

IBM DB2 10.1
for Linux, UNIX, and Windows

**开发 ADO.NET 和 OLE DB 应
用程序**

更新时间 2013 年 1 月

IBM

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

**开发 ADO.NET 和 OLE DB 应
用程序**

更新时间 2013 年 1 月

IBM

注意

使用此信息及其支持的产品前，请先阅读第 83 页的附录 B，『声明』下的常规信息。

修订版声明

此文档包含 IBM 的所有权信息。它在许可协议中提供，且受版权法的保护。本出版物中包含的信息不包括对任何产品的保证，且提供的任何语句都不需要如此解释。

您可在线或通过当地的 IBM 代表处订购 IBM 出版物。

- 要在线订购出版物，请转至 IBM 出版物中心，网址为：<http://www.ibm.com/shop/publications/order>
- 要查找当地的 IBM 代表处，请转至 IBM 全球联系人目录，网址为：<http://www.ibm.com/planetwide/>

要从美国或加拿大的 DB2 市场和销售部订购 DB2 出版物，请致电 1-800-IBM-4YOU（426-4968）。

您发送信息给 IBM 后，即授予 IBM 非独占权限，IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

目录

第 1 章 ADO.NET 应用程序开发	1	使用 IBM OLE DB Provider 来连接到数据源	58
部署 .NET 应用程序 (Windows)	2	ADO 应用程序	58
受支持的 .NET 开发软件	2	ADO 连接字符串关键字	58
在 Visual Studio 中集成 DB2	3	使用 Visual Basic ADO 应用程序连接至数据源	59
IBM Data Server Provider for .NET 概述	4	ADO 应用程序中的可更新可滚动游标	59
IBM Data Server Provider for .NET 数据库系统要求	5	ADO 应用程序的限制	59
编写应用程序以使用 IBM Data Server Provider for .NET	5	IBM OLE DB Provider 对 ADO 方法和属性的支持	60
构建 .NET 应用程序	20	C/C++ 应用程序的编译和链接以及 IBM OLE DB Provider	66
OLE DB .NET Data Provider	25	在 C/C++ 应用程序中使用 IBM OLE DB Provider 来连接到数据源	66
OLE DB .NET Data Provider 限制	26	COM+ 分布式事务支持和 IBM OLE DB Provider	66
提示与技巧	29	在 C/C++ 数据库应用程序中启用 COM+ 支持	66
ODBC .NET Data Provider	31	第 3 章 IBM Data Server Provider for .NET	69
ODBC .NET Data Provider 限制	31	IBM.Data.DB2 名称空间	70
第 2 章 IBM OLE DB Provider for DB2	39	IBM.Data.DB2Types 名称空间	73
IBM OLE DB Provider for DB2 支持的应用程序类型	39	附录 A. DB2 技术信息概述	75
OLE DB 服务	40	硬拷贝或 PDF 格式的 DB2 技术库	75
IBM OLE DB Provider 所支持的线程模型	40	从命令行处理器显示 SQL 状态帮助	77
通过 IBM OLE DB Provider 进行大对象处理	40	访问不同版本的 DB2 信息中心	78
IBM OLE DB Provider 所支持的模式行集	40	更新安装在计算机或内部网服务器上的 DB2 信息中心	78
IBM OLE DB Provider 自动启用的 OLE DB 服务	43	手动更新安装在计算机或内部网服务器上的 DB2 信息中心	79
数据服务	43	DB2 教程	81
IBM OLE DB Provider 所支持的游标方式	43	DB2 故障诊断信息	81
DB2 与 OLE DB 之间的数据类型映射	43	信息中心条款和条件	82
用于设置将数据从 OLE DB 类型转换为 DB2 类型的数据库转换	44	附录 B. 声明	83
用于将数据从 DB2 类型转换为 OLE DB 类型的数据库转换	47	索引	87
IBM OLE DB Provider 限制	50		
IBM OLE DB Provider 对 OLE DB 组件和接口的支持	51		
IBM OLE DB Provider 对 OLE DB 属性的支持	54		

第 1 章 ADO.NET 应用程序开发

下面主要说明 .NET Framework 的关键功能以及 .NET Framework 如何替换 COM 技术。

最近几年，Microsoft 一直致力推广一款新的 Windows 软件开发平台，即 .NET Framework。 .NET Framework 是 Microsoft 对组件对象模型 (COM) 技术的替代。 .NET Framework 的关键功能如下：

- 您可以使用 40 种以上不同的编程语言来编码 .NET 应用程序。进行 .NET 开发时，最流行的语言是 C# 和 Visual Basic .NET。
- .NET Framework 类库提供了用于构建 .NET 应用程序的构建块。这个类库与语言无关，它提供操作系统和应用程序服务的接口。
- 无论使用哪种语言，.NET 应用程序都将编译成中间语言 (IL)，这是一种字节码。
- 公共语言运行时 (CLR) 是 .NET Framework 的核心，它能够即时编译 IL 代码并接着运行该代码。在运行经过编译的 IL 代码时，CLR 将激活对象、验证其是否存在安全性问题、为其分配内存、执行这些对象并在执行完成后清除其内存。

借助这些功能，.NET Framework 有利于各种应用程序实现（例如 Windows 表单、Web 表单和 Web Service）、快速应用程序开发以及安全应用程序部署。COM 和 COM+ 已被证实无法实现或难以实现上述所有功能。

.NET Framework 通过 ADO.NET 提供了广泛的数据访问支持。ADO.NET 既支持连接式访问也支持断开连接式访问。在 ADO.NET 中，断开连接式数据访问的关键组件是 DataSet 类，其实例充当驻留在应用程序内存中的数据库高速缓存。

进行连接式访问和断开连接式访问时，应用程序都通过数据提供程序来使用数据库。各种数据库产品都包括它们自己的 .NET 数据提供程序，其中包括 DB2[®] for Windows。

.NET 数据提供程序提供了下列基本类的实现：

- Connection: 建立并管理数据库连接。
- Command: 对数据库执行 SQL 语句。
- DataReader: 从数据库读取结果集数据并返回该数据。
- DataAdapter: 将 DataSet 实例与数据库链接。通过 DataAdapter 实例，DataSet 可以读写数据库表数据。

Microsoft 提供了两个数据提供程序，即：OLE DB .NET Data Provider 和 ODBC .NET Data Provider。OLE DB .NET Data Provider 是通过 COM 互操作模块向 IBM[®] OLE DB Provider 反馈 ADO.NET 请求的桥接提供程序。ODBC .NET Data Provider 是向 IBM ODBC Driver 反馈 ADO.NET 请求的桥接提供程序。建议您不要使用这些 .NET 数据提供程序来访问 DB2 系列数据库。IBM Data Server Provider for .NET 是高性能的受管 ADO.NET 数据提供程序。建议您将此 .NET 数据提供程序与 DB2 系列数据库配合使用。使用 IBM Data Server Provider for .NET 进行 ADO.NET 数据库访问时，限制较少，并且性能显著优于 OLE DB 和 ODBC .NET 桥接提供程序。

部署 .NET 应用程序 (Windows)

为了简化 .NET 应用程序部署工作，IBM 提供了 IBM Data Server Driver Package，这是一个适用于大规模部署方案的小规模客户机。

如果相对于 IBM Data Server Driver Package 您更需要使用 IBM Data Server Runtime Client 的附加功能，那么可以改为使用该客户机。

开始之前

- 在执行部署之前，必须通过 Visual Studio 或命令行来构建 .NET 应用程序。有关构建 .NET 应用程序的更多信息，请参阅相关任务。
- 用于构建 .NET 应用程序的计算机以及将要部署 .NET 应用程序的计算机除了运行『受支持的 .NET 开发软件』所述的其他软件以外，还必须运行受支持的 Windows 操作系统版本：
 - 构建系统
 - Windows 操作系统
 - Visual Studio
 - .NET Framework Redistributable Package
 - .NET Framework Software Development Kit
 - 部署系统
 - Windows 操作系统
 - .NET Framework Redistributable Package

过程

要部署 .NET 应用程序，请执行下列操作：

1. 在将要部署应用程序的计算机上安装 IBM Data Server Driver Package。在安装期间，请将 IBM Data Server Driver Package 安装版本设置为缺省数据库客户机接口副本。

注：任何面向 IBM 数据服务器运行的现有数据库应用程序都将使用这个新的 IBM Data Server Driver Package 安装版本。在将所部署的 .NET 应用程序投入使用之前，请针对新驱动程序来测试那些应用程序。

2. 在将要运行应用程序的计算机上安装所构建的应用程序。

受支持的 .NET 开发软件

要开发和部署面向 IBM 数据服务器运行的 .NET 应用程序，您必须使用受支持的开发软件和操作系统。

开发和部署 .NET Framework 2.0、3.0、3.5 和 4.0 应用程序时支持的操作系统

开发和部署 .NET Framework 2.0、3.0、3.5 和 4.0 应用程序时支持以下操作系统：

- Windows XP Service Pack 2 (32 位和 64 位版本)
- Windows Server 2003 (32 位和 64 位版本)
- Windows Vista (32 位和 64 位版本)
- Windows Server 2008 (32 位和 64 位版本)

- Windows Server 2008 R2 (64 位版本)
- Windows 7 (32 位和 64 位版本)

开发和部署 .NET Framework 4.5 应用程序时支持的操作系统

开发和部署 .NET Framework 4.5 应用程序时支持以下操作系统:

- Windows Vista SP2 (32 位和 64 位版本)
- Windows Server 2008 SP2 (32 位和 64 位版本)
- Windows Server 2008 R2 (64 位版本)
- Windows 7 (32 位和 64 位版本)

Microsoft .NET Framework 4.5 受 V10.1 FP2 及更高级别修订包支持。

支持 .NET Framework 应用程序的开发软件

除 IBM 数据服务器客户机或驱动程序包之外, 您还需要其中一个受支持工具来开发 .NET Framework 应用程序:

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

Microsoft Visual Studio 2012 受 V10.1 FP2 及更高级别修订包支持。

支持 .NET Framework 应用程序的部署软件

除 IBM 数据服务器客户机或驱动程序包之外, 您还需要下列其中一个程序包来部署 .NET Framework 应用程序。在大多数情况下, Redistributable Package 随 Windows 安装一起提供。

- .NET Framework V2.0 Redistributable Package
- .NET Framework V3.0 Redistributable Package
- .NET Framework V3.5 Redistributable Package
- .NET Framework V4.0 Redistributable Package
- .NET Framework V4.5 Redistributable Package

安装 64 位 IBM Data Server Package 时, 会同时安装和配置 32 位和 64 位提供程序。

如果未安装 .NET Framework, 那么 IBM 数据服务器客户机和驱动程序安装程序将不会安装 IBM Data Server Provider for .NET。您必须以手动方式安装 IBM Data Server Provider for .NET。

在 Visual Studio 中集成 DB2

IBM Database Add-Ins for Visual Studio 是一组功能部件, 这些功能部件无缝地集成到 Visual Studio 开发环境中, 以使您能够使用 DB2 服务器以及开发 DB2 过程、函数和对象。

IBM Database Add-Ins for Visual Studio 针对 DB2 数据库提供一个简单接口。例如, 可以使用设计器和向导来创建数据库对象, 而不必使用 SQL。对于您需要编写 SQL 代码的情况, 集成式 DB2 SQL 编辑器提供了下列功能:

- 彩色 SQL 文本，旨在提高可读性
- 与 Microsoft Visual Studio IntelliSense 功能部件集成，这在您输入 DB2 脚本时支持智能自动补全

使用 IBM Database Add-Ins for Visual Studio，您可以完成下列任务：

- 打开各种 DB2 开发和管理工具。
- 在解决方案资源管理器中创建和管理 DB2 项目。
- 通过服务器资源管理器访问和管理 DB2 数据连接。
- 创建和修改 DB2 脚本，其中包括用于创建存储过程、函数、表、视图、索引和触发器的脚本。

Visual Studio 2008 和 2010

IBM Database Add-Ins for Visual Studio 作为可单独安装的组件随 DB2 客户机和 DB2 服务器一起提供。安装 DB2 产品完成后，您可以选择安装 IBM Database Add-Ins for Visual Studio。如果计算机上未安装 Visual Studio，那么将不会安装此加载件。在安装 Visual Studio 之后，您随时可以从 DB2 产品的设置菜单中安装此加载件。

有关使用 IBM Database Add-Ins 和 Data Server Provider for .NET 进行快速应用程序开发的更多详细信息，请访问 IBM Information Management and Visual Studio .NET 专区，网址为：<http://www.ibm.com/developerworks/data/zones/vstudio/index.html>。

IBM Data Server Provider for .NET 概述

IBM Data Server Provider for .NET 扩展了对 ADO.NET 接口的数据服务器支持。此提供程序使您能够安全而高性能地访问 IBM 数据服务器。

IBM Data Server Provider for .NET 客户机程序包中包括两个提供程序。这些提供程序有时又称为公共 .NET 提供程序。

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

通过 DB2 .NET Provider，.NET 应用程序可访问以下数据库管理系统：

- DB2 for Linux, UNIX, and Windows V9.1、V9.5、V9.7、V9.8 和 V10.1
- DB2 Universal Database™ V8 for Windows, UNIX, and Linux
- DB2 for z/OS® V8、V9 和 V10，通过 DB2 Connect™
- IBM DB2 for IBM i V5R4、V6R1 和 V7R1，通过 DB2 Connect（对于 IBM DB2 V9.7 FP4 及更高版本）
- IBM DB2 for IBM i V5R4 和 V6R1，通过 DB2 Connect（对于 IBM DB2 V9.7 FP3 及更低版本）
- IBM Informix® V11.10 及更高版本

余下主题讨论了公共 DB2 .NET Provider。

Informix 数据库服务器 .NET Provider (IBM.Data.Informix.dll)

通过 Informix 数据库服务器 .NET Provider，.NET 应用程序可访问以下数据库管理系统：

- IBM Informix V11.10 及更高版本

。有关此提供程序的更多信息，请参阅 IBM Informix Dynamic Server 信息中心。

要开发和运行使用 Data Server Provider for .NET 的应用程序，您需要 .NET Framework。

除 IBM Data Server Provider for .NET 之外，IBM Database Add-Ins for Visual Studio 还允许您使用 Microsoft Visual Studio 快速轻松地开发 IBM 数据库服务器 .NET 应用程序。还可以使用该加载件来创建数据库对象（例如，索引和表）以及开发服务器端对象（例如，存储过程和用户定义的函数）。

IBM Data Server Provider for .NET 数据库系统要求

应用程序可以使用 IBM Data Server Provider for .NET 来访问 IBM 数据库系统。

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

通过 DB2 .NET Provider，.NET 应用程序可访问以下数据库管理系统：

- DB2 for Linux, UNIX, and Windows V9.1、V9.5、V9.7、V9.8 和 V10.1
- DB2 Universal Database V8 for Windows, UNIX, and Linux
- DB2 for z/OS V8、V9 和 V10，通过 DB2 Connect
- IBM DB2 for IBM i V5R4、V6R1 和 V7R1，通过 DB2 Connect（对于 IBM DB2 V9.7 FP4 及更高版本）
- IBM DB2 for IBM i V5R4 和 V6R1，通过 DB2 Connect（对于 IBM DB2 V9.7 FP3 及更低版本）
- IBM Informix V11.10 及更高版本

余下主题讨论了公共 DB2 .NET Provider。

Informix 数据库服务器 .NET Provider (IBM.Data.Informix.dll)

通过 Informix 数据库服务器 .NET Provider，.NET 应用程序可访问以下数据库管理系统：

- IBM Informix V11.10 及更高版本

。有关此提供程序的更多信息，请参阅 IBM Informix Dynamic Server 信息中心。

对于 DB2 for i，需要在服务器上安装以下修订：APAR III3348。

编写应用程序以使用 IBM Data Server Provider for .NET

编写应用程序以使用 IBM Data Server Provider for .NET 要求了解 IBM Data Provider for .NET 中提供的功能。在确定您的应用程序的需求之后，您可以利用 IBM Data Provider for .NET 中所提供的功能。

使用 ADO.NET 公共基类进行通用编码

.NET Framework V2.0、V3.0 和 V3.5 提供了名为 System.Data.Common 的名称空间，该名称空间提供了一组可以由任何 .NET 数据提供程序共享的基类。这会让通用 ADO.NET 数据库应用程序开发方法（针对不同数据库间提供一致的编程接口）顺利进行。

以下 C# 代码演示建立数据库连接的通用方法。

```
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 对象是任何通用 ADO.NET 应用程序的开始点。此对象创建 .NET 数据提供程序对象（例如连接、数据适配器、命令和数据阅读器）的通用实例，这些实例与特定的数据库产品配合工作。在以上示例中，传递至 GetFactory 方法的 "IBM.Data.DB2" 字符串唯一标识 IBM Data Server Provider for .NET，并且导致 DbProviderFactory 实例初始化，该实例创建特定于 IBM Data Server Provider for .NET 的数据库提供程序对象实例。

DbConnection 对象可连接至 DB2 系列 Informix 数据库（就像 DB2Connection 对象，它实际上是从 DbConnection 继承的）。通过使用 DbConnectionStringBuilder 类，您可以确定数据提供程序的连接字符串关键字并生成定制连接字符串。以上示例中的代码检查 IBM Data Server Provider for .NET 中是否存在名为 "database" 的关键字，如果存在，那么会生成连接字符串以连接至 SAMPLE 数据库。

使用 IBM Data Server Provider for .NET 来从应用程序连接至数据库

使用 IBM Data Server Provider for .NET 时，会通过 DB2Connection 类建立数据库连接。

过程

要连接至数据库，请执行以下操作：

1. 创建用于存储连接参数的字符串。典型连接字符串的格式为：

```

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

```

可能的连接字符串的示例：

示例 1：

```

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

```

注：如果仅在连接字符串中指定数据库名称，那么必须将其他信息（例如，服务器、用户标识和密码）包括在 db2dsdriver.cfg 文件中。

示例 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. 将 connectString 传递至 DB2Connection 构造函数。

- 使用 C# 连接到数据库：

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

- 在 Visual Basic .NET 中连接到数据库:

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

3. 使用 DB2Connection 对象的 Open 方法正式连接至 connectionString 中标识的数据库。

使用 IBM Data Server Provider for .NET 进行连接合用

第一次对 DB2 数据库打开连接时，会创建连接池。关闭连接时，连接会进入该池，准备供需要连接的其他应用程序在同一进程内重复使用。

IBM Data Server Provider for .NET 使用一组规范化连接字符串属性来确定连接池。通过使用规范化属性，应用程序重复使用连接的机会将增加。

缺省情况下，IBM Data Server Provider for .NET 将启用连接合用。

注：您可以使用 Pooling=false 连接字符串关键字/值来关闭连接合用。但是，如果关闭连接合用，那么 COM+ 应用程序将不工作。

通过设置下列连接字符串关键字，您可以控制连接池的行为：

- 最小和最大池大小 (**MinPoolSize** 和 **MaxPoolSize**)
- 连接返回至池之前可保持空闲状态的时长 (**ConnectionLifetimeInPool**)

通过 IBM Data Server Provider for .NET 创建可信连接

.NET 通过连接字符串关键字来支持可信上下文。

在连接字符串中，可用的关键字如下所示：

- **TrustedContextSystemUserID** 或 **tcsuid**，此关键字指定要与连接配合使用的可信上下文 **SYSTEM AUTHID**。
- **TrustedContextSystemPassword** 或 **tcspwd**，此关键字指定与连接配合使用的可信上下文 **SYSTEM AUTHID** 的相应密码。

如果在未指定 **TrustedContextSystemUserID** 关键字值的情况下指定 **TrustedContextSystemPassword** 关键字，那么将抛出 **InvalidArgument** 异常。在可信上下文方案中，**UserID** 关键字也是必需的。

目前以下版本支持通过 IBM Data Server Provider for .NET 创建可信上下文：

- DB2 Database for Linux, UNIX, and Windows V9.5、V9.7、V9.8 和 V10.1
- DB2 Universal Database V9 and V10 for z/OS

示例

假定已在服务器上使用以下信息建立可信上下文：

```
CREATE TRUSTED CONTEXT ctxName1
BASED UPON CONNECTION USING SYSTEM AUTHID masteruser
ATTRIBUTES ( PROTOCOL 'TCP/IP',
```

```
ADDRESS '9.26.146.201',  
ENCRYPTION 'NONE' )  
ENABLEWITH USE FOR userapp1 WITH AUTHENTICATION, userapp2 WITH AUTHENTICATION;
```

SYSTEM AUTHID 为 masteruser, 其相应密码为 masterpassword。每个特定的用户/应用程序 (userapp1 和 userapp2) 都有相应的密码 (passapp1 和 passapp2)。

要使用此可信上下文, 应用程序将发出连接字符串, 如下所示:

- 应用程序 1

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

- 应用程序 2

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

注: 在可信上下文环境中, UserID 关键字与连接的最终用户相对应, 这与标准应用程序的情况相同。

按照 .NET 程序打开和关闭某个连接:

```
[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();
```

如果可信上下文处理由于您尚未在服务器上设置任何可信上下文而失败, 或者服务器不支持可信上下文, 那么将抛出 SQLCODE 为 CLI0197E 的错误。如果 TrustedContextSystemUserID 关键字值无效 (例如太长), 那么将抛出 SQLCODE 为 CLI0124E 的错误。服务器可能会报告 SQLCODE 为 SQL1046N、SQL30082N 或 SQL0969N 的错误, 并且本机错误代码为 -20361。任何这些错误都将导致 Open() 失败。

注: 可信上下文处理将在您下次与服务器进行通信时进行。

ADO.NET 数据库应用程序中的 SQL 数据类型表示

ADO.NET 数据库应用程序可引用 DB2 SQL 数据类型值作为参数值（用作所执行的 SQL 语句的组成部分）和变量，但是，必须使用适当的 IBM Data Server Provider for .NET 数据类型值和 .NET Framework 数据类型值以确保访问或检索值时不会发生数据截断或数据丢失。

必须使用 IBM Data Server Provider for .NET 对象，才能指定要用作所要执行的 SQL 语句一部分的参数值。DB2Parameter 对象用于表示要添加至 DB2Command 对象（它表示 SQL 语句）的参数。指定参数的数据类型值时，必须使用 IBM.Data.DB2Types 名称空间中提供的 IBM Data Server Provider for .NET 数据类型值。IBM.Data.DB2Types 名称空间提供类和结构以表示每个受支持的 DB2 SQL 数据类型。

对于可能临时保存 SQL 数据类型值的局部变量，必须使用适当 IBM Data Server Provider for .NET 数据类型（如 IBM.Data.DB2Types 名称空间中所定义）。

下表显示 DB2Type 数据类型、DB2 数据类型、Informix 数据类型、Microsoft .NET Framework 类型以及 DB2Types 类和结构之间的映射。

类别	DB2Types 类和结构	DB2Type 数据类型	DB2 数据类型	Informix 数据类型	.NET 数据类型
二进制数据	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[]
字符数据	DB2String	Char	CHAR	CHAR	String
	DB2String	VarChar	VARCHAR	VARCHAR	String
	DB2String	LongVarChar ¹	LONG VARCHAR	LVARCHAR	String
图形数据	DB2String	Graphic	GRAPHIC		String
	DB2String	VarGraphic	VARGRAPHIC		String
	DB2String	LongVarGraphic ¹	LONG VARGRAPHIC		String
LOB 数据	DB2Clob	Clob	CLOB	CLOB 和 TEXT	String
	DB2Blob	Blob	BLOB	BLOB 和 BYTE	Byte[]
	DB2Clob	DbClob	DBCLOB		String

1. DB2 .NET 公共语言运行时例程不支持将这些数据类型用作参数。
2. 类型为 DB2Type.Xml 的 DB2ParameterClass.ParameterName 属性可以接受下列类型的变量：String、byte[]、DB2Xml 以及 XmlReader。
3. 这些数据类型只适用于 DB2 for z/OS 和 DB2 for i V6R1 及更高版本。
4. 仅 DB2 for z/OS V9 及更高发行版以及 DB2 for Linux, UNIX and Windows V9.5 及更高发行版支持此数据类型。
5. Date 对象和 Time 对象可以是时间戳记字符串文字。Timestamp 对象可以是日期字符串文字

数字数据	DB2Int16	SmallInt	SMALLINT	BOOLEAN 和 SMALLINT	Int16
	DB2Int32	Integer	INT	INTEGER、INT 和 SERIAL	Int32
	DB2Int64	BigInt 和 BigSerial	BIGINT	BIGINT、BIGSERIAL、INT8 和 SERIAL8	Int64
	DB2Real 和 DB2Real370	Real	REAL	REAL 和 SMALLFLOAT	Single
	DB2Double	Double	DOUBLE PRECISION	DECIMAL (≤ 29) 和 DOUBLE PRECISION	Double
	DB2Double	Float	FLOAT	DECIMAL (32) 和 FLOAT	Double
	DB2Decimal	Decimal	DECIMAL	MONEY	Decimal
	DB2DecimalFloat	DecimalFloat	DECFLOAT(16 34) ¹⁴		Decimal
	DB2Decimal	Numeric	DECIMAL	DECIMAL (≤ 29) 和 NUMERIC	Decimal
日期/时间数据	DB2Date	Date	DATE	DATETIME (日期精度)	DateTime String ⁵
	DB2Time	Time	TIME	DATETIME (时间精度)	TimeSpan String ⁵
	DB2TimeStamp	Timestamp	TIMESTAMP	DATETIME (时间和日期精度)	DateTime String ⁵
	DB2TimeStamp Offset	TimestampWith TimeZone	TIMESTAMP WITH TIME ZONE	不适用	DateTimeOffset String ⁵
行标识数据	DB2RowId	RowId	ROWID		Byte[]
XML 数据	DB2Xml	Xml ²	XML		Byte[]

使用 IBM Data Server Provider for .NET 从应用程序中执行 SQL 语句

使用 IBM Data Server Provider for .NET 时，会通过 DB2Command 类（使用其方法 ExecuteReader() 和 ExecuteNonQuery() 及其属性 CommandText、CommandType 和 Transaction）执行 SQL 语句。

关于此任务

对于生成输出的 SQL 语句，可以使用 ExecuteReader() 方法，并且从 DB2Data Reader 对象检索结果。对于所有其他 SQL 语句，可以使用 ExecuteNonQuery() 方法。应将 DB2Command 对象的 Transaction 属性初始化为 DB2Transaction 对象。DB2Transaction 对象负责回滚和落实数据库事务。

使用 C# 执行 UPDATE 语句:

```
// 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();
```


在 Visual Basic .NET 中执行 UPDATE 语句:

```
' 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();
```

使用 C# 执行 SELECT 语句:

```
// 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();
```

在 Visual Basic .NET 中执行 SELECT 语句:

```
' 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()
```

应用程序执行数据库事务后, 必须使其回滚或落实。这是通过 DB2Transaction 对象的 Commit() 和 Rollback() 方法完成的。

使用 C# 回滚或落实事务:

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

在 Visual Basic .NET 中回滚或落实事务:

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

在 V9.7 FP2 及更高级别修订包中, DB2 for Linux, UNIX, and Windows 服务器支持通过执行匿名块返回结果集。在 V9.7 FP6 及更高级别修订包中, .NET 数据提供程序支持应用程序使用 DB2DataReader 或 DB2ResultSet 类来检索通过执行匿名块返回的结果集。 .NET 数据提供程序要检索通过执行匿名块返回的结果集, 数据库服务器必须支持 PL/SQL 语句, 并且必须允许数据库处理 PL/SQL 语句。 .NET 数据提供程序必须使用 BEGIN 语句 (而不是 BEGIN COMPOUND 语句) 针对通过执行匿名块返回的结果集声明游标。

使用 C# 时, 借助 DB2DataReader 类检索通过执行匿名块检索单个结果集:

```
...
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();
...
```

使用 C# 时, 借助 DB2ResultSet 类检索通过执行匿名块检索单个结果集:

```
...
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();
...
```

使用 C# 时, 借助 DB2DataReader 类检索通过执行匿名块检索多个结果集:

```
...
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();
...
```

使用 C# 时, 借助 DB2ResultSet 类检索通过执行匿名块检索多个结果集:

```
...
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();
...
```

使用 IBM Data Server Provider for .NET 从应用程序中读取结果集

使用 IBM Data Server Provider for .NET 时，会通过 DB2DataReader 对象读取结果集。DB2DataReader 方法 Read() 用来进至结果集中的下一行。

关于此任务

方法 GetString()、GetInt32()、GetDecimal() 和用于所有可用数据类型的其他方法可用于从输出的各列抽取数据。DB2DataReader 的 Close() 方法用于关闭 DB2DataReader 对象，后者应始终在读取完输出时完成。

使用 C# 读取结果集:

```
// 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();
```

使用 Visual Basic .NET 读取结果集:

```
' 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();
```

使用 IBM Data Server Provider for .NET 从应用程序中调用存储过程

使用 IBM Data Server Provider for .NET 时，可使用 DB2Command 对象来调用存储过程。

关于此任务

CommandType 属性的缺省值是 CommandType.Text。此值适用于 SQL 语句，并且还用于调用存储过程。但是，将 CommandType 设置为 CommandType.StoredProcedure 时，可以更方便地调用存储过程。在这种情况下，您只需要指定存储过程名以及任何参数。

使用存储过程时，可使用主变量、命名参数或定位参数来传递参数。但是，它们不能在同一 SQL 语句中组合使用。

以下 C# 和 Visual Basic 示例演示如何在将 CommandType 属性设置为 CommandType.StoredProcedure 或 CommandType.Text 的情况下调用名为 INOUT_PARAM 的存储过程。

过程

- 使用 C# 时, 通过将 DB2Command 的 CommandType 属性设置为 CommandType.Text 来调用存储过程:

```
// 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();
```

- 使用 Visual Basic 时, 通过将 DB2Command 的 CommandType 属性设置为 CommandType.Text 来调用存储过程:

```
' 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 和 EXECUTE PROCEDURE 都受支持。

- 使用 C# 时, 通过将 DB2Command 的 CommandType 属性设置为 CommandType.StoredProcedure 来调用存储过程。使用此方法时不支持使用命名参数:

```
// 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();
```

- 使用 Visual Basic 时, 通过将 DB2Command 的 CommandType 属性设置为 CommandType.StoredProcedure 来调用存储过程:

```

' 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()

```

- 通过参数名进行限定时，可按任何顺序将参数传递至存储过程。仅 DB2 for Linux, UNIX and Windows 数据服务器支持此命名参数功能。例如，在以下 SQL 语句中，定义了存储过程，然后带按另一顺序排列的参数调用了该存储过程：

```

CREATE PROCEDURE schema.my_proc ( IN var1 int, INOUT var2 int )
LANGUAGE SQL
BEGIN
-- procedure code here
END

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

```

同时访问 CURSOR 类型输出参数返回的结果集

使用 IBM Data Server Provider for .NET 时，DB2Type.Cursor 指定为同时访问输出参数中的所有游标。

关于此任务

对于具有多个 CURSOR 类型输出参数的存储过程，将 DB2TYPE.Cursor 绑定至输出参数对象允许同时访问输出参数中的所有游标。

例如，OrderDetails 存储过程声明三个游标，每个游标给出有关产品及其销售的相关信息。

```

CREATE OR REPLACE TYPE cur AS CURSOR
CREATE PROCEDURE OrderDetails (p_startDate TIMESTAMP, p_endDate TIMESTAMP,
OUT prodDetails 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;

```

调用程序需要同时访问这些游标，以便它可通过每个游标收集特定产品的相关信息并计算折扣。为提供对游标的同时访问，存储过程将这些游标作为输出参数返回。绑定 CURSOR 类型输出参数时，该应用程序必须将 DB2Type 设置为 DB2Type.Cursor 才能进行同时访问。

```

//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", DbType.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();

```

只能在调用 `ExecuteNonQuery` 方法后通过 `Value` 属性访问游标类型输出参数中的数据阅读器。

如果使用 `ExecuteReader` 或 `ExecuteResultSet` 方法执行该命令，那么会在 `DB2DataReader` 或 `DB2ResultSet` 对象中返回结果集。后续结果集必须通过调用 `NextResult` 方法顺序访问。尽管已绑定输出参数，但访问输出参数 `Value` 属性还是会导致 `InvalidOperationException` 异常，因为未使用 `ExecuteNonQuery` 方法执行该查询。

同步使用游标时，该应用程序可能需要先落实工作，然后继续读取游标。应用程序要发出落实命令而不损坏已打开的游标，该游标必须在存储过程内声明为可挂起。

使用 **System.Diagnostics.Trace** 跟踪 IBM Data Server Provider for .NET

本主题讨论使用 `System.Diagnostics.Trace` 跟踪 IBM Data Server Provider for .NET。

关于此任务

将使用 `System.Diagnostics.Trace` 来跟踪 .NET 公共 API 中的函数入口和出口跟踪点。将仅跟踪第一级调用，例如将仅跟踪客户的应用程序显式进行的提供程序调用。将不跟踪提供程序自身在内部对公共 API 进行的任何调用。

对于 `IBM.Data.DB2` 名称空间，`TraceSwitch` 将命名为 `DB2NMPTRACE`，而对于 `IBM.Data.Informix` 名称空间，`TraceSwitch` 将命名为 `IFXNMPTRACE`。

```

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

```

`TraceSwitch` 类使用 `System.Diagnostics.TraceLevel` 枚举来控制跟踪的信息级别 - 0 到 4 分别表示“关闭”、“错误”、“警告”、“信息”和“详细”。对于 .NET 提供程序，0 将表示 Off，而 1 到 4 将表示 On。

示例

一个简单程序的跟踪输出可能类似于如下所示:

注: 由应用程序自身转储的跟踪点带有 * 前缀。

```
* * 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
```

使用 pureQuery 来优化 .NET 应用程序中的查询

.NET 客户机驱动程序可以利用 pureQuery 技术中的功能。这些功能使现有的 .NET 应用程序查询能够作为静态 SQL 执行。静态查询可以避免需要在运行时准备某些语句。这有助于提高应用程序的安全性和性能。

有关更多信息, 请参阅<http://www.ibm.com/support/docview.wss?uid=swg27023946>

IBM Data Server Provider for .NET 对 Microsoft Entity Framework 的支持

通过 IBM Data Server Provider for .NET, 使用 IBM 数据服务器来利用 Microsoft ADO.NET Entity Framework。可使用受支持服务器版本来生成 EDM 模式、对实体应用程序写入和执行实体 SQL 和 LINQ 语句。

系统要求

IBM Data Server Provider for .NET 使用以下 IBM 数据服务器:

- IBM DB2 for Linux, UNIX, and Windows 的所有受支持级别
- DB2 for i V5R4、V6R1 和 V7R1, 通过 DB2 Connect (对于 IBM DB2 V9.7 FP4 及更高版本)
- DB2 for z/OS V8 或更高版本
- IBM Informix V11.170 或更高版本

必须具有带 Microsoft ADO.NET Entity Framework 的 Microsoft .NET Framework 3.5 SP1。要使用 Microsoft Entity Data Model 向导或 ADO.NET Entity Designer 来处理实体数据模型, 还需要 Microsoft Visual Studio 2008 或更高版本。

下表列示 IBM 实体提供程序支持的规范函数。该数据提供程序会将规范函数转换为对应数据源函数。

表 1. IBM 实体提供程序对规范函数的支持

规范函数类型	LIHQ 函数	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
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
	替换	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

表 1. IBM 实体提供程序对规范函数的支持 (续)

规范函数类型	LIHQ 函数	DB2 for Linux, UNIX, and Windows	DB2 for z/OS	DB2 for i	Informix
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

已知限制

要点: 某些规范函数依赖服务器。错误 SQL0440N* 指示您的服务器不支持某些功能。要获取有关受支持功能的更多信息，请与 IBM 服务器技术支持人员联系。

IBM Data Server Provider for .NET 对 Microsoft ADO.NET Entity Framework 的支持具有以下限制:

通用:

- 仅支持数据库的第一个方案; 使用 Entity Framework 之前, 所有数据库对象必须已存在。
- 不支持调用特定于存储的函数。
- 服务器资源管理器的“添加连接”对话框中设置的可信上下文连接属性不会传递至 Entity Framework 连接。

DB2 for z/OS:

- 数据类型 REAL 不受支持。应用程序必须在表的模式中使用 FLOAT 数据类型，或在客户机模式 (EDM) 中将类型指定为 FLOAT，即使服务器上的实际类型为 REAL 也是如此。
- 特定于 V8 和 V7: 可能对包括 Take、Top、First、Intersect 或 Except 表达式的查询生成异常，用于指示语法错误。未定义包括这些表达式的查询的结果。

示例:

```
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);
```

使用 Enterprise Library 数据访问模块

Enterprise Library 是应用程序块的集合，用于帮助开发者面对常见的开发挑战。应用程序块作为源代码提供，可以在开发项目按原样使用，也可以针对开发项目进行修改。

您可以从 <http://codeplex.com/entlibcontrib/SourceControl/PatchList.aspx> 获取用于 IBM 数据服务器的 Enterprise Library 数据访问模块以及其他模块。

有关如何安装 Enterprise Library 数据访问模块并将其与 IBM 数据服务器 (DB2、Informix 数据库服务器 和 U2) 配合使用的信息，请参阅下载程序包中的自述文件。

资源

下面是多个描述了如何使用数据访问模块的在线资源:

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

构建 .NET 应用程序

用于构建 .NET 应用程序的资源

构建 Visual Basic .NET 应用程序

DB2 产品提供了用于编译和链接 DB2 Visual Basic .NET 应用程序的 bldapp.bat 批处理文件。

此文件与可以使用此文件构建的样本程序一起放在 sqllib\samples\.NET\vb 目录中。该批处理文件采用一个参数（即 %1）来表示要编译的源文件的名称（不带 .vb 扩展名）。

关于此任务

本任务指导您完成使用 bldapp.bat 借助 DbAuth 样本文件构建 Visual Basic .NET 应用程序的基本步骤。

过程

要根据源文件 DbAuth.vb 来构建程序 DbAuth，请输入以下命令：

```
bldapp DbAuth
```

要确保您有运行可执行文件时所需的参数，可以根据输入的参数数目指定不同的参数组合：

1. 无参数。请仅输入程序名：

```
DbAuth
```

2. 1 个参数。请输入程序名和数据库别名：

```
DbAuth <db_alias>
```

3. 2 个参数。请输入程序名以及用户标识和密码：

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

4. 3 个参数。请输入程序名以及数据库别名、用户标识和密码：

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

5. 4 个参数。请输入程序名以及服务器名称、端口号、用户标识和密码：

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

6. 5 个参数。请输入程序名以及数据库别名、服务器名称、端口号、用户标识和密码：

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

下一步做什么

要构建和运行 LCTrans 样本程序，您需要遵循源文件 LCTrans.vb 中提供的更详细指示信息。

构建 C# .NET 应用程序

DB2 产品提供了用于编译和链接 DB2 C# .NET 应用程序的 bldapp.bat 批处理文件。此批处理文件与可以使用此文件构建的样本程序一起放在 sqllib\samples\.NET\cs 目录中。

该批处理文件采用一个参数（即 %1）来表示要编译的源文件的名称（不带 .cs 扩展名）。

关于此任务

本任务指导您完成使用 bldapp.bat 借助 DbAuth 样本文件构建 C# .NET 应用程序的基本步骤。

过程

要根据源文件 DbAuth.cs 来构建程序 DbAuth, 请输入以下命令:

```
bldapp DbAuth
```

要确保您有运行可执行文件时所需的参数, 可以根据输入的参数数目指定不同的参数组合:

1. 无参数。请仅输入程序名:

```
DbAuth
```

2. 1 个参数。请输入程序名和数据库别名:

```
DbAuth <db_alias>
```

3. 2 个参数。请输入程序名以及用户标识和密码:

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

4. 3 个参数。请输入程序名以及数据库别名、用户标识和密码:

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

5. 4 个参数。请输入程序名以及服务器名称、端口号、用户标识和密码:

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

6. 5 个参数。请输入程序名以及数据库别名、服务器名称、端口号、用户标识和密码:

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

下一步做什么

要构建和运行 LCTrans 样本程序, 您需要遵循源文件 LCTrans.cs 中提供的更详细指示信息。

Visual Basic .NET 应用程序的编译和链接选项

本主题描述编译和链接 Visual Basic .NET 应用程序时的各个可用选项。

在 Windows 上使用 Microsoft Visual Basic .NET 编译器构建 Visual Basic .NET 应用程序时, 使用下列编译和链接选项 (如在 bldapp.bat 批处理文件中所述)。

注: .NET Framework V1.1 仅在与 .NET Provider V9.5 及更低版本配合使用时受支持。

使用 bldapp 的独立 VB .NET 应用程序的编译和链接选项

独立 VB .NET 应用程序的编译和链接选项:

```
%BLDCOMP%
```

编译器的变量。缺省值为 vbc, 即 Microsoft Visual Basic .NET 编译器。

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

引用您正在使用的 .NET Framework 版本的 DB2 动态链接库。

```
%DB2PATH%
```

%DB2PATH% 变量表示 DB2 产品安装的根路径。%DB2PATH% 变量未出现在 IBM Data Server Driver for ODBC and CLI 或 Data Server Driver Package 安装上。使用 IBM Data Server Driver for ODBC and CLI 或 Data Server Driver Package 时, 将 %DB2PATH% 替换为安装了驱动程序产品的路径。

%VERSION%

应用程序支持多个版本的 .NET Framework。对于每个版本，DB2 都有一个动态链接库。对于 .NET Framework V2.0、3.0 和 3.5，%VERSION% 指向 netf20\ 子目录。

使用 **ldapp** 的松散耦合样本程序 LCTrans 的编译和链接选项:

%BLDCOMP%

编译器的变量。缺省值为 vbc，即 Microsoft Visual Basic .NET 编译器。

/out:RootCOM.dll

根据 RootCOM.vb 源文件输出 LCTrans 应用程序所使用的 RootCOM 动态链接库。

/out:SubCOM.dll

根据 SubCOM.vb 源文件输出 LCTrans 应用程序所使用的 SubCOM 动态链接库。

/target:library %1.cs

根据输入源文件 (RootCOM.vb 或 SubCOM.vb) 来创建动态链接库。

/r:System.EnterpriseServices.dll

引用 Microsoft Windows 系统企业服务数据链路库。

/r:"%DB2PATH%\bin\%VERSION%\IBM.Data.DB2.dll

引用您正在使用的 .NET Framework 版本的 DB2 动态链接库。

%DB2PATH%

%DB2PATH% 变量表示 DB2 产品安装的根路径。%DB2PATH% 变量未出现在 IBM Data Server Driver for ODBC and CLI 或 Data Server Driver Package 安装上。使用 IBM Data Server Driver for ODBC and CLI 或 Data Server Driver Package 时，将 %DB2PATH% 替换为安装了驱动程序产品的路径。

%VERSION%

应用程序支持多个版本的 .NET Framework。对于每个版本，DB2 都有一个动态链接库。对于 .NET Framework V2.0 和 3.0，%VERSION% 指向 netf20\ 子目录。

/r:System.Data.dll

引用 Microsoft Windows 系统数据动态链接库。

/r:System.dll

引用 Microsoft Windows 系统动态链接库。

/r:System.Xml.dll

引用 Microsoft Windows 系统 XML 动态链接库 (对于 SubCOM.vb)。

/r:SubCOM.dll

引用 SubCOM 动态链接库 (对于 RootCOM.vb 和 LCTrans.vb)。

/r:RootCOM.dll

引用 RootCOM 动态链接库 (对于 LCTrans.vb)。

请参阅编译器文档，以了解其他编译器选项。

C# .NET 应用程序编译和链接选项

本主题描述编译和链接 C# .NET 应用程序时的各种可用选项。

在 Windows 上使用 Microsoft C# 编译器构建 C++ 应用程序时，使用 DB2 中提供的编译和链接选项（如在 bldapp.bat 批处理文件中所述）。

注：.NET Framework V1.1 仅在与 .NET Provider V9.5 及更低版本配合使用时受支持。

使用 bldapp 的独立 C# 应用程序的编译和链接选项：

独立 C# 应用程序的编译和链接选项：

%BLDCOMP%

编译器的变量。缺省值为 csc，即 Microsoft C# 编译器。

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

引用您正在使用的 .NET Framework 版本的 DB2 动态链接库。

%VERSION%

应用程序支持多个版本的 .NET Framework。对于每个版本，DB2 都有一个动态链接库。对于 .NET Framework V2.0、3.0 和 3.5，%VERSION% 指向 netf20\ 子目录。

使用 bldapp 的松散耦合样本程序 LCTrans 的编译和链接选项：

%BLDCOMP%

编译器的变量。缺省值为 csc，即 Microsoft C# 编译器。

/out:RootCOM.d11

根据 RootCOM.cs 源文件输出 LCTrans 应用程序所使用的 RootCOM 动态链接库。

/out:SubCOM.d11

根据 SubCOM.cs 源文件输出 LCTrans 应用程序所使用的 SubCOM 动态链接库。

/target:library %1.cs

根据输入源文件（RootCOM.cs 或 SubCOM.cs）来创建动态链接库。

/r:System.EnterpriseServices.d11

引用 Microsoft Windows 系统企业服务数据链路库。

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

引用您正在使用的 .NET Framework 版本的 DB2 动态链接库。

%VERSION%

应用程序支持多个版本的 .NET Framework。对于每个版本，DB2 都有一个动态链接库。对于 .NET Framework V2.0、3.0 和 3.5，%VERSION% 指向 netf20\ 子目录。

/r:System.Data.d11

引用 Microsoft Windows 系统数据动态链接库。

/r:System.d11

引用 Microsoft Windows 系统动态链接库。

/r:System.Xml.d11

引用 Microsoft Windows 系统 XML 动态链接库（对于 SubCOM.cs）。

/r:SubCOM.d11

引用 SubCOM 动态链接库（对于 RootCOM.cs 和 LCTrans.cs）。

`/r:RootCOM.dll`

引用 RootCOM 动态链接库（对于 LCTrans.cs）。

请参阅编译器文档，以了解其他编译器选项。

OLE DB .NET Data Provider

OLE DB .NET Data Provider 使用 IBM DB2 OLE DB Driver，在 `ConnectionString` 对象中，将后者作为 `IBMDADB2` 进行引用。

OLE DB .NET Data Provider 支持的连接字符串关键字与 IBM OLE DB Provider for DB2 支持的那些关键字相同。我们不再测试此提供程序。建议用户使用 IBM Data Server Provider for .NET。

并且，OLE DB .NET Data Provider 的限制也与 IBM DB2 OLE DB Provider 相同。OLE DB .NET Data Provider 还有其他限制，以下主题对这些限制作了阐述：《开发 ADO.NET 和 OLE DB 应用程序》中的『OLE DB .NET Data Provider 限制』。

要使用 OLE DB .NET Data Provider，必须安装 .NET Framework V2.0、V3.0 或 V3.5。

对于 DB2 Universal Database for AS/400[®] R520、R530 和 R540，服务器上需要以下修订：APAR ii13348。

OLE DB .NET Data Provider 的所有支持的连接关键字显示在表 1 中：

表 2. OLE DB .NET Data Provider 的有用 `ConnectionString` 关键字

关键字	值	含义
PROVIDER	IBMDADB2	指定 IBM OLE DB Provider for DB2（必需）
DSN 或 Data Source	数据库别名	数据库目录中编目的 DB2 数据库别名
UID	用户标识	用于连接到 DB2 数据服务器的用户标识
PWD	密码	用于连接到 DB2 数据服务器的用户标识的密码

注：要获取 `ConnectionString` 关键字的完整列表，请参阅 Microsoft 文档。

创建 `OleDbConnection` 以连接到 SAMPLE 数据库的示例是：

```
[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 限制

我们已不再对 OLE DB .NET Data Provider 进行测试。建议用户使用 IBM Data Server Provider for .NET。

下表标识 OLE DB .NET Data Provider 的使用限制:

表 3. OLE DB .NET Data Provider 限制

类或功能	限制描述	受影响的 DB2 服务器
ASCII 字符流	<p>在使用 <code>DbType.AnsiString</code> 或 <code>DbType.AnsiStringFixedLength</code> 时, 不能将 ASCII 字符流与 <code>OleDbParameters</code> 配合使用。</p> <p>OLE DB .NET Data Provider 将抛出以下异常: "Specified cast is not valid"</p> <p>变通方法: 使用 <code>DbType.Binary</code> 而不是 <code>DbType.AnsiString</code> 或 <code>DbType.AnsiStringFixedLength</code>。</p>	所有
ADORecord	不支持 ADORecord。	所有
ADORecordSet 和 Timestamp	<p>如 MSDN 所述, ADORecordSet 变体时间将解析为 1 秒。因此, 将 DB2 Timestamp 列存储到 ADORecordSet 时, 秒的所有小数部分都将丢失。同样, 根据 ADORecordSet 填写 DataSet 之后, DataSet 中的 Timestamp 列将不会包含秒的任何小数部分。</p> <p>变通方法: 此变通方法仅对 DB2 Universal Database for Linux, UNIX, and Windows V8.1 FP4 或更高版本生效。为避免分数秒的损失, 可设置以下 CLI 关键字: MAPTIMESTAMPDESCRIBE = 2</p> <p>此关键字会将 Timestamp 描述为 WCHAR(26)。要设置该关键字, 请从 DB2 命令窗口执行以下命令: db2 update cli cfg for section common using MAPTIMESTAMPDESCRIBE 2</p>	所有
Chapters	Chapters 不受支持。	所有
键信息	同时打开了 IDataReader 时, OLE DB .NET Data Provider 无法检索键信息。	DB2 for VM/VSE
来自存储过程的键信息	<p>OLE DB .NET Data Provider 只能检索来自 DB2 for Linux, UNIX, and Windows 的存储过程所返回结果集的键信息。这是因为, 用于除 Linux、UNIX 和 Windows 以外的平台的 DB2 服务器不会返回有关在存储过程中打开的结果集的扩展描述信息。</p> <p>要检索 DB2 for Linux, UNIX, and Windows 上的存储过程所返回结果集的键信息, 需要在 DB2 服务器上设置以下注册表变量: db2set DB2_APM_PERFORMANCE=8</p> <p>设置这个服务器端 DB2 注册表变量将在服务器上长时间保留结果集元数据, 从而使 OLE DB 能够成功地检索键信息。但是, 根据服务器工作负载的不同, 元数据可能不会被保留到 OLE DB Provider 查询信息之时。因此, 无法保证始终能够获得存储过程所返回结果集的键信息。</p> <p>要检索关于 CALL 语句的任何键信息, 应用程序必须执行该 CALL 语句。调用 <code>OleDbDataAdapter.FillSchema()</code> 或 <code>OleDbCommand.ExecuteReader(CommandBehavior.SchemaOnly CommandBehavior.KeyInfo)</code> 并不会实际地执行存储过程调用。因此, 将不会检索到存储过程将要返回的结果集的任何键信息。</p>	所有

表 3. OLE DB .NET Data Provider 限制 (续)

类或功能	限制描述	受影响的 DB2 服务器
批处理 SQL 语句的键信息	<p>使用返回多个结果的批处理 SQL 语句时, FillSchema() 方法将尝试只检索批处理 SQL 语句列表中第一个 SQL 语句的模式信息。如果此语句未返回结果集, 那么将不会创建任何表。例如:</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>由于批处理 SQL 语句中的第一个语句是“INSERT”语句, 此语句不返回结果集, 因此将不会在数据集中创建任何表。</p>	所有
OleDbCommandBuilder	<p>如果 SELECT 语句包含任何具有下列数据类型的列, 那么 OleDbCommandBuilder 自动生成的 UPDATE、DELETE 和 INSERT 语句不正确:</p> <ul style="list-style-type: none"> • CLOB • BLOB • DBCLOB • LONG VARCHAR • LONG VARCHAR FOR BIT DATA • LONG VARGRAPHIC <p>如果正在连接到除 DB2 for Linux, UNIX, and Windows 以外的 DB2 服务器, 那么具有下列数据类型的列也将引起此问题:</p> <ul style="list-style-type: none"> • VARCHAR¹ • VARCHAR FOR BIT DATA¹ • VARGRAPHIC¹ • REAL • FLOAT 或 DOUBLE • TIMESTAMP <p>注:</p> <p>1. 如果将具有这些数据类型的列定义为大于 254 字节的 VARCHAR 值、大于 254 字节的 VARCHAR 值 FOR BIT DATA 或者大于 127 字节的 VARGRAPHIC, 那么这些列适用。仅当正在连接到除 DB2 for Linux, UNIX, and Windows 以外的 DB2 服务器时, 此条件才有效。</p> <p>OleDbCommandBuilder 将生成在 WHERE 子句中的相等比较中使用所有所选列的 SQL 语句, 但不能在相等比较中使用先前列示的数据类型。</p> <p>注: 注意, 此限制将影响依靠 OleDbCommandBuilder 来自动生成 UPDATE、DELETE 和 INSERT 语句的 OleDbDataAdapter.Update() 方法。如果生成的语句包含先前列示的任何一种数据类型, 那么 UPDATE 操作将失败。</p> <p>变通方法: 您需要显示除去属于先前通过已生成 SQL 语句的 WHERE 子句列示的数据类型的所有列。建议编写您自己的 UPDATE、DELETE 和 INSERT 语句。</p>	所有
OleDbCommandBuilder. DeriveParameters	<p>使用 DeriveParameters() 时, 区分大小写至关重要。OleDbCommand.CommandText 所指定存储过程名的大小写必须与 DB2 系统目录表中存储的大小写匹配。要查看存储过程名的存储方式, 请在不提供过程名限制的情况下调用 OpenSchema (OleDbSchemaGuid.Procedures)。这将返回所有存储过程名。缺省情况下, DB2 将存储过程名以大写方式进行存储, 因此, 您通常需要以大写方式指定存储过程名。</p>	所有

表 3. OLE DB .NET Data Provider 限制 (续)

类或功能	限制描述	受影响的 DB2 服务器
OleDbCommandBuilder. DeriveParameters	OleDbCommandBuilder.DeriveParameters() 方法在生成的 OleDbParameterCollection 中不包括 ReturnValue 参数。缺省情况下, SqlClient 和 IBM Data Server Provider for .NET 将该参数及 ParameterDirection.ReturnValue 添加到所生成的 ParameterCollection。	所有
OleDbCommandBuilder. DeriveParameters	对于重载的存储过程, OleDbCommandBuilder.DeriveParameters() 方法将失败。如果有多个名为“MYPROC”的存储过程, 并且每个存储过程都接受不同数目的参数或不同类型的参数, 那么 OleDbCommandBuilder.DeriveParameters() 将检索所有重载的存储过程的所有参数。	所有
OleDbCommandBuilder. DeriveParameters	如果应用程序未使用模式来限定存储过程, 那么 DeriveParameters() 将返回该过程名的所有参数。因此, 如果对于同一个过程名存在多个模式, 那么 DeriveParameters() 将返回所有同名过程的所有参数。	所有
OleDbConnection. ChangeDatabase	不支持 OleDbConnection.ChangeDatabase() 方法。	所有
OleDbConnection. ConnectionString	在连接字符串中使用不可打印字符 (例如“\b”、“\a”或“\O”) 将导致抛出异常。 下列关键字具有限制: Data Source 数据源是数据库的名称, 而不是服务器的名称。您可以指定 SERVER 服务器, 但 IBMDADB2 提供程序将忽略此关键字。 Initial Catalog 和 Connect Timeout 不支持这些关键字。通常, OLE DB .NET Data Provider 将忽略所有无法识别的和不受支持的关键字。但是, 指定这些关键字将引起以下异常: Multiple-step OLE DB operation generated errors. Check each OLE DB status value, if available. No work was done. ConnectionTimeout ConnectionTimeout 是只读的。	所有
OleDbConnection. GetOleDbSchemaTable	限制值区分大小写, 并且必须与系统目录表中存储的数据库对象的大小写 (缺省为大写) 匹配。 例如, 如果您已经以此方式创建表: CREATE TABLE abc(c1 SMALLINT) DB2 会以大写方式将表名 (“ABC”) 存储在系统目录中。因此, 您必须使用“ABC”作为限制值。例如: schemaTable = con.GetOleDbSchemaTable(OleDbSchemaGuid.Tables, new object[] { null, null, "ABC", "TABLE" }); 变通方法: 如果数据定义中需要区分大小写或空格, 那么必须将它们括在引号中。例如: cmd.CommandText = "create table \"Case Sensitive\"(c1 int); cmd.ExecuteNonQuery(); tablename = "\"Case Sensitive\""; schemaTable = con.GetOleDbSchemaTable(OleDbSchemaGuid.Tables, new object[] { null, null, tablename, "TABLE" });	所有
OleDbDataAdapter 和 DataColumnMapping	源列名区分大小写。它必须与 DB2 目录中存储的大小写 (缺省为大写) 匹配。 例如: colMap = new DataColumnMapping("EMPNO", "Employee ID");	所有

表 3. OLE DB .NET Data Provider 限制 (续)

类或功能	限制描述	受影响的 DB2 服务器
OleDbDataReader. GetSchemaTable	OLE DB .NET Data Provider 无法从不返回扩展描述信息的服务器检索扩展描述信息。如果您正在连接到不支持扩展描述的服务器（受影响的服务器），那么 IDataReader.GetSchemaTable() 所返回的元数据表中的下列各列无效： <ul style="list-style-type: none"> • IsReadOnly • IsUnique • IsAutoIncrement • BaseSchemaName • BaseCatalogName 	DB2 for OS/390® V7 或更低版本 DB2 for OS/400 DB2 for VM/VSE
存储过程：没有结果集的列名	DB2 for OS/390 V6.1 服务器不返回从存储过程返回的结果集的列名。OLE DB .NET Data Provider 将把这些未命名的列映射到序号位置（例如“1”、“2”或“3”）。这与 MSDN 中阐述的映射不同：“Column1”、“Column2”或“Column3”。	DB2 for OS/390 V6.1

提示与技巧

OLE DB .NET Data Provider 应用程序中的连接合用

OLE DB .NET Data Provider 使用 OLE DB 会话合用自动合用连接。

OLE DB .NET Data Provider 使用 OLE DB 会话合用自动合用连接。您可以使用连接字符串自变量来启用或禁用 OLE DB 服务（包括合用）。例如，以下连接字符串将禁用 OLE DB 会话合用和自动事务授权。

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

下表描述可以用于设置 OLE DB 服务的 ADO 连接字符串属性：

表 4. 使用 ADO 连接字符串属性来设置 OLE DB 服务

启用的服务	连接字符串中的值
所有服务（缺省）	"OLE DB Services = -1;"
除合用以外的所有服务	"OLE DB Services = -2;"
除合用和自动授权以外的所有服务	"OLE DB Services = -4;"
除客户机游标以外的所有服务	"OLE DB Services = -5;"
除客户机游标与合用以外的所有服务	"OLE DB Services = -6;"
无服务	"OLE DB Services = 0;"

有关 OLE DB 会话合用或资源合用以及如何通过覆盖 OLE DB 提供程序服务缺省值来禁用合用的更多信息，请参阅 MSDN 资料库中的 OLE DB Programmer's Reference，网址为：

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

OLE DB .NET Data Provider 应用程序中的时间列

下列各节描述如何在 OLE DB .NET Data Provider 应用程序中实现时间列。

下列各节描述如何在 OLE DB .NET Data Provider 应用程序中实现时间列。

使用参数标记进行插入

您想要将时间值插入到时间列中:

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

其中, 列 c1 是时间列。下面是两种将时间值与参数标记绑定的方法:

使用 `OleDbParameter.OleDbType = OleDbType.DBTime`

由于 `OleDbType.DBTime` 映射到 `TimeSpan` 对象, 因此您必须提供 `TimeSpan` 对象作为参数值。此参数值不能是 `String` 或 `DateTime` 对象, 它必须是 `TimeSpan` 对象。例如:

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

`TimeSpan` 的格式以格式为“[-]d.hh:mm:ss.ff”的字符串表示, MSDN 文档对此作了阐述。

使用 `OleDbParameter.OleDbType = OleDbType.DateTime`

这将强制 OLE DB .NET Data Provider 将参数值转换为 `DateTime` 对象 (而不是 `TimeSpan` 对象), 因此, 参数值可以是任何可以转换为 `DateTime` 对象的有效字符串/对象。这意味着, “11:20:30”之类的值有效。值还可以是 `DateTime` 对象。由于 `TimeSpan` 对象无法转换为 `DateTime` 对象, 因此, 值不能是 `TimeSpan` 对象 - `TimeSpan` 未实现 `IConvertible`。

例如:

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

检索

要检索时间列, 您需要使用 `IDataRecord.GetValue()` 方法或 `OleDbDataReader.GetTimeSpan()` 方法。

例如:

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

OLE DB .NET Data Provider 应用程序中的 ADORecordset 对象

使用 `ADORecordset` 对象时, 必须考虑 ADO 类型 `adDBTime` 类。

有关使用 `ADORecordset` 对象的注意事项。

- ADO 类型 `adDBTime` 类映射到 .NET Framework `DateTime` 类。`OleDbType.DBTime` 与 `TimeSpan` 对象相对应。
- 不能将 `TimeSpan` 对象赋予 `ADORecordset` 对象的 `Time` 字段。这是因为, `ADORecordset` 对象的 `Time` 字段应该包含 `DateTime` 对象。将 `TimeSpan` 对象赋予 `ADORecordset` 对象时, 您将接收到以下消息:

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

在 `Time` 字段中, 只能填充 `DateTime` 对象或者可以解析为 `DateTime` 对象的 `String`。

- 在 DataSet 中填充使用 OleDbDataAdapter 的 ADORecordset 时, ADORecordset 中的 Time 字段将转换为 DataSet 中的 TimeSpan 列。
- Recordsets 不存储主键或约束。因此, 根据使用 MissingSchemaAction.AddWithKey 的 Recordset 来填充 DataSet 时, 不会添加任何键信息。

ODBC .NET Data Provider

ODBC .NET Data Provider 使用 CLI 驱动程序对 DB2 数据源进行 ODBC 调用。因此, ODBC .NET Data Provider 支持的连接字符串关键字与 CLI 驱动程序支持的连接字符串关键字相同。我们不再测试此提供程序。建议用户使用 IBM Data Server Provider for .NET。

而且, ODBC .NET Data Provider 的限制与 CLI 驱动程序的限制相同。ODBC .NET Data Provider 还有其他限制, 以下主题对这些限制作了阐述: 《开发 ADO.NET 和 OLE DB 应用程序》中的『ODBC .NET Data Provider 限制』。

要使用 ODBC .NET Data Provider, 必须安装 .NET Framework V2.0、V3.0 或 V3.5。对于 DB2 Universal Database for AS/400 V5R4 及更低版本, 服务器上需要以下修订: APAR II13348。

表 1 中列示了 ODBC .NET Data Provider 的受支持的连接关键字:

表 5. ODBC .NET Data Provider 的有用 **ConnectionString** 关键字

关键字	值	含义
DSN	数据库别名	数据库目录中编目的 DB2 数据库别名
UID	用户标识	用于连接到 DB2 服务器的用户标识
PWD	密码	用于连接到 DB2 服务器的用户标识的密码

注: 要获取 **ConnectionString** 关键字的完整列表, 请参阅 Microsoft 文档。

以下代码是创建 OdbcConnection 以连接到 SAMPLE 数据库的示例:

```
[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 限制

我们已不再对 ODBC .NET Data Provider 进行测试。建议用户使用 IBM Data Server Provider for .NET。

下表标识 ODBC .NET Data Provider 的使用限制:

表 6. ODBC .NET Data Provider 限制

类或功能	限制描述	受影响的 DB2 服务器
ASCII 字符流	<p>在使用 DbType.AnsiString 或 DbType.AnsiStringFixedLength 时, 不能将 ASCII 字符流与 OdbcParameters 配合使用。</p> <p>ODBC .NET Data Provider 将抛出以下异常: "Specified cast is not valid"</p> <p>变通方法: 使用 DbType.Binary 而不是 DbType.AnsiString 或 DbType.AnsiStringFixedLength。</p>	所有
Command.Prepare	<p>如果 CommandText 自从上次准备后已更改, 那么在执行命令 (Command.ExecuteNonQuery 或 Command.ExecuteReader) 之前, 必须显式地运行 OdbcCommand.Prepare()。如果未再次调用 OdbcCommand.Prepare(), 那么 ODBC .NET Data Provider 将执行先前准备的 CommandText。</p> <p>例如:</p> <pre>[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>	所有
CommandBehavior.SequentialAccess	<p>使用 IDataReader.GetChars() 从通过 CommandBehavior.SequentialAccess 创建的阅读器读取内容时, 必须分配大小足以存放整个列的缓冲区。否则, 将发生以下异常:</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 OleRestrict.TestGetCharsAndBufferSize(IDbConnection con)</pre> <p>以下示例演示如何分配足够大的缓冲区:</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>对 GetChars() 的调用将抛出以下异常: "Requested range extends past the end of the array"</p> <p>要确保 GetChars() 不抛出上面提到的异常, 必须将 BufferSize 设置为列的大小, 如下所示:</p> <pre>Int32 iBufferSize = 10000;</pre> <p>注意, iBufferSize 的值 10,000 与分配给 CLOB 列 c1 的值 10K 相对应。</p>	所有

表 6. ODBC .NET Data Provider 限制 (续)

类或功能	限制描述	受影响的 DB2 服务器
CommandBehavior.SequentialAccess	<p>使用 OdbcDataReader.GetChars() 时, 如果没有更多可读取的数据, 那么 ODBC .NET Data Provider 将抛出以下异常:</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>	所有
CommandBehavior.SequentialAccess	<p>使用 OdbcDataReader.GetChars() 时, 可能无法使用较大的块大小, 例如值 5000。当您尝试使用较大的块大小时, ODBC .NET Data Provider 将抛出以下异常:</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>	所有
连接合用	<p>ODBC .NET Data Provider 不控制连接合用。连接合用由 ODBC 驱动程序管理器处理。有关连接合用的更多信息, 请参阅 MSDN 库中的 ODBC Programmer's Reference, 网址为:</p> <p>http://msdn.microsoft.com/library</p>	所有
DataColumnMapping	<p>源列名的大小写必须与系统目录表中使用的大小写 (缺省为大写) 匹配。</p>	所有
Decimal 列	<p>Decimal 列不支持参数标记。</p> <p>通常, 如果目标 SQLType 是 Decimal 列, 那么您将 OdbcType.Decimal 用于 OdbcParameter; 但是, 当 ODBC .NET Data Provider 检测到 OdbcType.Decimal 时, 它使用 C 类型 SQL_C_WCHAR 以及 SQLType 值 SQL_VARCHAR 来绑定参数, 这是无效的。</p> <p>例如:</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>您将接收到以下异常:</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>解决方法: 请完全使用字面值而不是 OdbcParameter 值。</p>	DB2 for VM/VSE

表 6. ODBC .NET Data Provider 限制 (续)

类或功能	限制描述	受影响的 DB2 服务器
键信息	<p>用于限定表名的模式名 (例如 MYSCHEMA.MYTABLE) 必须与连接用户标识匹配。ODBC .NET Data Provider 无法检索任何其指定模式与连接用户标识不同的键信息。</p> <p>例如:</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>	所有
键信息	<p>同时打开了 IDataReader 时, ODBC .NET Data Provider 无法检索键信息。当 ODBC .NET Data Provider 打开 IDataReader 时, 将在服务器上打开游标。如果已请求获取键信息, 那么它接着将调用 SQLPrimaryKeys() 或 SQLStatistic() 以获取键信息, 但这些模式函数不会打开另一个游标。因为 DB2 for VM/VSE 不支持挂起游标, 所以接着将关闭第一个游标。因此, IDataReader.Read() 调用 IDataReader 将引起以下异常:</p> <pre>System.Data.Odbc.OdbcException: ERROR [HY010] [IBM][CLI Driver] CLI0125E Function sequence error. SQLSTATE=HY010</pre> <p>解决方法: 您需要先检索键信息, 然后检索数据。例如:</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
键信息	<p>在 SQL 语句中引用数据库对象时, 使用的大小写必须与系统目录表中存储的数据库对象的大小写相同。缺省情况下, 数据库对象以大写方式存储在系统目录表中, 所以, 您通常需要使用大写。</p> <p>ODBC .NET Data Provider 将扫描 SQL 语句以检索数据库对象名并将其传递到模式函数 (例如 SQLPrimaryKeys 和 SQLStatistics), 这些模式函数将对系统目录表中的这些对象发出查询。数据库对象引用必须与这些对象在系统目录表中的存储方式完全匹配, 否则, 将返回空结果集。</p>	DB2 for OS/390 DB2 for OS/400 DB2 for VM/VSE
批处理非 SELECT SQL 语句的键信息	ODBC .NET Data Provider 无法检索未以“SELECT”开头的批处理语句的任何键信息。	DB2 for OS/390 DB2 for OS/400 DB2 for VM/VSE

表 6. ODBC .NET Data Provider 限制 (续)

类或功能	限制描述	受影响的 DB2 服务器
LOB 列	<p>ODBC .NET Data Provider 不支持 LOB 数据类型。因此，每当 DB2 服务器返回 SQL_CLOB (-99)、SQL_BLOB (-98) 或 SQL_DBCLOB (-350) 时，ODBC .NET Data Provider 将抛出以下异常：</p> <pre>"Unknown SQL type - -98" (for Blob column) "Unknown SQL type - -99" (for Clob column) "Unknown SQL type - -350" (for DbClob column)</pre> <p>任何直接或间接访问 LOB 列的方法都将失败。</p> <p>解决方法： 将 CLI/ODBC LongDataCompat 关键字设置为 1。这样做会强制 CLI 驱动程序将以下数据类型映射至 ODBC .NET Data Provider 可识别的数据类型：</p> <ul style="list-style-type: none"> • SQL_CLOB 至 SQL_LONGVARCHAR • SQL_BLOB 至 SQL_LONGVARIABLE • SQL_DBCLOB 至 SQL_WLONGVARCHAR <p>要设置 LongDataCompat 关键字，请从客户机上的 DB2 命令窗口运行以下 DB2 命令：</p> <pre>db2 update cli cfg for section common using longdatacompat 1</pre> <p>通过按如下所示使用连接字符串，还可在应用程序设置此关键字：</p> <pre>[C#] OdbcConnection con = new OdbcConnection("DSN=SAMPLE;UID=uid;PWD=myspw;LONGDATACOMPAT=1;");</pre> <p>有关所有 CLI/ODBC 关键字的列表，请参阅“UID CLI/ODBC 配置关键字”（在《DB2 CLI 指南和参考》中）。</p>	所有
OdbcCommand.Cancel	<p>在运行 OdbcCommand.Cancel 之后执行语句可能会引起以下异常：</p> <pre>"ERROR [24000] [Microsoft][ODBC Driver Manager] Invalid cursor state"</pre>	所有
OdbcCommandBuilder	<p>对于不支持转义字符的服务器，OdbcCommandBuilder 无法生成命令。当 OdbcCommandBuilder 生成命令时，它首先调用 SQLGetInfo，从而请求获取 SQL_SEARCH_PATTERN_ESCAPE 属性。如果服务器不支持转义字符，那么将返回空字符串，这将导致 ODBC .NET Data Provider 抛出以下异常：</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，仅适用于 DBCS 服务器；DB2 for VM/VSE，仅适用于 DBCS 服务器

表 6. ODBC .NET Data Provider 限制 (续)

类或功能	限制描述	受影响的 DB2 服务器
OdbcCommandBuilder	<p>使用 OdbcCommandBuilder 来自动生成 UPDATE、DELETE 和 INSERT 语句时，区分大小写至关重要。缺省情况下，除非您在创建期间通过在数据库对象两旁添加引号以区分大小写方式显式创建这些对象，否则 DB2 以大写方式将模式信息（例如表名和列名）存储在系统目录表中。因此，SQL 语句必须与目录中存储的大小写（缺省为大写）匹配。</p> <p>例如，如果已使用以下语句来创建表： "db2 create table mytable (c1 int) "</p> <p>那么 DB2 将把表名“mytable”作为“MYTABLE”存储在系统目录表中。</p> <p>以下代码示例演示如何正确使用 OdbcCommandBuilder 类：</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>在此示例中，如果未使用大写字符来引用表名，那么您将接收到以下异常： "Dynamic SQL generation for the UpdateCommand is not supported against a SelectCommand that does not return any key column information."</p>	所有
OdbcCommandBuilder	<p>当 SELECT 语句包含下面这些列数据类型时，OdbcCommandBuilder 生成的命令不正确： REAL FLOAT 或 DOUBLETIMESTAMP</p> <p>在 SELECT 语句的 WHERE 子句中，不能使用这些数据类型。</p>	DB2 for OS/390 DB2 for OS/400 DB2 for VM/VSE
OdbcCommandBuilder. DeriveParameters	<p>DeriveParameters() 方法映射到 SQLProcedureColumns，它使用 CommandText 属性作为存储过程的名称。由于 CommandText 未包含存储过程的名称（使用完整 ODBC 调用语法），因此将在根据 ODBC 调用语法标识过程名的情况下调用 SQLProcedureColumns。例如： "{ CALL myProc(?) }"</p> <p>这将生成一个空的结果集，在此结果集中，找不到过程的列。</p>	所有
OdbcCommandBuilder. DeriveParameters	<p>要使用 DeriveParameters()，请在 CommandText 中指定存储过程名（例如 cmd.CommandText = "MYPROC"）。过程名的大小写必须与系统目录表中存储的大小写匹配。DeriveParameters() 将返回它在系统目录表中找到的过程名的所有参数。在执行语句前，请不要忘记将 CommandText 重新更改为完整的 ODBC 调用语法。</p>	所有
OdbcCommandBuilder. DeriveParameters	<p>对于 ODBC .NET Data Provider 而言，不会返回 ReturnValue 参数。</p>	所有
OdbcCommandBuilder. DeriveParameters	<p>DeriveParameters() 不支持标准存储过程名。例如，为 CommandText = "MYSCHEMA.MYPROC" 调用 DeriveParameters() 将失败。这里，不会返回任何参数。</p>	所有
OdbcCommandBuilder. DeriveParameters	<p>对于重载的存储过程，DeriveParameters() 无效。SQLProcedureColumns 将返回存储过程的所有版本的所有参数。</p>	所有
OdbcConnection. ChangeDatabase	<p>不支持 OdbcConnection.ChangeDatabase() 方法。</p>	所有
OdbcConnection. ConnectionString	<ul style="list-style-type: none"> Server 关键字将被忽略。 Connect Timeout 关键字将被忽略。CLI 不支持连接超时，所以设置此属性不会影响该驱动程序。 连接合用关键字将被忽略。具体而言，这将影响下列关键字：Pooling、Min Pool Size、Max Pool Size、Connection Lifetime 和 Connection Reset。 	所有

表 6. ODBC .NET Data Provider 限制 (续)

类或功能	限制描述	受影响的 DB2 服务器
OdbcDataReader. GetSchemaTable	<p>ODBC .NET Data Provider 无法从不返回扩展描述信息的服务器检索扩展描述信息。因此, 如果您正在连接到不支持扩展描述的服务器 (受影响的服务器), 那么 IDataReader.GetSchemaTable() 所返回的元数据表中的下列各列无效:</p> <ul style="list-style-type: none"> • IsReadOnly • IsUnique • IsAutoIncrement • BaseSchemaName • BaseCatalogName 	<p>DB2 for OS/390 V7 或更低版本 DB2 for OS/400 DB2 for VM/VSE</p>
存储过程	<p>要调用存储过程, 需要指定完整的 ODBC 调用语法。</p> <p>例如, 要调用接受 VARCHAR(10) 作为参数的存储过程 MYPROC:</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: 注意, 即使您正在使用 CommandType.StoredProcedure, 也必须使用完整的 ODBC 调用语法。在 MSDN 中, 有关 OdbcCommand.CommandText 属性的内容对此语法作了阐述。</p>	<p>所有</p>
存储过程: 没有结果集的列名	<p>DB2 for OS/390 V6.1 服务器不返回从存储过程返回的结果集的列名。ODBC .NET Data Provider 将这些未名列映射至其序号位置 (例如, “1”、“2”和“3”)。这与 MSDN 中阐述的映射不同: "Column1"、"Column2"或"Column3"。</p>	<p>DB2 for OS/390 V6.1</p>
唯一索引提升为主键	<p>ODBC .NET Data Provider 将可空唯一索引提升为主键。这与 MSDN 文档中的陈述不同, 该文档指出, 不应将可空唯一索引提升为主键。</p>	<p>所有</p>

第 2 章 IBM OLE DB Provider for DB2

IBM OLE DB Provider for DB2 允许 DB2 充当 OLE DB 提供程序的资源管理器。此支持使基于 OLE DB 的应用程序能够使用 OLE 接口来抽取或查询 DB2 数据。

Microsoft OLE DB 是一组 OLE/COM 接口，它们使应用程序能够对各种信息源中存储的数据进行一致的访问。OLE DB 体系结构定义了 OLE DB 使用者和 OLE DB 提供程序。OLE DB 使用者是任何使用 OLE DB 接口的系统或应用程序；OLE DB 提供程序是任何提供 OLE DB 接口的组件。

IBM OLE DB Provider for DB2 的提供程序名称为 IBMDADB2，它使 OLE DB 使用者能够访问 DB2 数据库服务器上的数据。如果已安装 DB2 Connect，那么这些 OLE DB 使用者还可以访问主机 DBMS（例如 DB2 for z/OS、DB2 Server for VM and VSE 或 DB2 Universal Database for AS/400）上的数据。

IBM OLE DB Provider for DB2 提供了下列功能：

- 支持第 0 级 OLE DB 提供程序规范，其中包括某些附加的第 1 级接口。
- 自由的线程化提供程序实现，此实现使应用程序能够在单个线程中创建组件并在任何其他线程中使用那些组件。
- 错误查找服务，此服务用于返回 DB2 错误消息。

注意，IBM OLE DB Provider 驻留在客户机上，并且与同样受 DB2 数据库系统支持的 OLE DB 表功能有所不同。

本文档的后续章节将描述 IBM OLE DB Provider for DB2 的特定实现。有关 Microsoft OLE DB 2.0 规范的更多信息，请参阅 Microsoft Press 提供的 Microsoft OLE DB 2.0 Programmer's Reference and Data Access SDK。

版本一致性

IBM OLE DB Provider for DB2 与 Microsoft OLE DB 规范的 V2.7 或更高版本一致。

系统要求

请参阅 IBM OLE DB Provider for DB2 数据服务器的声明函，以了解受支持的 Windows 操作系统。

要安装 IBM OLE DB Provider for DB2，必须先在上面列示的其中一款受支持操作系统中运行。您还需要安装一个完整的 DB2 产品、IBM Data Server Driver for ODBC and CLI 或 IBM Data Server Driver Package。

IBM OLE DB Provider for DB2 支持的应用程序类型

使用 IBM OLE DB Provider for DB2，可以创建 Microsoft Visual Studio C++ 或 Visual Basic 应用程序。

使用 IBM OLE DB Provider for DB2，可创建下列类型的应用程序：

- ADO 应用程序，其中包括：

- Microsoft Visual Studio C++ 应用程序
- Microsoft Visual Basic 应用程序
- 使用 OLE DB .NET Data Provider 的 ADO.NET 应用程序
- 使用 OLE DB 接口来直接访问 IBMDADB2 的 C/C++ 应用程序，其中包括其数据访问使用者对象由 ATL COM AppWizard 生成的 ATL 应用程序。

OLE DB 服务

IBM OLE DB Provider 所支持的线程模型

IBM OLE DB Provider for DB2 支持自由线程化模型。这允许应用程序在一个线程中创建组件并在任何其他线程中使用那些组件。

IBM OLE DB Provider for DB2 支持自由线程化模型。这允许应用程序在一个线程中创建组件并在任何其他线程中使用那些组件。

通过 IBM OLE DB Provider 进行大对象处理

要通过 IBMDADB2 提供程序获取数据并将数据设置为存储器对象 (DBTYPE_IUNKNOWN), 请使用 ISequentialStream 接口。

要通过 IBMDADB2 提供程序获取数据并将数据设置为存储器对象 (DBTYPE_IUNKNOWN), 请使用 ISequentialStream 接口, 如下所示:

- 要将存储器对象与参数绑定, DBBINDING 结构中的 DBOBJECT 的 dwFlag 字段只能包含值 STGM_READ。IBMDADB2 将执行所绑定对象的 ISequentialStream 接口的 Read 方法。
- 要从存储器对象中获取数据, 应用程序必须运行该存储器对象的 ISequentialStream 接口的 Read 方法。
- 获取数据时, 长度部件的值是实际数据的长度, 而不是 IUnknown 指针的长度。

IBM OLE DB Provider 所支持的模式行集

下表列示 IDBSchemaRowset 所支持的模式行集。

下表列示 IDBSchemaRowset 所支持的模式行集。在行集中, 不受支持的列将设置为 null。

表 7. IBM OLE DB Provider for DB2 支持的模式行集

受支持的 GUID	受支持的限制	受支持的列	注意
DBSCHEMA _COLUMN_PRIVILEGES	COLUMN_NAME TABLE_NAME TABLE_SCHEMA	COLUMN_NAME GRANTEE GRANTOR IS_GRANTABLE PRIVILEGE_TYPE TABLE_NAME TABLE_SCHEMA	

表 7. IBM OLE DB Provider for DB2 支持的模式行集 (续)

受支持的 GUID	受支持的限制	受支持的列	注意
DBSCHEMA_COLUMNS	COLUMN_NAME TABLE_NAME TABLE_SCHEMA	CHARACTER_MAXIMUM_LENGTH CHARACTER_OCTET_LENGTH COLUMN_DEFAULT COLUMN_FLAGS COLUMN_HASDEFAULT COLUMN_NAME DATA_TYPE DESCRIPTIONIS_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_SCHEMAUPDATE_RULE	必须至少指定下列其中一项限制： PK_TABLE_NAME 或 FK_TABLE_NAME 不允许使用“%”通配符。
DBSCHEMA_INDEXES	TABLE_NAME TABLE_SCHEMA	CARDINALITY CLUSTERED COLLATION COLUMN_NAME INDEX_NAME INDEX_SCHEMA ORDINAL_POSITION PAGES TABLE_NAME TABLE_SCHEMATYPE UNIQUE	不支持排序顺序。如果指定了排序顺序，那么会将其忽略。
DBSCHEMA_PRIMARY_KEYS	TABLE_NAME TABLE_SCHEMA	COLUMN_NAME ORDINAL PK_NAME TABLE_NAME TABLE_SCHEMA	必须至少指定以下限制： TABLE_NAME 不允许使用“%”通配符。

表 7. IBM OLE DB Provider for DB2 支持的模式行集 (续)

受支持的 GUID	受支持的限制	受支持的列	注意
DBSCHEMA _PROCEDURE_PARAMETERS	PARAMETER_NAME PROCEDURE_NAME PROCEDURE_SCHEMA	CHARACTER_MAXIMUM_LENGTH CHARACTER_OCTET_LENGTH DATA_TYPE DESCRIPTIONIS_NULLABLE NUMERIC_PRECISION NUMERIC_SCALE ORDINAL_POSITION PARAMETER_DEFAULT PARAMETER_HASDEFAULT PARAMETER_NAME PARAMETER_TYPE PROCEDURE_NAME PROCEDURE_SCHEMATYPE_NAME	
DBSCHEMA_PROCEDURES	PROCEDURE_NAME PROCEDURE_SCHEMA	DESCRIPTION PROCEDURE_NAME PROCEDURE_SCHEMA PROCEDURE_TYPE	
DBSCHEMA_PROVIDER_TYPES	DATA_TYPE BEST_MATCH	AUTO_UNIQUE_VALUE BEST_MATCHCASE_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_NAMEUNSIGNED_ATTRIBUTE	
DBSCHEMA_STATISTICS	TABLE_NAME TABLE_SCHEMA	CARDINALITY TABLE_NAME TABLE_SCHEMA	不支持排序顺序。如果指定了排序顺序，那么会将其忽略。
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 自动启用的 OLE DB 服务

缺省情况下，IBM OLE DB Provider for DB2 通过在提供者的类标识 (CLSID) 下添加 DWORD 值为 0xFFFFFFFF 的注册表条目 OLEDB_SERVICES 来自动启用所有 OLE DB 服务。

缺省情况下，IBM OLE DB Provider for DB2 通过在提供者的类标识 (CLSID) 下添加 DWORD 值为 0xFFFFFFFF 的注册表条目 OLEDB_SERVICES 来自动启用所有 OLE DB 服务。此值的含义如下：

表 8. OLE DB 服务

启用的服务	DWORD 值
所有服务 (缺省)	0xFFFFFFFF
除合用和自动授权以外的所有服务	0xFFFFFFFFC
除客户机游标以外的所有服务	0xFFFFFFFFB
除合用、授权和游标以外的所有服务	0xFFFFFFFF8
无服务	0x00000000

数据服务

IBM OLE DB Provider 所支持的游标方式

IBM OLE DB Provider for DB2 提供了对只读、只前进、可更新可滚动和可更新可滚动游标的本机支持。

IBM OLE DB Provider for DB2 提供了对只读、只前进、可更新可滚动和可更新可滚动游标的本机支持。

DB2 与 OLE DB 之间的数据类型映射

IBM OLE DB Provider for DB2 支持在 DB2 数据类型与 OLE DB 数据类型之间进行数据类型映射。

下表提供受支持映射以及用于指示列数据类型和参数数据类型的可用名称的完整列表。

表 9. DB2 数据类型与 OLE DB 数据类型之间的数据类型映射

DB2 数据类型	OLE DB 数据类型指示符	OLE DB 标准类型名称	特定于 DB2 的名称
SMALLINT	DBTYPE_I2	"DBTYPE_I2"	"SMALLINT"
INTEGER	DBTYPE_I4	"DBTYPE_I4"	"INTEGER" 或 "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" 或 "DOUBLE PRECISION"
DECIMAL	DBTYPE_NUMERIC	"DBTYPE_NUMERIC"	"DEC" 或 "DECIMAL"
NUMERIC	DBTYPE_NUMERIC	"DBTYPE_NUMERIC"	"NUM" 或 "NUMERIC"

表 9. DB2 数据类型与 OLE DB 数据类型之间的数据类型映射 (续)

DB2 数据类型	OLE DB 数据类型指示符	OLE DB 标准类型名称	特定于 DB2 的名称
DATE	DBTYPE_DBDATE	"DBTYPE_DBDATE"	"DATE"
TIME	DBTYPE_DBTIME	"DBTYPE_DBTIME"	"TIME"
TIMESTAMP	DBTYPE_DBTIMESTAMP	"DBTYPE_DBTIMESTAMP"	"TIMESTAMP"
CHAR	DBTYPE_STR	"DBTYPE_CHAR"	"CHAR" 或 "CHARACTER"
VARCHAR	DBTYPE_STR	"DBTYPE_VARCHAR"	"VARCHAR"
LONG VARCHAR	DBTYPE_STR	"DBTYPE_LONGVARCHAR"	"LONG VARCHAR"
CLOB	DBTYPE_STR 和 DBCOLUMNFLAGS_ISLONG 或 DBPARAMFLAGS_ISLONG	"DBTYPE_CHAR" "DBTYPE_VARCHAR" "DBTYPE_LONGVARCHAR" 和 DBCOLUMNFLAGS_ISLONG 或 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 和 DBCOLUMNFLAGS_ISLONG 或 DBPARAMFLAGS_ISLONG	"DBTYPE_WCHAR" "DBTYPE_WVARCHAR" "DBTYPE_WLONGVARCHAR" 和 DBCOLUMNFLAGS_ISLONG 或 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 和 DBCOLUMNFLAGS_ISLONG 或 DBPARAMFLAGS_ISLONG	"DBTYPE_BINARY" "DBTYPE_VARBINARY" "DBTYPE_LONGVARBINARY" 和 DBCOLUMNFLAGS_ISLONG 或 DBPARAMFLAGS_ISLONG	"BLOB"

用于设置将数据从 OLE DB 类型转换为 DB2 类型的数据转换

IBM OLE DB Provider for DB2 支持用于设置将数据从 OLE DB 类型转换为 DB2 类型的数据转换。

从 OLE DB 类型到 DB2 类型的受支持的数据转换

下表列示从 OLE DB 类型到 DB2 类型的数据转换。注意，在某些情况下可能会发生数据截断，这取决于数据的类型和值。

表 10. 从 OLE DB 类型到 DB2 类型的数据转换

OLE DB 类型指示符	DB2 数据类型																							
	S M A L L I N T	I N T E G E R	B I G I N T	R E A L	F L O A T I N G	D E C I M A L N U M B E R	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 P H I C	L O N G V A R R A P H I C	D B C L O B	对于位数据			D A T A L I N K			
																		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													
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													
DBTYPE_CY																								
DBTYPE_DECIMAL	X	X	X	X	X	X				X	X													

表 10. 从 OLE DB 类型到 DB2 类型的数据转换 (续)

OLE DB 类型指示符	DB2 数据类型																						
	S M A L L I N T	I N T E G E R	B I G I N T	R E A L	F L O A T I N G	D E C I M A L N U M B E R	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 G R A P H I C	L O N G V A R R A Y	D B C L O B	对于位数 数据			D A T A L I N K		
																		C H A R	V A R C H A R	L O N G V A R C H A R			
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 - 待确定																							
DBTYPE_STR	X	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 - 待确定																							
DBTYPE_IDISPATCH																							
DBTYPE_IUNKNOWN										X	X	X	X	X	X	X	X	X	X	X	X	X	
DBTYPE_GUID																							
DBTYPE_ERROR																							
DBTYPE_BYREF																							
DBTYPE_ARRAY																							
DBTYPE_VECTOR																							
DBTYPE_UDT																							

表 10. 从 OLE DB 类型到 DB2 类型的数据转换 (续)

OLE DB 类型指示符	DB2 数据类型																				
	S M A L L I N T	I N T E G E R	B I G I N T	R E A L	F L O A T I N G	D E C I M A L N U M B 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	C P H I C C	V A R R A P H I C	L O N G V A R R A P H I C	对于位数 数据			D A T A B A S E L I N K	
																	C H A R	V A R C H A R	L O N G V A R C H A R		
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																					

用于将数据从 DB2 类型转换为 OLE DB 类型的数据转换

为了获取数据，IBM OLE DB Provider 允许将数据由 DB2 类型转换为 OLE DB 类型。

从 DB2 类型到 OLE DB 类型的受支持数据转换

下表列示从 DB2 类型到 OLE DB 类型的受支持数据转换。注意，在某些情况下可能会发生数据截断，这取决于数据的类型和值。

表 11. 从 DB2 类型到 OLE DB 类型的数据转换

OLE DB 类型指示符	DB2 数据类型																						
	SMALLINT	INTEGER	BIGINT	REAL	FLOAT	DECIMAL	NUMERIC	DATE	TIME	TIMESTAMP	CHAR	VARCHAR	VARCHAR	CLOB	GRAPHIC	VARGRAPHIC	LONG	对于位数数据			BLOB	LINK	
																		CHAR	VARCHAR	LONG			
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. 从 DB2 类型到 OLE DB 类型的数据转换 (续)

OLE DB 类型指示符	DB2 数据类型																				
	S M A L L I N T	I N T E G E R	B I N A R Y	R E F E R E N C E	F L O A T	D E C I M A L	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	V A R C H A R L O B	G R A P H I C	V A R G R A P H I C	L O N G	D B C L O B	对于位数 据			D A T A L I N K	
																	C H A R	V A R C H A R	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	X

表 11. 从 DB2 类型到 OLE DB 类型的数据转换 (续)

OLE DB 类型指示符	DB2 数据类型																				
	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	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	V A R C H A R L O B	G R A P H I C	V A R R A P H I C	D B C L O B	对于位数 数据			D A T A L I N 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						X
DBTYPE_DBTIMESTAMP						X	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		X
DBTYPE_PROP_VARIANT	X	X	X	X	X					X	X	X		X	X	X		X	X	X	X
DBTYPE_HCHAPTER																					
DBTYPE_VARNUMERIC																					

注: 当应用程序执行 ISequentialStream::Read 以便从存储器对象中获取数据时, 所返回的数据的格式取决于列数据类型:

- 对于非字符和非二进制数据类型而言, 列的数据作为一系列字节返回, 这些字节在操作系统中代表那些值。
- 对于字符数据类型而言, 数据先转换为 DBTYPE_STR。
- 对于 DBCLOB 而言, 数据先转换为 DBTYPE_WCHAR。

IBM OLE DB Provider 限制

IBM OLE DB Provider 的限制是 IBMDADB2 支持自动落实和用户控制的事务作用域, 不支持 RestartPosition, 并且 IBMDADB2 不会将通过 DBID 参数传递的表名包括在引号中。

IBM OLE DB Provider 的限制是:

- 使用 ITransactionLocal 接口, IBMDADB2 支持自动落实事务作用域和用户控制的事务作用域。自动落实事务作用域是缺省作用域。不支持嵌套事务。
- 当命令文本包含参数时, 不支持 RestartPosition。
- IBMDADB2 不会将通过 DBID 参数 (这些是 IOpenRowset 接口所使用的参数) 传递的表名括在引号中。而是, OLE DB 使用者必须在需要引号时对表名添加引号。

IBM OLE DB Provider 对 OLE DB 组件和接口的支持

下列各表列示 IBM OLE DB Provider for DB2 和 Microsoft OLE DB Provider for ODBC 所支持的 OLE DB 组件和接口。

下列各表列示 IBM OLE DB Provider for DB2 和 Microsoft OLE DB Provider for ODBC 所支持的 OLE DB 组件和接口。

表 12. Blob

接口	DB2	ODBC Provider
ISequentialStream	是	是

表 13. 命令

接口	DB2	ODBC Provider
IAccessor	是	是
ICommand	是	是
ICommandPersist	否	否
ICommandPrepare	是	是
ICommandProperties	是	是
ICommandText	是	是
ICommandWithParameters	是	是
IColumnsInfo	是	是
IColumnsRowset	是	是
IConvertType	是	是
ISupportErrorInfo	是	是

表 14. 数据源

接口	DB2	ODBC Provider
IConnectionPoint	否	是
IDBAsynchNotify (consumer)	否	否
IDBAsynchStatus	否	否
IDBConnectionPointContainer	否	是
IDBCreateSession	是	是
IDBDataSourceAdmin	否	否
IDBInfo	是	是
IDBInitialize	是	是
IDBProperties	是	是
IPersist	是	否
IPersistFile	是	是
ISupportErrorInfo	是	是

表 15. 枚举符

接口	DB2	ODBC Provider
IDBInitialize	是	是

表 15. 枚举符 (续)

接口	DB2	ODBC Provider
IDBProperties	是	是
IParseDisplayName	是	否
ISourcesRowset	是	是
ISupportErrorInfo	是	是

表 16. 错误查找服务

接口	DB2	ODBC Provider
IErrorLookUp	是	是

表 17. 错误对象

接口	DB2	ODBC Provider
IErrorInfo	是	否
ISQLErrorInfo (定制)	是	否

表 18. 多个结果

接口	DB2	ODBC Provider
IMultipleResults	是	是
ISupportErrorInfo	是	是

表 19. 行集

接口	DB2	ODBC Provider
IAccessor	是	是
IColumnsRowset	是	是
IColumnsInfo	是	是
IConvertType	是	是
IChapteredRowset	否	否
IConnectionPointContainer	是	是
IDBAsynchStatus	否	否
IParentRowset	否	否
IRowset	是	是
IRowsetChange	是	是
IRowsetChapterMember	否	否
IRowsetFind	否	否
IRowsetIdentity	是	是
IRowsetIndex	否	否
IRowsetInfo	是	是
IRowsetLocate	是	是
IRowsetNotify (使用者)	是	否
IRowsetRefresh	游标服务组件	是
IRowsetResynch	游标服务组件	是

表 19. 行集 (续)

接口	DB2	ODBC Provider
IRowsetScroll	是 ¹	是
IRowsetUpdate	游标服务组件	是
IRowsetView	否	否
ISupportErrorInfo	是	是
注:		
1. 要返回的值是近似值。将不会跳过已删除的行。		

表 20. 会话

接口	DB2	ODBC Provider
IAlterIndex	否	否
IAlterTable	否	否
IDBCreateCommand	是	是
IDBSchemaRowset	是	是
IGetDataSource	是	是
IIndexDefinition	否	否
IOpenRowset	是	是
ISessionProperties	是	是
ISupportErrorInfo	是	是
ITableDefinition	否	否
ITableDefinitionWithConstraints	否	否
ITransaction	是	是
ITransactionJoin	是	是
ITransactionLocal	是	是
ITransactionObject	否	否
ITransactionOptions	否	是

表 21. 视图对象

接口	DB2	ODBC Provider
IViewChapter	否	否
IViewFilter	否	否
IViewRowset	否	否
IViewSort	否	否

IBM OLE DB Provider 对 OLE DB 属性的支持

下表列示 IBM OLE DB Provider for DB2 所支持的 OLE DB 属性和缺省值。

下表列示 IBM OLE DB Provider for DB2 所支持的 OLE DB 属性:

表 22. IBM OLE DB Provider for DB2 所支持的属性: 数据源 (DBPROPSET_DATASOURCE)

属性	缺省值	R/W
DBPROP_MULTIPLECONNECTIONS	VARIANT_FALSE	R
DBPROP_RESETDATASOURCE	DBPROPVAL_RD_RESETALL	R/W

表 23. IBM OLE DB Provider for DB2 所支持的属性: DB2 数据源 (DBPROPSET_DB2DATASOURCE)

属性	缺省值	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 所支持的属性: 数据源信息 (DBPROPSET_DATASOURCEINFO)

属性	缺省值	R/W
DBPROP_ACTIVESESSIONS	0	R
DBPROP_ASYNCCTXNABORT	VARIANT_FALSE	R
DBPROP_ASYNCCTXNCOMMIT	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	不适用	R
DBPROP_DATASOURCEREADONLY	VARIANT_FALSE	R
DBPROP_DBMSNAME	不适用	R
DBPROP_DBMSVER	不适用	R
DBPROP_DSOTHREADMODEL	DBPROPVAL_RT_FREETHREAD	R
DBPROP_GROUPBY	DBPROPVAL_GB_CONTAINS_SELECT	R
DBPROP_IDENTIFIERCASE	DBPROPVAL_IC_UPPER	R
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

表 24. IBM OLE DB Provider for DB2 所支持的属性: 数据源信息 (DBPROPSET_DATASOURCEINFO) (续)

属性	缺省值	R/W
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	不适用	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	不适用	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
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	不适用	R

表 25. IBM OLE DB Provider for DB2 所支持的属性: 初始化 (DBPROPSET_DBINIT)

属性	缺省值	R/W
DBPROP_AUTH_PASSWORD	不适用	R/W
DBPROP_INIT_TIMEOUT (1)	0	R/W
DBPROP_AUTH_PERSIST _SENSITIVE_AUTHINFO	VARIANT_FALSE	R/W
DBPROP_AUTH_USERID	不适用	R/W
DBPROP_INIT_DATASOURCE	不适用	R/W
DBPROP_INIT_HWND	不适用	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	不适用	R/W

表 26. IBM OLE DB Provider for DB2 所支持的属性: 行集 (DBPROPSET_ROWSET)

属性	缺省值	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
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

表 26. IBM OLE DB Provider for DB2 所支持的属性: 行集 (DBPROPSET_ROWSET) (续)

属性	缺省值	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 DBPROPBAL_NP_ABOUTTODD DBPROPVAL_NP_SYNCHAFTER DBPROPVAL_NP_FAILEDTODD DBPROPVAL_NP_DIDEVENT	R
DBPROP_NOTIFYROWSETRELEASE	DBPROPVAL_NP_OKTODO DBPROPVAL_NP_ABOUTTODD	R
DBPROP_NOTIFYROWSETFETCHPOSITIONCHANGE	DBPROPVAL_NP_OKTODO DBPROPVAL_NP_ABOUTTODD	R
DBPROP_NOTIFYCOLUMNSET	DBPROPVAL_NP_OKTODO DBPROPVAL_NP_ABOUTTODD	R
DBPROP_NOTIFYROWDELETE	DBPROPVAL_NP_OKTODO DBPROPVAL_NP_ABOUTTODD	R
DBPROP_NOTIFYROWINSERT	DBPROPVAL_NP_OKTODO DBPROPVAL_NP_ABOUTTODD	R
DBPROP_ORDEREDBOOKMARKS	VARIANT_FALSE	R
DBPROP_OTHERINSERT	VARIANT_FALSE	R
DBPROP_OTHERUPDATEDELETE	VARIANT_FALSE	R/W
DBPROP_OWNINGINSERT	VARIANT_FALSE	R
DBPROP_OWNINGUPDATEDELETE	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 所支持的属性: DB2 行集 (DBPROPSET_DB2ROWSET)

属性	缺省值	R/W
DBPROP_ISLONGMINLENGTH	32000	R/W

表 28. IBM OLE DB Provider for DB2 所支持的属性: 会话 (DBPROPSET_SESSION)

属性	缺省值	R/W
DBPROP_SESS_AUTOCOMMITISOLEVELS	DBPROPVAL_TI_CURSORSTABILITY	R/W

注:

1. 仅当使用 TCP/IP 协议来连接到服务器时, 超时才适用。只有在 TCP/IP 套接字连接期间, 才会强制实施超时限制。如果套接字连接在指定的超时到期之前完成, 那么将不会对余下的初始化过程强制实施超时限制。如果使用了客户机重新路由功能, 那么超时将加倍。通常, 启用客户机重新路由功能后, 连接超时行为由客户机重新路由功能指定。

使用 IBM OLE DB Provider 来连接到数据源

以下示例显示如何使用 IBM OLE DB Provider for DB2 连接到 DB2 数据源

以下示例显示如何使用 IBM OLE DB Provider for DB2 连接到 DB2 数据源。

示例 1: 使用 ADO 的 Visual Basic 应用程序

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

示例 2: 使用 IDataInitialize 和服务组件的 C/C++ 应用程序

```
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 应用程序

ADO 连接字符串关键字

要使用 ADO 连接字符串关键字, 必须在 `keyword=value` 中指定关键字的值。每个键值对由分号隔开。表 1 进一步说明关键字、值和含义。

要指定 ADO (ActiveX 数据对象) 连接字符串关键字, 请在提供程序 (连接) 字符串中使用 关键字=值 格式来指定关键字。要指定多个关键字, 请使用分号 (;) 对各个关键字进行定界。

下表描述 IBM OLE DB Provider for DB2 所支持的关键字:

表 29. IBM OLE DB Provider for DB2 支持的关键字

关键字	值	含义
DSN	数据库别名	数据库目录中的 DB2 数据库别名。
UID	用户标识	用于连接到 DB2 服务器的用户标识。
PWD	UID 的密码	用于连接到 DB2 服务器的用户标识的密码。

其他 CLI 配置关键字也会影响 IBM OLE DB Provider 的行为。

使用 Visual Basic ADO 应用程序连接至数据源

要使用 IBM OLE DB Provider for DB2, 必须指定 IBMDADB2 提供程序名称。

要使用 IBM OLE DB Provider for DB2 连接至 DB2 数据源, 请指定 IBMDADB2 提供程序名称。

ADO 应用程序中的可更新可滚动游标

要访问可更新可滚动游标的 ADO 应用程序, 可以将游标位置设置为 adUseClient 或 adUseServer。将游标位置设置为 adUseServer 将导致游标在服务器上具体化。

IBM OLE DB Provider for DB2 提供了对只读、只前进、只读可滚动和可更新可滚动游标的本机支持。要访问可更新可滚动游标的 ADO 应用程序, 可以将游标位置设置为 adUseClient 或 adUseServer。将游标位置设置为 adUseServer 将导致游标在服务器上具体化。

ADO 应用程序的限制

ADO 应用程序在调用存储过程以及使用服务器端可滚动游标插入新行方面均受到限制, 并且不支持缺省参数值。

ADO 应用程序的限制是:

- 调用了存储过程的 ADO 应用程序必须创建并显式地绑定其参数。DB2 Server for VSE & VM 不支持用于自动生成参数的 Parameters.Refresh 方法。
- 不支持缺省参数值。
- 使用服务器端可滚动游标来插入新行时, 请使用带有 Fieldlist 和 Values 自变量的 AddNew() 方法。这比在对每个列执行 Update() 调用后调用不带自变量的 AddNew() 效率高。每个 AddNew() 和 Update() 调用都是对服务器发出的不同请求, 因此, 其效率不如单一的 AddNew() 调用。
- 新插入的行不可通过服务器端可滚动游标进行更新。
- 使用服务器端可滚动游标时, 包含长整型数据或 LOB 数据的表不可更新。

IBM OLE DB Provider 对 ADO 方法和属性的支持

IBM OLE DB Provider 支持下列 ADO 方法和属性。

IBM OLE DB Provider 支持下列 ADO 方法和属性:

表 30. 命令方法

方法/属性	OLE DB 接口/属性	IBM OLE DB 支持
Cancel	ICommand	是
CreateParameter		是
Execute		是

表 31. 命令属性

方法/属性	OLE DB 接口/属性	IBM OLE DB 支持
ActiveConnection	(特定于 ADO)	
命令文本	ICommandText	是
命令超时	ICommandProperties::SetProperties DBPROP_COMMANDTIMEOUT	是
CommandType	(特定于 ADO)	
预编译	ICommandPrepare	是
省	(特定于 ADO)	

表 32. 命令集合

方法/属性	OLE DB 接口/属性	IBM OLE DB 支持
参数	ICommandWithParameter DBSCHEMA _PROCEDURE_PARAMETERS	是
属性	ICommandProperties IDBProperties	是

表 33. 连接方法

方法/属性	OLE DB 接口/属性	IBM OLE DB 支持
BeginTrans CommitTrans RollbackTrans	ITransactionLocal	是 (但不能嵌套) 是 (但不能嵌套) 是 (但不能嵌套)
Execute	ICommandIOpenRowset	是
Open	IDBCreateSessionIDBInitialize	是

表 33. 连接方法 (续)

方法/属性	OLE DB 接口/属性	IBM OLE DB 支持
OpenSchema adSchemaColumnPrivileges adSchemaColumns adSchemaForeignKeys adSchemaIndexes adSchemaPrimaryKeys adSchemaProcedureParam adSchemaProcedures adSchemaProviderType adSchemaStatistics adSchemaTablePrivileges adSchemaTables	IDBSchemaRowset	是是是是是是是是是
Cancel		是

表 34. 连接属性

方法/属性	OLE DB 接口/属性	IBM OLE DB 支持
属性 adXactCommitRetaining adXactRollbackRetaining	ITransactionLocal	是是
CommandTimeout	ICommandProperties DBPROP_COMMAND_TIMEOUT	是
ConnectionString	(特定于 ADO)	
ConnectionTimeout	IDBProperties DBPROP_INIT_TIMEOUT	否
CursorLocation: adUseClient adUseNone adUseServer	(使用 OLE DB 游标服务) (未使用)	是否是
DefaultDataBase	IDBProperties DBPROP_CURRENTCATALOG	否
IsolationLevel	ITransactionLocal DBPROP_SESS _AUTOCOMMITISOLEVELS	是

表 34. 连接属性 (续)

方法/属性	OLE DB 接口/属性	IBM OLE DB 支持
方式	IDBProperties	否
adModeRead	DBPROP_INIT_MODE	是
adModeReadWrite		否
adModeShareDenyNone		否
adModeShareDenyRead		否
adModeShareDenyWrite		否
adModeShareExclusive		否
adModeUnknown		否
adModeWrite		否
提供程序	ISourceRowset::GetSourceRowset	是
省	(特定于 ADO)	
版本	(特定于 ADO)	

表 35. 连接集合

方法/属性	OLE DB 接口/属性	IBM OLE DB 支持
错误	IErrorRecords	是
属性	IDBProperties	是

表 36. 错误属性

方法/属性	OLE DB 接口/属性	IBM OLE DB 支持
描述	IErrorRecords	是
NativeError		是
号码		是
源		是
SQLState		是
HelpContext		否
HelpFile		否

表 37. 字段方法

方法/属性	OLE DB 接口/属性	IBM OLE DB 支持
AppendChunk	ISequentialStream	是
GetChunk		是

表 38. 字段属性

方法/属性	OLE DB 接口/属性	IBM OLE DB 支持
实际大小	IAccessor IRowset	是

表 38. 字段属性 (续)

方法/属性	OLE DB 接口/属性	IBM OLE DB 支持
属性 DataFormat DefinedSize 名称 NumericScale Precision 类型	IColumnInfo	是 是 是 是 是 是
OriginalValue	IRowsetUpdate	是 (游标服务)
UnderlyingValue	IRowsetRefresh IRowsetResynch	是 (游标服务) 是 (游标服务)
值	IAccessor IRowset	是

表 39. 字段集合

方法/属性	OLE DB 接口/属性	IBM OLE DB 支持
属性	IDBProperties IRowsetInfo	是

表 40. 参数方法

方法/属性	OLE DB 接口/属性	IBM OLE DB 支持
AppendChunk	ISequentialStream	是
属性 Direction 名称 NumericScale Precision Scale Size 类型	ICommandWithParameter DBSCHEMA _PROCEDURE_PARAMETERS	是 否 是 是 是 是 是
值	IAccessor ICommand	是

表 41. Parameter 集合

方法/属性	OLE DB 接口/属性	IBM OLE DB 支持
属性		是

表 42. RecordSet 方法

方法/属性	OLE DB 接口/属性	IBM OLE DB 支持
AddNew	IRowsetChange	是
Cancel		是
CancelBatch	IRowsetUpdate::Undo	是 (游标服务)
CancelUpdate		是 (游标服务)
Clone	IRowsetLocate	是
Close	IAccessor IRowset	是
CompareBookmarks		否
Delete	IRowsetChange	是
GetRows	IAccessor IRowset	是
Move	IRowset IRowsetLocate	是
MoveFirst	IRowset IRowsetLocate	是
MoveNext	IRowset IRowsetLocate	是
MoveLast	IRowsetLocate	是
MovePrevious	IRowsetLocate	是
NextRecordSet	IMultipleResults	是
Open	ICommand IOpenRowset	是
Requery	ICommand IOpenRowset	是
Resync	IRowsetRefresh	是 (游标服务)
Supports	IRowsetInfo	是
更新 UpdateBatch	IRowsetChange IRowsetUpdate	是 是 (游标服务)

表 43. RecordSet 属性

方法/属性	OLE DB 接口/属性	IBM OLE DB 支持
AbsolutePage	IRowsetLocate IRowsetScroll	是 是 ¹
AbsolutePosition	IRowsetLocate IRowsetScroll	是 是 ¹

表 43. RecordSet 属性 (续)

方法/属性	OLE DB 接口/属性	IBM OLE DB 支持
ActiveConnection	IDBCreateSession IDBInitialize	是
BOF	(特定于 ADO)	
Bookmark	IAccessor IRowsetLocate	是
CacheSize	IRowsetLocate 中的 cRows IRowset	是
CursorType adOpenDynamic adOpenForwardOnly adOpenKeySet adOpenStatic	ICommandProperties	否 是 是 是
EditMode	IRowsetUpdate	是 (游标服务)
EOF	(特定于 ADO)	
Filter	IRowsetLocate IRowsetView IRowsetUpdate IViewChapter IViewFilter	否
LockType	ICommandProperties	是
MarshalOption		否
MaxRecords	ICommandProperties IOpenRowset	是
PageCount	IRowsetScroll	是 ¹
PageSize	(特定于 ADO)	
Sort	(特定于 ADO)	
Source	(特定于 ADO)	
State	(特定于 ADO)	
状态	IRowsetUpdate	是 (游标服务)
注:		
1. 要返回的值是近似值。将不会跳过已删除的行。		

表 44. RecordSet 集合

方法/属性	OLE DB 接口/属性	IBM OLE DB 支持
字段	IColumnInfo	是
属性	IDBProperties IRowsetInfo::GetProperties	是

C/C++ 应用程序的编译和链接以及 IBM OLE DB Provider

使用了常量 CLSID_IBMDADB2 的 C/C++ 应用程序必须包括 SQLLIB\include 目录中的 ibmdadb2.h 文件。

使用了常量 CLSID_IBMDADB2 的 C/C++ 应用程序必须包括 SQLLIB\include 目录中的 ibmdadb2.h 文件。这些应用程序必须在 include 语句前定义 DBINITCONSTANTS。以下示例说明 C/C++ 预处理器伪指令的正确顺序:

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

在 C/C++ 应用程序中使用 IBM OLE DB Provider 来连接到数据源

要在 C/C++ 应用程序中使用 IBM OLE DB Provider for DB2 来连接到 DB2 数据源, 您可以使用两个 OLE DB 核心接口 (IDBPromptInitialize 或 IDataInitialize) 中的一个, 也可以调用 COM API CoCreateInstance。

要在 C/C++ 应用程序中使用 IBM OLE DB Provider for DB2 来连接到 DB2 数据源, 您可以使用两个 OLE DB 核心接口 (IDBPromptInitialize 或 IDataInitialize) 中的一个, 也可以调用 COM API CoCreateInstance。IDataInitialize 接口由 OLE DB 服务组件提供, 而 IDBPromptInitialize 由数据链路组件提供。

COM+ 分布式事务支持和 IBM OLE DB Provider

在 Windows 2000 或 XP 上的 Microsoft Component Services (COM+) 环境中运行的 OLE DB 应用程序可以使用 ITransactionJoin 接口。

在 Windows 2000 或 XP 上的 Microsoft Component Services (COM+) 环境中运行的 OLE DB 应用程序可以使用 ITransactionJoin 接口来加入与多个 DB2 for Linux, UNIX, and Windows、主机和 System i[®] 数据库服务器以及其他符合 COM+ 规范的资源管理器进行的分布式事务。

先决条件

要使用 IBM OLE DB Provider for DB2 所提供的 COM+ 分布式事务支持, 请确保服务器符合下列先决条件。

注: 这些要求仅适用于那些基于 Windows 并安装了 DB2 客户机的计算机。

- 带有 Service Pack 3 或更高版本的 Windows 2000
- Windows XP

在 C/C++ 数据库应用程序中启用 COM+ 支持

要以 COM+ 事务方式运行 C 或 C++ 应用程序, 您可以使用 CoCreateInstance 来创建 IBMDADB2 数据源实例, 获取会话对象, 然后使用 JoinTransaction。

要以 COM+ 事务方式运行 C 或 C++ 应用程序, 您可以使用 CoCreateInstance 来创建 IBMDADB2 数据源实例, 获取会话对象, 然后使用 JoinTransaction。有关更多信息, 请参阅有关如何将 C 或 C++ 应用程序连接到数据源的描述。

要以 COM+ 事务方式来运行 ADO 应用程序，请参阅有关如何将 C 或 C++ 应用程序连接到数据源 的描述。

要以事务方式使用 COM+ 软件包中的组件，请将该组件的 Transactions 属性设置为下列其中一个值：

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

有关这些值的信息，请参阅 COM+ 文档。

第 3 章 IBM Data Server Provider for .NET

IBM Data Server Provider for .NET 扩展了对 ADO.NET 接口的支持，并提供对 IBM 数据服务器的高性能安全访问。

IBM Data Server Provider for .NET 允许 .NET 应用程序访问下列数据库管理系统：

- DB2 for Linux, UNIX, and Windows V9.1、V9.5、V9.7 和 V9.8
- DB2 通用数据库 V8 for Windows, UNIX, and Linux
- DB2 for z/OS V8、V9 和 V10, 通过 DB2 Connect
- DB2 Universal Database for AS/400 V5R3 及更高版本（通过 DB2 Connect）
- IBM Informix V11.10 和 V11.50

除了 IBM Data Server Provider for .NET 以外，IBM Database Add-Ins for Visual Studio 还使您能够快速轻松地 为 Visual Studio 中的数据库开发 .NET 应用程序。还可以使用该加载件来创建数据库对象（例如，索引和表）以及开发服务器端对象（例如，存储过程和用户定义的函数）。

IBM Data Server Provider for .NET 的系统需求

在使用 DB2 安装程序安装 IBM Data Server Provider for .NET 之前，必须已在计算机上安装 .NET Framework。如果未安装 .NET Framework，那么 DB2 安装程序将不会安装 IBM Data Server Provider for .NET。

对于 DB2 Universal Database for AS/400 and System i，需要对服务器安装以下修订：APAR ii13348。

使用 IBM Data Server Provider for .NET 文档

IBM Data Server Provider for .NET 参考文档中包含特定于提供程序的详细信息。许多 IBM Data Server Provider for .NET 类从其他 .NET Framework 类或接口继承或实现成员。提供程序文档包括这些类中每个类的所有成员的摘要。有关所继承的特定成员的更详细信息，请参阅 Microsoft .NET Framework SDK 中的相应主题。

名称空间

名称空间	描述
IBM.Data.DB2	IBM 数据服务器的 .NET Framework 数据提供程序。
IBM.Data.DB2Types	为 IBM 数据服务器中的本机数据类型提供类。
IBM.Data.Informix	Informix 数据服务器的 .NET Framework 数据提供程序。

IBM.Data.DB2 名称空间

IBM.Data.DB2 名称空间是 IBM 数据服务器的 .NET Framework 数据提供程序。IBM Data Server Provider for .NET 扩展了对 ADO.NET 接口的支持，并提供了对数据的高性能安全访问。有时候将此提供程序称为 Common DB2 .NET Provider。

IBM Data Server Provider for .NET 允许 .NET 应用程序访问下列数据库管理系统：

- DB2 for Linux, UNIX, and Windows V9.1、V9.5、V9.7 和 V9.8
- DB2 for z/OS V8、V9 和 V10，通过 DB2 Connect
- DB2 Universal Database for AS/400 V5R3 及更高版本（通过 DB2 Connect）

除了 IBM Data Server Provider for .NET 以外，可使用 IBM Database Add-Ins for Visual Studio 快速轻松地 为 Visual Studio 中的 IBM 数据服务器开发 .NET 应用程序。还可以使用该加载件来创建数据库对象（例如，索引和表）以及开发服务器端对象（例如，存储过程和用户定义的函数）。

IBM Data Server Provider for .NET 的系统需求

在使用 DB2 安装程序安装 IBM Data Server Provider for .NET 之前，必须在计算机上已经安装了 .NET Framework。如果未安装 .NET Framework，那么 DB2 安装程序将不会安装 IBM Data Server Provider for .NET。

对于 DB2 for i V5R4 和更早的版本，服务器上需要安装以下修订：APAR ii13348。

使用 IBM Data Server Provider for .NET 文档

IBM Data Server Provider for .NET 参考文档中包含特定于提供程序的详细信息。许多 IBM Data Server Provider for .NET 类从其他 .NET Framework 类或接口继承或实现成员。提供程序文档包括这些类中每个类的所有成员的摘要。有关所继承的特定成员的更详细信息，请参阅 Microsoft .NET Framework SDK 中的相应主题。

开发 IBM Data Server Provider for .NET 应用程序

IBM Data Server Provider for .NET 提供了连接至数据库、执行命令和检索结果的功能。当处于断开连接状态时，可以直接处理这些结果，也可以将它们放置在 ADO.NET DataSet 中以供将来处理。在 DataSet 中时，数据可显示给用户、与来自多个源的其他数据组合或以远程方式在各层之间传递。然后，可以将 DataSet 中对数据执行的任何处理协调至数据库。

要使用 IBM Data Server Provider for .NET，必须将针对 IBM.Data.DB2 名称空间的 imports 或 using 语句添加至应用程序 .DLL。例如：

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

还必须在应用程序的项目中包括对 IBM.Data.DB2.dll 的引用。在 Visual Studio 中，可以在解决方案资源管理器中从项目的“引用”部分完成此操作，也可以使用 项目 > 添加引用 菜单选项来完成此操作。如果要从 Framework 2.0 或更高版本的命令行编译程序，那么 **csc** 或 **vbc** 命令还应包括以下选项：















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

要与其中一个受支持的数据服务器建立连接，将需要构造 DB2Connection 对象，并向其提供有效的 DB2 .NET 连接字符串。请参阅 DB2Connection.ConnectionString 属性，以了解有关受支持的关键字的信息。

有关如何以最佳方式使用 IBM.Data.DB2 名称空间的详细信息，请参阅有关下列 IBM Data Server Provider for .NET 类的文档：





- DB2DataAdapter
- DB2Command
- DB2Connection
- DB2DataReader

类


类	描述
 DB2BulkCopy	便于将一个数据源中的许多行复制到另一个数据源。
 DB2BulkCopyColumnMapping	表示从数据源表到目标表的列映射。
 DB2BulkCopyColumnMappingCollection	表示从数据源表到目标表的列映射的集合。
 DB2Command	表示要对数据库执行的 SQL 语句或存储过程。
 DB2CommandBuilder	自动生成用来使对 DataSet 所作的更改与相关联的数据库一致的单个表命令。
 DB2Connection	表示与数据库的打开的连接。
 DB2ConnectionStringBuilder	便于通过使用一般方法和特定于 IBM Data Server Provider for .NET 的方法来生成有效的连接字符串。
 DB2DataAdapter	表示用来填充 DataSet 和更新数据库的一组数据命令和数据库连接。
 DB2DataReader	提供了一种从数据库读取仅正向数据行流的方法。
 DB2DataSourceEnumerator	提供一种发现可视 IBM 系列数据源的方法。
 DB2Error	收集与数据库返回的警告或错误相关的信息。
 DB2ErrorCollection	收集由 DB2DataAdapter 生成的所有错误。
 DB2Exception	IBM 数据服务器返回错误时将生成的异常。
 DB2Factory	表示一组为数据提供程序创建 System.Data.Common 数据源类的实例的方法。
 DB2InfoMessageEventArgs	为 InfoMessage 事件提供数据。

类	描述
 DB2Parameter	表示 DB2Command 的参数以及其到 DataColumn 的映射（可选）。
 DB2ParameterCollection	表示与 DB2Command 相关的参数集合以及它们到 DataSet 中列的映射。
 DB2Permission	使 IBM Data Server Provider for .NET 能够确保用户具有足够的安全性级别来访问 IBM 数据库。
 DB2PermissionAttribute	使安全性操作与定制安全性属性相关联。
 DB2Record	表示一条只读记录。
 DB2ResultSet	提供一种从数据库多向浏览数据行的可绑定流的方法。此外，DB2ResultSet 实例还可以更新、删除和插入行。
 DB2RowsCopiedEventArgs	为 DB2RowsCopied 事件提供数据。
 DB2RowUpdatedEventArgs	为 RowUpdated 事件提供数据。
 DB2RowUpdatingEventArgs	为 RowUpdating 事件提供数据。
 DB2Transaction	表示要在数据库中执行的 SQL 事务。
 DB2UpdatableRecord	表示要在 DB2ResultSet 实例中创建的新行。
 DB2XmlAdapter	表示一种从数据库检索并填充 XPathDocument 实例的方法。

指派

代表	描述
 DB2InfoMessageEventHandler	表示将处理 DB2Connection 对象的 InfoMessage 事件的方法。
 DB2RowsCopiedEventHandler	表示将处理 DB2BulkCopy 对象的 DB2RowsCopied 事件的方法。
 DB2RowUpdatedEventHandler	表示将处理 DB2DataAdapter 对象的 RowUpdated 事件的方法。
 DB2RowUpdatingEventHandler	表示将处理 DB2DataAdapter 对象的 RowUpdating 事件的方法。

枚举

枚举	描述
 DB2BulkCopyOptions	指定要与 DB2BulkCopy 一起使用的选项。这些选项是位标志，使您能够将它们组合成位掩码。

枚举	描述
 DB2CursorType	指定要与 DB2ResultSet 一起使用的游标选项。这些选项是位标志，使您能够将它们组合成位掩码。
 DB2ResultSetOptions	指定要与 DB2ResultSet 一起使用的结果集选项。这些选项是位标志，使您能够将它们组合成位掩码。
 DB2Type	指定字段、属性或 DB2Parameter 的数据类型。

参考

『 IBM.Data.DB2Types 名称空间 』

IBM.Data.DB2Types 名称空间提供用于表示 .NET Framework Data Provider 的 DB2 数据类型的类和结构。

第 69 页的第 3 章, 『 IBM Data Server Provider for .NET 』

IBM Data Server Provider for .NET 扩展了对 ADO.NET 接口的支持, 并提供对 IBM 数据服务器的高性能安全访问。

IBM.Data.DB2Types 名称空间

IBM.Data.DB2Types 名称空间提供用于表示 .NET Framework Data Provider 的 DB2 数据类型的类和结构。

下表显示 DB2Type 数据类型、DB2 数据类型、 Informix 数据类型、 Microsoft .NET Framework 类型与 DB2Types 类和结构之间的映射。

表 45. 数据类型、类和结构之间的映射

类别	DB2Types 类和结构	DB2Type 数据类型	DB2 数据类型	Informix 数据类型	.NET 数据类型
数字	DB2Int16	SmallInt	SMALLINT	BOOLEAN 和 SMALLINT	Int16
	DB2Int32	Integer	INT	INTEGER、INT 和 SERIAL	Int32
	DB2Int64	BigInt 和 BigSerial	BIGINT	BIGINT、BIGSERIAL、INT8 和 SERIAL8	Int64
	DB2Real, DB2Real370	Real	REAL	REAL 和 SMALLFLOAT	Single
	DB2Double	Double	DOUBLE PRECISION	DECIMAL (≦31) 和 DOUBLE PRECISION	Double
	DB2Double	Float	FLOAT	DECIMAL (32) 和 FLOAT	Double
	DB2Decimal	Decimal	DECIMAL	MONEY	Decimal
	DB2Decimal Float	DecimalFloat	DECFLOAT (16 34)		Decimal
	DB2Decimal	Numeric	DECIMAL	DECIMAL (≦31) 和 NUMERIC	Decimal

6. DB2 .NET 公共语言运行时例程不支持将这些数据类型用作参数。

7. 类型 DB2Type.Xml 的 DB2ParameterClass.ParameterName 属性可以接受下列类型的变量: String、byte[]、DB2Xml 和 XmlReader。


8. 这些数据类型只适用于 DB2 for z/OS。

9. 只有 DB2 for z/OS V9 及更高发行版和 DB2 for Linux, UNIX, and Windows V9.5 及更高发行版才支持此数据类型。

表 45. 数据类型、类和结构之间的映射 (续)

类别	DB2Types 类和结构	DB2Type 数据类型	DB2 数据类型	Informix 数据类型	.NET 数据类型
日期/时间	DB2Date	Date	DATE	DATETIME (日期精度)	Datetime
	DB2Time	Time	TIME	DATETIME (时间精度)	TimeSpan
	DB2TimeStamp	Timestamp	TIMESTAMP	DATETIME (时间和日期精度)	DateTime
	DB2TimeStampOffset	Timestamp WithTimeZone	TIMESTAMP WITH TIME ZONE	不适用	DateTimeOffset
XML	DB2Xml	XmlIxfType.Xml	XML		Byte[]
字符数据	DB2String	Char	CHAR	CHAR	String
	DB2String	VarChar	VARCHAR	VARCHAR	String
	DB2String	LongVarChar	LONG VARCHAR	LVARCHAR	String
二进制数据	DB2Binary	Binary	CHAR FOR BIT DATA		Byte[]
	DB2Binary	Binary	BINARY		Byte[]
	DB2Binary	VarBinary	VARBINARY		Byte[]
	DB2Binary	Long VarBinary	LONG VARCHAR FOR BIT DATA		Byte[]
图形数据	DB2String	Graphic	GRAPHIC		String
	DB2String	VarGraphic	VARGRAPHIC		String
	DB2String	Long VarGraphic	LONG VARGRAPHIC		String
LOB 数据	DB2Clob	Clob	CLOB	CLOB 和 TEXT	String
	DB2Blob	Blob	BLOB	BLOB 和 BYTE	Byte[]
	DB2Clob	DbClob	DBCLOB		String
行标识	DB2RowId	RowId	ROWID		Byte[]

枚举

枚举	描述
 DB2RoundingMode	指定要对 DB2DecimalFloat 对象使用的舍入方式。

附录 A. DB2 技术信息概述

DB2 技术信息以多种可以通过多种方法访问的格式提供。

您可以通过下列工具和方法获得 DB2 技术信息:

- DB2 信息中心
 - 主题（任务、概念和参考主题）
 - 样本程序
 - 教程
- DB2 书籍
 - PDF 文件（可下载）
 - PDF 文件（在 DB2 PDF DVD 中）
 - 印刷版书籍
- 命令行帮助
 - 命令帮助
 - 消息帮助

注: DB2 信息中心主题的更新频率比 PDF 书籍或硬拷贝书籍的更新频率高。要获取最新信息, 请安装可用的文档更新或者参阅 ibm.com 上的 DB2 信息中心。

您可以在线访问 ibm.com 上的其他 DB2 技术信息, 例如技术说明、白皮书和 IBM Redbooks® 出版物。请访问以下网址处的 DB2 信息管理软件资料库站点: <http://www.ibm.com/software/data/sw-library/>。

文档反馈

我们非常重视您对 DB2 文档的反馈。如果您想就如何改善 DB2 文档提出建议, 请向 db2docs@ca.ibm.com 发送电子邮件。DB2 文档小组将阅读您的所有反馈, 但无法直接给您答复。请尽可能提供具体的示例, 这样我们才能更好地了解您所关心的问题。如果您要提供有关具体主题或帮助文件的反馈, 请加上标题和 URL。

请不要使用以上电子邮件地址与 DB2 客户支持机构联系。如果您遇到文档无法解决的 DB2 技术问题, 请与您当地的 IBM 服务中心联系以获得帮助。

硬拷贝或 PDF 格式的 DB2 技术库

下列各表描述 IBM 出版物中心（网址为 www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss）所提供的 DB2 资料库。可从 www.ibm.com/support/docview.wss?rs=71&uid=swg2700947 下载 PDF 格式的 DB2 V10.1 手册的英文版本和翻译版本。

尽管这些表标识书籍有印刷版, 但可能未在您所在国家或地区提供。

每次更新手册时, 表单号都会递增。确保您正在阅读下面列示的手册的最新版本。

注: DB2 信息中心的更新频率比 PDF 或硬拷贝书籍的更新频率高。

表 46. DB2 技术信息

书名	书号	是否提供印刷版	最近一次更新时间
<i>Administrative API Reference</i>	SC27-3864-00	是	2012 年 4 月
<i>Administrative Routines and Views</i>	SC27-3865-01	否	2013 年 1 月
<i>Call Level Interface Guide and Reference Volume 1</i>	SC27-3866-01	是	2013 年 1 月
<i>Call Level Interface Guide and Reference Volume 2</i>	SC27-3867-01	是	2013 年 1 月
<i>Command Reference</i>	SC27-3868-01	是	2013 年 1 月
数据库管理概念和配置参考	S151-1758-01	是	2013 年 1 月
数据移动实用程序指南和参考	S151-1756-01	是	2013 年 1 月
数据库监视指南和参考	S151-1759-01	是	2013 年 1 月
数据恢复及高可用性指南与参考	S151-1755-01	是	2013 年 1 月
数据库安全性指南	S151-1753-02	是	2013 年 1 月
<i>DB2 Workload Management Guide and Reference</i>	SC27-3891-01	是	2013 年 1 月
开发 ADO.NET 和 OLE DB 应用程序	S151-1765-01	是	2013 年 1 月
开发嵌入式 SQL 应用程序	S151-1763-01	是	2013 年 1 月
<i>Developing Java Applications</i>	SC27-3875-01	是	2013 年 1 月
<i>Developing Perl, PHP, Python, and Ruby on Rails Applications</i>	SC27-3876-00	否	2012 年 4 月
<i>Developing RDF Applications for IBM Data Servers</i>	SC27-4462-00	是	2013 年 1 月
开发用户定义的例程 (SQL 和外部例程)	S151-1761-01	是	2013 年 1 月
数据库应用程序开发入门	G151-1764-01	是	2013 年 1 月
Linux 和 Windows 上的 DB2 安装和管理入门	G151-1769-00	是	2012 年 4 月
全球化指南	S151-1757-00	是	2012 年 4 月
安装 DB2 服务器	G151-1768-01	是	2013 年 1 月
安装 IBM Data Server Client	G151-1751-00	否	2012 年 4 月
消息参考第 1 卷	S151-1767-01	否	2013 年 1 月

表 46. DB2 技术信息 (续)

书名	书号	是否提供印刷版	最近一次更新时间
消息参考第 2 卷	S151-1766-01	否	2013 年 1 月
<i>Net Search Extender</i> 管理和用户指南	S151-1905-01	否	2013 年 1 月
分区和集群指南	S151-1754-01	是	2013 年 1 月
<i>Preparation Guide for DB2 10.1 Fundamentals Exam 610</i>	SC27-4540-00	否	2013 年 1 月
<i>Preparation Guide for DB2 10.1 DBA for Linux, UNIX, and Windows Exam 611</i>	SC27-4541-00	否	2013 年 1 月
<i>pureXML</i> 指南	S151-1775-01	是	2013 年 1 月
<i>Spatial Extender User's Guide and Reference</i>	SC27-3894-00	否	2012 年 4 月
SQL 过程语言: 应用程序启用和支持	S151-1762-01	是	2013 年 1 月
<i>SQL Reference Volume 1</i>	SC27-3885-01	是	2013 年 1 月
<i>SQL Reference Volume 2</i>	SC27-3886-01	是	2013 年 1 月
<i>Text Search Guide</i>	SC27-3888-01	是	2013 年 1 月
故障诊断和调整数据库性能	S151-1760-01	是	2013 年 1 月
升级到 DB2 V10.1	S151-1770-01	是	2013 年 1 月
DB2 V10.1 新增内容	S151-1752-01	是	2013 年 1 月
XQuery 参考	S151-1774-01	否	2013 年 1 月

表 47. 特定于 DB2 Connect 的技术信息

书名	书号	是否提供印刷版	最近一次更新时间
DB2 Connect 安装和配置 DB2 Connect Personal Edition	S151-1773-00	是	2012 年 4 月
DB2 Connect 安装和配置 DB2 Connect 服务器	S151-1772-01	是	2013 年 1 月
DB2 Connect 用户指南	S151-1771-01	是	2013 年 1 月

从命令行处理器显示 SQL 状态帮助

DB2 产品针对可能充当 SQL 语句结果的条件返回 SQLSTATE 值。SQLSTATE 帮助说明 SQL 状态和 SQL 状态类代码的含义。

过程

要启动 SQL 状态帮助, 请打开命令行处理器并输入:

```
? sqlstate or ? class code
```

其中, *sqlstate* 表示有效的 5 位 SQL 状态, *class code* 表示该 SQL 状态的前 2 位。例如, ? 08003 显示 08003 SQL 状态的帮助, 而 ? 08 显示 08 类代码的帮助。

访问不同版本的 DB2 信息中心

您可以在 ibm.com[®] 上的不同信息中心中找到其他版本 DB2 产品的文档。

关于此任务

对于 DB2 V10.1 主题, *DB2 信息中心* URL 是 <http://publib.boulder.ibm.com/infocenter/db2luw/v10r1>。

对于 DB2 V9.8 主题, *DB2 信息中心* URL 是 <http://publib.boulder.ibm.com/infocenter/db2luw/v9r8/>。

对于 DB2 V9.7 主题, *DB2 信息中心* URL 是 <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/>。

对于 DB2 V9.5 主题, *DB2 信息中心* URL 是 <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5>。

对于 DB2 V9.1 主题, *DB2 信息中心* URL 是 <http://publib.boulder.ibm.com/infocenter/db2luw/v9/>。

对于 DB2 V8 主题, 请转至 *DB2 信息中心* URL: <http://publib.boulder.ibm.com/infocenter/db2luw/v8/>。

更新安装在计算机或内部网服务器上的 DB2 信息中心

安装在本地的 DB2 信息中心必须定期进行更新。

开始之前

必须已安装 DB2 V10.1 信息中心。有关详细信息, 请参阅安装 DB2 服务器中的“使用 DB2 安装向导来安装 DB2 信息中心”主题。所有适用于安装信息中心的先决条件和限制同样适用于更新信息中心。

关于此任务

可以自动或手动更新现有的 DB2 信息中心:

- 自动更新将更新现有的信息中心功能部件和语言。自动更新的一个优点是, 与手动更新相比, 信息中心的不可用时间较短。另外, 自动更新可设置为作为定期运行的其他批处理作业的一部分运行。
- 可以使用手动更新方法来更新现有的信息中心功能部件和语言。自动更新可以缩短更新过程中的停机时间, 但如果您想添加功能部件或语言, 那么必须执行手动过程。例如, 如果本地信息中心最初安装的是英语和法语版, 而现在还要安装德语版; 那么手动更新将安装德语版, 并更新现有信息中心的功能和语言。但是, 手动更新要求您手动停止、更新和重新启动信息中心。在整个更新过程期间信息中心不可用。在自动更新过程中, 信息中心仅在更新完成后停止工作以重新启动信息中心。

此主题详细说明了自动更新的过程。有关手动更新的指示信息，请参阅“手动更新安装在您的计算机或内部网服务器上的 DB2 信息中心”主题。

过程

要自动更新安装在计算机或内部网服务器上的 DB2 信息中心：

1. 在 Linux 操作系统上，
 - a. 浏览至信息中心的安装位置。缺省情况下，DB2 信息中心安装在 `/opt/ibm/db2ic/V10.1` 目录中。
 - b. 从安装目录浏览至 `doc/bin` 目录。
 - c. 运行 `update-ic` 脚本：

```
update-ic
```
2. 在 Windows 操作系统上，
 - a. 打开命令窗口。
 - b. 浏览至信息中心的安装位置。缺省情况下，DB2 信息中心安装在 `<Program Files>\IBM\DB2 Information Center\V10.1` 目录中，其中 `<Program Files>` 表示 `Program Files` 目录的位置。
 - c. 从安装目录浏览至 `doc\bin` 目录。
 - d. 运行 `update-ic.bat` 文件：

```
update-ic.bat
```

结果

DB2 信息中心将自动重新启动。如果更新可用，那么信息中心会显示新的以及更新后的主题。如果信息中心更新不可用，那么会在日志中添加消息。日志文件位于 `doc\eclipse\configuration` 目录中。日志文件名称是随机生成的编号。例如，`1239053440785.log`。

手动更新安装在计算机或内部网服务器上的 DB2 信息中心

如果您已在本地安装 DB2 信息中心，那么可从 IBM 获取文档更新并进行安装。

关于此任务

手动更新安装在本地的 DB2 信息中心要求您：

1. 停止计算机上的 DB2 信息中心，然后以独立方式重新启动信息中心。如果以独立方式运行信息中心，那么网络上的其他用户将无法访问信息中心，因而您可以应用更新。DB2 信息中心的工作站版本总是以独立方式运行。
2. 使用“更新”功能部件来查看可用的更新。如果有您必须安装的更新，那么请使用“更新”功能部件来获取并安装这些更新。

注：如果您的环境要求在一台未连接至因特网的机器上安装 DB2 信息中心更新，请使用一台已连接至因特网并已安装 DB2 信息中心的机器将更新站点镜像至本地文件系统。如果网络中有许多用户将安装文档更新，那么可以通过在本地也为更新站点制作镜像并为更新站点创建代理来缩短每个人执行更新所需要的时间。如果提供了更新包，请使用“更新”功能部件来获取这些更新包。但是，只有在单机方式下才能使用“更新”功能部件。

3. 停止独立信息中心，然后在计算机上重新启动 *DB2 信息中心*。

注：在 Windows 2008、Windows Vista 和更高版本上，稍后列示在此部分的命令必须作为管理员运行。要打开具有全面管理员特权的命令提示符或图形工具，请右键单击快捷方式，然后选择**以管理员身份运行**。

过程

要更新安装在您的计算机或内部网服务器上的 *DB2 信息中心*：


1. 停止 *DB2 信息中心*。
 - 在 Windows 上，单击**开始** > **控制面板** > **管理工具** > **服务**。右键单击 **DB2 信息中心** 服务，并选择**停止**。
 - 在 Linux 上，输入以下命令：

```
/etc/init.d/db2icdv10 stop
```
2. 以独立方式启动信息中心。
 - 在 Windows 上：
 - a. 打开命令窗口。
 - b. 浏览至信息中心的安装位置。缺省情况下，*DB2 信息中心*安装在 *Program Files\IBM\DB2 Information Center\V10.1* 目录中，其中 *Program Files* 表示 Program Files 目录的位置。
 - c. 从安装目录浏览至 *doc\bin* 目录。
 - d. 运行 *help_start.bat* 文件：

```
help_start.bat
```
 - 在 Linux 上：
 - a. 浏览至信息中心的安装位置。缺省情况下，*DB2 信息中心*安装在 */opt/ibm/db2ic/V10.1* 目录中。
 - b. 从安装目录浏览至 *doc/bin* 目录。
 - c. 运行 *help_start* 脚本：

```
help_start
```

系统缺省 Web 浏览器将打开以显示独立信息中心。

3. 单击**更新按钮** ()。（必须在浏览器中启用 JavaScript。）在信息中心的右边面板上，单击**查找更新**。将显示现有文档的更新列表。
4. 要启动安装过程，请检查您要安装的选项，然后单击**安装更新**。
5. 在安装进程完成后，请单击**完成**。
6. 要停止独立信息中心，请执行下列操作：
 - 在 Windows 上，浏览至安装目录中的 *doc\bin* 目录并运行 *help_end.bat* 文件：

```
help_end.bat
```

注：*help_end* 批处理文件包含安全地停止使用 *help_start* 批处理文件启动的进程所需的命令。不要使用 Ctrl-C 或任何其他方法来停止 *help_start.bat*。
 - 在 Linux 上，浏览至安装目录中的 *doc/bin* 目录并运行 *help_end* 脚本：

```
help_end
```

注: help_end 脚本包含安全地停止使用 help_start 脚本启动的进程所需的命令。不要使用任何其他方法来停止 help_start 脚本。

7. 重新启动 DB2 信息中心。

- 在 Windows 上, 单击开始 > 控制面板 > 管理工具 > 服务。右键单击 **DB2 信息中心** 服务, 并选择启动。
- 在 Linux 上, 输入以下命令:

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

结果

更新后的 DB2 信息中心将显示新的以及更新后的主题。

DB2 教程

DB2 教程帮助您了解 DB2 数据库产品的各个方面。这些课程提供了逐步指示信息。

开始之前

您可以在信息中心中查看 XHTML 版的教程: <http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/>。

某些课程使用了样本数据或代码。有关其特定任务的任何先决条件的描述, 请参阅教程。

DB2 教程

要查看教程, 请单击标题。

*pureXML 指南*中的『**pureXML**®』

设置 DB2 数据库以存储 XML 数据以及对本机 XML 数据存储器执行基本操作。

DB2 故障诊断信息

我们提供了各种各样的故障诊断和问题确定信息来帮助您使用 DB2 数据库产品。

DB2 文档

您可以在《故障诊断和调整数据库性能》或者 DB2 信息中心的“数据库基础”部分中找到故障诊断信息, 这些信息包含以下内容:

- 有关如何使用 DB2 诊断工具和实用程序来隔离和确定问题的信息。
- 一些最常见问题的解决方案。
- 旨在帮助您解决 DB2 数据库产品使用过程中可能会遇到的其他问题的建议。

IBM Support Portal

如果您遇到问题并且希望得到帮助以查找可能的原因和解决方案, 请访问 IBM Support Portal。这个技术支持站点提供了指向最新 DB2 出版物、技术说明、授权程序分析报告 (APAR 或错误修订)、修订包和其他资源的链接。可搜索此知识库并查找问题的可能解决方案。

访问 IBM 支持门户网站网址为: http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/DB2_for_Linux,_UNIX_and_Windows

信息中心条款和条件

如果符合以下条款和条件，那么授予您使用这些出版物的许可权。

适用性: 用户需要遵循 IBM Web 站点的使用条款及以下条款和条件。

个人使用: 只要保留所有的专有权声明，您就可以为个人、非商业使用复制这些出版物。未经 IBM 明确同意，您不可以分发、展示或制作这些出版物或其中任何部分的演绎作品。

商业使用: 只要保留所有的专有权声明，您就可以仅在企业内复制、分发和展示这些出版物。未经 IBM 明确同意，您不可以制作这些出版物的演绎作品，或者在您的企业外部复制、分发或展示这些出版物或其中的任何部分。

权利: 除非本许可权中明确授予，否则不得授予对这些出版物或其中包含的任何信息、数据、软件或其他知识产权的任何许可权、许可证或权利，无论是明示的还是暗含的。

IBM 保留根据自身的判断，认为对出版物的使用损害了 IBM 的权益（由 IBM 自身确定）或未正确遵循以上指示信息时，撤回此处所授予权限的权利。

只有您完全遵循所有适用的法律和法规，包括所有的美国出口法律和法规，您才可以下载、出口或再出口该信息。

IBM 对这些出版物的内容不作任何保证。这些出版物“按现状”提供，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的关于适销和适用于某种特定用途的保证。

IBM 商标: IBM、IBM 徽标和 ibm.com 是 International Business Machines Corp.，在全球许多管辖区域注册的商标或注册商标。其他产品和服务名称可能是 IBM 或其他公司的商标。当前的 IBM 商标列表，可从 Web 站点 www.ibm.com/legal/copytrade.shtml 获得

附录 B. 声明

本信息是为在美国提供的产品和服务编写的。有关非 IBM 产品的信息是基于首次出版此文档时的可获信息且会随时更新。

IBM 可能在其他国家或地区不提供本文中讨论的产品、服务或功能特性。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务，则由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以用书面方式将许可查询寄往：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

有关双字节字符集 (DBCS) 信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区： International Business Machines Corporation“按现状”提供本出版物，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的有关非侵权、适销和适用于某种特定用途的保证。某些国家或地区在某些交易中不允许免除明示或暗含的保证。因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些更改将编入本资料的新版本中。IBM 可以随时对本资料中描述的产品和/或程序进行改进和/或更改，而不另行通知。

本信息中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。那些 Web 站点中的资料不是此 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果了解有关程序的信息以达到如下目的: (i) 允许在独立创建的程序和其他程序 (包括本程序) 之间进行信息交换, 以及 (ii) 允许对已经交换的信息进行相互使用, 请与下列地址联系:

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

只要遵守适当的条款和条件, 包括某些情形下的一定数量的付费, 都可获得这方面的信息。

本资料中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际软件许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。因此, 在其他操作环境中获得的数据可能会有明显的不同。有些测量可能是在开发级的系统上进行的, 因此不保证与一般可用系统上进行的测量结果相同。此外, 有些测量是通过推算而估计的, 实际结果可能会有差异。本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行测试, 也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回, 而不另行通知, 它们仅仅表示了目标和意愿而已。

本信息可能包含在日常业务操作中使用的数据和报告的示例。为了尽可能完整地说明这些示例, 示例中可能会包括个人、公司、品牌和产品的名称。所有这些名称都是虚构的, 与实际商业企业所用的名称和地址的任何雷同纯属巧合。

版权许可:

本信息包括源语言形式的样本应用程序, 这些样本说明不同操作平台上的编程方法。如果是为按照在编写样本程序的操作平台上的应用程序编程接口 (API) 进行应用程序的开发、使用、经销或分发, 您可以任何形式对这些样本程序进行复制、修改、分发, 而无须向 IBM 付费。这些示例并未在所有条件下作全面测试。因此, IBM 不能担保或暗示这些程序的可靠性、可维护性或功能。此样本程序“按现状”提供, 且不附有任何种类的保证。对于使用此样本程序所引起的任何损坏, IBM 将不承担责任。

凡这些样本程序的每份拷贝或其任何部分或任何衍生产品, 都必须包括如下版权声明:

© (your company name) (year). 此部分代码是根据 IBM 公司的样本程序衍生出来的。
© Copyright IBM Corp. *_enter the year or years_*. All rights reserved.

商标

IBM 商标: IBM、IBM 徽标和 ibm.com 是 International Business Machines Corp., 在全球许多管辖区域注册的商标或注册商标。其他产品和服务名称可能是 IBM 或其他公

司的商标。当前的 IBM 商标列表，可从 Web 站点 www.ibm.com/legal/copytrade.shtml 上“版权和商标信息”部分获取。

下列各项是其他公司的商标或注册商标

- Linux 是 Linus Torvalds 在美国和/或其他国家或地区的注册商标。
- Java™ 和所有基于 Java 的商标和徽标是 Oracle 和/或其子公司的商标或注册商标。
- UNIX 是 The Open Group 在美国和其他国家或地区的注册商标。
- Intel、Intel 徽标、Intel Inside、Intel Inside 徽标、Celeron、Intel SpeedStep、Itanium 和 Pentium 是 Intel Corporation 或其子公司在美国和其他国家或地区的商标或注册商标。
- Microsoft、Windows、Windows NT 和 Windows 徽标是 Microsoft Corporation 在美国和/或其他国家或地区的商标。

其他公司、产品或服务名称可能是其他公司的商标或服务标记。

索引

[B]

帮助
SQL 语句 77

[D]

大对象 (LOB)
IBM OLE DB Provider 40

[G]

跟踪 IBM Data Server Provider
System.Diagnostics.Trace 16
更新
DB2 信息中心 78, 79
故障诊断
教程 81
联机信息 81

[J]

教程
故障诊断 81
列表 81
问题确定 81
pureXML 81
结果集
读取
IBM Data Server Provider for .NET 13

[K]

可信上下文
连接字符串关键字 7

[L]

连接关键字
ODBC .NET Data Provider 31
OLE DB .NET Data Provider 25
连接合用
IBM Data Server Provider for .NET 7
OLE DB .NET Data Provider 29

[M]

模式
行集 40

[S]

声明 83
数据类型
映射
OLE DB 与 DB2 43
ADO.NET 数据库应用程序 9
属性
OLE DB 54

[T]

条款和条件
出版物 82

[W]

文档
概述 75
使用条款和条件 82
印刷版 75
PDF 文件 75
问题确定
教程 81
可用的信息 81

[X]

系统要求
IBM OLE DB Provider for DB2 39
线程
IBM OLE DB Provider for DB2 39, 40

[Y]

应用程序
连接到数据源
IBM OLE DB Provider 66
ADO
可更新可滚动游标 59
限制 59
IBM OLE DB Provider 39
Visual Basic 59
应用程序开发
IBM Data Server Provider for .NET 4
IBM Database Add-Ins for Visual Studio 3
游标
可更新
ADO 应用程序 59

游标 (续)

可滚动

ADO 应用程序 59

IBM OLE DB Provider 43

A

ActiveX 数据对象 (ADO) 规范

IBM Data Server Provider for .NET 4

ADO 应用程序

存储过程 59

可更新可滚动游标 59

连接字符串关键字 59

限制 59

IBM OLE DB Provider 对 ADO 方法和属性的支持 60

ADORecordset 对象 30

ADO.NET 应用程序

公共基类 5

开发 1

C

COM

分布式事务支持 66

COM+ 应用程序

连接合用 7

C# .NET

应用程序

编译器选项 24

构建 (Windows) 21

链接选项 24

C/C++ 语言

应用程序

IBM OLE DB Provider 66

D

DB2 信息中心

版本 78

更新 78, 79

E

Enterprise Library 数据访问模块 20

I

IBM Data Server Provider for .NET

调用存储过程 13

读取结果集 13

概述 1, 4, 5, 21, 69

公共基类 5

连接到数据库 6

连接合用 7

IBM Data Server Provider for .NET (续)

数据库系统要求 5

数据类型 9

同时访问结果集 15

Microsoft Entity Framework 17

SQL 语句 10

IBM Database Add-Ins for Visual Studio

概述 3

IBM OLE DB Provider

安装 39

版本 39

概述 39

连接到数据源 58, 59

模式行集 40

使用者 39

数据转换

DB2 类型到 OLE DB 类型 47

OLE DB 类型到 DB2 类型 45

提供程序 39

系统要求 39

线程技术 40

限制 50

应用程序类型 39

游标 43, 59

ADO

方法 60

属性 60

应用程序 59

COM 支持 66

C/C++ 应用程序 66

LOB 40

OLE DB 服务 43

OLE DB 接口 51

OLE DB 属性 54

OLE DB 组件 51

Visual Basic 应用程序 59

IBM.Data.DB2 名称空间 70

IBM.Data.IfxTypesDB2Types 名称空间 73

M

Microsoft Entity Framework

IBM Data Server Provider for .NET 17

Microsoft OLE DB Provider for ODBC

OLE DB 支持 51

Microsoft Transaction Server (MTS)

在 DB2 中支持 66

COM 分布式事务支持 66

O

ODBC .NET Data Provider

概述 1, 31

限制 31

- OLE DB
 - 表函数 39
 - 使用 IBM OLE DB Provider 连接至数据源 58
 - 数据类型
 - 映射到 DB2 数据类型 43
 - 数据转换
 - 从 DB2 到 OLE DB 类型 47
 - 从 OLE DB 到 DB2 类型 45
 - 自动启用的服务 43
 - IBM OLE DB Provider 支持概述 51
 - OLE DB .NET Data Provider 所支持的属性 54
- OLE DB .NET Data Provider
 - 概述 1, 25
 - 限制 26
 - ADORRecordset 对象 30
 - OLE 应用程序
 - 连接合用 29
 - 时间列 29

- .NET (续)
 - C# 应用程序 (续)
 - 链接选项 24
 - 数据库连接 6
 - SQL 语句 10
 - Windows 21
 - IBM Data Server Provider
 - System.Diagnostics.Trace 16
 - pureQuery
 - 优化查询 17
 - Visual Basic 应用程序
 - 编译器选项 22
 - 存储过程 13
 - 结果集 13
 - 链接选项 22
 - 数据库连接 6
 - SQL 语句 10
 - Windows 21

S

- SAMPLE 数据库
 - 连接
 - ODBC .NET Data Provider 31
 - OLE DB .NET Data Provider 25
- SQL 语句
 - 帮助
 - 显示 77
 - 正在执行
 - IBM Data Server Provider for .NET 10
- SystemDiagnosticsTrace
 - 跟踪
 - IBM Data Server Provider for .NET 16

V

- Visual Basic
 - 数据控制支持 59
 - 应用程序 59
 - 游标注意事项 59
- Visual Basic .NET
 - 应用程序
 - 编译器选项 22
 - 构建 21
 - 链接选项 22

[特别字符]

- .NET
 - 应用程序部署 2
 - 应用程序开发软件 2
 - C# 应用程序
 - 编译器选项 24
 - 存储过程 13
 - 结果集 13



Printed in China

S151-1765-01



Spine information:

IBM DB2 10.1 for Linux, UNIX, and Windows

开发 ADO.NET 和 OLE DB 应用程序

