



Micro Focus Visual COBOL R4
InterSystems CACHE データベースアクセス
動作検証結果報告書

2011年7月27日

マイクロフォーカス株式会社

Copyright © 2011 Micro Focus. All Rights Reserved.

記載の会社名、製品名は、各社の商標または登録商標です。

1. 検証概要、目的及びテスト方法

1.1 検証概要

Visual COBOL の OpenESQL 機能を使用すると、ODBC ドライバと ADO.NET データプロバイダを経由して様々なデータベースにアクセスすることができます。ここでは、InterSystems CACHÉ データベースに ODBC/ADO.NET 経由で COBOL プログラムからアクセスできることを動作検証しました。

1.2 目的及びテスト方法

Micro Focus Visual COBOL は最新鋭の COBOL 言語開発・実行環境を提供します。COBOL 言語への埋め込み SQL 処理系を標準装備しており、ODBC ドライバと ADO.NET データプロバイダを経由して様々なデータベースにアクセスすることができます。製品出荷時に弊社にて動作検証できているのは Oracle, SQL Server, DB2 のみですが、ODBC 3.x, ADO.NET 仕様に準拠したデータソースであれば設計上問題なくアクセスすることができます。今回、InterSystems Caché for Windows (x86-32) 2010.2.4 の CACHEWEB:SAMPLES データベースに Visual COBOL からアクセスできることを実際に検証しました。

2. 検証環境

ソフトウェア

Windows 7 Professional x86 (32-Bit)

Visual Studio 2010 Professional

InterSystems Caché for Windows (x86-32) 2010.2.4

Micro Focus Visual COBOL R4

ハードウェア

Dell Latitude 520

Intel Core2 Duo T5500 1.66GHz

2.00 Gbyte memory

3. テスト内容

(1) ODBC 経由の Windows ネイティブアプリケーションからのアクセス

CACHÉ の標準インストールに添付されている ODBC データソース “CACHEWEB Samples” を使用して Samples データベース内のテーブルから行を取り出す COBOL プログラムを、

Windows ネイティブとして作成し、実行できることを検証しました。

プログラムの作成は Visual COBOL の OpenESQL アシスタントを使用し、既存のテーブルからほとんどを自動生成しています。

(2) ADO.NET 経由の.NET マネージャアプリケーションからのアクセス

Visual COBOL の ADO.NET Connection Editor によって Caché Managed Provider for .NET を使用したデータソースを、ODBC と同じ “CACHEWEB Samples” という名称で作成することにより、ODBC 接続向けに (1) で作成されたソースプログラムを 1 行も修正せずに、.NET マネージャアプリケーションとしてコンパイル・実行できることを検証しました。

4. 結果

4.1 Visual COBOL R4 のインストール

Visual Studio 2010 Professional をデフォルトインストールした後、Visual COBOL R4 をデフォルトインストールしました。

4.2 CACHÉ インストール

Caché for Windows (x86-32) 2010.2.4 評価版をデフォルトインストールしました。

4.3 プログラム実行結果

上記のテスト内容の両方を実行し、問題は検出されませんでした。

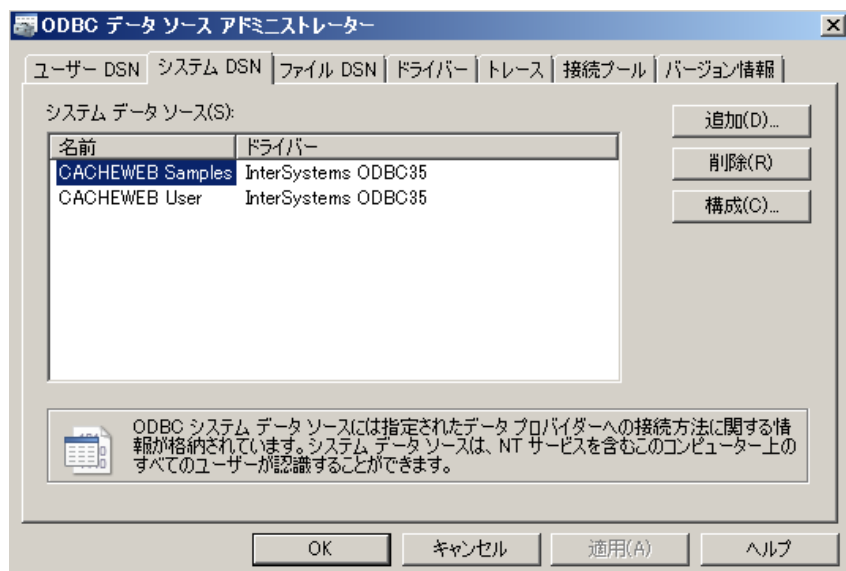
5. テスト結果及び考察

Visual COBOL R4からInterSystems CACHÉデータベースにODBC/ADO.NET経由で COBOL プログラムからアクセスする動作を検証することができました。

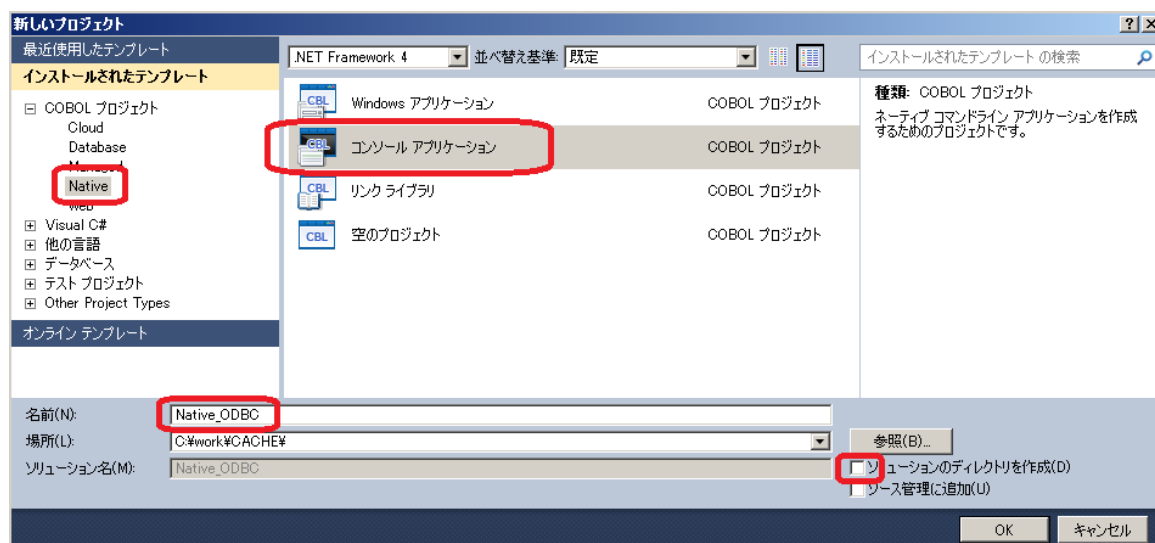
以上

付録-1. ODBC アクセスの検証手順

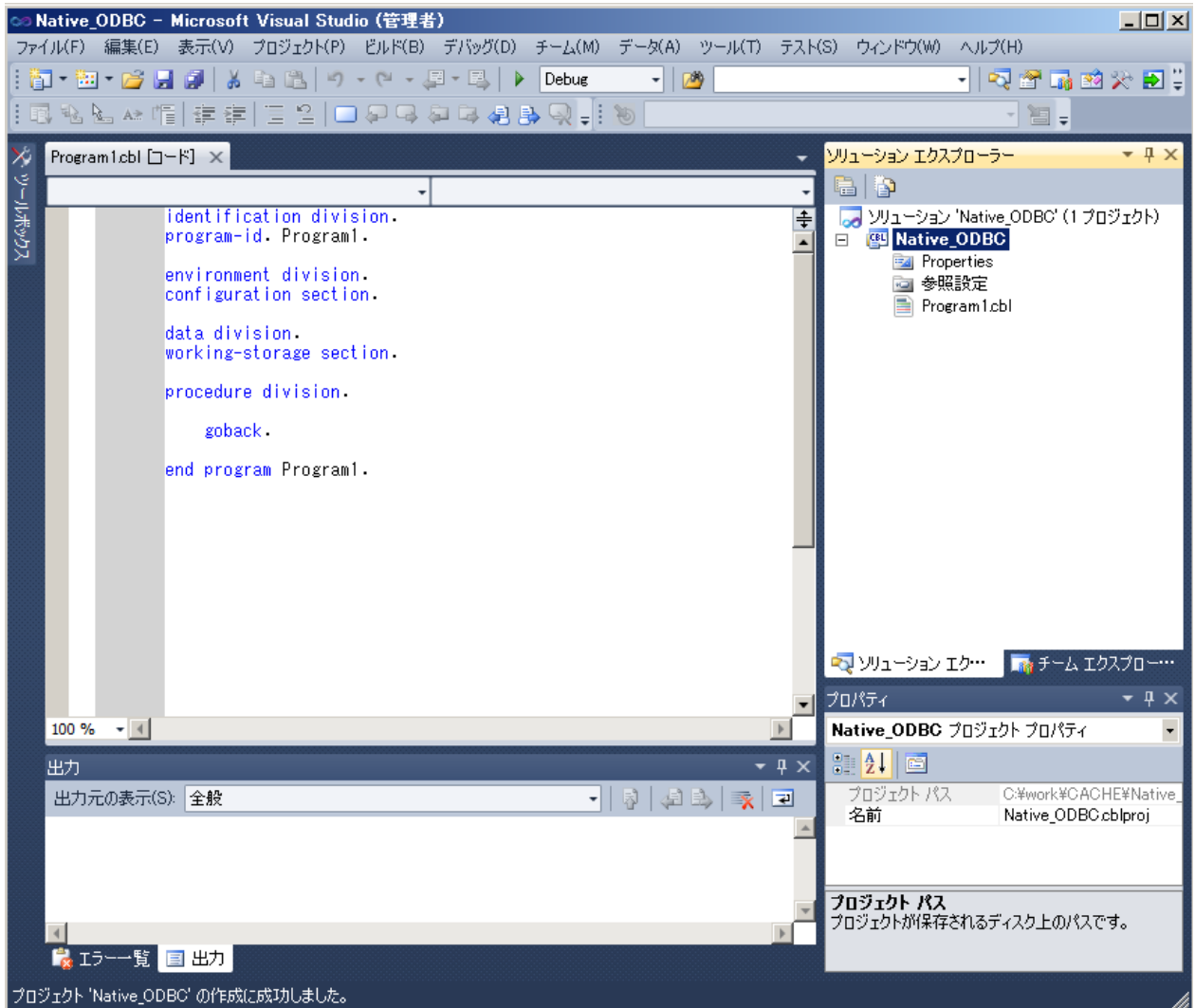
- 1) ODBC データソースアドミニストレータを開き、システムデータソースに“CACHEWEB Samples” が追加されていることを確認します。このデータソースは CACHE のインストール時に自動的に追加されています。



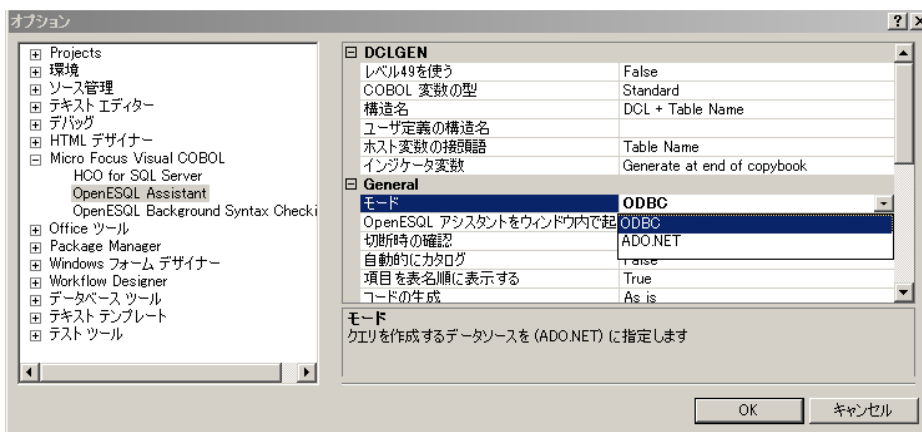
- 2) Visual COBOL for Visual Studio を開き [ファイル] > [新規作成] > [プロジェクト] を選択します。
- 3) 以下の通り [COBOL プロジェクト] の [Native] の [コンソールアプリケーション] を選択し、プロジェクト名を適当に指定します。



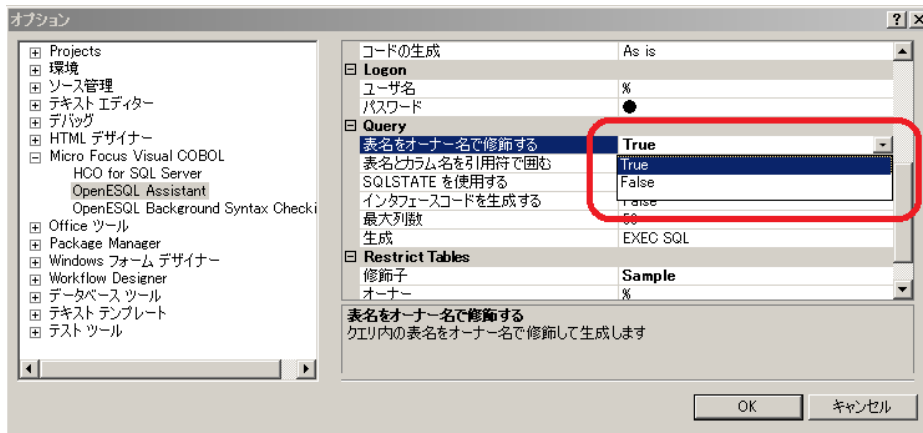
4) 以下のようにテンプレートが作成されます。



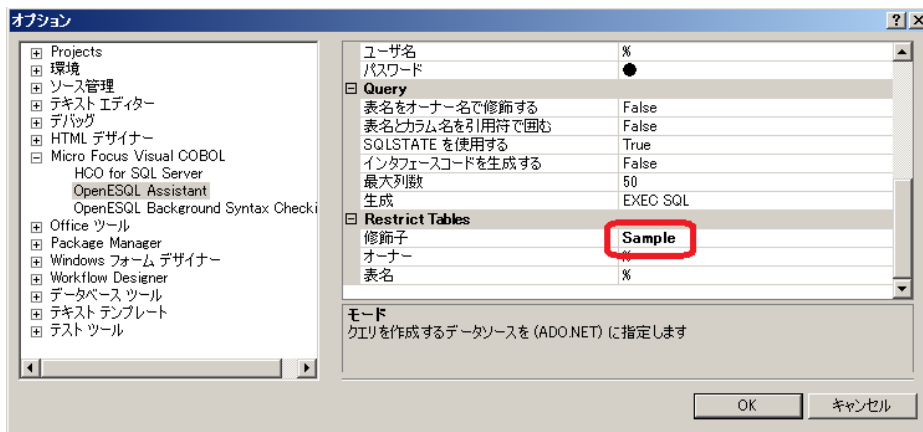
5) プログラムを作成する前に OpenESQL アシスタントを使ったクエリー自動生成に関する環境設定を行います。[ツール] > [オプション...] でオプションダイアログを開き、以下のように左ペインで [Micro Focus Visual COBOL] > [OpenESQL Assistant] を選択し、右ペインの [General] > [モード] で “ODBC” を選択します。これは、アシスタントが ODBC データソースを経由して対象データベースの情報を取得することを指示します。



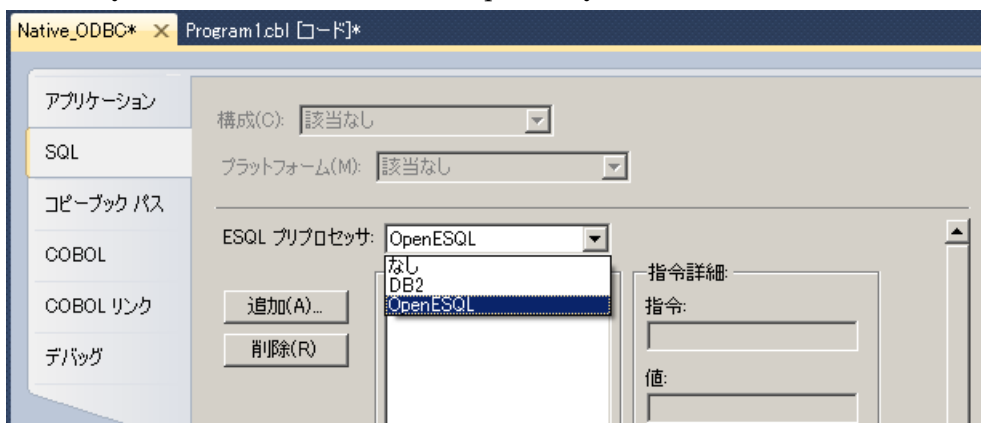
- 6) 同様に、[Query] > [表名をオーナー名で修飾する] で “True” を選択します。これは CACHE の SAMPLES データベース内に含まれている多数の同名のテーブルを一意に識別するために必要です。



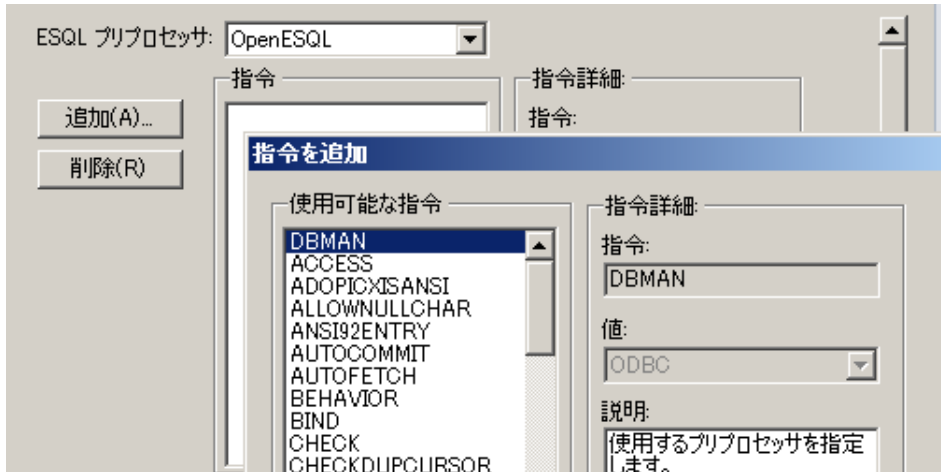
- 7) 同様に、[Restrict Tables] > [修飾子] に “Sample” を入力します。これは、アシスタントが表示するテーブルを SAMPLES データベースの中の一部に限定するために設定しておきます。



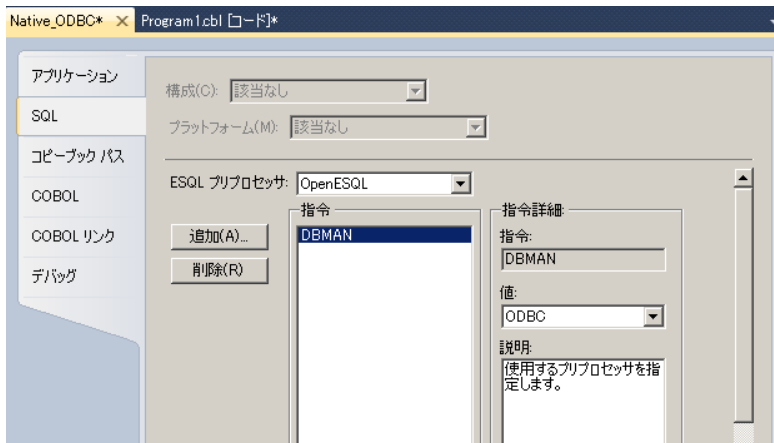
- 8) 次に、データベースアクセスを行うプログラムを開発するためのプロジェクトのプロパティを設定します。ソリューションエクスプローラ内で作成したプロジェクトを右クリックし、[プロパティ] を選択します。プロパティダイアログの [SQL] タブを開き、以下のように [ESQL プリプロセッサ] として OpenESQL を選択します。



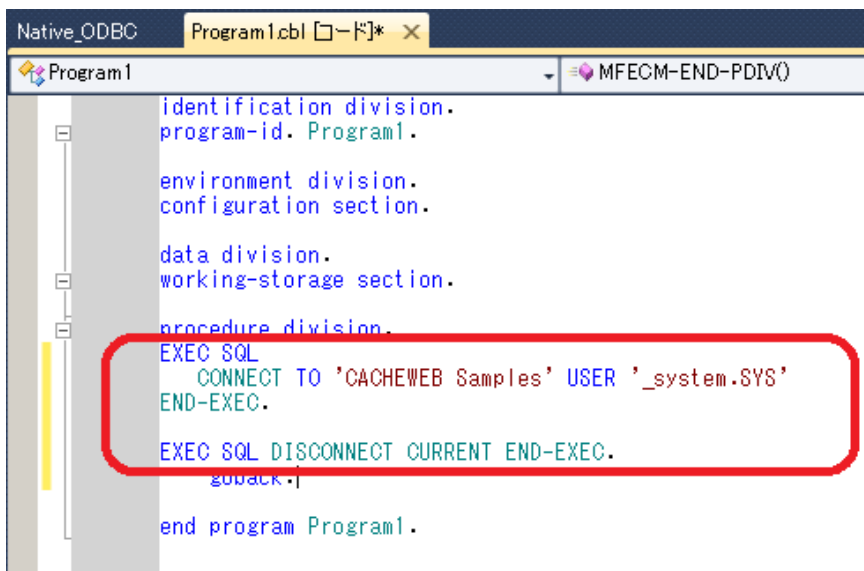
9) [追加...] ボタンをクリックし、以下のように“DBMAN=ODBC” 指令を追加します。



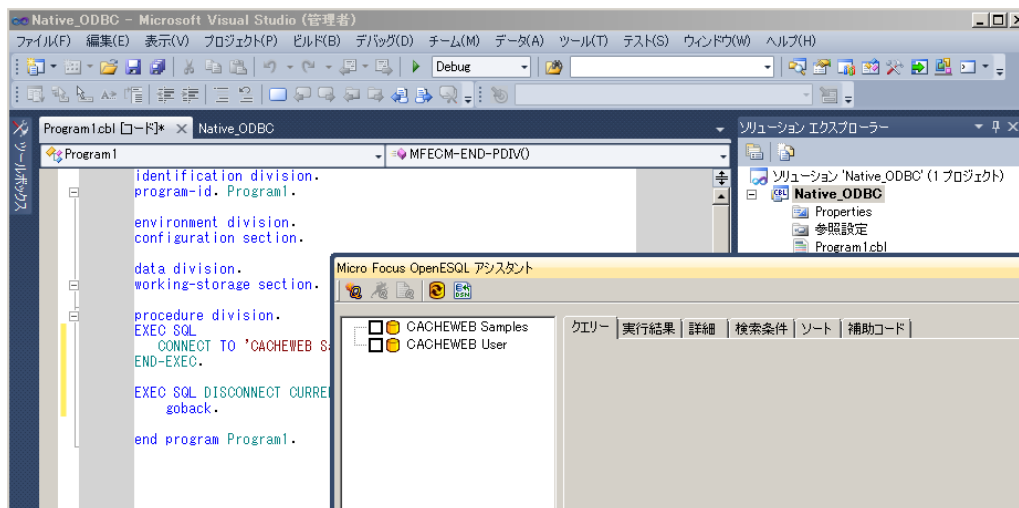
10) プロジェクトの SQL プロパティは以下のようになります。



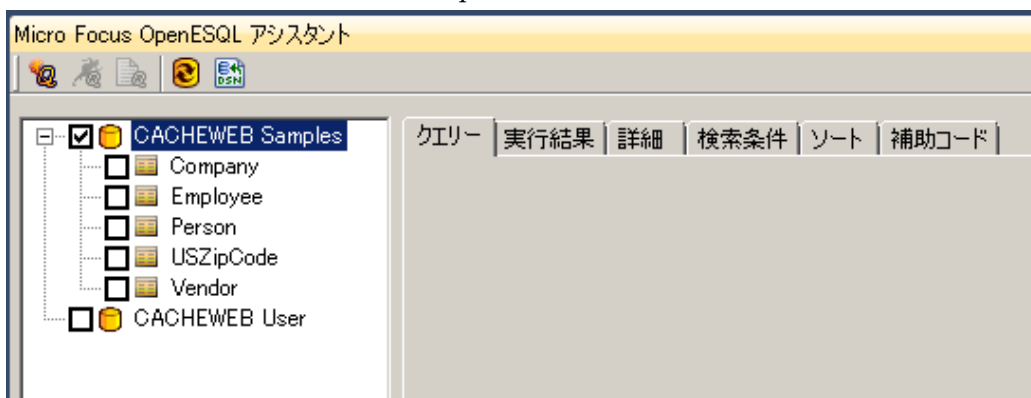
11) 以上でプロジェクトの準備が整いました。続いてプログラムの作成に進みます。まず、生成されているテンプレートの PROCEDURE DIVISION の先頭に以下のように、CACHE データベースへの接続・切断のコードをタイプしてください。



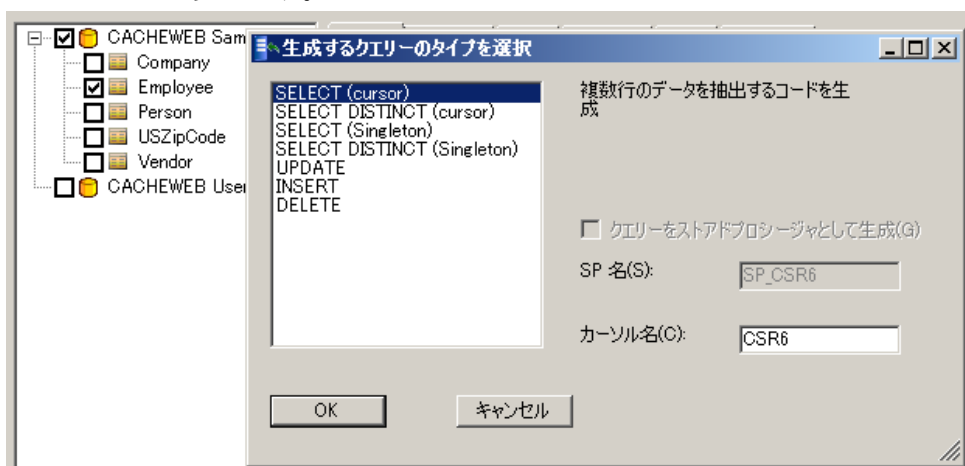
12) OpenESQL アシスタントを使用するとデータベースにアクセスするアプリケーションを大部分自動的に生成することが可能です。[ツール] > [Micro Focus Visual COBOL] > [OpenESQL アシスタント] を選択します。以下のように OpenESQL アシスタントが開きます。初めは Visual Studio にドッキングされて開きますので、ダイアログのタブを右クリックして [フローティング] を選択すると独立ウィンドウで開きます。



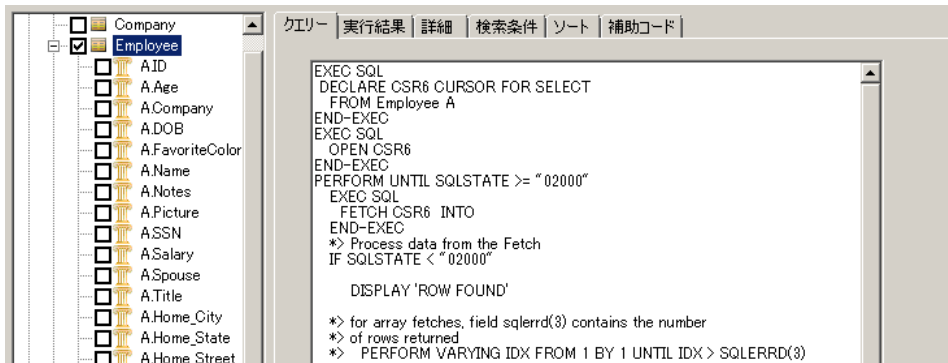
13) 二つの ODBC データソースが表示されています。ここでは “CACHEWEB Samples” をクリックします。以下のように Sample オーナーのテーブルだけが表示されます。



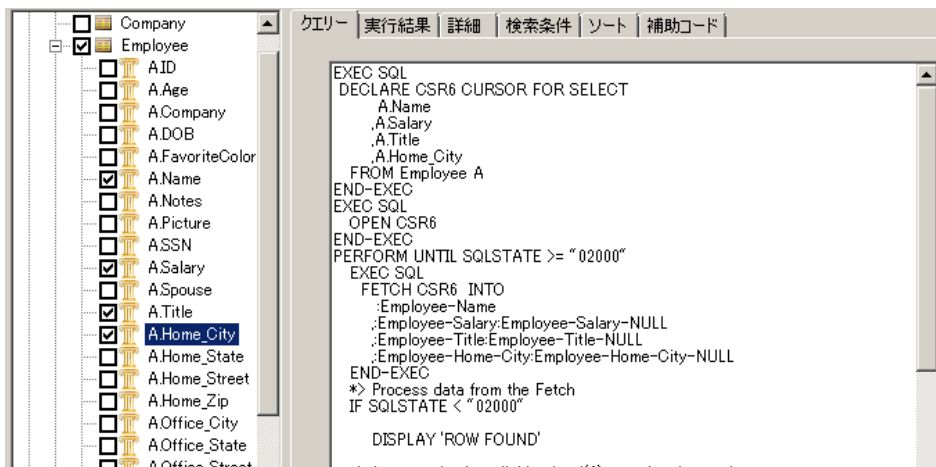
14) “Employee” テーブルをクリックすると以下のように生成するクエリーのタイプを尋ねるダイアログが現れます。



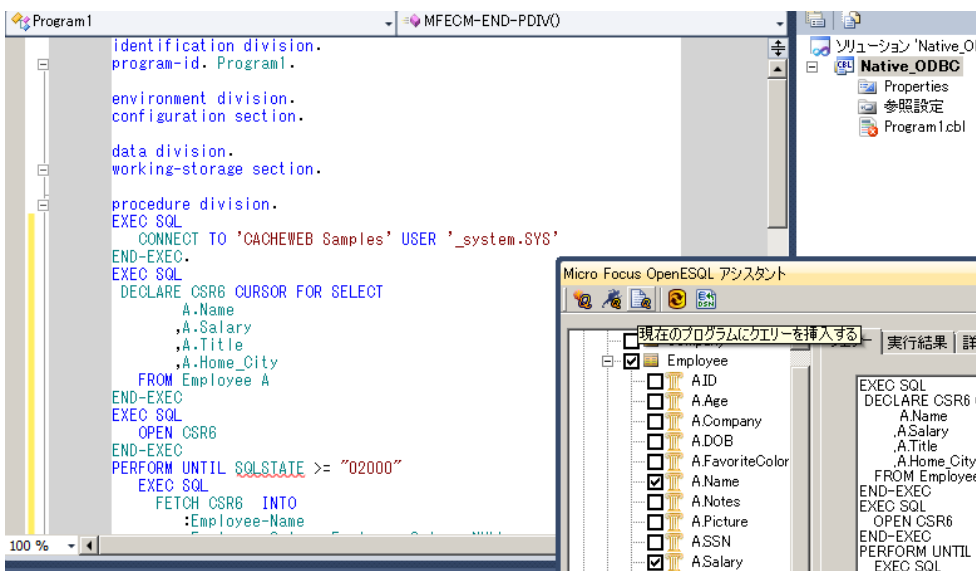
- 15) “SELECT (cursor)” を選択し [OK] をクリックすると、以下のように左ペインのツリーに Employee テーブルの列の一覧が表示され、右ペインに COBOL 埋め込み SQL のひな型が表示されます。



- 16) 左ペインでいくつかの列を順次クリックして行くと、それに対応して右ペインのクエリにも選択した列が追加されてゆきます。



- 17) エディタ上でカーソルを CONNECT と DISCONNECT の間に位置付けておき、OpenESQL アシスタントの [現在のプログラムにクエリーを挿入する] をクリックすると、エディタ上に自動生成されたコードがコピーされます。



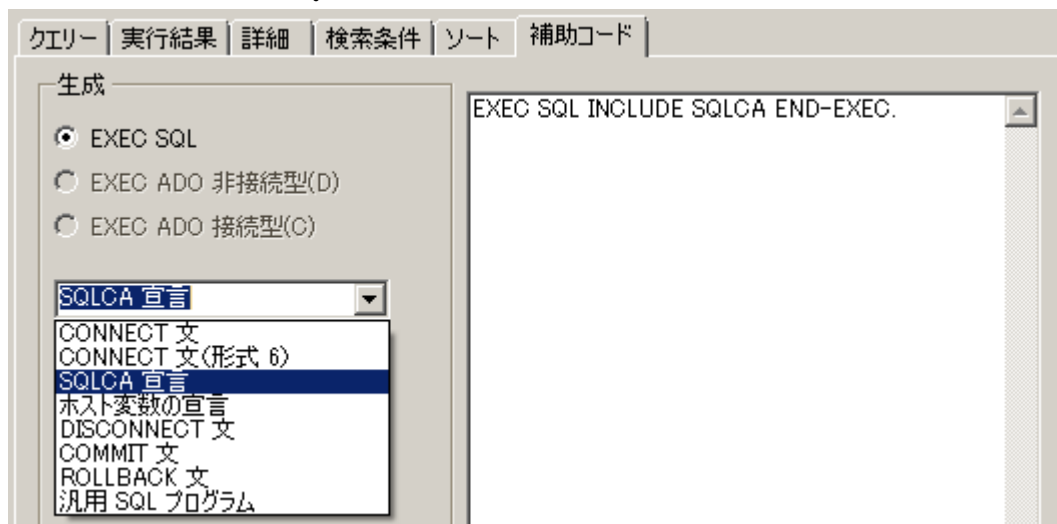
- 18) 生成されたコードはクエリーしているだけで何もしません。そこで以下のような DISPLAY 文を挿入し、取得した行の内容を表示させるようにします。

```
END-EXEC
*> Process data from the Fetch
IF SQLSTATE < "02000"


    DISPLAY Employee-Name "," Employee-Salary ","
           Employee-Title "," Employee-Home-City

*> for array fetches, field sqlerrd(3) contains the number
```

- 19) 以上で手続き部が完成しています。続いてデータ部を作成してプログラムを完成させます。データ部には SQLCA 構造体の宣言と、Employee テーブルのためのホスト変数の宣言が必要です。どちらもアシスタントで自動生成することができます。アシスタントの [補助コード] タブを開き、[SQL 宣言] を選択します。



- 20) エディタでカーソルを WORKING-STORAGE の先頭に位置付けてから、アシスタントの

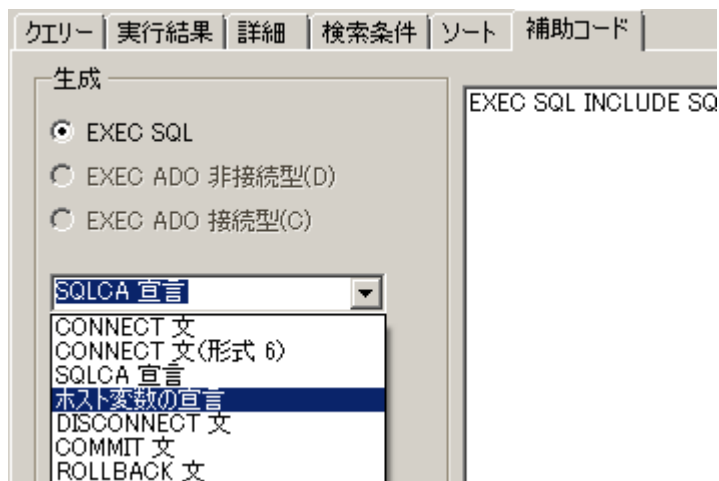
 ボタンをクリックすると、以下のように SQLCA の宣言が挿入されます。

```
configuration section.

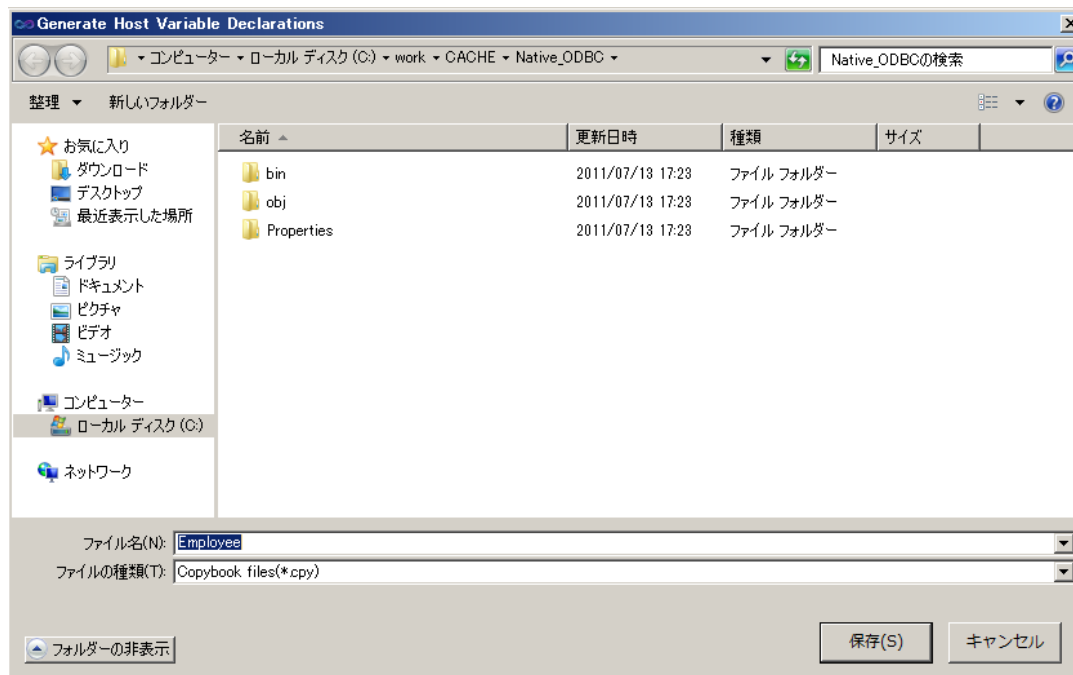
data division.
working-storage section.
EXEC SQL INCLUDE SQLCA END-EXEC.


procedure division.
EXEC SQL
CONNECT TO 'CACHEWEB Samples' USER '_system.SYS'
END-EXEC
```

21) 同じく [補助コード] タブで [ホスト変数の宣言] を選択します。



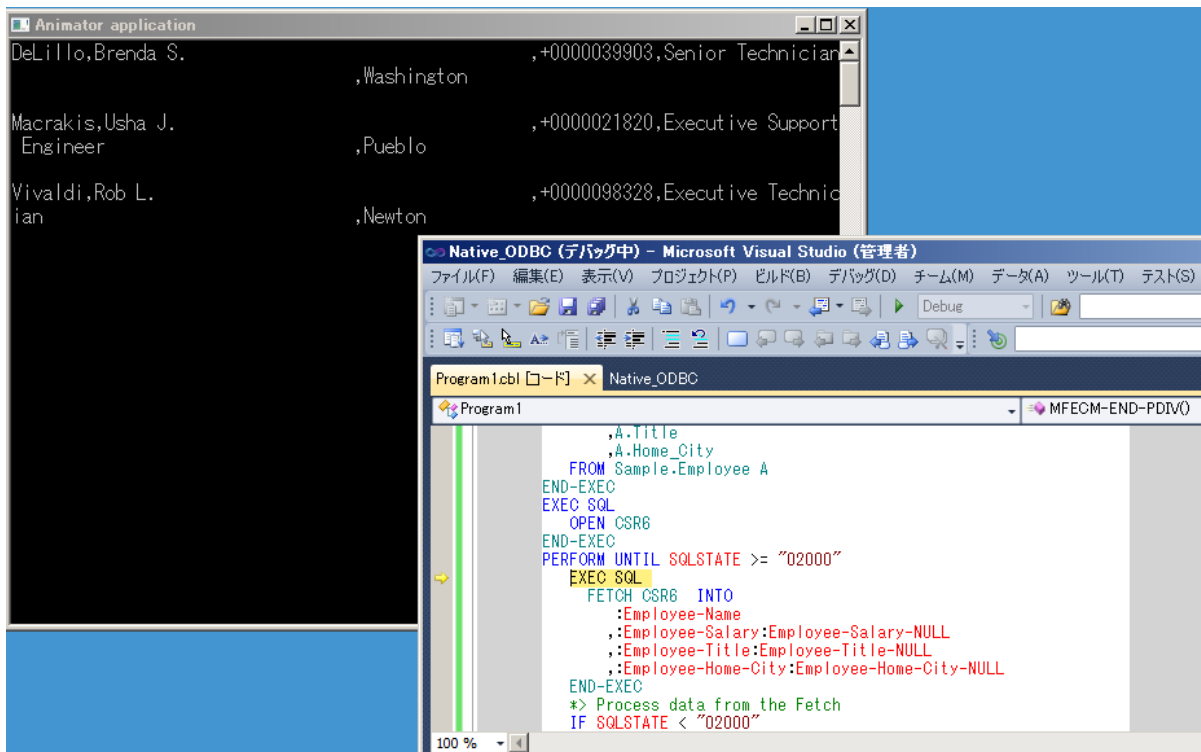
22) ホスト変数の宣言は直接コードに書きこまれるのではなく、他プログラムからも再利用可能な COPY メンバーとして生成され、プログラムにはその COPY 文が生成されます。以下のダイアログは生成される COPY メンバーの置き場所を問い合わせるものです。デフォルトでプロジェクトフォルダが示されていますのでそのまま [保存] をクリックします。



23) 再びアシスタントの  ボタンをクリックして、以下のように WORKING-STORAGE にホスト変数宣言の COPY 文を貼り付けます。

```
data division.  
working-storage section.  
EXEC SQL INCLUDE SQLCA END-EXEC.  
EXEC SQL INCLUDE Employee END-EXEC.  
  
procedure division.  
EXEC SQL
```

24) 以上でプログラムは完成しています。[F11]キー を打鍵し、プログラムをステップ実行して行くと以下のように、CACHE の Employee テーブルから一行ずつ取り出し DISPLAY 文でコンソールに結果を出力して行く様子が確認できます。



付録-2. ADO.NET アクセスの検証手順

Caché Managed Provider for .Net は CACHÉ データベースへのアクセスを提供する ADO.NET データプロバイダです。現在のところ .NET Framework 2.0 での提供となっていますが、.NET Framework 4.0 で稼働する COBOL のアセンブリからも利用することができます。ODBC 経由のネイティブな COBOL プログラムでも、ADO.NET 経由のマネージな COBOL プログラムでもコードの記述方法は変わりません。実際に、付録-1 で作成したプログラムをそのまま使用して、ADO.NET 経由のマネージプログラムにコンパイルして実行することができます。

- 1) Caché Managed Provider for .Net を利用可能とするために、CACHÉ ドキュメンテーションの以下の記述に従って `machine.config` を修正します。

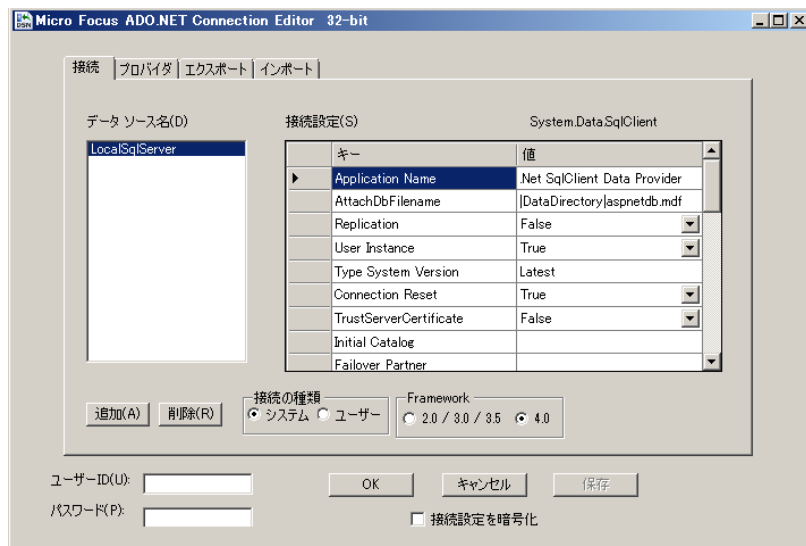
[Caché 言語バインディングおよびゲートウェイ] > [Caché Managed Provider...] > [Caché Managed Provider の設定] の「システム構成」の項

- 2) Caché Managed Provider for .Net をグローバルアセンブリに登録し、Visual COBOL の ADO.NET 接続エディタから参照可能にします。Windows スタートメニューから [すべてのプログラム] > [Microsoft Visual Studio 2010] > [Visual Studio Tools] > [Visual Studio コマンド プロンプト(2010)] で Visual Studio コマンドプロンプトを開き、Administrator 権限で以下のコマンドを実行します：

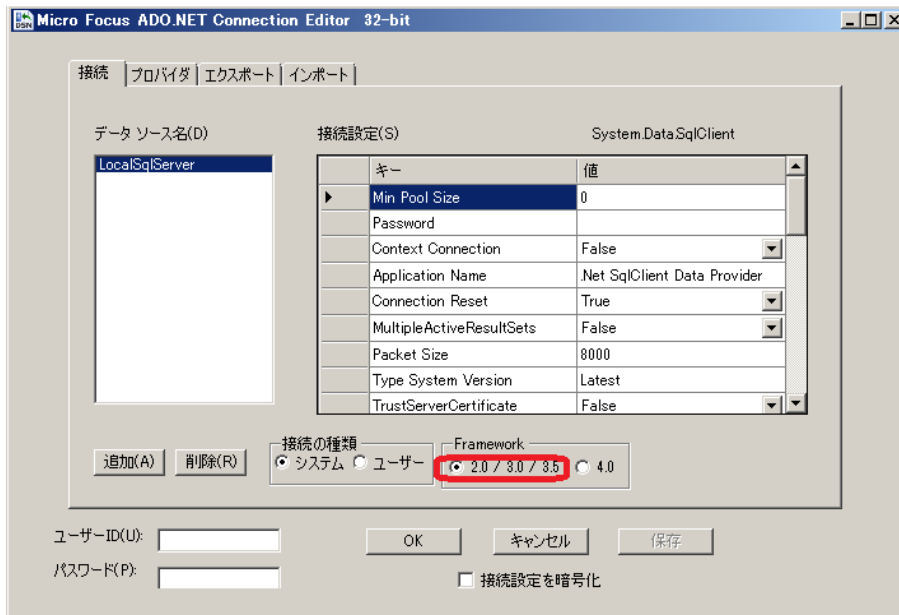
```
> cd C:\InterSystems\Caché\dev\dotnet\bin
> gacutil /i InterSystems.Data.CachéClient.dll
```

`gacutil` コマンドの詳細については、.NET Framework SDK のドキュメントを参照してください。

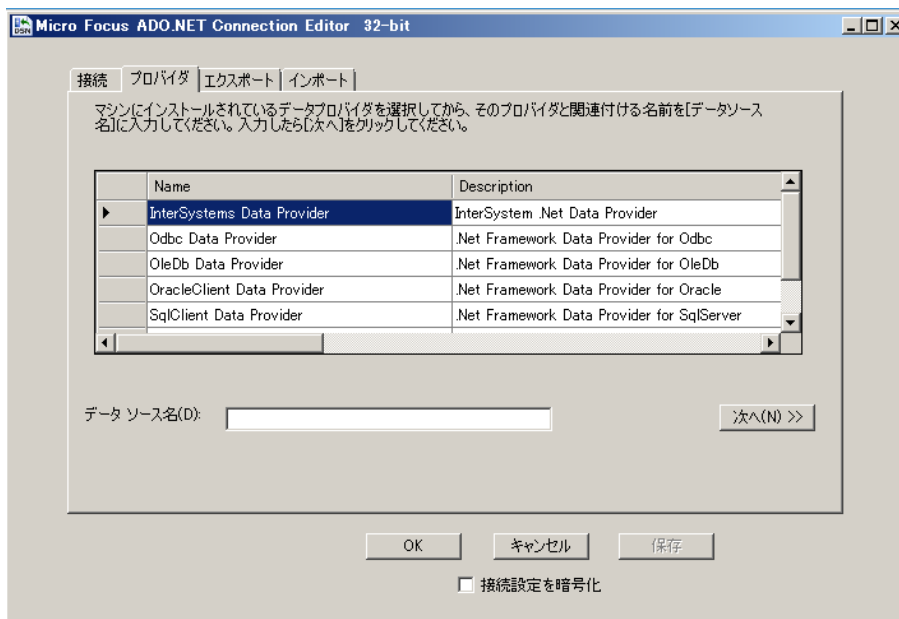
- 3) Caché Managed Provider for .Net を使用したデータソースを Visual COBOL ADO.NET 接続エディタで定義します。Windows スタートメニューから [すべてのプログラム] > [Microsoft Visual COBOL 2010] > [Visual COBOL データ接続] > [ADO.NET Connection Editor] を選択します。以下のダイアログが開きます。



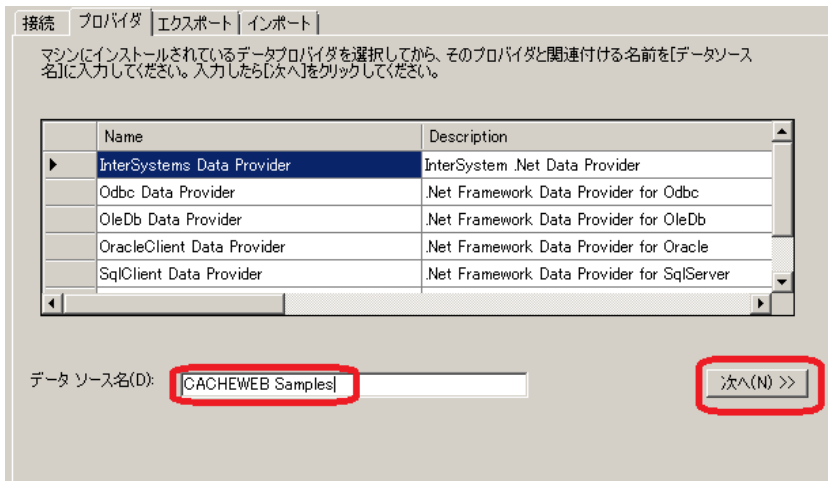
- 4) CACHÉ のプロバイダは .NET Framework 2.0 対応ですので、Framework の指定をデフォルトの 4.0 から以下のように変更します。



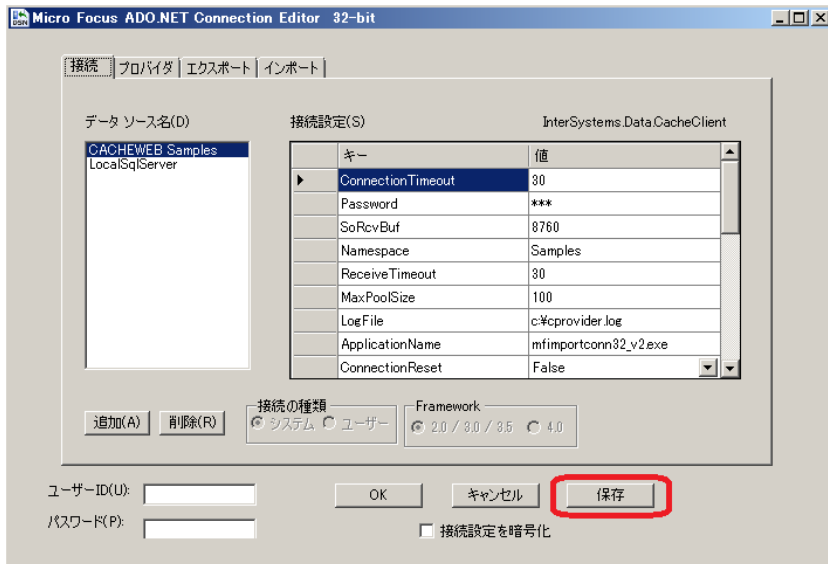
- 5) [プロバイダ]タブを開くと以下のように CACHÉ のプロバイダが表示されていることが確認できます。



- 6) このプロバイダを使用したデータソースを作成するため、データソース名に“CACHEWEB Samples”を入力し、[次へ]をクリックします。ODBC データソースと同じ名前を指定することにより、ODBC 用に作成されたソースプログラムを一切修正せずに利用できるようになります。

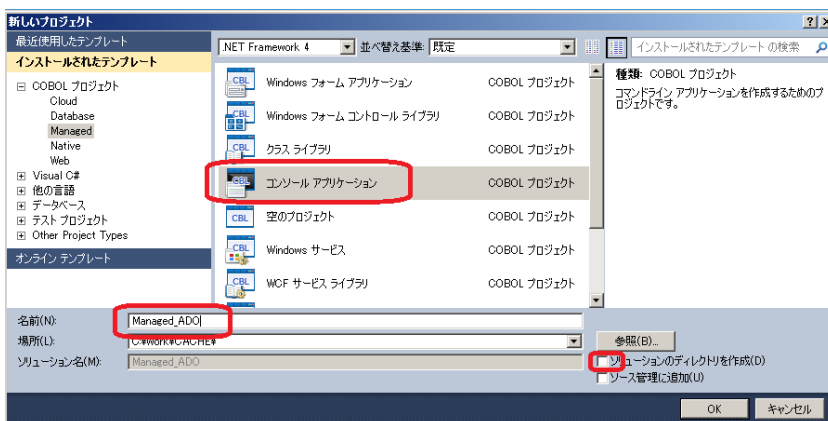


- 7) 以下のように新しいデータソース名が作成されますので [保存] をクリックしてこれを登録済みデータソースとして保存します。

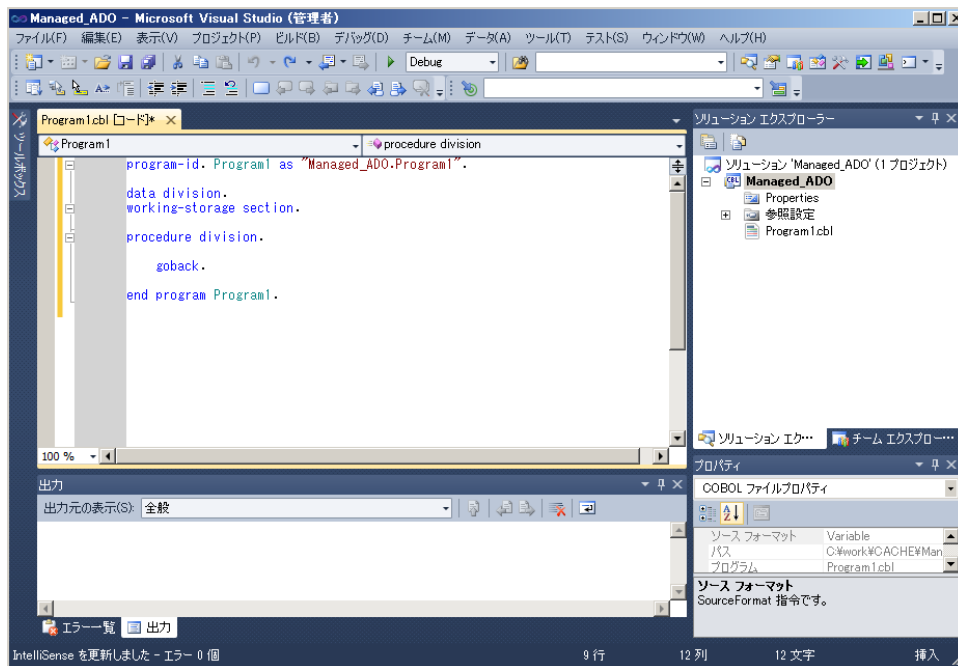


- 8) Visual COBOL for Visual Studio を開き [ファイル] > [新規作成] > [プロジェクト] を選択します。

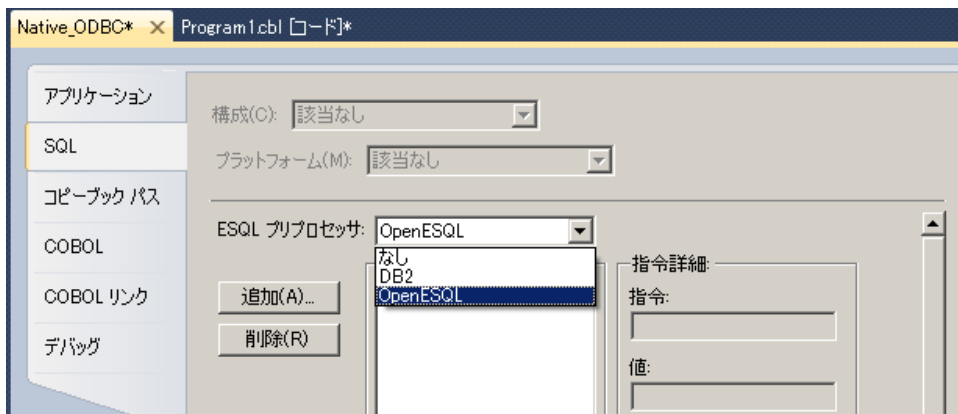
- 9) 以下の通り[COBOL プロジェクト] の [Managed] の [コンソールアプリケーション] を選択し、プロジェクト名を適当に指定します。



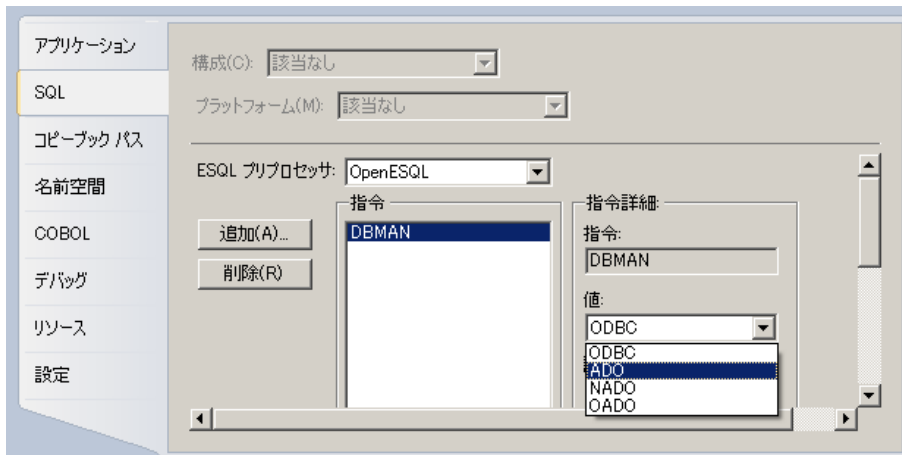
10) 以下のようにテンプレートが作成されます。



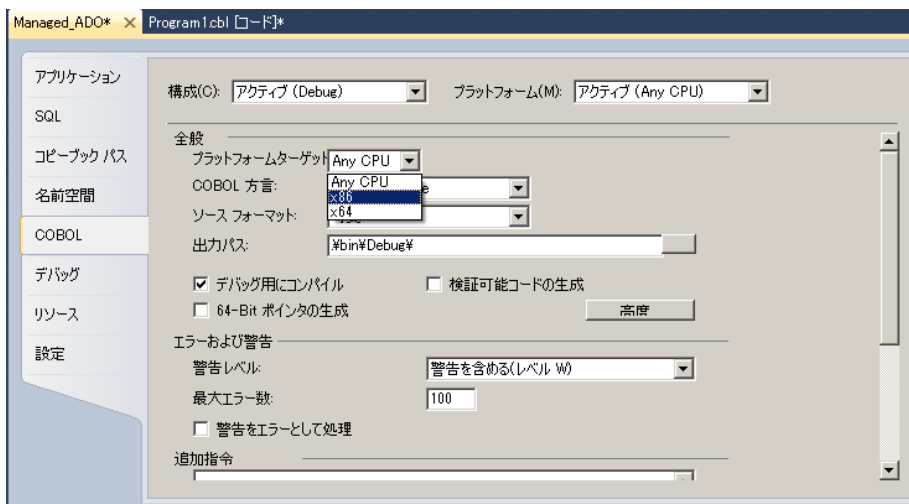
11) データベースアクセスを行うプログラムを開発するためのプロジェクトのプロパティを設定します。ソリューションエクスプローラ内で作成したプロジェクトを右クリックし、[プロパティ] を選択します。プロパティダイアログの [SQL] タブを開き、以下のように [ESQL プリプロセッサ] として OpenESQL を選択します。



- 12) [追加...] ボタンをクリックし、DBMAN 指令を追加した後以下のように ADO 指令に変更します。



- 13) ADO.NET を使用する COBOL アプリケーションでは、AnyCPU ではなくプラットフォームターゲットを明示的に指定する必要があります。プロジェクトのプロパティで「COBOL」タブを開き、以下のようにプラットフォームターゲットを「x86」に設定します。

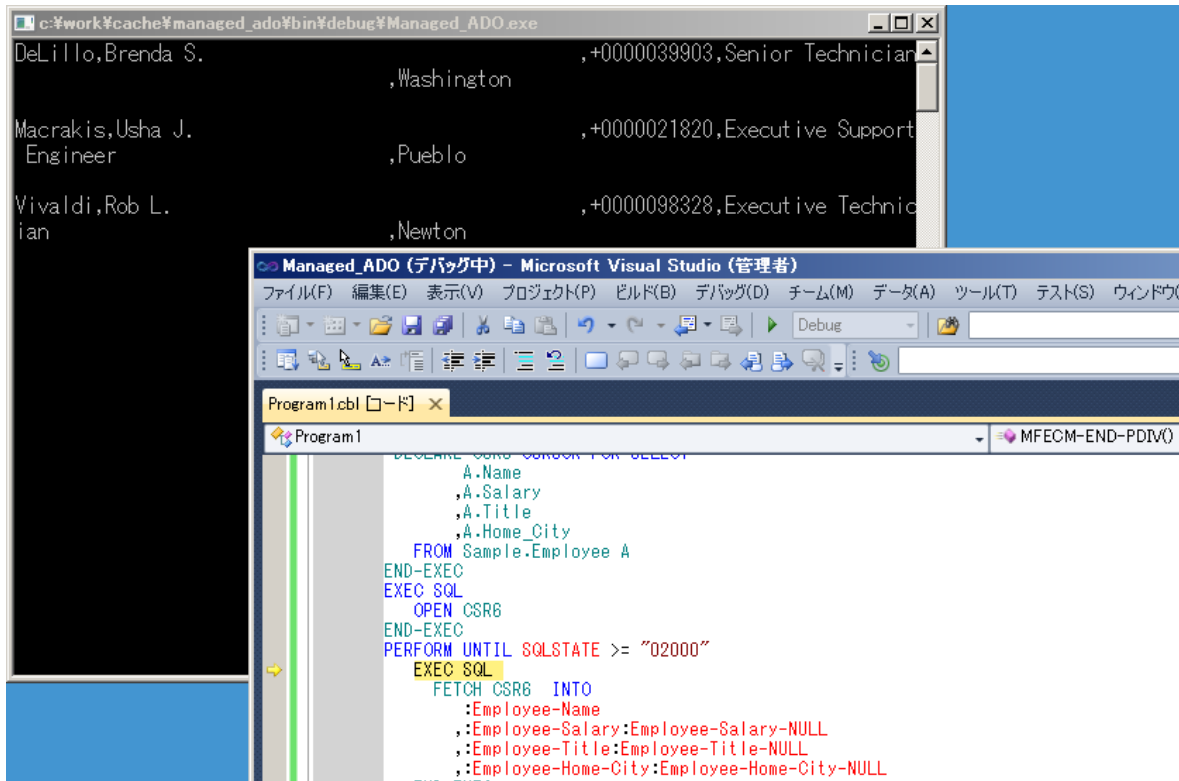


- 14) プログラムは「付録-1」で作成したものをそのまま使用しますので、プロジェクトにインポートします。テンプレートで生成された Program1.cbl は削除します。

- 15) ソリューションエクスプローラでプロジェクトを右クリックし[追加] > [既存の項目] を選択します。「付録-1」で作成された Program1.cbl と Employee.cpy を選択し [追加] をクリックすると以下の通りプロジェクトに同じソースが追加されます。



16) [F11]キーを打鍵し、プログラムをステップ実行して行くと以下のように、CACHÉの Employee テーブルから一行ずつ取り出し DISPLAY 文でコンソールに結果を出力して行く様子が確認できます。



以上