Micro Focus Visual COBOL for Eclipse

チュートリアル





はじめに

Micro Focus Visual COBOL for Eclipse は、高機能なオープンソースの IDE(統合開発環境) として広く普及する Eclipse 上で COBOL のアプリケーション開発を行うための製品です。 COBOL プログラマが既存の COBOL 資産を Windows、UNIX/Linux といったオープン環境で活 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最適な製品です。

本書は、Micro Focus Visual COBOL for Eclipse(Windows)を学ぶためのチュートリアルで す。本書の読者は、プログラミングの基礎知識があり、かつ Windows の基本操作を理解してい るものとします。

また、本書に掲載している画面イメージは Windows 10 Pro 64 bit 版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。



第1章 環境のセットアップ

Micro Focus Visual COBOL for Eclipse は、COBOL プログラミングの IDE として Eclipse を利用 します。本製品には、Eclipse 4.7(64 bit)がバンドルされていますが、既に 32bit 版 Eclipse 4.6 も しくは 64 bit 版 Eclipse 4.6 をお使いの場合は、お使いの Eclipse にプラグイン形式で本製品をイ ンストールすることも可能です。本章ではインストーラーを使った方法を紹介いたします。

1 入手したインストールプログラム vce_40.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。

III Micro Focus Visual COBOL for Eclipse セットアップ ー			\times
	Micro Focus Visual COBOL for Eclipse		
U Focus	バージョン 4.0.00243		
	製品インストール場所:		
	C:\Program Files (x86)\Micro Focus\Visual COBOL	参照	(<u>B</u>)
	Eclipse インストール場所:		
	C:\Users\Public\Micro Focus\Visual COBOL	参照	(<u>B</u>)
	ここをクロックレスフロレース ノートを表示		
	□ 同意する(A) <u>エンドユーザー使用許諾契約書</u> ●	インストール	-(I)



3 エンドユーザー使用許諾契約書が表示されます。

 Tンドユーザライセンス認知 MICRO FOCUS® ENTERPRISE DEVELOPER 4 MICRO FOCUS® ENTERPRISE SERVER FOR NET V4 MICRO FOCUS ENTERPRISE SERVER FOR NET V4 MICRO FOCUS ENTERPRISE TEST SERVER FOR NET V4 MICRO FOCUS ENTERPRISE TEST SERVER PAGE MICRO FOCUS ENTERPRISE MICRO FOCUS ENT

ブラウザが起動し、エンドユーザー使用許諾契約書の内容が表示されますので内容を確認しま す。

4 インストールを開始します。

表示された内容に問題なければ、**同意する(A)** にチェックを入れ [**インストール(I)**] ボタンを クリックしてインストールを開始します。

Micro Focus Visual COBOL	. for Eclipse セットアップ	-		\times
	Micro Focus Visual COBOL for Eclipse			
L'FOCUS	バージョン 4.0.00243			
	製品インストール場所:			
	C:\Program Files (x86)\Micro Focus\Visual COBOL		参照(<u>3</u>)
	Eclipse インストール場所:			
	C:\Users\Public\Micro Focus\Visual COBOL		参照(<u>3</u>)
	☑ 同意する(A) エンドユーザー使用許諾契約書		ストール	D



Micro Focus Visual COBO	. for Eclipse セットアップ	_		×
	Micro Focus Visual COBOL for Eclipse			
L Focus	セットアップ進行状況			
	処理中: Java Platform SE 8 U162			
		+ †	ンセル(の	

インストールの処理が完了したら、[閉じる(C)] ボタンをクリックします。

IIII Micro Focus Visual COBOL for Eclipse セットアップ		-		×
	Micro Focus Visual COBOL for Eclipse			
L Focus	セットアップの成功			
		ŀ	閉じる(<u>C</u>)	



以上で、チュートリアルの準備は終了しました。 Windows のスタートメニューに「Micro Focus Visual COBOL」が登録され、その配下に Visual COBOL for Eclipse が追加されていることを確認 してください。





第2章 Eclipse IDE に慣れよう

Eclipse 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 Eclipse に習熟されている方は、本章を読み飛ばしてください。

Eclipse の IDE を使う場合、ビュー、エディター、ツールバー、メニューバー、ステータスバー、 パースペクティブから構成されるワークベンチというウィンドウ内で作業します。パースペクティブ は、行いたい作業によって切り替えて利用するワークベンチのレイアウト(表示するビュー、メニュ

、ツールバーの種
 類や場所)のことを
 指します。パースペ
 クティブを切り替え
 ることによって目的
 の作業に応じた構成
 要素を揃えることが
 できます。各要素の
 配置はカスタマイズ
 可能です。



Eclipse のワークスペースとプロジェクトには、アプリケーションの作成に必要なビルドパス、デー タ接続、フォルダー、およびファイルを表す項目等が含まれています。 ワークスペースには複数のプ 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。COBOL エクス プローラーには、ワークスペースに紐づけられたプロジェクト、それらのプロジェクト内の項目が階層 状に表示されます。COBOL エクスプローラー上で目的の要素を検索し、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、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。パースペクティブによって は異なる名称のビューが同等の用途のため に用意されています。例えば Java パース ペクティブであればパッケージエクスプロ ーラーという Java 開発時に必要な要素を フィルター表示するビューが紐つけられて います。





COBOL パースペクティブに紐つくエディターには、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。ソー スコードを入力するとバックグラウンドチェックを実行して、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。

٥	*UDT0033Rcb	I 🔀 🖻 CUISCRN.сы	-	' 🗆
г	UDT003	3Rcbl 🕨		
E	· · · • • • *	A · 1 · B · · • · • · · · · · · · · · · · · ·	I · • · · · · 8	
Г	018100*	MODULE-NO. <2.0> *		^ =
	018200*	MODULE-ID. RENEWAL-DISCOUNT-RTN *		
	018300*	MODULE-NAME. 契約継続割引調整 *		
	018400*	*		
	⊖018500	RENEWAL-DISCOUNT-RTN SECTION.		
	⊖018600	RENEWAL-DISCOUNT-000.		
	018700*			
8	018800	EVALUTE CONTRACT-YRS		
	018900	⁹¹ COBCH0641S 手続き名宣言の直後に 😳 (終止符) がない。または認識できない動詞がある		
	019000	MOVE ZERO TO WK-RENEWAL-DISCOUNT-RATE		
8	019100	WHEN 1		
	019200	MOVE RENEWAL-INIT TO UK-RENEWAL-DISCOUNT-RATE		
8	019300	WHEN OTHER		
	019400	COMPUTE WK-RENEWAL-DISCOUNT-RATE =		
	019500	RENEWAL-INIT + (CONTRACT-YRS - 1) * RENEWAL-YRLY		
8	019600	END-EVALUATE.		
	019700	COMPUTE HOKEN-RYOUKIN =		
	019800	HOKEN-RYOUKIN * (1 - WK-RENEWAL-DISCOUNT-RATE).		
	019900*			¥
	<		>	

プロジェクトを右クリックし [プロパティー(R)] → [Micro Focus COBOL] と遷移し表示される 各画面では COBOL アプリケーションのビルド方法等を構成します。

📼 プロパティ: NativeCOBOL				• ×
フィルタ入力	COBOL		↓ •	⇒ • •
フィルタカカ > リソース ▲ Micro Foous ビルドオス ▲ ビルド構成 ▲ [C0B0L] S0L プリプロセッサ コード分析 追加のプリプロセッサ イベント ビルド環境 > リンク > プロラエクト設定 > 深行時構成 Protect Archives	COBOL New Configuration [使用中] 出力パス: New Configuration.bin エントリポイント: - ターグット放在 - ダインドンド - マイントの確認 - マプロシェクトの COBOL の設定の上書者 -	ブラットフォーム ダーゲット ● 32 ビット ◎ 64 ビット	 ・・ 構成の 構成の Micro Focus ライブラリ LBR (c/(ッケージ化) 	
Project Facets Server > Task Repository Task Tags > Velidation WikiText ビルダー プロジェクト参照 実行/デバッグ設定	 プロシェクトの (GBCL の数をの工事者) 「構成の圏内な設定を可能にする(<u>c</u>) 文字集合: 言語方言: ソース フォーマット: 	ASCII Micro Focus 固定	•	-
0	14 [m	デフォルトの復元(I) 通用(L) OK キャンセル	•



コンソールビュー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。問題ビューには、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。

📃 🖳 レンジ 🔄 Micro Focus Unit Testing
Micro Focus 신나는 InsuEstimOnline
[echo] Project Location: 'pathVar.ECLIPSE_HOME'=C:/Users/Public/Micro Focus/Visual COBOL/eclipse
[echo] Project Location: 'pathVar.PARENT_LOC'=C:/work/demo/EstimOnline
pre.build.cfg.New_Configuration:
cobol.compile.cfg.New_Configuration:
[cobo1]
[cobol] Compiling UDTOO33P.cbl
[cobol] * チェック終了:エラーはありません— コード 生成を開始します
[cobol] Compilation complete with no errors
cobol.link.cfg.New Configuration:
[cobollink] Linking UDT0033P.dll
[cobollink] Micro Focus COBOL - CBLLINK utility

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、デバッグ機能を利用します。 Eclipse 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、変数ビュー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。

~ 画 RunLegacyOnlineApp [COBOL アブリケーション] 名前	值
🗸 😪 Micro Focus デパツガ: C:¥work¥demo¥EstimOnline¥InsuEstimOnline¥New_Confi: 🛛 🗸 🗣 WRK-AREA	000000000 ¥v×v
✓ P アプリケーション スレッド: 1116 (一時停止)	00078000
C:¥work¥demo¥EstimOnline¥InsuEstimOnline¥New_Configuration.bin¥L	
C:¥work¥demo¥EstimOnline¥InsuEstimOnline¥New_Configuration.bin¥L	
C:¥work¥demo¥EstimOnline¥InsuEstimOnline¥New_Configuration.bin¥(> TODAY-BIN	ww
	Y
> 🕐 アプリケーション スレッド: 3060 (一時停止)	>
> 🧬 アプリケーション スレッド: 1044 (一時停止) 00078000	<u>^</u>
> 🔗 アプリケーション スレッド: 6080 (一時停止) 16進:	
> 🕐 アプリケーション スレッド: 6000 (一時停止) 00700	
> 🧬 アプリケーション スレッド: 5844 (一時停止) 0080 F	
- 🧑 アプリケーション スレッド: 4704 (一時停止)	Y
	2
🖸 UDT0033Rcbi 🕱 🖻 CUISCRN.cbi 🔯 CUISCRN.cbi	🗄 ፖታኑንብን 🛛 👘 🛃 🗢 🗖 🗖
☑ UDT0033Rcbi ►	✓ ② UDT0033P
·····* # 1 · 1 · • ··· · 2 ···· · · · · · 3 ··· · · · · 4 ··· · · ·	Environment Division
018400**	> To Data Division
©018500 RENEWAL-DISCOUNT-RTN SECTION.	V 💊 Procedure Division
© 018600 RENEWAL-DISCOUNT-000.	V MAIN-SYURI-RIN
018700*	MAIN-SYURI-000
018800 EVALUATE CONTRACT-YRS	MAIN-SYURI-999
018900 ₩₩₽₩ 0	SHIYUU-MUKUTEKI-RIN
019000 MOVE ZERO TO WK-RENEWAL-DISCOUNT-RATE	REINEWAL-DISCOUNT-RIN E DENEMAL DISCOUNT 200
019100 WHEN 1	RENEWAL-DISCOUNT-000 DEMONIAL DISCOUNT-000
019200 MOVE RENEWAL-INIT TO WK-RENEWAL-DISCOUNT-RATE	RENEWAL-DISCOUNT-999
019300 WHEN OTHER	> KEIYAKU-KYORI-KUBUN-RIN
019400 COMPUTE WK-RENEWAL-DISCOUNT-RATE =	>
UISSUU RENEWAL-INIT + (CONTRACT-YRS - 1) * RENEWAL-YRLY	MENKYOSHOU-IRO-RTN
U19600 END-EVALUATE.	A LINITENICUA UANI LOTNI
OTOTOO ADDINE HOVEN DVOUVIN -	> UNTENDRA-RAN-FRIM

デバッグが完了したアプリケーションは、アプリケーションサーバ等との連携機能を利用して自動 配備するか、ファイルを手動でコピーして、本番環境に配置します。

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、Visual COBOL for Eclipse を使って、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作成してみましょう。

1 Visual COBOL for Eclipse を起動します。

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

2 ワークスペースの保存先を選択します。

ワークスペースを保存する任意のフォルダを指定します。[参照(B)] ボタンをクリックしエクスプ ローラー経由で選択することも可能です。

🧰 Eclipse Launch	er		×
Select a direc	tory as workspace		
Eclipse uses the	workspace directory to store its preferences and development artifacts.		
ワークスペース(<u>W</u>):	C:¥work¥tutorial¥Eclipse	[参照(<u>B</u>)
□ ろの選択をデコォ	川よど」で使用し、会後での客間を表示したい(川)		
Recent Worksp			
, noconcironaja			
	OK		キャンセル

3 COBOL プロジェクトを作成します。

ファイル(F)メニューから 新規(N) → COBOL プロジェクト を選択します。



プロジェクト名欄に ConsoleHello と入力し [終了(F)] ボタンをクリックします。

🔤 COBOL プロジェクトの新規作成 - 🛛 🗙
COBOL プロジェクト
ワークスペースまたは外部の場所にCOBOL プロジェクトを作成します。
プロジェクト名(P): ConsoleHello
プロジェクト テンプレートを選択
😕 Micro Focus テンプレート
テンプレートの設定を構成
□ テンプレートの参照
場所;
ファイルシステムを選択: default ~
☑ デフォルト・ロケーションの使用(型)
ロケーション(L): C:¥work¥tutorial¥Eclipse¥ConsoleHello 参照(R)
ファイル・システムを選択(Y): <mark>デフォルト</mark> 〜
② 終了(E) キャンセル

4 COBOL プログラムを追加します。

COBOL エクスプローラービューにて **プロジェクト** フォルダを右クリックし

新規作成(N) → COBOL プログラムを選択します。

ファイル名はデフォルトの Program1.cbl を使用します。[終了(F)] ボタンをクリックします。

🔤 COBOL プログラムの新規作成 - ロ	×
COBOL プログラム エディタで開くことができる COBOL プログラムを新規作成します。	
含まれるプロジェクト: ConsoleHello 参 新規ファイル名: Program1.cbl	照
テンプレートを選択: ■ Micro Focus テンプレート	
<u>テンプレートの参照</u>	<u>i.b<u>x</u></u>
場所: ファイルシステムを選択: default ~	
② 終了(E) キャンセル	ŀ



5 エディターで COBOL ソースコードを入力します。

COBOL プロジェクトにて COBOL プログラムを新規に作成するとプログラムを構成する見出し部 (identification division)、環境部(environment division)、データ部(data division)、手続き部 (procedure division) や program-id 段落が埋め込まれたかたちで生成されます。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3行目の goback 文の手前に以下のように display 文を加えます。

なお、COBOL 正書法ではエディタービュー左右にある線で区切られた領域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。





6 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、エディタービュー上の Program1.cbl をアクテ ィブにしたまた**ファイル(F)** メニューの**保管(S)** 或いは **Ctrl + S** キーで保存します。これにより自 動的にコンパイラがキックされビルド処理が始まります。コンソールビューに正常にビルドできた旨の メッセージが出力されていることを確認します。

📃 א-עעב 🗐	🖹 問題 鹵 タスク 🔲 プロパティー 🏦 Table Results 🔐 Filter Definitions ピ コード カバレッジ 💽 Micro Focus Unit Testing
Micro Focus 빈다	: ConsoleHello
[cobollink]	cb11ds00001718.obj
[cobollink]	Creating library ConsoleHello.lib and object ConsoleHello.exp
[cobollink]	Microsoft (R) Manifest Tool version 6.2.9200.20789
[cobollink]	Copyright (c) Microsoft Corporation 2012.
[cobollink]	All rights reserved.
[cobollink]	Link complete with no errors
[cobollink]	
cobol.cfg.N	≥w_Configuration:
nature.spec	ific.build.cfg.New_Configuration:
post.build.	cfg.New_Configuration:
cobolbuild.	cfg.New_Configuration:
BUILD SUCCE Build finis	3SFUL hed with no errors.
Total time:	1 second

メモ:

Visual COBOL for Eclipse ではデフォルトでこのような自動ビルド機能が有効となっています。自動ビルド機能を無効にし、任意のタイミングでビルドしたい場合は

プロジェクトメニュー → 自動的にビルド

についたチェックを外します。ビルドする際はプロジェクトメニュー配下の「すべてビルド」或いは 「プロジェクトをビルド」を目的に応じて選択します。



7 COBOL アプリケーションをデバッグ実行します。

COBOL エクスプローラー上の Program1.cbl を右クリックし

デバッグ(D) → COBOL アプリケーション

を選択します。

パースペクティブ切り替えに関する確認メッセージには [はい(Y)] を選択します。



デバッグパースペクティブに切り替わりましたら、DISPLAY 文を実行する手前でステップが一時停止し ていますので、[**ステップイン**] アイコンを一回クリックし DISPLAY 文を実行します。

💷 Eclipse - デパック - ConsoleHello/Program1.cbl - Eclipse		- 0	×
ファイル(F) 編集(E) ソース リファクタリング ナビゲート(N) 検索 プロジェクト(P) 実行(R) ウ	ィンドウ(W) ヘルプ(H)		
i 🗂 • 🔚 🐚 i 🚸 • 🔘 • 💁 • i 🥭 🗀 🔗 • i 🖳 🔌 🗈 💷 🕺 😎 🧒	. e 🗐 🕾 😎 🍠 🔲 🌄 🖓 💷 🖬 🗔 ½ 👻	🖓 🔹 🏷 🔶	• 🗘 •
ステッ	ブイン(I)(F5) クイック・フ	<u>"クセス</u> 🛛 🖻 🛛	al 🔅
🏘 デパッグ 🕄 🚜 Servers 🍡 📚 🍷 🗖 🗖	(X)= 変数 🕴 🎭 ブレークポイント 😚 式) 📲 📄 🔻 (
 ConsoleHello.exe [ConsoleHello] [COBOL アプリケーション] 	名前 注	値	
■ Program1: 請または設置がなし(行:10) > 00 アブリケーションスレッド、1396(一時停止) > 00 アブリケーションスレッド、500(一時停止)			
) The 1999 (1997) 1999 (1999)	<		>
			\sim
			\sim
< >			>
🖻 Program1.cbl 🕱	🖓 🗗 📴 📴 🖓	în ↓ <mark>a</mark> ∼	
Program1 chi	✓		
······································	···6·····7··I·•···8	ent Division	
⊖ data division.	> Data Divi	sion Division	
working-storage section.	Vo Procedur	e Division	
⇒ procedure division. ⇒ ⊖ DISPLAY "Hello Morid".			
goback.			
end program Program1.			
	> *		



Animator application 画面に「Hello World」が表示されたことを確認できましたら、[ステップイン] アイコンを再度クリックしデバッグを終了します。





第4章 COBOL JVM アプリケーションの作成

本章では、Visual COBOL for Eclipse の COBOL JVM 機能を利用して Java バイトコードにコン パイルされた COBOL アプリケーションを作成する方法を紹介します。ここでは、Java プログラム から「Hello World」を表示する COBOL プログラム呼び出すアプリケーションを作成します。

1 Visual COBOL for Eclipse を起動します。

Windows のスタートメニューから、**Visual COBOL for Eclipse** をクリックし、第3章で使用したワークスペースを選択します。第3章から続けて実施される場合は、このステップはスキップしワークスペースを継続して使用します。

2 COBOL JVM プロジェクトを作成します。

ファイル(F)メニューから 新規(N) → COBOL JVM プロジェクト を選択します。

プロジェクト名欄に **ConsoleHelloJVM** と入力し [**終了(F)**] ボタンをクリックしま す。

💷 COBOL JVM プロジェクトの新規作成		_		Х
COBOL JVM プロジェクト ワークスペースまたは外部の場所に COBOL JVM プロジェク	トを作成します。			
プロジェクト名(P): ConsoleHelloJVM				
✓ デフォルト・ロケーションの使用(D) ロケーションの使用(C) ロケーション(1): ○ ゲッ・ork ¥tutorial¥Eclinse¥ConsoleH	elloIVM	_	恭昭(R)	
			\$2.74 (<u>10</u>	
フロジェクト テンフレートを選択				
		=5.71.	しの砂束を増	: ett
□ テンプレートの参照		5220-	110月12月11日12月13日	10 <u>%</u>
場所:			参照	
ファイルシステムを選択: default 🗸				
JRE				
● 実行環境 JRE の使用(<u>V</u>):	JavaSE-1.8			~
〇 プロジェクト固有の JRE を使用(<u>S</u>):	jre1.8.0_102			\sim
〇 デフォルト JRE の使用(<u>A</u>) (現在は 'jre1.8.0_102')			<u>JRE を構め</u>	<u>t</u>
ワーキング・セット				
□ ワーキング・セットにプロジェクトを追加(D)			新規(<u>₩</u>)	
ワーキング・セット(<u>ロ</u>):		\sim	選択(<u>E</u>)	
(P) < 戻る(B) 次へ((N) > 🚺 🕷	を了(F)	キャンセ	JL



3 Native 用のプロジェクトで使用したプログラムをコピーします。

COBOL エクスプローラービューにて **ConsoleHello** プロジェクト配下の COBOL プログラムフ ォルダに前章で追加した Program1.cbl を右クリックし **コピー** を選択します。

回 Eclipse - COBOL - ファイル(F) 編集(E)	Consol ソース	eHello/Progran リファクタリング	n1.cbl - Eclip ナビゲート(N)	ise i 検索	プロジェクト(P)) 実行(F	 ウィンド
* 🗊 🖛 🔁	- 0	- 💁 - 😕	🗀 🔗 🔻	E ⊿	/ 🔲 🍪 👎		Π 1=
🔓 COBOL I 🔀	ጜ ታビ	ゲーター 😤 Ap	plicati 📒	サーバー	I 🗆 🗖	3	Program1
✓ 2 ConsoleHello ✓ 2 COBOL 7	。 プログラム			E 4	₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽	-	Pro <u>c</u>
> 🙋 Progr	ran 1. hfig	新規作成(N)					>
Cons Cons Progr Progr	ole ′an ′an	開く(O) アプリケーション:	から開く				>
Prog	rar 🗈	วษ์-					Ctrl+C
> 🖉 ConsoleHello		貼り付け					Ctrl+V
	×	削除(D)					削除
	<u>_</u>	Remove from	Context		Ctrl→	Alt+シフト	+下へ
		移動(V)					

4 COBOL エクスプローラーのフィルターをカスタマイズします。

COBOL エクスプローラービュー中の右上のアイコンをクリックし、ビューのカスタマイズ を選択 します。

🔓 COBOL I 🔀	🕾 ታビゲーター	😤 Applicati	🧾 サ−/(+ I	77 🗅	j	🖻 Program1.cbl 🔀	
			🖻 🔄 🛱	, ⊕ _{\$} (_ ≂	z j	Program1.cbl	
🗸 🐸 ConsoleHell	0					トップレベル要素(T)	>
🗸 🔁 COBOL 🤅	プログラム						
> 🖸 Prog	ram1.cbl					ワーキング・セットの選択(W)	
🗸 🗁 New_Cor	nfiguration.bin					ワーキング・セットの選択解除(K)	
🔲 Cons	oleHello.exe					アクティブなワーキング・セットの編集(F)	
🗖 Prog	ram1.idy					277712-072 1277 E010200056(E/m	
📄 Prog	ram1.obj					ビューのカスタマイズ(C)	
📄 Prog	ram1.obj.1.tlog			4	2	エディタにリンク(L)	
🔉 🐸 ConsoleHell	MVlo				ν.		



カテゴリ外の空のフォルダのチェックを外し、OK ボタンをクリックします。

💷 使用可能なカスタマイズ	×
 ▶ 2/ルター № コンテンツ 適用するフ/ルターを選択してください(一致する項目は隠されます): フ/ルターの名前を入力 ✓ パリソース ✓ COBOL カテゴリ内の空のフォルダ □ COBOL コピーファイル プロジェクト 図 RSE Internal Projects □ カテゴリ外の空のフォルダ ✓ M均都 Micro Focus プロジェクト ○ 内部 TD プロジェクト ○ 内部 TD プロジェクト ○ 離承 COBOL ブログラム □ 閉じたプロジェクト □ 非 Micro Focus プロジェクト 	Program1.obj Program1.obj

COBOL for JVM プロジェクト配下に src フォルダが表示されます。

5 COBOL for JVM のプロジェクトにコピーしたプログラムを追加します。

COBOL for JVM のプロジェクト配下にある src フォルダを右クリックし

貼り付けを選択します。





6 COBOL から生成された JVM クラスをパッケージ化します。

COBOL エクスプローラービューにて COBOL for JVM のプロジェクトを右クリックし、[プロパテ イ(R)] を選択します。

[Micro Focus] > [ビルド構成]

へとナビゲートし展開される画面中の [追加指令] 欄に

ILNAMESPACE(com.vc30.tutorial) を記入し、[**OK**] ボタンをクリックします。

💷 プロパティ: ConsoleHelloJVM		– D X
Эн Јияда	ビルド構成	← ▼ ⇒ ▼
> リソース ✓ Micro Focus JVM ビルドパス > SQL プリプロセッサ コピーファイル	出力ディレクトリ名: bin	▲ ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●
ビルド環境 ビルド優先順位 Project Facets Task Repository Task Tags Validation	文字集合: 言語方言: ソースフォーマット:	ASCII ~ Micro Focus ~ 可変 ~
Wiki lext ビレダー プロジェクト参照 実行/デパッグ設定	 ✓ デパッグ用にコンパイル(D) □ 出力の表示 □ 動的呼び出しを使用 	□ 指令ファイルの生成 □ リストファイルを生成 □ インクリメンタル ビルドの使用(テクニカル プレビュー機ジ)
	警告レベル: 最大エラー数:	回復可能なIIラ-を含める(レベル E) ~ 100
	追加損令: ILNAMESPACE(com.vc30.tutorial)	
?	1	OK キャンセル



7 COBOL JVM Class ファイルをコンパイルします。

自動的にビルドが有効になっていれば、前ステップでプログラムを追加した時点でビルド処理が走ります。

コンソールビューで正常に処理できた旨のメッセージを確認できます。

🖳 コンソール 🙁 🔝 問題 🧔 タスク 🔲 プロパティー 🏫 Table Results 🔐 Filter Definitions 🔡 コード カバレッジ 💽 Micro Focus Unit Testing
Micro Focus 신나는 ConsoleHelloJVM
cobol.link.cfg.New_Configuration:
cobol.cfg.New_Configuration:
nature.specific.build.cfg.New_Configuration:
post.build.cfg.New_Configuration:
cobolbuild.cfg.New_Configuration:
BUILD SUCCESSFUL
Build finished with no errors.
Total time: 1 second

また、COBOL エクスプローラービューでは、追加したプログラムが .class の拡張子を持った JVM クラスとしてビルドされていることを確認できます。プログラム中にはパッケージ化する旨の記 述は加えていませんが先のステップで指定したコンパイラ指令により、パッケージ化された JVM クラ スとして生成されています

😤 COBOL I... 🐹 陆 ナピゲーター 😤 Applicati...





8 Java プロジェクトを作成します。

ファイル(F)メニューから 新規(N) → その他 を選択します。

Java 配下の Java プロジェクトを選択し [次へ(N)] ボタンをクリックします。

[eal 新規	_	
ウィザードを選択		
Java プロジェクトの作成		
ウィザード(<u>W)</u> :		
フィルタ入力		
✓ ➢ Java 授 Java プロジェクト		^
参 Java ワーキング・セット		
₩ ソース・フォルダ		~
(B) (N) > 終了(F)		キャンセル

プロジェクト名欄に	💷 新規 Java プロジェクト		—		×	
CallCobHello	J ava プロジェクトの 作成 Java プロジェクトをワークスペースまたは外部ロケーションに作	乍成します。		T	Jr.	
と入力し [終了(F)] ボタンをクリ						
ックします。	プロジェクト名(P): CallCobHello					
	☑ デフォルト・ロケーションの使用(型)					
	ロケーション(<u>L</u>): C:¥work¥tutorial¥Eclipse¥CallCobHello 参照(®					
	JRE					
	● 実行環境 JRE の使用(<u>V</u>):	JavaSE-1.8			~	
	〇 プロジェクト固有の JRE を使用(<u>S</u>):	jre1.8.0_102			\sim	
	○ デフォルト JRE の使用(<u>A</u>) (現在は 'jre1.8.0_102')		Ţ	<u>RE を構成</u>	<u>t</u>	
	プロジェクト・レイアウト					
	○ プロジェクト・フォルダをソースおよびクラス・ファイルのJ	レートとして使用(<u>U</u>)				
	● ソースおよびクラス・ファイルのフォルターを個別に作成	.(<u>0</u>)	<u>訊定</u>	目値を構成	<u>t</u>	
	ワーキング・セット					
	□ ワーキング・セットにプロジェクトを追加(D)		新	規(<u>₩</u>)		
	ワーキング・セット(<u>0</u>):		~ 選	択(E)		
	? < 戻る(B) 次へ	.(N) > 終了(E		キャンセ	JI.	



Java パースペクティブに関連付けられる旨のメッセージには [はい(Y)] を選択します。



9 Java クラスファイルを追加します。

パッケージエクスプローラービューにて CallCobHello プロジェクト配下の src フォルダを右ク リックし

新規(W) → クラスを選択します。

新規 Java クラスウィンドウでは

パッケージ欄 com.vc30.tutorial

名前欄 CallCobHello

のように入力します。また、「public static void main(String[] args)」にチェックを入れ、[終了

ソース・フォルダ(<u>D</u>):	CallCobHello/src	参照([
/(ッケージ(<u>K</u>):	comvc30.tutorial	参照()
III エンクロージング型(Y):		参照()
名前(<u>M</u>):	CallCobHello	
修飾子:	● public(P) ○デフォルト(U) ○ private(У) ○ protected(I) □ abstract(I) □ final(L) □ static(O	
スーパークラス(<u>S</u>):	java.lang.Object	参照(
インターフェース(<u>)</u> :		追加(;
		¤∧ +
作成するメソッド・スタブの	選択	
	public static void main(String[] args)	
	□ スーパークラスからのコフストラクター(G) ▽ 継承された抽象メソッド(H)	
	テンプレートの構成およびデフォルト値についてはここを参照)	
- コメントを追加しますかくし		

(F)] ボタンをクリックします。



10 COBOL JVM プロジェクト及び Visual COBOL のランタイムをビルドパスに追加 します。

パッケージエクスプローラービューにて CallCobHello プロジェクトを右クリックし [プロパティ (R)] を選択します。

[Java のビルド・パス]ページの [ライブラリー] タブを選択します。 [クラスフォルダの追加(C)] ボタンをクリックします。

ConsoleHelloJVM プロジェクト配下の [bin] フォルダにチェックを入れ、[OK] ボタンをクリックします。

💷 クラス・フォルダの選択 -	- [- X
ビルド・パスに追加するクラス・フォルダを選択してください(_):		
CallCobHello		
> > .sectings		
✓ ☐		
> 🗌 🗁 New_Configuration.bin		
✓ □ ☑ ConsoleHelloJVM		
settings		
> C > src		
✓ □		
> 🗌 🗁 Visual_COBOL		
RemoteSystems lempFiles		
新規フォルダの作成(<u>N</u>)		
ОК	ŧ	マンセル
		17 277



続いて、[**ライブラリーを追加(i)**] ボタンをクリックします。

[COBOL JVM 実行時システム] を選択し [次へ(N)] ボタンをクリックします。

💷 ライブラリーの追加	—		Х
ライブラリーの 追加 追加するライプラリー・タイプを選択します。		4	3
Connectivity Driver Definition CXF Runtime EAR Libraries JRE システム・ライプラリー			
JUnit Maven Managed Dependencies Server Runtime			
Web App Libraries プラグインの依存関係 ユーザー・ライプラリー			
(B) 次へ(N) > 終了(E)		キャンセル	ŀ

[終了(F)] ボタンをクリックします。

上で追加したライブラリーが画面に反映されていることを確認し、[OK] ボタンをクリックしプロパティ画面を閉じます。





11 エディターで Java ソースコードを入力します。

main メソッドに以下のコードを追記します。

Program1 cobclass = new Program1();

long res = cobclass.Program1();



: **ד**لا⁽⁾

従来の手続き型の COBOL プログラムを JVM クラスにコンパイルすると、プログラム名がクラス名と なり、PROCEDURE DIVISION 以下で記載された命令は、プログラムと同名のインスタンスメソッドと して実装されます。生成された JVM クラスは Java から生成される JVM クラスと同様に扱えるた め、Java 中で特別なロジックを記述することなく COBOL の呼び出し命令を記述できます。Eclipse 上 の Java のエディタも COBOL から生成されたクラスのシンボルを Java と同様に認識できるため、 Java エディタ上で COBOL プログラムに対してコードアシストの機能等を利用することができます。 コードアシストは Ctrl + Space で起動できます。





12 COBOL JVM Class ファイルをコンパイルします。

スペルミスがなければ、エディタービュー上の CallCobHello.java をアクティブにしたまたファ イル(F) メニューの保管(S) 或いは Ctrl + S キーで保存します。問題ビューに何もエラーが出力さ れていないことを確認します。



13 アプリケーションをデバッグ実行します。

COBOL プログラム中のメソッド呼び出す

long res = cobclass.Program1();

にブレークポイントを設定します。





同様に ConsoleHelloJVM プロジェクト 中の Program1.cbl の DISPLAY 文にもブレ ークポイントを指定します。



CallCobHello > src > com.vc30.tutorial 配下の CallCobHello.java を右クリックし

デバッグ(D) → デバッグの構成(B)を選択します。

Java アプリケーション を ダブルクリックします。	デバッグ構成 構成の作成、管理、および実行 Java アプリケーションのデバッグ	F
	 ご ● ※ ■ ⇒ ▼ ブルリタスカ Eclipse Data Tools Eclipse アブリケーション Generic Server Generic Server(External L HTTP Preview JZEE Preview Java アブリケーション Mara アブリケーション Java アブリケーション 	 このダイアログから起動設定を構成します: ご - 選択したタイプの構成を作成するには、「新規」ボタンを押します。 ご - 選択した構成をコピーするには、「複製」ボタンを押します。 ※ - 選択した構成を削除するには、「削除」ボタンを押します。 ジ - フィルタ・オプションを構成するには、「フィルタ」ボタンを押します。 - 既存構成を選択して、それを編集または表示します。 ソ<u>パースペクティブ</u>設定画面から、起動パースペクティブの設定を構成します。

[**ソース**] タブをクリックします。

[追加(A)] ボタンをクリックします。

👜 デパック構成		×
構成の作成、管理、および Java アプリケーションのデパッグ	\$実行 	Ť
 □ ※ □ ※ ▼ >メルタスカ □ cobol ユニット デ.▲ 	名前(№): CallCobHello ③ メイン (№= 引数 (▲) JRE (№) クラスパス (トッソース) (本) 環境 (□ 共通(C) ソース・ルックアップ・パス(①):	
編 COBOL ライブ記録 ■ Eclipse Data Tool: ● Eclipse アプリケーシ ● Generic Server ■ Generic Server(Ex	> 🗁 デフォルト	追加(<u>A</u>) 編集(<u>E</u>) 除去(<u>M</u>)



[COBOL プロジェクト] オ	を選択の上、	[OK] ボタンをクリックします
------------------	--------	------------------

💼 ソースの追加	×
コンテナ ーをソ ース・ルックアップ・パスに 追加 coBoL プロジェクトパス コンテナ	+ E
<mark>ほそCOBOL プロジェクト</mark> ● Java クラスパス変数 ほみJava プロジェクト	~
?	OK キャンセル

COBOL for JVM のプロジェクト [ConsoleHelloJVM] にチェックを入 れ、[OK] ボタンをクリックします。

👜 選択してください		—		\times		
追加するプロジェクトを	追加するプロジェクトを選択:					
ConsoleHello ConsoleHelloJVM						
	すべて選択(<u>S</u>)	選択	をすべて解	除(<u>D</u>)		
?	OK		キャンセ	л.		

[**デバッグ(D)**] ボタンをクリックします。

パースペクティブ切り替えの確認メッセージには [はい(Y)]を選択します。

[]] /(-)	スペクティブ切り替えの確認 ×
\bigcirc	この種類の起動は、中断時にデパッグパースペクティブが開くように構成されています。
	このデパッグ・パースペクティブは、アプリケーション・デパッグをサポートするために設計されています。こ れには、デパッグ・スタック、変数、およびプレークポイント管理を表示するビューが組み込まれていま す。
	このパースペクティブを開きますか?
□ 常に	この設定を使用する(<u>R</u>)
	(おい(Y) いいえ(N)



ステップインアイコンをクリックします。ステップ実行には、ステップイン、ステップオーバとメ ニューが用意されています。ステップインを選択した場合は Call 先のメソッドの中までステップ実行 し、ステップオーバーを選択した場合は、メソッドの中にステップを進めずメソッドを実行します。こ こでは、ステップオーバーを選択し COBOL プログラム中に指定したブレークポイントまでステップを 進めます。

			пх
$= 2\pi/(\pi)$ (an expression of the second state			
- ファイコル(r) 編集(c) フース(s) リンアシタリン?(i) J c) = r(n) 検索 / U y i 2(r(n) : 🐟 - □ 応: 表 - ○ - ○ - ○ - ◇ - : 回: > □ □ □ !! ? □ □ !! ?		**5 / • / •	
	テップ・オーバー(O)(F6)	<u>9199+7912X</u>	COLI 🎄 🖏
🏘 テ/(ワック 🛱 Servers 🍇 📚 🗢 🗖	(x)= 変数 😂 🎭 プレークポイント 🚳 式	🆾 📲 📄	~
✓ □ CallCobHello [Java アプリケーション]	名前		値
✓ 2? ローカル・ホスト上の com.vc30.tutorial.CallCobHello:50415	I args		String[0] (ID=
	> O cobclass		Program1 (ID:
C:¥Program Files¥Java¥jre1.8.0_102¥bin¥javaw.exe(2017/07/19 7:26:49)			
	<		>
			~
	<		>
🖻 Program1.cbl 🛛 DallCobHello.java 🔀		アウトライン 🛛	- 0
<pre>package com.vc30.tutorial;</pre>	^ <u>=</u>	🗊 🖻 📮 🏹 🗙	● 🗙 🗢
which class CallCobHello /		🖶 com.vc30.tutorial	
parite class callosmerie (~	CallCobHello	unid
public static void main(String[] args) {		👦 man(sonigu) : v	/oru
Program1 cobclass = new Program1();			
ing res - concrass.Flogram();			
)			
}	~		
<	>		

COBOL アプリケーションのソースの中にステップが進んでいることが確認できます。

参デバック XX 嶋 Servers	% ₽ ▽	(x)= 変数 🕴 💊 プレークポイント 🤞	ў́ д	ti ⇒ti E		- 0
 		名前			値	
Program1.cbl (7: 10 CallCobHello main(String[1) (7: 7)						
📙 C:¥Program Files¥Java¥jre1.8.0_102¥bin¥javaw.exe(20	017/07/19 7:32:38)	<				>
						\$
		<				>
💭 CallCobHello,java 💿 Program1.cbl 🛛 🗁 📴 איז איז איז 💿 🔁 איז				די גאליא 🕄 📲		
Program1.cbl Program1.cbl						
The second					ion	
data division.				n		
DIVINY "He lo Horld"						
goback.						
end program Program1.			~			
<			>			



再度、ステップインアイコンをクリックし、DISPLAY 文を実行するとコンソールに「Hello World」が表示されることが確認できます。

D	CallCobH	ello.java 😰 Program1.cbl 🔀		
	🖻 Pro	gram1.cbl		
	• • • •	• ·*à·1·B··•···2····•3····•4····•5····•6····•7··I·•···8		
	Θ	program-id. Program1 as "Program1".		
	Θ	environment division.		
		configuration section.		
	Θ	data division.		
		working-storage section.		
	Θ	procedure division.		
,2	Θ	DISPLAY "Hello World".		
\$		goback.		
		end program Program1.		
	<			
	ال ا			
Call	<u>_CallCobHello [Java</u> アプリケーション] C:¥Program Files¥Java¥jre1.8.0_102¥bin¥javaw.exe(2017/07/19 7:34:59)			
He	Hello World			

確認できましたら、アプリケーションが終了するまでステップを進めデバッグを終了させます。



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 Eclipse で作成しましょう。

1 Visual COBOL for Eclipse を起動します。

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

ワークスペースは第3章で用意したものをそのままご使用いただいても構いません。

2 COBOL プロジェクトを作成します。

ファイル(F)メニューから 新規(N) → COBOL プロジェクト を選択します。

: דע 🕅

ファイル(F) メニュー → 新規(N) にて候補として表示されるプロジェクトはワークスペース中でアク ティブなパースペクティブにより切り替わります。例えば、Visual COBOL が提供するパースペクティ ブがアクティブな場合、COBOL プロジェクトは候補として表示されます。一方、他の Java のパース ペクティブ等がアクティブな場合は候補に含まれないため、パースペクティブを COBOL に切り替えて から作業します。

プロジェクト名欄に LoadCSV と 入力し[終了(F)] ボタンをクリックしま す。

COBOL プロジェクトの新規作成 -	- 🗆	×
COBOL プロジェクト	-	-
ワークスペースまたは外部の場所にCOBOL プロジェクトを作成します。	\mathbf{k}	-
プロジェクト名(P)。LoadCSM		
プロジェクト テンプレートを選択		
ಚ Micro Focus テンプレート		
	/-トの設定を根	<u>載成</u>
□ テンプレートの参照		
場所:	参照	
ファイルシステムを選択: default ~		
☑ デフォルト・ロケーションの使用(型)		
ロケーション(L): C:¥work¥tutorial¥Eclipse¥LoadCSV	参照(<u>R</u>)	
ファイル・システムを選択(Y): デフォルト 🗸		
?)	?JV



3 COBOL プログラムを追加します。

COBOL エクスプローラービューにて LoadCSV プロジェクト フォルダを右クリックし 新規(N) → COBOL プログラムを選択します。

新規ファイル名欄には	Image: COBOL プログラムの新規作成 - - ×
LoadCSV.cbl を指定します。	COBOL プログラム エディタで開くことができる COBOL プログラムを新規作成します。
[終了(F)] ボタンをクリックし	
ます。	含まれるプロジェクト: LoadCSV 参照 新規ファイル名: LoadCSVicbl テンプレートを選択: Micro Focus テンプレート
	<u>テンプレートの設定を構成…</u> □ テンプレートの参照 場所:
	? 終了(E) キャンセル

4 コードエディターで COBOL ソースコードを入力します。

COBOL ソースファイルを新規に作成した直後の段階ではコンソールアプリケーションのひな形が埋め込まれています。ここでは、環境部(environment division)、データ部(data division)、手続き部 (procedure division) を書き換えます。

oadCSV 🖸	cbl 🛛
🖻 Lo:	adCSV.cbl
	• * A · 1 · B · • • · · · · 2 · · · • • · · · 3 · · · • • · · · 4 · · · • • · · · 5 · · · • • · · · 6 · · · • • · · · 7 · · I · • · · · 8
Θ	program-id. LoadCSV as "LoadCSV".
Θ	environment division. configuration section.
Θ	data division. working-storage section.
Θ	procedure division.
	goback.
	end program LoadCSV.



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".

🛃 LoadCSV.cbl 🔀 🛃 LoadCSV.cbl •···•**A·1·B··•···2···•*···3····•4····•5····•6····•6····•7··I·•···8 Θ program-id. LoadCSV as "LoadCSV". Θ environment division. Θ INPUT-OUTPUT SECTION. FILE-CONTROL. SELECT IN-FILE ASSIGN TO "Emp_Master.csv" LINE SEQUENTIAL. SELECT OUT-FILE ASSIGN TO "Emp Master.dat". L Θ data division. working-storage section. Θ procedure division. Θ goback. end program LoadCSV.



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	Х (50).
FD	OUT ·	-FILE.		
01	OUT-	-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.

🖻 LoadCSV.cbl	3	
LoadCS	cbl	
	·1·B·······2······3······4·····5······6···	•
Θ	rogram-id. LoadCSV as "LoadCSV".	
8	NVIFONMENT GIVISION. NDUT_OUTDUT SECTION	
Ŭ	TIE-CONTROL	
	SELECT IN-FILE ASSIGN TO "Emp Master.csv"	
	LINE SEQUENTIAL.	
	SELECT OUT-FILE ASSIGN TO "Emp_Master.dat".	
Θ	ata division.	
Θ	TILE SECTION.	
	D IN-FILE.	
	1 $\frac{IN-REC}{IN-REC}$ PIC X(50).	
	D OUT-FILE.	
	1 OUT-REC.	
	05 OUT-EMPNO PIC 9(8).	
	O5 FILLER PIC X.	
	O5 OUT-JNAME1 PIC N(5).	
	O5 OUT-JNAME2 PIC N(5).	
	O5 OUT-NAME1 PIC X(5).	
	OS OUT-NAMEZ PIC X(5).	
	OS OUT-GENDER PIC X.	
	O5 FILLER PIC X.	
	$05 OUT-DIV \qquad PIC N(5).$	
	OS OUT-EMPDATE PIC 9(8).	
	O5 FILLER PIC X.	
e	rocedure division.	



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.





5 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、エディタービュー上の LoadCSV.cbl をアクティ ブにしたまたファイル(F) メニューの保管(S) 或いは Ctrl + S キーで保存します。コンソールビュ ーに正常にビルドできた旨のメッセージが出力されていることを確認します。



6 CSV ファイルを作成します。

LoadCSV プロジェクト配下の New_Configuration.bin フォルダを右クリックし

新規(N) → ファイルを選択します。

ファイル名欄に Emp_Master.csv を指定し [終了(F)] ボタンをクリック します。

親フォルダを入力または選択(<u>E</u>):		
LoadCSV/New_Configuration.bin		
> 😂 CallCobHello		
> 🚑 ConsoleHello		
> ConsoleHelloJVM		
Internal IDProject		
LoadCSV		
RemoteSystemsTempFiles		
ファイル名(<u>M</u>): Emp_Master.csv		
1177		
拡張 >>(<u>A</u>)		


「この種類のファイル(.csv)を開くには、どのアプリを使いますか?」の問いがポップアップされた 場合は「メモ帳」を選択します。



以下のデータを入力し、ファイルを保存の上、閉じます。

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 33333339,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,エツコ,F,総務部,20040805,0 99999997,三井,薫,ミツイ,カオル,F,営業部,20050401,0



7 COBOL アプリケーションをデバッグ実行します。

New_Configuration.bin フォルダ配下の LoadCSV.exe を右クリックし

デバッグ(D) → COBOL アプリケーションを選択します。

パースペクティブ切り替えの確認メッセージには [はい(Y)]を選択します。



デバッガーは手続き部の最初の COBOL 文である open 文の処理前に一時停止している状態となっています。

👜 Eclipse - デパッグ - LoadCSV/LoadCSV.cbl - Eclipse	- 1	o x
ファイル(E) 編集(E) ソース リファクタリング ナビゲート(N) 検索 プロジェクト(P) 実行(B) ウイ	ンドウ(W) ヘルプ(出)	
i 🗂 🕶 🖫 🔞 i 🖳 i 🔌 💷 🔲 🛤 🖧 👁 . 🖉 🖃 🖓 🔛 📢 👎	🛂 📴 🔳 🔣 👯 🕶 🔿 🕶 💁 🗁 🔗 🕶 🦢 🖉	<p -="" -<="" td="" ⇒=""></p>
		🔤 🔆 💐
🏘 デパッグ 🕄 🚜 Servers 🎆 📚 🍷 🗖 🗖	(*)= 変数 🙁 💁 ブレークボイント 😚 式 🌇 🐴 🖻	~
 ✓ こ LoadCSV.exe [LoadCSV] [COBOL アブリケーション] ✓ 記 Micro Focus デパ(ツガ: C:¥work¥tutorial¥Eclipse¥LoadCSV¥New_Configuration.bin¥L ✓ ② アブリケーション スレッド: 5952 (一時停止) ご LoadCSV: PROC1 (行: 29) > ③ アブリケーション スレッド: 5060 (一時停止) > ③ アブリケーション スレッド: 404 (一時停止) > ③ アブリケーション スレッド: 6088 (一時停止) > ③ アブリケーション スレッド: 6088 (一時停止) 	各前 化 ◆ IN-FILE C	t Josed Last sta >
<	<	>
e LoadCSV.cbl ≅	😑 📴 ምዕኑንብን 🛙 🝃 🖧	▽ □ □
C LoadCSV.cbl C LoadCSV.cbl O5 FILLER PIC X. procedure division. PROC1. PROC1. PROC2. READ IN-FILE AT END GO TO PROC9. INITIALIZE OUT-REC. UNSTRING IN-REC DELIMITED BY "," INTO OUT-EMENO OUT-JNAME1 OUT-JNAME2	 ✓ ② LoadCSV > ③ Environment Division ◇ ③ Pile Section ◇ Procedure Division ○ PROC2 ○ PROC3 	



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、IN-REC に格納される値の変遷を追います。UNSTRING 文の IN-REC を選択します。

PROC2.
READ IN-FILE AT END GO TO PROC9.
INITIALIZE OUT-REC.
UNSTRING IN-REC DELIMITED BY ","
INTO OUT-E File Section, PIC X(50),参照= 1, サイズ= 50 /(イト
OUT-JIN-REC=
OUT-JNAME2

この状態で右クリックし、「項目を検査」を選択します。

「式ビューに追加」をクリックします。

Q IN-REC=
A
<

同様に出力ファイルに書き出すレコードの内容もトレースするため、INITIALIZE 文の **OUT-REC** を同様に式ビューに追加します。

(x)= 変数 💁 プレークポイント 👧 式 🔀	🆾 🍕 🕞 💠 🗶 💥 🗢 🖻
名前	値
> ^{x+y} "IN-REC"	
> ^X [†] ^y "OUT-REC"	
🐈 新しい式を追加	
	Q.
<	>



F5 キー(ステップイン)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。 式ビューの 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。

🔤 Eclipse - デパッグ - LoadCSV/LoadCSV.cbl - Eclipse	-	D X
ファイル(F) 編集(E) ソース リファクタリング ナビゲート(N) 検索 プロジェクト(P) 実行(R) ウインドウ(W)	ヘルプ(日)	
i 🗂 🕶 🔛 🕼 i 🖳 i 🔌 🕪 💷 🛋 🛠 🎿 👁 🗽 🕂 🐼 😓 🐼 🛃 🛃 💋 🔝	▣ 🗊 📴 🛊 ד 🔾 ד 🂁 ד 🤔 🖒 🖋 ד 🦢 ד 🏹 ד 🏷 ד 🔿 ד	
	クイック・アクセス	8 🔤 🎋 🖏
🎄 デパッグ 🐹 🦓 Servers 🍇 💱 マ 🖓 👔	🗆 🗱 🗣 ブレークポイント 💱 式 🛛 🌋 🆾 🦡 🕞 🛛 🕂	🗞 🗸 🗖 🗖
 ✓ ■ LoadCSV.exe [LoadCSV] [COBOL アブリケーション] ✓ ♪ Micro Focus デパッガ: C:Work Wutbrial WEclipse¥LoadCSVWNew_Configuration.bin¥LoadCSV.ex ✓ ♪ ブリケーション フルド 5952 (一時停止) ■ LoadCSV: PROC2 (行: 34) > ♪ アブリケーション スレッド: 5060 (一時停止) > ♪ アブリケーション スレッド: 5086 (一時停止) > ♪ アブリケーション スレッド: 5086 (一時停止) 	 名前 4 ※ ²/³/³ "IN-REC" 1111113,佐葉隆,均り,5%,M,宮業範,199804 ※ IN-REC 1111113,佐葉隆,均り,5%,M,宮業範,199804 ※ ³/³/³ "OUT-REC" ● 新しい式を追加 	
	11111113, 佐藤,隆, ,, , , , , , ,	~
<) (>
loadCSV.cbl 🔀	□ 📴 アウトライン 🛛 😜 .	a v
LoadCSV.cbl	V 🕑 LoadCSV	
·····*à·1·B·····2·····3·····4·····5·····5·····6····	Environment Division Dete Division	
<pre> procedure division. PROC1. OPEN INPUT IN-FILE. OPEN OUTPUT OUT-FILE.</pre>	Constant and a provision Source and a provision Source and a provision PROC1 PROC2 PROC2	
PROC2.	PROL9	
INITIALIZE OUT-REC.		
UNSTRING IN-REC DELIMITED BY ","	~	
<	>	

さらに **F5**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。 式ビューの 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。

参デパッグ ☆ 👭 Servers 🍇 💱 ▽ 🖻 🗖	(X)= 変数 💊 プレークポイント 😪 :	त 🛙	🏝 📲 📄 💠	× ¾ ▽ □ □
 LoadCSV.exe [LoadCSV] [COBOL アブリケーション]	名前 ◆ IN-REC" ◆ IN-REC ◆ OUT-REC ◆ OUT-REC ● OUT-BUT ● ILLER ● OUT-INAME1 ● OUT-NAME1 ● OUT-NAME1 ● OUT-NAME1 ● OUT-NAME1 ● OUT-NAME1 ● OUT-NAME1 ● OUT-NAME2 ● OUT-NAME2 ● OUT-RENDER ● FILLER ● OUT-EMPDATE ● FILLER	値 1111113, 佐藤, 月 1111113, 佐藤, 1111113, 佐藤 1111113, 佐藤 1111113 佐藤 隆 9月 9月 9月 9月 9月 9月 9月 9月 9月 9月 9月 9月 9月	â. y 1-9 3 Ar y M. 宮栄郡, 199004 â. y 1-9 3 Ar y M. 宮栄郡, 199004 隆 1913 3 Ar y J 隆 1913 3 Ar y J 隆 1919 3 Ar y J 隆 1919 3 Ar y J	
< >	<			>
loadCSV.cbl 🔀		- 8	🔠 アウトライン 🛙	Ş⇒↓ <mark>a</mark> ⊽ ⊓ □
LoadCSV.cbl OUT-EMPDATE END-UNSTRING. WRITE OUT-REC. G0 T0 PROC2. <	···7··I·•···8	• •	✓ ✓ ✓ LoadCSV SV Environment Division ✓ ✓ Data Division ✓ ♥ File Section ♥ Procedure Division ● PROC1 ● PROC2 ● PROC9	ion n

さらに F5 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。



F8(再開)キーを打鍵するか CSV ファイルからすべてのレコードを読み込むまで **F5**キーを押すと、 デバッガは終了します。





ワークスペース右上コーナにあるパースペクティブ切り替えボタンエリアにて「COBOL」を選択

し、COBOL パースペクティブに戻ります。



COBOL エクスプローラービューにて LoadCSV プロジェクト配下の New_Configuration.bin フォルダを右クリックし「**更新(F)**」を選択します。

Emp_Master.dat ファイルが作成されて いることを確認します。



Emp_Master.dat を右クリックし、

アプリケーションから開く → クラシックデータファイルツールを選択します。Visual COBOL に

付属する データファイルエディタが起 動します。[ファイル編成] 欄はデフォ ルトの「レコード順 – 固定長」のまま にしておきます。[最大の長さ] 欄には 「60」を指定します。

F -9771# IF 19	?	×		
ファイル名 C:¥work¥tutorial¥Eclipse¥LoadCSV¥New_Config¥Emp_Master.dat ファイルにファイルへッダがありません。 ファイルを開くために使用するファイル構成とレコード長を定義してください。				
ファイル定義				
ファイル編成(_): レコード順‐固定長	•			
最小の長さ(M) 0三日最大の長さ(X)	6Ç	÷		
ОК	キャンセノ	L I		



[**OK**] ボタンをクリックします。

プロファイルファイルを作成するかどうかの問いには [はい(Y)] を選択します。



社員9名分のデータが正常に固定長順編成ファイルに書き込まれていることを確認します。

🧊 Micro Focus データファイル ツー	ル - [Emp_Master.dat (固	定 長さ レコード順)]		- 0	×
🕈 ファイル(E) 編集(E) 表示()	♪ ファイル(E) 検索(<u>S</u>)	オプション(<u>0</u>) ツール(①	ウィンドウ(<u>W</u>)	ヘルプ(<u>H</u>)	- 8 ×
i 🛩 🔒					
ANSI VILA-AL					
11111113 佐藤 隆		M 営業部 19:	980401		
22222226 鈴木 尚 33333339 田中 直 4444442 山田 洋 555555555 伊藤 弘 666666668 木村 貴 77777771 中村 慎 88888884 橋本 悦 999999997 三井 薫	之 スズ [*] キ ナオユ ³ タナカ ナオミ タナカ ナオミ ヤマタ [*] ヨウィョ イトウ ヒロコ イトウ ヒロコ イトウ シンシ イトウ シンシ イトウ シンシ イトウ エツコ ミツイ カオル	F M 技術部 19: F 総務部 19: F 総務部 200 F 技術部 200 M 営業部 200 M 技術部 200 F 総務部 200 F 営業部 200 F 営業部 200	981015 990401 900701 910401 921220 930401 940805 950401		
 <mark> ダ</mark> Emp_Master.dat (固定 長さしコー	ド順) 🗙				
▲▶\ 出力 /					
準備OK			N/A	レコート 長 60	(固定) //



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 COBOL で作成します。

1 Visual COBOL for Eclipse を起動します。

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

ワークスペースは前章までで使用されたものをそのまま使用します。

2 COBOL プロジェクトを作成します。

ファイル(F)メニューから 新規(N) → COBOL プロジェクト を選択します。

プロジェクト名欄に BATCHRPT と入力し[終了(F)] ボタンをクリックします。

💷 COBOL プロジェクトの新規作成	_		Х
COBOL プロジェクト ワークスペースまたは外部の場所にCOBOL プロジェクトを作成します。		k	
プロジェクト名(P): BATCHRPT プロジェクト テンプレートを選択			
🔎 Micro Focus テンプレート			
<u> </u>	プレートの)設定を構	<u> </u>
□ テンプレートの参照			
場所:		参照…	
ファイルシステムを選択: default 〜			
✓ デフォルト・ロケーションの使用(D)			
ロケーション(<u>L</u>): C:¥work¥tutorial¥Eclipse¥BATCHRPT		参照(<u>R</u>)	
ファイル・システムを選択(<u>Y</u>): デフォルト 〜			
? 終了(E)		キャンセ	IL



3 COBOL プログラムを追加します。

COBOL エクスプローラービューにて **プロジェクト** フォルダを右クリックし

新規(N) → COBOL プログラムを選択します。

新規ファイル名欄には BATCHRPT.cbl を指定します。[終了(F)] ボタンをクリックします。

💷 COBOL プログラムの新規作成	_		×
COBOL プログラム エディタで開くことができる COBOL プログラムを新規作成します。		C	Ŷ
含まれるプロジェクト: BATCHRPT 新規ファイル名: BATCHRPT.cbl			菱照
テンプレートを選択:			
	テンプレート	の設定を	<u>毒成</u>
□ テンプレートの参照			
場所:		参照	
ファイルシステムを選択: default \vee			
?	終了(E)	キャンセ	μ



コードエディターで COBOL ソースコードを入力します。 4

*

*

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(エディタービュー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。 まだ、データ部のファイル定義が未入 力なので3件の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.
                  🛃 BATCHRPT.cbl 🖂
                     BATCHRPT.cbl
                           * 👌 - 1
                                                                                  I · •
                                                                                       ·8
                            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.
                    \Theta
                    Θ
                            INPUT-OUTPUT SECTION.
                            FILE-CONTROL.
                            INPUT FILE: EMPLOYEE RECORDS
                  0
                               SELECT EMP-SEQ-FILE
                                                     ASSIGN TO UT-S-EMPSEQ.
                             INPUT FILE: DATE SELECTION CRITERIA
                  0
                               SELECT IN-CNTL-CARD
                                                     ASSIGN TO UT-S-CNTLCARD.
                            OUTPUT REPORT FILE
                               SELECT EMP-HIRE-RPT
                                                     ASSIGN TO UT-S-HIRERPT
                  0
                            data division.
                  😑 コンソール 👔 問題 🙁 🧔 タスク 🖂 プロパティー 🌇 Table Results 🛛 Filter Definitions 🛛 🗳 コード カパレッジ 🗔 Micro Focus Unit Testing
                  3 errors, 1 warning, 0 others
                  記述/説明
                                                                        ロケーション
                                                  リソース
                                                           パス
                                                                                 タイプ
                                                                                           プログラム
                    🔇 エラー (3 項目)
                       🧭 COBCH0244S ファイル EMP-HIRE-RPT に対する F BATCHRPT.cbi /BATCHRPT
                                                                        行 21
                                                                                 COBOLの問題
                                                                                           BATCH RPT.cb
                       🚱 COBCH0244S ファイル EMP-SEQ-FILE に対する F_BATCHRPT.cbi /BATCHRPT
                                                                        行 15
                                                                                 COBOLの間題
                                                                                           BATCHRPTch
                      🚱 COBCH02445 ファイル IN-CNTL-CARD に対する F BATCHRPT.cbl /BATCHRPT
                                                                                 COBOLの問題
                                                                        行 18
                                                                                           BATCHRPT.ck
```



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
LABEL RECORDS ARE STANDARD.01CONTROL-RECORDPIC X(8).
- FD EMP-HIRE-RPT LABEL RECORDS ARE STANDARD. 01 RPT-RECORD PIC X (80).



 ① ヨケア が いい (1983) (2) アイア (1) クロパワト (1000) (2) コークパク (1000) (2) マーク・ション (2) パン (2) パン



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。

2HH 9		IMP-RECORD-IO-AREA J	>	項日18	、エノー(2/3/2/6	* 7
WOR	KING-	-STORAGE SECTION					
01	PRO	GRAM-FIELDS.					
•••	05	EOF-FLAG	PIC	X (01)	VALUE	' N' .	
		88 AT-EOF			VALUE	'Y'.	
		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	CONT	FROL-REC.					
	05	CNTL-DATE.					
		10 CNTL-YR	PIC	X (4)	VALUE	SPACE.	
		10 CNTL-MON	PIC	X (2)	VALUE	SPACE.	

** Employee Record Layout

10 CNTL-DAY

01 EMP-RECORD-IO-AREA. COPY EMPSEQ.

👩 BATCHRPT.cbl 🔀			
BATCHRPT.	cbl		
••••*A·1	··B···•···2····•····3····•	····4····•5····•6····•7··I·	• · · · · 8
01	RPT-RECORD	PIC X(80).	
⊖ ₩01	RKING-STORAGE SECTION.		
01	PROGRAM-FIELDS.		
	05 EOF-FLAG	PIC X(01) VALUE 'N'.	
	88 AT-EOF	VALUE 'Y'.	
	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	CONTROL-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		
6 01	EMP-RECORD-IO-AREA.		
8	COPY EMPSEQ.		
e pro	ceaure aivision.		

PIC X(2) VALUE SPACE.



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 X(2).	
	05	FILLER	PIC X	VALUE '/'.
	05	RPT-CURR-DD	PIC X(2).	
	05	FILLER	PIC X	VALUE '/'.
	05	RPT-CURR-YYYY	PIC X(4).	
	05	FILLER	PIC X(5)	VALUE SPACE.
01	DDT			
01				
	05		PIC X(0)	VALUE SPACES.
	05	RP1-CURR-HR	PIC $X(2)$.	
	05	FILLER	PIC X	VALUE :
	05	RPI-CURR-MIN	PIC $X(2)$.	
	05	FILLER	PIC X	VALUE ':'.
	05	RPT-CURR-SEC	PIC X(2).	
	05	FILLER	PIC X(5)	VALUE SPACE.

👔 ВАТСНЕ	RPT.cbl			
• • • • • • * ,	A·1·B·	•••••3••••	• • • • • • • • • • • • • • • • • • • •	5 · · · · • · · · · 6 · · · • • · · · · 7 · ·
	CO	PY EMPSEQ.		
*	* Rep	ort Lines		
	01 RP	T -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 Repo	rt'.	
	05	FILLER	PIC X(10)	VALUE SPACES.
	05	RPT-CURR-MM	PIC $X(2)$.	
	05	FILLER	PIC X	VALUE '/'.
	05	RPT-CURR-DD	PIC $X(2)$.	
	05	FILLER	PIC X	VALUE '/'.
	05	RPT-CURR-YYYY	PIC $X(4)$.	
	05	FILLER	PIC X(5)	VALUE SPACE.
	01 RP	T-TITLE-2.		
	05	FILLER	PIC X(67)	VALUE SPACES.
	05	RPT-CURR-HR	PIC X(2).	
	05	FILLER	PIC X	VALUE ':'.
	05	RPT-CURR-MIN	PIC $X(2)$.	
	05	FILLER	PIC X	VALUE ':'.
	05	RPT-CURR-SEC	PIC $X(2)$.	
	05	FILLER	PIC X(5)	VALUE SPACE.



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	覽'.	
	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'.		

BATCHRPT.cbl 🛛							
BATCHRPT.c	ы						
····*A·1	·B··	• • • • • 2 • • • • • • • • • • • • • •	·4 · ·	• • • • • •	5	6	7 · · I · • · ·
	05	FILLER	PIC	X(5)	VALUE	SPACE.	
01	RPT	TITLE-3.					
	05	FILLER	PIC	X(5)	VALUE	SPACE.	
	05	FILLER	PIC	X(7)	VALUE	1 * * * * * 1	
	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'.					
⊖ pro	cedu	re division.					



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	PIC	X (7)	VALUE '***** '.
	05	RPT-MSG	PIC	X (30)	VALUE SPACE.
	05	FILLER	PIC	X (2)	VALUE SPACE.
	05	RPT-TOT-RECS	PIC	ZZZ.	
	05	FILLER	PIC	X (33)	VALUE SPACE.

01 BLANK-LINE

PIC X (80) VALUE SPACE.

· · · • · * A ·	1.B		5 · · · • · · · · 6 · · • · • · · · · 7 · ·
	'雇用年数'.		
01	RPT-DETAIL-LINE.		
	05 FILLER	PIC X(5)	VALUE SPACES.
	05 RPT-EMP-DIV	PIC N(5)	VALUE SPACES.
	05 FILLER	PIC X	VALUE SPACES.
	05 RPT-EMP-NAME	PIC N(10)	VALUE SPACES.
	05 FILLER	PIC X	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 X	VALUE 1/1.
	05 RPT-EMP-HIRE-DD	PIC X(2)	VALUE SPACES.
	05 FILLER	PIC X	VALUE 1/1.
	05 RPT-EMP-HIRE-YYYY	PIC X(4)	VALUE SPACES.
	05 FILLER	PIC X(5)	VALUE SPACES.
	05 RPT-EMP-YRS-EMPL	PIC 29	VALUE ZEROES.
	05 FILLER	PIC X(12)	VALUE SPACE.
01	RPT-TOTAL-LINE.		
	05 FILLER	PIC X(5)	VALUE SPACE.
	05 FILLER	PIC X(7)	VALUE '***** '.
	05 RPT-MSG	PIC X(30)	VALUE SPACE.
	05 FILLER	PIC X(2)	VALUE SPACE.
	05 RPT-TOT-RECS	PIC ZZZ.	
	05 FILLER	PIC X(33)	VALUE SPACE.
01	DI ANIZ I THE	BTC V(90)	VALUE SPACE



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 T0 RPT-CURR-MM. MOVE CURR-DD T0 RPT-CURR-DD. CURR-YYYY TO MOVE RPT-CURR-YYYY. ACCEPT CURR-TIME FROM TIME. CURR-HR MOVE T0 RPT-CURR-HR. MOVE CURR-MIN TO RPT-CURR-MIN. MOVE CURR-SEC TO RPT-CURR-SEC.

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





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 Τ0 RPT-SELECTION-MM. MOVE CNTL-DAY Τ0 RPT-SELECTION-DD. MOVE CNTL-YR T0 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 RPT-COLUMNS BEFORE ADVANCING 1 LINE. WRITE RPT-RECORD FROM BLANK-LINE BEFORE ADVANCING 1 LINE.

1000-EXIT.

EXIT.





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 LINES	FROM	EMPLOYEE RECORD.
	MOVE EMPREC-DIV	Т0	RPT-EMP-DIV.
	MOVE SPACE	Т0	RPT-EMP-NAME.
	STRING EMPREC-JNAME1 DE		ED BY SPACE
	SPACE DE EMDREC_ INAME2 DE		ED BY SIZE
			RDT_EMD_NAME
		INTO	
	STRING EMPREC-SSN(1:7)	DELI	MITED BY SIZE
	'_'	DELI	MITED BY SIZE
	EMPREC-SSN (8:1)	DELI	MITED BY SIZE
		INTO	RPT-EMP-SSN.
	MOVE EMPREC-DOH-MM	T0	RPT-EMP-HIRE-MM.
	MOVE EMPREC-DOH-DD	T0	RPT-EMP-HIRE-DD.
	MOVE EMPREC-DOH-YYYY	T0	RPT-EMP-HIRE-YYYY.
	PERFORM 4000-COMPLITE-YEAR	S-FMPI	OYED THRU 4000-EXIT
	MOVE YRS-EMPLOYED	TO	RPT-EMP-YRS-EMPL.
	WRITE RPT-RECORD	FROM	RPT-DFTAIL-LINF
		BEFO	RE ADVANCING 1 LINE.
	ADD 1 TO EMP-REC-CNTR.	52.0	

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 🔀 - 8 BATCHRPT.cbl ŀ •••*A·1·B··•··2 ^ = 4000-EXIT. EXIT. 9000-CLOSE-AND-CLEANUP. IF EMP-REC-CNTR > 0 MOVE ·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-TE. 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. goback. - -💷 コンソール 🙁 📳 問題 🧔 タスク 🖂 プロパティー 🏫 Table Results 🛛 Filter Definitions 🗳 コード カバレッジ 📆 Micro Focus Unit Testing 🗶 🦎 🖹 🚮 🔛 🚽 🖃 🕶 😁 🕶 Micro Focus ビルド: BATCHRPT BUILD FAILED Build finished with 12 errors, O warnings, O notices and a maximum exit code of 12 C:\work\tutorial\Eclipse\BATCHRPT\.cobolBuild:173: Compilation errors have occurred

以上で BATCHRPT.cbl ソースプログラムの入力は終了です。 この時点でエラー件数が 12 であれば、問題ありません。先に進んでください。



5 コードエディターで COBOL コピーファイルを入力します。

COBOL エクスプローラービューにて COBOL プログラム フォルダを右クリックし 新規(N) → COBOL コピーファイルを選択します。

新規ファイル名欄には EMPSEQ.cpy を指定します。[終了(F)] ボタンをクリックします。

🔤 COBOL コピーファイルの新規作成	— [) X
COBOL コピーファイル エディターで開くためファイルを拡張子 .CPY で新規作成します。		
含まれるプロジェクト: BATCHRPT 新規ファイル名: EMPSEQ.cpy テンプレートを選択: ■ Micro Focus テンプレート		参照
	<u>テンプレートの</u> 設	<u>定を構成</u>
 □ テンプレートの参照 場所: ファイルシステムを選択: default ∨ 	 参	既
?	終了(<u>F)</u> キ	ャンセル



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-HIRE		
		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.





エディタービュー上の EMPSEQ.cpy をアクティブにしたまま [ファイル(F)] メニューの [保管 (S)] 或いは Ctrl + S キーで保存します。これによりアプリケーションがビルドされます。

コンソールビューにエラーなくビルドできた旨が出力されていることを確認して、次に進んでくだ さい。





6 COBOLコンパイル指令を追加します。

本項ではファイル名の割り当てを EXTERNAL(外部割り当て)に変更するためのコンパイラ指令を指 定します。

COBOL エクスプローラービューにて **BATCHRPT** プロジェクトを右クリックし [プロパティ (R)]を選択します。

Micro Focus COBOL > Project 設定 > COBOL とナビゲートし表示される画面では

警告レベル 警告を含める(レベル W)

追加指令欄 ASSIGN(EXTERNAL)

のように選択/入力し、[OK] ボタンをクリックします。

🕺 プロパティ: BATCHRPT			- 0	×
ЖИРХ Л	COBOL			
> リソース V Micro Focus ビルドパス > ビルド構成		・ ・・・・ - ・・・・・・ □ 出力の表示		Â
 ✓ プロジェクト設定 ✓ COBOL > SQL プリプロセッサ コード分析 追加のプリプロセッ 	警告レベル: 最大エラー数:	警告を含める(レベル W) 100		↓
ビルド環境 > 実行時構成 Project Facets > Task Repository Task Tags > Validation WikiText ビルダー プロジェクト参照 実行/デパック設定	追加損令: ASSIGN(EXTERNAL)		~	
< >		デフォルトの復元(1) 適用(<u>L)</u>	
?	·	OK	* ヤンセ	IL



問題ビューを選択し「ファイル名の接頭語を注記として扱う」 警告が 3 件表示されることを確認して、先に進んでください。

🛃 BATCHRPT.cbl 🕴 🛅 EMPSEQ.cpy					
☑ BATCHRPT.cbl ▶					
••••*À·1·B·••··2···•3···•	• • 4 • • • • • • • • •	.5	••••·7••I•	• · · · 8	
9000-CLOSE-AND-CLEANUP.					^ _
IF EMP-REC-CNTR > 0 MOVE 小処理レコード件数:・ MOVE EMP-REC-CNTR ELSE MOVE 小処理レコードなし・ END-IF.	TO RPT-MSG TO RPT- TO RPT-MSG	TOT-RECS			
DISPLAY '*** REPORT CREAT DISPLAY '*** VIEW:	ED SUCCESSF HIRERPT.DAT	ULLY ***'. ***'.			
WRITE RPT-RECORD FROM BLAN WRITE RPT-RECORD FROM RPT-	K-LINE TOTAL-LINE	BEFORE ADVANCIN BEFORE ADVANCIN	G 1 LINE. G 1 LINE.		
CLOSE EMP-SEQ-FILE IN-CNTL-CARD EMP-HIRE-RPT.					
⊖ 9000-EXIT. EXIT.					
goback.					~
<					>
🕒 コンソール 💌 問題 👷 🧔 タスク 🥅 プロパティー わ Table	Results 🔐 Filte	er Definitions 🛛 🖳 🗆 – K	ារ(ស្រុះ) 🔂 M	licro Focus Unit T	estina 🗆 🗖
	nesanes by rine		murry 🔄 🖉 📖	incro rocus onic r	
0 errors, 4 warnings, 0 others					0,
	リソース	<i>/</i> t͡x	ロケーション	タイプ	プログラム
✔ 💧 _警告(4項目)					
💰 COBCH1130W ファイル名の接頭語を注記として扱う	BATCH RPT.cbl	/BATCHRPT	行 15	COBOLの問題	BATCH RPT.cbl
🔐 COBCH1130W ファイル名の接頭語を注記として扱う	BATCH RPT.cbl	/BATCHRPT	行 18	COBOLの問題	BATCH RPT.cbl
COBCH1130W ファイル名の接頭語を注記として扱う COBCH1130W ファイル名の COBCH1130W COBCH1130W COBCH1130W COBCH1130W COBC	BATCHRPT.cbl	/BATCHRPT	行 21	COBOLの問題	BATCHRPT.cbl
10-カル変数 res の値が使用されていません	CallCobHello	/CallCobHello/src/	łт 7	Java 問題	



7 環境変数を構成します。

COBOL エクスプローラービューにて **BATCHRPT** プロジェクトを右クリックし[**プロパティ(R)**] を選択します。

Micro Focus COBOL > 実行時構成 > 環境変数とナビゲートします。

[追加(A)] ボタンをクリックします。

old プロパティ: BATCHRPT			— 🗆 X
ንብሥራን አ	環境変数		← → ⇒ →
> リソース FreeMarker Context > Micro Focus ビルド パマ	(注:ここで定義された変数は、任意の 環境スクリプト内の設定値を上書き)	D現在の設定値、または任意の指定された ます。)	
 ジレド構成 ジレド構成 プロジェクト設定 実行時構成 COBOL スイッチ 環境変数 Project Archives Project Facets Task Repository 	変数	値	追加(<u>A</u>) 福集(<u>E</u>) 削除(<u>R</u>)
Task Tags > Validation WikiText ビルダー プロジェクト参照 実行/デパック設定	実行する環境スクリプト: 場所: パラメータ: 回 関連付けられたプロジェクトのビルド	環境から値を継承	参照 デフォルトの復元(I) 適用(L)
(?)			0K キャンセル

変数欄 dd_EMPSEQ



值欄 Emp_Master.dat

環境変数を追加または変更します

のように入力し [OK] ボタンをクリックしま

す	-

変数:	dd_EMPSEQ	
值:	Emp_Master.dat	



	[追加(A)]	ボタンをクリックし更に追加し	· 変数を追加
ます	す。		環境変数を追加または変更します
	変数欄	dd_CNTLCARD	
	値欄	Cntl_Card.dat	変数: dd_CNTLCARD
	のように入力	コし [OK] ボタンをクリックしま	值: Cntl_Card.dat
す。			

[追加(A)] ボタンをクリックし更に追加し 🛛 🔃 変数を追加

ます。

変数欄 dd_HIRERPT

值欄 Hire_Report.dat

のように入力し [OK] ボタンをクリックしま

ー 環境変数を追加または変更します						
(<u>変</u> 数:	dd_HIRERPT				
l	值:	Hire_Report.dat				

す。

下図のように指定が反映されていることを確認して、[OK] ボタンをクリックします。

💷 プロパティ: BATCHRPT				— 🗆	×
ጋብሥፅኢታ	環境変数			(→ →	• •
 > リソース FreeMarker Context > Micro Focus ビルドパス > ビルドパス > ビルド構成 > プロジェクト設定 > 実行時構成 COBOL スイッチ 環境変数 Project Archives Project Facets 	(注: ここで定義された変数は、任意の5 環境スクリプト内の設定値を上書さしま 変数 dd_EMPSEQ dd_CNTLCARD dd_HIRERPT	現在の設定値、または任意の指定された ます。)		追加(<u>A</u>) 編集(<u>B</u>) 削除(<u>B</u>)	•
 Task Repository Task Tags Validation WikiText ビレダー プロジェクト参照 実行/デパッグ設定 	<	環境から値を継承	> デフォルトの復元	(D) 適用	R
?			ОК	キャンセ	μ



8 COBOL アプリケーションをビルドします。

プロパティー画面で [**OK**] ボタンをクリックしますと、自動ビルド機能によりビルド処理がキック されます。正常にビルドされることを確認してください。

Image: Source of the second	eee Eclipse - COBOL - BATCHRPT/BATCHRPT.cbl - E ファイル(F) 編集(F) ソース リファクタリング ナビゲー	Eclipse ー ロ ト(N) 検索 ゴロジェクト(P) 単行(R) ウェインドウ(W) ヘルプ(H)	×
2 function 2 function 2 function 3 function 3 function 3 function 4 function 4 function 4 function		Ĩ┆╧╶╦┆Q┊╬╺Ѻ╺♀ <mark>♀</mark> ╺┆ <i>╚ Ġ ┩</i> ╺┆┋╺│⋟╚,Ѻ╶╢╝╸╗┙┶╺┝╕	
C. 23 年 1. 24. 19. 1000 BATCHRPT Col 23 EMPSED.cpy BATCHRPT Col 2 BATCHRPT ConsoleHello		9/199-77927 🗄 😰 🔂 🔤	F 🐉
BATCHRFT.cbi) BATCHRFT.cbi) Consolettello Con	🔓 C 🛿 😘 J 😤 A 📃 サ 🖵 🗖	👔 BATCHRPT.cbi 🕴 🖻 EMPSEQ.cpy	
> ■ BATCHRPT > ■ CalcoleHelo > ■ ConsoleHelo > ■ ConsoleHelo > ■ LoadCSV > ■ LoadCSV </td <td>🖹 🚖 📮 🍫 🗢</td> <td>🔊 BATCHRPT.cbi 🕨</td> <td></td>	🖹 🚖 📮 🍫 🗢	🔊 BATCHRPT.cbi 🕨	
Subcontrol	> 🚰 BATCHRPT	·····*A·1·B······2·····3·····4·····5······6·····7··I·•···8	
MODE '处理レコード及し' TO RFT-HSG END-IF. DISPLAY '**** REPORT CREATED SUCCESSFULY ****'. DISPLAY '**** VIEW: HIRERPT.DAT ****'. WRITE RFT-RECORD FROM BLANK-LINE BEFORE ADVANCING 1 LINE. WRITE RFT-RECORD FROM FRT-TOTAL-LINE BEFORE ADVANCING 1 LINE. WRITE RFT-RECORD FROM FRT-TOTAL-LINE BEFORE ADVANCING 1 LINE. CLOSE EMP-SEQ-FILE IN-CNTL-CARD EMP-HIRE-RFT. 9000-EXIT. Procedure Division > EDD Data Division > Data Division > EDD Data Division > EDD Data Division > EDD DATA RIT = 1000-EXIT INCO FOCUSURE EXOMMAN-PROCESSING 3000-EXIT INCO FOCUSURE EXOM-EXIT INCO FOCUSURE EXOM-EXIT INCO FOCUSURE EXOM-EXIT INCO FOLD	CancolHello ConsoleHello ConsoleHelloJVM LoadCSV	IF EMP-REC-CMTR > 0 MOVE :処理レコード 件数:' TO RPT-MSG MOVE EMP-REC-CNTR TO RPT-TOT-RECS ELSE	^ _
BISPLAY '**** REPORT CREATED SUCCESSFULLY ***'. DISPLAY '**** VIEW: HIRERPT.DAT ****'. WRITE RPT-RECORD FROM BLANK-LINE BEFORE ADVANCING 1 LINE. CLOSE ENP-SEQ-FILE IN-CNIL-CARD EMP-HIRE-RPT. SOOO-EXIT. Procedure Division Procedure Division ONO-EXIT 1000-EXIT 1000-EXIT 1000-EXIT 2000-AXIN-PROCESSING 2000-EXIT 0000-EXIT 2000-EXIT 0000-EXIT 0000-EXIT <td< td=""><td></td><td>MOVE '処理レコードなし' TO RPT-MSG END-IF.</td><td></td></td<>		MOVE '処理レコードなし' TO RPT-MSG END-IF.	
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. BATCHRPT BATCHRPT BATCHRPT Procedure Division Sta Division Procedure Division Sta Division Non-START 1000-EXIT 2000-MAIN-PROCESSING 2000-EXIT 3000-EXIT 3000-EXIT 0000-EXIT 0000-EXIT <tr< td=""><td></td><td>DISPLAY '*** REPORT CREATED SUCCESSFULLY ***'. DISPLAY '*** VIEW: HIRERPT.DAT ***'.</td><td></td></tr<>		DISPLAY '*** REPORT CREATED SUCCESSFULLY ***'. DISPLAY '*** VIEW: HIRERPT.DAT ***'.	
CLOSE EMP-SEQ-FILE IN-CNTL-CARD EMP-HIRE-RPT. ● BATCHRPT ● BATCHRPT ● BATCHRPT ● BATCHRPT ● Drocedure Division ● Procedure Division ● Procedure Division ● Procedure Division ● Procedure Division ● Procedure Division ● Drocedure Division ● Divisi		WRITE RPT-RECORD FROM BLANK-LINE BEFORE ADVANCING 1 LINE. WRITE RPT-RECORD FROM RPT-TOTAL-LINE BEFORE ADVANCING 1 LINE.	
CLUSE EMP-HIRE-RPT. Image: Proceeding Division Image: Proceeding Division Image: Division Image: Division Imature.specific.build.cfg.			
 P70 ※ 正力 正式 □ ● 9000-EXIT. ● 9000-EXIT 		CLUSE EMP-SEQ-TILE IN-CNTL-CARD EMP-HIKE-RPT.	
Percedure Division Dota Division Dota Division Procedure Division O Dota Division	語アウ 🛛 📰 ブロ 📰 コピー 🖓 日	⊕ 9000-FXTT.	
 ● BAILHRY1 ● Data Division ● Data Divi	© NATCUPPT	RYTT	~
Data Division Data Division O Procedure Division O Procedure Division O Procedure Division O Divis	 BAILERPT Revironment Division 		_
✓ ● FOCEULIE DAISING ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲	> 🎝 Data Division	U コンソール X 👔 間題 🖉 ダスク 🔲 プロパティー 🌇 Table Results 🧃 Filter Definit 🍟 コード カバレッジ 💽 Micro Focus 🤤	• -
 1000-EXIT 2000-MAIN-PROCESSING 2000-EXIT 3000-FROCESS-RECORD 3000-EXIT 4000-COMPUTE-YEARS-EMPLOD 4000-CLOSE-AND-CLEANUP 9000-EXIT 9000-EXIT 	✓ ● 1000-START	Micro Focus 빈다: BATCHRPT	2 •
2000-EXIT 9000-EXIT 9000-EXIT 9000-EXIT 9000-EXIT 9000-EXIT 9000-EXIT 9000-EXIT 9000-EXIT	 1000-EXIT 2000-MAIN-PROCESSING 	nature.specific.build.cfg.New_Configuration:	^
 S000-EAT Cobolbuild.cfg.New_Configuration: Cobolbuild.cfg.New_Configuration: BUILD SUCCESSFUL BUILD SUCCESSFUL BUILD SUCCESSFUL Build finished with no errors. 	 2000-EXIT 3000-PROCESS-RECORD 	post.build.cfg.New_Configuration:	
 4000-EAT 9000-CLOSE-AND-CLEANUP 9000-EXIT 9000-EXIT 	 3000-EXIT 4000-COMPLITE-YEARS-EMPLOY 	cobolbuild.cfg.New_Configuration:	
 9000-EXIT 	 4000-EXIT 4000-EXIT 	BUILD SUCCESSFUL	
	 9000-CLOSE-AND-CLEANUP 9000-EXIT 	bulla finished with no errors.	
Total time: 0 seconds		Total time: 0 seconds	~
	<	<	>



9 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、COBOL エクスプローラービューより選択します。

右クリックより「コピー」を選択します。

BATCHRPT プロジェクト配下の New_Configuration.bin フォルダを選択し右クリックから 「貼り付け」を選択します。

COBOL エクスプローラー上でデータファイルコピーが反映されたことを確認します。



10 制御ファイルを作成します。

New_Configuration.bin フォルダを右クリックし

新規(N) → ファイルを選択します。

[ファイル名] 欄にて Cntl_Card.dat を指定し [終了(F)] ボタンをクリックします。

[20] 新規ファイル	_		×
ファイル 新しいファイルリソースを作成します。			
親フォルダを入力または選択(<u>E</u>):			
BATCHRPT/New_Configuration.bin			
 BATCHRPT New_Configuration.bin CallCobHello ConsoleHello ConsoleHelloJVM InternalTDProject LoadCSV 			
ファイル名(M): Cntl_Card.dat			
拡張 >>(<u>A</u>)			
? 終了(Ð	キャンセ	IL



データファイルツールが起動します。本チュートリアルではこのテクノロジープレビュー版ではな く従来より提供しているクラシック版を利用しますのでここでは [**キャンセル**] ボタンをクリックし、 続けて [×] アイコンをクリックしてデータファイルツールを閉じます。

🗾 Micro Focus データファイル ツ	ール (テクノロジープレビュー)					- • ×
ファイル 編集 表示 検索 ツール	レヘルプ					\sim
- 7 7 1 lu - 7 7 1 lu - 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	レ: データファイルを開く		ビゲーション	外観 15 ANSI V	フィル マー 無し ×	
r.	ファイル名 ―――					
	C:\work\tutorial\Eclipse\B/	ATCHRPT\New_Configuration	.bin\Cntl_Card.dat	资 参照。		
[構造ファイル (任意) —			资 参照。		
	ファイルの詳細					
	編成:	順次 🗸	形式:	00 - 既定	<pre>////////////////////////////////////</pre>	
			エンコード:	ANSI 🔍	•	
	可変長レコード	:				
	最小の長さ:	0	最大の長さ:	1		
	Cntl_Card.pro I	に詳細を保存				
				バックアッ	フ	
	 ,	有モードで開く 排他日	Eードで開く 🛛 🖛	キャンセル		

COBOL エクスプローラーにて Cntl_Card.dat を右クリックし、

アプリケーションから開く → クラシックデータファイルツール へとナビゲートします。

			.	MUVE I WEATER THAT
Cntl_	新規作成(N)	>		END-IF.
> 🔁 CallCobHello	開<(0)			DISPLAY '*** REPORT (
> 🖉 ConsoleHe	アプリケーションから開く	>	٥	クラシック データファイル ツール
🗸 🖉 LoadCSV 🛛 👔	JĽ-	Ctrl+C		テータファイル ツール
> 📴 COBOL 🕽 💼	貼り付け	Ctrl+V		テキストエディタ



[ファイル編成] 欄はデフォルトの 「レコード順 - 固定長」のままにし て、[最大の長さ] 欄には「8」を指定し ます。[OK] ボタンをクリックします。

7 ° -9771¥ I7° 19		?	×				
- ファイル名 C:¥work¥tutorial¥Eclipse¥BATCHRPT¥New_Config¥Cntl_Card.dat ファイルにファイルヘッダがありません。 ファイルを開くために使用するファイル構成とレコード長を定義してください。							
-7ァイル定義							
/ア1ル編成(型):		<u> </u>					
最小の長さ(<u>M</u>)	□ 🗄 最大の長ざ(X)	ξ_					
	ОК	キャンセル					

プロファイルファイルを作成するかどうかの問いには [はい(Y)] を選択します。



右クリックから [後にレコードを挿入する] を選択します。

🐮 Micro Focus データファイル ツール - Cntl_Card.dat (固定 長さ レコード順)					- 0	×
ファイル(F) 編集(E) 表示(V) ファイル(F) 検索(S) オプション(O) ツール(T)	ウィンドウ(W) ヘルプ(H)					
j 📂 🖬						
ANSI V +- au V IEEE V S						
🍞 Cntl_Card.dat (固定 長さ レコート・順)						×
	16進表示(<u>H</u>)	Alt+F2				
	貼り付け	Shift+Ins				
	前にレコードを挿入す	(B)				
	後にレコートを挿入す	а(<u>А</u>) F3				
】	レコードの削除(<u>D</u>)	F4				
	3 ファイル情報(1)					
< ▶ 出力 /						
後ろにレコードを挿入する	N/A	レコート 長 -	- íī	行-、列 -		NUM //



「20110101」を入力します。

更新の警告がポップアップされましたら、[現在のファイルの更新の警告をオフにする] にチェック を入れ、[はい] ボタンをクリックします。

更新の警告	?	×
●更新の警告 更新の警告はオンです。このレコートを更新しますか? ▼現在のファイルの更新の警告をオフにする		
(はい いいえ		



[**ファイル**] メニューから [保存] を選択してファイルを保存します。

Data File Editor を終了します。



11 COBOL アプリケーションをデバッグ実行します。

New_Configuration.bin フォルダ配下の BATCHRPT.exe を右クリックし

デバッグ(D) → COBOL アプリケーションを選択します。

パースペクティブ切り替えの確認メッセージには [はい(Y)]を選択します。



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 文の処理前に一時停止している状態となっています。

Eclipse - デパッグ - BATCHRPT/BATCHRPT.cbl - Eclipse ファイル(F) 撮集(E) ソース リファクタリング ナピケート(N) 検索 プロジェクト(P) 案行(R) ウィンドウ(W) ヘルプ(F)	×
	□
	クイック・アクセス 🕴 💼 🔯 🕹
☆ デパッグ 22 続 Servers 🧏 🝃 ▽ 🖵 🗖 (x)= 変	数 💁 ブレークポイント 🥳 式 🙁 🧼 🖄 🖘 🕞 🕂 💥 💥 🌣 😐 🗖
 ◇ ■ BATCHRPT.exe [BATCHRPT] [COBOL アプリケーション] ◇ ● Micro Focus デ(ワカ: C:\woork\#tutorial\#clipse\@ATCHRPT\New_Configuration.bin\#BATCHRP } ◇ ● アプリケーション スレッド: 40(-日時度止) ◇ ● アプリケーション スレッド: 4108 (一時停止) 	植 ダ "IN-REC" <評価中のエラ-> ダ "OUT-REC" <評価中のエラ-> がしい式を追加
	,
BATCHRPT.cbl IX EMPSEQ.cpy BATCHRPT.cbl IX OS RPT-TOT-RECS PIC ZZZ. OS RPT-TOT-RECS PIC ZZZ. OS FILLER PIC X(33) VALUE SPACE. O1 ELANK-LINE PIC X(80) VALUE SPACE. PROCEDUKE DIVISION. PERFORM 1000-START THRU 1000-EXIT. PERFORM 2000-NAIN-PROCESSING THRU 2000-EXIT. STOP RUN. 1000-START SECTION. OPEN INPUT EMP-SEQ-FILE	
ロンソール 窓 の タスク デバケグ メフセージ コンソール :[BATCHRPT.exe] [プロセス ID: 4948] Loaded: [User Module] BATCHRPT- Symbols loaded from: C:/work/tutorial/Eclip	Image: State of the state o



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、CONTROL-REC に格納される値の 変遷を追います。データ部の CONTROL-REC を選択します。

01	CONTRO	L-REC.			
	05 CN	TL Working-Storage S	Section, GROUP,	参照=1, サ∕	イズ <u>= 8 パイト</u>
	10				ACE.
	10	CNTL-MON	PIC	X(2)	VALUE SPACE.
	10	CNTL-DAY	PIC	X(2)	VALUE SPACE.

この状態で右クリックし、「項目を検査」を選択します。

「**式ビューに追加**」をクリックします。

✓ Q. CONTROL-REC=	
> 🔷 CNTL-DATE =	
	0
<	>
	式位-位追

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 を同じ要領で式ビューに追加します。(前章で追加した IN-REC 及び OUT-REC は式ビューか ら削除しても構いません。)

(×)= 変数 💊 プレークポイント 😪 式	x 🗠 🐗 🖻 🕂 🐳 🗴 🗸 🖓	
名前	値	
> *** "CONTROL-REC"		
> ≚ [*] ? ^y "EMP-RECORD-IO-AREA"	000	J
🐈 新しい式を追加		
		\sim
<		>
		1


手続き部 **1000-START 節**の READ 文に続く IF 文でエディタービュー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 文でエディタービューの左端をクリックし、ブレークポイントを設定します。





実行(R)メニューから 再開(M) を選択するか F8 キーを打鍵すると、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。

式ビュー中の 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。

(×)= 変数 🔍 プレークポイント 🕵 式	x 🕹 🦇 🕒 🖛 🖾 🛛 🕱	
名前	值	
> ^{X+y} "CONTROL-REC"	20110101	
X+Y "EMP-RECORD-IO-AREA"	000	
💠 新しい式を追加		
		^
		\sim
<		>

実行(R)メニューから 再開(M) を選択するか F8 キーを打鍵すると、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。

式ビューの 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。

ເສ]		🏝 🍂 🕞 🖕 🗶	💥 🗸 🗖 🗖
値			
20110101			
11111113 佐藤	隆	サトウ タカシ M 営業部	19980401
	t 窓 値 20110101 11111113 佐藤	t ⊠ 値 20110101 11111113 佐藤 隆	t X2 値 20110101 1111113佐藤 隆 サトク タカン M 営業部

同様に実行(R)メニューから 再開(M) を選択するか F8 キーを打鍵すると、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。

(×)= 変数 🕒 プレークポイント 👧 式 🛛				<u>≵</u> _ ⇒ta
名前	値			
> *** "CONTROL-REC"	20110101			
Image: Second	22222226 鈴木	尚之	スズキナオユキ M 技術部	19981015
EMP-RECORD-IO-AREA	22222226 鈴木	尚之	スズキナオユキM技術部	19981015
🗸 🧇 EMP-REC	22222226 鈴木	尚之	スズキナオユキ M 技術部	19981015
EMPREC-SSN	22222226			
FILLER				
EMPREC-JNAME1	鈴木			
EMPREC-JNAME2	尚之	展開して	集団項目の子要素の	
鈴木 16進: 9E9D848484		格納値を です。	確認することも可能	
7968101010				



さらに **F8**キーを8回、 **F5(ステップイン)**キーを1回打鍵すると、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.c	bl 🕱 🙋 EMPSEQ.cpy					
	🔊 BATCHRPT.cbi 🕨						
		*A·1·B······6·····7··I····8					
		WRITE RPT-RECORD FROM RPT-COLUMNS BEFORE ADVANCING 1 LINE. WRITE RPT-RECORD FROM BLANK-LINE BEFORE ADVANCING 1 LINE.	^				
	Θ	1000-EXIT. EXIT.					
	Θ	2000-MAIN-PROCESSING. READ EMP-SEQ-FILE INTO EMP-RECORD-IO-AREA AT END MOVE 'Y' TO EOF-FLAG.					
,e		IF NOT-AT-EOF PERFORM 3000-PROCESS-RECORD THRU 3000-EXIT END-IF.					
	Θ	2000-EXIT.					
\odot		EXIT.					
	Θ	3000-PROCESS-RECORD.					

実行(R)メニューから 再開(M) を選択するか STOP 文を実行するまで F5 キーを押すと、デバッガーは終了します。

🏇 デバック 😒 🚜 Servers ✓ (読 <終了しました>BATCHRPT.exe [BATCHRPT] [COBOL アプリケーション] 6^O Micro Focus デパッガ: C:¥work¥tutorial¥Eclipse¥BATCHRPT¥New_Configuration.bin¥BATCHRPT.exe (停止)

ワークスペース右上コーナにあるパースペクティブ切り替えボタンエリアにて「**COBOL**」を選択し、COBOL パースペクティブに戻ります。





COBOL エクスプローラービューにて BATCHRPT プロジェクト配下の New_Configuration.bin フォルダを右クリックし [更新(F)] を選択します。

New_Configuration.bin 配下にに Hire_Report.dat が生成されていることを確認します。



Hire_Report.dat を右クリックし

アプリケーションから開く → テキスト・エディターを選択します。

社員9名分のデータが表示されることを確認します1。

📄 Hire_Report.dat 🛿 🙍 B	3ATCHRPT.cbl			
Program: BATCHRP	Years	Employed Repor	t	08/15/2017
***** 2011年	∓ 1月 1日以前に入社	した社員一覧		08:04:38
部署名 社	t員名	社員番号	入社日	雇用年数
 営業部 位 技術部 新 総務部 田 営業部 山 技術部 伊 営業部 オ 技術部 中 総務部 構 営業部 三 	左藤 隆 隆二 戸 御 御 御 御 御 御 御 御 御 御 御 御 御 御 御 御 御 御	111111-3 222222-6 3333333-9 444444-2 5555555-5 66666666-8 777777-1 8888888-4 9999999-7	04/01/1998 10/15/1998 04/01/1999 07/01/2000 04/01/2001 12/20/2002 04/01/2003 08/05/2004 04/01/2005	19 19 18 17 16 15 14 13 12
***** 処理し	ノコード件数:	9		

¹ Eclipse におけるデフォルトのテキストエディタフォントがプロポーショナルになっている場合は多少 見た目が異なる可能性があります。この場合、テキストエディタ上で右クリックから [設定] を選択し [一般] > [外観] > [色とフォント] で表示されるページにてフォントを変更できます。



続いて、制御ファイルの値を変更してアプリケーションを実行します。 COBOL エクスプローラー上で Cntl_Card.dat ファイルを右クリックし アプリケーションから開く → テキスト・エディターを選択します。

Cntl_Card.dat ファイル中のデータを「20000101」に更新します。

BATCHRPT.cbl	📄 *Cntl_Card.dat 🛛	
20000101		

下図のような警告が出た場合は、はいを選択し、編集を進めてください。

📖 派生ファイルが検出されました	×
このファイルは派生ファイルです。編集しますか?	
□ 今後、派生ファイルを編集する場合に警告は表示されません。	(Ŋ)

[ファイル] メニュー > [保管] もしくは Ctrl + S を打鍵して、ファイルを保存します。

New_Configuration.bin フォルダ配下の BATCHRPT.exe を右クリックし

実行 → COBOL アプリケーション

を選択しアプリケーションを実行します。

コンソール画面がポップアップされたら、メッセージに従い、何等かのキーを打鍵しアプリケーションを終了します。

C:¥Windows¥SYSTEM32¥cmd.exe	_	×
*** REPORT CREATED SUCCESSFULLY ***		^
*** view: nikexri.Day *** 続行するには何かキーを押してください		
		~



Hire_Report.dat を右クリックし

アプリケーションから開く → テキスト・エディターを選択します。

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2。

Ø	BATCHRPT.cbl	📄 Cntl_Card.dat	📄 Hire_Report.dat 🛛		
	Program: BA	TCHRPT	Years Employed	Report	08/24/2017
	****	2000年 1月 1日以	前に入社した社員・	一覧	07:12:51
	部署名	社員名	社員番号	B 入社日	雇用年数
	営業部	佐藤 隆	1111111	-3 04/01/19	98 19
	技術部	鈴木 尚之	2222222	-6 10/15/19	98 19
	総務部	田中 直美	3333333	-9 04/01/18	199 18
	*****	処理レコード件数	रः	3	

続いて、制御ファイルの値を変更してアプリケーションを実行します。 COBOL エクスプローラー上で Cntl_Card.dat ファイルを右クリックし

アプリケーションから開く → テキスト・エディターを選択します。

Cntl_Card.dat ファイル中のデータを「19980101」に更新します。

🔬 BATCHRPT.cbl	📄 *Cntl_Card.dat 🖂	
1998þ101		

[**ファイル**] メニュー > [保管] もしくは Ctrl + S を打鍵して、ファイルを保存します。

² 正しく表示されない場合は、COBOL エクスプローラー上で一度ファイルを右クリックの上 **更新** を 選択しリフレッシュさせてください。



New_Configuration.bin フォルダ配下の BATCHRPT.exe を右クリックし 実行 → COBOL アプリケーションを選択しアプリケーションを実行します。

コンソール画面がポップアップされたら、メッセージに従い、何等かのキーを打鍵しアプリケーションを終了します。

C:¥Windows¥SYSTEM32¥cmd.exe	_	×
*** REPORT CREATED SUCCESSFULLY ***		~
*** VIEW: HIRERPT.DAT		
続行するには何かキーを押してください		
		\sim

Hire_Report.dat を右クリックし

アプリケーションから開く → テキスト・エディターを選択します。

処理されたレコードない旨の出力が表示されることを確認します。

🔊 BAT	TCH RPT.cbl	📄 Hire_Repor	t.dat 🖾						
P	rogram: BA	TCHRPT		Years	Employed Re	eport		08/24/2017	
	*****	1998年 1月	1日以前	前に入社	した社員一覧	覧		07:17:16	
	部署名	社員名			社員番号		入社日	雇用年数	
	*****	処理レコー	ドなし						



2018 年 9 月 21 日

第5版

https://www.microfocus.co.jp/