



機能紹介：ADO.NET とイン ターシステムズ製品

Version 2023.1
2024-01-02

機能紹介：ADO.NET とインターシステムズ製品

InterSystems IRIS Data Platform Version 2023.1 2024-01-02

Copyright © 2024 InterSystems Corporation

All rights reserved.

InterSystems®, HealthShare Care Community®, HealthShare Unified Care Record®, IntegratedML®, InterSystems Caché®, InterSystems Ensemble®, InterSystems HealthShare®, InterSystems IRIS®, および TrakCare は、InterSystems Corporation の登録商標です。HealthShare® CMS Solution Pack™ HealthShare® Health Connect Cloud™, InterSystems IRIS for Health™, InterSystems Supply Chain Orchestrator™, および InterSystems TotalView™ For Asset Management は、InterSystems Corporation の商標です。TrakCare は、オーストラリアおよび EU における登録商標です。

ここで使われている他の全てのブランドまたは製品名は、各社および各組織の商標または登録商標です。

このドキュメントは、インターシステムズ社(住所: One Memorial Drive, Cambridge, MA 02142)あるいはその子会社が所有する企業秘密および秘密情報を含んでおり、インターシステムズ社の製品を稼動および維持するためにのみ提供される。この発行物のいかなる部分も他の目的のために使用してはならない。また、インターシステムズ社の書面による事前の同意がない限り、本発行物を、いかなる形式、いかなる手段で、その全てまたは一部を、再発行、複製、開示、送付、検索可能なシステムへの保存、あるいは人またはコンピュータ言語への翻訳はしてはならない。

かかるプログラムと関連ドキュメントについて書かれているインターシステムズ社の標準ライセンス契約に記載されている範囲を除き、ここに記載された本ドキュメントとソフトウェアプログラムの複製、使用、廃棄は禁じられている。インターシステムズ社は、ソフトウェアライセンス契約に記載されている事項以外にかかるソフトウェアプログラムに関する説明と保証をするものではない。さらに、かかるソフトウェアに関する、あるいはかかるソフトウェアの使用から起こるいかなる損失、損害に対するインターシステムズ社の責任は、ソフトウェアライセンス契約にある事項に制限される。

前述は、そのコンピュータソフトウェアの使用およびそれによって起こるインターシステムズ社の責任の範囲、制限に関する一般的な概略である。完全な参照情報は、インターシステムズ社の標準ライセンス契約に記載され、そのコピーは要望によって入手することができる。

インターシステムズ社は、本ドキュメントにある誤りに対する責任を放棄する。また、インターシステムズ社は、独自の裁量にて事前通知なしに、本ドキュメントに記載された製品および実行に対する代替と変更を行う権利を有する。

インターシステムズ社の製品に関するサポートやご質問は、以下にお問い合わせください:

InterSystems Worldwide Response Center (WRC)

Tel: +1-617-621-0700

Tel: +44 (0) 844 854 2917

Email: support@InterSystems.com

目次

機能紹介：ADO.NET とインターシステムズ製品.....	1
1 ADO.NET はなぜ重要か	1
2 ADO.NET と InterSystems IRIS	1
3 ADO.NET の体験	2
3.1 開始の前に	2
3.2 Visual Studio プロジェクトの構成	2
3.3 ADO.NET を介した接続	3
3.4 管理ポータルでの変更の確認	4
4 ADO.NET の詳細	4

機能紹介：ADO.NET とインターシステムズ製品

この機能紹介ガイドでは、InterSystems ADO.NET Managed Provider を介して InterSystems IRIS® データ・プラットフォームに接続する方法を説明します。このガイドの完了後には、**InterSystems.Data.IRISClient.dll** アセンブリを使用するために Visual Studio プロジェクトの構成が完了し、InterSystems IRIS への ADO.NET 接続が確立し、.NET アプリケーションから複数の SQL 文を実行し、InterSystems IRIS システム管理ポータルでこれらの文の効果が確認された状態になります。

詳細にとらわれずに ADO.NET Managed Provider の基本機能を体験していただくため、以下の操作例はシンプルなのにしてあります。これらのアクティビティは、既定の設定と機能のみを使用する設計になっているので、ユーザはトピック外または過度に複雑な詳細を扱うことなく、機能の基本部分を十分に理解することができます。ADO.NET をプロダクション・システムで使用する際には、異なる処理が必要になることがあります。したがって、ここに示す ADO.NET の操作例と本番での操作を混同しないようにしてください。このドキュメントの最後に示すソースでは、ADO.NET をプロダクション環境で使用するために必要な良策が提供されています。

ADO.NET に関するその他のドキュメントについては、“[ADO.NET の詳細](#)”を参照してください。

[InterSystems IRIS の無料の評価版インスタンス](#)で実施できるものも含めて、すべての機能紹介を確認するには、“[インターシステムズの機能紹介](#)”を参照してください。

1 ADO.NET はなぜ重要か

ADO.NET は Microsoft .NET Framework によるデータ・アクセス・テクノロジーであり、データ・ソースへのアクセスを可能にします。データベース接続を確立するために使用され、多くのタイプのデータ・ソースに接続し、そのデータ・ソースに SQL による操作を実行するための標準となる、信頼性の高い方法を、.NET Framework プログラマに提供します。特に ADO.NET を以前に使用したことのあるユーザであれば、ADO.NET Managed Provider を介した InterSystems IRIS への接続は簡単です。.NET アプリケーションから InterSystems IRIS への ADO.NET 接続を確立することで、ユーザの .NET アプリケーションから SQL コマンドを InterSystems IRIS データベースに対して実行できるようになります。

InterSystems IRIS は初めてでも、.NET と SQL に精通しているユーザであれば、既にお持ちの専門知識を直に使用して、データベース・プラットフォームをよく理解することができます。開発環境では、ほんの数行のコードを使用するだけで、ADO.NET 接続と SQL コマンドをテストできます。

2 ADO.NET と InterSystems IRIS

InterSystems IRIS は ADO.NET 仕様に完全準拠する実装です。InterSystems ADO.NET Managed Provider は、データへの簡単なリレーショナル・アクセスを提供します。アプリケーションからの ADO.NET メソッド呼び出しを処理し、InterSystems IRIS に SQL 要求を送信します。さらに、呼び出し元のアプリケーション（この場合は .NET アプリケーション）に結果を返します。

ADO.NET を介した InterSystems IRIS への接続は単純なプロセスです。

InterSystems IRIS ADO.NET 機能を使用するには、まず **InterSystems.Data.IRISClient.dll** アセンブリを Visual Studio プロジェクトに依存関係として追加する必要があります。いくつかの設定の確認後、サンプル・コードを使用して InterSystems

IRIS への ADO.NET 接続を確立し、SQL クエリを実行します。**InterSystems.Data.IRISClient.dll** アセンブリは、全面的に .NET マネージド・コードを使用して実装されるため、.NET 環境での配置が容易です。スレッドセーフであるため、マルチスレッドの .NET アプリケーションで使用できます。

3 ADO.NET の体験

ADO.NET と InterSystems IRIS の操作方法を示す簡単なデモが開発されています。

3.1 開始の前に

この手順を使用するには、.NET フレームワークおよび Visual Studio がインストールされた、操作するための Windows システム、および接続先として、稼働している InterSystems IRIS インスタンスが必要です。InterSystems IRIS の選択肢としては、いくつかのタイプのライセンス付与されたインスタンスおよび無料の評価版インスタンスがあります。操作しているシステムでインスタンスをホストする必要はありません（ただし、相互のネットワーク・アクセスが必要です）。操作を実行するインスタンスをまだ用意できていない場合にインスタンスのタイプ別の導入方法の詳細を確認するには、“InterSystems IRIS の基礎：IDE の接続”の“[InterSystems IRIS の導入](#)”を参照してください。同じドキュメントの“[InterSystems IRIS 接続情報](#)”および“[.NET IDE](#)”の情報を使用して、Visual Studio を InterSystems IRIS インスタンスに接続します。

3.2 Visual Studio プロジェクトの構成

Visual Studio のメイン・メニューで、[ファイル] → [新規作成] → [プロジェクト] の順に選択することで、新規プロジェクトを作成します。表示されるダイアログで、[Visual C#] オプションをクリックして [コンソール・アプリ (.NET Framework)] を選択します。[名前] フィールドに、ADONET と入力します。[OK] をクリックします。これにより、.NET Framework を使用して新しいコンソール・アプリケーションが作成されます。

次に、Visual Studio のメイン・メニューで、[プロジェクト] → [ADONET プロパティ] を選択します。[ターゲット・フレームワーク] の下で [.NET Framework 4.5] を選択します。

3.2.1 アセンブリ参照の追加

InterSystems.Data.IRISClient.dll アセンブリをローカル・システムにインストールしておく必要があります。最新バージョンのライブラリは、“[InterSystems IRIS Driver Packages](#)” ページからダウンロードできます。InterSystems IRIS がローカル・マシンまたはアクセスできる別のシステムにインストールされている場合、このファイルは、`install-dir¥dev¥dotnet¥bin¥version` (`install-dir` はインスタンスのインストール・ディレクトリで、`version` はライブラリの .NET バージョン) にあります。

プロジェクトに **InterSystems.Data.IRISClient.dll** へのアセンブリ参照を追加するには、以下の手順に従います。

1. Visual Studio のメイン・メニューで、[プロジェクト] → [参照の追加] を選択します。
2. 表示されるウィンドウで [参照...] をクリックします。
3. **InterSystems.Data.IRISClient.dll** ファイルの場所を参照します。
4. このファイルを選択し、[追加] をクリックします。
5. [OK] をクリックします。

Visual Studio のソリューション・エクスプローラでは、[参照] の下に **InterSystems.Data.IRISClient.dll** アセンブリが表示されます。

3.3 ADO.NET を介した接続

この時点で、.NET アプリケーションから InterSystems IRIS に接続する準備ができました。InterSystems ADO.NET Managed Provider の接続文字列は、接続のプロパティを定義するキー/値ペアで構成されます。接続文字列の構文は、以下のとおりです。

```
Server=host_IP; Port=superserverPort; Namespace=namespace; Password=password; User ID=username;
```

これらの変数は、InterSystems IRIS インスタンスのホストの IP アドレス、インスタンスのスーパーサーバのポート、インスタンス上のネームスペース、およびインスタンスの資格情報を表します。これは、“[開始の前に](#)”で説明したように、Visual Studio をインスタンスに接続する際に使用した情報と同じです。

Visual Studio に貼り付けた後、以下のコードの情報を更新します。namespace は、ここに示しているように事前定義のネームスペース **USER** に設定することも、インストールしたインスタンスで作成した別のネームスペースに設定することもできます。

```
using System;
using InterSystems.Data.IRISClient;

namespace ADONET
{
    class Program
    {
        static void Main(string[] args)
        {
            String host = "<host>";
            String port = "<port>";
            String username = "<username>";
            String password = "<password>";
            String Namespace = "USER";

            IRISConnection IRISConnect = new IRISConnection();
            IRISConnect.ConnectionString = "Server = " + host
                + "; Port = " + port + "; Namespace = " + Namespace
                + "; Password = " + password + "; User ID = " + username;

            IRISConnect.Open();

            String sqlStatement1 = "CREATE TABLE People(ID int, FirstName varchar(255), LastName
varchar(255))";
            String sqlStatement2 = "INSERT INTO People VALUES (1, 'John', 'Smith')";
            String sqlStatement3 = "INSERT INTO People VALUES (2, 'Jane', 'Doe')";
            String queryString = "SELECT * FROM People";

            IRISCommand cmd1 = new IRISCommand(sqlStatement1, IRISConnect);
            IRISCommand cmd2 = new IRISCommand(sqlStatement2, IRISConnect);
            IRISCommand cmd3 = new IRISCommand(sqlStatement3, IRISConnect);
            IRISCommand cmd4 = new IRISCommand(queryString, IRISConnect);

            //ExecuteNonQuery() is used for CREATE, INSERT, UPDATE, and DELETE SQL Statements
            cmd1.ExecuteNonQuery();
            cmd2.ExecuteNonQuery();
            cmd3.ExecuteNonQuery();

            //ExecuteReader() is used for SELECT
            IRISDataReader Reader = cmd4.ExecuteReader();

            Console.WriteLine("Printing out contents of SELECT query: ");
            while (Reader.Read())
            {
                Console.WriteLine(Reader.GetValue(0).ToString() + ", " + Reader.GetValue(1).ToString()
+ ", " + \
                + Reader.GetValue(2).ToString());
            }

            Reader.Close();
            cmd1.Dispose();
            cmd2.Dispose();
            cmd3.Dispose();
            cmd4.Dispose();
        }
    }
}
```

```
        IRISConnect.Close();  
  
        Console.WriteLine("Press any key to continue...");  
        Console.ReadKey();  
    }  
}
```

[開始] ボタンをクリックするか、F5 キーを押すことで、コードを実行します。

接続とクエリが正常に完了すると、コンソール・ウィンドウに SELECT クエリの結果が表示されます。

3.4 管理ポータルでの変更の確認

次に、以下の手順を使用して、管理ポータルで結果を確認します。

1. “InterSystems IRIS の基礎：IDE の接続” で説明している [インスタンス用の URL](#) を使用して、ブラウザ内でインスタンスの管理ポータルを開きます。
2. コードで指定したネームスペース内にいない場合は、そのネームスペースに切り替えます (ページ上部の [ネームスペース:] インジケータの横にある [切り替え] をクリックします)。
3. [SQL] ページ ([システム・エクスプローラ] → [SQL]) に移動し、[クエリ実行] タブをクリックして、以下の SQL クエリを貼り付けます。

```
SELECT  
ID, FirstName, LastName  
FROM SQLUser.People
```

[実行] をクリックします。サンプル・コードで作成した People テーブルの内容がページに表示されます。

4 ADO.NET の詳細

ADO.NET、SQL、および InterSystems IRIS の詳細は、以下を参照してください。

[InterSystems Managed Provider for .NET の使用法](#)

[InterSystems SQL の使用法](#)

[ADO.NET の概要](#)