

Micro Focus メインフレームソリューション

スターターズキット

10. Enterprise Developer for Eclipse : リモート メインフレーム COBOL プロジェクト

当チュートリアルでは下記前提を踏まえて、リモートマシン（UNIX/Linux）とローカルマシン（Windows）間での リモート メインフレーム COBOL プロジェクトを Eclipse 上で実行する手順の習得を目的としています。

【前提：詳細は UNIX/Linux チュートリアルをご参照ください。】

- リモートマシンの DB へ接続を行う場合、DB 関連の環境変数が設定済であること。
- リモートマシンの DB へ接続を行う場合、リモートマシン内での DB 接続を確認済であること。
- リモートマシンの DB へ接続を行う場合、リモートマシンに XA リソースが作成済であること。
- XA リソースを使用する場合は、リモートマシンの Enterprise Server 開始時に XA リソースの正常ロードを確認済であること。
- リモートマシンにデバッグ対象となるアプリケーション一式が存在すること。
- リモートマシンに Samba がインストール済で稼働実績があること。

【手順】

- 1) リモートマシンの準備
- 2) Windows マップの設定
- 3) Eclipse のリモート接続作成
- 4) Eclipse のプロジェクト作成
- 5) Eclipse のデバッグ実行
- 6) リモートマシンの切断

10.1 リモートマシンの準備

ローカルマシンから接続するリモートマシンの準備を行うためルートユーザでログインします。

- 1) 環境変数 LANG に SJIS ロケールを設定します。

コマンド例 : `export LANG=ja_JP.sjis`

- 2) COBOL 実行環境を設定します。

製品をインストールしたフォルダ配下の bin に存在する `cobsetenv` を実行します。

コマンド例 : `./opt/mf/ED22U1GA/bin/cobsetenv`

```
[root@tok-rhel61-64 bin]# . cobsetenv
COBDIR set to /opt/mf/ED22U1GA
[root@tok-rhel61-64 bin]#
```

- 3) COBOL 作業モードを設定します。

COBOL の作業モード(32 ビットまたは 64 ビット)を指します。`cobmode` コマンドまたは環境変数「`COBMODE`」を使用して設定します。

32 ビット設定コマンド例 : `export COBMODE=32`

- 4) Directory Server を起動します。

`mfds`(Micro Focus Directory Server)コマンドを実行します。使用する環境によって、明示的に 32bit 環境用に `mfds32` コマンド、64bit 環境用に `mfds64` コマンドを実行することもできます。

コマンド例 : `mfds &`

上記 `&` を付加すると現在のプロセスとは別に `mfds` が開始されます。前項の COBOL 環境を基に起動されます。

- 5) ローカルマシンからアクセスを可能にするため、ルートユーザで接続ポートの解放を行います。

(下記“5000”部分で何も指定しない場合はデフォルトの“4075”)

コマンド例 : `$COBDIR/remotedev/startrdodaemon 5000`

```
[root@tok-rhel61-64 keit]# $COBDIR/remotedev/startrdodaemon 5000
Checking Java Version
Correct Java Version installed, proceeding
Starting RSE daemon...
[root@tok-rhel61-64 keit]# Daemon running on: tok-rhel61-64.microfocus.com, port
: 5000
```

- 6) Samba が未起動の場合は起動します。

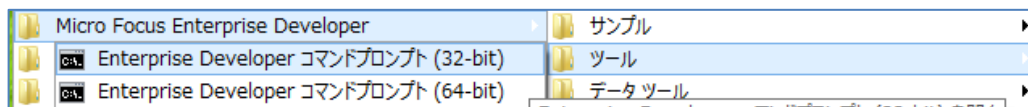
Samba 起動確認コマンド例 : `service smb status`

Samba 起動コマンド例 : `/usr/sbin/smbd -D`

10.2 Windows のマップ設定

ローカルマシンでマップを設定します。

- 1) コマンドプロンプトを管理者として実行します。エクスプローラのネットワークドライブの割り当てでも実行可能です。

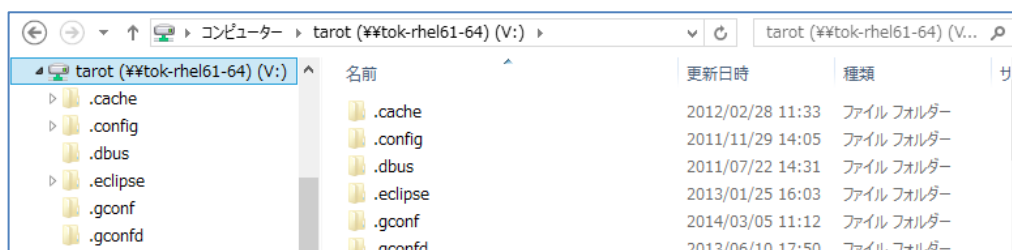


- 2) リモートマシンの共有エリアを権限があるユーザでマップします。ローカルマシンのエクスプローラで確認します。

コマンド例 : `net use v: ¥¥tok-rhel61-64¥tarot /user:taros password`

```
C:¥Users¥tarot¥Documents>net use v: ¥¥tok-rhel61-64¥tarot /user:taros password
コマンドは正常に終了しました。
```

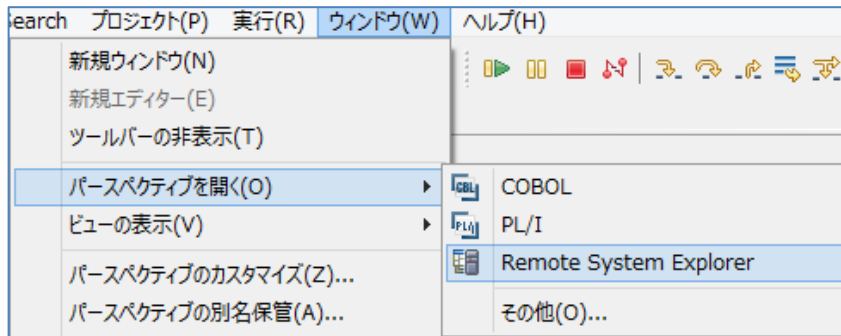
↓



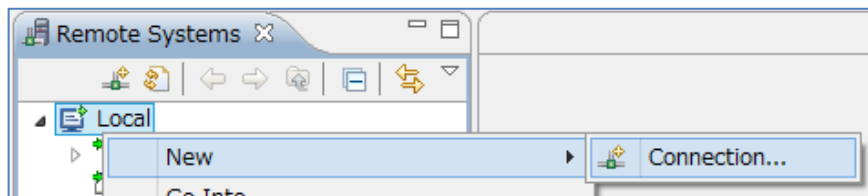
10.3 Eclipse のリモート接続作成

Eclipse 上の Remote System Explorer からリモートマシンへの接続設定を行います。

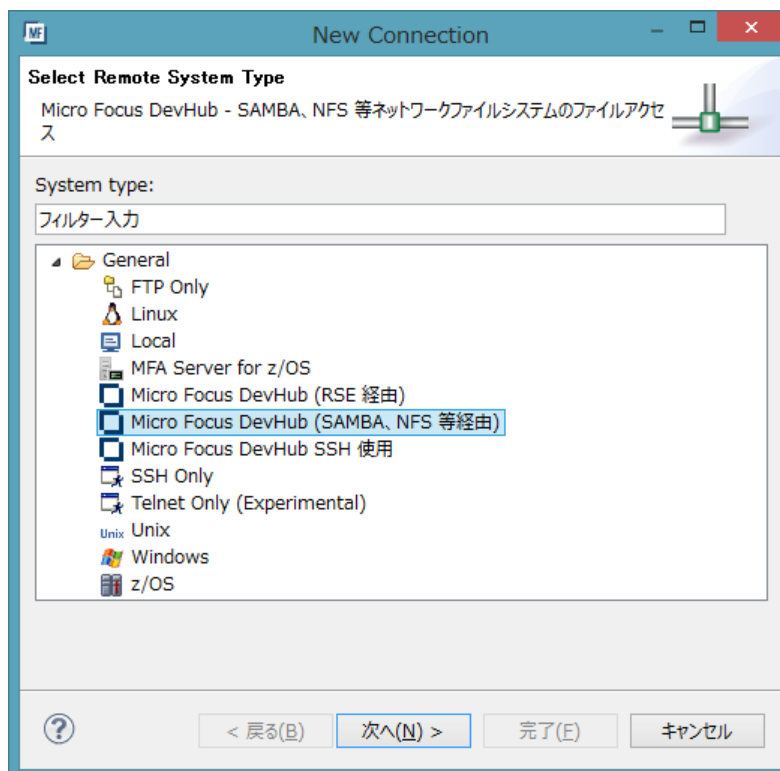
- 1) [ウィンドウ]>[パースペクティブを開く]>[Remote System Explorer] を選択して Remote System Explorer を開きます。



- 2) Remote System Explorer 上で[Local]>[New]>[Connection] を選択して新しい接続を作成します。



- 3) 「Micro Focus DevHub (SAMBA、NFS 等経由)」を選択して[次へ]ボタンをクリックします。



4) [Host name] と [Connection name] を入力して、[次へ]ボタンをクリックします。

Parent profile : TOK-W8-64T6s

Host name : tok-rhel61-64

Connection name : tok-rhel61-64

Description :

☒ Verify host name

< 戻る(B) 次へ(N) > 完了(E) キャンセル

5) ファイル : [Available Services] の [Launcher Properties] を選択し、[Properties] の [Deamon Port] へ リモートマシンで解放したポート番号を入力して[次へ]ボタンをクリックします。

ファイル
Define subsystem information

Configuration	Properties						
☒ dstore.files ☐ ssh.files	<table border="1"><thead><tr><th>プロパティ</th><th>値</th></tr></thead><tbody><tr><td>Daemon Port</td><td>5000</td></tr><tr><td>Launcher</td><td>Daemon</td></tr></tbody></table>	プロパティ	値	Daemon Port	5000	Launcher	Daemon
プロパティ	値						
Daemon Port	5000						
Launcher	Daemon						

Available Services

- DStore File Service
- DStore Connector Service
- Remote Server Launcher
- Launcher Properties

Description

The Server Launcher is an object for storing configuration details about how to

< 戻る(B) 次へ(N) > 完了(E) キャンセル

- 6) プロセス : [Available Services] 内の [Launcher Properties] を選択し、[Properties] の [Deamon Port] へ リモートマシンで解放したポート番号を入力して[次へ]ボタンをクリックします。

プロセス
Define subsystem information

Configuration

☒ dstore.processes

Properties

プロパティ	値
Daemon Port	5000
Launcher	Daemon

Available Services

- DStore Process Service
 - DStore Connector Service
 - Remote Server Launcher
 - Launcher Properties

Description

The Server Launcher is an object for storing configuration details about how to launch servers on the remote host.

< 戻る(B) 次へ(N) > 完了(E) キャンセル

- 7) シェル : [Available Services] の [Launcher Properties] を選択し、[Properties] の [Deamon Port] へ リモートマシンで解放したポート番号を入力して[次へ]ボタンをクリックします。

シェル
Define subsystem information

Configuration

☒ dstore.shells
☐ ssh.shells

Properties

プロパティ	値
Daemon Port	5000
Launcher	Daemon

Available Services

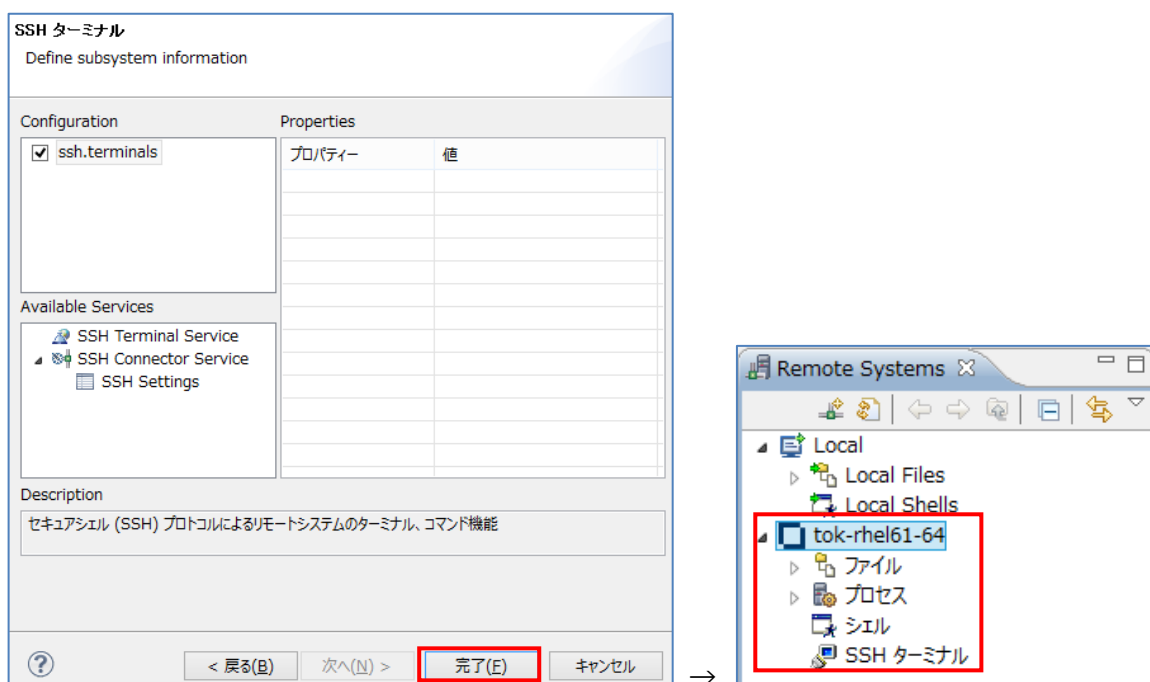
- DStore Shell Service
 - DStore Connector Service
 - Remote Server Launcher
 - Launcher Properties

Description

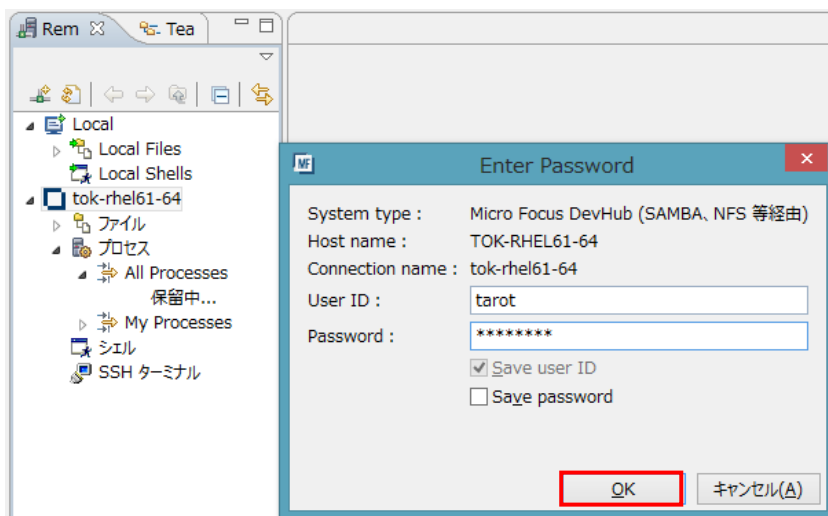
The Server Launcher is an object for storing configuration details about how to launch servers on the remote host.

< 戻る(B) 次へ(N) > 完了(E) キャンセル

8) [完了]ボタンをクリックすると、Remote System Explorer へ設定した接続が表示されます。



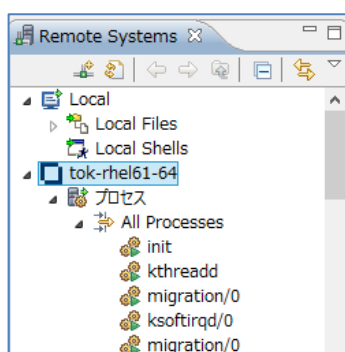
9) Remote System Explorer で追加した接続の [プロセス]>[All Processes] を展開するとユーザ ID とパスワードを要求されますので、正当な値を入力して OK をクリックしてください。



- 1 0) SSL を使用していない場合は下記メッセージが表示されます。次回からメッセージを表示したくない場合はチェックを入れてください。このまま接続を行いますので [はい] をクリックします。



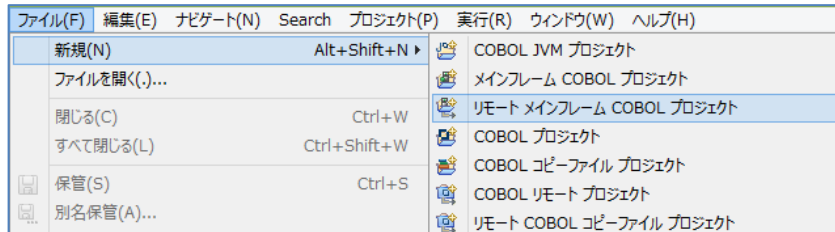
- 1 1) 接続が確立されてリモートマシンのプロセスが参照できることを確認します。



10.4 Eclipse のプロジェクト作成

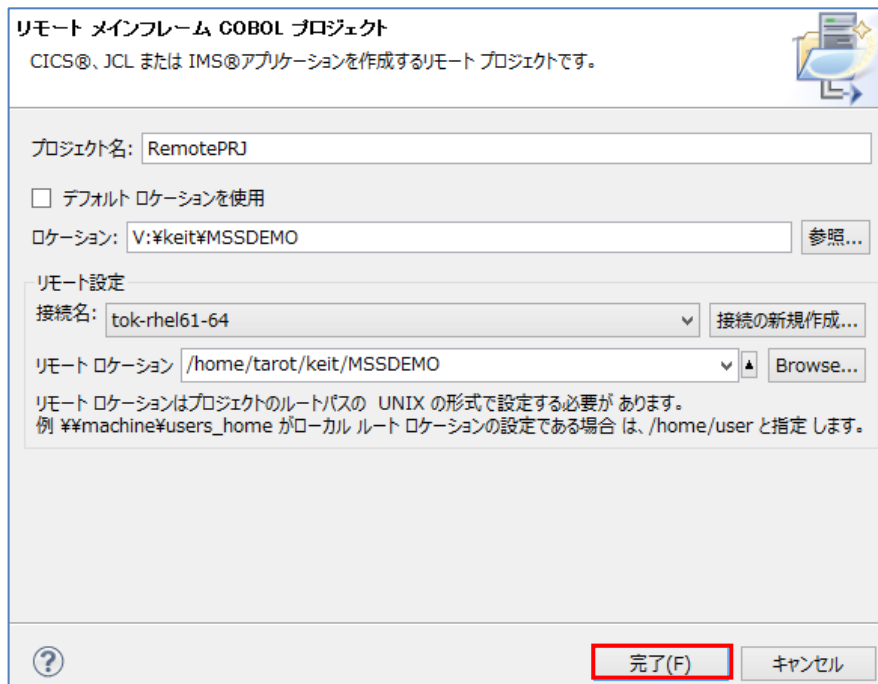
COBOL パースペクティブへ戻り、リモートアプリケーションのプロジェクトを新規に作成します。

- 1) プルダウンメニューから [ファイル]>[新規]>[リモート メインフレーム COBOL プロジェクト] を選択します。



- 2) 下記を設定して、[完了]ボタンをクリックします。

- ① プロジェクト名：任意の名称を入力します。
- ② ロケーション：ローカルのマップ先からリモートするプロジェクトロケーションを指定します。[参照]から指定可能です。
- ③ リモート ロケーション：リモート先ロケーションを指定します。接続名に依存した [Browse] から指定可能です。



⚠ 注意事項

書き込み権限がないロケーションを指定した場合はエラーになります。

ls -l コマンドなどで権限を確認後、chmod コマンドなどで適切な権限設定を前もって実行してください。

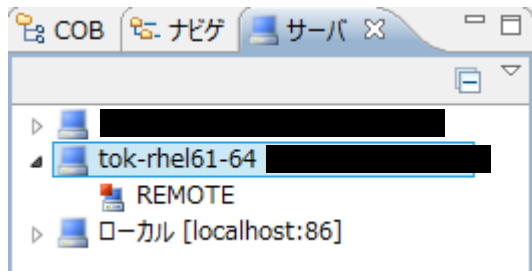
- 3) リモートマシンの Eclipse 上でビルドを行う場合も Eclipse を使用したチュートリアルと同様にプロジェクトプロパティやコンパイル設定を行います。

10.5 Eclipse のデバッグ実行

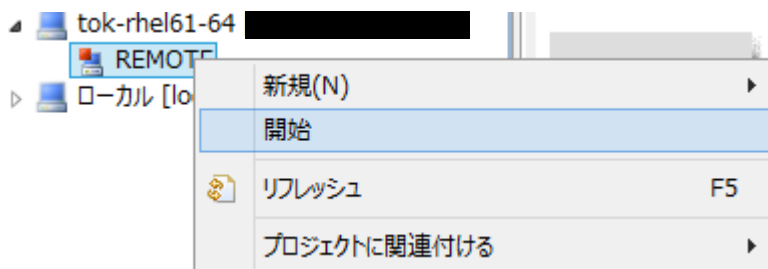
リモートデバッグを行います。

- 1) リモートマシンに存在する実行対象の Enterprise Server を開始します。
Enterprise Server の追加については「UNIX/Linux チュートリアル」をご参照ください。

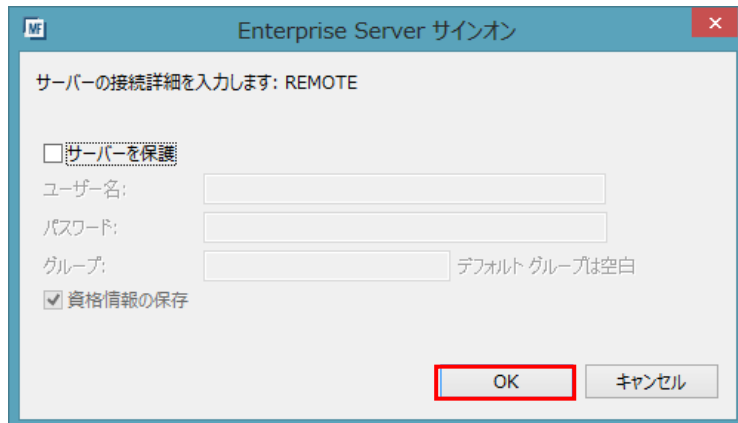
- ① サーバーエクスプローラーに表示されているリモートマシンを展開します。



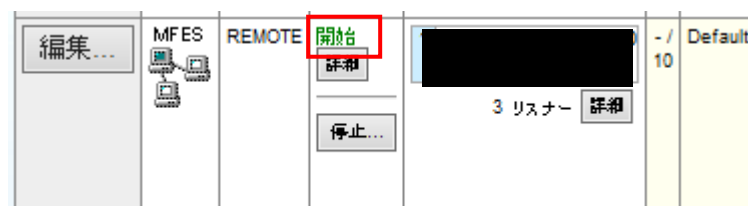
- ② REMOTE を右クリックして開始を選択します。



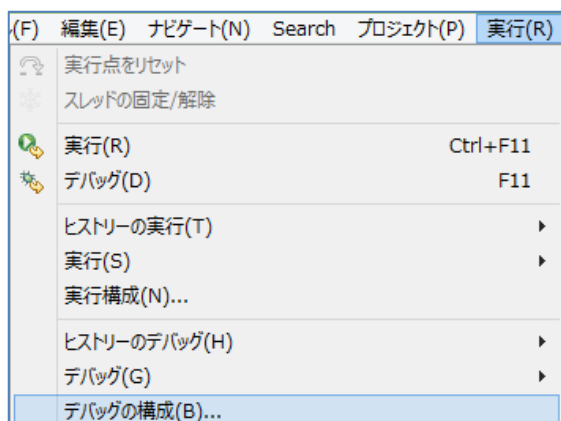
- ③ [OK] ボタンをクリックします。



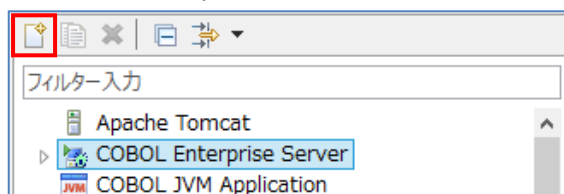
- ④ REMOTE が起動されました。



2) ローカルマシンの Eclipse プルダウンメニューの [実行]>[デバッグの構成] を選択します。

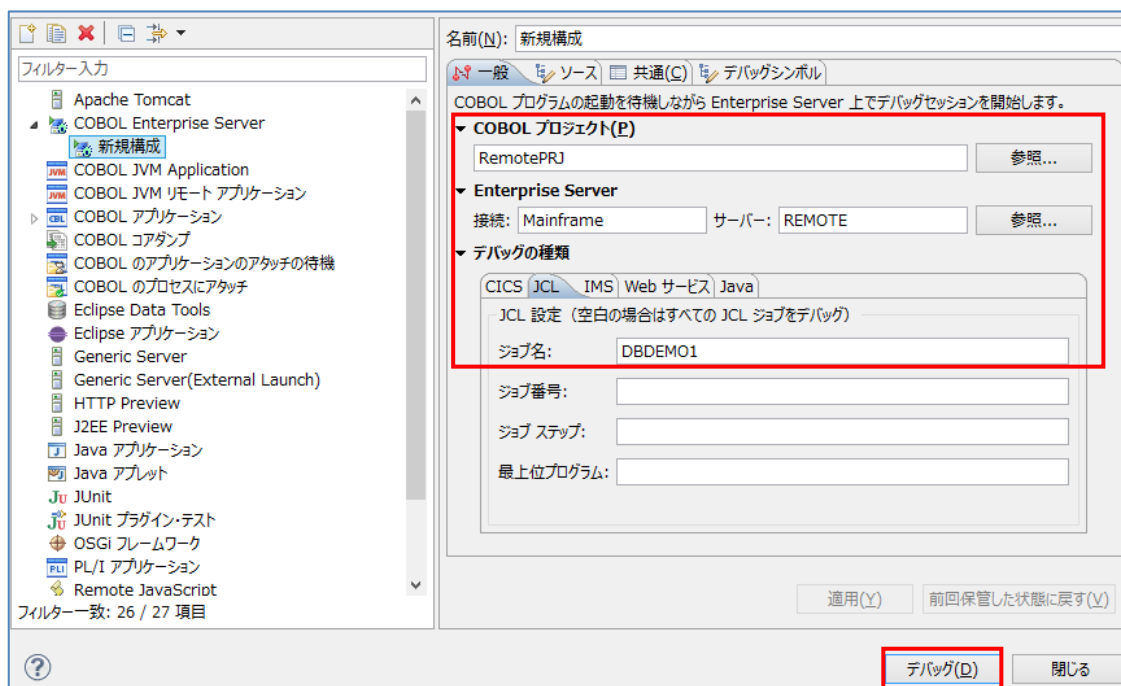


3) [COBOL Enterprise Server] を選択して [新規] ボタンをクリックします。

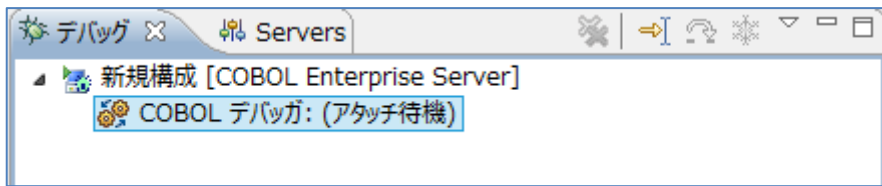


4) [新規構成] が作成されますので、下記項目を入力して [デバッグ] ボタンをクリックします。

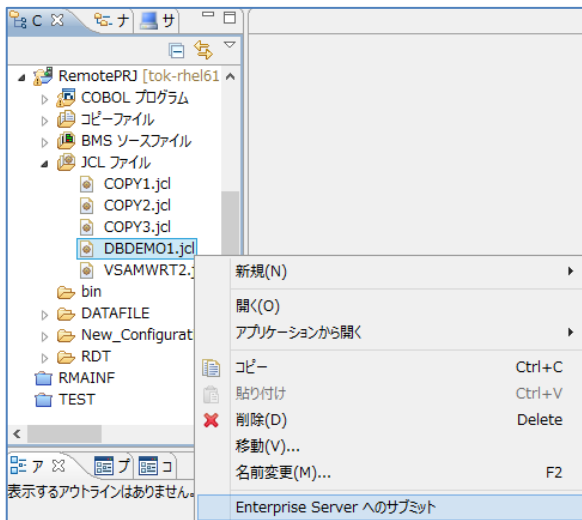
- ① COBOL プロジェクト：対象プロジェクト名を入力します。
- ② Enterprise Server：実行対象となるサーバーを指定します。
- ③ デバッグの種類：JCL を実行するため、JCL タブへサブミットする JCL 名称を入力します。



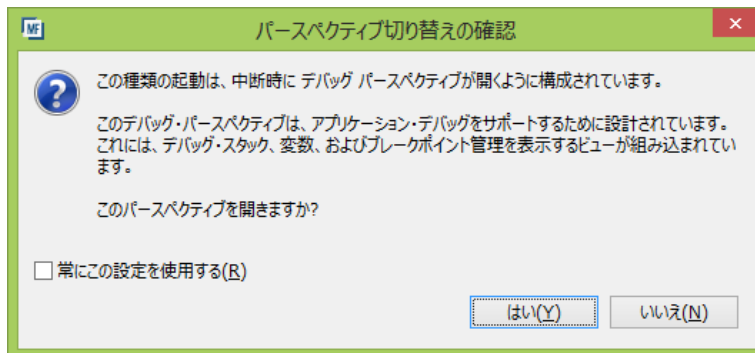
5) デバッグタブでアタッチ待機状態を確認します。



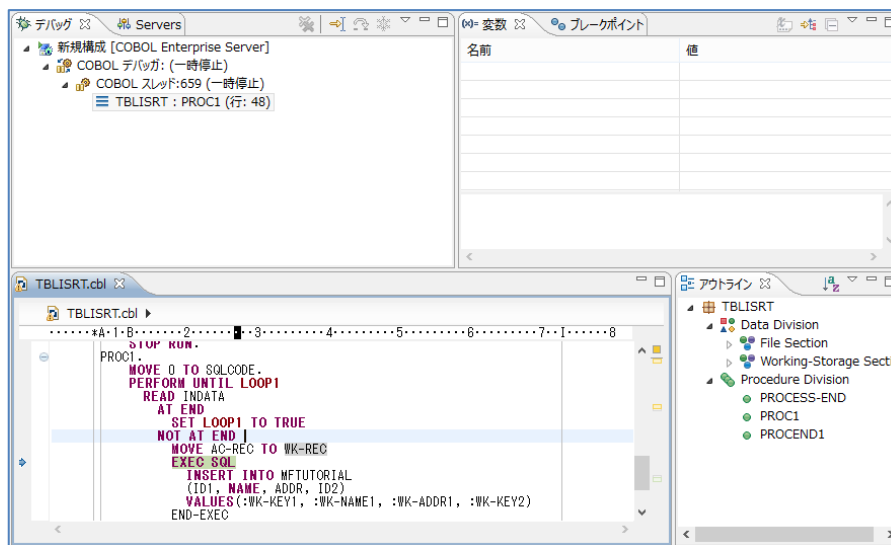
6) COBOL パースペクティブに戻り、ワークスペースから対象の JCL をサブミットします。



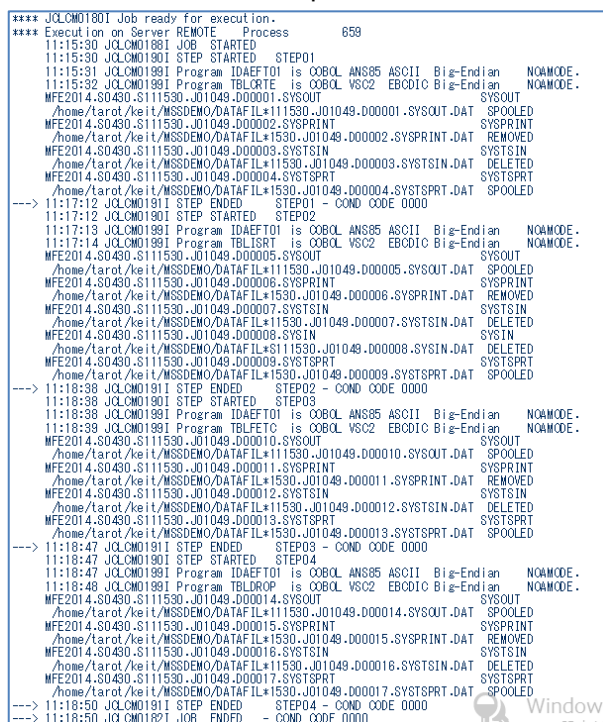
7) デバッグ用のパースペクティブを開きますので、下記画面では [はい] ボタンをクリックします。



8) プログラムのステップ実行が可能になります。

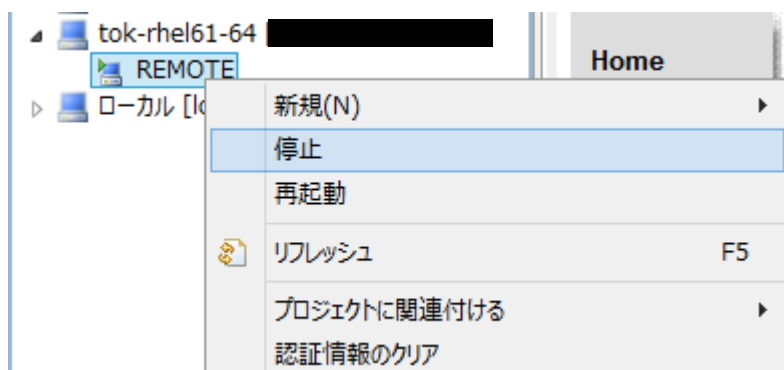


9) 実行終了後、対象の Enterprise Server のスプールを表示して、正常終了を確認します。



10)開始済の Enterprise Server を停止します。

- ① サーバークリエイターに表示されている REMOTE を右クリックして停止を選択します。



- ② REMOTE の停止を確認します。



10.6 リモートマシンの切断

ルートユーザでポートの閉鎖と Samba を終了します。

- 1) Samba を終了します。

Samba 終了コマンド例 : `service smb stop`

- 2) 接続済ポート（下記“5000”部分）を閉鎖します。

コマンド例 : `$COBDIR/remotedev/stoprdodaemon 5000`

```
[root@tok-rhel61-64 keit]# $COBDIR/remotedev/stoprdodaemon 5000
Process 22994 located:

root      22994 22991  0 Apr28 pts/1    00:01:08 java -DA_PLUGIN_PATH=/opt/mf/ED22U1_20140401/remotedev/rseserver/rseserv
er-3.5/ -DDSTORE_TRACING_ON=false org.eclipse.dstore.core.server.ServerLauncher

Do you wish to continue and kill it? (y/n): y
Kill signal sent to process 22994
[root@tok-rhel61-64 keit]#
```

- 3) Directory Server を停止します。

コマンド例 : `mfds -S 2`

オプションは以下の通りです。

-S option

where option = 1 Shutdown MFDS, leave ES servers started

option = 2 Shutdown MFDS, request shutdown of started ES servers

以上で リモート メインフレーム COBOL プロジェクト チュートリアルを終了します。