

SQL Anywhere と Microsoft .NET

はじめに

SQL Anywhere には、Microsoft .NET Framework との統合に対応するさまざまな機能が用意されています。各機能は、データベースを活用するアプリケーションの開発時に、SQL Anywhere データベースとのスムーズな連携を実現できるように設計されています。このホワイトペーパーでは、SQL Anywhere の Microsoft .NET 統合サポート機能の概要について説明するとともに、.NET Framework の概念の概要についても説明します。

さらに、SQL Anywhere での統合のポイントについても詳細に説明します。具体的には、XML とリレーショナル・データベースの連携、SQL Anywhere ADO.NET と .NET Data Provider によるデータ・ソースへの接続、ADO.NET Entity Framework と SQL Anywhere の連携使用、SQL Anywhere ASP.NET プロバイダによる Web サイトのセキュリティ管理、.NET 固有のストアド・プロシージャや関数の記述、Mobile Link 同期での .NET のサポートといったトピックを取り上げます。.NET の概念とそのサポート機能をよく理解しておけば、開発時に SQL Anywhere の機能をフルに活用して、目標をスムーズに達成できます。

必須ではありませんが、このホワイトペーパーでは、Microsoft .NET Framework とそのコンポーネントについて、ある程度の知識があることを前提としています。

.NET Framework について

.NET Framework は、分散環境でのアプリケーション開発の簡素化を目的として設計されたプログラミング・モデルです。.NET Framework の主要なコンポーネントとしては、次の 2 種類が挙げられます。

- **共通言語ランタイム**: 管理・保護されたアプリケーション実行環境。ランタイムには、実行中のコードを管理し、メモリやスレッドの管理タスクを処理する役割があります。仮想マシンでは、サポート対象の .NET 言語 (C#, Visual Basic, C++ など) が実行されます。
- **共通クラス・ライブラリ**: サポート対象のすべての言語で使用可能な共通クラスのライブラリ。.NET の導入以前には、言語ごとに異なる Windows API が用意されていました。

Microsoft 提供の .NET Compact Framework は、スマート・デバイス向けの .NET Framework のサブセットです。スマート・デバイスに管理コードを提供するとともに、PDA (Personal Digital Assistant) や携帯電話など、Microsoft のサポート対象デバイス上で、セキュアなダウンロード対応アプリケーションの実行を可能にします。

SQL Anywhere などのデータ管理ソフトウェアでは、「.NET のサポート」は次の 2 点を意味します。

- .NET Framework とプログラミング環境 (特に .NET の共通データベース API である ADO.NET) でのプログラムが可能であること
- XML のサポート (特に、データの XML 形式対応と、関数の Web サービス対応)

SQL Anywhere での .NET との統合のポイント

SQL Anywhere は、.NET を広範にサポートしています。具体的には、次のようなサポート機能が挙げられます。

- SQL Anywhere では、XML をエクスポート可能、またデータベースに XML データを格納可能
- SQL Anywhere では、XML Web サービスを介してデータを公開可能
- SQL Anywhere では、標準的な .NET OLE DB と ODBC Data Provider のサポートに加えて、SQL Anywhere .NET Data Provider を使用する ADO.NET インタフェースもサポート
- SQL Anywhere では、サポート対象の .NET 言語 (バージョン 2.0) を使用して、ストアド・プロシージャや関数を記述可能
- SQL Anywhere では、ADO.NET Entity Framework をサポートし、Visual Studio 用の統合コンポーネントを提供 (これにより、SQL Anywhere データベースからエンティティ・データ・モデルを生成可能)
- SQL Anywhere ASP.NET プロバイダを使用すると、ASP.NET Web サービスの開発者側は認証システムを実装して、SQL Anywhere データベースを基盤とする Web サイトを実行可能
- SQL Anywhere は Microsoft Visual Studio 2008 および 2010 と緊密に統合され、アプリケーションの開発時に SQL Anywhere データベースをスムーズに活用可能
- Mobile Link では、.NET API を同期クライアント (DBMLSync) に公開

次の項では、各機能の詳細について説明するとともに、.NET のサポート強化を目的として、SQL Anywhere の今後のリリースに追加される予定の機能やオプションについても説明します。

XML とリレーショナル・データベース

XML (Extensible Markup Language : 拡張マークアップ言語) は、テキスト形式での構造化データの表現に対応するマークアップ言語です。XML は、記述の容易さについては HTML、機能の柔軟性については SGML を設計の基盤としています。XML として固有のデータ形式は存在せず、適正なタグの名前や構造はアプリケーションごとに再定義できます。XML ドキュメントでは、DTD (Document Type Definition : ドキュメント型定義) または XML スキーマに基づいて、XML ファイルで使用される構造、要素、および属性を定義します。

.NET では、汎用のデータ交換形式として XML が使用されています。また、プロセス間通信のメカニズムとして、XML ベースの SOAP (Simple Object Application Protocol) が使用されています。

XML とリレーショナル・データベースで情報をやり取りする方法としては、次のような方法が挙げられます。

- リレーショナル・データベースへの XML のインポート
- リレーショナル・データベースからの XML のエクスポート
- データベースへの XML ドキュメントの格納
- XML クエリの実行
- XML ビューの作成
- Web サービスへのアクセス

SQL Anywhere の現行リリースでは、XML のエクスポートと、SQL Anywhere データベースへの XML ドキュメントの格納をサポートしています。XML ドキュメントの読み取りとその内容のインポートに対応するプロシージャや関数もいくつか用意されています。

リレーショナル・データベースへの XML のインポート

XML をリレーショナル・データベースにインポートする方法としては、アプリケーションで XML を解析してから、テーブルの挿入、削除、および更新を実行する SQL 文を生成するという方法が挙げられます。現在、組み込みの OpenXML プロシージャ／関数など、上記の処理に対応するパーサやツールがいくつか用意されています。

ADO.NET DataSet オブジェクトを使用する方法では、XML ドキュメントから DataSet にデータやスキーマを読み込むことができます。ReadXml メソッドは、スキーマとデータの両方を含む XML ドキュメントから DataSet の値を設定します。これに対して、ReadXmlSchema メソッドは、XML ドキュメントからスキーマだけを読み取ります。DataSet に XML ドキュメントのデータが設定された時点で、DataSet の変更内容をデータベースのテーブルに適用して更新できます。

現行リリースの SQL Anywhere では、この 2 つの方法のどちらでも、XML ドキュメントからデータベースにデータをインポートできます。

リレーショナル・データベースからの XML のエクスポート

FOR XML オプションを使用することで、SELECT 文で XML を直接エクスポートできます。

SQL Anywhere では、Interactive SQL から XML をエクスポートできます。OUTPUT 文では、生成された XML ファイルにクエリ結果を出力するという XML 形式をサポートしています。生成された XML ファイルは UTF-8 形式でエンコードされ、DTD が埋め込まれます。XML ファイルでは、バイナリ値は、バイナリデータを 2 つの 16 進数文字列で表現する文字データ (CDATA) ブロックにエンコードされます。

SQL Anywhere では、これ以外にも 2 種類の方法でデータを XML 形式でエクスポートできます。具体的には、ADO.NET DataSet オブジェクトを使用する方法と、アプリケーションで結果セットから XML ドキュメントを作成する方法です。

ADO.NET DataSet オブジェクトを使用すると、DataSet の内容を XML ドキュメントに保存できます。たとえば、データベースに対するクエリ結果から DataSet の値を設定して、DataSet のスキーマのみまたはスキーマとデータの両方を XML ファイルに保存できます。WriteXml メソッドは、スキーマとデータの両方を XML ファイルに保存します。これに対して、WriteXMLSchema メソッドは、スキーマのみを XML ファイルに保存します。

これ以外にも、SQL クエリを実行してから、アプリケーションでクエリの実行結果から XML ドキュメントを作成する方法もあります。

リレーショナルデータベースへの XML の格納

SQL Anywhere では、XML ドキュメントをデータベースに格納する場合に使用できるデータ型として、XML データ型と LONG VARCHAR データ型という 2 種類をサポートしています。この 2 種類のデータ型では共通して、XML ドキュメントが文字列としてデータベースに格納されます。

XML データ型では、データベース・サーバの文字セット・エンコードが使用されます。XML データ型は、文字列との相互キャストに対応するその他のデータ型と相互にキャストできます。文字列を XML にキャストする際に、文字列が形式を満たしているかどうかのチェックは行われません。

XML のクエリ

データベースに格納した XML に対してクエリを実行できると便利です。具体的には、組み込み関数とストアード・プロシージャを使用する方法があります。

SQL Anywhere での XML の使用法の詳細については、オンライン・ドキュメント『[データベースにおける XML の使用](#)』を参照してください。

XML Web サービス

Web サービスはプログラム可能なアプリケーション・ロジックであり、別の環境で実行されているアプリケーションからでも、HTTP、SOAP (Simple Object Access Protocol)、XML など、一連の標準的な基本 Web プロトコルに基づいてインターネット経由でアクセスが可能です。Web サービスは、特定のプラットフォームや言語に依存しないように設計されています。

WSDL (Web Services Description Language :Web サービス記述言語) は XML ベースの言語であり、Web サービスの内容とそのアクセス方法を記述する場合に使用します。UDDI (Universal Description Discovery and Integration) は、Web サービスのレジストリの役割を果たします。機能的には検索エンジンに類似していますが、Web サービス専用です。WSDL と UDDI のどちらも Web サービスで必須ではありませんが、これらを使用することにより、プログラムで Web サービスの検索 (UDDI の使用時) や使用 (WSDL 情報の使用時) が容易になります。

SQL Anywhere の Web サービスは、統合 HTTP サーバや、データベース サーバの SOAP 要求マネージャを介してサポートされています。これにより、HTTP 経由で SQL Anywhere に SOAP 要求を送ってから、SQL Anywhere から要求元の HTTP クライアントに応答を返すことができます。SQL Anywhere 12.0.0 からは、HTTP サービスで自動接続プーリングがサポートされ、プランのキャッシュによる効果が最大化されるとともに、潜在的なパフォーマンスの向上が実現しました。

この他にも、.NET データ・ペイロード形式を含む、データ・フォーマット用のオプションがいくつか用意されています。具体的には、伝送時にもデータ型指定が保持され、アプリケーションで適切に使用できるような形式でデータをフォーマットします。

[SQL Anywhere の Web サービス](#)の詳細については、オンライン・ドキュメントを参照してください。

ADO.NET と .NET Data Provider

ADO.NET は、ODBC、DAO、RDO、OLE DB、ADO といった一連の Microsoft 提供のデータ・アクセス API の中でも最新のものであり、Microsoft .NET Framework 向けに推奨されるデータ・アクセス・コンポーネントです。.NET Data Provider は、データ・ストアへのアクセスを提供し、アプリケーションによるデータ・ソースからのデータの取得や修正を可能にするツールです。ADO.NET では、.NET アプリケーションとのデータ統合を円滑にする目的で、データ・プロバイダ (別名、管理プロバイダ) を使用します。

.NET Framework は、Microsoft から出荷された時点で、SQL Server .NET Data Provider、OLE DB .NET Data Provider、および ODBC .NET Data Provider という 3 種類のデータ・プロバイダが用意されています。SQL Server .NET Data Provider を使用すると、Microsoft SQL Server バージョン 7.0 以降のデータベースに接続できます。これに対して、OLE DB .NET Data Provider を使用すると、OLE DB プロバイダを用意しているデータ・ソースに加えて、リレーショナル・データベース・システム (SQL Anywhere など) にもアクセスできます。.NET Framework では、ODBC データ・ソースへの接続用に、Microsoft の ODBC .NET Data Provider もサポートしています。

SQL Anywhere .NET Data Provider に収録されている OLE DB ドライバは、.NET Framework に収録されている 3 種類のデータ・プロバイダを補完する目的で実装されています。これにより、Windows プラットフォーム全般 (Windows Mobile など) 向けの標準 ADO.NET インタフェースを採用している、サポート対象の任意の .NET 言語でアプリケーションを開発できます。

これ以外にも、標準の非管理 SQL Anywhere Data Provider は、OLE DB や ODBC Data Provider 経由でアクセスできます。

ADO.NET プログラミング・インタフェース

各管理プロバイダは、データベースの操作や管理に対応する標準オブジェクトとして、次のクラスを実装しています。

- **Connection** : データ・ソースへの接続
- **Command** : データ・ソースに対するコマンドの実行 (SQL 文やストアド・プロシージャ)
- **DataReader** : コマンドの実行結果に対する転送専用、読み取り専用アクセスの提供
- **DataAdapter** : DataSet への値の設定と、データに対する更新の処理
- **Parameters** : Command オブジェクトに対するパラメータの受け渡し
- **Transaction** : COMMIT および ROLLBACK 機能の提供
- **Error, Exception** : エラー／警告メッセージの処理および収集

DataSet は、ADO.NET の主要なオブジェクトの 1 つです。データベースから取得されたデータ用の非接続型ストアであり、DataTable、DataRow、DataColumn、DataRelation といったオブジェクトの集合です。DataSet 自体は ADO.NET アーキテクチャの構成要素として、Microsoft から提供される汎用オブジェクトです。プロバイダの DataAdapter を使用して、DataSet の値を設定できます。また、DataSet に含まれているデータに変更を加えて、その内容を DataAdapter 経由でデータベースに適用できます。DataSet は管理プロバイダやデータベース・ドライバからは独立しており、XML 内のデータやスキーマ情報の読み書きに使用できます。

DataSet からデータベースを更新する場合は、注意が必要です。DataSet に対する変更内容は、接続されていないときに適用されます。つまり、アプリケーションでは、データベース内の対象ローはロックされていないということです。DataSet の変更内容をデータベースに適用する際に発生する可能性がある競合に対処できるよう、アプリケーションを設計する必要があります。

Mobile Link には、複製環境で競合が発生した場合の解決法が用意されています。Mobile Link はセッションベースのリレーショナル・データベースの同期システムであり、中央の統合データベースと多数のリモート・データベースの間での双方向データ同期に対応します。Mobile Link を使用すると、同期対象のデータ部分を選択できます。また、Mobile Link には、異なるリモート・データベースのデータに対する変更に伴う競合を解決するための機能が用意されています。

SQL Anywhere .NET Data Provider は、SQL Anywhere データベース向けのネイティブ .NET Data Provider です。他のサポート対象のデータ・プロバイダとは異なり、SQL Anywhere との直接通信が可能であり、ブリッジ・テクノロジーを必要としません。Windows Mobile 上での実行にも対応しています。

.NET Data Provider は iAnywhere.Data.SQLAnywhere ネームスペースを実装し、C# や Visual Basic .NET など、.NET のサポート対象言語でのプログラム作成や、SQL Anywhere データベースからのデータへのアクセスを可能にします。

次に示すのは、SQL Anywhere .NET Data Provider の主要な特徴の一部です。

- **アプリケーションの柔軟性**: SQL Anywhere .NET Data Provider では、DataReader や DataAdapter といったクラス全般をサポートしています。
- **.NET 言語のサポート**: SQL Anywhere .NET Data Provider は、C# や Visual Basic .NET など、.NET のサポート対象言語で使用できます。

- **接続プーリングのサポート**: SQL Anywhere .NET Data Provider では、接続プーリングをサポートしています。これにより、アプリケーションでは、データベースへの新規接続を確立する処理を繰り返す代わりに、既存の接続を再使用できます。結果として、パフォーマンスの向上をもたらす場合があります。
- **優れたパフォーマンス**: SQL Anywhere .NET Data Provider は直接的な実装であり、OLEDB や ODBC のブリッジを必要としません。処理は、.NET OLEDB Data Provider よりも高速です。
- **広範な Microsoft Windows のサポート**: SQL Anywhere .NET Data Provider では、Windows プラットフォームと Windows Mobile プラットフォームの両方をサポートしています。現在、Microsoft は Windows Mobile 上のデータへのアクセスを可能にする OLEDB や ODBC の管理プロバイダを提供していません。Windows Mobile 上で .NET Compact Framework を使用して SQL Anywhere にアクセスするには、SQL Anywhere .NET Data Provider が唯一の手段です。

ADO.NET Entity Framework

ADO.NET Entity Framework は、Microsoft .NET Framework 3.5 SP1 から導入された新しいテクノロジーです。このテクノロジーを使用して、開発者はリレーショナル・データの表現とアプリケーションのビジネス・ロジックの間にあるギャップを埋めることができます。データベースの設計は通常、DBA (Database Administrator : データベース管理者) が担当し、データの取得に最適化されたスキーマに情報全般を格納します。ただし、このスキーマはデータ・アクセスに対しては有用ですが、アプリケーションで必要とされるビジネス・オブジェクトを表現する場合、最適な方法ではありません。多くの場合、プログラマが設計や実装を求められるのは、リレーショナル・システムに格納されているデータを、エンタープライズ・ソリューションに不可欠な要素である、堅牢かつ再利用可能なオブジェクトに変換する独自のメソッドです。Entity Framework は、リレーショナル・データに基づいたビジネス・ロジックの設計を可能にする、直感的なアプローチをソフトウェア開発者に提供します。

EDM (Entity Data Model : エンティティ・データ・モデル) は、アプリケーションで使用されるデータの概念ビューです。既存のデータベース・スキーマに大量のデータが含まれていても、実際にクライアント・プログラムで必要とされるのはデータのサブセットのみであることは多々あります。たとえば、オンライン注文を処理する Microsoft Windows クライアント・アプリケーションについて考えると、必要な情報は顧客、製品、注文などに限られます。この場合、すべてのデータベース・テーブルを EDM に含むのではなく、アプリケーションに関連するテーブルだけを用意すれば済みます。

従来と同様に、最新のデータ・アクセス・テクノロジーをサポートするという方針に基づいて、SQL Anywhere では、開発者は ADO.NET Entity Framework を利用して、データベースを活用する高度なアプリケーションを開発できます。SQL Anywhere の Visual Studio との統合ツールを使用すると、SQL Anywhere データベースからエンティティ・データ・モデルを作成して、Entity Designer でモデルの視覚的な表現も確認できます。さらに、エンティティ、オブジェクト・サービス、および EntityClient Provider メソッドロジックに対する LINQ (Language Integrated Query : 統合言語クエリ) を使用して、データベースに格納されたデータにアクセスできます。エンティティ・クラスは部分的なクラス・エンティティであり、開発者はクラス・エンティティを拡張してプロパティやメソッドも含めることができます。この方法により、ビジネス・オブジェクトを補完して、必要なビジネス・ロジックを提供できます。

SQL Anywhere ASP.NET プロバイダ

Microsoft の ASP (Active Server Page) テクノロジーの後継である ASP.NET は、Web アプリケーションのフレームワークです。Web サイト開発者は ASP.NET を使用して、動的な Web サイト、Web アプリケーション、Web サー

ビスなどを構築できます。ASP.NET は CLR (Common Language Runtime : 共通言語ランタイム) を基盤としており、プログラマはサポート対象の .NET 言語を使用して ASP.NET のコードを記述できます。ASP.NET SOAP 拡張フレームワークでも、ASP.NET コンポーネントによる SOAP メッセージの処理が可能です。さらに、Visual Studio には、強力なライブラリやメソッドへのアクセスに加えて、多機能な統合型の Web 開発環境も用意されており、従来よりも容易に機能的な Web サイトの設計や実装を進めることができます。

認証・承認システムを Web アプリケーションに実装する作業は、Web 開発者にとって複雑なものでした。ASP.NET 2.0 以降では、証明書の格納や検証を行うコードの作成・修正、ユーザの作成・管理、Web サイトのステータス/パフォーマンスの監視といった一連の作業を Web 開発者が行う必要はなくなりました。代わりに、Web アプリケーションでのセキュリティ関連情報の管理に対応する、セキュアで拡張性に優れた実装機能が ASP.NET プロバイダに用意されています。必要な作業は、プロバイダを構成し、用意された ASP.NET Website Administration Tool を使用して Web サイトを管理するだけです。

SQL Anywhere ASP.NET プロバイダを使用すると、SQL Anywhere データベースをバックエンド・ストレージとして利用するセキュリティ・メカニズムを構築できます。ユーザの個人情報を格納・管理する目的に必要なスキーマを対象のデータベースに追加するには、ASP.NET プロバイダをインストールする必要があります。用意されているセットアップ・ウィザードを使用すれば、いくつかの構成を行うだけで、SQL Anywhere データベースに対してスキーマを自動的にセットアップすることもできます。

SQL Anywhere には、次の 5 種類のプロバイダが用意されています。

- **Membership Provider** : 認証・承認サービスを提供します。
- **Roles Provider** : 役割の作成、役割へのユーザの追加、および役割の削除に対応するメソッドを提供します。Roles Provider を使用して、グループへのユーザの割り当てや、パーミッションの管理を行います。
- **Profiles Provider** : ユーザの基本設定など、ユーザ情報の読み取り、格納、および取得に対応するメソッドを提供します。
- **Web Parts Personalization Provider** : カスタマイズしたコンテンツや Web ページのレイアウトのロードと格納に対応するメソッドを提供します。
- **Health Monitoring Provider** : 配備済み Web アプリケーションのステータス監視に対応するメソッドを提供します。

SQL Anywhere ASP.NET プロバイダの詳細については、オンライン・ドキュメントを参照してください。

CLR のストアド・プロシージャおよび関数

SQL Anywhere には、CLR のストアド・プロシージャおよび関数のサポートも用意されています。CLR のストアド・プロシージャおよび関数を使用すると、クラス・ライブラリ (.dll) 内に存在する、C#、VB.NET、その他の .NET 言語で記述された関数やプロシージャを参照できます。データベースでは、ライブラリがロードされ、SQL ストアド・プロシージャの代わりに関数が呼び出されます。これにより、任意の .NET 言語でデータを操作することが可能になり、ビジネス・ロジックの維持・再使用が実現します。

SQL Anywhere の CLR 外部環境の詳細については、オンライン・ドキュメントを参照してください。

Mobile Link の .NET サポート

Mobile Link はセッションベースの同期システムであり、統合データベースと呼ばれるメイン・データベースと多数のリモート・データベースの間で双方向の同期を可能にします。

Mobile Link 同期スクリプト

同期スクリプトは、Mobile Link 同期サーバの動作を制御する場合に使用します。一般に、同期スクリプトは、統合データベースの SQL 言語か Java でストアド・プロシージャとして記述されます。

SQL Anywhere では、サポート対象の .NET プログラミング言語を使用して、Mobile Link 同期スクリプトも記述できます。これにより、.NET Common Language Runtime の機能全般を利用して、C#、Visual Basic .NET、その他のサポート対象の .NET 言語でスクリプトを記述できます。

.NET 同期ロジックを使用すると、複数のデータベース・プラットフォームに対する操作が可能になり、RDBMS 間での移植が実現します。.NET 同期ロジックでは、Mobile Link を使用して、アプリケーション・サーバ、Web サーバ、およびファイルから取得したデータにアクセスできます。統合データベースの同期接続でデータにアクセスするには、同期ロジックで `iAnywhere` クラスを使用します。たとえば、.NET スクリプトを記述して、サーバの `authenticate_user` イベント発生時に外部サーバを使用してユーザ ID とパスワードを検証するような場合が考えられます。

スクリプトを使用すると、統合データベースにアップロードしたデータがコミットされる前にアクセスして操作することもできます。たとえば、コミットの前に変更を拒否することで、他のリモート環境には反映されません。外部プログラムを使用して統合データベース上のデータにアクセスしている場合は、コミットされるまで更新の確認や取り消しは実行できません。

同様の機能は、Ultra Light データベースでの使用もサポートされています。Ultra Light は SQL Anywhere の代替ソリューションであり、省フットプリントを特徴としています。具体的な用途としては、小型デバイス向けのリレーショナル・データベース・アプリケーションの構築・配備が挙げられます。

Direct Row API

Direct Row API からデータを直接 .NET 環境に受け渡して、データベースへのコミット前に未処理の同期データを操作する方法もあります。

DBMLSync 用の Mobile Link .NET API

DBMLSync は、クライアント側の同期を開始する同期クライアントです。Dbmlsync .NET API (`iAnywhere.Mobile Link.Clinet` ネームスペースを使用) に用意されたプログラミング・インタフェースを使用すると、C++ または .NET で記述された Mobile Link クライアント・アプリケーションで、同期の実行や、要求した同期の進行状況についての

フィードバックの受信が可能になります。これにより、アクセスできる同期結果の情報量が増えるとともに、同期処理をアプリケーションにシームレスに統合できます。

Mobile Link と同期スクリプトの詳細については、オンライン・ドキュメントを参照してください。

まとめ

SQL Anywhere は現在、広範な .NET のサポートを提供していますが、今後のリリースでも .NET のサポートを拡充していく予定です。

SQL Anywhere .NET Data Provider は、Windows Mobile 上で動作しているアプリケーションを含む、.NET アプリケーションへのネイティブ・アクセスを可能にします。さらに、SQL Anywhere データベースでは、XML のインポート、エクスポート、および格納が可能です。

SQL Anywhere では、ADO.NET Entity Framework テクノロジーをサポートしています。SQL Anywhere に用意された Visual Studio 用の統合コンポーネントを利用することで、SQL Anywhere データベースからエンティティ・データ・モデルを生成できます。エンティティ、オブジェクト・サービス、および EntityClient Provider メソッドに対する LINQ (Language Integrated Query : 統合言語クエリ) を使用することで、SQL Anywhere データベースに格納されたデータにアクセスできます。

SQL Anywhere ASP.NET プロバイダでは、Web アプリケーションの役割、メンバシップ、およびプロファイルの管理に対応する、セキュアで拡張性に優れた実装機能を利用して、SQL Anywhere データベースを基盤とする Web サイトを実現できます。

Mobile Link では、.NET Common Runtime Languages の機能全般を利用できます。具体的には、サポート対象の .NET 言語で同期スクリプトを記述できます。

法的注意

Copyright (C) 2010 iAnywhere Solutions, Inc. All rights reserved.

iAnywhere Solutions、iAnywhere Solutions（ロゴ）は、iAnywhere Solutions, Inc.とその系列会社の商標です。その他の商標はすべて各社に帰属します。

本書に記載された情報、助言、推奨、ソフトウェア、文書、データ、サービス、ロゴ、商標、図版、テキスト、写真、およびその他の資料（これらすべてを"資料"と総称する）は、iAnywhere Solutions, Inc.とその提供元に帰属し、著作権や商標の法律および国際条約によって保護されています。また、これらの資料はいずれも、iAnywhere Solutionsとその提供元の知的所有権の対象となるものであり、iAnywhere Solutionsとその提供元がこれらの権利のすべてを保有するものとします。

資料のいかなる部分も、iAnywhere Solutionの知的所有権のライセンスを付与したり、既存のライセンス契約に修正を加えることを認めるものではないものとします。

資料は無保証で提供されるものであり、いかなる保証も行われません。iAnywhere Solutionsは、資料に関するすべての陳述と保証を明示的に拒否します。これには、商業性、特定の目的への整合性、非侵害性の黙示的な保証を無制限に含みます。

iAnywhere Solutionsは、資料自体の、または資料が依拠していると思われる内容、結果、正確性、適時性、完全性に関して、いかなる理由であろうと保証や陳述を行いません。iAnywhere Solutionsは、資料が途切れていないこと、誤りがないこと、いかなる欠陥も修正されていることに関して保証や陳述を行いません。ここでは、「iAnywhere Solutions」とは、iAnywhere Solutions, Inc.またはSybase, Inc.とその部門、子会社、継承者、および親会社と、その従業員、パートナー、社長、代理人、および代表者と、さらに資料を提供した第三者の情報元や提供者を表します。