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」を入力します。
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。
このバージョンの次の製品を使用するには : Micro Focus Visual COBOL Development Hub 2.3 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

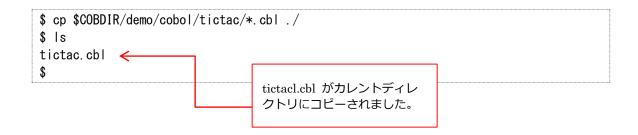
> ライブラリ探索パス³: \$COBDIR/lib

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。



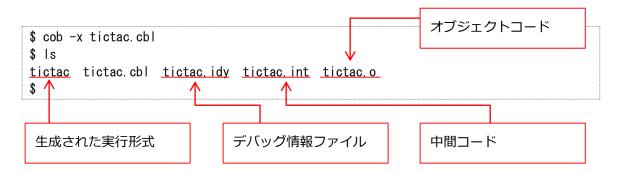
4 製品同梱サンプルをコピーします。

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へ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へ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】

① 実行形式を実行

\$./tictac

To select a square type a number between 1 and 9

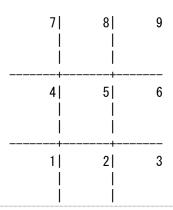
Shall I start ?

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 number between 1 and 9 Please select an empty square $\mathbf{0}$



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。



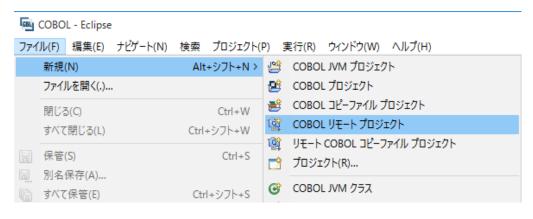
第4章 COBOL リモートプロジェクトの作成

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。

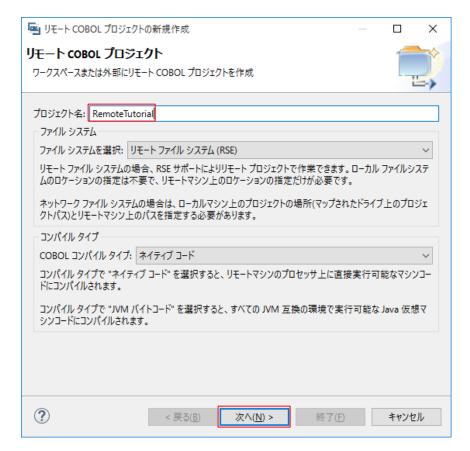
ワークスペースの指定は特にありません。

- 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。

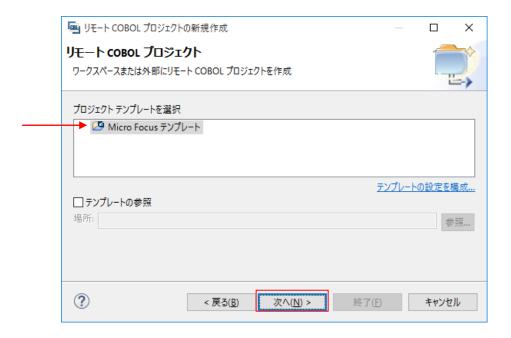




② [プロジェクト名] 欄にプロジェクト名を指定し、[次へ] ボタンをクリックします。 ファイルシステム及びコンパイルタイプはデフォルトのままにしておきます。

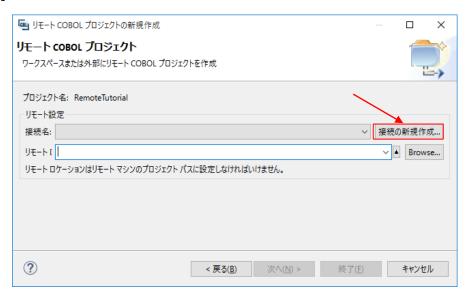


③ プロジェクトテンプレートはデフォルトの [Micro Focus テンプレート] を選択し [次へ] ボタンをクリックします。

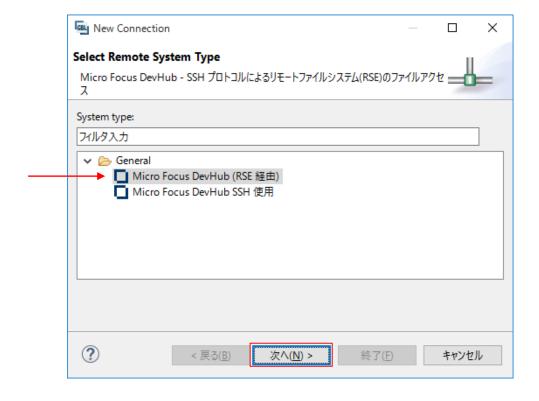




④ [接続の新規作成] ボタンをクリックします。

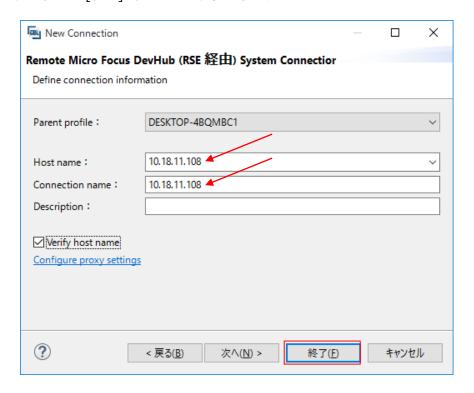


⑤ [Micro Focus DevHub(RSE 経由)] が選択されていることを確認し、[次へ] ボタンをクリックします。

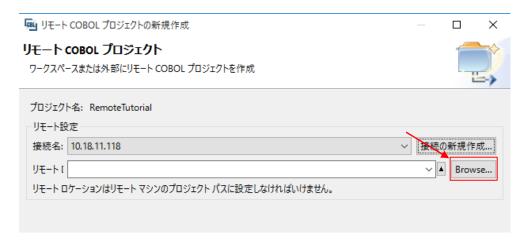




⑥ Windows 側にて UNIX/Linux サーバーの名前解決できるのであれば [Host name] 欄にそのホスト名を入力します。名前解決できない場合は、[Hostname] 欄にはそのサーバーの IP アドレスを指定します。[Connection name] 欄は自動で [Host name] 欄の値がコピーされます。指定が終わりましたら [終了] ボタンをクリックします。

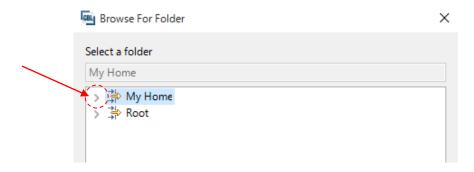


⑦ 「Browse] ボタンをクリックします。





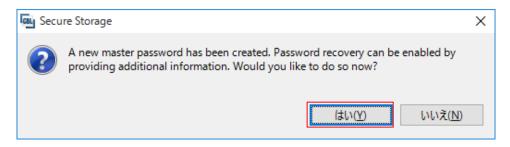
⑧ 「My Home」の左の展開アイコンをクリックします。



⑨ 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。

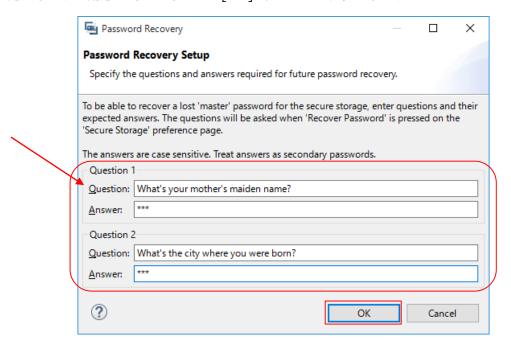


⑩ [Secure Storage] に関するポップアップが返ってきたら [はい] ボタンをクリックします。

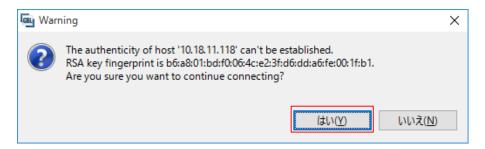




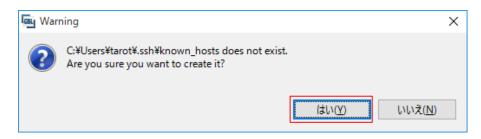
Password Recovery 用の質問と回答を登録します。ここでは、例として母親の旧姓と出生した都市名を記入します。用意ができましたら、[OK] ボタンをクリックします。



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。

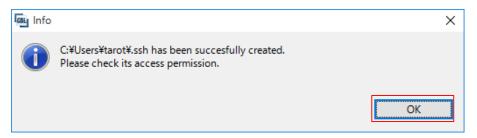


⑤ 下図のような警告にも [はい] ボタンをクリックします。

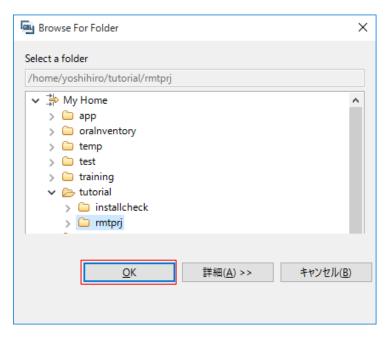




④ 下図のような .ssh が生成できた旨のメッセージが返ってきます。内容を確認し [**OK**] ボタンをクリックします。

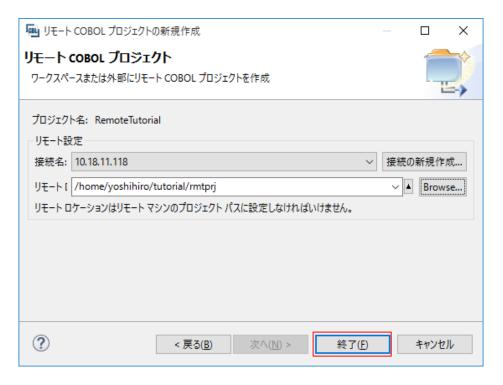


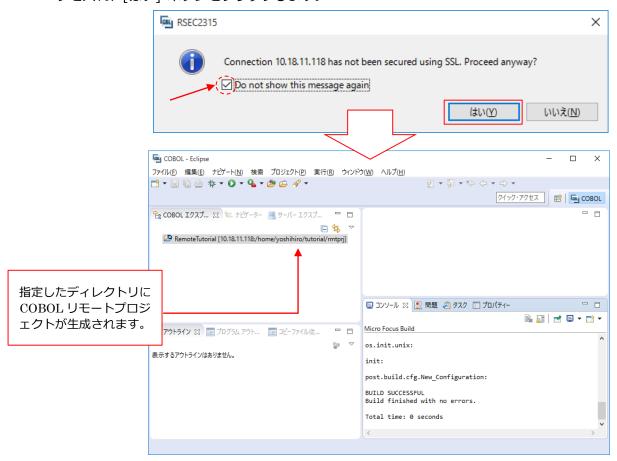
⑤ UNIX/Linux 側でソースや生成されるモジュール等を格納するプロジェクトディレクトリとして利用するディレクトリをツリーで選択し、[**OK**] ボタンをクリックします。





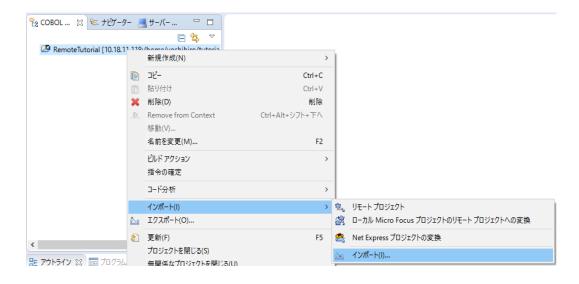
(f) 「終**了**] ボタンをクリックします。



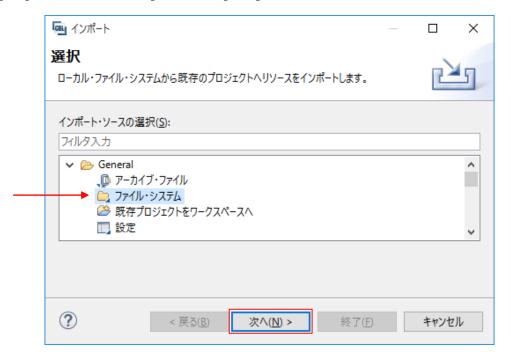




- 3 プロジェクトにリソースを追加します。
 - ① プロジェクトを右クリックし、「インポート] > 「インポート] を選択します。

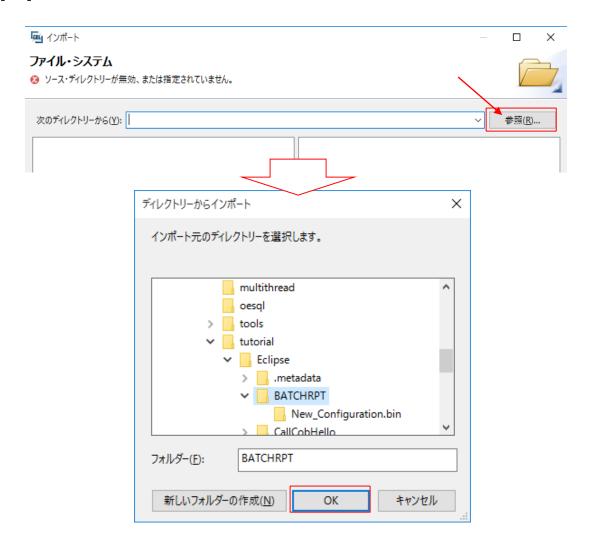


② [General] > [ファイル・システム] を選択し [次へ] ボタンをクリックします。



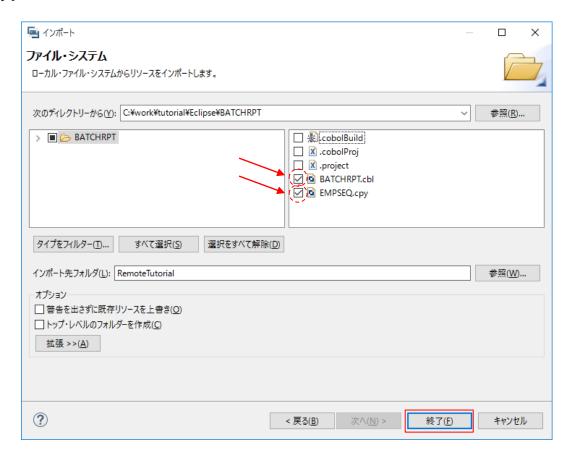


③ [参照] ボタンをクリックし、ポップアップするエクスプローラにて「Micro Focus Visual COBOL for Eclipse 自習書」で作成した [BATCHRPT] プロジェクトフォルダを選択し [OK] ボタンをクリックします。

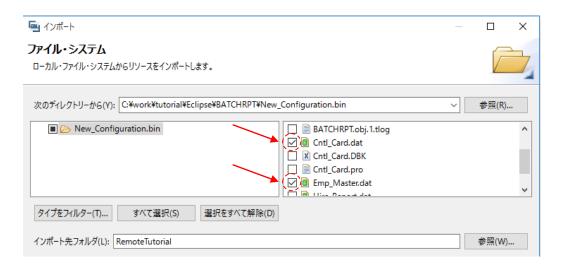




④ [BATCHRPT.cbl] 及び [EMPSEQ.cpy] にチェックを入れ、[終了] ボタンをクリックします。

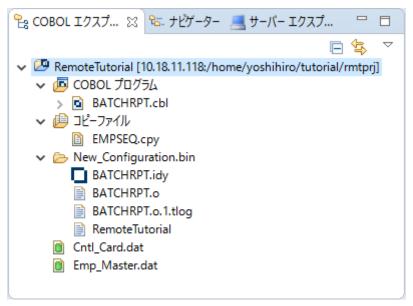


⑤ ③、④の要領で [Cntl_Card.dat] 及び [Emp_Master.dat] も BATCHRPT のプロジェクトフォルダ配下の New_Configuration.bin フォルダ下から COBOL リモートプロジェクト に追加します。

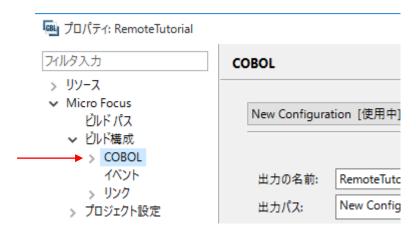




【リソース追加後のプロジェクト構成イメージ】

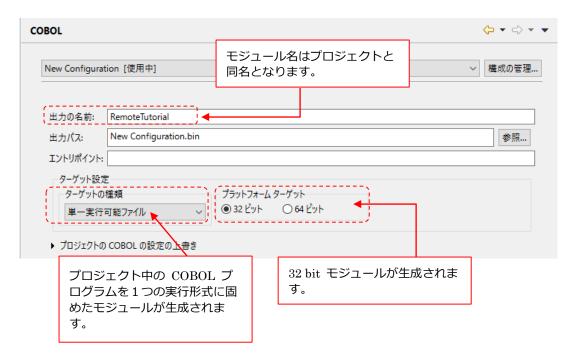


- 4 プロジェクト構成を設定します。
 - ① COBOL エクスプローラにてプロジェクトを右クリックし、[プロパティ] を選択します。
 - ② [Micro Focus] > [ビルド構成] > [COBOL] へとナビゲートします。





③ ビルド設定を確認します。

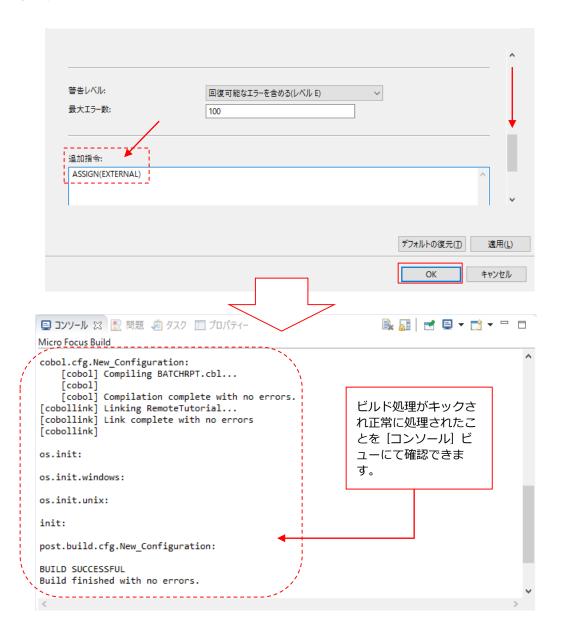


④ [プロジェクトの COBOL 設定の上書き] を展開し、[構成の固有な設定を可能にする] にチェックを入れます。





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。

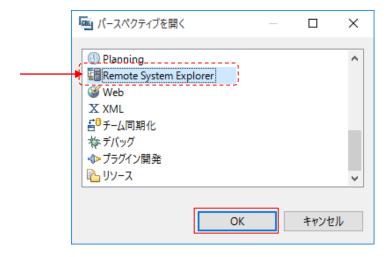


- 5 UNIX/Linux 上にリソースが生成されたことを確認します。
 - ① 画面右上の [パースペクティブを開く] アイコンをクリックします。

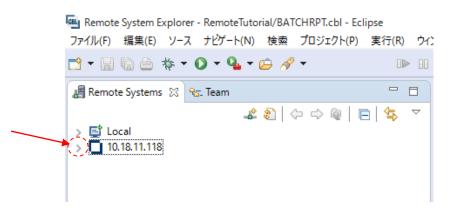




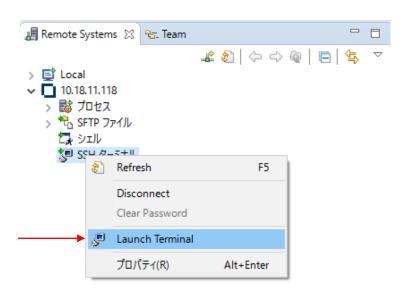
② [Remote System Explorer] を選択し、[OK] ボタンをクリックします。



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。

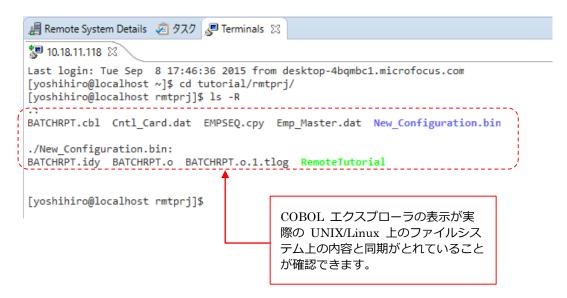


④ [SSH] ターミナルを右クリックし、[Launch Terminal] を選択します。





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。



×

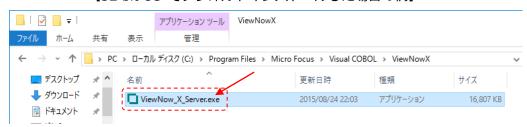


第5章 ViewNow X を起動

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。

① **<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。



【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。
- ③ 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。

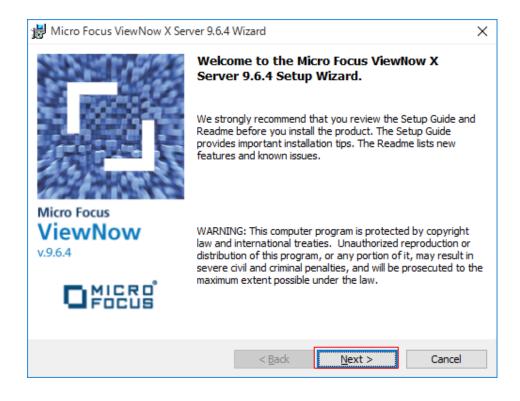




④ ライセンスが認証されたことを確認できたら [Next] ボタンをクリックします。

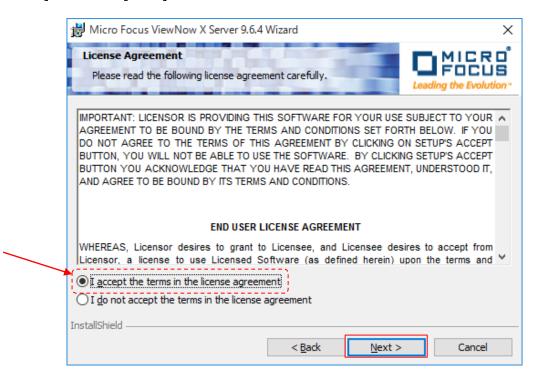


⑤ Setup Guide や Readme を一読する旨の案内や copyright に関する警告が出力されますが、特に問題なければ [Next] ボタンをクリックして進めます。

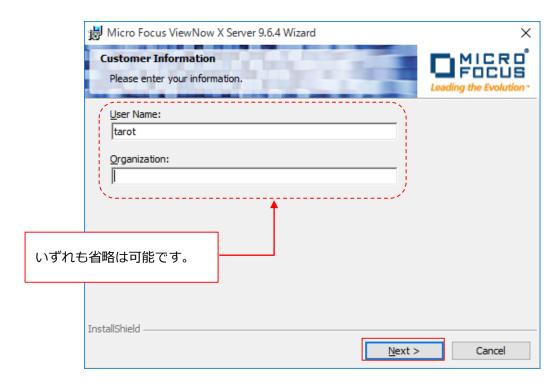




⑤ ライセンス使用許諾を一読の上、同意できれば、[I accept the terms in the license agreement] を選択し、[Next] ボタンをクリックして進めます。

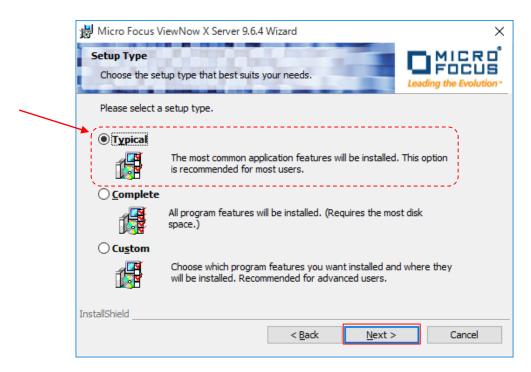


② [User Name] 欄や [Organization] 欄に適切な値を入力し、[Next] ボタンをクリックして進めます。

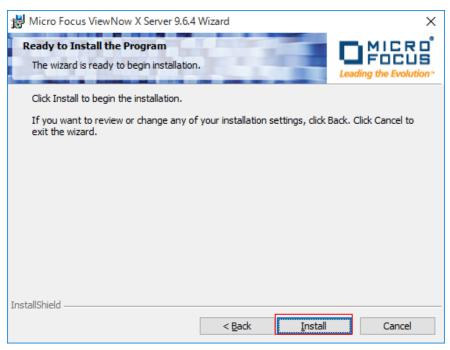




⑧ デフォルトの [Typical] を選択したまま [Next] ボタンをクリックします。



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。

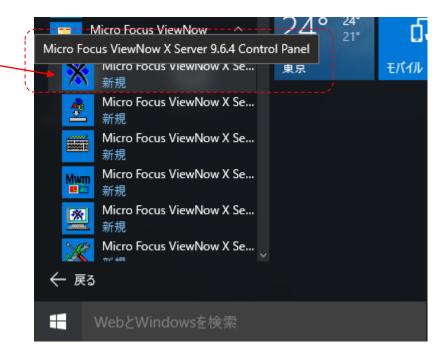




① 正常にインストールできた旨のメッセージが返ってきたら [Finish] ボタンをクリックして終了します。

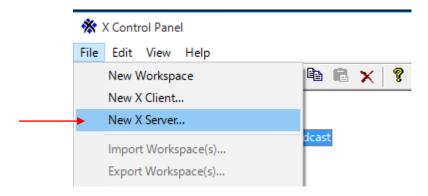


- 2 ViewNow X サーバーを起動します。
 - ① スタートメニューより [Micro Focus ViewNow X Server 9.6.4 Control Panel] を選択します。

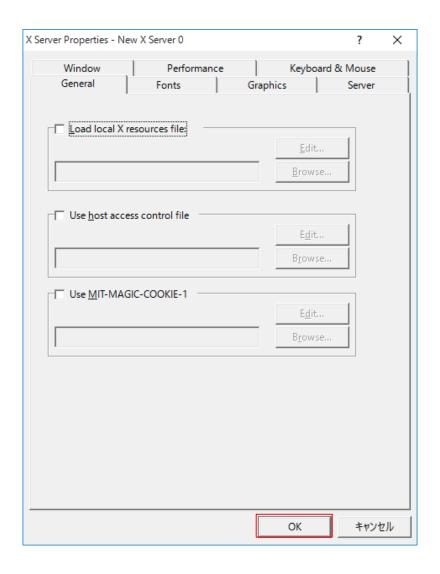




② X Control Panel にて [File] メニュー > [New X Server] を選択します。

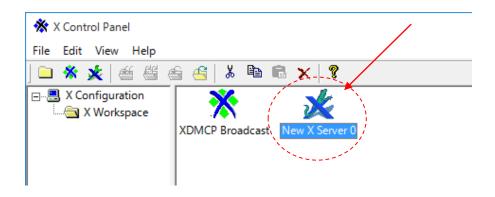


③ プロパティの設定画面がポップアップされますが、ここではデフォルトのまま [**OK**] ボタンを クリックします。

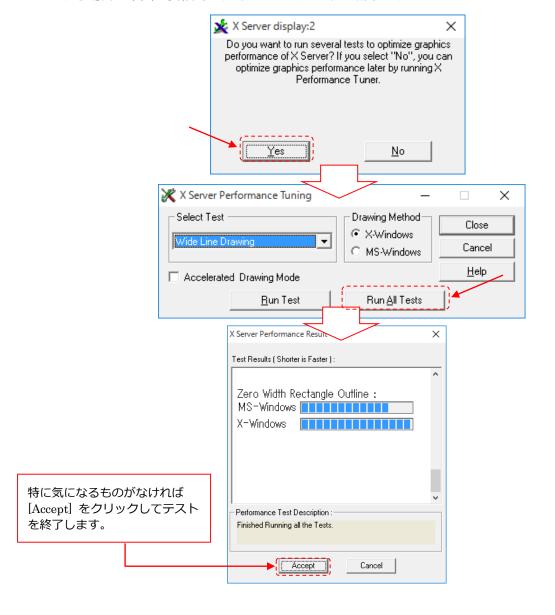




④ [New X Server 0] をダブルクリックします。

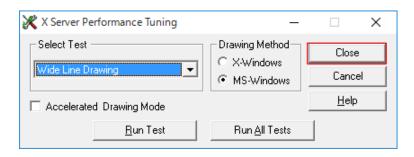


⑤ グラフィックパフォーマンステストに関するダイアログがポップアップされます。初めて起動 する場合は下図の要領でパフォーマンステストを流します。



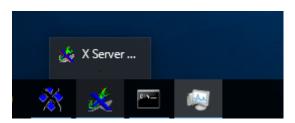


⑥ 再び Performance Tuning ウィンドウに戻りましたら、[Close] ボタンをクリックします。



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。





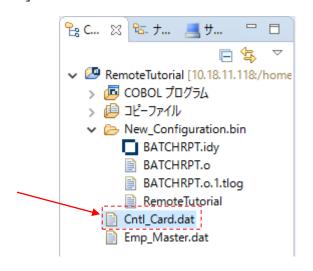
第6章 リモートデバッグ

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。

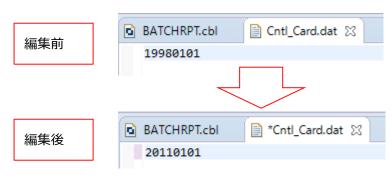
- 1 Visual COBOL for Eclipse が閉じている場合は、起動し第4章で使用した Eclipse ワークスペースを開きます。
- 2 制御ファイルのメンテナンスをします。

「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。

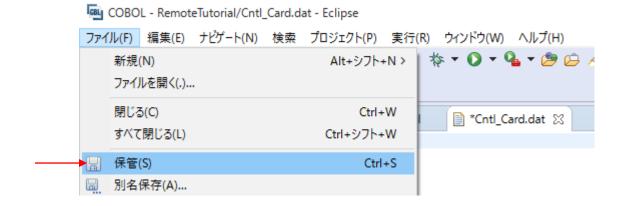


② 「20110101」に変更します。



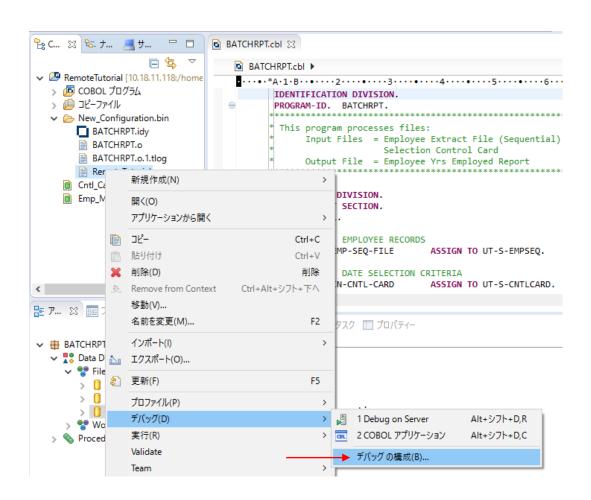


③ 「ファイル] メニューから [保管] を選択し変更を保存します。



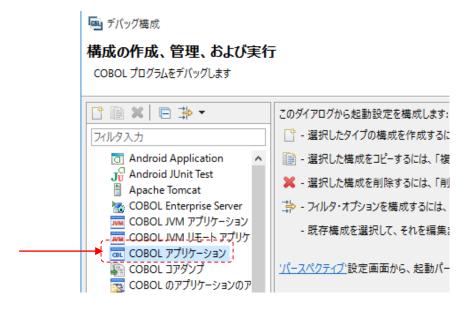
3 デバッグの構成の各種設定項目を指定します。

① 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。

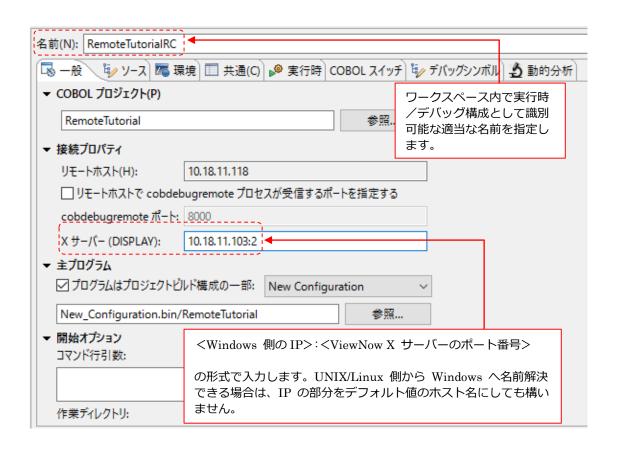




② [COBOL アプリケーション] をダブルクリックします。

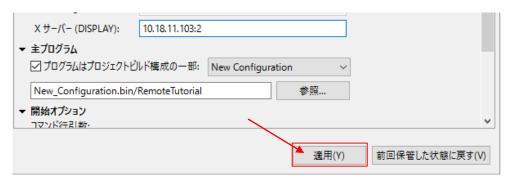


③ 「名前] 欄及び 「X サーバー(DISPLAY)] 欄へ値を設定します。





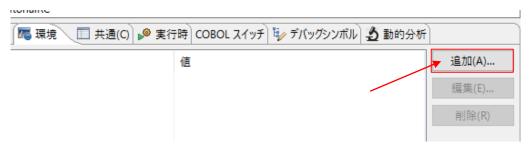
④ 「適用」ボタンをクリックし変更を保存します。



⑤ [環境] タブをクリックします。



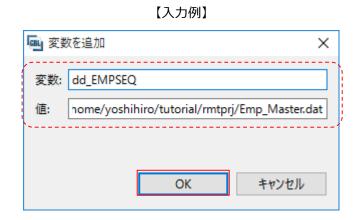
⑥ 「追加」ボタンをクリックします。



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



- 36 -



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

【入力例】

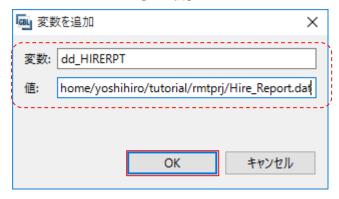


⑨ 更に同様に下記のエントリも追加します。

[変数] 欄 dd_HIRERPT

[値] 欄 〈プロジェクトディレクトリ〉/Hire_Report.dat

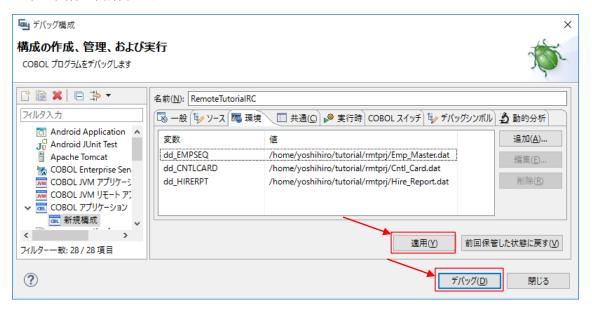
【入力例】



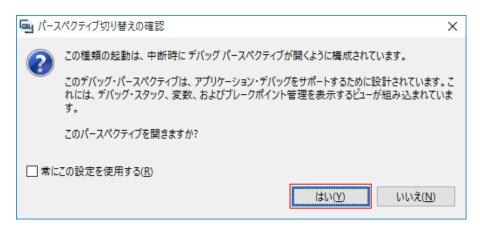


4 デバッグ実行を開始します。

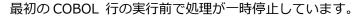
前のステップで指定したデバッグ構成ウィンドウにて [**適用**] ボタンに続き [**デバッグ**] をクリックしデバッグ実行を開始します。

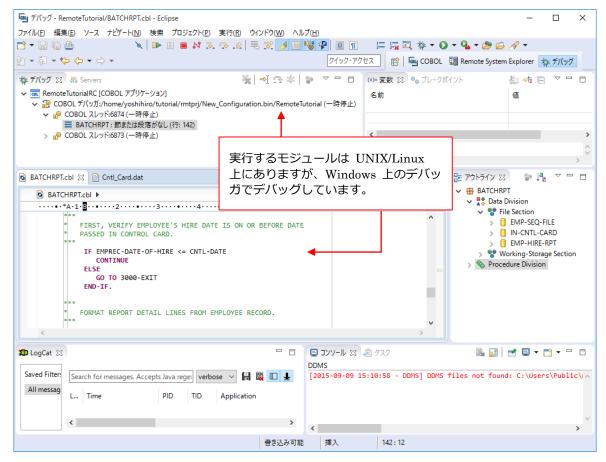


- 5 Eclipse 上のデバッガを使ってデバッグします。
 - ① パースペクティブの切り替えに関するメッセージに関しては [**はい**] ボタンをクリックしてデバッグパースペクティブへ切り替えます。

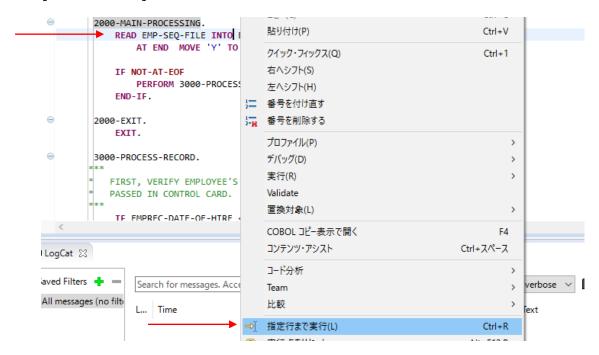






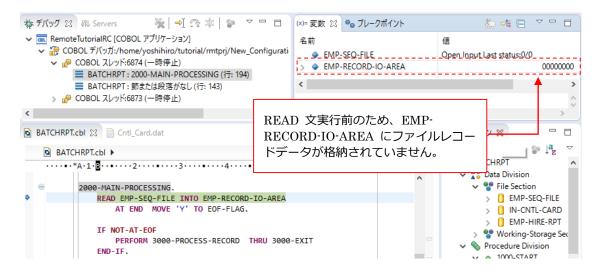


② [2000-MAIN-PROCESSING] 段落の最初の READ 文にカーソルを合わせ、右クリックから [指定行まで実行] を選択します。

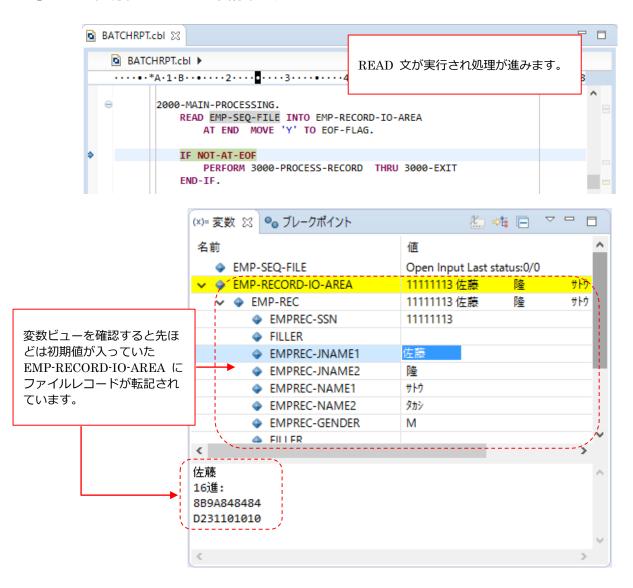




カーソル位置まで処理が進みます。



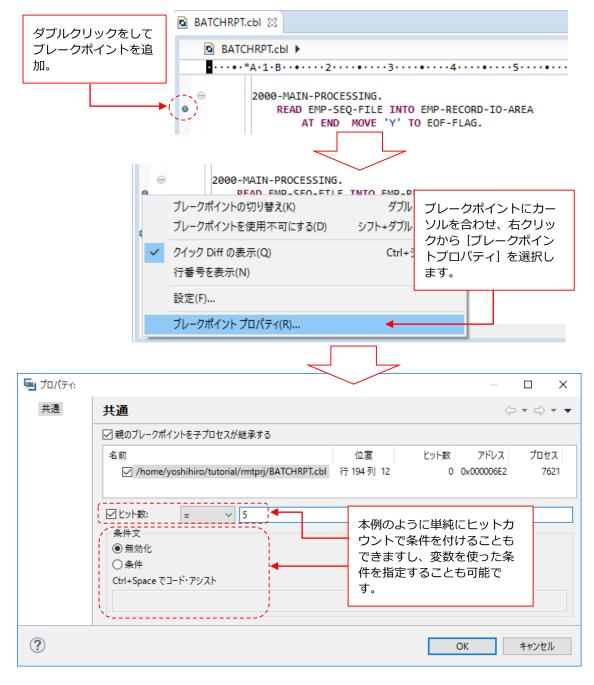
F5 を打鍵し、READ 文を実行します。





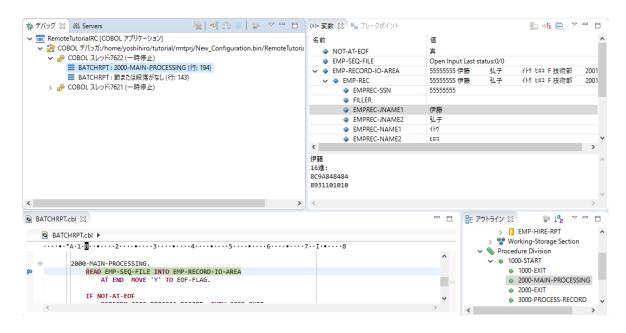
④ 条件付きブレークポイント機能を確認します。

【条件付きブレークポイントの例】

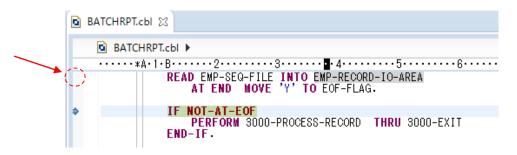




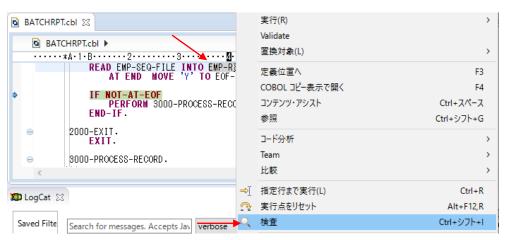
設定後、**F8** を打鍵しますと、設定した直後から5回目の READ 文のヒットでデバッガが一時 停止します。



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。

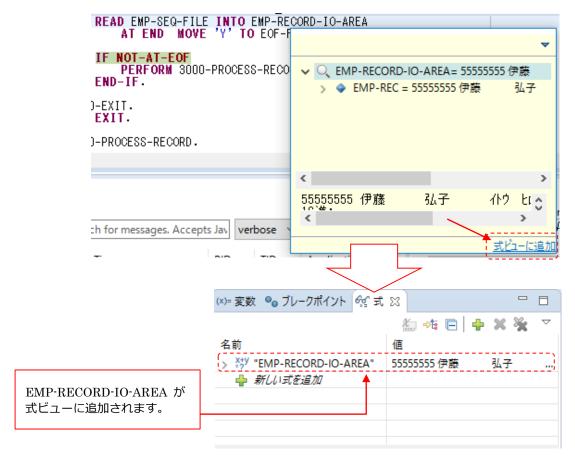


EMP-RECORD-IO-AREA の値が変わる度にデバッガを一時停止させます。EMP-RECORD-IO-AREA を選択し右クリックから [検査] を選択します。

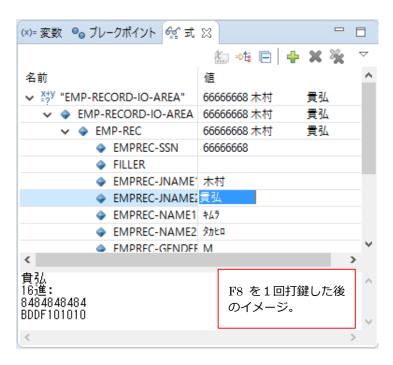




[式ビューに追加] をクリックします。



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。

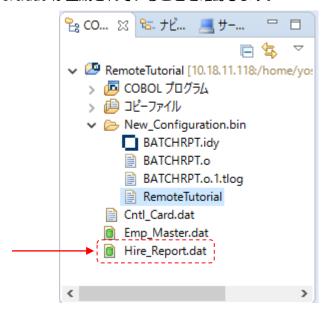


6 COBOL パースペクティブに戻します。

画面右上の COBOL パースペクティブアイコンを選択します。



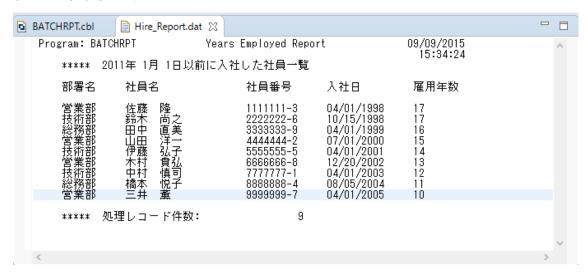
- 7 生成された帳票を確認します。
 - ① COBOL エクスプローラにて Hire_Report.dat が生成されていることを確認します。





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。

「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 03-5413-4800

URL http://www.microfocus.co.jp/