

Micro Focus Visual COBOL R4 InterSystems CACHÉ 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 オ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

 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 CACHÉ のインストー ル時に自動的に追加されています。

ドミニストレーター	×
SN ファイル DSN ドライバー トレース 接続プー	ル「バージョン情報」
	追加(D)
ドライバー	200A(D)
InterSystems ODBC35	
InterSystems ODBC35	構成(C)
ム テーダ ソースには指定されたテーダ ノロハイダーへの ています。 システム データ ソースは、 NT サービスを含む	J接続方法に関する情 3このコンピューター上の
「一が認識することができます。	
OK キャンセル 適用	(A) ^//プ
	ドミストレーター SN ファイル DSN ドライバー トレース 接続プー ドライバー InterSystems ODBC35 InterSystems ODBC35 InterSystems ODBC35 InterSystems ODBC35 InterSystems ODBC35 ム データ ソースには指定されたデータ プロバイダーへのています。システム データ ソースは、NT サービスを含む (小が認識することができます。) 通用 OK キャンセル 通用

- 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。

新しいプロジェクト			? ×
最近使用したテンプレート	NET Framework 4 • 並べ替え基準: 既定		インストールされたテンプレートの検索
インストールされたテンプレート			### copol ≓ar% bi
□ COBOL プロジェクト Object	Windows アプリケーション	COBOL プロジェクト	種類: COBOL ノロシェクト ネーティブ ユマンドライン アプリケーションを作成
Database	עבע-א אילא אי-אעב	COBOL プロジェクト	するためのプロジェクトです。
Native	リンク ライブラリ	COBOL プロジェクト	
 H Visual C# 他の言語 データベース デストプロジェクト Tストプロジェクト H Other Project Types 	<u>CBL</u> 空のプロジェクト	COBOL プロジェクト	
オンライン テンプレート			
名前(N): Native_ODBC			
場所(L): C:¥work¥CACHE	¥		参照(B)
ソリューション名(M): Native_ODBC		Ģ	ソコーションのディレクトリを作成(D) ソース管理(に追加(U)
			OK キャンセル

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。

オプション			<u>?</u> ×
オプジョン Projects 環境 ジース管理 テキストエディター デパッグ HTMLデザイナー Micro Focus Visual COBOL HCO for SOL Server OpenESOL Assistant OpenESOL Background Syntax Checki Office ツール Packase Manager Windows フォームデザイナー Workflow Designer データベース ツール テキスト デンプレート テスト ツール	 □ DCLGEN レベル49を使う COBOL 変数の型 構造名 ユーザ定義の構造名 ホスト変数の接閉語 インジケータ変数 □ General モード の のpenESQL アシスタントをウィンド 切断時の確認 自動的にカタログ 項目を表名順に表示する コードの生成 モード クロリ を作成するデータソースを(ADO 	False Standard DCL + Table Name Table Name Generate at end of copybook ODBC ODBC ADONET True As is	
× •		ОК	キャンセル

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 CACHÉ 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。

オプション			?)
 Projects □ 環境 □ ソース管理 □ テキスト エディター □ デキスト エディター 	コードの生成 日 Logon ユーザ名 パスワード	Asis %	×
 ● JTMJ デザイナー ● HTML デザイナー ■ Micro Focus Visual COBOL HCO for SQL Server OpenESQL Assistant OpenESQL Background Syntax Checki ● Office ツール ● Package Manager ● Windows フォーム デザイナー ● Workflow Designer ● データペース ツール ● デスト デンプレート ● テスト ツール 	 □ Cuery 素名をオーナー名で修飾する 表名とわうん名を引用符で囲む SOLSTATE を使用する インタフェースコードを生成する 最大列数 生成 □ Restrict Tables (修飾子 オーナー ま名をオーナー名で修飾する クエリ内の表名をオーナー名で修飾して 	True True False False EXEC SQL Sample 炎 生成します	
		ОК	キャンセル

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分に限定するために設定しておき

ます。

オプション			<u>? ×</u>
 Projects 環境 ソース管理 テキストエディター デドメクー デドメグー デドメグー デドメグー Micro Focus Visual COBOL HTML デザイナー Micro Focus Visual COBOL HOC for SOL Server OpenESOL Assistant OpenESOL Background Syntax Checki Office ツール Package Manager Windows フォーム デザイナー Workflow Designer データベース ツール ティスト テンプレート テスト ツール 	ユーザ名 パスワード ■ Query 表名をオーナー名で修飾する 表名をオーナー名で修飾する ひタフェースコードを生成する 最大列版 生成 ■ Restrict Tables 修飾子 オーナー 表名 モード クエリを作成するデータソースを (ADO NE	が False False True False 50 EXEC SQL Sample 派	
I		OK ##2	1211

 ※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。

Native_ODBC* × F	Program1cbl [□−ド]*
アプリケーション	構成(C): 該当なし
SQL	ブラットフォーム(M): 該当なし
コピーブック パス	
COBOL	ESQL ブリプロセッサ: OpenESQL なし、 本し、 上 指令詳細・ 一
COBOL リンク	DB2 Dbhn(A) DD2 OpenESQL 指令:
デバッグ	

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。

ESQL プリプロセッサ:	OpenESQL 指令	令詳細:
<u>追加(A)</u>	指4 指令を追加	<u>};</u>
H IPA (FV	使用可能な指令 DBMAN ACCESS ADOPICXISANSI ALLOWNULLCHAR ANSI92ENTRY AUTOCOMMIT AUTOFETCH BEHAVIOR BIND CHECK CHECKDUPCURSOR	指令詳細: 指令: DBMAN 値: ODBC 文 説明: 使用するプリプロセッサを指定 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。

Native_ODBC* ×	Program1cbl []-ド]*
Native_ODBC* × アプリケーション SQL コピーブック パス COBOL	構成(C): 該当なし ・ ブラットフォーム(M): 該当なし ・ ESQL プリプロセッサ: OpenESQL ・
COBOL リンク デバッグ	指令 道助(A) 肖顺余(R)

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、CACHÉ 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] > [Open ESQL 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。

Micro Focus OpenESQL アシスタント	
1 🙋 🦓 🗽 😢 🚮	
CACHEWEB Samples	クエリー 実行結果 詳細 検索条件 ソート 補助コード
Company	
USZIPCode	
CACHEWEB User	

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。

CACHEWEB Sam	生成するクエリーのタイプを選択		_ 🗆 >	[]
	SELECT (cursor) SELECT DISTINCT (cursor) SELECT (Singleton) SELECT DISTINCT (Singleton) UPDATE INSERT DELETE	複数行のデータを抽 成	出するコードを生	
		🗖 クエリーをストアト	"プロシージャとして生成(G)	
		SP 名(S):	SP_CSR6	
		カーソル名(C):	CSR6	
[[OK キャンセル]		
				//

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 テーブルのカラムの一覧が表示され、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カラム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



17) エディタ上でカーソルを CONNCET 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。 % Program1



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。



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。



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。

👓 Generate Host Variable	e Declarations					×
א-באנב • 🖉	!ー ▼ ローカル ディスク (C:) ▼ work ▼ CACHE ▼ Native_	ODBC -	🔻 🚰 🛛 Native	_ODBCの検索		2
整理 👻 新しいフォルダー					•	0
👉 お気に入り	名前 🔺	更新日時	種類	サイズ		
〕 ダウンロード	鷆 bin	2011/07/13 17:23	ファイル フォルダー			
■ デスクトップ	鷆 обј	2011/07/13 17:23	ファイル フォルダー			
置 取近数示した物用	鷆 Properties	2011/07/13 17:23	ファイル フォルダー			
 ⇒ ライブラリ ※キュメント ※ ピクチャ ピグチャ ビデオ シュージック シュージューター ニーカルディスク(C·) ネットワーク 						
ファイル名(N): Emplo	yee					•
ファイルの種類(T): Copyb	book files(*cpy)					-
● フォルダーの非表示			保存	F(S) =	キャンセノ	L

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.

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、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。

,+0000039903,Senior Technician▲ ,Washington	
,+0000021820,Executive Support	
,+0000098328,Executive Technic ,Newton	
∞ Native_ODBC (デバッグ中) - Microsoft Visual Studio (管理者)	
ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) チーム(.M) データ(A) ツール(T) テスト(S)
🔄 = 🔤 = 🚅 🛃 🦼 🔉 🖿 🛍 🌱 = 🔍 = 🜉 🕨 Debu	e - 🖄
🖪 💫 🏊 👘 津 津 🗉 😫 🗆 🖓 🤤 🚑 🛃 🔍	- I 🐌 🛛
Program1cbl [] - F] × Native_ODBC	
An Antonia	■ ● MFECM-END-PDIV()
<pre></pre>	r-NULL
	、Washington 、+0000039903,Senior Technician 、+0000021820,Executive Support 、Pueblo 、+0000098328,Executive Technic 、Newton Newt

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ることがありません。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

 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。

 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] > [Micro Focus 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。

データ ソース名(D)	接続調	设定(S)	System.Data.SqlClient	
LocalSqlServer		キー	値	-
	•	Application Name	Net SqlClient Data Provider.	
		AttachDbFilename	DataDirectory aspnetdb.mdf	
		Replication	False 💌	
		User Instance	True	
		Type System Version	Latest	
		Connection Reset	True	I
		TrustServerCertificate	False	I
		Initial Catalog		
J		Failover Partner		-
追加(A) 削除(R)	ー接続の種類 ● システム	C ユーザー Framework C 2.0 / 3.0 / 3.5	● 4.0	

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。

データ ソース名(D)	接続	設定(S)	System.Data.SqlClient	
LocalSqlServer		+-	値	•
	•	Min Pool Size	0	
		Password		
		Context Connection	False 💌	
		Application Name	Net SqlClient Data Provider	
		Connection Reset	True 💌	
		MultipleActiveResultSets	False 💌	
		Packet Size	8000	
		Type System Version	Latest	
J		TrustServerCertificate	False 🔻	•
追加(A) 肖耶徐(R)	ー接続の種類 ◎ システム	C ユーザー Framework C ユーザー 10 2.0 / 3.0 / 3.5	C 4.0	

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。

	Name	Description
•	InterSystems Data Provider	InterSystem .Net Data Provider
	Odbc Data Provider	.Net Framework Data Provider for Odbc
	OleDb Data Provider	Net Framework Data Provider for OleDb
	OracleClient Data Provider	Net Framework Data Provider for Oracle
	SqlClient Data Provider	.Net Framework Data Provider for SqlServer
-9	ソース名(D):	<u>次</u> へ(N) >>

 C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。

InterSystems Data Provider InterSystem Net Data Provider Odbc Data Provider Net Framework Data Provider for Odbc OleDb Data Provider Net Framework Data Provider for OleDb OracleClient Data Provider Net Framework Data Provider for Oracle SqlClient Data Provider Net Framework Data Provider for SqlServer	Name		Description
Odbc Data Provider Net Framework Data Provider for Odbc OleDb Data Provider Net Framework Data Provider for OleDb OracleClient Data Provider Net Framework Data Provider for Oracle SqlClient Data Provider Net Framework Data Provider for SqlServer	InterSys	tems Data Provider	InterSystem .Net Data Provider
OleDb Data Provider Net Framework Data Provider for OleDb OracleClient Data Provider Net Framework Data Provider for Oracle SqlClient Data Provider Net Framework Data Provider for SqlServer	Odbo Da	ata Provider	.Net Framework Data Provider for Odbc
OracleClient Data Provider Net Framework Data Provider for Oracle SqlClient Data Provider Net Framework Data Provider for SqlServer	OleDb D)ata Provider	.Net Framework Data Provider for OleDb
SqlClient Data Provider .Net Framework Data Provider for SqlServer	OracleC	lient Data Provider	Net Framework Data Provider for Oracle
	SqlClien	t Data Provider	Net Framework Data Provider for SqlServer

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。

データ ソース名(D)	接続調	发定(S)	InterSystems.Data.CacheClient
CACHEWEB Samples LocalSqlServer		+-	值
	•	ConnectionTimeout	30
		Password	***
		SoRcvBuf	8760
		Namespace	Samples
		ReceiveTimeout	30
		MaxPoolSize	100
		LogFile	c:¥cprovider.log
		ApplicationName	mfimportconn32_v2.exe
		ConnectionReset	False 🗾 💌
追加(A) 削除(R)	接続の種類 © システム	・ コーザー 「 ユーザー 「 ユーザー」 「 2,0 / 3,0 / 3,5	C 4.0

- 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。

新しいプロジェクト		<u>? ×</u>
最近使用したテンプレート	NET Framework 4 ・ 並べ替え基準: 既定	💌 🏢 インストールされたテンプレート の検索 👂
インストールされたテンプレート		▲ 新聞・COBOL プロジェクト
□ COBOL プロジェクト	Windows フォーム アブリケーション	COBOL プロジェクト コマンドライン アプリケーションを作成するためのプ
Database	Windows フォーム コントロール ライブラリ	COBOL ブロジェクト
Managed		
Native Web	CBL クラス ライブラリ	COBOL プロジェクト
Visual C# Mon===	ユンソール アプリケーション	COBOL プロジェクト
 田 把の言語 田 データベース 		
E テスト プロジェクト	<u>CBL</u> 空のプロジェクト	COBOL プロジェクト
Other Project Types	CBL Windows サービス	COBOL プロジェクト
オンライン テンノレート		
	WCF サービス ライブラリ	COBOL プロジェクト
名前(N): Managed ADO		
場所(L): C#MORK#CACHE	,	▼ 参照(B)
ソリューション名(M): Managed_ADO		
		■ ソース管理(c追加(U)
		OK キャンセル

10) 以下のようにテンプレートが作成されます。

⇔ Managed_ADO - Microsoft Visual Studio (管理者)	
ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) データ(A) ツール(T) う	テスト(S) ウィンドウ(W) ヘルプ(H)
: 🛅 = 🔠 = 💋 🛃 🦪 🐇 ங 🛝 🔊 = 🝽 = 💭 = 🛤 🕨 Debug 🔹 🧭	- 🛛 💀 🖓 🖬 📸 🏷 🔜 🖳
國 勉 🍆 🗠 帽 崇 崇 🗉 일 🗆 위 다 의 다 의 다 의 및 🚽 🐌	- Te -
/タ Program1cbl [コード]* ×	✓ ソリューション エクスプローラー ▼ # ×
🝸 😪 Program 1 🗸 🔹 🖓 procedure division	🚽 🖶 😰
📮 😑 program-id. Program1 as "Managed_ADO.Program1".	돶 🌄 ソリューション 'Managed_ADO' (1 プロジェクト)
data division.	Managed_ADO
working-storage section.	■ Toperius ■ ● ● 照設定
procedure division.	Program 1.cbl
goback -	
end program Program1.	
	マ マ
100 % 👻 4	
	▼ # × COBOL 77√ルプロパティ ・
出力元の表示(S): 全般	
	パス C:¥work¥CACHE¥Man
	プログラム Program1.cbl
	Jース フォーマット SourceFormat 指令です。
□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	
	0.4 10.71 10.大字 括3
Intellisense を使動しよした。- エラー 0 1回	

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。

Native_ODBC* ×	Program1cbl [コード]*
アプリケーション	構成(C): 該当なし
SQL	ブラットフォーム(M): 該当なし
コピーブック パス	
COBOL	ESQL プリプロセッサ: OpenESQL なし おし 「指令詳細
COBOL リンク	DB2 这加(A) DD2 DoenESQL 指令:
デバッグ	

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。

アプリケーション SQL	構成(C): 該当なし
コピーブック パス	ESQL プリプロセッサ: OpenESQL
名前空間	
COBOL	追加(A)
デバッグ	
リソース	ODBC V
設定	ADO NADO
	▲ OADO

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	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
Managed_ADO* \times	Program1cbl []-K]*
アプリケーション	構成(C): アクティブ (Debug) ▼ ブラットフォーム(M): アクティブ (Any CPU) ▼
SQL	
コピーブック パス	全般 ブラットフォームターゲットAny CPU ▼
名前空間	
COBOL	リース ジャーマット Artic Arti
デバッグ	▼ デバッグ用にコンパイル □ 検証可能コードの生成
リソース	□ 64-Bit ポインタの生成 高度

警告を含める(レベル W)

100

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。



エラーおよび警告

警告レベル

最大エラー数: □ 警告をエラーとして処理

追加指令

設定

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。

🔜 c:¥work¥cache¥managed	_ado¥bin¥debug¥Managed_ADO.exe		
DeLillo,Brenda S.	,+0000039903,Senior Technician▲ ,Washington		
Macrakis,Usha J. Engineer	,+0000021820,Executive Support ,Pueblo		
Vivaldi,Rob L. ian	,+0000098328,Executive Technic ,Newton		
	🖘 Managed_ADO (デバッグ中) – Microsoft Visual Studio (管理者)		
	ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) データ(A)	ツール(T) テ	・スト(S) ウィンドウ(
		<u>*</u>	
	[! ▣ % ‱ ☆ @ ≇ ≇ ☱ ≌ ⊐ ♀ ♀ ♀ ♀ ♀ ₽ , [! »		
	Program1.cbl []-K] ×		
	🏤 Program 1	🚽 🛋 MFI	ECM-END-PDIV()
	A.Name A.Salary A.Title A.Home_City FROM Sample.Employee A END-EXEC EXEC SQL OPEN CSR6 END-EXEC PERFORM UNTIL SOLSTATE >= "02000" EXEC SQL FETCH CSR6 INTO EXEC SQL FETCH CSR6 INTO :Employee-Name :Employee-Salary:Employee-Salary-NULL :Employee-Title:Employee-Title-NULL :Employee-Home-City:Employee-Home-City-NULL		

以上