

IBM DB2 Connect 9.7



バージョン 9 リリース 7



DB2 Connect ユーザーズ・ガイド
2009 年 11 月更新版

IBM DB2 Connect 9.7



バージョン 9 リリース 7



DB2 Connect ユーザーズ・ガイド
2009 年 11 月更新版

ご注意

本書および本書で紹介する製品をご使用になる前に、199 ページの『付録 B. 特記事項』に記載されている情報をお読みください。

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

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

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

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

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

原典： SC27-2434-01
IBM DB2 Connect 9.7
Version 9 Release 7
DB2 Connect User's Guide
Updated November, 2009

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第1刷 2009.11

© Copyright International Business Machines Corporation 1993, 2009.

目次

本書について	vii
------------------	-----

第 1 部 DB2 Connect の概念 1

第 1 章 DB2 Connect. 3

DB2 Connect の製品オファリング	3
DB2 Connect バージョン 8 で追加された機能	3
ホスト・データベース	5
DB2 Connect と SQL ステートメント	5
DB2 Connect 管理ユーティリティー	6
InfoSphere Federation Server および DB2 Connect	7

第 2 章 分散リレーショナル・データベース体系 9

DRDA とデータ・アクセス	9
DB2 Connect と DRDA	10
リモート作業単位	11
分散要求	12

第 3 章 DB2 Connect のシナリオ 15

ホスト・データベースへのダイレクト・アクセス	15
DB2 Connect Personal Edition を使用した System z ホストまたは IBM i DB2 データへのアクセス	17
接続サーバーとしての DB2 Connect サーバー製品	18
DB2 Connect と Web アプリケーション	20
DB2 Connect と IBM WebSphere	21
Java アプリケーション・サーバーとしての DB2 Connect	22
Web サーバー上の DB2 Connect	23
DB2 Connect とアプリケーション・サーバー	24
DB2 Connect とトランザクション処理モニター	27

第 2 部 DB2 Connect リファレンス 31

第 4 章 データベース・ディレクトリーの更新 33

システム・データベース・ディレクトリーの値	33
ノード・ディレクトリーの値	34
DCS ディレクトリーの値	35
ディレクトリー・カスタマイズ・ワークシート	40
同一のデータベースへの複数項目の定義	41
BiDi データの処理	41

第 5 章 DB2 Connect のセキュリティー 45

DB2 Connect を介したトラステッド接続	45
CLI を使用したトラステッド接続の作成および終了	46
CLI を使用したトラステッド接続のユーザーの切り替え	48
DB2 Connect の認証に関する考慮事項	50

Kerberos のサポート	51
z/OS セキュリティーに関するヒント	52
DB2 Connect でサポートされる認証タイプ	54

第 6 章 アプリケーションとユーティリティーのバインド (DB2 Connect) 57

第 7 章 マルチサイト更新 61

コントロール・センターを使ったマルチサイト更新の使用可能化	62
コントロール・センターを使ったマルチサイト更新のテスト	62
マルチサイト更新と同期点管理プログラム	63
DB2 Connect と XA 準拠トランザクション・マネージャーとの構成	64
疎結合トランザクションのための DB2 Connect サポート	65

第 8 章 DB2 Connect によるデータの移動 67

第 9 章 SQLCODE マッピング 71

SQLCODE マッピングをオフにする	71
SQLCODE マッピングの調整	71

第 10 章 データベース・システム・モニターと DB2 Connect. 77

リモート・クライアントの接続のモニター	77
Windows パフォーマンス モニタを使ったパフォーマンスのモニター	77
GET SNAPSHOT コマンドの使用	78
DCS アプリケーション状況	80
ヘルス・モニターおよびアラート	85
DB2 for z/OS ヘルス・モニターの概要	85
DB2 for z/OS ヘルス・モニターを開始、停止、およびリフレッシュする	86
推奨処置を表示、サブミット、および保管する	87
ヘルス・アラート・サマリーを表示する	90
ヘルス・アラート・オブジェクトを表示する	92

第 3 部 高可用性と DB2 Connect 93

第 11 章 ホスト・データベース接続のための高可用性とロード・バランシング 95

第 12 章 自動クライアント・リルートについての説明およびセットアップ (DB2 Connect) 97

第 13 章 クライアント接続ディストリビューター・テクノロジーに対する自動クライアント・リルトの構成	99
---	----

第 4 部 チューニングと DB2 Connect	101
--	------------

第 14 章 DB2 Connect のパフォーマンスの考慮事項	103
--	-----

第 15 章 ODBC アクセスの最適化	107
--------------------------------	-----

第 16 章 アプリケーション設計	109
-----------------------------	-----

第 17 章 接続の管理	113
接続プール	113
接続コンセントレーター	115
接続プールと接続コンセントレーター	121
WebSphere MQ トランザクション・マネージャーおよび DB2 for z/OS で必要な接続コンセントレーター	121

第 18 章 DB2 Connect サーバー Sysplex のサポート	123
System z SYSPLEX を活用する際の考慮事項	123
DB2 Sysplex の活用	124
Sysplex の構成の要件	125

第 19 章 クライアント Sysplex サポート	127
トランザクション・レベルのワークロード・バランシング (クライアント・サイド)	127
トランザクション・レベルのワークロード・バランシングの構成 (クライアント・サイド)	129
自動クライアント・リルト (クライアント・サイド)	131
自動クライアント・リルトの構成 (クライアント・サイド)	133
クライアント・アフィニティの構成	135
クライアント Sysplex サポートの使用に関する制限	140

第 20 章 DB2 Connect のチューニング	143
ホスト・データベースのチューニング	145
ネットワークのチューニングに関する考慮事項	145
システム・リソースの競合	147
DB2 Connect のパフォーマンスに関するトラブルシューティング	147
DB2 for z/OS のチューニング	148
DB2 Connect のデータ転送速度の向上	148
追加照会ブロック	149
RFC-1323 ウィンドウ・スケーリング	150
ホスト・データの変換	151
文字データのデータ・タイプ	151

ネットワーク・ハードウェア	151
-------------------------	-----

第 21 章 CLI/ODBC アプリケーションのパフォーマンス調整	155
--	-----

第 5 部 トラブルシューティング	157
------------------------------------	------------

第 22 章 トラブルシューティング DB2 Connect	159
関係のある情報の収集	159
初期接続が正常に行われなかった場合	159
初期接続後に発生する問題	160
診断ツール	162

第 23 章 DB2 Connect での DB2 トレース	163
db2trc を使用した DB2 トレースの取得	163
DB2 トレース・ファイルのダンプ	164
DB2 トレース・ファイルのフォーマット	165

第 24 章 DRDA トレース・ファイル	167
トレース・ユーティリティ	167
トレース出力	168
トレース出力ファイルの分析	169
トレース出力ファイル・サンプル	170
DRDA トレースの後続のバッファ情報	175

第 6 部 メッセージ	177
------------------------------	------------

第 25 章 一般的な DB2 Connect の問題	179
---------------------------------------	-----

第 7 部 付録	183
---------------------------	------------

付録 A. DB2 技術情報の概説	185
DB2 テクニカル・ライブラリー (ハードコピーまたは PDF 形式)	186
DB2 の印刷資料の注文方法	189
コマンド行プロセッサから SQL 状態ヘルプを表示する	190
異なるバージョンの DB2 インフォメーション・センターへのアクセス	190
DB2 インフォメーション・センターでの希望する言語でのトピックの表示	190
コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの更新	191
コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの手動更新	193
DB2 チュートリアル	195
DB2 トラブルシューティング情報	196
ご利用条件	196

付録 B. 特記事項	199	索引	203
----------------------	-----	--------------	-----

本書について

「DB2 Connect ユーザーズ・ガイド」は、DB2 Connect™ 製品について知り、それを使用するために必要なすべての情報を提供しています。DB2 Connect の概念については、DB2 Connect とネットワーク環境内の他の部分との関係を示す典型的なシナリオを使用して説明しています。データベース・ディレクトリー、システム間のセキュリティー、マルチサイト更新、データの移動、および DB2 Connect のモニターに関する考慮事項も扱っています。また、ネットワーク環境内での高可用性を DB2 Connect がどのようにサポートするかを示します。DB2 Connect およびネットワーク全体で良好なパフォーマンスを得ることについても紹介しており、起きる可能性のある問題のトラブルシューティングについて説明しているトピックもあります。

本書の対象読者

システム管理者、データベース管理者、システム通信の専門技術者が本書の全体または一部をお読みになることを想定しています。

第 1 部 DB2 Connect の概念

第 1 章 DB2 Connect

DB2 Connect では Linux[®]、UNIX[®]、および Windows[®] オペレーティング・システムで稼働する e-ビジネスおよび他のアプリケーションのために、IBM[®] メインフレーム・データベースに対する高速かつ堅固な接続を提供します。

DB2 Connect Personal Edition では System z[®] および IBM Power Systems[™] サーバーに直接接続するのに対し、DB2 Connect サーバー製品では、クライアントが DB2 Connect ゲートウェイを介して System z および IBM Power Systems サーバーにアクセスできるようにするために間接接続します。DB2 Connect サーバー製品では、お客様の環境に適した製品をお選びいただけるように、固有なパッケージングとライセンスのソリューションを各種ご用意しています。

DB2 Connect の製品オファリング

DB2 Connect には接続に関する複数のソリューションがあり、それには DB2 Connect Personal Edition、および以下に挙げる複数の DB2 Connect サーバー製品が含まれます。

- DB2 Connect Enterprise Edition
- DB2 Connect Application Server Edition
- DB2 Connect Unlimited Edition for System z
- DB2 Connect Unlimited Edition for System i[®]

DB2 Connect の製品オファリングの詳細については、www.ibm.com/software/data/db2/db2connect/ を参照してください。

DB2 Connect バージョン 8 で追加された機能

このセクションでは、DB2 Connect バージョン 8 で導入された機能拡張のサマリーを提供します。DB2 Connect の機能に影響する DB2[®] バージョン 9 で導入された変更のリストを確認したい場合、以下のトピックを参照してください。

- DB2 Connect バージョン 9.5 フィックスパックのサマリー
- DB2 Connect バージョン 9.1 フィックスパックのサマリー

DB2 Connect バージョン 8 リリース 2 で追加された機能

DB2 Connect バージョン 8.2 では、次のように機能が拡張されました。

- 自動クライアント・リルート

サーバーまたは DB2 Connect サーバーに対する TCP/IP 接続が失われた場合、代替サーバーが存在すれば、クライアントはその接続を自動的に再確立しようとします。代替サーバーは、サーバー・インスタンスで指定します。その代替サーバーのロケーションは、接続時にクライアントに送信されます。

- データ暗号化

クライアント/サーバー通信で、ネットワーク上を行き来するユーザー・データの暗号化機能が用意されました。

DB2 Connect バージョン 8 リリース 1 (すべてのフィックスパックと修正レベルを含む) で追加された機能

DB2 Connect バージョン 8.1 では、次のように機能が拡張されました。

- 長い SQL ステートメント (最高 2MB まで) のサポート

最高 2 MB までの SQL ステートメントが CLI アプリケーションや JDBC アプリケーションを実行できるようになりました。ただし、組み込みインターフェースには、引き続き 64K という制限があります。

- SQL ステートメントの発信元を識別する診断情報

どのアプリケーション・プログラムが DB2 for z/OS® の動的 SQL ステートメント・キャッシュに特定のステートメントを送出したのかを判別するための機能が用意されました。

- 列方向の入力配列

アプリケーションが 1 つの SQL ステートメントに対して複数のパラメーター・セットを提供できるようになりました。

- ネットワーク時間のモニター

新しいモニター・エレメントを使用して、データベース・レベルまたはアプリケーション・レベルで、データベース・アクティビティーとネットワーク・トラフィックをより詳細に把握できるようになりました。

- DB2 CLI の動的両方向スクロール・カーソルのサポート

DB2 Universal Database (UDB) for z/OS バージョン 8.1 以降のサーバーにアクセスするときに、DB2 CLI で動的両方向スクロール・カーソルがサポートされるようになりました。

- eWLM のサポート

ボトルネックを判別するために、ミドルウェア・グループを経由した終端間の作業単位をモニターする機能が用意されました。

- DB2 の ping コマンドの機能強化

DB2 ping コマンドで、要求と応答のパケット・サイズを指定できるようになりました。

注: DB2 Connect では、バージョン 7 クライアントから、バージョン 9 ゲートウェイを経由して、ホストに向けて PING コマンドを発行することをサポートしていません。

ホスト・データベース

データベース という用語は、この資料を通してリレーショナル・データベース管理システム (RDBMS) の記述について使用されます。DB2 Connect が通信するその他のシステムでは、若干異なる概念を記述するためにデータベースという用語が使用される場合があります。DB2 Connect において、データベースという用語は次のものも指します。

System z

DB2 for z/OS. LOCATION NAME によって識別される DB2 for z/OSサブシステム。LOCATION NAME は、TSO にログインした後で、利用可能ないずれかの照会ツールを使って以下に示す SQL 照会を発行することで判別できます。

```
select current server from sysibm.sysdummy1
```

さらに、LOCATION NAME はブートストラップ・データ・セット (BSDS) でも定義されており、DSNL004I メッセージ (LOCATION=location) に表示されます。これは、分散データ機能 (DDF) の起動時に書き込まれます。LOCATION NAME は、最高 8 個までの別名ロケーション名をサポートしているので、アプリケーションは、バージョン 8 の z/OS サーバーにアクセスするためにさまざまな dbalias 名を使用できます。z/OS の -display ddf コマンドを使用すれば、DB2 サーバーのロケーション名、ドメイン名、IP アドレス、ポート番号を確認できます。

VSE データベース・パーティション内で稼働している DB2 for VSE。その DBNAME によって識別される。

VM CMS 仮想計算機内で稼働している DB2 for VM。その DBNAME によって識別される。

IBM Power Systems サーバー

DB2 for IBM i。IBM i オペレーティング・システムの統合された部分。IBM Power Systems サーバー上には 1 つのデータベースだけを配置できます (ただし、独立した補助ストレージ・プールを使用するようにシステムを構成している場合は別です)。

DB2 Connect と SQL ステートメント

DB2 Connect は、アプリケーション・プログラムによりサブミットされた SQL ステートメントを、IBM メインフレーム・データベース・サーバーに転送します。

DB2 Connect は、以下のようなサポートされる DB2 API (アプリケーション・プログラミング・インターフェース) を使用して生成されるものを含め、有効な SQL ステートメントのほとんどすべてを転送することができます。

- JDBC
- SQLJ
- ADO.NET
- OLE DB
- ODBC
- Perl

- PHP
- pureQuery
- Python
- Ruby
- DB2 CLI
- 組み込み SQL

組み込み SQL サポート

組み込み SQL の処理方式には、静的 SQL と動的 SQL という 2 つのタイプがあります。静的 SQL は、事前に処理することによって、SQL ステートメントの実行に要する時間を最小化します。動的 SQL は、IBM メインフレーム・データベース・サーバーにサブミットされた時点で処理されます。動的 SQL はより柔軟な実行が可能です。処理速度が遅くなってしまうことがあります。静的 SQL を使用するか、動的 SQL を使用するかは、アプリケーション・プログラマーが判断します。DB2 Connect はそのどちらのタイプもサポートしています。

IBM メインフレーム・データベース・サーバーが異なると、SQL のインプリメント方法も異なります。DB2 Connect は、共通の IBM SQL に加えて、DB2 for z/OSDB2 Server for VM and VSE (以前の SQL/DS)、および DB2 for IBM i でインプリメントされる SQL も完全にサポートしています。データベースの独立性を維持するには、IBM SQL の利用を強くお勧めします。

DB2 Connect 管理ユーティリティー

重要: コントロール・センターおよびそれに関連したコンポーネントは、バージョン 9.7 で非推奨となり、将来のリリースで除去される可能性があります。詳しくは、「DB2 バージョン 9.7 の新機能」のトピック『コントロール・センター・ツールおよび DB2 管理サーバー (DAS) が推奨されなくなった』を参照してください。

以下のユーティリティーは、DB2 Connect の管理者に役立ちます。

- コマンド行プロセッサ (CLP) は、IBM メインフレーム・データベース・サーバー・データベースに対して SQL ステートメントを発行します。コマンド行プロセッサは、指定したデータベースに SQL ステートメントを渡します。
- DB2 コマンド・センターは、コマンド行プロセッサ (CLP) へのグラフィカル・インターフェースを提供します。
- インポートおよびエクスポート・ユーティリティーは、ワークステーション上のファイルと、IBM メインフレーム・データベース・サーバー・データベースの間で、データのロード、インポート、およびエクスポートを実行します。これらのファイルは、次にデータをデータベース、スプレッドシート、およびワークステーションで稼働中のその他のアプリケーションにインポートするために使用することができます。
- DB2 Connect サーバー製品を実行している場合、「イベント・ビューアー」と「パフォーマンス・モニター」を使用できます。イベント・ビューアーを使えば、DB2 Connect によってログに記録された例外イベントを表示することができます。パフォーマンス・モニターを使えば、DB2 Connect サーバーのパフォーマンスのモニターと管理をローカルまたはリモートに行うことができます。

- DB2 コントロール・センターは、DB2 Connect サーバーのすべての面を管理およびモニターすることができます。さらに、管理者はこれを使って DB2 for z/OS のデータベース・オブジェクト (表、ビュー、バッファ・プール、スレッドなど) を操作することもできます。
- データベース・システム・モニター・ユーティリティーは、システム管理者がシステム接続をモニターできるようにします。この機能は、DB2 Connect がサーバーとして稼働している時のみ使用可能です。このユーティリティーはまた、システム管理者がエラーの原因を判別するのに役立ちます。システム管理者は、クライアント・アプリケーションを、IBM メインフレーム・データベース・サーバーで実行中の対応ジョブと関連させることができます。

注: 前のリリースでは、コントロール・センターなどの DB2 グラフィカル管理ツールはすべてのプラットフォームでサポートされていました。バージョン 9 では、DB2 グラフィカル管理ツールは Windows x86、Windows x64 (AMD64/EM64T)、x86 上の Linux、および AMD64/EM64T 上の Linux でのみサポートされます。DB2 コマンド行プロセッサ (CLP) を使用した管理は、すべてのプラットフォームで可能です。

InfoSphere Federation Server および DB2 Connect

InfoSphere™ Federation Server はマルチベンダー・データ・ソース間でデータへのアクセスおよびデータの統合を提供する個別のプロダクト・オファリングですが、DB2 Connect は既存のホストおよびミッドレンジ・サーバーにある大容量データを活用できるようにします。

InfoSphere Federation Server は、データ・ソースの集合をあたかも単一のソースであるかのように表示および操作できるようにするので、情報を統合するのに役立ちます。呼び出し側アプリケーションに対して、データ・ソース・アクセスを完全に透過的にします。InfoSphere Federation Server は、DB2 Connect サーバー製品と連動して機能します。InfoSphere Federation Server は、DB2 ファミリー製品、Informix®、Oracle、Sybase、Teradata、および Microsoft® SQL Server データベースへのネイティブな読み取りおよび書き込みアクセスを提供します。さらに、InfoSphere Federation Server では、Documentum、IBM Lotus® Extended Search、表構造ファイル、および XML などの非リレーショナルおよびライフ・サイエンス・データ・ソースへの読み取りアクセスが可能です。フェデレーテッド・システム内のデータの照会を定式化するために使用できます。

第 2 章 分散リレーショナル・データベース体系

分散リレーショナル・データベース体系 (Distributed Relational Database Architecture™、DRDA®) は、複数のデータベース・システム (IBM と非 IBM の両方)、およびアプリケーション・プログラムの協働を可能にする一連のプロトコルです。DRDA を使用するリレーショナル・データベース管理製品であれば、どのような組み合わせであれ接続して、分散リレーショナル・データベース管理システムを形成することができます。DRDA は、何をどのように交換すべきかを定義することによって、システム間の通信を調整します。

作業単位

作業単位 (UOW) とは、単一の論理トランザクションのことです。作業単位は、全操作の実行が成功した場合の SQL ステートメントの順序、もしくは全体が失敗したと見なされた場合の SQL ステートメントの順序のどちらかによって構成されます。

分散作業単位

分散作業単位 (DUOW) (マルチサイト更新ともいう) では、1 つの作業単位内に複数のデータベース・サーバーが関係しています。DUOW には、以下のような特性があります。

- 作業単位ごとに 2 つ以上のデータベース管理サーバーが更新される。
- アプリケーションが作業の分散を誘導し、コミットを開始する。
- 作業単位ごとに複数の要求が存在できる。
- 1 つの要求につき 1 つのデータベース管理サーバーが存在する。
- コミットメントが複数のデータベース・サーバーにわたって調整される。

DRDA とデータ・アクセス

DRDA ではデータベース通信プロトコルが定義されますが、アプリケーション・プログラマーが使用するプログラミング・インターフェース、または API は定義されません。一般に、DRDA はアプリケーション・プログラムによって使用され、ターゲット DRDA サーバーが実行できる要求を渡します。現在使用可能なすべての DRDA サーバーは、アプリケーション・プログラムによって DB2 Connect 経由で送られる SQL 要求を実行することができます。

IBM は Windows、UNIX、および Linux オペレーティング・システム用の、SQL 要求を生成するためのツールをアプリケーション・プログラマーに提供します。これらのツールは DB2 クライアントの一部です。DB2 データベース・マネージャーでは、ADO.NET、JDBC、SQLJ、PHP、Perl DBI、組み込み SQL、DB2 コール・レベル・インターフェース (DB2 コール・レベル・インターフェース)、OLE DB などの複数のプログラミング・インターフェースがサポートされています。これらの API は、各種のプログラミング言語でアプリケーションを構築するプログラマーが使用します。

DB2 Connect と DRDA

DB2 Connect は、DRDA 体系をインプリメントしており、DB2 for IBM i、DB2 for IBM Power Systems、DB2 for z/OS、DB2 Server for VM and VSE、およびその他の DRDA 準拠のデータベース・サーバー内に保管されているデータにアクセスする際のコストと複雑さを減らすことができます。DRDA 体系を十分に活用することによって、DB2 Connect は、お客様の要求するシステム管理特性を備えた、高パフォーマンスかつ低コストのソリューションを提供します。

DRDA の用語で、アプリケーション・リクエスター (AR) とは、分散接続のアプリケーション側を処理するコードです。AR はデータを要求するアプリケーションです。DB2 Connect はアプリケーション・プログラムの代わりに、アプリケーション・リクエスターとして機能します。この場合、アプリケーション・プログラムは DB2 Connect ワークステーションに対してローカルにすることも、DB2 Connect とは別のリモート・クライアントに配置することもできます。

アプリケーション・サーバー (AS) とは、接続のデータベース側を処理するコードです。

DRDA は、アプリケーション・リクエスターとサーバーの間の多層接続もサポートしています。このトポロジでは、アプリケーション・リクエスターの接続先になるサーバーはアプリケーション・サーバーですが、それよりも下流にある他のすべてのサーバーはデータベース・サーバー (DS) と呼ばれ、アプリケーション・リクエスターとは直接対話しません。また、アプリケーション・リクエスターと最終のデータベース・サーバーの間にある各アプリケーション・サーバーまたはデータベース・サーバーは、データベース要求を発信するシステムでもなければ、要求に対するデータベース機能を実行するシステムでもないので、そのような役割を強調するために中間サーバーともいいます。DB2 Connect は、データベース・サーバーと中間サーバーの使用をサポートしています。

図 1 は、DB2 Connect ワークステーションと、ローカル・クライアントだけが配置されている場合の IBM メインフレーム・サーバーとの間のデータの流れを示します。

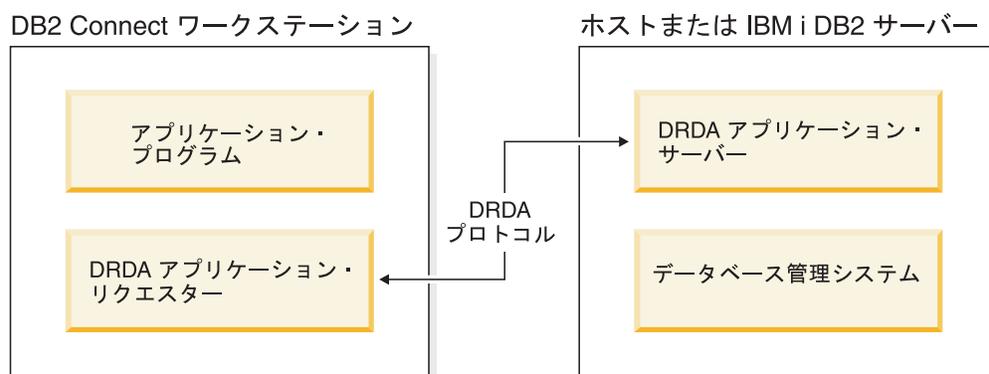


図 1. DB2 Connect サーバーと、ホストまたは IBM メインフレーム・サーバーとの間のデータ・フロー

DRDA サーバーのデータベース管理システムと IBM Data Server Client との間の接続を実施するために、DRDA は以下の体系を使用します。

- Character Data Representation Architecture (CDRA)
- 分散データ管理体系 (DDM)
- 定様式データ・オブジェクト・コンテンツ体系 (FD:OCA)
- 伝送制御プロトコル/インターネット・プロトコル (TCP/IP)

これらの体系は、ビルディング・ブロックとして使われます。ネットワーク上を流れるデータ・ストリームは、DRDA 体系によって指定されます。DRDA 体系は、分散リレーショナル・データベース・アクセスをサポートする、データ・ストリーム・プロトコルを文書化したものです。

いろいろなタイプの通信情報およびアクセス中の DRDA サーバー・データベース名を記録しているディレクトリーを使用することにより、要求を正しい宛先へ送信することができます。

リモート作業単位

リモート作業単位は、ユーザーまたはアプリケーション・プログラムが、作業単位ごとに 1 つのロケーションにおいてデータの読み取りまたは更新を行えるようにします。それは、1 つの作業単位内での 1 つのデータベースへのアクセスをサポートします。アプリケーション・プログラムは、複数のリモート・データベースを更新することができますが、1 つの作業単位内では 1 つのデータベースだけにアクセスすることができます。

リモート作業単位には、以下のような特性があります。

- 作業単位ごとに複数の要求 (SQL ステートメント) をサポートします。
- 作業単位ごとに複数のカーソルをサポートします。
- それぞれの作業単位が更新できるデータベースは 1 つだけです。
- アプリケーション・プログラムは作業単位のコミットまたはロールバックのいずれかを行います。特定のエラー状況では、データベース・サーバーまたは DB2 Connect が作業単位をロールバックする可能性があります。

例として、12 ページの図 2 には送金業務用アプリケーションが稼働しているデータベース・クライアントが示されています。このアプリケーションがアクセスするデータベースには、当座預金口座と普通預金口座の表とともに取引手数料の一覧も入っています。このアプリケーションでは次のような作業を行わなければなりません。

- 送金金額をユーザー・インターフェースから受け取る。
- その金額を普通預金口座から差し引いて、新しい残高を算出する。
- 料率表を読み込んで、算出した残高による普通預金口座に対する取引手数料を算出する。
- その取引手数料を普通預金口座から差し引く。
- 送金金額を当座預金口座に追加する。
- このトランザクション (作業単位) をコミットする。

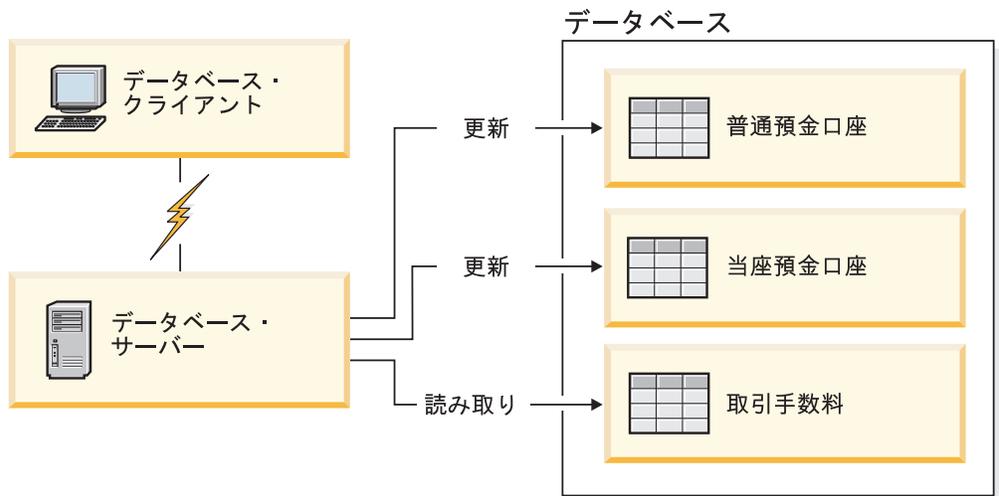


図2. 1つのトランザクションで単一のデータベースを使用する

このようなアプリケーションをセットアップするには、以下の作業が必要になります。

1. 普通預金口座、当座預金口座、取引手数料の一覧表のそれぞれに対応する表を、同一のデータベース内に作成する。
2. 物理的にリモートである場合は、適切な通信プロトコルを使うようデータベース・サーバーをセットアップする。
3. 物理的にリモートである場合は、ノードとデータベースをカタログし、そのデータベースがデータベース・サーバー上で識別されるようにする。
4. タイプ 1 接続を指定してアプリケーション・プログラムをプリコンパイルする。つまり、PREP コマンドで `CONNECT(1)` を指定する。

分散要求

分散要求 は分散データベース機能の 1 つで、アプリケーションやユーザーが、単一のステートメントで 2 つ以上の DBMS またはデータベースを参照する SQL ステートメントをサブミットすることを可能にします。例えば、2 つの異なる DB2 for z/OS サブシステムの表間で結合を行うことができます。

DB2 Connect は、複数のデータベースおよび DBMS 間での分散要求をサポートしています。例えば、DB2 表と Oracle ビューとの間で UNION 操作を実行することができます。サポートされている DBMS には、DB2 ファミリー製品 (DB2 Database for Linux, UNIX, and Windows、DB2 for z/OS、および DB2 for i など) のメンバー、および Oracle が含まれます。DB2 Connect を InfoSphere Federation Server と併用する場合は、マルチベンダー・サポートを利用できます。

分散要求は、データベース・オブジェクトのロケーションの透過性を実現するものです。(表やビューの中の) 情報が移動した場合に、その情報を要求するアプリケーションには変更を加えることなく、その情報への参照 (ニックネーム という) を更新できます。分散要求はまた、すべての DB2 SQL ダイアレクトをサポートしてい

るわけではない、あるいは特定の最適化機能をサポートしていない DBMS に対する補正 も行います。そのような DBMS では実行できない操作 (再帰的 SQL など) は、DB2 Connect の下で実行されます。

分散要求は半自律型の 方式で機能します。例えば、Oracle オブジェクトへの参照を含む DB2 照会を、Oracle アプリケーションが同じサーバーにアクセスしている最中にサブミットすることができます。また、分散要求は Oracle オブジェクトや他の DBMS オブジェクトへのアクセスを、独占したり (整合性やロックングに関する制約以上に) 制限したりすることはありません。

分散要求機能のインプリメンテーションは、DB2 Connect の単一インスタンス (フェデレーテッド (連合) データベースとして機能するデータベース) と、1 つまたは複数のリモート・データ・ソースで構成されます。フェデレーテッド・データベース には、データ・ソースとその特性を識別するカタログ項目が含まれています。データ・ソース は DBMS とデータで構成されます。アプリケーションがフェデレーテッド・データベースに接続する方法は、他の DB2 データベースに接続する方法と同じです。DB2 Connect のフェデレーテッド・データベースには、ユーザー・データを管理するためのライセンスは含まれていません。このデータベースの使用目的は、データ・ソースに関する情報の格納だけに限定されています。

フェデレーテッド・システムをセットアップした後は、複数のデータ・ソース内の情報に対して、それが 1 つの大規模なデータベース内にあるかのようにアクセスできます。1 つのフェデレーテッド・データベースにユーザーやアプリケーションが照会を送ると、必要に応じてそのフェデレーテッド・データベースは、DB2 ファミリーおよび Oracle のシステムからデータを取り出します。ユーザーやアプリケーションは照会でニックネームを指定します。これらのニックネームはデータ・ソースの中にある表やビューへの参照となるものです。エンド・ユーザーの観点からすると、ニックネームは別名のようなものです。

分散要求のパフォーマンスには多くの要因が影響します。最も重要なのは、データ・ソースとそのオブジェクトに関する正確な最新情報が、フェデレーテッド・データベースのグローバル・カタログに格納されるようにすることです。この情報は DB2 オプティマイザーによって利用され、データ・ソースでの評価に使われる操作をプッシュダウンするかどうかを決めるのに影響します。

第 3 章 DB2 Connect のシナリオ

DB2 Connect には、IBM メインフレーム・データベースのアクセス要件に応じた多様なソリューションが用意されています。ここでは、特定の要件や環境に適用できるいくつかのシナリオの概要を説明します。

ホスト・データベースへのダイレクト・アクセス

DB2 Connect の基本フィーチャーは、ワークステーション上で稼働しているデスクトップ・アプリケーションから、ホスト・データベースへの直接接続を提供することです。DB2 Connect ライセンスと共に IBM Data Server Driver Package を利用すれば、このソリューションを最も簡単に実現できます。

DB2 Connect Personal Edition がインストールされているそれぞれのワークステーションは、DB2 for z/OS、DB2 for IBM i、および DB2 Database for Linux, UNIX, and Windows サーバーに対し、TCP/IP による直接接続を確立することができます。さらに、アプリケーションが同一トランザクション内において DB2 ファミリーの複数のデータベースへ接続して更新することも可能です。このとき、2 フェーズ・コミットのプロトコルにより、データの整合性が完全に保たれます。

16 ページの図 3 は DB2 Connect Personal Edition がインストールされたワークステーションから、IBM メインフレーム・データベース・サーバーへの直接接続を示します。

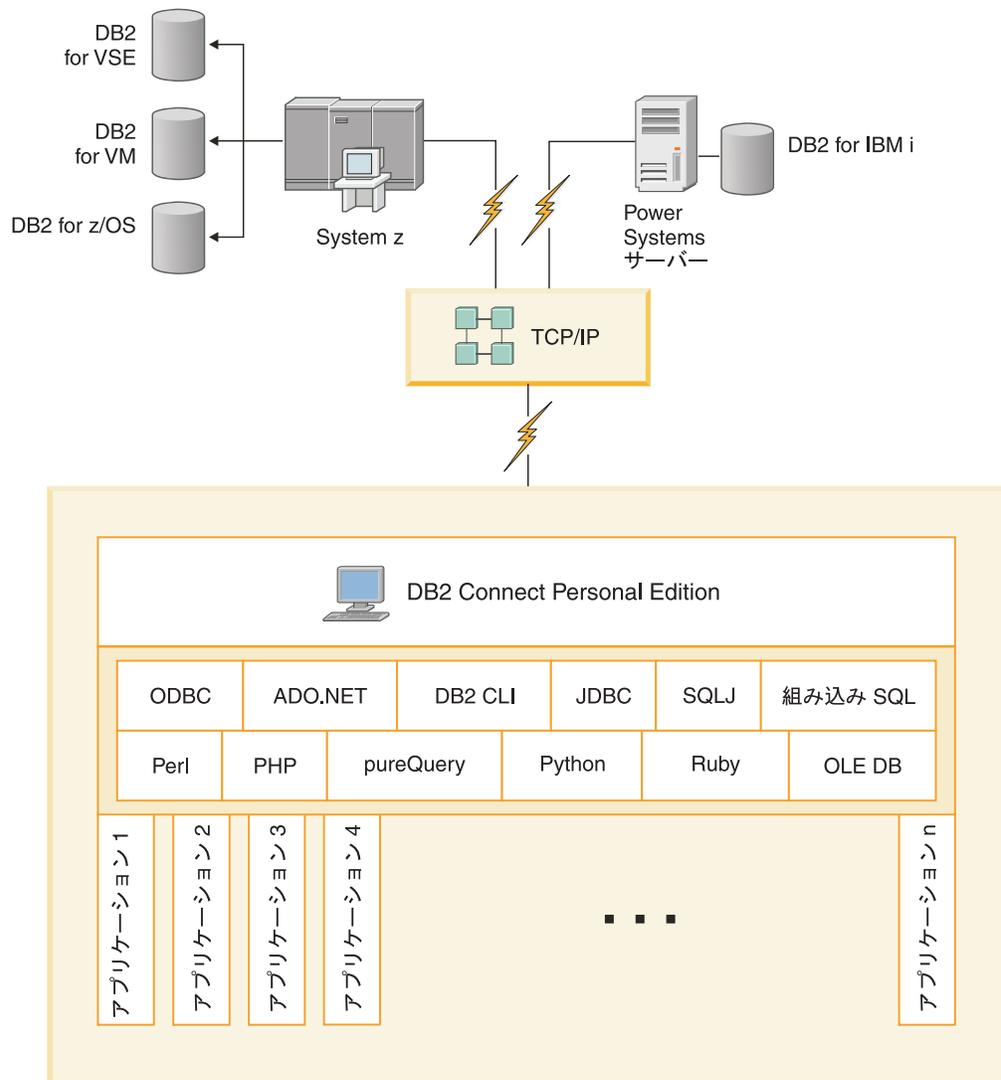


図3. DB2 Connect と、IBM メインフレーム・データベース・サーバーとの直接接続

注:

1. DB2 を DB2 Connect Personal Edition ワークステーションにインストールする必要はありません。DB2 Connect Personal Edition ワークステーションでの完全なリレーショナル・データベース管理システムを希望される場合は、DB2 をご注文ください。
2. IBM Data Server Client のすべての機能は、DB2 Connect Personal Edition で使用可能です。
3. SYSPLEX を使用可能にした DB2 for z/OS データベース・サーバーに対する接続が失われた場合、クライアントはその接続を自動的に再確立しようとします。

DB2 Connect Personal Edition を使用した System z ホストまたは IBM i DB2 データへのアクセス

中間サーバーのない直接接続は、非常に便利な望ましい構成です。IBM メインフレーム・データベース・サーバーが TCP/IP 接続をサポートする状況下では特に、直接接続が望ましいと言えます。そのような構成では、各 DB2 Connect ワークステーションは、IBM メインフレーム・データベース・サーバーとの直接接続を確立します。

TCP/IP 接続のためには、IBM メインフレーム・データベースが TCP/IP をサポートしていなければなりません。以下のバージョンがネイティブ TCP/IP 接続をサポートしています。

- DB2 for z/OS バージョン 7.1 以降
- DB2 for IBM i バージョン 5 リリース 1 以降
- DB2 Server for VM and VSE バージョン 7 以降

IBM メインフレーム・データベース・サーバーに接続するには、IBM Data Server Client へ追加することができる DB2 Connect ライセンスが必要です。

18 ページの図 4 は、DB2 Connect Personal Edition をインストールされ、しかも IBM メインフレーム・データベース・サーバーに直接接続されたワークステーションを示しています。

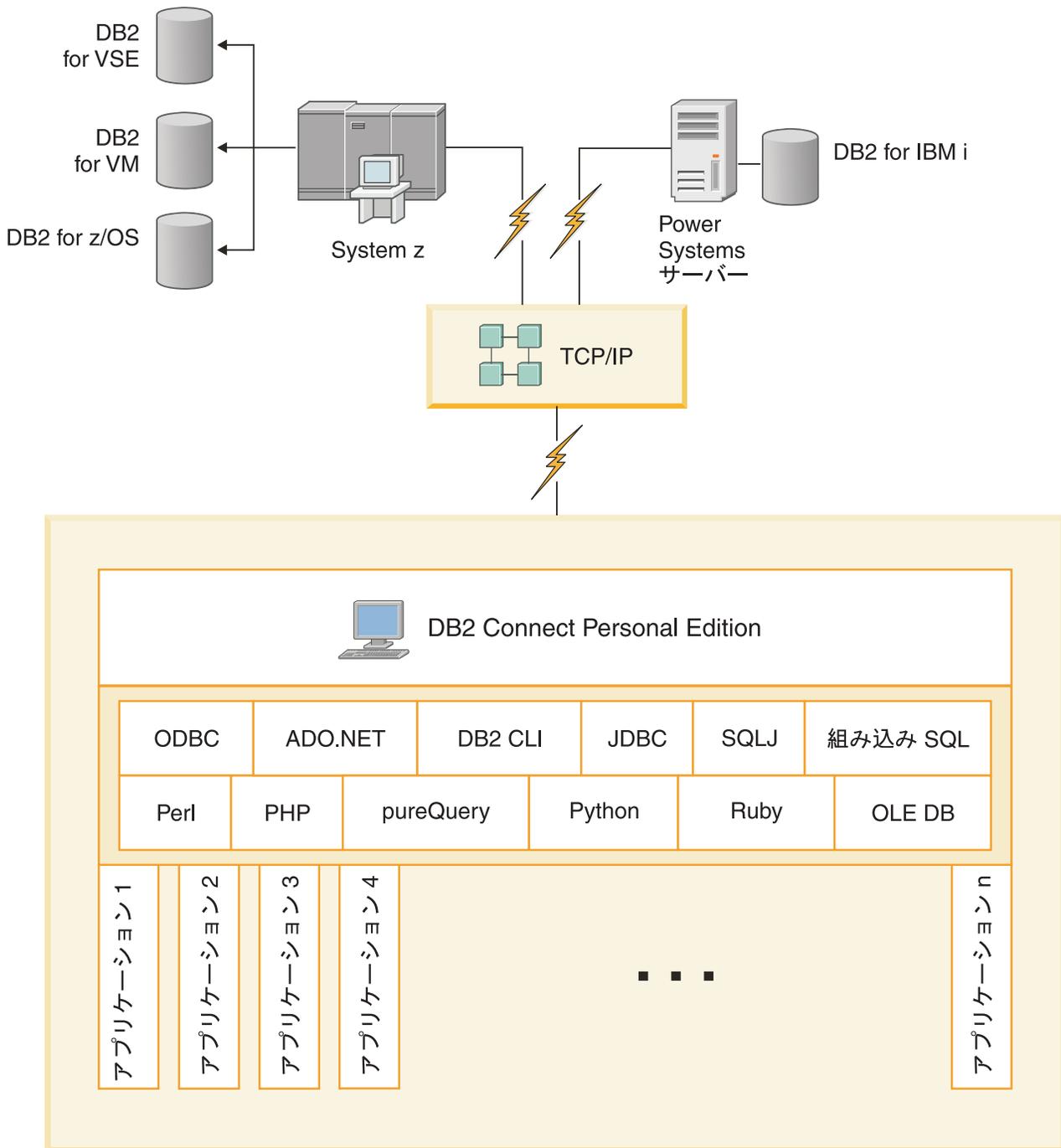


図4. DB2 Connect と、IBM メインフレーム・データベース・サーバーとの直接接続

接続サーバーとしての DB2 Connect サーバー製品

DB2 Connect サーバーを使用すれば、複数のクライアントが IBM メインフレームのデータに接続できます。また、企業データへのアクセスの確立や保守に要する労力を大幅に減らすことができます。19 ページの図5 は、DB2 クライアントから IBM メインフレーム・データベース・サーバーに対し、DB2 Connect Enterprise

Editionなどのような DB2 Connect サーバー製品を介して、間接接続を行う環境で IBM が提供するソリューションを示しています。

注: 間接接続は、Linux、UNIX、または Windows 上で実行される DB2 クライアントまたは JCC クライアントでのみサポートされます。これ以外のクライアントを使用して DB2 Connect サーバー製品を介して IBM メインフレーム・データベース・サーバーへの接続を試みると、SQL1334 エラーが発生します。

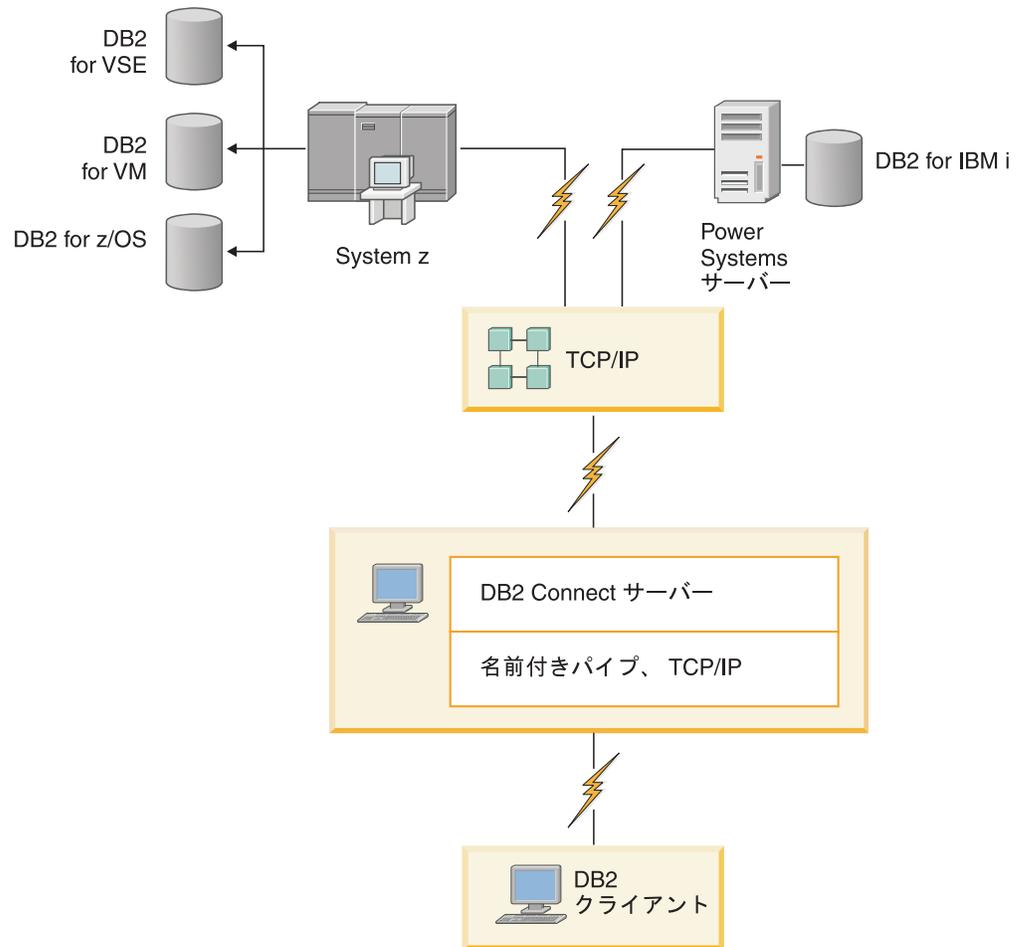


図 5. DB2 Connect Enterprise Edition

DB2 Connect サーバーに対する TCP/IP 接続が失われた場合、クライアントはその接続を自動的に再確立しようとします。クライアントはまず、元のサーバーに対する接続を再確立しようとします。その接続を再確立できない場合、クライアントは代替 DB2 Connect サーバーにフェイルオーバーします。(代替サーバーは、サーバー・インスタンスで指定します。その代替サーバーのロケーションは、接続時にクライアントに戻されます。) 代替サーバーへの接続を再確立できない場合、クライアントは、元のサーバーに対する接続を再確立しようとします。このようにクライアントは、接続を再確立するか再試行回数がタイムアウトになるまで、元のサーバーと代替サーバーの間を切り替えながら、接続を再確立しようとします。

DB2 Connect と Web アプリケーション

Web ブラウザーは急速な勢いで、オンライン・カタログからイントラネット・アプリケーションに至るあらゆるものの標準インターフェースになりつつあります。単純な Web アプリケーションであれば、Web サーバーだけで十分な場合もあります。それに対し、データベース・アクセスおよびトランザクション処理を必要とするような、大量データを処理するアプリケーションについては、DB2 Connect を使って Web 上で膨大な数のトランザクションを同時に処理するソリューションが IBM によって提供されています。

従来の CGI プログラミングの利点と限界

WWW 上の e-business アプリケーションは通常、ユーザーがバックエンドのデータベースを照会できるように CGI (Common Gateway Interface) を使っています。多くの企業では、内部でも Web アプリケーションを利用しており、この場合でも通常はバックグラウンドでデータベースが稼働しています。

ユーザーが Web ページ上のフォームに記入すると、それらのフォームは CGI 経由で Web サーバー上のアプリケーションまたはスクリプトに送られます。次に、そのスクリプトは提供されているデータベース API を使って、SQL 照会をホスト・データベースへ送信します。続いて、その同じスクリプトは照会の結果を表示する Web (HTML) ページを作成し、ユーザーの Web ブラウザーが表示できるようにそれを送り返します。例えば、ユーザーが特定の商品の在庫状況やサービスの利用状況、現在の価格などを照会できるオンライン・カタログはその一例です。

CGI アプリケーションは設計をシンプルにすることができ、保守するのも容易です。CGI はオペレーティング・システムと言語のいずれにも依存しない規格であるため、ほとんどすべてのコンピューティング・プラットフォームで利用できます。CGI プログラムは C++ で記述することも、Perl や PHP などのスクリプト言語で記述することもできます。

このように、CGI は Web ベースのアプリケーションにおける理想的なソリューションのように見えますが、実は重大な欠点があります。まず、CGI のプログラミング環境は他の API の場合ほど洗練されていません。加えて、大規模な e-commerce 操作では、スケーラビリティが問題となる場合があります。CGI アプリケーションを 1 つ起動するたびに、Web サーバーでは新しいプロセスが 1 つ作成されます。各プロセスはデータベースへの接続を独自のものとして確立し、各自の照会を送信する必要があります。大量のトランザクションの処理を行う環境では、このような制限によってパフォーマンスに重大な問題が生じることがあります。

DB2 Connect を Web サーバーと組み合わせて使えば、大量のデータを扱う堅固な e-commerce アプリケーションを作成できます。DB2 Connect には、Web ベースのアプリケーションのパフォーマンスを改善するいくつかのソリューションが用意されています。さらに、DB2 Connect のユーザーはストアド・プロシージャを利用することで、データベースに送られる照会の数を減らすことができます。

接続プールを利用すれば、データベースとの間の接続および切断の頻度を少なくすることができます。

Web サーバーのモジュールまたはプラグインとしての PHP の使用

PHP を CGI プログラミングで使用することは可能ですが、PHP は通常、Web サーバーのモジュールまたはプラグインとして使用されます。Apache のようなマルチプロセス Web サーバーでは、PHP 用の IBM DB2 ドライバーを使用することによりスケラビリティの問題を緩和することができます。マルチプロセス Web サーバーでは、Web サーバーの要求のサービスのためにプロセスのプールが再利用されます。Web 要求が発生するたびにデータベース接続を確立せずにはむように、持続的な接続を作成することができます。この環境では、単一 PHP スクリプトの有効範囲を超えて持続的な接続を作成することができます。後続の Web 要求で同一の接続が必要になる場合に接続が再利用されます。

DB2 Connect と IBM WebSphere

IBM WebSphere® は、PHP などのような従来のスクリプト・ツールで可能な範囲を超えた、より完全な e-business ソリューションを提供します。WebSphere Application Servers は PHP のスクリプト機能を実行するだけでなく、Web を介した複雑でハイエンドなサービスを、サーブレット、Active Server Pages、エンタープライズ JavaBeans™ の使用により可能にします。さらに、Java™、TCP/IP、HTTP、HTTPS、HTML、DHTML、XML、MIME、SMTP、IIOP、および X.509 などの Web ベースのテクノロジーに対するサポートも組み込まれています。WebSphere を使用することで以下のことが可能になります。

- 業界標準を利用することで、開発のスピードを上げて相互運用性を最大にする
- サード・パーティー製ツールのテクノロジーやアプリケーション・フレームワークをプラグインする (組み込む)
- Web サイトのコンテンツの性能と使用状況を分析する
- より多くのユーザーに対応しつつスループットを維持するために、サイトの規模を簡単に調整できる
- 多くの主要なオペレーティング環境 (AIX®、HP-UX、Linux、Novell NetWare、z/OS、IBM i、Solaris オペレーティング・システム、Microsoft Windows)でのデプロイ
- Apache、IBM、Netscape、および Microsoft など、既存の Web サーバーを使用できる

WebSphere は単一の製品ではなく、3 つの異なるマーケットを対象にしたファミリー製品です。WebSphere によるソリューションの中核を成しているのは、WebSphere Application Server です。

WebSphere Application Server が提供する環境は、3 種類のオブジェクトを対象にしています。最初のものは Java Server Pages (JSP) で、Active Server Pages (ASP) に類似しています。2 番目は Java サーブレットで構成されるコンポーネントであり、3 番目は Enterprise JavaBeans です。Enterprise JavaBeans は、大規模で堅固なエンタープライズ・クラスのアプリケーションを配置する際の標準になりつつあります。

WebSphere アプリケーションは、Web サーバーや DB2 と同じプラットフォーム上に置くことができます。DB2 for z/OS、DB2 Server for VM and VSE、および DB2 for IBM iの場合、WebSphere は DB2 Connect サーバー製品と同じプラットフォーム上に配置されます。

WebSphere には Rational® Application Developer (RAD) の他にも、複数のソリューションが用意されています。詳しくは、<http://www.ibm.com/software/webservers/appserv/was/> を参照してください。

Java アプリケーション・サーバーとしての DB2 Connect

代わりに Java を使用することで、スクリプト言語に関連した多くの欠点を回避できます。IBM は、Web トランザクションのどの段階でも Java を利用できるようにするアプレットとアプリケーションの両方を提供しています。IBM が提供するソリューションでは、いくつかの技術を組み合わせて利用できるようになっています。つまり、Perl DBI や Microsoft Active Server Pages のようなスクリプトによるソリューションを DB2 と組み合わせて使ったり、Java アプリケーション・サーバー (IBM WebSphere など) で実現されるより堅固なシステムへ移行できるということです。

Java プログラマーが使えるアプリケーション・プログラミング・インターフェース (API) は 2 つあります。その 1 つである JDBC は、Java を使ってデータ認識の Java アプレットや Java アプリケーション、さらに Java サブアプレット、Java Server Pages (JSP)、Enterprise Java Beans (EJB) を開発するためにサポートされています。JDBC はコール・レベルまたはメソッド呼び出しの API です。もう一方の Java API は、SQLJ です。SQLJ では、SQL を Java プログラム内にインラインで指定することが可能です。DB2 はこの両方の API を、Web トランザクションのクライアント側かサーバー側で使用できます。

クライアント側では、アプレット、データ認識のアプレット、アプリケーションなどがサポートされています。データベース側では、Java Enablement がデータベース・オブジェクト (ユーザー定義関数やストアド・プロシージャなど) で構成されています。

DB2 for z/OS、DB2 Server for VM and VSE、および DB2 for IBM iでは、Java アプリケーションを配置する方法が 2 種類あります。TCP/IP で DB2 Connect Personal Edition により作成される直接接続を使用するか、IBM メインフレーム・データ・サーバーへの接続が可能な DB2 Connect サーバー製品を介して接続することも可能です。

どちらの場合も、Web 上のユーザーはデータベースへアクセスするのに、標準的な Web ブラウザー以外の特別なソフトウェアを必要としません。また、インストールする必要があるのは、DB2 Connect サーバー製品といずれかの業界標準の Web サーバーだけです。Web サーバーと DB2 Connect が同じ物理マシン上にない場合は、IBM Data Server Client を Web サーバー上にインストールしなければなりません。

DB2 for z/OS の場合、その主要コンポーネントは中間層サーバーで稼働する DB2 Connect サーバー製品です。このコンポーネントが、DB2 for z/OS、DB2 Server for VM and VSE、および DB2 for i サーバーへの接続に加えて、JDBC サーバー

を有効にする機能を提供しています。この場合でも、クライアントの Web ブラウザーに特別なソフトウェアは必要ありません。

IBM では、Java アプリケーションとアプレットを開発するための多岐にわたるサポートとツールを用意しています。データベース・アプリケーションの開発のために、DB2 Database Enterprise Developer Edition では、テスト用として、DB2 製品および DB2 Connect 以外にも Rational Web Developer、IBM Data Studio、DB2 WebSphere Application Server などを提供しています。サード・パーティー製のツール、例えば NetBeans、Borland JBuilder や Symantec Visual Cafe なども、IBM のデータベース・ソリューションで利用できます。

Web サーバー上の DB2 Connect

IBM では、すべての DB2 Connect 製品で HTTP (Web) サーバーを提供しています。DB2 Connect Enterprise Edition などのような DB2 Connect サーバー製品では、Apache または Lotus Domino® Go といった Web サーバーを追加設定の必要なくご利用いただけるだけでなく、Microsoft Internet Information Server または Netscape Enterprise Server などの他の Web サーバーと組み合わせて利用することも可能です。

System z、IBM Power Systems、VM、および VSE の各システム上で稼働している DB2 ファミリーのデータベースを利用している場合は、Web サーバー側では DB2 Connect サーバー製品が必要になります。DB2 Connect サーバー製品には、Web サーバーがこれらの IBM メインフレーム・プラットフォームへアクセスできるようにするライブラリーと通信インターフェースが用意されています。Web サーバーと、System z、IBM Power Systems、VM、または VSE 上で稼働しているデータベースとの間の通信には、TCP/IP を使用できます。

注：IBM の Web ソリューションでは、複数のデータベースを同一の Common Gateway Interface (CGI) スクリプト (例えば PHP) で、あるいは特定の CGI スクリプトの同一トランザクション内で扱うことができます。

ストアード・プロシージャ

クライアント/サーバーの場合と同様、Web アプリケーションについて考慮すべき重要な点として、HTTP サーバーとバックエンド・データベースの間で発生するトラフィックを最小限に抑えることが挙げられます。これは、大多数の e-business アプリケーションで中心的な役割を果たす、大量データのトランザクション処理において特に重要な点です。

推奨されるアプローチは、CGI アプリケーションのプログラミングを、ストアード・プロシージャにカプセル化されたプログラミングおよびビジネス・ロジックと組み合わせる手法です。DB2 Database for Linux, UNIX, and Windows、および DB2 for z/OS、DB2 for IBM i、および DB2 for VSE はいずれも、ストアード・プロシージャを呼び出すときに同じパラメーター規則を使います。

通常の Web インターフェース・スクリプトの場合と同様、Web ブラウザーはフォームを Web サーバーに送信し、そこで Web インターフェース・スクリプトが実行されます。ただし、個々の SQL ステートメントが DB2 データベースに送られる代わりに、ストアード・プロシージャを実行するようにとの要求が送られます。こ

のストアード・プロシージャは、何もしなければ別々に実行されてしまういくつかの SQL ステートメントをカプセル化します。それで、ストアード・プロシージャを利用することによって、Web インターフェース・スクリプトとバックエンド・データベースとの間でやり取りされるメッセージの数を減らすことができます。

ストアード・プロシージャの主な利点は、HTTP サーバーとバックエンドの DB2 データベースの間におけるネットワーク・トラフィックが減少することです。

DB2 Connect とアプリケーション・サーバー

クライアント/サーバー型のアプリケーションが登場したとき、アプリケーション設計者は Windows などのプラットフォーム上でアプリケーションにグラフィカル・ユーザー・インターフェースを持たせることにより、ユーザビリティ（使いやすさ）を向上させてトレーニング・コストを削減することが可能になりました。同時に、多様なオペレーティング・システムとハードウェア・プラットフォームにある堅固なデータベース・サーバーに、データベース管理機能を柔軟な方法で送ることも可能になりました。

アプリケーション・ロジックがクライアント・ワークステーションに配布されるこのクライアント/サーバー・モデルのことを、通常は 2 層クライアント/サーバーと呼びます。この 2 層モデルでは、アプリケーションはクライアント層に配置され、データベース・サーバーはサーバー層、つまりバックエンド層にインプリメントされます。DB2 Connect は、データベース・サーバーが DB2 for z/OS、DB2 for IBM i、または DB2 Server for VM and VSE である 2 層クライアント/サーバー・アプリケーションを完全にサポートします。

クライアント/サーバー・アプリケーションのサイズが拡大するにつれて、2 層クライアント/サーバー・モデルには重大な限界があることが明らかになってきました。大量のビジネス・ロジックを数百、あるいは数千のクライアント・ワークステーションに配布することは、変更管理の作業を複雑で費用のかかるものにしてしまいました。また、ビジネス・ルールに何か変更を加えると、アプリケーションのクライアント部分を置き換えなければなりません。アプリケーションのそのような置き換えは多くの場合、ビジネス・ルールが一貫して適用されるようにするために、企業内のすべてのクライアント・ワークステーションで一斉に行うことが必要です。

システムの規模に関して明白になってきた 2 層クライアント/サーバー・モデルの別の欠点は、そのようなアプリケーションが消費するリソースの量です。数百、あるいは数千のファット・クライアント（太ったクライアント。2 層モデルにおけるクライアントはしばしばこのように呼ばれる）を配置することによって、各クライアント・ワークステーションにより高い処理能力と大きなディスク容量が要求されるようになりました。さらに、それぞれのクライアントが専用のデータベース接続を必要とし、そのような接続を保持することに関連したリソースも必要とされるため、データベース・サーバーの要件も非常に高くなっています。ビジネス・ロジックを配布することに対する 2 層クライアント/サーバーの依存性は、ストアード・プロシージャを広範囲にわたって使うことである程度軽減できますが、その他の欠点については、モデルを変更する以外に簡単に解決する方法はありません。

アプリケーション・サーバー・ソリューション

2 層クライアント/サーバー・アプリケーションにかかるコストとその複雑

さが増大するにつれ、大型アプリケーションの大部分は、多層からなるクライアント/サーバーの方向に発展しました。多層モデルでは、データベース層の役割は変わっていませんが、クライアント層には 1 つまたは複数の中間層が追加されます。追加される層は普通は 1 つなので、このモデルは 3 層 という名前になっています。

3 層モデルでは、クライアントの処理はユーザーとの対話だけにとどまり、ビジネス・ロジックは何も組み込まれません。また、中間層は 1 つまたは複数のアプリケーション・サーバーで構成されます。アプリケーション・サーバーを利用する目的は、ビジネス・プロセスとビジネス・ルールの背後にあるロジックを、堅固かつ費用効率の高い方法でインプリメントすることにあります。2 層モデルの場合にそうであったように、ビジネス・ルールのインプリメンテーションには多くの場合、パフォーマンスを向上させるためにストアド・プロシージャが補足的に使われます。

クライアント・ワークステーションは大量のアプリケーション・ロジックをインプリメントせずに、ユーザーとの対話だけを処理すればよくなるため、クライアント層のリソース要件は大幅に軽減されます。そのような理由から、3 層モデルにおけるクライアント層はよくシンクライアント (やせたクライアント) と呼ばれます。さらに、中心となるアプリケーション・サーバーがすべてのクライアントからの要求を処理するため、そのアプリケーション・サーバーはすべてのクライアント間のデータベース接続などのリソースを共用することが可能です。その結果、データベース・サーバーがアプリケーション・ユーザーごとに専用の接続を保持する必要がなくなります。

現在、業界には 3 層アプリケーション・サーバーの応用例がいくつも存在しています。ほとんどの ERP (エンタープライズ・リソース・プランニング) ベンダーは 3 層モデルを使って自社のアプリケーションをインプリメントしています。これには、SAP R/3 や PeopleSoft V7 といったアプリケーションが含まれます。それ以外の例としては、Siebel や Vantive などの代表的なエンタープライズ・リレーションシップ・マネージメント・ベンダーが含まれています。

アプリケーション・サーバーと DB2 Connect

DB2 Connect サーバー製品は複数階層アプリケーションをデプロイするための包括的サポートを提供します。DB2 Connect によるサポートは、アプリケーション・ロジックを開発するのに使用できる各種の API (ODBC、ADO、DB2 CLI、組み込み SQL、JDBC、SQLJ、Perl、PHP、OLE DB など) だけでなく、DB2 ファミリーのデータベース・サーバーと対話するための完全な通信基盤をも含んでいます。

DB2 Connect はさらに、データベース層が DB2 ファミリーの複数のデータベース・サーバーで構成されているようなシステムもサポートします。これにより、アプリケーション・サーバーは複数のデータベース・サーバー上に置かれているデータを 1 回のトランザクションで更新するトランザクションをインプリメントできます。

DB2 Connect が提供する 2 フェーズ・コミット・プロトコルのサポートは、そのような分散トランザクションの整合性を保証しています。例えば、アプリケーションは同じトランザクションで DB2 for z/OS データベースと DB2 Database for Linux, UNIX, and Windows のデータを更新することが可能です。また、分散要求サポートがインストールされて有効になっている場

合は、アプリケーションは同一トランザクション内で Oracle データベースの読み取りと DB2 ファミリーのデータベースの更新を行うことができます。

下記の図に示すように、DB2 Connect Enterprise Edition などの DB2 Connect サーバー製品によって、アプリケーション・サーバーとバックエンドのデータベース・サーバーとの間の接続機構および API が提供されています。

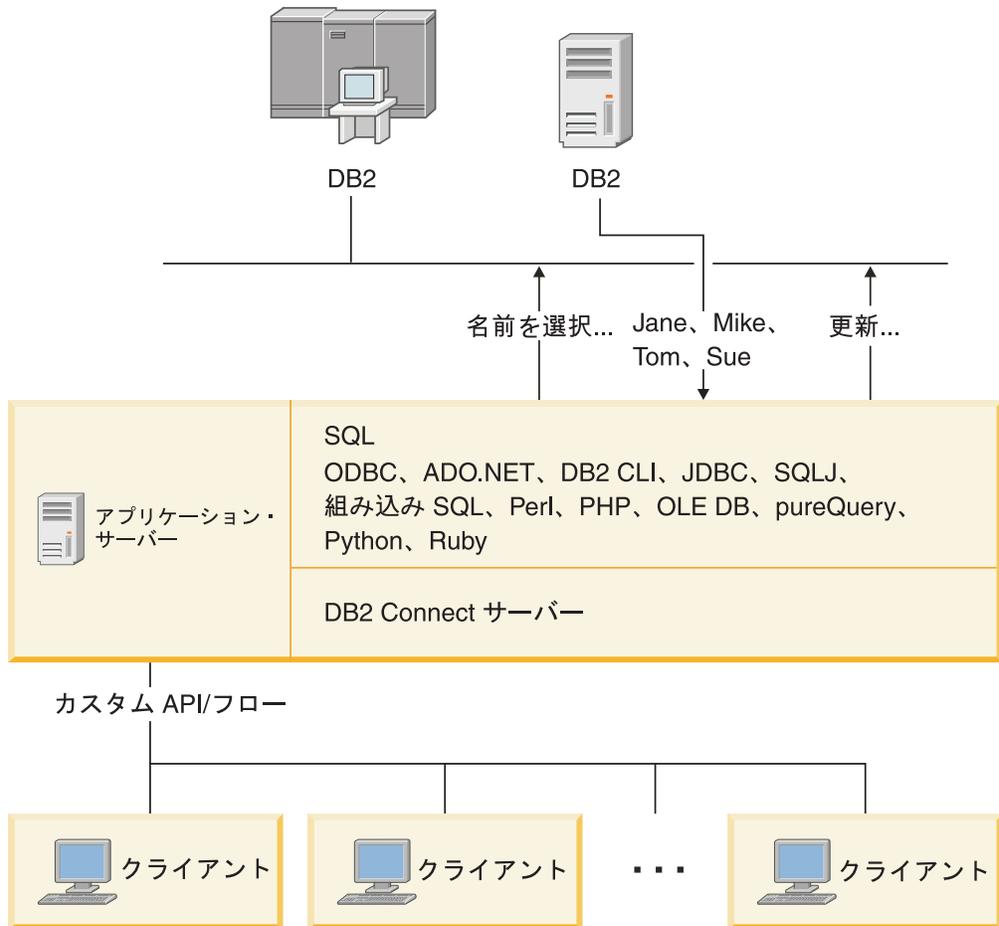


図 6. DB2 Connect によるアプリケーション・サーバーのサポート

接続プールなどの DB2 Connect の先進的なフィーチャーを利用すれば、アプリケーションのリソース要件は大幅に軽減され、アプリケーション・サーバーのインプリメンテーションも単純化できます。

DB2 Connect とアプリケーション・サーバーの構成

DB2 Connect サーバー製品は、アプリケーション・サーバーとともに使用する場合があります。DB2 Connect Personal Edition はアプリケーション・サーバーをサポートしておらず、それに必要なライセンスも含まれていません。さらに、アプリケーション・サーバーをインプリメントする予定があれば、ご使用の DB2 Connect の使用許諾条件をよく読み、入手する必要のあるライセンスの数を把握してください。

アプリケーション・サーバー環境では、DB2 Connect を配置する方法が 2 つあります。DB2 Connect サーバー製品は、次のいずれかにインストールできます。

- アプリケーション・サーバーのマシン
- 独立している通信サーバーのマシン

ほとんどのケースでは、DB2 Connect をアプリケーション・サーバーと同じサーバー上にインストールすることをお勧めします。DB2 Connect をアプリケーション・サーバー上にインストールすることで、アプリケーション・サーバーでインプリメントしようとしているフェールオーバーおよびロード・バランシングの仕組みに DB2 Connect を関与させることができます。このようにセットアップすると、DB2 Connect を別のサーバー上にインストールした場合に必要な余分のネットワーク・ホップがないため、パフォーマンスの向上を期待できます。さらに、別のサーバーをもう 1 つインストールして保守することも不要であるため、管理作業も単純化されます。

DB2 Connect サーバー製品が、アプリケーション・サーバーが稼働しているオペレーティング・システムやハードウェア・プラットフォームに対応していない場合は、DB2 Connect を別のサーバーにインストールすることをお勧めします。

DB2 Connect とトランザクション処理モニター

アプリケーション・サーバーを利用すれば、多数のユーザーが最小限のシステム・リソースでアプリケーションを実行できます。アプリケーション・サーバーを拡張して、調整されたトランザクションを、そのアプリケーション・サーバーが実行するアプリケーションから呼び出せるようにすることができます。このトランザクション調整機能は一般に、トランザクション処理 (TP) モニターとして知られています。TP モニターはアプリケーション・サーバーと連携して機能します。

トランザクションは、組織の日常業務を処理するときに生じる定型的なイベント (通常はサービス要求) と見なすことができます。トランザクションの規則正しい処理が、TP モニターで想定されている作業のタイプとなっています。

トランザクション処理

どの組織にも、その組織の運営方法を表現したルール (法則) と手順が存在します。これらのルールを具体化するユーザー・アプリケーションのことを、ビジネス・ロジックと呼ぶことができます。また、これらのビジネス・アプリケーションが実行するトランザクションのことを、しばしばトランザクション処理、あるいはオンライン・トランザクション処理 (OLTP) と呼びます。

商用 OLTP の主要な特性は以下のとおりです。

多数のユーザー

非常に多くの人々がビジネスの現在の状態に影響を与えるので、トランザクション処理は組織の大多数の人によって利用されるのが普通です。

反復性 コンピューターとの対話のほとんどは、何度も繰り返し実行される同一のプロセスであることが少なくありません。例えば、注文の入力や支払いの処理などは毎日、何回も使われます。

短時間の対話

組織内の人間がトランザクション処理システムと行う対話のほとんどは短時間で済みます。

共有データ

データは組織の状態を表すものであるため、データには 1 つのコピーしかありません。

データ整合性

データは組織の現在の状態を表していなければならず、内部的な矛盾があってはなりません。例えば、どの注文もカスタマー・レコードに関連付けられている必要があります。

低コスト/トランザクション

トランザクション処理はビジネスを行うときの直接コストを表しているため、システムのコストは最小でなければなりません。DB2 Connect を使用すれば、Linux、UNIX、Windows 上で稼働しているアプリケーション・サーバーの制御下にあるアプリケーションが、リモート LAN、IBM メインフレーム・データベース・サーバーに対してトランザクションを実行し、それらのトランザクションを TP モニターに調整させることが可能になります。

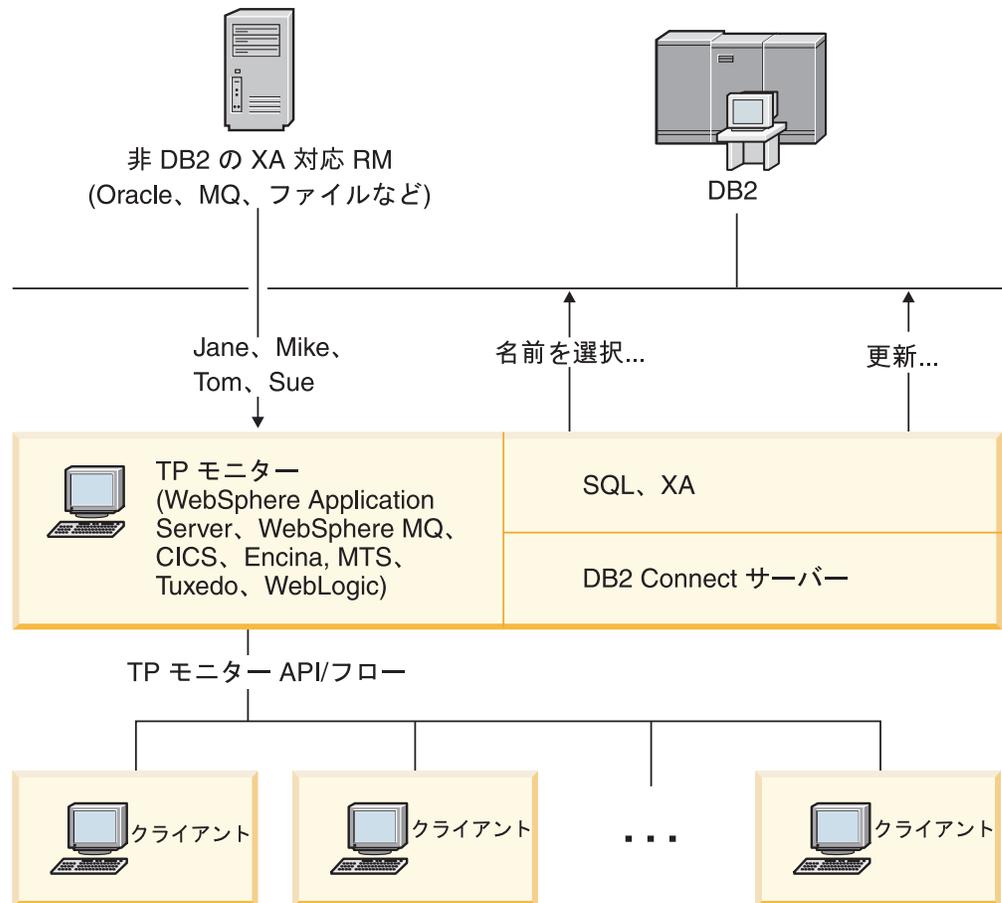


図 7. DB2 Connect による TP モニターのサポート

図 7 に示すように、DB2 Connect Enterprise Edition などの DB2 Connect サーバー製品によって、アプリケーション・サーバーとバックエンドのデータベース・サーバーとの間の接続機構および API が提供されています。

トランザクション処理モニターの例

現在、市販されている代表的な TP モニターには以下のものがあります。

- IBM WebSphere Application Server
- IBM WebSphere MQ
- IBM TxSeries CICS®
- BEA Tuxedo
- BEA WebLogic
- Microsoft Transaction Server (MTS)

リモートの IBM Power Systems、System z、LAN のデータベース・サーバーは、これらの TP モニターで調整されたトランザクション内で使用できます。

X/Open 分散トランザクション処理 (DTP) モデル

単一のトランザクションで複数のリソースを更新するのに、ビジネス・ロジックを実行するアプリケーションが必要になることがあります。例えば、ある口座から別

の口座への送金を実現する銀行業務アプリケーションは、一方のデータベース (送金元口座) からの引き落とし処理と、もう一方のデータベース (送金先口座) への入金処理を必要とするかもしれません。

これら 2 つのデータベースが別々のベンダーのものである可能性もあります。例えば、一方のデータベースが DB2 for z/OS で、もう一方が Oracle データベースになっている場合があります。このような場合、それぞれのデータベース・ベンダー独自のトランザクション・インターフェースを TP モニターごとにインプリメントするのではなく、TP モニターと、アプリケーションがアクセスするリソースとの間に共通のトランザクション・インターフェースが定義されています。このインターフェースは XA インターフェース として知られているものです。XA インターフェースを使用する TP モニターのことを XA 準拠トランザクション・マネージャー (TM) と呼びます。また、XA インターフェースをインプリメントする更新可能なリソースのことを XA 準拠リソース・マネージャー (RM) と呼びます。

上記の TP モニターはすべて XA 準拠 TM です。リモート・ホスト、IBM Power Systems、および DB2 の LAN ベースのデータベースは、DB2 Connect 経由でアクセスする場合は XA 準拠 RM になります。そのため、XA 準拠 TM を有する TP モニターであれば、トランザクションを実行するビジネス・アプリケーション内にある、ホスト、IBM Power Systems、および LAN ベースの DB2 のデータベース・サーバーを使用できます。

第 2 部 DB2 Connect リファレンス

第 4 章 データベース・ディレクトリーの更新

DB2 Connect では、データベースの接続情報を管理するために、以下のディレクトリーを使用します。

- システム・データベース・ディレクトリー。DB2 Connect がアクセスするすべてのデータベースの名前、ノード、および認証の情報があります。
- ノード・ディレクトリー。このディレクトリーには、DB2 Connect がアクセスするすべての IBM メインフレーム・データベース・サーバーのネットワーク・アドレスや通信プロトコル情報が含まれます。
- データベース接続サービス (DCS) ディレクトリー。このディレクトリーには、IBM メインフレーム・データベース・サーバー・データベースに固有の情報が含まれます。

注:

1. これらのディレクトリーを更新する前に、IBM メインフレーム・データベース・サーバー上とワークステーション上で通信を構成しておく必要があります。
2. データベース・ディレクトリーは構成アシスタント (CA) を使って更新することができます。

データベース・ディレクトリーを更新するには、次のようにします。

1. ディレクトリー・カスタマイズ・ワークシートを使ってデータベース・ディレクトリー情報を収集します。
2. コントロール・センターの『リモート・データベース・サーバー・マシンの情報を使用したディレクトリーの更新』のトピックを参照してください。

システム・データベース・ディレクトリーの値

システム・データベース・ディレクトリーは、データベース・マネージャーの各インスタンスに対して存在します。このディレクトリーには、このインスタンスに対してカタログされているデータベースごとに 1 つの項目が含まれます。DB2 Connect 製品では、システム・データベース・ディレクトリーには、各データベースの名前、別名、ノード名、および認証タイプに関する情報が含まれます。

システム・データベース・ディレクトリーには、次の情報を指定することができます。

データベース名

「DCS ディレクトリー・パラメーター (DCS Directory Parameter)」表に記入したのと同じ値。

データベース別名

IBM メインフレーム・データベース・サーバーの別名。この名前は、データベースにアクセスするどのアプリケーション・プログラムでも使用されます。デフォルト設定では、「データベース名」に指定した値が使用されます。

形式は、1 桁から 8 桁までの 1 バイト英数字です。(番号記号 (#)、アットマーク (@)、ドル記号 (\$)、および下線 (_) が含まれます。) 下線または数字で始めることはできません。

ノード名

「ノード・ディレクトリー・パラメーター (Node Directory Parameter)」表に記入したのと同じ値。

認証 DB2 Connect サーバーが元になっている接続に関して、ユーザー名およびパスワードの検証をどこで行うかを指定します。有効なオプションは、SERVER、SERVER_ENCRYPT、CLIENT、KERBEROS、SERVER_ENCRYPT_AES、および DATA_ENCRYPT です。システム・データベース・ディレクトリーでの GSSPLUGIN 認証タイプのサポートはありません。

ノード・ディレクトリーの値

ノード・ディレクトリーには、次の情報を指定することができます。

ノード名

リモート・データベースがある IBM メインフレーム・データベース・サーバー・システムのニックネーム。この名前は、ユーザー定義です。同じノード名を「ノード・ディレクトリー・パラメーター」表および「システム・データベース・ディレクトリー・パラメーター」表の両方に記入します。

形式は、1 桁から 8 桁までの 1 バイト英数字です。(番号記号 (#)、アットマーク (@)、ドル記号 (\$)、および下線 (_) が含まれます。) 下線または数字で始めることはできません。

プロトコル

TCP/IP でなければなりません。

セキュリティー・タイプ

実行されるセキュリティー・チェックのタイプ。TCP/IP ノードでは、SECURITY SOCKS は、ノードで SOCKS を使用できる状態にするように指定するオプションです。この場合、SOCKS を使用可能にするには、SOCKS_NS および SOCKS_SERVER 環境変数は必須であり、設定しなければなりません。

TCP/IP リモート・ホスト名または IP アドレス

TCP/IP ノードを定義している場合、リモート TCP/IP ホスト名またはリモート TCP/IP アドレス。ホスト名が指定される場合、DB2 Connect ワークステーションで、ドメイン・ネーム・サーバー (DNS) の検索を介して、またはローカル TCP/IP ホスト・ファイル内の項目によって解決されなければなりません。

DB2 for z/OS リモート・ホストの場合、ホスト名は、分散データ機能 (DDF) の起動時に DSNL004I メッセージ (DOMAIN=hostname) に示されます。-DISplay DDF コマンドも使用できます。

z/OS データ共用グループにアクセスする場合は、ドメイン名を DB2 グループの動的 VIPA アドレスにマップする必要があります。このアドレスは、最小ロードの DB2 メンバーに経路を定めます。特定のメンバーにアクセスするには、その DB2 メンバーの動的 VIPA アドレスを使用して、

SYSPLX ルーティングをオフにしてください。メンバー固有のドメイン名は、各メンバーの DSNL004I メッセージに表示されます。

TCP/IP サービス名またはポート番号

TCP/IP ノードを定義している場合、リモート TCP/IP サービス名またはポート番号。これは、リモート・ホストで TCP/IP に定義しなければなりません。ポート番号 446 を DRDA 用のデフォルト・ポート番号として登録しておく必要があります。

DB2 for z/OS リモート・ホストの場合、ポート番号はブートストラップ・データ・セット (BSDS) で PORT として定義され、分散データ機能 (DDF) の起動時に出される DSNL004I メッセージ (TCPPOINT=portnumber) でも提供されます。-DISplay DDF コマンドも使用できます。

z/OS データ共有グループにアクセスする場合は、ドメイン名を DB2 グループの動的 VIPA アドレスにマップする必要があります。このアドレスは、最小ロードの DB2 メンバーに経路を定めます。特定のメンバーにアクセスするには、その DB2 メンバーの動的 VIPA アドレスを使用して、SYSPLX ルーティングをオフにしてください。メンバー固有のドメイン名は、各メンバーの DSNL004I メッセージに表示されます。

注: TCP/IP 接続での 2 フェーズ・コミット再同期操作で用いられる 2 番目のポートは、サーバーにより割り当てられます。例えば、DB2 for z/OS ブートストラップ・データ・セットの場合、DB2 for z/OS へのインバウンド接続の再同期でのみ使用されるポート番号 (RESPORT) が割り当てられます。サービス名を定義する必要はありません。

DCS ディレクトリーの値

DCS ディレクトリー内では、以下の情報を指定することができます。

データベース名

IBM メインフレーム・データベース・サーバーのユーザー定義のニックネーム。同じデータベース名を「DCS ディレクトリー・パラメーター」表および「システム・データベース・ディレクトリー・パラメーター」表の両方に記入します。

形式は、1 桁から 8 桁までの 1 バイト英数字です。(番号記号 (#)、アットマーク (@)、ドル記号 (\$)、および下線 (_) が含まれます。) 下線または数字で始めることはできません。

ターゲット・データベース名

IBM メインフレーム・データベース・サーバー・システム上のデータベース。以下のとおりです。

System z

DB2 for z/OS サブシステム。その LOCATION NAME または z/OS サーバーで定義されているいずれかの別名 LOCATION 名によって識別される。

LOCATION NAME は、TSO にログインした後で、利用可能ないずれかの照会ツールを使って以下に示す SQL 照会を発行することで判別できます。

```
select current server from sysibm.sysdummy1
```

さらに、複数の LOCATION NAME はブートストラップ・データ・セット (BSDS) でも定義されており、DSNL004I メッセージ (LOCATION=location) に表示されます。これは、分散データ機能 (DDF) の起動時に書き込まれます。-DISplay DDF コマンドも使用できます。

z/OS データ共有グループにアクセスする場合は、ドメイン名を DB2 グループの動的 VIPA アドレスにマップする必要があります。このアドレスは、最小ロードの DB2 メンバーに経路を定めません。特定のメンバーにアクセスするには、その DB2 メンバーの動的 VIPA アドレスを使用して、SYSPLEX ルーティングをオフにしてください。メンバー固有のドメイン名は、各メンバーの DSNL004I メッセージに表示されます。

VSE または VM

データベース名 (DBNAME)

IBM Power Systems

リレーショナル・データベース名 (RDBNAME)

その他 Windows、Linux、および UNIX オペレーティング・システムの場合は、データベース・ディレクトリーにあるデータベース別名。

パラメーター・ストリング

デフォルトを変更する場合は、次のパラメーターの一部または全部を次の順序で指定します。

マッピング・ファイル

デフォルトの SQLCODE マッピングを上書きする SQLCODE マッピング・ファイルの名前。SQLCODE マッピングをオフにするときは、**NOMAP** を指定します。

注: 照会要求を処理する際、DRDA サーバーは、結果セットを表す、行のセットの形式でデータを戻します。各行とともに SQLCA も戻され、通常はゼロまたは正の sqlcode を含んでいます (+12 または +802 など)。DB2 Connect サーバーでカスタマイズされたマッピング・ファイルを使用すると、正の sqlcode がカスタマイズされたマッピング・ファイルに含まれ、カスタマイズされたマッピングを持つ場合 (例えば、異なる sqlcode にマップされる場合、あるいはカスタマイズされたトークン・マッピングを持つ場合)、マップは行われません。

以下は、強調すべき重要な点です。

1. 正の sqlcode は警告を表すのに対し、負の sqlcode はエラー条件を示します。負の sqlcode はすべて、どのマッピング・ファイルが使用されているかに関係なく、いつでもどのような環境においてもマップされます。正の sqlcode は、カスタマイズされたマッピング・ファイルに含まれ、変更なしでそれ自体にマップする場合はすべて、同様に常にマップされます。さらに、DB2 Connect

サーバーでカスタマイズされたマッピング・ファイルに含まれていない正の sqlcode も、常にマップされます。

2. デフォルトのマッピング・ファイルを使用する場合、またはホスト・データベースに直接接続する場合、sqlcode のマッピングは常にすべての sqlcode に関して実行されます。

,D これは 2 番目の定位置パラメーターです。このパラメーターを指定すると、以下の SQLCODE のいずれかが戻されるときにアプリケーションは IBM メインフレーム・データベース・サーバー・データベースから切断します。

```
SQL30000N  
SQL30040N  
SQL30050N  
SQL30051N  
SQL30053N  
SQL30060N  
SQL30070N  
SQL30071N  
SQL30072N  
SQL30073N  
SQL30074N  
SQL30090N
```

切断パラメーター **,D** を指定しなくても、次の SQLCODE が戻された場合に限り切断を行います。

```
SQL30020N  
SQL30021N  
SQL30041N  
SQL30061N  
SQL30081N
```

これらのコードの説明については、「メッセージ・リファレンス」を参照してください。

注: エラーのために DB2 Connect が切断すると、自動的にロールバックされます。

„INTERRUPT_ENABLED

これは 3 番目の定位置パラメーターです。

INTERRUPT_ENABLED は、最後のサーバーが割り込みをサポートしない場合にのみ適用されます。サーバーが DRDA 割り込みフローをサポートする場合、DB2 Connect は単に割り込み要求をサーバーに渡します。

DB2 Connect ワークステーションの DCS ディレクトリーで INTERRUPT_ENABLED が構成されている場合、IBM メインフレーム・データベース・サーバーに接続している状態でクライアント・アプリケーションが割り込みを発行すると、DB2 Connect は、接続をドロップし、作業単位をロールバックすることによって割り込みを実行します。この割り込み動作は、AIX および Windows 上でサポートされません。

そのアプリケーションは、サーバーへの接続が終了したことを示す sqlcode (-30081) を受け取ります。アプリケーションが追加のデータベース要求を処理するには、その後で IBM メインフレーム・データベース・サーバーへの新しい接続を確立する必要があります。 AIX V5.2 以上および Windows 以外のプラットフォームでは、アプリケーションが割り込み要求を受信したときの、DB2 Connect による自動切断オプションはサポートされていません。

注: このサポートは、任意のプラットフォームでの TCP/IP 接続に対して機能します。クライアントがソケットを kill する場合がありますが、サーバー処理系によって未解決の受信になるかどうかが決まります。 DB2 for z/OS は非同期のソケット呼び出しを利用するので、接続の切断を検出し、進行中の長時間実行 SQL ステートメントをロールバックすることができます。

,,,,,SYSPLEX

これは 6 番目の定位置パラメーターで、特定のデータベースに対して DB2 Connect SYSPLEX サポートを明示的に使用可能にするときに使用します。

,,,,,LOCALDATE="*<value>*"

これは 7 番目の定位置パラメーターで、DB2 Connect の日付形式サポートを使用可能にするときに使用します。これは、*<value>* に対する日付マスクによりインプリメントされます。以下のとおりです。

例えば、次の CLP (コマンド行プロセッサ) ステートメントを発行したとします。

```
catalog TCPIP node nynode remote myhost server myport
catalog dcs database nydb1 as new_york
catalog database nydb1 as newyork1 at node nynode
authentication server
```

ホスト・データベースにアクセスするためにデータベース別名 *newyork1* を使用します。日付マスクは指定されていないので、日付変換は行われません。

ここで新しい日付形式サポートを使用すると、以下の CLP コマンドを発行することができます。この場合、CLP が使用されており、パラメーター・ストリング自体が二重引用符を使って指定されているため、LOCALDATE 値は 2 組の二重引用符に囲まれています。また、LOCALDATE 指定から二重引用符が外されないようオペレーティング・システムのエスケープ文字 "\$" (円記号) が使用されていることにも注目してください。

```
catalog dcs database nydb2 as new_york
parms "$",,,,,LOCALDATE="$"¥"YYYYMMDD¥"¥"¥"
catalog database nydb2 as newyork2 at node nynode
authentication server
```

データベース別名 `newyork2` は同じホスト・データベースへのアクセスを提供しますが、日付形式マスクも指定します。この例は、日付形式マスクが `LOCALDATE` キーワードを使って指定されること、また日付形式マスクが `DCS` ディレクトリー・エントリーの `PARMS` フィールドの中で 7 番目の定位置パラメーターであることを示しています。

日付マスクを有効にするには、以下の条件をすべて満たしていなければなりません。

1. `Y`、`M`、および `D` の数列はそれぞれ最大 1 回しか指定できません。ここで `Y` は年、`M` は月、また `D` は日を表す数字です。
2. 1 つの数列における `Y` の最大数は 4 個です。
3. 1 つの数列における `M` の最大数は 2 個です。
4. 1 つの数列における `D` の最大数は 2 個です。

例えば、以下の日付マスクはすべて有効です。

`"YYyyMmDd"` - `Y`、`M`、および `D` の数字は大/小文字を区別しない
`"MM+DD+YYYY"` - 10 バイトを超えるマスクを指定することも、マスクに `Y`、`M`、`D` 以外の文字を指定することも可能
`"abcYY+MM"` - `D` の数列を指定しないことも可能

以下の日付マスクはすべて無効です。

`"YYYYyMMDD"` - 数列に `Y` が 5 つあるため無効
`"YYYYMDDM"` - `M` の数列が 2 つあるため無効

日付形式マスクが無効でもエラーは出されません。マスクが無視されるだけです。日付マスクが有効であっても、そのマスクが必ず使用されるとは限りません。有効な日付マスクに基づく日付形式変換は、以下の条件をすべて満たしている場合にのみ実行されます。

1. SQL エラーがない。
2. 出力が、ISO 準拠形式 (ISO および JIS) の日付値である。
3. 出力データ域の長さが少なくとも 10 バイトある。この長さは、日付形式変換を実行しない場合でも、データ値を出力データ域に保管するのに最低必要なサイズです。この要件は、日付形式マスクが 10 バイトより短いときにも当てはまります。
4. `DCS` ディレクトリー・エントリーで有効な日付形式マスクが指定されており、そのマスクが出力データ領域に保管可能である。

,,,,,,BIDI=<ccsid>

これは 9 番目の定位置パラメーターで、デフォルトのサーバー・データベース `BiDi CCSID` の上書きに使用する双方向 (`BiDi`) `CCSID` を指定します。例えば、次のようにします。

```
" , , , , , , , BIDI=xyz"
```

xyz は上書きする CCSID を表しています。

ディレクトリー・カスタマイズ・ワークシート

ディレクトリー・カスタマイズ・ワークシートには、収集する必要がある情報が示されています。そのワークシートのコピーをとって、ご使用のシステム値を記入しておくくと便利です。

ノード・ディレクトリー・パラメーター

表1. ノード・ディレクトリー・パラメーター

パラメーター	例	使用値
ノード名	DB2NODE	
リモート・ホスト名 (TCP/IP ノード)	ZOSHOST	
サーバー (TCP/IP サービス名またはポート番号)	db2inst1c (または 446)	

注:

1. DRDA 用のデフォルト TCP/IP ポート番号は 446 です。
2. IBM メインフレーム・データベース・サーバーが SECURITY SOCKS をサポートしているか分からない場合は、TCP/IP ノードに SECURITY を指定しないでください。

DCS ディレクトリー・パラメーター

表2. DCS ディレクトリー・パラメーター

パラメーター	例	使用値
データベース名	DB2DB	
ターゲット・データベース名	NEW_YORK3	
アプリケーション・リクエスター		
パラメーター・ストリング	".....LOCALDATE=¥"¥"YYMMDD¥"¥"¥"	

システム・データベース・ディレクトリー・パラメーター

表3. システム・データベース・ディレクトリー・パラメーター

パラメーター	例	使用値
データベース名	DB2DB	
データベース別名	NYC3	
ノード名	DB2NODE	
認証	SERVER	

同一のデータベースへの複数項目の定義

各データベースについて、3つのディレクトリー（ノード・ディレクトリー、DCSディレクトリー、およびシステム・データベース・ディレクトリー）のおのおのに少なくとも1つの項目を定義する必要があります。場合によっては、データベースに複数の項目を定義したいこともあります。

例えば、IBMメインフレーム・データベース・サーバーから移植されたアプリケーションについてはSQLCODEマッピングをオフにして、クライアント/サーバー環境用に開発されたアプリケーションについてはデフォルトのマッピングを受け入れることができます。これは次のように行います。

- ノード・ディレクトリー内に1つの項目を定義します。
- DCSディレクトリー内に、異なるデータベース名を持つ2つの項目を定義します。一方の項目には、パラメーター・STRINGにNOMAPを指定します。
- システム・データベース・ディレクトリーに、異なるデータベース別名を持つ2つの項目、およびDCSディレクトリーで指定した2つのデータベース名を定義します。

両方の別名とも同じデータベースにアクセスし、一方はSQLCODEマッピングを用いて、他方はSQLCODEマッピングなしです。

BiDi データの処理

以下のセクションは、z/OSサーバーだけに適用されます。DB2 for IBM iサーバーについては、完全なBiDiサポートがすでに用意されているので、このフィーチャーを使用可能にしないでください。

BiDiデータを様々なプラットフォームで正しく処理するには、以下のBiDi属性が必要です。

- 数字の形状 (ARABIC または HINDI)
- 方向 (RIGHT-TO-LEFT または LEFT-TO-RIGHT)
- 形状 (SHAPED または UNSHAPED)
- 対称スワッピング (YES または NO)
- テキストの形式 (LOGICAL または VISUAL)

プラットフォームによってデフォルトが異なるため、DB2データのあるプラットフォームから別のプラットフォームに送信するときに問題が生じます。例えば、WindowsプラットフォームではLOGICAL UNSHAPEDデータを使用しますが、z/OSでは通常SHAPED VISUALデータ形式を使用します。そのため、BiDi属性のサポートがないと、DB2 for z/OSからWindows版のDB2 Connectに送信されるデータは正しく表示されません。

DB2 Connectとサーバー上のデータベースとの間でデータを交換する場合、通常、着信データの変換を実行するのは受信側です。普通は、同じ変換規則が、通常のコード・ページ変換の拡張である、BiDiのレイアウト変換にも適用されます。ただし、現在のところBiDi特有のCCSIDまたはBiDiレイアウト変換をサポートしているホストDB2製品はありません。そのため、DB2 Connectでは、サーバー・

データベースから受信したデータに加えて、サーバー・データベースに送信するデータに対しても BiDi レイアウト変換を実行するオプション機能が追加されました。

サーバー・データベースに送信するデータに対して DB2 Connect が BiDi レイアウト変換を実行するには、サーバー・データベースの BiDi CCSID を上書きする必要があります。このことは、サーバー・データベースに対応する DCS データベース・ディレクトリー項目の PARMS フィールドで、BIDI パラメーターを使って行います。

このフィーチャーの使用については、例で説明します。

CCSID 62213 (BiDi ストリング・タイプ 5) を実行しているヘブライ語の IBM Data Server Client が、CCSID 424 (BiDi ストリング・タイプ 4) を実行している DB2 ホスト・データベースにアクセスするとします。ただし、DB2 ホスト・データベースに格納されているデータは CCSID 62245 (BiDi ストリング・タイプ 10) であることが分かっています。

この場合、2 つの問題が生じます。まず、CCSID 424 と 62245 の間で BiDi ストリング・タイプがどのように異なるかを DB2 ホスト・データベースは理解していません。また、DB2 ホスト・データベースが 62213 という IBM Data Server Client CCSID を識別できないという問題もあります。このデータベースがサポートするのは、CCSID 62213 と同じコード・ページに基づく CCSID 62209 (BiDi ストリング・タイプ 10) だけです。

まず、DB2 ホスト・データベースに送信されるデータが BiDi ストリング・タイプ 6 形式になっていることを確認する必要があります。また、DB2 ホスト・データベースから受信するデータに対して BiDi レイアウト変換を実行する必要があることを DB2 Connect に知らせなければなりません。DB2 ホスト・データベースに対しては次のカタログを使用します。

```
catalog dcs database nydb1 as TELAVIV parms ",,,,,,,,,BIDI=62245"
```

ここでは、DB2 ホスト・データベース CCSID 424 を 62245 によって上書きするよう DB2 Connect に指示します。この上書きには以下の処理が関係しています。

1. DB2 Connect が CCSID 62209 (BiDi ストリング・タイプ 10) を使って DB2 ホスト・データベースに接続する処理。
2. DB2 ホスト・データベースに送信するデータに対して、DB2 Connect が、CCSID 62213 (BiDi ストリング・タイプ 5) から CCSID 62209 (BiDi ストリング・タイプ 10) への BiDi レイアウト変換を実行する処理。
3. DB2 ホスト・データベースから受信したデータに対して、DB2 Connect が、CCSID 62245 (BiDi ストリング・タイプ 10) から CCSID 62213 (BiDi ストリング・タイプ 5) への BiDi レイアウト変換を実行する処理。

注:

1. BIDI パラメーターを有効にするには、環境変数またはレジストリー値 **DB2BIDI** を YES に設定しなければなりません。**DB2BIDI** は、DCS データベース・ディレクトリー項目がカタログされている DB2 Connect ワークステーションで設定

する必要があります。DB2 Connect サーバーに対してリモート側のクライアント上で実行されているアプリケーションの場合、**DB2BIDI** 変数をそのクライアントでも設定する必要があります。

2. DB2 ホスト・データベースに送信するデータに対して DB2 Connect がレイアウト変換を実行するようにするには、たとえ CCSID を上書きする必要がない場合でも、DCS データベース・ディレクトリーの PARMS フィールドに BIDI パラメーターを追加しなければなりません。その場合、CCSID にはデフォルトの DB2 ホスト・データベース CCSID を提供してください。
3. 状況によっては、双方向の CCSID を使うことにより SQL 照会自体が変更され、DB2 サーバーで認識されなくなることがあります。特に、他のストリング・タイプが使用可能な場合、IMPLICIT CONTEXTUAL および IMPLICIT RIGHT-TO-LEFT CCSID の使用は避けてください。SQL 照会に引用符付きストリングが含まれている場合、CONTEXTUAL CCSID を使用すると予期せぬ結果を招くことがあります。SQL ステートメントでは引用符付きストリングの使用を避け、可能であれば代わりにホスト変数を使用してください。

特定の双方向 CCSID を使用したときにこれらの推奨処置では修正できない問題が生じる場合、環境変数またはレジストリー値 **DB2BIDI** を NO に設定する必要があります。

パラメーター・ストリングの指定

DCS パラメーターの例を以下に示します (各行が 1 つのパラメーター・セットです)。

```
NOMAP
/u/username/sql1lib/map/dcs1new.map,D
,D
,,INTERRUPT_ENABLED
NOMAP,D,INTERRUPT_ENABLED,,,SYSPLEX,LOCALDATE="YYMMDD",,
```

別の方法として、パラメーター・ストリングを指定せずにデフォルトを受け入れることもできます。

注: UNIX システムでオペレーティング・システムのコマンド行から CLP を使用する場合は、パラメーター・ストリングの LOCALDATE マスクを指定するときに 2 対の二重引用符を指定しなければならないので、オペレーティング・システムのエスケープ文字 "¥" (円記号) を使用する必要があります。例えば、次のようにします。

```
db2 catalog dcs db x as y parms ¥",,,,,LOCALDATE=¥"¥"YYMMDD¥"¥"¥"
```

結果として、次の DCS ディレクトリー項目になります。

DCS 1 entry:

```
Local database name           = X
Target database name         = Y
Application requestor name    =
DCS parameters                = ,,,,,,LOCALDATE="YYMMDD"
Comment                       =
DCS directory release level   = 0x0100
```


第 5 章 DB2 Connect のセキュリティー

ユーザーは DB2 Connect およびアクセス対象のデータが含まれるデータベースに対してローカルである場合もリモートである場合もあるため、DB2 Connect の使用においてユーザー認証は重要です。ここでは、トラステッド接続および Kerberos サポートについて説明し、ホスト・マシン上のデータベースに関するセキュリティーの考慮事項も記載しています。

DB2 Connect を介したトラステッド接続

一部の DB2 データベース・サーバー製品はトラステッド・コンテキストをサポートしています。トラステッド・コンテキスト は、特に、クライアント・アプリケーションがトラステッド接続を作成できる条件をデータベース管理者が定義できるようにします。トラステッド接続 は通常の接続では不可能なタスクを実行できます。

トラステッド接続には、暗黙的および明示的という 2 つのタイプがあります。接続の作成時に、明示的または暗黙的なトラステッド接続を取得するか、あるいは通常の接続を取得するかは、表 4 に要約されているように、トラステッド接続を要求するかどうか、さらにはサーバーのトラステッド・コンテキストに定義された基準をその接続が満たしているかどうかによって依存します。

表 4. 種々のアクションの組み合わせにより生じる接続のタイプ

	接続が信頼可能になるためのサーバーの基準を満たす場合	接続が信頼可能になるためのサーバーの基準を満たさない場合
トラステッド接続となるように要求する	明示的トラステッド接続	通常の接続および警告 SQL20360W (SQLSTATE 01679) が戻される。
トラステッド接続となるように要求しない	暗黙的トラステッド接続	通常の接続

暗黙的トラステッド接続 は、接続の使用時にユーザーに一時的なロール特権を付与するという以外は、通常の接続と同じです。(該当する場合) 付与されるロール特権はトラステッド・コンテキストで指定され、その結果、接続は信頼できるものとなります。

暗黙的トラステッド接続は、DB2 Connect を使用して接続する任意のアプリケーションによって作成できます。暗黙的トラステッド接続の作成や使用の方法は、通常の接続と同じです。つまり、既存のアプリケーションが DB2 Connect を介して接続している限りは、そのアプリケーションが暗黙的トラステッド接続を活用するためにコード変更は必要ありません。

明示的トラステッド接続 は、暗黙的トラステッド接続と同じ仕方で、ユーザーに対して一時的なロール特権を付与します。加えて、明示的トラステッド接続では、その接続を介してアクションを実行する場合に使用する許可 ID を変更することができます。明示的トラステッド接続での許可 ID の変更は、『ユーザーの切り替え』

と呼ばれます。トラステッド接続を作成できるようにするトラステッド・コンテキストの一部として、切り替え可能な許可 ID と、切り替え時に指定の許可 ID でパスワードが必要かどうか定義されています。

ユーザーの切り替えにより、複数のユーザーで接続を共用することによるオーバーヘッドを大幅に削減することができます。特に、パスワードが不要なユーザー名の場合には、データベース・サーバーが許可 ID を認証しないのでそう言えます。しかしこのフィーチャーの使用時には、必ずご使用のアプリケーションで、許可 ID を検証および認証することなく、その許可 ID を切り替えることができないようにしなければなりません。そのようにしないと、システムにセキュリティー・ホールを作ってしまうます。

明示的トラステッド接続を作成して、CLI または JDBC を使用した DB2 Connect を介して接続する際に (確立された XA 接続を含む)、ユーザーを切り替えることができます。明示的トラステッド接続の作成とユーザーの切り替えには、特別な接続属性の設定が必要です。つまり、明示的トラステッド接続の利点を生かすには、既存のアプリケーションを変更する必要があります。

前述した相違点以外は、(明示的または暗黙的のどちらでも) トラステッド接続は通常の接続と同じ方法で使用できます。しかし明示的トラステッド接続を切断するには、接続が壊れているか切断状態にある場合であっても、それを必ず明示的に切断しなければなりません。そのようにしないと、その接続が使用しているリソースが解放されない可能性があります。暗黙的トラステッド接続の場合には、これは問題となりません。

注:

1.

重要: パスワードを提供しないでユーザーを切り替えると、データベース・サーバーの認証がバイパスされます。ご使用のアプリケーションが既に許可 ID を検証して認証済みでない限りは、アプリケーションがパスワードなしでその許可 ID に切り替えられないようにしてください。そのようにしないと、セキュリティー・ホールができてしまいます。

2. 明示的トラステッド接続では、クライアント認証を使用しないでください。これは、暗黙的トラステッド接続には当てはまりません。

3. 明示的トラステッド接続を使用するアプリケーションは、パスワードで保護され、許可されたユーザーだけがアクセス可能な機密保護機能のあるマシンで実行してください。これは、暗黙的トラステッド接続には当てはまりません。

CLI を使用したトラステッド接続の作成および終了

CLI による接続時に明示的トラステッド接続を作成できるのは、接続しているデータベース・サーバーがそれを許可するように構成されている場合です。

この手順においては、XA トランザクション・マネージャーを使用していないと想定します。XA トランザクション・マネージャーを使用している場合に唯一必要となるのは、トランザクション・マネージャーが `xa_open` を呼び出す際に構成値 `TCTX` を `TRUE` に設定するよう構成されているのを確認することです。そのように

設定されていると、任意の接続を明示的のトラステッド接続にすることが可能になります。接続が明示的のトラステッド接続であることを確認するには、ステップ 3 を参照してください。

- 接続先のデータベースでトラステッド・コンテキストがサポートされている必要があります。
- クライアントを信頼可能として認識するトラステッド・コンテキストが定義されている必要があります。
- トラステッド・コンテキストで指定されているシステム許可 ID を把握していなければなりません。トラステッド接続のシステム許可 ID は、接続の作成時にユーザー名としてサーバーに提供した許可 ID です。接続が特定のトラステッド・コンテキストによって信頼可能であるとされるには、システム許可 ID がそのトラステッド・コンテキストで指定されている必要があります。有効なシステム許可 ID とその ID のパスワードについては、セキュリティー管理者に尋ねてください。

この手順の例では、C 言語を使用し、conn は有効ではあるものの、まだ接続されていない接続ハンドルへのポインターとします。変数、rc は SQLRETURN というデータ・タイプであるとします。

1. 通常の接続で設定している接続属性に加え、SQLSetConnectAttr 関数を呼び出して接続属性 SQL_ATTR_USE_TRUSTED_CONTEXT を SQL_TRUE に設定します。

```
rc = SQLSetConnectAttr(
    conn,
    SQL_ATTR_USE_TRUSTED_CONTEXT, SQL_TRUE, SQL_IS_INTEGER
);
```

2. インスタンスに対して SQLConnect 関数を呼び出し、通常の接続と同じようにしてデータベースに接続します。ユーザー名としてシステム許可 ID を、パスワードとして許可 ID のパスワードを使用します。必ずエラーと警告を確認します。特に、表 5 にリストされている項目について確認してください。

表 5. トラステッド接続を作成できなかったことを示すエラー

SQLCODE	SQLSTATE	意味
SQL20360W	01679	接続はトラステッド接続として確立できませんでした。代わりに通常の接続として確立されました。

ユーザーに報告すべきエラーまたは警告がなければ、明示的のトラステッド接続が確立されます。

3. (オプション) SQLGetConnectAttr 関数を使用して接続属性 SQL_ATTR_USE_TRUSTED_CONTEXT の値を検査すると、確立された接続が明示的のトラステッド接続であることを検証できます。SQL_TRUE に設定されている場合には、接続は明示的のトラステッド接続です。
4. 接続の使用を終了する場合には、接続が壊れているか切断状態にある場合であっても、必ず明示的に切断する必要があります。明示的のトラステッド接続を明示的に切断しないと、接続が使用している一部のリソースが解放されない可能性があります。

注:

1. 明示的トラステッド接続では、クライアント認証を使用しないでください。これは、暗黙的トラステッド接続には当てはまりません。
2. 明示的トラステッド接続を使用するアプリケーションは、パスワードで保護され、許可されたユーザーだけがアクセス可能な機密保護機能のあるコンピューターでのみ実行してください。これは、暗黙的トラステッド接続には当てはまりません。

CLI を使用したトラステッド接続のユーザーの切り替え

コマンド行インターフェース (CLI) を使用して、明示的トラステッド接続のユーザーを切り替えることができます。トラステッド接続を使用したユーザーの切り替えに関する説明については、関連リンクのトピックを参照してください。

- 接続は明示的トラステッド接続として正常に作成されたものでなければなりません。
- 明示的トラステッド接続はトランザクションに存在することはできません。
- 明示的トラステッド接続を作成できるトラステッド・コンテキストでは、切り替えようとしている許可 ID への切り替えが許可されるように構成されている必要があります。

この手順の例では、C 言語を使用し、**conn** は接続済み明示的トラステッド接続へのポインターとします。変数、*rc* は **SQLRETURN** というデータ・タイプであるとします。変数 *newuser* は切り替えるユーザーの許可 ID を保持する文字ストリングへのポインターとします。変数 *passwd* はその許可 ID のパスワードを含む文字ストリングへのポインターとします。

1. **SQL_ATTR_TRUSTED_CONTEXT_USERID** 属性を設定するために **SQLSetConnectAttr** 関数を呼び出します。切り替える許可 ID を設定します。

```
rc = SQLSetConnectAttr(
    conn,
    SQL_ATTR_TRUSTED_CONTEXT_USERID, newuser, SQL_NTS
);
//Check for errors
```

必ずエラーと警告を確認します。特に、表 6 にリストされている項目について確認してください。

表 6. ユーザーの切り替え時に新しい許可 ID を設定できなかったことを示すエラー

SQLCODE	意味
CLI0106E	この接続は、接続されていません。
CLI0197E	この接続は、トラステッド接続ではありません。
CLI0124E	提供された値に問題があります。例えば、NULL でないこと、または長すぎないことを確認してください。
CLI0196E	接続が、ユーザーの切り替えを妨げる作業単位に関係しています。ユーザーを切り替えられるようにするには、接続はトランザクションに存在することはできません。

2. (このトラステッド接続が許可されるトラステッド・コンテキストで、切り替えようとしている許可 ID のパスワードを必要としない場合はオプション)

SQLSetConnectAttr 関数を呼び出し、SQL_ATTR_TRUSTED_CONTEXT_PASSWORD 属性を設定します。新しい許可 ID のパスワードを設定します。

```
rc = SQLSetConnectAttr(
    conn,
    SQL_ATTR_TRUSTED_CONTEXT_PASSWORD, passwd, SQL_NTS
);
//Check for errors
```

必ずエラーと警告を確認します。48 ページの表 6 と表 7 の両方にリストされている項目について確認してください。

表 7. ユーザーの切り替え時にパスワードを設定できなかったことを示すエラー

SQLCODE	意味
CLI0198E	属性 SQL_ATTR_TRUSTED_CONTEXT_USERID がまだ設定されていません。

3. 通常の接続として続行します。XA トランザクション・マネージャーを使用している場合には、次の要求の一部としてユーザー切り替えが試行されます。使用していない場合には、データベースにアクセスする関数 (例えば SQLExecDirect) の次の呼び出しを開始する直前に、このユーザー切り替えが試行されます。どちらの場合であっても、通常確認するエラーと警告に加えて、表 8 にリストされているエラーも確かめてください。表 8 にあるエラーは、ユーザー切り替えが失敗したことを示しています。

表 8. ユーザーの切り替えが失敗したことを示すエラー

SQLCODE	意味
SQL1046N	このトラステッド接続が許可されているトラステッド・コンテキストは、切り替えようとしている許可 ID への切り替えが許可されるように構成されていません。トラステッド・コンテキストを変更しない限りは、この許可 ID への切り替えはできません。
SQL30082N	提供されたパスワードが、切り替えている許可 ID に対して正しくありません。
ネイティブ・エラー -20361 を伴う SQL0969N	一部のデータベース・レベル制約には、ユーザーへの切り替えを妨げるものがあります。

ユーザー切り替えが失敗すると、別のユーザーへの切り替えが正常に行われるまでは接続は未接続の状態になります。未接続状態にあるトラステッド接続でユーザーを切り替えることができますが、未接続の状態ではデータベース・サーバーにはアクセスできません。未接続状態にある接続は、ユーザーの切り替えが正常に行われるまではそのままの状態です。

注:

1. **重要:** パスワードを提供しないでユーザーを切り替えると、データベース・サーバーの認証がバイパスされます。ご使用のアプリケーションが既に許可 ID を検証して認証済みでない限りは、アプリケーションがパスワードなしでその許可 ID に切り替えられないようにしてください。そのようにしないと、セキュリティ・ホールができてしまいます。

2. `SQL_ATTR_TRUSTED_CONTEXT_USERID` 属性に `NULL` 値を指定するのは、トラステッド・コンテキストのシステム許可 ID (明示的トラステッド接続が作成された際に使用したユーザー ID) を指定するのと同じです。
3. 明示的トラステッド接続で `SQL_ATTR_TRUSTED_CONTEXT_USERID` 接続属性の値を正常に設定すると、接続は即時にリセットされます。リセットされると、元の接続の接続属性を使用して新しい接続が作成されたかようになります。接続属性の値を、システム許可 ID、`NULL`、または属性が現在保持しているのと同じ値に設定しても、リセットが生じます。
4. `SQL_ATTR_TRUSTED_CONTEXT_PASSWORD` 属性が設定されると、トラステッド接続が許可されているトラステッド・コンテキストがその許可 ID の切り替えユーザーに関して認証を必要としない場合であっても、ユーザー切り替えプロセスでパスワードが認証されます。結果として、不要なオーバーヘッドが生じません。この規則は、トラステッド・コンテキストのシステム許可 ID には当てはまりません。トラステッド・コンテキストのシステム許可 ID が、その切り替え時に認証を必要としない場合には、パスワードが提供されても認証されません。

DB2 Connect の認証に関する考慮事項

DB2 Connect 管理者は、System z または IBM Power Systems データベースの管理者と協力して、ユーザー名とパスワードを検証する場所を決めることができます。

- クライアントで
- System z または IBM Power Systems サーバーで
- サード・パーティー・システム (Kerberos) を介したシングル・サインオンおよび検証

注: リモート・クライアントが認証タイプを指定していない場合、クライアントはデフォルトの `SERVER_ENCRYPT` になります。このタイプがサーバーに受け入れられないと、クライアントは、サーバーから戻された適切な値を使用して再試行します。パフォーマンスを最適な状態にするために、認証タイプは常にクライアントで指定するようにし、この余分なネットワーク・フローを行わずに済むようにしてください。

DB2 Connect バージョン 8.2.2 (バージョン 8.1 フィックスバック 9 に相当) 以降では、認証ネゴシエーション中、ゲートウェイはもう受動的な役割だけではありませんでした。その代わりに、ゲートウェイは積極的な役割を担います。ゲートウェイにおいてデータベース・ディレクトリー項目で認証タイプを指定すると、クライアントでカタログされた認証タイプをオーバーライドします。クライアント、ゲートウェイ、およびサーバーのすべてにおいて、互換性のあるタイプを指定する必要があります。ゲートウェイでカタログされた認証タイプがデータベース・ディレクトリー項目に指定されていない場合、`SERVER` 認証がサーバーで要求されるデフォルト・タイプになります。しかしサーバーが `SERVER` 認証をサポートしていない場合には、依然としてクライアントとサーバー間でネゴシエーションが生じます。この動作は、認証タイプが指定されていない場合にはデフォルトが `SERVER_ENCRYPT` になるクライアントの動作とは対照的です。

SET CLIENT API の DB2NODE または SQL_CONNECT_NODE オプションがクライアントで設定されている場合には、ゲートウェイでカタログされた認証タイプは使用されません。そのような場合のクライアントとサーバー間のネゴシエーションは、引き続き厳密なものとなります。

DB2 Connect では、以下の認証タイプが使用できます。

CLIENT

クライアントでユーザー名とパスワードの妥当性を検査します。

DATA_ENCRYPT

クライアント/サーバー通信時のユーザー・データの暗号化機能を提供します。この認証タイプは、IBM Power Systems データベース・サーバー上ではサポートされていません。

KERBEROS

クライアントが、従来の ID とパスワードの組み合わせの代わりに Kerberos 認証を使ってサーバーにログインできるようにします。この認証タイプでは、サーバーとクライアントの両方が Kerberos に対応している必要があります。

SERVER

System z または IBM Power Systems サーバー・データベースでユーザー名とパスワードの妥当性を検査します。

SERVER_ENCRYPT

SERVER 認証の場合と同様に、System z または IBM Power Systems データベース・サーバーでユーザー名とパスワードの妥当性を検査します。ただし、転送されるユーザー ID とパスワードはクライアントで暗号化されません。

SERVER_ENCRYPT_AES

転送されるユーザー ID とパスワードは、Advanced Encryption Standard (AES) 暗号化アルゴリズムを使用してクライアントで暗号化され、System z データベース・サーバーで妥当性検査されます。

Kerberos 認証が独特なのは、クライアントがユーザー ID やパスワードをサーバーに直接渡さないという点です。代わりに、Kerberos がサード・パーティーの認証メカニズムとして機能します。ユーザーが ID とパスワードをクライアント端末で一度入力するだけで、Kerberos はこのサインオンを検証します。この後で、Kerberos は自動的かつ安全に、ユーザーの許可を、要求されたローカルおよびネットワーク・サービスに渡します。したがって、ユーザーはリモート DB2 サーバーにログインする際に ID やパスワードを再入力する必要がありません。Kerberos 認証で提供されるシングル・サインオン機能では、DB2 Connect と接続先データベース・サーバーの両方が Kerberos をサポートしている必要があります。

注: GSSPLUGIN 認証タイプのサポートはありません。

Kerberos のサポート

チケット発行システムを扱う Kerberos 認証レイヤーは、Windows 2000 の Active Directory 機構に組み込まれています。クライアント側およびサーバー側のアプリケーションはそれぞれ、Kerberos SSP (セキュリティー・サポート・プロバイダー)

のクライアント・モジュールおよびサーバー・モジュールと通信を行います。セキュリティ・サポート・プロバイダー・インターフェース (SSPI) は、Kerberos SSP、および他のセキュリティ・プロトコルに高水準のインターフェースを提供します。

標準セットアップ

DB2 を Kerberos 認証を加えて構成するには、以下の両方をセットアップしてください。

- ネットワークで共有している Active Directory 内に、DB2 用の許可ポリシー (サービスとして)
- Kerberos 鍵配布センター (KDC) 相互間の信用できるリレーションシップ

最も単純なシナリオでは、少なくとも 1 つの KDC 信用リレーションシップを構成します。つまり、クライアント・ワークステーションを制御している KDC と、IBM Power Systems、または System z のいずれかのシステムとの間のリレーションシップです。OS/390[®] バージョン 2 リリース 10 または z/OS バージョン 1 リリース 2 は、ホストが UNIX KDC の役割を担えるようにする RACF[®] 機構を通して Kerberos チケット処理を行います。

DB2 Connect は通常どおり、3 層の設定でルーター機能を提供します。DB2 Connect は、Kerberos セキュリティーが使用される時の認証における、いかなる役割も想定していません。むしろ、単にクライアントのセキュリティ・トークンを DB2 for IBM i または DB2 for z/OS に渡すだけです。DB2 Connect ゲートウェイが、クライアントのメンバーやホストの Kerberos レルムのメンバーになる必要はありません。

下位互換性

Kerberos サポートのための DB2 最小要件は以下のとおりです。

IBM Data Server Client:

バージョン 8

DB2 Connect:

バージョン 8

DB2 for z/OS:

バージョン 7

z/OS セキュリティーに関するヒント

このトピックでは DB2 for z/OS データベース・サーバーに接続している DB2 Connect のセキュリティに関するヒントを紹介しています。

拡張セキュリティ・フィールド

DB2 for z/OS の「拡張セキュリティ (Extended Security)」フィールドが、YES になっていることを確かめてください。このフィールドは、DB2 for z/OS DSNTIPR パネルに表示されます。

拡張セキュリティー・コード

DB2 for z/OS バージョン 5.1 よりも前のバージョンでは、ユーザー ID またはパスワードを指定した接続要求が失敗した場合、SQL30082 理由コード 0 が返されるものの、問題となっている点に関して他の指示は出されませんでした。

DB2 for z/OS バージョン 5.1 では機能が拡張され、拡張セキュリティー・コードをサポートするようになりました。拡張セキュリティーを指定すると、理由コードのほかに (PASSWORD EXPIRED) などの診断が追加されます。

これを活用するには、DB2 for z/OS の拡張セキュリティー用に ZPARM インストール・パラメーターの値を YES に設定する必要があります。EXTSEC=YES に設定するには、DB2 for z/OS のインストール・パネル DSN6SYSP を使用します。これを設定するために、DDF パネル 1 (DSNTIPR) を使用することもできます。デフォルト値は、EXTSEC=NO です。パスワードの有効期限が切れた場合、Windows、Linux、UNIX、および DB2 Connect を使用している Web アプリケーションはエラー・メッセージ SQL30082 を受け取ります。

検査済みの TCP/IP セキュリティー

DB2 のセキュリティー・オプション AUTHENTICATION=CLIENT をサポートさせる場合、DB2 for z/OS のインストール・パネル DSNTIP4 (DDF パネル 2) を使用して、検査済みの TCP/IP セキュリティーを YES に設定します。

デスクトップ ODBC および Java アプリケーションのセキュリティー

ワークステーション ODBC および Java アプリケーションは動的 SQL を使用します。この動的 SQL はインストールの際にセキュリティー上の問題の原因となることがあります。DB2 for z/OS では、所有者とバインダーのどちらの権限でも動的 SQL を実行できる、新しいバインド・オプション DYNAMICRULES(BIND) が導入されています。

DB2 および DB2 Connect には、DB2CLI.INI 構成ファイル内に新しい CLI/ODBC 構成パラメーター CURRENTPACKAGESET があります。これは適切な特権を持つスキーマ名に設定することが必要です。SQL SET CURRENT PACKAGESET schema ステートメントは、アプリケーションの各接続ごとに自動的に出されます。

DB2CLI.INI を更新するためには、ODBC マネージャーを使用します。

パスワード変更サポート

ユーザー ID のパスワードの有効期限が切れた場合は、SQL CONNECT ステートメントから SQLCODE -30082 理由コード 1 のようなエラー・メッセージが戻されます。DB2 Connect ではパスワードをリモートで変更することが可能です。DB2 for z/OS では DRDA を介して以下の CONNECT ステートメントを発行し、ユーザーのパスワードを変更できます。

```
CONNECT TO <database> USER <userid> USING <password>  
NEW <new_password> CONFIRM <new_password>
```

パスワード変更には、DB2 構成アシスタントの「パスワード変更」ダイアログを使用することもできます。

DB2 Connect でサポートされる認証タイプ

このトピックでは、DB2 Connect によってサポートされる認証とセキュリティー設定の様々な組み合わせをリストします。

TCP/IP 接続用の認証タイプ

TCP/IP 通信プロトコルは、ネットワーク・プロトコル層での認証オプションをサポートしません。認証タイプは認証が行われる場所を決定します。この表に示される組み合わせだけが、DB2 Connect によってサポートされます。認証設定は、DB2 Connect サーバーのデータベース・ディレクトリー・エントリーで行います。

表9. 有効な認証シナリオ

シナリオ	認証設定	検証
1	CLIENT	クライアント
2	SERVER	IBM メインフレーム・データベース・サーバー
3	SERVER_ENCRYPT	IBM メインフレーム・データベース・サーバー
4	KERBEROS	Kerberos セキュリティー
5	DATA_ENCRYPT	ホスト
6	SERVER_ENCRYPT_AES	ホスト・データベース・サーバー

認証タイプについて

以下の解説は前述の接続と、表9 にリストされた接続に適用されます。各シナリオについて、より詳細に説明します。

- シナリオ 1 は、ユーザー名とパスワードがリモート・クライアントでのみ検証される場合です。ローカル・クライアントの場合、ユーザー名とパスワードが検証されるのは DB2 Connect サーバーだけです。

ユーザーは、サインオンしたロケーションで認証されるよう期待されています。ユーザー ID はネットワークを通して送信されますが、パスワードは送信されません。このタイプのセキュリティーは、すべてのクライアント・ワークステーションが信頼性の高い適切なセキュリティー機構を持っている場合にのみ使われます。

- シナリオ 2 は、ユーザー名とパスワードが IBM メインフレーム・データベース・サーバーでのみ検証される場合です。ユーザー ID とパスワードは、ネットワークを介してリモート・クライアントから DB2 Connect サーバーへ、次いで DB2 Connect サーバーから IBM メインフレーム・データベース・サーバーへ送信されます。
- シナリオ 3 はシナリオ 2 と同じですが、ユーザー ID とパスワードが暗号化されるという点が異なります。
- シナリオ 4 では、Kerberos のチケットがクライアントによって Kerberos KDC から獲得されます。チケットは、DB2 Connect を通してサーバーに変更なしで渡され、ここで、サーバーが妥当性検査を行います。
- シナリオ 5 はシナリオ 3 と同じですが、ユーザーのデータも暗号化され、DATA_ENCRYPT が IBM Power Systems データベース・サーバーをサポートしていないという点が異なります。

- シナリオ 6 はシナリオ 3 と同じですが、Advanced Encryption Standard (AES) 暗号化アルゴリズムが使用されるという点が異なります。

第 6 章 アプリケーションとユーティリティのバインド (DB2 Connect)

組み込み SQL を使用して開発されたアプリケーション・プログラムは、それが作動するときに使用する各データベースとバインドされていなければなりません。コマンド・センターおよび構成アシスタントが使用可能なプラットフォームでは、これらの機能を使用してこれを行うことができます。

バインドは、各データベースにつき、アプリケーションごとに一度実行されなければなりません。バインド・プロセスの間に、データベース・アクセス・プランは、実行される各 SQL ステートメントごとに保管されます。これらのアクセス・プランは、アプリケーション開発者によって提供され、(プリコンパイルの間に作成された) バインド・ファイル に入っています。バインドとは、IBM メインフレーム・データベース・サーバーによりそれらのバインド・ファイルを処理するプロセスを指します。

DB2 Connect で提供されているいくつかのユーティリティは組み込み SQL を使って開発されているため、IBM メインフレーム・データベース・サーバーにバインドするまでそれらをシステムで使用することはできません。DB2 Connect ユーティリティおよびインターフェースを使用しない場合、それらを各 IBM メインフレーム・データベース・サーバーにバインドする必要はありません。これらのユーティリティに必要なバインド・ファイルのリストは、次のファイルに含まれています。

- ddcsmvslst (System z の場合)
- ddcsvse.lst (VSE の場合)
- ddcsvm.lst (VM の場合)
- ddcs400.lst (IBM Power Systems の場合)

これらのファイルのリストの 1 つをデータベースにバインドすることは、個々のユーティリティをそのデータベースへバインドすることになります。

DB2 Connect サーバー製品がインストールされている場合、DB2 Connect の各種ユーティリティをそれぞれの IBM メインフレーム・データベース・サーバーにバインドするまで、そのシステムで使用することはできません。クライアント側のフィックスバックが同じレベルであることを前提とすると、関係するクライアント・プラットフォームの数に関係なく、ユーティリティをバインドする必要があるのは一度だけです。

例えば、Windows サーバー上の DB2 Connect Enterprise Edition を介して、DB2 for z/OS に接続する Windows クライアントが 10、AIX クライアントが 10 あるとします。この場合は以下のいずれかを実行します。

- ddcsmvslst を Windows クライアントの 1 つからバインドします。
- ddcsmvslst を AIX クライアントの 1 つからバインドします。
- ddcsmvslst を DB2 Connect サーバーからバインドします。

この例では以下の条件が前提になります。

- すべてのクライアントが同じサービス・レベルにある。そうでない場合、特定のサービス・レベルの各クライアントからバインドする必要があります。
- サーバーはクライアントと同じサービス・レベルにある。そうでない場合、サーバーからも同様にバインドする必要があります。

DB2 Connect ユーティリティーに加えて、組み込み SQL を使用するその他のすべてのアプリケーションについても、その作業のときに使用する各データベースへバインドする必要があります。バインドされていないアプリケーションは、実行する時、通常は SQL0805N エラー・メッセージを出します。バインドを必要とするすべてのアプリケーションについて追加的なバインド・リスト・ファイルを作成したいときがあるかもしれません。

バインド先の IBM メインフレーム・データベース・サーバーごとに、以下の手順を実行します。

1. ホストまたは IBM メインフレーム・データベース・サーバー管理システムに対して十分な権限があることを確認します。

System z

必要な許可は以下のとおりです。

- SYSADM または
- SYSCTRL または
- BINDADD および CREATE IN COLLECTION NULLID

注: BINDADD および CREATE IN COLLECTION NULLID 特権は、パッケージがまだない場合にのみ十分な権限であるといえます。例えば、はじめてパッケージを作成する場合などです。

パッケージがすでに存在しており、それらを再度バインドしている場合は、タスクを完成させるために必要な権限は、誰がオリジナル・バインドを行ったかにかかってきます。

A) もし同一の人がオリジナル・バインドを行い、再度のバインドも行っている場合は、上に挙げた権限のどれを持っている場合でも、バインドを完了させることができます。

B) 最初のバインドを行ったのとは違う人が、2番目のバインドを行う場合は、バインドを完了するのに SYSADM または SYSCTRL 権限が必要です。BINDADD および CREATE IN COLLECTION NULLID 権限を有するだけでは、バインドを完了させることができません。ただし、SYSADM または SYSCTRL 権限がなくても、パッケージの作成は可能です。この状況では、置き換えようとする既存のパッケージのおのおのについて、BIND 特権が必要になります。

VSE または VM

DBA 権限が必要です。バインド・コマンドについて GRANT オプションを使用しようとする場合 (各 DB2 Connect パッケージに対するアクセス権を個別に付与することを避けるために)、NULLID ユーザー ID は、以下の表についての権限を他のユーザーに付与する権限を持たなければなりません。

- system.syscatalog
- system.syscolumns
- system.sysindexes
- system.systabauth
- system.syskeycols
- system.syssynonyms
- system.syskeys
- system.syscolauth
- system.sysuserauth

VSE または VM システムでは、次の命令を発行することができます。

```
grant select on table to nullid with grant option
```

IBM Power Systems

NULLID コレクションにある *CHANGE 権限またはそれ以上の権限。

2. 以下と同様のコマンドを実行します。

```
db2 connect to DBALIAS user USERID using PASSWORD
db2 bind path@ddcsmvs.lst blocking all
      sqlerror continue messages ddcsmvs.msg grant public
db2 connect reset
```

ここで、*DBALIAS*、*USERID*、および *PASSWORD* は IBM メインフレーム・データベース・サーバーに適用されます。 *ddcsmvs.lst* は z/OS 用のバインド・リスト・ファイル、*path* はバインド・リスト・ファイルのロケーションです。

例えば、*drive:%sqllib%bnd%* はすべての Windows オペレーティング・システムに、*INSTHOME/sqllib/bnd/* はすべての Linux および UNIX オペレーティング・システムに適用されます。ここで、*drive* は DB2 Connect がインストールされている論理ドライブ、*INSTHOME* は DB2 Connect インスタンスのホーム・ディレクトリーです。

bind コマンドの *grant* オプションを使用することにより、PUBLIC に対し、または特定のユーザー名またはグループ ID に対し、EXECUTE 特権を付与することができます。 *bind* コマンドの *grant* オプションを使用しない場合は、GRANT EXECUTE (RUN) を個別に与えなければなりません。

バインド・ファイルについてのパッケージ名を見つけるには、次のコマンドを入力します。

```
ddcspkgn @bindfile.lst
```

例えば、次のようにします。

```
ddcspkgn @ddcsmvs.lst
```

出力は次のようになります。

Bind File	Package Name
f:%sqllib%bnd%db2ajgrt.bnd	SQLAB6D3

DB2 Connect の場合にこれらの値を判別するには、次のように *ddcspkgn* ユーティリティーを実行します。

```
ddcspkgn @ddcsmvs.lst
```

任意に、このユーティリティを使用して、個々のバインド・ファイルのパッケージ名を判別することができます。例えば、次のようにします。

```
ddcspkgn bindfile.bnd
```

注:

- a. バインド・オプション `sqlerror continue` は必ず使用しなければなりません。ただし、DB2 ツールやコマンド行プロセッサ (CLP) を使ってアプリケーションをバインドする場合、このオプションは自動的に指定されます。このオプションを指定すると、バインド・エラーが警告に変わります。それで、エラーのあるファイルをバインドしても、パッケージを作成することができます。言い換えると、特定のサーバー処理系が別のサーバー処理系の SQL 構文に無効であるとフラグを付ける場合であっても、複数のサーバーに対して 1 つのバインド・ファイルを使用することができます。そのため、特定の IBM メインフレーム・データベース・サーバーに対して任意のリスト・ファイル `ddcsxxx.lst` をバインドする場合、警告が出されることがあります。
- b. DB2 Connect 経由で DB2 データベースに接続する場合、バインド・リスト `db2ubind.lst` を使用してください。 `sqlerror continue` は IBM メインフレーム・データベース・サーバーに接続するときだけ有効なので指定しないでください。また、DB2 データベースに接続する場合は、DB2 Connect ではなく DB2 で提供される DB2 クライアントを使用するようにお勧めします。
3. 各アプリケーションまたはアプリケーションのリストをバインドするには、類似のステートメントを使用します。
4. DB2 の以前のリリースからのリモート・クライアントがある場合には、これらのクライアント上にあるユーティリティを DB2 Connect にバインドする必要があります。

第 7 章 マルチサイト更新

マルチサイト更新 (分散作業単位 (DUOW) および 2 フェーズ・コミットともいう) は、アプリケーションが、整合性を保証しながら、複数のリモート・データベース・サーバー上のデータを更新する機能です。例えば、ある口座から、異なるデータベース・サーバー上にある別の口座に送金する銀行業務トランザクションはその一例です。

そのようなトランザクションでは、別の口座への貸方処理に必要な更新がコミットされるまで、1 つの口座からの借方操作を行う更新をコミットしないことが非常に重要です。マルチサイト更新に関する考慮事項が当てはまるのは、これらの口座を表すデータがそれぞれ別のデータベース・サーバーにより管理される場合です。

DB2 製品は、マルチサイト更新を包括的にサポートしています。このサポートは、通常の SQL を使って開発されたアプリケーションだけでなく、X/Open XA インターフェース仕様をインプリメントしたトランザクション・モニター (TP モニター) を使ったアプリケーションでも利用できます。そのような TP モニターの製品の例としては、IBM TxSeries CICS、IBM Message and Queuing Series、IBM Component Broker Series、IBM San Francisco Project、さらに Microsoft Transaction Server (MTS)、BEA Tuxedo、その他が挙げられます。ネイティブ SQL のマルチサイト更新を使うか、TP モニターのマルチサイト更新を使うかで、セットアップ要件は異なります。

ネイティブ SQL のマルチサイト更新プログラムも TP モニターのマルチサイト更新プログラムも、CONNECT 2 SYNCPOINT TWOPHASE オプションを指定してプリコンパイルする必要があります。どちらのプログラムでも、SQL Connect ステートメントを使って、後続の SQL ステートメントでどのデータベースを使用するかを指定することができます。トランザクションを調整することを DB2 に伝える (つまり、データベース接続を確立するための `xa_open` を DB2 に送信する) TP モニターがない場合は、DB2 ソフトウェアがトランザクションを調整するために使用されます。

TP モニターのマルチサイト更新を使用する場合、アプリケーションは TP モニターの API (例えば、CICS SYNCPOINT、MTS SetAbort()) を使って、コミットまたはロールバックを要求しなければなりません。ネイティブ SQL のマルチサイト更新を使用する場合は、通常の SQL COMMIT および ROLLBACK を使用します。

TP モニターのマルチサイト更新は、DB2 と、非 DB2 リソース・マネージャー (Oracle、Informix、SQLServer など) の両方にアクセスするトランザクションを調整できます。ネイティブ SQL のマルチサイト更新は DB2 サーバーだけに使用します。

マルチサイト更新トランザクションを実行するには、分散トランザクションに関与する各データベースが、分散作業単位 (DUOW) をサポートしていなければなりません。現在、分散トランザクションに参加するための DUOW サポートを提供している DB2 サーバーは以下のとおりです。

- DB2 for Linux、UNIX および Windows のバージョン 8 以降

- DB2 for z/OS バージョン 7 以降
- DB2 for IBM i

サポートされているデータベース・サーバーであれば、どのような組み合わせでも 1 つの分散トランザクションで更新することができます。例えば、アプリケーションで Windows の、DB2 データベース、DB2 for z/OS データベース、および DB2 for i のデータベースにある複数の表を、1 つのトランザクションで更新することができます。

コントロール・センターを使ったマルチサイト更新の使用可能化

コントロール・センターを使ってマルチサイト更新を行えます。

マルチサイト更新を使用可能にするには、以下のようにします。

1. コントロール・センターを立ち上げます。
2. [+] 記号をクリックし、ツリー・ビューを展開します。
3. 構成するインスタンスを右マウス・ボタンで選択します。ポップアップ・メニューが開きます。
4. 「マルチサイト更新」->「構成」の順にメニュー項目を選択します。「マルチサイト更新ウィザード」がオープンします。
5. 「下記の TP モニター名を使用する」を選択して、トランザクション・プロセッサ (TP) モニターを指定します。このフィールドには、使用可能にした TP モニターのデフォルトが表示されます。TP モニターを使用しないときは、「TP モニターを使用しない」を選択します。

「次へ」をクリックします。
6. TP モニターを使用している場合は、同期点管理プログラムの設定値を指定します。TP モニターを使用していない場合は、トランザクション・マネージャー・データベースを指定します。
7. 「完了」をクリックします。

コントロール・センターを使ったマルチサイト更新のテスト

コントロール・センターを使ってマルチサイト更新のセットアップをテストできます。

マルチサイト更新をテストするには、以下のようにします。

1. インスタンスを右マウス・ボタンで選択し、ポップアップ・メニューから「マルチサイト更新」->「テスト」の順にメニュー・オプションを選択します。「マルチサイト更新のテスト」ウィンドウが開きます。
2. 「使用可能 (Available)」リスト・ボックスにある使用可能データベースから、テストするデータベースを選択します。選択したデータベースを「選択済み」リスト・ボックスに入れたり出したりするには、2 つのリスト・ボックスの間にある矢印ボタン (> と >>) を使います。また、選択したユーザー ID およびパスワードを変更する場合は、「選択済み」リスト・ボックスで直接編集することが可能です。

3. 選択が済んだら、「OK」をクリックします。「マルチサイト更新のテスト結果」ウィンドウがオープンします。
4. 「マルチサイト更新のテスト結果」ウィンドウには、選択したデータベースのうち、どれが更新のテストに成功し、また失敗したかが表示されます。このウィンドウには、失敗したデータベースの SQL コードとエラー・メッセージも表示されます。「閉じる」をクリックしてウィンドウを閉じます。
5. 「閉じる」をクリックして、「マルチサイト更新のテスト」ウィンドウを閉じます。

マルチサイト更新と同期点管理プログラム

IBM メインフレーム・データベース・サーバーでは、Linux、Windows、UNIX および Web アプリケーションから出された分散トランザクションに DB2 Connect が関与する必要があります。さらに、IBM メインフレーム・データベース・サーバーが関係した多くのマルチサイト更新シナリオでは、同期点管理プログラム (SPM) コンポーネントを構成する必要があります。DB2 インスタンスが作成されるときに、DB2 SPM は自動的にデフォルト設定で構成されます。

SPM が必要かどうかは、使用するプロトコル (TCP/IP)、および TP モニターを使用するかどうかによって決まります。以下の表では、SPM を使用する必要のあるシナリオがまとめられています。さらに、この表は Intel® または UNIX マシンから IBM メインフレームにアクセスする場合に、DB2 Connect が必要かどうかを示しています。マルチサイト更新では、TP モニターを使用する場合に DB2 Connect の SPM コンポーネントが必要です。

表 10. SPM を必要とするマルチサイト更新のシナリオ - TCP/IP

トランザクション・プロセッサ・モニターを使用	同期点管理プログラムが必要	必要な製品 (1 つ選択)	サポートされている IBM メインフレーム・データベース
はい	はい	DB2 Connect サーバー製品 DB2 Connect との DB2 Enterprise Server Edition ライセンスを適用	DB2 for z/OS V7 DB2 for z/OS V8 以降
いいえ	いいえ	DB2 Connect Personal Edition DB2 Connect サーバー製品 DB2 Connect との DB2 Enterprise Server Edition ライセンスを適用	DB2 for z/OS V7 DB2 for z/OS V8 以降

注: サポートされているデータベース・サーバーであれば、どのような組み合わせでも 1 つの分散トランザクションで更新することができます。例えば、アプリケーションで Windows の、DB2 データベース、DB2 for z/OS データベース、および DB2 for IBM i のデータベースにある複数の表すべてを、1 つのトランザクションで更新することができます。

DB2 Connect と XA 準拠トランザクション・マネージャーとの構成

ここでは、TP モニターで IBM Power Systems および System z のデータベース・サーバーを使用するのに必要な構成手順について説明します。

すでに操作可能な TP モニターがあって、DB2 Connect がインストールされていることを前提にしています。また、IBM メインフレーム・データベース・サーバーへの接続の構成とテストも済んでいなければなりません。

DB2 Connect が TP モニター内にある IBM Power Systems および System z のデータベース・サーバーを使用するように構成するには、以下の手順に従ってください。

1. TP モニターが DB2 XA スイッチにアクセスできるように構成します。DB2 XA スイッチは、TP モニターに DB2 Connect の XA API のアドレスを提供します。これを行う方法は TP モニターごとに異なっています。
2. TP モニターを DB2 の XA_OPEN スtring を使って構成します。これを行う方法は TP モニターごとに異なっています。また、DB2 の XA_OPEN スtring を TP モニターが使用するように構成する方法については、ご使用の TP モニターの資料を参照してください。
3. 必要に応じて、DB2 Connect 同期点管理プログラム (SPM) のデフォルトの構成パラメーターを変更します。IBM ホストおよび System i (バージョン 5 リリース 3 とそれ以前のバージョン) のデータベース・サーバーは現在のところ、XA インターフェースをサポートしていません。System i バージョン 5 リリース 4 とそれ以降のバージョンは XA を完全にサポートしています。

SPM は DB2 Connect のコンポーネントの 1 つで、XA の 2 フェーズ・コミット・プロトコルを、IBM メインフレーム・データベース・サーバーが使用する 2 フェーズ・コミット・プロトコルにマップします。デフォルトの状態では、DB2 インスタンスに SPM 構成パラメーターの事前定義値が指定されています。最も重要なパラメーターは、データベース・マネージャーの構成パラメーター SPM_NAME です。TCP/IP ホスト名の最初の 7 文字からとった名前がデフォルトになっています。

TCP/IP を使って DB2 for z/OS への接続を行うときは、デフォルトの設定値はどれも変更する必要はありません。この場合、SPM はすでに動作可能になっているため、SPM の構成作業は不要です。

疎結合トランザクションのための DB2 Connect サポート

疎結合トランザクションのための DB2 Connect 内でのサポートは、DB2 for IBM i バージョン 5 リリース 4 以降および DB2 for z/OS バージョン 7 以降を使用する XA 分散アプリケーションをインプリメントするユーザーを意図しています。このサポートにより、同じグローバル・トランザクションの異なるブランチが、DB2 for z/OS のロック・スペースを共有することができます。

疎結合トランザクションのサポートは、.NET および COM+ アプリケーションが対象となっています。

このフィーチャーにより、分散トランザクションのブランチが、同じグローバル・トランザクション内の別のブランチが原因でロック・タイムアウトやデッドロックになる時間枠を小さくすることができます。

第 8 章 DB2 Connect によるデータの移動

ホスト・データベース・システムとワークステーションの間でデータを移動する必要がある複合環境では、DB2 Connect (ホストとワークステーションの間のデータ転送のゲートウェイ) を使用できます (図 8 を参照)。

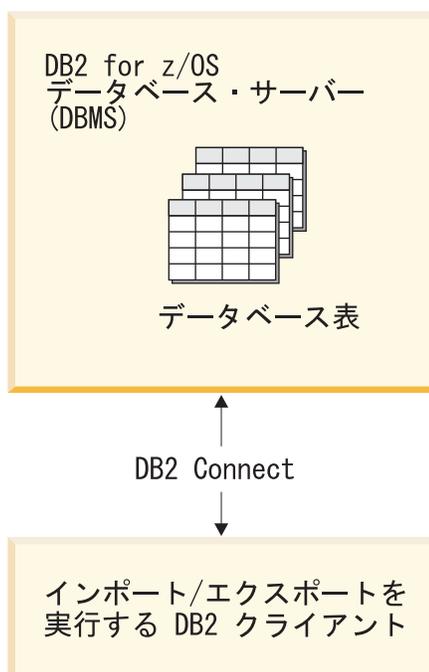


図 8. DB2 Connect によるインポート/エクスポート

DB2 のエクスポートおよびインポート・ユーティリティーを使用すると、IBM メインフレーム・サーバー・データベースから DB2 Connect ワークステーション上のファイルに、またはその逆にデータを移動できます。その後、このエクスポートおよびインポート・フォーマットをサポートしている他のすべてのアプリケーションやリレーショナル・データベース管理システムで、データを使用できます。例えば、IBM メインフレーム・サーバー・データベースから PC/IXF ファイルにデータをエクスポートして、さらにそれを DB2 Database for Linux, UNIX, and Windows データベースにインポートすることができます。

エクスポートおよびインポート操作は、データベース・クライアントから、または DB2 Connect ワークステーションから実行できます。

注:

1. エクスポートまたはインポートされるデータは、両方のデータベースに適用されるサイズとデータ・タイプの制約事項に従っていなければなりません。
2. インポートのパフォーマンスを改善するため、コンパウンド照会を使用することができます。インポート・ユーティリティーで compound ファイル・タイプ修飾

子を指定することにより、指定した数の照会ステートメントをブロックにまとめてください。このようにすればネットワーク・オーバーヘッドが少なくなり、応答時間が改善されます。

DB2 Connect を使用する場合、エクスポートおよびインポートの操作は次の条件を満たしている必要があります。

- ファイル・タイプは PC/IXF でなければなりません。
- インポート開始前に、データと互換性のある属性のターゲット表がターゲット・サーバー上に作成されていなければなりません。ソース表の属性を取得するには、db2look ユーティリティを使用できます。DB2 Connect によるインポートでは、サポートされているオプションは INSERT だけなので、表は作成できません。

これらの条件のいずれかが満たされていない場合、操作は失敗し、エラー・メッセージが戻されます。

注: 索引定義はエクスポートにおいて保管されず、インポートにおいて使用されません。

混合データ (1 バイト・データと 2 バイト・データの両方の入った列) をエクスポートまたはインポートする場合は、以下のことを考慮してください。

- EBCDIC でデータを保管するシステム (MVS、System z、IBM Power Systems、VM、および VSE) では、シフトアウトおよびシフトイン文字がそれぞれ 2 バイト・データの開始と終了を表します。データベース表の列の長さを定義する場合は、これらの文字のための十分な余地を見込んでください。
- 列データのパターンが一貫しているのではない限り、文字タイプの可変長列を使用することをお勧めします。

ワークステーションからホスト・サーバーへのデータの移動

データをホストまたは System i サーバー・データベースに移動するには、以下を実行してください。

1. DB2 表から PC/IXF ファイルにデータをエクスポートします。
2. INSERT オプションを使って、PC/IXF ファイルをホスト・サーバー・データベース内の互換性のある表にインポートします。

ホスト・サーバー・データベースからワークステーションにデータを移動するには、次のようにします。

1. ホスト・サーバー・データベースの表から PC/IXF ファイルにデータをエクスポートします。
2. PC/IXF ファイルを DB2 表にインポートします。

例

以下の例では、ワークステーションからホストまたは System i サーバー・データベースにデータを移動する方法を示します。

次のコマンドを発行して、外部 IXF フォーマットにデータをエクスポートします。

```
db2 export to staff.ixf of ixf select * from userid.staff
```

次のコマンドを発行して、ターゲット DB2 データベースへの DRDA 接続を確立します。

```
db2 connect to cbc664 user admin using xxx
```

まだ存在していない場合には、次のようにしてターゲット DB2 データベース・インスタンスにターゲット表を作成します。

```
CREATE TABLE mydb.staff (ID SMALLINT NOT NULL, NAME VARCHAR(9),  
DEPT SMALLINT, JOB CHAR(5), YEARS SMALLINT, SALARY DECIMAL(7,2),  
COMM DECIMAL(7,2))
```

データをインポートするには、以下のコマンドを発行します。

```
db2 import from staff.ixf of ixf insert into mydb.staff
```

IXF フォーマットのファイルからデータの各行が読み取られ、表 mydb.staff に行を挿入するために、SQL INSERT ステートメントが発行されます。すべてのデータがターゲット表に挿入されるまで、単一行が引き続き挿入されます。

詳細については、「Moving Data Across the DB2 Family」という IBM Redbooks® を参照してください。この Redbooks の資料は、<http://www.redbooks.ibm.com/redbooks/SG246905/> という URL にあります。

第 9 章 SQLCODE マッピング

異なる IBM リレーショナル・データベース製品は、同様のエラーについて必ずしも同じ SQLCODE を発行するわけではありません。SQLCODE が同じであるとしても、指定の異なるトークンを伴うことがあります。トークンのリストは、SQLCA の SQLERRMC フィールド内で渡されます。デフォルトでは、DB2 Connect は SQLCODE およびトークンを各 IBM メインフレーム・データベース・サーバーから該当する DB2 SQLCODE へマッピングします。

SQLCODE マッピングをオフにするときは、DCS ディレクトリーのパラメーター・ストリングで **NOMAP** を指定します。

アプリケーションを IBM メインフレーム・データベース・サーバー DB2 for z/OS などから直接移植する場合は、SQLCODE マッピングをオフにするとうよいでしょう。このようにすれば、アプリケーションが参照する SQLCODE を変更することなく、アプリケーションを使用することができます。

SQLCODE マッピングをオフにする

SQLCODE マッピングをオフにするときは、DCS ディレクトリーのパラメーター・ストリングで **NOMAP** を指定します。

アプリケーションを IBM メインフレーム・データベース・サーバー DB2 for z/OS などから直接移植する場合は、SQLCODE マッピングをオフにするとうよいでしょう。このようにすれば、アプリケーションが参照する SQLCODE を変更することなく、アプリケーションを使用することができます。

SQLCODE マッピングの調整

デフォルトでは、DB2 Connect は SQLCODE およびトークンを各 IBM メインフレーム・データベース・サーバーから該当する DB2 SQLCODE へマッピングします。以下のファイルは、デフォルトの SQLCODE マッピングのコピーです。

- dcs1dsn.map は、DB2 for z/OS の SQLCODE をマップします。
- dcs1ari.map は、DB2 Server for VM and VSE の SQLCODE をマップします。
- dcs1qsq.map は、DB2 for IBM i の SQLCODE をマップします。

Linux または UNIX オペレーティング・システム上の DB2 ではマッピングは必要ありません。

1. デフォルトの SQLCODE マッピングをオーバーライドする場合、または SQLCODE マッピングを持たない IBM メインフレーム・データベース・サーバー (非 IBM データベース・サーバー) を使用している場合は、これらのファイルの 1 つをコピーして、それを新規の SQLCODE マッピング・ファイルの基礎として使用することができます。そのファイルを直接編集するよりは、むしろそれをコピーすることによって、必要な場合いつでも元の SQLCODE マッピングを確実に参照できます。

2. DCS ディレクトリーのパラメーター・ストリングに、新規の SQLCODE マッピング・ファイルのファイル名を指定します。
3. 各マッピング・ファイルは、ASCII エディターを使用して作成、編集できる ASCII ファイルです。初期インストールでは、ファイルはインストール・パス内の map ディレクトリーに保管されます。

ファイルには、以下の特殊タイプの行を含めることができます。

- &&** ファイルの論理的な部分の開始。 && が最初に現れるよりも前の行はすべて、フリー・フォームのコメントと見なされて無視されます。ファイルが && の後に何も含んでいない場合は、SQLCODE マッピングは実行されません。また、前に説明したように、NOMAP パラメーターを使用することにより SQLCODE マッピングをオフにすることもできます。
- *** 行の最初の文字の場合、コメントを示します。
- W** 行にこの文字しかない場合、警告フラグが再マップされるべきことを示します。デフォルトでは、オリジナルの警告フラグが渡されます。W は大文字でなければなりません。

&& の後のその他のすべての行は、ブランクか、または以下の書式のマッピング・ステートメントのいずれかにしてください。

```
input_code [, output_code [, token_list]]
```

input_code は、以下のいずれかを表します。

sqlcode

IBM メインフレーム・データベース・サーバーからの SQLCODE

U すべての未定義の負の数の SQLCODE (このファイルにはリストされていないもの) は、指定された *output_code* にマップされます。*output_code* がこの行で指定されなかった場合は、元の SQLCODE が使用されます。この文字は必ず大文字です。

P すべての未定義の正の数の SQLCODE (このファイルにはリストされていないもの) は、指定された *output_code* にマップされます。*output_code* がこの行で指定されなかった場合は、元の SQLCODE が使用されます。この文字は必ず大文字です。

ccnn IBM メインフレーム・データベース・サーバーからの SQLSTATE クラス・コード。 *nn* は以下のいずれかです。

- 00** 無条件正常終了
- 01** 警告
- 02** データなし
- 21** カーディナリティー違反
- 22** データ例外
- 23** 制約違反
- 24** 無効なカーソル状態
- 26** 無効な SQL ステートメント ID
- 40** トランザクション・ロールバック

- 42 アクセス違反
- 51 無効なアプリケーション状態
- 55 前提条件の状態にないオブジェクト
- 56 その他の SQL または製品エラー
- 57 利用不能なリソースまたはオペレーター介入
- 58 システム・エラー

マッピング・ファイル内で明示的に指定されていないこのクラス・コードを伴うすべての SQLCODE に対して、指定の *output_code* が使用されます。この行で *output_code* が指定されない場合は、元の SQLCODE が、コピーされたトークンなしでそれ自体へマップされます。

cc の文字は必ず小文字です。

同じ *input_code* が 2 回以上マッピング・ファイルに現れた場合は、最初に現れたものが使用されます。*output_code* は、出力 SQLCODE を表します。値が指定されていない場合は、元の SQLCODE が使用されます。

また、出力コードを指定する場合は、以下のいずれかを指定することもできます。

- (s) 入力 SQLCODE に製品 ID (ARI, DSN または QSQ) をつなげたものが、SQLCA メッセージ・トークン・フィールドに入れられます。

元の SQLCODE は、唯一のトークンとして戻されます。このオプションは、+965 と -969 の例外を除き、未定義の SQLCODE を扱うように設計されています。+965 または -969 が *output_code* である場合、SQLCA の SQLERRMC フィールドに戻されるトークン・リストには、元の SQLCODE、次に製品 ID、続いて元のトークン・リストが含まれます。

s の文字は必ず小文字です。

(トークン・リスト)

トークンのリストで、コンマで分離します。特定のトークンをスキップするには、コンマだけで指定します。例えば、書式 (*t2,,t4*) は、1 番目と 3 番目の出力トークンは NULL であることを意味します。

各トークンは、番号の書式 (*n*) を持っており、オプションとしてその前に **c**、その後に **c** または **i** を付けます。以下のとおりに解釈されます。

- c** この位置にあるトークンのデータ・タイプは CHAR (デフォルト) です。**c** が *n* の前にある場合は、入力トークンを示します。**c** が *n* の後にある場合は、出力トークンを示します。**c** の文字は必ず小文字です。
- i** この位置にあるトークンのデータ・タイプは INTEGER です。**i** が *n* の後にある場合は、出力トークンを示します。**i** は *n* の前には来ません。というのは、IBM メインフレーム・データベース・サーバー製品は、CHAR トークンだけをサポートするからです。**i** の文字は必ず小文字です。
- n** どの IBM メインフレーム・データベース・サーバーのトークン

が使用されているかを示す数字です。それらの数字は、出力 SQLCA 内で配置要求される順序で配置されています。その数字は、IBM メインフレーム・データベース・サーバーのトークンを示しています。その配置は、トークンが SQLCA 内で並べられる順序を示しています。

例えば、IBM メインフレーム・データベース・サーバーが 2 つのトークン、1 および 2 を戻すとします。トークン 2 を出力 SQLCA の中でトークン 1 の前に表示する場合は、(2,1) と指定します。

複数のトークン数字は、それらをピリオドで接続することによって、1 つの CHAR 出力トークンを形成するよう結合させることができます。

コンマは、出力トークンを分離するために使用します。コンマの前にトークンが指定されていない場合、その位置では SQLCA 内に出力トークンは組み込まれません。最後に指定されたトークンに続いて出力 SQLCA に出現するトークンはすべて、NULL トークンへマップされます。

図 9 は、SQLCODE マッピング・ファイルのサンプルを示します。

&&			
-007	,	-007	, (1)
-010			
-060	,	-171	, (2)
...			
-204	,	-204	, (c1.2c)
...			
-633	,	-206	, (,c1i)
-30021	,	-30021	, (c1c,c2c)
cc00	,	+000	
...			
U	,	-969	, (s)
P	,	+965	, (s)

図 9. SQLCODE マッピング・ファイル

以下の説明は、上の図の一致する行番号に対応しています。

1. SQLCODE は -007 から -007 へマップされます。IBM メインフレーム・データベース・サーバーから受信する 1 番目の入力トークンは、1 番目の出力トークンとして使用され、デフォルトの設定である CHAR になります。他のトークンは、転送されません。
2. SQLCODE は、-010 から -010 へとマップされます (出力 SQLCODE が指定されていないので)。トークンは、出力 SQLCA に配置されません。
3. SQLCODE は -060 から -171 へマップされます。IBM メインフレーム・データベース・サーバーから受け取った 1 番目の入力トークンは、破棄されます。2 番目のトークンは、出力 SQLCA で 1 番目のトークンとして使用されます。それは CHAR です。出力 SQLCA に 2 番目のトークンはありません。
4. SQLCODE は -204 から -204 へマップされます。IBM メインフレーム・データベース・サーバーから受け取る 1 番目と 2 番目のトークンは CHAR です。

この 2 つの入力トークンは結合され、SQLCA で 1 番目の出力トークンになる 1 つの CHAR 出力トークンを形成します。

5. SQLCODE は -633 から -206 へマップされます。IBM メインフレーム・データベース・サーバーから受け取られた 1 番目の入力トークンは CHAR です。それは INTEGER へ変換され、出力 SQLCA で 2 番目のトークンとして使用されます。出力 SQLCA の 1 番目のトークンは、コンマで示されているように、NULL です。
6. SQLCODE は -30021 から -30021 へマップされます。IBM メインフレーム・データベース・サーバーから受け取る 1 番目および 2 番目の入力トークンは CHAR です。それらは、出力 SQLCA 内で 1 番目および 2 番目のトークンとして使用されます。
7. 00 クラスの SQLSTATE を指定した SQLCA 内の SQLCODE はすべて、SQLCODE +000 へマップされます。
8. 未定義の SQLCODE はすべて -969 へマップされます。まったく同一なものやマッピングが必要でないものも含め、マップ可能なコードのすべてがリストされている場合に限り、このオプションを使用してください。(s) オプションは、SQLCA の SQLERRMC フィールドに戻されるトークン・リストに、元の SQLCODE、次に、エラーが発生した製品、続いて元のトークン・リストが含まれることを示します。U 項目が組み込まれていない場合、リストされていないコードはすべて、マッピングされずに渡されます。
9. 未定義で正の SQLCODE はすべて、+965 へマップされます。まったく同一なものやマッピングが必要でないものも含め、マップ可能なコードのすべてがリストされている場合に限り、このオプションを使用してください。(s) オプションは、SQLCA の SQLERRMC フィールドに戻されるトークン・リストに、元の SQLCODE、次に、警告が発生した製品、続いて元のトークン・リストが含まれることを示します。P 項目が組み込まれていない場合、リストされていない正のコードはすべて、マッピングされずに渡されます。

第 10 章 データベース・システム・モニターと DB2 Connect

ここでは、DB2 Connect を使用する環境で接続およびパフォーマンスをモニターするいくつかの方法について説明します。行われるモニターのタイプは、オペレーティング・システムに固有のものであります。

リモート・クライアントの接続のモニター

データベース・システム・モニター を DB2 Connect Enterprise Edition のような DB2 Connect サーバー製品と併用し、リモート・クライアント接続をモニターできます。DB2 Connect サーバーにとってローカルであり、そのサーバー本体の上で稼働しているクライアントをモニターするには、次の変数を設定する必要があります。

```
db2set DB2CONNECT_IN_APP_PROCESS=NO
```

例えば、IBM メインフレーム・システムでエラーが発生したとき、システム管理者は、問題が DB2 Connect ワークステーションで起きたものかどうかを判別することができます。データベース・システム・モニターは、以下のものを相互に関連付けます。

- DRDA 相関トークン (CRRTKN) (無保護会話用)
- 作業単位 ID (UOWID) (DRDA-3 同期点マネージャーによって保護される 2 フェーズの接続用 (TCP/IP 接続で使用される))
- DB2 Connect 接続 ID (アプリケーション ID)

この情報は、どの DB2 Connect 接続が問題を起こしたかを示し、それによってシステム管理者は、DB2 Connect 接続を使用するその他のクライアントに影響を与えずに個別のクライアント・アプリケーションをそのシステムから強制切断することができます。

モニター・スイッチ状況のリスト

モニター・スイッチの状況をリストするには、`db2 get monitor switches` コマンドを発行してください。

Windows パフォーマンス モニタを使ったパフォーマンスのモニター

Windows オペレーティング・システムでは、DB2 アプリケーションのパフォーマンスをモニターするための便利なツールが用意されています。それは Windows 管理ツールの 1 つであるパフォーマンス モニタです。このツールは、システムのパフォーマンスをグラフに表示します。モニターの対象としてさまざまなシステム、データベース、通信関連の項目を選び、それらをまとめてグラフ表示にマップすることができます。

例えば、`GET SNAPSHOT FOR ALL DCS DATABASES` コマンドや `GET SNAPSHOT FOR ALL DCS APPLICATIONS` コマンドで取得できるレポートを、このモニターを使ってリアルタイムにグラフ化し、CPU 使用率といった値と直接比較

できます。また、種々の設定がデータベースや通信のパフォーマンスに与える影響も直接比較できます。設定値の個々の構成内容は PMC ファイルに保管して、後で取り出すことができます。

例えば、以下に示す図では CPU 使用率を軸に、いくつかの DB2 測定値をグラフ化しています。ここでグラフ化されている値の集合は、ファイル db2chart.pmc に保管されたものです。PMC ファイルはいくつでも保管することができ、それぞれのファイルにシステム・パフォーマンスの横断図を反映させることが可能です。

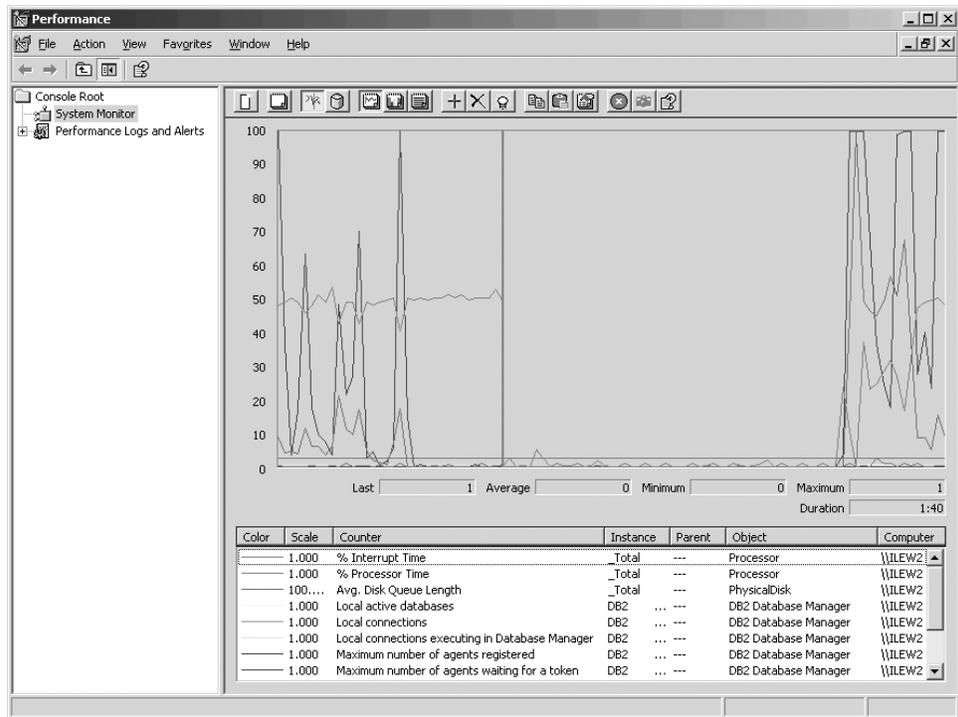


図 10. パフォーマンス モニタ

ローカル・アプリケーションのモニターを有効にするには、DB2CONNECT_IN_APP_PROCESS 環境変数をオフにする必要があります。

GET SNAPSHOT コマンドの使用

DB2 モニターは、有用なシステム情報を絶えず記録しています。GET SNAPSHOT コマンドを実行すれば、システム状況のサマリー情報をいつでも入手できます。モニター・スナップショットは、モニターするデータベース・マネージャーのインスタンスに対して SYSMANT, SYSCTRL, SYSADM のいずれかの権限があればとることができます。

DCS 情報をモニターするのに役立つスナップショット・コマンドは 5 つあります。それらのコマンドは以下のとおりです。

- GET SNAPSHOT FOR ALL DCS DATABASES
- GET SNAPSHOT FOR ALL DCS APPLICATIONS
- GET SNAPSHOT FOR DCS APPLICATION ...
- GET SNAPSHOT FOR DCS DATABASE ON db_alias

- GET SNAPSHOT FOR DCS APPLICATIONS ON db_alias

スナップショット・コマンドは要求した領域に関する詳細なレポートを生成します。

例えば、GET SNAPSHOT FOR DCS DATABASE ON DCSDB を発行すると、以下に示すレポートが生成されます。

DCS Database Snapshot

```
DCS database name           = DCSDB
Host database name         = GILROY
First database connect timestamp = 12-15-2001 10:28:24.596495
Most recent elapsed time to connect = 0.950561
Most recent elapsed connection duration = 0.000000
Host response time (sec.ms) = 0.000000
Last reset timestamp      =
Number of SQL statements attempted = 2
Commit statements attempted = 1
Rollback statements attempted = 0
Failed statement operations = 0
Total number of gateway connections = 1
Current number of gateway connections = 1
Gateway conn. waiting for host reply = 0
Gateway conn. waiting for client request = 1
Gateway communication errors to host = 0
Timestamp of last communication error = None
High water mark for gateway connections = 1
Rows selected              = 0
Outbound bytes sent        = 140
Outbound bytes received    = 103
```

このレポートは、データベース接続、パフォーマンス、SQL 要求のエラーやスループットなどに関する情報を提供します。しかし実際には、DB2 モニター・スナップショットにはさらに多くの詳細情報が含まれています。例えば、GET SNAPSHOT FOR ALL DCS APPLICATIONS コマンドを実行すると、以下のようなレポートが表示されます。

DCS Application Snapshot

```
Client application ID      = 09150F74.B6A4.991215152824
  Sequence number          = 0001
  Authorization ID         = SMITH
  Application name         = db2bp
  Application handle       = 1
  Application status       = waiting for request
  Status change time      = 12-15-2001 10:29:06.707086
  Client node              = sys143
  Client release level     = SQL06010
  Client platform         = AIX
  Client protocol          = TCP/IP
  Client codepage         = 850
  Process ID of client application = 49074
  Client login ID         = smith
  Host application ID     = G9150F74.B6A5.991215152825
  Sequence number         = 0000
  Database alias at the gateway = MVSDB
  DCS database name       = DCSDB
  Host database name      = GILROY
  Host release level      = DSN05012
  Host CCSID              = 500

Outbound communication address = 9.21.21.92 5021
Outbound communication protocol = TCP/IP
Inbound communication address = 9.21.15.116 46756
```

```

First database connect timestamp      = 12-15-2001 10:28:24.596495
Host response time (sec.ms)          = 0.000000
Time spent on gateway processing     = 0.000000
Last reset timestamp                 =
Rows selected                        = 0
Number of SQL statements attempted   = 2
Failed statement operations          = 0
Commit statements                    = 1
Rollback statements                  = 0
Inbound bytes received               = 404
Outbound bytes sent                   = 140
Outbound bytes received               = 103
Inbound bytes sent                    = 287
Number of open cursors               = 0
Application idle time                 = 1 minute and 32 seconds

UOW completion status                =
Previous UOW completion timestamp    = 12-15-2001 10:28:25.592631
UOW start timestamp                  = 12-15-2001 10:29:06.142790
UOW stop timestamp                   =
Elapsed time of last completed uow (sec.ms)= 0.034396

Most recent operation                 = Execute Immediate
Most recent operation start timestamp = 12-15-2001 10:29:06.142790
Most recent operation stop timestamp  = 12-15-2001 10:29:06.707053

Statement                            = Execute Immediate
Section number                        = 203
Application creator                   = NULLID
Package name                          = SQLC2C07
SQL compiler cost estimate in timerons = 0
SQL compiler cardinality estimate     = 0
Statement start timestamp              = 12-15-2001 10:29:06.142790
Statement stop timestamp                = 12-15-2001 10:29:06.707053
Host response time (sec.ms)            = 1.101612
Elapsed time of last completed stmt(sec.ms)= 0.564263
Rows fetched                          = 0
Time spent on gateway processing       = 0.013367
Inbound bytes received for statement   = 220
Outbound bytes sent for statement      = 130
Outbound bytes received for statement  = 49
Inbound bytes sent for statement       = 27
SQL statement text:
create table t12 (col1 int, col2 char)

```

DCS アプリケーション状況

システム・モニターでは、以下のような 3 つの形式の LIST DCS APPLICATIONS コマンドを提供しています。

- LIST DCS APPLICATIONS
- LIST DCS APPLICATIONS SHOW DETAIL
- LIST DCS APPLICATIONS EXTENDED

以下に示す出力例では、IBM メインフレーム・データベースのバージョンおよび TCP/IP のサポート・レベルによって、Host Application ID と Client Application ID の形式が異なる場合があります。

表 11. ホストのバージョンと TCP/IP のサポート・レベルを基にしたアプリケーション ID の形式

シナリオ	アプリケーション ID 形式
RDB マネージャー・レベル 7 より前をサポートするデータ・サーバーにアクセスするクライアントの場合	G91A0D3A.P8BC.060306212019
TCP/IP v4 を使用して RDB マネージャー・レベルの 8 以降をサポートするデータ・サーバーにアクセスするクライアントの場合	9.26.13.61.65289.060306213816
TCP/IP v6 を使用して RDB マネージャー・レベルの 8 以降をサポートするデータ・サーバーにアクセスするクライアントの場合	2002:91a:519:13:209:6bff:fe14:4fbb.7684.060306213741

LIST DCS APPLICATIONS

アプリケーション・レベルでモニターが提供する情報を表示するには、DB2 LIST DCS APPLICATIONS コマンドを実行してください。

TCP/IP 接続について以下の情報が戻されます (DB2 Connect から DB2 for z/OS への接続)。

Auth Id	Application Name	Appl. Handle	Host Application Id
NEWTON	db2cli.exe	7	G91A0D3A.P8BC.060306212019
NEWTON	db2cli.exe	25	9.26.13.61.65289.060306213816
NEWTON	db2cli.exe	20	2002:91a:519:13:209:6bff:fe14:4fbb.7684.060306213741

許可 ID

IBM メインフレーム・データベース・サーバーへのログオンに使用された許可 ID。これは、だれがアプリケーションを実行しているかを識別します。

アプリケーション名

クライアントで実行中のアプリケーションの名前で、DB2 Connect に認識される。最後のパス分離文字の後の最初の 20 バイトが使用可能です。

Appl. Handle

DB2 Connect ワークステーション上で実行中のエージェント。このエレメントを使用して、データベース・システム・モニターの情報をその他の診断情報とリンクさせることができます。また FORCE USERS コマンドまたは API を使用する際にはエージェント ID も必要となります。

ホスト・アプリケーション ID

次のいずれかです。

- DRDA 関連トークン (CRRTKN) (無保護会話用)
- 作業単位 ID (UOWID) (DRDA-3 同期点マネージャーによって保護される 2 フェーズの接続用 (TCP/IP 接続で使用される))

この固有の ID は、アプリケーションが IBM メインフレーム・データベース・サーバーに接続するときに生成されます。このエレメントをアプリケーション ID と一緒に使用することにより、アプリケーション情報のクライアント部分とサーバー部分を関連させることができます。

LIST DCS APPLICATIONS SHOW DETAIL

DB2 LIST DCS APPLICATIONS SHOW DETAIL コマンド形式が指定される場合、以下の情報を含む追加情報が表示されます。

Auth Id	Application Name		Appl. Handle	Client Application Id
NEWTON	db2cli.exe		37	2002:91a:519:13:209:6bff:fe14:4fbb.8196.060306214224

Seq#	Client DB Alias	Client Node	Client Release	Client Codepage	Host Application Id
00001	MDB	SAYYID	SQL09000	1252	G91A0D3A.P982.060306214231

Seq#	Host DB Name	Host Release
00001	MEXICO	DSN08015

クライアント・アプリケーション ID

DB2 Connect ワークステーションに接続されたアプリケーションを固有に識別します。アプリケーション ID については各種の形式があり、それらはクライアントと DB2 Connect ワークステーションとの間の通信プロトコルによって異なります。

この値によって、クライアントから DB2 Connect ワークステーションへの接続、および DB2 Connect ワークステーションから IBM メインフレーム・データベース・サーバーへの接続を相互に関連させることができます。

クライアント順序番号 (Seq#)

クライアント順序番号は、トランザクションの順序番号です。クライアント順序番号は、異なるシステムでトランザクションを関連させるのに役立ちます。

クライアント DB 別名

このデータベースの別名は、そのデータベースに接続するアプリケーションにより与えられます。このエレメントは、アプリケーションがアクセスしている実際のデータベースを識別するために使用できます。この名前とデータベース名との間のマッピングは、クライアント・ノードおよびデータベース・マネージャー・サーバー・ノードにあるデータベース・ディレクトリーを使用して行うことができます。

クライアント NNAME (ノード)

これはクライアント・アプリケーションが実行しているノードを識別します。この情報は、使用中のクライアント・プロトコルによって変わります。TCP/IP 接続されたクライアントの場合、これはホスト名です。

クライアント製品 ID (クライアント)

クライアント上で実行している製品およびバージョン。クライアント製品 ID は以下のとおりになります。

- SQL07010 (DB2 Universal Database™ および DB2 Connect 製品バージョン 7.1 およびそれらのクライアント用)
- SQL08010 (DB2 Universal Database および DB2 Connect 製品バージョン 8.1 およびそれらのクライアント用)
- SQL08020 (DB2 Universal Database および DB2 Connect 製品バージョン 8.2 およびそれらのクライアント用)
- SQL09120 (DB2 バージョン 9.1 製品、DB2 Connect 製品、およびそれらのクライアント用)

コード・ページ ID

モニターされるアプリケーションが開始するノードでのコード・ページ ID。

この情報を使用して、アプリケーション・コード・ページとデータベース・コード・ページ (IBM メインフレーム・データベース・サーバー・データベースの場合は、IBM メインフレーム・データベース・サーバー CCSID) との間のデータ変換がサポートされているかどうかを確認することができます。

アプリケーション・コード・ページが、データベース・システム・モニター実行中のコード・ページとは異なる場合、このコード・ページ・エレメントは、アプリケーションから受け渡されてデータベース・システム・モニターによって表示されたデータを手動で変換するのに役立ちます。例えば、それを使用して、アプリケーション名の変換に役立てることができます。

アウトバウンド順序番号

これは、アウトバウンド順序番号を表しています。アウトバウンド順序番号は、異なるシステム上でトランザクションを相関させるために使用されます。

ホスト・データベース名

アプリケーションが接続されるデータベースの実名。DCS ディレクトリ内では、これはターゲット・データベース名です。

ホスト製品 ID

サーバー上で実行している製品およびバージョン。その形式は PPPVVRRM で、各文字は以下のとおりです。

PPP IBM メインフレーム・データベース・サーバー製品を識別します (例えば、DSN は DB2 for z/OS、ARI は DB2 Server for VSE & VM、QSQ は DB2 for IBM i など)。

VV 2 桁のバージョン番号です。例えば、08。

RR 2 桁のリリース番号です。例えば、01。

M 1 文字の修正レベル (0 から 9 または A から Z) を表します。

LIST DCS APPLICATIONS EXTENDED

LIST DCS APPLICATIONS コマンドにオプション EXTENDED を指定して使用すると、拡張レポートを生成することができます。拡張レポートには、コマンドで SHOW DETAIL オプションを指定したときにリストされるすべてのフィールドと、9 つの新規フィールドがリストされます。

- DCS アプリケーション状況
- 状況変更時刻
- クライアント・プラットフォーム
- クライアント・プロトコル
- ホスト・コード化文字セット ID (CCSID)
- クライアント・ログイン ID
- クライアント・アプリケーションのプロセス ID
- ゲートウェイでのデータベース別名
- DCS データベース名

既存のコマンド・オプションはアプリケーションごとに 1 行を使ってフィールドを横にリストしましたが、新しいオプションは 1 行に 1 フィールドの形式で縦にリストします。

新しいコマンド構文は次のとおりです。

```
LIST DCS APPLICATIONS [ SHOW DETAIL | EXTENDED ]
```

このコマンドで新しいオプション EXTENDED を指定したときの出力例を示します。

List of DCS Applications - Extended Report

```
Client application ID      = 2002:91a:519:13:209:6bff:fe14:4fbb.8196.060306214224
Sequence number           = 00001
Authorization ID          = NEWTON
Trusted Authorization ID  =
Application name          = db2cli.exe
Application handle        = 37
Application status        = waiting for request
Status change time       = Not Collected
Client node               = SAYYID
Client release level      = SQL09000
Client platform           = NT
Client protocol           = TCP/IP
Client codepage           = 1252
Process ID of client application = 1192
Client login ID           = ISAYYID
Host application ID       = G91A0D3A.P982.060306214231
Sequence number           = 00001
Database alias at the gateway = MDB
DCS database name        = MDB
Host database name        = MEXICO
Host release level        = DSN08015
Host CCSID                = 1208
```

アプリケーション状況フィールドには、次の 3 つの値のいずれかが含まれます。

1. connect pending - outbound。これは、IBM メインフレーム・データベースへの接続要求が発行され、DB2 Connect が接続の確立されるのを待っていることを意味しています。
2. waiting for request。これは、IBM メインフレーム・データベースへの接続が確立され、DB2 Connect がクライアント・アプリケーションからの SQL ステートメントを待っていることを意味しています。
3. waiting for reply。これは、SQL ステートメントが IBM メインフレーム・データベースに送信されたことを意味しています。

状況変更時刻は、処理中にシステム・モニターの UOW スイッチがオンに変更された場合にのみレポートに表示されます。それ以外の場合は、“Not Collected” が表示されます。

ヘルス・モニターおよびアラート

DB2 for z/OS ヘルス・モニターは、オブジェクトの保守ポリシーを定期的に評価します。オブジェクトの保守が必要であるとヘルス・モニターが判断すると、ヘルス・アラートが作成されます。ヘルス・アラートに対する応答アクションは、表示したり、実行用にサブミットしたり、保存したりすることができます。

DB2 for z/OS ヘルス・モニターの概要

z/OS システムでは、DB2 for z/OS ヘルス・モニターが、モニター対象の各 DB2 サブシステムのタスクとして、またはデータ共有グループの専用メンバー上で開始されます。

DB2 for z/OS ヘルス・モニターは、ポリシーの定義に従い、スケジュールされた時間と間隔でオブジェクト保守ポリシーの評価を起動します。オブジェクト保守ポリシーは、DB2 コントロール・センターの「オブジェクト保守ポリシーの作成」ウィザードを使用して作成されます。各ポリシーの評価中、保守の推奨基準はオブジェクト保守ポリシーのしきい値セットに対して検査され、オブジェクト保守が必要であるか（つまり、COPY、REORG、RUNSTATS、STOSPACE、ALTER TABLESPACE、または ALTER INDEX が必要であるか）を判別し、該当する場合には表スペース、索引、およびストレージ・グループ・オブジェクトの制限状態（CHKP など）を識別します。ポリシー評価の際にオブジェクトがアラート状態にあることが識別されると、ポリシー・ヘルス・アラート連絡先の E メール・アドレスまたはポケットベル番号に通知が行われます。各 DB2 サブシステムのヘルス・アラート連絡先のリストはコントロール・センターで定義および管理されます。

ポリシーの評価をいつ起動するかを判別するためにヘルス・モニターが使用する、ポリシーの評価スケジュールのスナップショットは、最初にヘルス・モニターによりその開始時に取られます。このスケジュール・スナップショットは、ヘルス・モニターの開始時に指定されたリフレッシュ時刻、またはヘルス・モニターがリフレッシュ・コマンドを受け取った時にリフレッシュされます。ポリシーの評価スケジュールの変更は、スケジュールのリフレッシュが発生した時にヘルス・モニターにより検出されます。

ヘルス・モニターはコンソールから開始および停止され、それぞれ MVS システムの START および STOP コマンドを使用します。

DB2 ヘルス・モニターを開始するカタログ式プロシージャのサンプル (DSNHMONP)、および MVS システムまたはParallel Sysplex®内で複数の DB2 ヘルス・モニターを開始するカタログ式プロシージャのサンプル (DSNHMONA) が、インストール・ジョブ DSNTIJHM によりプロシージャ・ライブラリーに置かれます。

DB2 ヘルス・モニターまたは以下にリストされている関連タスクにより使用されるビュー、表、データ・セット、カタログ式プロシージャ、ストアード・プロシージャ、ユーザー定義関数、および結果セット表は、インストール・ジョブ DSNTIJCC および DSNTIJHM により作成されインストールされます。DSNTIJCC および DSNTIJHM は、FMID JDB771D および JDB881D と共に配送されます。

ポリシー評価ログ

DB2 ヘルス・モニターにより起動されるポリシー評価は、表 DSNACC.HM_EVAL_LOG にログ記録されます。ポリシー評価の開始時およびポリシー評価の終了時に 1 項目がログ記録されます。ログ項目は 7 日間保持され、その後表から削除されます。DSNTIJCC インストール・ジョブによりこの表に作成された DB2 ビュー DSNACC.HM_ALERT_PO_EV は、最後の評価反復が成功しなかったすべてのポリシーを表示するために使用できます。

DB2 for z/OS ヘルス・モニターを開始、停止、およびリフレッシュする

z/OS システムでは、DB2 for z/OS ヘルス・モニターが、モニター対象の各 DB2 サブシステムのタスクとして、またはデータ共有グループの専用メンバー上で開始されます。

- DB2 ヘルス・モニターを開始するには、以下の START MVS システム・コマンドを発行します。

```
S membername, DB2SSN=ssid, JOBNAME=HMONssid, TRACE=trace, REFRESH=nn
```

TRACE および REFRESH パラメーターはオプションです。

membername

DB2 ヘルス・モニターを開始するために実行するプロシージャ・ライブラリー・メンバー (DSNHMONP) を指定します。このカタログ式プロシージャは、DSNTIJHM インストール・ジョブにより作成されます。

ssid

モニター対象の DB2 サブシステムの名前または ID を指定します。

トレース

トレース・フラグを指定します。可能な値は以下のとおりです。

- ON - トレースをオンにします。トレース・レコードは SYSOUT に書き込まれます。
- OFF - トレースをオンにしません。

デフォルトは OFF です。

nn

ヘルス・モニターがポリシー評価を起動するために使用する、評価スケジュール・スナップショットをリフレッシュする時刻 (24 時間クロックを使用) を指定します。デフォルトは 22 です。

- 複数の DB2 ヘルス・モニターを開始するには、以下の START MVS システム・コマンドを発行します。

```
S membername
```

membername

複数の DB2 ヘルス・モニターを開始するために実行するプロシージャ・ライブラリー・メンバー (DSNHMONA)。

注: DSNHMONA を使用して、複数の DB2 ヘルス・モニターを 1 つの START コマンドで開始する前に、DSNHMONA プロシージャに指定された HMONPARM データ・セットには、モニター対象のサブシステムのリストを取り込む必要があります。カタログ式プロシージャおよびデータ・セットは、DSNTIJHM インストール・ジョブで作成されます。

- DB2 ヘルス・モニターがポリシー評価を起動する時を判別するために使用する、ポリシー評価スケジュール・スナップショットをリフレッシュするには、以下の MODIFY MVS システム・コマンドを発行します。

```
F HMONssid,APPL=REFRESH
```

ssid

リフレッシュする DB2 ヘルス・モニターがモニターしている DB2 サブシステムの名前または ID。

- DB2 ヘルス・モニターを停止するには、以下の STOP MVS システム・コマンドを発行します。

```
STOP HMONssid または P HMONssid
```

ssid

停止する DB2 ヘルス・モニターがモニターしている DB2 サブシステムの名前または ID。

推奨処置を表示、サブミット、および保管する

ポリシー評価時に識別されるアラート・オブジェクトの推奨処置を表示、サブミット、および保管するには、DSNTIJCC インストール・ジョブにより作成される、DB2 ストアード・プロシージャ SYSPROC.DSNACCHR を呼び出します。DSNACCHR は、ポリシー評価時に識別されるアラート・オブジェクトの推奨処置を判別し、推奨処置を実行する JCL ジョブを生成するストアード・プロシージャです。

以下の構文図は、DSNACCHR を呼び出すための SQL CALL ステートメントを示しています。DSNACCHR のリンケージ規則が GENERAL WITH NULLS であるので、パラメーターをホスト変数で渡す場合、NULL 標識をそれぞれのホスト変数

と共に組み込む必要があります。入力ホスト変数用の NULL 標識は、CALL ステートメントを実行する前に初期化する必要があります。

構文

```
▶▶CALL DSNACCHR (—query-type,—health-ind,—policy-id,—work-set,—————▶
▶dataset-name,—member-name,—save-opt,—trace-flag,—————▶
▶NULL—————▶NULL—————▶NULL—————▶
▶job-id,—jobname,—jcl-proc-time,—trace-flag,—last-statement,—————▶
▶return-code,—error-msg )—————▶▶
```

query-type

ポリシー評価時にアラート状態にあると識別されたオブジェクトに対する推奨処置の実行内容を指定します。可能な値は以下のとおりです。

- 0 - アラート・オブジェクトに対する推奨処置を JCL ジョブとして表示する。
- 1 - アラート・オブジェクトに対する推奨処置を実行する JCL ジョブをサブミットする。
- 2 - アラート・オブジェクトに対する推奨処置を実行する JCL ジョブをサブミットし、ジョブを保留キューに入れる。
- 3 - アラート・オブジェクトに対する推奨処置をライブラリー・メンバー内の JCL ジョブとして保管する。

query-type は INTEGER タイプの入力パラメーターです。

health-ind

DSNACCHR が JCL ジョブに組み込むアラートのタイプを指定する。可能な値は以下のとおりです。

- RS - 制限状態
- EX - エクステント超過
- RR - REORG が必要
- CR - COPY が必要
- RT - RUNSTATS が必要
- SS - STOSPACE が必要

health-ind は VARCHAR(4) タイプの入力パラメーターです。

policy-id

オブジェクト保守ポリシーを指定します。 *policy-id* は VARCHAR(7) タイプの入力パラメーターです。

work-set

DSNACCHR が JCL ジョブに組み込むアラート・オブジェクトを識別する、オブジェクト保守ポリシーの作業セットを識別します。この作業セットは、*policy-id* およ

び *health-ind* パラメーターで指定されたアラートのポリシーおよびタイプで識別する必要があります。 *work-set* は INTEGER タイプの入力パラメーターです。

dataset-name

パーティション・データ・セット (PDS) または拡張パーティション・データ・セット (PDSE) の完全修飾名を指定します。この値は、*query-type* が 3 の場合に指定する必要があります。 *dataset-name* は VARCHAR(44) タイプの入力パラメーターです。

member-name

オブジェクト保守 JCL ジョブが保管される、*dataset-name* パラメーターで指定されたパーティション・データ・セット (PDS) または拡張パーティション・データ・セット (PDSE) を指定します。この値は *query-type* が 3 の場合に指定する必要があります。 *member-name* は VARCHAR(8) タイプの入力パラメーターです。

save-opt

オブジェクト保守 JCL ジョブを保管する方法を指定します。この値は *query-type* が 3 の場合に指定する必要があります。可能な値は以下のとおりです。

- R - 置換
- A - 追加
- NM - 新規メンバー

save-opt は VARCHAR(2) タイプの入力パラメーターです。

trace-flag

トレースをオンにするかまたはオフにするかを指定します。可能な値は以下のとおりです。

- Y - トレースをオンにする
- N - トレースをオフにする

trace-flag は CHAR(1) タイプの入力パラメーターです。

job-ID

query-type が 1 または 2 の場合は、サブミットされたジョブのジョブ ID を指定します。 *job-id* は VARCHAR(8) タイプの出力パラメーターです。

jobname

query-type が 1 または 2 の場合は、サブミットされたジョブの名前を指定します。 *jobname* は VARCHAR(8) タイプの出力パラメーターです。

jcl-proc-time

要求が処理された時刻を指定します。 *jcl-proc-time* は TIMESTAMP タイプの出力パラメーターです。

last-statement

DSNACCHR が重大エラーを戻す場合 (戻りコード 12)、このフィールドにはエラーの発生時に実行中であった SQL ステートメントが入ります。 *last-statement* は VARCHAR(2500) タイプの出力パラメーターです。

return-code

DSNACCHR 実行からの戻りコード。可能な値は以下のとおりです。

- 0 - DSNACCHR は正常に実行されました。
- 12 - DSNACCHR は重大エラーで終了しました。 *error-msg* パラメーターには、エラーを説明するメッセージが入ります。 *last-statement* パラメーターには、エラーの発生時に実行中であった SQL ステートメントが入ります。

return-code は INTEGER タイプの出力パラメーターです。

error-msg

DSNACCHR が重大エラーを戻す場合 (戻りコード 12)、このフィールドには定様式 SQLCA を含むエラー・メッセージが入ります。 *error-msg* は VARCHAR(1331) タイプの出力パラメーターです。

DSNACCHR は、*query-type* パラメーターが 0 の場合に結果セットを戻します。結果セットには DSNACCHR により生成された JCL ジョブが含まれます。

DSNACCHR 結果セット表は、DSNTIJCC インストール・ジョブにより作成されます。表 12 は、結果セットのフォーマットを示しています。

表 12. DSNACCHR 結果セットのフォーマット

列名	データ・タイプ	説明
JCLSEQNO	INTEGER	表の行のシーケンス番号 (1,...,n)
JCLSTMT	VARCHAR(80)	JCL ステートメントを指定します

ヘルス・アラート・サマリーを表示する

HEALTH_OVERVIEW 関数は、ヘルス・アラート・サマリーの VSAM KSDS データ・セットからの情報を、DB2 表として戻します。このデータ・セットは、DSNTIJHM インストール・ジョブにより作成されます。

ヘルス・アラート・サマリー・データ・セットには、DB2 ヘルス・モニターの状態、およびその MVS システムまたは並列シプレックス上のヘルス・モニターにより過去または現在モニターされているすべての DB2 サブシステムのアラート・サマリー統計に関する情報が含まれます。これらの情報は、各 DB2 サブシステムおよびアラート推奨の行によりクライアントに戻されます。

関数の結果は、以下の列を持つ DB2 表です。

ip-addr

DB2 サーバーの IP アドレス。これは VARCHAR(40) タイプの列です。

db2-ssid

DB2 サブシステムのサブシステム ID。これは VARCHAR(4) タイプの列です。

health-ind

アラートのタイプ。可能な値は以下のとおりです。

- RS - 制限状態
- EX - エクステント超過
- RR - REORG が必要
- CR - COPY が必要
- RT - RUNSTATS が必要
- SS - STOSPACE が必要
- PO - ポリシー評価に失敗
- HM - ヘルス・モニター状態

health-ind は VARCHAR(4) タイプの列です。

host-name

DB2 サーバーの完全修飾ドメイン・ネーム。これは VARCHAR(255) タイプの列です。

summary-stats

health-ind が 'HM' の場合は、DB2 ヘルス・モニターの状態。可能な値は以下のとおりです。

- 0: ヘルス・モニターは開始されていない
- 1: ヘルス・モニターは開始済み
- -1: ヘルス・モニターの状態は不明

それ以外の場合、*health-ind* で指定されたアラート・タイプのアラート・オブジェクトの合計数。これは INTEGER タイプの列です。

alert-state

health-ind で指定されたアラートの状態。可能な値は以下のとおりです。

- 5 - アラーム
- 4 - アテンション
- 3 - 警告
- 0 - 正常

alert-state は、*health-ind* が 'HM' の場合は必ず 0 です。これは INTEGER タイプの列です。

関数の外部プログラム名は HEALTH_OVERVIEW であり、固有名は DSNACC.DSNACCHO です。この関数は、DSNTIJCC インストール・ジョブにより作成されます。

例: DB2 サブシステム 'ABCD' の COPY を必要とするアラート・オブジェクトの合計数を検出します。

```
SELECT SUMMARYSTATS FROM TABLE (DSNACC.HEALTH_OVERVIEW()) AS T
WHERE DB2SSID = 'ABCD'
AND HEALTHIND = 'CR';
```

ヘルス・アラート・オブジェクトを表示する

反復ポリシー評価の最後の正常な実行の際に識別されたアラート・オブジェクトは、そのオブジェクト・タイプに応じて、それらのアラート・オブジェクト・リポジトリ表に保管されます。

アラート・オブジェクトは以下のとおりです。

- 表スペース用の DSNACC.HM_MAINT_TS
- 索引用の DSNACC.HM_MAINT_IX
- ストレージ・グループ用の DSNACC.HM_MAINT_SG

DB2 はそれらのアラート・オブジェクト・リポジトリ表上に多くのビューを作成します。ビューおよびアラート・オブジェクト・リポジトリ表は、DSNTIJC インストール・ジョブにより作成されます。表 13 は、各ビューが定義されている表およびビュー記述をリストしています。すべてのビュー名および表名には、修飾子 DSNACC があります。

表 13. ヘルス・アラート・オブジェクトのビュー

ビュー名	表	ビュー記述
HM_ALERT_TS_RS	HM_MAINT_TS	制限状態のすべての表スペースを表示する
HM_ALERT_TS_EX	HM_MAINT_TS	エクステントがユーザー指定の限度を超えているすべての表スペースを表示する
HM_ALERT_TS_RR	HM_MAINT_TS	REORG を必要とするすべての表スペースを表示する
HM_ALERT_TS_CR	HM_MAINT_TS	COPY を必要とするすべての表スペースを表示する
HM_ALERT_TS_RT	HM_MAINT_TS	RUNSTATS を必要とするすべての表スペースを表示する
HM_ALERT_IX_RS	HM_MAINT_IX	制限状態のすべての索引を表示する
HM_ALERT_IX_EX	HM_MAINT_IX	エクステントがユーザー指定の限度を超えているすべての索引を表示する
HM_ALERT_IX_RR	HM_MAINT_IX	REORG を必要とするすべての索引スペースを表示する
HM_ALERT_IX_CR	HM_MAINT_IX	COPY を必要とするすべての索引を表示する
HM_ALERT_IX_RT	HM_MAINT_IX	RUNSTATS を必要とするすべての索引を表示する
HM_ALERT_SG_SS	HM_MAINT_SG	STOSPACE を必要とするすべてのストレージ・グループを表示する

第 3 部 高可用性と DB2 Connect

DB2 Connect を使用する環境での高可用性については、固有の考慮事項があります。何らかの理由でネットワーク内のデータベース・サーバーが使用できなくなった場合に、クライアント・ワークステーションをネットワーク内の代替データベース・サーバーに転送する機能は重要になります。

第 11 章 ホスト・データベース接続のための高可用性とロード・ バランシング

今日の情報技術 (IT) 市場では、昼夜を問わずデータを使用可能にする需要が高まっています。ビジネスが競合相手と競争し、継続的成長を維持していくためにもこの要求を満たさなければなりません。今日の多くの Web、e-business、さらにスプレッドシート・アプリケーションは、エンタープライズ・データの利用を必要としています。IBM メインフレーム・データベースへの信頼性のある、高速で、セキュアな接続を確立しなければなりません。この接続は継続的に使用可能でなければならず、限界の負荷状態の下でも高度な接続要求を処理できなければなりません。どうしたらこうした接続を構築できるでしょうか。

高可用性のシナリオ

ある企業が Windows、Linux、および UNIX 上で稼働する複数のワークステーションとアプリケーション・サーバーを持っているとします。それらのマシンは、複数の IBM メインフレーム・データベースにあるデータにアクセスする必要があります。それらのマシンで稼働するアプリケーションでは、データベースへの高速で信頼の置ける接続が求められます。システム全体は、TCP/IP を使用するイーサネット・ネットワークで接続されています。

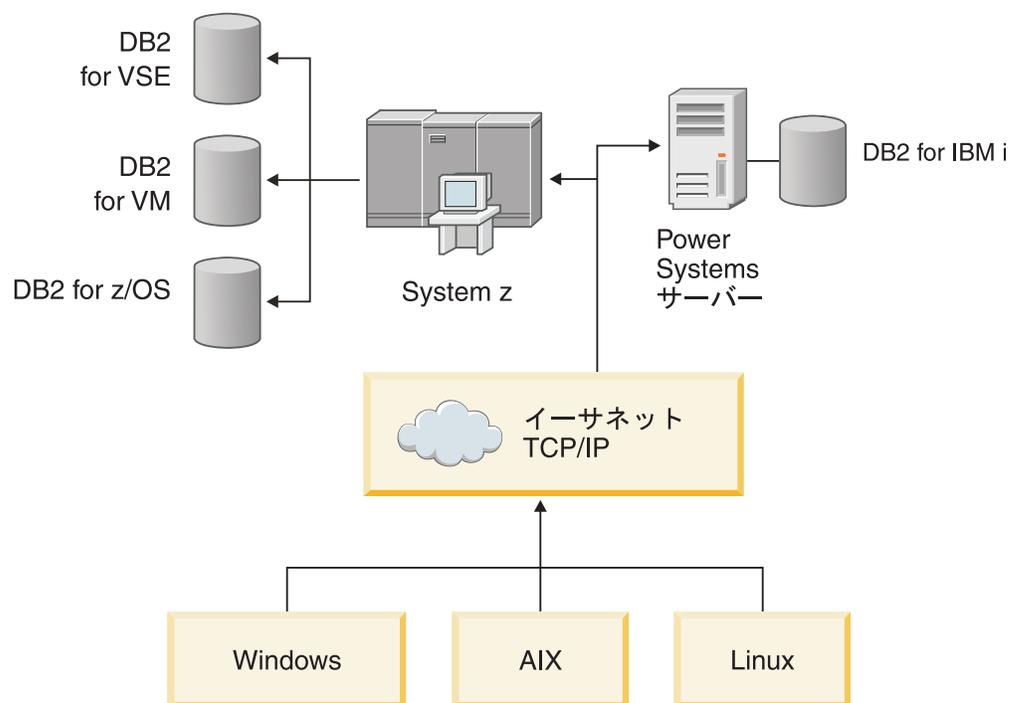


図 11. サンプル・ネットワーク・シナリオ

ワークステーションとアプリケーション・サーバーが、ホストと IBM メインフレーム・データベースを使用するためには、仲介として接続コンポーネントが必要です。このコンポーネントは IBM メインフレーム・データベースに、高可用性かつ堅

固な、高速接続を提供しなければなりません。それはさらに、接続量の将来の増加を見込んで、拡張が容易なものでなければなりません。

このトピックから関連したリンクを使用して、**DB2 Connect** と自動クライアント・リルート・フィーチャーを使用したソリューションに関する詳細情報を参照してください。

第 12 章 自動クライアント・リルートについての説明およびセットアップ (DB2 Connect)

自動クライアント・リルート・フィーチャーの主な目的は、IBM Data Server Client アプリケーションを通信障害からリカバリーし、中断を最小限に抑えてフィーチャーを続行できるようにすることです。その名前が示すように、連続稼働をサポートするうえでリルートは中心的な役割を果たします。ただしリルートは、クライアント接続によって識別される代替ロケーションが存在する場合にのみ可能です。非 DB2 Connect 高可用性環境でアクセスされるデータベースは通常、元の DB2 サーバーと代替 DB2 サーバーの間で同期されます。その際、高可用性クラスター・マルチプロセッサ (HACMP™) または高可用性災害時リカバリー (HADR) など、いくつかの手段の中の 1 つを使って同期が行われます。

ただし、DB2 Connect サーバーの場合にはローカル・データベースの同期に関する要件がないため、必要なのは、元の DB2 Connect サーバーと代替 DB2 Connect サーバーの両方のターゲット IBM メインフレーム・データベースを、同じデータベース別名を使ってアクセスできるようにカタログすることだけです。

注: DB2 Connect サーバー環境では、代替 DB2 Connect サーバーで、クライアントと DB2 Connect サーバーの間の自動リルートが有効となるように指定することができます。DB2 Connect Personal 製品またはサーバー製品と、IBM メインフレーム・データベース・サーバーとの間でリルートを行う場合、リモート・サーバーはそれ自体のために 1 つ以上の代替アドレスを提供する必要があります。DB2 for z/OS の場合、データベースが Sysplex データ共用環境である場合、複数のアドレスが認識されます。

Sysplex サポートが有効である場合、DB2 Connect とホスト・データベース・サーバーの間で Sysplex のリルート機能を構成できます。Sysplex のリルート機能は、Sysplex グループの元のメンバーとの通信が失われた後に DB2 Connect が他のメンバーへの接続を再試行するのを可能にする、DB2 Connect の機能です。DB2 Connect で Sysplex のリルート機能を有効にするために、データベース・ディレクトリーで代替サーバーをカタログする必要はありません。Sysplex サポートが有効である場合、Sysplex のリルート機能はデフォルトで有効になっています。

IBM Data Server Client で、DB2 Connect サーバーへの接続が失われたときに自動クライアント・リルート機能を使用して回復する機能を有効にするには、通信の損失が生じる前に代替 DB2 Connect サーバーの位置を指定する必要があります。UPDATE ALTERNATE SERVER FOR DATABASE コマンドを使用して、特定の IBM メインフレーム・データベースの代替 DB2 Connect サーバー・ロケーションを定義します。代替ホスト名およびポート番号は、コマンドの一部として指定します。このロケーションは、DB2 Connect サーバーのシステム・データベース・ディレクトリー・ファイルに保管されます。指定された代替 DB2 Connect サーバーの場所をすべてのクライアントのそのデータベースに適用させるためには、DB2 Connect サーバー・サイドで代替サーバーの場所を指定する必要があります。代替サーバーをクライアント・インスタンスで設定した場合、それは無視されます。

例えば、IBM メインフレーム・データベースのカタログにおいて、DB2 Connect サーバー S1 のデータベース別名 db1 (ホスト名 db2conn1、ポート番号 122) が使用されたとします。データベース管理者は、代替 DB2 Connect サーバー S2 のホスト名を db2conn2、ポート番号を 123 に指定することにします。データベース管理者は、DB2 Connect サーバー S1 で以下のようなコマンドを実行することができます。

```
db2 update alternate server for database db1 using hostname db2conn2 port 123
```

DB2 Connect サーバー S1 のデータベース別名 db1 の代替 DB2 Connect サーバー・ロケーションを指定した後、接続プロセスの一環として、代替サーバー・ロケーション情報が IBM Data Server クライアントに戻されます。何らかの理由 (通常は、SQL コード -30081 や SQL コード -1224 などの通信エラー) で IBM Data Server Client と DB2 Connect サーバー S1 の間の通信が失われると、IBM Data Server Client は元の DB2 Connect サーバー (S1) または代替 DB2 Connect サーバー (S2) を介し、これら 2 つのサーバーを切り替えながら、db1 への再接続を試みます。試行の間の時間間隔は最初は短く、試行を重ねるごとに徐々に長くなっていきます。

接続が成功すると、通信障害の後にデータベース接続が再確立されたことを示す SQL コード -30108 が戻されます。ホスト名または IP アドレス、およびサービス名またはポート番号が戻されます。IBM Data Server クライアント通信の再確立が元のサーバーに対しても代替サーバーに対しても不可能な場合、クライアントは元の通信障害に関するエラーだけをアプリケーションに戻します。

DB2 Connect サーバー環境での代替サーバー接続に関する以下の考慮事項にもご注意ください。

- リモート・クライアントとローカル・クライアントの両方のために DB2 Connect サーバーを使用してホストまたは IBM メインフレーム・データベースへのアクセスを提供する場合、システム・データベースのディレクトリー項目にある代替サーバーの接続情報に関して混乱が生じる可能性があります。この混乱を最小限に抑えるために、同じ IBM メインフレーム・データベースを表すように、システム・データベースのディレクトリーで 2 つの項目をカタログすることを検討してください。リモート・クライアント用に 1 つの項目、ローカル・クライアント用にもう 1 つの項目をカタログします。
- ターゲット DB2 for z/OS サーバーから戻される SYSPLEX 情報はすべて、DB2 Connect サーバーのキャッシュにのみ保持されます。ディスクには 1 つの代替サーバーだけが書き込まれます。代替サーバーまたはアクティブなサーバーが複数存在する場合、情報はメモリー内だけに維持され、プロセスが終了すると失われます。

第 13 章 クライアント接続ディストリビューター・テクノロジーに対する自動クライアント・リルートの構成

1 次データベース・サーバーで障害が発生した場合、ディストリビューターまたはディスパッチャー・テクノロジー (例えば WebSphere EdgeServer) はシステムの定義済みセットに対するクライアント・アプリケーション再接続要求の分散を行います。ディストリビューター・テクノロジーと DB2 自動クライアント・リルートを併用している場合、ディストリビューター自体を DB2 自動クライアント・リルート先の代替サーバーと見なす必要があります。

ディストリビューター・テクノロジーを次のような環境で使用していることがあります。

クライアント -> ディストリビューター・テクノロジー -> (DB2 Connect Server 1 または DB2 Connect Server 2) -> DB2 z/OS

ここで、

- ディストリビューター・テクノロジー・コンポーネントの TCP/IP ホスト名は **DThostname**
- DB2 Connect Server 1 の TCP/IP ホスト名は **GWYhostname1**
- DB2 Connect Server 2 の TCP/IP ホスト名は **GWYhostname2**
- DB2 z/OS サーバーの TCP/IP ホスト名は **zOShostname**

ディストリビューター・テクノロジーを使用していずれかの DB2 Connect Server にアクセスするために、クライアントは **DThostname** を使用してカタログされます。ディストリビューター・テクノロジーの介入により、**GWYhostname1** または **GWYhostname2** を使用することが決定します。決定後、クライアントはこれら 2 つの DB2 Connect ゲートウェイのうちのいずれかに直接ソケット接続します。選択された DB2 Connect サーバーへのソケット接続が確立されると、標準クライアント --> DB2 Connect サーバー --> DB2 z/OS の接続が成立します。

例えば、ディストリビューターが **GWYhostname2** を選択すると想定します。これにより、次の環境が生成されます。

クライアント -> DB2 Connect Server 2 -> DB2 z/OS

ディストリビューターは、何らかの通信障害があると、接続を再試行しません。このような環境においてデータベースでの自動クライアント・リルート・フィーチャーを使用可能にする場合、DB2 Connect サーバー (DB2 Connect Server 1 または DB2 Connect Server 2) 内の 1 つ以上の関連データベースの代替サーバーを、ディストリビューター (DThostname) としてセットアップする必要があります。DB2 Connect Server 1 が何らかの理由でロックされる場合、自動クライアント・リルートがトリガーされ、ディストリビューターを 1 次サーバーおよび代替サーバーの両方として使用してクライアント接続が再試行されます。このオプションを使用すると、ディストリビューター・フィーチャーと DB2 自動クライアント・リルート・フィーチャーを結合して保守できます。代替サーバーをディストリビューターのホ

スト名以外のホストに設定することによっても、クライアントに自動クライアント・リルート・フィーチャーが提供されます。ただし、クライアントは定義済み代替サーバーへの直接接続を確立して、ディストリビューター・テクノロジーをバイパスします。これによりディストリビューターとその価値が無効になります。

自動クライアント・リルート・フィーチャーは次の SQL コードをインターセプトします。

- sqlcode -20157
- sqlcode -1768 (理由コード = 7)

注: 「TCP キープアライブ」オペレーティング・システム構成パラメーターの設定値が高すぎる場合、クライアント・リルートはソケット障害に関して即時に知らされないことがあります。(この構成パラメーターの名前はプラットフォームによって異なることに注意してください。)

第 4 部 チューニングと DB2 Connect

DB2 Connect を使用してデータベース要求および応答をクライアント・ワークステーションとデータベース・サーバーとの間でやり取りするデータベース環境では、パフォーマンスの問題を考慮する際に検討する必要がある、固有の考慮事項があります。この環境でパフォーマンスを改善または維持するには、いくつかの方法があります。

第 14 章 DB2 Connect のパフォーマンスの考慮事項

パフォーマンスとは、与えられた特定のワークロードに対する、コンピューター・システムの振る舞いです。それは、利用可能なリソースと、それらがどのように使用され分配されるかによって影響を受けます。パフォーマンスを改善しようとする場合は、まず最初にパフォーマンスの具体的な意味を決定しなければなりません。

各種のパフォーマンス・メトリックを選ぶことができますが、その一部には次のものが含まれます。

応答時間

アプリケーションがデータベース要求を送信した時からアプリケーションが応答を受信する時までの時間間隔。

トランザクション・スループット

時間単位ごとに完了できる作業単位の数。作業単位は、行の取り出しや更新のように単純なものから、数百の SQL ステートメントが関係するような複雑なものまであります。

データ転送速度

時間単位ごとに DB2 Connect アプリケーションと IBM メインフレーム・データベースとの間で転送されるデータのバイト数。

パフォーマンスは、利用可能なハードウェアおよびソフトウェア・リソースによって限定されます。CPU、メモリー、およびネットワーク・アダプターは、ハードウェア・リソースの例です。AIX では、通信サブシステム、ページング・サブシステム、mbuf などがソフトウェア・リソースの例です。

データ・フロー

104 ページの図 12 は、IBM メインフレーム・データベース・サーバーとワークステーションの間を DB2 Connect を介して流れるデータの経路を示しています。

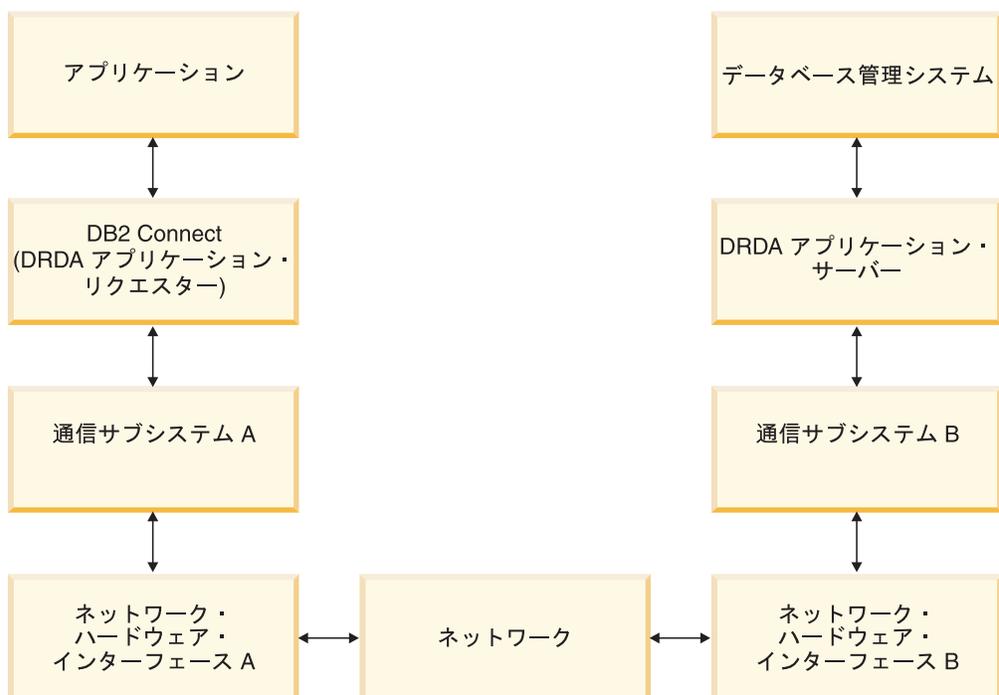


図 12. DB2 Connect 内のデータ・フロー

- IBM メインフレーム・データベースと、通信サブシステム B の部分は、通常同じシステム上で稼働します。このシステムは、1 つ以上の CPU、主記憶、入出力サブシステム、DASD、およびオペレーティング・システムで成り立っています。他のプログラムはこれらのコンポーネントを共用しているので、リソースの競合によってパフォーマンス問題が引き起こされる可能性があります。
- ネットワークは、ケーブル、ハブ、通信回線、スイッチ、その他の通信コントローラーの組み合わせで構成されています。例えば、ネットワーク・ハードウェア・インターフェース B は、3745 または 3172 あるいは IBM Power Systems サーバーのトークンリング・アダプターのような通信コントローラーになり得ます。ネットワーク・ハードウェア・インターフェース A と B の間には複数の伝送メディアが関係することもあります。
- ネットワーク・ハードウェア・インターフェース A は、トークンリング、イーサネット (Ethernet**), その他の LAN アダプター、または SDLC もしくは X.25 プロトコルをサポートするアダプターとすることができます。
- DB2 Connect と通信サブシステム A は、通常同じシステム上に配置されます。この説明の中では、アプリケーションもまた、同じシステム上にあることを想定しています。

ボトルネック

トランザクション・スループットは、システム内の最も遅いコンポーネントに依存します。パフォーマンスのボトルネックを識別できた場合は、構成パラメーターを変更したり、問題のコンポーネントにもっとリソースを割り振ったり、コンポーネントをアップグレードしたり、作業の一部の負荷を軽減するため新規のコンポーネントを追加したりして、多くの場合問題を軽減することができます。

各種のツールを利用して、各コンポーネントの照会時間がどれだけかかったかを判別することができます。このようにして、パフォーマンスを改善するにはどのコンポーネントを調整またはアップグレードしたらよいかについて構想が得られます。例えば、ある照会がその時間の 60% を DB2 Connect マシン内で使ったと判別した場合、おそらく DB2 Connect を調整するか、または (リモート・クライアントがある場合は) ネットワークにもう 1 つ別の DB2 Connect マシンを追加することができます。

ベンチマーク

ベンチマーク は、1 つの環境におけるパフォーマンスを別の環境におけるパフォーマンスと比較します。ベンチマークは、通常環境でテスト・アプリケーションを実行することから始めることができます。パフォーマンスの問題点が絞り込まれるにつれて、特殊化されたテスト・ケースを開発し、テストして観測する機能の範囲を限定していくことができます。

ベンチマークは、複雑にする必要はありません。特殊化されたテスト・ケースは、有効な情報を入手するのが目的なので、必ずしもアプリケーション全体をエミュレートする必要はありません。単純な計測から始めて、十分な理由がある場合だけ複雑性を増すようにします。

良いベンチマークの特性

- 各テストが反復できること。
- テストの各反復は同じシステム状態で開始されること。
- ベンチマーク用に使用されるハードウェアとソフトウェアは、ユーザーの実稼働環境と一致していること。
- 計測される機能またはアプリケーション以外には、活動状態にある他の機能またはアプリケーションがシステム内にないこと。他の活動がシステム内で実行中であることがシナリオに含まれている場合を除く。

注: 開始されたアプリケーションは、最小化またはアイドル時でもメモリーを使用します。このことは、ページングまたはベンチマーク結果のひずみの原因になり得ます。

パフォーマンス・ツール

次の表は、システム・パフォーマンスを計測するのに役立つツールのいくつかをリストしています。これらのツールは、それ自体がシステムのリソースを使用しますので、それらを常時活動状態にしておくことは望ましいことではありません。

表 14. CPU およびメモリーの使用量に関するパフォーマンス・ツール

システム	ツール	説明
AIX	vmstat、time、ps、tprof	DB2 Connect ワークステーションおよびリモート・クライアントでの CPU またはメモリーの競合問題についての情報を提供します。
HP-UX	vmstat、time、ps、monitor および glance (利用可能な場合)	

表 14. CPU およびメモリーの使用量に関するパフォーマンス・ツール (続き)

システム	ツール	説明
Windows	Microsoft パフォーマンス モニタ	

表 15. データベース・アクティビティに関するパフォーマンス・ツール

システム	ツール	説明
すべて	データベース・モニター	問題がデータベースから発しているかどうかを判別します。
System z	IBM Tivoli® OMEGAMON® XE for DB2 Performance Monitor on z/OS、ASG-TMON for DB2 (ASG)、および CA Insight Performance Monitor for DB2 for z/OS (Computer Associates International, Inc.)	
Windows	Microsoft パフォーマンス モニタ	

表 16. ネットワーク・アクティビティに関するパフォーマンス・ツール

システム	ツール	説明
AIX	netpmon	低水準のネットワーク統計を報告します。それには秒当たりに受信するパケットまたはフレームの数のような TCP/IP 統計が含まれます。
ネットワーク・コントローラ、例えば 3745	NetView® パフォーマンス・モニター	通信制御および VTAM® の使用効率を報告します。
Linux および UNIX	netstat	TCP/IP のトラフィックを取り扱います。

第 15 章 ODBC アクセスの最適化

DB2 データベースでは、ODBC を介した通信のパフォーマンスを改善するために設計された特別な最適化機能が準備されています。これらの拡張機能は Microsoft Access、Lotus Approach®、Visual Basic で使用することができます。DB2 の構成アシスタント (CA) を使用する際の ODBC スループットが高速になります。

最適化された ODBC を活動化するには、以下のようにします。

- 新しい接続を定義する場合:
 1. DB2 CA を開始します。
 2. 「選択済み」メニューを開き、「データベースの追加 (ウィザードを使用)...」を選択します。
 3. 「データ・ソース」ページが表示されるまで、ウィザードの各ページの指示に従います。
 4. 「CLI/ODBC 用にこのデータベースを登録」 をチェックします。
 5. このデータベースにアクセスする CLI/ODBC アプリケーションの登録方法を指定します。
 - 「システム・データ・ソースとして」はシステム上のすべてのユーザーに対して、データベースを使用できるようにすることを意味します。
 - 「ユーザー・データ・ソースとして」はデータベースにアクセスできるユーザーを自分だけに限定することを意味します。
 - 「ファイル・データ・ソースとして」はデータ・ソース情報を含むファイルが作成されることを意味します。TCP/IP 接続がある場合、このデータ・ソース・ファイルは他のワークステーションと共有することができます。接続がない場合はこのファイルを使用できるのはこのコンピューターに限定されます。
 6. 「データ・ソース名」を入力します。
 7. (オプション) 「最適化させるアプリケーション」リストからアプリケーションを選択し、特定のアプリケーションのデータ・ソース設定を最適化します。
 8. 「OK」をクリックして、CA を終了します。
- 既存の接続を更新する場合:
 1. DB2 CA を開始します。
 2. 最適化するデータベース別名をダブルクリックします。
 3. 「データ・ソース」をクリックします。
 4. 「CLI/ODBC 用にこのデータベースを登録」 をチェックします。
 5. このデータベースにアクセスする CLI/ODBC アプリケーションの登録方法を指定します。
 - 「システム・データ・ソースとして」はシステム上のすべてのユーザーに対して、データベースを使用できるようにすることを意味します。
 - 「ユーザー・データ・ソースとして」はデータベースにアクセスできるユーザーを自分だけに限定することを意味します。

- 「ファイル・データ・ソースとして」はデータ・ソース情報を含むファイルが作成されることを意味します。TCP/IP 接続がある場合、このデータ・ソース・ファイルは他のワークステーションと共有することができます。接続がない場合はこのファイルを使用できるのはこのコンピューターに限定されます。
6. 「データ・ソース名」を入力します。
 7. (オプション) 「最適化させるアプリケーション」リストからアプリケーションを選択し、特定のアプリケーションのデータ・ソース設定を最適化します。
 8. 「OK」をクリックして、CA を終了します。

第 16 章 アプリケーション設計

アプリケーションを作成するとき、いくつかの方法でパフォーマンスを改善することができます。

コンパウンド SQL およびストアード・プロシージャ

多くのコマンドと応答を受け渡しするアプリケーションでは、ネットワークのオーバーヘッドが重要になります。コンパウンド SQL とストアード・プロシージャは、このオーバーヘッドを軽減する 2 つの方策です。

1 つのアプリケーションがいくつかの SQL ステートメントをプログラミング・ロジックの介入なしに送信する場合、コンパウンド SQL を使用することができます。プログラミング・ロジックが SQL ステートメントのグループ内で必要な場合は、ストアード・プロシージャを使用することができます。

以下のものを除き実行可能なステートメントはすべて、コンパウンド SQL ステートメント内に含めることができます。

```
CALL  
FETCH  
CLOSE  
OPEN  
コンパウンド SQL  
Connect  
Prepare  
Release  
Describe  
Rollback  
Disconnect  
Set connection  
execute immediate
```

ストアード・プロシージャを使用すると、プログラム・ロジックがサーバーに入れられるので、ネットワーク・トラフィックを削減するのに役立ちます。プロシージャの終了時に自動的にコミットできます。また、結果セットを返すこともできます。こうするとクライアントのアプリケーション・ロジックを最小化できます。

要求のグループ化

関連する複数のデータベース要求 (SQL ステートメント) を 1 つのデータベース要求にグループ化すれば、ネットワークを通して伝送する要求と応答の数を減らすことができます。

例えば、以下のステートメントをグループ化して、

```
SELECT COL1, COL2, COL5, COL6 FROM TABLEA WHERE ROW_ID=1  
SELECT COL1, COL2, COL5, COL6 FROM TABLEA WHERE ROW_ID=2
```

次のようにすると、

```
SELECT COL1, COL2, COL5, COL6 FROM TABLEA WHERE ROW_ID=1 OR ROW_ID=2
```

ネットワークを通して送られる要求の数が減ります。

また、IN および BETWEEN のようなキーワードを使用することにより、返される行数を減らすことができます。さらに、UPDATE および DELETE ステートメントについて、WHERE、IN、および BETWEEN キーワードを使用することができます。

述部ロジック

必要な行および列だけを要求する場合に、述部ロジックを使用することができます。これは、データ伝送によるネットワーク・トラフィックおよび CPU オーバーヘッドを最小化します。

例えば、次の照会は使用しないようにします。

```
SELECT * FROM TABLEA
```

ROW_ID=1 を持つ TABLEA の 1 番目の行だけが実際に必要な場合や、1 番目と 2 番目の列だけがが必要な場合には、上の照会は使用しません。

データ・ブロッキング

サーバーからの大量のデータが予想される場合は、データ・ブロッキングを使用します。このブロッキングによって、ネットワーク帯域幅の使用は改善され、IBM メインフレーム・データベース・サーバーと DB2 Connect サーバーの両方の CPU オーバーヘッドが減少します。サイズに関係なく、送受信される各メッセージについて一定量の CPU とネットワークのオーバーヘッドがかかります。データ・ブロッキングは、同じ量のデータ転送に必要なとされるメッセージの数を減らします。

ブロッキングを使用すると、照会からのデータの 1 番目の行は、1 番目のブロックが受け取られるまではアプリケーションに送達されません。ブロッキングは、1 番目の行を探す検索時間を増加させますが、その後続く行については検索時間を短縮できます。

別の考慮事項は、使用されるメモリーの量です。メモリー作業セットは、ブロッキングがオンになると通常は増加します。

DB2 Connect 内では、各ブロック内で転送されるデータの量を制御することができます。

ブロッキングを呼び出すには、prep または bind コマンドの BLOCKING オプションを使用します。ブロッキングは、次の場合にオンになります。

- カーソルが読み取り専用になっているとき、または
- カーソルが未確定で、prep または bind コマンドの実行時にブロッキングが指定されているとき

注: 動的 SQL を使用している場合は、カーソルは常に未確定です。

BLOCKING を伴う SQL ステートメント

更新可能な SELECT ステートメント (UPDATE/DELETE WHERE CURRENT OF ステートメントを使用する) は、非ブロッキングの照会です。したがって、絶対に必要なときだけそれを使ってください。

更新可能な SELECT は、SELECT が完了した時と UPDATE/DELETE が発行される時との間にその行が決して変更されないようにします。このレベル

の並行性がアプリケーションにとって重要でない場合は、別の方法として、更新不可の SELECT から返される値に基づく検索条件を用いて、DELETE または UPDATE を使用します。

読み取り専用の SELECT については、FOR FETCH ONLY を指定します (ただし、VM および VSE ではサポートされていないため、指定できません)。

静的 SQL と動的 SQL

静的 SQL をできるだけ使用してください。それは、実行時 SQL セクション準備および未確定カーソルを回避します。動的 SQL の使用が避けられない場合は、ネットワーク・トラフィックを最小にしてパフォーマンスを改善するために、以下のことを行うことができます。

- ステートメントが SELECT であり、準備の必要がある場合は、PREPARE ... INTO SQLDA を実行します。SQLDA は、ユーザーの設定に必要なフルサイズに割り振ってください。列の最大数が x であり、それ以内にとどまると予想される場合は、SQLDA に x 個の SQLVAR を割り振ります。潜在的な列数が不確実な場合 (そしてメモリーが問題にならない場合) は、SQLVAR の最大数 (256) を使用します。

SQLDA の割り振りが、戻り SQLDA を保管するのに十分な大きさでない場合は、プログラムは、その結果を再度保管するのに十分な大きさの SQLDA を持つもう一つの DESCRIBE を発行する必要があります。このことは、ネットワーク・トラフィックを増加させます。

PREPARE そして DESCRIBE という順序を使用しないでください。PREPARE.....INTO ステートメントを使用すると、パフォーマンスは向上します。

- 静的にバインドされた SQL COMMIT または ROLLBACK ステートメントを、動的 COMMIT または ROLLBACK ステートメントの代わりに実行します。
- ステートメントが SELECT、COMMIT、または ROLLBACK ステートメントでない場合は、PREPARE および EXECUTE の順序の代わりに、EXECUTE IMMEDIATE を発行してステートメントを実行します。
- ODBC アプリケーションは動的 SQL を使用します。CLI/ODBC 静的プロファイル作成フィーチャーを使用すれば、パフォーマンスを改善することができます。このフィーチャーを使用すると、ODBC 呼び出しを取り込んで、データベース・パッケージに保管された静的ステートメントに変換することができます。実際に得られるパフォーマンスは、アプリケーションがどの程度複雑であるかによって決まります。

その他の SQL 考慮事項

コマンド行プロセッサ (CLP) を使用すると、一般に、プログラム内に動的 SQL を有する場合より動作が遅くなります。なぜなら CLP は、SQL をデータベース・エンジンへ発信する前に入力を構文解析する必要があるからです。また、CLP は、データを受け取った時にそれを形式化しますが、アプリケーションにとっては不必要なことです。

インタープリター言語 (例: REXX) による SQL ステートメントは、コンパイル言語 (例: C 言語) による同じ SQL ステートメントよりかなり処理が遅くなります。

CONNECT ステートメントについては、タイプ 1 およびタイプ 2 と呼ばれる 2 つのタイプがあります。タイプ 2 の接続を使用してデータベースへ接続した場合は、それまでの接続をドロップするのではなく、休止状態にします。その後で休止状態の接続に切り換えれば、ライブラリーのロードおよび内部データ構造のセットアップのオーバーヘッドを避けることができます。この理由から、タイプ 2 の接続を使用すれば、複数のデータベースにアクセスするアプリケーションについてはパフォーマンスを改善することができます。

第 17 章 接続の管理

接続の管理は、接続プールと接続コンセントレーターという 2 つの操作で構成されます。接続プールはデータベース接続のオーバーヘッドを削減し、接続ボリュームを管理します。接続コンセントレーターは、ホスト・データベース・サーバーが使用するリソースを最適化することにより、作業環境でのスケーラビリティを向上させます。ここでは、その両方の操作について説明します。

接続プール

DB2 Connect Enterprise Edition などのような DB2 Connect サーバー製品では、しばしば同時に行われる数千のクライアント要求に対するデータベース接続を提供します。データベース・サーバーへの接続の確立と切断は、リソースを集中的に使用するプロセスのため、データベース・サーバーと DB2 Connect サーバーの両方のパフォーマンスに悪影響を及ぼす場合があります。

この問題は、Web ページにアクセスするたびにデータベース・サーバーへの新規接続を行い、照会を実行してから接続を終了する必要がある Web 環境で顕著に見られます。このオーバーヘッドを減らすため、DB2 Connect サーバー製品は、接続プールを使用して、即座にアクセス可能なプールでデータベースへのオープン接続を維持します。

Web テクノロジーに基づくアプリケーションのほとんどは、短いトランザクションを大量に実行します。Web トランザクションは一般に、その独自の接続の一部として実行されます。つまり、トランザクションの実行とは、データベース接続を確立し、数行の SQL ステートメントを実行した後で、この接続を終了する一連の流れを意味します。接続を確立および切断するこの処理には多大なコストがかかります。これには、DB2 Connect エージェントの作成、このエージェントと DB2 サーバーとの間のネットワーク接続の確立、およびサーバー上での DB2 スレッドの作成が含まれます。接続を長く実行する場合にはこのコストはその接続で実行されるすべてのトランザクションによって償却されますが、一般的な Web トランザクションの場合は、トランザクションそのものを実行するコストがそのコストを超過してしまうことがよくあります。

接続プールは、確立された接続インフラストラクチャーを次の接続でも再び利用できるようにする技法です。DB2 Connect インスタンスの開始時に、コーディネーター・エージェントのプールが作成されます。接続要求が出されると、エージェントはその要求に割り当てられます。エージェントは DB2 サーバーに接続し、DB2 でスレッドが作成されます。アプリケーションが切断要求を出す際、エージェントはこの要求を DB2 サーバーに渡しません。その代わりに、エージェントはプールに戻されます。プール内のエージェントは DB2 サーバーと、対応する DB2 スレッドへの接続を所有します。別のアプリケーションが接続要求を出すと、このエージェントはその新しいアプリケーションに割り当てられます。安全な操作が行われるようにするために、ユーザー ID 情報が DB2 スレッドに渡され、次にこの情報がユーザー認証を実行します。

DB2 Connect の接続プールは、そのような環境でのパフォーマンスを大幅に向上させるものです。DB2 Connect は、データベースに対してオープンしている接続を使用可能なプールで保持します。クライアントが接続を要求すると、すでに使える状態になっている接続をこのプールから提供できます。接続プールによって、通常はそれらの接続をオープンしたりクローズしたりするのに費やされるオーバーヘッドが大幅に減少します。

接続プールの存在は、DB2 Connect を介してホストに接続するアプリケーションには影響を与えません。アプリケーションでホストからの切断が要求されると、DB2 Connect ではアプリケーションとのインバウンド接続は切断されますが、ホストとのアウトバウンド接続はプール内に維持されます。新しいアプリケーションが接続を要求すると、DB2 Connect は既存のプールからの接続を使用します。すでに存在している接続を使用すると、全体の接続時間が短縮されるだけでなく、ホストでの高い CPU 接続コストも削減されます。

DB2 Connect エージェントは、アイドルまたはアクティブの 2 つの状態のいずれかになっています。エージェントがアプリケーションの作業を実行している場合、そのエージェントはアクティブです。この作業が完了すると、エージェントはアイドル状態になり、同じアプリケーションまたは別のアプリケーションからの次の作業を待ちます。すべてのアイドル・エージェントは、アイドル・エージェント・プールとして知られている場所に一緒に保持されます。このプールのサイズは、**num_poolagents** 構成パラメーターを使用して構成できます。このパラメーターは、システムに維持させるアイドル・エージェントの最大数と同じです。このパラメーターをゼロに設定すると、接続プール・フィーチャーはオフになります。この構成パラメーターのデフォルトは、AUTOMATIC に設定され、値は 100 に設定されます。AUTOMATIC に設定されていると、DB2 Connect は、アイドル・エージェント・プール内のアイドル・エージェント数を自動的に管理します。

DB2 Connect は、最初のクライアント要求を受け取る前に、データベースへの接続を確立することはありません。あるいは、何らかのクライアントが要求を出す前にアイドル・エージェントのプールを満たすことができます。**num_initagents** 構成パラメーターを使用して、開始時にプールを満たすことができます。このパラメーターは、始動時に作成されるアイドル・エージェントの数を決定します。これらのアイドル・エージェントが、最初にホスト・データベース・サーバーに接続することはありません。

クライアントがホストへの接続を要求すると、DB2 Connect はホスト・データベース・サーバーに接続しているプールの中からエージェントを取得しようとします。それが失敗すると、アイドル・プールで使用可能なエージェントを検索します。プールが空の場合、DB2 Connect は新しいエージェントを作成します。

max_coordagents 構成パラメーターを使用して、同時にアクティブにできるエージェントの最大数を制御することができます。この数字を超えると、新しい接続はエラー SQL コード SQL1226 を発行して失敗します。(このコードは、同時に行うアウトバウンド接続の最大数を超過したことを意味します。) この構成パラメーターのデフォルトは、AUTOMATIC に設定され、値は 200 に設定されます。AUTOMATIC に設定されていると、DB2 Connect はコーディネーター・エージェントの数を自動的に管理します。

DB2 レジストリー変数 DB2CONNECT_IN_APP_PROCESS を使用すると、DB2 Connect サーバー製品と同じマシンで稼働しているアプリケーションが、アプリケーション・プロセス内で DB2 Connect を実行するか (デフォルトの動作)、アプリケーションが DB2 Connect サーバー製品に接続してからエージェント内でホスト接続を実行するか、いずれを行うかを指定できます。アプリケーションが接続プールを使用するには、DB2 Connect サーバー製品のエージェント内からホストに接続しなければならないので、DB2CONNECT_IN_APP_PROCESS を NO に設定する必要があります。

DB2 Connect の接続プールとアプリケーション・サーバーの接続プールの対比

接続プールは、多数のトランザクションをサポートするアプリケーションを基にした Web テクノロジーにとって必要不可欠なものです。現在ほとんどの Web アプリケーション・サーバーに、独自のデータベース接続のプール方式が備わっています。例えば、Microsoft MTS (COM+) と IBM WebSphere のどちらも、接続プールを提供しています。

これらのサーバーによってインプリメントされるアプリケーション・プール・メカニズムには、DB2 Connect サーバーによって提供されるものと大きな違いがあります。アプリケーション・サーバーは独自の使用だけのために接続をプールするため、一般に、ユーザー ID、パスワード、分離レベルなどはすべての接続で全く同じものとなるのが前提となります。さらに重要なのは、アプリケーション・サーバーがプールするのは同じプロセスで開始された接続だけである、という点です。つまり、別のマシン、ユーザー、またはプロセスからの接続はプールされません。アプリケーション・サーバーのこのようなプール技法は、アプリケーションの同じインスタンスによって確立された接続を再利用するには効果的ですが、複数のユーザー、サーバーなどからの接続をプールする場合にはまったく役に立ちません。

DB2 Connect サーバーが提供する接続プールは、アプリケーション、マシン、およびユーザーに全く依存しません。複数のクライアントからの接続、ユーザー ID がすべて異なるアプリケーション・サーバーからの接続は、お互いの接続をすべて再利用できるので、プールされたリソースをより有効に利用することができます。

どちらのタイプの接続プールを使用するのがよいのでしょうか？ それは、両方です。一般に、DB2 Connect の接続プールとアプリケーション・サーバーの接続プールの両方を使うのがよいと言えます。なぜなら、それらはお互いに連動していないからです。アプリケーション・サーバーの接続プールが使用可能になっているときでも、DB2 Connect の接続プールを使用して、複数のアプリケーション・サーバー、および DB2 Connect サーバーを使う他のクライアントは接続を再利用することができます。

接続コンセントレーター

接続コンセントレーターを利用すると、DB2 for z/OS データベース・サーバーが多数のワークステーションや Web ユーザーをサポートするのに必要なリソースを削減することができます。この機能により、DB2 for z/OS および DB2 Connect ソリューションのスケーラビリティが劇的に拡大し、さらに DB2 for z/OS のデータ共有環境において、障害の危険がないオペレーションとトランザクション・レベルのロード・バランシングを提供します。

接続コンセントレーターを使用すると、DB2 ホスト・サーバーでリソースを消費せずに複数のアプリケーションを接続したままにしておくことができます。アプリケーションでは何千人ものユーザーをアクティブに、そして DB2 ホスト・サーバーでは少数のスレッドのみをアクティブにすることができます。

DB2 Connect の接続コンセントレーター 技術を使用すると、DB2 Connect Enterprise Edition などの DB2 Connect サーバー製品は、同時にビジネス・トランザクションを実行する何千人ものユーザーをサポートすると共に、System z ホストまたは IBM Power Systems データベース・サーバーで必要とされるリソースを大幅に削減することができます。この技術は、すべてのアプリケーションからのワークロードを、かなり少ない数の System z ホスト接続または IBM Power Systems データベース・サーバー接続に集中させることにより、このことを成し遂げています。これは前述の接続プール機能とよく似ているように思われるかもしれませんが、実際には非常にボリュームの大きい OLTP (オンライン・トランザクション処理) アプリケーションのリソース使用量を減らすためのさらに洗練された方法です。

接続コンセントレーターではエージェントの概念が取り入れられ、エージェントはさらに 2 つのエンティティへと分割されています。

- 「論理エージェント」はアプリケーション接続を表します。
- 「コーディネーター・エージェント」は DB2 接続とスレッドを保持し、アプリケーションの要求を実行します。

新しいアプリケーションがホストに接続しようとする、その接続は論理エージェントに割り当てられます。データベースに SQL を渡すためにはコーディネーター・エージェントが必要であり、新しいトランザクションが開始されるとすぐにコーディネーター・エージェントが割り当てられます。このアーキテクチャーで重要なのは、コーディネーター・エージェントに次に示す性質があるという点です。

- 論理エージェントとの関連がなくなった
- コミットまたはロールバックによりトランザクションが完了すると、プールに戻される

もう 1 つの主な特色は、データ共用環境においてコーディネーター・エージェントを新規トランザクションに割り当てる方法です。DB2 Connect は System z ワークロード・マネージャー (WLM) の情報を使用する、洗練されたスケジューリング・アルゴリズムをインプリメントしています。この情報は WLM で設定された基準に従ってデータ共用グループのメンバー間でワークロードを分散するために使用されます。WLM は各メンバーの負荷だけでなく、各メンバーが利用可能であるかどうかも把握しています。このため、DB2 Connect は、障害が起きたメンバーまたは過負荷になっているメンバーから十分利用されていない稼働中のメンバーへの作業の透過的な再配置を行えます。DB2 Connect の接続コンセントレーターは、論理エージェントの最大数 (*max_connections*) をコーディネーター・エージェントの数 (*max_coordagents*) より大きい値に設定すると活動化されます。

接続プールは、アプリケーションが終了して接続が必要なくなるときに、接続を確立するのに必要なコストを節約します。言い換えると、プールした接続を別のアプリケーションが再使用するには、その前にアプリケーションが接続を切断する必要があります。

また、接続コンセントレーターを使用すると、DB2 Connect はアプリケーションがトランザクションを終了するとすぐ、別のアプリケーションで接続可能にすること

ができます。このとき、そのアプリケーションは接続を切断する必要はありません。本来、データベース・サーバー接続とそれに関連付けられたホストおよび DB2 Connect のリソースがアプリケーションで使用されるのは、アクティブなトランザクションがある場合だけです。トランザクションが完了するとすぐ、接続とそれに関連付けられているリソースは、トランザクションを実行する必要のある他のアプリケーションで使用できるようになります。

DB2 Connect の以前のバージョンでは、すべてのアクティブ・アプリケーションに、データベース接続だけでなくアプリケーション要求も管理するエンジン・ディスパッチ可能単位 (EDU) がありました。この EDU は通常、コーディネーター・エージェントと呼ばれていました。それぞれのコーディネーター・エージェントは、アプリケーションと EDU の状態またはコンテキストを追跡しました。各 EDU は、接続数の増加時に相当量のメモリーを必要とするため、エージェント間でのコンテキスト切り替えではさらにオーバーヘッドが増えてしまいます。

上記のアーキテクチャーでは、接続と EDU は 1 対 1 のリレーションシップにあります。しかし、接続コンセントレーターを使用すると、接続と EDU の関係を複数対 1 にすることができます。つまり、接続 (X) と EDU (Y) の関係は $X \geq Y$ になります。

接続コンセントレーターは、エージェントを 2 つのエンティティ (論理エージェントと作業エージェント) に分割します。論理エージェントはアプリケーションを表しますが、特定の EDU を参照することはありません。論理エージェントには、アプリケーションが必要とするすべての情報と制御ブロックが含まれています。n 個のアプリケーションがサーバーに接続している場合、そのサーバーには n 個の論理エージェントがあります。作業エージェントは、アプリケーションの要求を実行する物理 EDU ですが、指定されたアプリケーションへの永久接続は持ちません。作業エージェントは論理エージェントと連携して、トランザクションを実行します。さらに、その連携をトランザクション境界で終了し、使用可能なプールに戻ります。

ディスパッチャーとして知られるエンティティが、作業エージェントを論理エージェントに割り当てます。特定のコンピューティング・プラットフォームで開くことができるファイル・ハンドル数が制限されている場合、スケジューラー・インスタンスが複数になる場合があります。

接続コンセントレーターの制限

DB2 Connect サーバー・コンセントレーターの使用については、重要な制約事項がいくつかあります。システムで接続コンセントレーターの使用を試みる前に、以下の情報をすべて検討してください。

一般的な制限:

- コンセントレーターは、ローカル・クライアントからリモート・クライアントへのインバウンド接続を確立するに際し、TCP/IP プロトコルに依存します。TCP/IP またはローカル (IPC) を使用するインバウンド接続だけが、プールされたアウトバウンド接続を利用することができます。コンセントレーターは、Named PIPE などの他の通信プロトコルを経由した接続を受け入れますが、その接続で XA 集中フィーチャーを使用することはできません。

- XA と密接に結びついたトランザクション・サポートの場合、同じ XA トランザクションに参与するアプリケーションはすべて、同じ DB2 Connect サーバー・インスタンスを使用してホストに接続する必要があります。
- トランザクション境界で保留リソース (保留カーソルなど) を閉じるアプリケーションのみ、コンセントレーターの利点を享受することができます。保留カーソルを閉じないトランザクションは、依然として処理はされますが、専用の作業エージェントが割り当てられるため、コンセントレーターのフィーチャーをいずれも使用できません。
- 一時表を宣言する場合、その表はトランザクションまたは分岐境界で明示的にドロップする必要があります。表をドロップしないと、接続集中がオフになります。ただし、アプリケーションは処理を続行します。
- 同じ XA トランザクションに参与するすべてのアプリケーションに、同じ CCSID が必要です。また、同じユーザー ID を使用して接続する必要があります。
- 2 フェーズ接続をサポートするためにアウトバウンド接続が確立された場合、その接続のエージェントは 2 フェーズ接続をサポートするためにのみ使用することができます。同様に、1 フェーズ接続をサポートするために確立されたエージェントは、1 フェーズ接続だけをサポートします。
- コンセントレーターは、IBM Data Server Driver for JDBC and SQLJ を使用するアプリケーションと、動的 SQL を使用するコール・レベル・インターフェース (CLI) アプリケーションをサポートしています。コンセントレーターは各トランザクション境界で再度準備されるステートメントに依存するので、CLI アプリケーションは KEEP DYNAMIC も使用するべきではありません。
- 組み込み動的 SQL アプリケーションからの動的準備要求はリジェクトされません。静的 SQL を使用するか、動的 SQL ステートメント用の CLI を使用するよう、アプリケーションを変更する必要があります。
- 接続コンセントレーターがオンの場合、DB2 Connect サーバーへのインバウンド要求は SSL を使用できません。ただし、ターゲット・データベース・サーバーへのアウトバウンド要求では、SSL を使用できます。接続コンセントレーターがオフの場合、インバウンドおよびアウトバウンド要求の両方で SSL を使用することができます。

DB2 バージョン 9 またはバージョン 8 フィックスパック 13 (あるいはそれ以降) で作業している場合、DB2 Connect コンセントレーターのサポートを有効にするには IBM Power Systems バージョン 5 リリース 4 (PTF SI23726) が必要です。それがない場合には、接続コンセントレーターの XA 部分だけがサポートされます。

接続コンセントレーターの活動化

データベース・マネージャー構成パラメーター *max_coordagents* は、論理エージェントの最大数を設定します。 *max_connections* の値をデフォルトよりも大きい任意の値に設定することにより、コンセントレーター・フィーチャーをアクティブにすることができます。 *max_connections* のデフォルト値は、 *max_coordagents* の値と同じです。アプリケーションごとに 1 つの論理エージェントがあるため、 *max_connections* は実際にはデータベース・インスタンスに接続できるアプリケーションの数を制御し、 *max_coordagents* は同時にアクティブになれるインバウンド接

続の数を制御します。 *max_connections* は、 *max_coordagents* ~ 64,000 までの範囲の数値を取ります。デフォルトの論理エージェントの数は、 *max_coordagents* と同じです。

max_connections および *max_coordagents* を両方とも AUTOMATIC に設定することもできます。 *max_connections* が AUTOMATIC に設定された場合、接続の数は基本の構成値を超えて増加します。 *max_connections* と *max_coordagents* が両方とも AUTOMATIC に設定された場合、 *max_connections* は基本値を超えて増加でき、 *max_coordagents* は自動的に増加して、接続とコーディネーター・エージェントの間の集中率を維持します。

既存の構成パラメーターの中にも、エージェントを構成するために使われるものがあります。それには、以下のパラメーターが含まれます。

max_coordagents

アクティブなコーディネーター・エージェントの最大数。

num_poolagents

エージェント・プールのサイズ。エージェント・プールには、アクティブでないエージェントやアイドル状態のエージェントが含まれています。パフォーマンスを改善するために、 *num_poolagents* はクライアントの平均値と同じ値で構成します。

num_initagents

プール内の作業エージェントの初期数。これらはアイドル状態のエージェントです。

XA トランザクション・サポート

接続コンセントレーターのアーキテクチャーを使用すると、 DB2 Connect は DB2 for z/OS および DB2 for IBM i と密接に結びついた XA トランザクション・サポートを提供することができます。コンセントレーターは、他のすべてのトランザクションの場合と同じように、作業エージェントを特定の XA トランザクション (単一の XID) に関連付けます。しかし、XA トランザクションが *xa_end()* (分岐境界) によって終了する場合、作業エージェントが汎用プールに解放されることはありません。作業エージェントはその XA トランザクションに関連付けられたままです。別のアプリケーションが同じ XA トランザクションと結合すると、作業エージェントはそのアプリケーションに関連付けられます。

トランザクション境界を呼び出すと、エージェントはプールに戻されます。例えば、 *xa_prepare()* (読み取り専用)、 *xa_rollback()*、 *xa_recover()*、 *xa_forget()*、 *xa_commit()*、またはロールバックを引き起こすすべての XA エラーは、エージェントを通常のプールに戻します。 *xa_end()* が終わらせるのはトランザクションの分岐だけです。これは XID との関連付けを終わらせるには不十分です。

XA トランザクション・サポートの例

1. 4,000 以上の同時接続を必要とする環境について考えてみます。CGI アプリケーションを使用する Web サーバー、または多くのデスクトップ・ユーザーが存在するオフィス・システムでは、両方ともこの要件を超えてしまう可能性があります。このような場合、効率的な処理には、通常は DB2 Connect がスタンドアロ

ン・ゲートウェイとして動作することが求められます。すなわち、データベースと DB2 Connect を別々のマシンに置く必要があります。

DB2 Connect サーバー・システムは、データベース・マシンに対する 4,000 もの同時に行われるオープン接続を維持できない場合があります。たいていの場合、特定の瞬間に生じるトランザクション数は、同時接続の数よりもかなり小さくなります。そのため、システム管理者は、データベース構成パラメーターを以下のように設定することにより、システムの効率を最大にすることができます。

```
MAX_CONNECTIONS = 4,000
MAX_COORDAGENTS = 1,000
NUM_POOLAGENTS   = 1,000
```

ゲートウェイが同時に処理しているトランザクション数が 1,000 しかない場合でも、コンセントレーターは最大 4,000 の並行セッションをオープンし続けます。

2. 上記の例では、作業エージェントと論理エージェントの関連付けは、常に形成されたり解除されたりしています。アイドル状態でないそれらのエージェントは、データベースへの接続は維持していますが、特定のトランザクションに関与してはいません。そのため、接続を要求する任意の論理エージェント（アプリケーション）で 사용할ことができます。

XA トランザクションの場合は、いくらか異なっています。この例では、DB2 Connect ゲートウェイと System z または IBM Power Systems データベースで TP モニターを使用していることを想定しています。アプリケーションが接続を要求すると、コンセントレーターは、その要求に応じるためアクティブでないエージェントをアクティブにするか、新しい作業エージェントを作成します。アプリケーションが XA トランザクションを要求するものとします。このトランザクションに応じて XID が作成され、作業エージェントがそれに関連付けられます。

アプリケーションの要求が処理されると、`xa_end()` が発行され、作業エージェントからデタッチされます。作業エージェントと、トランザクションの XID との関連付けは残ります。このとき、このエージェントは、関連付けられている XID を持つトランザクション要求にのみ応じることができます。

この時点で、別のアプリケーションが非 XA トランザクションを要求する場合があります。他に使用可能な作業エージェントがない場合でも、XID に関連付けられたエージェントは 2 番目のアプリケーションで使用可能にされることはありません。それは、アクティブであると見なされます。2 番目のアプリケーション用には、新しい作業エージェントが作成されます。2 番目のアプリケーションがトランザクションを終了すると、そのアプリケーションの作業エージェントは使用可能なプールに解放されます。

一方、最初のエージェントの XID に関連付けられたトランザクションを要求している他のアプリケーションが、そのエージェントにアタッチしたりエージェントからデタッチされたりする場合は、そのエージェント専用の XA トランザクションが実行されます。そのトランザクションを要求するアプリケーションはすべて、この作業エージェント（空き状態であれば）に送信されます。

作業エージェントは、アプリケーションがトランザクション境界呼び出し (`xa_end()` ではない) を発行するまでは、汎用プールに解放されることはありません。

せん。例えば、アプリケーションが `xa_commit()` でトランザクションを終了する場合、その時点で作業エージェントは `XID` との関連付けを解除し、使用可能なプールに戻ります。この時点で、要求元のアプリケーションはすべて、別の `XA` トランザクションか非 `XA` トランザクションのいずれかの作業エージェントを使用することができます。

接続プールと接続コンセントレーター

接続プールと接続コンセントレーターは類似性があるように見えますが、インプリメンテーションが異なり、別の問題を処理します。接続プールはデータベース接続のオーバーヘッドを削減し、接続ボリュームを処理するのに役立ちます。接続コンセントレーターは、ホスト・データベース・サーバーの使用を最適化することにより、`DB2 for z/OS` および `DB2 Connect` ソリューションの拡張容易性を高めるのに役立ちます。

接続プールを使用している時には、接続を所有しているアプリケーションが切断要求を出してからでなければ、その接続を再利用することはできません。多くの 2 層のクライアント/サーバー・アプリケーションでは、ユーザーが就業時間中に切断することはありません。同様に、複数層アプリケーションにおいて、ほとんどのアプリケーション・サーバーはデータベース接続をサーバー始動時刻に確立し、アプリケーション・サーバーがシャットダウンするまでそれらの接続を解放しません。

そうした環境では、接続プールの利点はあったとしてもわずかでしょう。しかしながら、接続と切断の頻度がより高い、`Web` 環境およびクライアント/サーバー環境においては、接続プールによってパフォーマンスはかなり向上するでしょう。接続コンセントレーターは、ユーザー・アプリケーションをアクティブにしたまま、`SQL` トランザクションの継続時間のみ、ホスト・データベース・リソースを割り振ります。そうすることにより、すべてのアプリケーション接続がそれぞれのスレッドを持つよりも、`DB2` スレッドの数と消費するリソースがずっと小さくなるように構成できます。

障害の危険がないオペレーション、およびワークロードのロード・バランシングについて言うなら、新規のトランザクションごとに作業の再割り振りができる接続コンセントレーターは明らかに正しい選択です。代わりに、接続プールは限られた平衡化を接続時間中に行うことしかできません。

接続プールと接続コンセントレーターは、異なった問題を処理しますが、一緒に使用するべきです。

WebSphere MQ トランザクション・マネージャーおよび `DB2 for z/OS` で必要な接続コンセントレーター

`IBM WebSphere MQ` (以前は `IBM MQSeries`[®] と呼ばれた) 環境でアプリケーションを実行しているとき、`WebSphere MQ` は `XA` 準拠のトランザクション・マネージャーとなって、分散した 2 フェーズ・コミット・トランザクションを調整できます。`WebSphere MQ` がこの方法でトランザクション・マネージャーとして動作するとき、データ・ソースが `DB2` ファミリー製品からのものである場合は、いくつかの構成要件があります。

そのようなトランザクション・マネージャー環境での構成要件のほとんどは、既にどこかで文書化されています。例えば、DB2 ランタイム・クライアントで DB2 構成パラメーター TP_MON_NAME を 『MQ』 に設定する必要があります。

ただし、欠落していた構成要件もあります。DB2 for z/OS サーバー用のデータ・ソースに接続している場合、要件は DB2 Connect に固有のものとなります。WebSphere MQ を使用して DB2 for z/OS および DB2 for IBM i サーバーに関与する分散トランザクションを調整している場合は、DB2 Connect 接続コンцентрーター・フィーチャーをゲートウェイで有効にする必要があります。接続コンцентрーターは、**max_connections** 構成パラメーターの値が **max_coordagents** 構成パラメーターの値より大きい場合に有効になります。

接続コンцентрーターを有効にしない場合、結果として予期しないトランザクション動作が発生します。

第 18 章 DB2 Connect サーバー Sysplex のサポート

Sysplex とは、ハードウェアとソフトウェアを使って共同で作業する System z サーバーの集合のことです。Sysplex は、共同で作業するプロセッサ数を増やして、処理可能な作業量を増やすことによって、共同作業を調整します。処理能力の強化に加えて Sysplex では、各種レベルのハードウェアやソフトウェアを混合したり、システムを動的に追加したりするための柔軟性も実現されています。

Sysplex によって、DB2 Connect サーバーはデータ共有グループのさまざまなメンバー間でシームレスに接続のバランスを取ることが可能になります。さらに、Sysplex によって、DB2 Connect サーバーは障害が発生しているメンバーから別のメンバーへの切り替えを試行できます。この Sysplex のリルート機能は、DB2 Connect のフィーチャーの 1 つです。DB2 Connect サーバーの Sysplex サポートおよび Sysplex 用のリルート機能は、デフォルトで使用可能になっています。ホスト・データベースに対する Sysplex サポートは、SYSPLEX パラメーターをその DCS ディレクトリー項目から除去することによってオフにすることができます。ただし、DCS 項目そのものは (たとえそれに他のパラメーターが指定されていなくても) 除去しないでください。

Sysplex の自動クライアント・リルート機能のデフォルトの動作は、Sysplex を有効にした接続で通信障害が発生した場合に、再度接続し直すためのものです。DB2 Connect が DB2 for z/OS サーバーに接続するときに、最後に成功したリソースを保持しないトランザクションの時点までの特殊レジスター値が再生されます。

DB2_MAX_CLIENT_CONNRETRIES および **DB2_CONNRETRIES_INTERVAL** のレジストリー変数を使用して、自動クライアント・リルートを再試行するときの動作を厳密に構成することが可能です。使用不可に設定することもできます。この接続タイムアウト・レジストリー変数は **DB2TCP_CLIENT_CONTIMEOUT** です。

System z SYSPLEX を活用する際の考慮事項

DB2 Connect は、複数の Sysplex への接続を経路指定する際のロード・バランシングおよびフォールト・トレランスの機能を提供します。データ共有環境内で稼働する DB2 for z/OS データベース・サーバーに接続すると、DB2 Connect は、Workload Manager (WLM) から提示されたシステム負荷情報に基づいて、データ共有グループを構成する様々な DB2 サブシステムにワークロードを分散します。

DB2 Connect は、Sysplex メンバーを優先順位付けしたリストを WLM から受信します。各 Sysplex は、各接続アドレスごとに重みを付けた優先順位情報を戻します。次にこのリストは DB2 Connect で使われて、着信の接続要求が処理されますが、その処理では、最高の優先順位を割り振られた Sysplex メンバーにその要求が配布されます。優先順位情報の重みを付けられた Sysplex リストが、ロード・バランシング用に各接続中に獲得されます。DB2 Connect の接続コンセントレーターが使用可能になっていれば、そのリストは、各トランザクションの送付先の決定の際にも使用されます。

注: DB2 Connect の Sysplex の機能を活用するのに、System z 分散データ機能 (DDF) の構成を変更する必要はありません。

接続障害が起きた場合は、DB2 Connect には、代わりの Sysplex マシンへの接続を試みるというフォールト・トレランス機能が備わっています。認識される接続がすべて失敗した場合にのみ、アプリケーションにエラーが戻されます。

DB2 Connect Sysplex はエージェント・プールを念頭に設計されました。Sysplex が使用可能であり、参加中のメンバーへの接続が失われると、DB2 Connect は別の DDF メンバーへの接続を経路指定します。転送は Sysplex サーバーのリストに応じて実行されます。

コンセントレーターを追加すると、DB2 Connect は、トランザクション境界でワークロードの均等化を図ることができます。そのためには、DB2 Connect コンセントレーターを使用可能にする必要があります。

DB2 Sysplex の活用

典型的なシナリオとして、DB2 Connect サーバー (サーバー A) は、2 つの DB2 for z/OS サーバー (サーバー B および C) から成る Sysplex と通信するとします。

Sysplex サーバー B	Sysplex サーバー C
HOST_NAME=MVSHOST	HOST_NAME=MVSHOST1

このシナリオで、アプリケーションによって、次のようなコマンドが使われたとします。

```
db2 connect to aliasb user xxxxxxxx using xxxxxxxx
```

データベース MVSHOST への接続が確立されます。DB2 Connect サーバーと DCS ディレクトリー項目のどちらでも Sysplex を使用できるので、DB2 for z/OS は各 Sysplex 参加者 (MVSHOST および MVSHOST1) のネットワーク・アドレスを DB2 Connect に対して識別します。この情報を戻すのに DRDA4 プロトコルおよびメッセージ・フローが使用されます。最初の接続が確立されると、戻されるアドレス・リストは DB2 Connect ワークステーションのキャッシュに入れられます。初期 CONNECT が TCP/IP ノードに対して発行されると、IP アドレスが戻されます。

ロード・バランシングとフォールト・トレランスに使用される優先順位情報

DB2 for z/OS によって提供されるアドレス・リストには、各ネットワーク・アドレスの接続数を含め、優先順位情報が含まれています。リストは、DB2 Connect によって新しい接続が確立される度に更新されます。この追加情報はロード・バランシング、およびフォールト・トレランスのために使用されます。

DB2 Connect で使用するキャッシュ・アドレス・リスト

ALIASB へのデータベース接続が失敗すると、SQL30081N エラー・メッセージが出されて、接続はドロップされます。ALIASB への接続要求がさらに受け取られると、DB2 Connect は次のことをします。

1. DB2 for z/OS から戻された優先順位情報に基づいて、キャッシュに入れられたアドレス・リスト中の最高優先順位のサーバーを試します。このストラテジーは DB2 Connect によって必ず使用されます。これによって、ロード・บาลancing が達成されます。
2. この接続の試みが失敗すると、DB2 for z/OS によって戻されたリストで、優先順位の高いものから順に他のアドレスが試されます。このように、DB2 Connect は Sysplex 情報を活用してフォールト・トレランスを図ります。
3. 他のすべての接続の試みが失敗すると、DB2 Connect はカタログされたノード・ディレクトリーに入っているアドレスを使用して、ALIASB への接続を再度試みます。

sysplex パラメーターを指定した db2pd コマンド (db2pd -sysplex) は、Sysplex 環境に関連付けられたサーバーに関する情報を取得するために使用できます。

Sysplex の構成の要件

データベースの DCS ディレクトリー項目で、Sysplex (大文字小文字の区別なし) が 6 番目の定位置パラメーターに含まれていないと、データベースによっては Sysplex を活用することはできません。

第 19 章 クライアント Sysplex サポート

DB2 Connect ライセンスを持つ IBM データ・サーバー・クライアントおよび非 Java データ・サーバー・ドライバーは、DB2 for z/OS Sysplex にアクセスして、トランザクション・レベルのロード・バランシング、自動クライアント・リルート、および制限付き XA サポートを含む、Sysplex の全機能を使用することができるようになりました。

Sysplex は、データ共有グループを形成する DB2 システム (メンバーと呼ばれる) の集合です。1 つ以上のカップリング・ファシリティーによって、データ共有グループに対する高速キャッシングおよびロック処理が提供されます。Sysplex をワークロード・マネージャー (WLM)、動的仮想 IP アドレス (DVIPA)、および Sysplex Distributor とともに使用することによって、クライアントは、TCP/IP を介してネットワーク回復力のある方法で DB2 for z/OS データベースにアクセスしたり、データ共有グループ内のメンバー間でアプリケーションのトランザクションをバランスの取れた方法で分散したりできます。

これらの機能の中心となるのは、DB2 データ共有グループの各メンバーが接続境界およびトランザクション境界 (オプション) で戻すサーバー・リストです。このリストには、各 DB2 メンバーの IP アドレスおよび使用可能な容量が含まれます。この情報を使用して、クライアントは、トランザクションをバランスの取れた方法で分散したり、通信障害が生じた場合に使用する DB2 メンバーを識別したりすることができます。

サーバー・リストは、初めて正常に DB2 データベースに接続した時点で戻されます。このため、最初のデータベース接続は Sysplex Distributor が所有するグループ DVIPA に送信されなければなりません。少なくとも 1 つの DB2 メンバーが使用可能である場合、Sysplex Distributor はその要求をデータベースに経路指定します。サーバー・リストを受け取ると、クライアントは、サーバー・リスト内の情報に基づいて DB2 メンバーに直接アクセスします。

クライアント Sysplex サポートを構成するには、db2dsdriver 構成ファイルで設定値を指定します。

WLM、DVIPA、および Sysplex Distributor については詳しくは、DB2 for z/OS インフォメーション・センター (<http://publib.boulder.ibm.com/infocenter/dzichelp>) を参照してください。

トランザクション・レベルのワークロード・バランシング (クライアント・サイド)

DB2 Connect ライセンスを持つ IBM データ・サーバー・クライアントおよび非 Java データ・サーバー・ドライバーは、DB2 Connect サーバーを介さずに、トランザクション・レベルでワークロード・バランシングを実行できます。トランザクション・レベルのワークロード・バランシングを使用して、DB2 for z/OS Sysplex のメンバーのワークロードをトランザクションの開始時にリバランスすることができます。

ワークロード・バランシングは、アプリケーションごとに単一または複数の接続の並行トランザクションで実行されます。長時間実行される単一の接続のみを呼び出すアプリケーションで、トランザクションが同時にではなく連続して実行される場合、ワークロードは最大容量を持つメンバーにシフトされます。ワークロード・バランシング・アルゴリズムは、ワークロード管理のウェイトの変更を自動的に調整します。

以下は、クライアントが DB2 for z/OS Sysplex に接続し、トランスポートの再利用によるワークロード・バランシングが発生したときに行われる手順の概要を示しています。

1. クライアントが接続を確立すると、サーバーはメンバーのワークロード分散情報が含まれるサーバー・リストを戻します。

サーバー・リストはクライアントによってキャッシュされます。この存続期間は構成可能で、デフォルトは 30 秒です。

2. 新規のトランザクションの開始時に、クライアントはサーバー・リストを読み取って、容量に余裕のあるメンバーを識別し、容量に余裕のあるそのメンバーに関連付けられたアイドル・トランスポートがないかどうかトランスポート・プールを調べます。(アイドル・トランスポートとは、論理接続が関連付けられていないトランスポートのことです。)
 - アイドル・トランスポートが使用可能な場合、クライアントは論理接続をそのトランスポートと関連付けます。
 - ユーザー構成可能なタイムアウトが経過した後、使用可能なアイドル・トランスポートがトランスポート・プールに存在せず、トランスポート・プールがその制限に達しているために新規のトランスポートを割り振ることもできない場合、アプリケーションにエラーが戻されます。
3. トランザクションが実行されると、トランザクションはトランスポートに関連付けられたメンバーにアクセスします。
4. トランザクションが終了すると、クライアントは、トランスポートの再利用が論理接続で引き続き許可されているかどうかをサーバーに確認します。
5. トランスポートの再利用が許可されている場合、サーバーは、論理接続の実行環境に適用される特殊レジスターに関する SET ステートメントのリストを戻します。

クライアントはこれらのステートメントをキャッシュに入れておき、論理接続が新規のトランスポートと関連付けられたときにそれを再生して、実行環境を再構成します。

6. トランスポートの再利用が許可されている場合、論理接続はトランスポートから周期的に関連付けを解除され、サーバー・リストのクライアント・コピーはリフレッシュされます。
7. 新規トランザクションでワークロード・バランシングが必要な場合、クライアントは同じプロセスを使用して論理接続をトランスポートと関連付けます。

ヒント: クライアント・サイドのトランザクション・レベルのワークロード・バランシングは、単一アプリケーション・プロセスで実行されるトランザクションで行われます。クライアントで実行されている複数のアプリケーション・プロセスが同じ Sysplex 対応データベースにアクセスする場合、ワークロード・バランシング

は、他のプロセスとは独立してアプリケーション・プロセスごとに実行されます。複数のリモート・クライアントで実行されている複数のアプリケーションまたはプロセスからの複数の接続で、トランザクション・レベルのワークロード・バランシングを実行する場合は、DB2 Connect サーバーで利用可能なワークロード・バランシング機能を使用してください。

トランザクション・レベルのワークロード・バランシングの構成 (クライアント・サイド)

クライアントでトランザクション・レベルのワークロード・バランシングを構成するには、db2dsdriver 構成ファイルの設定値を指定します。

始める前に

トランザクション・レベルのワークロード・バランシングを実行するには、クライアントは TCP/IP 接続を使用しなければならず、DB2 Connect ライセンスを持っている必要があります。以下のクライアントは、トランザクション・レベルのワークロード・バランシングをサポートします。

- IBM Data Server Client
- IBM Data Server Runtime Client
- IBM Data Server Driver Package
- IBM Data Server Driver for ODBC and CLI

このタスクについて

このタスクは、トランザクション・レベルのワークロード・バランシングを構成します。これにより、クライアントは、DB2 Connect サーバーを介さずに、トランザクション・レベルでワークロード・バランシングを実行できます。

手順

1. db2dsdriver 構成ファイルで、データベース項目または DSN 項目の WLB サブセクションにある enableWLB パラメーターを「true」に設定して、トランザクション・レベルのワークロード・バランシングをオンにします。例えば、db2dsdriver 構成ファイルで以下のように指定します。

```
<database name="SAMPLE" host="v33ec065.my.domain.com" port="446">
  <!-- database-specific parameters -->
  <WLB>
    <!-- Sysplex WLB is disabled by default -->
    <parameter name="enableWLB" value="true" />
  </WLB>
</database>
```

デフォルトでは、enableWLB は「false」で、ワークロード・バランシングは無効です。

2. オプション: 以下のパラメーターに対して値を指定して、ワークロード・バランシングの設定を微調整してください。ほとんどのアプリケーションでは、これらのパラメーターのデフォルト値で十分でしょう。

表 17. db2dsdriver 構成ファイルでのワークロード・バランシングの設定

パラメーター	説明
maxTransports	トランスポート・プール内の最大トランスポート数を指定します。デフォルト値は -1 (無制限) です。その他の負の値は無効です。値を 0 にすると、ワークロード・バランシングは無効になります。
maxTransportIdleTime	アイドル・トランスポートがドロップされるまでの最大経過時間を秒数で指定します。デフォルトは 600 です。サポートされる最小値は 0 です。
maxTransportWaitTime	トランスポートが使用可能になるまでクライアントが待機する秒数を指定します。デフォルトは -1 (無制限) です。サポートされる最小値は 0 です。
maxRefreshInterval	サーバー・リストがリフレッシュされるまでの最大経過時間を秒数で指定します。デフォルトは 30 です。サポートされる最小値は 0 です。

詳しくは、DB2 インフォメーション・センターの db2dsdriver 構成ファイルに関するトピックを参照してください。

注: 既存のデータベース・マネージャー構成設定 MAX_COORDAGENTS、NUM_POOLAGENTS、および MAX_CONNECTIONS は、DB2 Connect サーバーでのワークロード・バランシングの構成にのみ使用されるため、クライアント・サイドで実行されるワークロード・バランシングには影響しません。

例

例 1: デフォルト設定を使用して、データベース SAMPLE に対してトランザクション・レベルのワークロード・バランシングを使用可能にします。

```
<database name="SAMPLE" host="v33ec065.my.domain.com" port="446">
  <!-- database-specific parameters -->
  <WLB>
    <!-- Sysplex WLB is disabled by default -->
    <parameter name="enableWLB" value="true" />
  </WLB>
</database>
```

例 2: 最大 95 のトランスポートを使用してデータベース SAMPLE に対するトランザクション・レベルのワークロード・バランシングを使用可能にします。

```
<database name="SAMPLE" host="v33ec065.my.domain.com" port="446">
  <!-- database-specific parameters -->
  <WLB>
    <!-- Sysplex WLB is disabled by default -->
    <parameter name="enableWLB" value="true" />
    <!-- maxTransports represents the maximum number of transports;
    defaults to -1 (unlimited) -->
    <parameter name="maxTransports" value="95" />
  </WLB>
</database>
```

例 3: 最大 80 のトランスポートと使用可能なトランスポートの最大待機時間 10 秒を指定したデータベース SAMPLE に対して、トランザクション・レベルのワークロード・バランシングを使用可能にします。

```
<database name="SAMPLE" host="v33ec065.my.domain.com" port="446">
  <!-- database-specific parameters -->
  <WLB>
    <!-- Sysplex WLB is disabled by default -->
    <parameter name="enableWLB" value="true" />
    <!-- maxTransports represents the maximum number of transports;
    defaults to -1 (unlimited) -->
    <parameter name="maxTransports" value="80" />
    <!-- maxTransportWaitTime represents the maximum elapsed time in
    number of seconds before a transport becomes available;
    defaults to -1 (unlimited)-->
    <parameter name="maxTransportWaitTime" value="10" />
  </WLB>
</database>
```

例 4: データベース SAMPLE に対するトランザクション・レベルのワークロード・バランシングを使用不可にします。

```
<database name="SAMPLE" host="v33ec065.my.domain.com" port="446">
  <!-- database-specific parameters -->
  <WLB>
    <!-- Sysplex WLB is already disabled by default -->
    <parameter name="enableWLB" value="false" />
  </WLB>
</database>
```

自動クライアント・リルート (クライアント・サイド)

自動クライアント・リルート (ACR) 機能は、IBM データ・サーバー・クライアントが DB2 for z/OS Sysplex のメンバーへの接続を失った場合のフェイルオーバー・サポートを提供します。ACR を使用することにより、クライアントは、Sysplex の使用可能なメンバーを介してデータベースへの再接続を試行し、失敗からリカバリできるようになります。

ワークロード・バランシングが有効になっている場合、ACR はデフォルトで有効になります。

クライアント・サイド ACR サポートは、DB2 Connect ライセンスを持つ IBM データ・サーバー・クライアントおよび非 Java データ・サーバー・ドライバで使用可能です。ACR を実行するために DB2 Connect サーバーは必要ありません。

ACR が有効になっている場合、クライアントが既存の接続で接続障害を検出すると、通常、以下のプロセスが行われます。

1. クライアントは、既存の接続を使用して SQL ステートメントの実行を試行し、障害を検出します。
2. クライアントは、サーバー・リストを使用して、アクセスする Sysplex を識別し、データベースへの再接続を試行します。このサーバー・リストは、サーバーによって戻されるリストにすることも、またはクライアント・アフィニティーが指定されている場合、クライアントによって指定される代替サーバー・リストにすることもできます。詳しくは、135 ページの『クライアント・アフィニティーの構成』を参照してください。

3. ACR がデータベースに再接続できる場合、クライアントは新しく確立された接続の実行環境を再構成します。失敗したデータベース接続が回復したことを示す、およびトランザクションがロールバックされたことを示す、エラー SQL30108N がアプリケーションに戻されます。その後、アプリケーションはさらにリカバリーを行います。これには、ロールバックされた作業を繰り返すことが含まれます。
4. ACR がデータベースに再接続できない場合、エラー SQL30081N がアプリケーションに戻されます。その後、アプリケーションは接続障害からリカバリーする必要があります (例えば、データベースへの接続をアプリケーション自体で試行することによって)。

クライアントが新規の接続で接続障害を検出した場合も、ACR が使用されます。ただし、この場合、失敗したデータベース接続が回復したことを示す、エラー SQL30108N はアプリケーションに戻されません。接続が正常に行われるか、エラー SQL30081N が戻されるかのいずれかです。

db2dsdriver 構成ファイルで設定値を指定することによって、ACR を無効にするか、あるいはさらに構成を行うことが可能です。

CLI および .NET アプリケーションのシームレス・フェイルオーバー

ACR が有効になっていて、トランザクションのターゲットが DB2 for z/OS である場合、CLI および .NET アプリケーションのシームレス・フェイルオーバーはデフォルトで有効です。シームレス・フェイルオーバーでは、アプリケーションがトランザクションの最初の SQL 操作で接続障害を検出した場合、ドライバーは失敗した SQL 操作を自動クライアント・リレポート処理の一部として再生します。接続が正常に行われると、エラーはアプリケーションに報告されず、トランザクションはロールバックされません。接続障害とそれに続くリカバリーは、アプリケーションからは認識されません。

db2dsdriver 構成ファイルで設定値を指定することによって、シームレス・フェイルオーバーを無効または再度有効にすることができます。

ACR およびシームレス・フェイルオーバーに関する DB2 Connect サーバーの考慮事項

クライアントが DB2 Connect サーバーを介して Sysplex にアクセスする場合、ACR とシームレス・フェイルオーバーの振る舞いは異なります。

DB2 Connect サーバーと Sysplex との間で障害が発生した場合、ACR は DB2 Connect サーバーによって実行されます。DB2 Connect サーバーのレベルがクライアントと同じあるいはそれ以上の場合、クライアントはシームレス・フェイルオーバーを実行できます。そうでない場合、クライアントはシームレス・フェイルオーバーを実行せず、失敗したデータベース接続が回復したことを示す、エラー SQL30108N がアプリケーションに戻されます。

クライアントと DB2 Connect サーバーとの間で障害が発生した場合、ACR はクライアントで DB2 Connect サーバーに対して実行されます。ただし、シームレス・フェイルオーバーは常に無効になり、エラー SQL30108N がアプリケーションに戻されます。

自動クライアント・リルトの構成 (クライアント・サイド)

ワークロード・バランシングが有効になっている場合、自動クライアント・リルト (ACR) はデフォルトで有効になります。db2dsdriver 構成ファイルで設定値を指定することによって、ACR を無効にするか、あるいはさらに構成を行うことが可能です。

始める前に

ACR を実行するには、クライアントは TCP/IP 接続を使用しなければならず、DB2 Connect ライセンスを持っている必要があります。以下のクライアントは、ACR をサポートします。

- IBM Data Server Client
- IBM Data Server Runtime Client
- IBM Data Server Driver Package
- IBM Data Server Driver for ODBC and CLI

このタスクについて

一部のアプリケーションでは、ACR またはシームレス・フェイルオーバーを無効にする必要があるかもしれませんが、あるいは逆に ACR をもっと構成する必要があるかもしれません。このタスクでは、ACR の構成を有効にするパラメーターについて説明します。

ACR の動作を変更するには、クライアント・アフィニティーを指定して、接続処理および ACR で使用するサーバー・リストを制御することもできます。詳しくは、135 ページの『クライアント・アフィニティーの構成』を参照してください。

手順

ACR を無効にする、あるいはさらに構成するには、以下のようになります。

- ACR またはシームレス・フェイルオーバー付きの ACR を無効にするには、db2dsdriver 構成ファイルの以下のパラメーターに「false」の値を指定してください。データベース項目の ACR サブセクションでこれらの設定値を指定します。

表 18. db2dsdriver 構成ファイルでの自動クライアント・リルトの設定

パラメーター	説明
enableAcr	ACR 機能をオン (true) またはオフ (false) のどちらにするかを指定します。トランザクション・レベルのワークロード・バランシングが有効になっている (enableWLB 設定値が true である) 場合、この設定値はデフォルトで true です。ACR が無効になっている場合、ACR の他の設定値はすべて無視されます。

表 18. db2dsdriver 構成ファイルでの自動クライアント・リルトの設定 (続き)

パラメーター	説明
enableSeamlessAcr	シームレス・フェイルオーバー機能を含む ACR をオン (true) またはオフ (false) のどちらにするかを指定します。ACR が有効になっていて、ターゲット・サーバーが DB2 for z/OS である場合、シームレス ACR はデフォルトで有効です。ターゲット・サーバーが DB2 for z/OS でない場合、この設定は無視され、シームレス ACR はサポートされません。

詳しくは、DB2 インフォメーション・センターの db2dsdriver 構成ファイルに関するトピックを参照してください。

- 最初の接続試行で障害が発生し、かつその接続で指定したホストが使用不可の場合に使用する、代替サーバー・リストを指定することができます。代替サーバー・リストを指定するには、database 項目の ACR サブセクションで enableAlternateServerListFirstConnect に「true」の値を指定します。enableAlternateServerListFirstConnect の値が「true」の場合、db2dsdriver 構成ファイルで ACR に対して指定されるその他の設定値に関係なく、シームレス・フェイルオーバー付きの ACR は、暗黙的に有効になります。このフィーチャーを使用するには、db2dsdriver 構成ファイルで代替サーバー・リストもセットアップする必要があります。

代替サーバー・リストのセットアップに関する情報は、135 ページの『クライアント・アフィニティーの構成』を参照してください。クライアント・アフィニティーについて説明するトピックの中で、代替サーバー・リストのセットアップに関する情報が扱われますが、この機能を使用するためにクライアント・アフィニティーを使用する必要はありません。実際、トランザクション・レベルのワークロード・balancingを有効にすると、この機能もサポートされます。これは、トランザクション・レベルのワークロード・balancingとクライアント・アフィニティーは同時に有効にできないにもかかわらずです。

結果

ACR が有効になっている場合、接続障害を検出したアプリケーションは、Sysplex の別のメンバーに経路指定されます。シームレス・フェイルオーバーが無効になっているか適用されない場合、エラー SQL30108N がアプリケーションに戻されません。

トランザクションの最初の SQL 操作で障害が発生し、シームレス・フェイルオーバーが有効 (および適用可能) になっている場合、クライアントは、失敗した SQL 操作を ACR 処理の一部として再生できます。接続が正常に行われると、アプリケーションにエラーは報告されません。

例

例 1: データベース SAMPLE に対して ACR を有効にして、シームレス・フェイルオーバーを無効にする

```

<database name="SAMPLE" host="v33ec065.my.domain.com" port="446">
  <!-- database-specific parameters -->
  <WLB>
    <!-- Sysplex WLB is disabled by default -->
    <parameter name="enableWLB" value="true" />
  </WLB>
  <ACR>
    <!-- ACR is already enabled by default when WLB is enabled -->
    <parameter name="enableAcr" value="true" />
    <!-- If ACR is enabled and the target server is DB2 for z/OS,
    Seamless ACR is enabled by default; if the target server is
    not DB2 for z/OS, this setting is ignored and seamless ACR
    is not supported. -->
    <parameter name="enableSeamlessAcr" value="false" />
  </ACR>
</database>

```

例 2: データベース SAMPLE に対して ACR を無効にする

```

<database name="SAMPLE" host="v33ec065.my.domain.com" port="446">
  <!-- database-specific parameters -->
  <WLB>
    <!-- Sysplex WLB is disabled by default -->
    <parameter name="enableWLB" value="true" />
  </WLB>
  <ACR>
    <!-- ACR is enabled by default when WLB is enabled -->
    <parameter name="enableAcr" value="false" />
  </ACR>
</database>

```

例 3: 元のホストが使用不可の場合、最初の接続時に代替サーバー・リストを使用するように指定する

```

<database name="SAMPLE" host="v33ec065.svl.ibm.com" port="446">
  <ACR>
    <parameter name="enableACR" value="true"/>
    <parameter name="enableAlternateServerListFirstConnect"
      value="true"/>
  </ACR>
</database>

```

クライアント・アフィニティーの構成

初期接続処理の間、および自動クライアント・リルート (ACR) の処理の間、試行されるサーバーの順序を制御するために、クライアント・アフィニティーを構成することができます。トランザクション・レベルのワークロード・balancing が明示的に有効になっていない場合、クライアント・アフィニティーを使用することができます。

このタスクについて

通常、クライアント Sysplex 機能を使用するアプリケーションは、データベース・サーバーによって戻される最新サーバーのリストを使用して、障害時にどのメンバーに接続するか決定します。このモデルでは、クライアントは最後に読み取ったサーバー・リストに基づいてリルートを行うので、再試行するサーバーの順序は、サーバーによって制御されます。しかし、クライアント・アフィニティーを構成し、初期接続処理の間に、またクライアントによって実行されるクライアント・リルート処理のための使用される代替サーバー・リストを指定することによって、この動

作をオーバーライドすることができます。クライアント・アフィニティ処理がアクティブな場合、ACR 処理で試行されるサーバーの順序は、クライアントによって制御されます。

クライアント・アフィニティを構成するため、db2dsdriver 構成ファイルの ALTERNATE_SERVER_LIST、AFFINITY_LIST、および CLIENT_AFFINITY の設定値を指定します。これらの設定値が指定された場合

- アプリケーションによって指定されたホストやポートではなく、代替サーバー・リストに基づいて接続を確立します。例えば、アプリケーションは Host = s1 を指定し、代替サーバー・リストは Serverlist = (s2, s1, s3) を指定した場合、初期接続は s1 にではなく s2 に試行されます。接続試行の順序は、(s2,s2,s2) (s1,s1,s1) (s3,s3,s3) になります。acrRetryInterval は、個々の接続試行の間 (この例では合計 8 回) に適用されます。
- シームレス・クライアント・リルートは、自動的に有効になります。クライアント・リルートは、ALTERNATE_SERVER_LIST グループのサーバーおよびポート・パラメーターで指定されたサーバーへ接続することによって実行されます。代替リストのそれぞれのサーバーは、代替リストの次のサーバーへ移る前に maxAcrRetries の数値の回数だけ試行されます。これは、リストのすべてのサーバーを試行するまで、または接続の確立に成功するまで続けられます。例えば、代替サーバー・リストが (s3,s1,s2) と定義され、maxAcrRetries が 3 と定義された場合、リルートの順序は、(s3,s3,s3) (s1,s1,s1) (s2,s2,s2) となります。acrRetryInterval は、個々の接続試行の間 (この例では合計 8 回) に適用されま

手順

db2dsdriver 構成ファイルのクライアント・アフィニティの設定値を指定するには、以下のようにします。

1. db2dsdriver 構成ファイル内の ACR グループで、ALTERNATE_SERVER_LIST の項目を指定することによって、代替サーバーのリストを宣言します。このリスト内に、それぞれのサーバーのポート番号とホスト名を指定します。例えば、次のようにします。

```
<database name="SAMPLE" host="v33ec065.sv1.ibm.com" port="446">
  <acr>
    <parameter name="enableACR" value="true"/>
    <alternate_server_list>
      <server name="server1"
        hostname="v33ec067.sv1.ibm.com"
        port="446" >
      </server>
      <server name="server2"
        hostname="v33ec066.sv1.ibm.com"
        port="446" >
      </server>
      <server name="server3"
        hostname="v33ec065.sv1.ibm.com"
        port="446" >
      </server>
    </alternate_server_list>
  .
  .
  .
</database>
```

このリストを指定することは、単に代替サーバーを宣言するだけです。これによって、動作が変わるわけではありません。

2. ACR グループの下で、AFFINITY_LIST に対して項目を指定することにより、サーバーの順序を特定します。例えば、次のようにします。

```
.  
. .  
. .  
  <affinity_list>  
    <list name="list1"  
      serverorder="server1,server2,server3" >  
    </list>  
    <list name="list2"  
      serverorder="server3,server2,server1" >  
    </list>  
  </affinity_list>  
  
. .  
. .
```

このリストを指定することは、単にサーバーの順序を指定するだけです。これによって、動作が変わるわけではありません。

3. ACR グループの下で、以下のクライアント・アフィニティー・サブグループの 1 つを指定して、クライアント・アフィニティーを有効にします。これらのサブグループを 1 つ指定すると、強制的にクライアント・アフィニティーが有効になります。このデータベースに接続するすべてのクライアントは、CLIENT_AFFINITY サブグループの 1 つに指定される必要があります。あるクライアントがこれらのサブグループのいずれにも見つからない場合、接続を試している間にエラーが発生します。CLIENT_AFFINITY サブグループが存在していると、ACR は暗黙的に有効になります。

- CLIENT_AFFINITY_DEFINED

クライアント・ホスト名から特定の AFFINITY_LIST 要素への固有のマッピングを指定します。クライアント・ホスト名は自己発見され、db2dsdriver 構成ファイルの項目と突き合わせられて、アフィニティー・リストが判別されます。例えば、次のようにします。

```
. .  
. .  
  <client_affinity_defined>  
    <!-- this section has specific defined affinities  
    -->  
    <client name="client1"  
      hostname="appsrv1.svl.ibm.com"  
      listname="list2" >  
    </client>  
    <client name="client2"  
      hostname="appsrv2.svl.ibm.com"  
      listname="list1" >  
    </client>  
  </client_affinity_defined>  
  
. .  
. .
```

- CLIENT_AFFINITY_ROUNDROBIN

ALTERNATE_SERVER_LIST 内にラウンドロビンの割り当てを指定します。この割り当てでは、ALTERNATE_SERVER_LIST のサーバーの数を法として、CLIENT_AFFINITY_ROUNDROBIN リスト内のクライアントをモジュロ演算した結果をインデックス (ゼロ・ベース) として、開始するサーバーを設定します。例えば、次のようにします。

```
.
.
.
<client_affinity_roundrobin>
  <!-- roundrobin picks the starting server as
        the client index number in this section (0 based)
        modulo the number of servers.
  -->
  <client name="client3"
        hostname="appsrv3.svl.ibm.com" >
    <!-- this entry is index 0, modulo 3, thus it gets:
          server1, server2, server3
    -->
  </client>
  <client name="client4"
        hostname="appsrv4.svl.ibm.com" >
    <!-- this entry is index 1, modulo 3, thus it gets:
          server2, server3, server1
    -->
  </client>
</client_affinity_roundrobin>
.
.
.
```

結果

以下の考慮事項および妥当性検査の規則をクライアント・アフィニティ処理に適用します。

- CLIENT_AFFINITY が有効 (つまり、db2dsdriver 構成ファイルに 1 つ以上の CLIENT_AFFINITY サブグループが存在する) な場合、maxAcrRetries のデフォルト値は 3 です。この値の有効範囲は 0 から MAX_INT までです。指定された値が 0 の場合、maxAcrRetries に 1 が指定された場合と同じように、ACR 処理が行われます。
- acrRetryInterval のデフォルト値は 0 です。この値の有効範囲は、0 から MAX_INT までです。値が 0 の場合、ドライバーで接続再試行つまり ACR 試行の間隔はありません。
- 1 つのクライアントに対する適格な代替サーバーの総数は、24 までです。適格な代替サーバーの数が 24 より大きい場合、SQL1042N が発生します。
- enableWLB と CLIENT_AFFINITY の設定は、相互排他です。enableWLB と CLIENT_AFFINITY を両方とも有効にすると、エラー SQL5162N が発生します。
- ALTERNATE_SERVER_LIST、AFFINITY_LIST、および CLIENT_AFFINITY グループでは必須属性を指定する必要があります。そうしないと、SQL5163N が発生します。
- CLIENT_AFFINITY が有効な場合、ALTERNATE_SERVER_LIST を空にすることはできません。違反すると、SQL5164N が発生します。
- CLIENT_AFFINITY が有効な場合、クライアントのホスト名を有効な代替サーバー・リストで解決する必要があります。違反すると、SQL5164N が発生します。

例えば、クライアントのホスト名が「xyz.domain.com,」で、そのホスト名がどの CLIENT_AFFINITY サブグループにも見つからなかった場合、SQL5164N が発生します。加えて、このホスト名が現れているのは、CLIENT_AFFINITY グループのうちの 1 つだけです。

例

以下の例では、CLIENT_AFFINITY サブグループを指定した db2dsdriver 構成ファイルのサンプルの database セクションを示します。

```
<database name="SAMPLE" host="v33ec065.svl.ibm.com" port="446">
  <acr>
    <parameter name="enableACR" value="true"/>
    <alternate_server_list>
      <server name="server1"
        hostname="v33ec067.svl.ibm.com"
        port="446" >
      </server>
      <server name="server2"
        hostname="v33ec066.svl.ibm.com"
        port="446" >
      </server>
      <server name="server3"
        hostname="v33ec065.svl.ibm.com"
        port="446" >
      </server>
    </alternate_server_list>
    <affinity_list>
      <list name="list1"
        serverorder="server1,server2,server3" >
      </list>
      <list name="list2"
        serverorder="server3,server2,server1" >
      </list>
    </affinity_list>
    <client_affinity_defined>
      <!-- this section has specific defined affinities
      -->
      <client name="client1"
        hostname="appsrv1.svl.ibm.com"
        listname="list2" >
      </client>
      <client name="client2"
        hostname="appsrv2.svl.ibm.com"
        listname="list1" >
      </client>
    </client_affinity_defined>
    <client_affinity_roundrobin>
      <!-- roundrobin picks the starting server as
        the client index number in this section (0 based)
        modulo the number of servers.
      -->
      <client name="client3"
        hostname="appsrv3.svl.ibm.com" >
        <!-- this entry is index 0, modulo 3, thus it gets:
        server1, server2, server3
      -->
      </client>
      <client name="client4"
        hostname="appsrv4.svl.ibm.com" >
        <!-- this entry is index 1, modulo 3, thus it gets:
        server2, server3, server1
      -->
```

```
        </client>
    </client_affinity_roundrobin>
</acr>
</database>
```

クライアント Sysplex サポートの使用に関する制限

IBM データ・サーバー・クライアントおよび非 Java データ・サーバー・ドライバで使用する可能なクライアント Sysplex サポートには、いくつかの制限があります。

- トランザクション・レベルのワークロード・バランシングの制限は以下のとおりです。
 - クライアントは TCP/IP 接続を使用する必要があります。
 - クライアント Sysplex サポートを使用する IBM データ・サーバー・クライアントおよびドライバの場合、サーバー・リストはアプリケーション・プロセス内でキャッシュされます。このため、リストは、ワークロード・バランシングのために 1 つ以上のスレッドから、そのプロセス内の接続によってのみ共有されます。
- 自動クライアント・リルトの制約事項:
 - クライアントは TCP/IP 接続を使用する必要があります。
 - Sysplex サーバーが、直前のトランザクションの最後でのトランスポートの再利用を許可しない場合、トランザクションの最初の SQL ステートメントでのシームレス・フェイルオーバーはサポートされません。例外は、トランスポートの再利用が KEEP DYNAMIC(YES) の結果として付与されない場合です。
 - SQL ステートメントを再生するために、以前に呼び出された API の履歴を、CLI ドライバが維持する必要がある場合、トランザクションの最初の SQL ステートメントでのシームレス・フェイルオーバーはサポートされません。これが発生する可能性があるシナリオには、実行時のデータの指定、複合 SQL の実行、および配列入力の使用が含まれます。
 - 自動コミットが使用可能になっている場合にシームレス・フェイルオーバーが発生すると、SQL ステートメントが複数回実行およびコミットされる可能性があります。この結果が発生するのは、元の自動コミットが正常に完了した後、自動クライアント・リルトをトリガーするイベントが発生した場合です。この動作を回避するには、自動コミットまたはシームレス・フェイルオーバーのいずれかを使用不可にします。
 - シームレス・フェイルオーバーはストアード・プロシージャの内部には適用されません。
 - シームレス・フェイルオーバーはフェデレーテッド・サーバー環境には適用されません。
- XA サポート:
 - 以下のトランザクション・マネージャー処理モデルはサポートされません。
 - デュアル・トランスポート。このモデルでは、トランスポート A を介したトランザクションは、xa_start から xa_end までメンバーに関連付けられますが、xa_prepare (読み取り専用)、xa_prepare に xa_commit または xa_rollback を加えたもの、または xa_rollback は、おそらく別のアプリケーション・プロセスから、トランスポート B を介して入れられます。このモ

デルを使用するトランザクション・マネージャーの例は、IBM WebSphere MQ および IBM Lotus Domino です。

- マルチ・トランスポート。このモデルには、同じトランザクションに対する、複数のアプリケーション・プロセスからの複数のトランスポートの使用が関係しています。このモデルを使用するトランザクション・マネージャーの例としては、BEA Tuxedo があります。
- マルチ・トランスポート処理モデルを使用する XA トランザクション・マネージャーの場合、中間層 DB2 Connect サーバーが依然として必要です。

第 20 章 DB2 Connect のチューニング

データベース・マネージャー構成ファイルの各種パラメーターは、DB2 Connect をチューニングするために使用することができます。

RQRIOBLK

RQRIOBLK パラメーターは、ネットワーク入出力ブロックの最大サイズを設定します。ブロック・サイズを大きくすれば、大きな要求のパフォーマンスを改善することができます。ブロック・サイズは、通常、単一行のデータを求める要求のような、小さな要求の応答時間には影響を与えません。

通常はブロック・サイズを大きくすると、DB2 Connect サーバー上で、より大きなメモリーが必要になります。このことは、作業セットのサイズを大きくし、小さなワークステーションで大量のページングが生じる原因となります。

アプリケーションの実行に際してそれほど大きなページングを起こさない場合は、デフォルトの DRDA ブロック・サイズ (32767) を使用します。その他の場合は、ページングがなくなるまで入出力ブロック・サイズを引き下げます。いったんページングが始まると、パフォーマンスの著しい低下が起こります。パフォーマンス・モニター・ツール (例えば、Linux および UNIX オペレーティング・システムの vmstat ツール) を使用して、システム上でページングが起きているかどうかを判別してください。

DIR_CACHE

DIR_CACHE パラメーターにより、ディレクトリー情報がキャッシュされるかどうかが決まります。キャッシュを使用する場合 (DIR_CACHE=YES)、ディレクトリー・ファイルが読み込まれてメモリーにキャッシュされます。これによって、内部ディレクトリー構造が作成され、接続が確立されるたびにディレクトリー・ファイルを読み取るというオーバーヘッドが最小化されます。

キャッシュを使用しない場合 (DIR_CACHE=NO)、データベースへの接続が行われるたびに、該当するディレクトリーがディスクから読み取られ、次に検索が行われます。要求項目が見つかった後、ディレクトリー検索に関係したすべてのメモリーが解放されます。

キャッシュを使用すると、共有ディレクトリー・キャッシュが db2start プロセスの間に作成され、DB2 が停止する時に解放されます。このキャッシュは、すべての DB2 サーバー・プロセス (db2agent) によって使用されます。また、アプリケーションがデータベースへのその最初の接続命令を発行した時、私用アプリケーション・ディレクトリー・キャッシュが作成され、アプリケーションが終了した時に解放されます。

各キャッシュは、システム・データベース・ディレクトリー、データベース接続サービス・ディレクトリー、およびノード・ディレクトリーのイメージを提供します。キャッシュは、ディレクトリー・ファイル入出力を削減し、ディレクトリー検索を最小化することにより、接続コストを引き下げます。

キャッシュされたディレクトリーが更新されても、その変更は即時にはキャッシュへ伝搬されません。ディレクトリー項目がキャッシュ内に見つからない場合は、元のディレクトリーが検索されます。

キャッシュによって、アプリケーションが存続する間必要とされる私用メモリーが大きくなります。キャッシュを使用しなければ、このメモリーはディレクトリー参照処理の間だけ必要になります。DB2 による共有メモリーの全般的な使用は少し増加します。データベース・エージェントの間で共有されるディレクトリー情報は共有メモリーへ移動するからです。キャッシュに必要なメモリーのサイズは、各ディレクトリー内の項目の数によります。

NUMDB

旧バージョンでは、DB2 Connect の動作は **NUMDB** 構成パラメーターの影響を受けませんでした。バージョン 8 以降この点が変更されました。このパラメーターは、クライアントが DB2 Connect サーバーによって接続可能なデータベースの最大数を示します。具体的には、DB2 Connect サーバー上でカタログ可能な別個のデータベース別名の最大数です。

他の DB2 Connect パラメーター

AGENTPRI および **MAXAGENTS** はバージョン 9.5 では推奨されていません。

MAXAGENTS の値を更新するコマンドは引き続き機能するため既存のアプリケーションに問題は起きませんが、値は無視されます。パラメーター名は構成リストには表示されません。以前は、指定の DB2 パーティションで作成できるエージェントの合計数は **MAXAGENTS** 構成パラメーターで制御されていました。現在、エージェントの構成は自動化されています。

デフォルトでは、**NUM_POOLAGENTS** は **AUTOMATIC** に設定され、デフォルト値である 100 が指定されます。さらに、デフォルトでは、**MAX_COORDAGENTS** も **AUTOMATIC** に設定され、デフォルト値として 200 が指定されます。

アカウントिंग・ストリングをクライアント・アプリケーションから DB2 Connect サーバーに送信するには、アカウントिंग情報の設定に API 固有の方法を使用します。API 固有の方法を使用することで、**DB2ACCOUNT** 環境変数を設定するよりも早く実行できるようになります。

IBM Data Server Driver for JDBC and SQLJ

`com.ibm.db2.jcc.DB2BaseDataSource.clientAccountingInformation` プロパティー

IBM Data Server Provider for .NET

`DB2Connection.ClientAccountingInformation` プロパティー

CLI/ODBC

`ClientAcctStr` CLI/ODBC 構成キーワード

組み込み SQL (C、C++、および COBOL)

`sqlsact` 関数

調整済みの **SQLCODE** マッピング・ファイルを必要としない場合は、デフォルトの **SQLCODE** マッピングを使用するか、または **SQLCODE** マッピングをオフにすることによって、パフォーマンスを改善することができます。デフォルトのマッピン

グ・ファイルは、DB2 Connect ライブラリーに組み込まれています。調整済みのマッピング・ファイルはディスクから読み取る必要がありますが、これがパフォーマンスに影響を与えます。

ホスト・データベースのチューニング

システム・パフォーマンスは、ホストまたは IBM メインフレーム・データベース・サーバーのパフォーマンスによって影響を受けます。それぞれのデータベース管理システムによって、異なるパフォーマンス・フィーチャーが備わっています。各種システムの SQL オプティマイザーは、例えば、同じアプリケーションを使用しても異なる挙動をとることがあり得ます。詳細については、IBM メインフレーム・データベース・サーバーのシステム・パフォーマンス資料を参照してください。

ジャーナルを回避するために、可能な場合には非コミット読み取り (UR) またはコミットなし (NC) のバインド・オプションを使用することによって、パフォーマンスを改善することができます。

注: UR を使用する場合、ジャーナルされていないデータは、読み取りはできますが更新はできません (これは、ブロッキングを ALL に設定している場合です)。

アプリケーション・サーバーおよびそれが提供するロックの細分性により、照会またはアプリケーションに使用される分離レベルは、パフォーマンスに著しい影響を与えることがあります。データベースは、適切なレベルの正規化、索引の効果的な使用、およびデータベース・スペースの適切な割り振りを行う必要があります。また、パフォーマンスは、以下のセクションで説明するとおり、使用するデータ・タイプによっても影響を受ける場合があります。

ネットワークのチューニングに関する考慮事項

分散データベース環境で全体のパフォーマンスを向上させるには、ネットワークからの遅延をなくすことが最善の方法です。一般に、ネットワーク管理者にとって効率的なネットワークとは、伝送と伝送の間にできる限り多くのデータが収集されるネットワークです。この考え方は、分散データベースなどのアプリケーションには当てはまりません。この考え方では、ネットワーク内に遅延を発生させるからです。エンド・ユーザーにはネットワークの効率がよいとは思えず、遅延していることだけが分かります。

ほとんどのネットワーク装置には遅延パラメーターがありますが、その大部分のデフォルトは分散データベースにとっては非常に不適切なものです。パフォーマンスを改善するには、この種のパラメーターを見付けて、可能であればゼロに設定する必要があります。また、装置のバッファ・サイズが十分で、データが脱落して再送が行われたりしないように確実に設定する必要があります。例えば、UNIX システムでは送信または受信キューの深さのデフォルトは通常 32 です。パフォーマンスを向上させるには、キューの深さを 150 に設定してください。DLC の設定でこれに対応するパラメーターは Receive Depth で、この値も 150 に設定する必要があります。

ほとんどの場合、IOBUF パラメーターは低過ぎる値に設定されています。この値は普通は 500 に設定されていますが、経験が示すところによると、特に ESCON® や 3172 などのチャネル接続の場合に、大量のデータを移動するには 3992 の値が最適です。

LAN システムでは、DLC や LLC の送信ウィンドウや受信ウィンドウのサイズによりパフォーマンスはかなり左右されます。送信値は 7 以上に設定する必要があります。また、ほとんどの構成では受信値を 4 以下に設定するのが最適です。

イーサネットを実行している場合は、TCP セグメントのサイズを 1500 バイトに設定する必要があります。トークンリングや FDDI ネットワークの場合はこの値を 4400 バイトにする必要があります。TCP/IP で ESCON アダプターを使用している場合はセグメントのサイズを常に 4096 にする必要があります。

最後に TCP/IP ネットワークの場合は、TCP 送信および受信バッファ・サイズを 32768 より大きな値に設定する必要があります。通常は 65536 の値が最適です。

注: ゲートウェイからサーバーへの接続 (アウトバウンド接続) を確立するには、クライアントからゲートウェイへの接続 (インバウンド接続) を確立する場合よりコストがかかります。数千のクライアントがゲートウェイを介してサーバーに対し接続と切断を頻繁に繰り返す環境では、アウトバウンド接続を確立するのに相当の処理時間を要します。DB2 Connect では TCP/IP 上で接続プーリングが行われます。クライアントがサーバーからの切断を要求すると、ゲートウェイではクライアントとのインバウンド接続はドロップされますが、サーバーとのアウトバウンド接続はプール内に維持されます。新しいクライアントからゲートウェイに対して接続要求がなされると、ゲートウェイによりプールから既存の接続が提供されるので、接続時間が全体として短縮され、サーバーにおける接続のための高い CPU コストが節約されます。

表 19 に、ネットワーク・パフォーマンスのチューニング方式をまとめます。

表 19. ネットワーク・パフォーマンスのチューニング方法

参照するもの	例	設定	注意
意図的な遅延	ネットワーク装置の遅延パラメーター	0 に設定します。	通常、デフォルトは大き過ぎます。
バッファ	IOBUF パラメーター	3992 に設定します。	ESCON や他のチャネル・アダプターの場合に特に有効です。
バッファ	RUSIZE	最適なサイズは 4096 です。	RUSIZE と RQRIOBLK を同サイズに設定すると最善のパフォーマンスが得られます。
バッファ	ペーシング	VPACING、PACING、および Mode Profiles を 63 に設定します。	適切な場合は適応ペーシングを使用します。
アダプターの設定	送信/受信キューの深さ	推奨値は 150。	通常、デフォルトは 32 です。

表 19. ネットワーク・パフォーマンスのチューニング方法 (続き)

参照するもの	例	設定	注意
TCP の設定	セグメント・サイズ	イーサネットでは 1500、トークンリングおよび FDDI では 4400 です。	TCP/IP 接続に ESCON アダプターを使用する場合は必ず 4096 に設定する必要があります。
TCP の設定	送信/受信スペースのサイズ	両方とも 64K にする必要があります。	Windows の場合、デフォルトはほんの 8192 です。Windows レジストリーで設定できます。

システム・リソースの競合

システム内の多くのタスクがシステム・リソースを求めて競合する場合は、パフォーマンスの低下があり得ます。以下の質問を考慮してください。

- CPU は飽和状態ですか。プロセス・オーバーヘッド低減のため、システムのアップグレード、システム・ワークロードの低減、およびシステムのチューニングを考慮してください。
- メモリーは、オーバー・コミットになっていますか。メモリーの実効ページ・セット低減のため、メモリーのアップグレード、システム・ワークロードの低減、およびシステムのチューニングを考慮してください。
- 通信アダプター/通信コントローラーは混雑していますか。ネットワークのアップグレードまたはトークンリング・カードのペア化を考慮してください。
- サブシステムの 1 つが特に混雑していますか。そしてこのサブシステムはデータ・パス上にありますか。
- 不必要なプロセスまたはタスクがシステム上で稼働していませんか。一般規則として、常時使用するサービスでなければ、システム・リソースの浪費になるため、構成したり起動したりしないことです。
- 少数のプロセスまたはタスクがリソースの大部分を使用していますか。それらは止められますか。それらの優先順位は下げられますか。それらが多くのリソースを使わないように改善できますか。

DB2 Connect のパフォーマンスに関するトラブルシューティング

DB2 Connect ユーザーが、IBM メインフレーム・サーバーに大きな照会をしていて、長い時間待ってもサーバーから応答がない場合、以下の領域を考慮して、考えられるパフォーマンス上の問題の原因を調べてください。

1. IBM メインフレーム・サーバーから多数のデータ・ブロックが返される照会に関しては (通常は 32K かそれ以上のデータ)、データベース・マネージャー構成パラメーター RQRIOBLK を必ず 32767 に設定してください。これは、コマンド行プロセッサ (CLP) を以下のように使用して行います。

```
db2 update database manager configuration using RQRIOBLK 32767
```

2. IBM RDB モード定義で定義した最大 RU サイズが適切な値に設定されているか確認してください。トークンリング・ハードウェアを使用して接続する場合、サ

イズは 4K を下回らない値に設定することをお勧めします。イーサネット・ハードウェアを使用して接続する場合、イーサネット・フレーム・サイズの最大値は 1536 バイトであることに注意してください。その値が限界になります。

DB2 for z/OS のチューニング

z/OS では、非アクティブなスレッド処理を最適化できます。V5 では最大で 25,000 までのクライアントを並行接続できます。ただし、いずれの場合も並行してアクティブにできる最大数は 1999 です。個々のワークステーション・クライアントは非アクティブになっても接続したままにしておくことができます。そのスレッドは、コミットのたびに非アクティブ・チェーンに組み込まれます。

DSNZPARM パラメーターの CMTSTAT、CONDBAT、および MAXDBAT はスレッド処理に影響を与えます。最高のパフォーマンスを得るには、CMTSTAT を INACTIVE に設定し、良好なパフォーマンスが得られる DBAT の最大接続数に CONDBAT を調整し、受け入れられるアクティブな DBAT の最大数に MAXDBAT を調整してください。

DB2 Connect のデータ転送速度の向上

照会の結果セット用の行ブロッキングに加えて、DB2 for z/OS は、DB2 Connect などのリモート・クライアントへの OPEN または FETCH 要求に対して、複数の照会ブロックを戻すこともできます。一度に 1 つのブロックの行データを要求してクライアントが DB2 for z/OS サーバーに繰り返し要求を送信する代わりに、オプションでサーバーが照会ブロックのいくつかを、通常送り返す分に加えて送り返すよう要求できるようになりました。そのような追加の照会ブロックを、追加照会ブロックといいます。

この新規フィーチャーにより、クライアントはネットワーク回線のターンアラウンド (これは、ネットワーク・パフォーマンスの大きな負担となっている) を最小限にすることができます。照会ブロックに関してクライアントからサーバーに送られる要求の数が減少することにより、パフォーマンスは著しく向上します。なぜパフォーマンスが向上するかというと、送受信の切り替えはパフォーマンスに負荷を与える操作だからです。DB2 Connect は、デフォルトで DB2 for z/OS サーバーからの追加照会ブロックを要求することにより、このパフォーマンスの機能強化を利用できるようになりました。

優先ネットワーク・プロトコルの TCP/IP で追加照会ブロックの戻り (それぞれ最大 32K バイト長) を充分利用するために、DB2 Connect の RFC-1323 の設計で、ウィンドウ・スケーリングの拡張機能が使用できるようになりました。このフィーチャーにより、TCP/IP は送受信のウィンドウ・サイズを動的に調整し、追加照会ブロックによって大量に戻される可能性のあるデータを効率よく収容することができます。

追加照会ブロック

DB2 for z/OS バージョン 7 以降を持つサーバー上の追加照会ブロック・サポートは、DB2 DDF インストール・パネル上の EXTRA_BLOCKS_SRV パラメーターで構成されます。このサポートは、要求に関して DB2 がクライアントに送り返すことのできる追加照会ブロックの最大数を制御することにより、構成されます。このパラメーターの値は 0 から 100 に設定できます。パラメーター値を 0 に設定すると、追加照会ブロックを戻すことができなくなります。このフィーチャーから最大の益を得るには、常にデフォルト値の 100 を使用してください (ネットワークにその設定を理想的としない特有の傾向がある場合を除く)。

クライアント側 (アプリケーションは DB2 for z/OS に、同じ場所にある DB2 Connect インストール・システムを直接介して、あるいは別個の DB2 Connect サーバー・インストール・システムを介してアクセスする) には、対応する DB2 Connect サポートをカーソル単位、またはステートメント単位で活動化するための様々な手段があります。

- カーソルで照会の行セット・サイズを使用する
- カーソルに関連した SELECT ステートメントで 'OPTIMIZE for N ROWS' 節を使用する
- カーソルに関連した SELECT ステートメントで 'FETCH FIRST N ROWS ONLY' 節を使用する

DB2 Connect は、別の SQL API を使って追加照会ブロック・サポートを使用可能にすることができます。

組み込み SQL

- ユーザーは、SELECT ステートメントそのものに 'OPTIMIZE for N ROWS' 節、または 'FETCH FIRST N ROWS ONLY' 節のいずれか、あるいはその両方を指定することにより、照会の追加照会ブロック・サポートを呼び出すことができます。
- 'OPTIMIZE for N ROWS' 節を使用すると、DB2 for z/OS は EXTRA_BLOCKS_SRV DDF インストール・パラメーター設定に従って、DB2 Connect に戻す指定の行数のブロック化を試みます。DB2 for z/OS は、最終的に戻される照会結果セットの合計行数を N に制限していないので、アプリケーションは N 行以上取り出すことを選択できます。
- 'FETCH FIRST N ROWS ONLY' 節も同じように機能しますが、DB2 for z/OS によって照会結果セットが N に制限される点が異なります。N を超える行数を取り出すと、SQL コードは +100 (データの終わり) になります。

CLI/ODBC

- ユーザーは、SQL_MAX_ROWS ステートメント属性を介して、照会の追加照会ブロック・サポートを呼び出すことができます。
- DB2 for z/OS 7.1 以降のサーバーの場合は、代わりに 'FETCH FIRST N ROWS ONLY' 節が使用されます。
 - バージョン 7 では、照会結果セットは DB2 for z/OS により N 行に制限されます。N を超える行数を取り出すと、SQL_NO_DATA_FOUND となります。

- バージョン 8 以降では、CLI は最初の N 行だけがクライアントのカーソル・マネージャーを介して、アプリケーションに戻されるようになります。

JDBC

ユーザーは、`setMaxRows` メソッドを介して、追加照会ブロック・サポートを照会のために呼び出すことができます。CLI/ODBC を使って使用可能にする場合と同様、DB2 for z/OS 6.x サーバーの場合、DB2 Connect は 'OPTIMIZE for N ROWS' 節を付けます。DB2 Connect は DB2 for z/OS 7.1 以降のサーバーの場合も、'FETCH FIRST N ROWS ONLY' 節を付けません。

RFC-1323 ウィンドウ・スケーリング

TCP/IP 用 RFC-1323 拡張機能をサポートするすべての Windows、Linux および UNIX プラットフォームでは、ウィンドウ・スケーリングがサポートされています。DB2 レジストリー変数 `DB2SORCVBUF` を用いて、DB2 for Windows、Linux、または UNIX でこのフィーチャーを使用可能にできます。ウィンドウ・スケーリングを有効にするには、このレジストリー変数を 64K より大きな値に設定しなければなりません。例えば、DB2 for Windows、Linux、または UNIX では、`db2set DB2SORCVBUF =65537` を発行できます。

最大の送信バッファおよび受信バッファのサイズは、その特定のオペレーティング・システムに依存します。ユーザーは、データベース・マネージャーの構成パラメーター `DIAGLEVEL` を 4 (通知) に設定し、管理通知ログ・ファイルのメッセージをチェックすることにより、構成されたバッファ・サイズが受け入れられたことを確かめることができます。

ウィンドウ・スケーリングが有効になるためには、接続の両端でウィンドウ・スケーリングが使用可能になっていなければなりません。つまり、直接オペレーティング・システムの TCP/IP スタックを通してであれ、間接的に DB2 製品を通してであれ、ワークステーションとホストの両方で使用可能になっているべきです。例えば、現在のところ DB2 for z/OS では、オペレーティング・システムで `TCPRCVBUFRSIZE` を 64K より大きな値に設定することによってのみ、ウィンドウ・スケーリングをアクティブにすることができます。ユーザーがリモート IBM Data Server Client を使用していて、DB2 Connect サーバー・ワークステーションを通して IBM メインフレーム DB2 データベースにアクセスする場合、クライアント側でもウィンドウ・スケーリングを使用可能にすることができます。同様に、IBM メインフレーム DB2 データベースが関係していない場合、リモート IBM Data Server Client とワークステーション DB2 サーバーの間でもウィンドウ・スケーリングを使用可能にすることができます。

ウィンドウ・スケーリングはネットワーク・パフォーマンスを向上させるために設計されていますが、ネットワーク・パフォーマンスの改善が常に期待通りに実現するわけではないという点に注意してください。いったんウィンドウ・スケーリングが使用可能になると、イーサネットやトークンリング LAN アダプターで使用されるフレームのサイズ、IP MTU のサイズ、さらに通信リンク全体にわたるルーターのその他の設定など、さまざまな要素の相互作用によりパフォーマンスが低下する

ことさえあります。そのために、デフォルトではウィンドウ・スケーリングは使用不可になっており、送信バッファおよび受信バッファ共に 64K に設定されています。

ウィンドウ・スケーリングを使用した場合の影響を査定できるようにしておき、ネットワークに必要な調整を行ってください。ネットワークを調整してネットワーク・パフォーマンスを改善する方法の概要については、<http://www.networking.ibm.com/> を参照してください。

ホスト・データの変換

情報が異なる環境の間 (Intel [Windows]、IEEE [Linux および UNIX オペレーティング・システム]、System z [VM、VSE、z/OS]、IBM Power Systems [OS/400] など) で転送される場合、数値データ・タイプ (10 進数、整数、浮動小数点など) を変換しなければならない場合があります。この変換はパフォーマンスに影響を与えます。

1 バイト文字のデータ変換の CPU コストは、数値データの変換コストよりも一般に小さいといえます (データ変換が必要な場合)。

DATE/TIME/TIMESTAMP のデータ変換コストは、1 バイト CHAR の場合とほぼ同じです。浮動小数点データの変換コストが最大です。アプリケーション設計者は、DB2 Connect ベースのアプリケーションを設計するときは、これらの事実の利点を取り入れることができます。

データベース表が 'FOR BIT DATA' と定義される列を持っている場合、アプリケーションとデータベースとの間で転送される文字データはデータ変換をなんら必要としません。このことは、IBM メインフレーム・データベース・サーバー上でデータを保存するときに利用することができます。

文字データのデータ・タイプ

文字データは、CHAR または VARCHAR のどちらかのデータ・タイプを持つことができます。どのデータ・タイプがより効率的かは、フィールド内のデータの代表的な長さによります。

- 実際のデータ・サイズがかなり変化する場合は、VARCHAR の方が、より効率的といえます。というのは、CHAR の場合はフィールドを埋めるために余分のブランク文字を追加する必要があるからです。これらのブランク文字は、他の文字と同様、ネットワークを通して伝送される必要があります。
- 実際のデータのサイズがそれほど変化しない場合は、CHAR の方がより効率的です。なぜなら各 VARCHAR フィールドには、長さ情報の数バイトがあり、それも伝送しなければならないからです。

ネットワーク・ハードウェア

以下の考慮事項は、ハードウェアに関するものです。

- ネットワークまたは伝送メディアの速度

パフォーマンスは、より速い伝送メディアの使用により改善されます。例えば、以下に挙げるような、いくつかの代表的なロー・データ転送速度があります。

チャンネル間 (光ファイバー)

4.0 MB/s

16 Mbps LAN

2.0 MB/s

チャンネル間 (正規)

1.0 MB/s

4 Mbps LAN

0.5 MB/s

高速 T1 搬送波 (1.544 Mbps)

0.193 MB/s

高速リモート 56 Kbps 電話回線

0.007 MB/s

19.6 Kbps モデム

0.002 MB/s

9600 bps モデム

0.001 MB/s

データ転送速度は、IBM メインフレーム・データベース・サーバーへのバスの中の最も遅い伝送メディアによって制約されます。

- ネットワーク・アダプターまたは通信コントローラー

ネットワーク・アダプターおよび通信コントローラーのメモリー使用については、注意深く計画を立てる必要があります。さらに、ネットワーク担当技術者と一緒に、コントローラーが DB2 Connect で発生した余分なトラフィックを取り扱う能力をもっているか確認してください。

- ネットワーク・トポロジー

データが LAN から LAN へと、また 1 つのネットワークから別のネットワークへとわたって移動する場合は、移動にかかる時間を考慮してください。ブリッジ、ルーター、およびゲートウェイは経過時間を上乗せすることになります。例えば、交差するブリッジの数を減らせば、各要求に必要なホップの数を減らすことになります。

ノード間の物理的距離も、考慮に入れなければなりません。メッセージが通信衛星によって転送される場合でも、転送時間は、光速 ($3 * 10^{*8}$ m/s) および送信者と受信者との間の往復距離によって制約を受けます。

- ネットワーク・トラフィック

ネットワークの帯域幅がフルに使用されている場合、単一アプリケーションについての応答時間とデータ転送速度が減少します。

データがネットワークの特定部分に蓄積されたとき、ネットワーク内に輻輳が起きます。例えば、バッファ・サイズが非常に小さい旧 NCP の場合などです。

- ネットワークの信頼性

ネットワークのエラー率が高い場合、ネットワークのスループットが減少して、データの再伝送のためにパフォーマンスの低下を招きます。

第 21 章 CLI/ODBC アプリケーションのパフォーマンス調整

CLI/ODBC とは、データベース・アプリケーションから呼び出し可能な SQL アプリケーション・プログラミング・インターフェースのことです。CLI 関数はシステム・カタログ表にアクセスする DB2 ストアード・プロシージャを呼び出します。

追加の処理で使用するメタデータ情報を集めるため、一部のアプリケーションでは ODBC API が使用されます。次に挙げる 10 種類のメタデータ API を呼び出すことができます。

- SQLTables
- SQLColumns
- SQLSpecialcolumns
- SQLStatistics
- SQLPrimarykeys
- SQLForeignkeys
- SQLTablePrivileges
- SQLColumnPrivileges
- SQLProcedures
- SQLProcedureColumns

上記で一覧にまとめたメタデータ API を使用する特定の CLI/ODBC アプリケーションでは、データベース内のオブジェクトのすべてを照会する場合があります。例えば、SQLTables 呼び出しはデータベースのすべての表のメタデータを要求します。大規模なシステムでは、このような要求はネットワークに大量のトラフィックを生み出し、相当な時間とサーバー・リソースが消費されます。

データベースに最初に接続した後の「情報収集」段階で初期 API 呼び出しが戻すデータの量を制限するために、いくつかの CLI/ODBC 初期化キーワードを使用することができます。これらのキーワードを設定するには、以下のようにします。

1. db2cli.ini ファイルを手動編集する。
2. クライアント構成アシスタント (サポートされているプラットフォームで) を使用して、データベースの ODBC/CLI 設定を変更する。
3. DBA コマンド行インターフェースを使用して、データベース CLI 構成を更新する。

キーワードは、以下のとおりです。

- DBName
- TableType
- SchemaList
- SysSchema
- GrantorList
- GranteeList

第 5 部 トラブルシューティング

第 22 章 トラブルシューティング DB2 Connect

DB2 Connect 環境には、複数のソフトウェア、ハードウェア、および通信製品が含まれます。トラブルシューティングの最も良い方法は、利用できるデータを排除および限定していき、結論 (エラーの発生箇所) に到達することです。

関係のある情報を収集し、あてはまるトピックを選択して行って、参照されたセクションへ進んでください。

関係のある情報の収集

トラブルシューティングには、問題の範囲を絞り込み、考えられる原因を調査することが含まれます。はじめに、関係する情報を収集し、分っている事柄、まだ収集していないデータや、省略できるパスなどの判別を行ってください。少なくとも以下の質問に答えるようにしてください。

- 初期の接続は成功しましたか。
- ハードウェアは正常に機能していますか。
- 通信パスは機能していますか。
- 以前のディレクトリー項目を無効にするような通信ネットワークの変更が行われましたか。
- データベースは始動していますか。
- 通信切断は 1 つ以上のクライアントと DB2 Connect サーバー (ゲートウェイ) の間ですか。DB2 Connect ゲートウェイと IBM メインフレーム・データベース・サーバーの間ですか。それとも DB2 Connect Personal Edition と IBM メインフレーム・データベース・サーバーの間ですか。
- メッセージ内容およびメッセージに戻されているトークンから、何を判別することができますか。
- 現時点での db2trc、db2pd、または db2support などの診断ツールの使用は効果がありそうですか。
- 他のマシンで同じような作業を行っている場合、正しく作動していますか。
- これがリモート・タスクの場合、ローカルで行えば正常に作動しますか。

初期接続が正常に行われなかった場合

以下の質問を検討して、インストール・ステップが正しく行われたか確認してください。

1. インストール・プロセスは正常に完了しましたか。
 - 前提条件のソフトウェア製品はすべて使用可能でしたか。
 - メモリーおよびディスク・スペースは十分ありましたか。
 - リモート・クライアント・サポートはインストールされましたか。
 - 通信ソフトウェアのインストールは、何のエラー状態もなく完了しましたか。

2. UNIX オペレーティング・システムの場合、製品インスタンスは作成されましたか。
 - root として、インスタンス所有者になるユーザーおよび sysadm グループになるグループを作成しましたか。
3. ライセンス情報は正常に処理されましたか (該当する場合)。
 - UNIX オペレーティング・システムの場合、ノード・ロック・ファイルを編集し、IBM が指定したパスワードを入力しましたか。
4. IBM メインフレーム・データベース・サーバーとワークステーションとの通信の構成は正常に行われましたか。
 - 考慮すべき 3 つの構成があります。
 - a. IBM メインフレーム・データベース・サーバーの構成は、サーバーに対するアプリケーション・リクエスターを識別します。IBM メインフレーム・サーバー・データベース管理システムは、そのリクエスターをロケーション、ネットワーク・プロトコル、およびセキュリティーの点で定義するシステム・カタログ項目を持つことになります。
 - b. DB2 Connect ワークステーション構成は、サーバーへのクライアント数を定義し、クライアントへの IBM メインフレーム・サーバーを定義します。
 - c. クライアント・ワークステーション構成では、ワークステーションの名前と通信プロトコルが定義されている必要があります。
 - 初期接続が実行されなかった場合の問題分析には、PU (物理装置) 名が完全であって正しいかどうか確認することが必要です。また、TCP/IP 接続を検査し、正しいポート番号とホスト名を指定しているかどうか確認することが必要です。
 - IBM メインフレーム・サーバー・データベース管理者およびネットワーク管理者の両方は、問題の診断に利用可能なユーティリティーを持っています。
5. IBM メインフレーム・サーバー・データベースを使用するために、IBM メインフレーム・サーバー・データベース管理システムにとって必要な権限レベルを所有していますか。
 - ユーザーのアクセス権限、表修飾子の規則、および予測される結果を考慮してください。
6. IBM メインフレーム・データベース・サーバーに対しコマンド行プロセッサ (CLP) を使用して SQL ステートメントを発行した場合、正常に実行できませんか。
 - コマンド行プロセッサを IBM メインフレーム・データベース・サーバーへバインドする手順を守り行いましたか。

初期接続後に発生する問題

問題の範囲を絞り込むのに役立つ始点として、次の質問が挙げられます。

1. 何か特別なもしくは異常な操作状況が見られますか。
 - それは新しいアプリケーションですか。
 - 新しいプロシージャが使用されていますか。

- システムに影響を与える可能性のある変更が最近行われましたか。例えば、アプリケーションまたはシナリオが正常に実行された後、ソフトウェア製品またはアプリケーションのどれかが変更されましたか。
 - アプリケーション・プログラムについて、そのプログラムを作成するためにどのようなアプリケーション・プログラミング・インターフェース (API) が使用されましたか。
 - そのソフトウェアまたは通信 API を使用した他のアプリケーションが、ユーザーのシステムで実行されたことがありますか。
 - フィックスバックを最近インストールしましたか。オペレーティング・システムにおいて、インストールされて以来使用されていない (もしくはロードされていない) フィーチャーを使用しようとして問題が発生した場合は、IBM の最新のフィックスバックを判別し、そのフィーチャーをインストールした後、それをロードしてください。
2. このエラーは以前にも起こりましたか。
 - 以前のエラー状態には文書化された解決法がありましたか。
 - システムへの参加者はだれでしたか。その人たちが、考えられる処置の方向性にヒントを与えることができませんか。
 3. ネットワークに関する情報を戻す通信ソフトウェア・コマンドを利用して、よく調べてみましたか。
 - TCP/IP コマンド、およびデーモンを使用して取り出した情報には、重要なものが含まれる可能性があります。
 4. *SQLCA* (*SQL* 連絡域) に戻される有用な情報がありますか。
 - 問題処理手順は、*SQLCODE* および *SQLSTATE* の各フィールドの内容を調べるステップを含んでいるはずです。
 - *SQLSTATE* によって、アプリケーション・プログラマーはデータベース製品の *DB2* ファミリーに共通するエラーのクラスについてテストすることができます。分散リレーショナル・データベース・ネットワーク内でこのフィールドは、共通の基礎を提供する場合があります。
 5. *START DBM* を、サーバーで実行しましたか。 加えて、*DB2COMM* 環境変数が、サーバーにリモート・アクセスするクライアント用に正しく設定されているかを確認してください。
 6. 同じ作業を行っている他のマシンは、サーバーに正常に接続することができましたか。サーバーに接続しようとしているクライアントが、最大数に達している場合があります。別のクライアントがサーバーから切断された場合、以前は接続できなかったクライアントは、今は接続することができますか。
 7. マシンは適正なアドレッシングを行っていますか。そのマシンがネットワーク上で固有のものかどうか検査してください。
 8. リモート接続をしている場合、適正な権限がクライアントに認可されていますか。インスタンスへの接続が正常に行われても、データベースや表レベルでは権限が認可されていない場合があります。
 9. これはリモート・データベースに接続する最初の実行マシンですか。分散環境内では、ネットワーク間のルーターやブリッジが、クライアントとサーバーとの通信をブロックする場合があります。例えば、TCP/IP を使用する場合は、リモート・ホストを PING できるかどうか確認してください。

診断ツール

問題が発生したときは、以下のツールを使用することができます。

- ダンプ・ファイル、トラップ・ファイル、エラー・ログ、通知ファイル、およびアラート・ログを含むすべての診断データは、診断データ・ディレクトリー・パス (**diagpath**) データベース・マネージャー構成パラメーターによって指定されたパスにあります。

この構成パラメーターの値が NULL の場合、診断データは以下のいずれかのディレクトリーまたはフォルダーに書き込まれます。

- Linux および UNIX 環境の場合は、`INSTHOME/sql/lib/db2dump` です。
`INSTHOME` はインスタンスのホーム・ディレクトリーです。
- サポートされている Windows 環境の場合
 - **DB2INSTPROF** 環境変数が設定されていない場合、`x:%SQLLIB%\DB2INSTANCE` が使用されます。`x:%SQLLIB` は **DB2PATH** レジストリー変数で指定されたドライブ参照およびディレクトリー、**DB2INSTANCE** の値はインスタンスの名前です。

注: ディレクトリーに `SQLLIB` という名前を付ける必要はありません。

- **DB2INSTPROF** 環境変数が指定されている場合は、`x:%DB2INSTPROF%\DB2INSTANCE` が使用されます。**DB2INSTPROF** はインスタンス・プロファイル・ディレクトリーの名前、**DB2INSTANCE** はインスタンスの名前 (デフォルトでは、Windows 32 ビットオペレーティング・システムの **DB2INSTDEF** の値) です。
- Windows オペレーティング・システムについては、イベント・ビューアーを使用して、管理通知ログを表示します。
- 使用できる診断ツールには、**db2trc**、**db2pd**、**db2support** および **db2diag** が含まれます。
- Linux および UNIX オペレーティング・システムについては、**ps** コマンド。これは活動状態のプロセスについてのプロセス状況情報を標準出力に戻すものです。
- UNIX オペレーティング・システムについては、コア・ファイル。これは重大エラーが起きたとき、現行ディレクトリー内で作成されます。コア・ファイルには、終了したプロセスのメモリー・イメージが入っていて、どの機能がエラーの原因となっているかを判別するのに使用できます。

第 23 章 DB2 Connect での DB2 トレース

環境内で行われているアクションや操作をトレースすると、問題のトラブルシューティングに役立つ情報が得られることがあります。DB2 データベース・サーバー製品内で取られたトレースは、取得、ダンプ、およびフォーマットが可能です。トレース機能は、DB2 データベース・サーバー製品の一部として提供されています。

db2trc を使用した DB2 トレースの取得

db2trc コマンドは、DB2 で提供されるトレース機能を制御します。このトレース機能は、操作に関する情報を記録し、その情報を読み取り可能な形式にフォーマットします。

なお、トレースの実行中は追加のオーバーヘッドが発生するという点に注意してください。このため、トレース機能を使用可能にすると、システムのパフォーマンスに影響が及ぶ可能性があります。

一般に DB2 トレースは、IBM ソフトウェア・サポートおよび開発チームがトラブルシューティングのために使用します。調査中の問題に関するより詳しい情報を得るためにトレースを実行することはできますが、DB2 のソース・コードに関する知識がなければ、その用途はごく限られたものになります。

しかし、それらを取得するよう要求される場合に備えて、トレースをオンにしたりトレース・ファイルをダンプしたりするための正しい方法を理解しておくことは大切です。

注: db2trc を使用するには、SYSADM、SYSCTRL または SYSMAINT 権限のいずれかが必要です。

使用可能なオプションの概要を把握するには、何もパラメーターを指定せずに db2trc コマンドを実行します。

```
C:\>db2trc
Usage: db2trc (chg|clr|dmp|flw|fmt|inf|off|on) options
```

特定の db2trc コマンド・パラメーターについての詳細情報を取得するには、-u オプションを使用します。例えば、トレースをオンにする方法の詳細を調べるには、次のコマンドを実行します。

```
db2trc on -u
```

これにより、DB2 トレースをオンにする際に指定可能なすべての追加オプション ("facilities" というラベルが付いている) に関する情報が提供されます。

トレースをオンにするときに、最も重要なオプションは -L です。このオプションは、トレースした情報を格納するのに使用するメモリー・バッファのサイズを指定します。バッファのサイズはバイト単位またはメガバイト単位で指定できます。(メガバイトを指定するには、値の後ろに "M" または "m" のいずれかを付加します。) トレース・バッファのサイズはメガバイト単位で 2 のべき乗にする必要

があります。この要件を満たしていないサイズを指定すると、バッファのサイズは自動的にメガバイト単位で 2 のべき乗となるうち最も近接した容量に端数切り捨てされます。

バッファが小さすぎると、情報が失われる可能性があります。デフォルトでは、バッファがいっぱいになると最も新しいトレース情報のみが保持されます。バッファが大きすぎると、ファイルを IBM ソフトウェア・サポート・チームに送信するのが難しくなる可能性があります。

比較的短い操作 (データベース接続など) をトレースする場合は、通常約 8 MB 程度で十分です。

```
C:¥> db2trc on -l 8M
トレースはオンになります。
```

しかし、より大きな操作をトレースする場合や、多数の処理が同時に進行するような場合は、より大きなトレース・バッファが必要になります。

トレースは、ほとんどのプラットフォームでいつでもオンにでき、各プラットフォームでの動作についても、上で説明したとおりです。ただし、以下のような注意すべき状況もあります。

1. 複数データベース・パーティション・システムでは、それぞれの物理データベース・パーティション (論理データベース・パーティションではない) に対して、トレースを実行する必要があります。
2. HP-UX、Linux、Solaris のプラットフォームでは、インスタンスの開始後にトレースをオフにすると、トレースを次回開始するときには、サイズの指定に関わりなく、ごく小さなバッファが使用されます。例えば、昨日 `db2trc on -l 8m` でトレースをオンにし、1 つのトレースを収集してから、トレースをオフにします (`db2trc off`)。今日は、インスタンスを終了して再始動する作業を行わず、メモリー・バッファを 32 MB に設定してトレースを実行するとしましょう (`db2trc on -l 32m`)。この場合、トレースはごく小さなバッファを使用します。これらのプラットフォームでトレースを効果的に実行するには、トレースをオンにしてから必要なバッファ・サイズでインスタンスを開始し、それから必要に応じてバッファを「クリア」してください。

DB2 トレース・ファイルのダンプ

ON オプションを使用してトレース機能を使用可能にした後、その後のインスタンスの作業はすべてトレースされます。

トレースが実行されている間は、`clr` オプションを使用してトレース・バッファをクリアできます。トレース・バッファにある既存の情報はすべて削除されます。

```
C:¥>db2trc clr
Trace has been cleared
```

トレースの対象となる操作が終了したら、次のように、`dmp` オプションとトレース・ファイルの名前を指定して、メモリー・バッファをディスクにダンプします。例:

```
C:¥>db2trc dmp trace.dmp
Trace has been dumped to file
```

トレース・バッファをディスクにダンプした後も、トレース機能の実行は継続されます。トレースをオフにするには、OFF オプションを使用します。

```
C:¥>db2trc off
Trace is turned off
```

DB2 トレース・ファイルのフォーマット

db2trc dmp というコマンドによって作成されるダンプ・ファイルは、バイナリー形式であり、読み取り可能ではありません。トレース・ファイルが読み取り可能かどうかを検証するには、バイナリー・トレース・ファイルをフォーマットして、フロー制御を表示し、フォーマットされた出力を NULL デバイスに送信します。

以下の例は、このタスクを実行するコマンドを表示します。

```
db2trc flw example.trc nul
```

example.trc は、dmp オプションを使用して作成されたバイナリー・ファイルです。

このコマンドの出力は、ファイルの読み取りに問題があったかどうか、およびトレースがラップされたかどうかを、明示的に知らせます。

この時点で、ダンプ・ファイルを IBM ソフトウェア・サポートに送信することができます。サポート・チームによって、ご使用の DB2 サービス・レベルに基づいてそのファイルがフォーマットされます。しかし、DB2 サポートに送信する前に、そのダンプ・ファイルを ASCII フォーマットにフォーマットするよう求められる場合があります。これは、flw オプションおよび fmt オプションを使用することで実行されます。バイナリー・ダンプ・ファイルと、作成する ASCII ファイルの名前を次のように指定する必要があります。

```
C:¥>db2trc flw trace.dmp trace.flw
C:¥Temp>db2trc flw trace.dmp trace.flw
Total number of trace records      : 18854
Trace truncated                    : NO
Trace wrapped                      : NO
Number of trace records formatted  : 1513 (pid: 2196 tid 2148 node: -1)
Number of trace records formatted  : 100 (pid: 1568 tid 1304 node: 0)
...
```

```
C:¥>db2trc fmt trace.dmp trace.fmt
C:¥Temp>db2trc fmt trace.dmp trace.fmt
Trace truncated                    : NO
Trace wrapped                      : NO
Total number of trace records      : 18854
Number of trace records formatted  : 18854
```

この出力で "Trace wrapped" が "YES" と表示されていれば、トレース・バッファが小さすぎて、トレース期間の間に収集された情報を全部は格納できなかったこととなります。場合によっては、折り返されたトレースでも特に問題はありません。最も新しい情報（これは、-i オプションが指定されていない限り、保持されているデフォルトの情報を表す）を対象としている場合は、トレース・ファイルに残っている情報だけで十分です。しかし、トレース期間の最初に起こったことを対象としている場合や、その期間に起こったことすべてを対象としている場合は、トレース・バッファのサイズを大きくして操作をやり直すこととなります。

バイナリー・ファイルを実際に読める形のテキスト・ファイルにフォーマットするときに使用できるオプションがあります。例えば、`db2trc fmt -xml trace.dmp trace.fmt` を使用すれば、バイナリー・データを変換し、その結果を XML 構文解析可能フォーマットに出力できます。追加のオプションについては、トレース・コマンド (`db2trc`) の詳細な説明を参照してください。

注意しなければならないことがもう 1 つあります。Linux と UNIX のオペレーティング・システムでは、重大エラーによってインスタンスがシャットダウンされると、DB2 によってトレース・バッファが自動的にディスクにダンプされます。このため、インスタンスの異常終了時にトレースを使用可能にすると、診断ディレクトリーに `db2trdmp.###` (### はデータベース・パーティション番号) という名前のファイルが作成されます。このような動作は、Windows のプラットフォームでは発生しません。その場合には、トレースを手動でダンプする必要があります。

要約すると、`db2trc` コマンドの一般的な順序の例は以下のようになります。

```
db2trc on -l 8M
db2trc clr
<Execute problem recreation commands>
db2trc dump db2trc.dmp
db2trc off
db2trc flw db2trc.dmp <filename>.flw
db2trc fmt db2trc.dmp <filename>.fmt
db2trc fmt -c db2trc.dmp <filename>.fmtc
```

第 24 章 DRDA トレース・ファイル

DRDA トレースを分析する前に、DRDA はデータ構造および通信構造を定義するためのオープン・スタンダードであることを理解しておく必要があります。DRDA は、例えば伝送用データの編成方法、その情報の通信方法などに関するいくつかの規則から成ります。

これらの規則は、以下の解説書で定義されています。

- DRDA V3 第 1 巻: Distributed Relational Database Architecture
- DRDA V3 第 2 巻: Formatted Data Object Content Architecture
- DRDA V3 第 3 巻: Distributed Data Management Architecture

これらの資料の PDF 版は www.opengroup.org から入手できます。

db2drdat ユーティリティーは DRDA アプリケーション・リクエスト (AR) と DB2 DRDA アプリケーション・サーバー (AS) の間でやり取りされるデータを記録します (例えば、DB2 Connect とホストまたは Power Systems サーバーのデータベース・サーバーとの間)。

トレース・ユーティリティー

db2drdat ユーティリティーを利用して、DB2 Connect サーバー (IBM Data Server Clientの代理) と IBM メインフレーム・データベース・サーバーとの間で交換されたデータを記録することができます。

データベース管理者として (またはアプリケーション開発者として)、このデータ・フローがどのように働くかを理解することは有用です。この知識は、特定の問題の原因を判別するのに役立つからです。例えば、CONNECT TO データベース・ステートメントを IBM メインフレーム・データベース・サーバーに対して発行したときに、コマンドが失敗して失敗を示す戻りコードを受け取ったとします。そのとき、どのような情報が IBM メインフレーム・データベース・サーバー管理システムに送られたかを正確に理解していれば、たとえ戻りコードの情報が一般的なものであったとしても、失敗の原因を判別することができます。多くの場合、ユーザーによる単純なエラーが失敗の原因となっています。

db2drdat からの出力は、DB2 Connect ワークステーションと IBM メインフレーム・データベース・サーバー管理システムとの間で交換されたデータ・ストリームをリストします。IBM メインフレーム・データベース・サーバーへ送られたデータには SEND BUFFER とラベル付けされ、IBM メインフレーム・データベース・サーバーから受け取られたデータは RECEIVE BUFFER とラベル付けされます。

受信バッファが SQLCA 情報を含んでいる場合、その後に、このデータの書式化された解釈が続き、SQLCA とラベル付けされます。SQLCA の SQLCODE フィールドは、IBM メインフレーム・データベース・サーバーにより戻されるとおりのマップされていない値です。送信バッファと受信バッファは、最も古いものから最も新しいものへの順でファイル内に配置されます。それぞれのバッファには、以下のものが入ります。

- プロセス ID
- SEND BUFFER、RECEIVE BUFFER、または SQLCA ラベル。バッファ内の 1 番目の DDM コマンドまたはオブジェクトは、DSS TYPE とラベル付けされています。

送信バッファと受信バッファ内の残りのデータは、以下のものを構成する 5 つの列に分けられます。

- バイト・カウント。
- 第 2 列および第 3 列は、2 つのシステム間で交換される DRDA データ・ストリームを ASCII または EBCDIC で表します。
- 第 2 列および第 3 列の ASCII 表示。
- 第 2 列および第 3 列の EBCDIC 表示。

トレース出力

db2drdat ユーティリティーは、*tracefile* に以下の情報を書き込みます。

- -r
 - DRDA 応答/オブジェクトのタイプ
 - 受信バッファ
- -s
 - DRDA 要求のタイプ
 - 送信バッファ
- -c
 - SQLCA
- TCP/IP エラー情報
 - 受信関数の戻りコード
 - 重大度
 - 使用したプロトコル
 - 使用した API
 - 機能
 - エラー番号

注:

1. コマンドの終了コードが、ゼロ値であれば正常に完了したことを、ゼロ以外の値であれば正常に完了しなかったことを示します。
2. 戻されるフィールドは、使用した API によって変わります。
3. 戻されるフィールドは、同じ API の場合でさえ、DB2 Connect が実行しているプラットフォームによって変わります。
4. db2drdat コマンドが既存ファイルへ出力しようとした場合、ファイルのパーミッションが消去を禁止していなければ以前のファイルは消去されてしまいます。

トレース出力ファイルの分析

以下の情報が db2drdat トレースに取り込まれます。

- クライアント・アプリケーションのプロセス ID (PID)
- データベース接続サービス (DCS) ディレクトリーでカタログされた RDB_NAME
- DB2 Connect CCSID (コード化文字セット ID)
- IBM メインフレーム・データベース・サーバーの CCSID。
- DB2 Connect システムの通信相手のホストまたは IBM メインフレーム・データベース・サーバー管理システム。

1 番目のバッファには、IBM メインフレーム・データベース・サーバー管理システムに送信される交換サーバー属性 (EXCSAT) およびアクセス RDB (ACCRDB) コマンドが入っています。そして、それらのコマンドを CONNECT TO データベース・コマンドの結果として送信します。2 番目のバッファには、DB2 Connect が IBM メインフレーム・データベース・サーバー管理システムから受け取る応答が入ります。このバッファには、交換サーバー属性応答データ (EXCSATRD) およびアクセス RDB 応答メッセージ (ACCRDBRM) が入っています。

EXCSAT

EXCSAT コマンドには、サーバー名 (SRVNAM) オブジェクトにより指定されたクライアントのワークステーション名が入っています。そのオブジェクトのコード点は X'116D' であり、DDM 仕様に従っています。EXCSAT コマンドは、1 番目のバッファにあります。EXCSAT コマンドでは、値 X'9481A292' (CCSID 500 によりコード化) は、X'116D' を除去すると、*mask* に変換されます。

EXCSAT コマンドには、EXTNAM (外部名) オブジェクトも含まれます。このオブジェクトは、しばしば、IBM メインフレーム・データベース管理システムについての診断情報に入れられます。それは、20 バイトのアプリケーション ID、続いて 8 バイトのプロセス ID (または 4 バイトのプロセス ID と 4 バイトのスレッド ID) から成ります。それは、コード点 X'115E' で表され、この例ではその値は db2bp で、空白が埋め込まれ、000C50CC へと続きます。Linux または UNIX IBM Data Server Client については、この値は ps コマンドを使用して関連させることができ、このコマンドは活動状態のプロセスについてのプロセス状況情報を標準出力に戻します。

ACCRDB

ACCRDB コマンドは、RDBNAM オブジェクトにある RDB_NAME を含んでいます。そのコード点は X'2110' です。ACCRDB コマンドは、1 番目のバッファの中で EXCSAT コマンドの後に続きます。ACCRDB コマンドでは、値 X'E2E3D3C5C3F1' は、X'2110' を除去すると、STLEC1 に変換されます。これは、DCS ディレクトリーにあるターゲット・データベース名フィールドに対応しています。

アカウンティング・ストリングのコード点は X'2104' です。

DB2 Connect ワークステーション用に構成されたコード・セットは、ACCRDB コマンドの中でコード点が X'119C' である CCSID オブジェクトの CCSIDSBC (1 バイト文字の CCSID) の位置により示されます。この例では、CCSIDSBC は X'0333' です。これは 819 になります。

コード点がそれぞれ X'119D' と X'119E' になっている追加のオブジェクト CCSIDDBC (2 バイト文字の CCSID) と CCSIDMBC (混合バイト文字の CCSID) も、この ACCRDB コマンドに存在します。この例では、CCSIDDBC は X'04B0' (1200)、CCSIDMBC は X'0333' (819) です。

EXCSATRD および ACCRDBRM

また CCSID 値は、IBM メインフレーム・データベース・サーバーから、2 番目のバッファ内にあるアクセス RDB 応答メッセージ (ACCRDBRM) にも戻されます。このバッファには、EXCSATRD とそれに続く ACCRDBRM が入っています。サンプルの出力ファイルには、IBM メインフレーム・データベース・サーバー・システム用の 2 つの CCSID の値が含まれます。値は 1208 (1 バイト文字および混合バイト文字の両方の場合) と 1200 (2 バイト文字の場合) になります。

IBM メインフレーム・データベース・サーバーから戻ってきているコード・ページを DB2 Connect が認識しない場合は、SQLCODE -332 がソースおよびターゲット・コード・ページと共にユーザーに戻されます。DB2 Connect から送信されたコード・セットを IBM メインフレーム・データベース・サーバーが認識しない場合、VALNSPRM (サポートされていないパラメーター値、DDM コード・ポイント X'1252') を戻し、ユーザー用に SQLCODE -332 に変換されます。

ACCRDBRM には、パラメーター PRDID (製品固有 ID、コード点は X'112E') も含まれています。値は、X'C4E2D5F0F8F0F1F5' (EBCDIC では DSN08015) です。標準では、DSN は DB2 for z/OS です。バージョン番号も示されます。ARI は DB2 Server for VSE & VM、SQL は DB2 データベースまたは DB2 Connect、QSQ は DB2 for IBM i です。

トレース出力ファイル・サンプル

以下の図は、出力例を示しており、DB2 Connect ワークステーションとホストまたは System i データベース・サーバーとの間で交換されるいくつかの DRDA データ・ストリームを例示しています。ユーザーの観点からは、コマンド行プロセッサ (CLP) を使用して CONNECT TO データベース・コマンドを実行しています。

171 ページの図 13 は TCP/IP 接続で DB2 Connect Enterprise Edition バージョン 9.1 および DB2 for z/OS バージョン 8 を使用します。

```
1 data DB2 UDB DRDA Communication Manager sqljcsend fnc (3.3.54.5.0.100)
  pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 0 probe 100
  bytes 16
```

```
Data1 (PD_TYPE_UINT,8) unsigned integer:
233
```

```
2 data DB2 UDB DRDA Communication Manager sqljcsend fnc (3.3.54.5.0.1177)
  pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 19532 probe 1177
  bytes 250
```

```
SEND BUFFER(AR):
```

	EXCSAT RQSDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	00C3D041000100BD 1041007F115E8482	...A.....A...^..	.C}.....";db
0010	F282974040404040 4040404040404040	...@@@@@@@@@@@@	2bp
0020	4040F0F0F0C3F5F0 C3C3F0F0F0000000	@@.....	000C50CC000...
0030	0000000000000000 0000000000000000
0040	0000000000000000 000000000060F0F0-00
0050	F0F1A2A495404040 4040404040404040@@@@@@@@	01sun
0060	4040404040404040 4040404040404040	@@@@@@@@@@@@	
0070	C4C5C3E5F8404040 F0A2A49540404040@@@...@@@	DECV8 0sun
0080	4040404040404040 4000181404140300	@@@@@@@@.....
0090	0724070008147400 05240F0008144000	.\$...t.\$...@.
00A0	08000E1147D8C4C2 F261C1C9E7F6F400	...G....a.....QDB2/AIX64.
00B0	08116D9481A29200 0C115AE2D8D3F0F9	..m.....Z.....	.._mask...]SQL09
00C0	F0F0F0	...	000

	ACCSEC RQSDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	0026D00100020020 106D000611A20003	.&.....m.....	..}.....s..
0010	00162110E2E3D3C5 C3F1404040404040	..!.....@@@@@STLEC1
0020	40404040404040	@@@@@	

```
3 data DB2 UDB DRDA Communication Manager sqljcreceive fnc (3.3.54.3.0.100)
  pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 110546200 probe 100
  bytes 12
```

```
Data1 (PD_TYPE_UINT,4) unsigned integer:
105
```

```
4 data DB2 UDB DRDA Communication Manager sqljcreceive fnc (3.3.54.3.0.1178)
  pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 110549755 probe 1178
  bytes 122
```

```
RECEIVE BUFFER(AR):
```

	EXCSATRD OBJDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	0059D04300010053 1443000F115EE5F8	.Y.C...S.C...^..	..}.....;V8
0010	F1C14BE2E3D3C5C3 F100181404140300	..K.....	1A.STLEC1.....
0020	0724070007147400 05240F0007144000	.\$...t.\$...@.
0030	0700081147D8C4C2 F20014116DE2E3D3	...G.....m.....QDB2..._STL
0040	C5C3F14040404040 4040404040000C11	...@@@@@@@@...]	EC1 ...
0050	5AC4E2D5F0F8F0F1 F5	Z.....]DSN08015

	ACCSECRD OBJDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	0010D0030002000A 14AC000611A20003}.....s..

```
5 data DB2 UDB DRDA Communication Manager sqljcsend fnc (3.3.54.5.0.100)
  pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 110656806 probe 100
  bytes 16
```

```
Data1 (PD_TYPE_UINT,8) unsigned integer:
233
```

図 13. トレース出力の例 (TCP/IP 接続)

6 data DB2 UDB DRDA Communication Manager sqljcsend fnc (3.3.54.5.0.1177)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 110659711 probe 1177
 bytes 250

SEND BUFFER(AR):

	SECCHK RQSDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	003CD04100010036	106E000611A20003	.<.A...6.n.....
0010	00162110E2E3D3C5	C3F1404040404040	..!.....@@@...
0020	40404040404000C	11A1D9858799F485	@...@...Regr4e
0030	A599000A11A09585	A6A39695vr....newton

	ACCRDB RQSDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	00ADD001000200A7	20010006210F2407!.\$..}
0010	00172135C7F9F1C1	F0C4F3C14BD7C1F8	..!5.....K...G91A0D3A.PA8
0020	F806030221064600	162110E2E3D3C5C3	...!.F...!.....8.....STLEC
0030	F140404040404040	4040404040000C11	.@...@...@...@...1
0040	2EE2D8D3F0F9F0F0	F000D002FD8E3C4/... .SQL09000....QTD
0050	E2D8D3C1E2C30016	00350006119C03335.....3 SQLASC.....
0060	0006119D04B00006	119E0333003C21043.

7 data DB2 UDB DRDA Communication Manager sqljcreceive fnc (3.3.54.3.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 259908001 probe 100
 bytes 12

Data1 (PD_TYPE_UINT,4) unsigned integer:
 176

8 data DB2 UDB DRDA Communication Manager sqljcreceive fnc (3.3.54.3.0.1178)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 259911584 probe 1178
 bytes 193

RECEIVE BUFFER(AR):

	SECCHKRM RPYDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	0015D0420001000F	1219000611490000	...B.....I..}
0010	000511A400	u.

	ACCRDBRM RPYDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	009BD00200020095	2201000611490000"....I..}
0010	000D002FD8E3C4E2	D8D3F3F7F0000C11	.../.....QTDSQL370...
0020	2EC4E2D5F0F8F0F1	F5001600350006115... .DSN08015.....
0030	9C04B80006119E04	B80006119D04B000
0040	0C11A0D5C5E6E3D6	D540400006212524@...!%\$...NEWTON
0050	34001E244E000624	4C00010014244D00	4..\$.N..\$.L...\$.M.+...<.....(.
0060	06244FFFFF000A11	E8091E768301BE00	.\$O.....v.....!.....Y....c...
0070	2221030000000005	68B3B8C7F9F1C1F0	"!.....h.....G91A0
0080	C4F3C1D7C1F8F840	4040400603022106@...!. D3APA88
0090	46000A11E8091E76	831389	F.....v....Y....c.i

9 data DB2 UDB DRDA Communication Manager sqljcsend fnc (3.3.54.5.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 2 nsec 364420503 probe 100
 bytes 16

Data1 (PD_TYPE_UINT,8) unsigned integer:
 10

図 14. トレース出力の例 (TCP/IP 接続) (続き)

10 data DB2 UDB DRDA Communication Manager sqljcsend fnc (3.3.54.5.0.1177)
 pid 807116 tid 1 cpid -1 node 0 sec 2 nsec 364440751 probe 1177
 bytes 27

SEND BUFFER(AR):

	RDBCMM RQSDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	000AD00100010004 200E}.....

11 data DB2 UDB DRDA Communication Manager sqljcreceive fnc (3.3.54.3.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 2 nsec 475009631 probe 100
 bytes 12

Data1 (PD_TYPE_UINT,4) unsigned integer:
 54

12 data DB2 UDB DRDA Communication Manager sqljcreceive fnc (3.3.54.3.0.1178)
 pid 807116 tid 1 cpid -1 node 0 sec 2 nsec 475014579 probe 1178
 bytes 71

RECEIVE BUFFER(AR):

	ENDUOWRM RPYDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	002BD05200010025 220C000611490004	..+.R...%"....I..	..}.....
0010	00162110E2E3D3C5 C3F1404040404040	..!.....@#####STLEC1
0020	4040404040400005 211501	@#####!...

	SQLCARD OBJDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	000BD00300010005 2408FF\$..	..}.....

13 data DB2 UDB DRDA Communication Manager sqljcsend fnc (3.3.54.5.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 721710319 probe 100
 bytes 16

Data1 (PD_TYPE_UINT,8) unsigned integer:
 126

14 data DB2 UDB DRDA Communication Manager sqljcsend fnc (3.3.54.5.0.1177)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 721727276 probe 1177
 bytes 143

SEND BUFFER(AR):

	EXCSQLIMM RQSDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	0053D0510001004D 200A00442113E2E3	.S.Q...M ..D!...	..}....(.....ST
0010	D3C5C3F140404040 4040404040404040#####	LEC1
0020	D5E4D3D3C9C44040 4040404040404040#####	NULLID
0030	4040E2D8D3C3F2C6 F0C1404040404040	@.....#####	SQLC2F0A
0040	4040404041414141 41484C5600CB0005	@#####AHLV....<.....
0050	2105F1	!..	..1

	SQLSTT OBJDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	002BD00300010025 2414000000001B64	..+.....%\$.....d	..}.....
0010	656C657465206672 6F6D206464637375	elete from ddcsu	..%.....?_.....
0020	73312E6D79746162 6C65FF	sl.mytable.	..._`./.%..

15 data DB2 UDB DRDA Communication Manager sqljcreceive fnc (3.3.54.3.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 832901261 probe 100
 bytes 12

Data1 (PD_TYPE_UINT,4) unsigned integer:
 102

図 15. トレース出力の例 (TCP/IP 接続) (続き)

16 data DB2 UDB DRDA Communication Manager sqljcReceive fnc (3.3.54.3.0.1178)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 832906528 probe 1178
 bytes 119

RECEIVE BUFFER(AR):

	SQLCARD OBJDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	0066D00300010060 240800FFFFFF3434	.f.....`\$.44	..}....-.....
0010	3237303444534E58 4F544C2000FFFFFFE	2704DSNXOTL+!.<.....
0020	0C00000000000000 00FFFFFFF000000
0030	00000000000572020 2057202020202020W W
0040	001053544C454331 2020202020202020	..STLEC1<.....
0050	2020000F44444353 5553312E4D595441	..DDCSUS1.MYTA(...
0060	424C450000FF	BLE...<.....

17 data DB2 UDB DRDA Communication Manager sqljcSend fnc (3.3.54.5.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 833156953 probe 100
 bytes 16

Data1 (PD_TYPE_UINT,8) unsigned integer:
 10

18 data DB2 UDB DRDA Communication Manager sqljcSend fnc (3.3.54.5.0.1177)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 833159843 probe 1177
 bytes 27

SEND BUFFER(AR):

	RDBRLLBCK RQSDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	000AD00100010004 200F}.....

19 data DB2 UDB DRDA Communication Manager sqljcReceive fnc (3.3.54.3.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 943302832 probe 100
 bytes 12

Data1 (PD_TYPE_UINT,4) unsigned integer:
 54

20 data DB2 UDB DRDA Communication Manager sqljcReceive fnc (3.3.54.3.0.1178)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 943306288 probe 1178
 bytes 71

RECEIVE BUFFER(AR):

	ENDUOWRM RPYDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	002BD05200010025 220C000611490004	..+R...%"....I..	..}.....
0010	00162110E2E3D3C5 C3F1404040404040	..!.....@@@...STLEC1
0020	4040404040400005 211502	@@@@@@...!..

	SQLCARD OBJDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	000BD00300010005 2408FF\$..	..}.....

図 16. トレース出力の例 (TCP/IP 接続) (続き)

DRDA トレースの後続のバッファ情報

後続の送信バッファと受信バッファを分析して、追加情報を得ることができます。次の要求はコミットを含んでいます。 `commit` コマンドは、IBM メインフレーム・データベース・サーバー管理システムが現行の作業単位をコミットするよう命令します。4番目のバッファは、IBM メインフレーム・データベース・サーバーのデータベース管理システムから、コミットまたはロールバックの結果として受け取られます。そこには最終作業単位の応答メッセージ (ENDUOWRM) が含まれ、それは現行の作業単位が終了したことを示します。

この例のトレース・エントリー 12 は、DDM コード点 X'2408' とそれに続く X'FF' が示しているとおりに、NULL の SQLCA を含んでいます。NULL の SQLCA (X'2408FF') は、成功 (SQLCODE 0) を示しています。

171 ページの図 13 は、トレース・エントリー 16 にエラー SQLCA を含んだ受信バッファの例を示しています。

第 6 部 メッセージ

第 25 章 一般的な DB2 Connect の問題

このトピックでは、DB2 Connect の使用時に接続問題が生じたときの最も一般的な症状をリストします。 どの場合でも、以下の形式で示されます。

- 表示されたメッセージに関連した、メッセージ番号と戻りコード (またはプロトコル固有の戻りコード) の組み合わせ。各メッセージと戻りコードの組み合わせには、個別の見出しがあり、この見出しはメッセージ番号順、その後に戻りコードの順で並べられます。
- 症状。通常は、サンプル・メッセージのリスト形式で示されます。
- 解決方法。エラーの推定原因が示されます。場合によっては、複数の解決方法が提示されることがあります。

SQL0965 または SQL0969

症状 DB2 for IBM i、DB2 for z/OS、および DB2 Server for VM and VSE からさまざまな異なる戻りコードとともに、メッセージ SQL0965 および SQL0969 が発行されます。

いずれかのメッセージが出された場合、そのメッセージを発行したデータベース・サーバー製品の資料で、元の SQL コードを調べる必要があります。

解決方法

IBM メインフレーム・データベースから受信された SQL コードを変換できません。 そのエラー・コードに基づいて問題を訂正してから、失敗したコマンドを再サブミットしてください。

SQL5043N

症状 1 つまたは複数の通信プロトコルに対するサポートが正常に開始できませんでした。ただし、コアとなるデータベース・マネージャーの機能は正常に開始されました。

おそらく、TCP/IP プロトコルが DB2 Connect サーバーで開始されていません。以前に成功したクライアント接続がまだ残っている可能性があります。

diaglevel = 4 の場合、db2diag ログ・ファイルに同様の項目が含まれている可能性があります。例えば次のようになります。

```
2001-05-30-14.09.55.321092 Instance:svtdbm5 Node:000
PID:10296(db2tcpm) Appid:none
common_communication sqlcctcpconnmgr_child Probe:46
DIA3205E Socket address "30090" configured in the TCP/IP
services file and
required by the TCP/IP server support is being used by another
process.
```

解決方法

この警告は、DB2 Connect (リモート・クライアントのサーバーとして動作している) が 1 つまたは複数のクライアント通信プロトコルを処理する際に問題が発生していることを示しています。これらのプロトコルは TCP/IP

およびその他のものであり、このメッセージは通常、DB2 Connect に定義されているこれらの通信プロトコルのいずれかが正しく構成されていないことを示しています。

DB2COMM プロファイル変数が定義されていないか、不正に定義されていることが原因かもしれません。一般に、問題は DB2COMM 変数とデータベース・マネージャー構成で定義した名前 (例えば、svccname または nname) との間のミスマッチの結果です。

可能性のあるシナリオとしては、以前は接続に成功しており、構成も変更していないのに、SQL5043 エラー・メッセージを受け取ってしまったというものです。これは、TCP/IP プロトコルの使用時にリモート・システムが何らかの理由で接続を異常終了したときに発生する可能性があります。これが発生した場合、接続はクライアント上にまだ存続しているように見えることがあり、下記のコマンドを実行することによってさらなる介入なしで、接続をリストアすることができます。

一番多いのは、DB2 Connect サーバーに接続しているクライアントの 1 つが TCP/IP ポート上でハンドルを持ったままになるというケースです。DB2 Connect サーバーに接続している各クライアント・マシン上で、以下のコマンドを入力します。

```
db2 terminate
db2stop
```

SQL30020

症状 SQL30020N 実行が、後続のコマンドおよび SQL ステートメントの正常な実行に影響を与える分散プロトコル・エラーのために失敗しました。

解決方法

このエラーが発生したら、サービス担当者に連絡してください。サービス担当者に連絡する前に db2support コマンドを実行します。

SQL30060

症状 SQL30060N "<authorization-ID>" が、処理 "<operation>" を実行する権限を持っていません。

解決方法

DB2 for z/OS への接続時に、コミュニケーション・データベース (CDB) 表が正しく更新されていません。

SQL30061

症状 誤った IBM メインフレーム・データベース・サーバー・ロケーションに接続しています。ターゲット・データベースが見つかりません。

解決方法

DCS ディレクトリー項目に誤ったサーバー・データベース名を指定した可能性があります。これが生じた場合、SQLCODE -30061 がアプリケーションに戻されます。

DB2 ノード、データベース、および DCS ディレクトリー項目を調べてください。DCS ディレクトリー項目のターゲット・データベース名のフィールドは、プラットフォームに基づいたデータベースの名前に対応していません。

ればなりません。例えば、DB2 for z/OS データベースの場合、使用する名前はブートストラップ・データ・セット (BSDS) の

『LOCATION=locname』フィールドで使用した名前と同じでなければなりません。これは、分散データ機能 (DDF) を開始するときの DSNL004I メッセージにも示されています (LOCATION=location)。

TCP/IP ノードへの正しいコマンドは次のとおりです。

```
db2 catalog tcpip node <node_name> remote <host_name_or_address>
server <port_no_or_service_name>
db2 catalog dcs database <local_name> as <real_db_name>
db2 catalog database <local_name> as <alias> at <node node_name>
authentication server
```

その後データベースへ接続するには、次のコマンドを実行します。

```
db2 connect to <alias> user <user_name> using <password>
```

SQL30081N (戻りコード 79)

症状

```
SQL30081N A communication error has been detected.
Communication protocol
being used: "TCP/IP". Communication API being used: "SOCKETS".
Location
where the error was detected: "". Communication function
detecting the error:
"connect". Protocol specific error code(s): "79", "*", "*".
SQLSTATE=08001
```

解決方法

このエラーは、リモート・クライアントが DB2 Connect サーバーへの接続に失敗した場合に発生する可能性があります。さらに、DB2 Connect サーバーから IBM メインフレーム・データベース・サーバーへの接続時にも発生することがあります。

1. DB2COMM プロファイル変数が、DB2 Connect サーバーで正しく設定されていない可能性があります。このことを確認してください。例えば、AIX で DB2 Enterprise Server Edition を実行している場合は、コマンド `db2set db2comm=tcpip` は `sqlllib/db2profile` に存在していなければなりません。
2. TCP/IP のサービス名およびポート番号の指定が、IBM Data Server Client と DB2 Connect サーバーで一致しない場合があります。両方のマシンで、TCP/IP services ファイル内の項目を確認してください。
3. DB2 Connect サーバーで DB2 が開始していることをチェックします。次のコマンドを使用して、データベース・マネージャー構成の `diaglevel` を 4 に設定してください。

```
db2 update dbm cfg using diaglevel 4
```

DB2 を停止して再始動したら、`db2diag` ログ・ファイルを参照して、DB2 TCP/IP 通信が開始していることをチェックします。次のような出力が含まれているはずですが。

```
2001-02-03-12.41.04.861119 Instance:svtdbm2 Node:00
PID:86496(db2sysc) Appid:none
common_communication sqlcctcp_start_listen Probe:80
DIA3000I "TCP/IP" protocol support was successfully started.
```

SQL30081N (プロトコル固有のエラー・コード 10032)

症状

```
SQL30081N A communication error has been detected.  
Communication protocol  
being used: "TCP/IP". Communication API being used: "SOCKETS".  
Location  
where the error was detected: "9.21.85.159". Communication  
function detecting  
the error: "send". Protocol specific error code(s): "10032",  
"*", "*".  
SQLSTATE=08001
```

解決方法

このエラー・メッセージは、TCP/IP 通信に失敗したマシンから切断しようとするときに受け取ることがあります。TCP/IP サブシステムの問題を修正してください。

ほとんどのマシンでは、単にそのマシンの TCP/IP プロトコルを再始動すれば問題は修正されます。時には、マシン全体を再生しなければならないこともあります。

CONNECT 時の SQL30082 RC=24

症状 SQLCODE -30082 指定されたユーザー名またはパスワードが正しくありません。

解決方法

CONNECT ステートメントにパスワードを指定する必要がある場合、正しく指定されているか確認してください。ターゲット・サーバー・データベースへ送信するときに、パスワードが使用できません。パスワードを IBM Data Server Client からターゲット・サーバー・データベースに送信する必要があります。特定のプラットフォーム、例えば AIX などでは、パスワードは CONNECT ステートメントに指定してある場合に限り、入手することができます。

第 7 部 付録

付録 A. DB2 技術情報の概説

DB2 技術情報は、以下のツールと方法を介して利用できます。

- DB2 インフォメーション・センター
 - トピック (タスク、概念、およびリファレンス・トピック)
 - DB2 ツールのヘルプ
 - サンプル・プログラム
 - チュートリアル
- DB2 資料
 - PDF ファイル (ダウンロード可能)
 - PDF ファイル (DB2 PDF DVD に含まれる)
 - 印刷資料
- コマンド行ヘルプ
 - コマンド・ヘルプ
 - メッセージ・ヘルプ

注: DB2 インフォメーション・センターのトピックは、PDF やハードコピー資料よりも頻繁に更新されます。最新の情報を入手するには、資料の更新が発行されたときにそれをインストールするか、ibm.com にある DB2 インフォメーション・センターを参照してください。

技術資料、ホワイト・ペーパー、IBM Redbooks 資料などのその他の DB2 技術情報には、オンライン (ibm.com) でアクセスできます。DB2 Information Management ソフトウェア・ライブラリー・サイト (<http://www.ibm.com/software/data/sw-library/>) にアクセスしてください。

資料についてのフィードバック

DB2 の資料についてのお客様からの貴重なご意見をお待ちしています。DB2 の資料を改善するための提案については、db2docs@ca.ibm.com まで E メールを送信してください。DB2 の資料チームは、お客様からのフィードバックすべてに目を通しますが、直接お客様に返答することはありません。お客様が関心をお持ちの内容について、可能な限り具体的な例を提供してください。特定のトピックまたはヘルプ・ファイルについてのフィードバックを提供する場合は、そのトピック・タイトルおよび URL を含めてください。

DB2 お客様サポートに連絡する場合には、この E メール・アドレスを使用しないでください。資料を参照しても、DB2 の技術的な問題が解決しない場合は、お近くの IBM サービス・センターにお問い合わせください。

DB2 テクニカル・ライブラリー (ハードコピーまたは PDF 形式)

以下の表は、DB2 ライブラリーについて説明しています。DB2 ライブラリーに関する詳細な説明については、www.ibm.com/shop/publications/order にある IBM Publications Center にアクセスしてください。英語および翻訳された DB2 バージョン 9.7 のマニュアル (PDF 形式) は、www.ibm.com/support/docview.wss?rs=71&uid=swg2700947 からダウンロードできます。

この表には印刷資料が入手可能かどうかを示されていますが、国または地域によっては入手できない場合があります。

資料番号は、資料が更新される度に大きくなります。資料を参照する際は、以下にリストされている最新版であることを確認してください。

注: DB2 インフォメーション・センターは、PDF やハードコピー資料よりも頻繁に更新されます。

表 20. DB2 の技術情報

資料名	資料番号	印刷資料が入手可能かどうか	最終更新
管理 API リファレンス	SC88-5883-01	入手可能	2009 年 11 月
管理ルーチンおよびビュー	SC88-5880-01	入手不可	2009 年 11 月
コール・レベル・イン ターフェース ガイドお よびリファレンス 第 1 巻	SC88-5885-01	入手可能	2009 年 11 月
コール・レベル・イン ターフェース ガイドお よびリファレンス 第 2 巻	SC88-5886-01	入手可能	2009 年 11 月
コマンド・リファレン ス	SC88-5884-01	入手可能	2009 年 11 月
データ移動ユーティリ ティー ガイドおよびリ ファレンス	SC88-5903-00	入手可能	2009 年 8 月
データ・リカバリーと 高可用性 ガイドおよび リファレンス	SC88-5904-01	入手可能	2009 年 11 月
データベース: 管理の 概念および構成リファ レンス	SC88-5870-01	入手可能	2009 年 11 月
データベースのモニタ リング ガイドおよびリ ファレンス	SC88-5872-01	入手可能	2009 年 11 月
データベース・セキュ リティ・ガイド	SC88-5905-01	入手可能	2009 年 11 月

表 20. DB2 の技術情報 (続き)

資料名	資料番号	印刷資料が入手可能 かどうか	最終更新
DB2 Text Search ガイド	SC88-5902-01	入手可能	2009 年 11 月
ADO.NET および OLE DB アプリケーションの開発	SC88-5874-01	入手可能	2009 年 11 月
組み込み SQL アプリケーションの開発	SC88-5875-01	入手可能	2009 年 11 月
Java アプリケーションの開発	SC88-5878-01	入手可能	2009 年 11 月
Perl、PHP、Python および Ruby on Rails アプリケーションの開発	SC88-5879-00	入手不可	2009 年 8 月
SQL および外部ルーチンの開発	SC88-5876-01	入手可能	2009 年 11 月
データベース・アプリケーション開発の基礎	GI88-4201-01	入手可能	2009 年 11 月
DB2 インストールおよび管理 概説 (Linux および Windows 版)	GI88-4202-00	入手可能	2009 年 8 月
グローバリゼーション・ガイド	SC88-5906-00	入手可能	2009 年 8 月
DB2 サーバー機能 インストール	GC88-5888-01	入手可能	2009 年 11 月
IBM データ・サーバー・クライアント機能 インストール	GC88-5889-00	入手不可	2009 年 8 月
メッセージ・リファレンス 第 1 巻	SC88-5897-00	入手不可	2009 年 8 月
メッセージ・リファレンス 第 2 巻	SC88-5898-00	入手不可	2009 年 8 月
Net Search Extender 管理およびユーザーズ・ガイド	SC88-5901-01	入手不可	2009 年 11 月
パーティションおよびクラスタリングのガイド	SC88-5907-01	入手可能	2009 年 11 月
pureXML ガイド	SC88-5895-01	入手可能	2009 年 11 月
Query Patroller 管理およびユーザーズ・ガイド	SC88-5908-00	入手不可	2009 年 8 月

表 20. DB2 の技術情報 (続き)

資料名	資料番号	印刷資料が入手可能 かどうか	最終更新
<i>Spatial Extender</i> および <i>Geodetic Data</i> <i>Management Feature</i> ユ ーザーズ・ガイドおよ びリファレンス	SC88-5900-00	入手不可	2009 年 8 月
<i>SQL</i> プロシージャ言 語: アプリケーション のイネーブルメントお よびサポート	SC88-5877-01	入手可能	2009 年 11 月
<i>SQL</i> リファレンス 第 1 巻	SC88-5881-01	入手可能	2009 年 11 月
<i>SQL</i> リファレンス 第 2 巻	SC88-5882-01	入手可能	2009 年 11 月
問題判別およびデータ ベース・パフォーマンス のチューニング	SC88-5871-01	入手可能	2009 年 11 月
<i>DB2</i> バージョン 9.7 へ のアップグレード	SC88-5887-01	入手可能	2009 年 11 月
<i>Visual Explain</i> チュー リアル	SC88-5899-00	入手不可	2009 年 8 月
<i>DB2</i> バージョン 9.7 の 新機能	SC88-5893-01	入手可能	2009 年 11 月
ワークロード・マネー ジャー ガイドおよびリ ファレンス	SC88-5894-01	入手可能	2009 年 11 月
<i>XQuery</i> リファレンス	SC88-5896-01	入手不可	2009 年 11 月

表 21. DB2 Connect 固有の技術情報

資料名	資料番号	印刷資料が入手可能 かどうか	最終更新
<i>DB2 Connect Personal</i> <i>Edition</i> インストールお よび構成	SC88-5891-01	入手可能	2009 年 11 月
<i>DB2 Connect</i> サーバー 機能 インストールおよ び構成	SC88-5892-01	入手可能	2009 年 11 月
<i>DB2 Connect</i> ユーザー ズ・ガイド	SC88-5890-01	入手可能	2009 年 11 月

表 22. Information Integration の技術情報

資料名	資料番号	印刷資料が入手可能かどうか	最終更新
Information Integration: フェデレーテッド・システム管理ガイド	SC88-4166-02	入手可能	2009 年 8 月
Information Integration: レプリケーションおよびイベント・パブリッシングのための ASNCLP プログラム・リファレンス	SC88-4167-04	入手可能	2009 年 8 月
Information Integration: フェデレーテッド・データ・ソース構成ガイド	SC88-4185-02	入手不可	2009 年 8 月
Information Integration: SQL レプリケーションガイドとリファレンス	SC88-4168-02	入手可能	2009 年 8 月
Information Integration: レプリケーションとイベント・パブリッシング 概説	GC88-4187-02	入手可能	2009 年 8 月

DB2 の印刷資料の注文方法

DB2 の印刷資料が必要な場合、オンラインで購入することができますが、すべての国および地域で購入できるわけではありません。DB2 の印刷資料については、IBM 営業担当員にお問い合わせください。DB2 PDF ドキュメンテーション DVD の一部のソフトコピー・ブックは、印刷資料では入手できないことに留意してください。例えば、「DB2 メッセージ・リファレンス」はどちらの巻も印刷資料としては入手できません。

DB2 PDF ドキュメンテーション DVD で利用できる DB2 の印刷資料の大半は、IBM に有償で注文することができます。国または地域によっては、資料を IBM Publications Center からオンラインで注文することもできます。お客様の国または地域でオンライン注文が利用できない場合、DB2 の印刷資料については、IBM 営業担当員にお問い合わせください。DB2 PDF ドキュメンテーション DVD に収録されている資料の中には、印刷資料として提供されていないものもあります。

注: 最新で完全な DB2 資料は、DB2 インフォメーション・センター (<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7>) で参照することができます。

DB2 の印刷資料は以下の方法で注文することができます。

- 日本 IBM 発行のマニュアルはインターネット経由でご購入いただけます。詳しくは <http://www.ibm.com/shop/publications/order> をご覧ください。資料の注文情報にアクセスするには、お客様の国、地域、または言語を選択してください。その後、各ロケーションにおける注文についての指示に従ってください。

- DB2 の印刷資料を IBM 営業担当員に注文するには、以下のようになります。
 1. 以下の Web サイトのいずれかから、営業担当員の連絡先情報を見つけてください。
 - IBM Directory of world wide contacts (www.ibm.com/planetwide)
 - IBM Publications Web サイト (<http://www.ibm.com/shop/publications/order>)。国、地域、または言語を選択し、お客様の所在地に該当する Publications ホーム・ページにアクセスしてください。このページから、「このサイトについて」のリンクにアクセスしてください。
 2. 電話をご利用の場合は、DB2 資料の注文であることをご指定ください。
 3. 担当者に、注文する資料のタイトルと資料番号をお伝えください。タイトルと資料番号は、186 ページの『DB2 テクニカル・ライブラリー (ハードコピーまたは PDF 形式)』でご確認いただけます。

コマンド行プロセッサから SQL 状態ヘルプを表示する

DB2 製品は、SQL ステートメントの結果の原因になったと考えられる条件の SQLSTATE 値を戻します。SQLSTATE ヘルプは、SQL 状態および SQL 状態クラス・コードの意味を説明します。

SQL 状態ヘルプを開始するには、コマンド行プロセッサを開いて以下のように入力します。

```
? sqlstate or ? class code
```

ここで、*sqlstate* は有効な 5 桁の SQL 状態を、*class code* は SQL 状態の最初の 2 桁を表します。

例えば、? 08003 を指定すると SQL 状態 08003 のヘルプが表示され、? 08 を指定するとクラス・コード 08 のヘルプが表示されます。

異なるバージョンの DB2 インフォメーション・センターへのアクセス

DB2 バージョン 9.7 のトピックを扱っている DB2 インフォメーション・センターの URL は、<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/> です。

DB2 バージョン 9.5 のトピックを扱っている DB2 インフォメーション・センターの URL は、<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/> です。

DB2 バージョン 9.1 のトピックを扱っている DB2 インフォメーション・センターの URL は、<http://publib.boulder.ibm.com/infocenter/db2luw/v9/> です。

DB2 バージョン 8 のトピックについては、DB2 インフォメーション・センターの URL <http://publib.boulder.ibm.com/infocenter/db2luw/v8/> にアクセスしてください。

DB2 インフォメーション・センターでの希望する言語でのトピックの表示

DB2 インフォメーション・センターでは、ブラウザの設定で指定した言語でのトピックの表示が試みられます。トピックがその指定言語に翻訳されていない場合は、DB2 インフォメーション・センターでは英語でトピックが表示されます。

- Internet Explorer Web ブラウザーで、指定どおりの言語でトピックを表示するには、以下のようにします。
 1. Internet Explorer の「ツール」 -> 「インターネット オプション」 -> 「言語 ...」 ボタンをクリックします。「言語の優先順位」ウィンドウがオープンします。
 2. 該当する言語が、言語リストの先頭の項目に指定されていることを確認します。
 - リストに新しい言語を追加するには、「追加...」 ボタンをクリックします。

注: 言語を追加しても、特定の言語でトピックを表示するのに必要なフォントがコンピューターに備えられているとはかぎりません。

 - リストの先頭に新しい言語を移動するには、その言語を選択してから、その言語が言語リストに先頭に行くまで「上に移動」 ボタンをクリックします。
 3. ページを最新表示します。希望する言語で DB2 インフォメーション・センターが表示されます。
- Firefox または Mozilla Web ブラウザーの場合に、希望する言語でトピックを表示するには、以下のようにします。
 1. 「ツール」 -> 「オプション」 -> 「詳細」 ダイアログの「言語」セクションにあるボタンを選択します。「設定」ウィンドウに「言語」パネルが表示されます。
 2. 該当する言語が、言語リストの先頭の項目に指定されていることを確認します。
 - リストに新しい言語を追加するには、「追加...」 ボタンをクリックしてから、「言語を追加」ウィンドウで言語を選択します。
 - リストの先頭に新しい言語を移動するには、その言語を選択してから、その言語が言語リストに先頭に行くまで「上に移動」 ボタンをクリックします。
 3. ページを最新表示します。希望する言語で DB2 インフォメーション・センターが表示されます。

ブラウザとオペレーティング・システムの組み合わせによっては、オペレーティング・システムの地域の設定も希望のロケールと言語に変更しなければなりません。

コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの更新

ローカルにインストールされた DB2 インフォメーション・センターは、定期的に更新する必要があります。

始める前に

DB2 バージョン 9.7 インフォメーション・センターが既にインストールされている必要があります。詳しくは、「DB2 サーバー機能 インストール」の『DB2 セットアップ・ウィザードによる DB2 インフォメーション・センターのインストール』

のトピックを参照してください。インフォメーション・センターのインストールに適用されるすべての前提条件と制約事項は、インフォメーション・センターの更新にも適用されます。

このタスクについて

既存の DB2 インフォメーション・センターは、自動で更新することも。手動で更新することもできます。

- 自動更新 - 既存のインフォメーション・センターのフィーチャーと言語を更新します。自動更新を使用すると、更新中にインフォメーション・センターが使用できなくなる時間が最小限で済むというメリットもあります。さらに、自動更新は、定期的に行う他のバッチ・ジョブの一部として実行されるように設定することができます。
- 手動更新 - 更新処理中にフィーチャーまたは言語を追加する場合に使用する必要があります。例えば、ローカルのインフォメーション・センターが最初は英語とフランス語でインストールされており、その後ドイツ語もインストールすることにした場合、手動更新でドイツ語をインストールし、同時に、既存のインフォメーション・センターのフィーチャーおよび言語を更新できます。しかし、手動更新ではインフォメーション・センターを手動で停止、更新、再始動する必要があります。更新処理の間はずっと、インフォメーション・センターは使用できなくなります。

手順

このトピックでは、自動更新のプロセスを詳しく説明しています。手動更新の手順については、『コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの手動更新』のトピックを参照してください。

コンピューターまたはイントラネット・サーバーにインストールされている DB2 インフォメーション・センターを自動で更新するには、次のようにします。

1. Linux オペレーティング・システムの場合、次のようにします。
 - a. インフォメーション・センターがインストールされているパスにナビゲートします。デフォルトでは、DB2 インフォメーション・センターは、`/opt/ibm/db2ic/V9.7` ディレクトリーにインストールされています。
 - b. インストール・ディレクトリーから `doc/bin` ディレクトリーにナビゲートします。
 - c. 次のように `ic-update` スクリプトを実行します。

```
ic-update
```
2. Windows オペレーティング・システムの場合、次のようにします。
 - a. コマンド・ウィンドウを開きます。
 - b. インフォメーション・センターがインストールされているパスにナビゲートします。デフォルトでは、DB2 インフォメーション・センターは、`<Program Files>%IBM%DB2 Information Center%Version 9.7` ディレクトリーにインストールされています (`<Program Files>` は「Program Files」ディレクトリーのロケーション)。

- c. インストール・ディレクトリーから doc¥bin ディレクトリーにナビゲートします。
- d. 次のように ic-update.bat ファイルを実行します。

```
ic-update.bat
```

結果

DB2 インフォメーション・センターが自動的に再始動します。更新が入手可能な場合、インフォメーション・センターに、更新された新しいトピックが表示されます。インフォメーション・センターの更新が入手可能でなかった場合、メッセージがログに追加されます。ログ・ファイルは、doc¥eclipse¥configuration ディレクトリーにあります。ログ・ファイル名はランダムに生成された名前です。例えば、1239053440785.log のようになります。

コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの手動更新

DB2 インフォメーション・センターをローカルにインストールしている場合は、IBM から資料の更新を入手してインストールすることができます。

このタスクについて

ローカルにインストールされた *DB2* インフォメーション・センター を手動で更新するには、以下のことを行う必要があります。

1. コンピューター上の *DB2* インフォメーション・センター を停止し、インフォメーション・センターをスタンドアロン・モードで再始動します。インフォメーション・センターをスタンドアロン・モードで実行すると、ネットワーク上の他のユーザーがそのインフォメーション・センターにアクセスできなくなります。これで、更新を適用できるようになります。*DB2* インフォメーション・センターのワークステーション・バージョンは、常にスタンドアロン・モードで実行されます。を参照してください。
2. 「更新」機能を使用することにより、どんな更新が利用できるかを確認します。インストールしなければならない更新がある場合は、「更新」機能を使用してそれを入手およびインストールできます。

注: ご使用の環境において、インターネットに接続されていないマシンに *DB2* インフォメーション・センター の更新をインストールする必要がある場合、インターネットに接続されていて *DB2* インフォメーション・センター がインストールされているマシンを使用して、更新サイトをローカル・ファイル・システムにミラーリングしてください。ネットワーク上の多数のユーザーが資料の更新をインストールする場合にも、更新サイトをローカルにミラーリングして、更新サイト用のプロキシを作成することにより、個々のユーザーが更新を実行するのに要する時間を短縮できます。

更新パッケージが入手可能な場合、「更新」機能を使用してパッケージを入手します。ただし、「更新」機能は、スタンドアロン・モードでのみ使用できます。

3. スタンドアロンのインフォメーション・センターを停止し、コンピューター上の *DB2* インフォメーション・センター を再開します。

注: Windows 2008、Windows Vista (およびそれ以上) では、このセクションの後の部分でリストされているコマンドは管理者として実行する必要があります。完全な管理者特権でコマンド・プロンプトまたはグラフィカル・ツールを開くには、ショートカットを右クリックしてから、「管理者として実行」を選択します。

手順

コンピューターまたはイントラネット・サーバーにインストール済みの *DB2* インフォメーション・センターを更新するには、以下のようになります。

1. *DB2* インフォメーション・センターを停止します。

- Windows では、「スタート」 → 「コントロール パネル」 → 「管理ツール」 → 「サービス」をクリックします。次に、「**DB2** インフォメーション・センター」サービスを右クリックして「停止」を選択します。
- Linux では、以下のコマンドを入力します。

```
/etc/init.d/db2icdv97 stop
```

2. インフォメーション・センターをスタンドアロン・モードで開始します。

- Windows の場合:
 - a. コマンド・ウィンドウを開きます。
 - b. インフォメーション・センターがインストールされているパスにナビゲートします。デフォルトでは、*DB2* インフォメーション・センターは、`Program_Files\IBM\DB2 Information Center\Version 9.7` ディレクトリーにインストールされています (`Program_Files` は Program Files ディレクトリーのロケーション)。
 - c. インストール・ディレクトリーから `doc\bin` ディレクトリーにナビゲートします。
 - d. 次のように `help_start.bat` ファイルを実行します。

```
help_start.bat
```
- Linux の場合:
 - a. インフォメーション・センターがインストールされているパスにナビゲートします。デフォルトでは、*DB2* インフォメーション・センターは、`/opt/ibm/db2ic/V9.7` ディレクトリーにインストールされています。
 - b. インストール・ディレクトリーから `doc/bin` ディレクトリーにナビゲートします。
 - c. 次のように `help_start` スクリプトを実行します。

```
help_start
```

システムのデフォルト Web ブラウザーが開き、スタンドアロンのインフォメーション・センターが表示されます。

3. 「更新」ボタン (🔄) をクリックします。(ブラウザーで JavaScript™ が有効になっている必要があります。) インフォメーション・センターの右側のパネルで、「更新の検索 (Find Updates)」をクリックします。既存の文書に対する更新のリストが表示されます。
4. インストール・プロセスを開始するには、インストールする更新をチェックして選択し、「更新のインストール」をクリックします。
5. インストール・プロセスが完了したら、「完了」をクリックします。

6. 次のようにして、スタンドアロンのインフォメーション・センターを停止します。

- Windows の場合は、インストール・ディレクトリーの `doc\bin` ディレクトリーにナビゲートしてから、次のように `help_end.bat` ファイルを実行します。

```
help_end.bat
```

注: `help_end` バッチ・ファイルには、`help_start` バッチ・ファイルを使用して開始したプロセスを安全に停止するのに必要なコマンドが含まれています。`help_start.bat` は、Ctrl-C や他の方法を使用して停止しないでください。

- Linux の場合は、インストール・ディレクトリーの `doc/bin` ディレクトリーにナビゲートしてから、次のように `help_end` スクリプトを実行します。

```
help_end
```

注: `help_end` スクリプトには、`help_start` スクリプトを使用して開始したプロセスを安全に停止するのに必要なコマンドが含まれています。他の方法を使用して、`help_start` スクリプトを停止しないでください。

7. **DB2** インフォメーション・センター を再開します。

- Windows では、「スタート」 → 「コントロール パネル」 → 「管理ツール」 → 「サービス」をクリックします。次に、「**DB2** インフォメーション・センター」サービスを右クリックして「開始」を選択します。

- Linux では、以下のコマンドを入力します。

```
/etc/init.d/db2icdv97 start
```

結果

更新された **DB2** インフォメーション・センター に、更新された新しいトピックが表示されます。

DB2 チュートリアル

DB2 チュートリアルは、DB2 製品のさまざまな機能について学習するのを支援します。この演習をとおして段階的に学習することができます。

はじめに

インフォメーション・センター (<http://publib.boulder.ibm.com/infocenter/db2help/>) から、このチュートリアルの XHTML 版を表示できます。

演習の中で、サンプル・データまたはサンプル・コードを使用する場合があります。個々のタスクの前提条件については、チュートリアルを参照してください。

DB2 チュートリアル

チュートリアルを表示するには、タイトルをクリックします。

「*pureXML* ガイド」の『**pureXML**®』

XML データを保管し、ネイティブ XML データ・ストアに対して基本的な操作を実行できるように、DB2 データベースをセットアップします。

「Visual Explain チュートリアル」の『Visual Explain』

Visual Explain を使用して、パフォーマンスを向上させるために SQL ステートメントを分析し、最適化し、調整します。

DB2 トラブルシューティング情報

DB2 データベース製品を使用する際に役立つ、トラブルシューティングおよび問題判別に関する広範囲な情報を利用できます。

DB2 ドキュメンテーション

トラブルシューティング情報は、「DB2 問題判別ガイド」、またはDB2 インフォメーション・センターの『データベースの基本』セクションにあります。ここでは、DB2 診断ツールおよびユーティリティーを使用して、問題を切り分けて識別する方法、最も頻繁に起こる幾つかの問題に対するソリューションについての情報、および DB2 データベース製品を使用する際に発生する可能性のある問題の解決方法についての他のアドバイスがあります。

DB2 Technical Support の Web サイト

現在問題が発生していて、考えられる原因とソリューションを検索したい場合は、DB2 Technical Support の Web サイトを参照してください。

Technical Support サイトには、最新の DB2 資料、TechNotes、プログラム診断依頼書 (APAR またはバグ修正)、フィックスパック、およびその他のリソースへのリンクが用意されています。この知識ベースを活用して、問題に対する有効なソリューションを探し出すことができます。

DB2 Technical Support の Web サイト (http://www.ibm.com/software/data/db2/support/db2_9/) にアクセスしてください。

ご利用条件

これらの資料は、以下の条件に同意していただける場合に限りご使用いただけます。

個人使用: これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、非商業的な個人による使用目的に限り複製することができます。ただし、IBM の明示的な承諾をえずに、これらの資料またはその一部について、二次的著作物を作成したり、配布 (頒布、送信を含む) または表示 (上映を含む) することはできません。

商業的使用: これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することができます。ただし、IBM の明示的な承諾をえずにこれらの資料の二次的著作物を作成したり、お客様の企業外で資料またはその一部を複製、配布、または表示することはできません。

ここで明示的に許可されているもの以外に、資料や資料内に含まれる情報、データ、ソフトウェア、またはその他の知的所有権に対するいかなる許可、ライセンス、または権利を明示的にも黙示的にも付与するものではありません。

資料の使用が IBM の利益を損なうと判断された場合や、上記の条件が適切に守られていないと判断された場合、IBM はいつでも自らの判断により、ここで与えた許可を撤回できるものとさせていただきます。

お客様がこの情報をダウンロード、輸出、または再輸出する際には、米国のすべての輸出入関連法規を含む、すべての関連法規を遵守するものとします。

IBM は、これらの資料の内容についていかなる保証もしません。これらの資料は、特定物として現存するままの状態を提供され、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任なしで提供されます。

付録 B. 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。IBM 以外の製品に関する情報は、本書の最初の発行時点で入手可能な情報に基づいており、変更される場合があります。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒242-8502
神奈川県大和市下鶴間1623番14号
日本アイ・ビー・エム株式会社
法務・知的財産
知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。 IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

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

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができませんが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのもと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを

経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。サンプル・プログラムは、現存するままの状態を提供されるものであり、いかなる種類の保証も提供されません。IBM は、これらのサンプル・プログラムの使用から生ずるいかなる損害に対しても責任を負いません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生した創作物には、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。© Copyright IBM Corp. _年を入れる_. All rights reserved.

商標

IBM、IBM ロゴおよび [ibm.com](http://www.ibm.com)[®] は、世界の多くの国で登録された International Business Machines Corp. の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。

以下は、それぞれ各社の商標または登録商標です。

- Linux は、Linus Torvalds の米国およびその他の国における商標です。
- Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標です。
- UNIX は、The Open Group の米国およびその他の国における登録商標です。
- Intel、Intel ロゴ、Intel Inside[®]、Intel Inside ロゴ、Intel[®] Centrino[®]、Intel Centrino ロゴ、Celeron[®]、Intel[®] Xeon[®]、Intel SpeedStep[®]、Itanium[®]、Pentium[®] は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。
- Microsoft、Windows、Windows NT[®]、および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

アプリケーション

- コンパウンド SQL 109
- ストアード・プロシージャ 109
- 設計 109
- バインド 57
- パフォーマンス
 - アプリケーション設計 109
- Web
 - DB2 Connect 20
- アプリケーション開発
 - アプリケーション設計 109
 - IBM Data Server Driver Package 15
 - ODBC 15
- アプリケーション名、モニター・エレメント 80
- アプリケーション・サーバー
 - DB2 Connect 24
 - DRDA 定義 10
- アプリケーション・リクエスト (AR)
 - パラメーター 40
 - DRDA 定義 10
- アラート・オブジェクト
 - 表示 92
- アラート・サマリー
 - 表示 90
- 意思決定支援システム (DSS) 167
- インポート・ユーティリティ
 - ホスト/ワークステーション間のデータ転送 67
- ウィザード
 - マルチサイト更新 62
- ウィンドウ・スケーリング
 - RFC-1323 拡張 150
- エクスポート・ユーティリティ
 - ホスト/ワークステーション間のデータ転送 67
- エラー
 - トラブルシューティング 159
- エラー・メッセージ
 - DB2 Connect 179
- 応答時間
 - DB2 Connect 103

[カ行]

開始

- DB2 for z/OS ヘルス・モニター 86

- キャッシュ・アドレス・リスト 124
- 競合
 - システム・リソース 147
- クライアント
 - 自動クライアント・リルート
 - DB2 for z/OS への接続 131
 - ワークロードの平衡化 128
 - Sysplex のサポート
 - 制限 140
 - クライアント DB 別名 80
 - クライアント・アフィニティー
 - 構成 135
 - クライアント・アプリケーション
 - 通信のリカバリー 97
- 権限
 - バインド 57
 - コード化文字セット ID (CCSID)
 - 双方向サポート
 - 詳細 35
 - コール・レベル・インターフェース (CLI)
 - アプリケーション
 - CURRENTPACKAGESET CLI/ODBC 構成パラメーター 52
 - 概要 155
 - トラステッド接続 45
- コア・ファイル
 - 問題判別 162
- 高可用性
 - DB2 Connect 93
- 交換サーバー属性コマンド 169
- 更新
 - データベース・ディレクトリー 33
 - DB2 インフォメーション・センター 191, 193
- 構成
 - パスワード変更 52
 - ホスト接続 15
- 構成パラメーター
 - agentpri 143
 - dir_cache 143
 - MAXDARI 143
 - max_coordagents
 - 概要 113
 - 詳細 115
 - numdb 143
 - num_initagents 113, 115
 - num_poolagents 113, 115
 - rqrioblk 143
 - TCP_KEEPALIVE 99
- コマンド
 - コミット 169
 - ACCRDB 169

コマンド (続き)
ACCRDBRM 169
ACCSEC 169
db2drdat
 概要 167
db2trc
 トレースの取得 163
 トレース・ファイルのフォーマット 165
EXCSAT 169
EXCSATRD 169
GET SNAPSHOT
 概要 78
SECCHK 169
コマンド行プロセッサ (CLP)
 パフォーマンス 109
 SQL ステートメント 6
コミット・コマンド 169
ご利用条件
 資料 196
コントロール・センター
 マルチサイト更新 62

[サ行]

サーバー
 アプリケーション
 DB2 Connect 24
最終作業単位の応答メッセージ (ENDUOWRM) 169
作業単位 (UOW)
 概要 9
 分散 61
 リモート 11
参照
 複数のデータベース項目の定義 41
時間帯
 概要 35
システム状況
 GET SNAPSHOT コマンド 78
システム・データベース・ディレクトリー
 値 33
 更新 33
システム・リソース
 競合 147
自動クライアント・リルート
 クライアント・サイド 131
 構成 133
 詳細 97
 接続失敗 99
 セットアップ 97
シナリオ
 TCP/IP セキュリティー 54
述部
 ロジック・パフォーマンス 109
照会ブロック
 DB2 Connect のデータ転送速度の向上 148

資料
 印刷 186
 概要 185
 使用に関するご利用条件 196
 注文 189
 PDF ファイル 186
診断情報
 概要 162
シンボリック宛先名
 大/小文字の区別 34
信頼関係
 DB2 Connect 45
ストアド・プロシージャ
 概要 23
スループット
 トランザクション 103
静的 SQL
 処理の影響 5
 パフォーマンス 109
セキュリティー
 タイプ 40
 ノード・ディレクトリーの値 34
 ヒント 52
 DB2 Connect 45
 DB2 for z/OS での拡張コード・サポート 52
 GRANT ステートメント 54
 Kerberos 51
 TCP/IP 54
接続
 管理 113
 再確立
 ホストに直接 15
 DB2 Connect Enterprise Edition 18
 失敗
 自動クライアント・リルート 99
 プール
 概要 113
 接続コンセントレーター 115
 利点 115
 ホストに直接 15
 DB2 Connect Enterprise Edition 18
 IBM i に直接 17
 IBM メインフレームに直接 15
 System z ホストに直接 17
接続コンセントレーター
 概要 113, 115
 作業エージェント 115
 接続の管理 113
 接続プールの比較 121
 DB2 Connect 122
接続サーバー
 DB2 Connect Enterprise Edition 18
接続プール
 概要 113
 接続コンセントレーターの比較 121
 接続の管理 113

ゾーン 10 進数データ・タイプ 151
送信バッファ
トレース・データ 167
双方向 CCSID サポート
BIDI パラメーター 35

[夕行]

ターゲット・データベース
名前 35, 40
チュートリアル
トラブルシューティング 196
問題判別 196
リスト 195
Visual Explain 195
チューニング
ネットワーク 145
パラメーター
agentpri 143
dir_cache 143
maxagents 143
MAXDARI 143
numdb 143
rqrioblk 143
ホスト・データベース 145
DB2 Connect 101
DB2 for z/OS 148
ツール
パフォーマンス 103
メモリー使用状況 103
CPU 使用率 103
追加照会ブロック
概要 149
EXTRA BLOCKS SRV パラメーター 149
通信
リカバリー 97
データ
ソース 12
転送
速度 103, 151
パフォーマンス 151
ホスト/ワークステーション間 67
フロー
DB2 Connect 10, 103
ブロッキング 109
データの移動
DB2 Connect 67
データベース
チューニング 145
名前
システム・データベース・ディレクトリー 33
ディレクトリー・カスタマイズ・ワークシート 40
DCS ディレクトリー 35
RDBNAM オブジェクト 169
パフォーマンス・ツール 103

データベース (続き)
別名
システム・データベース・ディレクトリー 33
ディレクトリー・カスタマイズ・ワークシート 40
ホスト 5
要求のグループ化 109
データベース接続サービス (DCS) ディレクトリー
値 35
項目の更新 33
データベース要求
パフォーマンスのためのグループ化 109
データベース・システム・モニター
概要 6
リモート・クライアント 77
データベース・ディレクトリー
更新 33
データベース接続サービス (DCS) 33
ノード 33
複数項目 41
データ・タイプ
ゾーン 10 進数 151
バック 10 進数 151
浮動小数点
ホスト・データの変換 151
変換
パフォーマンスの影響 151
文字 151
CHAR 151
INTEGER
ホスト・データの変換 151
VARCHAR
概要 151
停止
DB2 for z/OS ヘルス・モニター 86
定様式データ・オブジェクト・コンテンツ体系 (FDOCA) 10
ディレクトリー
カスタマイズ 40
システム・データベース
値 33
更新 33
ディレクトリー・キャッシュ・サポート構成パラメーター
DB2 Connect のチューニング 143
テスト
マルチサイト更新 62
トークン
SQLCODE 71
同期点管理プログラム (SPM)
構成パラメーター
デフォルト 64
シナリオ 63
動的 SQL
処理の影響 5
パフォーマンス
技法 109
CURRENTPACKAGESET CLI/ODBC 構成パラメーター 52
特記事項 199

- トラステッド接続
 - CLI/ODBC 46
 - CLI/ODBC を使用したユーザーの切り替え 48
 - DB2 Connect 45
- トラステッド・コンテキスト
 - CLI/ODBC サポート 46
 - DB2 Connect のサポート 45
- トラブルシューティング
 - オンライン情報 196
 - 情報の収集 159
 - 接続 159, 160
 - チュートリアル 196
 - トレース
 - db2trc コマンドを使用した取得 163
 - DRDA 170, 175
 - パフォーマンス 147
 - DB2 Connect 159, 179
- トランザクション
 - 作業単位 (UOW) 9
 - スループット
 - DB2 Connect 103
 - 疎結合
 - DB2 Connect 65
 - トランザクション処理モニター 27
 - 分散 61
 - マルチサイト更新 9, 61
 - 2 フェーズ・コミット 9
 - DB2 Connect Enterprise Edition 27
 - XA 分散アプリケーション 65
- トランザクション処理モニター
 - マルチサイト更新 61
 - 例 27
 - DB2 Connect 27
 - OLTP 27
 - Tuxedo 27
- トランスポート 128
- トランスポート・プール 128
- トレース
 - 出力ファイル 167, 168
 - 出力ファイル・サンプル 170
 - DB2 163, 164, 165
 - DB2 Connect 163
 - DB2 Connect とサーバーの間のデータ 167
 - DRDA
 - 解釈 167
 - サンプル 170
 - バッファ情報 175
- トレース・ユーティリティー (db2drdat) 167

[ナ行]

- 認証 40
 - 概要 50
 - 検証 50
 - システム・データベース・ディレクトリー 33

- 認証 (続き)
 - タイプ
 - デフォルト 50
 - CLIENT 50, 52
 - DATA_ENCRYPT 50
 - KERBEROS 50
 - SERVER 50
 - SERVER_ENCRYPT 50
 - SERVER_ENCRYPT_AES 50
 - DB2 Connect 54
 - REVOKE ステートメント 54
- ネットワーク
 - チューニング 145
 - データ転送速度の向上 151
 - パフォーマンス・ツール 103
- ノード
 - ディレクトリー
 - 値 34
 - 更新 33
 - 名前
 - システム・データベース値 33
 - ディレクトリー・カスタマイズ・ワークシート 40
 - ノード・ディレクトリーの値 34

[ハ行]

- ハードウェア
 - ネットワーク・パフォーマンス 151
- バインド
 - アプリケーション 57
 - 権限 57
 - パッケージ
 - DB2 Connect 57
 - ユーティリティー
 - DB2 Connect 57
- バインド・リスト
 - DB2 Connect 57
- パスワード
 - 変更
 - z/OS 52
- バック 10 進数データ・タイプ 151
- パッケージ
 - ホスト・データベース・サーバー 57
 - System i データベース・サーバー 57
- パフォーマンス
 - アプリケーション設計 109
 - コマンド行プロセッサ (CLP) による影響 109
 - システム・リソース 147
 - 接続コンセントレーター 121
 - 接続プール 121
 - ネットワーク・ハードウェア 151
- DB2 Connect
 - 概要 103
 - チューニング 101
 - 転送速度の向上 148
 - トラブルシューティング 147

パフォーマンス (続き)
 ODBC アクセス 107
 z/OS 148
 パラメーター
 ストリング 41
 ディレクトリー 40
 PRDID 169
 SYSPLEX 35
 パラメーター・ストリング
 コンマ 35
 二重コンマ 35
 日付
 時間帯サポート 35
 ブートストラップ・データ・セット (BSDS) パラメーター
 z/OS 34
 ファイルへのトレースのダンプ
 概要 164
 フェデレーテッド・データベース
 分散要求 12
 浮動小数点データ・タイプ
 変換 151
 プロセス状況ユーティリティー
 コマンド 162, 169
 ブロッキング
 データ 109
 ブロック・サイズ
 DB2 Connect 143
 分散作業単位
 概要 9
 サポートされているサーバー 61
 マルチサイト更新 61
 2 フェーズ・コミット 61
 分散データ管理 (DDM)
 分散リレーショナル・データベース体系 (DRDA) 10
 db2drdat 出力 167
 分散要求
 概要 12
 分散リレーショナル・データベース体系 (DRDA)
 概要 9
 データ・アクセス 9
 DB2 Connect 10
 ページング・ブロック・サイズ 143
 ヘルス・アラート
 DB2 for z/OS 85
 ヘルス・モニター
 DB2 for z/OS 85
 ヘルプ
 言語の構成 190
 SQL ステートメント 190
 変換
 ホスト 151
 ベンチマーク
 パフォーマンス 103
 ホスト・データベース
 接続
 高可用性 95

ホスト・データベース (続き)
 接続 (続き)
 ロード・バランシング 95
 DB2 Connect Personal Edition を使用したアクセス 15
 ボトルネック
 トランザクション 103
 パフォーマンス 103
 ポリシー評価ログ 85
 本書について vii

[マ行]

マルチサイト更新
 コントロール・センター 62
 使用可能にする 61
 テスト 62
 同期点管理プログラム 63
 分散作業単位 (DUOW) 61
 「マルチサイト更新の構成」ウィザード 62
 メモリー
 使用率に関するツール 103
 文字データ・タイプ 151
 モニター
 接続 77
 Windows パフォーマンス モニタ 77
 問題判別
 診断ツール
 概要 162
 接続 159
 接続後 160
 チュートリアル 196
 利用できる情報 196

[ヤ行]

ユーティリティー
 データベース・システム・モニター 6
 トレース 167
 バインド 57
 DB2 Connect 管理 6
 db2drdat 167
 ddcspkgn 57
 ps (プロセス状況) 162, 169

[ラ行]

リソース・アクセス管理機能 (RACF)
 認証 54
 リモート作業単位
 概要 11
 特性 11
 例 11
 例
 接続コンセントレーター 115
 XA コンセントレーター 115

ログ
ポリシー評価 85

[ワ行]

ワークシート
ディレクトリーのカスタマイズ 40
ワークロード・バランシング
クライアント用の構成 129
クライアント・サイド 128

[数字]

2 フェーズ・コミット
使用可能にする 61
TCP/IP 接続によって使用される resynch ポート 34

A

ACCRDB コマンド 169
ACCRDBRM コマンド 169
ACCSEC コマンド 169
AFFINITY_LIST db2dsdriver 設定 135
agentpri データベース・マネージャー構成パラメーター 143
ALTERNATE_SERVER_LIST db2dsdriver 設定 135
AS ターゲット・データベース名 35
ATOMIC コンパウンド SQL
DB2 Connect ではサポートされていない 109

B

BINDADD 権限
DB2 Connect 57

C

CGI (Common Gateway Interface) プログラミング 20
CHAR データ・タイプ
詳細 151
Character Data Representation Architecture (CDRA) 10
CLIENT 認証タイプ
DB2 Connect 50
CLIENT_AFFINITY db2dsdriver 設定 135
COMMIT ステートメント
静的にバインドされた 109
Common Gateway Interface (CGI) プログラミング
制限 20
利点 20
CPU
パフォーマンス・ツール 103
CREATE IN COLLECTION NULLID 権限 57
CURRENTPACKAGESET CLI/ODBC キーワード 52

D

D (切断) パラメーター 35
DATA_ENCRYPT 認証タイプ 50
DB2 Connect
概要 3
管理ユーティリティ 6
機能拡張
機能 3
構成
IBM Power Systems 64
IBM メインフレーム 64
System z 64
製品 3
セキュリティ 45
接続コンセントレーター 122
接続サーバーのシナリオ 15
データの移動 67
ホスト・サポート 15
メインフレーム・サポート 15
Enterprise Edition
接続サーバー 18
トランザクション処理モニター 27
Java アプリケーション・サーバー 22
Web アプリケーション 20
Web サーバー 23
XA 準拠トランザクション・マネージャー 64
Sysplex のサポート 123, 127
System i サポート 15
DB2 for z/OS
セキュリティ 52
ノード・ディレクトリーの値 34
ヘルス・モニター
アラート・オブジェクト 92
アラート・サマリー 90
開始 86
概要 85
推奨処置 87
停止 86
リフレッシュ 86
DYNAMICRULES (BIND) オプション 52
DB2 インフォメーション・センター
言語 190
更新 191, 193
バージョン 190
DB2 資料の印刷方法 189
db2drdat コマンド
出力ファイル 167
db2dsdriver 構成ファイル 129
db2trc コマンド
概要 163
トレース出力のダンプ 164
トレース出力のフォーマット 165
DCS (データベース接続サービス) ディレクトリー
「データベース接続サービス (DCS) ディレクトリー」を参照 35

dcslari.map ファイル 71
dcsldsn.map ファイル 71
dcslqsq.map ファイル 71
ddcs400.lst ファイル 57
ddcsmvs.lst ファイル 57
ddestrc ユーティリティ 168
ddcsvm.lst ファイル 57
ddcsvse.lst ファイル 57
DESCRIBE ステートメント
 コンパウンド SQL ステートメント 109
 PREPARE ステートメントによるパフォーマンス 109
dir_cache パラメーター 143

E

enableAcr パラメーター 133
enableSeamlessAcr パラメーター 133
enableWLB パラメーター 129
EXCSAT コマンド 169
EXCSATRD コマンド 169
EXECUTE IMMEDIATE ステートメント
 アプリケーション設計 109
EXTNAM オブジェクト 169

F

FOR FETCH ONLY 節
 SELECT ステートメント 109
FORCE コマンド 80

G

GET SNAPSHOT コマンド
 概要 78

I

IBM WebSphere
 概要 21
IBM データ・サーバー・クライアント
 自動クライアント・リルート
 クライアント・サイド 131
 ワークロード・バランシング 128
IBM データ・サーバー・ドライバー
 自動クライアント・リルート
 クライアント・サイド 131
 ワークロード・バランシング 128
InfoSphere Federation Server
 概要 7
INTEGER データ・タイプ
 ホスト・データの変換 151
INTERRUPT_ENABLED (切断) パラメーター 35

J

Java
 アプリケーション・サーバー
 API 22
 DB2 Connect 22
 JDBC 22
 SQLJ 22

K

Kerberos 認証プロトコル
 DB2 Connect 50
 OS/390 51
 z/OS 51

L

LIST DCS APPLICATIONS コマンド
 出力 80
LOCALDATE パラメーター 35

M

maxagents データベース・マネージャー構成パラメーター
 推奨されない 143
maxRefreshInterval パラメーター 129
maxTransportIdleTime パラメーター 129
maxTransports パラメーター 129
maxTransportWaitTime パラメーター 129
max_coordagents データベース・マネージャー構成パラメーター
 概要 113
 詳細 115
Microsoft Windows
 アプリケーション 15

N

NOMAP パラメーター
 DCS ディレクトリー・パラメーター 71
 SQL CODE マッピング 35
 SQL マッピングをオフにする 71
NONE 認証タイプ 54
NOT ATOMIC コンパウンド SQL
 アプリケーション設計 109
NULLID 57
numdb 構成パラメーター
 DB2 Connect 143
num_initagents データベース・マネージャー構成パラメーター
 アイドル・エージェント・プールの構成 113
 概要 115
num_poolagents データベース・マネージャー構成パラメーター
 アイドル・エージェント・プールの構成 113
 概要 115

O

ODBC

- アプリケーション
 - CURRENTPACKAGESET CLI/ODBC 構成パラメーター 52
- インターフェース 15
- 最適化アクセス 107
- CLI/ODBC アプリケーションのパフォーマンス調整 155

P

- PRDID パラメーター 169
- PREPARE ステートメント
 - アプリケーション設計 109
 - パフォーマンスへの影響 109
- PROGRAM 認証タイプ 54
- ps コマンド
 - 概要 162
 - EXTNAM オブジェクト 169

R

- RECEIVE BUFFER 167
- ROLLBACK ステートメント
 - 静的にバインドされた 109
- rqrioblk 構成パラメーター
 - チューニング 143

S

- SAME 認証タイプ 54
- SECCHK コマンド 169
- SELECT ステートメント
 - アプリケーション設計 109
 - 更新可能 109
 - FOR FETCH ONLY 109
- SERVER 認証タイプ
 - DB2 Connect 50
- SERVER_ENCRYPT 認証タイプ
 - DB2 Connect 50
- SERVER_ENCRYPT_AES 認証タイプ 50
- SET CURRENT PACKAGESET ステートメント 52
- SHOW DETAIL モニター・オプション 80
- SOCKS
 - ノード
 - 必須環境変数 34
- SQL
 - 静的 109
 - 動的 109
- SQL ステートメント
 - ヘルプ
 - 表示 190
 - COMMIT 109
 - DB2 Connect 5

SQL ステートメント (続き)

- DESCRIBE 109
- EXECUTE IMMEDIATE 109
- PREPARE 109
- ROLLBACK 109
- SELECT 109
 - SELECT の FOR FETCH ONLY 節 109
- SQL0965 エラー・コード 179
- SQL0969 エラー・コード 179
- SQL30020 エラー・コード 179
- SQL30060 エラー・コード 179
- SQL30061 エラー・コード 179
- SQL30073 エラー・コード 179
- SQL30081N エラー・コード 179
- SQL30082 エラー・コード 179
- SQL5043N エラー・コード 179
- SQLCA
 - データのバッファ 167
 - SQLCODE フィールド 167
- SQLCODE
 - マッピング 71
 - マッピング・ファイル 71
 - SQLCA 内のフィールド 167
- SQLDA
 - 割り振りサイズ 109
- SQLSTATE
 - クラス・コード 71
- SQL_ATTR_
 - TRUSTED_CONTEXT_PASSWORD
 - CLI を使用したトラステッド接続のユーザーの切り替え 48
 - TRUSTED_CONTEXT_USERID
 - CLI を使用したトラステッド接続のユーザーの切り替え 48
 - USE_TRUSTED_CONTEXT
 - CLI を使用したトラステッド接続の作成 46
- SRVNAM オブジェクト 169
- START MVS システム・コマンド 85
- STOP MVS システム・コマンド 85
- Sysplex
 - クライアント・サイド・サポートに関する制限 140
 - クライアント・サポート 127
 - 構成要件 125
 - 使用 124
 - パラメーター 35
 - フォールト・トレランス 124
 - 優先順位情報 124
 - ロード・バランシング 124
 - DB2 Connect のサポート 123
 - System z 123

T

- TCP/IP
 - 構成
 - ホスト接続 17

TCP/IP (続き)

- サービス名 34
- セキュリティー 52
- 認証シナリオ 54
- ポート番号 40
- ホスト名 40
- リモート・ホスト名 34, 40
- ACCSEC コマンド 169
- DOMAIN 34
- RESPORT 34
- resynch ポート 34
- RFC-1323 拡張 150
- SECCHK コマンド 169
- TCPSPORT 34

TCP_KEEPALIVE オペレーティング・システム構成パラメータ
ー 99

Tuxedo

- DB2 Connect Enterprise Edition 27

V

VARCHAR データ・タイプ

- 概要 151

W

Web アプリケーション

- ストアド・プロシージャー 23
- DB2 Connect 20

Web サーバー

- DB2 Connect 23

WebSphere

- 概要 21

WebSphere MQ

- DB2 Connect 122

Windows

- パフォーマンス モニタ
- DB2 アプリケーションのモニター 77

X

XA

- コンセントレーターの例 115
- トラステッド接続 45
- リソース・マネージャー 27

XA トランザクション・マネージャー

- 概要 27
- 接続コンセントレーター 115

X/Open 分散トランザクション処理 (DTP) モデル

- 概要 27

[特殊文字]

&&

- SQLCODE マッピング・ファイル 71



Printed in Japan

SC88-5890-01



日本アイ・ビー・エム株式会社

〒103-8510 東京都中央区日本橋箱崎町19-21

Spine information:

IBM DB2 Connect 9.7 バージョン 9 リリース 7

DB2 Connect ユーザーズ・ガイド

