TABLESPACES を使用して、サイズを変更)。

```
CREATE TABLESPACE TS1
MANAGED BY DATABASE
USING (FILE '/conts/cont0' 1000,
        DEVICE '/dev/rcont1' 500,
        FILE 'cont2' 700)
ALTER TABLESPACE TS1
RESIZE (FILE '/conts/cont0' 2000,
        DEVICE '/dev/rcont1' 2000,
        FILE 'cont2' 2000)
```

または

```
ALTER TABLESPACE TS1
RESIZE (ALL 2000)
```

または

```
ALTER TABLESPACE TS1
  EXTEND (FILE '/conts/cont0' 1000,
         DEVICE '/dev/rcont1' 1500,
         FILE 'cont2' 1300)
```

例 4: DATA_TS 表スペースにあるすべてのコンテナを 1000 ページだけ拡張します。

```
ALTER TABLESPACE DATA_TS
  EXTEND (ALL 1000)
```

例 5: INDEX_TS 表スペースにあるすべてのコンテナのサイズを 100 メガバイト (MB) に変更します。

```
ALTER TABLESPACE INDEX_TS
  RESIZE (ALL 100 M)
```

例 6: 3 つの新規コンテナを追加します。1 番目のコンテナを拡張し、2 番目をサイズ変更します。

```
ALTER TABLESPACE TS0
  ADD (FILE 'cont2' 2000, FILE 'cont3' 2000)
  ADD (FILE 'cont4' 2000)
  EXTEND (FILE 'cont0' 100)
  RESIZE (FILE 'cont1' 3000)
```

例 7: 表スペース TSO がパーティション 0、1 および 2 に存在します。新規コンテナをデータベース・パーティション 0 に追加します。データベース・パーティション 1 上のすべてのコンテナを拡張します。明示的に指定されたデータベース・パーティション (つまり、データベース・パーティション 0 および 1) 以外の、すべてのデータベース・パーティション上のコンテナをサイズ変更します。

```
ALTER TABLESPACE TS0
  ADD (FILE 'A' 200) ON DBPARTITIONNUM (0)
  EXTEND (ALL 200) ON DBPARTITIONNUM (1)
  RESIZE (FILE 'B' 500)
```

この例では RESIZE 文節がデフォルトのコンテナ文節で、データベース・パーティション 2 で実行されます。他の操作は明示的にデータベース・パーティション 0 および 1 に向けられるからです。しかしデータベース・パーティションが 2 つしかない場合、ステートメントは正常に実行されますが、デフォルトのコンテナが指定されたものを使用されていないことを示す警告が出されます (SQL1758W)。

関連資料:

- 1562 ページの『CREATE TABLESPACE』

CREATE TYPE (構造化)

CREATE TYPE ステートメントは、ユーザー定義の構造タイプを定義します。ユーザー定義構造タイプには、属性を含めないこともできますし、複数の属性も含めることもで

CREATE TYPE (構造化)

きます。構造タイプには、スーパータイプからの属性を継承するサブタイプを指定することができます。ステートメントの実行が正常に完了すると、属性値の検索と更新のためのメソッドが生成されます。また、このステートメントの実行が正常に完了すると、列内で使用する構造タイプのインスタンスを作成する関数と、該当の参照タイプとその表示タイプとをキャストする関数、およびその参照タイプ上の比較演算子 (=、<>、<、<=、>、および >=) をサポートする関数も生成されます。

また、CREATE TYPE ステートメントは、ユーザー定義構造タイプと一緒に使用されるユーザー定義メソッドの任意のメソッド仕様も定義します。

呼び出し:

このステートメントはアプリケーション・プログラムに組み込むことができ、また動的 SQL ステートメントを使用して出すことができます。DYNAMICRULES の実行動作がパッケージに効力を持つ場合にのみ、動的に準備できる実行可能ステートメントです (SQLSTATE 42509)。

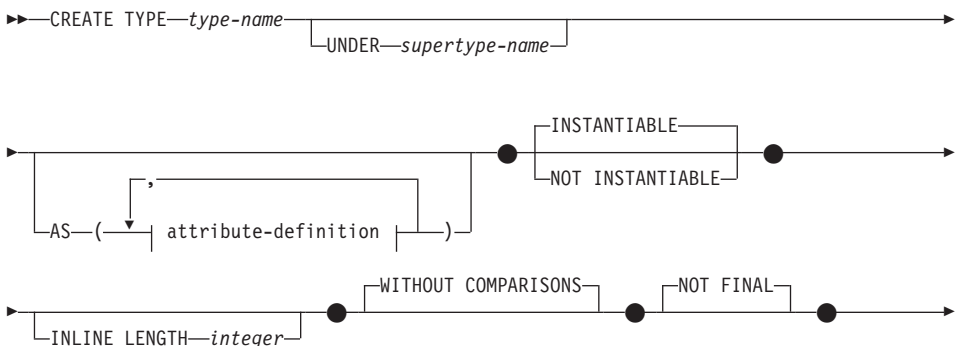
許可:

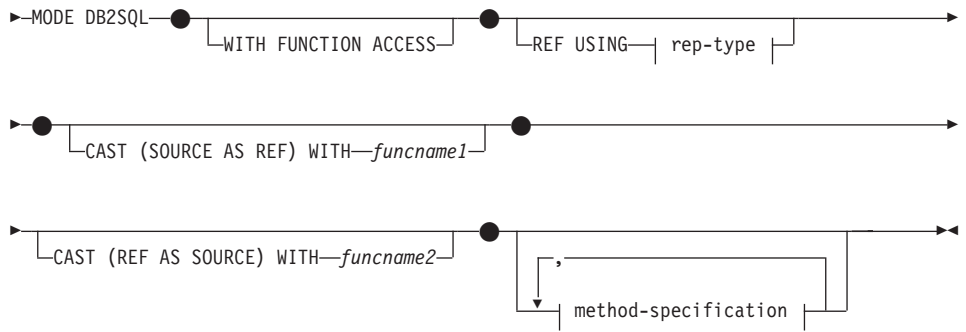
このステートメントの許可 ID には、以下の特権が少なくとも 1 つ含まれている必要があります。

- SYSADM または DBADM 権限
- データベースに対する IMPLICIT_SCHEMA 権限 (このタイプのスキーマ名が既存のスキーマを指していない場合)
- スキーマに対する CREATEIN 特権 (このタイプのスキーマ名が既存のスキーマを指している場合)

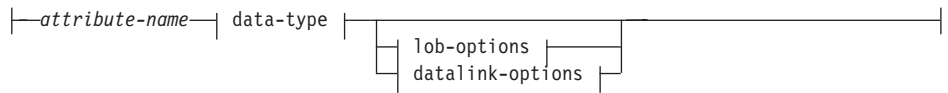
UNDER が指定されていて、このステートメントの許可 ID がタイプ階層のルート・タイプの定義者と同じではない場合には、SYSADM または DBADM 権限が必要です。

構文:

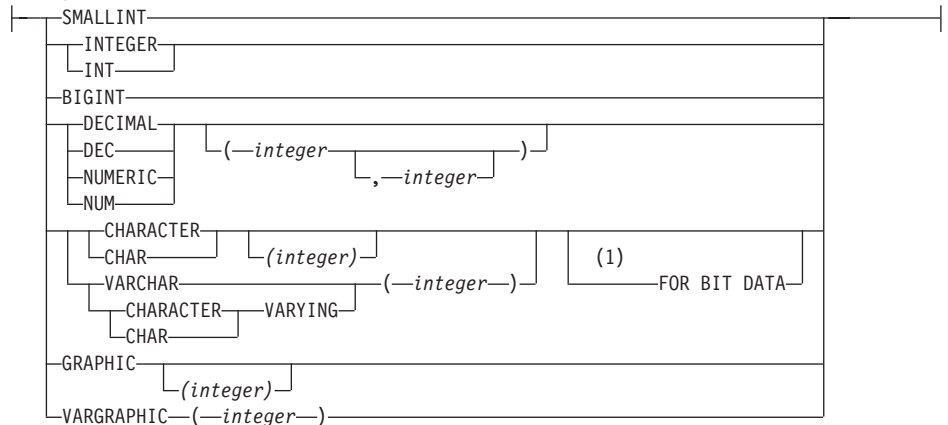




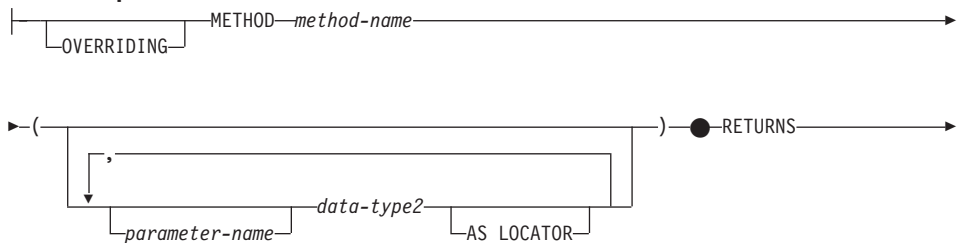
attribute-definition:



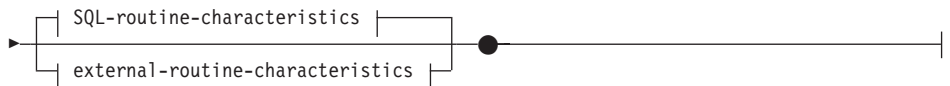
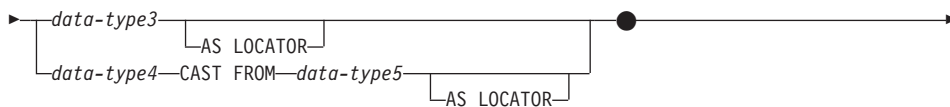
rep-type:



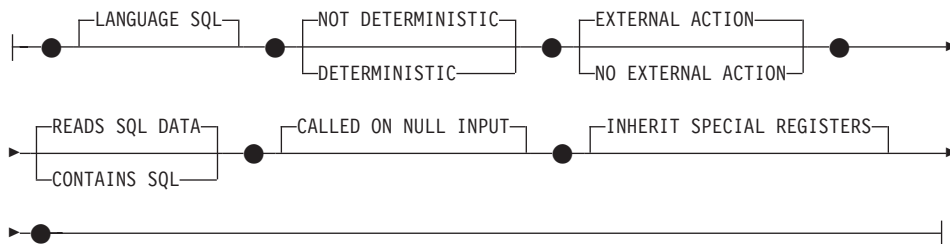
method-specification:



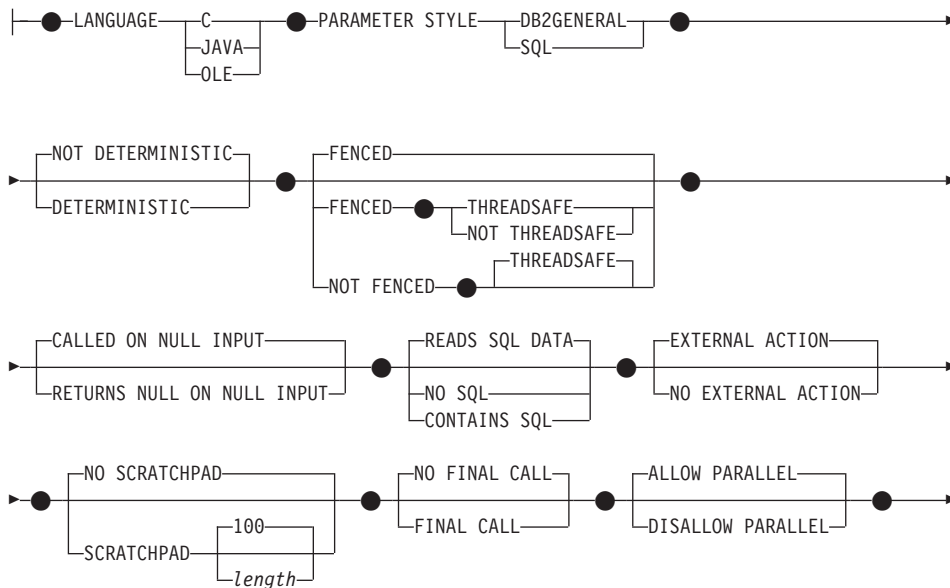
CREATE TYPE (構造化)



SQL-routine-characteristics:



external-routine-characteristics:



**注:**

- 1 FOR BIT DATA 文節とその後に続く他の列制約とは、任意の順序で指定できません。

説明:*type-name*

タイプの名前を指定します。名前 (暗黙または明示の修飾子を含む) は、カタログに既に記述されているその他のタイプ (組み込みタイプ、構造タイプ、特殊タイプを含む) と同じであってはなりません。非修飾名は、組み込みデータ・タイプ名と同一のものまたは BOOLEAN であってはなりません (SQLSTATE 42918)。動的 SQL ステートメントでは、CURRENT SCHEMA 特殊レジスタが、修飾子のないオブジェクト名の修飾子として使用されます。静的 SQL ステートメントでは、QUALIFIER プリコンパイル/バインド・オプションにより、修飾子のないオブジェクト名の修飾子が暗黙指定されます。

スキーマ名 (明示指定または暗黙指定) は、8 バイト以下でなければなりません (SQLSTATE 42622)。

述部のキーワードとして使用されるいくつかの名前は、システム使用に予約されており、*type-name* として使用することはできません (SQLSTATE 42939)。それらの名前は、SOME、ANY、ALL、NOT、AND、OR、BETWEEN、NULL、LIKE、EXISTS、IN、UNIQUE、OVERLAPS、SIMILAR、MATCH、および比較演算子です。

2 つの部分からなる *type-name* を指定する場合、スキーマ名を "SYS" で始めることはできません。違反すると、エラー (SQLSTATE 42939) になります。

UNDER *supertype-name*

この構造タイプが指定した *supertype-name* のサブタイプであることを指定します。*supertype-name* は既存の構造タイプを指定する必要があります (SQLSTATE 42704)。*supertype-name* がスキーマ名なしで指定される場合、SQL パス上でスキーマを検索することにより、タイプは解決されます。構造タイプには、上位タイプの属性すべてと、それに続く *attribute-definition* の追加属性が含まれます。

attribute-definition

構造タイプの属性を定義します。

attribute-name

属性の名前です。この構造タイプのその他の属性または上位タイプと同じ *attribute-name* を付けることはできません (SQLSTATE 42711)。

述部のキーワードとして使用されるいくつかの名前は、システム使用に予約されており、*attribute-name* として使用することはできません (SQLSTATE

CREATE TYPE (構造化)

42939)。それらの名前は、SOME、ANY、ALL、NOT、AND、OR、BETWEEN、NULL、LIKE、EXISTS、IN、UNIQUE、OVERLAPS、SIMILAR、MATCH、および比較演算子です。

data-type

属性のデータ・タイプです。これは、『CREATE TABLE』でリストされているデータ・タイプの1つで、LONG VARCHAR、LONG VARGRAPHIC、またはLONG VARCHARやLONG VARGRAPHICに基づいた特殊タイプ以外のものです (SQLSTATE 42601)。このデータ・タイプは既存のデータ・タイプを指定する必要があります (SQLSTATE 42704)。*data-type* がスキーマ名なしで指定される場合、SQLパス上でスキーマを検索することにより、タイプは解決されます。『CREATE TABLE』に種々のデータ・タイプの説明が記載されています。属性データ・タイプが参照タイプである場合、参照するターゲット・タイプはこのステートメントに既に存在する構造タイプであるか、またはこのステートメントで作成されたものでなければなりません (SQLSTATE 42704)。

タイプ DATALINK の属性を使って定義された構造タイプは、型付き表または型付きビューのデータ・タイプとしてのみ有効に使用することができます (SQLSTATE 01641)。

実行時に、該当タイプのインスタンスが、同一タイプまたはそのサブタイプの別のインスタンスを直接または間接に取り込むことを許容するタイプ定義を防止するため、その属性のいずれかが、自身を直接または間接に使用する仕方タイプを定義することはできません (SQLSTATE 428EP)。

lob-options

LOB タイプと関連したオプション (あるいはLOBに基づく特殊タイプ) を指定します。*lob-options* の詳細については、『CREATE TABLE』を参照してください。

datalink-options

DATALINK タイプと関連したオプション (あるいはDATALINKタイプに基づく特殊タイプ) を指定します。*datalink-options* の詳細については、『CREATE TABLE』を参照してください。

DATALINK タイプまたはDATALINKに基づいている特殊タイプでオプションが指定されないと、LINKTYPE URL およびNO LINK CONTROL オプションがデフォルト値になることに注目してください。

INSTANTIABLE または NOT INSTANTIABLE

構造タイプのインスタンスを作成できるかどうかを指定します。インスタンス化不能な構造タイプとは、以下のような意味です。

- インスタンス化不能タイプには、コンストラクター関数が生成されない
- インスタンス化不能タイプは、表またはビューのタイプとして使用することができない (SQLSTATE 428DP)

- インスタンス化不能タイプは、列のタイプとして使用することができる (その列には、NULL 値またはインスタンス化可能なサブタイプのインスタンスだけを挿入することができる)

インスタンス化不能タイプのインスタンスを作成するには、インスタンス化可能サブタイプを作成する必要があります。NOT INSTANTIABLE を指定すると、新規のタイプのインスタンスを作成できなくなります。

INLINE LENGTH *integer*

このオプションは、表の行内の残りの値とインラインで保管する構造タイプ列のインスタンスの最大サイズ (バイト数) を指示します。指定したインライン長よりも長い構造タイプまたはそのサブタイプのインスタンスは、LOB 値が処理されるのと同様の方法で、基本表の行とは別に保管されます。

指定した INLINE LENGTH が、新たに作成したタイプのコンストラクター関数の結果サイズよりも小さく (32 バイトに、属性ごとに 10 バイトを加算したもの)、しかも 292 バイトより小さいと、エラーが生じます (SQLSTATE 429B2)。属性数には、タイプのスーパータイプから継承されたすべての属性が含まれることに注意してください。

タイプの INLINE LENGTH は、指定値またはデフォルト値のどちらであっても、構造タイプを使用する列のデフォルトのインライン長になります。このデフォルトは、CREATE TABLE 時にオーバーライドすることができます。

型付き表のタイプとして構造タイプを使用すると、INLINE LENGTH には何の意味もなくなります。

構造タイプのデフォルトの INLINE LENGTH はシステムによって計算されます。この後に示す公式では、以下のような用語を使います。

短い属性 (*short attribute*)

SMALLINT、INTEGER、BIGINT、REAL、DOUBLE、FLOAT、DATE、または TIME のデータ・タイプのいずれかをもつ属性を指します。さらに、これらのタイプに基づいた特殊タイプまたは参照タイプも含まれます。

短くない属性 (*non-short attribute*)

残りのデータ・タイプのいずれか、またはこれらのデータ・タイプに基づく特殊タイプの属性を指します。

システムは、以下のようにデフォルトのインライン長を計算します。

1. 以下のような公式を使って、短くない属性の追加スペース所要量を割り出します。

$$\text{space_for_non_short_attributes} = \text{SUM}(\text{attributelength} + n)$$

n は以下のように定義されます。

- ネストされた構造タイプの属性には 0 バイト

CREATE TYPE (構造化)

- 非 LOB 属性には 2 バイト
- LOB 属性には 9 バイト

attributelength は、表 268 に示すとおり、属性に指定されているデータ・タイプに基づく値です。

2. 以下のような公式を使って、デフォルトの合計インライン長を計算します。

$$\text{default_length}(\text{structured_type}) = (\text{number_of_attributes} * 10) + 32 + \text{space_for_non-short_attributes}$$

number_of_attributes は、スーパータイプから継承される属性も含めた構造タイプの合計属性数です。ただし、*number_of_attributes* には、*structured_type* の任意のサブタイプに定義されているどの属性も含まれません。

表 268. 属性データ・タイプのバイト・カウント

属性データ・タイプ	バイト・カウント	
DECIMAL	(p/2)+1 の整数部分 (p は精度)	
CHAR (n)	n	
VARCHAR (n)	n	
GRAPHIC (n)	n * 2	
VARGRAPHIC (n)	n * 2	
TIMESTAMP	10	
DATALINK(n)	n + 54	
LOB タイプ	各 LOB 属性は、構造タイプ・インスタンス内に、実際の値の位置へのポインターとなる LOB 記述子を持っています。その記述子のサイズは、その LOB 属性に定義されている最大長によって異なります。	
	LOB の最大長	LOB 記述子のサイズ
	1 024	72
	8 192	96
	65 536	120
	524 000	144
	4 190 000	168
	134 000 000	200
	536 000 000	224
	1 070 000 000	256
	1 470 000 000	280
	2 147 483 647	316
特殊タイプ	特殊タイプのソース・タイプの長さ	
参照タイプ	参照タイプの基礎となる組み込みデータ・タイプの長さ	

表 268. 属性データ・タイプのバイト・カウント (続き)

構造タイプ	<code>inline_length(attribute_type)</code>
-------	--------------------------------------------

WITHOUT COMPARISONS

構造タイプのインスタンスで比較関数がサポートされていないことを示します。

NOT FINAL

この構造タイプを上位タイプとして使用できることを示します。

MODE DB2SQL

この文節は必須であり、このタイプでコンストラクター関数を直接呼び出すために使用します。

WITH FUNCTION ACCESS

将来作成されるメソッドを含め、該当タイプとそのサブタイプのすべてのメソッドに対して、関数表記を使ってアクセスできることを指示します。この文節を指定できるのは、UNDER 文節が指定されていない構造タイプの階層のルート・タイプだけです (SQLSTATE 42613)。この文節は、メソッドを呼び出す表記よりもこの形式の表記のほうが望ましいアプリケーションで、関数表記を使用できるようにするために提供されています。

REF USING *rep-type*

この構造タイプの参照タイプの表示 (基礎データ・タイプ) として使用される組み込みデータ・タイプとそのサブタイプをすべて定義します。この文節を指定できるのは、UNDER 文節が指定されていない構造タイプの階層のルート・タイプだけです (SQLSTATE 42613)。*rep-type* は、LONG VARCHAR、LONG VARCHARGRAPHIC、BLOB、CLOB、DBCLOB、DATALINK、または構造タイプであってはならず、32 672 バイト以下の長さでなければなりません (SQLSTATE 42613)。

構造タイプの階層のルート・タイプにこの文節を指定しない場合、REF USING VARCHAR(16) FOR BIT DATA が想定されます。

CAST (SOURCE AS REF) WITH *funcname1*

システムにより生成される関数で、データ・タイプ *rep-type* の値を、この構造タイプの参照タイプにキャストする関数の名前を定義します。*funcname1* の一部としてスキーマ名を指定することはできません (SQLSTATE 42601)。cast 関数は、構造タイプと同じスキーマ内で生成されます。この文節を指定しない場合、*funcname1* のデフォルト値は *type-name* (構造タイプの名前) になります。*funcname1(rep-type)* に一致する関数シグニチャーが、同じスキーマ内に存在してはなりません (SQLSTATE 42710)。

CAST (REF AS SOURCE) WITH *funcname2*

システムにより生成される関数で、この構造タイプの参照タイプ値を、データ・タイプ *rep-type* にキャストする関数の名前を定義します。*funcname2* の一部としてスキーマ名を指定することはできません (SQLSTATE 42601)。cast 関数は、構造

CREATE TYPE (構造化)

タイプと同じスキーマ内で生成されます。この文節を指定しない場合、*funcname2* のデフォルト値は *rep-type* (表示タイプの名前) になります。

method-specification

このタイプのメソッドを定義します。メソッドは、CREATE METHOD ステートメントで本体を与えられて初めて、実際に使用できるようになります (SQLSTATE 42884)。

OVERRIDING

定義するメソッドが、定義するタイプのスーパータイプのメソッドをオーバーライドすることを指定します。オーバーライドすることによって、サブタイプのメソッドを再インプリメントできるようになるので、より具体的な機能が提供されます。オーバーライドは、以下のタイプのメソッドではサポートされません。

- 表メソッドおよび行メソッド
- PARAMETER STYLE JAVA を使用して宣言される外部メソッド
- 索引拡張で述部として使用できるメソッド
- システムによって生成される mutator メソッドまたは observer メソッド

このようなメソッドをオーバーライド使用とすると、エラーになります (SQLSTATE 42745)。

メソッドを有効なオーバーライド・メソッドにする場合は、定義するタイプの適切なスーパータイプの 1 つに元のメソッドが 1 つ存在していなければならず、オーバーライド・メソッドと元のメソッドの間に以下の関係が存在している必要があります。

- 定義するメソッドと元のメソッドのメソッド名が同じである。
- 定義するメソッドと元のメソッドのパラメーターの数が同じである。
- 定義するメソッドの各パラメーターのデータ・タイプと、元のメソッドの対応するパラメーターのデータ・タイプが同一である。この要件では、暗黙の SELF パラメーターは考慮されません。

このような元のメソッドが存在しない場合は、エラーが戻されます (SQLSTATE 428FV)。

オーバーライド・メソッドは、以下の属性を元のメソッドから継承します。

- 言語
- 決定性の指示
- 外部アクションの指示
- 引き数に NULL 値がある場合にメソッドを呼び出すかどうかの指示
- 結果のキャスト (元のメソッドで指定されている場合)
- SELF AS RESULT の指示
- SQL データ・アクセスまたは CONTAINS SQL の指示

- 外部メソッドの場合は、以下のとおりです。
 - パラメーターのスタイル
 - パラメーターと結果のロケーターの指示 (元のメソッドで指定されている場合)
 - FENCED、SCRATCHPAD、FINAL CALL、ALLOW PARALLEL、および DBINFO の指示
 - INHERIT SPECIAL REGISTER および THREADSAFE の指示

method-name

定義しようとするメソッドを指定します。これは、修飾されていない SQL ID でなければなりません (SQLSTATE 42601)。メソッド名は、CREATE TYPE に使用されるスキーマで暗黙的に修飾されます。

述部のキーワードとして使用されるいくつかの名前は、システム使用に予約されており、*method-name* として使用することはできません (SQLSTATE 42939)。それらの名前は、SOME、ANY、ALL、NOT、AND、OR、BETWEEN、NULL、LIKE、EXISTS、IN、UNIQUE、OVERLAPS、SIMILAR、MATCH、および比較演算子です。

一般に、メソッドのシグニチャーがそれぞれ異なっている場合は、同じ名前を複数のメソッドに使用することができます。

parameter-name

パラメーター名を指定します。その名前は SELF であってはなりません。これは、メソッドの暗黙のサブジェクト・パラメーターの名前です (SQLSTATE 42734)。メソッドが SQL メソッドである場合、そのすべてのパラメーターに名前が付いていなければなりません (SQLSTATE 42629)。宣言するメソッドが別のメソッドをオーバーライドする場合は、パラメーター名は、オーバーライドされるメソッドの対応するパラメーターの名前と正確に一致している必要があります。そうでないと、エラーが戻されません (SQLSTATE 428FV)。

data-type2

各パラメーターのデータ・タイプを指定します。メソッドが受け取るはずの各パラメーターごとに 1 つの項目をこのリストに指定する必要があります。暗黙の SELF パラメーターを含め、90 を超える数のパラメーターを使うことはできません。この限界を超えると、エラーになります (SQLSTATE 54023)。

CREATE TABLE ステートメントに列タイプとして指定でき、しかもメソッドの作成に使用されている言語に対応するような SQL データ・タイプ指定と省略形を指定することができます。ユーザー定義関数とメソッドに関する SQL データ・タイプとホスト言語データ・タイプの対応については、DB2 アプリケーション開発ガイドの言語別の項を参照してください。

CREATE TYPE (構造化)

注: 該当する SQL データ・タイプが構造タイプである場合、ホスト言語データ・タイプに対するデフォルト・マッピングはありません。構造タイプとホスト言語データ・タイプとをマッピングするには、ユーザー定義のトランスフォーム関数を使用する必要があります。

DECIMAL (および NUMERIC) は、LANGUAGE C と OLE では無効です (SQLSTATE 42815)。

REF を指定することができますが、これには定義された有効範囲はありません。メソッドの本体で、まず参照タイプをキャストして有効範囲をもたせて初めて、パス式内でその参照タイプを使用できるようになります。同様に、メソッドから戻された参照も、まずキャストして有効範囲をもたせて初めて、パス式内で使用できるようになります。

AS LOCATOR

LOB タイプまたは LOB タイプに基づく特殊タイプの場合、AS LOCATOR 文節を追加することができます。これは、実際の値の代わりに LOB ロケーターをメソッドに渡すことを指定します。これにより、メソッドに渡すバイト数を大幅に削減することができ、パフォーマンスも向上します。メソッドにとって実際に必要になる値が数バイトだけである場合は特にそうです。

LOB または LOB に基づく特殊タイプ以外のタイプに対して AS LOCATOR を指定すると、エラーが発生します (SQLSTATE 42601)。

関数が FENCED の場合や、LANGUAGE が SQL の場合、AS LOCATOR 文節は指定できません (SQLSTATE 42613)。

宣言するメソッドが別のメソッドをオーバーライドする場合は、パラメーターの AS LOCATOR 指示は、オーバーライドされるメソッドの対応するパラメーターの AS LOCATOR 指示と正確に一致している必要があります (SQLSTATE 428FV)。

宣言するメソッドが別のメソッドをオーバーライドする場合は、各パラメーターの FOR BIT DATA 指示は、オーバーライドされるメソッドの対応するパラメーターの FOR BIT DATA 指示と正確に一致している必要があります。 (SQLSTATE 428FV)。

RETURNS

これは必須の文節であり、メソッドの結果を指定します。

data-type3

メソッドの結果のデータ・タイプを指定します。この場合、上記のメソッドのパラメーター *data-type2* の項で説明したのと全く同じ考慮事項が当てはまりません。

AS LOCATOR

LOB タイプまたは LOB タイプに基づく特殊タイプの場合、AS

LOCATOR 文節を追加することができます。これは、実際の値の代わりに LOB ロケーターがメソッドから渡されることを示します。

LOB または LOB に基づく特殊タイプ以外のタイプに対して AS LOCATOR を指定すると、エラーが発生します (SQLSTATE 42601)。

関数が FENCED の場合や、LANGUAGE が SQL の場合、AS LOCATOR 文節は指定できません (SQLSTATE 42613)。

定義するメソッドが別のメソッドをオーバーライドする場合は、この文節を指定できません (SQLSTATE 428FV)。

メソッドが別のメソッドをオーバーライドする場合、*data-type3* は、データ・タイプが構造タイプであれば、オーバーライドされるメソッドの結果のデータ・タイプのサブタイプでなければなりません。そうでない場合は、両方のデータ・タイプは同じでなければなりません (SQLSTATE 428FV)。

data-type4 **CAST FROM** *data-type5*

メソッドの結果のデータ・タイプを指定します。

この文節は、メソッド・コードから戻されたデータ・タイプとは異なるデータ・タイプを、呼び出しステートメントに戻すのに使用されます。*data-type5* は、*data-type4* パラメーターに対してキャスト可能でなければなりません。キャスト可能でないと、エラーになります (SQLSTATE 42880)。

data-type4 の長さ、精度または位取りは、*data-type5* から推断することができるので、*data-type4* に指定されるパラメーター化タイプの長さ、精度、または位取りを指定する必要はありません (指定は可能です)。代わりに、VARCHAR() のような空の括弧を使用できます。パラメーター値が異なるデータ・タイプ (REAL または DOUBLE) を示しているため、FLOAT() を使用することはできません (SQLSTATE 42601)。

特殊タイプは、*data-type5* に指定するタイプとしては無効です (SQLSTATE 42815)。

キャスト操作は実行時チェックの対象にもなり、その結果、変換エラーになる可能性があります。

AS LOCATOR

LOB タイプまたは LOB タイプに基づく特殊タイプの場合、AS LOCATOR 文節を追加することができます。これは、実際の値の代わりに LOB ロケーターがメソッドから渡されることを示します。

LOB または LOB に基づく特殊タイプ以外のタイプに対して AS LOCATOR を指定すると、エラーが発生します (SQLSTATE 42601)。

関数が FENCED の場合や、LANGUAGE が SQL の場合、AS LOCATOR 文節は指定できません (SQLSTATE 42613)。

CREATE TYPE (構造化)

定義するメソッドが別のメソッドをオーバーライドする場合は、この文節を指定できません (SQLSTATE 428FV)。

定義するメソッドが別のメソッドをオーバーライドする場合は、FOR BIT DATA 文節を指定できません (SQLSTATE 428FV)。

SPECIFIC *specific-name*

定義するメソッドのインスタンスに対する固有名を指定します。この名前は、メソッドの本体の作成やメソッドのドロップのときに使用することができます。これは、メソッドの呼び出しには使用できません。 *specific-name* (関数名) の非修飾形式は SQL ID です (最大長 18)。修飾形式は、スキーマ名とその後続くピリオドと SQL ID です。暗黙または明示の修飾子も含め、その名前が、アプリケーション・サーバーに存在する別の個別メソッド名を指定するものであってはなりません。そうでない場合は、エラーになります (SQLSTATE 42710)。

specific-name は、既存の *method-name* と同じでも構いません。

修飾子を指定しない場合、*type-name* に使用された修飾子が使用されます。修飾子を指定する場合は、*type-name* の明示修飾子または暗黙修飾子と同じでなければなりません。そうでない場合は、エラーになります (SQLSTATE 42882)。

specific-name の指定がない場合、固有の名前がデータベース・マネージャーによって生成されます。生成される固有名は、SQL の後に文字のタイム・スタンプが続く名前です (SQLyymmddhhmmssxxx)。

SELF AS RESULT

このメソッドがタイプ保存メソッドであることを指定します。その意味は以下のとおりです。

- 宣言された戻りタイプは、宣言されたサブジェクト・タイプと同じでなければなりません (SQLSTATE 428EQ)。
- SQL ステートメントがコンパイルされ、タイプ保存メソッドに解決されると、そのメソッド結果の静的タイプは、サブジェクト引き数の静的タイプと同じになります。
- メソッドをインプリメントする場合、結果の動的タイプが、サブジェクト引き数の静的タイプと同じになる (SQLSTATE 2200G) ようにし、そしてその結果 NULL にならない (SQLSTATE 22004) ようにする必要があります。

定義するメソッドが別のメソッドをオーバーライドする場合は、この文節を指定できません (SQLSTATE 428FV)。

SQL-routine-characteristics

CREATE METHOD を使ってこのタイプに定義されるメソッド本体の特性を指定します。

LANGUAGE SQL

この文節を使って、単一の RETURN ステートメントを使って SQL でメソッドを作成することを指示します。メソッド本体は、CREATE METHOD ステートメントを使って指定します。

NOT DETERMINISTIC または DETERMINISTIC

この文節はオプションですが、特定の引き数の値に対してメソッドが常に同じ結果を戻すか (DETERMINISTIC)、それとも状態値に応じてメソッドの結果が異なるか (NOT DETERMINISTIC) を指定します。つまり DETERMINISTIC メソッドは、同じ入力を使用して正しく呼び出した場合に常に同じ結果を戻します。NOT DETERMINISTIC を指定すると、同じ入力によって常に同じ結果が生じる利点に基づく最適化ができなくなります。メソッド本体が特殊レジスターにアクセスしたり、別の非 deterministic ルーチンを呼び出したりする場合、明示的または暗黙的に NOT DETERMINISTIC を指定しなければなりません (SQLSTATE 428C2)。

EXTERNAL ACTION または NO EXTERNAL ACTION

この文節はオプションであり、データベース・マネージャーによって管理されていないオブジェクトの状態を変更する処置をメソッドが行うか否かを指定します。EXTERNAL ACTION を指定すると、外部からメソッドへの影響がないことを前提とした最適化ができなくなります。(たとえば、メッセージの送信、警報音による通知、ファイルへのレコードの書き込みなど。)

READS SQL DATA または CONTAINS SQL

どのタイプの SQL ステートメントを実行できるかを指示します。サポートされている SQL ステートメントは RETURN ステートメントであるので、式が副照会であるかどうかで区別を行います。

READS SQL DATA

SQL データを変更しない SQL ステートメントを、メソッドで実行できることを指定します (SQLSTATE 42985)。SQL ステートメント内でニックネームを参照することはできません (SQLSTATE 42997)。

CONTAINS SQL

SQL データの読み取りも変更も行わない SQL ステートメントを、メソッドで実行できることを指定します (SQLSTATE 42985)。

CALLED ON NULL INPUT

このオプション文節は、引き数が NULL 値か否かに関係なくユーザー定義メソッドを呼び出すことを指定します。これは、NULL 値を戻す場合も、通常 (NULL 値以外の) 値を戻す場合もあります。ただし、NULL の引き数値の有無のテストはメソッドが行う必要があります。

定義するメソッドが別のメソッドをオーバーライドする場合は、この文節を指定できません (SQLSTATE 428FV)。

CREATE TYPE (構造化)

NULL CALL は、CALLED ON NULL INPUT の同義語として使うことができます。

INHERIT SPECIAL REGISTERS

このオプションの文節は、メソッド内の更新可能特殊レジスターが、初期値を呼び出し側ステートメントの環境から継承することを指定します。カーソルの SELECT ステートメントから呼び出されるメソッドの場合は、初期値はカーソルがオープンされた環境から継承されます。ネストされたオブジェクトで呼び出されるルーチン (トリガーまたはビューなど) の場合は、初期値はランタイム環境から継承されます (オブジェクト定義からは継承されません)。

特殊レジスターに対する変更が、関数の呼び出し元に戻されることはありません。

更新不能の特殊レジスター (日時特殊レジスターなど) は、現在実行中のステートメントのプロパティを反映するので、デフォルト値に設定されます。

external-routine-characteristics

LANGUAGE

この文節は必須で、ユーザー定義メソッドの本体が準拠している言語インターフェース規則を指定するのに使用します。

C これは、データベース・マネージャーが、ユーザー定義メソッドを C の関数であるかのように呼び出すことを意味します。ユーザー定義メソッドは、標準 ANSI C プロトタイプで定義されている C 言語の呼び出しおよびリンケージの規則に準拠していなければなりません。

JAVA

データベース・マネージャーは、Java クラスのメソッドとしてユーザー定義メソッドを呼び出します。

OLE

データベース・マネージャーは、OLE 自動化オブジェクトによって公開されたメソッドとして、ユーザー定義メソッドを呼び出します。メソッドは、*OLE Automation Programmer's Reference* に記述されている OLE 自動化データ・タイプと呼び出しメカニズムに準拠している必要があります。

LANGUAGE OLE は、Windows 32 ビット・オペレーティング・システムで保管されたユーザー定義メソッドに対してのみサポートされます。

THREADSAFE は、LANGUAGE OLE で定義されたメソッドに指定することはできません (SQLSTATE 42613)。

PARAMETER STYLE

この文節は、メソッドに対してパラメーターを渡し、そこから値を戻すのに用いる規則を指定するのに使用されます。

DB2GENERAL

Java クラスのメソッドとして定義された外部メソッドとの間で、パラメー

ターを渡し、値を戻す場合に用いる規則を指定します。これは、**LANGUAGE JAVA** を使用する場合にだけ指定する必要があります。

DB2GENERAL の同義語として値 **DB2GENRL** が使用可能です。

SQL

C 言語の呼び出しとリンケージの規則、または **OLE** 自動化オブジェクトによって公開されたメソッドに準拠する規則を、この外部メソッドとの間でパラメーターを渡し、値を戻す場合の規則として指定します。これは、**LANGUAGE C** または **LANGUAGE OLE** を使用する場合に指定する必要があります。

DETERMINISTIC または **NOT DETERMINISTIC**

この文節はオプションですが、特定の引き数の値に対してメソッドが常に同じ結果を戻すか (**DETERMINISTIC**)、それとも状態値に応じてメソッドの結果が異なるか (**NOT DETERMINISTIC**) を指定します。つまり **DETERMINISTIC** メソッドは、同じ入力を使用して正しく呼び出した場合に常に同じ結果を戻します。**NOT DETERMINISTIC** を指定すると、同じ入力によって常に同じ結果が生じる利点に基づく最適化ができなくなります。

NOT DETERMINISTIC メソッドの例として、ある部署の社員の通し番号をランダムに返すメソッドが挙げられます。**DETERMINISTIC** メソッドの例として、多角形の面積を計算するメソッドが挙げられます。

FENCED または **NOT FENCED**

この文節は、データベース・マネージャーの操作環境のプロセスまたはアドレス・スペースでメソッドを実行しても「安全」か (**NOT FENCED**)、そうでないか (**FENCED**) を指定します。

メソッドが **FENCED** として登録されると、データベース・マネージャーは、その内部リソース (データ・バッファーなど) を保護して、そのメソッドからアクセスされないようにします。多くのメソッドは、**FENCED** または **NOT FENCED** のどちらかで実行するように選択することができます。一般に、**FENCED** として実行されるメソッドは、**NOT FENCED** として実行されるものと同じようには実行されません。

注意:

十分にチェックされていないメソッドに **NOT FENCED** を使用すると、**DB2** の保全性に危険を招く場合があります。**DB2** では、発生する可能性のある一般的な不注意による障害の多くに対して、いくつかの予防措置がとられていますが、**NOT FENCED** ユーザー定義メソッドが使用される場合には、完全な保全性を確保できません。

LANGUAGE OLE または **NOT THREADSAFE** を指定したメソッドには、**FENCED** のみを指定できます (**SQLSTATE 42613**)。

CREATE TYPE (構造化)

メソッドが **FENCED** で **NO SQL** オプションが指定されている場合、**AS LOCATOR** 文節を指定できません (SQLSTATE 42613)。

メソッドを **NOT FENCED** として登録するには、**SYSADM** 権限、**DBADM** 権限、または特殊権限 (**CREATE_NOT_FENCED_ROUTINE**) が必要です。

THREADSAFE または **NOT THREADSAFE**

メソッドを他のルーチンと同じプロセスで実行しても『安全』か (**THREADSAFE**)、そうでないか (**NOT THREADSAFE**) を指定します。

メソッドが **OLE** 以外の **LANGUAGE** で定義される場合:

- メソッドが **THREADSAFE** に定義されている場合には、データベース・マネージャーは他のルーチンと同じプロセスでメソッドを呼び出すことができます。一般に、スレッド・セーフにするには、メソッドはどのグローバルあるいは静的データ域をも使用してはなりません。多くのプログラミング解説書には、スレッド・セーフ・ルーチンの作成に関する説明が含まれています。 **FENCED** および **NOT FENCED** メソッドの両方が **THREADSAFE** になることが可能です。
- メソッドが **NOT THREADSAFE** として定義される場合には、データベース・マネージャーは他のルーチンと同じプロセスにメソッドを決して呼び出しません。

FENCED メソッドについては、**LANGUAGE** が **JAVA** の場合、**THREADSAFE** がデフォルトです。これ以外のすべての言語の場合は、**NOT THREADSAFE** がデフォルトです。メソッドが **LANGUAGE OLE** とともに定義される場合には、**THREADSAFE** は指定されません (SQLSTATE 42613)。

NOT FENCED メソッドについては、**THREADSAFE** がデフォルトです。 **NOT THREADSAFE** を指定することはできません (SQLSTATE 42613)。

RETURNS NULL ON NULL INPUT または **CALLED ON NULL INPUT**

このオプション文節を使用すると、非サブジェクト引き数のいずれかが **NULL** 値の場合に、外部メソッドを呼び出さないようにすることができます。

RETURNS NULL ON NULL INPUT が指定されており、実行時にメソッドの引き数のいずれかが **NULL** 値の場合、このメソッドは呼び出されず、結果は **NULL** 値になります。

CALLED ON NULL INPUT を指定すると、**NULL** 値の引き数の数に関係なくメソッドが呼び出されます。これは、**NULL** 値を戻す場合も、通常の (**NULL** 値以外の) 値を戻す場合もあります。ただし、**NULL** の引き数値の有無のテストはメソッドが行う必要があります。

値 **NULL CALL** は、上位互換またはファミリーの互換性のために、**CALLED ON NULL INPUT** の同義語として使うことができます。同様に、**NOT NULL CALL** は、**RETURNS NULL ON NULL INPUT** の同義語として使用できます。

以下の 2 通りの場合、この指定は無視されます。

- 対象となる引き数が NULL の場合。この場合、メソッドは実行されずに結果は NULL になります。
- パラメーターがないものとしてメソッドを定義した場合。この場合、この NULL 引き数条件が成立することはありません。

NO SQL、CONTAINS SQL、READS SQL DATA

メソッドから SQL ステートメントが発行されるかどうかと、もし発行されればどのタイプかを示します。

NO SQL

メソッドはどの SQL ステートメントも実行できないことを指示します (SQLSTATE 38001)。

CONTAINS SQL

SQL データの読み取りも変更も行わない SQL ステートメントを、メソッドで実行できることを指定します (SQLSTATE 38004 または 42985)。どのメソッドでもサポートされていないステートメントは、これとは異なるエラーを戻します (SQLSTATE 38003 または 42985)。

READS SQL DATA

SQL データを変更しない SQL ステートメントを、メソッドで実行できることを指定します (SQLSTATE 38002 または 42985)。どのメソッドでもサポートされていないステートメントは、これとは異なるエラーを戻します (SQLSTATE 38003 または 42985)。

EXTERNAL ACTION または NO EXTERNAL ACTION

この文節はオプションであり、データベース・マネージャーによって管理されていないオブジェクトの状態を変更する処置をメソッドが行うか否かを指定します。EXTERNAL ACTION を指定すると、外部からメソッドへの影響がないことを前提とした最適化ができなくなります。

NO SCRATCHPAD または SCRATCHPAD *length*

この文節はオプションであり、この外部メソッドに対してスクラッチパッドを用意するか否かを指定するのに使用できます。メソッドを再入可能にすることを強くお勧めします。再入可能にすると、スクラッチパッドが、呼び出しのたびにメソッドに「状態を保管」させる手段になります。

SCRATCHPAD を指定すると、ユーザー定義メソッドの最初の呼び出し時に、その外部メソッドによって使用されるスクラッチパッドにメモリーが割り振られます。このスクラッチパッドには、以下の特性があります。

- *length* を指定して、スクラッチパッドのバイト単位のサイズを設定します。これは 1 ~ 32,767 でなければなりません (SQLSTATE 42820)。デフォルト値は 100 です。
- すべて 'X'00' に初期化されます。

CREATE TYPE (構造化)

- その有効範囲は、該当の SQL ステートメントです。SQL ステートメントでの外部メソッドに対する参照ごとに 1 つのスクラッチパッドがあります。

したがって、次のステートメントのメソッド X が SCRATCHPAD キーワードを指定して定義されると、3 つのスクラッチパッドが割り当てられます。

```
SELECT A, X..(A) FROM TABLEB
WHERE X..(A) > 103 OR X..(A) < 19
```

ALLOW PARALLEL が指定されているか、またはデフォルト値として使用された場合、その有効範囲は上記とは異なります。メソッドが複数のパーティションで実行される場合、メソッドが処理されるそれぞれのパーティションにおいて、SQL ステートメントでのメソッドへのそれぞれの参照ごとにスクラッチパッドが割り当てられます。同様に、パーティション内並列処理をオンにして照会が実行される場合、3 つ以上のスクラッチパッドが割り当てられることがあります。

スクラッチパッドは持続します。その内容は、外部メソッドの呼び出しごとに保存されます。外部メソッドのある呼び出しによってスクラッチパッドに加えられた変更はいずれも、次の呼び出し時に存続しています。データベース・マネージャーは、各 SQL ステートメントの実行開始時に、スクラッチパッドを初期設定します。各副照会の実行開始時には、データベース・マネージャーによってスクラッチパッドがリセットされます。FINAL CALL オプションが指定されている場合、システムは、スクラッチパッドのリセットに先立って、最終呼び出しを行います。

スクラッチパッドは、外部メソッドが獲得できるシステム・リソース (メモリーなど) の中央点として使用することもできます。メソッドは、最初の呼び出しでメモリーを獲得し、そのアドレスをスクラッチパッドに保管して、後の呼び出しでそれを参照することができます。

このようにシステム・リソースが獲得される場合、FINAL CALL キーワードも指定する必要があります。そうすると、ステートメントの最後で特殊な呼び出しが行われ、外部メソッドは獲得したシステム・リソースをすべて解放することができます。

SCRATCHPAD を指定すると、ユーザー定義メソッドを呼び出すたびに、スクラッチパッドをアドレッシングする外部メソッドに追加の引き数が渡されません。

NO SCRATCHPAD を指定すると、外部メソッドに対してスクラッチパッドは割り振られず、渡されません。

NO FINAL CALL または FINAL CALL

この文節はオプションであり、外部メソッドに対する最終呼び出しが行われる

か否かを指定します。このような最終呼び出しの目的は、外部メソッドが獲得したシステム・リソースすべてを解放できるようにすることです。外部メソッドがメモリーなどのシステム・リソースを獲得し、それをスクラッチパッドに固定するような状況では、これを **SCRATCHPAD** キーワードと共に使用すると便利です。

FINAL CALL を指定すると、実行時に、呼び出しのタイプを指定する外部メソッドに追加の引き数が渡されます。呼び出しのタイプは以下のとおりです。

- 通常呼び出し。SQL 引き数が渡され、結果が戻されることが予期されます。
- 最初の呼び出し。この SQL ステートメントのメソッドに対する参照に対応する外部メソッドの最初の呼び出しです。最初の呼び出しは通常呼び出しです。
- 最終呼び出し。外部メソッドがリソースを解放できるようにするそのメソッドに対する最終呼び出しです。最終呼び出しは、通常呼び出しではありません。この最終呼び出しは、以下の時点で行われます。
 - ステートメント終了時。これは、カーソルの関係するステートメントでカーソルがクローズされた場合、あるいはステートメントが実行を終了した場合に発生します。
 - トランザクション終了時。これは、通常のステートメント終了が発生しなかった場合に発生します。たとえば、何らかの理由で、アプリケーションのロジックが、カーソルをクローズしないようになっている場合があります。

WITH HOLD として定義されたカーソルがオープンされている間に、コミット操作が発生すると、それ以降のカーソルのクローズ時、またはアプリケーションの終了時に最終呼び出しが行われます。

NO FINAL CALL を指定すると、「呼び出しタイプ」の引き数は外部メソッドに渡されず、最終呼び出しは行われません。

ALLOW PARALLEL または **DISALLOW PARALLEL**

この文節はオプションで、メソッドへの 1 つの参照で、メソッドの呼び出しを並列化できるか否かを指定します。一般には、ほとんどのスカラー・メソッドは並列化可能ですが、並列化できないメソッド (1 つのスクラッチパッドのコピーに依存するメソッドなど) もあります。スカラー・メソッドに対して **ALLOW PARALLEL** または **DISALLOW PARALLEL** を指定すると、DB2 はその指定を受け入れます。

メソッドにどちらのキーワードが当てはまるかを判別するには、以下の点を検討する必要があります。

- メソッドのすべての呼び出しが、互いに完全に独立していますか? **YES** の場合には、**ALLOW PARALLEL** を指定します。

CREATE TYPE (構造化)

- メソッドを呼び出すごとに、次の呼び出しに関係する値を提供するスクラッチパッドが更新されますか? (たとえば、カウンターの増分によって。) YES の場合には、DISALLOW PARALLEL を指定するか、またはデフォルトを受け入れます。
- 1 つのパーティションでのみ起こる必要のある外部アクションがメソッドによって実行されますか? YES の場合には、DISALLOW PARALLEL を指定するか、またはデフォルトを受け入れます。
- コストのかかる初期化処理の実行回数を最小にするためだけに、スクラッチパッドを使用していますか? YES の場合には、ALLOW PARALLEL を指定します。

いずれの場合も、すべての外部メソッドの本体は、データベースのすべてのパーティションで使用可能なディレクトリーにある必要があります。

構文図は、デフォルト値が ALLOW PARALLEL であることを示しています。しかし、ステートメントで以下のオプションの少なくとも 1 つが指定されている場合は、デフォルトは DISALLOW PARALLEL です。

- NOT DETERMINISTIC
- EXTERNAL ACTION
- SCRATCHPAD
- FINAL CALL

NO DBINFO または DBINFO

この文節はオプションで、DB2 において既知である特定の情報を追加の呼び出し時に引き数としてメソッドに渡すか (DBINFO)、または渡さないか (NO DBINFO) を指定します。NO DBINFO がデフォルト値です。DBINFO は、LANGUAGE OLE ではサポートされません (SQLSTATE 42613)。定義するメソッドが別のメソッドをオーバーライドする場合は、この文節を指定できません (SQLSTATE 428FV)。

DBINFO を指定すると、以下の情報をもつ構造がメソッドに渡されます。

- データベース名 - 現在接続されているデータベースの名前。
- アプリケーション ID - データベースへの接続ごとに確立された、固有のアプリケーション ID。
- アプリケーション許可 ID - アプリケーション実行時の許可 ID。このメソッドとアプリケーションとの間でネストされているメソッドは無関係です。
- コード・ページ - データベースのコード・ページを識別します。
- スキーマ名 - 表名とまったく同じ条件のもとでは、スキーマの名前が入ります。その他の場合はブランクです。

- 表名 - メソッド参照が UPDATE ステートメントの SET 文節の右側にある場合、または INSERT ステートメントの VALUES リストの項目である場合のいずれかに限り、更新または挿入される表の非修飾名が入ります。その他の場合はブランクです。
- 列名 - 表名とまったく同じ条件で、更新または挿入される列の名前が入ります。その他の場合はブランクです。
- データベースのバージョン/リリース - メソッドを呼び出すデータベース・サーバーのバージョン、リリース、および修正レベルを識別します。
- プラットフォーム - サーバーのプラットフォーム・タイプが入ります。
- 表メソッドの結果の列番号 - メソッドには当てはまりません。

INHERIT SPECIAL REGISTERS

このオプションの文節は、メソッド内の特殊レジスターが、初期値を呼び出し側ステートメントから継承することを指定します。カーソルの場合は、初期値はカーソルがオープンされる時に継承されます。

特殊レジスターに対する変更が、メソッドの呼び出し元に戻されることはありません。

一部の特殊レジスター (日時特殊レジスターなど) は、現在実行中のステートメントのプロパティを反映するので、呼び出し元からの継承は行われません。

注:

- **互換性**
 - DB2 UDB (OS/390 および z/OS 版) との互換性:
 - 以下の構文が許容されます。
 - DETERMINISTIC の代わりに NOT VARIANT を指定できます。
 - NOT DETERMINISTIC の代わりに VARIANT を指定できます。
 - CALLED ON NULL INPUT の代わりに NULL CALL を指定できます。
 - RETURNS NULL ON NULL INPUT の代わりに NOT NULL CALL を指定できます。
 - 以前のバージョンの DB2 との互換性:
 - PARAMETER STYLE SQL の代わりに PARAMETER STYLE DB2SQL を指定できます。
 - まだ存在していないスキーマ名を用いて構造タイプを作成すると、ステートメントの許可 ID に IMPLICIT_SCHEMA 権限がある場合に限り、そのスキーマが暗黙的に作成されます。そのスキーマの所有者は SYSIBM になります。スキーマに対する CREATEIN 特権が PUBLIC に付与されます。
 - 属性なしで定義された構造化サブタイプは、属性をすべて上位タイプから継承するサブタイプを定義します。UNDER 文節も他のどの属性も指定しない場合、タイプは、属性なしの、タイプ階層のルート・タイプになります。

CREATE TYPE (構造化)

- タイプ階層に新たにサブタイプを追加すると、パッケージが無効になることがあります。パッケージは、その新しいタイプのスーパータイプに依存していると、無効になることがあります。このような従属関係は、TYPE 述部または TREAT 指定を使用した結果として生じます。
- 構造タイプは 4082 個を超える属性をもつことはできません (SQLSTATE 54050)。
- 関数と同じシグニチャーをもつメソッド (関数の最初のパラメーター・タイプと、メソッドのサブジェクト・タイプの比較による) を指定することはできません。
- 元のメソッドは、別のメソッドをオーバーライドしたり、元のメソッドによってオーバーライドされたりしてはなりません (SQLSTATE 42745)。さらに、関数とメソッドは、オーバーライド関係にあってはなりません。つまり、関数は、サブジェクト S を第 1 パラメーターとしてもつメソッドであると見なされる場合、S のスーパータイプの別のメソッドをオーバーライドしてはならず、S のサブタイプの別のメソッドによってオーバーライドされてはならないという意味です (SQLSTATE 42745)。
- ある構造タイプを作成すると、そのタイプで使用される一連の関数とメソッドが自動的に生成されます。これらの関数とメソッドはすべて、構造タイプと同じスキーマ内で生成されます。生成された関数またはメソッドのシグニチャーが、このスキーマに存在する関数のシグニチャーと競合またはそれをオーバーライドする場合、このステートメントは失敗します (SQLSTATE 42710)。構造タイプをドロップしないで、生成された関数またはメソッドをドロップすることはできません (SQLSTATE 42917)。以下のような関数とメソッドが生成されます。

- 関数

- 参照比較

REF(*type-name*) という参照タイプでは、=、<>、<、<=、>、>= という名前の 6 つの比較関数が生成されます。これらの関数はそれぞれ REF(*type-name*) というタイプのパラメーターを 2 つ受け取ってから、真、偽、または不明という値を戻します。REF(*type-name*) の比較演算子は、REF(*type-name*) の基礎データ・タイプと同じ動作をするように定義されます。(タイプ階層に含まれる参照表示タイプはすべて同一のものです。これにより、REF(S) と REF(T) の比較が可能になります (S と T が共通の上位タイプを持っている場合)。表の OID 列は、表階層だけで固有になるので、1 つの表階層の REF(T) 値を別の表階層の REF(T) 値と「等しい」ものにすることができます (それぞれが異なった行を参照していても))。

参照タイプの有効範囲は比較の対象にはなりません。

- Cast 関数

生成された参照タイプである REF(*type-name*) とこの参照タイプの基礎データ・タイプとの間をキャストするために 2 つの cast 関数が生成されます。

- 基礎タイプから参照タイプへとキャストする関数の名前は、暗黙的または明示的な *funcname1* です。

この関数の形式は以下のとおりです。

```
CREATE FUNCTION funcname1 (rep-type)
  RETURNS REF(type-name) ...
```

- 参照タイプから基礎タイプ (参照タイプの) へとキャストする関数の名前は、暗黙的または明示的な *funcname2* です。

この関数の形式は以下のとおりです。

```
CREATE FUNCTION funcname2 ( REF(type-name) )
  RETURNS rep-type ...
```

ある種の *rep-type* には、定数からのキャストを操作する *funcname1* を使って生成された追加の *cast* 関数があります。

- rep-type* が SMALLINT の場合、追加で生成された *cast* 関数の形式は以下のとおりです。

```
CREATE FUNCTION funcname1 (INTEGER)
  RETURNS REF(type-name)
```

- rep-type* が CHAR(*n*) の場合、追加で生成された *cast* 関数の形式は以下のとおりです。

```
CREATE FUNCTION funcname1 ( VARCHAR(n) )
  RETURNS REF(type-name)
```

- rep-type* が GRAPHIC(*n*) の場合、追加で生成された *cast* 関数の形式は以下のとおりです。

```
CREATE FUNCTION funcname1 (VARGRAPHIC(n))
  RETURNS REF(type-name)
```

それらの演算子や *cast* 関数を SQL ステートメントで正しく使用するには、SQL パスに構造タイプのスキーマ名が組み込まれていなければなりません。

- コンストラクター関数

コンストラクター関数は、そのタイプの新しいインスタンスを構成可能にするために生成されます。この新しいインスタンスでは、スーパータイプから継承する属性も含め、そのタイプのどの属性も NULL になります。

生成されるコンストラクター関数の形式は、以下のとおりです。

```
CREATE FUNCTION type-name ( )
  RETURNS type-name
  ...
```

NOT INSTANTIABLE を指定すると、コンストラクター関数は生成されません。構造タイプがタイプ DATALINK の属性を持っていると、コンストラクター機能の呼び出しは失敗します (SQLSTATE 428ED)。

- メソッド

- observer メソッド

構造タイプの各属性ごとに *observer* メソッドが定義されます。observer メソッドは、各属性ごとに属性タイプを戻します。対象が NULL の場合、observer メソッドは、属性タイプの NULL 値を戻します。

CREATE TYPE (構造化)

たとえば、C1..STREET、C1..CITY、C1..COUNTRY、および C1..CODE を使って、構造タイプ ADDRESS のインスタンスの属性を監視することができます。

生成される observer メソッドのメソッド・シグニチャーは、以下のようなステートメントが実行された場合に似ています。

```
CREATE TYPE type-name
...
METHOD attribute-name()
  RETURNS attribute-type
```

type-name は、構造タイプ名です。

- mutator メソッド

構造タイプの各属性ごとに、タイプ保存の mutator メソッドが定義されます。構造タイプのインスタンス内の属性を変更するには、mutator メソッドを使用します。mutator メソッドは、各属性ごとに、サブジェクトのコピーの指定属性に引き数を割り当てることで変更されたそのコピーを戻します。

たとえば、C1..CODE('M3C1H7') を使って、構造タイプ ADDRESS のインスタンスを突然変異することができます。サブジェクトが NULL の場合、mutator メソッドはエラーを生じます (SQLSTATE 2202D)。

生成される mutator メソッドのメソッド・シグニチャーは、以下のようなステートメントが実行された場合に似ています。

```
CREATE TYPE type-name
...
METHOD attribute-name (attribute-type)
  RETURNS type-name
```

属性のデータ・タイプが SMALLINT、REAL、CHAR、または GRAPHIC である場合、定数を使用する突然変異をサポートするため、以下のような追加の mutator メソッドが生成されます。

- *attribute-type* が SMALLINT の場合、追加の mutator はタイプ INTEGER の引き数をサポートします。
 - *attribute-type* が REAL の場合、追加の mutator はタイプ DOUBLE の引き数をサポートします。
 - *attribute-type* が CHAR の場合、追加の mutator はタイプ VARCHAR の引き数をサポートします。
 - *attribute-type* が GRAPHIC の場合、追加の mutator はタイプ VARGRAPHIC の引き数をサポートします。
- 列タイプとして構造タイプを使用する場合、そのタイプのインスタンスの長さには、実行時に 1 GB を超えてはなりません (SQLSTATE 54049)。
- 既存の構造タイプの新しいサブタイプを作成する (列タイプとして使用するため) 場合、それに関連した既存の構造タイプのサポートとしてすでに作成されているすべてのトランスフォーム関数を再検査し、必要があれば更新する必要があります。その新

しいタイプが、特定のタイプとして同じ階層内にあっても、あるいはネストされたタイプの階層内にあっても、そのタイプに関連した既存のトランスフォーム関数を変更して、新規のサブタイプによって導入される新しい属性の一部または全部を組み込む必要があると考えられます。概して、それは、UDF とクライアント・アプリケーションから構造タイプにアクセスさせるための特定のタイプ (またはタイプ階層) に関連した一連のトランスフォーム関数であるため、特定の複合階層内のすべての属性 (つまり、すべてのサブタイプとそのネストされた構造タイプの推移的な閉止を含む) をサポートするように、トランスフォーム関数を作成する必要があります。

既存のタイプの新しいサブタイプを作成すると、作成されたタイプのスーパータイプで定義されるメソッドで、しかもオーバーライドが可能なメソッドに従属するすべてのパッケージは無効になります。

• 表アクセスの制限

メソッドが READS SQL DATA として定義されている場合は、メソッド内のステートメントは、このメソッドを呼び出したステートメントによって変更される表にアクセスすることはできません (SQLSTATE 57053)。たとえば、メソッド BONUS() が READS SQL DATA として定義されているとします。ステートメント UPDATE DEPTINFO SET SALARY = SALARY + EMP..BONUS() が呼び出されると、BONUS メソッド内の SQL ステートメントは、EMPLOYEE 表を読み取ることができません。

• 特権

- ユーザー定義タイプの定義者は、構造タイプ用に自動的に生成されるすべてのメソッドおよび関数に対する EXECUTE 特権 WITH GRANT OPTION を常に受け取ります。EXECUTE 特権は、CREATE METHOD ステートメントを使用してメソッド本体が定義されない限り、CREATE TYPE ステートメントで明示的に指定されるメソッドに対しては付与されません。ユーザー定義タイプの定義者には、ALTER TYPE ステートメントを使用してメソッド指定をドロップする権利があります。CREATE DISTINCT TYPE の間に自動的に生成されるすべての関数での EXECUTE 特権は、PUBLIC に与えられます。
 - SQL ステートメントで外部メソッドを使用する場合は、メソッドの定義者は、メソッドが使用するどのパッケージに対しても EXECUTE 特権を持っている必要があります。
- 索引拡張を定義するには、NO SQL として定義されたルーチンしか使用できません (SQLSTATE 428F8)。

例:

例 1: 部門のタイプを作成します。

```
CREATE TYPE DEPT AS
  (DEPT_NAME VARCHAR(20),
   MAX_EMPS INT)
  REF USING INT
  MODE DB2SQL
```


CREATE TYPE (構造化)

例 2: 従業員タイプおよびマネージャー副タイプから構成される階層タイプを作成します。

```
CREATE TYPE EMP AS
  (NAME      VARCHAR(32),
   SERIALNUM INT,
   DEPT      REF(DEPT),
   SALARY    DECIMAL(10,2))
MODE DB2SQL

CREATE TYPE MGR UNDER EMP AS
  (BONUS     DECIMAL(10,2))
MODE DB2SQL
```

例 3: アドレスのタイプ階層を作成します。アドレスは、列のタイプとして使用するためのものです。インライン長は指定されていないので、DB2 がデフォルト長を計算します。該当のアドレスが、特定の入力アドレスにどのくらい近いかを計算する外部メソッドを、アドレス・タイプ定義内にカプセル化します。CREATE METHOD ステートメントを使ってメソッド本体を作成します。

```
CREATE TYPE address_t AS
  (STREET    VARCHAR(30),
   NUMBER    CHAR(15),
   CITY      VARCHAR(30),
   STATE     VARCHAR(10))
NOT FINAL
MODE DB2SQL
  METHOD SAMEZIP (addr address_t)
  RETURNS INTEGER
  LANGUAGE SQL
  DETERMINISTIC
  CONTAINS SQL
  NO EXTERNAL ACTION

  METHOD DISTANCE (address_t)
  RETURNS FLOAT
  LANGUAGE C
  DETERMINISTIC
  PARAMETER STYLE SQL
  NO SQL
  NO EXTERNAL ACTION

CREATE TYPE germany_addr_t UNDER address_t AS
  (FAMILY_NAME VARCHAR(30))
NOT FINAL
MODE DB2SQL

CREATE TYPE us_addr_t UNDER address_t AS
  (ZIP VARCHAR(10))
NOT FINAL
MODE DB2SQL
```

例 4: ネストされた構造タイプ属性をもつタイプを作成します。


```
CREATE TYPE PROJECT AS
  (PROJ_NAME VARCHAR(20),
   PROJ_ID INTEGER,
   PROJ_MGR MGR,
   PROJ_LEAD EMP,
   LOCATION ADDR_T,
   AVAIL_DATE DATE)
MODE DB2SQL
```

関連資料:

- *SQL* リファレンス 第 1 巻 の『基本述部』
- 1484 ページの『CREATE TABLE』
- *SQL* リファレンス 第 2 巻 の『SET PATH ステートメント』
- *SQL* リファレンス 第 1 巻 の『特殊レジスター』

関連サンプル:

- 『dtstruct.sqlC -- Create, use, drop a hierarchy of structured types and typed tables (C++)』

CREATE VIEW

CREATE VIEW ステートメントは、1 つまたは複数の表、ビュー、またはニックネームに基づくビューを作成します。

呼び出し:

このステートメントはアプリケーション・プログラムに組み込むことができ、また動的 SQL ステートメントを使用して出すことができます。 DYNAMICRULES の実行動作がパッケージに効力を持つ場合にのみ、動的に準備できる実行可能ステートメントです (SQLSTATE 42509)。

許可:

ステートメントの許可 ID によって保持されている特権には、少なくとも以下のいずれかが含まれていなければなりません。

- SYSADM または DBADM 権限、または
- 全選択に指定された表、ビュー、またはニックネームのそれぞれに対して、
 - その表またはビューに対する CONTROL 特権、または
 - その表またはビューに対する SELECT 特権

および以下の少なくとも 1 つ

- データベースに対する IMPLICIT_SCHEMA 権限 (ビューの暗黙または明示のスキーマ名が存在しない場合)

CREATE VIEW

- スキーマに対する CREATEIN 特権 (ビューのスキーマ名が既存のスキーマを指している場合)

副ビューを作成するには、このステートメントの許可 ID が以下の条件に適合している必要があります。

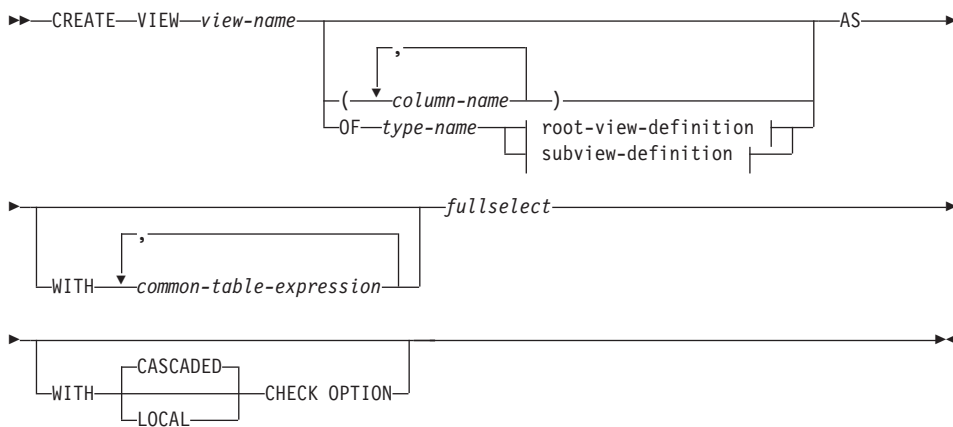
- 表階層のルート表の定義者と同じ名前である。
- 副ビューの基礎表に対して SELECT WITH GRANT 権限を持っているか、またはスーパービューがこのビューの定義者以外にはだれにも SELECT 権限を付与していない。

グループ特権は、CREATE VIEW ステートメントで指定された表やビューに対しては考慮されません。

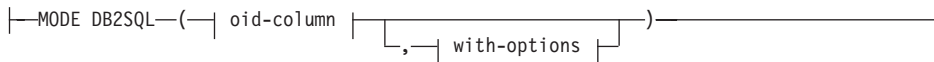
特権は、連合データベースのニックネームにビューを定義するときには考慮されません。このニックネームで示されている表またはビューのデータ・ソースの許可要件は、照会の処理時に適用されます。ステートメントの許可 ID は、別のリモート許可 ID へマップできます。

ビューの定義者が SYSADM 権限を持つために、ビューの作成しかできない場合、ビュー作成のため、その定義者には明示的な DBADM 権限が付与されます。

構文:



root-view-definition:



subview-definition:

```

|—MODE DB2SQL—| under-clause | (—| with-options |—) | —| EXTEND—|

```

oid-column:

```

|—REF IS—oid-column-name—USER GENERATED—| —| UNCHECKED—|

```

with-options:

```

|—column-name—WITH OPTIONS—| —| SCOPE—typed-table-name—| —| typed-view-name—| —| READ ONLY—|

```

under-clause:

```

|—UNDER—superview-name—INHERIT SELECT PRIVILEGES—|

```

説明:*view-name*

ビューの名前を指定します。暗黙または明示の修飾子を含む名前は、カタログに記述されている表、ビュー、ニックネーム、または別名を指定するものであってはなりません。修飾子は、SYSIBM、SYSCAT、SYSFUN、または SYSSTAT であってはなりません (SQLSTATE 42939)。

この名前は、作動不能なビューの名前と同じであっても構いません (1479 ページの『作動不能ビュー』を参照)。このような場合、作動不能なビューは、CREATE VIEW ステートメントに指定した新しいビューによって置き換えられます。作動不能なビューが置き換えられると、ユーザーに警告 (SQLSTATE 01595) が出されません。バインド・オプション SQLWARN を NO に設定してアプリケーションがバインドされた場合は、警告は戻されません。

column-name

ビューの列の名前を指定します。列名のリストを指定する場合、リスト中の列の名前の数は、全選択の結果表の列の数と同じ数でなければなりません。各 *column-name* (列名) は、固有で、しかも非修飾でなければなりません。列名のリストの指定がない場合、ビューの列は、全選択の結果表の列名を継承します。

全選択の結果表の列名が重複している場合、または無名の列がある場合には、列名のリストを指定する必要があります (SQLSTATE 42908)。無名列とは、定数、関数、式、またはセット演算から派生した列で、選択リストの AS 文節によって名前が指定されていない列を指します。

OF *type-name*

ビューの列が *type-name* で指定される構造タイプの属性に基づいていることを指定します。 *type-name* の指定にスキーマ名が含まれていない場合、そのタイプ名は

CREATE VIEW

SQL パス上のスキーマを探索することによって決まります (このパスは、静的 SQL の場合は FUNCPATH プリプロセス・オプションによって、動的 SQL の場合は CURRENT PATH レジスターによって定義されます)。ここに指定するタイプ名は、既存のユーザー定義タイプ名で (SQLSTATE 42704)、かつインスタンス化の可能な構造タイプでなければなりません (SQLSTATE 428DP)。

MODE DB2SQL

この文節は、型付きビューのモードを指定するために使用されます。これは、現在サポートされている唯一有効なモードです。

UNDER *superview-name*

このビューが *superview-name* の副ビューであることを指定します。スーパービューは既存のビューでなければならず (SQLSTATE 42704)、このビューは *type-name* のすぐ上位にある上位タイプである構造タイプで定義する必要があります (SQLSTATE 428DB)。 *view-name* と *superview-name* のスキーマ名は、同じでなければなりません (SQLSTATE 428DQ)。 *superview-name* で指定されるビューには、 *type-name* で既に定義された既存の副ビューを含めることはできません (SQLSTATE 42742)。

表の列には、スーパービューのオブジェクト ID 列が含まれています。オブジェクト ID 列のタイプは REF(*type-name*) に変更されており、 *type-name* の属性に基づく列が続きます (ここでいうタイプには、上位タイプの属性も含まれていることを念頭に置いてください)。

INHERIT SELECT PRIVILEGES

スーパービューに対して SELECT 特権を持つユーザーやグループはすべて、新しく作成した副ビューに対しても同様の特権を付与されます。この特権は、副ビュー定義者によって付与されたものと見なされます。

OID-column

型付きビューのオブジェクト ID 列を定義します。

REF IS *OID-column-name* USER GENERATED

オブジェクト ID (OID) 列をビューの最初の列として定義することを指定します。ビュー階層のルート・ビューには、OID が必須です (SQLSTATE 428DX)。このビューは副ビュー以外の型付きビュー (OF 文節が必須) でなければなりません (SQLSTATE 42613)。この列の名前は *OID-column-name* という形式で定義されますが、構造タイプ *type-name* のいかなる属性の名前とも同一であることはできません (SQLSTATE 42711)。 *fullselect* で指定した最初の列は、REF(*type-name*) というタイプでなければなりません (キャストして適切なタイプにする必要があるかもしれません)。UNCHECKED を指定しない場合、索引 (主キー、ユニーク制約、ユニーク索引、または OID 列) を使用して固有性を強制できる列 (NULL 可能ではない) に基づいている必要があります。この列をオブジェクト ID 列 または OID 列 といいます。USER GENERATED というキーワードは、行を挿入する際にユーザーが OID 列の初

期値を提供しなければならないことを指しています。行を挿入した後は、OID 列を更新することはできません (SQLSTATE 42808)。

UNCHECKED

固有であることをシステムが証明できない場合でも、型付きビュー定義のオブジェクト ID の列を固有であると見なすように定義します。この属性は、以下のような型付きビュー階層に定義されている表またはビューでの使用を想定しています。すなわち、そのデータが固有性規則に準拠しているものの、システムが固有性を証明できる規則には準拠していないことをユーザーが認識しているという場合です。UNCHECKED オプションは、複数の階層や従来型の表またはビューにまでわたっているビューの階層に必須のオプションです。UNCHECKED を指定する場合、ユーザーの責任でビューの各行に固有の OID が確実にあるようにします。ユーザーがこの特性を保証しなかったために、ビューに重複した OID 値が入ってしまうと、固有でない OID 値のいずれかを含むパスの式または Deref 演算子はエラーになります (SQLSTATE 21000)。

with-options

型付きビューの列に適用される追加オプションを定義します。

column-name WITH OPTIONS

追加オプションを指定する列の名前を指定します。 *column-name* は、ビューの *type-name* に定義されている (継承されてはいない) 属性名に対応していなければなりません。この列は参照タイプである必要があります (SQLSTATE 42842)。また、すでにスーパービューに存在する列に対応することはできません (SQLSTATE 428DJ)。列名は、ステートメント内の 1 つの WITH OPTIONS SCOPE 文節に 1 回しか指定できません (SQLSTATE 42613)。

SCOPE

参照タイプ列の有効範囲を指定します。参照解除演算子の左オペランド、または Deref 関数の引き数として使用する列には、すべて有効範囲を指定する必要があります。

ターゲット表またはターゲット・ビューが定義するために、後続する ALTER VIEW ステートメント (有効範囲が継承されていない場合) まで、参照タイプ列の有効範囲指定を遅らせることができます (通常は、相互参照表および相互参照ビューの場合に適用する)。ビューの参照タイプ列で有効範囲が指定されていないのに、基礎表またはビュー列の有効範囲が指定された場合、基礎列の有効範囲は参照タイプ列によって継承されます。基礎表またはビューの列に有効範囲がない場合には、この列に有効範囲は指定されません。読み取り専用カーソルと更新可能カーソルの詳細については、1477 ページを参照してください。

typed-table-name

型付き表の名前。この表は既に存在しているものか、作成する表と同じ名前のものでなければなりません (SQLSTATE 42704)。 *column-name* のデータ・タイプは REF(S) でなければなりません。 S は *typed-table-name* の

CREATE VIEW

タイプを表します (SQLSTATE 428DM)。値が *typed-table-name* の既存行を実際に参照していることを確認するための、*column-name* の既存値の検査は行われません。

typed-view-name

型付きビューの名前。このビューは既に存在しているものか、作成するビューと同じ名前のものでなければなりません (SQLSTATE 42704)。*column-name* のデータ・タイプは REF(S) でなければなりません。S は *typed-view-name* のタイプを表します (SQLSTATE 428DM)。値が *typed-view-name* の既存行を実際に参照していることを確認するための、*column-name* の既存値の検査は行われません。

READ ONLY

列を読み取り専用列として指定します。このオプションは、列を読み取り専用に指定し、副ビュー定義は、同じ列の式を暗黙的に読み取り専用に指定することができます。

AS ビュー定義を指定します。

WITH *common-table-expression*

後続の全選択で使用される共通表式を定義します。型付きビューを定義するときには、共通表式は指定できません。

全選択 (*fullselect*)

ビューを定義します。どのような場合でも、ビューは、SELECT が実行されたとしたらその結果となるような行で構成されます。全選択でホスト変数、パラメーター・マーカ、または宣言済み一時表を参照することはできません。ただし、パラメーター化されたビューを SQL 表関数として作成することは可能です。

型付きビューおよび副ビューの場合：*fullselect* は、以下の規則に準拠していなければなりません。そうでない場合、エラーが戻されます (SQLSTATE 428EA 何も指定されていない場合)。

- 全選択に、DBPARTITIONNUM または HASHEDVALUE 関数、非 deterministic 関数、または外部アクションを持つように定義されている関数への参照を含めることはできません。
- ビューの本体は、単一の副選択か複数の副選択の UNION ALL で構成する必要があります。ビューの本体に直接加わっている各副選択を、ビューの分岐と呼びます。ビューには、1 つかそれ以上の分岐があります。
- 各分岐の FROM 文節は、単一の表またはビュー (その分岐の基礎表またはビューといい、必ずしも型付きではない) で構成される必要があります。
- 各分岐の基礎表またはビューは、別々の階層にする必要があります (つまり、ビューは、同じ階層内の基礎表またはビューが付いた複数の分岐を持つことはできません)。
- 型付きビュー定義の分岐はいずれも GROUP BY または HAVING を指定できません。

- ビューの本体に UNION ALL が含まれる場合、階層内にあるルート・ビューの OID 列に UNCHECKED オプションを指定する必要があります。

ビューおよび副ビューの階層の場合：BR1 および BR2 が、階層内のビュー定義に現れる分岐になるようにします。T1 を BR1 の基礎表またはビューに、T2 を BR2 の基礎表またはビューにします。この場合は以下のようになります。

- T1 および T2 が同じ階層でない場合、ビュー階層にあるルート・ビューの OID 列に UNCHECKED オプションを指定する必要があります。
- T1 および T2 が同じ階層にある場合、行セットが結合しないことを十分保証する述部または ONLY 文節を、BR1 および BR2 に含める必要があります。

EXTEND AS を使って定義された型付きのビューの場合：副ビューの本体内の各分岐について：

- 各分岐の基礎表は、即時スーパービューのいくつかの基礎表の副表（必ずしも適切ではない）でなければなりません。
- SELECT リストの式は、副ビューの非継承列に割り当てることができるものでなければなりません (SQLSTATE 42854)。

AS (EXTEND なし) を使って定義された型付き副ビューの場合：

- 副ビューの本体内にあるそれぞれの分岐について、SELECT リストにある式は、副ビューの継承列と非継承列の宣言済みタイプに割り当てられるようにする必要があります (SQLSTATE 42854)。
- 副ビューで指定した階層上のそれぞれの分岐の OID 式は、ルート・ビュー内の同じ階層上の分岐の OID 式と同じでなければなりません (キャスト以外)。
- スーパービュー内の READ ONLY として (暗黙的または明示的に) 指定されていない列の式は、その副ビュー内の同じ基礎階層上のすべての分岐と同じでなければなりません。

WITH CHECK OPTION

ビューによって挿入または更新される行すべてが、ビューの定義にしたがっていないければならないという制約を指定します。ビューの定義に従わない行とは、ビューの検索条件を満たしていない行です。

WITH CHECK OPTION は、以下のいずれかの条件が真である場合には指定できません。

- ビューが読み取り専用である場合 (SQLSTATE 42813)。挿入が許されていない更新可能なビューに対して WITH CHECK OPTION を指定すると、制約は更新にのみ適用されます。
- ビューが、NODENUMBER または PARTITION 関数、非 deterministic 関数、または外部アクションを伴う関数を参照する場合 (SQLSTATE 42997)。
- ニックネームがビューの更新の対象である場合。

CREATE VIEW

- ビューの更新の対象である INSTEAD OF トリガーが定義されているビューである場合 (SQLSTATE 428FQ)。

WITH CHECK OPTION を省略すると、ビューを使用するどのような挿入操作または更新操作のチェックにおいても、ビューの定義は使用されません。ただし、ビューが WITH CHECK OPTION が指定された他のビューに直接または間接的に従属する場合には、挿入操作または更新操作の過程で、何らかのチェックが行われる場合があります。この場合、ビューの定義が使用されるわけではないため、ビューの定義にしたがっていないビューを介して、行が挿入または更新される可能性があります。

CASCADED

ビュー *V* に対する WITH CASCADED CHECK OPTION 制約は、*V* が従属するいずれかの更新可能ビューから、制約としての検索条件を *V* が継承することを意味します。さらに、*V* に従属するすべての更新可能ビューも、このような制約の対象になります。したがって、*V* の検索条件と、*V* が従属している各ビューの検索条件との AND を取ったものが、*V* あるいは *V* に従属するいずれかのビューの挿入または更新に対して適用される制約となります。

LOCAL

ビュー *V* に対する WITH LOCAL CHECK OPTION 制約は、*V* の検索条件が、*V* または *V* に従属するいずれかのビューの挿入あるいは更新に対する制約として適用されることを意味しています。

以下の例は、CASCADED と LOCAL の差異を示しています。以下のような更新可能なビューを想定します (Y は、一覧表の見出しに示しているように、LOCAL または CASCADED に置き換えます)。

```
V1 defined on table T
V2 defined on V1 WITH Y CHECK OPTION
V3 defined on V2
V4 defined on V3 WITH Y CHECK OPTION
V5 defined on V4
```

以下の表は、挿入または更新された行をチェックするのに使われる検索条件を示しています。

	Y が LOCAL の場合	Y が CASCADED の場合
V1 でのチェック条件:	対象となるビューなし	対象となるビューなし
V2 でのチェック条件:	V2	V2、V1
V3 でのチェック条件:	V2	V2、V1
V4 でのチェック条件:	V2、V4	V4、V3、V2、V1
V5 でのチェック条件:	V2、V4	V4、V3、V2、V1

また、以下のような更新可能ビューについても考えてみます。これは、デフォルトの `CASCADED` オプションを使用した場合の `WITH CHECK OPTION` の効果を示しています。

```
CREATE VIEW V1 AS SELECT COL1 FROM T1 WHERE COL1 > 10

CREATE VIEW V2 AS SELECT COL1 FROM V1 WITH CHECK OPTION

CREATE VIEW V3 AS SELECT COL1 FROM V2 WHERE COL1 < 100
```

以下の `INSERT` ステートメントは `V1` を使用するものですが、`V1` に `WITH CHECK OPTION` が指定されておらず、また `V1` が、`WITH CHECK OPTION` の指定された他のどのビューにも従属していないため、このステートメントは成功します。

```
INSERT INTO V1 VALUES(5)
```

以下の `INSERT` ステートメントは `V2` を使用するものですが、`V2` に `WITH CHECK OPTION` が指定されており、挿入 (`INSERT`) によって `V2` の定義にしたがっていない行が作成されるため、このステートメントはエラーになります。

```
INSERT INTO V2 VALUES(5)
```

以下の `INSERT` ステートメントでは `V3` を使用しています。`V3` に `WITH CHECK OPTION` は指定されていませんが、これは、`WITH CHECK OPTION` の指定された `V2` の従属であるため、エラーになります (`SQLSTATE 44000`)。

```
INSERT INTO V3 VALUES(5)
```

以下の `INSERT` ステートメントも、`V3` を使用しています。これは `V3` の定義に準拠していませんが、成功します (`V3` には `WITH CHECK OPTION` が指定されていません)。これは、`WITH CHECK OPTION` の指定された `V2` の定義に従ったものになっています。

```
INSERT INTO V3 VALUES(200)
```

注:

- **互換性**
 - 以前のバージョンの DB2 との互換性:
 - `FEDERATED` キーワードは、キーワード `CREATE` および `VIEW` の間に指定できます。しかし `FEDERATED` キーワードは無視されます。これは連合オブジェクトがビュー定義で使用される場合は、警告は戻されないからです。
- まだ存在していないスキーマ名を用いてビューを作成すると、ステートメントの許可 ID に `IMPLICIT_SCHEMA` 権限がある場合に限り、そのスキーマが暗黙的に作成されます。そのスキーマの所有者は `SYSIBM` になります。スキーマに対する `CREATEIN` 特権が `PUBLIC` に付与されます。

CREATE VIEW

- ビュー列は、列が式から派生するとき以外は、基本表またはビューの NOT NULL WITH DEFAULT 属性を継承します。基本表に制約が定義されている場合、更新可能ビューに行が挿入または更新される時、それらの制約 (主キー、参照完全性、およびチェック制約) に対するチェックが行われます。
- 定義の中で作動不能ビューを使用して新しいビューを作成することはできません (SQLSTATE 51024)。
- このステートメントでは、宣言済み一時表をサポートしていません (SQLSTATE 42995)。
- 削除可能ビュー:** 削除操作の INSTEAD OF トリガーがビューに定義されている場合、または以下のいずれかが当てはまる場合、ビューは削除可能です。
 - 外部全選択の各 FROM 文節には、基本表 (OUTER 文節なし)、削除可能ビュー (OUTER 文節なし)、削除可能なネストした表式、または削除可能な共通表式 (ニックネームが識別不能) のいずれかが 1 つだけ指定されている
 - 外部全選択に VALUES 文節が含まれない
 - 外部全選択に GROUP BY 文節も HAVING 文節も含まれない
 - 外部全選択の選択リストに列関数が含まれない
 - 外部全選択に、UNION ALL を除くセット演算 (UNION、EXCEPT、または INTERSECT) が含まれない
 - UNION ALL のオペランドの基本表が同じ表ではなく、各オペランドが削除可能
 - 外部全選択の選択リストに DISTINCT が含まれない
- 更新可能ビュー:** 更新操作の INSTEAD OF トリガーがビューに定義されている場合、または以下のいずれかが当てはまる場合、ビューの列は更新可能です。
 - ビューが削除可能 (削除の INSTEAD OF トリガーとは無関係) であり、列の解決結果が基本表の列 (非参照操作は使用しない) となり、READ ONLY オプションが指定されない
 - ビューの全選択に UNION ALL が含まれる場合、UNION ALL のオペランドの対応するすべての列のデータ・タイプおよびデフォルト値が正確に一致する (長さ、または精度と位取りを含む)

ビューのいずれかの列が更新可能なら、ビューは更新可能です。

- 挿入可能ビュー:**
 - 挿入操作の INSTEAD OF トリガーがビューに定義されている場合、またはビューの少なくとも 1 つの列が更新可能であり、ビューの全選択に UNION ALL が含まれない場合、そのビューは挿入可能です。
 - 特定の行が基本表のうちの正確に 1 つのチェック制約を満たしている場合のみ、その行をビュー (UNION ALL を含む) に挿入できます。
 - 更新不可の列が含まれているビューに挿入するには、それらの列を列リストから除外する必要があります。

- **読み取り専用ビュー:** ビューが読み取り専用 であるのは、削除可能でも、更新可能でも、挿入可能でもない 場合です。

ビューが読み取り専用かどうかは、SYSCAT.VIEWS カタログ・ビューの READONLY 列に示され、INSTEAD OF トリガーが考慮されることはありません。

- 共通表式とネストした表式は、削除可能かどうか、更新可能かどうか、挿入可能かどうか、または読み取り専用かどうかを判別する上で、同じ一連の規則に従います。
- **作動不能ビュー:** 作動不能ビュー とは、SQL ステートメントで使用できなくなったビューのことです。ビューは、以下の場合に作動不能になります。
 - ビュー定義が従属している特権が取り消された場合
 - ビュー定義が従属している表、ニックネーム、別名、または関数などのオブジェクトがドロップされた場合
 - ビュー定義が従属しているビューが作動不能になった場合
 - ビュー定義のスーパービューであるビュー (副ビュー) が作動不能になった場合

実際には、作動不能ビューとは、ビュー定義が間違っただけでドロップされたビューです。たとえば、別名がドロップされると、その別名を使用して定義されているビューすべてが作動不能になります。それに従属するすべてのビューも作動不能になり、そのビューに従属するパッケージは無効になります。

作動不能ビューを明示的に作成し直すか、あるいはドロップされるまで、その作動不能ビューを使用するステートメントのコンパイルはできません (SQLSTATE 51024)。ただし、CREATE ALIAS、CREATE VIEW、DROP VIEW、および COMMENT ON TABLE の各ステートメントは例外です。作動不能ビューが明示的にドロップされるまで、その修飾名を使って別の表や別名を作成することはできません (SQLSTATE 42710)。

作動不能ビューは、作動不能ビューの定義テキストを使用して、CREATE VIEW ステートメントを発行することにより、再作成することができます。このビュー定義テキストは、SYSCAT.VIEWS カタログの TEXT 列に保管されます。作動不能ビューを再作成する場合は、他のユーザーがそのビューに対して必要となる特権すべてを明示的に付与する必要があります。これは、ビューが作動不能と見なされると、ビューのすべての許可レコードが削除されるためです。作動不能ビューを再作成するために、それを明示的に削除する必要はありません。作動不能ビューと同じ *view-name* を指定して CREATE VIEW ステートメントを発行すると、作動不能ビューは置き換えられ、CREATE VIEW ステートメントは警告を戻します (SQLSTATE 01595)。

作動不能ビューであることは、SYSCAT.VIEWS カタログ・ビューの VALID 列が X、また SYSCAT.TABLES カタログ・ビューの STATUS 列が X であることによって示されます。

- **特権**

CREATE VIEW

ビューの定義者は、ビューに対する SELECT 特権と、ビューをドロップする権利を常に与えられます。ビューの定義者は、その定義者が全選択で指定されたすべての基本表、ビューまたはニックネームに対する CONTROL 特権を持っている場合、あるいは定義者が SYSADM 権限または DBADM 権限を持っている場合にのみ、そのビューに対する CONTROL 特権が付与されます。

ビューの定義者は、そのビューが読み取り専用でなく、定義者が基礎となるオブジェクトに対して対応する特権を持っている場合に、そのビューに対する INSERT、UPDATE、列レベルの UPDATE、または DELETE の特権を与えられます。

ビューの定義者にそれらの特権が与えられるのは、それらの特権の派生元の特権がビューの作成時に存在している場合に限りです。定義者は、これらの特権を直接持っているか、または PUBLIC の特権として持っていることが必要です。特権は、連合データベースのニックネームにビューを定義するときには考慮されません。ただし、ニックネームにビューを使用する場合、ユーザーの許可 ID には、そのニックネームがデータ・ソースで参照する表またはビューに対する適切な選択特権がなければなりません。もしその特権がなければ、エラーが戻されます。ビューの定義者がメンバーであるグループを持つ特権は考慮されません。

副ビューが作成されると、すぐ上のスーパービューに対して持っている SELECT 特権が自動的に副ビューに対しても付与されます。

• 有効範囲および REF 列

ビュー定義の全選択で参照タイプ列を選択する際には、必要なターゲット・タイプと有効範囲について考慮してください。

- 必要なターゲット・タイプおよび有効範囲が基礎表または基礎ビューのものと同じ場合には、参照列をそのまま選択することができます。
- 有効範囲を変更する必要がある場合には、WITH OPTIONS SCOPE 文節を使って、必要な有効範囲の表とビューを定義します。
- 参照のターゲット・タイプを変更する必要がある場合には、まず列を参照の対象となっている参照タイプにキャストしてから、新規の参照タイプへもキャストする必要があります。この場合、有効範囲は参照タイプへのキャストで指定できますし、WITH OPTIONS SCOPE 文節を使っても指定できます。たとえば、REF(TYP1) SCOPE TAB1 として定義された Y 列を選択したとしましょう。そして、この列を REF(VTYP1) SCOPE VIEW1 として定義するとします。この場合、選択リスト項目は以下のようになります。

```
CAST(CAST(Y AS VARCHAR(16) FOR BIT DATA) AS REF(VTYP1) SCOPE VIEW1)
```

- **ID 列:** ビューの列は、ビュー定義の全選択内にある対応する列のエレメントが表の ID 列の名前であるか、基本表の ID 列の名前に直接または間接的にマップされたビューの列の名前である場合に ID 列と見なされます。

これ以外の場合にはすべて、ビューの列は ID のプロパティを取得しません。たとえば:

- ビュー定義の選択リストに ID 列の名前のインスタンスが複数含まれている (つまり、同じ列を複数回選択している) 場合

- ビュー定義に結合が関与している場合
- ビュー定義の列に ID 列を参照する式が含まれている場合
- ビュー定義に UNION が含まれている場合

挿入先のビューにおいて、ビュー定義の選択リストに直接または間接的に基本表の ID 列の名前が含まれている場合は、INSERT ステートメントが基本表の ID 列を直接参照する場合と同じ規則が適用されます。

- **連合ビュー:** 連合ビューとは、全選択内にあるいずれかのニックネームへの参照を含むビューのことです。この種のニックネームが存在する場合、そのビューで使用する許可モデルは、ビューが照会で順番に参照されるときに変更されます。

ビューを作成しても、ビュー定義者が基礎データ・ソース表またはビューにアクセスできるかどうかを判別する特権検査は行われません。連合データベースでの表またはビューに対する特権検査は、最低でもそうしたオブジェクトに対する SELECT 特権をビュー定義者に要求することにより、通常どおり行われます。

連合ビューが照会で順番に参照される場合、その照会を発行したデータ・ソースおよび許可 ID (または、その照会がマッピングするリモート許可 ID) に対する照会になるニックネームには、データ・ソース表またはビューにアクセスするのに必要な特権がなければなりません。連合ビューを参照する照会を発行する許可 ID には、統合サーバーに存在する (連合でない) 表またはビューに対する追加の特権は必要ありません。

例:

例 1: PROJECT 表に基づくビュー MA_PROJ を作成します。このビューには、文字 'MA' で始まるプロジェクト番号 (PROJNO) を持つ行だけを入れます。

```
CREATE VIEW MA_PROJ AS SELECT *
FROM PROJECT
WHERE SUBSTR(PROJNO, 1, 2) = 'MA'
```

例 2: 例 1 と同様にビューを作成します。ただし、プロジェクト番号 (PROJNO)、プロジェクト名 (PROJNAME)、およびプロジェクトの責任者 (RESPEMP) の列だけを選択します。

```
CREATE VIEW MA_PROJ
AS SELECT PROJNO, PROJNAME, RESPEMP
FROM PROJECT
WHERE SUBSTR(PROJNO, 1, 2) = 'MA'
```

例 3: 例 2 と同様のビューを作成します。ただし、ビューの中でプロジェクトの責任者の列を IN_CHARGE と呼びます。

```
CREATE VIEW MA_PROJ
(PROJNO, PROJNAME, IN_CHARGE)
AS SELECT PROJNO, PROJNAME, RESPEMP
FROM PROJECT
WHERE SUBSTR(PROJNO, 1, 2) = 'MA'
```

CREATE VIEW

注: 列名のいずれか 1 つだけを変更する場合でも、ビューの 3 つの列すべての名前を MA_PROJ の後の括弧内に指定する必要があります。

例 4: PRJ_LEADER という名前のビューを作成します。このビューには、PROJECT 表の最初の 4 つの列 (PROJNO、 PROJNAME、 DEPTNO、 RESPEMP) と、プロジェクトの責任者 (RESPEMP) のラストネーム (LASTNAME) を入れます。ラストネームは、EMPLOYEE 表の EMPNO を PROJECT 表の RESPEMP と突き合わせることで、EMPLOYEE 表から入手します。

```
CREATE VIEW PRJ_LEADER
AS SELECT PROJNO, PROJNAME, DEPTNO, RESPEMP, LASTNAME
FROM PROJECT, EMPLOYEE
WHERE RESPEMP = EMPNO
```

例 5: 例 4 と同様のビューを作成します。ただし、列 PROJNO、 PROJNAME、 DEPTNO、 RESPEMP、および LASTNAME に加えて、担当者の給与総額 (SALARY + BONUS + COMM) を入れます。また、平均スタッフ数 (PRSTAFF) が 1 より大きいプロジェクトだけを選択します。

```
CREATE VIEW PRJ_LEADER
(PROJNO, PROJNAME, DEPTNO, RESPEMP, LASTNAME, TOTAL_PAY )
AS SELECT PROJNO, PROJNAME, DEPTNO, RESPEMP, LASTNAME, SALARY+BONUS+COMM
FROM PROJECT, EMPLOYEE
WHERE RESPEMP = EMPNO
AND PRSTAFF > 1
```

全選択に、TOTAL_PAY として式 SALARY+BONUS+COMM を指定することによって、以下のように、列名リストの指定を省略することができます。

```
CREATE VIEW PRJ_LEADER
AS SELECT PROJNO, PROJNAME, DEPTNO, RESPEMP,
LASTNAME, SALARY+BONUS+COMM AS TOTAL_PAY
FROM PROJECT, EMPLOYEE
WHERE RESPEMP = EMPNO AND PRSTAFF > 1
```

例 6: 以下の図に示す表とビューがあると想定します。

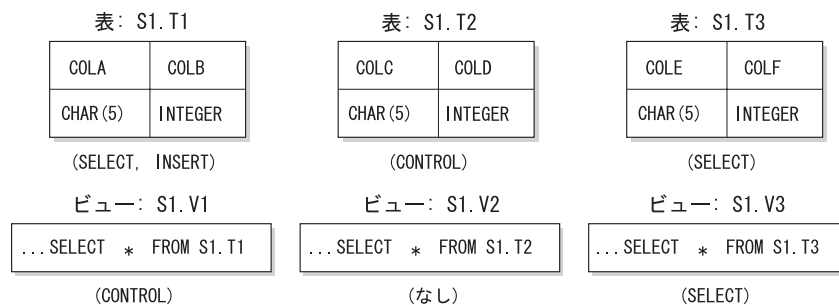


図 18. 例 6 の表とビュー

ユーザー ZORPIE (DBADM または SYSADM のいずれの権限も持たない) は、各オブジェクトの下の括弧内に示している権限を与えられています。

1. 以下のステートメントによって作成するビューに対して、ZORPIE は CONTROL 特権を獲得します。

```
CREATE VIEW VA AS SELECT * FROM S1.V1
```

これは、ZORPIE が S1.V1 に対して CONTROL 権限を持っているからです。DBADM または SYSADM のいずれかの権限を持つユーザーが、S1.V1 に対する CONTROL 権限を ZORPIE に与えている必要があります。基礎となる基本表に対して、どのような特権を与えられているかは関係ありません。

2. ZORPIE は、以下のようなビューの作成は許されません。

```
CREATE VIEW VB AS SELECT * FROM S1.V2
```

これは、ZORPIE には、S1.V2 に対する CONTROL も SELECT も与えられていないからです。基礎となる基本表 (S1.T2) に対して CONTROL を与えられているかどうかは、関係ありません。

3. 以下のステートメントによって作成するビューに対して、ZORPIE は CONTROL 特権を獲得します。

```
CREATE VIEW VC (COLA, COLB, COLC, COLD)
AS SELECT * FROM S1.V1, S1.T2
WHERE COLA = COLC
```

これは、ZORPIE.VC の全選択では、ビュー S1.V1 と S1.T2 を参照しており、ZORPIE はその両方に対する CONTROL を持っているからです。ビュー VC は読み取り専用で、INSERT、UPDATE、または DELETE のいずれの権限も ZORPIE には与えられないことに注意してください。

4. 以下のステートメントによって作成するビューに対して、ZORPIE は SELECT 特権を入手します。

```
CREATE VIEW VD (COLA,COLB, COLE, COLF)
AS SELECT * FROM S1.V1, S1.V3
WHERE COLA = COLE
```

これは、ZORPIE.VD の全選択で 2 つのビュー S1.V1 および S1.V3 を参照しており、ZORPIE はその 1 つに対しては SELECT 特権を、もう 1 つに対しては CONTROL 特権を与えられているからです。ZORPIE.VD に対する特権として、2 つの特権のうち低い方の特権である SELECT が ZORPIE に与えられます。

5. 以下のビュー定義では、ZORPIE はビュー VE に対して WITH GRANT OPTION を伴う INSERT、UPDATE および DELETE の各特権と、SELECT 特権を与えられます。

```
CREATE VIEW VE
AS SELECT * FROM S1.V1
WHERE COLA > ANY
(SELECT COLE FROM S1.V3)
```


CREATE VIEW

ZORPIE の VE に対する特権は、主として S1.V1 に対する特権によって決定されます。S1.V3 は副照会で参照されているだけなので、ビュー VE の作成には S1.V3 に対する SELECT 特権だけが必要です。ビューの定義者は、ビューの定義で参照されているすべてのオブジェクトに対して CONTROL を持っている場合のみ、そのビューに対して CONTROL を入手します。ZORPIE は S1.V3 に対する CONTROL を持っていないので、VE に対する CONTROL は与えられません。

関連概念:

- SQL リファレンス 第 1 巻の『照会』

関連資料:

- 1552 ページの『CREATE FUNCTION (SQL スカラー、表、または行)』

CREATE TABLE

CREATE TABLE ステートメントは表を定義します。定義には、その表の名前と、その列の名前および属性を含める必要があります。定義には、主キーやチェック制約など、表の他の属性を含めることができます。

グローバル一時表を宣言するには、DECLARE GLOBAL TEMPORARY TABLE ステートメントを使用します。

呼び出し:

このステートメントはアプリケーション・プログラムに組み込むことができ、また動的 SQL ステートメントを使用して出すことができます。DYNAMICRULES の実行動作がパッケージに効力を持つ場合にのみ、動的に準備できる実行可能ステートメントです (SQLSTATE 42509)。

許可:

ステートメントの許可 ID によって保持されている特権には、少なくとも以下のいずれかが含まれていなければなりません。

- SYSADM または DBADM 権限
- データベースに対する CREATETAB 権限および表スペースに対する USE 特権に加えて、以下のいずれか。
 - データベースに対する IMPLICIT_SCHEMA 権限 (表の暗黙的または明示的スキーマ名が存在しない場合)
 - スキーマに対する CREATEIN 特権 (表のスキーマ名が既存のスキーマを指す場合)

副表を定義する場合には、許可 ID は表階層のルート表の定義者と同じでなければなりません。

外部キーを定義する場合には、ステートメントの許可 ID が保持する特権として、親表に対する以下のいずれかが必要になります。

- その表に対する REFERENCES 特権
- 指定の親キーのそれぞれの列に対する REFERENCES 特権
- 表に対する CONTROL 特権
- SYSADM または DBADM 権限

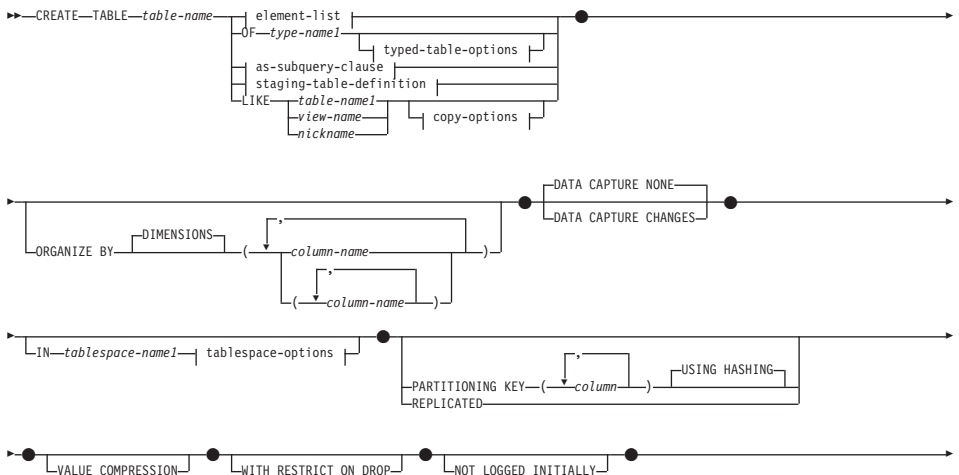
(全選択を使用して) マテリアライズ照会表を定義するには、このステートメントの許可 ID に、全選択で識別された個々の表またはビューに対する以下の特権が少なくとも 1 つ含まれている必要があります。

- 表またはビューに対する SELECT 特権と、 REFRESH DEFERRED または REFRESH IMMEDIATE が指定されている場合には ALTER 特権
- 表またはビューに対する CONTROL 特権
- SYSADM または DBADM 権限

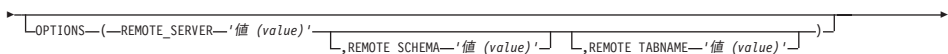
マテリアライズ照会表に関連付けられたステーキング表を定義するには、ステートメントの許可 ID に、以下の特権の少なくとも 1 つが含まれている必要があります。

- マテリアライズ照会表での CONTROL 特権または ALTER 特権、なおかつ、マテリアライズ照会表での全選択で識別された個々の表またはビューに対する以下の特権が少なくとも 1 つ含まれている必要があります。
 - その表またはビューに対する SELECT 特権および ALTER 特権
 - 表またはビューに対する CONTROL 特権
- SYSADM または DBADM 権限

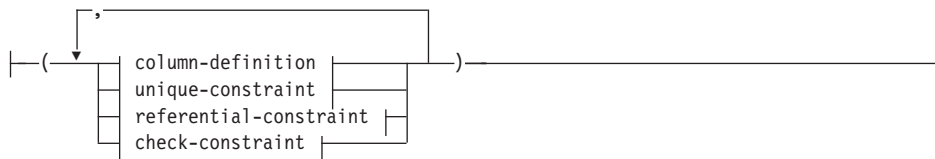
構文:



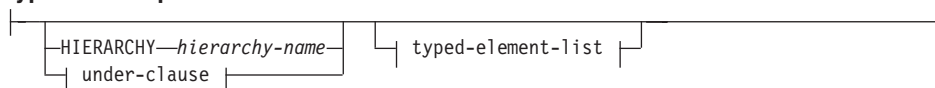
CREATE TABLE



element-list:



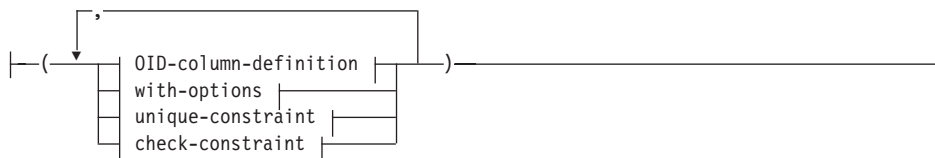
typed-table-options:



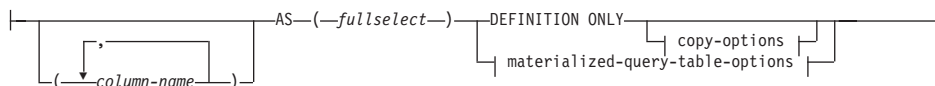
under-clause:



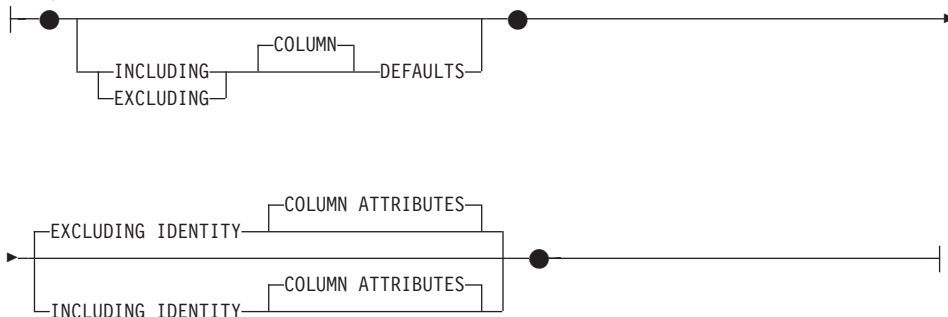
typed-element-list:



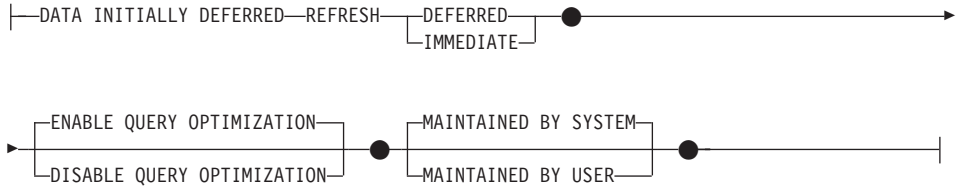
as-subquery-clause:



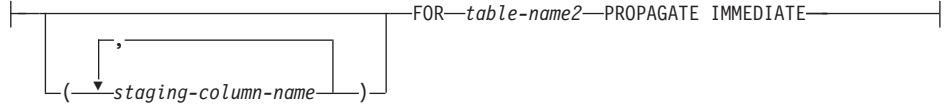
copy-options:



materialized-query-table-options:



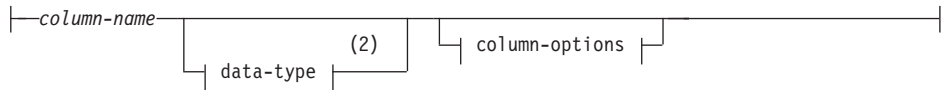
staging-table-definition:



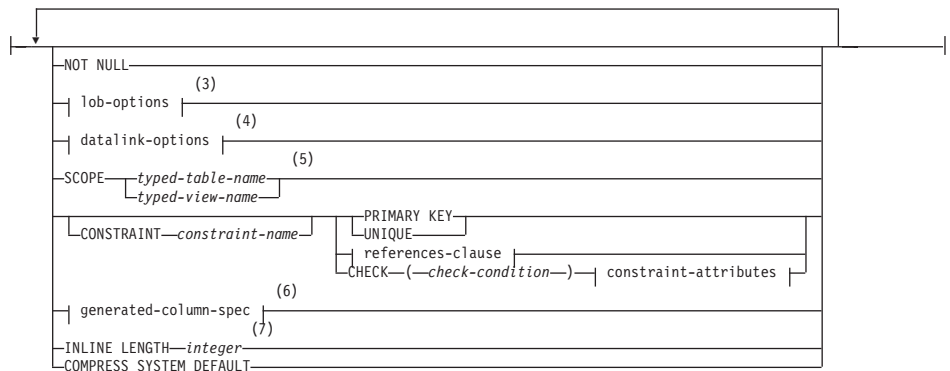
tablespace-options:



column-definition:



column-options:



注:

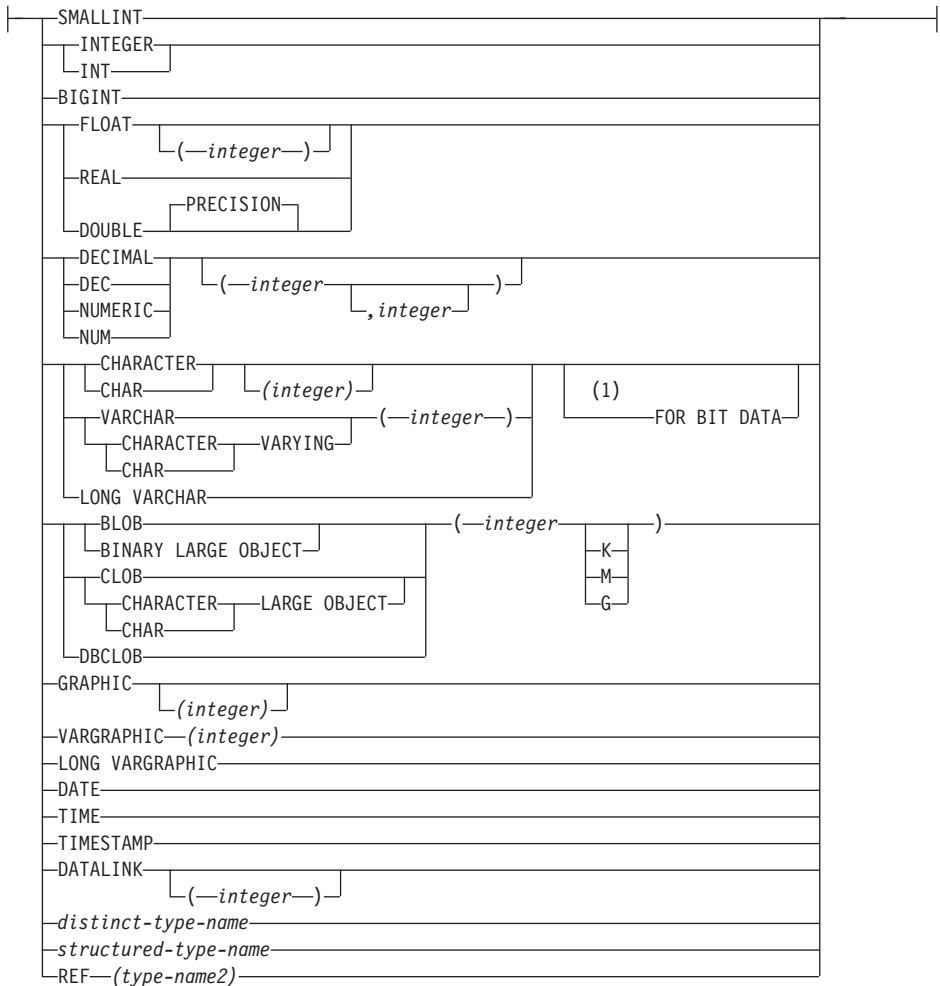
- 1 表の索引をどの表スペースに含めるかを指定できるのは、表を作成する場合だけです。
- 2 選択する最初の column-option が、generation-expression を指定した

CREATE TABLE

generated-column-spec の場合、data-type を省略することができます。これは、generation-expression の処理結果のデータ・タイプから判別されます。

- 3 lob-options (LOB オプション) 文節は、ラージ・オブジェクト・タイプ (BLOB、CLOB、および DBCLOB) と、ラージ・オブジェクト・タイプに基づく特殊タイプに対してのみ適用されます。
- 4 datalink-options 文節は、DATALINK タイプと、DATALINK タイプに基づく特殊タイプに対してのみ適用されます。
- 5 SCOPE 文節は REF タイプに対してのみ適用されます。
- 6 IDENTITY 列属性は、複数のパーティションをもつデータベースでは サポートされません。
- 7 INLINE LENGTH は、構造タイプとして定義された列に対してのみ用います。

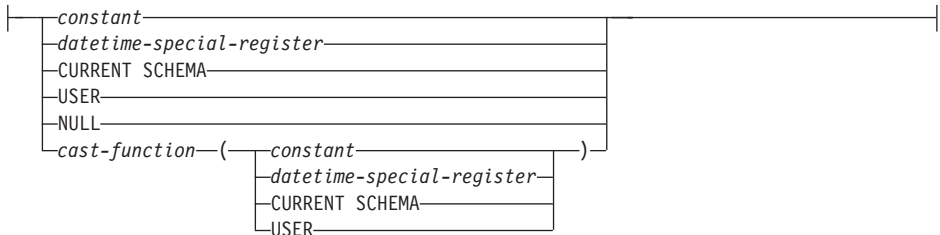
data-type:



注:

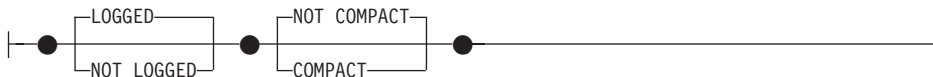
- 1 FOR BIT DATA 文節とその後に続く他の列制約とは、任意の順序で指定できます。

default-values:

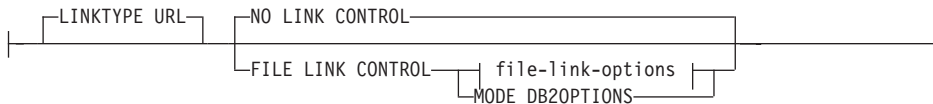


CREATE TABLE

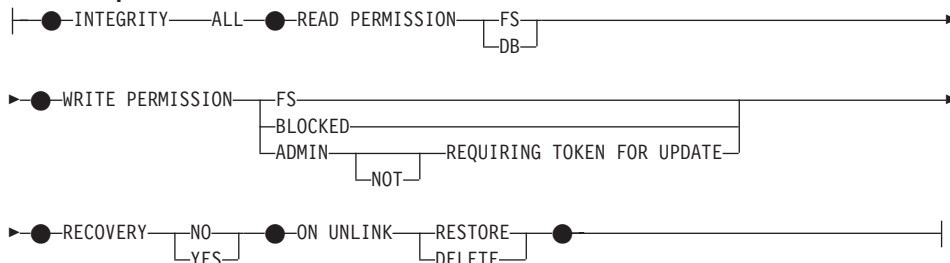
lob-options:



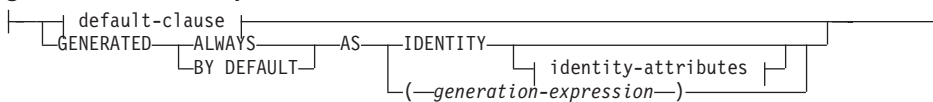
datalink-options:



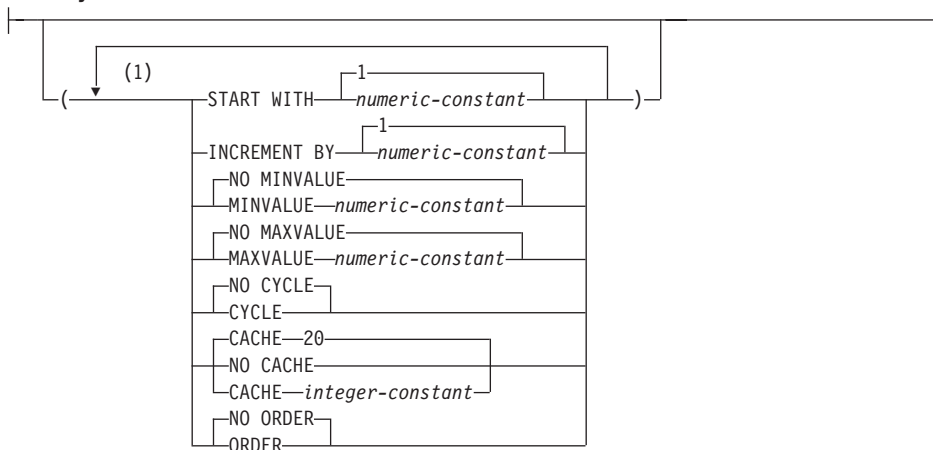
file-link-options:



generated-column-spec:



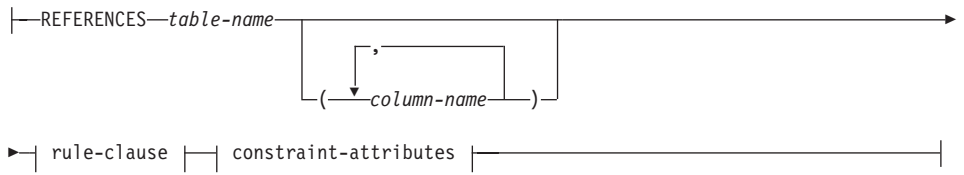
identity-attributes:



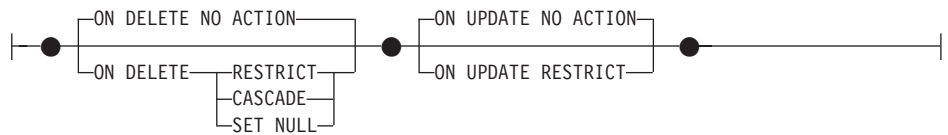
注:

- 1 同じ文節を複数回指定することはできません。

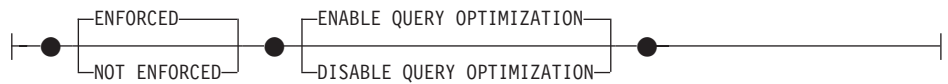
references-clause:



rule-clause:



constraint-attributes:



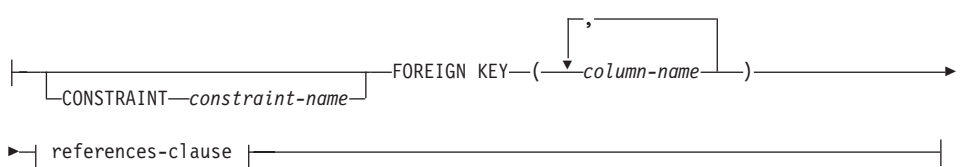
default-clause:



unique-constraint:



referential-constraint:



CREATE TABLE

check-constraint:

```
┌──────────────────────────────────┐ CHECK (—check-condition—) ────────────────────>
└── CONSTRAINT —constraint-name ─┘
▶ | constraint-attributes | ───────────────────────────────────────────────────▶
```

OID-column-definition:

```
┌ REF IS —OID-column-name — USER GENERATED ───────────────────────────────────▶
```

with-options:

```
┌ —column-name — WITH OPTIONS ─── | column-options | ───────────────────────────▶
```

説明:

システム保守済みマテリアライズ照会表およびユーザー保守済みマテリアライズ照会表は、別個におのおのを識別する必要が生じない限りは、マテリアライズ照会表という共通用語で参照されます。

table-name

表の名前を指定します。暗黙または明示の修飾子を含む名前は、カタログに記述されている表、ビュー、または別名を指定するものであってはなりません。スキーマ名は SYSIBM、SYSCAT、SYSFUN、または SYSSTAT であってはなりません (SQLSTATE 42939)。

OF *type-name1*

表の列が *type-name1* で指定される構造タイプの属性に基づいていることを指定します。*type-name1* の指定にスキーマ名が含まれていない場合、そのタイプ名は SQL パス上のスキーマを探索することによって決まります (このパスは、静的 SQL の場合は FUNCSPATH プリプロセス・オプションによって、動的 SQL の場合は CURRENT PATH レジスターによって定義されます)。このタイプ名は、既存のユーザー定義タイプ名である (SQLSTATE 42704) 必要があり、また、少なくとも 1 つの属性があつて (SQLSTATE 42997)、しかもインスタンス化可能な構造タイプでなければなりません (SQLSTATE 428DP)。

UNDER が指定されていない場合には、オブジェクト ID 列を指定する必要があります (OID-column-definition を参照)。このオブジェクト ID 列は、その表の最初の列になります。オブジェクト ID 列の後に、*type-name1* の属性に基づく列が続きます。

HIERARCHY *hierarchy-name*

表階層に関連する階層表を指定します。同時に、これは階層のルート表としても作成されます。型付き表階層に含まれる副表のデータはすべて、この階層表に保管されます。階層表を SQL ステートメントで直接に参照することはできません。

hierarchy-name は *table-name* になります。暗黙または明示のスキーマ名の入った *hierarchy-name* は、カタログに記述されている表、ニックネーム、ビュー、または別名を指定するものであってはなりません。スキーマ名を指定する場合、作成する

表のスキーマ名と同じにする必要があります (SQLSTATE 428DQ)。ルート表の定義時にこの文節を省略すると、システムによって名前が生成されます。この名前は、作成する表の名前とその後の固有な接尾部で構成される ID であり、既存の表、ビュー、別名、およびニックネームの ID の中で固有のものです。

UNDER *supertable-name*

表が *supertable-name* の副表であることを指定します。スーパー表は既存の表でなければならず (SQLSTATE 42704)、かつ表は *type-name1* のすぐ上のスーパータイプである構造タイプを使用して定義しなければなりません (SQLSTATE 428DB)。*table-name* と *supertable-name* のスキーマ名は、同じでなければなりません (SQLSTATE 428DQ)。*supertable-name* で指定される表には、*type-name1* で既に定義された既存の副表を含めることはできません (SQLSTATE 42742)。

表の列には、スーパー表のオブジェクト ID 列が含まれています。この列のタイプは、REF(*type-name1*) に変更されており、*type-name1* の属性に基づく列が続きます (ここでいうタイプには、スーパータイプの属性も含まれていることを念頭に置いてください)。属性名は OID 列名と同じものにするにはできません (SQLSTATE 42711)。

表スペース、データ取り込みなど他の表オプションは、最初のうちログされません。また、区分化キーを指定することはできません。これらのオプションはスーパー表から継承されます (SQLSTATE 42613)。

INHERIT SELECT PRIVILEGES

スーパー表に対して SELECT 特権を保持するユーザーやグループはすべて、新しく作成した副表に対しても同様の特権を付与されます。この特権は、副表定義者によって付与されたものと見なされます。

element-list

表の要素を定義します。これには、表の列と制約の定義が含まれます。

typed-element-list

型付き表の追加要素を定義します。これには、列の追加オプション、オブジェクト ID 列 (ルート表のみ) の追加、表の制約事項などが含まれます。

as-subquery-clause

表の定義が照会の結果に基づいている場合に、その表は照会に基づくマテリアライズ照会表となります。

column-name

表の列の名前を指定します。列名のリストを指定する場合、リスト中の列の名前の数は、全選択の結果表の列の数と同じ数でなければなりません。各 *column-name* (列名) は、固有で、しかも非修飾でなければなりません。列名のリストの指定がない場合、表の列は、全選択の結果表の列名を継承します。

CREATE TABLE

全選択の結果表に、無名列の重複列名がある場合には、列名のリストを指定する必要があります (SQLSTATE 42908)。無名列とは、定数、関数、式、またはセット演算から派生した列で、選択リストの AS 文節によって名前が指定されていない列を指します。

AS 表の定義および表に含まれるデータの判別に使用する照会をこの後に指定します。

fullselect

表の基礎となる照会を定義します。作成される列定義は、同じ照会で定義したビューの定義と同じになります。

各選択リスト・エレメントには名前が必要です (式には AS 文節を使用します。)。 *as-subquery-clause* は、マテリアライズ照会表の属性を定義します。選択されたオプションは、以下のような全選択の内容も定義します。

DEFINITION ONLY が指定されている場合、型付き表または型付きビューを参照しない有効な全選択を指定することができます。

REFRESH DEFERRED または REFRESH IMMEDIATE が指定されていると、全選択で以下のものを指定できません (SQLSTATE 428EC)。

- マテリアライズ照会表、宣言済み一時表、または任意の FROM 文節での型付き表への参照
- ビューの全選択が、マテリアライズ照会表の全選択に関してリストされたいずれかの制限に違反する場合の、そのビューへの参照
- 参照タイプまたは DATALINK タイプ (あるいはこれらのタイプに基づく特殊タイプ) である式
- 外部アクションを指定する関数
- SQL で書かれた関数
- 物理的特性に依存する関数 (たとえば、DBPARTITIONNUM、HASHEDVALUE)
- システム・オブジェクトに対する表またはビュー参照 (Explain 表も指定できません)
- 構造タイプまたは LOB タイプ (または LOB タイプに基づいた特殊タイプ) である式
- CONTAINS SQL または READS SQL DATA のいずれかによって定義されたユーザー定義関数

REFRESH IMMEDIATE は、以下の場合に指定されます。

- 全選択が必ず副選択であること
- 全選択にニックネームへの参照を含めることができない (SQLSTATE 428EC)
- 副選択に以下のものが含まれていないこと
 - deterministic 関数でない関数

- スカラー全選択
- 全選択を持つ述部
- 特殊レジスター
- REPLICATED が指定 (この場合には GROUP BY 文節は許可されない) されない限り、副選択に GROUP BY が必ず入っていること。
- サポートされている列関数は SUM、COUNT、COUNT_BIG、および GROUPING (DISTINCT は指定しない)。選択リストには COUNT(*) または COUNT_BIG(*) 列が含まれていなければなりません。マテリアライズ照会表の選択リストには、SUM(X) が含まれています。この X には、NULL 可能な引き数が入ります。マテリアライズ照会表の選択リストには、COUNT(X) も含まれていなければなりません。これらの列関数は、式の一部とすることはできません。
- FROM 文節で複数の表またはビューを参照している場合、明示的な INNER JOIN 構文を使わずに内部結合を 1 つだけ定義できます。
- すべての GROUP BY 項目が選択リストに含まれていること
- GROUPING SETS、CUBE、および ROLLUP がサポートされていること。選択リスト内の GROUP BY 項目とそれに関連した GROUPING 列関数は、結果セットのユニーク・キーを形成していなければなりません。したがって、以下の制約事項が満たされていなければなりません。
 - どのグループ・セットも反復することはできない。たとえば、ROLLUP(X,Y), X は指定できません。これは GROUPING SETS((X,Y),(X),(X)) と同等だからです。
 - X が、GROUPING SETS、CUBE、または ROLLUP 内に出現する NULL 可能な GROUP BY 項目である場合、選択リスト内に GROUPING(X) がなければならぬ
 - 定数でのグループ化は行えない
- HAVING 文節は許可されていない
- 複数パーティションのデータベース・パーティション・グループ内の場合、区分化キーは GROUP BY 項目のサブセットでなければならない。そうでなければ、REPLICATED が指定されなければならない
- REPLICATED が指定される場合には、表はユニーク・キーを持っていないなければならない

全選択に GROUP BY 文節が含まれるマテリアライズ照会表は、全選択で参照される表からの要約されたデータです。そのようなマテリアライズ照会表は、要約表 と呼ばれます。要約表は、マテリアライズ照会表が特殊化したタイプの表です。

DEFINITION ONLY

照会は、表を定義するときだけに使われます。表は照会結果を使って移植されず、

CREATE TABLE

REFRESH TABLE ステートメントは使用できません。CREATE TABLE ステートメントが完了すると、表はマテリアライズ照会表と見なされなくなります。

表の列は、全選択の結果である列の定義に基づいて定義されます。全選択によって FROM 文節の 1 つの表が参照される場合、その表の列である選択リスト項目は、参照される表の列名、データ・タイプ、そして NULL 可能特性を使って定義されます。

materialized-query-table-options

マテリアライズ照会表の属性の再生可能オプションを定義します。

DATA INITIALLY DEFERRED

データは CREATE TABLE ステートメントの一部として表に挿入されません。データを表に挿入するには、*table-name* (表名) を指定する REFRESH TABLE ステートメントが使用されます。

REFRESH

表のデータを保守する方法を示します。

DEFERRED

REFRESH TABLE ステートメントを使っていつでも表のデータをリフレッシュできます。表のデータには、REFRESH TABLE ステートメント処理時のスナップショットである照会結果が反映されるにすぎません。この属性を定義したシステム保守済みマテリアライズ照会表には、INSERT、UPDATE、または DELETE ステートメントを使用できません (SQLSTATE 42807)。この属性を定義したユーザー保守済みマテリアライズ照会表には、INSERT、UPDATE、または DELETE ステートメントを使用できません。

IMMEDIATE

DELETE、INSERT、または UPDATE の一部として基礎表に加えられた変更は、マテリアライズ照会表にカスケードされます。その場合、表の内容は、どのポイント・イン・タイム指定でも、指定した *subselect* (副選択) を処理する場合と同じ内容になります。この属性を定義したマテリアライズ照会表には、INSERT、UPDATE、または DELETE ステートメントを使用できません (SQLSTATE 42807)。

ENABLE QUERY OPTIMIZATION

適切な状況下では、マテリアライズ照会表を照会最適化に使用することができます。

DISABLE QUERY OPTIMIZATION

マテリアライズ照会表を照会の最適化に使用しません。それでもその表を直接照会することはできます。

MAINTAINED BY SYSTEM

マテリアライズ照会表のデータがシステムによって保守されるように指定します。これがデフォルトです。

MAINTAINED BY USER

マテリアライズ照会表のデータがユーザーによって保守されるように指定します。ユーザーは、ユーザー保守済みマテリアライズ照会表に対して、更新、削除、また挿入操作を許可されます。システム保守済みマテリアライズ照会表で使用される **REFRESH TABLE** ステートメントは、ユーザー保守済みマテリアライズ照会表からは呼び出せません。 **REFRESH DEFERRED** マテリアライズ照会表だけが、**MAINTAINED BY USER** として定義されることが可能です。

staging-table-definition

関連付けられたマテリアライズ照会表を通して間接的に、ステージング表によりサポートされる照会を定義します。マテリアライズ照会表の基礎表も、関連付けられたステージング表の基礎表です。ステージング表は、基礎表の内容と同期化するためにマテリアライズ照会表に適用する必要のある変更を収集します。

staging-column-name

ステージング表の列の名前を指定します。列名のリストを指定する場合、ステージング表を定義するマテリアライズ照会表の列の名前の数よりも、2 つ名前の数が多くなければなりません。そのマテリアライズ照会表がマテリアライズ照会表の複製である場合、またはマテリアライズ照会表を定義している照会が **GROUP BY** 文節を含んでいない場合には、ステージング表を定義するマテリアライズ照会表の列の名前の数よりも、3 つ名前の数が多くなければなりません。各 **column-name** (列名) は、固有かつ非修飾でなければなりません。列名のリストの指定がない場合、表の列は、関連付けられたマテリアライズ照会表の列名を継承します。追加の列は **GLOBALTRANSID** および **GLOBALTRANSTIME** と名づけられ、3 番目の列が必要な場合には **OPERATIONTYPE** という名前が付けられます。

表 269. ステージング表に加えられる追加の列

列名	データ・タイプ	列の説明
GLOBALTRANSID	CHAR(8) FOR BIT DATA	伝搬された各行のグローバル・トランザクション ID
GLOBALTRANSTIME	CHAR(13) FOR BIT DATA	トランザクションのタイム・スタンプ
OPERATIONTYPE	SMALLINT	伝搬された行に対する操作。挿入、更新、または削除のいずれか。

関連付けられたマテリアライズ照会表の任意の列が、生成された列名と重複する場合には、列名のリストを指定する必要があります (SQLSTATE 42711)。

FOR *table-name2*

ステージング表の定義に使用されるマテリアライズ照会表を指定します。名前 (暗黙的または明示的なスキーマ名を含む) は、**REFRESH DEFERRED** に定義

CREATE TABLE

された現行サーバーに存在するマテリアライズ照会表を指定していなければなりません。関連付けられたマテリアライズ照会表の全選択は、**REFRESH IMMEDIATE** オプションによってマテリアライズ照会表を作成するのに使用した全選択と同じ制限事項および規則に従う必要があります。

ステージング表の内容が関連付けられたマテリアライズ照会表および基礎ソース表と整合する場合には、マテリアライズ照会表をリフレッシュするために、**REFRESH TABLE** ステートメントを呼び出して、ステージング表の内容を使用できます。

PROPAGATE IMMEDIATE

削除、挿入、または更新操作の一部として基礎表に加えられた変更は、ステージング表の同じ削除、挿入、または更新操作にカスケードされます。ステージング表が不整合としてマークされていないのであれば、任意のポイント・イン・タイムに、ステージング表の内容は最後にマテリアライズ照会表をリフレッシュしてから基礎表までの差分変更します。

LIKE *table-name1* または *view-name* または *nickname*

表の列の名前と記述が、指定された表 (*table-name1*)、ビュー (*view-name*)、またはニックネーム (*nickname*) の列とまったく同じであることを指定します。LIKE の後に指定する名前は、カタログに存在する表、ビューまたはニックネーム、あるいは宣言済み一時表を識別するものでなければなりません。型付き表または型付きビューを指定することはできません (SQLSTATE 428EC)。

LIKE は、*n* 列を暗黙的な定義で使います。*n* は、指定した表、ビューまたはニックネームにおける列数です。

- 表を指定すると、暗黙的な定義には *table-name1* のおのおのの列の列名、データ・タイプ、および NULL 可能特性が入ります。EXCLUDING COLUMN DEFAULTS を指定しないと、列のデフォルト値も入ります。
- ビューを指定すると、暗黙的な定義には *view-name* に指定した全選択のおのおのの結果列の列名、データ・タイプ、および NULL 可能特性が入ります。
- ニックネームを指定すると、暗黙的な定義には *nickname* のおのおのの列の列名、データ・タイプ、および NULL 可能特性が入ります。

copy-attributes 文節に基づいて、列のデフォルトと ID 列属性を組み込んだり除外したりすることができます。さらにこの暗黙的な定義には、指定した表、ビュー、またはニックネームの他の属性は含まれません。したがって、新しい表にはユニーク制約、外部キー制約、トリガー、または索引はありません。表は IN 文節で暗黙的にまたは明示的に指定した表スペースの中に作成されます。また、任意指定の他の文節を指定した場合に限り、この表にその任意指定の文節が含まれます。

copy-options

これらのオプションは、ソース結果表の定義 (表、ビュー、または全選択) の追加属性をコピーするかどうかを指定します。

INCLUDING COLUMN DEFAULTS

ソース結果表の定義の更新可能な各列の列デフォルトをコピーします。更新可能でない列では、作成される表の対応列にデフォルトが定義されないことになります。

LIKE *table-name* が指定され、しかも *table-name* が基本表または宣言済み一時表を示す場合、INCLUDING COLUMN DEFAULTS がデフォルトになります。

EXCLUDING COLUMN DEFAULTS

ソース結果表の定義から列デフォルトはコピーされません。

この文節がデフォルトです。ただし、LIKE *table-name* が指定され、かつ *table-name* が基本表または宣言済み一時表を示す場合を除きます。

INCLUDING IDENTITY COLUMN ATTRIBUTES

可能であれば、ソース結果表の定義から ID 列属性がコピーされます。ID 列属性をコピーできるのは、表、ビュー、または全選択内の対応する列のエレメントが、識別特性をもつ基本表列名に直接または間接にマップされる表列の名前またはビュー列の名前である場合です。これら以外の場合はずべて、新規表の列には識別特性は備わりません。たとえば、以下のような場合です。

- 全選択の選択リストに、ID 列名の複数のインスタンスが入っている場合 (つまり、同一列の複数回の選択の場合)
- 全選択の選択リストに複数の ID 列が含まれている (つまり、結合が関与している) 場合
- ID 列が選択リスト内の式に組み込まれている場合
- 全選択にセット演算 (UNION (合併)、EXCEPT (差)、または INTERSECT (論理積)) が含まれている場合

EXCLUDING IDENTITY COLUMN ATTRIBUTES

ソース結果表の定義から ID 列属性はコピーされません。

ORGANIZE BY DIMENSIONS (*column-name,...*)

表データをクラスター化するために使用する、各列または列のグループのディメンションを指定します。ディメンション・リストで括弧を使用すると、列のグループは 1 つのディメンションとして扱われよう指定されます。DIMENSIONS キーワードはオプションです。

クラスタリング・ブロック索引はそれぞれ指定されたディメンション用に自動的に保持され、文節で使用されるすべての列で構成されるブロック索引は、どのクラスタリング・ブロック索引にもすべての列が含まれていない場合には保持されます。ORGANIZE BY 文節で使用される列の集合は、CREATE INDEX ステートメントの規則に従う必要があります。

ORGANIZE BY 文節で指定された各列名は、表に対して定義されなければならない (SQLSTATE 42703)、ディメンションはディメンション・リストで複数回現れることはありません (SQLSTATE 42709)。

CREATE TABLE

表のページは、表スペースのエクステンツ・サイズと同じサイズのブロックに配置され、各ブロックの行すべては同じディメンション値の組み合わせを含みます。

column-definition

列の属性を定義します。

column-name

表を構成する列の名前を指定します。名前を修飾したり、表の複数の列に対して同じ名前を使用することはできません (SQLSTATE 42711)。

表には、以下のものを指定できます。

- 4K ページ・サイズの場合、最大 500 列。列のバイト・カウントは 4 005 を超えてはなりません。
- 8K ページ・サイズの場合、最大 1 012 列。列のバイト・カウントは 8 101 を超えてはなりません。
- 16K ページ・サイズの場合、最大 1 012 列。列のバイト・カウントは 16 293 を超えてはなりません。
- 32K ページ・サイズの場合、最大 1 012 列。列のバイト・カウントは 32 677 を超えてはなりません。

詳細については、1535 ページの『行サイズ』を参照してください。

data-type

以下のリストのタイプのいずれかを指定します。データ・タイプは、以下のとおりです。

SMALLINT

短整数。

INTEGER または **INT**

長精度整数。

BIGINT

64 ビット整数。

FLOAT(*integer*)

単精度または倍精度の浮動小数点数 (*integer* の値によって異なる)。

integer (整数) は、1 ~ 53 の範囲の整数でなければなりません。1 ~ 24 の値は単精度、25 ~ 53 の値は倍精度を示します。

また、以下を指定することもできます。

REAL 単精度浮動小数点。

DOUBLE 倍精度浮動小数点。

DOUBLE-PRECISION 倍精度浮動小数点。

FLOAT 倍精度浮動小数点。

DECIMAL(*precision-integer, scale-integer*) または **DEC**(*precision-integer, scale-integer*)

10 進数。最初の整数は数値の精度、つまり数字の総桁数です。これは、1 ～ 31 の範囲で指定できます。2 番目の整数は、数値の位取り、つまり、小数点の右側の桁数です。これは、0 ～ 数値の精度までの範囲で指定できます。

精度と位取りが指定されない場合、5,0 のデフォルト値が使用されます。

NUMERIC および **NUM** は、**DECIMAL** および **DEC** の同義語として使用可能です。

CHARACTER(*integer*)、または **CHAR** (*integer*)、または **CHARACTER**、または **CHAR**

長さ *integer* (整数) の固定長文字ストリング。長さは、1 ～ 254 の範囲で指定できます。長さの指定がない場合は、1 文字の長さを指定したものと同じと見なされます。

VARCHAR(*integer*)、または **CHARACTER VARYING**(*integer*)、または **CHAR VARYING**(*integer*)

長さが *integer* の可変長文字ストリング。長さは、1 ～ 32,672 の範囲で指定できます。

LONG VARCHAR

最大長 32 700 の可変長文字ストリング。

FOR BIT DATA

列の内容をビット (2 進) データとして扱うように指定します。他のシステムとのデータ交換の過程で、コード・ページ変換は行われません。比較は、データベース照合シーケンスに関係なく 2 進で行われます。

BLOB または **BINARY LARGE OBJECT**(*integer [K | M | G]*)

バイナリー・ラージ・オブジェクト・ストリング (最大長をバイト単位で指定)。

長さは、1 ～ 2 147 483 647 バイトの範囲で指定できます。

integer (整数) だけを指定した場合は、それが最大長になります。

integer K (大文字または小文字) を指定した場合、最大長は *integer* の 1,024 倍になります。 *integer* の最大値は 2 097 152 です。計算結果が 2 147 483 648 を超える K、M、または G の倍数を指定すると、使用される実際の値は 2 147 483 647 (2 ギガバイト - 1 バイト) になります。これは LOB 列の最大長です。

integer M を指定した場合、最大長は *integer* の 1 048 576 倍になります。 *integer* の最大値は 2 048 です。

integer G を指定した場合、最大長は *integer* の 1 073 741 824 倍になります。 *integer* の最大値は 2 です。

CREATE TABLE

長さの指定がない場合、長さが 1 048 576 (1 メガバイト) であると想定されます。

1 ギガバイトを超える BLOB ストリングを作成するには、NOT LOGGED オプションを指定する必要があります。

integer と K、M、または G の間には、任意の数のスペースを使用できます。スペースは必須ではありません。たとえば、以下の例はすべて有効です。

BLOB(50K) BLOB(50 K) BLOB (50 K)

CLOB または **CHARACTER (CHAR) LARGE OBJECT**(*integer* [*K* | *M* | *G*])

文字ラージ・オブジェクト・ストリング (最大長をバイト単位で指定)。

integer *K* | *M* | *G* の意味は、BLOB の場合と同じです。

長さの指定がない場合、長さが 1 048 576 (1 メガバイト) であると想定されます。

1 ギガバイトを超える CLOB ストリングを作成するには、NOT LOGGED オプションを指定する必要があります。

CLOB 列の場合に、FOR BIT DATA 文節を指定することはできません。ただし、CLOB 列に CHAR FOR BIT DATA ストリングを割り当てることができ、CLOB ストリングに CHAR FOR BIT DATA ストリングを連結することができます。

DBCLOB(*integer* [*K* | *M* | *G*])

2 バイト文字ラージ・オブジェクト (最大長を 2 バイト文字の数で指定)。

integer *K* | *M* | *G* の意味は、BLOB の場合に類似しています。指定する数値が 2 バイト文字 1 個を 1 文字と数えた値であることと、最大サイズが 2 バイト文字 1 073 741 823 個であるという点が違います。

長さの指定がない場合、長さが 1 048 576 (2 バイト文字) であると想定されます。

1 ギガバイトを超える DBCLOB ストリングを作成するには、NOT LOGGED オプションを指定する必要があります。

GRAPHIC(*integer*)

長さ *integer* (整数) の固定長 GRAPHIC ストリング。長さは、1 ~ 127 の範囲で指定できます。長さの指定がない場合、長さが 1 であると想定されます。

VARGRAPHIC(*integer*)

長さが *integer* の可変長 GRAPHIC ストリング。長さは、1 ~ 16,336 の範囲で指定できます。

LONG VARCHAR

最大長 16,350 の可変長 GRAPHIC ストリングを示します。

DATE

日付を示します。

TIME

時刻を示します。

TIMESTAMP

タイム・スタンプを示します。

DATALINK または **DATALINK(*integer*)**

データベース外に保管されたデータ・ファイルへのリンクを示します。

表内の列は "アンカー値" で構成されます。このアンカー値には、外部データだけではなく、任意コメントへのリンクを確立したり保持したりするための参照情報が含まれています。

DATALINK 列の長さは 200 バイトです。 *integer* を指定する場合、200 でなければなりません。長さの指定がない場合、長さは 200 バイトであると想定されます。

DATALINK 値とは、一連の組み込みスカラー関数を持つカプセル化された値です。 DATALINK 値は DLVALUE という名前の関数が作成し、また特別な環境で DATALINK 値を構成するためには DLNEWCOPY、DLPREVIOUSCOPY、および DLREPLACECONTENT という名前の関数が使用されます。(DLVALUE は、正規の DATALINK 値を構成するために使用してください。) DATALINK 値から属性を抽出するために、以下の関数を使用することができます。

- DLCOMMENT
- DLLINKTYPE
- DLURLCOMPLETE
- DLURLCOMPLETEONLY
- DLURLCOMPLETEWRITE
- DLURLPATH
- DLURLPATHONLY
- DLURLPATHWRITE
- DLURLSCHEME
- DLURLSERVER

DATALINK 列には、以下の制限事項があります。

- 列は索引の一部であってはならない。したがって、この列を主キーまたはユニーク制約の列として組み込むことはできません (SQLSTATE 42962)。

CREATE TABLE

- 列は参照制約の外部キーであってはならない (SQLSTATE 42830)。
- この列にデフォルト値 (WITH DEFAULT) を指定することはできない。列を NULL 値にすることができる場合、この列のデフォルト値は NULL です (SQLSTATE 42894)。

distinct-type-name

ユーザー定義タイプの中で特殊タイプであるものを示します。スキーマ名を伴わない特殊タイプ名を指定した場合、その特殊タイプ名は SQL パスのスキーマから探索することによって解決されます (このパスは、静的 SQL の場合は FUNCPATH プリプロセス・オプションによって、動的 SQL の場合は CURRENT PATH レジスターによって定義されます)。

特殊タイプを使用して列を定義する場合、その列のデータ・タイプはその特殊タイプになります。列の長さや位取りは、それぞれ特殊タイプのソース・タイプの長さや位取りになります。

特殊タイプを使用して定義された列が参照制約の外部キーである場合、主キーの対応する列のデータ・タイプは、同じ特殊タイプでなければなりません。

structured-type-name

ユーザー定義タイプの中で構造タイプであるものを示します。構造タイプ名の指定にスキーマ名が含まれていない場合、その構造タイプ名は SQL パスのスキーマから探索することによって決まります (このパスは、静的 SQL の場合は FUNCPATH プリプロセス・オプションによって、動的 SQL の場合は CURRENT PATH レジスターによって定義されます)。

構造タイプを使用して列を定義する場合、その列の静的データ・タイプはその構造タイプになります。その列には、*structured-type-name* のサブタイプである動的タイプをもつ値を組み込むことができます。

構造タイプを使用して定義された列を、主キー、ユニーク制約、外部キー、索引キー、または区分化キー内で使うことはできません (SQLSTATE 42962)。

列が、構造タイプを使用して定義されていて、ネストのいずれかのレベルで参照タイプ属性をもっている場合、その参照タイプ属性の有効範囲は解除されます。そのような属性を参照操作で使用するには、CAST 指定を使って SCOPE を明示的に指定する必要があります。

タイプ DATALINK の属性をもつ構造タイプか、または DATALINK をソースとして派生された特殊タイプを使って列が定義されている場合、その列は NULL にしかできません。このようなタイプの場合にコンストラクター関数を使うと、エラーが戻される (SQLSTATE 428ED) ので、このタイプのインスタンスは列に挿入できません。

REF (*type-name2*)

型付き表への参照。 *type-name2* の指定にスキーマ名が含まれていない場

合、そのタイプ名は SQL パス上のスキーマを探索することによって決まります (このパスは、静的 SQL の場合は FUNCSPATH プリプロセス・オプションによって、動的 SQL の場合は CURRENT PATH レジスタによって定義されます)。この列の基礎を成すデータ・タイプは、CREATE TYPE ステートメントの REF USING 文節で *type-name2* に対して指定された表示データ・タイプに基づくか、または *type-name2* の入ったデータ・タイプ階層のルート・タイプに基づきます。

column-options

表の列に関連した追加オプションを定義します。

NOT NULL

列に NULL 値が入るのを防止します。

NOT NULL を指定しない場合、列に NULL 値を含めることができます。また、そのデフォルト値は、NULL 値または WITH DEFAULT 文節で指定される値のいずれかになります。

lob-options

LOB データ・タイプのオプションを指定します。

LOGGED

列に対して行われた変更をログに書き込むことを指定します。このような列のデータは、データベース・ユーティリティー (RESTORE DATABASE など) によってリカバリーすることができます。LOGGED はデフォルト値です。

1 ギガバイトを超える LOB はログ記録することができず (SQLSTATE 42993)、10 メガバイトを超える LOB はログ記録されない可能性があります。

NOT LOGGED

列に対して行われた変更をログに書き込まないことを指定します。

NOT LOGGED は、コミットやロールバックの操作には影響しません。つまり、トランザクションがロールバックされても、LOB の値がログ記録されるか否かに関係なくデータベースの整合性は保持されます。ログ記録されないので、ロールフォワード操作中、バックアップまたはロード操作の後の LOB データは、ロールフォワード操作中にログ・レコードを再生させることになった LOB 値をゼロで置換したのになります。クラッシュ・リカバリーの過程で、コミットされた変更とロールバックされた変更すべてに、予期された結果が反映されます。

COMPACT

後続の付加操作で使用するためのスペースを LOB ストレージの最後に残すことなく、LOB 列の値で消費されるディスク・スペースを最小限にすることを指定します (LOB 値が使用する最後のグループ内の

CREATE TABLE

余分なディスク・スペースすべてを解放します)。このようにしてデータを保管した場合、列に対する付加操作 (長さを増加する操作) のパフォーマンスが低下することがあります。

NOT COMPACT

列の LOB 値に対する将来の変更に備えて、いくらかのスペースを挿入するように指定します。これがデフォルトです。

datalink-options

DATALINK データ・タイプに関連したオプションを指定します。

LINKTYPE URL

リンク・タイプを URL として定義します。

NO LINK CONTROL

ファイルが存在するかどうかを判別するチェックを行わないことを指定します。URL の構文だけがチェックされます。データベース・マネージャーによってファイルが制御されることはありません。

FILE LINK CONTROL

ファイルが存在するかどうかを判別するチェックが行われるように指定します。データベース・マネージャーがファイルをより一層制御できるように、追加オプションが使用されます。

file-link-options

データベース・マネージャーによるファイル・リンク制御の度合いを定義する追加オプション。

INTEGRITY

DATALINK 値と実ファイルの間に存在するリンクの保全レベルを指定します。

ALL

DATALINK 値として指定されているものは、すべてデータベース・マネージャーによって制御されており、それらは標準的なファイル・システム・プログラム・インターフェースを使って削除したり名前変更したりすることはできません。

READ PERMISSION

DATALINK 値に指定されているファイルの読み取り許可を決定する方法を指定します。

FS 読み取りアクセス許可は、ファイル・システム許可によって決められます。列からファイル名を検索することなく、そのようなファイルにアクセスできます。

DB

読み取りアクセス許可は、データベースによって決められます。有効なファイル・アクセス・トークンを渡してから、表

から検出された DATALINK 値をオープン操作で戻すことによつてのみファイルへのアクセスが可能になります。

WRITE PERMISSION

DATALINK 値に指定されているファイルへの書き込み許可を決定する方法を指定します。

FS 書き込みアクセス許可は、データベースによって決められます。列からファイル名を検索することなく、そのようなファイルにアクセスできます。

BLOCKED

書き込みアクセスはブロックされます。このファイルはどのインターフェースを介しても直接更新することはできません。この情報を更新するには、代替メカニズムを使用する必要があります。たとえば、ファイルがコピーされて、そのコピーが更新されるとします。その場合には、DATALINK 値もファイルの新しいコピーを指すように更新されなければなりません。

ADMIN

書き込み許可は、Data Links Manager によって判別されます。有効な書き込みトークンを渡してから、表から検出された DATALINK 値をオープン操作で DLURLCOMPLETEWRITE または DLURLPATHWRITE スカラー関数を使用して戻すことによつてのみ、ファイルへの書き込みアクセスが可能になります。この値は、READ PERMISSION DB も指定された場合にのみ指定が可能です。指定されたリンク・ファイルに対するアクセス権は、Data Links Manager で定義され、保守されます。

ユーザー (更新者) が有効な書き込みトークンを持って書き込み用にファイルをオープンすると、他のユーザーは有効な読み取りまたは書き込みトークンを使用して、読み取り用にファイルをオープンすることは依然として可能です。しかし同じ書き込みトークンを使用して、書き込み用にファイルを繰り返しオープンできるのは、同じ更新者だけです。また同じ更新者は、その後に読み取り操作を実行するためにも同じ書き込みトークンが必要です。

REQUIRING TOKEN FOR UPDATE

ファイル更新を完了するには、ファイルをオープンして変更するのに使用した書き込みトークンは、SQL UPDATE ステートメントに DLNEWCOPY または DLPREVIOUSCOPY スカラー関数を呼び出す際に指定したファイル参照に含まれていなければなりません。

CREATE TABLE

NOT REQUIRING TOKEN FOR UPDATE

ファイル更新を完了するためには、SQL UPDATE ステートメントに DLNEWCOPY または DLPREVIOUSCOPY スカラー関数を呼び出す際に指定したファイル参照には書き込みトークンは必要ではありません。

RECOVERY

この列の値によって参照されているファイルの特定時点でのリカバリーを DB2 がサポートしているかどうかを指定します。

YES

DB2 は、この列の値で参照されているファイルの特定時点でのリカバリーをサポートしています。この値は、INTEGRITY ALL と WRITE PERMISSION BLOCKED または WRITE PERMISSION ADMIN とが指定されている場合にだけ指定することができます。

NO

ポイント・イン・タイム・リカバリーがサポートされていないことを指定します。

ON UNLINK

DATALINK 値が変更または削除された（リンク解除された）ときに実行する処置を指定します。WRITE PERMISSION FS が使用される際には、適用できないことに注意してください。

RESTORE

ファイルがリンク解除されたときに、ファイルがリンクされていた時点で許可を持つ所有者に、データ・リンク・ファイル・マネージャーがファイルを戻すように指定します。ユーザーがファイル・サーバーから登録を解除されている場合には、ファイルは特別に事前定義された "dfmunknown" ユーザー ID に割り当てられます。この値は、INTEGRITY ALL と WRITE PERMISSION BLOCKED または WRITE PERMISSION ADMIN とが指定されている場合にだけ指定することができます。

DELETE

リンク解除される際にファイルが削除されるように指定します。この値を指定することができるのは、READ PERMISSION DB および WRITE PERMISSION BLOCKED または WRITE PERMISSION ADMIN も指定されている場合だけです。

MODE DB2OPTIONS

このモードは、一連のデフォルト・ファイル・リンク・オプションを定義します。 DB2OPTIONS によって定義されるデフォルト値は、以下のとおりです。

- INTEGRITY ALL
- READ PERMISSION FS
- WRITE PERMISSION FS
- RECOVERY NO

WRITE PERMISSION FS が使用されている場合、ON UNLINK は適用できません。

SCOPE

参照タイプ列の有効範囲を指定します。

参照解除演算子の左オペランド、または Deref 関数の引き数として使用する列には、すべて有効範囲を指定する必要があります。 ターゲット表が定義されるように、後続する ALTER TABLE ステートメントまで参照タイプ列の指定を遅らせることができます (通常は、相互参照表の場合に適用する)。

typed-table-name

型付き表の名前。この表は既に存在しているものか、作成する表と同じ名前のものでなければなりません (SQLSTATE 42704)。

column-name のデータ・タイプは REF(S) でなければなりません。 S は *typed-table-name* のタイプを表します (SQLSTATE 428DM)。

column-name に割り当てられた値が、 *typed-table-name* に存在する行を実際に参照しているかどうかを示す検査は行われません。

typed-view-name

型付きビューの名前。このビューは既に存在しているものか、作成するビューと同じ名前のものでなければなりません (SQLSTATE 42704)。 *column-name* のデータ・タイプは REF(S) でなければなりません。 S は *typed-view-name* のタイプを表します (SQLSTATE 428DM)。 *column-name* に割り当てられた値が、 *typed-view-name* に存在する行を実際に参照しているかどうかを示す検査は行われません。

CONSTRAINT *constraint-name*

制約の名前を指定します。 *constraint-name* (制約名) は、同じ CREATE TABLE ステートメントにすでに指定されている制約を指定するものであってはなりません (SQLSTATE 42710)。

CREATE TABLE

この文節が省略された場合は、表に定義されている既存の制約の ID の中で固有な 18 文字の ID がシステムによって生成されます。(ID は、"SQL" と、タイム・スタンプに基づいて生成される 15 の数字から構成されます。)

PRIMARY KEY 制約またはユニーク制約とともに使用した場合、この *constraint-name* は、制約をサポートするために作成される索引の名前として使用されます。

PRIMARY KEY

これは、1 つの列からなる主キーを定義する簡単な方法として用意されています。つまり、PRIMARY KEY が列 C の定義で指定されている場合、その効果は、PRIMARY KEY(C) 文節を独立した文節として指定する場合と同じです。

表が副表である場合、主キーはスーパー表から継承されるので (SQLSTATE 429B3)、主キーを指定することはできません。

この後の *unique-constraint* の説明の中の PRIMARY KEY を参照してください。

UNIQUE

これは、1 つの列からなるユニーク・キーを定義する簡単な方法です。すなわち、UNIQUE を列 C の定義に指定すると、UNIQUE(C) 文節を独立した文節として指定した場合と同じ結果になります。

表が副表である場合、ユニーク制約はスーパー表から継承されるので (SQLSTATE 429B3)、ユニーク制約を指定することはできません。

この後の *unique-constraint* の説明の中の UNIQUE を参照してください。

references-clause

これは、1 つの列からなる外部キーを定義する簡単な方法として用意されています。つまり、*references-clause* が列 C の定義に指定されている場合、その効果は、列として C しか指定されていない FOREIGN KEY 文節の一部として *references-clause* が指定された場合と同じになります。

後述の *referential-constraint* の *references-clause* の項を参照してください。

CHECK (*check-condition*)

これは、1 つの列に適用されるチェック制約を定義する簡単な方法として用意されています。後述の CHECK (*check-condition*) を参照してください。

INLINE LENGTH *integer*

このオプションは、構造タイプを使って定義された列に対してだけ有効であり (SQLSTATE 42842)、行内の残りの値とともにインラインで保管する構造タイプのインスタンスの最大バイト・サイズを指示します。インライ

ンで保管できない構造タイプのインスタンスは、LOB 値が処理されるのに似た方法で、基本表の行とは別に保管されます。これは自動的に行われます。

構造タイプ列のデフォルトの `INLINE LENGTH` は、このタイプのインライン長になります (明示的に指定するか、または `CREATE TYPE` ステートメント内のデフォルトとして)。構造タイプの `INLINE LENGTH` が 292 未満の場合、列の `INLINE LENGTH` には値 292 が使われます。

注: サブタイプのインライン長は、デフォルトのインライン長には含まれませんが、それは、`CREATE TABLE` 時に明示的に `INLINE LENGTH` を指定して、現在および将来のサブタイプに対処できるようにしておかないと、サブタイプのインスタンスはインラインに適合しないことがあることを意味します。

明示的な `INLINE LENGTH` の値は少なくとも 292 でなければならず、32672 を超えてはなりません (SQLSTATE 54010)。

COMPRESS SYSTEM DEFAULT

システム・デフォルト値 (つまり、特定の値が指定されない場合にデータ・タイプとして使用されるデフォルト値) が最小限のスペースを使用して保管されるように指定します。 `VALUE COMPRESSION` 文節が指定されていない場合には警告が出され (SQLSTATE 01648)、システム・デフォルト値は最小限のスペースを使用しては保管されません。

システム・デフォルト値がこのような方法で保管されると、列上での許可や更新操作の際に余分なチェックが行われるために、若干パフォーマンスが低下します。

基本データ・タイプは、`DATE`、`TIME`、または `TIMESTAMP` であってはなりません (SQLSTATE 42842)。基本データ・タイプが可変長ストリングの場合には、この文節は無視されます。表が `VALUE COMPRESSION` に設定されている場合は、長さ 0 のストリング値は自動的に圧縮されます。

generated-column-spec

default-clause

列のデフォルト値を指定します。

WITH

オプション・キーワード。

DEFAULT

`INSERT` で値が提供されなかった場合、もしくは `INSERT` や `UPDATE` で `DEFAULT` が指定されている場合に、デフォルト値を提供します。 `DEFAULT` キーワードの後にデフォルト値が指定

CREATE TABLE

されていない場合、使用されるデフォルト値は列のデータ・タイプによって異なります。『ALTER TABLE』を参照してください。

列を DATALINK として定義する場合、デフォルト値は指定できません (SQLSTATE 42613)。可能なデフォルト値は NULL だけです。

列が型付き表の列に基づいている場合、デフォルト値の定義時には特定のデフォルト値を指定する必要があります。型付き表のオブジェクト ID の列には、デフォルト値を指定することはできません (SQLSTATE 42997)。

列が特殊タイプを使用して定義される場合、列のデフォルト値は、特殊タイプにキャストされたソース・データ・タイプのデフォルト値になります。

構造タイプを使用して列を定義する場合は、*default-clause* を指定できません (SQLSTATE 42842)。

column-definition から DEFAULT を省略すると、その列のデフォルト値として NULL 値が使用されます。そのような列を NOT NULL と定義すると、その列には有効なデフォルトはなくなります。

default-values

default-values に指定できるデフォルト値のタイプは、以下のとおりです。

constant

列のデフォルト値として定数を指定します。指定する定数は、以下の条件を満たしていなければなりません。

- 第 3 章に示されている割り当ての規則にしたがって、その列に割り当てることができる値でなければなりません。
- その列が浮動小数点データ・タイプとして定義されている場合を除き、浮動小数点の定数を指定してはなりません。
- 定数が 10 進定数の場合、その列のデータ・タイプの位取りを超えるゼロ以外の数字を含めてはなりません (たとえば、DECIMAL(5,2) の列のデフォルト値として 1.234 を指定することはできません)。
- 指定する定数が 254 文字を超えてはなりません。この制約には、引用符文字や 16 進定数の X などの先行文字も含まれます。さらに、定数が *cast-function* の引き数の場合には、完全修飾された関数名から取った文字や括弧も含めて、この制限を超えてはなりません。

datetime-special-register

INSERT、UPDATE、または LOAD の実行時における日時特殊レジスターの値 (CURRENT DATE、CURRENT TIME、または CURRENT TIMESTAMP) を、その列のデフォルト値として指定します。その列のデータ・タイプは、指定した特殊レジスターに対応するデータ・タイプでなければなりません (たとえば、CURRENT DATE を指定した場合、データ・タイプは DATE でなければなりません)。

CURRENT SCHEMA

INSERT、UPDATE、または LOAD の実行時における CURRENT SCHEMA 特殊レジスターの値を、その列のデフォルト値として指定します。CURRENT SCHEMA を指定した場合、その列のデータ・タイプは、CURRENT SCHEMA の長さ属性よりも長いか等しい文字ストリングでなければなりません。

USER

INSERT、UPDATE、または LOAD の実行時における USER 特殊レジスターの値を、その列のデフォルト値として指定します。USER を指定した場合、その列のデータ・タイプは、USER の長さ属性よりも長いか等しい文字ストリングでなければなりません。

NULL

その列のデフォルト値として NULL を指定します。NOT NULL の値が指定された場合は、DEFAULT NULL を同じ列定義に指定できますが、その列をデフォルト値に設定しようとするエラーが生じます。

cast-function

この形式のデフォルト値は、特殊タイプ (distinct type)、BLOB、または日時 (DATE、TIME、または TIMESTAMP) データ・タイプとして定義された列に対してのみ使用することができます。特殊タイプの場合、BLOB や日時タイプに基づく例外があり、関数名が列の特殊タイプの名前に一致していなければなりません。スキーマ名で修飾されている場合には、その特殊タイプのスキーマ名と同じでなければなりません。修飾されていない場合には、関数の解決に用いるスキーマ名は特殊タイプのスキーマ名と同じでなければなりません。日時タイプに基づく特殊タイプで、デフォルト値が定数の場合、必ず関数を使用する必要があります。さらに、その関数名は、暗黙または明示のスキーマ名 SYSIBM を持つ特殊タイプのソース・タイプ名に一致していなければなりません。他の日時列の場合は、対応する日時関数も使用できま

CREATE TABLE

す。 BLOB に基づく BLOB または特殊タイプの場合も、関数を使用する必要があります。その関数名は、暗黙または明示のスキーマ名 SYSIBM を持つ BLOB でなければなりません。

constant

引き数として定数を指定します。指定する定数は、特殊タイプのソース・タイプに関する定数の規則 (特殊タイプでない場合は、データ・タイプに関する定数の規則) にしたがっていないければなりません。 *cast-function* が BLOB の場合には、定数として文字列定数を指定する必要があります。

datetime-special-register

CURRENT DATE、CURRENT TIME、または CURRENT TIMESTAMP を指定します。列の特殊タイプのソース・タイプは、指定した特殊レジスターに対応するデータ・タイプでなければなりません。

CURRENT SCHEMA

CURRENT SCHEMA 特殊レジスターの値を指定します。列の特殊タイプのソース・タイプのデータ・タイプは、CURRENT SCHEMA 特殊レジスターの長さ属性よりも長い等しい文字列でなければなりません。 *cast-function* が BLOB の場合には、長さ属性が 8 バイト以上でなければなりません。

USER

USER 特殊レジスターを指定します。列の特殊タイプのソース・タイプのデータ・タイプは、少なくとも 8 バイトの長さの文字列でなければなりません。 *cast-function* が BLOB の場合には、長さ属性が 8 バイト以上でなければなりません。

指定した値が無効な場合、エラー (SQLSTATE 42894) が発生します。

GENERATED

DB2 が列の値を生成することを指示します。その列が IDENTITY 列と見なされることになる場合には、GENERATED を指定する必要があります。

ALWAYS

行が表に挿入されるときや、*generation-expression* の結果値が変更されるたびに、DB2 は常に列の値を生成することを指示します。この式の結果は、表に保管されます。データ伝搬、またはア

ンロードおよび再ロード操作を実行しているのであれば、GENERATED ALWAYS の値をお勧めします。GENERATED ALWAYS は、生成列に必須の値です。

BY DEFAULT

行が表に挿入されたり、更新されるときに、明示的に値を指定しないかぎり、列に DEFAULT を指定して DB2 が列に値を生成することを指示します。データ伝搬を使用したり、アンロード / 再ロードを行ったりするときは、BY DEFAULT の値をお勧めします。

明示的には指示されませんが、値の固有性を確保するには、固有の 1 列の索引を生成列で定義する必要があります。

AS IDENTITY

列をこの表の ID 列にすることを指定します。1 つの表には 1 つしか IDENTITY 列があってはなりません (SQLSTATE 428C1)。列に関連付けられたデータ・タイプがゼロの位取りの完全な数値タイプになっているか、ソース・タイプのユーザー定義特殊タイプがゼロの位取りの完全な数値タイプになっている場合だけ、IDENTITY キーワードが指定可能です (SQLSTATE 42815)。ゼロの位取りの SMALLINT、INTEGER、BIGINT、または DECIMAL や、これらのタイプのうちのいずれかに基づいた特殊タイプは、完全な数値タイプと見なされます。これに対して、単精度および倍精度の浮動小数点は、近似数値データ・タイプと見なされます。参照タイプは、完全な数値タイプで表されていても、ID 列と定義することはできません。

ID 列は暗黙で NOT NULL になります。ID 列は DEFAULT 文節を持つことができません (SQLSTATE 42623)。

START WITH *numeric-constant*

ID 列の最初の値を指定します。この値は、小数点の右側に非ゼロの数字がない (SQLSTATE 42815) かぎり、この列に割り当てることができる正または負の値にすることができます。デフォルトは、昇順シーケンスであれば MINVALUE、降順シーケンスであれば MAXVALUE です。

INCREMENT BY *numeric-constant*

連続した ID 列値の間隔を指定します。この値は、小数点の右側に非ゼロの数字がない (SQLSTATE 428FA) かぎり、この列に割り当てることができる正または負の値にすることができます (SQLSTATE 42815)、長精度整数定数の値を超えることはありません (SQLSTATE 42820)。

この値が負の場合、これは降順シーケンスです。この値が 0 の場合、または正の場合は、昇順シーケンスになります。デフォルトは 1 です。

NO MINVALUE または MINVALUE

降順 ID 列が値の生成を循環または停止する最小値、あるいは最大値に達した後、昇順 ID 列が循環する最小値を指定します。

NO MINVALUE

昇順シーケンスの場合、値は START WITH 値、または START WITH が指定されなかった場合には 1 です。降順シーケンスの場合、列のデータ・タイプの最小値です。これがデフォルトです。

MINVALUE *numeric-constant*

最小値である数値定数を指定します。この値は、小数点の右側に非ゼロの数字がない (SQLSTATE 428FA) かぎり、この列に割り当てることができる正または負の値にすることができます (SQLSTATE 42815) が、最大値以下でなければなりません (SQLSTATE 42815)。

NO MAXVALUE または MAXVALUE

昇順 ID 列が値の生成を循環または停止する最大値、あるいは最小値に達した後、降順 ID 列が循環する最大値を指定します。

NO MAXVALUE

昇順シーケンスの場合、値は列のデータ・タイプの最大値です。降順シーケンスの場合、値は START WITH 値、または START WITH が指定されなかった場合には -1 です。これがデフォルトです。

MAXVALUE *numeric-constant*

最大値である数値定数を指定します。この値は、小数点の右側に非ゼロの数字がない (SQLSTATE 428FA) かぎり、この列に割り当てることができる正または負の値にすることができます (SQLSTATE 42815) が、最小値以下でなければなりません (SQLSTATE 42815)。

NO CYCLE または CYCLE

その最大値または最小値が生成された後、この ID 列が値の生成を続行するかどうかを指定します。

NO CYCLE

最大値または最小値に達した後、ID 列について値が生成されないことを指定します。これがデフォルトです。

CYCLE

最大値または最小値に達した後、この列について値の生成が続行されることを指定します。このオプションが使用されると、昇順 ID 列が最大値に達した後は、その最小値が生成さ

れます。降順 ID 列が最小値に達した後は、その最大値が生成されます。ID 列の最大値および最小値は、循環に使用される範囲を決定します。

CYCLE が有効な場合、DB2 が ID 列について重複する値を生成する可能性があります。固有値を望む場合には、明示的には指示されませんが、値の固有性を確保するには、固有の 1 列の索引を生成列で定義する必要があります。このような ID 列にユニーク索引が存在し、固有ではない値が生成されると、エラーが起こります (SQLSTATE 23505、SQLCODE -803)。

NO CACHE または CACHE

特定の事前割り振り値を、高速アクセスできるようメモリーに保存するかどうかを指定します。ID 列で新しい値が必要になった場合に、キャッシュの中のものを使えないときは、新しいキャッシュ・ブロックの末尾をログ記録する必要があります。ただし、ID 列で新しい値が必要になった場合に、キャッシュの中に未使用の値があるときは、その識別値を割り振ったほうが、ログ記録なくて済むので高速化されます。これはパフォーマンスおよびチューニング・オプションです。

NO CACHE

ID 列の値を事前割り振りしないことを指定します。ID シーケンスの値が複数のロケーションにキャッシュされる環境で、ID 値が要求のシーケンスで生成されなければならない場合には、NO CACHE オプションを使用する必要があります。

このオプションが指定されると、ID 列の値はキャッシュに保管されません。この場合、新しい ID 値が要求されるたびに、ログに対して非同期入出力が行われます。

CACHE *integer-constant*

事前割り振りされ、メモリーに保管される ID シーケンスの値の数を指定します。ID 列について値が生成される場合、値を事前割り振りしてキャッシュに保管しておく、ログへの非同期入出力が少なくなります。

ID 列に新しい値が必要でも未使用の値がキャッシュにない場合、値の割り振りによりログへの入出力の待機が呼び出されます。ただし、ID 列に新しい値が必要で、未使用の値がキャッシュにあれば、その ID 値の割り振りが、ログへの入出力なしで素早く行われます。

システム障害に起因するものであっても通常のものであっても、データベースの活動解除が起こると、コミットされたス

CREATE TABLE

ステートメントで使用されていないキャッシュ済みシーケンス値はすべて失われます。つまり使用されなくなります。データベースの活動解除が起きたら失われる可能性のある ID 列値の最大数は、CACHE オプションに指定された値になります。(データベースが ACTIVATE コマンドまたは API を使用して明示的に活動化されない場合には、最終アプリケーションの接続をデータベースから切断すると、暗黙の活動解除が行われます。)

最小値は 2 です (SQLSTATE 42815)。デフォルト値は CACHE 20 です。

NO ORDER または ORDER

要求の順序で ID 値が生成されるかどうかを指定します。

NO ORDER

要求の順序での値を生成する必要がないことを指定します。これがデフォルトです。

ORDER

要求の順序で値が生成されることを指定します。

AS (*generation-expression*)

列定義が式に基づくことを指定します。(GENERATED ALWAYS 列の式にユーザー定義の外部関数が入っている場合に、その関数の実行可能ファイルを変更する (引き数ごとに異なる結果を得るため) と、データの不整合を生じることがあります。これが生じないようにするには、SET INTEGRITY ステートメントを使って、新しい値を強制的に生成させます。) *generation-expression* には、以下のいずれも入れることができません (SQLSTATE 42621)。

- 副照会
- 列関数
- 参照解除操作または Deref 関数
- 非 deterministic であるユーザー定義関数または組み込み関数
- EXTERNAL ACTION オプションを使用するユーザー定義関数
- CONTAINS SQL または READS SQL DATA のいずれかによって定義されたユーザー定義関数
- ホスト変数またはパラメーター・マーカー
- 特殊レジスター
- 列リスト内で後から定義されている列の参照
- 他の生成列の参照

列のデータ・タイプは *generation-expression* の結果データ・タイプに基づいています。CAST 指定を使って特定のデータ・タイプを強制的

に使用し、有効範囲を設けることができます (参照タイプの場合のみ)。 *data-type* を指定すると、適切な割り当て規則にしたがって、値が列に割り当てられます。 NOT NULL 列オプションを使わない限り、生成された列は暗黙で NULL 可能と見なされます。生成される列のデータ・タイプは、同等性を定義されているものでなければなりません。ただしこの場合、LONG VARCHAR、LONG VARGRAPHIC、LOB データ・タイプ、DATALINK、構造タイプ、および、これらのタイプに基づいた特殊タイプの列を除きます (SQLSTATE 42962)。

OID-column-definition

型付き表のオブジェクト ID 列を定義します。

REF IS *OID-column-name* **USER GENERATED**

オブジェクト ID 列 (OID) を表の最初の列として定義することを指定します。表階層のルート表では、OID が必須です (SQLSTATE 428DX)。この表は副表以外の型付き表 (OF 文節が必須) でなければなりません (SQLSTATE 42613)。この列の名前は *OID-column-name* という形式で定義されますが、構造タイプ *type-name1* のいかなる属性の名前とも同一であることはできません (SQLSTATE 42711)。さらに、この列はタイプ REF (*type-name1*)、NOT NULL で定義され、システム必須のユニーク索引 (デフォルトの索引名) が生成されます。この列はオブジェクト ID 列 または *OID* 列 として参照されます。USER GENERATED というキーワードは、行を挿入する際にユーザーが *OID* 列の初期値を提供しなければならないことを指しています。行を挿入した後は、*OID* 列を更新することはできません (SQLSTATE 42808)。

with-options

型付き表の列に適用される追加オプションを定義します。

column-name

追加オプションを指定する列の名前を指定します。 *column-name* (列名) は、同じくスーパー表の列ではない表の列の名前に対応していません (SQLSTATE 428DJ)。列名は、ステートメント内の 1 つの WITH OPTIONS 文節に 1 回だけしか指定できません (SQLSTATE 42613)。

タイプ定義 (CREATE TYPE) の一部としてオプションが既に指定されている場合には、ここで指定されているオプションは CREATE TYPE のオプションをオーバーライドします。

WITH OPTIONS *column-options*

指定した列にオプションを定義します。前述の *column-options* を参照してください。表が副表である場合、主キーまたはユニーク制約を指定することはできません (SQLSTATE 429B3)。

CREATE TABLE

DATA CAPTURE

データベース間のデータの複製に関する特殊な情報を、ログに書き込むかどうかを指定します。この文節は、副表を作成する際には指定できません (SQLSTATE 42613)。

表が型付き表である場合、このオプションはサポートされません (SQLSTATE 428DH または 42HDR)。

NONE

追加情報をログに記録しないことを指定します。

CHANGES

この表に対する SQL 変更についての追加情報をログに書き込むことを指定します。このオプションは、表を複製する場合で、キャプチャー・プログラムを使用してログからこの表に対する変更内容を取り込む場合に必須です。

カタログ・パーティション以外のパーティションにデータが置かれるように表が定義されている場合 (複数パーティションのデータベース・パーティション・グループ、またはカタログ・パーティション以外のパーティションを持つデータベース・パーティション・グループ)、このオプションはサポートされません (SQLSTATE 42997)。

表のスキーマ名 (暗黙または明示名) が 18 バイトより長い場合、このオプションはサポートされません (SQLSTATE 42997)。

WITH RESTRICT ON DROP

表をドロップできないこと、また、表を含む表スペースをドロップできないことを指定します。

IN *tablespace-name*

表を作成する表スペースを指定します。その表スペースは存在していなければならず、ステートメントの許可 ID がもつ USE 特権の対象の正規 (REGULAR) 表スペースでなければなりません。他の表スペースが指定されていない場合、表のすべての部分がこの表スペースに保管されます。副表は表階層のルート表から表スペースを継承するので、この文節を作成の際に指定することはできません (SQLSTATE 42613)。この文節を指定しない場合には、この表の表スペースは以下のように決められます。

```
IF ユーザーが USE 特権を持つ表スペース IBMDEFAULTGROUP が
  十分なページ・サイズをもって存在する場合
  THEN それを選択します
ELSE IF ユーザーが USE 特権を持つ表スペースが
  十分なページ・サイズをもって存在する場合
  (複数表スペース修飾の場合は下記を参照)
  THEN それを選択します
ELSE エラーを出します (SQLSTATE 42727)
```

ELSE IF 条件で複数の表スペースが指定されている場合、ステートメントの許可 ID がもつ USE 特権の対象の最小限必要なページ・サイズをもつ表スペースを選択します。複数の表スペースがそれにあてはまる場合、以下のどれに USE 特権が付与されているかに応じて優先順位が決められます。

1. 許可 ID
2. 許可 ID を保有するグループ
3. PUBLIC

それでも複数の表スペースがそれにあてはまる場合は、最終選択はデータベース・マネージャーによって行われます。

表スペースの決定は、以下の時点で変更することができます。

- 表スペースをドロップまたは作成するとき
- USE 特権を付与または取り消すとき

十分な表のページ・サイズは、行のバイト・カウントか列の数のいずれかによって決まります。詳細については、1535 ページの『行サイズ』を参照してください。

tablespace-options

索引または長形式列の値 (あるいはその両方) が保管される表スペースを指定します。表スペースのタイプについては、『CREATE TABLESPACE』を参照してください。

INDEX IN *tablespace-name2*

表に索引を作成する場合の表スペースを指定します。このオプションは、IN 文節に指定された 1 次表スペースが DMS 表スペースである場合にのみ使用できます。指定する表スペースは、既存であり、ステートメントの許可 ID がもつ USE 特権が対象とする REGULAR または LARGE DMS 表スペースであり、*tablespace-name1* と同じデータベース・パーティション・グループになければなりません (SQLSTATE 42838)。

表の索引を入れる表スペースの指定は、その表の作成時のみ行うことができる点に注意してください。索引用の表スペースに対する USE 特権の検査は、表の作成時にしか行われません。その後の索引の作成時に CREATE INDEX ステートメントの許可 ID が、表スペースに対する USE 特権をもつことをデータベース・マネージャーによって要求されることはありません。

LONG IN *tablespace-name3*

長形式列 (LONG VARCHAR、LONG VARGRAPHIC、LOB データ・タイプ、これらのいずれかをソース・タイプとする特殊タイプ、またはインラインで保管できない値をもつユーザー定義の構造タイプで定

CREATE TABLE

義されたすべての列) の値が保管される表スペースを指定します。このオプションは、IN 文節に指定された 1 次表スペースが DMS 表スペースである場合にのみ使用できます。表スペースは、既存であり、ステートメントの許可 ID がもつ USE 特権が対象とする LARGE DMS 表スペースであり、*tablespace-name1* と同じデータベース・パーティション・グループになければなりません (SQLSTATE 42838)。

表の長形式列と LOB 列を入れる表スペースの指定は、その表の作成時にのみ行うことができる点に注意してください。長形式列と LOB 列用の表スペースに対する USE 特権の検査は、表の作成時にしか行われません。その後の長形式列と LOB 列の追加時に ALTER TABLE ステートメントの許可 ID が、表スペースに対する USE 特権をもつことがデータベース・マネージャーで要求されることはありません。

PARTITIONING KEY (*column-name*,...)

表のデータが区分化されている場合に、区分化キーを指定します。各 *column-name* (列名) は、表の列を指定するものでなければなりません。また、同じ列を複数回指定することはできません。LONG VARCHAR、LONG VARGRAPHIC、BLOB、CLOB、DBCLOB、DATALINK、これらのタイプのいずれかに基づいた特殊タイプ、または構造タイプであるデータ・タイプの列を、区分化キーの一部として使用することはできません (SQLSTATE 42962)。副表では表階層のルート表から区分化キーが継承されるため、区分化キーを指定することはできません (SQLSTATE 42613)。

この文節の指定がなく、この表が複数パーティションのデータベース・パーティション・グループに存在する場合、その区分化キーは以下のように定義されます。

- 表が型付き表である場合、オブジェクト ID 列が区分化キーになります。
- 主キーが指定されている場合は、その主キーの最初の列が区分化キーになります。
- これら以外の場合、LOB、LONG VARCHAR、LONG VARGRAPHIC、DATALINK 列、これらのタイプのうちのいずれかに基づいた特殊タイプ、または構造タイプの列以外のデータ・タイプの最初の列が区分化キーになります。

デフォルトの区分化キーの要件を満たす列が存在しない場合、表は区分化キーなしで作成されます。このような表は、単一パーティションのデータベース・パーティション・グループに対して定義された表スペースでのみ許されます。

単一パーティションのデータベース・パーティション・グループに対して定義された表スペースの表の場合、長形式以外の一連の列はどれも区分

化キーの定義に使用することができます。このパラメーターの指定がない場合、区分化キーは作成されません。

区分化キーに関する制約事項については、1530 ページを参照してください。

USING HASHING

データ配分の区分化方式として、ハッシュ関数を使用することを指定します。これは、サポートされる唯一の区分化方式です。

REPLICATED

表が定義される表スペースのデータベース・パーティション・グループの各データベース・パーティションに対して、表に保管されたデータを物理的に複製することを指定します。つまり、これらのデータベース・パーティションにはそれぞれ、表のデータすべてのコピーが存在することになります。このオプションは、マテリアライズ照会表にのみ指定できます (SQLSTATE 42997)。

VALUE COMPRESSION

ほとんどのデータ・タイプで NULL および長さ 0 のデータ値を、さらに効率的に保管するよう指定します。また、使用される行形式を判断します。表が型付き表である場合、このオプションがサポートされるのは、型付き表階層のルート表だけです (SQLSTATE 428DR)。

データ・タイプが BLOB、CLOB、DBCLOB、LONG VARCHAR、または LONG VARGRAPHIC の列の長さ 0 のデータ値を、最小限のスペースを使用して保管します。NULL 値は、追加バイトを使用することなく、それぞれ保管されます。これをサポートする行形式は、各データ・タイプのバイト・カウントを識別し、更新の際にデータ・フラグメントの原因となる傾向があります。新しい行形式 (COMPRESS SYSTEM DEFAULT オプションによって列を指定する) を使用しても、列のシステム・デフォルト値をより効率的に保管できます。

NOT LOGGED INITIALLY

表を作成する作業単位と同一の作業単位の挿入、削除、更新、索引の作成、索引のドロップ、または表の変更操作によって表に対して行われた変更はログ記録されません。このオプションを使用する際の他の考慮事項については、このステートメントの『注』のセクションを参照してください。

カタログの変更と、ストレージに関連する情報は、以後の作業単位で表に対して行われた操作と同様にすべてログ記録されます。

注: 活動化された NOT LOGGED INITIALLY 属性を持つ表に対してログに記録されない活動が生じ、ステートメントに障害が起こる (ロールバックが発生する)、または ROLLBACK TO SAVEPOINT が実行される場合には、その作業単位全体がロールバックされます

CREATE TABLE

(SQL1476N)。さらに、NOT LOGGED INITIALLY 属性が活動化されている表は、ロールバックされた後にアクセス不能としてマークされ、ドロップしかできなくなります。したがって、NOT LOGGED INITIALLY 属性が活動化されている作業単位内のエラーは、最小限に抑えられます。

OPTIONS

連合システムでは、DB2 を使用してリモート表を作成、変更、またはドロップすることができます。この機能は、*透過性 DDL* として知られています。

リモート表がデータ・ソース上に作成される前に、連合サーバーはそのデータ・ソースへアクセスするように構成されなければなりません。この構成には、データ・ソースのラッパーの作成、リモート表を置くサーバーのサーバー定義の指定、DB2 とデータ・ソース・サーバー間のユーザー・マッピングの作成が含まれます。

CREATE TABLE ステートメントに含むことができるものに関して、*透過性 DDL* には以下のようないくつかの制約があります。

- 列および主キーのみがリモート表に作成可能です。
- マテリアライズ照会表は、リモート・データ・ソースには作成不可能です。
- リモート・データ・ソースは、CREATE TABLE ステートメント内の列データ・タイプおよび主キー・オプションをサポートする必要があります。データ・ソースの、サポートしていない要求への応答方法によって、エラーが返されるか、または要求が無視される可能性があります。

リモート・サーバーは、OPTIONS 文節に指定する必要があります。作成される表のリモート名またはリモート・スキーマをオーバーライドするには、OPTIONS 文節を使用できます。

リモート表が *透過性 DDL* を使用して、DB2 で作成された場合、ニックネームは自動的にそのリモート表に対して作成されます。

REMOTE_SERVER 'value'

CREATE SERVER ステートメントに指定されたサーバー名。

REMOTE_SCHEMA 'value'

作成されるリモート表のスキーマ名を指定します。しかし、このパラメーターはオプションですが、スキーマ名を指定することをお勧めします。指定されていない場合は、ローカル AUTHID (英大文字) がリモート・スキーマ名に使用されます。

REMOTE_TABNAME 'value'

作成されるリモート表の非修飾名を指定します。値が指定されないと、ローカル名が使用されます。

unique-constraint

ユニーク制約または主キー制約を定義します。表に区分化キーがある場合、ユニーク・キーまたは主キーは区分化キーのスーパーセットである必要があります。副表

である表では、ユニーク制約または主キー制約を指定することはできません (SQLSTATE 429B3)。主キーまたはユニーク・キーは、ディメンションのサブセットにはなりません (SQLSTATE 429BE)。表がルート表である場合、表とそのすべての副表に対して制約が適用されます。

CONSTRAINT *constraint-name*

主キー制約、またはユニーク制約の名前を指定します。

UNIQUE (*column-name,...*)

指定した列で構成されるユニーク・キーを定義します。指定する列は NOT NULL として定義されていなければなりません。各 *column-name* (列名) は、表の列を指定するものでなければなりません。また、同じ列を複数回指定することはできません。

指定する列の数は 16 を超えてはならず、保管されるそれらの長さの合計は 1024 を超えてはなりません (保管される長さの詳細は、1536 ページの『バイト・カウント』を参照)。変数キー部分をもつユニーク・キーは、DB2_INDEX_2BYTEVARLEN レジストリー変数が ON になっている場合には、255 より大きいサイズが可能です。列の長さ属性が 1024 バイト以内に収まる場合でも、LOB、LONG VARCHAR、LONG VARGRAPHIC、DATALINK、これらのタイプのうちのいずれかに基づく特殊タイプ、または構造タイプは、ユニーク・キーの一部として使用できません (SQLSTATE 54008)。

ユニーク・キーにある一連の列は、主キーまたは他のユニーク・キーの一連の列と同じにすることはできません (SQLSTATE 01543)。LANGLEVEL が SQL92E または MIA の場合には、エラーが戻されます (SQLSTATE 42891)。

表が副表である場合、ユニーク制約はスーパー表から継承されるので、(SQLSTATE 429B3)、ユニーク制約を指定することはできません。

カタログに記録されている表の記述には、ユニーク・キーとそのユニーク索引が含まれます。ユニーク索引は、それぞれの列について昇順に指定されたシーケンスで、列に対して自動的に作成されます。索引の名前は、表の作成時にスキーマに存在する既存の索引と競合しない場合、*constraint-name* (制約名) と同じになります。索引名が競合する場合は、名前は SQL の後に文字のタイム・スタンプ (yymmddhhmmssxxx) が続き、スキーマ名として SYSIBM を伴う名前になります。

PRIMARY KEY (*column-name,...*)

指定された列で構成される主キーを定義します。この文節を複数回指定することはできず、指定する列は NOT NULL として定義されていなければなりません。各 *column-name* (列名) は、表の列を指定していなければなりません。また、同じ列を複数回指定することはできません。

指定する列の数は 16 を超えてはならず、保管されるそれらの長さの合計は 1024 を超えてはなりません (保管される長さの詳細は、1536 ページの『バイ

CREATE TABLE

ト・カウント』を参照)。変数キー部分をもつ主キーは、DB2_INDEX_2BYTEVARLEN レジストリー変数が ON になっている場合には、255 より大きいサイズが可能です。列の長さ属性が 1024 バイト以内に収まる場合でも、LOB、LONG VARCHAR、LONG VARGRAPHIC、DATALINK、これらのタイプのうちのいずれかに基づく特殊タイプ、または構造タイプは、主キーの一部として使用できません (SQLSTATE 54008)。

主キーの一連の列は、ユニーク・キーの一連の列と同じであってはなりません (SQLSTATE 01543)。LANGLEVEL が SQL92E または MIA の場合には、エラーが戻されます (SQLSTATE 42891)。

1 つの表には、主キーを 1 つだけ定義することができます。

表が副表である場合、主キーはスーパー表から継承されるので (SQLSTATE 429B3)、主キーを指定することはできません。

カタログに記録されている表の記述には、主キーとその主索引が含まれます。ユニーク索引は、それぞれの列について昇順に指定されたシーケンスで、列に対して自動的に作成されます。索引の名前は、表の作成時にスキーマに存在する既存の索引と競合しない場合、*constraint-name* (制約名) と同じになります。索引名が競合する場合は、名前は SQL の後に文字のタイム・スタンプ (yyymmddhhmmssxxx) が続き、スキーマ名として SYSIBM を伴う名前になります。

表に区分化キーがある場合、*unique-constraint* (ユニーク制約) の列は区分化キーの列のスーパーセットである必要があります。列の順序は重要ではありません。

referential-constraint

参照制約を定義します。

CONSTRAINT *constraint-name*

参照制約の名前を指定します。

FOREIGN KEY (*column-name,...*)

指定した *constraint-name* (制約名) の参照制約を定義します。

T1 を、ステートメントの対象となる表であると想定します。参照制約の外部キーは、指定された列で構成されます。列名リストの各名前は、T1 の列を指定していなければならない、同じ列を複数回指定することはできません。指定する列の数は 16 を超えてはならず、保管されるそれらの長さの合計は 1024 を超えてはなりません (保管される長さの詳細は、1536 ページの『バイト・カウント』を参照)。外部キーは、255 バイトよりも大きい長さの可変長列で定義できます。LOB、LONG VARCHAR、LONG VARGRAPHIC、DATALINK、これらのタイプのうちのいずれかに基づく特殊タイプ、または構造タイプの列を、外部キーの一部として使用することはできません (SQLSTATE 42962)。外部キーの列の数は、親キーの列の数と同じでなければならない、対応する列のデータ・タイプは互換性があることが必要です (SQLSTATE 42830)。2 つの列

の記述は、それらの列が互換性のあるデータ・タイプ (両方の列が数字、文字ストリング、グラフィック、日付 / 時間であるか、または同じ特殊タイプ) であれば互換性があります。

references-clause

参照制約の親表と親キーを指定します。

REFERENCES *table-name*

REFERENCE 文節に指定される表は、カタログに記述された基本表を識別している必要がありますが、カタログ表を示すものであってはなりません。

参照制約の外部キー、親キー、および親表が、以前に指定した参照制約の外部キー、親キー、および親表と同じである場合、参照制約は重複しません。重複した参照制約は無視され、警告が出されます (SQLSTATE 01543)。

以下の説明では、T2 は指定した親表を示し、T1 は作成する (または変更する) 表を示します。 (T1 と T2 は同じ表である可能性もあります)。

指定された外部キーの列の数は、T2 の親キーと同じ数でなければなりません。また、外部キーの *n* 番目の列の記述は、その親キーの *n* 番目の列の記述と互換性がなければなりません。この規則において、日付 / 時刻の列はストリング列と互換性があるとは見なされません。

(column-name,...)

参照制約の親キーは、指定された列で構成されます。各 *column-name* は、T2 の列を指定する非修飾名でなければなりません。同じ列を重複して指定することはできません。

列名のリストは、親キーまたは T2 に存在するユニーク制約の一連の列と一致している (順序は任意) 必要があります (SQLSTATE 42890)。列名のリストの指定がない場合、T2 に親キーがある必要があります (SQLSTATE 42888)。列名リストを省略すると、指定されているとおりのシーケンスでその主キーの列が暗黙に指定されます。

FOREIGN KEY 文節で指定される参照制約は、T2 が親であり、T1 が従属である関係を定義します。

rule-clause

従属表に対するアクションを指定します。

ON DELETE

親表の行が削除された場合、従属表でどのようなアクションを行うかを指定します。以下の 4 つのアクションがあります。

- NO ACTION (デフォルト値)
- RESTRICT
- CASCADE

CREATE TABLE

- SET NULL

削除規則は、T2 の行が DELETE または伝搬による削除操作の対象であり、その行の従属行が T1 にある場合に、適用されます。 p は、そのような T2 の行を表すと想定します。

- RESTRICT または NO ACTION を指定すると、エラーになり、行は削除されません。
- CASCADE を指定すると、T1 の p の従属行に削除操作が伝搬します。
- SET NULL が指定された場合、T1 の p のそれぞれの従属行の外部キーの NULL 可能な列が NULL 値に設定されます。

SET NULL は、外部キーの列に NULL 可能な列がない限り指定してはなりません。この文節を省略すると、暗黙に ON DELETE NO ACTION が指定されます。

複数の表の関係するサイクルでは、そのサイクル内の削除規則がすべて CASCADE でない限り、表がそれ自体に連結削除されてはなりません。したがって、新しい関係がサイクルを形成し、T2 がすでに T1 に連結削除されている場合には、削除規則が CASCADE で、しかもそのサイクルの他の削除規則がすべて CASCADE であるのではない限り、制約を定義することはできません。

T1 が複数のパスによって T2 に連結削除されている場合、それらのパスの全体または一部を構成する関係のうち T1 が従属であるものの削除規則は同じでなければならない、SET NULL にすることはできません。NO ACTION アクションと RESTRICT アクションの処理は同じです。したがって、削除規則が r の関係で T1 が T3 の従属表であるとすると、 r が SET NULL のときに、以下の条件のいずれかが存在するなら、参照制約を定義することはできません。

- T2 と T3 は同じ表です。
- T2 は T3 の子孫であり、T3 からの行の削除は T2 にカスケードされます。
- T3 は T2 の子孫であり、T2 からの行の削除は T3 にカスケードされます。
- T2 と T3 はともに同一の表の従属表であり、その表からの行の削除は T2 および T3 へカスケードされます。

r が SET NULL 以外の場合、参照制約は定義できますが、FOREIGN KEY 文節で暗黙的または明示的に指定される削除規則は r と同一でなければなりません。

親表または従属表が型付き表階層のメンバーである参照制約に対する上記の規則の適用の場合、それぞれの階層内の任意の表に対して適用されるすべての参照制約が考慮に入られます。

ON UPDATE

親表の行が更新された場合に従属表に対して行うアクションを指定します。この文節はオプションです。ON UPDATE NO ACTION はデフォルト値であり、ON UPDATE RESTRICT はそれに代えて指定できる唯一のものであります。

NO ACTION と RESTRICT の差異については、このステートメントの『注』のセクションを参照してください。

constraint-attributes

参照保全またはチェック制約に関連付けられた属性を定義します。

ENFORCED

この制約は、挿入、更新、削除などの通常の操作中にデータベース・マネージャーによって課せられます。

NOT ENFORCED

この制約は、挿入、更新、削除などの通常の操作中にデータベース・マネージャーによって課せられません。この制約に適合することが分かっている表データだけに、制約を指定してください。

ENABLE QUERY OPTIMIZATION

適切な状況下では、制約を照会最適化に使用することができます。

DISABLE QUERY OPTIMIZATION

制約を照会の最適化に使用できません。

check-constraint

チェック制約を定義します。*check-constraint* (チェック制約) は、偽以外に評価されなければならない *search-condition* (検索条件) です。

CONSTRAINT *constraint-name*

チェック制約の名前を指定します。

CHECK (*check-condition*)

チェック制約を定義します。チェック条件 (*check-condition*) は、以下の場合を除いて、検索条件 です。

- 列参照が、作成する表の列でなければならない場合
- *search-condition* に TYPE 述部を入れることはできません。
- これには、以下のどれも入れることはできません (SQLSTATE 42621)。
 - 副照会
 - 参照解除操作または、有効範囲をもつ参照引き数がオブジェクト ID (OID) 列以外の列である Deref 関数

CREATE TABLE

- SCOPE 文節をもつ CAST 指定
- 列関数
- deterministic 関数でない関数
- 外部アクションをもつと定義された関数
- CONTAINS SQL または READS SQL DATA のいずれかによって定義されたユーザー定義関数
- ホスト変数
- パラメーター・マーカー
- 特殊レジスター
- ID 列以外の生成列の参照

column-definition の一部としてチェック制約を指定する場合、その同じ列に対してのみ列参照を行うことができます。表定義の一部として指定されたチェック制約には、それ以前に CREATE TABLE ステートメントで定義されている列を指定する列参照を含めることができます。チェック制約の矛盾、重複条件、または同等条件についてはチェックされません。したがって、矛盾したチェック制約や冗長なチェック制約が定義可能であるため、実行時にエラーになる可能性があります。

チェック条件 "IS NOT NULL" も指定できますが、列の NOT NULL 属性を使用することによって直接的に NULL 値可を指定するようにしてください。たとえば、salary が NULL に設定された場合に、CHECK (salary + bonus > 30000) は受け入れられます。これは、CHECK 制約は満たされるか未知かのどちらかでなければならず、この場合 salary は未知であるためです。一方、給与 (salary) が NULL に設定されている場合に、CHECK (salary IS NOT NULL) は偽となり、制約違反と見なされます。

チェック制約は、表に対して行の挿入または更新が行われる時点で適用されます。表で定義されるチェック制約は、その表の副表すべてに自動的に適用されます。

規則:

- すべての構造タイプ列のインライン長さも含め、列のバイト・カウントの合計は、表スペースのページ・サイズに基づく行サイズの限界を超えてはなりません (SQLSTATE 54010)。詳細については、1536 ページの『バイト・カウント』を参照してください。型付き表の場合、表階層のルート表の列や表階層内の各副表で新たに追加される列すべてに対しては、バイト・カウントが適用されます (追加される副表列は、NULL 値不可として定義されたとしても、バイト・カウントの際には NULL 値可と見なされます)。また、各行がどの副表からきたものかを識別するため、4 バイトのオーバーヘッドが追加されます。

- 表内に存在する列の数は、1,012 個を超えてはなりません (SQLSTATE 54011)。型付き表の場合は、表階層内のすべての副表タイプに含まれている属性の合計が 1,010 個を超えてはなりません。
- 型付き表のオブジェクト ID 列は更新できません (SQLSTATE 42808)。
- 表に対して定義されたユニーク・キー制約または主キー制約は、区分化キーのスーパーセットでなければなりません (SQLSTATE 42997)。
- 以下の表には、*file-link-options* でサポートされている DATALINK オプションの組み合わせが示されています (SQLSTATE 42613)。WRITE PERMISSION ADMIN は、READ PERMISSION DB とだけ組み合わせることが可能です。(RECOVERY と ON UNLINK 文節の他の組み合わせはサポートされません。)

表 270. 有効な DATALINK ファイル制御オプションの組み合わせ：この表にない組み合わせはすべてサポートされず、SQLSTATE 42613 となります。

INTEGRITY	READ PERMISSION	WRITE PERMISSION	RECOVERY	ON UNLINK
ALL	FS	FS	NO	該当なし
ALL	FS	BLOCKED	NO	RESTORE
ALL	FS	BLOCKED	YES	RESTORE
ALL	DB	BLOCKED	NO	RESTORE
ALL	DB	BLOCKED	NO	DELETE
ALL	DB	BLOCKED	YES	RESTORE
ALL	DB	BLOCKED	YES	DELETE
ALL	DB	ADMIN	NO	RESTORE
ALL	DB	ADMIN	NO	DELETE
ALL	DB	ADMIN	YES	RESTORE
ALL	DB	ADMIN	YES	DELETE

以下の規則は、パーティション・データベースに対してのみ適用されます。

- LOB、LONG VARCHAR、LONG VARGRAPHIC、DATALINK、これらのタイプのうちのいずれかに基づく特殊タイプ、または構造タイプの列だけで構成された表は、単一パーティションのデータベース・パーティション・グループで定義されている表スペース内でしか作成することができません。
- 複数パーティションのデータベース・パーティション・グループに対して定義された、表スペースの表の区分化キー定義は変更できません。
- 型付き表の区分化キー列は OID 列にする必要があります。

注:

- 互換性

CREATE TABLE

- 以前のバージョンの DB2 との互換性:
 - CONSTRAINT キーワードは、参照文節を定義する *column-definition* から省略できます。
 - *constraint-name* (制約名) を FOREIGN KEY に続けて (CONSTRAINT キーワードなし) 指定することができます。
 - SUMMARY は CREATE の後に任意に指定できます。
- 以前のバージョンの DB2 との互換性と整合性:
 - コンマは、*identity-attributes* 文節では複数のオプションを分離するのに使用できます。
- DB2 UDB (OS/390 および z/OS 版) との互換性:
 - 以下の構文はデフォルトの振る舞いとして受け入れられます。
 - CCSID ASCII
 - CCSID UNICODE
 - IN database-name.tablespace-name
 - IN DATABASE database-name
 - FOR MIXED DATA
 - FOR SBCS DATA
 - 以下の構文もサポートされています。
 - NOMINVALUE、NOMAXVALUE、NOCYCLE、NOCACHE、および NOORDER
- まだ存在していないスキーマ名を用いて表を作成すると、ステートメントの許可 ID に IMPLICIT_SCHEMA 権限がある場合に限り、そのスキーマが暗黙的に作成されます。そのスキーマの所有者は SYSIBM になります。スキーマに対する CREATEIN 特権が PUBLIC に付与されます。
- 外部キーが指定されると、
 - 親表に削除使用のあるパッケージはすべて無効になります。
 - 親キーの少なくとも 1 つの列に対して更新使用の指定があるパッケージは、すべて無効になります。
- 副表を作成すると、表階層内の表のいずれかに従属しているパッケージがすべて無効になります。
- それぞれ 4,000 および 2,000 より大きい数値の VARCHAR および VARGRAPHIC 列は、SYSFUN スキーマの関数での入力パラメーターとして使用しないでください。関数にこの長さを超過する引き数値を指定して呼び出すと、エラーが発生します (SQLSTATE 22001)。

- ID 列は、複数パーティションをもつデータベースではサポートされません (SQLSTATE 42997)。データベースで複数の区画が存在する場合は、ID 列を作成できません。ID 列を組み込まれたデータベースを、複数の区画を指定して開始することはできません。
- 参照制約の削除規則または更新規則として NO ACTION または RESTRICT を使用すると、制約がいつ適用されるかが決まります。RESTRICT の削除規則または更新規則は、CASCADE や SET NULL などの変更規則を伴う参照制約を含む他のすべての制約の前に適用されます。NO ACTION の削除規則または更新規則は、他の参照制約の後で適用されます。このことが、削除または更新中に違いとして現れる場合はほとんどありません。この違いが明白になる例の 1 つとして、互いに関連する複数の表の UNION ALL として定義されたビューからの行の削除があります。

```
Table T1 is a parent of table T3; delete rule as noted below.
Table T2 is a parent of table T3; delete rule CASCADE.
```

```
CREATE VIEW V1 AS SELECT * FROM T1 UNION ALL SELECT * FROM T2
```

```
DELETE FROM V1
```

表 T1 が RESTRICT の削除規則を伴う表 T3 の親である場合に、T3 に T1 の親キーの子行があると、制約違反 (SQLSTATE 23001) になります。

表 T1 が表 T3 の親で、T3 の削除規則が NO ACTION である場合、T1 からの削除に対して NO ACTION 削除規則が適用される前に T2 から行を削除すると、削除規則 CASCADE によって、その子行が削除される場合があります。T2 からの削除で、T3 の T1 の親キーの子行すべてが削除されたわけではない場合は、制約違反 (SQLSTATE 23504) になります。

戻される SQLSTATE は、削除規則または更新規則が RESTRICT か NO ACTION かによって異なります。

- 複数パーティションのデータベース・パーティション・グループに対して定義された表スペースの複数の表の場合、区分化キーを選択する際に表のコロケーションについて考慮する必要があります。以下に考慮事項をリストします。
 - コロケーションのためには、各表は同じデータベース・パーティション・グループにある必要があります。表スペースは異なっても構いませんが、同じデータベース・パーティション・グループに定義されている必要があります。
 - コロケーションのためには、各表の区分化キーの列の数は同じである必要があり、対応するキーの列はパーティション互換である必要があります。
 - 区分化キーの選択も、結合のパフォーマンスに影響します。表を他の表と頻繁に結合する場合は、結合する列を両方の表の区分化キーにすることを考慮してください。
- NOT LOGGED INITIALLY 文節は、表の中に、FILE LINK CONTROL 属性のある DATALINK 列が存在するときには使えません (SQLSTATE 42613)。

CREATE TABLE

- 代替のソース (別の表やファイル) からのデータを使用して大きな結果セットを作成する必要があり、表のリカバリーが不要な場合は、NOT LOGGED INITIALLY オプションが有効です。このオプションを使用すると、データのログ記録にかかるオーバーヘッドが節減されます。このオプションを指定する場合、以下の考慮事項が適用されます。
 - 作業単位がコミットされると、その作業単位の過程で表に対して行われた変更はすべてディスクにフラッシュされます。
 - ロールフォワード・ユーティリティーを実行した際に、データベース中の表がロード・ユーティリティーによって移植されたか、または NOT LOGGED INITIALLY オプションを使用して作成されたことを示すログ記録が見つかったら、表は使用不能としてマークされます。その後 DROP TABLE ログが見つかったら、表はロールフォワード・ユーティリティーによってドロップされます。除去しない場合、データベースの回復後に表にアクセスを試みると、エラーが出されます (SQLSTATE 55019)。許される唯一の操作は表のドロップです。
 - データベースまたは表スペースのバックアップの一環として、このような表をバックアップすると、表のリカバリーが可能になります。
- CURRENT REFRESH AGE を ANY にセットし、CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION をシステム保守済みマテリアライズ照会表に含まれるようにセットすると、照会の処理を最適化するとき、ENABLE QUERY OPTIMIZATION を指定して定義された REFRESH DEFERRED システム保守済みマテリアライズ照会表を使用できます。CURRENT REFRESH AGE を ANY にセットし、CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION をユーザー保守済みマテリアライズ照会表に含まれるようにセットすると、照会の処理を最適化するとき、ENABLE QUERY OPTIMIZATION を指定して定義された REFRESH DEFERRED ユーザー保守済みマテリアライズ照会表を使用できます。ENABLE QUERY OPTIMIZATION を指定して定義された REFRESH IMMEDIATE マテリアライズ照会表は、必ず最適化の対象として考慮に入れられます。この最適化で REFRESH DEFERRED または REFRESH IMMEDIATE マテリアライズ照会表を使用できるようにするには、すでに説明された規則以外の特定の規則に全選択を適合させる必要があります。全選択の条件は、以下の規則にしたがっていなければなりません。
 - GROUP BY 文節を指定した副選択、または 1 つの表参照を指定した副選択になっている。
 - 選択リストのどこにも DISTINCT が含まれていない。
 - 特殊レジスターが含まれていない。
 - deterministic 関数でない関数が含まれていない。

マテリアライズ照会表の作成時に指定した照会が上記の規則に適合しなければ、警告が戻されます (SQLSTATE 01633)。

- マテリアライズ照会表が `REFRESH IMMEDIATE` で定義されている場合、基礎となる表の挿入、更新、または削除操作を行うことになる変更をしようとすると、エラーになる可能性があります。エラーが発生すると、基礎となる表の挿入、更新、または削除は失敗します。
- 連合システムでは、リレーショナル・データ・ソースまたはローカル表のニックネームは、マテリアライズ照会表を作成する基礎表として使用できます。非リレーショナル・データ・ソースのニックネームは、サポートされていません。ニックネームが基礎表のうちの 1 つである場合は、`REFRESH DEFERRED` オプションを使用する必要があります。
- 親表または従属表が表階層の一部を成すように参照制約を定義することができます。その場合、参照制約は以下のような効果を生じます。
 1. `INSERT`、`UPDATE`、および `DELETE` ステートメントの効果は以下のとおりです。
 - `PT` が親表で `DT` が従属表である参照制約が存在する場合、非 `NULL` の外部キーをもつ `DT` の行 (またはその副表のいずれか) ごとに、それに合致する親キーをもつ行が `PT` (またはその副表のいずれか) 内に必ず存在することが制約によって確実になります。アクションの始動の仕方に関係なく、この規則は、`PT` または `DT` の行に影響を与えるすべてのアクションに対して適用されます。
 2. `DROP TABLE` ステートメントの効果は以下のとおりです。
 - ドロップ済みの表が親表または従属表である参照制約では、制約はドロップされます。
 - ドロップ済みの表のスーパー表が親表である参照制約では、そのドロップ済みの表の行は、スーパー表からの削除を考慮されます。参照制約がチェックされて、削除行ごとに削除規則が呼び出されます。
 - ドロップ済みの表のスーパー表が従属表である参照制約の場合、制約はチェックされません。従属表から行を削除しても、参照制約違反にはなりません。
- **特権:** 表が作成されると、その表の定義者には `CONTROL` 特権が付与されます。副表が作成されると、各ユーザーまたはグループが持っているそのすぐ上のスーパー表に対する `SELECT` 特権が副表に対しても自動的に付与され、その場合は表定義者から特権が付与されたこととなります。
- **行サイズ:** 表の行で許可される最大バイト数は、表が作成される表スペースのページ・サイズによって決まります (`tblspace-name1`)。以下のリストでは、各表スペースのページ・サイズに関連した行サイズの制限と列数の制限を示します。

表 271. 各表スペースのページ・サイズの列数および行サイズの制限

ページ・サイズ	行サイズの制限	列数の制限
4K	4 005	500
8K	8 101	1 012
16K	16 293	1 012

CREATE TABLE

表 271. 各表スペースのページ・サイズの列数および行サイズの制限 (続き)

ページ・サイズ	行サイズの制限	列数の制限
32K	32 677	1 012

表の実際の列数については、以下の公式によってさらに制限されます。

$$\text{Total Columns} * 8 + \text{Number of LOB Columns} * 12 + \text{Number of Datalink Columns} * 28 \leq \text{row size limit for page size.}$$

- **バイト・カウント:** 以下の表は、NULL 値を使えない列のバイト・カウントを、列のデータ・タイプ別に示したものです。値の圧縮をしない表では、NULL が許される各列には追加のバイトが必要です。

表が構造タイプに基づいている場合には、副表が定義されているか否かにかかわらず、副表の行を識別するために 4 バイトのオーバーヘッドが確保されます。さらに、追加される副表列は、NULL 値不可と定義されたとしたら、バイト・カウント用に NULL 値可能なものと見なされる必要があります。

保管長の計算の際に、索引または制約 (制約は索引によって実施されることに注意してください) で 1024 バイトの制限を超えていないこと、オーバーヘッドは 4 バイトではなく 2 バイトであることを確認してください。

表 272. データ・タイプごとの列のバイト・カウント

データ・タイプ	VALUE COMPRESSION が表に対して活動状態 になるときのバイト・ カウント	VALUE COMPRESSION が表に対して暗黙的または 明示的に非活動状態である ときのバイト・カウント。 列が NULL を受け入れる 場合、示されたバイト・ カウントに 1 が加算される
ROW OVERHEAD	2	0
INTEGER	6	4
SMALLINT	4	2
BIGINT	10	8
REAL	6	4
DOUBLE	10	8
DECIMAL	$(p/2)+3$ の整数部分 (p は精度)	$(p/2)+1$ の整数部分 (p は精度)
CHAR(n)	$n+2$	n
VARCHAR(n)	$n+2$	$n+4$ (表); $n+2$ (索引)
LONG VARCHAR	22	24
GRAPHIC(n)	$n*2+2$	$n*2$

表 272. データ・タイプごとの列のバイト・カウント (続き)

データ・タイプ	VALUE COMPRESSION が表に対して活動状態に なるときのバイト・ カウント	VALUE COMPRESSION が表に対して暗黙的または 明示的に非活動状態である ときのバイト・カウント。 列が NULL を受け入れる 場合、示されたバイト・ カウントに 1 が加算される
VARGRAPHIC(<i>n</i>)	$(n*2)+2$	$(n*2)+4$ (表); $(n*2)+2$ (索引)
LONG VARGRAPHIC	22	24
DATE	6	4
TIME	5	3
TIMESTAMP	12	10
DATALINK(<i>n</i>)	$n+52$	$n+54$
LOB の最大長 1024	70 ¹	72
LOB の最大長 8192	94	96
LOB の最大長 65 536	118	120
LOB の最大長 524 000	142	144
LOB の最大長 4 190 000	166	168
LOB の最大長 134 000 000	198	200
LOB の最大長 536 000 000	222	224
LOB の最大長 1 070 000 000	254	256
LOB の最大長 1 470 000 000	278	280
LOB の最大長 2 147 483 647	314	316

¹ 各 LOB 値は、その基底レコードに、実際の値の位置へのポインターとなる LOB 記述子を持っています。その記述子のサイズは、列に定義されている最大長によって異なります。

特殊タイプの場合、バイト・カウントは特殊タイプのソース・タイプの長さに相当します。参照タイプの場合には、バイト・カウントは、参照タイプの基礎となる組み込みデータ・タイプの長さに相当します。構造タイプでは、バイト・カウントは `INLINE LENGTH + 4` です。INLINE LENGTH は、`column-options` 文節内の列に指定された (または暗黙で計算された) 値です。

CREATE TABLE

- **ディメンション列:** ディメンション列の各特殊値は、表の別々のブロックに割り当てられるので、“INTEGER(ORDER_DATE)/100”などの式でのクラスタリングはお勧めできません。この場合、表に生成された列が定義可能で、その後この生成された列は ORGANIZE BY DIMENSIONS 文節で使用されるかもしれません。式が表の列に関連して単調な場合には、この列の述部範囲を満たすためにディメンション索引が DB2 によって使用される可能性があります。たとえば、式が単に *column-name + some-positive-constant* の場合には、これは単調な増加です。ユーザー定義関数、特定の組み込み機能、および 1 つの式で複数の列を使用すると、単調化やその検出を防げます。

非単調な式を持つ、または単調化を識別できない、生成された列に関するディメンションを作成できますが、スライスの範囲照会やこうしたディメンションのセル境界はサポートされません。同等性および IN 述部は、スライスまたはセルによって処理できます。

生成された関数 fn に関して以下の事柄が真の場合には、生成された列は単調です。

- 単調な増加。

値 x1 および x2 のペアのすべての可能性において、 $x2 > x1$ ならば $fn(x2) > fn(x1)$ となります。たとえば、以下のような場合です。

SALARY - 10000

- 単調な減少。

値 x1 および x2 のペアのすべての可能性において、 $x2 > x1$ ならば $fn(x2) < fn(x1)$ となります。たとえば、以下のような場合です。

-SALARY

- 単調な非減少。

値 x1 および x2 のペアのすべての可能性において、 $x2 > x1$ ならば $fn(x2) \geq fn(x1)$ となります。たとえば、以下のような場合です。

SALARY/1000

- 単調な非増加。

値 x1 および x2 のペアのすべての可能性において、 $x2 > x1$ ならば $fn(x2) \leq fn(x1)$ となります。たとえば、以下のような場合です。

-SALARY/1000

式 "PRICE*DISCOUNT" は単調ではありません。表の複数の列が関係するからです。

例:

例 1: DEPARTX 表スペースに表 TDEPT を作成します。DEPTNO、DEPTNAME、MGRNO、および ADMRDEPT は列の名前です。CHAR は、列が文字データを含むこ

とを意味しています。NOT NULL は、列に NULL 値を含めることができないことを示します。VARCHAR は、列のデータが可変長文字データであることを意味します。主キーは、列 DEPTNO で構成されます。

```
CREATE TABLE TDEPT
  (DEPTNO CHAR(3) NOT NULL,
   DEPTNAME VARCHAR(36) NOT NULL,
   MGRNO CHAR(6),
   ADMRDEPT CHAR(3) NOT NULL,
   PRIMARY KEY(DEPTNO))
IN DEPARTX
```

例 2: SCHED 表スペースに表 PROJ を作成します。PROJNO、PROJNAME、DEPTNO、RESPEMP、PRSTAFF、PRSTDATE、PRENDATE、および MAJPROJ は列の名前です。CHAR は、列が文字データを含むことを意味しています。DECIMAL は、列のデータがバック 10 進数データであることを意味します。5.2 の 5 は 10 進数の桁数、2 は小数点以下の桁数を示します。NOT NULL は、列に NULL 値を含めることができないことを示します。VARCHAR は、列のデータが可変長文字データであることを意味します。DATE は、列のデータが 3 つの部分からなる形式 (年、月、日) の日付情報であることを示しています。

```
CREATE TABLE PROJ
  (PROJNO CHAR(6) NOT NULL,
   PROJNAME VARCHAR(24) NOT NULL,
   DEPTNO CHAR(3) NOT NULL,
   RESPEMP CHAR(6) NOT NULL,
   PRSTAFF DECIMAL(5,2) ,
   PRSTDATE DATE ,
   PRENDATE DATE ,
   MAJPROJ CHAR(6) NOT NULL)
IN SCHED
```

例 3: 不明の給与はすべて 0 と見なされる EMPLOYEE_SALARY という名前の表を作成します。表スペースが指定されていないので、IN tablespace-name1 文節について記述された規則に基づいてシステムが選択した表スペースに、表が作成されます。

```
CREATE TABLE EMPLOYEE_SALARY
  (DEPTNO CHAR(3) NOT NULL,
   DEPTNAME VARCHAR(36) NOT NULL,
   EMPNO CHAR(6) NOT NULL,
   SALARY DECIMAL(9,2) NOT NULL WITH DEFAULT)
```

例 4: 給与 (SALARY) と距離 (MILES) の合計用の特殊タイプを作成し、デフォルト表スペースに作成される表の列として使用します。動的 SQL ステートメントでは、CURRENT SCHEMA 特殊レジスターが JOHNDOE で、CURRENT PATH がデフォルト値であると想定します ("SYSIBM","SYSFUN","JOHNDOE")。

SALARY の値の指定がない場合には、それを 0 に設定します。また、LIVING_DIST の値の指定がない場合には、それを 1 マイルに設定します。

CREATE TABLE

```
CREATE DISTINCT TYPE JOHNDOE.T_SALARY AS INTEGER WITH COMPARISONS
```

```
CREATE DISTINCT TYPE JOHNDOE.MILES AS FLOAT WITH COMPARISONS
```

```
CREATE TABLE EMPLOYEE
  (ID      INTEGER NOT NULL,
   NAME    CHAR (30),
   SALARY  T_SALARY NOT NULL WITH DEFAULT,
   LIVING_DIST MILES  DEFAULT MILES(1) )
```

例 5: 画像 (IMAGE) と音声 (AUDIO) 用の特殊タイプを作成し、表の列として使用します。表スペースが指定されていないので、`IN tablespace-name1` 文節について記述された規則に基づいてシステムが選択した表スペースに、表が作成されます。CURRENT PATH はデフォルト値であると想定します。

```
CREATE DISTINCT TYPE IMAGE AS BLOB (10M)
```

```
CREATE DISTINCT TYPE AUDIO AS BLOB (1G)
```

```
CREATE TABLE PERSON
  (SSN     INTEGER NOT NULL,
   NAME    CHAR (30),
   VOICE   AUDIO,
   PHOTO   IMAGE)
```

例 6: HUMRES 表スペースに表 EMPLOYEE を作成します。表には、以下のような制約を定義します。

- 部門番号 (DEPT) の値は、10 ~ 100 の範囲でなければならない。
- 従業員のジョブ (JOB) は、'Sales'、'Mgr'、または 'Clerk' のいずれかでなければならない。
- 1986 年以前からの従業員の給与 (SALARY) はすべて \$40,500 以上でなければならない。

注: チェック制約に含まれる列が NULL 値可能である場合、それらも NULL になる可能性があります。

```
CREATE TABLE EMPLOYEE
  (ID      SMALLINT NOT NULL,
   NAME    VARCHAR(9),
   DEPT    SMALLINT CHECK (DEPT BETWEEN 10 AND 100),
   JOB     CHAR(5) CHECK (JOB IN ('Sales','Mgr','Clerk')),
   HIREDATE DATE,
   SALARY  DECIMAL(7,2),
   COMM    DECIMAL(7,2),
   PRIMARY KEY (ID),
   CONSTRAINT YEARSAL CHECK (YEAR(HIREDATE) > 1986
    OR SALARY > 40500)
  )
IN HUMRES
```

例 7: PAYROLL 表スペースに全体が含まれる表を作成します。


```
CREATE TABLE EMPLOYEE .....
  IN PAYROLL
```

例 8: データ部分が ACCOUNTING にあり、索引部分が ACCOUNT_IDX にある表を作成します。

```
CREATE TABLE SALARY.....
  IN ACCOUNTING INDEX IN ACCOUNT_IDX
```

例 9: 表を作成して、SQL の変更内容をデフォルトのフォーマットでログ記録します。

```
CREATE TABLE SALARY1 .....
```

または

```
CREATE TABLE SALARY1 .....
```

例 10: 表を作成して、SQL の変更内容を拡張フォーマットでログ記録します。

```
CREATE TABLE SALARY2 .....
```

例 11: SCHED 表スペースに表 EMP_ACT を作成します。EMPNO、PROJNO、ACTNO、EMPTIME、EMSTDATE、および EMENDATE は列の名前です。表には、以下のような制約を定義します。

- すべての行の一連の列、EMPNO、PROJNO、および ACTNO の値は、固有でなければなりません。
- PROJNO の値は、PROJECT 表の PROJNO 列の既存の値と一致していなければならない。プロジェクトが削除される場合、そのプロジェクトを参照する EMP_ACT の行もすべて削除される。

```
CREATE TABLE EMP_ACT
(EMPNO      CHAR(6) NOT NULL,
 PROJNO     CHAR(6) NOT NULL,
 ACTNO      SMALLINT NOT NULL,
 EMPTIME    DECIMAL(5,2),
 EMSTDATE   DATE,
 EMENDATE   DATE,
 CONSTRAINT EMP_ACT_UNIQ UNIQUE (EMPNO,PROJNO,ACTNO),
 CONSTRAINT FK_ACT_PROJ FOREIGN KEY (PROJNO)
                                REFERENCES PROJECT (PROJNO) ON DELETE CASCADE
)
IN SCHED
```

ユニーク制約を課すために、EMP_ACT_UNIQ という名前のユニーク索引が同じスキーマ内に自動的に作成されます。

例 12: アイス・ホッケーの栄誉の殿堂に入る、有名なゴールについての情報を保持する表を作成します。この表には、ゴールをきめたホッケー選手の名前、ゴールをきめら

CREATE TABLE

れたゴールキーパーの名前、日付と場所、ゴールについての説明文などの情報がリストされます。さらに、可能ならば、その試合についての新聞記事やゴールのスチール写真と動画の保管先を示します。新聞記事はリンク接続されるので、削除したり、名前を変更したりできませんが、この間、既存の表示アプリケーションや更新アプリケーションは操作を続ける必要があります。スチール写真やムービーはリンクされてからアクセスできるようになりますが、この操作はすべて DB2 によって制御されます。リンクが解除されると、スチール写真はリカバリーされて元の所有者に戻されます。一方、ムービー写真はリカバリーされず、リンクが解除された時点で削除されます。説明列と 3 つの DATALINK 列は NULL 値可能です。

```
CREATE TABLE HOCKEY_GOALS
( BY_PLAYER      VARCHAR(30)  NOT NULL,
  BY_TEAM        VARCHAR(30)  NOT NULL,
  AGAINST_PLAYER VARCHAR(30)  NOT NULL,
  AGAINST_TEAM   VARCHAR(30)  NOT NULL,
  DATE_OF_GOAL   DATE          NOT NULL,
  DESCRIPTION     CLOB(5000),
  ARTICLES        DATALINK    LINKTYPE URL FILE LINK CONTROL MODE DB2OPTIONS,
  SNAPSHOT        DATALINK    LINKTYPE URL FILE LINK CONTROL
                                INTEGRITY ALL
                                READ PERMISSION DB WRITE PERMISSION BLOCKED
                                RECOVERY YES ON UNLINK RESTORE,
  MOVIE           DATALINK    LINKTYPE URL FILE LINK CONTROL
                                INTEGRITY ALL
                                READ PERMISSION DB WRITE PERMISSION BLOCKED
                                RECOVERY NO ON UNLINK DELETE )
```

例 13: EMPLOYEE 表に例外表が必要であるとします。これは、以下のステートメントを使用して作成できます。

```
CREATE TABLE EXCEPTION_EMPLOYEE AS
(SELECT EMPLOYEE.*,
  CURRENT_TIMESTAMP AS TIMESTAMP,
  CAST (' ' AS CLOB(32K)) AS MSG
FROM EMPLOYEE
) DEFINITION ONLY
```

例 14: 以下属性を持つ以下のような表スペースがあるとします。

TBSPACE	PAGESIZE	USER	USERAUTH
DEPT4K	4096	BOBBY	Y
PUBLIC4K	4096	PUBLIC	Y
DEPT8K	8192	BOBBY	Y
DEPT8K	8192	RICK	Y
PUBLIC8K	8192	PUBLIC	Y

- RICK が以下のような表を作成した場合、バイト・カウントは 4005 未満なので、その表は表スペース PUBLIC4K に入れられます。しかし BOBBY が同じ表を作成した場合、以下のような明示的な権限付与があつて BOBBY は USE 特権を保有しているので、その表は表スペース DEPT4K に入れられます。

```
CREATE TABLE DOCUMENTS
(SUMMARY VARCHAR(1000),
REPORT VARCHAR(2000))
```

- BOBBY が以下のような表を作成した場合、バイト・カウントは 4005 を超えるので、その表は表スペース DEPT8K に入れられます。また、明示的な権限付与によって BOBBY は USE 特権を保有します。しかし DUNCAN が同じ表を作成すると、それは表スペース PUBLIC8K に入れられます。DUNCAN には該当する特権がないからです。

```
CREATE TABLE CURRICULUM
(SUMMARY VARCHAR(1000),
REPORT VARCHAR(2000),
EXERCISES VARCHAR(1500))
```

例 15: 構造タイプ EMP を使って定義された LEAD 列をもつ表を作成します。LEAD 列に 300 バイトの INLINE LENGTH を指定します。これは、300 バイト以内に収まらない LEAD のインスタンスをすべて、その表以外に保管すること (LOB 値の処理方法と同様に、基本表とは別個に) を指示します。

```
CREATE TABLE PROJECTS (PID INTEGER,
LEAD EMP INLINE LENGTH 300,
STARTDATE DATE,
...)
```

例 16: DEPTNO、DEPTNAME、MGRNO、ADMRDEPT、および LOCATION という名前の 5 つの列を持つ表 DEPT を作成します。DB2 によって常に値が生成されるよう、列 DEPT を IDENTITY 列として定義することにします。DEPT 列の値は、500 から始まり、1 ずつ増分する必要があります。

```
CREATE TABLE DEPT
(DEPTNO SMALLINT NOT NULL
GENERATED ALWAYS AS IDENTITY
(START WITH 500, INCREMENT BY 1),
DEPTNAME VARCHAR (36) NOT NULL,
MGRNO CHAR(6),
ADMRDEPT SMALLINT NOT NULL,
LOCATION CHAR(30))
```

例 17: YEAR 列で区画に分割され、REGION および YEAR 列にディメンションを持つ SALES 表を作成します。データは、YEAR 列のハッシュ値に従って、パーティション内に分散しています。各パーティションで、それらのパーティション上の REGION および YEAR 列の値の固有な組み合わせを基にして、データはエクステントに編成されるされます。

```
CREATE TABLE SALES (
CUSTOMER VARCHAR(80),
REGION CHAR(5),
YEAR INTEGER )
PARTITIONING KEY ( YEAR )
ORGANIZE BY DIMENSIONS ( REGION, YEAR )
```

CREATE TABLE

例 18: PURCHASEDATE 列から生成された、PURCHASEYEARMONTH 列を持つ SALES 表を作成します。式を使用して、元の PURCHASEDATE 列に対して単調な列を作成し、それによって、ディメンションとして使用するのに適切です。表は REGION 列で区画に分割されており、各パーティションで PURCHASEYEARMONTH 列に従って、エクステントに編成されています。つまり、違う領域が違うパーティションにあり、違う購入月が、これらのパーティション内で違うセル (またはエクステントのセット) に属します。

```
CREATE TABLE SALES (  
    CUSTOMER VARCHAR(80),  
    REGION CHAR(5),  
    PURCHASEDATE DATE,  
    PURCHASEYEARMONTH INTEGER GENERATED ALWAYS AS  
        ( INTEGER(PURCHASEDATE)/100 )  
    PARTITIONING KEY ( REGION )  
    ORGANIZE BY DIMENSIONS ( PURCHASEYEARMONTH )
```

例 19 CUSTOMERNUM 列から生成された、CUSTOMERNUMDIM 列を持つ CUSTOMER 表を作成します。式を使用して、元の CUSTOMERNUM 列に対して単調な列を作成し、それによって、ディメンションとして使用するのに適切です。表は CUSTOMERNUMDIM 列に従ってセルに編成される、表内のそれぞれのセルには、50 人の顧客が入っています。固有索引が CUSTOMERNUM に作成された場合、カスタマー番号は、表内のエクステントの特定のセットに、50 の値のセットがあるように、クラスタ化されます。

```
CREATE TABLE CUSTOMER (  
    CUSTOMERNUM INTEGER,  
    CUSTOMERNAME VARCHAR(80),  
    ADDRESS VARCHAR(200),  
    CITY VARCHAR(50),  
    COUNTRY VARCHAR(50),  
    CODE VARCHAR(15),  
    CUSTOMERNUMDIM INTEGER GENERATED ALWAYS AS ( CUSTOMERNUM/50 )  
    ORGANIZE BY DIMENSIONS ( CUSTOMERNUMDIM )
```

例 20: リモート表 EMPLOYEE を透過性 DDL を使用して連合システムに作成します。

```
CREATE TABLE EMPLOYEE (  
    EMP_NO CHAR(6) NOT NULL,  
    FIRST_NAME VARCHAR(12) NOT NULL,  
    MID_INT CHAR(1) NOT NULL,  
    LAST_NAME VARCHAR(15) NOT NULL,  
    HIRE_DATE DATE,  
    JOB CHAR(8),  
    SALARY DECIMAL(9,2),  
    PRIMARY KEY (EMP_NO) )  
    OPTIONS (REMOTE_SERVER 'ORASERVER',  
    REMOTE_SCHEMA 'J15USER1', REMOTE_TABNAME 'EMPLOY' )
```

関連概念:

- 管理ガイド: プランニング の『多次元クラスタリング』

関連タスク:

- 450 ページの『透過 DDL を使用したリモート表の作成および変更』

関連資料:

- SQL リファレンス 第 1 巻 の『副選択』
- 1402 ページの『ALTER TABLE』
- 1562 ページの『CREATE TABLESPACE』
- 1610 ページの『DECLARE GLOBAL TEMPORARY TABLE』
- 1364 ページの『割り当てと比較』
- SQL リファレンス 第 1 巻 の『パーティションの互換性データ・タイプ』

関連サンプル:

- 『dtudt.c -- How to create, use, and drop user-defined distinct types. (CLI)』
- 『tbconstr.c -- How to work with constraints associated with tables (CLI)』
- 『tbcreate.c -- How to create, alter and drop tables (CLI)』
- 『dtudt.sqc -- How to create, use, and drop user-defined distinct types (C)』
- 『tbconstr.sqc -- How to create, use, and drop constraints (C)』
- 『tbcreate.sqc -- How to create and drop tables (C)』
- 『tbident.sqc -- How to use identity columns (C)』
- 『tbtrig.sqc -- How to use a trigger on a table (C)』
- 『dtudt.sqC -- How to create, use, and drop user-defined distinct types (C++)』
- 『tbconstr.sqC -- How to create, use, and drop constraints (C++)』
- 『tbcreate.sqC -- How to create and drop tables (C++)』
- 『tbtrig.sqC -- How to use a trigger on a table (C++)』
- 『DtUdt.java -- How to create, use and drop user defined distinct types (JDBC)』
- 『TbConstr.java -- How to create, use and drop constraints (JDBC)』
- 『TbCreate.java -- How to create and drop tables (JDBC)』
- 『TbGenCol.java -- How to use generated columns (JDBC)』
- 『TbIdent.java -- How to use Identity Columns (JDBC)』
- 『TbTrig.java -- How to use triggers (JDBC)』
- 『DtUdt.sqlj -- How to create, use and drop user defined distinct types (SQLj)』
- 『TbConstr.sqlj -- How to create, use and drop constraints (SQLj)』
- 『TbCreate.sqlj -- How to create and drop tables (SQLj)』
- 『TbIdent.sqlj -- How to use Identity Columns (SQLj)』
- 『TbTrig.sqlj -- How to use triggers (SQLj)』

CREATE TABLE

- 『impexp.sqb -- Export and import tables with table data (MF COBOL)』

CREATE METHOD

このステートメントは、すでにユーザー定義の構造タイプの定義の一部となっているメソッド指定に、メソッド本体を関連付けるために使用されます。

呼び出し:

このステートメントはアプリケーション・プログラムに組み込むことができ、また動的 SQL ステートメントを使用して出すことができます。 DYNAMICRULES の実行動作がパッケージに効力を持つ場合にのみ、動的に準備できる実行可能ステートメントです (SQLSTATE 42509)。

許可:

ステートメントの許可 ID によって保持されている特権には、少なくとも以下のいずれかが含まれていなければなりません。

- SYSADM または DBADM 権限
- CREATE METHOD ステートメントで参照される構造タイプのスキーマに対する CREATEIN 特権
- CREATE METHOD ステートメントで参照される構造タイプの DEFINER

外部メソッド本体をそのメソッド指定に関連付けるためには、ステートメントの許可 ID に以下の少なくとも 1 つが含まれている必要もあります。

- SYSADM または DBADM 権限
- データベースに対する CREATE_EXTERNAL_ROUTINE 権限

SQL メソッドを作成する場合、全選択で識別される表、ビュー、またはニックネームのそれぞれに対して、ステートメントの許可 ID に以下の特権が含まれている必要もあります。

- その表、ビュー、またはニックネームに対する CONTROL 特権、または
- その表、ビュー、またはニックネームに対する SELECT 特権

および以下の少なくとも 1 つ

- データベースに対する IMPLICIT_SCHEMA 権限 (ビューの暗黙または明示のスキーマ名が存在しない場合)
- スキーマに対する CREATEIN 特権 (ビューのスキーマ名が既存のスキーマを指している場合)。

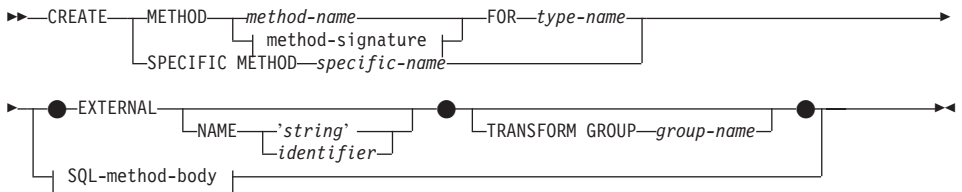
SQL メソッドの定義者が SYSADM 権限を持つために、メソッドの作成しかできない場合、メソッド作成のため、その定義者には暗黙的に DBADM 権限が付与されます。

PUBLIC 以外のグループ特権は、CREATE METHOD ステートメントで指定された表やビューに対しては考慮されません。

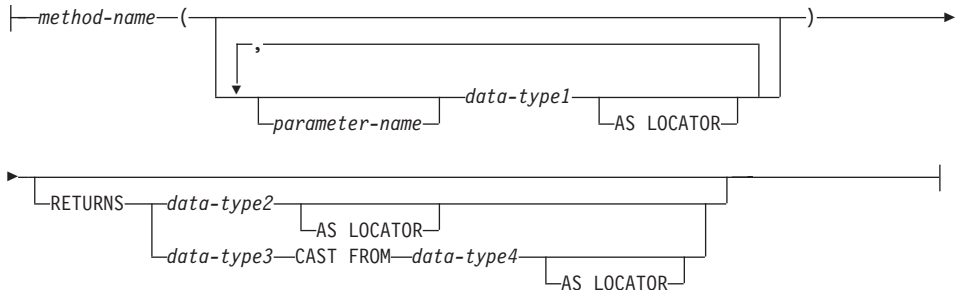
このニックネームで示されている表またはビューのデータ・ソースの許可要件は、メソッドが呼び出される時に適用されます。接続の許可 ID は、別のリモート許可 ID へマップできます。

許可 ID の権限が不十分で、操作を実行できない場合には、エラーになります (SQLSTATE 42502)。

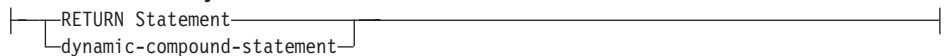
構文:



method-signature:



SQL-method-body:



説明:

METHOD

ユーザー定義の構造タイプに関連付けられる既存のメソッド指定を識別します。メソッド指定は、以下のいずれかの方法で識別できます。

method-name

メソッド本体に対して定義するメソッド指定の名前を指定します。暗黙的スキーマは、サブジェクト・タイプ (*type-name*) のスキーマです。この *method-name* のある *type-name* には、1 つしかメソッドを指定できません (SQLSTATE 42725)。

CREATE METHOD

method-signature

定義するメソッドを一意的に識別できるメソッド・シグニチャーを指定します。このメソッド・シグニチャーは、`CREATE TYPE` または `ALTER TYPE` ステートメントで提供されたメソッド指定と一致しなければなりません (SQLSTATE 42883)。

method-name

メソッド本体に対して定義するメソッド指定の名前を指定します。暗黙的スキーマは、サブジェクト・タイプ (*type-name*) のスキーマです。

parameter-name

パラメーター名を指定します。パラメーター名がメソッド・シグニチャーにより提供される場合、これらは適合するメソッド指定の対応する部分と全く同じでなければなりません。このステートメントでは、文書化だけのためにパラメーター名がサポートされています。

data-type1

各パラメーターのデータ・タイプを指定します。

AS LOCATOR

LOB タイプまたは LOB タイプに基づく特殊タイプの場合、`AS LOCATOR` 文節を追加することができます。

RETURNS

この分節は、メソッドの出力を指定します。`RETURNS` 文節がメソッド・シグニチャーにより提供される場合、これは `CREATE TYPE` の対応するメソッド指定の対応する部分と全く同じでなければなりません。このステートメントでは、文書化だけのために `RETURN` 文節がサポートされています。

data-type2

出力のデータ・タイプを指定します。

AS LOCATOR

LOB タイプまたは LOB タイプに基づく特殊タイプの場合、`AS LOCATOR` 文節を追加することができます。これは、実際の値の代わりに `LOB` ロケーターが、メソッドにより戻されることを指定します。

data-type3 CAST FROM data-type4

この形式の `RETURNS` 文節は、関数コードから戻されたデータ・タイプとは異なるデータ・タイプを、呼び出しステートメントに戻すのに使用されます。

AS LOCATOR

LOB タイプまたは LOB タイプに基づく特殊タイプの場合、`AS LOCATOR` 文節を使用して、`LOB` ロケーターが実際の値の代わりにメソッドから戻されるように指定できます。

FOR *type-name*

指定されたメソッドを関連付けるタイプを指定します。この名前は、カタログにすでに記述されているタイプを示すものでなければなりません (SQLSTATE 42704)。動的 SQL ステートメントでは、CURRENT SCHEMA 特殊レジスターが、修飾子のないオブジェクト名の修飾子として使用されます。静的 SQL ステートメントでは、QUALIFIER プリコンパイル/バインド・オプションにより、修飾子のないオブジェクト名の修飾子が暗黙指定されます。

SPECIFIC METHOD *specific-name*

CREATE TYPE 時に指定されたか、デフォルト値として与えられた値を使用して、特定のメソッドを識別します。specific-name は、指定したスキーマまたは暗黙のスキーマのメソッド指定を識別する必要があります。そうでない場合、エラーになります (SQLSTATE 42704)。

EXTERNAL

この文節は、この CREATE METHOD ステートメントを使用して登録するメソッドが、外部プログラミング言語で作成されたコードに基づいており、文書化されたリンケージの規則とインターフェースにしたがっていることを示します。CREATE TYPE で適合するメソッド指定は、SQL 以外の LANGUAGE を指定する必要があります。このメソッドが呼び出されると、メソッドのサブジェクトが、暗黙の最初のパラメーターとしてインプリメンテーションに渡されます。

NAME 文節の指定がない場合、"NAME *method-name*" が想定されます。

NAME

この文節は、定義するメソッドをインプリメントするユーザー作成コードの名前を指定します。

'string'

'string' オプションは、最大 254 文字のストリング定数です。ストリングに使用される形式は、指定した LANGUAGE によって異なります。特定の言語規則に関する詳細は、『CREATE FUNCTION (外部スカラー)』を参照してください。

identifier

指定する identifier は SQL ID です。SQL ID は、ストリングのライブラリー ID として使用されます。区切られた ID でない場合、ID は大文字に変換されます。ID がスキーマ名で修飾されている場合、スキーマ名の部分は無視されます。この形式の NAME は、LANGUAGE C でのみ使用可能です (CREATE TYPE のメソッド指定で定義)。

TRANSFORM GROUP *group-name*

メソッドを呼び出す際のユーザー定義の構造タイプのトランスフォーメーションに使用するトランスフォーム・グループを指定します。メソッド定義には、ユーザー定義の構造タイプが含まれているため、トランスフォームが必要です。

CREATE METHOD

ここで、トランスフォーム・グループ名を指定することを強くお勧めします。この文節が指定されない場合、使用されるデフォルトのグループ名は DB2_FUNCTION です。参照された構造タイプに、指定した (またはデフォルトの) グループ名が定義されていない場合には、エラーになります (SQLSTATE 42741)。同様に、指定したグループ名または構造タイプに、必須の FROM SQL または TO SQL トランスフォーム関数が定義されていない場合には、エラーになります (SQLSTATE 42744)。

SQL-method-body

CREATE TYPE のメソッド仕様が LANGUAGE SQL の場合、SQL-method-body は、どのようにメソッドがインプリメントされるかを定義します。

SQL-method-body は、以下のメソッド仕様のパーツにしたがってなければなりません。

- DETERMINISTIC または NOT DETERMINISTIC (SQLSTATE 428C2)
- EXTERNAL ACTION または NO EXTERNAL ACTION (SQLSTATE 428C2)
- CONTAINS SQL または READS SQL DATA (SQLSTATE 42985)

パラメーター名を SQL-method-body で参照することができます。メソッドのサブジェクトは、暗黙的な最初のパラメーター SELF としてメソッド・インプリメンテーションに渡されます。

詳細については、『複合 SQL (動的)』 および 『RETURN ステートメント』を参照してください。

規則:

- CREATE TYPE または ALTER TYPE ステートメントを使用して、前もってメソッド指定を定義していなければ、CREATE METHOD は使用できません (SQLSTATE 42723)。
- 作成されるメソッドがオーバーライド・メソッドの場合には、以下のメソッドに従属するパッケージは無効になります。
 - オリジナル・メソッド
 - 作成されるスーパータイプのメソッドをサブジェクトとして持つ、他のオーバーライド・メソッド

注:

- メソッドが SQL を許可する場合、外部プログラムは、連合オブジェクトへのアクセスを試行してはなりません (SQLSTATE 55047)。

- **特権**

メソッドの定義者は、メソッドに対する EXECUTE 特権と、メソッドをドロップする権利を常に与えられます。

EXTERNAL メソッドが作成されると、メソッドの定義者は EXECUTE WITH GRANT OPTION 特権を常に受け取ります。

SQL メソッドが作成されると、メソッドの定義者がメソッドを定義するために必要なすべての特権に対して WITH GRANT OPTION を持っている場合、または定義者が SYSADM や DBADM 権限を持っている場合には、メソッドに対する EXECUTE WITH GRANT OPTION 特権のみが定義者に与えられます。メソッドの定義者にそれらの特権が与えられるのは、それらの特権の派生元の特権がメソッドの作成時に存在している場合に限りです。定義者は、これらの特権を直接持っているか、または PUBLIC の特権として持っていることが必要です。メソッドの定義者がメンバーであるグループを持つ特権は考慮されません。メソッドを使用する場合、接続済みのユーザーの許可 ID には、そのデータ・ソースでニックネームが参照する表またはビューに対する適切な特権がなければなりません。

- **表アクセスの制限**

メソッドが READS SQL DATA として定義されている場合は、メソッド内のステートメントは、このメソッドを呼び出したステートメントによって変更される表にアクセスすることはできません (SQLSTATE 57053)。

例:

例 1:

```
CREATE METHOD BONUS (RATE DOUBLE)
FOR EMP
RETURN SELF..SALARY * RATE
```

例 2:

```
CREATE METHOD SAMEZIP (addr address_t)
RETURNS INTEGER
FOR address_t
RETURN
(CASE
  WHEN (self..zip = addr..zip)
  THEN 1
  ELSE 0
END)
```

例 3:

```
CREATE METHOD DISTANCE (address_t)
FOR address_t
EXTERNAL NAME 'addresslib!distance'
TRANSFORM GROUP func_group
```

関連資料:

- *SQL* リファレンス 第 2 巻 の『RETURN ステートメント』
- *SQL* リファレンス 第 2 巻 の『コンパウンド SQL (動的) ステートメント』
- 1618 ページの『CREATE FUNCTION (外部スカラー)』

CREATE FUNCTION (SQL スカラー、表、または行)

CREATE FUNCTION (SQL スカラー、表、または行)

このステートメントは、ユーザー定義の SQL スカラー、表、または行関数を定義するのに使用されます。スカラー関数は、呼び出されるたびに 1 つの値を戻し、SQL 式が有効な個所であればどこでも有効です。表関数は、FROM 文節で使用でき、表を戻します。行関数は、トランスフォーム関数として使用でき、行を戻します。

呼び出し:

このステートメントはアプリケーション・プログラムに組み込むことができ、また動的 SQL ステートメントを使用して出すことができます。DYNAMICRULES の実行動作がパッケージに効力を持つ場合にのみ、動的に準備できる実行可能ステートメントです (SQLSTATE 42509)。

許可:

ステートメントの許可 ID によって保持されている特権には、少なくとも以下のいずれかが含まれていなければなりません。

- SYSADM または DBADM 権限
- 全選択に指定された表、ビュー、またはニックネームのそれぞれに対して、
 - その表、ビュー、またはニックネームに対する CONTROL 特権、または
 - その表、ビュー、またはニックネームに対する SELECT 特権

および以下の少なくとも 1 つ

- データベースに対する IMPLICIT_SCHEMA 権限 (ビューの暗黙または明示のスキーマ名が存在しない場合)
- スキーマに対する CREATEIN 特権 (ビューのスキーマ名が既存のスキーマを指している場合)

PUBLIC 以外のグループ特権は、CREATE FUNCTION ステートメントで指定された表やビューに対しては考慮されません。

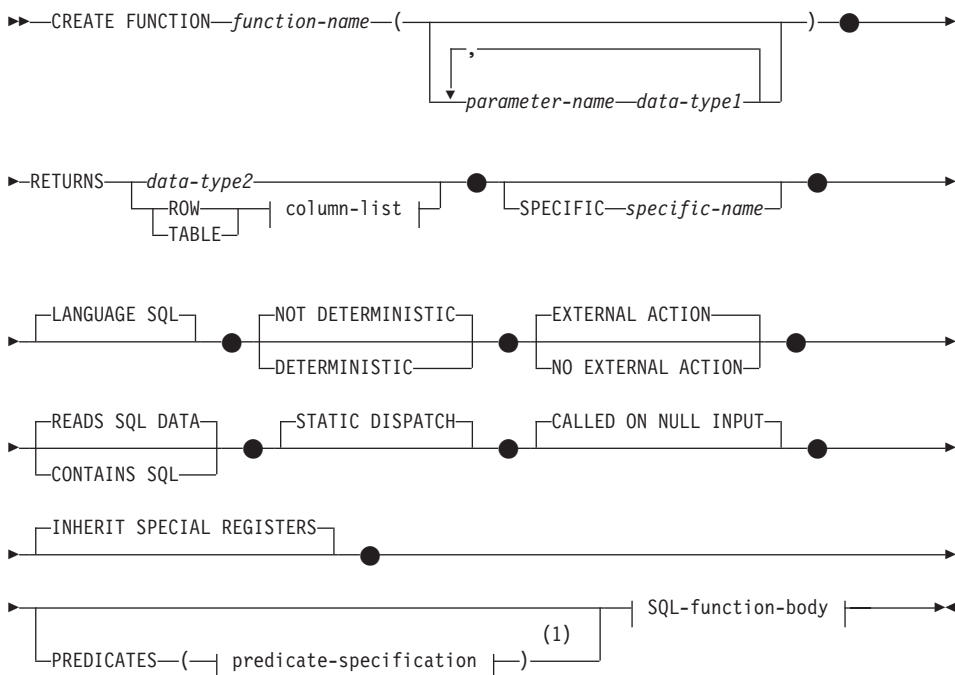
このニックネームで示されている表またはビューのデータ・ソースの許可要件は、関数が呼び出される時に適用されます。接続の許可 ID は、別のリモート許可 ID へマップできます。

関数の定義者が SYSADM 権限を持つために、関数の作成しかできない場合、関数作成のため、その定義者には暗黙的に DBADM 権限が付与されます。

許可 ID の権限が不十分で、操作を実行できない場合には、エラー (SQLSTATE 42502) になります。

構文:

CREATE FUNCTION (SQL スカラー、表、または行)



column-list:



SQL-function-body:



注:

- 1 RETURNS がスカラー結果 (`data-type2`) を指定している場合のみ有効です。

説明:

function-name

定義する関数の名前を指定します。これは、関数を指定する修飾または非修飾の名前です。 *function-name* (関数名) の非修飾形式は SQL ID です (最大長 18)。動的 SQL ステートメントでは、CURRENT SCHEMA 特殊レジスタが、修飾子のないオブジェクト名の修飾子として使用されます。静的 SQL ステートメントでは、QUALIFIER プリコンパイル/バインド・オプションにより、修飾子のないオブジェクト名の修飾子が暗黙指定されます。修飾形式は、*schema-name* の後にピリオドと SQL ID が続きます。

CREATE FUNCTION (SQL スカラー、表、または行)

暗黙または明示の修飾子を含む名前、およびパラメーターの数と各パラメーターのデータ・タイプ (データ・タイプの長さ、精度、または位取りの各属性には関係なく) は、カタログに記述されている関数を指定するものであってはなりません (SQLSTATE 42723)。非修飾名とパラメーターの数およびデータ・タイプとの組み合わせは、そのスキーマ内では当然固有ですが、複数のスキーマ間で固有である必要はありません。

2 つの部分からなる名前を指定する場合、“SYS” で始まる *schema-name* (スキーマ名) は使用できません (SQLSTATE 42939)。

述部のキーワードとして使用されるいくつかの名前は、システム使用に予約されており、*function-name* として使用することはできません (SQLSTATE 42939)。それらの名前は、SOME、ANY、ALL、NOT、AND、OR、BETWEEN、NULL、LIKE、EXISTS、IN、UNIQUE、OVERLAPS、SIMILAR、MATCH、および比較演算子です。

関数のシグニチャーに何らかの差異があれば、同じ名前を複数の関数に使用することができます。禁止されてはいませんが、外部ユーザー定義関数の名前として、組み込み関数と同じ名前を指定すべきではありません。

parameter-name

この関数の他のすべてのパラメーター名と異なる名前。

data-type1

パラメーターのデータ・タイプを指定します。

- CREATE TABLE ステートメントの *data-type1* の定義で指定可能な SQL データ・タイプ仕様と省略形を指定することができます。
- REF を指定することはできますが、このとき REF の有効範囲は指定できません。システムがパラメーターまたは結果の有効範囲を妨げることはしません。関数本体では、最初に参照タイプをキャストして有効範囲を指定してからでなければ、参照解除操作は使用できません。同様に、SQL 関数により戻される参照も、最初にこれに有効範囲を指定してからでなければ参照解除操作は使用できません。
- LONG VARCHAR および LONG VARGRAPHIC データ・タイプは使用できません (SQLSTATE 42815)。

RETURNS

これは必須の文節であり、関数の出力のタイプを指定します。

data-type2

出力のデータ・タイプを指定します。

このステートメントでは、前述の関数パラメーター *data-type1* で説明した SQL 関数のパラメーターと同じ考慮事項が適用されます。

ROW *column-list*

関数の出力が単一の行であることを指定します。関数が複数の行を出力する場合

CREATE FUNCTION (SQL スカラー、表、または行)

合、エラーになります (SQLSTATE 21505)。 *column-list* には、少なくとも 2 つの列を組み込まなければなりません (SQLSTATE 428F0)。

行関数は、構造タイプの変換関数としてのみ使用できます (1 つの構造タイプをパラメーターとして使用し、基本タイプのみ戻します)。

TABLE *column-list*

関数の出力が表であることを指定します。

column-list

ROW または TABLE 関数で戻される列名およびデータ・タイプのリスト。

column-name

この列の名前を指定します。名前を修飾することはできず、行の複数の列に対して同じ名前を使用することはできません。

data-type3

列のデータ・タイプを指定します。SQL 関数のパラメーターによりサポートされていれば、どのデータ・タイプでも構いません。

SPECIFIC *specific-name*

定義する関数のインスタンスに対する固有名を指定します。この特定名は、この関数をソース関数として使用する場合、この関数をドロップする場合、またはこの関数にコメントを付ける場合に使用することができます。これは、関数の呼び出しには使用できません。 *specific-name* (関数名) の非修飾形式は SQL ID です (最大長 18)。修飾形式は、*schema-name* の後にピリオドと SQL ID が続きます。暗黙または明示の修飾子も含め、その名前が、アプリケーション・サーバーに存在する別の関数インスタンスを識別するものであってはなりません。さもないと、エラーになります (SQLSTATE 42710)。

specific-name は、既存の *function-name* (関数名) と同じでも構いません。

修飾子を指定しない場合、*function-name* に使用された修飾子が使用されます。修飾子を指定する場合は、*function-name* の明示修飾子または暗黙修飾子と同じでなければなりません。そうでない場合は、エラーになります (SQLSTATE 42882)。

specific-name の指定がない場合、固有の名前がデータベース・マネージャーによって生成されます。生成される固有名は、SQL の後に文字のタイム・スタンプが続く名前です (SQLyymmddhhmmssxxx)。

LANGUAGE SQL

関数が SQL を使用して書かれていることを指定します。

DETERMINISTIC または NOT DETERMINISTIC

この文節は任意指定で、特定の引き数の値に対して関数が常に同じ結果を戻すか (DETERMINISTIC)、それとも状態値に依存して関数の結果が影響を受けるか (NOT DETERMINISTIC) を指定します。つまり DETERMINISTIC 関数は、同一の入力で

CREATE FUNCTION (SQL スカラー、表、または行)

正しい呼び出しが行われたとき、常に同じ表を返します。NOT DETERMINISTIC を指定すると、同じ入力によって常に同じ結果が生じる利点に基づく最適化ができなくなります。

関数の本体が特殊レジスターにアクセスしたり、他の非 deterministic 関数を呼び出す場合には、NOT DETERMINISTIC を明示的または暗黙的に指定しなければなりません (SQLSTATE 428C2)。

NO EXTERNAL ACTION または EXTERNAL ACTION

このオプションの文節は、データベース・マネージャーによって管理されていないオブジェクトの状態を変更するアクションを関数が行うかどうかを指定します。

NO EXTERNAL ACTION を指定すると、システムは関数が外部に影響を与えないことを前提とした最適化を使用できます。

関数の本体が外部アクションのある別の関数を呼び出す場合、EXTERNAL ACTION を明示的または暗黙的に指定しなければなりません (SQLSTATE 428C2)。

READS SQL DATA または CONTAINS SQL

どのタイプの SQL ステートメントを実行できるかを指示します。サポートされている SQL ステートメントは RETURN ステートメントであるので、式が副照会であるかどうかで区別を行います。

READS SQL DATA

SQL データを変更しない SQL ステートメントを、関数により実行できるように指示します (SQLSTATE 42985)。

CONTAINS SQL

SQL データの読み取りも変更もしない SQL ステートメントを、関数により実行できるように指示します (SQLSTATE 42985)。

STATIC DISPATCH

このオプション文節は、問題解決時に DB2 が関数のパラメーターの静的タイプ (宣言済みタイプ) に基づいて関数を選択するよう指示します。

CALLED ON NULL INPUT

なんらかの引き数が NULL 値であるかどうかにかかわらず、関数が呼び出されることを指定します。これは、NULL 値を戻す場合も、NULL 値以外の値を戻す場合もあります。NULL の引き数値の有無のテストはユーザー定義関数が行う必要があります。

CALLED ON NULL INPUT の代わりに NULL CALL という句を使用できます。

INHERIT SPECIAL REGISTERS

このオプション文節は、関数の更新可能な特殊レジスターが、呼び出しステートメントの環境からの初期値を継承するよう指示します。カーソルの select ステートメントに呼び出された関数の場合、初期値はカーソルがオープンした際の環境から継

CREATE FUNCTION (SQL スカラー、表、または行)

承します。ネストされたオブジェクト (たとえば、トリガーまたはビュー) に呼び出されるルーチンの場合、初期値は実行時環境 (オブジェクト定義ではない) から継承します。

特殊レジスターに対する変更が、関数の呼び出し元に戻されることはありません。

一部の特殊レジスター (日時特殊レジスターなど) は、現在実行中のステートメントのプロパティを反映するので、呼び出し元からの継承は行われません。

PREDICATES

この関数を使用する述部の場合、この文節は索引拡張を使用できることを示し、オプションの SELECTIVITY 文節を使用して述部の検索条件を指定できます。

PREDICATES 文節が指定された場合、関数は NO EXTERNAL ACTION を指定した DETERMINISTIC として定義しなければなりません (SQLSTATE 42613)。

predicate-specification

述部指定に関する詳細は、『CREATE FUNCTION (外部スカラー)』を参照してください。

SQL-function-body

関数の本体を指定します。パラメーター名を SQL-function-body で参照することができます。あいまい参照を避けるため、パラメーター名は関数名で修飾できます。

SQL-function-body が動的複合ステートメントであれば、ここには少なくとも 1 つの RETURN ステートメントが組み込まれていなければならない、また関数が呼び出されるときに RETURN ステートメントが実行されなければなりません (SQLSTATE 42632)。関数が表または行関数であれば、RETURN ステートメントは、動的複合ステートメントの最後の関数でなければなりません。RETURN ステートメントを 1 つだけ組み込むことができます (SQLSTATE 429BD)。

注:

- **互換性**
 - 以前のバージョンの DB2 との互換性:
 - CALLED ON NULL INPUT の代わりに NULL CALL を指定できます。
- 関数本体内での関数呼び出しの解決は、CREATE FUNCTION ステートメントに対して有効な関数パスにしたがって実行され、関数が作成された後も変更されません。
- SQL 関数に、何らかの日付または時刻の特殊レジスターへの参照が複数含まれている場合、すべての参照は同じ値を戻します。そして、関数を呼び出したステートメントでのレジスター呼び出しにより戻される値と同じになります。
- SQL 関数の本体には、これ自体または他の関数やこれを呼び出すメソッドに対する再帰呼び出しを組み込むことはできません。そのような関数は、呼び出しの対象として存在できないからです。
- 関数またはメソッドを作成するすべてのステートメントで、以下の規則が課されます。

CREATE FUNCTION (SQL スカラー、表、または行)

- 関数のシグニチャーは、メソッドのシグニチャーと同じであってはなりません (関数の最初の *parameter-type* と、メソッドの *subject-type* を比較)。
- 関数とメソッドは、オーバーライド関係にあってはなりません。つまり、関数が、最初のパラメーターをサブジェクトとするメソッドである場合、これが他のメソッドをオーバーライドしたり、他のメソッドによりオーバーライドされたりすることはできません。オーバーライド・メソッドについての詳細は、『CREATE TYPE (構造化)』 ステートメントを参照してください。
- 関数にはオーバーライドが適用されないため、2 つの関数がメソッドである場合に、一方が他方をオーバーライドする状態で存在することは可能です。

上記の規則で *parameter-types* を比較する目的で、以下ようになります。

- パラメーター名、長さ、AS LOCATOR、および FOR BIT DATA は無視されます。
 - サブタイプとそのスーパータイプは異なるものと見なされます。
- **表アクセスの制限**
関数が READS SQL DATA に定義されている場合には、関数のいかなるステートメントも、関数を呼び出したステートメントによって変更されている表にはアクセスできません (SQLSTATE 57053)。たとえば、ユーザー定義関数 `BONUS()` が READS SQL DATA に定義されているとします。ステートメント `UPDATE EMPLOYEE SET SALARY = SALARY + BONUS(EMPNO)` が呼び出される場合、`BONUS` 関数の SQL ステートメントは、`EMPLOYEE` 表からは読み取りは行えません。
 - **特権**

関数の定義者は、関数に対する EXECUTE 特権と、関数をドロップする権利を常に与えられます。関数を定義するのに必要なすべての特権に対して定義者が WITH GRANT OPTION を持っている場合、または定義者が SYSADM か DBADM 権限を持っている場合には、関数の定義者には、関数に対する WITH GRANT OPTION も与えられます。

関数の定義者にそれらの特権が与えられるのは、それらの特権の派生元の特権が関数の作成時に存在している場合に限りです。定義者は、これらの特権を直接持っているか、または PUBLIC の特権として持っていることが必要です。関数の定義者がメンバーであるグループを持つ特権は考慮されません。関数を使用する場合、接続済みのユーザーの許可 ID には、そのデータ・ソースでニックネームが参照する表またはビューに対する適切な特権がなければなりません。

例:

例 1: 既存のサインおよびコサイン関数を使用して、値のタンジェントを戻すスカラー関数を定義します。

```
CREATE FUNCTION TAN (X DOUBLE)
  RETURNS DOUBLE
  LANGUAGE SQL
```

CREATE FUNCTION (SQL スカラー、表、または行)

```
CONTAINS SQL
NO EXTERNAL ACTION
DETERMINISTIC
RETURN SIN(X)/COS(X)
```

例 2: 構造タイプ PERSON のトランスフォーム関数を定義します。

```
CREATE FUNCTION FROMPERSON (P PERSON)
RETURNS ROW (NAME VARCHAR(10), FIRSTNAME VARCHAR(10))
LANGUAGE SQL
CONTAINS SQL
NO EXTERNAL ACTION
DETERMINISTIC
RETURN VALUES (P..NAME, P..FIRSTNAME)
```

例 3: 指定された部門番号の従業員を戻す表関数を定義します。

```
CREATE FUNCTION DEPTEMPLOYEES (DEPTNO CHAR(3))
RETURNS TABLE (EMPNO CHAR(6),
                LASTNAME VARCHAR(15),
                FIRSTNAME VARCHAR(12))
LANGUAGE SQL
READS SQL DATA
NO EXTERNAL ACTION
DETERMINISTIC
RETURN
SELECT EMPNO, LASTNAME, FIRSTNAME
FROM EMPLOYEE
WHERE EMPLOYEE.WORKDEPT = DEPTEMPLOYEES.DEPTNO
```

例 4: スtringを反転するスカラー関数を定義します。

```
CREATE FUNCTION REVERSE(INSTR VARCHAR(4000))
RETURNS VARCHAR(4000)
DETERMINISTIC NO EXTERNAL ACTION CONTAINS SQL
BEGIN ATOMIC
DECLARE REVSTR, RESTSTR VARCHAR(4000) DEFAULT '';
DECLARE LEN INT;
IF INSTR IS NULL THEN
RETURN NULL;
END IF;
SET (RESTSTR, LEN) = (INSTR, LENGTH(INSTR));
WHILE LEN > 0 DO
SET (REVSTR, RESTSTR, LEN)
= (SUBSTR(RESTSTR, 1, 1) || REVSTR,
SUBSTR(RESTSTR, 2, LEN - 1),
LEN - 1);
END WHILE;
RETURN REVSTR;
END
```

関連資料:

- *SQL* リファレンス 第 1 巻 の『基本述部』

CREATE FUNCTION (SQL スカラー、表、または行)

- 1441 ページの『CREATE TYPE (構造化)』
- *SQL* リファレンス 第 2 巻 の『RETURN ステートメント』
- *SQL* リファレンス 第 2 巻 の『コンパウンド SQL (動的) ステートメント』
- 1618 ページの『CREATE FUNCTION (外部スカラー)』
- 1360 ページの『ルーチンで使用可能な SQL ステートメント』
- *SQL* リファレンス 第 1 巻 の『特殊レジスター』

ALTER PROCEDURE

ALTER PROCEDURE ステートメントは、プロシージャのプロパティを変更して、既存のプロシージャを変更します。

呼び出し:

このステートメントはアプリケーション・プログラムに組み込むことができ、また動的 SQL ステートメントを使用して出すことができます。 DYNAMICRULES の実行動作がパッケージに効力を持つ場合にのみ、動的に準備できる実行可能ステートメントです (SQLSTATE 42509)。

許可:

ステートメントの許可 ID によって保持されている特権には、少なくとも以下のいずれかが含まれていなければなりません。

- SYSADM または DBADM 権限
- プロシージャのスキーマに対する ALTERIN 特権
- SYSCAT.ROUTINES の DEFINER 列に記録されているそのプロシージャの定義者

プロシージャの EXTERNAL NAME を変更するには、ステートメントの許可 ID の特権に、以下の特権の少なくとも 1 つが含まれている必要があります。

- SYSADM または DBADM 権限
- データベースに対する CREATE_EXTERNAL_ROUTINE 権限

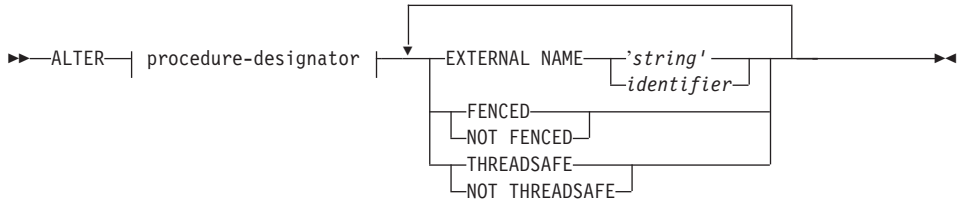
隔離されないようにプロシージャを変更するには、ステートメントの許可 ID の特権に以下の特権の少なくとも 1 つが含まれている必要があります。

- SYSADM または DBADM 権限
- データベースに対する CREATE_NOT_FENCED_ROUTINE 権限

隔離されるようにプロシージャを変更するには、さらに別の権限や特権は必要ありません。

許可 ID の権限が不十分で、操作を実行できない場合には、エラー (SQLSTATE 42502) になります。

構文:



説明:

procedure-designator

変更されるプロシージャを一意的に識別します。

EXTERNAL NAME 'string' または identifier

プロシージャをインプリメントするユーザー作成コードの名前を指定します。このオプションは、外部プロシージャを変更する際のみ指定できます (SQLSTATE 42849)。

FENCED または NOT FENCED

プロシージャをデータベース・マネージャーの操作環境のプロセスまたはアドレス・スペースで実行しても安全か (NOT FENCED)、そうでないか (FENCED) を指定します。多くのプロシージャは、 FENCED として実行するか NOT FENCED として実行するかを選択が可能です。

プロシージャが FENCED に変更されると、データベース・マネージャーは、その内部リソース (データ・バッファなど) を隔離して、そのプロシージャからアクセスされないようにします。一般に、FENCED として実行されるプロシージャは、 NOT FENCED として実行されるものと同じようには実行されません。

注意:

適切にコード化、検討、およびテストされていないプロシージャに **NOT FENCED** を使用すると、**DB2** の保全性に危険を招く場合があります。**DB2** では、発生する可能性のある一般的な不注意による障害の多くに対して、いくつかの予防措置がとられていますが、**NOT FENCED** ストアード・プロシージャが使用される場合には、完全な保全性を確保できません。

NOT THREADSAFE を宣言したプロシージャは、NOT FENCED には変更できません (SQLSTATE 42613)。

プロシージャが AS LOCATOR を定義した任意のパラメーターを有していて、NO SQL オプションも指定されている場合には、このプロシージャは FENCED には変更できません (SQLSTATE 42613)。

このオプションは LANGUAGE OLE プロシージャを変更できません (SQLSTATE 42849)。

ALTER PROCEDURE

このオプションは、外部プロシージャを変更する際にのみ指定できます (SQLSTATE 42849)。

THREADSAFE または NOT THREADSAFE

プロシージャを他のルーチンと同じプロセスで実行しても安全か (THREADSAFE)、そうでないか (NOT THREADSAFE) を指定します。

プロシージャが OLE 以外の LANGUAGE で定義される場合:

- プロシージャが THREADSAFE として定義されている場合には、データベース・マネージャーは他のルーチンと同じプロセスにプロシージャを呼び出すことができます。一般に、スレッド・セーフになるには、プロシージャはどのグローバルあるいは静的データ域をも使用してはなりません。多くのプログラミング解説書には、スレッド・セーフ・ルーチンの作成に関する説明が含まれています。FENCED および NOT FENCED プロシージャの両方を THREADSAFE にすることができます。
- プロシージャが NOT THREADSAFE に定義される場合には、データベース・マネージャーは他のルーチンと同じプロセスにプロシージャを決して呼び出しません。隔離されたプロシージャだけが、NOT THREADSAFE になり得ます (SQLSTATE 42613)。

このオプションは、外部プロシージャを変更する際にのみ指定できます (SQLSTATE 42849)。

このオプションは LANGUAGE OLE プロシージャを変更できません (SQLSTATE 42849)。

注:

- SYSIBM、SYSFUN、または SYSPROC スキーマ (SQLSTATE 42832) のプロシージャは変更できません。
- LANGUAGE SQL として宣言されたプロシージャは変更できません (SQLSTATE 42917)。

例:

プロシージャ PARTS_ON_HAND() が隔離されないように変更します。

```
ALTER PROCEDURE PARTS_ON_HAND() NOT FENCED
```

関連資料:

- *SQL* リファレンス 第 2 巻 の『CREATE PROCEDURE ステートメント』

CREATE TABLESPACE

CREATE TABLESPACE ステートメントは、データベースに新しい表スペースを作成し、その表スペースにコンテナを割り当て、その表スペース定義と属性をカタログに記録します。

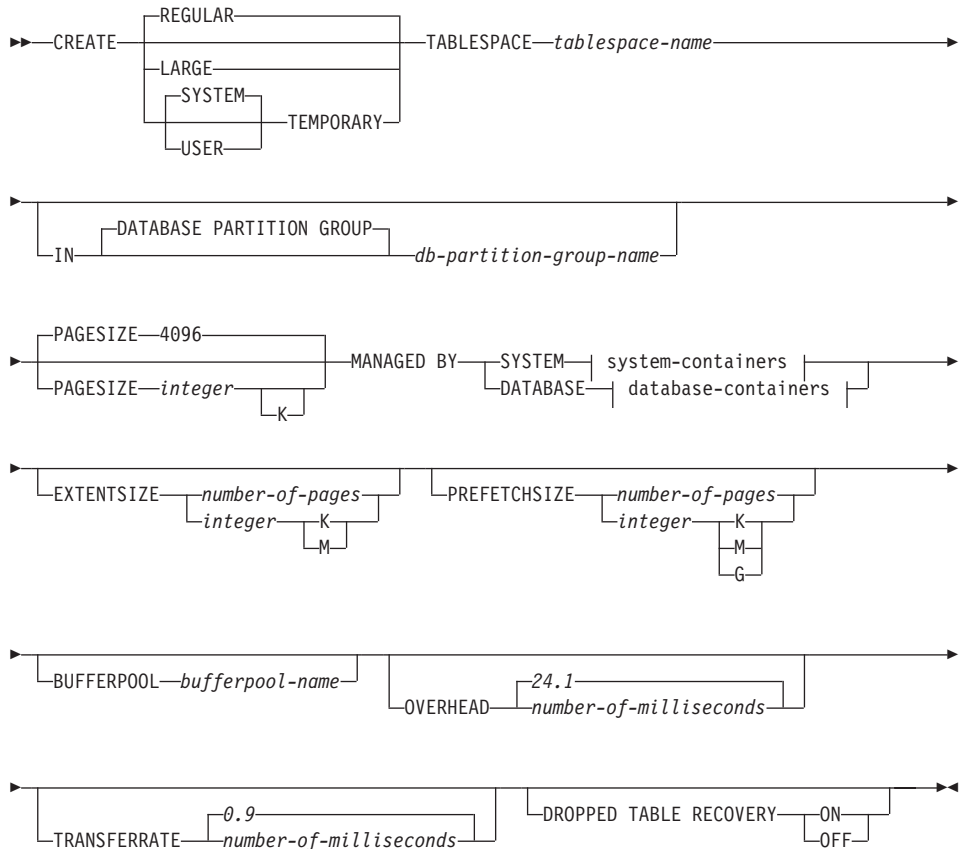
呼び出し:

このステートメントは、アプリケーション・プログラムに組み込むか、あるいは対話式に発行することができます。DYNAMICRULES の実行動作がパッケージに効力を持つ場合にのみ、動的に準備できる実行可能ステートメントです (SQLSTATE 42509)。

許可:

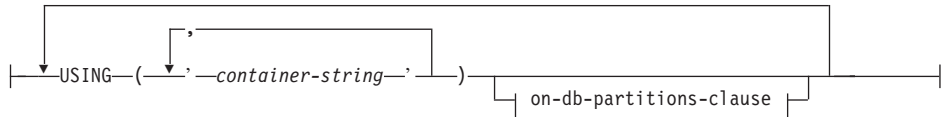
このステートメントの許可 ID には、SYSCTRL 権限または SYSADM 権限がなければなりません。

構文:

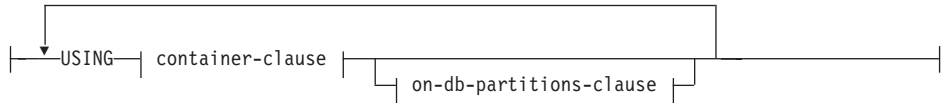


CREATE TABLESPACE

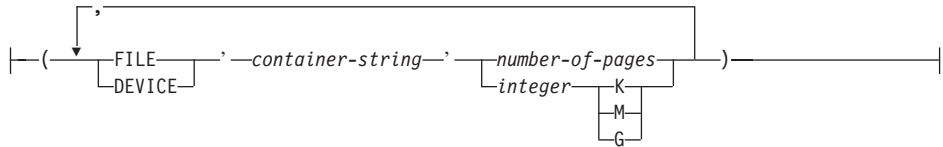
system-containers:



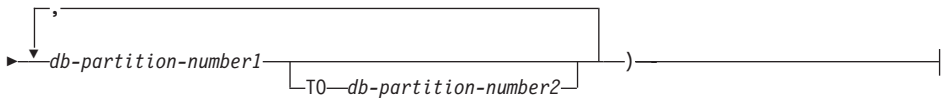
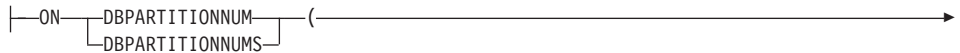
database-containers:



container-clause:



on-db-partitions-clause:



説明:

REGULAR

一時表を除くすべてのデータを保管します。

LARGE

長形式または LOB の表の列を保管します。また、構造タイプ列または索引データを保管することもできます。表スペースは DMS 表スペースでなければなりません。

SYSTEM TEMPORARY

一時表 (データベース・マネージャーがソートや結合などの操作を実行するのに使用する作業域) を保管します。キーワード SYSTEM は任意指定です。一時表はこのような表スペースにのみ保管することができるので、データベースには、常に少なくとも 1 つの SYSTEM TEMPORARY 表スペースがなければならぬ点に注意してください。一時表スペースは、データベースの作成時に自動的に作成されません。

USER TEMPORARY

宣言されたグローバル一時表を保管します。データベースの作成時にユーザー一時表スペースは存在しないことに注意してください。宣言済み一時表を定義できるよう、該当する USE 特権を使って少なくとも 1 つの USER TEMPORARY 表スペースを作成する必要があります。

tablespace-name

表スペースの名前を指定します。これは、1 つの部分からなる名前です。これは、SQL ID です (通常 ID または区切り ID)。 *tablespace-name* (表スペース名) は、すでにカタログに存在している表スペースを指定するものであってはなりません (SQLSTATE 42710)。 *tablespace-name* を文字 SYS で始めることはできません (SQLSTATE 42939)。

IN DATABASE PARTITION GROUP *db-partition-group-name*

表スペースのデータベース・パーティション・グループを指定します。該当のデータベース・パーティション・グループは存在していなければなりません。 SYSTEM TEMPORARY 表スペースの作成の際に指定できるデータベース・パーティション・グループは、IBMTEMPGROUP だけです。 DATABASE PARTITION GROUP キーワードはオプションです。

データベース・パーティション・グループを指定しないと、デフォルトのデータベース・パーティション・グループ (IBMDEFAULTGROUP) が、REGULAR、LARGE、および USER TEMPORARY 表スペースに使用されます。 SYSTEM TEMPORARY 表スペースには、デフォルト・データベース・パーティション・グループ IBMTEMPGROUP が使われます。

PAGESIZE *integer* [K]

表スペースに使用するページのサイズを定義します。接尾部 K を持たない *integer* の有効値は、4 096 または 8 192、16 384、または 32 768 です。接尾部 K を持つ *integer* の有効値は、4 または 8、16、または 32 です。ページ・サイズがこれらのいずれの値にも該当しない場合 (SQLSTATE 428DE)、あるいはページ・サイズが表スペースと関連付けられたバッファ・プールのページ・サイズと同じではない場合 (SQLSTATE 428CB) には、エラーが起きます。 デフォルト値は 4 096 バイト (4K) ページです。 *integer* と K の間には、任意の数のスペースを使用できます (スペースなしでも可)。

MANAGED BY SYSTEM

表スペースを、システム管理スペース (SMS) 表スペースとして指定します。

system-containers

SMS 表スペースに対するコンテナを指定します。

USING ('*container-string*',...)

SMS 表スペースに対して、表スペースに属し、表スペースのデータの保管先となる 1 つまたは複数のコンテナを指定します。コンテナ・ストリング (*container-string*) の長さは、240 バイトを超えてはなりません。

CREATE TABLESPACE

各 *container-string* は、絶対ディレクトリー名または相対ディレクトリー名にすることができます。ディレクトリー名が絶対ではない場合は、データベース・ディレクトリーからの相対ディレクトリーになります。ディレクトリー名のコンポーネントのいずれかが存在しない場合は、それがデータベース・マネージャーによって作成されます。表スペースをドロップすると、データベース・マネージャーによって作成されたすべてのコンポーネントが削除されます。コンテナ・ストリングで指定されたディレクトリーが存在する場合、そのディレクトリーにはファイルやサブディレクトリーがあってはなりません (SQLSTATE 428B2)。

container-string の形式は、オペレーティング・システムによって異なります。コンテナは、オペレーティング・システムの通常の方法で指定されます。たとえば、Windows のディレクトリー・パスはドライブ文字と “:” から始まり、UNIX 系システムでは “/” から始まります。

リモート・リソース (LAN でリダイレクトされたドライブや NFS でマウントされたファイル・システムなど) は、現在、Network Appliance Filers、IBM iSCSI、または IBM Network Attached Storage を使用する場合にのみサポートされます。

on-db-partitions-clause

パーティション・データベースにおいて、コンテナを作成するパーティションを指定します。この文節を指定しない場合、他のどの *on-db-partitions-clause* にも明示的に指定されていないデータベース・パーティション・グループ内のパーティションでコンテナが作成されます。データベース・パーティション・グループ IBMTEMPGROUP で定義されている SYSTEM TEMPORARY 表スペースについては、*on-db-partitions-clause* を指定しないと、データベースに追加されたすべての新しいパーティションでもコンテナが作成されます。

MANAGED BY DATABASE

表スペースを、データベース管理スペース (DMS) 表スペースとして指定します。

database-containers

DMS 表スペースに対するコンテナを指定します。

USING

container-clause を導きます。

container-clause

DMS 表スペースに対してコンテナを指定します。

(FILE|DEVICE '*container-string*' *number-of-pages*,...)

DMS 表スペースに対して、表スペースに属し、表スペースのデータの保管先となる 1 つまたは複数のコンテナを指定します。コンテナのタイプ (FILE または DEVICE) とそのサイズ (PAGESIZE ページの数) を指定します。このサイズは整数値としても指定でき、その後 K (K バイトの場合)、M (M バイトの場合)、または G (G バイトの場合) を付けま

す。このように指定した場合、ページ・サイズで分割されたバイト数のフロアは、コンテナのページ数を判別するために使用します。FILE と DEVICE のコンテナを混合して指定できます。コンテナ・ストリング (*container-string*) の長さは、254 バイトを超えてはなりません。

FILE コンテナの場合、*container-string* は、絶対ファイル名または相対ファイル名でなければなりません。絶対ファイル名以外のファイル名は、データベース・ディレクトリーからの相対パス名になります。ディレクトリー名のコンポーネントのいずれかが存在しない場合は、それがデータベース・マネージャーによって作成されます。ファイルが存在しない場合、データベース・マネージャーによってそのファイルが作成され、指定されたサイズに初期化されます。表スペースをドロップすると、データベース・マネージャーによって作成されたすべてのコンポーネントが削除されます。

注: ファイルが存在する場合は上書きされ、指定したサイズより小さい場合には拡張されます。指定したサイズよりもファイルの方が大きくても、ファイルは切り捨てられません。

DEVICE コンテナの場合、*container-string* は装置名でなければなりません。その装置はすでに存在していなければなりません。

すべてのコンテナは、データベース全体を通して固有でなければなりません。1 つのコンテナは、1 つの表スペースにのみ属することができます。コンテナごとに異なるサイズにすることができますが、すべてのコンテナが同じサイズの場合にパフォーマンスは最高になります。

container-string の正しい形式は、オペレーティング・システムによって異なります。コンテナは、オペレーティング・システムの通常の方法で指定されます。

リモート・リソース (LAN でリダイレクトされたドライブや NFS でマウントされたファイル・システムなど) は、現在、Network Appliance Filers、IBM iSCSI、または IBM Network Attached Storage を使用する場合にのみサポートされます。

on-db-partitions-clause

パーティション・データベースにおいて、コンテナを作成するパーティションを指定します。この文節を指定しない場合、他のどの *on-db-partitions-clause* にも明示的に指定されていないデータベース・パーティション・グループ内のパーティションでコンテナが作成されます。データベース・パーティション・グループ IBMTEMPGROUP で定義されている SYSTEM TEMPORARY 表スペースについては、*on-db-partitions-clause* を指定しないと、データベースに追加されたすべての新しいパーティションでもコンテナが作成されます。

CREATE TABLESPACE

on-db-partitions-clause

パーティション・データベースにおいて、コンテナを作成するパーティションを指定します。

ON DBPARTITIONNUMS

特定のパーティションを指定することを示すキーワードです。
DBPARTITIONNUM は DBPARTITIONNUMS の同義語です。

db-partition-number1

データベース・パーティション番号を指定します。

TO *db-partition-number2*

パーティション番号の範囲を指定します。 *db-partition-number2* の値は、*db-partition-number1* の値よりも大きいか等しい値でなければなりません (SQLSTATE 428A9)。この表スペースのデータベース・パーティション・グループにパーティションが組み込まれると、コンテナが作成されるパーティションに、指定したパーティション番号の範囲 (指定したパーティション番号を含む) のすべてのパーティションが組み込まれます。

番号によって指定するパーティションと、パーティションの範囲内のすべてのパーティションは、表スペースを定義するデータベース・パーティション・グループに存在している必要があります (SQLSTATE 42729)。あるパーティション番号を明示的に、または範囲の中で指定できるのは、このステートメントのただ 1 つの *on-db-partitions-clause* の中だけです (SQLSTATE 42613)。

EXTENTSIZE *number-of-pages*

次のコンテナに移る前にコンテナに書き込まれる PAGESIZE ページの数を指定します。このエクステント・サイズ値は、後に K (K バイトの場合)、または M (M バイトの場合) を付けた整数値として指定することもできます。このように指定した場合、バイト数をページ・サイズで割った値が、エクステント・サイズの値を判別するために使用します。データが保管されていくにつれて、データベース・マネージャーはコンテナ間を繰り返し循環します。

デフォルト値は DFT_EXTENT_SZ データベース構成パラメーターによって指定されます。有効な範囲は 2 ~ 256 ページです。

PREFETCHSIZE *number-of-pages*

データのプリフェッチの実行中に、表スペースから読み取られる PAGESIZE ページの数を指定します。このプリフェッチ・サイズ値は、後に K (K バイトの場合)、M (M バイトの場合)、または G (G バイトの場合) を付けた整数値としても指定できます。このように指定した場合、ページ・サイズで分割されたバイト数のフロアが、プリフェッチ・サイズのページ値の数を判別するために使用されます。プリフェッチでは、照会に必要なデータがその照会で参照される前に読み取られるため、照会では入出力の実行を待たずに済みます。

デフォルト値は DFT_PREFETCH_SZ 構成パラメーターによって指定されます。

BUFFERPOOL *bufferpool-name*

この表スペースの表に対して使用するバッファ・プールの名前を指定します。バッファ・プールは存在している必要があります (SQLSTATE 42704)。これを指定しない場合、デフォルトのバッファ・プール (IBMDEFAULTBP) が使用されます。バッファ・プールのページ・サイズは、表スペースに指定された (またはデフォルト指定された) ページ・サイズと一致していなければなりません (SQLSTATE 428CB)。バッファ・プールに対して、この表スペースのデータベース・パーティション・グループを定義する必要があります (SQLSTATE 42735)。

OVERHEAD *number-of-milliseconds*

入出力制御装置のオーバーヘッドとディスク・シーク待ち時間をミリ秒単位で指定する数値リテラルです (整数、10 進数、または浮動小数点数)。この数値がすべてのコンテナで同一でない場合、それは表スペースに属するすべてのコンテナの平均でなければなりません。この値は、照会の最適化の過程で入出力コストを判別するのに使用されます。

TRANSFERRATE *number-of-milliseconds*

number-of-milliseconds は、1 ページをメモリーに読み込むための時間をミリ秒単位で指定する数値リテラルです (整数、10 進数、浮動小数点数)。この数値がすべてのコンテナで同一でない場合、それは表スペースに属するすべてのコンテナの平均でなければなりません。この値は、照会の最適化の過程で入出力コストを判別するのに使用されます。

DROPPED TABLE RECOVERY

ROLLFORWARD コマンドの RECOVER TABLE ON オプションを使用すれば、指定した表スペースでドロップされた表を回復できる場合があります。この文節は、REGULAR 表スペースにのみ指定できます (SQLSTATE 42613)。

注:• **互換性**

- 以前のバージョンの DB2 との互換性:
 - DBPARTITIONNUM の代わりに NODE を指定できます。
 - DBPARTITIONNUMS の代わりに NODES を指定できます。
 - DATABASE PARTITION GROUP の代わりに NODEGROUP を指定できます。
 - LARGE の代わりに LONG を指定できます。
- 表スペースをデータベース管理スペースにするか、それともシステム管理スペースにするかの選択は、トレードオフの関係を含む基本的な選択です。
- 各コンテナ定義には、53 バイトに加えて、コンテナ名を保管するのに必要なバイト数が必要です。表スペースのすべてのコンテナ名を結合した長さは、20 480 バイトを超えることはできません (SQLSTATE 54034)。
- データベースに複数の一時 (TEMPORARY) 表スペースが存在する場合、それらは、使用率のバランスを調整するために、「ラウンドロビン」式に使用されます。

CREATE TABLESPACE

- パーティション・データベースで、複数のデータベース・パーティションが同じ物理ノードに存在する場合、このようなデータベース・パーティションに同じ装置または特定のパスを指定することはできません (SQLSTATE 42730)。この環境の場合、それぞれのデータベース・パーティションごとに固有の *container-string* を指定するか、または相対パス名を使用してください。
- SMS または DMS コンテナの作成時にコンテナ・ストリング構文にデータベース・パーティション式を指定することができます。データベース・パーティション式は一般に、パーティション・データベース・システムで複数の論理データベース・パーティションを使用する場合に指定します。この指定により、コンテナ名がノード (データベース・パーティション・サーバー) 間で固有のものとなります。この式を指定する場合、データベース・パーティション番号はコンテナ名の一部となるか、あるいは、追加の引き数を指定すれば、引き数の結果はコンテナ名の一部となります。

データベース・パーティション式を示すには、引き数 " \$N" ([ブランク]\$N) を使用します。データベース・パーティション式はコンテナ名内で自由に使用できます。また、複数のデータベース・パーティションを指定できます。データベース・パーティション式を終了するにはスペース文字を使用します。スペースの後ろにあるものは、データベース・パーティション式が評価された後でコンテナ名に付加されます。コンテナ名の中で、データベース・パーティション式の後ろにスペースがない場合は、ストリングの残りは式の一部であると見なされます。引き数は、以下のいずれかの形式でのみ使用できます。

表 273. コンテナを作成するための引き数：演算子は左から右へと評価されます。この例では、データベース・パーティション番号は 5 であると想定します。

構文	例	値
[blank]\$N	" \$N"	5
[blank]\$N+[number]	" \$N+1011"	1016
[blank]\$N%[number]	" \$N%3" ^a	2
[blank]\$N+[number]%[number]	" \$N+12%13"	4
[blank]\$N%[number]+[number]	" \$N%3+20"	22

^a % はモジュラスです。

例:

```
CREATE TABLESPACE TS1 MANAGED BY DATABASE USING
(device '/dev/rcont $N' 20000)
```

On a two database partition system, the following containers would be created:

```
/dev/rcont0 - on DATABASE PARTITION 0
/dev/rcont1 - on DATABASE PARTITION 1
```

```
CREATE TABLESPACE TS2 MANAGED BY DATABASE USING
(file '/DB2/containers/TS2/container $N+100' 10000)
```

CREATE TABLESPACE

On a four database partition system, the following containers would be created:

```
/DB2/containers/TS2/container100 - on DATABASE PARTITION 0
/DB2/containers/TS2/container101 - on DATABASE PARTITION 1
/DB2/containers/TS2/container102 - on DATABASE PARTITION 2
/DB2/containers/TS2/container103 - on DATABASE PARTITION 3
```

CREATE TABLESPACE TS3 MANAGED BY SYSTEM USING

```
(' /TS3/cont $N%2', '/TS3/cont $N%2+2')
```

On a two database partition system, the following containers would be created:

```
/TS3/cont0 - On DATABASE PARTITION 0
/TS3/cont2 - On DATABASE PARTITION 0
/TS3/cont1 - On DATABASE PARTITION 1
/TS3/cont3 - On DATABASE PARTITION 1
```

データベース・パーティションが 5 の場合、以下のコンテナは、

```
'/dbdir/node $N /cont1'
'/ $N+1000 /file1'
'$N%10 /container'
'/dir/ $N%5+2000 /dmscont'
```

以下のように作成されます。

```
'/dbdir/node5/cont1'
'/1005/file1'
'5/container'
'/dir/2000/dmscont'
```

例:

例 1: UNIX 系システムで、それぞれ 10 000 の 4K ページの 3 つの装置を使用する通常の DMS 表スペースを作成します。それらの入出力特性も指定します。

```
CREATE TABLESPACE PAYROLL
MANAGED BY DATABASE
USING (DEVICE '/dev/rhdisk6' 10000,
       DEVICE '/dev/rhdisk7' 10000,
       DEVICE '/dev/rhdisk8' 10000)
OVERHEAD 24.1
TRANSFERRATE 0.9
```

例 2: 3 つの別個のドライブの 3 つのディレクトリーを使用し、エクステント・サイズを 64 ページ、プリフェッチ・サイズを 32 ページに指定して、Windows NT/2000 で通常の SMS 表スペースを作成します。

```
CREATE TABLESPACE ACCOUNTING
MANAGED BY SYSTEM
USING ('d:¥acc_tbsp', 'e:¥acc_tbsp', 'f:¥acc_tbsp')
EXTENTSIZ 64
PREFETCHSIZE 32
```


CREATE TABLESPACE

例 3: それぞれ 50,000 ページの 2 つのファイル、および 256 ページのエクステン
ト・サイズを使用して、UNIX で一時 DMS 表スペースを作成します。

```
CREATE TEMPORARY TABLESPACE TEMPSPACE2
MANAGED BY DATABASE
USING (FILE '/tmp/tempspace2.f1' 50000,
FILE '/tmp/tempspace2.f2' 50000)
EXTENTSIZE 256
```

例 4: UNIX のデータベース・パーティションで、データベース・パーティション・グ
ループ ODDNODEGROUP (パーティション 1、3、5) に DMS 表スペースを作成しま
す。すべてのパーティションで、装置 /dev/rhdisk0 の 10 000 個の 4K ページを使用し
ます。また、それぞれのパーティションに、40 000 個の 4K ページがあるパーティシ
ョン固有の装置を指定します。

```
CREATE TABLESPACE PLANS
MANAGED BY DATABASE
USING (DEVICE '/dev/rhdisk0' 10000, DEVICE '/dev/rn1hd01' 40000)
ON DBPARTITIONNUM (1)
USING (DEVICE '/dev/rhdisk0' 10000, DEVICE '/dev/rn3hd03' 40000)
ON DBPARTITIONNUM (3)
USING (DEVICE '/dev/rhdisk0' 10000, DEVICE '/dev/rn5hd05' 40000)
ON DBPARTITIONNUM (5)
```

関連サンプル:

- 『tbtemp.sqc -- How to use a declared temporary table (C)』
- 『TbTemp.java -- How to use Declared Temporary Table (JDBC)』

DROP

DROP ステートメントは、オブジェクトを削除します。そのオブジェクトに直接または
間接的に従属するオブジェクトがある場合、それらも削除されるか、または作動不能に
なります。オブジェクトを削除すると、その記述がカタログから削除され、そのオブジ
ェクトを参照するパッケージがあれば無効になります。

呼び出し:

このステートメントはアプリケーション・プログラムに組み込むことができ、また動的
SQL ステートメントを使用して出すことができます。DYNAMICRULES の実行動作が
パッケージに効力を持つ場合にのみ、動的に準備できる実行可能ステートメントです
(SQLSTATE 42509)。

許可:

2 つの部分からなる名前が使用可能なオブジェクトをドロップする場合、DROP ステ
ートメントの許可 ID が持つ特権には、以下のいずれかが含まれている必要があります。
これらが 1 つも含まれていない場合はエラーが戻されます (SQLSTATE 42501)。

DROP

ALIAS	<i>alias-name</i>	
BUFFERPOOL	<i>bufferpool-name</i>	
EVENT MONITOR	<i>event-monitor-name</i>	
FUNCTION	<i>function-name</i>	RESTRICT
	(<i>data-type</i>)	
		RESTRICT
SPECIFIC FUNCTION	<i>specific-name</i>	
FUNCTION MAPPING	<i>function-mapping-name</i>	
	(1)	
INDEX	<i>index-name</i>	
INDEX EXTENSION	<i>index-extension-name</i>	RESTRICT
METHOD	<i>method-name</i>	FOR <i>type-name</i> RESTRICT
	(<i>datatype</i>)	
		RESTRICT
SPECIFIC METHOD	<i>specific-name</i>	
NICKNAME	<i>nickname</i>	
DATABASE PARTITION GROUP	<i>db-partition-group-name</i>	
PACKAGE	<i>schema-name.</i> <i>package-id</i>	VERSION <i>version-id</i> RESTRICT
PROCEDURE	<i>procedure-name</i>	RESTRICT
	(<i>data-type</i>)	
		RESTRICT
SPECIFIC PROCEDURE	<i>specific-name</i>	
SCHEMA	<i>schema-name</i>	RESTRICT
SEQUENCE	<i>sequence-name</i>	RESTRICT
SERVER	<i>server-name</i>	
TABLE	<i>table-name</i>	
TABLE HIERARCHY	<i>root-table-name</i>	
	<i>tablespace-name</i>	
TABLESPACE	<i>tablespace-name</i>	
TABLESPACES	<i>tablespace-name</i>	
TRANSFORM	ALL FOR <i>type-name</i>	
TRANSFORMS	<i>group-name</i>	
TRIGGER	<i>trigger-name</i>	
	TYPE <i>type-name</i> RESTRICT	
	(2)	
	DISTINCT	
TYPE MAPPING	<i>type-mapping-name</i>	
USER MAPPING FOR	<i>authorization-name</i> SERVER <i>server-name</i>	
	USER	
VIEW	<i>view-name</i>	
VIEW HIERARCHY	<i>root-view-name</i>	
WRAPPER	<i>wrapper-name</i>	

注:

- 1 *Index-name* には、索引、あるいは索引指定のどちらかの名前を指定できます。

- 任意のユーザー定義タイプをドロップするのに DATA を使用することもできます。

説明:**ALIAS** *alias-name*

ドロップする別名を指定します。 *alias-name* (別名) は、カタログに記述されている別名を指定する名前であればなりません (SQLSTATE 42704)。指定した別名は削除されます。

別名を参照するすべての表、ビュー、およびトリガーは作動不能になります。(これには、CREATE TRIGGER ステートメントの ON 文節で参照されている表と、トリガー SQL ステートメントで参照されているすべての表が含まれます。)

BUFFERPOOL *bufferpool-name*

ドロップするバッファー・プールを指定します。 *bufferpool-name* (バッファー・プール名) は、カタログに記述されているバッファー・プールを指定していなければなりません (SQLSTATE 42704)。そのバッファー・プールに表スペースが割り当てられていない場合もあります (SQLSTATE 42893)。 IBMDEFAULTBP バッファー・プールはドロップできません (SQLSTATE 42832)。 DB2 で使用するバッファー・プール・メモリーは、すぐに解放されます。ディスク装置は、次にデータベースへ接続するときまで解放できません。

EVENT MONITOR *event-monitor-name*

ドロップするイベント・モニターを指定します。 *event-monitor-name* (イベント・モニター名) は、すでにカタログに存在するイベント・モニターを指定していなければなりません (SQLSTATE 42704)。

指定したイベント・モニターが ON の場合は、エラー (SQLSTATE 55034) が戻されます。それ以外の場合は、イベント・モニターは削除されます。

イベント・モニターをドロップする時点でそのイベント・モニターのターゲット・パスにイベント・ファイルが存在する場合、そのイベント・ファイルは削除されません。ただし、それと同じターゲット・パスを指定した新しいイベント・モニターが作成されると、それらのイベント・ファイルは削除されます。

WRITE TO TABLE イベント・モニターをドロップする場合、表情報は SYSCAT.EVENTTABLES カatalog・ビューからドロップされますが、表そのものはドロップされません。

FUNCTION

ドロップするユーザー定義関数 (完全な関数または関数テンプレートのいずれか) のインスタンスを指定します。指定する関数インスタンスは、カタログに記述されたユーザー定義関数でなければなりません。 CREATE DISTINCT TYPE ステートメントによって暗黙に生成された関数はドロップできません。

関数のインスタンスを指定する方法としては、以下のようにいくつかの方法があります。

FUNCTION *function-name*

特定の関数を指定します。 *function-name* (関数名) の関数インスタンスが 1 つだけ存在している場合にのみ有効です。このように指定する関数には、任意の数のパラメーターが定義されていても構いません。動的 SQL ステートメントでは、CURRENT SCHEMA 特殊レジスターが、修飾子のないオブジェクト名の修飾子として使用されます。静的 SQL ステートメントでは、QUALIFIER プリコンパイル/バインド・オプションにより、修飾子のないオブジェクト名の修飾子が暗黙指定されます。指定したスキーマまたは暗黙のスキーマにこの名前前の関数が存在しない場合は、エラー (SQLSTATE 42704) になります。指定したスキーマまたは暗黙のスキーマに、この関数の特定インスタンスが複数存在する場合は、エラー (SQLSTATE 42725) になります。

FUNCTION *function-name (data-type,...)*

ドロップする関数を固有に指定する関数シグニチャーを指定します。関数選択のアルゴリズムは使用されません。

function-name

ドロップする関数の関数名を指定します。動的 SQL ステートメントでは、CURRENT SCHEMA 特殊レジスターが、修飾子のないオブジェクト名の修飾子として使用されます。静的 SQL ステートメントでは、QUALIFIER プリコンパイル/バインド・オプションにより、修飾子のないオブジェクト名の修飾子が暗黙指定されます。

(data-type,...)

これは、CREATE FUNCTION ステートメント上で (対応する位置に) 指定されたデータ・タイプに一致していなければなりません。データ・タイプ (*data-type*) の数、およびデータ・タイプを論理的に連結した値が、ドロップする特定の関数インスタンスを識別するのに使用されます。

data-type が修飾なしの場合は、SQL パス上でスキーマを検索することによってタイプ名が決定されます。REFERENCE タイプに指定するデータ・タイプ名にも同様の規則が当てはまります。

パラメーター化データ・タイプの長さ、精度、または位取りを指定する必要はありません。空の括弧をコーディングすることによって、一致データ・タイプの検索時にそれらの属性を無視するように指定することができます。

パラメーター値が異なるデータ・タイプ (REAL または DOUBLE) を示しているため、FLOAT() を使用することはできません (SQLSTATE 42601)。

長さ、精度、または位取りをコーディングする場合、その値は、CREATE FUNCTION ステートメントで指定された値と完全に一致していなければなりません。

0<n<25 は REAL を意味し、24<n<54 は DOUBLE を意味するので、FLOAT(n) のタイプは、n に定義された値と一致している必要はありません。マッチングは、タイプが REAL か DOUBLE かに基づいて行われます。

RESTRICT

RESTRICT キーワードを指定すると、以下のいずれかの従属関係が存在する場合は、関数をドロップしないという規則が適用されます。

- 別のルーチンがその関数に基づいている。
- ビューがその関数を使用している。
- トリガーがその関数を使用している。

RESTRICT は、デフォルトの動作です。

指定したスキーマまたは暗黙のスキーマに、指定したシグニチャーを持つ関数がない場合は、エラー (SQLSTATE 42883) になります。

SPECIFIC FUNCTION *specific-name*

関数の作成時に指定された特定関数名、またはデフォルト値として使用された特定関数名を使用して、ドロップする特定のユーザー定義関数を指定します。動的 SQL ステートメントでは、CURRENT SCHEMA 特殊レジスターが、修飾子のないオブジェクト名の修飾子として使用されます。静的 SQL ステートメントでは、QUALIFIER プリコンパイル/バインド・オプションにより、修飾子のないオブジェクト名の修飾子が暗黙指定されます。 *specific-name* (特定名) は、指定したスキーマまたは暗黙のスキーマの特定関数のインスタンスを指定していなければなりません。そうでない場合、エラー (SQLSTATE 42704) になります。

RESTRICT

RESTRICT キーワードを指定すると、以下のいずれかの従属関係が存在する場合は、関数をドロップしないという規則が適用されます。

- 別のルーチンがその関数に基づいている。
- ビューがその関数を使用している。
- トリガーがその関数を使用している。

RESTRICT は、デフォルトの動作です。

SYSIBM、SYSFUN、または SYSPROC スキーマ (SQLSTATE 42832) の関数はドロップできません。

他のオブジェクトが関数に従属している場合があります。そのような関数をドロップする場合には、その前にそのような従属オブジェクトをすべてドロップしておく必要があります (作動不能としてマークされるパッケージは除く)。従属オブジェクトを伴う関数をドロップしようとする、エラー (SQLSTATE 42893) になります。それらの従属関係のリストについては、1591 ページを参照してください。

関数がドロップ可能な場合、その関数がドロップされます。

ドロップする特定関数に付属するパッケージがある場合には、それは作動不能としてマークされます。そのようなパッケージが暗黙のうちに再バインドされることはありません。 **BIND** コマンドまたは **REBIND** コマンドを使って再バインドするか、 **PREP** コマンドを使って再作成する必要があります。

FUNCTION MAPPING *function-mapping-name*

ドロップする関数マッピングを指定します。 *function-mapping-name* は、カタログに記述されているユーザー定義関数マッピングを指定していなければなりません (SQLSTATE 42704)。その関数マッピングはデータベースから削除されます。

デフォルトの関数マッピングは、ドロップできませんが、使用不可にすることができます。

ドロップされる関数マッピングに付属しているパッケージは、無効になります。

INDEX *index-name*

ドロップする索引または索引指定を指定します。 *index-name* (索引名) は、カタログに記述されている索引または索引指定を識別していなければなりません (SQLSTATE 42704)。索引は、システムに必須の主キーまたはユニーク制約の索引であってはならず、複製されたマテリアライズ照会表の索引であってもなりません (SQLSTATE 42917)。指定した索引または索引指定は削除されます。

ドロップされる索引または索引指定に付属しているパッケージは、無効になります。

INDEX EXTENSION *index-extension-name* **RESTRICT**

ドロップする索引拡張を指定します。 *index-extension-name* (索引拡張名) は、カタログに記述されている索引拡張を指定する名前であればなりません (SQLSTATE 42704)。 **RESTRICT** キーワードは、この索引拡張の定義にしたがって索引を定義できないという規則を課します (SQLSTATE 42893)。

METHOD

ドロップするメソッド本体を指定します。指定するメソッドの本体は、カタログに記述されているメソッドでなければなりません (SQLSTATE 42704)。 **CREATE TYPE** ステートメントによって暗黙的に生成されたメソッドをドロップすることはできません。

DROP METHOD によって、メソッドの本体は削除されますが、メソッドの指定 (シグニチャー) はサブジェクト・タイプの定義の一部として残されます。メソッドの本体をドロップした後、メソッドの指定は **ALTER TYPE DROP METHOD** を使用してサブジェクト・タイプの定義から削除することができます。

ドロップするメソッド本体は、以下のようないくつかの方法で指定することができます。

METHOD *method-name*

ドロップする特定のメソッドを指定します。この方法は、対象となるタイプ

type-name に、*method-name* という名前のメソッド・インスタンスが 1 つしかないことが明らかな場合にのみ有効です。この方法を用いる場合は、メソッドにいくつのパラメーターが定義されていても構いません。タイプ *type-name* に、指定された名前のメソッドが存在しない場合は、エラーが戻されます (SQLSTATE 42704)。指定されたデータ・タイプに、そのメソッドの特定のインスタンスが複数存在する場合も、エラーが戻されます (SQLSTATE 42725)。

METHOD *method-name* (*data-type*,...)

ドロップするメソッドを一意的に識別できるメソッド・シグニチャーを指定します。メソッド選択のアルゴリズムは使用されません。

method-name

指定したタイプの中から、ドロップするメソッドのメソッド名を指定します。指定する名前は、修飾なしの ID でなければなりません。

(*data-type*,...)

データ・タイプを指定します。ここで指定されるデータ・タイプは、CREATE TYPE または ALTER TYPE ステートメントで、メソッドの指定の対応する位置に指定されたデータ・タイプと一致していなければなりません。データ・タイプの数とデータ・タイプを論理的に連結した値から、ドロップする特定のメソッドが識別されます。

data-type が修飾なしの場合は、SQL パス上でスキーマを検索してタイプ名が決定されます。

パラメーター化データ・タイプの長さ、精度、または位取りを指定する必要はありません。空の括弧をコーディングすることによって、一致データ・タイプの検索時にそれらの属性を無視するように指定することができます。

パラメーター値が異なるデータ・タイプ (REAL または DOUBLE) を示しているため、FLOAT() を使用することはできません (SQLSTATE 42601)。

ただし、長さ、精度、または位取りをコーディングする場合、その値は、CREATE TYPE ステートメントで指定された値と完全に一致していなければなりません。

$0 < n < 25$ は REAL を意味し、 $24 < n < 54$ は DOUBLE を意味するので、FLOAT(*n*) のタイプは、*n* に定義された値と一致している必要はありません。マッチングは、タイプが REAL か DOUBLE かに基づいて行われます。

指定されたデータ・タイプに、指定されたシグニチャーを持つメソッドが存在しない場合は、エラーが戻されます (SQLSTATE 42883)。

FOR *type-name*

指定したメソッドのドロップを行うタイプを指定します。ここで指定される名前は、カタログにすでに記述されているタイプを示すものでなければなりません。

ん (SQLSTATE 42704)。動的 SQL ステートメントでは、CURRENT SCHEMA 特殊レジスターが、修飾子のないタイプ名の修飾子として使用されます。静的 SQL ステートメントでは、QUALIFIER プリコンパイル / バインド・オプションによって、修飾子のないタイプ名に修飾子が暗黙指定されます。

RESTRICT

RESTRICT キーワードを指定すると、以下のいずれかの従属関係が存在する場合は、メソッドをドロップしないという規則が適用されます。

- 別のルーチンがそのメソッドに基づいている。
- ビューがそのメソッドを使用している。
- トリガーがそのメソッドを使用している。

RESTRICT は、デフォルトの動作です。

SPECIFIC METHOD *specific-name*

CREATE TYPE または ALTER TYPE ステートメントにおいてユーザーが指定した名前、もしくはデフォルトで指定された名前を使用して、ドロップする特定のメソッドを識別します。特定名 (*specific-name*) に修飾子が付いていない場合、動的 SQL ステートメントでは、CURRENT SCHEMA 特殊レジスターが、修飾子のない特定名の修飾子として使用されます。静的 SQL ステートメントでは、QUALIFIER プリコンパイル / バインド・オプションにより、修飾子のない特定名に修飾子が暗黙指定されます。 *specific-name* に指定される名前は、メソッドの名前でなければなりません。メソッド名ではない名前が指定された場合は、エラーになります (SQLSTATE 42704)。

RESTRICT

RESTRICT キーワードを指定すると、以下のいずれかの従属関係が存在する場合は、メソッドをドロップしないという規則が適用されます。

- 別のルーチンがそのメソッドに基づいている。
- ビューがそのメソッドを使用している。
- トリガーがその関数を使用している。

RESTRICT は、デフォルトのメソッドです。

メソッドに他のオブジェクトが従属している場合があります。そのような場合は、メソッドをドロップする前にそれらの従属関係をすべて除去する必要があります (ただし、そのメソッドがドロップされると作動不能としてマークされるパッケージは例外です)。そのような従属関係を持つメソッドをドロップしようとする、エラーが戻されます (SQLSTATE 42893)。

ドロップできる状態にあれば、メソッドはドロップされます。

ドロップする特定のメソッドに従属しているパッケージは、そのメソッドがドロップされると、作動不能としてマークされます。そのようなパッケージが暗黙的に再

バインドされることはありません。これらのパッケージは、BIND コマンドまたは REBIND コマンドを使用して再バインドするか、あるいは PREP コマンドを使用して再作成する必要があります。

ドロップされる特定のメソッドが別のメソッドをオーバーライドする場合、オーバーライドされるメソッド (および、ドロップされる特定のメソッドのスーパータイプでこのメソッドをオーバーライドするメソッド) に従属したパッケージはすべて無効になります。

NICKNAME *nickname*

削除するニックネームを指定します。ニックネームは、カタログにリストされていなければなりません (SQLSTATE 42704)。そのニックネームはデータベースから削除されます。

ニックネームに関連した列および索引に関するすべての情報が、カタログから削除されます。ニックネームに従属したマテリアライズ照会表はドロップされます。ニックネームに従属した索引指定はドロップされます。ニックネームに従属するビューは、作動不能としてマークされます。ドロップされた索引指定または作動不能ビューに従属するパッケージはいずれも無効になります。ニックネームが参照するデータ・ソースは影響を受けません。

DATABASE PARTITION GROUP *db-partition-group-name*

ドロップするデータベース・パーティション・グループを指定します。

db-partition-group-name パラメーターは、カタログに記述されているデータベース・パーティション・グループを指定していなければなりません (SQLSTATE 42704)。これは、1 つの部分からなる名前です。

データベース・パーティション・グループをドロップすると、データベース・パーティション・グループで定義されたすべての表スペースがドロップされます。そのような表スペース内の表に対して従属関係がある既存のデータベース・オブジェクト (パッケージや参照制約など) は、ドロップされるか、または無効になり (該当する場合)、従属するビューとトリガーは作動不能になります。

システム定義のデータベース・パーティション・グループはドロップできません (SQLSTATE 42832)。

現在データ再分散が行われているデータベース・パーティション・グループに対して DROP DATABASE PARTITION GROUP ステートメントを発行すると、データベース・パーティション・グループのドロップ操作は失敗し、エラーが戻されます (SQLSTATE 55038)。ただし、部分的に再分散されたデータベース・パーティション・グループはドロップできます。データベース・パーティション・グループは、REDISTRIBUTE DATABASE PARTITION GROUP コマンドが完了するまで実行されなかった場合は、部分的に再分散の状態になります。これは、エラーまたは FORCE APPLICATION ALL コマンドによって割り込まれた場合に起こる可能性が

DROP

あります。(部分的に再分散されたデータベース・パーティション・グループの場合、SYSCAT.DBPARTITIONGROUPS カタログの REBALANCE_PMAP_ID は -1 ではありません。)

PACKAGE *schema-name.package-id*

ドロップするパッケージを指定します。スキーマ名が指定されていない場合、パッケージ ID は暗黙的にデフォルト・スキーマになります。スキーマ名およびパッケージ ID は、明示的または暗黙的に指定されたバージョン ID とともに、カタログに記述されているパッケージを指定していなければなりません (SQLSTATE 42704)。指定したパッケージが削除されます。ドロップするパッケージが、*schema-name.package-id* で指定された唯一のパッケージである場合 (つまり、他のバージョンは存在しない場合)、そのパッケージに対する特権もすべて削除されます。

VERSION *version-id*

ドロップするパッケージ・バージョンを指定します。値が指定されない場合には、空ストリングがバージョンのデフォルトになります。同じパッケージ名が付けられていてもバージョンは異なる、複数のパッケージが存在する場合、DROP ステートメントを 1 回呼び出すときに、1 つのパッケージ・バージョンだけをドロップできます。

PROCEDURE

ドロップするストアード・プロシージャのインスタンスを指定します。指定するプロシージャ・インスタンスは、カタログに記述されたストアード・プロシージャでなければなりません。

プロシージャ・インスタンスを指定する方法としては、以下のようにいくつかの方法があります。

PROCEDURE *procedure-name*

特定のプロシージャを指定します。この方法は、*procedure-name* で指定したプロシージャ・インスタンスがスキーマ内に 1 つしか存在しないことが明らかの場合にのみ有効です。この方法で指定するプロシージャには、パラメーターがいくつ定義されていても構いません。指定したスキーマまたは暗黙のスキーマに該当する名前のプロシージャが存在しない場合は、エラーが戻されます (SQLSTATE 42704)。動的 SQL ステートメントでは、CURRENT SCHEMA 特殊レジスターが、修飾子のないオブジェクト名の修飾子として使用されます。静的 SQL ステートメントでは、QUALIFIER プリコンパイルバインド・オプションにより、修飾子のないオブジェクト名の修飾子が暗黙指定されます。指定したスキーマまたは暗黙のスキーマにこのプロシージャの特定インスタンスが複数存在する場合は、エラーが戻されます (SQLSTATE 42725)。

RESTRICT

RESTRICT キーワードを指定すると、トリガー定義にプロシージャの名

前が付けられた CALL ステートメントが含まれる場合に、そのプロシージャーはドロップされずに済みます。RESTRICT は、デフォルトの動作です。

PROCEDURE *procedure-name* (*data-type*,...)

ドロップするプロシージャーを一意に識別するプロシージャー・シグニチャーを指定します。プロシージャー選択のアルゴリズムは使用されません。

procedure-name

ドロップするプロシージャーのプロシージャー名を指定します。動的 SQL ステートメントでは、CURRENT SCHEMA 特殊レジスターが、修飾子のないオブジェクト名の修飾子として使用されます。静的 SQL ステートメントでは、QUALIFIER プリコンパイル/バインド・オプションにより、修飾子のないオブジェクト名の修飾子が暗黙指定されます。

(*data-type*,...)

データ・タイプを指定します。ここで指定されるデータ・タイプは、CREATE PROCEDURE ステートメントの対応する位置に指定されたデータ・タイプと一致していなければなりません。データ・タイプ (*data-type*) の数、およびデータ・タイプを論理的に連結した値を使用して、ドロップする特定のプロシージャーが識別されます。

data-type が修飾なしの場合は、SQL パス上でスキーマを検索することによってタイプ名が決定されます。REFERENCE タイプに指定するデータ・タイプ名にも同様の規則が当てはまります。

パラメーター化データ・タイプの長さ、精度、または位取りを指定する必要はありません。空の括弧をコーディングすることによって、一致データ・タイプの検索時にそれらの属性を無視するように指定することができます。

パラメーター値が異なるデータ・タイプ (REAL または DOUBLE) を示しているため、FLOAT() を使用することはできません (SQLSTATE 42601)。

ただし、長さ、精度、または位取りをコーディングする場合、その値は、CREATE FUNCTION ステートメントにおける指定に完全に一致していなければなりません。

$0 < n < 25$ は REAL を意味し、 $24 < n < 54$ は DOUBLE を意味するので、FLOAT(*n*) のタイプは、*n* に定義された値と一致している必要はありません。マッチングは、タイプが REAL か DOUBLE かに基づいて行われます。

指定したスキーマまたは暗黙のスキーマに、指定されたシグニチャーを持つプロシージャーがない場合は、エラーが戻されます (SQLSTATE 42883)。

SPECIFIC PROCEDURE *specific-name*

プロシージャの作成時にユーザーが指定した特定のプロシージャ名か、デフォルト値として与えられたプロシージャ名を使用して、ドロップする特定のストアード・プロシージャを識別します。動的 SQL ステートメントでは、CURRENT SCHEMA 特殊レジスターが、修飾子のないオブジェクト名の修飾子として使用されます。静的 SQL ステートメントでは、QUALIFIER プリコンパイルバインド・オプションにより、修飾子のないオブジェクト名の修飾子が暗黙指定されます。 *specific-name* に指定される名前は、指定したスキーマまたは暗黙のスキーマに含まれる特定プロシージャのインスタンスを識別するものでなければなりません。それ以外の名前が指定された場合は、エラーが戻されます (SQLSTATE 42704)。

RESTRICT

RESTRICT キーワードを指定すると、トリガー定義にプロシージャの名前が付けられた CALL ステートメントが含まれる場合に、そのプロシージャはドロップされずに済みます。 RESTRICT は、デフォルトの動作です。

SYSIBM、SYSFUN、または SYSPROC スキーマのプロシージャはドロップできません (SQLSTATE 42832)。

SCHEMA *schema-name* **RESTRICT**

ドロップする特定のスキーマを指定します。 *schema-name* に指定するスキーマ名は、カタログに記述されているスキーマを識別するものでなければなりません (SQLSTATE 42704)。 RESTRICT キーワードは、データベースから削除するスキーマとして指定したスキーマにオブジェクトを定義できないという規則を課します (SQLSTATE 42893)。

SEQUENCE *sequence-name*

ドロップする特定のシーケンスを識別します。暗黙的または明示的スキーマ名を含む *sequence-name* は、現在のサーバーに存在するシーケンスを固有に識別していなければなりません。この名前によるシーケンスが、明示的または暗黙的に指定されたスキーマに存在しない場合、エラー (SQLSTATE 42704) が起こります。

RESTRICT オプションはデフォルトで、以下のいずれかの従属関係が存在する場合は、シーケンスがドロップされないようにします。

- トリガーの NEXTVAL または PREVVVAL 式がこのシーケンスを指定するようなトリガーが存在する (SQLSTATE 42893)。
- ルーチン本体の NEXTVAL 式がこのシーケンスを指定するような SQL 関数または SQL メソッドが存在する (SQLSTATE 42893)。

SERVER *server-name*

カタログから定義をドロップするデータ・ソースを指定します。 *server-name* に指

定するサーバー名は、カタログに記述されているデータ・ソースを識別するものでなければなりません (SQLSTATE 42704)。そのデータ・ソースの定義は削除されません。

データ・ソースに常駐する表およびビューのニックネームはすべてドロップされます。また、これらのニックネームに従属する索引指定もすべてドロップされます。ドロップされたサーバー定義に従属するユーザー定義関数マッピング、ユーザー定義タイプ・マッピング、およびユーザー・マッピングもすべてドロップされます。ドロップされたサーバー定義、関数マッピング、ニックネーム、および索引指定に依存するパッケージはすべて無効になります。

TABLE *table-name*

ドロップする基本表、宣言済み一時表、マテリアライズ照会表、またはステージング表を指定します。 *table-name* に指定する表名は、カタログに記述されている表が、宣言済み一時表を指定する場合は、スキーマ名 SESSION によって修飾され、アプリケーションに存在する一時表の名前でなければなりません (SQLSTATE 42704)。型付き表の副表は、それぞれスーパー表に従属しています。したがって、スーパー表をドロップする前には、副表をすべてドロップする必要があります (SQLSTATE 42893)。指定された表はデータベースから削除されます。

その表を参照するすべての索引、主キー、外部キー、チェック制約、マテリアライズ照会表、およびステージング表はドロップされます。表を参照するすべてのビューおよびトリガーは、作動不能になります。(これには、CREATE TRIGGER ステートメントの ON 文節で参照されている表と、トリガー SQL ステートメントで参照されているすべての表が含まれます。)ドロップされたオブジェクトまたは作動不能としてマークされたオブジェクトに従属するすべてのパッケージは無効になります。これには、副表よりも上位の階層であるスーパー表に従属するパッケージが含まれます。参照列の中で、ドロップされた表を参照の有効範囲として定義したものがあれば、参照範囲は無効になります。

宣言済み一時表にパッケージが従属することはありません。したがって、宣言済み一時表がドロップされてもパッケージが無効になることはありません。

DATALINK 列にリンクされたファイルはすべてリンク解除されます。リンク解除操作は非同期で実行されるので、ファイルを他の操作ですぐに使用することはできない場合があります。

連合システムでは、透過性 DDL を使用して作成されたりリモート表はドロップできません。リモート表をドロップすると、その表に関連したニックネームもドロップされ、そのニックネームに従属するパッケージが無効化されます。

表階層から副表をドロップすると、その副表に関連した列はアクセスできなくなります (ただし、列の数や行のサイズの制限に関しては考慮されます)。副表をドロップすると、スーパー表から副表の列がすべて削除されてしまいます。その結果、スーパー表に定義したトリガーや参照保全制約が活動化することがあります。

宣言済み一時表が、現在の作業単位または保管点がアクティブになる前に作成されたものである場合は、その一時表をドロップすると機能上で表がドロップされてしまうため、アプリケーションからその一時表にアクセスすることができなくなります。しかし、表スペースでは、作業単位がコミットされるまで、あるいは保管点が終了するまで、依然としてこの表が予約された状態にあるため、**USER TEMPORARY** 表スペースをドロップしたり、**USER TEMPORARY** 表スペースのデータベース・パーティション・グループを再配布することはできません。宣言済み一時表がドロップされると、**DROP** がコミットされたかロールバックされたかにかかわらず、表に含まれていたデータはすべて破棄されます。

表に **RESTRICT ON DROP** 属性があると、その表はドロップできません。

TABLE HIERARCHY *root-table-name*

ドロップする型付き表階層を指定します。*root-table-name* で指定する型付き表は、型付き表階層のルート表でなければなりません (**SQLSTATE 428DR**)。

root-table-name で指定する型付き表とその表のすべての副表が、データベースから削除されます。

ドロップされた表を参照するすべての索引、マテリアライズ照会表、ステージング表、主キー、外部キー、およびチェック制約はドロップされます。ドロップされた表を参照するすべてのビューおよびトリガーは、作動不能になります。ドロップされたオブジェクトまたは作動不能としてマークされたオブジェクトに從属するすべてのパッケージは無効になります。参照列の中で、ドロップされた表を参照の有効範囲として定義したものがあれば、参照範囲は無効になります。

DATALINK 列にリンクされたファイルはすべてリンク解除されます。リンク解除操作は非同期で実行されるので、ファイルを他の操作ですぐに使用することはできない場合があります。

単一の副表をドロップする場合とは違い、表階層をドロップしても、階層内にある任意の表の削除トリガーが活動化したり、削除された行が記録されたりすることはありません。

TABLESPACE または **TABLESPACES** *tablespace-name*

ドロップされる表スペースを指定します。*tablespace-name* (表スペース名) は、カタログに記述されている表スペースを指定していなければなりません (**SQLSTATE 42704**)。これは、1 つの部分からなる名前です。

表の一部がドロップされる表スペースに保管され、1 つかそれ以上の部分がドロップされない別の表スペースに保管されている場合、この表スペースはドロップされません (このような表は前もってドロップする必要があります)。また、その表スペースに存在する表に **RESTRICT ON DROP** 属性がある場合も、この表スペースはドロップされません (**SQLSTATE 55024**)。システム表スペースはドロップできません (**SQLSTATE 42832**)。データベースに **TEMPORARY** 表スペースが 1 つしか存在しない場合は、**SYSTEM TEMPORARY** 表スペースをドロップすることはできません。宣言済み一時表が作成されている **USER TEMPORARY** 表スペースはドロップできません (**SQLSTATE 55039**)。 **USER TEMPORARY** 表スペースでは、宣言

済み一時表が削除されても、`DROP TABLE` を含む作業単位がコミットされるまでは、その表スペースは使用中と見なされます。

表スペースをドロップすると、その表スペースに定義されているオブジェクトはすべてドロップされます。パッケージや参照制約などのその表スペースに從属する既存のすべてのデータベース・オブジェクトはドロップされるか、または無効になり、從属しているビューやトリガーは作動不能になります。

ユーザーによって作成されたコンテナは削除されません。`CREATE TABLESPACE` でデータベース・マネージャーによって作成されたコンテナ名のパスに含まれているディレクトリーは、いずれも削除されます。データベース・ディレクトリーの下にあるすべてのコンテナは削除されます。`SMS` 表スペースでは、すべての接続が切斷されるか `DEACTIVATE DATABASE` コマンドが出されるまで削除は行われません。

TRANSFORM ALL FOR *type-name*

ユーザー定義データ・タイプ *type-name* に定義されたすべてのトランスフォーム・グループがドロップされることを示します。これらのグループで参照されるトランスフォーム関数はドロップされません。動的 SQL ステートメントでは、`CURRENT SCHEMA` 特殊レジスターが、修飾子のないオブジェクト名の修飾子として使用されます。静的 SQL ステートメントでは、`QUALIFIER` プリコンパイル / バインド・オプションによって、修飾子のないオブジェクト名の修飾子が暗黙指定されます。*type-name* に指定されるタイプ名は、カタログに記述されているユーザー定義タイプを識別するものでなければなりません (SQLSTATE 42704)。

type-name に定義されているトランスフォームが存在しない場合は、エラーが戻されます (SQLSTATE 42740)。

`DROP TRANSFORM` は、`CREATE TRANSFORM` の逆の処理を行います。`DROP TRANSFORM` は、指定されたデータ・タイプで特定のグループに関連付けられたトランスフォーム関数を未定義の状態にします。これらのグループに関連付けられていた関数は引き続き存在しており、明示的に呼び出すことができますが、これらの関数にはもはやトランスフォーム・プロパティーは含まれていないので、ホスト言語環境で値を交換するためにこれらの関数が暗黙的に呼び出されることはありません。

トランスフォーム・グループの中に SQL 以外の言語で書かれたユーザー定義関数 (またはメソッド) があり、その関数が、ユーザー定義タイプ *type-name* に定義されたそのグループのトランスフォーム関数のいずれかに從属している場合、そのトランスフォーム・グループはドロップされません (SQLSTATE 42893)。このようなユーザー定義関数が從属しているトランスフォーム関数は、*type-name* で定義された参照先のトランスフォーム・グループに関連付けられています。そのため、パッケージが属しているトランスフォーム関数が、指定されたトランスフォーム・グループと関連付けられていると、そのパッケージは作動不能としてマークされてしまいます。

TRANSFORMS *group-name* **FOR** *type-name*

ユーザーが定義したデータ・タイプ *type-name* から、指定したトランスフォーム・グループがドロップされることを示します。このグループで参照されるトランスフォーム関数はドロップされません。動的 SQL ステートメントでは、CURRENT SCHEMA 特殊レジスターが、修飾子のないオブジェクト名の修飾子として使用されます。静的 SQL ステートメントでは、QUALIFIER プリコンパイル / バインド・オプションによって、修飾子のないオブジェクト名の修飾子が暗黙指定されます。*type-name* に指定されるタイプ名は、カタログに記述されているユーザー定義タイプを識別するものでなければなりません (SQLSTATE 42704)。また、*group-name* には、*type-name* に存在しているトランスフォーム・グループを指定する必要があります。

TRIGGER *trigger-name*

ドロップするトリガーを指定します。*trigger-name* (トリガー名) は、カタログに記述されているトリガーを指定していなければなりません (SQLSTATE 42704)。指定したトリガーは削除されます。

トリガーをドロップすると、特定のパッケージが無効としてマークされます。

trigger-name がビューに対して INSTEAD OF トリガーを指定する場合、そのビューに対する更新を行うことにより、他のトリガーはそのトリガーに従属できます。

TYPE *type-name*

ドロップするユーザー定義タイプを指定します。動的 SQL ステートメントでは、CURRENT SCHEMA 特殊レジスターが、修飾子のないオブジェクト名の修飾子として使用されます。静的 SQL ステートメントでは、QUALIFIER プリコンパイル / バインド・オプションにより、修飾子のないオブジェクト名の修飾子が暗黙指定されます。構造タイプでは、関連した参照タイプもドロップされます。*type-name* (タイプ名) は、カタログに記述されているユーザー定義タイプを指定していなければなりません。DISTINCT 文節が指定されている場合、*type-name* (タイプ名) は、カタログに記述されている特殊タイプを指定していなければなりません。

RESTRICT

以下の場合、このタイプはドロップされません (SQLSTATE 42893)。

- 表またはビューの列のタイプとして使用されるタイプである。
- サブタイプが含まれている。
- 型付き表または型付きビューのデータ・タイプとして使用されている構造タイプである。
- 他の構造タイプの属性として使用されるタイプである。
- 表の列のタイプに *type-name* のインスタンスが含まれている可能性がある。これには、列のタイプが *type-name* である場合や、列に関連付けられたタイプ階層以外のロケーションで *type-name* が使用される場合などがあります。

もっと典型的な例としては、どのタイプ (T) であれ、表の列のタイプで *type-name* が直接または間接的に使用されている場合には、T をドロップすることはできません。

- タイプが、表またはビューの参照タイプ列のターゲット・タイプ、または別の構造タイプの参照タイプ属性である。
- このタイプ、あるいはこのタイプを参照する値が、ドロップできない関数やメソッドのパラメーター・タイプまたは戻り値タイプである。
- このタイプ、またはこのタイプを参照する値が SQL 関数やメソッドの本体で使用されているが、パラメーター・タイプや戻り値タイプではない。
- このタイプがチェック制約、トリガー、ビュー定義、または索引の拡張で使用されている。

ドロップされるタイプを使用する関数の場合、ユーザー定義タイプがドロップ可能であると、ドロップするそのタイプ (またはドロップするタイプを参照するもの) のパラメーターまたは戻り値が含まれているすべての関数 (F) (特定名は SF) に、以下の DROP FUNCTION ステートメントが実行されることになります。

DROP SPECIFIC FUNCTION SF

このステートメントがカスケードして、従属する関数もドロップされる可能性があります。ユーザー定義タイプへの従属関係に基づいて、それらの関数もすべてドロップ・リストに含まれている場合には、ユーザー定義タイプのドロップは正常に処理されます (そうでない場合、SQLSTATE 42893 のエラーになります)。

ドロップされるタイプを使用するメソッドの場合、ユーザー定義タイプがドロップ可能であると、ドロップするそのタイプ (またはドロップするタイプを参照するもの) のパラメーターまたは戻り値が含まれているタイプ T1 のメソッド (M) (特定名は SM) に、以下のステートメントが実行されることになります。

DROP SPECIFIC METHOD SM ALTER TYPE T1 DROP SPECIFIC METHOD SM

これらのメソッドに従属しているオブジェクトがあると、DROP TYPE 操作が失敗する場合があります。

ドロップするタイプのスーパータイプで定義されるメソッドに従属し、オーバーライドに適したパッケージはすべて、無効になります。

RESTRICT は、デフォルトの動作を反映するものなので、RESTRICT を指定することはオプションです。

TYPE MAPPING *type-mapping-name*

ドロップするユーザー定義のデータ・タイプ・マッピングを指定します。

type-mapping-name (タイプ・マッピング名) は、カタログに記述されているデータ・タイプ・マッピングを指定していなければなりません (SQLSTATE 42704)。指定したデータ・タイプ・マッピングがデータベースから削除されます。

その他にドロップされるオブジェクトはありません。

USER MAPPING FOR *authorization-name* | **USER SERVER** *server-name*

ドロップするユーザー・マッピングを指定します。このマッピングは、連合データベースにアクセスするために使う許可名を、データ・ソースにアクセスするために使う許可名に関連付けます。これら 2 つのうち最初の許可名は、*authorization-name* で指定されるか、または特殊レジスター **USER** によって参照されます。

server-name は、アクセスするのに 2 番目の許可名を使用するデータ・ソースを指定します。

authorization-name は、カタログにリストされていなければなりません (SQLSTATE 42704)。 *server-name* に指定するサーバー名は、カタログに記述されているデータ・ソースを識別するものでなければなりません (SQLSTATE 42704)。ユーザー・マッピングが削除されます。

その他にドロップされるオブジェクトはありません。

VIEW *view-name*

削除するビューを指定します。 *view-name* (ビュー名) は、カタログに記述されているビューを指定していなければなりません (SQLSTATE 42704)。型付き表の副ビューは、それぞれスーパー表に従属しています。したがって、スーパービューをドロップする前に、サブビューをすべてドロップする必要があります (SQLSTATE 42893)。

指定したビューは削除されます。直接的または間接的にそのビューに従属するビューまたはトリガーの定義は、作動不能としてマークされます。作動不能というマークが付いた表に従属するマテリアライズ照会表はすべてドロップされます。ドロップされたビューまたは作動不能としてマークされたビューに従属するパッケージはいずれも無効になります。これには、サブビューよりも上位の階層であるスーパービューに従属するパッケージが含まれます。参照列の中で、ドロップされたビューを参照の有効範囲として定義したものがあれば、参照範囲は無効になります。

VIEW HIERARCHY *root-view-name*

ドロップする型付きビュー階層を指定します。 *root-view-name* で指定する型付きビューは、型付きビュー階層のルート・ビューでなければなりません (SQLSTATE 428DR)。 *root-view-name* で指定する型付きビューとそのビューのすべての副表が、データベースから削除されます。

直接的または間接的にドロップされたビューに従属するビューまたはトリガーの定義は、作動不能としてマークされます。ドロップされたビューやトリガー、または作動不能としてマークされたビューやトリガーに従属するパッケージはいずれも、

無効になります。参照列の中で、ドロップされたビューや作動不能とマークされたビューを参照の有効範囲として定義したものがあれば、参照範囲は無効になります。

WRAPPER *wrapper-name*

ドロップするラッパーを指定します。 *wrapper-name* (ラッパー名) は、カタログに記述されているラッパーを指定していなければなりません (SQLSTATE 42704)。そのラッパーは削除されます。

そのラッパーに従属するすべてのサーバー定義、ユーザー定義関数マッピング、およびユーザー定義データ・タイプ・マッピングはドロップされます。ドロップされたサーバー定義に従属するユーザー定義関数マッピング、ニックネーム、ユーザー定義データ・タイプ・マッピング、およびユーザー・マッピングもすべてドロップされます。ドロップされたニックネームに従属する索引指定はすべてドロップされ、こうしたニックネームに従属するビューはすべて、作動不能としてマークが付けられます。ドロップされたオブジェクトと作動不能ビューに従属するすべてのパッケージは無効になります。

規則:

従属関係: 1592 ページの表 274 は、オブジェクト相互間従属関係を示します。カタログには明示的に記録されない従属関係があります。たとえば、パッケージが従属している制約の記録はありません。このリストには、以下の 4 つの異なるタイプの従属関係が示されています。

- R** 制限 (Restrict) を意味します。従属オブジェクトが存在する限り、その基礎となるオブジェクトはドロップできません。
- C** カスケード (Cascade) を意味します。基礎となるオブジェクトをドロップすると、その従属オブジェクトも同時にドロップされます。ただし、その従属オブジェクトにさらに他のオブジェクトに対する制限 (R) 従属関係があり、それによってその従属オブジェクトを削除できない場合には、基礎となるオブジェクトのドロップは失敗します。
- X** 作動不能 (Inoperative) を意味します。基礎となるオブジェクトをドロップすると、その従属オブジェクトは作動不能になります。ユーザーが何らかの明示的な処置を取るまで、それは作動不能のままになります。
- A** 自動無効化 / 再有効化 (Automatic Invalidation/Revalidation) を意味します。基礎となるオブジェクトをドロップすると、従属オブジェクトは無効になります。データベース・マネージャーは、無効になったオブジェクトを再度有効にしようとします。

関数またはメソッドによって使用されるか、関数またはメソッドによって直接あるいは間接に呼び出されるプロシージャによって使用されるパッケージ

DROP

は、ルーチンが MODIFIES SQL DATA として定義される場合にのみ、自動的に再度有効にされます。ルーチンが MODIFIES SQL DATA でなければ、エラーが戻されます (SQLSTATE 56098)。

DROP ステートメントのパラメーターおよびオブジェクトには、結果的に空白行または列になるため、表 274 に示されていないものもあります。

- EVENT MONITOR、PACKAGE、PROCEDURE、SCHEMA、TYPE MAPPING、および USER MAPPING DROP ステートメントには、オブジェクトの従属関係はありません。
- 別名、バッファ・プール、区分化キー、特権、およびプロシージャのオブジェクト・タイプには、DROP ステートメントの従属関係はありません。
- 指定した作業単位 (UOW) の内側にある A DROP SERVER、DROP FUNCTION MAPPING、または DROP TYPE MAPPING ステートメントは、以下に示すいずれかの条件下で処理することができます。
 - ステートメントが単一のデータ・ソースを参照し、このデータ・ソース内の表またはビューのニックネームを参照する SELECT ステートメントが、UOW にすでに含まれている場合 (SQLSTATE 55006)。
 - ステートメントがデータ・ソースの区分 (たとえば、特定のタイプおよびバージョンのすべてのデータ・ソース) を参照し、こうしたデータ・ソースの 1 つの内側にある表またはビューのニックネームを参照する SELECT ステートメントが、UOW にすでに含まれている場合 (SQLSTATE 55006)。

表 274. 従属関係

	I N D E X E S																T Y P E		U S E R	
オブジェクト・ タイプ →	C O N S T R A I N E R T	F U N C T I O N S T R A I N E R T	C O N S T R A I N E R T	I N D E X E S	M E T H O D S	N I C K N A M E S	O B J E C T S	P R O C E D U R E S	P R O C E D U R E S	R E F E R E N C E S	S E R V E R S	T Y P E S	T Y P E S	U S E R S	U S E R S	U S E R S				
ステートメント ↓	T	N	G	X	N	D	E	P	E ³¹	R	E	E	R	E	G	G	W			
ALTER FUNCTION	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-	-	-			
ALTER METHOD	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-	-	-			
ALTER NICKNAME	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-	-	-			
ALTER PROCEDURE	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-	-	-			
ALTER SERVER	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-	-	-			

表 274. 従属関係 (続き)

オブジェクト・ タイプ →	C O N S T R A I N E R T	F U N C T I O N S	U N D E R P A R T I T I O N S	X E M P L E S	I N D E X E S	M E M B E R S	N I C K N A M E S	N O D E S	P R O C E D U R E S	S E R V E R S	T A B L E S	T E M P O R A R Y T A B L E S	T Y P E S	U S E R S	V I E W S	
ステートメント ↓	T	N	G	X	N	D	E	P	E ³¹	R	E	E	R	E	G	W
ALTER TABLE DROP CONSTRAINT	C	-	-	-	-	-	-	-	A ¹	-	-	-	-	-	-	-
ALTER TABLE DROP PARTITIONING KEY	-	-	-	-	-	-	-	R ²⁰	A ¹	-	-	-	-	-	-	-
ALTER TYPE ADD ATTRIBUTE	-	-	-	-	R	-	-	-	A ²³	-	R ²⁴	-	-	-	-	R ¹⁴
ALTER TYPE ALTER METHOD	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-	-
ALTER TYPE DROP ATTRIBUTE	-	-	-	-	R	-	-	-	A ²³	-	R ²⁴	-	-	-	-	R ¹⁴
ALTER TYPE ADD METHOD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ALTER TYPE DROP METHOD	-	-	-	-	-	R ²⁷	-	-	-	-	-	-	-	-	-	-
CREATE METHOD	-	-	-	-	-	-	-	-	A ²⁸	-	-	-	-	-	-	-
CREATE TYPE	-	-	-	-	-	-	-	-	A ²⁹	-	-	-	-	-	-	-
DROP ALIAS	-	R	-	-	-	-	-	-	A ³	-	R ³	-	X ³	-	-	X ³
DROP BUFFERPOOL	-	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-
DROP DATABASE PARTITION GROUP	-	-	-	-	-	-	-	-	-	-	C	-	-	-	-	-
DROP FUNCTION	R	R ⁷	R	-	R	R ⁷	-	-	X	-	R	-	R	-	-	R
DROP FUNCTION MAPPING	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-	-
DROP INDEX	R	-	-	-	-	-	-	-	A	-	-	-	-	-	-	R ¹⁷
DROP INDEX EXTENSION	-	R	-	R	-	-	-	-	-	-	-	-	-	-	-	-
DROP METHOD	R	R ⁷	R	-	R	R	-	-	X/A ³⁰	-	R	-	R	-	-	R
DROP NICKNAME	-	R	-	C	-	R	-	-	A	-	C ¹¹	-	-	-	-	X ¹⁶

DROP

表 274. 従属関係 (続き)

	C	F	I	N	D	E	N	O	T	Y	U
オブジェクト・ タイプ →	O	U	X	X	N	N	A	B	P	E	S
	N	N	E	E	M	C	E	A	S	E	R
	R	C	P	I	N	E	K	G	C	E	T
	A	T	P	N	S	T	N	R	K	R	A
	I	I	I	D	I	H	A	O	A	V	B
	N	O	N	E	O	O	M	U	G	E	L
ステートメント ↓	T	N	G	X	N	D	E	P	E ³¹	R	E
DROP PROCEDURE	-	R ⁷	-	-	-	R ⁷	-	-	A	-	-
DROP SEQUENCE	-	R	-	-	-	R	-	-	A	-	R
DROP SERVER	-	C ²¹	C ¹⁹	-	-	-	C	-	A	-	C ¹⁹
DROP TABLE	C	R	-	C	-	-	-	-	A ⁹	-	RC ¹¹
DROP TABLE HIERARCHY	C	R	-	C	-	-	-	-	A ⁹	-	RC ¹¹
DROP TABLESPACE	-	-	-	C ⁶	-	-	-	-	-	CR ⁶	-
DROP TRANSFORM	-	R	-	-	-	-	-	X	-	-	-
DROP TRIGGER	-	-	-	-	-	-	-	-	A ¹	-	X ²⁶
DROP TYPE	R ¹³	R ⁵	-	-	R	-	-	-	A ¹²	-	R ¹⁸
DROP VIEW	-	R	-	-	-	-	-	-	A ²	-	X ¹⁶
DROP VIEW HIERARCHY	-	R	-	-	-	-	-	-	A ²	-	X ¹⁶
DROP WRAPPER	-	-	C	-	-	-	-	-	C	-	-
REVOKE 特権 ¹⁰	-	CR ²⁵	-	-	-	CR ²⁵	-	-	A ¹	-	CX ⁸

1 この従属関係は、これらの制約、トリガー、または区分化キーを持つ表に従属することによって、暗黙的に決まります。

2 パッケージに、ビューに影響を与える INSERT、UPDATE、または DELETE ステートメントが含まれている場合、そのパッケージはビューの基本表に対して挿入、更新、または削除の操作を行うこととなります。UPDATE の場合、パッケージは UPDATE によって修正される基本表の各列ごとに更新操作を実行します。

型付きビューに対して操作を行うステートメントがパッケージに含まれている場合、同じビュー階層内でビューを作成したりドロップしたりすると、パッケージが無効になります。

3 パッケージ、マテリアライズ照会表、ステージング表、ビュー、トリガーが別

名を使用する場合、その別名と、その別名が参照するオブジェクトの両方に従属することになります。別名がチェーニングしている場合、そのチェーンの中の別名ごとに従属関係が作成されます。

別名自体は、どのような従属関係も持ちません。存在していないオブジェクトに対しても、別名を定義できます。

- 4 あるユーザー定義タイプ **T** を別のユーザー定義タイプ **B** に従属させるには、**T** が以下の条件を満たしていなければなりません。
 - 属性のデータ・タイプとして **B** を指定している
 - REF(**B**) の属性を持っている
 - スーパータイプとして **B** を持っている
- 5 データ・タイプをドロップすると、その効果がカスケードして、パラメーターや結果タイプとしてそのデータ・タイプを使用する関数やメソッド、そしてそのデータ・タイプで定義されているメソッドもドロップされることとなります。それらの関数やメソッドが互いに依存していても、そのことがそれらの関数やメソッドのドロップを防ぐことにはなりません。ただし、本体でそのデータ・タイプを使用している関数やメソッドに対しては、制約セマンティクスが適用されます。
- 6 表スペースまたは表スペースのリストをドロップすると、指定した表スペース内に完全に含まれているすべての表やリストがドロップされることとなります。ただし、表が複数の表スペース (異なる表スペース内の索引または長形式列) にわたり、そうした表スペースがドロップされるリストにない場合、これらの表スペースは表が存在する限りはドロップできません。
- 7 従属関数が **SOURCE** 文節内の基本関数の名前である場合、その関数は別の特定の関数に従属します。また、従属のルーチンが **SQL** で書かれており、その本体で基本のルーチンを使用する場合も、関数やメソッドは別の特定の関数やメソッドに従属することができます。加えて、構造タイプのパラメーターや戻りタイプをもつ外部のメソッドや関数も、1 つまたは複数のトランスフォーム関数に従属することができます。
- 8 マテリアライズ照会表がドロップされたり、ビューが作動不能になるのは、**SELECT** 特権がない場合だけです。作動不能にされたビューが型付きビュー階層に含まれていれば、そのサブビューもすべて作動不能になります。
- 9 パッケージに、表 **T** に影響を与える **INSERT**、**UPDATE**、または **DELETE** ステートメントが含まれている場合、そのパッケージは **T** に対して挿入、更新、または削除の操作を行うこととなります。 **UPDATE** の場合、パッケージは **UPDATE** によって修正される **T** の各列ごとに更新操作を実行します。

型付き表に対して操作を行うステートメントがパッケージに含まれている場合、同じ表階層内で表を作成したりドロップしたりすると、パッケージが無効になります。

- 10 列に対する特権を個々に取り消すことはできないので、列レベルでの従属関係は存在しません。
- パッケージ、トリガー、またはビューの FROM 文節で OUTER(Z) が使用されている場合、Z のすべての副表またはサブビューで SELECT 特権に対する従属関係が存在します。同じように、パッケージ、トリガー、またはビューで Deref(Y) が使用されていて、Y が Z という表またはビューをターゲットとする参照タイプである場合、Z のすべての副表またはサブビューで SELECT 特権に対する従属関係が存在します。
- 11 マテリアライズ照会表は、基礎表、あるいは表定義の全選択で指定されたニックネームに従属しています。
- カスケードのセマンティクスが、従属するマテリアライズ照会表に適用されません。
- 副表はスーパー表に従属しており、この従属関係はルート表にまで及びます。従属するすべての副表がドロップされるまで、スーパー表はドロップできません。
- 12 TYPE 述部またはサブタイプ処理の式 (TREAT *expression* AS *data-type*) を使用した結果、パッケージは構造タイプに従属することができます。パッケージは、TYPE 述部の右辺、または TREAT 式の右辺で指定した構造タイプすべてと従属関係にあります。構造タイプをドロップしたり作成したりして、パッケージと従属関係にあるサブタイプを変更すると、ステートメントが無効になる場合があります。
- ドロップするタイプのスーパータイプで定義されるメソッドに従属し、オーバーライドに適したパッケージはすべて、無効になります。
- 13 あるタイプがチェック制約またはトリガーで使用されている場合、チェック制約またはトリガーはこのタイプに従属する関係にあります。チェック制約またはトリガーの TYPE 述部で使用される、構造タイプのサブタイプに従属しません。
- 14 あるタイプがビュー定義で使用されている場合、ビューはこのタイプに従属する関係にあります (型付きビューのタイプも含まれます)。ビュー定義内の TYPE 述部で使用される、構造タイプのサブタイプに従属しません。
- 15 サブビューはスーパービューに従属しており、この従属関係はルート・ビューにまで及びます。従属するすべてのサブビューがドロップされるまで、スーパービューはドロップできません。ビューの従属関係の詳細については、¹⁶ を参照してください。
- 16 トリガーまたはビューは参照解除操作または Deref 関数のターゲット表やターゲット・ビューにも従属しています。FROM 文節のトリガーまたはビューで OUTER(Z) を含むものは、トリガーまたはビューが作成された時点で存在した Z の副表またはサブビューすべてに対して従属関係にあります。

- 17 型付きビューはユニーク索引が存在しているかどうかによって依存していることがあります、それによってオブジェクト ID 列が固有なものにすることができます。
- 18 表はユーザー定義データ・タイプ (特殊タイプまたは構造タイプ) に従属している場合があります、それには以下の理由があります。
- そのタイプが列のタイプとして使用されている
 - そのタイプが表のタイプとして使用されている
 - そのタイプが表のタイプの属性として使用されている
 - そのタイプが、表の列タイプまたは表のタイプの属性を表す、参照タイプのターゲット・タイプとして使用されている
 - そのタイプが、表の列のタイプによって直接または間接的に使用されている
- 19 サーバーをドロップすると、カスケード的に、そのネーム・サーバーに作成した関数マッピングとタイプ・マッピングがドロップされます。
- 20 複数パーティションのデータベース・パーティション・グループにある表に対して区分化キーが定義されている場合、この区分化キーは必須です。
- 21 従属している OLE DB 表関数に "R" 従属オブジェクト (DROP FUNCTION を参照) が含まれている場合は、サーバーをドロップできません。
- 22 SQL 関数またはメソッドは、その本体によって参照されるオブジェクトに従属することができます。
- 23 *type-name* T のタイプ TA の属性 A がドロップされると、以下の DROP ステートメントが実際に実行されます。
- ```

Mutator method: DROP METHOD A (TA) FOR T
Observer method: DROP METHOD A () FOR T
ALTER TYPE T
 DROP METHOD A(TA)
 DROP METHOD A()

```
- 24 以下のような場合に、表はユーザー定義による構造データ・タイプの属性に従属することがあります。
1. 表が、*type-name* またはそのサブタイプのいずれかに基づく型付き表である。
  2. 表に、*type-name* を直接または間接的に参照するタイプの列が含まれている。
- 25 定義された関数またはメソッド本体に SELECT 特権がなくなると、SQL 関数の本体またはメソッド本体で使用される表またはビューに対する SELECT 特権の REVOKE により、特権を失った関数またはメソッド本体のドロップが試行されます。これらの関数またはメソッド本体がビュー、トリガー、関数、またはメソッド本体で使用されている場合は、これをドロップすることはできないので、結果として REVOKE が制約されます。それ以外の場合は、REVOKE がカスケードしてそれらの関数はドロップされます。

- 26 トリガーは、INSTEAD OF トリガーが定義されるビューを変更して、INSTEAD OF トリガーが実行される場合、INSTEAD OF トリガーに従属しません。
- 27 他のメソッドによってオーバーライドされた元のメソッドのメソッド宣言は、ドロップすることができません (SQLSTATE -2)。
- 28 作成されるメソッド本体のメソッドが、別のメソッドをオーバーライドするものと宣言される場合、オーバーライドされるメソッド (および、作成されるメソッドのスーパータイプでこのメソッドをオーバーライドするメソッド) に従属したパッケージはすべて無効になります。
- 29 既存のタイプの新しいサブタイプが作成されると、作成されるタイプのスーパータイプで定義されるメソッド (および、オーバーライドに適しているメソッド (たとえば、no mutator や observer)) に従属するパッケージはすべて無効になります。
- 30 ドロップされるメソッド本体の特定メソッドが、別のメソッドをオーバーライドするものと宣言される場合、オーバーライドされるメソッド (および、ドロップされる特定メソッドのスーパータイプでこのメソッドをオーバーライドするメソッド) に従属したパッケージはすべて無効になります。
- 31 キャッシュに入れられた SQL には、パッケージと同じセマンティクスがあります。

## 注:

### • 互換性

- 以前のバージョンの DB2 との互換性:
  - DATABASE PARTITION GROUP の代わりに NODEGROUP を指定できます。
- DB2 UDB (OS/390 および z/OS 版) との互換性:
  - ALIAS の代わりに SYNONYM を指定できます。
  - PACKAGE の代わりに PROGRAM を指定できます。
- ユーザー定義関数を使用中に、そのユーザー定義関数をドロップすることは有効です。また、ユーザー定義関数への参照を含むステートメントでカーソルがオープンされているようにすることができます。そのカーソルがオープンされている間に、カーソルのフェッチがエラーになることなくその関数をドロップすることができます。
- ユーザー定義関数に従属しているパッケージが実行されている場合、そのパッケージが現行の作業単位を完了するまで、別の許可 ID からその関数をドロップすることはできません。その時点で、関数はドロップされ、パッケージは作動不能になります。このパッケージの次の要求はエラーになり、パッケージの明示再バインドが必要であることが示されます。
- 関数本体を必要とするアプリケーションが実行されている時に、関数本体がドロップされる場合があります (これは関数のドロップとは違うことです)。ステートメントの

代わりにデータベース・マネージャーが関数本体をストレージにロードする必要があるかどうかに応じて、ステートメントはエラーになる場合もあれば、エラーにならない場合もあります。

- ドロップされた表の中に **DATALINK** 列によってリンクされているファイルが含まれている場合には、それらのファイルはリンク解除されてから、データ・リンク列の定義に応じてリストアされたり削除されたりします。
- データベースに対して構成された **DB2 Data Links Manager** を使用できないときに、**DROP TABLE** または **DROP TABLESPACE** を使って **DATALINK** 列を含んだ表がドロップされると、操作は失敗します (SQLSTATE 57050)。
- 明示的に指定された **UDF** に記録されている従属関係に加えて、トランスフォームが暗黙的に必要な場合には以下の従属関係が記録されます。
  1. 構造タイプのパラメーターや関数またはメソッドの結果にトランスフォームが必要な場合は、その関数またはメソッドに、**TO SQL** か **FROM SQL** の必要なトランスフォーム関数に対する従属関係が記録されます。
  2. パッケージに含まれている **SQL** ステートメントでトランスフォーム関数が必要になる場合は、そのパッケージに、**TO SQL** か **FROM SQL** の指定されたトランスフォーム関数に対する従属関係が記録されます。

上記の部分では、トランスフォームを暗黙的に呼び出すことによって従属関係が記録される場合のみを扱っているため、関数、メソッド、あるいはパッケージ以外のオブジェクトが、暗黙的に呼び出されたトランスフォーム関数に従属することはありません。一方、トランスフォーム関数を明示的に呼び出した場合 (たとえば、ビューやトリガーなどで) は、これらの他のタイプのオブジェクトが通常どおりトランスフォーム関数に従属します。したがって、トランスフォームに対するこれらの「明示的な」タイプの従属がドロップされることによって、**DROP TRANSFORM** が失敗する場合があります (SQLSTATE 42893)。

- 従属関係カタログでは、暗黙的なトランスフォームによる関数への従属と明示的に関数を呼び出すことによって生じる従属とを区別していません。したがって、トランスフォーム関数に対する明示的な呼び出しは書かないよう勧められています。このようなインスタンスでは、単に **SQL** の式に明示的な呼び出しが含まれているという理由で、関数上のトランスフォーム・プロパティーがドロップされなかったり、パッケージが作動不能としてマークされてしまいます。
- **IDENTITY** 列のシーケンスを作成したシステムを、**DROP SEQUENCE** ステートメントでドロップすることはできません。
- シーケンスがドロップされると、シーケンスに関する特権もすべてドロップされ、そのシーケンスを参照するパッケージはすべて無効になります。

例:

例 1: 表 **TDEPT** をドロップします。

```
DROP TABLE TDEPT
```

## DROP

例 2: ビュー VDEPT をドロップします。

```
DROP VIEW VDEPT
```

例 3: 許可 ID HEDGES が別名をドロップします。

```
DROP ALIAS A1
```

別名 HEDGES.A1 がカタログから除去されます。

例 4: Hedges は別名のドロップを試みますが、既存の表の名前である (別名でない) T1 を別名として指定しています。

```
DROP ALIAS T1
```

このステートメントはエラーになります (SQLSTATE 42809)。

例 5:

BUSINESS\_OPS データベース・パーティション・グループをドロップします。このデータベース・パーティション・グループをドロップするには、まずデータベース・パーティション・グループ内の表スペース (ACCOUNTING と PLANS) をドロップする必要があります。

```
DROP TABLESPACE ACCOUNTING
DROP TABLESPACE PLANS
DROP DATABASE PARTITION GROUP BUSINESS_OPS
```

例 6: Pellow は CENTRE 関数をドロップします。この関数は、ドロップする関数インスタンスであることを示すためにシグニチャーを使用して、PELLOW スキーマに作成したものです。

```
DROP FUNCTION CENTRE (INT,FLOAT)
```

例 7: McBride は FOCUS92 関数をドロップします。この関数は、ドロップする関数インスタンスであることを示すために特定名を使用して、PELLOW スキーマに作成したものです。

```
DROP SPECIFIC FUNCTION PELLOW.FOCUS92
```

例 8: CHEM スキーマから関数 ATOMIC\_WEIGHT をドロップします。このスキーマには、この名前関数は 1 つしかないことが分かっています。

```
DROP FUNCTION CHEM.ATOMIC_WEIGHT
```

例 9: トリガー SALARY\_BONUS をドロップします。このトリガーにより、従業員は指定の条件で給与に加えてボーナスを受け取ります。

```
DROP TRIGGER SALARY_BONUS
```

例 10: 現在使用していない SHOESIZE という名前特殊データ・タイプをドロップします。

**DROP DISTINCT TYPE SHOESIZE**

例 11: SMITHPAY イベント・モニターをドロップします。

**DROP EVENT MONITOR SMITHPAY**

例 12: CREATE SCHEMA の例 2 で RESTRICT を使用して作成したスキーマをドロップします。PART という名前の表をまず削除する必要があることに注意してください。

**DROP TABLE PART  
DROP SCHEMA INVENTORY RESTRICT**

例 13: Macdonald は DESTROY プロシージャをドロップします。このプロシージャは、ドロップするプロシージャ・インスタンスであることを示すために特定名を使用して、EIGLER スキーマに作成したものです。

**DROP SPECIFIC PROCEDURE EIGLER.DESTROY**

例 14: BIOLOGY スキーマからプロシージャ OSMOSIS をドロップします。このスキーマには、この名前プロシージャは 1 つしかないことが分かっています。

**DROP PROCEDURE BIOLOGY.OSMOSIS**

例 15: ユーザー SHAWN は、連合データベースにアクセスするとき、ORACLE1 という Oracle データ・ソースのデータベースにアクセスするときでは、異なる許可 ID を使用しました。2 つの許可でマッピングが作成されましたが、SHAWN がそのデータ・ソースにアクセスする必要はなくなりました。マッピングをドロップします。

**DROP USER MAPPING FOR SHAWN SERVER ORACLE1**

例 16: ニックネームが参照するデータ・ソース表の索引が削除されました。オプティマイザーにこの索引を認識させるために作成した索引指定をドロップします。

**DROP INDEX INDEXSPEC**

例 17: トランスフォーム・グループ MYSTRUCT1 をドロップします。

**DROP TRANSFORM MYSTRUCT1 FOR POLYGON**

例 18: PERSONNEL スキーマで EMP データ・タイプからメソッド BONUS をドロップします。

**DROP METHOD BONUS (SALARY DECIMAL(10,2)) FOR PERSONNEL.EMP**

例 19: 制限を使用して ORG\_SEQ からシーケンスをドロップします。

**DROP SEQUENCE ORG\_SEQ**

例 20: リモート表 EMPLOYEE が、連合システムに透過性 DDL を使用して作成されました。この表へのアクセスは、もう必要ありません。リモート表 EMPLOYEE をドロップします。

## DROP

### DROP TABLE EMPLOYEE

#### 関連タスク:

- 450 ページの『透過 DDL を使用したリモート表の作成および変更』

#### 関連資料:

- *SQL* リファレンス 第 2 巻 の『CREATE TRIGGER ステートメント』
- 1469 ページの『CREATE VIEW』
- *SQL* リファレンス 第 2 巻 の『CREATE FUNCTION MAPPING ステートメント』

#### 関連サンプル:

- 『dbstat.sqb -- Reorganize table and run statistics (MF COBOL)』
- 『dtstruct.sqC -- Create, use, drop a hierarchy of structured types and typed tables (C++)』
- 『tbconstr.sqC -- How to create, use, and drop constraints (C++)』
- 『tbcreate.sqC -- How to create and drop tables (C++)』
- 『tbtrig.sqC -- How to use a trigger on a table (C++)』
- 『DbSeq.java -- How to create, alter and drop a sequence in a database (JDBC)』
- 『TbConstr.java -- How to create, use and drop constraints (JDBC)』
- 『TbCreate.java -- How to create and drop tables (JDBC)』
- 『TbTemp.java -- How to use Declared Temporary Table (JDBC)』
- 『TbTrig.java -- How to use triggers (JDBC)』
- 『UDFDrop.db2 -- How to uncatalog the Java UDFs contained in UDFsrv.java 』
- 『spdrop.db2 -- How to uncatalog the stored procedures contained in spserver.sqc 』
- 『tbconstr.sqc -- How to create, use, and drop constraints (C)』
- 『tbcreate.sqc -- How to create and drop tables (C)』
- 『tbtemp.sqc -- How to use a declared temporary table (C)』
- 『tbtrig.sqc -- How to use a trigger on a table (C)』
- 『TbConstr.sqlj -- How to create, use and drop constraints (SQLj)』
- 『TbCreate.sqlj -- How to create and drop tables (SQLj)』
- 『TbTrig.sqlj -- How to use triggers (SQLj)』

## EXECUTE

EXECUTE ステートメントは、準備済み SQL ステートメントを実行します。

#### 呼び出し:

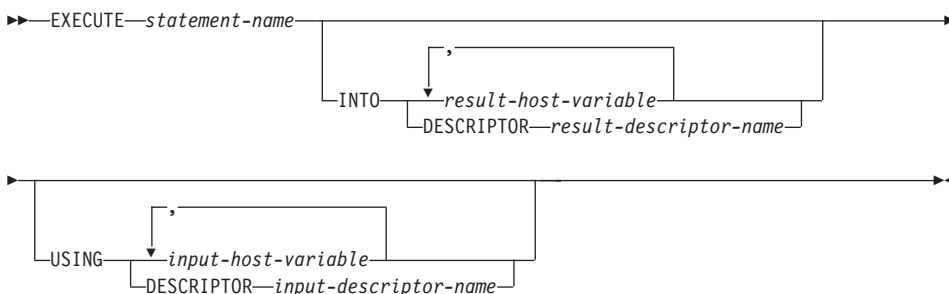
このステートメントは、アプリケーション・プログラムに組み込む方法でのみ使用可能です。これは、動的に作成できない実行可能ステートメントです。

#### 許可:

ステートメントの実行時に許可検査が行われるステートメント (DDL、GRANT、および REVOKE ステートメント) の場合、このステートメントの許可 ID の特権には、PREPARE ステートメントで指定されている SQL ステートメントを実行するための特権が含まれていなければなりません。ステートメントの許可 ID は、バインド・オプション DYNAMICRULES の影響を受けることがあります。

許可検査がステートメントの準備の時点で行われるステートメント (DML) の場合、このステートメントを使用するために必要な権限はありません。

#### 構文:



#### 説明:

##### *statement-name*

実行する準備済みのステートメントを指定します。 *statement-name* (ステートメント名) はすでに準備済みのステートメントを指定していなければならず、またそのステートメントが SELECT ステートメントであってはなりません。

##### INTO

この後に、準備済みステートメントの出力パラメーター・マーカ (? ) から値を受け取るために使用される、ホスト変数のリストを指定します。

動的 CALL ステートメントの場合は、ストアード・プロシージャに対する OUT および INOUT 引き数に使用されるパラメーター・マーカは、出力パラメーター・マーカです。ステートメントに出力パラメーター・マーカを使用する場合は、 INTO 文節を指定する必要があります (SQLSTATE 07007)。

##### *result-host-variable, ...*

ホスト変数の宣言規則にしたがって、該当プログラムで宣言されているホスト変数を指定します。変数の数は、準備されるステートメントの出力パラメーター・マーカの数と同じでなければなりません。 *n* 番目の変数は、準備済みス

## EXECUTE

ステートメントの  $n$  番目のパラメーター・マーカーに対応します。場合によっては、ロケーター変数とファイル参照変数も、パラメーター・マーカーの宛先として指定できます。

### DESCRIPTOR *result-descriptor-name*

出力 SQLDA を指定します。その内容は、ホスト変数についての有効な記述でなければなりません。

EXECUTE ステートメントが処理される前に、ユーザーは、入力 SQLDA の以下のフィールドを設定する必要があります。

- SQLDA に用意する SQLVAR のエレメント数を示す SQLN
- SQLDA に割り振るストレージのバイト数を示す SQLDABC
- ステートメントの処理時にその SQLDA の使用される変数の数を示す SQLD
- 変数の属性を示す SQLVAR のオカレンス

SQLDA には、すべての SQLVAR オカレンスが入るだけの十分なストレージがなければなりません。

LOB または構造化データ・タイプの実出力データを入れる必要がある場合には、各パラメーター・マーカーごとに 2 つの SQLVAR 項目が必要になります。

SQLD に設定する値は、ゼロ以上で SQLN 以下でなければなりません。

## USING

この後に、準備済みステートメントの入力パラメーター・マーカー (?) に置き換わる値を含むホスト変数のリストを指定します。

動的 CALL ステートメントの場合、ストアード・プロシージャーに対する IN および INOUT 引き数に使用されるパラメーター・マーカーは、入力パラメーター・マーカーです。その他のすべてのステートメントの場合、すべてのパラメーター・マーカーは入力パラメーター・マーカーです。ステートメントに出力パラメーター・マーカーを使用する場合は、USING 文節を指定する必要があります (SQLSTATE 07004)。

### *input-host-variable, ...*

ホスト変数の宣言規則にしたがって、該当プログラムで宣言されているホスト変数を指定します。変数の数は、準備されるステートメントの入力パラメーター・マーカーの数と同じでなければなりません。  $n$  番目の変数は、準備済みステートメントの  $n$  番目のパラメーター・マーカーに対応します。場合によっては、ロケーター変数とファイル参照変数も、パラメーター・マーカーの値のソースとして指定できます。

### DESCRIPTOR *input-descriptor-name*

入力 SQLDA を指定します。その内容は、ホスト変数についての有効な記述でなければなりません。

EXECUTE ステートメントが処理される前に、ユーザーは、入力 SQLDA の以下のフィールドを設定する必要があります。



- SQLDA に用意する SQLVAR のエレメント数を示す SQLN
- SQLDA に割り振るストレージのバイト数を示す SQLDABC
- ステートメントの処理時にその SQLDA の使用される変数の数を示す SQLD
- 変数の属性を示す SQLVAR のオカレンス

SQLDA には、すべての SQLVAR オカレンスが入るだけの十分なストレージがなければなりません。したがって、SQLDABC の値は  $16 + \text{SQLN} * (N)$  以上でなければなりません (N は 1 つの SQLVAR オカレンスの長さ)。

LOB または構造化データ・タイプの入力データを入れる必要がある場合には、各パラメーター・マーカごとに 2 つの SQLVAR 項目が必要になります。

SQLD に設定する値は、ゼロ以上で SQLN 以下でなければなりません。

#### 注:

- 準備済みステートメントを実行する前に、各入力パラメーター・マーカはそれに対応するホスト変数の値によって置き換えられます。型付きパラメーター・マーカの場合、ターゲット変数の属性は CAST 指定によって指定されます。タイプなしパラメーター・マーカの場合、ターゲット変数の属性はパラメーター・マーカのコンテキストにしたがって決定されます。

V は、パラメーター・マーカ P に対応するホスト変数を表します。V の値は、列への値の割り振り規則にしたがって、P のターゲット変数に割り当てられます。したがって、

- V はターゲットと互換でなければなりません。
- V がストリングの場合、その長さはターゲットの長さ属性を超えることはできません。
- V が数値の場合、V の整数部分の絶対値はターゲットの整数部分の絶対値の最大を超えることはできません。
- V の属性がターゲットの属性と同一でない場合、その値はターゲットの属性に合うように変換されます。

準備済みステートメントを実行すると、P の代わりに使用される値は P のターゲット変数になります。たとえば、V が CHAR(6) でターゲットが CHAR(8) の場合、P の代わりに使用される値は V の値にブランクを 2 個付加したものになります。

- 動的 CALL ステートメントの場合は、準備済みステートメントの実行後は、OUT および INOUT の各引き数の戻り値は、引き数に使用された出力パラメーター・マーカに対応するホスト変数に割り当てられます。型付きパラメーター・マーカの場合、ターゲット変数の属性は CAST 指定によって指定されます。タイプなしパラメーター・マーカの場合、ターゲット変数の属性は、ストアード・プロシージャのパラメーターの定義によって指定されます。

V は、パラメーター・マーカ P に対応する出力ホスト変数を表し、ストアード・プロシージャの引き数 A に使用されます。A の値は、列から値を検索するための規則にしたがって V に割り当てられます。したがって、

## EXECUTE

- V は A と互換でなければなりません。
  - V がストリングの場合、その長さは A の長さより短いものであってはなりません。そうでないと A の値は切り捨てられます。
  - V が数値の場合、V の整数部分の最大絶対値は A の整数部分の絶対値より小さいものであってはなりません。
  - V の属性が A の属性と同一でない場合、A の値は V の属性に合うように変換されます。
- **動的 SQL ステートメント・キャッシング:** 動的および静的 SQL ステートメントの実行に必要な情報は、静的 SQL ステートメントが最初に参照された時点、または動的 SQL ステートメントが最初に準備された時点で、データベース・パッケージ・キャッシングに入れます。この情報は、無効になるか、キャッシュ・スペースが他のステートメントで必要になるか、またはデータベースがシャットダウンされるまでは、パッケージ・キャッシュに存続します。

SQL ステートメントが実行または準備される場合に、要求を出したアプリケーションに関連するパッケージ情報が、システム・カタログからパッケージ・キャッシュにロードされます。個々の SQL ステートメントの実際の実行可能セクションもキャッシュに入れます。静的 SQL セクションは、該当のステートメントが最初に参照された時点で、システム・カタログから読み取られてパッケージ・キャッシュに入れられ、動的 SQL セクションは作成後にキャッシュに直接入れられます。動的 SQL セクションは、PREPARE や EXECUTE IMMEDIATE などの明示的なステートメントによって作成されます。一度作成された動的 SQL ステートメントのセクションが、スペース管理のために削除された場合や、環境の変化によって無効になった場合に、システムによるステートメントの暗黙的な準備によって、再作成されることがあります。

各 SQL ステートメントは、データベース・レベルでキャッシュされ、アプリケーション間で共有できます。静的 SQL ステートメントは、同じパッケージを使用してアプリケーション間で共有されます。動的 SQL ステートメントは、同じコンパイル環境と、厳密に同じステートメント・テキストを使用してアプリケーション間で共有されます。アプリケーションによって発行される各 SQL ステートメントのテキストは、アプリケーションにローカルにキャッシュされ、暗黙的な準備が必要な場合に使用されます。アプリケーション・プログラム中の各 PREPARE ステートメントは、1 つのステートメントをキャッシュできます。アプリケーション・プログラム中のすべての EXECUTE IMMEDIATE ステートメントは、同じスペースを共用し、これらの EXECUTE IMMEDIATE ステートメントに対しては、キャッシュされるステートメントは同時に 1 つしか存在しません。それぞれ異なる SQL ステートメントに対して、同じ PREPARE またはいずれかの EXECUTE IMMEDIATE ステートメントが何度も発行される場合は、最後のステートメントだけがキャッシュに入れられ、再使用の対象になります。キャッシュの使用を最適化するには、アプリケーションの開始時に多くの異なる PREPARE ステートメントを一度に発行し、その後必要に応じて EXECUTE または OPEN ステートメントを発行することです。

動的 SQL ステートメントのキャッシングを使用すると、ステートメントを一度作成すれば、ステートメントを再度準備しなくても複数の作業単位にわたってステートメントを再使用できます。環境が変わった場合には、必要に応じてシステムはステートメントを再コンパイルします。

以下のイベントは、次の PREPARE、EXECUTE、EXECUTE IMMEDIATE、または OPEN の要求時に、キャッシュされた動的ステートメントが暗黙的に準備される原因となる環境またはデータ・オブジェクトの変更の例です。

- ALTER FUNCTION
- ALTER METHOD
- ALTER NICKNAME
- ALTER PROCEDURE
- ALTER SERVER
- ALTER TABLE
- ALTER TABLESPACE
- ALTER TYPE
- CREATE FUNCTION
- CREATE FUNCTION MAPPING
- CREATE INDEX
- CREATE METHOD
- CREATE PROCEDURE
- CREATE TABLE
- CREATE TEMPORARY TABLESPACE
- CREATE TRIGGER
- CREATE TYPE
- DROP (すべてのオブジェクト)
- 表または索引の RUNSTATS
- ビューが作動不能になる原因となるすべてのアクション
- システム・カタログ表の統計の UPDATE
- SET CURRENT DEGREE
- SET PATH
- SET QUERY OPTIMIZATION
- SET SCHEMA
- SET SERVER OPTION

キャッシュに入れられる動的 SQL ステートメントから予想される動作の概略は、以下ようになります。

## EXECUTE

- *PREPARE* 要求: 以後同じステートメントの準備に、セクションが有効であればステートメントのコンパイルのコストがかかりません。現在キャッシュに入れられているセクションのコストとカーディナリティーの見積もりが戻されます。それらの値は、同じ SQL ステートメントに対するそれより前の *PREPARE* から戻される値とは違っている場合があります。 *COMMIT* または *ROLLBACK* ステートメントの後に *PREPARE* ステートメントを発行する必要はありません。
- *EXECUTE* 要求: 元の *PREPARE* 以後にステートメントが無効になった場合に、ステートメントを暗黙的に準備するコストが *EXECUTE* ステートメントにかかることがあります。セクションが暗黙的に準備される場合、当初の *PREPARE* ステートメントの環境でなく、現行の環境が使用されます。
- *EXECUTE IMMEDIATE* 要求: 以後同じステートメントに対して *EXECUTE IMMEDIATE* ステートメントを出す際に、セクションが有効であればステートメントのコンパイルのコストがかかりません。
- *OPEN* 要求: 当初の *PREPARE* ステートメント以後にステートメントが無効になった場合、ステートメントを暗黙的に準備するコストが動的に定義されたカーソルに対する *OPEN* 要求にかかることがあります。セクションが暗黙的に準備される場合、当初の *PREPARE* ステートメントの環境でなく、現行の環境が使用されます。
- *FETCH* 要求: 予想される動作の変化はありません。
- *ROLLBACK*: ロールバック操作の影響を受ける作業単位で準備されたか暗黙的に準備された動的 SQL ステートメントだけが無効になります。
- *COMMIT*: 動的 SQL ステートメントは無効になりませんが、確立されたロックは解放されます。 *WITH HOLD* カーソルとして定義されていないカーソルはクローズされ、そのロックは解放されます。オープンされている *WITH HOLD* カーソルは、そのパッケージとセクション・ロックを保持し、コミット処理中およびコミット処理後にアクティブのセクションを保護します。

暗黙の準備の過程でエラーが生じると、その暗黙の準備の原因となった要求にエラーが戻されます (SQLSTATE 56098)。

### 例:

例 1: この C の例では、パラメーター・マーカを伴う *INSERT* ステートメントが準備され、実行されます。ホスト変数 h1 - h4 は、TDEPT の形式に対応します。

```
strcpy (s,"INSERT INTO TDEPT VALUES(?,?,?,?)");
EXEC SQL PREPARE DEPT_INSERT FROM :s;
.
.
.
.
.
.
.
EXEC SQL EXECUTE DEPT_INSERT USING :h1, :h2,
:h3, :h4;
```

例 2: この *EXECUTE* ステートメントは *SQLDA* を使用します。

**EXECUTE S3 USING DESCRIPTOR :sqlda3**

例 3: 従業員に賞与を与えるための以下のストアード・プロシージャを考慮します。

```
CREATE PROCEDURE GIVE_BONUS (IN EMPNO INTEGER,
 IN DEPTNO INTEGER,
 OUT CHEQUE INTEGER,
 INOUT BONUS DEC(6,0))
```

...

ストアード・プロシージャを C アプリケーションから動的に呼び出します。ストアード・プロシージャは、以下のホスト変数を入力として取ります。

- *employee*。従業員の ID 番号。
- *dept*。部門番号。
- *bonus*。従業員の賞与。

ストアード・プロシージャは、以下の値をホスト変数に戻します。

- *cheque\_no*。小切手の ID 番号。
- *bonus*。実際の賞与額 (調整後の)

```
strcpy (s, "CALL GIVE_BONUS(?, ?, ?, ?)");
EXEC SQL PREPARE DO_BONUS FROM :s;
.
.
/* Check for successful execution and put values into
 :employee, :dept, and :bonus */
.
.
EXEC SQL EXECUTE DO_BONUS INTO :cheque_no, :bonus
 USING :employee, :dept, :bonus;
.
.
/* Check for successful execution and process the
 values returned in :cheque_no and :bonus */
```

**関連資料:**

- *SQL* リファレンス 第 1 巻 の『ID』
- *SQL* リファレンス 第 2 巻 の『PREPARE ステートメント』
- 1322 ページの『SQLDA (SQL 記述子域)』

**関連サンプル:**

- 『dbuse.sqc -- How to use a database (C)』
- 『fnuse.sqc -- How to use built-in SQL functions (C)』
- 『tut\_use.sqc -- How to modify a database (C)』
- 『udfcli.sqc -- Call a variety of types of user-defined functions (C)』
- 『dbuse.sqC -- How to use a database (C++)』
- 『fnuse.sqC -- How to use built-in SQL functions (C++)』

## EXECUTE

- 『tut\_use.sqlC -- How to modify a database (C++)』
- 『udfcli.sqlC -- Call a variety of types of user-defined functions (C++)』
- 『inpsrv.sqlb -- Demonstrates stored procedures using the SQLDA structure (MF COBOL)』

## DECLARE GLOBAL TEMPORARY TABLE

DECLARE GLOBAL TEMPORARY TABLE ステートメントは、現行セッションの一時表を定義します。宣言済み一時表の記述は、システム・カタログには現れません。これは永続的なものではなく、他のセッションと共用することもできません。同じ名前の宣言されたグローバル一時表を定義するセッションであっても、一時表の記述はそのセッションによって異なります。セッションが終了すると、表の行は削除され、一時表の記述はドロップされます。

### 呼び出し:

このステートメントはアプリケーション・プログラムに組み込むことができ、また動的 SQL ステートメントを使用して出すことができます。このステートメントは、動的に作成できる実行可能ステートメントです。

### 許可:

ステートメントの許可 ID によって保持されている特権には、少なくとも以下のいずれかが含まれていなければなりません。

- SYSADM または DBADM 権限
- USER TEMPORARY 表スペースでの USE 権限

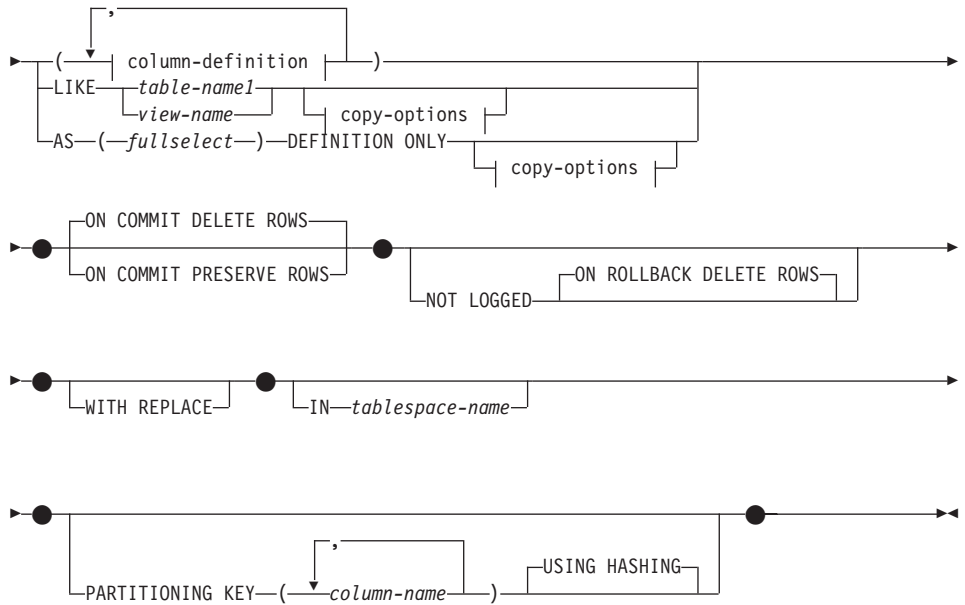
LIKE または全選択を使用して表を定義する場合、ステートメントの許可 ID の特権に、識別されているそれぞれの表またはビューに対する以下の権限が少なくとも 1 つ以上含まれている必要があります。

- その表またはビューに対する SELECT 特権
- 表またはビューに対する CONTROL 特権
- SYSADM または DBADM 権限

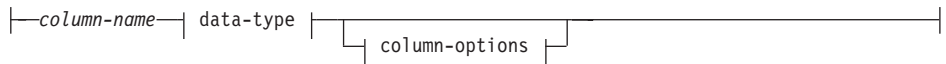
### 構文:

▶—DECLARE GLOBAL TEMPORARY TABLE—*table-name*—————▶

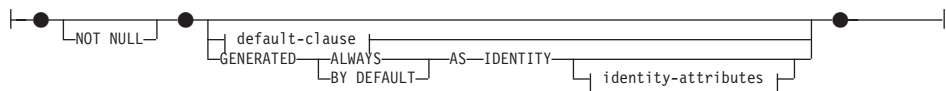
## DECLARE GLOBAL TEMPORARY TABLE



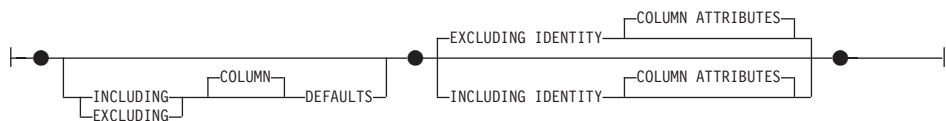
### column-definition:



### column-options:



### copy-options:



### 説明:

#### table-name

一時表の名前を示します。修飾子を明示的に指定する場合は、`SESSION` でなければなりません。そうしないと、エラーになります (SQLSTATE 428EK)。修飾子が指定されなければ、暗黙的に `SESSION` が指定されます。

同じ `table-name` の宣言されたグローバル一時表を定義するセッションであっても、その宣言されたグローバル一時表の記述はそれぞれのセッションによって異なります。

## DECLARE GLOBAL TEMPORARY TABLE

す。 *table-name* を使用する宣言済み一時表がセッション内にすでに存在している場合は、 `WITH REPLACE` 文節を指定する必要があります (SQLSTATE 42710)。

表、ビュー、別名、またはニックネームについては、同じ名前および同じスキーマ名 (SESSION) を持つものがカタログ内にすでに存在していても構いません。このような場合には、以下のような処理が行われます。

- 宣言されているグローバル一時表 *table-name* は、正常に定義されます (エラーや警告は戻されません)。
- `SESSION.table-name` への参照はすべて、カタログ内ですでに定義されている `SESSION.table-name` ではなく、宣言されたグローバル一時表に対して行われます。

### column-definition

一時表の列の属性を定義します。

### column-name

表を構成する列の名前を指定します。名前を修飾したり、表の複数の列に対して同じ名前を使用することはできません (SQLSTATE 42711)。

表には、以下のものを指定できます。

- 4K ページ・サイズの場合、最大 500 列。列のバイト・カウントは 4 005 を超えてはなりません。
- 8K ページ・サイズの場合、最大 1 012 列。列のバイト・カウントは 8 101 を超えてはなりません。
- 16K ページ・サイズの場合、最大 1 012 列。列のバイト・カウントは 16 293 を超えてはなりません。
- 32K ページ・サイズの場合、最大 1 012 列。列のバイト・カウントは 32 677 を超えてはなりません。

詳細については、『CREATE TABLE』の『行サイズ』を参照してください。

### data-type

許容されるタイプについては、『CREATE TABLE』の *data-type* を参照してください。宣言されたグローバル一時表では、BLOB、CLOB、DBCLOB、LONG VARCHAR、LONG VARGRAPHIC、DATALINK、参照、および構造タイプを使用できませんのでご注意ください (SQLSTATE 42962)。なお、この例外として、これらの制限されたタイプをソースとする特殊タイプがあります。

FOR BIT DATA は、文字ストリング・データ・タイプの一部として指定することができます。

### column-options

表の列に関連した追加オプションを定義します。



## NOT NULL

列に NULL 値が入るのを防止します。NULL 値の指定については、『CREATE TABLE』の NOT NULL を参照してください。

## default-clause

デフォルトの指定については、『CREATE TABLE』の *default-clause* を参照してください。

## IDENTITY および identity-attributes

ID 列の指定については、『CREATE TABLE』の IDENTITY および *identity-attributes* を参照してください。

## LIKE *table-name1* または *view-name*

表の列の名前と記述が、指定された表 (*table-name1*)、ビュー (*view-name*)、またはニックネーム (*nickname*) の列とまったく同じであることを示します。LIKE の後に指定する名前は、カタログに存在している表、ビュー、またはニックネーム、あるいは宣言済み一時表を識別するものでなければなりません。型付き表または型付きビューを指定することはできません (SQLSTATE 428EC)。

LIKE は、*n* 列の暗黙的な定義で使用します。*n* は、指定した表またはビューに含まれる列の数を表します。

- 表を指定すると、暗黙的な定義には *table-name1* のおのおのの列の列名、データ・タイプ、および NULL 可能特性が入ります。EXCLUDING COLUMN DEFAULTS を指定しないと、列のデフォルト値も入ります。
- ビューを指定すると、暗黙的な定義には *view-name* に指定した全選択のおおのの結果列の列名、データ・タイプ、および NULL 可能特性が入ります。

*copy-attributes* 文節に基づいて、列のデフォルトと ID 列属性を組み込んだり除外したりすることができます。

指定した表やビューのこの他の属性は、暗黙的な定義には含まれません。したがって、新しい表にはユニーク制約、外部キー制約、トリガー、または索引はありません。表は、IN 文節の指定にしたがって、表スペースの中に明示的または暗黙的に作成されます。

*table-name1* や *view-name* には、作成するグローバル一時表と同じ名前を使用することはできません (SQLSTATE 428EC)。

## AS (*fullselect*) DEFINITION ONLY

表定義が、照会式の結果による列定義に基づいていることを示します。AS (*fullselect*) は、宣言されたグローバル一時表に対する *n* 列の暗黙的な定義で使用されます。*n* は、*fullselect* の結果として得られる列の数を表します。新しい表の列は、これらの *fullselect* で得られた列に基づいて定義されます。選択リストの各エレメントの名前は、それぞれ固有なものでなければなりません (SQLSTATE 42711)。SELECT 文節で AS 文節を使用すると、それぞれのエレメントに固有の名前を付けることができます。

## DECLARE GLOBAL TEMPORARY TABLE

暗黙的な定義では、*fullselect* の各結果列について、その列名、データ・タイプ、および NULL 可能特性を定義します。

### copy-options

これらのオプションでは、ソースの結果表定義 (表、ビュー、または全選択) から付加的な属性をコピーするかどうかを指定します。

### INCLUDING COLUMN DEFAULTS

ソース結果表の定義の更新可能な各列の列デフォルトをコピーします。更新可能でない列では、作成される表の対応列にデフォルトが定義されないことになります。

LIKE *table-name1* が指定されており、かつ *table-name1* が基本表か宣言済み一時表である場合に限り、この INCLUDING COLUMN DEFAULTS がデフォルトとして使用されます。

### EXCLUDING COLUMN DEFAULTS

列のデフォルトは、ソースの結果表定義からコピーされません。

この文節がデフォルトです。ただし、LIKE *table-name* が指定され、かつ *table-name* が基本表または宣言済み一時表を示す場合を除きます。

### INCLUDING IDENTITY COLUMN ATTRIBUTES

この文節を使用すると、ソースの結果表定義から ID 列の属性 (START WITH、INCREMENT BY、および CACHE の値) がコピーされます。これらの属性をコピーできるのは、表、ビュー、または全選択内の対応する列の要素が、ID のプロパティが含まれている基本表の列名に直接または間接的にマップされた表の列の名前、またはビューの列の名前である場合です。これ以外の場合は、新しい一時表の列に ID のプロパティは定義されません。たとえば:

- 全選択の選択リストに ID 列の名前のインスタンスが複数含まれている (つまり、同じ列を複数回選択している) 場合
- 全選択の選択リストに複数の ID 列が含まれている (つまり、結合が関与している) 場合
- ID 列が選択リスト内の式に組み込まれている場合
- 全選択にセット演算 (UNION (合併)、EXCEPT (差)、または INTERSECT (論理積)) が含まれている場合

### EXCLUDING IDENTITY COLUMN ATTRIBUTES

ソース結果表の定義から ID 列属性はコピーされません。

### ON COMMIT

COMMIT 操作の実行時にグローバル一時表で行うアクションを指定します。

### DELETE ROWS

表にオープンされている WITH HOLD カーソルがなければ、すべての行が表から削除されます。これがデフォルトです。

## PRESERVE ROWS

表の行が保存されます。

## NOT LOGGED

表の作成を含め、表に対して行われた変更は記録されません。ROLLBACK (または ROLLBACK TO SAVEPOINT) 操作が実行されると、作業単位 (または保管点) で表を作成していた場合には、その表はドロップされます。そして作業単位 (または保管点) で表をドロップしていた場合には、表がリストアされますが、行はリストアされません。

## ON ROLLBACK DELETE ROWS

ROLLBACK (または ROLLBACK TO SAVEPOINT) 操作が実行されるときに、記録されていないグローバル一時表に対して取られるアクションを指定します。表データが変更されている場合は、すべての行が削除されます。

## WITH REPLACE

ユーザーが指定した名前を持つグローバル一時表がすでに存在している場合は、既存の表をこのステートメントで定義した一時表で置き換える (および既存の表の行をすべて削除する) よう指示します。

WITH REPLACE が指定されていない場合は、現行セッションにすでに存在しているグローバル一時表の名前を指定することはできません (SQLSTATE 42710)。

## IN *tablespace-name*

グローバル一時表をインスタンス化する表スペースを指定します。ここでは、既存の USER TEMPORARY 表スペースを指定する必要があります (SQLSTATE 42838)。また、ステートメントの許可 ID にはその表スペースに対する USE 特権が含まれていなければなりません (SQLSTATE 42501)。この文節が指定されない場合、表をインスタンス化する表スペースは USER TEMPORARY 表スペースの中から選択され、その中のステートメントの許可 ID に USE 特権が含まれている表スペースで、かつ必要なページ・サイズに最も適したサイズの表スペースが使用されます。複数の表スペースがそれにあてはまる場合、以下のどれに USE 特権が付与されているかに応じて優先順位が決められます。

1. 許可 ID
2. 許可 ID を保有するグループ
3. PUBLIC

それでも複数の表スペースがそれにあてはまる場合は、最終選択はデータベース・マネージャーによって行われます。条件に合う USER TEMPORARY がない場合はエラーが戻されます (SQLSTATE 42727)。

表スペースの決定は、以下の時点で変更することができます。

- 表スペースをドロップまたは作成するとき
- USE 特権を付与または取り消すとき

## DECLARE GLOBAL TEMPORARY TABLE

十分な表のページ・サイズは、行のバイト・カウントか列の数のいずれかによって決まります。詳細については、『CREATE TABLE』の『行サイズ』を参照してください。

### PARTITIONING KEY (*column-name,...*)

表のデータが区分化されている場合に、区分化キーを指定します。各 *column-name* (列名) は、表の列を指定するものでなければなりません。また、同じ列を複数回指定することはできません。

この文節の指定がなく、この表が複数パーティション・データベースのパーティション・グループに存在する場合は、その区分化キーが宣言済み一時表の最初の列として定義されます。

宣言済み一時表では、単一パーティションのデータベース・パーティション・グループに定義された表スペースにおいて、すべての列の集合を区分化キーの定義に使用することができます。このパラメーターの指定がない場合、区分化キーは作成されません。

### USING HASHING

データ配分の区分化方式として、ハッシュ関数を使用することを指定します。これは、サポートされる唯一の区分化方式です。

#### 注:

##### • 互換性

- DB2 for OS/390 および z/OS との互換性を保つために、以下のようことができます。
  - 以下の構文はデフォルトの振る舞いとして受け入れられます。
    - CCSID ASCII
    - CCSID UNICODE

- ユーザー一時表スペースは、ユーザー定義の一時表が宣言される前に、存在しなくてはなりません (SQLSTATE 42727)。

- **宣言されたグローバル一時表の参照:** 宣言されたグローバル一時表の記述は DB2 カタログ (SYSCAT.TABLES) に現れないため、この記述は永続的なものではなく、またデータベース接続によって共有することもできません。したがって、同じ *table-name* で宣言されたグローバル一時表を定義するセッションであっても、その宣言されたグローバル一時表の記述はそれぞれのセッションによって異なる可能性があります。

SQL ステートメント (DECLARE GLOBAL TEMPORARY TABLE ステートメントは除く) を使用して宣言されたグローバル一時表を参照するためには、その表をスキーマ名 SESSION で明示的または暗黙的に修飾する必要があります。 *table-name* が SESSION で修飾されていない場合、宣言されたグローバル一時表は参照を決定する際に認識されません。

## DECLARE GLOBAL TEMPORARY TABLE

グローバル一時表が名前によって宣言されていない接続で `SESSION.table-name` を参照する場合は、カタログ内の持続オブジェクトから参照先が決定されます。そのオブジェクトが存在しない場合はエラーが戻されます (SQLSTATE 42704)。

- バインドしているパッケージに、`SESSION` によって暗黙的または明示的に修飾された表を参照する静的 SQL ステートメントが含まれている場合、それらのステートメントは静的にはバインドされません。これらのステートメントは、呼び出されると、パッケージのバインドにおいて `VALIDATE` オプションが選択されているかどうかにかかわらず増分的にバインドされます。ステートメントの実行時には、宣言済み一時表が存在する場合はその一時表に対して、存在しない場合は永続表に対して各表の参照が行われます。このどちらも存在しない場合はエラーが戻されます (SQLSTATE 42704)。
- **特権:** 宣言されたグローバル一時表を定義する場合、その表を定義するユーザーには、表をドロップする権限も含めて、その表に対するすべての表特権が付与されます。加えて、`PUBLIC` に対しても同様の特権が付与されます。( `GRANT` オプションによって付与される特権はありません。また、これらの特権はいずれもカタログ表には現れません。) これらの特権を持つユーザーは、すでに定義されている宣言されたグローバル一時表を参照するセッションで、すべての SQL ステートメントを実行することができます。
- **インスタンス化と終了:** 以下の説明において、それぞれ P はセッションを、T はセッション P 中の宣言済み一時表を示しています。
  - T の空のインスタンスは、P で実行される `DECLARE GLOBAL TEMPORARY TABLE` ステートメントの結果として作成されます。
  - P で実行されるすべての SQL ステートメントでは T を参照することができます。そして、P で T を参照する場合は、すべてその同じインスタンスが参照されます。
  - SQL プロシーチャーの複合ステートメント (`BEGIN` と `END` で定義される) で `DECLARE GLOBAL TEMPORARY TABLE` ステートメントを指定すると、宣言されたグローバル一時表の有効範囲が複合ステートメントだけでなく接続にまで広がり、表は複合ステートメントの外部からも認識されるようになります。表は複合ステートメントの `END` で暗黙的にドロップされません。宣言されたグローバル一時表は、その表が明示的にドロップされない限り、同じ名前を使用してセッション内の他の複合ステートメントで複数回定義することはできません。
  - `ON COMMIT DELETE ROWS` 文節が暗黙的または明示的に指定された場合は、P においてコミット操作が作業単位で終了し、T に属する `WITH HOLD` カーソルが 1 つも P にオープンされていない状態になると、操作 `DELETE FROM SESSION.T` がコミットに組み込まれます。
  - P において、作業単位または保管点でロールバック操作が終了し、その作業単位または保管点に `SESSION.T` への変更が含まれている場合、このロールバック操作には `SESSION.T` からの `DELETE` 操作が含まれます。

## DECLARE GLOBAL TEMPORARY TABLE

P において、作業単位または保管点でロールバック操作が終了し、その作業単位または保管点に SESSION.T への宣言が含まれている場合、このロールバック操作には DROP SESSION.T 操作が含まれます。

P において、作業単位または保管点でロールバック操作が終了し、その作業単位または保管点に宣言済み一時表 SESSION.T のドロップが含まれている場合、このロールバック操作によって表のドロップは取り消されますが、表の行は復元されません。

- アプリケーションによって、宣言された T が終了されたり、データベースとの接続が切断された場合、T はドロップされ、そのインスタンス化された行は破棄されます。
- T の宣言が行われたサーバーへの接続が終了すると、T はドロップされ、そのインスタンス化された行は破棄されます。
- **宣言されたグローバル一時表の使用に関する制限:** 宣言されたグローバル一時表には、以下のような使用上の制限があります。
  - ALTER、COMMENT、GRANT、LOCK、RENAME、または REVOKE ステートメントでこの一時表を指定することはできません (SQLSTATE 42995)。
  - CREATE ALIAS、CREATE FUNCTION (SQL スカラー、表、または行)、CREATE INDEX、CREATE TRIGGER、または CREATE VIEW ステートメントでこの一時表を参照することはできません (SQLSTATE 42995)。
  - 参照制約でこの一時表を指定することはできません (SQLSTATE 42995)。

### 関連資料:

- 1484 ページの『CREATE TABLE』

### 関連サンプル:

- 『tbtemp.sql -- How to use a declared temporary table (C)』
- 『TbTemp.java -- How to use Declared Temporary Table (JDBC)』

## CREATE FUNCTION (外部スカラー)

このステートメントは、ユーザー定義の外部スカラー関数をアプリケーション・サーバーに登録する場合に使用されます。スカラー関数は、呼び出されるたびに 1 つの値を返し、SQL 式が使用可能な個所であれば一般に使用することができます。

### 呼び出し:

このステートメントはアプリケーション・プログラムに組み込むことができ、また動的 SQL ステートメントを使用して出すことができます。DYNAMICRULES の実行動作がパッケージに効力を持つ場合にのみ、動的に準備できる実行可能ステートメントです (SQLSTATE 42509)。

### 許可:

## CREATE FUNCTION (外部スカラー)

ステートメントの許可 ID によって保持されている特権には、少なくとも以下のいずれかが含まれていなければなりません。

- SYSADM または DBADM 権限
- データベースに対する CREATE\_EXTERNAL\_ROUTINE 権限、および以下の少なくとも 1 つ。
  - データベースに対する IMPLICIT\_SCHEMA 権限 (関数のスキーマ名が既存のスキーマを指していない場合)
  - スキーマに対する CREATEIN 特権 (関数のスキーマ名が既存のスキーマを指している場合)

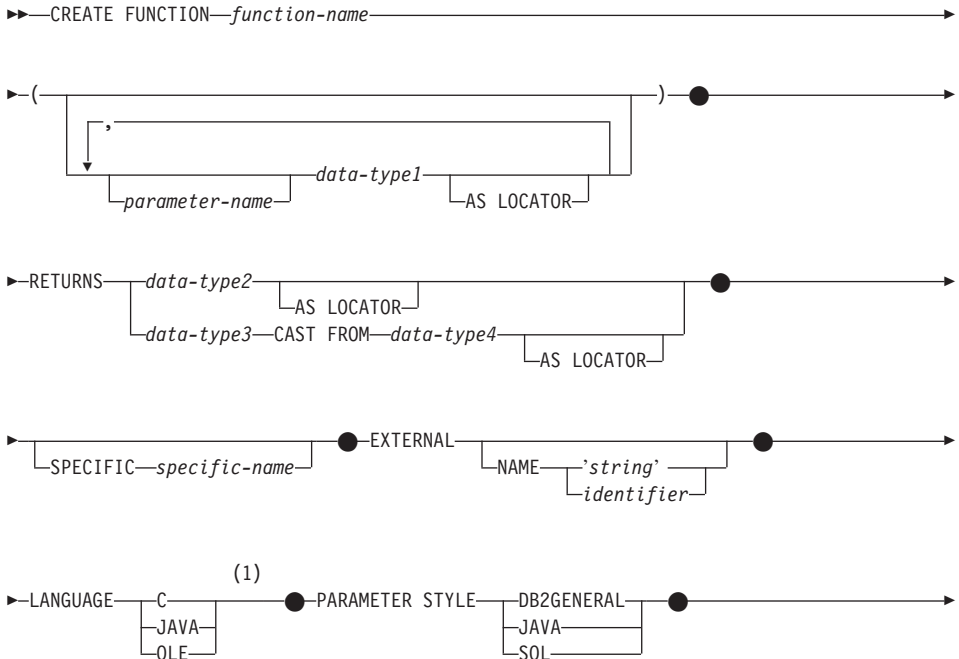
非隔離の関数を作成するには、ステートメントの許可 ID の特権に、以下の特権の少なくとも 1 つが含まれている必要があります。

- データベースに対する CREATE\_NOT\_FENCED\_ROUTINE 権限
- SYSADM または DBADM 権限

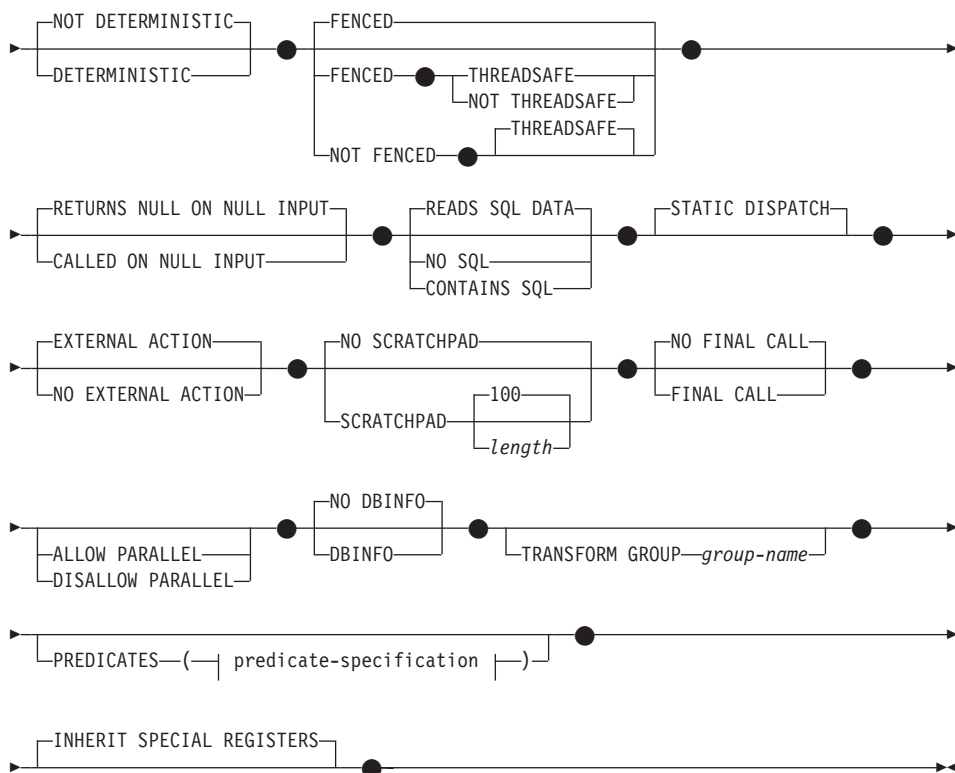
隔離関数を作成する場合には、さらに別の権限や特権は必要ありません。

許可 ID の権限が不十分で、操作を実行できない場合には、エラー (SQLSTATE 42502) になります。

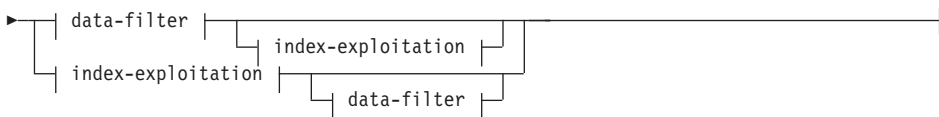
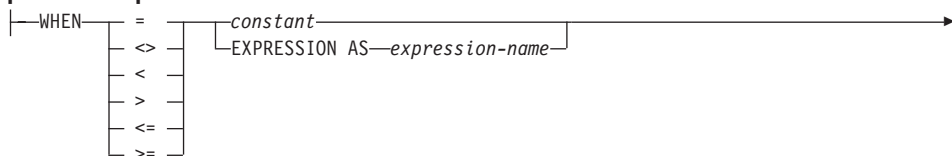
### 構文:



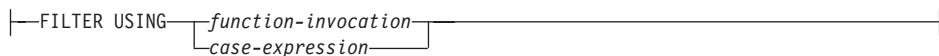
## CREATE FUNCTION (外部スカラー)



### predicate-specification:



### data-filter:



### index-exploitation:



|—SEARCH BY—  
 └─EXACT─┘ INDEX EXTENSION—*index-extension-name*—

└─exploitation-rule—┘

**exploitation-rule:**

|—WHEN KEY—(—*parameter-name1*—)—

└─USE—*search-method-name*—(—*parameter-name2*—)—

**注:**

- 1 LANGUAGE SQL もサポートされています。

**説明:***function-name*

定義する関数の名前を指定します。これは、関数を指定する修飾または非修飾の名前です。*function-name* (関数名) の非修飾形式は SQL ID です (最大長 18)。動的 SQL ステートメントでは、CURRENT SCHEMA 特殊レジスターが、修飾子のないオブジェクト名の修飾子として使用されます。静的 SQL ステートメントでは、QUALIFIER プリコンパイル/バインド・オプションにより、修飾子のないオブジェクト名の修飾子が暗黙指定されます。修飾形式は、*schema-name* の後にピリオドと SQL ID が続きます。最初のパラメーターが構造タイプの場合、修飾名は、最初のパラメーターのデータ・タイプと同じではありません。

暗黙または明示の修飾子を含む名前、およびパラメーターの数と各パラメーターのデータ・タイプ (データ・タイプの長さ、精度、または位取りの各属性には関係なく) は、カタログに記述されている関数またはメソッドを指定するものではありません (SQLSTATE 42723)。非修飾名とパラメーターの数およびデータ・タイプとの組み合わせは、そのスキーマ内では当然固有ですが、複数のスキーマ間で固有である必要はありません。

2 つの部分から成る名前を指定する場合、“SYS” で始まる *schema-name* (スキーマ名) は使用できません。使用した場合、エラー (SQLSTATE 42939) になります。

述部のキーワードとして使用される多くの名前は、システム使用として予約されており、*function-name* として使用することはできません。それらの名前は、SOME、ANY、ALL、NOT、AND、OR、BETWEEN、NULL、LIKE、EXISTS、IN、UNIQUE、OVERLAPS、SIMILAR、MATCH、および比較演算子です。この規則に違反すると、エラーになります (SQLSTATE 42939)。

## CREATE FUNCTION (外部スカラー)

一般に、関数のシグニチャーに何らかの差異がある場合には、同じ名前を複数の関数に使用することができます。

禁止されてはいませんが、意図的にオーバーライドを行う場合を除き、外部ユーザが一定義関数の名前として、組み込み関数と同じ名前を指定するのは避けるべきです。異なる意味を持つ関数に組み込みのスカラー関数または列関数と同じ名前 (たとえば、LENGTH、VALUE、MAX など) を与えることは、たとえその引き数が一致していたとしても、動的 SQL ステートメントの過程で、あるいは静的 SQL アプリケーションの再バインド時に問題が生じます。すなわち、アプリケーションが失敗することがあり、また、さらに悪いケースとして、外見上は正常に実行されていても、結果が異なる場合があります。

### *parameter-name*

後続の関数定義で使用できるパラメーターを指定します。パラメーター名は、述部指定の *index-exploitation* 文節にある関数のパラメーターを参照するのに必要です。

### *(data-type1,...)*

関数の入力パラメーターの数を指定するとともに、各パラメーターのデータ・タイプを指定します。このリストには、関数が受け取ることを予期している各パラメーターごとに 1 つの項目を指定する必要があります。パラメーターの数は 90 を超えることはできません。この限界を超えると、エラー (SQLSTATE 54023) になります。

パラメーターのない関数も登録可能です。この場合、指定するデータ・タイプがない場合でも、括弧はコーディングする必要があります。たとえば、以下のようになります。

```
CREATE FUNCTION WOOFER() ...
```

その対応するすべてのパラメーターのタイプがまったく同じである場合でも、1 つのスキーマ中に名前が同じ 2 つの関数があることはありません。このタイプの比較では長さ、精度、および位取りは考慮されません。したがって、CHAR(8) と CHAR(35)、また DECIMAL(11,2) と DECIMAL (4,3) は、それぞれ同じタイプと見なされます。ユニコード・データベースの場合には、CHAR(13) と GRAPHIC(8) は、それぞれ同じタイプと見なされます。さらに、DECIMAL と NUMERIC などのように、この目的で複数のタイプが同じタイプとして扱われることがあります。シグニチャーが重複していると、SQL エラー (SQLSTATE 42723) になります。

たとえば、以下のステートメントの場合、

```
CREATE FUNCTION PART (INT, CHAR(15)) ...
CREATE FUNCTION PART (INTEGER, CHAR(40)) ...
CREATE FUNCTION ANGLE (DECIMAL(12,2)) ...
CREATE FUNCTION ANGLE (DEC(10,7)) ...
```

2 番目と 4 番目のステートメントは、重複する関数と見なされ、エラーになります。

*data-type1*

パラメーターのデータ・タイプを指定します。

- CREATE TABLE ステートメントの *data-type1* の定義で指定が可能で、関数の作成に使用されている言語において対応するものがある SQL データ・タイプ仕様と省略形を指定できます。
- DECIMAL (および NUMERIC) は、LANGUAGE C と OLE では無効です (SQLSTATE 42815)。
- REF(*type-name*) は、パラメーターのタイプとして指定できます。ただし、パラメーターに有効範囲を指定してはなりません。
- 適切なトランスフォーム関数が、関連するトランスフォーム・グループに存在する場合には、構造タイプを指定できます。

**AS LOCATOR**

LOB タイプまたは LOB タイプに基づく特殊タイプの場合、AS LOCATOR 文節を追加することができます。これは、実際の値の代わりに LOB ロケーターを UDF に渡すことを指定します。これにより、UDF に渡すバイト数を大幅に減らすことができ、パフォーマンスも向上します (特に、UDF にとって実際に必要になる値が数バイトだけである場合)。

以下の例は、パラメーター定義の AS LOCATOR 文節の使用法を示しています。

```
CREATE FUNCTION foo (CLOB(10M) AS LOCATOR, IMAGE AS LOCATOR)
...
```

ここで、IMAGE は LOB タイプの 1 つに基づく特殊タイプであると想定します。

また、引き数のプロモーション目的には、AS LOCATOR 文節の効果はないことに注意してください。この例では、タイプはそれぞれ CLOB と IMAGE であると見なされるので、関数に CHAR 引き数または VARCHAR 引き数が最初の引き数として渡されます。同様に、関数シグニチャーに対して AS LOCATOR の効果はありません。関数シグニチャーは、(a) "関数解決" と呼ばれるプロセスによって DML で参照された場合、(b) COMMENT ON や DROP などの DDL ステートメントで参照された場合に関数をマッチングする際に使用されます。実際に、この文節はシグニチャーの指定のない COMMENT ON や DROP で使用しても、しなくても構いません。

LOB または LOB に基づく特殊タイプ以外のタイプに対して AS LOCATOR を指定すると、エラー (SQLSTATE 42601) が発生します。

関数が FENCED で NO SQL オプションを持っている場合、AS LOCATOR 文節は指定できません (SQLSTATE 42613)。

## CREATE FUNCTION (外部スカラー)

### RETURNS

これは必須の文節であり、関数の出力を指定します。

*data-type2*

出力のデータ・タイプを指定します。

この場合、上記の関数パラメーター *data-type1* で説明した外部関数のパラメーターと同じ考慮事項が適用されます。

### AS LOCATOR

LOB タイプまたは LOB タイプに基づく特殊タイプの場合、AS LOCATOR 文節を追加することができます。これは、実際の値の代わりに LOB ロケーターが UDF から渡されることを示します。

*data-type3* **CAST FROM** *data-type4*

出力のデータ・タイプを指定します。

この形式の RETURNS 文節は、関数コードから戻されたデータ・タイプとは異なるデータ・タイプを、呼び出しステートメントに戻すのに使用されます。たとえば、以下の例で、

```
CREATE FUNCTION GET_HIRE_DATE(CHAR(6))
 RETURNS DATE CAST FROM CHAR(10)
```

...

CHAR(10) の値が関数コードからデータベース・マネージャーに戻され、データベース・マネージャーは、その値を DATE に変換して、変換された値を呼び出し側ステートメントに渡します。 *data-type4* は、 *data-type3* パラメーターに対してキャスト可能でなければなりません。キャスト可能でない場合、エラー (SQLSTATE 42880) になります。

*data-type3* の長さ、精度または位取りは、 *data-type4* から推断することができるので、 *data-type3* に指定されるパラメーター化タイプの長さ、精度、または位取りを指定する必要はありません (指定は可能です)。代わりに、空の括弧を使用できます (たとえば、VARCHAR() など)。パラメーター値が異なるデータ・タイプ (REAL または DOUBLE) を示しているため、FLOAT() を使用することはできません (SQLSTATE 42601)。

特殊タイプおよび構造タイプは、 *data-type4* に指定するタイプとしては無効です (SQLSTATE 42815)。

キャスト操作は、変換エラーになる可能性がある実行時チェックの対象にもなります。

### AS LOCATOR

*data-type4* の指定が、LOB タイプまたは LOB タイプに基づく特殊タイプの場合、AS LOCATOR 文節を追加することができます。これは、実際の値の代わりに LOB ロケーターが UDF から戻されることを示します。

**SPECIFIC** *specific-name*

定義する関数のインスタンスに対する固有の名を指定します。この特定名は、この関数をソース関数として使用する場合、この関数をドロップする場合、またはこの関数にコメントを付ける場合に使用することができます。これは、関数の呼び出しには使用できません。 *specific-name* (関数名) の非修飾形式は SQL ID です (最大長 18)。修飾形式は、*schema-name* の後にピリオドと SQL ID が続きます。暗黙または明示の修飾子も含め、その名前が、アプリケーション・サーバーに存在する別の関数インスタンスまたはメソッド指定を識別するものであってはなりません。そうでない場合、エラー (SQLSTATE 42710) になります。

*specific-name* は、既存の *function-name* (関数名) と同じでも構いません。

修飾子を指定しない場合、*function-name* に使用された修飾子を使用されます。修飾子を指定する場合は、*function-name* の明示修飾子または暗黙修飾子と同じでなければなりません。そうでない場合、エラー (SQLSTATE 42882) になります。

*specific-name* の指定がない場合、固有の名前がデータベース・マネージャーによって生成されます。生成される固有の名は、SQL の後に文字のタイム・スタンプが続く名前です (SQLyymmddhhmmssxxx)。

**EXTERNAL**

この文節は、外部プログラミング言語で作成され、文書化されたリンケージの規則とインターフェースに準拠している新しい関数を登録するのに、CREATE FUNCTION ステートメントが使用されていることを示します。

NAME 文節を指定しない場合、"NAME *function-name*" が想定されます。

**NAME** '*string*'

この文節は、定義している関数をインプリメントするユーザー作成コードの名前を指定します。

'*string*' オプションは、最大 254 文字のストリング定数です。ストリングに使用される形式は、指定した LANGUAGE によって異なります。

- LANGUAGE C の場合

指定する *string* (ストリング) は、作成しているユーザー定義関数を実行するためにデータベース・マネージャーが呼び出すライブラリー名と、そのライブラリー中の関数名です。ライブラリー (およびそのライブラリー中の関数) は、CREATE FUNCTION ステートメントの実行時に存在している必要はありません。ただし、関数が SQL ステートメントで使用される時点では、そのライブラリーとそのライブラリー中の該当の関数は存在していなければならない。データベース・サーバーのマシンからアクセス可能でなければなりません。そうでない場合、エラー (SQLSTATE 42724) が戻されます。

```
→ 'library_id | absolute_path_id | !func_id'
```

単一引用符内に、余分なブランクを使用することはできません。

## CREATE FUNCTION (外部スカラー)

### *library\_id*

関数を含むライブラリー名を指定します。データベース・マネージャーは、`.../sqllib/function` ディレクトリー (UNIX 系システムの場合)、または `..¥instance_name¥function` ディレクトリー (DB2INSTPROF レジストリー変数で指定した Windows オペレーティング・システムの場合) を調べてそのライブラリーを探します。その中で、データベース・マネージャーは、データベース・マネージャーの実行に使用されている制御 `sqllib` ディレクトリーを見つけ出します。たとえば、UNIX 系システムの制御 `sqllib` ディレクトリーは、`/u/$DB2INSTANCE/sqllib` です。

UNIX 系システムの *library\_id* が `'myfunc'` で、データベース・マネージャーが `/u/production` から実行されている場合、ライブラリー `/u/production/sqllib/function/myfunc` から関数を探索します。

Windows オペレーティング・システムの場合、*library\_id* が関数ディレクトリーにないと、データベース・マネージャーは `LIBPATH` または `PATH` を探します。

### *absolute\_path\_id*

関数を含んでいるファイルの絶対パス名を指定します。

たとえば、UNIX 系システムの場合、`'u/jchui/mylib/myfunc'` を指定すると、データベース・マネージャーは `/u/jchui/mylib` を調べて `myfunc` 共有ライブラリーを探します。

Windows オペレーティング・システムの場合、`'d:¥mylib¥myfunc'` を指定すると、データベース・マネージャーは `d:¥mylib` ディレクトリーからダイナミック・リンク・ライブラリー `myfunc.dll` ファイルをロードします。

注: Windows オペレーティング・システムで C または C++ のルーチンを登録しているとき、`EXTERNAL NAME` 文節内でルーチン本体識別しているときに、次のことに注意してください。絶対パス ID がルーチン本体の識別に使用されている場合は、`.dll` 拡張子を必ず付加してください。

### *! func\_id*

呼び出される関数の入り口点名を指定します。!`func_id` は、ライブラリー ID と関数 ID との間の区切り文字です。`! func_id` を省略すると、データベース・マネージャーはライブラリーのリンク時に確立されたデフォルトの入り口点を使用します。

たとえば、UNIX 系システムで `'mymod!func8'` と指定すると、データベース・マネージャーはライブラリー

`$inst_home_dir/sqllib/function/mymod` を調べて、そのライブラリー内の入り口点 `func8` を使用します。

## CREATE FUNCTION (外部スカラー)

Windows オペレーティング・システムの場合 'mymod!func8' を指定すると、データベース・マネージャーは mymod.dll ファイルをロードして、そのダイナミック・リンク・ライブラリー (DLL) の func8() 関数を呼び出します。

ストリングの形式が正しくない場合には、エラー (SQLSTATE 42878) になります。

すべての外部関数の本体は、データベースのすべてのパーティションで使用可能なディレクトリーにある必要があります。

- LANGUAGE JAVA の場合

指定する *string* には、作成中のユーザー定義関数を実行するためにデータベース・マネージャーが呼び出す、任意指定の jar ファイル、クラス ID、およびメソッド ID が含まれています。クラス ID とメソッド ID は、CREATE FUNCTION ステートメントの実行時には存在している必要はありません。jar\_id を指定する場合、ID は、CREATE FUNCTION ステートメントの実行時に存在していなければなりません。ただし、関数を SQL ステートメントで使用する時点で、メソッド ID は存在していなければならず、データベース・サーバーのマシンからアクセス可能でなければなりません。そうでない場合、エラー (SQLSTATE 42724) になります。

▶▶ ' [ jar\_id : ] class\_id . [ ! ] method\_id ' ◀◀

単一引用符内に、余分な空白を使用することはできません。

### jar\_id

jar の集合をデータベースへインストールしたときに、その jar の集合に付けられた jar ID を指定します。これは、単純 ID またはスキーマ修飾 ID のいずれかにすることができます。たとえば、'myJar' や 'mySchema.myJar' のようになります。

### class\_id

Java オブジェクトのクラス ID を指定します。クラスがパッケージの一部である場合、クラス ID の部分に完全なパッケージ接頭部 (例: 'myPacks.UserFuncs') が含まれている必要があります。Java 仮想マシンは、ディレクトリー '.../myPacks/UserFuncs/' 中のクラスを探します。Windows オペレーティング・システムでは、Java 仮想マシンはディレクトリー '...¥myPacks¥UserFuncs¥' を探索します。

### method\_id

呼び出す Java オブジェクトのメソッド名を指定します。

- LANGUAGE OLE の場合

指定する *string* は、作成中のユーザー定義関数を実行するためにデータベース・マネージャーが呼び出す、OLE のプログラム ID (progid) またはクラス ID (clsid)、およびメソッド ID です。プログラム ID またはクラス ID、



## CREATE FUNCTION (外部スカラー)

およびメソッド ID は、CREATE FUNCTION ステートメントの実行時に存在している必要はありません。ただし、関数を SQL ステートメントで使用する時点で、メソッド ID は存在していなければならず、データベース・サーバーのマシンからアクセス可能でなければなりません。そうでない場合、エラー (SQLSTATE 42724) になります。

▶▶' \_\_\_\_\_!-method\_id-' \_\_\_\_\_▶▶  
      └──progid──┘  
      └──clsid──┘

単一引用符内に、余分なブランクを使用することはできません。

### *progid*

OLE オブジェクトのプログラム ID を指定します。

*progid* は、データベース・マネージャーには解釈されず、実行時に OLE API に転送されるだけです。指定する OLE オブジェクトは、作成可能である必要があり、実行時バインディング (ディスパッチに基づくバインディングとも呼ばれる) をサポートしている必要があります。

### *clsid*

作成する OLE オブジェクトのクラス ID を指定します。OLE オブジェクトが *progid* を指定して登録されていない場合に、*progid* を指定する代わりに使用することができます。*clsid* の形式は以下のとおりです。

{nnnnnnnnn-nnnn-nnnn-nnnn-nnnnnnnnnnnnn}

ここで 'n' は英数字です。*clsid* は、データベース・マネージャーには解釈されず、実行時に OLE API に転送されるだけです。

### *method\_id*

呼び出す OLE オブジェクトのメソッド名を指定します。

## NAME *identifier*

指定する *identifier* は SQL ID です。SQL ID は、ストリングの *library-id* として使用されます。区切られた ID でない場合、ID は大文字に変換されます。ID がスキーマ名で修飾されている場合、スキーマ名の部分は無視されません。この形式の NAME は、LANGUAGE C でのみ使用可能です。

## LANGUAGE

この文節は必須で、ユーザー定義関数の本体が準拠している言語インターフェース規則を指定するのに使用します。

**C** これは、データベース・マネージャーが、ユーザー定義関数を C の関数であるかのように呼び出すことを意味します。ユーザー定義関数は、標準 ANSI C プロトタイプで定義されている C 言語の呼び出しおよびリンケージの規則に準拠していなければなりません。

**JAVA** データベース・マネージャーは、Java クラスのメソッドとしてユーザー定義関数を呼び出します。



**OLE** データベース・マネージャーは、OLE 自動化オブジェクトによって公開されたメソッドとして、ユーザー定義関数を呼び出します。ユーザー定義関数は、*OLE Automation Programmer's Reference* に説明されている、OLE 自動化データ・タイプと呼び出しメカニズムに準拠している必要があります。

LANGUAGE OLE は、DB2 (Windows オペレーティング・システム版) で保管されたユーザー定義関数に対してのみサポートされます。

LANGUAGE OLE を指定した UDF には、THREADSAFE は指定できません (SQLSTATE 42613)。

### PARAMETER STYLE

この文節は、関数にパラメーターを渡し、関数から値を戻すのに用いる規則を指定するために使用します。

### DB2GENERAL

Java クラスのメソッドとして定義された外部関数との間で、パラメーターを渡し、値を戻す場合に用いる規則を指定します。これは、LANGUAGE JAVA を使用する場合にだけ指定する必要があります。

DB2GENERAL の同義語として値 DB2GENRL が使用可能です。

### JAVA

関数は、Java 言語および SQLj ルーチンの仕様に準拠する、パラメーターの受け渡し規則を使用します。これは、LANGUAGE JAVA が使用され、パラメーターまたは戻りタイプに構造タイプがない場合にのみ指定できます (SQLSTATE 429B8)。PARAMETER STYLE JAVA 関数は、FINAL CALL、SCRATCHPAD または DBINFO 文節をサポートしていません。

### SQL

この外部関数との間でパラメーターを渡し、値を戻す場合の規則として、C 言語の呼び出しとリンケージの規則、または OLE 自動化オブジェクトによって公開されたメソッドに準拠する規則を指定します。これは、LANGUAGE C または LANGUAGE OLE を使用する場合に指定する必要があります。

### DETERMINISTIC または NOT DETERMINISTIC

この文節は任意指定で、特定の引き数の値に対して関数が常に同じ結果を戻すか (DETERMINISTIC)、それとも状態値に依存して関数の結果が影響を受けるか (NOT DETERMINISTIC) を指定します。つまり DETERMINISTIC を伴う関数は、同じ入力を指定して呼び出しが行われた場合に常に同じ結果を戻します。NOT DETERMINISTIC を指定すると、同じ入力によって常に同じ結果が生じる利点に基づく最適化ができなくなります。乱数を生成する関数は、NOT DETERMINISTIC 関数の例です。入力の平方根を求める関数は、DETERMINISTIC 関数の例です。

### FENCED または NOT FENCED

この文節は、関数をデータベース・マネージャーの操作環境のプロセスまたはアドレス・スペースで実行しても“安全”かを指定します。

## CREATE FUNCTION (外部スカラー)

関数が **FENCED** として登録されると、データベース・マネージャーは、その内部リソース (データ・バッファーなど) を保護して、その関数からアクセスされないようにします。多くの関数は、**FENCED** または **NOT FENCED** のどちらかで実行するように選択することができます。一般に、**FENCED** として実行される関数は、**NOT FENCED** として実行されるものと同じようには実行されません。

### 注意:

適切にコード化、検討、およびテストされていない関数に **NOT FENCED** を使用すると、**DB2** の健全性に危険を招く場合があります。**DB2** では、発生する可能性のある一般的な不注意による障害の多くに対して、いくつかの予防措置がとられています。 **NOT FENCED** ユーザー定義関数が使用される場合には、完全な健全性を確保できません。

**LANGUAGE OLE** または **NOT THREADSAFE** を指定した関数には、**FENCED** のみを指定できます (SQLSTATE 42613)。

関数が **FENCED** で **NO SQL** オプションを持っている場合、**AS LOCATOR** 文節は指定できません (SQLSTATE 42613)。

ユーザー定義関数を **NOT FENCED** として登録するには、**SYSADM** 権限、**DBADM** 権限、または特殊権限 (**CREATE\_NOT\_FENCED\_ROUTINE**) が必要です。

### **THREADSAFE** または **NOT THREADSAFE**

関数を他のルーチンと同じプロセスで実行しても安全か (**THREADSAFE**)、そうでないか (**NOT THREADSAFE**) を指定します。

関数が **OLE** 以外の **LANGUAGE** で定義される場合:

- 関数が **THREADSAFE** に定義されている場合には、データベース・マネージャーは他のルーチンと同じプロセスで関数を呼び出すことができます。一般に、スレッド・セーフにするには、関数はどのグローバルあるいは静的データ域をも使用してはなりません。多くのプログラミング解説書には、スレッド・セーフ・ルーチンの作成に関する説明が含まれています。 **FENCED** および **NOT FENCED** 関数の両方が **THREADSAFE** になることが可能です。
- 関数が **NOT THREADSAFE** に定義される場合には、データベース・マネージャーは関数を他のルーチンと同じプロセスに決して呼び出しません。

**FENCED** 関数の場合、**LANGUAGE** が **JAVA** なら **THREADSAFE** がデフォルトです。これ以外のすべての言語の場合は、**NOT THREADSAFE** がデフォルトです。関数が **LANGUAGE OLE** に定義される場合には、**THREADSAFE** は指定できません (SQLSTATE 42613)。

**NOT FENCED** 関数の場合には、**THREADSAFE** がデフォルトです。 **NOT THREADSAFE** を指定することはできません (SQLSTATE 42613)。

### **RETURNS NULL ON NULL INPUT** または **CALLED ON NULL INPUT**

このオプション文節を使用すると、引き数のいずれかが NULL 値の場合に、外部関数を呼び出さないようにすることができます。ユーザー定義関数がパラメーターなしで定義される場合、この NULL 引き数条件が引き起こされることはなく、この仕様のコーディング方法はそれほど重要ではなくなります。

**RETURNS NULL ON NULL INPUT** が指定されており、実行時に関数の引き数のいずれかが NULL 値の場合、このユーザー定義関数は呼び出されず、結果は NULL 値になります。

**CALLED ON NULL INPUT** が指定されると、引き数が NULL 値か否かに関係なくユーザー定義関数が呼び出されます。これは、NULL 値を戻す場合も、通常の (NULL 値以外の) 値を戻す場合もあります。ただし、NULL の引き数値の有無のテストは UDF が行う必要があります。

値 NULL CALL は、上位互換またはファミリーの互換性のために、**CALLED ON NULL INPUT** の同義語として使うことができます。同様に、**NOT NULL CALL** は、**RETURNS NULL ON NULL INPUT** の同義語として使用できます。

### **NO SQL, CONTAINS SQL, READS SQL DATA**

関数から SQL ステートメントが発行されるかどうかと、もし発行されればどのタイプかを示します。

#### **NO SQL**

関数はどの SQL ステートメントも実行できないことを指示します (SQLSTATE 38001)。

#### **CONTAINS SQL**

SQL データの読み取りも変更も行わない SQL ステートメントを、関数で実行できることを指定します (SQLSTATE 38004 または 42985)。どの関数でもサポートされていないステートメントは、これとは異なるエラーを戻します (SQLSTATE 38003 または 42985)。

#### **READS SQL DATA**

SQL データを変更しない SQL ステートメントを、関数で実行できることを指定します (SQLSTATE 38002 または 42985)。どの関数でもサポートされていないステートメントは、これとは異なるエラーを戻します (SQLSTATE 38003 または 42985)。

### **STATIC DISPATCH**

このオプション文節は、問題解決時に DB2 が関数のパラメーターの静的タイプ (宣言済みタイプ) に基づいて関数を選択するよう指示します。

### **NO EXTERNAL ACTION** または **EXTERNAL ACTION**

このオプションの文節は、データベース・マネージャーによって管理されていないオブジェクトの状態を変更するアクションを関数が行うかどうかを指定します。

## CREATE FUNCTION (外部スカラー)

EXTERNAL ACTION を指定すると、関数に外部の影響がないことを前提とした最適化ができなくなります。(たとえば、メッセージの送信、警報音による通知、ファイルへのレコードの書き込みなど。)

### NO SCRATCHPAD または SCRATCHPAD *length*

この文節はオプションであり、この外部関数に対してスクラッチパッドを用意するか否かを指定するのに使用することができます。(ユーザー定義関数を再入可能にすることを強くお勧めします。再入可能にすると、スクラッチパッドによってある呼び出しと次の呼び出しとの間に関数が“状態を保管する”手段が用意されます。)

SCRATCHPAD を指定すると、ユーザー定義関数の最初の呼び出し時に、その外部関数によって使用されるスクラッチパッドにメモリーが割り振られます。このスクラッチパッドには、以下の特性があります。

- *length* を指定すると、スクラッチパッドのサイズをバイト単位で設定できます。この値は 1 ~ 32 767 で指定できます (SQLSTATE 42820)。デフォルト・サイズは 100 バイトです。
- すべて X'00' に初期化されます。
- その有効範囲は、該当の SQL ステートメントです。SQL ステートメントでの外部関数に対する参照ごとに 1 つのスクラッチパッドがあります。したがって、以下のステートメントの関数 UDFX が SCRATCHPAD キーワードを指定して定義されている場合、3 つのスクラッチパッドが割り当てられます。

```
SELECT A, UDFX(A) FROM TABLEB
WHERE UDFX(A) > 103 OR UDFX(A) < 19
```

ALLOW PARALLEL が指定されているか、またはデフォルト値として使用された場合、その有効範囲は上記とは異なります。関数が複数のパーティションで実行される場合、関数が処理されるそれぞれのパーティションにおいて、SQL ステートメントでの関数へのそれぞれの参照ごとにスクラッチパッドが割り当てられます。同様に、パーティション内並列処理をオンにして照会が実行される場合、3 つ以上のスクラッチパッドが割り当てられることがあります。

- スクラッチパッドは持続します。その内容は、外部関数のある呼び出しから次の呼び出しになっても持続します。外部関数のある呼び出しによってスクラッチパッドに対して行われた変更はいずれも、次の呼び出し時に持続しています。データベース・マネージャーは、各 SQL ステートメントの実行開始時に、スクラッチパッドを初期設定します。各副照会の実行開始時には、データベース・マネージャーによってスクラッチパッドがリセットされます。FINAL CALL オプションが指定されている場合、システムは、スクラッチパッドのリセットに先立って、最終呼び出しを行います。
- これは、外部関数が獲得するシステム・リソース (メモリーなどの) の中央点として使用することもできます。関数は、最初の呼び出しでメモリーを獲得し、そのアドレスをスクラッチパッドに保管して、後の呼び出しでそれを参照することができます。

(このようにシステム・リソースが獲得される場合、FINAL CALL キーワードも指定する必要があります。これにより、ステートメントの最後で特殊な呼び出しが行われ、外部関数は獲得したシステム・リソースをすべて解放することができます。)

SCRATCHPAD を指定すると、ユーザー定義関数を呼び出すたびに、スクラッチパッドをアドレッシングする外部関数に追加の引き数が渡されます。

NO SCRATCHPAD を指定すると、外部関数に対してスクラッチパッドは割り振られず、渡されません。

SCRATCHPAD は、PARAMETER STYLE JAVA 関数ではサポートされていません。

### FINAL CALL または NO FINAL CALL

この文節はオプションであり、外部関数に対する最終呼び出しが行われるか否かを指定します。このような最終呼び出しの目的は、外部関数が、獲得したシステム・リソースすべてを解放できるようにすることです。外部関数がメモリーなどのシステム・リソースを獲得し、それをスクラッチパッドに固定するような状況では、これを SCRATCHPAD キーワードと共に使用すると便利です。FINAL CALL を指定した場合、実行時点で以下が行われます。

- 呼び出しのタイプを指定する追加の引き数が外部関数に渡されます。呼び出しのタイプは以下のとおりです。
  - 通常呼び出し。SQL 引き数が渡され、結果が戻されることが予期されます。
  - 最初の呼び出し。この SQL ステートメントのユーザー定義関数に対する参照に対応する外部関数の最初の呼び出し。最初の呼び出しは通常呼び出しです。
  - 最終呼び出し。外部関数がリソースを解放できるようにするための、その関数に対する最終呼び出し。最終呼び出しは、通常呼び出しではありません。この最終呼び出しは、以下の時点で行われます。
    - ステートメント終了時。これは、カーソルの関係するステートメントでカーソルがクローズされた場合、あるいはステートメントが実行を終了した場合に発生します。
    - 並列タスクの終了時。これは、関数が並列タスクで実行された場合に発生します。
    - トランザクション終了時または割り込み時。これは、通常のステートメント終了が発生しなかった場合に発生します。たとえば、何らかの理由で、アプリケーションのロジックが、カーソルをクローズしないようになっている場合があります。このタイプの最終呼び出しの際、CLOSE カーソル以外は、SQL ステートメントが発行されない可能性があります (SQLSTATE 38505)。このタイプの最終呼び出しは、「呼び出しタイプ」の引き数の特殊値で指示されます。

## CREATE FUNCTION (外部スカラー)

WITH HOLD として定義されたカーソルがオープンされている間に、コミット操作が発生すると、それ以降のカーソルのクローズ時、またはアプリケーションの終了時に最終呼び出しが行われます。

NO FINAL CALL を指定すると、“呼び出しタイプ” の引き数は外部関数に渡されず、最終呼び出しは行われません。

FINAL CALL は、PARAMETER STYLE JAVA 関数ではサポートされていません。

### ALLOW PARALLEL または DISALLOW PARALLEL

この文節はオプションで、関数への 1 つの参照に対して、関数の呼び出しを並列化できるか否かを指定します。一般には、ほとんどのスカラー関数は並列化可能ですが、並列化できない関数 (1 つのスクラッチパッドのコピーに依存する関数など) もあります。スカラー関数に対して ALLOW PARALLEL または DISALLOW PARALLEL を指定すると、DB2 はその指定を受け入れます。関数にどちらのキーワードが当てはまるかを判別するには、以下の点について検討する必要があります。

- UDF のすべての呼び出しが、互いに完全に独立していますか? YES の場合には、ALLOW PARALLEL を指定します。
- UDF を呼び出すごとに、次の呼び出しに関係する値を提供するスクラッチパッドが更新されますか? (たとえば、カウンターの増分など。) YES の場合には、DISALLOW PARALLEL を指定するか、またはデフォルトを受け入れます。
- 1 つのパーティションでのみ起こる必要のある外部アクションが UDF によって実行されますか? YES の場合には、DISALLOW PARALLEL を指定するか、またはデフォルトを受け入れます。
- コストのかかる初期化処理の実行回数を最小にするためだけに、スクラッチパッドを使用していますか? YES の場合には、ALLOW PARALLEL を指定します。

いずれの場合も、すべての外部関数の本体は、データベースのすべてのパーティションで使用可能なディレクトリーにある必要があります。

ステートメントで以下の 1 つ以上のオプションが指定されている場合以外は、デフォルト値は ALLOW PARALLEL です。

- NOT DETERMINISTIC
- EXTERNAL ACTION
- SCRATCHPAD
- FINAL CALL

これらのオプションのいずれかが指定または暗黙指定されている場合は、デフォルト値は DISALLOW PARALLEL です。

### INHERIT SPECIAL REGISTERS

このオプション文節は、関数の更新可能な特殊レジスターが、呼び出しステートメントの環境からの初期値を継承することを指定します。カーソルの select ステート



メントで呼び出される関数の場合、初期値はカーソルがオープンした際の環境から継承します。ネストされたオブジェクト (たとえば、トリガーまたはビュー) に呼び出されるルーチンの場合、初期値は (オブジェクト定義から継承するのではなく) 実行時環境から継承します。

特殊レジスターに対する変更が、関数の呼び出し元に戻されることはありません。

更新不能の特殊レジスター (日時特殊レジスターなど) は、現在実行中のステートメントのプロパティを反映するので、デフォルト値に設定されます。

### NO DBINFO または DBINFO

この文節はオプションで、DB2 において既知である特定の情報を追加の呼び出し時引き数として UDF に渡すか (DBINFO)、または渡さないか (NO DBINFO) を指定します。NO DBINFO がデフォルト値です。DBINFO は、LANGUAGE OLE ではサポートされません (SQLSTATE 42613)。また、PARAMETER STYLE JAVA でもサポートされません。

DBINFO を指定すると、以下の情報を含む構造が UDF に渡されます。

- データベース名 - 現在接続されているデータベースの名前。
- アプリケーション ID - データベースへの接続ごとに確立された、固有のアプリケーション ID。
- アプリケーション許可 ID - アプリケーション実行時の許可 ID。この UDF とアプリケーションとの間でネストされている UDF は無関係。
- コード・ページ - データベースのコード・ページを識別します。
- スキーマ名 - 表名とまったく同じ条件のもとでは、スキーマの名前が入ります。その他の場合はブランクです。
- 表名 - UDF 参照が UPDATE ステートメントの SET 文節の右側にある場合、または INSERT ステートメントの VALUES リストの項目である場合のいずれかに限り、更新または挿入される表の非修飾名が入ります。その他の場合はブランクです。
- 列名 - 表名とまったく同じ条件で、更新または挿入される列の名前が入ります。その他の場合はブランクです。
- データベースのバージョン / リリース - UDF を呼び出すデータベース・サーバーのバージョン、リリースおよび修正レベルを識別します。
- プラットフォーム - サーバーのプラットフォーム・タイプが入ります。
- 表関数の結果の列番号 - 外部スカラー関数には当てはまりません。

### TRANSFORM GROUP *group-name*

関数を呼び出す際のユーザー定義の構造タイプのトランスフォーメーションに使用するトランスフォーム・グループを指定します。関数定義にパラメーターまたは RETURNS データ・タイプとしてユーザー定義の構造タイプが含まれている場合、トランスフォームが必要になります。この文節が指定されない場合には、デフォルトのグループ名 DB2\_FUNCTION が使用されます。指定した (またはデフォルト

## CREATE FUNCTION (外部スカラー)

の) *group-name* が、参照された構造タイプに定義されていない場合、エラーになります (SQLSTATE 42741)。指定した *group-name* または構造タイプに必須の FROM SQL または TO SQL トランスフォーム関数が定義されていない場合には、エラーになります (SQLSTATE 42744)。

トランスフォーム関数は、FROM SQL および TO SQL の両方とも、指定された場合も暗黙的に指定されている場合でも、構造タイプと組み込みタイプ属性とのトランスフォームを適切に実行する SQL 関数でなければなりません。

### PREDICATES

述部でこの関数が使用されるときに実行される、フィルター操作や索引拡張の活用を定義します。述部仕様では、検索条件のオプションの SELECTIVITY 文節を指定できます。PREDICATES 文節が指定された場合、関数は NO EXTERNAL ACTION を指定した DETERMINISTIC として定義しなければなりません (SQLSTATE 42613)。

### WHEN *comparison-operator*

比較演算子 ("=", "<", ">", ">=", "<=", "<>") を使用した述部での、関数の特定の使用を導入します。

#### *constant*

関数の RETURNS タイプに比較可能なデータ・タイプを使用して、定数値を指定します (SQLSTATE 42818)。述部が同じ比較演算子とこの定数でこの関数を使用する場合、指定されたフィルターおよび索引の活用がオプティマイザーにより考慮されます。

### EXPRESSION AS *expression-name*

式に名前を提供します。述部が同じ比較演算子と式でこの関数を使用する場合、指定されたフィルターおよび索引の活用が行われます。この式には、式名が割り当てられ、検索関数の引き数として使用できるようになっています。 *expression-name* は、作成されている関数のいずれかの *parameter-name* と同じにすることはできません (SQLSTATE 42711)。式が指定される際に、その式のタイプが識別されます。

### FILTER USING

結果表をさらにフィルター操作する際に使用する、外部関数またはケース式の指定を許可します。

#### *function-invocation*

結果表の追加のフィルター操作の実行に使用できるフィルター関数を指定します。これは定義された関数のバージョンであり (述部で使用)、ユーザー定義述部で実行される行数を減らし、行を限定するかどうかを判別します。索引により生成される結果が、ユーザー定義述部に期待される結果に近い場合には、フィルター関数を適用する効果はあまりありません。これを指定しない場合は、データのフィルター操作は実行されません。



## CREATE FUNCTION (外部スカラー)

この関数は、任意の *parameter-name*、*expression-name*、または定数を引き数として使用でき (SQLSTATE 42703)、整数を戻します (SQLSTATE 428E4)。戻り値 1 の場合は行が保持され、その他の場合は破棄されます。

この関数は、以下の要件を満たしていなければなりません。

- LANGUAGE SQL で定義されていなければなりません (SQLSTATE 429B4)。
- NOT DETERMINISTIC または EXTERNAL ACTION で定義してはなりません (SQLSTATE 42845)。
- 任意のパラメーターのデータ・タイプとして構造化データ・タイプがあってはなりません (SQLSTATE 428E3)。
- 副照会が含まれてはなりません (SQLSTATE 428E4)。

引き数が他の関数またはメソッドを呼び出す場合、このネストされた関数またはメソッドにもこれらの 4 つの規則が課されます。ただし、引き数が組み込みデータ・タイプに評価されるかぎり、システム生成の *observer* メソッドをフィルター関数 (または、引き数として使用される任意の関数またはメソッド) への引き数として使用することができます。

関数の定義者は、指定されたフィルター関数に対して EXECUTE 特権を持っていなければなりません。

### *case-expression*

結果表をさらにフィルター操作するためのケース式を指定します。

*searched-when-clause* および *simple-when-clause* では、*parameter-name*、*expression-name*、または定数を使用できます (SQLSTATE 42703)。

FILTER USING *function-invocation* に指定された規則を使って、外部関数を結果式として使用することができます。 *case-expression* で参照される関数またはメソッドはすべて、*function-invocation* にリストされている 4 つの規則に適合することも必要です。

*case-expression* の中では副照会は使用できません (SQLSTATE 428E4)。

ケース式は整数を戻さなければなりません (SQLSTATE 428E4)。結果式で戻り値が 1 の場合は行が保持され、その他の場合は破棄されます。

### *index-exploitation*

索引を活用するために使用する索引拡張の検索メソッドによって、規則のセットを定義します。

### SEARCH BY INDEX EXTENSION *index-extension-name*

索引拡張を指定します。 *index-extension-name* は、既存の索引拡張を指定する必要があります。

### EXACT

述部評価の時に索引検索が厳密に行われることを指定します。索引検索後、オリジナルのユーザー定義の述部関数も、フィルターも適用する必要

## CREATE FUNCTION (外部スカラー)

がないことを DB2 に指示するのに EXACT を使用します。EXACT 述部は、索引検索が述部と同じ結果を戻す場合に便利です。

EXACT が指定されない場合には、索引検索後、オリジナルのユーザー定義述部が適用されます。索引が類似した述部を提供するのにとどまると思われる場合には、EXACT オプションは指定しないでください。

索引検索が使用されない場合には、フィルター関数とオリジナルの述部を適用する必要があります。

### *exploitation-rule*

検索ターゲットおよび検索引き数を記述し、さらにこれらを使用して索引拡張で定義した検索メソッドを介して索引検索を実行する方法を記述します。

### **WHEN KEY** (*parameter-name1*)

検索ターゲットを定義します。1つのキーにつき1つしか、探索ターゲットを指定できません。*parameter-name1* 値は、定義された関数のパラメーター名を指定します (SQLSTATE 42703 または 428E8)。

データ・タイプ *parameter-name1* は、索引拡張で指定したソース・キーのデータ・タイプに適合しなければなりません (SQLSTATE 428EY)。この適合は、組み込みおよび特殊データ・タイプで厳密に一致しなければならず、構造タイプの同じタイプ階層内になければなりません。

指定されたパラメーターの値が、指定された索引拡張に基づく索引により網羅される列である場合、この文節は真となります。

### **USE** *search-method-name*(*parameter-name2*,...)

検索引き数を定義します。索引拡張で定義されている検索メソッドから、使用する検索メソッドを指定します。*search-method-name* は、索引拡張で定義される検索メソッドと適合しなければなりません (SQLSTATE 42743)。*parameter-name2* 値は、定義された関数のパラメーター名、または EXPRESSION AS 文節の *expression-name* を指定します (SQLSTATE 42703)。これは、検索ターゲットに指定したパラメーター名と異ならないければなりません (SQLSTATE 428E9)。パラメーターの数と各 *parameter-name2* のデータ・タイプは、索引拡張の検索メソッドに定義されるパラメーターに適合しなければなりません (SQLSTATE 42816)。この適合は、組み込みおよび特殊データ・タイプで厳密に一致しなければならず、構造タイプの同じタイプ階層内になければなりません。

### 注:

#### • 互換性

- DB2 UDB (OS/390 および z/OS 版) との互換性:
  - 以下の構文はデフォルトの振る舞いとして受け入れられます。
    - ASUTIME NO LIMIT
    - NO COLLID

- PROGRAM TYPE SUB
- STAY RESIDENT NO
- 以前のバージョンの DB2 との互換性:
  - PARAMETER STYLE SQL の代わりに PARAMETER STYLE DB2SQL を指定できます。
  - DETERMINISTIC の代わりに NOT VARIANT を、また NOT DETERMINISTIC の代わりに VARIANT を指定することができます。
  - CALLED ON NULL INPUT の代わりに NULL CALL を、また RETURNS NULL ON NULL INPUT の代わりに NOT NULL CALL を指定できます。
- あるデータ・タイプが他のデータ・タイプにキャスト可能かどうかの判別では、CHAR や DECIMAL などのパラメーター化データ・タイプの長さまたは精度と位取りは考慮されません。したがって、ソース・データ・タイプの値をターゲット・データ・タイプの値にキャストしようとする、関数の使用時にエラーになる可能性があります。たとえば、VARCHAR は DATE にキャストできますが、実際にはソース・タイプが VARCHAR(5) と定義されている場合には、関数の使用時にエラーになります。
- ユーザー定義関数のパラメーターのデータ・タイプを選択する場合は、入力値に影響を与えるプロモーションの規則を考慮してください (『データ・タイプのプロモーション』を参照してください)。たとえば、入力値として使用できる定数のデータ・タイプは、予期される以外の組み込みデータ・タイプである可能性があり、さらには、予期されるデータ・タイプにプロモートできない場合があります。プロモーションの規則にしたがって、一般にパラメーターには以下のデータ・タイプを使用するようにしてください。
  - SMALLINT ではなく INTEGER
  - REAL ではなく DOUBLE
  - CHAR ではなく VARCHAR
  - GRAPHIC ではなく VARGRAPHIC
- プラットフォーム間での UDF の移植性を保つためには、以下のデータ・タイプは使用しないようにする必要があります。
  - FLOAT- 代わりに DOUBLE または REAL を使用してください。
  - NUMERIC- 代わりに DECIMAL を使用してください。
  - LONG VARCHAR- 代わりに CLOB (または BLOB) を使用してください。
- 関数とメソッドは、オーバーライド関係にはなりません (SQLSTATE 42745)。オーバーライドについての詳細は、『CREATE TYPE (構造化)』を参照してください。
- 関数のシグニチャーは、メソッドのシグニチャーと同じではありません (関数の最初の *parameter-type* と、メソッドの *subject-type* を比較) (SQLSTATE 42723)。

## CREATE FUNCTION (外部スカラー)

- まだ存在していないスキーマ名を用いて関数を作成すると、ステートメントの許可 ID に `IMPLICIT_SCHEMA` 権限がある場合に限り、そのスキーマが暗黙に作成されます。そのスキーマの所有者は `SYSIBM` になります。スキーマに対する `CREATEIN` 特権が `PUBLIC` に付与されます。
- 索引拡張を定義するには、`NO SQL` として定義されたルーチンしか使用できません (`SQLSTATE 428F8`)。
- 関数が `SQL` を許可する場合、外部プログラムは、連合オブジェクトへのアクセスを試行してはなりません (`SQLSTATE 55047`)。

### 表アクセスの制限

関数が `READS SQL DATA` に定義されている場合には、関数のいかなるステートメントも、関数を呼び出したステートメントによって変更されている表にはアクセスできません (`SQLSTATE 57053`)。たとえば、ユーザー定義関数 `BONUS()` が `READS SQL DATA` に定義されているとします。ステートメント `UPDATE EMPLOYEE SET SALARY = SALARY + BONUS(EMPNO)` が呼び出される場合、`BONUS` 関数の `SQL` ステートメントは、`EMPLOYEE` 表からは読み取りは行えません。

### 特権

- 関数の定義者は、関数に対する `EXECUTE WITH GRANT OPTION` 特権と、関数をドロップする権利を常に与えられます。
- 関数を `SQL` ステートメントで使用する時点で、関数の定義者はその関数によって使用されるすべてのパッケージに対して `EXECUTE` 特権を持っていないければなりません。

### 例:

例 1: `Pellow` は、自身の `PELLOW` スキーマに `CENTRE` 関数を登録します。デフォルト値のあるキーワードはデフォルト値を使い、関数特定名はシステムに生成させることにします。

```
CREATE FUNCTION CENTRE (INT, FLOAT)
 RETURNS FLOAT
 EXTERNAL NAME 'mod!middle'
 LANGUAGE C
 PARAMETER STYLE SQL
 DETERMINISTIC
 NO SQL
 NO EXTERNAL ACTION
```

例 2: ここで、`McBride` (`DBADM` 権限を持つ) が `PELLOW` スキーマに別の `CENTRE` 関数を登録し、関数にデータ定義言語でその後使用するための明示的な特定名を付け、すべてのキーワード値を明示的に指定します。また、この関数はスクラッチパッドを使用し、おそらく後続の結果に影響するデータをスクラッチパッドに蓄積します。`DISALLOW PARALLEL` が指定されているので、関数への参照は並列化されず、したがって 1 つのスクラッチパッドを使用して一度限りの初期化と結果の保管が行われます。

```

CREATE FUNCTION PELLOW.CENTRE (FLOAT, FLOAT, FLOAT)
 RETURNS DECIMAL(8,4) CAST FROM FLOAT
 SPECIFIC FOCUS92
 EXTERNAL NAME 'effects!focalpt'
 LANGUAGE C PARAMETER STYLE SQL
 DETERMINISTIC FENCED NOT NULL CALL NO SQL NO EXTERNAL ACTION
 SCRATCHPAD NO FINAL CALL
 DISALLOW PARALLEL

```

例 3: 以下の例は、以下の式を計算する C 言語のユーザー定義関数です。

```
output = 2 * input - 4
```

入力が NULL 値の場合には (そしてその場合のみ)、NULL 値を戻します。これは、CREATE FUNCTION ステートメントで NOT NULL CALL を指定することにより、より簡単に (つまり NULL 値検査を行わずに) 作成することができます。CREATE FUNCTION ステートメントは、以下のとおりです。

```

CREATE FUNCTION ntest1 (SMALLINT)
 RETURNS SMALLINT
 EXTERNAL NAME 'ntest1!nudft1'
 LANGUAGE C PARAMETER STYLE SQL
 DETERMINISTIC NOT FENCED NULL CALL
 NO SQL NO EXTERNAL ACTION

```

プログラム・コードは、以下のとおりです。

```

#include "sqlsystem.h"
/* NUDFT1 IS A USER_DEFINED SCALAR FUNCTION */
/* udft1 accepts smallint input
and produces smallint output
implementing the rule:
if (input is null)
set output = null;
else
set output = 2 * input - 4;
*/
void SQL_API_FN nudft1
(short *input, /* ptr to input arg */
short *output, /* ptr to where result goes */
short *input_ind, /* ptr to input indicator var */
short *output_ind, /* ptr to output indicator var */
char sqlstate[6], /* sqlstate, allows for null-term */
char fname[28], /* fully qual func name, nul-term */
char finst[19], /* func specific name, null-term */
char msgtext[71]) /* msg text buffer, null-term */
{
/* first test for null input */
if (*input_ind == -1)
{
/* input is null, likewise output */
*output_ind = -1;
}
else
{

```

## CREATE FUNCTION (外部スカラー)

```
/* input is not null. set output to 2*input-4 */
*output = 2 * (*input) - 4;
/* and set out null indicator to zero */
*output_ind = 0;
}
/* signal successful completion by leaving sqlstate as is */
/* and exit */
return;
}
/* end of UDF: NUDFT1 */
```

例 4: 以下の例では、ストリングの中で最初に現れる母音の位置を戻す Java UDF を登録します。UDF は Java で書かれており、隔離して実行されるクラス javaUDFs の findvwl メソッドです。

```
CREATE FUNCTION findv (CLOB(100K))
RETURNS INTEGER
FENCED
LANGUAGE JAVA
PARAMETER STYLE JAVA
EXTERNAL NAME 'javaUDFs.findvwl'
NO EXTERNAL ACTION
CALLED ON NULL INPUT
DETERMINISTIC
NO SQL
```

例 5: この例では、タイプ SHAPE の 2 つのパラメーター g1 および g2 を入力として取るユーザー定義述部 WITHIN を概説します。

```
CREATE FUNCTION within (g1 SHAPE, g2 SHAPE)
RETURNS INTEGER
LANGUAGE C
PARAMETER STYLE SQL
NOT VARIANT
NOT FENCED
NO SQL
NO EXTERNAL ACTION
EXTERNAL NAME 'db2sefn!SDEspatialRelations'
PREDICATES
WHEN = 1
FILTER USING mbrOverlap(g1..xmin, g1..ymin, g1..xmax, g1..ymax,
g2..xmin, g2..ymin, g2..xmax, g2..ymax)
SEARCH BY INDEX EXTENSION gridIndex
WHEN KEY(g1) USE withinExplRule(g2)
WHEN KEY(g2) USE withinExplRule(g1)
```

WITHIN 関数の記述は、任意のユーザー定義の関数の記述に類似しているものの、以下を追加することにより、この関数がユーザー定義の述部で使用できることを指定することができます。

- **PREDICATES WHEN = 1** は、DML ステートメントの WHERE 文節でこの関数が  
within(g1, g2) = 1

と表されるときに、述部はユーザー定義の述部として扱われ、索引拡張 *gridIndex* で定義される索引は、この述部に適合する行を検索するのに使用されるように指定します。定数が指定される場合には、DML ステートメントで指定される定数は、索引の作成ステートメントで指定される定数と完全に一致していなければなりません。この条件は、主に、結果タイプが 1 または 0 のいずれかになるブール式に対応するように提供されています。他の場合には、EXPRESSION 文節を選択するとよいでしょう。

- **FILTER USING mbrOverlap** は、フィルター関数 *mbrOverlap* を参照します。これは、WITHIN 述部の低コスト・バージョンです。上の例では、*mbrOverlap* 関数は入力として最小の境界長方形を使用し、これらが重なるかどうかを素早く判別します。2 つの入力の形の最小の境界長方形が重ならない場合、*g1* が *g2* に含まれることはありません。このようにして、タプルを安全に廃棄でき、コストの高い WITHIN 述部のアプリケーションを避けることができます。
- **SEARCH BY INDEX EXTENSION** 文節は、索引拡張と検索ターゲットの組み合わせをこのユーザー定義の述部で使用できることを指定します。

例 6: この例では、タイプ POINT の 2 つのパラメーター P1 および P2 を入力として取るユーザー定義述部 DISTANCE を概説します。

```
CREATE FUNCTION distance (P1 POINT, P2 POINT)
 RETURNS INTEGER
 LANGUAGE C
 PARAMETER STYLE SQL
 NOT VARIANT
 NOT FENCED
 NO SQL
 NO EXTERNAL ACTION
 EXTERNAL NAME 'db2sefn!SDEDistances'
 PREDICATES
 WHEN > EXPRESSION AS distExpr
 SEARCH BY INDEX EXTENSION gridIndex
 WHEN KEY(P1) USE distanceGrRule(P2, distExpr)
 WHEN KEY(P2) USE distanceGrRule(P1, distExpr)
```

DISTANCE 関数の記述は、任意のユーザー定義関数の記述に類似しているものの、以下の追加により、この関数が述部で使用される場合に、この述部がユーザー定義述部であることを指定します。

- **PREDICATES WHEN > EXPRESSION AS distExpr** も、有効な述部指定です。WHEN 文節で式が指定されると、この述部が DML ステートメントのユーザー定義述部であるかどうかを判別するために、この式の結果タイプが使用されます。たとえば、以下ようになります。

```
SELECT T1.C1
 FROM T1, T2
 WHERE distance (T1.P1, T2.P1) > T2.C2
```

述部指定 *distance* は、2 つのパラメーターを入力として使用し、タイプ INTEGER の T2.C2 を使用して結果を比較します。(特定の定数を使用する場合は異なり) 式

## CREATE FUNCTION (外部スカラー)

の右辺のデータ・タイプのみ問題となるため、CREATE FUNCTION DDL にある EXPRESSION 文節を選択して、比較値としてワイルドカードを指定するとよいでしょう。

別の方法として、以下のものも有効なユーザー定義述部です。

```
SELECT T1.C1
FROM T1, T2
WHERE distance(T1.P1, T2.P1) > distance (T1.P2, T2.P2)
```

現在のところ、右辺しか式として扱われないという制限があります。左辺の項は、ユーザー定義述部用のユーザー定義関数です。

- **SEARCH BY INDEX EXTENSION** 文節は、索引拡張と検索ターゲットの組み合わせをこのユーザー定義の述部に使用できることを指定します。distance 関数の場合、distExpr として指定された式も範囲生成関数 (索引拡張の一部として定義) に渡される検索引き数の 1 つです。式の ID は、式の名前を定義するのに使用され、引き数として範囲生成関数に渡されます。

### 関連資料:

- SQL リファレンス 第 1 巻 の『基本述部』
- 1441 ページの『CREATE TYPE (構造化)』
- 1552 ページの『CREATE FUNCTION (SQL スカラー、表、または行)』
- 1360 ページの『ルーチンで使用可能な SQL ステートメント』
- SQL リファレンス 第 1 巻 の『特殊レジスター』
- SQL リファレンス 第 1 巻 の『データ・タイプのプロモーション』
- SQL リファレンス 第 1 巻 の『データ・タイプ間のキャスト』

## CREATE FUNCTION (外部表)

このステートメントは、ユーザー定義の外部表関数をアプリケーション・サーバーに登録する場合に使用されます。

表関数 は、SELECT の FROM 文節で使用することができ、行を一度に 1 行戻すことによって、SELECT に表を戻します。

### 呼び出し:

このステートメントはアプリケーション・プログラムに組み込むことができ、また動的 SQL ステートメントを使用して出すことができます。DYNAMICRULES の実行動作がパッケージに効力を持つ場合にのみ、動的に準備できる実行可能ステートメントです (SQLSTATE 42509)。

### 許可:



ステートメントの許可 ID によって保持されている特権には、少なくとも以下のいずれかが含まれていなければなりません。

- SYSADM または DBADM 権限
- データベースに対する CREATE\_EXTERNAL\_ROUTINE 権限、および以下の少なくとも 1 つ。
  - データベースに対する IMPLICIT\_SCHEMA 権限 (関数の暗黙または明示のスキーマ名が存在しない場合)
  - スキーマに対する CREATEIN 特権 (関数のスキーマ名が存在する場合)

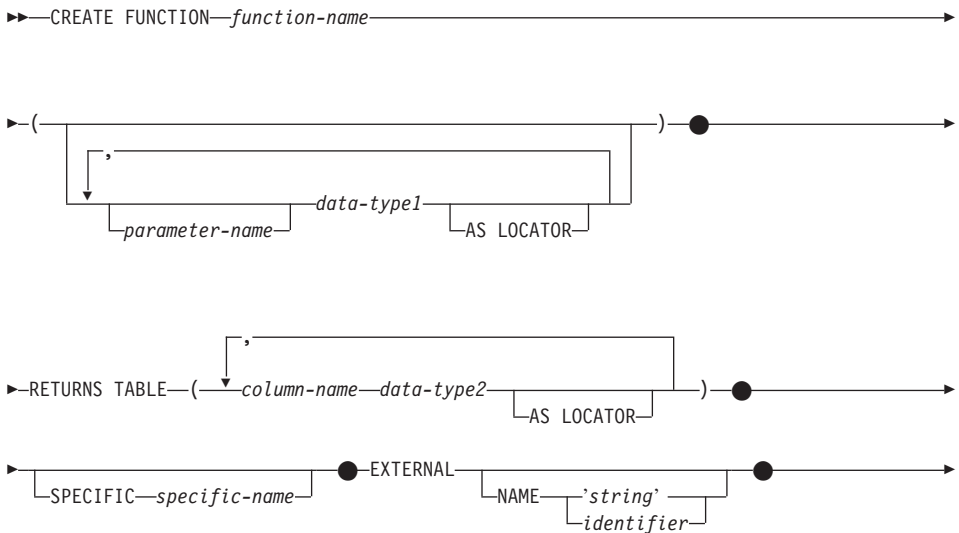
非隔離の関数を作成するには、ステートメントの許可 ID の特権に、以下の特権の少なくとも 1 つが含まれている必要があります。

- データベースに対する CREATE\_NOT\_FENCED\_ROUTINE 権限
- SYSADM または DBADM 権限

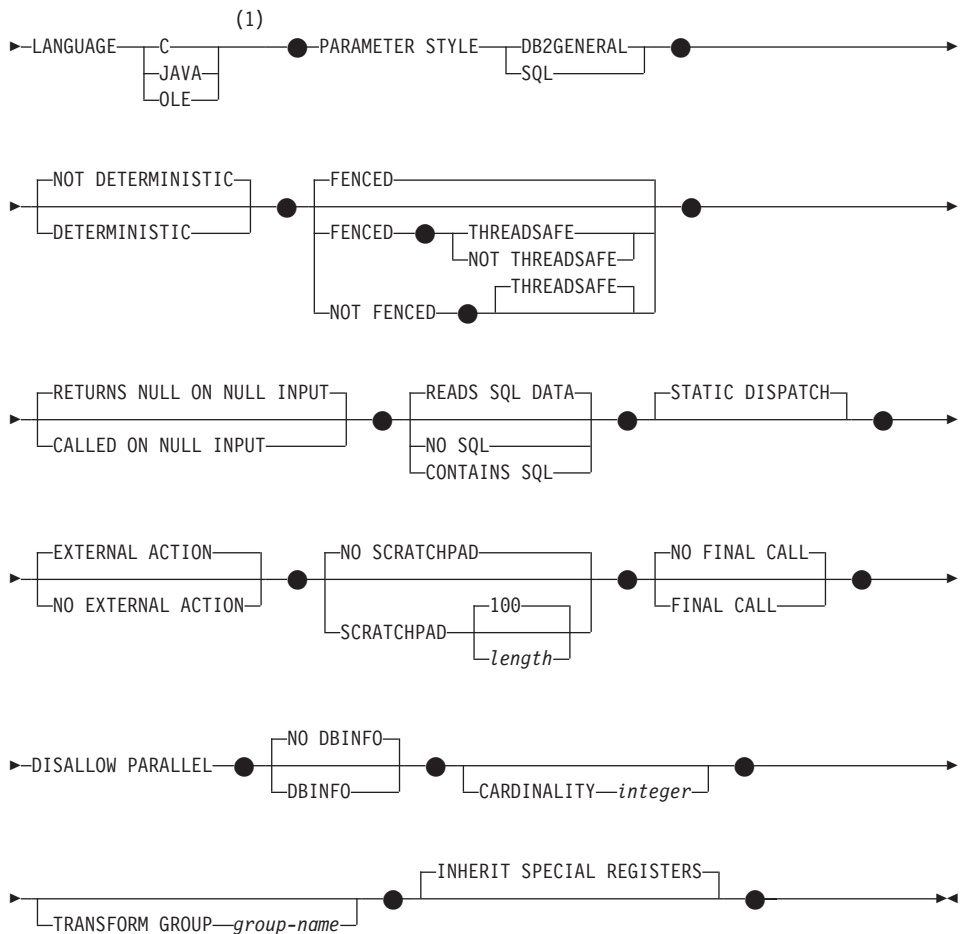
隔離関数を作成する場合には、さらに別の権限や特権は必要ありません。

許可 ID の権限が不十分で、操作を実行できない場合には、エラー (SQLSTATE 42502) になります。

**構文:**



## CREATE FUNCTION (外部表)



### 注:

- 1 LANGUAGE OLE DB 外部表関数の作成の詳細は、『CREATE FUNCTION (OLE DB 外部表)』を参照してください。LANGUAGE SQL 表関数の作成の詳細は、『CREATE FUNCTION (SQL スカラー、表または行)』を参照してください。

### 説明:

#### *function-name*

定義する関数の名前を指定します。これは、関数を指定する修飾または非修飾の名前です。*function-name* (関数名) の非修飾形式は SQL ID です (最大長 18)。動的 SQL ステートメントでは、CURRENT SCHEMA 特殊レジスターが、修飾子のないオブジェクト名の修飾子として使用されます。静的 SQL ステートメントでは、QUALIFIER プリコンパイル/バインド・オプションにより、修飾子のないオブジェ

クト名の修飾子が暗黙指定されます。修飾形式は、*schema-name* の後にピリオドと SQL ID が続きます。最初のパラメーターが構造タイプの場合、修飾名は、最初のパラメーターのデータ・タイプと同じであってはなりません。

暗黙または明示の修飾子を含む名前、およびパラメーターの数と各パラメーターのデータ・タイプ (データ・タイプの長さ、精度、または位取りの各属性には関係なく) は、カタログに記述されている関数を指定するものであってはなりません (SQLSTATE 42723)。非修飾名とパラメーターの数およびデータ・タイプとの組み合わせは、そのスキーマ内では当然固有ですが、複数のスキーマ間で固有である必要はありません。

2 つの部分からなる名前を指定する場合、'SYS' で始まる *schema-name* (スキーマ名) は使用できません (SQLSTATE 42939)。

述部のキーワードとして使用されるいくつかの名前は、システム使用に予約されており、*function-name* として使用することはできません (SQLSTATE 42939)。それらの名前は、SOME、ANY、ALL、NOT、AND、OR、BETWEEN、NULL、LIKE、EXISTS、IN、UNIQUE、OVERLAPS、SIMILAR、MATCH、および比較演算子です。

関数のシグニチャーに何らかの差異があれば、同じ名前を複数の関数に使用することができます。禁止されてはいませんが、外部ユーザー定義関数の名前として、組み込み関数と同じ名前を指定すべきではありません。

#### *parameter-name*

パラメーターのオプション名を指定します。これは、この関数の他のパラメーターすべての名前と異なる名前にする必要があります。

#### *(data-type1,...)*

関数の入力パラメーターの数を指定するとともに、各パラメーターのデータ・タイプを指定します。このリストには、関数が受け取ることを予期している各パラメーターごとに 1 つの項目を指定する必要があります。パラメーターの数は 90 を超えることはできません。この限界を超えると、エラー (SQLSTATE 54023) になります。

パラメーターのない関数も登録可能です。この場合、指定するデータ・タイプがない場合でも、括弧はコーディングする必要があります。たとえば、以下のようになります。

```
CREATE FUNCTION WOOFER() ...
```

その対応するすべてのパラメーターのタイプがまったく同じである場合でも、1 つのスキーマ中に名前が同じ 2 つの関数があってはなりません。このタイプの比較では長さ、精度、および位取りは考慮されません。したがって、CHAR(8) と CHAR(35)、また DECIMAL(11,2) と DECIMAL (4,3) は、それぞれ同じタイプと見なされます。ユニコード・データベースの場合には、CHAR(13) と GRAPHIC(8) は、それぞれ同じタイプと見なされます。さらに、DECIMAL と NUMERIC など

## CREATE FUNCTION (外部表)

のように、この目的で複数のタイプが同じタイプとして扱われることがあります。シングニチャーが重複していると、SQL エラー (SQLSTATE 42723) になります。

たとえば、以下のステートメントの場合、

```
CREATE FUNCTION PART (INT, CHAR(15)) ...
CREATE FUNCTION PART (INTEGER, CHAR(40)) ...

CREATE FUNCTION ANGLE (DECIMAL(12,2)) ...
CREATE FUNCTION ANGLE (DEC(10,7)) ...
```

2 番目と 4 番目のステートメントは、重複する関数と見なされ、エラーになります。

### *data-type1*

パラメーターのデータ・タイプを指定します。

- CREATE TABLE ステートメントの *data-type* の定義で指定が可能で、関数の作成に使用されている言語において対応するものがある SQL データ・タイプ仕様と省略形を指定できます。
- DECIMAL (および NUMERIC) は、LANGUAGE C と OLE では無効です (SQLSTATE 42815)。
- REF(*type-name*) は、パラメーターのデータ・タイプとして指定できます。ただし、パラメーターに有効範囲を指定してはなりません (SQLSTATE 42997)。
- 適切なトランスフォーム関数が、関連するトランスフォーム・グループに存在する場合には、構造タイプを指定できます。

### AS LOCATOR

LOB タイプまたは LOB タイプに基づく特殊タイプの場合、AS LOCATOR 文節を追加することができます。これは、実際の値の代わりに LOB ロケーターを UDF に渡すことを指定します。これにより、UDF に渡すバイト数を大幅に減らすことができ、パフォーマンスも向上します (特に、UDF にとって実際に必要になる値が数バイトだけである場合)。

以下の例は、パラメーター定義の AS LOCATOR 文節の使用法を示しています。

```
CREATE FUNCTION foo (CLOB(10M) AS LOCATOR, IMAGE AS LOCATOR)
...
```

ここで、IMAGE は LOB タイプの 1 つに基づく特殊タイプであると想定します。

また、引き数のプロモーション目的には、AS LOCATOR 文節の効果はないことに注意してください。この例では、タイプはそれぞれ CLOB と IMAGE であると見なされるので、関数に CHAR 引き数または VARCHAR 引き数が最初の引き数として渡されます。同様に、関数シングニ

チャーに対して AS LOCATOR の効果はありません。関数シグニチャーは、(a) "関数解決" と呼ばれるプロセスによって DML で参照された場合、(b) COMMENT ON や DROP などの DDL ステートメントで参照された場合に関数をマッチングする際に使用されます。実際に、この文節はシグニチャーの指定のない COMMENT ON や DROP で使用しても、しなくても構いません。

LOB または LOB に基づく特殊タイプ以外のタイプに対して AS LOCATOR を指定すると、エラー (SQLSTATE 42601) が発生します。

関数が FENCED で NO SQL オプションを持っている場合、AS LOCATOR 文節は指定できません (SQLSTATE 42613)。

## RETURNS TABLE

関数の出力が表であることを指定します。このキーワードに続く括弧は、表の列の名前とタイプのリストを区切るもので、他の指定 (たとえば、制約) のない単純な CREATE TABLE ステートメントの形式と類似しています。255 列以内が許可されます (SQLSTATE 54011)。

### *column-name*

この列の名前を指定します。名前を修飾することはできず、表の複数の列に対して同じ名前を使用することはできません。

### *data-type2*

列のデータ・タイプを指定します。構造タイプ以外であれば、特定の言語において、UDF 作成のパラメーターとしてサポートされるどのようなデータ・タイプでも構いません (SQLSTATE 42997)。

## AS LOCATOR

*data-type2* が LOB タイプまたは LOB タイプに基づく特殊タイプの場合、このオプションを使用すると、関数は結果表でインスタンス化される LOB 値のロケーターを戻します。

この文節で使用できる有効なタイプについては、『CREATE FUNCTION (外部スカラー)』で説明されています。

## SPECIFIC *specific-name*

定義する関数のインスタンスに対する固有名を指定します。この特定名は、この関数をソース関数として使用する場合、この関数をドロップする場合、またはこの関数にコメントを付ける場合に使用することができます。これは、関数の呼び出しには使用できません。*specific-name* (関数名) の非修飾形式は SQL ID です (最大長 18)。修飾形式は、*schema-name* の後にピリオドと SQL ID が続きます。暗黙または明示の修飾子も含め、その名前が、アプリケーション・サーバーに存在する別の関数インスタンスを指定するものであってはなりません。そうでない場合、エラー (SQLSTATE 42710) になります。

*specific-name* は、既存の *function-name* (関数名) と同じでも構いません。

## CREATE FUNCTION (外部表)

修飾子を指定しない場合、*function-name* に使用された修飾子が使用されます。修飾子を指定する場合は、*function-name* の明示修飾子または暗黙修飾子と同じでなければなりません。そうでない場合、エラー (SQLSTATE 42882) になります。

*specific-name* の指定がない場合、固有の名前がデータベース・マネージャーによって生成されます。生成される固有の名は、SQL の後に文字のタイム・スタンプが続く名前です (SQLyymmddhhmmssxxx)。

### EXTERNAL

この文節は、外部プログラミング言語で作成され、文書化されたリンケージの規則とインターフェースに準拠している新しい関数を登録するのに、CREATE FUNCTION ステートメントが使用されていることを示します。

NAME 文節を指定しない場合、"NAME *function-name*" が想定されます。

### NAME 'string'

この文節は、定義する関数をインプリメントするためのユーザー作成コードを指定します。

'string' オプションは、最大 254 文字のストリング定数です。ストリングに使用される形式は、指定した LANGUAGE によって異なります。

#### • LANGUAGE C の場合

指定する *string* (ストリング) は、作成しているユーザー定義関数を実行するためにデータベース・マネージャーが呼び出すライブラリー名と、そのライブラリー中の関数名です。ライブラリー (およびそのライブラリー中の関数) は、CREATE FUNCTION ステートメントの実行時に存在している必要はありません。ただし、関数が SQL ステートメントで使用される時点で、そのライブラリーとそのライブラリー内の関数が存在していなければならず、しかもデータベース・サーバーのマシンからアクセス可能でなければなりません。

```
▶▶ ' library_id _____ '
 └─ absolute_path_id ─┘ └─!─ func_id ─┘
```

単一引用符内に、余分なブランクを使用することはできません。

#### *library\_id*

関数を含むライブラリー名を指定します。データベース・マネージャーは、.../sqllib/function ディレクトリー (UNIX 系システム)、または ...¥instance\_name¥function ディレクトリー (DB2INSTPROF レジストリー変数で指定した Windows 32 ビット・オペレーティング・システム) のライブラリーを探します。ここで、データベース・マネージャーの実行に使用される制御 sqllib ディレクトリーを、データベース・マネージャーが検索します。たとえば、UNIX 系システムの制御 sqllib ディレクトリーは、/u/\$DB2INSTANCE/sqllib です。

UNIX 系システムの *library\_id* が 'myfunc' で、データベース・マネージャーが /u/production から実行されている場合、ライブラリー /u/production/sqllib/function/myfunc から関数を探索します。

Windows 32 ビット・オペレーティング・システム の場合、*library\_id* が関数ディレクトリーにないと、データベース・マネージャーは LIBPATH または PATH を探します。

#### *absolute\_path\_id*

関数の絶対パス名を指定します。

たとえば、UNIX 系システムで '/u/jchui/mylib/myfunc' を指定すると、データベース・マネージャーは /u/jchui/mylib の中から myfunc 関数を探します。

Windows の場合、'd:¥mylib¥myfunc' を指定すると、データベース・マネージャーは d:¥mylib ディレクトリーから myfunc.dll ファイルをロードします。

**注:** Windows オペレーティング・システムで C または C++ のルーチンを登録しているとき、EXTERNAL NAME 文節内でルーチン本体識別しているときに、次のことに注意してください。絶対パス ID がルーチン本体の識別に使用されている場合は、.dll 拡張子を必ず付加してください。

#### *! func\_id*

呼び出される関数の入り口点名を指定します。! は、ライブラリー ID と関数 ID との間の区切り文字です。! *func\_id* を省略すると、データベース・マネージャーはライブラリーのリンク時に確立されたデフォルトの入り口点を使用します。

たとえば、UNIX 系システムで 'mymod!func8' と指定すると、データベース・マネージャーはライブラリー

\$inst\_home\_dir/sqllib/function/mymod を調べて、そのライブラリー内の入り口点 func8 を使用します。

Windows 32 ビット・オペレーティング・システム の場合 'mymod!func8' を指定すると、データベース・マネージャーは mymod.dll ファイルをロードして、そのダイナミック・リンク・ライブラリー (DLL) の func8() 関数を呼び出します。

ストリングの形式が正しくない場合には、エラー (SQLSTATE 42878) になります。

いずれの場合も、すべての外部関数の本体は、データベースのすべてのパーティションで使用可能なディレクトリーにある必要があります。

- LANGUAGE JAVA の場合

## CREATE FUNCTION (外部表)

指定する *string* には、作成中のユーザー定義関数を実行するためにデータベース・マネージャーが呼び出す、任意指定の jar ファイル、クラス ID、およびメソッド ID が含まれています。クラス ID とメソッド ID は、CREATE FUNCTION ステートメントの実行時には存在している必要はありません。jar\_id を指定する場合、ID は、CREATE FUNCTION ステートメントの実行時に存在していなければなりません。ただし、関数を SQL ステートメントで使用する時点で、メソッド ID は存在しなければならず、データベース・サーバーのマシンからアクセス可能でなければなりません。

→ ' [ jar\_id : ] class\_id [ ! ] method\_id ' →

単一引用符内に、余分なブランクを使用することはできません。

### jar\_id

jar の集合をデータベースヘインストールしたときに、その jar の集合に付けられた jar ID を指定します。これは、単純 ID またはスキーマ修飾 ID のいずれかにすることができます。たとえば、'myJar' や 'mySchema.myJar' のようになります。

### class\_id

Java オブジェクトのクラス ID を指定します。クラスがパッケージの一部である場合、クラス ID の部分に完全なパッケージ接頭部 (例: 'myPacks.UserFuncs') が含まれている必要があります。Java 仮想マシンは、ディレクトリー '.../myPacks/UserFuncs/' 中のクラスを探します。Windows 32 ビット・オペレーティング・システムでは、Java 仮想マシンはディレクトリー '...¥myPacks¥UserFuncs¥' を探索します。

### method\_id

呼び出す Java オブジェクトのメソッド名を指定します。

#### • LANGUAGE OLE の場合

指定する *string* は、作成中のユーザー定義関数を実行するためにデータベース・マネージャーが呼び出す、OLE のプログラム ID (progid) またはクラス ID (clsid)、およびメソッド ID です。プログラム ID またはクラス ID、およびメソッド ID は、CREATE FUNCTION ステートメントの実行時に存在している必要はありません。ただし、関数を SQL ステートメントで使用する時点で、メソッド ID は存在していなければならず、データベース・サーバーのマシンからアクセス可能でなければなりません。そうでない場合、エラー (SQLSTATE 42724) になります。

→ ' [ progid ] ! [ method\_id ] ' →  
    └── [ clsid ] ─┘

単一引用符内に、余分なブランクを使用することはできません。

### progid

OLE オブジェクトのプログラム ID を指定します。



*progid* は、データベース・マネージャーには解釈されず、実行時に OLE API に転送されるだけです。指定する OLE オブジェクトは、作成可能である必要があり、実行時バインディング (ディスパッチに基づくバインディングとも呼ばれる) をサポートしている必要があります。

#### *clsid*

作成する OLE オブジェクトのクラス ID を指定します。OLE オブジェクトが *progid* を指定して登録されていない場合に、*progid* を指定する代わりに使用することができます。*clsid* の形式は以下のとおりです。

```
{nnnnnnnnnn-nnnnn-nnnnn-nnnnnnnnnnnnnnnnn}
```

ここで 'n' は英数字です。*clsid* は、データベース・マネージャーには解釈されず、実行時に OLE API に転送されるだけです。

#### *method\_id*

呼び出す OLE オブジェクトのメソッド名を指定します。

### NAME *identifier*

この文節は、定義している関数をインプリメントするユーザー作成コードの名前を指定します。指定する *identifier* は SQL ID です。SQL ID は、ストリングの *library-id* として使用されます。区切られた ID でない場合、ID は大文字に変換されます。ID がスキーマ名で修飾されている場合、スキーマ名の部分は無視されます。この形式の NAME は、LANGUAGE C でのみ使用可能です。

### LANGUAGE

この文節は必須で、ユーザー定義関数の本体が準拠している言語インターフェース規則を指定するのに使用します。

**C** これは、データベース・マネージャーが、ユーザー定義関数を C の関数であるかのように呼び出すことを意味します。ユーザー定義関数は、標準 ANSI C プロトタイプで定義されている C 言語の呼び出しおよびリンケージの規則に準拠していなければなりません。

**JAVA** データベース・マネージャーは、Java クラスのメソッドとしてユーザー定義関数を呼び出します。

**OLE** データベース・マネージャーは、OLE 自動化オブジェクトによって公開されたメソッドとして、ユーザー定義関数を呼び出します。ユーザー定義関数は、*OLE Automation Programmer's Reference* に説明されている、OLE 自動化データ・タイプと呼び出しメカニズムに準拠している必要があります。

LANGUAGE OLE は、DB2 (Windows 32 ビット・オペレーティング・システム版) で保管されたユーザー定義関数に対してのみサポートされません。

## CREATE FUNCTION (外部表)

LANGUAGE OLE DB 外部表関数の作成の詳細は、『CREATE FUNCTION (OLE DB 外部表)』を参照してください。

### PARAMETER STYLE

この文節は、関数にパラメーターを渡し、関数から値を戻すのに用いる規則を指定するために使用します。

#### DB2GENERAL

Java クラスのメソッドとして定義された外部関数との間で、パラメーターを渡し、値を戻す場合に用いる規則を指定します。これは、LANGUAGE JAVA を使用する場合にだけ指定する必要があります。

DB2GENERAL の同義語として値 DB2GENRL が使用可能です。

#### SQL

C 言語の呼び出しとリンケージの規則に準拠する外部関数との間で、パラメーターを渡し、値を戻すために用いる規則を指定します。これは、LANGUAGE C または LANGUAGE OLE を使用する場合に指定する必要があります。

### DETERMINISTIC または NOT DETERMINISTIC

この文節は任意指定で、特定の引き数の値に対して関数が常に同じ結果を戻すか (DETERMINISTIC)、それとも状態値に依存して関数の結果が影響を受けるか (NOT DETERMINISTIC) を指定します。つまり DETERMINISTIC 関数は、同一の入力で正しい呼び出しが行われたとき、常に同じ表を返します。NOT DETERMINISTIC を指定すると、同じ入力によって常に同じ結果が生じる利点に基づく最適化ができなくなります。NOT DETERMINISTIC 表関数の例として、ファイルなどのデータ・ソースからデータを検索する関数があります。

### FENCED または NOT FENCED

この文節は、関数をデータベース・マネージャーの操作環境のプロセスまたはアドレス・スペースで実行しても“安全”か (NOT FENCED)、そうでないか (FENCED) を指定します。

関数が FENCED として登録されると、データベース・マネージャーは、その内部リソース (データ・バッファーなど) を保護して、その関数からアクセスされないようにします。多くの関数は、FENCED または NOT FENCED のどちらかで実行するように選択することができます。一般に、FENCED として実行される関数は、NOT FENCED として実行されるものと同じようには実行されません。

#### 注意:

適切にコード化、検討、およびテストされていない関数に NOT FENCED を使用すると、DB2 の保全性に危険を招く場合があります。DB2 では、発生する可能性のある一般的な不注意による障害の多くに対して、いくつかの予防措置がとられています。NOT FENCED ユーザー定義関数が使用される場合には、完全な保全性を確保できません。

LANGUAGE OLE または NOT THREADSAFE を指定した関数には、FENCED のみを指定できます (SQLSTATE 42613)。

関数が FENCED で NO SQL オプションを持っている場合、AS LOCATOR 文節は指定できません (SQLSTATE 42613)。

ユーザー定義関数を NOT FENCED として登録するには、SYSADM 権限、DBADM 権限、または特殊権限 (CREATE\_NOT\_FENCED\_ROUTINE) が必要です。

### THREADSAFE または NOT THREADSAFE

関数を他のルーチンと同じプロセスで実行しても安全か (THREADSAFE)、そうでないか (NOT THREADSAFE) を指定します。

関数が OLE 以外の LANGUAGE で定義される場合:

- 関数が THREADSAFE に定義されている場合には、データベース・マネージャは他のルーチンと同じプロセスで関数を呼び出すことができます。一般に、スレッド・セーフにするには、関数はどのグローバルあるいは静的データ域をも使用してはなりません。多くのプログラミング解説書には、スレッド・セーフ・ルーチンの作成に関する説明が含まれています。FENCED および NOT FENCED 関数の両方が THREADSAFE になることが可能です。
- 関数が NOT THREADSAFE に定義される場合には、データベース・マネージャは関数を他のルーチンと同じプロセスに決して呼び出しません。

FENCED 関数の場合、LANGUAGE が JAVA なら THREADSAFE がデフォルトです。これ以外のすべての言語の場合は、NOT THREADSAFE がデフォルトです。関数が LANGUAGE OLE に定義される場合には、THREADSAFE は指定できません (SQLSTATE 42613)。

NOT FENCED 関数の場合には、THREADSAFE がデフォルトです。NOT THREADSAFE を指定することはできません (SQLSTATE 42613)。

### RETURNS NULL ON NULL INPUT または CALLED ON NULL INPUT

このオプション文節を使用すると、引き数のいずれかが NULL 値の場合に、外部関数を呼び出さないようにすることができます。ユーザー定義関数がパラメータなしで定義される場合、この NULL 引き数条件が引き起こされることはなく、この仕様のコーディング方法はそれほど重要ではなくなります。

RETURNS NULL ON NULL INPUT が指定されており、表関数 OPEN が実行されるときに、関数の引き数のいずれかが NULL 値の場合、ユーザー定義関数は呼び出されません。試行した表関数スキャンの結果は、空の表 (行のない表) になります。

CALLED ON NULL INPUT が指定されると、引き数が NULL 値か否かに関係なくユーザー定義関数が呼び出されます。これは、NULL 値を戻す場合も、通常の (NULL 値以外の) 値を戻す場合もあります。ただし、NULL の引き数値の有無のテストは UDF が行う必要があります。

## CREATE FUNCTION (外部表)

値 NULL CALL は、上位互換またはファミリーの互換性のために、CALLED ON NULL INPUT の同義語として使うことができます。同様に、NOT NULL CALL は、RETURNS NULL ON NULL INPUT の同義語として使用できません。

### NO SQL、CONTAINS SQL、READS SQL DATA

関数から SQL ステートメントが発行されるかどうかと、もし発行されればどのタイプかを示します。

#### NO SQL

関数はどの SQL ステートメントも実行できないことを指示します (SQLSTATE 38001)。

#### CONTAINS SQL

SQL データの読み取りも変更も行わない SQL ステートメントを、関数で実行できることを指定します (SQLSTATE 38004 または 42985)。どの関数でもサポートされていないステートメントは、これとは異なるエラーを戻します (SQLSTATE 38003 または 42985)。

#### READS SQL DATA

SQL データを変更しない SQL ステートメントを、関数で実行できることを指定します (SQLSTATE 38002 または 42985)。どの関数でもサポートされていないステートメントは、これとは異なるエラーを戻します (SQLSTATE 38003 または 42985)。

### STATIC DISPATCH

このオプション文節は、問題解決時に DB2 が関数のパラメーターの静的タイプ (宣言済みタイプ) に基づいて関数を選択するよう指示します。

### NO EXTERNAL ACTION または EXTERNAL ACTION

このオプションの文節は、データベース・マネージャーによって管理されていないオブジェクトの状態を変更するアクションを関数が行うかどうかを指定します。EXTERNAL ACTION を指定すると、関数に外部の影響がないことを前提とした最適化ができなくなります。(たとえば、メッセージの送信、警報音による通知、ファイルへのレコードの書き込みなど。)

### NO SCRATCHPAD または SCRATCHPAD *length*

この文節はオプションであり、この外部関数に対してスクラッチパッドを用意するか否かを指定するのに使用することができます。(ユーザー定義関数を再入可能にすることを強くお勧めします。再入可能にすると、スクラッチパッドによってある呼び出しと次の呼び出しとの間に関数が“状態を保管する”手段が用意されます。)

SCRATCHPAD を指定すると、ユーザー定義関数の最初の呼び出し時に、その外部関数によって使用されるスクラッチパッドにメモリーが割り振られます。このスクラッチパッドには、以下の特性があります。

- *length* を指定すると、スクラッチパッドのサイズをバイト単位で設定できます。この値は 1 ~ 32 767 で指定できます (SQLSTATE 42820)。デフォルト値は 100 です。
- すべて 'X'00' に初期化されます。
- その有効範囲は、該当の SQL ステートメントです。SQL ステートメントでの外部関数に対する参照ごとに 1 つのスクラッチパッドがあります。したがって、以下のステートメントの UDFX 関数が、SCRATCHPAD キーワードを使用して定義されると、2 つのスクラッチパッドが割り当てられます。

```
SELECT A.C1, B.C2
FROM TABLE (UDFX(:hv1)) AS A,
TABLE (UDFX(:hv1)) AS B
WHERE ...
```

- スクラッチパッドは持続します。スクラッチパッドは、ステートメントの実行開始時に初期化され、ある呼び出しから次の呼び出しにスクラッチパッドの状態を保存するために、外部表関数で使用することができます。UDF に FINAL CALL キーワードも指定されている場合、DB2 がスクラッチパッドを変更することはありません。また、特殊 FINAL 呼び出しがなされると、スクラッチパッドに固定されていたすべてのリソースが解放されます。

NO FINAL CALL が指定またはデフォルト指定されている場合は、DB2 が OPEN 呼び出しごとにスクラッチパッドを初期化し直すので、外部表関数は CLOSE 呼び出し時に、スクラッチパッドに固定されているすべてのリソースに対して終結処理を行います。FINAL CALL または NO FINAL CALL の判別、およびスクラッチパッドの関連する動作は、重要な考慮事項です。表関数が副照会または結合で使用されるときは、ステートメントの実行中に複数の OPEN 呼び出しが生じ得るので、特に重要です。

- これは、外部関数が獲得するシステム・リソース (メモリーなどの) の中央点として使用することもできます。関数は、最初の呼び出しでメモリーを獲得し、そのアドレスをスクラッチパッドに保管して、後の呼び出しでそれを参照することができます。

(上で概説したように、FINAL CALL/NO FINAL CALL キーワードは、スクラッチパッドの再初期化を制御するために使用され、スクラッチパッドに固定されているリソースを表関数が解放する時期を指示します。)

SCRATCHPAD を指定すると、ユーザー定義関数を呼び出すたびに、スクラッチパッドをアドレッシングする外部関数に追加の引き数が渡されます。

NO SCRATCHPAD を指定すると、外部関数に対してスクラッチパッドは割り振られず、渡されません。

### FINAL CALL または NO FINAL CALL

この文節はオプションであり、外部関数に対する最終呼び出し (および別個の最初の呼び出し) が行われるか否かを指定します。この文節は、スクラッチパッドが再初期化される時期も制御します。NO FINAL CALL が指定されている場合は、

## CREATE FUNCTION (外部表)

DB2 はオープン、取り出しおよびクローズの 3 つのタイプの呼び出ししか行うことができません。しかし、FINAL CALL が指定されている場合は、オープン、取り出しおよびクローズに加えて、表関数に対して最初の呼び出しと最終呼び出しを行うことができます。

外部表関数の場合、どのオプションが選択されたかにかかわらず、呼び出しタイプ引き数は常に存在します。

割り込みかトランザクションの終了のために最終呼び出しが行われると、UDF は CLOSE カーソル以外の SQL ステートメントを発行できません (SQLSTATE 38505)。こうした最終呼び出しの状況の場合には、「呼び出しタイプ」の引き数に特殊値が渡されます。

### DISALLOW PARALLEL

この文節は、関数への単一の参照に対して、関数の呼び出しを並列化できないことを指定します。表関数は常に 1 つのパーティションで実行されます。

### NO DBINFO または DBINFO

この文節はオプションで、DB2 において既知である特定の情報を追加の呼び出し時引き数として UDF に渡すか (DBINFO)、または渡さないか (NO DBINFO) を指定します。NO DBINFO がデフォルト値です。DBINFO は、LANGUAGE OLE ではサポートされません (SQLSTATE 42613)。

DBINFO を指定すると、以下の情報を含む構造が UDF に渡されます。

- データベース名 - 現在接続されているデータベースの名前。
- アプリケーション ID - データベースへの接続ごとに確立された、固有のアプリケーション ID。
- アプリケーション許可 ID - アプリケーション実行時の許可 ID。この UDF とアプリケーションとの間でネストされている UDF は無関係。
- コード・ページ - データベースのコード・ページを識別します。
- スキーマ名 - 外部表関数には適用されません。
- 表名 - 外部表関数には適用されません。
- 列名 - 外部表関数には適用されません。
- データベースのバージョン / リリース - UDF を呼び出すデータベース・サーバーのバージョン、リリースおよび修正レベルを識別します。
- プラットフォーム - サーバーのプラットフォーム・タイプが入ります。
- 表関数の結果の列番号 - この関数を参照する特定のステートメントに実際に必要な、表関数の結果の列番号の配列。表関数の場合に限り、すべての列の値でなく必要な列の値だけを戻すことによって、UDF を最適化することを可能にします。

### CARDINALITY *integer*

この文節はオプションで、関数によって戻されると予想される行数の見積もりを最



適化のために指定します。 *integer* の値の有効範囲は、 0 ～ 9 223 372 036 854 775 807 (両端の値を含む) です。

表関数に対して *CARDINALITY* 文節の指定がない場合、 *DB2* はデフォルト値として有限の値を想定します (*RUNSTATS* ユーティリティが統計を収集していない表に対して想定される値と同じ)。

警告: 関数が事実上無限のカーディナリティーを持っている (すなわち、呼び出されるといつでも行を戻し、"end-of-table" 条件を戻さない) 場合、正しく機能するために "end-of-table" 条件を必要とする照会は無限に実行されるので、照会を中断させる必要があります。このような照会の例として、*GROUP BY* および *ORDER BY* を使用する照会があります。このような *UDF* は書かないことをお勧めします。

### TRANSFORM GROUP *group-name*

関数を呼び出す際のユーザー定義の構造タイプのトランスフォーメーションに使用するトランスフォーム・グループを指定します。関数定義にパラメーター・データ・タイプとしてユーザー定義の構造タイプが含まれている場合、トランスフォームが必要になります。この文節が指定されない場合には、デフォルトのグループ名 *DB2\_FUNCTION* が使用されます。指定した (またはデフォルトの) *group-name* が、参照された構造タイプに定義されていない場合、エラーになります (*SQLSTATE 42741*)。指定した *group-name* または構造タイプに必須の *FROM SQL* 変換関数が定義されていない場合には、エラーになります (*SQLSTATE 42744*)。

### INHERIT SPECIAL REGISTERS

このオプション文節は、関数の更新可能な特殊レジスターが、呼び出しステートメントの環境からの初期値を継承することを指定します。カーソルの *select* ステートメントで呼び出される関数の場合、初期値はカーソルがオープンした際の環境から継承します。ネストされたオブジェクト (たとえば、トリガーまたはビュー) に呼び出されるルーチンの場合、初期値は (オブジェクト定義から継承するのではなく) 実行時環境から継承します。

特殊レジスターに対する変更が、関数の呼び出し元に戻されることはありません。

更新不能の特殊レジスター (日時特殊レジスターなど) は、現在実行中のステートメントのプロパティを反映するので、デフォルト値に設定されます。

#### 注:

- ユーザー定義関数のパラメーターのデータ・タイプを選択する場合は、入力値に影響を与えるプロモーションの規則を考慮してください。たとえば、入力値として使用される可能性のある定数のデータ・タイプは、予期される以外の組み込みデータ・タイプである可能性があり、さらには、予期されるデータ・タイプにプロモートできない場合があります。プロモーションの規則にしたがって、一般にパラメーターには以下のデータ・タイプを使用するようにしてください。
  - *SMALLINT* ではなく *INTEGER*
  - *REAL* ではなく *DOUBLE*
  - *CHAR* ではなく *VARCHAR*

## CREATE FUNCTION (外部表)

- GRAPHIC ではなく VARGRAPHIC
- プラットフォーム間での UDF の移植性を保つためには、以下のデータ・タイプの使用をお勧めします。
  - FLOAT ではなく DOUBLE または REAL
  - NUMERIC ではなく DECIMAL
  - LONG VARCHAR ではなく CLOB (または BLOB)
- まだ存在していないスキーマ名を用いて関数を作成すると、ステートメントの許可 ID に IMPLICIT\_SCHEMA 権限がある場合に限り、そのスキーマが暗黙に作成されます。そのスキーマの所有者は SYSIBM になります。スキーマに対する CREATEIN 特権が PUBLIC に付与されます。
- 索引拡張を定義するには、NO SQL として定義されたルーチンしか使用できません (SQLSTATE 428F8)。
- 関数が SQL を許可する場合、外部プログラムは、連合オブジェクトへのアクセスを試行してはなりません (SQLSTATE 55047)。
- **表アクセスの制限**

関数が READS SQL DATA に定義されている場合には、関数のいかなるステートメントも、関数を呼び出したステートメントによって変更されている表にはアクセスできません (SQLSTATE 57053)。たとえば、ユーザー定義関数 BONUS() が READS SQL DATA に定義されているとします。ステートメント UPDATE EMPLOYEE SET SALARY = SALARY + BONUS(EMPNO) が呼び出される場合、BONUS 関数の SQL ステートメントは、EMPLOYEE 表からは読み取りは行えません。
- **互換性**
  - DB2 for z/OS および OS/390 との互換性:
    - 以下の構文はデフォルトの振る舞いとして受け入れられます。
      - ASUTIME NO LIMIT
      - NO COLLID
      - PROGRAM TYPE SUB
      - STAY RESIDENT NO
  - 以前のバージョンの DB2 との互換性:
    - PARAMETER STYLE SQL の代わりに PARAMETER STYLE DB2SQL を指定できます。
    - DETERMINISTIC の代わりに NOT VARIANT を指定できます。
    - NOT DETERMINISTIC の代わりに VARIANT を指定できます。
    - CALLED ON NULL INPUT の代わりに NULL CALL を指定できます。
    - RETURNS NULL ON NULL INPUT の代わりに NOT NULL CALL を指定できます。
- **特権**



- 関数の定義者は、関数に対する EXECUTE WITH GRANT OPTION 特権と、関数をドロップする権利を常に与えられます。
- 関数を SQL ステートメントで使用する時点で、関数の定義者はその関数によって使用されるすべてのパッケージに対して EXECUTE 特権を持っていなければなりません。

### 例:

例 1: 以下の例では、テキスト管理システムにおいて既知の各文書の 1 つの文書 ID 列からなる行を戻す表関数を登録しています。最初のパラメーターは指定された対象領域をマッチングし、2 番目パラメーターには指定されたストリングが入ります。

単一セッションのコンテキスト内では UDF は常に同じ表を戻すため、UDF は DETERMINISTIC として定義されています。DOCMATCH からの出力を定義する RETURNS 文節に注意してください。それぞれの表関数に対して、FINAL CALL を指定する必要があります。さらに、この表関数は並列して実行できないので、DISALLOW PARALLEL キーワードが追加されています。DOCMATCH の出力のサイズは大きく変動しますが、DB2 オプティマイザーにとって有用な CARDINALITY 20 が代表値として指定されています。

```
CREATE FUNCTION DOCMATCH (VARCHAR(30), VARCHAR(255))
 RETURNS TABLE (DOC_ID CHAR(16))
 EXTERNAL NAME '/common/docfuncs/rajiv/udfmatch'
 LANGUAGE C
 PARAMETER STYLE SQL
 NO SQL
 DETERMINISTIC
 NO EXTERNAL ACTION
 NOT FENCED
 SCRATCHPAD
 FINAL CALL
 DISALLOW PARALLEL
 CARDINALITY 20
```

例 2: 以下の例では、Microsoft Exchange のメッセージのメッセージ・ヘッダー情報と、部分的なメッセージ・テキストの検索に使用する OLE 表関数を登録しています。

```
CREATE FUNCTION MAIL()
 RETURNS TABLE (TIMERECEIVED DATE,
 SUBJECT VARCHAR(15),
 SIZE INTEGER,
 TEXT VARCHAR(30))
 EXTERNAL NAME 'tfmail.header!list'
 LANGUAGE OLE
 PARAMETER STYLE SQL
 NOT DETERMINISTIC
 FENCED
 CALLED ON NULL INPUT
 SCRATCHPAD
```

## CREATE FUNCTION (外部表)

FINAL CALL  
NO SQL  
EXTERNAL ACTION  
DISALLOW PARALLEL

### 関連資料:

- *SQL* リファレンス 第 1 巻 の『基本述部』
- *SQL* リファレンス 第 2 巻 の『CREATE FUNCTION (OLE DB 外部表) ステートメント』
- 1552 ページの『CREATE FUNCTION (SQL スカラー、表、または行)』
- 1618 ページの『CREATE FUNCTION (外部スカラー)』
- 1360 ページの『ルーチンで使用可能な SQL ステートメント』
- *SQL* リファレンス 第 1 巻 の『特殊レジスター』
- *SQL* リファレンス 第 1 巻 の『データ・タイプのプロモーション』

## CREATE FUNCTION (ソースまたはテンプレート)

このステートメントは、以下の目的で使用されます。

- 他の既存のスカラー関数または列関数に基づくユーザー定義関数を、アプリケーション・サーバーに登録する。
- 連合サーバーとして指定されたアプリケーション・サーバーに、関数テンプレートを登録する。関数テンプレートとは、実行可能コードを含まない部分関数のことです。ユーザーは、データ・ソース関数へマッピングする目的でこれを作成します。マッピングを作成したら、連合サーバーへサブミットする照会に、その関数テンプレートを指定できます。そのような照会を処理する場合、連合サーバーは、テンプレートのマップ先のデータ・ソース関数を呼び出し、値を戻します。この値のデータ・タイプは、テンプレートの定義の RETURNS 部分にある値に対応するものです。

### 呼び出し:

このステートメントはアプリケーション・プログラムに組み込むことができ、また動的 SQL ステートメントを使用して出すことができます。DYNAMICRULES の実行動作がパッケージに効力を持つ場合にのみ、動的に準備できる実行可能ステートメントです (SQLSTATE 42509)。

### 許可:

ステートメントの許可 ID によって保持されている特権には、少なくとも以下のいずれかが含まれていなければなりません。

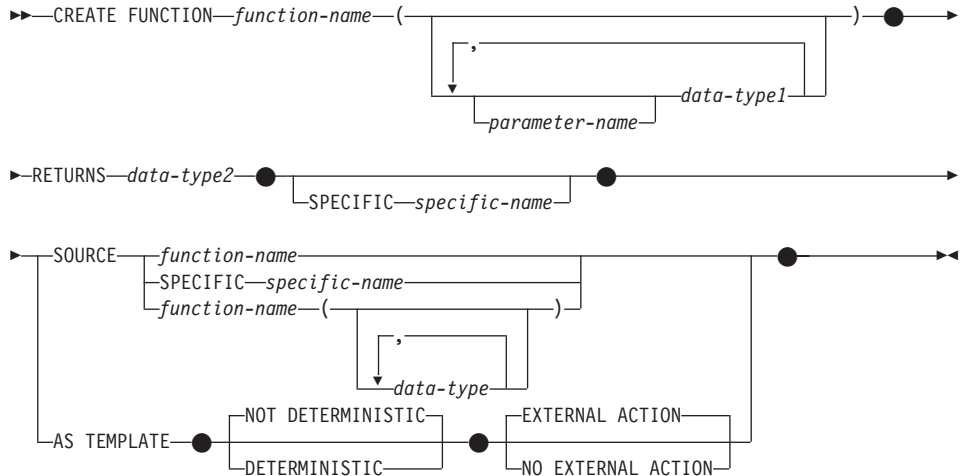
- SYSADM または DBADM 権限
- データベースに対する IMPLICIT\_SCHEMA 権限 (関数の暗黙または明示のスキーマ名が存在しない場合)
- スキーマに対する CREATEIN 特権 (関数のスキーマ名が存在する場合)

## CREATE FUNCTION (ソースまたはテンプレート)

許可 ID の権限が不十分で、操作を実行できない場合には、エラー (SQLSTATE 42502) になります。

SOURCE 文節で参照される関数に対する権限は必要ありません。

### 構文:



### 説明:

#### *function-name*

定義する関数または関数テンプレートを指定します。これは、関数を指定する修飾または非修飾の名前です。 *function-name* (関数名) の非修飾形式は SQL ID です (最大長 18)。動的 SQL ステートメントでは、CURRENT SCHEMA 特殊レジスタが、修飾子のないオブジェクト名の修飾子として使用されます。静的 SQL ステートメントでは、QUALIFIER プリコンパイル/バインド・オプションにより、修飾子のないオブジェクト名の修飾子が暗黙指定されます。修飾形式は、*schema-name* の後にピリオドと SQL ID が続きます。

暗黙または明示の修飾子を含む名前、およびパラメーターの数と各パラメーターのデータ・タイプ (データ・タイプの長さ、精度、または位取りの各属性には関係なく) は、カタログに記述されている関数または関数テンプレートを指定するものであってはなりません (SQLSTATE 42723)。非修飾名とパラメーターの数およびデータ・タイプとの組み合わせは、そのスキーマ内では当然固有ですが、複数のスキーマ間で固有である必要はありません。

2 つの部分からなる名前を指定する場合、'SYS' で始まる *schema-name* (スキーマ名) は使用できません (SQLSTATE 42939)。

述部のキーワードとして使用されるいくつかの名前は、システム使用に予約されており、*function-name* として使用することはできません (SQLSTATE 42939)。それ

## CREATE FUNCTION (ソースまたはテンプレート)

らの名前は、SOME、ANY、ALL、NOT、AND、OR、BETWEEN、NULL、LIKE、EXISTS、IN、UNIQUE、OVERLAPS、SIMILAR、MATCH、および比較演算子です。

既存の関数と同一の機能をユーザー定義特殊タイプに対してサポートする目的で、その既存の関数に基づくユーザー定義関数を指定する場合は、該当の既存の関数と同じ名前を使用することができます。これにより、ユーザーは、追加定義の必要性を特に意識することなく、同じ関数のユーザー定義特殊タイプ版を使用することができます。一般に、関数のシグニチャーに何らかの差異がある場合には、同じ名前を複数の関数に使用することができます。

*(data-type,...)*

関数または関数テンプレートの入力パラメーターの数を指定するとともに、各パラメーターのデータ・タイプを指定します。このリストには、関数または関数テンプレートが受け取ることを予期している各パラメーターごとに、1つの項目を指定する必要があります。パラメーターの数は90を超えることはできません。この限界を超えると、エラー (SQLSTATE 54023) になります。

パラメーターのない関数または関数テンプレートも登録可能です。この場合、指定するデータ・タイプがない場合でも、括弧はコーディングする必要があります。たとえば、

```
CREATE FUNCTION WOOFER() ...
```

その対応するすべてのパラメーターのタイプがまったく同じである場合でも、1つのスキーマ中に名前が同じ2つの関数または関数テンプレートがあってはなりません。(この制限は、同じ名前を持つスキーマ内の関数または関数テンプレートにも適用されます。) このタイプの比較では長さ、精度、および位取りは考慮されません。したがって、CHAR(8)とCHAR(35)、またDECIMAL(11,2)とDECIMAL(4,3)は、それぞれ同じタイプと見なされます。ユニコード・データベースの場合には、CHAR(13)とGRAPHIC(8)は、それぞれ同じタイプと見なされます。さらに、DECIMALとNUMERICなどのように、この目的で複数のタイプが同じタイプとして扱われることがあります。シグニチャーが重複していると、SQLエラー (SQLSTATE 42723) になります。

たとえば、以下のステートメントの場合、

```
CREATE FUNCTION PART (INT, CHAR(15)) ...
CREATE FUNCTION PART (INTEGER, CHAR(40)) ...
```

```
CREATE FUNCTION ANGLE (DECIMAL(12,2)) ...
CREATE FUNCTION ANGLE (DEC(10,7)) ...
```

2番目と4番目のステートメントは、重複する関数と見なされ、エラーになります。

## CREATE FUNCTION (ソースまたはテンプレート)

### *parameter-name*

パラメーターのオプション名を指定します。これは、この関数の他のパラメーターすべての名前と異なる名前にする必要があります。

### *data-type1*

パラメーターのデータ・タイプを指定します。

ソース・スカラー関数では、SOURCE 文節で指定された関数の対応するパラメーターのタイプにキャスト可能であれば、任意の有効な SQL データ・タイプを使用できます。REF(*type-name*) データ・タイプをパラメーターのデータ・タイプとして指定することはできません。

関数がソース関数から派生するので、パラメーター化データ・タイプの長さ、精度、または位取りを指定する必要はありません (指定は可能です)。代わりに、CHAR() などのように、空の括弧を使用できます。パラメーター化データ・タイプは、特定の長さ、位取り、または精度によって定義可能なデータ・タイプです。パラメーター化データ・タイプは、ストリング・データ・タイプと 10 進データ・タイプです。

## RETURNS

この文節は必須で、関数または関数テンプレートの出力を指定します。

### *data-type2*

出力のデータ・タイプを指定します。

ソース関数の結果のタイプからキャスト可能であれば、任意の有効な SQL データ・タイプ (特殊タイプも同様) が有効です。

上記のような、ソース関数のパラメーターの場合、パラメーター化タイプのパラメーターを指定する必要はありません。その代わりに、VARCHAR() のように、空の括弧を使用できます。

関数が他の関数に基づいている場合に RETURNS 文節のデータ・タイプの指定に適用される考慮事項と規則については、このステートメントの『規則』セクションを参照してください。

## SPECIFIC *specific-name*

定義する関数のインスタンスに対する固有名を指定します。この特定名は、この関数をソース関数として使用する場合、この関数をドロップする場合、またはこの関数にコメントを付ける場合に使用することができます。これは、関数の呼び出しには使用できません。*specific-name* (関数名) の非修飾形式は SQL ID です (最大長 18)。修飾形式は、*schema-name* の後にピリオドと SQL ID が続きます。暗黙または明示の修飾子も含め、その名前が、アプリケーション・サーバーに存在する別の関数インスタンスを指定するものであってはなりません。そうでない場合、エラー (SQLSTATE 42710) になります。

*specific-name* は、既存の *function-name* (関数名) と同じでも構いません。

## CREATE FUNCTION (ソースまたはテンプレート)

修飾子を指定しない場合、*function-name* に使用された修飾子が使用されます。修飾子を指定する場合は、*function-name* の明示修飾子または暗黙修飾子と同じでなければなりません。そうでない場合、エラー (SQLSTATE 42882) になります。

*specific-name* の指定がない場合、固有の名前がデータベース・マネージャーによって生成されます。生成される固有の名前は、SQL の後に文字のタイム・スタンプが続く名前です (SQLyymmddhhmmssxxx)。

### SOURCE

作成する関数が、データベース・マネージャーにとって既知の他の関数 (ソース関数) によってインプリメントされる関数であることを指定します。ソース関数は、組み込み関数 (COALESCE、DBPARTITIONNUM、NULLIF、HASHEDVALUE、TYPE\_ID、TYPE\_NAME、TYPE\_SCHEMA、または VALUE を除く) か、以前に作成したユーザー定義のスカラー関数のいずれかになります。

SOURCE 文節は、スカラー関数または列関数に対してのみ指定が可能で、表関数には指定できません。

SOURCE 文節によって、他の関数と同一の機能を持たせることが可能になります。

#### *function-name*

ソースとして使用される特定の関数を指定します。ステートメントの許可 ID が EXECUTE 特権を持ち、この *function-name* を名前とする特定の関数が、スキーマに実際に 1 つだけ存在している場合にのみ有効です。この構文変数は、組み込み関数であるソース関数に対しては無効です。

非修飾名を指定すると、現行 SQL パス (CURRENT PATH 特殊レジスタの値) がその関数を見つけるのに使用されます。EXECUTE 特権のあるステートメントの許可 ID の名前を持つ関数が含まれている、関数パスの最初のスキーマが選択されます。

指定されたスキーマにこの名前関数が見つからないか、あるいは名前が修飾されておらず、この名前関数が関数パスにない場合は、エラー (SQLSTATE 42704) になります。指定したスキーマまたは見つかったスキーマに、この関数の許可された特定インスタンスが複数個ある場合には、エラー (SQLSTATE 42725) になります。その名前関数が存在し、ステートメントの許可 ID が関数に対して EXECUTE 特権を持っていない場合には、エラー (SQLSTATE 42501) が生じます。

### SPECIFIC *specific-name*

関数の作成時に指定されたか、またはデフォルト値として使用された *specific-name* (特定名) を使用して、ソースとして使用する特定のユーザー定義関数を指定します。この構文変数は、組み込み関数であるソース関数に対しては無効です。

非修飾名を指定すると、現行 SQL パスがその関数を見つけるのに使用されます。EXECUTE 特権のあるステートメントの許可 ID の特定の名前を持つ関数が含まれている、関数パスの最初のスキーマが選択されます。

## CREATE FUNCTION (ソースまたはテンプレート)

指定されたスキーマにこの *specific-name* の関数が見つからないか、あるいは名前が修飾されておらず、この *specific-name* の関数が SQL パスにない場合は、エラー (SQLSTATE 42704) になります。 *specific-name* の関数が存在し、ステートメントの許可 ID が関数に対して EXECUTE 特権を持っていない場合には、エラー (SQLSTATE 42501) が出されます。

### *function-name (data-type,...)*

ソース関数を固有に指定する関数シグニチャーを指定します。組み込み関数であるソース関数に対しては、これが唯一有効な構文変数です。

同じ関数名と SOURCE 文節に指定されたデータ・タイプを持つ複数の関数の中から 1 つの関数を選択するために、関数解決の規則が適用されます。ただし、選択された関数の各パラメーターのデータ・タイプは、ソース関数に指定された対応するデータ・タイプと、まったく同じタイプでなければなりません。

### *function-name*

ソース関数の関数名を指定します。非修飾名を指定すると、ユーザーの SQL パスのスキーマが考慮されます。

### *data-type*

これは、CREATE FUNCTION ステートメントで対応する位置 (コンマで区切られた) に指定されたデータ・タイプに一致していなければなりません。

パラメーター化データ・タイプの長さ、精度、または位取りを指定する必要はありません。代わりに、空の括弧をコーディングすることによって、データ・タイプの一致を調べる際にそれらの属性を無視するように指定することができます。たとえば、DECIMAL() は、データ・タイプが DECIMAL(7,2) として定義されているパラメーターと一致します。

パラメーター値が異なるデータ・タイプ (REAL または DOUBLE) を示しているため、FLOAT() を使用することはできません (SQLSTATE 42601)。

ただし、長さ、精度、または位取りをコーディングする場合、その値は、CREATE FUNCTION ステートメントにおける指定に完全に一致していなければなりません。これは、意図した通りの関数が確実に使用されるようにする場合に便利です。また、データ・タイプの同義語は一致と見なされることにも注意してください (たとえば、DEC と NUMERIC は一致しません)。

$0 < n < 25$  は REAL を意味し、 $24 < n < 54$  は DOUBLE を意味するので、FLOAT(n) のタイプは、n に定義された値と一致している必要はありません。マッチングは、タイプが REAL か DOUBLE かに基づいて行われません。



## CREATE FUNCTION (ソースまたはテンプレート)

指定したスキーマまたは暗黙のスキーマに、指定したシグニチャーを持つ関数がない場合は、エラー (SQLSTATE 42883) になります。

### AS TEMPLATE

このステートメントが、実行可能コードを含む関数ではなく、関数テンプレートを作成するために使われることを示しています。

### DETERMINISTIC または NOT DETERMINISTIC

この文節は任意指定で、特定の引き数の値に対して関数が常に同じ結果を返すか (DETERMINISTIC)、それとも状態値に依存して関数の結果が影響を受けるか (NOT DETERMINISTIC) を指定します。つまり DETERMINISTIC 関数は、同一の入力で正しい呼び出しが行われたとき、常に同じ表を返します。NOT DETERMINISTIC を指定すると、同じ入力によって常に同じ結果が生じる利点に基づく最適化ができなくなります。

関数の本体が特殊レジスターにアクセスしたり、他の非 deterministic 関数を呼び出す場合には、NOT DETERMINISTIC を明示的または暗黙的に指定しなければなりません (SQLSTATE 428C2)。

### NO EXTERNAL ACTION または EXTERNAL ACTION

このオプションの文節は、データベース・マネージャーによって管理されていないオブジェクトの状態を変更するアクションを関数が行うかどうかを指定します。NO EXTERNAL ACTION を指定すると、システムは関数が外部に影響を与えないことを前提とした最適化を使用できます。

関数の本体が外部アクションのある別の関数を呼び出す場合、EXTERNAL ACTION を明示的または暗黙的に指定しなければなりません (SQLSTATE 428C2)。

### 規則:

- 便宜上、この項では作成する関数を CF と呼び、SOURCE 文節で指定する関数を SF と呼びます (許される 3 つの構文のどれが SF の指定に使用されたかは関係ありません)。
  - CF と SF のそれぞれの非修飾名が異なる名前であっても構いません。
  - 他の関数のソースとして指定された関数自体が、別の関数をソースとして使用した関数であっても構いません。間接的に呼び出された関数がエラーになると、アプリケーションをデバッグすることが極めて難しくなるので、この機能を使用する場合には細心の注意を払う必要があります。
  - 以下の文節は、SOURCE 文節と共に指定した場合には無効になります (CF はこれらの属性を SF から継承するからです)。
    - CAST FROM ...
    - EXTERNAL ...
    - LANGUAGE ...
    - PARAMETER STYLE ...



## CREATE FUNCTION (ソースまたはテンプレート)

- DETERMINISTIC/NOT DETERMINISTIC
- FENCED/NOT FENCED
- RETURNS NULL ON NULL INPUT/CALLED ON NULL INPUT
- EXTERNAL ACTION/NO EXTERNAL ACTION
- NO SQL/CONTAINS SQL/READS SQL DATA
- SCRATCHPAD/NO SCRATCHPAD
- FINAL CALL/NO FINAL CALL
- RETURNS TABLE (...)
- CARDINALITY ...
- ALLOW PARALLEL/DISALLOW PARALLEL
- DBINFO/NO DBINFO
- THREADSAFE/NOT THREADSAFE
- INHERIT SPECIAL REGISTERS

これらの規則に違反すると、エラー (SQLSTATE 42613) になります。

- CF の入力パラメーターの数は、SF のパラメーターの数と同じでなければなりません。異なる場合には、エラー (SQLSTATE 42624) になります。
- 以下の場合には、CF にパラメーター化データ・タイプの長さ、精度、または位取りを指定する必要がありません。
  - 関数の入力パラメーター
  - その RETURNS パラメーター

代わりに、VARCHAR() のように、データ・タイプの一部として空の括弧を使用することにより、長さ、精度、および位取りがソース関数と同じであるか、あるいはキャストによって決定されるように指定することができます。

ただし、長さ、精度、または位取りを指定した場合には、CF における値と SF の対応する値とが比較チェックされます。これについては、以下で入力パラメーターと戻り値とに分けて説明します。

- CF の入力パラメーターの指定は、SF の入力パラメーターの指定と比較チェックされます。CF の各パラメーターのデータ・タイプは、SF の対応するパラメーターのデータ・タイプと同じであるか、あるいはキャスト可能でなければなりません。同じタイプでないか、あるいはキャスト可能ではないパラメーターがある場合には、エラー (SQLSTATE 42879) になります。

この規則は、CF の使用時に発生し得るエラーに対して何らかの保証を与えるものではありません。CF パラメーターのデータ・タイプや長さ、または精度属性に一致する引き数は、対応する SF パラメーターの方が長さが短かったり精度が劣る場合には、割り当てることができません。一般に、CF のパラメーターの長さ属性や精度属性は、対応する SF パラメーターのそれよりも大きくしてはなりません。

## CREATE FUNCTION (ソースまたはテンプレート)

- CF の RETURNS データ・タイプの指定は、SF のそれと比較チェックされます。キャスト後の SF の最終の RETURNS データ・タイプは、CF の RETURNS のデータ・タイプと同じか、あるいはそれにキャスト可能でなければなりません。そうでない場合、エラー (SQLSTATE 42866) になります。

この規則は、CF の使用時に発生し得るエラーに対して何らかの保証を与えるものではありません。SF の RETURNS データ・タイプのデータ・タイプや長さもしくは精度属性に一致する結果の値は、CF の RETURNS データ・タイプの方が長さが短かったり精度が劣る場合には、割り当てることができません。長さもしくは精度属性が SF の RETURNS データ・タイプのそれよりも小さい CF の RETURNS データ・タイプを指定することにした場合は、十分に注意を払ってください。

### 注:

- あるデータ・タイプが他のデータ・タイプにキャスト可能かどうかの判別では、CHAR や DECIMAL などのパラメーター化データ・タイプの長さまたは精度と位取りは考慮されません。したがって、ソース・データ・タイプの値をターゲット・データ・タイプの値にキャストしようとする、関数の使用時にエラーになる可能性があります。たとえば、VARCHAR は DATE にキャストできますが、実際にはソース・データ・タイプが VARCHAR(5) と定義されている場合には、関数の使用時にエラーになります。
- ユーザー定義関数のパラメーターのデータ・タイプを選択する場合は、入力値に影響を与えるプロモーションの規則を考慮してください (『データ・タイプのプロモーション』を参照してください)。たとえば、入力値として使用できる定数のデータ・タイプは、予期される以外の組み込みデータ・タイプである可能性があり、さらには、予期されるデータ・タイプにプロモートできない場合があります。プロモーションの規則にしたがって、一般にパラメーターには以下のデータ・タイプを使用するようにしてください。
  - SMALLINT ではなく INTEGER
  - REAL ではなく DOUBLE
  - CHAR ではなく VARCHAR
  - GRAPHIC ではなく VARGRAPHIC
- まだ存在していないスキーマ名を用いて関数を作成すると、ステートメントの許可 ID に IMPLICIT\_SCHEMA 権限がある場合に限り、そのスキーマが暗黙に作成されます。そのスキーマの所有者は SYSIBM になります。スキーマに対する CREATEIN 特権が PUBLIC に付与されます。
- データ・ソース関数を認識する連合サーバーの場合、この関数は連合データベースにあるもう一方の関数にマップする必要があります。データベースに対となる関数がない場合、ユーザーがそれを作成してマッピングする必要があります。

対となるもう一方は、関数 (スカラーまたはソース) か、関数テンプレートとすることができます。ユーザーが関数を作成して必要なマッピングを行うと、その関数を指

## CREATE FUNCTION (ソースまたはテンプレート)

定する照会を処理するたびに、DB2 は、(1) その関数を呼び出すときの戦略と、データ・ソース関数を呼び出すときの戦略を比べ、(2) オーバーヘッドが少ないと思われる関数を呼び出します。

ユーザーが関数テンプレートとマッピングを作成すると、そのテンプレートを指定する照会を処理するたびに、DB2 は、マッピング先のデータ・ソース関数を呼び出します (ただし、この関数を呼び出すためのアクセス・プランが存在する場合)。

### • 特権

関数の定義者は、関数に対する EXECUTE 特権と、関数をドロップする権利を常に与えられます。以下のいずれかが真の場合には、関数の定義者には WITH GRANT OPTION も与えられます。

- ソース関数が、組み込み関数である。
- 関数の定義者が、ソース関数に対して EXECUTE WITH GRANT OPTION を持っている。
- 関数が、テンプレートである。

### 例:

例 1: Pellow がオリジナルの CENTRE 関数を作成した後に、別のユーザーがその関数に基づいて関数を作成します。この関数は、整数引き数のみを受け入れるという点だけが異なります。

```
CREATE FUNCTION MYCENTRE (INTEGER, INTEGER)
 RETURNS FLOAT
 SOURCE PELLOW.CENTRE (INTEGER, FLOAT)
```

例 2: 組み込みの INTEGER データ・タイプに基づく特殊タイプ HATSIZE が作成済みです。それぞれの部門の平均の帽子サイズを計算するには、AVG 関数を使用すると便利です。これは、以下のようにして簡単に実行できます。

```
CREATE FUNCTION AVG (HATSIZE) RETURNS HATSIZE
 SOURCE SYSIBM.AVG (INTEGER)
```

特殊タイプの作成により必要な cast 関数が生成され、引き数の場合は HATSIZE から INTEGER に、関数の結果の場合は INTEGER から HATSIZE にキャストすることが可能です。

例 3: 連合システムで、表統計を倍精度浮動小数点付きの値で戻す Oracle UDF を起動します。連合サーバーは、この関数と連合データベース側の対となる関数との間でマッピングが行われる場合にだけ、この関数を認識することができます。ところが、そのような対となる関数は存在していません。それで、対となる関数を関数テンプレートの形で指定して、このテンプレートを NOVA というスキーマに割り当てることを決定します。以下のコードを使用して、テンプレートを連合サーバーに登録します。

```
CREATE FUNCTION NOVA.STATS (DOUBLE, DOUBLE)
 RETURNS DOUBLE
 AS TEMPLATE DETERMINISTIC NO EXTERNAL ACTION
```

## CREATE FUNCTION (ソースまたはテンプレート)

例 4: 連合システムで、特定の組織の従業員に対して支給されるボーナスの総額を戻す Oracle UDF を呼び出します。連合サーバーは、この関数と連合データベース側の対となる関数との間でマッピングが行われる場合にだけ、この関数を認識することができます。ところがそのような対となる関数は存在しないため、ユーザーが関数テンプレートの形で作成します。以下のコードを使用して、このテンプレートを連合サーバーに登録します。

```
CREATE FUNCTION BONUS ()
 RETURNS DECIMAL (8,2)
 AS TEMPLATE DETERMINISTIC NO EXTERNAL ACTION
```

### 関連資料:

- *SQL* リファレンス 第 1 巻 の『関数』
- *SQL* リファレンス 第 1 巻 の『基本述部』
- *SQL* リファレンス 第 2 巻 の『CREATE FUNCTION MAPPING ステートメント』
- *SQL* リファレンス 第 1 巻 の『データ・タイプのプロモーション』
- *SQL* リファレンス 第 1 巻 の『データ・タイプ間のキャスト』

## CREATE PROCEDURE (外部)

CREATE PROCEDURE (外部) ステートメントは、アプリケーション・サーバーで外部プロシージャを定義するのに使用します。

### 呼び出し:

このステートメントはアプリケーション・プログラムに組み込むことができ、また動的 SQL ステートメントを使用して出すことができます。 DYNAMICRULES の実行動作がパッケージに効力を持つ場合にのみ、動的に準備できる実行可能ステートメントです (SQLSTATE 42509)。

### 許可:

ステートメントの許可 ID によって保持されている特権には、少なくとも以下のいずれかが含まれていなければなりません。

- SYSADM または DBADM 権限
- データベースに対する CREATE\_EXTERNAL\_ROUTINE 権限、および以下の少なくとも 1 つ。
  - データベースに対する IMPLICIT\_SCHEMA 権限 (プロシージャのスキーマ名が既存のスキーマを指していない場合)
  - スキーマに対する CREATEIN 特権 (プロシージャのスキーマ名が既存のスキーマを指している場合)

## CREATE PROCEDURE (外部)

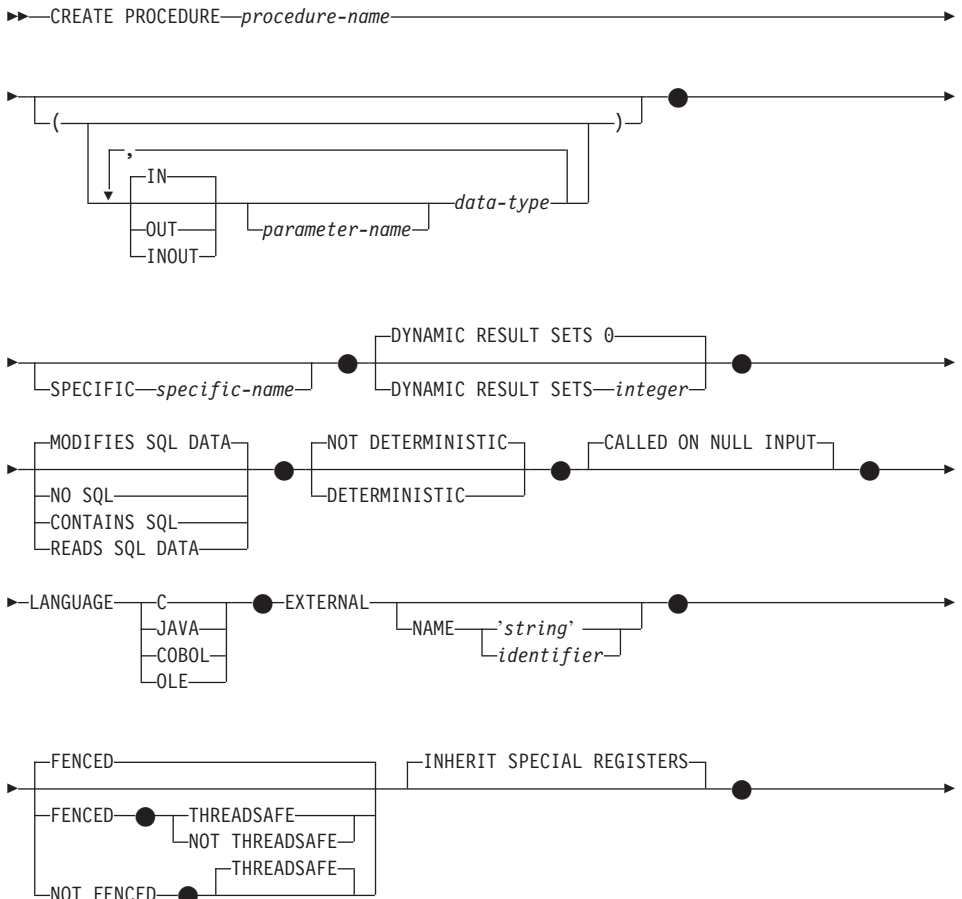
非隔離のストアード・プロシージャーを作成するには、ステートメントの許可 ID の特権に以下の特権の少なくとも 1 つが含まれている必要があります。

- データベースに対する CREATE\_NOT\_FENCED\_ROUTINE 権限
- SYSADM または DBADM 権限

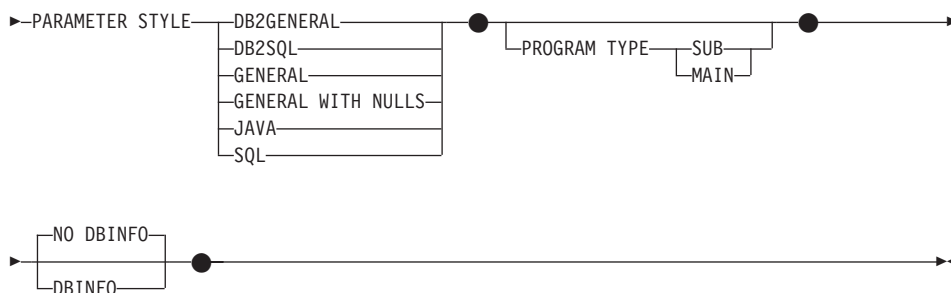
隔離ストアード・プロシージャーを作成する場合、追加の権限や特権は必要ありません。

許可 ID の権限が不十分で、操作を実行できない場合には、エラー (SQLSTATE 42502) になります。

### 構文:



## CREATE PROCEDURE (外部)



### 説明:

#### *procedure-name*

定義するプロシージャの名前を指定します。この名前は、プロシージャを指定する修飾または非修飾の名前です。 *procedure-name* (プロシージャ名) の非修飾形式は SQL ID です (最大長 128)。動的 SQL ステートメントでは、CURRENT SCHEMA 特殊レジスターが、修飾子のないオブジェクト名の修飾子として使用されます。静的 SQL ステートメントでは、QUALIFIER プリコンパイルバインド・オプションにより、修飾子のないオブジェクト名の修飾子が暗黙指定されます。修飾形式は、*schema-name* の後にピリオドと SQL ID が続きます。

暗黙または明示の修飾子を含む名前と、パラメーターの数との組み合わせは、カタログにすでに記述されているプロシージャを指定するものであってはなりません (SQLSTATE 42723)。非修飾名とパラメーターの数との組み合わせは、複数のスキーマ間で固有である必要はありません。

2 つの部分からなる名前を指定する場合、'SYS' で始まる *schema-name* (スキーマ名) は使用できません (SQLSTATE 42939)。

#### (IN | OUT | INOUT *parameter-name data-type*,...)

プロシージャのパラメーターを指定し、各パラメーターのモード、データ・タイプ、およびオプション名を指定します。このリストには、プロシージャが予期する各パラメーターごとに 1 つの項目を指定する必要があります。

1 つのスキーマに同じ名前の 2 つのプロシージャがある場合、パラメーターの数をまったく同一にすることはできません。シグニチャーが重複していると、SQL エラー (SQLSTATE 42723) になります。

たとえば、以下のステートメントの場合、

```
CREATE PROCEDURE PART (IN NUMBER INT, OUT PART_NAME CHAR(35)) ...
CREATE PROCEDURE PART (IN COST DECIMAL(5,3), OUT COUNT INT) ...
```

**IN** パラメーターをプロシージャの入力パラメーターとして指定します。プロシージャ内でパラメーターに加えられるすべての変更は、制御が戻されると SQL アプリケーションの呼び出しは行なえなくなります。デフォルトは IN です。

**OUT**

パラメーターをプロシージャの出力パラメーターとして指定します。

**INOUT**

パラメーターを、プロシージャの入力および出力パラメーターの両方として指定します。

*parameter-name*

パラメーターの名前を任意に指定します。パラメーター名は、プロシージャで固有でなければなりません (SQLSTATE 42734)。

*data-type*

パラメーターのデータ・タイプを指定します。

- CREATE TABLE ステートメントのデータ・タイプの定義に指定することが可能で、プロシージャの作成に使用されている言語に対応するものがある SQL データ・タイプ指定と省略形を指定できます。
- ユーザー定義データ・タイプはサポートされていません (SQLSTATE 42601)。
- LONG VARCHAR および LONG VARGRAPHIC データ・タイプは、SQL プロシージャのパラメーター・タイプとしてサポートされていません。

**SPECIFIC** *specific-name*

定義するプロシージャのインスタンスに対する固有名を指定します。この特定名は、このプロシージャをドロップする場合、またはこのプロシージャにコメントを付ける場合に使用することができます。これは、プロシージャの呼び出しには使用できません。 *specific-name* (関数名) の非修飾形式は SQL ID です (最大長 18)。修飾形式は、*schema-name* の後にピリオドと SQL ID が続きます。暗黙または明示の修飾子も含め、その名前が、アプリケーション・サーバーに存在する別のルーチン・インスタンスを指定するものであってはなりません。そうでない場合、エラー (SQLSTATE 42710) になります。

*specific-name* は、既存の *procedure-name* と同じでも構いません。

修飾子を指定しない場合、*procedure-name* に使用された修飾子が使用されます。修飾子を指定する場合は、*procedure-name* の明示修飾子または暗黙修飾子と同じでなければなりません。そうでない場合、エラー (SQLSTATE 42882) になります。

*specific-name* の指定がない場合、固有の名前がデータベース・マネージャーによって生成されます。生成される固有の名前は、SQL の後に文字のタイム・スタンプが続く名前です (SQLlyymmddhhmmssshhn)。

**DYNAMIC RESULT SETS** *integer*

ストアド・プロシージャから戻される結果セットの上限の見積もりを指定します。

## CREATE PROCEDURE (外部)

### NO SQL、CONTAINS SQL、READS SQL DATA、MODIFIES SQL DATA

ストアド・プロシージャから SQL ステートメントが発行されるかどうかと、もし発行されればどのタイプかを示します。

#### NO SQL

ストアド・プロシージャはどの SQL ステートメントも実行できないことを指示します (SQLSTATE 38001)。

#### CONTAINS SQL

SQL データの読み取りも変更も行わない SQL ステートメントを、ストアド・プロシージャで実行できることを指定します (SQLSTATE 38004)。どのストアド・プロシージャでもサポートされていないステートメントは、これとは異なるエラーを戻します (SQLSTATE 38003)。

#### READS SQL DATA

SQL データを変更しない SQL ステートメントを、ストアド・プロシージャで実行できることを指定します (SQLSTATE 38002 または 42985)。どのストアド・プロシージャでもサポートされていないステートメントは、これとは異なるエラーを戻します (SQLSTATE 38003)。

#### MODIFIES SQL DATA

このストアド・プロシージャは、ストアド・プロシージャでサポートされていないステートメント以外のすべての SQL ステートメントを実行できることを指定します (SQLSTATE 38003)。

### DETERMINISTIC または NOT DETERMINISTIC

この文節は、同一の引き数値に対してプロシージャが常に同じ結果を戻すか (DETERMINISTIC)、それとも状態値に依存してプロシージャの結果が影響を受けるか (NOT DETERMINISTIC) を指定します。つまり DETERMINISTIC が指定されたプロシージャは、同じ入力を指定して正しく呼び出した場合に常に同じ結果を戻します。

現在、この文節はストアド・プロシージャの処理に影響を与えません。

### CALLED ON NULL INPUT

CALLED ON NULL INPUT は、ストアド・プロシージャに常に適用されます。これは、任意の引き数が NULL かどうかにかかわらず、ストアド・プロシージャが呼び出されることを意味します。OUT または INOUT パラメーターは、NULL 値を戻す場合も、通常の (NULL 値以外の) 値を戻す場合もあります。NULL の引き数値の有無のテストはストアド・プロシージャで行う必要があります。

### LANGUAGE

この文節は必須で、ストアド・プロシージャの本体が準拠している言語インターフェイス規則を指定するのに使用されます。

**C** データベース・マネージャーは、ストアド・プロシージャを C プロシージャであるかのように呼び出します。ストアド・プロシージャは、標準



ANSI C プロトタイプで定義されている C 言語の呼び出し規則およびリンケージ規則に準拠していなければなりません。

### JAVA

データベース・マネージャーは、Java クラス内のメソッドとしてストアード・プロシージャを呼び出します。

### COBOL

データベース・マネージャーは、プロシージャを COBOL プロシージャであるかのように呼び出します。

### OLE

データベース・マネージャーは、OLE 自動化オブジェクトによって公開されたメソッドであるものとしてストアード・プロシージャを呼び出します。ストアード・プロシージャは、OLE 自動化データ・タイプと呼び出しメカニズムに準拠している必要があります。さらに OLE 自動化オブジェクトは、プロセス内サーバー (DLL) としてインプリメントされる必要もあります。これらの制約事項については、*OLE Automation Programmer's Reference* で説明されています。

LANGUAGE OLE は、DB2 (Windows オペレーティング・システム版) に保管されているストアード・プロシージャに対してのみサポートされます。LANGUAGE OLE を指定したプロシージャには、THREADSAFE は指定できません (SQLSTATE 42613)。

### EXTERNAL

この文節は、この CREATE PROCEDURE ステートメントを使用して登録する新しいプロシージャが、外部プログラミング言語で作成されたコードに基づいており、文書化されたリンケージの規則とインターフェースにしたがっていることを示します。

NAME 文節の指定がない場合、"NAME *procedure-name*" が想定されます。

### NAME 'string'

この文節は、定義するプロシージャをインプリメントするユーザー作成コードの名前を指定します。

'string' オプションは、最大 254 文字のストリング定数です。ストリングに使用される形式は、指定した LANGUAGE によって異なります。

- LANGUAGE C の場合

指定する *string* は、ライブラリー名と作成しているストアード・プロシージャを実行するためにデータベース・マネージャーが呼び出すそのライブラリー中のプロシージャです。ライブラリー (およびそのライブラリー中のプロシージャ) は、CREATE PROCEDURE ステートメントの実行時に存在している必要はありません。ただし、プロシージャが呼び出される時点

## CREATE PROCEDURE (外部)

では、該当のライブラリーとそのライブラリー中の該当のプロシージャは存在していなければならない、またデータベース・サーバーのマシンからアクセス可能でなければなりません。

→ ' library\_id ' →  
└─ absolute\_path\_id ─┘ └─ !-proc\_id ─┘

名前は、単一引用符で囲む必要があります。余分なブランクを使用することはできません。

### *library\_id*

該当のプロシージャが入っているライブラリーの名前を指定します。データベース・マネージャーは、 `.../sqllib/function/unfenced` ディレクトリーと `.../sqllib/function` ディレクトリー (UNIX 系システムの場合)、または `...¥instance_name¥function¥unfenced` ディレクトリーと `...¥instance_name¥function` ディレクトリー (DB2INSTPROF レジストリー変数で指定した Windows オペレーティング・システムの場合) を調べてそのライブラリーを探します。その中で、データベース・マネージャーは、データベース・マネージャーの実行に使用されている制御 `sqllib` ディレクトリーを見つけ出します。たとえば、UNIX 系システムの制御 `sqllib` ディレクトリーは、 `/u/$DB2INSTANCE/sqllib` です。

UNIX 系システムの *library\_id* が `'myproc'` の場合に、データベース・マネージャーが `/u/production` から実行されていれば、データベース・マネージャーはライブラリー `/u/production/sqllib/function/unfenced/myfunc` と `/u/production/sqllib/function/myfunc` からプロシージャを見つけます。

Windows オペレーティング・システムの場合、関数ディレクトリーで *library\_id* が見つからないと、データベース・マネージャーは `LIBPATH` または `PATH` を調べます。

これらのディレクトリーのいずれかに存在しているストアード・プロシージャは、登録済み属性を使用しません。

### *absolute\_path\_id*

プロシージャの絶対パス名を指定します。

たとえば、UNIX 系システムの場合、`'/u/jchui/mylib/myproc'` を指定すると、データベース・マネージャーは `/u/jchui/mylib` を調べて `myproc` プロシージャを探索します。

Windows オペレーティング・システムの場合、`'d:¥mylib¥myproc'` を指定すると、データベース・マネージャーは `d:¥mylib` ディレクトリーから `myproc.dll` ファイルをロードします。

注: Windows オペレーティング・システムで C または C++ のルーチンを登録しているとき、EXTERNAL NAME 文節内でルーチン本

体識別しているときに、次のことに注意してください。絶対パス ID がルーチン本体の識別に使用されている場合は、.dll 拡張子を必ず付加してください。

#### ! *proc\_id*

呼び出すプロシージャの入り口点の名前を指定します。! は、ライブラリー ID とプロシージャ ID との間の区切り文字です。! *proc\_id* を省略すると、データベース・マネージャーはライブラリーのリンク時に確立されたデフォルトの入り口点を使用します。

たとえば、UNIX 系システムで 'mymod!proc8' を指定すると、データベース・マネージャーはライブラリー \$inst\_home\_dir/sql/lib/function/mymod を調べて、そのライブラリー内の入り口点 proc8 を使用します。

Windows オペレーティング・システムの場合、'mymod!proc8' を指定すると、データベース・マネージャーは mymod.dll ファイルをロードして、そのダイナミック・リンク・ライブラリー (DLL) の proc8() 関数を呼び出します。

ストリングの形式が正しくない場合には、エラー (SQLSTATE 42878) になります。

ストアード・プロシージャの本体は、マウントされてデータベースのすべてのパーティションで使用可能なディレクトリーに入っていない限りなりません。

#### • LANGUAGE JAVA の場合

指定する *string* には、作成中のストアード・プロシージャを実行するためにデータベース・マネージャーが呼び出す、任意指定の jar ファイル、クラス ID、およびメソッド ID が含まれています。クラス ID とメソッド ID は、CREATE PROCEDURE ステートメントの実行時には存在している必要はありません。jar\_id を指定する場合、ID は、CREATE PROCEDURE ステートメントの実行時に存在していなければなりません。ただし、プロシージャを呼び出す時点では、該当のクラス ID とメソッド ID が存在し、データベース・サーバーのマシンからアクセス可能でなければなりません。そうでない場合、エラー (SQLSTATE 42884) になります。

→ ' \_\_\_\_\_class\_id\_\_\_\_\_! \_\_\_\_\_method\_id\_\_\_\_\_'  
 └── jar\_id : ─┘ └──┘

名前は、単一引用符で囲む必要があります。余分なブランクを使用することはできません。

#### jar\_id

jar の集合をデータベースへインストールしたときに、その jar の集合に付けられた jar ID を指定します。これは、単純 ID またはスキーマ修飾 ID のいずれかにすることができます。たとえば、'myJar' や 'mySchema.myJar' のようになります。

## CREATE PROCEDURE (外部)

### *class\_id*

Java オブジェクトのクラス ID を指定します。クラスがパッケージの一部である場合、クラス ID の一部に完全なパッケージ接頭部 (たとえば、`'myPacks.StoredProcs'`) が含まれている必要があります。Java 仮想マシンは、ディレクトリー `'./myPacks/StoredProcs/'` の中のクラスを探します。Windows オペレーティング・システムでは、Java 仮想マシンはディレクトリー `'..¥myPacks¥StoredProcs¥'` を探索します。

### *method\_id*

呼び出す Java クラスのメソッド名を指定します。

#### • LANGUAGE OLE の場合

指定する文字列は、ステートメントが作成しているストアード・プロシージャを実行するためにデータベース・マネージャが呼び出す OLE のプログラム ID (*progid*) またはクラス ID (*clsid*)、およびメソッド ID (*method\_id*) です。プログラム ID またはクラス ID、およびメソッド ID は、CREATE PROCEDURE ステートメントの実行時に存在している必要はありません。ただし、関数を CALL ステートメントで使用する時点で、メソッド ID は存在していなければならず、データベース・サーバーのマシンからアクセス可能でなければなりません。そうでない場合、エラー (SQLSTATE 42724) になります。

▶▶' *progid* | *method\_id* | *clsid* '▶▶

名前は、単一引用符で囲む必要があります。余分なブランクを使用することはできません。

### *progid*

OLE オブジェクトのプログラム ID を指定します。

*progid* は、データベース・マネージャには解釈されず、実行時に OLE に転送されるだけです。指定する OLE オブジェクトは、作成可能である必要があります、実行時バインディング (ディスパッチに基づくバインディングとも呼ばれる) をサポートしている必要があります。規約では、*progid* は以下のような形式になります。

```
<program_name>.<component_name>.<version>
```

これは規約でしかなく、厳密な規則ではないので、*progids* をこれとは異なる形式にしても構いません。

### *clsid*

作成する OLE オブジェクトのクラス ID を指定します。OLE オブジェクトが *progid* を指定して登録されていない場合に、*progid* を指定する代わりに使用することができます。*clsid* の形式は以下のとおりです。

```
{nnnnnnnn-xxxx-xxxx-xxxx-xxxxxxxxxxxx}
```

ここで 'n' は英数字です。 *clsid* は、データベース・マネージャーには解釈されず、実行時に OLE API に転送されるだけです。

*method\_id*

呼び出す OLE オブジェクトのメソッド名を指定します。

#### NAME *identifier*

指定する *identifier* は SQL ID です。SQL ID は、ストリングの *library-id* として使用されます。区切られた ID でない場合、ID は大文字に変換されます。ID がスキーマ名で修飾されている場合、スキーマ名の部分は無視されます。この形式の NAME は、LANGUAGE C でのみ使用可能です。

#### FENCED または NOT FENCED

この文節は、ストアード・プロシージャをデータベース・マネージャーの操作環境のプロセスまたはアドレス・スペースで実行しても『安全』か (NOT FENCED)、否か (FENCED) を指定します。

ストアード・プロシージャが FENCED として登録されると、データベース・マネージャーは、その内部リソース (データ・バッファなど) を保護して、そのプロシージャからアクセスされないようにします。すべてのプロシージャは、FENCED として実行するか NOT FENCED として実行するかの選択が可能です。一般に、FENCED として実行されるプロシージャは、NOT FENCED として実行されるものと同じようには実行されません。

#### 注意:

十分にチェックされていないプロシージャに NOT FENCED を使用すると、DB2 の健全性に危険を招く場合があります。DB2 では、発生する可能性のある一般的な不注意による障害の多くに対して、いくつかの予防措置がとられていますが、NOT FENCED ストアード・プロシージャが使用される場合には、完全な健全性を確保できません。

ストアード・プロシージャを NOT FENCED として登録するには、SYSADM 権限、DBADM 権限、または特殊な権限 (CREATE\_NOT\_FENCED) が必要です。LANGUAGE OLE または NOT THREADSAFE を指定した関数には、FENCED のみを指定できます。

#### THREADSAFE または NOT THREADSAFE

プロシージャを他のルーチンと同じプロセスで実行しても安全か (THREADSAFE)、そうでないか (NOT THREADSAFE) を指定します。

プロシージャが OLE 以外の LANGUAGE で定義される場合:

- プロシージャが THREADSAFE として定義されている場合には、データベース・マネージャーは他のルーチンと同じプロセスにプロシージャを呼び出すことができます。一般に、スレッド・セーフになるには、プロシージャはどのグローバルあるいは静的データ域をも使用してはなりません。多くのプログラミン

## CREATE PROCEDURE (外部)

グ解説書には、スレッド・セーフ・ルーチンの作成に関する説明が含まれています。FENCED および NOT FENCED プロシーチャーの両方を THREADSAFE にすることができます。

- プロシーチャーが NOT THREADSAFE に定義される場合には、データベース・マネージャーは他のルーチンと同じプロセスにプロシーチャーを決して呼び出しません。

FENCED プロシーチャーの場合、LANGUAGE が JAVA なら THREADSAFE がデフォルトです。これ以外のすべての言語の場合は、NOT THREADSAFE がデフォルトです。プロシーチャーが LANGUAGE OLE に定義される場合には、THREADSAFE は指定できません (SQLSTATE 42613)。

NOT FENCED プロシーチャーの場合には、THREADSAFE がデフォルトです。NOT THREADSAFE を指定することはできません (SQLSTATE 42613)。

### INHERIT SPECIAL REGISTERS

このオプション文節は、プロシーチャーの更新可能な特殊レジスターが、呼び出しステートメントの環境からの初期値を継承するように指定します。

特殊レジスターに対する変更が、プロシーチャーの呼び出し元に戻されることはありません。

更新不能の特殊レジスター (日時特殊レジスターなど) は、現在実行中のステートメントのプロパティを反映するので、デフォルト値に設定されます。

### PARAMETER STYLE

この文節は、ストアード・プロシーチャーとの間でパラメーターを渡し、値を戻すのに用いる規則を指定するのに使用されます。

#### DB2GENERAL

ストアード・プロシーチャーは、Java メソッドを使用するために定義された規則に従ったパラメーターの受け渡し規則を使用します。これは、LANGUAGE JAVA を使用する場合にだけ指定する必要があります。

#### DB2SQL

CALL ステートメントのパラメーターの他に、以下の引き数がストアード・プロシーチャーに渡されます。

- CALL ステートメントの各パラメーターの NULL 標識を含むベクトル
- DB2 へ戻される SQLSTATE
- ストアード・プロシーチャーの修飾名
- ストアード・プロシーチャーの特定名
- DB2 へ戻される SQL 診断ストリング

これは、LANGUAGE C、COBOL、または OLE を使用する場合にだけ、指定することができます。

**GENERAL**

ストアード・プロシージャは、パラメーター受け渡しメカニズムを使用します。ここでは、ストアード・プロシージャは CALL で指定したパラメーターを受け取ります。パラメーターは言語ごとに直接に渡されることになっているので、SQLDA 構造は使われません。これは、LANGUAGE C を使用する場合にだけ、指定することができます。

NULL 標識がプログラムに直接渡されることはありません。

**GENERAL WITH NULLS**

GENERAL で指定した CALL ステートメントのパラメーターの他に、別の引き数がストアード・プロシージャに渡されます。この別の引き数は、CALL ステートメントの各パラメーター用の、NULL 標識のベクトルです。これは、C では short int の配列になります。これは、LANGUAGE C を使用する場合にだけ、指定することができます。

**JAVA**

ストアード・プロシージャは、Java 言語および SQLj ルーチンの仕様に準拠する規則に従ったパラメーターの受け渡し規則を使用します。IN/OUT および OUT パラメーターは、戻り値を処理するために単一項目配列として渡されます。これは、LANGUAGE JAVA を使用する場合にだけ指定する必要があります。

PARAMETER STYLE JAVA プロシージャでは、DBINFO または PROGRAM TYPE 文節はサポートされていません。

**SQL**

CALL ステートメントのパラメーターの他に、以下の引き数がストアード・プロシージャに渡されます。

- CALL ステートメントの各パラメーターの NULL 標識
- DB2 へ戻される SQLSTATE
- ストアード・プロシージャの修飾名
- ストアード・プロシージャの特定名
- DB2 へ戻される SQL 診断ストリング

これは、LANGUAGE C、COBOL、または OLE を使用する場合にだけ、指定することができます。

**PROGRAM TYPE**

ストアード・プロシージャでのパラメーターのスタイルが、メインルーチンなのかサブルーチンなのかを指定します。

**SUB**

ストアード・プロシージャのパラメーターは、別々の引き数として渡されません。

## CREATE PROCEDURE (外部)

### MAIN

ストアード・プロシージャのパラメータは、引き数カウンター、および引き数のベクトルとして渡されます (argc、argv)。呼び出すストアード・プロシージャの名前も、"main" となります。このタイプのストアード・プロシージャは、独立した実行可能ファイルではなく、共有ライブラリーと同じ方法で作成する必要があります。

PROGRAM TYPE のデフォルトは SUB です。PROGRAM TYPE MAIN は、LANGUAGE C または COBOL で、なおかつ PARAMETER STYLE GENERAL、GENERAL WITH NULLS、SQL、または DB2SQL の場合だけ有効です。

### DBINFO または NO DBINFO

DB2 において既知である特定の情報が呼び出されたときに、その情報を追加の呼び出し時引き数としてストアード・プロシージャに渡すか (DBINFO)、または渡さないか (NO DBINFO) を指定します。NO DBINFO がデフォルト値です。DBINFO は、LANGUAGE OLE ではサポートされません (SQLSTATE 42613)。これは PARAMETER STYLE JAVA、または DB2GENERAL でもサポートされません。

DBINFO を指定すると、以下の情報を含む構造がストアード・プロシージャに渡されます。

- データベース名 - 現在接続されているデータベースの名前。
- アプリケーション ID - データベースへの接続ごとに確立された、固有のアプリケーション ID。
- アプリケーション許可 ID - アプリケーション実行時の許可 ID。
- コード・ページ - データベースのコード・ページを識別します。
- データベースのバージョン / リリース - ストアード・プロシージャを呼び出すデータベース・サーバーのバージョン、リリース、および修正レベルを識別します。
- プラットフォーム - サーバーのプラットフォーム・タイプが入ります。

DBINFO 構造はすべての外部ルーチンで共通で、プロシージャに関係ない追加のフィールドを含みます。

### 注:

- **互換性**
  - DB2 UDB (OS/390 および z/OS 版) との互換性:
    - 以下の構文はデフォルトの振る舞いとして受け入れられます。
      - ASUTIME NO LIMIT
      - COMMIT ON RETURN NO
      - NO COLLID



- STAY RESIDENT NO
- 以前のバージョンの DB2 との互換性:
  - DYNAMIC RESULT SETS の代わりに RESULT SETS を指定できます。
  - CALLED ON NULL INPUT の代わりに NULL CALL を指定できます。
  - DB2GENERAL の代わりに DB2GENRL を指定できます。
  - GENERAL の代わりに SIMPLE CALL を指定できます。
  - GENERAL WITH NULLS の代わりに SIMPLE CALL WITH NULLS を指定できます。
  - PARAMETER STYLE DB2DARI はサポートされています。
- まだ存在していないスキーマ名を用いてプロシージャを作成すると、ステートメントの許可 ID に IMPLICIT\_SCHEMA 権限がある場合に限り、そのスキーマが暗黙に作成されます。そのスキーマの所有者は SYSIBM になります。スキーマに対する CREATEIN 特権が PUBLIC に付与されます。
- 特権
  - プロシージャの定義者は、プロシージャに対する EXECUTE WITH GRANT OPTION 特権と、プロシージャをドロップする権利を常に与えられます。
  - プロシージャを SQL ステートメントで使用する時点で、プロシージャの定義者はそのプロシージャによって使用されるすべてのパッケージに対して EXECUTE 特権を持っていない限りなりません。

**例:**

例 1: Java で書かれたストアード・プロシージャのプロシージャ定義を作成します。このプロシージャは、パーツ番号を渡されて、パーツの価格と現在入手可能な数量を戻します。

```
CREATE PROCEDURE PARTS_ON_HAND (IN PARTNUM INTEGER,
 OUT COST DECIMAL(7,2),
 OUT QUANTITY INTEGER)
EXTERNAL NAME 'parts.onhand'
LANGUAGE JAVA PARAMETER STYLE JAVA
```

例 2: C で書かれたストアード・プロシージャのプロシージャ定義を作成します。このプロシージャは、部品番号を渡されて、部品を構成するパーツの数とパーツの合計価格、およびパーツ番号、数量、各パーツの単価をリストする結果セットを戻します。

```
CREATE PROCEDURE ASSEMBLY_PARTS (IN ASSEMBLY_NUM INTEGER,
 OUT NUM_PARTS INTEGER,
 OUT COST DOUBLE)
EXTERNAL NAME 'parts!assembly'
DYNAMIC RESULT SETS 1 NOT FENCED
LANGUAGE C PARAMETER STYLE GENERAL
```

**関連資料:**

## CREATE PROCEDURE (外部)

- 1360 ページの『ルーチンで使用可能な SQL ステートメント』
- *SQL* リファレンス 第 1 巻 の『特殊レジスター』

### 関連サンプル:

- 『spcreate.db2 -- How to catalog the stored procedures contained in spserver.sqc 』
- 『spcreate.db2 -- Catalog the DB2 CLI stored procedures contained in spserver.sqc 』
- 『SpCreate.db2 -- How to catalog the stored procedures contained in SpServer.java 』
- 『SpCreate.db2 -- How to catalog the stored procedures contained in SpServer.sqlj 』

## CREATE PROCEDURE (SQL)

CREATE PROCEDURE (SQL) ステートメントは、アプリケーション・サーバーで SQL プロシージャを定義するのに使用します。

### 呼び出し:

このステートメントはアプリケーション・プログラムに組み込むことができ、また動的 SQL ステートメントを使用して出すことができます。 DYNAMICRULES の実行動作がパッケージに効力を持つ場合にのみ、動的に準備できる実行可能ステートメントです (SQLSTATE 42509)。

### 許可:

ステートメントの許可 ID によって保持されている特権には、少なくとも以下のいずれかが含まれていなければなりません。

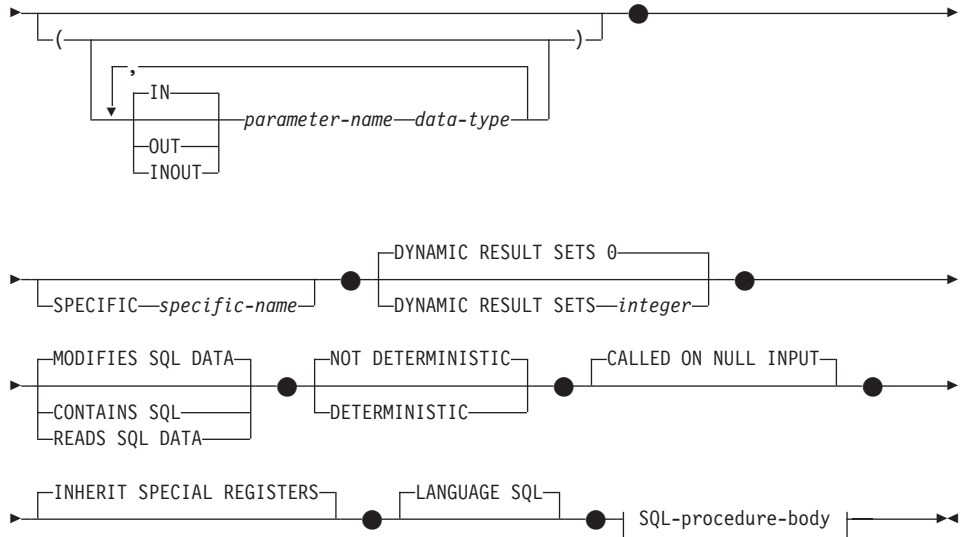
- SYSADM または DBADM 権限
- データベースに対する IMPLICIT\_SCHEMA 権限 (プロシージャの暗黙または明示のスキーマ名が存在しない場合)
- スキーマに対する CREATEIN 特権 (プロシージャのスキーマ名が既存のスキーマを指している場合)

このステートメントの許可 ID に SYSADM 権限または DBADM 権限がない場合には、ステートメントの許可 ID が持つ特権に、プロシージャ本体で指定される SQL ステートメントを呼び出すのに必要なすべての特権が含まれていることも必要です。

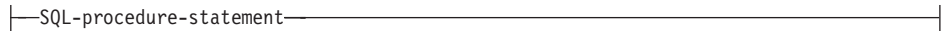
許可 ID の権限が不十分で、操作を実行できない場合には、エラー (SQLSTATE 42502) になります。

### 構文:

►—CREATE PROCEDURE—*procedure-name*—►



**SQL-procedure-body:**



**説明:**

*procedure-name*

定義するプロシージャの名前を指定します。この名前は、プロシージャを指定する修飾または非修飾の名前です。 *procedure-name* (プロシージャ名) の非修飾形式は SQL ID です (最大長 128)。動的 SQL ステートメントでは、CURRENT SCHEMA 特殊レジスタが、修飾子のないオブジェクト名の修飾子として使用されます。静的 SQL ステートメントでは、QUALIFIER プリコンパイル / パインド・オプションによって、修飾子のないオブジェクト名の修飾子が暗黙指定されます。修飾形式は、*schema-name* の後にピリオドと SQL ID が続きます。

暗黙または明示の修飾子を含む名前と、パラメーターの数との組み合わせは、カタログにすでに記述されているプロシージャを指定するものであってはなりません (SQLSTATE 42723)。非修飾名とパラメーターの数との組み合わせは、そのスキーマ内では固有ですが、複数のスキーマ間で固有である必要はありません。

2 つの部分から成る名前を指定する場合、“SYS” で始まる *schema-name* (スキーマ名) は使用できません。使用した場合、エラー (SQLSTATE 42939) になります。

**(IN | OUT | INOUT parameter-name data-type,...)**

プロシージャのパラメーターを指定し、各パラメーターのモード、名前、およびデータ・タイプを指定します。このリストには、プロシージャが予期する各パラメーターごとに 1 つの項目を指定する必要があります。

## CREATE PROCEDURE (SQL)

パラメーターのないプロシージャも登録可能です。この場合、指定するデータ・タイプがない場合でも、括弧はコーディングする必要があります。たとえば、以下のようにします。

```
CREATE PROCEDURE SUBWOOFER() ...
```

1 つのスキーマに同じ名前の 2 つのプロシージャがある場合、パラメーターの数をまったく同一にすることはできません。シグニチャーが重複していると、SQL エラー (SQLSTATE 42723) になります。

たとえば、以下のステートメントの場合、

```
CREATE PROCEDURE PART (IN NUMBER INT, OUT PART_NAME CHAR(35)) ...
CREATE PROCEDURE PART (IN COST DECIMAL(5,3), OUT COUNT INT) ...
```

### IN | OUT | INOUT

パラメーターのモードを指定します。

**IN** パラメーターをプロシージャの入力パラメーターとして指定します。プロシージャ内でパラメーターに加えられるすべての変更は、制御が戻されると SQL アプリケーションの呼び出しは行なえなくなります。デフォルトは IN です。

**OUT** パラメーターをプロシージャの出力パラメーターとして指定します。

**INOUT** パラメーターを、プロシージャの入力および出力パラメーターの両方として指定します。

### *parameter-name*

パラメーターの名前を指定します。パラメーター名は、プロシージャで固有でなければなりません (SQLSTATE 42734)。

### *data-type*

パラメーターのデータ・タイプを指定します。

- CREATE TABLE ステートメントのデータ・タイプの定義に指定することが可能で、プロシージャの作成に使用されている言語に対応するものがある SQL データ・タイプ指定と省略形を指定できます。
- ユーザー定義データ・タイプはサポートされていません (SQLSTATE 42601)。
- LONG VARCHAR および LONG VARGRAPHIC データ・タイプは、SQL プロシージャのパラメーター・タイプとしてサポートされていません。

### **SPECIFIC** *specific-name*

定義するプロシージャのインスタンスに対する固有名を指定します。この特定名は、このプロシージャをドロップする場合、またはこのプロシージャにコメントを付ける場合に使用することができます。これは、プロシージャの呼び出しには使用できません。 *specific-name* (関数名) の非修飾形式は SQL ID です (最大長

18)。修飾形式は、*schema-name* の後にピリオドと SQL ID が続きます。暗黙または明示の修飾子も含めて、その名前が、アプリケーション・サーバーに存在する他のプロシージャ・インスタンスを指定するものであってはなりません。そうでない場合、エラー (SQLSTATE 42710) になります。

*specific-name* は、既存の *procedure-name* と同じにすることができます。

修飾子を指定しない場合、*procedure-name* に使用された修飾子が使用されます。修飾子を指定する場合は、*procedure-name* の明示修飾子または暗黙修飾子と同じでなければなりません。そうでない場合、エラー (SQLSTATE 42882) になります。

*specific-name* の指定がない場合、固有の名前がデータベース・マネージャーによって生成されます。生成される固有の名前は、SQL の後に文字のタイム・スタンプが続く名前です (SQLyymmddhhmmsshhn)。

#### **DYNAMIC RESULT SETS** *integer*

ストアード・プロシージャから戻される結果セットの上限の見積もりを指定します。

#### **CONTAINS SQL、READS SQL DATA、MODIFIES SQL DATA**

プロシージャに含まれる SQL ステートメントのデータ・アクセスのレベルを示します。

##### **CONTAINS SQL**

SQL データの読み取りも変更も行わない SQL ステートメントを、ストアード・プロシージャで実行できることを指定します (SQLSTATE 38004 または 42985)。どのストアード・プロシージャでもサポートされていないステートメントは、これとは異なるエラーを戻します (SQLSTATE 38003 または 42985)。

##### **READS SQL DATA**

SQL データを変更しない SQL ステートメントを、ストアード・プロシージャで実行できることを指定します (SQLSTATE 38002 または 42985)。どのストアード・プロシージャでもサポートされていないステートメントは、これとは異なるエラーを戻します (SQLSTATE 38003 または 42985)。

##### **MODIFIES SQL DATA**

このストアード・プロシージャは、ストアード・プロシージャでサポートされていないステートメント以外のすべての SQL ステートメントを実行できることを指定します (SQLSTATE 38003 または 42985)。

#### **DETERMINISTIC** または **NOT DETERMINISTIC**

この文節は、同一の引き数値に対してプロシージャが常に同じ結果を戻すか (DETERMINISTIC)、それとも状態値に依存してプロシージャの結果が影響を受けるか (NOT DETERMINISTIC) を指定します。つまり DETERMINISTIC が指定されたプロシージャは、同じ入力を指定して正しく呼び出した場合に常に同じ結果を戻します。

現在、この文節はストアード・プロシージャの処理に影響を与えません。

## CREATE PROCEDURE (SQL)

### CALLED ON NULL INPUT

CALLED ON NULL INPUT は、ストアード・プロシージャーに常に適用されます。これは、任意の引き数が NULL かどうかにかかわらず、ストアード・プロシージャーが呼び出されることを意味します。OUT または INOUT パラメーターは、NULL 値を戻す場合も、通常の (NULL 値以外の) 値を戻す場合もあります。NULL の引き数値の有無のテストはストアード・プロシージャーで行う必要があります。

### INHERIT SPECIAL REGISTERS

このオプション文節は、プロシージャーの更新可能な特殊レジスターが、呼び出しステートメントの環境からの初期値を継承するよう指定します。ネストされたオブジェクト (たとえば、トリガーまたはビュー) に呼び出されるルーチンの場合、初期値は (オブジェクト定義から継承するのではなく) 実行時環境から継承します。

特殊レジスターに対する変更が、プロシージャーの呼び出し元に戻されることはありません。

更新不能の特殊レジスター (日時特殊レジスターなど) は、現在実行中のステートメントのプロパティを反映するので、デフォルト値に設定されます。

### LANGUAGE SQL

この文節は、プロシージャー本体が SQL 言語に書き込まれるように指定するのに使用します。

### SQL-procedure-body

SQL プロシージャーの本体である SQL ステートメントを指定します。複合ステートメント内に複数の SQL-procedure-statement を指定することができます。

#### 注:

##### • 互換性

- DB2 UDB (OS/390 および z/OS 版) との互換性:
  - 以下の構文はデフォルトの振る舞いとして受け入れられます。
    - ASUTIME NO LIMIT
    - COMMIT ON RETURN NO
    - NO COLLID
    - STAY RESIDENT NO
- 以前のバージョンの DB2 との互換性:
  - DYNAMIC RESULT SETS の代わりに RESULT SETS を指定できます。
  - CALLED ON NULL INPUT の代わりに NULL CALL を指定できます。
- まだ存在していないスキーマ名を用いてプロシージャーを作成すると、ステートメントの許可 ID に IMPLICIT\_SCHEMA 権限がある場合に限り、そのスキーマが暗黙に作成されます。そのスキーマの所有者は SYSIBM になります。スキーマに対する CREATEIN 特権が PUBLIC に付与されます。

## - 特権

プロシージャの定義者は、プロシージャに対する EXECUTE WITH GRANT OPTION 特権と、プロシージャをドロップする権利を常に与えられます。

## 例:

例 1: 社員の給与の中央値を戻す SQL プロシージャを作成します。給与の中央値を超える給与を得ている全社員の氏名、肩書き、および給与の入った結果セットを戻します。

```
CREATE PROCEDURE MEDIAN_RESULT_SET (OUT medianSalary DOUBLE)
 RESULT SETS 1
 LANGUAGE SQL
BEGIN
 DECLARE v_numRecords INT DEFAULT 1;
 DECLARE v_counter INT DEFAULT 0;

 DECLARE c1 CURSOR FOR
 SELECT CAST(salary AS DOUBLE)
 FROM staff
 ORDER BY salary;
 DECLARE c2 CURSOR WITH RETURN FOR
 SELECT name, job, CAST(salary AS INTEGER)
 FROM staff
 WHERE salary > medianSalary
 ORDER BY salary;

 DECLARE EXIT HANDLER FOR NOT FOUND
 SET medianSalary = 6666;

 SET medianSalary = 0;
 SELECT COUNT(*) INTO v_numRecords
 FROM STAFF;
 OPEN c1;
 WHILE v_counter < (v_numRecords/2 + 1)
 DO
 FETCH c1 INTO medianSalary;
 SET v_counter = v_counter + 1;
 END WHILE;
 CLOSE c1;
 OPEN c2;
END
```

## 関連資料:

- 1360 ページの『ルーチンで使用可能な SQL ステートメント』
- SQL リファレンス 第 1 巻 の『特殊レジスター』

## 関連サンプル:

- 『basecase.db2 -- To create the UPDATE\_SALARY SQL procedure 』
- 『nestcase.db2 -- To create the BUMP\_SALARY SQL procedure 』

## CREATE PROCEDURE (SQL)

- 『nestedsp.db2 -- To create the OUT\_AVERAGE, OUT\_MEDIAN and MAX\_SALARY SQL procedures』
- 『resultset.db2 -- To create the MEDIAN\_RESULT\_SET SQL procedure 』

---

## メッセージ解説書

### DBI メッセージ

このセクションでは、DB2 製品のインストールおよび構成時に生成される可能性のあるエラー・メッセージについて説明します。メッセージは番号順にリストされています。

---

#### DBI1001I 使用法 :

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

**説明:** db2icrt コマンドに無効な引き数が入力されました。このコマンドの有効な引き数は以下のとおりです。

**-hl-?** 使用法情報を表示する

**-d** デバッグ・モードをオンにする

**-a AuthType**

インスタンスの認証タイプ  
(SERVER、CLIENT、または  
SERVER\_ENCRYPT)

**-p PortName**

このインスタンスで使用されるポート名  
あるいはポート番号

**-s InstType**

作成されるインスタンスのタイプ  
(wse、ese、あるいは client)

**-u FencedID**

分離した UDF および分離したストアド・プロシージャが実行されるシステム下のユーザー名 DB2 クライアントのみがインストールされている場合はこのフラグは不要です。

**-w WordWidth**

作成されるインスタンスの幅 (バイト単位: 32 または 64) です。(AIX、Solaris オペレーティング環境では) これを機能させるためには、DB2 の前提条件バージョン (32 ビットまたは 64 ビット) がインストールされていなければなりません。デフォルトの幅の値は 32 ビットです。

**InstName**

インスタンスの名前

**ユーザーの処置:** このコマンドに関する詳細については、「概説およびインストール」マニュアルを参照してください。以下のようにコマンドを入力し直してください。

```
db2icrt [-a AuthType]
 -u FencedID InstName
```

---

#### DBI1002I 使用法 :

```
db2uit [-d] [-Q] [-D]
 [-q <field>]
 [-i <id>]
 [-a <action>]
 [-r <runlevels>]
 [-p <process>]
```

**説明:** 無効な引き数が db2uit コマンドに対して入力されました。このコマンドの有効な引き数は以下のとおりです。



- d** デバッグ・モードをオンにする
- q** 検索パラメーターを指定してフィールドを照会する。 **-q** が指定されていない場合 (**-i ID** がすでに指定されている場合には) 更新、または挿入になります。有効フィールド: i, r, a, p
- i** エントリーの ID
- r** 実行レベル
- a** 実行するアクション
- p** 立ち上げる処理
- Q** 静止: 出力なし
- D** レコードの削除

**FileName**

変更するファイル

戻りコード:

- q** 一致するフィールドがなければ非ゼロ、あればゼロ。

**update/insert**

ファイルを更新できなかった場合は非ゼロ。それ以外はゼロ。

**ユーザーの処置:** このコマンドに関する詳細については、「概説およびインストール」マニュアルを参照してください。このコマンドを再度試行します。

**DBI1003I 使用法 :**

```
dlfmcrct [-b BackupDir]
 -p PortName
 -g DLFMGid
 DLFMAdmin
```

**説明:** 無効な引き数が dlfmcrct コマンドに対して入力されました。このコマンドの有効な引き数は以下のとおりです。

- hl-?** 使用法情報を表示する

- p** このインスタンスで使用されるポート名あるいはポート番号
- b** BackupDir は、dlfm データベースのバックアップを作成するために使用されるディレクトリーです。

**-g DLFMGid**

DLFM 管理グループ (このグループは DLFMAdmin の 2 次グループでなければなりません)

**DLFMAdmin**

DB2 Data Links Manager 管理者の名前

**ユーザーの処置:** 以下のようにコマンドを入力し直してください。

```
dlfmcrct -p PortNum -g DLFMGid DLFMAdmin
```

**DBI1005I 使用法 :**

**db2idrop InstName**

**説明:** 間違った引き数が db2idrop コマンドに対して入力されました。このコマンドの有効な引き数は以下のとおりです。

- hl-?** 使用法情報を表示する
- d** デバッグ・モードをオンにする

**InstName**

インスタンスの名前

**ユーザーの処置:** 以下のようにコマンドを入力し直してください。

```
db2idrop InstName
```

**DBI1006I 使用法 :**

**db2idrop [-f] InstName**

**説明:** 間違った引き数が db2idrop コマンドに対して入力されました。このコマンドの有効な引き数は以下のとおりです。

- hl-?** 使用法情報を表示する

- d** デバッグ・モードをオンにする
- f** 強制アプリケーション・フラグこのフラグが指定された場合、インスタンスを使用しているすべてのアプリケーションは強制的に終了します。

#### InstName

インスタンスの名前

**ユーザーの処置:** 以下のようにコマンドを入力し直してください。

```
db2idrop [-f] InstName
```

---

#### DBI1007I 使用法 :

##### dlfmdrop DLFMAdmin

**説明:** 無効な引き数が dlfmdrop コマンドに対して入力されました。このコマンドの有効な引き数は以下のとおりです。

**-hl-?** 使用法情報を表示する

#### DLFMAdmin

DB2 Data Links Manager 管理者の名前

**ユーザーの処置:** 以下のようにコマンドを入力し直してください。

```
dlfmdrop DLFMAdmin
```

---

#### DBI1008E インストールするメディアと AIX のバージョンが一致しません。

**説明:** AIX に DB2 64 ビットをインストールするには、AIX バージョン 4 および AIX バージョンの CD が必要です。

db2setup は、ご使用のシステムが AIX バージョン 5 を実行していることを検出しましたが、AIX バージョン 4 用の DB2 CD からインストールしています。

**ユーザーの処置:** AIX バージョン 5 用の DB2 CD を挿入して、インストールをやり直してください。

---

#### DBI1009E インストールするメディアと AIX のバージョンが一致しません。

**説明:** AIX に DB2 64 ビットをインストールするには、AIX バージョン 4 および AIX バージョンの CD が必要です。

db2setup は、ご使用のシステムが AIX バージョン 4 を実行していることを検出しましたが、AIX バージョン 5 用の DB2 CD からインストールしています。

**ユーザーの処置:** AIX バージョン 4 用の DB2 CD を挿入して、インストールをやり直してください。

---

#### DBI1011I 使用法 :

```
db2iupdt [-hl-?] [-d] [-k] [-s] [-D]
 [-a AuthType]
 [-w WordWidth]
 [-u FencedID]
 {-eInstName}
```

**説明:** 間違った引き数が db2iupdt コマンドに入力されました。このコマンドの有効な引き数は以下のとおりです。

**-hl-?** 使用法情報を表示する

**-d** デバッグ・モードをオンにする

**-k** 更新中の現行インスタンス・タイプを保持する

**-s** 既存の SPM ログ・ディレクトリーを無視する

**-D** 現在のコード・レベルより低いコードにインスタンスを更新する。このオプションは、複数のフィックスパック環境でのみ有効です。

#### -a AuthType

インスタンスの認証タイプ (SERVER、CLIENT、DCS、SERVER\_ENCRYPT、または DCS\_ENCRYPT)。デフォルトは SERVER です。

## **-w WordWidth**

作成されるインスタンスの幅 (バイト単位: 32 または 64) です。 (AIX、Solaris オペレーティング環境) **-e** が指定されている場合は無視されます。これを機能させるためには、DB2 の前提条件バージョン (32 ビットまたは 64 ビット) がインストールされていないとなりません。

## **-u FencedID**

分離した UDF および分離したストアード・プロシージャーが実行されるシステム下のユーザー名 DB2 クライアントのみがインストールされている場合はこのフラグは不要です。

**-e** すべてのインスタンスが更新されるよう指定します。

## **InstName**

インスタンスの名前

**ユーザーの処置:** 正しいオプションおよび引き数でコマンドを再入力してください。

---

## **DBI1012I 使用法 :**

**db2iupdt [-f] -u FencedID  
InstName**

**説明:** 間違った引き数が db2iupdt コマンドに入力されました。このコマンドの有効な引き数は以下のとおりです。

**-hl-?** 使用法情報を表示する

**-d** デバッグ・モードをオンにする

**-f** 強制アプリケーション・フラグこのフラグが指定された場合、インスタンスを使用しているすべてのアプリケーションは強制的に終了します。

## **-u FencedID**

分離した UDF および分離したストアード・プロシージャーが実行されるシステム

下のユーザー名 DB2 クライアントのみがインストールされている場合はこのフラグは不要です。

## **InstName**

インスタンスの名前

**ユーザーの処置:** 以下のようにコマンドを入力し直してください。

db2iupdt [-f] -u FencedID InstName

---

## **DBI1013I 使用法 :**

**dlfmupdt [-hl-?] DLFMAdmin**

**説明:** 無効な引き数が dlfmupdt コマンドに対して入力されました。このコマンドの有効な引き数は以下のとおりです。

**-hl-?** 使用法情報を表示する

## **DLFMAdmin**

DB2 Data Links Manager 管理者の名前

**ユーザーの処置:** 以下のようにコマンドを入力し直してください。

dlfmupdt [-hl-?] DLFMAdmin

---

## **DBI1014E 32 ビットへのインスタンス移行はサポートされていません。**

**説明:** 64 ビットから 32 ビットへの移行は、このバージョンではサポートされていません。

**ユーザーの処置:** 処置は必要ありません。

---

## **DBI1015I 使用法 :**

**db2imigr [-a AuthType]  
-u FencedID  
-g DLFMGid  
InstName**

**説明:** 間違った引き数が入力されました。このコマンドの有効な引き数は以下のとおりです。

**-hl-?** 使用法情報を表示する

**-d** デバッグ・モードをオンにする

**-a AuthType**

インスタンスの認証タイプ (SERVER、CLIENT、または SERVER\_ENCRYPT)

**-u FencedID**

分離した UDF および分離したストアド・プロシージャが実行されるシステム下のユーザー名 DB2 クライアントのみがインストールされている場合はこのフラグは不要です。

**-g DFLMGid**

DLFM インスタンスのバージョン 7 以前のバージョンを移行する場合、指定が必要

**InstName**

インスタンスの名前

**ユーザーの処置:** このコマンドに関する詳細については、「概説およびインストール」マニュアルを参照してください。以下のようにコマンドを入力し直してください。

```
db2imigr [-a AuthType]
 [-u FencedID InstName]
```

---

**DBI1018I 使用法 :**

**db2istop [-hl-?] [-d] [-f] InstName**

**説明:** 間違った引き数が入力されました。このコマンドの有効な引き数は以下のとおりです。

**-hl-?** 使用法情報を表示する

**-d** デバッグ・モードをオンにする

**-f** 強制アプリケーション・フラグこのフラグが指定された場合、インスタンスを使用しているすべてのアプリケーションは強制的に終了します。

**InstName**

インスタンスの名前

**ユーザーの処置:** 以下のようにコマンドを入力し直してください。

```
db2istop [-hl-?] [-d] [-f] InstName
```

---

**DBI1020I 使用法 :**

**db2setup [-hl-?] [-d] [-r RespFile]**

**説明:** 間違った引き数が入力されました。このコマンドの有効な引き数は以下のとおりです。

**-hl-?** 使用法情報を表示する

**-d** デバッグ・モードをオンにする

**-r RespFile**

DB2 のインストールに使用される応答ファイルの名前

**ユーザーの処置:** 以下のようにコマンドを入力し直してください。

```
db2setup [-hl-?] [-d] [-r RespFile]
```

---

**DBI1021I 使用法 :**

**db2imigrev [-hl-?] [-m Version] InstName**

**説明:** 間違った引き数が入力されました。このコマンドの有効な引き数は以下のとおりです。

**-hl-?** 使用法情報を表示する

**-d** デバッグ・モードをオンにする

**-m** DB2 のバージョンを指定します。

**Version**

逆方向へ移行されるインスタンスのバージョンです。

**InstName**

DB2 のバージョン 8 から以前のバージョンに移行されるインスタンスの名前。

**ユーザーの処置:** 以下のようにコマンドを入力し直してください。

db2imigrev [-hl-?] InstName

---

**DBI1022I**    使用法 :

**db2imigrev [-hl-?]  
[-f] InstName**

**説明:** 間違った引き数が入力されました。このコマンドの有効な引き数は以下のとおりです。

- hl-?**    使用法情報を表示する
- d**        デバッグ・モードをオンにする
- f**        強制アプリケーション・フラグこのフラグが指定された場合、インスタンスを使用しているすべてのアプリケーションは強制的に終了します。

**InstName**

DB2 のバージョン 8 から以前のバージョンに移行されるインスタンスの名前。

**ユーザーの処置:** 以下のようにコマンドを入力し直してください。

db2imigrev [-hl-?] [-f] InstName

---

**DBI1023I**    使用法 :

**db2ginfo [-hl-?] [-y]  
[-c InstName  
[-a AuthType]  
[-u FencedID]]  
OutputDir**

**説明:** db2ginfo コマンドに無効な引き数が入力されました。このコマンドの有効な引き数は以下のとおりです。

- hl-?**    使用法を表示する
- y**        スクリプト表示 (警告を表示するのみ)。
- c InstName**  
      テストするインスタンスの指定。

**-a AuthType**

認証タイプ (SERVER、CLIENT、または SERVER\_ENCRYPT) を指定。

**-u FencedID**

隔離したユーザー ID を指定する

**OutputDir**

出力ファイル dbginfo.txt が置かれるディレクトリー。

**ユーザーの処置:** 以下のようにコマンドを入力し直してください。

db2ginfo [-hl-?] [-y]  
      [-c InstName [-a AuthType]  
      [-u FencedID]] OutputDir

---

**DBI1024I**    使用法 :

**db2iauto [-hl-?] -onl-off  
InstName**

**説明:** 間違った引き数が入力されました。このコマンドの有効な引き数は以下のとおりです。

- hl-?**    使用法情報を表示する
- onl-off**  
      インスタンスの自動開始を使用可能 / 使用不可にする

**InstName**

インスタンスの名前

**ユーザーの処置:** 以下のようにコマンドを入力し直してください。

db2iauto [-hl-?] -onl-off InstName

---

**DBI1025I**    使用法 :

**dascrt [-d] -u ASUser**

**説明:** dascrt コマンドに対して無効な引き数が入力されました。このコマンドの有効な引き数は以下のとおりです。

**-d** DB2 サービスで使用するために、デバッグ・モードに入る

**-u ASUser**

DAS の実行に使用するユーザー

**ユーザーの処置:** このコマンドに関する詳細については、「概説およびインストール」マニュアルを参照してください。以下のようにコマンドを入力し直してください。

```
dasprt -u ASUser
```

---

**DBI1026I** 使用法 :

**dlfmfsmd [-hl-?] [-j] dlfsMntPt**

**説明:** dlfmfsmd コマンドに無効な引き数が入力されました。このコマンドの有効な引き数は以下のとおりです。

**-hl-?** 使用法情報を表示する

**-j** 名前付きファイル・システムを dlfs から jfs (AIX) または ufs (Solaris オペレーティング環境) に変更する

**dlfsMntPt**

Data Links Manager フィルター・ファイル・システムのマウント・ポイント (スラッシュ / も含む)

**ユーザーの処置:** 以下のようにコマンドを入力し直してください。

```
dlfmfsmd dlfsMntPt
```

---

**DBI1027I** 使用法 :

**dasdrop [-d]**

**説明:** dasdrop コマンドに対して無効な引き数が入力されました。このコマンドの有効な引き数は以下のとおりです。

**-d** DB2 サービスで使用するために、デバッグ・モードに入る

**ユーザーの処置:** このコマンドに関する詳細につ

いては、「概説およびインストール」マニュアルを参照してください。以下のようにコマンドを入力し直してください。

```
dasdrop
```

---

**DBI1028I** 使用法 :

**dasupdt [-d]**

**説明:** dasupdt コマンドに無効な引き数が入力されました。このコマンドの有効な引き数は以下のとおりです。

**-d** DB2 サービスで使用するために、デバッグ・モードに入る

**ユーザーの処置:** このコマンドに関する詳細については、「概説およびインストール」マニュアルを参照してください。以下のようにコマンドを入力し直してください。

```
dasupdt
```

---

**DBI1029I** 使用法 :

**db2nqadm [start|stop]**

**説明:** 無効な引き数が db2nqadm コマンドに対して入力されました。このコマンドの有効な引き数は以下のとおりです。

**start** すべての NetQ サービスおよびデーモンを開始する

**stop** すべての NetQ サービスおよびデーモンを停止する

いずれかの引き数を指定してください。

**ユーザーの処置:** 正しい引き数で、コマンドを再入力してください。

---

**DBI1030E** カーネル・パラメーター *name* は *value* に設定してください。

**説明:** DB2 ではある特定のカーネル・パラメーターを更新する必要があります。

#### ユーザーの処置:

- 「概説およびインストール」マニュアルの記述どおりに、必要なすべてのカーネル・パラメーターを更新します。
- システムをリブートします。
- このコマンドを再度試行します。

---

**DBI1031E** 選択した製品をインストールするには、まず **DB2 UDB Workgroup Server Edition** または **DB2 UDB Enterprise Server Edition** のいずれかをインストールする必要があります。

#### ユーザーの処置:

- DB2 UDB Workgroup Server Edition または DB2 UDB Enterprise Server Edition のいずれかをインストールしてください。
- 製品をもう一度インストールしてください。

---

**DBI1032E** 選択した製品は、バージョン 4.3 未満の **AIX** にはインストールできません。

**説明:** 選択した DB2 製品には、バージョン 4.3 以上の **AIX** が必要です。

---

**DBI1033E** 選択した製品には、**DB2 Enterprise Server Edition** を最初にインストールする必要があります。

#### ユーザーの処置:

- いずれかの DB2 UDB Enterprise Server Edition をインストールしてください。
- 製品をもう一度インストールしてください。

---

**DBI1034W** 前提条件パッケージが見つかりません。

**説明:** パッケージ “<pkg-name>” のいずれかをインストールするよう選択しましたが、インストー

ルされている “<pkg-name>” ファイル・セットのレベルが “<name>” よりも低くなっているか、またはこれを削除できません。

**ユーザーの処置:** “<pkg-name>” ファイル・セットがバージョン “<name>” またはそれ以上であることを確認してください。

“<pkg-name>” のレベルは DB2 のインストールに影響を与えません。

---

**DBI1035E** ファイル・システム *File-System* のマウントに失敗しました。

**説明:** 示されているファイル・システムをマウントしようとしたのですが、失敗しました。

**ユーザーの処置:** ファイル・システムが定義されていることを確認してください。マウント・コマンドのエラーを訂正して、やり直してください。

---

**DBI1036E** ファイル・システム *File-System* のアンマウントに失敗しました。

**説明:** 示されているファイル・システムをアンマウントしようとしたのですが、失敗しました。

**ユーザーの処置:** アンマウント・コマンドのエラーを訂正して、やり直してください。

---

**DBI1037E** ファイル・システム *File-System* は存在しません。

**説明:** 示されているファイル・システムは、このワークステーションに定義されていません。

**ユーザーの処置:** ワークステーションにファイル・システムを定義して、やり直してください。

---

**DBI1038E** ファイル・システム *File-System* を変更できません。

**ユーザーの処置:** ファイル・システム定義を備えたシステム・ファイルへの書き込み許可があるかどうか確認して、やり直してください。



---

**DBI1039W** 前提条件パッケージが見つかりません。

**説明:** パッケージ: “<name>” のいずれかをインストールするよう選択しましたが、“<name>” ファイル・セットのレベルが “<name>” よりも低くなっています。

**ユーザーの処置:** CD で提供されている “<name>” ファイル・セット・バージョン “<name>” をインストールしてください。

---

**DBI1040I** 以下のメッセージの翻訳バージョンはこのシステムでは使用できません。

**ユーザーの処置:** 以下のメッセージの翻訳バージョンはこのシステムでは使用できません。

PRODDIR/Readme/LOCALE ディレクトリーの “インストール・ノート” ファイルを参照してください。PRODDIR は製品のディレクトリーで LOCALE はロケール名です。たとえば、プロダクト・ディレクトリーは以下のものがあります。

/usr/opt/db2\_08\_xx、ここで xx は 01 または FPn、ここで n は FixPak 番号  
または /opt/IBM/db2/V8.x、ここで x は 1 または FPn、ここで n は FixPak 番号

---

**DBI1041E** ファイル・システム *inputMntPt* を *dlfs* に変換できません。

**説明:** ファイル・システム “/”、“/var”、“/usr”、および “/tmp” を *dlfs* ファイル・システムに変換できません。これは、*dlfs* ファイル・システムを作成すると起きるおそれのあるブート問題が起こらないようにするための措置です。

**ユーザーの処置:** 別のファイル・システムを指定してコマンドをもう一度実行してください。

---

**DBI1042E** 基本ファイル・システム *BaseFS* を *dlfs* に変更できません。

**説明:** Aix では、*vfs* が *fsm* または *jfs* の場合にのみ、ファイル・システムを *dlfs* に変更できます。他の *vfs* では *dlfs* はサポートされてません。

**ユーザーの処置:** *jfs* または *fsm* ファイル・システムでコマンドをもう一度実行してください。

---

**DBI1043E** 選択した製品の場合、最初に **DB2 Administration Client** をインストールする必要があります。

**ユーザーの処置:**

- DB2 Administration Client をインストールしてください。
- 製品をもう一度インストールしてください。

---

**DBI1050E** サポートされていないオペレーティング・システム - *OS-name*、バージョン *OS-ver*。

**説明:** 現行のオペレーティング・システムがサポートされていないか、あるいはオペレーティング・システムのこのバージョンがサポートされていません。「概説およびインストール」マニュアルを参照してください。

---

**DBI1051E** このプログラムを *root* で実行することはできません。

**説明:** このプログラムは *root* 以外のユーザー ID でのみ実行できます。

**ユーザーの処置:** *root* 以外のユーザー ID でログインしてから、コマンドを再発行してください。



---

**DBI1052E** このプログラムを実行するには **root** である必要があります。

**説明:** このプログラムは **root** ユーザー ID でのみ実行できます。このプログラムを実行するには特別な特権が必要です。

**ユーザーの処置:** **root** でログインしてから、コマンドを再発行してください。

---

**DBI1053E** *fsys-type* のファイル・システム・タイプはサポートされていません。

**説明:** このタイプのファイル・システム上にインスタンスのホーム・ディレクトリーを作成することはサポートされていません。インスタンスのホーム・ディレクトリーは、ローカル・ファイル・システムあるいは NFS が取り付けられたファイル・システムに存在する必要があります。

**ユーザーの処置:** サポートされているファイル・システムにホーム・ディレクトリーを変更してインスタンスの再作成をしてください。

---

**DBI1055E** メッセージ・ファイル *file-name* が見つかりません。

**説明:** インスタンス・コマンドが必要とするメッセージ・ファイルが、システムから失われていますが、これが削除されているか、またはデータベース製品が正しくインストールされていない可能性があります。

**ユーザーの処置:** メッセージ・ファイルを備えた製品オプションが正しくインストールされているかを調べてください。検証エラーがある場合は、製品オプションを再インストールしてください。

---

**DBI1057E** 必須ロケール *locale* が無効です。

**説明:** 必須ロケールのディレクトリー、`/usr/lib/locale/LANG/LC_MESSAGES` が存在しません。ここで **LANG** は現行ロケール名です。

**ユーザーの処置:** 指定されたロケールが正しくインストールされたかどうか確認してください。さ

らに、読み取りと実行の許可がそのディレクトリーに正しく設定されているかチェックしてください。

---

**DBI1060E** 無効なパッケージ名 *pkg-name*。

**説明:** 間違った名前が入力されました。そのパッケージは存在しないか、または名前が間違えて入力されました。

**ユーザーの処置:** 与えられたパッケージの名前が配布メディアに存在するかどうか確認してください。存在する場合、名前につづりの間違いがないか調べてください。すべてのパッケージ名は小文字である必要があります。

---

**DBI1061E** ファイル・セットまたはパッケージ名 *pkg-name* がありません。

**説明:** ソフトウェアの前提条件を検証中にエラーが起きました。このコマンドを使用する前に上記のファイル・セットまたはパッケージをインストールする必要があります。

**ユーザーの処置:** 与えられたパッケージの名前がシステムにインストールされているかどうか確認してください。

---

**DBI1062E** ファイル・セットまたはパッケージ名 *pkg-name* がありません。

**説明:** このコマンドを使用する前に上記のファイル・セットまたはパッケージをインストールする必要があります。

**ユーザーの処置:** ファイル・セットまたはパッケージをインストールして、コマンドを再発行してください。

---

**DBI1063E** PTF またはパッチ *patch-name* がありません。

**説明:** このコマンドを使用する前に、上記の PTF またはパッチをインストールする必要があります。このコマンドを正常に実行するには、この PTF またはパッチが必要です。

**ユーザーの処置:** 必要な PTF またはパッチをインストールして、コマンドを再発行してください。

---

**DBI1065E** プログラム *program-name* が異常終了しました。

**説明:** 実行中にエラーが発生し、このプログラムが異常終了しました。

**ユーザーの処置:** 問題を訂正してコマンドを再実行してください。

---

**DBI1066E** プログラム *program-name* は要求によって終了されました。

**説明:** このプログラムはユーザーによって終了されました。

**ユーザーの処置:** 同じコマンドを発行しプログラムを再始動してください。

---

**DBI1067E** ロケール *locale-name* の DB2 製品ライブラリーはインストールされていません。

**説明:** DB2DIR/doc/<locale-name>/html ディレクトリーに、tar 圧縮された HTML ファイルがありません。ここで、

DB2DIR =  
/usr/opt/db2\_08\_xx、ここで xx は 01 または FPn、ここで n は FixPak 番号  
または /opt/IBM/db2/V8.x、ここで x は 1 または FPn、ここで n は FixPak 番号

**ユーザーの処置:** 必要なロケールの DB2 製品ライブラリーをインストールして、このコマンドを再発行してください。

---

**DBI1068E** ディレクトリー *dir-name* に HTML ファイルのすべてを圧縮解除および un-tar するのに十分なスペースがありません。

**説明:** ファイル・システムがいっぱいなため、圧縮解除および un-tar を行った後で、HTML ファイルをすべて上記のディレクトリーに保管できません。

**ユーザーの処置:** ファイル・システムのサイズを増やすか、またはファイル・システムで十分なディスク・スペースを解放してください。コマンドをもう一度発行してください。

---

**DBI1069E** 予期しないエラーです。関数 = *fname*、戻りコード = *return-code*

**説明:** このプログラムの実行中、想定外のエラーが発生しました。

**ユーザーの処置:** 問題が解決しない場合は、IBM サポートに連絡して以下の情報を伝えてください。

- メッセージ番号
- 関数名
- 戻りコード
- 問題記述

---

**DBI1070I** プログラム *program-name* は正常に完了しました。

---

**DBI1072I** db2licm が nodelock ファイルを更新できませんでした。詳細情報については、ログ・ファイル *log-name* を参照してください。

**説明:** db2licm が nodelock ファイルにライセンスを追加できなかったため、DB2 は製品ライセンスがインストールされるまで試供版のライセンスで実行されます。

**ユーザーの処置:** コマンドを再実行し、改善しない場合には、nodelock ファイルにライセンス・キ

一を手動で追加してください。ライセンスの詳細情報については、「概説およびインストール」を参照してください。

---

**DBI1079I** 出力はログ・ファイル *log-name* に保存されています。

**説明:** 処理済みの、または失敗したすべての操作は、このログ・ファイルに保管されています。

**ユーザーの処置:** ログ・ファイルは修正しないでください。このファイルは IBM 技術サポートが参照するためのものです。

---

**DBI1080E** ディスクがいっぱいです。 *dir-name* には *KB KB* 最低限必要ですが、使用可能なスペースは *KB KB* しかありません。

**説明:** ファイル・システムあるいはディレクトリに十分なフリー・スペースがありません。

**ユーザーの処置:** ディスク・スペースを開放してコマンドを再試行してください。

---

**DBI1081E** ファイルまたはディレクトリ *file-name* がありません。

**説明:** このコマンドの処理に必要なファイルまたはディレクトリがありません。

**ユーザーの処置:** ファイルがいずれかのデータベース製品に属するものである場合は、その製品が正しくインストールされているかどうかを調べて、必要であればその製品を再インストールしてください。ファイルがインスタンスに属するものである場合は、そのインスタンスがすでに除去されているか、または壊れている可能性があります。

与えられたディレクトリの名前がシステムに存在しているかどうか確認してください。存在する場合、名前につづりの間違いがないか調べてください。UNIX のどのファイルおよびディレクトリの名前でも、大文字小文字が区別されることに注意してください。

---

**DBI1082E** ファイルまたはディレクトリ *file-name* がすでに存在していません。

**説明:** コマンドが作成すべきファイルまたはディレクトリが、すでに存在しています。

**ユーザーの処置:** 示されたファイルまたはディレクトリを調べてください。前にコマンドが正常に完了した結果としてファイルまたはディレクトリが存在する場合は、処置は必要ありません。そうでない場合は、コマンドを再発行する前に、示されたファイルまたはディレクトリを名前変更または除去する必要があります。

---

**DBI1083E** *file-name* を除去しようとしたが、失敗しました。

**説明:** 与えられたファイルまたはディレクトリの削除中にエラーが発生しました。これはこのファイルまたはディレクトリの所有権が不相当である可能性があります。

**ユーザーの処置:** 示されたファイルまたはディレクトリのファイル許可または所有権を調整して、コマンドを再発行してください。

---

**DBI1084E** *file-name* を作成しようとしたが、失敗しました。

**説明:** 与えられたファイルまたはディレクトリの作成中にエラーが発生しました。

**ユーザーの処置:** ファイル・システムに十分なスペースがあること、また必要なディレクトリに対して書き込み許可があることを確かめてください。推奨ディレクトリ許可は *u=rwx,go=r* です。

---

**DBI1085E** ファイルまたはディレクトリ *file-name* を *file-name* へ移動しようとしたが、失敗しました。

**説明:** 与えられたファイルまたはディレクトリの再配置中にエラーが発生しました。コマンドは

正常に処理できませんでした。

**ユーザーの処置:** 示されたファイルまたはディレクトリが移動できなかった理由を判別して、コマンドを再発行してください。

---

**DBI1086E** ファイルまたはディレクトリ *file-name* を *file-name* へコピーしようとしたが、失敗しました。

**説明:** 与えられたファイルまたはディレクトリのコピー中にエラーが発生しました。コマンドは正常に処理できませんでした。

**ユーザーの処置:** 示されたファイルまたはディレクトリがコピーできなかった理由を判別して、コマンドを再発行してください。

---

**DBI1087E** リンク *filename* を作成しようとしたが、失敗しました。

**説明:** 与えられたリンクを作成できません。親ディレクトリの書き込み許可がオフになっているか、あるいはこのファイルまたはディレクトリと同一の名前がすでに存在していることが考えられます。

**ユーザーの処置:** 親ディレクトリの許可をチェックし、インストール処理を再始動してください。推奨ディレクトリ許可は `u=rwx,go=r` です。

---

**DBI1088E** ディレクトリ *directory* で無効なアクセス許可が検出されました。

**説明:** このエラーが発生する場合、複数の理由がある可能性があります。以下の状態のいずれかが発生していると考えられます。

- 与えられた名前がディレクトリ名でない、あるいは読み取りおよび実行許可がない。
- 与えられたディレクトリは作成できるがアクセスできない。このパスにある親ディレクトリのいずれかの許可が、正しく設定されていないと考えられます。

**ユーザーの処置:** 与えられたディレクトリの許可をチェックし、インストール処理を再始動してください。推奨ディレクトリ許可は `u=rwx,go=r` です。

---

**DBI1089E** *backup-dir* ディレクトリに現行インスタンス関連情報を保管中のエラーです。

**説明:** このエラーが発生する場合、複数の理由がある可能性があります。以下の状態のいずれかが発生していると考えられます。

- 指定されたディレクトリに十分な許可がないか、あるいは書き込み許可がありません。
- ファイル・システムに残っているスペースがありません。

**ユーザーの処置:** 適切な訂正処置を行なって、コマンドを再発行してください。

---

**DBI1090E** *file-name* の *parameter* を更新しようとしたが、失敗しました。

**説明:** 提供されたファイルの更新でエラーが発生しました。

**ユーザーの処置:** “<parameter>” のすべてのオカレンスが “<file-name>” で “<value>” に設定されているか、確認してください。

---

**DBI1091E** *directory* に対する許可を変更しようとしたが、失敗しました。

**説明:** 与えられたファイルまたはディレクトリの所有権の変更中にエラーが発生しました。これは、このファイルまたはディレクトリの所有権が不適当である可能性があります。

**ユーザーの処置:** 与えられたディレクトリの許可および所有権をチェックし、インストール処理を再始動してください。推奨ディレクトリ許可は `u=rwx,go=r` です。

---

**DBI1092E** *directory* に対する所有権を変更しようとしたが、失敗しました。

**説明:** 与えられたファイルまたはディレクトリーの所有権の変更中にエラーが発生しました。これはこのファイルまたはディレクトリーの所有権が不適當である可能性があります。

**ユーザーの処置:** 与えられたディレクトリーの所有権をチェックし、インストール処理を再始動してください。

---

**DBI1093E** *directory* に対するグループ所有権を変更しようとしたが、失敗しました。

**説明:** 与えられたファイルまたはディレクトリーの所有権の変更中にエラーが発生しました。これは、このファイルまたはディレクトリーの所有権が不適當である可能性があります。

**ユーザーの処置:** 与えられたディレクトリーの所有権をチェックし、インストール処理を再始動してください。

---

**DBI1094E** ディレクトリー・アクセス・エラーです。

**説明:** ツールが、ディレクトリー・サーバーでの入出力操作中にアクセス・エラーを見つけました。

**ユーザーの処置:** ディレクトリー・サーバーがオンラインで、LAN でアクセス可能であることを確認してください。

---

**DBI1095W** ファイルまたはディレクトリー *name* が見つかりません。

**ユーザーの処置:** ファイル / ディレクトリーを提供するファイル・セット / パッケージがインストールされていることを確認してください。

---

**DBI1096E** ファイルまたはディレクトリー *file-name* がすでに存在していません。

**説明:** コマンドが作成すべきファイルまたはディレクトリーが、すでに存在しています。

**ユーザーの処置:** 示されたファイルまたはディレクトリーを調べてください。このファイルまたはディレクトリーが必要でない場合、除去してコマンドを再実行してください。そうでない場合は、コマンドを再発行する前に、示されたファイルまたはディレクトリーを名前変更または除去する必要があります。

---

**DBI1097E** ファイルまたはディレクトリー *file-name* が存在しません。

**説明:** コマンドが必要なファイルまたはディレクトリーが存在しません。

**ユーザーの処置:** ファイルまたはディレクトリーが存在しない場合、ファイルまたはディレクトリーを提供するソフトウェアをインストールする必要がある場合があります。ファイルまたはディレクトリーが存在する場合、PATH 変数が正しく設定されているかどうか、チェックしてください。ファイルまたはディレクトリーが PATH に設定されたら、コマンドを再実行してください。

---

**DBI1100E** インスタンス・ロックが検出されました。別のインスタンス管理コマンドが実行中です。

**説明:** 別のインスタンス管理コマンド (たとえば、db2icrt、db2idrop、db2iupdt、または db2imigr) が実行中であるため、このコマンドが失敗しました。インスタンス管理コマンドは、排他ロックを獲得しているため、同一インスタンスに対して同時にコマンドを実行することができません。

**ユーザーの処置:** コマンドを再発行する前に、インスタンス・コマンドの他のすべてのオカレンスの実行が完了するのを待ってください。他のインスタンス・コマンドが実行されていない場合、以

下のいずれかのディレクトリーで、ファイル  
“instance.lock” を削除してください。

/usr/opt/db2\_08\_xx、ここで xx は 01 または FPn、  
ここで n は FixPak 番号  
または /opt/IBM/db2/V8.x、ここで x は 1 または  
FPn、ここで n は FixPak 番号

---

#### **DBI1101E** インスタンス名が無効です。

**説明:** インスタンス名は、既存ユーザーのログイン名でなければならず、以下の名前は使用できません。

1. 8 文字より長い名前。
2. “sql”、“ibm” または “sys” で始まる名前。
3. 数字で始まる名前、または a ~ z、\$, #、@、\_、0 ~ 9 以外の文字を使用した名前。

**ユーザーの処置:** 有効なインスタンス名を指定して、コマンドを再発行してください。

---

#### **DBI1103E** Administration Server はすでに存在しています。

**説明:** Administration Server がすでにご使用のシステムに作成されていることを検出しました。1 つのシステムに Administration Server は 1 つしか作成できません。

**ユーザーの処置:** Administration Server を再作成する場合、先に Administration Server をドロップしないと再作成できません。

---

#### **DBI1104E** Administration Server はドロップできません。

**説明:** Administration Server をドロップしようとして、失敗しました。システムは、Administration Server が存在しないことを検出しました。

**ユーザーの処置:** アクションは必要ありません。

---

#### **DBI1105E** DB2 Data Links Manager 管理者はすでに存在します。

**説明:** システムが、DB2 Data Links Manager 管理者がシステムにすでに作成されていることを検出しました。DB2 Data Links Manager 管理者は、システムごとに 1 つしか作成できません。

**ユーザーの処置:** DB2 Data Links Manager 管理者を再作成したい場合、その前に DB2 Data Links Manager 管理者をドロップする必要があります。

---

#### **DBI1106E** DB2 Data Links Manager 管理者をドロップできません。

**説明:** DB2 Data Links Manager 管理者をドロップしようとしたのですが、失敗しました。システムが、DB2 Data Links Manager 管理者が存在しないか、または別のユーザー ID で作成されていることを検出しました。

**ユーザーの処置:** “dlfmllist” コマンドで報告されている DB2 Data Links Manager 管理者の名前が正しいことを確認して、やり直してください。

---

#### **DBI1107E** DB2 Data Links Manager 管理者を作成できません。

**説明:** DCE によって DB2 Data Links Manager 管理者を作成するには、システム上の DFS がバージョン 3.1 またはそれ以上でなければなりません。

---

#### **DBI1108E** 有効なグループ名または数値が指定されていません。

**説明:** dlfm インスタンスを移行するときに、移行するインスタンスが DB2 v7 以前の場合は、-g DLFMGid を指定する必要があります。

DLFMGid には、数値のグループ ID かグループ名を指定することができます。このグループは、DLFMAdmin の 2 次グループでなければなりません。



**ユーザーの処置:** V8 以降の dlfm インスタンスを移行するときに、そのインスタンスにすでにセットアップされている DLFGid と同じでない -g DLFGid が指定された場合は、指定されたパラメーターは無視されて、古い Gid が使用されます。

---

**DBI1109E** カーネル・パラメーターはこのコマンドを使用する前に更新される必要があります。

**説明:** 正しく構成されていないカーネル・パラメーターがあります。

**ユーザーの処置:**

- 「概説およびインストール」マニュアルの記述どおりに、必要なすべてのカーネル・パラメーターを更新します。
- システムをリブートします。
- このコマンドを再度試行します。

---

**DBI1111E** FencedID パラメーターはこのコマンドでは **-u flag** を使用して指定する必要があります。

**説明:** FencedID パラメーターはこのコマンドを使用して指定する必要があります。FencedID パラメーターは分離したユーザー定義関数 (UDF) および分離したストアード・プロシージャが動作するユーザーの名前に設定されます。

**ユーザーの処置:** 引き数 “-u FencedID” を追加してコマンドを再入力してください。ここで FencedID は 分離した UDF あるいは分離したストアード・プロシージャが動作するユーザーの名前です。

セキュリティ上の理由から、FencedID としてインスタンス名を使用しないことをお勧めします。ただし、ユーザー定義関数 (UDF) あるいはストアード・プロシージャを使用する予定のない場合に、FencedID をインスタンス名に設定すると、FencedID に別のユーザーを作成する手間が省けます。

---

**DBI1112E** FencedID パラメーター *fenced-id* は無効です。

**説明:** 分離したユーザー定義関数およびストアード・プロシージャが実行されるユーザー名を指定する FencedID パラメーターは以下のいずれかにしてください。

- 既存ユーザーのログイン名。
- root または bin ユーザーの設定は不可。

**ユーザーの処置:** 有効な FencedID パラメーターでこのコマンドを再度試行してください。

---

**DBI1113W** FencedID パラメーターの前の値 *old-value* が異なっています。新規の値 *new-value* が無視されます。

**説明:** コマンド行で入力された FencedID パラメーターはこのユーザー ID の前に使用されたパラメーターと異なります。このパラメーターの新規の値は無視されます。

---

**DBI1115E** 無効な AuthType パラメーター *auth-type* が **-a** フラグを付けて指定されました。

**説明:** インスタンスに使用される認証タイプを指定する AuthType パラメーターが無効です。有効な認証タイプは、以下のとおりです。

- SERVER
- CLIENT
- SERVER\_ENCRYPT

**ユーザーの処置:** 有効な AuthType パラメーターを指定して、コマンドを再発行してください。

---

**DBI1120E** インスタンス *inst-name* がドロップできません。

**説明:** まだアクティブである可能性のあるインスタンスをドロップしようとしてしました。

---

**DBI1121E** “db2idrop” を使用して Administration Server をドロップできません。

**説明:** “db2idrop” を使用して Administration Server のドロップをしようとした。この操作はレジストリーの不整合状態を引き起こすため許可されません。

**ユーザーの処置:** “dasidrop” コマンドを使用して Administration Server を除去してください。

---

**DBI1122E** インスタンス *inst-name* が更新できません。

**説明:** インスタンスを更新しようとした。このインスタンスは以下の理由で更新できません。

- この “db2iupdt” コマンドはこのインスタンスを更新するためには使用できない。
- インスタンスがまだアクティブである。

**ユーザーの処置:** 適切なバージョンの “db2iupdt” コマンドを使用していることを確認してください。また、インスタンスで実行中の db2 処理がないことも確認してください。コマンドを再発行してください。

---

**DBI1123E** “db2iupdt” を使用して Administration Server を更新できません。

**説明:** “db2iupdt” を使用して Administration Server の更新をしようとした。この操作はレジストリーの不整合状態を引き起こすため許可されません。

**ユーザーの処置:** “dasiupdt” コマンドを使用して Administration Server を更新してください。

---

**DBI1124E** インスタンス *inst-name* が移行できません。

**説明:** インスタンスを移行しようとした。このインスタンスは以下の理由で移行できません。

- インスタンスがまだアクティブである。

- このインスタンスの移行がサポートされていない。
- このバージョンの “db2imigr” がこのインスタンスを移行するためには使用できない。

**ユーザーの処置:** インスタンスが、移行に使用できて、適切なバージョンの “db2imigr” コマンドを使用しているかを確認してください。インスタンスの移行の詳細については、ご使用のプラットフォームの「概説およびインストール」マニュアルを参照してください。

---

**DBI1125E** “db2idrop” では DB2 Data Links Manager 管理者をドロップできません。

**説明:** “db2idrop” を使用して DB2 Data Links Manager 管理者をドロップしようとした。この操作はレジストリーの不整合状態を引き起こすため許可されません。

**ユーザーの処置:** “dlfmdrop” コマンドを使用して、DB2 Data Links Manager 管理者を除去してください。

---

**DBI1126W** 1 つ以上のビューがデータベース移行に影響する可能性があります。

**説明:** DB2 が、変更されたデータベース・エンティティーに從属するビューを見つけました。これは、データベース移行の失敗を引き起こす原因となります。ユーザーの応答については Release.Notes を参照してください。

---

**DBI1128E** “dlfmdrop” を使用して Administration Server をドロップできません。

**説明:** “dlfmdrop” を使用して Administration Server のドロップをしようとした。この操作はレジストリーの不整合状態を引き起こすため許可されません。

**ユーザーの処置:** “dasidrop” コマンドを使用して Administration Server を除去してください。



---

**DBI1129E “dlfmupdt” を使用して Administration Server を更新できません。**

**説明:** “dlfmupdt” を使用して Administration Server の更新をしようとした。この操作はレジストリーの不整合状態を引き起こすため許可されません。

**ユーザーの処置:** “dasiupdt” コマンドを使用して Administration Server を更新してください。

---

**DBI1131E ユーザー ID *user-id* が無効です。**

**説明:** 与えられたユーザー ID にアクセスしようとして失敗しました。以下の状態のいずれかが発生しました。

- このユーザー ID がシステムにない。
- このユーザーのホーム・ディレクトリーが正しく設定されていない。
- DB2 で必要なユーザー属性のいずれかが設定されていない。
- このユーザーの UID が 0 である。

**ユーザーの処置:** 有効なホーム・ディレクトリーの有効なユーザー ID、シェル、1 次グループ、および 2 次グループが使用されているか、確認してください。必要なら、ユーザーを新規作成してください。

---

**DBI1135E ユーザー ID *user-id* の 1 次グループ *group-name* が無効です。**

**説明:** インスタンス・ユーザー ID の基本グループ名には、以下を使用することはできません。

1. “guest”、“admin”、“user” または “local” のいずれかの名前。
2. “sql” または “ibm” で始まる名前。
3. 数値で始まる名前、あるいは a ~ z、\$、#、@、または 0 ~ 9 以外の文字を使った名前。

**ユーザーの処置:** インスタンス・ユーザー ID の 1 次グループを有効な名前を持つグループに変更

して、コマンドを再発行してください。

---

**DBI1136W OLAP Start Kit がインストールされていません。**

**説明:** 移行しているインスタンスは OLAP 関数を持っています。OLAP Starter Kit は DB2 V8 にはありません。このインスタンスは、-F パラメーターが指定された場合のみ移行されます。

**ユーザーの処置:** プログラムが打ち切られた場合は、移行を強制するための -F パラメーターを指定して、同じ db2imigr コマンドを実行してください。OLAP 関数はなくなります。

---

**DBI1137W サーバー製品がインストールされていません。**

**説明:** 移行しているインスタンスはサーバー・インスタンスです。DB2 V8 サーバー製品が検出されていません。このインスタンスは、-F パラメーターが指定された場合のみ移行されます。

**ユーザーの処置:** プログラムが打ち切られた場合は、移行を強制するための -F パラメーターを指定して、同じ db2imigr コマンドを実行してください。DB2 サーバー関数はなくなります。DB2 サーバー製品をインストールし、移行しているインスタンスに対して手動で db2iupdt を実行し、サーバー関数を再度使用可能にする必要があります。

---

**DBI1138W Query Patroller がインストールされていません。**

**説明:** 移行しているインスタンスは Query Patroller 関数を持っています。インストールされている DB2 V8 で、Query Patroller が検出されません。このインスタンスは、-F パラメーターが指定された場合のみ移行されます。

**ユーザーの処置:** プログラムが打ち切られた場合は、移行を強制するための -F パラメーターを指定して、同じ db2imigr コマンドを実行してください。Query Patroller は存在しません。Query

Patroller for V8 をインストールし、移行しているインスタンスに対して手動で dqpcrt を実行して、Query Patroller の関数を再度使用可能にする必要があります。

---

**DBI1139E** データ・リンク・ファイル・マネージャーがインストールされていません。

**説明:** 移行しているインスタンスにはデータ・リンク関数があります。インストールされている DB2 V8 で、データ・リンク・ファイル・マネージャーが検出されません。インスタンスは移行されません。

**ユーザーの処置:** DB2 V8 用のデータ・リンク・ファイル・マネージャーをインストールして、再度コマンドを実行してください。

---

**DBI1140W** コマンドが強制されています。上記の警告で示された障害のポイントを超過して、発行されたコマンドを強制するための **-F** フラグが指定されています。

---

**DBI1141E** **Visual Warehouse 3.1** のインスタンスが検出されました。**Visual Warehouse 3.1** のインスタンスを **V8** に移行することはサポートされていません。**Visual Warehouse 5.2** またはそれ以上だけを **V8** に移行できます。

---

**DBI1142W** **Visual Warehouse 5.2** のインスタンスが検出されました。インスタンスは移行されますが、ウェアハウスのメタデータは移行されません。

---

**DBI1143W** **Relational Connect** がインストールされていません。

**説明:** 移行先のバージョンで **Relational Connect** が検出されませんでした。DB2 および非 DB2 データ・ソース、またはそのいずれかのデータ・ソースの設定ファイルが、移行しているインスタンスで検出されました。この構成は移行されますが、これらの構成パラメーターを使用するには、**Relational Connect for DB2 V8** をインストールする必要があります。

**ユーザーの処置:** 構成パラメーターを使用するために、**Relational Connect for DB2 V8** をインストールしてください。

---

**DBI1144E** **Relational Connect** がインストールされていません。

**説明:** 移行先のバージョンで **Relational Connect** が検出されませんでした。非 DB2 データ・ソースを持っていることを示す環境変数が検出されました。**Relational Connect** がインストールされていないと、このインスタンスは移行できません。

**ユーザーの処置:** **Relational Connect** をインストールして、コマンドをやり直してください。

---

**DBI1145W** 同期点マネージャー (SPM) ログ・ディレクトリーがすでにあります。

**説明:** 同期点マネージャー (SPM) ログ・ディレクトリーがすでにあります。SPM の新規バージョンはログの古いバージョンを処理できません。この SPM の新規バージョンを使用する前に、未確定のトランザクションがないかどうか、確認する必要があります。

**ユーザーの処置:**

- 未確定のトランザクションに関心がない場合、db2iupdt コマンドに **-s** フラグを付けて実行し、インスタンスを更新して処理を続行しま

す。この結果、`sqllib/spmlog50` の下に、`sqllib/spmlog` ディレクトリーのコピーを作成します。

- 未確定のトランザクションの存在に関心がない場合には、DB2 製品を前のバージョンに戻す必要があり、また未確定のトランザクションに関する製品の `readme` にある指示に従う必要があります。

---

**DBI1170E** **-w** フラグは、可能な入力として **64** または **32** だけを受け入れません。

**説明:** `db2icrt` または `db2iupdt` コマンドを使用している場合、オプション・フラグ `-w` に指定できるのは値 `32` または `64` だけです。64 ビット・インスタンスを作成するには、`db2icrt` に `-w 64` と指定します。また 32 ビット・インスタンスを 64 ビットに更新するには、`db2iupdt` に `-w 64` と指定します。このような場合でなければ、`-w` フラグは必要ありません。

**ユーザーの処置:** 正しい引き数でコマンドを出し直してください。

---

**DBI1171E** **DB2 64 ビット・サポートはインストールされていません。**

**説明:** 64 はこのプラットフォームでサポートされていないか、暗黙指定になっています。

**ユーザーの処置:**

- Linux IA64 の場合、64 ビット・インスタンスを作成するために、`-w` オプションを指定せずにコマンドを再発行してください。
- その他のプラットフォームでは、32 ビット・インスタンスを作成するために、`-w 64` オプションを指定せずにコマンドを再発行してください。

---

**DBI1172W** **64 ビットのインスタンスを作成または更新しようとしている時に、`bos.rte.libc` と `bos.adt.prof` のいずれかまたは両方のインストール済みレベルが、64 ビットのインスタンスに必要な最低要件より低いことがわかりました。**

**説明:** `bos.rte.libc` と `bos.adt.prof` のいずれかまたは両方のインストール済みレベルが 4.3.3.50 より低くなっています。このレベルは 64 ビットのインスタンスをサポートするために必要な最低限の要件です。

**ユーザーの処置:** 処理を進める前に、以下の PTF のいずれかまたは両方をダウンロードしてシステムを更新してください。

- `bos.rte.libc` の場合、PTF は U473728.bff です。
- `bos.adt.prof` の場合、PTF は U473581.bff です。

PTF のダウンロードに関する情報は <http://www.ibm.com/aix> から使用できます。

警告ポイントを超えて実行を強制するために、`-F` パラメーターを指定してコマンドを再発行することができます。

---

**DBI1173W** **`db2iupdt` がこのインスタンスを 64 ビット非サーバー・インスタンスに更新しました。**

**説明:** Enterprise Server Edition がインストール済みの場合、32 ビット Workgroup サーバー・インスタンスを 64 ビットのサーバー・インスタンスにのみ更新することができます。

**ユーザーの処置:** サーバー・インスタンスに更新したくない場合は、更新されたインスタンスを継続して使用することができます。このインスタンスを 32 ビット Workgroup サーバー・インスタンスに戻したい場合は、以下のコマンドを実行してください。

```
db2iupdt -w 32 <instance-name>
```

---

**DBI1174E** db2icrt/db2imgr が 64 ビット非サーバー・インスタンスを作成しました。

**説明:** 64 ビット Workgroup サーバー・インスタンスを作成またはこれに移行することはできません。64 ビットのサーバー・インスタンスを実行するには、Enterprise Server Edition がインストール済みである必要があります。

**ユーザーの処置:** サーバー・インスタンスを必要としない場合は、作成された、あるいは移行されたインスタンスを継続して使用することができます。64 ビットのサーバー・インスタンスを使用するには、DB2 V8 Enterprise Server Edition をインストールし、以下のコマンドを実行してください。

```
db2iupdt -w 64 <instance-name>
```

---

**DBI1175W** 64 ビット・インスタンスの作成または更新を試行中に、Solaris オペレーティング環境のインストール・レベルが 64 ビットの最小要件よりも低いことが検出されました。

**説明:** Solaris オペレーティング環境のインストール・レベルは、64 ビットのインスタンスのサポートに必要な最小レベルの 5.7 より低くなっています。

**ユーザーの処置:** 先に進む前に、ご使用のシステムを Solaris オペレーティング環境のを最小必須レベルに更新してください。

警告ポイントを超えて実行を強制するために、-F パラメーターを指定してコマンドを再発行することができます。

---

**DBI1176I** 使用法 :

```
installAltFixPak [-h] [-s]
```

**説明:** 誤った引き数が指定されました。このコマンドの有効な引き数は以下のとおりです。

**-h** 使用法情報を表示する

**1712** リリース情報

**-s** /usr/opt/db2\_08\_01 または /opt/IBM/db2/V8.1 にインストールされているのとまったく同じファイル・セット / パッケージを (サポートされている製品の) 代替パスにインストールする。

installAltFixPak が、/usr/opt/db2\_08\_01 または /opt/IBM/db2/V8.1 に DB2 がインストールされていないことを検出すると、-s オプションは使用されません。この場合は、db2\_install ユーティリティーを呼び出します。

**ユーザーの処置:** 以下のようにコマンドを入力し直してください。

```
installAltFixPak [-h] [-s]
```

---

**DBI1177W** インスタンスが現在使用しているレベルより低いコード・レベルにインスタンスを更新します。

**説明:** 現在インスタンスが使用しているレベルより低いコード・レベルに、インスタンスを更新しようとした。

**ユーザーの処置:** 現在のレベルよりも低いコード・レベルにインスタンスを更新するには、以下のように -D オプションを指定して db2iupdt コマンドを発行してください。

```
db2iupdt -D <instance-name>
```

---

**DBI1181E** DB2 64 ビット・サポートはインストールされていません。

**説明:** 64 ビット・インスタンスはこのプラットフォームではサポートされていません。

**ユーザーの処置:** 先に進む前に、以下のいずれかを行ってください。

- ご使用のシステムを 64 ビット DB2 を実行する最小必須レベルに更新し、コマンドを再発行します。

- 移行中または更新中のインスタンスをドロップし、32 ビット・インスタンスとして再作成します。

---

### **DBI1182W Warehouse Manager Connectors がインストールされていません。**

**説明:** 移行しているインスタンスは Warehouse Manager Connectors 関数を持っています。インストールされている DB2 V8 で、Warehouse Manager Connectors が検出されませんでした。このインスタンスは、-F パラメーターが指定された場合にのみ移行されます。

**ユーザーの処置:** 移行を強制するための -F パラメーターを指定して、db2imigr コマンドを再実行してください。Warehouse Manager Connectors はなくなります。Warehouse Manager Connectors 関数を再び使用可能にするには、Warehouse Manager Connectors for V8 をインストールし、移行しているインスタンスで db2iupdt を手動で実行する必要があります。

---

### **DBI1183W Spatial Extender がインストールされていません。**

**説明:** 移行しているインスタンスは Spatial Extender 関数を持っています。インストールされている DB2 V8 で、Spatial Extender が検出されませんでした。このインスタンスは、-F パラメーターが指定された場合にのみ移行されます。

**ユーザーの処置:** 移行を強制するための -F パラメーターを指定して、db2imigr コマンドを再実行してください。Spatial Extender はなくなります。Spatial Extender 関数を再び使用可能にするには、Spatial Extender for V8 をインストールし、移行しているインスタンスで db2iupdt を手動で実行する必要があります。

---

### **DBI1184W Life Sciences Data Connect がインストールされていません。**

**説明:** 移行しているインスタンスは Life Sciences Data Connect 関数を持っています。イン

ストールされている DB2 V8 で、Life Sciences Data Connect が検出されませんでした。このインスタンスは、-F パラメーターが指定された場合にのみ移行されます。

**ユーザーの処置:** 移行を強制するための -F パラメーターを指定して、db2imigr コマンドを再実行してください。Life Sciences Data Connect はなくなります。Life Sciences Data Connect 関数を再び使用可能にするには、Life Sciences Data Connect for V8 をインストールし、移行しているインスタンスで db2iupdt を手動で実行する必要があります。

---

**DBI1185I** サーバー・プロトコル *protocol* は、すでにサポートされていません。DB2COMM から除去されました。

---

**DBI1186I** 使用法 :

**db2cdbcr [-d] -n CDBName**

**説明:** 間違った引き数が db2cdbcr コマンドに入力されました。このコマンドの有効な引き数は以下のとおりです。

**-hl-?** 使用法情報を表示する

**-d** デバッグ・モードをオンにする

**-n CDBName**

CDBName は、作成するウェアハウス・コントロール・データベースの名前です。

**ユーザーの処置:** 以下のようにコマンドを入力し直してください。

db2cdbcr -n CDBName

---

**DBI1187E** DB2 ウェアハウス・コントロール・データベースのセットアップを完了できません。

**説明:** DB2 ウェアハウス・コントロール・データベースを作成しようとしたのですが、失敗しました。

**ユーザーの処置:** ノード・グループ (ノード 0 に作成) と表スペース (どちらも FLG32K と呼ばれる) も同様に作成する必要があります。詳細については DB2 データウェアハウスのドキュメンテーションを参照してください。

---

**DBI1190I** db2setup が DB2 セットアップ・ウィザードを準備中です。このウィザードがプログラムのセットアップ操作を案内します。お待ちください。

---

**DBI1191I** 提供された応答ファイルにしたがって、db2setup が DB2 をインストールおよび構成しています。お待ちください。

---

**DBI1200E** ディレクトリー *directory-name* が、バージョン *version-number* インスタンスではありません。

**説明:** 示されたディレクトリーが、データベース製品の必要なバージョンを指していないために、コマンドが処理できません。

**ユーザーの処置:** 示されたインスタンスでは、コマンドを実行できません。

---

**DBI1202E** インスタンス *inst-name* の移行がサポートされていません。

**説明:** インスタンスは以下の理由で、移行できません。

1. このバージョンのインスタンスからの移行がサポートされていない。

2. インスタンスがすでに製品の現行バージョンを使用していて移行が必要ない。

**ユーザーの処置:** このインスタンスが移行に対して有効であるか確認して有効なインスタンス名を指定して、コマンドを再試行します。

---

**DBI1205E** 新しいバージョンに移行できない 1 つまたは複数のローカル・データベースが検出されました。エラー・リストのログファイル *logfile-name* を調べてください。

**説明:** 考えられるデータベース上のエラーは以下の通りです。

- バックアップ保留中
- ロールフォワード保留中
- データベースの不整合
- 1 つまたは複数の表スペースは正常な状態でない
- SYSCAT、SYSSTAT、または SYSFUN というスキーマ名を持つデータベース・オブジェクトがデータベースに入っている
- SYSFUN.DIFFERENCES 関数に依存しているデータベース・オブジェクトの入ったデータベース。違反していると考えられるオブジェクトは以下のとおりです。
  - 制約
  - 関数
  - トリガー
  - ビュー

**ユーザーの処置:** ファイル “<logfile-name>” の各データベースに対して、以下の訂正処置を行ってください。

- バックアップ保留中 - データベースをバックアップしてください。
- ロールフォワード保留中 - ログの終わりまでデータベースをロールフォワードし、停止してください。



- データベースの不整合 - 一致した状態に戻すようにデータベースを再始動してください。
- 表スペースが正常な状態でない - データベースをロールフォワードしてください。
- 無効なスキーマ - オブジェクトをドロップし、正しいスキーマ名 (修飾子) でオブジェクトを再作成してください。オブジェクトが表の場合、最初にそのデータを抽出し、表をドロップし、正しいスキーマ名で表を再作成してから、新しい表にデータをインポートまたはロードしてください。
- SYSFUN.DIFFERENCES に依存しているオブジェクト -
  - 制約 - 制約をドロップするように表を変更してください。
  - 関数 - 関数をドロップしてください。
  - トリガー - トリガーをドロップしてください。
  - ビュー - ビューをドロップしてください。

---

**DBI1211E** *directory-name* の移行が失敗しました。

**説明:** ディスク・スペースまたはメモリーの不足などのシステム・エラーが、ディレクトリーの移行中に見つかりました。インスタンスの移行は失敗しました。

**ユーザーの処置:** コマンドを再発行する前に、十分なディスク・スペースとメモリーがあることを確認してください。

---

**DBI1212W** *path* にあるローカル・データベースの移行が失敗しました。

**説明:** 想定外のエラーが発生したため、ローカル・データベース・ディレクトリーの移行処理は完了しません。

**ユーザーの処置:** エラー・ファイル “db2mgdbd.err” およびローカル・データベース・ディレクトリーのバックアップ

“sqldbdir/sqllddir.bak” は上記のディレクトリー・パスに作成されます。

使用しているディレクトリーの修理をするには、2 つのファイルを保持し、可能な方法についての指示を IBM サービスにおたずねください。ローカル・データベースがない場合、ローカル・データベースにカタログされているデータベースはこれ以上のアクセスはできません。

---

**DBI1220W** *path* 上のノード・ディレクトリーの移行は失敗しました。

**説明:** 想定外のエラーが発生したため、ノード・ディレクトリーの移行処理は完了しません。

**ユーザーの処置:** リカバリーのオプションには以下の 2 通りがあります。

1. ノード・ディレクトリーを除去し、ノード項目を再カタログしてください。
2. エラー・ファイル “db2mgndd.err” およびノード・ディレクトリーのバックアップ “sqlnodir/sqlnndir.bak” は上記で言及しているディレクトリー・パスに作成されます。  
使用しているディレクトリーの修理をするには、2 つのファイルを保持し、可能な方法についての指示を IBM サービスにおたずねください。ノード・ディレクトリーをリカバリーするには、IBM サービスに連絡してください。

---

**DBI1221W** ノード・ディレクトリーが壊れているので、移行できません。

**説明:** ノード・ディレクトリー・ファイルが同じでないか、またはノード・ディレクトリー・ファイルが壊れています。

**ユーザーの処置:** ノード・ディレクトリーを除去し、ノード項目を再カタログしてください。

---

**DBI1222W システム・データベース・ディレクトリーが壊れているので、移行できません。**

**説明:** 1 次およびバックアップ・システム・データベース・ディレクトリーが同一でないか、あるいはシステム・データベース・ディレクトリーが壊れています。

**ユーザーの処置:** システム・データベース・ディレクトリーを除去し、すべてのデータベース項目を再カタログしてください。

---

**DBI1223W ローカル・データベース・ディレクトリーが壊れているので、移行できません。**

**説明:** 1 次およびバックアップ・データベース・ディレクトリーが同一でないか、あるいはローカル・データベース・ディレクトリーが壊れています。

**ユーザーの処置:** エラー・ファイル “db2mgdbd.err” およびローカル・データベース・ディレクトリーのバックアップ “sqldbdir/sqldddir.bak” は上記のディレクトリー・パスに作成されます。

使用しているディレクトリーの修理をするには、2 つのファイルを保持し、可能な方法についての指示を IBM サービスにおたずねください。ローカル・データベースがない場合、ローカル・データベースにカタログされているデータベースはこれ以上のアクセスはできません。

---

**DBI1225W カタログ済みローカル・データベースの認証タイプが変更されました。**

**説明:** インスタンスに指定されている認証タイプと異なる認証タイプを持つ、1 つ以上のカタログ済みデータベース項目が見つかりました。何も処置を行なわないと、すべてのカタログ済みローカル・データベース項目が、それらが属しているインスタンスの認証タイプを想定します。

**ユーザーの処置:** インスタンスに指定されている

認証と同じでないカタログ済みローカル・データベース項目のリストについては、インスタンスのホーム・ディレクトリーのファイル migration.log をチェックしてください。データベースに前の認証タイプを維持させたい場合は、インスタンスの認証タイプを変更するか、またはデータベースに必要な認証タイプを持つ別のインスタンスに移動することができます。ただし、インスタンスの認証タイプを変更する前に、すべてのカタログ済みローカル・データベースに、新しい認証タイプを持たせることを確認する必要があります。

---

**DBI1240E データベース・マネージャーが始動できませんでした。**

**説明:** カタログ済みローカル・データベースが移行可能かどうかをチェックしようとしたときに、データベース・マネージャーが始動できませんでした。

**ユーザーの処置:** データベース・マネージャーが始動できなかった理由を解決して、コマンドを再発行してください。

---

**DBI1250E アプリケーションは、インスタンス *instance-name* をまだ使用中です。**

**説明:** 指定されたインスタンスを使用している実行中のアプリケーションが、まだ存在します。コマンドを正常に完了させるには、このインスタンスを使用しているすべてのアプリケーションを終了する必要があります。以下のコマンドを実行すると、現在インスタンスを使用中のアプリケーションのリストを取得できます。

```
db2 list applications
```

**ユーザーの処置:** アプリケーションが終了するのを待つか、またはアプリケーションを明示的に強制終了させることができます。インスタンス所有者としてログオンし、以下のコマンドを実行することもできます。

```
db2 force application all
```



上記のコマンドを使用して終了したときに、想定外の動作をするアプリケーションがある可能性があることに注意してください。すべてのアプリケーションが停止したあとで、“db2istop” コマンドを使用してデータベース・マネージャーをしてください。

---

**DBI1260E** 応答ファイルから読み取ることができません。

**説明:** 指定された応答ファイルからの読み取りに失敗しました。応答ファイルに読み取りアクセスがあり、応答ファイルへの指定パスが正しいことを確認してください。

**ユーザーの処置:** アクセス許可および応答ファイルの位置を訂正して、再試行してください。

---

**DBI1261E** 値は、キーワードでは無効です。

**説明:** 応答ファイルで指定された値は、応答キーワードとして無効です。有効な入力については、サンプル応答ファイルを参照してください。

**ユーザーの処置:** 応答ファイルの次の行にある問題を訂正して、再試行してください。

---

**DBI1262E** 応答ファイルに不明のキーワードがあります。

**説明:** 応答ファイルで指定されたキーワードは無効です。有効なキーワードについては、サンプル応答ファイルを参照してください。

**ユーザーの処置:** 応答ファイルの次の行にある問題を訂正して、再試行してください。

---

**DBI1264E** *program-name* 実行中にエラーが発生しました。詳細情報については、インストール・ログ・ファイル *log-name* を参照してください。

**説明:** 処理済みの、または失敗したすべての操作は、このログ・ファイルに保管されています。

**ユーザーの処置:** ログ・ファイルは修正しないで

ください。このファイルは IBM 技術サポートが参照するためのものです。

---

**DBI1266I** 詳細情報については、ログ・ファイル *log-name* を参照してください。

**説明:** 処理済みの、または失敗したすべての操作は、このログ・ファイルに保管されています。

**ユーザーの処置:** ログ・ファイルは修正しないでください。このファイルは IBM 技術サポートが参照するためのものです。

---

**DBI1268E** ファイル・セット *fileset* はインストール・メディアでは使用できません。

**説明:** インストールされるファイル・セットのいずれかが、インストール・メディアで見つかりません。ファイル・セットがインストール・メディアに見つからない場合には、インストールができません。

**ユーザーの処置:** ファイル・セットがインストール・メディアにあるかどうかを確認してください。ファイル・セットがインストール・メディアに見つからない場合には、このファイル・セットの選択を取り消し、再試行してください。

---

**DBI1270E** 応答ファイルにエラーがあります。

**説明:** 応答ファイルの処理中にエラーを検出しました。問題が訂正されない限り、インストールは続行できません。

**ユーザーの処置:** 応答ファイルの次の行にある問題を訂正して、再試行してください。

---

**DBI1278W** 応答ファイルから警告が出されました。

**説明:** 応答ファイルの処理中に警告を検出しました。DB2 インストーラーは停止せずに続行します。

**ユーザーの処置:** 必要であれば、応答ファイルの

次の行にある問題を訂正して、再試行してください。

---

**DBI1279I** 応答ファイルから注意が出されました。

**説明:** 応答ファイルの処理中に注意を検出しました。DB2 インストーラーは停止せずに続行します。

**ユーザーの処置:** これは通知でしかありません。アクションは必要ありません。

---

**DBI1281E** データベース・マネージャー構成ファイルが初期化できませんでした。

**説明:** データベース・マネージャー構成ファイルの初期化中にエラーが起きました。DB2 インスタンスを作成または移行できませんでした。

**ユーザーの処置:** ログ・ファイルを参照して問題を解明してください。問題を解決してコマンドを再試行してください。問題が解決しない場合は、IBM サポートに連絡してください。

---

**DBI1282W** データベース・マネージャー構成ファイルがマージできませんでした。

**説明:** 2 つのデータベース・マネージャー構成ファイルのマージ中に、エラーが見つかりました。前のデータベース・マネージャー構成ファイルが、新しいデータベース・マネージャー構成ファイルとマージできませんでした。

**ユーザーの処置:** 新しいデータベース・マネージャー構成ファイル内の値を調べて、必要ならパラメーターを更新してください。

---

**DBI1283E** *instance* に対する通信情報が更新できません。

**説明:** SVCENAME パラメーターまたは DB2COMM レジストリー値、あるいはその両方が、インスタンス作成中に更新できませんでした。

**ユーザーの処置:** SVCENAME パラメーターがデータベース・マネージャー構成ファイルで “<profile-name>” に設定されていて、以下の例を使用して更新されることを確認してください。

```
db2 update dbm cfg using
 SVCENAME “<profile-name>”
```

また、DB2COMM 値は、以下のコマンドを実行して tcpip に更新されることを確認してください。

```
db2set DB2COMM=tcpip
```

---

**DBI1290E** プロファイル・レジストリー *profile-name* を作成しようとしたが、失敗しました。

**説明:** プロファイル・レジストリーの作成中にエラーが発生しました。以下の状態のいずれかが発生していると考えられます。

- プロファイル・レジストリーを備えたファイルに誤ったアクセス許可がある。
- プロファイル・レジストリーが正しく設定されていない。
- ファイル・システムに十分な空きスペースがない。

**ユーザーの処置:** 以下のディレクトリーのいずれかにあるインスタンス・リスト・プロファイルのファイル許可をチェックしてください。

/usr/opt/db2\_08\_xx、ここで xx は 01 または FPn、ここで n は FixPak 番号  
または /opt/IBM/db2/V8.x、ここで x は 1 または FPn、ここで n は FixPak 番号

---

**DBI1291E** インスタンス *instance-name* がインスタンス・リストに見つかりませんでした。

**説明:** 示されたインスタンスが、インスタンス・リストに見つかりませんでした。

**ユーザーの処置:** “db2ilist” コマンドによって報

告されたインスタンスのリストが正しいことを確認してください。有効なインスタンス名を指定して、コマンドを再実行してください。

---

**DBI1292E** インスタンス *instance-name* はすでにインスタンス・リストにあります。

**説明:** 作成または移行中のインスタンスが、すでにインスタンス・リストにあります。

**ユーザーの処置:** “db2iset -I” コマンドで報告されているインスタンス・リストの一部でない別のインスタンス名を使用してください。

---

**DBI1293E** このシステムには **DB2 Query Patroller** サーバーがセットアップされていません。

**ユーザーの処置:** DB2 インストーラーによって DB2 Query Patroller がインストールされていない場合、このインストーラーを使用して DB2 Query Patroller サーバーで使用するインスタンスを正しくセットアップします。インスタンスがセットアップされている場合は、それが `default.env` ファイルに記録されていて、そのファイルが世界中で読み取り可能であることを確認してください。`default.env` ファイルは以下のいずれかにあります。

`/usr/opt/db2_08_xx`、ここで `xx` は 01 または FPn、ここで `n` は FixPak 番号  
または `/opt/IBM/db2/V8.x`、ここで `x` は 1 または FPn、ここで `n` は FixPak 番号

---

**DBI1295E** インスタンス・リストを更新できませんでした。

**説明:** インスタンスをインスタンス・リストへ追加できなかったか、またはインスタンス・リストから除去できませんでした。インスタンス・リストからインスタンスを追加または削除しているときにエラーが発生しました。以下の状態のいずれかが発生していると考えられます。

- レジストリー・プロファイルに誤ったアクセス許可がある。
- プロファイル・レジストリーが正しく設定されていない。
- ファイル・システムに十分な空きスペースがない。

**ユーザーの処置:** 以下のディレクトリーのいずれかにあるインスタンス・リスト・プロファイルのファイル許可をチェックしてください。

`/usr/opt/db2_08_xx`、ここで `xx` は 01 または FPn、ここで `n` は FixPak 番号  
または `/opt/IBM/db2/V8.x`、ここで `x` は 1 または FPn、ここで `n` は FixPak 番号

---

**DBI1297E** インスタンス・プロファイル *profile-name* を更新できませんでした。

**説明:** インスタンス・プロファイル・レジストリーの更新中にエラーが発生しました。以下の状態のいずれかが発生していると考えられます。

- インスタンス・プロファイル・レジストリーに誤ったアクセス許可がある。
- プロファイル・レジストリーが正しく設定されていない。
- ファイル・システムに十分な空きスペースがない。

**ユーザーの処置:** 以下のディレクトリーのいずれかにあるインスタンス・リスト・プロファイルのファイル許可をチェックしてください。

`/usr/opt/db2_08_xx`、ここで `xx` は 01 または FPn、ここで `n` は FixPak 番号  
または `/opt/IBM/db2/V8.x`、ここで `x` は 1 または FPn、ここで `n` は FixPak 番号

**DBI1300N** **db2set** は DB2 プロファイル変数を表示、設定、あるいは取り除きます。

```
db2set [[[variable]=[value]]
 [-g|-i instance[
 node-number]]]
 [-all] [-null]
 [-r instance[node-number]]
 [-n DAS node[
 -u user[-p password]]]
 [-ll|-lr] [-v] [-ull|-ur]
 [-?|-h]
```

**説明:** コマンド・オプションは以下のとおりです。

- g** グローバル・プロファイル変数にアクセスする
- i** 現行あるいはデフォルト値の代わりに使用するインスタンス・プロファイルを指定する
- n** リモート DB2 Administration Server ノード名を指定する
- u** Administration Server 付加機構で使用するユーザー ID を指定する
- ul** ユーザー・プロファイル変数にアクセスする
- ur** ユーザー・プロファイル変数を最新表示する
- p** Administration Server 付加機構で使用するパスワードを指定する
- r** 与えられたインスタンスに対するプロファイル・レジストリーをリセットする
- l** すべてのインスタンス・プロファイルをリストする
- lr** サポートされたすべてのレジストリー変数をリストする
- v** Verbose モード

**-?** コマンド・ヘルプ・メッセージを表示する

**-h** **-?** オプションと同じ

**-all** 以下のように定義されているローカル環境変数のすべてのオカレンスを表示する

- 環境は、[e] で示されます。
- ユーザー・レベル・レジストリーは、[u] で示されます。
- ノード・レベル・レジストリーは、[n] で示されます。
- インスタンス・レベル・レジストリーは、[i] で示されます。
- グローバル・レベル・レジストリーは、[g] で示されます。

**-null** 指定されたレジストリー・レベルで変数の値を null に設定し、変数値の検索順序で定義されている次のレジストリー・レベルの変数を参照できないようにする。

**注:**

- 変数名なしの **db2set** では 定義されている変数をすべて表示します。
- **db2set <variable>** では **<variable>** の値を表示します。
- **db2set <variable>=(nothing)** では **<variable>** を削除します。
- **db2set <variable>=<value>** では **<variable>** の値を修正します。
- **db2set <variable> -null** では **<variable>** の値を NULL に設定します。
- **db2set <variable> -all** では 定義されているすべての **<variable>** の値を表示します。
- **db2set -ur** では 現行ユーザー・プロファイルを最新表示します。
- **db2set <variable> -ul** では ユーザー・レベルで定義された **<variables>** を表示します。

- `db2set -all` では、すべてのレジストリー・レベルで定義されているすべての変数を表示します。

---

**DBI1302E 無効なパラメーターを検出しました。**

**説明:** 無効なパラメーターが使用されました。

**ユーザーの処置:** 使用法のヘルプ・メッセージを調べるには、`-?` を使ってください。

---

**DBI1303W 変数が設定されていません。**

**説明:** 変数がプロファイル・レジストリーに設定されていません。

**ユーザーの処置:** 必要な処置はありません。

---

**DBI1304E 予期しないエラーです。**

**説明:** ツールで想定外のシステム・エラーが発生しました。

**ユーザーの処置:** DB2 サービス担当者に連絡してください。

---

**DBI1305E プロファイル・レジストリーが見つかりません。**

**説明:** ターゲット・マシンにはプロファイル・レジストリー設定がありません。

**ユーザーの処置:** DB2 のインストールでターゲット・マシンにレジストリーを作成してください。

---

**DBI1306N インスタンス・プロファイルが定義されていません。**

**説明:** インスタンスがターゲット・マシン・レジストリーで定義されていません。

**ユーザーの処置:** 既存のインスタンス名を指定するか、あるいは必要なインスタンスを作成してください。

---

**DBI1307N インスタンス・ノード・プロファイルが定義されていません。**

**説明:** インスタンス・ノードがターゲット・マシン・レジストリーで定義されていません。

**ユーザーの処置:** 必要な DB2 製品パーツをインストールしてレジストリーを作成してください。

---

**DBI1308E メモリー不足状態が発生しました。**

**説明:** ツールで「メモリー・リソース不足」エラーが発生しました。

**ユーザーの処置:** システムのメモリー・リソースが少なくなっています。不要なアプリケーションを終了するか、またはあとで再試行してください。

---

**DBI1309E システム・エラー。**

**説明:** ツールでオペレーティング・システム・エラーが発生しました。

**ユーザーの処置:** システム・エラーがレジストリーにアクセス中に発生しました。レジストリーがある場所のファイル・システムにメモリーが十分あるか、およびレジストリーがリモートの場合、LAN 接続が有効であるか確認してください。

---

**DBI1310E リモート・レジストリー・アクセスがサポートされていません。**

**説明:** ツールではリモート・レジストリー・オプションをサポートしていません。

**ユーザーの処置:** レジストリーにリモート・アクセスする方法についてはコマンド・リファレンスを参照してください。

---

**DBI1311I レジストリー・インスタンス・プロファイルをリスト中...**

---

**DBI1312I 定義済みグローバル変数をリスト中...**

---

**DBI1313I** 定義済みインスタンス変数をリスト中...

---

**DBI1314I** サポートされているすべてのレジストリー変数をリスト中...

---

**DBI1315W** 不明なメッセージ！

---

**DBI1316E** インスタンスが **MPP** ではありません。

**説明:** ターゲット・インスタンスが DB2 MPP インスタンスではありません。

**ユーザーの処置:** MPP インスタンスを選択してください。

---

**DBI1317E** インスタンス・ノードがすでにあります。

**説明:** インスタンス・ノードが重複して存在しています。

**ユーザーの処置:** 別のインスタンス・ノードを選択してください。

---

**DBI1320W** 警告:

**説明:** このスクリプトは、起こりうる問題のデバッグのために IBM サポートが有益な情報を収集するために使用されます。この情報は、機密情報場合があります。出力ファイル db2ginfo.txt は IBM サポートに送信する前に編集することができます。

**ユーザーの処置:** 上記の警告を認識および受諾したことを通知するには、このスクリプトを -y フラグ付きで実行してください。

---

**DBI1330W** 処置が別のインスタンスに影響する可能性があります。

**説明:** ファイル IWH.environment には、すべてのインスタンスのグローバル情報が入っています。

す。このファイルを更新すると、すべてのインスタンスに影響する可能性があります。続けますか？

---

**DBI1332E** IWH.environment のテンプレート・ファイルが見つかりません。

**説明:** IWH.environment のテンプレート・ファイルが必要です。

---

**DBI1351E** このコマンドを実行するには、インスタンス所有者でなければなりません。

**説明:** このコマンドを実行するには、DB2 Query Patroller サーバー・インスタンス所有者でなければなりません。

**ユーザーの処置:** DB2 Query Patroller サーバー・インスタンスがファイル default.env に正しく記録されていることを確認します。DB2 Query Patroller サーバー・インスタンス所有者としてログインし、このコマンドを再発行してください。default.env ファイルは以下のいずれかにあります。

/usr/opt/db2\_08\_xx、ここで xx は 01 または FPn、ここで n は FixPak 番号  
または /opt/IBM/db2/V8.x、ここで x は 1 または FPn、ここで n は FixPak 番号

---

**DBI1352E** インスタンス *instance-name* は **ESE** インスタンスではありません。

**説明:** DB2 Query Patroller サーバー / エージェントは、ESE インスタンスに作成する必要があります。

**ユーザーの処置:** 有効な ESE インスタンス名を使ってこのコマンドを再実行するか、またはこのコマンドを再実行する前に、新しい ESE インスタンスを作成してセットアップしてください。

---

**DBI1353E DB2 Query Patroller サーバーはすでにインスタンスにセットアップされています。**

**説明:** DB2 Query Patroller サーバーは 1 つの DB2 UDB インスタンスにのみセットアップできます。

**ユーザーの処置:** コマンド `dqplist` を実行して、DB2 Query Patroller サーバーとして使用されているインスタンスの名前を見つけます。インスタンス名が正しい場合には、サーバーのセットアップに `dqpct` を実行する必要はありません。エージェントをセットアップする場合は、正しいインスタンス名を使用する必要があります。このインスタンスがエージェントをセットアップする目的で使用されない場合、以下のコマンドを実行して除去します。

```
dqpdrop inst_name
```

次に、`dqpct` コマンドを再実行してください。

---

**DBI1354E インスタンス *instance-name* が DB2 Query Patroller サーバー・インスタンスではありません。**

**ユーザーの処置:** `dqplist` で正しいサーバー・インスタンス名を探して、コマンドを再実行してください。

---

**DBI1355I 使用法 :**

```
dqpct [-hl-?] -s|-a
-p PortName InstName
```

**説明:** 間違った引き数が入力されました。このコマンドの有効な引き数は以下のとおりです。

**PortName**

DB2 Query Patroller サーバー / エージェントで使用するポート名

**InstaName**

DB2 Query Patroller サーバー・インスタンスとして指定するインスタンスの名前

**-s** 指定のインスタンスに DB2 Query Patroller サーバーを作成する

**-a** 指定のインスタンスに DB2 Query Patroller エージェントを作成する

**-hl-?** 使用法情報を表示する

**-a** オプションは、複数パーティションのデータベース・インスタンスにのみ有効です。

**ユーザーの処置:** 正しい構文を使用してコマンドを再入力してください。

---

**DBI1356I 使用法 :**

```
dqplist [-hl-?]
```

**説明:** 間違った引き数が入力されました。このコマンドの有効な引き数は以下のとおりです。

**-hl-?** 使用法情報を表示する

**ユーザーの処置:** 正しい構文を使用してコマンドを再入力してください。

---

**DBI1357I 使用法 :**

```
dqpdrop [-hl-?] InstName
```

**説明:** 間違った引き数が入力されました。このコマンドの有効な引き数は以下のとおりです。

**InstName**

DB2 Query Patroller サーバーを除去したいインスタンスの名前

**-hl-?** 使用法情報を表示する

このコマンドは、DB2 Query Patroller サーバーが作成されたノードでのみ発行できます。

**ユーザーの処置:** 正しい構文を使用してコマンドを再入力してください。



---

**DBI1358W DB2 Query Patroller プロファイル・ファイルを変更できませんでした。**

**説明:** dqpprofile ファイルか dqpcshrc ファイルのいずれか、またはその両方を更新しようとしたが、失敗しました。考えられる原因は以下のとおりです。

- これらのファイルがインスタンス・ホーム・ディレクトリーの下 の sqllib ディレクトリーに存在しない。
- これらのファイルへの書き込み許可がない。
- /tmp ディレクトリーに一時ファイルを作成できなかった。

**ユーザーの処置:** これらのファイルが存在しているか、またファイルの許可について調べます。/tmp への書き込みが可能であることを確認してください。コマンドをもう一度発行してください。

---

**DBI1359E DB2 Query Patroller サーバーをインスタンス *instance-name* から除去できません。**

**説明:** DB2 Query Patroller サーバーを指定のインスタンスから除去しようとしたが、失敗しました。

**ユーザーの処置:** ファイル sqllib/cfg/dqplevel を手操作でインスタンス所有者のホーム・ディレクトリーから除去してください。また、ファイル default.env を編集して、DQPSEVER 行をこのファイルから除去します。default.env ファイルは以下のいずれかにあります。

/usr/opt/db2\_08\_xx、ここで xx は 01 または FPn、ここで n は FixPak 番号  
または /opt/IBM/db2/V8.x、ここで x は 1 または FPn、ここで n は FixPak 番号

---

**DBI1360E DB2 Query Patroller のセットアップを完了できません。**

**説明:** DB2 Query Patroller サーバーをセットアップしようとしたが、失敗しました。考えられる原因は以下のとおりです。

- DB2 Query Patroller サーバーのデータベースを作成できなかった。
- 指定のノードにノード・グループを作成できなかった。
- ノード・グループに表スペースを作成できなかった。

**ユーザーの処置:** データベースまたはノード・グループの作成が失敗した場合、これを手操作で作成してコマンドを再実行してください。

表スペースの作成が失敗した場合は、指定したパスへの書き込み許可を持っていることを確認します。表スペースを手操作で作成してみてください。

---

**DBI1361E DB2 Query Patroller サーバー・コードがインストールされていません。**

**説明:** DB2 Query Patroller サーバー・コードがインストールされていないため、現行操作の完了が要求されました。

**ユーザーの処置:** DB2 Query Patroller サーバー・コードをインストールして、コマンドを再試行してください。

---

**DBI1362W IWM スキーマ・オブジェクトが検出されました。**

**説明:** DB2 が IWM スキーマ・オブジェクトを検出しましたが、これは以前の IWM のインストールによって作成された可能性があります。

**ユーザーの処置:** 既存の IWM スキーマ・オブジェクトを保持したい場合、何もする必要はありません。既存のオブジェクトを除去して新しいオブジェクトを再作成したい場合は、dqpssetup コマ



ンドを `-o` フラグ付きで再実行してください。

---

**DBI1363E IWM スキーマ・オブジェクトを作成できませんでした。**

**説明:** IWM スキーマ・オブジェクトを作成しようとしたが、失敗しました。

**ユーザーの処置:** ファイル `db2_qp_schema` および `iwm_schema.sql` が両方ともディレクトリー `/usr/opt/db2_08_xx/bin` にあることを確認します。ここで `xx` は 01 または FPn、ここで `n` は FixPak 番号です。

両方のファイルが存在する場合、IWM スキーマ作成プロシージャの出力が `/tmp/iwmschcr.log` にあります。エラーを訂正して、`dqpsetup` コマンドを再実行してください。

---

**DBI1364E 前の DB2 Query Patroller スキーマ・オブジェクトが存在しません。**

**説明:** 移行の対象の旧 DB2 Query Patroller スキーマ・オブジェクトが存在しません。

**ユーザーの処置:** 正しい構文で `dqpsetup` を再実行してください。正しい構文を調べるには `'dqpsetup -h'` を使用してください。

---

**DBI1365E ノード番号が定義されていません。**

**説明:** ノード番号が `db2nodes.cfg` に定義されていません。

**ユーザーの処置:** `db2nodes.cfg` ファイルを調べて、必要な項目が存在することを確認します。`db2nodes.cfg` ファイルに記録されているものとまったく同じコード番号を使用して、コマンドを再入力してください。

---

**DBI1366E コンテナ `sms-path` はすでに使用中です。**

**説明:** 表スペースのコンテナとして指定されたディレクトリーはすでに存在します。

**ユーザーの処置:** 表スペースのコンテナに別の

パスを使用して、コマンドを再実行してください。

---

**DBI1367E `bind-list` にリストされているファイルをバインドできませんでした。**

**説明:** 指定のファイルにリストされたバインド・ファイルのバインドで問題が起きました。

**ユーザーの処置:** バインドを手操作で行ってください。

---

**DBI1368E DQP データベース `db2dbdft` に接続できませんでした。**

**説明:** このデータベースは存在しないと思われる。

**ユーザーの処置:** `dqpsetup` に必要なパラメーターをすべて使用して、新しい DQP インスタンスをセットアップするよう指定してください。このインスタンスを移行することはできません。

---

**DBI1370W ユーザー `inst-name` の `.profile` あるいは `.login` ファイルの修正はできません。**

**説明:** DB2 は、このユーザーの `.profile` ファイルまたは `.login` ファイルを修正することができません。これらのファイルが存在しないか、または書き込み許可がない可能性があります。変更を行わないと、DB2 Query Patroller を使用するためにこのユーザーでログインするたびに、手操作で環境を設定する必要があります。

**ユーザーの処置:** DB2 環境を自動的に設定する方法に関する詳細については、「概説およびインストール」をご覧ください。

---

**DBI1371E Query Patroller サーバーをセットアップするための引き数が見つかりません。**

**説明:** DB2 Query Patroller サーバーがインストールされている場合、Query Patroller インスタンスをサーバーとしてセットアップする必要があります

ます。そのため、以下の引き数を与えなければなりません。

1. ノード・グループ
2. ノード番号
3. データベース名
4. 表スペース
5. 結果表スペース
6. 表スペース・パス

オプション:

DMS 表スペース・サイズ

**ユーザーの処置:** 応答ファイルに必須またはオプション引き数、あるいは両方の引き数を指定してください。

---

**DBI1372E Query Patroller サーバー・インスタンスをセットアップできません。**

**説明:** Query Patroller サーバー・インスタンスをセットアップするには、DB2 Query Patroller サーバーがインストールされていなければなりません。

**ユーザーの処置:**

- DB2 Query Patroller サーバーをインストールしてください。
- Query Patroller エージェントをセットアップしたい場合、ノード・グループ、ノード番号、データベース名、表スペース、結果表スペース、表スペース・パス、および DMS 表スペース・サイズの引き数を取り除いてください。

---

**DBI1373E 新しい ESE インスタンスの作成中は、Query Patroller サーバー・インスタンスをセットアップできません。**

**ユーザーの処置:** ESE インスタンスを作成し、別々のステップでインスタンスを Query Patroller サーバーとしてセットアップしてください。

---

**DBI1375E ESE をインストールせずに、Query Patroller インスタンスをセットアップすることはできません。**

**ユーザーの処置:** ESE と DB2 Query Patroller エージェント / サーバーをインストールしてください。

---

**DBI1376E Query Patroller インスタンスをセットアップするには、IWM ユーザーのサービス名およびポート番号を与える必要があります。**

**ユーザーの処置:**

- Query Patroller インスタンスをセットアップしたい場合、IWM ユーザーのサービス名およびポート番号を入力してください。
- Query Patroller インスタンスをセットアップしたくない場合、ノード・グループ、ノード番号、データベース名、表スペース、結果表スペース、表スペース・パス、および DMS 表スペース・サイズの指定を応答ファイルから取り除いてください。

---

**DBI1377N db2isetup は、ご使用のコンピューターで適切な Java ランタイム環境を見付けられませんでした。Java ランタイム環境 v.13 が存在する場合は、JAVA\_HOME 環境変数を設定して、再度コマンドの実行を試行してください。それ以外の場合は、DB2 のインストールの手引きで、ご使用のオペレーティング・システムに対して推奨されている Java 環境に関する情報を確認してください。**

**DBI1378N** **db2setup** は、ご使用のコンピューターで適切な **Java** ランタイム環境を見付けられませんでした。**Java** ランタイム環境 **v.13** が存在する場合は、**JAVA\_HOME** 環境変数を設定して、再度コマンドの実行を試行してください。それ以外の場合は、**DB2** のインストールの手引きで、ご使用のオペレーティング・システムに対して推奨されている **Java** 環境に関する情報を確認してください。適切な **Java** ランタイム環境が得られない場合は、**doc\_install** スクリプトを使用するコマンドからインストールすることができます。このスクリプトの使用方法を表示するには、パラメーターを指定せずにこのスクリプトを実行してください。

**DBI1400N** **db2licm** コマンドの構文に誤りがあります。

**説明:** コントロール・センターがない場合には、**db2licm** ツールが基本ライセンス機能を実行します。それによって、ローカル・システムにインストールされたライセンスが追加、除去、リスト、および変更されます。

**db2licm** [-a filename]  
[-e HARD | SOFT]  
[-p prod-password  
REGISTERED | CONCURRENT]  
[-r prod-password]  
[-u prod-password num-users]  
[-n prod-password num-processors]  
[-l]  
[-v]  
[-?]

コマンド・オプションは以下のとおりです。

**-a** 製品のライセンスを追加します。有効な

ライセンス情報の入ったファイル名を指定します。これはライセンス製品 CD に収められていますが、お持ちでない場合は、IBM 担当員または販売代理店に連絡してください。

- n** 権利のあるプロセッサの数を更新します。この製品で使用できるプロセッサの数を指定してください。
- e** システムの制約方針を更新します。有効な値は **HARD** および **SOFT** です。**HARD** は非ライセンス要求が許可されないことを指定します。**SOFT** は非ライセンス要求が制限なしでログに記録されることを指定します。
- l** すべての製品を使用可能なライセンス情報とともにリストします。
- p** システムで使用するライセンス方針タイプを更新します。キーワード **CONCURRENT**、**REGISTERED**、または **CONCURRENT REGISTERED** を指定できます。
- r** 製品のライセンスを除去します。ライセンスが除去されると、製品機能は「試供版」モードになります。特定の製品のパスワードを取得するには、**-l** オプション付きでコマンドを呼び出します。
- u** 購入したユーザー権利数を更新します。権利を購入した製品のパスワード、およびユーザーの数を指定してください。
- v** バージョン情報を表示します。
- ?** ヘルプ情報を表示します。このオプションを指定すると、他のすべてのオプションは無視され、ヘルプ情報だけが表示されます。

**ユーザーの処置:** 有効なパラメーターを使用してコマンドを再入力してください。

---

**DBI1401I** コマンド行 DB2 ライセンス・マネージャー

**説明:** コントロール・センターがない場合には、db2licm ツールが基本ライセンス機能を実行します。それによって、ローカル・システムにインストールされたライセンスが追加、除去、リスト、および変更されます。

```
db2licm [-a filename]
 [-e prod-password HARD | SOFT]
 [-p prod-password
 REGISTERED | CONCURRENT]
 [-r prod-password]
 [-u prod-password num-users]
 [-n prod-password num-processors]
 [-l]
 [-v]
 [-?]
```

コマンド・オプションは以下のとおりです。

- a** 製品のライセンスを追加します。有効なライセンス情報の入ったファイル名を指定します。これはライセンス製品 CD に収められていますが、お持ちでない場合は、IBM 担当員または販売代理店に連絡してください。
- n** 権利のあるプロセッサの数を更新します。この製品で使用できるプロセッサの数を指定してください。
- e** システムの制約方針を更新します。有効な値は HARD および SOFT です。HARD は非ライセンス要求が許可されないことを指定します。SOFT は非ライセンス要求が制限なしでログに記録されることを指定します。
- l** すべての製品を使用可能なライセンス情報とともにリストします。
- p** システムで使用するライセンス方針タイプを更新します。キーワード

CONCURRENT、REGISTERED、または CONCURRENT REGISTERED を指定できます。

- r** 製品のライセンスを除去します。ライセンスが除去されると、製品機能は「試供版」モードになります。特定の製品のパスワードを取得するには、-l オプション付きでコマンドを呼び出します。
- u** 購入したユーザー権利数を更新します。権利を購入した製品のパスワード、およびユーザーの数を指定してください。
- v** バージョン情報を表示します。
- ?** ヘルプ情報を表示します。このオプションを指定すると、他のすべてのオプションは無視され、ヘルプ情報だけが表示されます。

---

**DBI1402I** ライセンスが正常に追加されました。

---

**DBI1403I** ライセンスが正常に除去されました。

---

**DBI1404N** 製品パスワードが見つかりません。

**説明:** 指定されたパスワードが無効であるか、またはこの製品のライセンスが nodelock ファイルに見つかりませんでした。

**ユーザーの処置:** -l オプションを指定してこのコマンドを発行し、入力されたパスワードがこのアクションを実行しようとしている製品にとって正しい製品パスワードであるかチェックします。nodelock パスワードを使用している場合は、この製品のライセンス・キーが nodelock ファイルにインストールされているかチェックしてください。

---

**DBI1405I** ライセンス方針タイプが正常に更新されました。

---

**DBI1406N** 無効なライセンス方針タイプです。

**説明:** 入力されたライセンス方針タイプが、指定された製品には無効でした。

**ユーザーの処置:** 有効なライセンス・ポリシーを入力してください。オプションは以下のとおりです。

- CONCURRENT
- REGISTERED
- CONCURRENT REGISTERED

---

**DBI1407N** 無効なライセンス証書ファイルです。

**説明:** ライセンス証書ファイルが正しい形式ではありません。

**ユーザーの処置:** 正しいライセンス証書形式を持つファイルの名前を入力してください。

---

**DBI1408N** ファイル *file-name* がオープンできませんでした。

**説明:** ファイルが見つからないか、ファイルへのアクセスが拒否されました。

**ユーザーの処置:** 存在していて、オープン可能なファイルの名前を入力し、コマンドを再試行してください。

---

**DBI1409N** 無効な制約方針タイプです。

**説明:** 指定された制約方針タイプはこの製品には無効です。

**ユーザーの処置:** 指定の製品がサポートする有効な制約方針タイプを入力してください。

---

**DBI1410I** 並列ユーザーが正常に更新されました。

---

**DBI1411I** 制約方針タイプが正常に更新されました。

---

**DBI1412W** ハード・ストップ制約方針は、ライセンス違反が生じたときに非ライセンス・ユーザーによる製品の使用を停止します。

---

**DBI1413W** ソフト・ストップ制約方針は、ライセンス違反をログに記録しますが、非ライセンス・ユーザーによる製品の使用を許可します。

---

**DBI1416N** ライセンスを自動的に **nodelock** ファイルに追加できませんでした。

**説明:** 戻りコードは “<return-code>” です。

**ユーザーの処置:** ライセンス証明書が読み取り可能であることを確認してください。また、手操作でライセンスを **nodelock** ファイルに入力することもできます。指示については、ライセンス・ファイルを参照してください。

---

**DBI1417N** 指定されたライセンスを **nodelock** ファイルから除去できません。

**説明:** 戻りコードは “<return-code>” です。

**ユーザーの処置:** この製品のライセンスが **nodelock** ファイルに存在することを確認してください。

---

**DBI1418I** このマシンでライセンスされたプロセッサ数が正しく更新されました。

---

**DBI1419N** ライセンスされたプロセッサ数を更新しているときにエラーが起きました。

**説明:** 戻りコードは “<return-code>” です。

---

**DBI1420N** この製品は、このライセンス・ポリシーのタイプをサポートしていません。

**説明:** 指定されたライセンス・ポリシーはこの製品に適用されないか、またはサポートされていません。

**ユーザーの処置:** 有効なライセンス・ポリシーを入力してください。

---

**DBI1421N** 指定された製品は、このシステムにインストールされていません。

**説明:** 製品がインストールされるまで、この製品のライセンス・ポリシーを構成することはできません。

**ユーザーの処置:** 製品をインストールしてからコマンドを実行するか、または正しい製品パスワードを指定してください。システムにインストールされている製品をリストするには、db2licm -l を使用してください。

---

**DBI1422N** 並列ユーザーの数は更新されませんでした。

**説明:** 戻りコードは “<return-code>” です。

**ユーザーの処置:** この製品で並列ポリシーが可能であることを確認してください。

---

**DBI1423N** このオプションには、インスタンスの作成が必要です。

**説明:** この処置を行うために必要な機能は、インスタンスの作成後にアクセス可能になります。

**ユーザーの処置:** インスタンスを作成し、このコマンドを出し直してください。

---

**DBI1424N** プロセッサ情報にアクセスしているときに、予期しないエラーが起りました。

**説明:** 戻りコードは “<return-code>” です。

**ユーザーの処置:** なし。

---

**DBI1425E** **DB2 OLAP Server** のライセンスを更新することはできません。  
**DB2 OLAP** 処理は、現在アクティブにあります。

**説明:** DB2 OLAP Server の実行時に、DB2 で DB2 OLAP Server のライセンスを更新することはできません。

**ユーザーの処置:** OLAP ライセンスを更新するには、OLAP プロセスをすべて停止してからこの DB2 ライセンスを再インストールしてください。

---

**DBI1426I** この製品は現在、この製品のライセンス・コピーに付属するご使用条件の指定に基づいてご使用いただくことができます。この製品をご使用いただくには、以下のディレクトリーにある **IBM** ご使用条件への同意が必要です: *dir-name*

---

**DBI1427I** この製品は現在、この製品の評価版 (「お試し版」) に付属するご使用条件の指定に基づいてご使用いただくことができます。この製品をご使用いただくには、以下のディレクトリーにある **IBM** ご使用条件への同意が必要です: *dir-name*

---

**DBI1428N** ライセンスされたプロセッサ数を更新しているときにエラーが起りました。

**説明:** 入力されたライセンス・プロセッサの数が、この製品で許可されているライセンス・プロセッサの最大数を超過しています。



**ユーザーの処置:** 定義されている最大数を超えないライセンス・プロセッサの数を入力してください。ご使用のシステムのプロセッサの数がこの製品に対して許可されているプロセッサの最大数を超える場合は、IBM 担当員または許可されている販売業者にご連絡ください。

---

**DBI1429N** この製品は、このライセンス・ポリシーの組み合わせをサポートしていません。

**ユーザーの処置:** 有効なライセンス・ポリシーの組み合わせを入力してください。たとえば、有効な組み合わせとして "CONCURRENT REGISTERED" を指定できます。

---

**DBI1430N** ライセンス日付がオペレーティング・システムの日付よりも後であるため、ライセンスを **nodelock** ファイルに追加できませんでした。

**ユーザーの処置:** 証明書ファイルをチェックして、ライセンスの開始日が現在日付 (オペレーティング・システムに設定されている日付) よりも前であることを確認してください。

---

**DBI1431N** このユーザーは、指定されたアクションを実行するための適切な権限を持っていません。

**説明:** このプログラムは、root ユーザー ID または SYSADM 権限を持つユーザー ID でしか実行できません。

**ユーザーの処置:** このコマンドを実行するのに適切な許可を持つユーザー ID を使ってログインしてください。

---

**DBI1500I** 使用法 :

**db2inst [-f response-file]**

**説明:** 間違った引き数が入力されました。コマンドが、「DB2 インストーラー」応答ファイルの

パスの前にオプションのフラグ **-f** を指定しました。

**ユーザーの処置:** 有効な引き数を指定して、コマンドを再入力してください。

---

**DBI1501E** 内部エラーが発生しました。

**説明:** 内部処理を実行中にエラーを検出しました。

**ユーザーの処置:** ファイルが置かれているファイル・システムが損傷していないか調べてください。問題が解決しない場合は、IBM サポートに連絡して以下の情報を伝えてください。

1. メッセージ番号
2. 内部エラー・メッセージ記述
3. 問題記述

---

**DBI1502E** ファイル *file-name* のオープンあるいは読み取り中にエラーが発生しました。

**説明:** ファイルのオープンあるいは読み取り中にエラーが発生しました。以下のエラーのいずれかが発生していると考えられます。

- ファイルのオープンあるいは読み取りで入出力エラーが起きた。
- 予定の値がファイルに見つからない。
- ファイル内のデータの順序が誤っている。

**ユーザーの処置:** ファイルがユーザーによって修正されている場合、ファイル内のデータが有効かどうか調べてください。ファイルがユーザーによって修正されていない場合、以下の情報を IBM サポート担当者に連絡してください。

1. メッセージ番号
2. メッセージ記述
3. 問題記述

---

**DBI1503E** ファイル *file-name* のオープンあるいは書き込み中にエラーが発生しました。

**説明:** ファイルのオープンあるいは書き込み中にエラーが発生しました。以下のエラーのいずれかが発生していると考えられます。

- 与えられたディレクトリーに誤ったアクセス許可がある。
- ファイル・システムに十分な空きスペースがない。

**ユーザーの処置:** 親ディレクトリーの許可が正しく設定されているかチェックしてください。問題が解決しない場合は、IBM サポートに連絡して以下の情報を伝えてください。

1. メッセージ番号
2. メッセージ記述
3. 問題記述

---

**DBI1504E** メッセージ・ファイルのオープンでエラーが起きました。

**説明:** “DB2 インストーラー”メッセージ・カタログ・ファイルのオープンおよび読み取りをしようとして失敗しました。DB2 Installer を開始する前に、以下の 2 つのリソース・ファイル -- db2inst.rcx と db2inst.cat を以下のロケーションで検索します。

- DB2 製品ディレクトリー、または
- 配布先メディア、または
- 環境変数 DB2IRCX および DB2ICAT

**ユーザーの処置:** 環境変数 DB2IRCX と DB2ICAT を db2inst.rcx と db2inst.cat の絶対ロケーションに設定してください。

---

**DBI1505E** メモリーを割り振ろうとして失敗しました。

**説明:** メモリーを割り振ろうとして、エラーを検出しました。

**ユーザーの処置:** システム上で実行中の他のアプリケーションで、メモリーを大量に使用している可能性のあるものを終了してください。問題が解決しない場合は、IBM サポートに連絡して以下の情報を伝えてください。

1. メッセージ番号
2. メッセージ記述
3. 問題記述

---

**DBI1507E** DB2 インストーラーのインスタンスはすでに開始しています。

**説明:** DB2 インストーラーを開始しようとして、エラーを検出しました。DB2 インストーラーの別のインスタンスがまだ実行中です。

**ユーザーの処置:** DB2 インストーラーのすべてのインスタンスを終了して、インストール・プロセスを再始動してください。問題が解決しない場合、ロック・ファイル /tmp/db2inst.lck を除去してから、DB2 インストーラーを再始動してください。

---

**DBI1515E** サブプロセスに対するリソースを割り振ろうとして失敗しました。

**説明:** サブプロセスを開始しようとして、エラーを検出しました。

**ユーザーの処置:** システム上で実行中の他のアプリケーションで、リソースを大量に使用している可能性のあるものを終了してください。問題が解決しない場合は、IBM サポートに連絡して以下の情報を伝えてください。

1. メッセージ番号
2. 問題記述

---

**DBI1516E** サブプロセスを終了しようとして失敗しました。

**説明:** サブプロセスを終了しようとして、エラーを検出しました。

**ユーザーの処置:** システム上で実行中の他のアプ



リケーションで、リソースを大量に使用している可能性のあるものを終了してください。問題が解決しない場合は、IBM サポートに連絡して以下の情報を伝えてください。

1. メッセージ番号
2. 問題記述

---

**DBI1517E** サブプロセスでコマンドを実行しようとして失敗しました。

**説明:** サブプロセスでコマンドを実行しようとして、エラーを検出しました。以下のいずれかの問題が発生しました。

- コマンドが存在しない。
- コマンド検索パスが不完全である。
- コマンドに誤ったアクセス許可がある。
- システム・リソースに問題がある。

**ユーザーの処置:** 問題を訂正して再度試行してください。問題が解決しない場合は、IBM サポートに連絡して以下の情報を伝えてください。

1. メッセージ番号
2. 問題記述

---

**DBI1520E** 端末表示の最小サイズは 24 x 80 です。

**説明:** 現在の端末またはウィンドウ・サイズが十分な大きさではありません。このプログラムでは最低 24 行 80 列の端末表示が必要です。

**ユーザーの処置:** 端末またはウィンドウ・サイズをチェックして再試行してください。

---

**DBI1521E** 端末機能情報の読み取りに失敗しました。

**説明:** 端末表示の初期化をしようとして、エラーを検出しました。このエラー・タイプが発生する場合は、以下の 2 つが考えられます。

- この機能が、環境変数 TERM を検出できないか、あるいは端末についての正しい terminfo データベース項目を検出できないかのいずれかである。
- 端末表示を初期化中に使用可能なメモリー・スペースが十分でない。

**ユーザーの処置:** 環境変数 TERM を正しい端末タイプに設定してください。問題が解決しない場合には、システム上で実行中の他のアプリケーションで、メモリーを大量に使用している可能性のあるものを終了してください。それでも問題が解決しない場合は、IBM サポートに連絡して以下の情報を伝えてください。

1. メッセージ番号
2. 問題記述

---

**DBI1522E** 端末機能情報のリストアに失敗しました。

**説明:** 端末表示を元の状態にリストアしようとして、エラーを検出しました。

**ユーザーの処置:** 環境変数 TERM を正しい端末タイプに設定してください。問題が解決しない場合には、システム上で実行中の他のアプリケーションで、メモリーを大量に使用している可能性のあるものを終了してください。それでも問題が解決しない場合は、IBM サポートに連絡して以下の情報を伝えてください。

1. メッセージ番号
2. 問題記述

---

**DBI1530E** イメージ・ファイルをアンパックできません。

**説明:** 配布されたメディアからの tar イメージをアンパック中にエラーが発生しました。以下の状態のいずれかが発生しました。

- 与えられたパス名が存在していないディレクトリーである。
- ファイル・システムに十分なスペースがない。
- 与えられたパスのアクセス許可が誤っている。

- tar コマンドが現行検索パスで見つからない。

**ユーザーの処置:** インストール処理を再始動して、正しいパス名で再試行してください。

---

**DBI1540E 無効な配布メディア・パスが指定されました。**

**説明:** インストール可能なイメージを配布メディアに位置指定しようとして、エラーを検出しました。以下のエラーのいずれかが発生していると考えられます。

- 与えられたパスと関連したファイル・システムが、正しく取り付けられていない。
- 誤ったアクセス許可が、与えられたパスに割り当てられている。

**ユーザーの処置:** パスをチェックしてコマンドを再試行してください。UNIX のどのファイルおよびディレクトリーの名前でも、大文字小文字が区別されます。

---

**DBI1541E 与えられた状況ファイルからのオープンまたは読み取りに失敗しました。**

**説明:** 状況ファイルからのオープンまたは読み取りを行おうとしてエラーを検出しました。状況ファイルを手動で修正していない場合、以下のいずれかのエラーが発生しました。

- 誤ったファイル・パスが指定された。
- 誤ったアクセス許可が、状況ファイルに割り当てられている。
- 与えられたパスと関連したファイル・システムが、正しく取り付けられていない。

**ユーザーの処置:** パスをチェックしてコマンドを再試行してください。UNIX のどのファイルおよびディレクトリーの名前でも、大文字小文字が区別されます。問題が解決しない場合は、IBM サポートに連絡して以下の情報を伝えてください。

1. メッセージ番号
2. 問題記述

### 3. 状況ファイル

---

**DBI1550E ファイル・システムに十分なディスク・スペースがありません。**

**説明:** ファイル・システムにもっと多くのディスク・スペースを割り振ろうとしてエラーが発生しました。

**ユーザーの処置:** 指定の処理に十分なスペースが解放されてから、コマンドを再試行してください。

---

**DBI1570I 使用法 :**

**db2olset InstName**

**説明:** 間違った引き数が db2olset コマンドに入力されました。このコマンドの有効な引き数は以下のとおりです。

**-hl-?** 使用法情報を表示する

**-d** デバッグ・モードをオンにする

**InstName**

OLAP スターター・キットで使用できる  
ようセットアップしたいインスタンスの  
名前

**ユーザーの処置:** 以下のようにコマンドを入力し直してください。

db2olset InstName

---

**DBI1631E サンプル・データベースを作成中にエラーが発生しました。**

**説明:** 以下の状態のいずれかが発生しました。

- ファイル・システムに十分なディスク・スペースがありません。
- DB2 製品が正しくインストールされていないか、あるいはコンポーネント・ファイルに抜けがある。

**ユーザーの処置:** ディスク・スペースの問題ではない場合、製品の再インストールを行います。問

題が解決しない場合、トレースをオンにして CLP を介してデータベースを作成するステップを再試行してください。次に、ファイルにトレース情報を保管して、IBM サポート担当者に以下の情報を伝えてください。

- メッセージ番号
- 問題記述
- トレース・ファイル

---

#### **DBI1632E 「自動開始」インスタンスの構成中にエラーが発生しました。**

**説明:** 以下の状態のいずれかが発生しました。

- インスタンスを「自動開始」する機能がすでに使用可能である。
- “/etc/inittab” ファイルに入力を追加しようとして、エラーが発生した。これはファイル許可の問題である場合があります。
- db2uit ツールがない。

**ユーザーの処置:** 上記 3 つのいずれかの状態に当てはまる場合、問題を訂正してコマンドを再試行してください。問題が解決しない場合は、IBM サポートに連絡して以下の情報を伝えてください。

- メッセージ番号
- 問題記述

---

#### **DBI1633E インスタンスを「自動開始」する構成を削除中にエラーが発生しました。**

**説明:** 以下の状態のいずれかが発生しました。

- インスタンスを「自動開始」する機能が使用できない。
- “/etc/inittab” ファイルの入力を削除しようとして、エラーが発生した。これはファイル許可の問題である場合があります。
- db2uit ツールがない。

**ユーザーの処置:** 上記 3 つのいずれかの状態に当てはまる場合、問題を訂正してコマンドを再試

行してください。問題が解決しない場合は、IBM サポートに連絡して以下の情報を伝えてください。

- メッセージ番号
- 問題記述

---

#### **DBI1634W /etc/rc.db2v08 を更新して、dlfs ファイル・システムを自動マウントすることができません。**

**説明:** /etc/rc.db2v08 を更新して、システム・リブート時に dlfs ファイル・システムの自動マウントを可能にしようとしたましたが、失敗しました。

**ユーザーの処置:** ファイル /etc/rc.db2v08 を手操作で編集し、以下の行を追加してください。

```
if [-x /etc/rc.dlfs]; then
 /etc/rc.dlfs
fi
```

---

#### **DBI1635E Administration Server グループ・リストに追加中に、エラーが発生しました。**

**説明:** DB2 インスタンスを Administration Server グループ・リストに追加しようとして、エラーを検出しました。

**ユーザーの処置:** このコマンドを NIS クライアントで実行中の場合、DB2 インスタンスのグループ名をご使用の NIS サーバーの Administration Server の第 2 グループに追加してみてください。

---

#### **DBI1637W 「了解」をクリックすると、前のインスタンス設定は失われます。**

**ユーザーの処置:** 続行するには「了解」をクリックしてください。取り消すには「取消」をクリックしてください。

---

**DBI1639E 新規インスタンスをセットアップできません。**

**説明:** 与えられたインスタンス名は既存のインスタンスに属していません。

**ユーザーの処置:**

- 新しいインスタンスを作成してください。
  - 対話式インストーラーを使用している場合、インスタンス作成パネルで新しいインスタンスを作成するよう選択してください。
  - 応答ファイル・インストーラーを使用している場合、ユーザーについての追加情報を与えてください。
- 既存のインスタンスの名前を入力してください。

---

**DBI1640W 指定されたインスタンスはサーバー・インスタンスではありません。**

**説明:** DB2 サーバー・インスタンスとして構成できるのは、DB2 コンポーネントの一部だけです。

**ユーザーの処置:**

- 指定されたインスタンスがクライアント・インスタンスで、DB2 サーバー製品がインストールされている場合、db2iupdt を実行してクライアント・インスタンスをサーバー・インスタンスに更新してください。
- 指定されたインスタンスが Data Link Administration Server である場合、インストーラーを終了し、必要であれば、インスタンスをコマンド行でセットアップしてください。

---

**DBI1651E 指定の UID は無効です。**

**説明:** 無効な UID が入力されています。以下の状態のいずれかが発生しました。

- 与えられた UID はシステム上の既存のユーザーにすでに割り当てられている。
- 与えられた UID が大きすぎるか、あるいは無効文字が入っている。

- 与えられた UID が 100 より小さいか等しい。

**ユーザーの処置:** 別のユーザー ID で再試行してください。

---

**DBI1652E 指定のユーザー名が無効です。**

**説明:** 無効な名前が入力されました。以下の状態のいずれかが発生しました。

- 与えられたユーザー名がシステムにすでに存在している。
- 与えられたユーザー名がシステムにすでに存在しているが別のグループ名に属している。
- 与えられたユーザー名が大きすぎるか、あるいは無効文字が入っている。

**ユーザーの処置:** 別のユーザー名で再試行してください。

---

**DBI1653E 指定のグループが無効です。**

**説明:** 無効なグループ ID あるいはグループ名が入力されています。以下の状態のいずれかが発生しました。

- 与えられたグループ ID がすでに存在しているがシステムの別のグループ名に属しているか、あるいは与えられたグループ名がすでに存在しているが別のグループ ID に属している。
- 与えられたグループ ID あるいはグループ名が大きすぎるか、あるいは無効文字が入っている。
- 与えられたグループ ID が 100 より小さいか等しい。

**ユーザーの処置:** 別のグループ ID あるいはグループ名で、再試行してください。

---

**DBI1654E 指定したディレクトリーが無効です。**

**説明:** 無効なディレクトリーが入力されています。以下の状態のいずれかが発生しました。

- 与えられたユーザーはすでに存在するがシステム上に別のホーム・ディレクトリーを指定している。
- 与えられたディレクトリーが大きすぎるか、あるいは無効文字が入っている。

**ユーザーの処置:** システム上で指定されたユーザーのホーム・ディレクトリーあるいは別のディレクトリーを指定して再試行してください。

---

**DBI1655E 指定されたパスワードが無効です。**

**説明:** 無効なパスワードが入力されています。以下の状態のいずれかが発生しました。

- 与えられたパスワードの長さがゼロである。
- 与えられたパスワードが検証パスワードと一致しない。
- 与えられたパスワードに無効文字が入っている。

**ユーザーの処置:** 別のパスワードで再試行してください。

---

**DBI1657E インスタンス名はすでに使用されています。**

**説明:** 同じ名前を持つインスタンスがすでにシステムに存在します。

**ユーザーの処置:** “db2ilist” コマンドによって報告されたインスタンスのリストで確認してください。別のユーザー名で再試行してください。

---

**DBI1701E 指定された 1 つまたは複数の TCP/IP パラメーターが無効です。**

**説明:** 以下の状態のいずれかが発生しました。

- サービス名の長さが 14 文字を超えています。
- 指定されたポート番号は範囲外にあります。この番号は 1024 から 65535 の間の数値にしてください。

**ユーザーの処置:** 問題を訂正して再度試行してください。

---

**DBI1702E 指定されたサービス名あるいはポート番号は TCP/IP サービス・ファイルに存在する値と対立します。**

**説明:** ユーザーによって入力されたサービス名あるいはポート番号は TCP/IP サービス・ファイルに存在する値と競合します。サービス名は別のポート番号ですすでに使用されている可能性があるか、あるいはポート番号が別のサービス名ですすでに使用されている可能性があります。

**ユーザーの処置:** サービス・ファイルにすでにある項目と競合しないサービス名およびポート番号を指定してください。

---

**DBI1703E 有効なサービス名あるいはポート番号が見つかりません。**

**説明:** このインスタンスの有効なサービス名あるいはポート番号が TCP/IP サービス・ファイルに見つかりません。この情報は存在しないか、無効かのいずれかです。

“DB2 Extended Server Edition” 製品をインストールした場合、このコマンドの使用前に、インスタンスのサービス名を指定する必要があります。

**ユーザーの処置:** 詳しくは、「DB2 Enterprise - Extended Edition」を参照してください。このインスタンスの有効なサービス名およびポート番号を指定して、このコマンドを再入力してください。

---

**DBI1704W TCP/IP はシステムで実行されていません。**

**説明:** TCP/IP はシステムで実行されていません。TCP/IP サービス・ファイルは更新されません。

---

**DBI1709E TCP/IP サービス・ファイルを更新中にエラーが発生しました。**

**説明:** サービス名およびポート番号の TCP/IP サービス・ファイルへの追加の試行に失敗しました。

**ユーザーの処置:** ユーザーがノード・ディレクトリー項目のポート番号ではなく、サービス名を使用したい場合、ノードを手動でアンカタログしてからこのサービス名を使用して再度カタログしてください。サービス・ファイルも手操作で更新してください。ネットワーク情報サービス (NIS) が使用されている場合、ローカル・サービス・ファイルを更新する可能性があります、NIS サーバーは手操作で更新してください。この場合、ノードもポート番号を使用してカタログされました。

---

**DBI1711E 指定された 1 つまたは複数の IPX/SPX パラメーターが無効です。**

**説明:** 以下の状態のいずれかが発生しました。

- fileserv, objectname, あるいは ipx\_socket パラメーターが null である。
- fileserv パラメーターが \* に設定されているが objectname は \* に設定されていない。
- 指定された objectname あるいは ipx\_socket 値が固有でない。
- 指定された ipx\_socket 値が有効範囲にない。

**ユーザーの処置:** 問題を訂正して再度試行してください。

---

**DBI1715E NetWare ユーザー ID または NetWare パスワードが無効です。**

**説明:** 指定する NetWare ユーザー ID および NetWare パスワードは存在するもので NetWare ファイル・サーバーの DB2 オブジェクト名を登録するために使用され、監視あるいはそれと同等の権限が必要です。

**ユーザーの処置:** 問題を訂正して再度試行してください。

---

**DBI1720E 指定されたユーザー ID あるいはグループ ID は NIS が実行中のため追加されません。**

**説明:** NIS がシステムで実行中です。新規ユーザー ID あるいはグループ ID は NIS がシステムで実行中の間は作成されません。

**ユーザーの処置:** 以前作成されたユーザー ID およびグループ ID を使用してください。

---

**DBI1722E 指定されたサービスは NIS が実行中のため /etc/services ファイルに追加されません。**

**説明:** NIS がシステムで実行中です。新規サービスは NIS がシステムで実行中の間は /etc/services に作成されません。

**ユーザーの処置:** /etc/services ファイルの以前作成されたサービス名およびポート番号を使用して再試行してください。

---

**DBI1725W データウェアハウス・エージェント用のエージェント・デーモンおよびロガー・サービスを作成できません。**

**説明:** TCP/IP はシステムで実行されていません。インストールは続行されますが、エージェント・デーモンとロガー・サービスを手操作で作成する必要があります。

**ユーザーの処置:** システムで TCP/IP を活動化し、インストーラーを実行してください。

---

**DBI1740W セキュリティー・リスクです。**

**説明:** ここでデフォルト・インスタンスのユーザー ID を使用すると、このインスタンスにシステム上の他のインスタンスに対するフル・アクセス許可が与えられます。セキュリティ上の理由



から別のユーザー ID を使用することをお勧めします。

**ユーザーの処置:** セキュリティー・リスクの可能性を回避するには、別のユーザー ID を使用してください。

---

#### **DBI1741W プロトコルが検出されません。**

**説明:** 選択したプロトコルが検出されません。DB2 インストーラーはプロトコルに必要な設定のすべてを更新することはできません。ただし、このプロトコルの設定に対する値は提供することができます。

**ユーザーの処置:** 処置は必要ありません。

---

#### **DBI1744W DB2 Data Links Manager 管理者が作成されていません。**

**説明:** DB2 Data Links Manager 管理者を作成していません。DB2 Data Links Manager 管理者を作成していないと、DB2 Data Links Manager を管理することができません。

**ユーザーの処置:** 処置は必要ありません。

---

#### **DBI1745W Administration Server が作成されません。**

**説明:** Administration Server を作成していません。Administration Server がなければ、DB2 データベースへの接続の構成を自動で行なうクライアント・ツールをサポートするサービスを提供できず、サーバー・システムまたはコントロール・センターを使用してリモート・クライアントから DB2 を管理することができません。

**ユーザーの処置:** 処置は必要ありません。

---

#### **DBI1746W DB2 インスタンスは作成されません。**

**説明:** DB2 インスタンスを作成していません。DB2 インスタンスはデータの保管およびアプリケーションの実行環境です。

**ユーザーの処置:** 処置は必要ありません。

---

#### **DBI1747W 指定されたディレクトリーはすでに存在します。**

**説明:** ホーム・ディレクトリー・フィールドで指定されたディレクトリーはすでに存在します。このディレクトリーの使用を選択すると、許可上の問題が発生する可能性があります。

**ユーザーの処置:** これが問題となる場合には、別のディレクトリーを選択してください。

---

#### **DBI1750W セキュリティー・リスクです。**

**説明:** ここでデフォルト・インスタンスのユーザー ID を使用すると、このインスタンスにシステム上の他のインスタンスに対するフル・アクセス許可が与えられます。セキュリティ上の理由から別のユーザー ID を使用することをお勧めします。

**ユーザーの処置:**

- 処理を継続する場合には「了解」を押してください。
- 「取消」を押して別のユーザー ID を入力してください。

---

#### **DBI1751W プロトコルが検出されません。**

**説明:** 選択したプロトコルが検出されません。DB2 インストーラーはプロトコルに必要な設定のすべてを更新することはできません。ただし、このプロトコルの設定に対する値は提供することができます。

**ユーザーの処置:**

- プロトコルの設定を提供するのであれば「了解」を押してください。
- 無視するのであれば「取消」を押してください。

---

**DBI1753W Administration Server が作成されません。**

**説明:** Administration Server を作成していません。DB2 Administration Server なしでは、データウェアハウス・エージェントは完全にセットアップされません。

**ユーザーの処置:**

- 処理を継続する場合には「了解」を押してください。
- 戻るには、「取消」を押してください。

---

**DBI1754W DB2 Data Links Manager 管理者が作成されていません。**

**説明:** DB2 Data Links Manager 管理者を作成していません。DB2 Data Links Manager 管理者を作成していないと、DB2 Data Links Manager を管理することができません。

**ユーザーの処置:**

- 処理を継続する場合には「了解」を押してください。
- 戻るには、「取消」を押してください。

---

**DBI1755W Administration Server が作成されません。**

**説明:** Administration Server を作成していません。Administration Server がなければ、DB2 データベースへの接続の構成を自動で行なうクライアント・ツールをサポートするサービスを提供できず、サーバー・システムまたはコントロール・センターを使用してリモート・クライアントからDB2 を管理することができません。

**ユーザーの処置:**

- 処理を継続する場合には「了解」を押してください。
- 戻るには、「取消」を押してください。

---

**DBI1756W DB2 インスタンスは作成されません。**

**説明:** DB2 インスタンスを作成していません。DB2 インスタンスはデータの保管およびアプリケーションの実行環境です。

**ユーザーの処置:**

- 処理を継続する場合には「了解」を押してください。
- 戻るには、「取消」を押してください。

---

**DBI1757W 指定されたディレクトリーはすでに存在します。**

**説明:** ホーム・ディレクトリー・フィールドで指定されたディレクトリーはすでに存在します。このディレクトリーの使用を選択すると、許可上の問題が発生する可能性があります。

**ユーザーの処置:**

- このディレクトリーを使用するばあいは「了解」を押してください。
- 戻るには、「取消」を押してください。

---

**DBI1758W DB2 インスタンスまたは Administration Server を検出しました。**

**説明:** 全 DB2 製品の除去が選択されました。しかし、DB2 インスタンスまたは Administration Server がシステムで検出されました。最初にこれらのインスタンスをドロップせずに DB2 製品を除去すると、DB2 インスタンスが後で正しく機能できない場合があります。

**ユーザーの処置:**

- 処理を継続する場合には「了解」を押してください。
- 戻るには、「取消」を押してください。



---

**DBI1759W DB2 インスタンスまたは Administration Server 構成を検出しました。**

**説明:** 選択した Administration Server がありません。これによって、Administration Server および DB2 インスタンス構成が廃棄されてしまいます。

**ユーザーの処置:**

- 処理を継続する場合には「了解」を押してください。
- 戻るには、「取消」を押してください。

---

**DBI1760E *pkg-name* をセットアップするには、パッケージをインストールする必要があります。**

**説明:** インスタンスの DB2 コンポーネントを構成するよう選択しました。パッケージがインストールされていなければ、これを行うことはできません。

**ユーザーの処置:** パッケージをインストールして、もう一度やり直してください。

---

**DBI1761W ソフトウェア前提条件が、ファイル・セットまたはパッケージ *pkg-name* で違反しています。**

**説明:** ソフトウェアの前提条件を上記ファイル・セットまたはパッケージで検証中にエラーが起きました。処理を続行すると、製品が正しく機能しない可能性があります。前提条件をインストールしないまま処理を続行しますか？

**ユーザーの処置:**

- 前提条件をインストールしないまま処理を続行する場合には「了解」を押してください。
- インストールをやめるのであれば「取消」を押してください。

---

**DBI1762W ソフトウェア依存関係が、ファイル・セットまたはパッケージ *pkg-name* で違反しています。**

**説明:** ソフトウェアの依存関係を上記ファイル・セットまたはパッケージで検証中にエラーが起きました。処理を続行すると、ソフトウェア依存関係をすべて削除する可能性があります。ソフトウェア依存関係を除去しますか？

**ユーザーの処置:**

- 処理を継続する場合には「了解」を押してください。
- 戻るには、「取消」を押してください。

---

**DBI1763I 英語の HTML ドキュメンテーション・ファイルもインストールされません。**

**説明:** 英語の HTML ドキュメンテーション・ファイルは、英語以外の HTML ドキュメンテーション・ファイルをインストール対象として選択した場合に必要です。これは、ユーザーが DB2 ドキュメンテーション・ライブラリー全体を検索することができるかどうかを確認するためのものです。

---

**DBI1765W 存在しないユーザー ID *InstName* がレジストリーから検出されました。**

**説明:** レジストリーに存在しないユーザー ID が入っています。この状態は、最初にインスタンスをドロップせずにユーザー ID のみを除去したために発生します。DB2 の通常の操作には影響しませんが、レジストリーのインスタンス・リストからこのユーザー ID を除去することをお勧めします。

**ユーザーの処置:** 以下のコマンドを指定してレジストリーからこのユーザー ID を除去してください。  
db2iset -d "<InstName>"

---

**DBI1766W** *userId* の第 2 グループ・リストを変更することができません。

**説明:** 与えられたユーザー ID の第 2 グループ・リストを変更しようとしてコード “<code>” が返されました。以下の状態のいずれかが発生しました。

- NIS が実行中である。
- 与えられたユーザー ID で現在実行中の処理がある。

**ユーザーの処置:** Administration Server が正しく機能するように、グループ ID “<groupId>” をユーザー ID “<userId>” の第 2 グループ・リストに追加する必要があります。

- 与えられたユーザー ID で実行中の処理がある場合には、この処理をすべて終了して、上記の指示にしたがってこのユーザー ID の 2 次グループ・リストをセットアップしてください。
- このコマンドを NIS クライアントで実行中の場合、上記の指示にしたがってご使用の NIS サーバーにあるユーザー ID の第 2 グループをセットアップしてください。

---

**DBI1767W** DB2 レジストリーには破壊された情報が入っている可能性があります。

**説明:** レジストリーには無効なインスタンス情報が入っている可能性があります。これは、インスタンスあるいは Administration Server が、DB2 製品を除去する前に、正しく除去されていないことが原因となっている可能性があります。製品を再インストールして、レジストリー中の既存のインスタンス情報を保持したい場合には、「取消」を押してください。そうでない場合には、「了解」を押して、レジストリーを再作成します。レジストリーを再作成しますか。

**ユーザーの処置:**

- レジストリーを再作成するには、「了解」を押してください。

- 現在のレジストリー情報を保持するには、「取消」を押してください。

---

**DBI1768W** *db2profile* をユーザー *inst-name* の **.profile** ファイルに、あるいは *db2cshrc* をこのユーザーの **.login** ファイルに追加できません。

**説明:** DB2 は、このユーザーの .profile ファイルまたは .login ファイルを修正することができません。これらのファイルが存在しないか、または書き込み許可がない可能性があります。修正をしないと、DB2 を使用するために、このユーザーでログインするたびに、手動で環境を設定する必要があります。

**ユーザーの処置:** 以下のいずれかの処理を行ってください。

- .profile に *db2profile* を追加する
- .login file にソース *db2cshrc* を追加する

DB2 環境を自動的に設定する方法に関する詳細については、「概説およびインストール」をご覧ください。

---

**DBI1769W** DB2 プロファイル項目を *inst-name* の **.profile** あるいは **.login** ファイルから除去することができません。

**説明:** DB2 は、このユーザーの .profile ファイルまたは .login ファイルにある DB2 環境設定行を注釈にすることができません。次回このユーザーでログインする時に、ファイル *db2profile* (あるいは *db2cshrc*) が見つからないというメッセージを受け取る可能性があります。これは、以下のいずれかが原因となっています。

- 2 ファイル (.profile と .login) が存在していない。
- 2 ファイルに対する書き込み許可がない。

**ユーザーの処置:** *db2profile* あるいは *db2cshrc* の脱落に関するメッセージを受け取った場合、ご使用の .profile あるいは .login ファイルを検索

し、以下の行を注釈にしてください。

```
. sqllib/db2profile .profile ファイルから
source sqllib/db2cshrc .login ファイルから
```

---

**DBI1770E** ファイル `/etc/vfs` を更新することができません。

**説明:** 以下の DB2 Data Links Manager 特定項目が `/etc/vfs` ファイルに必要です。

```
dlfs dlfs_num /sbin/helpers/dlfs_mnthelp none
```

`dlfs_num` は 12 に設定されるか、または 12 がすでに別の `fs` に割り当てられている場合は、8 から 15 までの番号を設定します。この行を `/etc/vfs` ファイルに挿入しようとして、インストール処理が失敗しました。これは、以下のいずれかが原因となっている可能性があります。

- このファイルに対する書き込み許可がない。
- 8 から 15 までの番号がすべて使用されており、`dlfs` に割り当てられる番号がない。

**ユーザーの処置:** 手動で `/etc/vfs` ファイルを編集して、`dlfs` 項目を追加してください。

---

**DBI1771I** ファイル `/etc/vfs` は `/tmp/db2.etcvfs.backup` にコピーされました。

**説明:** DB2 Data Links Manager を作動させるためには、以下の行を `/etc/vfs` ファイルに挿入する必要があります。

```
dlfs dlfs_num /sbin/helpers/dlfs_mnthelp none
```

オリジナルの `/etc/vfs` ファイルのバックアップ・コピーは `/tmp/db2.etcvfs.backup` にあります。

---

**DBI1775W** 既存の `/etc/rc.dlfs` ファイルが変更されました。

**説明:** DB2 が、既存の `/etc/rc.dlfs` ファイルを新しい `dlfs` ファイル・システム・マウント項目で変更しました。

**ユーザーの処置:** ファイル `/etc/rc.dlfs` を調べて、すべての項目が正しいことを確認してください。

---

**DBI1780W** DB2 Data Links Manager は、現行オペレーティング・システムのバージョンではサポートされていません。

**説明:** 現行オペレーティング・システムのバージョンは、DB2 Data Links Manager の実行をサポートしていません。ただし、ご自分の責任において、製品をインストールして構成することはできます。DB2 Data Links Manager をサポートしているオペレーティング・システムを以下にリストします。

- “<OSlist>”

---

**DBI1781E** 使用方法: `-l <language> -t <topic> [-p <path>] [-d]`

有効なトピックには、以下のものがあります。

**core** コア DB2 情報

**admin** 管理情報

**ad** アプリケーション  
開発情報

**wareh** ビジネス・インテリ  
ジェンス情報

**conn** DB2 Connect 情報

**start** 入門情報

**tutr** チュートリアル情報

**opt** オプションのコンポー  
ネント情報

有効な言語には、以下のものがあり  
ます。

---

**DBI1782E** 言語が指定されていません。

**説明:** これらの使用可能言語の中から少なくとも  
1 つの言語を指定する必要があります。

**ユーザーの処置:** 完全リストを表示するには、パ  
ラメーターを指定せずにコマンドを発行してくだ  
さい。

---

**DBI1783E** トピックが指定されていません。

**説明:** これらの使用可能トピックの中から少なく  
とも 1 つのトピックを指定する必要がありま  
す。

**ユーザーの処置:** 完全リストを表示するには、パ  
ラメーターを指定せずにコマンドを発行してくだ  
さい。

---

**DBI1784E** 言語は使用できません。

**説明:** インストールに使用できない言語を指定し  
ました。

**ユーザーの処置:** 完全リストを表示するには、パ  
ラメーターを指定せずにコマンドを発行してくだ  
さい。

---

**DBI1785E** トピック・パッケージは使用できま  
せん。

**説明:** インストールに使用できない文書パッケー  
ジを指定しました。

**ユーザーの処置:** 完全リストを表示するには、パ  
ラメーターを指定せずにコマンドを発行してくだ  
さい。

---

**DBI1790E** **ODSSI** ユーティリティはまだイ  
ンストールされていないため、**DB2**  
検索索引を作成することはできません。

**説明:** ODSSI ユーティリティはまだインスト  
ールされていません。以下のユーティリティ  
(config\_search、config\_view および config\_help)  
は、DB2 オンライン文書の検索索引を作成するた  
めに必要です。

**ユーザーの処置:** ODSSI ユーティリティをイ  
ンストールして、以下のコマンドを実行してくだ  
さい。

```
/opt/IBM/db2/V8.x/doc/db2insthtml lang_locale,
ここで x は 1 または FPn、
ここで n は FixPak 番号
```

このコマンドは、言語ロケールである lang\_locale  
に、DB2 ドキュメンテーションの検索索引を作成  
します。どの言語でドキュメンテーションをイン  
ストールしても、英語のドキュメンテーションも  
インストールされることに注意してください。英  
語ファイルの索引を作成するには、以下のコマ  
ンドを実行してください。

```
/opt/IBM/db2/V8.x/doc/db2insthtml en_US,
ここで x は 1 または FPn、
ここで n は FixPak 番号
```

---

**DBI1791W** **html** ファイルで **DB2** 検索索引を  
作成 / 除去する時に発生する可能  
性があるエラーです。

**説明:** DB2 は、システム・ユーティリティ  
config\_view と config\_help を実行して、  
SCOHELP で使用される DB2 検索索引の作成あ  
るいは除去を行います。ただし、システム上に、  
正しく作成 / 除去されていない DB2 以外の別の  
索引がある場合、エラー・メッセージが戻る可能  
性があります。

**ユーザーの処置:**

- DB2 索引を作成していた場合、SCOHELP を開始して、DB2 項目がトピック・パネルに追加されているかどうか、調べてください。  
"TCP/IP" のようなストリングをテスト検索して、その結果が得られるか調べます。検索が可能であれば、DB2 検索索引は正常に作成されています。
- DB2 索引を除去していた場合、SCOHELP を開始して、DB2 項目がトピック・パネルから除去されているかどうか、確認してください。項目がなくなっていると、DB2 索引は正常に除去されています。
- DB2 索引が正常に作成あるいは除去されているかどうか確認することができない場合、IBM サポートに連絡してください。

---

**DBI1792I DB2 ドキュメンテーション・ファイルの検索索引を作成しています。お待ちください ...**

**説明:** DB2 ドキュメンテーション・ファイルの検索索引を作成しています。選択した言語によって、時間がかかる場合があります。

---

**DBI1793W ユーザー ID *userID* は不明です。**

**説明:** 指定されたユーザーが不明か、または現行システム上で見つかりません。

**ユーザーの処置:** ユーザーがこのシステムに存在することを確認してから再試行してください。

---

**DBI1794E 1 つのグループに入れようとしたアイコンの数が多すぎます。**

**説明:** デスクトップ・マネージャーによって許可された数を超えるアイコンを必須フォルダーに入れようとしたとき。

**ユーザーの処置:** デスクトップから不要なアイコンを除去して、コマンドを再試行してください。

---

**DBI1795E すべての必須アイコンを作成するための空きメモリーが足りません。**

**説明:** アイコンの生成中にアイコン作成ユーティリティーがメモリーを使い果たしました。

**ユーザーの処置:** 不要なプログラムをクローズして、もう一度やり直してください。

---

**DBI1796W 適切な許可なしでディレクトリーにアイコンを作成しようとした。**

**説明:** このユーザー ID は、要求されたユーザーのアイコンを作成するために必要な許可を持っていません。

**ユーザーの処置:** ルート権限を持つユーザー、または、ユーザー・ディレクトリー内にファイルを作成するために必要な許可を持つユーザーとしてログオンして、コマンドを再試行してください。

---

**DBI1797I アイコンが正常に作成されました。**

---

**DBI1900N Windows NT クラスタ・サポートはアクセスできません。**

**説明:** DB2 は Windows NT クラスタにアクセスできません。

**ユーザーの処置:** Windows NT クラスタ・サポートが正しくインストールされていてクラスタ・サービスが開始済みであることを確認してください。

---

**DBI1901N DB2/NT クラスタ・サービスに対するユーザーの権利を更新しようとしてエラーが発生しました。**

**説明:** DB2 サービスで使用するよう指定されたアカウントに対して必要なユーザー権利を与えるようとしたときに DB2 でエラーが起きました。

**ユーザーの処置:** 有効なアカウントが指定されていてユーザーの権利を与えるのに必要な特権があることを確認してください。

---

**DBI1902N DB2/NT クラスター・サービス構成を更新しようとしてエラーが発生しました。**

**説明:** DB2/NT クラスター・サービス構成を更新しようとして DB2 でエラーが起きました。

**ユーザーの処置:** サービス・データベースがロックされていないこととアクセス可能であることを確認してください。また、DB2/NT クラスター・サービスが作成されていることも確認してください。

---

**DBI1903N クラスター・レジストリーのアクセス中にエラーが起きました。**

**説明:** Windows NT クラスター・レジストリーを読み取りあるいは更新しようとして DB2 でエラーが起きました。

**ユーザーの処置:** Windows NT クラスター・サポートが正しくインストールされていてクラスター・サービスが開始済みであることを確認してください。現在のログオン・ユーザー・アカウントにクラスター・レジストリーにアクセスできるだけの権限があることを確認してください。

---

**DBI1904N DB2 インスタンスがクラスターされていません。**

**説明:** DB2 はクラスター操作を行うためにインスタンスにアクセスしようとしたがインスタンスは DB2 クラスター・インスタンスとして識別されませんでした。

**ユーザーの処置:** インスタンスがクラスター・オプションで作成されたか確認してください。インスタンスがドロップされてクラスター・オプションで再作成される可能性があります。ノード・ディレクトリー、データベース・ディレクトリー、およびデータベース・マネージャー構成ファイルは失われます。

---

**DBI1905N DB2 クラスター操作中にシステム・エラーが発生しました。**

**説明:** DB2 がクラスター操作を実行中にシステム・エラーを検出しました。

**ユーザーの処置:** Windows NT オペレーティング・システムが、クラスター・サポートに対して適切なレベルにあることを確認してください。

また、Windows NT クラスター・サポートが正しくインストールされていてクラスター・サービスが開始済みであることを確認してください。

---

**DBI1906N DB2 クラスター・インスタンスにノードを追加しようとしてエラーが発生しました。**

**説明:** DB2 クラスター・インスタンスをサポートするノードを更新しようとしてエラーが起きました。

**ユーザーの処置:** リモート・システムがオンラインで LAN でアクセス可能であることを確認してください。また、リモート・システムでレジストリーの更新を行うのに必要な特権があることも確認してください。

---

**DBI1907N DB2 クラスター・インスタンスからノードを削除しようとしてエラーが発生しました。**

**説明:** DB2 クラスター・インスタンスからノードを削除しようとしてエラーが発生しました。

**ユーザーの処置:** リモート・システムがオンラインで LAN でアクセス可能であることを確認してください。また、リモート・システムでレジストリーの更新を行うのに必要な特権があることも確認してください。

ワークステーション自体からクラスター・サポートをドロップできません。このタスクを実行するためにインスタンスをドロップしてください。



---

**DBI1908N** リモート・システム上にインスタンス・プロファイルを作成しようとしてエラーが発生しました。

**説明:** クラスタ・サポートを使用可能にするために、リモート・システム上にインスタンス・プロファイルを作成しようとしてエラーを検出しました。

**ユーザーの処置:** リモート・システムがオンラインで LAN でアクセス可能であることを確認してください。また、リモート・システムでレジストリーの更新を行うのに必要な特権があることも確認してください。

---

**DBI1909N** リモート・システム上のインスタンス・プロファイルを削除しようとしてエラーが発生しました。

**説明:** クラスタ・サポートを使用不能にするために、リモート・システム上のインスタンス・プロファイルを削除しようとしてエラーを検出しました。

**ユーザーの処置:** リモート・システムがオンラインで LAN でアクセス可能であることを確認してください。また、リモート・システムでレジストリーの更新を行うのに必要な特権があることも確認してください。

---

**DBI1910N** 使用法 : **DB2NCRT**  
**-I:Instance-Name -C:cluster-node**

**説明:** ユーザーが DB2NCRT コマンドに対して誤った引き数を指定しました。

**ユーザーの処置:** クラスタされたインスタンスおよびクラスタ・ノードに対する有効なワークステーション名を指定してコマンドを再発行してください。

---

**DBI1911N** 使用法 : **DB2NLIST**  
**-I:Instance-Name**

**説明:** ユーザーが DB2NLIST コマンドに対して誤った引き数を指定しました。

**ユーザーの処置:** クラスタ化されたインスタンスを指定してコマンドを再発行してください。

---

**DBI1912I** **DB2** クラスタ・コマンドは成功しました。

**説明:** ユーザー要求は正常に処理されました。

**ユーザーの処置:** アクションは必要ありません。

---

**DBI1913N** リモート・マシン・レジストリーへの接続ができません。

**説明:** リモート・マシンの Windows NT レジストリーに接続しようとして DB2 でエラーが起きました。

**ユーザーの処置:** ターゲット・マシンが実行中でコンピューター名が正しいことを確認してください。また、現在のログオン・ユーザー・アカウントにリモート・マシン・レジストリーにアクセスできるだけの権限があることを確認してください。

---

**DBI1914N** クラスタ・レジストリーへの接続ができません。

**説明:** Windows NT クラスタ・レジストリーに接続しようとして DB2 でエラーが起きました。

**ユーザーの処置:** Windows NT クラスタ・サポートが正しくインストールされていてクラスタ・サービスが開始済みであることを確認してください。

---

**DBI1915N ターゲット・マシンは Windows NT クラスタに所属しません。**

**説明:** コマンドは以下の理由のいずれかで失敗しました。

- (1) db2iclus ユーティリティが DB2 MSCS インスタンスからマシンを除去しようとしたがターゲット・マシンが DB2 MSCS インスタンスに追加されていない。
- (2) db2iclus ユーティリティがマシンを DB2 MSCS インスタンスに追加しようとしたがターゲット・マシンに、必要な Windows NT クラスタのサポート・ソフトウェアがインストールされていない。

**ユーザーの処置:** 問題の原因に対応する処置にしたがってください。

- (1) "db2iclus list" コマンドを実行してマシンが DB2 MSCS インスタンスの一部であることを確認する。
- (2) ターゲット・マシンに Windows NT クラスタ・ソフトウェアをインストールしてコマンドをやり直す。

---

**DBI1916N 指定されたインスタンス名が無効です。**

**説明:** インスタンス名パラメーターで指定されたインスタンスがローカル・マシンに存在しません。

**ユーザーの処置:** 有効なインスタンスのリストを検出するには、db2ilist コマンドを実行してください。有効なインスタンス名を使用して、コマンドを再発行してください。

---

**DBI1918N Windows NT レジストリーのアクセス中にエラーが起きました。**

**説明:** ターゲット・マシンの Windows NT レジストリーを読み取りあるいは更新しようとして

DB2 でエラーが起きました。

**ユーザーの処置:** 現在のログオン・ユーザー・アカウントにターゲット・マシンの Windows NT レジストリーにアクセスできるだけの権限があることを確認してください。問題が解決しない場合は IBM サービス技術員に連絡して助言を受けてください。

---

**DBI1919N TCP/IP サービス・ファイルのアクセス中にエラーが起きました。**

**説明:** TCP/IP サービス・ファイルの読み取りあるいは更新中に DB2 でエラーが起きました。

**ユーザーの処置:** システムにサービス・ファイルが存在しておりそのファイルが読み取りおよび書き込み権限にアクセスすることができることを確認してください。また、ファイルの内容が有効で、サービス・ファイルに項目の重複がないことも確認してください。

---

**DBI1920N DB2 インスタンス・プロファイル・パスが無効です。**

**説明:** DB2 インスタンス・プロファイル・パスが存在しないか、書き込みアクセスが現在のログオン・ユーザー・アカウントに与えられていないため、指定した DB2 インスタンス・プロファイル・パスに DB2 はアクセスできません。

**ユーザーの処置:** DB2 インスタンス・プロファイル・ポイントのパス名が有効なディレクトリーとなっていて、現在のログオン・ユーザー・アカウントがそのディレクトリーに対して書き込みアクセスがあることを確認してください。

---

**DBI1921N ユーザー・アカウントあるいはパスワードが無効です。**

**説明:** ユーザー・アカウントあるいはパスワードが無効です。

**ユーザーの処置:** 正しいアカウントでコマンドを再発行してください。



---

**DBI1922N** ホスト名が無効です。

**説明:** TCP/IP ホスト名パラメーターが無効であるか存在しないかのいずれかです。

**ユーザーの処置:** TCP/IP がシステムで操作可能であることを確認してください。ドメイン・ネーム・サーバーを使用している場合、ドメイン・ネーム・サーバーがアクティブであることを確認してください。正しい TCP/IP ホスト名を使用してコマンドを再発行してください。

---

**DBI1923N** TCP/IP ポート範囲パラメーターが指定されていません。

**説明:** パーティション・データベース・インスタンスに対する TCP/IP ポート範囲がサービス・ファイルに追加されていない場合にこのインスタンスを作成するには、db2icrt ユーティリティには TCP/IP ポート範囲の指定が必要です。

**ユーザーの処置:** システムで使用可能な TCP/IP ポートの範囲を選択して db2icrt コマンドに `-r` オプションを付けて再発行してください。

---

**DBI1924N** コンピューター名が無効です。

**説明:** コンピューター名パラメーターが無効です。

**ユーザーの処置:** ターゲット・マシンが実行中でコンピューター名が正しいことを確認してください。また、現在のログオン・ユーザー・アカウントにマシン・レジストリーに接続できるだけの権限があることを確認してください。

---

**DBI1925N** DB2 サービスの状況を照会できません。

**説明:** DB2 サービスの状況の照会が DB2 ではできません。

**ユーザーの処置:** DB2 サービスがターゲット・マシンに存在しているか確認してください。現在のログオン・ユーザー・アカウントにターゲット・マシンのサービス状況を照会できるだけの権

限があることを確認してください。

---

**DBI1926N** パスワードが期限切れです。

**説明:** パスワードの期限が切れています。

**ユーザーの処置:** パスワードを変更した後で、新しいパスワードを使用して要求を再試行してください。

---

**DBI1927N** 使用法 :

```
db2iclus { ADD /u:Username,
Password
 [/m:Machine name] |
 DROP [/m:Machine name] |
 MIGRATE /p:Instance profile
 path | UNMIGRATE }
 [/i:instance name]
 [/DAS DASname]
 [/c:Cluster name]
```

**説明:** このコマンドの有効な引き数は以下のとおりです。

**ADD** DB2 MSCS インスタンスに MSCS ノードを追加する

**DROP** DB2 MSCS インスタンスから MSCS ノードを除去する

**MIGRATE**  
MSCS 以外のインスタンスを MSCS インスタンスに移行する

**UNMIGRATE**  
MSCS 移行を取り消す

このコマンドの有効なオプションは以下のとおりです。

**/DAS** DAS インスタンス名を指定する。このオプションは、DB2 Administration Server に対してクラスター操作を実行するときに必要です。

**/c** デフォルト / 現行クラスターと異なる場合 MSCS クラスター名を指定する

**/p** インスタンス・プロファイル・パスを指定する。このオプションは、MSCS 以外のインスタンスを MSCS インスタンスに移行するときに必要です。

**/u** DB2 サービスのアカウント名およびパスワードを指定する。このオプションは別の MSCS ノードを DB2 MSCS パーティション・データベース・インスタンスに追加する時に必要です。

**/m** MSCS ノードを追加または除去するためのリモート・コンピューター名を指定する

**/i** デフォルト / 現行インスタンス名と異なる場合インスタンス名を指定する

**ユーザーの処置:** 有効なパラメーターを使用してコマンドを再入力してください。

---

#### **DBI1928N ユーザー・アカウントが指定されていません。**

**説明:** パーティション・データベース・インスタンスの作成あるいはノードの追加を行う時には、ユーザー・アカウント・パラメーターを指定する必要があります。Windows NT 環境では、インスタンス・ディレクトリーが置かれたネットワーク共有にアクセスするには、有効な Windows NT アカウントのもとでデータベース・マネージャー・パーティション (あるいは MPP ノード) を実行する必要があります。

**ユーザーの処置:** コマンドに `-u` オプションを指定して、ユーザー・アカウント名とパスワードを指定して、再発行してください。

---

#### **DBI1929N インスタンスは MSCS サポートですでに構成されています。**

**説明:** インスタンスは MSCS サポートですでに構成されているため MSCS インスタンスへのインスタンスの移行が失敗しました。

**ユーザーの処置:** アクションは必要ありません。

---

#### **DBI1930N ターゲット・マシンはすでに DB2 MSCS インスタンスの一部です。**

**説明:** ターゲット・マシンがすでに DB2 MSCS インスタンスの一部であるため MSCS ノードの DB2 MSCS インスタンスへの追加が失敗しました。

**ユーザーの処置:** 処置は必要ありません。

---

#### **DBI1931N データベース・パーティション・サーバー (ノード) がアクティブです。**

**説明:** データベース・パーティション・サーバーがアクティブのためドロップあるいは修正はできません。

**ユーザーの処置:** データベース・パーティション・サーバーがドロップあるいは変更の前に停止していることを確認してください。データベース・パーティション・サーバーを停止するには、以下のように `db2stop` コマンドを使用します。

```
db2stop nodenum <node-number>
```

---

#### **DBI1932N リモート・マシン上に同一インスタンスが存在しているためリモート・マシン上にデータベース・パーティション・サーバーを追加できません。**

**説明:** リモート・マシンにインスタンスが存在しているため `db2nrcr` コマンドはリモート・マシン上に新規のデータベース・パーティション・サーバーを追加できません。

**ユーザーの処置:** リモート・マシン上のインスタンスが使用できない場合リモート・マシン上で `db2idrop` コマンドを実行してインスタンスを除去してください。

---

**DBI1933N** システムにデータベースが存在しているためデータベース・パーティション・サーバーは新規のマシンに移動されません。

**説明:** db2nchg コマンドが /m:machine オプションを指定して発行されるとこのコマンドはデータベース・パーティション・サーバーを新規のマシンに移動します。システムにデータベースが存在している場合には db2nchg は失敗します。

**ユーザーの処置:** データベースが存在する時に新規マシンにデータベース・パーティション・サーバーを移動するには、db2start restart コマンドを使用してください。db2start コマンドの詳細については、『DB2 コマンド・リファレンス』を参照してください。

---

**DBI1934N** 使用法 :

**db2iupdt InstName**  
    /u:username,password  
    [/p:instance profile path]  
    [/r:baseport,endport]  
    [/h:hostname]

**説明:** 無効な引き数が db2iupdt コマンドに対して入力されました。このコマンドの有効な引き数は以下のとおりです。

**InstName**

インスタンスの名前

**/u** DB2 サービスのアカウント名およびパスワードを指定する。このオプションはパーティション・データベース・インスタンスを作成する時に必要です。

**/p** 更新されたインスタンスに対して新規のインスタンス・プロファイル・パスを指定する

**/r** MPP モードで実行する時にパーティション・データベース・インスタンスで使用される TCP/IP ポートの範囲を指定する。ローカル・マシンのサービス・フ

ァイルはこのオプションが指定された場合に、以下の項目で更新されます。

DB2\_InstName baseport/tcp  
DB2\_InstName\_END endport/tcp

**/h** 現行マシンに TCP/IP ホスト名が複数ある場合、デフォルトの TCP/IP ホスト名を上書きする。

**ユーザーの処置:** 有効な引き数を使用してコマンドを再入力してください。

---

**DBI1935N** db2iupdt コマンドがインスタンス・ディレクトリーの更新に失敗しました。

**説明:** 以下の理由のいずれかで db2iupdt コマンドがインスタンス・ディレクトリーの更新に失敗しました。

- (1) db2audit.cfg ファイルが ¥SQLLIB¥CFG ディレクトリーから欠落している。
- (2) db2iupdt コマンドにインスタンス・ディレクトリー内にファイルあるいはサブディレクトリーを作成するのに必要な権限がない。

**ユーザーの処置:** db2audit.cfg ファイルが ¥SQLLIB¥CFG ディレクトリーに存在し現行のログオン・アカウントにインスタンス・ディレクトリーにファイルおよびディレクトリーを作成するだけの権限があることを確認してください。インスタンス・ディレクトリーは ¥SQLLIB¥InstName にあります (ここで InstName はインスタンスの名前です)。

---

**DBI1936N** db2iupdt コマンドはデータベース・マネージャー構成ファイルの更新に失敗しました。

**説明:** 前のデータベース・マネージャー構成ファイルが、壊れているか欠落しているため、db2iupdt コマンドはデータベース・マネージャー構成ファイルの更新ができませんでした。現在の

インスタンスは不整合状態で使用できません。

**ユーザーの処置:** デフォルトのデータベース・マネージャー構成ファイルをインスタンス・ディレクトリにコピーしてコマンドを再実行してください。インスタンス・ディレクトリは `¥SQLLIB¥InstName` にあります。デフォルトのデータベース・マネージャー構成ファイルは `db2system` という名前で `¥sqllib¥cfg` ディレクトリにあります。

---

**DBI1937W db2ncrt** コマンドは正常にノードを追加しました。このノードは、すべてのノードを再び停止および開始するまでアクティブになりません。

**説明:** すべてのノードが `STOP DATABASE MANAGER (db2stop)` コマンドで同時に停止されない限り、`db2nodes.cfg` ファイルが更新されて新規ノードが組み込まれることはありません。ファイルが更新されない限り、既存のノードは新規ノードと通信できません。

**ユーザーの処置:** `db2stop` を発行してすべてのノードを停止してください。すべてのノードが正常に停止したら、`db2start` を発行して、新規のノードを含め、すべてのノードを開始してください。

---

**DBI1940N DB2 Administration Server はアクティブです。**

**説明:** DB2 Administration Server はアクティブのため変更できません。

**ユーザーの処置:** DB2 Administration Server を変更する前に、これが停止されていることを確認してください。DB2 Administration Server を停止するには、以下のように `db2admin` コマンドを使用してください。

```
db2admin stop
```

---

**DBI1941W DB2 Administration Server が存在しません。**

**説明:** DB2 Administration Server がこのマシンで見つかりませんでした。

**ユーザーの処置:** マシン上に DB2 Administration Server を作成してください。

---

**DBI1942N DB2 Administration Server がすでにターゲット・マシンに存在しています。**

**説明:** DB2 Administration Server がすでにターゲット・マシンに存在しているため、操作を完了できませんでした。

**ユーザーの処置:** DB2 Administration Server をターゲット・マシンから除去し、操作を再試行してください。

---

**DBI1943N DB2 Administration Server はクラスタ化されていません。**

**説明:** DB2 が DB2 Administration Server にアクセスしてクラスタ操作を実行しようとしたが、DB2 Administration server が MSCS サポートで構成されていません。

**ユーザーの処置:** DB2 Administration Server が MSCS サポートで構成されていることを確認してください。

---

**DBI1944N DB2 Administration Server は MSCS サポートですすでに構成されています。**

**説明:** DB2 Administration Server を移行しようとしたが、DB2 Administration Server が MSCS サポートですすでに構成されているため失敗しました。

**ユーザーの処置:** アクションは必要ありません。

---

**DBI1950W** インスタンス *instance-name* はすでにインスタンス・リストにありません。

**説明:** 作成または移行中のインスタンスが、すでにインスタンス・リストにあります。

**ユーザーの処置:** db2ilist コマンドによって報告されたインスタンスのリストが正しいかを調べてください。

---

**DBI1951W** インスタンス *instance-name* がインスタンス・リストに見つかりませんでした。

**説明:** 示されたインスタンスが、インスタンス・リストに見つかりませんでした。

**ユーザーの処置:** db2ilist コマンドによって報告されたインスタンスのリストが正しいかを調べてください。

---

**DBI1952E** インスタンス名 *instance-name* が無効です。

**説明:** インスタンス名が無効です。以下のものは使えません。

1. 8 文字より長い名前。
2. “sql”、“ibm” または “sys” で始まる名前。
3. 数字で始まる名前、または a ~ z、\$, #、@、\_、0 ~ 9 以外の文字を使用した名前。

**ユーザーの処置:** 有効なインスタンス名を指定して、コマンドを再発行してください。

---

**DBI1953E** インスタンスがアクティブ中です。

**説明:** アクティブのインスタンスをドロップしようとした。

**ユーザーの処置:** db2stop を発行 (そのインスタンスのユーザー ID から) して、インスタンスを停止する前に、インスタンスを使用しているアプ

リケーションがすべて終了しているかを確認してください。

---

**DBI1954E** インスタンス名は **DB2 Administration Server** で使用中です。

**説明:** インスタンスを DB2 Administration Server で使用中のため作成できません。

**ユーザーの処置:** 別のインスタンス名を指定して、コマンドを再試行してください。

---

**DBI1955E** 使用法 : **db2idrop [-h] [-f] InstName**

**説明:** 間違った引き数が db2idrop コマンドに対して入力されました。このコマンドの有効な引き数は以下のとおりです。

- h 使用法情報を表示する
- f 強制アプリケーション・フラグこのフラグが指定された場合、インスタンスを使用しているすべてのアプリケーションは強制的に終了します。

**InstName**  
インスタンスの名前

**ユーザーの処置:** 以下のようにコマンドを入力し直してください。

```
db2idrop [-h] [-f] InstName
```

---

**DBI1956E** 使用法: **db2ilist [-w 32|64] [-p] [-a] [inst\_name]**

**説明:** 間違った引き数が db2ilist コマンドに入力されました。このコマンドの有効な引き数は以下のとおりです。

- h 使用法情報を表示する
- w 32|64 32 または 64 ビット・インスタンスを

リストする。 -w オプションは -p オプションと共に使用でき、 -a オプションに取り替えられます。

- p** インスタンスが実行される DB2 インストール・パスをリストする。 -p オプションは -a オプションと共に使用でき、 -a オプションに取り替えられます。
- a** インスタンスと関連した DB2 インストール・パス、およびそのビット幅情報 (32 か 64) 情報を含む、すべての関連情報を戻します。

#### inst\_name

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

**ユーザーの処置:** 以下のようにコマンドを入力し直してください。

```
db2ilist [-w 32|64] [-p] [-a] [inst_name]
```

---

#### DBI1957E db2icrt コマンドの構文に誤りがあります。

**説明:** db2icrt ユーティリティーはデータベース・インスタンスを作成します。

```
db2icrt InstName [-s {eselwselclientstandalone}]
 [-p instance profile path]
 [-u username,password]
 [-h hostname]
 [-c Cluster name]
 [-r baseport,endport]
 [-?]
```

コマンド・オプションは以下のとおりです。

- s** 作成するインスタンスのタイプを指定する
- ese** Enterprise Server Edition
- wse** Workgroup Server Edition

**client** クライアント

#### standalone

ローカル・クライアントを持つ  
データベース・サーバー

- p** インスタンス・プロファイル・パスを指定する。
- u** DB2 サービスのアカウント名およびパスワードを指定する。 このオプションは ese インスタンスを作成する時に必要です。
- h** 現行マシンに TCP/IP ホスト名が複数ある場合、デフォルトの TCP/IP ホスト名を上書きする。 TCP/IP ホスト名はデフォルトのノード (ノード 0) を作成するときに使用されます。
- c** MSCS クラスタ名を指定する。このオプションは MSCS をサポートする DB2 インスタンスを作成するために指定されます。
- r** MPP モードで実行する時にパーティション・データベース・インスタンスで使用される TCP/IP ポートの範囲を指定する。 ローカル・マシンのサービス・ファイルはこのオプションが指定された場合に、以下の項目で更新されます。

|                  |                |
|------------------|----------------|
| DB2_InstName     | baseport/tcp   |
| DB2_InstName_1   | baseport+1/tcp |
| DB2_InstName_2   | baseport+2/tcp |
| DB2_InstName_END | endport/tcp    |
- ?** ヘルプを表示する

**ユーザーの処置:** 有効なパラメーターを使用してコマンドを再入力してください。

---

#### DBI1958N インスタンス・プロファイルがレジストリーに追加できません。

**説明:** レジストリーにインスタンス・プロファイルを追加しようとしてエラーが起きました。 インスタンスは作成されませんでした。



**ユーザーの処置:** IBM サービス技術員に連絡してください。

---

**DBI1959N** インスタンス・ディレクトリーが作成できません。

**説明:** 新規インスタンスに必要なファイルあるいはディレクトリーを作成しようとしてエラーが発生しました。インスタンスは作成されませんでした。

**ユーザーの処置:** インスタンス・ディレクトリーが作成されている位置に書き込みアクセスがあることを確認してください。インスタンス・ディレクトリーは製品がインストールされているパスの下に作成されます。DB2INSTPROF プロファイル変数を使用してインスタンス・ディレクトリーに対する別の位置を指定する場合があります。

---

**DBI1960N** DB2 サービスが作成できません。

**説明:** 以下の理由により、DB2 サービスを登録中にエラーが発生しました。

1. 同じ名前のサービスがすでに存在しているか、または削除としてマークされたけれども、次回システムがリブートされるまでクリーンアップされていません。
2. Windows サービスを作成するための十分なアクセス権がありません。

**ユーザーの処置:**

1. インスタンスの名前が、どの既存の Windows サービスとも一致していないことを確認してください。システムのサービスのリストは、Windows レジストリーの HKEY\_LOCAL\_MACHINE¥System ¥CurrentControlSet¥Services の下にあります。このサービスが削除としてマークされている場合は、マシンをリブートして、システムからサービスを除去する必要があります。
2. ローカル管理者グループに属しているユーザー・アカウントにログオンして、操作を再試行してください。

---

**DBI1961N** 新規インスタンスに対するノード・キーがレジストリーに追加できません。

**説明:** インスタンス・プロファイル・レジストリーで新規のインスタンスに対するノード・キーを追加中にエラーが発生しました。

**ユーザーの処置:** IBM サービス技術員に連絡してください。

---

**DBI1962N** 新規ノードがレジストリーに追加できません。

**説明:** インスタンス・プロファイル・レジストリーに新規のノード・キーを追加中にエラーが発生しました。

**ユーザーの処置:** IBM サービス技術員に連絡してください。

---

**DBI1963N** インスタンスのアカウントに対するユーザーの権利を与えているときにエラーが発生しました。

**説明:** DB2 サービスのアカウントを構成しているときに、アカウントに以下のユーザーの権利を与える必要があります。

1. オペレーティング・システムの一部として機能
2. トークン・オブジェクトの作成
3. 割り当てを増やす
4. サービスとしてログオンする
5. プロセス・レベル・トークンの置き換え

インスタンスのアカウントに対するユーザーの権利を与えているときにエラーが発生しました。

**ユーザーの処置:** マシンが NT ドメインに属している場合、1 次ドメイン・コントローラーがアクティブでネットワークからアクセス可能であることを確認してください。そうでない場合には IBM サービス技術員に連絡してください。

---

**DBI1964N** ログオン・アカウントが DB2 サービスに割り当てられません。

**説明:** DB2 サービスにログオン・アカウントを割り当て中にエラーが発生しました。

**ユーザーの処置:** ログオン・アカウントのユーザー名およびパスワードが有効であることを確認してください。

---

**DBI1965N** ノード *node-number* がノード・リストに見つかりませんでした。

**説明:** 指定されたノードがノード・リストで見つかりませんでした。

**ユーザーの処置:** DB2NLIST コマンドを使用してノードのリストを表示して、ノードが存在することを確認してください。

---

**DBI1966N** ノードがレジストリーから削除できません。

**説明:** インスタンス・プロファイル・レジストリーから指定のノードを削除中にエラーが発生しました。

**ユーザーの処置:** IBM サービス技術員に連絡してください。

---

**DBI1967N** DB2 サービスが登録解除できません。

**説明:** DB2 サービスの登録解除中にエラーが発生しました。

**ユーザーの処置:** NT サービスを削除するのに十分な権限があることを確認してください。

---

**DBI1968N** プロファイル・レジストリーのノード構成を変更中にエラーが発生しました。

**説明:** プロファイル・レジストリーのノード構成を変更中にエラーが発生しました。

**ユーザーの処置:** IBM サービス技術員に連絡してください。

---

**DBI1969N** インスタンス・ディレクトリーで新規ファイルを作成中にエラーが起きました。

**説明:** インスタンス・ディレクトリーで新規ファイルを作成中に内部エラーが起きました。

**ユーザーの処置:** インスタンス・ディレクトリーに書き込みアクセスがあることを確認してください。

---

**DBI1970N** インスタンス・ディレクトリーでファイルからの読み取りまたはファイルへの書き込み中にエラーが発生しました。

**説明:** インスタンス・ディレクトリーでファイルからの読み取りまたはファイルへの書き込み中に内部エラーが発生しました。

**ユーザーの処置:** IBM サービス技術員に連絡してください。

---

**DBI1971N** インスタンス・プロファイルがレジストリーから除去できません。

**説明:** レジストリーから インスタンス・プロファイルを削除中にエラーが発生しました。

**ユーザーの処置:** IBM サービス技術員に連絡してください。

---

**DBI1972N** インスタンス・ディレクトリーが除去できません。

**説明:** 指定されたインスタンスに属する必要なファイルあるいはディレクトリーを削除中にエラーが発生しました。

**ユーザーの処置:** インスタンス・ディレクトリーがある位置への書き込みアクセスがあることを確認してください。



---

**DBI1973N DB2 サービスを開始するための構成が自動的に失敗しました。**

**説明:** DB2 サービスを自動的に開始する設定中にエラーが発生しました。

**ユーザーの処置:** マシンをリブートしてコマンドを再度試行してください。問題が続く場合、IBM サービス担当者に連絡してください。

---

**DBI1974N プロファイル変数 DB2ADMINSERVER がプロファイル・レジストリーに設定できません。**

**説明:** プロファイル・レジストリーにプロファイル変数 DB2ADMINSERVER を設定中にエラーが発生しました。

**ユーザーの処置:** IBM サービス技術員に連絡してください。

---

**DBI1975N 環境変数 DB2ADMINSERVER がプロファイル・レジストリーから除去できません。**

**説明:** プロファイル・レジストリーの環境変数 DB2ADMINSERVER を削除中にエラーが発生しました。

## SQL0200 - SQL0299

---

**SQL0203N 参照する列 *name* が確定できません。**

**説明:** 列 *name* がステートメントで使用されており、その列が参照可能な列が複数あります。これは、以下のいずれかによる可能性があります。

- FROM 文節に指定された、同じ名前の列を持つ 2 つの表
- 選択リストの複数の列に適用される名前を参照する ORDER BY 文節

**ユーザーの処置:** IBM サービス技術員に連絡してください。

---

**DBI1980W UPM でユーザー ID *userID* を作成できません。**

**説明:** 製品のインストール中に提供するユーザー ID が、ユーザー・プロファイル管理 (UPM) で作成されませんでした。このユーザー ID は DB2 Administration Server を開始するために必要です。

**ユーザーの処置:** UPM を使用するユーザー ID とパスワードを手動で作成し、次に DB2 Administration Server を手動で開始します。“db2admin” コマンドを使用するユーザー ID とパスワードの組み合わせを使用するために DB2 Administration Server を設定する必要がある可能性があります。

- それが古いまたは新しい遷移変数を参照する場合は、指示するための相関名を使用しない CREATE TRIGGER ステートメントの対象表の列に対する参照

列名には、どの表の列なのかを明確にするために、より詳細な情報が必要です。

ステートメントは処理できません。

**ユーザーの処置:** 列名に修飾子を追加してください。修飾子は表名または相関名です。選択リストの列の名前の変更が必要になる場合があります。

**sqlcode:** -203

**sqlstate:** 42702

---

**SQL0204N** *name* は未定義の名前です。

**説明:** このエラーは、以下のいずれかが原因です。

- *name* によって示されているオブジェクトが、データベースに定義されていません。
- データ・タイプが使用されています。このエラーは、以下の理由で起きる可能性があります。
  - *name* が修飾されている場合は、この名前のデータ・タイプがデータベースに存在しません。
  - *name* が修飾されていない場合は、ユーザーの関数パスに、必要なデータ・タイプが属しているスキーマが入っていません。
  - パッケージがバインド時間より前の作成タイム・スタンプのデータベースには、データ・タイプはありません (静的ステートメントに該当します)。
  - データ・タイプが CREATE TYPE ステートメントの UNDER 文節にある場合は、タイプ名は定義されたタイプと同じの可能性があります。これは有効ではありません。
- 関数が以下のいずれかで参照されています。
  - DROP FUNCTION ステートメント
  - COMMENT ON FUNCTION ステートメント
  - CREATE FUNCTION ステートメントの SOURCE 文節

*name* が修飾されている場合は、関数が存在しません。 *name* が修飾されていない場合は、この名前の関数が現在の関数パスのいずれのスキーマにも存在しません。関数が COALESCE、NULLIF、あるいは VALUE の組み込み関数に基づくことはできないことに注意してください。

この戻りコードは、すべてのタイプのデータベース・オブジェクトに対して生成されます。

連合システム・ユーザー: *name* によって識別されるオブジェクトがデータベースで定義されていないか、*name* が DROP NICKNAME ステートメントでニックネームになっていません。

一部のデータ・ソースは、*name* に適切な値を提供しません。この場合、メッセージ・トークンは“OBJECT:<data source> TABLE/VIEW”の形式になります。これは、指定されたデータ・ソースの実際の値が不明であることを示します。

ステートメントは処理できません。

**ユーザーの処置:** オブジェクト名 (必須の修飾子を含む) が、SQL ステートメントに正しく指定されており、それが存在することを確認してください。 SOURCE 文節にデータ・タイプまたは関数が抜けている場合は、オブジェクトが存在しない可能性があるか、またはオブジェクトはどこかに存在しているが、スキーマが関数パスに存在しない可能性があります。

連合システム・ユーザー: ステートメントが DROP NICKNAME の場合は、オブジェクトが実際にニックネームであるかどうかを確認してください。オブジェクトは連合データベースまたはデータ・ソースに存在しない可能性があります。連合データベース・オブジェクトとデータ・ソースオブジェクト (存在する場合) の存在を確認してください。

**sqlcode:** -204

**sqlstate:** 42704

---

**SQL0205N** 列または属性 *name* が、*object-name* で定義されていません。

**説明:** *object-name* が表またはビューである場合は、*name* は、*object-name* で定義されていない列です。 *object-name* が構造化タイプである場合は、*name* は、*object-name* で定義されていない属性です。

連合システム・ユーザー: *object-name* はニックネームを参照できます。

ステートメントは処理できません。

**ユーザーの処置:** *object-name* が表またはビューである場合は、列および表またはビュー名 (必須の修飾子を含む) が、SQL ステートメントに正しく指定されていることを確認してください。

*object-name* が構造化タイプの場合は、属性およびタイプ名 (必須の修飾子を含む) が SQL ステートメントに正しく指定されていることを確認してください。

また、REORG または IMPORT 時にこのエラーを受け取った場合は、索引の列名が DB2 管理ガイド で定義されているデータベース・マネージャー命名規則に違反している可能性があります。

**sqlcode:** -205

**sqlstate:** 42703

---

**SQL0206N** 使用されているコンテキストで、*name* は無効です。

**説明:** このエラーは、以下の場合に起きる可能性があります。

- INSERT または UPDATE ステートメントの場合は、指定された列が表の列でないか、あるいは挿入または更新のオブジェクトとして指定されたビューではありません。
- SELECT または DELETE ステートメントの場合は、示された列が、ステートメント内の FROM 文節で識別された表またはビューの列ではありません。
- ORDER BY 文節については、指定した列が、許可されない副選択の関連参照となっています。
- CREATE TRIGGER、CREATE METHOD、または CREATE FUNCTION ステートメントの場合は、以下の通りです。
  - 参照名 *name* は、列名、ローカル変数、または遷移変数には解決されません。

- SIGNAL ステートメントに指定された条件名 *name* は、まだ宣言されていません。
- CREATE TRIGGER ステートメントの場合は、以下の通りです。
  - OLD または NEW 関連名を使用せずに、参照が対象となる表の列に対して行われませんでした。
  - トリガー・アクションの SET 遷移変数ステートメントの割り当ての左側が、新しい遷移変数のみがサポートされる場所で、古い遷移変数を指定しました。
- PREDICATES 文節を指定された CREATE FUNCTION ステートメントの場合は、以下の通りです。
  - SQL 関数の RETURN ステートメントが、パラメーターではない変数、または RETURN ステートメントの有効範囲にない他の変数を参照しています。
  - FILTER USING 文節が、パラメーター名ではない、または WHEN 文節内の式名ではない変数を参照しています。
  - 索引指数規則の検索ターゲットが、作成中の関数のパラメーター名に一致していません。
  - 索引指数規則の検索ターゲットが、EXPRESSION AS 文節内の式名、または作成中の関数のパラメーター名に一致していません。
- CREATE INDEX EXTENSION ステートメントの場合は、RANGE THROUGH 文節または FILTER USING 文節が、文節で使用できるパラメーター名ではない変数を参照しています。

ステートメントは処理できません。

**ユーザーの処置:** 名前が SQL ステートメントに正しく指定されていることを確認してください。SELECT ステートメントの場合は、すべての必須の表が FROM 文節に指定されていることを確認してください。ORDER BY 文節の副選択については関連列参照がないので、注意してください。表に対して関連名を使用している場合は、後続の

参照が、表名ではなく、相関名を使用していることを確認してください。

CREATE TRIGGER ステートメントの場合は、新しい遷移変数のみが SET 遷移変数ステートメントの割り当ての左側に指定されており、対象となる表の列に対する参照に、相関名が指定されていることを確認してください。

**sqlcode:** -206

**sqlstate:** 42703

---

**SQL0207N** セット演算子を使用する SELECT ステートメントの ORDER BY 文節では、列名を使用できません。

**説明:** セット演算子を持つ SELECT ステートメントに、列名を指定する ORDER BY 文節が入っています。この場合、ORDER BY 文節の列のリストは、整数しか持つことができません。

ステートメントは処理できません。

**ユーザーの処置:** ORDER BY 文節の列リストの中には、整数のみを指定してください。

注: このエラーは、バージョン 2 以前の DB2 と DB2 Connect を介してアクセスされるホストにのみ適用されます。

**sqlcode:** -207

**sqlstate:** 42706

---

**SQL0208N** 列 *name* は結果表に含まれないので、ORDER BY 文節は無効です。

**説明:** ORDER BY リスト中に指定されている列 *name* が、SELECT リスト中に指定されておらず、結果表にも無いため、ステートメントは有効ではありません。SELECT ステートメントの全選択が副選択でない場合、結果表の列のみが、その結果の順序付けに使用できます。

ステートメントは処理できません。

**ユーザーの処置:** 示された列を結果表へ追加するか、または ORDER BY 文節から削除して、ステ

ートメントの構文を訂正してください。

**sqlcode:** -208

**sqlstate:** 42707

---

**SQL0212N** *name* は、重複した表指定であるか、あるいはトリガー定義の REFERENCING 文節に重複指定されています。

**説明:** *name* によって指定されている表、ビュー、別名、または相関名が、同一の FROM 文節中にある表、ビュー、別名、または相関名と同じです。

ステートメントが CREATE TRIGGER の場合は、REFERENCING 文節が、対象となる表と同じ名前を指定する場合は、あるいは OLD または NEW 相関名、あるいは NEW\_TABLE または OLD\_TABLE ID の複数に対して、同じ名前を持つ場合があります。

ステートメントは処理できません。

**ユーザーの処置:** SELECT ステートメントの FROM 文節を訂正してください。相関名を表名、ビュー名、別名に関連させて、表名、ビュー名、別名、または相関名が FROM 文節内の他の表名、ビュー名、別名、または相関名と同じにならないようにしてください。

CREATE TRIGGER ステートメントの場合は、REFERENCING 文節内の名前を、重複しないように変更してください。

**sqlcode:** -212

**sqlstate:** 42712

---

**SQL0214N** ORDER BY 文節の位置の式、または *clause-type* 文節の *expression-start-or-order-by-position* で始まる式が無効です。理由コード = *reason-code*

**説明:** 以下のような *reason-code* で示された理由で、*clause-type* 文節の式

*expression-start-or-order-by-position* の最初の部分で定義した式が無効です。

- 1 SELECT ステートメントの全選択は、副選択ではありません。このタイプの SELECT ステートメントについては、ORDER BY 文節に式は許可されません。 *clause-type* が ORDER BY である場合のみ、この理由コードが起きます。
- 2 SELECT 文節に DISTINCT が指定され、SELECT リストの式にちょうど一致する式ではありません。 *clause-type* が ORDER BY である場合のみ、この理由コードが起きます。
- 3 ORDER BY 文節に列関数があるため、グループ化が起きています。 *clause-type* が ORDER BY である場合のみ、この理由コードが起きます。
- 4 GROUP BY 文節の式をスカラー全選択にすることはできません。 *clause-type* が GROUP BY である場合のみ、この理由コードが起きます。
- 5 GROUP BY 文節では、参照解除演算子の左方は可変関数にできません。 *clause-type* が GROUP BY である場合のみ、この理由コードが起きます。

ステートメントは処理できません。

**ユーザーの処置:** 以下のように *reason-code* が指定する理由に基づき、SELECT ステートメントを修正してください。

- 1 ORDER BY 文節から式を除去してください。結果の列を参照する場合は、ソート・キーを単一整数または単一系列名の形式に変更してください。
- 2 SELECT 節から DISTINCT を除去するか、またはソート・キーを単一整数または単一系列名の形式に変更してください。

- 3 GROUP BY 文節を追加するか、または ORDER BY 文節から列関数を除去してください。
- 4 GROUP BY 文節から、あらゆるスカラー全選択を除去してください。スカラー全選択に基づいた結果の列をグループ化させる場合は、ネストした表の式または共通の表の式を使用し、結果の列としての式で結果の表をまず提供してください。
- 5 GROUP BY 文節で、参照解除演算子の左方から任意の可変関数を除去してください。

**sqlcode:** -214

**sqlstate:** 42822

---

**SQL0216N** 述部演算子の両側にあるエレメントの数が一致しません。述部演算子は *predicate-operator* です。

**説明:** 述部には、述部演算子の右または左側 (または両側) に、エレメントのリストが備えられています。エレメント数は両側で同じものでなくてはなりません。これらのエレメントは括弧で囲まれた式のリスト内に、または全選択にリストを選択するエレメントとして表示される可能性があります。

ステートメントは処理できません。

**ユーザーの処置:** 述部演算子で、エレメント数が一致していない述部を訂正してください。

**sqlcode:** -216

**sqlstate:** 428C4

---

**SQL0217W** 情報の解釈要求しか実行されていないので、ステートメントは実行されませんでした。

**説明:** いずれかの解釈特殊レジスターの現在の値が、EXPLAIN に設定されています。この値を使



例えば、動的 SQL ステートメントの準備と解釈を行うことはできますが、動的ステートメントを実行することはできません。

**ユーザーの処置:** この状態になったインターフェースまたはアプリケーションから、適切な SET ステートメントを発行して、EXPLAIN 以外を設定するように、該当する解釈特殊レジスタの値を変更してください。

**sqlcode:** +217

**sqlstate:** 01604

---

### SQL0219N 必須の解釈表 *name* がありません。

**説明:** 解釈機能が呼び出されましたが、必須の解釈表 *name* を見つけることができませんでした。解釈機能を呼び出す前に、解釈表を作成する必要があります。

**ユーザーの処置:** 必須の解釈表を作成してください。 `sqllib` の下の `misc` ディレクトリーの `EXPLAIN.DDL` というファイルで使用可能な解釈表を作成するには、SQL データ定義言語ステートメントが必要です。

**sqlcode:** -219

**sqlstate:** 42704

---

### SQL0220N 解釈表 *name*、列 *name2* は定義が正しくないか、または欠落しています。

**説明:** 解釈機能が呼び出されましたが、解釈表 *name* の定義が予期されたものではありませんでした。以下の原因のため、定義が正しくありません。

- 正しくない列数が定義されています。( *name2* が数値の場合)
- 正しくないデータ・タイプが列に割り当てられています ( *name2* が列名の場合)。

**ユーザーの処置:** 示された解釈表の定義を訂正してください。 `sqllib` の下の `misc` ディレクトリー

の `EXPLAIN.DDL` というファイルで使用可能な解釈表を作成するには、SQL データ定義言語ステートメントが必要です。

**sqlcode:** -220

**sqlstate:** 55002

---

### SQL0222N カーソル *cursor-name* を使用するホールに対して操作が試行されました。

**説明:** `SQLSTATE` が 24510 の場合、エラーが発生しました。 `SENSITIVE STATIC` と定義されているカーソル *cursor-name* を使って位置指定の更新または削除が試みられ、現在行は、削除ホールまたは更新ホールと識別されました。ホールが発生するのは、カーソル *cursor-name* の結果表の現在行に対応するデータベース内の行を `DB2` が更新または削除しようとしたときに、それに対応する行がもう基本表内に存在しない場合です。

`SQLSTATE` が 02502 の場合、これは警告です。カーソル *cursor-name* のフェッチの処理中に削除ホールまたは更新ホールが検出されました。ホールが発生するのは、カーソル *cursor-name* の結果表の現在行に対応する行をデータベースから `DB2` が再フェッチしようとしたときに、それに対応する行がもう基本表内に存在しない場合です。データは戻されません。

削除ホールが起きるのは、基本表内の対応する行が削除されてしまっている場合です。

更新ホールが起きるのは、基本表内の対応行が更新されてしまって、カーソルの `SELECT` ステートメントに指定されている検索条件をその更新後の行がもう満足しなくなった場合です。

ステートメントは処理できません。カーソルはホール上に置かれたままになります。

**ユーザーの処置:** `FETCH` ステートメントを発行し、ホールではない行上にカーソルを位置付けます。

**sqlcode:** -222

sqlstate: 02502,24510

---

**SQL0224N** 結果表がカーソル *cursor-name* を使用した基本表と一致しません。

**説明:** 位置指定 UPDATE または DELETE が、カーソル *cursor-name* (SENSITIVE STATIC として定義) を使用して、結果表の列の値が基本表行の現行値と一致しない行で試行されました。基本表の行が、結果表に取り出されたときから位置指定 UPDATE または DELETE が処理されたときまでに更新されたため、行は一致しません。

ステートメントは処理できません。カーソルの位置は変更されません。

**ユーザーの処置:** 分離レベルを変更して、基本表の行をカーソル操作中に再度更新できないようにするか、あるいは FETCH INSENSITIVE を行うようにアプリケーションを変更して、位置付けされた UPDATE または DELETE を再試行してください。

sqlcode: -224

sqlstate: 24512

---

**SQL0225N** カーソルが **SCROLL** として定義されていないため、カーソル *cursor-name* の **FETCH** ステートメントが無効です。

**説明:** スクロール不能カーソル *cursor-name* に、以下のいずれかのキーワードとともに **FETCH** ステートメントが指定されました。 **PRIOR**、**FIRST**、**LAST**、**BEFORE**、**AFTER**、**CURRENT**、**ABSOLUTE**、または **RELATIVE**。スクロール不能カーソルに指定できるのは **NEXT** だけです。データは取り出されません。

ステートメントは処理できません。カーソルの位置は変更されません。

**ユーザーの処置:** **FETCH** ステートメントの現在の取り出し方向キーワード (**PRIOR** または **FIRST**) を除去し、**NEXT** を指定してください。

また、カーソルの定義をスクロール可能に変更する方法もあります。

sqlcode: -225

sqlstate: 42872

---

**SQL0227N** カーソル *cursor-name* の位置が不明のため、**FETCH NEXT**、**PRIOR**、**CURRENT**、または **RELATIVE** は許可されません (*sqlcode*、*sqlstate*)。

**説明:** *cursor-name* のカーソル位置が不明です。カーソル *cursor-name* の直前の複数行 **FETCH** が、取り出された複数の行の処理中にエラー (*SQLCODE sqlcode*、*SQLSTATE sqlstate*) になっています。要求された行はエラーのためプログラムに戻すことはできず、カーソル位置は不明のままになりました。

標識構造が直前の複数行 **FETCH** で提供されていた場合は、正の *SQLCODE* が返され、取り出された行はすべてアプリケーション・プログラムに返されています。

ステートメントは処理できません。カーソルの位置は変更されません。

**ユーザーの処置:** カーソルをクローズして再オープンし、位置をリセットしてください。スクロール可能カーソルの場合は、**FETCH** ステートメントを変更して、他の取り出しステートメント (**FIRST**、**LAST**、**BEFORE**、**AFTER**、**ABSOLUTE** など) の 1 つを指定し、有効なカーソル位置を設定して、データの行を取り出してください。

sqlcode: -227

sqlstate: 24513

---

**SQL0228N** **FOR UPDATE** 文節が読み取り専用カーソル *cursor-name* に指定されました。

**説明:** カーソル *cursor-name* は **INSENSITIVE SCROLL** として定義されていますが、対応する

SELECT ステートメントに FOR UPDATE 文節が入っています。

ステートメントは処理できません。

**ユーザーの処置:** 読み取り専用カーソルを定義するには、INSENSITIVE を DECLARE CURSOR に指定しますが、カーソルの SELECT ステートメントの一部として FOR UPDATE 文節を指定しないでください。

**sqlcode:** -228

**sqlstate:** 42620

---

**SQL0231W** カーソル *cursor-name* の現在位置が現在行の FETCH には無効です。

**説明:** FETCH CURRENT または FETCH RELATIVE 0 ステートメントがスクロール可能カーソル *cursor-name* に発行されました。カーソルが結果表の行に位置付けられていないため、操作は無効です。FETCH BEFORE または FETCH AFTER ステートメントに続く、あるいは SQLCODE +100 になった FETCH ステートメントに続く現在行の FETCH は許可されません。

ステートメントは処理できません。カーソルの位置は変更されません。

**ユーザーの処置:** 現在行の取り出しを行う前に、カーソルが結果表の行に位置付けられていることを確認してください。

**sqlcode:** +231

**sqlstate:** 02000

---

**SQL0236W** SQLDA が *integer1* SQLVAR 項目しか指定していません。 *integer2* SQLVAR 項目が *integer3* 列に必要です。 SQLVAR 項目が設定されていません。

**説明:** SQLDA の「SQLN」フィールドの値は、少なくとも結果セットの列数と同じ大きさでなければなりません。

データベース・マネージャーは SQLVAR 項目を

設定しません (さらに、SQLDOUBLED フラグは「オフ」(すなわち、スペース文字) に設定されません)。

**ユーザーの処置:** SQLDA の「SQLN」フィールドの値を、メッセージに示されている値まで増やして (SQLDA がその容量をサポートするための十分な大きさになるように)、ステートメントの再実行依頼を行ってください。

**sqlcode:** +236

**sqlstate:** 01005

---

**SQL0237W** SQLDA が *integer1* SQLVAR 項目しか指定していません。記述している列の少なくとも 1 つは異なるタイプであるため、*integer2* SQLVAR 項目を指定する必要があります。2 次 SQLVAR 項目が設定されていません。

**説明:** 結果セットの列の少なくとも 1 つが異なるタイプなので、結果セットの列数と同じ数の SQLVAR 項目について、スペースを 2 回指定する必要があります。データベース・マネージャーは基本 SQLVAR 項目のみを設定します (さらに、SQLDOUBLED フラグはオフ (すなわち、スペース文字) に設定されます)。

**ユーザーの処置:** 結果セットの異なるタイプに関する追加情報が必要でない場合は、アクションを行う必要はありません。異なるタイプの情報が必要な場合は、SQLDA の「SQLN」フィールドの値を、メッセージに示されている値まで増やして (その容量をサポートするのに十分な大きさの SQLDA があることを確認した後)、ステートメントの再実行依頼を行う必要があります。

**sqlcode:** +237

**sqlstate:** 01594



---

**SQL0238W** **SQLDA** が *integer1* **SQLVAR** 項目しか指定していません。記述している列の少なくとも 1 つが **LOB** または構造化タイプであるため、*integer3* 列に *integer2* **SQLVAR** 項目が必要です。**SQLVAR** 項目が設定されていません。

**説明:** 結果セットの列の少なくとも 1 つが **LOB** または構造化タイプなので、結果セットの列数と同じ数の **SQLVAR** 項目について、スペースを 2 回指定する必要があります。結果セットの 1 つ以上の列が異なるタイプである可能性にも注意してください。

データベース・マネージャーは **SQLVAR** 項目を設定しません (さらに、**SQLDOUBLED** フラグはオフ (すなわち、スペース文字) に設定されます)。

**ユーザーの処置:** **SQLDA** の「**SQLN**」フィールドの値を、メッセージに示されている値まで増やして (その容量をサポートするのに十分な大きさの **SQLDA** があることを確認した後で)、ステートメントの再実行依頼を行ってください。

**sqlcode:** +238

**sqlstate:** 01005

---

**SQL0239W** **SQLDA** が *integer1* **SQLVAR** 項目しか指定していません。記述している列の少なくとも 1 つが特殊タイプまたは参照タイプであるため、*integer3* 列に *integer2* **SQLVAR** 項目が必要です。**SQLVAR** 項目が設定されていません。

**説明:** 結果セットの列が特殊タイプまたは参照タイプの場合、結果セットの列数と同じ数の **SQLVAR** 項目について、スペースを 2 回指定する必要があります。

データベース・マネージャーは **SQLVAR** 項目を設定しません (さらに、**SQLDOUBLED** フラグは

オフ (すなわち、スペース文字) に設定されます)。

**ユーザーの処置:** 特殊タイプまたは参照タイプの情報が必要な場合は、**SQLDA** の「**SQLN**」フィールドの値を、メッセージに示されている値まで増やして (その容量をサポートするのに十分な大きさの **SQLDA** があることを確認した後で)、ステートメントの再実行依頼を行う必要があります。結果セットの特殊タイプまたは参照タイプに関する追加情報が必要でない場合は、結果セットの列数に適合するのに十分な **SQLVAR** 項目のみを指定して、ステートメントの再実行依頼を行うことができます。

**sqlcode:** +239

**sqlstate:** 01005

---

**SQL0242N** タイプ *object-type* の *object-name* というオブジェクトが、オブジェクトのリストで複数回指定されました。

**説明:** タイプ *object-type* のオブジェクト名のリストで、*object-name* というオブジェクトが複数回指定されました。ステートメントの操作をオブジェクトで複数回実行することはできません。

**ユーザーの処置:** リスト内の重複したオブジェクトを訂正し、重複するオカレンスを取り除いてください。

**sqlcode:** -242

**sqlstate:** 42713

---

**SQL0243N** **SENSITIVE** カーソル *cursor-name* を指定された **SELECT** ステートメントに定義できません。

**説明:** カーソル *cursor-name* は **SENSITIVE** と定義されていますが、**SELECT** ステートメントの内容では、カーソルの一時結果表が **DB2** で作成されなければならない、このカーソル外部で加えられた変更を必ず表示できるとは限りません。この状況は、照会の内容に応じて結果表が読み取り

専用になったときに起きます。たとえば、照会に結合が入っている場合は、結果表は読み取り専用になります。この場合、カーソルは INSENSITIVE または ASENSITIVE として定義されている必要があります。

ステートメントは処理できません。

**ユーザーの処置:** 結果表が読み取り専用でなくなるように照会の内容を変更するか、またはカーソルのタイプを INSENSITIVE または ASENSITIVE に変更してください。

**sqlcode:** -243

**sqlstate:** 36001

---

**SQL0244N FETCH** で指定された **SENSITIVITY** *sensitivity* がカーソル *cursor-name* では無効です。

**説明:** FETCH で指定されたセンシティブティティー・オプション *sensitivity* がカーソル *cursor-name* で有効になっているセンシティブティティー・オプションと矛盾しています。下のリストに、FETCH で指定可能なオプションが示されています:

|                        |                    |
|------------------------|--------------------|
| <b>DECLARE CURSOR</b>  |                    |
| <b>FETCH Statement</b> |                    |
| INSENSITIVE            | INSENSITIVE        |
| SENSITIVE STATIC       | SENSITIVE          |
|                        | or INSENSITIVE     |
| SENSITIVE DYNAMIC      | SENSITIVE          |
| SENSITIVE              | SENSITIVE          |
| ASENSITIVE             | INSENSITIVE        |
|                        | or SENSITIVE       |
|                        | (depending on the  |
|                        | effective          |
|                        | sensitivity of the |
|                        | cursor)            |

スクロール不能カーソルの場合、センシティブティティー・オプションは指定できません。

ステートメントは処理できません。

**ユーザーの処置:** FETCH で指定されたセンシテ

ィビティブティー・オプションを変更または除去してください。

**sqlcode:** -244

**sqlstate:** 428F4

---

**SQL0257N ロー・デバイス・コンテナ**は、現在このプラットフォームではサポートされていません。

**説明:** 「DEVICE」コンテナの使用が試みられました。これは、現在このプラットフォームではサポートされていません。

**ユーザーの処置:** 代わりに、「FILE」コンテナまたはシステム管理表スペースを使用してください。

**sqlcode:** -257

**sqlstate:** 42994

---

**SQL0258N 表スペースの再平衡が保留**または進行中は、コンテナを追加することはできません。

**説明:** 以下に示す条件の 1 つが成立しています。

1. コンテナ操作に必要な ALTER TABLESPACE ステートメントが、以前にこのアプリケーションか別のアプリケーションによって発行されていますが、まだコミットされていません。
2. コンテナ操作に必要な ALTER TABLESPACE ステートメントが発行およびコミットされ、表スペースは現在、再平衡化中です。

**ユーザーの処置:**

1. 可能であれば、コミットされていない作業単位をロールバックし、単一の ALTER TABLESPACE ステートメントを発行して、すべてのコンテナ操作を実行してください。可能でない場合は、再平衡が完了するまで待った後で、もう一度やり直してください。同

じ ALTER TABLESPACE ステートメントで、表スペースへのページの追加と、表スペースからのページの除去の両方を行うことはできません。

2. 再平衡が完了するまで待った後で、もう一度やり直してください。

**sqlcode:** -258

**sqlstate:** 55041

---

**SQL0259N** 表スペースのコンテナ・マップが複雑過ぎます。

**説明:** マップ構造は、表スペースのアドレス・スペースがさまざまなコンテナにマップされる方法に関するレコードを保持しています。これが複雑になりすぎると、表スペース・ファイルに適合しなくなります。

**ユーザーの処置:** コンテナ間でデータをもっと均等に分散するために、表スペースの再平衡が必要になる場合があります。そうすれば、マッピングが簡潔になる可能性があります。

これがうまくいかない場合は、可能な限り同じサイズのコンテナを作成してください。表スペースをバックアップした後で、データベース管理ユーティリティを使用してコンテナを変更すれば、既存のコンテナ・サイズを変更することができます。その後で、表スペースを新しいコンテナにリストアしてください。

**sqlcode:** -259

**sqlstate:** 54037

---

**SQL0260N** LONG 列、DATALINK 列、または構造化タイプ列であるため、列 *column-name* は区分化キーに属することができません。

**説明:** 区分化キーは LONG タイプ列、DATALINK 列、または構造化タイプ列を持つことができません。

ステートメントは処理できません。

**ユーザーの処置:** 区分化キーに LONG 列、DATALINK 列、または構造化タイプ列を使用しないでください。

表に LONG 列、DATALINK 列、または構造化タイプ列しかない場合、区分化キーに使用できる列をその表に追加するか、または区分化キーなしで単一ノードのノード・グループに表を定義してください。

**sqlcode:** -260

**sqlstate:** 42962

---

**SQL0262N** 区分化キーとして使用できる列が存在しないため、表 *table-name* をデータベース・パーティション・グループ *group-name* に作成できません。

**説明:** 表 *table-name* をデータベース・パーティション・グループに作成できません。この表には、区分化キーとして使用できる少なくとも 1 つの列が存在している必要があります。以下のデータ・タイプの列は、区分化キーで使用できません。

- LONG VARCHAR、LONG VARGRAPHIC、BLOB、CLOB、DBCLOB
- DATALINK
- 構造タイプ
- 上記データ・タイプのいずれかを基にしたユーザー定義データ・タイプ

ステートメントは処理できません。

**ユーザーの処置:** 区分化キーに使用できる 1 つ以上の列を持つ表を作成するか、単一データベース・パーティション・グループにこの表を作成してください。

**sqlcode:** -262

**sqlstate:** 428A2

---

**SQL0263N** *node-number-1* から *node-number-2* までのノード範囲は無効です。2 次ノード番号は最初のノード番号より大きいか、または同等にしてください。

**説明:** 指定されたノード範囲が有効ではありません。

ステートメントは処理できません。

**ユーザーの処置:** このステートメント内のノード範囲を訂正し、要求を再試行してください。

**sqlcode:** -263

**sqlstate:** 428A9

---

**SQL0264N** 表が複数ノードのノード・グループ *name* で定義された表スペースに表が常駐しているため、区分化キーの追加またはドロップはできません。

**説明:** 単一ノードのノード・グループの区分化キーのみを追加またはドロップできます。

ステートメントは処理できません。

**ユーザーの処置:** 以下の 1 つを実行し、要求を再試行してください:

- 区分化キーを使用する同一表を定義してください。
- 単一ノードのノード・グループにこのノード・グループを再分散してください。

**sqlcode:** -264

**sqlstate:** 55037

---

**SQL0265N** ノード *node-number* は重複ノードです。

**説明:** CREATE NODEGROUP ステートメントに対し、ON NODES 文節内で 1 度のみノード表示が可能です。

CREATE TABLESPACE および ALTER TABLESPACE ステートメントに対し、ノードは

1 度のみ、1 つの ON NODES 文節でのみ表示可能です。

ALTER NODEGROUP ステートメントまたは REDISTRIBUTE NODEGROUP コマンドで、以下のいずれかが発生しました。

- ノードが ADD NODES または DROP NODES 文節で複数回表示された。
- ノードが ADD NODES および DROP NODES 文節の両方で複数回表示された。
- 追加されるノードは、すでにノード・グループのメンバーである。

ステートメントは処理できません。

**ユーザーの処置:** ON NODES、ADD NODES、または DROP NODES 文節内のノード名またはノード番号がユニークであることを確認してください。CREATE TABLESPACE および ALTER TABLESPACE ステートメントに対し、ノードが複数の ON NODES 文節で表示されていないかを確認してください。

さらに、ALTER NODEGROUP ステートメントまたは REDISTRIBUTE NODEGROUP コマンドでは、以下のようにしてください。

- ADD NODES および DROP NODES 文節の両方でノードを指定しないでください。
- ノードがすでにノード・グループに定義されている場合は、ADD NODES 文節からノードを除去してください。

**sqlcode:** -265

**sqlstate:** 42728

---

**SQL0266N** ノード *node-number* は定義されていません。

**説明:** このノード *node-number* は以下のいずれかの理由のため有効ではありません。

- ノード番号が有効な範囲である 0 から 999 までの間にない
- ノードがノード構成ファイル内にない

- ノードがノード・グループの一部でないため、要求された操作を処理しない

ステートメントは処理できません。

**ユーザーの処置:** 以下の条件に従います。

- 有効範囲内のノード番号を伴うステートメント、コマンドまたは API を発行する
- システムにノードを追加するプロシージャーを続ける
- ステートメント、コマンドまたは API 内の指定されたノードから、ノードを除去する

**sqlcode:** -266

**sqlstate:** 42729

**SQL0268N ノード・グループを再分散している間は *operation* を実行できません。**

**説明:** 以下のいずれかが起きていると思われるです。

- このノード・グループは再分散されています。現行操作が完了するまで、このノード・グループを変更、ドロップまたは再分散できません。
- 表のノード・グループが再分散されている間は、表の区分化キーをドロップできません。

ステートメントは処理できません。

**ユーザーの処置:** 再分散が完了するまで待ち、要求を再試行してください。

**sqlcode:** -268

**sqlstate:** 55038

**SQL0269N データベースに区分化マップの最大数が含まれています。**

**説明:** データベースに区分化マップの最大数 (32,768) が入っているため、新規ノード・グループの作成、ノード・グループの変更、または既存のノード・グループの再分散ができません。

ステートメントは処理できません。

**ユーザーの処置:** データベースの 1 つ以上のノード・グループをドロップしてください。

注: ノード・グループのドロップは、ノード・グループに常駐する表スペース、表およびビューなどのようなデータベース・オブジェクトをすべてドロップします。

**sqlcode:** -269

**sqlstate:** 54033

**SQL0270N 関数をサポートしていません (理由コード = *reason-code*)。**

**説明:** 以下の理由コードによって示されているような制限に違反しているため、このステートメントを処理できません。

- 1 主キー、各ユニーク制約、および各ユニーク索引には表のパーティション列 (列はほかの順序で表示している可能性があります) がすべて入っているようにしている。
- 2 区分化キー列の値の更新はサポートされていない。
- 3 外部キーが ON DELETE SET NULL で定義されているとき、NULL 可能な区分化キー列をこの外部キーに組み込むことはできません。このような制約を定義した結果、区分化キー列を更新することになるため、これは理由コード 2 の特殊事例です。
- 4 カタログ・パーティション以外の複数パーティション・ノード・グループあるいは単一パーティション・ノード・グループを使用している定義済み表は DATA CAPTURE CHANGES をサポートしません。
- 5 WITH CHECK OPTION 文節で作成されたビューは、以下の場合関数 (または関数を使用する参照ビュー) を使用できません。
  - 非決定的の場合

- 副次作用がある場合
- データの配置に関連する場合 (たとえば、ノード番号またはパーティション関数)。

これらの関数は、新しいビューが **CASCADED** チェック・オプションで作成された場合に参照ビュー内にあってはなりません。

- 6** ユーザー定義の特殊タイプに対してトランスフォームは定義できません。
- 7** 長いフィールドは、4K のページ・サイズの表スペースを使用するときのみに、定義できます。 **LARGE TABLESPACE** は、4K のページ・サイズを使用するのみ作成できます。
- 8** 構造化タイプは、DB2 バージョン 7.1 以前の表または構造化タイプ属性データ・タイプの列としてサポートされていません。
- 9** トリガーは型付き表ではサポートされません。
- 10** 単一のデフォルト表スペースは、4K ページ・サイズを指定した表スペースに位置する必要がある **LOB** 列が表にあるため、および表の行サイズあるいは列数に、8K ページ・サイズを指定した表スペースが必要であるため、選択することができません。
- 11** 型付き表あるいはビューは、属性を持たない構造化タイプを使用して作成することができません。
- 12** ソース・キー・パラメーターのタイプは、ユーザー定義の構造化タイプか、または **LOB**、**DATALINK**、**LONG VARCHAR**、あるいは **LONG VARGRAPHIC** から生成された特殊タイプである必要があります。
- 13** チェック制約は型付き表で定義すること

ができないか、または **WITH CHECK OPTION** 文節が型付きビューで指定することができません。

- 14** 参照制約は型付き表に定義することができないか、または型付き表である親表に定義することができません。
- 15** デフォルト値は参照タイプ列で定義されません。
- 16** 参照データ・タイプあるいは構造化データ・タイプを、DB2 バージョン 7.1 以前のパラメーター・データ・タイプまたはユーザー定義関数の戻りデータ・タイプとして使用することはできません。また、有効範囲参照データ・タイプを、ルーチンのパラメーター・データ・タイプまたは戻りタイプとして使用することはできません。構造化タイプを表または行関数の戻り列として使用することはできません。
- 17** **SET CONSTRAINTS** ステートメントは型付き表に対して使用することができません。
- 18** 列レベル **UPDATE** と **REFERENCES** 特権は、型付き表またはビューでは付与されません。
- 19** 特定のデフォルト値は、型付き表の列に対するデフォルト定義の時に指定する必要があります。
- 20** **ALTER TABLE** はマテリアライズ照会表ではサポートされていません。
- 21** マテリアライズ照会表の基本表である表で列の長さを変更することはできません。
- 22** マテリアライズ照会表は、**CREATE SCHEMA** ステートメントに定義できません。
- 23** **REPLICATED** は **REFRESH DEFERRED** で定義されたマテリアライズ照会表にのみ指定できます。



- 24 BEFORE トリガーで起動されたアクションは、REFRESH IMMEDIATE で定義されたマテリアライズ照会表を参照できません。
- 25 SET CONSTRAINTS ステートメントには、1 つのマテリアライズ照会表しか指定できません。
- 26 再分散されているノード・グループに、複製されたマテリアライズ照会表が少なくとも 1 つ含まれています。
- 27 複製されたマテリアライズ照会表は、この複製された表を構成する 1 つ以上の列にユニーク索引が存在しない表には定義できません。
- 28 型付き表またはマテリアライズ照会表は名前変更できません。
- 29 FOR EXCEPTION 文節は、SET CONSTRAINTS ステートメントのマテリアライズ照会表には指定できません。
- 30 型付き表および型付きビューは CREATE SCHEMA ステートメントで定義できません。
- 31 区分化キーは 500 を超える列では定義できません。
- 32 カタログ・パーティション以外の複数パーティション・ノード・グループまたは単一パーティション・ノード・グループを使用して定義された表は、FILE LINK CONTROL で定義された DATALINK 列をサポートしません。
- 33 REFRESH IMMEDIATE で定義されたマテリアライズ照会表の基本表を、カスケード効果を持つ (つまり、ON DELETE CASCADE または ON DELETE SET NULL オプションを持つ) 参照制約の子にすることはできません。
- 34 基礎オブジェクトの関連機能は、現在のリリースではサポートされていません。
- 35 シーケンス列または ID 列をマルチノード・データベース環境で作成することはできません。
- 36 シーケンス列または ID 列が存在する場合、マルチノード・データベースのデータベース活動化は許可されていません。
- 38 索引拡張子を使用する索引は、複数パーティション・ノード・グループではサポートされていません。
- 39 ニックネームまたは OLE DB 表関数は、SQL 関数または SQL 方式の本体で直接的に、または間接的に参照できません。
- 40 IDENTITY\_VAL\_LOCAL 関数は、トリガーまたは SQL 関数で 使用することはできません。
- 41 SQL 変数ステートメントは、ローカル変数と遷移変数の両方に割り当てられません。これはサポートされていません。
- 42 マルチノード・データベース内での、SQL 制御ステートメントを使ったトリガー、方式、または関数の実行、および動的複合ステートメントの実行は許可されません。
- 43 要求した 1 つ以上のオプションが、現在サポートされていません。
- 44 以下の EXPLAIN MODES は、MPP、SMP、および Data Joiner ではサポートされていません。
- COUNT CARDINALITIES
  - COMPARE CARDINALITIES
  - ESTIMATE CARDINALITIES
- 45 多次元クラスタリング (MDC) 表では、APPEND モードはサポートされていません。
- 46 多次元クラスタリング (MDC) 表では、INPLACE 再編成はサポートされていません。

- 47 多次元クラスタリング (MDC) 表では、索引の拡張はサポートされていません。
- 48 多次元クラスタリング (MDC) 表の次元仕様に対する変更は、サポートされていません。
- 49 多次元クラスタリング (MDC) 表では、クラスター索引はサポートされていません。
- 50 多次元クラスタリング (MDC) 表では、ユーザー定義一時表は、サポートできません。
- 51 非カタログ・データベース・パーティションからの、LANGUAGE SQL を指定した CREATE PROCEDURE は、サポートされていません。
- 52 生成された列の式は、この列が ORGANIZE BY 文節で使用されている場合、変更できません。
- 53 LONG VARCHAR、LONG VARCHAR、LONG VARGRAPHIC、DATALINK、LOB タイプ、このいずれかのタイプの特殊タイプ、または構造化タイプは、スクロール可能カーソルの選択リストには指定できません。
- 54 指定されたカタログ表では、INPLACE 表の再編成はサポートされていません。
- 55 連合データベース・サポートとコンセントレーター機能を同時にアクティブにすることはできません。
- 56 オンライン索引の再編成は、ALLOW WRITE モードの地理情報索引ではサポートされていません。
- 57 オンライン索引の再編成は、ALLOW WRITE モードの多次元クラスタリング (MDC) 表ではサポートされていません。
- 58 XML データ・タイプは一時データ・タイプとしてしか使用できず、データベースに保管したり、アプリケーションに戻すことはできません。
- 59 SQL ステートメントを含む関数または方式は、並列環境では使用できません。
- 60 型付き表に依存する、タイプ付きまたはタイプなしのタイプ VIEW があるため、ALTER TABLE ALTER COLUMN SET INLINE LENGTH ステートメントは許可されません。
- 61 テキスト検索関数をチェック制約または生成された列の式に使用することはできません。
- 62 WITH CHECK OPTION 文節を、直接テキスト検索関数を参照しているビューまたはテキスト検索関数を参照している他のビューに依存するビューと一緒に使用することはできません。
- 63 LOB タイプ、LOB タイプ上の特殊タイプ、または構造化タイプの列は、スクロール可能カーソルの選択リストには指定できません。
- 64 このプラットフォームでは、連合プロセッシングはサポートされていません。
- 65 ニックネームのローカル・タイプを現行タイプから指定タイプに変更することはできません。
- 66 組み込まれたトランスフォーム・グループ SYSSTRUCT はサポートされていません。
- ユーザーの処置:** 理由コードに対応するアクションは、以下のとおりです。
- 1 CREATE TABLE、ALTER TABLE または CREATE UNIQUE INDEX ステートメントを訂正する。
- 2 複数パーティションの表の区分化キー列を更新しない、または、パーティション列の新規値を使用して行を削除した後挿入しようとしなない。



- 3 区分化キー列を NULL 使用不能にするか、別の ON DELETE アクションを指定するか、または表の区分化キーを変更して、外部キーに区分化キーの列が組み込まれないようにしてください。
- 4 DATA CAPTURE NONE を指定するかあるいは表がカタログ・パーティションを指定する単一パーティション・ノード・グループの表スペースにあるかどうか確認してください。
- 5 WITH CHECK OPTION 文節を使用したり、ビュー定義から関数またはビューを除去しないでください。
- 6 トランスフォームはユーザー定義の特殊タイプに対して自動的に行われます。CREATE TRANSFORM ステートメントはユーザー定義の構造化タイプのみを使用してください。
- 7 長いフィールドの入った任意の表に対して、4K ページ・サイズの表スペースを使用してください。DMS 表スペースを使用する場合は、長いフィールドを、4K ページ・サイズの表スペース、または別のページ・サイズの表、表スペースか索引データで位置づけられます。LARGE TABLESPACE を定義する場合は、PAGESIZE 4K を使用してください。
- 8 DB2 バージョン 7.1 以前のサーバーであれば、CREATE TABLE ステートメントまたは ALTER TYPE ADD COLUMN ステートメントに構造化タイプの列データ・タイプがないことを確認してください。CREATE TYPE ステートメントまたは ALTER TYPE ADD ATTRIBUTE ステートメントでのどの属性データ・タイプも構造化タイプでないことを確認してください。
- 9 型付き表ではトリガーを定義しないでください。
- 10 表の行サイズあるいは行数を減らすか、
- またはロング・データが 4K ページ・サイズを指定した表スペースにあり、基本データが 8K ページ・サイズを指定した表スペースにあるような 2 つの表スペースを指定してください。
- 11 型付き表あるいはビューを作成する場合は、最低でも 1 つの属性を定義している構造化タイプを指定します。
- 12 ソース・キー・パラメーターのタイプである場合、ユーザー定義の構造化タイプか、または LOB、DATALINK、LONG VARCHAR、あるいは LONG VARCHARGRAPHIC から生成された特殊タイプだけを使用してください。
- 13 型付き表の CREATE TABLE または ALTER TABLE ステートメントでは、チェック制約を指定しないようにします。型付きビューの CREATE VIEW ステートメントでは、WITH CHECK OPTION 文節を指定しないでください。
- 14 CREATE TABLE または ALTER TABLE ステートメントでは、型付き表をチェック制約を指定しないようにします。
- 15 CREATE TABLE または ALTER TABLE ステートメントで参照データ・タイプを指定した列に対して、DEFAULT 文節を指定しないようにします。
- 16 DB2 バージョン 7.1 以前のサーバーであれば、ユーザー定義関数の作成時に構造化タイプ・パラメーターまたは戻りタイプを指定しないでください。また、有効範囲内参照タイプをパラメーターまたは戻りタイプとして指定しないでください。構造化タイプを表または行関数の戻り列として指定しないでください。
- 17 SET CONSTRAINTS ステートメントにある型付き表を指定しないようにします。

- 18 型付き表またはビューに対して REFERENCES または UPDATE 特権を付与するときに、特定の列名を組み込まないようにします。
- 19 型付き表の列で DEFAULT 文節を指定する時に、特定の値を組み込むようにします。
- 20 マテリアライズ照会表をドロップし、任意の属性を指定して再作成します。
- 21 マテリアライズ照会表をドロップし、基本表の列の長さを変更してマテリアライズ照会表を再作成します。
- 22 CREATE SCHEMA ステートメントの外側で CREATE SUMMARY TABLE ステートメントを発行してください。
- 23 REPLICATED 指定を除去するか、またはマテリアライズ照会表の定義に REFRESH DEFERRED が指定されていることを確認します。
- 24 BEFORE トリガーのトリガー・アクションにあるマテリアライズ照会表への参照を除去します。
- 25 各マテリアライズ照会表ごとに、別々の SET CONSTRAINTS IMMEDIATE CHECKED ステートメントを発行します。
- 26 ノード・グループにある複製されたマテリアライズ照会表をすべてドロップし、REDISTRIBUTE NODEGROUP コマンドを再発行します。複製されたマテリアライズ照会表を再作成します。
- 27 マテリアライズ照会表に定義された列のサブセットが、基本表のユニーク索引を構成する列のセットでもあることを確認します。
- 28 型付き表あるいはマテリアライズ照会表名を変更するには、その表をドロップして、新しい名前で作成し直すしか方法がありません。表をドロップした場合、
- その表に依存する他のオブジェクトにも影響する可能性があり、その表に対する特権は失われます。
- 29 SET CONSTRAINTS ステートメントから FOR EXCEPTION 文節を除去します。
- 30 CREATE SCHEMA ステートメントの外部で、型付きビューまたは型付き表の CREATE ステートメントを発行します。
- 31 区分化キー内の列の数を減らします。
- 32 DATALINK 列に NO LINK CONTROL を指定するか、またはカタログ・パーティションを指定する単一パーティション・ノード・グループの表スペースに表が置かれていることを確認します。複数パーティション・ノード・グループに再配布する場合は、再配布を継続するために表をドロップする必要があります。
- 33
- REFRESH IMMEDIATE を使って子として定義されているマテリアライズ照会表の基本表に、カスケード効果を持つ (つまり、ON DELETE CASCADE または ON DELETE SET NULL オプションを指定した) 参照制約を定義しないでください。または
  - カスケード効果を持つ (つまり、ON DELETE CASCADE または ON DELETE SET NULL オプションを指定した) 参照制約の子である基本表を持つ REFRESH IMMEDIATE マテリアライズ照会表を定義しないでください。
- 34 エラーはサポートされていないオブジェクトの関連機能の使用を除去することによって訂正することができます。
- 35 属性 “GENERATED [ALWAYS | BY

- DEFAULT] AS IDENTITY ...”を列から除去するか、またはシーケンスを作成しないでください。
- 36** 新規ノードをドロップして、単一ノード構成に戻してください。ノードがさらに必要であれば、新規ノードを追加する前に、ID 列のあるシーケンスまたは表をドロップする必要があります。
- 38** 索引拡張子を使用する索引を、複数パーティション・ノード・グループの表に作成することはできません。索引拡張子を使用する索引がノード・グループ内の表に存在している間、そのノード・グループを複数パーティション・ノード・グループにすることはできません。このような索引をドロップしてパーティションをノード・グループに追加する（この場合、索引を再作成できません）か、またはノード・グループを未変更のままにしてください。
- 39** ニックネームまたは OLE DB 表関数への参照を除去するか、または間接的にこれらのいずれかを参照しているオブジェクトへの参照を除去してください。
- 40** IDENTITY\_VAL\_LOCAL 関数の呼び出しをトリガー定義または SQL 関数定義から除去してください。
- 41** 割り当てを 2 つの別々のステートメントに分割します。一方のステートメントは SQL 変数にのみ値を割り当て、もう一方のステートメントは遷移変数にのみ値を割り当てなければなりません。
- 42** 新規ノードをドロップして、単一ノード構成に戻してください。ノードがさらに必要であれば、制御ステートメントをもっているトリガー、関数、または方式をドロップする必要があります。
- 43** runstats コマンドを再発行し、サポートされていないオプションを設定してください。
- 44** これらの EXPLAIN モードを SMP、MPP、および Data Joiner で使用することはできません。可能であれば、順次モードで照会を実行してみてください。あるいは、EXPLAIN モードを YES または EXPLAIN に設定すると、実際のカーディナリティー以外の情報については、同じ情報が提供されます。
- 45** 多次元クラスタリング (MDC) 表に対する ALTER TABLE ステートメントに APPEND 文節を指定しないでください。
- 46** INPLACE オプションを指定せずに、REORG コマンドを再発行してください。
- 47** 多次元クラスタリング (MDC) 表に対して、CREATE INDEX ステートメントに EXTENSION 文節を指定しないでください。
- 48** 多次元クラスタリング (MDC) 表をドロップし、変更した次元指定を使って表を再作成してください。
- 49** 多次元クラスタリング (MDC) 表に対して、CREATE INDEX ステートメントに CLUSTER 文節を指定しないでください。
- 50** 宣言済みグローバル一時表に対し、CREATE TABLE ステートメントに ORGANIZE BY 文節を指定しないでください。
- 51** カタログ・データベース・パーティションから LANGUAGE SQL を指定して CREATE PROCEDURE を発行してください。
- 52** 式を変更しないでください。ORGANIZE BY 文節や、生成列であるそのメンバーの生成式の構成を変更するには、その表をドロップし、再作成して、表に再移植する必要があります。
- 53** スクロール可能カーソルの選択リストを

- 変更するか、これらのタイプに列を組み込まないでください。
- 54 INPLACE オプションを指定せずに、REORG コマンドを再発行してください。
- 55 集線装置または連合データベース・サポートをオフにしてください。集線装置をオフにするには、データベース・マネージャー・パラメーター MAX\_CONNECTIONS を MAX\_COORDAGENTS 以下に設定してください。連合データベース・システムのサポートをオフにするには、データベース・マネージャー構成の FEDERATED パラメーターを NO に設定してください。
- 56 REORG INDEXES コマンドを再発行し、ALLOW NONE または ALLOW READ を指定してください。
- 57 REORG INDEXES コマンドを再発行し、ALLOW NONE または ALLOW READ を指定してください。
- 58 XML 入力を受け入れる関数の 1 つに XML データを入力し、関数の出力をデータベースに保管するか、アプリケーションに戻してください。
- 59 並列環境では、NO SQL オプションを指定して定義された関数と方式だけを使用してください。
- 60 型付き表に依存するビューをドロップしてください。型付き表で ALTER TABLE ALTER COLUMN SET INLINE LENGTH ステートメントを再発行し、ドロップされたビューを再作成してください。
- 61 テキスト検索関数をチェック制約または生成された列の式に使用しないでください。可能であれば、CONTAINS ではなく LIKE 関数を使用してください。

- 62 このビューで WITH CHECK OPTION 文節を指定しないでください。
- 63 スクロール可能カーソルの選択リストを変更するか、これらのタイプに列を組み込まないでください。
- 64 このプラットフォームでは、連合プロセッシングを実行しないでください。
- 65 ニックネームのローカル・タイプを現行タイプから指定タイプに変更しないでください。
- 66 SYSSTRUCT をトランスフォーム・グループとして指定しないでください。

sqlcode: -270

sqlstate: 42997

---

**SQL0271N** *fid fid* を伴う表の索引ファイルがないか、無効です。

**説明:** *fid fid* を伴う表の索引ファイルは処理中に要求されます。このファイルは無くなっているか、有効でないかのいずれかです。

このステートメントを処理することができず、アプリケーションは、まだデータベースに接続されています。この条件は、この表の索引を使用しないほかのステートメントには影響を及ぼしません。

**ユーザーの処置:** すべてのユーザーが、そのデータベースから切断されていることを確認し、すべてのノードに RESTART DATABASE コマンドを実行してください。そのあと要求を再試行してください。

この索引 (または索引群) は、データベースが再始動されるときに再作成されます。

sqlcode: -271

sqlstate: 58004

---

**SQL0276N** リストア・ベンディング状態にあるため、データベース *name* に接続することはできません。

**説明:** 接続が完了する前にデータベースをリストアしてください。

接続は行われていません。

**ユーザーの処置:** データベースをリストアし、再度 CONNECT ステートメントを発行してください。

**sqlcode:** -276

**sqlstate:** 08004

---

**SQL0279N** データベース接続が COMMIT 処理中に終了しました。トランザクションが確定していない可能性があります。理由コード = *reason-code*

**説明:** コミット処理がエラーを検出しました。このトランザクションは、コミット状態に入っていますが、コミット処理は完了していない可能性があります。このアプリケーションのデータベース接続は、終了しています。

エラーの原因は、*reason-code* に示されています。

- 1 トランザクションで呼び出されたノードが失敗しています。
- 2 ノードの一つに対してコミットが拒否されました。詳細については管理通知ログをチェックしてください。

**ユーザーの処置:** エラーの原因を判別してください。最も一般的なエラーの原因はノード障害または接続障害なので、システム管理者に連絡して援助を求める必要があります。RESTART DATABASE コマンドはこのトランザクションのコミット処理を完了します。

**sqlcode:** -279

**sqlstate:** 08007

---

**SQL0280W** ビュー、トリガー、またはマテリアライズ照会表 *name* が、既存の作動不能のビュー、トリガーまたはマテリアライズ照会表を置換しました。

**説明:** 既存の作動不能のビュー、トリガー、またはマテリアライズ照会表 *name* が以下のように置き換えられました。

- CREATE VIEW ステートメントの結果としての新しいビュー定義
- CREATE TRIGGER ステートメントの結果としての新しいトリガー定義
- CREATE SUMMARY TABLE ステートメントの結果としての新しいマテリアライズ照会表定義

**ユーザーの処置:** 必要ありません。

**sqlcode:** +280

**sqlstate:** 01595

---

**SQL0281N** 表スペース *tablespace-name* はシステム管理表スペースであるため、追加コンテナでは変更できません。

**説明:** 追加コンテナはシステム管理表スペースに追加できません。この場合に対する例外は、ノード・グループが表スペースなしでノードを追加するように修正され、次にコンテナが ALTER TABLESPACE コマンドを使用して新規ノード上に一度追加された場合です。一般的に、追加のコンテナを加えるには、表スペースがデータベースに管理されている必要があります。

ステートメントは処理できません。

**ユーザーの処置:** システム管理表スペースにさらにコンテナを増やすには、表スペースをドロップしてからコンテナを増やして表スペース再度作成し、それぞれのコンテナがコンテナ・サイズの限度と同じサイズであるか、あるいはそれより小さいサイズであるか、を確認するか、また DMS 表スペースに変更してください。



**sqlcode:** -281

**sqlstate:** 42921

---

**SQL0282N** 表スペース内の少なくとも 1 つの表 *table-name* が、他の表スペースに 1 つ以上の部分を持っているため、表スペース *tablespace-name* がドロップできません。

**説明:** 示された表スペースの表に、その表のすべての部分が入っているわけではありません。複数の表スペースが指定された場合は、指定された表スペースのいずれかにある表に、リスト内にその表のすべての部分が入っているわけではありません。基本表、索引、または長いデータが他の表スペースに存在する可能性があるため、表スペースのドロップによって表が完全にドロップされません。そのため、表が不整合状態に置かれ、そのために表スペースをドロップできません。

**ユーザーの処置:** 表スペースのドロップを試行する前に表スペース *tablespace-name* に入っているすべてのオブジェクトがこの表スペースのすべての部分を収容していることを確認するか、またはリスト内の部分の入ったこれらの表スペースをドロップに組み込みます。

これには、表スペースをドロップする前に、表 *table-name* のドロップが必要になる場合があります。

**sqlcode:** -282

**sqlstate:** 55024

---

**SQL0283N** **SYSTEM TEMPORARY** 表スペース *tablespace-name* だけが、データベース内で *page-size* ページ・サイズを持つ **SYSTEM TEMPORARY** 表スペースであるため、ドロップすることはできません。

**説明:** データベースには、カタログ表スペースのページ・サイズと同じページ・サイズの **SYSTEM TEMPORARY** 表スペースが少なくとも 1 つ入っ

ている必要があります。表スペース *tablespace-name* をドロップすると、*page-size* ページ・サイズを持つ最後の **SYSTEM TEMPORARY** 表スペースがデータベースからドロップされます。

**ユーザーの処置:** この表スペースのドロップを試行する前に、データベースに他の *page-size* ページ・サイズの **SYSTEM TEMPORARY** 表スペースがあることを確認してください。

**sqlcode:** -283

**sqlstate:** 55026

---

**SQL0284N** 文節 *clause* の後に続く表スペース *tablespace-name* が *tablespace-type* 表スペースであるため、表は作成されませんでした。

**説明:** **CREATE TABLE** または **DECLARE GLOBAL TEMPORARY TABLE** ステートメントが、文節 *clause* の後に、この文節に有効な表スペースのタイプではない表スペース *tablespace-name* を指定しました。

これは以下の状況で起きます。

- 通常の表の場合、*tablespace-name* が **IN** 文節に指定されていて、表スペースが **REGULAR** 表スペースではありません。
- 宣言された一時表の場合、*tablespace-name* が **IN** 文節に指定されていて、表スペースが **USER TEMPORARY** 表スペースではありません。
- *tablespace-name* が **LARGE** 文節に指定されていますが、表スペースが、データベースによって管理される **LARGE** ではありません。
- *tablespace-name* が **INDEX IN** 文節に指定され、表スペースが、データベース、表スペースによって管理される **REGULAR** または **LARGE** ではありません。

**ユーザーの処置:** **CREATE TABLE** ステートメントを訂正して、*clause* 文節に適切なタイプで表スペースを指定してください。

**sqlcode:** -284

**sqlstate:** 42838

---

**SQL0285N** 1 次表スペース *tablespace-name* がシステム管理表スペースであるため、表 *table-name* の索引と長い列のいずれか、または両方を、独立表スペースに割り当てることができません。

**説明:** 1 次表スペースがシステム管理表スペースの場合は、表のすべての部分はその表スペースに入っている必要があります。1 次表スペース、索引表スペース、長い表スペースがデータベース管理表スペースの場合にのみ、表の部分を独立表スペースに持つことができます。

**ユーザーの処置:** 1 次表スペースにデータベース管理表スペースを指定するか、または表の部分を他の表スペースに割り当てないでください。

**sqlcode:** -285

**sqlstate:** 42839

---

**SQL0286N** 許可 ID *user-name* が使用を許可されている少なくとも *pagesize* のページ・サイズを持つデフォルトの表スペースが検出されませんでした。

**説明:** CREATE TABLE または DECLARE GLOBAL TEMPORARY TABLE ステートメントで表スペースが指定されませんでした。また、許可 ID *user-name* が USE 特権を持っている正しいタイプ (宣言済み一時表では USER TEMPORARY) の表スペースで十分なページ・サイズ (*pagesize* 以上) のものが見つかりませんでした。

表に、十分なページ・サイズがあるかどうかは、行のバイト・カウントあるいは列の数で判別されます。

**ユーザーの処置:** *pagesize* 以上のページ・サイズがある正しいタイプ (REGULAR または USER TEMPORARY) の表スペースが存在すること、ま

た、その表スペースに対して許可 ID *user-name* が USE 特権を持っていることを確認してください。

**sqlcode:** -286

**sqlstate:** 42727

---

**SQL0287N** SYSCATSPACE はユーザー・オブジェクトには使用できません。

**説明:** CREATE TABLE または GRANT USE OF TABLESPACE ステートメントが、カタログ表のために予約されている表スペース SYSCATSPACE を指定しました。

**ユーザーの処置:** 別の表スペース名を指定してください。

**sqlcode:** -287

**sqlstate:** 42838

---

**SQL0288N** 大きな表スペースは、MANAGED BY SYSTEM を使って定義できません。

**説明:** 定義される表スペースは、ラージ・オブジェクトと長ストリングで使用されます。これらは、データベース管理スペースに定義された表スペースにのみ格納することができます。したがって、システム管理スペースを使用するように、大きな表スペースを定義できません。

**ユーザーの処置:** キーワード LARGE を除去するか、CREATE TABLESPACE ステートメントで MANAGED BY DATABASE に変更してください。

**sqlcode:** -288

**sqlstate:** 42613

---

**SQL0289N** 表スペース *tablespace-name* の新規ページを割り振れません。

**説明:** 以下に示す条件の 1 つが成立しています。

1. この SMS 表スペースに割り当てられたコンテナのいずれかが、最大サイズに達しました。これが、エラーの原因である可能性があります。
2. この DMS 表スペースに割り当てられているすべてのコンテナがいっぱいです。これが、エラーの原因である可能性があります。
3. この DMS 表スペースの表スペース・オブジェクト表がいっぱいです。
4. 再平衡が進行中ですが、新しく追加されるスペースを使用可能にするのに十分なものには進んでいません。
5. リダイレクト・リストアの実行先のコンテナが小さ過ぎます。
6. リダイレクト・リストアに続いてロールフォワードが実行されていますが、この表スペースに割り当てられたコンテナはすべていっぱいです。
7. コンテナ追加をスキップするロールフォワードが実行され、この表スペースに割り当てられたすべてのコンテナがいっぱいになっています。
8. 5 つ未満の使用可能エクステントを使って表スペースを作成しようとしてしました。

**ユーザーの処置:** エラーの原因に対応するアクションを実行してください。

1. DMS TABLESPACE に切り替えるか、あるいは (ディレクトリー数) >= (最大表サイズ / 最大ファイル・サイズ) のようなディレクトリー (パス) を使用して、SMS TABLESPACE を再作成してください。最大ファイル・サイズはオペレーティング・システムによって変わることに注意してください。
2. 再平衡プログラムが新しいページを使用可能にした後で、新しいコンテナを DMS 表スペースに追加して、操作をやり直してください。
3. この DMS 表スペースから不要な表をドロップしてください。

4. 再平衡プログラムが進行するのを待ってください。
5. リダイレクト・リストアを大きなコンテナで再度実行してください。
6. リダイレクト・リストアを大きなコンテナで再度実行してください。
7. コンテナの追加を許可してロールフォワードを再度実行するか、リダイレクト・リストアを大きなコンテナに対して実行してください。
8. 表スペースが少なくとも 5 つの使用可能エクステントを持っていることを確認して、CREATE TABLESPACE ステートメントを再発行してください。

**sqlcode:** -289

**sqlstate:** 57011

---

**SQL0290N 表スペースのアクセスが許されていません。**

**説明:** 意図されたアクセスに対して無効な状態にある表スペースにアクセスしようとする処理は、許されていません。

- 表スペースが静止状態の場合は、表スペースを静止状態に保留している処理のみが、その表スペースに対するアクセスを許されます。
- 表スペースが他のいずれかの状態の場合は、指定されたアクションを実行する処理のみが、その表スペースに対するアクセスを許されます。
- アクティブのシステムまたは宣言された一時表の入った、SYSTEM または USER TEMPORARY 表スペースはドロップできません。
- 表スペースが "リストア・ペンディング" 状態でない限り、SET CONTAINER API はコンテナ・リストの設定に使用できません。

**ユーザーの処置:** 解決策は以下の通りです。



- 表スペースが静止状態の場合は、その表スペースで静止共有または静止更新の獲得を試みてください。あるいは、表スペースのリセットの静止を試みてください。
- 表スペースが他のいずれかの状態の場合は、アクセスする前に、その表スペースが通常の状態に戻るまで待ってください。

表スペースの状態に関する詳細については「管理ガイド」を参照してください。

**sqlcode:** -290

**sqlstate:** 55039

---

**SQL0291N 状態変換は、表スペースでは使用できません。**

**説明:** 表スペースの状態の変更が試みられました。新しい状態が表スペースの現在の状態との整合性を持っていないか、または特定状態をオフにしようとしたが、表スペースがその状態ではありません。

**ユーザーの処置:** 表スペースの現在の状態に応じて、バックアップの完了、ロードの完了、ロールフォワードの完了などが起きると、表スペースの状態が変わります。表スペース状態に関する詳細については、「システム管理ガイド」を参照してください。

**sqlcode:** -291

**sqlstate:** 55039

---

**SQL0292N 内部データベース・ファイルが作成できませんでした。**

**説明:** 内部データベース・ファイルが作成できませんでした。

**ユーザーの処置:** そのファイルの入ったディレクトリーが、アクセス可能 (たとえば、取り付けられている) であること、およびデータベース・インスタンス所有者によって書き込み可能であることをチェックしてください。

**sqlcode:** -292

**sqlstate:** 57047

---

**SQL0293N 表スペース・コンテナのアクセス・エラーです。**

**説明:** このエラーは、以下のいずれかの条件によって発生した可能性があります。

- コンテナ (ディレクトリー、ファイルまたはロー・デバイス) が見つかりませんでした。
- コンテナに、適切な表スペースに所有されていることを示すタグが付いていません。
- コンテナ・タグが壊れています。

このエラーはデータベースの始動時および ALTER TABLESPACE SQL ステートメントの処理時に返されます。

**ユーザーの処置:** 以下のアクションを試行してください。

1. ディレクトリー、ファイル、または装置が存在し、ファイル・システムがマウントされている (それが独立したファイル・システム上にある場合) ことを確かめてください。コンテナは、データベース・インスタンス所有者によって、読み書き可能でなければなりません。
2. 最新のバックアップがある場合は、表スペースまたはデータベースのリストアを試みてください。正しくないコンテナのためにリストアが失敗し、コンテナが DEVICE タイプでない場合は、まず手操作でそのコンテナを取り除いてください。

エラーが SWITCH ONLINE オプション付きの ALTER TABLESPACE SQL ステートメントの処理から返された場合は、上記で記述された問題を訂正した後にステートメントを再発行してください。

エラーが残る場合、IBM サービス担当者に連絡してください。

**sqlcode:** -293

**sqlstate:** 57048

---

**SQL0294N** コンテナはすでに使用中です。

**説明:** 表スペース・コンテナが共有できないことがあります。このエラーの原因として可能性のあるものは以下のとおりです。

- CREATE TABLESPACE または ALTER TABLESPACE ステートメントに、他の表スペースですすでに使用中のコンテナが組み込まれていました。
- CREATE TABLESPACE または ALTER TABLESPACE ステートメントに、ドロップされている表スペースからのコンテナが組み込まれていましたが、DROP ステートメントはコミットされていませんでした。
- ノードを追加するのに使用される ALTER NODEGROUP ステートメントが、同じ物理ノードにある LIKE ノードのコンテナを使用していました。そのため、これらのコンテナはすでに使用中となっています。
- CREATE TABLESPACE または ALTER TABLESPACE ステートメントが、単一の物理ノードの 2 つ以上の論理ノードにある同じコンテナを使用しようとしていました。同じコンテナを、同じ物理ノードの 2 つ以上のノードに対して使用することはできません。
- ADD NODE コマンドまたは API が、同じ物理ノードにある LIKE ノードの SYSTEM TEMPORARY 表スペースからコンテナを使用しました。そのため、これらのコンテナはすでに使用中となっています。
- CREATE TABLESPACE または ALTER TABLESPACE ステートメントに、もう存在していないけれども正しくドロップされていない、別のデータベースからのコンテナが組み込まれていました。実際、このコンテナは使用されていませんが、使用中であるとタグ付けされています。そのため、タグが外されるまで、DB2 はコンテナの使用を許可しませ

ん。ただし、タグが外されるときに、このコンテナが同じデータベースまたは別のデータベースによって使用中ではないことを確認することが重要です。タグを外したときにコンテナが使用中であれば、関係するデータベースは損傷を受けます。

**ユーザーの処置:** コンテナがユニークであるかを確認してください。

- CREATE または ALTER TABLESPACE ステートメントに対し、表スペースに別のコンテナを指定してください。
- ドロップされた表スペースに属するコンテナが組み込まれた CREATE または ALTER TABLESPACE ステートメントの場合、DROP ステートメントがコミットされてから再度試行するか、あるいは別のコンテナを指定してください。
- ALTER NODEGROUP ステートメントに対し、WITHOUT TABLESPACES 文節を使用してこのステートメントを再発行し、新規ノードのユニーク・コンテナを作成するのに ALTER TABLESPACE ステートメントを使用してください。
- 物理ノード上に複数の論理ノードが組み込まれた環境にある CREATE または ALTER TABLESPACE ステートメントの場合、同じコンテナがこのような論理ノードで指定されていないことを確認してください。
- ADD NODE コマンドまたは API に対し、WITHOUT TABLESPACES 文節を使用してステートメントを再発行し、SYSTEM TEMPORARY 表スペースの新規ノードでユニーク・コンテナを作成するのに ALTER TABLESPACE ステートメントを使用してください。
- もう存在しなくても正しくドロップされていないデータベースに属していた DMS コンテナの使用を試みている場合、db2untag ユーティリティを使用して DB2 コンテナ・タグを外すことができます。このタグが外されると DB2 はコンテナの解放を考慮し、このコン

テナーは CREATE TABLESPACE または ALTER TABLESPACE ステートメントで使用できます。

注: db2untag の使用には十分に気を付けてください。データベースによって使用されているコンテナに対して db2untag コマンドを出すと、そのコンテナを使用していたデータベース、および現在でも使用しているデータベースの両方が損傷を受けます。

**sqlcode:** -294

**sqlstate:** 42730

---

### SQL0295N 表スペースのすべてのコンテナ名を結合した長さが、長すぎます。

**説明:** コンテナのリストを格納するために必要な合計スペースが、表スペース・ファイルのこの表スペースに割り当てられたスペースを超えました。

**ユーザーの処置:** 以下の 1 つ以上を試みてください。

- シンボリック・リンク、取り付けられたファイル・システムなどを使用して、新しいコンテナ名を短くしてください。
- 表スペースのバックアップを行った後で、データベース管理ユーティリティを使用して、コンテナの数と名前の長さ、またはそのいずれかを減らしてください。その後で、表スペースを新しいコンテナにリストアしてください。

**sqlcode:** -295

**sqlstate:** 54034

---

### SQL0296N 表スペースの限界を超えています。

**説明:** このデータベースには、最大数の表スペースが入っています。もう作成することはできません。

**ユーザーの処置:** もう使用されていない表スペース

を削除してください。表スペースのすべてのデータを 1 つの表スペースに移動し、他の表スペースを削除して、小さな表スペースを結合してください。

**sqlcode:** -296

**sqlstate:** 54035

---

### SQL0297N コンテナのパス名が長すぎます。

**説明:** コンテナ名を指定する絶対パスが、最大許容長を超えています。データベース・ディレクトリに関連するパスとして、コンテナが指定されている場合は、それら 2 つの値の連結が最大長を超えてはなりません。

管理通知ログで詳細を参照できます。

**ユーザーの処置:** パスの長さを短くしてください。

**sqlcode:** -297

**sqlstate:** 54036

---

### SQL0298N コンテナ・パスが正しくありません。

**説明:** コンテナ・パスが、以下のいずれかの要件に違反しています。

- コンテナ・パスは、有効な完全修飾された絶対パス、または有効な相対パスでなければなりません。文字は、データベース・ディレクトリに関連して解釈されます。
- EXTEND、REDUCE、RESIZE および DROP 操作に対し、指定されたコンテナ・パスが存在している必要があります。
- パスはインスタンス ID に対して読み / 書き可能でなければなりません (UNIX ベース・システムのファイル許可をチェックしてください)。
- コンテナはコマンドに指定したタイプでなければなりません (ディレクトリ、ファイルまたは装置)。
- システム管理表スペースのコンテナ (ディレクトリ) は、コンテナとして指定された場

合は空でなければならず、他のコンテナの下にネストしてはなりません。

- 1 つのデータベースに対するコンテナは、別のデータベースのディレクトリの下に位置してはならず、別のデータベースに対して現れるディレクトリの下にもなれない場合があります。この規則は、形式が `SQLnnnnn` ('n' は数字) のディレクトリには適用されません。
- コンテナは、オペレーティング・システムのファイル・サイズ制限内でなければなりません。
- ドロップ済みデータベース管理表スペースのコンテナ (ファイル) は、すべてのエージェントが終了および開始した後で、システム管理表スペースのコンテナ (ディレクトリ) としてのみ再発行できます。
- リダイレクト・リストア中に、SMS コンテナが DMS 表スペースに指定されたか、あるいは DMS コンテナが SMS 表スペースに指定されました。
- EXTEND、REDUCE、RESIZE、または DROP 操作に対して指定されたコンテナ・タイプは、コンテナが作成されたときに指定されたコンテナ・タイプ (FILE または DEVICE) と一致しません。

## SQL0400 - SQL0499

**SQL0401N** 演算 *operator* のオペランドのデータ・タイプが一致していません。

**説明:** SQL ステートメント内の演算 *operator* に、数値と非数値オペランドが混在しているか、または演算オペランドが互換ではありません。

**連合システム・ユーザー:** このデータ・タイプ違反は、データ・ソースまたは連合サーバーで起きた可能性があります。

一部のデータ・ソースは、*operator* に適切な値を提供しません。この場合、メッセージ・トークンは “<data source>:UNKNOWN” の形式になります。これは、指定されたデータ・ソースの実際の値が不明であることを示します。

このメッセージは、コンテナへのアクセスを DB2 に禁止する、その他の予期しないエラーが発生した場合にも返されます。

**ユーザーの処置:** 別のコンテナ・ロケーションを指定するか、またはコンテナを変更して DB2 に受け入れ可能にし (ファイル許可の変更など)、もう一度やり直してください。

**sqlcode:** -298

**sqlstate:** 428B2

---

**SQL0299N** コンテナは、すでに表スペースに割り当てられています。

**説明:** 追加しようとしたコンテナが、すでに表スペースに割り当てられていました。

**ユーザーの処置:** 別のコンテナを選択して、もう一度やり直してください。

**sqlcode:** -299

**sqlstate:** 42731

ステートメントは処理できません。

**ユーザーの処置:** すべてのオペランドのデータ・タイプをチェックして、データ・タイプが比較可能であること、およびステートメント内の使用法に互換性があることを確認してください。

SQL ステートメントのオペランドがすべて正しく、しかもビューにアクセスしている場合には、ビューのすべてのオペランドのデータ・タイプをチェックしてください。

**連合システム・ユーザー:** 原因が不明の場合、要求の失敗を引き起こすデータ・ソースと問題を分離して (障害の起こったデータ・ソースを識別するための手順については問題判別の手引き を参

照してください。) そのデータ・ソースのデータ・タイプ制約事項を検証してください。

**sqlcode:** -401

**sqlstate:** 42818

---

**SQL0402N** 算術関数または演算 *operator* のオペランドのデータ・タイプが、数値ではありません。

**説明:** 非数値オペランドが、算術関数または演算子 *operator* に対して指定されています。

ステートメントは処理できません。

**ユーザーの処置:** SQL ステートメントの構文を調べ、指定された関数または演算子のすべてのオペランドが数値となるように修正してください。

連合システム・ユーザー: 理由が不明な場合は、要求を失敗させたデータ・ソースに対して問題を分離し (障害の起きたデータ・ソースを識別するための手順については「問題判別の手引き」を参照してください)、そのデータ・ソースに適用した操作を調べてください。

**sqlcode:** -402

**sqlstate:** 42819

---

**SQL0403W** 新しく定義された別名 *name* は、現在未定義のオブジェクト *name2* に解決されています。

**説明:** 別名 <name> は、以下のように定義されています。

- データベースにまだ定義されていない表またはビュー。
- データベースに定義されていない表またはビューによって解決される他の別名。

オブジェクト <name2> は未定義のオブジェクトです。SQL ステートメント (CREATE ALIAS 以外) が、新しく作成された別名を正常に使用する

するには、このオブジェクトが存在している必要があります。指定された別名 <name> は作成されず。

別名の作成について、作動不能なビューは未定義であると見なされます。

**ユーザーの処置:** SQL ステートメント (CREATE ALIAS 以外) の新しく作成された別名を使用する前に、未定義オブジェクト <name2> が定義されていることを確認してください。

**sqlcode:** +403

**sqlstate:** 01522

---

**SQL0404N** UPDATE または INSERT ステートメント内のストリングが、列 *name* に対して長すぎます。

**説明:** INSERT または UPDATE ステートメント内で指定された値が、示されている列に保管できるストリングの最大長を超えています。

ステートメントは処理できません。

**注:** INSERT または UPDATE ステートメントの構文に応じて、SQLCA 内に *name* が返されることも返されないこともあります。

**ユーザーの処置:** オブジェクトとなる列の長さをチェックして、プログラムまたは SQL ステートメントを修正して、挿入または更新するストリングの長さが最大長を超えないようにしてください。

**sqlcode:** -404

**sqlstate:** 22001

---

**SQL0405N** 数値リテラル *literal* は、値が有効な値の範囲外になっているため、無効です。

**説明:** 示された数値リテラルが適切な範囲内にありません。

SQL 値の適切な範囲は以下の通りです。





DEFAULT のない NOT NULL として宣言されています。したがって、以下の制約を受けません。

- NULL のデフォルト値はその列に挿入できない。
  - 更新はその列の NULL のデフォルト値を設定できない。
  - トリガーの SET 遷移変数ステートメントは、その列の NULL のデフォルト値を設定できない。
- INSERT ステートメントの列名リストに、表定義で NOT NULL および WITH DEFAULT なしで宣言された列がありません。
  - INSERT ステートメントのビューに、基本表定義で NOT NULL および WITH DEFAULT なしで宣言された列がありません。

*name* の値が形式 “TBSpaceID=n1, TABLEID=n2, COLNO=n3” であれば、エラーが出されたときに SQL ステートメントの列名が使用可能ではありませんでした。示されている値は、NULL 値を許可していない基本表の表スペース、表、および列番号を表しています。

連合システム・ユーザー: この状態は、連合サーバーまたはデータ・ソースによって検出されます。一部のデータ・ソースは、*name* に適切な値を提供しません。この場合、メッセージ・トークンは “<data source>:UNKNOWN” の形式になります。これは、指定されたデータ・ソースの実際の値が不明であることを示します。

ステートメントは処理できません。

**注:** いくつかの状況では、トークン *name* が充てんされない場合があります (SQLCA の 「sqlerrmc」 フィールドが充てんされません)。

**ユーザーの処置:** オブジェクト表定義を調べて、NOT NULL 属性を持つ表の列と、WITH DEFAULT 属性を持たない表の列を判別した後で、SQL ステートメントを修正してください。

*name* の値が形式 “TBSpaceID=n1, TABLEID=n2, COLNO=n3” であれば、以下の照会を使用して表名および列名を判別することができます。

```
SELECT C.TABSCHEMA, C.TABNAME,
 C.COLNAME
FROM SYSCAT.TABLES AS T,
 SYSCAT.COLUMNS AS C
WHERE T.TBSPACEID = n1
AND T.TABLEID = n2
AND C.COLNO = n3
AND C.TABSCHEMA = T.TABSCHEMA
AND C.TABNAME = T.TABNAME
```

この照会で識別される表および列は、SQL ステートメントに障害があったビューの基本表だと考えられます。

連合システム・ユーザー: 理由が不明な場合は、要求を失敗させたデータ・ソースに対して問題を分離し (障害の起きたデータ・ソースを識別するための手順については「問題判別の手引き」を参照してください)、そのデータ・ソースのオブジェクト定義を調べてください。デフォルト (NULL および NOT NULL) はデータ・ソースの間で同じである必要はないことに注意してください。

**sqlcode:** -407

**sqlstate:** 23502

---

**SQL0408N** 値には、その割り当てターゲットのデータ・タイプとの互換性がありません。ターゲット名は *name* です。

**説明:** SQL ステートメントによって列、パラメーター、SQL 変数、または遷移変数に割り当てられる値のデータ・タイプに、その割り当てターゲットの宣言されたデータ・タイプとの互換性はありません。両方が、以下のデータ・タイプでなければなりません。

- 数値
- 文字
- 図形

- 日付または文字
- 時刻または文字
- タイム・スタンプまたは文字
- データ・リンク
- 同一の特殊タイプ
- 値のターゲット・タイプが、列のターゲット・タイプのサブタイプである参照タイプ
- 同じユーザー定義構造化タイプ: そうでなければ、値の静的タイプは、ターゲットの静的タイプ (宣言されたタイプ) のサブタイプでなければなりません。ホスト変数が入っている場合、関連するホスト変数の組み込みタイプが、ステートメントのトランスフォーム関数に定義されている TO SQL トランスフォーム関数のパラメーターと互換性がなければなりません。

ステートメントは処理できません。

**ユーザーの処置:** ステートメント (およびターゲット表またはビュー) を調べて、ターゲット・データ・タイプを判別してください。割り当てられている変数、式、またはリテラル値が割り当てターゲットとして正しいデータ・タイプであることを確認してください。ユーザー定義の構造化タイプの場合、ステートメントに対するトランスフォーム・グループで定義された TO SQL トランスフォーム関数のパラメーターを、割り当て済みターゲットとしても考慮してください。

**sqlcode:** -408

**sqlstate:** 42821

#### **SQL0409N COUNT 関数のオペランドが無効です。**

**説明:** SQL ステートメントに指定されているような、COUNT 関数のオペランドは SQL 構文の規則に適合しません。COUNT(\*) および COUNT(DISTINCT 列) のみが許されています。

ステートメントは処理できません。

**ユーザーの処置:** COUNT(\*) または COUNT(DISTINCT 列) を指定してください。

注: このメッセージは、バージョン 2 以前の DB2 のバージョンにのみ適用されます。

**sqlcode:** -409

**sqlstate:** 42607

#### **SQL0410N 浮動小数点リテラル literal が、30 文字を超えています。**

**説明:** 示された浮動小数点リテラルが、先行する 0 を除いても 30 文字を超える長さになっています。浮動小数点リテラルの最大長は 30 文字です。

ステートメントは処理できません。

**ユーザーの処置:** 示されたりテラルを短くしてください。

**sqlcode:** -410

**sqlstate:** 42820

#### **SQL0412N 1 つの列しか許可されていない副照会から複数の列が返されました。**

**説明:** SQL ステートメントのコンテキストでは、結果として 1 つの列だけを持つよう副選択が指定されています。

ステートメントは処理できません。

**ユーザーの処置:** スカラー副選択が 1 つだけ許可されている場合は、1 つの列だけを指定してください。

**sqlcode:** -412

**sqlstate:** 42823

#### **SQL0413N 数値データ・タイプの変換中に、オーバーフローが発生しました。**

**説明:** SQL ステートメントの処理中、ある数値タイプのデータを別のタイプへ変換するときにオーバーフローが起きました。数値変換は SQL の標準規則にしたがって実行されます。

連合システム・ユーザー: 数値変換は、連合サー



パー、データ・ソース、またはその両方で起きる可能性があります

ステートメントは処理できません。データの検索、更新、または削除は実行されませんでした。

**ユーザーの処置:** SQL ステートメントの構文を調べて、エラーの原因を判別してください。問題がデータに依存する場合は、エラーが起きたときに処理されていたデータを調べる必要があります。

**連合システム・ユーザー:** 理由が不明な場合は、要求を失敗させたデータ・ソースに対して問題を分離し (障害の起きたデータ・ソースを識別するための手順については「問題判別の手引き」を参照してください)、そのデータ・ソースのデータ範囲制約事項を調べてください。

**sqlcode:** -413

**sqlstate:** 22003

---

**SQL0415N** 対応する列のデータ・タイプは、セット演算子を含む全選択または **INSERT** または全選択の **VALUES** 文節の複数行で互換性がありません。

**説明:** このエラーが発生する可能性のあるステートメントはたくさんあります。

- このエラーは、セット操作 (UNION、INTERSECT、または EXCEPT) の入った SELECT または VALUES ステートメント内で発生する可能性があります。SELECT または VALUES ステートメントを作成する副選択または全選択の対応する列は、互換性がありません。
- このエラーは、複数行に挿入している INSERT ステートメント内で発生する可能性があります。この場合、VALUES 文節内で指定された行の対応する列は、互換性がありません。
- このエラーは、複数行を伴い使用される VALUES 文節がある SELECT または VALUES ステートメント内で発生する可能性

があります。この場合、VALUES 文節内で指定された行の対応する列は、互換性がありません。

以下のいずれかの理由で列に互換性がありません。

- 2 つの列が両方とも文字になっていない。
- 2 つの列が両方とも数字になっていない。
- 2 つの列が両方とも日付になっていない。
- 2 つの列が両方とも時刻になっていない。
- 2 つの列が両方ともタイム・スタンプになっていない。
- 2 つの列が両方ともグラフィックになっていない。
- 2 つの列が同じユーザー定義の異なるタイプになっていない。

列のデータ・タイプが文字、日付、時刻またはタイム・スタンプの場合、対応する列がリテラル・ストリングになります。

ステートメントは処理できません。

**ユーザーの処置:** SELECT ステートメントで使用される列名または VALUES 文節の式を訂正し、対応するすべての列が互換タイプになるようにしてください。

**sqlcode:** -415

**sqlstate:** 42825

---

**SQL0416N** UNION ALL 以外のセット演算子で接続された SELECT または VALUES ステートメントには、**254** バイトを超える結果列を指定できません。

**説明:** セット演算子で接続された SELECT または VALUES ステートメントのいずれかが、254 バイトより長い結果列を指定しています。254 バイトより長い VARCHAR または VARGRAPHIC 結果列は、UNION ALL セット演算子とのみ使用することができます。

ステートメントは処理できません。

**ユーザーの処置:** UNION 演算子の代わりに、UNION ALL を使用するか、または 254 バイトより長い結果列を、SELECT または VALUES ステートメントから取り除いてください。

**sqlcode:** -416

**sqlstate:** 42907

---

**SQL0417N** 準備されたステートメント・ストリングに、同じ演算子のオペランドとしてパラメーター・マーカーが含まれています。

**説明:** PREPARE または EXECUTE IMMEDIATE のオブジェクトとして指定されたステートメント・ストリングに、CAST 指定のない同一の演算子のオペランドとして使用されているパラメーター・マーカーを持つ述部または式が含まれています。例:

? > ?

ステートメントは処理できません。

**ユーザーの処置:** このような構文はサポートされていません。CAST 指定を使用して、少なくとも 1 つのパラメーター・マーカーをデータ・タイプに指定してください。

**sqlcode:** -417

**sqlstate:** 42609

---

**SQL0418N** ステートメントに、無効なパラメーター・マーカーの使用が含まれています。

**説明:** タイプなしパラメーター・マーカーは、以下では使用できません。

- SELECT リスト内
- 日時算術処理の唯一の引き数として
- いくつかの場合には、スカラー関数の唯一の引き数として
- GROUP BY 文節中のソート・キーとして

パラメーター・マーカーは、以下で使用することはできません。

- 準備されたステートメントではないステートメント内
- CREATE VIEW ステートメントの全選択内
- CREATE TRIGGER ステートメントのトリガー・アクション内
- DB2 Query Patroller によって取り込まれた照会内

ステートメントは処理できません。

**ユーザーの処置:** ステートメントの構文を訂正してください。タイプなしパラメーター・マーカーが許されない場合は、CAST 指定を使用して、パラメーター・マーカーをデータ・タイプに指定してください。

**sqlcode:** -418

**sqlstate:** 42610

---

**SQL0419N** 結果の位取りが負になるため、10 進数の除算は無効です。

**説明:** 指定された 10 進除算は、結果の位取りが負の値になるために有効ではありません。

10 進除算の結果の位取りを計算するために内部的に使用される式は、以下の通りです。

$$\text{Scale of result} = 31 - np + ns - ds$$

ここで、np は分子の精度、ns は分子の位取り、ds は分母の位取りを表します。

**連合システム・ユーザー:** 10 進数除法は、連合サーバー、データ・ソース、またはその両方で起きる可能性があります指定された 10 進数除法の結果は、そのデータ・ソースの位取りが無効になります。

ステートメントは処理できません。

**ユーザーの処置:** 10 進除算で使用される可能性のあるすべての列の精度と位取りを調べてください。整数または短整数は、計算のために 10 進数

に変換される場合があることに注意してください。

**連合システム・ユーザー:** 理由が不明な場合は、要求を失敗させたデータ・ソースに対して問題を分離し（障害の起きたデータ・ソースを識別するための手順については「問題判別の手引き」を参照してください）、そのデータ・ソースのデータ範囲制約事項を調べてください。

**sqlcode:** -419

**sqlstate:** 42911

---

**SQL0420N** 無効な文字が、関数 *function-name* の文字ストリング引き数で見つかりました。

**説明:** 関数 *function-name* が、数値 SQL 定数では無効な文字の入った文字ストリング引き数を持っています。関数は、ターゲット・データ・タイプとして *function-name* を使用する CAST 指定の使用の結果として、呼び出された可能性があります。SQL ステートメントに使用された関数またはデータ・タイプが、*function-name* の同義語である可能性があります。

10 進文字を DECIMAL 関数に指定する場合は、それがデフォルト 10 進文字の代わりに使用する必要がある文字になります。

**ユーザーの処置:** 指定する場合は、数値タイプに変換される文字ストリングが、10 進文字を使用する数値 SQL 定数に有効な文字のみを備えていることを確認してください。

**sqlcode:** -420

**sqlstate:** 22018

---

**SQL0421N** セット演算子のオペランドまたは **VALUES** 文節が、列数と同じ数ではありません。

**説明:** UNION、EXCEPT、INTERSECT などのセット演算子のオペランドは、列数と同じ数でなければなりません。VALUES 文節の行は、列数と

同じ数を持つ必要があります。

ステートメントは処理できません。

**ユーザーの処置:** SQL ステートメントを修正して、各オペランドまたは VALUES 文節の各行が、列数とまったく同じ数を持つようにしてください。

**sqlcode:** -421

**sqlstate:** 42826

---

**SQL0423N** ロケータ変数 *variable-position* は、現在、どんな値も表示していません。

**説明:** ロケータ変数にエラーがあります。有効な結果セットロケータまたはそれに割り当てられた LOB ロケータ変数値がないか、変数と関連したロケータが解放されているか、あるいは結果セット・カーソルがクローズされています。

*variable-position* が提供される場合には、エラーのある変数の序数位置が指定される変数セットに示されます。エラーを検出した時点によっては、データベース・マネージャーが *variable-position* を判別できないことがあります。

*variable-position* は序数位置の代わりに、関数名が識別したユーザー定義の関数から戻ったロケータ値がエラーであること示す "function-name RETURNS" 値を備えている可能性があります。

**ユーザーの処置:** ステートメントの実行の前に、SQL ステートメントで使用されるロケータ変数に有効な値があるように、プログラムを訂正してください。LOB 値は、SELECT INTO ステートメント、VALUES INTO ステートメント、あるいは FETCH ステートメントによって、ロケータ変数に割り当てることができます。結果セット・ロケータは、ASSOCIATE LOCATORS ステートメントから返されます。結果セット・ロケータ値は、基となる SQL カーソルがオープンしている場合にのみ有効です。コミットまたはロールバック操作で SQL カーソルがクローズ

されると、カーソルと関連した結果セット・ロケータはもはや有効ではありません。

戻りカーソルであった場合、割り振りの前にカーソルがオープンされていることを確認してください。

**sqlcode:** -423

**sqlstate:** 0F001

---

**SQL0426N アプリケーションの実行環境では、動的コミットは無効です。**

**説明:** CONNECT TYPE 2 環境 または CICS などの分散トランザクション処理 (DTP) 環境で実行中のアプリケーションが、SQL 動的 COMMIT ステートメントを実行しようとした。SQL 動的 COMMIT ステートメントは、この環境では実行できません。

連合システム・ユーザー: SQL 動的 COMMIT ステートメントをパススルー・セッションで実行することはできません。

**ユーザーの処置:**

- DTP 環境によって提供されるコミット・ステートメントを使用して、コミットを実行してください。たとえば、CICS 環境の場合、これは CICS SYNCPOINT コマンドになります。
- このステートメントがストアード・プロシージャ内で実行された場合は、ステートメントを完全に削除してください。

連合システム・ユーザー: COMMIT ステートメントをコメントにするか、または静的ステートメントとしてコーディングしてください。その後、プログラムを再実行依頼してください。

**sqlcode:** -426

**sqlstate:** 2D528

---

**SQL0427N アプリケーションの実行環境では、動的ロールバックは無効です。**

**説明:** CONNECT TYPE 2 環境 または CICS などの分散トランザクション処理 (DTP) 環境で実行中のアプリケーションが、SQL 動的 ROLLBACK ステートメントを実行しようとした。SQL 動的 ROLLBACK ステートメントは、この環境では実行できません。

連合システム・ユーザー: SQL 動的 ROLLBACK ステートメントをパススルー・セッションで実行することはできません。

**ユーザーの処置:**

- DTP 環境によって提供されるロールバック・ステートメントを使用して、ロールバックを実行してください。たとえば CICS 環境の場合、これは CICS SYNCPOINT ROLLBACK コマンドになります。
- このステートメントがストアード・プロシージャ内で実行された場合は、ステートメントを完全に削除してください。

連合システム・ユーザー: ROLLBACK ステートメントをコメントにするか、または静的ステートメントとしてコーディングしてください。その後、プログラムを再実行依頼してください。

**sqlcode:** -427

**sqlstate:** 2D529

---

**SQL0428N SQL ステートメントは、作業単位の最初のステートメントとしてのみ許可されています。**

**説明:** この SQL ステートメントは、作業単位を開始する他の SQL ステートメントよりも前に実行する必要があります。以下の状況が考えられます。

- SQL ステートメントは作業単位の先頭になければならず、SQL が作業単位内の接続に対して出されている

- SQL ステートメントは作業単位の先頭になければならず、WITH HOLD カーソルが接続に対してオープンされている

ステートメントが DISCONNECT ALL である場合、すべての接続に対して DISCONNECT が実行されるため、いずれかの接続が上記の制限に違反すると、要求が失敗することに注意してください。

**ユーザーの処置:** SQL ステートメント処理の前に、COMMIT または ROLLBACK を出してください。WITH HOLD カーソルがある場合、それらをクローズする必要があります。ステートメントが SET INTEGRITY であれば、COMMIT THRESHOLD 文節を除去してください。

**sqlcode:** -428

**sqlstate:** 25001

---

**SQL0429N 並列 LOB ロケーターの最大数を超えました。**

**説明:** 作業単位ごとに、最大 32,000 の並列 LOB ロケーターが、DB2 ではサポートされています。

**ユーザーの処置:** もっと少ない並列 LOB ロケーターしか必要としないようにプログラムを修正して、もう一度やり直してください。

**sqlcode:** -429

**sqlstate:** 54028

---

**SQL0430N ユーザー定義関数 *function-name* (特定名 *specific-name*) が異常終了しました。**

**説明:** 示された UDF にコントロールがあるときに、異常終了が起きました。

**ユーザーの処置:** UDF を修正する必要があります。UDF の作成者またはデータベース管理者に連絡してください。修正されるまで、その UDF は使用するべきではありません。

**sqlcode:** -430

**sqlstate:** 38503

---

**SQL0431N ユーザー定義関数 *function-name* (特定名 *specific-name*) が、ユーザーによって割り込まれました。**

**説明:** 示された UDF にコントロールがあるときに、ユーザー / クライアント割り込みが起きました。

**ユーザーの処置:** これは、無限ループまたは待機などの UDF の問題を示している可能性があります。問題が続く場合 (すなわち、割り込みを行うことが必要になった場合に、同じエラー状態になる) は、UDF 作成者またはデータベース管理者に連絡してください。問題が修正されるまでは、その UDF を使用するべきではありません。

**sqlcode:** -431

**sqlstate:** 38504

---

**SQL0432N パラメーター・マーカが、ユーザー定義タイプ名、または参照ターゲット・タイプ名 *udt-name* を持つことができません。**

**説明:** ステートメントのパラメーター・マーカが、それが使用されているコンテキストに基づいて、ユーザー定義タイプ *udt-name* またはターゲット・タイプ *udt-name* を持つ参照タイプを持っていると判断されました。パラメーター・マーカは、データ・タイプが割り当ての一部 (INSERT の VALUES 文節、または UPDATE の SET 文節) ではないか、あるいは CAST 指定を使用したユーザー定義特殊データ・タイプまたは参照データ・タイプに明示的にキャストされていない場合は、そのデータ・タイプとしてユーザー定義タイプまたは参照タイプを持つことはできません。

ステートメントは処理できません。

**ユーザーの処置:** パラメーター・マーカのユーザー定義特殊データ・タイプまたは参照データ・



タイプに、明示キャストを使用してください。あるいは、ユーザー定義特殊データ・タイプの列に対応するソース・データ・タイプにキャストするか、参照データ・タイプの列に対応するタイプにキャストしてください。

**sqlcode:** -432

**sqlstate:** 42841

---

**SQL0433N** 値 *value* が長すぎます。

**説明:** 値 *value* が、その値をいくつかの方法でトランスフォームするために呼び出される、システム (組み込み) キャストまたは調整関数による切り捨てを要求しました。この値が使用されている場所では、切り捨てが許されていません。

トランスフォームされる値は、以下のいずれかです。

- ユーザー定義関数 (UDF) に対する引き数
- UPDATE ステートメントの SET 文節に対する入力
- 表に INSERT される値
- 他の特定のコンテキストでのキャストまたは調整関数に対する入力
- そのデータ・タイプおよび長さが反復の初期化部分で判別され、反復の反復部分で大きくできる反復して参照される列

ステートメントは失敗します。

**ユーザーの処置:** *value* が SQL ステートメントのリテラル・ストリングの場合は、その使用目的に対して長すぎます。

*value* がリテラル・ストリングでない場合は、SQL ステートメントを調べて、トランスフォームが行われる場所を判別してください。トランスフォームに対する入力が長すぎるか、またはターゲットが短すぎます。

問題を訂正して、ステートメントを再実行してください。

**sqlcode:** -433

**sqlstate:** 22001

---

**SQL0434W** 文節 *clause* のサポートされない値が、値 *value* で置き換えられました。

**説明:** 文節 *clause* に指定された値がサポートされておらず、示されたサポートされている値 *value* で置き換えられました。

**ユーザーの処置:** 選択された値が受け入れ可能であれば、変更する必要はありません。そうでない場合は、*clause* に有効な値を指定してください。

**sqlcode:** +434

**sqlstate:** 01608

---

**SQL0435N** 無効な SQLSTATE *sqlstate* が関数 RAISE\_ERROR に指定されていません。

**説明:** RAISE\_ERROR 関数に指定された SQLSTATE は、アプリケーション定義 SQLSTATE の規則にしたがっていません。

**ユーザーの処置:** 関数 RAISE\_ERROR に指定された SQLSTATE を訂正してください。

SQLSTATE は、5 文字の文字ストリングでなければなりません。これは、長さ 5 で定義された CHAR タイプ、または長さ 5 以上で定義された VARCHAR タイプでなければなりません。

SQLSTATE の値は、以下のような、アプリケーション定義 SQLSTATE の規則に従う必要があります。

- 各文字は、数字のセット ('0' から '9')、またはアクセントの付かない大文字 ('A' から 'Z') でなければなりません。
- エラー・クラスではないので、SQLSTATE クラス (最初の 2 文字) を '00'、'01'、または '02' することはできません。
- SQLSTATE クラス (最初の 2 文字) が文字 '0' から '6' または 'A' から 'H' で始まっている場合、サブクラス (最後の 3 文字) は 'I' から 'Z' までの文字で始まらなければなりません。

- SQLSTATE クラス (最初の 2 文字) が文字 '7'、'8'、'9' または 'I' から 'Z' で始まっている場合、サブクラス (最後の 3 文字) は '0' から '9' または 'A' から 'Z' のいずれでもかまいません。

**sqlcode:** -435

**sqlstate:** 428B3

---

**SQL0436N C 言語の NULL 終了文字ストリング・ホスト変数から、終了の NULL 文字が抜けています。**

**説明:** C プログラミング言語の入力ホスト変数コードの値には、ストリングの最後に NULL 終了文字が必要です。

ステートメントは処理できません。

**ユーザーの処置:** 入力ホスト変数の値が、NULL 終了文字で終了していることを確認してください。

**sqlcode:** -436

**sqlstate:** 22024

---

**SQL0437W この複合照会のパフォーマンスが最適ではない可能性があります。理由コード: *reason-code***

**説明:** 照会が複雑なために、使用できないリソースを要求しているか、または最適化境界条件が見つかったために、ステートメントのパフォーマンスが次善のパフォーマンスになる可能性があります。以下が理由コードのリストです。

- 1 メモリーの制約のため、結合列挙方式が変更されました。
- 2 照会の複雑さのため、結合列挙方式が変更されました。
- 3 オプティマイザのアンダーフローが起きました。
- 4 オプティマイザのオーバーフローが起きました。

- 5 照会最適化クラスが低すぎました。

- 6 オプティマイザが無効な統計を無視しました。

ステートメントは処理されます。

**ユーザーの処置:** 以下の 1 つ以上を行ってください。

- データベース構成ファイルのステートメント・ヒープ (stmheap) の大きさを増やしてください。(理由コード 1)
- ステートメントをより簡単な SQL ステートメントに分割してください。(理由コード 1,2,3,4)
- 述部に必要以上の応答セットを指定していないことを確認してください。(理由コード 3)
- 現在の照会最適化クラスを、低い値に変更してください。(理由コード 1,2,4)
- 照会に関連した表に対して、Runstats を発行してください。(理由コード 3,4)
- 現在の照会最適化クラスを、高い値に変更してください。(理由コード 5)
- 照会で呼び出された表およびその対応索引に対して、RUNSTATS を再発行します。つまり、表と索引の統計が一致するように、AND INDEXES ALL 文節を使用します。(理由コード 6)

**sqlcode:** +437

**sqlstate:** 01602

---

**SQL0438N アプリケーションで次の診断テキストを持つエラーが起きました: *text***

**説明:** このエラーまたは警告は、トリガーの RAISE\_ERROR 関数または SIGNAL SQLSTATE ステートメントの実行の結果として発生しました。'01' または '02' で始まる SQLSTATE 値は警告を示します。

**ユーザーの処置:** アプリケーションのドキュメンテーションを参照してください。

**sqlcode:** -438, +438

**sqlstate:** アプリケーション定義

---

**SQL0439N** ユーザー定義関数 *function-name* が関数 *source-function* によって間接的に実行され、その結果、エラー *sqlcode* が発生しました。

**説明:** 関数 *function-name* がユーザーのステートメント内で参照されていました。ただし、SOURCE 文節がこの関数の定義に使用されていたため、関数 *source-function* が実際にその関数を実施することは避けられました。( *function-name* から *source-function* へは、直接または間接的な定義パスが存在する可能性があります。) コンパイル時に、*source-function* のカプセル化プログラム(関数に代わって作動する DB2 コード) が、*sqlcode* で示されるエラーを返しました。

**ユーザーの処置:** 修正を行う前に、実際のエラー状況を理解する必要があります。 *sqlcode* の説明を調べてください。 *source-function* が組み込み関数の場合、組み込み関数がユーザーのステートメントで直接参照されたときには、*sqlcode* が問題を示します。 *source-function* がユーザー定義関数の場合は、メッセージが、引き数のいずれかまたは関数の結果を使用して、最も可能性のある問題を示します。

問題を訂正して再度試行してください。

**sqlcode:** -439

**sqlstate:** 428A0

---

**SQL0440N** 互換性のある引き数を持つ、タイプ *routine-type* の *routine-name* という名前の許可されたルーチンが見つかりませんでした。

**説明:** これは、データベース・マネージャーが参照を実行するために使用できるルーチンを見つけれない場合、ルーチン *routine-name* への参照で起こります。原因は以下のとおりです。

- *routine-name* が間違っって指定されたか、またはデータベースに存在しません。
- 修飾付き参照が行われましたが、修飾子の指定が正しくありませんでした。
- ユーザーの SQL パスに、必要な関数または方式が属しているスキーマが入っていなかったために、無修飾参照が使用されました。
- 間違った数の引き数が組み込まれていました。
- 正しい数の引き数が関数または方式参照に組み込まれていましたが、引き数の 1 つ以上のデータ・タイプが正しくありませんでした。
- ルーチンが、パッケージのバインド時に使用されたものと同じ関数 ID を持つデータベースに存在しません(静的ステートメントに該当します)。
- UPDATE ステートメントで使用されている属性割り当てに対応する mutator 方式が見つかりませんでした。属性の新しい値のデータ・タイプが、属性のデータ・タイプと同じ、またはプロモート可能なデータ・タイプではありません。
- ルーチンの起動側が、ルーチンを実行するよう許可されていません。

**ユーザーの処置:** 問題を修正して、やり直してください。これには、カタログ・アクセス、ステートメントに対する変更、ルーチンの起動側への実行特権、新しい関数の追加、および SQL パスに対する変更も含まれます。

**sqlcode:** -440

**sqlstate:** 42884

---

**SQL0441N** *function-name* 関数を持つキーワード DISTINCT または ALL の使用が無効です。

**説明:** 原因としては、以下のものが考えられます。

- キーワード DISTINCT または ALL が関数 *function-name* を参照する括弧内で検出され、関数はスカラー関数として解決されました。ス



カラー関数でのキーワード DISTINCT または ALL の使用は無効です。

- キーワード DISTINCT が、サポートされていない列関数で使用されました。これらの関数には、COVARIANCE、CORRELATION、および REGR で始まる線形回帰関数が含まれません。
- 関数がキーワード ALL または DISTINCT をサポートしている列関数であるという前提でしたが、それが解決した関数は列関数ではありませんでした。

#### ユーザーの処置:

- スカラー関数を使用されている場合は、キーワード DISTINCT または ALL を除去してください。スカラー関数には無効です。
- 関数が DISTINCT または ALL キーワードをサポートしない列関数である場合は、キーワードを除去してください。
- 列関数を使用されている場合は、関数解決に問題があります。関数パスをチェックして必要な関数がいずれかのスキーマに存在するかどうかを調べ、関数名のつづり、パラメーターの数およびタイプについて SYSFUNCTIONS カタログもチェックしてください。

エラーを訂正して、もう一度やり直してください。

**sqlcode:** -441

**sqlstate:** 42601

---

**SQL0442N** ルーチン *routine-name* の参照中にエラーが起きました。引き数の最大許容数 (90) を超えました。

**説明:** ルーチン *routine-name* への参照に指定された引き数が多すぎます。最大許容数は 90 です。

**ユーザーの処置:** ステートメントに、正しい数の引き数を使用されていることを確認して、もう一度やり直してください。

**sqlcode:** -442

**sqlstate:** 54023

---

**SQL0443N** ルーチン *routine-name* (特定名 *specific-name*) が、診断テキスト *text* とともにエラー **SQLSTATE** を返しました。

**説明:** ルーチン *routine-name* (特定名 *specific-name*) がメッセージ・テキスト *text* とともに **SQLSTATE** を DB2 に返しました。このルーチンは、ユーザー定義関数またはユーザー定義方式だと考えられます。

**ユーザーの処置:** エラーの意味を理解する必要があります。データベース管理者、またはルーチンの作成者に連絡してください。

**SYSFUN** スキーマの **IBM** 提供関数によって検出されたエラーは、すべて **SQLSTATE 38552** を戻します。メッセージのメッセージ・テキスト部分は、以下の形式となっています。

**SYSFUN:nn**

この場合の **nn** は以下の意味をもつ理由コードです。

- 01** 数値が範囲外
- 02** ゼロによる除算
- 03** 算術オーバーフローまたはアンダーフロー
- 04** 無効なデータ形式
- 05** 無効な時刻形式
- 06** 無効なタイム・スタンプ形式
- 07** タイム・スタンプ期間の無効な文字表示
- 08** 無効なインターバル・タイプ (1、2、4、8、16、32、64、128、256 のいずれかでなければならない。)
- 09** スtringが長すぎる
- 10** String関数の長さまたは位置が範囲外になっている

- 11 浮動小数点数では無効な文字表示である
- 12 メモリー不足
- 13 予期しないエラー

SYSIBM または SYSPROC スキーマの IBM 提供ルーチンによって検出されたエラー、および SYSFUN スキーマの IBM 提供プロシージャによって検出されたエラーは、すべて SQLSTATE 38552 を戻します。メッセージのメッセージ・テキスト部分には、メッセージ番号が含まれていません。このメッセージ番号は、SQLCODE (SQL0572N など)、DBA エラー・メッセージ (DBA4747 など)、またはルーチンが遭遇したエラーを示すその他の内容の場合があります。メッセージ番号のメッセージにトークンが定期的に含まれる場合は、これらのトークンの値は db2diag.log ファイルだけにあります。

**sqlcode:** -443

**sqlstate:** (ルーチンに返された SQLSTATE)

---

**SQL0444N** ルーチン *routine-name* (特定名 *specific-name*) が、アクセスできないライブラリーまたはパス *library-or-path*、関数 *function-code-id* のコードで実行されています。理由コード: *code*

**説明:** DBMS が、ルーチン *routine-name* (特定名 *specific-name*) を実行するコードの本体にアクセスを試みましたが、理由コード *code* (下のリストで説明) によってアクセスできません。ルーチンを実行するファイルは *library-or-path* で、関数は *function-code-id* で示されています。

(提供可能なトークンの合計長の制約のために、最後の 2 つのトークンが切り捨てられる場合があります。これが起きた場合は、ルーチンに定義されているライブラリーまたは全パス、および関数コード ID を判別するために、カタログ内のルーチンの定義へのアクセスが必要になる場合があります。)

**ユーザーの処置:** 与えられる理由コードは、以下の通りです。

- 1 パス名 *library-or-path* が、最大値 (255 バイト) を超えています。ルーチン定義をもっと短いパスを指定するように変更する必要があるか、または DB2 インスタンス・パス名が長すぎます。カタログ定義を調べて、どちらの場合であるかを判別してください。関数本体を、もっと短いパス名のディレクトリーに移動することが必要になる場合があります。
- 2 DB2 インスタンス・パス名が DB2 から検索できませんでした。システム管理者に連絡してください。
- 3 パス *library-or-path* が見つかりませんでした。ルーチンの作成者またはデータベース管理者に連絡してください。ルーチン定義またはルーチンそれ自身のロケーションを訂正する必要があります。
- 4 *library-or-path* でファイルが見つかりませんでした。ルーチンの作成者またはデータベース管理者に連絡してください。ルーチン定義またはルーチンのロケーションを訂正するか、ルーチンを再リンクする必要があります。

OS/2 では、この理由コードは UDF DLL 名が形式 (8.3) より長い場合に発生する可能性があります。たとえば、"abcdefgh99.dll" の名前は形式 (10.3) を持つため、その結果理由コード 4 を持つこのメッセージになります。解決法は、名前を受け入れ可能な形式、たとえば "abcdef99.dll" に変更することです。

上記に加えて、ルーチンが共有ライブラリーあるいは DLL を必要として、共有ライブラリーが見つからない (UNIX ベース・システムでは LIBPATH 環境変数で、INTEL システムでは PATH 環境変数で指定されるディレクトリーの連結を使用する) 場合、この理由コードが結果

として発生します。この理由コードの前に、複数レベルの間接原因がある場合があります。たとえば、ルーチン本体 X が見つかった場合、共有ライブラリー Y が必要となります。Y はここで検出されるものです。ただし、Y には Z が必要で、Z が見つからない場合には、SQL0444N 理由コード 4 が結果として出されます。

- 5 関数の入ったライブラリーをロードするためのメモリーが足りないか、1 つ以上のシンボルを解決できませんでした。ルーチンの作成者またはデータベース管理者に連絡して、ライブラリーが正しくリンクされているかを確認してください。外部関数などの関連する記号を解決するために、すべての必須ライブラリーが使用可能でなければなりません。メモリー不足と判別した場合は、DB2 に使えるメモリーを増やすようにシステム構成を変更する必要があります。
- 6 関数 *function-code-id* が、指定されたモジュールで見つかりませんでした。ルーチンの作成者またはデータベース管理者に連絡してください。ルーチン定義または関数それ自身を訂正する必要があります。
- 7 関数名として指定された記号 (*function-code-id*) は、指定されたライブラリーにある有効な関数の名前ではありません。ルーチンの作成者またはデータベース管理者に連絡してください。ルーチン定義または関数それ自身を訂正する必要があります。
- 8 上記の理由以外で、“ロード” システム関数が失敗しました。モジュールがリンクされていないか、または正しくリンクされていない可能性があります。
- 9 メモリーが不足しているため、*library-or-path* で識別されるライブラリーの関数名 *function-code-id* を解決でき

ません。ルーチンの作成者またはデータベース管理者に連絡して、関数の入ったライブラリーが正しくリンクされているかを確認してください。DB2 サーバーがもっと多くのメモリーを使用できるように、システム構成の変更が必要になる場合があります。

- 10 loadquery システム呼び出しが失敗しました。これは、unix ベース・システムでのみ発生し、データベース・マネージャー自体が正しくインストールされていないことの徴候です。システム管理者に連絡してください。
- 11 エージェント処理が、libdb2.a library に存在しているべき特定のデータベース・マネージャー関数を検索していましたが、見つかりませんでした。これは、unix ベース・システムでのみ発生し、データベース・マネージャーが正しくインストールされていないことの徴候です。システム管理者に連絡してください。
- 15 アクセスが拒否されています。これは、Windows NT 環境で、ルーチン定義ステートメントの EXTERNAL NAME 指定に完全パスが指定されていない場合や、関数が <db2 installation path>%function ディレクトリーに存在しないため、PATH 環境変数を使った検索が必要になる場合に発生することがあります。たとえば、PATH で、関数の含まれるディレクトリーの前に LAN ドライブが含まれており、DB2 インスタンスが SYSTEM アカウトで実行されている場合、この理由コードの結果になることがあります。ルーチン定義ステートメントに指定されている EXTERNAL NAME が完全パスで指定されていること、または <db2 installation path>%function ディレクトリーに関数が存在することを確認してください。上記の 2 つとも正しい場合に、DB2 インスタンスを実行するア

カウントが PATH 環境変数内の関数を  
含むディレクトリーの前で LAN ドライ  
ブにアクセスできないために、エラーが  
発生する場合は、DB2 サービスを構成し  
て、PATH 環境変数内のすべてのネット  
ワーク・パスへのアクセス権があるユー  
ザー・アカウントで DB2 サービスが実  
行されるようにしてください。DB2 サー  
ビスにユーザー・アカウントを割り当て  
る場合は、「概説およびインストール」  
ブックに記載されている必要なすべての  
拡張ユーザー権限をアカウントに認可し  
てください。

**その他:** 未確認システム障害が起きました。 コ  
ードを書き留めて、システム管理者に連  
絡してください。

このメッセージの情報でエラーを診断できない場  
合、システム管理者に連絡して援助を求める必要  
がある場合があります。

連合システム・ユーザー: このユーザー定義関数  
が関数テンプレートである場合 (そのため、連合  
サーバーに存在するためのコードが必要ない場  
合)、この関数がリモート・データ・ソースで評価  
されるようにするために、SQL ステートメントま  
たは統計の変更を考慮する必要があります。

**sqlcode:** -444

**sqlstate:** 42724

---

**SQL0445W** 値 *value* が切り捨てられました。

**説明:** 値 *value* が、その値をいくつかの方法で  
トランスフォームするために呼び出される、シス  
テム (組み込み) キャストまたは調整関数によっ  
て切り捨てられました。これは警告状況です。

トランスフォームされる値がルーチン (ユーザー  
定義ルーチン UDF または方式) の出力で、ルー  
チン定義に CAST FROM の指定があるため、ま  
たは UDF が他の関数から呼び出されているため  
に、結果のトランスフォームが必要なため、トラ

ansフォームが行われます。

**ユーザーの処置:** 出力が预期されていること、お  
よび切り捨てが预期しない結果の原因とならない  
ことを確認してください。

**sqlcode:** +445

**sqlstate:** 01004

---

**SQL0447W** ステートメントに、文節 *clause* を  
含む重複指定があります。

**説明:** *clause* キーワードが、ステートメントに 2  
回以上入っています。これは警告状態です。

**ユーザーの処置:** 重複が意図されたものである場  
合、または重複が障害を起こさないと判断した場  
合は、対応する必要はありません。"障害" と  
は、たとえば、他の必須キーワードが抜けている  
ことを示しています。

**sqlcode:** +447

**sqlstate:** 01589

---

**SQL0448N** ルーチン *routine-name* の定義中に  
エラーが起きました。パラメータ  
ーの最大許容数 (90 がユーザー定  
義関数用、32767 がストアード・  
プロシージャー用) を超えました。

**説明:** ルーチン *routine-name* の定義中、指定さ  
れたパラメーターが多すぎます。ルーチン定義  
ステートメントは CREATE FUNCTION、  
CREATE PROCEDURE、CREATE TYPE (方式  
定義)、または ALTER TYPE (方式定義) だと思  
われます。

**ユーザーの処置:** ステートメントを変更して、パ  
ラメーターの数を減らしてください。

**sqlcode:** -448

**sqlstate:** 54023

---

**SQL0449N** ステートメント定義ルーチン *routine-name* で、無効な形式のライブラリー / 関数識別が **EXTERNAL NAME** 文節に含まれています。

**説明:** ユーザー定義関数 (UDF)、ユーザー定義方式、またはストアード・プロシージャ *routine-name* の CREATE ステートメントの EXTERNAL NAME 文節にエラーが見つかりました。ライブラリー / 関数識別の規則は、以下の通りです。

名前の形式は '*<a>!<b>*' または '*<a>*' です。単一引用符内には、ブランクを使用できません。*<a>* は以下のいずれかです。

- ファイルの全パス識別 (たとえば、AIX では /u/slick/udfs/math、OS/2 では d:¥myfunc¥math)
- sqllib ディレクトリーの 'function' ディレクトリーに存在することが想定されるファイル名 (たとえば math)

*<b>* を省略すると、指定したファイルがリンクされたときに定義される入り口点がデフォルトになります。 *<b>* を指定すると、ルーチン本体として呼び出される *<a>* 内の入り口点 (関数) として識別されます。

**ユーザーの処置:** 問題を訂正して再度試行してください。考えられる原因として、名前の先頭または最後にブランクまたは '!' が入っている可能性があります。

**sqlcode:** -449

**sqlstate:** 42878

---

**SQL0450N** ルーチン *routine-name* (特定名 *specific-name*) が、長すぎる結果値、**SQLSTATE** 値、メッセージ・テキスト、またはスクラッチパッドを生成しました。

**説明:** ルーチン *routine-name* (特定名 *specific-name*) からの戻りにおいて、以下のいずれ

かに割り振られているバイト数を超えるバイト数を返されたことを DB2 が見つけました。

- 結果値 (ルーチン定義に基づきます)。原因としては、以下のものが考えられます。
  - 結果バッファーに移動したバイトが多すぎる。
  - データ・タイプは VARCHAR のようにデータ値を null で区切る必要があり、区切りの null は定義されたサイズの範囲内ではない。
  - DB2 はこの値の前に 2- あるいは 4- バイト長の値を予想し、この長さが結果に定義されたサイズを超過しています。
  - ルーチンが LOB ロケーターを戻しました。このロケーターに関連する LOB 値は結果の定義されたサイズを超えています。

ルーチンの結果引き数の定義は、データ・タイプの要件に適合している必要があります。詳細については、アプリケーション開発の手引きを参照してください。

- **SQLSTATE** 値 (NULL 終止符を含めて 6 バイト)
- メッセージ・テキスト (NULL 終止符を含めて 71 バイト)
- スクラッチパッドの内容 (CREATE FUNCTION で宣言された長さ)

これは許されていません。

また、スクラッチパッドの長さフィールドがルーチンによって変更された場合にも、このメッセージが返されます。

**ユーザーの処置:** データベース管理者、またはルーチンの作成者に連絡してください。

**sqlcode:** -450

**sqlstate:** 39501



---

**SQL0451N** ルーチン *routine-name* を定義しているステートメント内の *data-item* 定義に、与えられている言語で作成された非ソース関数には適切でないデータ・タイプ *type* が含まれています。

**説明:** ルーチン *routine-name* を定義しているステートメントの *data-item* 部分でエラーが起きました。ユーザーのステートメントに、無効なタイプ *type*、または無効なタイプ *type* に基づいているユーザー定義タイプ (UDT) が入っています。ルーチン定義は CREATE FUNCTION、CREATE PROCEDURE、CREATE TYPE (方式定義)、または ALTER TYPE (方式定義) だと思われれます。

*data-item* は、ステートメントの問題の領域を識別するトークンです。たとえば、"PARAMETER 2"、"RETURNS"、または "CAST FROM" です。

**ユーザーの処置:** どの状態が起きたかを判別して、適切なアクションを取ってください。適切なアクションには、以下が含まれます。

- ルーチン定義をサポートされているタイプに変更 (たとえば DECIMAL から FLOAT) してください。ここには、ルーチン本体それ自身の変更がかかわってくる可能性があり、ルーチンの使用での cast 関数の使用も関与する可能性があります。
- 新しい (適切な) ユーザー定義タイプを作成するか、または既存の UDT の定義を変更してください。

**sqlcode:** -451

**sqlstate:** 42815

---

**SQL0452N** ホスト変数 *variable-position* によって参照されているファイルにアクセスできません。理由コード:  
*reason-code*

**説明:** "nth" (この場合、n は *variable-position*) ホスト変数によって参照されるファイルへのアクセスしようとしたとき、またはアクセス中にエラーが発生しました。理由は *reason-code* です。ホスト変数の位置を判別できなかった場合、<*variable-position*> は 0 に設定されます。以下の理由コードが考えられます。

- 01 - ファイル名の長さが無効か、またはファイル名とパスのいずれか、または両方の形式が無効です。
- 02 - ファイル・オプションが無効です。以下のいずれかの値を持つ必要があります。

SQL\_FILE\_READ

- 既存のファイルから読み込む

SQL\_FILE\_CREATE

- 書き込み用に新しいファイルを作成する

SQL\_FILE\_OVERWRITE

- 既存のファイルを上書きする  
ファイルが存在しない場合は、

ファイルを作成する

SQL\_FILE\_APPEND

- 既存のファイルに追加する  
ファイルが存在しない場合は、

ファイルを作成する

- 03 - ファイルが見つかりませんでした。
- 04 - SQL\_FILE\_CREATE オプションが、既存のファイルと同じ名前を持つファイルに指定されました。
- 05 - ファイルへのアクセスが拒否されました。ユーザーが、ファイルをオープンするための許可を持っていません。
- 06 - ファイルへのアクセスが拒否されました。ファイルが非互換モードで使用中です。書き込まれるファイルが、排他モードでオープンされています。
- 07 - ファイルへの書き込み中に、ディスクがいっぱいになりました。

- 08 - ファイルの読み取り中に、予期しないファイル終わりが見つかりました。
- 09 - ファイルのアクセス中に、メディア・エラーが起きました。
- 10 - ファイルから読み取り中に、不完全または無効なマルチバイト文字が検出されました。
- 11 - ファイル・コード・ページからアプリケーションの漢字コード・ページにデータを変換中に、エラーが検出されました。

#### ユーザーの処置:

理由コード 01 の場合は、ファイル名の長さ、ファイル名およびパスを訂正してください。

理由コード 02 の場合は、有効なファイル・オプションを指定してください。

理由コード 03 の場合は、ファイルにアクセスする前に、指定したファイルが存在することを確認してください。

理由コード 04 の場合は、ファイルが必要ないときは、そのファイルを削除するか、または存在しないファイル名を指定してください。

理由コード 05 の場合は、ユーザーが、ファイルに対するアクセス (正しいファイル許可) を持っていることを確認してください。

理由コード 06 の場合は、別のファイルを使用するか、またはそのファイルにアクセスする必要があるときは、ファイルが同時にアクセスされないように、アプリケーションを変更してください。

理由コード 07 の場合は、不必要なファイルを削除して、ディスク・スペースを解放するか、または十分なディスク・スペースを持つ別のドライブ / ファイル・システムに常駐するファイルを指定してください。また、オペレーティング・システムまたはユーザー・ファイル・サイズの限界に達していないことを確認してください。ユーザーのアプリケーションのコード・ページがマルチバイト・エンコード・スキーマを使用していて、最後の文字の部分だけが書き込み可能な場合には、そのファイルに完全書式の文字だけが入っていることを確認してください。

理由コード 08 の場合は、ファイルが入力に使用されているときは、ファイル全体が読み取られる前に、そのファイルが変更されていないことを確認してください。

理由コード 09 の場合は、ファイルが常駐しているメディアのすべてのエラーを修正してください。

理由コード 10 の場合は、アプリケーションのコード・ページに基づいて有効なマルチバイト文字だけがファイルに入っていることを確認するか、あるいはファイルの内容と同じコード・ページのもとで実行している間にその要求を投入してください。

理由コード 11 の場合には、日本語 EUC などのファイルのコード・ページと、UCS-2 などのアプリケーションの漢字コード・ページとの間の文字変換サポートがインストールされていることを確認してください。

**sqlcode:** -452

**sqlstate:** 428A1

**SQL0453N** ルーチン *routine-name* を定義しているステートメントの **RETURNS** 文節で、問題が識別されています。

**説明:** ルーチン *routine-name* の結果をキャストする問題が識別されました。CAST FROM データ・タイプが、RETURNS データ・タイプにキャストできないので、問題が起きました。データ・タイプ間のキャストの詳細については、*SQL* リファレンス を参照してください。

**ユーザーの処置:** RETURNS または CAST FROM 文節を変更して、CAST FROM データ・タイプが RETURNS データ・タイプにキャストできるようにしてください。

**sqlcode:** -453

**sqlstate:** 42880

---

**SQL0454N** ルーチン *routine-name* の定義で与えられているシグニチャーが、すでにスキーマに存在する他のルーチンまたはタイプのシグニチャーに一致しています。

**説明:** 関数のシグニチャーは関数名、関数に定義されているパラメーターの数、およびパラメーターのタイプの順序リスト (パラメーター・タイプは考慮されません) によって構成されています。

方式のシグニチャーは、方式名、方式のサブジェクト・タイプ、方式に定義されているパラメーターの数、およびパラメーターのタイプの順序リスト (パラメーター・タイプは考慮されません) によって構成されています。

プロシーチャーのシグニチャーは、プロシーチャーの名前およびプロシーチャーに定義されているパラメーターの数 (データ・タイプは考慮されません) から構成されています。

この場合、以下のいずれかが考えられます。

- 作成される関数と同じシグニチャーを持つ関数またはプロシーチャー (*routine-name*) が、すでにスキーマに存在します。
- 追加される方式指定、または作成される方式本体として同じシグニチャーを持つサブジェクト・タイプの方式 (*routine-name*) が存在します。

ステートメントは処理できません。

**ユーザーの処置:** 既存のルーチンが必要な関数を提供しているかどうかを判別してください。提供していない場合、ルーチンのシグニチャーを変更する必要があります。(たとえば、ルーチン名を変更)

**sqlcode:** -454

**sqlstate:** 42723

---

**SQL0455N** ルーチン *routine-name* において、**SPECIFIC** 名として与えられているスキーマ名 *schema-name1* が、ルーチンのスキーマ名 *schema-name2* に一致していません。

**説明:** **SPECIFIC** 名が 2 つの部分の名前として指定されている場合、*schema-name1* 部分は、*routine-name* の *schema-name2* 部分と同じでなければなりません。*routine-name* の *schema-name2* 部分が、直接指定されている場合があること、またはステートメントの許可 ID に対するデフォルトを持っている場合があることに注意してください。ルーチンが方式であれば、*schema-name* は、その方式のサブジェクト・タイプのスキーマ名を指します。

**ユーザーの処置:** ステートメントを訂正して、もう一度やり直してください。

**sqlcode:** -455

**sqlstate:** 42882

---

**SQL0456N** ルーチン *routine-name* の定義において、**SPECIFIC** 名 *specific-name* はすでにスキーマに存在していません。

**説明:** ユーザーは明示 **SPECIFIC** 名 *specific-name* をルーチン *routine-name* の定義に与えましたが、この名前はスキーマ内の関数、方式、またはプロシーチャーの **SPECIFIC** 名として存在しています。

**ユーザーの処置:** 新しい **SPECIFIC** 名を選択して、もう一度やり直してください。

**sqlcode:** -456

**sqlstate:** 42710



---

**SQL0457N** システム使用のために予約されているため、名前 *name* を関数、方式、ユーザー定義データ・タイプ、または構造化データ・タイプ属性に付けることはできません。

**説明:** 示されている名前はシステム使用のために予約されているので、ユーザー定義関数、方式、ユーザー定義データ・タイプ、または構造化データ・タイプを作成できません。関数名、特定タイプ名、構造化タイプ名、あるいは属性名に対して使用されない名前には、以下のものがあります。

```
"=", "<", ">", ">=", "<=",
"&=", "&>", "&<,"
"!=", "!>", "!<," "<>,"
SOME, ANY, ALL, NOT, AND, OR,
BETWEEN, NULL, LIKE, EXISTS, IN,
UNIQUE, OVERLAPS, SIMILAR, and MATCH.
```

**ユーザーの処置:** システム使用のために予約されていない関数、方式、ユーザー定義データ・タイプ、あるいは構造化データ・タイプ属性の名前を選択してください。

**sqlcode:** -457

**sqlstate:** 42939

---

**SQL0458N** シングニチャーによるルーチン *routine-name* への参照において、一致するルーチンが見つかりませんでした。

**説明:** シングニチャーによる関数、方式、またはストアド・プロシージャー *routine-name* への参照において、一致する関数、方式、またはストアド・プロシージャーが見つかりませんでした。

パラメーターを受け入れることができるデータ・タイプが使用されている場合、タイプ・パラメーターはオプションです。たとえば CHAR(12) の場合、CHAR(12) とパラメーターを指定でき、また CHAR() とパラメーターを省略することができます。パラメーターを指定すると、DBMS は、データ・タイプとデータ・タイプ・パラメーター

に正確に一致するものだけを受け入れます。パラメーターを省略すると、DBMS データ・タイプのみ的一致するものを受け入れます。CHAR() 構文は、一致する関数が見つからなかったときにデータ・タイプを無視するよう DBMS に指示する方法を提供します。

DROP FUNCTION/PROCEDURE および COMMENT ON FUNCTION/PROCEDURE ステートメントでは、無修飾参照がステートメント許可 ID で修飾され、問題が見つかる可能性があるスキーマになることにも注意してください。

CREATE 関数の SOURCE 文節では、修飾が現在の関数パスから行われます。この場合、一致する関数はパス全体に存在しません。

関数は、COALESCE、NULLIF、NODENUMBER、PARTITION、RAISE\_ERROR、TYPE\_ID、TYPE\_NAME、TYPE\_SCHEMA、または VALUE 組み込み関数に基づくことはできません。

ステートメントは処理できません。

**ユーザーの処置:** 以下の対応を行ってください。

- 正しいスキーマが備わるように、関数パスを変更します。
- データ・タイプの指定からパラメーターを取り除きます。
- SPECIFIC 名を使用して、シングニチャーの代わりに、関数あるいはプロシージャーを参照します。

**sqlcode:** -458

**sqlstate:** 42883

---

**SQL0461N** データ・タイプ *source-data-type* を持つ値を、タイプ *target-data-type* に CAST できません。

**説明:** ステートメントに、データ・タイプ *target-data-type* にキャストされるデータ・タイプ *source-data-type* を持つ最初のオペランドが指定さ

れた CAST が入っています。このキャストはサポートされていません。

**ユーザーの処置:** キャストがサポートされるように、ソースまたはターゲットのいずれかのタイプを変更してください。事前定義されたデータ・タイプについては、「SQL リファレンス」を参照してください。ユーザー定義の異なるタイプの入ったキャストの場合、キャストは基本データ・タイプとユーザー定義の異なるタイプ間、または基本データ・タイプにプロモート可能なデータ・タイプから、ユーザー定義の異なるタイプに対してのみ行われます。

**sqlcode:** -461

**sqlstate:** 42846

---

**SQL0462W** ルーチン *routine-name* (特定名 *specific-name*) が、診断テキスト *text* とともに警告 **SQLSTATE** を返しました。

**説明:** ルーチン *routine-name* (特定名 *specific-name*) が、メッセージ・テキスト *text* とともに形式 01Hxx の **SQLSTATE** を DB2 に返しました。

**ユーザーの処置:** 警告の意味を理解する必要があります。データベース管理者、またはルーチンの作成者に連絡してください。

**sqlcode:** +462

**sqlstate:** 01Hxx

---

**SQL0463N** ルーチン *routine-name* (特定名 *specific-name*) が、診断テキスト *text* とともに無効な **SQLSTATE** *state* を返しました。

**説明:** ルーチンが返すことができる有効な **SQLSTATE** は 38xxx (エラー)、38502 (エラー)、および 01Hxx (警告) です。このルーチン *routine-name* (特定名 *specific-name*) が、無効な **SQLSTATE** *state* をメッセージ・テキスト *text* と

ともに返しました。ルーチンはエラー状態です。

**ユーザーの処置:** ルーチンを修正する必要があります。データベース管理者、またはルーチンの作成者に連絡してください。間違った **SQLSTATE** のアプリケーション重要度をルーチンの作成者から得ることもできます。

**sqlcode:** -463

**sqlstate:** 39001

---

**SQL0464W** プロシージャ *procedure-name* が *generated-nbr-results* 照会結果セットを返しました。このプロシージャは、定義された制限 *max-nbr-results* を超えています。

**説明:** *procedure-name* と名付けられたストアード・プロシージャが正常に完了しました。しかし、そのストアード・プロシージャは、プロシージャが返すことができる照会結果セットの数について定義された制限を超えていました。

**generated-nbr-results**

ストアード・プロシージャによって返された照会結果セットの数を示します。

**max-nbr-results**

ストアード・プロシージャの照会結果セット数について定義された制限を示します。

最初の *max-nbr-results* の照会結果セットのみが SQL CALL ステートメントを発行した SQL プログラムに返されます。

考えられる原因は以下のとおりです: クライアントによって課された DRDA 制限のため、ストアード・プロシージャが *generated-nbr-results* 結果セットを返せない。DRDA クライアントが、MAXRSLCNT DDM コード・ポイントによってこの制限を確立した。

**ユーザーの処置:** SQL ステートメントが成功し

ました。SQLWARN9 フィールドが 'Z' に設定されました。

**sqlcode:** +464

**sqlstate:** 0100E

---

**SQL0465N** 分離モード処理の開始、初期化、または通信を行うことができません。  
理由コード *code*

**説明:** 分離モード・ルーチン (ユーザー定義関数または方式) の実行に関係する、システム関連問題があります。問題の正確な特質は、*code* で示されています。これは、ユーザーの問題ではありません。考えられる理由コードは以下のとおりです。

ルーチン処理エラー

- 21:** 内部データまたはアプリケーション・データの初期化が失敗しました。
- 22:** シグナル・ハンドラーの登録が失敗しました。
- 23:** エージェント処理に、REQUEST QUEUE のアクセス許可を付与することに失敗しました。
- 24:** ルーチン処理の共有メモリーへの接続が失敗しました。
- 25:** REPLY QUEUE のオープンが失敗しました。
- 26:** REPLY QUEUE への書き込みが失敗しました。
- 27:** REQUEST QUEUE の作成が失敗しました。
- 28:** REQUEST QUEUE からの読み取りが失敗しました。
- 29:** ルーチン処理が消滅しました。
- 30:** ルーチン処理が USER INTERRUPT シグナルを受け取りました。

- 31:** ルーチン・モジュールをアンロードできませんでした。
- 32:** モジュールのロード / アンロードに使用される制御ブロックに対するストレージの割り振りが失敗しました。
- 33:** エージェント処理からルーチン処理に SIGINT の送信できませんでした。
- 34:** OLE ライブラリーの初期化に失敗しました。
- 35:** OLE DB 初期化サービス・コンポーネントの初期化に失敗しました。
- 40:** ルーチン処理で内部エラーが発生しました。

エージェント処理エラー

- 41:** ルーチン処理を spawn できませんでした。
- 42:** REPLY QUEUE の作成が失敗しました。
- 43:** REPLY QUEUE からの読み取りが失敗しました。
- 44:** REQUEST QUEUE のオープンが失敗しました。
- 45:** REQUEST QUEUE への書き込みが失敗しました。
- 47:** ルーチン処理に、UDFP 共有メモリー・セットに対するアクセス許可を付与できませんでした。
- 48:** ルーチン処理に REPLY QUEUE に対するアクセス許可を付与できませんでした。
- 49:** モジュールのロード / アンロードに使用される制御ブロックに対するストレージの割り振りが失敗しました。
- 50:** ルーチン・コードまたはエージェント・コードの実行中にエージェント処理が消滅しました。

**51:** 非分離ルーチン・コードの実行中に、エージェント処理が `USER INTERRUPT` を受け取りました。

**60:** ルーチン処理で内部エラーが発生しました。

**ユーザーの処置:** データベースまたはシステム管理者に連絡してください。

**sqlcode:** -465

**sqlstate:** 58032

---

**SQL0466W** プロシージャ *procedure-name* が、ストアード・プロシージャから *number-results* の結果セットを返しました。

**説明:** このメッセージは、`CALL SQL` ステートメントの発行の結果として返されます。これは、ストアード・プロシージャ *procedure-name* に、関連する *number-results* の結果セットがあることを示しています。

ステートメントは正しく完了しました。

**ユーザーの処置:** 必要ありません。

**sqlcode:** +466

**sqlstate:** 0100C

---

**SQL0467W** プロシージャ *procedure-name* には、別の結果セットが含まれていません。合計 *max-nbr-results* の結果セットがあります。

**説明:** このメッセージはカーソルのクローズの結果として戻されます。これは、ストアード・プロシージャ *procedure-name* に別の結果セットが存在し、次の結果セットでカーソルがオープンされたことを示しています。ストアード・プロシージャには、合計 *max-nbr-results* の結果セットがあります。

ステートメントは正しく完了しました。

**ユーザーの処置:** 必要ありません。フェッチは、

次の結果セットで続行できます。

**sqlcode:** +467

**sqlstate:** 0100D

---

**SQL0469N** パラメーター・モード (`IN`、`OUT`、または `INOUT`) は、特定名 *specific-name* のプロシージャ *procedure-name* のパラメーター (パラメーター番号 *number*、名前 *parameter-name*) には無効です。

**説明:** 以下のエラーのいずれかが発生していると考えられます。

- `SQL` プロシージャのパラメーターが `OUT` と宣言され、プロシージャ本体の入力として使用されている
- `SQL` プロシージャのパラメーターが `IN` と宣言され、プロシージャ本体で変更されている

**ユーザーの処置:** パラメーターの属性を `INOUT` に変更するか、またはプロシージャ内でのパラメーターの使用を変更してください。

**sqlcode:** -469

**sqlstate:** 42886

---

**SQL0470N** ユーザー定義ルーチン *routine-name* (特定名 *specific-name*) には、パスされなかった引き数 *argument* の `NULL` 値があります。

**説明:** ルーチンには `NULL` 標識を渡さないパラメーター・スタイルで定義された `NULL` 値のある入力引き数が入っているか、あるいはこのパラメーターのデータ・タイプは `NULL` 値をサポートしません。

**ユーザーの処置:** ルーチンが `NULL` 値で呼び出される場合は、パラメーター・スタイルと入力タイプが `NULL` 値を受け入れ可能であることを確認してください。関数の場合、`"RETURNS`

NULL ON NULL INPUT” で関数を作成することができます。

**sqlcode:** -470

**sqlstate:** 39004

---

**SQL0471N** 理由 *reason-code* のためにルーチン *name* の呼び出しが失敗しました。

**説明:** DB2 Universal Database for OS/390 サーバーでルーチン *name* が呼び出されました。DB2 理由コード *reason-code* で説明されている条件のため、ルーチンの呼び出しは失敗しました。ステートメントは処理できません。MVS システム・コンソール上でこのエラーを説明する DSNX9xx メッセージを表示できます。

**ユーザーの処置:** DB2 Universal Database for OS/390 サーバーのドキュメンテーションを調べて、DB2 理由コードで説明されている条件を正してください。

**sqlcode:** -471

**sqlstate:** 55023

---

**SQL0472N** 1 つ以上のカーソルが、関数または方式 *routine-name* (特定名 *specific-name*) によってオープンされたままでした。

**説明:** 関数または方式 *routine-name* (特定名 *specific-name*) は、呼び出しているステートメントが完了する前にすべてのカーソルをクローズしませんでした。呼び出しているステートメントが完了する前に、関数および方式はすべてのカーソルをクローズする必要があります。

**ユーザーの処置:** 関数または方式の開発者に問い合わせてください。呼び出しているステートメントが完了する前にすべてのカーソルがクローズされていることを確認するために、関数または方式を再書き込みする必要があります。

**sqlcode:** -472

**sqlstate:** 24517

---

**SQL0473N** システム事前定義タイプと同じ名前を持つユーザー定義データ・タイプは作成できません。

**説明:** 作成されるデータ・タイプの名前に、システム事前定義データ・タイプと同一名であるか、または BOOLEAN である無修飾名があります。これは許可されていません。区切り文字を追加しても、有効な名前にはなりません。

ステートメントを処理できませんでした。

**ユーザーの処置:** 他の ID を使用するように、ステートメントを修正してください。

**sqlcode:** -473

**sqlstate:** 42918

---

**SQL0475N** SOURCE 関数の結果タイプ *type-1* は、ユーザー定義関数 *function-name* の RETURNS タイプ *type-2* にはキャストできません。

**説明:** ソースであるユーザー定義関数 (UDF) の CREATE を有効にするには、ソース関数の結果タイプ (*type-1*) が作成される関数の RETURNS タイプ (*type-2*) にキャスト可能でなければなりません。これらのデータ・タイプ間でサポートされるキャストはありません。データ・タイプ間のキャストの詳細については、SQL リファレンスを参照してください。

**ユーザーの処置:** RETURNS データ・タイプまたは SOURCE 関数識別を変更して、SOURCE 関数の結果タイプが RETURNS データ・タイプにキャストできるようにしてください。

**sqlcode:** -475

**sqlstate:** 42866

---

**SQL0476N** ルーチン *function-name* への参照がシグニチャーなしで行われましたが、ルーチンはそのスキーマでユニークではありません。

**説明:** 関数あるいはストアド・プロシージャに対するシグニチャーのない参照は許されていますが、示された関数あるいはストアド・プロシージャ *function-name* がそのスキーマ内でユニークでなければならないのに、ユニークではありませんでした。ルーチンが方式であれば、シグニチャーのない参照が許可されていますが、ここで示されている方式はデータ・タイプとしてユニークではありません。

DROP FUNCTION/PROCEDURE および COMMENT ON FUNCTION/PROCEDURE ステートメントでは、無修飾参照がステートメント許可 ID で修飾され、問題が見つかる可能性があるスキーマになることに注意してください。CREATE 関数の SOURCE 文節では、修飾が現在の関数パスから行われます。この場合は、この名前を持つ関数の入ったパスの最初のスキーマが、同じ名前前の別の関数を持っています。

連合システム・ユーザー: パススルー・セッションでは、ステートメントが CREATE FUNCTION MAPPING ステートメントの場合、このエラーは、関数のマッピングを 1 つのリモート関数から複数のローカル関数に作成しようとしたことを示します。

**ユーザーの処置:** 以下のいずれかを行って、参照を訂正してください。

- シグニチャーを完了する
- 目的のルーチンの SPECIFIC 名を使用する
- SQL パスを変更する

その後で再試行してください。

**sqlcode:** -476

**sqlstate:** 42725

---

**SQL0478N** オブジェクト・タイプ *object-type1* は、このオブジェクトに從属するタイプ *object-type2* のオブジェクト *object-name* が存在するため、**DROP** または **REVOKE** できません。

**説明:** この *object-type1* には從属関係が存在するため、要求された DROP または REVOKE は処理できません。タイプ *object-type2* (オブジェクト *object-name* はその一例) のオブジェクトに定義されている制約從属関係があります。

この從属関係は間接的である可能性があります。つまり、指定されたオブジェクトは、ドロップされるオブジェクトに從属しているか、REVOKE の影響を受けるオブジェクトに從属する別のオブジェクトに從属しています。

**例:**

- 関数 F1 のソースが関数 F2 である
- ビュー V1 が F1 を使って定義されている
- F2 での F1 の直接依存性、および F2 での V1 の間接依存性のために、F2 をドロップする試みが失敗する

**ユーザーの処置:** このオブジェクトに対する從属関係を除去してから、要求を再発行してください。

**sqlcode:** -478

**sqlstate:** 42893

---

**SQL0480N** プロシージャ *procedure-name* は呼び出されていません。

**説明:** ASSOCIATE LOCATORS ステートメントで識別されているプロシージャがアプリケーション処理内で呼び出されていないか、またはプロシージャは呼び出されていますが、ステートメントの前に暗黙的または明示的なコミットが行われました。

**ユーザーの処置:** CALL ステートメントにプロシージャ名を指定するための構文が、



ASSOCIATE LOCATORS ステートメントの構文と同じになるよう、ステートメントを訂正してください。修飾されていない名前がプロシージャールを呼び出すために使用されている場合、別のステートメントに 1 部分の名前も指定する必要があります。ステートメントを出し直してください。

**sqlcode:** -0480

**sqlstate:** 51030

---

**SQL0481N** **GROUP BY** 文節に、*element-2* 内でネストされている *element-1* が含まれています。

**説明:** 以下のネストのタイプは、GROUP BY 文節内で許可されていません:

- CUBE CUBE、ROLLUP、または GEL 内
- ROLLUP CUBE、ROLLUP、または GEL 内
- () CUBE、ROLLUP、または GEL 内
- GROUPING SET GROUPING SET、CUBE、ROLLUP、または GEL 内

GEL は GROUP BY 文節の構文図でグループ化式リストとして表示されるエレメントを表しています。

いくつかのインスタンスでは、値 “---” が *element 2* について示されます。この場合、“---” は CUBE、ROLLUP、GROUPING SET、または GEL のいずれかを表しています。

ステートメントは処理できません。

**ユーザーの処置:** ネストを除去する GROUP BY 文節を修正してください。

**sqlcode:** -481

**sqlstate:** 428B0

---

**SQL0483N** ユーザー定義関数 *function-name* の **CREATE** のパラメーターの数が、**SOURCE** 関数のパラメーターの数と一致しません。

**説明:** 別の関数がソースであるユーザー定義関数 *function-name* の作成が試みられました。以下のいずれかの状態になっています。

- SOURCE 文節が、関数名 (入力パラメーター・リスト) を使用してソース関数を識別しましたが、リスト内のタイプの数が、作成される関数のパラメーターの数と異なります。
- SOURCE 文節が、異なる構文を使用してソース関数を識別しましたが、その関数のタイプの数が、作成される関数のパラメーターの数と異なります。

**ユーザーの処置:** SOURCE 関数と作成される関数のパラメーターの数は、同じでなければなりません。SOURCE 関数の識別を、以下のように変更する必要があります。

- 入力パラメーター・リストを修正する。
- 適切な関数を識別するように、関数名または関数特定名を修正する。

関数の正しい解決が行われるように、関数パスの修正が必要になる場合もあります。

**sqlcode:** -483

**sqlstate:** 42885

---

**SQL0486N** **BOOLEAN** データ・タイプは、現在内部的にのみサポートされています。

**説明:** ステートメントの 1 つ以上のデータ・タイプが BOOLEAN です。これは、DB2 の現在のバージョンではサポートされていません。

**ユーザーの処置:** データ・タイプを変更して、ステートメントの再実行依頼を行ってください。

**sqlcode:** -486

**sqlstate:** 42991

---

**SQL0487N** ルーチン *routine-name* (特定名 *specific-name*) が、SQL ステートメントを実行しようと試みました。

**説明:** ルーチンの本体を実行するためのプログラムは、SQL ステートメントの実行を許可されていません。このルーチン *routine-name* (特定名 *specific-name*) には、SQL ステートメントが入っています。

**ユーザーの処置:** SQL ステートメントを除去した後で、プログラムを再コンパイルしてください。ルーチンを定義しているステートメントへの指定を許可されている SQL のレベルを調べてください。

**sqlcode:** -487

**sqlstate:** 38001

---

**SQL0489N** **SELECT** または **VALUES** リスト項目内の関数 *function-name* が、**BOOLEAN** 結果を作成しました。

**説明:** 関数 *function-name* が、ブール結果を戻す述部としての使用を定義されています。このような結果は、選択リストでは無効です。

ステートメントは処理できません。

**ユーザーの処置:** 関数名を訂正するか、関数の使用を除去してください。

**sqlcode:** -489

**sqlstate:** 42844

---

**SQL0490N** SQL ステートメントに直接指定されている数値 *number* は、このコンテキストにおける許容値 (*minval,maxval*) の範囲外です。

**説明:** 数値 (*number*) は、指定されたコンテキストで無効です。このコンテキストでの最小の許容値は、*minval* です。このコンテキストで許容される最大値は、*maxval* です。n は *minval* と

*maxval* で指定された範囲内でなければなりません (*minval* =< n => *maxval*)。

**ユーザーの処置:** ステートメントで、値 n を有効な値に変更してください。

**sqlcode:** -490

**sqlstate:** 428B7

---

**SQL0491N** ルーチン *routine-name* の定義には **RETURNS** 文節、さらに **EXTERNAL** 文節 (他の必須キーワードとともに)、**RETURN** ステートメント、または **SOURCE** 文節のいずれかが必要です。

**説明:** 必要な文節が、ルーチン *routine-name* の定義に見つかりません。EXTERNAL が指定されている場合は、以下のいずれかの文節も指定されている必要があります:

LANGUAGE、PARAMETER

STYLE、DETERMINISTIC または NOT

DETERMINISTIC、および EXTERNAL ACTION または NO EXTERNAL ACTION。

**ユーザーの処置:** 足りない文節を追加した後で、もう一度やり直してください。

**sqlcode:** -491

**sqlstate:** 42601

---

**SQL0492N** ユーザー定義関数 *function-name*、パラメーター番号 *number* の **CREATE** に問題があります。**SOURCE** 関数との不一致になる場合があります。

**説明:** 関数 *function-name* の位置 *number* にあるパラメーターがエラーであるため、CREATE が実行できません。ソース関数の位置 *number* にあるパラメーターが、作成中の関数の対応するパラメーターにキャストできません。

**ユーザーの処置:** 適切な処置には以下が含まれます。



- 別のソース関数を識別します。
- 作成中の関数のパラメーターのデータ・タイプを変更して、ソース関数のデータ・タイプがこのデータ・タイプにキャストできるようにします。

**sqlcode:** -492

**sqlstate:** 42879

**SQL0493N** ルーチン *routine-name* (特定名 *specific-name*) が、構文的または数值的に無効な日付、時刻、またはタイム・スタンプ値を返しました。

**説明:** ユーザー定義関数 (UDF) または方式 *routine-name* (特定名 *specific-name*) が、無効な日付、時刻、またはタイム・スタンプ値を返しました。

構文的に無効な日付の値の例は '1994-12\*25' で、'\*' を '.' にする必要があります。数的に無効な時刻の値の例は '11.71.22' で、時間には 71 分は存在しません。

**ユーザーの処置:** ルーチンを修正する必要があります。データベース管理者、またはルーチンの作成者に連絡してください。

**sqlcode:** -493

**sqlstate:** 22007

**SQL0495N** コスト・カテゴリー *cost-category* で、*estimate-amount1* プロセッサ秒の見積もられたプロセッサ・コスト (*estimate-amount2* サービス単位) が、*limit-amount* サービス単位のリソース制限エラーしきい値を超えています。

**説明:**

動的 INSERT、UPDATE、DELETE、または SELECT SQL ステートメントの準備の結果、リソース限定表 (RLST) で指定されたエラーしきい

値を超えたコスト見積もりになりました。

DB2 のコスト・カテゴリー値が "B" になった場合もこのエラーが発行され、RLST の RLF\_CATEGORY\_B 列で指定されたデフォルトのアクションがエラーを発行します。

**estimate\_amount1**

準備された

INSERT、UPDATE、DELETE または SELECT ステートメントが実行された場合のコスト見積もり (プロセッサ秒)。

**estimate\_amount2**

準備された

INSERT、UPDATE、DELETE または SELECT ステートメントが実行された場合のコスト見積もり (サービス単位)。

**cost-category**

この SQL ステートメントについての DB2 のコスト・カテゴリー。使用可能な値は A または B です。

**limit-amount**

RLST の RLFASUERR 列で指定されたエラーしきい値 (サービス単位)。

動的 INSERT、UPDATE、DELETE、または SELECT ステートメントの準備に失敗しました。

**ユーザーの処置:** コスト・カテゴリー値が "B" であるためにこの SQLCODE が返された場合は、ステートメントがパラメーター・マーカースを使用しているか、参照される表と列について使用できない統計が存在する可能性があります。管理者が、参照された表でユーティリティ `RUNSTATS` を実行したことを確認してください。また、ステートメントが実行されるときに UDF が呼び出されるか、INSERT、UPDATE、または DELETE ステートメントについては、変更された表にトリガーが定義されている可能性もあります。このステートメントについて `DSN_STATEMNT_TABLE` または `IFCID 22` レコードをチェックして、この SQL ステートメントがコスト・カテゴリー "B" になった理由を判別

してください。問題を変えられない場合、または統計を入手できない場合は、管理者に問い合わせ、RLST の RLF\_CATEGORY\_B 列の値をステートメントを実行できるようにする "Y" に変更するか、エラーではなく警告を返す "W" に変更してください。

SQL ステートメントがプロセッサ・リソースを多く使用しすぎていることが警告の原因である場合は、ステートメントが効率良く実行されるために書き直してみてください。あるいは、管理者に RLST のエラーしきい値を増やすように依頼してください。

**sqlcode:** -495

**sqlstate:** 57051

---

**SQL0499N** カーソル *cursor-name* は、プロシージャ *procedure-name* から、この結果セットまたは別の結果セットに対して、すでに割り当てられています。

**説明:** 結果セットにカーソルを割り当てようとしたが、複数のカーソルがすでにプロシージャ

## SQL1000 - SQL1099

---

**SQL1000N** *alias* は、有効なデータベース別名ではありません。

**説明:** コマンドまたは *api* に指定された別名が、有効ではありません。別名は、1 から 8 文字 (MBCS を使用している国ではバイト) でなければならず、すべての文字を、データベース・マネージャーの基本文字セットから使用する必要があります。

コマンドは処理されません。

**ユーザーの処置:** 正しい別名を指定して、コマンドを再発行してください。

— *procedure-name* に割り振られています。

**ユーザーの処置:** ターゲットの結果セットが以前にカーソルに割り当てられているか、判別してください。複数のカーソルがプロシージャ *procedure-name* に割り振られている場合、1 つのカーソルだけが、ストアード・プロシージャの結果セットを処理するために使用されるようにしてください。

**sqlcode:** -499

**sqlstate:** 24516

---

**SQL1001N** *name* は、有効なデータベース名ではありません。

**説明:** コマンドに指定されたデータベース名の構文が無効です。データベース名は、1 から 8 文字でなければならず、すべての文字をデータベース・マネージャーの基本文字セットから使用する必要があります。

コマンドは処理されません。

**ユーザーの処置:** 正しいデータベース名を指定して、コマンドを再発行してください。

**sqlcode:** -1001

**sqlstate:** 2E000

---

**SQL1002N** *drive* は、有効なドライブではありません。

**説明:** コマンドに指定されたドライブが無効です。ドライブは、データベースまたはデータベース・ディレクトリーが存在する、ディスク・ドライブまたはハード・ディスク・パーティションを示す 1 文字 (A から Z) です。

コマンドは処理されません。

**ユーザーの処置:** 正しいドライブを指定して、コマンドを再発行してください。

---

**SQL1003N** 構文が正しくないためにパスワードが有効ではないか、またはパスワードが、指定されたデータベースのパスワードと一致しません。

**説明:** パスワードの長さは 18 文字以下です。ただし、パスワードが APPC 会話でチェックされる場合は、8 文字以下でなければなりません。

**ユーザーの処置:** パスワードが許容限界より長いことを確認してください。

**sqlcode:** -1003

**sqlstate:** 28000

---

**SQL1004C** コマンドの処理に十分なストレージが、ファイル・システムにありません。

**説明:** コマンドを処理するには、指定されたファイル・システムのストレージが十分ではありません。

Windows 環境のパーティション・データベース環境では、パーティション・データベース・グループのノードにはそれぞれ、CREATE DATABASE コマンドを成功させるために使用できる (使用可能スペースが含まれている) まったく同一の物理ハード・ディスク指定 (文字) がある必要があります。物理ハード・ディスクは、データベース・マネージャー構成で指定されます。

DFTDBPATH がブランクのままの場合、デフォル

トは DB2 がインスタンス所有マシン (db2 インストール・パス) にインストールされているハード・ディスクとなります。

コマンドは処理されません。

**ユーザーの処置:** 別のファイル・システムを選択するか、またはデータベース・マネージャー関数にスペースを与えるために、指定したファイル・システムからデータベース・ファイル以外のいくつかのファイルを削除してください。

OS/2 および Windows 環境のパーティション・データベース環境では、以下のステップにしたがってください。

- どのハード・ディスク指定 (文字) が必要か決定する。ドライブ文字は、エラー・メッセージで指定されています。
- データベース・パーティションのどのノードが問題を起しているか判別する。インスタンス所有ノードの db2diag.log ファイルでこの情報を検索することができます。
- 問題を起しているそれぞれのノードで、ドライブ上の問題を訂正するか、またはデータベース・マネージャー構成でのドライブ指定を変更して、パーティション・データベース・グループのノードごとに、同じドライブが使用可能 (十分なスペースがある) となるようにしてください。
- コマンドを再発行してください。

---

**SQL1005N** データベース別名 *name* は、すでにローカル・データベース・ディレクトリーまたはシステム・データベース・ディレクトリーのどちらかに存在しています。

**説明:** 指定された別名は、すでに使用されています。catalog database コマンドに別名を指定しないと、データベース名が別名として使用されます。データベースの作成時には、別名はデータベース名と同じになります。

このエラーは、すでにシステム・データベース・

ディレクトリーに別名が存在するのに、`catalog database` コマンドを出したために起きた可能性があります。

`create database` コマンドでこのエラーが起こった場合は、以下の可能性が考えられます。

- 別名が、すでにシステム・データベース・ディレクトリーおよびローカル・データベース・ディレクトリーに存在する。
- 別名が、すでにシステム・データベース・ディレクトリーに存在しますが、ローカル・データベース・ディレクトリーには存在しない。
- 別名が、すでにローカル・データベース・ディレクトリーに存在しますが、システム・データベース・ディレクトリーには存在しない。

**ユーザーの処置:** `catalog database` コマンドの場合は、システム・データベース・ディレクトリーから別名をアンカタログした後で、オリジナル・コマンドを再発行するか、または異なる別名でデータベースをカタログしてください。

`create database` コマンドの場合は、上記の 3 つの状況に応じて、以下の操作を行ってください。

- 別名を使用しているデータベースをドロップしてください。その後で、オリジナル・コマンドを再発行してください。
- 別名をアンカタログしてください。その後で、オリジナル・コマンドを再発行してください。
- 別名をシステム・データベース・ディレクトリーにカタログしてください。同じ別名を使用しているデータベースをドロップしてください。その後で、オリジナル・コマンドを再発行してください。

---

**SQL1006N** アプリケーションのコード・ページ *code-page* が、データベースのコード・ページ *code-page* と一致していません。

**説明:** データベースのアクティブ・コード・ページが、作成時のアクティブ・コード・ページと異

なっているため、アプリケーションがデータベースに接続できませんでした。

コマンドは処理されません。

**ユーザーの処置:** 現在のアプリケーション・プログラムを終了して、オペレーティング・システムに戻ってください。処理のコード・ページを変更して、アプリケーション・プログラムを再始動してください。

---

**SQL1007N** 表スペースのオブジェクト用のページの検索でエラーが起きました。

**説明:** 表スペースの壊れた内部データベース・ページ、または内部ロジック・エラーが存在します。

**ユーザーの処置:** オブジェクトまたは表スペースの使用を続けしないでください。オブジェクトおよび表スペースをチェックするために、IBM サービスに連絡してください。

**sqlcode:** -1007

**sqlstate:** 58034

---

**SQL1008N** 無効な表スペース ID です。

**説明:** 指定された表スペース ID が存在しません。現在の最大表スペース ID より大きいか、または表スペースがドロップされています。

**ユーザーの処置:** データベースの使用を続けしないでください。エラー・ログの診断情報を保管して、IBM サービスに連絡してください。

**sqlcode:** -1008

**sqlstate:** 58036

---

**SQL1009N** このコマンドは無効です。

**説明:** このコマンドは、クライアント専用ワークステーション上、またはリモート・データベースに対する発行ではサポートされていません。このようなコマンドの例は、ローカル・データベースのカタログです。

コマンドは処理されません。

**ユーザーの処置:** 解決策は以下のとおりです。

- クライアント専用でないワークステーション、またはデータベースが常駐するワークステーション上で、指定したコマンドを発行してください。
- データベースが正しくカタログされていることを確認してください。
- 別のコマンドを実行してください。

---

**SQL1010N** *type* は無効なタイプ・パラメーターです。

**説明:** Database Environment コマンドに指定されたタイプが、有効ではありません。これは、間接データベースの場合は '0'、リモート・データベースの場合は '1' でなければなりません。

さらに、AIX、OS/2、Windows NT、Windows 95 プラットフォームの場合は、DCE グローバル名を持つデータベースのタイプを '3' にすることができます。

コマンドは処理されません。

**ユーザーの処置:** 有効なタイプを指定して、コマンドを再発行してください。

---

**SQL1011N** 間接項目に対する CATALOG DATABASE コマンドに、パスが指定されていません。

**説明:** CATALOG DATABASE コマンドが、間接項目に対して発行されましたが、パスが指定されていません。間接項目は、データベースが常駐するパスを指定する必要があります。

コマンドは処理されません。

**ユーザーの処置:** 必要なパスを指定してコマンドを再発行するか、またはタイプを変更してください。

---

**SQL1012N** リモート項目に対する CATALOG DATABASE コマンドに、ノード名が指定されていません。

**説明:** リモート項目に対する CATALOG DATABASE コマンドに、*nodename* パラメーターが指定されていません。リモート項目は、データベースのノード名を指定する必要があります。コマンドは処理されません。

**ユーザーの処置:** *nodename* パラメーターまたは別のタイプを指定して、コマンドを再発行してください。

---

**SQL1013N** データベース別名またはデータベース名 *name* が見つかりませんでした。

**説明:** コマンドに指定されたデータベース名または別名が、既存のデータベースではないか、あるいはそのデータベースが (クライアントまたはサーバー) データベースでディレクトリーに見つかりませんでした。

**ユーザーの処置:** 示されたデータベース名が、システム・データベース・ディレクトリーに存在することを確認してください。データベース名がシステム・データベース・ディレクトリーに存在しない場合は、データベースが存在しないか、またはデータベース名がカタログされていないかのどちらかです。

データベース名がシステム・データベース・ディレクトリーに存在し、項目タイプが INDIRECT の場合は、データベースが指定したローカル・データベース・ディレクトリーに存在することを確認してください。項目タイプが REMOTE の場合は、データベースが存在し、サーバー・ノードのデータベース・ディレクトリーにカタログされていることを確認してください。

AT NODE 文節をとまなう CREATE DATABASE の場合、データベース名が INDIRECT の入力タイプおよび -1 と同等ではないカタログ・ノード番号をとまなうシステム・データベース・ディレ

クトリーにあることを確認してください。

連合システム・ユーザー: 上記に加えて、SYSCAT.SERVERS に指定されているデータベース名がすべて有効であるかどうかをチェックしてください。SYSCAT.SERVERS 項目で指定されたデータベースが存在しない項目は訂正してください。

**sqlcode:** -1013

**sqlstate:** 42705

---

**SQL1014W** 走査中のディレクトリー、ファイル、またはリストには、これ以上の項目はありません。

**説明:** ディレクトリー、ファイル、またはリストのスキャンは完了されます。

**ユーザーの処置:** アクションは必要ありません。

---

**SQL1015N** 前のセッションが正常に終了しなかったため、データベースを再始動する必要があります。

**説明:** 前のセッションが異常終了 (たとえば、電源障害) したために、データベースを再始動する必要があります。

コマンドは処理されません。

**ユーザーの処置:** このメッセージ (SQLCODE) を受け取ることにより、アプリケーションが警告メッセージを出力して、データベースをリカバーする時間を要求することができます。データベースを再始動するには、RESTART DATABASE コマンドを発行してください。パーティション・データベース・サーバー環境では、このコマンドはすべてのノードに対して発行されます。

**sqlcode:** -1015

**sqlstate:** 55025

---

**SQL1016N** CATALOG NODE コマンドで指定された *local\_lu* 別名 *name* は無効です。

**説明:** CATALOG NODE コマンドで指定されたローカル LU ( *local\_lu* ) 別名は使用できません。ローカル LU 別名はローカル SNA LU 別名で、1 から 8 文字でなければならず、ブランク文字は使用できません。

コマンドは処理されません。

**ユーザーの処置:** 別名が使用可能な LU 名であることを確認してください。名前に使用されている文字を確認してください。有効な LU 名を使用して、コマンドを再発行してください。

---

**SQL1017N** CATALOG NODE コマンドで指定されたモード・パラメーター *mode* は無効です。

**説明:** CATALOG NODE コマンドで指定されている *mode* パラメーターは使用できません。

*mode* パラメーターが、コミュニケーション・マネージャーがセッションのセットアップに使用する通信プロファイルを識別しています。モードは 1 から 8 文字でなければなりません。有効な文字は大文字または小文字の A から Z、0 から 9、#、@、および \$ です。先頭の文字は英字でなければなりません。システムは小文字を大文字に変更します。

コマンドは処理されません。

**ユーザーの処置:** 名前が使用可能なモード名であることを確認してください。名前に使用されている文字を確認してください。正しいモードを指定して、コマンドを再発行してください。

---

**SQL1018N** CATALOG NODE コマンドで指定されたノード名 *name* は、すでに存在しています。

**説明:** CATALOG NODE コマンドの *nodename* パラメーターに指定されているノード名は、すで



にこのファイル・システムのノード・ディレクトリーにカタログされています。

コマンドは処理されません。

**ユーザーの処置:** *nodename* パラメーターが正しくタイプされている場合は、処理を継続してください。

ノードのカタログ情報が無効な場合は、カタログされているノードをアンカタログした後で、コマンドを再発行してください。ノードのカタログ情報が有効な場合は、新しいノード名を定義して、それを使用してコマンドを再実行してください。

---

#### SQL1019N コマンドに指定されたノード名 *name* が無効です。

**説明:** コマンドに指定されたノード名が無効です。ノード名は 1 から 8 文字で、すべての文字はデータベース・マネージャーの基本文字セットから使用する必要があります。指定されたノード名を、ローカル・インスタンス名と同一名にすることはできません。

コマンドは処理されません。

**ユーザーの処置:** DB2INSTANCE 環境変数の値を表示して、ノード名がローカル・インスタンス名と同一名でないことを確認してください。

UNIX オペレーティング・システムでは、以下のコマンドを入力して DB2INSTANCE 環境変数を表示してください。

```
echo $DB2INSTANCE
```

Windows および OS/2 オペレーティング・システムでは、以下のコマンドを入力して

DB2INSTANCE 環境変数を表示してください。

```
echo %DB2INSTANCE%
```

正しいノード名を指定して、コマンドを再発行してください。

---

#### SQL1020C ノード・ディレクトリーがいっぱい です。

**説明:** ノード・ディレクトリーには、これ以上項目が入りません。

コマンドは処理されません。

**ユーザーの処置:** ノード・ディレクトリー内の不要な項目をアンカタログしてください。

---

#### SQL1021N UNCATALOG NODE コマンドで 指定されたノード名 *name* が、見 つかりません。

**説明:** コマンドに指定された *nodename* が、ノード・ディレクトリーに見つかりませんでした。

コマンドは処理されません。

**ユーザーの処置:** *nodename* パラメーターが正しい場合は、ノードがすでにアンカタログされている可能性があり、処理を継続することができません。その他の場合は、正しいノード名を使用して、コマンドを再発行してください。

---

#### SQL1022C コマンドの処理に使用できる、十分な メモリーがありません。

**説明:** コマンドの処理に使用できるランダム・アクセス・メモリー (RAM) が不足しています。

リモート・プロシージャが呼び出された場合は、そのリモート・プロシージャが、許容最大値 (4K) より大きなローカル可変スペースを使用する可能性があります。

ステートメントにユーザー定義関数 (UDF) が含まれている場合は、*ASLHEAPSZ* データベース・マネージャー構成パラメーターによってコントロールされているメモリー・セットが、使用可能なメモリーより大きい可能性があります。

コマンドは処理されません。

**ユーザーの処置:** アプリケーションを停止してください。解決策は以下のとおりです。

- CONFIG.SYS ファイルの MEMMAN NO SWAP、NO MOVE オプションを、SWAP、MOVE に変更してください。
- バックグラウンド処理を終了してください。
- UDF が失敗しているステートメントに関連する場合、ASLHEAPSZの入ったメモリーの割り振りを定義する構成パラメーターの値を減らしてください。
- もっと多くのランダム・アクセス・メモリー (RAM) をインストールしてください。
- リモート・プロシージャが呼び出された場合は、そのリモート・プロシージャが 4K 以下のローカル可変スペースを使用することを確認してください。
- リモート・データ・サービスを使用している場合は、アプリケーションごとに少なくとも 1 ブロックが使用されるので、サーバーとクライアント構成でリモート・データ・サービスのヒープ・サイズ (rsheapsz) を増やしてください。
- OS/2 システムの場合は、CONFIG.SYS ファイルの MEMMAN ステートメントの PROTECT を NOPROTECT に変更してください。それによって、もっと多くのメモリー・スペースがアプリケーションで使用可能になりますが、OS/2 の特定の保護機能が使用できなくなります。詳細と、これが使用環境に適切かどうかを判別するには、OS/2 のドキュメンテーションを参照してください。
- OS/2 システムの場合は、min\_priv\_mem データベース・マネージャー構成パラメーターの値を増やしてください。これにより、データベース・マネージャーが db2start 時に、もっと多くの専用メモリー・スペースを予約できます。  
注: これはバージョン 2 以前の DB2 のリリースにのみ適用されます。

**sqlcode:** -1022

**sqlstate:** 57011

---

### SQL1023C 通信での会話が失敗しました。

**説明:** 通信による会話でエラーが起きました。

コマンドは処理されません。

**ユーザーの処置:** オリジナル・コマンドを再発行してください。エラーが続く場合は、通信管理者に連絡してください。

連合システム・ユーザー: この状態はデータ・ソースでも検出できます。

**sqlcode:** -1023

**sqlstate:** 08001

---

### SQL1024N データベース接続が存在しません。

**説明:** データベースに対する接続が存在しません。SQL CONNECT ステートメントが先に実行されるまでは、他の SQL ステートメントは処理できません。

コマンドは処理されません。

**ユーザーの処置:** データベースからの接続中にエラーが起きた場合は、処理が続けられます。エラーが他の SQL ステートメントで起きた場合は、SQL CONNECT ステートメントを発行して、コマンドまたはステートメントの再実行依頼を行ってください。

**sqlcode:** -1024

**sqlstate:** 08003

---

### SQL1025N データベースがまだアクティブになっているために、データベース・マネージャーが停止されませんでした。

**説明:** データベース・マネージャーの制御下にあるデータベースに接続されているアプリケーションが存在する場合、あるいはデータベースがアクティブにされている場合には、stop database manager コマンドを処理することはできません。

何も処理されません。



**ユーザーの処置:** 通常、アクションは必要ありません。データベース・マネージャーを停止するには、すべての活動アプリケーションを、使用しているすべてのデータベースから切り離す必要があります。この代りに、FORCE コマンドを使用してアプリケーションを強制的に切断し、DEACTIVATE コマンドを使用してデータベースを非活動化することができます。

---

**SQL1026N** データベース・マネージャーはすでにアクティブになっています。

**説明:** start database manager コマンドは、すでに処理されています。

コマンドは処理されません。

**ユーザーの処置:** コマンドがすでに処理されているので、アプリケーションは処理を継続することができます。

---

**SQL1027N** ノード・ディレクトリーが見つかりません。

**説明:** ノード・ディレクトリーが見つからないため、list node directory コマンドは処理できません。

コマンドは処理されません。

**ユーザーの処置:** 適切なパラメーターを使用して、CATALOG NODE コマンドを実行し、現在のコマンドを再発行してください。

---

**SQL1029N** CATALOG NODE コマンドで指定された partner\_lu 別名 name は無効です。

**説明:** CATALOG NODE コマンドで指定される partner\_lu 別名が、指定されていないか、無効な文字を含んでいます。partner\_lu 別名はパートナー SNA LU 別名で、1 から 8 文字でなければならず、ブランク文字は使用できません。

コマンドは処理されません。

**ユーザーの処置:** partner\_lu のつづりエラーを調

べてください。別名が使用可能な LU 名であることを確認してください。別名に使用されている文字を確認してください。正しい partner\_lu を使用してコマンドを再実行依頼してください。

---

**SQL1030C** データベース・ディレクトリーがいっぱいです。

**説明:** システム・データベース・ディレクトリーまたはローカル・データベース・ディレクトリーは、これ以上項目を保留できません。

コマンドは処理されません。

**ユーザーの処置:** ディレクトリー内の不要な項目をアンカタログしてください。ローカル・データベース・ディレクトリーがいっぱいの場合は、別のファイル・システムに新しいデータベースを作成してください。

---

**SQL1031N** 指定されたファイル・システムには、データベース・ディレクトリーが見つかりません。

**説明:** システム・データベース・ディレクトリーまたはローカル・データベース・ディレクトリーを見つめることができませんでした。データベースが作成されていないか、または正しくカタログされていません。

コマンドは処理されません。

**ユーザーの処置:** データベースが、正しいパスの指定で作成されていることを確認してください。Catalog Database コマンドは、データベースが常駐するディレクトリーを指定するパス・パラメーターを持っています。

**sqlcode:** -1031

**sqlstate:** 58031

---

**SQL1032N** start database manager コマンドが実行されていません。

**説明:** start database manager コマンドが処理されていません。データベース・マネージャー、SQL

ステートメント、ユーティリティを発行する前に、このコマンドを処理する必要があります。

コマンドは処理されません。

**ユーザーの処置:** データベース・マネージャー開始コマンドを実行して、現在のコマンドを再発行してください。

複数の論理ノードを使用している場合、DB2NODE 環境変数が正しく設定されているか、確認してください。アプリケーションのノードを示す DB2NODE 環境は接続を試行します。DB2NODE をこのアプリケーションと同じホストで定義されたノードの 1 つのノードに設定する必要があります。

**sqlcode:** -1032

**sqlstate:** 57019

---

### SQL1033N データベース・ディレクトリーは現在使用中のためアクセスできません。

**説明:** データベース・ディレクトリーは現在更新中のためアクセスできません。また、データベース・ディレクトリーが何らかの理由ですでにアクセスされている場合は、更新のためにディレクトリーにアクセスすることはできません。こうした状態は、システム・データベース・ディレクトリーまたはローカル・データベース・ディレクトリーのいずれでも発生します。

コマンドは処理されません。

**ユーザーの処置:** アクセスが完了するまで待ち、このコマンドを再発行してください。

**sqlcode:** -1033

**sqlstate:** 57019

---

### SQL1034C データベースが壊れています。アプリケーションは、データベースから切断されました。データベースを処理するすべてのアプリケーションは停止しました。

**説明:** データベースに障害が起きました。リカバーするまで使用できません。データベースに接続されていたすべてのアプリケーションは切断され、データベース上でアプリケーションを実行していたすべての処理が停止しました。

コマンドは処理されません。

**ユーザーの処置:** RESTART DATABASE コマンドを実行して、データベースをリカバーしてください。RESTART コマンドが整合性のために失敗した場合は、データベースのバックアップからのリストアが必要になる可能性があります。パーティション・データベース・サーバー環境で、syslog をチェックしてバックアップからデータベースをリストアする前のノードまたは通信障害のために RESTART コマンドが失敗しているか検索してください。失敗している場合、データベース・マネージャーが起動していて、すべてのノード間で通信が使用可能であることを確認してから、再始動コマンドを再度実行してください。

ロールフォワード処理中にこのエラーが起きた場合は、データベースをバックアップからリストアして、ロールフォワードをもう一度実行する必要があります。

パーティション・データベース環境では、RESTART データベース・コマンドがノードごとに実行されていることに注意してください。データベースがすべてのノードで再始動しているか確認するには、以下のコマンドを使用します。

```
db2_all db2 restart database
<database_name>
```

このコマンドは、すべての未確定トランザクションが解決したことを確認するには、数回実行する必要があります。

サンプル・データベースをインストールしている場合は、それをドロップしてサンプル・データベースを再インストールしてください。

**sqlcode:** -1034

**sqlstate:** 58031

---

**SQL1035N** このデータベースは現在使用中です。

**説明:** 下記のいずれかの状態が存在します。

- 排他使用が要求されましたが、そのデータベースは、(同一処理内の)他のユーザーにより、共有データベースとしてすでに使用されています。
- 排他使用が要求されましたが、そのデータベースはすでに排他データベースとして使用されています。(2つの異なった処理が、同じデータベースをアクセスしようとしています)
- データベースへの接続の最大数に達しました。
- データベースが、他のシステムの他のユーザーによって使用されています。
- 活動化 / 非活動化 データベースが処理中です。

コマンドは処理されません。

**ユーザーの処置:** 解決策は以下のとおりです。

- データベースが使用中でなくなったときに、コマンドを再発行してください。
- 現在のユーザーに一致するように許可を変更するか、またはデータベースが使用中でなくなるまで待ってください。
- データベースが排他使用でなくなるまで待ってください。
- 他のシステムの他のユーザーが、データベースから切断するまで待ってください。

**sqlcode:** -1035

**sqlstate:** 57019

---

**SQL1036C** データベースのアクセス中に、入出力エラーが発生しました。

**説明:** 少なくとも1つのデータベース・ファイルに、以下の入出力エラーが起きました。

- システムが、データベース・ファイルのオープン、読み取り、または書き込みを行うことができません。
- システムが、データベース・ファイルまたはデータベースのディレクトリーを作成中にエラーが起きたために、データベースを作成することができません。
- システムが、データベース・ファイルまたはデータベースのディレクトリーを削除中にエラーが起きたために、データベースをドロップすることができません。
- システムがデータベース・ファイルまたはデータベースのディレクトリーを作成または削除中に割り込みを受信したため、システムはデータベースを作成できません。
- システムは接続中にデータベース・サブディレクトリーまたはデータベース構成ファイルを位置指定することはできません。

データベースは使用することができません。

**ユーザーの処置:** データベースを処理しているときにエラーが起きた場合は、コマンドを再発行してください。エラーが続く場合は、データベースをバックアップ・バージョンからリストアしてください。

CREATE DATABASE または DROP DATABASE コマンドを処理しているときにエラーが起きた場合は、後続の CREATE DATABASE または DROP DATABASE コマンドが、正常に処理されなかった CREATE DATABASE または DROP DATABASE コマンドが残したファイルおよびディレクトリーを削除しようとしています。

サンプル・データベースをインストールしている場合は、それをドロップしてサンプル・データベースを再インストールしてください。

データベースへの接続を試行中にエラーが発生した場合は、トレースを入手し、データベースのリカバーができるか、弊社サポート担当者に連絡してください。

**sqlcode:** -1036

**sqlstate:** 58030

---

#### **SQL1037W** ノード・ディレクトリーが空です。

**説明:** ノード・ディレクトリーの内容を読み取ろうとしましたが、項目が存在しません。

**ユーザーの処置:** アクションは必要ありません。

**sqlcode:** +1037

**sqlstate:** 01606

---

#### **SQL1038C** ノード・ディレクトリーにアクセス中に、入出力エラーが起きました。

**説明:** 入出力エラーのために、ノード・ディレクトリーがアクセスできませんでした。

コマンドは処理されません。

**ユーザーの処置:** コマンドを再発行してください。エラーが続く場合は、ノード・ディレクトリー (sqllib ディレクトリーの下 `sqlnodir`) を取り除いて、もう一度ノード名をネットワークにカタログしてください。

**sqlcode:** -1038

**sqlstate:** 58031

---

#### **SQL1039C** データベース・ディレクトリーにアクセス中に入出力エラーが発生しました。

**説明:** システム・データベース・ディレクトリーまたはローカル・データベース・ディレクトリーにアクセスできません。このエラーは、システムがデータベースをカタログまたはアンカタログしているときのみでなく、ディレクトリーにカタログされているデータベースにアクセスしているときにも起きる可能性があります。

コマンドは処理されません。

**ユーザーの処置:** 解決策は以下のとおりです。

- ディスケット・システムでエラーが起きた場合には、正しいディスクがドライブに挿入されていて、その使用準備ができていることを確認してください。ディスクが書き込み保護になっていないことを確認してください。
- データベース・ディレクトリーが損傷を受けている場合には、カタログされているデータベースをバックアップ・バージョンからリストアして、カタログしてください。

サンプル・データベースをインストールしている場合は、それをドロップしてサンプル・データベースを再インストールしてください。

**sqlcode:** -1039

**sqlstate:** 58031

---

#### **SQL1040N** データベースに接続されているアプリケーションの数が、すでに最大数に達しています。

**説明:** データベースに接続されているアプリケーションの数が、データベースの構成ファイルに定義されている最大値と同じです。

コマンドは処理されません。

**ユーザーの処置:** ほかのアプリケーションがデータベースから切断するまでお待ちください。もっと多くのアプリケーションを並行して実行する必要がある場合は、`maxappls` の値を増やしてください。すべてのアプリケーションがデータベースから切断されて、データベースが再始動されると、新しい値が反映されます。

**sqlcode:** -1040

**sqlstate:** 57030

---

**SQL1041N 並列処理できる最大数のデータベースが、すでに始動しています。**

**説明:** アプリケーションが、非アクティブ・データベースを始動しようとしたのですが、アクティブ・データベースの数が、システム構成ファイルに定義されている最大値に達しています。

コマンドは処理されません。

**ユーザーの処置:** データベースの 1 つが非アクティブになるのを待ってください。より多くのデータベースを同時にアクティブにする必要がある場合は、*numdb* の値を増やしてください。新しい値は、次のデータベース・マネージャーが正常に始動した後に反映されます。

**sqlcode:** -1041

**sqlstate:** 57032

---

**SQL1042C 予期しないシステム・エラーが発生しました。**

**説明:** システム・エラーが発生しました。このエラーの理由としては、データベース・マネージャーが正しくインストールされていないか、または環境が正しくセットアップされていない可能性があります。

OS/2 で、データベース・マネージャーの始動中の場合、このエラーの最も一般的な理由は、NET.ACC ファイルが壊れていることです。

コマンドは処理されません。

**ユーザーの処置:** データベース・マネージャーの始動中に OS/2 でエラーが起き、NET.ACC ファイルが疑わしい場合は、システムの NET.ACC ファイルを、OS/2 インストール・ディスクット用の DB2 のディスクット 1 にある NET.ACC で置き換えてください。

データベースへの接続中にエラーが起きた場合は、トレースを取得 (以下に指示があります) して、IBM サポートに連絡してください。

問題が上記の助言に当てはまらない場合は、シス

テムの日付と時刻が正しく設定されていること、およびシステムが十分なメモリを持ち、スワッピング / ページング・スペースが使用可能なことを確認してください。

現在のコマンドを再発行してください。

エラーが続く場合は、データベース・マネージャーの再始動を行ってください。

まだエラーが続く場合は、データベース・マネージャーを再インストールしてください。

トレースがアクティブの場合は、オペレーティング・システムのコマンド・プロンプトから、独立トレース機能呼び出ししてください。以下の情報を用意して、テクニカル・サービス担当者に提供してください。

必要な情報:

- 問題記述
- SQLCODE またはメッセージ番号
- SQLCA の内容 (ある場合)
- トレース・ファイル (可能であれば)

連合システム・ユーザー: 必要な場合は、要求を拒否しているデータ・ソースに問題を分離してください。(障害の起きたデータ・ソースを識別する手順については「問題判別の手引き」を参照してください。) 問題がデータ・ソースのものである場合、データ・ソースのための問題判別手順にしたがってください。

**sqlcode:** -1042

**sqlstate:** 58004

---

**SQL1043C データベース・マネージャーが、システム・カタログを初期化できませんでした。エラー *error* が返されました。**

**説明:** システム・カタログの初期化時に、CREATE DATABASE コマンドが失敗しました。

**ユーザーの処置:** このメッセージのメッセージ番

号 (SQLCODE) とエラーを記録してください。

トレースがアクティブの場合は、オペレーティング・システムのコマンド・プロンプトから、独立トレース機能呼び出ししてください。次に、以下の情報を用意して、テクニカル・サービス担当者に提供してください。

- 環境: アプリケーション
- 必要な情報:
  - 問題記述
  - SQLCODE またはメッセージ番号とエラー ID
  - SQLCA の内容 (ある場合)
  - トレース・ファイル (可能であれば)

---

#### SQL1044N 割り込みによって、処理が取り消されました。

**説明:** ユーザーが割り込みキー・シーケンスを押した可能性があります。

処理は停止します。

連合システム・ユーザー: この状態はデータ・ソースでも検出できます。

**ユーザーの処置:** 割り込みを処理するために、処理を継続してください。

サンプル・データベースをインストールしている場合は、それをドロップしてサンプル・データベースを再インストールしてください。

データベース・マネージャーを開始している場合は、DB2 コマンドを実行する前に、db2stop を発行してください。

**sqlcode:** -1044

**sqlstate:** 57014

---

#### SQL1045N データベースが、正しくカタログされていないために見つかりません。

**説明:** データベース・ディレクトリーの間接項目が、別の非 HOME 項目を指しています。

コマンドは処理されません。

**ユーザーの処置:** directory scan コマンドを使用して、すべての関連するデータベース・ディレクトリーの項目を確認してください。

**sqlcode:** -1045

**sqlstate:** 58031

---

#### SQL1046N 許可 ID が無効です。

**説明:** ログオン時に指定された許可 ID が、データ・ソースまたはデータベース・マネージャーのどちらかに無効です。以下のいずれかが起きました。

- Windows プラットフォームでは 30 文字、その他のプラットフォームでは 8 文字を超える文字が許可 ID に入っています。
- 許可 ID に、許可 ID では無効な文字が入っています。有効な文字は A から Z、a から z、0 から 9、#、@ および \$ です。
- 許可 ID が、PUBLIC または public です。
- 許可 ID が、SYS、sys、IBM、ibm、SQL、または sql で始まっています。
- 許可はデータ・ソース特定命名規則に違反しています。

コマンドは処理されません。

**ユーザーの処置:** 有効な許可 ID を使用して、ログオンしてください。

連合システム・ユーザー: 必要な場合は、要求を拒否しているデータ・ソースに問題を分離します。(障害の起きたデータ・ソースを識別する手順については「トラブルシューティング・ガイド」を参照してください。) 次に、そのデータ・ソースに有効な許可 ID を使用してください。

**sqlcode:** -1046

**sqlstate:** 28000



---

**SQL1047N** このアプリケーションはほかのデータベースにすでに接続されています。

**説明:** アプリケーションは他のデータベースに接続されているときには、データベースを作成できません。

すでに他のデータベースに接続している場合は、データベースにバインド・ファイルをバインドすることは許されていません。

コマンドは処理されません。

**ユーザーの処置:** アクティブ・データベースから切断して、コマンドを再発行してください。

---

**SQL1048N** **START USING DATABASE** または **CONNECT TO** コマンドの **use** パラメーター *parameter* は無効です。共有アクセスの場合は 'S'、排他使用の場合は 'X'、単一ノードで排他使用の場合は 'N' を使用してください。DB2 Connect 接続の場合、S のみがサポートされます。N は MPP 構成でのみサポートされます。

**説明:** START USING DATABASE または CONNECT TO コマンドの *use* パラメーターは、共有使用時には 'S'、排他使用時には 'X' でなければなりません。DB2 Connect を使用してデータベースに接続している場合には、共有アクセスのみが許可されています。これらの値に対して、SQLENV.H ファイルで略号が提供されています。

コマンドは処理されません。

**ユーザーの処置:** 有効な *use* パラメーター (略号も使用できます) を使用して、コマンドを再発行してください。

---

**SQL1049C** アプリケーション状態がエラーです。データベース接続は失われました。

**説明:** データベースへの接続が切り離されました。

コマンドは処理されません。

**ユーザーの処置:** CONNECT RESET ステートメントを発行してください。

---

**SQL1050N** このデータベースはホーム・データベースなので、アンカタログできません。

**説明:** UNCATALOG DATABASE コマンドに指定されたデータベースは、ホーム・データベースです。データベースがドロップされたときに、ディレクトリー項目がドロップされるため、ホーム・データベースはアンカタログできません。

コマンドは処理されません。

**ユーザーの処置:** データベース名が正しく指定されている場合は、処理を継続してください。

---

**SQL1051N** データベース・ディレクトリーのパス *path* が、存在しません。

**説明:** コマンドの *database directory* パラメーター、またはデータベース・ディレクトリー項目に指定されたパスが無効です。その名前のファイル・システムは存在しません。

コマンドは処理されません。

**ユーザーの処置:** データベース・ディレクトリーの正しいパスを指定して、コマンドを再発行してください。

**sqlcode:** -1051

**sqlstate:** 57019

---

**SQL1052N** データベース・パス *path* が存在しません。

**説明:** コマンドの *path* パラメーターに指定されたパスが無効です。その名前のパスは存在しません。

Windows 環境のパーティション・データベース環境では、パーティション・データベース・グループのノードにはそれぞれ、CREATE DATABASE コマンドを成功させるために使用できる (使用可能スペースが含まれている) まったく同一の物理ハード・ディスク指定 (文字) がある必要があります。物理ハード・ディスクは、データベース・マネージャー構成で指定されます。DFTDBPATH がブランクのままの場合、デフォルトは DB2 がインスタンス所有マシン (db2 インストール・パス) にインストールされているハード・ディスクとなります。

コマンドは処理されません。

**ユーザーの処置:** 正しいデータベース・パスを指定して、コマンドを再発行してください。

OS/2 および Windows 環境のパーティション・データベース環境では、以下のステップにしたがってください。

- どのハード・ディスク指定 (文字) が必要か決定する。ドライブ文字は、エラー・メッセージで指定されています。
- データベース・パーティションのどのノードが問題を起しているか判別する。インスタンス所有ノードの db2diag.log ファイルでこの情報を検索することができます。
- 問題を起しているそれぞれのノードで、ドライブ上の問題を訂正するか、またはデータベース・マネージャー構成でのドライブ指定を変更して、パーティション・データベース・グループのノードごとに、同じドライブが使用可能 (十分なスペースがある) となるようにしてください。
- コマンドを再発行してください。

---

**SQL1053N** 割り込みは、すでに処理されています。

**説明:** システムは現在割り込みを処理しているため、他の割り込みは受け付けられません。

割り込み要求が無視されます。

**ユーザーの処置:** 現在の割り込み処理が完了するのを待って、コマンドを再発行してください。

---

**SQL1054N** COMMIT が進行中なので、割り込みはできません。

**説明:** このシステムは、現在 COMMIT を処理しています。ユーザーが割り込みキー・シーケンスを入力しました。

割り込み要求が無視されます。

連合システム・ユーザー: この状態はデータ・ソースでも検出できます。

**ユーザーの処置:** COMMIT 完了するまで待って、割り込み要求の再実行依頼を行ってください。

---

**SQL1055N** ROLLBACK が進行中なので、割り込みはできません。

**説明:** このシステムは、現在 ROLLBACK を処理しています。ユーザーが割り込みキー・シーケンスを入力しました。

割り込み要求が無視されます。

連合システム・ユーザー: この状態はデータ・ソースでも検出できます。

**ユーザーの処置:** ROLLBACK が完了するまで待って、割り込み要求の再実行依頼を行ってください。



---

**SQL1056N** すでに 8 つのデータベース・ディレクトリー走査がオープンしていません。

**説明:** この処理の 8 つのデータベース・ディレクトリー走査は、すでにオープンされています。8 つ以上の走査のオープンは許されていません。

コマンドは処理されません。

**ユーザーの処置:** 1 つ以上の CLOSE DIRECTORY SCAN コマンドを実行して、オリジナル・コマンドを再発行してください。

**sqlcode:** -1056

**sqlstate:** 54029

---

**SQL1057W** システム・データベース・ディレクトリーが空です。

**説明:** システム・データベース・ディレクトリーの内容を読み取ろうとしましたが、項目が存在しませんでした。

**ユーザーの処置:** アクションは必要ありません。

**sqlcode:** +1057

**sqlstate:** 01606

---

**SQL1058N** **Directory Scan** コマンドの **handle** パラメーターが無効です。

**説明:** Directory Scan コマンドに指定されている **handle** パラメーターが有効ではありません。

**handle** は、OPEN DIRECTORY SCAN または OPEN NODE DIRECTORY SCAN コマンドから返されたものでなければなりません。

コマンドは処理されません。

**ユーザーの処置:** 有効な **handle** パラメーターを指定して、コマンドを再発行してください。

---

**SQL1059N** **Open Scan** コマンドが出されていないために、**Get Next** コマンドが処理できません。

**説明:** スキャンをオープンする前に、directory scan コマンドが実行されました。

コマンドは処理されません。

**ユーザーの処置:** OPEN DIRECTORY SCAN または OPEN NODE DIRECTORY SCAN コマンドを実行して、現在のコマンドを再発行してください。

---

**SQL1060N** ユーザー *authorization-ID* は **CONNECT** 特権を持っていません。

**説明:** 示された許可 ID には、データベースにアクセスするための **CONNECT** 特権が与えられていません。データベースに接続する前に、**CONNECT** 特権を付与される必要があります。

連合システム・ユーザー: この状態はデータ・ソースでも検出できます。

コマンドは処理されません。

**ユーザーの処置:** データベースのシステム管理者またはデータベース管理者に連絡して、許可 ID に対して **GRANT CONNECT** 要求を発行してください。コマンドを再発行してください。

連合システム・ユーザー: 必要な場合は、要求を拒否しているデータ・ソースに問題を分離します。(障害の起きたデータ・ソースを識別する手順については「トラブルシューティング・ガイド」を参照してください。) 次に、そのデータ・ソースに付与された特権が正しいものであるかどうかを確認してください。

**sqlcode:** -1060

**sqlstate:** 08004

---

---

**SQL1061W** RESTART は正常に終了しましたが、データベースに対する未確定トランザクションが存在しています。

**説明:** 未確定トランザクションが見つかったことを除いて、RESTART は正常に終了しました。データベースは使用可能ですが、データベースへの最後の接続をドロップする前に、未確定のトランザクションを解決しないと、次にデータベースを使用する前に、再び RESTART が必要になります。

**ユーザーの処置:** 未確定トランザクションを解決するか、またはデータベースを使用するときは常に、データベースの RESTART を準備してください。(XA/DTP 環境で) データベースを使用していたトランザクション・マネージャー (TM) が使用可能な場合は、管理者が TM を使用して、未確定トランザクションを解決する必要があります。または、十分に注意して、管理者が CLP を使用して、トランザクションを経験的に完了することもできます。

パーティション・データベース・サーバー環境では、RESTART データベース・コマンドがノードごとに実行されていることに注意してください。データベースをすべてのノードで再始動するには、以下のコマンドを使用してください。

```
db2_all db2 restart database
<database_name>
```

このコマンドを実行すると、すべてのノードが使用中の場合、未確定トランザクションを解決します。

このコマンドは、すべての未確定トランザクションが解決したことを確認するには、数回実行する必要があります。

---

**SQL1062N** データベース・パス *path* が見つかりませんでした。

**説明:** コマンドで指定されたデータベース *path* パラメーターが存在しません。パスが指定され

ていない場合は、システム構成ファイルに定義されているデフォルト・パスが使用され、それが存在しませんでした。

Windows 環境のパーティション・データベース環境では、パーティション・データベース・グループのノードにはそれぞれ、CREATE DATABASE コマンドを成功させるために使用できる (使用可能スペースが含まれている) まったく同一の物理ハード・ディスク指定 (文字) がある必要があります。物理ハード・ディスクは、データベース・マネージャー構成で指定されます。DFTDB がブランクのままの場合、デフォルトは DB2 がインスタンス所有マシン (db2 インストール・パス) にインストールされているハード・ディスクとなります。

コマンドは処理されません。

**ユーザーの処置:** パスまたはデフォルト・パスを調べて、システムに存在することを確認してください。コマンドを再発行してください。

OS/2 および Windows 環境のパーティション・データベース環境では、以下のステップにしたがってください。

- どのハード・ディスク指定 (文字) が必要か決定する。ドライブ文字は、エラー・メッセージで指定されています。
- データベース・パーティションのどのノードが問題を起こしているか判別する。インスタンス所有ノードの db2diag.log ファイルでこの情報を検索することができます。
- 問題を起こしているそれぞれのノードで、ドライブ上の問題を訂正するか、またはデータベース・マネージャー構成でのドライブ指定を変更して、パーティション・データベース・グループのノードごとに、同じドライブが使用可能 (十分なスペースがある) となるようにしてください。
- コマンドを再発行してください。

---

**SQL1063N** データベース・マネージャーの始動処理が正常に終了しました。

**説明:** データベース・マネージャーを始動させるコマンドが、正常に終了しました。

---

**SQL1064N** データベース・マネージャーの停止処理が正常に終了しました。

**説明:** データベース・マネージャーを停止させるコマンドが、正常に終了しました。

---

**SQL1065W** データベースは作成されましたが、リスト *list-name* 内の 1 つ以上のバインド・ファイルでエラーが起きました。以下のファイルがバインドされませんでした: *list*

**説明:** 1 つ以上のユーティリティが、データベースにバインドされませんでした。リスト・ファイル *list-name* には、バインド・ファイルのリストが入っています。 *list* の番号は、リスト・ファイル内のバインドされていないファイルの相対位置を示します。

リストされているユーティリティ・バインド・ファイルは、新しく作成されたデータベースへはバインドされません。

**ユーザーの処置:** 示されたユーティリティは、データベースにバインドできる可能性があります。バインド・プログラム呼び出しでは、*format* オプションは使用できません。

---

**SQL1066N** DB2START 処理が正常に終了しました。IPX/SPX プロトコル・サポートは正常に始動されませんでした。

**説明:** IPX/SPX プロトコル・サポートの始動に失敗しました。リモート・クライアントは IPX/SPX を使用して、サーバーに接続することができません。以下の理由が考えられます。

- ワークステーションが、NetWare ファイル・サーバーにログインしていません。

- ワークステーションが、NetWare ファイル・サーバー・バインダリーにオブジェクトを作成するための許可を持っていません。
- ネットワークの別のデータベース・マネージャーが、データベース・マネージャー構成ファイルに指定されているオブジェクト名と同じ名前を使用しています。

**ユーザーの処置:** ワークステーションが NetWare ファイル・サーバーにログインしており、ファイル・サーバーのバインダリーに、オブジェクトを作成するための十分な許可を持っていることを確認してください。SUPERVISOR またはそれと同等の ID でログインする必要があります。また、データベース・マネージャー構成ファイルに指定されているオブジェクト名が、ネットワーク内のすべてのデータベース・マネージャーに対してユニークであることも確認してください。訂正を行って、DB2STOP を実行した後で、もう一度 DB2START を実行してください。

問題が続く場合は、オペレーティング・システム・コマンド・プロンプトに、DB2TRC ON -L 0X100000 をタイプしてください。DB2START を再実行した後で、コマンド・プロンプトに、"DB2TRC DUMP ファイル名" をタイプして、トレース情報を保管してください。トレースをオフにするには、DB2TRC OFF をタイプしてください。トレース情報をサービス・コーディネーターに渡してください。

---

**SQL1067N** DB2STOP 処理が正常に終了しました。IPX/SPX プロトコル・サポートは正常に停止されませんでした。

**説明:** IPX/SPX プロトコル・サポートの停止に失敗しました。以下の理由が考えられます。

- ワークステーションが、NetWare ファイル・サーバーにログインしていません。
- ワークステーションが、NetWare ファイル・サーバー・バインダリーのオブジェクトを削除するための権限を持っていません。

**ユーザーの処置:** ワークステーションが NetWare ファイル・サーバーにログインしており、ファイル・サーバーのバインダリーのオブジェクトを削除するための十分な権限を持っていることを確認してください。 SUPERVISOR またはそれと同等の ID でログインする必要があります。 訂正を行って、もう一度 DB2STOP を実行してください。

問題が続く場合は、オペレーティング・システム・コマンド・プロンプトに、DB2TRC ON -L 0X100000 をタイプしてください。 DB2STOP を再実行した後で、コマンド・プロンプトに、"DB2TRC DUMP ファイル名" をタイプして、トレース情報を保管してください。 トレースをオフにするには、DB2TRC OFF をタイプしてください。トレース情報をサービス・コーディネーターに渡してください。

---

**SQL1068N CONNECT または ATTACH ステートメントのユーザー ID *user-ID* を所有しているドメインが、DB2DOMAINLIST 環境変数に定義されていません。**

**説明:** CONNECT TO または ATTACH TO ステートメントのユーザー ID が、DB2DOMAINLIST 環境変数に定義されているドメインに属していません。

**ユーザーの処置:** DB2SET コマンドを使用して、そのユーザー ID を所有しているドメインの名前を DB2DOMAINLIST 環境変数に指定してください。

**sqlcode:** -1068

**sqlstate:** 08004

---

**SQL1069N データベース *name* は、ホーム・データベースではありません。**

**説明:** データベースがローカル・データベースではありません。 ローカル・データベースは、システム・データベース・ディレクトリーに間接的にカタログされており、この項目が、同じノード

のローカル・データベース・ディレクトリーのホーム項目を示しています。 リモート・データベースはドロップできません。

コマンドは処理されません。

**ユーザーの処置:** 示されたデータベース名が正しくない場合は、正しいデータベース名を使用して、コマンドを再発行してください。 示されたデータベース名が正しく、そのデータベース名をデータベース・ディレクトリーから除去したい場合は、UNCATALOG DATABASE コマンドを使用してください。

---

**SQL1070N *database name* パラメーターのアドレスが無効です。**

**説明:** アプリケーション・プログラムが、*database name* パラメーターに無効なアドレスを使用しました。 そのアドレスが割り振られていないバッファーを指しているか、またはそのバッファー内の文字ストリングに NULL 終止符がありません。

コマンドは処理されません。

**ユーザーの処置:** アプリケーション・プログラムを修正して、有効なアドレスを使用し、入力ストリングが NULL で終了するようにしてください。

---

**SQL1071N *database alias* パラメーターのアドレスが無効です。**

**説明:** アプリケーション・プログラムが、このパラメーターに無効なアドレスを使用しました。 そのアドレスが割り振られていないバッファーを指しているか、またはそのバッファー内の文字ストリングに NULL 終止符がありません。

コマンドは処理されません。

**ユーザーの処置:** 有効なアドレスがアプリケーション・プログラムで使用され、入力ストリングが null で終了していることを確認してください。

**SQL1072C** データベース・マネージャー・リソースが、不整合状態にあります。データベース・マネージャーが異常終了したか、あるいは他のアプリケーションによるシステム・リソースの使用が、データベース・マネージャーが使用しているシステム・リソースと競合している可能性があります。システム・リソースのクリーンアップが必要になる場合があります。

**説明:** データベース・マネージャー・リソースが不整合状態にあるために、要求が失敗しました。以下の場合に、これが起きる可能性があります。

- DB2 が異常終了した (UNIX ベース・システムの場合、たとえば、処理が `stop database manager` コマンドではなく、`"kill"` コマンドで終了した場合に、これが起きる可能性があります)。
- 他のアプリケーションまたはユーザーが、データベース・マネージャー・リソースを取り除いた (UNIX ベース・システムの場合、たとえば、十分な特権を持つユーザーが、データベース・マネージャーが所有している処理間通信 (IPC) リソースを、誤って `"ipcrm"` コマンドで除去した可能性があります)。
- 他のアプリケーションによるシステム・リソースの使用法が、データベース・マネージャーによるシステム・リソースの使用法と矛盾している (UNIX ベース・システムの場合、別のアプリケーションが、データベース・マネージャーで IPC リソースの作成に使用したキーと同じキーを使用している可能性があります)。
- データベース・マネージャーの別のインスタンスが、同じリソースを使用している可能性がある。2 つのインスタンスが異なったファイル・システム上にあって、複数の `sqllib` ディレクトリーが同じ `i` ノードを偶然もっている (`i` ノードは IPC キーを獲得するために使用される) 場合に、UNIX ベース・システムでこれが起こることがあります。

**ユーザーの処置:** リソースのクリーンアップが必要になる可能性があります。

- インスタンス ID の下で実行中のすべてのデータベース・マネージャー・プロセスを取り除いてください (UNIX ベース・システムの場合、`"ps -eaf -u <instance id> | grep db2"` を使用すると、インスタンス ID の下で実行中のすべてのデータベース・マネージャー・プロセスをリストすることができ、`"kill -9 <process id>"` コマンドを使用すると、それらを取り除くことができます)。
- インスタンス ID の下で実行されているアプリケーションが他に存在しないことを確認した後で、そのインスタンス ID が所有しているすべてのリソースを取り除いてください (UNIX ベース・システムの場合、`"ipcs | grep <instance id>"` コマンドを使用すると、インスタンス ID が所有しているすべての IPC リソースをリストすることができ、`"ipcrm -[qlmls] <id>"` コマンドを使用すると、それらを取り除くことができます)。
- データベース・マネージャーの他のインスタンスが実行中の場合は、`i` ノード競合が発生する可能性があります。これは、2 つのインスタンスを同時にアクティブにすることはできませんが、個別に開始できることを示しています。いずれかのインスタンスの IPC キーの生成に使用される `i` ノードを変更する必要があります。単一ノード・インスタンスの場合は、`sqllib` directory からインスタンス所有者として以下を行います。
  - `.ftok` ファイルを削除する  
`rm .ftok`
  - 新しい `.ftok` ファイルを作成する  
`touch .ftok`マルチノード・インスタンスの場合は、インスタンス所有者として以下を行います。
  - `sqllib` と同じレベルで別のディレクトリーを作成する。



- `sqllib` のもとにあるすべてのものを新しいディレクトリーに移動する。
  - `sqllib` を削除する。
  - 新しいディレクトリーを `sqllib` の名前に変更する。
- データベース・マネージャー・インスタンスを再始動してください。

---

#### SQL1073N ノード・ディレクトリーのリリース番号が正しくありません。

**説明:** ノード・ディレクトリーのリリース番号が、製品で使用可能なリリース番号と一致しません。ノード・ディレクトリーが、以前のリリースで作成されたものと思われます。

**ユーザーの処置:** すべてのノード項目を再カタログして、コマンドを再発行してください。

---

#### SQL1074N `password` パラメーターのアドレスが無効です。

**説明:** アプリケーション・プログラムが、このパラメーターに無効なアドレスを使用しました。そのアドレスが割り振られていないバッファーを指しているか、またはそのバッファー内の文字ストリングに `NULL` 終止符がありません。

コマンドは処理されません。

**ユーザーの処置:** 有効なアドレスがアプリケーション・プログラムで使用され、入力ストリングが `null` で終了していることを確認してください。

---

#### SQL1075N `database comment` パラメーターのアドレスが無効です。

**説明:** アプリケーション・プログラムが、このパラメーターに無効なアドレスを使用しました。そのアドレスが割り振られていないバッファーを指しているか、またはそのバッファー内の文字ストリングに `NULL` 終止符がありません。

コマンドは処理されません。

**ユーザーの処置:** 有効なアドレスがアプリケーション・

プログラムで使用され、入力ストリングが `null` で終了していることを確認してください。

---

#### SQL1076N `count` パラメーターのアドレスが無効です。

**説明:** アプリケーション・プログラムが、`count` パラメーターに無効なアドレスを使用しました。

コマンドは処理されません。

**ユーザーの処置:** 有効なアドレスが、アプリケーション・プログラムで使用されていることを確認してください。

---

#### SQL1077N `handle` パラメーターのアドレスが無効です。

**説明:** アプリケーション・プログラムが、`handle` パラメーターに無効なアドレスを使用しました。

コマンドは処理されません。

**ユーザーの処置:** 有効なアドレスが、アプリケーション・プログラムで使用されていることを確認してください。

---

#### SQL1078N `buffer` パラメーターのアドレスが無効です。

**説明:** アプリケーション・プログラムが、`buffer` パラメーターに無効なアドレスを使用しました。

コマンドは処理されません。

**ユーザーの処置:** 有効なアドレスが、アプリケーション・プログラムで使用されていることを確認してください。

---

#### SQL1079N `nodename` パラメーターのアドレスが無効です。

**説明:** アプリケーション・プログラムが、無効な `nname` パラメーター・アドレスを使用しました。そのアドレスが割り振られていないバッファーを指しているか、またはそのバッファー内の文字ストリングに `NULL` 終止符がありません。

コマンドは処理されません。

**ユーザーの処置:** 有効なアドレスがアプリケーション・プログラムで使用され、入力ストリングが null で終了していることを確認してください。

---

**SQL1080N** *local\_lu name* パラメーターのアドレスが無効です。

**説明:** アプリケーション・プログラムが、*local\_lu name* パラメーターに無効なアドレスを使用しました。そのアドレスが割り振られていないバッファを指しているか、またはそのバッファ内の文字ストリングに NULL 終止符がありません。

コマンドは処理されません。

**ユーザーの処置:** 有効なアドレスがアプリケーション・プログラムで使用され、入力ストリングが null で終了していることを確認してください。

---

**SQL1081N** *partner\_lu* パラメーターのアドレスが無効です。

**説明:** アプリケーション・プログラムが、*partner\_lu* パラメーターに無効なアドレスを使用しました。そのアドレスが割り振られていないバッファを指しているか、またはそのバッファ内の文字ストリングに NULL 終止符がありません。

コマンドは処理されません。

**ユーザーの処置:** 有効なアドレスがアプリケーション・プログラムで使用され、入力ストリングが null で終了していることを確認してください。

---

**SQL1082N** *mode* パラメーターのアドレスが無効です。

**説明:** アプリケーション・プログラムが、*mode* パラメーターに無効なアドレスを使用しました。そのアドレスが割り振られていないバッファを指しているか、またはそのバッファ内の文字ストリングに NULL 終止符がありません。

コマンドは処理されません。

**ユーザーの処置:** 有効なアドレスがアプリケーション・プログラムで使用され、入力ストリングが null で終了していることを確認してください。

---

**SQL1083N** データベース記述ブロックが処理できません。理由コード =*reason-code*

**説明:** アプリケーションが CREATE DATABASE コマンドを発行しましたが、データベース記述ブロック (DBDB) が、以下のいずれかの理由コードが原因で処理できませんでした。

- DBDB のアドレスが無効です。(理由コード 01)
- DBDB の「SQLDBDID」フィールドの値が無効です。(理由コード 02) これは、値 SQLDBDB1 に設定される必要があります。
- DBDB の「SQLDBCSS」フィールドの値が無効です。(理由コード 04)

コマンドは処理されません。

**ユーザーの処置:** エラーを訂正して、コマンドを再発行してください。

---

**SQL1084C** 共有メモリー・セグメントを割り振ることができません。

**説明:** データベース・マネージャーが、Database Environment コマンドまたは SQL CONNECT ステートメントの処理中に、セグメントを割り振ることができませんでした。

dbheap パラメーターが小さすぎる可能性があります。

**ユーザーの処置:** メッセージ番号 (SQLCODE) を記録してください。このエラーの原因としては、データベース・マネージャー、またはデータベース・マネージャーの処理が試みられていた環境に、十分なリソースが存在しなかった可能性があります。データベース・マネージャーの要求を満足するための十分なメモリー・リソースが、使用可能なことを確認してください。アクティブ

である必要がないバックグラウンド処理のクローズも必要になる場合があります。

十分なメモリー・リソースが十分にあってもこの問題が続く場合は、オペレーティング・システムのコマンド・プロンプトで、独立トレース機能呼び出してください。以下の情報を用意して、テクニカル・サービス担当者に提供してください。

必要な情報:

- 問題記述
- SQLCODE またはメッセージ番号
- SQLCA の内容 (ある場合)
- トレース・ファイル (可能であれば)

**sqlcode:** -1084

**sqlstate:** 57019

---

#### SQL1085N アプリケーション・ヒープを割り振ることができません。

**説明:** データベース構成ファイルに指定された 4K ページ単位のアプリケーション・ヒープを、データベース・マネージャーが割り振ることができなかつたので、アプリケーションはデータベースに接続することができませんでした。システムに 4K ページがありません。コマンドは処理されません。

**ユーザーの処置:** 以下に示す方法で解決してください。

- データベース構成ファイル内のアプリケーション・ヒープの大きさ (applheapsz) を減らしてください。
- データベース構成ファイル内のアプリケーションの最大数を減らしてください。
- バックグラウンド処理を終了してください。
- メモリーを増やしてください。

**sqlcode:** -1085

**sqlstate:** 57019

---

#### SQL1086C オペレーティング・システム・エラー - *error* が起きました。

**説明:** コマンドが、オペレーティング・システムからエラーを受け取ったために、これ以上処理を続けることができません。

コマンドは処理されません。

**ユーザーの処置:** オペレーティング・システムのユーザー・マニュアルを参照して、エラーの詳細を解析してください。

---

#### SQL1087W データベースは作成されましたが、リスト・ファイル *name* のオープン中にエラーが起きました。ユーティリティーはデータベースにバインドされません。

**説明:** CREATE DATABASE は、ユーティリティー・バインド・ファイルのリストが入っているリスト・ファイルをオープンできませんでした。リスト・ファイルは *sqllib* サブディレクトリーの *bnd* サブディレクトリーに置いてください。

ユーティリティー・バインド・ファイルは、新しく作成されたデータベースへはバインドされません。

**ユーザーの処置:** ユーティリティーをデータベースにバインドしてください。バインド・プログラム呼び出しには *format* オプションを使用しないでください。

---

#### SQL1088W データベースは作成されましたが、ユーティリティーのバインド中にエラーが起きました。ユーティリティーはデータベースにバインドされません。

**説明:** CREATE DATABASE または MIGRATE DATABASE が、ユーティリティー・バインド・ファイルをデータベースにバインドできませんでした。

ユーティリティー・バインド・ファイルは、新し



く作成されたデータベースまたは移行されたデータベースにはバインドされません。

**ユーザーの処置:** ユーティリティをデータベースにバインドしてください。バインド・プログラム呼び出しには `format` オプションを使用しないでください。

---

**SQL1089W** データベースは作成されましたが、ユーティリティのバインドに中断が起きました。ユーティリティはデータベースにバインドされません。

**説明:** データベースへユーティリティをバインド中に、`CREATE DATABASE` コマンドが割り込みを受けました。割り込みキー・シーケンスが押された可能性があります。

ユーティリティ・バインド・ファイルは、新しく作成されたデータベースへはバインドされません。

**ユーザーの処置:** ユーティリティをデータベースにバインドしてください。バインド・プログラム呼び出しには `format` オプションを使用しないでください。

---

**SQL1090C** プリコンパイルされたアプリケーション・プログラムまたはユーティリティのリリース番号が無効です。

**説明:** プリコンパイルされたアプリケーション・プログラムまたはユーティリティのリリース番号が、インストールされているデータベース・マネージャーのリリース番号と互換ではありません。

アプリケーション・プログラムが、前のレベルのデータベース・マネージャー・ライブラリーまたは `DLL` を、データベース・マネージャー構成ファイルのインストール・バージョンにアクセス中に使用する場合は、エラーが発生する可能性があります。

コマンドは処理されません。

**ユーザーの処置:** アプリケーション処理に対して選択されたデータベース・マネージャー・ライブラリーのバージョンまたは `DLL` が前のもの (古い) ではないことをチェックしてください。

問題が続行する場合は、現在のデータベース・マネージャーを使用して、プリコンパイル処理を繰り返してください。互換リリース・レベルのデータベース・マネージャーでプリコンパイルされたアプリケーション・プログラムのみを使用してください。

---

**SQL1091C** データベースのリリース番号が無効です。

**説明:** データベースのリリース番号が、インストールされているバージョンのデータベース・マネージャーのリリース番号と互換性がありません。これはデータベースが作成されたときのリリース番号、データベースが最後に移行されたときのリリース番号、またはカタログに定義されている最新のバージョン、リリース、変更、フィックスパック・レベルである可能性があります。

コマンドは処理されません。移行またはリストアでエラーが起きた場合は、互換性のないリリースのデータベースを移行またはリストアしようとしています。エラーがフィックスパックスの除去後の最初の接続で発生した場合は、より高いレベルのデータベース・マネージャー・コードを使用することを定義されているデータベースに接続しようとしています。

**ユーザーの処置:** 互換性のあるリリースのデータベース・マネージャーを使用して作成されたデータベースだけを使用してください。移行またはリストア中にエラーが起きた場合は、最初に、現行リリースのデータベース・マネージャーで移行できるリリースに、データベースを移行する必要があります。エラーがフィックスパックスの除去後の最初の接続で発生した場合は、ユーティリティを使用してデータベースをフィックスパック・レベルに更新する前にデータベースをバックアップからリストアする必要があります。

**sqlcode:** -1091

**sqlstate:** 08004

---

**SQL1092N** *authorization-ID* には要求されたコマンドを実行する許可がありません。

**説明:** そのコマンドまたは処理に対する適切な権限を持たずに、コマンドまたは処理を実行しようとしてしました。

コマンドは処理されません。

連合システム・ユーザー: この状態はデータ・ソースでも検出できます。

**ユーザーの処置:** 正しい権限を持つユーザーとしてログオンして、失敗したコマンドまたは処理をやり直してください。正しい権限には、SYSADM、SYSCTRL、SYSMAINT、DBADM が含まれます。DBADM はデータベースに付与され、他のすべての権限は、データベース・マネージャー構成に定義されているグループのメンバーによって決定されます (たとえば、データベース・マネージャー構成ファイルの *sysctrl\_group* が、'beatles' として定義されている場合、SYSCTRL 権限を得るには、グループ 'beatles' に属している必要があります)。試みられたコマンドまたは操作に必要な権限のリストについては、コマンド・リファレンス または *SQL* リファレンスを参照してください。

Windows 環境で Kerberos 認証を使用している場合、ドメイン・アカウントでマシンにログオンしていることを確認してください。Windows 環境では、ドメイン・ユーザーだけが Kerberos 認証を使用できます。

LDAP サポートを使用している場合、ユーザーが DB2 Connect ゲートウェイが CATALOG DATABASE、NODE および DCS DATABASE コマンドを実行する権限があることを確認してください。クライアントまたはゲートウェイで "UPDATE DBM CFG USING CATALOG\_NOAUTH YES" コマンドを呼び出して、この問題を訂正してください。

連合システム・ユーザー: 必要な場合は、要求を拒否しているデータ・ソースに問題を分離します。(障害の起きたデータ・ソースを識別する手順については「トラブルシューティング・ガイド」を参照してください。) 次に、指定された許可 ID がそのデータ・ソースに適切な権限であるかどうかを確認してください。

権限の要求については、システム管理者に連絡してください。適切な許可を持たずに、コマンドを実行しないでください。

---

**SQL1093N** ユーザーはログオンしていません。

**説明:** ユーザーは、許可を必要とするコマンドが処理される前に、ログオンしておく必要があります。このエラーの原因には、以下が含まれます。

- ユーザー ID が取得できない。
- ログオン時に、予期しないオペレーティング・システムのエラーが起きた。
- アプリケーションがバックグラウンド処理で実行されている。
- ユーザーがログオンを取り消した。

コマンドは処理されません。

**ユーザーの処置:** 有効なユーザー ID でログオンして、コマンドを再発行してください。さまざまな並列処理がログオンを要求しているときは、数秒待ってから、ログオン手順を繰り返してください。

**sqlcode:** -1093

**sqlstate:** 51017

---

**SQL1094N** ノード・ディレクトリーは更新中なので、アクセスできません。

**説明:** ノード・ディレクトリーの更新中は、ノード・ディレクトリーを走査することも、使用することもできません。また、データベース・ディレクトリーが何らかの理由ですでにアクセスされている場合は、更新のためにディレクトリーにアクセスすることはできません。

コマンドは処理されません。

**ユーザーの処置:** 更新が完了してからコマンドを再発行してください。

**sqlcode:** -1094

**sqlstate:** 57009

---

**SQL1095N** すでに 8 つのノード・ディレクトリー走査がオープンしています。

**説明:** 8 つのノード・ディレクトリー走査がこの処理でオープンされており、これ以上走査をオープンすることはできません。

コマンドは処理されません。

**ユーザーの処置:** 1 つ以上の CLOSE NODE DIRECTORY SCAN コマンドを実行してください。コマンドを再発行してください。

**sqlcode:** -1095

**sqlstate:** 54029

---

**SQL1096N** このコマンドは、このノード・タイプには無効です。

**説明:** コマンドをサポートしないノードでコマンドが実行されたか、またはこのノード・タイプを正しくセットアップしていないシステム環境が見つかりました。たとえば、データベースが、クライアント・ノードで LOCAL としてカタログされています。

コマンドは処理されません。

**ユーザーの処置:** コマンドおよびパラメーターが、ノード・タイプに対して正しいことを確認してください。また、コマンドが処理される環境が正しいことを確認してください。コマンドを再発行してください。

---

**SQL1097N** ノード名がノード・ディレクトリーにありません。

**説明:** リモート・データベースのデータベース・ディレクトリーにリストされているノード名、ま

たはアタッチするコマンドに指定されているノード名が、ノード・ディレクトリーにカタログされていません。

コマンドは処理されません。

**連合システム・ユーザー:** この状態はデータ・ソースでも検出できます。

**ユーザーの処置:** データベース・ディレクトリーにリストされているノード名、またはアタッチ・コマンドのオブジェクトのノード名が、ノード・ディレクトリーにカタログされていることを確認してください。ノードがノード・ディレクトリーにリストされない場合は、CATALOG NODE コマンドを発行してください。

**連合システム・ユーザー:** 上記でリストされたアクションに加えて、すべての SYSCAT.SERVERS 項目にリストされたノード名が正しいことを検証してください。ノードがノード・ディレクトリーにリストされず、サーバーが DB2 ファミリーのメンバーである場合は、そのノードに CATALOG NODE コマンドを実行してください。

**sqlcode:** -1097

**sqlstate:** 42720

---

**SQL1098N** このデータベースには、アプリケーションがすでに接続されています。

**説明:** データベースへの接続が要求されましたが、アプリケーションは指定されたデータベースにすでに接続されています。

コマンドは処理されません。

**ユーザーの処置:** 処置は必要ありません。

**sqlcode:** -1098

**sqlstate:** 53056

---

**SQL1099N** ディスケットが書き込み禁止になっています。

**説明:** 書き込み操作が、書き込み禁止になっているディスク内のデータベースに対して試みられました。

## SQL1100 - SQL1199

---

**SQL1100W** **Catalog Database** コマンドに指定されたノード名 *name* が、ノード・ディレクトリーにカタログされていません。

**説明:** Catalog Database コマンドがノード名 *name* を指定しましたが、ノード・ディレクトリーにカタログされていません。リモート・データベースを使用する前に、ノード名をカタログ作成する必要があります。

CATALOG DATABASE コマンドは正常に処理されます。

**ユーザーの処置:** CATALOG NODE コマンドを実行してください。

---

**SQL1101N** ノード *node-name* のリモート・データベース *name* に、指定された許可 ID とパスワードでアクセスできませんでした。

**説明:** ノード *node-name* 上のリモート・データベース *name* への接続が要求されましたが、(リモート許可表か実行時に) このノードに対して指定された許可 ID とパスワードの組み合わせをリモート・ノードが受け入れませんでした。

要求は処理できません。

**ユーザーの処置:** リモート・システム用の有効な許可 ID およびパスワードを使用して、要求の再実行依頼を行なってください。

ステートメントは処理できません。

**ユーザーの処置:** 正しいディスクットを使用していることを確認してください。必要に応じて、ディスクットから保護を取り除いてください。

---

**SQL1102N** データベース名が指定されませんでした。

**説明:** 移行時にデータベース名が指定されていませんでした。

ステートメントは処理できません。

**ユーザーの処置:** 移行用のデータベース名を指定してください。

---

**SQL1103W** **Migrate Database** コマンド処理が正常に終了しました。

**説明:** Migrate コマンドが正常に終了しました。データベースがすでに現在のレベルで、移行する必要がない場合でも、このメッセージが返されることに注意してください。

処理を続行します。

**ユーザーの処置:** アクションは必要ありません。

---

**SQL1104N** **program name** パラメーターのアドレスが無効です。

**説明:** アプリケーション・プログラムが、*program name* に無効なアドレスを使用しました。

コマンドは処理されません。

**ユーザーの処置:** アプリケーション・プログラムで正しいアドレスを使用してください。

---

**SQL1105N SQL CONNECT RESET** ステートメントは、リモート・アプリケーション・インターフェース・プロシージャでは許されていません。

**説明:** リモート・アプリケーション・プロシージャに、SQL CONNECT RESET ステートメントが含まれています。

リモート・プロシージャは続行できません。

**ユーザーの処置:** SQL CONNECT RESET ステートメントを取り除いて、リモート・プロシージャを再実行してください。

**sqlcode:** -1105

**sqlstate:** 38003

---

**SQL1106N 指定された DLL name モジュールはロードされましたが、関数 function は実行できませんでした。**

**説明:** DLL (ダイナミック・リンク・ライブラリー) 内の関数が見つかりませんでした。

コマンドは処理されません。

**ユーザーの処置:** DLL モジュールが正しく作成されていることを確認してください。定義ファイルのモジュールを参照してください。

**sqlcode:** -1106

**sqlstate:** 42724

---

**SQL1107N 指定された DLL name をロード中に、割り込みを受けました。**

**説明:** DLL (ダイナミック・リンク・ライブラリー) をロードしているときに、コマンドが割り込みを受けました。(Ctrl+Break が押された可能性があります)

処理は停止します。

**ユーザーの処置:** コマンドを再発行してください。

**sqlcode:** -1107

**sqlstate:** 42724

---

**SQL1108N 指定された DLL name をロード中に、予期しない入出力エラーまたはオペレーティング・システム・エラーを受け取りました。**

**説明:** 「プログラム名」フィールドに指定された DLL (ダイナミック・リンク・ライブラリー) をロードするときに、予期しないエラーが起きました。

コマンドは処理されません。

**ユーザーの処置:** 現在のコマンドを再発行してください。それでも、エラーが続く場合は、データベース・マネージャを再インストールしてください。

インストールがエラーを修正しない場合は、エラー・メッセージ番号 (SQLCODE)、および可能であれば、SQLCA 内のすべての情報を記録してください。

トレースがアクティブの場合は、オペレーティング・システムのコマンド・プロンプトから、独立トレース機能呼び出してください。テクニカル・サービス担当者に連絡してください。

**sqlcode:** -1108

**sqlstate:** 42724

---

**SQL1109N 指定された DLL name がロードできませんでした。**

**説明:** 指定された DLL (ダイナミック・リンク・ライブラリー) モジュールが見つかりませんでした。

コマンドは処理されません。

**ユーザーの処置:** 指定したファイルが、システムの LIBPATH に指定されたサブディレクトリーに存在することを確認してください。

**sqlcode:** -1109

**sqlstate:** 42724

---

**SQL1110N** 指定されたデータ域が無効で、使用できませんでした。

**説明:** データ域が正しく初期化されていません。

コマンドは処理されません。

**ユーザーの処置:** ユーザー指定の「入力 SQLDA」または「出力 SQLDA」フィールドが、正しく初期化されていることを確認してください。

---

**SQL1111N** 指定されたプログラム名 *name* は無効です。

**説明:** DLL (ダイナミック・リンク・ライブラリー) モジュールまたはプログラム名の構文が正しくありません。

コマンドは処理されません。

**ユーザーの処置:** DLL またはプログラム名が正しく指定されていることを確認してください。

**sqlcode:** -1111

**sqlstate:** 42724

---

**SQL1112N** 指定された DLL *name* のロードに十分なシステム・リソースがありません。

**説明:** 指定された DLL (ダイナミック・リンク・ライブラリー) モジュールをロードするためのランダム・アクセス・メモリー (RAM) が、十分ではありません。

コマンドは処理されません。

**ユーザーの処置:** アプリケーションを停止してください。以下に示す方法で解決してください。

- CONFIG.SYS ファイルの MEMMAN NO SWAP、NO MOVE オプションを、SWAP、MOVE に変更してください。
- バックグラウンド処理を終了してください。
- メモリー割り振りを定義する構成パラメーターの値を減らしてください。

- もっと多くのランダム・アクセス・メモリー (RAM) をインストールしてください。

**sqlcode:** -1112

**sqlstate:** 42724

---

**SQL1113N** 出力 SQLDA の *sqlvar n* のデータ・タイプが、*type-1* から *type-2* に変更されました。

**説明:** リモート・ストアード・プロシージャが、出力 SQLDA 内の *n* 番目の SQLVAR のデータ・タイプを変更しました。*(n* は、不一致が見つかった最初の SQLVAR の通し番号を示します) ストアード・プロシージャはデータを戻しません。

**ユーザーの処置:** リモート・ストアード・プロシージャを修正して、出力 SQLDA のデータ・タイプ情報が変更されないようにしてください。

**sqlcode:** -1113

**sqlstate:** 39502

---

**SQL1114N** 出力 SQLDA の *sqlvar n* のデータ長が、*length-1* から *length-2* に変更されました。

**説明:** リモート・ストアード・プロシージャが、出力 SQLDA 内の *n* 番目の *sqlvar* のデータ長を変更しました (*n* は、不一致が発見された最初の SQLVAR の通し番号を示します)。

ストアード・プロシージャはデータを戻しません。

**ユーザーの処置:** リモート・ストアード・プロシージャを修正して、出力 SQLDA のデータの長さ情報が変更されないようにしてください。

**sqlcode:** -1114

**sqlstate:** 39502



---

**SQL1115N** 出力 SQLDA の **sqlvars** の数が、*count-1* から *count-2* に変更されました。

**説明:** リモート・プロシージャが、出力 SQLDA の「sql<sub>d</sub>」フィールドを変更しました (sql<sub>d</sub> は、SQLDA 内の使用された sqlvar の数を示します)。

ストアード・プロシージャはデータを戻しません。

**ユーザーの処置:** リモート・ストアード・プロシージャを修正して、出力 SQLDA の「sql<sub>d</sub>」フィールドが変更されないようにしてください。

**sqlcode:** -1115

**sqlstate:** 39502

---

**SQL1116N** 「バックアップ・ペンディング中」のために、データベース *name* の接続または活動化を行うことはできません。

**説明:** ロールフォワード・リカバリーの開始点を用意するために、指定されたデータベースがバックアップを要求しています。

接続は行われていません。

**ユーザーの処置:** BACKUP ルーチン呼び出すことにより、データベースをバックアップするか、またはロールフォワードを必要としない場合は、ログを保存して、ユーザー出口データベース構成パラメーターをオフに設定してください。

**sqlcode:** -1116

**sqlstate:** 57019

---

**SQL1117N** 「ロールフォワード保留中」のために、データベース *name* の接続または活動化を行うことはできません。

**説明:** 指定されたデータベースは、ロールフォワード・リカバリーが使用可能な状態でリストアさ

れましたが、ロールフォワードは行なわれません。

接続は行われていません。

連合システム・ユーザー: この状態はデータ・ソースでも検出できます。

**ユーザーの処置:** データベースをロールフォワードするか、または ROLLFORWARD コマンドを使用して、ロールフォワードを行なわないことを指示してください。データベースのロールフォワードを行なわないと、データベースの最後のバックアップ以降に書かれたレコードが、データベースに適用されないことに注意してください。

連合システム・ユーザー: 必要な場合は、要求を拒否しているデータ・ソースに問題を分離します。(障害の起きたデータ・ソースを識別する手順については「トラブルシューティング・ガイド」を参照します。) 次に、そのデータ・ソースに適切なリカバリー・アクションを行い、データ・ソースを整合点までリカバーします。

**sqlcode:** -1117

**sqlstate:** 57019

---

**SQL1118N** 前のバックアップが完了していないために、データベース *name* の接続または活動化を行うことはできません。

**説明:** バックアップ処理中にシステム・エラーが起きたので、データベースが不整合状態になっています。

接続は行われていません。

連合システム・ユーザー: この状態はデータ・ソースでも検出できます。

**ユーザーの処置:** BACKUP コマンドを発行して、もう一度コマンドをやり直してください。

連合システム・ユーザー: 必要な場合は、要求を拒否しているデータ・ソースに問題を分離します。(障害の起きたデータ・ソースを識別する手順については「トラブルシューティング・ガイド」

を参照してください。) 次に、コマンドを試行する前にデータ・ソースに対して BACKUP コマンドを発行してください。

**sqlcode:** -1118

**sqlstate:** 57019

---

**SQL1119N** 前のリストアが完了していないために、データベース *name* の接続または活動化を行うことはできません。

**説明:** リストア処理中にシステム・エラーが起きたので、データベースが不整合状態になっています。

接続は行われていません。

連合システム・ユーザー: この状態はデータ・ソースにより検出できます。

**ユーザーの処置:** RESTORE コマンドを発行して、もう一度コマンドをやり直してください。

連合システム・ユーザー: 必要な場合は、要求を拒否しているデータ・ソースに問題を分離します。(障害の起きたデータ・ソースを識別する手順については「トラブルシューティング・ガイド」を参照してください。) 次に、コマンドを再試行する前に、データ・ソースに対して RESTORE コマンドを発行してください。

**sqlcode:** -1119

**sqlstate:** 57019

---

**SQL1120N** 前のバックアップまたはリストアが完了していないために、データベース *name* の接続または活動化を行うことはできません。

**説明:** バックアップまたはリストア処理中にシステム・エラーが起きたので、データベースが不整合状態になっています。バックアップまたはリストアのどちらが処理されていたかが決定できません。

接続は行われていません。

連合システム・ユーザー: この状態はデータ・ソースでも検出できます。

**ユーザーの処置:** BACKUP または RESTORE コマンドを発行してから、もう一度コマンドをやり直してください。

連合システム・ユーザー: 必要な場合は、要求を拒否しているデータ・ソースに問題を分離します。(障害の起きたデータ・ソースを識別する手順については「トラブルシューティング・ガイド」を参照してください。) 次に、コマンドを再試行する前に、データ・ソースに対して BACKUP または RESTORE コマンドを発行してください。

**sqlcode:** -1120

**sqlstate:** 57019

---

**SQL1121N** *node structure* パラメーターのアドレスが無効です。

**説明:** アプリケーションが、*node structure* パラメーターに無効なアドレスを使用しました。そのアドレスが割り振られていないバッファを指しているか、または必須入力を含むための十分なバッファがありません。

コマンドは処理されません。

**ユーザーの処置:** プログラムが必要なバッファ領域の割り振りを行なっていることを確認して、コマンドを再実行してください。

---

**SQL1122N** *protocol structure* パラメーターのアドレスが無効です。

**説明:** アプリケーションが、*protocol structure* パラメーターに無効なアドレスを使用しました。そのアドレスが割り振られていないバッファを示しているか、または正しくないプロトコル・バッファを示しているかのどちらかです。

コマンドは処理されません。

**ユーザーの処置:** プロトコルが、ノード構造の「プロトコル」フィールドにもとづいた必須バッファ領域の割り振りを行なっていることを確認し



て、コマンドを再発行してください。

---

**SQL1123N** プロトコル *type* が無効です。

**説明:** Catalog コマンドのノード構造に指定したプロトコル・タイプが、認識されない値です。有効なプロトコル・タイプは、*sqlenv* ヘッダー・ファイルに定義されています。

コマンドは処理されません。

**ユーザーの処置:** ノード構造のプロトコル・タイプを確認して、コマンドを再発行してください。

---

**SQL1124N** リモート・ワークステーション名 *name* が無効です。

**説明:** Catalog コマンドの NETBIOS プロトコル構造に指定したリモート・ワークステーション名が、指定されていないか、または無効な文字を含んでいます。ワークステーション名は 1 から 8 文字でなければなりません。有効な文字は A から Z、a から z、0 から 9、#、@ および \$ です。最初の文字は、英字または特殊文字 #、@、または \$ でなければなりません。小文字は、システムによって大文字に変更されます。

コマンドは処理されません。

**ユーザーの処置:** リモート・ワークステーション名に指定された文字を確認してください。有効なワークステーション名を使用して、コマンドを再発行してください。

---

**SQL1125N** アダプター番号 *number* が無効です。

**説明:** Catalog コマンドの NETBIOS プロトコル構造に指定されたアダプター番号が無効です。

コマンドは処理されません。

**ユーザーの処置:** アダプター番号が有効なことを確認して、コマンドを再発行してください。

---

**SQL1126N** ネットワーク ID *ID* が無効です。

**説明:** Catalog コマンドの APPN プロトコル構造のネットワーク ID が無効です。ネットワーク ID が、リモート LU (LU) が存在する SNA ネットワークを識別しています。ネットワーク ID は 1 から 8 文字でなければなりません。有効な文字は A から Z、a から z、0 から 9、#、@ および \$ です。最初の文字は、英字または特殊文字 #、@、または \$ でなければなりません。小文字は、システムによって大文字に変更されます。

コマンドは処理されません。

**ユーザーの処置:** ネットワーク ID に指定された文字を確認してください。有効なネットワーク ID を使用して、コマンドを再発行してください。

---

**SQL1127N** リモート LU 名 *name* が無効です。

**説明:** Catalog コマンドの APPN プロトコル構造に指定されたリモート LU (LU) 名が無効です。リモート LU 名は、リモート SNA LU 名で、1 から 8 文字でなければなりません。有効な文字は A から Z、a から z、0 から 9、#、@ および \$ です。最初の文字は、英字または特殊文字 #、@、または \$ でなければなりません。小文字は、システムによって大文字に変更されません。

コマンドは処理されません。

**ユーザーの処置:** リモート LU 名に指定された文字を確認してください。有効なリモート LU 名を使用して、コマンドを再発行してください。

---

**SQL1128W** SAVECOUNT は無視されます。

**説明:** MDC 表にロードする場合、整合性点はサポートされていません。

**ユーザーの処置:** アクションは必要ありません。

---

**SQL1129N** 新しい処理を生成するためのリソースが不十分なために、新しい **DARI (ストアード・プロシージャー) 処理** が開始できませんでした。

**説明:** 新しい処理を生成するためのリソースが不十分なために、新しい **DARI (ストアード・プロシージャー) 処理** が開始できませんでした。

**ユーザーの処置:** 以下のいずれかを実行します。

- DB2 を使用しているユーザー数を減らしてください。
- システム処理の上限を増やしてください。

**sqlcode:** -1129

**sqlstate:** 42724

---

**SQL1130N** サーバー上で同時に存在可能な **DARI 処理** の最大数に達したために、新しい **DARI (ストアード・プロシージャー) 処理** が開始できませんでした。

**説明:** サーバー上で同時に存在可能な **DARI 処理** の最大数に達したために、新しい **DARI (ストアード・プロシージャー) 処理** が開始できませんでした。

**ユーザーの処置:** **DARI 処理** の最大数を指定する *maxdari* 構成パラメーターを増やしてください。詳細については、データベース構成パラメーター *maxdari* を参照してください。

**sqlcode:** -1130

**sqlstate:** 42724

---

**SQL1131N** **DARI (ストアード・プロシージャー) 処理** が異常終了しました。

**説明:** このエラーの原因として、以下のことが考えられます。

- **DARI** ルーチン内にコーディング・エラー (すなわちセグメント違反) がある。

- 他の処理がシグナルを使用して、**DARI 処理** を終了させました。

**ユーザーの処置:** 以下を行った後で、**DARI 要求** を再始動してください。

- **DARI** プロシージャーにプログラミング・エラーがないことを確認してください。
- **DARI 処理** に対して、終了シグナルを送っているユーザーがいらないことを確認してください。

**sqlcode:** -1131

**sqlstate:** 38503

---

**SQL1132N** このコマンドは、**DARI (ストアード・プロシージャー) 内での実行** を許されていません。

**説明:** **DARI (ストアード・プロシージャー)** の有効範囲内で、無効なコマンドが発行されました。

**DARI** プロシージャーは、続行できません。

**ユーザーの処置:** **DARI** プロシージャー内の不当なコマンドを取り除いて、再実行してください。

**sqlcode:** -1132

**sqlstate:** 38003

---

**SQL1133N** 出力 **SQLDA** の **sqlvar (index = n)** 内のポインター・アドレスが、**DARI (ストアード・プロシージャー)** 関数内で変更されました。

**説明:** 出力 **SQLDA** の **sqlvar** 内の “sqlind” または “sqldata” ポインターが、ユーザーが用意した **DARI** 関数内で変更されました。

**ストアード・プロシージャー** はデータを戻しません。

**ユーザーの処置:** 出力 **SQLDA** 内の示された **sqlvar** の使用法を修正して、**DARI (ストアード・プロシージャー)** 関数ルーチンの中で、ポインター・アドレスが変更されないようにしてください。

**sqlcode:** -1133

**sqlstate:** 39502

---

**SQL1134N** データベース認証タイプが **CLIENT** の場合、このコマンドは **DARI** (ストアド・プロシージャー) の有効範囲内では許されていません。

**説明:** データベースの認証タイプが **CLIENT** の場合、**ALL SYSADM** コマンドは **DARI** (内部記憶式プロシージャー) 内で実行できません。

ストアド・プロシージャーはデータを戻しません。

**DARI** プロシージャーは、続行できません。

**ユーザーの処置:** **DARI** プロシージャー内の不当なコマンドを取り除いて、再実行してください。

**sqlcode:** -1134

**sqlstate:** 38003

---

**SQL1135N** データベースの作成時に、無効なセグメント数が指定されました。

**説明:** セグメント数に指定された値が範囲外です。有効範囲は 1 から 256 です。

**ユーザーの処置:** セグメント数を指定し直して、もう一度データベースを作成してください。

---

**SQL1136N** データベースの作成時に、無効な値がデフォルト表スペース・エクステント・サイズ (**dft\_extentsize**) に指定されました。

**説明:** デフォルト表スペース・エクステント・サイズ (**dft\_extentsize**) に指定された値が範囲外です。有効範囲は 2 から 256 です。

**ユーザーの処置:** 表スペース・エクステント・サイズを訂正して、やり直してください。

---

**SQL1137W** データベース *dbalias* のドロップ時に、データベース・マネージャーがデータベース・パスまたはいくつかのコンテナを除去できませんでした。クリーンアップが必要です。

**説明:** コンテナのリストがアクセスできなかったか、またはコンテナまたはデータベース・ディレクトリーの除去中に、障害が起きました。

**ユーザーの処置:** システム管理コンテナ (ディレクトリー)、およびデータベース管理ファイル・コンテナを、オペレーティング・システム・コマンドを使用して、手操作で除去することが必要になる可能性があります。装置コンテナを解放するには、担当の **IBM** 技術員に連絡してください。

**New Log Path** 構成パラメーターでログ・ディレクトリーが変更されている場合は、ログ・ディレクトリー・ファイル・システムを手操作で取り外し、ログおよびデータベース・ディレクトリーを除去してください。

---

**SQL1138W** ユニーク索引 *name* が据え置きユニーク・チェックをサポートするために移行しました。新規索引は作成されません。

**説明:** **CREATE INDEX** 処理が既存の索引で試行されました。索引が据え置きユニーク・チェックをサポートするために移行していないため、この移行は実行されませんでした。

ユニーク索引の移行された形式によって、行ごとに更新が行われるのではなく更新ステートメントの最後に索引列のユニーク性チェックを複数の行で行うことができます。

**ユーザーの処置:** アクションは必要ありません。

**sqlcode:** +1138

**sqlstate:** 01550

---

**SQL1139N** 表スペースの合計サイズが大き過ぎます。

**説明:** 現在の表スペースの合計サイズが大き過ぎます。REGULAR または USER TEMPORARY 表スペースのサイズは 0xFFFFF (16777215) ページに制限され、SYSTEM TEMPORARY または LONG 表スペースのサイズは 2 テラバイト (2 TB) に制限されています。

**ユーザーの処置:** 詳細については管理通知ログをチェックしてください。表スペースのサイズを減らして、SQL ステートメントを訂正してください。

**sqlcode:** -1139

**sqlstate:** 54047

---

**SQL1140W** コスト・カテゴリー *cost-category* で、*estimate-amount1* プロセッサ秒の見積もられたプロセッサ・コスト (*estimate-amount2* サービス単位) が、*limit-amount* サービス単位のリソース制限警告しきい値を超えています。

**説明:** 動的 INSERT、UPDATE、DELETE、または SELECT SQL ステートメントの準備の結果、リソース限定表 (RLST) に指定された警告しきい値を超えるコスト見積もりが発生しました。

この警告は、DB2 のコスト・カテゴリー値が "B" であり、RLST の RLF\_CATEGORY\_B 列に指定されたデフォルトのアクションが警告の発行である場合にも発行されます。

**estimate\_amount1**

準備された

INSERT、UPDATE、DELETE または SELECT ステートメントが実行された場合のコスト見積もり (プロセッサ秒)。

**estimate\_amount2**

準備された

INSERT、UPDATE、DELETE または

SELECT ステートメントが実行された場合のコスト見積もり (サービス単位)。

**cost-category**

この SQL ステートメントについての DB2 のコスト・カテゴリー。使用可能な値は A または B です。

**limit-amount**

RLST の RLFASUWARN 列に指定されている警告しきい値 (サービス単位)。

動的 INSERT、UPDATE、DELETE、または SELECT ステートメントの準備は成功しました。準備されたステートメントを実行して、RLST に指定された ASUTIME 値を超える場合は、SQLCODE -905 が発行される可能性があります。

**ユーザーの処置:** 警告を扱って、ステートメントの実行を許可するか、ステートメントの実行を行わないことを決定するためのアプリケーション論理が存在することを確認してください。コスト・カテゴリー値が "B" であるためにこの SQLCODE が返された場合は、ステートメントがパラメーター・マーカを使用しているか、参照される表と列について使用できない統計が存在する可能性があります。管理者が、参照された表でユーティリティ RUNSTATS を実行したことを確認してください。また、ステートメントが実行されるときに UDF が呼び出されるか、INSERT、UPDATE、または DELETE ステートメントについては、変更された表にトリガーが定義されている可能性もあります。このステートメントについて DSN\_STATEMNT\_TABLE または IFCID 22 レコードをチェックして、この SQL ステートメントがコスト・カテゴリー "B" になった理由を判別してください。

SQL ステートメントがプロセッサ・リソースを多く使用しすぎていることが警告の原因である場合は、ステートメントが効率良く実行されるために書き直してみてください。もう 1 つのオプションとして、RLST の警告しきい値を上げることを管理者に要請することもできます。

**sqlcode:** +1140

sqlstate: 01616

---

**SQL1141N** 操作は完了しましたが、エラーまたは警告がありました。この詳細は結果ファイル *file-name* にあります。このファイルは、db2inspf ユーティリティを使ってフォーマットする必要があります。

**説明:** 区分化データベース環境では、ファイル拡張子はデータベース・パーティションのノード番号に対応しています。ファイルは、DIAGPATH データベース・マネージャー構成パラメーターで指定されているディレクトリーにあります。

**ユーザーの処置:** db2inspf ユーティリティを使って、検査データ結果ファイル *file-name* をフォーマットします。

---

**SQL1142N** ファイル *file-name* はすでに使用中のため、この操作は完了できません。

**説明:** INSPECT コマンドに指定されたファイル名を使用する既存ファイルが存在します。

コマンドは処理されません。

**ユーザーの処置:** まだ使用されていない別のファイル名を使ってコマンドを再実行依頼してください。または、既存ファイル *file-name* を除去して、コマンドを再実行依頼してください。

区分化データベース環境では、ファイル拡張子はデータベース・パーティションのノード番号に対応しています。ファイルは、DIAGPATH データベース・マネージャー構成パラメーターで指定されているディレクトリーにあります。

---

**SQL1143N** ファイル *file-name* でファイル・エラーが起きたために、操作を完了できません。

**説明:** 入出力エラーのために、ファイルにアクセスできませんでした。システムが、ファイルのオープン、読み取り、または書き込みを行うこと

ができません。ファイルが不完全であるか、またはディスクがいっぱいの可能性があります。

ユーティリティは処理を停止します。

**ユーザーの処置:** 可能であれば、SQLCA からすべてのエラー情報を記録してください。追加情報については、診断ログ・ファイルを調べてください。

データベース・パーティション環境では、ファイル拡張子はデータベース・パーティションのノード番号に対応します。ファイルは、DIAGPATH データベース・マネージャー構成パラメーターで指定されているディレクトリーにあります。

---

**SQL1144N** 現在のトランザクションは、索引の作成に失敗して、ロールバックされました。sqlcode = *sqlcode*。

**説明:** 同じトランザクションで、最後に残った表の索引がドロップされ、同じ表について新しい索引が作成されました。この索引の作成は、エラー *sqlcode* で失敗したか、または ROLLBACK TO SAVEPOINT ステートメントによりロールバックされました (sqlcode=0)。最後に残った表の索引の索引ドロップがまだコミットされていない場合、索引作成のロールバックは、正常に完了できません。このいずれの場合も、トランザクション全体がロールバックされます。索引は、ALTER TABLE ステートメントのユニーク・キーまたは主キーの制約をドロップまたは追加しても、作成またはドロップできる点に留意してください。

**ユーザーの処置:** 可能であれば、トランザクション全体がロールバックされるのを避けるために、同じ表について新しい索引を作成する前に、索引のドロップをコミットしてください。sqlcode が 0 でない場合は、その sqlcode のメッセージで、訂正アクションを確認してください。ロールバック・トランザクションのすべてのステートメントを再発行する必要があります。sqlcode がゼロで、索引が savepoint 有効範囲内でドロップされている場合は、同じ表に対する古い索引をドロップ



ブする前に、新しい索引を作成するためにステートメントの再順序付けが必要な場合があります。ユニーク・キーまたは主キーのドロップと追加が同じ ALTER TABLE ステートメントにある場合は、2 つのステートメントで行う必要があります。1 つ目のステートメントで ADD を実行し、2 つ目のステートメントでドロップを実行します。

**sqlcode:** -1144

**sqlstate:** 40507

---

**SQL1145N** ゲートウェイ集線装置を使用している場合、**PREPARE** ステートメントはサポートされていません。理由コード: *reason-code*

**説明:** 以下のいずれかの *reason-code* によって、ステートメントが失敗しました。

- 1 ゲートウェイ集線機能が ON になっている場合、組み込み SQL にある動的に準備されたステートメントはサポートされていません。この構成では、クライアントが CLI アプリケーションである場合のみ、動的に準備されたステートメントがサポートされています。
- 2 ゲートウェイ集線機能が ON になっている場合、動的に準備された SET ステートメントはサポートされていません。

**ユーザーの処置:** 理由コードに基づいて、以下のいずれかのアクションを行ってください。

- 1 動的 SQL ステートメントに CLI を使用するようアプリケーションを変更するか、または静的 SQL を使用するようアプリケーションを変更する
- 2 SET ステートメントに EXECUTE IMMEDIATE を使用する

**sqlcode:** -1145

**sqlstate:** 560AF

---

**SQL1146N** 表 *table-name* に索引がありません。

**説明:** 索引再編成で指定した *table-name* に索引がありません。

**ユーザーの処置:** 有効な表名を使用して、コマンドを再発行してください。

---

**SQL1147W** TOTALFREESPACE は、MDC 表では無視されます。

**説明:** 表のフリー・スペースは MDC 表に対してよく管理されているため、TOTALFREESPACE ファイル・タイプ修飾子は不必要なので無視されます。

**ユーザーの処置:** アクションは必要ありません。

---

**SQL1150N** *user id* パラメーターのアドレスが無効です。

**説明:** アプリケーション・プログラムが、このパラメーターに無効なアドレスを使用しました。そのアドレスが割り振られていないバッファーを指しているか、またはそのバッファー内の文字ストリングに NULL 終止符がありません。

コマンドは処理されません。

**ユーザーの処置:** 有効なアドレスがアプリケーション・プログラムで使用され、入力ストリングが null で終了していることを確認してください。

---

**SQL1160N** DOS *network-protocol* TSR がロードされていません。

**説明:** 指定された通信プロトコルの終了後常駐型 (TSR) ネットワーク・ドライバはロードされていません。TSR はネットワーク通信を使用する前にロードされている必要があります。

**ユーザーの処置:** 指定した通信プロトコルの TSR が正常にロードされていることを確認した後で、もう一度アプリケーションを実行してください。

---

**SQL1161W** 調整処理が失敗しました。  
**DataLink** 列が **DB2 DataLinks Manager** で定義されていません。  
詳細については管理通知ログをチェックしてください。

**説明:** 表の **DataLink** 列に関するメタデータ情報が **DB2 DataLinks Manager** にありません。調整処理が失敗しました。表はデータ・リンク調整不可 (**DRNP**) 状態になります。

**ユーザーの処置:** 表をデータ・リンク調整不可 (**DRNP**) 状態から解除するには、「管理ガイド」の「データ・リンク調整不可 (**DRNP**) 状態から表を解除する」に記載されている手順にしたがってください。

---

**SQL1162W** 調整処理が失敗しました。例外処理中に **DB2 DataLinks Manager** が使用できなくなりました。

**説明:** 表データによって参照されている **DB2 DataLinks Manager** の 1 つが、調整の例外処理中に使用できなくなりました。調整処理が失敗しました。表はデータ・リンク調整ペンディング (**DRP**) 状態になります。

**ユーザーの処置:** もう一度調整を実行します。

---

**SQL1163N** タイプ *ident-type* の ID 名 *ident-name* が長すぎるために、データ・キャプチャーについて表を使用可能にすることができませんでした。

**説明:** データ・キャプチャーは、特定の長さを超える ID タイプではサポートされていません。試みられた変更の処理中に、タイプ *ident-type* の ID *ident-name* が長すぎるのがわかりました。データ・キャプチャーを可能にするための ID タイプと最大長は以下のとおりです。

1. 列。データ・キャプチャーを使用可能にするためには、列名は 18 バイト以下でなければなりません。

2. 表。データ・キャプチャーを使用可能にするためには、表名は 18 バイト以下でなければなりません。
3. スキーマ。データ・キャプチャーを使用可能にするためには、スキーマ名は 18 バイト以下でなければなりません。

**ユーザーの処置:** この表のデータ・キャプチャーを使用可能にする場合は、問題の ID が上に示されている最大値を超えていないことを確認してください。そうでない場合は、長い ID 名を使用するために表のデータ・キャプチャーをできなくしてください。

**sqlcode:** -1163

**sqlstate:** 42997

---

**SQL1164N** **SQL** ステートメントで使用されているタイプ *type* の **SQLDA** あるいはホスト変数が無効です。理由コード *reason-code*、ホスト変数 **/SQLVAR** 番号 *var-number*。

**説明:** **SQLDA** あるいは **SQL** ステートメントのホスト変数を処理している間にエラーが起きました。

呼び出しパラメーター・リストはプリコンパイラーで作成されますが、アプリケーション・プログラマーがプリコンパイラーの出力を修正し、アプリケーション・プログラムで **SQL** で始まる変数名を使用するか、あるいは別の方法で呼び出しパラメーター・リストを上書きする場合には正しくない可能性があります。

また **SQLDA** がアプリケーションによって直接渡される場合正しく初期化されない可能性があります。

ホスト変数 **/SQLDA** タイプ :

- 1 入力ホスト変数あるいは **SQLDA**
- 2 出力ホスト変数あるいは **SQLDA**

ホスト変数を指定した **SQL** ステートメントでは、ホスト変数番号を使用してステートメント

(あるいはコンパウンド SQL の場合はサブステートメント) の最初からカウントし無効なホスト変数を探し出すことができます。SQLDA を使用したステートメントでは SQLVAR 番号が無効な SQLVAR の検出に使用されます。入力 SQLDA では入力ホスト変数あるいは SQLVAR をカウントするだけです。出力も同様です。この番号の基本は 1 で、すべての理由コードに適用できるわけではないことに注意してください。理由コードは以下のように解釈されます。

- 1 SQLDA.SQLN が SQLDA.SQLD より小さい。
- 2 SQLVAR.SQLTYPE が無効である。
- 3 SQLVAR.SQLLEN あるいは SQLVAR2.SQLLONGLEN で指定した長さが SQLVAR.SQLTYPE で与えられた SQL タイプに対して間違っている。
- 4 ラージ・オブジェクト SQLVAR はあるが、SQLDA.SQLDAID の SQLDOUBLED フィールドが '2' に設定されていない
- 5 入力 varchar が現行の長さ (varchar 自身の長さフィールドから) が最大長より大きくなっているものを提供している。最大長は宣言 (ホスト変数の場合) あるいは SQLVAR.SQLLEN の設定 (ユーザー定義 SQLDA の場合) によって判別されます。
- 6 入力ラージ・オブジェクトの現行の長さ (LOB 自身の長さフィールドあるいは SQLVAR2.SQLDATALEN ポインターで示される) が最大長より大きく渡される。最大長は宣言 (ホスト変数の場合) あるいは SQLVAR2.SQLLONGLEN の設定 (ユーザー定義 SQLDA の場合) によって判別されます。
- 7 2 バイト文字ラージ・オブジェクトには SQLVAR2.SQLDATALEN ポインターで

示される奇数値があり、これが常にバイトで、DBCLOB に対してもそうである。

- 8 SQLDATA ポインターが無効であるか、あるいは不十分なストレージを示している
- 9 SQLIND ポインターが無効であるか、あるいは不十分なストレージを示している
- 10 SQLDATALEN ポインターが無効であるか、あるいは不十分なストレージを示している
- 11 入力ホスト変数 /SQLVARS の特定値が現行 SQL ステートメントで予想される

ステートメントは処理できません。

**ユーザーの処置:** 示されたエラーについて、アプリケーション・プログラムを調べてください。プログラマーは、プリコンパイラー出力を変更すべきではないことに注意してください。

**sqlcode:** -1164

**sqlstate:** 07002

---

**SQL1165W** 値がホスト変数のデータ・タイプの範囲外なので、その値をホスト変数に割り当てることができません。

**説明:** ホスト変数リストへの FETCH、VALUES、または SELECT は、ホスト変数が検索された値を保留するのに十分な大きさでないため、失敗しました。

このステートメント処理は -2 の null 標識を戻し続行しました。

**ユーザーの処置:** 表定義が現在のものであり、ホスト変数が適切なデータ・タイプであることを確認してください。SQL データ・タイプの範囲については、SQL リファレンスを参照してください。

**sqlcode:** +1165

**sqlstate:** 01515



---

**SQL1166W** ゼロによる除算が試みられました。

**説明:** 算術式の処理でゼロの除算が起きました。この警告は、警告の原因となった行とは別の行で戻ってくる場合があります。たとえば、述部での算術式で、または照会がシステム一時表を使用して処理を行っている場合、これは起こります。null 標識が -2 に設定されているときにはいつでも警告が戻されるために再度発行される可能性があります。

ステートメント処理は続けられ、null を除算式の結果として使用し null 標識 -2 を戻すことが考えられます。

**ユーザーの処置:** SQL ステートメントを調べて、問題の原因を判別してください。問題がデータによるものであれば、エラーが起きたときに処理されていたデータを調べてください。

**sqlcode:** +1166

**sqlstate:** 01564

---

**SQL1167W** 算術オーバーフロー、またはその他の算術例外が発生しました。

**説明:** 算術式の処理が算術オーバーフロー、アンダーフロー、あるいは他の算術例外を起こしました。この警告は、警告の原因となった行とは別の行で戻ってくる場合があります。たとえば、述部での算術式で、または照会がシステム一時表を使用して処理を行っている場合、これは起こります。null 標識が -2 に設定されているときにはいつでも警告が戻されるために再度発行される可能性があります。

ステートメント処理は続けられ、null を算術式の結果として使用し null 標識 -2 を戻すことが考えられます。

**ユーザーの処置:** SQL ステートメントを調べて、問題の原因を判別してください。問題がデータによるものであれば、エラーが起きたときに処理されていたデータを調べてください。データ・タイプの有効範囲については、SQL リファレンスを参照してください。

**sqlcode:** +1167

**sqlstate:** 01519

---

**SQL1178N** *object-name* と呼ばれる連合 *object-type* は、連合データベース・オブジェクトを参照していません。

**説明:** *object-name* で識別されているタイプ *object-type* のオブジェクトは、キーワード FEDERATED を使って定義されていますが、ステートメントの全選択は連合データベース・オブジェクトを参照していません。

ステートメントは処理できません。

**ユーザーの処置:** キーワード FEDERATED をステートメントから除去してください。

**sqlcode:** -1178

**sqlstate:** 429BA

---

**SQL1179W** 呼び出し側がデータ・ソース・オブジェクトについて必要な権限を持っていることを、*object-name* と呼ばれる *object-type* が必要としていると思われる。

**説明:** *object-name* で識別されているオブジェクトは、データ・ソースに実際のデータが存在する連合オブジェクト (OLE DB 表関数、連合ルーチン、連合ビュー、またはニックネームなど) を参照しています。データ・ソース・データにアクセスしている場合、ユーザー・マッピングおよび許可検査は、操作を開始したユーザーに基づいています。

*object-type* が SUMMARY TABLE であれば、操作はマテリアライズ照会表のデータをリフレッシュしています。リフレッシュを行う REFRESH TABLE または SET INTEGRITY ステートメントを呼び出したユーザーに、データ・ソースにある基礎データ・ソース・オブジェクトにアクセスするための権限が必要だと思われる。

*object-type* VIEW であれば、データ・ソースにあ

る基礎データ・ソース・オブジェクトにアクセスするための権限が、ビューのユーザーに必要だと思われる。

*object-type* が PROCEDURE、FUNCTION、または METHOD の場合、ルーチンの呼び出し元は、データ・ソースにあるそのルーチンの SQL ステートメントの基礎データ・ソース・オブジェクトへのアクセスに必要な特権を持っていることが要求される可能性があります。

いずれの場合も、データ・ソース・オブジェクトへのアクセスが試行されたときに、許可エラーが発生する可能性があります。

**ユーザーの処置:** オブジェクトに対する特権を付与するだけでは、データ・ソースからそのデータにアクセスする操作をサポートするのに十分ではない場合があります。基礎データ・ソース・オブジェクトのデータ・ソースで、ユーザー・アクセスを付与する必要がある可能性があります。

**sqlcode:** +1179

**sqlstate:** 01639

---

**SQL1180N** ルーチン *routine-name* (特定名 *specific-name*) により、OLE エラーが発生しました。

**HRESULT=***hresult* 診断テキスト:  
*message-text*

**説明:** DB2 は、ユーザー定義関数 (UDF) またはストアード・プロシージャ *routine-name* (特定名 *specific-name*) との通信を試行中、OLE エラー・コードを受け取りました。HRESULT *hresult* が戻された OLE エラー・コードで、*message text* が検索されたエラー・メッセージです。

以下に、エラー・メッセージ、HRESULTS、および考えられる原因のリストの一部を示します。エラー・メッセージ・テキストは OLE によって変更される可能性があり、新規のエラー・コードが OLE によって追加される場合もあります。

**不明なインターフェース (0x80020001):**

指定の OLE オブジェクトは IDispatch インターフェースをサポートしません。

**タイプの不一致 (0x80020005):**

SQL データ・タイプの 1 つ以上のものが方式引き数のデータ・タイプと一致しません。

**不明な名前 (0x80020006):**

指定の方式名は指定の OLE オブジェクトで見つかりません。

**無効なパラメーター数 (0x8002000E):**

方式に渡された引き数の数が方式で受け入れた引き数の数と相違しています。

**無効なクラス・ストリング (0x800401F3):**

指定の ProgID あるいは CLSID が無効です。

**クラスが登録されていない (0x80040154):**

CLSID が正しく登録されません。

**アプリケーションが見つからない (0x800401F5):**

ローカル・サーバー EXE が見つかりません。

**クラスの DLL が見つからない (0x800401F8):**

処理中の DLL が見つかりません。

**サーバー実行の失敗 (0x80080005):**

OLE オブジェクトの作成ができません。

**ユーザーの処置:** 特殊用語の意味を含む、完全な文書については、OLE プログラマー参考書を参照してください。

**sqlcode:** -1180

**sqlstate:** 42724

---

**SQL1181N** ルーチン *routine-name* (特定名 *specific-name*) が、記述 *message-text* とともに例外を起こしました。

**説明:** ユーザー定義関数 (UDF) またはストアード・プロシージャ *routine-name* (特定名 *specific-name*) が例外を起こしました。メッセージ・テキストには、ルーチンによって返された例

外のテキスト記述が示されています。

**ユーザーの処置:** ユーザーは例外の意味を理解する必要があります。ルーチンの作成者に連絡してください。

**sqlcode:** -1181

**sqlstate:** 38501

---

**SQL1182N** ユーザー定義関数 *function-name* が指定された **OLE DB provider** のデータ・ソース・オブジェクトを初期化できませんでした。

**HRESULT=***hresult* 診断テキスト:  
*message-text*

**説明:** 指定された OLE DB provider の OLE DB データ・ソース・オブジェクトをインスタンス化または初期化できませんでした。 *hresult* は返された OLE DB エラー・コードで、 *message-text* は検索されたエラー・メッセージです。

以下に HRESULTS および考えられる原因のリストの一部を示します。

**0x80040154**

クラス (OLE DB provider) が登録されていません。

**0x80040E73**

指定された初期化ストリングが指定に準拠していません。

**0x80004005**

指定されていないエラー (初期化中)。

**ユーザーの処置:** OLE DB provider の正しい登録と、接続ストリング内のパラメーターの初期化を確認します。OLE DB コア・コンポーネント内のデータ・リンク API 接続ストリング構文と HRESULT コードの完全なドキュメンテーションについては、Microsoft OLE DB Programmer's Reference and Data Access SDK を参照してください。

**sqlcode:** -1182

**sqlstate:** 38506

---

**SQL1183N** ユーザー定義関数 *function-name* が指定された **OLE DB provider** から **OLE DB** エラーを受け取りました。 **HRESULT=***hresult* 診断テキスト: *message-text*

**説明:** 指定された OLE DB provider が OLE DB エラー・コードを返しました。 *hresult* は返された OLE DB エラー・コードで、 *message-text* は検索されたエラー・メッセージです。

以下に HRESULTS および考えられる原因のリストの一部を示します。

**0x80040E14**

コマンドに、1 つ以上のエラーが入っていました。たとえば、パススルー・コマンド・テキストの構文エラーです。

**0x80040E21**

エラーが発生しました。たとえば、提供された columnID が無効です (DB\_INVALIDCOLUMN)。

**0x80040E37**

指定された表が存在しません。

**ユーザーの処置:** HRESULT コードの完全なドキュメンテーションについては、Microsoft OLE DB Programmer's Reference and Data Access SDK を参照してください。

**sqlcode:** -1183

**sqlstate:** 38506

---

**SQL1184N** 1 つ以上の **EXPLAIN** 表が、現在のバージョンの **DB2** を使用して作成されていません。

**説明:** DB2EXMIG を使用して表が移行されるか、DB2 の現在のバージョンの EXPLAIN.DDL CLP スクリプトを使用して表がドロップまたは再作成されるまで、EXPLAIN はこれらの表に挿入できません。

**ユーザーの処置:** DB2EXMIG を使用して表を移行するか、DB2 の現在のバージョンの

EXPLAIN.DDL CLP スクリプトを使用して表をドロップまたは再作成してください。コマンドを再発行してください。

**sqlcode:** -1184

**sqlstate:** 55002

---

**SQL1185N** **FEDERATED** *value* が、パッケージのバインドで誤って使用されています。

**説明:** *value* が NO であれば、パッケージ内の少なくとも 1 つの静的 SQL ステートメントに、ニックネーム、または OLE DB 表関数か OLE DB プロシージャのいずれかへの参照が入っています。この場合、パッケージをバインドするために FEDERATED YES を指定する必要があります。

*value* が YES であれば、パッケージ内の静的 SQL ステートメントに、ニックネーム、または OLE DB 表関数か OLE DB プロシージャのいずれかへの参照が入っていません。この場合、パッケージをバインドするために FEDERATED NO を指定する必要があります。

パッケージは作成されていません。

**ユーザーの処置:** 正しい FEDERATED オプションを指定してください。

---

**SQL1186N** **タイプ** *object-type*、名前 *object-name* のオブジェクトが、**FEDERATED** 文節が指定されずに、統合されたオブジェクトになるように変更されているか、または統合されたオブジェクトとして作成されています。

**説明:** *object-name* で識別されているオブジェクトは、データ・ソースに実際のデータが存在する連合オブジェクト (OLE DB 表関数、連合ルーチン、連合ビュー、またはニックネームなど) を参照しています。

非連合ビューが変更中であり、全選択が現在、直

接または間接的に連合データベース・オブジェクトを参照している場合、FEDERATED を指定する必要があります。

連合ビューが変更中であり、全選択がまだ、直接または間接的に連合データベース・オブジェクトを参照している場合、NOT FEDERATED を指定する必要があります。

マテリアライズ照会表を作成中であり、全選択が、直接または間接的に連合データベース・オブジェクトを参照している場合、NOT FEDERATED を指定することはできません。

SQL ルーチンを作成中であり、そのルーチンが、直接または間接的に連合データベース・オブジェクトを参照している場合、NOT FEDERATED を指定することはできません。

**ユーザーの処置:** 非連合ビューを連合ビューにするよう変更するには、FEDERATED 文節を指定してください。

統合されたビューへ継続する統合されたビューを変更するには、NOT FEDERATED 文節を指定しないでください。

全選択が連合データベース・オブジェクトを直接または間接的に参照するマテリアライズ照会表を作成するには、NOT FEDERATED 文節を指定しないでください。

連合データベース・オブジェクトを直接または間接的に参照する SQL ルーチンを作成するには、NOT FEDERATED 文節を指定しないでください。

**sqlcode:** -1186

**sqlstate:** 429BA

---

**SQL1187W** データベースは正常に作成または移行されましたが、詳細デッドロック・イベント・モニター *event-monitor-name* を作成中に、エラーが発生しました。詳細デッドロック・イベント・モニターは作成されていません。

**説明:** CREATE DATABASE または MIGRATE DATABASE は、データベースに対して詳細デッドロック・イベント・モニターを作成できませんでした。

**ユーザーの処置:** 必要であれば、詳細デッドロック・イベント・モニターを作成してください。

---

**SQL1188N** SELECT または VALUES ステートメントの列 *src-col-num* が、表列 *tgt-col-num* と非互換です。ソース列は **sqltype** *src-sqltype*、ターゲット列は **sqltype** *tgt-sqltype* です。

**説明:** ソース列とターゲット列が非互換です。SELECT または VALUES ステートメントの列の順序に誤りがあるか、METHOD P または METHOD N 指定の順序に誤りがある可能性があります。

**ユーザーの処置:** ソース列とターゲット列が互換性を持つようにコマンドを訂正して、コマンドを再発行してください。

---

**SQL1189N** 指定された **LOAD METHOD** は、指定されたファイル・タイプには無効です。

**説明:** 指定されたファイル・タイプに、無効な METHOD が指定されました。

**ユーザーの処置:** 別の METHOD を使ってコマンドを再発行してください。

---

**SQL1190N** **LOAD** ユーティリティは、**SQLCODE** *sqlcode*、**SQLSTATE** *sqlstate*、およびメッセージ・トークン *token-list* を検出しました。

**説明:** LOAD ユーティリティは、SQLCODE *sqlcode* のエラーを検出して、処理を停止しました。

**ユーザーの処置:** メッセージ・リファレンスで、SQLCODE *sqlcode* を参照してください。必要な訂正アクションを実行して、コマンドを再発行してください。

---

**SQL1191N** **METHOD** パラメーターに指定された列 *col-spec* は存在しません。

**説明:** 列 *col-spec* は、照会の結果に存在しません。

**ユーザーの処置:** 無効な列指定を訂正して、コマンドを再発行してください。

---

**SQL1192W** 現在のファイル・タイプに指定された入力ソースの数を多過ぎます。許可される最大数は、*max-input-sources* です。

**説明:** 指定されたファイル・タイプでは、単一ロードに対して *max-input-sources* を超える入力ソースを指定することは許可されません。

**ユーザーの処置:** *max-input-sources* を超えない入力ソースの数を指定して、コマンドを再発行してください。

---

**SQL1193I** ユーティリティは、**SQL** ステートメント *statement* からのデータのロードを開始しています。

**説明:** これは、SQL ステートメントからのロードが開始されたことを示す通知メッセージです。

**ユーザーの処置:** アクションは必要ありません。

---

**SQL1194W** 列番号 *col-num* に対応する行 *row-num* の値が無効です。この行は拒否されています。

**説明:** ターゲット列 *col-num* に対応する値がその列に有効な値でないため、行番号 *row-num* は拒否されました。

考えられる理由は以下のとおりです。

- 数値がそのターゲット列の最小または最大範囲内に入っていません。
- ターゲット列に対して文字値が長過ぎます。

**ユーザーの処置:** 入力値を調べてください。必要に応じて、ターゲット列 *col-num* をより適切に定義された列で置き換えて新規表を定義し、処理を繰り返してください。

---

**SQL1195N** ツール・カタログ・データベース *database-name* の作成または初期化が *server-name* で失敗しました。  
**SQLCODE = *SQLCODE***。

**説明:** ツール・カタログ・データベース *database-name* の作成または初期化が *server-name* で失敗しました。SQLCODE = *SQLCODE*。

**ユーザーの処置:** 操作から戻された SQLCODE についてユーザー応答を参照してください。

---

**SQL1196W** バックアップが正常に完了しました。ただし、1 つ以上のデータ・リンク・ファイル・マネージャーに接触できません。このバックアップ・イメージのタイム・スタンプは *backup-image-timestamp* です。

**説明:** バックアップが正常に完了しました。バックアップ処理中、1 つ以上のデータ・リンク・ファイル・マネージャーが利用不能でした。そのデータ・リンク・ファイル・マネージャーが利用可能になったときに、そのデータ・リンク・ファイル・マネージャーから追加の検査が必要です。

**ユーザーの処置:** バックアップ処理に利用不能だったデータ・リンク・ファイル・マネージャーを立ち上げてください。すべてのデータ・リンク・ファイル・マネージャーへの接触が可能になると、DB2 は自動的にこのバックアップ・イメージの最終検査を完了します。

---

**SQL1197N** コマンド *command* のキーワード *keyword* に対して無効な値が指定されました。

**説明:** *command* に指定された *keyword* 値は無効です。数値の場合は、値が定義された範囲外にある可能性があります。それ以外のタイプの場合は、指定された値が、定義されている有効な値のセットにありません。

**ユーザーの処置:** *command* の解説ドキュメンテーションから、有効な値を判別し、適切な変更を行ってください。このエラーが API によって戻された場合は、*command* API の解説処理から、該当する API オプションに有効な値を判別し、適切な変更を行ってください。

---

**SQL1198N** このコマンドは、現在の下位レベルのクライアント / サーバー構成ではサポートされていません。理由コード = *reason-code*

**説明:** 入力されたコマンドは、V8 より前のクライアントまたはサーバーに関連する、現在の構成ではサポートされていません。以下の理由コードが考えられます。

- 1 V8 以降のゲートウェイを介した、V8 より前のクライアントから DB2 UDB サーバーへのスクロール可能カーソルはサポートされていません。
- 2 V8 以降のゲートウェイを介した、V8 より前のクライアントから DB2 サーバーへのユーティリティー・コマンドはサポートされていません。
- 3 V8 以降のゲートウェイを介して V8 よ



り前のサーバーにアクセスすることはサポートされていません。

- 4 V8 以降のクライアントから V8 より前のサーバーへのユーティリティ・コマンドはサポートされていません。
- 98 V8 以降のクライアントから V8 より前のサーバーへの操作はサポートされていません。
- 99 V8 より前のクライアントから V8 以降のサーバーへの操作はサポートされていません。

**ユーザーの処置:** アクションは理由コードに基づいて、以下のように行います。

- 1 ゲートウェイを介して下位レベルのクライアントからスクロール可能カーソルを使用しない。クライアント / サーバーの直接接続を使用するか、クライアントを互換性のあるリリースにアップグレードしてください。
- 2 ゲートウェイを介して下位レベルのクラ

## SQL1200 - SQL1299

**SQL1200N object** パラメーターが無効です。

**説明:** COLLECT DATABASE STATUS 関数呼び出しの object パラメーターに指定された値が無効です。有効な値は以下のとおりです。

### SQL\_E\_DATABASE

状況が単一データベースに対して収集されることを示します。

### SQL\_E\_DRIVE

状況が単一パス上のすべての LOCAL データベースに対して収集されることを示します。

### SQL\_E\_LOCAL

状況がすべての LOCAL データベースに対して収集されることを示します。

コマンドは処理されません。

クライアントからユーティリティ・コマンドを発行しない。クライアント / サーバーの直接接続を使用するか、クライアントを互換性のあるリリースにアップグレードしてください。

- 3 互換性のあるレベルのゲートウェイを使用するか、(ゲートウェイを使用しない、) サーバーへの直接接続を構成する。
- 4 互換性のあるクライアント・レベルを使用して V8 ユーティリティ・コマンドを発行する。
- 98 互換性のあるクライアント・レベルを使用してこのコマンドを発行するか、サーバーのコード・レベルをアップグレードする。
- 99 互換性のあるサーバー・レベルを使用してこのコマンドを発行するか、クライアントのコード・レベルをアップグレードする。

**ユーザーの処置:** object パラメーターを訂正して、もう一度 COLLECT DATABASE STATUS 関数呼び出しを行ってください。

**SQL1201N status** パラメーターが無効です。

**説明:** COLLECT DATABASE STATUS 関数呼び出しの status パラメーターに指定された値が無効です。有効な値は以下のとおりです。

### SQL\_E\_SYSTEM

システム状況が収集されます。

### SQL\_E\_DATABASE

システム状況とデータベース状況が収集されます。

### SQL\_E\_ALL

システム状況、データベース状況、ユーザー状況が収集されます。

コマンドは処理されません。

**ユーザーの処置:** `status` パラメーターを訂正して、もう一度 `COLLECT DATABASE STATUS` 関数呼び出しを行なってください。

---

**SQL1202N** 操作状況がまだ集められていません。

**説明:** `GET NEXT DATABASE STATUS BLOCK` 関数呼び出しまたは `FREE DATABASE STATUS RESOURCES` 関数呼び出し内の `handle` パラメーターに指定された値が無効です。ハンドルは、`COLLECT DATABASE STATUS` 関数呼び出しから返される正の関数値でなければなりません。

これは、処理から行なわれた 2 度目の `COLLECT DATABASE STATUS` 呼び出しです。最初の `COLLECT DATABASE STATUS` 呼び出しは終了し、そのハンドルは使用できません。

コマンドは処理されません。

**ユーザーの処置:** `handle` パラメーターを訂正して、もう一度 `COLLECT DATABASE STATUS` 関数呼び出しを行ってください。

---

**SQL1203N** このデータベースには、接続しているユーザーがありません。

**説明:** データベースのユーザー状況が要求されましたが、そのデータベースにはユーザーが接続されていません。

コマンドは処理されません。

**ユーザーの処置:** データベース名と接続状況を確認してください。現在使用されているデータベースを使用して、コマンドを再発行してください。

---

**SQL1204N** コード・ページ `code-page` とテリトリー・コード `territory-code` のいずれか、またはその両方が、インストールされているデータベース・マネージャーのバージョンでサポートされていません。

**説明:** このバージョンのデータベース・マネージャーは、アクティブ・コード・ページまたはテリトリー・コード、あるいはその両方をサポートしていません。

コマンドは処理されません。

このバージョンのデータベース・マネージャーがサポートしているアクティブ・コード・ページとテリトリー・コードを選択してください。

連合システム・ユーザー: 必要に応じて、要求を拒否しているデータ・ソースの問題を分離し (障害が起きたデータ・ソースの識別手順については、「トラブルシューティング・ガイド」を参照)、次に連合サーバーとデータ・ソースの両方でサポートされているアクティブ・コード・ページとテリトリー・コードを選択してください。

**ユーザーの処置:** 現在のプログラムを終了して、オペレーティング・システムに戻ってください。

**sqlcode:** -1204

**sqlstate:** 22522

---

**SQL1205N** 指定されているコード・ページ `code-page` かテリトリー・コード `territory` のいずれか、またはその両方が無効です。

**説明:** このバージョンの DB2 は、`Create Database` コマンドによって指定されたアクティブ・コード・ページかテリトリー・コード、またはその両方をサポートしていません。

コマンドは処理されません。

**ユーザーの処置:** DB/2 でサポートされている有効なコード・ページとテリトリー・コードの詳細については、`コマンド・リファレンス` の中で



Create Database コマンドを参照してください。

---

**SQL1206N PRUNE LOGFILE**はこのデータベース構成ではサポートされません。

**説明:** PRUNE LOGFILE は以下の場合にはサポートされません。

1. LOGRETAIN と USEREXIT の両方が NO に設定されている。
2. アクティブなログ・ファイル・パスがロー・デバイスに設定されている。

**ユーザーの処置:** このデータベースに PRUNE LOGFILE コマンドを実行しないでください。

---

**SQL1207N コミュニケーション・マネージャー構成ファイル *name***が見つかりません。

**説明:** CATALOG NODE コマンドに指定されたコミュニケーション・マネージャー構成ファイル名が、指定されたパスまたはデフォルト・ドライブの CMLIB ディレクトリーに見つかりませんでした。

コマンドは処理されません。

**ユーザーの処置:** 正しいコミュニケーション・マネージャー構成ファイル名とパスを使用して、コマンドを再発行してください。

---

**SQL1209W CATALOG NODE** 関数に指定された **partner\_lu** 名 *name* が存在しません。名前が作成されました。

**説明:** CATALOG NODE 関数に指定された論理パートナー装置名が、デフォルト・ドライブの CMLIB ディレクトリーに存在するコミュニケーション・マネージャー構成ファイルにありません。

示された名前が LU プロファイルが作られました。

**ユーザーの処置:** アクションは必要ありません。

---

**SQL1210W 1** つ以上の DOS リクエスター **/WINDOWS** リクエスター構成ファイル・パラメーターにデフォルト値が返されました。

**説明:** 1 つ以上の DOS リクエスター **/WINDOWS** リクエスター構成ファイル・パラメーターに、デフォルト値が使用されます。パラメーターが、DOS リクエスター **/WINDOWS** リクエスター構成ファイルに定義されていないか、この構成ファイルがオープンできないか、またはファイルの読み取り中にエラーが起きた可能性があります。

**ユーザーの処置:** DOS リクエスター **/WINDOWS** リクエスター構成ファイルが適切なパスに存在し、パラメーターが明示的に定義されていることを確認してください。

---

**SQL1211N コンピューター名 *name*** が無効です。

**説明:** カタログ・コマンドの NPIPE プロトコル構造で指定されたコンピューター名は無効です。コンピューター名のサイズは 15 文字またはそれ以下でなくてはなりません。

コマンドは処理されません。

**ユーザーの処置:** コンピューター名が有効であるかを確認し、コマンドを再発行してください。

---

**SQL1212N インスタンス名 *name*** が無効です。

**説明:** カタログ・コマンドで指定されたこのインスタンス名は無効です。インスタンス名のサイズは 8 文字またはそれ以下でなくてはなりません。

コマンドは処理されません。

**ユーザーの処置:** インスタンス名が有効であるかを確認し、コマンドを再発行してください。

---

**SQL1213N** パスワード変更 LU 名 *name* が無効です。

**説明:** CATALOG コマンドの APPN プロトコル構造に指定されたパスワード変更 LU (LU) 名が無効です。

パスワード変更 LU 名はリモート SNA LU 名であり、1 から 8 文字でなければなりません。有効な文字は A から Z、a から z、0 から 9、#、@ および \$ です。最初の文字は、英字または特殊文字 #、@、または \$ でなければなりません。小文字は、システムによって大文字に変更されます。

コマンドは処理されません。

**ユーザーの処置:** パスワード変更 LU 名に指定された文字を確認してください。

有効なパスワード変更 LU 名を指定してコマンドを再発行してください。

---

**SQL1214N** トランザクション・プログラム名 *name* が無効です。

**説明:** CATALOG コマンドの APPN プロトコル構造に指定されたトランザクション・プログラム (TP) 名が無効です。

TP 名はリモート SNA アプリケーション TP 名であり、1 から 64 文字でなければなりません。有効な文字は A から Z、a から z、0 から 9、#、@ および \$ です。最初の文字は、英字または特殊文字 #、@、または \$ でなければなりません。

コマンドは処理されません。

**ユーザーの処置:** TP 名に指定された文字を確認してください。

TP 名を指定してコマンドを再発行してください。

---

**SQL1215N** LAN アダプター・アドレス *address* が無効です。

**説明:** CATALOG コマンドの APPN プロトコル構造に指定された LAN アダプター・アドレスが無効です。

LAN アダプター・アドレスはリモート SNA LAN アダプター・アドレスであり、12 の 16 進数でなければなりません。

コマンドは処理されません。

**ユーザーの処置:** LAN アダプター・アドレスを確認してください。

有効な LAN アダプター・アドレスを指定してコマンドを再実行依頼してください。

---

**SQL1216N** 漢字データと漢字関数は、このデータベースではサポートされていません。

**説明:** データベースのコード・ページは、漢字データをサポートしません。データ・タイプ GRAPHIC、VARGRAPHIC、LONG VARGRAPHIC が、このデータベースには無効です。GRAPHIC リテラルと VARGRAPHIC スカラー関数が、このデータベースには無効です。

ステートメントは処理できません。

連合システム・ユーザー: この状態はデータ・ソースでも検出できます。

**ユーザーの処置:** 有効なデータ・タイプを使用して、コマンドを再発行してください。

**sqlcode:** -1216

**sqlstate:** 56031

---

**SQL1217N** REAL データ・タイプがターゲット・データベースによってサポートされていません。

**説明:** SQL 操作は入力または出力変数として REAL のデータ・タイプ (単精度浮動小数点数) を使用しています。REAL データ・タイプがこ

の要求のターゲット・データベースにサポートされていません。

このステートメントは処理できません。

**ユーザーの処置:** ご使用のアプリケーション内の SQL データ・タイプ `DOUBLE` と一致する宣言を使用する SQL データ・タイプ `REAL` に相当するホスト変数の宣言を置換してください。

**sqlcode:** -1217

**sqlstate:** 56099

---

**SQL1218N** 現在、バッファース・プール `buffpool-num` で使用可能なページはありません。

**説明:** 現在、バッファース・プールのすべてのページが使用中です。別のページの使用の要求が失敗しました。

ステートメントは処理できません。

**ユーザーの処置:** バッファース・プールは、すべてのデータベース処理またはスレッドにページ作成できるほど大きくはありません。バッファース・プールが小さすぎるか、アクティブ処理またはアクティブ・スレッドが多すぎます。

再実行すれば、このステートメントが正常である可能性があります。このエラーがしばしば発生する場合は、以下のいずれかのまたはすべてのアクションがこの失敗を防ぐのに役立ちます。

1. バッファース・プールのサイズを大きくする。
2. データベース・エージェントおよび / または接続の最大数を減らす。
3. 並列処理の最大数を減らす。
4. このバッファース・プールの表スペースのプリフェッチ・サイズを減らす。
5. いくつかの表スペースを別のバッファース・プールへ移動させる。

**sqlcode:** -1218

**sqlstate:** 57011

---

**SQL1219N** 専用仮想メモリーの割り振りができないために、要求が失敗しました。

**説明:** 要求を処理するための専用仮想メモリーが十分でないために、インスタンスが割り振られませんでした。これは、共有メモリー割り振りが他の (関連のない) 処理で行われた結果として起きる場合があります。

**ユーザーの処置:** 問題の解決法は、以下のとおりです。

- OS/2 の場合は、`min_priv_mem` 構成パラメーターを増やします。これにより、インスタンスが開始されたときに、もっと多くの専用仮想メモリーが予約されます。
- マシンで実行されているアプリケーション、特に共有メモリーを大量に使用するアプリケーションを停止してください。

**sqlcode:** -1219

**sqlstate:** 57011

---

**SQL1220N** データベース・マネージャー共有メモリー・セットを割り振ることができません。

**説明:** データベース・マネージャーが、共有メモリー・セットを割り振ることができませんでした。このエラーの原因としては、データベース・マネージャー、またはデータベース・マネージャーの処理が試みられていた環境に、十分なメモリー・リソースが存在しなかった可能性があります。この問題の原因となるメモリー・リソースには、以下が含まれます。

- システムに割り振られている共有メモリー ID の数
- システムが使用可能なページングまたはスワッピングの容量
- システムが使用可能な物理メモリーの容量

**ユーザーの処置:** 以下の 1 つ以上を行ってください。

- データベース・マネージャーとシステムで実行中の他のプログラムの要求を満たすだけのメモリー・リソースが、使用可能であることを確認してください。
- メモリー・セットに影響を与えるデータベース・マネージャー構成パラメーターを減らして、このメモリー・セットに対するデータベース・マネージャーの必要メモリー量を減らしてください。これらのパラメーターには、*maxagents*、*maxdari*、および *numdb* があります。
- 該当する場合は、システムを使用している他のプログラムを停止してください。

---

**SQL1221N** アプリケーション・サポート層ヒープを割り振ることができません。

**説明:** アプリケーション・サポート層ヒープを割り振ることができませんでした。このエラーの原因としては、データベース・マネージャー、またはデータベース・マネージャーの処理が試みられていた環境に、十分なメモリー・リソースが存在しなかった可能性があります。この問題の原因となるメモリー・リソースには、以下が含まれます。

- システムに割り振られている共有メモリー ID の数
- システムが使用可能なページングまたはスワッピングの容量
- システムが使用可能な物理メモリーの容量

**ユーザーの処置:** 以下の 1 つ以上を行ってください。

- データベース・マネージャーとシステムで実行中の他のプログラムの要求を満たすだけのメモリー・リソースが、使用可能であることを確認してください。
- aslheapsz* 構成パラメーターを減らしてください。
- 該当する場合は、システムを使用している他のプログラムを停止してください。

**sqlcode:** -1221

**sqlstate:** 57011

---

**SQL1222N** その要求の処理に使用できる十分なストレージが、アプリケーション・サポート層ヒープにありません。

**説明:** アプリケーション・サポート層ヒープのすべての利用可能なメモリーを使いきってしまいました。

**ユーザーの処置:** *aslheapsz* 構成パラメーターを増やしてください。

**sqlcode:** -1222

**sqlstate:** 57011

---

**SQL1223N** この要求を処理するためのエージェントを開始できませんでした。

**説明:** *maxagents* 構成パラメーターを超えてしまうために、要求は失敗しました。

**ユーザーの処置:** *maxagents* 構成パラメーターを増やすか、またはデータベースを使用するユーザー数を減らすか、もしくはその両方を行ってください。

**sqlcode:** -1223

**sqlstate:** 57019

---

**SQL1224N** データベース・エージェントが、要求を処理するために開始できなかったか、あるいはデータベース・システムのシャットダウンまたは強制コマンドによって終了されました。

**説明:** このメッセージは、以下の場合に出されません。

- データベース・サーバーでデータベース・マネージャーが始動していません。
- データベース・マネージャーが停止しました。
- データベース・マネージャーが、すでに最大数のエージェントを割り振っています。

- データベース・エージェントが、システム管理者により強制終了されました。
- 主要データベース・マネージャー処理の異常終了のために、データベース・エージェントが終了しました。
- アプリケーションがローカル・プロトコルで複数コンテキストを使用しています。この場合、接続の数は単一処理が接続できる共有メモリー・セグメントの数によって制限されます。たとえば、AIX での制限は 1 処理ごとに 10 の共有メモリー・セグメントになります。
- 接続しているときに、SYSADM 権限を持つユーザーが FORCE QUIESCE コマンドを発行しました。ご使用のユーザー ID にはデータベース / インスタンスの CONNECT QUIESCE 権限がないか、あるいは CONNECT QUIESCE 権限のあるグループに属しているため、データベース / インスタンスから切断されました。

その他の連合サーバーのケースは以下の通りです。

- オペレーティング・システム・レベルでユーザーごとの処理の最大数 (AIX では maxuproc) を超えました。
- TCP/IP プロトコルを使用しているクライアント / サーバー環境では、クライアントで TCP/IP サービス名に割り当てられたポート番号はサーバーのポート番号と異なります。

この状態は連合サーバーまたはデータ・ソースで検出できます。

#### ユーザーの処置:

- データベース要求を再発行してください。接続が確立できない場合は、データベース・マネージャーが正常に始動していることを確認してください。さらに、*maxagents* データベース・マネージャー構成パラメーターが適切に構成されていることを確認してください。
- データベース / インスタンスを静止解除する SYSADM を持つか、ユーザーを静止グループに追加してください。

連合システム・ユーザーは以下も行う必要があります。

- 問題を、要求を拒否したデータ・ソースと分離して、(問題判別の手引きを参照して、失敗するデータ・ソースを識別するための手続きにしたがってください。) 通信サブシステムがアクティブであること、データベース・マネージャーおよび要求された通信プロトコル・サーバーがデータベース・サーバーで始動していることを確認する。
- AIX オペレーティング・システムでは、maxuproc の設定をして、必要があれば変更する。maxuproc は、与えられた連合サーバーの下で実行できる処理の数を制限します。デフォルト設定は 40 です。

maxuproc の現在の設定をチェックするには、以下のコマンドを使用します。

```
lsattr -E -l sys0
```

指定された連合サーバーの下で現在実行中の処理の数を表示するには以下のコマンドを使用してください。

```
ps -ef | grep instdjl | wc -l
```

この場合は “instdjl” は連合サーバーのインスタンス名です。

maxuproc を変更するには、以下のコマンドを使用します。

```
chdev -l sys0 -a maxuproc='nn'
```

この場合 **nn** は maxuproc の新規整数値です。

アプリケーションがローカル・プロトコルで複数コンテキストを使用している場合、アプリケーション内の接続数を減らすか、別のプロトコル (たとえば、TCP/IP) に切り替えてください。AIX バージョン 4.2.1 またはこれ以上の場合、環境変数 EXTSHM を ON に設定し、単一処理が接続できる共有メモリーのセグメント数を増やすことができます。

**sqlcode:** -1224

**sqlstate:** 55032

---

**SQL1225N** オペレーティング・システムの処理、スレッド、またはスワップ・スペースの限界に達したために、要求は失敗しました。

**説明:** オペレーティング・システムの処理、スレッド、またはスワップ・スペースの限界に達しました。管理通知ログでこの問題に関する詳細情報を調べてください。AIX システムでは、maxuproc 値が小さすぎる可能性があります。

OS/2 ベースのシステムでは、THREADS CONFIG.SYS 値が小さすぎる可能性があります。

**ユーザーの処置:** 管理通知ログを調べて、限界に達したものを判別し、その限界値を大きくしてください。

**sqlcode:** -1225

**sqlstate:** 57049

---

**SQL1226N** クライアント接続の最大数がすでに開始されています。

**説明:** クライアント接続数は、すでにシステム構成ファイルに定義された最大値と同等です。

この操作は処理できません。

**ユーザーの処置:** ほかのアプリケーションがデータベースから切断するまでお待ちください。1 つ以上のアプリケーションを並行して実行する必要がある場合は、max\_connections の値を増やしてください。新しい値は、次のデータベース・マネージャーが正常に始動した後に反映されません。max\_connections の値が max\_coordagents の値よりも大きい場合は、コンセントレーター機能がオンになります。

**sqlcode:** -1226

**sqlstate:** 57030

---

**SQL1227N** 列 *column* のカタログ統計 *value* は、ターゲット列の範囲外か、フォーマットが無効か、または他の統計との関係において矛盾があります。  
理由コード = *code*

**説明:** 更新可能カタログに指定された統計の値またはフォーマットは、無効であるか、範囲外であるか、または矛盾しています。値、範囲、およびフォーマットに対する最も一般的なチェックは以下のとおりです (*code* に対応):

- 1 数値統計は -1 または  $\geq 0$  でなければなりません。
- 2 パーセントを表す数値統計 (たとえば、CLUSTERRATIO) は、0 と 100 の間でなければなりません。
- 3 HIGH2KEY、LOW2KEY に関連する規則は、以下のとおりです。
  - HIGH2KEY、LOW2KEY の値のデータ・タイプは、対応するユーザー列と同じデータ・タイプでなければなりません。
  - HIGH2KEY、LOW2KEY の値の長さは、33 またはターゲット列のデータ・タイプの最大長より短くなければなりません。
  - 対応する列に 3 つ以上の異なる値が存在する場合は、常に HIGH2KEY が  $>$  LOW2KEY でなければなりません。列に 3 つ未満の異なる値が存在する場合は、HIGH2KEY が LOW2KEY と同じでもかまいません。
- 4 PAGE\_FETCH\_PAIRS に関連する規則は、以下のとおりです。
  - PAGE\_FETCH\_PAIRS 統計の個々の値は、ブランク区切り文字によって分離されていなければなりません。
  - PAGE\_FETCH\_PAIRS 統計の個々の値は、10 桁より大きくてはならず、最



大整数値 (MAXINT = 2147483647) より小さくしなければなりません。

- CLUSTERFACTOR が > 0 の場合は、常に有効な PAGE\_FETCH\_PAIRS 値でなければなりません。
- 単一の PAGE\_FETCH\_PAIR 統計には、正確に 11 対が存在する必要があります。
- PAGE\_FETCH\_PAIRS のバッファー・サイズ項目は、昇順の値でなければなりません。また、PAGE\_FETCH\_PAIRS 項目のいずれのバッファー・サイズの値も、NPAGES が対応する表のページ数である MIN( NPAGES、524287) より大きくすることはできません。
- PAGE\_FETCH\_PAIRS の "fetches" 項目は、NPAGES より少ない個別の fetches 項目を持たずに、降順の値でなければなりません。また、PAGE\_FETCH\_PAIRS 項目の "fetch" サイズの値は、対応する表の CARD (カーディナリティー) 統計より大きくしないでください。
- バッファー・サイズの値が 2 つの連続した列で同じ場合は、ページ・フェッチの値もその両方で同じでなければなりません。

5 CLUSTERRATIO と CLUSTERFACTOR に関連する規則は、以下のとおりです。

- CLUSTERRATIO の有効な値は -1、または 0 と 100 の間です。
- CLUSTERFACTOR の有効な値は -1、または 0 と 1 の間です。
- CLUSTERRATIO または CLUSTERFACTOR のどちらかは、常に -1 でなければなりません。
- CLUSTERFACTOR が正の値の場合は、有効な PAGE\_FETCH\_PAIR 統計が伴わなければなりません。

6 列 (SYSCOLUMNS の COLCARD 統計) または列グループ (SYSCOLGROUPS の COLGROUPCARD) のカーディナリティーは、対応する表 (SYSTABLES の CARD 統計) のカーディナリティーより大きくすることはできません。

7 データ・タイプ LONG VARCHAR、LONG VARGRAPHIC、BLOB、CLOB、DBCLOB、またはユーザー定義の構造化タイプを持つ列で、統計はサポートされていません。

8 統計が、このエンティティーに関連する別の統計と矛盾しているか、またはこのコンテキストでは無効です。

**ユーザーの処置:** 新しいカタログ統計が、上記の範囲 / 長さ / 形式チェックを満足していることを確認してください。

統計に対する更新が、内部リレーションシップ (たとえば、cardinality) において整合性を持っていることを確認してください。

**sqlcode:** -1227

**sqlstate:** 23521

---

**SQL1228W DROP DATABASE** は完了しましたが、データベース別名またはデータベース名 *name* が *num* ノードに見つかりません。

**説明:** drop database コマンドは正常に完了しましたが、データベース別名またはデータベース名が見つからないノードがあります。DROP DATABASE AT NODE がこのノードですすでに実行されている可能性があります。

**ユーザーの処置:** これは注意メッセージです。応答は必要ありません。

---

**SQL1229N** 現行トランザクションがシステム・エラーのためロールバックしています。

**説明:** 以下のいずれかが起こりました。

1. ノード障害または通信障害といったシステム・エラーが発生しました。アプリケーションは直前の COMMIT にロールバックされません。

DB2 ユーティリティ関数では、各関数は以下ようになります。

#### インポート

アプリケーションがロールバックされます。COMMITCOUNT パラメータを使用した場合、操作が直前のコミット・ポイントにロールバックします。

**Reorg** 操作を打ち切り、再実行する必要があります。

**再分散** 操作は打ち切りますが、正常に終了している操作もある可能性があります。「続行」オプションで要求を再度出すと、失敗したところから操作を再始動します。

#### ロールフォワード

操作が打ち切られ、データベースはロールフォワード・ペンディング状態のままです。コマンドを再実行してください。

#### バックアップ / リストア

操作を打ち切り、再実行する必要があります。

2. FCM (高速コミュニケーション・マネージャー) コミュニケーションは、DB2 インスタンスのすべてのノードにおいて同じではありません。すべてのノードのサービス・ファイルをチェックし、ポートが同じであることを確認してください。ポートは以下の形式を使用して定義されました。

```
DB2_<instance> xxxx/tcp
DB2_<instance>_END xxxx/tcp
```

<instance> は DB2 インスタンス名、xxxx はポート番号です。これらのポート番号が DB2 リモート・クライアント・サポートに使用されていないことを確認してください。

#### ユーザーの処置:

1. 要求を再度試行してください。エラーが残る場合、管理通知ログでこの問題に関する詳細情報を調べてください。このエラーが最も多く発生する理由は、ノード障害のため、システム管理者に連絡して援助してもらうことが必要な場合があります。  
高速スピード・スイッチが使用されている SP 環境では、このエラーは高速スピード・スイッチで障害が起きる徴候です。
2. すべてのノードにおいて同じになるように、サービス・ポートを更新し、要求を再実行してください。

SQLCA の 6 番目の SQL エラー・フィールドが、ノード障害を検出したノード番号を示します。障害を検出したノードに関して、障害を起こしたノードを識別する db2diag ログにメッセージが入ります。

**sqlcode:** -1229

**sqlstate:** 40504

---

**SQL1230W** 指定されたエージェント ID の中に、強行できないものが少なくとも 1 つありました。

**説明:** Force コマンドに指定されたエージェント ID の中に、強行できないものが少なくとも 1 つありました。この警告の原因としては、以下が考えられます。

- 存在しないエージェント ID、または無効なエージェント ID が指定されました。



- エージェント ID の収集と Force コマンドの発行までの間に、エージェントがデータベース・マネージャーから切断されました。
- エージェント ID が実行できない並列エージェントに対して指定されました。

**ユーザーの処置:** 存在しないエージェント ID、または無効なエージェント ID を指定した場合は、有効なエージェント ID を使用して、コマンドを再発行してください。

---

**SQL1231N 無効な Force count が指定されました。**

**説明:** Force コマンドの count パラメーターに指定された値が有効ではありません。指定する値は、正の整数または SQL\_ALL\_USERS でなければなりません。0 の値はエラーになります。

**ユーザーの処置:** count の値を訂正して、コマンドを再発行してください。

---

**SQL1232N 無効な Force mode が指定されました。**

**説明:** Force コマンドの mode パラメーターに指定された値が有効ではありません。Force コマンドでは、非同期モードのみがサポートされています。パラメーターは、値 SQL\_ASYNCCH に設定される必要があります。

**ユーザーの処置:** mode を SQL\_ASYNCCH に設定してコマンドを再発行してください。

---

**SQL1233N この文節またはスカラー関数は、UCS-2 であるグラフィック・データのみでサポートされます。**

**説明:** UCS-2 はこのデータベースではサポートされていません。UCS-2 のサポートは、以下の場合に必要です。

- トラフィック・ストリング式を VARCHAR スカラー関数の最初の引き数として指定している

- 2 番目の引き数が指定されている場合、文字ストリング式を VARGRAPHIC スカラー関数の最初の引き数として指定している
- 形式 UX'hex-digits' を使用している UCS-2 16 進定数を指定している

ステートメントは処理できません。

**ユーザーの処置:** 有効なデータ・タイプを使用して、コマンドを再発行してください。

**sqlcode:** -1233

**sqlstate:** 560AA

---

**SQL1240N 静止状態の最大数に達しました。**

**説明:** すでに 5 つの処理によって静止されている表スペースに対して、静止状態の獲得が試みられました。

**ユーザーの処置:** いずれかの処理が静止状態を解放するのを待って、もう一度やり直してください。

---

**SQL1241N データベースの作成時に、無効な値が *tbls-name* 表スペースに指定されました。属性は *string* です。**

**説明:** 表スペース属性の値が範囲外です。create database api に使用されている sqletsdesc 構造の形式については、「アプリケーション開発ガイド」の「データ構造」セクションを参照してください。識別された属性は、この構造のフィールド名です。

**ユーザーの処置:** create database 要求を訂正してください。

---

**SQL1244W トランザクション・マネージャー・データベース *server-name* の切断が、次の COMMIT で起きます。**

**説明:** TM データベースとして活動中のデータベースに対して、切断が発行されました。次の COMMIT が処理されるまで、切断は完了しません。

**ユーザーの処置:** TM データベースとして活動中のデータベースを即座に切断する必要がある場合は、処理を続ける前に、COMMIT ステートメントを発行してください。

**sqlcode:** +1244

**sqlstate:** 01002

---

**SQL1245N** 接続限界に達しました。このクライアントからは、これ以上接続できません。

**説明:** 接続数が制限されているか、または事前定義される必要がある環境で、並列データベース接続の最大数に達しました。これが起きる可能性がある主な例には、NETBIOS プロトコルの使用があります。

**ユーザーの処置:** 可能なアクションは以下の通りです。

- SET CLIENT コマンドまたは API を使用して、「MAX NETBIOS CONNECTIONS」フィールドを必要な並列接続の最大数に設定してください。これは、接続を行う前に実行する必要があります。

**sqlcode:** -1245

**sqlstate:** 08001

---

**SQL1246N** 接続が存在している間は、接続設定を変更できません。

**説明:** 以下のいずれかが起きました。

- SET CLIENT API を使用するアプリケーションの接続設定の変更が試みられました。1 つ以上の接続が存在するために、拒否されました。
- アプリケーションに、DB2 コール・レベル・インターフェース API 呼び出しと組み込み SQL の入った関数への呼び出しの両方が入っており、接続管理が CLI API で呼び出されませんでした。

**ユーザーの処置:** 可能なアクションは以下の通りです。

- SET CLIENT API (sqlsetc または sqlgsetc) あるいは CLP コマンドを実行する前に、アプリケーションがすべてのサーバーから切断されていることを確認してください。
  - CLI がアプリケーションで使用されている場合は、すべての接続管理要求が、DB2 コール・レベル・インターフェース API 経由で発行されていることを確認してください。
- 

**SQL1247N** XA トランザクション処理環境で実行するアプリケーションには、**SYNCPOINT TWOPHASE** 接続設定を使用する必要があります。

**説明:** アプリケーションが、オプション SYNCPOINT ONEPHASE または SYNCPOINT NONE でプリコンパイルされているか、または SYNCPOINT 接続設定が、SET CLIENT API を使用して、上記のいずれかのオプションに変更されました。これらの設定は、トランザクション処理同期点コマンド (たとえば、CICS SYNCPOINT) を実行するアプリケーションには無効です。SYNCPOINT ONEPHASE が、デフォルト・プリコンパイラー・オプションであることに注意してください。

**ユーザーの処置:** 可能なアクションは以下の通りです。

- プリコンパイラー・オプション SYNCPOINT TWOPHASE を使用して、もう一度アプリケーションをプリコンパイルしてください。
- 他の処理の前に、接続オプションを SYNCPOINT TWOPHASE に設定するために、SET CLIENT API が呼び出されるように、プログラムを変更してください。

**sqlcode:** -1247

**sqlstate:** 51025

---

**SQL1248N** データベース *database-alias* は、トランザクション・マネージャーに定義されていません。

**説明:** トランザクション・マネージャーによってオープンされていないデータベースのアクセスが試みられました。2 フェーズ・コミットに使用するには、データベースがトランザクション・マネージャーに定義されている必要があります。

**ユーザーの処置:** 分散トランザクション処理環境のトランザクション・マネージャーにリソース・マネージャーとして、データベースを定義してください。たとえば、CICS 環境の場合、XAD ファイルにデータベースを追加し、データベース別名を XAD 項目の XAOpen スtringに指定する必要があります。

**sqlcode:** -1248

**sqlstate:** 42705

---

**SQL1251W** 経験的照会に戻されるデータはありません。

**説明:** データベースに未確定トランザクションも、終了して同期点処理に入るのを待っているトランザクションもありません。

**ユーザーの処置:** アクションは必要ありません。

---

**SQL1259N** データベース・パーティション *partition-list* 上のデータベース *db-name* に対する **ARCHIVE LOG** コマンドは、**SQLCODE** *SQLCODE* のために失敗しました。

**説明:** データベース・パーティション *partition-list* 上のデータベース *db-name* に対して **ARCHIVE LOG** コマンドを実行中に **SQL** エラーが発生しました。

**ユーザーの処置:** 「メッセージ・リファレンス」の **SQLCODE** でユーザー応答を参照し、**ARCHIVE LOG** コマンドをもう一度実行してください。

---

**SQL1260N** データベース *name* は、ノード *node-list* でのロールフォワード・リカバリー用に構成されていません。

**説明:** 指定されたデータベースは指定ノードで、ロールフォワード・リカバリー用に構成されません。", ..." がノード・リストの終わりに表示されている場合、完全なリストを見るには管理通知ログを調べてください。

データベースは指定のノードでロールフォワードされません。

(注: パーティション・データベース・サーバーを使用している場合、ノード番号は、エラーを起しているノードを示しています。そうでない場合、これは関係のないものなので無視してください。)

**ユーザーの処置:** 指定ノードでリカバリーが必要か確認して、次にこのノードのデータベースのバックアップで最新のバージョンをリストアしてください。

---

**SQL1261N** データベース *name* はノード *node-list* でロールフォワード・ペンディング状態ではありません。そのためこのノードでロールフォワードする必要はありません。

**説明:** 指定のデータベースは指定ノードでロールフォワード保留状態にありません。これはデータベースがリストアされていないか **WITHOUT ROLLING FORWARD** オプションでリストアされたか、ロールフォワード・リカバリーがこのノードで完了したために起こる場合があります。

", ..." がノード・リストの終わりに表示されている場合、完全なリストを見るには管理通知ログを調べてください。

データベースはロールフォワードされません。

(注: パーティション・データベース・サーバーを使用している場合、ノード番号は、エラーを起しているノードを示しています。そうでない場

合、これは関係のないものなので無視してください。)

**ユーザーの処置:** 以下を行ってください。

1. 指定ノードでリカバリーが必要か確認してください。
2. このノードのデータベースのバックアップ・バージョンをリストアしてください。
3. ROLLFORWARD DATABASE コマンドを実行してください。

---

**SQL1262N** データベース *name* のロールフォワードに指定されたポイント・イン・タイムが無効です。

**説明:** 停止ポイント・イン・タイム値に指定された `timestamp` パラメーターが有効ではありません。 `timestamp` は ISO 形式

(YYYY-MM-DD-hh.mm.ss.<sssss> で、YYYY は年、MM は月、DD は日、hh は時、mm は分、ss は秒を表し、sssss はオプションでマイクロ秒を表します) で入力する必要があります。

データベースはロールフォワードされません。

**ユーザーの処置:** `timestamp` が正しい形式で入力されていることを確認してください。

ROLLFORWARD DATABASE コマンドを実行する場合は、2105 より大きい年を指定していないことを確認してください。

---

**SQL1263N** アーカイブ・ファイル *name* は、ノード *node-number* 上のデータベース *name* に対して有効なログ・ファイルではありません。

**説明:** 指定のアーカイブ・ログ・ファイルが、データベース・ログ・ディレクトリーまたはオーバーフロー・ログ・ディレクトリーで見つかりましたが、ファイルが無効でした。

ロールフォワード・リカバリー処理は停止します。

(注：パーティション・データベース・サーバー

を使用している場合、ノード番号は、エラーを起こしているノードを示しています。そうでない場合、これは関係のないものなので無視してください。)

**ユーザーの処置:** 正しいアーカイブ・ログ・ファイルを判別するには、QUERY STATUS オプションを付けて、ROLLFORWARD DATABASE コマンドを実行してください。正しいアーカイブ・ログ・ファイルをデータベース・ログ・ディレクトリーに移すか、またはデータベースが整合性のある状況の場合は、正しいアーカイブ・ファイルを指すようにログ・パスを変更して、ROLLFORWARD DATABASE コマンドを再発行してください。別の方法として、正しいアーカイブ・ファイルを指しているオーバーフロー・ログ・パスを使用して、コマンドを再発行してください。

---

**SQL1264N** アーカイブ・ファイル *name* は、ノード *node-number* 上のデータベース *database-name* に属していません。

**説明:** 示されたアーカイブ・ログ・ファイルがログ・ディレクトリー、またはオーバーフロー・ログ・ディレクトリーで見つかりましたが、指定されたデータベースには属していません。

ロールフォワード・リカバリー処理は停止します。

(注：パーティション・データベース・サーバーを使用している場合、ノード番号は、エラーを起こしているノードを示しています。そうでない場合、これは関係のないものなので無視してください。)

**ユーザーの処置:** 正しいアーカイブ・ログ・ファイルを判別するには、QUERY STATUS オプションを付けて、ROLLFORWARD DATABASE コマンドを実行してください。正しいアーカイブ・ログ・ファイルをデータベース・ログ・ディレクトリーに移すか、またはデータベースが整合性のある状況の場合は、正しいアーカイブ・ファイルを

指すようにログ・パスを変更して、  
ROLLFORWARD DATABASE コマンドを再発行してください。別の方法として、正しいアーカイブ・ファイルを指しているオーバーフロー・ログ・パスを使用して、コマンドを再発行してください。

---

**SQL1265N** アーカイブ・ログ・ファイル *name* が、ノード *node-number* のデータベース *name* に対する現行のログ・シーケンスに関連していません。

**説明:** ロールフォワード・リカバリーの場合は、ログ・ファイルが正しいシーケンスで処理される必要があります。ログ・ファイルのシーケンスは、リストアされたデータベース、または処理されたログ・ファイルによって決定されます。これに加えて、表スペース・レベル・ロールフォワード・リカバリーの場合は、ログ・ファイルが、データベースの現在の状態が達したシーケンスで処理される必要があります。指定のアーカイブ・ログ・ファイルが、指定ノードのデータベースのログ・ディレクトリまたはオーバーフロー・ログ・パスで見つかりましたが、ログ・ファイルが正しいログ・シーケンスではありませんでした。

ロールフォワード・リカバリー処理は停止します。

(注：パーティション・データベース・サーバーを使用している場合、ノード番号は、エラーを起こしているノードを示しています。そうでない場合、これは関係のないものなので無視してください。)

**ユーザーの処置:** 正しいアーカイブ・ログ・ファイルを判別するには、QUERY STATUS オプションを付けて、ROLLFORWARD DATABASE コマンドを実行してください。正しいアーカイブ・ログ・ファイルをデータベース・ログ・ディレクトリに移すか、またはデータベースが整合性のある状況の場合は、正しいアーカイブ・ファイルを指すようにログ・パスを変更して、

ROLLFORWARD DATABASE コマンドを再発行してください。別の方法として、正しいアーカイブ・ファイルを指しているオーバーフロー・ログ・パスを使用して、コマンドを再発行してください。

---

**SQL1266N** データベース *name* が、指定されたポイント・イン・タイムを過ぎた *timestamp* までロールフォワードされました。

**説明:** ロールフォワードが、指定されたデータベース・ログ・ファイルの現在時刻より前のタイム・スタンプで停止するよう要求されています。これは、データベースあるいは表スペースのサブセットを、時刻までにロールフォワードするときにあります。

データベースのロールフォワード処理が停止します。

**ユーザーの処置:** 正しい時刻を指定するか、またはデータベースまたは表スペース・サブセットをバックアップ・バージョンからリストアして、ROLLFORWARD コマンドを再発行してください。

---

**SQL1267N** システムが、現在の PATH 環境変数で *db2uexit* を見つけることができませんでした。

**説明:** ユーザー提供ファイル *db2uexit* が、現在の PATH 環境変数に存在しないか、またはファイルが存在しないために、見つかりませんでした。

コマンドは処理されません。

**ユーザーの処置:** *db2uexit* へのパスを含むように、現在の PATH 環境変数を更新するか、または *db2uexit* ファイルを作成して、必要に応じて、PATH 環境変数を更新してください。

---

**SQL1268N** ノード *node-number* のデータベース *name* のログ・ファイル *logfile* を検索中、エラー *error* により、ロールフォワード・リカバリーが停止されました。

**説明:** ロールフォワードの処理は、ログ・ファイルを検索するために *db2uext2* を呼び出します。エラーは *db2uext2* で起きた可能性があります。

ロールフォワード処理は停止します。データベースまたは表スペースは、まだロールフォワード・ペンディングの状態です。

(注：パーティション・データベース・サーバーを使用している場合、ノード番号は、エラーを起こしているノードを示しています。そうでない場合、これは関係のないものなので無視してください。)

**ユーザーの処置:** エラー、およびロールフォワード・リカバリーの再開または終了の記述については、*管理ガイド* のユーザー出口文書を参照してください。

---

**SQL1269N** 表スペース・レベル・ロールフォワード・リカバリーがすでに実行中です。

**説明:** 表スペース・レベル・ロールフォワード・リカバリーの使用が試みられましたが、すでに実行されています。一時点では、1つのエージェントのみが、ロールフォワード・リカバリーを実行することができます。

**ユーザーの処置:** 表スペース・レベル・ロールフォワード・リカバリーが完了するまで待ってください。リカバリーが必要な表スペースがまだある場合は、もう一度表スペース・レベル・ロールフォワード・リカバリーを開始してください。

---

**SQL1270C** LANG 環境変数が *string* にセットされています。この言語はサポートされていません。

**説明:** LANG 環境変数が、データベース・マネージャーがサポートしていない言語に設定されています。処理を継続できません。

**ユーザーの処置:** LANG 環境変数を、サポートされている言語に設定してください。詳細については、*管理ガイド* の「各国語サポート」付録を参照してください。

連合システム・ユーザー: NLS 情報については *連合システム・ガイド* を参照してください。

---

**SQL1271W** データベース *name* は回復されましたが、1つ以上の表スペースがノード *node-list* でオフラインです。

**説明:** このメッセージは、クラッシュ・リカバリー、データベース・レベル・ロールフォワード・リカバリー、または表スペース・レベル・ロールフォワード・リカバリーの後に、出される場合があります。データベースのレベル・ロールフォワード・リカバリーについて、データベースは STOP オプションが指定されている場合使用可能です。指定ノードにある1つ以上の表スペースは使用できません。これは以下の場合に起きません。

- STOP オプションが時刻までの表スペース・レベル・ロールフォワード・リカバリーに指定されていない
- 1つ以上の表スペースがリカバリー中にエラーを受け取った
- 表スペース・レベル・ロールフォワード・リカバリーがすでに進行中である間に追加の表スペースがリストアされた
- データベースが以前に表スペースがロールフォワードされたポイント・イン・タイムを過ぎてロールフォワードされると、関連するすべての表スペースがリストア・ペンディング状態になる



オフラインの表スペースは LIST TABLESPACES コマンドあるいは db2dart ユーティリティで指定されるノードで識別されます。管理通知ログが、特定の表スペースに関する詳細情報を提供します。

"、..." がノード・リストの終わりに表示されている場合、完全なリストを見るには管理通知ログを調べてください。

(注: パーティション・データベース・サーバーを使用している場合、ノード番号は、エラーを起こしているノードを示しています。そうでない場合、これは関係のないものなので無視してください。)

**ユーザーの処置:** 必要に応じて表スペースの修復あるいはリストアを行いロールフォワード・リカバリーを実行します。同じエラーが起きる場合、表スペース・レベル・ロールフォワード・リカバリーをオフラインで実行してみてください。

---

### SQL1272N データベース *name* に対する表スペース・レベル・ロールフォワード・リカバリーが、ノード *node-list* で完了する前に停止しました。

**説明:** すべての修飾表スペースがロールフォワードされる前に、表スペース・レベル・ロールフォワード・リカバリーが指定ノードで停止しました。これは、以下のいずれかによって起きる可能性があります。

- トランザクション表がいっぱいである。
- ロールフォワードされた表スペースのすべてが入出力エラーを受け取った。
- ポイント・イン・タイム表スペース・レベルのロールフォワードでロールフォワードされた表スペースのいずれかに入出力エラーが起きた。
- ポイント・イン・タイム表スペース・レベルのロールフォワードでロールフォワードされた表スペースのいずれかに対して、変更を行ったアクティブなトランザクションを検出した。こ

のトランザクションは未確定トランザクションである場合があります。

- 表スペース・レベル・ロールフォワードが中断し、再開する前にロールフォワードしていたすべての表スペースが再度リストアされた場合にも起きる可能性があります。

"、..." がノード・リストの終わりに表示されている場合、完全なリストを見るには管理通知ログを調べてください。

(注: パーティション・データベース・サーバーを使用している場合、ノード番号は、エラーを起こしているノードを示しています。そうでない場合、これは関係のないものなので無視してください。)

**ユーザーの処置:** 原因については、管理通知ログをチェックしてください。原因にしたがって以下のいずれかを行ってください。

- LIST TABLESPACES コマンドを使用して表スペースが入出力エラーを受け取ったかを判別してください。その場合、表スペースを修復します。
- トランザクション表がいっぱいの場合、MAXAPPLS データベース構成パラメーターを組み込むか、あるいは表スペース・レベル・ロールフォワードをオフラインで実行してみてください。
- 原因がアクティブあるいは未確定のトランザクションにある場合、トランザクションを完了してください。
- 前の表スペース・レベル・ロールフォワードを中断した後で、表スペースがリストアされた場合、前の表スペース・レベル・ロールフォワードはこの時点で取り消されます。以下の表スペース・レベル・ロールフォワード・コマンドが、ロールフォワード・ペンディング状態の表スペースを調べます。

表スペース・レベル・ロールフォワード・リカバリーを再度実行してください。

---

**SQL1273N** ノード *node-number* にログ・ファイル *name* がないため、データベース *name* のロールフォワード・リカバリーは、指定された停止点 (ログの終わりまたはポイント・イン・タイム) に到達できません。

**説明:** ロールフォワード・データベース・ユーティリティがデータベース・ログ・ディレクトリー、または指定ノードのオーバーフロー・ログ・ディレクトリーで指定のアーカイブ・ログ・ファイルを見つけることができません。

ロールフォワード・リカバリーは停止しました。

(注：パーティション・データベース・サーバーを使用している場合、ノード番号は、エラーを起こしているノードを示しています。そうでない場合、これは関係のないものなので無視してください。)

**ユーザーの処置:** 以下のいずれかを実行します。

- 示されたアーカイブ・ログ・ファイルをデータベース・ログ・ディレクトリーに移すか、またはデータベースが整合状態の場合は、ログ・パスを正しいアーカイブ・ファイルに変更して、**ROLLFORWARD DATABASE** コマンドを再発行してください。別の方法として、正しいアーカイブ・ファイルを指しているオーバーフロー・ログ・パスを使用して、コマンドを再発行してください。
- 抜けているログ・ファイルが見つからない場合には、すべてのノードで、データベース / 表スペースをリストアップし抜けているログ・ファイルのタイム・スタンプより早いタイム・スタンプでポイント・イン・タイムの指定をしてください。

---

**SQL1274N** データベース *name* にロールフォワード・リカバリーが必要であり、ポイント・イン・タイムはログの最後に設定する必要があります。

**説明:** データベースがロールフォワードされる必要があります。データベース・レベル・ロールフォワード・リカバリーでは、ログの最後までデータベース・レベル・ロールフォワードがすでに進行中であるため、ポイント・イン・タイムをログの最後に設定する必要があります。ロールフォワードを継続するには、同じ停止時間を指定する必要があります。

以下の理由で、表スペース・レベル・ロールフォワード・リカバリーの場合は、ポイント・イン・タイムをログの最後に設定する必要があります。

- システム・カタログにはロールフォワード・リカバリーが必要です。システム・カタログはいつも、ほかのすべての表スペースと整合性を保つために、ログの最後にロールフォワードする必要があります。
- ログの最後に、表スペース・レベル・ロールフォワードがすでに進行中です。ロールフォワードを継続するには、同じ停止時間を指定する必要があります。
- 表スペース・レベル・ロールフォワードに指定されたポイント・イン・タイムが、データベースの終わりを過ぎています。時刻が間違っている可能性があります。これが指定したいリカバリー停止時刻の場合は、**END OF LOGS** オプションを使用して、コマンドを再サブミットしてください。

データベースはロールフォワードされません。

**ユーザーの処置:** **ROLLFORWARD TO END OF LOGS** を指定して、**ROLLFORWARD** コマンドを再発行してください。



---

**SQL1275N** ノード *node-list* のデータベース *name* には指定された時刻より後の情報が含まれるため、ロールフォワードに渡される停止時刻は、*timestamp* と同じかそれより後にする必要があります。

**説明:** ロールフォワードされたデータベースあるいは少なくとも表スペースのいずれか 1 つがオンラインでバックアップされました。詳細仮想タイム・スタンプが表スペース・バックアップにあります。ロールフォワードに渡される停止時刻は、指定ノードのオンライン・バックアップの終了時刻よりも大きいかなければなりません。

"、..." がノード・リストの終わりに表示されている場合、完全なリストを見るには管理通知ログを調べてください。

(注: パーティション・データベース・サーバーを使用している場合、ノード番号は、エラーを起こしているノードを示しています。そうでない場合、これは関係のないものなので無視してください。)

**ユーザーの処置:** 以下のいずれかを実行します。

- 停止時刻を *timestamp* よりも大きいかなしくして、コマンドを再発行してください。時刻は CUT (Coordinated Universal Time) で指定する必要があります。
- ノードの前のバックアップをリストアして、ROLLFORWARD DATABASE コマンドを再発行します。

---

**SQL1276N** データベース *name* は、ロールフォワードが *timestamp* と同等かそれよりも大きなポイント・イン・タイムを渡すまでは、ロールフォワード・ペンディング状態を抜け出すことができません。これは、ノード *node-number* に、指定された時刻よりも後の情報が含まれるためです。

**説明:** 呼び出し元のアクション

SQLUM\_ROLLFWD\_STOP、SQLUM\_STOP、SQLUM\_ROLLFWD\_COMPLETE、

SQLUM\_COMPLETE を指定して、データベースあるいは表スペースのサブセットについて、ロールフォワード・ペンディング状態を解除させる要求が行なわれました。ただし、ロールフォワードされたデータベースあるいは少なくとも表スペースのいずれか 1 つがオンラインでバックアップされました。指定ノードのオンライン・バックアップ・タイム・スタンプの終わりにデータベースがロールフォワードされるまで、この要求を付与することはできません。

このエラーは、要求されたりカバリーを実行するために、すべてのログ・ファイルが提供されているわけではない場合にも発生します。

(注: パーティション・データベース・サーバーを使用している場合、ノード番号は、エラーを起こしているノードを示しています。そうでない場合、これは関係のないものなので無視してください。)

**ユーザーの処置:** ROLLFORWARD コマンドに指定された停止時刻が *timestamp* より小さい場合は、*timestamp* と同等かそれより大きな停止時間で、コマンドを再実行依頼してください。

ログ・ファイルがすべて提供されているか、確認してください。ROLLFORWARD QUERY STATUS コマンドは、どのログ・ファイルが次に処理されるかを示します。ログ・ファイルの欠落に対する理由には、以下のものがあります。

- ログ・パスが変更となった。ファイルは、前のログ・パスにあります。

- ユーザー出口プログラムは、ログ・ファイルが別のロケーションに保存されているため、検索することができない。

欠落ログ・ファイルが見つかった場合、これをログ・パスにコピーし、コマンドを再実行します。

---

**SQL1277N** リストアで、1 つ以上の表スペース・コンテナがアクセス不能であることが検出されたか、あるいはコンテナの状態を「ストレージを定義してください」に設定しました。

**説明:** リストアは、リストア中の各表スペースが必要とするコンテナが、現在システムでアクセス可能であるかどうかをチェックします。アクセス可能な場合は、コンテナが存在しないと、リストアがコンテナを作成します。コンテナが作成できない場合、現在別の表スペースが使用中の場合、または他の理由でアクセスできない場合は、リストアを続ける前に、必要なコンテナのリストを訂正する必要があります。

これがリダイレクト・リストアの場合、リストアされている表スペースの各コンテナの状態は、「ストレージを定義してください」に設定されません。それによって、ストレージを再定義するのに、コンテナに対して **SET TABLESPACE CONTAINERS** api またはコマンドを使用することができます。

**ユーザーの処置:** リストア中の各表スペースのコンテナのリストを判別するには、**TABLESPACE CONTAINER QUERY** api を使用してください。各表スペースの更新したリストを指定するには、**SET TABLESPACE CONTAINERS** api を使用してください。この api により、このリストがコンテナの初期リスト (すなわち、後続のロールフォワードが、データベース・ログに記録されている「コンテナの追加」処理を再実行します) か、または最終リスト (ロールフォワードが「コンテナの追加」処理を再実行しません) かを指定することができます。

コンテナが読み取り専用の可能性もあります。

この場合、リストアを続行するために必要なアクションは、コンテナへの読み取り / 書き込みアクセスの付与だけです。

---

**SQL1278W** ロールフォワード処理が正常に完了しました。アクティブあるいは未確定のトランザクションでは、ノード *node-list* でのロールバックが必要です。

**説明:** 時刻に対する表スペースのサブセットのロールフォワードは正常に完了しましたが、以下の状態のいずれか、あるいは両方が起きました。

1. 指定時刻でアクティブなトランザクションが 1 つまたは複数存在します。トランザクションごとに、表スペース・サブセットの表スペースでロールバックされます。
2. 指定時刻で未確定のトランザクションが 1 つまたは複数存在します。未確定のトランザクションごとに、表スペース・サブセットの表スペースでロールバックされます。

表スペース・サブセットの表スペースでロールバックされたトランザクションは、ロールフォワードを行っていない別の表スペースにコミットされたままの可能性もあります。

"、..." がノード・リストの終わりに表示されている場合、完全なリストを見るには管理通知ログを調べてください。

(注: パーティション・データベース・サーバーを使用している場合、ノード番号は、エラーを起こしているノードを示しています。そうでない場合、これは関係のないものなので無視してください。)

**ユーザーの処置:** 管理通知ログには、ロールフォワード・リカバリーでロールバックされたロールバックの詳細が含まれます。

---

**SQL1279W** いくつかの無効な索引が再作成されていない可能性があります。

**説明:** データベースの索引が再作成されるのは、データベースの再始動中です。いくつかの無効な索引の再作成を妨げるエラーが、再始動中に起きました。管理通知ログで詳細を参照できます。

データベースの再始動または Reorg は成功しました。

**ユーザーの処置:** 管理通知ログを調べて索引が再作成できなかった理由を判別し、問題を訂正してください。表が最初にアクセスされたときに、表の無効な索引が再作成されます。

---

**SQL1280N** ロールフォワードに渡される停止時間は、データベース *name* の *timestamp* と同等かそれより小さくしてください。これは、少なくとも 1 つの表スペースが、すでにこのポイント・イン・タイムまでロールフォワードされたためです。

**説明:** 時刻に対するロールフォワードで指定された表スペースの少なくとも 1 つが、以前にすでにロールフォワードされています。これ以上ロールフォワードを行うことはできません。

**ユーザーの処置:** 以下のいずれかを実行します。

- 停止時刻 *timestamp* を指定してコマンドを再実行してください。
- すべての表スペースを再度リストアし、*timestamp* より前の停止時刻を指定してコマンドを再実行してください。
- 表スペースの、前の時刻のロールフォワードで行ったバックアップをリストアし、この時刻と同一の停止時刻でコマンドを再実行してください。

時刻は CUT (Coordinated Universal Time) で指定する必要があります。

---

**SQL1281N** パイプ *pipe-name* に障害が起きたために、データベース *database-alias* への接続が困難です。

**説明:** DB2 サーバーがパイプを壊したために、接続が失われました。現在のトランザクションはロールフォワードされました。

**ユーザーの処置:** 現在のコマンドを再発行してください。エラーが続く場合は、技術サービス担当者に連絡してください。

トレースがアクティブの場合は、オペレーティング・システムのコマンド・プロンプトから、独立トレース機能呼び出ししてください。以下の情報を用意して、技術サービス担当者に連絡してください。

必要な情報:

- 問題記述
- SQLCODE またはメッセージ番号
- SQLCA の内容 (可能であれば)
- トレース・ファイル (可能であれば)

**sqlcode:** -1281

**sqlstate:** 40504

---

**SQL1282N** *pipe-name* 上のパイプ・インスタンスがすべて使用中であるために、データベース *database-alias* への接続が失敗しました。

**説明:** 接続が DB2 によって拒否されたので、Named PIPE への接続が失敗しました。Named PIPE で許される接続の数には制限があります。

**ユーザーの処置:** DB2 サーバーの接続制限を増やすか、または Named PIPE を使用しているいくつかのアプリケーションを終了させて、接続リソースを解放してください。

**sqlcode:** -1282

**sqlstate:** 08001

---

**SQL1283N** パイプ *pipe-name* が別の処理で使用  
中なので、データベース  
*database-alias* への接続が失敗しま  
した。

**説明:** Named PIPE の名前が、すでに別の処理に  
よって使用されています。Named PIPE サポート  
は開始しませんでした。

**ユーザーの処置:** 環境変数 DB2PIPENAME を設  
定して違う名前を選択するか、または Named  
PIPE を使用する別のプログラムに異なるパイプ  
名を使用させます。

---

**SQL1284N** パイプ *pipe-name* が見つからない  
ために、データベース  
*database-alias* への接続が失敗しま  
した。

**説明:** サーバーが Named PIPE サポートを開始  
していなかったか、またはサーバーが Named  
PIPE に対して別の名前を使用しています。

**ユーザーの処置:** データベース・マネージャーを  
始動して、Named PIPE サポートを開始してくだ  
さい。Named PIPE サポートが開始されている場  
合は、環境変数 DB2PIPENAME を同じ値に設定  
して、Named PIPE の名前をクライアントとサー  
バーの間で同じにしてください。

**sqlcode:** -1284

**sqlstate:** 08001

---

**SQL1285N** パイプ *pipe-name* が無効なため  
に、データベース *database-alias* へ  
の接続が失敗しました。

**説明:** 環境変数 DB2PIPENAME によって設定さ  
れた代替パイプ名が無効です。

**ユーザーの処置:** 環境変数 DB2PIPENAME の値  
は、有効なパイプ名でなければなりません。パ  
イプ名は 8 バイトを超えてはならず、通常のフ  
ァイル名と同じ構文制約にしたがう必要がありま  
す。

**sqlcode:** -1285

**sqlstate:** 08001

---

**SQL1286N** オペレーティング・システムが、パ  
イプ *pipe-name* のリソースを使い  
果たしたために、データベース  
*database-alias* への接続が困難で  
す。

**説明:** オペレーティング・システムがリソース  
(スワッピング・スペース、ディスク・スペース、  
ファイル) を使い果たしたために、Named PIPE  
が失敗しました。現在のトランザクションはロー  
ルフォワードされました。

**ユーザーの処置:** システム・リソースを解放し  
て、もう一度やり直してください。

**sqlcode:** -1286

**sqlstate:** 40504

---

**SQL1287N** 名前付きの *pipe* を検出できないた  
めに、インスタンス *instance* のア  
タッチが正常に実行されていま  
せん。

**説明:** サーバーがその Named PIPE のサポート  
を開始していないか、あるいはインスタンス名が  
正しくありません。

**ユーザーの処置:** データベース・マネージャーが  
そのサーバーで始動されていて、名前付きのパ  
イプのサポートが開始されていることを確認して  
ください。インスタンス名が正しいことを確認し  
てください。

---

**SQL1288N** リモート・サーバーは、この操作を  
サポートしていません。

**説明:** リモート・サーバーは 64 ビットのプラッ  
トフォームで実行されており、バージョン 7 (ま  
たはそれ以前) の 32 ビットのクライアントから  
の非 SQL 要求をサポートしていません。

**ユーザーの処置:** サポートされているクライアン

トからこの操作を実行してください。

---

**SQL1290N DFT\_CLIENT\_COMM** データベース・マネージャー構成パラメーター、または **DB2CLIENTCOMM** 環境変数の値は無効です。

**説明:** 正しくない値が指定されたか、または指定されたプロトコルが、ターゲット・データベースによってサポートされていません。許容される値は:

- UNIX プラットフォーム: TCPIP および APPC
- OS/2: TCPIP, APPC, IPXSPX, および NETBIOS
- Windows: TCPIP, APPC (Windows 32 ビット専用)、NETBIOS, および NPIPE

複数の値を指定する場合は、それらをコンマで区切る必要があります。

このメッセージは、接続で呼び出された中間ノードから返される場合があることに注意してください。たとえば、DB2 Connect ゲートウェイ経由で DRDA サーバーに接続しようとしており、クライアント・ワークステーションがグローバル・ディレクトリー・サービスを使用していない場合は、このメッセージが DB2 Connect ゲートウェイから返される可能性があります。

**ユーザーの処置:** 値を訂正して、もう一度やり直してください。

**sqlcode:** -1290

**sqlstate:** 08001

---

**SQL1291N** ディレクトリー・サービス・エラーが見つかりました。サービス:  
*directory-services-type*、**API:** *API*、**関数:** *function*、**エラー・コード:**  
*rc*。

**説明:** ディレクトリー・サービス・サブシステムによって、エラーが見つけられました。詳細については、トークンの値を参照してください。

以下は、トークンの値の説明です。

*directory-services-type*

使用されたディレクトリー・サービスのタイプ。有効なトークンは、以下のとおりです。

- DCE

*API*

上記のディレクトリー・サービスのアクセスに使用された、アプリケーション・プログラミング・インターフェース。有効なトークンは、以下のとおりです。

- XDS/XOM

*function*

エラー・コードを返したディレクトリー・サービス・サブシステム関数の名前。

*rc*

上記関数から戻されたエラー・コード。値の意味は、使用している API によって異なります。

*ds\_read* などの XDS 関数の場合、戻りコードの値は、DCE 組み込みファイル *xds.h* で見つかります。

*om\_get* などの XOM 関数の場合、戻りコードの値は、DCE 組み込みファイル *xom.h* で見つかります。

このメッセージは、接続で呼び出された中間ノードから返される場合があることに注意してください。たとえば、DB2 Connect ゲートウェイ経由で DRDA サーバーに接続しようとしており、クライアント・ワークステーションがグローバル・ディレクトリー・サービスを使用していない場合は、このメッセージが DB2 Connect ゲートウェイから返される可能性があります。

**ユーザーの処置:** 以下を確認してください。

- ディレクトリー・サービスを提供する製品が、正しくインストールされ、使用可能になっていること。
- ディレクトリー・サービス提供者 (たとえば、DCE) がログインを要求している場合は、ディ

レクトリー項目をアクセスするための適切な許可を持って、ディレクトリー・サービスにログインしていること。

問題が続く場合は、システム管理者またはデータベース管理者、もしくは両者に連絡し、提供されたトークンのセットを使用して、問題の原因を判別してください。

**sqlcode:** -1291

**sqlstate:** 08001

---

**SQL1292N** データベースまたはデータベース・マネージャー・インスタンスのグローバル名が無効です。

**説明:** データベースまたはデータベース・マネージャー・インスタンスのグローバル名には、NULL を使用することはできず、255 文字を超えることもできません。グローバル名前は `"/..."` または `"/:"` で始まる必要があります。

このメッセージは、接続で呼び出された中間ノードから返される場合があることに注意してください。たとえば、DB2 Connect ゲートウェイ経由で DRDA サーバーに接続しようとしており、クライアント・ワークステーションがグローバル・ディレクトリー・サービスを使用していない場合は、このメッセージが DB2 Connect ゲートウェイから返される可能性があります。

**ユーザーの処置:** グローバル名を訂正して、もう一度やり直してください。

---

**SQL1293N** グローバル・ディレクトリー項目で、エラーが見つかりました。エラー・コード : *error-code*

**説明:** 使用しているグローバル・データベース・ディレクトリー項目のいずれかで、エラーが見つかりました。グローバル名を訂正して、もう一度やり直してください。

**1** データベース・オブジェクトに、認証情報が入っていません。

**2** データベース・オブジェクトとデータベース・ロケーター・オブジェクトの両方に、通信プロトコル情報が入っていません。

**10** 項目がデータベース・オブジェクトではありません。

**11** データベース・オブジェクトの固有データベースが見つからないか、または長すぎるかのどちらかです。

**12** データベース・オブジェクトのデータベース・プロトコルが、見つからないか、または長すぎるかのどちらかです。

**13** データベース・オブジェクトで、無効な認証情報が見つかりました。

**14** データベース・オブジェクトで、十分でない、または無効な通信プロトコル情報が見つかりました。

**15** データベース・ロケーター・オブジェクト名が、データベース・オブジェクトではありません。

**16** データベース・オブジェクトのデータベース・ロケーター・オブジェクト名が無効です。

**20** 項目がデータベース・ロケーター・オブジェクトではありません。

**22** データベース・ロケーター・オブジェクトで、十分でない、または無効な通信プロトコル情報が見つかりました。

**30** 項目がルーティング情報オブジェクトではありません。

**31** ターゲット・データベース情報が、経路指定情報オブジェクトで見つかりませんでした。



- 32 ルーティング情報オブジェクトのターゲット・データベースの情報が不十分です。
- 33 適切なゲートウェイが、経路指定情報オブジェクトで見つかりませんでした。
- 34 ゲートウェイでの認証のためのフラグが無効です。
- 35 ゲートウェイのデータベース・ロケータ・オブジェクト名が無効です。
- 36 経路指定情報オブジェクトのターゲット・データベース情報属性のデータベース名が、見つからないか、または長すぎるかのどちらかです。
- 37 経路指定情報オブジェクトのターゲット・データベース情報属性のデータベース・プロトコルが、見つからないか、または長すぎるかのどちらかです。

DCE サブシステムが操作不能の場合、または DCE ディレクトリー項目を読むための十分な特権を持っていない場合も、このメッセージが表示される場合があることに注意してください。

このメッセージは、接続で呼び出された中間ノードから返される場合があることに注意してください。たとえば、DB2 Connect ゲートウェイ経由で DRDA サーバーに接続しようとしており、クライアント・ワークステーションがグローバル・ディレクトリー・サービスを使用していない場合は、このメッセージが DB2 Connect ゲートウェイから返される可能性があります。

**ユーザーの処置:** DCE サブシステムが操作可能で、ディレクトリー項目を読むための適切な特権を持っていることを確認してください。問題が続く場合は、データベース管理者に連絡して、ディレクトリー項目のエラーを訂正してください。これらのディレクトリー・オブジェクトの形式については、*管理ガイド* を参照してください。

**sqlcode:** -1293

**sqlstate:** 08001

---

**SQL1294N** グローバル・ディレクトリー・アクセスに使用されているディレクトリー・パス名が、指定されていないかまたは無効です。

**説明:** グローバル・ディレクトリー・サービスを使用するには、ディレクトリー・パス名を、*dir\_path\_name* データベース・マネージャー構成パラメーター、または DB2DIRPATHNAME 環境変数のどちらかに指定する必要があります。名前が指定されていないか、または指定した名前が有効ではありません。

このメッセージは、接続で呼び出された中間ノードから返される場合があることに注意してください。たとえば、DB2 Connect ゲートウェイ経由で DRDA サーバーに接続しようとしており、クライアント・ワークステーションがグローバル・ディレクトリー・サービスを使用していない場合は、このメッセージが DB2 Connect ゲートウェイから返される可能性があります。

**ユーザーの処置:** 正しい名前をデータベース管理者に尋ね、正しい名前を指定して、もう一度やり直してください。

**sqlcode:** -1294

**sqlstate:** 08001

---

**SQL1295N** グローバル・ディレクトリー・アクセスに使用されている経路指定情報オブジェクトが、指定されていないかまたは無効です。

**説明:** このクライアントに対してネイティブではないデータベース・プロトコルを使用して、リモート・データベースにアクセスするグローバル・ディレクトリー・サービスを使用するには、ルーティング情報オブジェクトの名前を、*route\_obj\_name* データベース・マネージャー構成パラメーター、または DB2ROUTE 環境変数に指定する必要があります。名前が指定されていないか、または指定した名前が有効ではありません。

このメッセージは、接続で呼び出された中間ノードから返される場合があることに注意してください。たとえば、DB2 Connect ゲートウェイ経由で DRDA サーバーに接続しようとしており、クライアント・ワークステーションがグローバル・ディレクトリー・サービスを使用していない場合は、このメッセージが DB2 Connect ゲートウェイから返される可能性があります。

**ユーザーの処置:** 正しいオブジェクト名をデータベース管理者に尋ね、正しい名前を指定して、もう一度やり直してください。

**sqlcode:** -1295

**sqlstate:** 08001

---

**SQL1296N** DIR\_TYPE パラメーターが NONE でない場合は、データベース・マネージャー構成パラメーターである DIR\_PATH\_NAME および DIR\_OBJ\_NAME に、有効な値を指定する必要があります。

**説明:** これらのパラメーターには相互関係があります。DIR\_TYPE の値が NONE の場合は、他の 2 つの値は無視されます。DIR\_TYPE の値が NONE でない場合は、他の 2 つとも、有効な値を持っている必要があります。DIR\_TYPE が

## SQL1700 - SQL1799

---

**SQL1700N** データベースの移行中に、予約済みスキーマ名 *name* がデータベースで見つかりました。

**説明:** データベースに、現在のデータベース・マネージャーによって予約されているスキーマ名 *name* を使用する、1 つ以上のデータベース・オブジェクトが入っています。

コマンドは処理されません。

**ユーザーの処置:** 予約済みスキーマ名を使用するすべてのデータベース・オブジェクトがドロップされていることを確認し、別のスキーマ名を使用して、オブジェクトを再作成してください。こ

NONE でない場合は、以下の規則が適用されます。

1. DIR\_PATH\_NAME と DIR\_OBJ\_NAME の値は、NULL (またはブランク) にすることはできません。
2. DIR\_TYPE の値が DCE の場合は、DIR\_PATH\_NAME の値を、特殊な DCE ストリングである "/../" または "/./" で始める必要があります。

**ユーザーの処置:** DIR\_TYPE の値を変更する場合は、最初に、DIR\_PATH\_NAME と DIR\_OBJ\_NAME パラメーターに有効な値が指定されていることを確認してください。DIR\_PATH\_NAME または DIR\_OBJ\_NAME パラメーターをブランクにする場合は、最初に、DIR\_TYPE を NONE に設定してください。

---

**SQL1297N** このコマンドは、このプラットフォームでは現在サポートされていません。

**説明:** このコマンドで要求された関数は、このプラットフォームではサポートされていません。

**ユーザーの処置:** このコマンドを使用しないでください。

の修正は、データベースが最初に作成されたりリリースのデータベース・マネージャーで行う必要があります。もう一度移行を行う前に、予約済みスキーマ名が使用されていないことを確認してください。その後で、データベース・マネージャーの現在のリリースで、データベースの移行コマンドを再発行してください。

---

**SQL1701N** 最後のセッションが異常終了したために、データベースは移行できません。

**説明:** 最後のデータベース操作が、前のデータベース・マネージャーの下で、異常終了 (たとえ



ば、電源障害) しました。データベースが再始動されるまでは、データベースの移行を行なうことはできません。

コマンドは処理されません。

**ユーザーの処置:** データベースを再始動する必要があります。データベースが最後にアクセスしたリリースのデータベース・マネージャーを使用する `RESTART DATABASE` コマンドを実行する必要があります。その後で、データベース・マネージャーの現在のリリースで、データベースの移行コマンドを再発行してください。

---

**SQL1702W** *protocol* 接続マネージャーが、正常に始動しました。

**ユーザーの処置:** アクションは必要ありません。

---

**SQL1703W** データベースの移行中は、`db2event` ディレクトリーを作成することはできません。

**説明:** データベースは正常に移行されましたが、`db2event` ディレクトリーは作成できませんでした。

これは警告です。

**ユーザーの処置:** イベント・モニターを使用する場合は、`db2event` ディレクトリーを作成する必要があります。`db2event` ディレクトリーは、移行済みデータベースが常駐するデータベース・ディレクトリーに作成する必要があります。移行済みデータベースのデータベース・ディレクトリーは、`LIST DATABASE DIRECTORY` を呼び出すことにより判別できます。

---

**SQL1704N** データベースの移行に失敗しました。理由コード *reason-code*。

**説明:** データベースの移行に失敗しました。理由コードは以下の通りです。

- 1 無効なスキーマ名が見つかりました。
- 2 データベースが移行可能ではありません。

ん。データベースが、以下のいずれかの状態であった可能性があります。

- バックアップ・ペンディング状態
- ロールフォワード保留状態
- トランザクション不整合状態

- 3 データベース・ログがいっぱいです。
- 4 ディスク・スペースが足りません。
- 5 データベース構成ファイルを更新できません。
- 6 データベースを再配置できません。
- 7 データベース・サブディレクトリーまたはデータベース・ファイルのいずれかにアクセスできません。
- 8 データベース・コンテナ・タグを更新できません。
- 9 表スペースのアクセスが許されていません。
- 10 無効なタイプ名が見つかりました。
- 11 プロシージャの特定名が関数の特定名と競合しています。
- 12 Datajoinder 抽象データ・タイプが見つかりました。
- 13 Datajoinder 拡張索引が見つかりました。
- 17 DMS システム・カタログ表スペースから新規ページを割り振ることができません。
- 18 ユニコード・データベースの移行で、重複する関数を検出しました。

ユニコード・データベースの場合、以下は同等と見なされます。

CHAR と GRAPHIC  
VARCHAR と VARGRAPHIC  
LONG VARCHAR と LONG VARGRAPHIC  
CLOB と DBCLOB

同等のストリング・タイプだけが異なる関数シングニチャーは“重複”と考えら

れ、このデータベースに共存することはできません。たとえば、`func1(CHAR, GRAPHIC, VARCHAR)`と`func1(GRAPHIC, GRAPHIC, VAR GRAPHIC)`は重複です。

19 重複する特定名が検出されました。

`SYSIBM.SYSFUNCTIONS`と`SYSIBM.SYSPROCEDURES`は、データベースの移行で`SYSIBM.SYSROUTINES`カタログにマージされました。ただし、特定名は`SYSIBM.SYSROUTINES`の中でユニークである必要があります。

20 ADT データ・タイプを検出しました。

データベースで、1 つ以上の要約データ・タイプが見つかりました。移行は失敗しました。すべての要約データ・タイプは、現在のリリースにデータベースを移行する前にドロップする必要があります。

21 データベースの移行は、カタログ・ノードとほかのノードでは完了しましたが、すべてのノードで完了したわけではありません。一部のノードで、ノード障害が報告されています。その結果、これらのノードでは移行が行われていません。

22 カatalog・ノードがノード障害を報告しているため、データベースの移行は失敗しました。

23 データベースの移行に失敗しました。連合データベースを 64 ビットのインスタンスに移行する機能はサポートされていません。

| 24 イベント・モニター・ディレクトリーの作成中にエラーが発生しました (データベース・ディレクトリー内のディレクトリー `db2event` の下の `db2detaildeadlock`)。|

**ユーザーの処置:** 理由コードに基づいた解決策は、以下のとおりです。

1 予約済みのスキーマ名は `SYSIBM`、`SYSCAT`、`SYSSTAT`、および `SYSFUN` です。これらのスキーマ名を 1 つ以上使用するすべてのデータベース・オブジェクトがドロップされていることを確認し、別のスキーマ名を使用して、オブジェクトを再作成してください。この修正は、現在のリリースの前に使用していたリリースのデータベース・マネージャーで行う必要があります。現在のリリースで、データベースの移行コマンドを再発行してください。

2 現在のリリースの前に使用していたリリースのデータベース・マネージャーに戻ることによって、データベースの状態を訂正して、データベースに必要な訂正アクションを実行してください。現在のリリースで、データベースの移行コマンドを再発行してください。

3 データベース構成パラメーター `logfilsiz` または `logprimary` を大きな値に増やしてください。データベースの移行コマンドを再発行してください。

4 十分なディスク・スペースがあることを確認して、データベースの移行コマンドを再発行してください。

5 データベース構成ファイルも更新に問題がありました。データベース構成ファイルが他のユーザーによって排他的に保持されておらず、更新可能であることを確認してください。データベースの移行コマンドを再発行してください。問題が続く場合は、IBM 技術員に連絡してください。

6 データベース・バックアップからデータベースをリストアしてください。

7 データベース・バックアップからデータベースをリストアしてください。

8 データベースの移行コマンドを再発行し

- てください。問題が続く場合、IBM サービス担当者に連絡してください。
- 9** 現在のリリースの前に使用していたリリースのデータベース・マネージャーに戻ることによって、表スペースを訂正してください。表スペースの訂正に必要なアクションについてはメッセージ SQL0290N を参照してください。
- 10** タイプ名がシステムで予約済みです。タイプと、そのタイプを使用するデータベース・オブジェクトがドロップされ、予約済みではないタイプ名を使用して再作成されたことを確認してください。この修正は、現在のリリースの前に使用していたリリースのデータベース・マネージャーで行う必要があります。現在のリリースで、データベースの移行コマンドを再発行してください。
- 11** プロシージャがドロップされていることを確認し、別の特定名を使用してプロシージャを再作成してください。この修正は、現在のリリースの前に使用していたリリースのデータベース・マネージャーで行う必要があります。現在のリリースで、データベースの移行コマンドを再発行してください。
- 12** 抽象データ・タイプがドロップされていることを確認してください。この修正は、現在のリリースの前に使用していたリリースのデータベース・マネージャーで行う必要があります。現在のリリースで、データベースの移行コマンドを再発行してください。
- 13** 拡張索引がドロップされていることを確認してください。この修正は、現在のリリースの前に使用していたリリースのデータベース・マネージャーで行う必要があります。現在のリリースで、データベースの移行コマンドを再発行してください。
- 17** 以前のデータベース・マネージャー・システムで、データベースのバックアップをリストアしてください。表スペースにさらにコンテナを追加してください。データベースの移行用に 70% のフリー・スペースを割り振らなければなりません。現行リリースに戻ってから、データベースを移行します。
- 18** 直前のデータベース・マネージャー・システムに、データベースのバックアップをリストアしてください。重複する関数の 1 つをドロップしてください。必要であれば、別のシグニチャーを使って関数を作成し直してください。データベース・マネージャー・システムの現行リリースに戻してください。データベースの移行コマンドを再発行してください。「概説およびインストール」にある、DB2 の現行リリースへの移動前に行う、データベースの移行準備の確認方法に関する指示にしたがってください。
- 19** 直前のデータベース・マネージャー・システムに、データベースのバックアップをリストアしてください。関数またはプロシージャをドロップしてください。必要であれば、別のシグニチャーを使ってそれを作成し直してください。データベース・マネージャー・システムの現行リリースに戻してください。データベースの移行コマンドを再発行してください。「概説およびインストール」にある、DB2 の現行リリースへの移動前に行う、データベースの移行準備の確認方法に関する指示にしたがってください。
- 20** 直前のデータベース・マネージャー・システムに、データベースのバックアップをリストアしてください。要約データ・タイプをすべてドロップしてください。データベース・マネージャー・システムの現行リリースに戻してください。データベースの移行コマンドを再発行してく

ださい。「概説およびインストール」にある、DB2 の現行リリースへの移動前に行う、データベースの移行準備の確認方法に関する指示にしたがってください。

- 21** 管理通知ログをチェックして、ノード障害を報告しているノードを判別してください。ノード障害状態を訂正し、データベースの移行コマンドを再発行してください。移行は、移行を要するノードでしか行われないため、どのノードからでもコマンドを発行できます。
- 22** カタログ・ノードでのノード障害状態を訂正してください。移行コマンドを再発行してください。
- 23** 連合データベースを 32 ビットのインスタンスに移行してください。
- 24** ディレクトリー db2detaildeadlock がデータベース・ディレクトリーに存在する場合は除去し、移行コマンドを再実行依頼します。

---

**SQL1705W** データベース・ディレクトリー項目を、現在のリリース・レベルに更新できません。

**説明:** 1 つ以上のデータベース別名が、前のリリースから移行されたばかりのデータベースのデータベース・ディレクトリーで更新できませんでした。

**ユーザーの処置:** 移行済みデータベースのデータベース別名をアンカタログし、同じ情報を使用してデータベース別名を再カタログしてください。

---

**SQL1706W** ワード・サイズ・インスタンス移行中に、このインスタンスのノード・ディレクトリーでローカルではないデータベースが少なくとも 1 つ見つかりました。

**説明:** ワード・サイズ・インスタンス移行の実行中、このインスタンスで作成されていないデータ

ベースが少なくとも 1 つ見つかりました。移行を正しく完了するには、このようなデータベースが、このインスタンスと同じワード・サイズを持っていなければなりません。

**ユーザーの処置:** インスタンスでカタログされたデータベースがすべて、同じワード・サイズを持つようにしてください。

---

**SQL1707N** インスタンス・ワード・サイズを移行できません。

**説明:** インスタンスのワード・サイズを移行しようと試みているときに、エラーがありました。IBM サービス技術員に連絡してください。

**ユーザーの処置:** IBM サービス技術員に連絡してください。

---

**SQL1708W** データベース移行が完了しましたが、警告コード *warning-code* が出力されています。

**説明:** データベースの移行が完了しましたが、警告があります。警告コードは以下のとおりです。

**1** いくつかのノードを移行できなかった

**ユーザーの処置:** 警告コードに基づく解決方法:

**1** そのノードに対してデータベース移行コマンドを出し直してください。

---

**SQL1749N** 表 *table-name* が **NOT LOGGED INITIALLY** と一緒に作成されていないため、**NOT LOGGED INITIALLY** 属性はこの表では活動化されません。

**説明:** 表 *table-name* が指定された **NOT LOGGED INITIALLY** 属性と一緒に作成されなかったため、この属性は **ALTER TABLE** を使用して活動化できません。

ステートメントは処理できません。

**ユーザーの処置:** **NOT LOGGED INITIALLY** 文

節を ALTER TABLE ステートメントから除去してください。

**sqlcode:** -1749

**sqlstate:** 429AA

---

**SQL1750N** 外部キーは **NOT LOGGED INITIALLY** 文節で作成された親キーの表 *table-name* を参照できません。

**説明:** NOT LOGGED INITIALLY 文節で作成された表は外部キーで参照できません。

**ユーザーの処置:** ALTER または CREATE TABLE ステートメントの *table-name* へ外部キーを除去してください。

**sqlcode:** -1750

**sqlstate:** 429A0

---

**SQL1751N** ノード・グループの結果に区分化マップで使用可能なノードが入っている可能性があります。

**説明:** ノード・グループには、少なくとも 1 つは区分化マップで使用可能なノードが入っていません。ノードが ノード・グループ WITHOUT TABLESPACES に追加される場合、そのノードにノード・グループで定義された表スペース用のコンテナがないため、区分化マップ内にこのノードを組み込むことはできません。ノードがほかのノードに追加され、ほかのノードにノード・グループの表スペース用のコンテナがない場合は、ノードを区分化マップに組み込むことができます。

**ユーザーの処置:** 少なくとも 1 つのノードも追加しないで、すべてのノード・グループのノードをドロップしないでください。表スペースがノード・グループにすでに定義されているが表がない場合、少なくとも 1 つのノードがすべての表スペースに入っていることを確認してください。

**sqlcode:** -1751

**sqlstate:** 428C0

---

**SQL1752N** この表スペースをノード・グループ *ngname* で作成できません。

**説明:** 表スペースが SYSTEM TEMPORARY 表スペースの場合のみ、ノード・グループ IBMTEMPGROUP を指定できます。

**ユーザーの処置:** SYSTEM TEMPORARY 表スペースの場合、ノード・グループ IBMTEMPGROUP を指定してください。そのほかの表スペースの場合、IBMTEMPGROUP 以外のノード・グループを指定してください。

**sqlcode:** -1752

**sqlstate:** 429A1

---

**SQL1753N** ノード *node-number* には、ノード・グループ IBMTEMPGROUP に定義されたすべての SYSTEM TEMPORARY 表スペースのコンテナがありません。

**説明:** ノードが、このデータベースのノード・グループ IBMTEMPGROUP に定義されている、すべての SYSTEM TEMPORARY 表スペースにコンテナを定義していなければ、そのノードをノード・グループに組み込むことはできません。

**ユーザーの処置:** ALTER TABLESPACE ステートメントを発行し、このデータベースの SYSTEM TEMPORARY 表スペースのノードごとにコンテナを追加してください。

**sqlcode:** -1753

**sqlstate:** 57052

---

**SQL1754N** この索引表スペースまたはログ表スペースが 1 次表スペースと同じノード・グループ内にありません。

**説明:** CREATE TABLE ステートメントで指定されたすべての表スペースは、同じノード・グループに属していません。

**ユーザーの処置:** CREATE TABLE ステートメントで指定されたすべての表スペースは、同じノード・グループに属していることを確認してください。

**sqlcode:** -1754

**sqlstate:** 42838

---

**SQL1755N** ノード *node-number* は、ノード・グループ *ngname* で定義されたすべての表スペースのコンテナを持っていません。

**説明:** ノードは、ノードがノード・グループの再分散操作に組み込まれる前に、ノード・グループで定義されたすべての表スペースを定義したコンテナがなくてはなりません。

**ユーザーの処置:** ALTER TABLESPACE ステートメントを発行し、このノード・グループで定義されたすべての表スペース用のノードのコンテナを追加してください。

---

**SQL1756N** 複数の文節は、ON NODES 文節を使用せずにコンテナを指定します。

**説明:** CREATE TABLESPACE の場合、ON NODES 文節を使用しない USING 文節を一度のみ指定できます。

ALTER TABLESPACE の場合、ON NODES 文節を使用しない ADD 文節を一度のみ指定できます。

このステートメントは処理されませんでした。

**ユーザーの処置:** ステートメントを訂正して、もう一度やり直してください。

**sqlcode:** -1756

**sqlstate:** 428B1

---

**SQL1757N** ON NODES 文節を使用しない USING 文節がなくなっています。

**説明:** CREATE TABLESPACE ステートメントで、各 USING 文節が ON NODES 文節を指定します。しかし、すべてのノードがノード・グループに組み込まれていない場合、すべてのノード・グループはコンテナがありません。

このステートメントは処理できません。

**ユーザーの処置:** ON NODES 文節を使用しない USING 文節が指定されているまたは、ノード・グループの各ノードが一回 ON NODES 文節に組み込まれているかを確認してください。

**sqlcode:** -1757

**sqlstate:** 428B1

---

**SQL1758W** 特定ノードに指定されていないコンテナは、ほかのノードの表スペースで使用されません。

**説明:** ALTER TABLESPACE および CREATE TABLESPACE ステートメントには、このノード・グループのすべてのノードに対するコンテナ指定が組み込まれています。ON NODES 文節が後に続かないコンテナの指定は、冗長であるため無視されています。

このステートメントは処理されていません。

**ユーザーの処置:** 複数ノードでコンテナが必要な場合、ALTER TABLESPACE ステートメントを発行し、必要なコンテナを追加してください。

**sqlcode:** -1758

**sqlstate:** 01589

---



---

**SQL1759W** 再分散されたノード・グループはノード・グループ *nodegroup-name* のオブジェクトの区分化データを変更し、複数ノードを追加して組み込みまたはドロップしてドロップする必要があります。

**説明:** この警告は ALTER NODEGROUP または ALTER TABLESPACE ステートメントを使用する変更が、変更されるノード・グループのマップの区分化の原因になっていないことを示しています。ノード・グループの区分化マップを、ノード・グループで定義された表スペースまたは区分化マップにないドロップされるノードを使用して定義された表がない場合、次のステートメントで即時に変更のみします。

この警告は以下の場合に発行されます。

- 1 つ以上のノードが ALTER NODEGROUP ADD NODE を使用して追加された
- 1 つ以上のノードが ALTER NODEGROUP DROP NODE を使用してドロップされた
- コンテナが表スペースに対して追加され、それ以降コンテナは使用されるノードは要求されません。

これらのすべての場合、表はすでにノード・グループ内の表スペースを使用して定義されています。

**ユーザーの処置:** データ区分化のためにノードを組み込みまたは除外したい場合、REDISTRIBUTE NODEGROUP コマンドまたは API を発行してください。それに代わる方法として、すべての表をドロップするには、ノード・グループの表スペースを使用してください。

**sqlcode:** +1759

**sqlstate:** 01618

---

**SQL1760N** ストアード・プロシージャ *procedure-name* に対する CREATE ステートメントは有効な LANGUAGE 文節、EXTERNAL 文節および PARAMETER STYLE 文節を持っていない限りなりません。

**説明:** プロシージャ *procedure-name* の CREATE に、必須の文節がありません。LANGUAGE、EXTERNAL および PARAMETER STYLE を指定する必要があります。

**ユーザーの処置:** 足りない文節を追加した後で、もう一度やり直してください。

**sqlcode:** -1760

**sqlstate:** 42601

---

**SQL1761N** ノード・グループ *ngname* がバッファ・プール *bpname* に定義されていません。

**説明:** 表スペースのノード・グループがバッファ・プールに定義されていません。表スペースは、このノード・グループとバッファ・プールの組み合わせを使用するために作成したり変更したりできません。

ステートメントは処理できません。

**ユーザーの処置:** 可能なアクションは以下のとおりです。

- 表スペースのノード・グループを定義しているバッファ・プールを指定する
- バッファ・プールを変更して表スペースのノード・グループを追加する
- CREATE TABLESPACE の場合、バッファ・プールに定義されているノード・グループを指定する

**sqlcode:** -1761

**sqlstate:** 42735

---

**SQL1762N** アクティブ・ログ・ファイルに割り振るだけの十分なスペースがないため、データベースに接続することができません。

**説明:** アクティブ・ログ・ファイルに割り振るだけの十分なスペースがありません。以下のことが原因として考えられます。

- リカバリー・ログを保管するのに使用するデバイスで使用できるスペースが足りない。
- ユーザー出口が使用可能である場合は、正しくないパス、正しくないインストール・ディレクトリー、共用違反、またはその他の問題のために、ユーザー出口プログラムが失敗した可能性があります。

**ユーザーの処置:** それぞれの原因について、以下のように対処してください。

- データベースが少なくとも LOGPRIMARY ログ・ファイルで開始するために、DB2 では新規ログを割り振るための追加のスペースを必要とする場合があるので、1 次ログ用にデバイスに十分なスペースがあることを確認してください。リカバリー・ログが非アクティブであっても、スペースを解放するためにリカバリー・ログを削除してはいけません。
- ユーザー出口プログラムを手操作で呼び出すことにより、ユーザー出口プログラムが正しく操作していることを確認してください。ユーザー出口プログラムのコンパイルおよびインストールについては、ユーザー出口プログラムのサンプル・ソース・コードで提供されている指示を検討してください。アーカイブ宛先パスが存在しているかを確認してください。

最後の手段として、LOGPRIMARY または LOGFILSIZ データベース構成パラメーター、あるいはこの両方の値を減らして、アクティブ・ログ・ファイルの設定を小さくして使用するようにしてください。これにより、ディスク・スペースの所要量が減ります。

問題を判別し、訂正した後に、接続ステートメントを再発行してください。

---

**SQL1763N** ALTER TABLESPACE ステートメントに、複数の矛盾するコンテナー操作があります。

**説明:** 指定されたコンテナー操作は矛盾しています。同一ノードに対する同一 ALTER TABLESPACE ステートメントの中で、BEGIN NEW STRIPE SET コンテナー操作を、他のコンテナー操作 (ADD、DROP、EXTEND、REDUCE、または RESIZE) と一緒に使用することはできません。同一ノードに対する同一 ALTER TABLESPACE ステートメントの中で、表スペースに (ADD、EXTEND、または RESIZE を使って) ページを追加したり、(DROP、REDUCE、または RESIZE を使って) 表スペースからページを除去することはできません。

**ユーザーの処置:** BEGIN NEW STRIPE SET コンテナー操作とほかの操作を同一ステートメントの中で使用する場合は、BEGIN NEW STRIPE SET 操作を、その操作独自の ALTER TABLESPACE ステートメントに移動してください。表スペースへのページの追加と、表スペースからのページの除去の両方を行う場合は、2 つの ALTER TABLESPACE ステートメントを使用してください。

**sqlcode:** -1763

**sqlstate:** 429BC

---

**SQL1764N** ALTER TABLESPACE で、RESIZE アクション用に指定されたサイズが、現在の表スペースの大きさより小さくなっています。

**説明:** ALTER TABLESPACE ステートメントの RESIZE アクションを使用して指定したサイズが、現在の表スペースの大きさより小さくなっています。コンテナーのサイズのみを大きくすることができます。

**ユーザーの処置:** 表スペース・コンテナーの現在



のサイズの値よりも大きなサイズを指定してください。

**sqlcode:** -1764

## SQL20100 - SQL20199

---

**SQL20102N** ルーチン *routine-name* の **CREATE** または **ALTER** ステートメントが、このルーチンに許可されていない *option-name* オプションを指定しました。

**説明:** ルーチン *routine-name* を作成または変更するときに、オプション *option-name* が指定されました。このルーチンの他の特性により、このオプションはこのルーチンには適用されません。

**ユーザーの処置:** ALTER ステートメントの場合、正しいルーチンが指定されていることを確認してください。または、失敗したオプションを除去して、ステートメントを再発行してください。

**sqlcode:** -20102

**sqlstate:** 42849

---

**SQL20108N** 結果セットには、ストアード・プロシージャ *procedure-name* によってオープンされたカーソル *cursor-name* の位置 *position-number* にサポートされていないデータ・タイプが含まれています。

**説明:** 列の少なくとも 1 つ、列 *position-number* に DRDA アプリケーション・リクエスター (クライアント) または DRDA アプリケーション・サーバー (サーバー) のいずれかにサポートされていないデータ・タイプが含まれているため、*procedure-name* で示されているストアード・プロシージャは、*cursor-name* で示されている照会結果の少なくとも 1 つを返すことができません。このようなストアード・プロシージャへの呼び出しは失敗します。

**ユーザーの処置:** サーバーのストアード・プロシ

**sqlstate:** 560B0

ージャ *procedure-name* のカーソル *cursor-name* のための OPEN ステートメント (およびその後の FETCH ステートメント) を変更して、列 *position-number* でサポートされていないデータ・タイプを選択しないようにしてください。ストアード・プロシージャを呼び出したクライアント・アプリケーションは、ストアード・プロシージャの変更を反映するように変更しなければならない場合があります。

**sqlcode:** -20108

**sqlstate:** 56084

---

**SQL20109W** DB2 デバッガー・サポートでエラーが起きました。理由コード:  
*reason-code*

**説明:** デバッガー・サポートで、デバッグを使用不可にするエラー状況になりましたが、通常の実行には影響ありません。以下が理由コードのリストです。

1. デバッガー・サポートがインストールされていません。
2. デバッガー表のデバッガー・クライアントの IP アドレスに構文エラーがあります。
3. デバッガー・バックエンドとデバッガー・クライアントとの通信でタイムアウトが起きました。
4. デバッガー表 DB2DBG.ROUTINE\_DEBUG にアクセスしているときに、問題が起きました。

**ユーザーの処置:**

1. DB2 サーバー・マシンのデバッガー・オプションがインストールされているかどうか確認してください。

2. デバッガー表の IP アドレスが正しい構文になっていることを確認してください。
3. クライアントのデバッガー・デーモンが開始され、クライアントとサーバーのポートが一致していることをチェックしてください。
4. デバッガー表を正しいレイアウトで作成したことを確認してください。

**sqlcode:** +20109

**sqlstate:** 01637

---

**SQL20111N** このコンテキストでは **SAVEPOINT**、**RELEASE SAVEPOINT**、または **ROLLBACK TO SAVEPOINT** ステートメントを発行できません。理由コード = *reason-code*

**説明:** 以下の理由コードによって示されているような制限に違反しているため、このステートメントを処理できません。

1. 保管点をトリガー内で出すことはできません。
2. 保管点をグローバル・トランザクション内で出すことはできません。

ステートメントは処理できません。

**ユーザーの処置:** トリガーまたはグローバル・トランザクションにある **SAVEPOINT**、**RELEASE SAVEPOINT**、または **ROLLBACK TO SAVEPOINT** ステートメントを除去してください。

**sqlcode:** -20111

**sqlstate:** 3B503

---

**SQL20112N** **SAVEPOINT** がすでに存在し、ネストされた **SAVEPOINTS** はサポートされていないため、**SAVEPOINT** を設定することができません。

**説明:** **SAVEPOINT** または **ATOMIC** コンパウンド SQL ステートメントでエラーが起きました。保管点がすでに存在し、この環境ではネストされた保管点はサポートされていません。

**ユーザーの処置:** 既存の保管点名を再設定する必要がある場合、**RELEASE SAVEPOINT** ステートメントを出して既存の保管点を解放してから、**SAVEPOINT** ステートメントを出し直してください。 **ATOMIC** コンパウンド SQL の場合、複合ステートメントの終わりに達するまで、**SAVEPOINT** を設定することはできません。

**sqlcode:** -20112

**sqlstate:** 3B002

---

**SQL20113N** **SELF AS RESULT** で定義された方式 *method-id* から **NULL** を返すことはできません。

**説明:** 方式 ID *method-id* の方式が **SELF AS RESULT** で定義されています。この方式の呼び出しが構造化タイプの **NULL** ではないインスタンスを使用していたため、**NULL** インスタンスを返すことはできません。

**ユーザーの処置:** 方式の戻り値として **NULL** が返されないよう、方式設定を変更してください。1つの方法として、返される構造化タイプの属性をすべて **NULL** に変更することが可能です。障害のあった方式の名前を判別するには、以下の照会を使用してください。

```
SELECT FUNCSCHEMA, FUNCNAME,
 SPECIFICNAME
FROM SYSCAT.FUNCTIONS
WHERE FUNCID = method-id
```

**sqlcode:** -20113

**sqlstate:** 22004

---

**SQL20114W** 表 *table-name* の列 *column-name* の長さは、**USER** デフォルト値の定義された長さに対して十分ではありません。

**説明:** 列 *column-name* が、128 バイトに満たない長さで定義されています。この列に文節 **DEFAULT USER** が指定されました。USER 特殊レジスターが **VARCHAR(128)** と定義されているため、列の長さを超えるユーザー ID を持つユーザーが *table-name* にデフォルト値を割り当てようとした場合、エラーが起こります。ユーザー ID が列の長さを超えているユーザーは、この列を挿入、あるいは列をデフォルト値に更新することができません。

**ユーザーの処置:** システム標準によって、列の長さを超える ID が許可されていない場合、この警告を無視できます。この警告が出されないようにするには、列の長さを少なくとも 128 バイトにする必要があります。表をドロップしてから再作成することによって、またデータ・タイプが **VARCHAR** であれば、**ALTER TABLE** で列の長さを大きくすることによって、この列の長さを変更できます。

**sqlcode:** +20114

**sqlstate:** 01642

---

**SQL20115N** トランスフォーム・グループ *group-name* で、*routine-type* *routine-name* を *transform-type* トランスフォーム関数として使用することはできません。

**説明:** *routine-type* が **FUNCTION** であれば、*routine-name* によって定義されている関数は組み込み関数であるため、これをトランスフォーム関数として使用することはできません。*routine-type* が **METHOD** であれば、*routine-name* によって定義されている方式は方式であるため、これをト

ランスフォーム関数として使用することはできません。

ステートメントは処理できません。

**ユーザーの処置:** トランスフォーム・グループ *group-name* の *transform-type* トランスフォーム関数として、異なる関数を指定してください。

**sqlcode:** -20115

**sqlstate:** 428EX

---

**SQL20116N** 検索ターゲット *parameter-name* のデータ・タイプが、索引拡張子 *index-extension-name* に指定されているソース・キーのデータ・タイプに一致していません。

**説明:** 検索ターゲットが組み込みまたは特殊データ・タイプである場合、そのタイプは、索引拡張子に指定されているソース・キーのデータ・タイプに一致していなければなりません。検索ターゲットのデータ・タイプが構造化タイプであれば、索引拡張子のソース・キーのデータ・タイプと同じ構造化タイプ階層になければなりません。

ステートメントは処理できません。

**ユーザーの処置:** 索引拡張子のソース・キーに一致するデータ・タイプで検索ターゲットを指定してください。

**sqlcode:** -20116

**sqlstate:** 428EY

---

**SQL20117N** OLAP 関数のウィンドウ指定は無効です。理由コード = *reason-code*

**説明:** OLAP 関数呼び出しのウィンドウ指定 (**OVER** 文節) が正しく指定されていません。*reason-code* が、誤った指定について指示しています。

1 ウィンドウ指定で、**ORDER BY** なしで **RANGE** または **ROWS** が指定されています。

- 2 RANGE が指定されていますが、ウィンドウ ORDER BY 文節に複数の sort-key-expression が入っています。
- 3 RANGE が指定されていますが、ウィンドウ ORDER BY 文節の sort-key-expression のデータ・タイプを持つ減算式に、範囲値のデータ・タイプを使用することができません。
- 4 CURRENT ROW の後に UNBOUNDED PRECEDING が、あるいは CURRENT ROW の前に UNBOUNDED FOLLOWING が指定されています。

ステートメントは処理できません。

**ユーザーの処置:** ウィンドウ指定を変更して、*reason-code* によって示されている無効な指定を訂正してください。

- 1 RANGE または ROWS を指定しているウィンドウ指定ごとにウィンドウ ORDER BY 文節を追加します。
- 2 RANGE を備えたウィンドウ指定が、ウィンドウ ORDER BY 文節ごとに sort-key-expression を 1 つずつ持つようにします。
- 3 RANGE を備えたウィンドウ指定ごとに、ウィンドウ ORDER BY 文節の sort-key-expression から範囲値 (数値タイプまたは日付時刻タイプでなければなりません) を減算できることを確認してください。日付時刻 sort-key-expression の場合、範囲値は、適切な精度および位取りを持つ特定の日付時刻期間 DECIMAL タイプでなければなりません。
- 4 “BETWEEN” および “CURRENT ROW” を使用しているウィンドウ指定で、“UNBOUNDED PRECEDING” を “AND CURRENT ROW” の前に、または “UNBOUNDED FOLLOWING” を “CURRENT ROW AND” の後に指定してください。

**sqlcode:** -20117

**sqlstate:** 428EZ

---

**SQL20118N** 構造化タイプ *type-name* で、属性の数が許可されている最大数を超過しています。最大数は *max-value* です。

**説明:** 構造化タイプ *type-name* の定義で、構造化タイプごとに許可されている属性 (継承属性を含む) の最大数を超過しています。継承属性を含む属性の最大数は *max-value* です。

ステートメントは処理できません。

**ユーザーの処置:** 構造化タイプの属性の数が制限を超えないようにしてください。

**sqlcode:** -20118

**sqlstate:** 54050

---

**SQL20119N** ROW 関数は、少なくとも 2 つの列を定義していなければなりません。

**説明:** RETURNS 文節に ROW を指定する関数は、少なくとも 2 つの列がある列リストを備えていなければなりません。

**ユーザーの処置:** RETURNS 文節から ROW キーワードを除去してスカラー関数にするか、または RETURNS 文節の列リストに複数の列を指定してください。

**sqlcode:** -20119

**sqlstate:** 428F0

---

**SQL20120N** SQL TABLE 関数は表結果を返さなければなりません。

**説明:** RETURNS 文節に TABLE を指定する SQL 関数は、表である結果を返さなければなりません。スカラー全選択の場合は例外ですが、スカラー式は SQL TABLE 関数の結果として返されません。

**ユーザーの処置:** RETURNS 文節から TABLE キーワードを除去してスカラー関数にするか、または TABLE 関数本体の RETURNS ステートメントに全選択を指定してください。

**sqlcode:** -20120

**sqlstate:** 428F1

---

**SQL20121N WITH RETURN または SCROLL**  
はカーソル *cursor-name* にいずれか  
1 つだけ指定できます。

**説明:** WITH RETURN と SCROLL の両方がカーソル *cursor-name* に指定されましたが、これは許されていません。

ステートメントは処理できません。

**ユーザーの処置:** DECLARE CURSOR ステートメントを変更して、NO SCROLL を指定 (または SCROLL キーワードを除去) するか、WITH RETURN 文節を除去してください。

**sqlcode:** -20121

**sqlstate:** 428F3

---

**SQL20123** スクロール可能カーソルに対して戻された結果セットが、1 行目より前に位置付けられていないため、ストアード・プロシージャ *procedure* への CALL は失敗しました。

**説明:** スクロール可能結果セットがストアード・プロシージャ *procedure* への CALL によって戻され、1 つまたは複数のカーソルが 1 行目よりも前に位置付けられていません。

ストアード・プロシージャへの CALL は失敗しました。このストアード・プロシージャに定義されたすべての結果セットは、呼び出し元に戻る前にクローズされました。スクロール可能カーソルは、結果セットから FETCH するために使用できません。このストアード・プロシージャによって行われたアクションはロールバックされず、エラーがストアード・プロシージャの

実行の終わりに検出されたため、このストアード・プロシージャによって開始された外部アクションは完了しました。

**ユーザーの処置:** 呼び出し元に戻る前に、結果セットカーソルが 1 行目より前に位置付けられるように、ストアード・プロシージャの内容を変更してください。

**sqlcode:** -20123

**sqlstate:** 560B1

---

**SQL20128** カーソル *cursor-name* はスクロール可能ですが、結果表には表関数からの出力を組み込むことができません。

**説明:** カーソル *cursor-name* はスクロール可能になるよう定義されていますが、結果表には表関数からの出力が組み込まれます。この組み合わせはサポートされません。

ステートメントは処理できません。

**ユーザーの処置:** スクロール可能にならないようにカーソルの定義を変更するか、または結果表に表関数からの出力が組み込まれないことを確認してください。

**sqlcode:** -20128

**sqlstate:** 428F6

---

**SQL20131N** タイプ *object-type* のオブジェクト番号 *object-number* が、オブジェクトのリストに複数回指定されました。

**説明:** タイプ *object-type* のオブジェクト名のリストに、オブジェクト番号 *object-number* が複数回指定されました。ステートメントの操作をオブジェクトで複数回実行することはできません。

**ユーザーの処置:** リスト内の重複したオブジェクトを訂正し、重複するオカレンスを取り除いてください。(MDC の場合、オブジェクト・タイプは “dimension” となります。)

**sqlcode:** -20131

**sqlstate:** 42713

---

**SQL20133N** 操作 *operation-name* は外部ルーチン *routine-name* に対して実行できません。この操作は SQL ルーチンでのみ実行できます。

**説明:** 操作 *operation-name* を外部ルーチン *routine-name* に対して実行しようとした。しかし、この操作は SQL ルーチンに対してのみ実行が可能です。操作は正常に完了しませんでした。

**ユーザーの処置:** 指定した名前が SQL ルーチンを識別することを確認してください。

**sqlcode:** -20133

**sqlstate:** 428F7

---

**SQL20134N** ルーチン *routine-name* の SQL アーカイブ (SAR) ファイルがサーバーで作成できませんでした。

**説明:** DB2 が指定されたルーチンのライブラリーまたはバインド・ファイルを見つけられないため、ルーチン *routine-name* の SQL アーカイブ (SAR) の作成が失敗しました。バインド・ファイルは、DB2 バージョン 7.1、フィックスパック 2 以降を使用して作成された SQL ルーチンでのみ使用可能です。

**ユーザーの処置:** DB2 バージョン 7.1、フィックスパック 2 以降を使用して、サーバーにプロシージャを再作成し、操作をやり直してください。

**sqlcode:** -20134

**sqlstate:** 55045

---

**SQL20135N** 指定された SQL アーカイブがターゲット環境に適合しません。理由コード = *reason-code*

**説明:** 指定された SQL アーカイブが、以下のいずれかの理由でターゲット環境に適合しません。

- 1 ターゲット環境のオペレーティング・システムが、SQL アーカイブが作成されたオペレーティング・システムと異なります。
- 2 ターゲット環境のデータベース・タイプおよびレベルが、SQL アーカイブが作成されたデータベース・タイプおよびレベルと異なります。

**ユーザーの処置:** SQL アーカイブが作成された環境がターゲット環境と一致しているかを確認して、コマンドを再発行してください。環境が一致しない場合は、ターゲット環境を使用して SQL ルーチンを手操作で作成する必要があります。

**sqlcode:** -20135

**sqlstate:** 55046

---

**SQL20136N** ルーチン *routine-name* (特定名 *specific-name*) が、連合オブジェクトにアクセスしようとした。

**説明:** 外部関数またはメソッド内の SQL ステートメントは、1 つまたは複数の連合オブジェクトにアクセスしようとした。このステートメントは、ルーチン *routine-name* (特定名 *specific-name*) から実行されます。現行では、外部関数またはメソッドからの連合オブジェクトへのアクセスはサポートされていません。

**ユーザーの処置:** ルーチンから連合オブジェクトへの参照を除去します。

**sqlcode:** -20136

**sqlstate:** 55047



---

**SQL20138N** ルーチン *routine-name* (特定名 *specific-name*) は **MODIFIES SQL DATA (SQL データの変更)** と定義されていないため、ステートメントは説明されない可能性があります。

**説明:** ルーチン *routine-name* (特定名 *specific-name*) は CONTAINS SQL (SQL を含む) または READS SQL DATA (SQL データの読み取り) と定義されていますが、SQL ステートメントを EXPLAIN しようとした。SQL ステートメントを説明するには、EXPLAIN 表への書き込みが必要ですが、これは MODIFIES SQL DATA (SQL データの変更) ルーチンでしか許可されません。

**ユーザーの処置:** CONTAINS SQL (SQL を含む) または READS SQL DATA (SQL データの読み取り) と定義されているルーチンから SQL ステートメントの EXPLAIN を試行しないでください。

**sqlcode:** -20138

**sqlstate:** 42985

---

**SQL20139N** 直前のステートメントが失敗したか割り込まれたため、SQL ステートメントはルーチン *routine-name* (特定名 *specific-name*) で発行されない可能性があります。

**説明:** ルーチン *routine-name* (特定名 *specific-name*)、またはネストされたルーチンの実行中、ステートメントが失敗してロールバックが必要であるか、割り込みが発生しました。データベース・マネージャーが必要なリカバリーを実行できるように、最外部のステートメントから呼び出されるすべてのルーチンが実行を完了し、コントロールが最外部ステートメントに戻される必要があります。このリカバリーが完了するまで、SQL ステートメントは発行されない可能性があります。

**ユーザーの処置:** ルーチンは実行を継続します。

ルーチンはこの後の SQL ステートメントを実行せず、呼び出し元のステートメントにできるだけ速やかにコントロールを戻す必要があります。

割り込みの初期障害のリカバリーは、すべてのルーチンが完了した時点で、データベース・マネージャーにより自動的に実行されます。

**sqlcode:** -20139

**sqlstate:** 51038

---

**SQL20140W VALUE COMPRESSION** が表に対して非活動状態であるため、**COMPRESS** 列属性は無視されました。

**説明:** 以下の状態のいずれかが発生していると考えられます。

1. VALUE COMPRESS が表に対して非活動状態であるため、列に指定された COMPRESS SYSTEM DEFAULT が無視された。
2. DEACTIVATED VALUE COMPRESSION が指定され、列が COMPRESS SYSTEM DEFAULT とともに定義された。

**ユーザーの処置:** 列に指定された COMPRESS を許可するには、表を変更し、表の VALUE COMPRESSION を活動化してください。

**sqlcode:** 20140

**sqlstate:** 01648

---

**SQL20142N シーケンス *sequence-name* を指定されているようには使用できません。**

**説明:** *sequence-name* が、使用されないコンテキストで参照されました。 *sequence-name* は、ID 列用のシステムで生成されたシーケンスであってはなりません。これらのシーケンスは、COMMENT ON SEQUENCE、DROP SEQUENCE、GRANT または REVOKE ステートメント、あるいは NEXTVAL か PREVVAL 式では参照されません。

**ユーザーの処置:** このコンテキストに、ユーザー定義のシーケンス・オブジェクトの名前を指定してください。

**sqlcode:** -20142

**sqlstate:** 428FB

---

**SQL20143N ENCRYPTION PASSWORD 値が**  
設定されていないため、暗号化または暗号化解除関数が失敗しました。

**説明:** ENCRYPTION PASSWORD 値が設定されていません。

**ユーザーの処置:** SET ENCRYPTION PASSWORD ステートメントを発行して、ENCRYPTION PASSWORD 値を設定してください。パスワードの長さは最小 6 バイトから最大 127 バイトまでです。

**sqlcode:** -20143

**sqlstate:** 51039

---

**SQL20144N 指定されたパスワードの長さが 6**  
バイト未満か、または 127 バイトを超えていたため、暗号化パスワードが無効です。

**説明:** データは 6 バイトから 127 バイトの長さのパスワードで暗号化されなければなりません。

**ユーザーの処置:** パスワード長が 6 バイトから 127 バイトの範囲にあることを確認してください。

**sqlcode:** -20144

**sqlstate:** 428FC

---

**SQL20145N 暗号化解除関数が失敗しました。**  
暗号化解除に使用されたパスワードが、データの暗号化に使用されたパスワードと一致しません。

**説明:** データは、暗号化に使用されたものと同じパスワードを使用して暗号化解除されなければなりません。

**ユーザーの処置:** データの暗号化と暗号化解除に同じパスワードが使用されているかを確認してください。

**sqlcode:** -20145

**sqlstate:** 428FD

---

**SQL20146N 暗号化解除関数が失敗しました。**  
データは暗号化されません。

**説明:** データは ENCRYPT 関数の結果でなければなりません。

**ユーザーの処置:** データ・タイプが ENCRYPT 関数の結果であることを確認してください。

**sqlcode:** -20146

**sqlstate:** 428FE

---

**SQL20147N ENCRYPT 関数が失敗しました。**  
複数回の暗号化はサポートされていません。

**説明:** すでに暗号化されたデータを再び暗号化することはできません。

**ユーザーの処置:** データが暗号化されていないことを確認してください。

**sqlcode:** -20147

**sqlstate:** 55048



---

**SQL20148N** ルーチン *routine-name* (名前 *specific-name*) は、コンパウンド本体の最後の SQL ステートメントとして RETURN ステートメントを持っています。

**説明:** RETURN ステートメントは、SQL ROW または TABLE 関数でコンパウンド本体の最後の SQL ステートメントでなければなりません。ルーチン本体内では、他の RETURN ステートメントは許可されていません。

**ユーザーの処置:** RETURN ステートメントが 1 つのみで、そのステートメントがコンパウンド本体の最後の SQL ステートメントであることを確認してください。

**sqlcode:** -20148

**sqlstate:** 429BD

---

**SQL20149W** バッファース・プール構成は完了しましたが、次にデータベースが再始動されるまでは有効になりません。

**説明:** バッファース・プールは正常に構成されましたが、変更は即時に有効になりません。変更は、データベースが再始動したときに有効になります。

**ユーザーの処置:** 次のデータベースの再始動時に変更が有効となるために、アクションを行う必要はありません。

**sqlcode:** 20149

**sqlstate:** 01649

---

**SQL20150N** バッファース・プールのブロック・ページ数が、バッファース・プールのサイズには大きすぎます。

**説明:** NUMBLOCKPAGES で指定されたブロック・ページ数を、SIZE で指定されたバッファース・プールのページ数の 98% より大きくしないでください。

**ユーザーの処置:** NUMBLOCKPAGES をゼロに

設定することにより、このバッファース・プールのブロック・ベースの入出力を使用不可にするか、NUMBLOCKPAGES の値を SIZE の値の 98% より大きくしないようにしてください。

**sqlcode:** -20150

**sqlstate:** 54052

---

**SQL20151N** BLOCKSIZE に指定された値が有効範囲内にありません。

**説明:** BLOCKSIZE の値の有効範囲は 2 から 256 の間です。

**ユーザーの処置:** BLOCKSIZE の値を、2 以上、256 以下に変更してください。最適な値は、エクステント・サイズです。

**sqlcode:** -20151

**sqlstate:** 54053

---

**SQL20152N** 指定されたバッファース・プールが現在ブロック・ベースではありません。

**説明:** バッファース・プールのブロック・エリアで使用するページ数を指定せずに、BLOCKSIZE オプションが使用されました。

**ユーザーの処置:** BLOCKSIZE オプションと一緒に、NUMBLOCKPAGES の値を指定してください。

**sqlcode:** -20152

**sqlstate:** 428FF

---

**SQL20153N** データベースの分割イメージが延期状態にあります。

**説明:** データベースの分割イメージが延期状態にある間は使用できません。

**ユーザーの処置:** 以下の 3 つのオプションのいずれかを使用して db2inidb コマンドを発行し、このデータベースの分割イメージの入出力を再開してください。

- db2inidb <db-name> as mirror
- db2inidb <db-name> as snapshot
- db2inidb <db-name> as standby

マルチノード環境では、データベースを使用できるようにするためには、db2inidb ツールをノードごとに実行する必要があります。db2inidb ツールはマルチノード・データベースのノードごとに並行して実行することが可能です。

**sqlcode:** -20153

**sqlstate:** 55040

**SQL20154** 行に対してターゲット表を決定できないため、要求されたビュー *view-name* への挿入操作は許可されません。理由コード = *reason-code*

**説明:** 指定したビューに UNION ALL 照会が含まれています。 *reason-code* は、指定した行が以下のいずれかであることを示します。

1. 基礎となる基本表のチェック制約を満たしていない。
2. 1 つ以上の基礎となる基本表のすべてのチェック制約を満たしている。

連合システム・ユーザー: データ・ソースの他の特定の制限により、行の挿入が妨げられている可能性もあります。

**ユーザーの処置:** 行セットをパーティションする、基礎となる基本表で使用されているチェック制約が、挿入される行のセットをカバーとすることを確認してください。たとえば、チェック制約 (T1.c1 in (1,2)) on T1、および (T2.c1 in (2,3)) on T2、およびビュー V1 を T1 と T2 の union として指定。

1. 行 c1 = 4 は、基礎となる基本表のチェック制約を満たしていない。
2. 行 c1 = 2 は、基礎となる基本表のチェック制約を満たしている。

連合システム・ユーザー: 理由が不明な場合には、要求を失敗させたデータ・ソースに対して問題を分離し (「問題判別の手引き」を参照してください)、そのオブジェクト定義と更新制限を調べてください。

**sqlcode:** -20154

**sqlstate:** 23513

**SQL20155N** 指定したイベント・モニターのターゲット表が無効です。理由コード = *reason-code*

**説明:** CREATE EVENT MONITOR ステートメントの処理中に、あるいはイベント・モニターを活動化しているときに、ターゲット表が無効であると判別されました。 *reason-code* は以下のいずれかです。

1. 少なくとも 1 つの列名が、イベント・モニターのデータ・エレメント ID と一致しない。
2. 少なくとも 1 つの列に、イベント・モニターのデータ・エレメント ID のデータ・タイプと互換性のないデータ・タイプがある。
3. 少なくとも 1 つの列名が、イベント・モニターのデータ・エレメント ID と一致するが、そのエレメントはターゲット表で許可されていない。
4. 表の行サイズが、表スペースのページ・サイズに対して大きすぎる。
5. 必須列がない。

**ユーザーの処置:** 詳細については、管理通知ログをチェックし、表の定義を訂正してください。

**sqlcode:** -20155

**sqlstate:** 55049

---

**SQL20156W** イベント・モニターは正常に活動化されましたが、いくつかのモニター情報が脱落した可能性があります。

**説明:** イベント・モニターは正常に活動化されましたが、以下のいずれかの理由により、イベント・モニターのいくつかの情報が脱落した可能性があります。

- 列のデータ・タイプが、イベント・モニター・エレメント ID を保留するのに必要なデータ・タイプよりも小さい。このデータは切り捨てられます。
- ターゲット表が SYSCAT.EVENTTABLES で見つかったが、表がデータベースに存在しない。対応する表のモニター情報は記録されません。

**ユーザーの処置:** 詳細情報については管理通知ログをチェックしてください。必要であれば、すべてのターゲット表を作成するイベント・モニターを再作成してください。

**sqlcode:** 20156

**sqlstate:** 01651

---

**SQL20157N** ユーザー *authorization-ID* に **QUIESCE\_CONNECT** 特権がありません。

**説明:** 指定された許可 ID に、現在静止モードのデータベースまたはインスタンスにアクセスするための QUIESCE\_CONNECT 特権がありません。静止状態のデータベースまたはインスタンスに接続する前に、QUIESCE\_CONNECT 特権を付与される必要があります。

**ユーザーの処置:** 静止解除が完了するのを待機するか、データベースのシステム管理者またはデータベース管理者に連絡して、許可 ID に対して GRANT QUIESCE\_CONNECT 要求を発行してください。コマンドを再発行してください。

**sqlcode:** -20157

**sqlstate:** 08004

---

**SQL20158N** 関数は、接続されている DB2 Data Links Manager のレベルではサポートされていません。

**説明:** 基礎 DATALINK 機能は、接続されている DB2 Data Links Manager のレベルではサポートされていません。

**ユーザーの処置:** エラーを訂正するには、サポートされていない DATALINK 機能の使用を除去するか、DB2 Data Links Manager をデータベース・サーバーと同じレベルにアップグレードしてください。

**sqlcode:** -20158

**sqlstate:** 42997

---

**SQL20159W** ステートメント・コンテキストのために、分離文節は無視されます。

**説明:** インライン SQL として処理される複合ステートメントに含まれるステートメントで分離文節が指定されています。分離文節は無視され、分離レベルが、複合ステートメント内のすべてのステートメントで使用されます。

**ユーザーの処置:** 応答は必要ありません。この警告を回避するには、分離文節を除去してください。

**sqlcode:** 20159

**sqlstate:** 01652

---

**SQL20160W** 許可は **USER** *userid* に対して付与されました。グループは、許可名が 8 バイトより長いために考慮に入れられませんでした。

**説明:** 許可名の長さが 8 バイトを超えています。特権は、一致する名前があるシステムで定義されたグループを認識せずに、認可名 *userid* でユーザーに認可されています。処理を続行します。

**ユーザーの処置:** 認可がユーザーに対してのものである場合、アクションは必要ありません。認可

がグループに対してのものである場合、8 バイトを超えるグループ名はサポートされていないため、代替のグループ名を選択する必要があります。この警告メッセージを防ぐには、許可名の前に USER キーワードを指定してください。

**sqlcode:** 20160

**sqlstate:** 01653

---

**SQL20161W** 列名 *column-name* は、イベント・モニター表には無効な列です。

**説明:** INCLUDES または EXCLUDES 文節に指定された列名が、作成しているイベント・モニター表のどの有効な列とも一致しません。

**ユーザーの処置:** 指定した列名を確認して、訂正してください。

**sqlcode:** -20161

**sqlstate:** 428AA

---

**SQL20162N** バッファ・プールで、ブロック・ベースの入出力と拡張ストレージの両方を使用することはできません。

**説明:** ブロック・ベースの入出力と拡張ストレージの両方を使用するバッファ・プールを作成または変更しようとしてしました。

**ユーザーの処置:** バッファ・プールで、ブロック・ベースの入出力と拡張ストレージの両方を使用していないことを確認してください。

**sqlcode:** -20162

**sqlstate:** 428FF

---

**SQL20167N** 共有メモリの設定が、要求を処理するのに十分な大きさではありません。

**説明:** 以下のいずれかのリソース要求に対して、共有メモリ設定のメモリが不十分です。

- FCM 接続エントリー
- FCM メッセージ・アンカー

- FCM 要求ブロック

**ユーザーの処置:** 要求を再度試行してください。エラーが続く場合は、インスタンスを停止および再始動し、(DB2 によって自動的に更新される) 共有メモリ設定サイズ拡大を有効にして、要求を再試行してください。

**sqlcode:** -20167

**sqlstate:** 57011

---

**SQL20168N ALTER BUFFERPOOL ステートメントが現在進行中です。**

**説明:** ALTER 操作が進行中の場合は、バッファ・プールをドロップまたは変更することはできません。

**ユーザーの処置:** ALTER 操作が完了するまで待機してください。進行中の ALTER 操作の進行を確認するには、スナップショット・モニターを使用してください。

**sqlcode:** -20168

**sqlstate:** 55051

---

**SQL20169W バッファ・プールが開始されていません。**

**説明:** ステートメントは成功しましたが、バッファ・プールが開始されてから有効になります。

**ユーザーの処置:** ステートメントは正常に完了しました。データベースが活動化されたときに有効になります。これは、すべてのアプリケーションが切断された後に起こります。即時にステートメントを有効にするには、バッファ・プールが開始されたときに再サブミットしてください。バッファ・プールを開始するには、IMMEDIATE オプションを使用してバッファ・プールをドロップおよび再作成する方法が可能な場合があります。

**sqlcode:** +20169

**sqlstate:** 01654

---

**SQL20170N** 指定したアクションを行うための十分なスペースが、表スペース *tablespace-name* ありません。

**説明:** DROP、REDUCE、RESIZE コンテナ・アクションのうち 1 つ以上を使用して、スペースを表スペースから除去しています。ところが、除去中のスペース量が、スペース量の上限基準点を超えています。

**ユーザーの処置:** スナップショット・モニターを使用して、表スペースで使用できるページ数と、表スペースの上限基準点を調べてください。これらの値の差が、除去できるページの最大数です。

**sqlcode:** -20170

**sqlstate:** 57059

---

**SQL20173W** イベント・モニターが正常に作成されましたが、少なくとも 1 つのイベント・モニター・ターゲット表がすでに存在しています。

**説明:** イベント・モニターを作成中に、データベース・マネージャーが 1 つ以上のターゲット表を作成しようとしたのですが、それらの名前の表はすでに存在していることが判別しました。イベント・モニターは正常に作成されましたが、ターゲット表は同じ名前のある表を置換せず、作成されませんでした。イベント・モニターが活動化されると、これらの表の使用を試みます。活動化処理でこれらの表が適切でない判断されると、イベント・モニターの活動化で失敗する可能性があります。

**ユーザーの処置:** 詳細情報については管理通知ログをチェックしてください。必要であれば、すべてのターゲット表をドロップし、すべてのターゲット表を作成するイベント・モニターを再作成してください。

**sqlcode:** 20173

**sqlstate:** 01655

---

**SQL20178N** ビュー *view-name* には、すでに **INSTEAD OF operation** トリガーが定義されています。

**説明:** ビュー *view-name* には、すでに **INSTEAD OF** トリガーが指定された操作 (UPDATE, DELETE, INSERT) で定義されています。ビューの操作ごとに 1 つの **INSTEAD OF** トリガーのみを定義できます。

ステートメントは処理できません。 **INSTEAD OF** トリガーは作成されませんでした。

**ユーザーの処置:** 新規 **INSTEAD OF** トリガーが必要であれば、既存のトリガーをドロップしてから、新規トリガーを作成してください。

**sqlcode:** -20178

**sqlstate:** 428FP

---

**SQL20179N** **WITH CHECK OPTION** を使用してビュー *view-name* が定義されているため、**INSTEAD OF** トリガーを作成できません。

**説明:** **INSTEAD OF** トリガーを対称ビュー (**WITH CHECK OPTION** を使用して定義したビュー) で、あるいは、直接または間接的に対称ビューが定義されたビューで定義することはできません。ビュー *view-name* は **INSTEAD OF** トリガーのターゲット・ビューである可能性があります。あるいは、トリガーのターゲット・ビューに直接または間接的に依存するビューである可能性があります。

ステートメントは処理できません。 **INSTEAD OF** トリガーは作成されませんでした。

**ユーザーの処置:** 対称ビューが対称でないビューと置き換えられるのでなければ、**INSTEAD OF** トリガーをビューに定義することはできません。ビューをドロップし、**INSTEAD OF** トリガーを許可するよう再作成する必要があります。

**sqlcode:** -20179

**sqlstate:** 428FQ

---

**SQL20188N** *name* によって識別される主キーまたはユニーク・キーは、**ORGANIZE BY** 文節の列のサブセットです。

**説明:** 主キーまたはユニーク・キーのすべての列が、表の **ORGANIZE BY** 文節に組み込まれています。この場合、この表のページの各ブロックに 1 つの行しか含まれず、そのブロックの残りのスペースがすべて無駄になるため、これは許可されていません。

指定された場合、*name* は主キーまたはユニーク制約の制約名です。制約名が指定されなかった場合は、*name* が 3 つのピリオドが後に続く主キーまたはユニーク制約文節に指定された最初の列名になります。ユニーク索引が作成される場合、*name* がそのユニーク索引の名前になります。

ステートメントは処理できません。

**ユーザーの処置:** キーが次元のサブセットにならないように、主キーまたはユニーク・キーの定義、可能であれば、ユニーク索引定義または **ORGANIZE BY** 文節を変更してください。

**sqlcode:** -20188

**sqlstate:** 429BE

---

**SQL20189W** バッファ・プール操作 (**CREATE/ALTER**) は、メモリ不足により、次回データベースが再始動されるまで有効になりません。

**説明:** **CREATE** または **ALTER BUFFERPOOL** ステートメントが発行されて、正常に完了しましたが、メモリ不足のため、作成 / 変更は **DEFERRED** (延期) されました。この変更は、次回データベースが始動されたときに有効になります。

**ユーザーの処置:** バッファ・プールの活動化またはサイズ変更を、次の始動まで待ちたくない場合は、メモリ・リソースを解放し、(同じサイズか異なるサイズで) 再試行することができます。

す。削減できるメモリ・リソースには、他のバッファ・プール、データベース・ヒープ、カタログ・キャッシュ、パッケージ・キャッシュ、およびユーティリティ・ヒープなどがあります。これらは、リソースに応じ、**ALTER/DROP BUFFERPOOL** または **UPDATE DATABASE CONFIGURATION** コマンドを使って削減できます。今後、バッファ・プール・メモリの動的割り振りのための余分なメモリを予約するためには、**DATABASE\_MEMORY** データベース構成パラメータを増やすことができます。

再試行しない場合:

1. 失敗したのが **ALTER BUFFERPOOL** の場合は、バッファ・プールの現在のランタイム・サイズで実行を続けます。バッファ・プールの現在のランタイム・サイズは、データベース・モニターを使って確認できます。
2. 失敗したのが **CREATE BUFFERPOOL** の場合は、バッファ・プールで作成される表スペースは、一時的に (次の始動まで)、一致するページ・サイズを持つ、隠れたバッファ・プールに入れられます。隠れたバッファ・プールは小さいため、この結果、性能が低下する可能性があります。

再試行する場合:

1. **ALTER BUFFERPOOL** の場合、コマンドを再発行してください。
2. **CREATE BUFFERPOOL** の場合、バッファ・プールをドロップして、コマンドを再発行してください。

**sqlcode:** 20189

**sqlstate:** 01657

---

**SQL20190N** データの不整合の可能性があるため、連合挿入、更新、または削除操作はコンパイルできません。

**説明:** データ・ソースがアプリケーション savepoint サポートを提供しておらず、サーバー・



オプション 'iud\_app\_svpt\_enforce' が 'Y' に設定されている場合、連合挿入、更新、または削除操作は再コンパイル中にブロックされます。これにより、連合挿入、更新、または削除処理中にエラーを検出した場合、ランタイム実行時に発生する可能性のある、データ不整合の発生が回避されます。

**ユーザーの処置:** 解決策は以下のとおりです。

- サーバー・オプション 'iud\_app\_svpt\_enforce' を 'N' に変更する。
- 挿入、更新、または削除を直接データ・ソースに適用する。

**sqlcode:** -20190

**sqlstate:** 0A503

---

**SQL20191N** 動的に準備された **CALL** ステートメントの **INOUT** パラメーターの場合、同じホスト変数を **USING** 文節と **INTO** 文節の両方で使用する必要があります。

**説明:** **CALL** ステートメントでは、**INOUT** パラメーターは単一ホスト変数に対応している必要があります。動的に準備された **CALL** ステートメントを実行する場合は、同じホスト変数を **EXECUTE** ステートメントの **USING** 文節と **INTO** 文節の両方に指定する必要があります。

**ユーザーの処置:** **INOUT** パラメーターのパラメーター・マーカに対応するホスト変数を指定する場合は、**EXECUTE** ステートメントの **USING** 文節と **INTO** 文節の両方に同じホスト変数を使用してください。**SQLDA** を使って **INOUT** パラメーターのパラメーター・マーカに対応するホスト変数を指定する場合は、対応する **SQLVAR** の **SQLDATA** ポインターが同じホスト変数を指している必要があります。

**sqlcode:** -20191

**sqlstate:** 560BB

---

**SQL20192N** 指定されたモードは、パーティション・データベース環境でしかサポートされていません。

**説明:** **RECOMMEND PARTITIONINGS** または **EVALUATE PARTITIONINGS** がパーティション・データベース環境で呼び出されませんでした。これら 2 つのモードは、パーティション・データベース環境でのみサポートされています。

**ユーザーの処置:** パーティション・データベース環境で **RECOMMEND PARTITIONINGS** または **EVALUATE PARTITIONINGS** を呼び出してください。

**sqlcode:** -20192

**sqlstate:** 56038

---

**SQL20193N** ファイル *file-name* にアクセス中にエラーが発生しました。理由コード: *reason-code*

**説明:** ファイル名が *db2nodes.cfg* の場合、以下のような理由コードが考えられます。

1. *db2nodes.cfg* ファイルにフォーマット・エラーが含まれていることを示している。
2. 通信エラーのため、コマンドを完了できなかった。
3. メモリー割り振りエラーのため、コマンドを完了できなかった。

**ユーザーの処置:** *db2nodes.cfg* ファイルのフォーマット・エラーを修正します。*db2diag.log* ファイルで、通信またはメモリー割り振りエラーを解決するための詳細情報をチェックします。アプリケーションを再実行します。

**sqlcode:** -20193

**sqlstate:** 560BC

---

## SQL20194N バッファー・プール

*bufferpool-name* は、データベース・パーティション *dbpartitionnum* に存在しません。

**説明:** ALTER BUFFERPOOL ステートメントは、バッファー・プール *bufferpool-name* を指定していますが、これはデータベース・パーティション *dbpartitionnum* に存在しません。

**ユーザーの処置:** ALTER DATABASE PARTITION GROUP ステートメントを使って、データベース・パーティション *dbpartitionnum* を、すでにバッファー・プール *bufferpool-name* が定義されているデータベース・パーティション・グループに追加します。バッファー・プールが特定のデータベース・パーティション・グループと関連付けられていない場合は、そのデータベース・パーティションを任意のデータベース・パーティション・グループに追加するか、このデータベース・パーティションのために新しいデータベース・パーティション・グループを作成します。ALTER BUFFERPOOL ステートメントを再度発行します。

**sqlcode:** -20194

**sqlstate:** 53040

---

## SQL20195N パスの名前変更構成ファイル

*config-file* の行番号 *line-number* を処理中に、エラーを検出しました。  
理由コード = *reason-code*

**説明:** 現在のクラッシュまたはロールフォワード・リカバリー処理中に、パスの名前変更構成ファイル *config-file* が、コンテナの名前変更で使用されています。しかし、このファイルのステートメントを処理している間に、*line-number* 行目でエラーが検出され、それによりリカバリー処理の進行が妨げられました。このエラーの説明は、以下の理由コードで示されています。

1. 構文が無効である。

2. コンテナ・パスの長さがコンテナ・パスに許可されている最大長を超えている。
3. 指定されたコンテナ・パスが直前の行にすでにリストされている。
4. データベース・パスの長さがデータベース・パスに許可されている最大長を超えている。
5. 指定されたデータベース・パスが直前の行にすでにリストされている。
6. データベース・パスは、絶対パスでなければならない。
7. データベース・パスが誤って指定されている。

**ユーザーの処置:** 以下に記述されているように構成ファイルに該当する変更を行い、コマンドを再発行してください。

1. 構文エラーを訂正する。
2. 短いコンテナ・パスを指定する。
3. ファイルから重複するコンテナ・パスを除去する。
4. 短いデータベース・パスを指定する。
5. ファイルから重複するデータベース・パスを除去する。
6. 相対パスではなく、絶対パスでデータベース・パスを指定する。
7. ドライブ名の後にコロンを付ける形式でデータベース・パスを指定する (たとえば C:)

**sqlcode:** -20195

**sqlstate:** 08504

---

**SQL20196N FROM SQL 関数または方式から戻される 1 つまたは複数の組み込みタイプが、TO SQL 関数または方式のパラメーターである、対応する組み込みタイプと一致しません。**

**説明:** FROM SQL トランスフォーム関数または方式から戻される組み込みタイプは、対応する TO SQL トランスフォーム関数または方式のパラ



メーター・リストにあるタイプと一致している必要があります。

**ユーザーの処置:** 別の FROM SQL トランスフォーム関数または方式か、TO SQL トランスフォーム関数または方式を選択するか、FROM SQL トランスフォーム関数または方式か TO SQL トランスフォーム関数または方式を変更して、FROM SQL 関数または方式から戻される各組み込みタイプが、TO SQL トランスフォーム関数または方式のパラメーターである、対応する組み込みタイプと一致することを確認してください。

**sqlcode:** -20196

**sqlstate:** 428FU

---

**SQL20197N** *method-name* をオーバーライド方式として定義できません。理由コード *reason-code*。

**説明:** *method-name* をオーバーライド方式として定義しようとした。この方式と元の方式の間のオーバーライド・リレーションシップは、*reason-code* によって指定された以下のいずれかの理由により作成できません。

- 1 同じ名前を持つ元の方式が見つからない。
- 2 元の方式とオーバーライド方式の持つパラメーターの数が同じではない。
- 3 オーバーライド方式のパラメーターのデータ・タイプが、元の方式の対応するパラメーターのデータ・タイプと一致しない。
- 4 オーバーライド方式のパラメーターのパラメーター名が、元の方式の対応するパラメーターのパラメーター名と一致しない。
- 5 オーバーライド方式のパラメーターのロケーター指示が、元の方式の対応するパラメーターのロケーター指示と一致しない。

- 6 オーバーライド方式のパラメーターの FOR BIT DATA 指示が、元の方式の対応するパラメーターの FOR BIT DATA 指示と一致しない。
- 7 オーバーライド方式の RETURNS 文節に、AS LOCATOR 文節として CAST FROM 文節または FOR BIT DATA 文節が含まれています。
- 8 以下の継承された方式属性の 1 つが指定されている: SELF AS RESULT、SQL ルーチン特性、または外部ルーチン特性。
- 9 オーバーライド方式の戻りタイプが、元の方式の戻りタイプと一致しない。
- 10 オーバーライド方式の戻りタイプが、以下のいずれかサブタイプでもない構造型である。
  - 元の方式の戻りタイプ。
  - すでに元の方式をオーバーライドしているすべての方式の各戻りタイプ。

**ユーザーの処置:** 以下のように、*reason-code* によって指定された理由に基づいて方式定義を変更してください。

- 1 構造型のスーパータイプに対する既存の方式を識別する方式名を使用する。
- 2 元の方式と同じ数のパラメーターをオーバーライド方式に定義する。
- 3 元の方式の対応するパラメーターのデータ・タイプと一致するように、データ・タイプを変更する。
- 4 元の方式の対応するパラメーターの名前と一致するように、パラメーター名を変更する。
- 5 元の方式の対応するパラメーターのロケーター指示と一致するように、ロケーター指示を変更する。
- 6 元の方式の対応するパラメーターの

FOR BIT DATA 指示と一致するように、FOR BIT DATA 指示を変更する。

- 7 オーバーライド方式に対する AS LOCATOR 文節、CAST FROM 文節、および FOR BIT DATA 文節を除去する。
- 8 SELF AS RESULT 方式属性と、SQL ルーチン特性または外部ルーチン特性を除去する。
- 9 元の方式の戻りタイプと一致するように、戻りタイプを変更する。
- 10 元の方式と、元の方式をすでにオーバーライドしているすべての方式のサブタイプになるように、戻りタイプを変更する。

## SQL20200 - SQL20299

---

**SQL20200N** *url* が見つからなかったため、*jar-id* のインストールまたは置換が失敗しました。

**説明:** jar インストールまたは置換プロシージャで指定された URL が、有効な jar ファイルを表していませんでした。

**ユーザーの処置:** 有効な jar ファイルを識別する URL で、jar インストールまたは置換プロシージャを再実行してください。

**sqlcode:** -20200

**sqlstate:** 46001

---

**SQL20201N** jar 名が無効であるため、*jar-id* のインストール、置換または除去が失敗しました。

**説明:** jar インストール、置換または除去プロシージャに、無効な jar 名が指定されました。たとえば、jar ID が不適切な形式であるか、置換または除去の対象として存在しないか、あるいはすでに存在するためにインストールできない可能性があります。

**sqlcode:** -20197

**sqlstate:** 428FV

---

**SQL20198N** 方式 *method-name* は、自分自身を再帰呼び出ししています。

**説明:** 方式が自分自身を再帰的に呼び出そうとしました。再帰の性質は、直接または間接の場合があります。

**ユーザーの処置:** 方式の定義を変更してください。

**sqlcode:** -20198

**sqlstate:** 55054

---

**ユーザーの処置:** jar id が正しい形式であることを確認してください。jar id が存在する場合、インストールする前にそれを除去するようお勧めします。除去または置換プロシージャの場合は、jar id が存在することを確認してください。

**sqlcode:** -20201

**sqlstate:** 46002

---

**SQL20202N** *class* が使用中であるため、*jar-id* の置換または除去が失敗しました。

**説明:** jar ファイル内で指定されているクラスが定義済みプロシージャによって現在使用されているか、置換 jar ファイルにプロシージャが定義されている指定のクラスが入っていません。

**ユーザーの処置:** ドロップされるクラスを参照しているすべてのプロシージャがドロップされていることを確認し、置換または除去プロシージャを再実行依頼してください。

**sqlcode:** -20202

**sqlstate:** 46003

---

**SQL20203N** ユーザー定義関数またはプロシージャ *function-name* が、無効なシグニチャーのある Java 方式を持っています。

**説明:** 関数またはプロシージャを実装するために使用される java 方式のシグニチャーが無効でした。たとえば、方式が対応する create ステートメントのパラメーターにマッピング可能でないパラメーターを持っているか、プロシージャの方式が戻り値を指定している可能性があります。

**ユーザーの処置:** Java 方式に一致するパラメーターを指定して、対応する CREATE ステートメントを再発行するか、パラメーターまたは Java 方式の戻り値を訂正してクラスを再作成してください。

**sqlcode:** -20203

**sqlstate:** 46007

---

**SQL20204N** ユーザー定義関数またはプロシージャ *function-name* が単一の Java 方式にマッピングできませんでした。

**説明:** 示されている関数またはプロシージャが一致する Java 方式を見つけれなかったか、複数の一致する Java 方式を見つけました。

**ユーザーの処置:** Java 方式、あるいは対応する create ステートメントを訂正して、関数またはプロシージャ呼び出しが 1 つの Java 方式を決定できるようにしてください。

**sqlcode:** -20204

**sqlstate:** 46008

---

**SQL20205N** ユーザー定義関数またはプロシージャ *function-name* に、方式に渡すことができなかった NULL 値を持つ入力引き数があります。

**説明:** "CALLED ON NULL INPUT" で作成された関数、またはプロシージャが NULL 値を持

つ入力パラメーターを持っていますが、この引き数の Java データ・タイプは NULL 値をサポートしていません。NULL 値をサポートしていない Java データ・タイプの例は BOOLEAN、BYTE、SHORT、INT、LONG、または DOUBLE です。

**ユーザーの処置:** 方式が NULL 値で呼び出される場合は、入力 Java タイプが NULL 値を受け入れ可能であることを確認してください。関数の場合、"RETURNS NULL ON NULL INPUT" で関数を作成することができます。

**sqlcode:** -20205

**sqlstate:** 39004

---

**SQL20206W** プロシージャ *function-name* が返した結果セットが多すぎます。

**説明:** 示されているプロシージャが、CREATE PROCEDURE ステートメントで指定された数よりも多い結果セットを返しました。

**ユーザーの処置:** 返される結果セットが少なくなるようにプロシージャを変更するか、このプロシージャをドロップして再作成し、適切な数の結果セットを指定してください。

**sqlcode:** +20206

**sqlstate:** 0100E

---

**SQL20207N** *jar-id* の jar インストールまたは除去プロシージャが、デプロイメント記述子の使用を指定しました。

**説明:** jar インストールまたは置換プロシージャの DEPLOY または UNDEPLOY パラメーターが非ゼロでした。このパラメーターはサポートされていないので、ゼロにする必要があります。

**ユーザーの処置:** DEPLOY または UNDEPLOY パラメーターをゼロに設定して、プロシージャを再実行してください。

**sqlcode:** -20207

**sqlstate:** 46501

---

**SQL20208N** 表 *table-name* を作成できません。  
理由コード = *reason-code*

**説明:** 以下の理由コードによって示されている制限のため、表を作成できません。

1. ステージング表の定義に使用される表が、REFRESH DEFERRED オプションを指定したマテリアライズ照会表ではない。
2. ステージング表の定義に使用される表には、すでに関連するステージング表がある。

ステートメントは処理できません。

**ユーザーの処置:** 理由コードに対応するアクションは、以下のとおりです。

1. ステージング表を定義するには、REFRESHED DEFERRED オプションを使ってマテリアライズ照会表を指定する。
2. ステージング表と関連しないマテリアライズ照会表を指定する。

**sqlcode:**-20208

**sqlstate:**428FG

---

**SQL20209N** *option-type* オプションは、表 *table-name* には無効です。理由コード *reason-code*

**説明:** 以下の理由で、指定されたオプションは無効です。

1. チェック・ペンディング、no access 状態の表で READ ACCESS オプションを指定できない。
2. 表にチェックされていない健全性タイプがある場合は、FULL ACCESS オプションは無効である。
3. データなし移行モードにない表で FULL ACCESS オプションは無効である。

4. 表がデータなし移行モードでない場合は、FULL ACCESS オプションを IMMEDIATE UNCHECKED 文節とともに指定することはできない。
5. ステージング表でない表で PRUNE オプションは無効である。
6. PRUNE および INCREMENTAL オプションを同時に指定することはできない。

**ユーザーの処置:**

1. READ ACCESS オプションを指定しない。
2. FULL ACCESS オプションを SET INTEGRITY ... IMMEDIATE CHECKED ステートメントとともに使用するか、あるいは SET INTEGRITY ... IMMEDIATE UNCHECKED ステートメントの健全性オプションとして ALL を指定する。
3. FULL ACCESS オプションを IMMEDIATE CHECKED オプションとともに使用する。
4. FULL ACCESS オプションを IMMEDIATE UNCHECKED 文節なしで使用する。
5. ステージング表でない表をステートメントから除去する。
6. PRUNE または INCREMENTAL のいずれか 1 つを指定する。

**sqlcode:**-20209;

**sqlstate:**428FH;

---

**SQL20210N** ORDER OF *table-designator* が指定されました。この *table-designator* は ORDER BY 文節を含みません。

**説明:** *table-designator* の順序に基づいて配列するよう指定されましたが、*table-designator* には ORDER BY 文節は含まれていないので配列されません。

**ユーザーの処置:** *table-designator* の指定に ORDER BY 文節を追加するか、ORDER BY の代わりにソート・キー指定を使用してください。

**sqlcode:** -20210

**sqlstate:** 428FI

---

**SQL20211N ORDER BY または FETCH FIRST n ROWS ONLY の指定が無効です。**

**説明:** ORDER BY または FETCH FIRST n ROWS ONLY は以下においては使用できません。

- ビューの外部全選択
- SQL 表関数の RETURN ステートメントにおける外部全選択
- マテリアライズ照会表の定義
- 括弧で囲まれていない副選択

**ユーザーの処置:** 以下のケースでは、以下のよう  
にしてください。

**subselect**

ORDER BY または FETCH FIRST n ROWS ONLY を含む副選択を括弧で囲んでください。

**FETCH FIRST n ROWS ONLY**

where 文節の述部とともに ROW\_NUMBER() OVER() 文節を使用してください。例:

```
SELECT name FROM
 (SELECT
 ROW_NUMBER() OVER() AS m, name
 FROM emp
) AS e
WHERE m < 10
```

**ORDER BY**

ビュー、マテリアライズ照会表、または SQL 表関数を代わりに使って、照会で ORDER BY を使用してください。

**sqlcode:** -20211

**sqlstate:** 428FJ

---

**SQL20212N ユーザー定義ルーチン**

*function-name* が Java クラス *class-name* を JAR *JAR-name* からロードしようとして例外を検出しました。オリジナル例外 *underlying-exception*。

**説明:** ClassNotFoundException が発生しました。ClassNotFoundException は、クラスが見つからなかった原因をさらに詳しく説明するオリジナルの Java 例外を参照している可能性があります。たとえば、基となる例外は、JAR ファイルの読み取り試行中に発生した入出力エラーか、またはデータ・ディクショナリーから JAR を読み取り中に発生した SQL エラーである可能性があります。以下のトークンのうち、SQLCA 制限内に収まるだけのトークンが戻されます。

*function-name* は、実行により

ClassNotFoundException を検出した外部 Java 関数またはプロシーチャーの特定名を識別します。

*class-name* は、定義の見つからない Java クラスを識別します。

*JAR-name* は、識別されたクラスを含むことを予期されている、インストール済み JAR を任意で識別します。外部 Java 関数またはプロシーチャーがインストール済み JAR に含まれるよう定義されていないかぎり、(none) が指定されます。

*underlying-exception* には、この

ClassNotFoundException の結果として発生した、基となる例外の toString() が任意で含まれます。基となる例外がない場合は、(none) が指定されず。

**ユーザーの処置:** ALTER PROCEDURE を実行して、クラスの場所を指定したり、指定された JAR またはシステム・クラスパスにクラスがあることを確認したり、*underlying-exception* によって報告された条件を訂正します。

**sqlcode:** -20212

**sqlstate:** 46103

---

**SQL20214N ORDER OF *table-designator* が指定されましたが、この表指定子は配列されませんでした。**

**説明:** ORDER OF 文節は、*table-designator* の順序に基づいて結果表の配列を指定しましたが、配列は実行されませんでした。 *table-designator* に関連する照会に ORDER BY 文節がありません。または、ORDER BY 文節のソート・キーが定数式です。

**ユーザーの処置:** *table-designator* に関連する照会に有効な ORDER BY 文節を追加するか、ORDER OF 文節を使用する代わりにソート・キー指定を使用してください。

**sqlcode:** -20214

**sqlstate:** 428FI

---

**SQL20223N ENCRYPT または DECRYPT 関数が失敗しました。暗号化機能は使用できません。**

**説明:** 暗号化または暗号化解除要求を実行するための暗号化機能が使用できません。

**ユーザーの処置:** 暗号化関数がインストールされていない場合は、ENCRYPT または DECRYPT 関数を使用する前にインストールしてください。暗号化機能がインストールされている場合は、それが正しく機能していることを確認してください。

**sqlcode:** -20223

**sqlstate:** 560BF

---

**SQL20225W バッファ・プール操作 (DROP) は、バッファ・プールが使用中のため、次回データベースが再始動されるまで有効になりません。**

**説明:** DROP BUFFERPOOL ステートメントが発行されて、正常に完了しましたが、表スペースがこのバッファ・プールをまだ使用しています。表スペースを他のバッファ・プールに再割り当

てることはできますが、データベースを再始動するまでは、表スペースの再割り当ては有効になりません。この操作を確定する前に問題の表スペースがドロップされると、その時点で、メモリーからバッファ・プールが削除されます。このバッファ・プールは、次にデータベースを再始動するまでメモリー内に残るので、その他のバッファ・プールを作成または変更を行うと、そのことでこのバッファ・プールが再利用される場合は、その作成または変更操作は据え置かれます。

**ユーザーの処置:** メモリーからバッファ・プールを削除するには、データベースを再始動してください。

**sqlcode:** 20225

**sqlstate:** 01657

---

**SQL20230N プロシージャ名は、CALL ステートメントのホスト変数によって指定されない場合があります、引き数は USING DESCRIPTOR 文節で使用されない場合があります。**

**説明:** プロシージャ名は CALL ステートメントで ID として指定され、引き数は明示的に指定されなければなりません。ホスト変数はプロシージャ名には使用されない可能性があります。USING DESCRIPTOR 文節は、引き数の指定には使用されません。

**ユーザーの処置:** CALL ステートメントは、プロシージャ名の ID を提供し、引き数を明示的にリストするために再書き込みされる必要があります。プロシージャ名または引き数のいずれかが実行時にわからない場合、動的に準備された CALL ステートメントを使用します。

V8 以前のリリースから移行されたアプリケーションでは、アプリケーションが動的に準備された CALL ステートメントを使用するために再書き込みされるまで、CALL\_RESOLUTION DEFERRED プリコンパイル・オプションを使用することができます。

**sqlcode:** -20230



## SQLSTATE メッセージ

このセクションには、SQLSTATE とその意味がリストされています。SQLSTATE はクラス・コードによってグループ化されており、サブコードについては、対応する表をご覧ください。

表 275. SQLSTATE クラス・コード

| クラス・コード | 意味                      | サブコードについては、「...」を参照してください。 |
|---------|-------------------------|----------------------------|
| 00      | 無条件正常終了                 | 1916 ページの表 276             |
| 01      | 警告                      | 1916 ページの表 277             |
| 02      | データなし                   | 1920 ページの表 278             |
| 07      | 動的 SQL エラー              | 1920 ページの表 279             |
| 08      | 接続例外                    | 1921 ページの表 280             |
| 09      | トリガー・アクション例外            | 1921 ページの表 281             |
| 0A      | 機能がサポートされていない           | 1921 ページの表 282             |
| 0D      | ターゲット・タイプ指定が無効          | 1922 ページの表 283             |
| 0F      | トークンが無効                 | 1922 ページの表 284             |
| 0K      | RESIGNAL ステートメントが無効     | 1922 ページの表 285             |
| 20      | CASE ステートメントにケースが見つからない | 1922 ページの表 286             |
| 21      | カーディナリティー違反             | 1923 ページの表 287             |
| 22      | データ例外                   | 1923 ページの表 288             |
| 23      | 制約違反                    | 1924 ページの表 289             |
| 24      | カーソル状態が無効               | 1925 ページの表 290             |
| 25      | トランザクション状態が無効           | 1926 ページの表 291             |
| 26      | SQL ステートメント ID が無効      | 1926 ページの表 292             |
| 28      | 許可指定が無効                 | 1926 ページの表 293             |
| 2D      | トランザクション終了が無効           | 1926 ページの表 294             |
| 2E      | 接続名が無効                  | 1927 ページの表 295             |
| 34      | カーソル名が無効                | 1927 ページの表 296             |
| 36      | カーソル感度例外                | 1927 ページの表 297             |
| 38      | 外部関数例外                  | 1927 ページの表 298             |
| 39      | 外部関数呼び出し例外              | 1929 ページの表 299             |
| 3B      | SAVEPOINT が無効           | 1929 ページの表 300             |
| 40      | トランザクションのロールバック         | 1930 ページの表 301             |

表 275. *SQLSTATE* クラス・コード (続き)

| クラス・コード | 意味                    | サブコードについては、「...」を参照してください。 |
|---------|-----------------------|----------------------------|
| 42      | 構文エラーまたはアクセス規則違反      | 1930 ページの表 302             |
| 44      | WITH CHECK OPTION 違反  | 1943 ページの表 303             |
| 46      | Java DDL              | 1943 ページの表 304             |
| 51      | アプリケーション状態が無効         | 1943 ページの表 305             |
| 53      | 無効なオペランドまたは矛盾する指定     | 1945 ページの表 306             |
| 54      | SQL または製品の限界を超過       | 1945 ページの表 307             |
| 55      | オブジェクトが前提条件の状態にない     | 1946 ページの表 308             |
| 56      | その他の SQL または製品エラー     | 1947 ページの表 309             |
| 57      | リソースが使用不能またはオペレーターの介入 | 1949 ページの表 310             |
| 58      | システム・エラー              | 1950 ページの表 311             |

### クラス・コード 00 無条件正常終了

表 276. クラス・コード 00: 無条件正常終了

| <i>SQLSTATE</i> 値 | 意味                                |
|-------------------|-----------------------------------|
| 00000             | 操作は正常に実行されました。警告または例外状態は発生していません。 |

### クラス・コード 01 警告

表 277. クラス・コード 01: 警告

| <i>SQLSTATE</i> 値 | 意味                                      |
|-------------------|-----------------------------------------|
| 01002             | DISCONNECT エラーが起きました。                   |
| 01003             | NULL 値が、列関数の引き数から除去されました。               |
| 01004             | ストリングの値が、ホスト変数に割り当てられたときに切り捨てられました。     |
| 01005             | SQLDA の項目数が不足しています。                     |
| 01007             | 特権が付与されていません。                           |
| 0100C             | 1 つ以上の adhoc の結果セットが、プロシージャから返されました。    |
| 0100D             | クローズしていたカーソルが、チェーン内の次の結果セットで再度オープンしました。 |



表 277. クラス・コード 01: 警告 (続き)

| SQLSTATE 値 | 意味                                                         |
|------------|------------------------------------------------------------|
| 0100E      | プロシージャは結果セットを、最大許容数を超えて生成しました。最初の整数結果セットのみが呼び出し元に戻されました。   |
| 01503      | 結果列の数が、指定されたホスト変数の数よりも大きくなっています。                           |
| 01504      | UPDATE または DELETE ステートメントに、WHERE 文節がありません。                 |
| 01506      | 算術演算の結果である無効な日付を訂正するため、DATE または TIMESTAMP の値が調整されました。      |
| 01509      | ユーザーの仮想計算機に十分なストレージがないため、カーソルについてはブロッキングが取り消されました。         |
| 01515      | 列の非 null 値がホスト変数の範囲外にあるため、null 値がホスト変数に割り当てられました。          |
| 01516      | 不適当な WITH GRANT OPTION が無視されました。                           |
| 01517      | 変換できない文字を、置換文字で置き換えました。                                    |
| 01519      | 数値が範囲外であるため、null 値がホスト変数に割り当てられました。                        |
| 01524      | 列関数の結果には、算術式を評価することで発生した null 値は含まれません。                    |
| 01526      | 分離レベルが自動調整されました。                                           |
| 01539      | 接続は成功しましたが、SBCS 文字のみが使用できます。                               |
| 01543      | 重複した制約が無視されました。                                            |
| 01545      | 修飾されていない列名が、相関参照として解釈されました。                                |
| 01550      | 指定された記述を持つ索引がすでに存在しているため、索引が作成されませんでした。                    |
| 01560      | 冗長 GRANT は無視されます。                                          |
| 01562      | データベース構成ファイル内のログへの新たなパス (newlogpath) が無効です。                |
| 01563      | ログ・ファイルへの現在のパス (logpath) が無効です。ログ・ファイル・パスはデフォルトにリセットされました。 |
| 01564      | ゼロで割り算を行ったため、null 値がホスト変数に割り当てられました。                       |
| 01586      | 参照構造の親表についての制約を OFF に設定したため、1 つ以上の下位表が自動的にチェック保留状態に置かれました。 |
| 01589      | ステートメントに余分な指定があります。                                        |

表 277. クラス・コード 01: 警告 (続き)

| SQLSTATE 値 | 意味                                                                                                                                                                                                                                                                                                                                                                                         |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 01592      | <p>SOURCE 関数を参照する CREATE FUNCTION ステートメントが、以下のいずれかの状態になっています。</p> <ul style="list-style-type: none"> <li>• 入力パラメーターの長さ、精度、または位取りが、対応するソース関数のそれよりも大きい。</li> <li>• RETURNS または CAST FROM パラメーターの長さ、精度、または位取りが、ソース関数のそれよりも小さい。</li> <li>• CREATE FUNCTION ステートメントの CAST FROM パラメーターの長さ、精度、または位取りが、RETURNS パラメーターのそれよりも大きい。</li> </ul> <p>実行時に切り捨てが実行される可能性があります (また、実行時にエラーが起きる可能性があります)。</p> |
| 01594      | ALL 情報のための SQLDA の項目数が不足しています (明確に区別された名前を返すために十分な記述子がありません)。                                                                                                                                                                                                                                                                                                                              |
| 01595      | ビューが、既存の無効にされたビューと置き換えられました。                                                                                                                                                                                                                                                                                                                                                               |
| 01596      | 長ストリング・データ・タイプに基づいた異なるタイプに対して、比較関数が作成されませんでした。                                                                                                                                                                                                                                                                                                                                             |
| 01598      | アクティブなイベント・モニターを活動化しようとしたか、または非アクティブのイベント・モニターを非活動化しようとした。                                                                                                                                                                                                                                                                                                                                 |
| 01599      | BIND オプションが REBIND で無視されました。                                                                                                                                                                                                                                                                                                                                                               |
| 01602      | 最適化レベルが低下しました。                                                                                                                                                                                                                                                                                                                                                                             |
| 01603      | CHECK DATA 処理が制約違反を見つけ、それらを例外表に移動しました。                                                                                                                                                                                                                                                                                                                                                     |
| 01604      | SQL ステートメントが解釈されましたが、実行されませんでした。                                                                                                                                                                                                                                                                                                                                                           |
| 01605      | 再帰共通表式に無限ループが入っている可能性があります。                                                                                                                                                                                                                                                                                                                                                                |
| 01606      | ノードまたはシステム・データベース・ディレクトリーが空です。                                                                                                                                                                                                                                                                                                                                                             |
| 01607      | 読み取り専用トランザクションのノードの時間の間の違いが、定義されているしきい値を超えました。                                                                                                                                                                                                                                                                                                                                             |
| 01609      | プロシージャは結果セットを、最大許容数を超えて生成しました。最初の整数結果セットのみが呼び出し元に戻されました。                                                                                                                                                                                                                                                                                                                                   |
| 01610      | 1 つ以上の ad hoc の結果セットが、プロシージャから返されました。                                                                                                                                                                                                                                                                                                                                                      |
| 01611      | クローズしていたカーソルが、チェーン内の次の結果セットで再度オープンしました。                                                                                                                                                                                                                                                                                                                                                    |
| 01616      | 見積もり CPU コストがリソースの限度を超過しています。                                                                                                                                                                                                                                                                                                                                                              |
| 01618      | データの区分化を変更するには、ノード・グループの再分散が必要です。                                                                                                                                                                                                                                                                                                                                                          |

表 277. クラス・コード 01: 警告 (続き)

| SQLSTATE 値 | 意味                                                                   |
|------------|----------------------------------------------------------------------|
| 01620      | UNION ALL の基本表の一部は、同一の表です。                                           |
| 01621      | 検索した LOB の値は変更されています。                                                |
| 01622      | ステートメントは正常に完了しましたが、その後でシステム・エラーが発生しました。                              |
| 01623      | DEGREE の値は無視されます。                                                    |
| 01626      | データベースにアクティブ・バッファ・プールは 1 つだけです。                                      |
| 01627      | 表が調整ペンディングまたは調整不能状態のため、 DATALINK 値は無効である可能性があります。                    |
| 01632      | 同時接続の数が製品の公称定義の数を超過しました。                                             |
| 01633      | マテリアライズ照会表は、照会の処理を最適化するために使用することはできません。                              |
| 01636      | 非増分データの保全本性はデータベース・マネージャーによる確認がされないままになっています。                        |
| 01637      | デバッグは使用できません。                                                        |
| 01639      | 連合オブジェクトでは、呼び出し側がデータ・ソース・オブジェクトに対して必要な特権を持っていることが必要です。               |
| 01641      | データ・リンク・タイプ属性が、構造化タイプの使用を制限しています。                                    |
| 01642      | 列の長さは、許可されている USER デフォルト値の最大長のために十分ではありません。                          |
| 01643      | SQL ルーチンで SQLSTATE または SQLCODE 変数への割り当てが上書きされたと思われるため、ハンドラーを活動化しません。 |
| 01645      | SQL プロシーチャーの実行可能プログラムはデータベース・カタログに保管されません。                           |
| 01648      | VALUE COMPRESSION が表に対して非活動状態であるため、COMPRESS 列属性は無視されました。             |
| 01649      | バッファ・プール構成は完了しましたが、次にデータベースが再始動されるまでは有効になりません。                       |
| 01650      | 索引と表の統計が矛盾しています。                                                     |
| 01651      | イベント・モニターは正常に活動化されましたが、いくつかのモニター情報が脱落した可能性があります。                     |
| 01652      | ステートメント・コンテキストのために、分離文節は無視されます。                                      |
| 01653      | 許可はユーザーに対して認可されました。グループは、許可名が 8 バイトより長いために考慮に入れられませんでした。             |
| 01654      | バッファ・プールが開始されていません。                                                  |

表 277. クラス・コード 01: 警告 (続き)

| SQLSTATE 値 | 意味                                                           |
|------------|--------------------------------------------------------------|
| 01655      | イベント・モニターが正常に作成されましたが、少なくとも 1 つのイベント・モニター・ターゲット表がすでに存在しています。 |
| 01657      | メモリー不足のため、バッファ・プール操作は、次回データベースの始動まで有効になりません。                 |
| 01H51      | MQSeries アプリケーション・メッセージング・インターフェース・メッセージが切り捨てられました。          |
| 01HXX      | 有効な警告 SQLSTATE が、ユーザー定義関数または外部プロシージャ・CALL によって返されました。        |

## クラス・コード 02 データなし

表 278. クラス・コード 02: データなし

| SQLSTATE 値 | 意味                                                                                                                                                                                                                                             |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 02000      | 以下のいずれかの例外が起きました。 <ul style="list-style-type: none"> <li>SELECT INTO ステートメントの結果、または INSERT ステートメントの副選択の結果が、データのない表になった。</li> <li>探索型の UPDATE または DELETE ステートメントで指定された行数がゼロになった。</li> <li>FETCH ステートメントで参照されたカーソルの位置が、結果表の最終行の後になった。</li> </ul> |
| 02501      | カーソル位置が現在行の FETCH に対して無効です。                                                                                                                                                                                                                    |
| 02502      | 削除または更新で欠陥が検出された。                                                                                                                                                                                                                              |

## クラス・コード 07 動的 SQL エラー

表 279. クラス・コード 07: 動的 SQL エラー

| SQLSTATE 値 | 意味                                                                 |
|------------|--------------------------------------------------------------------|
| 07001      | ホスト変数の数がパラメーター・マーカースの数として正しくありません。                                 |
| 07002      | 呼び出しパラメーター・リストまたは制御ブロックが無効です。                                      |
| 07003      | EXECUTE ステートメントで識別されたステートメントが、select ステートメントであるか、または準備された状態にありません。 |
| 07004      | 動的パラメーターに USING 文節または INTO 文節が必要です。                                |

表 279. クラス・コード 07: 動的 SQL エラー (続き)

| SQLSTATE 値 | 意味                                               |
|------------|--------------------------------------------------|
| 07005      | カーソルのステートメント名が、カーソルと関連付けられない準備されたステートメントを識別しました。 |
| 07006      | データ・タイプが適切でないため、入力ホスト変数を使用できません。                 |

## クラス・コード 08 接続例外

表 280. クラス・コード 08: 接続例外

| SQLSTATE 値 | 意味                                                                                                 |
|------------|----------------------------------------------------------------------------------------------------|
| 08001      | アプリケーション・リクエスターが接続を確立できません。                                                                        |
| 08002      | 接続がすでに存在します。                                                                                       |
| 08003      | 接続が存在しません。                                                                                         |
| 08004      | アプリケーション・サーバーが、接続の確立を拒否しました。                                                                       |
| 08007      | 不明なトランザクション解像度です。                                                                                  |
| 08502      | トランザクション・マネージャーが使用できないため、TWOPHASE の SYNCPOINT を使用して実行中のアプリケーション処理によって発行された CONNECT ステートメントが失敗しました。 |
| 08504      | 指定されたパスの名前変更構成ファイルの処理中に、エラーが検出されました。                                                               |

## クラス・コード 09 トリガー・アクション

表 281. クラス・コード 09: トリガー・アクション

| SQLSTATE 値 | 意味                       |
|------------|--------------------------|
| 09000      | トリガー SQL ステートメントが失敗しました。 |

## クラス・コード 0A サポートされていない機能

表 282. クラス・コード 0A: サポートされていない機能

| SQLSTATE 値 | 意味                                   |
|------------|--------------------------------------|
| 0A001      | 処理が接続可能状態にないため、CONNECT ステートメントは無効です。 |
| 0A502      | このデータベース・インスタンスではアクションまたは操作ができません。   |

表 282. クラス・コード 0A: サポートされていない機能 (続き)

| SQLSTATE | 意味                                           |
|----------|----------------------------------------------|
| 0A503    | データの不整合の可能性があるので、連合挿入、更新、または削除操作はコンパイルできません。 |

### クラス・コード 0D ターゲット・タイプ指定が無効

表 283. クラス・コード 0D: ターゲット・タイプ指定が無効

| SQLSTATE | 意味                                           |
|----------|----------------------------------------------|
| 0D000    | ターゲット構造化データ・タイプ指定は、ソース構造化データ・タイプの正しいサブタイプです。 |

### クラス・コード 0F 無効なトークン

表 284. クラス・コード 0F: 無効なトークン

| SQLSTATE | 意味                         |
|----------|----------------------------|
| 0F001    | LOB トークン変数は、現在何も値を表していません。 |

### クラス・コード 0K RESIGNAL ステートメントが無効

表 285. クラス・コード 0K: RESIGNAL ステートメントが無効

| SQLSTATE | 意味                             |
|----------|--------------------------------|
| 0K000    | RESIGNAL ステートメントがハンドラー内にありません。 |

### クラス・コード 20 CASE ステートメントにケースが見つからない

表 286. クラス・コード 20: CASE ステートメントにケースが見つからない

| SQLSTATE | 意味                            |
|----------|-------------------------------|
| 20000    | CASE ステートメント用のケースが見つかりませんでした。 |

## クラス・コード 21 カーディナリティー違反

表 287. クラス・コード 21: カーディナリティー違反

| SQLSTATE 値 | 意味                                                       |
|------------|----------------------------------------------------------|
| 21000      | SELECT INTO の結果が複数行の結果表になったか、または基本述部の副照会の結果が複数の値になっています。 |
| 21501      | 自己参照表への複数行の INSERT は無効です。                                |
| 21502      | 主キーの複数行 UPDATE は無効です。                                    |
| 21504      | RESTRICT または SET NULL の削除規則を持つ自己参照表からの複数行 DELETE は無効です。  |
| 21505      | 行関数は 1 行のみを戻さなければなりません。                                  |

## クラス・コード 22 データ例外

表 288. クラス・コード 22: データ例外

| SQLSTATE 値 | 意味                                                                                                   |
|------------|------------------------------------------------------------------------------------------------------|
| 22001      | 文字データの右側が切り捨てられました。たとえば、更新または挿入の値が、列には長すぎるストリングである、またはホスト変数が小さすぎるため、日時の値をそのホスト変数に割り当てられない、などが考えられます。 |
| 22002      | 値が NULL、または標識パラメーターがないことが見つかりました。たとえば、標識変数が指定されていないため、NULL 値がホスト変数に割り当てられない、などが考えられます。               |
| 22003      | 数値が範囲を超えています。                                                                                        |
| 22004      | PARAMETER STYLE GENERAL と定義されているプロシージャから、または NULL ではない引き数で呼び出されているタイプ保護方式から NULL 値を返すことはできません。       |
| 22007      | 無効な日時形式が検出されました。これは、無効なストリング表現または値が指定されたことが考えられます。                                                   |
| 22008      | 日時フィールドにオーバーフローが起きました。たとえば、日付またはタイム・スタンプの算術演算の結果が、有効な日付の範囲内でないことが考えられます。                             |
| 2200G      | 最も特定のタイプが一致しません。                                                                                     |
| 22011      | サブストリング・エラーが起きました。たとえば、SUBSTR の引き数が範囲内でないことが考えられます。                                                  |
| 22012      | 0 による除算は無効です。                                                                                        |

表 288. クラス・コード 22: データ例外 (続き)

| SQLSTATE 値 | 意味                                                                                 |
|------------|------------------------------------------------------------------------------------|
| 22018      | CAST、DECIMAL、FLOAT、または INTEGER スカラー関数については、文字の値は無効です。                              |
| 22019      | LIKE 述部に無効なエスケープ文字があります。                                                           |
| 22021      | 文字がコード化文字セットにありません。                                                                |
| 22024      | NUL で終了する入力ホスト変数またはパラメーターに、NUL があります。                                              |
| 22025      | LIKE 述部文字列・パターンに、無効なエスケープ文字のオカレンスがあります。                                            |
| 2202D      | NULL インスタンスは mutator 方式で使用されます。                                                    |
| 22501      | 可変長文字列の長さコントロール・フィールドが、負の値になっているか、または最大値を超えています。                                   |
| 22504      | 混合データの値が無効です。                                                                      |
| 22506      | TOD クロックが誤動作しているか、またはオペレーティング・システムの timezone パラメーターが範囲外であるため、日時特殊レジスターに対する参照が無効です。 |
| 22522      | CCSID の値が、完全に無効であるか、データ・タイプまたはサブタイプに対して無効であるか、またはエンコード・スキーマに対して無効です。               |
| 22526      | キー・トランスフォーム関数が行を生成しなかったか、または重複する行を生成しました。                                          |
| 22527      | 複数行の INSERT 操作に対して無効な入力データが検出されました。                                                |

## クラス・コード 23 制約違反

表 289. クラス・コード 23: 制約違反

| SQLSTATE 値 | 意味                                             |
|------------|------------------------------------------------|
| 23001      | 親キーの更新または削除が、RESTRICT 更新または削除の規則によって妨げられています。  |
| 23502      | 挿入または更新の値が NULL ですが、列に NULL 値を入れることはできません。     |
| 23503      | 外部キーの挿入または更新の値が無効です。                           |
| 23504      | 親キーの更新または削除が、NO ACTION 更新または削除の規則によって妨げられています。 |



表 289. クラス・コード 23: 制約違反 (続き)

| SQLSTATE 値 | 意味                                                          |
|------------|-------------------------------------------------------------|
| 23505      | ユニーク索引またはユニーク制約で定められている制約に対する違反が起きました。                      |
| 23510      | RLST 表によるコマンド使用時の制約違反が発生しました。                               |
| 23511      | チェック制約が削除を制限しているため、親行を削除できません。                              |
| 23512      | 表に制約定義を満たしていない行があるため、チェック制約を追加できません。                        |
| 23513      | INSERT または UPDATE の結果の行が、チェック制約定義に合いません。                    |
| 23514      | データ・チェック処理が制約違反を見つけました。                                     |
| 23515      | 表にある指定されたキーの値が重複しているため、ユニーク索引を作成できないか、またはユニーク制約を追加できませんでした。 |
| 23520      | 外部キーの値が、親表の親キーとすべて等しくないため、外部キーを定義できません。                     |
| 23521      | カタログ表の更新が、内部制約に違反します。                                       |
| 23522      | ID 列またはシーケンスの値の範囲を使い果たしました。                                 |

## クラス・コード 24 無効なカーソル状態

表 290. クラス・コード 24: 無効なカーソル状態

| SQLSTATE 値 | 意味                                                           |
|------------|--------------------------------------------------------------|
| 24501      | 識別されたカーソルがオープンしていません。                                        |
| 24502      | OPEN ステートメントで識別されたカーソルが、すでにオープンしていません。                       |
| 24504      | UPDATE、DELETE、SET、または GET ステートメントで識別されたカーソルが、行に位置付けられていません。  |
| 24506      | PREPARE で識別されたステートメントは、オープン・カーソルのステートメントです。                  |
| 24510      | UPDATE または DELETE 操作が削除ホールまたは更新ホールに対して試行されました。               |
| 24512      | 結果表が基本表と一致しません。                                              |
| 24513      | カーソル位置が不明のため、FETCH NEXT、PRIOR、CURRENT、または RELATIVE は許可されません。 |
| 24514      | 以前のエラーによって、このエラーを使用できません。                                    |

表 290. クラス・コード 24: 無効なカーソル状態 (続き)

| SQLSTATE | 意味                                  |
|----------|-------------------------------------|
| 値        |                                     |
| 24516    | カーソルが結果セットにすでに割り当てられています。           |
| 24517    | カーソルが、外部関数または方式によりオープンされたままになっています。 |

### クラス・コード 25 無効なトランザクション状態

表 291. クラス・コード 25: 無効なトランザクション状態

| SQLSTATE | 意味                                   |
|----------|--------------------------------------|
| 値        |                                      |
| 25000    | 更新操作が、アプリケーション実行環境では無効です。            |
| 25001    | ステートメントは、作業単位の最初のステートメントとしてのみ許可されます。 |
| 25501    | ステートメントは、作業単位の最初のステートメントとしてのみ許可されます。 |

### クラス・コード 26 無効な SQL ステートメント ID

表 292. クラス・コード 26: 無効な SQL ステートメント ID

| SQLSTATE | 意味                   |
|----------|----------------------|
| 値        |                      |
| 26501    | 識別されたステートメントが存在しません。 |

### クラス・コード 28 無効な許可指定

表 293. クラス・コード 28: 無効な許可指定

| SQLSTATE | 意味        |
|----------|-----------|
| 値        |           |
| 28000    | 許可名が無効です。 |

### クラス・コード 2D 無効なトランザクション終了

表 294. クラス・コード 2D: 無効なトランザクション終了

| SQLSTATE | 意味                                       |
|----------|------------------------------------------|
| 値        |                                          |
| 2D521    | SQL COMMIT または ROLLBACK が、現在の操作環境では無効です。 |

表 294. クラス・コード 2D: 無効なトランザクション終了 (続き)

| SQLSTATE | 意味                                             |
|----------|------------------------------------------------|
| 2D522    | COMMIT と ROLLBACK は、ATOMIC 複合ステートメントでは許可されません。 |
| 2D528    | 動的 COMMIT が、アプリケーション実行環境では無効です。                |
| 2D529    | 動的 ROLLBACK が、アプリケーション実行環境では無効です。              |

### クラス・コード 2E 無効な接続

表 295. クラス・コード 2E: 無効な接続名

| SQLSTATE | 意味        |
|----------|-----------|
| 2E000    | 接続名が無効です。 |

### クラス・コード 34 無効なカーソル名

表 296. クラス・コード 34: 無効なカーソル名

| SQLSTATE | 意味          |
|----------|-------------|
| 34000    | カーソル名が無効です。 |

### クラス・コード 36 無効なカーソル指定

表 297. クラス・コード 36: 無効なカーソル指定

| SQLSTATE | 意味                                            |
|----------|-----------------------------------------------|
| 36001    | SENSITIVE カーソルは、指定した select ステートメントには定義できません。 |

### クラス・コード 38 外部関数例外

表 298. クラス・コード 38: 外部関数例外

| SQLSTATE | 意味                                        |
|----------|-------------------------------------------|
| 38XXX    | 外部ルーチンまたはトリガーにより、有効なエラー SQLSTATE が戻されました。 |
| 38001    | 外部ルーチンは、SQL ステートメントの実行を許可されません。           |

表 298. クラス・コード 38: 外部関数例外 (続き)

| SQLSTATE 値 | 意味                                                                                                        |
|------------|-----------------------------------------------------------------------------------------------------------|
| 38002      | ルーチンがデータを変更しようとしたが、このルーチンは <b>MODIFIES SQL DATA</b> (SQL データの変更) として定義されていません。                            |
| 38003      | ステートメントはルーチンでは許可されていません。                                                                                  |
| 38004      | ルーチンがデータの読み取りを試みましたが、このルーチンは <b>READS SQL DATA</b> (SQL データの読み取り) として定義されていませんでした。                        |
| 38501      | ユーザー定義の関数、外部プロシージャ、またはトリガー ( <b>SIMPLE CALL</b> または <b>SIMPLE CALL WITH NULLS</b> を使用) の呼び出し中にエラーが発生しました。 |
| 38502      | 外部関数は、SQL ステートメントの実行を許可されていません。                                                                           |
| 38503      | ユーザー定義関数が異常終了しました ( <b>ABEND</b> )。                                                                       |
| 38504      | ユーザーがループ状態を停止するために、ユーザーによってユーザー定義関数が中断されました。                                                              |
| 38505      | SQL ステートメントは、 <b>FINAL CALL</b> のルーチンでは許可されません。                                                           |
| 38506      | <b>OLE DB Provider</b> で、関数がエラーを起こして失敗しました。                                                               |
| 38552      | <b>SYSPFUN</b> スキーマの関数 ( <b>IBM</b> 提供) が異常終了しました。                                                        |
|            | メッセージ・テキストで以下の理由コードの 1 つを検出することができます。                                                                     |
| <b>01</b>  | 数値が範囲外                                                                                                    |
| <b>02</b>  | ゼロによる除算                                                                                                   |
| <b>03</b>  | 算術オーバーフローまたはアンダーフロー                                                                                       |
| <b>04</b>  | 無効なデータ形式                                                                                                  |
| <b>05</b>  | 無効な時刻形式                                                                                                   |
| <b>06</b>  | 無効なタイム・スタンプ形式                                                                                             |
| <b>07</b>  | タイム・スタンプ期間の無効な文字表示                                                                                        |
| <b>08</b>  | 無効なインターバル・タイプ (1、 2、 4、 8、 16、 32、 64、 128、 256 のいずれかでなければならない。)                                          |
| <b>09</b>  | ストリングが長すぎる                                                                                                |
| <b>10</b>  | ストリング関数の長さまたは位置が範囲外になっている                                                                                 |
| <b>11</b>  | 浮動小数点数では無効な文字表示である                                                                                        |
| 38553      | システム・スキーマ内のルーチンがエラーにより終了しました。                                                                             |
| 38H01      | <b>MQSeries</b> 関数が初期化に失敗しました。                                                                            |
| 38H02      | <b>MQSeries</b> アプリケーション・メッセージング・インターフェースがセッションを終了できませんでした。                                               |
| 38H03      | <b>MQSeries</b> アプリケーション・メッセージング・インターフェースがメッセージを正常に処理できませんでした。                                            |

表 298. クラス・コード 38: 外部関数例外 (続き)

| SQLSTATE | 意味                                                                    |
|----------|-----------------------------------------------------------------------|
| 38H04    | MQSeries アプリケーション・メッセージング・インターフェースがメッセージを送信できませんでした。                  |
| 38H05    | MQSeries アプリケーション・メッセージング・インターフェースがメッセージの読み取りまたは受信に失敗しました。            |
| 38H06    | MQSeries アプリケーション・メッセージング・インターフェース・サブスクリプション (サブスクリプション解除) 要求が失敗しました。 |

### クラス・コード 39 外部関数呼び出し例外

表 299. クラス・コード 39: 外部関数呼び出し例外

| SQLSTATE | 意味                                 |
|----------|------------------------------------|
| 39001    | ユーザー定義関数が無効な SQLSTATE を返しました。      |
| 39004    | IN または INOUT 引き数では NULL 値は許可されません。 |
| 39501    | 引き数に関連する目印が修正されました。                |

### クラス・コード 3B SAVEPOINT が無効

表 300. クラス・コード 3B: SAVEPOINT が無効

| SQLSTATE | 意味                                                                                        |
|----------|-------------------------------------------------------------------------------------------|
| 3B001    | 保管点が無効です。                                                                                 |
| 3B002    | 保管点が最大数に達しました。                                                                            |
| 3B501    | 重複する保管点名が削除されました。                                                                         |
| 3B502    | RELEASE または ROLLBACK TO SAVEPOINT が指定されましたが、保管点は存在しません。                                   |
| 3B503    | SAVEPOINT、RELEASE SAVEPOINT、または ROLLBACK TO SAVEPOINT は、トリガーまたはグローバル・トランザクションでは許可されていません。 |

## クラス・コード 40 トランザクション・ロールバック

表 301. クラス・コード 40: トランザクション・ロールバック

| SQLSTATE 値 | 意味                                      |
|------------|-----------------------------------------|
| 40001      | 自動ロールバックでデッドロックまたはタイムアウトが起きました。         |
| 40003      | ステートメント完了が不明です。                         |
| 40504      | システム・エラーのため、作業単位がロールバックされました。           |
| 40506      | 現在のトランザクションは、SQL エラーのためロールバックしました。      |
| 40507      | 現在のトランザクションは、索引を作成するのに失敗したためロールバックしました。 |

## クラス・コード 42 構文エラーまたはアクセス規則違反

表 302. クラス・コード 42: 構文エラーまたはアクセス規則違反

| SQLSTATE 値 | 意味                                           |
|------------|----------------------------------------------|
| 42501      | 許可 ID に、識別されたオブジェクトに対して指定された操作を実行する権限がありません。 |
| 42502      | 許可 ID に、指定された操作を実行する権限がありません。                |
| 42504      | 指定された許可名から、指定された権限を除去できません。                  |
| 42506      | 所有者の許可が失敗しました。                               |
| 42508      | 指定されたデータベース権限は PUBLIC にはできません。               |
| 42509      | SQL ステートメントは STATICRULES オプションのため許可されません。    |
| 42511      | DATALINK 値を検索できません。                          |
| 42601      | 文字、トークン、または文節が、無効もしくは欠けています。                 |
| 42602      | 名前に無効な文字が見つかりました。                            |
| 42603      | 未終了ストリング定数が見つかりました。                          |
| 42604      | 無効な数値またはストリング定数が見つかりました。                     |
| 42605      | スカラー関数に指定された引き数の数が無効です。                      |
| 42606      | 無効な 16 進定数が見つかりました。                          |
| 42607      | 列関数のオペランドが無効です。                              |
| 42608      | VALUES での NULL または DEFAULT の使用は無効です。         |
| 42609      | 演算子または述部のオペランドが、すべてパラメーター・マーカースです。           |
| 42610      | パラメーター・マーカースは許可されていません。                      |

表 302. クラス・コード 42: 構文エラーまたはアクセス規則違反 (続き)

| SQLSTATE 値 | 意味                                                       |
|------------|----------------------------------------------------------|
| 42611      | 列または引き数の定義が無効です。                                         |
| 42612      | ステートメント・ストリングが、示されているコンテキストでは受け入れられない SQL ステートメントです。     |
| 42613      | 文節が相互に排他的です。                                             |
| 42614      | 重複キーワードは無効です。                                            |
| 42615      | 無効な代替が見つかりました。                                           |
| 42617      | ステートメント・ストリングがブランクまたは空です。                                |
| 42618      | ホスト変数は許可されていません。                                         |
| 42620      | 読み取り専用 SCROLL が UPDATE 文節で指定されました。                       |
| 42621      | チェック制約が無効です。                                             |
| 42622      | 名前またはラベルが長すぎます。                                          |
| 42623      | DEFAULT 文節を指定できません。                                      |
| 42625      | CASE 式が無効です。                                             |
| 42627      | RETURNS 文節は EXPRESSION AS 文節を使用して、述部を指定する前に指定する必要があります。  |
| 42628      | 複数の TO SQL または FROM SQL トランスフォーム関数が、トランスフォーム定義に定義されています。 |
| 42629      | SQL ルーチンにパラメーター名を指定する必要があります。                            |
| 42630      | ネストされた複合ステートメントに SQLSTATE または SQLCODE 変数宣言を指定することはできません。 |
| 42631      | SQL 関数または方式内の RETURN ステートメントには、戻り値が入っていないなければなりません。      |
| 42701      | INSERT または UPDATE ステートメントで重複する列名が見つかりました。                |
| 42702      | 重複する名前があるため、列の参照が未確定です。                                  |
| 42703      | 未定義の列、属性、またはパラメーター名がありました。                               |
| 42704      | 未定義のオブジェクトまたは制約名が見つかりました。                                |
| 42705      | 未定義のサーバー名が見つかりました。                                       |
| 42707      | ORDER BY の列名が、結果表の列を識別していません。                            |
| 42709      | PRIMARY、UNIQUE、または FOREIGN KEY 文節で重複する列名が見つかりました。        |
| 42710      | 重複するオブジェクトまたは制約名が見つかりました。                                |

表 302. クラス・コード 42: 構文エラーまたはアクセス規則違反 (続き)

| SQLSTATE 値 | 意味                                                                          |
|------------|-----------------------------------------------------------------------------|
| 42711      | オブジェクト定義または ALTER ステートメントの中で、列名または属性名が重複していました。                             |
| 42712      | 重複する表指定が FROM 文節で見つかりました。                                                   |
| 42713      | オブジェクトのリストで、重複オブジェクトが検出されました。                                               |
| 42720      | リモート・データベースのノード名が、ノードのディレクトリーに見つかりませんでした。                                   |
| 42723      | 同じシグニチャーを持つ関数が、すでにスキーマに存在します。                                               |
| 42724      | ユーザー定義関数またはプロシージャに使用される外部プログラムにアクセスできません。                                   |
| 42725      | 関数または方式が (シグニチャーまたは特定のインスタンス名を使わずに、) 直接参照されましたが、その関数または方式の特定インスタンスが複数存在します。 |
| 42726      | 名前派生表に重複する名前が見つかりました。                                                       |
| 42727      | 新しい表に、デフォルト 1 次表スペースがありません。                                                 |
| 42728      | 重複するノードが、ノード・グループ定義に見つかりました。                                                |
| 42729      | ノードが定義されていません。                                                              |
| 42730      | コンテナ名が、別の表スペースによってすでに使用されています。                                              |
| 42731      | コンテナ名が、この表スペースによってすでに使用されています。                                              |
| 42732      | SET CURRENT PATH ステートメントで、重複スキーマ名が見つかりました。                                  |
| 42734      | 重複するパラメーター名、SQL 変数名、カーソル名、条件名、またはラベルが見つかりました。                               |
| 42735      | 表スペースのノード・グループがバッファー・プールに定義されていません。                                         |
| 42736      | LEAVE ステートメントに指定されているラベルが見つからないか、または無効です。                                   |
| 42737      | ハンドラーに指定されている条件が定義されていません。                                                  |
| 42738      | 重複する列名または名前のない列が、FOR ステートメントの DECLARE CURSOR ステートメントに指定されました。               |
| 42739      | 重複するトランスフォームが見つかりました。                                                       |
| 42740      | 指定されたタイプのトランスフォームが見つかりませんでした。ドロップされたトランスフォームはありません。                         |
| 42741      | トランスフォーム・グループがデータ・タイプに定義されていません。                                            |



表 302. クラス・コード 42: 構文エラーまたはアクセス規則違反 (続き)

| SQLSTATE 値 | 意味                                                                          |
|------------|-----------------------------------------------------------------------------|
| 42742      | 型付き表または型付きビュー階層の中に、同じタイプの副表またはサブビューがすでに存在しています。                             |
| 42743      | 索引拡張子の中に検索方式が見つかりません。                                                       |
| 42744      | TO SQL または FROM SQL トランスフォーム関数が、トランスフォーム・グループに定義されていません。                    |
| 42745      | ルーチンが、既存の方式とのオーバーライド・リレーションシップを定義しています。                                     |
| 42746      | 同じタイプ階層で、方式と構造化タイプを同じ名前にすることはできません。                                         |
| 42802      | 挿入値または更新値の数が、列の数と同じではありません。                                                 |
| 42803      | 列がグループ列ではないため SELECT または HAVING 文節での列の参照が無効であるか、または GROUP BY 文節での列の参照が無効です。 |
| 42804      | CASE 式の結果式に互換性がありません。                                                       |
| 42805      | ORDER BY 文節の整数が、結果表の列を識別していません。                                             |
| 42806      | データ・タイプに互換性がないため、ホスト変数に値を割り当てられません。                                         |
| 42807      | INSERT、UPDATE、または DELETE は、このオブジェクトでは許可されません。                               |
| 42808      | INSERT または UPDATE ステートメントで識別された列が更新できません。                                   |
| 42809      | 識別されたオブジェクトは、ステートメントが適用するタイプのオブジェクトではありません。                                 |
| 42810      | ビューが FOREIGN KEY 文節で識別されました。                                                |
| 42811      | 指定された列数が、SELECT 文節の列数と同じではありません。                                            |
| 42813      | 指定されたビューには、WITH CHECK OPTION を使用できません。                                      |
| 42815      | データ・タイプ、長さ、位取り、値、または CCSID が無効です。                                           |
| 42816      | 式の日時の値または期間が無効です。                                                           |
| 42818      | 演算子または関数のオペランドに互換性がありません。                                                   |
| 42819      | 算術演算のオペランド、または数値を必要とする関数のオペランドが数値ではありません。                                   |
| 42820      | 数値制約が長すぎるか、またはそのデータ・タイプの範囲内にない値を持っています。                                     |
| 42821      | 更新または挿入の値に、列との互換性がありません。                                                    |

表 302. クラス・コード 42: 構文エラーまたはアクセス規則違反 (続き)

| SQLSTATE 値 | 意味                                                                      |
|------------|-------------------------------------------------------------------------|
| 42823      | 1 つの列しか許可されていない副照会から複数の列が返されました。                                        |
| 42824      | LIKE のオペランドが文字列ではないか、または最初のオペランドが列ではありません。                              |
| 42825      | UNION、INTERSECT、EXCEPT、または VALUES の行に、互換性のある列がありません。                    |
| 42826      | UNION、INTERSECT、EXCEPT、または VALUES の行が、同じ数の列を持っていません。                    |
| 42827      | UPDATE または DELETE で識別された表が、カーソルによって指定された表と同じではありません。                    |
| 42828      | UPDATE または DELETE ステートメントのカーソルによって指定された表を修正できないか、またはカーソルが読み取り専用です。      |
| 42829      | カーソルによって指定された結果表を修正できないため、FOR UPDATE OF が無効です。                          |
| 42830      | 外部キーが親キーの記述に適合しません。                                                     |
| 42831      | 主キーまたはユニーク・キーの列では、NULL 値は許可されていません。                                     |
| 42832      | 操作がシステム・オブジェクトでは許可されていません。                                              |
| 42834      | 外部キーでは NULL 値が許可されていないため、SET NULL は指定できません。                             |
| 42835      | 名前派生表の間では、循環参照は指定できません。                                                 |
| 42836      | 再帰的な名前派生表の指定は無効です。                                                      |
| 42837      | 列の属性が現在の列属性と非互換であるため、列を変更できません。                                         |
| 42838      | 無効な表スペースの使用が見つかりました。                                                    |
| 42839      | 索引と長い列は、表から独立した表スペースには入れられません。                                          |
| 42840      | 無効な AS CAST オプションの使用が見つかりました。                                           |
| 42841      | パラメーター・マーカーは、ユーザー定義タイプまたは参照タイプにはできません。                                  |
| 42842      | 指定されたオプションが、列の記述と矛盾するため、列の定義が無効です。                                      |
| 42845      | 無効な VARIANT または EXTERNAL ACTION 関数の使用が見つかりました。                          |
| 42846      | ソース・タイプからターゲット・タイプへのキャストはサポートされません。                                     |
| 42852      | GRANT または REVOKE で指定された権限が無効であるか、または矛盾しています。(たとえば、ビューでの GRANT ALTER など) |

表 302. クラス・コード 42: 構文エラーまたはアクセス規則違反 (続き)

| SQLSTATE | 意味                                                                                                                   |
|----------|----------------------------------------------------------------------------------------------------------------------|
| 42853    | オプションの代替が両方とも指定されていたか、または同じオプションが複数回指定されています。                                                                        |
| 42854    | 選択リストの結果列データ・タイプは、型付きビューまたはマテリアライズ照会表定義に定義されているタイプと非互換です。                                                            |
| 42855    | このホスト変数に対する LOB の割り当ては許可されません。このカーソルでの、この LOB のすべてのフェッチに対するターゲット・ホスト変数は、ロケーターあるいは LOB 変数である必要があります。                  |
| 42858    | 操作を指定対象に適用させることができません。                                                                                               |
| 42863    | REXX で未定義のホスト変数が見つかりました。                                                                                             |
| 42866    | CREATE FUNCTION ステートメントの中の CAST FROM 文節または RETURNS 文節に入っているデータ・タイプが、ソース関数から戻されたデータ・タイプまたは関数内の RETURN ステートメントに適合しません。 |
| 42872    | FETCH ステートメント文節がカーソル定義と非互換です。                                                                                        |
| 42875    | CREATE SCHEMA で作成するオブジェクトには、スキーマ名と同じ修飾子を付ける必要があります。                                                                  |
| 42877    | 列名は修飾できません。                                                                                                          |
| 42878    | 無効な関数またはプロシージャ名が EXTERNAL キーワードで使用されました。                                                                             |
| 42879    | CREATE FUNCTION ステートメントの 1 つ以上の入力パラメーターのデータ・タイプが、ソース関数の対応するデータ・タイプに適合しません。                                           |
| 42880    | CAST TO と CAST FROM のデータ・タイプが一致しないか、または固定ストリングが必ず切り捨てられる可能性があります。                                                    |
| 42881    | 行ベース関数の使用が無効です。                                                                                                      |
| 42882    | 特定のインスタンス名の修飾子が、関数名の修飾子と等しくありません。                                                                                    |
| 42883    | 一致するシグニチャーの関数または方式が見つかりませんでした。                                                                                       |
| 42884    | 指定された名前と互換性のある引き数を持つルーチンが見つかりませんでした。                                                                                 |
| 42885    | CREATE FUNCTION ステートメントで指定した入力パラメーターの数が、SOURCE 文節で指定した関数によって与えられた数と一致しません。                                           |
| 42886    | IN、OUT、または INOUT パラメーター属性が一致しません。                                                                                    |
| 42887    | コンテキストとの関係で関数が無効です。                                                                                                  |

表 302. クラス・コード 42: 構文エラーまたはアクセス規則違反 (続き)

| SQLSTATE 値 | 意味                                                                                  |
|------------|-------------------------------------------------------------------------------------|
| 42888      | 表に主キーがありません。                                                                        |
| 42889      | 表にはすでに主キーがあります。                                                                     |
| 42890      | 列リストが参照文節で指定されていますが、識別された親表が、指定された列名によるユニーク制約を持っていません。                              |
| 42891      | 重複する UNIQUE 制約がすでに存在します。                                                            |
| 42893      | 別のオブジェクトが従属しているため、オブジェクトまたは制約をドロップできません。                                            |
| 42894      | DEFAULT 値が無効です。                                                                     |
| 42895      | 静的 SQL で、入力ホスト変数のデータ・タイプにプロシージャまたはユーザー定義関数のパラメーターとの互換性がないため、その入力ホスト変数は使用できません。      |
| 428A0      | ユーザー定義関数が基づいている関数でエラーが起きました。                                                        |
| 428A1      | ホスト・ファイル変数によって参照されたファイルにアクセスできません。                                                  |
| 428A2      | 表が区分化キーを持っていないため、複数ノード・ノード・グループに表を割り当てられません。                                        |
| 428A3      | 無効なパスがイベント・モニターに指定されています。                                                           |
| 428A4      | 無効な値がイベント・モニターのオプションに指定されています。                                                      |
| 428A5      | SET INTEGRITY ステートメントに指定されている例外表が、正しい構造ではないか、あるいは生成された列、制約、またはトリガーによって定義されています。     |
| 428A6      | SET INTEGRITY ステートメントに指定されている例外表は、チェック中表の 1 つと同じにはできません。                            |
| 428A7      | チェック中の表の数が、SET INTEGRITY ステートメントに指定されている例外表の数に一致しません。                               |
| 428A8      | 親表がチェック・ベンディング状態であるときに、下位表で SET INTEGRITY ステートメントを使用してチェック・ベンディング状態をリセットすることはできません。 |
| 428A9      | ノード範囲が無効です。                                                                         |
| 428AA      | 列名が、イベント・モニター表には無効な列です。                                                             |
| 428B0      | ROLLUP、CUBE、または GROUPING SETS に違法なネストがあります。                                         |
| 428B1      | 特定のノードについて指定していない表スペースのコンテナ指定の数が誤りです。                                               |

表 302. クラス・コード 42: 構文エラーまたはアクセス規則違反 (続き)

| SQLSTATE 値 | 意味                                                                   |
|------------|----------------------------------------------------------------------|
| 428B2      | コンテナのパス名が無効です。                                                       |
| 428B3      | 無効な SQLSTATE が RAISE_ERROR で指定されました。                                 |
| 428B7      | SQL ステートメントで指定された数値は、有効範囲外です。                                        |
| 429BE      | 主キーまたはユニーク・キーは、DIMENSIONS 文節にある列のサブセットです。                            |
| 428C0      | ノード・グループで唯一のノードであるため、そのノードをドロップすることはできません。                           |
| 428C1      | 1 つの ROWID 列のみ表に指定できます。                                              |
| 428C2      | 関数本体を調べた結果、指定された文節は CREATE FUNCTION ステートメントで指定されていなければならないことがわかりました。 |
| 428C4      | 述部オペレーターの各サイドにあるエレメントの数が同じではありません。                                   |
| 428C5      | データ・ソースからのデータ・タイプについて、データ・タイプのマッピングが見つかりません。                         |
| 428C9      | INSERT または UPDATE のターゲット列として ROWID 列を指定することはできません。                   |
| 428CA      | 追加モードの表にはクラスター索引を作成できません。                                            |
| 428CB      | 表スペースのページ・サイズは、それに関連するバッファ・プールのページ・サイズと一致していなければなりません。               |
| 428D1      | DATALINK の値によって参照されたファイルにアクセスできません。                                  |
| 428D4      | FOR に指定されているカーソルを OPEN、CLOSE、または FETCH ステートメントで参照することはできません。         |
| 428D5      | 終了ラベルが開始ラベルに一致しません。                                                  |
| 428D6      | UNDO は NOT ATOMIC ステートメントでは許可されていません。                                |
| 428D7      | 条件値は許可されていません。                                                       |
| 428D8      | SQLSTATE または SQLCODE 変数の宣言あるいは使用は許可されていません。                          |
| 428DB      | スーパータイプ、スーパー表、またはスーパービューとしてのオブジェクトは無効です。                             |
| 428DC      | 関数または方式は、このタイプのトランスフォームとして無効です。                                      |
| 428DD      | 必要なトランスフォームが定義されていません。                                               |
| 428DE      | PAGESIZE 値はサポートされていません。                                              |
| 428DF      | CREATE CAST に指定されているデータ・タイプが無効です。                                    |

表 302. クラス・コード 42: 構文エラーまたはアクセス規則違反 (続き)

| SQLSTATE | 意味                                                                   |
|----------|----------------------------------------------------------------------|
| 428DG    | CREATE CAST に指定されている関数が無効です。                                         |
| 428DH    | 操作は型付き表で無効です。                                                        |
| 428DJ    | 継承された列あるいは属性の変更あるいはドロップができません。                                       |
| 428DK    | 参照列の有効範囲はすでに定義されています。                                                |
| 428DL    | 外部あるいはソース関数のパラメーターに、定義済みの有効範囲がありません。                                 |
| 428DM    | 参照タイプの有効範囲表またはビューが無効です。                                              |
| 428DN    | SCOPE が外部関数の RETURNS 文節で指定されていないか、ソース関数の RETURNS 文節で定義されているかのいずれかです。 |
| 428DP    | タイプは構造化タイプではありません。                                                   |
| 428DQ    | 副表またはサブビューに、スーパー表またはスーパービューでない別のスキーマ名を指定することはできません。                  |
| 428DR    | 操作を副表に適用させることができません。                                                 |
| 428DS    | 指定された列の索引は、副表には定義できません。                                              |
| 428DT    | 式のオペランドの有効な有効範囲参照タイプは無効です。                                           |
| 428DU    | タイプが必須タイプ階層の中に入っていません。                                               |
| 428DV    | 参照解除演算子の左側オペランドが無効です。                                                |
| 428DW    | オブジェクト ID 列は参照解除演算子を使用して参照できません。                                     |
| 428DX    | 型付き表または型付きビュー階層のルート表またはルート・ビューを定義するために、オブジェクト ID の列が必要です。            |
| 428DY    | 副表の表統計データを更新できません。                                                   |
| 428DZ    | オブジェクト ID 列を更新できません。                                                 |
| 428E0    | 索引の定義が索引拡張子の定義と一致しません。                                               |
| 428E1    | 範囲作成関数の結果が、索引拡張子のキー・トランスフォーム関数の結果と矛盾しています。                           |
| 428E2    | キー・ターゲット・パラメーターの数あるいはタイプが索引拡張子のキー・トランスフォーム関数の数あるいはタイプと一致しません。        |
| 428E3    | 索引拡張子内の関数の引き数が無効です。                                                  |
| 428E4    | 関数は、CREATE INDEX EXTENSION ステートメントでサポートされていません。                      |
| 428E5    | ユーザー定義述部で指定できるのは SELECTIVITY 文節だけです。                                 |
| 428E6    | ユーザー定義述部にある方式の検索引き数が、対応する索引拡張子の検索方式内の検索引き数と一致しません。                   |

表 302. クラス・コード 42: 構文エラーまたはアクセス規則違反 (続き)

| SQLSTATE 値 | 意味                                                        |
|------------|-----------------------------------------------------------|
| 428E7      | ユーザー定義の述部中の比較演算子の後に続くオペランドのタイプが RETURNS データ・タイプと一致しません。   |
| 428E8      | 検索ターゲットまたは検索引き数パラメーターが、作成された関数のパラメーター名に一致しません。            |
| 428E9      | 引き数パラメーター名は同一の指数規則中で検索ターゲットおよび検索引き数の両方として出現しません。          |
| 428EA      | 型付きビューの全選択は無効です。                                          |
| 428EB      | スーパービュー内の列が更新可能であるなら、そのサブビュー内のそれに対応する列を読み取り専用にすることはできません。 |
| 428EC      | マテリアライズ照会表に指定された全選択が無効です。                                 |
| 428ED      | データ・リンクまたは参照タイプ属性を指定した構造化タイプは構成されません。                     |
| 428EE      | オプションがこのデータ・ソースでは無効です。                                    |
| 428EF      | オプションの値はこのデータ・ソースで無効です。                                   |
| 428EG      | このデータ・ソースに必要なオプションが欠落しています。                               |
| 428EH      | すでに定義済みのオプションを追加できません。                                    |
| 428EJ      | 追加されていないオプションを設定またはドロップできません。                             |
| 428EK      | 宣言されたグローバル一時表の修飾子は SESSION でなければなりません。                    |
| 428EL      | トランスフォーム関数は、関数または方式での使用では無効です。                            |
| 428EM      | TRANSFORM GROUP 文節が必要です。                                  |
| 428EN      | 使用されていないトランスフォーム・グループが指定されています。                           |
| 428EP      | 直接的に、または間接的に構造化タイプをそれ自身に依存させることはできません。                    |
| 428EQ      | ルーチンの戻りタイプをサブジェクト・タイプと同じにすることはできません。                      |
| 428ER      | 方式本文がドロップする前に、方式指定をドロップさせることはできません。                       |
| 428ES      | 方式本文が、方式指定の言語タイプに対応していません。                                |
| 428EU      | TYPE または VERSION がサーバー定義に指定されていません。                       |
| 428EV      | パススルー機能は、データ・ソースのタイプのためにサポートされていません。                      |

表 302. クラス・コード 42: 構文エラーまたはアクセス規則違反 (続き)

| SQLSTATE 値 | 意味                                                              |
|------------|-----------------------------------------------------------------|
| 428EW      | この表をマテリアライズ照会表に変換できないか、マテリアライズ照会表からこの表に変換できません。                 |
| 428EX      | 組み込み関数または方式であるため、ルーチンをトランスフォーム関数として使用できません。                     |
| 428EY      | ユーザー定義述部にある検索ターゲットのデータ・タイプが、指定された索引拡張子のソース・キーのデータ・タイプに一致していません。 |
| 428EZ      | OLAP 関数のウィンドウ指定は無効です。                                           |
| 428F0      | ROW 関数は少なくとも 2 つの列に組み込まれなければなりません。                              |
| 428F1      | SQL TABLE 関数は表結果を返さなければなりません。                                   |
| 428F2      | SQL プロシージャ内の RETURN ステートメント値のデータ・タイプは INTEGER でなければなりません。       |
| 428F3      | SCROLL および WITH RETURN は同時に指定できません。                             |
| 428F4      | FETCH で指定された SENSITIVITY はカーソルには許可されていません。                      |
| 428F6      | カーソルはスクロール可能ですが、結果表には表関数からの出力が関連しません。                           |
| 428F7      | SQL ルーチンにのみ適用する操作が、外部ルーチンで行われました。                               |
| 428F9      | シーケンス式はこのコンテキストでは指定できません。                                       |
| 428FA      | 10 進数の位取りをゼロにする必要があります。                                         |
| 428FB      | シーケンス名は、ID 列用のシステムで生成されたシーケンスではいけません。                           |
| 428FC      | 暗号化パスワードの長さが無効です。                                               |
| 428FD      | 暗号化解除に使用されたパスワードが、データの暗号化に使用されたパスワードと一致しません。                    |
| 428FE      | データが ENCRYPT 関数の結果ではありません。                                      |
| 428FF      | バッファ・プールの指定が無効です。                                               |
| 428FG      | ステージング表の定義に使用される表が無効です。                                         |
| 428FH      | SET INTEGRITY オプションが無効です                                        |
| 428FI      | ORDER OF が指定されましたが、この table-designator は ORDER BY 文節を含みません。     |
| 428FJ      | ORDER BY は、ビューまたはマテリアライズ照会表の外部全選択では許可されません。                     |



表 302. クラス・コード 42: 構文エラーまたはアクセス規則違反 (続き)

| SQLSTATE 値 | 意味                                                                                                      |
|------------|---------------------------------------------------------------------------------------------------------|
| 428FP      | サブジェクト・ビューでの各操作では、1 つの INSTEAD OF トリガーが許可されています。                                                        |
| 428FQ      | INSTEAD OF トリガーのビュー名に、WITH CHECK OPTION を使用して定義したビュー (対称ビュー) を、あるいは、直接、間接的に対称ビューが定義されたビューを指定することはできません。 |
| 428FU      | FROM SQL トランスフォーム関数または方式から戻された組み込みタイプが、TO SQL トランスフォーム関数または方式の対応する組み込みタイプと一致しません。                       |
| 428FV      | 方式をオーバーライド方式として定義できません。                                                                                 |
| 42901      | 列関数に列名がありません。                                                                                           |
| 42903      | WHERE 文節または SET 文節に、列関数などの無効な参照があります。                                                                   |
| 42904      | コンパイル・エラーのため、SQL プロシージャは作成されませんでした。                                                                     |
| 42907      | ストリングが長すぎます。                                                                                            |
| 42908      | 必要な列リストがステートメントにありません。                                                                                  |
| 42910      | このステートメントは、複合ステートメントでは許可されません。                                                                          |
| 42911      | 10 進数の除算で、結果の位取りが負の値になるものは無効です。                                                                         |
| 42912      | 列がカーソルの select ステートメントの UPDATE 文節で識別されていないため、この列を更新できません。                                               |
| 42914      | 副照会で参照された表が影響を受けるため、DELETE は無効です。                                                                       |
| 42915      | 無効な参照制約が見つかりました。                                                                                        |
| 42916      | 別名が反復チェーンになるため、別名を作成できません。                                                                              |
| 42917      | オブジェクトを明示的にドロップまたは変更できません。                                                                              |
| 42918      | システム定義のデータ・タイプ名 (たとえば INTEGER) で、ユーザー定義のデータ・タイプを作成することはできません。                                           |
| 42919      | ネストされた複合ステートメントは許可されていません。                                                                              |
| 42921      | コンテナを表スペースに追加できません。                                                                                     |
| 42925      | 再帰的名前派生表は SELECT DISTINCT を指定できません。UNION ALL の指定が必要です。                                                  |
| 42928      | WITH EMPTY TABLE は、この表に指定できません。                                                                         |
| 42932      | プログラム準備の前提事項に誤りがあります。                                                                                   |
| 42939      | 指定された ID はシステム使用のために予約されているため、この名前は使用できません。                                                             |

表 302. クラス・コード 42: 構文エラーまたはアクセス規則違反 (続き)

| SQLSTATE 値 | 意味                                                                      |
|------------|-------------------------------------------------------------------------|
| 42962      | 長い列、LOB 列、または構造化タイプ列は、索引、キー、または制約では使用することができません。                        |
| 42968      | 現行ソフトウェア・ライセンスがないため、接続が失敗しました。                                          |
| 42969      | 内部の制約あるいは無効なセクション番号のため、パッケージが作成されず、現行の作業単位がロールバックされました。                 |
| 42972      | 複数のオペランド列にある、結合条件参照列にある式。                                               |
| 42985      | ステートメントはルーチンでは許可されていません。                                                |
| 42987      | ステートメントが、プロシージャまたはトリガーで許可されていません。                                       |
| 42989      | 式に基づいている GENERATED 列を BEFORE トリガーで使用することはできません。                         |
| 42991      | BOOLEAN データ・タイプは、現在内部的にのみサポートされています。                                    |
| 42993      | 定義された列が、ログに記録するには大きすぎます。                                                |
| 42994      | ロー・デバイス・コンテナは、現在このシステムではサポートされていません。                                    |
| 42995      | 要求された関数は、グローバル一時表に適用されません。                                              |
| 42997      | このバージョンの DB2 アプリケーション・リクエスト、DB2 アプリケーション・サーバー、または両者の組み合わせでは、機能はされていません。 |
| 429A0      | 最初にログを取得しないように定義されている親表を、外部キーで参照することはできません。                             |
| 429A1      | ノード・グループが表スペースについて有効ではありません。                                            |
| 429A9      | DataJoiner が SQL ステートメントを処理することはできません。                                  |
| 429B2      | 構造型タイプまたは列に指定されているインライン長さの値が小さすぎます。                                     |
| 429B3      | オブジェクトが副表に定義されている可能性があります。                                              |
| 429B4      | データ・フィルター関数は LANGUAGE SQL 関数にはなれません。                                    |
| 429B5      | 索引拡張子内のインスタンス・パラメーターのデータ・タイプが無効です。                                      |
| 429B8      | PARAMETER STYLE JAVA で定義されたルーチンは、パラメーターまたは戻りタイプとして構造化タイプを持つことができません。    |
| 429B9      | DEFAULT または NULL を属性割り当てに使用することはできません。                                  |
| 429BA      | FEDERATED キーワードは、連合データベース・オブジェクトと一緒に使用する必要があります。                        |

表 302. クラス・コード 42: 構文エラーまたはアクセス規則違反 (続き)

| SQLSTATE 値 | 意味                                                 |
|------------|----------------------------------------------------|
| 429BB      | パラメーターまたは属性に指定されているデータ・タイプは、SQL ルーチンではサポートされていません。 |
| 429BC      | ALTER TABLESPACE ステートメントに、複数のコンテナ・アクションがあります。      |

## クラス・コード 44 WITH CHECK OPTION 違反

表 303. クラス・コード 44: WITH CHECK OPTION 違反

| SQLSTATE 値 | 意味                                               |
|------------|--------------------------------------------------|
| 44000      | 結果の行がビュー定義を満たしていないため、INSERT または UPDATE は許可されません。 |

## クラス・コード 46 Java DDL

表 304. クラス・コード 46: Java DDL

| SQLSTATE 値 | 意味                                  |
|------------|-------------------------------------|
| 46001      | Java DDL - 無効な URL。                 |
| 46002      | Java DDL - 無効な jar 名。               |
| 46003      | Java DDL - 無効なクラスの削除。               |
| 46007      | Java DDL - 無効なシグニチャー。               |
| 46008      | Java DDL - 無効な方式指定。                 |
| 46103      | Java ルーチンが ClassNotFound 例外を検出しました。 |
| 46501      | Java DDL - オプションのコンポーネントが設定されていません。 |

## クラス・コード 51 無効なアプリケーション状態

表 305. クラス・コード 51: 無効なアプリケーション状態

| SQLSTATE 値 | 意味                                     |
|------------|----------------------------------------|
| 51002      | SQL ステートメントの実行要求に対応するパッケージが見つかりませんでした。 |
| 51003      | 整合性トークンが一致しません。                        |

表 305. クラス・コード 51: 無効なアプリケーション状態 (続き)

| SQLSTATE 値 | 意味                                                                                              |
|------------|-------------------------------------------------------------------------------------------------|
| 51004      | SQLDA のアドレスが無効です。                                                                               |
| 51005      | 以前のシステム・エラーによって、このエラーを使用できません。                                                                  |
| 51008      | プリコンパイルされたプログラムのリリース番号が無効です。                                                                    |
| 51015      | バインド時にエラーが検出されたセクションを実行しようとしてしました。                                                              |
| 51017      | ユーザーはログオンしていません。                                                                                |
| 51021      | アプリケーション処理がロールバック操作を実行するまで、SQL ステートメントを実行できません。                                                 |
| 51022      | 接続 (現行または休止のどちらか) が CONNECT ステートメントで指定されたサーバーにすでに存在するときは、許可名を指定した CONNECT は無効です。                |
| 51023      | データベースは、データベース・マネージャーの他のインスタンスがすでに使用中です。                                                        |
| 51024      | ビューの操作不能のマークが付いているため、ビューは使用できません。                                                               |
| 51025      | XA トランザクション処理環境のアプリケーションが、SYNCPOINT TWOPHASE でバインドされていません。                                      |
| 51026      | イベント・モニターのターゲット・バスが、すでに他のイベント・モニターによって使用されているため、イベント・モニターをオンにできません。                             |
| 51027      | 表がユーザー保守のマテリアライズ照会表であるか、チェック・ペンディング状態でないため、SET INTEGRITY ステートメントの IMMEDIATE CHECKED オプションは無効です。 |
| 51028      | パッケージが操作不能とマークされているので、使用できません。                                                                  |
| 51030      | ALLOCATE CURSOR または ASSOCIATE LOCATORS ステートメントで参照されているプロシージャは、アプリケーション処理内で呼び出されていません。           |
| 51034      | MODIFIES SQL DATA (SQL データの変更) を使って定義されたルーチンは、そのルーチンが呼び出されたコンテキストで無効です。                         |
| 51035      | 値がこのセッションでまだシーケンスについて生成されていないため、PREVVAL 式は使用できません。                                              |
| 51038      | SQL ステートメントは、このルーチンによって最早発行されない可能性があります。                                                        |
| 51039      | ENCRYPTION PASSWORD 値が設定されていません。                                                                |

## クラス・コード 53 無効なオペランドまたは矛盾する指定

表 306. クラス・コード 53: 無効なオペランドまたは矛盾する指定

| SQLSTATE 値 | 意味                                         |
|------------|--------------------------------------------|
| 53040      | 指定されたバッファ・プールは、指定されたデータベース・パーティションに存在しません。 |

## クラス・コード 54 SQL または製品の限界の超過

表 307. クラス・コード 54: SQL または製品の限界の超過

| SQLSTATE 値 | 意味                                              |
|------------|-------------------------------------------------|
| 54001      | ステートメントが長すぎるか、または複雑すぎます。                        |
| 54002      | ストリング定数が長すぎます。                                  |
| 54004      | ステートメントの SELECT または INSERT リストにある表名または項目が多すぎます。 |
| 54006      | 連結の結果が長すぎます。                                    |
| 54008      | キーが長すぎるか、キーの持っている列が多すぎるか、またはキー列が長すぎます。          |
| 54010      | 表のレコード長が長すぎます。                                  |
| 54011      | 表またはビューに指定されている列の数が多すぎます。                       |
| 54023      | 関数またはプロシージャのパラメーターまたは引き数の数が、限界を超えています。          |
| 54028      | 並行 LOB ハンドルが最大数に達しました。                          |
| 54029      | オープン・ディレクトリー走査の最大数に達しました。                       |
| 54030      | イベント・モニターの最大数がすでにアクティブです。                       |
| 54031      | 最大数のファイルが、すでにイベント・モニターに割り当てられています。              |
| 54032      | 表が最大サイズに達しました。                                  |
| 54033      | 区分化マップの最大数に達しました。                               |
| 54034      | 表スペースのすべてのコンテナ名を結合した長さが長すぎます。                   |
| 54035      | 内部オブジェクトの制限を超えました。                              |
| 54036      | コンテナのパス名が長すぎます。                                 |
| 54037      | 表スペースのコンテナ・マップが複雑すぎます。                          |
| 54038      | ネストされたルーチンまたはトリガーの最大の深さを超えました。                  |
| 54045      | タイプ階層の最大レベルを超えています。                             |

表 307. クラス・コード 54: SQL または製品の限界の超過 (続き)

| SQLSTATE 値 | 意味                                        |
|------------|-------------------------------------------|
| 54046      | 索引拡張子内の最大許容可能パラメーター数を超過しています。             |
| 54047      | 表が最大サイズを超えました。                            |
| 54048      | 十分なページ・サイズの TEMPORARY 表スペースが存在しません。       |
| 54049      | 構造化タイプのインスタンスの長さがシステム制限を超過しています。          |
| 54050      | 許可されている最大属性が構造化タイプで超過しています。               |
| 54052      | バッファ・プールのブロック・ページ数が、バッファ・プールのサイズには大きすぎます。 |
| 54053      | BLOCKSIZE に指定された値が有効範囲内にありません。            |

### クラス・コード 55 前提条件の状態にないオブジェクト

表 308. クラス・コード 55: 前提条件の状態にないオブジェクト

| SQLSTATE 値 | 意味                                                             |
|------------|----------------------------------------------------------------|
| 55001      | データベースを移行する必要があります。                                            |
| 55002      | 説明表が正しく定義されていません。                                              |
| 55006      | オブジェクトが現在同じアプリケーション処理によって使用されているため、オブジェクトをドロップできません。           |
| 55007      | オブジェクトが現在同じアプリケーション処理によって使用されているため、オブジェクトを変更できません。             |
| 55009      | システムが、読み取り専用ファイルまたは書き込み保護ディスクットに対して書き込みを試みました。                 |
| 55012      | クラスタリング索引はすでに表に存在しています。                                        |
| 55019      | 表が無効な状態にあるため、操作できません。                                          |
| 55022      | ファイル・サーバーは、このデータベースに登録されていません。                                 |
| 55023      | ルーチン呼び出しエラーが起きました。                                             |
| 55024      | 表に関連するデータも他の表スペースにあるため、表スペースをドロップできません。                        |
| 55025      | データベースを再始動する必要があります。                                           |
| 55026      | TEMPORARY 表スペースをドロップできません。                                     |
| 55031      | エラー・マッピング・ファイルの形式に誤りがあります。                                     |
| 55032      | このアプリケーションが始動された後に、データベース・マネージャーが停止されたため、CONNECT ステートメントは無効です。 |

表 308. クラス・コード 55: 前提条件の状態にないオブジェクト (続き)

| SQLSTATE | 意味                                                            |
|----------|---------------------------------------------------------------|
| 55033    | イベント・モニターが作成または修正された同じ作業単位で、イベント・モニターを活動化できません。               |
| 55034    | イベント・モニターが操作に対して無効な状態にあります。                                   |
| 55035    | 表は保護されているため、ドロップできません。                                        |
| 55036    | ノードが区分化マップからドロップされていないため、ノードをドロップできません。                       |
| 55037    | 表が複数ノード・ノード・グループにないため、区分化キーをドロップできません。                        |
| 55038    | ノード・グループが再平衡されているため、ノード・グループを使用できません。                         |
| 55039    | 表スペースが適切な状態にないため、アクセスまたは状態の遷移は許可されません。                        |
| 55041    | 再平衡の進行中は、コンテナを表スペースに追加できません。                                  |
| 55043    | タイプに基づく型付き表または型付きビューが存在している場合、構造化タイプの属性は変更できません。              |
| 55045    | 必要なコンポーネントがサーバーで使用可能になっていないため、ルーチンの SQL アーカイブ (SAR) を作成できません。 |
| 55046    | 指定された SQL アーカイブがターゲット環境に適合しません。                               |
| 55047    | 外部関数またはメソッドが、連合オブジェクトにアクセスしようとした。                             |
| 55048    | 暗号化されたデータは暗号化できません。                                           |
| 55049    | イベント・モニター表が正しく定義されていません。                                      |
| 55051    | ALTER BUFFERPOOL ステートメントが現在進行中です。                             |
| 55054    | 方式をオーバーライド方式として定義できません。                                       |

## クラス・コード 56 その他の SQL または製品エラー

表 309. クラス・コード 56: その他の SQL または製品エラー

| SQLSTATE | 意味                                                      |
|----------|---------------------------------------------------------|
| 56031    | 混合データと DBCS データが、このシステムではサポートされていないため、文節またはスカラー関数が無効です。 |
| 56033    | 長ストリング列の挿入または更新値は、ホスト変数または NULL である必要があります。             |

表 309. クラス・コード 56: その他の SQL または製品エラー (続き)

| SQLSTATE 値 | 意味                                                                                |
|------------|-----------------------------------------------------------------------------------|
| 56038      | この環境ではサポートされない要求機能です。                                                             |
| 56084      | DRDA では、LOB データはサポートされていません。                                                      |
| 56091      | コンパウンド SQL ステートメントを実行した結果、複数のエラーが起きました。                                           |
| 56092      | 許可名が、ユーザー ID とグループ ID の両方であるため、許可のタイプを判別できません。                                    |
| 56095      | BIND オプションが無効です。                                                                  |
| 56097      | LONG VARCHAR および LONG VARGRAPHIC フィールドは、DEVICE 上にビルドされる TABLESPACE では許可されません。     |
| 56098      | 暗黙の再バインドまたは準備中に、エラーが発生しました。                                                       |
| 56099      | REAL データ・タイプがターゲット・データベースによってサポートされていません。                                         |
| 560A0      | LOB 値におけるアクションが失敗しました。                                                            |
| 560AA      | このシステムで UCS-2 がサポートされていないため、文節またはスカラー関数が無効です。                                     |
| 560AC      | ラッパー定義は、データ・ソースの指定されたタイプまたはバージョンに使用できません。                                         |
| 560AF      | ゲートウェイ集線装置を使用している場合、PREPARE ステートメントはサポートされていません。                                  |
| 560B0      | 新しいサイズ値は、表スペース・コンテナのサイズ変更には無効です。                                                  |
| 560B1      | ストアード・プロシージャのカーソル指定が無効です。                                                         |
| 560B7      | 複数行 INSERT の場合、シーケンス式の使用は各行について同じでなければなりません。                                      |
| 560BB      | 動的に準備された CALL ステートメントの INOUT パラメーターの場合、同じホスト変数を USING 文節と INTO 文節の両方で使用する必要があります。 |
| 560BC      | ファイルにアクセス中にエラーが発生しました。                                                            |
| 560BD      | 連合サーバーは、データ・ソースから予期しないエラー・コードを受け取りました。                                            |
| 560BF      | 暗号化機能は使用できません。                                                                    |



## クラス・コード 57 リソースが使用不能、またはオペレーターの介入

表 310. クラス・コード 57: リソースが使用不能、またはオペレーターの介入

| SQLSTATE 値 | 意味                                                      |
|------------|---------------------------------------------------------|
| 57001      | 1 次索引がないため、その表は使用できません。                                 |
| 57003      | 指定されたバッファ・プールはアクティブではありません。                             |
| 57007      | DROP または ALTER が保留中のため、オブジェクトを使用できません。                  |
| 57009      | 仮想記憶またはデータベース・リソースが一時的に使用不能になっています。                     |
| 57011      | 仮想記憶またはデータベース・リソースを使用できません。                             |
| 57012      | 非データベース・リソースを使用できません。これは、以降のステートメントの正常な実行には影響しません。      |
| 57013      | 非データベース・リソースを使用できません。これは、以降のステートメントの正常な実行に影響する可能性があります。 |
| 57014      | 要求にしたがって処理が取り消されました。                                    |
| 57016      | 表スペースがアクティブではないので、アクセスできません。                            |
| 57017      | 文字変換が定義されていません。                                         |
| 57019      | リソースに問題があるため、ステートメントが失敗しました。                            |
| 57020      | データベースのあるドライブがロックされています。                                |
| 57021      | ディスク・ドライブのドアが開いています。                                    |
| 57022      | ステートメントの許可 ID が適切なデータ・スペースを所有していないため、表が作成されませんでした。      |
| 57030      | アプリケーション・サーバーへの接続が、インストール先定義の限界を超えている可能性があります。          |
| 57032      | 並列処理できる最大数のデータベースが、すでに始動しています。                          |
| 57033      | 自動ロールバックなしで、デッドロックまたはタイムアウトが起きました。                      |
| 57036      | このトランザクション・ログは、現在のデータベースにはありません。                        |
| 57046      | データベースまたはインスタンスが静止しているため、新しいトランザクションを開始できません。           |
| 57047      | ディレクトリがアクセス不能のため、内部データベース・ファイルを作成できません。                 |
| 57048      | 表スペースのコンテナにアクセス中に、エラーが起きました。                            |
| 57049      | オペレーティング・システム処理限界に達しました。                                |
| 57050      | ファイル・サーバーは現在使用できません。                                    |

表 310. クラス・コード 57: リソースが使用不能、またはオペレーターの介入 (続き)

| SQLSTATE 値 | 意味                                                  |
|------------|-----------------------------------------------------|
| 57051      | 見積もり CPU コストがリソースの限度を超過しています。                       |
| 57052      | そのノードは、すべての TEMPORARY 表スペースについてコンテナがないため使用できません。    |
| 57053      | 操作が競合するため、この表に対してこの操作を実行することはできません。                 |
| 57055      | 使用できる TEMPORARY 表スペースのページ・サイズが不足しています。              |
| 57056      | データベースが NO PACKAGE LOCK モードであるため、パッケージは使用できません。     |
| 57057      | SQL ステートメントの一連の DRDA での前の条件のため、SQL ステートメントは実行されません。 |
| 57059      | 指定したアクションを行うための十分なスペースが、表スペースにありません。                |

## クラス・コード 58 システム・エラー

表 311. クラス・コード 58: システム・エラー

| SQLSTATE 値 | 意味                                                                          |
|------------|-----------------------------------------------------------------------------|
| 58004      | システム・エラー (必ずしも後続の SQL ステートメントの正常な実行を妨げるものではありません) が起きました。                   |
| 58005      | システム・エラー (後続の SQL ステートメントの正常な実行を妨げます) が起きました。                               |
| 58008      | 分散プロトコル・エラーのため、実行が失敗しました。このエラーは、後続の DDM コマンドまたは SQL ステートメントの正常な実行には影響しません。  |
| 58009      | 会話の割り振り解除の原因となる分散プロトコル・エラーのため、実行が失敗しました。                                    |
| 58010      | 分散プロトコル・エラーのため、実行が失敗しました。このエラーは、後続の DDM コマンドまたは SQL ステートメントの正常な実行に影響を与えません。 |
| 58011      | バインド処理の進行中は、DDM コマンドは無効です。                                                  |
| 58012      | 指定したパッケージ名と整合性トークンを持つバインド処理がアクティブではありません。                                   |
| 58014      | DDM コマンドはサポートされていません。                                                       |

表 311. クラス・コード 58: システム・エラー (続き)

| SQLSTATE | 意味                                              |
|----------|-------------------------------------------------|
| 58015    | DDM オブジェクトはサポートされていません。                         |
| 58016    | DDM パラメーターはサポートされていません。                         |
| 58017    | DDM パラメーターの値がサポートされていません。                       |
| 58018    | DDM 応答メッセージがサポートされていません。                        |
| 58023    | システム・エラーのため、現行プログラムが取り消されました。                   |
| 58030    | 入出力エラーが起きました。                                   |
| 58031    | システム・エラーのため、接続が失敗しました。                          |
| 58032    | 分離モードのユーザー定義関数に処理を使用できません。                      |
| 58034    | DMS 表スペース内のオブジェクトについてページを検索しているときに、エラーが見つかりました。 |
| 58035    | DMS 表スペース内のオブジェクトのためにページを解放している間に、エラーが見つかりました。  |
| 58036    | 指定された内部表スペース ID が存在しません。                        |




---

## 追加のトピックの更新

---

### コントロール・センター

コントロール・センターは、以下の方法で開始することができます。

- 別のツールの「ツール (Tools)」メニューから「コントロール・センター (Control Center)」を選択する。
- 別のツールのツールバーから  アイコンをクリックする。
- **db2cc** コマンドを入力する。
- Windows® システムでは、「スタート」ボタンをクリックして、「プログラム」 → 「IBM® DB2®」 → 「汎用管理ツール (General Administration Tools)」 → 「コントロール・センター (Control Center)」を選択する。
- Linux システムでは、デスクトップの「IBM DB2」フォルダーを開いて、「コントロール・センター (Control Center)」を選択する。

**注:** 「コントロール・センター」を開始するには、インスタンス所有者の権限を持つユーザーでログインしている必要があります。

コントロール・センターは、システム、DB2 Universal Database™ インスタンス、DB2 Universal Database for OS/390® and z/OS™ サブシステム、データベース、および表やビューなどのデータベース・オブジェクトを管理するために使用します。

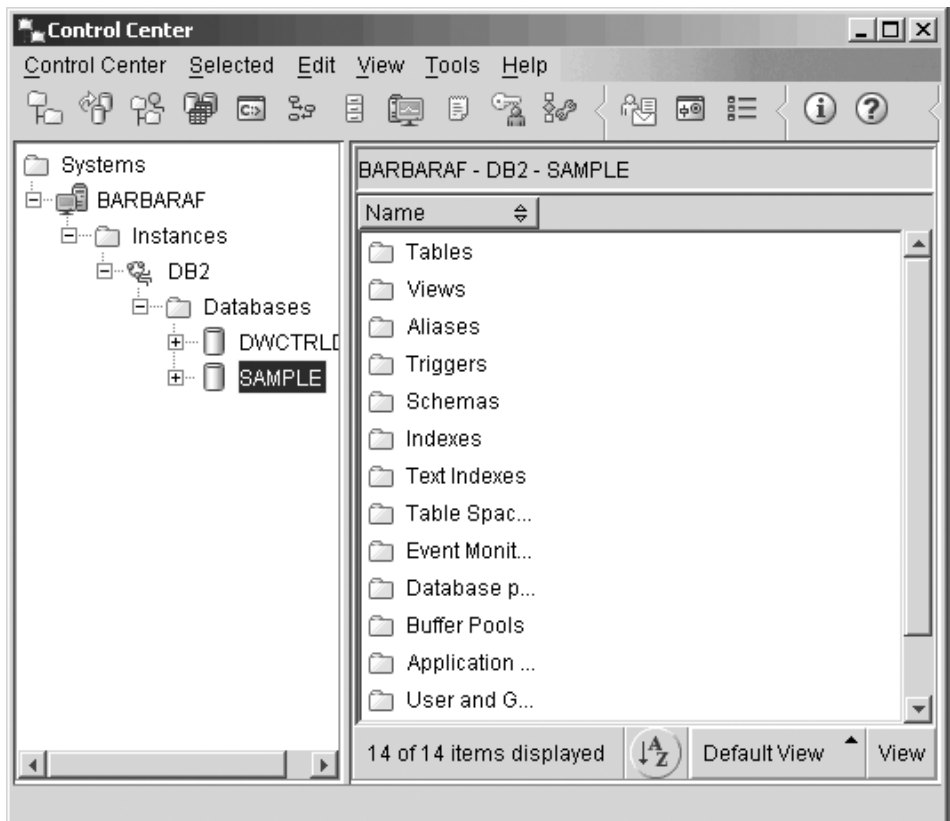


図 19. 「コントロール・センター (Control Center)」 ウィンドウ

コントロール・センターでは、システム、インスタンス、データベース、およびデータベース・オブジェクトのすべてを管理できます。コントロール・センターから他のセンターおよびツールをオープンして、照会、ジョブ、およびスクリプトの最適化、データウェアハウジング・タスクの実行、ストアード・プロシーチャーの作成、DB2 コマンドを使用した処理に役立てることもできます。

以下は、コントロール・センターを使用して実行できる主なタスクの一部です。

- DB2 システム、連合システム、DB2 for z/OS and OS/390 システム、インスタンス、データベース、およびデータベース・オブジェクトを、オブジェクト・ツリーに追加する。
- データベース・オブジェクトを管理する。データベース、表スペース、表、ビュー、索引、トリガー、およびスキーマを作成、変更、およびドロップできます。ユーザーを管理することもできます。
- データを管理する。データをロード、インポート、エクスポート、および再編成できます。統計を収集することもできます。

- データベースまた表スペースのバックアップまたはリストアによる、予防保守を実行する。
- インスタンスおよびデータベースを構成および調整する。
- データベース接続 (DB2 Connect™ サーバーおよびサブシステムなど) を管理する。
- DB2 for z/OS and OS/390 サブシステムを管理する。
- アプリケーションを管理する。
- アクセス・プランを表示する Visual Explain を使用して、照会を分析する。
- コマンド・センターやヘルス・センターなどの他のツールを立ち上げる。

多くの場合、これらのタスクをより迅速にまた簡単に実行するのに役立つ、ウィザードやランチパッドが用意されています。

**注:** コントロール・センターを介してタスクをスケジュールするには、まずツール・カタログを作成する必要があります。DB2 のインストール時にツール・カタログを作成しなかった場合は、「ツール (Tools)」 → 「ツール設定 (Tools Settings)」 → 「スケジューラー設定 (Scheduler Settings)」を選択して、ツール・カタログを作成できます。

## ツールの呼び出し

以下のツールは、ツールバー上のアイコンを選択することによって呼び出すことができます。



コントロール・センター



レプリケーション・センター



サテライト管理センター



データウェアハウス・センター (データウェアハウジング・オプション付き)



コマンド・センター



タスク・センター



インフォメーション・カタログ・センター (データウェアハウジング・オプション付き)



ヘルス・センター



ジャーナル



ライセンス・センター



デベロップメント・センター



インフォメーション・センター

Visual Explain ツールを開始するには、データベース名を右マウス・ボタン・クリックして、「**EXPLAIN されたステートメント・ヒストリーの表示 (Show Explained Statements History)**」または「**SQL の EXPLAIN (Explain SQL)**」のどちらかを選択します。

メモリー・ビジュアライザーを開始するには、インスタンスを右マウス・ボタン・クリックして、「**メモリー使用量の表示 (View Memory Usage)**」を選択します。

## ストレージ管理ビュー

「ストレージ管理」ビューを使用するには、データベース、データベース・パーティション、または表スペースを右マウス・ボタン・クリックして、「**ストレージの管理 (Manage Storage)**」を選択します。

「ストレージ管理」ビューは、データベースのストレージ状態をモニターするために使用します。「ストレージ管理」ビューは、ストレージ管理表およびストアド・プロシージャのグラフィカル・インターフェースです。ストレージ管理ビューでは、データベース、データベース・パーティション・グループ、または表スペースのストレージ・スナップショットをとることができます。

表スペースのスナップショットをとる場合、指定された表スペースの有効範囲内に定義された表、索引、およびコンテナーに関して、システム・カタログおよびデータベース・モニターから統計情報が収集されます。データベースまたはデータベース・パーティション・グループのスナップショットをとると、指定されたデータベースまたはデータベース・パーティション・グループ内に定義されているすべての表スペースに関して、統計情報が収集されます。データベースのスナップショットをとると、そのデータベース内のすべてのデータベース・パーティション・グループに関して、統計情報が収集されます。

以下のような異なるタイプのストレージ・スナップショットがあり、ストレージの異なった面をモニターするのに役立ちます。

- スペース使用量は、表スペースのスナップショットを使用してモニターできます。



- パーティション・データベースの場合に限り、データ・スキュー (データベース分散) は、データベース・パーティション・グループのスナップショットを介して最適なものをモニターできます。
- 索引のクラスター化率は、データベース・パーティション・グループ・スナップショットと表スペース・スナップショットの両方でキャプチャーすることができます。

索引のクラスター化率は、索引フォルダーの詳細ビューを介して提供されます。「ストレージ管理」ビューによって、データ・スキュー、スペース使用量、および索引クラスター化率に関して、しきい値を設定することもできます。ターゲット・オブジェクトが指定されたしきい値を超えると、「ストレージ管理」ビューの中のそのオブジェクトとその親オブジェクトのアイコンに、警告フラグまたはアラーム・フラグのマークが付けられます。

## 管理できるオブジェクト

コントロール・センターでは、以下のような多くの DB2 オブジェクトを管理できます。

- システム
- インスタンス
- データベース
- 表スペース
- 表
- ビュー
- 索引
- スキーマ
- トリガー
- ストアード・プロシージャ
- ユーザー定義タイプ
- ユーザー定義関数
- ユーザー定義メソッド
- パッケージ
- 別名
- イベント・モニター
- バッファーク・プール
- ユーザーおよびグループ

オブジェクトに対して実行できるアクションを調べるには、オブジェクトを選択して右マウス・ボタンをクリックします。ポップアップ・メニューに、有効なアクションがリストされます。

---

## ブラウザをオンライン・ヘルプおよびトピックの表示用に構成

Internet Explorer を使用している場合、ブラウザ設定を構成して、オンライン・ヘルプおよびトピックを最適に表示できるようにする必要があります。検索ウィンドウが開いたとき、インフォメーション・センターからのリンクが 3 フレームのレイアウトにではなく、検索ウィンドウにロードされます。

### 手順:

オンライン・ヘルプおよびトピックを表示するように Internet Explorer を構成するには、「ツール」 → 「インターネット オプション」 → 「詳細設定」 → 「ブラウザ」 → 「ショートカットを起動するためにウィンドウを再使用する」チェック・ボックスをクリアします。一度この設定を変更すると、Java インフォメーション・センターからのリンクは、新規のブラウザ・ウィンドウに開かれます。

---

## DB2 インフォメーション・センターの呼び出し

DB2 インフォメーション・センターを使用すると、DB2 Universal Database および DB2 Connect をビジネスで最大限に活用するのに必要な情報すべてにアクセスできます。

### 前提条件:

DB2 インフォメーション・センターを呼び出す前に、以下が必要です。

- DB2 オンライン情報を *DB2 HTML* ドキュメンテーション CD からインストールする。
- ブラウザーをオンライン・ヘルプおよびトピックの表示用に構成する。

### 手順:

DB2 インフォメーション・センターをデスクトップから呼び出すには、以下のようになります。

- Windows オペレーティング・システム: 「スタート」 → 「プログラム」 → 「IBM DB2」 → 「情報」 → 「DB2 ドキュメンテーション・セット」 をクリックする。

DB2 インフォメーション・センターをコマンド行から呼び出すには、以下のようになります。

- Linux および UNIX オペレーティング・システム: **db2help** コマンドを発行する。
- Windows オペレーティング・システム: **db2docs.exe** コマンドを発行する。

### 関連概念:

- 88 ページの『ブラウザからアクセスする DB2 インフォメーション・センター』

### 関連タスク:

- *Data Links Manager* 概説およびインストール の『ブラウザから DB2 インフォメーション・センターにアクセスしてトピックを検索する』
- *Data Links Manager* 概説およびインストール の『DB2 HTML ドキュメンテーション CD から技術資料を直接参照する』
- 168 ページの『マシンにインストールされている HTML ドキュメンテーションをアップデートする』
- *Data Links Manager* 概説およびインストール の『DB2 文書の検索』
- 1958 ページの『ブラウザをオンライン・ヘルプおよびトピックの表示用に構成』



---

## 付録. 特記事項

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品、プログラムまたはサービスの操作性の評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権の許諾については、下記の宛先に書面にてご照会ください。

〒106-0032 東京都港区六本木 3-2-31

IBM World Trade Asia Corporation

Licensing

**以下の保証は、国または地域の法律に沿わない場合は、適用されません。**IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

本書に含まれる情報には、技術的に不正確なもの、または誤植が含まれる場合があります。これらに対する変更は、定期的に行われます。これらの変更は、資料の改訂版に含まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

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

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム（本プログラムを含む）との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Canada Limited  
Office of the Lab Director  
8200 Warden Avenue  
Markham, Ontario  
L6G 1C7  
CANADA

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームの

アプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。したがって IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。

それぞれの複製物、サンプル・プログラムのすべての部分、またはすべての派生した創作物には、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。© Copyright IBM Corp. \_西暦年\_. All rights reserved.

---

## 商標

以下は、IBM Corporation の商標です。

|                                                 |                  |
|-------------------------------------------------|------------------|
| ACF/VTAM                                        | LAN Distance     |
| AISPO                                           | MVS              |
| AIX                                             | MVS/ESA          |
| AIXwindows                                      | MVS/XA           |
| AnyNet                                          | Net.Data         |
| APPN                                            | NetView          |
| AS/400                                          | OS/390           |
| BookManager                                     | OS/400           |
| C Set++                                         | PowerPC          |
| C/370                                           | pSeries          |
| CICS                                            | QBIC             |
| Database 2                                      | QMF              |
| DataHub                                         | RACF             |
| DataJoiner                                      | RISC System/6000 |
| DataPropagator                                  | RS/6000          |
| DataRefresher                                   | S/370            |
| DB2                                             | SP               |
| DB2 Connect                                     | SQL/400          |
| DB2 Extenders                                   | SQL/DS           |
| DB2 OLAP Server                                 | System/370       |
| DB2 Universal Database                          | System/390       |
| Distributed Relational<br>Database Architecture | SystemView       |
| DRDA                                            | Tivoli           |
| eServer                                         | VisualAge        |
| Extended Services                               | VM/ESA           |
| FFST                                            | VSE/ESA          |
| First Failure Support Technology                | VTAM             |
| IBM                                             | WebExplorer      |
| IMS                                             | WebSphere        |
| IMS/ESA                                         | WIN-OS/2         |
| iSeries                                         | z/OS             |
|                                                 | zSeries          |

以下は、他社の商標または登録商標です。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。



"Pentium" "MMX" "ProShare" "LANDesk" および "Action Media" は Intel Corporationの米国およびその他の国における商標。  
(Intelの商標は<http://www.intel.com/sites/corporate/tradmarx.htm>で参照できます)。

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

UNIX は、The Open Group がライセンスしている米国およびその他の国における登録商標です。

他の会社名、製品名およびサービス名などはそれぞれ各社の商標または登録商標です。



# 索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

## [ア行]

- アクセス・パス, 最適化 1258
- アナライザー
  - OS/400 版
    - 作成, SQL パッケージの 663, 681
  - UNIX 版, 呼び出しパラメーター 518
  - Windows 版, 呼び出しパラメーター 518
- アナライザー・レポート
  - asnanalyze コマンド 517
- アプライ修飾子
  - アプライ・プログラムの始動時に使用 478, 488
  - 数, 関連したサブスクリプション・セットの 687, 689
  - 命名規則 633
- アプライ・コントロール表
  - SUBS\_COLS (サブスクリプション列) 500
  - SUBS\_MEMBR (サブスクリプション・メンバー) 502
  - SUBS\_SET (サブスクリプション・セット) 507
- アプライ・トレース (APPLYTRACE) 表
  - ブルーニング 617, 630
- アプライ・トレール (APPLYTRAIL) 表
  - ブルーニング 617, 630
- アプライ・プログラム
  - 許可要件 663, 670
  - コネクティビティ 663, 664
- アプライ・プログラム (続き)
  - 実行時処理ステートメント 721, 722
  - データ・ブロッキング 687, 694
  - トランザクション処理 687
  - トランザクション・モード処理 696
  - 表モード処理 687, 696
  - ミニサイクル 687, 694
  - ユーザー ID 663, 670
- OS/400 版
  - 開始 478, 488
  - 作成, SQL パッケージの 663, 681
  - 設定 663
  - セットアップ 680, 682
  - 停止 478, 489
- UNIX 版
  - 開始 478, 533
  - 構成 663, 678
  - 設定 663
  - セットアップ 676
  - 停止 478, 489
  - バインディング 663
  - バインド 678
  - パスワード・ファイル 663, 673
  - apply\_path パラメーター 478, 480, 533, 535
  - apply\_qual パラメーター 478, 480, 533, 534
  - control\_server パラメーター 478, 481, 533, 535
  - copyonce パラメーター 478, 481, 533, 538
  - delay パラメーター 478, 482, 533, 539
  - errwait パラメーター 478, 482, 533, 540
  - inamsg パラメーター 478, 483, 533, 537
- アプライ・プログラム (続き)
  - UNIX 版 (続き)
    - loadxit パラメーター 478, 483, 533, 536
    - logreuse パラメーター 478, 483, 533, 536
    - logstdout パラメーター 478, 484, 533, 536
    - notify パラメーター 478, 484, 533, 537
    - opt4one パラメーター 478, 484, 533, 539
    - pwdfile パラメーター 478, 485, 533, 535
    - sleep パラメーター 478, 485, 533, 538
    - spillfile パラメーター 478, 485, 533, 541
    - sqlerrcontinue パラメーター 478, 486, 533, 541
    - term パラメーター 478, 487, 533, 540
    - trlreuse パラメーター 478, 487, 533, 539
  - Windows 版
    - 開始 478, 533
    - 構成 663, 678
    - 設定 663
    - セットアップ 676
    - 操作 478
    - 停止 478, 489
    - バインディング 663
    - バインド 678
    - パスワード・ファイル 663, 673
    - apply\_path パラメーター 478, 480, 533, 535
    - apply\_qual パラメーター 478, 480, 533, 534
    - control\_server パラメーター 478, 481, 533, 535

アプライ・プログラム (続き)

Windows 版 (続き)

copyonce パラメーター 478, 481, 533, 538  
 delay パラメーター 478, 482, 533, 539  
 errwait パラメーター 478, 482, 533, 540  
 inamsg パラメーター 478, 483, 533, 537  
 loadxit パラメーター 478, 483, 533, 536  
 logreuse パラメーター 478, 483, 533, 536  
 logstdout パラメーター 478, 484, 533, 536  
 notify パラメーター 478, 484, 533, 537  
 opt4one パラメーター 478, 484, 533, 539  
 pwdfile パラメーター 478, 485, 533, 535  
 sleep パラメーター 478, 485, 533, 538  
 spillfile パラメーター 478, 485, 533, 541  
 sqlerrcontinue パラメーター 478, 486, 533, 541  
 term パラメーター 478, 487, 533, 540  
 trlreuse パラメーター 478, 487, 533, 539

z/OS 版

開始 478, 533, 613, 614  
 設定 663  
 セットアップ 682  
 停止 478, 489  
 apply\_path パラメーター 478, 480, 533, 535  
 apply\_qual パラメーター 478, 480, 533, 534  
 control\_server パラメーター 478, 481, 533, 535  
 copyonce パラメーター 478, 481, 533, 538

アプライ・プログラム (続き)

z/OS 版 (続き)

db2\_subsystem パラメーター 482, 535  
 delay パラメーター 478, 482, 533, 539  
 errwait パラメーター 478, 482, 533, 540  
 inamsg パラメーター 478, 483, 533, 537  
 loadxit パラメーター 478, 483, 533, 536  
 logreuse パラメーター 478, 483, 533, 536  
 logstdout パラメーター 478, 484, 533, 536  
 notify パラメーター 478, 484, 533, 537  
 opt4one パラメーター 478, 484, 533, 539  
 pwdfile パラメーター 478, 485, 533, 535  
 sleep パラメーター 478, 485, 533, 538  
 spillfile パラメーター 478, 485, 533, 541  
 term パラメーター 478, 487, 533, 540  
 trlreuse パラメーター 478, 487, 533, 539

アプリケーション

コントロール・ヒープ、設定

342

サンプル・プログラム 1113

ODBC 119

アプリケーション設計

照合シーケンスの設定 926

アプリケーションの移行 1021

アプリケーション・コントロール・ヒープ・サイズ構成パラメーター 342

アプリケーション・プログラミング

グ・インターフェース (API)

スレッド間にコンテキストを設定する

sqlAttachToCtx() 742

アプリケーション・プログラミング

グ・インターフェース (API) (続き)

スレッド間にコンテキストを設定する (続き)

sqlBeginCtx() 742

sqlDetachFromCtx() 742

sqlEndCtx() 742

sqlGetCurrentCtx() 742

sqlInterruptCtx() 742

sqlSetTypeCtx() 742

アプリケーション・プログラム

SQLj、実行例 747

アプレット

各種ヒント 772

JDBC のビルド 776

SQLj のビルド 778

異機種種のデータ・ソース

制約事項

CCD 表の場合 637

update-anywhere 637

登録 637

異機種種のレプリケーション

制限

update-anywhere 650

制約事項

集約表 687, 707

CCD 表 645

multi-tier レプリケーション

687, 710

update-anywhere 687, 713

ソースの登録 640

移行

アプリケーション 1021

データベース 394

AIX で 134

DB2 394

推奨事項 106

DB2 Data Links for AIX パージョン 6.1 からの 134

DB2 Data Links for AIX パージョン 7.1 からの 134

DB2 Data Links for AIX パージョン 7.2 からの 134

DB2 File Manager for Solaris 124

移行 (続き)

DB2 File Manager for  
Windows 143

Solaris 上で 124

Windows 上の 143

XML Extender をバージョン 8 へ  
1129

一時表  
ユーザー定義 326

一時ファイル  
LOAD コマンド 1149

イベントに基づくスケジューリング  
687, 699

イベント・モニター  
DROP ステートメント 1572

イメージ  
バックアップ 399

インスタンス・プロファイル・レジ  
ストリー 308

インスタンス・レベル・プロファイ  
ル・レジストリー 308

インストール  
ウェアハウス・エージェント  
AIX 223  
iSeries 214  
Linux 223  
Solaris オペレーティング環境  
223  
z/OS 197

DB2 製品  
手動での 161

Web 用インフォメーション・カ  
タログ・センター  
AIX で 216  
Windows NT 上で 188

インストールする準備  
ウェアハウス・エージェント  
194

インターバル・タイミング 687, 699

インターフェース言語 111

インフォメーション・カタログ管理  
ウィザード 215

インポート  
タグ言語ファイル  
コマンド行からの 1096  
データ 1234

インポート (続き)

CLI LOAD ユーティリティ  
による 870

インポート機能  
クライアント・プロファイルの構  
成 116

打ち切られた API およびデータ構造  
1021

ウィザード  
インフォメーション・カタログ管  
理 215  
マルチサイト更新 93

ウェアハウス管理インターフェース  
開始  
AIX 1095

ウェアハウス・エージェント  
インストールする準備 194  
ソフトウェア要件 194  
ロギングの開始 213

AIX  
インストール 223  
実行可能ファイル 190

iSeries  
インストール 214

Linux  
インストール 223  
実行可能ファイル 190

Solaris オペレーティング環境  
インストール 223  
実行可能ファイル 190

z/OS  
インストール 197  
概要 225  
ユーザー定義のプログラム  
203  
DataPropagator ステップの自動  
化に使用 211

ウェアハウス・エージェント・デー  
モン  
z/OS  
開始 202  
開始済みとして開始 226

ウェアハウス・サーバー  
開始と停止  
AIX 1076

ウェアハウス・サーバー (続き)

実行状況の確認  
AIX 1077

ウェアハウス・ステップ  
スケジューリング  
トリガー・プログラムで 204

ウェアハウス・トランスフォーマー  
データのクリーン 1089

z/OS  
環境変数変数データでの文字  
の削減 208  
DB2 for z/OS での設定 206

ウェアハウス・プロセス  
タスク・フロー 1088

ウェアハウス・ロガー  
開始と停止  
AIX 1076  
実行状況の確認  
AIX 1077

ウォーム・スタート、キャプチャ  
ー・プログラム  
UNIX 版 542, 547  
Windows 版 542, 547  
z/OS 版 542, 547

エージェント処理  
applheapsz 構成パラメーター  
340

エクスポート  
データベース表ファイル 1134  
ファイル・タイプ修飾子 1134  
DB2 Data Links Manager の考慮  
事項 1134

エクスポート機能  
クライアント・プロファイルの作  
成 98

エクスポート・ユーティリティ  
DB2 Data Links Manager 421  
「エラー発生時にキャプチャーを停  
止」オプション 637, 648

エラー・メッセージ  
データベース記述ブロック構造  
926  
無効なチェックサム、データベー  
ス構成ファイル 1276  
ロールフォワード中 1007

エラー・メッセージ (続き)

Data Links ファイル・マネージャ  
ー 1033

延期

キャプチャー・プログラム

UNIX 版 526

Windows 版 526

z/OS 版 526

応答ファイル

キーワード

Windows 32 ビット・オペレー  
ティング・システムの 163

大文字小文字の区別

カーソル名引き数 860

オブジェクト ID (OID) 1484

CREATE TABLE ステートメント  
1484

CREATE VIEW ステートメント  
1469

オブジェクト・タイプ

更新

インフォメーション・カタロ  
グ・センターのウィンドウを  
使用した 1097

タグ言語を使用した 1097

オブジェクト・リンクおよび組み込  
み

DB2 AD クライアントでのサポー  
ト 757

オペレーティング・システム

サポートされる AIX バージョ  
ン 759

サポートされる HP-UX バージ  
ョン 761

サポートされる Linux バージ  
ョン 762

サポートされる Solaris バージ  
ョン 765

サポートされる Windows バージ  
ョン 766

DB2 によってサポートされた  
1122

## [力行]

カーソル

ロールバック間の保留 827

外部 CCD 表

概要 687

multi-tier レプリケーション 687,  
710

外部キー

制約名規則 1484

ALTER TABLE による追加  
1402

ALTER TABLE によるドロップ  
1402

外部トリガー・プログラム

実行ステップ 1085

例 1086

会計情報ストリング設定 API 1019  
開始

アプライ・プログラム

OS/400 478

OS/400 版 488

UNIX 478, 533

UNIX 版 478, 533

Windows 478, 533

Windows 版 478, 533

z/OS 478, 533, 613

z/OS 版 478, 533, 614

ウェアハウス・エージェント・デ  
ーモン

開始済みタスクとして 226

z/OS 202

キャプチャー・プログラム

UNIX 542

UNIX 版 542

Windows 542

Windows サービスの使用 634

Windows 版 542

z/OS 542, 613

z/OS 版 542, 613

レプリケーション・アラート・モ  
ニター

z/OS 613

z/OS 版 613

型付きビュー

サブビューの定義 1469

カタログ

TCP/IP ノード 95

カタログ統計

カタログ表の説明 318

カタログ表

説明 318

登録 637

カタログ表、登録 638

カタログ・ノード 445

カタログ・ビュー

CASTFUNCTIONS 1357

COLIDENTATTRIBUTES 1401

FUNCMAPPINGS 1356

FUNCPARMS (ROUTINEPARMS  
を参照) 1334

FUNCTIONS (ROUTINES を参  
照) 1336

INDEXES 1344

PREDICATESPECS 1358

PROCEDURES (ROUTINES を参  
照) 1336

PROCPARMS (ROUTINEPARMS  
を参照) 1334

ROUTINEPARMS (以前の  
FUNCPARMS、  
PROCPARMS) 1334

ROUTINES (以前の  
FUNCTIONS、  
PROCEDURES) 1336

SEQUENCES 1358

SYSSTATINDEXES 1351

SYSSTAT.FUNCTIONS  
(SYSSTAT.ROUTINES を参  
照) 1349

SYSSTAT.ROUTINES (以前は  
SYSSTAT.FUNCTIONS) 1349

TRANSFORMS 1357

各国語

考慮事項 190

各国語サポート (NLS)

双方向 CCSID サポート 270

活動化、サブスクリプション・セッ  
トの 687, 693

監査

ギャップ、データの 687, 708

コールド・スタート 687, 708

監査 (続き)

ソース・データ 637, 647

環境変数 308

キャプチャー・プログラム 663, 676

DB2CODEPAGE 190, 663, 677

DB2DBDFT 663, 677

DB2INSTANCE 190, 663, 676

LANG 190

LC\_ALL 190

LIBPATH 663, 677

環境変数データ・セット

文字数の削減 208

管理

許可要件 663, 666

基礎集約表

使用法 687, 707

定義 687, 704

記述子

ヘッダー・フィールド

初期設定値 819

レコード・フィールド

初期設定値 819

CLI 関数、単一フィールドの取得 905

CLI 関数、単一フィールドの設定 910

記述データ

有効なデータ・タイプ 1102

既存の表、ターゲットとして 687, 715

逆方向タイプ・マッピング

構文 456

説明 427

キャッシュ

EXECUTE ステートメント 1602

ギャップの検出 687, 708

キャプチャー・エンキュー

(CAPENQ) 表 517

キャプチャー・コントロール表

CAPENQ (キャプチャー・エンキュー) 517

CAPTRACE (キャプチャー・トレース) 521

SIGNAL (シグナル) 522

キャプチャー・コントロール・サーバー

複数のキャプチャー・スキーマ 663, 675

キャプチャー・スキーマ

使用、複数の 675

複数の使用 663

命名規則 633

キャプチャー・トリガー

許可要件 663, 670

キャプチャー・トレース

(CAPTRACE) 表

構造 521

プルーニング 617, 629

キャプチャー・パラメーターの変更

UNIX 版 526

Windows 版 526

z/OS 版 526

キャプチャー・プログラム

許可要件 663, 669

コールド・スタートの防止 617, 630

コネクティビティー 663, 664

実行、複数の 663, 675

設定、環境変数 663, 676

ユーザー ID 663, 669

OS/400 版

許可要件 663, 668

作成、SQL パッケージの

663, 681, 682

ジャーナルおよびジャーナ

ル・レシーバーの管理 663, 684

ジャーナル・エントリー・タイプ 734

設定 663

セットアップ 680, 682

UNIX 版

ウォーム・スタート・パラメーター 542, 547

延期 526

開始 542

コールド・スタート・パラメーター 542, 547

構成 663, 677

再開 526

キャプチャー・プログラム (続き)

UNIX 版 (続き)

再初期化 526

状況 526

設定 663

セットアップ 676

操作 526

停止 526

バインディング 663

バインド 677

パラメーターの変更 526

プルーニング 526

autoprunce パラメーター 526, 529, 542, 544

autostop パラメーター 526, 529, 542, 544

capture\_path パラメーター 542, 543

capture\_schema パラメーター 526, 527, 542, 543

capture\_server パラメーター 526, 527, 542, 543

commit\_interval パラメーター 526, 529, 542, 544

lag\_limit パラメーター 526, 542, 544

logreuse パラメーター 526, 530, 542, 545

logstdout パラメーター 526, 530, 542, 545

memory\_limit パラメーター 526, 530, 542, 545

monitor\_interval パラメーター 526, 530, 542, 545

monitor\_limit パラメーター 526, 531, 542, 545

prune\_interval パラメーター 526, 531, 542, 546

retention\_limit パラメーター 526, 531, 542, 546

sleep\_interval パラメーター 526, 531, 542, 546

startmode パラメーター 542, 547

term パラメーター 526, 532, 542, 548

キャプチャー・プログラム (続き)

UNIX 版 (続き)

trace\_limit パラメーター 526,  
532, 542, 548

Windows 版

ウォーム・スタート・パラメ  
ーター 542, 547

延期 526

開始 542

コールド・スタート・パラメ  
ーター 542, 547

構成 663, 677

再開 526

再初期化 526

状況 526

設定 663

セットアップ 676

操作 526

停止 526

バインディング 663

バインド 677

パラメーターの変更 526

プルーニング 526

autoprune パラメーター 526,  
529, 542, 544

autostop パラメーター 526,  
529, 542, 544

capture\_path パラメーター  
542, 543

capture\_schema パラメーター  
526, 527, 542, 543

capture\_server パラメーター  
526, 527, 542, 543

commit\_interval パラメーター  
526, 529, 542, 544

lag\_limit パラメーター 526,  
542, 544

logreusel パラメーター 526,  
530, 542, 545

logstdout パラメーター 526,  
530, 542, 545

memory\_limit パラメーター  
526, 530, 542, 545

monitor\_interval パラメーター  
526, 530, 542, 545

キャプチャー・プログラム (続き)

Windows 版 (続き)

monitor\_limit パラメーター  
526, 531, 542, 545

prune\_interval パラメーター  
526, 531, 542, 546

retention\_limit パラメーター  
526, 531, 542, 546

sleep\_interval パラメーター  
526, 531, 542, 546

startmode パラメーター 542,  
547

term パラメーター 526, 532,  
542, 548

trace\_limit パラメーター 526,  
532, 542, 548

z/OS 版

ウォーム・スタート・パラメ  
ーター 542, 547

延期 526

開始 542, 613

コールド・スタート・パラメ  
ーター 542, 547

再開 526

再初期化 526

状況 526

設定 663

セットアップ 682

操作 526

停止 526

パラメーターの変更 526

プルーニング 526

autoprune パラメーター 526,  
529, 542, 544

autostop パラメーター 526,  
529, 542, 544

capture\_path パラメーター  
542, 543

capture\_schema パラメーター  
526, 527, 542, 543

capture\_server パラメーター  
526, 527, 542, 543

commit\_interval パラメーター  
526, 529, 542, 544

lag\_limit パラメーター 526,  
542, 544

キャプチャー・プログラム (続き)

z/OS 版 (続き)

logreusel パラメーター 526,  
530, 542, 545

memory\_limit パラメーター  
526, 530, 542, 545

monitor\_interval パラメーター  
526, 530, 542, 545

monitor\_limit パラメーター  
526, 531, 542, 545

prune\_interval パラメーター  
526, 531, 542, 546

retention\_limit パラメーター  
526, 531, 542, 546

sleep\_interval パラメーター  
526, 531, 542, 546

startmode パラメーター 542,  
547

term パラメーター 526, 532,  
542, 548

trace\_limit パラメーター 526,  
532, 542, 548

キャプチャー・プログラムの中断

UNIX 版 526

Windows 版 526

z/OS 版 526

キャプチャー・モニター (CAPMON)  
表

プルーニング 617, 629

許可

アプライ・プログラム 663, 670

管理のための 663, 666, 668

キャプチャー・トリガーの場合  
663, 670

キャプチャー・プログラムの場合  
663, 669

レプリケーション・アラート・モ  
ニター用 663, 672

行

サブセット化 637

ソースでの 644

ターゲットでの 716

使用可能、レプリケーションに  
637, 644

定義、ターゲット表での 687,  
716



行 (続き)

登録、ソース表内の 637, 644

行 (水平方向) のサブセット化 637

ソースでの 644

ターゲットでの 716

行キャプチャー規則 637, 644

競合検出

概要 637, 655

要件 637, 647

レベル 637, 656

共通表式

再帰 1317

定義 1317

select ステートメント 1317

切り捨て

出力ストリング 860

数値 1364

クエスチョン・マーク (?)

EXECUTE パラメーター・マーク  
ー 1602

区分化キー

考慮事項 1484

定義、表の作成時 1484

ALTER TABLE ステートメント  
1402

ALTER TABLE による追加  
1402

ALTER TABLE によるドロップ  
1402

区分化キーでのハッシュ化 1484

組み込み SQL

ファイルのビルド 769

DB2 AD クライアントでのサポ  
ート 757

クライアント・プロファイル

インポート機能を使用した構成  
116

エクスポート機能を使用した作成  
98

クラス・ライブラリー

Java 781

グローバル・レベル・プロファイ

ル・レジストリー 308

クロスローダー・ユーティリティー

478, 498

結果セット

ハンドルに関連する、CLI 関数  
923

結合

区分化キーの考慮事項 1484

ソース 637

結合、ソースとしての 659

言語

DB2 のサポート 242

現行レシーバーのサイズ 663, 684

コード・セット

DB2 のサポート 242

コード・ページ

サポート 111

ユーロ通貨記号による 233

1394 のユニコードへの変換

前の変換表 304

DB2 のサポート 242

EXPORT コマンド 1134

IMPORT コマンド 1234

Shift JIS のユニコードへの変換

前の変換表 304

コード・ページ・ファイル・タイプ

修飾子 945, 1149

コールド・スタート、キャプチャ

ー・プログラム

回避 617, 630

UNIX 版 542, 547

Windows 版 542, 547

z/OS 版 542, 547

コア・レベル関数 796

更新

競合 637, 655

削除および挿入として 637, 649

更新可能ビュー 1469

更新済み主キー列 637, 649

構成

アプライ・プログラム

UNIX 版 663, 678

Windows 版 663, 678

キャプチャー・プログラム

UNIX 版 663, 677

Windows 版 663, 677

構成 (続き)

クライアントからサーバーへの接  
続

コマンド行プロセッサ

(CLP) 97

コネクティビティー 663, 664

データベース

更新 1276

レプリケーション・アラート・モ  
ニター

UNIX 663

UNIX 版 679

Windows 663

Windows 版 679

構成アシスタント (CA)

クライアント・プロファイルの構  
成 116

クライアント・プロファイルの作  
成 98

データベース接続の構成

一般 94

構成パラメーター

applheapsz 340

app\_ctl\_heap\_sz 342

collate\_info 363

DB2

APPLHEAPSZ 663

DBHEAP 663

LOGBUFSZ 663

LOGFILSIZ 663

LOGPRIMARY 663

LOGSECOND 663

MAXAPPLS 663

mon\_heap\_sz 363

priv\_mem\_thresh 365

構成パラメーター、DB2 用

APPLHEAPSZ 677

DBHEAP 677

LOGBUFSZ 677

LOGFILSIZ 677

LOGPRIMARY 677

LOGSECOND 677

MAXAPPLS 677

構造化照会言語 (SQL)

基本オペランド、割り当てと比較  
1364

- 構造化照会言語 (SQL) (続き)
- 比較演算の概要 1364
  - 割り当て 1364
- 構造タイプ
- DROP ステートメント 1572
- 互換性
- 演算タイプに関する規則 1364
  - 規則 1364
  - データ・タイプ 1364
  - データ・タイプの要約 1364
- コネクティビティー
- コントロール表の障害のリカバー 617, 631
  - DB2 プラットフォーム間 663, 664, 665
- コマンド
- AUTOCONFIGURE 1131
  - db2advise 1286
  - db2dlmmg 124, 134, 143
  - db2imigr 124, 134, 143
  - db2inidb 1315
  - db2inspf 1218
  - db2licm 122, 180, 185
  - db2look 1289
  - db2relocatedb 1305
  - db2set 124, 134, 143
  - db2sqljcustomize 1298
  - db2sqljprint 1304
  - db2support 1310
  - EXPORT 1134
  - hostname 143
  - IMPORT 1234
  - INSPECT 1144
  - LOAD 1149
  - nslookup 143
  - PRECOMPILE 1190
  - QUIESCE 1219
  - REORGCHK 1222
  - RESTORE DATABASE 1267
  - RUNSTATS 1258
  - START DATABASE MANAGER 1279
  - UNQUIESCE 1266
  - UPDATE DATABASE CONFIGURATION 1276
- コマンド (続き)
- 参照: レプリケーション・コマンド
- コマンド行プロセッサ (CLP)
- クライアントからサーバーへの接続の構成 97
  - 名前付きパイプの構成 155
  - ノードのカタログ 95
  - DB2 AD クライアント 757
- コンテキスト
- マルチスレッド DB2 アプリケーションでの設定 742
- コンテナー
- 名前 399
  - CREATE TABLESPACE ステートメント 1562
- コンテナー 文節、CREATE TABLESPACE ステートメント 1562
- コンデンス CCD 表 687
- コントロール表
- キャプチャー・サーバー 512
  - 許可要件、OS/400 の場合 682
  - クイック・リフェレンス 概観 724
  - キャプチャー・サーバー 512
- 再バインド、パッケージおよびプラン 617, 624
- 再編成 617, 625
- 作成
- 非 DB2 リレーショナル・ソースの場合 663, 674
  - 複数のセット 663, 675
  - IASP グループ内 663, 674
  - OS/400 での 663, 674
  - UNIX および Windows での 663, 673
  - z/OS での 663, 673
- 静的 617, 626
- 接続障害のリカバー 617, 631
- 動的 617, 623
- 入出力エラーのリカバー 617, 631
- 付与、権限の、OS/400 の場合 668
- ブルーニング 617, 626
- コントロール表 (続き)
- 保守 617, 623
  - CAPENQ (キャプチャー・エンキュー) 517
  - CAPMON (キャプチャー・モニター) ブルーニング 617, 629
  - CAPTRACE (キャプチャー・トレース) 構造 521
  - ブルーニング 617, 629
  - CCD (整合変更データ) ターゲット・サーバー 731
  - MONTRAIL (モニター・トレール) 729
  - OS400 許可要件 663
  - OS/400 での権限の付与 663
  - OS/400 の権限の取り消し 600
  - RUNSTATS ユーティリティ 617, 624
  - SIGNAL (シグナル) 522
  - SUBS\_COLS (サブスクリプション列) 500
  - SUBS\_MEMBR (サブスクリプション・メンバー) 502
  - SUBS\_SET (サブスクリプション・セット) 507
- コントロール・センター
- 説明 1953
  - マルチサイト更新 93
- コントロール・センター拡張機能
- オブジェクトの変更 338
  - オブジェクト例の追加 335
  - フォルダーの追加 332
  - プラグイン・デベロッパーのガイドライン 339
- コンパイラー
- サポートされる AIX バージョン 759
  - サポートされる HP-UX バージョン 761
  - サポートされる Linux バージョン 762
  - サポートされる Solaris バージョン 765

コンパイラー (続き)  
サポートされる Windows パージ  
ョン 766  
ファイルのビルド 769  
コンパイル  
SQLj プログラム  
例 747  
コンパイル・オプション  
HP-UX  
CLI アプリケーション 865  
CLI ルーチン 866  
Windows  
CLI ルーチン 844  
コンパウンド SQL  
CLI 836  
コンパウンド・ファイル・タイプ修  
飾子 1234  
コンプリート CCD 表 687  
コンプレッション・ディクショナリ  
ー (z/OS) 617, 622

## [サ行]

サーバー定義  
作成の概要 445  
サーバー・オプション  
説明 445  
有効な設定 437  
サービス  
Windows SCM 634  
再開  
キャプチャー・プログラム  
UNIX 版 526  
Windows 版 526  
z/OS 版 526  
再帰  
照会 1317  
再帰的共通表式 1317  
再キャプチャー、変更の  
(update-anywhere) 650  
再バインド  
パッケージおよびプラン 617  
再バインド、パッケージおよびプラン  
624  
再編成  
コントロール表 625

作業単位 (UOW) 表  
ブルーニング 617, 628  
列、CCD 表の 687, 708  
索引  
再編成 1222  
主キー、マッチングに使用 1402  
ターゲット表 687, 718  
統計 1258  
ユニーク・キー、マッチングに使  
用 1402  
DROP ステートメントを使用した  
削除 1572  
索引アドバイザー  
使用 1107  
索引名  
主キー制約 1484  
ユニーク制約 1484  
削除表のビュー 1469  
作成  
コントロール表 663  
多次元表 279  
作成、コントロール表の 673  
サテライト管理  
サテライト関数と DB2 Personal  
Edition との組み合わせ 87  
サテライト・コントロール・サーバ  
ー  
インストール  
エラー・メッセージ 398  
作動不能ビュー 1469  
差分リフレッシュ・レプリケーショ  
ン 637  
参照： 変更キャプチャー・レプ  
リケーション  
サブスクライブ、ソースの 687, 690  
サブスクリプション列 (SUBS\_COLS)  
表 500  
サブスクリプション・イベント  
(SUBS\_EVENT) 表  
通知、イベントの 687, 699  
サブスクリプション・サイクル 694  
サブスクリプション・セット  
数、アプライ修飾子の 687, 689  
活動化レベル 687, 693  
行 687, 716  
作成 687, 690

サブスクリプション・セット (続き)  
参照保全 687, 715  
実行時処理ステートメント 721,  
722  
処理モード 687, 696  
スケジューリング  
イベントに基づく 687, 699  
時間に基づく 687, 699  
ストアド・プロシージャ  
687, 698  
設定の追加 580  
追加 580  
データ整合性 687, 715  
ミニサイクル 687, 694  
メンバーの追加 687, 701  
列 687, 716  
multi-tier レプリケーション 687,  
710  
SQL ステートメント 687, 698  
update-anywhere レプリケーショ  
ン 687, 712  
サブスクリプション・セット  
(SUBS\_SET) 表 507  
サブスクリプション・セット・メン  
バー  
数、サブスクリプション・セット  
ごとの 687, 688  
設定への追加 687  
選択、ターゲット・タイプの  
687, 704  
追加 557, 701  
定義、ターゲット・キーの 687,  
718  
適用、行のサブセット 687, 716  
適用、列のサブセット 687, 716  
マッピング、データ・タイプの  
687, 718  
マッピング、列間の 687, 718  
multi-tier レプリケーション 687,  
710  
update-anywhere レプリケーショ  
ン 687, 712  
サブスクリプション・メンバー  
(SUBS\_MEMBR) 表 478, 497, 502

サブセット化  
 行、ターゲットでの変更の 687, 716  
 登録済みの行、変更を含む 637, 644  
 登録済み列の 637, 642  
 列、ターゲットでの 687, 716  
 サポートされない API およびデータ構造 1021  
 サポートされないデータ・タイプ・マッピング 456  
 算出列 716  
 作成 721, 723  
 ソース表 687, 707  
 ターゲット表で 687  
 CD 表 687, 707  
 参照タイプ  
 比較 1364  
 参照保全 715  
 レプリケーション 687  
 サンプル・プログラム 1113  
 シーケンス  
 DROP ステートメント 1572  
 時間に基づくスケジューリング 687, 699  
 シグナル (SIGNAL) 表  
 構造 522  
 プルーニング 629  
 シグナル表  
 構造 522  
 プルーニング 617  
 システム開始タスク 613, 615  
 システム変更ジャーナル管理 663, 684  
 システム要件  
 Data Links Manager  
 AIX 134  
 Solaris オペレーティング環境 124  
 Windows 143  
 システム・クロック  
 AIX、同期化 134  
 Solaris オペレーティング環境、同期化 124  
 Windows、同期化 143  
 システム・コンテナ、CREATE  
 TABLESPACE ステートメント 1562  
 システム・モニター・スイッチ  
 連合 476  
 実行管理機能オブジェクト 663, 685  
 実行時処理 687, 698, 721, 722  
 自動ブルーニング 617, 627  
 自動リスタート・マネージャー (ARM) 613, 615  
 シフトイン文字、割り当てで切り捨てられない 1364  
 ジャーナル  
 エントリー・タイプ 734  
 開始 663, 684  
 管理 663, 684  
 作成 663, 683  
 作成、ソース表の 663, 683  
 使用 682  
 使用、リモート・ジャーナル関数 657  
 セットアップ 682  
 デフォルトのメッセージ・キュー 663, 686  
 登録、ソースとして 637, 638  
 メッセージ・キュー 663  
 リモート・ジャーナル機能 637  
 レプリケーション 663  
 レプリケーションのセットアップ 663  
 QSQRN ジャーナル 663, 682  
 ジャーナル・メッセージ・キュー 686  
 ジャーナル・レシーバー  
 管理 663, 684  
 作成、ソース表の 663, 683  
 しきい値 663, 684  
 システム管理 663, 684  
 ジャーナル・レシーバー削除出口ルーチン 663, 686  
 保守 617, 618  
 保存 617, 622  
 ユーザー管理 663, 685  
 ジャーナル・レシーバー削除出口ルーチン  
 削除 686  
 ジャーナル・レシーバー削除出口ルーチン (続き)  
 除去 663  
 説明 663, 686  
 登録 663, 687  
 修飾子  
 ファイル・タイプ  
 EXPORT コマンド 1134  
 IMPORT コマンド 1234  
 LOAD コマンド 1149  
 従属関係  
 オブジェクト間の相互 1572  
 集約表  
 基礎集約 687, 707  
 変更集約 687, 707  
 主キー  
 作成 1484  
 使用される、ターゲットとして 687, 719  
 相対レコード番号、OS/400 の場合 658  
 ドロップ  
 ALTER TABLE の使用 1402  
 論理パーティション化 637, 649  
 ALTER TABLE による追加 1402  
 OS400 での相対レコード番号 637  
 述部  
 定義、ターゲット表の 687, 716  
 取得  
 単一記述子フィールド CLI 関数 905  
 データ関数 CLI 関数 895  
 手動でのデータベースの追加 94  
 準備済み SQL ステートメント  
 実行 1602  
 ホスト変数置換 1602  
 順方向タイプ・マッピング  
 構文 456  
 説明 1381  
 初期化全選択 1317  
 照会  
 再帰 1317

- 照会 (続き)
  - 例
    - SELECT ステートメント 1317
- 照合順序
  - ストリング比較の規則 1364
  - ユーザー定義 926
  - ユニコード 273
- 情報の表示
  - バックアップ・ユーティリティ 399
- シリアライゼーション
  - SQL ステートメント実行 742
- 垂直方向 (列) のサブセット化
  - ソースでの 637, 642
  - ターゲットでの 687, 716
- 水平方向 (行) のサブセット化
  - ソースでの 637, 644
  - ターゲットでの 687, 716
- 数値
  - スケール 1322
  - 精度 1322
  - 比較 1364
  - SQL 操作での割り当て 1364
- スキーマ
  - 新規データベースに 926
  - 命名規則 633
- スケール
  - 数値の
    - SQLLEN 変数によって決まる 1322
  - データの
    - SQL における数値変換 1364
    - SQL における比較 1364
    - SQLLEN 変数によって決まる 1322
- スケジューラー
  - DB2 administration server (DAS) 327
- スケジューリング
  - ウェアハウス・ステップ
    - トリガー・プログラムで 204
    - サブスクリプション・セット 687, 698, 699
- ステージング、データの 710
- ステージングされたレプリケーション 687, 711
- ステージング表
  - データ 687
- ステップ
  - 印刷 1081
  - 追加 1081
  - データウェアハウス・センター外部からの開始 1085
  - SAP 1081
  - Web トラフィック 1081
- ステップ・タスク・フロー
  - 説明 1084
- ストアード・プロシージャ
  - 作成、構文 1672, 1686
  - ストレージ管理ツールに対する 377
  - 操作、データの 721, 722
  - 定義、サブスクリプション・セットの 687, 698
  - ファイルのビルド 769
- ストリング
  - 割り当て変換規則 1364
- ストリング引き数
  - 長さ 860
  - 入力 860
- ストレージ管理スナップショット 305
- ストレージ管理ツール
  - ストアード・プロシージャ 377
  - ストレージ管理ビュー 305
- ストレージ管理ビュー
  - 表 377
- ストレージ構造
  - ALTER TABLESPACE ステートメント 1434
  - CREATE TABLESPACE ステートメント 1562
- スナップショット
  - ストレージ 305
- スレッド
  - スレッド・アプリケーション 1022
- スレッド (続き)
  - マルチ
    - DB2 アプリケーションでの使用 742
  - スレッド間にコンテキストを渡す 742
- 制限
  - 異機種のレプリケーション 645
  - ストアード・プロシージャ 723
  - 非 DB2 リレーショナル・データ・ソース 650
  - 列名の制限 646
  - Microsoft SQL Server 645
  - Sybase 645
- 整合変更データ (CCD) 表
  - 外部
    - 概要 687
    - multi-tier レプリケーション 687, 710
  - 構造
    - ターゲット・サーバー 731
  - コンデンス 687
  - コンプリート 687
- 使用法
  - 履歴または監査 687, 708
  - multi-tier レプリケーション 687, 710
  - 追加、UOW 列の 687, 708
- 内部
  - 概要 687
  - 複数のターゲット 687, 708
  - 非 DB2 リレーショナル・データ・ソース 637
  - 使用、CCD 表の 640
  - 非コンデンス 687
  - 非コンプリート 687
  - 非リレーショナル・データ・ソース
    - 使用、CCD 表の 637, 638
    - CCD 表の保守 637, 662
    - レプリケーション・ソース 687, 710
- 整数
  - 10 進数への変換の要約 1364

## 生成列

CREATE TABLE ステートメント  
1484

静的なコントロール表 617, 626

## 精度

数値、SQLLEN 変数によって決ま  
る 1322

## 制約

非 DB2 リレーショナル・デー  
タ・ソース 655

ALTER TABLE による追加  
1402

ALTER TABLE によるドロップ  
1402

DATALINK 値 655

DB2 Enterprise Server  
Edition 643

## 制約事項

異機種のレプリケーション 637,  
687, 710, 713

既存のターゲット表 687, 715  
ストアド・プロシージャ  
721

地理情報データ・タイプ 603,  
604

データ圧縮 603

データ暗号化 603

データ・タイプ 603, 604

特殊データ・タイプ 603, 604

非 DB2 リレーショナル・デー  
タ・ソース 637

ビュー 637, 661

ユーザー定義のデータ・タイプ  
603, 604

ユーティリティ・プログラム  
603

要約データ・タイプ 603, 604

列名の制限 637

CCD 表 687, 713

DATALINK 値 637, 687, 713

DB2 Enterprise Server  
Edition 637

DB2 エクステンダー・ラージ・  
オブジェクト 603, 605

EDITPROC 文節 603, 604

FIELDPROC 文節 603, 604

## 制約事項 (続き)

LOB データ・タイプ 687, 713

LONG VARGRAPHIC データ・タ  
イプ 603, 604

Microsoft SQL Server 637

Oracle ソース 603, 604

Oracle 表内の LONG 列 603,  
604

Sybase 637

VALIDPROC 文節 603, 604

WHERE 文節 687, 717

z/OS 603

## 接続

複数 827

CLI 関数データ・ソースへ  
888

iSeries サーバーへの 663, 665

z/OS サーバーへの 663, 665

## 設定

アプライ・プログラム

OS/400 663

Windows 663

環境変数、キャプチャー・プログ  
ラム 663

キャプチャー・プログラム

OS/400 663

Windows 663

単一記述子フィールド CLI 関数  
910

## 設定、環境変数

キャプチャー・プログラム 676

## 接頭部

変更前イメージ 637

接頭部、変更前イメージ 648

## セットアップ

アプライ・プログラム

OS/400 版 680

UNIX 版 676

Windows 版 676

キャプチャー・プログラム

OS/400 版 680

UNIX 版 676

Windows 版 676

ジャーナル 682

レプリケーション・アラート・モ  
ニター 679

## 全選択

初期化 1317

反復 1317

CREATE VIEW ステートメント  
1469

## ソース

サブスクリプト 687, 690

システム、保守 617

## 登録

行 637

非 DB2 リレーショナル 637,  
640

ビュー 637, 658, 661

列 637

DB2 表 637, 638

IMS データ・ソース 637,  
638

登録、行の 644

登録、列の 642

登録オプション

エラー発生時にキャプチャー  
を停止 637, 648

行 (水平方向) のサブセット化  
637, 644

競合検出 637, 655

更新、削除および挿入として  
637, 649

再キャプチャー、変更の  
(update-anywhere) 650

使用、リモート・ジャーナル  
637, 657

相対レコード番号 637, 658  
フル・リフレッシュ・コピー  
637, 643

変更キャプチャー・レプリケ  
ーション 637, 643

変更後イメージ列 637, 645

変更前イメージ接頭部 637,  
648

変更前イメージ列 637, 645  
変更の再キャプチャー

(update-anywhere) 637

列 (垂直方向) のサブセット化  
637, 642

マッピング、ターゲットへの  
687, 701

ソース (続き)  
ログ、保守 617  
CCD (整合変更データ) 表 687,  
710  
CCD 表、保守 637  
CCD 表の保守 662  
ソース表  
作成、ジャーナルの 683  
ジャーナルの作成 663  
脱落したデータのリトリブ  
617, 632  
保守 617, 618  
ソース・システム、保守 618  
ソース・ログ、保守 618  
ソート  
結果の順序付け 1364  
ストリングの比較 1364  
ゾーン 10 進数ファイル・タイプ修  
飾子 945, 1149  
相関 ID 637, 659  
操作  
キャプチャー・プログラム 526  
比較 1364  
割り当て 1364  
操作、データの  
作成、算出列の 721, 723  
サブスクリプションでの 721,  
722  
登録での 721  
名前変更、列の 721, 723  
相対タイミング 687, 699  
相対レコード番号  
サポート、OS/400 に対する 658  
主キーとしての、OS/400 の場合  
658  
使用される、ターゲットとして  
687, 719  
OS400 でのサポート 637  
OS400 での主キーとして 637  
挿入可能ビュー 1469  
双方向 CCSID 270  
ソフトウェア要件  
ウェアハウス・エージェント  
194

## [タ行]

ターゲット  
キー 687  
索引 687  
ターゲット索引 718  
ターゲット表  
新しい列 723  
基礎集約  
使用法 687, 707  
説明 687  
定義 704  
サブセットの適用  
行 687  
列 687  
新規列 721  
定義  
行 687  
ターゲット・キー 687  
列 687  
定義、行の 716  
定義、ターゲット・キーの 718  
定義、列の 716  
適用、行のサブセット 716  
適用、列のサブセット 716  
フラグメント化 687, 716  
変更集約  
使用法 687, 707  
説明 687  
定義 704  
ポイント・イン・タイム  
使用法 687, 706  
説明 687  
定義 704  
保守 617, 632  
マッピング、ソースへの 687,  
701  
ユーザー定義 687, 706, 715  
ユーザー・コピー  
使用法 687, 706  
説明 687  
定義 704  
レプリカ  
使用法 687, 712  
説明 687  
定義 704

ターゲット表 (続き)  
CCD (整合変更データ) 687, 731  
概要 704  
構造 731  
ターゲット・キー 718  
ターゲット・キー列  
更新 720  
ターゲット・キー列、更新 687  
代理文字  
ユニコード 273  
タイプ 2 JDBC ドライバー  
機能強化 90  
タイプ 3 JDBC ドライバー 90  
タイプ 4 JDBC ドライバー  
DB2 サポート 90  
タイミング 687  
タイム・スタンプ・モニター・ス  
ッチ 476  
大量のレプリケーション・ジョブ  
687, 694  
タグ言語ファイル  
インポート  
コマンド行からの 1096  
記述データ・タイプ 1102  
多次元表 279  
データの移動 279  
列式を次元として使用 279  
SMS 表スペース内 279  
単精度浮動データ・タイプ 1484  
単調 279  
チェック制約  
ALTER TABLE ステートメント  
1402  
CREATE TABLE ステートメント  
1484  
チュートリアル 227  
地理情報データ  
データ・タイプ 603  
地理情報データ・タイプ 604  
ツール  
カタログ・データベース 327  
追加  
データベース 94  
データ  
検索  
ソース表から 617



- データ (続き)
  - ソース表からのリトリブ 632
  - 操作 721
  - トランスフォーム 721
    - 作成、算出列の 723
    - サブスクリプションでの 722
    - 登録での 721
    - 名前変更、列の 723
    - 防止、二重削除の 637, 661
  - データ圧縮の制約事項 603
  - データ暗号化
    - 制約事項 603
  - データ暗号化の制約事項 603
  - データウェアハウス・センター
    - 外側からのステップの開始 1085
  - データウェアハウス・センター管理
    - インターフェース
      - 開始
        - AIX 1095
  - データ共有モード 613, 617
  - データ構造
    - バック 10 進数 1322
  - データ整合性 687, 715
  - データの移動
    - 多次元表へ 279
    - データベース間 1234
  - データベース
    - アクセス
      - マルチスレッド 742
    - 移行 394
    - 構成 117
    - 再編成 1222
    - 作成 926
    - 照合情報 363
    - 統計 1258
    - 表のファイルへのエクスポート
      - 1134
    - ファイルの表へのインポート
      - 1234
    - ファイルの表へのロード 1149
    - 別のコンテキストを使用 742
    - リストア (再構築) 1267
    - CREATE TABLESPACE ステートメント 1562
  - データベース検査 API 938
- データベース構成の更新コマンド 663, 677
- データベース作成 API 926
- データベース接続
  - 構成
    - 構成アシスタント (CA) を使用した 94
    - ディスクバリーを使用した 154
    - テスト 117
  - データベース接続をしないログの読み取り API 1024
  - データベース接続をしないログの読み取りの終了 API 420
  - データベース接続をしないログの読み取りの初期化 API 1005
  - データベースの構成
    - 更新 1276
    - ネットワーク・パラメーター値 1276
  - データベースの再配置 1305
  - データベース・ディレクトリー
    - 記述構造 276
  - データベース・マネージャー
    - 開始 1279
  - データ変換
    - C から SQL データ・タイプへ 853
    - SQL から C データ・タイプへ 845
  - データ・ソース
    - サポートされるバージョン 1398
  - 接続
    - CLI 関数 888
    - 説明 1398
    - データの削除 475
    - データ・タイプ・マッピング 456
  - データ・タイプ
    - インフォメーション・カタログ・センターによる使用 1102
    - 行 1441
    - 構造化された 1441
    - サポートされない 448
    - 制約事項 603, 604
    - マッピング、列間の 687, 718
- データ・タイプ (続き)
  - 要約 1441
  - レプリケーション
    - ラージ・オブジェクト (LOB) 603, 604
    - DATALINK 値 603, 605
  - BIGINT (G) 1102
  - BLOB (B) 1102
  - CHAR (C) 1102
  - CLOB (O) 1102
  - CREATE TYPE (構造化) ステートメント 1441
  - DATALINK 1363
  - DATE (D) 1102
  - DECIMAL (E) 1102
  - DOUBLE (U) 1102
  - INTEGER 1102
  - Java 782
  - LONG VARCHAR (L) 1102
  - REAL (R) 1102
  - SMALLINT (S) 1102
  - TIME (M) 1102
  - TIMESTAMP (T) 1102
  - VARCHAR (V) 1102
  - データ・タイプ・マッピング
    - 逆方向 427, 456
    - サポートされないデータ・タイプ 456
    - 順方向 456, 1381
    - 説明 448
    - 変更 456
  - データ・ブロッキング 687, 694
  - データ・リンク
    - レプリケーション 605
  - データ・リンク・ファイル・マネージャー (DLFM)
    - エラー・メッセージ 1033
  - 定義
    - ステップ 1081
  - 停止
    - アプライ・プログラム
      - OS/400 478
      - OS/400 版 489
      - UNIX 478
      - UNIX 版 489
      - Windows 478



停止 (続き)

アプライ・プログラム (続き)

Windows 版 489

z/OS 478

z/OS 版 489

キャプチャー・プログラム

UNIX 版 526

Windows 版 526

z/OS 版 526

ディスクバリー機能

データベース接続の構成 154

出口ルーチン

ジャーナル・レシーバー削除

(OS/400) 663, 686

ASNDLCOPY 603, 608

ASNDONE

使用 478, 490, 491

ASNLOAD

カスタマイズ 478, 497

使用 478, 493

OS400 版 478

OS/400 版 498

UNIX 版 478, 494

Windows 版 478, 494

z/OS 版 478, 495

デフォルト

アプライ・パラメーターの

(UNIX、Windows、z/OS) 479

アプライ・プログラムのパラメー

ターの (OS/400) 489

デフォルト値

アプライ・パラメーターの

(UNIX、Windows、z/OS) 478

アプライ・プログラムのパラメー

ターの (OS/400) 478

列

ALTER TABLE ステートメン  
ト 1402

CREATE TABLE ステートメ  
ント 1484

テリトリリー・コード

DB2 のサポート 242

透過 DDL

データ・タイプ・マッピング

456

リモート表の作成 450

同義語

DROP ALIAS ステートメント  
1572

統計

索引の再編成 1222

データベース 1258

REORGCHK 1222

動的 SQL

SQLDA の使用 1322

動的なコントロール表 617, 623

登録

オプション、ソース用の

エラー発生時にキャプチャー

を停止 637, 648

行 (水平方向) のサブセット化

637, 644

競合検出 637, 655

更新、削除および挿入として

637, 649

再キャプチャー、変更の

(update-anywhere) 650

使用、リモート・ジャーナル

637, 657

相対レコード番号 637, 658

フル・リフレッシュ・コピー

637, 643

変更キャプチャー・レプリケ

ーション 637, 643

変更後イメージ列 637, 645

変更前イメージ接頭部 637,

648

変更前イメージ列 637, 645

変更の再キャプチャー

(update-anywhere) 637

列 (垂直方向) のサブセット化

637, 642

削除 602

除去 602

追加 570

非 DB2 リレーショナル・デー

タ・ソース 637, 640

ビュー

概要 637, 658, 661

DB2 表 637, 638

IMS データ・ソース 637, 638

ドキュメンテーション

表示 1958

特殊タイプ 603

比較 1364

DROP ステートメント 1572

特殊データ・タイプ 604

特殊なデータ・タイプ

ラージ・オブジェクト (LOB)、複

製 603

レプリケーション

ラージ・オブジェクト

(LOB) 604

DATALINK 値 605

DATALINK 値、複製 603

独立補助記憶域プール (IASP) グルー

プ 663, 674

特権

管理者グループ、Windows 143

データベース

作成中の付与 926

トラブルシューティング

asnlrc コマンド 549

Spatial Extender

サンプル・プログラム 1106

runGseDemo の使用 1106

トラブルシューティング・コマンド

asnlrc 549

トランザクション

処理 687

トランザクション・モード処理 696

トランスフォーマー

設定 1081

トランスフォーム、データの

作成、算出列の 723

サブスクリプションでの 722

登録での 721

名前変更、列の 723

トランスフォーメーション

作成、算出列の 721

サブスクリプションでの 721

登録での 721

名前変更、列の 721

DROP ステートメント 1572

トリガー

ドロップ 1572

トレース  
UNIX 549  
Windows 549  
z/OS 549  
トレース機能  
UNIX 版 549  
Windows 版 549  
z/OS 版 549  
ドロップされた表のリカバリー 410

## [ナ行]

内部 CCD 表  
概要 687  
複数のターゲット 687, 708  
内部結合、ソースとしての 637, 659  
名前  
アプライ修飾子の規則 633  
キャプチャー・スキーマの規則 633  
表示名 634, 636  
モニター修飾子の規則 633  
レプリケーション・サービスの 634, 635  
Windows サービス用 634  
名前付きパイプ  
構成  
CLP を使用した 155  
名前変更  
列 721  
名前変更、列の 723  
二重削除 637, 661  
ニックネーム  
クロスローダー・ユーティリティ  
の 478, 498  
制限  
CCD 表の場合 645  
制約事項  
集約表 687, 707  
CCD 表の場合 637  
multi-tier レプリケーション  
687, 710  
update-anywhere 637, 650,  
687, 713  
登録 637, 640  
Microsoft SQL Server、作成 459

入出力エラーのリカバー、コントロ  
ール表 617, 631  
認証  
エンド・ユーザー 663  
認証、エンド・ユーザー  
UNIX 版 664, 672  
Windows 版 664, 672  
ヌル終了ストリング  
CLI アプリケーション内 860  
ネットワーク  
コネクティビティ 663  
ネットワーク・コネクティビティ  
664  
ノード  
説明 445  
ノード・レベル・プロファイル・レ  
ジストリー 308

## [ハ行]

バージョン・レベル  
Data Links Manager と DB2 サー  
バー 143  
倍精度浮動データ・タイプ 1484  
バイナリー・ラージ・オブジェクト  
(BLOB)  
レプリケーションの考慮事項  
603, 604  
バインディング  
アプライ・プログラム  
UNIX 版 663  
Windows 版 663  
z/OS 版 663  
アプリケーション変数  
CLI 関数 917  
暗黙的に作成されたスキーマ  
1190  
エラー 926  
キャプチャー・プログラム  
UNIX 版 663  
Windows 版 663  
z/OS 版 663  
ユーティリティ 119  
列の配列  
CLI 関数 917

バインディング (続き)  
レプリケーション・アラート・モ  
ニター  
UNIX 版 663  
Windows 版 663  
バインド  
アプライ・プログラム  
UNIX 版 678  
Windows 版 678  
z/OS 版 682  
キャプチャー・プログラム  
UNIX 版 677  
Windows 版 677  
z/OS 版 682  
レプリケーション・アラート・モ  
ニター  
UNIX 版 679  
Windows 版 679  
バインド・ファイル  
およびパッケージ名 868  
パスルー  
考慮事項、制約事項 468  
COMMIT ステートメント 468  
SET PASSTHRU RESET ステ  
ートメント 468  
SET PASSTHRU ステートメント  
468  
パスワード・ファイル  
保管 663, 672  
バックアップ  
イメージ 399  
コンテナ名 399  
バックアップ・コマンド 663  
バックアップ・ユーティリティ  
概要 399  
情報の表示 399  
トラブルシューティング 399  
パッケージ  
再バインド 617  
DROP FOREIGN KEY、従属関係  
への影響 1402  
DROP PRIMARY KEY、従属関係  
への影響 1402  
DROP UNIQUE キー、従属関係  
への影響 1402

- パッケージ (続き)
  - DROP ステートメントを使用した削除 1572
- パッケージ、再バインド 624
- パッケージ名
  - およびバインディング 868
- 発行
  - ヒープ・サイズの拡大 1080
- バッチ・ジョブ
  - 実行 613, 614
- バッチ・ファイル
  - 説明、~の 769
- バッファ・プール
  - DROP ステートメントを使用した削除 1572
- パフォーマンス
  - 改善
    - REORGCHK コマンド 1222
    - 区分化キーの推奨事項 1484
- パラメーター、呼び出し
  - アナライザー
    - UNIX 版 518
    - Windows 版 518
  - アプライ・プログラム
    - OS/400 版 478, 488
    - UNIX 版 478, 479, 533, 534
    - Windows 版 478, 479, 533, 534
    - z/OS 版 478, 479, 533, 534
  - キャプチャー・プログラム
    - UNIX 版 542, 543
    - Windows 版 542, 543
    - z/OS 版 542, 543
- パラメーター・マーカー
  - EXECUTE ステートメント 1602
- 反復全選択 1317
- 非 DB2 リレーショナル・データ・ソース
  - 使用、CCD 表の 637, 640
  - 制限
    - update-anywhere 650
  - 制約事項
    - 集約表 687, 707
    - multi-tier レプリケーション 687, 710
- 非 DB2 リレーショナル・データ・ソース (続き)
  - 制約事項 (続き)
    - update-anywhere 637, 655, 687, 713
    - 登録 637, 640
  - ヒープ・サイズ
    - データウェアハウス・センター・メタデータの発行用に拡大 1080
  - 非アクティブなサブスクリプション・セット 687, 693
  - 比較、基本 SQL 操作 1364
  - 非活動化
    - サブスクリプション・セット 687, 693
  - 非互換性
    - バージョン 8 282
  - 非コンデンス CCD 表 687
  - 非コンプリート CCD 表 687
  - 非同期のイベント 742
  - ビュー
    - 更新可能 1469
    - 削除可能 1469
    - 作成 1469
    - 作動不能 1469
    - 使用、相関 ID の 637, 659
    - 制約事項 637, 658, 661
    - 挿入可能 1469
    - 登録
      - 概要 637, 658
      - ソースとして 637, 661
    - 別名 1572
    - 読み取り専用 1469
    - 列名 1469
  - DROP ステートメントを使用した削除 1572
- 表
  - 結合
    - 区分化キーの考慮事項 1484
  - コントロール表
    - 再編成 617, 625
    - 作成 663, 673
    - 静的 617, 626
    - 接続障害のリカバー 617, 631
    - 動的 617, 623
- 表 (続き)
  - コントロール表 (続き)
    - 入出力エラーのリカバー 617, 631
    - ブルーニング 617, 626
    - 保守 617, 623
    - RUNSTATS ユーティリティ 617, 624
  - 再編成
    - 必要かどうかの判断 1222
  - 削除
    - DROP ステートメントを使用した 1572
  - 作成
    - SQL ステートメント命令 1484
  - 作成の許可 1484
  - 生成列 1402
  - ターゲット表 617
    - 保守 632
    - 参照：ターゲット表
  - 追加
    - 列、ALTER TABLE 1402
  - 統計
    - 説明 1258
  - 登録
    - 非 DB2 リレーショナル 637, 640
    - DB2 637, 638
  - 名前
    - ALTER TABLE ステートメントの 1402
    - ファイルのインポート 1234
    - ファイルのロード 1149
    - ファイルへのエクスポート 1134
    - 別名 1572
    - 変更 1402
  - CAPENQ (キャプチャー・エンキュー) 517
  - CAPMON (キャプチャー・モニター) 617, 629
  - CAPTRACE (キャプチャー・トレース) 521, 617, 629
  - CCD (整合変更データ)
    - ターゲット・サーバー 731
  - CCD 表の保守 637, 662

表 (続き)

IDENTITY 列 336  
 SIGNAL (シグナル) 522  
 SUBS\_COLS (サブスクリプション列) 500  
 SUBS\_MEMBR (サブスクリプション・メンバー) 478, 497, 502  
 SUBS\_SET (サブスクリプション・セット) 507

表構造ファイル 1398

標識機能、SQL 92 および MVS 適合 757

表式

共通表式 1317

表示名 634, 636

表スペース

索引、CREATE TABLE ステートメント 1484

作成

CREATE TABLESPACE ステートメント 1562

識別、CREATE TABLE ステートメント 1484

ドロップ

DROP ステートメント 1572  
 ページ・サイズ 1562

DROP ステートメントを使用した削除 1572

表名、CREATE TABLE ステートメントの 1484

表モード処理 687, 696

非リレーショナル・データ・ソース使用、CCD 表の 637, 638  
 データ・タイプ・マッピングの指定 448

CCD 表の保守 637, 662

ヒント

アプライ・トレール表からの行の削除 478, 488

アプライ・プログラムがセットを正常に処理したかのチェック 478, 487

サービスが正しくセットアップされたことの検証 634, 635

ヒント (続き)

スリーブを使用するか copyonce  
 パラメーターを使用するか 478, 482, 485

セットの追加処理にストアード・プロシージャを使用 478, 490

ASNDONE でストアード・プロシージャを使用 478, 491

ファイル

asndone.smp 478, 491

asnload.ini 478, 498

IWH.environment 228

\*.APP.log 478, 483

\*.err 478, 486

\*.sqsq 478, 486, 487

ファイルのビルド

説明、~の 769

ファイル・コピー・デーモン

ASNDLCOPYD 603, 611

DLFM\_ASNCOPYD 603, 610

ファイル・システム

ディレクトリー階層を現時点まで移動する 1031

ファイル・システム・ディレクトリー階層 1031

ファイル・タイプ修飾子

エクスポート・ユーティリティー 1134

ロード API 945

IMPORT コマンド 1234

LOAD コマンド 1149

ファイル・フォーマット

表のファイルへのエクスポート 1134

ファイルの表へのインポート 1234

フォーマット検査結果コマンド

1218

複数インスタンス

Tivoli Storage Manager での使用 407

複数のターゲット表 687, 708

浮動小数点から 10 進数への変換 1364

プラグイン

ガイドライン 339

コンパイル 311

実行 311

フラグメント化

垂直

ソースでの 637, 642

ターゲットでの 687, 716

水平

ソースでの 637, 644

ターゲットでの 687, 716

プラン

競合検出 637, 655

プラン、再バインド 617, 624

プルーニング

アプライ・トレース

(APPLYTRACE) 表 617, 630

アプライ・トレール

(APPLYTRAIL) 表 617, 630

キャプチャー・トレース

(CAPTRACE) 表 617, 629

キャプチャー・プログラム

UNIX 526

UNIX 版 526

Windows 526

Windows 版 526

z/OS 526

z/OS 版 526

キャプチャー・モニター

(CAPMON) 表 617, 629

コントロール表 617, 626

シグナル (SIGNAL) 表 617, 629

CD (変更データ) 表 617, 628

UOW (作業単位) 表 617, 628

フル・リフレッシュのカスケード 687

フル・リフレッシュ・コピー

登録オプション 637, 643

Apply for iSeries 658

CCD 表内 687

iSeries のアプライ 637

プロシージャ

作成、構文 1672, 1686

作成の許可 1672, 1686

ブロッキング因数 687, 694

プロファイル・レジストリー 308

分離レベル  
DELETE ステートメント 1317  
別名  
DROP ステートメントを使用した  
削除 1572  
ヘルプ  
表示 1958  
変換  
数値の位取りと精度のまとめ  
1364  
日時からストリング変数 1364  
変換規則  
比較 1364  
割り当て 1364  
変更  
キャプチャー・パラメーター  
UNIX 526  
Windows 526  
z/OS 526  
変更キャプチャー・レプリケーシ  
ョン  
説明 637, 644  
登録オプション 637, 643  
変更後イメージ列 637, 645  
変更集約表  
使用法 687, 707  
定義 687, 704  
変更前イメージ接頭部 637, 648  
変更前イメージ列  
制限 646  
制約事項 637  
登録 637, 645  
変更集約表 687, 717  
変更データ (CD) 表  
ブルーニング 617, 628  
要約、内容の 687, 707  
変更の再キャプチャー  
(update-anywhere) 637  
変数  
DB2CODEPAGE 190  
DB2INSTANCE 190  
LANG 190  
LC\_ALL 190  
ポート番号  
TCP/IP  
DLFM (AIX) 134

ポート番号 (続き)  
TCP/IP (続き)  
DLFM (Solaris オペレーティ  
ング環境) 124  
DLFM (Windows) 143  
ポイント・イン・タイム表  
使用法 687, 706  
ホスト変数  
パラメーター・マーカの置換  
1602  
ホスト名 143  
ホスト・アプリケーション  
DB2 Universal Database サーバー  
へのアクセス 157

## [マ行]

マスター表 (update-anywhere)  
概要 687, 712  
再キャプチャー、変更の 637,  
650  
マッピング  
ソースからターゲット 687, 701  
ソース列からターゲット列への  
687, 718  
データ・タイプ、表間の 687,  
718  
マテリアライズ照会表  
定義 1484  
マルチサイト更新  
コントロール・センター 93  
マルチサイト更新ウィザード 93  
マルチスレッド・アプリケーション  
ファイルのビルド 769  
ミニサイクル 687, 694  
ミラーリングされたデータベースの  
初期化コマンド 1315  
メタデータ  
データウェアハウス・センターか  
らの発行 1094  
メッセージ・キュー、ジャーナル用  
663, 686  
メモリー  
applheapsz 構成パラメーター  
340

メモリー要件  
Data Links Manager  
AIX 134  
Solaris オペレーティング環境  
124  
Windows 143  
文字ストリング  
解釈 860  
等価、照合順序の例 1364  
等価の定義 1364  
長さ 860  
比較 1364  
割り当て 1364  
文字変換  
比較に関する規則 1364  
割り当てに関する規則 1364  
文字ラージ・オブジェクト (CLOB)  
レプリケーションの考慮事項  
604  
モニター修飾子、命名規則 633  
モニター・コントロール表  
MONTRAIL (モニター・トレ  
ール) 729  
モニター・スイッチ  
連合 476  
モニター・トレール (MONTRAIL)  
表 729  
問題判別  
サテライト・コントロール・サー  
バーのインストール 398  
問題分析および環境収集ツール・コ  
マンド 1310

## [ヤ行]

ユーザー ID  
アプライ・プログラム 663, 670  
キャプチャー・トリガー 663  
キャプチャー・トリガーの場合  
670  
キャプチャー・プログラム 663  
キャプチャー・プログラムの場合  
669  
許可 663, 669  
パスワード・ファイル 663, 672

- ユーザー ID (続き)
    - レプリケーション・アラート・モニター 663
    - レプリケーション・アラート・モニター用 672
  - ユーザー定義関数 (UDF)
    - ファイルのビルド 769
  - CREATE FUNCTION (SQL スカラー、表、または行) ステートメント 1552
  - CREATE FUNCTION (外部スカラー) ステートメント 1618
  - CREATE FUNCTION (外部表) ステートメント 1644
  - CREATE FUNCTION (ソースまたはテンプレート) ステートメント 1662
  - DROP ステートメント 1572
  - ユーザー定義タイプ (UDT)
    - 構造タイプ 1484
    - サポートされないデータ・タイプ 448
    - 特殊データ・タイプ、CREATE TABLE ステートメント 1484
    - レプリケーション 603
  - ユーザー定義の一時表
    - 作成 326
  - ユーザー定義のデータ・タイプ 604
  - ユーザー定義のデータ・タイプ・マッピング 456
  - ユーザー定義のプログラム
    - z/OS 203
      - ステップ完了 232
  - ユーザー定義表 687, 706, 715
  - ユーザー・アカウント
    - 権限、Windows 143
      - Windows の DB2 サーバーのインストールに必要 392
  - ユーザー・コピー表
    - 使用法 687, 706
    - 定義 687, 704
  - ユーティリティ
    - バインディング 119
    - プログラムの制約事項 603
  - ユーロ通貨記号
    - サポートの使用不可 233
  - ユーロ通貨記号のサポートを使用不可にする 233
  - 有効範囲
    - 追加された列による定義 1402
  - ALTER TABLE ステートメントによる追加 1402
  - CREATE TABLE ステートメントによる定義 1484
  - CREATE VIEW ステートメント 1469
  - ユニーク制約
    - ALTER TABLE ステートメント 1402
    - ALTER TABLE による追加 1402
    - ALTER TABLE によるドロップ 1402
    - CREATE TABLE ステートメント 1484
  - ユニーク・キー
    - ALTER TABLE ステートメント 1402
    - CREATE TABLE ステートメント 1484
  - ユニコード
    - コード・ページ 1394 の変換
      - 前の変換表 304
      - Shift JIS X0213 の変換
        - 前の変換表 304
    - 要約データ・タイプ 603, 604
    - 要約表
      - 定義 1484
    - 呼び出しパラメーター
      - アナライザー
        - UNIX 版 518
        - Windows 版 518
      - アプリ・プログラム
        - OS/400 版 478, 488
        - UNIX 版 478, 479, 533, 534
        - Windows 版 478, 479, 533, 534
        - z/OS 版 478, 479, 533, 534
      - キャプチャー・プログラム
        - UNIX 版 542, 543
        - Windows 版 542, 543
        - z/OS 版 542, 543
  - 読み取り従属関係 637, 657
  - 読み取り専用ビュー 1469
- ## [ラ行]
- ラージ・オブジェクト (LOB)
    - レプリケーションの考慮事項 603, 604
  - ライセンス・センター
    - インストール
      - DB2 Connect ライセンス・キー 121
      - DB2 ライセンス・キー 187
    - ファイル、DB2 製品 184
    - ライセンス・タイプの設定
      - ライセンス・センターの使用 123
      - db2licm コマンドを使用した 185
  - ラッチ、マルチスレッドによる状況 742
  - ラッパー
    - Microsoft SQL Server、作成 459
  - ランタイム・サービス
    - マルチスレッド
      - ラッチへの影響 742
  - リカバリ
    - データベース 1267
    - ドロップされた表 410
    - ドロップされた表、ロールフォワード・ユーティリティ 410
    - ロールフォワードなし 1267
  - リストア
    - 旧バージョンの DB2 データベース 1267
  - リモート
    - ソースとしてのジャーナル 637
    - ソース表 637
    - データ・タイプ 456
    - 表、作成 450
  - リモート・ジャーナル、ソースとして 657
  - リモート・ソース表 657
  - 履歴データ
    - ソース・データ 637, 647
    - CCD 表 687, 708

- ルーチン
  - 使用可能な SQL ステートメント 1360
  - ファイルのビルド 769
- 列
  - 計算 687, 716
  - サブセット化
    - ソースでの 637, 642
    - ターゲットでの 687, 716
  - 算出 721, 723
  - 使用可能、レプリケーションに 637, 642
  - ストリング割り当ての規則 1364
  - 制約名、FOREIGN KEY、規則 1484
  - 相対レコード番号、OS/400 での 658
  - 定義、ターゲット表での 687, 716
  - 登録、ソース表内の 637, 642
  - 名前変更 721, 723
  - 表への追加、ALTER TABLE 1402
  - 変更後イメージ 637, 645
  - 変更前イメージ 637, 645
  - マッピング、ソースからターゲットへの 687, 718
  - ALTER TABLE ステートメントによる追加 1402
  - iSeries での相対レコード番号 637
  - NULL 値、ALTER TABLE、回避 1402
- 列 (垂直方向) のサブセット化
  - ソースでの 637, 642
  - ターゲットでの 687, 716
- 列オプション
  - CREATE TABLE ステートメント 1484
- 列式
  - 多次元表 279
- 列属性 874
- 列属性を戻す 874
- 例
  - 外部トリガー・プログラム 1086
- レジストリー
  - AIX、変数 134
  - Solaris、変数 124
  - Windows
    - 変数 143
- レジストリー変数
  - 環境変数 308
  - DB2ACCOUNT 366
  - DB2BIDI 366
  - DB2BPVARS 346
  - DB2CHKPTR 346
  - DB2CODEPAGE 366, 663, 677
  - DB2CONNECTIN\_APP\_PROCESS 343
  - DB2DBDFT 366, 663, 677
  - DB2DBMSADDR 366
  - DB2DISABLE\_FLUSH\_LOG 366
  - DB2DISCOVERYTIME 366
  - DB2DOMAINLIST 343
  - DB2ENVLIST 343
  - DB2INCLUDE 366
  - DB2INSTANCE 343, 663, 676
  - DB2INSTDEF 366
  - DB2INSTOWNER 366
  - DB2INSTPROF 343
  - DB2LIBPATH 343
  - DB2LIC\_STAT\_SIZE 366
  - DB2LOCALE 366
  - DB2MAXFSCRSEARCH 346
  - DB2MEMDISCLAIM 346
  - DB2MEMMAXFREE 346
  - DB2NBDISCOVERRCVBUFS 366
  - DB2NODE 343
  - DB2NTMEMSIZE 346
  - DB2NTNOCACHE 346
  - DB2NTPRICLASS 346
  - DB2NTWORKSET 346
  - DB2OPTIONS 366
  - DB2PARALLEL\_IO 343
  - DB2PATH 343
  - DB2PRIORITIES 346
  - DB2SLOGON 366
  - DB2TERRITORY 366
  - DB2TIMEOUT 366
  - DB2TRACEFLUSH 366
  - DB2TRACENAME 366
- レジストリー変数 (続き)
  - DB2TRACEON 366
  - DB2TRCSYSERR 366
  - DB2USE\_PAGE\_CONTAINER\_TAG 343
  - DB2YIELD 366
  - DB2\_ANTIJOIN 371
  - DB2\_APM\_PERFORMANCE 346
  - DB2\_AVOID\_PREFETCH 346
  - DB2\_AWE 346
  - DB2\_BINSORT 346
  - DB2\_CORRELATED\_PREDICATES 371
  - DB2\_DARI\_LOOKUP\_ALL 346
  - DB2\_ENABLE\_BUFDPD 346
  - DB2\_EXTENDED\_OPTIMIZATION 346
  - DB2\_HASH\_JOIN 371
  - DB2\_INLIST\_TO\_NLJN 371
  - DB2\_LIKE\_VARCHAR 371
  - DB2\_MINIMIZE\_LIST\_PREFETCH 371
  - DB2\_MMAP\_READ 346
  - DB2\_MMAP\_WRITE 346
  - DB2\_NEW\_CORR\_SQ\_FF 371
  - DB2\_OVERRIDE\_BPF 346
  - DB2\_PINNED\_BP 346
  - DB2\_PRED\_FACTORIZE 371
  - DB2\_REDUCED\_OPTIMIZATION 371
  - DB2\_SELECTIVITY 371
  - DB2\_SORT\_AFTER\_TQ 346
  - DB2\_STPROC\_LOOKUP\_FIRST 346
- レプリカ表
  - 再キャプチャー、変更の 650
  - 定義 704
  - 定義、読み取り / 書き込みターゲットの 712
- レプリカ表 (update-anywhere)
  - 概要 687
  - 再キャプチャー、変更の 637
  - 定義 687
- レプリケーション・アナライザー
  - OS/400
    - 作成、SQL パッケージの 663



レプリケーション・アナライザ (続  
き)

OS/400 版  
作成、SQL パッケージの  
681  
UNIX 版、呼び出しパラメーター  
518  
Windows 版、呼び出しパラメータ  
ー 518

レプリケーション・アラート・モニ  
ター

許可要件 663, 672  
UNIX  
バインディング 663  
UNIX 版  
バインド 679  
Windows  
バインディング 663  
Windows 版  
バインド 679

z/OS  
開始 613

z/OS 版  
開始 613

レプリケーション・コマンド

データベース構成の更新 663,  
677  
データベースのバックアップ  
663, 677  
ASNL2RNx 613  
CRTJRNRCV 663, 683  
MODIFY 613  
OS/400

ADDDPRREG 570  
ADDDPRSUB 580  
ADDDPRSUBM 557  
ADDEXITPGM 663  
ANZDPRJRN 663  
CHGJRN 663  
CRTJRN 663  
GRTPRAUT 663  
GRTOBJAUT 663  
RCVJRNE 663  
RMVDPRREG 602  
RMVEXITPGM 663  
RVKDPRAUT 600

レプリケーション・コマンド (続き)

OS/400 (続き)

STRJRNPf 663  
WRKREGINF 663  
OS/400 版  
ADDDPRREG 570  
ADDDPRSUB 580  
ADDDPRSUBM 557  
ADDEXITPGM 687  
ANZDPRJRN 686  
CHGJRN 685  
CRTJRN 683  
GRTPRAUT 682  
GRTOBJAUT 682  
RCVJRNE 684  
RMVDPRREG 602  
RMVEXITPGM 686  
RVKDPRAUT 600  
STRDPRAPY 489  
STRJRNPf 684  
WRKREGINF 687

SETXCF 613

STRDPRAPY 478

UNIX

asnapply 533  
asncap 542  
asncccmd 526  
asntrc 549  
UNIX 版  
asnanalyze 517  
asnapply 533  
asncap 542  
asncccmd 526  
asntrc 549

Windows

asnapply 533  
asncap 542  
asncccmd 526  
asntrc 549

Windows 版

asnanalyze 517  
asnapply 533  
asncap 542  
asncccmd 526  
asntrc 549

レプリケーション・コマンド (続き)

z/OS

asnapply 533  
asncap 542  
asncccmd 526  
asntrc 549

z/OS 版

asnapply 533  
asncap 542  
asncccmd 526  
asntrc 549  
MODIFY 614

レプリケーション・サービス

作成 634, 635  
操作 634, 636  
ドロップ 634, 637  
名前 634, 635

レプリケーション・ステップの印刷  
1081

レプリケーション・センター

コネクティビティ 663, 664  
IASP に対する制限 663, 674

レプリケーション・ソース

結合 637, 659  
サブスクライブ 687, 690  
登録

行 637, 644

非 DB2 リレーショナル・デー  
ータ・ソース 637, 640

ビュー 637, 661

列 637, 642

DB2 表 637, 638

IMS データ・ソース 637,  
638

マッピング、ターゲットへの  
687, 701

CCD (整合変更データ) 表 687,  
710

CCD 表の保守 637, 662

連合システム

DELETE ステートメント 475

ローカル

データ・タイプ 456

ロード

ファイルからデータベース表へ  
1149



ロード (続き)  
ファイル・タイプ修飾子 1149  
ロード API 945  
ロード・ユーティリティ  
一時ファイル 1149  
ファイル・タイプ修飾子 945  
呼び出し可能、CLI から 870  
DB2 Data Links Manager 426  
ロールフォワード・ユーティリ  
ティ  
ドロップされた表のリカバリー  
410  
ロールフォワード・リカバリー 677  
表スペース 402  
レプリケーション内 663  
ロギング  
ウェアハウス・エージェント用開  
始 213  
初期ロギングなしでの表の作成  
1484  
ログ  
リカバリー、割り振り 926  
ログ・レコード  
コンプレッション・ディクショナ  
リー (z/OS) 617, 622  
保守 617, 618  
保存 617, 619  
ロケール  
サポート 111  
ロジカル・パーティション  
キー 637  
ロック  
パフォーマンス要因 313  
論理パーティション化キー  
説明 649

## [ワ行]

割り当て  
基本 SQL 操作 1364

## [数字]

10 進数への変換の要約、整数の  
1364

2 バイト文字  
割り当て時の切り捨て 1364  
2 バイト文字ラージ・オブジェクト  
(DBCLOB)  
レプリケーションの考慮事項  
603, 604  
3 層レプリケーション構成 687, 711

## A

action precompile/bind オプション  
1190  
ActiveX データ・オブジェクト  
DB2 AD クライアントでのサポ  
ート 757  
ADD COLUMN 文節、処理順序  
1402  
ADDDPRREG コマンド 570  
ADDDPRSUB コマンド 580  
ADDDPRSUBM コマンド 557  
ADDEXITPGM コマンド 663, 687  
administration server 305  
AIX  
インストールに関する考慮事項  
134  
ウェアハウス・エージェント  
インストール 223  
システム・クロック 134  
ディスク・スペース要件 134  
メモリー要件 134  
db2setup と SMIT 134  
TCP/IP ポート番号 134  
ALIAS 文節  
DROP ステートメント 1572  
ALTER PROCEDURE ステートメン  
ト 1560  
ALTER TABLE ステートメント  
構文図 1402  
必要な許可 1402  
例 1402  
ALTER TABLE ステートメントの  
ADD 文節 1402  
ALTER TABLE ステートメントの  
DROP CHECK 文節 1402  
ALTER TABLE ステートメントの  
DROP CONSTRAINT 文節 1402

ALTER TABLE ステートメントの  
DROP PARTITIONING KEY 文節  
1402  
ALTER TABLESPACE ステートメン  
ト  
説明 1434  
anyorder ファイル・タイプ修飾子  
945, 1149  
ANZDPRJRN コマンド 663, 686  
API  
プリコンパイラーのカスタマイズ  
1019  
db2AutoConfig 934  
db2Backup 412  
db2Inspect 938  
db2Load 945  
db2ReadLogNoConn 1024  
db2ReadLogNoConnInit 1005  
db2ReadLogNoConnTerm 420  
db2Restore 993  
db2Rollforward 1007  
db2Runstats 985  
sqlcrea 926  
sqlsact 1019  
APPLHEAPSZ 構成パラメーター  
677  
使用法 340  
レプリケーション内 663  
APPLYTRACE (アプライ・トレース)  
表  
プルーニング 617, 630  
APPLYTRAIL (アプライ・トレール)  
表  
プルーニング 617, 630  
apply\_path パラメーター 478, 480,  
533, 535  
apply\_qual パラメーター 478, 480,  
533, 534  
app\_ctl\_heap\_sz 342  
ARM (自動リスタート・マネージャ  
ー) 613, 615  
AS 文節  
CREATE VIEW ステートメント  
1469  
ASC インポート・ファイル・タイプ  
1234

asnanalyze コマンド 517  
asnapply コマンド 533  
asncap コマンド 542  
asnccmd コマンド 526  
ASNDLCOPY 出口ルーチン 603, 608  
ASNDLCOPYD ファイル・コピー・デーモン 603, 611  
ASNDONE 出口ルーチン 使用 478, 490, 491  
リジェクトされたトランザクション 637, 657  
asndone.smp ファイル 478, 491  
ASNL2RNx コマンド 613  
ASNLOAD 出口ルーチン エラー処理 478, 493  
クロスローダー・ユーティリティの使用 478, 498  
説明 478, 493  
前提条件 478, 493  
動作のカスタマイズ 478, 497  
ファイル、生成される 478, 495  
asnload.ini ファイルの使用 478, 498  
DATALINK レプリケーションの 603, 607  
OS/400 版 478, 498  
UNIX 版 478, 494  
Windows 版 478, 494  
z/OS 版 478, 495  
asnload.ini ファイル 478, 498  
ASNPLXFY ユーティリティ 613, 617  
asntrc コマンド 549  
ASUTIME  
CREATE FUNCTION (外部スカラー) ステートメントにおける 1618  
CREATE FUNCTION (外部表) ステートメント 1644  
CREATE PROCEDURE ステートメントの 1672, 1686  
Autoconfigure API 934  
AUTOCONFIGURE コマンド 1131  
autoprunce パラメーター 526, 542  
asncap コマンドで使用 544

autoprunce パラメーター (続き)  
asnccmd コマンドで使用 529  
autostop パラメーター 526, 529, 542, 544

## B

Backup database API 412  
backup database (データベースのバックアップ) コマンド 677  
BigDecimal Java データ・タイプ 782  
BIGINT SQL データ・タイプ C への変換 845  
CREATE TABLE ステートメントの 1484  
Java 782  
BINARY LARGE OBJECT データ・タイプ 1484  
BINARY SQL データ・タイプ C への変換 845  
binarynumerics ファイル・タイプ修飾子 945, 1149  
bindfile precompile オプション 1190  
BLAST 1398  
BLOB SQL データ・タイプ C への変換 845  
BLOB データ・タイプ  
CREATE TABLE ステートメントの 1484  
Java 782  
BLOB (バイナリー・ラージ・オブジェクト)  
レプリケーションの考慮事項 603, 604  
blocking precompile/bind オプション 1190  
BUFFERPOOL 文節  
ALTER TABLESPACE ステートメント 1434  
CREATE TABLESPACE ステートメント 1562  
DROP ステートメント 1572

## C

### C

サポートされるバージョン  
AIX 759  
HP-UX 761  
Linux 762  
Solaris オペレーティング環境 765  
Windows 766  
ファイルのビルド 769  
CALL プロシージャ  
定義、サブスクリプション・セットの 687, 698  
レプリケーション前後の実行時処理 721, 722  
CAPENQ (キャプチャー・エンキュー) 表 517  
CAPMON (キャプチャー・モニター) 表  
ブルーニング 617, 629  
CAPTRACE (キャプチャー・トレース) 表  
構造 521  
ブルーニング 617, 629  
capture\_path パラメーター 542, 543  
capture\_schema パラメーター 526, 527, 542, 543  
capture\_server パラメーター 526, 527, 542, 543  
CASCADE 削除規則 1484  
CCD (整合変更データ) 表  
外部  
概要 687  
multi-tier レプリケーション 687, 710  
構造  
ターゲット・サーバー 731  
コンデンス 687  
コンプリート 687  
使用方法  
履歴または監査 687, 708  
multi-tier レプリケーション 687, 710  
追加、UOW 列の 687, 708

CCD (整合変更データ) 表 (続き)  
   内部  
     概要 687  
     複数のターゲット 687, 708  
   非 DB2 リレーショナル・データ・ソース 637  
     使用, CCD 表の 640  
   非コンデンス 687  
   非コンプリート 687  
   非リレーショナル・データ・ソース  
     使用, CCD 表の 637, 638  
     CCD 表の保守 637, 662  
     レプリケーション・ソース 687, 710  
 CCSID (coded character set identifier)  
   双方向サポート 270  
   CREATE TABLE ステートメントの 1484  
   DECLARE GLOBAL TEMPORARY TABLE ステートメントの 1610  
 CCSIDG precompile/bind オプション 1190  
 CCSIDM precompile/bind オプション 1190  
 CCSIDS precompile/bind オプション 1190  
 CD (変更データ) ビュー 637, 658  
 CD (変更データ) 表  
   結合用 637, 659  
   ビュー用の 637, 658  
   プルーニング 617, 628  
   要約, 内容の 687, 707  
 CHAR SQL データ・タイプ  
   C への変換 845  
 CHAR VARYING データ・タイプ 1484  
 CHAR データ・タイプ  
   Java 782  
 CHARACTER VARYING データ・タイプ 1484  
 CHARACTER データ・タイプ 1484  
 chardel ファイル・タイプ修飾子  
   インポート 1234  
   エクスポート 1134  
   ロード 945, 1149  
 chardel ファイル・タイプ修飾子 (続き)  
   ロード 945, 1149  
 charsub precompile/bind オプション 1190  
 CHGJRN コマンド 663, 685  
 Clean Data トランスフォーマー 1089  
 CLI (コール・レベル・インターフェース)  
   概要 796  
   コンバウンド SQL 836  
   自動バインド 839  
   ストアード・プロシージャ呼び出し 839  
 CLI サンプル・プログラム・ファイル 863  
 CLIPKG CLI/ODBC キーワード 874  
 CLOB (文字ラージ・オブジェクト) レプリケーションの考慮事項 604  
 CLOB (文字ラージ・オブジェクト) データ・タイプ  
   列の作成 1484  
   レプリケーションの考慮事項 603  
   C への変換 845  
   Java 782  
 cnulreqd precompile/bind オプション 1190  
 COBOL 言語  
   サポートされる AIX バージョン 759  
   サポートされる HP-UX バージョン 761  
   サポートされる Solaris オペレーティング・システム 765  
   サポートされる Windows バージョン 766  
   ファイルのビルド 769  
 coldel ファイル・タイプ修飾子  
   インポート 1234  
   エクスポート 1134  
   ロード 945, 1149  
 collate\_info 構成パラメーター 363  
 COLLATING\_SEQUENCE サーバー・オプション  
   有効な設定 437  
 collection precompile/bind オプション 1190  
 COLLID  
   CREATE FUNCTION (外部スカラー) ステートメントにおける 1618  
   CREATE FUNCTION (外部表) ステートメント 1644  
   CREATE PROCEDURE ステートメントの 1672, 1686  
 COMMIT ON RETURN  
   CREATE PROCEDURE ステートメントの 1672, 1686  
 COMMIT ステートメント  
   パススルー 468  
 commit\_interval パラメーター  
   asnscap コマンドで使用 542, 544  
   asnccmd コマンドで使用 526, 529  
 COMM\_RATE サーバー・オプション  
   有効な設定 437  
 connect precompile オプション 1190  
 CONNECTSTRING サーバー・オプション  
   有効な設定 437  
 control\_server パラメーター 478, 481, 533, 535  
 copyonce パラメーター 478, 481, 533, 538  
 CPU\_RATIO サーバー・オプション  
   有効な設定 437  
 CREATE DATABASE  
   例 324  
 CREATE FUNCTION (SQL スカラー、表、または行) ステートメント 1552  
 CREATE FUNCTION (外部スカラー) ステートメント 1618  
 CREATE FUNCTION (外部表) ステートメント 1644  
 CREATE FUNCTION ステートメント  
   外部表 1644

- CREATE FUNCTION ステートメント (続き)  
SQL スカラー、表、または行  
1552
- CREATE FUNCTION (ソースまたは  
テンプレート) ステートメント  
1662
- CREATE FUNCTION (ソース) ステ  
ートメント 1662
- CREATE METHOD ステートメント  
1546
- CREATE PROCEDURE (SQL) ステ  
ートメント  
説明 1686
- CREATE PROCEDURE (外部) ステ  
ートメント  
説明 1672
- CREATE TABLE ステートメント  
構文図 1484
- CREATE TABLESPACE ステートメ  
ント  
説明 1562
- CREATE TYPE MAPPING ステート  
メント  
例 456
- CREATE TYPE (構造化) ステートメ  
ント 1441
- CREATE VIEW ステートメント  
説明 1469
- CREATE VIEW ステートメントの  
CHECK 文節 1469
- CRTJRN コマンド 663, 683
- CRTJRNRCV コマンド 663, 683
- C++  
サポートされる AIX バージョン  
759  
サポートされる HP-UX バージョ  
ン 761  
サポートされる Linux バージョ  
ン 762  
サポートされる Solaris オペレー  
ティング環境 765  
サポートされる Windows バージ  
ョン 766  
ファイルのビルド 769
- C/C++ アプリケーション  
マルチスレッドデータベース・ア  
クセス 742
- ## D
- Data Links Manager  
システム要件  
AIX 134  
Solaris オペレーティング環境  
124  
Windows 143  
バージョン、Windows 上の DB2  
サーバー 143  
メモリー要件  
AIX 134  
Solaris オペレーティング環境  
124  
Windows 143  
レプリケーション 603  
レプリケーション・デーモン  
603
- Data Links Manager 管理者ユーザー  
ID  
AIX 134  
Solaris オペレーティング環境  
124
- Data Links Manager レプリケーシ  
ョン・デーモン 610
- DATABASE PARTITION GROUP 文  
節  
DROP ステートメント 1572
- DATALINK 値  
制約事項 655, 713  
保管、更新の 649  
レプリケーション 605  
ASNDLCOPY 出口ルーチン 608  
ASNDLCOPYD ファイル・コピ  
ー・デーモン 611  
DLFM\_ASNCOPYD ファイル・コ  
ピー・デーモン 610
- DATALINK データ・タイプ  
値 603  
サポートされない 448  
制約事項 637  
説明 1363
- DATALINK データ・タイプ (続き)  
BNF 指定 1078  
CREATE TABLE ステートメント  
1484  
DROP ステートメント 1572
- DataPropagator  
レプリケーション、z/OS ウェア  
ハウス・エージェントでの自動  
化 211
- DATE SQL データ・タイプ  
C への変換 845
- DATE データ・タイプ  
表の作成 1484  
Java 782
- dateformat ファイル・タイプ修飾子  
945, 1149, 1234
- datesiso 1134
- datesiso ファイル・タイプ修飾子  
945, 1149, 1234
- DATETIME precompile/bind オプショ  
ン 1190
- DB2  
移行  
概要 394  
推奨事項 106  
インターフェース言語 111  
コード・ページ 111  
ライセンス・ファイル 184  
ロケール 111
- DB2 Administration Server (DAS)  
概要 305  
構成 337  
スケジューラーのセットアップと  
構成 327
- DB2 CLI サンプル・プログラム・フ  
ァイル 863
- DB2 Connect  
コントロール・センター 1953
- DB2 Connect ライセンス・キー  
ライセンス・センターを使用した  
インストール 121  
db2licm コマンドを使用したイン  
ストール 122
- DB2 Data Links Manager  
インスタンス間のエクスポート  
421

- DB2 Data Links Manager (続き)
  - インポート・ユーティリティ 425
  - エクスポート・ユーティリティ 421
  - ロード・ユーティリティ 426
- DB2 Enterprise Edition
  - 組み合わせ、Extended Enterprise Edition との 87
- DB2 Enterprise Server Edition
  - 制約 643
  - 制約事項 637
- DB2 for iSeries
  - デフォルトの逆方向データ・タイプ・マッピング 427
  - LOB サポート 467
- DB2 for UNIX および Windows
  - デフォルトの逆方向データ・タイプ・マッピング 427
  - デフォルトの順方向タイプ・マッピング 1381
- DB2 for z/OS および OS/390
  - デフォルトの逆方向データ・タイプ・マッピング 427
  - デフォルトの順方向タイプ・マッピング 1381
- DB2 HTML ドキュメンテーション
  - 更新 89
- DB2 (iSeries 版)
  - デフォルトの順方向データ・タイプ・マッピング 1381
- DB2 Personal Edition
  - インスタンスおよびデータベースの移行
    - Linux 101
    - データベースの移行
      - Windows 103
- DB2 Server for VM および VSE
  - デフォルトの逆方向データ・タイプ・マッピング 427
  - デフォルトの順方向タイプ・マッピング 1381
- DB2 SQLj プロファイル・カスタマイザー・コマンド 1298
- DB2 SQLj プロファイル・プリンター・コマンド 1304
- DB2 Universal Database
  - 製品およびパッケージの変更 87
  - Enterprise Server Edition 87
  - Workgroup Edition、名前変更 87
- DB2 Universal Database for z/OS
  - ウェアハウス・トランスフォーマーの設定 206
- DB2 Universal Database サーバー
  - アクセス
    - ホスト・アプリケーションからの 157
    - AS/400 アプリケーションからの 157
    - サポートされるプロトコル
      - ホストからの 158
      - OS/400 からの 158
- DB2 インフォメーション・センター
  - 88
  - 呼び出し 1958
- DB2 エクステンダー
  - 制約事項 603, 605
- DB2 クライアント
  - 概要 108
  - カタログ
    - TCP/IP ノード 95
- DB2 サーバー
  - インストール
    - Windows 396
    - 手動インストール後の設定 159
- DB2 索引アドバイザー 1286
- DB2 索引アドバイザー・コマンド 1286
- DB2 製品
  - 手動でのインストール 161
- DB2 チュートリアル 227
- DB2 統計および DDL 抽出ツール・コマンド 1289
- DB2 ドキュメンテーション
  - UNIX でのインストール 104
- DB2 ビュー
  - 登録 637, 661
- DB2 表
  - 登録 638
- DB2 ライセンス・キー
  - インストール
    - ライセンス・センターの使用 187
    - db2licm コマンドを使用した 185
- DB2 ライセンス・ポリシー
  - 設定
    - db2licm コマンドを使用した 180
- DB2 レプリケーション
  - 許可要件 663, 666
- DB2ACCOUNT レジストリー変数 366
- db2advsi 1286
- db2AutoConfig API 934
- db2Backup API 412
- DB2BIDI レジストリー変数 366
- DB2BPVARS 346
- DB2CHKPTR 346
- DB2CODEPAGE 環境変数 190, 663, 677
- DB2CODEPAGE レジストリー変数 366
- DB2CONNECTINAPPPROCESS 環境変数 343
- DB2DARILOOKUPALL 346
- DB2DBDFT 環境変数 663, 677
- DB2DBDFT レジストリー変数 366
- DB2DBMSADDR レジストリー変数 366
- DB2DISABLEFLUSHLOG レジストリー変数 366
- DB2DISCOVERYTIME レジストリー変数 366
- DB2DOMAINLIST 343
- DB2ENVLIST 343
- DB2INCLUDE レジストリー変数 366
- db2inidb コマンド 1315
- db2Inspect API 938
- db2inspf コマンド 1218
- DB2INSTANCE 環境変数 190, 343, 663, 676
- DB2INSTDEF レジストリー変数 366

DB2INSTOWNER レジストリー変数 366

DB2INSTPROF 環境変数 343

DB2LIBPATH 343

db2licm コマンド 122, 185

DB2 ライセンス・キーのインストール 185

DB2 ライセンス・ポリシーの設定 180

DB2LICSTATSIZE レジストリー変数  
レジストリー変数 366

db2Load API 945

DB2LOCALE レジストリー変数 366

db2look コマンド 1289

DB2MAXFSCRESEARCH パフォーマンス変数 346

DB2MEMDISCLAIM パフォーマンス変数 346

DB2MEMMAXFREE パフォーマンス変数 346

DB2NBDISCOVERRCVBUFS レジストリー変数 366

DB2NODE 343

DB2NTMEMSIZE 346

DB2NTPRICLASS 346

DB2NTWORKSET 346

DB2OPTIONS レジストリー変数 366

DB2PARALLELIO 343

DB2PATH 環境変数 343

DB2PRIORITIES 346

db2ReadLogNoConn API 1024

db2ReadLogNoConnInit API 1005

db2ReadLogNoConnTerm API 420

db2relocatedb コマンド 1305

db2Restore API 993

db2Rollforward API 1007

db2Runstats API 985

db2set コマンド 308

DB2SLOGON レジストリー変数 366

db2sqljcustomize コマンド 1298

db2sqljprint コマンド 1304

db2start コマンド 1279

db2support コマンド 1310

DB2TERRITORY レジストリー変数 366

DB2TIMEOUT レジストリー変数 366

DB2TRACEFLUSH レジストリー変数 366

DB2TRACENAME レジストリー変数 366

DB2TRACEON レジストリー変数 366

DB2TRCSYSERR レジストリー変数 366

DB2YIELD レジストリー変数 366

DB2\_ANTIJOIN 371

DB2\_APM\_PERFORMANCE 346

DB2\_AVOID\_PREFETCH 346

DB2\_AWE 346

DB2\_BINSORT 346

DB2\_CORRELATED\_PREDICATES 371

DB2\_ENABLE\_BUFDPD 346

DB2\_EXTENDED\_OPTIMIZATION 346

DB2\_HASH\_JOIN 371

DB2\_INLIST\_TO\_NLJN 371

DB2\_LIKE\_VARCHAR 371

DB2\_MINIMIZE\_LIST\_PREFETCH 371

DB2\_MMAP\_READ 346

DB2\_MMAP\_WRITE 346

DB2\_NEW\_CORR\_SQ\_FF 371

DB2\_OVERRIDE\_BPF 346

DB2\_PINNED\_BP 346

DB2\_PRED\_FACTORIZE 371

DB2\_REDUCED\_OPTIMIZATION 371

DB2\_SELECTIVITY 371

DB2\_SORT\_AFTER\_TQ 346

DB2\_STPROC\_LOOKUP\_FIRST 346

db2\_subsystem パラメーター 482, 535

DB2\_USE\_PAGE\_CONTAINER\_TAG 343

DBADM 668

DBCLOB (2 バイト文字ラージ・オブジェクト)  
レプリケーションの考慮事項 604

DBCLOB SQL データ・タイプ C への変換 845

DBCLOB データ・タイプ  
レプリケーションの考慮事項 603

CREATE TABLE ステートメント  
の 1484

Java 782

DBHEAP 構成パラメーター 663, 677

DBNAME サーバー・オプション  
有効な設定 437

dec precompile/bind オプション 1190

decdel precompile/bind オプション 1190

DECIMAL SQL データ・タイプ C への変換 845

DECIMAL データ・タイプ  
浮動小数点からの変換 1364

Java 782

DECLARE GLOBAL TEMPORARY TABLE 326

DECLARE GLOBAL TEMPORARY TABLE ステートメント  
説明 1610

decplusblank ファイル・タイプ修飾子 945, 1134, 1149, 1234

decpt ファイル・タイプ修飾子 945, 1134, 1149, 1234

deferred\_prepare precompile オプション 1190

degree precompile/bind オプション 1190

delay パラメーター 478, 482, 533, 539

delprioritychar ファイル・タイプ修飾子 945, 1149, 1234

DiagIdentifier 引き数  
ヘッダー・フィールド 798  
レコード・フィールド 798

disconnect precompile オプション 1190

DISTINCT TYPE 文節  
DROP ステートメント 1572

dldel ファイル・タイプ修飾子 945, 1134, 1149, 1234

- DLFM (Data Links ファイル・マネージャ) エラー・メッセージ 1033
- DLFM コマンド  
dlfm\_dump 1074
- DLFM ユーザー・アカウント 124, 134, 143
- DLFM\_ASNCOPYD ファイル・コピー・デーモン 603, 610
- dlfm\_dump コマンド  
構文 1074
- dlnadmin ユーザー名  
Windows 143
- DMS 表スペース  
CREATE TABLESPACE ステートメント 1562
- Documentum 1398
- double Java データ・タイプ 782
- DOUBLE SQL データ・タイプ  
C への変換 845
- DOUBLE データ・タイプ 1484
- DOUBLE-PRECISION データ・タイプ 1484
- DPR 登録 (OS/400)  
削除 602  
除去 602  
追加 570
- DROP FOREIGN KEY 文節  
ALTER TABLE ステートメント 1402
- DROP PRIMARY KEY 文節  
ALTER TABLE ステートメント 1402
- DROP TRANSFORM 1572
- DROP UNIQUE 文節  
ALTER TABLE ステートメント 1402
- DROP ステートメント  
説明 1572
- DSMICONFIG 407
- DSMIDIR 407
- DSMILOG 407
- DSNUTILS 1081
- dumpfile ファイル・タイプ修飾子 945, 1149
- DYNAMICRULES precompile/bind オプション 1190
- ## E
- EDITPROC 文節  
制約事項、圧縮 603, 604
- errwait パラメーター 478, 482, 533, 540
- Excel ファイル 1398
- EXECUTE ステートメント  
説明 1602
- explain bind オプション 1190
- explsnap precompile/bind オプション 1190
- EXPORT コマンド 1134
- ## F
- fastparse ファイル・タイプ修飾子 945, 1149
- federated precompile/bind オプション 1190
- FIELDPROC 文節  
制約事項、圧縮 603, 604  
ALTER TABLE ステートメントの 1402
- FLOAT SQL データ・タイプ  
C への変換 845
- FLOAT データ・タイプ 1484  
Java 782
- FOLD\_ID サーバー・オプション  
有効な設定 437
- FOLD\_PW サーバー・オプション  
有効な設定 437
- FOR BIT DATA 文節  
CREATE TABLE ステートメント 1484
- FOR FETCH ONLY 文節  
SELECT ステートメント 1317
- FOR READ ONLY 文節  
SELECT ステートメント 1317
- FOR 文節  
CREATE TABLE ステートメント 1484
- forcein ファイル・タイプ修飾子 945, 1149, 1234
- FOREIGN KEY 文節  
制約名、規則 1484  
複数パス、使用結果 1484
- CASCADE 文節、伝搬サマリー 1484
- CREATE TABLE ステートメント 1484
- IDENTITY、規則 1484
- RESTRICT 文節、禁止 1484
- SET NULL 文節、操作 1484
- funcpath precompile/bind オプション 1190
- ## G
- generatedignore ファイル・タイプ修飾子 945, 1149, 1234
- generatedmissing ファイル・タイプ修飾子 945, 1149, 1234
- generatedoverride ファイル・タイプ修飾子 945, 1149
- generic precompile/bind オプション 1190
- GRAPHIC SQL データ・タイプ  
C への変換 845
- GRAPHIC データ・タイプ  
CREATE TABLE に対する 1484  
Java 782
- GROUP BY 文節  
サブスクリプション・セット 687
- GRTPRAUT コマンド  
レプリケーションの 663  
SQL パッケージへの特権の付与 682
- GRTOBJAUT コマンド 663, 682
- ## I
- IASP グループ 663, 674
- IBM 以外のソース  
サポートされるバージョンおよびリリース・レベル 1078



IDENTITY 列  
新規表での定義 336  
CREATE TABLE ステートメント  
1484  
identityignore 1234  
identityignore ファイル・タイプ修飾  
子 945, 1149  
identitymissing ファイル・タイプ修飾  
子 945, 1149, 1234  
identityoverride ファイル・タイプ修  
飾子 945, 1149  
IFILE サーバー・オプション  
有効な設定 437  
IGNORE\_UDT サーバー・オプション  
有効な設定 437  
implieddecimal ファイル・タイプ修飾  
子 945, 1149, 1234  
IMPORT コマンド 1234  
IMPORT ユーティリティ  
DB2 Data Links Manager 425  
IMS DataPropagator 637, 638  
IMS データ・ソース  
使用、CCD 表の 637, 638  
登録 637, 638  
CCD 表の保守 637, 662  
IN  
CREATE TABLE ステートメント  
の 1484  
inamsg パラメーター 478, 483, 533,  
537  
INDEX キーワード  
DROP ステートメント 1572  
indexfreespace ファイル・タイプ修飾  
子 945, 1149  
indexixf ファイル・タイプ修飾子  
1234  
indexschema ファイル・タイプ修飾子  
1234  
Informix  
デフォルトの逆方向タイプ・マッ  
ピング 427  
デフォルトの順方向タイプ・マッ  
ピング 1381  
insert precompile/bind オプション  
1190  
INSPECT コマンド 1144

Int Java データ・タイプ 782  
INTEGER SQL データ・タイプ  
C への変換 845  
INTEGER データ・タイプ 1484  
Java 782  
IO\_RATIO 比率・オプション  
有効な設定 437  
iSeries  
アプリケーション 157  
ウェアハウス・エージェント  
インストール 214  
iSeries サーバー  
接続 663, 665  
isolation precompile/bind オプション  
1190  
IUD\_APP\_SVPT\_ENFORCE サーバ  
ー・オプション  
有効な設定 437  
IWH.environment ファイル、サンプル  
228

## J

Java  
アプレット  
使用のポイント 772  
環境のセットアップ 741  
クラス・ライブラリー 781  
サポート、DB2 AD クライアント  
757  
サポートされる AIX JDK バージ  
ョン 759  
サポートされる HP-UX JDK バ  
ージョン 761  
出力ファイル 744  
ソース・ファイル 744  
パッケージおよびクラス、  
COM.ibm.db2.app 782  
ビルド  
JDBC アプレット 776  
SQLj アプリケーション 787  
SQLj アプレット 778  
HP-UX 環境のセットアップ 751  
Linux  
サポートされる JDK バージョ  
ン 762  
Java (続き)  
Solaris オペレーティング環境  
サポートされる JDK バージョ  
ン 765  
SQLj (Embedded SQL for Java)  
アプレット 746  
制約 746  
UNIX 環境のセットアップ 750  
Windows  
サポートされる JDK バージョ  
ン 766  
Java データ・タイプ  
BigDecimal 782  
Blob 782  
Double 782  
Int 782  
java.math.BigDecimal 782  
Short 782  
String 782  
java.math.BigDecimal Java データ・タ  
イプ 782  
JCL  
アプライ・プログラムの開始  
613  
アプライ・プログラムの始動  
614  
キャプチャー・プログラムの開始  
613  
キャプチャー・プログラムの始動  
614  
レプリケーション・アラート・モ  
ニターの開始 613, 614  
JDBC  
アプレット、使用のポイント  
772  
アプレットのビルド 776  
ドライバーの機能強化 90  
DB2 AD クライアントでのサポー  
ト 757  
JDBC 1.2 ドライバー  
使用すべきでない 90  
K  
keepblanks ファイル・タイプ修飾子  
945, 1149, 1234



## L

lag\_limit パラメーター 526, 542, 544  
LANG 環境変数 190  
langlevel precompile オプション 1190  
LC\_ALL 環境変数 190  
level precompile オプション 1190  
LIBPATH 663, 677  
Linux  
    インスタンスおよびデータベースの移行 101  
    インストール  
        ウェアハウス・エージェント 223  
        ウェアハウス・エージェント  
            インストール 223  
        実行可能ファイル 190  
LOAD コマンド 1149  
loadxlt パラメーター 478, 483, 533, 536  
LOB (ラージ・オブジェクト)  
    レプリケーションの考慮事項 604  
    update-anywhere の制約事項 713  
LOB (ラージ・オブジェクト) データ・タイプ  
    プラン 467  
    レプリケーションの考慮事項 603  
    update-anywhere の制約事項 687  
lobsinfile ファイル・タイプ修飾子 945, 1134, 1149, 1234  
LOGBUFSZ 構成パラメーター 663, 677  
LOGFILSIZ 構成パラメーター 663, 677  
LOGIN\_TIMEOUT サーバー・オプション  
    有効な設定 437  
LOGPRIMARY 構成パラメーター 663, 677  
logreuse パラメーター  
    アプライ用 478, 533  
    キャプチャー用 526, 542

logreuse パラメーター (アプライ用) 483, 536  
logreuse パラメーター (キャプチャー用) 530, 545  
LOGSECOND 構成パラメーター 677  
    ラージ・ワークステーション 663  
logstdout パラメーター  
    アプライ用 478, 533  
    キャプチャー用 526, 542  
logstdout パラメーター (アプライ用) 484, 536  
logstdout パラメーター (キャプチャー用) 530, 545  
LONG VARCHAR データ・タイプ  
    CREATE TABLE に対する 1484  
    Java 782  
LONG VARGRAPHIC スtringの比較、使用制限 1364  
LONG VARGRAPHIC データ・タイプ 604  
    制約事項 603  
    Java 782  
longerror precompile オプション 1190  
LONGVARBINARY データ・タイプ  
    C への変換 845  
LONGVARCHAR データ・タイプ  
    C への変換 845  
LONGVARGRAPHIC データ・タイプ  
    C への変換 845

## M

MANAGED BY 文節、CREATE TABLESPACE ステートメント 1562  
MAXAPPLS 構成パラメーター 663, 677  
MAX\_SYNCH\_MINUTES、データ・ブロッキング 687, 694  
MDC 表 279  
memory\_limit パラメーター  
    asncap コマンドで使用 542, 545

memory\_limit パラメーター (続き)  
    asncmd コマンドで使用 526, 530  
messages precompile/bind オプション 1190  
METHOD 文節  
    DROP ステートメント 1572  
Microsoft SQL Server  
    アクセス先の構成 459  
    デフォルトの逆方向データ・タイプ・マッピング 427  
    デフォルトの順方向タイプ・マッピング 1381  
    レプリケーションの制約事項 637, 645  
    連合サーバーのセットアップ 469  
modifiers ファイル・タイプ  
    ロード API 945  
MODIFY コマンド 613, 614  
monitor\_interval パラメーター  
    キャプチャー用 526, 542  
monitor\_interval パラメーター (キャプチャー用) 530, 545  
monitor\_limit パラメーター 526, 531, 542, 545  
MONTRAIL (モニター・トレール) 表 729  
mon\_heap\_sz 構成パラメーター 363  
multi-tier レプリケーション  
    定義、サブスクリプション・セットの 687, 710  
MVS (Multiple Virtual Storage)  
    コンソール 613  
MVS コンソール 614

## N

NetBIOS  
    構成  
        クライアントでの 156  
        論理アダプター番号の判別 156  
NICKNAME 文節  
    DROP ステートメント 1572  
NO ACTION 削除規則 1484

nochecklengths ファイル・タイプ修飾子 945, 1149, 1234  
 NODE サーバー・オプション  
   有効な設定 437  
 nodefaults ファイル・タイプ修飾子 1234  
 nodoubledel ファイル・タイプ修飾子 945, 1134, 1149, 1234  
 noeofchar ファイル・タイプ修飾子 945, 1149, 1234  
 noheader ファイル・タイプ修飾子 945, 1149  
 nolinemacro precompile オプション 1190  
 norowwarnings ファイル・タイプ修飾子 945, 1149  
 NOT NULL 文節  
   CREATE TABLE ステートメント 1484  
 notify パラメーター 478, 484, 533, 537  
 notypeid ファイル・タイプ修飾子 1234  
 NTFS  
   ドライブの共用 143  
   フォーマット済みドライブ 143  
 NULL CALL  
   CREATE TYPE (構造化) ステートメントの 1441  
 NULL 値、SQL  
   割り当て 1364  
 nullindchar ファイル・タイプ修飾子 945, 1149, 1234  
 NUMERIC SQL データ・タイプ  
   C への変換 845  
 NUMERIC データ・タイプ  
   Java 782

**O**  
 ODBC  
   デフォルトの順方向タイプ・マッピング 1381  
 ODBC (open database connectivity)  
   および DB2 CLI 796  
   コア・レベル関数 796  
 ODBC (open database connectivity) (続き)  
   使用可能アプリケーション 119  
 OF 文節  
   CREATE VIEW ステートメント 1469  
 OID 列 1484  
 ON UPDATE 文節 1484  
 on-db-partitions 文節  
   CREATE TABLESPACE ステートメント 1562  
 opt4one パラメーター 478, 484, 533, 539  
 OPTION 文節  
   CREATE VIEW ステートメント 1469  
 optlevel precompile オプション 1190  
 Oracle  
   デフォルトの逆方向タイプ・マッピング 427  
   デフォルトの順方向タイプ・マッピング 1381  
 Oracle データ・ソース  
   制約事項 637  
 OS/400  
   データ・ソース、リモート・ジャーナル処理 637  
 OS/400 データ・ソース  
   リモート・ジャーナリングを使用する 657  
 output precompile オプション 1190  
 owner precompile/bind オプション 1190

**P**  
 PACKAGE 文節  
   DROP ステートメント 1572  
 packages precompile オプション 1190  
 packeddecimal ファイル・タイプ修飾子 945, 1149  
 PACKET\_SIZE サーバー・オプション  
   有効な設定 437  
 pagefreespace ファイル・タイプ修飾子 945, 1149  
 PATCH1 CLI/ODBC キーワード 872  
 PATCH2 CLI/ODBC キーワード 873  
 PC/IXF ファイルのインポート時に使用 1234  
 PLAN\_HINTS サーバー・オプション  
   有効な設定 437  
 PRECOMPILE コマンド 1190  
 PREP コマンド 1190  
 preprocessor precompile オプション 1190  
 PRIMARY KEY 文節  
   ALTER TABLE ステートメント 1402  
   CREATE TABLE ステートメント 1484  
 priv\_mem\_thresh 構成パラメーター 365  
 PROGRAM TYPE  
   CREATE FUNCTION (外部スカラー) ステートメントにおける 1618  
   CREATE FUNCTION (外部表) ステートメント 1644  
 PROGRAM、DROP ステートメントの 1572  
 prune\_interval パラメーター 526, 531, 542, 546  
 PUSHDOWN サーバー・オプション  
   有効な設定 437  
 pwdfile パラメーター 478, 485, 533, 535

**Q**  
 qualifier precompile/bind オプション 1190  
 queryopt precompile/bind オプション 1190  
 QUIESCE コマンド 1219

## R

RCVJRNE コマンド 663, 684  
REAL SQL データ・タイプ  
C への変換 845  
REAL データ・タイプ  
CREATE TABLE ステートメント  
の 1484  
Java 782  
REBIND\_ROUTINE\_PACKAGE プロ  
シージャー 1400  
reclen ファイル・タイプ修飾子  
1234  
ロード 1149  
ロード API 945  
release precompile/bind オプション  
1190  
REORGANIZE TABLE コマンド  
コントロール表 617  
REORGCHK コマンド 1222  
Restore database API 993  
RESTORE DATABASE コマンド  
1267  
RESTRICT 削除規則 1484  
retention\_limit パラメーター 526,  
531, 542, 546  
REXX 言語  
サポートされる AIX バージョン  
759  
サポートされる Windows バージ  
ョン 766  
DB2 AD クライアントでのサポー  
ト 757  
RMVDPREG コマンド 602  
RMVEXITPGM コマンド 663, 686  
Rollforward Database API 1007  
ROWID 605  
ROWID データ・タイプ  
レプリケーション内 603  
RRN 637, 658  
Runstats API 985  
RUNSTATS コマンド  
構文 1258  
RUNSTATS ユーティリティ 617,  
624  
Run-Time Client、関数の削除 87

RVKDPRAUT コマンド 600  
構文 600

## S

SAP  
ステップ 1081  
SCHEMA 文節  
DROP ステートメント 1572  
SCM (Service Control Manager)  
レプリケーション・サービスの作  
成 634, 635  
レプリケーション・サービスの操  
作 634, 636  
レプリケーション・サービスのド  
ロップ 634, 637  
レプリケーション・サービスの命  
名 634, 635  
SCOPE 文節  
ALTER TABLE ステートメント  
1402  
CREATE TABLE ステートメント  
1484  
CREATE VIEW ステートメント  
1469  
SELECT ステートメント  
定義 1317  
例 1317  
EXPORT コマンド内 1134  
Service Control Manager (SCM)  
レプリケーション・サービスの作  
成 634, 635  
レプリケーション・サービスの操  
作 634, 636  
レプリケーション・サービスのド  
ロップ 634, 637  
レプリケーション・サービスの命  
名 634, 635  
SET NULL 削除規則 1484  
SET PASSTHRU ステートメント  
考慮事項 468  
SETXCF コマンド 613  
Shift JIS X0213 コード・ページ  
前の変換表 304  
short Java データ・タイプ 782

SIGALRM 信号  
データベース・マネージャーの開  
始 1279  
SIGINT 信号  
データベース・マネージャーの開  
始 1279  
sleep パラメーター 478, 485, 533,  
538  
sleep\_interval パラメーター 526,  
531, 542, 546  
SMALLINT データ・タイプ  
静的 SQL 1484  
C への変換/C++ 845  
Java 782  
SMS 表スペース  
CREATE TABLESPACE ステート  
メント 1562  
Solaris オペレーティング環境  
インストール  
考慮事項 124  
要件、ディスク・スペース  
124  
ウェアハウス・エージェント  
インストール 223  
カーネル構成パラメーター 124  
システム・クロック 124  
メモリー要件  
Data Links Manager 124  
TCP/IP ポート番号 124  
UFS の製品バージョン・レベル  
124  
spillfile パラメーター 478, 485, 533,  
541  
SQL アクセス・グループ 796  
SQL オブジェクトの削除 1572  
SQL ステートメント  
実行時処理 721, 722  
定義、サブスクリプション・セッ  
トの 687, 698  
ルーチンで使用可能 1360  
ALTER PROCEDURE 1560  
ALTER TABLE 1402  
ALTER TABLESPACE 1434  
CREATE FUNCTION (SQL スカ  
ラー、表、または行) 1552

|                             |                                |                                   |
|-----------------------------|--------------------------------|-----------------------------------|
| SQL ステートメント (続き)            | SQLD フィールド、SQLDA の             | SQLj (embedded SQL for Java) (続き) |
| CREATE FUNCTION (外部スカラ      | 1322                           | カスタマイザー・オプション                     |
| ) 1618                      | SQLDA (SQL 記述子域)               | 749                               |
| CREATE FUNCTION (外部         | 内容 1322                        | 制約 746                            |
| 表) 1644                     | SQLDABC フィールド、SQLDA の          | ファイルのビルド 769                      |
| CREATE FUNCTION (ソースまた      | 1322                           | プログラム                             |
| はテンプレート) 1662               | SQLDAID フィールド、SQLDA の          | ビルド 787                           |
| CREATE FUNCTION (ソー         | 1322                           | DB2 AD クライアントでのサポー                |
| ス) 1662                     | SQLDATA フィールド、SQLDA の          | ト 757                             |
| CREATE METHOD 1546          | 1322                           | SQLLEN フィールド、SQLDA の              |
| CREATE PROCEDURE            | SQLDATALEN フィールド、SQLDA         | 1322                              |
| (SQL) 1686                  | の 1322                         | SQLLONGLEN フィールド、SQLDA            |
| CREATE PROCEDURE (外         | SQLDATATYPE_NAME フィール          | の 1322                            |
| 部) 1672                     | ド、SQLDA の 1322                 | SQLN フィールド、SQLDA の                |
| CREATE TABLE 1484           | SQLDriverConnect CLI 関数        | 1322                              |
| CREATE TABLESPACE 1562      | 構文 888                         | SQLNAME フィールド、SQLDA の             |
| CREATE TYPE (構造化) 1441      | sqlAttachToCtx() API 742       | 1322                              |
| CREATE VIEW 1469            | sqlBeginCtx() API 742          | SQLNextResult CLI 関数 923          |
| DECLARE GLOBAL              | sqlcrea API 926                | sqlrules precompile オプション 1190    |
| TEMPORARY TABLE 1610        | sqlDetachFromCtx() API 742     | SQLSetDescField CLI 関数            |
| DROP 1572                   | sqlEndCtx() API 742            | 構文 910                            |
| DROP TRANSFORM 1572         | sqlGetCurrentCtx() API 742     | SQLTYPE フィールド、SQLDA の             |
| EXECUTE 1602                | sqlInterruptCtx() API 742      | 1322                              |
| SQL ステートメント実行               | sqlerrcontinue パラメーター 478,     | SQLVAR フィールド、SQLDA の              |
| シリアライゼーション 742              | 486, 533                       | 1322                              |
| SQL 操作                      | sqlerror precompile/bind オプション | 1322                              |
| 基本 1364                     | 1190                           | sqlwarn precompile/bind オプション     |
| SQL データ・タイプ                 | sqlerrorcontinue パラメーター 541    | 1190                              |
| Java 782                    | sqlsact API 1019               | SQL_ATTR_                         |
| SQL パッケージ                   | sqlSetTypeCtx() API 742        | ACCESS_MODE 802                   |
| 作成、アプライ・プログラム用の             | SQLExtendedBind CLI 関数 917     | AUTOCOMMIT 802                    |
| 681                         | sqlflag precompile オプション 1190  | AUTO_IPD 802                      |
| 作成、キャプチャー・プログラム             | SQLGetData CLI 関数              | CLISCHEMA 802                     |
| 用の 681, 682                 | 構文 895                         | CLOSE_BEHAVIOR 802                |
| 作成、レプリケーション・アナラ             | SQLGetDescField CLI 関数         | CONNECTION_DEAD 802               |
| イザー用 681                    | 構文 905                         | CONNECTION_TIMEOUT 802            |
| レプリケーションの 663               | SQLIND フィールド、SQLDA の           | CONNECTTYPE 802                   |
| SQL プロシージャ                  | 1322                           | CONNECT_NODE 802                  |
| 作成 780                      | SQLj (embedded SQL for Java)   | CONN_CONTEXT 802                  |
| プリコンパイルおよびバインド・             | アプリケーション                       | CURRENT_CATALOG 802               |
| オプションのカスタマイズ 754            | 例 747                          | CURRENT_SCHEMA 802                |
| Windows 環境のセットアップ           | アプレット                          | DB2ESTIMATE 802                   |
| 755                         | 使用 772                         | DB2EXPLAIN 802                    |
| sqlca precompile オプション 1190 | 制約 746                         | DB2_SQLERRP 802                   |
| SQLColAttribute CLI 関数      | ビルド 778                        | ENLIST_IN_DTC 802                 |
| 構文 874                      |                                | INFO_ACCTSTR 802                  |

## SQL\_ATTR\_ (続き)

INFO\_APPLNAME 802  
 INFO\_USERID 802  
 INFO\_WRKSTNNAME 802  
 LOGIN\_TIMEOUT 802  
 LONGDATA\_COMPAT 802  
 MAXCONN 802  
 METADATA\_ID 802  
 ODBC\_CURSORS 802  
 PACKET\_SIZE 802  
 QUIET\_MODE 802  
 SYNC\_POINT 802  
 TRACE 802  
 TRACEFILE 802  
 TRANSLATE\_LIB 802  
 TRANSLATE\_OPTION 802  
 TXN\_ISOLATION 802  
 WCHARTYPE 802  
 SQL\_C\_BINARY 853  
 SQL\_C\_BIT 853  
 SQL\_C\_CHAR 853  
 SQL\_C\_DATE 853  
 SQL\_C\_DBCHAR 853  
 SQL\_C\_DOUBLE 853  
 SQL\_C\_FLOAT 853  
 SQL\_C\_LONG 853  
 SQL\_C\_SHORT 853  
 SQL\_C\_TIME 853  
 SQL\_C\_TIMESTAMP 853  
 SQL\_C\_TINYINT 853  
 SQL\_DESC\_  
 ALLOC\_TYPE  
     初期設定値 819  
 ARRAY\_SIZE  
     初期設定値 819  
 ARRAY\_STATUS\_PTR  
     初期設定値 819  
 AUTO\_UNIQUE\_VALUE 874  
     初期設定値 819  
 BASE\_COLUMN\_NAME 874  
     初期設定値 819  
 BASE\_TABLE\_NAME 874  
     初期設定値 819  
 BIND\_OFFSET\_PTR  
     初期設定値 819

## SQL\_DESC\_ (続き)

BIND\_TYPE  
     初期設定値 819  
 CASE\_SENSITIVE 874  
     初期設定値 819  
 CATALOG\_NAME 874  
     初期設定値 819  
 CONCISE\_TYPE 874  
     初期設定値 819  
 COUNT 874  
 DATA\_PTR  
     初期設定値 819  
 DATETIME\_INTERVAL\_CODE  
     初期設定値 819  
 DATETIME\_INTERVAL\_  
     PRECISION  
     初期設定値 819  
 DISPLAY\_SIZE 874  
     初期設定値 819  
 DISTINCT\_TYPE 874  
 FIXED\_PREC\_SCALE 874  
     初期設定値 819  
 INDICATOR\_PTR  
     初期設定値 819  
 LABEL 874  
 LENGTH 874  
     初期設定値 819  
 LITERAL\_PREFIX 874  
     初期設定値 819  
 LITERAL\_SUFFIX 874  
     初期設定値 819  
 LOCAL\_TYPE\_NAME 874  
     初期設定値 819  
 NAME 874  
     初期設定値 819  
 NULLABLE 874  
     初期設定値 819  
 NUM\_PREC\_RADIX  
     初期設定値 819  
 NUM\_PREX\_RADIX 874  
 OCTET\_LENGTH 874  
     初期設定値 819  
 OCTET\_LENGTH\_PTR  
     初期設定値 819  
 PARAMETER\_TYPE  
     初期設定値 819

## SQL\_DESC\_ (続き)

PRECISION 874  
     初期設定値 819  
 ROWS\_PROCESSED\_PTR  
     初期設定値 819  
 SCALE 874  
     初期設定値 819  
 SCHEMA\_NAME 874  
     初期設定値 819  
 SEARCHABLE 874  
     初期設定値 819  
 TABLE\_NAME 874  
     初期設定値 819  
 TYPE 874  
     初期設定値 819  
 TYPE\_NAME 874  
     初期設定値 819  
 UNNAMED 874  
     初期設定値 819  
 UNSIGNED 874  
     初期設定値 819  
 UPDATABLE 874  
     初期設定値 819  
 SQL\_DIAG\_  
 CLASS\_ORIGIN 798  
 COLUMN\_NUMBER 798  
 CONNECTION\_NAME 798  
 CURSOR\_ROW\_COUNT 798  
 DYNAMIC\_FUNCTION 798  
 DYNAMIC\_FUNCTION\_  
     CODE 798  
 MESSAGE\_TEXT 798  
 NATIVE 798  
 NUMBER 798  
 RETURNCODE 798  
 ROW\_COUNT 798  
 ROW\_NUMBER 798  
 SERVER\_NAME 798  
 SQLSTATE 798  
 SUBCLASS\_ORIGIN 798  
 SQL\_NTS 860  
 START DATABASE MANAGER コ  
     マンド 1279  
 startmode パラメーター 542, 547

STAY RESIDENT  
 CREATE FUNCTION (外部スカラ  
 ー) ステートメント 1618  
 CREATE FUNCTION (外部表) ス  
 テートメント 1644  
 CREATE PROCEDURE ステート  
 メント 1672, 1686

STMG\_CONTAINER 表 377  
 STMG\_CURR\_THRESHOLD 表 377  
 STMG\_DATABASE 表 377  
 STMG\_DBPARTITION 表 377  
 STMG\_DBPGROUP 表 377  
 STMG\_HIST\_THRESHOLD 表 377  
 STMG\_INDEX 表 377  
 STMG\_OBJECT 表 377  
 STMG\_OBJECT\_TYPE 表 377  
 STMG\_ROOT\_OBJECT 表 377  
 STMG\_TABLE 表 377  
 STMG\_TABLESPACE 表 377  
 STMG\_TBPARTITION 表 377  
 STMG\_THRESHOLD\_REGISTRY 表  
 377

strdel precompile/bind オプション  
 1190

STRDPAPY コマンド 478, 489

String Java データ・タイプ 782

striptblanks ファイル・タイプ修飾子  
 945, 1149, 1234

striptnulls ファイル・タイプ修飾子  
 945, 1149, 1234

STRJRNPf コマンド 663, 684

SUBS\_COLS (サブスクリプション列)  
 表 500

SUBS\_EVENT (サブスクリプショ  
 ン・イベント) 表  
 通知、イベントの 687, 699

SUBS\_MEMBR (サブスクリプショ  
 ン・メンバー) 表 478, 497, 502

SUBS\_SET (サブスクリプション・セ  
 ット) 表 507

subtableconvert ファイル・タイプ修飾  
 子 1149

Sybase  
 デフォルトの逆方向データ・タイ  
 プ・マッピング 427

Sybase (続き)  
 デフォルトの順方向タイプ・マッ  
 ping 1381  
 レプリケーションの制約事項  
 637, 645

syncpoint precompile オプション  
 1190

SYNONYM、DROP ステートメント  
 の 1572

SYSADM 668

SYSADM 権限  
 レプリケーション 663

SYSPROC.CAPTURE\_STORAGEEMGMT  
 \_INFO ストアード・プロシージャ  
 ー 377

SYSPROC.CREATE\_STORAGEEMGMT  
 \_TABLES ストアード・プロシージ  
 ャー 377

SYSPROC.DROP\_STORAGEEMGMT  
 \_TABLES ストアード・プロシージ  
 ャー 377

## T

TABLE HIERARCHY 文節、DROP  
 ステートメント 1572

TABLE 文節  
 CREATE FUNCTION (外部表) ス  
 テートメント 1644  
 DROP ステートメント 1572

target precompile オプション 1190

TCP/IP  
 AIX 134  
 Solaris オペレーティング環境  
 124  
 Windows 143

term パラメーター  
 アプライ用 478, 533  
 キャプチャー用 526, 542

term パラメーター (アプライ  
 用) 487, 540

term パラメーター (キャプチャー  
 用) 532, 548

text precompile/bind オプション  
 1190

TIME SQL データ・タイプ  
 C への変換 845

TIME データ・タイプ  
 CREATE TABLE ステートメント  
 の 1484  
 Java 782

timeformat ファイル・タイプ修飾子  
 945, 1149, 1234

TIMEOUT サーバー・オプション  
 有効な設定 437

TIMESTAMP データ・タイプ  
 C への変換 845  
 CREATE TABLE ステートメント  
 の 1484  
 Java 782

timestampformat ファイル・タイプ修  
 飾子 945, 1149, 1234

Tivoli Storage Manager (TSM)  
 クライアント・セットアップ  
 407  
 使用 407  
 タイムアウト問題の解決策 407  
 バックアップの制限 407  
 BACKUP DATABASE コマンド  
 での 407  
 RESTORE DATABASE コマンド  
 での 407

totalreespace ファイル・タイプ修飾  
 子 945, 1149

trace\_limit パラメーター  
 asncap コマンドで使用 542, 548  
 asncmd コマンドで使用 526,  
 532

transform precompile/bind オプション  
 1190

TRLREUSE パラメーター 478, 533

trlreuse パラメーター 487, 539

TSO 613, 614

TYPE 文節  
 DROP ステートメント 1572

## U

UCS-2 273

UDT (ユーザー定義タイプ)  
 サポートされない 448

UNDER 文節、CREATE VIEW ステートメント 1469  
Unicode (UCS-2)  
コード・ページ 273  
CCSID 273  
DB2 のサポート 273  
UNIQUE 文節  
ALTER TABLE ステートメント 1402  
CREATE TABLE ステートメント 1484  
UNIX  
DB2 オンライン・ドキュメンテーションのインストール 104  
UNQUIESCE コマンド 1266  
UOW (作業単位) 表  
ブルーニング 617, 628  
列、CCD 表の 687, 708  
UPDATE DATABASE CONFIGURATION コマンド 1276  
update-anywhere レプリケーション競合検出  
概要 637, 655  
要件 637, 647, 655  
再キャプチャー、変更の 637, 650  
定義、サブスクリプション・セットの 687, 712  
usedefaults ファイル・タイプ修飾子 945, 1149, 1234  
UTF-8 273

## V

validate precompile/bind オプション 1190  
VALIDPROC  
文節 603  
ALTER TABLE ステートメントの 1402  
VALIDPROC 文節 604  
VARBINARY SQL データ・タイプ  
C への変換 845  
VARCHAR データ・タイプ  
C への変換 845

VARCHAR データ・タイプ (続き)  
CREATE TABLE ステートメント 1484  
Java 782  
VARCHAR\_NO\_TRAILING\_  
BLANKS サーバー・オプション  
有効な設定 437  
VARGRAPHIC データ・タイプ  
C への変換 845  
Java 782  
VARIANT、CREATE TYPE (構造化)  
ステートメントの 1441  
version precompile オプション 1190  
VIEW HIERARCHY 文節、DROP ステートメント 1572  
VIEW 文節  
CREATE VIEW ステートメント 1469  
DROP ステートメント 1572  
Visual Basic  
サポートされる Windows パージョン 766

## W

wchartype precompile オプション 1190  
Web トラフィック・ポーリング  
レプリケーション・ステップ 1081  
Web 用インフォメーション・カタログ・センター  
インストール  
AIX で 216  
Windows NT 上で 188  
カスタマイズ 218  
WHERE 文節  
行サブセット 687, 716  
Windows  
インストール  
DB2 サーバー 396  
DB2 サーバー、ユーザー・アカウント 392  
応答ファイルのキーワード 163

Windows (続き)  
サービス  
キャプチャー・プログラムの開始 634  
Service Control Manager 634  
DB2 Personal Edition でのデータベースの移行 103  
Windows Service Control Manager (SCM) 634  
Windows サービス名 634  
WITH CHECK OPTION 文節、CREATE VIEW ステートメント 1469  
WITH DEFAULT 文節、ALTER TABLE ステートメント 1402  
WITH OPTIONS 文節  
CREATE VIEW ステートメント 1469  
WITH 共通表式 1317  
WITH 文節  
CREATE VIEW ステートメント 1469  
WRKREGINF コマンド 663, 687

## X

XML 1398  
XML Extender  
概要 1122  
使用可能なオペレーティング・システム 1122  
XML 文書  
DB2 に保管する 1122  
XTClient 構文 1086  
X/Open Company 796  
X/Open SQL CLI 796

## Z

z/OS  
ウェアハウス・エージェント  
インストール 197  
開始済みとしてデーモンの開始 226  
概要 225  
デーモンの開始 202

## z/OS (続き)

ウェアハウス・エージェント (続き)

ユーザー定義のプログラム

203

DataPropagator ステップの自動化に使用 211

ウェアハウス・ステップ、スケジューリング 204

ウェアハウス・トランスフォーマー

環境変数変数データでの文字の削減 208

DB2 for z/OS での設定 206

サーバー、接続 663

ユーザー定義のプログラム

ステップ完了 232

## z/OS サーバー

接続 665

## [特殊文字]

\*.APP.log ファイル 478, 483

\*.err ファイル 478, 486

\*.sqe ファイル 478, 486, 487

? (クエスチョン・マーク)

EXECUTE パラメーター・マーカ

ー 1602







Printed in Japan

日本アイ・ビー・エム株式会社

〒106-8711 東京都港区六本木3-2-12