Micro Focus Visual COBOL

自習書

~ UNIX/Linux 版 リモート開発編 ~





はじめに

Micro Focus Visual COBOL は、約 40 年にも及ぶ実績を築きながらも今も機能拡充し進化を 続ける COBOL コンパイラを備えた COBOL 開発環境製品です。この Visual COBOL の UNIX/Linux 版製品が装備するリモート開発機能を利用すれば高品質、高機能なオープンソース の統合開発環境(IDE)として広く普及する Eclipse 上でこれらの環境をターゲットとしたアプリ ケーションを直接開発することが可能です。Visual COBOL に付属する Eclipse に対して Micro Focus は COBOL の開発向けに様々な開発補助機能が実装しており、ユーザは高い生産性を見込 めます。更に、製造・試験工程で直接 UNIX/Linux 側のリソースを利用できるため、従来のクロ ス開発機能と比べ効率は一層上がります。

本書は、Micro Focus Visual COBOL の UNIX/Linux が提供するリモート開発機能を学ぶた めの自習書です。本書の読者は、事前に「Micro Focus Visual COBOL for Eclipse 自習書」の 内容を一通り体験していることを前提とします。本書の内容は UNIX/Linux 版をご利用のユーザ 向けに同書の続版のかたちで執筆しています。

また、本書に掲載している画面イメージは Windows 10 Enterprise 32 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。UNIX/Linux を想定したコマンドイメージは Red Hat Enterprise Linux 7.1 で取得しています。Visual COBOL が提供するコマンドは全 UNIX/Linux 版で基本的に共通です。しかし、OS コマンドに 関しては OS によっては異なる場合もあるため、異なる場合はそれぞれ適切な OS コマンドに置 き換えて実行してください。



第1章 自習環境の準備

Visual COBOL の UNIX/Linux 版の開発ライセンスは Windows にインストールして利用する Micro Focus Visual COBOL for Eclipse と UNIX/Linux 環境にインストールする Micro Focus Visual COBOL Development Hub がセットになったライセンスです。Windows 側の環境について は事前に「Micro Focus Visual COBOL for Eclipse 自習書」の内容に従い済ませて整えておいてくだ さい。本章では、Micro Focus Visual COBOL Development Hub のセットアップについて紹介しま す。

1 ダウンロードしたインストーラをターゲットの OS ヘファイル転送します。

2 ReadMe を確認し、インストール要件が揃っていることを確認します。

3 転送されたインストーラを解凍します。

4 スーパーユーザ権限を持ったユーザへ切り替えます。

5 解凍したインストーラへ実行権限を与えます¹。

【実行例】

chmod +x setup_visualcobol_devhub_2.3_redhat_x86_64
#

¹ インストーラのファイル名は、「setup_visualcobol_devhub_2.3_〈プラットフォーム名〉」の形式で構成されており、x86_64 RedHat 版以外の製品を利用される場合はこの部分が異なるため、注意してください。実行時はファイル名に合わせて適切な名前に置き換えてください。



6 インストーラを開始します²。

【実行例】

# ./setup_visualcobol_devhub_2.3_redhat_x86_64 -installlocation=/opt/mf/VC23				
Micro Focus Product - Product Extractor www.microfocus.com				
Please Wait. Extracting Payload				
Creating work area				
~中略~				
製品をインストールする前に「使用許諾契約」のコピーが必要な場合は、 同意しないで、次のコマンドでインストーラを再度実行してください :				
./setup_visualcobol_devhub_2.3_redhat_x86_64 -EULA 使用許諾契約の条件を確認 し、問題なければ「y」を入力				
使用許諾契約の条件に同意しますか? (y/n): y < しょ g。 Micro Focus Visual COBOL Development Hub 2.3 の SOA サポートを構成するには、 \$COBDIR/bin/casperm.sh を実行してください。				
-=-==================================				
環境を設定するため、"cobsetenv"を実行してください。				
. /opt/mf/VC23/bin/cobsetenv				
 #				

² デフォルトのインストールディレクトリは「/opt/microfocus/VisualCOBOL」です。本例では 「-installlocation=/opt/mf/VC23」を指定し、インストールディレクトリを変更しています。



第2章 Development Hub のインストール確認

Visual COBOL Development Hub は本書で紹介するリモート開発機能に加えて従来の Micro Focus の COBOL 製品が提供するコマンドラインインターフェース機能も引き継いでいます。本章で は前章でインストールした Visual COBOL Development Hub が正しくインストールされたことをこ のコマンドラインインターフェースを使ったコンパイル及びテスト実行作業を通じて確認します。

1 ライセンスが未投入の場合は、ReadMe 等に従い、ライセンスを投入します。

2 一般ユーザに戻ります。

3 Visual COBOL の利用に必要な環境変数を整えます。

Visual COBOL Development Hub をインストールすると Visual COBOL の利用に最低限必要な環境変数をセットアップするスクリプトが

<インストールディレクトリ>/bin/cobsetenv

に用意されます。本ステップではこのセットアップスクリプトを実行して環境変数設定をします。

【実行例】

\$. /opt/mf/VC23/bin/cobsetenv
COBDIR set to /opt/mf/VC23
\$

このスクリプトにより設定される主な環境変数を下記に記します。

- > COBDIR: 製品のベースディレクトリ(インストールディレクトリ)
- > PATH: \$COBDIR/bin
- > ライブラリ探索パス³: \$COBDIR/lib

³ LD_LIBARY_PATH, LIBPATH, SHLIB_PATH 等、プラットフォームによって環境変数名は異なり ます。



4 製品同梱サンプルをコピーします。

Visual COBOL Development Hub をインストールすると \$COBDIR/demo ディレクトリ配 下にサンプルプログラム及びビルドスクリプトがカテゴリ分けされて配置されます。ここで は、このサンプル中における簡単なコンソールアプリケーションプログラムをワークディレク トリにコピーします。



5 コピーしたプログラムを実行形式にコンパイルします。

Visual COBOL は COBOL プログラムを実行形式、ライブラリファイル、呼び出し可能な共 有オブジェクト、動的ロードモジュール等、目的に応じて適切な形式ヘビルドする機能を有し ます。ここでは、コピーしたサンプルプログラムを実行形式ファイルヘシングルステップでビ ルドします。下記のコマンド実行結果からもわかるように、この1つのコマンドにより、中間 コード、オブジェクトコードの生成並びに実行形式へのリンクが処理されていることがわかり ます。





6 ビルドしたアプリケーションをテスト実行します。

Visual COBOL Development Hub にはテスト実行機能が装備されており、コンパイル・ビルドしたモジュールを同環境上でテスト実行することが可能です。ここではこの機能を使って、生成した実行形式をテスト実行してみます。tictac は〇×ゲームのロジックを COBOL で組み上げたものとなります。プロンプトに従って〇×ゲームを進めてみてください。

【実行イメージ】

実行形式を実行

\$./tictac To select a square type a number between 1 and 9 Shall I start ?

② 先攻/後攻を選択、本例では player が先攻となるよう選択

Shall I start ? n

③ ゲーム画面に切り替わるので、フィールドを選択してゲームを実行

To select a square type a nu Please select an empty squar	mber betwee e O	n 1 and	9	
	7 	8 	9	
	 4 	+ 5 	6	
	+ 1 	2 	3	

④ ゲーム終了後、アプリケーションの終了を選択

To select a square type a number between 1 and 9	
You win	
Play again ? n	



第3章 リモートサーバーを起動

リモート開発は実際の操作対象が UNIX/Linux 側にあるにもかかわらず Windows 上の Eclipse にてあたかもローカルのリソースをコーディング編集やデバッグするかのようにして操作させることを 可能にする技術です。このリモート開発を実行するにあたり、UNIX/Linux 側では Windows からの 操作要求を受け付けるためのリモートサーバーを起動する必要があります。UNIX/Linux と Windows の間の接続には、Eclipse が提供する RSE(Remote System Explorer) フレームワーク、もしくは SAMBA や NFS のようなネットワークファイルシステムが利用できます。ここではパフォーマンスの 観点で有利な RSE で接続してみます。この RSE に関してもプロジェクトのポリシー(スーパーユー ザ権限を持ったユーザの利用制限、ファイヤウォール等)に応じて柔軟に対応できるよう、デーモンを 使って接続を自動確立させる方法並びに SSH でマニュアル接続させる方法を用意しています。本章で は、デーモンによる自動接続を使った方法を紹介します。

1 スーパーユーザの権限を持ったユーザに切り替えます。

2 ユーザーロケールを SJIS に設定します。

3 Visual COBOL の利用に必要な環境変数を整えます⁴。

【実行例】

. /opt/mf/VC23/bin/cobsetenv
COBDIR set to /opt/mf/VC23
#

4 デーモンを起動します⁵。

\$COBDIR/remotedev/startrdodaemon Checking Java Version Correct Java Version installed, proceeding Starting RSE daemon... Daemon running on: localhost.localdomain, port: 4075

⁴ OS によっては、su コマンドの実行時にライブラリ探索パスをクリアするものもあるようです。前章 で設定した環境変数が正しく引き継げていれば本作業は不要です。

⁵ デフォルトでは、4075 ポートはデーモンに、ランダムな 5 桁のポートには各 Windows との通信用 に割り当てます。これらのポートがファイヤウォール等により閉じている場合は、

^{\$}COBDIR/remotedev/startrdodaemon <デーモンのポート番号> <Windows との通信ポート範囲> のような形式で任意のポートへ割り当てることも可能です。



第4章 COBOL リモートプロジェクトの作成

前章にて UNIX/Linux 環境側で Windows と通信するための準備作業が完了しました。ここから は、Windows 上にインストールされた Visual COBOL for Eclipse を使って UNIX/Linux 環境上に 直接 COBOL アプリケーションをビルド生成してみます。アプリケーションのリソースは事前学習で 利用した「Micro Focus Visual COBOL for Eclipse 自習書」で用意したものを利用します。

1 Visual COBOL for Eclipse を起動します。

ワークスペースの指定は特にありません。

2 COBOL リモート プロジェクトを作成します。

① [ファイル] メニューから [新規] > [COBOL リモートプロジェクト] を選択します。

CBL	COBO	L - Eclipse	:				
ファ	イル(F)	編集(E)	ナビゲート(N)	検索	プロジェクト(P) 3	実行(R) ウィンドウ(W) ヘルプ(H)
	新規(N)		Alt	+シフト+N >	鬯	COBOL JVM プロジェクト
	ファイノ	レを開く(.)				2	🖇 COBOL プロジェクト
	閉じる	5(C)			Ctrl+W	2	🖇 COBOL コピーファイル プロジェクト
	すべて	閉じる(L)		Ctrl	+シフト+W	ष्ट्र	🕴 COBOL リモート プロジェクト
		-				Ŷ	🛊 リモート COBOL コピーファイル プロジェクト
	保管(S)			Ctrl+S		プロジェクト(R)
	別名(杀存(Α)				~	
	すべて	保管(E)		Ctr	l+シフト+S	6	COBOL JVIVI 75X



[プロジェクト名]欄にプロジェクト名を指定し、[次へ]ボタンをクリックします。
 ファイルシステム及びコンパイルタイプはデフォルトのままにしておきます。

New リモート COBOL プロジェクトの新規作成		\times
リモート cobol プロジェクト	1	
ワークスペースまたは外部にリモート COBOL プロジェクトを作成	- 10	×.
	_	/
プロジェクト名: RemoteTutorial		
ファイル システム		
ファイル システムを選択: リモート ファイル システム (RSE)		\sim
リモート ファイル システムの場合、RSE サポートによりリモート プロジェクトで作業できます。ローカル ムのロケーションの指定は不要で、リモートマシン上のロケーションの指定だけが必要です。	ファイルシス	77
ネットワーク ファイル システムの場合は、ローカルマシン上のプロジェクトの場所(マップされたドライフ クトパス)とリモートマシン上のパスを指定する必要があります。	「上のプロシ	ĴΙ
コンパイル タイプ		
COBOL コンパイル タイプ: ネイティブ コード		\sim
コンパイルタイプで "ネイティブ コード" を選択すると、リモートマシンのプロセッサ上に直接実行可 ドにコンパイルされます。	能なマシン	
コンパイル タイプで "JVM バイトコード" を選択すると、すべての JVM 互換の環境で実行可能な シンコードにコンパイルされます。	Java 仮想	<u>ک</u>
(P) < 戻る(B) 次へ(N) > 終了(E)	キャンセノ	IL

③ プロジェクトテンプレートはデフォルトの [Micro Focus テンプレート] を選択し [次へ] ボ タンをクリックします。

_

「国」リモート COBOL プロジェクトの新規作成	— 🗆 X
リモート cobol プロジェクト	
ワークスペースまたは外部にリモート COBOL プロジェクトを作成	
プロジェクト テンプレートを選択	
Micro Focus テンプレート	
□ テンプレートの参照	
場所:	参照
〈戻る(B) 次へ(N) >	終了(E) キャンセル



④ [接続の新規作成] ボタンをクリックします。

🔤 リモート COBOL プロジェクトの新規作	■成	— 🗆 X
リモート cobol プロジェクト		
ワークスペースまたは外部にリモート COBC	DL プロジェクトを作成	
プロジェクト名: RemoteTutorial		~
リモート設定		
接続名:		∨ 接続の新規作成
IJŦ-ŀ(✓ ▲ Browse
リモート ロケーションはリモート マシンのプ	ロジェクト パスに設定しなければいけません。	
(?)	< 戻る(B) 次へ(N) >	終了(E) キャンセル
0		

⑤ [Micro Focus DevHub(RSE 経由)] が選択されていることを確認し、[次へ] ボタンをクリックします。

New Connection	on					
Select Remote S	ystem Type				п	
Micro Focus Devi ス	Hub - SSH プロトコル	レによるリモートファイルシン	ステム(RSE)のファイ	ルアク	₽ = Ů	
System type:						
7 ィルタ入力						
	c D 11 1 co					
Micro	Focus DevHub SS	H 使用				



⑥ Windows 側にて UNIX/Linux サーバーの名前解決できるのであれば [Host name] 欄にそのホスト名を入力します。名前解決できない場合は、[Hostname] 欄にはそのサーバーの IP アドレスを指定します。[Connection name] 欄は自動で [Host name] 欄の値がコピーされます。指定が終わりましたら [終了] ボタンをクリックします。

New Connection		×	
Remote Micro Focus Define connection info	DevHub (RSE 経由) System Connectior ormation		
Parent profile:	DESKTOP-4BQMBC1		~
Host name:	10.18.11.108		~
Connection name :	10.18.11.108		
Description :			
Verify host name	gs		
?	< 戻る(B) 次へ(N) > 終了(F)	キャンセ	ll -

⑦ [Browse] ボタンをクリックします。

🔤 リモート COBOL プロジェクトの新規作成	— 🗆 X
リモート cobol プロジェクト	
ワークスペースまたは外部にリモート COBOL プロジェクトを作成	
プロジェクト名: RemoteTutorial	
リモート設定	
接続名: 10.18.11.118	◇ 接続の新規作成
JE-FI	✓ ▲ Browse
リモート ロケーションはリモート マシンのプロジェクト パスに設定しなければいけません。	



⑧ [My Home] の左の展開アイコンをクリックします。

Browse For Folder	×
Select a folder	
My Home	
Ny Home Noot	

 ③ UNIX/Linux 側で利用する一般ユーザの認証情報を [User ID] 欄及び [Password] 欄に入 力します。[Save password] にチェックを入れ、[OK] ボタンをクリックします。

Enter Password	×
System type: Host name: Connection name: <u>U</u> ser ID: <u>P</u> assword (optional):	Micro Focus DevHub (RSE 経由) YMRHEL71 10.18.11.118 yoshihiro
· · ·	②Save password QK キャンセル(A)

⑩ [Secure Storage] に関するポップアップが返ってきたら [はい] ボタンをクリックします。





① Password Recovery 用の質問と回答を登録します。ここでは、例として母親の旧姓と出生した都市名を記入します。用意ができましたら、[OK] ボタンをクリックします。

	Password Recovery — 🗆 🗙
	Password Recovery Setup
	Specify the questions and answers required for future password recovery.
	To be able to recover a lost 'master' password for the secure storage, enter questions and their expected answers. The questions will be asked when 'Recover Password' is pressed on the 'Secure Storage' preference page.
\searrow	The answers are case sensitive. Treat answers as secondary passwords.
	Question 1
	Question: What's your mother's maiden name?
	Answer: ***
	Question 2
	Question: What's the city where you were born?
	Answer: ***
	OK Cancel

② 下図のような警告が返ってきましたら [はい] ボタンをクリックします。



③ 下図のような警告にも [はい] ボタンをクリックします。

Ę	8L Warr	ning >	×
	?	C:¥Users¥tarot¥.ssh¥known_hosts does not exist. Are you sure you want to create it?	
		(はい(<u>Y</u>) いいえ(<u>N</u>)	



④ 下図のような .ssh が生成できた旨のメッセージが返ってきます。内容を確認し [OK] ボタン
 をクリックします。



INIX/Linux 側でソースや生成されるモジュール等を格納するプロジェクトディレクトリとして利用するディレクトリをツリーで選択し、[OK] ボタンをクリックします。

Browse For Folder		×
Select a folder		
/home/yoshihiro/tutorial/rmtprj		
 ✓ [±]→ My Home > [□] app > [□] oralnventory > [□] temp > [□] test > [□] training ∨ [□] tutorial > [□] installcheck > [□] rmtprj 		^
	₩2000	キャンセル(P)
	a+和(<u>A</u>) >>	777 UN(<u>B</u>)



¹⁶ [終了] ボタンをクリックします。

🔤 リモート COBOL プロジェクトの新規作成	_		Х
リモート cobol プロジェクト		1	-\$
ワークスペースまたは外部にリモート COBOL プロジェクトを作成			\$
プロジェクト名: RemoteTutorial			
リモート設定			
接続名: 10.18.11.118	~ 接続	の新規作	成
リモート [/home/yoshihiro/tutorial/rmtprj	~	Brow	se
リモート ロケーションはリモート マシンのプロジェクト パスに設定しなければいけません。			
? <戻る(B) 次へ(N) > 終了(<u>E)</u>	キャンセ	JL

① 下図のようなポップアップが返ってきたら [Do not show this message again] にチェックを入れ、[はい] ボタンをクリックします。

	RSEC2315 ×	:
	Connection 10.18.11.118 has not been secured using SSL. Proceed anyway? ・(ア)Do not show this message again はい(Y) いいえ(N)	
] ファイルヒ 撮無E ァヒァートM 検索 フロシェクトP 実行R ウイントウW ヘルフ(出) ゴ * 🗒 🙆 🍐 株 * O * 💁 * 🎘 e 🎘 e A * · · · · · · · · · · · · · · · · · ·	
	クイック・アクセス 😰 🖬 COBOL	L
	• COBOL 17スオ □ □ • □ • □ • □	
指定したディレクトリに		
COBOL ワモートノロシ ェクトが生成されます。	アウトライン 23 画 プログラム アウト… ■ コピーファイル後… □ □ Micro Focus Build	r
	するアウトラインはありません。 家示するアウトラインはありません。 init:	
	post.build.cfg.New_Configuration:	
	Build finished with no errors. Total time: 0 seconds	
	< >	j



3 プロジェクトにリソースを追加します。

プロジェクトを右クリックし、[インポート] > [インポート] を選択します。

Be COBOL ※ № ナピガータ	- 2	ש-א-וי □ ש לבי ⊂			
🖉 RemoteTutorial [10.18.1	1 110.	/home/uschibire/tutoria 新規作成(N)	>	1	
		⊐Ľ-	Ctrl+C		
	Ē	貼り付け	Ctrl+V	ι.	
	×	削除(D)	削除	ι.	
	<u>.</u>	Remove from Context	Ctrl+Alt+シフト+下へ	ι.	
		移動(V)		ι.	
		名前を変更(M)	F2	ι.	
		ビルド アクション	>		
		指令の確定		ι.	
		コード分析	>		
		インポート(I)	>	Ê,	リモート プロジェクト
	2	エクスポート(O)		寄	ローカル Micro Focus プロジェクトのリモート プロジェクトへの変換
	8	更新(F)	F5	2	Net Express プロジェクトの変換
E アウトライン 23 mm プログラム		プロジェクトを閉じる(S) 毎期係なプロジェクトを閉じる(U)	n	è	インポート(I)

② [General] > [ファイル・システム] を選択し [次へ] ボタンをクリックします。

👜 インポート				
選択 ローカル・ファイル	・システムから既存のフ	プロジェクトヘリソースを	をインポートします。	è
インポート・ソーン	(の選択(<u>S</u>):			
	ral ーカイブ・ファイル ァイル・システム E存プロジェクトをワーク 定	2ペースへ		



③ [参照] ボタンをクリックし、ポップアップするエクスプローラにて「Micro Focus Visual COBOL for Eclipse 自習書」で作成した [BATCHRPT] プロジェクトフォルダを選択し
 [OK] ボタンをクリックします。

🔤 インポート			_		×
ファイル・システム シース・ディレクトリーが無効、または指定されて	ುಕರ್ರ.				
次のディレクトリーから(<u>Y</u>):			~	参照(<u>R</u>).	
ディレクトリーカ	^らインポート	×			
インポート元	のディレクトリーを選択します。				
	 multithread oesql tools tutorial Eclipse .metadata BATCHRPT New_Configuration.bin CallCobHello 	<			
フォルダー(<u>F</u>): BATCHRPT				
新しいフ	ォルダーの作成(<u>N</u>) OK キャンセ	<i>ا</i> لا			



④ [BATCHRPT.cbl] 及び [EMPSEQ.cpy] にチェックを入れ、[終了] ボタンをクリックします。

「man インポート	_		×
ファイル・システム ローカル・ファイル・システムからリソースをインポートします。			
次のディレクトリーから(Y): C:¥work¥tutorial¥Eclipse¥BATCHRPT ~		参照(<u>R</u>)	
> ■ ➢ BATCHRPT ∴ cobolBuild ∴ cobolProj ∴ .cobolProj ∴ .cobolProj …			
タイプをフィルター(I) すべて選択(S) 選択をすべて解除(D)			
インボート先フォルダ(L): RemoteTutorial		参照(<u>W</u>).	
オプション □ 警告を出さずに既存リソースを上書き(<u>O</u>) □ トップ・レベルのフォルダーを作成(<u>C</u>) 拡張 >>(<u>A</u>)			
⑦ < 戻る(B) 次へ(N) > 終了(F)		キャンセノ	ŀ

⑤ ③、④の要領で [Cntl_Card.dat] 及び [Emp_Master.dat] も BATCHRPT のプロジェク
 トフォルダ配下の New_Configuration.bin フォルダ下から COBOL リモートプロジェクト
 に追加します。

「「」 インポート	_		×
ファイル・システム ローカル・ファイル・システムからリソースをインポートします。			
次のデイレクトリーから(Y): C:¥work¥tutorial¥Eclipse¥BATCHRPT¥New_Configuration.bin	~	参照(R)	
New_Configuration.bin			^
Card.dat			
Cntl_Card.pro			
Vie Provedet			~
タイプをフィルター(T) すべて選択(S) 選択をすべて解除(D)			
インポート先フォルダ(L): RemoteTutorial		参照(W)	



【リソース追加後のプロジェクト構成イメージ】



- 4 プロジェクト構成を設定します。
 - ① COBOL エクスプローラにてプロジェクトを右クリックし、[**プロパティ**] を選択します。
 - ② [Micro Focus] > [ビルド構成] > [COBOL] へとナビゲートします。





③ ビルド設定を確認します。

COBOL	< → < <
New Configuration [使用中]	モジュール名はプロジェクトと 同名となります。 くうちょう くうしょう くうしょう くうしょう くうしょう くうしょう くうしょう くうしょう しゅうしょう しゅう しゅうしょう しゅうしょう しゅうしょう しゅうしょう しゅう しゅうしょう しゅうしょう しゅう しゅう しゅうしょう しゅうしょう しゅう しゅう しゅう しゅう しゅう しゅう しゅう しゅう しゅう しゅ
出力の名前: RemoteTutorial	
出力パス: New Configuration.bin	参照
エントリポイント:	
ターゲット設定 ターゲットの種類 単一実行可能ファイル ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	ブラットフォームターゲット ● 32 ビット ○ 64 ビット
プロジェクト中の COBOL フ ログラムを1つの実行形式に めたモジュールが生成されま す。	プ 32 bit モジュールが生成されま こ固 す。 E

④ [プロジェクトの COBOL 設定の上書き] を展開し、[構成の固有な設定を可能にする] にチェ ックを入れます。

ターゲット設定 ターゲットの種類 単一実行可能ファイル 〜 ④ 32 ビット 〇 64 ビット
 プロジェクトの COBOL の設定の上書き (図購成の固有な設定を可能にする(C))



 ⑤ 下へスクロールし、[追加指令] 欄に「ASSIGN(EXTERNAL)」を入力し [OK] ボタンをク リックします。



5 UNIX/Linux 上にリソースが生成されたことを確認します。

① 画面右上の [パースペクティブを開く] アイコンをクリックします。





② [Remote System Explorer] を選択し、[OK] ボタンをクリックします。



③ [Remote Systems] ビューにて、プロジェクトを作成する際に作成した接続を展開します。



④ [SSH] ターミナルを右クリックし、[Launch Terminal] を選択します。

📕 Remote Sy	stem	s 🛙	ବ୍ସ <u>-</u> Team								
				\$ 8	\bigcirc	\$	Q		E	\$₽¢	\bigtriangledown
> 📑 Local	11 11	Q									
> 👪 プロ	セス	·									
> 🔁 SF	TP 73	rtil									
ार ≰ा हा	ᆹᇩ	. 									
13 - L	8	Refre	sh		FS	5					
		Disco	onnect								
		Clear	Password								
	æ	Laun	ch Terminal								
		プロパ	ティ(R)	Alt+	Ente	r					



⑤ プロジェクトディレクトリとして用意したディレクトリの中身を確認します。



⑥ COBOL パースペクティブのアイコンをクリックしてパースペクティブを COBOL に戻します。

– 🗆 🗙

	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
E	COBOL	🔚 Remote System Explorer
_	- 8	📴 ፖኃኑ 🕱 🗖 🗖



第5章 ViewNow X を起動

リモート開発でデバッグする際、ACCEPT 文や DISPLAY 文によるコンソール入出力は X の技術 を用いて、Windows 側に表示させます。そのため、リモート開発にてデバッグ/テスト実行する際 は、Windows 側で X サーバーを起動する必要があります。Micro Focus Visual COBOL for Eclipse をインストールすると Micro Focus ViewNow X という X サーバーのインストーラも併せて配備し ます。Windows 端末上に既に他の X サーバーをインストールしていればそれを利用することも可能 ですが、未インストールの場合はこの ViewNow X をインストールしてリモート開発時に利用するこ とが可能です。本章ではこの ViewNow X をインストール・起動し、続くリモートデバッグ作業に備 えます。

1 ViewNow X をインストールします。

くVisual COBOL for Eclipse のインストールフォルダ> ¥ViewNowX フォルダ配下に ViewNow X のインストーラ ViewNow_X_Server.exe が格納されていることを確認しま す。

📕 🔄 📑 =	アプリケーション ツール ViewNowX			
ファイル ホーム 共有 表:	示 管理			
← → • ↑ <mark>·</mark> • PC • □	ーカル ディスク (C:) » Program Files » Micro	Focus > Visual COBC	L → ViewNowX	~
🛄 デスクトップ 🖈 ^ 名言	ή ^	更新日時	種類	サイズ
븆 ダウンロード 🖈 🔲	ViewNow_X_Server.exe	2015/08/24 22:03	アプリケーション	16,807 KB
🖺 K#1X7F 🔺 🚬 🚬				

【32 bit OS でデフォルトインストールした場合の例】

- 2 ViewNow_X_Server.exe をダブルクリックします。
- 予め取得したライセンスを [License Key] 欄に指定し、[Validate] ボタンをクリックします。

Micro Focus ViewNow X Server 9.6.4 Validation Wizard Select Evaluation or enter a valid license key	E MICRO FOCUS Leading the Evolution
O Evaluation	
License Key:	⊻alidate



④ ライセンスが認証されたことを確認できたら [Next] ボタンをクリックします。



⑤ Setup Guide や Readme を一読する旨の案内や copyright に関する警告が出力されます が、特に問題なければ [Next] ボタンをクリックして進めます。





⑥ ライセンス使用許諾を一読の上、同意できれば、[I accept the terms in the license agreement] を選択し、[Next] ボタンをクリックして進めます。

讇	Micro Focus ViewNow X Server 9.6.4 Wizard X
	Please read the following license agreement carefully. Leading the Evolution
1 4 5 6 7	MPORTANT: LICENSOR IS PROVIDING THIS SOFTWARE FOR YOUR USE SUBJECT TO YOUR A AGREEMENT TO BE BOUND BY THE TERMS AND CONDITIONS SET FORTH BELOW. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT BY CLICKING ON SETUP'S ACCEPT BUTTON, YOU WILL NOT BE ABLE TO USE THE SOFTWARE. BY CLICKING SETUP'S ACCEPT BUTTON YOU ACKNOWLEDGE THAT YOU HAVE READ THIS AGREEMENT, UNDERSTOOD IT, AND AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS.
	END USER LICENSE AGREEMENT
	WHEREAS, Licensor desires to grant to Licensee, and Licensee desires to accept from Licensor, a license to use Licensed Software (as defined herein) upon the terms and Y
	I accept the terms in the license agreement I do not accept the terms in the license agreement
Ins	stallShield <u>Rext > Cancel</u>

⑦ [User Name] 欄や [Organization] 欄に適切な値を入力し、[Next] ボタンをクリックして進めます。

	📸 Micro Focus ViewNow	X Server 9.6.4 Wizard		×
	Customer Information Please enter your inform	ation.		MICRO FOCUS Leading the Evolution"
	User Name: tarot Organization:			
いずれも	5省略は可能です。			
	InstallShield		Next >	Cancel



⑧ デフォルトの [**Typical**] を選択したまま [**Next**] ボタンをクリックします。

😸 Micro Focus V	闄 Micro Focus ViewNow X Server 9.6.4 Wizard							
Setup Type Choose the set	Setup Type Choose the setup type that best suits your needs.							
Please select a	Please select a setup type.							
	The most common application features will be installed. is recommended for most users.	This option						
() <u>C</u> omplete	All program features will be installed. (Requires the mos space.)	st disk						
⊂ Cu <u>s</u> tom	Choose which program features you want installed and will be installed. Recommended for advanced users.	d where they						
InstallShield	< <u>B</u> ack <u>N</u> ext >	Cancel						

⑨ [**インストール**] ボタンをクリックしてインストールを開始します。

B Micro Focus ViewNow X Server 9.6.4 Wizard	×
Ready to Install the Program	
The wizard is ready to begin installation.	Leading the Evolution
Click Install to begin the installation.	
If you want to review or change any of your installation settings, click exit the wizard.	Back. Click Cancel to
InstallShield	
< <u>B</u> ack <u>I</u> nstal	Cancel



① 正常にインストールできた旨のメッセージが返ってきたら [Finish] ボタンをクリックして終 了します。



- 2 ViewNow X サーバーを起動します。
 - スタートメニューより [Micro Focus ViewNow X Server 9.6.4 Control Panel] を選択 します。





② X Control Panel にて [File] メニュー > [New X Server] を選択します。



③ プロパティの設定画面がポップアップされますが、ここではデフォルトのまま [**OK**] ボタンを クリックします。

X Server Properties - New	v X Server 0			?	×
Window	Performance		Keyboard a	& Mouse	
General	Fonts	Graphi	cs	Server	l
Load local X re	sources file:				
			<u>E</u> dit		
			<u>B</u> rowse		
Use <u>h</u> ost acces	s control file				
			E <u>d</u> it		
			B <u>r</u> owse		
	0.000K/F 1				
Use <u>M</u> II-MAG	C-COOKIE-1		Edit		
			Bussie		
I			Browse		
			ОК	キャンセ	zJL



④ [New X Server 0] をダブルクリックします。

🛠 X Control Panel	/
File Edit View Help	
] 🗀 🚸 🌟 🕌 👹 e	S 🔄 👗 🗈 🖻 🗙 🗐 🎗
□	* *
	XDMCP Broadcast New X Server 0

⑤ グラフィックパフォーマンステストに関するダイアログがポップアップされます。初めて起動 する場合は下図の要領でパフォーマンステストを流します。





⑥ 再び Performance Tuning ウィンドウに戻りましたら、[Close] ボタンをクリックします。

💥 X Server Performance Tuning	_	X
Select Test	Drawing Method	Close
Wide Line Drawing	MS-Windows	Cancel
C Accelerated Drawing Mode		<u>H</u> elp
<u>B</u> un Test	Run <u>A</u> ll Tests	

3 ViewNow X サーバーが使用中のポートを確認します。

ポート番号は起動毎に変わることがあります。ポート番号は、Windows のタスクバーにてカーソルをホバーして確認できます。しかし、本書執筆で使用している環境のように下図のような省略表示しかできないこともあります。



その場合、下図のようなかたちでタスクマネージャをより確認します。

👰 タスク マネージャー				_		
ファイル(F) オプション(O) 表示(V)						
プロセス パフォーマンス アプリの履歴	スタートアップ ユーザー 詳	細 サービス				
之前	计能	5%	× 32%	0% ディフク	0% Հարդ-2	
 ViewNow X Server 	17(7)25	0%	4.6 MB	0 MB/秒	0 Mbps	^
🔆 X Server display:2						
1						
	本例ではポート番号 されています。	2 が使用				



第6章 リモートデバッグ

ここまでの作業にて、Windows 上の Eclipse プロジェクトから直接 UNIX/Linux 側に実行形式 を生成させました。本章ではこの生成されたモジュールを UNIX/Linux 上で実行させつつも Windows 上のデバッガでその処理を操作してみます。

- Visual COBOL for Eclipse が閉じている場合は、起動し第4章で使用した
 Eclipse ワークスペースを開きます。
- 2 制御ファイルのメンテナンスをします。

「Micro Focus Visual COBOL for Eclipse 自習書」では最終的に該当する社員情報が見つからなくなるようメンテナンスしました。ここでは初期値に戻し検索条件を有効にします。

① COBOL エクスプローラにて [Cntl_Card.dat] をダブルクリックします。



② 「20110101」に変更します。

炉焦盐	BATCHRPT.cbl	📄 Cntl_Card.dat 🛛
柳朱則	19980101	
編集後	BATCHRPT.cbl	📄 *Cntl_Card.dat 🛛
	20110101	



③ [ファイル] メニューから [保管] を選択し変更を保存します。

GBL	🔤 COBOL - RemoteTutorial/Cntl_Card.dat - Eclipse						
ファ	イル(F) 編集(E)	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)
	新規(N) ファイルを開く(.)			Alt+シフト+	+N >	ॄ • ⊙ •	9⊒ - (2 9 (2) ∧
	閉じる(C) すべて閉じる(L)			Ctrl+ Ctrl+シフト+	·w ·w	I 📄 *Cntl_	Card.dat 🛛
	保管(S)			Ctrl	+S		
B	別名保存(A)						

- 3 デバッグの構成の各種設定項目を指定します。
 - COBOL エクスプローラにて [New_Configuration] 配下に生成されているプロジェクトと
 同名の実行形式を右クリックし [デバッグ] > [デバッグの構成] を選択します。

% C % % 5 ✓ Ø RemoteTu > Ø COBOI > Ø COBOI > Ø COBOI > Ø S > Ø COBOI > Ø Ø > Ø S > Ø S Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø	torial プログ アイル Config TCHR TCHR TCHR	g □ □ [10.18.11.118:/home JJJ uration.bin PT.idy PT.o PT.o.1.tlog	BATCHRPT.cbl BATCHRPT BATCHRPT BATCHRPT BATCHRPT BATCHRPT BATCHRPT BATCHRPT BATCHRPT.cbl BATCHR	S T.cbl B ENTIFICAT OGRAM-ID. SGRAM-ID. Input Outpu	<pre>.23456 ION DIVISION. BATCHRPT. am processes files: Files = Employee Extract File (Sequential) Selection Control Card t File = Employee Yrs Employed Report</pre>
Cntl_C		新規作成(N) 開く(O) アプリケーションから開 コピー 貼り付け	<	> Ctrl+C Ctrl+V	DIVISION. SECTION. EMPLOYEE RECORDS MP-SEQ-FILE ASSIGN TO UT-S-EMPSEQ.
< E 7 🛛 📰	2	刊時(D) Remove from Cont 移動(V) 名前を変更(M)	ext Ctrl+Alt+가	· 則味 7ト+下へ F2	DATE SELECTION CRITERIA N-CNTL-CARD ASSIGN TO UT-S-CNTLCARD.
BATCHRP Solution S	2 2	インポート(I) エクスポート(O) 更新(F)		> F5	
> 🚺 > 📽 Wo > 🗞 Proceo		デバッグ(D) 実行(R) Validate Team		>	↓ 1 Debug on Server Alt+シフト+D,R ■ 2 COBOL アプリケーション Alt+シフト+D,C → デバッグの構成(B)



② [COBOL アプリケーション] をダブルクリックします。



③ [名前] 欄及び [X サーバー(DISPLAY)] 欄へ値を設定します。

, <u></u> -				
名前(N): RemoteTutorialRC	•			
🗟 一般 🦻 ソース 🌄 環	境 🔲 共通(C) 👂 実行時 COBOL スイ	ッチ 🧤 デバッ	グシンボル	👌 動的分析
▼ COBOL プロジェクト(P)		ワークス	ペース内	で実行時
RemoteTutorial	参照	ノデバック つ能か適	グ構成と 当か名前	して識別を指定し
▼ 接続プロパティ		ます。	יייים, יס, ר	
リモートホスト(H):	10.18.11.118			
🗌 リモートホストで cobdeb	ugremote プロセスが受信するポートを指定す	3		
<u>cobdebugremote ポート:</u>	8000			
X サーバー (DISPLAY):	10.18.11.103:2			
▼ 主プログラム				
☑ プログラムはプロジェクトビ	ド構成の一部: New Configuration	~		
New_Configuration.bin/	RemoteTutorial 🛛 🐡	瑶		
▼開始オプション コマンド行引数:	<windows ip="" 側の="">:<viewnow< td=""><td>X サーバーの</td><td>ロポート者</td><td>番号></td></viewnow<></windows>	X サーバーの	ロポート者	番号>
	の形式で入力します。UNIX/Linux できる場合は、IP の部分をデフォル	側から Wind レト値のホス	dows へ ト名にし	名前解決 ても構い
作業ディレクトリ:	ません。		= 0	



④ [適用] ボタンをクリックし変更を保存します。

X サーバー (DISPLAY): 10.18.11.103:	2	
▼ 主プログラム		
☑ プログラムはプロジェクトビルド構成の一部	New Configuration \sim	
New_Configuration.bin/RemoteTutoria	l 参照	
▼ 開始オプション		~
IV /N(TEIW)		
	🎽 適用(Y)	前回保管した状態に戻す(V)

⑤ [環境] タブをクリックします。

ſ	
	名前(N): RemoteTutorialRC
	□ 一般 🤟 ソータ 🚾 環境 🔄 共通(C) 🔎 実行時 COBOL スイッチ 🦭 デバッグシンボル 🗳 動的分析
	 ▼ COBOL プロジェクト(P)

⑥ [追加] ボタンをクリックします。

□ 共通(C) ▶ 実行	5時 COBOL スイッ:	チ 🧤 デバッグシンボル	3 動的分析	
	値			✔ 追加(A)
				編集(E)
				削除(R)

- ⑦ 下記のように入力し [OK] ボタンをクリックします。
 - [変数] 欄 dd_EMPSEQ
 - [値] 欄 Emp_Master.dat までのフルパス

【入力例】

國愛	yを追加 ×	
変数:	dd_EMPSEQ	
值:	home/yoshihiro/tutorial/rmtprj/Emp_Master.dat	
`		-1
	OK キャンセル	



- ⑧ ⑤、⑥の要領で下記のエントリも追加します。
 - [変数] 欄 dd_CNTLCARD
 - [値] 欄 Cntl_Card.dat までのフルパス

【入	カ	例]
	//	ניעו	

d_CNTLCARD nome/yoshihiro/tutorial/rmtprj/Cntl_Card.dat	
nome/yoshihiro/tutorial/rmtprj/Cntl_Card.dat	:
OK キャンセル	
	OK キャンセル

- ⑨ 更に同様に下記のエントリも追加します。
 - [変数] 欄 dd_HIRERPT
 - [値] 欄 <プロジェクトディレクトリ>/Hire_Report.dat



國変数	xē追加 ×
変数:	dd_HIRERPT
值:	home/yoshihiro/tutorial/rmtprj/Hire_Report.dat
`	
	OK キャンセル



4 デバッグ実行を開始します。

前のステップで指定したデバッグ構成ウィンドウにて [**適用**] ボタンに続き [**デバッグ**] をクリッ クしデバッグ実行を開始します。

 デバッグ構成 構成の作成、管理、および COBOL プログラムをデバッグします 	美行		×
 □ □<th>名前(N): RemoteTutorialRC</th><th>_ [] 共通(<u>C</u>) 🔎 実行時 COBOL スイッチ 🧤 デバッ</th><th>グシンボル 💪 動的分析</th>	名前(N): RemoteTutorialRC	_ [] 共通(<u>C</u>) 🔎 実行時 COBOL スイッチ 🧤 デバッ	グシンボル 💪 動的分析
 C Android Application 	変数 dd_EMPSEQ dd_CNTLCARD dd_HIRERPT	值 /home/yoshihiro/tutorial/rmtprj/Emp_Master.dat /home/yoshihiro/tutorial/rmtprj/Cntl_Card.dat /home/yoshihiro/tutorial/rmtprj/Hire_Report.dat	<u>追加(A)</u> 編集(<u>E</u>) 削除(<u>R</u>)
■ 新規構成 、 フィルターー致: 28 / 28 項目		適用(<u>Y</u>) デ/	前回保管した状態に戻す(<u>い</u>) パッグ(<u>D</u>) 閉じる

- 5 Eclipse 上のデバッガを使ってデバッグします。
 - パースペクティブの切り替えに関するメッセージに関しては [はい] ボタンをクリックしてデ バッグパースペクティブへ切り替えます。





最初の COBOL 行の実行前で処理が一時停止しています。

The and アメリト 間 後端 アレンロビット アレース マイン マンロ マンロ マンロ マンロ マンロ マンロ マンロ マンロ マンロ マン	「呵」デバッグ - RemoteTutorial/BATCHRPT.cbl - Eclipse	実行(D) ウィンドウ(MA) A II	-f(H)		-	o x
・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・				- 🖂 🚸 - 🔿	-0-0	
② ・③ ・⑤ ・			2- 2-			
サデバッグ 23 #3 Servers ※ 「 ● C3 た 1 ブリケーション」 ※ 「 ● Renote Tutorial(C (C080L アブリケーション) ※ ● C080L スレッド6874(一等停止) ■ BATCHRPT: 簡要は情報感気し(学: 142) > ● C080L スレッド6873(一等停止) ■ BATCHRPT: E12(142) > ● C080L スレッド6873(一等停止) ■ BATCHRPT: E12(142) ※ ● C080L スレッド6873(一等停止) ■ BATCHRPT: E12(142) ● C080L スレッド6873(一等停止) ● E12(142) ● BATCHRPT: E12(142) ● C080L スレッド6873(- 等停止) ● BATCHRPT: E12(142) ●	約 ▼ 卻 ▼ ♥		クイック・アクセス	😤 🌆 COBOL 🕴	Remote System Explorer	参 デバッグ
 ■ Remote Lutorial(ClOBOL アジリテ-2e2) ● COBOL スレッド6874(一等停止) ● COBOL スレッド6873(一等停止) ● EATCHRPT: 調素だ結果落が気し(音: H2) ● COBOL スレッド6873(一等停止) ■ EATCHRPT: 調素だ結果落が気し(音: H2) ● COBOL スレッド6873(一等停止) ■ EATCHRPT.tol ※ ● Cott_Card.dat ● FIRST, VERIFY EMPLOYEE'S HIRE DATE IS ON OR BEFORE DATE ● FIRST, VERIFY EMPLOYEE'S HIRE DATE IS ON OR BEFORE DATE ● FIRST, VERIFY EMPLOYEE'S HIRE DATE IS ON OR BEFORE DATE ● FIRST, VERIFY EMPLOYEE'S HIRE DATE IS ON OR BEFORE DATE ● COBOL TAU-9F-HIRE < CONTILUE ELSE GO TO 30000-EXIT ENO-FIC ● FORMAT REPORT DETAIL LINES FROM EMPLOYEE RECORD. ● FORMAT REPORT DETAIL LINES FROM EMPLOYEE RECORD. ● DO TID Application ■ DI TID Application ■ DI TID Application ■ BATCHRETE ■ 142, 12 	なデパッグ ☆ ぷ Servers	🍇 ୶ 🔿 🕸	🌮 ▽ 🗆 🗍 (×)= 変数	🛿 💁 ブレークボイ	(ント 🏝 📲	B ∼ ⊓ B
② COBOL デバジボhome/yoshihio/tutorial/metg/Mew_Configuration.bin/Remote/Lutorial (一時停止) ③ COBOL スレッド6073 (一時停止) ③ BATCHRPT: 罰素治超感形なし(許 142) > ② COBOL スレッド6073 (一時停止) ③ BATCHRPT: 罰素治超感形なし(許 142) > ③ COBOL スレッド6073 (一時停止) ③ BATCHRPT: 罰素治超感形なし(許 142) ③ BATCHRPT: 「罰素治超感活なし(許 142) ③ BATCHRPT: 「罰素治超感活なし(許 142) ③ BATCHRPT: Di またいのののののののののののののののののののののののののののののののののののの	✓ ■ RemoteTutorialRC [COBOL アプリケーション]		名前		値	
COBOL AU-Philod 4(一時作) ★ Cobol AU-Philod 4(一時) ★ Cobol AU-Philod 4((一前) ★	 COBOL デバッガ:/home/yoshihiro/tutorial/rmtprj/New_ 	Configuration.bin/RemoteT	itorial (一時停止)			
DONAL REPORT DETAIL LINES FROM EMPLOYEE RECORD. DOMS	✓ ● COBOL スレット:6874 (一時停止) ■ DATCURPT: 第十たけの支払たし (年, 142)	↑				
ま行するモジュールは UNIX/Linux 上にありますが、Windows 上のデバッ プでデバッグしています。	■ BAICHKP1: 即または技法がなし(17:142) ▲ COBOL 71mk 6873 (一時停止)		4			>
ま行するモジュールは UNIX/Linux 上にありますが、Windows 上のデバッ ガでデバッグしています。 BATCHRPTcbl ※ Cott_Card.dat Easter FiltsT, VERIFY EMPLOYEE'S HIRE DATE IS ON OR BEFORE DATE CONTINUE ELSE GO TO 3000-EXIT ENO-IF. FORMAT REPORT DETAIL LINES FROM EMPLOYEE RECORD. FORMAT REPORT DETAIL LINES FROM EMPLOYEE RECORD. Saved Filter Saved Filter Lu: Time PID TID Application PID TID Application BATCHRPT HAD TID Application HAD TID HAD HAD <p< td=""><td>) The concert (1,1913 (1,1912)</td><td></td><td></td><td></td><td></td><td></td></p<>) The concert (1,1913 (1,1912)					
BATCHRPT.cbl ※ Cht_Card.dat Licab Dist af が、Windows 上のデバッ Dist CHRPT.cbl ※ Dist CHRPT.cbl ※ Dist Christer Minintx Licab Dist af が、Windows 上のデバッ が # BATCHRPT が # BATCHRPT が # BATCHRPT ジ Dist Division ジ # BATCHRPT ジ # Dist Division ジ # BATCHRPT ジ # Dist Division ジ # File Section ジ # Dist Division ジ # Procedure Division ジ # Dist Division ジ #		宝行するモジ	$- \frac{1}{2} + $	linux		. Ç
■ BATCHRPT.cbl ☆ Cntl_Card.dat 上にとおりますのか、Windows 上のテイベッ ガでデバッグしています。 ■ BATCHRPT.cbl ☆ ガでデバッグしています。 ■ BATCHRPT.cbl ☆ ■ BATCHRPT.cbl ↓ ■ BATCHRPT.cbl				o T'u'		
● BATCHRPT.cbl → ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	🖻 BATCHRPT.cbl 🔀 📄 Cntl_Card.dat	上にめります	ひ、Windows 上	のテハツ	🔡 アウトライン 🛿 👘 🛛	
Image: Solution of the second sec		ガでデバッグ	しています。		✓	
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. *** FORMAT REPORT DETAIL LINES FROM EMPLOYEE RECORD. Saved Filter All messag L Time PID TID Application *** ##3\A TIE #4\ 142:12					🗸 🌄 Data Division	
FIRST, VERIFY EMPLOYEE'S HIRE DATE IS ON OR BEFORE DATE PASSED IN CONTROL CARD. IF EMPSEC-DATE-OF-HIRE <= CNTL-DATE CONTINUE ELSE GO TO 3000-EXIT END-IF. FORMAT REPORT DETAIL LINES FROM EMPLOYEE RECORD. IDUGCAT Source From Employee Record Control	***			^	V 🚏 File Section	
PASSED IN CONTROL CARD. PASSED IN CONTROL CARD. IF EMP-IRE-RPT IF EMP-IRE-RPT EUSE GO TO 3000-EXIT END-IF. *** FORMAT REPORT DETAIL LINES FROM EMPLOYEE RECORD. Saved Filter: All message L Time PID TID Application C	<pre>* FIRST, VERIFY EMPLOYEE'S HIRE DATE</pre>	IS ON OR BEFORE DATE			> EMP-SEC	Q-FILE
IF EMPREC-DATE-OF-HIRE <= CNTL-DATE CONTINUE ELSE GO TO 3000-EXIT END-IF. IF EMPREC-DATE-OF-HIRE <= CNTL-DATE CONTINUE ELSE GO TO 3000-EXIT END-IF. Image: Source of the two of the two of the two of two	<pre>* PASSED IN CONTROL CARD. ****</pre>					-CARD
CONTINUE ELSE GO TO 3000-EXIT END-IF.	IF EMPREC-DATE-OF-HIRE <= CNTL-DA	TE			Working-Sto	rage Section
ELSE GO TO 3000-EXIT END-IF. ■ FORMAT REPORT DETAIL LINES FROM EMPLOYEE RECORD. ■ DVY-ル SS @ 97.7 ■ D LogCat SS Saved Filter: All message L Time PID TID Application ■ B12,24 TIE ■ 142:12	CONTINUE		•		> S Procedure Divisi	on
END-IF: FORMAT REPORT DETAIL LINES FROM EMPLOYEE RECORD. Saved Filter: All message L Time PID TID Application END-IF: TID END-IF: TID END-IF:	ELSE 60 TO 3000-EXIT					
	END-IF.					
FORMAT REPORT DETAIL LINES FROM EMPLOYEE RECORD.						
Cooker KLPONT BLIALE LINES THOM EMPEDIELE KLEOND. Cooker KLEOND.	* FORMAT REPORT DETAIL LINES FROM FM	DLOVEE RECORD				
く	***	PLOTEL RECORD.		<u> </u>		
DMS LogCat ② L Time PID TID Application C DMS L Time PID TID Application C DMS L Time PID TID Application C DMS L Time PID TID Application C DMS L Time PID TID Application C L Time PID TID Application L Time PID TID Application C L Time PID TID Application L Time PID TID Appl	<			>		
Description De						
Saved Filter: All message L Time PID TID Application C C C C C C C C C C C C C C C C C C C	DogCat 🔀		ארעב 🔄 🖾 ארעב 🖳		🖹 🛃 🛛 🖬 🖬	· 📑 🕶 🗖
Saved Filter: All messag L Time PID TID Application <			DDMS			
All messag L Time PID TID Application く ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	Saved Filter: Search for messages. Accepts Java rege: verbos	e 🗸 🖬 🖳 🗖 🕹	[2015-09-09 15:10:58	- DDMS] DDMS f	iles not found: C:\Us	ers\Public\/ ^
	All messag	Application				
く > 申念込み可能 挿入 142:12		Application				
						\sim
⇒き込み可能 挿入 142:12		>	<			>
		書き込み可能	挿入 142	: 12		

② [2000-MAIN-PROCESSING] 段落の最初の READ 文にカーソルを合わせ、右クリックから
 ⑤ [指定行まで実行] を選択します。

Θ	2000-MAIN-PROCESSING.			
	READ EMP-SEQ-FILE INTO	E	貼り付け(P)	Ctrl+V
	AT END MOVE 'Y' TO		クイック・フィックス(Q)	Ctrl+1
	IF NOT-AT-EOF		右ヘシフト(S)	
	PERFORM 3000-PROCES	5	左ヘシフト(H)	
	END-IF.	1 2	番号を付け直す	
Θ	2000-EXIT.	1 2-36	番号を削除する	
EXIT.		プロファイル(P)	>	
Θ	3000-PROCESS-RECORD.		デバッグ(D)	>
	* FIRST, VERIFY EMPLOYEE'S		実行(R)	>
	* PASSED IN CONTROL CARD.		Validate	
			置換対象(L)	>
<	TE EPERECTUALETCE TIRE		COBOL コピー表示で開く	F4
LogCat	x		コンテンツ・アシスト	Ctrl+スペース
-			コード分析	>
aved Filte	ers 🕂 💳 Search for messages. Acc	e	Team	> verbose 🗸
All messa	iges (no filti L Time		比較	> Text
		⇒]	指定行まで実行(L)	Ctrl+R
		0		411 542.0



カーソル位置まで処理が進みます。



③ **F5**を打鍵し、READ 文を実行します。







④ 条件付きブレークポイント機能を確認します。



【条件付きブレークポイントの例】



設定後、F8 を打鍵しますと、設定した直後から5回目の READ 文のヒットでデバッガが一時 停止します。



⑤ ウォッチ式(監視式)機能を確認します。

ブレークポイントをダブルクリックし、ブレークポイントを解除します。



EMP-RECORD-IO-AREA の値が変わる度にデバッガを一時停止させます。EMP-RECORD-IO-AREA を選択し右クリックから [検査] を選択します。

BATCHRPT	Col 🛛		実行(R)	>
BATC	СНRРТ.сЫ) *А·1·В·····2····3··· 4 ···- Д ·		Validate 置換対象(L)	>
\$	READ EMP-SEQ-FILE INTO EMP-R AT END MOVE 'Y' TO EOF- IF NOT-AT-EOF PERFORM 3000-PROCESS-RECC END-IF.		定義位置へ COBOL コピー表示で開く コンテンツ・アシスト 参照	F3 F4 Ctrl+スペース Ctrl+シフト+G
Θ	EXIT. 3000-PROCESS-RECORD.		コード分析 Team	>
<		ьT		>
🗊 LogCat 🛛		⇒1 ♠	指定行まで実行(L) 実行点をリセット	Ctrl+R Alt+F12,R
Saved Filte	Search for messages. Accepts Jav	Q	検査	Ctrl+シフト+I



[式ビューに追加] をクリックします。



以降、F8 を打鍵すると EMP-RECORD-IO-AREA の値が変わる度にデバッガが一時停止します。





⑥ デバッガの動作が確認できましたら、デバッガが終了するまで F8 を連続で打鍵します。

デバッガが終了した旨を [**デバッグ**] ビューより確認できます。



6 COBOL パースペクティブに戻します。

画面右上の COBOL パースペクティブアイコンを選択します。

	_		×
🕸 • 💊 • 9a • 🔊 🗁 🖋 • 🖢 - 🖓 •	% р (() -	> •
7セス 😰 🔤 COBOL 🔚 Remote System Expl	orer	参 デバッ	ヷ
ス 🗣 ブレークポイント 🗶	⇒ti [

- 7 生成された帳票を確認します。
 - ① COBOL エクスプローラにて Hire_Report.dat が生成されていることを確認します。





② COBOL エクスプローラ中の Hire_Report.dat をダブルクリックします。

BATCHRPT.cbl	Report.dat 🛛			
Program: BATCHRPT	Years Employed Repo	rt	09/09/2015	<u>^</u>
***** 2011年 1月	1日以前に入社した社員一覧		10.04.24	
部署名 社員名	社員番号	入社日	雇用年数	
宮枝物務部 佐藤木 空枝物務部部 山田藤 技総常新部部 山田藤 大総営術部部 中田 技総営業 竹部部 中村 大統 総営 業部部 中村 本 本 十 村 本 本 十	隆 1111111-3 尚之 222222-6 直美 3333333-9 洋一 444444-2 弘子 555555-5 貫弘 6666666-8 慎司 777777-1 悦子 8888888-4 牽 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	17 16 15 14 13 12 11	
***** 処理レコー	ド件数: 9			~
<				2

「Micro Focus Visual COBOL for Eclipse 自習書」で確認したのと同じ帳票が生成されてい ることが確認できます⁶:

⁶ 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] を選択し変 更できます。



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/