



問題判別ガイド



問題判別ガイド

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、127 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。

当版に関する特記事項

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。

IBM 発行のマニュアルに関する情報のページ

<http://www.ibm.com/jp/manuals/>

こちらから、日本語版および英語版のオンライン・ライブラリーをご利用いただけます。また、マニュアルに関するご意見やご感想を、上記ページよりお送りください。今後の参考にさせていただきます。

(URL は、変更になる場合があります)

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

原 典： GI11-7857-01
DB2 Version 9.5 for Linux, UNIX, and Windows
Troubleshooting Guide

発 行： 日本アイ・ビー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

目次

本書について	v
------------------	---

第 1 章 トラブルシューティングについて 1

トラブルシューティングの概要	1
First Occurrence Data Capture について	3
一般的な障害問題に基づく診断情報の収集	4
診断情報の自動収集のための構成	4
FODC の一部として収集されたデータとその配置	5
自動 FODC データ生成	6
First Occurrence Data Capture 情報	6
DB2 Query Patroller と First Occurrence Data Capture (FODC)	9
First Occurrence Data Capture (FODC) を使用する モニター機能と監査機能	10
First Occurrence Data Capture (FODC) を使用する グラフィック・ツール	11
管理通知ログ・ファイルについて	12
管理通知ログ	12
管理通知ログ・ファイルのエラー・キャプチャー レベルの設定	13
管理通知ログ・ファイル項目の解釈	13
DB2 診断ログ (db2diag.log) ファイルについて	15
診断ログ・ファイルのエラー・キャプチャー・レ ベルの設定	15
db2diag.log ファイルの情報レコードの解釈	15
診断ログ・ファイル項目の解釈	16
db2cos (コールアウト・スクリプト) 出力ファイル ダンプ・ファイル	19
21	
22	
22	
23	
24	
24	
25	
26	
26	
26	

第 2 章 DB2 のトラブルシューティング 31

現行リリースのトラブルシューティング・ガイド ス	31
高可用性のトラブルシューティング	32
インストールのトラブルシューティング	32
パーティション・データベース環境のトラブル シューティング	33
最適化ガイドラインおよびプロファイルのトラ ブルシューティング	34
ストレージ・キー・サポートのトラブルシュー ティング	37

データ・コンプレッション・ディクショナリーが 自動的に作成されない	38
グローバル変数問題のトラブルシューティング	39
ワークロード管理のトラブルシューティング	41
スクリプトのトラブルシューティング	42
データ不整合のトラブルシューティング	42
索引とデータの不整合のトラブルシューティング	42
DB2 についてのデータの収集	43
インストール問題についてのデータの収集	43
データ移動問題についてのデータの収集	44
DAS およびインスタンス管理の問題についてのデ ータの収集	45
DB2 についてのデータの分析	46
インストールの問題に関するデータの分析	46
IBM ソフトウェア・サポートへのデータの送信	47

第 3 章 DB2 Connect のトラブルシュー ティング 49

トラブルシューティング	49
関係のある情報の収集	49
初期接続が正常に行われなかった場合	49
初期接続後に発生する問題	50
診断ツール	52
一般的な DB2 Connect の問題	52

第 4 章 トラブルシューティングのための ツール 57

内部戻りコードに関する詳細情報	57
db2dart ツールの概要	59
INSPECT と db2dart の比較	59
db2diag を使用した db2diag.log ファイルの分析	62
db2greg を使用したグローバル・レジストリーの表示 および変更 (UNIX)	64
製品のバージョンとサービス・レベルの識別	64
db2look を使用したデータベースの模造	65
システムにインストールされている DB2 製品のリス ト表示 (Linux および UNIX)	68
db2pd を使用したモニターおよびトラブルシューテ ィング	70
db2support を使用した環境情報の収集	80
基本的なトレース診断	83
DB2 トレース	84
DRDA トレース	88
コントロール・センターのトレース	96
JDBC トレース	96
CLI トレース・ファイル	99
プラットフォーム固有のツール	105
診断ツール (Windows)	105
診断ツール (Linux および UNIX)	105

第 5 章 既知の問題の検索方法	109
トラブルシューティングのリソース	110
第 6 章 DB2 製品フィックスの入手	111
フィックスパックの適用	111
フィックスパック、テスト・フィックス、および APAR	113
テスト・フィックスの適用	115
付録 A. DB2 技術情報の概説	117
DB2 テクニカル・ライブラリー (ハードコピーまたは PDF 形式)	118
DB2 の印刷資料の注文方法	120
コマンド行プロセッサから SQL 状態ヘルプを表示する	121

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	121
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	122
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	122
DB2 チュートリアル	125
DB2 トラブルシューティング情報	125
ご利用条件	126

付録 B. 特記事項	127
-------------------	------------

索引	131
-----------	------------

本書について

本書には、DB2[®] データベース・クライアントおよびサーバーに関連する問題の解決を開始するための情報を記載しています。これによって、以下のことを行う助けが得られます。

- 問題とエラーを簡便な方法で識別する
- 問題を症状に基づいて解決する
- 使用可能な診断ツールについて知る
- 日常の操作のトラブルシューティング・ストラテジーを作成する

本書の対象読者

本書は、DB2 データベース・クライアントおよびサーバーのお客様、ユーザー、システム管理者、データベース管理者 (DBA)、通信スペシャリスト、アプリケーション開発者、およびテクニカル・サポート担当員を対象としています。本書を使用するには、以下についてよく理解している必要があります。

- 通信、リレーショナル・データベース、ローカル・エリア・ネットワーク (LAN) のそれぞれの概念
- ハードウェアおよびソフトウェアの要件とオプション
- ネットワークの全体構成
- ネットワーク上で稼働するアプリケーション・プログラムおよびその他の機能
- DB2 データベースの基本的な管理タスク
- インストール済み製品のガイド「概説およびインストール」で説明されている、インストールおよび初期作業についての情報

第 1 章 トラブルシューティングについて

DB2 データベース製品で作業中の何らかの時点で、問題が発生する場合があります。こうした問題は、データベース・マネージャーから報告されることもありますし、データベースに対して実行中のアプリケーションから報告されることもあります。また、データベースが「正常ではない」というユーザーからのフィードバックとして報告されることもあります。ここで述べる概念とツールは、データベースの操作で発生する実際の問題や問題と見なされるものに対するトラブルシューティング作業の概要を説明し、読者のトラブルシューティング作業を支援することを目的としています。適切なときに適切なデータを収集することが重要であることを強調する意味で、ツールの説明は First Occurrence Data Capture から始まります。データベース・マネージャーがデータベースの操作に関するデータをキャプチャーするために使用する他のログおよびファイルについても、オペレーティング・システムの診断ツールを含めて説明します。

トラブルシューティングの概要

良い問題分析の最初のステップは、問題を完全に記述することです。問題記述なしでは、問題の原因を突き止めるためにどこから始めたらいかがが分かりません。このステップでは、以下のような基本的な質問を自問自答します。

- どのような症状か
- 問題が発生しているのはどこか
- 問題が発生するのはいつか
- どのような条件で問題が発生するか
- 問題は再現可能か

これらの質問や他の質問に答えることは、ほとんどの問題に対する正しい記述に結びつき、問題解決し始めるための最良の方法です。

どのような症状か

問題を記述し始める際にまず考えられる質問は「何が問題か」というものです。これは平易な質問のように思えます。しかし、問題をより明確にするには、これをさらにいくつかの質問に分割することができます。以下のような質問です。

- 問題を報告しているのは誰または何か
- エラー・コードおよびエラー・メッセージはどのようなものか
- どのような障害が起きるか (ループ、ハング、異常終了、性能低下、結果の誤りなど)
- ビジネスへはどのような影響があるか

問題が発生しているのはどこか

問題の発生箇所の判別は必ずしも容易ではありませんが、問題の解決において最も重要なステップの 1 つです。報告を行うコンポーネントと障害が起きているコンポーネントの間には、多くのテクノロジーの層が存在していることがあります。問

題を突き止める際に考慮すべきコンポーネントをほんのいくつか挙げるだけでも、ネットワーク、ディスク、ドライバーなどがあります。

- 問題はプラットフォームに固有のものか、それとも複数のプラットフォームに共通のものか
- 現行の環境および構成はサポートされているか
- アプリケーションはデータベース・サーバー上でローカルに実行しているか、それともリモート・サーバー上で実行しているか
- ゲートウェイが関係しているか
- データベースは個々のディスク上にあるか、それとも RAID ディスク・アレイにあるか

これらのタイプの質問は、問題の層を切り分けるのに役立つとともに、問題の発生源を判別するのに不可欠です。1つの層が問題を報告しているからといって、根本原因がそこに存在するとは限らないことに注意してください。

問題がどこで発生しているかを識別することには、問題が存在している環境を理解することが含まれます。必ずいくらかの時間を取って、問題の発生している環境を完全に記述する必要があります。それには、オペレーティング・システムとそのバージョン、該当するすべてのソフトウェアとバージョン、およびハードウェア情報が含まれます。実行環境が、サポートされている構成になっているかどうかを確認してください。というのは、一緒に実行するよう意図されていないソフトウェア・レベル、または一緒に十分にテストされていないソフトウェア・レベルを突き止めることで解明可能な問題は少なくないからです。

問題が発生するのはいつか

障害につながる詳細なイベントを時系列上に並べることは、問題分析における別の不可欠なステップです。一回限りの現象である場合には特にそうです。この作業の最も簡単な方法は、過去にさかのぼって行くことです。すなわち、エラーが報告された時刻（ミリ秒単位にまで至る、可能な限り正確な時刻）を起点に、入手可能なログと情報を過去にさかのぼって作業を進めていきます。通常はいずれかの診断ログで最初に見つかる疑わしいイベントまで調べれば十分です。しかし、これは必ずしも容易に行えることではなく、経験を積むことが必要です。特に、テクノロジーの層が複数あり、それぞれに独自の診断情報がある場合には、どこでやめるかを判断するのは困難です。

- 問題は、昼間または夜間の特定の時刻にのみ発生するか
- どれほどの頻度で発生するか
- 問題の報告される時刻まで、どのような一連のイベントが生じているか
- 問題が発生するのは、既存のソフトウェアまたはハードウェアのアップグレードや、新規のソフトウェアまたはハードウェアのインストールといった環境の変更後か

このような質問に答えることにより、イベントを時系列に並べるのに役立ち、調査の基準枠を持つことができます。

どのような条件で問題が発生するか

問題の発生時に、他に何が実行されていたかを知ることは、完全な問題記述のために重要です。特定の環境または特定の条件下で問題が発生する場合、それが問題の原因のかぎとなる標識であることがあります。

- 同じタスクを実行すると、常にその問題が発生するか
- 問題が表面化するには、特定の順序でイベントが発生する必要があるか
- 同時に他のアプリケーションにも障害が起きるか？

これらのタイプの質問に答えることにより、問題が発生する環境を解明し、依存関係を明らかにするのに役立ちます。複数の問題がほぼ同時に発生するというだけでは、必ずしもそれらが関連していることにはならないことを覚えておいてください。

問題は再現可能か

問題記述および調査の観点から見て「理想的」な問題とは、再現可能な問題です。再現可能な問題の場合はたいがい、調査の助けになるツールやプロシーチャーのセットがより多く利用できます。その結果、再現可能な問題は、通常デバッグおよび解決をより容易にします。

しかし、再現可能な問題にも欠点があります。すなわち、ビジネス上の大きな影響を伴う問題の場合、これを繰り返し発生させるわけにはいきません。この場合、可能であれば、テストまたは開発環境で問題を再現するほうが望ましいでしょう。

- テスト・マシン上で問題を再現できるか
- 複数のユーザーまたはアプリケーションが、同じタイプの問題に遭遇するか
- 単一コマンド、コマンドのセット、特定のアプリケーション、または独立型アプリケーションのいずれかを実行して、問題を再現できるか
- DB2 コマンド行から等価のコマンドまたは照会を入力して、その問題を再現できるか

多くの場合、テストまたは開発環境で単一の問題を再現するのが望ましいでしょう。通常、そのほうが、調査を行う際により一層の柔軟性と制御が得られるからです。

First Occurrence Data Capture について

First Occurrence Data Capture (FODC) とは、エラーの発生時に DB2 データベース製品が自動的に取り込む一式の診断情報に当てはまる用語です。その情報によって、診断情報を得るためにエラーを再現する必要性が小さくなります。

FODC に関係したトピックでは、診断情報の重要性と、それを集めるためのさまざまな方式を紹介します。データベース・ツール、ユーティリティー、および DB2 製品に関連したいくつかのログがあり、それらは、データベースでの問題をトラブルシューティングする際に役立ちます。

一般的な障害問題に基づく診断情報の収集

インスタンスに影響を与える障害が発生した時点で、診断情報がパッケージに自動的に収集されるように設定できます。パッケージ内の情報を手動で作成することも可能です。

DB2 のインスタンスやデータベースで作業しているときにトラブルが発生した場合は、その問題が起きた時点でデータを収集してください。First Occurrence Data Collection (FODC) は、DB2 環境でトラブルが発生したときに実行されていたアクションを表すために使用する用語です。db2pdcfg ツールを使用して DB2FODC レジストリー変数にオプションを設定することによって、障害の発生時にどのデータを収集するのかを制御できます。DB2FODC レジストリー変数のオプションを変更するには、db2pdcfg -fodc コマンドを使用します。これらのオプションは、FODC 状態のデータ・キャプチャーに関するデータベース・システムの動作に影響を与えます。

診断情報の自動収集

データベース・マネージャーは、自動 First Occurrence Data Capture (FODC) のために db2fodc コマンドを呼び出します。

DB2 診断ログと他のトラブルシューティング・ファイルに障害を関連付けるために、管理通知ログと db2diag.log の両方に診断メッセージが書き込まれます。診断メッセージには、FODC ディレクトリー名と、FODC ディレクトリーが作成された時のタイム・スタンプが組み込まれます。FODC パッケージの記述ファイルは、新しい FODC ディレクトリーに配置されます。

診断情報の手動収集

ハングする可能性がある状況や、重大なパフォーマンス上の問題がある状況に関する情報は、First Occurrence Data Collection コマンド (db2fodc) を使用して収集できます。db2fodc コマンドを実行すると、新しいディレクトリー FODC_hang_<timestamp> が、現在の診断パスに自動的に作成されます。その後、db2cos_hang スクリプトが実行されます。このスクリプトは、データを収集して FODC サブディレクトリーに配置するデータ収集操作を制御します。この FODC サブディレクトリーが存在するかどうかは、db2fodc コマンドの実行方法や DB2 レジストリー変数の構成に依存しています。

診断情報の自動収集のための構成

データベース・マネージャーが自動的にアクションを実行するには、どのアクションを実行するのかをデータベース・マネージャーに対して指定する必要があります。

データベース操作中にエラーや警告が検出されたときに、データベース・マネージャーが実行するアクションを示すフラグを設定します。実行するアクションとしては、次のようなアクションがあります。

- db2diag.log のスタック・トレースの生成。(デフォルト)
- コールアウト・スクリプト db2cos の実行。(デフォルト)
- トレース (db2trc) コマンドの停止。

First Occurrence Data Capture (FODC) オプションの変更

問題判別動作用の DB2 データベースの構成 (db2pdcfg) コマンドを使用して、First Occurrence Data Capture (FODC) オプションを変更します。DB2FODC レジストリー変数の FODC オプションを設定するには、db2pdcfg ツールを使用します。これらのオプションは、FODC 状態のデータ・キャプチャーに関するデータベース・システムの動作に影響を与えます。

FODC の一部として収集されたデータとその配置

インスタンス内で発生した障害のタイプによっては、First Occurrence Data Capture (FODC) の結果としてサブディレクトリーが作成され、特定の内容が収集されます。一連のサブディレクトリーが作成され、それぞれのファイルやログが格納されます。

FODC ディレクトリーに、以下のサブディレクトリーが 1 つ以上作成されます。

- DB2CONFIG (DB2 の構成出力とファイルが入ります)
- DB2PD (db2pd の出力または出力ファイルが入ります)
- DB2SNAPS (DB2 のスナップショットが入ります)
- DB2TRACE (DB2 のトレースが入ります)
- OSCONFIG (オペレーティング・システムの構成ファイルが入ります)
- OSSNAPS (オペレーティング・システムのモニター情報が入ります)
- OSTRACE (オペレーティング・システムのトレースが入ります)

DB2FODC の構成や db2fodc の実行モードによっては、これらのディレクトリーが常に存在するとは限りません。

障害のタイプに応じて、以下の内容が FODC ディレクトリーとサブディレクトリーにあります。

- トラップ・ファイル
- 障害発生時のデータ・キャプチャーで生成され、各種のコンポーネントによって完成された、あらゆる種類のバイナリー・ファイルとプレーン・テキスト・ダンプ・ファイル
- db2evlog のイベント・ログ・ファイル
- DB2 トレース・ダンプ (障害発生時にトレースが実行されていた場合)
- コア・ファイルを含むディレクトリー
- DB2FODC ログ・ファイル
 - 手動 FODC では、「ログ」ファイルを 1 つだけ使用します。ハングの場合は db2fodc_hang.log、不正ページの場合は db2fodc_badpage.log です。
- データ破損に関連した情報
 - プロセス情報: ps (UNIX の場合) と db2pd -edus 出力
 - db2support によって現在収集されている追加情報 (オプション)
 - errpt -a 出力 (AIX)
 - UNIX プラットフォームのシステム・ログ。例えば、SunOS では /var/adm/messages、HP/UX では /var/adm/syslog.log になります。これが行わ

れるのは、これらのファイルが収集されている可能性がある場合です (Linux の場合は、syslog ファイルをコピーするためのルート・アクセスが必要です)。

自動 FODC データ生成

障害が発生したときに、自動 First Occurrence Data Capture (FODC) が有効になっていると、症状に基づいてデータが収集されます。障害の診断に必要な情報に応じて、それぞれ固有のデータが収集されます。

障害の発生元を通知するために、1 つまたは多数のメッセージ (「重大」と定義されているメッセージなど) が使用されます。

トラップ・ファイルには、以下のような情報が含まれます。

- 仮想メモリーの空き容量
- トラップが発生した時点での製品の構成パラメーターに関連した値とレジストリ変数
- トラップが発生した時点で DB2 製品が使用していたメモリーの概算量
- 障害のコンテキストを示す情報

ロー・スタック・ダンプが ASCII トラップ・ファイルに含まれていることもあります。

データベース・マネージャー内の各コンポーネントに固有のダンプ・ファイルが、該当する FODC パッケージ・ディレクトリーに格納されています。

First Occurrence Data Capture 情報

First Occurrence Data Capture (FODC) は、DB2 インスタンスに関するシナリオ・ベースのデータをキャプチャーするためのプロセスです。FODC は、DB2 ユーザーが特定の症状に基づいて手動で呼び出すこともできれば、事前定義のシナリオまたは症状が検出されたときに自動的に呼び出すこともできます。この情報があれば、診断情報を取得するためにエラーを再現する必要性は少なくなります。

FODC 情報は、以下のファイルで確認できます。

管理通知ログ (「instance_name.nfy」)

- オペレーティング・システム: すべて
- デフォルト・ロケーション:
 - Linux® および UNIX®: *diagpath* データベース・マネージャー構成パラメーターで指定されたディレクトリーにあります。
 - Windows®: 「イベント ビューア」ツールを使用します (「スタート」 > 「コントロール パネル」 > 「管理ツール」 > 「イベント ビューア」)。
- インスタンスが作成されると自動的に作成されます。
- 有効なイベントが発生すると、DB2 は管理通知ログに情報を書き込みます。情報は、データベースおよびシステム管理者が使用するためのものです。このファイルに記録されるメッセージのタイプは、*notifylevel* 構成パラメーターによって決定されます。

DB2 診断ログ (「db2diag.log」)

- オペレーティング・システム: すべて
- デフォルト・ロケーション: *diagpath* データベース・マネージャー構成パラメーターで示されるディレクトリーにあります。
- インスタンスが作成されると自動的に作成されます。
- このテキスト・ファイルには、インスタンスで発生するエラーおよび警告に関する診断情報が入ります。この情報は、トラブルシューティングに使用され、IBM® ソフトウェア・サポートが使用するためのものです。このファイルに記録されるメッセージのタイプは、*diaglevel* データベース・マネージャー構成パラメーターによって決定されます。

DB2 管理サーバー (DAS) 診断ログ (「db2dasdiag.log」)

- オペレーティング・システム: すべて
- デフォルト・ロケーション:
 - Linux および UNIX: DASHOME/das/dump にあります。DASHOME は、DAS 所有者のホーム・ディレクトリーです。
 - Windows: DAS ホーム・ディレクトリーの「dump」フォルダーにあります。例えば、C:\Program Files\IBM\SQLLIB\DB2DAS00\dump のようになります。
- DAS が作成されると自動的に作成されます。
- このテキスト・ファイルには、DAS で発生するエラーおよび警告に関する診断情報が入ります。

DB2 イベント・ログ (「db2eventlog.xxx」。「xxx」はデータベース・パーティションの番号)

- オペレーティング・システム: すべて
- デフォルト・ロケーション: *diagpath* データベース・マネージャー構成パラメーターで指定されるディレクトリーにあります。
- インスタンスが作成されると自動的に作成されます。
- DB2 イベント・ログ・ファイルとは、データベース・マネージャーで生じるインフラストラクチャー・レベルのイベントの循環ログのことです。このファイルはサイズが一定で、インスタンスの実行時に記録される特定のイベントに関する循環バッファの働きをします。インスタンスを停止するたびに、以前のイベント・ログは付加されずに置き換えられます。インスタンスがトラップした場合は、db2eventlog.XXX.crash ファイルも生成されます。これらのファイルは、IBM ソフトウェア・サポートが使用するためのものです。

DB2 コールアウト・スクリプト (db2cos) 出力ファイル

- オペレーティング・システム: すべて
- デフォルト・ロケーション: *diagpath* データベース・マネージャー構成パラメーターで指定されるディレクトリーにあります。
- パニック、トラップ、またはセグメンテーション違反が発生すると自動的に作成されます。db2pdcfg コマンドを使って指定された特定の問題のシナリオで作成される場合もあります。

- デフォルトの db2cos スクリプトは、db2pd コマンドを呼び出してアンタッチ方式で情報を集めます。db2cos 出力ファイルの内容は、db2cos スクリプトに含まれるコマンドに応じて異なります。
- db2cos スクリプトは、出荷時に bin/ ディレクトリーに入っています。UNIX の場合、このディレクトリーは読み取り専用です。このスクリプトの独自の変更可能バージョンを作成する場合は、db2cos スクリプトを adm/ ディレクトリーにコピーしてください。そのバージョンのスクリプトは、自由に変更できます。スクリプトが adm/ ディレクトリーにある場合は、そのバージョンが実行されます。そうでない場合は、bin/ ディレクトリーにあるデフォルト・バージョンが実行されます。

ダンプ・ファイル

- オペレーティング・システム: すべて
- デフォルト・ロケーション: *diagpath* データベース・マネージャー構成パラメーターで指定されるディレクトリーにあります。
- 特定の問題のシナリオが現れると自動的に作成されます。
- エラー状態によっては、失敗したプロセス ID の名前が付いたバイナリー・ファイルに追加情報が記録されます。これらのファイルは、IBM ソフトウェア・サポートが使用するためのものです。

トラップ・ファイル

- オペレーティング・システム: すべて
- デフォルト・ロケーション: *diagpath* データベース・マネージャー構成パラメーターで指定されるディレクトリーにあります。
- インスタンスが異常終了すると自動的に作成されます。db2pd コマンドを使って随意に作成することもできます。
- データベース・マネージャーは、トラップやセグメンテーション違反、例外などにより処理が続行できない時、トラップ・ファイルを生成します。

コア・ファイル

- オペレーティング・システム: Linux および UNIX
- デフォルト・ロケーション: *diagpath* データベース・マネージャー構成パラメーターで指定されるディレクトリーにあります。
- DB2 インスタンスが異常終了するとオペレーティング・システムによって作成されます。
- 特に、コア・イメージには、DB2 のほとんどまたはすべてのメモリー割り振りが組み込まれます。その情報は、問題を説明するために必要な場合があります。

トラップ、パニック、データ破壊、停止状態 (データベースやインスタンスを使用できない状態) などのトラブルシューティングの作業を容易にするために、特定の障害タイプに関連した DB2 診断ファイルは、新しい First Occurrence Data Capture (FODC) ディレクトリーに送られます。このディレクトリーは、インスタンス診断パスに作成されます。

注: この FODC ディレクトリーの名前は FODC_<symptom>_<timestamp> です。DB2 診断ファイルは FODC パッケージに入ります。このディレクトリーとパッケージが作成されるのは、AIX と Linux のオペレーティング・システムに限られません。

DB2 Query Patroller と First Occurrence Data Capture (FODC)

DB2 Query Patroller の問題を調査する必要がある場合は、起きた障害の推定原因に関する情報が含まれているログがあります。

qpdiaq.log

- オペレーティング・システム: すべて
- デフォルト・ロケーション: *diagpath* データベース・マネージャー構成パラメーターで示されるディレクトリーにあります。
- Query Patroller システムがアクティブになると自動的に作成されます。
- Query Patroller についての情報および診断レコードが含まれます。この情報は、トラブルシューティングに使用され、IBM ソフトウェア・サポートが使用するためのものです。

qpmigrate.log

- オペレーティング・システム: すべて
- デフォルト・ロケーション: *diagpath* データベース・マネージャー構成パラメーターで示されるディレクトリーにあります。
- *qpmigrate* ユーティリティーによって自動的に作成されます。*qpmigrate* コマンドは、Query Patroller のインストール時に暗黙的に実行することも (Query Patroller を実行する対象の既存のデータベースを指定した場合)、インストール後に明示的に実行することもできます。
- Query Patroller があるバージョンから別のバージョンへマイグレーションされる際の情報およびエラー・メッセージをキャプチャーします。これは、Query Patroller 管理者が使用するためのものです。

qpsetup.log

- オペレーティング・システム: すべて
- デフォルト・ロケーション: *diagpath* データベース・マネージャー構成パラメーターで示されるディレクトリーにあります。
- *qpsetup* ユーティリティーによって自動的に作成されます。*qpsetup* コマンドは、Query Patroller のインストール時に暗黙的に実行することも (Query Patroller を実行する対象の既存のデータベースを指定した場合)、インストール後に明示的に実行することもできます。
- *qpsetup* ユーティリティーの実行中に発生する情報およびエラー・メッセージをキャプチャーします。これは、Query Patroller 管理者が使用するためのものです。

qpuser.log

- オペレーティング・システム: すべて
- デフォルト・ロケーション: *diagpath* データベース・マネージャー構成パラメーターで示されるディレクトリーにあります。

- Query Patroller システムがアクティブになると自動的に作成されます。
- Query Patroller に関する通知メッセージが含まれています。例えば、Query Patroller が停止した時や開始した時を示します。これは、Query Patroller 管理者が使用するためのものです。

First Occurrence Data Capture (FODC) を使用するモニター機能と監査機能

モニター機能または監査機能の問題を調査する必要がある場合は、起きた障害の推定原因に関する情報が含まれているログがあります。

DB2 監査ログ (db2audit.log)

- オペレーティング・システム: すべて
- デフォルト・ロケーション:
 - Windows: \$DB2PATH¥instance_name¥security ディレクトリーにあります。
 - Linux および UNIX: \$HOME¥sqllib¥security ディレクトリーにあります (\$HOME はインスタンス所有者のホーム・ディレクトリー)。
- db2audit 機能の開始時に作成されます。
- 一連の事前定義されたデータベース・イベントについて DB2 監査機能によって生成された監査レコードが含まれます。

DB2 ガバナー・ログ (mylog.x、x はガバナーが実行されているデータベース・パーティションの数)

- オペレーティング・システム: すべて
- デフォルト・ロケーション:
 - Windows: \$DB2PATH¥instance_name¥log ディレクトリーにあります。
 - Linux および UNIX: \$HOME¥sqllib¥log ディレクトリーにあります (\$HOME はインスタンス所有者のホーム・ディレクトリー)。
- ガバナー・ユーティリティーの使用時に作成されます。ログ・ファイル名のベースは、db2gov コマンドで指定されます。
- ガバナー・デーモンによって実行されたアクション (例えば、アプリケーションの強制実行、ガバナー構成ファイルの読み取り、ユーティリティーの開始または終了) に関する情報や、エラーと警告に関する情報を記録します。

イベント・モニター・ファイル (例えば、00000000.evt)

- オペレーティング・システム: すべて
- デフォルト・ロケーション: ファイル・イベント・モニターの作成時に、すべてのイベント・レコードは CREATE EVENT MONITOR ステートメントで指定されたディレクトリーに書き込まれます。
- イベントの発生時にイベント・モニターによって生成されます。
- イベント・モニターと関連したイベント・レコードが含まれます。

First Occurrence Data Capture (FODC) を使用するグラフィック・ツール

コマンド・エディター、データウェアハウス・センター、インフォメーション・カタログ・センターの問題を調査する必要がある場合は、起きた障害の推定原因に関する情報が含まれているログがあります。

コマンド・エディターのログ

- オペレーティング・システム: すべて
- デフォルト・ロケーション: このログ・ファイルの名前およびロケーションは、DB2 ツールバーの「コマンド・エディター」ページを使用して指定されます。パスが指定されていない場合、ログは Windows では `$DB2PATH\sqlib\tools` ディレクトリーに、Linux および UNIX では `$HOME/sqlib/tools` ディレクトリー (HOME はインスタンス所有者のホーム・ディレクトリー) に保管されます。
- コマンド・エディターで「**コマンド履歴をファイルに記録する**」を選択し、ファイルとロケーションを指定した場合に作成されます。
- コマンド・エディターからのコマンドおよびステートメントの実行履歴が含まれます。

データウェアハウス・センターの IWH2LOGC.log ファイル

- オペレーティング・システム: すべて
- デフォルト・ロケーション: `VWS_LOGGING` 環境変数で指定されるディレクトリーにあります。デフォルト・パスは、Windows では `$DB2PATH\sqlib\logging` ディレクトリー、Linux および UNIX では `$HOME/sqlib/logging` ディレクトリー (HOME はインスタンス所有者のホーム・ディレクトリー) です。
- ロガーが停止した場合にデータウェアハウス・センターによって自動的に作成されます。
- データウェアハウス・センターおよび OLE サーバーによって書き込まれたメッセージで、ロガーが停止した状況で送信できなかったものが含まれます。このログは、データウェアハウス・センターの「ログ・ビューアー」ウィンドウを使用して表示できます。

データウェアハウス・センターの IWH2LOG.log ファイル

- オペレーティング・システム: すべて
- デフォルト・ロケーション: `VWS_LOGGING` 環境変数で指定されるディレクトリーにあります。デフォルト・パスは、Windows では `$DB2PATH\sqlib\logging` ディレクトリー、Linux および UNIX では `$HOME/sqlib/logging` ディレクトリー (HOME はインスタンス所有者のホーム・ディレクトリー) です。
- データウェアハウス・センターが開始できない場合、またはトレースがアクティブにされたときにデータウェアハウス・センターによって自動的に作成されます。
- データウェアハウス・センターのロガーが開始できず、データウェアハウス・センターのログ (IWH2LOGC.log) に書き込めない状況についての診断情報が含まれます。このログは、データウェアハウス・センターの「ログ・ビューアー」ウィンドウを使用して表示できます。

データウェアハウス・センターの IWH2SERV.log ファイル

- オペレーティング・システム: すべて
- デフォルト・ロケーション: VWS_LOGGING 環境変数で指定されるディレクトリーにあります。デフォルト・パスは、Windows では \$DB2PATH¥sqlib¥logging ディレクトリー、Linux および UNIX では \$HOME/sqlib/logging ディレクトリー (HOME はインスタンス所有者のホーム・ディレクトリー) です。
- データウェアハウス・センターのサーバー・トレース機能によって自動的に作成されます。
- データウェアハウス・センターのスタートアップ・メッセージが含まれます。また、サーバー・トレース機能によって作成されるメッセージを記録します。このログは、データウェアハウス・センターの「ログ・ビューアー」ウィンドウを使用して表示できます。

インフォメーション・カタログ・センターのタグ・ファイル EXPORT ログ

- オペレーティング・システム: すべて
- デフォルト・ロケーション: エクスポートされるタグ・ファイル・パスおよびログ・ファイル名は、インフォメーション・カタログ・センターのエクスポート・ツールの「オプション」タブで指定されます。
- インフォメーション・カタログ・センターのエクスポート・ツールによって生成されます。
- タグ・ファイルのエクスポート情報が含まれます。例えば、エクスポート・プロセスの開始と停止の日時などです。また、エクスポート操作中に発生したエラー・メッセージも含まれます。

インフォメーション・カタログ・センターのタグ・ファイル IMPORT ログ

- オペレーティング・システム: すべて
- デフォルト・ロケーション: インポートされるタグ・ファイル・パスおよびログ・ファイル名は、インフォメーション・カタログ・センターのインポート・ツールで指定されます。
- インフォメーション・カタログ・センターのインポート・ツールによって生成されます。
- タグ・ファイルのインポート履歴情報が含まれます。例えば、インポート・プロセスの開始と停止の日時などです。また、インポート操作中に発生したエラー・メッセージも含まれます。

管理通知ログ・ファイルについて

管理通知ログ・ファイルの紹介として、ログに記録されるエラー・タイプのさまざまなレベルと、エラー収集レベルの設定方法の説明があります。ログがいったん作成されると、特定のデータベース問題に関してさらに理解できるように、ログ・レコードの解釈方法についてのガイダンスが与えられます。

管理通知ログ

DB2 データベース・マネージャーが管理通知ログに書き込むのは、DB2 ユーティリティーの状況 (REORG、BACKUP など)、クライアント・アプリケーション・エラ

一、サービス・クラスの変更、ライセンス交付アクティビティ、ログ・ファイル・パスおよびストレージの問題、モニターおよび索引付けアクティビティ、表スペースの問題などの情報です。データベース管理者はこの情報を使用することにより、問題の診断やデータベースの調整を行ったり、単にデータベースのモニターを行ったりすることもできます。

管理通知ログ・メッセージは、標準化されたメッセージ・フォーマットで db2diag.log にも記録されます。

通知メッセージには、提供されている SQLCODE を補足する追加情報が備えられています。イベントのタイプ、および収集される情報の詳しさのレベルは、NOTIFYLEVEL 構成パラメーターによって決定されます。

管理通知ログ・ファイルのエラー・キャプチャー・レベルの設定

DB2 がどのような情報を管理通知ログに記録するかは、NOTIFYLEVEL 設定によって決まります。

- 現在の設定値を確認するには、GET DBM CFG コマンドを発行します。

以下のような変数を見つけてください。

```
Notify Level (NOTIFYLEVEL) = 3
```

- 設定値を変更するには、UPDATE DBM CFG コマンドを使用します。例えば、以下のようにします。

```
DB2 UPDATE DBM CFG USING NOTIFYLEVEL X
```

ここで、X はご希望の通知レベルです。

管理通知ログ・ファイル項目の解釈

テキスト・エディターを使って、問題が発生した疑いのあるマシンの管理通知ログ・ファイルを表示します。最新のイベントは、ファイルの一番下に記録されています。一般的に、各項目には以下のような部分があります。

- タイム・スタンプ
- エラーが報告された場所。アプリケーション ID を使って、サーバーとクライアントのログにあるアプリケーションに関係した項目を見つけることができます。
- エラーを説明する診断メッセージ (通常「DIA」または「ADM」で始まる)。
- 利用可能なサポート・データ (SQLCA データ構造、追加のダンプ・ファイルまたはトラップ・ファイルの場所を指すポインターなど)。

管理ログは、データベース内のすべてのログと同じく、どんどん大きくなっていきます。ファイルに書き込まれる内容によって、他のログよりも早く大きくなるログもあります。ログが大きくなりすぎたら、バックアップしてから消去します。新しいログは、システムで次に必要になったときに自動生成されます。

以下の例はサンプル・ログ項目のヘッダー情報で、ログのすべての部分が示されています。

注: すべてのログ項目にこれらの部分がすべて含まれるとは限りません。

```
2006-02-15-19.33.37.630000 1 Instance:DB2 2 Node:000 3
PID:940(db2syscs.exe) TID: 660 4 Appid:*LOCAL.DB2.020205091435 5
recovery manager 6 sqlpresr 7 Probe:1 8 Database:SAMPLE 9
ADM1530E 10 Crash recovery has been initiated. 11
```

凡例:

1. メッセージのタイム・スタンプ。
2. メッセージを生成したインスタンスの名前。
3. マルチパーティション・システムの場合、メッセージを生成したデータベース・パーティション。(非パーティション・データベースでは、この値は "000" です。)
4. メッセージが生成される原因になったプロセスのプロセス ID (PID) (プロセスの名前、スレッド ID (TID) が後に続きます)。
- 5.

プロセスが作動しているアプリケーションの識別番号。この例では、メッセージを生成したプロセスは、ID が *LOCAL.DB2.020205091435 であるアプリケーションのプロセスとして作動しています。

この値は *appl_id* モニター・エレメント・データと同じになります。この値を解釈する方法の詳細については、*appl_id* モニター・エレメントの資料を参照してください。

特定のアプリケーション ID の詳細情報を識別するには、以下のいずれかを行ってください。

- DB2 サーバー上で LIST APPLICATIONS コマンドを使用するか、DB2 Connect™ ゲートウェイ上で LIST DCS APPLICATIONS を使用して、アプリケーション ID のリストを表示します。このリストから、エラーが起きたクライアントに関する情報を判別できます (ノード名、TCP/IP アドレスなど)。
 - GET SNAPSHOT FOR APPLICATION コマンドを使用して、アプリケーション ID のリストを表示します。
6. メッセージを書き込んでいる DB2 コンポーネント。db2AdminMsgWrite API を使ってユーザー・アプリケーションによって書き込まれたメッセージの場合、このコンポーネントは 『User Application』 となります。
 7. メッセージを出している関数の名前。この関数は、メッセージを書き込んでいる DB2 コンポーネントの中で機能します。db2AdminMsgWrite API を使ってユーザー・アプリケーションによって書き込まれたメッセージの場合、この関数は 『User Function』 となります。
 8. ユニークな内部 ID。この番号を使って、DB2 お客様サポートや開発部門は DB2 ソース・コード内のメッセージを報告した場所を見つけることができます。
 9. エラーが発生したデータベース。
 10. エラーのタイプと番号を 16 進数コードで示すメッセージ (存在する場合)。
 11. ログに記録されたイベントを説明するメッセージ・テキスト (存在する場合)。

DB2 診断ログ (db2diag.log) ファイルについて

DB2 診断ログ・ファイルの内容として、提示されるログ内容のさまざまなレベルの説明が、エラー収集レベルの設定方法の指示と共に含まれています。

診断ログ・ファイルのエラー・キャプチャー・レベルの設定

DB2 診断ログ・ファイルには、DB2 によって記録されるテキスト情報が格納されます。この情報はトラブルシューティングに使用され、主として DB2 お客様サポートのためのものです。

DB2 が db2diag.log 内にどのような情報を記録するかは、DIAGLEVEL 設定によって決まります。

- 現在の設定値を確認するには、コマンド GET DBM CFG を発行します。

以下のような変数を見つけてください。

```
Diagnostic error capture level          (DIAGLEVEL) = 3
```

- 値を動的に変更するには、UPDATE DBM CFG コマンドを使用します。

データベース・マネージャの構成パラメーターをオンラインで変更するには、以下のようにします。

```
db2 attach to <instance-name>
db2 update dbm cfg using <parameter-name> <value>
db2 detach
```

例えば、以下のようにします。

```
DB2 UPDATE DBM CFG USING DIAGLEVEL X
```

ここで、X はご希望の通知レベルです。再現可能な問題を診断する場合には、サポート担当員はトラブルシューティング実行時に DIAGLEVEL 4 を使用するよう勧めるでしょう。

db2diag.log ファイルの情報レコードの解釈

db2diag.log 内の最初のメッセージとして、常に情報レコードが記録されます。

以下は、情報レコードの例です。

```
2006-02-09-18.07.31.059000-300 I1H917          LEVEL: Event
PID       : 3140          TID    : 2864          PROC   : db2start.exe
INSTANCE: DB2           NODE    : 000
FUNCTION: DB2 UDB, RAS/PD component, _pdlogInt, probe:120
START    : New Diagnostic Log file
DATA #1 : Build Level, 124 bytes
Instance "DB2" uses "32" bits and DB2 code release "SQL09010"
with level identifier "01010107".
Informational tokens are "DB2 v9.1.0.190", "s060121", "", Fix Pack "0".
DATA #2 : System Info, 1564 bytes
System: WIN32_NT MYSRVR Service Pack 2 5.1 x86 Family 15, model 2, stepping 4
CPU: total:1 online:1 Cores per socket:1 Threading degree per core:1
Physical Memory(MB): total:1024 free:617 available:617
Virtual Memory(MB): total:2462 free:2830
Swap Memory(MB): total:1438 free:2213
Information in this record is only valid at the time when this file was created
(see this record's time stamp)
```

情報レコードは、それぞれの論理パーティション上の db2start ごとに出力されます。このため、論理パーティションごとに 1 つ、合わせて複数の情報レコードが存在します。情報レコードには各パーティションごとに異なるメモリー値が含まれるため、この情報が役立つことがあります。

診断ログ・ファイル項目の解釈

db2diag.log 分析ツール (db2diag) は、db2diag.log ファイルのフィルター処理とフォーマットに使用します。標準化されたメッセージ・フォーマットで管理通知ログ・メッセージの追加部分が db2diag.log に記録された場合は、データベースで何が起こったのかを理解するために、db2diag.log を最初に表示することをお勧めします。

db2diag を使用する代わりに、テキスト・エディターを使用して、問題が発生した可能性のあるマシンの診断ログ・ファイルを表示できます。最新のイベントは、ファイルの一番下に記録されています。

注: 管理ログと診断ログは、*継続的に* 大きくなります。大きくなりすぎた場合は、バックアップを取って、ファイルを消去してください。次にファイル・セットが必要になったとき、システムによって新しいファイル・セットが自動生成されます。

以下の例はサンプル・ログ項目のヘッダー情報で、ログのすべての部分が示されています。

注: すべてのログ項目にこれらの部分がすべて含まれるとは限りません。すべての db2diag.log レコードに必ず含まれるのは、最初のいくつかのフィールド (タイム・スタンプから TID まで) および FUNCTION だけです。

```
2007-05-18-14.20.46.973000-240 1 I27204F655 2 LEVEL: Info 3  
PID : 3228 4 TID : 8796 5 PROC : db2syscs.exe 6  
INSTANCE: DB2MPP 7 NODE : 002 8 DB : WIN3DB1 9  
APPHDL : 0-51 10 APPID: 9.26.54.62.45837.070518182042 11  
AUTHID : UBADM 12  
EDUID : 8796 13 EDUNAME: db2agntp 14 (WIN3DB1) 2  
FUNCTION: 15 DB2 UDB, data management, sqlInitDBCB, probe:4820  
DATA #1 : 16 String, 26 bytes  
Setting ADC Threshold to:  
DATA #2 : unsigned integer, 8 bytes  
1048576
```

凡例:

1. メッセージのタイム・スタンプとタイム・ゾーン。

注: db2diag.log のタイム・スタンプには、タイム・ゾーンが含まれます。例えば、2006-02-13-14.34.35.965000-300 の場合、"-300" は UTC (協定世界時、以前は GMT と呼ばれていた) とアプリケーション・サーバーの現地時間との差異です (単位は 分)。つまり -300 とは UTC マイナス 5 時間という意味で、米国の東部標準時 (EST) がこれに該当します。

2. レコード ID フィールド。db2diag.log のレコード ID は、DB2 診断ログが作成されたプラットフォームで現在のメッセージが記録されている場所のファイル・オフセット (例えば「27204」) とメッセージの長さ (例えば「655」) を示します。
3. エラー・メッセージに関連した診断レベル。例えば、Info (通知)、Warning (警告)、Error (エラー)、Severe (重大)、または Event (イベント)。

4. プロセス ID
5. スレッド ID
6. プロセス名
7. メッセージを生成したインスタンスの名前。
8. マルチパーティション・システムの場合、メッセージを生成したデータベース・パーティション。(非パーティション・データベースでは、この値は "000" です。)
9. データベース名
10. アプリケーション・ハンドル。この値は、db2pd 出力ファイルおよびロック・ダンプ・ファイルの値と同じになります。コーディネーター・パーティション番号の後にダッシュ (-)、その後にコーディネーター索引番号が示されます。
11. プロセスが作動しているアプリケーションの識別番号。この例では、メッセージを生成したプロセスは、ID 9.26.54.62.45837.070518182042 のアプリケーションのために作動しています。

TCP/IP によって生成されるアプリケーション ID は、3 つのセクションから構成されます。

1. **IP アドレス:** 最大 8 個の 16 進文字を使用する 32 ビットの数値として表されます。
2. **ポート番号:** 4 個の 16 進文字として表されます。
3. このアプリケーションのインスタンスを表す**ユニーク ID**。

注: 16 進数の IP アドレスまたはポート番号が 0 から 9 で始まる場合、それぞれ G から P に変更されます。例えば、"0" は "G" に、"1" は "H" にそれぞれマップされます。IP アドレス AC10150C.NA04.006D07064947 は次のように解釈されます: IP アドレスは AC10150C のままで、172.16.21.12 に変換されます。ポート番号は NA04 です。最初の文字 "N" は "7" にマップされます。したがって、16 進形式のポート番号は 7A04 となり、これは 10 進形式で 31236 に変換されます。

この値は *appl_id* モニター・エレメント・データと同じになります。この値を解釈する方法の詳細については、*appl_id* モニター・エレメントの資料を参照してください。

特定のアプリケーション ID の詳細情報を識別するには、以下のいずれかを行ってください。

- DB2 サーバー上で LIST APPLICATIONS コマンドを使用するか、DB2 Connect ゲートウェイ上で LIST DCS APPLICATIONS を使用して、アプリケーション ID のリストを表示します。このリストから、エラーが起きたクライアントに関する情報を判別できます (データベース・パーティション名、TCP/IP アドレスなど)。
- GET SNAPSHOT FOR APPLICATION コマンドを使用して、アプリケーション ID のリストを表示します。
- db2pd -applications -db <sample> コマンドを使用します。

- 12 許可 ID。

- 13 エンジン・ディスパッチ可能単位 ID。
- 14 エンジン・ディスパッチ可能単位の名前。
- 15. メッセージを書き込んだ製品の製品名 (「DB2」)、コンポーネント名 (「data management」)、関数名 (「sqlInitDBCB」)、関数内のプローブ点 (「4820」)。
- 16. 呼び出した関数から返された情報。複数のデータ・フィールドが返される可能性があります。

サンプル db2diag.log の項目をすでに見ましたので、可能なすべてのフィールドのリストをご覧ください。

```
<timestamp><timezone>          <recordID>          LEVEL: <level> (<source>)
PID      : <pid>                TID   : <tid>        PROC  : <procName>
INSTANCE: <instance>          NODE  : <node>       DB    : <database>
APPHDL   : <appHandle>        APPID : <appID>
AUTHID   : <authID>
EDUID    : <eduID>            EDUNAME: <engine dispatchable unit name>
FUNCTION: <prodName>, <compName>, <funcName>, probe:<probeNum>
MESSAGE  : <messageID> <msgText>
CALLED   : <prodName>, <compName>, <funcName>  OSERR: <errorName> (<errno>)
RETCODE  : <type>=<retCode> <errorDesc>
ARG #N   : <typeTitle>, <typeName>, <size> bytes
... argument ...
DATA #N  : <typeTitle>, <typeName>, <size> bytes
... data ...
```

上記の例で説明されなかったフィールドは以下のとおりです。

•

<source> は、ログに記録されたエラーの発生源を示します。(サンプルの最初の行の最後にあります。)可能な値は以下のとおりです。

- origin - メッセージは、エラーが発生した場所 (開始点) である関数によって記録されました
- OS - エラーはオペレーティング・システムによって生成されました
- received - エラーは別のプロセス (クライアント/サーバー) から受信されました
- sent - エラーは別のプロセス (クライアント/サーバー) に送信されました

•

MESSAGE は記録されたメッセージです。以下のもので構成されます。

- <messageID> - メッセージ番号 (例えば ECF=0x9000004A または DIA8604C)
- <msgText> - エラーの説明

CALLED フィールドが存在する場合、<msgText> は、CALLED 関数から戻されたエラーがメッセージを記録した関数 (FUNCTION フィールドで識別される) に与えた影響を表します。

•

CALLED はエラーを戻した関数です。以下のもので構成されます。

- <prodName> - 製品名 ("OS"、"DB2"、"DB2 Tools"、または "DB2 Common")
- <compName> - コンポーネント名 (システム呼び出しの場合は '-')

- <funcName> - 呼び出された関数名
- OSERR は、CALLED システム呼び出しによって戻されたオペレーティング・システム・エラーです。(CALLED と同じ行の最後にあります。)以下のもので構成されます。
 - <errorName> - システム固有のエラー名
 - <errno> - オペレーティング・システムのエラー番号
- ARG セクションは、エラーを戻した関数呼び出しの引数をリストします。以下のもので構成されます。
 - <N> - 関数の呼び出しにおける引数の位置
 - <typeTitle> - N 番目の引数のタイプ名に関連したラベル
 - <typeName> - ログに記録された引数のタイプ名
 - <size> - ログに記録された引数のサイズ
- DATA には、ロギング機能によってダンプされた追加のデータが示されます。以下のもので構成されます。
 - <N> - ダンプ対象のデータ・オブジェクトの順序番号
 - <typeTitle> - ダンプ対象のデータのラベル
 - <typeName> - ログに記録されたデータ・フィールドのタイプ名 (例えば PD_TYPE_UINT32、PD_TYPE_STRING)
 - <size> - データ・オブジェクトのサイズ

db2cos (コールアウト・スクリプト) 出力ファイル

データベース・マネージャーが、パニック、トラップ、セグメンテーション違反、例外のために処理を続けられなくなると、デフォルトで db2cos スクリプトが呼び出されます。それぞれのデフォルトの db2cos スクリプトは、db2pd コマンドを呼び出してアンラッチ方式で情報を収集します。db2cos スクリプトの名前は、db2cos_hang、db2cos_trap などのような形になります。それぞれのスクリプトは同じように動作します。例外は db2cos_hang で、これは db2fodc ツールから呼び出されます。

デフォルトの db2cos スクリプトは、bin ディレクトリーにあります。UNIX オペレーティング・システムの場合、このディレクトリーは読み取り専用です。db2cos スクリプト・ファイルは、adm ディレクトリーにコピーし、必要に応じてその場所を変更することも可能です。db2cos スクリプトが adm ディレクトリーにあれば、そのスクリプトが実行されます。そうでない場合は、bin ディレクトリーにあるスクリプトが実行されます。

複数のパーティション構成では、トラップを検出したパーティションのトラッピングするエージェントについてのみ、スクリプトが呼び出されます。他のパーティションから情報を集める必要がある場合は db2_all コマンドを使用するように db2cos スクリプトを更新し、すべてのパーティションが同じマシンにある場合は db2pd コマンドに -alldbpartitionnums オプションを指定します。

db2cos の呼び出しをトリガーするシグナルのタイプも、db2pdcfg -cos コマンドによって構成できます。デフォルトの構成は、パニックやトラップ発生時には db2cos スクリプトを実行します。ただし、生成されたシグナルはデフォルトでは db2cos スクリプトを起動しません。

パニック、トラップ、セグメンテーション違反、または例外が起きたときのイベント順序は以下のとおりです。

1. トラップ・ファイルが作成される
2. シグナル・ハンドラーが呼び出される
3. db2cos スクリプトが呼び出される (db2cos 設定が有効な場合)
4. 項目が管理通知ログに記録される
5. 項目が db2diag.log に記録される

db2cos スクリプトの db2pd コマンドによって集められるデフォルト情報には、オペレーティング・システム、インストールされている DB2 製品のバージョンとサービス・レベル、データベース・マネージャー、およびデータベース構成についての詳細に加え、エージェント、メモリー・プール、メモリー・セット、メモリー・ブロック、アプリケーション、ユーティリティ、トランザクション、バッファ・プール、ロック、トランザクション・ログ、表スペース、およびコンテナの状態についての情報が含まれます。さらに、動的キャッシュ、静的キャッシュ、およびカタログ・キャッシュの状態、表と索引の統計、リカバリー状況についての情報や、再最適化された SQL ステートメントとアクティブなステートメントのリストも提供されます。さらに情報を収集する必要がある場合は、他のコマンドを追加して db2cos スクリプトを更新します。

デフォルトの db2cos スクリプトを呼び出すと、DIAGPATH データベース・マネージャー構成パラメーターで指定されているディレクトリーに出力ファイルが生成されます。それらのファイルには、XXX.YYY.ZZZ.cos.txt という名前が付けられます (XXX はプロセス ID (PID)、YYY はスレッド ID (TID)、ZZZ はデータベース・パーティション番号 (単一パーティション・データベースの場合は 000) です)。複数のスレッドがトラップされる場合は、それぞれのスレッドごとに、db2cos スクリプトが別々に呼び出されます。PID および TID の組み合わせが複数回発生する場合、データはファイルに追加されます。タイム・スタンプが表記されるので、出力の反復を区別することができます。

db2cos 出力ファイルには、db2cos スクリプトに指定されているコマンドに応じて、さまざまな情報が組み込まれます。デフォルトのスクリプトを変更しなければ、以下のような項目が出力されます (詳細な db2pd 出力が続きます)。

```
2005-10-14-10.56.21.523659
PID       : 782348           TID  : 1           PROC  : db2cos
INSTANCE: db2inst1        NODE  : 0           DB    : SAMPLE
APPHDL   :                 APPID: *LOCAL.db2inst1.051014155507
FUNCTION: oper system services, sqloEDUCodeTrapHandler, probe:999
EVENT    : Invoking /home/db2inst1/sqlllib/bin/db2cos from
oper system services sqloEDUCodeTrapHandler
Trap Caught
```

```
Instance db2inst1 uses 64 bits and DB2 code release SQL09010
```

```
...
```

```
Operating System Information:
```

```
OSName:   AIX
```

```
NodeName: n1
Version: 5
Release: 2
Machine: 000966594C00
```

...

db2diag.log にはオカレンスに関連した項目も含まれます。例えば、以下のようになります。

```
2005-10-14-10.42.17.149512-300 I19441A349          LEVEL: Event
PID      : 782348          TID : 1          PROC : db2sysc
INSTANCE: db2inst1      NODE : 000
FUNCTION: DB2 UDB, trace services, pdInvokeCalloutScript, probe:10
START    : Invoking /home/db2inst1/sqllib/bin/db2cos from oper system
services sqloEDUCodeTrapHandler

2005-10-14-10.42.23.173872-300 I19791A310          LEVEL: Event
PID      : 782348          TID : 1          PROC : db2sysc
INSTANCE: db2inst1      NODE : 000
FUNCTION: DB2 UDB, trace services, pdInvokeCalloutScript, probe:20
STOP     : Completed invoking /home/db2inst1/sqllib/bin/db2cos

2005-10-14-10.42.23.519227-300 E20102A509          LEVEL: Severe
PID      : 782348          TID : 1          PROC : db2sysc
INSTANCE: db2inst1      NODE : 000
FUNCTION: DB2 UDB, oper system services, sqloEDUCodeTrapHandler, probe:10
MESSAGE  : ADM0503C An unexpected internal processing error has occurred. ALL
          DB2 PROCESSES ASSOCIATED WITH THIS INSTANCE HAVE BEEN SHUTDOWN.
          Diagnostic information has been recorded. Contact IBM Support for
          further assistance.

2005-10-14-10.42.23.520111-300 E20612A642          LEVEL: Severe
PID      : 782348          TID : 1          PROC : db2sysc
INSTANCE: db2inst1      NODE : 000
FUNCTION: DB2 UDB, oper system services, sqloEDUCodeTrapHandler, probe:20
DATA #1 : Signal Number Recieved, 4 bytes
11
DATA #2 : Siginfo, 64 bytes
0x0FFFFFFFFFD5C0 : 0000 000B 0000 0000 0000 0009 0000 0000 .....
0x0FFFFFFFFFD5D0 : 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0FFFFFFFFFD5E0 : 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0FFFFFFFFFD5F0 : 0000 0000 0000 0000 0000 0000 0000 0000 .....
```

ダンプ・ファイル

ダンプ・ファイルは、エラー発生時に問題を診断するのに役立つと思われる追加情報 (内部制御ブロックなど) がある場合に作成されます。ダンプ・ファイルに書き込まれるデータ項目はそれぞれ、問題の判別に役立つようにタイム・スタンプが関連付けられています。ダンプ・ファイルはバイナリー・フォーマットで、DB2 お客様サポート担当者用です。

ダンプ・ファイルが作成または追加されると、書き込まれたデータの時刻およびタイプを示した項目が db2diag.log に作成されます。これらの db2diag.log の項目は、以下のようになっています。

```
2007-05-18-12.28.11.277956-240 I24861950A192 LEVEL: Severe
PID:1056930 TID:225448 NODE:000 Title: dynamic memory buffer
Dump File:/home/svtdbm5/sqllib/db2dump/1056930.225448.000.dump.bin
```

注: パーティション・データベース環境では、ファイル拡張子でパーティション番号が識別されます。例えば、以下の項目は、ダンプ・ファイルがパーティション 10 で実行されている DB2 プロセスによって作成されたことを示します。

Dump File: /home/db2/sqllib/db2dump/6881492.2.010.dump.bin

トラップ・ファイル

DB2 は、トラップやセグメンテーション違反、例外などにより処理が続行できない時、トラップ・ファイルを生成します。

DB2 によって予約されたすべてのシグナルまたは例外は、トラップ・ファイルに記録されます。トラップ・ファイルには、エラーが発生したときに実行されていた関数シーケンスも含まれています。このシーケンスは、「関数呼び出しスタック」または「スタック・トレース」と呼ばれることもあります。トラップ・ファイルには、シグナルまたは例外がキャッチされた時のプロセスの状態に関する追加情報も含まれています。

ファイルは、DIAGPATH データベース・マネージャ構成パラメーターで指定されているディレクトリーにあります。

すべてのプラットフォームで、トラップ・ファイル名はプロセス ID (PID) で始まり、スレッド ID (TID)、パーティション番号 (単一パーティション・データベースの場合は 000) と続き、「.trap.txt」で終わります。

必ずしもインスタンスが異常終了したとは限らないものの、スタックを参照すると役立つ特定の条件が生じた場合に、コードによって生成される診断トラップもあります。これらのトラップの名前は、10 進形式の PID の後にパーティション番号 (単一パーティション・データベースの場合は 0) を付けた形になります。

例:

- 6881492.2.000.trap.txt は、プロセス ID (PID) が 6881492、スレッド ID (TID) が 2 のトラップ・ファイルです。
- 6881492.2.010.trap.txt は、プロセスとスレッドがパーティション 10 で実行されているトラップ・ファイルです。

db2pd コマンドに `-stack all` または `-dump` オプションを指定すると、オンデマンドでトラップ・ファイルを生成することができます。しかし、一般的にこれは、DB2 サポートが要求した場合にのみ行います。

スタック・トレース・ファイルは、db2pd `-stacks` コマンドまたは db2pd `-dumps` コマンドで生成できます。これらのファイルは、トラップ・ファイルと同じ内容ですが、診断専用として生成されます。その名前は、6881492.2.000.stack.txt のようになります。

トラップ・ファイルのフォーマット (Windows)

トラップ・ファイル (*.TRP) をフォーマットするため、db2xpvt.exe というツールが使用可能です。これは、DB2 データベースのバイナリー・トラップ・ファイルを、人間が理解できる ASCII ファイルにフォーマットします。

このツールは、トラップ・ファイルをフォーマットするのに DB2 シンボル・ファイルを使用します。これらの .PDB ファイルのサブセットが DB2 データベース製品に付属しています。

「DB30882416.TRP」という名前のトラップ・ファイルが DIAGPATH に作成された場合、以下のようにフォーマットできます。

```
db2xprt DB30882416.TRP DB30882416.FMT
```

プラットフォーム固有のエラー・ログ情報

DB2 以外のファイルやユーティリティの中にも、問題の分析に役立つものが多数あります。これらのファイルやユーティリティは、根本原因を判別するうえで、DB2 ファイルに用意される情報と同じほど重要な役割を果たすことがよくあります。他のファイルやユーティリティは、以下の領域に関するログやトレースに含まれる情報にアクセスします。

- オペレーティング・システム
- アプリケーションおよびサード・パーティー・ベンダー
- ハードウェア

稼働環境によっては、上記以外の場所でも役立つ情報が見つかる可能性があります。このため、システムの問題をデバッグするときには、可能性のあるすべての領域をよく調査する必要があります。

オペレーティング・システム

各オペレーティング・システムには、アクティビティや障害を追跡するための独自の診断ファイル・セットがあります。通常、最も一般的な (そして最も役立つ) ファイルは、エラー・レポートまたはイベント・ログです。この情報を集める方法は以下のとおりです。

- AIX®: /usr/bin/errpt -a コマンド
- Solaris: /var/adm/messages* ファイルまたは /usr/bin/dmesg コマンド
- Linux: /var/log/messages* ファイルまたは /bin/dmesg コマンド
- HP-UX: /var/adm/syslog/syslog.log ファイルまたは /usr/bin/dmesg コマンド
- Windows: システム、セキュリティ、およびアプリケーションのイベント・ログ・ファイルと、windir¥drwtsn32.log ファイル (windir は Windows のインストール先ディレクトリ)

各オペレーティング・システムには、これ以外にもトレース・ユーティリティやデバッグ・ユーティリティが提供されています。どのような詳細情報が利用可能かについては、ご使用のオペレーティング・システムの資料やサポート資料を参照してください。

アプリケーションおよびサード・パーティー・ベンダー

各アプリケーションには独自のログ・ファイルと診断ファイルがあります。このようなファイルを DB2 からの情報の補足として利用すれば、問題が存在する可能性のある領域をより正確に把握できます。

ハードウェア

ハードウェア装置は、通常、オペレーティング・システムのエラー・ログの中に情報を記録します。ただし、追加情報が必要になることもあります。そのような場合、ご使用の環境内の特定のハードウェアに関するどのような診断ファイルやユーティリティーが存在するかを識別する必要があります。一例として、不正なページや何らかの破損が DB2 によって報告されることがあります。ほとんどの場合、そのような報告の原因はディスクの問題であり、ハードウェアの診断情報を調査する必要があります。どのような詳細情報が利用可能かについては、ご使用のハードウェアの資料やサポート資料を参照してください。

ハードウェア・ログの情報など、一部の情報には時間の制約があります。エラーが発生したら、関係する情報源からできるだけ早くできるだけたくさんの情報を収集するように最善を尽くす必要があります。

以上をまとめると、問題を完全に理解して評価するには、DB2、アプリケーション、オペレーティング・システム、および基礎となるハードウェアから得られるすべての情報を集める必要があるかもしれません。db2support ツールは必要な DB2 情報とオペレーティング・システム情報のほとんどを自動的に収集しますが、こうして得られる情報以外にも、調査に役立つ情報が見つかる可能性があることに注意してください。

システム・コア・ファイル (Linux および UNIX)

プログラムが異常終了した場合、コア・ファイルがシステムによって作成され、終了処理のメモリー・イメージが保管されます。メモリー・アドレス違反、不正命令、バス・エラー、およびユーザー生成終了シグナルなどのエラーで、コア・ファイルにダンプが保管されます。

コア・ファイルの名前は「core」で、DB2FODC レジストリー変数の値で別に構成されない限り、デフォルトで diagpath に置かれます。システム・コア・ファイルと、DB2 トラップ・ファイルは別であることに注意してください。

システム・コア・ファイル情報へのアクセス (Linux および UNIX)

dbx システム・コマンドは、どの関数がシステム・コア・ファイルの作成の原因になったかを判別するのに役立ちます。これは、データベース・マネージャーがエラーであるか、またはオペレーティング・システムまたはアプリケーション・エラーがこの問題の原因かどうかのいずれかかを識別するのを助ける、簡単なチェックです。

- dbx コマンドがインストールされている。このコマンドは、オペレーティング・システム固有のコマンドです。AIX と Solaris の場合は、dbx を使用します。HP-UX の場合は xdb、Linux の場合は gdb をそれぞれ使用します。
- AIX では、chdev コマンドまたは smitty を使用して、フル・コア・オプションが使用可能かどうかを確認してください。

以下のステップを使用して、コア・ファイル・ダンプが発生する原因になった関数を判別できます。

1. UNIX コマンド・プロンプトから、以下のコマンドを入力します。


```
dbx program_name core_filename
```

program_name は異常終了したプログラムの名前、*core_filename* はコア・ファイル・ダンプを含むファイルの名前です。 *core_filename* パラメーターはオプションです。指定しない場合は、デフォルト名 "core" が使用されます。

2. コア・ファイル内の呼び出しスタックを調べます。これを行う方法についての情報は、UNIX コマンド・プロンプトから `man dbx` を発行することによって入手できます。
3. `dbx` コマンドを終了するには、`dbx` プロンプトで `quit` と入力します。

以下の例は、`dbx` コマンドを使用して、"main" というプログラムのコア・ファイルを読み取る方法を表示しています。

1. コマンド・プロンプトで、次を入力します。

```
dbx main
```

2. 以下のような出力がディスプレイに表示されます。

```
dbx version 3.1 for AIX.  
Type 'help' for help.  
reading symbolic information ...  
[using memory image in core]  
segmentation.violation in freeSegments at line 136  
136      (void) shmdt((void *) pcAddress[i]);
```

3. コア・ダンプの原因の関数名は "freeSegments" です。`dbx` プロンプトに `where` と入力して、障害点をポイントするプログラム・パスを表示します。

```
(dbx) where  
freeSegments(numSegs = 2, iSetId = 0x2ff7f730, pcAddress = 0x2ff7f758, line  
136  
in "main.c"  
main (0x1, 2ff7f7d4), line 96 in "main.c"
```

この例では、`main.c` の行 96 から呼び出された、`freeSegments` の行 136 でエラーが発生しました。

4. `dbx` コマンドを終了するには、`dbx` プロンプトで `quit` と入力します。

イベント・ログへのアクセス (Windows)

Windows イベント・ログの情報もまた、役立つ場合があります。以下の 3 種類のイベント・ログをすべて取得することをお勧めします。とくに、DB2 のクラッシュ、またはシステム・リソースに関連したその他の不可解なエラーが発生した場合には、通常、システム・イベント・ログが最も役立ちます。

- システム
- アプリケーション
- セキュリティー

Windows イベント ビューアを使用してイベント・ログを表示します。ビューアを起動する方法は、ご使用の Windows オペレーティング・システムによって異なります。

例えば、Windows XP でイベント ビューアを開くには、「スタート」->「コントロール パネル」をクリックします。「管理ツール」を選択してから、「イベント ビューア」をダブルクリックします。

イベント・ログのエクスポート (Windows)

Windows イベント ビューアから、2 つの形式 (ログ・ファイル形式と、テキストまたはコンマ区切り形式) でイベント・ログをエクスポートすることができます。

Windows イベント ビューアからイベント・ログをエクスポートします。

- ログ・ファイル形式 (*.evt) では、データを (例えば他のマシン上の) イベント・ビューアに再ロードできます。この形式を使用した場合、ビューアを使って時間の順序を切り替えたり、特定のイベントのフィルター処理や前後への移動を行うことができ便利です。
- テキストまたはコンマ区切り形式 (それぞれ *.txt と *.csv) では、ほとんどのテキスト・エディターでログを開くことができます。また、タイム・スタンプに関連して起こる可能性がある問題を回避できます。イベント・ログを .evt 形式でエクスポートした場合、タイム・スタンプは協定世界時で保存され、ビューア内でそのマシンの現地時間に変換されます。このため、よく注意しないと、タイム・ゾーンの違いのために重要なイベントを見逃す可能性があります。さらに、テキスト・ファイルは検索が容易です。

ワトソン博士のログ・ファイルへのアクセス (Windows)

ワトソン博士のログ drwtsn32.log は、システムで発生したすべての例外を履歴として記録します。ワトソン博士のログよりも DB2 トラップ・ファイルの方が役立ちますが、ワトソン博士のログはシステム全体の安定性を評価したり、DB2 トラップの履歴を保存するために使用できます。

ワトソン博士のログ・ファイルは次のロケーションで見つけることができます。 デフォルト・パスは <install_drive>:\Documents and Settings ¥All Users¥Documents¥DrWatson です。

DB2 データベースの診断と OS の診断の結合

メモリー、スワップ・ファイル、CPU、ディスク・ストレージ、その他のリソースに関連した問題を診断するには、所定のオペレーティング・システムがこれらのリソースをどのように管理するかについて十分に理解しておく必要があります。リソース関連の問題を定義するには、最低でも、そのリソースがどれだけ存在するか、ユーザーごとにどのようなリソース制限があるかを知っておく必要があります。(関係のある制限は、DB2 インスタンス所有者のユーザー ID に対するものであるのが一般的です。)

以下は、入手する必要のある重要な構成情報の一部です。

- オペレーティング・システムのパッチ・レベル、インストール済みソフトウェア、およびアップグレード履歴
- CPU の数
- RAM の容量
- スワップおよびファイル・キャッシュの設定
- ユーザー・データとファイル・リソースの制限、およびユーザーごとの処理限界

- IPC リソースの制限 (メッセージ・キュー、共用メモリー・セグメント、セマフォ)
- ディスク・ストレージのタイプ
- マシンがその他の用途に使用されているかどうか (DB2 がリソースを求めて競合するかどうか)
- 認証が行われる場所

ほとんどのプラットフォームには、リソースの情報を取得するための簡単なコマンドがあります。しかし、こうした情報を手動で取得する必要があるのはまれです。db2support ユーティリティーが、このデータ、およびその他多くのデータを収集するからです。db2support (オプション -s および -m を指定した場合) によって生成される detailed_system_info.html ファイルには、この情報を収集するために使用される多数のオペレーティング・システム・コマンドの構文が記載されています。

以下の演習の目的は、各種の DB2 診断ファイルからシステム構成とユーザー環境の情報を発見するのを助けることです。最初の演習では、db2support ユーティリティーの実行に関するステップを示します。次の演習では、トラップ・ファイルを取り上げます。トラップ・ファイルには、DB2 によって生成されるさらに多くのデータが含まれています。これらのデータは、ユーザー環境やリソースの制限を把握するのに役立ちます。

演習 1: db2support コマンドの実行

1. db2start コマンドを使って DB2 インスタンスを開始します。
2. db2support からの出力を格納するためのディレクトリーを作成します (既に SAMPLE データベースがあるものと仮定しています)。
3. そのディレクトリーに移動して、以下を実行します。
db2support <directory> -d sample -s -m
4. コンソール出力を確認します。特に、収集されている情報の種類に注目します。

以下のような出力が表示されるはずです (Windows で実行した場合)。

```

...
Collecting "System files"
    "db2cache.prf"
    "db2cos9402136.0"
    "db2cos9402840.0"
    "db2dbamr.prf"
    "db2diag.bak"
    "db2eventlog.000"
    "db2misc.prf"
    "db2nodes.cfg"
    "db2profile.bat"
    "db2system"
    "db2tools.prf"
    "HealthRulesV82.reg"
    "db2dasdiag.log"
...
Collecting "Detailed operating system and hardware information"
Collecting "System resource info (disk, CPU, memory)"
Collecting "Operating system and level"
Collecting "JDK Level"
Collecting "DB2 Release Info"
Collecting "DB2 install path info"
Collecting "Registry info"

```

```
...
Creating final output archive
  "db2support.html"
  "db2_sql1lib_directory.txt"
  "detailed_system_info.html"
  "db2supp_system.zip"
  "dbm_detailed.supp_cfg"
  "db2diag.log"
db2support is now complete.
An archive file has been produced: "db2support.zip"
```

5. 次に、Web ブラウザーを使って detailed_system_info.html ファイルを表示します。ご使用の各システムで、以下の情報を確認してください。

- CPU の数
- オペレーティング・システムのレベル
- ユーザー環境
- ユーザー・リソースの制限 (UNIX ulimit コマンド)

演習 2: DB2 トラップ・ファイルから環境情報を見付ける

1. DB2 インスタンスが開始されていることを確認し、その後以下を発行します。

```
db2pd -stack all
```

呼び出しスタックは、診断ディレクトリー (データベース・マネージャー構成パラメーター DIAGPATH で定義されている) 内のファイルに置かれます。

2. いずれかのトラップ・ファイルで、以下を見付けます。

- DB2 のコード・レベル
- Data seg top (要求された専用アドレス・スペースの最大サイズ)
- Cur data size (専用アドレス・スペースの上限)
- Cur core size (コア・ファイル・サイズの上限)
- Signal Handlers (この情報は、必ずしもすべてのトラップ・ファイルに表示されるとは限らない)
- Environment variables (この情報は、必ずしもすべてのトラップ・ファイルに表示されるとは限らない)
- map output (ロードされたライブラリー)

Windows からのトラップ・ファイルの例 (短縮版)

```
...
<DB2TrapFile version="1.0">
<Trap>
<Header>
DB2 build information: DB2 v9.1.0.190 s060121 SQL09010
timestamp: 2006-02-17-14.03.43.846000
uname: S:Windows
comment:
process id: 940
thread id: 3592
</Header>
<SystemInformation>
Number of Processors: 1
Processor Type: x86 Family 15 Model 2 Stepping 4
OS Version: Microsoft Windows XP, Service Pack 2 (5.1)
Current Build: 2600
</SystemInformation>
<MemoryInformation>
```

```

<Usage>
Physical Memory:    1023 total,    568 free.
Virtual Memory :    2047 total,    1882 free.
Paging File   :    2461 total,    2011 free.
Ext. Virtual   :         0 free.
</Usage>
</MemoryInformation>
<EnvironmentVariables>
<![CDATA[
[e] DB2PATH=C:\Program Files\IBM\SQLLIB
[g] DB2_EXTSECURITY=YES
[g] DB2SYSTEM=MYSRVR
[g] DB2PATH=C:\Program Files\IBM\SQLLIB
[g] DB2INSTDEF=DB2
[g] DB2ADMINSERVER=DB2DAS00
]]></EnvironmentVariables>

```

DB2 とシステム・イベントまたはエラーとの相関

システム・メッセージとエラー・ログは見過ごされることが多くあります。問題の定義と調査の最初の段階で時間を取って 1 つの簡単な作業さえ行っておけば、問題解決までに要する時間を、時間単位、日単位、さらには週単位で減らすことができます。その作業とは、さまざまなログの項目同士を比較して、時間と、参照しているリソースの両面で関連がありそうな項目がないか注意することです。

必ずしも常に問題診断に関係があるわけではありませんが、システム・ログで最良の手掛かりが簡単に見つかることは少なくありません。報告されたシステム問題と DB2 エラーとの相関を見出すことができれば、多くの場合、DB2 の症状の直接的な原因をすでに識別できていることになります。分かりやすい例としては、ディスク・エラー、ネットワーク・エラー、およびハードウェア・エラーがあります。分かりにくいのは、別のマシンで報告される問題です。例えば、ドメイン・コントローラーが接続時間や認証に影響を与える可能性がある場合などです。

特に、新しいシステムで問題が報告される場合には、システム・ログを調べて安定度を分析できます。共通アプリケーションで断続的にトラップが生じているなら、それは基盤となるハードウェアに障害があることを示すしるしかもしれません。

以下に、システム・ログが提供する他の情報をいくつか挙げます。

- システムのリポートなどの重大なイベント
- システム上で生じた DB2 のトラップ (および障害が起こった他のソフトウェアからのエラー、トラップ、または例外) の時間的順序
- カーネルのパニック、out-of-filesystem-space、および out-of-swap-space などのエラー (これが原因でシステムが新しいプロセスを作成またはフォークできないことがあります)

システム・ログを調べることで、db2diag.log 内のクラッシュ項目を、懸念される要因から除外できることがあります。DB2 管理通知または DB2 診断ログの中で、先にエラーがないのにクラッシュ・リカバリーがある場合、DB2 クラッシュ・リカバリーはシステムのシャットダウンの結果生じたものと考えられます。

情報を相関させるというこの原則は、あらゆるソースからのログ、および識別可能なあらゆるユーザーの症状にも適用できます。例えば、別のアプリケーションのログから相関関係にある項目を識別して文書化することは、たとえそれを十分に解釈できないとしても、大いに役立つ場合があります。

この情報を加えることで、サーバー、および問題発生時に生じていたさまざまなイベントすべてについての十分な理解が得られます。

第 2 章 DB2 のトラブルシューティング

一般にトラブルシューティングには、問題を切り分け、識別してから、解決方法を探ることが求められます。このセクションでは、DB2 製品の特定のフィーチャーに関連したトラブルシューティング情報を提供します。

共通問題が識別されるにつれて、検出された事項がチェックリストの形でこのセクションに追加されていきます。チェックリストでは解決方法に到達できない場合は、追加の診断データを収集してご自分で分析し、そのデータを IBM ソフトウェア・サポートに分析用に提出することができます。

以下の質問により、適切なトラブルシューティング・タスクに誘導されます。

1. 既知のすべてのフィックスパックを適用しましたか? まだしていないなら、「DB2 サーバー機能 概説およびインストール」の『フィックスパックの適用』を検討してください。
2. 問題が発生するのは以下の場合ですか?
 - DB2 データベース・サーバーまたはクライアントのインストール中。この場合は、本書の別の場所にある『インストール問題についてのデータの収集』のトピックを参照してください。
 - インスタンスまたは DB2 管理サーバー (DAS) の作成、ドロップ、更新、またはマイグレーション中。この場合は、本書の別の場所にある『DAS およびインスタンス管理の問題についてのデータの収集』のトピックを参照してください。
 - EXPORT、IMPORT、LOAD、または db2move コマンドを使用してデータを移動中。この場合は、本書の別の場所にある『データ移動問題についてのデータの収集』のトピックを見てください。

問題がこれらのカテゴリーのいずれにも属さなくても、IBM ソフトウェア・サポートに連絡を取る場合には基本的な診断データが必要になることがあります。本書の別の場所にある『DB2 についてのデータの収集』のトピックを参照してください。

現行リリースのトラブルシューティング・ガイダンス

新規および変更されたデータベース機能およびフィーチャーで作業しているときに遭遇する可能性のある問題に関するガイダンスが提供されています。データベース操作の最近の変更の結果として、新規または変更されたフィーチャーの最適な利用法が忘れられたり、見過ごされたりすることがあります。以下のトラブルシューティング・トピックは、可能性のある問題の領域を示唆し、本リリースで導入された新規および変更済みデータベース機能およびフィーチャーを十分に利用するための方向性を提案します。

高可用性のトラブルシューティング

AIX 6.1 では Tivoli System Automation for Multiplatforms (SA MP) Base Component は DB2 バージョン 9.5 GA によってインストールされない

DB2 バージョン 9.5 GA 高可用性機能に含まれている IBM Tivoli® SA MP Base Component は、AIX 6.1 オペレーティング・システムをサポートしていません。AIX 6.1 用の適切なバージョンの SA MP Base Component を取得するには、DB2 バージョン 9.5 フィックスパック 1 以降をインストールしてください。

症状

DB2 バージョン 9.5 GA データベース製品を AIX 6.1 にインストールすると、インストーラーは AIX 6.1 が使用されていることを検出し、SA MP Base Component はインストールされません。

原因

DB2 バージョン 9.5 GA にバンドルされている SA MP Base Component は AIX 6.1 をサポートしていません。

問題の解決

DB2 バージョン 9.5 フィックスパック 1 以降を AIX 6.1 にインストールすると、SA MP Base Component は正常にインストールされます。

インストールのトラブルシューティング

システム WPAR (AIX) 上のデフォルト・パスに DB2 データベース製品をインストールする際のエラー

AIX 6.1 のシステム・ワークロード・パーティション (WPAR) 上のデフォルト・インストール・パス (/opt/IBM/db2/V9.5) に DB2 データベース製品をインストールすると、種々のエラーが発生する可能性があります。こうした問題を回避するには、WPAR に対してのみアクセス可能なファイル・システム上に DB2 データベース製品をインストールします。

症状

DB2 データベース製品をシステム WPAR の /usr または /opt ディレクトリーにインストールすると、ディレクトリーの構成方法によっては様々なエラーが発生する可能性があります。システム WPAR を、/usr および /opt ディレクトリーをグローバル環境と共有するように構成することもできれば (この場合、/usr および /opt ディレクトリーは WPAR から読み取り可能ですが、書き込みアクセスは不可になります)、/usr および /opt ディレクトリーのローカル・コピーを持つように構成することもできます。

最初のシナリオの場合に DB2 データベース製品をグローバル環境のデフォルト・パスにインストールすると、システム WPAR にこのインストール環境が表示されます。このため DB2 は WPAR にインストールされたかのように思われますが、DB2 インスタンスを作成しようとする試みは次のエラーを生じさせることになりま

す。DBI1288E プログラム db2icrt の実行に失敗しました。このプログラムが失敗したのは、ディレクトリーまたはファイル /opt/IBM/db2/V9.5/profiles.reg,/opt/IBM/db2/V9.5/default.env に対する書き込み許可がないためです。

2 番目のシナリオで DB2 データベース製品をグローバル環境のデフォルト・パスにインストールしてから、WPAR で /usr および /opt ディレクトリーのローカル・コピーを作成すると、DB2 データベース製品のインストール環境もコピーされます。これによって、システム管理者がこのデータベース・システムを用いようとするとき、予期しない問題が生じる可能性があります。DB2 データベース製品は別のシステムを対象としていたので、不正確な情報がコピーされる場合があります。例えば、もともとはグローバル環境で作成された DB2 インスタンスが WPAR に存在するかのように見えることがあります。これは、実際にどのインスタンスがシステムにインストールされているかをシステム管理者が判別する上で混乱の原因となります。

原因

こうした問題の原因は、DB2 データベース製品をシステム WPAR の /usr または /opt ディレクトリー上にインストールすることにあります。

問題の解決

DB2 データベース製品を、グローバル環境のデフォルト・パスにインストールしないでください。

WPAR に対してのみアクセス可能なファイル・システムをマウントし、そのファイル・システムに DB2 データベース製品をインストールします。

パーティション・データベース環境のトラブルシューティング

パーティション・データベースを持つ環境でトラブルシューティングするには、独特の考慮事項があります。パーティション・データベース環境以外で稼働しているときに有効な特定のエントリーがホスト・ファイルにある場合、高速コミュニケーション・マネージャー (FCM) で問題が発生します。

127.0.0.2 に関連した FCM 問題 (Linux および UNIX)

パーティション・データベース環境では、/etc/hosts ファイルに 127.0.0.2 の項目があると、高速コミュニケーション・マネージャー (FCM) が問題を検出する可能性があります。

症状

環境によっては、さまざまなエラー・メッセージが発生します。例えば、データベースを作成するときに、「SQL1229N システム・エラーのため、現在のトランザクションがロールバックされました。SQLSTATE=40504」というエラーが発生する可能性があります。

原因

この問題の原因は、/etc/hosts ファイルに IP アドレス 127.0.0.2 の項目が存在することです (127.0.0.2 は、マシンの完全修飾ホスト名にマップされています)。例:

127.0.0.2 ServerA.ibm.com ServerA

「ServerA.ibm.com」は、完全修飾ホスト名です。

環境

問題は、DB2 Enterprise Server Edition と DB2 Database Partitioning Featureに限定されます。

問題の解決

/etc/hosts ファイルから項目を除去するか、その項目をコメントに変換します。例:

```
# 127.0.0.2 ServerA.ibm.com ServerA
```

暗号化ファイル・システムにおけるデータベース・パーティションの作成 (AIX)

AIX 6.1 では、JFS2 ファイル・システムまたはファイル・セットの暗号化機能がサポートされています。この機能では、DB2 データベース製品のパーティション化されたデータベース環境ではサポートされていません。AIX 上で、パーティション化されたデータベース環境を EFS (暗号化ファイル・システム) を使用して作成しようとすると、SQL10004C エラーが生じます。

症状

複数パーティション・データベース環境にある暗号化ファイル・システムでデータベースを作成しようとすると、次のエラーを受け取ります。SQL10004C データベース・ディレクトリーのアクセス中に入出力エラーが発生しました。

```
SQLSTATE=58031
```

原因

この時点で、EFS (暗号化ファイル・システム) を使用して AIX 上にパーティション化されたデータベース環境を作成することはできません。パーティション化されたデータベース・パーティションは rsh または ssh を使用するの、EFS 内の鍵ストアが失われると、暗号化ファイル・システムに格納されているデータベース・ファイルにデータベース・パーティションはアクセスできなくなります。

問題の診断

DB2 診断ログ・ファイル (db2diag.log) には以下のエラー・メッセージが組み込まれます。OSERR : ENOATTR (112) "属性がありません (No attribute found)"

問題の解決

パーティション化されたデータベース環境でデータベースを正常に作成するには、すべてのマシンで使用できるファイル・システムがあること、およびそれが暗号化ファイル・システムではないことが必要です。

最適化ガイドラインおよびプロファイルのトラブルシューティング

最適化ガイドライン (最適化プロファイルにより渡される) の診断サポートは、EXPLAIN 表を介して行われます。

オブティマイザーが最適化ガイドラインを適用しない場合には、SQL0437W 警告、理由コード 13 を受け取ります。最適化ガイドラインが適用されなかった理由を説明する診断情報が EXPLAIN 表に追加されます。オブティマイザーの診断出力を受け取るための EXPLAIN 表には次の 2 つがあります。

- EXPLAIN_DIAGNOSTIC - この表の各項目は、特定のステートメントの最適化に関する診断メッセージを表します。各診断メッセージは、数字コードを使用して示されます。
- EXPLAIN_DIAGNOSTIC_DATA - この表の各項目は、EXPLAIN_DIAGNOSTIC 表内の特定の診断メッセージと関連した診断データです。

診断用 Explain 表を作成するための DDL を以下の 36 ページの図 1 に示します。

以下のステップは、最適化ガイドラインの使用時に発生する問題をトラブルシューティングするときに役立ちます。

1. 「データベース・パフォーマンスのチューニング」の『最適化ガイドラインが使用されていることの確認』を参照します。
2. 組み込みの 『EXPLAIN_GET_MSGS 表関数』（「管理ルーチンおよびビュー」）を使用して完全なエラー・メッセージを調べます。

これらの手順を実行しても問題の原因を突き止めることができない場合には、診断データの収集を開始して、IBM ソフトウェア・サポートに連絡を取ることを考慮してください。

```

CREATE TABLE EXPLAIN_DIAGNOSTIC
( EXPLAIN_REQUESTER VARCHAR(128) NOT NULL,
  EXPLAIN_TIME       TIMESTAMP   NOT NULL,
  SOURCE_NAME        VARCHAR(128) NOT NULL,
  SOURCE_SCHEMA      VARCHAR(128) NOT NULL,
  SOURCE_VERSION     VARCHAR(64)  NOT NULL,
  EXPLAIN_LEVEL      CHAR(1)     NOT NULL,
  STMTNO             INTEGER      NOT NULL,
  SECTNO             INTEGER      NOT NULL,
  DIAGNOSTIC_ID     INTEGER      NOT NULL,
  CODE               INTEGER      NOT NULL,
  PRIMARY KEY (EXPLAIN_REQUESTER,
               EXPLAIN_TIME,
               SOURCE_NAME,
               SOURCE_SCHEMA,
               SOURCE_VERSION,
               EXPLAIN_LEVEL,
               STMTNO,
               SECTNO,
               DIAGNOSTIC_ID),
  FOREIGN KEY (EXPLAIN_REQUESTER,
               EXPLAIN_TIME,
               SOURCE_NAME,
               SOURCE_SCHEMA,
               SOURCE_VERSION,
               EXPLAIN_LEVEL,
               STMTNO,
               SECTNO)
  REFERENCES EXPLAIN_STATEMENT ON DELETE CASCADE);

CREATE TABLE EXPLAIN_DIAGNOSTIC_DATA
( EXPLAIN_REQUESTER VARCHAR(128) NOT NULL,
  EXPLAIN_TIME       TIMESTAMP   NOT NULL,
  SOURCE_NAME        VARCHAR(128) NOT NULL,
  SOURCE_SCHEMA      VARCHAR(128) NOT NULL,
  SOURCE_VERSION     VARCHAR(64)  NOT NULL,
  EXPLAIN_LEVEL      CHAR(1)     NOT NULL,
  STMTNO             INTEGER      NOT NULL,
  SECTNO             INTEGER      NOT NULL,
  DIAGNOSTIC_ID     INTEGER      NOT NULL,
  ORDINAL            INTEGER      NOT NULL,
  TOKEN              VARCHAR(1000),
  TOKEN_LONG         BLOB(3M) NOT LOGGED,
  FOREIGN KEY (EXPLAIN_REQUESTER,
               EXPLAIN_TIME,
               SOURCE_NAME,
               SOURCE_SCHEMA,
               SOURCE_VERSION,
               EXPLAIN_LEVEL,
               STMTNO,
               SECTNO,
               DIAGNOSTIC_ID)
  REFERENCES EXPLAIN_DIAGNOSTIC ON DELETE CASCADE);

```

注: EXPLAIN_REQUESTER、EXPLAIN_TIME、SOURCE_NAME、SOURCE_SCHEMA、SOURCE_VERSION、EXPLAIN_LEVEL、STMTNO、SECTNO の各列は、EXPLAIN_STATEMENT 表の外部キーと、EXPLAIN_DIAGNOSTIC と EXPLAIN_DIAGNOSTIC_DATA の親子関係を設定するために、両方の表に含まれています。

図 1. 診断用 Explain 表を作成するための DDL

この DDL は、sqllib ディレクトリーの misc サブディレクトリーに置かれている EXPLAIN.DDL ファイルに組み込まれています。

ストレージ・キー・サポートのトラブルシューティング

ストレージ保護キー (スレッド・レベルのハードウェア・キー) は、無効なアクセスの試行からメモリーを保護することにより、DB2 エンジンの回復力を高めるために使用されます。このフィーチャーを使用可能にする際にエラーが発生した場合、またはこのフィーチャーが使用可能になっているときにトラップが発生した場合は、以下のステップを実行してください。

レジストリー変数エラーの診断

レジストリー変数 『DB2_MEMORY_PROTECT』 および

『DB2_THREAD_SUSPENSION』(「データ・サーバー、データベース、およびデータベース・オブジェクトのガイド」) を設定したときに、「値が無効です」(DBI1301E) エラーが返されました。このエラーは、以下のいずれかの理由によって生じます。

- レジストリー変数に指定された値が無効である。該当する変数、DB2_MEMORY_PROTECT または DB2_THREAD_SUSPENSION のレジストリー変数の使用法を参照してください。
- DB2_MEMORY_PROTECT 変数の設定時にエラーが発生した場合は、ハードウェアおよびオペレーティング・システムがストレージ保護キーをサポートしておらず、フィーチャーを使用可能にできない可能性があります。ストレージ保護キーは POWER6 プロセッサで使用でき、AIX 5L バージョン 5.3 (5300-06 Technology Level を適用) オペレーティング・システムの時点でサポートされています。
- DB2_THREAD_SUSPENSION 変数を ON に設定しているときにエラーが発生した場合は、DB2_THREAD_SUSPENSION 変数を設定するより前に DB2_MEMORY_PROTECT 変数が設定されていません。db2set コマンドを使用して DB2_MEMORY_PROTECT 変数を YES に設定します。次に、db2set コマンドを使用して DB2_THREAD_SUSPENSION 変数を ON に設定します。最後に、DB2 インスタンスを停止および再始動して、レジストリー変数の変更をアクティブにします。

トラップの診断

DB2 インスタンスは、発生したトラップに対する First Occurrence Data Capture (FODC) パッケージを準備します。データベースの回復力を高めるように DB2 インスタンスが構成されており、DB2_MEMORY_PROTECT 変数が YES に、DB2_THREAD_SUSPENSION 変数が ON に設定されている場合、DB2 インスタンスはトラップが持続可能なものかどうかも判別しています。「持続可能な」という用語は、トラップされた DB2 エンジン・スレッドが中断または終了され、DB2 インスタンスは実行を続けることを意味します。以下のステップを実行します。

1. テキスト・エディターを使用して管理通知ログ・ファイルを表示します。トラップが持続可能なもので、DB2 インスタンスがまだ実行中の場合は、エラー・メッセージ 『ADM14010C』(「メッセージ・リファレンス 第 1 巻」) が表示されます。そうでない場合は、エラー・メッセージ 『ADM14011C』(「メッセージ・リファレンス 第 1 巻」) が表示され、DB2 インスタンスがシャットダウンします。
2. 上記の該当するエラー・メッセージで指定されている FODC 情報のディレクトリー名に注意します。

3. トラップが持続的になった場合、なるべく早いタイミングで DB2 インスタンスを停止します。トラップが持続中は DB2 エンジン・スレッドが中断されるため、**db2stop** および STOP DATABASE MANAGER コマンドを使用して DB2 インスタンスを停止しようとする、と、ハングします。その代わりに、**db2_kill** コマンドを使用して DB2 インスタンスを停止し、中断した DB2 エンジン・スレッドを除去する必要があります。
4. **db2start** または START DATABASE MANAGER コマンドを使用して、DB2 インスタンスを再始動します。
5. IBM お客様サポートに連絡し、FODC 診断情報を参照してトラップの原因を解決します。

データ・コンプレッション・ディクショナリーが自動的に作成されない

大容量の表なのに、データ・コンプレッション・ディクショナリーが作成されません。データ・コンプレッション・ディクショナリーが期待どおりに作成されなかった理由を調べます。

以下のような状況が起こり得ます。

- COMPRESS 属性を YES に設定した表があります。
- その表は一定の期間存在しており、データが追加されたり除去されたりしました。
- その表のサイズは、しきい値のサイズに近くなっています (1 MB から 2 MB 程度)。データ・コンプレッション・ディクショナリーが自動的に作成されるようにしたいと考えます。
- 表のサイズがしきい値サイズよりも大きくなるように、表データ取り込み操作 (INSERT、LOAD INSERT、REDISTRIBUTE など) を実行します。
- データ・コンプレッション・ディクショナリーの自動作成が実行されません。データ・コンプレッション・ディクショナリーは作成されず、表に配置されませんでした。この時点で表に追加されたデータが圧縮されると期待していましたが、データは依然として圧縮解除されたままです。

なぜ表データは圧縮されなかったのでしょうか。

表サイズは、データ・コンプレッション・ディクショナリーが自動的に作成されるしきい値サイズよりも大きかったものの、チェックされた別の条件があります。つまり、ディクショナリーが作成されるためには、表の中に十分なデータが存在していなければならない、という条件です。表データに対する過去のアクティビティーには、データの削除や除去が含まれていた可能性があります。つまり、表の中にデータのない大きなセクションが存在していた可能性がある、ということです。このような場合に、表サイズのしきい値に達した (または、しきい値を超えた) 大きな表があっても、ディクショナリーが作成されるほどの十分なデータが表の中に存在しないということがあります。

表に対して多数のアクティビティーを実行している場合は、定期的に表を再編成する必要があります。そうしなければ、表のサイズは大きくても、データがそれほど多く取り込まれていない、という状況が生じる可能性があります。表を再編成すれば、フラグメント化したデータを除去して、表のデータをコンパクトにすることが

できます。再編成後は、表が小さくなり、データの取り込み具合が高密度になります。再編成した表は、表のデータ量をより正確に反映することになり、データ・コンプレッション・ディクショナリーが自動的に作成されるしきい値サイズよりも小さくなる可能性があります。

表の再編成が必要かどうかを確認するには、REORGCHK コマンドを使用します。

グローバル変数問題のトラブルシューティング

一般に、グローバル変数に関連したアプリケーションのトラブルシューティングは、その問題を検出したユーザーがグローバル変数の READ 許可を持っている限り、問題にはなりません。READ 許可さえあれば、VALUES(Global Variable Name) ステートメントを実行して、グローバル変数の値を確認できるからです。場合によっては、アプリケーションを実行しているユーザーがグローバル変数の READ アクセス権を持っていないこともあります。

シナリオ 1 では、グローバル変数を参照するときに起こり得る問題と、その簡単な解決策を取り上げます。シナリオ 2 では、よくある状況として、グローバル変数の READ 許可を適切なユーザーに与えなければならない場合を取り上げます。

シナリオ 1

グローバル変数の参照は、正しく修飾する必要があります。名前が同じでも、スキーマの異なる変数が存在する場合に、PATH レジスター値の前のほうで正しくないスキーマが検出される可能性があります。1 つの解決策は、グローバル変数の参照を完全に修飾することです。

シナリオ 2

アプリケーション開発者 (developerUser) が、いくつかのグローバル変数の値に基づいて、プロシージャ、ビュー、トリガーなどの複雑なセットを作成します。それらのグローバル変数は、その開発者だけが読み取りアクセス権を持っています。その状況で、アプリケーションのエンド・ユーザー (finalUser) がログインして、developerUser によって作成された環境を使用して SQL を実行し始めます。finalUser は、developerUser に対して、参照できるはずのデータを参照できないと苦情を言います。この問題のトラブルシューティングとして、developerUser は、自分の許可 ID を finalUser の許可 ID に変更してから、finalUser としてログインし、finalUser と同じ SQL を実行してみます。developerUser は、finalUser の言い分が正しいことを確認できました。確かに問題があります。

developerUser は、自分が参照できるグローバル変数値と同じグローバル変数を finalUser が参照できるかどうかを確認しなければなりません。developerUser は、SET SESSION USER を実行して、finalUser が参照できるグローバル変数値を確認します。この問題を確認して解決するための推奨方法は、以下のとおりです。

developerUser は、finalUser として SET SESSION USER を使用する許可をセキュリティ管理者 (secadmUser) に申請します。developerUser は、本人としてログインしてから、SET SESSION AUTHORIZATION ステートメントを使用して、SESSION_USER 特殊レジスターに finalUser の許可を設定します。問題の SQL を実行した後に、別の SET SESSION AUTHORIZATION ステートメントを使用し

て、developerUser にスイッチバックします。developerUser は、VALUES ステートメントを実行して、グローバル変数の実際の値を確認します。

以下は、developerUser がデータベースで実行したアクションを示すサンプル SQL です。

```
#####
# developerUser connects to database and creates needed objects
#####

db2 "connect to sample user developerUser using xxxxxxxx"

db2 "create table security.users ¥
(userid varchar(10) not null primary key, ¥
firstname varchar(10), ¥
lastname varchar(10), ¥
authlevel int)"

db2 "insert into security.users values ('ZUBIRI', 'Adriana', 'Zubiri', 1)"
db2 "insert into security.users values ('SMITH', 'Mary', 'Smith', 2)"
db2 "insert into security.users values ('NEWTON', 'John', 'Newton', 3)"

db2 "create variable security.gv_user varchar(10) default (SESSION_USER)"
db2 "create variable security.authorization int default 0"

# Create a procedure that depends on a global variable
db2 "CREATE PROCEDURE SECURITY.GET_AUTHORIZATION() ¥
SPECIFIC GET_AUTHORIZATION ¥
RESULT SETS 1 ¥
LANGUAGE SQL ¥
    SELECT authlevel INTO security.authorization ¥
    FROM security.users ¥
    WHERE userid = security.gv_user"

db2 "grant all on variable security.authorization to public"
db2 "grant execute on procedure security.get_authorization to public"
db2 "terminate"

#####
# secadmUser grants setsessionuser
#####
db2 "connect to sample user secadmUser using xxxxxxxx"
db2 "grant setsessionuser on user finalUser to user developerUser"
db2 "terminate"

#####
# developerUser will debug the problem now
#####

echo "-----"
echo " Connect as developerUser "
echo "-----"
db2 "connect to sample user developerUser using xxxxxxxx"

echo "-----"
echo " SET SESSION AUTHORIZATION = finalUser "
echo "-----"
db2 "set session authorization = finalUser"

echo "--- TRY to get the value of gv_user as finalUser (we should not be able to)"
db2 "values(security.gv_user)"

echo "--- Now call the procedure---"
db2 "call security.get_authorization()"

echo "--- if it works it should return 3 ---"
```

```

db2 "values(security.authorization)"

echo "-----"
echo " SET SESSION AUTHORIZATION = developerUser "
echo "-----"

db2 "set session authorization = developerUser"

echo "--- See what the variable looks like ----"
db2 "values(security.gv_user)"

db2 "terminate"

```

ワークロード管理のトラブルシューティング

ワークロードとは、データベース内の 1 つ以上の作業単位 (UOW) をグループ化するためのユーザー定義基準で構成されるデータベース・オブジェクトです。ワークロード・オカレンスには、ワークロードに関連したデータベース接続内の 1 つ以上の UOW が含まれています。別のデータベース接続にある別の UOW セットと同じワークロードが関連付けられていると、その UOW セットは別のワークロード・オカレンスと見なされます。

各ワークロードの複数のワークロード・オカレンスをシステム上で同時に実行することも可能です。

ワークロードまたはワークロード・オカレンスで問題や障害が発生した場合は、それぞれに関して特定の種類の情報を確認することができます。例えば、以下のような種類の情報を確認します。

- ワークロード・オカレンスのリスト。
WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURENCES 表関数を使用します。この関数の出力には、システム全体でのアプリケーションの固有の ID や固有の UOW ID が含まれています。
- ワークロードとワークロード・オカレンスの ID。
WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES 表関数を使用します。この関数の出力には、固有の UOW ID と UOW 内の固有のアクティビティ ID が含まれています。
- ワークロード・オカレンスの下で現在実行されているすべてのアクティビティ (および要求) のリスト。
WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES 表関数を使用します。この関数の出力には、固有の UOW ID と UOW 内の固有のアクティビティ ID が含まれています。
- その時点でメモリーにあるワークロード情報の内容。
db2pd -workloads コマンドを使用します。このコマンドの出力には、ワークロード・リスト、ワークロード権限情報、それぞれのワークロードの並行ワークロード・オカレンスの数のカウンターが含まれています。

さらに、ワークロード管理に関する問題のトラブルシューティング中に IBM と連絡をとる必要がある場合は、その状態に関連した db2agent についての情報も収集する必要があります。必要な情報を収集するには、db2pd -stack を使用します。IBM に連絡すると、DB2 トレース (db2trc) を実行するように依頼される可能性があります。

スクリプトのトラブルシューティング

データベース・エンジンで実行されているプロセスに基づく内部ツールまたはスクリプトが存在することもあります。すべてのエージェント、プリフェッチャー、ページ・クリーナーが、1つのマルチスレッド・プロセスに含まれている各スレッドと見なされるようになると、それらのツールやスクリプトが機能しなくなる可能性があります。

スレッド化プロセスに合わせて、内部ツールやスクリプトを変更する必要があります。例えば、ps コマンドを呼び出してプロセス名をリストしてから、特定のエージェント・プロセスに対してタスクを実行するスクリプトがあるとします。その場合は、スクリプトを作成し直す必要があります。

問題判別データベース・コマンド db2pd には、すべてのエージェント名とそれぞれのスレッド ID をリストするための新規オプション `-edu` (「engine dispatchable unit」、つまりエンジン・ディスパッチ可能単位の略) があります。db2pd `-stack` コマンドは引き続き、スレッド化エンジンを操作して、個々の EDU スタック、または現在のノードのすべての EDU スタックをダンプできます。

データ不整合のトラブルシューティング

データベース内のどこにデータ不整合があるかを診断することは非常に重要です。データ不整合を特定する 1つの方法は、INSPECT コマンドの出力を使用して、どこに問題があるかを確認することです。不整合が見つければ、その問題をどのように処理するかを決めます。

データ整合性の問題が確認された場合は、2つのオプションがあります。

- DB2 サービスに連絡して、データ不整合のリカバリーの支援を要請できます。
- データ整合性の問題があるデータベース・オブジェクトをいったんドロップしてから再作成できます。

データ不整合の証拠があるデータベース、表スペース、表をチェックするために、INSPECT コマンドの `INSPECT CHECK` パリエーションを使用します。INSPECT CHECK コマンドの結果が生成されたら、db2inspf コマンドを使用して、検査結果をフォーマットする必要があります。

INSPECT コマンドが完了しない場合は、DB2 サービスに連絡してください。

索引とデータの不整合のトラブルシューティング

表の正しいデータに素早くアクセスするには、索引が正確である必要があります。そうでない場合は、データベースが破損しています。

索引とデータの不整合に関するオンライン・チェックを実行するには、オブジェクト相互チェック文節の `INDEXDATA` オプションを使用して INSPECT コマンドを使用します。索引データ・チェックは、INSPECT コマンドの使用時にデフォルトでは実行されないため、明示的に要求する必要があります。

INSPECT が `INDEXDATA` 検査を実行しているときに、索引データ不整合のためにエラーが検出されると、エラー・メッセージ `SQL1141N` が返されます。このエラー・メッセージが返されると、同時にデータ診断情報が収集され、db2diag.log にダ

ンプされます。緊急メッセージも管理通知ログに記録されます。db2diag.log ファイルの内容のフィルター処理やフォーマットには、db2diag.log 分析ツールを使用します。

ロックに関する影響

INSPECT コマンドの INDEXDATA オプションを使用して索引とデータの不整合をチェックしている間、検査対象の表は IS モードでのみロックされます。

INDEXDATA オプションを指定した場合のデフォルトでは、明示的に指定されたレベル文節オプションの値だけが使用されます。明示的に指定されていないレベル文節オプションについては、デフォルト・レベル (INDEX NORMAL と DATA NORMAL) が NORMAL から NONE に上書きされます。

DB2 についてのデータの収集

時には、単に症状をトラブルシューティングするだけでは問題を解決できない場合があります。そのような場合、診断データを収集する必要があります。収集する必要がある診断データと、そのデータの収集元のソースは、調査中の問題のタイプにより異なります。以下の手順は、問題を IBM ソフトウェア・サポートに送信する際に通常提供する必要がある情報の基本セットを収集する方法を示しています。

完全な出力を得るには、インスタンス所有者が db2support ユーティリティーを呼び出す必要があります。

圧縮ファイル・アーカイブ内の診断情報の基本セットを収集するには、次のように db2support コマンドを入力します。

```
db2support <output_directory> -s -d <database_name> -c
```

-s を使用すると、使用中のハードウェアとオペレーティング・システムに関するシステムの詳細が得られます。-d を使用すると、指定されたデータベースに関する詳細が得られます。-c を使用すると、指定されたデータベースに対する接続試行が可能になります。

出力は適切に編成されて ZIP アーカイブ db2support.zip に圧縮されるため、任意のシステムに転送して簡単に解凍することができます。

特定の症状、または製品の特定の部分の問題については、追加データを収集する必要があるかもしれません。問題別の『データの収集』文書を参照してください。

次に以下のいずれかのタスクを行うことができます。

- データを分析する
- IBM ソフトウェア・サポートにデータを送信する

インストール問題についてのデータの収集

インストール問題が発生しているものの、問題の原因が判別できない場合、問題を診断して解決するために、あなた自身または IBM ソフトウェア・サポートが使用できる診断データを収集してください。

インストール問題の診断データを収集するには、以下のようになります。

1. オプション: トレースを有効にしてインストールの試行を繰り返します。例えば、以下のようになります。

```
db2setup -t trace.out
```

または

```
setup -t trace.out
```

2. インストール・ログ・ファイルを見つけます。

- Windows では、デフォルトのファイル名は *DB2-ProductAbbreviation-DateTime.log* です。例えば、*DB2-ESE-Wed Jun 21 11_59_37 2006.log* のようになります。インストール・ログのデフォルトのロケーションは、"My Documents"¥DB2LOG¥ ディレクトリーです。
- Linux および UNIX では、デフォルトのファイル名は、*db2setup.log*、*db2setup.his*、および *db2setup.err* です。

トレース (またはデバッグ・モード) を有効にした状態で問題が再現される場合、*dascrt.log*、*dasdrop.log*、*dasupdt.log*、*db2icrt.log.PID*、*db2idrop.log.PID*、*db2imigr.log.PID*、*db2iupdt.log.PID* などの追加ファイルが作成される場合があります。ここで、PID はプロセス ID です。

これらすべてのファイルのデフォルトのロケーションは */tmp* ディレクトリーです。

3. オプション: データを IBM ソフトウェア・サポートに送信する場合、DB2 のデータも収集してください。

データ移動問題についてのデータの収集

データ移動コマンドの実行中に問題が発生し、その問題の原因が判別できない場合は、問題を診断して解決するために、あなた自身または IBM ソフトウェア・サポートが使用できる診断データを収集してください。

- *db2move* コマンドに関連した問題のデータを収集するには、コマンドを発行したディレクトリーに移動してください。コマンドで指定したアクションに応じて、以下のファイルを見つけます。
 - COPY アクションの場合、*COPY.timestamp.ERR* および *COPYSHEMA.timestamp.MSG* というファイルを探します。さらに *LOAD_ONLY* または *DDL_AND_LOAD* モードのいずれかを指定した場合、*LOADTABLE.timestamp.MSG* というファイルも探してください。
 - EXPORT アクションの場合、*EXPORT.out* というファイルを探してください。
 - IMPORT アクションの場合、*IMPORT.out* というファイルを探してください。
 - LOAD アクションの場合、*LOAD.out* というファイルを探してください。
- EXPORT、IMPORT、または LOAD コマンドに関連した問題のデータを収集するには、コマンドに *MESSAGES* パラメーターが含まれているかどうかを判別します。含まれている場合、出力ファイルを収集します。これらのユーティリティーは、現行ディレクトリーおよびデフォルト・ドライブ以外を指定しない場合にはそれらを宛先として使用します。
- REDISTRIBUTE コマンドに関連した問題のデータを収集するには、*databasename.database_partition_groupname.timestamp* (Linux および UNIX の場合) および *databasename.database_partition_groupname.date.time* (Windows の場合) と

いうファイルを探してください。それは、`$HOME/sqllib/db2dump` ディレクトリー
または `$DB2PATH/sqllib/redis` にそれぞれ置かれています (`$HOME` はインスタ
ンス所有者のホーム・ディレクトリー)。

DAS およびインスタンス管理の問題についてのデータの収集

DB2 管理サーバー (DAS) またはインスタンス管理を実行中に問題が発生している
ものの、問題の原因が判別できない場合、問題を診断して解決するために、あなた
自身または IBM ソフトウェア・サポートが使用できる診断データを収集してくだ
さい。

以下の手順は、問題を再現でき、Linux または UNIX 上で DB2 を使用している場
合のみ、使用できます。

DAS またはインスタンス管理問題の診断データを収集するには、以下のようにし
ます。

1. トレースまたはデバッグ・モードを有効にして、失敗したコマンドを繰り返しま
す。コマンドの例を以下に示します。

```
db2setup -t trace.out
dascrt -u DASUSER -d
dasdrop -d
dasmigr -d
dasupdt -d
db2icrt -d INSTNAME
db2idrop INSTNAME -d
db2imigr -d INSTNAME
db2iupdt -d INSTNAME
```

2. 診断ファイルを見つけます。複数のファイルが存在する可能性があるため、該当
するすべてのファイルを確認するためにタイム・スタンプを比較してくだ
さい。

デフォルトでは、出力は `/tmp` ディレクトリーに出力されます。

ファイル名の例として、`dascrt.log`、`dasdrop.log`、`dasupdt.log`、`db2icrt.log.PID`、
`db2idrop.log.PID`、`db2imigr.log.PID`、`db2iupdt.log.PID` などがあります。こ
こで、PID はプロセス ID です。

3. 診断ファイルを IBM ソフトウェア・サポートに提出します。

問題が `db2start` または `START DATABASE MANAGER` コマンドの失敗である場
合、`insthome/sqllib/log` ディレクトリーで (`insthome` はインスタンス所有者のホー
ム・ディレクトリー) `db2start.timestamp.log` という名前のファイルを探してくださ
い。同様に、問題が `db2stop` または `STOP DATABASE MANAGER` コマンドの失
敗である場合、`db2stop.timestamp.log` という名前のファイルを探してください。これ
らのファイルは、`start_stop_time` データベース・マネージャー構成パラメーターで指
定した長さの時間内にデータベース・マネージャーがコマンドに応答しなかった場
合にのみ、出力されます。

DB2 についてのデータの分析

データを収集した後、そのデータが現在の特定の問題を解決するためにどのように役立つかを判別する必要があります。分析のタイプは、調査中の問題のタイプと、収集したデータにより異なります。以下の手順は、基本的な DB2 診断データの調査を開始する方法を示しています。

診断データを分析するには、以下のアクションを実行します。

- データのさまざまな断片が互いにどのように関連するかについて明確に理解するようにします。例えば、データが複数のシステムにまたがる場合、データのどの断片がどのソースのものなのかが分かるように、データの編成を工夫します。
- タイム・スタンプを調べて、問題が発生したタイミングと診断データの各断片とが関連していることを確認します。データのソースが異なると、タイム・スタンプの形式が異なる場合がある点に注意してください。さまざまなイベントが発生した時刻が分かるように、それぞれのタイム・スタンプ形式における各要素の表示順序を理解しておきます。
- 問題に関する情報が含まれる可能性が最も高いデータ・ソースを判別し、そこで分析を開始します。例えば、問題がインストールに関連している場合、製品全般またはオペレーティング・システムのログ・ファイルから分析を開始するのではなく、インストール・ログ・ファイルがあるならば、そこから分析を開始します。
- 分析の具体的な方法はデータ・ソースごとに固有のものですが、ほとんどのトレースおよびログ・ファイルに当てはまる 1 つのヒントは、問題が発生したデータ内のポイントをまず最初に突き止めることです。そのポイントを突き止めたら、問題の根本原因を解明するために、そのデータを時間をさかのぼって調べることができます。
- ある問題を調査していて、作動中の環境と作動していない環境におけるその問題の比較データがある場合には、各環境のオペレーティング・システムと製品構成の詳細を比較することから始めます。

インストールの問題に関するデータの分析

インストールの問題に関する診断データを収集したら、データを分析して、問題の原因を判別できます。これらのステップはオプションです。問題の原因が簡単に判明しない場合は、データを IBM ソフトウェア・サポートに送信してください。

以下のステップでは、インストール問題についてのデータの収集で説明されているファイルを入手済みであることを前提としています。

1. 適切なインストール・ログ・ファイルを参照していることを確認します。ファイルの作成日や、ファイル名に含まれているタイム・スタンプ (Windows オペレーティング・システムの場合) を確認します。
2. インストールが正常に完了したかどうかを判別します。
 - Windows オペレーティング・システムの場合、正常に完了したことは、インストール・ログ・ファイルの末尾にある、以下に類似したメッセージによって示されます。


```
Property(C): INSTALL_RESULT = Setup Complete Successfully
=== Logging stopped: 6/21/2006 16:03:09 ===
MSI (c) (34:38) [16:03:09:109]:
Product: DB2 Enterprise Server Edition - DB2COPY1 -- Installation operation
completed successfully.
```

- Linux および UNIX オペレーティング・システムの場合、正常に完了したことは、インストール・ログ・ファイル (デフォルトの名前は db2setup.log) の末尾にあるメッセージによって示されます。
3. オプション: エラーが発生したかどうかを判別します。インストールが正常に完了したものの、インストール・プロセス中にエラー・メッセージを受け取った場合、そのエラーをインストール・ログ・ファイル内で見つけます。

- Windows オペレーティング・システムの場合、大部分のエラーは先頭に ERROR: または WARNING: が付いています。例えば、以下のようになります。

```
1: ERROR:An error occurred while running the command
"D:¥IBM¥SQLLIB¥bin¥db2.exe
CREATE TOOLS CATALOG SYSTOOLS USE EXISTING DATABASE TOOLSDB FORCE" to
initialize and/or migrate the DB2 tools catalog database. The return value is "4".
```

```
1: WARNING:A minor error occurred while installing "DB2 Enterprise Server
Edition - DB2COPY1" on this computer. Some features may not function
correctly.
```

- Linux および UNIX オペレーティング・システムでは、Java™ によってエラーが戻された場合 (例えば、例外やトラップ情報)、デフォルト名が db2setup.err というファイルが存在します。

インストール・トレースを有効にしてあった場合は、インストール・ログ・ファイルに含まれる項目は多くなり、より詳細なものになります。

このデータを分析しても問題解決の助けにならない場合、IBM との保守契約があれば、問題報告書をオープンできます。IBM ソフトウェア・サポートでは、収集したデータの送信を依頼します。また、実行した分析についてお尋ねすることもあります。

調査しても問題が解決しない場合は、データを IBM ソフトウェア・サポートに送信してください。

IBM ソフトウェア・サポートへのデータの送信

この手順では、IBMソフトウェア・サポートを使用してすでに Problem Management Record (PMR) を開いていると想定しています。

ログ・ファイル、構成ファイルなどの診断データを、以下のいずれかの方法を使って IBM ソフトウェア・サポートに送信することができます。

- FTP
- Electronic Service Request (ESR) ツール
- ファイルを (FTP で) Enhanced Centralized Client Data Repository (EcuRep) に送信するには、次のようにします。
 1. 収集したデータ・ファイルを ZIP または TAR 形式に圧縮し、Problem Management Record (PMR) ID にしたがって圧縮ファイルに名前を付けます。

ファイルを PMR に正しく関連付けるために、xxxxx.bbb.ccc.yyy.yyy という命名規則を使用する必要があります (xxxxx は PMR 番号、bbb は PMR のブランチ番号、ccc は PMR のテリトリー・コード、yyy.yyy はファイル名です)。

2. FTP ユーティリティーを使用して、サーバー ftp.emea.ibm.com に接続します。
 3. ユーザー ID は「anonymous」でログインして、パスワードに自分の E メール・アドレスを入力します。
 4. toibm ディレクトリーに進みます。例えば、cd toibm を実行します。
 5. オペレーティング・システムに固有のサブディレクトリーの 1 つに移動します。例えば、aix、linux、unix、windows などのサブディレクトリーがあります。
 6. バイナリー・モードに変更します。例えば、コマンド・プロンプトで bin を入力します。
 7. put コマンドを使用して、ファイルをサーバー上に配置します。ファイルに名前を付けてサーバー上に配置する場合は、以下のファイル命名規則を使用します。PMR が更新され、xxxx.bbb.ccc.yyy.yyy という形式でファイルの格納場所がリストされます。(xxx は PMR 番号、bbb はブランチ、ccc はテリトリー・コード、yyy.yyy は tar.Z または xyz.zip のようなファイル・タイプの記述です。) ファイルを FTP サーバーに送信することはできますが、それを更新することはできません。後でファイルを変更する必要がある場合には、新しいファイル名を作成する必要があります。
 8. quit コマンドを入力します。
- ESR ツールを使用してファイルを送信するには、以下のようになります。
 1. ESR にサインオンします。
 2. ウェルカム・ページで、「**Enter a report number**」フィールドに PMR 番号を入力し、「**Go**」をクリックします。
 3. 「**Attach Relevant File**」フィールドまでスクロールダウンします。
 4. 「**Browse**」をクリックして、IBM ソフトウェア・サポートに送信するログ、トレース、またはその他の診断ファイルを見つけます。
 5. 「**Submit**」をクリックします。ファイルは FTP で IBM ソフトウェア・サポートに転送され、PMR と関連付けられます。

EcuRep サービスの詳細については、IBM EMEA Centralized Customer Data Store Service を参照してください。

ESR についての詳細は、Electronic Service Request (ESR) ヘルプを参照してください。

第 3 章 DB2 Connect のトラブルシューティング

DB2 Connect を使用するシステム環境では、ローカル・データベースの場合に通常関係する範囲を超えた、固有の問題が発生する可能性があります。問題に関する情報収集では、通信プロトコルおよびハードウェアの関与も必要になります。加えて、リモート・データベースが問題発生源になる可能性があるため、考慮する必要があります。

トラブルシューティング

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、分かっている事柄、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 ゲートウェイとホストまたは System i™ データベース・サーバーの間ですか。それとも DB2 Connect Personal Edition とホストまたは System i 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、ローカルに行っても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. ホストまたは System i 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. ホストまたは System i 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。ホストまたは System i 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となります。
 - 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、クライアントへのホストまたは System i 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。
 - ホストまたは System i 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. ホストまたは System i サーバー・データベースを使用するために、ホストまたは System i 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. ホストまたは System i 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ホストまたは System i 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。
 - フィックスパ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パックを判別し、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必要があります。
 - *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. *DB2START* 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。たとえば、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/sqlllib/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** および **db2support** 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。

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 i5/OS、DB2 for z/OS®、および DB2 for VM & VSE からの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

ホストまたは i5/OS 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発信してください。

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.log 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、svcname 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 1 つあげられるのは、以前に成功した接続がそのままになっており、構成が変更されていないのに、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 OS/390® and 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 誤ったホストまたは System i 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。例えば、DB2 Universal Database™ for z/OS and OS/390 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
```

その後に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 サーバーからホストまたは System i 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仕様が、IBM データ・サーバー・クライアント 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.log` 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。

```
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。

問題を修正する方法は、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ることです。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

必要であれば CONNECT ステートメントに正しいパスワードを指定してあ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使用できないパスワードです。パスワードを IBM データ・サーバー・クライアント 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 4 章 トラブルシューティングのためのツール

データベース・マネージャー内部の戻りコード、DB2 製品の一部であるツール、各種のトレース、およびオペレーティング・システムの一部であるツールが、問題のトラブルシューティングの際にすべて使用されます。データベースの問題を調査しているときには、各ツールがユーザーを支援するデータと情報、および DB2 サポートを提供します。

内部戻りコードに関する詳細情報

内部戻りコードには、ZRC 値と ECF 値という 2 つのタイプがあります。これらは、IBM ソフトウェア・サポートが使用するための診断ツールでのみ通常は表示される戻りコードです。例えば、これらは DB2 トレース出力および db2diag.log ファイルに出現します。

ZRC 値と ECF 値は基本的には同じ目的のためのものですが、形式が少し異なります。各 ZRC 値には、以下の特性があります。

- クラス名
- コンポーネント
- 理由コード
- 関連した SQLCODE
- SQLCA メッセージ・トークン
- 説明

一方、ECF 値は以下のもので構成されます。

- セット名
- 製品 ID
- コンポーネント
- 説明

ZRC 値と ECF 値は通常、負の数値で、エラー状態を表すために使用されます。ZRC 値は、それが表すエラーのタイプによってグループ化されています。このグループ分けは、「クラス」と呼ばれます。例えば、「SQLZ_RC_MEMHEP」で始まる名前の ZRC 値は、一般にメモリー不足に関するエラーです。同様に、ECF 値は「セット」にグループ化されています。

ZRC 値を含む db2diag.log 項目の例を以下に示します。

```
2006-02-13-14.34.35.965000-300    I17502H435    LEVEL: Error
PID      : 940                TID   : 660    PROC  : db2syscs.exe
INSTANCE: DB2                NODE  : 000    DB   : SAMPLE
APPHDL   : 0-1433            APPID: *LOCAL.DB2.050120082811
FUNCTION: DB2 UDB, data protection, sqlpsize, probe:20
RETCODE  : ZRC=0x860F000A=-2045837302=SQLO_FNEX "File not found."
                DIA8411C A file "" could not be found.
```

この ZRC 値に関する完全な詳細は、例えば次のような db2diag コマンドを使用して入手できます。

```

c:¥>db2diag -rc 0x860F000A

Input ZRC string '0x860F000A' parsed as 0x860F000A (-2045837302).

ZRC value to map: 0x860F000A (-2045837302)
      V7 Equivalent ZRC value: 0xFFFFE60A (-6646)

ZRC class :
      Critical Media Error (Class Index: 6)
Component:
      SQL0 ; oper system services (Component Index: 15)
Reason Code:
      10 (0x000A)

Identifer:
      SQL0_FNEX
      SQL0_MOD_NOT_FOUND
Identifer (without component):
      SQLZ_RC_FNEX

Description:
      File not found.

Associated information:
      Sqlcode -980
SQL0980C A disk error occurred. Subsequent SQL statements cannot be
processed.

      Number of sqlca tokens : 0
      Diaglog message number: 8411

```

コマンド db2diag -rc -2045837302 または db2diag -rc SQL0_FNEX を発行した場合も、同じ情報が戻されます。

ECF 戻りコードの出力の例を以下に示します。

```

c:¥>db2diag -rc 0x90000076

Input ECF string '0x90000076' parsed as 0x90000076 (-1879048074).

ECF value to map: 0x90000076 (-1879048074)

ECF Set :
      setecf (Set index : 1)
Product :
      DB2 Common
Component:
      OSSe
Code:
      118 (0x0076)

Identifier:
      ECF_LIB_CANNOT_LOAD

Description:
      Cannot load the specified library

```

db2diag コマンド出力内で最も価値のあるトラブルシューティング情報は、説明および関連情報 (ZRC 戻りコードのみ) です。

ZRC 値または ECF 値の完全なリストを参照するには、それぞれコマンド db2diag -rc zrc および db2diag -rc ecf を使用します。

db2dart ツールの概要

db2dart コマンドは、データベースとその内部のオブジェクトの体系的な正確さを検査するのに使用できます。他の方法ではアクセスできない表からデータを抽出するために、データベース制御ファイルの内容を表示するのに使用できます。

使用可能なオプションすべてを表示するには、パラメーターを何も指定せず、単に db2dart コマンドを発行します。表スペース ID などのパラメーターを必要とするオプションによっては、コマンド行で明示的に指定されていない場合、プロンプトが出されます。

デフォルトでは、db2dart ユーティリティーは databaseName.RPT という名前のレポート・ファイルを作成します。単一パーティションのデータベース・パーティション環境の場合、ファイルは現行ディレクトリーで作成されます。複数パーティションのデータベース・パーティション環境の場合、ファイルは診断ディレクトリー中のサブディレクトリーの下に作成されます。サブディレクトリーは DART#### で、### はデータベース・パーティション番号です。

db2dart ユーティリティーは、ディスクから直接読み取ることによってデータベース中のデータおよびメタデータにアクセスします。そのため、まだアクティブな接続があるデータベースに対してこのツールを実行してはなりません。接続が存在する場合、このツールはバッファ・プール中のページまたはメモリー内の制御構造などを認識しないので、結果として誤ったエラーが報告される可能性があります。同様に、クラッシュ・リカバリーが必要か、またはロールフォワード・リカバリーが完了していないデータベースに対して db2dart を実行すると、ディスク上のデータの不整合の性質のため、同様の不整合が発生する場合があります。

INSPECT と db2dart の比較

INSPECT コマンドを使用すれば、データベースのアーキテクチャーの保全性を検査して、データベース内のページの整合性をチェックできます。INSPECT コマンドでは、表オブジェクトと表スペースの構造が有効であることをチェックします。オブジェクト間妥当性検査により、オンラインで索引とデータの整合性検査を実行できます。db2dart コマンドは、データベースのアーキテクチャーの正確さを検査し、検出されたあらゆるエラーを報告します。

INSPECT コマンドは、db2dart コマンドと同じく、データベース、表スペース、表をチェックします。この 2 つのコマンドの大きな違いは、db2dart を実行する前にはデータベースを非活動化する必要がありますが、INSPECT ではデータベース接続が必要で、データベースへのアクティブ接続が同時に存在する場合でも実行できるという点です。

データベースを非活動化しなかった場合、db2dart の実行結果は信頼できません。

db2dart コマンドと INSPECT コマンドが実行するテストの違いを以下の表にまとめます。

表 1. 表スペースに対する db2dart と INSPECT のフィーチャーの比較

実行されるテスト	db2dart	INSPECT
SMS 表スペース		

表1. 表スペースに対する db2dart と INSPECT のフィーチャーの比較 (続き)

実行されるテスト	db2dart	INSPECT
表スペース・ファイルの検査	あり	なし
内部ページ・ヘッダー・フィールドの内容の検証	あり	あり
DMS 表スペース		
複数のオブジェクトが指し示すエクステント・マップの検査	あり	なし
すべてのエクステント・マップ・ページでの整合性ビット・エラーの検査	なし	あり
すべてのスペース・マップ・ページでの整合性ビット・エラーの検査	なし	あり
内部ページ・ヘッダー・フィールドの内容の検証	あり	あり
エクステント・マップが表スペース・マップと一致することの検査	あり	なし

表2. データ・オブジェクトに対する db2dart と INSPECT のフィーチャーの比較

実行されるテスト	db2dart	INSPECT
データ・オブジェクトでの整合性ビット・エラーの検査	あり	あり
特殊制御行の内容の検査	あり	なし
可変長列の長さや位置の検査	あり	なし
表の行での LONG VARCHAR、LONG VARGRAPHIC、およびラージ・オブジェクト (LOB) の記述子の検査	あり	なし
サマリー合計ページ、使用済みページ、およびフリー・スペースの割合の検査	なし	あり
内部ページ・ヘッダー・フィールドの内容の検証	あり	あり
各行レコード・タイプとその長さの検査	あり	あり
行がオーバーラップしていないことの検査	あり	あり

表3. 索引オブジェクトに対する db2dart と INSPECT のフィーチャーの比較

実行されるテスト	db2dart	INSPECT
整合性ビット・エラーの検査	あり	あり

表 3. 索引オブジェクトに対する *db2dart* と *INSPECT* のフィーチャーの比較 (続き)

実行されるテスト	<i>db2dart</i>	<i>INSPECT</i>
索引キーの位置と長さ、およびオーバーラップがあるかどうかの検査	あり	あり
索引におけるキーの順序付けの検査	あり	なし
サマリー合計ページと使用済みページの判別	なし	あり
内部ページ・ヘッダー・フィールドの内容の検証	あり	あり
ユニーク・キーの一意性の検査	あり	なし
指定された索引項目のデータ行が存在するかどうかの検査	なし	あり
データ値に照らした各キーの検査	なし	あり

表 4. ブロック・マップ・オブジェクトに対する *db2dart* と *INSPECT* のフィーチャーの比較

実行されるテスト	<i>db2dart</i>	<i>INSPECT</i>
整合性ビット・エラーの検査	あり	あり
サマリー合計ページと使用済みページの判別	なし	あり
内部ページ・ヘッダー・フィールドの内容の検証	あり	あり

表 5. 長いフィールドおよび *LOB* オブジェクトに対する *db2dart* と *INSPECT* のフィーチャーの比較

実行されるテスト	<i>db2dart</i>	<i>INSPECT</i>
割り振り構造の検査	あり	あり
サマリー合計ページと使用済みページの判別 (<i>LOB</i> オブジェクトのみ)	なし	あり

さらに、*db2dart* コマンドを使用すると、以下のアクションを実行できます。

- データ・ページのフォーマット設定とダンプ
- 索引ページのフォーマット設定とダンプ
- 区切り付き ASCII でのデータ行のフォーマット設定
- 索引に無効のマークを付ける

INSPECT コマンドを使用して、これらのアクションを実行することはできません。

db2diag を使用した db2diag.log ファイルの分析

データベース管理者とシステム管理者が使用するための 1 次ログ・ファイルは管理通知ログです。db2diag.log ファイルは、トラブルシューティングの目的で DB2 サポートが使用するためのものです。

管理通知ログ・メッセージは、標準化されたメッセージ・フォーマットで db2diag.log にも記録されます。

db2diag ツールによって、db2diag.log で使用可能な大量の情報をフィルターに掛けたり、フォーマットすることが可能です。db2diag.log のレコードをフィルター処理することで、問題のトラブルシューティングに必要なレコードを見つけるための時間を短縮できます。

例 1: データベース名による db2diag.log のフィルター処理

インスタンス内にデータベースがいくつかあり、データベース SAMPLE に関連したメッセージだけを参照する場合には、以下のように db2diag.log をフィルターに掛けることができます。

```
db2diag -g db=SAMPLE
```

そうすると、以下のように、“DB: SAMPLE” を含んだ db2diag.log レコードだけが表示されます。

```
2006-02-15-19.31.36.114000-300 E21432H406          LEVEL: Error
PID      : 940                TID : 660          PROC : db2syscs.exe
INSTANCE: DB2                NODE : 000         DB   : SAMPLE
APPHDL   : 0-1056            APPID: *LOCAL.DB2.060216003103
FUNCTION: DB2 UDB, base sys utilities, sqlDatabaseQuiesce, probe:2
MESSAGE  : ADM7507W Database quiesce request has completed successfully.
```

例 2: プロセス ID による db2diag.log のフィルター処理

以下のコマンドを使用して、パーティション 0、1、2、または 3 で実行している、プロセス ID (PID) 2200 のプロセスによって生成されるすべての重大エラー・メッセージを表示できます。

```
db2diag -g level=Severe,pid=2200 -n 0,1,2,3
```

このコマンドは 2 種類の異なる方法で書けることに注意してください。すなわち、db2diag -l severe -pid 2200 -n 0,1,2,3 と書くこともできます。-g オプションは大/小文字の区別のある検索を指定するので、この場合は “Severe” なら問題ありませんが、“severe” が使用されていると失敗することにも注意してください。これらのコマンドは、以下のような、これらの要件にかなう db2diag.log レコードを正常に検索します。

```
2006-02-13-14.34.36.027000-300 I18366H421          LEVEL: Severe
PID      : 2200                TID : 660          PROC : db2syscs.exe
INSTANCE: DB2                NODE : 000         DB   : SAMPLE
APPHDL   : 0-1433            APPID: *LOCAL.DB2.060213193043
FUNCTION: DB2 UDB, data management, sqlPoolCreate, probe:273
RETCODE  : ZRC=0x8002003C=-2147352516=SQLB_BAD_CONTAINER_PATH
          "Bad container path"
```

例 3: db2diag ツール出力のフォーマット

以下のコマンドでは、2006年1月1日より後に生じた、非重大エラーと重大エラーを含む、パーティション 0、1、または 2 に関して記録されたレコードすべてをフィルタリングします。これによって適合するレコードが出力されます。最初の行にはタイム・スタンプ、パーティション番号、およびレベルが表示され、2 番目の行には pid、tid、およびインスタンス名が表示され、その後エラー・メッセージが表示されます。

```
db2diag -time 2006-01-01 -node "0,1,2" -level "Severe, Error" | db2diag -fmt
"Time: %{ts}
Partition: %node Message Level: %{level} %nPid: %{pid} Tid: %{tid}
Instance: %{instance}%nMessage: @{msg}%n"
```

以下に、作成される出力の例を挙げます。

```
Time: 2006-02-15-19.31.36.099000 Partition: 000 Message Level: Error
Pid: 940 Tid:940 Instance: DB2
Message: ADM7506W Database quiesce has been requested.
```

詳細については、以下のコマンドを発行してください。

- db2diag -help は、すべての使用可能なオプションの簡略説明を示します。
- db2diag -h brief は、すべてのオプションの記述を示します (例は表示しません)。
- db2diag -h notes は、使用上の注意および制約事項を示します。
- db2diag -h examples は、始めに参照できるいくつかの例を示します。
- db2diag -h tutorial は、使用可能なすべてのオプションの例を示します。
- db2diag -h all は、オプションの最も完全なリストを示します。

例 4: 各種機能から送られてくるメッセージのフィルター処理

以下の例では、データベース・マネージャー内の特定の機能から送られてくるメッセージだけ (またはすべての機能から送られてくるメッセージ) を表示する方法を示します。サポートされている機能は、以下のとおりです。

- ALL (すべての機能からのレコードを返します)
- MAIN (db2diag.log などの DB2 一般診断ログや管理通知ログからのレコードを返します)
- OPSTATS (オペティマイザー統計に関連したレコードを返します)

MAIN 機能から送られてくるメッセージを読み取るには、以下のようになります。

```
db2diag -facility MAIN
```

OPSTATS 機能から送られてくるメッセージを表示し、フィルターによって Severe レベルのレコードに絞り込むには、以下のようになります。

```
db2diag -fac OPSTATS -level Severe
```

使用可能なすべての機能から送られてくるメッセージを表示し、フィルターによって instance=harmistr と level=Error のレコードに絞り込むには、以下のようになります。

```
db2diag -fac all -g instance=harmistr,level=Error
```

OPSTATS 機能から送られてくる Error レベルのメッセージをすべて表示してから、Timestamp と PID フィールドを特定の形式で出力するには、以下のようになります。

```
db2diag -fac opstats -level Error -fmt " Time :%{ts} Pid :%{ts}"
```

db2greg を使用したグローバル・レジストリーの表示および変更 (UNIX)

グローバル・レジストリーは、UNIX および Linux プラットフォームにのみ存在します。

- root インストールの場合、グローバル・レジストリー・ファイルは /var/db2/global.reg (HP-UX では /var/opt/db2/global.reg)にあります。
- 非 root インストールの場合、グローバル・レジストリー・ファイルは、\$HOME/sqllib/global.reg にあります (\$HOME は非 root ユーザーのホーム・ディレクトリーです)。

グローバル・レジストリーは、以下の 3 つの異なるレコード・タイプから成っています。

- "Service": Service レコードには、製品レベルの情報が含まれています。例えば、バージョンやインストール・パスなどです。
- "Instance": Instance レコードには、インスタンス・レベルの情報が含まれています。例えば、インスタンス名、インスタンス・パス、バージョン、start-at-boot フラグなどです。
- "Variable": Variable レコードには、変数レベルの情報が含まれています。例えば、変数名、変数値、コメントなどです。

グローバル・レジストリーは db2greg ツールを使用して表示できます。このツールは sqllib/bin にあります。また、install ディレクトリーの bin の下にもあります (root としてログインしたときに使用するため)。

グローバル・レジストリーは、db2greg ツールを使用して編集できます。root インストールでのグローバル・レジストリーの編集には、root 権限が必要です。

db2greg ツールは、DB2 お客様サポートからの要請があった場合にだけ使用すべきです。

製品のバージョンとサービス・レベルの識別

db2level コマンドを使用すれば、ご使用の DB2 インスタンスのバージョンとサービス・レベル (ビルド・レベルおよびフィックスパック番号) を判別できます。DB2 インスタンスが最新のサービス・レベルであるかどうかを判別するには、db2level の出力結果と、DB2 サポートの Web サイト (<http://www.ibm.com/software/data/db2/udb/support.html>) のフィックスパック・ダウンロード・ページに表示される情報を比べてください。

db2level コマンドを Windows システムで実行した場合の典型的な結果は次のようなものです。

```
DB21085I Instance "DB2" uses "32" bits and DB2 code release "SQL09010" with
level identifier "01010107".
Informational tokens are "DB2 v9.1.0.189", "n060119", "", and Fix Pack "0".
Product is installed at "c:\SQLLIB" with DB2 Copy Name "db2build".
```

4 つの情報トークンを組み合わせれば、ご使用の DB2 インスタンスの正確なサービス・レベルを固有に識別できます。IBM お客様サポートに連絡して支援を受けるには、この情報が必要になります。

JDBC または SQLJ アプリケーションで SQLJ および JDBC 用の IBM DB2 ドライバーを使用している場合には、以下のように db2jcc ユーティリティを実行することによってドライバーのレベルを判別できます。

```
db2jcc -version
```

```
IBM DB2 JDBC Driver Architecture 2.3.63
```

db2look を使用したデータベースの模造

あるデータベースと構造が類似した別のデータベースを作成できると便利な場合がよくあります。例えば、新しいアプリケーションやリカバリー・プランを実動システムでテストするより、ほぼ同じ構造とデータを持つテスト・システムを作成して、代わりにそのテスト・システムでテストする方が理にかなっています。こうすることにより、実動システムの方は、テストによってパフォーマンス低下の影響を受けたり、エラーがあるアプリケーションによって誤ってデータを破壊されたりする心配がなくなります。また、問題（無効な結果やパフォーマンスの問題など）を調査する際には、実動システムとまったく同じテスト・システムを使った方が問題をデバッグしやすい場合もあります。

db2look ツールを使用すると、あるデータベースのデータベース・オブジェクトを別のデータベースに複製するのに必要な DDL ステートメントを抽出することができます。このほか、あるデータベースから別のデータベースに統計情報を複製するのに必要な SQL ステートメントや、データベース構成、データベース・マネージャ構成、およびレジストリー変数の複製に必要なステートメントを生成することもできます。これは重要なことです。なぜなら、これにより、新しいデータベースに元のデータベースとまったく同じデータ集合が含まれていなくても、2 つのシステムで同じアクセス・プランが選択されるようにすることができるからです。

db2look ツールの詳細については、「DB2 コマンド・リファレンス」を参照してください。オプションのリストについては、何もパラメーターを指定せずにこのツールを実行すると表示されます。-h オプションを使用すると、より詳しい使い方が表示されます。

db2look によるデータベース内の表の模造

データベース内の表の DDL を抽出するには、-e オプションを使用します。例として、SAMPLE2 という SAMPLE データベースのコピーを作成してみます。最初のデータベース内のすべてのオブジェクトが新しいデータベースに作成されるようにします。

```
C:¥>db2 create database sample2
DB20000I The CREATE DATABASE command completed successfully.
C:¥>db2look -d sample -e > sample.ddl
-- USER is:
-- Creating DDL for table(s)
-- Binding package automatically ...
-- Bind is successful
-- Binding package automatically ...
-- Bind is successful
```

注: ユーザー定義のスペース、データベース・パーティション・グループ、およびバッファ・プールの DDL も作成する場合には、上記のコマンドの `-e` の後に `-l` フラグを追加します。デフォルトのデータベース・パーティション・グループ、バッファ・プール、および表スペースは抽出されません。なぜなら、それらはすでにデフォルトで、すべてのデータベースに存在しているからです。これらを模造する場合には、それらを手動で変更する必要があります。

`sample.ddl` ファイルをテキスト・エディターで開きます。このファイルの DDL を新しいデータベースに対して実行するには、`CONNECT TO SAMPLE` ステートメントを `CONNECT TO SAMPLE2` に変更する必要があります。 `-l` オプションを使用した場合には、表スペース・コマンドに関連したパスも、適切なパスを指すように変更する必要があるかもしれません。この作業の間に、ファイルの内容の残りの部分を見てください。サンプル・データベース内のすべてのユーザー表について、`CREATE TABLE`、`ALTER TABLE`、および `CREATE INDEX` ステートメントがあるはずです。

```
...
-----
-- DDL Statements for table "DB2"."ORG"
-----

CREATE TABLE "DB2"."ORG" (
  "DEPTNUMB" SMALLINT NOT NULL ,
  "DEPTNAME" VARCHAR(14) ,
  "MANAGER" SMALLINT ,
  "DIVISION" VARCHAR(10) ,
  "LOCATION" VARCHAR(13) )
  IN "USERSPACE1" ;
...
```

接続ステートメントを変更したら、以下のように、そのステートメントを実行します。

```
C:¥>db2 -tvf sample.ddl > sample2.out
```

出力ファイル `sample2.out` で、すべてが正しく実行されていることを確認します。エラーが発生した場合は、エラー・メッセージで問題を確認できます。それらの問題を修正したら、もう一度ステートメントを実行します。

出力を見るとわかるように、すべてのユーザー表の DDL がエクスポートされています。これはデフォルトの動作ですが、操作に含める表を細かく指定するためのオプションもあります。例えば、`STAFF` 表と `ORG` 表のみを含めるには、以下のように `-t` オプションを使用します。

```
C:¥>db2look -d sample -e -t staff org > staff_org.ddl
```

スキーマ `DB2` を持つ表だけを含めるには、以下のように `-z` オプションを使用します。

```
C:¥>db2look -d sample -e -z db2 > db2.ddl
```

表の統計の模造

テスト・データベースの目的がパフォーマンスのテストやパフォーマンスの問題のデバッグにある場合は、両方のデータベースで同一のアクセス・プランが生成されるようにすることが重要です。最適化プログラムは、統計、構成パラメーター、レ

ジストリー変数、および環境変数に基づいてアクセス・プランを生成します。これらが 2 つのシステムで同一であれば、同じアクセス・プランが生成されと考えられます。

両方のデータベースにまったく同じデータをロードし、同じオプションを指定して `runstats` を実行すれば、統計は同じになるはずですが、データベースに格納されているデータが異なっていたり、データの一部しかテスト・データベースで使われていなかったりすると、まったく異なる統計になります。このような場合は、`db2look` を使用すると、実動データベースから統計情報を収集して、それをテスト・データベースに追加することができます。そのためには、更新可能なカタログ表の `SYSSTAT` セットに対する `UPDATE` ステートメントと、すべての表に対する `RUNSTATS` コマンドを作成します。

統計ステートメントを作成するためのオプションは `-m` です。 `SAMPLE/SAMPLE2` の例に戻って、`SAMPLE` からの統計を収集し、それらを `SAMPLE2` に追加します。

```
C:¥>db2look -d sample -m > stats.dml
-- USER is:
-- Running db2look in mimic mode
```

前と同様に、出力ファイルを編集して、`CONNECT TO SAMPLE` ステートメントを `CONNECT TO SAMPLE2` に変更する必要があります。ここでまたファイルの残りの部分を見て、`RUNSTATS` ステートメントや `UPDATE` ステートメントがどのようになっているのかを確認します。

```
...
-- Mimic table ORG
RUNSTATS ON TABLE "DB2"."ORG" ;

UPDATE SYSSTAT.INDEXES
SET NLEAF=-1,
    NLEVELS=-1,
    FIRSTKEYCARD=-1,
    FIRST2KEYCARD=-1,
    FIRST3KEYCARD=-1,
    FIRST4KEYCARD=-1,
    FULLKEYCARD=-1,
    CLUSTERFACTOR=-1,
    CLUSTERRATIO=-1,
    SEQUENTIAL_PAGES=-1,
    PAGE_FETCH_PAIRS='',
    DENSITY=-1,
    AVERAGE_SEQUENCE_GAP=-1,
    AVERAGE_SEQUENCE_FETCH_GAP=-1,
    AVERAGE_SEQUENCE_PAGES=-1,
    AVERAGE_SEQUENCE_FETCH_PAGES=-1,
    AVERAGE_RANDOM_PAGES=-1,
    AVERAGE_RANDOM_FETCH_PAGES=-1,
    NUMRIDS=-1,
    NUMRIDS_DELETED=-1,
    NUM_EMPTY_LEAFS=-1
WHERE TABNAME = 'ORG' AND TABSCHEMA = 'DB2' ;
...
```

DDL を抽出する `-e` オプションの場合と同様、`-t` および `-z` オプションを使用して、表集合を指定できます。

構成パラメーターと環境変数の抽出

最適化プログラムは、統計、構成パラメーター、レジストリー変数、および環境変数に基づいてプランを選択します。統計の場合と同様に、必要な構成更新ステートメントや構成設定ステートメントも、db2look を使って生成できます。そのためには、-f オプションを使用します。例:

```
c:¥>db2look -d sample -f>config.txt
-- USER is: DB2INST1
-- Binding package automatically ...
-- Bind is successful
-- Binding package automatically ...
-- Bind is successful
```

config.txt には、以下のような出力が含まれます。

```
-- This CLP file was created using DB2LOOK Version 9.1
-- Timestamp: 2/16/2006 7:15:17 PM
-- Database Name: SAMPLE
-- Database Manager Version: DB2/NT Version 9.1.0
-- Database Codepage: 1252
-- Database Collating Sequence is: UNIQUE

CONNECT TO SAMPLE;

-----
-- Database and Database Manager configuration parameters
-----

UPDATE DBM CFG USING cpuspeed 2.991513e-007;
UPDATE DBM CFG USING intra_parallel NO;
UPDATE DBM CFG USING comm_bandwidth 100.000000;
UPDATE DBM CFG USING federated NO;

...

-----
-- Environment Variables settings
-----

COMMIT WORK;

CONNECT RESET;
```

注: DB2 コンパイラーに影響を与えるパラメーターおよび変数だけが組み込まれます。コンパイラーに影響を与えるレジストリー変数がデフォルト値に設定されている場合、それは "Environment Variables settings" の下に表示されません。

システムにインストールされている DB2 製品のリスト表示 (Linux および UNIX)

db2ls コマンドへのシンボリック・リンクを /usr/local/bin ディレクトリー内で使用できるようにするために、少なくとも 1 つの DB2 バージョン 9 製品が root ユーザーによって既にインストールされている必要があります。

DB2 製品では複数のコピーをシステムにインストールしたり、DB2 製品およびファイチャーのインストール先パスを自由に選択したりできるためには、何がどこにインストールされているかを把握しておくためのツールが必要となります。サポート

されている Linux および UNIX オペレーティング・システムでは、db2ls コマンドを実行すると、システムにインストールされている DB2 製品とフィーチャー (DB2 バージョン 9 HTML 文書を含む) がリスト表示されます。

db2ls コマンドを使用して、以下の内容をリストに表示できます。

- DB2 製品がインストールされているシステム内の場所、および DB2 製品レベル
- 特定のインストール・パスにあるすべての、または特定の DB2 製品およびフィーチャー

db2ls コマンドがリスト表示する出力は、使用する ID によって異なります。

- db2ls コマンドを root 権限で実行する場合、ルート DB2 インストールだけが照会されます。
- db2ls コマンドを非ルート (non-root) ID で実行する場合、ルート DB2 インストール、および一致する非ルート ID が所有する非ルート・インストールが照会されます。その他の非ルート (non-root) ID が所有する DB2 インストールは照会されません。

DB2 製品を照会する方法は、db2ls コマンドしかありません。Linux または UNIX オペレーティング・システム固有のユーティリティー (pkginfo、rpm、SMIT、または swlist など) を使用して DB2 製品を照会することはできません。DB2 インストール環境との照会およびインターフェースに使用する既存のスクリプトで、固有のインストール・ユーティリティーを含むものは、変更する必要があります。

Windowsオペレーティング・システムで db2ls コマンドを使用することはできません。

DB2 製品がインストールされているシステム内のパス、および DB2 製品レベルをリスト表示するには、次を実行します。

db2ls

コマンドを実行すると、システムにインストールされている DB2 製品ごとに次の情報がリスト表示されます。

- インストール・パス
- レベル
- フィックスパック
- 特別なインストール番号。この列は、IBM DB2 サポートが使用します。
- インストール日付。この列には、DB2 製品の最終変更日時が表示されます。
- インストーラー UID。この列には、DB2 製品をインストールした UID が表示されます。

特定のインストール・パス内にある DB2 製品またはフィーチャーに関する情報をリストするには、*q* パラメーターを指定する必要があります。

db2ls -q -p -b baseInstallDirectory

ここで、

- *q* は、製品またはフィーチャーを照会することを指定します。このパラメーターは必須です。DB2 バージョン 8 製品を照会すると、空白値が戻されます。

- p は、リスト表示にフィーチャーのリストではなく製品を表示することを指定します。
- b は、製品またはフィーチャーのインストール・ディレクトリーを指定します。インストール・ディレクトリーからコマンドを実行しない場合は、このパラメーターは必須です。

指定するパラメーターに応じて、コマンドは以下の情報をリストします。

- インストール・パス。これは一度だけ指定され、フィーチャーごとにはリストされません。
- 以下の情報が表示されます。
 - インストール済みのフィーチャーの応答ファイル ID、または p オプションが指定されている場合はインストール済みの製品の応答ファイル ID。例えば、ENTERPRISE_SERVER_EDITION。
 - フィーチャー名、または p オプションが指定されている場合は製品名。
 - 製品のバージョン、リリース、修正レベル、フィックスパック・レベル (VRMF)。例えば、9.5.0.0。
 - フィックスパック (該当する場合)。例えば、フィックスパック 1 がインストールされている場合は、表示される値は 1 になります。フィックスパック 1a などの暫定フィックスパックの場合も同様です。
- いずれかの製品の VRMF 情報が一致しない場合は、出力リストの末尾に警告メッセージが表示されます。このメッセージは、フィックスパックを適用するよう指示します。

db2pd を使用したモニターおよびトラブルシューティング

db2pd ツールは、トラブルシューティングのために使用します。このツールによって、DB2 メモリー・セットからの情報を素早く即時に返すことが可能です。

このツールは、ラッチを獲得したりエンジン・リソースを使用したりせずに情報を収集します。このため、db2pd が情報を収集している間に、取得対象の情報の内容が変わる可能性があります。つまり、データは完全には正確でないかもしれません (この可能性を考慮に入れる必要があります)。変更メモリー・ポインターが見つかった場合、db2pd が異常終了するのを防ぐために、シグナル・ハンドラーが使われます。その結果、例えば「Changing data structure forced command termination (データ構造が変更されたためコマンドが強制終了されました)」というメッセージが出力に含まれる可能性があります。この点を考慮に入れて使用すれば、このツールはトラブルシューティングに役立ちます。ラッチなしで情報を収集することには 2 つの利点があります (より高速な検索、エンジン・リソースとの競合がない)。

特定の SQLCODE、ZRC コードまたは ECF コードの発生時にデータベース管理システムに関する情報をキャプチャーする場合は db2pdcfg -catch コマンドを使用します。エラーをキャッチすると、db2cos (コールアウト・スクリプト) が起動します。任意の db2pd コマンド、オペレーティング・システム・コマンド、または問題解決のために必要なその他のコマンドを実行するために、db2cos ファイルを動的に変更することができます。テンプレート db2cos ファイルは、UNIX および Linux では、`sqllib/bin` にあります。Windows オペレーティング・システムの場合、db2cos は `$DB2PATH\bin` ディレクトリーにあります。

以下は、トラブルシューティングを円滑にする db2pd の使用例を集めたものです。

シナリオ 1: ロック待機の診断

db2pd -db <データベース名> -locks -transactions -applications -dynamic コマンドを使用すると、以下のような結果が得られます。

```
Locks:
Address          TranHdl Lockname                                     Type Mode Sts Owner Dur HldCnt Att ReleaseFlg
0x07800000202E5238 3      0002000200000040000000052 Row  ..X G  3    1  0    0x0000 0x40000000
0x07800000202E4668 2      0002000200000040000000052 Row  ..X W* 2    1  0    0x0000 0x40000000
```

-db データベース名オプションを使用して指定したデータベースの場合、最初の結果はそのデータベースのロックを示します。TranHdl 2 が、TranHdl 3 によって保持されているロックを待機していることがわかります。

```
Transactions:
Address          AppHandl [nod-index] TranHdl Locks State Tflag      Tflag2      Firstltn      Lastltn      LogSpace SpaceReserved TID              AxRegCnt GXID
0x0780000020251880 11      [000-00011] 2      4  READ  0x00000000 0x00000000 0x000000000000 0x000000000000 0 0              0x000000000087 1 0
0x0780000020252900 12      [000-00012] 3      4  WRITE 0x00000000 0x00000000 0x000000FA000C 0x000000FA000C 113 154            0x000000000088 1 0
```

TranHdl 2 が AppHandl 11 に関連付けられていて、TranHdl 3 が AppHandl 12 に関連付けられていることがわかります。

```
Applications:
Address          AppHandl [nod-index] NumAgents CoordPid Status      C-AnchID C-StmtUID L-AnchID L-StmtUID Appid
0x0780000006879E0 12      [000-00012] 1      1073336 UOW-Waiting 0      0      17      1      *LOCAL.burford.060303225602
0x078000000685E80 11      [000-00011] 1      1040570 UOW-Executing 17     1      94      1      *LOCAL.burford.060303225601
```

AppHandl 12 は、動的ステートメント 17, 1 を最後に実行したのがわかります。AppHandl 11 は、動的ステートメント 17, 1 を現在実行中で、最後に実行ステートメント 94, 1 を実行しました。

```
Dynamic SQL Statements:
Address          AnchID StmtUID NumEnv NumVar NumRef NumExe Text
0x07800000209FD800 17     1      1      1      2      2      update pdtest set c1 = 5
0x07800000209FCCC0 94     1      1      1      2      2      set lock mode to wait 1
```

テキスト列は、ロック・タイムアウトに関連している SQL ステートメントを示しています。

シナリオ 2: 待機中のすべてのロックをキャプチャーするための -wlocks オプションの使用

以下の出力例のアプリケーション 1 (AppHandl 47) は挿入を実行し、アプリケーション 2 (AppHandl 46) がその表で選択を行っています。

```
venus@boson:/home/venus =>db2pd -wlocks -db pdtest

Database Partition 0 -- Database PDTEST -- Active -- Up 0 days 00:01:22

Locks being waited on :
AppHandl [nod-index] TranHdl Lockname                                     Type Mode Conv Sts CoordEDU  AppName  AuthID  AppID
47      [000-00047] 8      0002000400000000840000652 Row  ..X  G  5160    db2bp   VENUS   *LOCAL.venus.071207213730
46      [000-00046] 2      0002000400000000840000652 Row  .NS  W  5913    db2bp   VENUS   *LOCAL.venus.071207213658
```

シナリオ 3: ロック所有者およびロック待機者に関する詳細な実行時情報をキャプチャーするための -apinfo オプションの使用

以下の出力例は、前述のシナリオ 2 と同じ条件でキャプチャーされています。

```
venus@boson:/home/venus =>db2pd -apinfo 47 -db pdtest

Database Partition 0 -- Database PDTEST -- Active -- Up 0 days 00:01:30

Application :
```

```
Address : 0x0780000001676480
AppHandl [nod-index] : 47 [000-00047]
Application PID : 876558
Application Node Name : boson
IP Address: n/a
Connection Start Time : (1197063450)Fri Dec 7 16:37:30 2007
Client User ID : venus
System Auth ID : VENUS
Coordinator EDU ID : 5160
Coordinator Partition : 0
Number of Agents : 1
Locks timeout value : 4294967294 seconds
Locks Escalation : No
Workload ID : 1
Workload Occurrence ID : 2
Trusted Context : n/a
Connection Trust Type : non trusted
Role Inherited : n/a
Application Status : UOW-Waiting
Application Name : db2bp
Application ID : *LOCAL.venus.071207213730
```

List of inactive statements of current UOW :

```
UOW-ID : 2
Activity ID : 1
Package Schema : NULLID
Package Name : SQLC2G13
Package Version :
Section Number : 203
SQL Type : Dynamic
Isolation : CS
Statement Type : DML, Insert/Update/Delete
Statement : insert into pdtest values 99
```

```
venus@boson:/home/venus =>db2pd -apinfo 46 -db pdtest
```

```
Database Partition 0 -- Database PDTEST -- Active -- Up 0 days 00:01:39
```

Application :

```
Address : 0x0780000000D77A60
AppHandl [nod-index] : 46 [000-00046]
Application PID : 881102
Application Node Name : boson
IP Address: n/a
Connection Start Time : (1197063418)Fri Dec 7 16:36:58 2007
Client User ID : venus
System Auth ID : VENUS
Coordinator EDU ID : 5913
Coordinator Partition : 0
Number of Agents : 1
Locks timeout value : 4294967294 seconds
Locks Escalation : No
Workload ID : 1
Workload Occurrence ID : 1
Trusted Context : n/a
Connection Trust Type : non trusted
Role Inherited : n/a
Application Status : Lock-wait
Application Name : db2bp
Application ID : *LOCAL.venus.071207213658
```

List of active statements :

```
*UOW-ID : 3
Activity ID : 1
Package Schema : NULLID
Package Name : SQLC2G13
```



```

Package Version :
Section Number : 201
SQL Type :      Dynamic
Isolation :     CS
Statement Type : DML, Select (blockable)
Statement :     select * from pdtest

```

シナリオ 4: ロッキング問題を検討するときのコールアウト・スクリプトの使用

db2cos 出力ファイルを見つけます。ファイルのロケーションは、データベース・マネージャー構成パラメーター `DIAGPATH` によって制御されます。出力ファイルの内容は、db2cos ファイルに入力するコマンドによって異なります。以下は、db2cos ファイルに `db2pd -db sample -locks` コマンドが含まれている場合に提供される出力の例です。

```

Lock Timeout Caught
Thu Feb 17 01:40:04 EST 2006
Instance DB2
Database: SAMPLE
Partition Number: 0
PID: 940
TID: 2136
Function: sqlplnfd
Component: lock manager
Probe: 999
Timestamp: 2006-02-17-01.40.04.106000
AppID: *LOCAL.DB2...
AppHdl:
...
Database Partition 0 -- Database SAMPLE -- Active -- Up 0 days 00:06:53
Locks:
Address      TranHdl Lockname                                     Type Mode Sts Owner Dur HldCnt Att Rlse
0x402C6B30 3          00020003000000040000000052 Row  ..X W* 3    1  0    0  0x40

```

この中で、`W*` はタイムアウトになったロックです。ロック・タイムアウトは、ロックがより高いモードに変換されるときにも発生します。その場合、出力結果には、『`W*`』ではなく、『`C*`』が付きます。ただし、この場合も、ロック待機が発生しています。db2cos ファイルにある他の `db2pd` コマンドによって提供される出力を参照しながら、この出力結果をトランザクション、アプリケーション、エージェント、SQL ステートメントにマップすることが可能です。出力の特定部分を絞り込んだり、他のコマンドを使用することによって、必要な情報を収集できます。例えば、待機状況のロックだけを出力するために、`-locks wait` オプションを使用するよう `db2pd` コマンド・オプションを変更できます。必要に応じて、`-app` オプションや `-agent` オプションを含めることも可能です。

シナリオ 5: アプリケーションと動的 SQL ステートメントのマップ

コマンド `db2pd -applications` は、動的 SQL ステートメントの現行および最新アンカー ID と、ステートメント・ユニーク ID を報告します。これにより、アプリケーションから動的 SQL ステートメントへの直接的なマッピングが可能になります。

```

db2pd -app -dyn

Applications:
Address      AppHndl [nod-index] NumAgents  CoorPid  Status
0x00000002006D2120 780      [000-00780] 1          10615    UOW-Executing

C-AnchID C-StmtUID  L-AnchID L-StmtUID  Appid
163      1          110      1          *LOCAL.burford.050202200412

```

Dynamic SQL Statements:

Address	AnchID	StmtUID	NumEnv	NumVar	NumRef	NumExe	Text
0x0000000220A02760	163	1	2	2	2	1	CREATE VIEW MYVIEW
0x0000000220A0B460	110	1	2	2	2	1	CREATE VIEW YOURVIEW

シナリオ 6: メモリー使用量のモニター

メモリー使用量を知りたい場合は db2pd -memblock コマンドが役立ちます。例:

All memory blocks in DBMS set.

Address	PoolID	PoolName	BlockAge	Size(Bytes)	I	LOC	File
0x0780000000740068	62	resynch	2	112	1	1746	1583816485
0x0780000000725688	62	resynch	1	108864	1	127	1599127346
0x07800000001F4348	57	ostrack	6	5160048	1	3047	698130716
0x07800000001B5608	57	ostrack	5	240048	1	3034	698130716
0x07800000001A0068	57	ostrack	1	80	1	2970	698130716
0x07800000001A00E8	57	ostrack	2	240	1	2983	698130716
0x07800000001A0208	57	ostrack	3	80	1	2999	698130716
0x07800000001A0288	57	ostrack	4	80	1	3009	698130716
0x0780000000700068	70	apmh	1	360	1	1024	3878879032
0x07800000007001E8	70	apmh	2	48	1	914	1937674139
0x0780000000700248	70	apmh	3	32	1	1000	1937674139
...							

この後に、ソートされた「プールごとの」出力が続きます。

Memory blocks sorted by size for ostrack pool:

PoolID	PoolName	TotalSize(Bytes)	TotalCount	LOC	File
57	ostrack	5160048	1	3047	698130716
57	ostrack	240048	1	3034	698130716
57	ostrack	240	1	2983	698130716
57	ostrack	80	1	2999	698130716
57	ostrack	80	1	2970	698130716
57	ostrack	80	1	3009	698130716

Total size for ostrack pool: 5400576 bytes

Memory blocks sorted by size for apmh pool:

PoolID	PoolName	TotalSize(Bytes)	TotalCount	LOC	File
70	apmh	40200	2	121	2986298236
70	apmh	10016	1	308	1586829889
70	apmh	6096	2	4014	1312473490
70	apmh	2516	1	294	1586829889
70	apmh	496	1	2192	1953793439
70	apmh	360	1	1024	3878879032
70	apmh	176	1	1608	1953793439
70	apmh	152	1	2623	1583816485
70	apmh	48	1	914	1937674139
70	apmh	32	1	1000	1937674139

Total size for apmh pool: 60092 bytes

...

出力の最終セクションでは、全セットについて、メモリー・コンシューマーをソートします。

All memory consumers in DBMS memory set:

PoolID	PoolName	TotalSize(Bytes)	%Bytes	TotalCount	%Count	LOC	File
57	ostrack	5160048	71.90	1	0.07	3047	698130716
50	sqlch	778496	10.85	1	0.07	202	2576467555
50	sqlch	271784	3.79	1	0.07	260	2576467555
57	ostrack	240048	3.34	1	0.07	3034	698130716
50	sqlch	144464	2.01	1	0.07	217	2576467555
62	resynch	108864	1.52	1	0.07	127	1599127346
72	eduah	108048	1.51	1	0.07	174	4210081592
69	krdbh	73640	1.03	5	0.36	547	4210081592
50	sqlch	43752	0.61	1	0.07	274	2576467555
70	apmh	40200	0.56	2	0.14	121	2986298236
69	krdbh	32992	0.46	1	0.07	838	698130716
50	sqlch	31000	0.43	31	2.20	633	3966224537

```

50      sqlch      25456                0.35  31      2.20  930  3966224537
52      kerh       15376                0.21   1      0.07  157  1193352763
50      sqlch      14697                0.20   1      0.07  345  2576467555
...

```

UNIX および Linux では、専用メモリーのメモリー・ブロックを報告することもできます。例:

```
db2pd -memb pid=159770
```

All memory blocks in Private set.

```

Address          PoolID   PoolName  BlockAge  Size(Bytes) I LOC  File
0x0000000110469068 88      private  1         2488      1 172  4283993058
0x0000000110469A48 88      private  2         1608      1 172  4283993058
0x000000011046A0A8 88      private  3         4928      1 172  4283993058
0x000000011046B408 88      private  4         7336      1 172  4283993058
0x000000011046D0C8 88      private  5          32       1 172  4283993058
0x000000011046D108 88      private  6         6728      1 172  4283993058
0x000000011046EB68 88      private  7          168     1 172  4283993058
0x000000011046EC28 88      private  8          24       1 172  4283993058
0x000000011046EC68 88      private  9          408     1 172  4283993058
0x000000011046EE28 88      private 10         1072     1 172  4283993058
0x000000011046F288 88      private 11         3464     1 172  4283993058
0x0000000110470028 88      private 12          80      1 172  4283993058
0x00000001104700A8 88      private 13          480     1 1534 862348285
0x00000001104702A8 88      private 14          480     1 1939 862348285
0x0000000110499FA8 88      private 80        65551    1 1779 4231792244
Total set size: 94847 bytes

```

Memory blocks sorted by size:

```

PoolID   PoolName  TotalSize(Bytes)  TotalCount  LOC  File
88      private  65551             1           1779 4231792244
88      private  28336             12          172  4283993058
88      private  480               1           1939 862348285
88      private  480               1           1534 862348285
Total set size: 94847 bytes

```

シナリオ 7: どのアプリケーションが表スペースを消費しているかの確認

db2pd -tcbstats を使用すれば、表に対する挿入の数を識別することができます。以下は、TEMP1 という名前のユーザー定義グローバル一時表のサンプル情報です。

```

TCB Table Information:
Address          TbspaceID  TableID  PartID  MasterTbs  MasterTab  TableName  SchemaNm  ObjClass  DataSize  LfSize  LobSize  XMLSize
0x0780000020B62AB0 3          2        n/a     3          2          TEMP1     SESSION  Temp      966      0        0        0

TCB Table Stats:
Address          TableName  Scans  UDI  PgReorgs  NoChgUpdts  Reads  FscrUpdates  Inserts  Updates  Deletes  OvFlReads  OvFlCrtes
0x0780000020B62AB0 TEMP1      0      0      0          0            0      43968        0        0        0        0

```

その後、db2pd -tablespaces コマンドを使用することにより、表スペース 3 の情報を入手できます。以下はその出力例です。

```

Tablespace 3 Configuration:
Address          Type  Content  PageSz  ExtentSz  Auto  Prefetch  BufID  BufIDDisk  FSC  NumCntrs  MaxStripe  LastConsecPg  Name
0x0780000020B1B5A0 DMS  UsrTmp  4096    32        Yes  32        1      1          On  1        0          31          TEMPSPACE2

Tablespace 3 Statistics:
Address          TotalPgs  UsablePgs  UsedPgs  PndFreePgs  FreePgs  HWM  State  MinRecTime  NQuiescers
0x0780000020B1B5A0 5000      4960      1088     0            3872    1088  0x00000000 0          0

Tablespace 3 Autoresize Statistics:
Address          AS  AR  InitSize  IncSize  IIP  MaxSize  LastResize  LRF
0x0780000020B1B5A0 No  No  0          0        No  0        None        No

Containers:
Address          ContainNum  Type  TotalPgs  UseablePgs  StripeSet  Container
0x0780000020B1DCC0 0          File  5000      4960        0          /home/db2inst1/tempspace2a

```

FreePgs 列を参照することにより、スペースが埋められつつあることが分かります。フリー・ページの値が減るほど、使用可能なスペースは少なくなります。FreePgs の値と UsedPgs の値の和が UsablePgs の値と等しくなっていることにも注目してください。

それが分かれば、表 TEMP1 を使用している動的 SQL ステートメントを識別できます。

```

db2pd -db sample -dyn
Database Partition 0 -- Database SAMPLE -- Active -- Up 0 days 00:13:06

Dynamic Cache:
Current Memory Used      1022197
Total Heap Size         1271398
Cache Overflow Flag     0
Number of References    237
Number of Statement Inserts 32
Number of Statement Deletes 13
Number of Variation Inserts 21
Number of Statements    19

Dynamic SQL Statements:
Address      AnchID StmtUID NumEnv NumVar NumRef NumExe Text
0x0000000220A08C40 78 1 2 2 3 2 declare global temporary table temp1 (c1 char(6)) not logged
0x0000000220A08D960 253 1 1 1 24 24 insert into session.temp1 values('TEST')

```

最後に、これを db2pd -app 出力と付き合わせることで、アプリケーションを識別することができます。

```

Applications:
Address      AppHandl [nod-index] NumAgents  CoorPid Status
0x0000000200661840 501 [000-00501] 1 11246 UOW-Waiting

C-AnchID C-StmtUID L-AnchID L-StmtUID Appid
0 0 253 1 *LOCAL.db2inst1.050202160426

```

db2pd を以前に使用したときの動的 SQL ステートメントの要求の結果であるアンカー ID (AnchID) 値は、関連したアプリケーションの要求と共に使用されます。アプリケーションの結果は、最後のアンカー ID (L-AnchID) 値がアンカー ID (AnchID) 値と同じであることを示しています。db2pd の 1 回の実行の結果が次の db2pd の実行で使用されます。

db2pd -agent からの出力は、アプリケーションによって読み込まれた行数 (Rowsread 列) と書き込まれた行数 (Rowswrtn 列) を示しています。これらの値によって、アプリケーションが何を完了し、何をまだ完了していないかがわかります。

```

Address      AppHandl [nod-index] AgentPid Priority Type DBName
0x0000000200698080 501 [000-00501] 11246 0 Coord SAMPLE

State      ClientPid Userid ClientNm Rowsread Rowswrtn LkTmOt
Inst-Active 26377 db2inst1 db2bp 22 9588 NotSet

```

db2pd -agent 要求からの AppHandl と AgentPid の値を、db2pd -app 要求からの AppHandl と CoorPiid の対応値に逆にマップできます。

内部一時表が表スペースを埋めている可能性があれば、これらのステップは若干異なります。再び db2pd -tcbstats を使用して、多数の挿入を持つ表を識別します。以下は、暗黙的一時表のサンプル情報です。

```

TCB Table Information:
Address      TbspaceID TableID PartID MasterTbs MasterTab TableName SchemaNm ObjClass DataSize ...
0x0780000020CC0D30 1 2 n/a 1 2 TEMP (00001,00002) <30> <JMC Temp 2470 ...
0x0780000020CC14B0 1 3 n/a 1 3 TEMP (00001,00003) <31> <JMC Temp 2367 ...
0x0780000020CC21B0 1 4 n/a 1 4 TEMP (00001,00004) <30> <JMC Temp 1872 ...

TCB Table Stats:
Address      TableName Scans UDI PgReorgs NoChgUpdts Reads FscrUpdates Inserts ...
0x0780000020CC0D30 TEMP (00001,00002) 0 0 0 0 0 0 43219 ...
0x0780000020CC14B0 TEMP (00001,00003) 0 0 0 0 0 0 42485 ...
0x0780000020CC21B0 TEMP (00001,00004) 0 0 0 0 0 0 0 ...

```

この例では、命名規則「TEMP (TbspaceID, TableID)」を持つ表に多数の挿入があります。これらは暗黙的一時表です。SchemaNm 列の値には、AppHandl の値が SchemaNm の値と連結するという命名規則があります。これにより、操作を実行しているアプリケーションを識別できます。

その後、その情報を `db2pd -tablespaces` からの出力にマップし、表スペース 1 で使用されるスペースを確認できます。表スペース統計内の `UsedPgs` を `UsablePgs` との関連で注目してください。

```
Tablespace Configuration:
Address      Id      Type Content PageSz ExtentSz Auto Prefetch BufID BufIDDisk FSC NumCtrs MaxStripe LastConsecPg Name
0x07800000203FB5A0 1      SMS SysTmp 4096 32      Yes 320 1 1      On 10 0 31      TEMPSPACE1

Tablespace Statistics:
Address      Id      TotalPgs UsablePgs UsedPgs PndFreePgs FreePgs HWM State MinRecTime NQuiescers
0x07800000203FB5A0 1      6516 6516 6516 0 0 0 0x00000000 0 0

Tablespace Autoresize Statistics:
Address      Id      AS AR InitSize IncSize IIP MaxSize LastResize LRF
0x07800000203FB5A0 1      No No 0 0 No 0 None No

Containers:
...
```

その後、コマンド `db2pd -app` を使用して、アプリケーション・ハンドル 30 および 31 を識別することができます (これらは `-tcbstats` 出力で確認されたものです)。

```
Applications:
Address      AppHandl [nod-index] NumAgents CoorPid Status C-AnchID C-StmtUID L-AnchID L-StmtUID Appid
0x07800000006FB880 31 [000-00031] 1 4784182 UOW-Waiting 0 0 107 1 *LOCAL.db2inst1.051215214142
0x07800000006F9CE0 30 [000-00030] 1 8966270 UOW-Executing 107 1 107 1 *LOCAL.db2inst1.051215214013
```

最後に、`db2pd -dyn` コマンドを使ってこれを動的 SQL にマップします。

```
Dynamic SQL Statements:
Address      AnchID StmtUID NumEnv NumVar NumRef NumExe Text
0x0780000020B296C0 107 1 1 1 43 43 select c1, c2 from test group by c1,c2
```

シナリオ 8: リカバリーのモニター

コマンド `db2pd -recovery` は、リカバリーの進行状況を確認するために使用できるいくつかのカウンターを表示します。「Current Log (現在のログ)」および「Current LSN (現在の LSN)」はログの位置を示します。「CompletedWork」は、完了済みのバイト数のカウントです。

```
Recovery:
Recovery Status 0x00000401
Current Log S0000005.LOG
Current LSN 000002551BEA
Job Type ROLLFORWARD RECOVERY
Job ID 7
Job Start Time (1107380474) Wed Feb 2 16:41:14 2005
Job Description Database Rollforward Recovery
Invoker Type User
Total Phases 2
Current Phase 1

Progress:
Address      PhaseNum Description StartTime CompletedWork TotalWork
0x0000000200667160 1 Forward Wed Feb 2 16:41:14 2005 2268098 bytes Unknown
0x0000000200667258 2 Backward NotStarted 0 bytes Unknown
```

シナリオ 9: トランザクションによって使用されているリソースの量の確認

コマンド `db2pd -transactions` は、ロックの数、最初のログ・シーケンス番号 (LSN)、最後の LSN、使用ログ・スペース、および予約済みスペースに関する情報を提供します。この情報はトランザクションの動作を理解するうえで役立ちます。

```
Transactions:
Address      AppHandl [nod-index] TranHdl Locks State Tflag
0x000000022026D980 797 [000-00797] 2 108 WRITE 0x00000000
0x000000022026E600 806 [000-00806] 3 157 WRITE 0x00000000
0x000000022026F280 807 [000-00807] 4 90 WRITE 0x00000000

Tflag2 Firstlsn Lastlsn LogSpace SpaceReserved
0x00000000 0x000001072262 0x0000010B2C8C 4518 95450
```

```
0x00000000 0x000001057574 0x0000010B3340 6576      139670
0x00000000 0x00000107CF0C 0x0000010B2FDE 3762      79266
```

```
TID           AxRegCnt    GXID
0x000000000451 1           0
0x0000000003E0 1           0
0x000000000472 1           0
```

シナリオ 10: ログの使用状況のモニター

db2pd -logs コマンドは、データベースのログ使用状況をモニターするのに役立ちます。出力の「Pages Written (書き込み済みページ数)」を見れば、ログの使用が進行しているかどうか判別できます。

```
Logs:
Current Log Number      2
Pages Written           846
Method 1 Archive Status Success
Method 1 Next Log to Archive 2
Method 1 First Failure  n/a
Method 2 Archive Status Success
Method 2 Next Log to Archive 2
Method 2 First Failure  n/a

Address      StartLSN      State      Size  Pages  Filename
0x0000000023001BF58 0x000001B58000 0x00000000 1000  1000  S0000002.LOG
0x0000000023001BE98 0x000001F40000 0x00000000 1000  1000  S0000003.LOG
0x00000000230008F58 0x000002328000 0x00000000 1000  1000  S0000004.LOG
```

この出力から、2つの問題を識別することができます。

1. アーカイブに問題がある場合、「Archive Status (アーカイブ状況)」が値「Failure (失敗)」に設定されます。これは、最新のログのアーカイブが失敗したことを示します。あるいは、アーカイブ失敗が継続しているためにログがアーカイブされない場合には、「First Failure (最初に失敗)」に設定されます。
2. ログのアーカイブ処理が非常に遅い場合には、「Next Log to Archive (次のアーカイブ対象ログ)」の値が「Current Log Number (現在のログ番号)」の値より小さくなります。このために、ログ・パスがいっぱいになる可能性があります。ログ・パスが完全にいっぱいになると、データベース内のデータはまったく変更されなくなる可能性があります。

シナリオ 11: SYSPLEX リストの表示

db2pd -sysplex コマンドを使用しない場合、SYSPLEX リストを報告する唯一の方法は、DB2 トレースを介する方法です。

```
Sysplex List:
Alias:          HOST
Location Name:  HOST1
Count:          1

IP Address      Port      Priority  Connections  Status  PRDID
1.2.34.56      400      1         0             0
```

シナリオ 12: スタック・トレースの生成

Windows オペレーティング・システムの場合は db2pd -stack all コマンド (UNIX オペレーティング・システムの場合は -stack) を使用すれば、現在のデータベース・

パーティションにあるすべてのプロセスのスタック・トレースを生成できます。プロセスやスレッドがループ状態または停止状態にあると疑われる場合には、このコマンドを反復して使用できます。

`db2pd -stack <eduid>` コマンドを実行して、特定のエンジン・ディスパッチ可能単位 (EDU) の現在の呼び出しスタックを取得できます。例:

```
db2pd -stack 137
```

```
Attempting to dump stack trace for eduid 137.  
See current DIAGPATH for trapfile.
```

DB2 プロセスに対するすべての呼び出しスタックを確認するには、`db2pd -stack all` コマンドを使用します。例えば、Windows オペレーティング・システムであれば、以下のようにします。

```
db2pd -stack all
```

```
Attempting to dump all stack traces for instance.  
See current DIAGPATH for trapfiles.
```

複数の物理ノードのあるパーティション・データベース環境では、コマンド `db2_all` "; `db2pd -stack all`" を使用することにより、すべてのパーティションから情報を取得できます。しかし、同じマシン上の複数の論理パーティションだけから成る環境では、`db2pd -alldbp -stacks` を使用した方が速く動作します。

シナリオ 13: データベース・パーティションのメモリー統計の表示

`db2pd -dbptnmem` コマンドは、DB2 サーバーが現在使用しているメモリーの量を表示します。また、大きな区切りで、サーバーのどの領域がそのメモリーを使用しているかを表示します。

AIX マシン上での `db2pd -dbptnmem` からの出力例を以下に示します。

Database Partition Memory Controller Statistics

```
Controller Automatic: Y  
Memory Limit:      122931408 KB  
Current usage:     651008 KB  
HWM usage:         651008 KB  
Cached memory:    231296 KB
```

これらのデータ・フィールドおよび列に関する説明を以下に示します。

- 「Controller Automatic (自動コントローラー)」: **instance_memory** 構成パラメーターが **AUTOMATIC** に設定されている場合、**Y** になります。これは、データベース・マネージャーが自動的にメモリー使用量の上限を決定することを意味します。
- 「Memory Limit (メモリーの限度)」: DB2 サーバーの消費できるメモリーの上限。これは、**instance_memory** 構成パラメーターの値です。
- 「Current usage (現在の使用量)」: サーバーが現在使用しているメモリーの量。
- 「HWM usage (HWM 使用量)」: データベース・パーティションがアクティブ化されたとき (`db2start` コマンドの実行時) 以来消費された、最高水準点 (HWM)、つまりピーク時のメモリー使用量。

- 「Cached memory (キャッシュ済みメモリー)」: 現在の使用量のうちどの程度の量が、現在使用中でないものの、パフォーマンス上の理由で今後のメモリー要求のためにキャッシュに入れられているか。

AIX 上での `db2pd -dbptnmem` からの出力例の続きを以下に示します。

```
Individual Memory Consumers:
Name                Mem Used (KB)  HWM Used (KB)  Cached (KB)
=====
APPL-DBONE          160000         160000         159616
DBMS-name            38528          38528          3776
FMP_RESOURCES        22528          22528           0
PRIVATE              13120          13120           740
FCM_RESOURCES        10048          10048           0
LCL-p606416          128            128             0
DB-DBONE             406656         406656         67200
```

DB2 サーバー内のすべての登録済みのメモリーの「コンシューマー」が、使用しているメモリーの合計量とともにリストされます。列の説明は以下のとおりです。

- 「Name (名前)」: メモリーの「コンシューマー」の簡潔な識別名。例として、以下のようなものが含まれます。
 - APPL-<dbname>。データベース <dbname> について消費されるアプリケーション・メモリー。
 - DBMS-xxx。グローバル・データベース・マネージャーのメモリー所要量。
 - FMP_RESOURCES。db2fmps との通信に必要なメモリー。
 - PRIVATE。各種の専用メモリー所要量。
 - FCM_RESOURCES。高速コミュニケーション・マネージャーのリソース。
 - LCL-<pid>。ローカル・アプリケーションとの通信に使用されるメモリー・セグメント。
 - DB-<dbname>。データベース <dbname> について消費されるデータベース・メモリー。
- 「Mem Used (KB) (使用されているメモリー (KB))」: そのコンシューマーに現在割り当てられているメモリーの量。
- 「HWM Used (KB) (使用された HWM (KB))」: コンシューマーが使用した最高水準点、あるいはピーク時のメモリー。
- 「Cached (KB) (キャッシュ済み (KB))」: 「Mem Used (KB) (使用されているメモリー (KB))」のうち、現在使用中ではないものの、今後のメモリー割り振りのためにすぐに使用可能なメモリーの量。

db2support を使用した環境情報の収集

DB2 の問題に関する情報を集めるとき、実行すべき最も重要な DB2 ユーティリティーは `db2support` です。 `db2support` ユーティリティーは、DB2 とシステムのあらゆる診断情報を自動的に収集します。さらに、オプションとして、問題の状況について問い合わせる「質問と回答」形式の対話も可能です。

`db2support` ユーティリティーを使用すると `GET DATABASE CONFIGURATION FOR <database name>` または `LIST TABLESPACES SHOW DETAIL` のようなコマンドを手動で入力する必要がないため、ユーザー・エラーを未然に防ぐことができ

ます。さらに、どの特定のコマンドを実行するか、どのファイルを収集するかを指示する必要がないため、データの収集をより短時間で行うことができます。

- コマンド `db2support -h` を実行して、コマンド・オプションの詳細なリストを表示します。
- 適切な `db2support` コマンドを使用してデータを収集します。

`db2support` ユーティリティーがエラーなしに必要なすべての情報を収集するためには、`SYSADM` 権限を持つユーザー (例えばインスタンス所有者) によって実行されなければなりません。`SYSADM` 権限を持たないユーザーが `db2support` ユーティリティーを実行した場合、`QUERY CLIENT` や `LIST ACTIVE DATABASES` といったコマンドが実行される結果として、SQL エラー (例えば `SQL1092N`) が発生するかもしれません。

IBM サポートに情報を送るために `db2support` ユーティリティーを使用する場合には、問題がシステムで発生している最中に `db2support` コマンドを実行してください。そのようにして、ツールはタイムリーな情報 (オペレーティング・システムのパフォーマンスの詳細など) を収集します。問題発生時にユーティリティーを実行できない場合でも、First Occurrence Data Capture (FODC) 診断ファイルが自動的に生成されるので、問題が停止した後に `db2support` コマンドを実行できます。

ほとんどの場合、問題をデバッグするためには、以下のような基本的オプションを指定して起動すれば、必要な情報が収集されます (なお、`-c` オプションを使用すると、ユーティリティーはデータベースへの接続を確立します)。

```
db2support <output path> -d <database name> -c
```

出力は適切に編成されて ZIP アーカイブ `db2support.zip` に圧縮されるため、任意のシステムに転送して簡単に解凍することができます。

`db2support` がどのような情報をキャプチャーするかは、コマンドを呼び出す方法、データベース・マネージャーが開始済みかどうか、およびデータベースに接続できるかどうかによって異なります。

すべての状況において、`db2support` ユーティリティーは以下の情報を収集します。

- `db2diag.log`
- すべてのトラップ・ファイル
- ロック・リスト・ファイル
- ダンプ・ファイル
- さまざまなシステム関連ファイル
- さまざまなシステム・コマンドからの出力
- `db2cli.ini`

さらに、状況に応じて、`db2support` ユーティリティーは以下の情報も収集することがあります。

- アクティブ・ログ・ファイル
- バッファ・プールと表スペース (`SQLSPCS.1`、`SQLSPCS.2`) の制御ファイル (`-d` オプションを使用した場合)

- db2dump ディレクトリーの内容
- より詳しいシステム情報 (-s オプションを使用した場合)
- データベース構成の設定値 (-d オプションを使用した場合)
- データベース・マネージャーの構成設定ファイル
- ログ・ファイルのヘッダー・ファイル (-d オプションを使用した場合)
- リカバリー履歴ファイル (-d オプションを使用した場合)

HTML 形式のレポート db2support.html には、常に以下の情報が含まれます。

- 問題記録 (PMR) 番号 (-n を指定した場合)
- オペレーティング・システムおよびレベル (例えば AIX 5.1)
- DB2 リリース情報
- 32 ビット環境かそれとも 64 ビット環境かの標識
- DB2 インストール・パス情報
- db2nodes.cfg の内容
- CPU の数、ディスクの数、メモリー容量
- インスタンスにあるデータベースのリスト
- レジストリー情報と環境 (PATH および LIBPATH を含む)
- 現在のファイル・システムのディスク空きスペース、および UNIX の i ノード
- Java SDK のレベル
- データベース・マネージャー構成
- データベース・リカバリー履歴ファイルのリスト
- sqllib ディレクトリーの ls -lR 出力 (または Windows でこれに相当するもの)
- LIST NODE DIRECTORY コマンドの結果
- LIST ADMIN NODE DIRECTORY コマンドの結果
- LIST DCS DIRECTORY コマンドの結果
- LIST DCS APPLICATIONS EXTENDED コマンドの結果
- すべてのインストール済みソフトウェアのリスト

-s オプションを指定した場合には、以下の情報が db2support.html ファイルに表示されます。

- 詳細なディスク情報 (パーティションのレイアウト、種類、LVM 情報など)
- 詳細なネットワーク情報
- カーネル統計
- ファームウェアのバージョン
- その他のオペレーティング・システム固有のコマンド

DB2 が開始済みの場合には、以下の追加情報が db2support.html ファイルに含まれます。

- クライアント接続の状態
- データベース構成とデータベース・マネージャー構成 (データベース構成を含めるには -d オプションが必要です)
- CLI の構成

- メモリー・プールの情報 (サイズと使用量)。-d オプションを使用すると、詳細なデータが収集されます。
 - LIST ACTIVE DATABASES コマンドの結果
 - LIST DCS APPLICATIONS コマンドの結果
- c オプションを指定し、データベース接続が正常に確立されている場合には、以下の情報が db2support.html ファイルに含まれます。
- ユーザー表の数
 - データベース・データのおおよそのサイズ
 - データベースのスナップショット
 - アプリケーションのスナップショット
 - バッファ・プール情報
 - LIST APPLICATIONS コマンドの結果
 - LIST COMMAND OPTIONS コマンドの結果
 - LIST DATABASE DIRECTORY コマンドの結果
 - LIST INDOUBT TRANSACTIONS コマンドの結果
 - LIST DATABASE PARTITION GROUPS コマンドの結果
 - LIST DBPARTITIONNUMS コマンドの結果
 - LIST ODBC DATA SOURCES コマンドの結果
 - LIST PACKAGES/TABLES コマンドの結果
 - LIST TABLESPACE CONTAINERS コマンドの結果
 - LIST TABLESPACES コマンドの結果
 - LIST DRDA® IN DOUBT TRANSACTIONS コマンドの結果

基本的なトレース診断

DB2 で問題が繰り返し起こる場合、トレースによって、追加情報を収集できる場合があります。通常の状態では、DB2 お客様サポートに依頼された場合にのみ、トレースを使用してください。トレースを取る処理には、トレース機能の設定、問題の再現、およびデータの収集が必要になります。

トレースによって集められる情報量は、急速に増加します。トレースを取る場合はエラーの状態のみを収集し、できる限りその他のアクティビティを避けてください。トレースを取るときは、問題の再現に、最小のシナリオを使用してください。

トレースを収集すると、DB2 インスタンスのパフォーマンスに悪影響が及ぶことがあります。性能低下の程度は、問題のタイプ、およびトレース情報の収集に使用されるリソースの数に応じて異なります。

トレースが求められるときには、DB2 お客様サポートでは以下の情報が用意されています。

- 簡単で段階的な手順
- 各トレースを実行する場所についての説明
- 何をトレースすべきかについての説明
- トレースを求める理由についての説明

- バックアウト手順 (すべてのトレースを無効にする方法など)

どのトレースを取得するかに関しては、DB2 お客様サポートの指示に従う必要がありますが、いつ特定のトレースを取得するよう求められるかについては、以下に示すようないくつかの一般ガイドラインがあります。

- インストール中に問題が発生し、デフォルトのインストール・ログがその問題の原因を調べるのに十分ではない場合には、インストールのトレースが適切です。
- GUI (グラフィカル・ユーザー・インターフェース) ツールのいずれかで問題が発生し、DB2 コマンド・ウィンドウにある明示コマンドを介して実行した際には、その同じアクションが成功した場合、コントロール・センターのトレースが適切です。これによって収集されるのは、コントロール・センターから起動可能なツールに関連した問題のみです。
- CLI アプリケーションで問題が生じ、アプリケーションの外部でその問題を再作成できない場合は、CLI トレースが適切です。
- JDBC アプリケーションで問題が生じ、アプリケーションの外部でその問題を再作成できない場合には、JDBC トレースが適切です。
- 問題が DRDA レイヤーで通信中の情報と直接関連しているなら、DRDA トレースが適切です。
- トレースが実行できるその他のすべての状態では、DB2 トレースが最も適切でしょう。

トレース情報は、エラーの診断に常に役立つとはかぎりません。例えば、以下の状態では、エラー状態を収集しない場合があります。

- 指定したトレース・バッファ・サイズは、トレース・イベントの完全セットを保持するのに十分な大きさがなく、トレースがファイルへの書き込みを停止したとき、またはトレースがラップされた時に、情報が失われました。
- トレース・シナリオが、エラー状態を再現しませんでした。
- エラー状態は再現されましたが、問題が発生した場所などの前提事項が間違っていました。例えば、サーバーで実際のエラーが発生している間に、トレースはクライアント・ワークステーションで収集されました。

DB2 トレース

内部ユーティリティを使用して DB2 トレースを取得する方法が論じられます。トレース・バッファ内のトレース・データからいったん DB2 トレース・ファイルが作成されたら、出力をフォーマットして読みやすくする必要があります。そのファイル内の情報は、特定の問題に対処するために、DB2 サポートで使うことができます。

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、DB2 のサポート・チームや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られます。

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 DB2 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、約 8MB 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に、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すると、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` 装置に送信します。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 DB2 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>.fmtd
```

DRDA トレース

分散リレーショナル・データベース体系 (Distributed Relational Database Architecture™: DRDA) では、共通の構造によってリンクされたデータベース間でデータが受け渡しされる方法を定義しています。DB2 Connect は DRDA を使用して、データベース・クライアント・ワークステーションを、ホスト上または System i™ マシン上のデータベースにリンクします。そのような環境で問題を経験する場合には、DRDA トレース・ファイル、トレースを作成するために使用されるユーティリティ、生成される出力、および出力の分析方法について知る必要があります。DRDA トレースのサンプルも示されます。最後に、後続の送信および受信バッファを確認するよう、お勧めします。それらには、経験している問題に関係のある情報が含まれている場合があるからです。

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 Series i データベース・サーバーとの間)。

トレース・ユーティリティ

db2drdat ユーティリティを利用して、DB2 Connect サーバー (IBM データ・サーバー・クライアントの代理) 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、特定の問題の起点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・ステートメントをホストまたは System i データベース・サーバーに対して発行したが、コマンドが失敗して、失敗の戻りコードを受け取ったとします。そのとき、どのような情報がホストまたは System i 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。ユーザー自身による単純なエラーが、多くの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ホストまたは System i 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。ホストまたは System i 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、ホストまたは System i 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、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、ホストまたは System i 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、ファイル内で最も古いものから順に、最新のものへと配置されます。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. 終了コードのゼロ値は、そのコマンドが正常に完了したことを示し、ゼロ以外の値は、そのコマンドが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、すでに存在しているファイルへ出力を送信した場合、ファイル上の許可により消去を禁止しているのではない限り、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)
- ホストまたは System i 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 System i 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、ホストまたは System i 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 がホストまたは System i 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、ホストまたは System i データベース管理システムについての診断情報に入れます。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、ブランクが埋め込まれ、000C50CC へと続きます。Linux または UNIX IBM データ・サーバー・クライアントについては、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、ホストまたは System i 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、ホストまたは System i 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。

ホストまたは System i 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・セットをホストまたは System i 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 Universal Database for z/OS and OS/390 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 i5/OS 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。

92 ページの図 2 は TCP/IP 接続で DB2 Connect Enterprise Edition バージョン 9.1 および DB2 Universal Database (UDB) 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	..!.....@@@@@@STLECI
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.STLECI.....
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2. 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.....	..}.....>...s..
0010	00162110E2E3D3C5 C3F1404040404040	..!.....@@@STLEC1
0020	40404040404000C 11A1D9858799F485	@@@@.....Regr4e
0030	A599000A11A09585 A6A39695	vr....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 !.\$.	..}....x.....
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	000511A400u.

	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..	..}....n.....
0010	000D002FD8E3C4E2 D8D3F3F7F0000C11	.../.....QTDSL370...
0020	2EC4E2D5F0F8F0F1 F5001600350006115...	.DSN08015.....
0030	9C04B80006119E04 B80006119D04B000
0040	0C11A0D5C5E6E3D6 D540400006212524@...!%\$...NEWTON
0050	34001E244E000624 4C00010014244D00	4..\$.N..\$.L...\$.M.+...<.....(.
0060	06244FFFFF000A11 E8091E768301BE00	.\$0.....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3. 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	@@@AAAAAHLV....<.....
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4. 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	0000000000572020 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 833159653 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5. 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

それ以降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、ホストまたは System i 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番目のバッファーは、ホストまたは System i データベース・サーバー管理システムから、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' が示しているとおおり、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。

92 ページの図 2 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ーの例を示しています。

コントロール・センターのトレース

コントロール・センターで問題のトレースを試みる前に、DB2 コマンドから明示的なコマンドを介して同等のアクションを実行したときに同じ問題が生じないか、まず確認しておくようお勧めします。コントロール・センター内でタスク (またはコントロール・センターから起動できる、他の GUI ツールの 1 つ) を実行すると、しばしば「コマンドの表示」ボタンが表示されます。これを使用すると、ツールが使用するコマンドの構文がそのとおりに示されます。このコマンドを DB2 コマンド・プロンプトからそのまま実行すると成功するのに、GUI ツールから実行すると失敗する場合は、コントロール・センターのトレースを取得するのが適当です。

コントロール・センター内でのみ再現可能な問題のトレースを取得するには、以下のようにしてコントロール・センターを起動します。

```
db2cc -tf filename
```

これによってコントロール・センターのトレース機能がオンになり、指定したファイルにトレースの出力が保管されます。出力ファイルは、Windows では <DB2 インストール・パス>%sqllib%tools に、UNIX および Linux では /home/<userid>/sqllib/tools に保管されます。

注: トレース機能を使用可能に設定してコントロール・センターを起動した場合は、できるだけ少ないステップで問題を再現してください。不要または無関係なツールの項目をクリックしないようにしてください。問題を再現したら、コントロール・センターを閉じます (問題を再現するために他の GUI ツールを開いた場合はそれらも閉じます)。

結果として取得したトレース・ファイルは、分析のため DB2 サポートに送信する必要があります。

JDBC トレース

使用している JDBC ドライバーのタイプによって、実行中のアプリケーションまたはストアド・プロシージャについてトレース・ファイルを取得する方法が異なります。それらのさまざまな方法をここに示します。

Linux、UNIX、および Windows 用の DB2 JDBC Type 2 ドライバーを使用するアプリケーションのトレースの取得

このタイプのトレースは、以下の場所で問題が発生する場合に有効です。

- Linux、UNIX、および Windows 用の DB2 JDBC Type 2 ドライバー (DB2 JDBC Type 2 ドライバー) を使用する JDBC アプリケーション
- DB2 JDBC ストアード・プロシージャの中

注: db2cli.ini ファイルに追加できるキーワードは数多くあります。こうしたキーワードは、アプリケーション動作に影響を及ぼす可能性があります。こうしたキーワードは、アプリケーション問題を解決することもあるれば、その原因となる場合もあります。CLI 資料では取り上げられていないキーワードもあります。そのようなキーワードは DB2 サポートでなければ使用不能です。文書化されていないキーワードが db2cli.ini ファイルにある場合、おそらくそれは DB2 サポートの推奨によるものです。DB2 JDBC Type 2 ドライバーは、内部で DB2 CLI ドライバーを使用してデータベースにアクセスします。例えば、Java の getConnection() メソッドは、内部で DB2 JDBC Type 2 ドライバーによって DB2 CLI SQLConnect() 関数にマップされます。その結果、Java 開発者は DB2 CLI トレースを、DB2 JDBC トレースを補完するものとして活用できます。

1. トレース・ファイルのパスを作成します。すべてのユーザーが書き込み可能なパスを作成するのは重要なことです。

例えば、Windows では次のようにします。

```
mkdir c:%temp%trace
```

Linux および UNIXの場合:

```
mkdir /tmp/trace
chmod 777 /tmp/trace
```

2. CLI 構成キーワードを更新します。これを行うには、以下の 2 つの方法があります。

- db2cli.ini ファイルを手動で編集する。
 - a. プレーン・テキスト・エディターで、db2cli.ini ファイルをオープンします。

デフォルトで、DB2 CLI/ODBC 構成キーワード・ファイル (db2cli.ini) は Windows プラットフォームでは sqllib ディレクトリーにあり、UNIX プラットフォームでは CLI/ODBC アプリケーションを実行しているデータベース・インスタンスの sqllib/cfg ディレクトリーにあります。ODBC ドライバー・マネージャーを使用して Windows プラットフォーム上にユーザー・データ・ソースを構成する場合、db2cli.ini はユーザーのホーム (プロファイル) ディレクトリーに作成することができるかもしれませんが。環境変数 DB2CLIINIPATH を使用して、このファイルのデフォルトをオーバーライドして別の場所を指定することもできます。

- b. このファイルに以下のセクションを追加します (COMMON セクションが既に存在する場合には、変数の追加だけを行います)。

```
[COMMON]
JDBCTrace=1
JDBCTracePathName=<path>
JDBCTraceFlush=1
```

ここで、<path> は、例えば Windows では C:%temp%trace、Linux または UNIX オペレーティング・システムでは /tmp/trace などとなります。

- c. ファイルの最後に空白行を少なくとも 1 行設けて、ファイルを保管します。(これで一部の構文解析エラーを回避できます。)
- UPDATE CLI CFG コマンドを使用して db2cli.ini ファイルを更新する。次のコマンドを実行します。

```
db2 UPDATE CLI CFG FOR SECTION COMMON USING JDBCTrace 1
db2 UPDATE CLI CFG FOR SECTION COMMON USING JDBCTracePathName <path>
```

ここで、<path> は、例えば Windows では C:\temp\%trace、Linux または UNIX オペレーティング・システムでは /tmp/trace などとなります。

```
db2 UPDATE CLI CFG FOR SECTION COMMON USING JDBCTraceFlush 1
```

トレース機能を使ってアプリケーションの問題を診断する場合には、それによりアプリケーション・パフォーマンスに影響があり、テスト・アプリケーションだけではなくすべてのアプリケーションに影響を及ぼすことを覚えておいてください。このため、問題の識別後には忘れずにこの機能をオフにすることが大切です。

3. 次のコマンドを実行して、適切なキーワードが設定されて、取得されていることを検査します。

```
db2 GET CLI CFG FOR SECTION COMMON
```

4. アプリケーションを再開します。

db2cli.ini ファイルはアプリケーション開始時のみに読み取られるため、変更を有効にするには、アプリケーションを再開する必要があります。

JDBC ストアード・プロシージャをトレースする場合には、DB2 インスタンスの再開が必要です。

5. エラーをキャプチャーします。 アプリケーションを実行し、エラーが生成されたら終了します。トレース時に実行している JDBC アプリケーションだけを用いて問題を再現するというような状況は、できるだけ避けてください。そうするのなら、トレース・ファイルをより明快なものにできます。
6. JDBC トレースを使用不可にします。

手動で db2cli.ini の [COMMON] セクションに JDBCTrace=0 キーワードを設定するか、次のコマンドを実行します。

```
db2 UPDATE CLI CFG FOR SECTION COMMON USING Trace 0
db2 UPDATE CLI CFG FOR SECTION COMMON USING JDBCTrace 0
```

7. 実行中およびトレース中のアプリケーションを再開します。
8. トレース・ファイルを収集します。

JDBC トレース・ファイルは、JDBCTracePathName キーワードで指定されたパスに書き込まれます。生成されるファイル名すべてには、.trc という拡張子が最後に付きます。問題再現の際にトレース・パスに生成された、すべてのファイルが必要となります。

DB2 Universal JDBC ドライバーを使用するアプリケーションのトレースの取得

DB2 Universal JDBC ドライバーを使用している SQLJ または JDBC アプリケーションがある場合、以下のいくつかの方法で JDBC トレースを使用できます。

- DataSource インターフェースを使用してデータ・ソースに接続する場合、DataSource.setTraceLevel() および DataSource.setTraceFile() メソッドを使用してトレースを使用可能にします。
- DriverManager インターフェースを使用してデータ・ソースに接続する場合、トレースを使用可能にする最も簡単な方法は、接続を取得する前に DriverManager に logWriter を設定することです。

例:

```
DriverManager.setLogWriter(new PrintWriter(new FileOutputStream("trace.txt")));
```

- DriverManager インターフェースを使用している場合、別の方法として、ドライバのロード時に、 traceFile と traceLevel プロパティを URL の一部として指定できます。

例:

```
String databaseURL =
"jdbc:db2://hal:50000/sample:traceFile=c:/temp/foobar.txt;" ;
```

CLI トレース・ファイル

CLI トレースは、DB2 CLI ドライバーにアクセスするアプリケーションに関する情報を取り込みます。

CLI トレースは、DB2 CLI ドライバーの内部的な働きに関する微小な情報を提供します。

このタイプのトレースは、以下の場所で問題が発生する場合に有効です。

- CLI アプリケーション
- ODBC アプリケーション (ODBC アプリケーションが DB2 CLI インターフェースを使用して DB2 にアクセスした後)
- DB2 CLI ストアード・プロシージャ
- JDBC アプリケーションおよびストアード・プロシージャ

ODBC アプリケーションを診断する際、ODBC トレースまたは DB2 CLI トレースを使用して問題を判別するのが最も簡単という場合は少なくありません。ODBC ドライバー・マネージャーを使用している場合、ODBC トレースを取るための機能が提供されている可能性があります。ODBC のトレースを使用可能にする方法を判別するには、ご使用のドライバー・マネージャーの資料を参照してください。DB2 CLI トレースは DB2 に固有のものです。一般的な ODBC トレースより多くの情報が含まれている場合が少なくありません。通常、これらの両方のトレースは非常に似ています。両方ともアプリケーションからの CLI 呼び出しの項目と出口点をリストしており、これらの呼び出しに対するパラメーターと戻りコードも含まれています。

Linux、UNIX、および Windows 用の DB2 JDBC Type 2 ドライバー (DB2 JDBC Type 2 ドライバー) は、データベースにアクセスするために DB2 CLI ドライバーに依存しています。そのため、Java 開発者は DB2 CLI トレースを使用可能にすることにより、自分のアプリケーションがさまざまなソフトウェア層を介してデータ

ベースと対話する方法に関する追加情報を得ることができます。DB2 JDBC および DB2 CLI トレース・オプションは (どちらも db2cli.ini ファイルで設定されますが)、相互に依存していません。

CLI トレースの使用可能化

CLI トレースは、db2cli.ini ファイルに特定のエントリーを追加することによって使用可能になります。

注: db2cli.ini ファイルに追加できるキーワードは数多くあります。こうしたキーワードは、アプリケーション動作に影響を及ぼす可能性があります。こうしたキーワードは、アプリケーション問題を解決することもあるれば、その原因となる場合もあります。CLI 資料では取り上げられていないキーワードもあります。そのようなキーワードは DB2 サポートでなければ使用不能です。文書化されていないキーワードが db2cli.ini ファイルにある場合、おそらくそれは DB2 サポート・チームの推奨によるものです。

デフォルトでは、DB2 CLI/ODBC 構成キーワード・ファイルは、Windows オペレーティング・システムでは sqllib ディレクトリーにあり、Linux および UNIX オペレーティング・システムでは CLI/ODBC アプリケーションを実行しているデータベース・インスタンスの sqllib/cfg ディレクトリーにあります。ODBC ドライバー・マネージャーを使用して、Windows プラットフォームでユーザー・データ・ソースを構成する場合、ユーザーのホーム (プロファイル) ディレクトリーに db2cli.ini を作成することができます。環境変数 DB2CLIINIPATH を使用して、このファイルのデフォルトをオーバーライドして別の場所を指定することもできます。

1. トレース・ファイルのパスを作成します。

すべてのユーザーが書き込み可能なパスを作成するのは重要なことです。例えば、Windows では次のようにします。

```
mkdir c:%temp%trace
```

Linux および UNIXの場合:

```
mkdir /tmp/trace
chmod 777 /tmp/trace
```

2. CLI 構成キーワードを更新します。

これは、db2cli.ini ファイルを手動で編集するか、あるいは UPDATE CLI CFG コマンドを使用することによって実行できます。

オプション A: db2cli.ini ファイルを手動で編集する。

- a. プレーン・テキスト・エディターで、db2cli.ini ファイルをオープンします。
- b. このファイルに以下のセクションを追加します (COMMON セクションが既に存在する場合には、変数の追加だけを行います)。

```
[COMMON]
Trace=1
TracePathName=<path>
TraceComm=1
TraceFlush=1
TraceTimeStamp=1
TraceScript=1
```

ここで、<path> は、例えば Windows では C:\temp\trace、Linux および UNIX では /tmp/trace などとなります。

- c. ファイルの最後にブランク行を少なくとも 1 行設けて、ファイルを保管します。(これで一部の構文解析エラーを回避できます。)

オプション B: UPDATE CLI CFG コマンドを使用して db2cli.ini ファイルを更新する。次のコマンドを実行します。

```
db2 UPDATE CLI CFG FOR SECTION COMMON USING Trace 1
db2 UPDATE CLI CFG FOR SECTION COMMON USING TracePathName <path>
```

ここで、<path> は、例えば Windows では C:\temp\trace、Linux および UNIX では /tmp/trace などとなります。

```
db2 UPDATE CLI CFG FOR SECTION COMMON USING TraceComm 1
db2 UPDATE CLI CFG FOR SECTION COMMON USING TraceFlush 1
db2 UPDATE CLI CFG FOR SECTION COMMON USING TraceTimeStamp 3
```

3. db2cli.ini の構成を確認します。

次のコマンドを実行して、適切なキーワードが設定されて、取得されていることを検査します。

```
db2 GET CLI CFG FOR SECTION COMMON
```

4. アプリケーションを再開します。

db2cli.ini ファイルが読み取られるのは、アプリケーションの開始時に限られるので、変更を有効にするにはアプリケーションを再始動する必要があります。

CLI ストアード・プロシージャをトレースすると、DB2 インスタンスが再始動します。

5. エラーをキャプチャーします。

アプリケーションを実行し、エラーが生成されたら終了します。問題の再現に関連したアプリケーションだけをトレース時に実行するというように、この状態を絞り込むことができれば、トレース分析はより明確なものになります。

6. CLI トレースを使用不可にします。

db2cli.ini の [COMMON] セクションで **Trace** キーワードを手動でゼロの値に設定するか、以下のコマンドを実行します。

```
db2 UPDATE CLI CFG FOR SECTION COMMON USING Trace 0
```

その後、実行中およびトレース中の可能性があるアプリケーションがあれば、それを再始動してください。

7. トレース情報を収集します。

CLI トレース・ファイルは、**TracePathName** キーワードで指定されたパスに書き込まれます。生成されたファイル名は、p<pid>t<tid>.cli の形式になります。<pid> は、オペレーティング・システムが割り当てたプロセス ID、および <tid> は、アプリケーション処理によって生成されるスレッドごとの数値カウンター (0 から始まる) です。例えば、p1234t1.cli のようになります。DB2 サポートを利用して問題を診断する場合、トレース・パスに生成されたファイルをすべて提供する必要があります。

トレース機能を使ってアプリケーションの問題を診断する場合には、それによりアプリケーション・パフォーマンスに影響があり、テスト・アプリケーションだけではなくすべてのアプリケーションに影響を及ぼすことを覚えておいてください。このため、問題の識別後には忘れずにこの機能をオフにすることが大切です。

CLI トレース・ファイル内の入出力パラメーターの解釈

通常の関数の場合と同様、DB2 CLI 関数には入出力パラメーターがあります。これらの入出力パラメーターが DB2 CLI トレースに現れることがあります。これにより、各アプリケーションが特定の CLI API を呼び出す方法が詳細にわかります。CLI トレースに現れた CLI 関数の入出力パラメーターは、資料の CLI リファレンスの節にある CLI 関数の定義と対照できます。

以下は、CLI トレース・ファイルの断片です。

```
SQLConnect( hDbc=0:1, szDSN="sample", cbDSN=-3, szUID="",
            cbUID=-3, szAuthStr="", cbAuthStr=-3 )
----> Time elapsed - +6.960000E-004 seconds

SQLRETURN  SQLConnect      (
SQLHDBC    ConnectionHandle, /* hdbc */
SQLCHAR    *FAR ServerName, /* szDSN */
SQLSMALLINT NameLength1,      /* cbDSN */
SQLCHAR    *FAR UserName,   /* szUID */
SQLSMALLINT NameLength2,    /* cbUID */
SQLCHAR    *FAR Authentication, /* szAuthStr */
SQLSMALLINT NameLength3);   /* cbAuthStr */
```

CLI 関数の最初の呼び出しには、入力パラメーターとそれに割り当てられた値が表示されます (該当する場合)。

CLI 関数が戻されると、結果として生じた出力パラメーターが表示されます。以下に例を示します。

```
SQLAllocStmt( phStmt=1:1 )
<--- SQL_SUCCESS Time elapsed - +4.444000E-003 seconds
```

このケースでは、CLI 関数 `SQLAllocStmt()` は、出力パラメーター `phStmt` と値 `"1:1"` (接続ハンドル 1、ステートメント・ハンドル 1) を戻しています。

CLI トレース内の動的 SQL の分析

`SQLPrepare()` および `SQLBindParameter()` でパラメーター・マーカを宣言および使用することにより、DB2 CLI トレースは動的 SQL がどのように実行されるかも示します。これにより、どの SQL ステートメントが実行されるかを実行時に判別できます。

以下のトレース・エントリは、SQL ステートメントの準備を示しています ('?' はパラメーター・マーカを示します)。

```
SQLPrepare( hStmt=1:1, pszSqlStr=
"select * from employee where empno = ?",
cbSqlStr=-3 )
----> Time elapsed - +1.648000E-003 seconds
( StmtOut="select * from employee where empno = ?" )
SQLPrepare( )
<--- SQL_SUCCESS Time elapsed - +5.929000E-003 seconds
```


以下のトレース・エントリーは、パラメーター・マーカを最大長 7 の CHAR と
してバインディングする様子を示しています。

```
SQLBindParameter( hStmt=1:1, iPar=1, fParamType=SQL_PARAM_INPUT,  
fCType=SQL_C_CHAR, fSQLType=SQL_CHAR, cbColDef=7, ibScale=0,  
rgbValue=&00854f28, cbValueMax=7, pcbValue=&00858534 )  
---> Time elapsed - +1.348000E-003 seconds  
SQLBindParameter( )  
<--- SQL_SUCCESS Time elapsed - +7.607000E-003 seconds
```

次いで動的 SQL ステートメントが実行されます。 rgbValue="000010" は、実行時
にアプリケーションによってパラメーター・マーカから置き換えられた値を示し
ています。

```
SQLExecute( hStmt=1:1 )  
---> Time elapsed - +1.317000E-003 seconds  
( iPar=1, fCType=SQL_C_CHAR, rgbValue="000010" - X"303030303130",  
pcbValue=6, piIndicatorPtr=6 )  
sqlccsend( ulBytes - 384 )  
sqlccsend( Handle - 14437216 )  
sqlccsend( ) - rc - 0, time elapsed - +1.915000E-003  
sqlccrecv( )  
sqlccrecv( ulBytes - 1053 ) - rc - 0, time elapsed - +8.808000E-003  
SQLExecute( )  
<--- SQL_SUCCESS Time elapsed - +2.213300E-002 seconds
```

CLI トレース内の時間情報の解釈

DB2 CLI トレースから時間情報を収集する方法がいくつかあります。デフォルトで
は、CLI トレースは、この特定のスレッドで最後の CLI API 呼び出しを実行して
からアプリケーションで費やされた時間を収集します。これには DB2 で費やされ
た時間だけでなく、クライアントとサーバーの間のネットワーク時間も含まれま
す。例:

```
SQLAllocStmt( hDbc=0:1, phStmt=&0012ee48 )  
---> Time elapsed - +3.964187E+000 seconds
```

(この時間値は、最後の CLI API が呼び出されてからアプリケーションで費やされ
た時間を示しています)

```
SQLAllocStmt( phStmt=1:1 )  
<--- SQL_SUCCESS Time elapsed - +4.444000E-003 seconds
```

(関数は完了しているので、この時間値は、ネットワーク時間を含め、DB2 で費やさ
れた時間を示します)

時間情報を収集する別の方法は、CLI キーワード TraceTimeStamp を使用すること
です。このキーワードを使用すると、DB2 CLI API 呼び出しが呼び出されたり、結
果が戻されたりするたびに、タイム・スタンプが生成されます。キーワードの表示
オプションは 4 つあります。タイム・スタンプ情報なし、プロセッサ・ティック
および ISO タイム・スタンプ、プロセッサ・ティック、ISO タイム・スタンプの
4 つです。

これは、「CLI0125E - 関数シーケンス・エラー」など、時間に関連した問題を扱う
際にたいへん役立つ可能性があります。また、マルチスレッド・アプリケーション
を扱う際に、どのイベントが最初に発生したかを判別するためにも役立つことがあ
ります。

CLI トレース内の不明な値の解釈

DB2 CLI 関数が、CLI トレース内の入力パラメーターの値として "Unknown value" (不明な値) を戻すことがあります。これが生じ得るのは、DB2 CLI ドライバーがその入力パラメーターに特有の値を予期しているのに、アプリケーションが別の値を提供している場合です。例えば、CLI 関数の旧式の定義に従っている場合や、推奨されない CLI 関数を使用している場合に、これが生じ得ます。

CLI 関数呼び出しが "Option value changed" または "Keyset Parser Return Code" を返すこともあります。これはキー・セット・カーソルがメッセージを表示することの結果として生じます。例えば、カーソルが何らかの理由で静的カーソルに格下げされる場合などです。

```
SQLExecDirect( hStmt=1:1, pszSqlStr="select * from org", cbSqlStr=-3 )
  ---> Time elapsed - +5.000000E-002 seconds
( StmtOut="select * from org" )
( COMMIT=0 )
( StmtOut=" SELECT A.TABSCHEMA, ..... )
( StmtOut=" SELECT A.TABSCHEMA, ..... )
( Keyset Parser Return Code=1100 )

SQLExecDirect( )
  <--- SQL_SUCCESS_WITH_INFO   Time elapsed - +1.06E+001 seconds
```

上記の CLI トレースでは、キー・セット・パーサーが戻りコード 1100 を示しています。これは、表のユニーク索引または主キーがなく、したがってキー・セット・カーソルが作成できなかったことを示しています。これらの戻りコードは外部化されていないので、戻りコードの意味についてさらに情報が必要な場合、現時点では DB2 サポートに連絡を取る必要があります。

SQLError または SQLDiagRec を呼び出すと、カーソル・タイプが変更されたことが示されます。アプリケーションは、どの属性が変更されたかを判別するために、カーソル・タイプと並行性を照会する必要があります。

マルチスレッド CLI トレース出力の解釈

CLI トレースは、マルチスレッド・アプリケーションをトレースできます。マルチスレッド・アプリケーションをトレースする最善の方法は、CLI キーワード TracePathName を使用することです。これにより、p<pid>t<tid>.cli という名前のトレース・ファイルが作成されます (<tid> は、アプリケーションの実際のスレッド ID)。

実際のスレッド ID を知る必要のある場合は、この情報を CLI トレースのヘッダーから調べることができます。

```
[ Process: 3500, Thread: 728 ]
[ Date & Time:          02/17/2006 04:28:02.238015 ]
[ Product:             QDB2/NT DB2 v9.1.0.190 ]
...
```

CLI キーワード TraceFileName を使用すれば、マルチスレッド・アプリケーションを 1 つのファイルにトレースできます。この方法を使うとユーザーの選んだ 1 つのファイルが生成されますが、読みにくくなるかもしれません。というのは、あるスレッドの API が別のスレッドの別の API と同時に実行されることがあり、トレースを検査するときに混同してしまう恐れがあるためです。

通常は TraceTimeStamp をオンにするようお勧めします。そうすれば、特定の API の実行された時刻を見て、イベントの本当の順序を判別することができます。これは、1 つのスレッドが別のスレッドの中で問題を引き起こすといった問題を調査する際にたいへん役立ちます (例えば、CLI0125E - 関数シーケンス・エラーなど)。

プラットフォーム固有のツール

ご使用のプラットフォームに関連した診断情報を収集する、トラブルシューティング・コマンド、パフォーマンス・モニター・ユーティリティー、その他の方式があります。それらのツールは、Windows オペレーティング・システム、または Linux および UNIX オペレーティング・システムの一部として提供されます。

診断ツール (Windows)

Windows オペレーティング・システムでは、以下の診断ツールが使用可能です。

イベント・ビューアー、パフォーマンス・モニターおよび他の管理ツール

「管理ツール」フォルダーには、イベント・ログへのアクセスおよびパフォーマンス情報へのアクセスを含む、さまざまな診断情報が提供されています。

タスク マネージャ

「タスク マネージャ」には、Windows サーバーで実行されているすべてのプロセスと、メモリー使用の詳細が表示されます。どの DB2 プロセスが実行されているかを確認し、パフォーマンス上の問題を診断するには、このツールを使用します。このツールを使用すると、メモリー使用量、メモリーの限界、使用されているスワッパー・スペース、およびプロセスのメモリー漏えいを判別できます。

「タスク マネージャ」をオープンするには、Ctrl + Alt + Delete を押し、選択可能なオプションから「タスク マネージャ」をクリックします。

ワトソン博士

「ワトソン博士」ユーティリティーは、一般保護障害 (GPF) の発生時に呼び出されます。これは、問題の診断に役立つ可能性のあるデータをログに記録し、この情報をファイルに保管します。このユーティリティーを開始するには、コマンド行に `drwatson` と入力します。

診断ツール (Linux および UNIX)

このセクションでは、Linux および UNIX プラットフォームにおける、トラブルシューティングおよびパフォーマンス・モニターのために必要ないくつかのコマンドについて説明しています。これらのコマンドの詳細は、コマンド行でコマンドの前に、「man」と入力すると表示されます。システムで、発生している問題の原因を識別するのに役立つデータを収集および処理するには、これらのコマンドを使用します。いったん収集したデータは、問題に詳しい担当者が調べることもできますし、要請に応じて DB2 サポートに送ることもできます。

トラブルシューティング・コマンド (AIX)

以下の AIX システム・コマンドが、DB2 のトラブルシューティングに役立ちます。

errpt errpt コマンドは、ハードウェア・エラーおよびネットワーク障害などのシステム・エラーを報告します。

- 各エラーごとに 1 行の概要を表示するには、errpt を使用します。
- 各エラーごとに 1 ページの詳細ビューを表示するには、errpt -a を使用します。
- エラー番号が "1581762B" のエラーには、errpt -a -j 1581762B を使用します。
- 過去にページング・スペースを使い尽くしたかどうかを確認するには、errpt | grep SYSVMM を使用します。
- トークンリング・カードまたはディスクに問題があるかどうかを確認するには、"disk" および "tr0" 句を errpt の出力で確認します。

lsps lsps -a コマンドは、ページング・スペースの使用状況をモニターおよび表示します。

lsattr このコマンドは、さまざまなオペレーティング・システムのパラメータを表示します。例えば、以下のコマンドを使用してデータベース・パーティション上の実際のメモリーの量を確認します。

```
lsattr -l sys0 -E
```

xmperf

Motif を使用する AIX システムでは、このコマンドはシステム関連のパフォーマンス・データを収集および表示する、グラフィカル・モニターを開始します。このモニターでは、各データベース・パーティションごとの 3 ディメンションのダイアグラムが単一のウィンドウに表示されるので、高水準なモニターに役立ちます。しかし、アクティビティが低い場合、このモニターからの出力は限定された値になります。

spmon 並列システム・サポート・プログラム (PSSP) の一部としてシステム・パーティションを使用するときには、SP スイッチがすべてのワークステーションで実行されていることを確認しなければならない場合があります。すべてのデータベース・パーティションの状況を表示するには、コントロール・ワークステーションから、以下のコマンドのいずれかを使用します。

- spmon -d ASCII 出力用
- spmon -g グラフィカル・ユーザー・インターフェース用

別の方法では、スイッチがダウンしているかを調べるのに、データベース・パーティション・ワークステーションから netstat -i コマンドを使用します。スイッチがダウンしている場合は、データベース・パーティション名の横にアスタリスク (*) が表示されます。例えば、以下のようになります。

```
css0* 65520 <Link>0.0.0.0.0
```

スイッチがオンの場合は、アスタリスクは表示されません。

トラブルシューティング・コマンド (Linux および UNIX)

以下のシステム・コマンドは、指定されていない場合は Linux、および AIX を含むすべての UNIX システム用です。

- df** df コマンドでは、ファイル・システムがフルかどうかを表示できます。
- すべてのファイル・システム (マウントされたものを含む) に、どれだけのフリー・スペースがあるかを表示するには、df を使用します。
 - 名前に "dev" を含むすべてのファイル・システムに、どれだけのフリー・スペースがあるかを表示するには、df | grep dev を使用します。
 - ホーム・ファイル・システムに、どれだけのフリー・スペースがあるかを表示するには、df /home を使用します。
 - ファイル・システム "tmp" に、どれだけのフリー・スペースがあるかを表示するには、df /tmp を使用します。
 - マシンに十分なフリー・スペースがあるかどうかを確認するには、以下のコマンドからの出力を確認します。df /usr、df /var、df /tmp、および df /home
- truss** このコマンドは、1 つ以上のプロセスでのシステム呼び出しのトレースに便利です。
- pstack** Solaris 2.5.1 以上で使用可能です。/usr/proc/bin/pstack コマンドはスタック・トレースバック情報を表示します。/usr/proc/bin ディレクトリーには、中断されていると思われるプロセスをデバッグする、その他のツールが含まれています。

パフォーマンス・モニター・ツール

以下のツールは、システムのパフォーマンスをモニターするために使用できます。

- vmstat** このコマンドは、何かが中断されているのか、またはただ時間がかかっているのかを判別するのに役立ちます。ページイン (pi) およびページアウト (po) 列に検出されるページング率をモニターすることができます。その他の重要な列は、割り振られた仮想記憶域 (avm) および、空いている仮想記憶域 (fre) の量です。
- iostat** このコマンドは、I/O アクティビティのモニターに役立ちます。読み取りおよび書き込み率を使用して、特定の SQL 操作に必要な時間を見積もれます。(それらがシステムでの唯一のアクティビティの場合)
- netstat** このコマンドでは、各データベース・パーティション上のネットワーク・トラフィックと、検出されたエラー・パケットの数を知ることができます。これは、ネットワークの問題を切り分けるのに役立ちます。
- system file** Solaris オペレーティング・システムで使用可能です。/etc/system ファイルには、一度にシステムで許可される最大ユーザー数、ユーザーごとの最大プロセス数、およびリソースのサイズおよび数に関するプロセス間通信 (IPC) 制限などの、カーネル構成限界の定義が含まれています。これらの限界は、Solaris オペレーティング・システム・マシンでの DB2 のパフォーマンスに影響を与えるため、重要です。

第 5 章 既知の問題の検索方法

既知の問題を説明している資料には、DB2 APAR、ホワイトペーパー、IBM Redbooks™ 資料、技術情報、マニュアルなど、多数のリソースがあります。発生した問題の解決策がすでに存在するかどうかを素早く判別するには、これらの資料 (およびその他のリソース) を効率的に検索する必要があります。

検索する前に、問題の状況をはっきり把握しておく必要があります。

問題の状況をはっきり理解したら、既存の解決策が見つかる可能性を高くするために、検索キーワードのリストを作成する必要があります。以下のヒントを参考にしてください。

1. 複数のキーワードを検索で使用します。より適切な検索語を使用すれば、より良い検索結果が得られます。
2. まず結果を絞り込んで検索し、次に必要に応じて検索の幅を広げていきます。例えば、戻された結果が少なすぎる場合には、重要度の低い検索語をいくつか消去して再び検索してみます。あるいは、どのようなキーワードを使うべきかわからない場合には、最初に少数のキーワードを使って幅広く検索し、戻された結果を調べれば、追加すべきキーワードを選択できるでしょう。
3. (複数の単語からなる) 特定の句を検索した方が効率的な場合もあります。例えば、引用符を使って "administration notification file" (管理通知ファイル) と入力すれば、入力内容と同じ語句がこの順序で含まれるドキュメントだけが検出されます。(これら 3 つの単語の任意の組み合わせが含まれるドキュメントがすべて検出されるわけではありません。)
4. ワイルドカードを使用します。特定の SQL エラーが発生した場合、キーワードに「SQL5005<wildcard>」を使うことができます (<wildcard> は検索対象のリソースに応じた適切なワイルドカード)。こうすれば、単に「SQL5005」または「SQL5005c」を使って検索した場合よりも多くの結果が戻されるでしょう。
5. 例えば、インスタンスが異常終了してトラップ・ファイルが生成された場合には、トラップまたはコア・ファイルのスタック・トレースバック内の最初のいくつかの関数をキーワードとして使って既知の問題を検索してください。戻された結果の数が多すぎる場合には、「trap」、「abend」、「crash」などのキーワードを追加してみてください。
6. オペレーティング・システム固有の語句 (例えばシグナル番号やエラー番号の値) を見つけたい場合には、値そのものではなく、定数をキーワードとして検索してみてください。例えば、エラー番号 27 ではなく「EFBIG」を検索します。

一般的には、以下のような検索語を使用すれば成功率が高くなります。

- 実行したコマンドを記述する語句
- 症状を記述する語句
- 診断情報に含まれる語句

トラブルシューティングのリソース

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に関する広範囲な情報を利用できます。

DB2 ドキュメンテーション

トラブルシューティング情報は、DB2 インフォメーション・センターおよび DB2 ライブラリー (PDF 資料) で提供されています。DB2 インフォメーション・センターで、(ブラウザ・ウィンドウの左側の) ナビゲーション・ツリーの「サポートおよびトラブルシューティング」ブランチを参照すると、DB2 トラブルシューティング・ドキュメンテーションの完全なリストが見つかります。

DB2 Technical Support の Web サイト

問題が発生した場合、可能性のある原因と解決策を見つけるうえで DB2 Technical Support の Web サイトが役立ちます。Technical Support のサイトには、最新の DB2 資料、技術情報、プログラム診断依頼書 (APAR)、およびフィックスパックへのリンクが掲載され、最新の DB2 内部エラー・コード・リストその他のリソースが提供されています。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 サイトのアドレス: <http://www.ibm.com/software/data/db2/udb/support>

第 6 章 DB2 製品フィックスの入手

フィックスパックには、製品テスト中に IBM が検出した問題、および製品使用中にお客様が見つけた問題に対する、コード更新および修正が入っています。最新のフィックスパックを見つける方法、およびデータベース環境にフィックスを適用する方法が論じられます。

フィックスパックの適用

問題のない運用を続けるために、ご使用の DB2 環境を常に最新のフィックスパック・レベルで実行することをお勧めします。フィックスパックを正常にインストールするには、インストール前およびインストール後に必要なタスクをすべて実行します。

DB2 フィックスパックは、IBM でのテストの際に検出された問題に対するフィックス (プログラム診断依頼書 (APAR))、アップデート、およびお客様から報告された問題のフィックスを含んでいます。各フィックスパックに含まれている APARLIST.TXT ファイルでは、含まれているフィックスについて説明されています。

フィックスパックは累積されます。つまり、ある任意のバージョンの DB2 の最新のフィックスパックには、同じバージョンの DB2 のそれまでのフィックスパックを更新した内容がすべて入っているということです。

使用できるフィックスパック・イメージは、以下のとおりです。

- 単一サーバー・イメージ。

単一サーバー・イメージには、すべての DB2 サーバー製品および IBM Data Server Clientに必要な、新規および更新されたコードが含まれます。複数の DB2 サーバー製品が単一の場所にインストールされている場合、DB2 サーバーのフィックスパックは、保守コード更新をすべてのインストールされた DB2 サーバー製品に適用します。Data Server Client のフィックスパックは、1 つの DB2 サーバーのフィックスパック (つまり、Enterprise Server Edition、Workgroup Server Edition、Express Edition、Personal Edition、Connect Enterprise Edition、Connect Application Server Edition、Connect Unlimited Edition for zSeries[®]、および Connect Unlimited Edition for i5/OS[®] の各サーバー製品のいずれか 1 つを保守可能なフィックスパック) に含まれています。DB2 サーバーのフィックスパックを使用して、Data Server Clientをアップグレードできます。

また、単一サーバー・イメージは、すべての DB2 データベース・サーバー製品の特定のフィックスパック・レベルでのデフォルトの DB2 試用版ライセンスでのインストールに使用することもできます。

- その他の DB2 データベース製品ごとのフィックスパック。

このフィックスパックは、サーバー以外のデータベース製品またはアドオン製品をインストールする場合にのみ使用します。例えば、IBM Data Server Runtime Client や Query Patroller などです。

インストールしている DB2 製品が DB2 サーバー製品または Data Server Client のみの場合は、このタイプのフィックスパックは使用しないでください。代わりに、単一サーバー・イメージのフィックスパックを使用します。

Windows プラットフォームの場合、複数の DB2 データベース製品 (それには Data Server Client または DB2 サーバーではない製品が少なくとも 1 つ含まれている) が 1 つの DB2 コピー内にインストールされていれば、それに対応する製品固有のフィックスパックをすべてダウンロードして解凍してから、フィックスパックのインストール・プロセスを開始する必要があります。

- Universal フィックスパック (Linux または UNIX プラットフォームの場合のみ)。

Universal フィックスパックは、既に複数の DB2 データベース製品がインストールされている場合のインストールに用います。

インストールしている DB2 製品が DB2 サーバー製品または Data Server Client のみの場合は、Universal フィックスパックは必要ありません。この場合は、単一サーバー・イメージのフィックスパックを使用してください。

制約事項

- DB2 バージョン 9.5 フィックスパックは、DB2 バージョン 9.5 一般出荷版 (GA) またはフィックスパック・レベルのコピーにのみ適用可能です。
- フィックスパックをインストールする前に、すべての DB2 インスタンス、DAS、および更新される DB2 コピーに関連するアプリケーションを停止してください。
- データベース・パーティション・フィーチャー (DPF) を使用している場合、フィックスパックのインストールの前に、すべてのノード上のデータベース・マネージャーを停止する必要があります。フィックスパックは、インスタンス所有ノードおよび他のすべてのパーティション・ノードにインストールする必要があります。インスタンスに参加しているすべてのコンピューターを同じフィックスパック・レベルにアップグレードする必要があります。
- Linux または UNIX オペレーティング・システムの場合:
 - DB2 製品がネットワーク・ファイル・システム (NFS) 上にある場合、フィックスパックをインストールする前に、すべてのインスタンス、DB2 Administration Server (DAS)、プロセス間通信 (IPC)、および同じ NFS マウント・インストールを使用する他のマシン上のアプリケーションが完全に停止していることを確認する必要があります。
 - システム・コマンド `fuser` または `lsof` が使用できない場合、`installFixPack` コマンドはロード済みの DB2 ファイルを検出できません。DB2 ファイルがロードされていないことを確認し、フィックスパックをインストールするためのオーバーライド・オプションを指定

する必要があります。UNIX では、ロード済みファイルをチェックするために `fuser` コマンドが必要です。Linux 上では、`fuser` コマンドまたは `lsuf` コマンドが必要です。

オーバーライド・オプションの詳細については、`installFixPack` コマンドを参照してください。

- クライアント・アプリケーション上では、フィックスパックを適用した後に、アプリケーションの自動バインドを実行するために、ユーザーはバインド権限を持っている必要があります。
- DB2 フィックスパックをインストールしても、IBM Data Studio Administration Console または IBM Data Studio にはサービスは提供されません。

Linux または UNIX 上での非ルート・インストールの場合、ルート・ベースのフィーチャー (High Availability やオペレーティング・システム・ベースの認証など) は、`db2rfe` コマンドを使用することにより有効にすることができます。ルート・ベースのフィーチャーが DB2 製品のインストール後に使用可能になっていた場合、それらのフィーチャーを再び使用可能にするために、フィックスパックを適用するたびに `db2rfe` コマンドを再実行する必要があります。詳しくは、以下の非ルート関連のリンクを参照してください。

Linux または UNIX オペレーティング・システム上で、各国語がインストールされている場合、それぞれの各国語フィックスパックも別途必要になります。各国語フィックスパックのみをインストールすることはできません。Universal フィックスパックまたは製品固有のフィックスパックも一緒に適用されていなければならない。なおかつそれらの両方のフィックスパック・レベルが同じでなければなりません。例えば、Universal フィックスパックを Linux または UNIX 上の英語以外の DB2 データベース製品に適用する場合、DB2 データベース製品を更新するには Universal フィックスパックと各国語フィックスパックの両方を適用する必要があります。

複数の DB2 コピーが同一システム上にある場合、それらのコピーのバージョンとフィックスパック・レベルはそれぞれ異なっている可能性があります。1 つ以上の DB2 コピーにフィックスパックを適用したい場合、それぞれの DB2 コピーにフィックスパックを 1 つずつインストールする必要があります。

フィックスパック、テスト・フィックス、および APAR

プログラム診断依頼書 (APAR) は、変更の加えられていない現行リリースの IBM プログラムの欠陥とされるものによって生じた問題の正式な報告書です。APAR は、IBM によってテスト中に見つかった問題、およびお客様によって報告された問題を説明しています。

APAR で説明されている問題を解決する、修正された DB2 コードは、フィックス・パックまたはテスト・フィックスで配布されることがあります。

フィックスパック

フィックスパックは、APAR フィックスの累積のコレクションです。特に、フィックスパックでは、DB2 の新規リリースの間に生じた APAR を扱っています。これは、特定の保守レベルに上げることを目的としたものです。フィックスパックには、以下の特性があります。

- 累積的なものです。DB2 のリリースの中には、フィックスパックで、そのリリースの以前のフィックスパックで出荷されたすべての APAR フィックスが置き換えられる、つまり、それらがすべて含まれるものもあります。
- 多数の APAR を含んでいます。
- DB2 Technical Support の Web サイトで公開されており、パスポート・アドバンテージ・プログラムを利用して製品を購入したお客様が一般にこれを入手できます。
- IBM によって十分にテストされています。
- Readme およびリリース・ノートがセットに含まれます。
 - フィックスパックの Readme には、フィックスパックのインストールおよび除去に関する指示が記載されています。
 - リリース・ノートに、製品に対する変更についての情報が記載されています。

注: APAR の修正がフィックスパックで提供されると、APAR のステータスは「オープン」から「プログラム・エラーとしてクローズ」に変わります。個々の APAR のステータスは、DB2 Technical Support の Web サイト上の APAR の説明を調べることによって判別できます。

テスト・フィックス

テスト・フィックスは、問題の報告を受けて、テストの目的で特定のお客様に提供される一時的な解決策です。テスト・フィックスには、以下の特性があります。

- 通常は、1 つの APAR を含みます。
- DB2 サポートから入手するもので、一般に出荷されるものではありません。
- IBM によって限られた範囲でテストされています。
- 最小限のドキュメンテーション (テスト・フィックスの適用方法、修正される APAR、テスト・フィックスの除去に関する指示など) が含まれます。

テスト・フィックスは、新たな問題が明らかになり、その問題に対する回避策がなく、次のフィックスパックが入手可能になるまで待つことができないという状況の場合に提供されます。例えば、問題によって業務に重大な影響が生じている場合、APAR がフィックスパックで対応されるまでの間その状況を軽減するためのテスト・フィックスが提供されることがあります。

問題のない運用を続けるために、ご使用の DB2 環境を常に最新のフィックスパック・レベルで実行することをお勧めします。新しいフィックスパックが入手可能になったことの通知を受け取るには、DB2 Technical Support の Web サイト (<http://www.ibm.com/software/data/db2/udb/support>) で My Support の E メール更新に登録してください。

テスト・フィックスの適用

テスト・フィックスは、問題の報告を受けて、テストの目的で特定のお客様に提供される一時的なフィックスです。各テスト・フィックスには、**Readme** が付属しています。テスト・フィックスの **Readme** には、テスト・フィックスのインストールおよびアンインストールに関する指示や、テスト・フィックスに組み込まれている APAR (ある場合) のリストが記載されています。

各テスト・フィックスには固有の前提条件があります。詳細は、テスト・フィックスに付属の **Readme** を参照してください。

テスト・フィックスには以下の 2 つのタイプがあります。

- 個々の DB2 製品についてのテスト・フィックス。これらのテスト・フィックスは、製品の既存のインストールに適用することも、既存の DB2 インストールがない場合に製品のフル・インストールを行うために使用することもできます。
- ユニバーサル・テスト・フィックス (Linux および UNIX のみ)。ユニバーサル・テスト・フィックスは、複数の DB2 製品がインストールされているインストール環境のためのものです。

各国語がインストールされている場合、個別の各国語テスト・フィックスも必要になることがあります。各国語テスト・フィックスは、インストールされている DB2 製品と同じテスト・フィックス・レベルである場合にのみ適用できます。ユニバーサル・テスト・フィックスを適用する場合、DB2 製品を更新するには、ユニバーサル・テスト・フィックスと各国語テスト・フィックスの両方を適用する必要があります。

テスト・フィックスを DB2 お客様サポートから入手し、**Readme** の指示に従って、テスト・フィックスのインストール、テスト、および除去 (必要な場合) を行います。

テスト・フィックスを複数パーティションのデータベース・パーティション環境にインストールする場合、システムはオフラインになっている必要があり、インスタンスに参加するすべてのコンピューターを同じテスト・フィックス・レベルにアップグレードする必要があります。

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。英語の DB2 バージョン 9.5 のマニュアル (PDF 形式) とその翻訳版は、www.ibm.com/support/docview.wss?rs=71&uid=swg2700947 からダウンロードできます。

この表には印刷資料が入手可能かどうかを示されていますが、国または地域によっては入手できない場合があります。

資料番号は、資料が更新される度に大きくなります。資料を参照する際は、以下にリストされている最新版であることを確認してください。

注: DB2 インフォメーション・センターは、PDF やハードコピー資料よりも頻繁に更新されます。

表 6. DB2 の技術情報

資料名	資料番号	印刷資料が入手可能かどうか
管理 API リファレンス	SC88-4431-01	入手可能
管理ルーチンおよびビュー	SC88-4435-01	入手不可
コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻	SC88-4433-01	入手可能
コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻	SC88-4434-01	入手可能
コマンド・リファレンス	SC88-4432-01	入手可能
データ移動ユーティリティガイドおよびリファレンス	SC88-4421-01	入手可能
データ・リカバリーと高可用性ガイドおよびリファレンス	SC88-4423-01	入手可能
データ・サーバー、データベース、およびデータベース・オブジェクトのガイド	SC88-4259-01	入手可能
データベース・セキュリティ・ガイド	SC88-4418-01	入手可能
ADO.NET および OLE DB アプリケーションの開発	SC88-4425-01	入手可能
組み込み SQL アプリケーションの開発	SC88-4426-01	入手可能
Java アプリケーションの開発	SC88-4427-01	入手可能
Perl および PHP アプリケーションの開発	SC88-4428-01	入手不可
SQL および外部ルーチンの開発	SC88-4429-01	入手可能
データベース・アプリケーション開発の基礎	GC88-4430-01	入手可能

表 6. 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
DB2 インストールおよび管理 概説 (Linux および Windows 版)	GC88-4439-01	入手可能
国際化対応ガイド	SC88-4420-01	入手可能
メッセージ・リファレンス 第 1 巻	GI88-4109-00	入手不可
メッセージ・リファレンス 第 2 巻	GI88-4110-00	入手不可
マイグレーション・ガイド	GC88-4438-01	入手可能
Net Search Extender 管理および ユーザーズ・ガイド	SC88-4630-01	入手可能
パーティションおよびクラスタ リングのガイド	SC88-4419-01	入手可能
Query Patroller 管理およびユー ザーズ・ガイド	SC88-4611-00	入手可能
IBM データ・サーバー・クライ アント機能 概説およびインス トール	GC88-4441-01	入手不可
DB2 サーバー機能 概説および インストール	GC88-4440-01	入手可能
Spatial Extender and Geodetic Data Management Feature ユー ザーズ・ガイドおよびリファレ ンス	SC88-4629-01	入手可能
SQL リファレンス 第 1 巻	SC88-4436-01	入手可能
SQL リファレンス 第 2 巻	SC88-4437-01	入手可能
システム・モニター ガイドお よびリファレンス	SC88-4422-01	入手可能
問題判別ガイド	GI88-4108-01	入手不可
データベース・パフォーマンス のチューニング	SC88-4417-01	入手可能
Visual Explain チュートリアル	SC88-4449-00	入手不可
新機能	SC88-4445-01	入手可能
ワークロード・マネージャー ガイドおよびリファレンス	SC88-4446-01	入手可能
pureXML ガイド	SC88-4447-01	入手可能
XQuery リファレンス	SC88-4448-01	入手不可

表 7. 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
DB2 Connect Personal Edition 概説およびインストール	GC88-4443-01	入手可能

表7. DB2 Connect 固有の技術情報 (続き)

資料名	資料番号	印刷資料が入手可能かどうか
DB2 Connect サーバー機能 概説およびインストール	GC88-4444-01	入手可能
DB2 Connect ユーザーズ・ガイド	SC88-4442-01	入手可能

表8. 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
Information Integration: フェデレーテッド・システム 管理ガイド	SC88-4166-01	入手可能
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	SC88-4167-02	入手可能
Information Integration: フェデレーテッド・データ・ソース 構成ガイド	SC88-4185-01	入手不可
Information Integration: SQL レプリケーション ガイドおよびリファレンス	SC88-4168-01	入手可能
Information Integration: レプリケーションとイベント・パブリッシング 概説	GC88-4187-01	入手可能

DB2 の印刷資料の注文方法

DB2 の印刷資料が必要な場合、オンラインで購入することができますが、すべての国および地域で購入できるわけではありません。DB2 の印刷資料については、IBM 営業担当員にお問い合わせください。DB2 PDF ドキュメンテーション DVD の一部のソフトコピー・ブックは、印刷資料では入手できないことに留意してください。例えば、「DB2 メッセージ・リファレンス」はどちらの巻も印刷資料としては入手できません。

DB2 PDF ドキュメンテーション DVD で利用できる DB2 の印刷資料の大半は、IBM に有償で注文することができます。国または地域によっては、資料を IBM Publications Center からオンラインで注文することもできます。お客様の国または地域でオンライン注文が利用できない場合、DB2 の印刷資料については、IBM 営業担当員にお問い合わせください。DB2 PDF ドキュメンテーション DVD に収録されている資料の中には、印刷資料として提供されていないものもあります。

注: 最新で完全な DB2 資料は、DB2 インフォメーション・センター (<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5>) で参照することができます。

DB2 の印刷資料は以下の方法で注文することができます。

- 日本 IBM 発行のマニュアルはインターネット経由でご購入いただけます。詳しくは <http://www.ibm.com/shop/publications/order> の「ご注文について」をご覧ください。資料の注文情報にアクセスするには、お客様の国、地域、または言語を選択してください。その後、各ロケーションにおける注文についての指示に従ってください。
- DB2 の印刷資料を IBM 営業担当員に注文するには、以下のようになります。
 1. 以下の Web サイトのいずれかから、営業担当員の連絡先情報を見つけてください。
 - IBM Directory of world wide contacts (www.ibm.com/planetwide)
 - IBM Publications Web サイト (<http://www.ibm.com/shop/publications/order>)
国、地域、または言語を選択し、お客様の所在地に該当する Publications ホーム・ページにアクセスしてください。このページから、「このサイトについて」のリンクにアクセスしてください。
 2. 電話をご利用の場合は、DB2 資料の注文であることをご指定ください。
 3. 担当者に、注文する資料のタイトルと資料番号をお伝えください。タイトルと資料番号は、118 ページの『DB2 テクニカル・ライブラリー (ハードコピーまたは PDF 形式)』でご確認いただけます。

コマンド行プロセッサから SQL 状態ヘルプを表示する

DB2 は、SQL ステートメントの結果の原因になったと考えられる条件の SQLSTATE 値を戻します。SQLSTATE ヘルプは、SQL 状態および SQL 状態クラス・コードの意味を説明します。

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

```
? sqlstate or ? class code
```

ここで、*sqlstate* は有効な 5 桁の SQL 状態を、*class code* は SQL 状態の最初の 2 桁を表します。

例えば、? 08003 を指定すると SQL 状態 08003 のヘルプが表示され、? 08 を指定するとクラス・コード 08 のヘルプが表示されます。

異なるバージョンの DB2 インフォメーション・センターへのアクセス

DB2 バージョン 9.5 のトピックを扱っている DB2 インフォメーション・センターの URL は、<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/>です。

DB2 バージョン 9 のトピックを扱っている DB2 インフォメーション・センターの URL は <http://publib.boulder.ibm.com/infocenter/db2luw/v9/>です。

DB2 バージョン 8 のトピックについては、バージョン 8 のインフォメーション・センターの URL <http://publib.boulder.ibm.com/infocenter/db2luw/v8/>にアクセスしてください。

DB2 インフォメーション・センターでの希望する言語でのトピックの表示

DB2 インフォメーション・センターでは、ブラウザの設定で指定した言語でのトピックの表示が試みられます。トピックがその指定言語に翻訳されていない場合は、DB2 インフォメーション・センターでは英語でトピックが表示されます。

• Internet Explorer Web ブラウザーで、指定どおりの言語でトピックを表示するには、以下のようにします。

1. Internet Explorer の「ツール」->「インターネット オプション」->「言語...」ボタンをクリックします。「言語の優先順位」ウィンドウがオープンします。
2. 該当する言語が、言語リストの先頭の項目に指定されていることを確認します。
 - リストに新しい言語を追加するには、「追加...」ボタンをクリックします。

注: 言語を追加しても、特定の言語でトピックを表示するのに必要なフォントがコンピューターに備えられているとはかぎりません。

- リストの先頭に新しい言語を移動するには、その言語を選択してから、その言語が言語リストに先頭に行くまで「上に移動」ボタンをクリックします。
 3. ブラウザー・キャッシュを消去してから、ページを最新表示します。希望する言語で DB2 インフォメーション・センターが表示されます。
- Firefox または Mozilla Web ブラウザーの場合に、希望する言語でトピックを表示するには、以下のようにします。

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. ブラウザー・キャッシュを消去してから、ページを最新表示します。希望する言語で DB2 インフォメーション・センターが表示されます。

ブラウザとオペレーティング・システムの組み合わせによっては、オペレーティング・システムの地域の設定も希望のロケールと言語に変更しなければならない場合があります。

コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの更新

DB2 インフォメーション・センターをローカルにインストールしている場合は、IBM から資料の更新を入手してインストールすることができます。

ローカルにインストールされた DB2 インフォメーション・センターを更新するには、以下のことを行う必要があります。

1. コンピューター上の DB2 インフォメーション・センターを停止し、インフォメーション・センターをスタンドアロン・モードで再始動します。インフォメーション・センターをスタンドアロン・モードで実行すると、ネットワーク上の他のユーザーがそのインフォメーション・センターにアクセスできなくなります。これで、更新を適用できるようになります。非管理者および非 root の DB2 インフォメーション・センターは常にスタンドアロン・モードで実行されます。を参照してください。
2. 「更新」機能を使用することにより、どんな更新が利用できるかを確認します。インストールする更新がある場合は、「更新」機能を使用してそれを入手およびインストールできます。

注: ご使用の環境において、インターネットに接続されていないマシンに DB2 インフォメーション・センターの更新をインストールする必要がある場合は、インターネットに接続されていて DB2 インフォメーション・センターがインストールされているマシンを使用して、更新サイトをローカル・ファイル・システムにミラーリングする必要があります。ネットワーク上の多数のユーザーが資料の更新をインストールする場合にも、更新サイトをローカルにミラーリングして、更新サイト用のプロキシを作成することにより、個々のユーザーが更新を実行するのに要する時間を短縮できます。

更新パッケージが入手可能な場合、「更新」機能を使用してパッケージを入手します。ただし、「更新」機能は、スタンドアロン・モードでのみ使用できます。

3. スタンドアロンのインフォメーション・センターを停止し、コンピューター上の DB2 インフォメーション・センターを再開します。

注: Windows Vista の場合、下記のコマンドは管理者として実行する必要があります。完全な管理者特権でコマンド・プロンプトまたはグラフィカル・ツールを起動するには、ショートカットを右クリックしてから、「管理者として実行」を選択します。

コンピューターまたはイントラネット・サーバーにインストール済みの 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/db2icdv95 stop
```
2. インフォメーション・センターをスタンドアロン・モードで開始します。
 - Windows の場合:
 - a. コマンド・ウィンドウを開きます。
 - b. インフォメーション・センターがインストールされているパスにナビゲートします。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 9.5 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/V9.5 ディレクトリーにインストールされています。
- b. インストール・ディレクトリーから doc/bin ディレクトリーにナビゲートします。
- c. 次のように help_start スクリプトを実行します。

```
help_start
```

システムのデフォルト Web ブラウザーが起動し、スタンドアロンのインフォメーション・センターが表示されます。

3. 「更新」ボタン (🔄) をクリックします。インフォメーション・センターの右側のパネルで、「更新の検索 (Find Updates)」をクリックします。既存の文書に対する更新のリストが表示されます。
4. インストール・プロセスを開始するには、インストールする更新をチェックして選択し、「更新のインストール」をクリックします。
5. インストール・プロセスが完了したら、「完了」をクリックします。
6. 次のようにして、スタンドアロンのインフォメーション・センターを停止します。

- Windows の場合は、インストール・ディレクトリーの doc¥bin ディレクトリーにナビゲートしてから、次のように help_end.bat ファイルを実行します。

```
help_end.bat
```

注: help_end バッチ・ファイルには、help_start バッチ・ファイルを使用して開始したプロセスを安全に終了するのに必要なコマンドが含まれています。help_start.bat は、Ctrl-C や他の方法を使用して終了しないでください。

- Linux の場合は、インストール・ディレクトリーの doc/bin ディレクトリーにナビゲートしてから、次のように help_end スクリプトを実行します。

```
help_end
```

注: help_end スクリプトには、help_start スクリプトを使用して開始したプロセスを安全に終了するのに必要なコマンドが含まれています。他の方法を使用して、help_start スクリプトを終了しないでください。

7. DB2 インフォメーション・センターを再開します。

- 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/db2icdv95 start
```

更新された DB2 インフォメーション・センターに、更新された新しいトピックが表示されます。

DB2 チュートリアル

DB2 チュートリアルは、DB2 製品のさまざまな機能について学習するのを支援します。この演習をとおして段階的に学習することができます。

はじめに

インフォメーション・センター (<http://publib.boulder.ibm.com/infocenter/db2help/>) から、このチュートリアルの XHTML 版を表示できます。

演習の中で、サンプル・データまたはサンプル・コードを使用する場合があります。個々のタスクの前提条件については、チュートリアルを参照してください。

DB2 チュートリアル

チュートリアルを表示するには、タイトルをクリックします。

「*pureXML* ガイド」の『**pureXML™**』

XML データを保管し、ネイティブ XML データ・ストアに対して基本的な操作を実行できるように、DB2 データベースをセットアップします。

「*Visual Explain* チュートリアル」の『**Visual Explain**』

Visual Explain を使用して、パフォーマンスを向上させるために SQL ステートメントを分析し、最適化し、調整します。

DB2 トラブルシューティング情報

DB2 製品を使用する際に役立つ、トラブルシューティングおよび問題判別に関する広範囲な情報を利用できます。

DB2 ドキュメンテーション

トラブルシューティング情報は、DB2 問題判別ガイド、または DB2 インフォメーション・センターの「サポートおよびトラブルシューティング」セクションにあります。ここには、DB2 診断ツールおよびユーティリティーを使用して、問題を切り分けて識別する方法、最も頻繁に起こる幾つかの問題に対するソリューションについての情報、および DB2 製品を使用する際に発生する可能性のある問題の解決方法についての他のアドバイスがあります。

DB2 Technical Support の Web サイト

現在問題が発生していて、考えられる原因とソリューションを検索したい場合は、DB2 Technical Support の Web サイトを参照してください。

Technical Support サイトには、最新の DB2 資料、TechNotes、プログラム診断依頼書 (APAR またはバグ修正)、フィックスパック、およびその他のリソースへのリンクが用意されています。この知識ベースを活用して、問題に対する有効なソリューションを探し出すことができます。

DB2 Technical Support の Web サイト (<http://www.ibm.com/software/data/db2/udb/support.html>) にアクセスしてください。

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。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒106-8711
東京都港区六本木 3-2-12
日本アイ・ビー・エム株式会社
法務・知的財産
知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。 IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書は、IBM 以外の Web サイトおよびリソースへのリンクまたは参照を含む場合があります。IBM は、本書より参照もしくはアクセスできる、または本書からリンクされた IBM 以外の Web サイトもしくは第三者のリソースに対して一切の責任を負いません。IBM 以外の Web サイトにリンクが張られていることにより IBM が当該 Web サイトを推奨するものではなく、またその内容、使用もしくはサイトの所有者について IBM が責任を負うことを意味するものではありません。また、IBM は、お客様が IBM Web サイトから第三者の存在を知ることになった場合にも (もしくは、IBM Web サイトから第三者へのリンクを使用した場合にも)、お客様と第三者との間のいかなる取引に対しても一切責任を負いません。従って、お客様は、IBM が上記の外部サイトまたはリソースの利用について責任を負うものではなく、また、外部サイトまたはリソースからアクセス可能なコンテンツ、サービス、

製品、またはその他の資料一切に対して IBM が責任を負うものではないことを承諾し、同意するものとします。第三者により提供されるソフトウェアには、そのソフトウェアと共に提供される固有の使用条件が適用されます。

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

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生した創作物には、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。 © Copyright IBM Corp. _年を入れる_. All rights reserved.

商標

以下は、International Business Machines Corporation の米国およびその他の国における商標です。

pureXML	OS/390
DB2 Connect	Passport Advantage
分散リレーショナル・データベース体系	z/OS
Redbooks	System i
IBM	DB2
zSeries	AIX
Tivoli	DRDA
ibm.com	i5/OS
iSeries	

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。
 - Windows は、Microsoft Corporation の米国およびその他の国における商標です。
- 他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

意思決定支援システム (DSS) 88
イベント・モニター
 トラブルシューティング 10
インストール
 エラー・ログ 43
 DB2 製品のリスト 68
インストールの問題
 トラブルシューティング 32
 分析 46
エラー
 トラブルシューティング 49
エラー・メッセージ
 DB2 Connect 52
オペレーティング・システム
 トラブルシューティング・ツール 105

[カ行]

監査機能
 トラブルシューティング 10
管理通知ログ 6, 13
 解釈 13
 紹介 12
グローバル変数
 トラブルシューティング 39
グローバル・レジストリー
 変更 64
現行リリース
 トラブルシューティング 31
検索
 手法 109
コア・ファイル
 問題判別 52
 Linux システム 24
 UNIX システム 24
交換サーバー属性コマンド 90
更新
 DB2 インフォメーション・センター 123
コマンド
 コミット 90
 ACCRDB 90
 ACCRDBRM 90
 ACCSEC 90
 db2dart 59

コマンド (続き)
 db2diag 62
 db2drdat 88
 db2inspf 42
 db2level 64
 db2look 65
 db2ls 68
 db2pd 19
 例 70
 db2pdcfg 4
 db2support 26, 80
 db2trc 84, 86
 EXCSAT 90
 EXCSATRD 90
 INSPECT 59
 SECCHK 90
コマンド・エディター
 トラブルシューティング 11
コミット・コマンド
 トレース出力バッファ 90
ご利用条件
 資料の使用 126
コントロール・センター
 トレース 96
コンプレッション・ディクショナリー
 作成されない 38

[サ行]

最終作業単位の応答メッセージ (ENDUOWRM) 90
最適化
 ガイドライン
 トラブルシューティング 34
最適化プロファイル
 トラブルシューティング 34
索引データの整合性 42
作成
 データベース 34
システム・コア・ファイル
 識別 24
 Linux 24
 UNIX 24
システム・コマンド
 dbx (UNIX) 24
 gdb (Linux) 24
 xdb (HP-UX) 24
受信バッファ 88
資料
 印刷 118
 注文 120
 概要 117

資料 (続き)

- 使用に関するご利用条件 126
- PDF 118
- 診断情報
 - アプリケーション 23
 - インスタンス管理の問題 45
 - インストールの問題 43
 - 概要 43, 52
 - データ移動の問題 44
 - ハードウェア 23
 - 分析 46
 - ログ・ファイル 15
 - ワトソン博士のログ 26
 - DB2 管理サーバー (DAS) の問題 45
 - First Occurrence Data Capture (FODC)
 - 構成 4
 - 説明 4
 - ファイル 6
 - IBM ソフトウェア・サポートへの送信 47
- Linux
 - システム・コア・ファイル 24
 - 情報の取得 23
 - 診断ツール 105
- UNIX
 - システム・コア・ファイル 24
 - 情報の取得 23
 - 診断ツール 105
- Windows
 - イベント・ログ 25
 - 情報の取得 23
 - 診断ツール 105
- 診断データの分析 46
- スクリプト
 - トラブルシューティング 42
- ストレージ・キー
 - トラブルシューティング 37
- スレッド 42
- 送信バッファ
 - トレース・データ 88

[タ行]

- ダンプ・ファイル
 - エラー・レポート 21
- チュートリアル
 - トラブルシューティング 125
 - 問題判別 125
 - Visual Explain 125
- ツール
 - 診断
 - 概要 105
 - Windows 105
- 通知レベル構成パラメーター
 - 更新 13
- データ
 - 不整合 42

- データウェアハウス・センター
 - トラブルシューティング 11
 - データベース
 - 名前
 - RDBNAM オブジェクト 90
 - 破損 42
 - データベース分析およびレポート・ツール・コマンド
 - 概要 59
 - データベース・エンジンのプロセス 42
 - テスト・フィックス
 - 説明 113
 - タイプ 115
 - 適用 115
 - 特記事項 127
 - トラップ・ファイル 22
 - フォーマット (Windows) 22
 - トラブルシューティング 1, 26
 - インストールの問題 32
 - オンライン情報 125
 - 概要 1, 49
 - 現行リリース 31
 - 高可用性の問題 32
 - 作成
 - データベース 34
 - 紹介 1
 - 情報の収集 43, 47, 49, 64, 70, 80
 - 診断データ
 - インストールに関する 43
 - 基本セットの収集 43
 - 自動収集 4
 - 収集の構成 4
 - 手動での収集 4
 - データ移動に関する 44
 - DAS またはインスタンス管理に関する 45
 - ストレージ・キー 37
 - 接続 49, 50
 - 説明 31
 - チュートリアル 125
 - ツール 57
 - トレース機能 83, 84
 - コントロール・センターのトレース 96
 - CLI および ODBC アプリケーション 99, 100
 - DRDA 91, 95
 - JDBC アプリケーション 96, 98
 - 問題の解決策の検索 109
 - 問題の再現 65
 - リソース 110
 - DB2 Connect 52
- トレース
 - 概要 83
 - 出力ファイル 88, 89
 - 出力ファイル・サンプル 91
 - CLI 99
 - 分析 102, 103, 104
 - DB2 Connect とサーバーの間のデータ 88

トレース (続き)
DRDA
 解釈 88
DRDA トレースのバッファ情報 95
トレース機能
 コントロール・センターのトレース 96
 トラブルシューティングの概要 83
CLI アプリケーション 100
DB2 トレース 84, 86
DRDA トレース 91, 95
JDBC アプリケーション 98
 トレース・オプションの構成 96
トレース・ユーティリティ (db2drdat) 88

[ハ行]

パーティション・データベース
 トラブルシューティング 33
パラメーター
 PRDID 90
ファイルへのトレースのダンプ
 概要 86
フィックスパック
 説明 113
 適用 111
 入手 111
プログラム診断依頼書 (APAR) 113
プロセス状況ユーティリティ 52, 90
分散データ管理 (DDM)
 db2drdat 出力 88
分散リレーショナル・データベース体系 (DRDA)
 トレース 88
ヘルプ
 言語の構成 122
 SQL ステートメント 121
本書について v

[マ行]

戻りコード
 内部 57
問題判別
 診断ツール
 概要 52
 接続 49
 接続後 50
 チュートリアル 125
 利用できる情報 125

[ヤ行]

ユーティリティ
 トレース 88
 プロセス状況 90
 db2drdat 88

ユーティリティ (続き)
 ps (プロセス状況) 52, 90

[ラ行]

ログ・ファイル
 管理 13

[ワ行]

ワークロード管理
 トラブルシューティング 41

A

ACCRDB コマンド 90
ACCRDBRM コマンド 90
ACCSEC コマンド 90

C

CLI (コール・レベル・インターフェース)
 アプリケーション
 トレース機能の構成 100
 トレース
 トラブルシューティングの概要 99
 トレース機能
 開始 100

D

DB2 Connect
 トラブルシューティング 49
DB2 JDBC Type 2 ドライバー
 トレース機能の構成 96
DB2 Universal JDBC ドライバー
 トレース機能の構成 98
DB2 インフォメーション・センター
 言語 122
 更新 123
 バージョン 121
 別の言語で表示する 122
DB2 ガバナー
 トラブルシューティング 10
DB2 資料の印刷方法 120
DB2 製品
 リスト 68
DB2 トレース 84
db2cli.ini ファイル
 トレース構成 100
db2cos スクリプト
 出力ファイル 19
db2dart コマンド
 トラブルシューティングの概要 59

db2dart コマンド (続き)
 INSPECT コマンドの比較 59

db2diag コマンド
 例 62

db2diag.log ファイル
 解釈
 概要 16
 情報レコード 15
 db2diag ツールの使用 62
 管理通知ログ・メッセージ 13
 紹介 15
 First Occurrence Data Capture (FODC) 情報 6

db2drdat コマンド
 出力ファイル 88

DB2FODC レジストリー変数
 診断情報の収集 4

db2inspf コマンド
 トラブルシューティング 42

db2level コマンド
 サービス・レベルの識別 64
 バージョン・レベルの識別 64

db2look コマンド
 データベースの作成 65

db2ls コマンド
 インストールされている製品およびフィーチャーのリスト表示 68

db2pd コマンド
 デフォルトの db2cos スクリプトによって収集される出力 19
 トラブルシューティングの例 70

db2pdcfg コマンド
 DB2FODC レジストリー変数でのオプションの設定 4

db2support コマンド
 実行 26
 説明 80

db2trc コマンド
 概要 84
 トレース出力のダンプ 86
 トレース出力のフォーマット 86

ddcstrc ユーティリティ 89

DDM (分散データ管理)
 db2drdat 出力 88

diaglevel 構成パラメーター
 更新 15

E

ECF 戻りコード
 概要 57

EXCSAT コマンド 90

EXCSATRD コマンド 90

EXTNAM オブジェクト 90

F

FCM 問題のトラブルシューティング 33

First Failure Data Capture (FFDC)
 トラップ・ファイル 22

First Occurrence Data Capture
 紹介 3

First Occurrence Data Capture (FODC)
 サブディレクトリー 5
 説明 6
 ダンプ・ファイル 21
 データの生成 6
 トラップ・ファイル 22
 プラットフォーム固有の 23

I

IBM と連絡をとる 47

INSPECT CHECK 42

INSPECT コマンド 59

J

Java Database Connectivity (JDBC)
 アプリケーション
 トレース機能の構成 96, 98
 トレース 96

L

Linux
 DB2 製品のリスト 68

O

ODBC (Open Database Connectivity)
 アプリケーション
 トレース機能の構成 100

P

PRDID パラメーター 90

ps コマンド
 概要 52
 EXTNAM オブジェクト 90

Q

Query Patroller
 トラブルシューティング 9

S

SECCHK コマンド 90

SQL ステートメント
ヘルプを表示する 121
SQL0965 エラー・コード 52
SQL0969 エラー・コード 52
SQL1338 エラー・コード 52
SQL30020 エラー・コード 52
SQL30060 エラー・コード 52
SQL30061 エラー・コード 52
SQL30073 エラー・コード 52
SQL30081N エラー・コード 52
SQL30082 エラー・コード 52
SQL5043N エラー・コード 52
SQLCA (SQL 連絡域)
データのバッファ 88
SQLCODE フィールド 88
SQLCODE
SQLCA 内のフィールド 88
SRVNAM オブジェクト 90

T

TCP/IP
ACCSEC コマンド 90
SECCHK コマンド 90
Tivoli System Automation for Multiplatforms
高可用性 32

U

UNIX
DB2 製品のリスト 68

V

Visual Explain
チュートリアル 125

Z

ZRC 戻りコード
説明 57



Printed in Japan

GI88-4108-01



日本アイ・ビー・エム株式会社
〒106-8711 東京都港区六本木3-2-12

Spine information:

DB2 Version 9.5 for Linux, UNIX, and Windows

問題判別ガイド

