

**IBM DB2 10.1
for Linux, UNIX, and Windows**

**ADO.NET および OLE DB ア
プリケーションの開発**
2013 年 1 月更新版

IBM

IBM DB2 10.1
for Linux, UNIX, and Windows

**ADO.NET および OLE DB ア
プリケーションの開発**
2013 年 1 月更新版



ご注意

本書および本書で紹介する製品をご使用になる前に、99ページの『付録 B. 特記事項』に記載されている情報をお読みください。

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

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

- オンラインで資料を注文するには、IBM Publications Center (<http://www.ibm.com/shop/publications/order>) 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、IBM Directory of Worldwide Contacts (<http://www.ibm.com/planetwide/>) をお調べください。

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

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

原典： SC27-3873-01
IBM DB2 10.1
for Linux, UNIX, and Windows
Developing ADO.NET and OLE DB
Applications
Updated January, 2013

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第1刷 2012.12

© Copyright IBM Corporation 2006, 2013.

目次

第 1 章 ADO.NET アプリケーション開発 1

.NET アプリケーションのデプロイ (Windows)	2
サポートされる .NET 開発ソフトウェア	3
Visual Studio での DB2 統合	4
IBM Data Server Provider for .NET	5
IBM Data Server Provider for .NET のデータベース・システム要件	6
IBM Data Server Provider for .NET を使用するためのアプリケーションのプログラミング	7
.NET アプリケーションの構築	24
OLE DB .NET Data Provider	30
OLE DB .NET Data Provider の制約事項	31
ヒント	35
ODBC .NET Data Provider	38
ODBC .NET Data Provider の制約事項	38

第 2 章 IBM OLE DB Provider for DB2 47

IBM OLE DB Provider for DB2 でサポートされているアプリケーション・タイプ	48
OLE DB サービス	48
IBM OLE DB Provider でサポートされているスレッド・モデル	48
IBM OLE DB Provider によるラージ・オブジェクトの操作	48
IBM OLE DB Provider でサポートされているスキーマ行セット	49
IBM OLE DB Provider で自動的に使用可能になる OLE DB サービス	51
データ・サービス	52
IBM OLE DB Provider でサポートされているカーソル・モード	52
DB2 と OLE DB の間のデータ・タイプ・マッピング	52
OLE DB タイプから DB2 タイプにデータを設定するためのデータ変換	53
DB2 タイプから OLE DB タイプにデータを設定するためのデータ変換	56
IBM OLE DB Provider の制約事項	59
IBM OLE DB Provider での OLE DB コンポーネントおよびインターフェースのサポート	60
IBM OLE DB Provider での OLE DB プロパティのサポート	63
IBM OLE DB Provider によるデータ・ソースへの接続	67

ADO アプリケーション	68
ADO 接続ストリング・キーワード	68
Visual Basic ADO アプリケーションによるデータ・ソースへの接続	69
ADO アプリケーションにおける更新可能な両方向スクロール・カーソル	69
ADO アプリケーションの制限	69
IBM OLE DB Provider での ADO メソッドおよびプロパティのサポート	70
C/C++ アプリケーションのコンパイルおよびリンクと IBM OLE DB Provider	76
IBM OLE DB Provider による、C/C++ アプリケーションでのデータ・ソースへの接続	76
COM+ 分散トランザクションのサポートと IBM OLE DB Provider	76
C/C++ データベース・アプリケーションでの COM+ サポートの使用可能化	77

第 3 章 IBM Data Server Provider for .NET 79

IBM.Data.DB2 ネーム・スペース	80
IBM.Data.DB2Types ネーム・スペース	84

付録 A. DB2 技術情報の概説 87

DB2 テクニカル・ライブラリー (ハードコピーまたは PDF 形式)	88
コマンド行プロセッサから SQL 状態ヘルプを表示する	90
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	91
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	91
コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの手動更新	93
DB2 チュートリアル	95
DB2 トラブルシューティング情報	96
ご利用条件	96

付録 B. 特記事項 99

索引 103

第 1 章 ADO.NET アプリケーション開発

.NET Framework の主な機能と、COM テクノロジーが .NET Framework にどのように置き換わったかを中心に、以下で説明しています。

近年、Microsoft では Windows 用の新規のソフトウェア開発プラットフォームの普及を促進してきました。これは、.NET Framework として知られています。 .NET Framework は、Microsoft の Component Object Model (COM) テクノロジーに代わるものです。 主要な .NET Framework 機能は以下のとおりです。

- .NET アプリケーションを 40 以上の異なるプログラミング言語でコーディングできます。 .NET 開発用の最も一般的な言語は C# および Visual Basic .NET です。
- .NET Framework クラス・ライブラリーは、.NET アプリケーションの構築に使用する構築ブロックを提供します。 このクラス・ライブラリーは言語に左右されず、オペレーティング・システムおよびアプリケーション・サービスへのインターフェースを提供します。
- ご使用の .NET アプリケーション (言語に関係なく) は、バイトコードの一種である中間言語 (IL) にコンパイルされます。
- Common Language Runtime (CLR) は .NET Framework の核心部で、IL コードをオンザフライにコンパイルし、それを実行します。 コンパイルした IL コードを実行すると、CLR はオブジェクトを活動化し、セキュリティ許可を検証し、メモリーを割り振り、それらのオブジェクトを実行します。実行が完了したならメモリーをクリーンアップします。

これらの機能を使用して、.NET Framework は広範囲に及ぶアプリケーション・インプリメンテーション (例えば、Windows フォーム、Web フォーム、および Web サービスなど)、迅速なアプリケーション開発、および安全なアプリケーション・デプロイメントを容易にします。 COM および COM+ は、前述のどの機能を実行する場合にも不十分、または煩雑であることが分かっています。

.NET Framework は、ADO.NET を介して大規模なデータ・アクセス・サポートを提供します。 ADO.NET は接続アクセスと切断アクセスの両方をサポートします。 ADO.NET での切断データ・アクセスのキー・コンポーネントは DataSet クラスです。このクラスのインスタンスは、アプリケーションのメモリー内にあるデータベース・キャッシュの役割をします。

接続アクセスと切断アクセスのどちらの場合でも、アプリケーションはデータ・プロバイダーとして知られるものを介してデータベースを使用します。 さまざまなデータベース製品 (DB2[®] for Windows を含む) には、独自の .NET データ・プロバイダーが含まれています。

.NET データ・プロバイダーは、以下の基本クラスのインプリメンテーションを特色としています。

- **Connection:** データベース接続を確立および管理します。
- **Command:** SQL ステートメントをデータベースに対して実行します。
- **DataReader:** データベースから結果セット・データを読み取って、戻します。

- DataAdapter: DataSet インスタンスをデータベースにリンクします。DataAdapter インスタンスを介して、DataSet はデータベース表データの読み取りと書き込みを行えます。

Microsoft では、2 つのデータ・プロバイダー (OLE DB .NET Data Provider および ODBC .NET Data Provider) を提供しています。OLE DB .NET Data Provider は、ADO.NET 要求を IBM® OLE DB Provider に送信する (COM interop モジュールによって) ブリッジ・プロバイダーです。ODBC .NET Data Provider は、ADO.NET 要求を IBM ODBC ドライバーに送信するブリッジ・プロバイダーです。これらの .NET データ・プロバイダーは、DB2 ファミリー・データベースへのアクセスにはお勧めしません。IBM Data Server Provider for .NET は、ハイパフォーマンスの、管理された ADO.NET データ・プロバイダーです。DB2 ファミリー・データベースとともに使用する場合は、この .NET データ・プロバイダーがお勧めです。IBM Data Server Provider for .NET を使用する ADO.NET データベース・アクセスは、OLE DB および ODBC .NET ブリッジ・プロバイダーと比べて、制限が少なく、極めて良好なパフォーマンスを提供します。

.NET アプリケーションのデプロイ (Windows)

.NET アプリケーションのデプロイメントを簡単に行えるように、IBM では IBM Data Server Driver Package を用意しています。これは、大規模なデプロイメント・シナリオで使用するのに理想的な、スモール・フットプリント・クライアントです。

IBM Data Server Driver Package のフィーチャー以外の追加フィーチャーが必要な場合は、代わりとして IBM Data Server Runtime Client を使用できます。

始める前に

- デプロイメントの前に、.NET アプリケーションを構築する必要があります。これは、Visual Studio またはコマンド行のいずれかを使用して行います。.NET アプリケーションの構築について詳しくは、関連タスクを参照してください。
- .NET アプリケーションの構築に使用するコンピューター、また .NET アプリケーションをデプロイするコンピューターには、3 ページの『サポートされる .NET 開発ソフトウェア』で説明されているように、サポートされるバージョンの Windows オペレーティング・システムおよび他のソフトウェアが必要です。
 - 構築システム
 - Windows オペレーティング・システム
 - Visual Studio
 - .NET Framework 再頒布可能パッケージ
 - .NET Framework Software Development Kit
 - デプロイメント・システム
 - Windows オペレーティング・システム
 - .NET Framework 再頒布可能パッケージ

手順

.NET アプリケーションをデプロイするには、以下のようにします。

1. IBM Data Server Driver Package を、アプリケーションのデプロイ先のコンピューターにインストールします。インストール中に、IBM Data Server Driver Package のインストールをデフォルト・データベース・クライアント・インターフェース・コピーに設定します。

注: IBM データ・サーバーに対して実行される既存のデータベース・アプリケーションがあれば、それらはこの新しくインストールされた IBM Data Server Driver Package を使用します。デプロイした .NET アプリケーションをロールアウトする前に、それらのアプリケーションを新規ドライバーに対してテストします。

2. 構築したアプリケーションを、それが実行されるコンピューターにインストールします。

サポートされる .NET 開発ソフトウェア

IBM データ・サーバーに対して実行する .NET アプリケーションを開発してデプロイするには、サポートされている開発ソフトウェアとオペレーティング・システムを使用する必要があります。

.NET Framework 2.0、3.0、3.5、および 4.0 アプリケーションの開発とデプロイのためにサポートされているオペレーティング・システム

以下のオペレーティング・システムが、.NET Framework 2.0、3.0、3.5、および 4.0 アプリケーションの開発とデプロイでサポートされます。

- Windows XP、Service Pack 2 (32 ビット版、64 ビット版)
- Windows Server 2003 (32 ビット版、64 ビット版)
- Windows Vista (32 ビット版、64 ビット版)
- Windows Server 2008 (32 ビット版、64 ビット版)
- Windows Server 2008 R2 (64 ビット版)
- Windows 7 (32 ビット版、64 ビット版)

.NET Framework 4.5 アプリケーションの開発およびデプロイでサポートされるオペレーティング・システム

以下のオペレーティング・システムが、.NET Framework 4.5 アプリケーションの開発とデプロイでサポートされます。

- Windows Vista Service Pack 2 (32 ビット版、64 ビット版)
- Windows Server 2008 Service Pack 2 (32 ビット版、64 ビット版)
- Windows Server 2008 R2 (64 ビット版)
- Windows 7 (32 ビット版、64 ビット版)

Microsoft .NET Framework 4.5 は、バージョン 10.1 フィックスパック 2 以降のフィックスパックでサポートされます。

.NET Framework アプリケーション用にサポートされる開発ソフトウェア

.NET Framework アプリケーションを開発するには、IBM Data Server Clientまたはドライバー・パッケージに加えて、以下のサポートされるツールのいずれかが必要です。

- Visual Studio 2008
- Visual Studio 2010
- Visual Studio 2012

Microsoft Visual Studio 2012 は、バージョン 10.1 フィックスパック 2 以降のフィックスパックでサポートされています。

.NET Framework アプリケーション用にサポートされるデプロイメント・ソフトウェア

.NET Framework アプリケーションをデプロイするには、IBM Data Server Clientまたはドライバー・パッケージに加えて、以下のパッケージのいずれかが必要です。ほとんどの場合、.NET Framework 再頒布可能パッケージは、Windows のインストール済み環境に組み込まれています。

- .NET Framework Version 2.0 再頒布可能パッケージ
- .NET Framework Version 3.0 再頒布可能パッケージ
- .NET Framework Version 3.5 再頒布可能パッケージ
- .NET Framework Version 4.0 再頒布可能パッケージ
- .NET Framework Version 4.5 再頒布可能パッケージ

64 ビット IBM Data Server Package をインストールすると、32 ビットおよび 64 ビットの両方のプロバイダーがインストールされて構成されます。

.NET Framework がインストールされていないと、IBM Data Server Client およびドライバー・インストーラーによって IBM Data Server Provider for .NET がインストールされることはありません。IBM Data Server Provider for .NET を手動でインストールする必要があります。

Visual Studio での DB2 統合

IBM Database Add-Ins for Visual Studio は、Visual Studio 開発環境にシームレスに統合するための機能の集合で、これにより DB2 サーバーと連携して、DB2 プロシージャ、関数、およびオブジェクトを開発できます。

IBM Database Add-Ins for Visual Studio は、DB2 データベースに簡単なインターフェースを提供します。例えば、SQL を使用するのではなく、デザイナーとウィザードを使用してデータベース・オブジェクトの作成を行うことができます。また SQL コードを作成する必要がある場合には、組み込まれている DB2 SQL エディターに以下の機能が備えられています。

- より読みやすくするために色付けされた SQL テキスト
- DB2 スクリプトの入力時のインテリジェント・オートコンプリートを提供するための Microsoft Visual Studio IntelliSense 機能との統合

IBM Database Add-Ins for Visual Studio を使用すると、以下のことが可能です。

- 各種の DB2 開発ツールや管理ツールのオープン
- ソリューション・エクスプローラーでの DB2 プロジェクトの作成および管理
- サーバー・エクスプローラーからの DB2 データ接続のアクセスと管理
- ストアード・プロシージャ、関数、表、ビュー、索引、およびトリガーを作成するスクリプトなどの DB2 スクリプトの作成および変更

Visual Studio 2008 および 2010

IBM Database Add-Ins for Visual Studio は、インストール可能な別個のコンポーネントとして DB2 Client および DB2 サーバーに組み込まれています。DB2 製品のインストールを終えると、IBM Database Add-Ins for Visual Studio をインストールするためのオプションが表示されます。コンピュータ上に Visual Studio をインストールしていない場合には、このアドインはインストールされません。Visual Studio をインストールすると、DB2 製品のセットアップ・メニューからいつでもアドインをインストールできます。

迅速なアプリケーション開発を可能にするために IBM Database Add-In と Data Server Provider for .NET を使用する方法については、IBM Information Management and Visual Studio .NET ゾーン (<http://www.ibm.com/developerworks/data/zones/vstudio/index.html>) にアクセスしてください。

IBM Data Server Provider for .NET

IBM Data Server Provider for .NET は、ADO.NET インターフェースのデータ・サーバー・サポートを拡張するものです。このプロバイダーは、IBM データ・サーバーへのハイ・パフォーマンスでセキュアなアクセスを提供します。

2 つのプロバイダーは、いずれも IBM Data Server Provider for .NET クライアント・パッケージに含まれています。それらのプロバイダーは、しばしば Common .NET Provider と呼ばれます。

DB2 .NET Provider (IBM.Data.DB2.dll)

DB2 .NET Provider を使用すると、.NET アプリケーションにおいて、以下のデータベース管理システムにアクセスできます。

- DB2 for Linux, UNIX, and Windows、バージョン 9.1、バージョン 9.5、バージョン 9.7、バージョン 9.8、およびバージョン 10.1
- DB2 Universal Database™ Version 8 for Windows, UNIX, and Linux
- DB2 for z/OS® バージョン 8、バージョン 9、およびバージョン 10 (DB2 Connect™ 経由)
- IBM DB2 for IBM i バージョン 5 リリース 4、バージョン 6 リリース 1、およびバージョン 7 リリース 1 (DB2 Connect 経由) (IBM DB2 バージョン 9.7 フィックス・パック 4 以上のバージョン用)
- IBM DB2 for IBM i バージョン 5 リリース 4 およびバージョン 6 リリース 1 (DB2 Connect 経由) (IBM DB2 バージョン 9.7 フィックス・パック 3 以前のバージョン用)
- IBM Informix® バージョン 11.10 以降

これらのトピックの残りの部分では、Common DB2 .NET Provider について説明します。

Informix データベース・サーバー .NET Provider (IBM.Data.Informix.dll)

Informix データベース・サーバー .NET Provider を使用すると、.NET アプリケーションにおいて、以下のデータベース管理システムにアクセスできます。

- IBM Informix バージョン 11.10 以降

このプロバイダーについて詳しくは、IBM Informix Dynamic Server Information Center を参照してください。

Data Server Provider for .NET を使用するアプリケーションを開発および実行するには、.NET Framework が必要です。

IBM Data Server Provider for .NET に加えて、IBM Database Add-Ins for Visual Studio を使用すると、Microsoft Visual Studio を使用することによって、IBM データ・サーバー用の .NET アプリケーションを短時間で簡単に開発できます。このアドインを使用して、索引や表などのデータベース・オブジェクトを作成し、ストアード・プロシージャやユーザー定義関数などのサーバー・サイド・オブジェクトを開発することができます。

IBM Data Server Provider for .NET のデータベース・システム要件

アプリケーションは、IBM Data Server Provider for .NET を使用して、IBM データベース・システムにアクセスします。

DB2 .NET Provider (IBM.Data.DB2.dll)

DB2 .NET Provider を使用すると、.NET アプリケーションにおいて、以下のデータベース管理システムにアクセスできます。

- DB2 for Linux, UNIX, and Windows、バージョン 9.1、バージョン 9.5、バージョン 9.7、バージョン 9.8、およびバージョン 10.1
- DB2 Universal Database Version 8 for Windows, UNIX, and Linux
- DB2 for z/OS バージョン 8、バージョン 9、およびバージョン 10 (DB2 Connect 経由)
- IBM DB2 for IBM i バージョン 5 リリース 4、バージョン 6 リリース 1、およびバージョン 7 リリース 1 (DB2 Connect 経由) (IBM DB2 バージョン 9.7 フィックス・パック 4 以上のバージョン用)
- IBM DB2 for IBM i バージョン 5 リリース 4 およびバージョン 6 リリース 1 (DB2 Connect 経由) (IBM DB2 バージョン 9.7 フィックス・パック 3 以前のバージョン用)
- IBM Informix バージョン 11.10 以降

これらのトピックの残りの部分では、Common DB2 .NET Provider について説明します。

Informix データベース・サーバー .NET Provider (IBM.Data.Informix.dll)

Informix データベース・サーバー .NET Provider を使用すると、.NET アプリケーションにおいて、以下のデータベース管理システムにアクセスできます。

- IBM Informix バージョン 11.10 以降

このプロバイダーについて詳しくは、IBM Informix Dynamic Server Information Center を参照してください。

DB2 for i の場合、サーバー上で APAR II13348 の修正が必要です。

IBM Data Server Provider for .NET を使用するためのアプリケーションのプログラミング

IBM Data Server Provider for .NET を使用するためのアプリケーションをプログラミングするには、IBM Data Provider for .NET で使用可能なフィーチャーを理解している必要があります。アプリケーションの要件を判別しないと、IBM Data Provider for .NET で使用可能なフィーチャーを活用できるようになりません。

ADO.NET 共通基本クラスを使用した汎用コーディング

.NET Framework バージョン 2.0、3.0、および 3.5 は、System.Data.Common というネーム・スペースを特色としています。これは、任意の .NET データ・プロバイダーが共用できる基本クラスのセットを備えています。これにより、異なる複数のデータベースで共通に使用できる一定のプログラミング・インターフェースを備えた汎用 ADO.NET データベース・アプリケーション開発アプローチが容易になります。

以下の C# は、データベース接続を確立するための汎用の方法を示します。

```
DbProviderFactory factory = DbProviderFactories.GetFactory("IBM.Data.DB2");
DbConnection conn = factory.CreateConnection();
DbConnectionStringBuilder sb = factory.CreateConnectionStringBuilder();

if( sb.ContainsKey( "Database" ) )
{
    sb.Remove( "database" );
    sb.Add( "database", "SAMPLE" );
}

conn.ConnectionString = sb.ConnectionString;

conn.Open();
```

DbProviderFactory オブジェクトは、汎用 ADO.NET アプリケーションの開始点です。このオブジェクトは、.NET データ・プロバイダー・オブジェクトの汎用インスタンス (接続、データ・アダプター、コマンド、およびデータ読取装置など) を作成します。これは、特定のデータベース製品とともに実行します。前述の例において、GetFactory メソッドに渡される "IBM.Data.DB2" スtring は、IBM Data Server Provider for .NET を固有に識別します。そして、DbProviderFactory インスタンスの初期化が実行され、その結果、IBM Data Server Provider for .NET に固有のデータベース・プロバイダー・オブジェクト・インスタンスが作成されます。

DbConnection オブジェクトは、DB2Connection オブジェクト (実際には DbConnection から継承されたもの) と同じように DB2 ファミリーの Informix データベースに接続できます。DbConnectionStringBuilder クラスを使用して、データ・プロバイダーの接続ストリング・キーワードを判別し、カスタム接続ストリングを生成できます。前述の例のコードは、IBM Data Server Provider for .NET 中に

database というキーワードが存在するかどうかをチェックし、存在する場合、SAMPLE データベースに接続するための接続ストリングを生成します。

IBM Data Server Provider for .NET を使用してアプリケーションからデータベースに接続する

IBM Data Server Provider for .NET を使用する場合、データベース接続は DB2Connection クラスによって確立されます。

手順

データベースに接続するには、次のようにします。

1. 接続パラメーターを格納するストリングを作成します。典型的な接続ストリングの形式は、以下のとおりです。

```
Server=<ip address/localhost>:<port number>;
Database=<db name>;
UID=<userID>;
PWD=<password>;
Connect Timeout=<Timeout value>
```

以下は、考えられる接続ストリングの例です。

例 1:

```
String connectionString = "Database=SAMPLE";
// When used, attempts to connect to the SAMPLE database.
```

注: 接続ストリングの中にデータベース名のみを指定する場合、サーバー、ユーザー ID、パスワードなどのその他の情報は db2dsdriver.cfg ファイルの中を含める必要があります。

例 2:

```
String cs = "Server=srv:50000;
Database=SAMPLE;
UID=db2adm;
PWD=ab1d;Connect Timeout=30";
// When used, attempts to connect to the SAMPLE database on the server
// 'srv' through port 50000 using 'db2adm' and 'ab1d' as the user id and
// password respectively. If the connection attempt takes
// more than thirty seconds, the attempt will be terminated and an error
// will be generated.
```

2. connectionString を DB2Connection のコンストラクターに渡します。

- C# でデータベースに接続する

```
String connectionString = "Database=SAMPLE";
DB2Connection conn = new DB2Connection(connectionString);
conn.Open();
return conn;
```

- Visual Basic .NET でデータベースに接続する

```
Dim connectionString As String = "Database=SAMPLE"
Dim conn As DB2Connection = new DB2Connection(connectionString)
conn.Open()
Return conn
```

3. DB2Connection オブジェクトの Open メソッドを使用することによって、connectionString の中で識別されているデータベースに正式に接続します。

IBM Data Server Provider for .NET での接続プーリング

DB2 データベースとの接続が最初にオープンされると、その時点で接続プールが作成されます。接続がクローズされると、それらはプールに入れられ、同じプロセスの中で接続を必要とする他のアプリケーションにより再利用可能な状態になります。

IBM Data Server Provider for .NET では、接続プール判別のために、一連の接続ストリング属性の正規化されたセットが使用されます。正規化属性を使用することにより、アプリケーションが接続を再利用する可能性が大きくなります。

IBM Data Server Provider for .NET では、接続プールがデフォルトで有効になります。

注: Pooling=false という接続ストリング・キーワード/値の対を使用して、接続プーリングをオフにできます。しかし、接続プールの機能をオフにすると、COM+ アプリケーションは動作しなくなります。

以下の接続ストリング・キーワードを設定して、接続プールの動作を制御できます。

- 最小および最大プール・サイズ (**MinPoolSize** および **MaxPoolSize**)
- 接続がアイドル状態になってからプールに戻されるまでの時間 (**ConnectionLifetimeInPool**)

IBM Data Server Provider for .NET を介したトラステッド接続の作成

.NET アプリケーションで接続ストリング・キーワードを使用してトラステッド・コンテキストがサポートされます。

接続ストリングでは、以下のキーワードを使用できます。

- **TrustedContextSystemUserID** または **tcuserid**。これは、接続で使用されるトラステッド・コンテキスト **SYSTEM AUTHID** を指定します。
- **TrustedContextSystemPassword** または **tcspwd**。これは、接続で使用されるトラステッド・コンテキスト **SYSTEM AUTHID** に対応するパスワードを指定します。

TrustedContextSystemUserID キーワード値を使用せずに

TrustedContextSystemPassword キーワードを指定した場合、**InvalidArgument** 例外がスローされます。また、トラステッド・コンテキスト・シナリオでは **UserID** キーワードも必要です。

IBM Data Server Provider for .NET を介したトラステッド・コンテキストは、現在、以下のバージョンでサポートされています。

- DB2 Database for Linux, UNIX, and Windows, バージョン 9.5、バージョン 9.7、バージョン 9.8、およびバージョン 10.1
- DB2 Universal Database for z/OS バージョン 9 およびバージョン 10

例

以下の情報を使用して、トラステッド・コンテキストがサーバー上に確立されたとします。

```
CREATE TRUSTED CONTEXT ctxName1
BASED UPON CONNECTION USING SYSTEM AUTHID masteruser
ATTRIBUTES ( PROTOCOL 'TCPIP',
              ADDRESS '9.26.146.201',
              ENCRYPTION 'NONE' )

ENABLE
WITH USE FOR userapp1 WITH AUTHENTICATION, userapp2 WITH AUTHENTICATION;
```

SYSTEM AUTHID masteruser には、対応するパスワード masterpassword があります。特定のユーザー/アプリケーション userapp1、および userapp2 には、それぞれ対応するパスワード passapp1 および passapp2 があります。

このトラステッド・コンテキストを使用するために、例えば以下のような接続ストリングがアプリケーションによって発行されます。

- アプリケーション 1

```
database=db;server=server1:446;  
UserID=userapp1;Password=passapp1;  
TrustedContextSystemUserID=masteruser;TrustedContextSystemPassword=masterpassword
```

- アプリケーション 2

```
database=db;server=server1:446;  
UserID=userapp2;Password=passapp2;  
TrustedContextSystemUserID=masteruser;TrustedContextSystemPassword=masterpassword
```

注: トラステッド・コンテキスト状態では、標準的なアプリケーションの場合と同じように、キーワード UserID が接続のエンド・ユーザーに対応します。

次の .NET プログラムは、接続のオープンとクローズを行います。

```
[C#]  
DB2Connection conn = new DB2Connection();  
  
conn.ConnectionString = "database=db;server=server1:446;"  
    + "UserID=userapp1;Password=passapp1;"  
    + "TrustedContextSystemUserID=masteruser;"  
    + "TrustedContextSystemPassword=masterpassword;"  
  
conn.Open();  
  
// Do processing as userapp1, such as querying tables  
  
conn.Close();  
  
conn.ConnectionString = "database=db;server=server1:446;UserID=userapp2;"  
    + "Password=passapp2;TrustedContextSystemUserID=masteruser;"  
    + "TrustedContextSystemPassword=masterpassword;"  
  
conn.Open();  
  
// Do processing as userapp2  
  
conn.Close();
```

サーバー上にトラステッド・コンテキストが設定されなかったためにトラステッド・コンテキスト処理が失敗した場合、あるいはサーバーでトラステッド・コンテキストがサポートされない場合には、SQLCODE CLI0197E とともにエラーがスローされます。TrustedContextSystemUserID キーワード値が無効である場合 (例えば長すぎる場合) には、SQLCODE CLI0124E とともにエラーがスローされます。サーバーは、固有エラー・コード -20361 を伴う SQLCODE SQL1046N、SQL30082N、または SQL0969N とともにエラーを報告する場合があります。これらのどのエラーが発生した場合も、Open() は失敗します。

注: トラステッド・コンテキスト処理は、サーバーとの次回の通信で行われます。

ADO.NET データベース・アプリケーションでの SQL データ・タイプ表記

ADO.NET データベース・アプリケーションでは、SQL ステートメントの実行の環境で使われるパラメーター値および変数として、DB2 SQL データ・タイプ値を参照できます。ただし、値のアクセス時または取得時にデータの切り捨てや損失が発生しないよう、適切な IBM Data Server Provider for .NET データ・タイプ値と .NET Framework データ・タイプ値を使用する必要があります。

SQL ステートメントの実行で使用されるパラメーター値を指定するには、IBM Data Server Provider for .NET オブジェクトを使用する必要があります。SQL ステートメントを表す DB2Command オブジェクトに追加するパラメーターを表すために、DB2Parameter オブジェクトが使用されます。パラメーターのデータ・タイプ値を指定する際には、IBM.Data.DB2Types 名前空間で使用可能な IBM Data Server Provider for .NET データ・タイプ値を使用する必要があります。IBM.Data.DB2Types 名前空間には、サポートされる DB2 SQL データ・タイプのそれぞれを表すクラスおよび構造体が提供されています。

SQL データ・タイプ値を一時的に保持する可能性のあるローカル変数に関しては、IBM.Data.DB2Types 名前空間で定義されている該当する IBM Data Server Provider for .NET データ・タイプを使用する必要があります。

次の表は、DB2Type データ・タイプ、DB2 データ・タイプ、Informix データ・タイプ、Microsoft .NET Framework 型、および DB2Types クラスと構造体の対応を示しています。

分類	DB2Types クラスと構造体	DB2Type データ・タイプ	DB2 データ・タイプ	Informix データ・タイプ	.NET データ・タイプ
バイナリー・データ	DB2Binary	Binary	CHAR FOR BIT DATA		Byte[]
	DB2Binary	Binary ³	BINARY		Byte[]
	DB2Binary	VarBinary ³	VARBINARY		Byte[]
	DB2Binary	LongVarBinary ¹	LONG VARCHAR FOR BIT DATA		Byte[]
文字データ	DB2String	Char	CHAR	CHAR	String
	DB2String	VarChar	VARCHAR	VARCHAR	String
	DB2String	LongVarChar ¹	LONG VARCHAR	LVARCHAR	String
グラフィック・データ	DB2String	Graphic	GRAPHIC		String
	DB2String	VarGraphic	VARGRAPHIC		String
	DB2String	LongVarGraphic ¹	LONG VARGRAPHIC		String

1. これらのデータ・タイプは、DB2 .NET 共通言語ランタイム・ルーチンではパラメーターとしてサポートされていません。
2. DB2 の DB2Type.Xml 型の ParameterClass.ParameterName プロパティーには、String 型、byte[] 型、DB2Xml 型、および XmlReader 型の変数を入れることができます。
3. これらのデータ・タイプは、DB2 for z/OS および DB2 for i V6R1 以降にのみ該当します。
4. このデータ・タイプは、DB2 for z/OS バージョン 9 以降のリリースおよび DB2 for Linux, UNIX, and Windows バージョン 9.5 以降のリリースでのみサポートされています。
5. 日付および時刻のオブジェクトには、タイム・スタンプのストリング・リテラルが可能です。タイム・スタンプ・オブジェクトとしては、日付ストリング・リテラルが可能です。

LOB データ	DB2Clob	Clob	CLOB	CLOB, TEXT	String
	DB2Blob	Blob	BLOB	BLOB, BYTE	Byte[]
	DB2Clob	DbClob	DBCLOB		String
数値データ	DB2Int16	SmallInt	SMALLINT	BOOLEAN, SMALLINT	Int16
	DB2Int32	Integer	INT	INTEGER, INT, SERIAL	Int32
	DB2Int64	BigInt, BigSerial	BIGINT	BIGINT, BIGSERIAL, INT8, SERIAL8	Int64
	DB2Real, DB2Real370	Real	REAL	REAL, SMALLFLOAT	Single
	DB2Double	Double	DOUBLE PRECISION	DECIMAL (≤ 29), DOUBLE PRECISION	Double
	DB2Double	Float	FLOAT	DECIMAL (32), FLOAT	Double
	DB2Decimal	Decimal	DECIMAL	MONEY	Decimal
	DB2DecimalFloat	DecimalFloat	DECFLOAT(16 34) ¹⁴		Decimal
	DB2Decimal	Numeric	DECIMAL	DECIMAL (≤ 29), NUMERIC	Decimal
日付/時刻データ	DB2Date	Date	DATE	DATETIME (日付精度)	DateTime String ⁵
	DB2Time	Time	TIME	DATETIME (時刻精度)	TimeSpan String ⁵
	DB2TimeStamp	Timestamp	TIMESTAMP	DATETIME (時刻と日付の精度)	DateTime String ⁵
	DB2TimeStamp Offset	TimestampWith TimeZone	TIMESTAMP WITH TIME ZONE	なし	DateTimeOffset String ⁵
行 ID データ	DB2RowId	RowId	ROWID		Byte[]
XML データ	DB2Xml	Xml ²	XML		Byte[]

IBM Data Server Provider for .NET を使用したアプリケーションからの SQL ステートメントの実行

IBM Data Server Provider for .NET を使用する場合、SQL ステートメントは、DB2Command クラスのメソッド ExecuteReader() および ExecuteNonQuery()、およびそのプロパティ CommandText、CommandType、および Transaction を使用して実行します。

このタスクについて

出力を生成する SQL ステートメントの場合は、ExecuteReader() メソッドを使用でき、その結果を DB2DataReader オブジェクトから取り出すことができます。その他のすべての SQL ステートメントの場合は、ExecuteNonQuery() メソッドを使用できます。DB2Command オブジェクトの Transaction プロパティを DB2Transaction オブジェクトに初期化する必要があります。DB2Transaction オブジェクトは、データベース・トランザクションのロールバックとコミットを担当します。

C# で UPDATE ステートメントを実行する

```
// assume a DB2Connection conn
DB2Command cmd = conn.CreateCommand();
DB2Transaction trans = conn.BeginTransaction();
cmd.Transaction = trans;
cmd.CommandText = "UPDATE staff " +
    " SET salary = (SELECT MIN(salary) " +
    " FROM staff " +
    " WHERE id >= 310) " +
    " WHERE id = 310";

cmd.ExecuteNonQuery();
```

Visual Basic .NET で UPDATE ステートメントを実行する

```
' assume a DB2Connection conn
DB2Command cmd = conn.CreateCommand();
DB2Transaction trans = conn.BeginTransaction();
cmd.Transaction = trans;
cmd.CommandText = "UPDATE staff " +
    " SET salary = (SELECT MIN(salary) " +
    " FROM staff " +
    " WHERE id >= 310) " +
    " WHERE id = 310";
cmd.ExecuteNonQuery();
```

C# で SELECT ステートメントを実行する

```
// assume a DB2Connection conn
DB2Command cmd = conn.CreateCommand();
DB2Transaction trans = conn.BeginTransaction();
cmd.Transaction = trans;
cmd.CommandText = "SELECT deptnumb, location " +
    " FROM org " +
    " WHERE deptnumb < 25";

DB2DataReader reader = cmd.ExecuteReader();
```

Visual Basic .NET で SELECT ステートメントを実行する

```
' assume a DB2Connection conn
Dim cmd As DB2Command = conn.CreateCommand()
Dim trans As DB2Transaction = conn.BeginTransaction()
cmd.Transaction = trans
cmd.CommandText = "UPDATE staff " +
    " SET salary = (SELECT MIN(salary) " +
    " FROM staff " +
    " WHERE id >= 310) " +
    " WHERE id = 310";

cmd.ExecuteNonQuery()
```

アプリケーションでデータベース・トランザクションを実行した後、それをロールバックするか、またはコミットする必要があります。そのためには、DB2Transaction オブジェクトの Commit() メソッドおよび Rollback() メソッドを使用します。

C# でトランザクションをロールバックまたはコミットする

```
// assume a DB2Transaction object conn
trans.Rollback();
...
trans.Commit();
```

Visual Basic.NET でトランザクションをロールバックまたはコミットする

```
' assume a DB2Transaction object conn
trans.Rollback()
...
trans.Commit()
```

バージョン 9.7 フィックスパック 2 以降のフィックスパックでは、DB2 for Linux, UNIX, and Windows サーバーは無名ブロックの実行から結果セットを戻すことをサポートしています。バージョン 9.7 フィックスパック 6 以降のフィックスパックでは、.NET データ・プロバイダーは、DB2DataReader クラスまたは DB2ResultSet クラスを使用してアプリケーションが無名ブロックの実行から結果セットを取り出すことをサポートしています。 .NET データ・プロバイダーが無名ブロックの実行から結果セットを取り出すには、データベース・サーバーで PL/SQL ステートメントがサポートされている必要があり、また、データベースで PL/SQL ステートメントが処理できるようになっている必要があります。 .NET データ・プロバイダーは、BEGIN COMPOUND ステートメントではなく BEGIN ステートメントを使用して、無名ブロックの実行から戻される結果セットに対するカーソルを宣言する必要があります。

C# で DB2DataReader クラスを使用して無名ブロックの実行から単一の結果セットを取り出すには、次のようにします。

```
...
cmd.CommandText = "begin " +
    "declare cursor1 cursor with return to client with hold for select c1 from t1; " +
    "open cursor1; " +
    "end;";
//Returns a result set by opened cursor cursor1
DB2DataReader dr = cmd.ExecuteReader();
while (dr.Read())
{
    //Process read data if required
}
dr.Close();
...
```

C# で DB2ResultSet クラスを使用して無名ブロックの実行から単一の結果セットを取り出すには、次のようにします。

```
...
cmd.CommandText = "begin " +
    "declare cursor1 cursor with return to client with hold for select c1 from t1; " +
    "open cursor1; " +
    "end;";
//Returns a result set by opened cursor cursor1
DB2ResultSet ds = cmd.ExecuteResultSet(DB2CursorType.ForwardOnly);
while (ds.Read())
{
    //Process read data if required
}
ds.Close();
...
```

C# で DB2DataReader クラスを使用して無名ブロックの実行から複数の結果セットを取り出すには、次のようにします。

```
...
cmd.CommandText = " begin " +
    "declare cursor1 cursor with return to client with hold for select c1 from t1; " +
    "declare cursor2 cursor with return to client for select c2 from t2; " +
    "open cursor1; " +
    "open cursor2; " +
    "end;";
//Returns multiple result sets by opened cursors
DB2DataReader dr = cmd.ExecuteReader();
while (dr.Read())
{
    //Process read data if required from cursor1
}
```

```

}
dr.NextResult(); //Get next result set
while (dr.Read())
{
    //Process read data if required from cursor2
}
dr.Close();
...

```

C# で DB2ResultSet クラスを使用して無名ブロックの実行から複数の結果セットを取り出すには、次のようにします。

```

...
cmd.CommandText = " begin " +
    "declare cursor1 cursor with return to client with hold for select c1 from t1; " +
    "declare cursor2 cursor with return to client for select c2 from t2; " +
    "open cursor1; " +
    "open cursor2; " +
    "end;";
//Returns multiple result sets by opened cursors
DB2ResultSet ds = cmd.ExecuteResultSet(DB2CursorType.ForwardOnly);
while (ds.Read())
{
    //Process read data if required from cursor1
}
ds.NextResult(); //Get next result set
while (ds.Read())
{
    //Process read data if required from cursor2
}
ds.Close();
...

```

IBM Data Server Provider for .NET を使用したアプリケーションからの結果セットの読み取り

IBM Data Server Provider for .NET を使用する場合、結果セットを読み取るには、DB2DataReader オブジェクトを使用します。結果セットの中で次の行に進むには、DB2DataReader の Read() メソッドを使用します。

このタスクについて

出力に含まれる個々の列からデータを抽出するには、メソッド GetString()、GetInt32()、GetDecimal()、およびその他の使用可能なすべてのデータ・タイプ用のメソッドを使用します。出力の読み取りが終了したら、DB2DataReader オブジェクトを必ずクローズしなければなりません。そのためには、DB2DataReader の Close() メソッドを使用します。

C# で結果セットを読み取る

```

// assume a DB2DataReader reader
Int16 deptnum = 0;
String location="";

// Output the results of the query
while(reader.Read())
{
    deptnum = reader.GetInt16(0);
    location = reader.GetString(1);
    Console.WriteLine("    " + deptnum + " " + location);
}
reader.Close();

```

Visual Basic .NET で結果セットを読み取る

```

' assume a DB2DataReader reader
Dim deptnum As Int16 = 0
Dim location As String ""

' Output the results of the query
Do While (reader.Read())
    deptnum = reader.GetInt16(0)
    location = reader.GetString(1)
    Console.WriteLine("    " & deptnum & " " & location)
Loop
reader.Close();

```

IBM Data Server Provider for .NET を使用したアプリケーションからのストアード・プロシージャの呼び出し

IBM Data Server Provider for .NET を使用する場合、DB2Command オブジェクトを使用することによってストアード・プロシージャを呼び出すことができます。

このタスクについて

CommandType プロパティのデフォルト値は、CommandType.Text です。これは SQL ステートメントに適した値であり、ストアード・プロシージャを呼び出すためにも使用できます。ただし、CommandType に CommandType.StoredProcedure を設定したほうが、ストアード・プロシージャの呼び出しはより容易になります。この場合は、ストアード・プロシージャ名とパラメーターのみを指定する必要があります。

ストアード・プロシージャの作業を実行する場合、ホスト変数、名前付きパラメーター、または位置によるパラメーターを使用することにより、パラメーターを渡すことができます。しかし、同じ SQL ステートメントの中でそれらを組み合わせで使用することはできません。

以下に示す C# および Visual Basic の例は、INOUT_PARAM というストアード・プロシージャを呼び出す方法を示します。その際、CommandType プロパティを CommandType.StoredProcedure または CommandType.Text のいずれかに設定します。

手順

- C# で DB2Command の CommandType プロパティを CommandType.Text に設定することによって、ストアード・プロシージャを呼び出します。

```

// assume a DB2Connection conn
DB2Transaction trans = conn.BeginTransaction();
DB2Command cmd = conn.CreateCommand();
String procName = "INOUT_PARAM";
String procCall = "CALL INOUT_PARAM (@param1, @param2, @param3)";
cmd.Transaction = trans;
cmd.CommandType = CommandType.Text;
cmd.CommandText = procCall;

// Register input-output and output parameters for the DB2Command
cmd.Parameters.Add( new DB2Parameter("@param1", "Value1");
cmd.Parameters.Add( new DB2Parameter("@param2", "Value2");
DB2Parameter param3 = new DB2Parameter("@param3", IfxType.Integer);
param3.Direction = ParameterDirection.Output;
cmd.Parameters.Add( param3 );

```

```
// Call the stored procedure
Console.WriteLine(" Call stored procedure named " + procName);
cmd.ExecuteNonQuery();
```

- Visual Basic で DB2Command の CommandType プロパティを CommandType.Text に設定することによって、ストアド・プロシージャを呼び出します。

```
' assume a DB2Connection conn
Dim trans As DB2Transaction = conn.BeginTransaction()
Dim cmd As DB2Command = conn.CreateCommand()
Dim procName As String = "INOUT_PARAM"
Dim procCall As String = "CALL INOUT_PARAM (?, ?, ?)"
cmd.Transaction = trans
cmd.CommandType = CommandType.Text
cmd.CommandText = procCall

' Register input-output and output parameters for the DB2Command
...

' Call the stored procedure
Console.WriteLine(" Call stored procedure named " & procName)
cmd.ExecuteNonQuery()
```

注: CALL と EXECUTE PROCEDURE は、いずれもサポートされています。

- C# で DB2Command の CommandType プロパティを CommandType.StoredProcedure に設定することによって、ストアド・プロシージャを呼び出します。この方法を使用する場合、名前付きパラメーターはサポートされません。

```
// assume a DB2Connection conn
DB2Transaction trans = conn.BeginTransaction();
DB2Command cmd = conn.CreateCommand();
String procName = "INOUT_PARAM";
cmd.Transaction = trans;
cmd.CommandType = CommandType.StoredProcedure;
cmd.CommandText = procName;

// Register input-output and output parameters for the DB2Command
...

// Call the stored procedure
Console.WriteLine(" Call stored procedure named " + procName);
cmd.ExecuteNonQuery();
```

- Visual Basic で DB2Command の CommandType プロパティを CommandType.StoredProcedure に設定することによって、ストアド・プロシージャを呼び出します。

```
' assume a DB2Connection conn
Dim trans As DB2Transaction = conn.BeginTransaction()
Dim cmd As DB2Command = conn.CreateCommand()
Dim procName As String = "INOUT_PARAM"
cmd.Transaction = trans
cmd.CommandType = CommandType.StoredProcedure
cmd.CommandText = procName

' Register input-output and output parameters for the DB2Command
...

' Call the stored procedure
Console.WriteLine(" Call stored procedure named " & procName)
cmd.ExecuteNonQuery()
```

- パラメーター名で修飾する場合、複数のパラメーターを任意の順序でストアード・プロシージャに渡すことができます。この名前付き引数フィーチャーは、DB2 for Linux, UNIX, and Windows データ・サーバーでのみサポートされます。例えば、以下の SQL ステートメントでは、ストアード・プロシージャを定義した後、定義とは異なる順序でパラメーターを指定してそれを呼び出しています。

```
CREATE PROCEDURE schema.my_proc ( IN var1 int, INOUT var2 int )
LANGUAGE SQL
BEGIN
  -- (プロシージャのコード)
END

CALL my_proc (var2=>@param2, var1=>@param1)
```

CURSOR タイプの出力パラメーターによって戻された結果セットへの同時アクセス

IBM Data Server Provider for .NET を使用している場合、DB2Type.Cursor は出力パラメーターのすべてのカーソルに同時にアクセスするように指定されます。

このタスクについて

複数の CURSOR タイプ出力パラメーターのあるストアード・プロシージャの場合、DB2TYPE.Cursor を出力パラメーター・オブジェクトにバインドすると、出力パラメーターに含まれるすべてのカーソルに同時にアクセスできます。

例えば、OrderDetails ストアード・プロシージャで 3 つのカーソルを宣言し、それぞれのカーソルが製品とその売り上げについての関連情報を提供するとします。

```
CREATE OR REPLACE TYPE cur AS CURSOR
CREATE PROCEDURE OrderDetails (p_startDate TIMESTAMP, p_endDate TIMESTAMP,
  OUT prodOrderDetails cur, OUT prodOrderDetails cur, OUT custOrderDetails cur)
LANGUAGE SQL
BEGIN
  SET prodDetails = CURSOR WITH HOLD FOR
  SELECT p.pid, price, quantity FROM products p, inventory i
  WHERE p.pid = i.pid AND p.pid IN (SELECT DISTINCT pid FROM orders) ORDER BY pid;
  SET prodOrderDetails = CURSOR WITH HOLD FOR
  SELECT pid, COUNT(*), SUM (quantity) FROM orders
  WHERE date >= p_startDate AND date <= p_endDate GROUP BY pid ORDER BY pid;
  SET custOrderDetails = CURSOR WITH HOLD FOR
  SELECT pid, custID, COUNT(*), SUM(quantity) FROM orders
  WHERE date >= p_startDate AND date <= p_endDate
  GROUP BY pid, custID ORDER by pid, custID;
  OPEN prodDetails;
  OPEN prodOrderDetails;
  OPEN custOrderDetails;
END;
```

それぞれのカーソルから特定の製品についての関連情報を収集して割引を計算できるように、呼び出し元はこれらのカーソルに同時にアクセスする必要があります。カーソルへの同時アクセスを可能にするために、ストアード・プロシージャはこれらのカーソルを出力パラメーターとして戻します。CURSOR タイプの出力パラメーターをバインドして同時アクセスが行われるようにするときは、アプリケーションで DB2Type を DB2Type.Cursor に設定する必要があります。

```
//C# Code sample
cmd.CommandText = "CALL OrderDetails(
  @p_startDate, @p_endDate, @prodDetails, @prodOrderDetails, @custOrderDetails)";
cmd.Parameters.Add("@p_startDate", DateTime.Parse("1/1/2010"));
cmd.Parameters.Add("@p_endDate", DateTime.Parse("12/31/2010"));
cmd.Parameters.Add("@prodDetails", DB2Type.Cursor);
cmd.Parameters["@prodDetails"].Direction = ParameterDirection.Output;
cmd.Parameters.Add("@prodOrderDetails", DB2Type.Cursor);
cmd.Parameters["@prodOrderDetails"].Direction = ParameterDirection.Output;
cmd.Parameters.Add("@custOrderDetails", DB2Type.Cursor);
cmd.Parameters["@custOrderDetails"].Direction = ParameterDirection.Output;
cmd.ExecuteNonQuery();
DB2DataReader prodDetailsDR =
  (DB2DataReader)cmd.Parameters["@prodDetails"].Value;
DB2DataReader prodOrderDetailsDR =
  (DB2DataReader)cmd.Parameters["@prodOrderDetails"].Value;
```

```

DB2DataReader custOrderDetailsDR =
(DB2DataReader)cmd.Parameters["@custOrderDetails"].Value;
while (prodOrderDetailsDR.Read())
{
    pid = prodOrderDetailsDR.GetInt32(0);
    numOrders = prodOrderDetailsDR.GetInt32(1);
    totalOrderQuantity = prodOrderDetailsDR.GetInt32(2);
    prodDetailsDR.Read();
    price = prodDetailsDR.GetDecimal(1);
    currentInventory = prodDetailsDR.GetInt32(2);
    int totalCustOrders = 0;
    while (custOrderDetailsDR.Read())
    {
        custID = custOrderDetailsDR.GetInt32(1);
        numOrdersByCust = custOrderDetailsDR.GetInt32(2);
        totalCustOrders += numOrdersByCust;
        totalOrderQuantityByCust = custOrderDetailsDR.GetInt32(3);
        //Calculate discount based on numOrders, numOrdersByCust,
        // totalOrderQuantity, totalOrderQuantityByCust, price and currentInventory
        if (totalCustOrders == numOrders) //done with this pid
            break;
    }
}
prodDetailsDR.Close();
prodOrderDetailsDR .Close();
custOrderDetailsDR .Close();

```

CURSOR タイプの出力パラメーターからデータを読み取るプログラムへは、ExecuteNonQuery メソッドを起動した後でのみ Value プロパティーからアクセスできます。

ExecuteReader メソッドか ExecuteResultSet メソッドのいずれかを使用してコマンドが実行されると、結果セットは DB2DataReader オブジェクトまたは DB2ResultSet オブジェクトで戻されます。それ以降の結果セットへは、NextResult メソッドを呼び出すことにより順にアクセスしなければなりません。出力パラメーターはバインドされているものの、出力パラメーター Value プロパティーにアクセスすると、InvalidOperationException 例外が発生します。これは、照会が ExecuteNonQuery メソッドを使用して実行されなかったことによるものです。

カーソルを同時に操作しているとき、アプリケーションはカーソルの読み取りを続行する前に、操作をコミットしようとする場合があります。オープン・カーソルを破壊することなくアプリケーションがコミットを発行するためには、ストアード・プロシージャーの中でカーソルを保留可能なものとして宣言しておく必要があります。

System.Diagnostics.Trace を使用した IBM Data Server Provider for .NET のトレース

このトピックでは、System.Diagnostics.Trace を使用した IBM Data Server Provider for .NET のトレースについて説明します。

このタスクについて

.NET 共通 API における関数の入り口および出口のトレース・ポイントは、System.Diagnostics.Trace を使用してトレースされます。第 1 レベル呼び出しのみがトレースされます。例えば、顧客のアプリケーションによって明示的に行われたプロバイダーの呼び出しのみがトレースされます。プロバイダー自体によるパブリック API へ内部的な呼び出しはトレースされません。

IBM.Data.DB2 の名前空間では TraceSwitch は DB2NMPTRACE という名前に、IBM.Data.Informix の名前空間では TraceSwitch は IFXNMPTRACE という名前になります。

```

<configuration>
  <system.diagnostics>
    <switches>

```

```
<add name="DB2NMPTRACE" value="1" />
</switches>
</system.diagnostics>
</configuration>
```

TraceSwitch クラスは System.Diagnostics.TraceLevel 列挙型を使用して、トレースされる情報のレベルを制御します。すなわち、0 から 4 で、Off、Error、Warning、Info、および Verbose に対応します。.NET プロバイダーの場合は、0 が Off を表し、1 から 4 が On を表します。

例

以下の出力は、簡単なプログラムのトレース出力の例です。

注: アプリケーション自体によってダンプされたトレース・ポイントには * の接頭部が付けられます。

```
* * Started tracing program
* Creating connection
DB2Connection.DB2Connection1 api entry - database=nmpfvту;
DB2Connection.DB2Connection1 api exit, rc = 0
* Opening connection
DB2Connection.Open api entry
DB2Connection.Open api exit, rc = 0
* Closing connection
DB2Connection.Close api entry
DB2Connection.Close api exit, rc = 0
* Ending program
DB2Connection.~DB2Connection api entry
DB2Connection.~DB2Connection api exit, rc = 0
DB2Connection.Dispose api entry
DB2Connection.Dispose api exit, rc = 0
```

pureQuery を使用した、.NET アプリケーションでの照会の最適化

.NET クライアント・ドライバは、pureQuery テクノロジーに備わって要るフィーチャーを活用することができます。これらのフィーチャーを使用して、既存の .NET アプリケーション照会を静的 SQL として実行することができます。静的照会では、実行時に特定のステートメントを準備する必要がなくなります。これが、アプリケーションでのセキュリティーおよびパフォーマンスの向上につながる場合があります。

詳しくは、<http://www.ibm.com/support/docview.wss?uid=swg27023946>を参照してください。

IBM Data Server Provider for .NET の Microsoft Entity Framework のサポート

IBM Data Server Provider for .NET を使用することにより、IBM データ・サーバーで Microsoft ADO.NET Entity Framework を活用できます。サポートされているサーバー・バージョンを使用して、EDM スキーマを生成したり、Entities アプリケーションのために Entity SQL および LINQ ステートメントを作成および実行したりできます。

システム要件

IBM Data Server Provider for .NET は、以下の IBM データ・サーバーと共に機能します。

- IBM DB2 for Linux, UNIX, and Windows のすべてのサポート対象レベル
- DB2 for i バージョン 5 リリース 4、バージョン 6 リリース 1、およびバージョン 7 リリース 1 (DB2 Connect 経由) (IBM DB2 バージョン 9.7 フィックスパック 4 以降のバージョン用)
- DB2 for z/OS バージョン 8 以降
- IBM Informix バージョン 11.170 以降

Microsoft .NET Framework 3.5 SP1 以降および Microsoft ADO.NET Entity Framework が必要です。Microsoft Entity Data Model ウィザードまたは ADO.NET Entity Designer を使用してエンティティ・データ・モデルを操作するには、Microsoft Visual Studio 2008 以降も必要です。

以下の表は、IBM エンティティ・プロバイダーでサポートされる正規関数をリストしています。正規関数は、データ・プロバイダーによって、対応するデータ・ソース関数に変換されます。

表 1. IBM エンティティ・プロバイダーでの正規関数サポート

正規関数のタイプ	LINQ 関数	DB2 for Linux, UNIX, and Windows	DB2 for z/OS	DB2 for i	Informix
Aggregate	Average	Y	Y	Y	Y
	BigCount	Y	Y	Y	Y
	Count	Y	Y	Y	Y
	Maximum	Y	Y	Y	Y
	Minimum	Y	Y	Y	Y
	NewGuid	Y*	Y*	Y*	Y*
	StDev	Y	Y	Y	Y
	StDevP	Y	Y	Y	Y
	Sum	Y	Y	Y	Y
	Var	Y	Y	Y	Y
	VarP	Y	Y	Y	Y
Bitwise	BitWiseAnd	Y	Y*	Y*	Y
	BitWiseNot	Y	Y*	Y*	Y
	BitWiseOr	Y	Y*	Y*	Y
	BitWiseXor	Y	Y*	Y*	Y
Math	Abs	Y	Y	Y	Y
	Ceiling	Y	Y	Y	Y
	Floor	Y	Y	Y	Y
	Power	Y	Y	Y	Y
	Round (value,digits)	Y	Y	Y	Y
	Truncate (value,digits)	Y	Y	Y	Y

表 1. IBM エンティティ・プロバイダーでの正規関数サポート (続き)

正規関数の タイプ	LIHQ 関数	DB2 for Linux, UNIX, and Windows	DB2 for z/OS	DB2 for i	Informix
String	Concat	Y	Y	Y	Y
	Contains	Y	Y	Y	Y*
	EndsWith	Y	Y	Y	Y
	IndexOf	Y	Y	Y	Y*
	Left	Y	Y	Y	Y
	Length	Y	Y	Y	Y
	LTrim	Y	Y	Y	Y
	Replace	Y	Y	Y	Y
	Right	Y	Y	Y	Y
	RTrim	Y	Y	Y	Y
	StartsWith	Y	Y	Y	Y
	Substring	Y	Y	Y	Y
	ToLower	Y	Y	Y	Y
	ToUpper	Y	Y	Y	Y
	Trim	Y	Y	Y	Y
Datetime	AddNanoseconds	Y	Y	Y	Y
	AddMicroseconds	Y	Y	Y	Y
	AddMilliseconds	Y	Y	Y	Y
	AddSeconds	Y	Y	Y	Y
	AddMinutes	Y	Y	Y	Y
	AddHours	Y	Y	Y	Y
	AddDays	Y	Y	Y	Y
	AddMonths	Y	Y	Y	Y
	AddYears	Y	Y	Y	Y
	CreateDateTime	Y	Y	Y	Y
	CreateDateTimeOffset		Y		
	CurrentDateTimeOffset		Y		
	CreateTime	Y	Y	Y	Y
	CurrentDateTime	Y	Y	Y	Y
	CurrentUtcDateTime	Y	Y	Y	
	Day	Y	Y	Y	Y
	DayOfYear	Y	Y	Y	Y
	DiffNanoseconds	Y	Y	Y	Y*
	DiffMicroseconds	Y	Y	Y	Y*
	DiffMilliseconds	Y	Y	Y	Y*
	DiffSeconds	Y	Y	Y	Y*
	DiffMinutes	Y	Y	Y	Y*
	DiffHours	Y	Y	Y	Y*
	DiffDays	Y	Y	Y	Y*
	DiffMonths	Y	Y	Y	Y*
	DiffYears	Y	Y	Y	Y*
	GetTotalOffsetMinutes		Y		
	Hour	Y	Y	Y	Y
	Millisecond	Y	Y	Y	Y
	Minute	Y	Y	Y	Y
	Month	Y	Y	Y	Y
	Second	Y	Y	Y	Y
	Truncate (datetime exp)	Y	Y	Y	Y
Year	Y	Y	Y	Y	

既知の制約

重要: 正規関数の中には、サーバーに依存しているものがあります。エラー SQL0440N* は、サーバーが特定の機能をサポートしていないことを示します。サポートされる機能について詳しくは、サーバーの IBM 技術サポートにお問い合わせください。

以下の既知の制限が、Microsoft ADO.NET Entity Framework での IBM Data Server Provider for .NET サポートに適用されます。

一般:

- データベースの最初のシナリオのみがサポートされます。つまり、Entity Framework を使用する前に、すべてのデータベース・オブジェクトが存在している必要があります。
- ストア固有の関数の呼び出しはサポートされていません。
- サーバー・エクスプローラーの「接続の追加」ダイアログで設定したトラステッド・コンテキスト接続プロパティは、Entity Framework 接続に渡されません。

DB2 for z/OS:

- データ・タイプ REAL はサポートされていません。アプリケーションは、表のスキーマで FLOAT データ・タイプを使用するか、サーバーでの実際のタイプが REAL であっても、クライアント・スキーマ (EDM) でタイプを FLOAT として指定する必要があります。
- バージョン 8 とバージョン 7 に固有の情報:
Take、Top、First、Intersect、または Except の各式を含む照会で、ある種の構文エラーを示す例外が生成される場合があります。これらの式を含む照会の結果は未定義です。

例:

```
1) var query = from o in context.Orders
where o.ShipCity == "Seattle"
select o;
var result = query.First();

2) var mexico =
context.OrderDetails.Where(od => od.Order.ShipCountry
== "Mexico").Select(od => od.Product);
var canada =
context.OrderDetails.Where(od => od.Order.ShipCountry
== "Canada").Select(od => od.Product);
var query = mexico.Intersect(canada);

3) var query =
context.Customers.Select(e => e).Except(context.Orders.Where
(o => o.ShipCountry == "Mexico").Select(o => o.Customer));

4) var query = context.Orders.Include("OrderDetails").Top("1");

5) var query = context.Orders.Include("OrderDetails").
Include("OrderDetails.Product").Take(3).Select(o => o);
```

Enterprise Library データ・アクセス・モジュールの使用

Enterprise Library は、開発者が開発作業で直面する一般的な問題に対処するのに役立つ、アプリケーション・ブロックの集まりです。アプリケーション・ブロックはソース・コードの形で提供され、そのまま使用することも開発プロジェクトに応じて修正することも可能です。

IBM データ・サーバーのための Enterprise Library データ・アクセス・モジュールは、その他のモジュールと共に <http://codeplex.com/entlibcontrib/SourceControl/PatchList.aspx> で入手できます。

IBM データ・サーバー (DB2、Informix データベース・サーバー、および U2) と共に Enterprise Library データ・アクセス・モジュールをインストールして使用方法については、ダウンロードしたパッケージに含まれている readme ファイルを参照してください。

リソース

以下に、データ・アクセス・モジュールの使用方法を説明しているオンライン・リソースを紹介します。

- EntLib Contrib Project ホーム・ページ: <http://www.codeplex.com/entlibcontrib>
- patterns & practices for Enterprise Library: <http://www.codeplex.com/entlib>
- Microsoft Enterprise Library ホーム・ページ: <http://msdn.microsoft.com/en-us/library/cc467894.aspx>
- IBM DB2 for .NET: <http://www.ibm.com/software/data/db2/windows/dotnet.html>

.NET アプリケーションの構築

.NET アプリケーションを構築するためのリソース

Visual Basic .NET アプリケーションの構築

DB2 製品には、DB2 Visual Basic .NET アプリケーションのコンパイルおよびリンク用の `bldapp.bat` バッチ・ファイルが用意されています。

このバッチ・ファイルは、このファイルを使用して構築できるサンプル・プログラムと一緒に `sqllib¥samples¥.NET¥vb` ディレクトリーに置かれています。このバッチ・ファイルは 1 つのパラメーター `%1` をとります。このパラメーターは、コンパイルするソース・ファイルの名前を (`.vb` 拡張を付けずに) 指定します。

このタスクについて

このタスクは、DbAuth サンプル・ファイルを使用し `bldapp.bat` を介して Visual Basic .NET アプリケーションを構築する基本ステップをたどります。

手順

ソース・ファイル `DbAuth.vb` からプログラム `DbAuth` を構築するには、次のように入力します。

```
bldapp DbAuth
```

実行可能ファイルを実行する際に必要なパラメーターを確実に指定するようにするには、以下のように入力する情報の数に応じてさまざまなパラメーターの組み合わせを指定できます。

1. パラメーターなし。次のように、プログラム名のみ入力します。

```
DbAuth
```

2. 1 つのパラメーター。次のように、プログラム名に加えてデータベース別名を入力します。

```
DbAuth <db_alias>
```

3. 2 つのパラメーター。次のように、プログラム名に加えてユーザー ID とパスワードを入力します。

```
DbAuth <userid> <passwd>
```

4. 3 つのパラメーター。次のように、プログラム名に加えてデータベース別名、ユーザー ID、およびパスワードを入力します。

```
DbAuth <db_alias> <userid> <passwd>
```

5. 4 つのパラメーター。次のように、プログラム名に加えてサーバー名、ポート番号、ユーザー ID、およびパスワードを入力します。

```
DbAuth <server> <portnum> <userid> <passwd>
```

6. 5 つのパラメーター。次のように、プログラム名に加えてデータベース別名、サーバー名、ポート番号、ユーザー ID、およびパスワードを入力します。

```
DbAuth <db_alias> <server> <portnum> <userid> <passwd>
```

次のタスク

LCTrans サンプル・プログラムを構築して実行するには、ソース・ファイル LCTrans.vb に示されている詳細な指示に従う必要があります。

C# .NET アプリケーションの構築

DB2 製品には、DB2 C# .NET アプリケーションのコンパイルおよびリンク用の bldapp.bat バッチ・ファイルが用意されています。このバッチ・ファイルは、sqllib\samples\.NET\cs ディレクトリーに入っています。そこには、このファイルでビルド可能なサンプル・プログラムも含まれています。

このバッチ・ファイルには 1 個のパラメーター %1 があります。それは、コンパイル対象のソース・ファイルの名前 (.cs 拡張子を除いたもの) を指定します。

このタスクについて

このタスクは、DbAuth サンプル・ファイルを使用し bldapp.bat を介して C# .NET アプリケーションを構築する基本ステップをたどります。

手順

ソース・ファイル DbAuth.cs からプログラム DbAuth を構築するには、次のように入力します。

```
bldapp DbAuth
```

実行可能ファイルを実行する際に必要なパラメーターを確実に指定するようにするには、以下のように入力する情報の数に応じてさまざまなパラメーターの組み合わせを指定できます。

1. パラメーターなし。次のように、プログラム名のみ入力します。

```
DbAuth
```

2. 1 つのパラメーター。次のように、プログラム名に加えてデータベース別名を入力します。

```
DbAuth <db_alias>
```

3. 2 つのパラメーター。次のように、プログラム名に加えてユーザー ID とパスワードを入力します。

```
DbAuth <userid> <passwd>
```

4. 3 つのパラメーター。次のように、プログラム名に加えてデータベース別名、ユーザー ID、およびパスワードを入力します。

```
DbAuth <db_alias> <userid> <passwd>
```

5. 4 つのパラメーター。次のように、プログラム名に加えてサーバー名、ポート番号、ユーザー ID、およびパスワードを入力します。

```
DbAuth <server> <portnum> <userid> <passwd>
```

6. 5 つのパラメーター。次のように、プログラム名に加えてデータベース別名、サーバー名、ポート番号、ユーザー ID、およびパスワードを入力します。

```
DbAuth <db_alias> <server> <portnum> <userid> <passwd>
```

次のタスク

LCTrans サンプル・プログラムを構築して実行するには、ソース・ファイル LCTrans.cs に示されている詳細な指示に従う必要があります。

Visual Basic .NET アプリケーションのコンパイルとリンクのオプション

このトピックは、Visual Basic .NET アプリケーションのコンパイルおよびリンク時に利用できるさまざまなオプションを説明します。

以下に示すコンパイルとリンクのオプションは、Windows 上で Microsoft Visual Basic .NET コンパイラーを使用して、Visual Basic .NET アプリケーションを構築するために使用できます。これらは、bldapp.bat バッチ・ファイルの中で示されているものです。

注: .NET Framework バージョン 1.1 は、.NET Provider バージョン 9.5 以前でのみサポートされます。

bldapp を使用するスタンドアロン VB .NET アプリケーションのコンパイルおよびリンクのオプション

スタンドアロン VB .NET アプリケーションのコンパイルおよびリンクのオプション

%BLDCOMP%

コンパイラー用の変数です。デフォルトは vbc (Microsoft Visual Basic .NET コンパイラー) です。

/r:"%DB2PATH%"¥bin¥%VERSION%¥IBM.Data.DB2.d11

使用している .NET Framework のバージョンに対応する DB2 ダイナミック・リンク・ライブラリーを参照します。

%DB2PATH%

%DB2PATH% 変数は DB2 製品のインストール済み環境のルート・パスを表します。 %DB2PATH% 変数は、IBM Data Server Driver for ODBC and CLI または Data Server Driver Package のインストール済み環境には存在しません。 IBM Data Server Driver for ODBC and CLI または Data Server Driver Package を使用している場合は、%DB2PATH% をドライバー製品がインストールされているパスに置き換えてください。

%VERSION%

アプリケーション用にサポートされている .NET Framework のバージョンはいくつかあります。 DB2 には、各バージョンに対応するダイナミック・リンク・ライブラリーがあります。 .NET Framework バージョン 2.0、3.0、および 3.5 の場合、%VERSION% は netfx2¥ サブディレクトリーを示します。

疎結合サンプル・プログラム LCTrans のコンパイルおよびリンク・オプション (bldapp を使用)

%BLDCOMP%

コンパイラー用の変数です。 デフォルトは vbc (Microsoft Visual Basic .NET コンパイラー) です。

/out:RootCOM.d11

LCTrans アプリケーションが使用する RootCOM ダイナミック・リンク・ライブラリーを、RootCOM.vb ソース・ファイルから出力します。

/out:SubCOM.d11

LCTrans アプリケーションが使用する SubCOM ダイナミック・リンク・ライブラリーを、SubCOM.vb ソース・ファイルから出力します。

/target:library %1.cs

入力ソース・ファイル (RootCOM.vb または SubCOM.vb) からダイナミック・リンク・ライブラリーを作成します。

/r:System.EnterpriseServices.d11

Microsoft Windows の System EnterpriseServices データ・リンク・ライブラリーを参照します。

/r:"%DB2PATH%"¥bin¥%VERSION%¥IBM.Data.DB2.d11

使用している .NET Framework のバージョンに対応する DB2 ダイナミック・リンク・ライブラリーを参照します。

%DB2PATH%

%DB2PATH% 変数は DB2 製品のインストール済み環境のルート・パスを表します。 %DB2PATH% 変数は、IBM Data Server Driver for ODBC and CLI または Data Server Driver Package のインストール済み環境には存在しません。 IBM Data Server Driver for

ODBC and CLI または Data Server Driver Package を使用している場合は、%DB2PATH% をドライバー製品がインストールされているパスに置き換えてください。

%VERSION%

アプリケーション用にサポートされている .NET Framework のバージョンはいくつかあります。DB2 には、各バージョンに対応するダイナミック・リンク・ライブラリーがあります。 .NET Framework バージョン 2.0 および 3.0 の場合、%VERSION% は netf20¥ サブディレクトリーを示します。

/r:System.Data.d11

Microsoft Windows の System Data ダイナミック・リンク・ライブラリーを参照します。

/r:System.d11

Microsoft Windows の System ダイナミック・リンク・ライブラリーを参照します。

/r:System.Xml.d11

Microsoft Windows の System XML ダイナミック・リンク・ライブラリーを参照します (SubCOM.vb 用)。

/r:SubCOM.d11

SubCOM ダイナミック・リンク・ライブラリーを参照します (RootCOM.vb および LCTrans.vb 用)。

/r:RootCOM.d11

RootCOM ダイナミック・リンク・ライブラリーを参照します (LCTrans.vb 用)。

他のコンパイラー・オプションについては、コンパイラーの資料をご覧ください。

C# .NET アプリケーションのコンパイルとリンクのオプション

このトピックでは、C# .NET アプリケーションをコンパイルおよびリンクする際に利用できるさまざまなオプションについて説明します。

Windows 上で Microsoft C# コンパイラーを使用して C# アプリケーションを構築する場合に DB2 で使用可能なコンパイルとリンクのオプション。これらは、bldapp.bat バッチ・ファイルの中で示されているものです。

注: .NET Framework バージョン 1.1 は、.NET Provider バージョン 9.5 以前でのみサポートされます。

スタンドアロン C# アプリケーションのコンパイルおよびリンクのオプション (bldapp を使用)

スタンドアロン C# アプリケーションのコンパイルおよびリンクのオプション

%BLDCOMP%

コンパイラー用の変数です。デフォルトは、csc (Microsoft C# コンパイラー) です。

/r:"%DB2PATH%"¥bin¥%VERSION%IBM.Data.DB2.d11

使用している .NET Framework のバージョンに対応する DB2 ダイナミック・リンク・ライブラリーを参照します。

%VERSION%

アプリケーション用にサポートされている .NET Framework のバージョンはいくつかあります。DB2 には、その各バージョンに対応するダイナミック・リンク・ライブラリーがあります。 .NET Framework バージョン 2.0、3.0、および 3.5 の場合、%VERSION% は netf20¥ サブディレクトリーを示します。

疎結合サンプル・プログラム LCTrans のコンパイルおよびリンク・オプション (bldapp を使用)**%BLDCOMP%**

コンパイラー用の変数です。デフォルトは、csc (Microsoft C# コンパイラー) です。

/out:RootCOM.d11

LCTrans アプリケーションが使用する RootCOM ダイナミック・リンク・ライブラリーを、RootCOM.cs ソース・ファイルから出力します。

/out:SubCOM.d11

LCTrans アプリケーションが使用する SubCOM ダイナミック・リンク・ライブラリーを、SubCOM.cs ソース・ファイルから出力します。

/target:library %1.cs

入力ソース・ファイル (RootCOM.cs または SubCOM.cs) からダイナミック・リンク・ライブラリーを作成します。

/r:System.EnterpriseServices.d11

Microsoft Windows の System EnterpriseServices データ・リンク・ライブラリーを参照します。

/r:"%DB2PATH%"¥bin¥%VERSION%IBM.Data.DB2.d11

使用している .NET Framework のバージョンに対応する DB2 ダイナミック・リンク・ライブラリーを参照します。

%VERSION%

アプリケーション用にサポートされている .NET Framework のバージョンはいくつかあります。DB2 には、各バージョンに対応するダイナミック・リンク・ライブラリーがあります。 .NET Framework バージョン 2.0、3.0、および 3.5 の場合、%VERSION% は netf20¥ サブディレクトリーを示します。

/r:System.Data.d11

Microsoft Windows の System Data ダイナミック・リンク・ライブラリーを参照します。

/r:System.d11

Microsoft Windows の System ダイナミック・リンク・ライブラリーを参照します。

/r:System.Xml.dll

Microsoft Windows の System XML ダイナミック・リンク・ライブラリーを参照します (SubCOM.cs 用)。

/r:SubCOM.dll

SubCOM ダイナミック・リンク・ライブラリーを参照します (RootCOM.cs および LCTrans.cs 用)。

/r:RootCOM.dll

RootCOM ダイナミック・リンク・ライブラリーを参照します (LCTrans.cs 用)。

他のコンパイラー・オプションについては、コンパイラーの資料をご覧ください。

OLE DB .NET Data Provider

OLE DB .NET Data Provider は、ConnectionString オブジェクト内では IBMDADB2 として示される IBM DB2 OLE DB Driver を使用します。

OLE DB .NET Data Provider でサポートされている接続ストリング・キーワードは、IBM OLE DB Provider for DB2 でサポートされている接続ストリング・キーワードと同じです。今後このプロバイダーはテストされません。IBM Data Server Provider for .NET を使用することをお勧めします。

また、OLE DB .NET Data Provider には、IBM DB2 OLE DB Provider と同じ制約事項があります。OLE DB .NET Data Provider に対しては追加の制約事項があり、それについては「*ADO.NET* および *OLE DB* アプリケーションの開発」の『OLE DB .NET Data Provider の制約事項』のトピックで説明されています。

OLE DB .NET Data Provider を使用するには、.NET Framework バージョン 2.0、3.0、または 3.5 のいずれかをインストールする必要があります。

DB2 Universal Database for AS/400® R520、R530 および R540、の場合、サーバー上で APAR ii13348 の修正を適用する必要があります。

OLE DB .NET Data Provider でサポートされているすべての接続キーワードを表 1 に示します。

表 2. 有用な、OLE DB .NET Data Provider の **ConnectionString** キーワード

キーワード	値	意味
PROVIDER	IBMDADB2	IBM OLE DB Provider for DB2 を指定します (必須)
DSN または データ・ソース	データベース別名	データベース・ディレクトリーにカタログされた DB2 データベース別名。
UID	user ID	DB2 データ・サーバーへの接続に使用するユーザー ID
PWD	password	DB2 データ・サーバーへの接続に使用するユーザー ID のパスワード

注: `ConnectionString` キーワードの完全なリストは、Microsoft 資料を参照してください。

以下に、`OleDbConnection` を作成して `SAMPLE` データベースに接続する例を示します。

```
[Visual Basic .NET]
Dim con As New OleDbConnection("Provider=IBMDADB2;" +
    "Data Source=sample;UID=userid;PWD=password;")
con.Open()

[C#]
OleDbConnection con = new OleDbConnection("Provider=IBMDADB2;" +
    "Data Source=sample;UID=userid;PWD=password;");
con.Open()
```

OLE DB .NET Data Provider の制約事項

今後 OLE DB .NET Data Provider はテストされません。 IBM Data Server Provider for .NET を使用することをお勧めします。

以下の表は、OLE DB .NET Data Provider の使用に関係した制約事項を示しています。

表 3. OLE DB .NET Data Provider の制約事項

クラスまたはフィーチャー	制約事項の説明	影響を受ける DB2 サーバー
ASCII 文字ストリーム	<p><code>DbType.AnsiString</code> または <code>DbType.AnsiStringFixedLength</code> を使用している場合、<code>OleDbParameters</code> で ASCII 文字ストリームを使用することはできません。</p> <p>OLE DB .NET Data Provider が次の例外をスローします。</p> <p>"Specified cast is not valid"</p> <p>回避策: <code>DbType.AnsiString</code> または <code>DbType.AnsiStringFixedLength</code> を使用する代わりに <code>DbType.Binary</code> を使用してください。</p>	すべて
ADORECORD	ADORECORD はサポートされていません。	すべて
ADORECORDSET および TIMESTAMP	<p>MSDN で説明されているように、<code>ADORECORDSET</code> の変化時間単位は、1 秒として解決されます。そのため、DB2 <code>Timestamp</code> 列が <code>ADORECORDSET</code> に保管されるとき、小数点未満の秒数はすべて失われます。同様に、<code>DataSet</code> に <code>ADORECORDSET</code> からデータを取り込んだ後は、<code>DataSet</code> の <code>Timestamp</code> 列に、小数点未満の秒数は含まれません。</p> <p>回避策: この回避策は DB2 Universal Database for Linux, UNIX, and Windows バージョン 8.1 フィックスバック 4 以降でのみ有効です。秒の小数部が失われるのを避けるため、次の CLI キーワードを設定できます。</p> <p><code>MAPTIMESTAMPDESCRIBE = 2</code></p> <p>このキーワードにより、<code>Timestamp</code> は <code>WCHAR(26)</code> として記述されます。このキーワードを設定するには、DB2 コマンド・ウィンドウから以下のコマンドを実行します。</p> <p><code>db2 update cli cfg for section common using MAPTIMESTAMPDESCRIBE 2</code></p>	すべて
Chapters	Chapters はサポートされていません。	すべて
キー情報	OLE DB .NET Data Provider は、 <code>IDataReader</code> を開くのと同時にキー情報を検索することはできません。	DB2 for VM/VSE

表 3. OLE DB .NET Data Provider の制約事項 (続き)

クラスまたはフィーチャー	制約事項の説明	影響を受ける DB2 サーバー
ストアード・プロシージャのキー情報	<p>OLE DB .NET Data Provider は、DB2 for Linux, UNIX, and Windows からのみ、ストアード・プロシージャによって戻された結果セットに関するキー情報を検索できます。なぜなら、Linux、UNIX、および Windows 以外のプラットフォーム用の DB2 サーバーは、ストアード・プロシージャで開かれた結果セットの拡張記述情報を戻さないからです。</p> <p>DB2 for Linux, UNIX, and Windows 上で、ストアード・プロシージャによって戻された結果セットのキー情報を検索するには、DB2 サーバーで、以下のレジストリー変数を設定する必要があります。</p> <pre>db2set DB2_APM_PERFORMANCE=8</pre> <p>このサーバー・サイドの DB2 レジストリー変数を設定すると、サーバー上で結果セットのメタデータがより長い期間使用可能になり、OLE DB が正常にキー情報を検索できるようになります。しかし、サーバーのワークロードによっては、OLE DB Provider が情報を照会するのに十分な期間メタデータが使用可能にならないことがあります。したがって、ストアード・プロシージャから戻された結果セットに関しては、キー情報を常に入手できるという保障はありません。</p> <p>CALL ステートメントに関するキー情報を検索するためには、アプリケーションが CALL ステートメントを実行しなければなりません。</p> <pre>OleDbDataAdapter.FillSchema() または OleDbCommand.ExecuteReader(CommandBehavior.SchemaOnly CommandBehavior.KeyInfo)</pre> <p>を呼び出しても、ストアード・プロシージャ呼び出しは実際には実行されません。したがって、ストアード・プロシージャによって戻される結果セットのキー情報は検索されません。</p>	すべて
バッチ SQL ステートメントからのキー情報	<p>複数の結果を戻すバッチ SQL ステートメントを使用している場合、FillSchema() メソッドは、バッチ SQL ステートメント・リストの中の最初の SQL ステートメントについてのみ、スキーマ情報の検索を試行します。このステートメントが結果セットを戻さなければ、表は作成されません。以下に例を示します。</p> <pre>[C#] cmd.CommandText = "INSERT INTO ORG(C1) VALUES(1000); SELECT C1 FROM ORG;"; da = new OleDbDataAdapter(cmd); da.FillSchema(ds, SchemaType.Source);</pre> <p>バッチ SQL ステートメントの最初のステートメントが、結果セットを戻さない INSERT ステートメントなので、データ・セット内に表は作成されません。</p>	すべて

表 3. OLE DB .NET Data Provider の制約事項 (続き)

クラスまたはフィーチャ	制約事項の説明	影響を受ける DB2 サーバー
OleDbCommandBuilder	<p>SELECT ステートメントに以下のデータ・タイプの列が含まれている場合、OleDbCommandBuilder によって自動的に生成された UPDATE、DELETE、および INSERT ステートメントは正しくありません。</p> <ul style="list-style-type: none"> • CLOB • BLOB • DBCLOB • LONG VARCHAR • LONG VARCHAR FOR BIT DATA • LONG VARGRAPHIC <p>DB2 for Linux, UNIX, and Windows 以外の DB2 サーバーに接続している場合は、以下のデータ・タイプの列も問題を引き起こします。</p> <ul style="list-style-type: none"> • VARCHAR¹ • VARCHAR FOR BIT DATA¹ • VARGRAPHIC¹ • REAL • FLOAT または DOUBLE • TIMESTAMP <p>注:</p> <p>1. これらのデータ・タイプの列は、254 バイトより大きい VARCHAR 値、254 バイトより大きい VARCHAR 値 FOR BIT DATA、または 127 バイトより大きい VARGRAPHIC として定義されている場合は適切です。この条件は、DB2 for Linux, UNIX, and Windows 以外の DB2 サーバーに接続している場合にのみ有効です。</p> <p>OleDbCommandBuilder は、WHERE 節の等価比較で選択されたすべての列を使用する SQL ステートメントを生成しますが、以前にリストされたデータ・タイプを等価比較に使用することはできません。</p> <p>注: この制限は、OleDbCommandBuilder を利用して UPDATE、DELETE、および INSERT ステートメントを自動生成する IDbDataAdapter.Update() メソッドに影響することにご注意ください。生成されたステートメントに、前にリストされたデータ・タイプがいずれか 1 つ含まれていると、UPDATE 操作は失敗します。</p> <p>回避策: 生成された SQL ステートメントの WHERE 文節から、前述のデータ・タイプのすべての列を明示的に除去することが必要になります。</p> <p>UPDATE、DELETE、および INSERT ステートメントは、自分でコーディングすることを推奨します。</p>	すべて
OleDbCommandBuilder. DeriveParameters	<p>DeriveParameters() を使用する際は、大文字小文字の区別が重要です。</p> <p>OleDbCommand.CommandText で指定されたストアド・プロシージャ名は、DB2 システム・カタログ表に保管されている名前と同じケースでなければなりません。ストアド・プロシージャ名がどのように保管されているかを確認するには、プロシージャ名制限なしで OpenSchema(OleDbSchemaGuid.Procedures) を呼び出します。これは、すべてのストアド・プロシージャ名を戻します。デフォルトで、DB2 はストアド・プロシージャ名を大文字で保管します。したがって、ほとんどの場合、ストアド・プロシージャ名は大文字で指定する必要があります。</p>	すべて

表 3. OLE DB .NET Data Provider の制約事項 (続き)

クラスまたはフィーチャー	制約事項の説明	影響を受ける DB2 サーバー
OleDbCommandBuilder. DeriveParameters	OleDbCommandBuilder.DeriveParameters() メソッドは、生成される OleDbParameterCollection 内に ReturnValue パラメーターを含めません。SqlClient および IBM Data Server Provider for .NET はデフォルトで、生成される ParameterCollection に、ParameterDirection.ReturnValue と共にパラメーターを追加します。	すべて
OleDbCommandBuilder. DeriveParameters	多重定義されたストアード・プロシージャの場合、OleDbCommandBuilder.DeriveParameters() メソッドは失敗します。MYPROC という名前の複数のストアード・プロシージャがあり、それぞれが異なる数または異なるタイプのパラメーターを取る場合、OleDbCommandBuilder.DeriveParameters() では、多重定義されているすべてのストアード・プロシージャのすべてのパラメーターが検索されます。	すべて
OleDbCommandBuilder. DeriveParameters	アプリケーションがストアード・プロシージャをスキーマで修飾していない場合、DeriveParameters() では、そのプロシージャ名のすべてのパラメーターが戻されます。したがって、同一のプロシージャ名に対して複数のスキーマが存在する場合、DeriveParameters() では、同じ名前を持つすべてのプロシージャのすべてのパラメーターが戻されます。	すべて
OleDbConnection. ChangeDatabase	OleDbConnection.ChangeDatabase() メソッドはサポートされていません。	すべて
OleDbConnection. ConnectionString	<p>接続ストリングの中で %b、%a、または %O などの印刷不可文字を使用すると、例外が出されます。</p> <p>以下のキーワードには、制約があります。</p> <p>Data Source データ・ソースは、サーバーではなく、データベースの名前です。 SERVER キーワードは指定できますが、IBMDADB2 プロバイダーからは無視されます。</p> <p>Initial Catalog および Connect Timeout これらのキーワードはサポートされていません。通常、OLE DB .NET Data Provider は、認識されない非サポート・キーワードはすべて無視します。しかし、これらのキーワードを指定すると、次の例外が発生します。 Multiple-step OLE DB operation generated errors. Check each OLE DB status value, if available. No work was done.</p> <p>ConnectionTimeout ConnectionTimeout は読み取り専用です。</p>	すべて

表 3. OLE DB .NET Data Provider の制約事項 (続き)

クラスまたはフィーチャー	制約事項の説明	影響を受ける DB2 サーバー
OleDbConnection. GetOleDbSchemaTable	<p>制限値は、大文字小文字の区別があり、システム・カタログ表に保管されているデータベース・オブジェクトとケースが一致している必要があります。デフォルトではこれは大文字です。</p> <p>例えば、次のようにして表を作成したとします。</p> <pre>CREATE TABLE abc(c1 SMALLINT)</pre> <p>DB2 は表の名前を大文字 ("ABC") でシステム・カタログに保管します。そのため、"ABC" を制限値として使用する必要があります。例えば、次のようにします。</p> <pre>schemaTable = con.GetOleDbSchemaTable(OleDbSchemaGuid.Tables, new object[] { null, null, "ABC", "TABLE" });</pre> <p>回避策: データ定義に大文字小文字の区別やスペースが必要な場合、その前後に引用符を付けてください。以下に例を示します。</p> <pre>cmd.CommandText = "create table \"Case Sensitive\"(c1 int)"; cmd.ExecuteNonQuery(); tablename = "\"Case Sensitive\""; schemaTable = con.GetOleDbSchemaTable(OleDbSchemaGuid.Tables, new object[] { null, null, tablename, "TABLE" });</pre>	すべて
OleDbDataAdapter および DataColumnMapping	<p>ソース列名には、大文字小文字の区別があります。これは、DB2 カタログに保管されるケースと一致している必要があります、それはデフォルトで大文字です。</p> <p>以下に例を示します。</p> <pre>colMap = new DataColumnMapping("EMPNO", "Employee ID");</pre>	すべて
OleDbDataReader. GetSchemaTable	<p>OLE DB .NET Data Provider は、拡張記述情報を戻さないサーバーからは、拡張記述情報を検索できません。拡張記述をサポートしていないサーバー (影響を受けるサーバー) に接続している場合、IDataReader.GetSchemaTable() から戻されたメタデータ表の中の以下の列は無効となります。</p> <ul style="list-style-type: none"> • IsReadOnly • IsUnique • IsAutoIncrement • BaseSchemaName • BaseCatalogName 	DB2 for OS/390®, V7 以前 DB2 for OS/400 DB2 for VM/VSE
ストアド・プロシージャ: 結果セットの列名はなし	<p>DB2 for OS/390 バージョン 6.1 サーバーは、ストアド・プロシージャから戻された結果セットの列名を戻しません。OLE DB .NET Data Provider は、これらの無名列を、その順序位置 (例えば、"1"、"2"、"3") にマップします。これは、MSDN で説明されているマッピング ("Column1", "Column2", "Column3") とは異なります。</p>	DB2 for OS/390 バージョン 6.1

ヒント

OLE DB .NET Data Provider アプリケーションでの接続プーリング

OLE DB .NET Data Provider は、OLE DB セッション・プーリングを使用して、自動的に接続をプールします。

OLE DB .NET Data Provider は、OLE DB セッション・プーリングを使用して、自動的に接続をプールします。プーリングを含む OLE DB サービスを使用可能または使用不可にするには、接続ストリング引数を使用します。例えば、次の接続ストリングは、OLE DB セッション・プーリングおよび自動トランザクション参加を使用不可にします。

Provider=IBMDADB2;OLE DB Services=-4;Data Source=SAMPLE;

次の表では、OLE DB サービスを設定するために使用できる、ADO 接続ストリング属性について説明しています。

表4. ADO 接続ストリング属性を使用した OLE DB サービスの設定

使用可能サービス	接続ストリングの値
すべてのサービス (デフォルト)	"OLE DB Services = -1;"
プーリングを除くすべてのサービス	"OLE DB Services = -2;"
プーリングおよび自動的参加を除くすべてのサービス	"OLE DB Services = -4;"
クライアント・カーソルを除くすべてのサービス	"OLE DB Services = -5;"
クライアント・カーソルおよびプーリングを除くすべてのサービス	"OLE DB Services = -6;"
サービスなし	"OLE DB Services = 0;"

OLE DB セッション・プーリングまたはリソース・プーリングについて、および OLE DB プロバイダー・サービスのデフォルトをオーバーライドしてプーリングを無効にする方法についての詳細は、以下のアドレスの MSDN ライブラリーの「OLE DB Programmer's Reference」を参照してください。

<http://msdn.microsoft.com/library>

OLE DB .NET Data Provider アプリケーションの時刻列

以下のセクションでは、OLE DB .NET Data Provider アプリケーションに、時刻列をインプリメントする方法について説明します。

以下のセクションでは、OLE DB .NET Data Provider アプリケーションに、時刻列をインプリメントする方法について説明します。

パラメーター・マーカーを使用した挿入

次のように、時刻列に時刻値を挿入しようとしています。

```
command.CommandText = "insert into mytable(c1) values( ? )";
```

列 c1 は時刻列です。ここでは、時刻値をパラメーター・マーカーにバインドする 2 つの方法があります。

```
OleDbParameter.OleDbType = OleDbType.DBTime を使用する
```

OleDbType.DBTime は TimeSpan オブジェクトにマップするので、TimeSpan オブジェクトをパラメーター値として指定する必要があります。パラメーター値は、String または DateTime オブジェクトにはできません。これは TimeSpan オブジェクトでなければなりません。以下に例を示します。

```
p1.OleDbType = OleDbType.DBTime;  
p1.Value = TimeSpan.Parse("0.11:20:30");  
rowsAffected = cmd.ExecuteNonQuery();
```

MSDN の資料で説明されているとおり、TimeSpan の形式は、[-]d.hh:mm:ss.ff の形式のストリングで表記されます。

`OleDbParameter.OleDbType = OleDbType.DateTime` を使用する

この場合、OLE DB .NET Data Provider は、パラメーター値を `TimeSpan` オブジェクトではなく、`DateTime` オブジェクトに変換します。したがって、パラメーター値は、`DateTime` オブジェクトに変換できる任意の有効なストリング/オブジェクトにできます。つまり、11:20:30 などの値を用いることができます。値を `DateTime` オブジェクトにすることもできます。値を `TimeSpan` オブジェクトにすることはできません。なぜなら、`TimeSpan` オブジェクトは `DateTime` オブジェクトに変換できないからです。`TimeSpan` は `IConvertible` をインプリメントしていません。

以下に例を示します。

```
p1.OleDbType = OleDbType.DBTimeStamp;  
p1.Value = "11:20:30";  
rowsAffected = cmd.ExecuteNonQuery();
```

検索

時刻列を検索するには、`IDataRecord.GetValue()` メソッドまたは `OleDbDataReader.GetTimeSpan()` メソッドを使用する必要があります。

以下に例を示します。

```
TimeSpan ts1 = ((OleDbDataReader)reader).GetTimeSpan( 0 );  
TimeSpan ts2 = (TimeSpan) reader.GetValue( 0 );
```

OLE DB .NET Data Provider アプリケーションの ADORecordset オブジェクト

`ADORecordset` オブジェクトを使用するとき、ADO タイプの `adDBTime` クラスを考慮する必要があります。

`ADORecordset` オブジェクトの使用に関する考慮事項を以下に示します。

- ADO タイプ `adDBTime` クラスは、.NET Framework の `DateTime` クラスにマップされます。`OleDbType.DBTime` は、`TimeSpan` オブジェクトに対応します。
- `TimeSpan` オブジェクトを `ADORecordset` オブジェクトの `Time` フィールドに割り当てることはできません。なぜなら、`ADORecordset` オブジェクトの `Time` フィールドでは、`DateTime` オブジェクトが期待されているからです。`TimeSpan` オブジェクトを `ADORecordset` オブジェクトに割り当てると、次のメッセージが表示されます。

```
Method's type signature is not Interop compatible.
```

`Time` フィールドに入れることができるのは、`DateTime` オブジェクトか、`DateTime` オブジェクトに解析できる `String` のみです。

- `OleDbDataAdapter` を使用して `DataSet` に `ADORecordset` を取り込むとき、`ADORecordset` の `Time` フィールドは、`DataSet` の `TimeSpan` 列に変換されません。
- `Recordsets` には主キーや制約は保管されません。したがって、`MissingSchemaAction.AddWithKey` を使用して `DataSet` に `Recordset` からデータを取り込む際は、キー情報は追加されません。

ODBC .NET Data Provider

ODBC .NET Data Provider は、CLI ドライバーを使用して、DB2 データ・ソースに対して ODBC 呼び出しを行います。したがって、ODBC .NET Data Provider がサポートする接続ストリング・キーワードは、CLI ドライバーがサポートする接続ストリング・キーワードと同じです。今後このプロバイダーはテストされません。IBM Data Server Provider for .NET を使用することをお勧めします。

また、ODBC .NET Data Provider には、CLI ドライバーと同じ制約事項があります。ODBC .NET Data Provider に対しては追加の制約事項があり、それについては「ADO.NET および OLE DB アプリケーションの開発」の『ODBC .NET Data Provider の制約事項』のトピックで説明されています。

ODBC .NET Data Provider を使用するには、.NET Framework バージョン 2.0、3.0、または 3.5 のいずれかをインストールする必要があります。DB2 Universal Database for AS/400 V5R4 以前の場合、サーバー上で APAR II13348 の修正を適用する必要があります。

ODBC .NET Data Provider でサポートされている接続キーワードを表 1 に示します。

表 5. 有用な、ODBC .NET Data Provider の **ConnectionString** キーワード

キーワード	値	意味
DSN	データベース別名	データベース・ディレクトリーにカタログされた DB2 データベース別名。
UID	user ID	DB2 サーバーへの接続に使用するユーザー ID
PWD	password	DB2 サーバーへの接続に使用するユーザー ID のパスワード

注: **ConnectionString** キーワードの完全なリストは、Microsoft 資料を参照してください。

以下のコードに、OdbcConnection を作成して SAMPLE データベースに接続する例を示します。

```
[Visual Basic .NET]
Dim con As New OdbcConnection("DSN=sample;UID=userid;PWD=password;")
con.Open()
```

```
[C#]
OdbcConnection con = new OdbcConnection("DSN=sample;UID=userid;PWD=password;");
con.Open()
```

ODBC .NET Data Provider の制約事項

今後 ODBC .NET Data Provider はテストされません。IBM Data Server Provider for .NET を使用することをお勧めします。

以下の表は、ODBC .NET Data Provider の使用に関係した制約事項を示しています。

表 6. ODBC .NET Data Provider の制約事項

クラスまたはフィーチャー	制約事項の説明	影響を受ける DB2 サーバー
ASCII 文字ストリーム	<p>DbType.AnsiString または DbType.AnsiStringFixedLength を使用している場合、OdbcParameters で ASCII 文字ストリームを使用することはできません。</p> <p>ODBC .NET Data Provider が次の例外をスローします。</p> <p>"Specified cast is not valid"</p> <p>回避策: DbType.AnsiString または DbType.AnsiStringFixedLength を使用する代わりに DbType.Binary を使用します。</p>	すべて
Command.Prepare	<p>最後の準備の後に CommandText が変更されている場合は、コマンド (Command.ExecuteNonQuery または Command.ExecuteReader) を実行する前に、OdbcCommand.Prepare() を明示的に実行する必要があります。</p> <p>OdbcCommand.Prepare() を再び呼び出さないと、ODBC .NET Data Provider は以前に準備された CommandText を実行します。</p> <p>以下に例を示します。</p> <pre data-bbox="431 743 997 932">[C#] command.CommandText="select CLOB('ABC') from table1"; command.Prepare(); command.ExecuteReader(); command.CommandText="select CLOB('XYZ') from table2"; // This ends up re-executing the first statement command.ExecuteReader();</pre>	すべて

表 6. ODBC .NET Data Provider の制約事項 (続き)

クラスまたはフィーチャー	制約事項の説明	影響を受ける DB2 サーバー
CommandBehavior.SequentialAccess	<p>CommandBehavior.SequentialAccess で作成されたリーダーから、IDataReader.GetChars() を使用して読み取る場合は、列全体を保持するのに十分な大きさのバッファを割り振る必要があります。そうしないと、次の例外が発生します。</p> <pre>Requested range extends past the end of the array. at System.Runtime.InteropServices.Marshal.Copy(Int32 source, Char[] destination, Int32 startIndex, Int32 length) at System.Data.Odbc.OdbcDataReader.GetChars(Int32 i, Int64 dataIndex, Char[] buffer, Int32 bufferSize, Int32 length) at OleDbRestrict.TestGetCharsAndBufferSize(IDbConnection con)</pre> <p>次の例で、十分なバッファを割り振る方法を示します。</p> <pre>CREATE TABLE myTable(c0 int, c1 CLOB(10K)) SELECT c1 FROM myTable;</pre> <pre>[C#] cmd.CommandText = "SELECT c1 from myTable"; IDataReader reader = cmd.ExecuteReader(CommandBehavior.SequentialAccess); Int32 iChunkSize = 10; Int32 iBufferSize = 10; Int32 iFieldOffset = 0; Char[] buffer = new Char[iBufferSize]; reader.Read(); reader.GetChars(0, iFieldOffset, buffer, 0, iChunkSize);</pre> <p>GetChars() を呼び出すと、次の例外が出されます。</p> <pre>"Requested range extends past the end of the array"</pre> <p>GetChars() によって前述の例外が出されないようにするためには、次のようにして、BufferSize に列のサイズを設定する必要があります。</p> <pre>Int32 iBufferSize = 10000;</pre> <p>iBufferSize の値 10,000 は、CLOB 列 c1 に割り振られている値 10K と対応します。</p>	すべて
CommandBehavior.SequentialAccess	<p>ODBC .NET Data Provider は、OdbcDataReader.GetChars() を使用しているときに読み取るデータがなくなると、次の例外を出します。</p> <pre>NO_DATA - no error information available at System.Data.Odbc.OdbcConnection.HandleError(HandleRef hrHandle, SQL_HANDLE hType, RETCODE retcode) at System.Data.Odbc.OdbcDataReader.GetData(Int32 i, SQL_C sqlctype, Int32 cb) at System.Data.Odbc.OdbcDataReader.GetChars(Int32 i, Int64 dataIndex, Char[] buffer, Int32 bufferSize, Int32 length)</pre>	すべて
CommandBehavior.SequentialAccess	<p>OdbcDataReader.GetChars() を使用するときは、値 5000 などの大きなチャンク・サイズは使用できません。大きなチャンク・サイズを使用しようとすると、ODBC .NET Data Provider が次の例外をスローします。</p> <pre>Object reference not set to an instance of an object. at System.Runtime.InteropServices.Marshal.Copy(Int32 source, Char[] destination, Int32 startIndex, Int32 length) at System.Data.Odbc.OdbcDataReader.GetChars(Int32 i, Int64 dataIndex, Char[] buffer, Int32 bufferSize, Int32 length) at OleDbRestrict.TestGetCharsAndBufferSize(IDbConnection con)</pre>	すべて

表 6. ODBC .NET Data Provider の制約事項 (続き)

クラスまたはフィーチャー	制約事項の説明	影響を受ける DB2 サーバー
接続プール	<p>ODBC .NET Data Provider は接続プーリングを制御しません。接続プーリングは、ODBC Driver Manager によって取り扱われます。接続プーリングについて詳しくは、MSDN ライブラリーの「ODBC Programmer's Reference」を参照してください。MSDN ライブラリーの URL は以下のとおりです。</p> <p>http://msdn.microsoft.com/library</p>	すべて
DataColumnMapping	<p>ソース列名の大文字小文字は、システム・カタログ表で使用されている大文字小文字と一致している必要があります。これは、デフォルトで大文字です。</p>	すべて
10 進列	<p>10 進列では、パラメーター・マーカはサポートされません。</p> <p>通常、ターゲット SQLType が Decimal 列の場合は、OdbcParameter に OdbcType.Decimal を使用します。しかし、ODBC .NET Data Provider は OdbcType.Decimal を見付けると、SQL_C_WCHAR の C タイプと SQL_VARCHAR の SQLType を使用してパラメーターをバインドし、それは無効となります。</p> <p>以下に例を示します。</p> <pre>[C#] cmd.CommandText = "SELECT dec_col FROM MYTABLE WHERE dec_col > ? "; OdbcParameter p1 = cmd.CreateParameter(); p1.DbType = DbType.Decimal; p1.Value = 10.0; cmd.Parameters.Add(p1); IDataReader rdr = cmd.ExecuteReader();</pre> <p>次の例外が戻されます。</p> <pre>ERROR [07006] [IBM][CLI Driver][SQLDS/VM] SQL0301N The value of input host variable or parameter number "" cannot be used because of its data type. SQLSTATE=07006</pre> <p>回避策: OdbcParameter 値を使用するのではなく、リテラルのみを使用してください。</p>	DB2 for VM/VSE
キー情報	<p>表名を修飾するために使用されるスキーマ名 (例えば、MYSHEMA.MYTABLE) は、接続ユーザー ID と一致していなければなりません。ODBC .NET Data Provider は、指定されたスキーマが接続ユーザー ID と異なるキー情報は検索できません。</p> <p>以下に例を示します。</p> <pre>CREATE TABLE USERID2.TABLE1(c1 INT NOT NULL PRIMARY KEY);</pre> <pre>[C#] // Connect as user bob odbcCon = new OdbcConnection("DSN=sample;UID=bob;PWD=mypassword"); OdbcCommand cmd = odbcCon.CreateCommand(); // Select from table with schema USERID2 cmd.CommandText="SELECT * FROM USERID2.TABLE1"; // Fails - No key info retrieved da.FillSchema(ds, SchemaType.Source); // Fails - SchemaTable has no primary key cmd.ExecuteReader(CommandBehavior.KeyInfo) // Throws exception because no primary key cbuilder.GetUpdateCommand();</pre>	すべて

表 6. ODBC .NET Data Provider の制約事項 (続き)

クラスまたはフィーチャー	制約事項の説明	影響を受ける DB2 サーバー
キー情報	<p>ODBC .NET Data Provider は、IDataReader を開くのと同時にキー情報を検索することはできません。ODBC .NET Data Provider が IDataReader を開くと、サーバー上でカーソルが開きます。キー情報が要求された場合、これは SQLPrimaryKeys() または SQLStatistic() を呼び出してキー情報を取得しますが、これらのスキーマ関数は他のカーソルを開きます。DB2 for VM/VSE はカーソル保留をサポートしていないため、最初のカーソルはクローズされます。その結果、IDataReader.Read() が IDataReader を呼び出すと、次の例外が発生します。</p> <pre>System.Data.Odbc.OdbcException: ERROR [HY010] [IBM][CLI Driver] CLI0125E Function sequence error. SQLSTATE=HY010</pre> <p>回避策: 最初にキー情報を取得してからデータを取得する必要があります。以下に例を示します。</p> <pre>[C#] OdbcCommand cmd = odbcCon.CreateCommand(); OdbcDataAdapter da = new OdbcDataAdapter(cmd); cmd.CommandText = "SELECT * FROM MYTABLE"; // Use FillSchema to retrieve just the schema information da.FillSchema(ds, SchemaType.Source); // Use FillSchema to retrieve just the schema information da.Fill(ds);</pre>	DB2 for VM/VSE
キー情報	<p>SQL ステートメントの中では、データベース・オブジェクトは、データベース・オブジェクトをシステム・カタログ表に保管するのに用いられているケース (大文字小文字) と同じケースを使用して参照する必要があります。デフォルトでは、データベース・オブジェクトは大文字でシステム・カタログ表に保管されるので、ほとんどの場合は、大文字を使用する必要があります。</p> <p>ODBC .NET Data Provider は、SQL ステートメントをスキャンしてデータベース・オブジェクト名を検索し、それらを、システム・カタログ表内のこれらのオブジェクトについての照会を発行する SQLPrimaryKeys および SQLStatistics などのスキーマ関数に渡します。データベース・オブジェクトの参照は、システム・カタログ表にそれらが保管されている状態と完全に一致していなければなりません。そうでないと、空の結果セットが戻されます。</p>	DB2 for OS/390 DB2 for OS/400 DB2 for VM/VSE
バッチの非選択 SQL ステートメントのキー情報	ODBC .NET Data Provider は、SELECT で始まっていないバッチ・ステートメントのキー情報は検索できません。	DB2 for OS/390 DB2 for OS/400 DB2 for VM/VSE

表 6. ODBC .NET Data Provider の制約事項 (続き)

クラスまたはフィーチャー	制約事項の説明	影響を受ける DB2 サーバー
LOB 列	<p>ODBC .NET Data Provider は LOB データ・タイプをサポートしていません。そのため、DB2 サーバーが SQL_CLOB (-99)、SQL_BLOB (-98)、または SQL_DBCLOB (-350) を戻すと、 ODBC .NET Data Provider は次の例外を出します。</p> <p>"Unknown SQL type - -98" (Blob 列の場合) "Unknown SQL type - -99" (Clob 列の場合) "Unknown SQL type - -350" (DbClob 列の場合)</p> <p>直接または間接的に LOB 列にアクセスするメソッドはどれも失敗します。</p> <p>回避策: CLI/ODBC LongDataCompat キーワードを 1 に設定します。そのようにすると、CLI ドライバーは、ODBC .NET Data Provider によって認識されるデータ・タイプへの、次のデータ・タイプ・マッピングを行います。</p> <ul style="list-style-type: none"> • SQL_CLOB から SQL_LONGVARCHAR へ • SQL_BLOB から SQL_LONGVARBINARY へ • SQL_DBCLOB から SQL_WLONGVARCHAR へ <p>LongDataCompat キーワードを設定するには、クライアント・マシンの DB2 コマンド・ウィンドウから次の DB2 コマンドを実行してください。</p> <pre>db2 update cli cfg for section common using longdatacompat 1</pre> <p>次のように接続ストリングを使用して、このキーワードをアプリケーションに設定することもできます。</p> <pre>[C#] OdbcConnection con = new OdbcConnection("DSN=SAMPLE;UID=uid;PWD=mypwd;LONGDATACOMPAT=1;");</pre> <p>すべての CLI/ODBC キーワードのリストについては、「DB2 CLI ガイドおよびリファレンス」の『UID CLI/ODBC 構成キーワード』を参照してください。</p>	すべて
OdbcCommand.Cancel	<p>OdbcCommand.Cancel の実行後にステートメントを実行すると、次の例外が発生することがあります。</p> <pre>"ERROR [24000] [Microsoft][ODBC Driver Manager] Invalid cursor state"</pre>	すべて
OdbcCommandBuilder	<p>OdbcCommandBuilder は、エスケープ文字をサポートしないサーバーに対するコマンドの生成に失敗します。OdbcCommandBuilder は、コマンドを生成するときに、まず SQLGetInfo を呼び出して、SQL_SEARCH_PATTERN_ESCAPE 属性を要求します。サーバーがエスケープ文字をサポートしていない場合は空ストリングが戻され、ODBC .NET Data Provider は次の例外をスローします。</p> <pre>Index was outside the bounds of the array. at System.Data.Odbc.OdbcConnection.get_EscapeChar() at System.Data.Odbc.OdbcDataReader.GetTableNameFromCommandText() at System.Data.Odbc.OdbcDataReader.BuildMetaDataInfo() at System.Data.Odbc.OdbcDataReader.GetSchemaTable() at System.Data.Common.CommandBuilder.BuildCache(Boolean closeConnection) at System.Data.Odbc.OdbcCommandBuilder.GetUpdateCommand()</pre>	DB2 for OS/390、DBCS サーバーのみ; DB2 for VM/VSE、DBCS サーバーのみ

表 6. ODBC .NET Data Provider の制約事項 (続き)

クラスまたはフィーチャー	制約事項の説明	影響を受ける DB2 サーバー
OdbcCommandBuilder	<p>OdbcCommandBuilder を使用して UPDATE、DELETE、および INSERT ステートメントを自動生成する場合、大文字小文字の区別は重要です。システム・カタログ表が、(作成時にデータベース・オブジェクトの前後に引用符を追加することによって) 明示的に大文字小文字の区別付きで作成されているのでない限り、デフォルトで DB2 はスキーマ情報 (表名や列名など) を大文字でシステム・カタログ表に保管します。したがって、SQL ステートメントは、カタログに保管されているケース (デフォルトでは大文字) と一致している必要があります。</p> <p>例えば、次のステートメントを使用して表を作成したとします。</p> <pre>"db2 create table mytable (c1 int) "</pre> <p>この場合、DB2 は表名 mytable をシステム・カタログ表に MYTABLE として保管します。</p> <p>次のコード例は、OdbcCommandBuilder クラスの正しい使用法を例示しています。</p> <pre>[C#] OdbcCommand cmd = odbcCon.CreateCommand(); cmd.CommandText = "SELECT * FROM MYTABLE"; OdbcDataAdapter da = new OdbcDataAdapter(cmd); OdbcCommandBuilder cb = new OdbcCommandBuilder(da); OdbcCommand updateCmd = cb.GetUpdateCommand();</pre> <p>この例では、表名を大文字で参照しないと、次の例外を受け取ります。</p> <pre>"Dynamic SQL generation for the UpdateCommand is not supported against a SelectCommand that does not return any key column information."</pre>	すべて
OdbcCommandBuilder	<p>SELECT ステートメントに以下の列データ・タイプが含まれている場合、OdbcCommandBuilder によって生成されたコマンドは正しくありません。</p> <pre>REAL FLOAT または DOUBLE TIMESTAMP</pre> <p>これらのデータ・タイプは、SELECT ステートメントの WHERE 節では使用できません。</p>	DB2 for OS/390 DB2 for OS/400 DB2 for VM/VSE
OdbcCommandBuilder. DeriveParameters	<p>DeriveParameters() メソッドは、SQLProcedureColumns にマップされ、ストアド・プロシージャの名前に CommandText プロパティを使用します。CommandText には、(完全な ODBC 呼び出し構文を使用する) ストアド・プロシージャの名前は含まれないので、SQLProcedureColumns は、ODBC 呼び出し構文に基づいて識別されたプロシージャ名で呼び出されます。以下に例を示します。</p> <pre>"{ CALL myProc(?) }"</pre> <p>この結果は、空の結果セットとなり、ここからはプロシージャの列は見付かりません。</p>	すべて
OdbcCommandBuilder. DeriveParameters	<p>DeriveParameters() を使用するには、CommandText にストアド・プロシージャの名前を指定します (例えば、cmd.CommandText = "MYPROC")。プロシージャ名は、システム・カタログ表に保管されているケースと一致していなければなりません。DeriveParameters() では、システム・カタログ表内で見付かる、そのプロシージャ名のすべてのパラメーターが戻されます。ステートメントを実行する前に、CommandText を元の完全な ODBC 呼び出し構文に戻すのを忘れないでください。</p>	すべて
OdbcCommandBuilder. DeriveParameters	<p>ODBC .NET Data Provider に関しては、ReturnValue パラメーターは戻されません。</p>	すべて

表 6. ODBC .NET Data Provider の制約事項 (続き)

クラスまたはフィーチャー	制約事項の説明	影響を受ける DB2 サーバー
OdbcCommandBuilder. DeriveParameters	DeriveParameters() は、完全修飾ストアド・プロシージャ名をサポートしていません。例えば、CommandText = "MYSCHEMA.MYPROC" に対して DeriveParameters() を呼び出すと失敗します。このとき、パラメーターは戻されません。	すべて
OdbcCommandBuilder. DeriveParameters	DeriveParameters() は、多重定義のストアド・プロシージャに対しては機能しません。SQLProcedureColumns は、ストアド・プロシージャのすべてのバージョンのすべてのパラメーターを戻します。	すべて
OdbcConnection. ChangeDatabase	OdbcConnection.ChangeDatabase() メソッドはサポートされていません。	すべて
OdbcConnection. ConnectionString	<ul style="list-style-type: none"> Server キーワードは無視されます。 Connect Timeout キーワードは無視されます。CLI は接続タイムアウトをサポートしていないので、このプロパティを設定しても、ドライバに影響はありません。 接続プーリング・キーワードは無視されます。具体的には、これは Pooling、Min Pool Size、Max Pool Size、Connection Lifetime、および Connection Reset というキーワードに影響します。 	すべて
OdbcDataReader. GetSchemaTable	<p>ODBC .NET Data Provider は、拡張記述情報を戻さないサーバーからは、拡張記述情報を検索できません。そのため、拡張記述をサポートしていないサーバー (影響を受けるサーバー) に接続している場合、IDataReader.GetSchemaTable() から戻されたメタデータ表の中の以下の列は無効となります。</p> <ul style="list-style-type: none"> IsReadOnly IsUnique IsAutoIncrement BaseSchemaName BaseCatalogName 	DB2 for OS/390, バージョン 7 以降 DB2 for OS/400 DB2 for VM/VSE
ストアド・プロシ ジャー	<p>ストアド・プロシージャを呼び出すには、完全な ODBC 呼び出し構文を指定する必要があります。</p> <p>例えば、VARCHAR(10) をパラメーターとして取るストアド・プロシージャ MYPROC を呼び出すには、次のようにします。</p> <pre>[C#] OdbcCommand cmd = odbcCon.CreateCommand(); cmd.CommandType = CommandType.Text; cmd.CommandText = "{ CALL MYPROC(?) }"; OdbcParameter p1 = cmd.CreateParameter(); p1.Value = "Joe"; p1.OdbcType = OdbcType.NVarChar; cmd.Parameters.Add(p1); cmd.ExecuteNonQuery();</pre> <p>注: CommandType.StoredProcedure を使用している場合であっても、完全な ODBC 呼び出し構文を使用する必要があります。これは、MSDN の中の OdbcCommand.CommandText プロパティの項で説明されています。</p>	すべて
ストアド・プロシ ジャー: 結果セットの列 名はなし	DB2 for OS/390 バージョン 6.1 サーバーは、ストアド・プロシージャから戻された結果セットの列名を戻しません。ODBC .NET Data Provider は、これらの無名列を、その順序位置 (例えば、"1"、"2"、"3") にマップします。これは、MSDN で説明されているマッピング ("Column1", "Column2", "Column3") とは異なります。	DB2 for OS/390 バ ージョン 6.1
ユニーク索引の主キー へのプロモーション	ODBC .NET Data Provider は、NULL 可能なユニーク索引を主キーにプロモートします。これは MSDN の説明と異なります。MSDN では、NULL 可能なユニーク索引は主キーにプロモートすべきでないと言明されています。	すべて

第 2 章 IBM OLE DB Provider for DB2

IBM OLE DB Provider for DB2 を使用すれば、DB2 は OLE DB Provider のリソース・マネージャーとして機能できます。このサポートにより、OLE DB2 ベースのアプリケーションは、OLE インターフェースを使用して DB2 データの抽出や照会が可能です。

Microsoft OLE DB は、さまざまな情報ソースに保管されているデータに対し、単一アクセスをアプリケーションに提供する、OLE/COM インターフェースのセットです。OLE DB のアーキテクチャーでは、OLE DB Consumer と OLE DB Provider を定義しています。OLE DB Consumer は、OLE DB インターフェースを使用するシステムまたはアプリケーションで、OLE DB Provider は、OLE DB インターフェースを公開するコンポーネントです。

IBM OLE DB Provider for DB2 (Provider 名は IBMDADB2) を使用すれば、OLE DB Consumer は DB2 データベース・サーバー上のデータにアクセスできます。DB2 Connect がインストールされていれば、これらの OLE DB consumer は、DB2 for z/OS、DB2 Server for VM and VSE、または DB2 Universal Database for AS/400 などのホスト DBMS 上のデータにもアクセスすることができます。

IBM OLE DB Provider for DB2 には以下の機能が備わっています。

- OLE DB Provider 仕様のサポート・レベル 0。いくつかの付加的なレベル 1 インターフェースが含まれます。
- フリー・スレッド Provider のインプリメンテーション。アプリケーションは、1 つのスレッドで作成したコンポーネントを他の任意のスレッドで使用できます。
- DB2 エラー・メッセージを戻すエラー検索サービス。

IBM OLE DB Provider はクライアントに存在し、OLE DB 表関数 (これも DB2 データベース・システムでサポートされる) とは異なるものであることに注意してください。

本書の以下の節では、IBM OLE DB Provider for DB2 固有のインプリメンテーションについて説明します。Microsoft OLE DB 2.0 仕様の詳細については Microsoft Press から出版されている「Microsoft OLE DB 2.0 Programmer's Reference and Data Access SDK」を参照してください。

準拠バージョン

IBM OLE DB Provider for DB2 は、Microsoft OLE DB 仕様のバージョン 2.7 以降に準拠しています。

システム要件

サポートされている Windows オペレーティング・システムについては、IBM OLE DB Provider for DB2 データ・サーバーに関するアナウンス・レターを参照してください。

IBM OLE DB Provider for DB2 をインストールするには、上記のサポートされているオペレーティング・システムのいずれかをまず実行している必要があります。さらに DB2 製品全体、IBM Data Server Driver for ODBC and CLI、または IBM Data Server Driver Package をインストールする必要があります。

IBM OLE DB Provider for DB2 でサポートされているアプリケーション・タイプ

IBM OLE DB Provider for DB2 を使用して、Microsoft Visual Studio C++ または Visual Basics のアプリケーションを作成できます。

IBM OLE DB Provider for DB2 を使用して、以下のタイプのアプリケーションを作成できます。

- ADO アプリケーション。以下のものが含まれます。
 - Microsoft Visual Studio C++ アプリケーション
 - Microsoft Visual Basic アプリケーション
- OLE DB .NET Data Provider を使用する ADO.NET アプリケーション
- OLE DB インターフェースを使用して IBMDADB2 に直接アクセスする C/C++ アプリケーション。Data Access Consumer Object が ATL COM AppWizard によって生成される ATL アプリケーションが含まれます。

OLE DB サービス

IBM OLE DB Provider でサポートされているスレッド・モデル

IBM OLE DB Provider for DB2 では、フリー・スレッド・モデルがサポートされています。このモデルでは、アプリケーションが、1 つのスレッドで作成したコンポーネントを他の任意のスレッドで使用できます。

IBM OLE DB Provider for DB2 では、フリー・スレッド・モデルがサポートされています。このモデルでは、アプリケーションが、1 つのスレッドで作成したコンポーネントを他の任意のスレッドで使用できます。

IBM OLE DB Provider によるラージ・オブジェクトの操作

IBMDADB2 プロバイダーでデータをストレージ・オブジェクト (DBTYPE_IUNKNOWN) として取得したり設定したりするには、ISequentialStream インターフェースを使用します。

IBMDADB2 プロバイダーでデータをストレージ・オブジェクト (DBTYPE_IUNKNOWN) として取得したり設定したりするには、以下のように ISequentialStream インターフェースを使用します。

- ストレージ・オブジェクトをパラメーターにバインドするには、DBBINDING 構造の DBOBJECT で、dwFlag フィールドに値 STGM_READ だけを含めることができます。IBMDADB2 では、バインド済みオブジェクトの ISequentialStream インターフェースの Read メソッドが実行されます。

- ストレージ・オブジェクトからデータを取得するには、アプリケーションがそのストレージ・オブジェクトの ISequentialStream インターフェースで Read メソッドを実行する必要があります。
- データを取得したとき、長さ部分の値は実データの長さであり、IUnknown ポインターの長さではありません。

IBM OLE DB Provider でサポートされているスキーマ行セット

以下の表は、IDBSchemaRowset によってサポートされているスキーマ行セットを示しています。

以下の表は、IDBSchemaRowset によってサポートされているスキーマ行セットを示しています。行セットのうちサポートされていない列は NULL に設定されます。

表7. IBM OLE DB Provider for DB2 でサポートされているスキーマ行セット

サポートされている GUID	サポートされている制限	サポートされている列	注
DBSCHEMA_COLUMN_PRIVILEGES	COLUMN_NAME TABLE_NAME TABLE_SCHEMA	COLUMN_NAME GRANTEE GRANTOR IS_GRANTABLE PRIVILEGE_TYPE TABLE_NAME TABLE_SCHEMA	
DBSCHEMA_COLUMNS	COLUMN_NAME TABLE_NAME TABLE_SCHEMA	CHARACTER_MAXIMUM_LENGTH CHARACTER_OCTET_LENGTH COLUMN_DEFAULT COLUMN_FLAGS COLUMN_HASDEFAULT COLUMN_NAME DATA_TYPE DESCRIPTION IS_NULLABLE NUMERIC_PRECISION NUMERIC_SCALE ORDINAL_POSITION TABLE_NAME TABLE_SCHEMA	
DBSCHEMA_FOREIGN_KEYS	FK_TABLE_NAME FK_TABLE_SCHEMA PK_TABLE_NAME PK_TABLE_SCHEMA	DEFERRABILITY DELETE_RULE FK_COLUMN_NAME FK_NAME FK_TABLE_NAME FK_TABLE_SCHEMA ORDINAL PK_COLUMN_NAME PK_NAME PK_TABLE_NAME PK_TABLE_SCHEMA UPDATE_RULE	PK_TABLE_NAME または FK_TABLE_NAME の制限のうち少なくとも 1 つを指定する必要があります。 %" ワイルドカードは使用できません。

表 7. IBM OLE DB Provider for DB2 でサポートされているスキーマ行セット (続き)

サポートされている GUID	サポートされている制限	サポートされている列	注
DBSCHEMA_INDEXES	TABLE_NAME TABLE_SCHEMA	CARDINALITY CLUSTERED COLLATION COLUMN_NAME INDEX_NAME INDEX_SCHEMA ORDINAL_POSITION PAGES TABLE_NAME TABLE_SCHEMA TYPE UNIQUE	ソート順序はサポートされていません。指定されているソート順序は無視されます。
DBSCHEMA_PRIMARY_KEYS	TABLE_NAME TABLE_SCHEMA	COLUMN_NAME ORDINAL PK_NAME TABLE_NAME TABLE_SCHEMA	少なくとも TABLE_NAME の制限を指定する必要があります。 "%" ワイルドカードは使用できません。
DBSCHEMA _PROCEDURE_PARAMETERS	PARAMETER_NAME PROCEDURE_NAME PROCEDURE_SCHEMA	CHARACTER_MAXIMUM_LENGTH CHARACTER_OCTET_LENGTH DATA_TYPE DESCRIPTION IS_NULLABLE NUMERIC_PRECISION NUMERIC_SCALE ORDINAL_POSITION PARAMETER_DEFAULT PARAMETER_HASDEFAULT PARAMETER_NAME PARAMETER_TYPE PROCEDURE_NAME PROCEDURE_SCHEMA TYPE_NAME	
DBSCHEMA_PROCEDURES	PROCEDURE_NAME PROCEDURE_SCHEMA	DESCRIPTION PROCEDURE_NAME PROCEDURE_SCHEMA PROCEDURE_TYPE	

表7. IBM OLE DB Provider for DB2 でサポートされているスキーマ行セット (続き)

サポートされている GUID	サポートされている制限	サポートされている列	注
DBSCHEMA_PROVIDER_TYPES	DATA_TYPE BEST_MATCH	AUTO_UNIQUE_VALUE BEST_MATCH CASE_SENSITIVE CREATE_PARAMS COLUMN_SIZE DATA_TYPE FIXED_PREC_SCALE IS_FIXEDLENGTH IS_LONG IS_NULLABLE LITERAL_PREFIX LITERAL_SUFFIX LOCAL_TYPE_NAME MINIMUM_SCALE MAXIMUM_SCALE SEARCHABLE TYPE_NAME UNSIGNED_ATTRIBUTE	
DBSCHEMA_STATISTICS	TABLE_NAME TABLE_SCHEMA	CARDINALITY TABLE_NAME TABLE_SCHEMA	ソート順序はサポートされていません。指定されているソート順序は無視されます。
DBSCHEMA _TABLE_PRIVILEGES	TABLE_NAME TABLE_SCHEMA	GRANTEE GRANTOR IS_GRANTABLE PRIVILEGE_TYPE TABLE_NAME TABLE_SCHEMA	
DBSCHEMA_TABLES	TABLE_NAME TABLE_SCHEMA TABLE_TYPE	DESCRIPTION TABLE_NAME TABLE_SCHEMA TABLE_TYPE	

IBM OLE DB Provider で自動的に使用可能になる OLE DB サービス

デフォルトでは、IBM OLE DB Provider for DB2 は、プロバイダーのクラス ID (CLSID) の下に DWORD 値が 0xFFFFFFFF であるレジストリー項目 OLEDB_SERVICES を追加することにより、すべての OLE DB サービスを自動的に有効にします。

デフォルトでは、IBM OLE DB Provider for DB2 は、プロバイダーのクラス ID (CLSID) の下に DWORD 値が 0xFFFFFFFF であるレジストリー項目 OLEDB_SERVICES を追加することにより、すべての OLE DB サービスを自動的に有効にします。この値の意味は以下のとおりです。

表 8. OLE DB サービス

使用可能になっているサービス	DWORD 値
すべてのサービス (デフォルト)	0xFFFFFFFF
プールおよび AutoEnlistment を除くすべて	0xFFFFFFFFC
クライアント・カーソルを除くすべて	0xFFFFFFFFB
プール、エンリスト、およびカーソルを除くすべて	0xFFFFFFFF8
サービスなし	0x00000000

データ・サービス

IBM OLE DB Provider でサポートされているカーソル・モード

IBM OLE DB Provider for DB2 では、読み取り専用カーソル、前方スクロール・カーソル、読み取り専用の両方向スクロール・カーソル、更新可能な両方向スクロール・カーソルがネイティブでサポートされています。

IBM OLE DB Provider for DB2 では、読み取り専用カーソル、前方スクロール・カーソル、読み取り専用の両方向スクロール・カーソル、更新可能な両方向スクロール・カーソルがネイティブでサポートされています。

DB2 と OLE DB の間のデータ・タイプ・マッピング

IBM OLE DB Provider for DB2 では、DB2 データ・タイプと OLE DB データ・タイプの間のデータ・タイプ・マッピングがサポートされています。

以下の表は、サポートされているマッピングと使用可能な名前の完全なリストを示すことにより、列およびパラメーターのデータ・タイプを示しています。

表 9. DB2 データ・タイプと OLE DB データ・タイプの間のデータ・タイプ・マッピング

DB2 データ・タイプ	OLE DB データ・タイプ標識	OLE DB 標準タイプ名	DB2 固有の名前
SMALLINT	DBTYPE_I2	"DBTYPE_I2"	"SMALLINT"
INTEGER	DBTYPE_I4	"DBTYPE_I4"	"INTEGER" または "INT"
BIGINT	DBTYPE_I8	"DBTYPE_I8"	"BIGINT"
REAL	DBTYPE_R4	"DBTYPE_R4"	"REAL"
FLOAT	DBTYPE_R8	"DBTYPE_R8"	"FLOAT"
DOUBLE	DBTYPE_R8	"DBTYPE_R8"	"DOUBLE" または "DOUBLE PRECISION"
DECIMAL	DBTYPE_NUMERIC	"DBTYPE_NUMERIC"	"DEC" または "DECIMAL"
NUMERIC	DBTYPE_NUMERIC	"DBTYPE_NUMERIC"	"NUM" または "NUMERIC"
DATE	DBTYPE_DBDATE	"DBTYPE_DBDATE"	"DATE"
TIME	DBTYPE_DBTIME	"DBTYPE_DBTIME"	"TIME"

表9. DB2 データ・タイプと OLE DB データ・タイプの間のデータ・タイプ・マッピング (続き)

DB2 データ・タイプ	OLE DB データ・タイプ標識	OLE DB 標準タイプ名	DB2 固有の名前
TIMESTAMP	DBTYPE_DBTIMESTAMP	"DBTYPE_DBTIMESTAMP"	"TIMESTAMP"
CHAR	DBTYPE_STR	"DBTYPE_CHAR"	"CHAR" または "CHARACTER"
VARCHAR	DBTYPE_STR	"DBTYPE_VARCHAR"	"VARCHAR"
LONG VARCHAR	DBTYPE_STR	"DBTYPE_LONGVARCHAR"	"LONG VARCHAR"
CLOB	DBTYPE_STR および DBCOLUMNFLAGS_ISLONG または DBPARAMFLAGS_ISLONG	"DBTYPE_CHAR" "DBTYPE_VARCHAR" "DBTYPE_LONGVARCHAR" および DBCOLUMNFLAGS_ISLONG または DBPARAMFLAGS_ISLONG	"CLOB"
GRAPHIC	DBTYPE_WSTR	"DBTYPE_WCHAR"	"GRAPHIC"
VARGRAPHIC	DBTYPE_WSTR	"DBTYPE_WVARCHAR"	"VARGRAPHIC"
LONG VARGRAPHIC	DBTYPE_WSTR	"DBTYPE_WLONGVARCHAR"	"LONG VARGRAPHIC"
DBCLOB	DBTYPE_WSTR および DBCOLUMNFLAGS_ISLONG または DBPARAMFLAGS_ISLONG	"DBTYPE_WCHAR" "DBTYPE_WVARCHAR" "DBTYPE_WLONGVARCHAR" および DBCOLUMNFLAGS_ISLONG または DBPARAMFLAGS_ISLONG	"DBCLOB"
CHAR(n) FOR BIT DATA	DBTYPE_BYTES	"DBTYPE_BINARY"	
VARCHAR(n) FOR BIT DATA	DBTYPE_BYTES	"DBTYPE_VARBINARY"	
LONG VARCHAR FOR BIT DATA	DBTYPE_BYTES	"DBTYPE_LONGVARBINARY"	
BLOB	DBTYPE_BYTES および DBCOLUMNFLAGS_ISLONG または DBPARAMFLAGS_ISLONG	"DBTYPE_BINARY" "DBTYPE_VARBINARY" "DBTYPE_LONGVARBINARY" および DBCOLUMNFLAGS_ISLONG または DBPARAMFLAGS_ISLONG	"BLOB"

OLE DB タイプから DB2 タイプにデータを設定するためのデータ変換

IBM OLE DB Provider for DB2 では、OLE DB タイプから DB2 タイプにデータを設定するためのデータ変換がサポートされています。

OLE DB タイプから DB2 タイプへのサポートされているデータ変換

以下の表に、OLE DB タイプから DB2 タイプへのデータ変換を示します。データのタイプや値に応じて、一部のケースではデータの切り捨てが起こる場合があることに注意してください。

表 10. OLE DB タイプから DB2 タイプへのデータ変換

OLE DB タイプ	DB2 データ・タイプ															
	SMALLINT	INTEGER	BIGINT	REAL	FLOAT	DECIMAL	DATE	TIME	TIMESTAMP	VARCHAR	VARIABLE CHAR	LONG VARCHAR	LONG VARRR	ビット・データの場合		DATA LINK
														CHAR	VARCHAR	
DBTYPE_EMPTY																
DBTYPE_NULL																
DBTYPE_RESERVED																
DBTYPE_I1	X	X	X	X	X	X			X	X						
DBTYPE_I2	X	X	X	X	X	X			X	X						
DBTYPE_I4	X	X	X	X	X	X			X	X						
DBTYPE_I8	X	X	X	X	X	X			X	X						
DBTYPE_UI1	X	X	X	X	X	X			X	X						
DBTYPE_UI2	X	X	X	X	X	X			X	X						
DBTYPE_UI4	X	X	X	X	X	X			X	X						
DBTYPE_UI8	X	X	X	X	X	X			X	X						
DBTYPE_R4	X	X	X	X	X	X			X	X						
DBTYPE_R8	X	X	X	X	X	X			X	X						

表 10. OLE DB タイプから DB2 タイプへのデータ変換 (続き)

OLE DB タイプ標識	DB2 データ・タイプ																			
	S M A L L I N T	I N T E G E R	B I T	R E A L	D U M M	D E C I M A L	D A T E	T I M E	C H A R	V A R C H	L O N G	C L O B	G R A P H I C	V A R R A Y	L O N G	ビット・データの 場合			D A T A	
																D B C H A R	V A R C H	L O N G		
DBTYPE_CY																				
DBTYPE_DECIMAL	X	X	X	X	X	X			X	X										
DBTYPE_NUMERIC	X	X	X	X	X	X			X	X										
DBTYPE_DATE																				
DBTYPE_BOOL	X	X	X	X	X	X			X	X										
DBTYPE_BYTES			X			X			X	X	X				X		X	X	X	
DBTYPE_BSTR - 未定																				
DBTYPE_STR	X	X	X	X	X	X	X	X	X	X	X		X	X	X		X	X	X	X
DBTYPE_WSTR													X	X	X					
DBTYPE_VARIANT - 未定																				
DBTYPE_IDISPATCH																				
DBTYPE_IUNKNOWN									X	X	X	X	X	X	X	X	X	X	X	
DBTYPE_GUID																				
DBTYPE_ERROR																				
DBTYPE_BYREF																				

表 10. OLE DB タイプから DB2 タイプへのデータ変換 (続き)

OLE DB タイプ	DB2 データ・タイプ																ビット・データの 場合		D A T A L I N K	
	S M A L L I N T	I N T E G E R	B I N A R Y	R E A L	D U M P	D E C I M A L	T I M E S T A M P	C H A P T E R	V A R C H A R	V A R C H A R L O N G	C O L L E C T I O N	G R A M M A T I C A L	V A R I A N T	V A R I A N T	D I S T I N C T	C H A R	V A R C H A R	L O N G		
																				C H A R
OLE DB タイプ標識																				
DBTYPE_ARRAY																				
DBTYPE_VECTOR																				
DBTYPE_UDT																				
DBTYPE_DBDATE						X	X	X	X											
DBTYPE_DBTIME							X	X	X	X										
DBTYPE_DBTIMESTAMP						X	X	X	X	X										
DBTYPE_FILETIME																				
DBTYPE_PROP_VARIANT																				
DBTYPE_HCHAPTER																				
DBTYPE_VARNUMERIC																				

DB2 タイプから OLE DB タイプにデータを設定するためのデータ変換

データを取得するため、IBM OLE DB Provider では DB2 タイプから OLE DB タイプへのデータ変換を行うことができます。

DB2 タイプから OLE DB タイプへのサポートされているデータ変換

以下の表は、DB2 タイプから OLE DB タイプへのサポートされるデータ変換を示します。データのタイプや値に応じて、一部のケースではデータの切り捨てが起こる場合があることに注意してください。

表 11. DB2 タイプから OLE DB タイプへのデータ変換

OLE DB タイプ標識	DB2 データ・タイプ																					
	S M A L L I N T	I N T E G E R	B I T	R E A L	F L O A T I N G	D E C I M A L N U M E R I C	D A T E	T I M E	T I M E S T A M P	C H A R	V A R C H A R	L O N G V A R C H A R	C L O B	G R A P H I C	V A R R A Y C H A R	L O N G V A R R A Y C H A R	ビット・データの 場合			D A T A B L O B		
																	C H A R	V A R C H A R	L O N G V A R C H A R			
DBTYPE_EMPTY																						
DBTYPE_NULL																						
DBTYPE_RESERVED																						
DBTYPE_I1	X	X		X	X	X				X	X	X		X	X	X		X	X	X		X
DBTYPE_I2	X	X		X	X	X				X	X	X		X	X	X		X	X	X		X
DBTYPE_I4	X	X		X	X	X				X	X	X		X	X	X		X	X	X		X
DBTYPE_I8	X	X	X	X	X	X				X	X	X		X	X	X		X	X	X		X
DBTYPE_UI1	X	X		X	X	X				X	X	X		X	X	X		X	X	X		X
DBTYPE_UI2	X	X		X	X	X				X	X	X		X	X	X		X	X	X		X
DBTYPE_UI4	X	X		X	X	X				X	X	X		X	X	X		X	X	X		X
DBTYPE_UI8	X	X	X	X	X	X				X	X	X		X	X	X		X	X	X		X
DBTYPE_R4	X	X		X	X	X				X	X	X		X	X	X		X	X	X		X
DBTYPE_R8	X	X		X	X	X				X	X	X		X	X	X		X	X	X		X
DBTYPE_CY	X	X		X	X	X				X	X	X		X	X	X		X	X	X		X
DBTYPE_DECIMAL	X	X		X	X	X				X	X	X		X	X	X		X	X	X		X
DBTYPE_NUMERIC	X	X		X	X	X				X	X	X		X	X	X		X	X	X		X

表 11. DB2 タイプから OLE DB タイプへのデータ変換 (続き)

OLE DB タイプ標識	DB2 データ・タイプ																				
	S M A L L I N T	I N T E G E R	B I T	R E A L	D U M M	D E C I M A L	D A T E	T I M E	C H A R	V A R C H	L O N G	C L O B	G R A P H I C	V A R R A Y	V A R R A Y	D B C L O B	ビット・データの 場合			D A T A	
																	C H A R	V A R C H	L O N G		
DBTYPE_DATE	X	X		X	X		X	X	X	X	X		X	X	X					X	
DBTYPE_BOOL	X	X		X	X	X				X	X	X		X	X	X		X	X	X	X
DBTYPE_BYTES	X	X		X	X	X	X	X	X	X	X		X	X	X		X	X	X		X
DBTYPE_BSTR	X	X	X	X	X	X	X	X	X	X	X		X	X	X		X	X	X		X
DBTYPE_STR	X	X	X	X	X	X	X	X	X	X	X		X	X	X		X	X	X		X
DBTYPE_WSTR	X	X	X	X	X	X	X	X	X	X	X		X	X	X		X	X	X		X
DBTYPE_VARIANT	X	X	X	X	X	X	X	X	X	X	X		X	X	X		X	X	X		X
DBTYPE_IDISPATCH																					
DBTYPE_IUNKNOWN	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
DBTYPE_GUID									X	X	X		X	X	X		X	X	X		X
DBTYPE_ERROR																					
DBTYPE_BYREF																					
DBTYPE_ARRAY																					
DBTYPE_VECTOR																					
DBTYPE_UDT																					
DBTYPE_DBDATE							X	X	X	X	X		X	X	X		X	X	X		X

表 11. DB2 タイプから OLE DB タイプへのデータ変換 (続き)

OLE DB タイプ標識	DB2 データ・タイプ																			
	S M A L L I N T	I N T E G E R	B I N A R Y	R E A L	D U M M	D E C I M A L	T I M E S T A M P	C H A R	V A R C H A R	L O N G V A R C H A R	C L O B	G R A M M A T I C A L	V A R I A N T	D B C L O B	ビット・データの 場合			D A T A B L O C K		
															C H A R	V A R C H A R	L O N G			
DBTYPE_DBTIME						X	X	X	X	X		X	X	X						X
DBTYPE_DBTIMESTAMP						X	X	X	X	X		X	X	X		X	X	X		X
DBTYPE_FILETIME			X			X	X	X	X	X		X	X	X		X	X	X		X
DBTYPE_PROP_VARIANT	X	X	X	X	X				X	X	X		X	X	X		X	X	X	X
DBTYPE_HCHAPTER																				
DBTYPE_VARNUMERIC																				

注: アプリケーションが ISequentialStream::Read を実行してストレージ・オブジェクトからデータを取得するとき、戻されるデータのフォーマットは列のデータ・タイプによって決まります。

- 文字でないデータおよびバイナリー・データ・タイプの場合、列のデータはそのオペレーティング・システムでその値を表すバイトのシーケンスとして表されます。
- 文字データ・タイプの場合、データは最初に DBTYPE_STR に変換されます。
- DBCLOB の場合、データは最初に DBTYPE_WCHAR に変換されます。

IBM OLE DB Provider の制約事項

IBM OLE DB Provider には制限があります。つまり、IBMDADB2 は自動コミットをサポートしますがユーザー制御のトランザクション・スコープ RestartPosition はサポートされず、IBMDADB2 は DBID パラメーターを介して渡される表名を引用符で囲みません。

IBM OLE DB Provider の制限は、以下のとおりです。

- IBMDADB2 では、ITransactionLocal インターフェースにより、自動コミットおよびユーザー制御のトランザクション・スコープがサポートされています。デフォルトのスコープは自動コミット・トランザクション・スコープです。ネストされたトランザクションはサポートされていません。

- RestartPosition は、コマンド・テキストにパラメーターが含まれている場合にサポートされません。
- IBMDADB2 では、DBID パラメーター (IOpenRowset インターフェースで使用されるパラメーター) を介して渡される表名が引用符で囲まれません。その代わりに、引用符が必要な場合に OLE DB Consumer で表名に引用符を追加する必要があります。

IBM OLE DB Provider での OLE DB コンポーネントおよびインターフェースのサポート

以下の表は、IBM OLE DB Provider for DB2 および Microsoft OLE DB Provider for ODBC でサポートされている OLE DB コンポーネントおよびインターフェースを示しています。

以下の表は、IBM OLE DB Provider for DB2 および Microsoft OLE DB Provider for ODBC でサポートされている OLE DB コンポーネントおよびインターフェースを示しています。

表 12. BLOB

インターフェース	DB2	ODBC Provider
ISequentialStream	はい	はい

表 13. コマンド

インターフェース	DB2	ODBC Provider
IAccessor	はい	はい
ICommand	はい	はい
ICommandPersist	いいえ	いいえ
ICommandPrepare	はい	はい
ICommandProperties	はい	はい
ICommandText	はい	はい
ICommandWithParameters	はい	はい
IColumnsInfo	はい	はい
IColumnsRowset	はい	はい
IConvertType	はい	はい
ISupportErrorInfo	はい	はい

表 14. データ・ソース

インターフェース	DB2	ODBC Provider
IConnectionPoint	いいえ	はい
IDBAsynchNotify (Consumer)	いいえ	いいえ
IDBAsynchStatus	いいえ	いいえ
IDBConnectionPointContainer	いいえ	はい
IDBCreateSession	はい	はい
IDBDataSourceAdmin	いいえ	いいえ

表 14. データ・ソース (続き)

インターフェース	DB2	ODBC Provider
IDBInfo	はい	はい
IDBInitialize	はい	はい
IDBProperties	はい	はい
IPersist	はい	いいえ
IPersistFile	はい	はい
ISupportErrorInfo	はい	はい

表 15. 列挙子

インターフェース	DB2	ODBC Provider
IDBInitialize	はい	はい
IDBProperties	はい	はい
IParseDisplayName	はい	いいえ
ISourcesRowset	はい	はい
ISupportErrorInfo	はい	はい

表 16. エラー検索サービス

インターフェース	DB2	ODBC Provider
IErrorLookUp	はい	はい

表 17. エラー・オブジェクト

インターフェース	DB2	ODBC Provider
IErrorInfo	はい	いいえ
ISQLErrorInfo (カスタム)	はい	いいえ

表 18. 複数結果

インターフェース	DB2	ODBC Provider
IMultipleResults	はい	はい
ISupportErrorInfo	はい	はい

表 19. 行セット

インターフェース	DB2	ODBC Provider
IAccessor	はい	はい
IColumnsRowset	はい	はい
IColumnsInfo	はい	はい
IConvertType	はい	はい
IChapteredRowset	いいえ	いいえ
IConnectionPointContainer	はい	はい
IDBAsynchStatus	いいえ	いいえ
IParentRowset	いいえ	いいえ
IRowset	はい	はい

表 19. 行セット (続き)

インターフェース	DB2	ODBC Provider
IRowsetChange	はい	はい
IRowsetChapterMember	いいえ	いいえ
IRowsetFind	いいえ	いいえ
IRowsetIdentity	はい	はい
IRowsetIndex	いいえ	いいえ
IRowsetInfo	はい	はい
IRowsetLocate	はい	はい
IRowsetNotify (Consumer)	はい	いいえ
IRowsetRefresh	Cursor Service Component	はい
IRowsetResynch	Cursor Service Component	はい
IRowsetScroll	はい ¹	はい
IRowsetUpdate	Cursor Service Component	はい
IRowsetView	いいえ	いいえ
ISupportErrorInfo	はい	はい
注:		
1. 戻される値は近似値です。削除された行はスキップされません。		

表 20. セッション

インターフェース	DB2	ODBC Provider
IAlterIndex	いいえ	いいえ
IAlterTable	いいえ	いいえ
IDBCreateCommand	はい	はい
IDBSchemaRowset	はい	はい
IGetDataSource	はい	はい
IIndexDefinition	いいえ	いいえ
IOpenRowset	はい	はい
ISessionProperties	はい	はい
ISupportErrorInfo	はい	はい
ITableDefinition	いいえ	いいえ
ITableDefinitionWithConstraints	いいえ	いいえ
ITransaction	はい	はい
ITransactionJoin	はい	はい
ITransactionLocal	はい	はい
ITransactionObject	いいえ	いいえ
ITransactionOptions	いいえ	はい

表 21. ビュー・オブジェクト

インターフェース	DB2	ODBC Provider
IViewChapter	いいえ	いいえ
IViewFilter	いいえ	いいえ

表 21. ビュー・オブジェクト (続き)

インターフェース	DB2	ODBC Provider
IViewRowset	いいえ	いいえ
IViewSort	いいえ	いいえ

IBM OLE DB Provider での OLE DB プロパティのサポート

以下の表は、IBM OLE DB Provider for DB2 でサポートされている OLE DB プロパティとデフォルト値を示しています。

以下の表は、IBM OLE DB Provider for DB2 でサポートされている OLE DB プロパティを示しています。

表 22. IBM OLE DB Provider for DB2 でサポートされているプロパティ: データ・ソース (DBPROPSET_DATASOURCE)

プロパティ	デフォルト値	R/W
DBPROP_MULTIPLECONNECTIONS	VARIANT_FALSE	R
DBPROP_RESETDATASOURCE	DBPROPVAL_RD_RESETALL	R/W

表 23. IBM OLE DB Provider for DB2 でサポートされているプロパティ: DB2 データ・ソース (DBPROPSET_DB2DATASOURCE)

プロパティ	デフォルト値	R/W
DB2PROP_REPORTISLONGFORLONGTYPES	VARIANT_FALSE	R/W
DB2PROP_RETURNCHARASWCHAR	VARIANT_TRUE	R/W
DB2PROP_SORTBYORDINAL	VARIANT_FALSE	R/W

表 24. IBM OLE DB Provider for DB2 でサポートされているプロパティ: データ・ソース情報 (DBPROPSET_DATASOURCEINFO)

プロパティ	デフォルト値	R/W
DBPROP_ACTIVESESSIONS	0	R
DBPROP_ASYNC_TXN_ABORT	VARIANT_FALSE	R
DBPROP_ASYNC_TXN_COMMIT	VARIANT_FALSE	R
DBPROP_BYREFACCESSORS	VARIANT_FALSE	R
DBPROP_COLUMNDEFINITION	DBPROPVAL_CD_NOTNULL	R
DBPROP_CONCATNULLBEHAVIOR	DBPROPVAL_CB_NULL	R
DBPROP_CONNECTIONSTATUS	DBPROPVAL_CS_INITIALIZED	R
DBPROP_DATASOURCENAME	なし	R
DBPROP_DATASOURCE_READONLY	VARIANT_FALSE	R
DBPROP_DBMSNAME	なし	R
DBPROP_DBMSVER	なし	R
DBPROP_DSOTHRADMODEL	DBPROPVAL_RT_FREETHREAD	R
DBPROP_GROUPBY	DBPROPVAL_GB_CONTAINS_SELECT	R
DBPROP_IDENTIFIER_CASE	DBPROPVAL_IC_UPPER	R

表 24. IBM OLE DB Provider for DB2 でサポートされているプロパティ: データ・ソース情報
(DBPROPSET_DATASOURCEINFO) (続き)

プロパティ	デフォルト値	R/W
DBPROP_MAXINDEXSIZE	0	R
DBPROP_MAXROWSIZE	0	R
DBPROP_MAXROWSIZEINCLUDESBLOB	VARIANT_TRUE	R
DBPROP_MAXTABLEINSELECT	0	R
DBPROP_MULTIPLEPARAMSETS	VARIANT_FALSE	R
DBPROP_MULTIPLERESULTS	DBPROPVAL_MR_SUPPORTED	R
DBPROP_MULTIPLESTORAGEOBJECTS	VARIANT_TRUE	R
DBPROP_MULTITABLEUPDATE	VARIANT_FALSE	R
DBPROP_NULLCOLLATION	DBPROPVAL_NC_LOW	R
DBPROP_OLEOBJECTS	DBPROPVAL_OO_BLOB	R
DBPROP_ORDERBYCOLUMNSINSELECT	VARIANT_FALSE	R
DBPROP_OUTPUTPARAMETERAVAILABILITY	DBPROPVAL_OA_ATEXECUTE	R
DBPROP_PERSISTENTIDTYPE	DBPROPVAL_PT_NAME	R
DBPROP_PREPAREABORTBEHAVIOR	DBPROPVAL_CB_DELETE	R
DBPROP_PROCEDURETERM	"STORED PROCEDURE"	R
DBPROP_PROVIDERFRIENDLYNAME	"IBM OLE DB Provider for DB2"	R
DBPROP_PROVIDERNAME	"IBMDADB2.DLL"	R
DBPROP_PROVIDEROLEDBVER	"02.7"	R
DBPROP_PROVIDERVER	なし	R
DBPROP_QUOTEIDENTIFIERCASE	DBPROPVAL_IC_SENSITIVE	R
DBPROP_ROWSETCONVERSIONSONCOMMAND	VARIANT_TRUE	R
DBPROP_SCHEMATERM	"SCHEMA"	R
DBPROP_SCHEMAUSAGE	DBPROPVAL_SU_DML_STATEMENTS DBPROPVAL_SU_TABLE_DEFINITION DBPROPVAL_SU_INDEX_DEFINITION DBPROPVAL_SU_PRIVILEGE_DEFINITION	R
DBPROP_SQLSUPPORT	DBPROPVAL_SQL_ODBC_EXTENDED DBPROPVAL_SQL_ESCAPECLAUSES DBPROPVAL_SQL_ANSI92_ENTRY	R
DBPROP_SERVERNAME	なし	R
DBPROP_STRUCTUREDSTORAGE	DBPROPVAL_SS_ISEQUENTIALSTREAM	R
DBPROP_SUBQUERIES	DBPROPVAL_SQ_CORRELATEDSUBQUERIES DBPROPVAL_SQ_COMPARISON DBPROPVAL_SQ_EXISTS DBPROPVAL_SQ_IN DBPROPVAL_SQ_QUANTIFIED	R

表 24. IBM OLE DB Provider for DB2 でサポートされているプロパティ: データ・ソース情報 (DBPROPSET_DATASOURCEINFO) (続き)

プロパティ	デフォルト値	R/W
DBPROP_SUPPORTEDTXNDDL	DBPROPVAL_TC_ALL	R
DBPROP_SUPPORTEDTXNISOLEVELS	DBPROPVAL_TI_CURSORSTABILITY DBPROPVAL_TI_READCOMMITTED DBPROPVAL_TI_READUNCOMMITTED DBPROPVAL_TI_SERIALIZABLE	R
DBPROP_SUPPORTEDTXNISORETAIN	DBPROPVAL_TR_COMMIT_DC DBPROPVAL_TR_ABORT_NO	R
DBPROP_TABLETERM	"TABLE"	R
DBPROP_USERNAME	なし	R

表 25. IBM OLE DB Provider for DB2 でサポートされているプロパティ: 初期化 (DBPROPSET_DBINIT)

プロパティ	デフォルト値	R/W
DBPROP_AUTH_PASSWORD	なし	R/W
DBPROP_INIT_TIMEOUT (1)	0	R/W
DBPROP_AUTH_PERSIST _SENSITIVE_AUTHINFO	VARIANT_FALSE	R/W
DBPROP_AUTH_USERID	なし	R/W
DBPROP_INIT_DATASOURCE	なし	R/W
DBPROP_INIT_HWND	なし	R/W
DBPROP_INIT_MODE	DB_MODE_READWRITE	R/W
DBPROP_INIT_OLEDBSERVICES	0xFFFFFFFF	R/W
DBPROP_INIT_PROMPT	DBPROMPT_NOPROMPT	R/W
DBPROP_INIT_PROVIDERSTRING	なし	R/W

表 26. IBM OLE DB Provider for DB2 でサポートされているプロパティ: 行セット (DBPROPSET_ROWSET)

プロパティ	デフォルト値	R/W
DBPROP_ABORTPRESERVE	VARIANT_FALSE	R
DBPROP_ACCESSORDER	DBPROPVAL_AO_RANDOM	R
DBPROP_BLOCKINGSTORAGEOBJECTS	VARIANT_FALSE	R
DBPROP_BOOKMARKS	VARIANT_FALSE	R/W
DBPROP_BOOKMARKSKIPPED	VARIANT_FALSE	R
DBPROP_BOOKMARKTYPE	DBPROPVAL_BMK_NUMERIC	R
DBPROP_CACHEDEFERRED	VARIANT_FALSE	R/W
DBPROP_CANFETCHBACKWARDS	VARIANT_FALSE	R/W
DBPROP_CANHOLDROWS	VARIANT_FALSE	R
DBPROP_CANSROLLBACKWARDS	VARIANT_FALSE	R/W
DBPROP_CHANGEINSERTEDROWS	VARIANT_FALSE	R
DBPROP_COMMITPRESERVE	VARIANT_TRUE	R/W

表 26. IBM OLE DB Provider for DB2 でサポートされているプロパティ: 行セット (DBPROPSET_ROWSET) (続き)

プロパティ	デフォルト値	R/W
DBPROP_COMMANDTIMEOUT	0	R/W
DBPROP_DEFERRED	VARIANT_FALSE	R
DBPROP_IAccessor	VARIANT_TRUE	R
DBPROP_IColumnsInfo	VARIANT_TRUE	R
DBPROP_IColumnsRowset	VARIANT_TRUE	R/W
DBPROP_IConvertType	VARIANT_TRUE	R
DBPROP_IMultipleResults	VARIANT_FALSE	R/W
DBPROP_IRowset	VARIANT_TRUE	R
DBPROP_IRowChange	VARIANT_FALSE	R/W
DBPROP_IRowsetFind	VARIANT_FALSE	R
DBPROP_IRowsetIdentity	VARIANT_TRUE	R
DBPROP_IRowsetInfo	VARIANT_TRUE	R
DBPROP_IRowsetLocate	VARIANT_FALSE	R/W
DBPROP_IRowsetScroll	VARIANT_FALSE	R/W
DBPROP_IRowsetUpdate	VARIANT_FALSE	R
DBPROP_ISequentialStream	VARIANT_TRUE	R
DBPROP_ISupportErrorInfo	VARIANT_TRUE	R
DBPROP_LITERALBOOKMARKS	VARIANT_FALSE	R
DBPROP_LITERALIDENTITY	VARIANT_TRUE	R
DBPROP_LOCKMODE	DBPROPVAL_LM_SINGLEROW	R/W
DBPROP_MAXOPENROWS	32767	R
DBPROP_MAXROWS	0	R/W
DBPROP_NOTIFICATIONGRANULARITY	DBPROPVAL_NT_SINGLEROW	R/W
DBPROP_NOTIFICATION PHASES	DBPROPVAL_NP_OKTODO DBPROPVAL_NP_ABOUTTOD DBPROPVAL_NP_SYNCHAFTER DBPROPVAL_NP_FAILEDTOD DBPROPVAL_NP_DIDEVENT	R
DBPROP_NOTIFYROWSETRELEASE	DBPROPVAL_NP_OKTODO DBPROPVAL_NP_ABOUTTOD	R
DBPROP_NOTIFYROWSETFETCHPOSITIONCHANGE	DBPROPVAL_NP_OKTODO DBPROPVAL_NP_ABOUTTOD	R
DBPROP_NOTIFYCOLUMNSET	DBPROPVAL_NP_OKTODO DBPROPVAL_NP_ABOUTTOD	R
DBPROP_NOTIFYROWDELETE	DBPROPVAL_NP_OKTODO DBPROPVAL_NP_ABOUTTOD	R
DBPROP_NOTIFYROWINSERT	DBPROPVAL_NP_OKTODO DBPROPVAL_NP_ABOUTTOD	R

表 26. IBM OLE DB Provider for DB2 でサポートされているプロパティ: 行セット (DBPROPSET_ROWSET) (続き)

プロパティ	デフォルト値	R/W
DBPROP_ORDEREDBOOKMARKS	VARIANT_FALSE	R
DBPROP_OTHERINSERT	VARIANT_FALSE	R
DBPROP_OTHERUPDELETEDELETE	VARIANT_FALSE	R/W
DBPROP_OWNINGINSERT	VARIANT_FALSE	R
DBPROP_OWNINGUPDELETEDELETE	VARIANT_FALSE	R
DBPROP_QUICKRESTART	VARIANT_FALSE	R/W
DBPROP_REMOVEDELETED	VARIANT_FALSE	R/W
DBPROP_ROWTHREADMODEL	DBPROPVAL_RT_FREETHREAD	R
DBPROP_SERVERCURSOR	VARIANT_TRUE	R
DBPROP_SERVERDATAONINSERT	VARIANT_FALSE	R
DBPROP_UNIQUEROWS	VARIANT_FALSE	R/W
DBPROP_UPDATABILITY	0	R/W

表 27. IBM OLE DB Provider for DB2 でサポートされているプロパティ: DB2 行セット (DBPROPSET_DB2ROWSET)

プロパティ	デフォルト値	R/W
DBPROP_ISLONGMINLENGTH	32000	R/W

表 28. IBM OLE DB Provider for DB2 でサポートされているプロパティ: セッション (DBPROPSET_SESSION)

プロパティ	デフォルト値	R/W
DBPROP_SESS_AUTOCOMMITISOLEVELS	DBPROPVAL_TI_CURSORSTABILITY	R/W

注:

1. TCP/IP プロトコルを使用してサーバーに接続する場合にのみ、タイムアウトを適用できます。タイムアウトは TCP/IP sock 接続時にのみ強制されます。指定されたタイムアウトの有効期限が切れる前に sock 接続が完了する場合、初期化プロセスの残りの部分に対しては、タイムアウトが強制されなくなります。クライアント・リルート機能が使用される場合、タイムアウトは二倍になります。一般に、クライアント・リルートが有効になっている場合、接続タイムアウト動作はクライアント・リルートによって指示されます。

IBM OLE DB Provider によるデータ・ソースへの接続

以下の例は、IBM OLE DB Provider for DB2 を使用して DB2 データ・ソースに接続する方法を示しています。

以下の例は、IBM OLE DB Provider for DB2 を使用して DB2 データ・ソースに接続する方法を示しています。

例 1: ADO を使用する Visual Basic アプリケーション

```
Dim db As ADODB.Connection
Set db = New ADODB.Connection
db.Provider = "IBMDADB2"
db.CursorLocation = adUseClient
...
```

例 2 IDataInitialize と Service Component を使用する C/C++ アプリケーション

```
hr = CoCreateInstance (
    CLSID_MSDAINITIALIZE,
    NULL,
    CLSCTX_INPROC_SERVER,
    IID_IDataInitialize,
    (void**)&pIDataInitialize);

hr = pIDataInitialize->CreateDBInstance(
    CLSID_IBMDADB2, // ClassID of IBMDADB2
    NULL,
    CLSCTX_INPROC_SERVER,
    NULL,
    IID_IDBInitialize,
    (IUnknown**)&pIDBInitialize);
```

ADO アプリケーション

ADO 接続ストリング・キーワード

ADO 接続ストリング・キーワードを使用するには、キーワードの値を `keyword=value` で指定する必要があります。キーと値のペアごとに、セミコロンで区切ります。表 1 では、キーワード、値、および意味について詳しく説明しています。

ADO (ActiveX Data Objects) 接続ストリング・キーワードを指定するには、Provider (接続) ストリングで `keyword=value` のフォーマットを使用してキーワードを指定します。複数のキーワードを指定する場合はセミコロン (;) で区切ります。

以下の表は、IBM OLE DB Provider for DB2 でサポートされているキーワードを示しています。

表 29. IBM OLE DB Provider for DB2 でサポートされているキーワード

キーワード	値	意味
DSN	データベース別名の名前	データベース・ディレクトリーで使用される DB2 データベース別名。
UID	ユーザー ID	DB2 サーバーへの接続に使用するユーザー ID。
PWD	UID のパスワード	DB2 サーバーへの接続に使用するユーザー ID のパスワード。

他の CLI 構成キーワードも、IBM OLE DB Provider の動作に影響します。

Visual Basic ADO アプリケーションによるデータ・ソースへの接続

IBM OLE DB Provider for DB2 を使用するには、IBMDADB2 プロバイダーの名前を指定する必要があります。

IBM OLE DB Provider for DB2 を使用して DB2 データ・ソースに接続するには、IBMDADB2 Provider 名を指定します。

ADO アプリケーションにおける更新可能な両方向スクロール・カーソル

更新可能な両方向スクロール・カーソルにアクセスする ADO アプリケーションはカーソル位置を `adUseClient` または `adUseServer` に設定する必要があります。カーソル位置を `adUseServer` に設定すると、カーソルがサーバー上で実体化されません。

IBM OLE DB Provider for DB2 では、読み取り専用カーソル、前方スクロール・カーソル、読み取り専用の両方向スクロール・カーソル、更新可能な両方向スクロール・カーソルがネイティブでサポートされています。更新可能な両方向スクロール・カーソルにアクセスする ADO アプリケーションはカーソル位置を `adUseClient` または `adUseServer` に設定する必要があります。カーソル位置を `adUseServer` に設定すると、カーソルがサーバー上で実体化されます。

ADO アプリケーションの制限

ADO アプリケーションには、ストアード・プロシージャの呼び出し、およびサーバー・サイドで両方向スクロール・カーソルを使用した新規行の挿入に制限があり、デフォルトのパラメーター値はサポートされていません。

ADO アプリケーションの制限は以下のとおりです。

- ストアード・プロシージャを呼び出す ADO アプリケーションでは、事前にパラメーターが作成されて明示的にバインドされていなければなりません。パラメーターを自動生成するための `Parameters.Refresh` 方式は、DB2 Server for VSE & VM ではサポートされていません。
- デフォルトのパラメーター値はサポートされていません。
- サーバー・サイドの両方向スクロール・カーソルを使用して新しい行を挿入する場合、`AddNew()` 方式を `Fieldlist` および `Values` 引数値を指定して使用します。これは、各列に対して `Update()` 呼び出しに続いて引数なしの `AddNew()` を呼び出すより効率的です。`AddNew()` および `Update()` の各呼び出しはサーバーに対する別個の要求です。したがって、これは `AddNew()` を単独で呼び出すより効率が悪くなります。
- 新規に挿入された行は、サーバー・サイドの両方向スクロール・カーソルでは更新できません。
- サーバー・サイドの両方向スクロール・カーソルを使用している場合には、LONG データ、または LOB データを持つ表は更新できません。

IBM OLE DB Provider での ADO メソッドおよびプロパティのサポート

IBM OLE DB Provider は、以下の ADO メソッドおよびプロパティをサポートしています。

IBM OLE DB Provider は、以下の ADO メソッドおよびプロパティをサポートしています。

表 30. コマンド・メソッド

メソッド/プロパティ	OLE DB インターフェース/プロパティ	IBM OLE DB サポート
Cancel	ICommand	はい
CreateParameter		はい
Execute		はい

表 31. コマンド・プロパティ

メソッド/プロパティ	OLE DB インターフェース/プロパティ	IBM OLE DB サポート
ActiveConnection	(ADO 固有)	
Command Text	ICommandText	はい
Command Timeout	ICommandProperties::SetProperties DBPROP_COMMANDTIMEOUT	はい
CommandType	(ADO 固有)	
Prepared	ICommandPrepare	はい
State	(ADO 固有)	

表 32. コマンド・コレクション

メソッド/プロパティ	OLE DB インターフェース/プロパティ	IBM OLE DB サポート
Parameters	ICommandWithParameter DBSCHEMA _PROCEDURE_PARAMETERS	はい
Properties	ICommandProperties IDBProperties	はい

表 33. 接続メソッド

メソッド/プロパティ	OLE DB インターフェース/プロパティ	IBM OLE DB サポート
BeginTrans CommitTrans RollbackTrans	ITransactionLocal	はい (ただし、非ネスト) はい (ただし、非ネスト) はい (ただし、非ネスト)

表 33. 接続メソッド (続き)

メソッド/プロパティ	OLE DB インターフェース/プロパティ	IBM OLE DB サポート
Execute	ICommand IOpenRowset	はい
Open	IDBCreateSession IDBInitialize	はい
OpenSchema adSchemaColumnPrivileges adSchemaColumns adSchemaForeignKeys adSchemaIndexes adSchemaPrimaryKeys adSchemaProcedureParam adSchemaProcedures adSchemaProviderType adSchemaStatistics adSchemaTablePrivileges adSchemaTables	IDBSchemaRowset	はい はい はい はい はい はい はい はい はい はい はい
Cancel		はい

表 34. 接続プロパティ

メソッド/プロパティ	OLE DB インターフェース/プロパティ	IBM OLE DB サポート
Attributes adXactCommitRetaining adXactRollbackRetaining	ITransactionLocal	はい はい
CommandTimeout	ICommandProperties DBPROP_COMMAND_TIMEOUT	はい
ConnectionString	(ADO 固有)	
ConnectionTimeout	IDBProperties DBPROP_INIT_TIMEOUT	いいえ
CursorLocation: adUseClient adUseNone adUseServer	(OLE DB Cursor Service を使用) (使用されず)	はい いいえ はい
DefaultDataBase	IDBProperties DBPROP_CURRENTCATALOG	いいえ
IsolationLevel	ITransactionLocal DBPROP_SESS _AUTOCOMMITISOLEVELS	はい

表 34. 接続プロパティ (続き)

メソッド/プロパティ	OLE DB インターフェース/プロパティ	IBM OLE DB サポート
Mode adModeRead adModeReadWrite adModeShareDenyNone adModeShareDenyRead adModeShareDenyWrite adModeShareExclusive adModeUnknown adModeWrite	IDBProperties DBPROP_INIT_MODE	いいえ はい いいえ いいえ いいえ いいえ いいえ いいえ
Provider	ISourceRowset::GetSourceRowset	はい
State	(ADO 固有)	
バージョン	(ADO 固有)	

表 35. 接続コレクション

メソッド/プロパティ	OLE DB インターフェース/プロパティ	IBM OLE DB サポート
Errors	IErrorRecords	はい
Properties	IDBProperties	はい

表 36. エラー・プロパティ

メソッド/プロパティ	OLE DB インターフェース/プロパティ	IBM OLE DB サポート
Description NativeError Number Source SQLState	IErrorRecords	はい はい はい はい はい
HelpContext HelpFile		いいえ いいえ

表 37. フィールド・メソッド

メソッド/プロパティ	OLE DB インターフェース/プロパティ	IBM OLE DB サポート
AppendChunk GetChunk	ISequentialStream	はい はい

表 38. フィールド・プロパティ

メソッド/プロパティ	OLE DB インターフェース/プロパティ	IBM OLE DB サポート
Actual Size	IAccessor IRowset	はい
Attributes DataFormat DefinedSize Name NumericScale Precision Type	IColumnInfo	はい はい はい はい はい はい
OriginalValue	IRowsetUpdate	はい (カーソル・サービス)
UnderlyingValue	IRowsetRefresh IRowsetResynch	はい (カーソル・サービス) はい (カーソル・サービス)
値	IAccessor IRowset	はい

表 39. フィールド・コレクション

メソッド/プロパティ	OLE DB インターフェース/プロパティ	IBM OLE DB サポート
Properties	IDBProperties IRowsetInfo	はい

表 40. パラメーター・メソッド

メソッド/プロパティ	OLE DB インターフェース/プロパティ	IBM OLE DB サポート
AppendChunk	ISequentialStream	はい
Attributes Direction Name NumericScale Precision Scale Size Type	ICommandWithParameter DBSCHEMA _PROCEDURE_PARAMETERS	はい いいえ はい はい はい はい はい
値	IAccessor ICommand	はい

表 41. パラメーター・コレクション

メソッド/プロパティ	OLE DB インターフェース/プロパティ	IBM OLE DB サポート
Properties		はい

表 42. レコード・セット・メソッド

メソッド/プロパティ	OLE DB インターフェース/プロパティ	IBM OLE DB サポート
AddNew	IRowsetChange	はい
Cancel		はい
CancelBatch	IRowsetUpdate::Undo	はい (カーソル・サービス)
CancelUpdate		はい (カーソル・サービス)
Clone	IRowsetLocate	はい
Close	IAccessor IRowset	はい
CompareBookmarks		いいえ
Delete	IRowsetChange	はい
GetRows	IAccessor IRowset	はい
Move	IRowset IRowsetLocate	はい
MoveFirst	IRowset IRowsetLocate	はい
MoveNext	IRowset IRowsetLocate	はい
MoveLast	IRowsetLocate	はい
MovePrevious	IRowsetLocate	はい
NextRecordSet	IMultipleResults	はい
Open	ICommand IOpenRowset	はい
Requery	ICommand IOpenRowset	はい
Resync	IRowsetRefresh	はい (カーソル・サービス)
Supports	IRowsetInfo	はい
Update UpdateBatch	IRowsetChange IRowsetUpdate	はい はい (カーソル・サービス)

表 43. レコード・セット・プロパティ

メソッド/プロパティ	OLE DB インターフェース/プロパティ	IBM OLE DB サポート
AbsolutePage	IRowsetLocate IRowsetScroll	はい はい ¹
AbsolutePosition	IRowsetLocate IRowsetScroll	はい はい ¹
ActiveConnection	IDBCreateSession IDBInitialize	はい
BOF	(ADO 固有)	
Bookmark	IAccessor IRowsetLocate	はい
CacheSize	IRowsetLocate 内の cRows IRowset	はい
CursorType adOpenDynamic adOpenForwardOnly adOpenKeySet adOpenStatic	ICommandProperties	いいえ はい はい はい
EditMode	IRowsetUpdate	はい (カーソル・サービス)
EOF	(ADO 固有)	
Filter	IRowsetLocate IRowsetView IRowsetUpdate IViewChapter IViewFilter	いいえ
LockType	ICommandProperties	はい
MarshallOption		いいえ
MaxRecords	ICommandProperties IOpenRowset	はい
PageCount	IRowsetScroll	はい ¹
PageSize	(ADO 固有)	
Sort	(ADO 固有)	
Source	(ADO 固有)	
State	(ADO 固有)	
Status	IRowsetUpdate	はい (カーソル・サービス)
注:		
1. 戻される値は近似値です。削除された行はスキップされません。		

表 44. レコード・セット・コレクション

メソッド/プロパティ	OLE DB インターフェース/プロパティ	IBM OLE DB サポート
Fields	IColumnInfo	はい
Properties	IDBProperties IRowsetInfo::GetProperties	はい

C/C++ アプリケーションのコンパイルおよびリンクと IBM OLE DB Provider

定数 CLSID_IBMDADB2 を使用する C/C++ アプリケーションには、`SQLLIB¥include` ディレクトリーにある `ibmdadb2.h` ファイルを組み込む必要があります。

定数 CLSID_IBMDADB2 を使用する C/C++ アプリケーションには、`SQLLIB¥include` ディレクトリーにある `ibmdadb2.h` ファイルを組み込む必要があります。これらのアプリケーションでは、`include` ステートメントの前に `DBINITCONSTANTS` を定義する必要があります。以下の例は、C/C++ プリプロセッサ・ディレクティブの正しい順序を示しています。

```
#define DBINITCONSTANTS
#include "ibmdadb2.h"
```

IBM OLE DB Provider による、C/C++ アプリケーションでのデータ・ソースへの接続

C/C++ アプリケーションで IBM OLE DB Provider for DB2 を使用して DB2 データ・ソースに接続するには、2 つの OLE DB コア・インターフェース (`IDBPromptInitialize` および `IDataInitialize`) のいずれかを使用するか、COM API `CoCreateInstance` を呼び出すことができます。

C/C++ アプリケーションで IBM OLE DB Provider for DB2 を使用して DB2 データ・ソースに接続するには、2 つの OLE DB コア・インターフェース (`IDBPromptInitialize` および `IDataInitialize`) のいずれかを使用するか、COM API `CoCreateInstance` を呼び出すことができます。 `IDataInitialize` インターフェースは OLE DB Service Component によって提供され、`IDBPromptInitialize` は Data Links Component によって提供されます。

COM+ 分散トランザクションのサポートと IBM OLE DB Provider

Windows 2000 または XP の Microsoft Component Services (COM+) 環境で実行される OLE DB アプリケーションは、`ITransactionJoin` インターフェースを使用できます。

Windows 2000 または XP 上の Microsoft Component Services (COM+) 環境で実行されている OLE DB アプリケーションは、`ITransactionJoin` インターフェースを使用することにより、複数の DB2 for Linux, UNIX, and Windows、ホスト、および System i® データベース・サーバーや COM+ 仕様に準拠する他のリソース・マネージャーとの間で行われる分散トランザクションに参加することができます。

前提条件

IBM OLE DB Provider for DB2 に備わっている COM+ 分散トランザクション・サポートを使用するには、ご使用のサーバーが以下の前提条件を満たしていることを確かめてください。

注：これらの要件は、DB2 クライアントがインストールされている Windows ベースのコンピューターにのみ適用されます。

- Windows 2000 (Service Pack 3 以降)
- Windows XP

C/C++ データベース・アプリケーションでの COM+ サポートの使用可能化

C または C++ アプリケーションを COM+ トランザクション・モードで実行するには、CoCreateInstance を使用して IBMDADB2 データ・ソース・インスタンスを作成し、セッション・オブジェクトを入手して、JoinTransaction を使用できます。

C または C++ アプリケーションを COM+ トランザクション・モードで実行するには、CoCreateInstance を使用して IBMDADB2 データ・ソース・インスタンスを作成し、セッション・オブジェクトを入手して、JoinTransaction を使用できます。詳しくは、C または C++ アプリケーションをデータ・ソースに接続する方法に関する説明を参照してください。

ADO アプリケーションを COM+ トランザクション・モードで実行するには、C または C++ アプリケーションをデータ・ソースに接続する方法に関する説明を参照してください。

COM+ パッケージ内のコンポーネントをトランザクション・モードで使用するには、そのコンポーネントの Transactions プロパティを以下の値のいずれかに設定します。

- 『Required』
- 『Required New』
- 『Supported』

これらの値については、COM+ の資料を参照してください。

第 3 章 IBM Data Server Provider for .NET

IBM Data Server Provider for .NET は、ADO.NET インターフェースのサポートを拡張し、IBM データ・サーバーへのパフォーマンスの高いセキュアなアクセスを提供します。

IBM Data Server Provider for .NET を使用すると、.NET アプリケーションから次のデータベース管理システムへのアクセスが可能になります。

- DB2 for Linux, UNIX, and Windows バージョン 9.1、バージョン 9.5、バージョン 9.7、およびバージョン 9.8
- Windows、UNIX、および Linux 用の DB2 Universal Database バージョン 8
- DB2 for z/OS バージョン 8、バージョン 9、およびバージョン 10 (DB2 Connect 経由)
- DB2 Universal Database for AS/400 バージョン 5 リリース 3 以降 (DB2 Connect 経由)
- IBM Informix バージョン 11.10 およびバージョン 11.50

IBM Data Server Provider for .NET に加えて、IBM Database Add-Ins for Visual Studio を使用すると、Visual Studio でデータベース対応 .NET アプリケーションを迅速かつ容易に開発できます。このアドインを使用して、索引や表などのデータベース・オブジェクトを作成し、ストアド・プロシージャやユーザー定義関数などのサーバー・サイド・オブジェクトを開発することができます。

IBM Data Server Provider for .NET のシステム要件

DB2 Install プログラムを使用して IBM Data Server Provider for .NET をインストールする前に、.NET Framework がコンピューター上にインストール済みでなければなりません。.NET Framework がインストールされていない場合、DB2 Install プログラムは IBM Data Server Provider for .NET をインストールできません。

DB2 Universal Database for AS/400 and System i の場合、以下のフィックスがサーバーに必要です: APAR ii13348

IBM Data Server Provider for .NET 資料の使用方法

IBM Data Server Provider for .NET 参照資料には、プロバイダー固有の詳しい情報を記載しています。IBM Data Server Provider for .NET クラスの多くは、他の .NET Framework クラスまたはインターフェースのメンバーを継承または実装しています。プロバイダー資料には、これらの各クラス内にあるすべてのメンバーのサマリーが含まれています。継承された特定のメンバーについての詳細は、Microsoft .NET Framework SDK 内の該当するトピックを参照してください。

ネーム・スペース

ネーム・スペース	説明
IBM.Data.DB2	IBM データ・サーバーのための、.NET Framework データ・プロバイダーです。

ネーム・スペース	説明
IBM.Data.DB2Types	IBM データ・サーバー内のネイティブ・データ・タイプ用のクラスを提供します。
IBM.Data.Informix	Informix データ・サーバーのための、.NET Framework データ・プロバイダーです。

IBM.Data.DB2 ネーム・スペース

IBM.Data.DB2 ネーム・スペースは、IBM データ・サーバーのための .NET Framework Data Provider です。IBM Data Server Provider for .NET は、ADO.NET インターフェースのサポートを拡張し、データへのパフォーマンスの高いセキュアなアクセスを提供します。このプロバイダーは、Common DB2 .NET Provider と呼ばれることもあります。

IBM Data Server Provider for .NET を使用すると、.NET アプリケーションから次のデータベース管理システムへのアクセスが可能になります。

- DB2 for Linux, UNIX, and Windows バージョン 9.1、バージョン 9.5、バージョン 9.7、およびバージョン 9.8
- DB2 for z/OS バージョン 8、バージョン 9、およびバージョン 10 (DB2 Connect 経由)
- DB2 Universal Database for AS/400 バージョン 5 リリース 3 以降 (DB2 Connect 経由)

IBM Data Server Provider for .NET に加えて、IBM Database Add-Ins for Visual Studio を利用すると、IBM データ・サーバー向けの .NET アプリケーションを Visual Studio で迅速かつ容易に開発できます。このアドインを使用して、索引や表などのデータベース・オブジェクトを作成し、ストアード・プロシージャやユーザー定義関数などのサーバー・サイド・オブジェクトを開発することができます。

IBM Data Server Provider for .NET のシステム要件

DB2 Install プログラムを使用して IBM Data Server Provider for .NET をインストールする前に、.NET Framework がコンピューター上にインストール済みでなければなりません。 .NET Framework がインストールされていない場合、DB2 Install プログラムは IBM Data Server Provider for .NET をインストールしません。

DB2 for i V5R4 以前の場合、サーバー上で APAR II13348 の修正を適用する必要があります。

IBM Data Server Provider for .NET 資料の使用方法

IBM Data Server Provider for .NET 参照資料には、プロバイダー固有の詳しい情報を記載しています。IBM Data Server Provider for .NET クラスの多くは、他の .NET Framework クラスまたはインターフェースのメンバーを継承または実装しています。プロバイダー資料には、これらの各クラス内にあるすべてのメンバーのサマリーが含まれています。継承された特定のメンバーについての詳細は、Microsoft .NET Framework SDK 内の該当するトピックを参照してください。

IBM Data Server Provider for .NET アプリケーションの開発

IBM Data Server Provider for .NET は、データベースへの接続、コマンドの実行、および結果の取得の機能を提供します。それらの結果は、直接処理することも、ADO.NET DataSet に入れて切断状態のときに処理を続行することもできます。DataSet に入れたデータは、ユーザーへの公開、複数のソースからの他のデータとの結合、または複数の層の間でリモート操作の受け渡しが可能です。データが DataSet 内に存在するときにそのデータに対して実行する処理に基づいて、データベースを調整することができます。

IBM Data Server Provider for .NET を使用するには、Imports または using ステートメントで IBM.Data.DB2 ネーム・スペースをアプリケーション .DLL に追加する必要があります。例:

```
[Visual Basic]
Imports IBM.Data.DB2
[C#]
using IBM.Data.DB2;
```

さらに、IBM.Data.DB2.d11 への参照をアプリケーションのプロジェクトに組み込む必要があります。Visual Studio ではこれを、Solution Explorer で自分のプロジェクトの References セクションから、または **Project > Add Reference** メニュー・オプションから行うことができます。Framework 2.0 以降のコマンド行からプログラムをコンパイルする場合は、**csc** または **vbc** コマンドに以下のオプションを組み込む必要があります。

```
/r:<install dir>\bin\netf20\IBM.Data.DB2.d11
```

サポートされているデータ・サーバーのうちの 1 つへの接続を確立するには、DB2Connection オブジェクトを構成し、そこに有効な DB2 .NET 接続ストリングを提供する必要があります。サポートされているキーワードの詳細は DB2Connection.ConnectionString プロパティの項を参照してください。

IBM.Data.DB2 ネーム・スペースの最も有効な使用方法の詳細は、以下の IBM Data Server Provider for .NET クラスに関する資料を参照してください。

- DB2DataAdapter
- DB2Command
- DB2Connection
- DB2DataReader

クラス

クラス	説明
 DB2BulkCopy	あるデータ・ソースから別のデータ・ソースへの行のコピーを容易にします。
 DB2BulkCopyColumnMapping	データ・ソース表から宛先表への列のマッピングを表します。
 DB2BulkCopyColumnMappingCollection	データ・ソース表から宛先表への列のマッピングの集合を表します。
 DB2Command	データベースに対して実行する SQL ステートメントまたはストアード・プロシージャを表します。

クラス	説明
 DB2CommandBuilder	関連するデータベースをDataSet に対する変更に基づいて調整するために使用する単一表コマンドを自動生成します。
 DB2Connection	データベースへのオープン接続を表します。
 DB2ConnectionStringBuilder	有効な接続ストリングを生成するための、汎用アプローチおよび IBM Data Server Provider for .NET 固有のアプローチを容易に行えるようにします。
 DB2DataAdapter	DataSet にデータを追加してデータベースを更新するために使用する、データ・コマンドおよびデータベースへの接続のセットを表します。
 DB2DataReader	順方向のみのデータ行のストリームをデータベースから読み取る手段を提供します。
 DB2DataSourceEnumerator	可視の IBM ファミリー・データ・ソースを検出する方法を提供します。
 DB2Error	データベースによって戻された警告またはエラーに関連する情報を収集します。
 DB2ErrorCollection	DB2DataAdapterによって生成させるすべてのエラーを収集します。
 DB2Exception	IBM データ・サーバーによってエラーが戻されるときに生成される例外。
 DB2Factory	データ・プロバイダー用の System.Data.Common データ・ソース・クラスのインスタンスを作成するためのメソッドのセットを表します。
 DB2InfoMessageEventArgs	InfoMessage イベントにデータを提供します。
 DB2Parameter	DB2Command へのパラメーターを表し、オプションで、DataColumn へのマッピングを表します。
 DB2ParameterCollection	DB2Command に関連したパラメーターのコレクションと、DataSet 内の列に対する各パラメーターのマッピングを表します。
 DB2Permission	IBM データベースにアクセスするのに必要なセキュリティ・レベルがユーザーにあることを、IBM Data Server Provider for .NET が確認できるようにします。
 DB2PermissionAttribute	セキュリティ・アクションをカスタム・セキュリティ属性に関連付けます。
 DB2Record	読み取り専用レコードを表します。
 DB2ResultSet	データベースからのバインド可能なデータ行のストリームによって、両方向スクロールを提供します。加えて、DB2ResultSet インスタンスから行を更新、削除、および挿入することもできます。
 DB2RowsCopiedEventArgs	DB2RowsCopied イベントにデータを提供します。
 DB2RowUpdatedEventArgs	RowUpdated イベントにデータを提供します。

クラス	説明
 DB2RowUpdatingEventArgs	RowUpdating イベントにデータを提供します。
 DB2Transaction	データベースで行う SQL トランザクションを表します。
 DB2UpdatableRecord	DB2ResultSet インスタンスに作成される新規行を表します。
 DB2XmlAdapter	データベースからの取り出しと XPathDocument インスタンスへの入力の手段を表します。

デリゲート

デリゲート	説明
 DB2InfoMessageEventHandler	DB2Connection オブジェクトの InfoMessage イベントを処理するメソッドを表します。
 DB2RowsCopiedEventHandler	DB2BulkCopy オブジェクトの DB2RowsCopied イベントを処理するメソッドを表します。
 DB2RowUpdatedEventHandler	DB2DataAdapter オブジェクトの RowUpdated イベントを処理するメソッドを表します。
 DB2RowUpdatingEventHandler	DB2DataAdapter オブジェクトの RowUpdating イベントを処理するメソッドを表します。

列挙

列挙	説明
 DB2BulkCopyOptions	DB2BulkCopy で使用するオプションを指定します。それらのオプションはビット・フラグであり、ビット・マスク内でそれらを組み合わせることができます。
 DB2CursorType	DB2ResultSet で使用するカーソル・オプションを指定します。それらのオプションはビット・フラグであり、ビット・マスク内でそれらを組み合わせることができます。
 DB2ResultSetOptions	DB2ResultSet で使用する結果セット・オプションを指定します。それらのオプションはビット・フラグであり、ビット・マスク内でそれらを組み合わせることができます。
 DB2Type	フィールド、プロパティ、または DB2Parameter のデータ・タイプを指定します。

参照情報

84 ページの『IBM.Data.DB2Types ネーム・スペース』
 IBM.Data.DB2Types ネーム・スペースは、.NET Framework Data Provider 用の DB2 データ・タイプを表すクラスと構造体を提供します。

IBM Data Server Provider for .NET は、ADO.NET インターフェースのサポートを拡張し、IBM データ・サーバーへのパフォーマンスの高いセキュアなアクセスを提供します。

IBM.Data.DB2Types ネーム・スペース

IBM.Data.DB2Types ネーム・スペースは、.NET Framework Data Provider 用の DB2 データ・タイプを表すクラスと構造体を提供します。

以下の表は、DB2Type データ・タイプ、DB2 データ・タイプ、Informix データ・タイプ、Microsoft .NET Framework のタイプ、および DB2Types クラスおよび構造体間のマッピングを示しています。

表 45. データ・タイプ、クラスおよび構造体間のマッピング

カテゴリー	DB2Types クラスと構造体	DB2Type データ・タイプ	DB2 データ・タイプ	Informix データ・タイプ	.NET データ・タイプ
Numeric	DB2Int16	SmallInt	SMALLINT	BOOLEAN, SMALLINT	Int16
	DB2Int32	Integer	INT	INTEGER, INT, SERIAL	Int32
	DB2Int64	BigInt, BigSerial	BIGINT	BIGINT, BIGSERIAL, INT8, SERIAL8	Int64
	DB2Real, DB2Real370	Real	REAL	REAL, SMALLFLOAT	Single
	DB2Double	Double	DOUBLE PRECISION	DECIMAL (≤31), DOUBLE PRECISION	Double
	DB2Double	Float	FLOAT	DECIMAL (32), FLOAT	Double
	DB2Decimal	Decimal	DECIMAL	MONEY	Decimal
	DB2Decimal Float	DecimalFloat	DECFLOAT (16 34)		Decimal
	DB2Decimal	Numeric	DECIMAL	DECIMAL (≤31), NUMERIC	Decimal
Date/Time	DB2Date	Date	DATE	DATETIME (日付精度)	Datetime
	DB2Time	Time	TIME	DATETIME (時刻精度)	TimeSpan
	DB2TimeStamp	Timestamp	TIMESTAMP	DATETIME (時刻と日付の精度)	DateTime
	DB2TimeStampOffset	Timestamp WithTimeZone	TIMESTAMP WITH TIME ZONE	なし	DateTimeOffset
XML	DB2Xml	XmlIxfType.Xml	XML		Byte[]
文字データ	DB2String	Char	CHAR	CHAR	String
	DB2String	VarChar	VARCHAR	VARCHAR	String
	DB2String	LongVarChar	LONG VARCHAR	LVARCHAR	String
バイナリー・データ	DB2Binary	Binary	CHAR FOR BIT DATA		Byte[]
	DB2Binary	Binary	BINARY		Byte[]
	DB2Binary	VarBinary	VARBINARY		Byte[]
	DB2Binary	Long VarBinary	LONG VARCHAR FOR BIT DATA		Byte[]

6. これらのデータ・タイプは、DB2 .NET 共通言語ランタイム・ルーチンではパラメーターとしてサポートされていません。

7. DB2Type.Xml タイプの DB2ParameterClass.ParameterName プロパティは、String、byte[]、DB2Xml、および XmlReader タイプの変数を受け入れることができます。

8. これらのデータ・タイプを適用できるのは、DB2 for z/OS だけです。

9. このデータ・タイプは、DB2 for z/OS バージョン 9 以降のリリース、および DB2 for Linux, UNIX, and Windows バージョン 9.5 以降のリリースでのみサポートされます。

表 45. データ・タイプ、クラスおよび構造体間のマッピング (続き)

カテゴリー	DB2Types クラスと構造体	DB2Type データ・タイプ	DB2 データ・タイプ	Informix データ・タイプ	.NET データ・タイプ
グラフィック・データ	DB2String	Graphic	GRAPHIC		String
	DB2String	VarGraphic	VARGRAPHIC		String
	DB2String	Long VarGraphic	LONG VARGRAPHIC		String
LOB データ	DB2Clob	Clob	CLOB	CLOB, TEXT	String
	DB2Blob	Blob	BLOB	BLOB, BYTE	Byte[]
	DB2Clob	DbClob	DBCLOB		String
Row ID	DB2RowId	RowId	ROWID		Byte[]

列挙

列挙	説明
 DB2RoundingMode	DB2DecimalFloat オブジェクトで使用する丸めモードを指定します。

付録 A. DB2 技術情報の概説

DB2 技術情報は、さまざまな方法でアクセスすることが可能な、各種形式で入手できます。

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)
 - 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、IBM Publications Center (www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss) から利用できる DB2 ライブラリーについて説明しています。英語および翻訳された DB2 バージョン 10.1 のマニュアル (PDF 形式) は、www.ibm.com/support/docview.wss?rs=71&uid=swg27009474 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 46. DB2 の技術情報

資料名	資料番号	印刷資料が入手可能かどうか	最終更新
管理 API リファレンス	SA88-4671-00	入手可能	2012 年 4 月
管理ルーチンおよびビュー	SA88-4672-01	入手不可	2013 年 1 月
コール・レベル・イン ターフェース ガイドお よびリファレンス 第 1 巻	SA88-4676-01	入手可能	2013 年 1 月
コール・レベル・イン ターフェース ガイドお よびリファレンス 第 2 巻	SA88-4677-01	入手可能	2013 年 1 月
コマンド・リファレン ス	SA88-4673-01	入手可能	2013 年 1 月
データベース: 管理の 概念および構成リファ レンス	SA88-4662-01	入手可能	2013 年 1 月
データ移動ユーティリ ティー: ガイドおよび リファレンス	SA88-4693-01	入手可能	2013 年 1 月
データベースのモニタ リング ガイドおよびリ ファレンス	SA88-4663-01	入手可能	2013 年 1 月
データ・リカバリーと 高可用性 ガイドおよび リファレンス	SA88-4694-01	入手可能	2013 年 1 月
データベース・セキュ リティー・ガイド	SA88-4695-01	入手可能	2013 年 1 月

表 46. DB2 の技術情報 (続き)

資料名	資料番号	印刷資料が入手可能 かどうか	最終更新
DB2 ワークロード管理 ガイドおよびリファレ ンス	SA88-4685-01	入手可能	2013 年 1 月
ADO.NET および OLE DB アプリケーション の開発	SA88-4665-01	入手可能	2013 年 1 月
組み込み SQL アプリ ケーションの開発	SA88-4666-01	入手可能	2013 年 1 月
Java アプリケーション の開発	SA88-4669-01	入手可能	2013 年 1 月
Perl、PHP、Python お よび Ruby on Rails ア プリケーションの開発	SA88-4670-00	入手不可	2012 年 4 月
IBM データ・サーバー 用の RDF アプリケー ション開発	SA88-5083-00	入手可能	2013 年 1 月
SQL および外部ルーチ ンの開発	SA88-4667-01	入手可能	2013 年 1 月
データベース・アプリ ケーション開発の基礎	GI88-4279-01	入手可能	2013 年 1 月
DB2 インストールおよ び管理 概説 (Linux お よび Windows 版)	GI88-4280-00	入手可能	2012 年 4 月
グローバリゼーショ ン・ガイド	SA88-4696-00	入手可能	2012 年 4 月
DB2 サーバー機能 イ ンストール	GA88-4679-01	入手可能	2013 年 1 月
IBM データ・サーバ ー・クライアント機能 インストール	GA88-4680-00	入手不可	2012 年 4 月
メッセージ・リファレ ンス 第 1 巻	SA88-4688-01	入手不可	2013 年 1 月
メッセージ・リファレ ンス 第 2 巻	SA88-4689-01	入手不可	2013 年 1 月
Net Search Extender 管 理およびユーザース・ ガイド	SA88-4691-01	入手不可	2013 年 1 月
パーティションおよび クラスタリングのガイ ド	SA88-4697-01	入手可能	2013 年 1 月
Preparation Guide for DB2 10.1 Fundamentals Exam 610	SC27-4540-00	入手不可	2013 年 1 月

表 46. DB2 の技術情報 (続き)

資料名	資料番号	印刷資料が入手可能 かどうか	最終更新
<i>Preparation Guide for DB2 10.1 DBA for Linux, UNIX, and Windows Exam 611</i>	SC27-4541-00	入手不可	2013 年 1 月
<i>pureXML ガイド</i>	SA88-4686-01	入手可能	2013 年 1 月
<i>Spatial Extender ユーザーズ・ガイドおよびリファレンス</i>	SA88-4690-00	入手不可	2012 年 4 月
<i>SQL プロシージャ言語: アプリケーションのイネーブルメントおよびサポート</i>	SA88-4668-01	入手可能	2013 年 1 月
<i>SQL リファレンス 第 1 巻</i>	SA88-4674-01	入手可能	2013 年 1 月
<i>SQL リファレンス 第 2 巻</i>	SA88-4675-01	入手可能	2013 年 1 月
<i>Text Search ガイド</i>	SA88-4692-01	入手可能	2013 年 1 月
<i>問題判別およびデータベース・パフォーマンスのチューニング</i>	SA88-4664-01	入手可能	2013 年 1 月
<i>DB2 バージョン 10.1 へのアップグレード</i>	SA88-4678-01	入手可能	2013 年 1 月
<i>DB2 バージョン 10.1 の新機能</i>	SA88-4684-01	入手可能	2013 年 1 月
<i>XQuery リファレンス</i>	SA88-4687-01	入手不可	2013 年 1 月

表 47. DB2 Connect 固有の技術情報

資料名	資料番号	印刷資料が入手可能 かどうか	最終更新
<i>DB2 Connect Personal Edition</i> インストールおよび構成	SA88-4681-00	入手可能	2012 年 4 月
<i>DB2 Connect サーバー機能</i> インストールおよび構成	SA88-4682-01	入手可能	2013 年 1 月
<i>DB2 Connect ユーザーズ・ガイド</i>	SA88-4683-01	入手可能	2013 年 1 月

コマンド行プロセッサから SQL 状態ヘルプを表示する

DB2 製品は、SQL ステートメントの結果として生じる可能性がある状態に対応した SQLSTATE 値を戻します。SQLSTATE ヘルプは、SQL 状態および SQL 状態クラス・コードの意味を説明します。

手順

SQL 状態ヘルプを開始するには、コマンド行プロセッサを開いて以下のように入力します。

```
? sqlstate または ? class code
```

ここで、*sqlstate* は有効な 5 桁の SQL 状態を、*class code* は SQL 状態の最初の 2 桁を表します。

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

他のバージョンの DB2 製品の資料は、ibm.com[®] のそれぞれのインフォメーション・センターにあります。

このタスクについて

DB2 バージョン 10.1 のトピックを扱っている DB2 インフォメーション・センターの URL は、<http://publib.boulder.ibm.com/infocenter/db2luw/v10r1> です。

DB2 バージョン 9.8 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/v9r8/> です。

DB2 バージョン 9.7 のトピックを扱っている DB2 インフォメーション・センターの URL は、<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/> です。

DB2 バージョン 9.5 のトピックを扱っている DB2 インフォメーション・センターの URL は、<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5> です。

DB2 バージョン 9.1 のトピックを扱っている DB2 インフォメーション・センターの URL は、<http://publib.boulder.ibm.com/infocenter/db2luw/v9/> です。

DB2 バージョン 8 のトピックについては、DB2 インフォメーション・センターの URL (<http://publib.boulder.ibm.com/infocenter/db2luw/v8/>) を参照してください。

コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの更新

ローカルにインストールした DB2 インフォメーション・センターは、定期的に更新する必要があります。

始める前に

DB2 バージョン 10.1 インフォメーション・センターが既にインストール済みである必要があります。詳しくは、「DB2 サーバー機能 インストール」の『DB2 セットアップ・ウィザードによる DB2 インフォメーション・センターのインストール』のトピックを参照してください。インフォメーション・センターのインストールに適用されるすべての前提条件と制約事項は、インフォメーション・センターの更新にも適用されます。

このタスクについて

既存の DB2 インフォメーション・センターは、自動で更新することも手動で更新することもできます。

- 自動更新は、既存のインフォメーション・センターのフィーチャーと言語を更新します。自動更新を使用すると、手動更新と比べて、更新中にインフォメーション・センターが使用できなくなる時間が短くなるというメリットがあります。さらに、自動更新は、定期的に行う他のバッチ・ジョブの一部として実行されるように設定することができます。
- 手動更新は、既存のインフォメーション・センターのフィーチャーと言語の更新に使用できます。自動更新は更新処理中のダウン時間を減らすことができますが、フィーチャーまたは言語を追加する場合は手動処理を使用する必要があります。例えば、ローカルのインフォメーション・センターが最初は英語とフランス語でインストールされており、その後ドイツ語もインストールすることにした場合、手動更新でドイツ語をインストールし、同時に、既存のインフォメーション・センターのフィーチャーおよび言語を更新できます。しかし、手動更新ではインフォメーション・センターを手動で停止、更新、再始動する必要があります。更新処理の間はずっと、インフォメーション・センターは使用できなくなります。自動更新処理では、インフォメーション・センターは、更新を行った後に、インフォメーション・センターを再始動するための停止が発生するだけで済みます。

このトピックでは、自動更新のプロセスを詳しく説明しています。手動更新の手順については、『コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの手動更新』のトピックを参照してください。

手順

コンピューターまたはイントラネット・サーバーにインストールされている DB2 インフォメーション・センターを自動更新する手順を以下に示します。

1. Linux オペレーティング・システムの場合、次のようにします。
 - a. インフォメーション・センターがインストールされているパスにナビゲートします。デフォルトでは、DB2 インフォメーション・センターは、`/opt/ibm/db2ic/V10.1` 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 `update-ic` スクリプトを実行します。

```
update-ic
```
2. Windows オペレーティング・システムの場合、次のようにします。
 - a. コマンド・ウィンドウを開きます。
 - b. インフォメーション・センターがインストールされているパスにナビゲートします。デフォルトでは、DB2 インフォメーション・センターは、`<Program Files>%IBM%DB2 Information Center%バージョン 10.1` 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 update-ic.bat ファイルを実行します。

```
update-ic.bat
```

タスクの結果

DB2 インフォメーション・センターが自動的に再始動します。更新が入手可能な場合、インフォメーション・センターに、更新された新しいトピックが表示されます。インフォメーション・センターの更新が入手可能でなかった場合、メッセージがログに追加されます。ログ・ファイルは、doc¥eclipse¥configuration ディレクトリーにあります。ログ・ファイル名はランダムに生成された名前です。例えば、1239053440785.log のようになります。

コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの手動更新

DB2 インフォメーション・センターをローカルにインストールしている場合は、IBM から資料の更新を入手してインストールすることができます。

このタスクについて

ローカルにインストールされた DB2 インフォメーション・センター を手動で更新するには、以下のことを行う必要があります。

1. コンピューター上の DB2 インフォメーション・センター を停止し、インフォメーション・センターをスタンドアロン・モードで再始動します。インフォメーション・センターをスタンドアロン・モードで実行すると、ネットワーク上の他のユーザーがそのインフォメーション・センターにアクセスできなくなります。これで、更新を適用できるようになります。DB2 インフォメーション・センターのワークステーション・バージョンは、常にスタンドアロン・モードで実行されます。を参照してください。
2. 「更新」機能を使用することにより、どんな更新が利用できるかを確認します。インストールしなければならない更新がある場合は、「更新」機能を使用してそれを入手およびインストールできます。

注: ご使用の環境において、インターネットに接続されていないマシンに DB2 インフォメーション・センター の更新をインストールする必要がある場合、インターネットに接続されていて DB2 インフォメーション・センター がインストールされているマシンを使用して、更新サイトをローカル・ファイル・システムにミラーリングしてください。ネットワーク上の多数のユーザーが資料の更新をインストールする場合にも、更新サイトをローカルにミラーリングして、更新サイト用のプロキシを作成することにより、個々のユーザーが更新を実行するのに要する時間を短縮できます。

更新パッケージが入手可能な場合、「更新」機能を使用してパッケージを入手します。ただし、「更新」機能は、スタンドアロン・モードでのみ使用できます。

3. スタンドアロンのインフォメーション・センターを停止し、コンピューター上の DB2 インフォメーション・センター を再開します。

注: Windows 2008、Windows Vista (およびそれ以上) では、このセクションの後の部分でリストされているコマンドは管理者として実行する必要があります。完全な管理者特権でコマンド・プロンプトまたはグラフィカル・ツールを開くには、ショートカットを右クリックしてから、「管理者として実行」を選択します。

手順

コンピューターまたはイントラネット・サーバーにインストール済みの *DB2* インフォメーション・センター を更新するには、以下のようにします。

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/db2icdv10 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¥バージョン 10.1` 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/V10.1` ディレクトリーにインストールされています。
 - b. インストール・ディレクトリーから `doc/bin` ディレクトリーにナビゲートします。
 - c. 次のように `help_start` スクリプトを実行します。

```
help_start
```
- システムのデフォルト Web ブラウザーが開き、スタンドアロンのインフォメーション・センターが表示されます。
3. 「更新」ボタン (🔄) をクリックします。(ブラウザーで JavaScript が有効になっている必要があります。) インフォメーション・センターの右側のパネルで、「更新の検索」をクリックします。既存の文書に対する更新のリストが表示されます。
 4. インストール・プロセスを開始するには、インストールする更新をチェックして選択し、「更新のインストール」をクリックします。
 5. インストール・プロセスが完了したら、「完了」をクリックします。

6. 次のようにして、スタンドアロンのインフォメーション・センターを停止します。

- Windows の場合は、インストール・ディレクトリーの `doc\bin` ディレクトリーにナビゲートしてから、次のように `help_end.bat` ファイルを実行します。

```
help_end.bat
```

注: `help_end` バッチ・ファイルには、`help_start` バッチ・ファイルを使用して開始したプロセスを安全に停止するのに必要なコマンドが含まれています。`help_start.bat` は、Ctrl-C や他の方法を使用して停止しないでください。

- Linux の場合は、インストール・ディレクトリーの `doc/bin` ディレクトリーにナビゲートしてから、次のように `help_end` スクリプトを実行します。

```
help_end
```

注: `help_end` スクリプトには、`help_start` スクリプトを使用して開始したプロセスを安全に停止するのに必要なコマンドが含まれています。他の方法を使用して、`help_start` スクリプトを停止しないでください。

7. **DB2** インフォメーション・センター を再開します。

- Windows では、「スタート」 > 「コントロール パネル」 > 「管理ツール」 > 「サービス」をクリックします。次に、「**DB2** インフォメーション・センター」サービスを右クリックして「開始」を選択します。

- Linux では、以下のコマンドを入力します。

```
/etc/init.d/db2icdv10 start
```

タスクの結果

更新された **DB2** インフォメーション・センター に、更新された新しいトピックが表示されます。

DB2 チュートリアル

DB2 チュートリアルは、DB2 データベース製品のさまざまな機能について学習するための支援となります。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/db2luw/v10r1/>) 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。

DB2 トラブルシューティング情報

DB2 データベース製品を使用する際に役立つ、トラブルシューティングおよび問題判別に関する広範囲な情報を利用できます。

DB2 の資料

トラブルシューティング情報は、「問題判別およびデータベース・パフォーマンスのチューニング」または *DB2* インフォメーション・センター の『データベースの基本』セクションにあります。ここには、以下の情報が記載されています。

- DB2 診断ツールおよびユーティリティーを使用した、問題の切り分け方法および識別方法に関する情報。
- 最も一般的な問題のうち、いくつかの解決方法。
- DB2 データベース製品で発生する可能性のある、その他の問題の解決に役立つアドバイス。

IBM サポート・ポータル

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見つけるには、IBM サポート・ポータル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。

IBM サポート・ポータル (http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/DB2_for_Linux,_UNIX_and_Windows) 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。

適用度: これらのご利用条件は、IBM Web サイトのあらゆるご利用条件に追加で適用されるものです。

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。

権利: 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。

IBM の商標: IBM、IBM ロゴおよび ibm.com は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。

付録 B. 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。IBM 以外の製品に関する情報は、本書の最初の発行時点で入手可能な情報に基づいており、変更される場合があります。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510
東京都中央区日本橋箱崎町19番21号
日本アイ・ビー・エム株式会社
法務・知的財産
知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。 IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

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

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Canada Limited
U59/3600
3600 Steeles Avenue East
Markham, Ontario L3R 9Z7
CANADA

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができませんが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのも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。サンプル・プログラムは、現存するままの状態を提供されるものであり、いかなる種類の保証も提供されません。IBM は、これらのサンプル・プログラムの使用から生ずるいかなる損害に対しても責任を負いません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生した創作物には、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。© Copyright IBM Corp. _年を入れる_. All rights reserved.

商標

IBM、IBM ロゴおよび ibm.com は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。

以下は、それぞれ各社の商標または登録商標です。

- Linux は、Linus Torvalds の米国およびその他の国における商標です。
- Java™ およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。
- 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。
- インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Celeron、Intel SpeedStep、Itanium、Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。
- Microsoft、Windows、Windows NT、および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

アプリケーション

データ・ソースへの接続

IBM OLE DB Provider 76

ADO

更新可能な両方向スクロール・カーソル 69

制限 69

IBM OLE DB Provider 48

Visual Basic 69

アプリケーション開発

IBM Data Server Provider for .NET 5

IBM Database Add-Ins for Visual Studio 4

[カ行]

カーソル

更新可能

ADO アプリケーション 69

両方向スクロール

ADO アプリケーション 69

IBM OLE DB Provider 52

結果セット

読み取り

IBM Data Server Provider for .NET 15

更新

DB2 インフォメーション・センター 91, 93

ご利用条件

資料 96

[サ行]

システム要件

IBM OLE DB Provider for DB2 47

資料

印刷 88

概要 87

使用に関するご利用条件 96

PDF ファイル 88

スキーマ

行セット 49

スレッド

IBM OLE DB Provider for DB2 47, 48

接続キーワード

ODBC .NET Data Provider 38

OLE DB .NET Data Provider 30

接続プール

IBM Data Server Provider for .NET 9

OLE DB .NET Data Provider 35

[タ行]

チュートリアル

トラブルシューティング 96

問題判別 96

リスト 95

pureXML 95

データ・タイプ

マッピング

OLE DB と DB2 52

ADO.NET データベース・アプリケーション 11

特記事項 99

トラステッド・コンテキスト

接続ストリング・キーワード 9

トラブルシューティング

オンライン情報 96

チュートリアル 96

[ハ行]

プロパティ

OLE DB 63

ヘルプ

SQL ステートメント 91

[マ行]

問題判別

チュートリアル 96

利用できる情報 96

[ラ行]

ラージ・オブジェクト (LOB)

IBM OLE DB Provider 48

A

ActiveX Data Object (ADO) の仕様

IBM Data Server Provider for .NET 5

ADO アプリケーション

更新可能な両方向スクロール・カーソル 69

ストアド・プロシージャ 69

制限 69

接続ストリング・キーワード 68

ADO アプリケーション (続き)

- IBM OLE DB Provider での ADO メソッドおよびプロパティのサポート 70
- ADORRecordset オブジェクト 37
- ADO.NET アプリケーション
 - 開発 1
 - 共通基本クラス 7

C

COM

- 分散トランザクション・サポート 76

COM+ アプリケーション

- 接続プール 9

C# .NET

- アプリケーション
 - 構築 (Windows) 25
 - コンパイラー・オプション 28
 - リンク・オプション 28

C/C++ 言語

- アプリケーション
 - IBM OLE DB Provider 76

D

DB2 インフォメーション・センター

- 更新 91, 93
- バージョン 91

E

- Enterprise Library データ・アクセス・モジュール 24

I

IBM Data Server Provider for .NET

- 概要 1, 5, 7, 24, 79
- 共通基本クラス 7
- 結果セットの読み取り 15
- 結果セットへの同時アクセス 18
- ストアード・プロシージャの呼び出し 16
- 接続プール 9
- データベースへの接続 8
- データベース・システム要件 6
- データ・タイプ 11
- Microsoft Entity Framework 20
- SQL ステートメント 12

IBM Data Server Provider のトレース

- System.Diagnostics.Trace 19

IBM Database Add-Ins for Visual Studio

- 概要 4

IBM OLE DB Provider

- アプリケーション・タイプ 48
- インストール 47
- カーソル 52, 69

IBM OLE DB Provider (続き)

- 概要 47
- コンシューマー 47
- システム要件 47
- スキーマ行セット 49
- スレッド化 48
- 制限 59
- データ変換
 - DB2 タイプを OLE DB タイプへ 56
 - OLE DB タイプを DB2 タイプへ 54

- データ・ソースへの接続 67, 69

- バージョン 47

- プロバイダー 47

ADO

- アプリケーション 68, 69
- プロパティ 70
- メソッド 70

- COM サポート 76, 77

- C/C++ アプリケーション 76

- LOB 48

- OLE DB インターフェース 60

- OLE DB コンポーネント 60

- OLE DB サービス 51

- OLE DB プロパティ 63

- Visual Basic アプリケーション 69

- IBM.Data.DB2 ネーム・スペース 80

- IBM.Data.IfzTypesDB2Types ネーム・スペース 84

M

Microsoft Entity Framework

- IBM Data Server Provider for .NET 20

Microsoft OLE DB Provider (ODBC 用)

- OLE DB のサポート 60

Microsoft Transaction Server (MTS)

- COM 分散トランザクション・サポート 76
- DB2 でのサポート 77

O

ODBC .NET Data Provider

- 概要 1, 38
- 制約事項 38

OLE DB

- 自動的に使用可能になるサービス 51

データ変換

- DB2 から OLE DB タイプへ 56

- OLE DB タイプから DB2 タイプへ 54

データ・タイプ

- DB2 データ・タイプへのマッピング 52

表関数 47

- IBM OLE DB Provider サポートの概要 60

- IBM OLE DB Provider によるデータ・ソースへの接続 67

- OLE DB .NET Data Provider がサポートするプロパティ

- 63

OLE DB .NET Data Provider

概要 1, 30

制約事項 31

ADORRecordset オブジェクト 37

OLE アプリケーション

時刻列 36

接続プール 35

.NET (続き)

Visual Basic アプリケーション (続き)

ストアド・プロシージャ 16

データベース接続 8

リンク・オプション 26

SQL ステートメント 12

Windows 24

S

SAMPLE データベース

接続

ODBC .NET Data Provider 38

OLE DB .NET Data Provider 30

SQL ステートメント

実行

IBM Data Server Provider for .NET 12

ヘルプ

表示 91

SystemDiagnosticsTrace

トレース

IBM Data Server Provider for .NET 19

V

Visual Basic

アプリケーション 69

カーソルの考慮事項 69

データ制御サポート 69

Visual Basic .NET

アプリケーション

構築 24

コンパイラー・オプション 26

リンク・オプション 26

[特殊文字]

.NET

アプリケーション開発ソフトウェア 3

アプリケーションのデプロイメント 2

C# アプリケーション

結果セット 15

コンパイラー・オプション 28

ストアド・プロシージャ 16

データベース接続 8

リンク・オプション 28

SQL ステートメント 12

Windows 25

IBM Data Server Provider

System.Diagnostics.Trace 19

pureQuery

照会の最適化 20

Visual Basic アプリケーション

結果セット 15

コンパイラー・オプション 26



Printed in Japan

SA88-4665-01



日本アイ・ビー・エム株式会社
〒103-8510 東京都中央区日本橋箱崎町19-21

Spine information:

IBM DB2 10.1 for Linux, UNIX, and Windows

ADO.NET および OLE DB アプリケーションの開発

