# Micro Focus Visual COBOL for Visual Studio 2013







### はじめに

Micro Focus Visual COBOL for Visual Studio 2013 は、Microsoft の最新 Windows 開発環境 である Visual Studio 2013 の強力な統合開発環境(IDE)上で COBOL アプリケーションプログラ ム開発を可能とする COBOL 開発環境製品です。 COBOL プログラマが既存の COBOL 資産を Windows 環境で活用するだけでなく、COBOL プログラミング経験のない C#などのプログラマ が初めて COBOL アプリケーション開発を行う場合にも最適な製品です。

本書は、Micro Focus Visual COBOL for Visual Studio 2013 を学ぶための自習書です。本書 の読者は、プログラミングの基礎知識をもち、かつ Windows の基本操作を理解しているものとし ます。 なお、本書に沿って製品を実際に操作しながら学習するためには、以下の製品が必要で す。

Micro Focus Visual COBOL 2.3J for Visual Studio 2013

また、本書に掲載している画面イメージは Windows Server 2012 R2 でキャプチャしています。他の Windows OS では多少異なる場合がありますが、ご了承ください。

Visual COBOL は Microsoft が提供する Visual Studio のバージョン固有の機能に関連する ものを除いて各 Visual Studio 版で共通機能を提供しています。そのため、本書で紹介する内容 は Visual Studio 2012 版並びに Visual Studio 2015 版でも同様にお試しいただくことができ ます。



## 第1章 自習環境の準備

Micro Focus Visual COBOL for Visual Studio 2013 は、COBOLプログラミングの IDE として Microsoft Visual Studio 2013 の IDE を利用します。 自習環境用に、Microsoft Visual Studio 2013 (Professional / Premium / Ultimate Edition のいずれか)をセットアップ済みの PC か、以下の日本 マイクロソフト社ダウンロードセンターから Microsoft Visual Studio 2013 Shell (Integrated) 再頒 布可能パッケージを事前にダウンロードしてセットアップした PC を準備してください。

https://www.microsoft.com/ja-JP/download/details.aspx?id=40777

(リンク検証・・・2015/09/01)

いずれの Visual Studio 2013 ソフトウェアもセットアップされていない状態でインストーラを起動すると英語版の Microsoft Visual Studio 2013 Shell (Integrated) がインストールされます。

1 ダウンロードした vcvs2013\_23.exe をダブルクリックします。

## 2 表示されるセットアップ画面で エンドユーザ使用許諾契約書 をクリックします。

Mia Mia	ro Focus Visual COBOL for Visual Studio 2013 セットアップ 🛛 🗕 🗖 🗙
	Micro Focus Visual COBOL for Visual Studio 2013
	パージョン 2.3.00209
	製品インストール場所: C:\Program Files (x86)\Micro Focus\Visual COBOL\
	□ 同意する(A) エンドユーザー使用許諾契約書 インストール(D)



3 使用許諾契約書の内容を確認します。

- -X (二) (三) (三) C:¥Users¥Administrator¥Ap ター さ (二) MICRO FOCUS × 重要:ライセンサーは、以下の条項に拘束されるとの購入者の同意を条件として、本ソフトウェアを購入者に提供します。本契約の条項にセットアップ画面の承諾ボタンを クリックする方法によって同意することができない 場合には、購入者は本ソフトウェアを使用することができません。セットアップ画面の承諾ボタンクリックすることにより、 本契約を読み、理解し、かつその条項により拘束されることに合意したことを認めたことになります。 ~ エンドユーザ使用許諾契約書 ライセンサーは、ライセンシーに対し、本契約に定める条項に従ってライセンス・ソフトウェア(以下に定義します。)を使用するライセンスを許諾することを希望しており、ラ イセンシーは、当該使用許諾を受けることを希望しています。従って、ライセンサーとライセンシーは、以下のとおり合意します。 エンドユーザ使用許諾契約書(以下「本契約」といいます。)において、下記の用語は、下記の意味を有するものとします。 「ブレード」とは、それぞれCPU、ハード・ディスク及びメモリを内蔵した、1つのシャーシに格納される複数のサーバー・モジュールを意味します。 「コア」とは、コンピュータ・チップの中のブロセシング・ユニット 1つを意味し、論理演算ユニット、レジスタ、instruction fetcher、及びdata fetcher/storer から構成さ れます。「CPU」(Central Processor Unit)とは、データ処理ユニットを意味し、通常は一つの基盤に乗せられたマイクロチップを意味し、複数のコアから構成される場 合もあります。コンピュータ「ボックス」は、複数のCPUを内蔵することができます。「ボックス」は「サー バー」や「マシン」(又は「メインフレーム」)と呼ばれる場合がありま す。 「ライセンサー」とは、ライセンシーがライ† フォーカスが指名した法人を意味します。 イセンス・ソフトウェアを取得する国において、当該ライセンス・ソフトウェアのライセンスを付与する権限を与えられている、マイクロ 「ライセンス・ソフトウェア」とは、オブジェクトコード形式のライセンサー・コンピュータ・プログラム、そのユーザマニュアル(以下「マニュアル」といいます。)及びライセン サーがライセンシーに提供するその他の補足 資料を意味し、ソフトウェアのセキュリティキー、変更若しくは追加等を含みますが、これらに限定されません。ライセンス・ソ フトウェアには、ライセンシーが本契約第6条及び第7条に基づき受ける。若しくはライセンシーが別途契約し、料金を支払った保守サービスに基づき受けるライセンス・ ソフトウェアの新リリース、アップデート、リビジョン、変更、追加、訂正及び補修等も含まれるものとします。本契約の開始後、「ライセンス・ソフトウェア」には、ライセン シーがライセンスを受けている、旧バージョンのライセンス・ソフトウェアも含まれるものとし、本契約は当該が、「ジョンについても適用されるものとします。ライセンサーが 本契約第6条に基づき提供する場合を解決、本契判は、新リリース、アップテート、リビジョン、変更、追加、訂正及び補修に対する権利をらっそレンサージで はありません。ライセンス・ソフトウェア商品はWindows、Linux、Unix又はその他のブラットフォームで使用可能であり、各ブラットフォームについて、個別のライセン ス・ソフトウェア商品のライセンスが必要となります。 「製品注文書」とは、(i)ライセンシーが購入した各ソフトウェア・ライセンスを記載し、ライセンシーが作成した書類で(ii)ライセンサー又はその正規販売店が承諾した書類 を意味します。ライセンサーは、ライセンサー が承諾したことを書面により確認したとき、又はライセンス・ソフトウェアをライセンシーに交付したときのいずれか早い方に より、製品注文書を承諾します。各製品注文書は個別の契約を構成し、当該契約には本契約の条項を含むものとします。本契約の条項とライセンサーが書面により通 式に按権した署名者により明示的に承諾された製品注文書の条項との間に齟齬がある場合には、製品注文書の条項に従うものとします。本契約若しくは製品注文書に 関連してライセンシーが発行する購入注文書又はこれに類する書類に含まれるライセンシーの条項は通用されないものとし、発行された当該書類は、注文されたライセ ンス・ソフトウェア、ライセンスの数及び支払われる金額を確認するための事務的な理由によってのみ発行されるものであり、法的な効力はないものとします。 「ディザスタリカパリ・システム」とは、(i)ディザスタ(災害)が発生し、元のプロダクション・システムが使用できない場合、(ii)ディザスタリカパリ・テスティング及び/又は(ii)システム診断又はディザスタリカパリ・システム自体の保守サービスのために使用されるシステムを意味します。ディザスタリカパリ・システムには、ライセンス・ シートウーの思想を特定が見ていた。シート・オントロ・ステントを開いた。としたマステムを感味します。ディザスタリカパリ・システムには、ライセンス・

4 インストールを開始します。

問題がなければ、**同意します(A)** にチェックを入れ [インストール(I)] ボタンを押下してイン ストールを開始します。

Mic Mic	ro Focus Visual COBOL for Visual Studio 2013 セットアップ	- 🗆 X
	Micro Focus Visual COBOL for Visual Studio 2013	
	バージョン 2.3.00209	
	製品インストール場所:	
	C:\Program Files (x86)\Micro Focus\Visual COBOL\	
	●同意する(A) エンドユーザー使用許諾契約書	インストール()



5	セッ	トア	ップ	を終	了	し	ま	す	0
---	----	----	----	----	---	---	---	---	---

[閉じる(C)] ボタンを押下します。



以上で、自習環境の準備は終了しました。 Windows のスタートメニューに Visual COBOL for Visual Studio 2013 が登録されていることを確認してください。





## 第2章 Visual Studio 2013 IDE に慣れよう

Microsoft Visual Studio 2013 の IDE を初めて利用する COBOL プログラマのために、概要を簡単 に説明します。 既に Microsoft Visual Studio 2013 に習熟されている方は、本章を読み飛ばしてくだ さい。

Microsoft Visual Studio 2013の IDE は、メニューバー、ツールバー、左、下または右にドッキン

グまたは自動的に非表示になる 各種ツールウィンドウ、エディ ター領域など、複数の要素で構 成されます。 IDE 内の要素の 配置は、適用した設定とその後 に加えたカスタマイズ内容によ って異なります。



Visual Studio 2013 のソリューションとプロジェクトには、アプリケーションの作成に必要な参照、データ接続、フォルダー、およびファイルを表す項目が含まれています。 ソリューションには複

数のプロジェクトを含めることがで き、プロジェクトには、通常、複数の 項目が含まれます。 ソリューションエ クスプローラには、ソリューション、 それらのプロジェクト、そのプロジェ クト内の項目が表示されます。 ソリュ ーション エクスプローラーを使用する と、編集するファイルを開く、プロジ ェクトに新規ファイルを追加する、ソ リューション、プロジェクト、および 項目のプロパティを表示するなどの操 作を実行できます。





Visual Studio 2013 のソースコードエディターには、COBOL 予約語とデータ名や手続き名などの 利用者語を色分け表示したり、COBOL スニペットなど COBOL 言語固有の機能拡張が含まれます。ソ

スコードを入力する
 とバックグラウンドチ
 エックを実行して、赤
 の波線でエラー箇所を
 強調表示します。その
 エラー箇所にマウスポ
 インタを移動すればエ
 ラー内容を確認した
 り、定義への移動、他
 の参照検索などの操作
 が可能です。

🔩 Micro	pFocus. demo. UC	T0027P	→  E File Section		
	1 · · · 1	0 · · · · 20 · · · · 30	40 50 60	70 • • • 80	g
▶ 40	004000	COPY UDCUU99	I.		<del>+</del>
41	004100 01	LK-SHISAN-AMOUNT	PIC 9(08).	4	
42	0042007			*	or base
44		TROOLDORE	DIVISION	*	<b>C</b> 1
45	004500*			*	11
48	±004600 PRC	CEDURE	DIVISION		<b>a</b> •
47	004700	USING	LK-KIMEI-HOKENSHA-INFO		
48	004800		LK-SHISAN-AMOUNT.		<b>1</b>
- 49	004900				<b>P</b> .
50	005000*			*	- (16)
51	005100*	MODULE-NO.		*	<u></u>
52	005200*	MODULE-ID.	MAIN-SYOKI-KIN Filo (BRA tukter Statement - Cul 2003) In The	*	
53	005300*	MODULE-NAME.	記名保険者情報官理ファイル登録処理	*	
34	005500 WAI	N SYODI DIN	SECTION	*	158
50	⊟003300 MAI ⊡005800 WAI	N_SVORT_000	SECTION.		
57	⊨005700 <b>*</b>	N 010K1 000.			E How
58	1005800*		<< ファイルのオープン >>		188 ·
59	005900	OPEN EXTEND HOKENSHA	-FILE.		
60	=006000∗				100
61	006100*		<< レコードのセット >>		-
62	006200	MOVE LK-KIMEI-HOKENS	HA-INFO TO KIMEI-HOKENSHA-INFO.		CONC.
63	006300	MOVE LK-SHISAN-AMOUN	T TO SHISAN-AMOUNT.		
64	<u>⊨</u> 006400 <b></b> *				
65	006500*		<< レコードの書き込み >>		
66	006600	WRITE HOKENSHA-REC.			
6	⊡UU6/UU <b>≭</b>				
50	006800*		<< ファイルのクロース >>		
70	000000	OLOGE HUKENONA-FILE.			
71		N-SVORI-999.			
	1007200 MAI	GOBACK.			
12	1001200				

Visual Studio 2013 のビルド構成では、プラットフォームの選択、プロジェクトまたはソリューションのビルド方法を定義します。プロジェクトタイプごとに、デバッグとリリースのデフォルト構成があり、独自の構成を作成することも可能です。コンソールウィンドウにはビルド時のメッセージやアプリケーションのコンソール出力等が表示されます。問題ウィンドウには、不正な構文、キーワードのスペルミス、型の不一致などのコンパイルエラーが表示されます。





ビルドしたアプリケーションは、実行時の論理エラーやセマンティックエラーなどの問題を検出し

て修正するために、デバ ッガーを使用します。 Visual Studio 2013 の デバッガーは、コードを ステップ実行したり様々 な条件を設定したブレー クポイントで実行を中断 させ、変数ウィンドウや ウォッチ式などのツール を使用してローカル変数 やその他の関連データを 調べることができます。

✓       dotNETusingCOBOLBizLogic (デバッグ中) - Microsoft Visual Studio (管理者)         ファイル(F) 編集(E) 表示(V) カッシュクト(P) ビルド(B) デパッグ(D) チーム(M) ツール(T) テスト(S) 分析(N) ウィンI         ● • ●   ② • ③   ■ ■   ⑦ • ③ •   ▶ 続行(C) • ④ • Debug •   Any CPU •   □   □   □ = ●         プロセス [4848] WPF_Presentation_Layε • 図 ライフサイクル イベント • スレッド [4772] メインスレッド	ドウ(W) へ)   10   → 9 スタックフレーム
COBLIB.UDT0026DB	
10 % ● 目動変数 ● 年 × 呼び出し履歴 名前 値 型 ● COBClassLibraries.dllISHIYOU-MOKUTEKI-RTM ● COBClassLibraries.dllISHIYOU-MOKUTEKI-RTM COBCLassLibraries.dllISHIYOU-MOKUTEKI-RTM COBCLassLibraries.dllISHIYOU-MOKUTEKI-RTM COBCLassLibraries.dllISHIYOU-MOKUTEKI-RTM COBCLassLibraries.dllISHIYOU-MOKUTEKI-RTM COBCLassLibraries.dllISHIYOU-MOKUTEKI-RTM COBCLassLibraries.dllISHIYOU-MOKUTEKI-RTM COBCLassLibraries.dllISHIYOU-MOKUTEKI-RTM COBCLassLibraries.dllISHIYOU-MOKUTEKI-RTM COBCLassLibraries.dllISHIYOU-MOKUTEKI-RTM COBCLassLibraries.dllISHIYOU-MOKUTEKI-RTM COBCLassLibraries.dllISHIYOU-MOKUTEKI-RTM COBCLASSLIBARIES.dllISHIYOU-MOKUTEKI-RTM COBCLASSLIBARIES.dllISHIYOU-MOKUTEKI-RTM COBCLASSLIBARIES.dllISHIYOU-MOKUTEKI-RTM COBCLASSLIBARIES.dllISHIYOU-MOKUTEKI-RTM COBCLASSLIBARIES.dllISHIYOU-MOKUTEKI-RTM COBCLASSLIBARIES.dllISHIYOU-MOKUTEKI-RTM COBCLASSLIBARIES.dllISHIYOU-MOKU	

デバッグが完了したアプリ ケーションは、Windows イン ストーラーを使用するか、ファ イルを手動でコピーして、本番 環境に配置します。

Visual Studio 2013 では、 無償の配備ソリューション InstallSheild Limited Edition をご利用できます。

	新しいプロジェクト	? ×
▶ 最近使用したファイル	.NET Framework 4.5.1 + 並べ替え基準: 既定	<ul> <li>・ iii E インストール済 P ・</li> </ul>
<ul> <li>インストール済み</li> <li>▶ Visual F# SQL Server</li> <li>▶ JavaScript Python</li> <li>Other Project Types</li> <li>セットアップに設置</li> </ul>	InstallShield Limited E… セットアップと配置	種類: セットアップと配置 InstallShield Limited Edition のサポートを 有効にする指示を提供します
visual Sculid 991 サンプル ▲ ▶ オンライン	→	
名前( <u>N</u> ): Setup1 場所( <u>L</u> ): C:¥wor ソリューション( <u>S</u> ): 新しいツ ソリューション名( <u>M</u> ): Setup1	k¥test¥AAA - [ リューションを作成する -	参照(E) ✓ ソリューションのディレクトリを作成(D)
	i	ソース管理に追加(U) OK キャンセル

なお、本番環境には COBOL Server が事前にインストールされている必要があります。



## 第3章 はじめての Visual COBOL

それでは、Windows のコマンドプロンプト画面に「Hello World」を表示する COBOL アプリケー ションを Visual COBOL for Visual Studio 2013 で作成します。

1 Visual COBOL for Visual Studio 2013 を起動します。

Windows のスタートメニューから、Visual COBOL for Visual Studio 2013 をクリックしま



2 使用するテンプレートを選択します。

インストールされたテンプレートの一覧から COBOL プロジェクト、Native、コンソールアプリ



**ケーション**を選択しま す。 **ソリューション** のディレクトリを作成 (D) がチェックされ ていることを確認し、 名前(N)に ConsoleHello と入 力し、[OK] ボタン を押下します。



3 コードエディターで COBOL ソースコードを入力します。

プロジェクト「ConsoleHello」の作成が成功すると、COBOL 専用のコードエディターが起動しま す。エディター画面には、コンソールアプリケーションのひな形が表示されています。 COBOL ソー スは、見出し部(identification division)、環境部(environment division)、データ部(data division)、手続き部(procedure division)で構成されますが、今回は「Hello World」を表示して終了 するプログラムなので、手続き部に DISPLAY 文を書き加えるだけです。

なお、COBOL 正書法ではエディター画面左右にあるグレー部分を特別な領域として利用するので、 通常のソースコードはこれを避けて入力します。

マイル(E)       編集(E)       表示(Y)       プロジェカト(E)       ビルド(E)       デバッグ(D)         マーロ       昭       雪       当       マーマ       ▶       開始・       ●       Debug	チーム(M) ツール(I) テスト( <u>S)</u> - x86 ・ ↓	マ ロ ク 分析(N) ウィンドウ - 1 日 唱 1 目 1 目	イック起動 (Ctrl+Q) (W) ヘルプ(出)   ■ 11 11 11	ראש <b>ב 9</b> א אראש ₪
Program1.cbl >> X Program1.cbl	edure Division		ソリューション エクスプロー           ●         ●           ●         ●           ソリューション エクスプロー           ■         ソリューション エクスプロー           ■         ソリューション エクスプロー           ■         ソリューション エクスプロー           ■         ●           ■         Propertie           ●         ●           ●         ●           ●         ●           ●         ●           ●         ●           ●         ●           ●         ●           ●         ●           ●         ●           ●         ●           ●         ●           ●         ●           ●         ●           ●         ●           ●         ●           ●         ●	5- ・ ↓ × ● 回 面 ♪ - ▼ 5- の検索(Ctrl+:) ♪ ・ vrials'(1 力ジェクト) lo s 1. cbl -ム エクスカー・・ クラス ピュー ・ ↓ × エクトプロパティ ・
2 100 % - 4		►	プロジェクト パス	C:¥work¥Tutorials¥VS20
		<b>-</b> ↓ ×	名前	ConsoleHello.cblproj
このソール ウィンドウにメンバーの呼び出し階層を表示するには、コードエディ ターでメンバー名を右クリックし、「呼び出し階層の表示」をクリックします。 呼び出し階層 エラー一覧 データ ツール操作 出力 シンボルの検索結果	呼び出しサイト	場所 🔺	その他	
準備完了		11 行	34列 34戈	マ字 挿入 🧋

DISPLAY "Hello World".



4 COBOL アプリケーションをビルドします。

終止符(ピリオド)を含めてスペルミス がなければ、ソリューション構成が Debug、ソリューションプラットフォー ムが x86 であることを確認して、ビルド (B)メニューから ソリューションのビル ド(B) を選択します。 出力ウィンドウに ビルド結果が表示されるので、すべてのビ ルドが正常終了したことを確認します。



5 COBOL アプリケーションをデバッグ実行します。

デバッグ(D)メニューから ステップイン(I) を選択すると、コマンドプロンプト画面が開き、デバ

ッガーがステップ実行を開始 します。 デバッガーは手続き 部の最初の COBOL 文である display 文を実行する前の状態 で停止します。今回は調べる ローカル変数がないので、そ のまま **ステップイン(I)** を選 択し、ステップ実行を進めま す。



コマンドプロンプト画面に「Hello World」が表示されたことを確認して、デバッグを終了します。



#### 第4章 Visual COBOLの画面操作

続いて、ウィンドウ画面のボタンを押して「Hello World」を表示する COBOL アプリケーションを Visual COBOL for Visual Studio 2013 で作成します。

#### Visual COBOL for Visual Studio 2013 を起動します。 1

Windows のスタートメニューから、Visual COBOL for Visual Studio 2013 をクリックしま



#### 使用するテンプレートを選択します。 2

**ームアプリケーション**を

ョンのディレクトリを作

成(D) がチェックされて

いることを確認して、名

前(N)に WinHello と入

下します。



インストールされたテンプレートの一覧から COBOL プロジェクト、Managed、Windows フォ

- 12 -



3 フォームデザイナーでウィンドウを作成します。

プロジェクト「WinHello」の作成が成功すると、フォームデザイナーが起動します。 デザイナー画面に Form1 ウィンドウが表示されるので、画面左に表示される ツールボックス を選 択して展開します。 表示されたツールボックスから Button コントロールを選択し、Form1 ウィン ドウ上にドラッグ&ドロップします。



Form1 ウィンドウ上にボタンが表示されると、プロパティが Button1 ボタンに切り替わります。 プロパティを下方向にスクロールして「表示」セクションの Text を選択します。





テキストの値を「Button1」から「Say Hello」に変更しま す。



ツールボックスをスクロールして Label コ ントロールを選択し、Form1 ウィンドウ上に ドラッグ&ドロップします。



プロパティをスクロールして「表示」セクションの **Text** を選択 し、テキストの値を削除します。





以上でウィンドウ画面の作成は終了です。

F¢	orm1.cbl (דידי	1) ⊅ X		設定後のフォーム
			Form1	イメージ
			Say Hello	
			<b>P</b>	

4 コードエディターで COBOL ソースコードを入力します。

次に、デザイナー画面上の Say Hello ボタンをダブルクリックすると、COBOL 専用のコードエディターが起動します。

エディター画面には、Windows フォームアプリケーションのひな形が表示されます。 ここでは Say Hello ボタンをクリックした時の処理を記述するので、button1\_Click メソッドの手続き部に以 下の move 文を追加します。

Form1.cbl 👳 🗙 Fo	orm1.cbl [デザイン]	-
🔩 WinHello. Form1	<ul> <li>O<sub>a</sub> button1_Click(sender AS object, e AS System.EventA</li> </ul>	Args) -
1	• 10 • • • • 20 • • • • 30 • • • • 40 • • • • 50 • • • • 60 • • • • 70 • • • • 80 • • • 90 • •	· · 100 · X
1 🗉	class-id WinHello.Form1 is partial	÷
2	inherits type System.Windows.Forms.Form.	etilitidi arealaria
4	working-storage section.	7
6 📮	method-id NEW.	ntik arfitter 4
	procedure division. invoke self::InitializeComponent goback.	
10 11	end method.	
12 🖻	method-id button1_Click final private. pro <del>podure division using by value sender as ob</del> ject e as type System.EventArgs.	
15	end method.	
17	end class.	
10		

move "Hello World!" to self::label1::Text.



5 COBOL アプリケーションをビルドします。

スペルミスがなければ、ビルド(B)メニューから WinHelloのビルド(U) を選択します。

×	🕽 🛛 WinForm - Mic	rosoft Visual Studi	○ (管	<sup>き</sup> 理者)					
ファ	イル(F) 編集(E) 表	見示(V) プロジェクト(P)	UI	*(B) デバッグ(D)	<i>∓−</i> Д(М)	୬−ル(T)	テスト(S)	分析(N)	ウィンドウ
	G • 🗇 📅 • 🖕 I	🛯 🚰 🦻 - 🤍 - 🗍	*	ソリューションのビルド(	в)	Ctrl+	Shift+B	ें 🔚 🖷	13 2
ļ	Form1 chi + X Fo	orm1.chl[국ザイン]		ソリューションのリビルド	(R)				
	winHello, Form1			ソリエーションのクリーン	(C)			ect. e AS S	System.
Ŕ	1	· 10 · · · · 20 · · ·		ソリューションでコードダ	う析を実行(Y)	Alt+	F11	· · 80 ·	9
K		class-id WinHello F	*	WinHello のビルド(U	Ŋ				
	3	milerrea	_	WinHello のりじしド	(E)				
Ϋ́	4	working-storage sec		WinHello のクリーン	(N)				
4	6 🖻	method-id NEW.		バッチ ビルド(T)					
	8	invoke self::Ir		構成マネージャー(0)					
LC.	9	goback.							
м	11	enu methou.							
P	12 白	<pre>method-id button1_C procedure division</pre>	lick	final private.	or ac obier	t e e ti	no Svetom	EventAre	
Ser	14	move "Hello Wor	Id!"	to self::label	1::Text.	,	pe oystem	·LVCIICHI &	••
ver	15	end method.							
E	17	end class.							
Ĕ	18								

出力ウィンドウにビルド結果が表示されますので、ビルドが正常終了したことを確認します。





6 COBOL アプリケーションを実行します。

デバッグ(D)メニューから デバッグなしで開始(H) を選択すると、Form1 ウィンドウが開きます。

×	WinForm - Mic	crosoft Visual Studio (管理者	i)							7 <b>P</b>	1 2
רד 🖓	イル(F) 編集(E) 昇	長示(V) プロジェクト(P) ビルド(B) 🤇	デノ	(ッグ(D) チーム(M)	ツ−ル(T)	テスト(S)	分析(N)	ウィンドウ <b>(</b> )	N)	<i>へ</i> ルプ(H	0
- š. (	G • 🖸 🕆 🔁	💾 🔐   ジ - 🤉 - 🕨 開始 - 🤇		ウィンドウ(W)				•		51 71 X	۽ 1
4	Form1.cbl ⇒ × F	orm1.cbl [デザイン]		グラフィックス(C)				•			-
×.	🔩 WinHello. Form1		►	デバッグ開始(S)			F5	_	entA	rqs)	Ŧ
Ŕ	1	• 10 • • • 20 • • • 30 • •	►	デバッグなしで開始(H)			Ctrl+F5		• •	· 100 ·	Х
Ľ		class-id WinHello.Forml is p inherits type Syst	e <sup>©</sup>	プロセスにアタッチ(P)						÷	
	3			その他のデバッグ ターゲ	'wh			•		and the second	
Ϊ	4	working-storage section.		例外(X)			Ctrl+Alt	:+ E			
<u>ب</u> ور ا	6 8	method-id NEW.		パフォーマンスと診断(F)	)		Alt+F2			*	
퉰	8 1	invoke self::Initialize0	ς.	ス <del>テ</del> ップ イン <b>(I)</b>			F11				
Ķ	9 10	goback. end method.	G,	ステップ オーバー(0)			F10			-	
<u> </u>		method-id button1 Click fina	1	Windows Phone 70	プリケーション分	析の開始	Alt+F1			_	
S د	13 🖻	procedure division using by		ブレーケポイントの設定/	/解除(G)		F9				
BVe	15	end method.		ブレークポイントの作成(	(B)			•			
Ę.	16	and allows	22	すべてのブレーケポイント	~の削除(D)		Ctrl+Sh	ift+F9			
J'SI	18	enu crass.		オプションと設定(G)							
14			۶	WinHello のプロパティ	·						
3											

Form1 ウィンドウの Say Hello ボタンをクリックして「Hello World!」の表示を確認します。

Form1	_ □	x
Say Hello		



## 第5章 Visual COBOL のファイル入出力

次に、エクセルやメモ帳で作成した CSV ファイルを読み込んで、固定長順編成ファイルを作成する COBOL アプリケーションを Visual COBOL for Visual Studio 2013 で作成しましょう。

1 Visual COBOL for Visual Studio 2013 を起動し、使用するテンプレートを選択し ます。

Windows のスタートメニューから、Visual COBOL for Visual Studio 2013 をクリックし、 新規プロジェクト を選択します。 インストールされたテンプレートの一覧から COBOLプロジェク ト、Native、コンソールアプリケーションを選択します。 ソリューションのディレクトリを作成(D) がチェックされていることを確認して、名前(N)に LoadCSVFile と入力し、 OK をクリックしま す。

		新しいプロジェクト		? ×
▶ 最近使用したファイル	.NET	Framework 4.5.1 - 並べ替え基準: 既定	- 📰 📰 - 1221	トール済み テンプレート の検索 (Ctrl- 🔎 -
▲ インストール済み	CBL	Enterprise Server アプリケーション	COBOL プロジェクト 💧 タイ:	ブ: COBOL プロジェクト
▲ テンプレート ▲ COBOL プロジェクト	CBL	Windows アプリケーション	ネー: COBOL カジェクト るため	ティブ コマンドライン アプリケーションを作成す めのプロジェクトです。
Cloud Database	C:\	コンソール アプリケーション	COBOL プロジェクト	
Managed		リンク ライブラリ	COBOL プロジェクト	
Web		空のプロジェクト	COBOL プロジェクト	
▶ Visual Basic ▶ Visual C#		Micro Focus INT/GNT	COBOL プロジェクト	
<ul> <li>► オンライン</li> </ul>		Web Service Client Application from WSDL/J	SON COBOL プロジェクト	
		オンラインでテンプレートを検索するには、ここをク	リックします。	
名前(N): LoadCSVFile	)			
場所(L): C:¥work¥Tuto	prials¥VS	2013	▼ 参照(	( <u>B</u> )
ソリューション名( <u>M</u> ): NativeConsole	2		בעע	ーションのディレクトリを作成( <u>D</u> )
			□ y-z	<管理に追加(U)
				<u>ОК</u> +руел



2 コードエディターで COBOL ソースコードを入力します。

プロジェクト「LoadCSVFile」の作成が成功すると、COBOL 専用のコードエディターが起動しま す。エディター画面にコンソールアプリケーションのひな形が表示されるので、環境部(environment division)、データ部(data division)、手続き部(procedure division)を書き換えます。

まず、環境部の構成節(configuration section)を削除し、以下の入出力節(input-output section) を追加します。 まだ、データ部のファイル定義が未入力なので in-file と out-file がエラーとなります が、ここでは無視して構いません。

INPUT-OUTPUT SECTION. FILE-CONTROL. SELECT IN-FILE ASSIGN TO "Emp\_Master.csv" LINE SEQUENTIAL. SELECT OUT-FILE ASSIGN TO "Emp\_Master.dat".





次に、データ部の作業場所節(working-storage section)を削除し、以下のファイル節(file section)を追加します。 なお、データ部のファイル定義を入力したので、環境部のエラーは無くなります。

FIL	e se	CTION.		
FD	IN-	FILE.		
01	IN-	REC	PIC	X(50).
FD	OUT	-FILE.		
01	0UT	-REC.		
	05	OUT-EMPNO	PIC	9(8).
	05	FILLER	PIC	Х.
	05	OUT-JNAME1	PIC	N(5).
	05	OUT-JNAME2	PIC	N(5).
	05	OUT-NAME1	PIC	Х(5).
	05	OUT-NAME2	PIC	Х(5).
	05	OUT-GENDER	PIC	Х.
	05	FILLER	PIC	Х.
	05	OUT-DIV	PIC	N(5).
	05	OUT-EMPDATE	PIC	9(8).
	05	FILLER	PIC	Х.





最後に、手続き部の goback 文を削除し、以下の 手続き文を追加します。

PROC1. OPEN INPUT IN-FILE. OPEN OUTPUT OUT-FILE. PROC2. READ IN-FILE AT END GO TO PROC9. INITIALIZE OUT-REC. UNSTRING IN-REC DELIMITED BY "," INTO OUT-EMPNO OUT-JNAME1 OUT-JNAME2 OUT-NAME1 OUT-NAME2 OUT-GENDER OUT-DIV OUT-EMPDATE END-UNSTRING. WRITE OUT-REC. GO TO PROC2.

PROC9.

CLOSE IN-FILE OUT-FILE. STOP RUN.





3 COBOL アプリケーションをビルドします。

終止符(ピリオド)を含めてスペルミスがなければ、ソリューション構成が Debug、ソリューション

プラットフォームが x86 である ことを確認して、ビルド(B)メニ ューから ソリューションのビルド (B)を選択します。 出力ウィン ドウにビルド結果が表示されるの で、すべてのビルドが正常終了し たことを確認します。

<b>א</b> יידר	NativeConsole - Microsoft バル(F) 編集(E) 表示(V) プロジ	: Visual Stud エクト(P) ビル	dio (管理者) F(B) デバッグ(D) チーム(M)	) ツール(T) テスト(S)	マロク 分析(N) ウ
	9 - 0   🛅 - 🏜 💾 🤊 -	C - 1 🕷	ソリューションのビルド(B)	Ctrl+Shift+B	) 🖕 👘 🗉
<del>9</del> -/(-	Program1.cbl 😐 🗙		ソリューションのリビルド(R) ソリューションのクリーン(C)		
エクスプローラー	I····Ie···           24         05 OUT           25         05 OUT           26         05 FIL           27         procedure d           27         procedure d	-DIV -EMPDATE LER livision.	ソリューションでコード分析を実行( LoadCSVFile のビルド(U) LoadCSVFile のビルド(E) LoadCSVFile のクリーン(N)	Y) Alt+F11	• • 80 • X
ツールボックフ	28 E PROCI. 29 OPEN I 30 OPEN 0 31 PROC2.		バッチ ビルド(T) 構成マネージャー(O)		
SQL Se	33 READ I 34 INITIAL 35 UNSTRIN 36 INTO	IZE OUT-REC IG IN-REC D OUT-EMPNO	D. DELIMITED BY ","		

4 CSV ファイルを作成します。

デバッグフォルダ(<第5章1で指定したフォルダ> ¥LoadCVSFile¥LoadCVSFile¥bin¥x86¥debug)にメモ帳などを利用して以下の Emp\_Master.csv フ ァイルを作成します。

11111113,佐藤,隆,サトウ,タカシ,M,営業部,19980401,0 22222226,鈴木,尚之,スズキ,ナオユキ,M,技術部,19981015,0 3333339,田中,直美,タナカ,ナオミ,F,総務部,19990401,0 44444442,山田,洋一,ヤマダ,ヨウイチ,M,営業部,20000701,0 55555555,伊藤,弘子,イトウ,ヒロコ,F,技術部,20010401,0 666666668,木村,貴弘,キムラ,タカヒロ,M,営業部,20021220,0 77777771,中村,慎司,ナカムラ,シンジ,M,技術部,20030401,0 88888884,橋本,悦子,ハシモト,エソコ,F,総務部,20040805,0 99999997,三井,薫,ミツイ,カオル,F,営業部,20050401,0

3	Debug			_ □	x
🔄 💿 🔻 🕈 📙 « NativeCo	nsole $\blacktriangleright$ LoadCSVFile $\blacktriangleright$ bin $\blacktriangleright$ x86	▶ Debug v	<b>さ</b> Debugの検索		P
🛧 お気に入り	▲ 名前 ▲	更新日時	種類	サイズ	
📕 ダウンロード 🛛 🗕 🛶	Emp_Master.csv	2015/09/03 14:59	CSV ファイル	1 KB	
📰 デスクトップ	LoadCSVFile.exe	2015/09/03 14:59	アプリケーション	16 KB	
📃 最近表示した場所	Program1.idy	2015/09/03 14:58	IDY ファイル	42 KB	
■ PC つ エロン MITCL WIDI & C 3 個の項目	¥			[	



5 COBOL アプリケーションをデバッグ実行します。

100 自

**デバッグ(D)**メニューから ステップイ ン(I) を選択するか F11 キーを押す と、コマンドプロンプト画面が開き、デ バッガーがステップ実行を開始します。 デバッガーは手続き部の最初の COBOL 文である open 文で実行を中断します。



入力ファイルから読み込んだレ コードの内容を確認するため、 unstring 文の in-rec 上で右クリ ックして **ウォッチ式の追加(W)** を選択します。

27 E PROCE 28 PROCI- 29 OPEN INPUT 30 OPEN OUTPUT 31 - 32 E PROC2- 33 READ IN-FILE 34 INITIALIZE 0	n. IN-FILE. OUT-FILE. AT END GO TO PROCS. UT-REC.	
35 UNSTRING IN- 36 INTO OUT-E 37 OUT-J 38 OUT-J		Ctrl+K, Ctrl+X Ctrl+K, Ctrl+S
39 OUT-N 40 OUT-N 41 OUT-G 42 OUT-D	■ 定義へ移動(G) すべての参照の検索(A) ∞ 呼び出し増留の表示(H)	F12 Shift+F12 Ctrl+K, Ctrl+T
43         001-E           44         END-UNSTRING.           45         WRITE           46         GO TO PROC2	D クイックブラウズ Micro Focus Code Analysis	Alt+Q
47 48	ブレーケポイント(B) COBOL ウォッチポイントを追加 ウォッチ式の追加(W)	•
加変数	↔ 並列ウォッチの追加(P)	

同様に出力ファイルに書き出す レコードの内容を確認するため、 initialize 文の out-rec 上で右ク リックして ウォッチ式の追加 (W) を選択します。





F11 キーを 3 回押すと、デバッガーは read 文実行後、処理を中断します。

ウォッチ式の in-rec の値 には CSV ファイルから読み 込んだ最初のレコードが表示 されます。

Program1.cbl 👳	×
🔩 PROGRAM1	- 🗇 PROC2() -
1	· · 10 · · · · 20 · · · · 30 · · · · 40 · · · · 50 · · · · 60 · · · · 70 · · · · 80 · · · · 90
30	OPEN OUTPUT OUT-FILE.
31	PD002
33	READ IN-FILE AT END GO TO PROCO
⇒ 34	INITIALIZE OUT-REC.
35	UNSTRING IN-REC DELIMITED BY ","
37	OUT-JNAME1
38	OUT-JNAME2
39	
40	OUT-GENDER
42	OUT-DIV
100 % 👻 🖣	
<b>ウ</b> ォッチ 1	▼ 및 × 呼び出し履歴 ▼ 및 ×
名前	値
IN-REC	11111113,佐藤,隆,카ウ,勿汐,M,営業部,19980401,0 Q マ PIC X(50 🛛 Program1.PROC2 COB
Þ 🥥 OUT-REC	{長さ=60}: " " Q, - GROUP
	· · · · · · · · · · · · · · · · · · ·
自動変数 ローカル	<u>ሳክሃቶ1</u> 🔲 🖬 🖬 🖬 🖬 🕅

さらに F11 キーを 2回押すと、デバッガーは unstring 文を実行後、処理を中断します。

ウォッチ式の out-rec の値には 出力ファイルへ書き出す最初のレコ ードが表示されます。

Program1.cbl ⇔ ×			-			
🔩 PROGRAM1	- 🗇 PROC2()		-			
1 · · · 10 · · · 20 ·	· · · 30 · · · · 40 · · · · 50 ·	• • • • 60 • • • • 70 •	· · · 80 · · · 90 X			
39 40 41 42 42 44 44 44 45 WRITE 001-F 46 G0 T0 PROC	1         1					
47						
100 %						
ウ <del>オ</del> ッチ 1		▼ 早 × _ 呼び	出し履歴 ▼ H ×			
名前	値	型 🔺 💈	名前 言語 🗠			
IN-REC	11111113 佐藤 隆 サトウ タカシ M 営業部	1 9 PIC X(50 🛛 🖒 P	Program1.PROC2 COB			
▲ OUT-REC	{長さ=60}: "1111113 佐藤 隆	Q - GROUP				
OUT-EMPNO OF OUT-REC	11111113	PIC 9(8)				
FILLER OF OUT-REC		Q PIC X				
OUT-JNAME1 OF OUT-REC	佐藤	9 PIC N(5)				
OUT-JNAME2 OF OUT-REC	隆	9 PIC N(5)				
OUT-NAME1 OF OUT-REC	<u> </u>	Q PIC X(5)				
OUT-NAME2 OF OUT-REC	タカシ	9 PIC X(5)				
OUT-GENDER OF OUT-REC	M	Q PIC X				
FILLER OF OUT-REC		Q PIC X				
OUT-DIV OF OUT-REC	営業部	Q PIC N(5)				
OUT-EMPDATE OF OUT-REC	19980401	PIC 9(8)				
FILLER OF OUT-REC		Q - PICX	-			
自動変数 ローカル ウオッチ 1		(E	<b>J D V</b>			

さらに F11 キーを 4 回押すと、デバッガーは initialize 文を実行後、処理を中断します。

ウォッチ式の in-rec の値には CSV ファイルから読み込んだ 2 番目のレコードが表示され、out-rec

の値は initialize 文で初期 化されています。

Program	n1.cbl + ×		-
🔩 PROG	GRAM1	- 🛛 PROC2()	•
	1	. 10 20 30 40 50 60	70 · · · 80 · · · 90 X
32	E F	PROC2.	÷
33		READ IN-FILE AT END GO TO PROC9.	
34		INITIALIZE OUT-REC.	51
35		UNSTRING IN-REC DELIMITED BY ","	<b>6</b> .8
36		INTO OUT-EMPNO	
37		UUT-JNAMET	
38		OUT-JNAME2	• • • • • • • • • • • • • • • • • • •
100 %	• 4		► •
ウォッチ 1		<del>▼</del> ╄ ×	呼び出し履歴 ▼ 早 ×
名前		値 型 ^	名前言語~
91	N-REC	22222226,鈴木,尚之,スズキ,ナオ2キ,M,技衚部,19981015,0 Q 💡 PIC X(50	Program1.PROC2 COB
	OUT-REC	{長さ=60}: "00000000 00 Q マ GROUP	
		▼	<b>V</b>
自動変数	て ローカル <mark>ウ</mark> ァ	hýť 1	



**デバッグ(D)**メニューから 続行(C) を選択するか CSV ファイルからすべてのレコードを読み込む まで F11 キーを押すと、デバッガーは終了します。



デバッグフォルダ(<第5章1で指定したフォルダ>

¥LoadCVSFile¥LoadCVSFile¥bin¥x86¥debug)に **Emp\_Master.dat** ファイルが作成されます。テキ ストエディタなどでファイルを開き、社員9名分のデータが表示されることを確認します。下図は、 Tera Pad を使って 60 桁で折り返し表示した例です。

3	Debug			_ □	x
🔄 🍥 🔻 🕇 퉬 « NativeCo	onsole + LoadCSVFile + bin + x86	▶ Debug v	C Debugの検	*	<i>م</i>
☆ お気に入り	名前	更新日時	種類	サイズ	
📕 ダウンロード	Emp_Master.csv	2015/09/03 14:59	CSV ファイル	1 KB	
📰 デスクトップ ――	> Emp_Master.dat	2015/09/03 15:17	DAT ファイル	1 KB	
週 最近表示Lた場所	LoadCSVFile.exe	2015/09/03 14:59	アプリケーション	16 KB	
	Program1.idy	2015/09/03 14:58	IDY ファイル	42 KB	
p PC					
(回) 100 MITCH WIDT 400 4個の項目					

😨 Emp_Master.dat - TeraPad 🗕 🗖	x
ファイル(E) 編集(E) 検索(S) 表示(V) ウィンドウ(W) ツール(T) ヘルプ(H)	
🖸 🛱 💾 🚄 X 🐚 💼 🗠 🗠 🔎 🥬 😡	
Image: Constraint of the second se	
< >	
1行: 1桁 標準 [60] SJIS CRLF 挿入	H



## 第6章 Visual COBOL のバッチアプリケーション

本章では、第5章で作成した固定長順編成ファイルを読み込んでレポートファイルを作成するバッ チアプリケーションを Visual COBOL for Visual Studio 2013 で作成します。

1 Visual COBOL for Visual Studio 2013 を起動し、使用するテンプレートを選択し ます。

Windows のスタートメニューから、Visual COBOL for Visual Studio 2013 をクリック し、新規プロジェクト を選択します。 インストールされたテンプレートの一覧から COBOLプロジ ェクト、Native、コンソールアプリケーションを選択します。 ソリューションのディレクトリを作成 (D) がチェックされていることを確認して、名前(N)に BATCHRPT と入力し、OK をクリックしま す。

	新しいプロジェクト	? X
▶ 最近使用したファイル	.NET Framework 4.5.1 → 並べ替え基準: 既定	▼ # E インストール済み テンプレート の検索 (Ctrl・ タ -
▲ インストール済み	Enterprise Server アプリケーション	COBOL プロジェクト
▲ テンノレート ▲ COBOL プロジェクト	Windows アプリケーション	でのBOL 力ジェクト でのBOL 力ジェクト
Cloud Database	CBL コンソール・アプリケーション	COBOL JUSIDH
Managed	して リンク ライブラリ	COBOL プロジェクト
Web	空のプロジェクト	COBOL 力ISI力ト
▶ Visual Basic ▶ Visual C#	Micro Focus INT/GNT	COBOL プロジェクト -
▼	Web Service Client Application from WSDL/J	JSON COBOL プロジェクト
	オンラインでテンプレートを検索するには、ここをク	クリックします。
名前(N): BAT CHRPT		
場所(L): C:¥work¥Tutori	als¥VS2013	▼ 参照( <u>B</u> )
ソリューション(S): 新しいソリューション	を作成する	
ソリューション名(M): BatchDemo		
		<ul> <li>ソース管理に追加(U)</li> <li>OK</li> <li>キャンセル</li> </ul>



2 コードエディターで COBOL ソースコードを入力します。

プロジェクト「BATCHRPT」の作成が成功すると、COBOL 専用のコードエディターが起動しま す。エディター画面にコンソールアプリケーションのひな形が表示されるので、ソリューションエクス プローラーでソースプログラム「Program1.cbl」を右クリックして**名前の変更(M)**を選択し、プロ グラム名を「BATCHRPT.cbl」に書き換えます。

40 • • •	· 150	• • • • 160 • • • • 170 •	<ul> <li>         ・ ・ ・ ↓         ・ ・ ・ 188 ×         ・ ・ ・ 188 ×         ・         ・         ・</li></ul>
	6	開<(0)	
		ファイルを開くアプリケーションの	選択(N)
	Ţ	コンパイル	
		<i>ここ</i> まで検索(S)	
	Ē	新しい ソリューション エクスプロ	Iーラー のビュー(N)
		プロジェクトから除外(J)	
		プロジェクトを作成	
	Ж	切り取り(T)	Ctrl+X
	ŋ	⊐ピ–(Y)	Ctrl+C
	×	削除(D)	Del
$\rightarrow$	10	名前の変更(M)	ーム エクスプロ・・・・ クラス ピュ
		指令の確定	<del>т</del> Ф
		-	ק בעוע בעיב בעיב
		· · · 180 X	
		+	yリューション エクスプローラー の検索 (Ctrl+:)
			🕢 ソリューション 'BatchDemo' (1 プロジェクト)
			▲ BATCHRPT
		=	🔎 Properties
		-	
		2	
		-	



本章では既存資産の流用を想定して COBOL 正書法に従った伝統的スタイルのソースコードを入力し ますので、アスタリスクで始まるコメント行が 7 列目(エディター画面左側のグレー領域の右端)から始 まるよう注意して、以下の見出し部と環境部を入力します。 この時点では、データ部のファイル定義 未入力によるエラーとなりますが、ここでは無視して構いません。

IDENTIFICATION DIVISION. PROGRAM-ID. BATCHRPT. \* This program processes files: Input Files = Employee Extract File (Sequential) \* \* Selection Control Card \* Output File = Employee Yrs Employed Report \* ENVIRONMENT DIVISION. INPUT-OUTPUT SECTION. FILE-CONTROL. \* INPUT FILE: EMPLOYEE RECORDS SELECT EMP-SEQ-FILE ASSIGN TO UT-S-EMPSEQ. \* INPUT FILE: DATE SELECTION CRITERIA SELECT IN-CNTL-CARD ASSIGN TO UT-S-CNTLCARD. \* OUTPUT REPORT FILE SELECT EMP-HIRE-RPT ASSIGN TO UT-S-HIRERPT. マ ワ ワイック起動 (Ctrl+Q) ₽ = = × 🔀 BatchDemo - Microsoft Visual Studio (管理者) ファイル(E) 編集(E) 表示(Y) プロジェクト(E) ビルド(B) デバッグ(D) チーム(M) ツール(I) テスト(S) 分析(N) ウィンドウ(W) ヘルプ(H) サインイン 🎦 G - 〇 | 🎦 - 🔄 🔛 🚰 | ウ - ペ - | ト 開始 - 〇 - Debug - 🗴 x86 📲 🗐 🚚 🚛 🌆 🎬 🎽 🏛 🎁 🏭 🏭 🏭 📲 BATCHRPT.cbl → × • ¤ × - O Procedure Division ROGRAM1 · 86 エカスフローラー DENTIFICATION DIVISION. PROGRAM-10. BATCHRPT. ソリューション エクスプローラー の検索 (Ctrl+:) P ÷ 🐻 ソリューション 'BatchDemo' (1 プロジェクト) tanti 1000 1000 1000 1000 1000 A BATCHRPT 🔑 Properties ツールボックス BAT CHRPT. cbl -----Ser and ENVIRONMENT DIVISION INPUT-OUTPUT SECTION FILE-CONTROL SQL Server \* INPUT FILE: EMPLOYEE RECORDS SELECT EMP-SEQ-FILE ソリューション エク・・・ チーム エクスプロー クラスビュ ASSIGN TO UT-S-EMPSEQ. \* INPUT FILE: DATE SELECTION CRITERIA SELECT IN-CATL-CARD ASSIGN TO UT-S-CNTLCARD. - u × オブジェク COBOL ファイルプロパティ \* OUTPUT REPORT FILE SELECT EMP-HIRE-RPT ASSIGN TO UT-S-HIRERPT. □ 全般 data di orking-storage section. ソース フォーマット Fixed パス C:¥work¥Tutorials¥VS2 100 % プログラム BAT CHRPT. cbl 一覧 BATCHRPT プロジェクト ▼ - 😢 4 エラー 🗼 0 警告 🛛 0 メッセージ エラー一覧を検索 Q マネージョード false الاعترار HOPE 列 COBCH0244 : ファイル IN-CNTL-CARD に対 BATCHRPT.cbl する FD 記述項がない 18 19 BATCHRPT 3 2 ソース フォーマット 3 COBCH0244:ファイル EMP-HIRE-RPT に対 BATCHRPT.cbl する FD 記述項がれい 21 19 BATCHRPT SourceFormat 指令です。 シンボルの検索結果 データッール操作 呼び出し階層 エラー一覧 「備完」



データ部のファイル節を入力します。 なお、データ部のファイル定義を入力したので、環境部のエ ラーは無くなります。

> DATA DIVISION. FILE SECTION.

- FD EMP-SEQ-FILE LABEL RECORDS ARE STANDARD. 01 EMPLOYEE-RECORD PIC X (60).
- FDIN-CNTL-CARD<br/>LABEL RECORDS ARE STANDARD.01CONTROL-RECORDPIC X (8).
- FD EMP-HIRE-RPT LABEL RECORDS ARE STANDARD. 01 RPT-RECORD PIC X (80).





データ部の作業場所節で PROGRAM-FIELDS、CONTROL-REC データ項目を入力します。 COPY 文で外部参照する EMP-RECORD-IO-AREA データ項目はエラーとなりますが、無視して構いません。

WORKING-STORAGE SECTION.

01 PROGRAM-FIELDS.						
	05	EOF-FLAG	PIC X(01)	VALUE	'N'.	
		88 AT-EOF		VALUE	Ϋ́.	
		88 NOT-AT-EOF		VALUE	'N'.	
	05	COUNTERS.				
		10 EMP-REC-CNTR	PIC 9(05)	VALUE	0.	
		10 LINE-CTR	PIC 9(03)	VALUE	0.	
		10 LINE-MAX	PIC 9(03)	VALUE	60.	
	05	CURR-DATE.				
		10 CURR-YYYY	PIC 9(4).			
		10 CURR-MM	PIC 9(2).			
		10 CURR-DD	PIC 9(2).			
	05	CURR-TIME.				
		10 CURR-HR	PIC 9(2).			
		10 CURR-MIN	PIC 9(2).			
		10 CURR-SEC	PIC 9(2).			
	05	YRS-EMPLOYED	PIC 9(03)	COMP-3	VALUE 0	•
01	CON	TROL-REC.				
	05	CNTL-DATE.				
		10 CNTL-YR	PIC X(4)	VALUE	SPACE.	
		10 CNTL-MON	PIC X(2)	VALUE	SPACE.	
		10 CNTL-DAY	PIC X(2)	VALUE	SPACE.	

- \*\* Employee Record Layout
- 01 EMP-RECORD-IO-AREA. COPY EMPSEQ.

BATCHRPT.cbl + ×	
🔩 PROGRAM1	
1 · · · 10 · · · 20 · · · ·	30 • • • 40 • • • 50 • • • 60 • • • 70 • • • 80 • 5
36 WORKING-STORAGE SECTI	ON. +
37	and the second s
38 01 PROGRAM-FIELDS.	
39 U5 EOF-FLAG	PIC X(U1) VALUE INT.
40 88 AT-EUF 41 99 NOT-AT-EOF	
41 05 COUNTERS	VALUE N -
43 10 EMP-REC-CN	ITR PIC 9(05) VALUE 0.
44 10 LINE-CTR	PIC 9(03) VALUE 0
45 10 LINE-MAX	PIC 9(03) VALUE 60.
46 05 CURR-DATE.	
47 10 CURR-YYYY	PIC 9(4).
48 TU CURR-MM	
49 10 CORR-DD	PIC 9(2).
51 10 CURR-HR	PIC 9(2).
52 10 CURR-MIN	PIC 9(2).
53 10 CURR-SEC	PIC 9(2).
54 05 YRS-EMPLOYED	PIC 9(03) COMP-3 VALUE 0.
55	1000 1001
56 U1 CONTROL-REC.	
58 UD UNIL-DATE - 10 ONTL-VD	
59 10 CNTL-TR	PIC X(2) VALUE SPACE.
60 10 CNTL-DAY	PIC X(2) VALUE SPACE.
61	
62 🔪 🔹 🕹 🕹 🕹 🕹 🕹 🕹 62	ayout
63 01 EMP-RECORD-IO-AREA	
ee   procedure division.	
87 roback	
100 % 👻 🖣	•



データ部の作業場所節で RPT-TITLE-1 と RPT-TITLE-2 データ項目を入力します。

**	Repo	rt Lines				
01	RPT	-TITLE-1.				
	05	FILLER	PIC	X (20)	VALUE	
		'Program: BATCHRPT'.				
	05	FILLER	PIC	X (10)	VALUE	SPACES.
	05	FILLER	PIC	X (25)	VALUE	
		'Years Employed Report'				
	05	FILLER	PIC	X (10)	VALUE	SPACES.
	05	RPT-CURR-MM	PIC	Х(2).		
	05	FILLER	PIC	Х	VALUE	'/'.
	05	RPT-CURR-DD	PIC	Х(2).		
	05	FILLER	PIC	Х	VALUE	'/'.
	05	RPT-CURR-YYYY	PIC	X(4).		
	05	FILLER	PIC	X (5)	VALUE	SPACE.
01	RPT	-TITLE-2.				
	05	FILLER	PIC	X (67)	VALUE	SPACES.
	05	RPT-CURR-HR	PIC	X(2).		
	05	FILLER	PIC	Х	VALUE	' :' .
	05	RPT-CURR-MIN	PIC	Х(2).		
	05	FILLER	PIC	Х	VALUE	' :' .
	05	RPT-CURR-SEC	PIC	Х(2).		
	05	FILLER	PIC	X (5)	VALUE	SPACE.

BATCHRPT.cbl 👳 🗙		-
🔩 PROGRAM1	<ul> <li>Ø Procedure Division</li> </ul>	•
11	.0 • • • • 20 • • • • 30 • • • • 40 • • • • 50 • • • • 60 • • • • 70 • •	• • 80 • X
1       1       1         61       62       **         63       01         64       01         65       66         67       01         68       69         70       71         72       73         73       74         75       76         77       78         79       80         81       01         82       83         84       85         86       87         89       90         90       90         91       91	Employee Record Layout         EMP-RECORD-IO-AREA.         COPY_EMPSEQ.         Report Lines         RPT-TITLE-1.         05 FILLER       PIC X(20) VALUE         'Program: BATCHRPT'.         05 FILLER       PIC X(10) VALUE SPACES.         05 FILLER       PIC X(22) VALUE         'Years Employed Report'.       PIC X(10) VALUE SPACES.         05 FILLER       PIC X(22).         05 FILLER       PIC X(20).         05 RPT-CURR-MIN       PIC X(20).         05 FILLER       PIC X(20).	
100 % 👻 🖣		



作業場所節で RPT-TITLE-3 と RPT-COLUMNS データ項目を入力します。

01	RPT	-TITLE-3.		
	05	FILLER	PIC X(5)	VALUE SPACE.
	05	FILLER	PIC X(7)	VALUE '***** '.
	05	RPT-SELECTION-YYYY	PIC 9(4).	
	05	FILLER	PIC X(2)	VALUE '年'.
	05	RPT-SELECTION-MM	PIC Z9.	
	05	FILLER	PIC X(2)	VALUE '月'.
	05	RPT-SELECTION-DD	PIC Z9.	
	05	FILLER	PIC X(24)	VALUE
		'日以前に入社した社員一	覧'.	
	05	FILLER	PIC X(12)	VALUE SPACE.
01	RPT	-COLUMNS.		
	05	FILLER	PIC X(5)	VALUE SPACES.
	05	FILLER	PIC X(11)	VALUE
		'部署名'.		
	05	FILLER	PIC X(21)	VALUE
		'社員名'.		
	05	FILLER	PIC X(14)	VALUE
		'社員番号'.		
	05	FILLER	PIC X(15)	VALUE
		'入社日'.		
	05	FILLER	PIC X(14)	VALUE
		'雇用年数'.		

BAT CHRPT. cbl	<b>₽</b> X					<b>.</b>
🔩 PROGRAM1			🗕 🛛 🖓 Procedure	e Divisi <b>o</b> n		•
1 ·	· · · 1	0 · · · 20 · · · 30 · ·	40	50 • • • • 60 • •	· · · 70 · ·	• • 80 • X
88		05 FILLER	PIC X(5)	VALUE SPACE.		÷
89	01	DDT_TITLE_9				- 10-000 M - - 10-0000 -
91		05 FILLER	PIC X(5)	VALUE SPACE.		"We are
92	1	05 FILLER	PIC X(7)	VALUE '***** '	'• \	The sea
93		05 RPI-SELECTION-YYYY 05 FILLER	PIC 9(4). PIC X(2)	VALUE '年'。		1897° a 1897° a
95		05 RPT-SELECTION-MM	PIČ Ž9.			HTM.a
96		05 FILLER	PIC X(2)	VALUE '月'.		207 EL 1997 IRE 195 IRE
97		05 FILLER	PIC 29. PIC X(24)	VALUE		
99		'日以前に入社した社	員一覧'			16 8 66
100		05 FILLER	PIC X(12)	VALUE SPACE.		1 T
102	01	RPT-COLUMNS.				
103		05 FILLER	PIC X(5)	VALUE SPACES.		
104		UD FILLER '部署名'.	PIC X(11)	VALUE		
106		05 FILLER	PIC X(21)	VALUE		
107		<ol> <li></li></ol>	$\mathbf{PIC} \mathbf{V}(1A)$	VALUE		10 000
109		'社員番号'.	110 //(14/	VALUE		8
110		05 FILLER	PIC X(15)	VALUE		<b>--</b>
112	$\mathbf{A}$	\ 05 FILLER	PIC X(14)	VALUE		
113		'雇用年数'				
114	nro	cedure division.				
110	PIC					_
100 %					P	



作業場所節で RPT-DETAIL-LINE、RPT-TOTAL-LINE と BLANK-LINE データ項目を入力します。

01	RPT-DETAIL-LINE.					
	05	FILLER	PIC	X (5)	VALUE	SPACES.
	05	RPT-EMP-DIV	PIC	N(5)	VALUE	SPACES.
	05	FILLER	PIC	Х	VALUE	SPACES.
	05	RPT-EMP-NAME	PIC	N(10)	VALUE	SPACES.
	05	FILLER	PIC	Х	VALUE	SPACES.
	05	RPT-EMP-SSN	PIC	X (9)	VALUE	SPACES.
	05	FILLER	PIC	X (5)	VALUE	SPACES.
	05	RPT-EMP-HIRE-MM	PIC	X (2)	VALUE	SPACES.
	05	FILLER	PIC	Х	VALUE	'/'.
	05	RPT-EMP-HIRE-DD	PIC	X (2)	VALUE	SPACES.
	05	FILLER	PIC	Х	VALUE	'/'.
	05	RPT-EMP-HIRE-YYYY	PIC	X (4)	VALUE	SPACES.
	05	FILLER	PIC	X (5)	VALUE	SPACES.
	05	RPT-EMP-YRS-EMPL	PIC	Z9	VALUE	ZEROES.
	05	FILLER	PIC	X (12)	VALUE	SPACE.
01	RPT	-TOTAL-LINE.				
	05	FILLER	PIC	X (5)	VALUE	SPACE.
	05 FILLER 05 RPT-MSG			X (7)	VALUE	'**** ' <sub>.</sub>
				X (30)	VALUE	SPACE.
	05	FILLER	PIC	X (2)	VALUE	SPACE.
	05 RPT-TOT-RECS			ZZZ.		

01 BLANK-LIN	E
--------------	---

05 FILLER

PIC X(80) VALUE SPACE.

PIC X (33) VALUE SPACE.

BATCHRPT.cbl 👳 🗙			
🔩 PROGRAM1	•	Procedure Division	
1 · · · 1	0 • • • • 20 • • • • 30 • • • •	40 • • • • 50 • • •	• 60 • • • • 70 • • • 80 •
112 113 114	05 FILLER '雇用年数'.	PIC X(14) VALUE	
115 01 116 1 117 1 118 1 119 1 120 1 121 1 122 1 123 1 124 1 125 1 126 1 127 1 128 1 199 1 190 10 10 10 10 100 100 100 100 100 100	RPT-DETAIL-LINE.         05       FILLER         05       RPT-EMP-DIV         05       FILLER         05       FILLER         05       FILLER         05       FILLER         05       RPT-EMP-SSN         05       FILLER         05       FILLER         05       FILLER         05       RPT-EMP-HIRE-DD         05       FILLER         05       RPT-EMP-YRS-EMPL         05       FILLER         05       RPT-EMP-YRS-EMPL	PIC X(5)         VALUE S           PIC N(5)         VALUE S           PIC X         VALUE S           PIC X(2)         VALUE S           PIC X(3)         VALUE S           PIC X(2)         VALUE S           PIC X(4)         VALUE S           PIC X(5)         VALUE S           PIC X(5)         VALUE S           PIC X(5)         VALUE S           PIC X(5)         VALUE S           PIC X(12)         VALUE S	SPACES . SPACES
131 132 133 134 135 136 137 138 139 140 141 141 142 pro	RPT-TOTAL-LINE. OS FILLER OS FILLER OS RPT-MSG OS FILLER OS RPT-TOT-RECS OS FILLER BLANK-LINE	PIC X(5) VALUE S PIC X(7) VALUE Y PIC X(30) VALUE S PIC X(2) VALUE S PIC ZZZ. PIC X(33) VALUE S PIC X(80) VALUE S	SPACE. SPACE. SPACE. SPACE.
100 % 👻 🖣			•



最後に、手続き部の 1000-START 節の前半部分を入力します。PERFORM 文で参照する手続き名が 未定義なのでエラーが 5 件増えますが、気にせず先に進んでください。

PROCEDURE DIVISION. PERFORM 1000-START THRU 1000-EXIT. PERFORM 2000-MAIN-PROCESSING THRU 2000-EXIT UNTIL AT-EOF. PERFORM 9000-CLOSE-AND-CLEANUP THRU 9000-EXIT. STOP RUN. 1000-START SECTION. OPEN INPUT EMP-SEQ-FILE IN-CNTL-CARD. OPEN OUTPUT EMP-HIRE-RPT. \*\*\* \* SET UP AND WRITE REPORT TITLE AND COLUMN HEADINGS \*\*\* ACCEPT CURR-DATE FROM DATE YYYYMMDD. MOVE CURR-MM TO RPT-CURR-MM. MOVE CURR-DD TO RPT-CURR-DD. MOVE CURR-YYYY TO RPT-CURR-YYYY. ACCEPT CURR-TIME FROM TIME.

MOVECURR-HRTORPT-CURR-HR.MOVECURR-MINTORPT-CURR-MIN.MOVECURR-SECTORPT-CURR-SEC.

WRITE RPT-RECORD FROM RPT-TITLE-1 BEFORE ADVANCING 1 LINE. WRITE RPT-RECORD FROM RPT-TITLE-2 BEFORE ADVANCING 1 LINE.

🕏 PROGRAM1	<ul> <li>Procedure Division</li> </ul>	
1 · · · 10	9 • • • • 20 • • • • 30 • • • • 40 • • • • 50 • • • • 60 • • • • 70 • • • •	80 ·
1         1 <th1< th=""> <th1< th=""> <th1< th=""> <th1< th=""></th1<></th1<></th1<></th1<>	PERFORM       20       30       40       50       60       70       1000010000000000000000000000000000000	
165	WRITE RPT-RECORD FROM RPT-TITLE-T BEFORE ADVANCING T LINE.	analisiana.
168 169	goback.	



手続き部の1000-START節の後半部分を入力します。

\*\*\*
\* READ CONTROL CARD FILE TO GET DATE FOR SELECTION CRITERIA.
\* IF FILE IS EMPTY, DEFAULT CNTL-DATE TO CURRENT DATE.
\*\*\*
READ IN-CNTL-CARD INTO CONTROL-REC.
IF CNTL-DATE = SPACES
MOVE CURR-DATE TO CNTL-DATE
END-IF.
\* ACCEPT CNTL-DATE FROM SYSIN.
MOVE CNTL-MON TO RPT-SELECTION-MM.
MOVE CNTL-DAY TO RPT-SELECTION-DD.
MOVE CNTL-YR TO RPT-SELECTION-YYY.

WRITE RPT-RECORD FROM RPT-TITLE-3 BEFORE ADVANCING 1 LINE. WRITE RPT-RECORD FROM BLANK-LINE BEFORE ADVANCING 1 LINE.

WRITE RPT-RECORD FROM RPT-COLUMNS BEFORE ADVANCING 1 LINE. WRITE RPT-RECORD FROM BLANK-LINE BEFORE ADVANCING 1 LINE.

1000-EXIT.

EXIT.

BATCHRPT.cbl 👳 🗙	•
🔩 PROGRAM1	- 🛛 Procedure Division -
<u> </u>	0 · · · · 20 · · · · 30 · · · · 40 · · · · 50 · · · · 60 · · · · 70 · · · · 80 · X
165 166 167	WRITE RPT-RECORD FROM RPT-TITLE-1 BEFORE ADVANCING 1 LINE.
167         ***           168         ***           169         *           170         *           171         *           172         *           173         *           174         *           175         *           176         *           177         *           178         *           179         *           180         *	READ CONTROL CARD FILE TO GET DATE FOR SELECTION CRITERIA. IF FILE IS EMPTY, DEFAULT CNTL-DATE TO CURRENT DATE. READ IN-CNTL-CARD INTO CONTROL-REC. IF CNTL-DATE = SPACES MOVE CURR-DATE TO CNTL-DATE END-IF. ACCEPT CNTL-DATE FROM SYSIN. MOVE CNTL-DATE FROM SYSIN. MOVE CNTL-DAY TO RPT-SELECTION-MM. MOVE CNTL-DAY TO RPT-SELECTION-DD.
182 183 184 185 186 187 188 189 190 ⊟ 100 191 192 193 194 195 enc	WOVE CNIL-YR TO RPT-SELECTION-YYYY. WRITE RPT-RECORD FROM RPT-TITLE-3 BEFORE ADVANCING 1 LINE. WRITE RPT-RECORD FROM BLANK-LINE BEFORE ADVANCING 1 LINE. WRITE RPT-RECORD FROM BLANK-LINE BEFORE ADVANCING 1 LINE. DO-EXIT. EXIT. goback. H program Program1.



手続き部の 2000-MAIN-PROCESSING 段落と 3000-PROCESS-RECORD 段落の前半部分を入力し

#### ます。

```
2000-MAIN-PROCESSING.
     READ EMP-SEQ-FILE INTO EMP-RECORD-IO-AREA
         AT END MOVE 'Y' TO EOF-FLAG.
     IF NOT-AT-EOF
         PERFORM 3000-PROCESS-RECORD THRU 3000-EXIT
     END-IF.
 2000-EXIT.
    EXIT.
 3000-PROCESS-RECORD.
***
   FIRST, VERIFY EMPLOYEE'S HIRE DATE IS ON OR BEFORE DATE
*
    PASSED IN CONTROL CARD.
*
***
     IF EMPREC-DATE-OF-HIRE <= CNTL-DATE
        CONTINUE
     ELSE
        GO TO 3000-EXIT
     END-IF.
```





#### 手続き部の 3000-PROCESS-RECORD 段落の後半部分を入力します。

*** *	FORMAT REPORT DETAIL LIN	IES FROM EMPLOYEE RECORD.
-leileile	MOVE EMPREC-DIV	TO RPT-EMP-DIV.
	MOVE SPACE STRING EMPREC-JNAME1 SPACE EMPREC-JNAME2	TO RPT-EMP-NAME. DELIMITED BY SPACE DELIMITED BY SIZE DELIMITED BY SPACE INTO RPT-EMP-NAME.
	STRING EMPREC-SSN (1:7) '_' EMPREC-SSN (8:1)	DELIMITED BY SIZE DELIMITED BY SIZE DELIMITED BY SIZE INTO RPT-EMP-SSN.
	MOVE EMPREC-DOH-MM MOVE EMPREC-DOH-DD MOVE EMPREC-DOH-YYYY	TO RPT-EMP-HIRE-MM. TO RPT-EMP-HIRE-DD. TO RPT-EMP-HIRE-YYYY.
	PERFORM 4000-COMPUTE-YE MOVE YRS-EMPLOYED	EARS-EMPLOYED THRU 4000-EXIT. TO RPT-EMP-YRS-EMPL.
	WRITE RPT-RECORD ADD 1 TO EMP-REC-CNTR.	FROM RPT-DETAIL-LINE BEFORE ADVANCING 1 LINE.

3000-EXIT.

EXIT.





手続き部の 4000-COMPUTE-YEARS-EMPLOYED 段落を入力します。

4000-COMPUTE-YEARS-EMPLOYED.

\*\*\*\*

- \* DETERMINE YEARS OF EMPLOYMENT BY SUBTRACTING HIRE YEAR
- \* FROM CURRENT YEAR.

\*\*\*\*

COMPUTE YRS-EMPLOYED = CURR-YYYY - EMPREC-DOH-YYYY.

4000-EXIT. EXIT.





手続き部の 9000-CLOSE-AND-CLEANUP 段落を入力します。

9000-CLOSE-AND-CLEANUP.

IF EMP-REC-CNTR > 0MOVE '処理レコード件数:' TO RPT-MSG MOVE EMP-REC-CNTR TO RPT-TOT-RECS ELSE MOVE '処理レコードなし' TO RPT-MSG END-IF. DISPLAY '\*\*\* REPORT CREATED SUCCESSFULLY \*\*\*'. DISPLAY '\*\*\* VIEW: HIRERPT. DAT \*\*\*' WRITE RPT-RECORD FROM BLANK-LINE BEFORE ADVANCING 1 LINE. WRITE RPT-RECORD FROM RPT-TOTAL-LINE BEFORE ADVANCING 1 LINE. CLOSE EMP-SEQ-FILE IN-CNTL-CARD EMP-HIRE-RPT. 9000-EXIT. EXIT. BATCHRPT.cbl 🔩 PROGRAM1 - 📀 Procedure Division . . 50 . . . . 60 . 1. 4000-EXIT. EXIT. ÷ 9000-CLOSE-AND-CLEANUP. 25 25 IF EMP-REC-CNTR > 0 MOVE '処理レコード件数:' MOVE EMP-REC-CNTR TO RPT-MSG TO RPT-TOT-RECS ELSE 26 MOVE '処理レコードなし' TO RPT-MSG 26 26 26 26 26 26 26 26 26 END-IF. DISPLAY '\*\*\* REPORT CREATED SUCCESSFULLY \*\*\*'. DISPLAY '\*\*\* VIEW: HIREPT.DAT \*\*\*'. WRITE RPT-RECORD FROM BLANK-LINE BEFORE ADVANCING 1 LINE. WRITE RPT-RECORD FROM RPT-TOTAL-LINE BEFORE ADVANCING 1 LINE. CLOSE EMP-SEQ-FILE IN-CNTL-CARD EMP-HIRE-RPT. goback 文及び END PROGRAM は削除します。 9000-EXIT. EXIT. 100 % 👻 🖣 エラー一覧 • 4 × エラーー覧を検索 Q 🏹 🖸 12 エラー 🔽 🛯 警告 🛛 🗊 0 メッセージ 説明 ファイル 行 列 プロジェクト 😢 2 COBCH0217: このレベルの前述項目の長さがゼ BATCHRPT.cbl 67 8 BATCHRPT ロである COBCH0012:作用対象 EMPREC-DATE-OF-BATCHRPT.cbl 33 209 15 BATCHRPT HIRE が宣言されていない COBCH0562: ELSE 句に対する IF を指定して BATCHRPT.cbl BATCHRPT 211 12 34 いない。この文を無視する 出力 シンボルの検索結果 データッール操作 呼び出し階層 エラー一覧

以上で BATCHRPT.cbl ソースプログラムの入力は終了です。 ここでエラーが 12 件であれば、先 に進んでください。



3 コードエディターで COBOL コピーファイルを入力します。

ソリューションエクスプローラーでプロジェクト「BATCHRPT」を右クリックして 追加(D)、新しい項目(W) を選択します。

		Υ C	<b>P</b>	クイック起動 (Ctrl+Q	) 👂	- 8	×
-プ(H)						サインイン	> 🖪
и ма 🖕 🛛	Ŧ	5 2 / # # # <b>D</b> 6 / # 8					
				- ソリューション エク	スプローラー	Ŧ	Ψ×
Division				- COA	'o • <i>2</i> 🗗 🐌	= 4	"
00 · · · 110 · · · 120 · · · 130 · · · 140		· 150 · · · · 160 · · · · 170 · · · · 1	180	× ソリューション エク	スプローラー の検索	(Ctrl+:)	<u>ہ</u>
		100	-	👩 ソリューショ)	> 'BatchDemo' (	(1 プロジェク	ŀ)
		PUSCIN			HRPT		
	hictori			-C 4	perties TCHRPT chl		
		クローン(N)			I GITTE TADI		
	-	とこみ いえか(い) 奈日 オキソリューション エカフゴローラー のビュー(N	un l				
	<b>B</b> <sup>2</sup>		V)				
「回新しい項目(W) Ctrl+Shift+A		追加(D)		•			
た こうちゅう 「「」「」「」「」「」「」「」「」「」「」「」「」「」「」「」「」「」」「」「」		既存の COBOL 項目を追加					
゛ 新しいフォルダー(D)	ă	NuGet パッケージの管理					
	Ф	スタートアップ プロジェクトに設定(A)					

インストールされたテンプレートの一覧から COBOLプロジェクト項目、コピーブックを選択しま す。 名前(N)に EMPSEQ.cpy と入力し、追加(A) をクリックします。

	新しい項目の追加 - B.	ATCHRPT	? X
▲ インストール済み	並べ替え基準: 既定 → 誹詐 ▮	=	インストール済み テンプレート の検索 (Ctrl: 🔎 -
▲ COBOL プロジェクト項目 Native	COBOL プログラム	COBOL 加ジェクト項目	種類: COBOL プロジェクト項目 コピーブック変新現作成します。
PowerShell	D 28-799	COBOL 加ジェクト項目	
▶ オンライン	アプリケーション 構成 ファイル	COBOL プロジェクト項目	
	או איד ג-צוי 🚹	COBOL 加ジェクト項目	
	オンラインでテンプレートを検索するには、	<u>ここをクリックします。</u>	
名前(N): EMPSEQ.cpy			
			追加(A) キャンセル



EMPSEQ.cpy へ EMP-RECORD-IO-AREA データ項目のレコード記述を入力します。

05	EMP	-REC.		
	10	EMPREC-SSN	PIC X(08)	VALUE SPACE.
	10	FILLER	PIC X(01)	VALUE SPACE.
	10	EMPREC-JNAME1	PIC N(05)	VALUE SPACE.
	10	EMPREC-JNAME2	PIC N(05)	VALUE SPACE.
	10	EMPREC-NAME1	PIC X(05)	VALUE SPACE.
	10	EMPREC-NAME2	PIC X(05)	VALUE SPACE.
	10	EMPREC-GENDER	PIC X(01)	VALUE SPACE.
	10	FILLER	PIC X(01)	VALUE SPACE.
	10	EMPREC-DIV	PIC N(05)	VALUE ZERO.
	10	EMPREC-DATE-OF-HIR	Ξ.	
		15 EMPREC-DOH-YYY	Y PIC 9(04)	VALUE ZEROES
		15 EMPREC-DOH-MM	PIC 9(02)	VALUE ZEROES
		15 EMPREC-DOH-DD	PIC 9(02)	VALUE ZEROES
	10	FILLER	PIC X(01)	VALUE SPACE.

EMPSEQ.cpy 🗢 🗙 BA	AT CHRPT. cbl
🔩 EMPSEQ	- 💿 Procedure Division -
1	10 · · · · 20 · · · · 30 · · · · 40 · · · · 50 · · · · 60 · · · · 70 · · · 80 · · X
	EMPLOYEE SEQUENTIAL FILE LAYOUT
4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19	Q5 EMP-REC. 10 EMPREC-SSN PIC X(08) VALUE SPACE. 10 FILLER PIC X(01) VALUE SPACE. 10 EMPREC-JNAME1 PIC N(05) VALUE SPACE. 10 EMPREC-JNAME2 PIC N(05) VALUE SPACE. 10 EMPREC-NAME2 PIC X(05) VALUE SPACE. 10 EMPREC-NAME2 PIC X(05) VALUE SPACE. 10 EMPREC-GENDER PIC X(01) VALUE SPACE. 10 EMPREC-GENDER PIC X(01) VALUE SPACE. 10 EMPREC-DATE-OF-HIRE. 15 EMPREC-DOH-YYYY PIC 9(04) VALUE ZEROES. 15 EMPREC-DOH-MM PIC 9(02) VALUE ZEROES. 16 FILLER PIC X(01) VALUE ZEROES. 17 EMPREC-DOH-DD PIC 9(02) VALUE ZEROES. 18 EMPREC-DOH-DD PIC 9(02) VALUE ZEROES. 19 FILLER PIC X(01) VALUE SPACE.
100 % 👻 🖣	



ビルド(B) メニューから ソリューションのリビルド(R) を選択し、一度コンパイルします。

×	BatchDerr	io - Microso	oft Visual St	udio	(管理者)					7 <b>Ç</b>	) グ
ファ	イル <b>(F) 編集(E</b>	) 表示(V)	プロジェクト(P)	U	ド(B) デバッグ(D)	チーム(	(M)	ツール <b>(</b> T)	テスト(S)	分析(N)	ウイン
	G • 🗇 📅 •	😩 🗳 💾	9 - 9 - 1	*	ソリューションのビルド(	в)		Ctrl+	Shift+B	) 🖕 🖷	E
4	EMPSEO.cpv	BATCHR	RPT.cbl 👳 🗙		ソリューションのリビルド	(R)					•
5	4 BATCHRPT				ソリューションのクリーン	(C)					-
R	1 ·	· · · 10 · ·	20		ソリューションでコード分	)析を実行	<del>Γ</del> (Υ)	Alt+	F11	· · 80 ·	·Х
Ř	202	EXI	IT.	*	BATCHRPT のビルド	(U)				÷	_
	204	3000-PR	ROCESS-RECORD		BATCHRPT ருப்ப	ሶ(E)					
U I	205 🖻 206	*** * FIRS	ST, VERIFY E		BATCHRPT のクリー	>(N)					

エディター画面の BATCHRPT.cbl [コード]タブをクリックして、表示(V)メニューから エラーー 覧(I) を選択します。 エラーが 0 件であることを確認して、次に進んでください。

M BatchDemo - Microsoft Visual Studio (管理者)	マ  ゆ //w/ 記動 (Ctrl+Q)  P  -  ロ ×
アイル(上) 編集(上) 表示(Y) ノロジェクト(L) アハック(L) チーム(M) ツール(L) アスト(L) ◎ • ○   習 • 😩 💾 💾   フ • ♡ → 開始 • ○ → Debug •   x86 • ↓ 💷   第 =	分析(M) ウインドウ(W) ベルレ(H) サインイン M ※ 吉 唯[ 国 強   ■ 領 領 領 復 ]。"
Image: Construct of the second sec	
215       ***       FORMAT REPORT DETAIL LINES FROM EMPLOYEE RECORD.         217       ***       MOVE EMPREC-DIV       TO       RPT-EMP-DIV.         100%       MOVE SPACE       TO       RPT-EMP-NAME.         100%       MOVE SPACE       TO       TO         100%       MOVE SPACE       TO       TO         100%       TO       TO       TO         100%       TO       TO       TO         100%       TO       TO       TO         100%       TO	コピーファイルとプログ ラムの依存関係がツリー 状で表示されます。 <sup>1/1</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup> <sup>1/2</sup>



4 COBOLコンパイル指令を追加します。

ファイル名の割り当てを EXTERNAL(外部割り当て)に変更するため、プロジェクト(P)メニューから BATCHRPT のプロパティ(P) を選択します。



COBOL タブを選択し 追加指令に assign(external) を入力し、プロパティファイルを保存しま

す。

BATCHRPT 🗢 🗙 EMPSEQ.cpy	BATCHRPT.cbl	<u> </u>
アプリケーション SQL	構成(C): アクティブ (Debug)    プラットフォーム(M): アクティブ (x86)	
コピーブック プリプロセッサ	□ 指令ファイルの生成 □ リストファイルを生成	
СОВОГ	▶ □ コードカバレッジを有効にする □ プロファイラを有効にする	
COBOL リンク デバッグ	追加指令	
Micro Focus Code Analysis	Pssign(excernar)	~
	[	高度
	ビルド設定	



5 アプリケーション構成ファイルを作成します。

ソリューションエクスプローラーでプロジェクト「BATCHRPT」を右クリックして追加(D)、新し

い項日(W)					🔄 א אין דאר אין דער צער ד	クスプローラー	- ₽ ×
を選択しま					0 0 <b>(</b> )	`o • 2 🗇 🗿 🖇	• 🗕 👘
で送入しる					ב לפל-בווע	:クスプローラー の検索 (Ct	rl+:) 🔑 -
す。				_	אין	ョン 'BatchDemo' (1 プ	םאַבלא)
			*	ビルド(U)		CHRPI	
				リビルド(E)		AT CHRPT. cbl	
				クリーン(N)		I EMPSEQ.cpy	
				<u>ここまで検索(S)</u>		MPSEQ.cpy	
				新しい ソリューション エクスプローラー のビュー(N)			
	Ĩ□ 新しい項目(W)	Ctrl+Shift+A		追加(D)	•		
	<b>1</b> 既存の項目(G)	Shift+Alt+A		既存の COBOL 項目を追加			
	👘 新しいフォルダー(D)		苗	NuGet パッケージの管理			
	泰昭		Ф	スタートアップ プロジェクトに設定(A)			
-	De ancie			-> 10BZ=3			

インストールされたテンプレートの一覧から COBOL プロジェクト項目、アプリケーション構成フ

アイルを選択し、追加 (A) をクリックしま す。ファイル名はデフ ォルトのままで構いま せん。

		新しい項目の追加 -	- BATCHRPT	? X
▲ インストール済み	並べ替え基	準: 既定	• # E	インストール済み テンプレート の検索 (Ctrl・ 🔎 -
▲ COBOL プロジェクト項目 Native PowerShell	<b>⊡</b> =	COBOL プログラム コピーブック	COBOL 加ジェクト項目 COBOL 加ジェクト項目	種類: COBOL プロジェクト項目 アプリケーションの設定を構成するために使うファ イルです。
▶ オンライン		アプリケーション 構成 ファイル	COBOL プロジェクト項目	
	י (ב	Z 7™11	COBOL 九ジェクト項目	
	đ	ンラインでテンプレートを検索するに	は、ここをクリックします。	
名前( <u>N</u> ): Application.con	fig			追加(A) キャンセル

生成されたファイルをダブルクリックします。アプリケーションの設定で名前に dd\_EMPSEQ、値

に Emp_Master.dat を入	アプリケーションの設定	x
カし、 <b>設定</b> をクリックしま	環境 COBOL スイッチ 実行時構成	
す。	変数 値 名前 dd_EMPSEQ 値 Emp_Master.dat 設定 削除	
	OK         キャンセル	



**アプリケーションの設定**で名前 に dd\_CNTLCARD、値に Cntl\_Card.dat を入力し、**設定** をクリックします。

	アプリケー	ションの設定
環境	COBOL スイッチ 実行時構成	
空港	<u></u>	値
dd	_EMPSEQ	Emp_Master.dat
名前	ĵ dd_CNTLCARD	
値	Cntl_Card.dat	
		設定 削除
Ok	キャンセル	

**アプリケーションの設定**で名 前に dd\_HIRERPT、値に Hire\_Report.dat を入力し、 設定をクリックします。

לת	りケーションの設定 エ
環境 COBOL スイッチ 実行時構成	
変数 dd_EMPSEQ dd_CNTLCARD	値 Emp_Master.dat Cntl_Card.dat
名前 dd_HIREKPI	
	1 設定   削除
ОК <b>+</b> трусл	

**アプリケーションの設定**で

**OK** をクリックします。

		:	アプリケー	ーションの設定	x
環境	COBOL スイッチ	実行時構成			
变; dd, dd, dd,	数 _EMPSEQ _CNTLCARD _HIRERPT			値 Emp_Master.dat Cntl_Card.dat Hire_Report.dat	
名神	jū				
				設定 削除	
OI	< <b>\$</b> \$77.72	Սե			



6 COBOL アプリケーションをビルドします。

ソリューション構成が Debug、ソリューションプラットフォームが x86 であることを確認して、 ビルド(B)メニューから 🔀 BatchDemo - 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) ツール(T) テスト(S) 分 ソリューションのリビル G - ○ | 酒 - 当 || || || || ワ - ペ - | | || || || - ションのビルド(B) Ctrl+Shift+B ド(R) を選択します。 出 ソリューションのリビルド(R) BATCHRPT\* + × EMPSEQ.cpy ソリューションのクリーン(C) 7 カウィンドウにビルド結 アプリケーション ソリューションでコード分析を実行(Y) Alt+F11 構成 エクスプロ-71 SOL 🛓 BATCHRPT のビルド(U) 果が表示されるので、す コピーブック BATCHRPT のリビルド(E) プリプロセッサ CHRPT のクリーン(N) べてのビルドが正常終了 したことを確認します。 出力 • I X - 🖺 🖆 🎽 🔁 出力元の表示(S): ビルド ---- すべそのリビルド開始: ブロジェクト:BATCHRPT, 構成:Debus x88 またi¥work¥Tutorials¥VS2013¥BatchDemo¥BATCHRPT¥BATCHRPT.cbl のコンハイル中 \* Generating obj¥x86¥Debus¥BATCHRPT \* Data: 2712 Code: 3960 Literals: COBOL ユンバイル:1 個 正常終了または最新の状態 0 個 失敗。 1272 300000 コン・トービーローロット またしる取利レイルを 0 回 大社Xo BATCHRPT -> C:¥work¥Tutorials4VS2013¥BatchDemo¥BATCHRPtbii¥X86¥Debug¥BATCHRPT.exe ======= すべてリビルド: 1 正常終了、0 失敗、0 スキッブ ======== BATCHRPT 出力 シンボルの検索結果 データツール操作 呼び出し階層 エラー一覧

リビルドがすべて正常に終了しまし

#### 7 入力ファイルをコピーします。

前章で作成した **Emp\_Master.dat** ファイルを、デバッグフォルダ(<第5章第5章1で指定した フォルダ> ¥LoadCVSFile¥LoadCVSFile ¥bin¥x86¥debug)からデバッグフォルダ(<第6章1で指定 したフォルダ> ¥BATCHRPT¥BATCHRPT¥bin¥x86¥debug)にコピーします。





8 制御ファイルを作成します。

デバッグフォルダ(<第6章1で指定したフォルダ>¥BATCHRPT¥BATCHRPT¥bin¥x86¥debug) にメモ帳などを利用して以下のデータが記述された Cntl\_Card.dat ファイルを作成します。

#### 20110101

🎉   🕞 🚺 = I		Debug		_ □	x
ファイル ホーム 共有 表示					~ <b>?</b>
⋲ 💿 ▼ ↑ 퉬 « bin ▶ x86	5 🕨	Debug v 🖒	Debugo	D検索	Q,
鷆 Tutorials	^	名前		更新日時	種類
🕌 Edipse		🗾 BATCHRPT. exe		2015/09/03 17:38	アプリク
Databorna		📄 BAT CHRPT. exe. mfgcf		2015/09/03 17:36	MFGC
		BAT CHRPT.idy		2015/09/03 17:38	IDY 7
BAI CHRPI		Cntl_Card.dat		2015/09/03 17:43	DAT 5
🧊 bin 鷆 x86		Emp_Master.dat		2015/09/03 15:17	DAT 🤉
🍌 Debug					
퉬 obj 📔 Properties	≡				
🐌 NativeConsole					
] Tutorials	Ц.				
Min Form	$\mathbf{\mathbf{v}}$	<			>
5個の項目 1個の項目を選択 8,	КЧÞ	~			:==

🞯 Cntl_Card.dat - TeraPad 💶 🗖	x
ファイル(E) 編集(E) 検索(S) 表示(Y) ウィンドウ(W) ツール(T) ヘルプ(H)	
Q C Q V N N N N N N N N N N N N N N N N N N	
	~
	>
1行: 9桁 標準 [60] SJIS CRLF 挿入	H



9 COBOL アプリケーションをデバッグ実行します。

デバッグ(D)メニューから ステップイン(I) を選択するか F11 キーを押すと、コマンドプロンプ

ト画面が開き、デバッガー がステップ実行を開始しま す。 デバッガーは手続き部 の最初の COBOL 文である PERFORM 文を実行する手 前で処理を中断します。



制御ファイルから読み込んだレコードの 内容を確認するため、データ部の CONTROL-REC上で右クリックして ウ オッチ式の追加(W)を選択します。



同様に入力ファイルから読み込んだレ コードの内容を確認するため、データ部の EMP-RECORD-IO-AREA 上で右クリッ クして ウォッチ式の追加(W) を選択しま す。





手続き部 1000-START 節の READ 文に続く IF 文でエディター画面の左端をクリックし、ブレーク

ポイントを設定します。



同様に手続き部 2000-MAIN-PROCESSING 段落の READ 文に続く IF 文でエディター画面の左端



デバッグ(D)メニューから 続行(C) を選択するか F5 キーを押すと、デバッガーは最初のブレーク ポイントで実行を中断します。

ウォッチ式の CONTROL-REC の値 に制御ファイルから読 み込んだレコードが表 示されます。

BATCHRPT	EMPSEQ.cpy	BATCHRPT.cbl → ×						-
🔩 BAT CHRPT	🔩 BATCHRPT - 🛛 🗘 1000-START() -							
• • • • • • • • • • • • • • • • • • •	*** * READ CONTROL * IF FILE IS EN * READ IN-CNT IF CNTL-DAT MOVE CL END-IF. * ACCEPT CNTL	CARD FILE TO GET DAT PTY, DEFAULT ONTL-DA L-CARD INTO CONTROL- E = SPACES RR-DATE TO CNTL-DATE	E FOR SELECTION CRITE TE TO CURRENT DATE. REC.	••• 70 •		• 80 • • • • •		106 X
100 /8 · ·				- I	x	亚心虫 屍座	, 	Π×
名前	<u>値</u> -REC {長さ=8}:"2 ORD-IO-AR {長さ=60}:"	:0110101"	0000000, d	型 GROUP GROUP	· ·	名前 BATCHRPT.10 BATCHRPT.PA	00-START C	記 :0B :0B
目動変数 ローカル	ウオッチ 1					呼… ブレ… コ	₹··· {₹···	出力



デバッグ(D)メニューから 続行(C) を選択するか F5 キーを押すと、デバッガーは2番目のブレー クポイントで実行を中断します。

ウォッチ式の EMP-RECORD-IO-AREA の値に 入力ファイルから読み込ん だ1番目のレコードが表示 されます。

195         AT END           196         197           197         197           198         PERFOR           199         200           201         2000-EXIT.           202         4	MOVE 'Y' TO EOF-FLAG EOF M 3000-PROCESS-RECORD	THRU 3000-	EXIT		
ウオッチ 1				<b>-</b> ₽	х
名前	値			型	-
▷ 🔎 CONTROL-REC	{長さ=8}: "20110101"		9	GROUP	
🔎 🥥 EMP-RECORD-IO-AREA	{長さ=60}: "11111113 佐藤	隆 카	9 975 NQ	GROUP	
🖌 🧉 EMP-REC OF EMP-RECORD-	{長さ=60}: "11111113 佐藤	隆助	ሳ タカシ ℕ  ቍ	GROUP	
EMPREC-SSN OF EMP-RE	11111113		Q +	PIC X(8)	
FILLER OF EMP-REC OF EI			Q	PIC X	
EMPREC-JNAME1 OF EMI	佐藤		Q. +	PIC N(5)	
EMPREC-JNAME2 OF EMI	隆		Q	PIC N(5)	
EMPREC-NAME1 OF EMP	<u> </u>		Q	PIC X(5)	
EMPREC-NAME2 OF EMP	<u>タカシ</u>		Q	PIC X(5)	
EMPREC-GENDER OF EMI	M		Q	PIC X	
FILLER OF EMP-REC OF EI			Q	PIC X	
EMPREC-DIV OF EMP-REC	営業部		Q	PIC N(5)	
EMPREC-DATE-OF-HIRE	{長さ=8}: "19980401"		Q, <del>.</del>	GROUP	
FILLER OF EMP-REC OF EI			٩.	PIC X	
					T
自動変数 ローカル ウォッチ 1					

同様に デバッグ(D)メニューから 続行(C) を選択するか F5 キーを押すと、デバッガーは2番目

のブレークポイントで実行を 中断します。

ウォッチ式の EMP-RECORD-IO-AREA の値に 入力ファイルから読み込んだ 2 番目のレコードが表示され ます。

· ウォッチ 1			<b>-</b> ₽ ×
名前	値		型 🔺
CONTROL-REC	<del>{長さ=8}: "20110101"</del>	٩	GROUP
🖌 🥥 EMP-RECORD-IO-AREA	{長さ=60}: "22222226 鈴木 尚	6之 スズキ ナオユキ�・	GROUP
A @ EMP-REC OF EMP-RECORD-	{長さ=60}: "22222226 鈴木 尚	6之 スズキ ナオユキ�・	GROUP
EMPREC-SSN OF EMP-RE	22222226	Q	PIC X(8)
FILLER OF EMP-REC OF EI		Q	PICX
EMPREC-JNAME1 OF EMI	鈴木	Q.+	PIC N(5)
EMPREC-JNAME2 OF EMI	尚之	Q	PIC N(5)
EMPREC-NAME1 OF EMP	λλ°‡	Q	PIC X(5)
EMPREC-NAME2 OF EMP	ŵg	Q	PIC X(5)
EMPREC-GENDER OF EMI	M	Q, -	PICX
FILLER OF EMP-REC OF EI		Q	PICX
EMPREC-DIV OF EMP-REC	技術部	Q, -	PIC N(5)
🕨 🥥 EMPREC-DATE-OF-HIRE i	{長さ=8}: "19981015"	Q, -	GROUP
FILLER OF EMP-REC OF EI		Q, +	PICX
			<b>T</b>
自動変数 ローカル ウオッチ 1			

さらに **F5** キーを 8 回、 **F11** キーを 1 回押すと、デバッガーは 2 番目のブレークポイントに続く EXIT 文で実行を中断します。

IF 文の条件式は、入力ファイル がファイル終了状態であることを示 しています。

	BATCHRPT	EMPSEQ.cpy	BATCHRPT.cbl 👳 🗙	
	🔩 BAT CHRPT			- ♀ 2000-EXIT()
	1.	· · · 10 · · · · 20 ·	· · · 30 · · · 40	• • • • 50 • • • • 60 • • • • 70 •
	195	AT END	MOVE 'Y' TO EOF-FL	AG.
-	196		0E	
	109	IF NUI-AI-E	2000 DDOOFEE DEOOD	D THDU 9000 EVIT
	100	END_IE	JUUU-PROCESS-RECOR	D THRU 3000-EXIT
	200	LND-11 •		
	201	2000-EXIT.		
$\rightarrow$	⇒ 202	EXIT.		
	203			
	204 🖻	3000-PROCESS-RE	CORD.	
	205	***		
	100 % 👻 🖣 👘			



デバッグ(D)メニューから 続行(C) を選択するか STOP 文を実行するまで F11 キーを押すと、デ バッガーは終了します。



デバッグフォルダ(<第6章1で指定したフォルダ>¥BATCHRPT¥BATCHRPT¥bin¥x86¥debug) に **Hire\_Report.dat** ファイルが作成されるので、メモ帳などでファイルを開き、社員9名分のデー タが表示されることを確認します。



0       □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□	4 5		11年。1月。1日以前 社員名	前に入社した社員一覧	入社日		=	
	7 8 9 10 11 12 13 14 15 16 17 18	2000000000000000000000000000000000000	佐藤 隆	1111111-3 222222-6 3333333-9 444444-2 555555-5 6666666-8 7777777-1 8888888-4 39999999-7	04/01/1998 04/01/1998 04/01/1999 04/01/2000 04/01/2001 04/01/2003 04/01/2003 04/01/2005			
٢	<	1				>		1



デバッグフォルダ(<第6章1で指定したフォルダ>¥BATCHRPT¥BATCHRPT¥bin¥x86¥debug) の Cntl\_Card.dat ファイルを以下の値に更新します。

20000101



**デバッグ(D)**メニューから **デバッグなしで開始(H)** を選択するか **Ctrl+F5** キーを押すと、コマンドプロンプト画面が開くので、任意のキーを押してアプリケーションを実行します。

デバッグフォルダ(<第6章1で指定したフォルダ>¥BATCHRPT¥BATCHRPT¥bin¥x86¥debug) の **Hire\_Report.dat** ファイルを開いて、2000年1月1日以前に入社した社員3名分のデータだけ が表示されることを確認します。

🞯 Hire_Report.dat - TeraPad 🗕 🗖	x
ファイル(E) 編集(E) 検索(S) 表示(Y) ウィンドウ(W) ツール(I) ヘルプ(H)	
🖸 🖆 🗳 X 🐚 🛍 🖉 💁 💭 💭	
Construction (10, 1, 1, 1, 120, 1, 1, 1, 130, 1, 1, 1, 140, 1, 1, 1, 150, 1, 1, 1, 160, 1, 1, 1, 170, 1, 1, 1, 180     Construct State (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1	-
3   1999-1*****。2000年。1月、1日以前に入社した社員一覧。1999-1999-1999-1999-1999-1999-1999-199	=
5 000000部署名000000社員名000000000000000社員番号000000入社日00000000雇用年数0000000↓ 6	-
7営業部	
10 11 12 [EOF]	
	$\mathbf{r}$
1行: 1桁 標準 [80] SJIS CRLF 挿入	



デバッグフォルダ(<第6章1で指定したフォルダ>¥BATCHRPT¥BATCHRPT¥bin¥x86¥debug) の Cntl\_Card.dat ファイルを以下の値に更新します。

19980101



デバッグ(D)メニューから デバッグなしで開始(H) を選択するか Ctrl+F5 キーを押すと、コマンドプロンプト画面が開くので、任意のキーを押してアプリケーションを実行します。

デバッグフォルダ(<第6章1で指定したフォルダ>¥BATCHRPT¥BATCHRPT¥bin¥x86¥debug) の **Hire\_Report.dat** ファイルを開いて、処理レコードなしが表示されることを確認します。

Hire_Report.dat - TeraPad	x I
ファイル(E) 編集(E) 検索(S) 表示(Y) ウィンドウ(W) ツール(I) ヘルプ(H)	
0	< III >
	>
1行: 1桁 標準 [80] SJIS CRLF 挿入	ы



2015年11	月 01 日	初版
マイクロフォ	ーカス株式会社	
〒106-0032	東京都港区六本木 7-18-18 住友不動産六本木通ビル 9F	
電話 URL	03-5413-4800 http://www.microfocus.co.	.jp/