

Visual COBOL チュートリアル

RESTful Web サービスによる COBOL 資産の再利用

Eclipseリモート開発編

1

1 目的

Visual COBOL に付属する COBOL 専用のアプリケーションサーバー「Enterprise Server」は、ネイティブにコンパ イルした COBOL のビジネスロジックを REST API を利用し Web サービスとして呼び出す機能を提供しています。 RESTful の Web サービスとして呼び出しを行う場合、JSON 形式でやり取りが可能であれば呼び出し側のプログラムに 依存することなく連携できるようになります。

このドキュメントでは COBOL のソースコードに手を加えることなくビジネスロジックとして Enterprise Server にディプロイ し、それを Visual COBOL のクライアント生成機能を使って動作確認用のクライアントを作成し連携する方法を説明しま す。

2 前提条件

本チュートリアルでは、Linux サーバーとのリモート開発を行います。リモート開発については、「Visual COBOL チュートリ アル COBOL 開発: Linux/UNIX 版 リモート開発編」を参照してください。 また、本資料は下記の環境を前提に作成されています。サポートしているプラットフォームであれば Linux/UNIX でも利用 可能です。 開発クライアント(Windows) ソフトウェア OS Windows Server 10 COBOL 開発環境製品 Visual COBOL 10.0 for Eclipse Patch Update 01 開発クライアント(Linux) ソフトウェア OS Rocky Linux 9.4 COBOL 開発環境製品 Visual COBOL Development Hub 10.0 Patch Update 01

チュートリアル用サンプルプログラム

下記のリンクから事前にチュートリアル用のサンプルファイルをダウンロードして、任意のフォルダに解凍しておいてください。 サンプルプログラムのダウンロード



内容

- 1 目的
- 2 前提条件
- 3 チュートリアル手順
 - 3.1 Windows クライアントでの開発準備作業
 - 3.2 チュートリアルファイルのインポート
 - 3.3 リモートサーバーの Enterprise Server の追加と起動
 - 3.4 ディプロイフォルダー、データファイルの準備
 - 3.5 RESTful Web サービスの開発作業
 - 3.6 COBOL アプリケーションを Enterprise Server ヘディプロイ
 - 3.7 Enterprise Server インスタンスへの環境設定と有効化
 - 3.8 RESTful Web サービスのテスト
 - 3.9 RESTful Web サービスのデバッグ
 - 3.10 サーバーの停止

Rocket software

3 チュートリアル手順

2

3.1 Windows クライアントでの開発準備作業

- 1) Visual COBOL for Eclipse を起動
 - ① [スタート] メニュー > [すべてのアプリ] > [Micro Focus Visual COBOL] > [Visual COBOL for Eclipse] を選択します。

ワークスペースには任意のフォルダーを選択してください。

- 2) ネイティブ COBOL プロジェクトの作成とプログラムソースのインポート
 - [ファイル(F)]メニュー > [新規(N)] > [リモート COBOL プロジェクト] を選択し、プロジェクト名に "RemoteNativeCOBOL"を入力、ファイル システムに "セキュアシェル (SSH) ファイル システム"を選択して、 [次へ(N)] をクリックします。

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: RemoteNativeCOBOL
ファイル システム
ファイル システムを選択: セキュアシェル (SSH) ファイル システム ~
セキュアシェル (SSH) ファイル システムを使用すると、SSH 接続サポートのみを使用してリモート プロジェクトを処 理できます。ローカル ファイル システム上の場所を指定する必要はありませんが、リモート マシン上の場所のみ指 定する必要があります。
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 ファイルシステムの場所の指定は不要で、リモートマシン上の場所の指定だけが必要です。
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ブ上のプロジェク 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。
(?) < 戻る(B) 次へ(N) > 終了(F) キャンセル
[プロジェクト テンプレート] に""を選択して、[次へ(N)] をクリックします。



	『にリモート COBOL プロ	ジェクトを作成		- E->	
				,	
プロジェクト テンプレートを	を選択 Nプレート raa limba				
B Micro Focus 7	ジプレート [64 ビット]				
			<u></u>	レートの設定を構成	
🗌 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 ~				
?	< 戻る(B)	次へ(N) >	終了(F)	キャンセル	
「接続の新規作成	2] をクリックします	0			
	-				
ブロジェクト名: Remo	teNativeCOBOL				
リモート設定					
リモ−ト設定 接続名:				✓ 接続の新規作成	ì
リモート設定 接続名: [Micro Focus D	DevHub SSH Ø	ゆ)を選択し、[次へ(N)] をク	と 接続の新規作成 リックします。	
リモート設定 接続名: Micro Focus D Jモート・システム・タ	DevHub SSH の ソプの選択	ゆ]を選択し、[次へ(N)] をク	送続の新規作成 リックします。	2
リモート設定 接続名: [Micro Focus D Jモート・システム・タ Micro Focus DevHub	DevHub SSH の イプの選択 > への SSH アクセス用の)み] を選択し、[_{接続}	次へ(N)] をク	✓ 接続の新規作成 リックします。	
リモート設定 接続名: Micro Focus D Jモート・システム・タ Micro Focus DevHub	DevHub SSH の イプの選択 ¹ への SSH アクセス用の)み] を選択し、[_{接続}	次へ(N)] をク	✓ 接続の新規作成 リックします。	2
リモート設定 接続名: [Micro Focus D Jモート・システム・タ Micro Focus DevHub システム・タイプ:	DevHub SSH の イブの選択 9 への SSH アクセス用の)み] を選択し、[_{接続}	次へ(N)] をク	支続の新規作成 リックします。	2
リモート設定 接続名: Micro Focus D Jモート・システム・タ Micro Focus DevHub システム・タイプ: フィルタ入力	DevHub SSH の イプの選択 9 への SSH アクセス用の	ゆみ] を選択し、[_{接続}	次へ(N)] をク	接続の新規作成 リックします。	
リモート設定 接続名: [Micro Focus D Jモート・システム・タ Micro Focus DevHub システム・タイプ: フィルタ入力 > 0	DevHub SSH の イプの選択 かの SSH アクセス用の)み] を選択し、[^{接続}	次へ(N)] をク	送続の新規作成 リックします。	2
リモート設定 接続名: [Micro Focus D Jモート・システム・タ Micro Focus DevHub システム・タイプ: フィルタ入力 > 一 Micro Focus 」 Micro Focus	DevHub SSH の マイプの選択 ロへの SSH アクセス用の S DevHub SSH のみ S DevHub SSH 使用)み] を選択し、[^{接続}	次へ(N)] をク	送続の新規作成 リックします。	
リモート設定 接続名: [Micro Focus D Jモート・システム・タ Micro Focus DevHub システム・タイプ: フィルタ入力 ~ 一般 Micro Focus Micro Focus	PevHub SSH の イプの選択 かへの SSH アクセス用の S DevHub SSH のみ S DevHub SSH 使用)み] を選択し、[_{接続}	次へ(N)] をク	接続の新規作成 リックします。 ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	
リモート設定 接続名: [Micro Focus D Jモート・システム・タ Micro Focus DevHub システム・タイプ: フィルタ入力 × ○ 一般 □ Micro Focus □ Micro Focus	DevHub SSH の マイプの選択 ロへの SSH アクセス用の S DevHub SSH のみ S DevHub SSH 使用)み] を選択し、[^{接続}	次へ(N)] をク	<	2
リモート設定 接続名: [Micro Focus D Jモート・システム・タ Micro Focus DevHub システム・タイプ: フィルタ入力 >	DevHub SSH の マイプの選択 ロへの SSH アクセス用の S DevHub SSH のみ S DevHub SSH 使用	ゆみ] を選択し、[^{接続}	次へ(N)] をク	<	
リモート設定 接続名: [Micro Focus D リモート・システム・タ Micro Focus DevHub システム・タイプ: フィルタ入力 × ○ 一般 ■ Micro Focus ■ Micro Focus	PevHub SSH の イプの選択 かへの SSH アクセス用の SDevHub SSH のみ SDevHub SSH 使用)み] を選択し、[接続	次へ(N)] をク	 接続の新規作成 リックします。 	
リモート設定 接続名: [Micro Focus D Jモート・システム・タ Micro Focus DevHub システム・タイプ: フィルタ入力 × 2 一般 ■ Micro Focus ■ Micro Focus	DevHub SSH の マイプの選択 ロへの SSH アクセス用の S DevHub SSH のみ S DevHub SSH 使用)み]を選択し、[^{)接続}	次へ(N)] をク	⊻接続の新規作成 リックします。	
リモート設定 接続名: [Micro Focus D Jモート・システム・夕 Micro Focus DevHub システム・タイプ: フィルタ入力 > ● 一般 ■ Micro Focus	DevHub SSH の ¹ イプの選択 o への SSH アクセス用の s DevHub SSH のみ s DevHub SSH 使用	ゆみ] を選択し、[_{接続}	次へ(N)] をク	✓ 接続の新規作成 リックします。	
リモート設定 接続名: [Micro Focus D Jモート・システム・タ Micro Focus DevHub システム・タイプ: フィルタ入力 × ○ 一般 □ Micro Focus □ Micro Focus	PevHub SSH の イプの選択 かへの SSH アクセス用の S DevHub SSH のみ S DevHub SSH 使用	Dみ]を選択し、[接続	次へ(N)] をク	<	
リモート設定 接続名: [Micro Focus D Jモート・システム・タ Micro Focus DevHub システム・タイプ: フィルタ入力 ▼ ● ● ● ● ● Micro Focus ● Micro Focus	DevHub SSH の ・イプの選択 o への SSH アクセス用の s DevHub SSH のみ s DevHub SSH 使用	Dみ]を選択し、[次へ(N)] をク	 接続の新規作成 リックします。 	
リモート設定 接続名: Micro Focus D Jモート・システム・タ Micro Focus DevHub システム・タイプ: フィルタ入力 × C 一般 Micro Focus Micro Focus	PevHub SSH の ¹ イプの選択 ² への SSH アクセス用の ³ DevHub SSH のみ ³ DevHub SSH 使用	Dみ]を選択し、[接続	次へ(N)] をク	✓ 接続の新規作成 リックします。	
リモート設定 接続名: [Micro Focus D Jモート・システム・タ Micro Focus DevHub システム・タイプ: フィルタ入力 ▼ 2 - 99	DevHub SSH の Pイプの選択 P への SSH アクセス用の S DevHub SSH のみ S DevHub SSH 使用	 かみ]を選択し、[送続 	次へ(N)] をク	 接続の新規作成 リックします。 	

4



リモート 1 システム 接続情報の定義	接続(Micro Focus	DevHub SSH ගැ))	
親プロファイル:	win10-v-na			~
ホスト名:	172.22.242.111			~
接続名:	Linux			
記述/説明:				
☑ ホスト名を検証 <u>プロキシ</u> -設定を構成				
?	< 戻る(B)	次へ(N) >	終了(F)	キャンセル

⑥ [DevHub の場所] に Development Hub 製品のインストールディレクトリを入力し、[終了(F)] をクリックします。

DevHub SSH アクセス

Micro Focus DevHub のインストール場所の定義

Micro Focus DevHub 1 これは 'DevHub Ssh Ace	ンストール ディレクトリの場 cess' サブシステムのプロパテ	所はフルパスである必 ィを使用して後で変見	要があります。 更できます。	
<u>D</u> evHub の場所: /opt/	microfocus/VisualCOBC	L		\sim
?	< 戻る(<u>B</u>)	次へ(<u>N</u>) >	終了(<u>F</u>)	キャンセル

⑦ プロジェクトの保存先を指定し、[終了(F)] をクリックします。



リモート COBOL プロシ	ንェクト		
ワークスペースまたは外部	こリモート COBOL プロジェクトを	作成	-E->
プロジェクト名: Remotel	JativeCOBOL		
リモート設定			
接続名: Linux			◇ 接続の新規作成
リモート(/home/tarot/	RemoteNativeCOBOL		~▲参照
リモートの場所はリモート	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。	
	= - /0	A. 1. 42.7	
\bigcirc	< 戻る(<u>B</u>) 次へ	√(<u>N</u>) > 終了	(E) キャンセル

認証ダイアログが表示されますので、パスワードを入力し、[パスワードを保管(C)]にチェックをしたうえで、[OK] をクリ

ックします。					
システム・タイプ:	Micro Focus	DevHub SSH	のみ		
ホスト名:	172.22.242.1	111			
接続名:	Linux				
ユーザー ID:	tarot				
パスワード(任意)(B):	******				
	🖂 ユーザー II	Oの保管			
	☑ パスワード	を保管(C)			
				1	
	OK	キャンセル	(A)		
以下のようなダイアログ	が表示された場	易合は、[はい(`	Y)] を	クリックします	す。
The authenticity of host	'172.22.242.111	' can't be establ	ished.		
ECDSA key fingerprint is Are you sure you want to	5a:f0:39:0e:28: continue conr	63:25:87:8b:c2:do necting?	2:0a:5b:8	81:96:02.	
			lt	tい(Y)	เงเงิร์(N)

- 3) 文字コードの指定を行います。
 - Shift-JIS を指定して日本語を表示する場合、文字コードの指定を明確に行う必要があります。最初に、 [Window]メニュー > [設定] より [一般] > [ワークスペース] とナビゲートし、テキストファイルエンコードを「デフォ ルト(windows-31j)」に変更し、[適用して閉じる] をクリックします。



フィルタ入力	ワークスぺース ጐ ▼ ᠅ ▼ ᠅
▼ 一般 ^ Capabilities	ワークスペースの開始およびシャットダウン設定については、 <u>開始およびシャットダウン'</u> を参照してください。
> Security	□ ネイティブのフックまたはポーリングを使用して更新(R)
UI フリーズ・モニタ	✓ アクセス時に更新(S)
> User Storage Se	□ 無関係なプロジェクトを常にプロンプトなしで閉じる(C)
Web ブラウザ	ワークスペース保管間隔 (分)(W): 5
> エディタ	
+-	ウィンドウのタイトル
クイック検索	✓ ワークスペース名を表示(E): RESTful
フレーハルに	□ パースペクティブ名を表示(T)
サービス・ポリシー	 ワークスペースのフルパスを表示(F): C:¥work¥RESTful
トレース	□ プロダクト名を表示
> ネットワーク接続	
ハンドラーをリンク	
パースペクティブ	ノロンエントを用く際に、参照するノロンエントを用く、ノロノント 🎽
ノロシェクト・ネーフ	不明なプロジェクトの性質を以下のように報告(A): 警告 ~
> 開始およびシャッ	Report missing project encoding as: 警告 ~
> 外観	
検索	ミノフテム・エクスプローラーを記動するコマンド(X): explorer /F /select=\${selected resource loc}
通知	
比較/パッチ	
> Ant	デキスト・プアイル・エンコート(1) 新規プキスト・プアイルの行区切り文字(F) ● デフォリト(1) (Mindawa 24)
Aspecti Compiler	
> Gradle	○その他(O): windows-31 ◇ ○その他(H): Windows ◇
> HTML (Wild Web I Y	デフォルトの復元(T) 適用(I)
< >	
? 눱 🖆 📀	適用して閉じる キャンセル

Preference Recorder のダイアログが表示された場合は [キャンセル] を選択してください。

② 次に作成した COBOL プロジェクトを選択した状態で、マウスの右クリックにてコンテクストメニューを表示し、「プロパティ」を選択します。[Micro Focus] > [プロジェクト設定] > [COBOL] とナビゲートし、 [一般] > [ソース エンコーディング]を "UTF-8" から "ANSI" に変更し、[適用(L)] をクリックします。



フィルタ入力	COBOL	¢	• • • •
> リソース			
Coverage			
✓ Micro Focus ビルダー	フィルタテキストを入力 国		
ビルド パス	設定	值	^
> ビルド構成	▶ 一般		
✓ プロジェクト設定	文字セット	ASCII	
COBOL	ソース エンコーディング	ANSI	
ビルド環境	COBOL 方言	Micro Focus	
指令の確定	ソース フォーマット	固定	
> 実行時構成	デバッグ用にコンパイル	はい	
WikiText	EXIT PROGRAM を GOBACK としてタ	型理 ANSI	
サーバー	詳細	いいえ	
タスク・タグ	.GNT にコンパイル	いいえ	
> タスク・リポジトリー	✔ 出力		
ビルダー	指令ファイルを生成する	いいえ	
プロジェクト・ネーチャー	リストファイルを生成	いいえ	
プロジェクト・ファセット	コード カバレッジを有効にする	false	
プロジェクト参照	プロファイラを有効にする	false	
>検証	✓ Iラ-/警告		
実行/デバッグ設定	警告レベル	回復可能なエラーを含める(レベル E)	~
	ソース エンコーディング		^
	SOURCE-ENCODING はソース プログラムのエ	ンコーディングをコンパイラに渡します。その後、RUNTIME-ENCODING	指 >
	COBOL コンパイル設定:		
	CHARSET"ASCII" SOURCE-ENCODING"A EXITPROGRAM"ANSI" NOTESTCOVER NO	NSI" DIALECT"MF" SOURCEFORMAT"fixed" NOLIST anim DPROFILE WARNING"1" MAX-ERROR"100"	^
			\checkmark
)** III (1)
		デノオルトの復元(<u>I</u>)	週用(<u>L</u>)

- 4) ビルドオプションの変更
 - 現在の画面より、[.GNT にコンパイル]を「はい」に変更し、[追加指令]に "ASSIGN(EXTERNAL)" を指定し、 [適用(L)] ボタンをクリックします。

設定	値		,
	Micro Focus		
ソースフォーマット	固定		
デバッグ用にコンパイル	(tu)		
EXIT PROGRAM を GOBACK とし	て処理 ANSI		
	いいえ		
.GNT にコンパイル	はい		
✔ 出力			
指令ファイルを生成する	いいえ		
リストファイルを生成	いいえ		
コード カバレッジを有効にする	false		
プロファイラを有効にする	false		
✔ エラー/警告			
警告レベル	回復可能なエラーを含める	(レベル E)	
最大エラー数	100		
✔ 追加指令			
追加指令	ASSIGN(EXTERNAL)		
自加指令			
ンパイラに渡す追加のCOBOLコンパイラ指	令です		
OBOL コンパイル設定:			

② [Micro Focus] > [ビルド構成] > [リンク] をクリックし、[ターゲットの種類] を「すべて INT/GNT ファイル」に変更し、[適用して閉じる] をクリックします。

設定 V Linkage	値	^
✓ Linkage 出力のを前		
山市の友前		
ロノルウロ別	RemoteNativeCOBOL	
出力パス	New Configuration.bin	
エントリポイント		
ターゲットの種類	すべて INT/GNT ファイル	
ビット数	64 ビット	
.LBR にパッケージ化	いいえ	
COBOL以外のアプリケーションから呼び出し可能	ししえ	
サービスを COBOL アーカイブ (.car) ファイルにパッケージ化	いいえ	
マルチスレ ッド	いいえ	
実行時モデル	共有	
現在の実行時システムだけにバインドする いいえ		
出力の種類	コンソール	
ターゲット オペレーティング システム	Unix/Linux	
詳細 いいえ		
cpp ライブラリを含める	いいえ	
未定義シンボルでエラー	いいえ	
エントリポイント アドレスを読み込む	いいえ	
Windows リソースファイル		
迫加指会		~
	 出力パス エントリポイント ターゲットの種類 ビット数 LBR にパッケージ化 COBOL 以外のアブリケーションから呼び出し可能 サービスを COBOL アーカイブ (.car) ファイルにパッケージ化 マルチスレッド 実行時子デル 現在の実行時システムだけにパインドする 出力の種類 ターゲット オペレーティング システム 詳細 cpp ライブラリを含める 未定義シンボルでエラー エントリポイント アドレスを読み込む Windows リソースファイル 注加球会 ターゲットの種類 	出力パス New Configuration.bin エントリボイント すべて INT/GNT ファイル ダーゲットの種類 すべて INT/GNT ファイル ビット数 64 ビット LBR にパッケージ化 いいえ COBOL 以外のアブリケーションから呼び出し可能 いいえ サービスを COBOL アーカイブ (car) ファイルにパッケージ化 いいえ マルチスレッド いいえ 東行時モデル 共存 現在の実行時システムだけにパインドする バいえ ガカの種類 コンソール ターゲット オペレーティング システム Unix/Linux 詳細 いいえ にpp ライブラリを含める いいえ エントリボイント アドレスを読み込む いいえ ゾレースファイル - シーケットの種類 -

3.2 チュートリアルファイルのインポート

- 1) チュートリアルファイルのインポート
 - ① 作成したプロジェクトを選択し、マウスの右クリックでコンテクストメニューを開き、[インポート(I)] > [インポート(I)] を 選択します。
 - ② [一般] > [ファイル・システム] を選択し、[次へ] をクリックします。

1ンホート・ウイザードの フィルタ入力	選択(S):		
~ ⊱ 一般			
、 ⁽⁾⁾ アーカイブ・ ⁽⁾ ファイル・シ ⁽⁾ フォルダーま	ファイル ステム たはアーカイブ由来のプロジ	ェクト	
営 既存プロジ	ェクトをワークスペースへ		
設定			
> 🗁 Git			
> 🗁 Gradle			



③ [参照(R)] ボタンをクリックし、ダウンロードしたサンプルファイルを展開したフォルダーを指定します。下図では "C:¥vc-tutorial-rest02" に展開したファイルを指定しています。ここで "BOOK-INFO.cpy"と "BOOK.cbl"を指定し、[終了(F)] ボタンをクリックします。

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:¥vc-tutorial-rest02		~ 参照(R)
> 🔳 😂 vc-tutorial-rest02	BOOK.cbi BOOK-INFO.CPY BOOKSCRN.cbi BOOKSCRN.cbi Readme.txt	
タイプをフィルター(T) すべて選択(S) 選択をすへ	べて解除(D)	
インポート先フォルダ(L): RemoteNativeCOBOL		参照(W)
オブション 学告を出さずに既存リソースを上書き(O) トップ・レベルのフォルダーを作成(C) 拡張 >>(A)		
0	< 戻る(B) 次へ(N) >	終了(F) キャンセル

自動的にビルドが行われ、プロジェクト配下が以下のようになります。

Ŀ	С)В	×	10プ	0	28	Appl	<u>씨</u> 카	- <i>K</i>	ł
										~
v	B	Rei	not	eNativ	/eCO	BOL	[Linux:	/hom	e/taro	ot/
	v	Þ	COE	BOLプ	ログラ	4				
		>	<u>o</u> e	BOOK.	cbl					
	v	Þ	⊐Ľ-	-ファイ)	V					
			B E	BOOK-	INFC).CP	Y			
	v	B	Nev	v_Con	figura	atio	n.bin			
			R E	BOOK.	gnt					
			E E	BOOK.	gnt.1	.tlog	9			
			🗆 E	BOOK.	idy					
				BOOK.	objlis	st				



3.3 リモートサーバーの Enterprise Server の追加と起動

- 1) リモートサーバー上でサーバーの起動
 - ① ターミナル画面上で、管理者権限を持つユーザーで、リモートサーバーにログインします。
 - ② 以下のコマンドを実行します。

```
. /opt/microfocus/VisualCOBOL/bin/cobsetenv
```



Directory Server の起動

以下のコマンドを実行します。

mfds&

[±] mfds&			
1] 2876			
ŧ			

Enterprise Server Common Web Admin(ESCWA) の起動

以下のコマンドを実行します。

```
escwa --BasicConfig.MfRequestedEndpoint="tcp:*:10086" --write=true &
```

```
補足)
```

デフォルトでは外部からアクセスすることはできません。外部アクセスを許可するためのオプションを指定しています。

```
# escwa --BasicConfig.MfRequestedEndpoint="tcp:*:10086" --write=true &
[2] 2893
[root@rocky9-v-na mfds]# 2024-10-18 12:03:48.498 Loaded COBOL Run Time Envir
onment Extension
GkCobExInit at 0x7fe9f0e00e29
2024-10-18 12:03:48.499 New thread high-water mark: 1 threads are now running
2024-10-18 12:03:48.499 MFCS server "ESCWA" running as process 2893
2024-10-18 12:03:48.499 GK-OS version 2.11.2
2024-10-18 12:03:48.499 GK-Utility version 2.11.3
2024-10-18 12:03:48.500 GkCobGetFuncAddr: 4
2024-10-18 12:03:48.500 ES Common Web Administration version: 6.1.1
2024-10-18 12:03:48.504 Common Web Admin http endpoint starting on endpoint:
tcp:*:10086
```

2) Eclipse 側の設定

#

③ Eclipse に戻り、[サーバーエクスプローラー] タブを選択します。

```
Secolarity COB... × L プロ... PAppl... ■ サーバ... ■ Anal... □

× E SemoteNativeCOBOL [Linux:/home/tarot/RemoteNativeCC

× I COBOL プログラム

> I BOOK.cbl
```

Rocket software

④ [+] ボタンをクリックします。

🔓 COB	🔓 プロ	😤 Appl	黒 サーバ	×	📇 Anal		-	
					~	÷	Ð	000
i ESC	WA インスタ	ソス未定義						

リモートサーバーの情報を入力し、[OK] をクリックします。 (5)

接続の新規作成

既存の Enterprise Server Common Web Administration インスタンスへの接続を新規作成します

名前:	Linux			
サーバアドレス (IPv4/ホスト名):	172.22.242.111			
サーバーポート:	10086			
ITLS 有効				
TLS 設定				
CA 証明書:				
		参照		
?			終了(F)	キャンセル

認証情報を入力し、[OK] をクリックします。

 ◇サーバーに認証情報が必要 認証情報 ユーザー名: SYSAD パスワード: ******** ◇認証情報の保存 ②認証情報がクリアされるまで、再度プロンプトを表示しない OK キャンセル 補足) この情報は、以下のコマンドで取得できます。 mfsecretsadmin read microfocus/temp/admin # mfsecretsadmin read microfocus/temp/admin {"mfl lser":"SYSAD" "mfPassword":"88tud18a"} 		
認証情報 ユーザー名: SYSAD パスワード: ******* ✓ 認証情報の保存 ○ 認証情報がクリアされるまで、再度プロンプトを表示しない OK キャンセル 補足) この情報は、以下のコマンドで取得できます。 mfsecretsadmin read microfocus/temp/admin # mfsecretsadmin read microfocus/temp/admin {"mfl lser":"SYSAD" "mfPassword":"88tud18a"}	☑ サーバーに認証情報が必要	
ユーザー名: SYSAD パスワード: ******* ✓ 認証情報の保存 □ 認証情報がクリアされるまで、再度プロンプトを表示しない OK キャンセル 補足) この情報は、以下のコマンドで取得できます。 mfsecretsadmin read microfocus/temp/admin # mfsecretsadmin read microfocus/temp/admin {"mfl lser":"SYSAD" "mfPassword":"88tud18a"}	認証情報	
パスワード: ******* ✓ 認証情報の保存 □ 認証情報がクリアされるまで、再度プロンプトを表示しない OK キャンセル 補足) この情報は、以下のコマンドで取得できます。 mfsecretsadmin read microfocus/temp/admin # mfsecretsadmin read microfocus/temp/admin {"mfl lser":"SYSAD" "mfPassword":"88tud18a"}	ユーザー名: SYSAD	
 ✓ 認証情報の保存 □ 認証情報がクリアされるまで、再度プロンプトを表示しない OK キャンセル 補足) この情報は、以下のコマンドで取得できます。 mfsecretsadmin read microfocus/temp/admin # mfsecretsadmin read microfocus/temp/admin {"mfl lser":"SYSAD" "mfPassword":"88tud18a"} 	パスワード: *******	
 □ 認証情報がクリアされるまで、再度プロンプトを表示しない OK キャンセル 補足) この情報は、以下のコマンドで取得できます。 mfsecretsadmin read microfocus/temp/admin # mfsecretsadmin read microfocus/temp/admin {"mfl lser":"SYSAD" "mfPassword":"88tud18a"} 	✓ 認証情報の保存	
補足) この情報は、以下のコマンドで取得できます。 mfsecretsadmin read microfocus/temp/admin # mfsecretsadmin read microfocus/temp/admin {"mfUser":"SYSAD" "mfPassword":"88tud18a"}	 認証情報がクリアされるまで、再度プロンプトを表示しない OK キャンセル 	
この情報は、以下のコマンドで取得できます。 mfsecretsadmin read microfocus/temp/admin # mfsecretsadmin read microfocus/temp/admin {"mfUser":"SYSAD" "mfPassword":"88tud18a"}	補足)	
mfsecretsadmin read microfocus/temp/admin # mfsecretsadmin read microfocus/temp/admin {"mfUser":"SYSAD" "mfPassword":"88tud18a"}	この情報は、以下のコマンドで取得できます。	
# mfsecretsadmin read microfocus/temp/admin {"mfUser"."SYSAD" "mfPassword"."88tud18a"}	mfsecretsadmin read microfocus/temp/admin	
{"mfl.lser"·"SYSAD" "mfPassword"·"88tud18a"}	# mfsecretsadmin read microfocus/temp/admin	
	{"mfUser":"SYSAD", "mfPassword":"88tudJ8a"}	
上記の場合、[ユーザー名] に "SYSAD"、[パスワード] に "88tudJ8a" を入力します。	上記の場合、[ユーザー名] に "SYSAD"、[パスワード] に "88tudJ8a" を入力します。	

しばらくすると、Directory Server 情報が表示されます。

Rocket software



⑥ リモートサーバー配下の「ESDEMO64」を右クリックし、コンテクストメニューから [開始] を選択します。もし、ダイ

アログが表示されたらそのまま[OK]ボタンを押してください。



起動すると、ESDEMO64の左側に緑色のアイコンが表示されます。

🔓 COB... 陷 プロ... 😤 Appl... 🔜 サーバ

- ✓ Iinux [172.22.242.111:10086]
 - Default [127.0.0.1:86]
 - > 📇 ESDEMO
 - > 🔚 ESDEMO64

3.4 ディプロイフォルダー、データファイルの準備

- 1) COBOL エクスプローラーの表示設定
 - ① COBOL エクスプローラーに戻ります。
 - ② COBOL エクスプローラー右上の「↓↑」アイコンの右横にあるアイコンをクリックし、[フィルタとカスタマイズ(F)] を 選択します。

陰 COB × ြ ブロ と Appl 当サーバ 当 Anal ビ マ 🕞 🔩 🔍 🍕	000		
 ✓ 28 RemoteNativeCOBOL [Linux:/home/tarot/RemoteNative ✓ 29 COBOL プログラム 	(COBOL JVM プロジェクト表示(R) トップレベル要素(T)	> >
 > 図 BOOK.cbl ✓ (型) コピーファイル © BOOK-INFO.CPY ○ BOOK-INFO.CPY 		ワーキング・セットの選択(W) ワーキング・セットの選択解除(K) アクティブなワーキング・セットの編集(E)	
 New_Configuration.bin BOOK.gnt BOOK.gnt.1.tlog 	(6) 17	1 ウィンドウ・ワーキング・セット フィルタとカスタマイズ(F)	

③ [カテゴリ外の空のフォルダ] にチェックされている場合は、チェックを外したのち、[OK] ボタンをクリックします。



☑ .* リソース			
✓ COBOL JVM 出	カフォルダ		
COBOL カテゴリ	内の空のフォルダ		
COBOL JĽ-Jľ	٬イル プロジェクト		
└── RSE 内部プロジェ	ロクト		
□ カテゴリ外の空の	フォルダ		
✓ 内部 Micro Foc	us プロジェクト		
小 内部 TD プロジェ	: クト		
☑ 合成メンバ			
☑ 空のカテゴリ			
🗌 空のパッケージ			
☑ 空の親パッケージ			

- 2) ディプロイ用フォルダーの作成
 - 「RemoteNativeCOBOL」プロジェクトを右クリックし、コンテクストメニューから [新規作成(N)] > [フォルダー] を 選択します。

📑 🕶 🗐 🐚 📮 🖏 🛊 🕶 💽 🕶 💁 🕶	:			- un A	
		新規作成(N)	>		COBOL JVM フロジェクト
		表示方法(W)	Alt+シフト+W >	20 	COBOL JVM ユーット テスト フロシェクト COBOL コピーファイル プロジェクト
✓	D	כול	Ctrl+C	2	COBOL JUJIJA
✓ 個 COBOL プログラム	Ē	貼り付け	Ctrl+V	6	COBOL ユニット テスト プロジェクト
> 🖻 BOOK.cbl	×	削除(D)	削除	썉	COBOL/Java 相互運用機能のプロジェクト
∽ 👜 コピーファイル	Ð.	コンテキストから除去	Ctrl+Alt+シフト+下	2	リモート COBOL JVM プロジェクト
BOOK-INFO.CPY		移動(V)		罉	リモート COBOL コピーファイル プロジェクト
✓ ➢ New_Configuration.bin		名前を変更(M)	F2	(リモート COBOL プロジェクト
BOOK.gnt		ビルド アクション	>	1	リモート COBOL ユニット テスト プロジェクト
BOOK.gnt.1.tlog		タスクのスキャン			プロジェクト(R)
BOOK.idy		指令の確定		DP	COBOL コピーファイル
BOOK.objlist		ファイル指令の削除		Ø	COBOL プログラム
		コード公析	>	Ľ	スタンドアロン ファイル
				Ľ	リモート スタンドアロン ファイル
		インボート(i)	>	: \$	Java インターフェイス
		エクスボート(O)		89	REST Web サービス
<	\$	更新(F)	F5	:3	SOAP Web サービス
🏗 アウトライン × 🔤 プログラム アウト 🥅		プロジェクトを閉じる(S)		2	リモート リンク フォルダ
		無関係なプロジェクトを閉じる(U)			
アウトラインを提供するアクティブなエディターはあ		Source	>		3QL ノアイル ファイル
		リモートシステムビューで表示		Ŷ	フォルダー

 ② 「RemoteNativeCOBOL」プロジェクトが選択されていることを確認の上、フォルダ名に "dat" を指定し、[終了 (F)] ボタンをクリックします。



フォルダを人力または	:選択(E):			
emoteNativeCOBC	DL			
} <> <}				
🖻 🗁 InternalTDPro	oject			
😕 RemoteNativ	eCOBOL [Linux:/home/ta	rot/RemoteNative	COBOL]	
🗁 RemoteSyster	mslempFiles			
+ .ズク(N))- dat				
tルダ名(N): dat				
tルダ名(N): <mark>dat</mark> 拡張 >>(A)				

- 3) データファイルのインポート
 - ① dat フォルダーを選択し、マウスの右クリックにてコマンドでコンテクストメニューを開き、[インポート(i)] > [インポート

(I)] を選択	1	ます。				
🔓 COB 🗙 🍋	プロ.	😤 Appl 🔳 サーバ 🌡	🖫 Anal 🗖 🗖	BOOK	(.cbl	x
✓ ⊯ RemoteNat ✓ ፼ COBOL	tive プロ:	マ COBOL [Linux:/home/tarot/ グラム	RemoteNativeCOE	■ B	00	K.cbl ▶ *A·1-₿··•···2····3····4····4····5·
> 🖻 BOOI マ 🔑 コピーファ BOOI	K.cb イル K-IN	NFO.CPY		0 0		PROCEDURE DIVISION USING LNK-FUNCTION LNK-B-DETAILS
🗁 dat		新規作成(N)		>	١.	MAIN SECTION.
> 🥭 New_C		表示方法(W)	Alt+シ	ワト+W >		CALL "CRI TOUPPER" USING INK-R-TEXT-DET
		コピー		Ctrl+C	Ŀ	BY VALUE LENGTH LNK-B-TE
		貼り付け		Ctrl+V 尚服会	Ŀ	RETURNING LS-CALL-STATUS
	~	コンテキストから除去	Ctrl+Alt+シ	_{用リレホ} フト+下	Ŀ	EVALUATE TRUE
		移動(V) 名前を変更(M)		F2		WHEN READ-RECORD PERFORM DO-READ-RECORD
<		タスクのスキャン				WHEN ADD-RECORD
譯 アウトライン ×		インポート(i)		>	Ê,	リモート プロジェクト
	4	エクスポート(O)…			8	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
V 🕑 BOOK	8	更新(F)		F5	ė	Net Express プロジェクトの変換
> To Enviror		Source		>	è	インボート(I)

[一般] > [ファイル・システム] を選択し、[次へ(N)] ボタンをクリックします。



選択

ローカル・ファイル・システムから既存のプロジェクトへリソースをインポートします。



インポート・ウィザードの選択(S):				
フィルタ入力				
▼ ⊱ 一般				^
 ◎ アーカイブ・ファイル ○ ファイル・システム ○ フォルダーまたはアーカイ ☆ 閃存プロジェクトをワーク □ 設定 	'ブ由来のプロジェク ウスペースへ	٢		
> 🗁 EJB				
> 🗁 Git				
> 🗁 Gradle				
> 🗁 J2EE				
> 🗁 Maven				
> 🗁 Micro Focus				
> 🗁 Micro Focus インターフェイ	イスマッパー			
> 🗁 Oomph				~
?	< 戻る(B)	次へ(N) >	終了(F)	キャンセル

③ [参照(R)] をクリックし、チュートリアル用ファイルを展開したフォルダー配下の DAT を指定したうえで、 BOOKINFO.DAT にチェックを行い、「終了(F)]をクリックします。

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:¥vc-	tutorial-rest02¥DAT			~	参照(R)
DAT			FO bk.DAT FO.DAT		
タイプをフィルター(T) すべ インポート先フォルダ(L): Remote	て選択(S) 選択 NativeCOBOL/dat	をすべて解除(D)			参照(W)
オプション 学告を出さずに既存リソースを トップ・レベルのフォルダーを作作 拡張 >>(A)	上書き(O) 				
?	< 戻る(B)	次へ(N) >	終了(F)		キャンセル

BOOKINFO.DAT が DAT フォルダー配下に表示されます。



 * COB... ×
 * プロ...
 * Appl...
 *

 *
 *
 *
 *

 *
 *
 *
 *

 *
 *
 *
 *

 *
 *
 *
 *

 *
 *
 *
 *

 *
 *
 *
 *

 *
 *
 *
 *

 *
 *
 *
 *

 *
 *
 *
 *

 *
 *
 *
 *

 *
 *
 *
 *

 *
 *
 *
 *

 *
 *
 *
 *

 *
 *
 *
 *

 *
 *
 *
 *

 *
 *
 *
 *

 *
 *
 *
 *

 *
 *
 *
 *

 *
 *
 *
 *

 *
 *
 *
 *

 *
 *
 *
 *

 *
 *
 *
 *

 <

3.5 RESTful Web サービスの開発作業

- 1) RESTful Web サービスのプロファイル作成
 - RESTful Web サービスとして利用するビジネスロジックを処理するプログラム「BOOK.cbl」を右クリックし、コンテクストメニューから [新規作成(N)] > [REST Web サービス] を選択します。

BCOB X BTD	0~ ^	aal 💷 🕂 18 🕅 Aaal 🖻		DOOK-HUN		
		新規作成(N)		>	2	COBOL JVM プロジェクト
✓ 2 RemoteNativeCOB✓ 2 COBOL プログラム> 2 BOOK.cbl		開く(O) 表示方法(W) アプリケーションから開く		Alt+シフト+W > >		COBOL JVM ユニット テスト プロジェクト COBOL コピーファイル プロジェクト COBOL プロジェクト COBOL ユニット テスト プロジェクト
と → 🔑 コピーファイル		כול -		Ctrl+C		COBOL/Java 相互運用機能のプロジェクト
> 🗁 dat	Ē	貼り付け		Ctrl+V	e	リモート COBOL JVM プロジェクト
> 🗁 New_Configura	×	削除(D)		削除	(リモート COBOL コピーファイル プロジェクト
	<u>.</u>	コンテキストから除去	Ct	rl+Alt+シフト+下	e	リモート COBOL プロジェクト
		移動(V)				リモート COBOL ユニット テスト プロジェクト
		名前を変更(M)		F2		プロジェクト(R)
		ビルド アクション		>	ď	COBOL コピーファイル
		タスクのスキャン			đ	COBOL プログラム
		指令の確定			B	COBOL ユニット テスト
		プログラムをコピーファイルに変換			Ľ	スタンドアロン ファイル
1		ファイル指令の削除			Ľ	リモート スタンドアロン ファイル
)`		プログラムのフォーマット		Alt+シフト+F	2ŵ	lava インターフェイス
🏗 アウトライン 🗙 🔤 プログ		コピーファイル グラフ			2 \$	REST Web サービス

② REST Web サービスの新規作成ウィザードが表示されます。[Web サービス名] 欄に "BOOKREST" を指定します。[マッピング] 欄は「無し」を選択、[マップするプログラム] 欄には「RemoteNativeCOBOL/BOOK.cbl」が選択されていることを確認し [終了(F)] ボタンをクリックします。



REST Web サービスの新規作成 このページで REST Web サービスを新規作成します								
Neb サービス名: BOOKREST マッピング: 〇 デフォルト ④ 無し マップするプログラム: RemoteNativeCOBOL/BOOK.cbl		参照						
?	終了(F)	キャンセル						

- 2) 書籍データ検索機能オペレーションの作成
 - ① 1) で作成した Web サービスプロファイル「BOOKREST」を右クリックし、コンテクストメニューから [新規作成

웥 COB × ြb プロ 🔮	s Ap	opl 🥌 サーバ 🖳 Anal 🍟 🗖	BOOK.cbl	BOC I:	KREST ×
 ✓ 28 RemoteNativeCOBOI ✓ 20 COBOL プログラム > ② BOOK.cbi > ④ コピーファイル > ③ Web サービス 	L (L	inux:/home/tarot/RemoteNativeCOE	名前		PICTURE
at BOOKREST		新規作成(N)	>	않	COBOL JVM プロジェクト
> ➢ New_Configurati > ➢ repos	×	削除 プロパティ(P) ディブロイ 検査 開く サービスの更新 クライアント生成	削除	计学会 计学校 学校	COBOL JVM ユニット テスト プロジェクト COBOL Jビーファイル プロジェクト COBOL プロジェクト COBOL ユニット テスト プロジェクト COBOL/Java 相互運用機能のプロジェクト リモート COBOL JVM プロジェクト リモート COBOL コピーファイル プロジェクト リモート COBOL プロジェクト リモート COBOL ユニット テスト プロジェクト
🏝 アウトライン 🗙 📰 プログラ	<i>i</i> 47	アウト 🔤 コピーファイル従 " 🗖			プロジェクト(R)
アウトラインを提供するアクティブなエディターはありません。				 □ スタンドアロ □ リモート スタ ● API リソーフ 	スタンドアロン ファイル リモート スタンドアロン ファイル API リソース
				£*	オペレーション

 オペレーションプロパティウィンドウが表示されます。[オペレーション名] 欄に "SEARCHBOOK" を入力して、[OK] をクリックします。

一般	HTTP	ユーザ出口	
オペレーミ	ションは運	髪択されたエントリポイン	小 インタ−フェイスを使用して COBOL プログラムを起動するのに使用されます
オペレージ	ション名:	SEARCHBOOK	
エントリオ	ポイント:	воок	~

ОК	キャンセル

③ [BOOKREST] > [SEARCHBOOK] をダブルクリックします。





- ④ Eclipse IDE メニューから [ウィンドウ(W)] -> [設定(P)] を選択し、[Micro Focus] > [サービスインターフェー
 - ス] > [インターフェイスマッパー] を選択します。[COBOL 割り当てペインの表示] にチェックを入れ [適用して閉じ

る] ボタンをクリックします。

フィルタ入力	インターフェイスマッパー	← ▼ ⇒ ₹ 8
> 一般 ^	インターフェイスマッパー設定	
> Ant		
AspectJ Compiler		
> CSS (Wild Web Developer)	▶ COBOL 割り当 (ハイ)の扱い	
> Gradle		
> HTML (Wild Web Developer)		
> J2EE		
> Java		
> Java 永続化		
JDT Weaving		
> JS/TS (Wild Web Developer)		
> Language Servers		
> Maven		
✓ Micro Focus		
> AWM		
> COBOL		
Enterprise Server		
コンテナー		
✓ サービスインターフェイス		
CICS Web サービス		
インターフェイスマッパー		
デバッグ		
テンプレート		
> データベース		
> ビルダー		
ヘルプ		
リモート JRE		
> 一般		
検索		
> 統合化トレー7機能 >	デフォルトの復元(T)	適用(L)
< >		
		de un vide II
	適用して閉じる	キャンセル

※Preference Recorder のダイアログが表示されたら [キャンセル] を選択してください。

 ⑤ LINKAGE SECTION の COBOL の変数「LNK-FUNCTION」を [COBOL 割り当て] にドラッグ&ドロップ します。





に "1"を設定して [OK] ボタンをクリックします。



 ⑦ [SEARCHBOOK オペレーション - インターフェースフィールド] にて右クリックから[インターフェースグループの新規作 成]を選択し、[名前]に"in_group"と入力し、[OK]ボタンをクリックします。

	名前: in_group
■ インターフェイスグループの新規作成	OCCURS: 0
インターフェイスクループの新規作成	方向: ⑧入力 〇 出力
	場所 ●ボディ ○パス ○ クエリ
	OK キャンセル

⑧ 作成したグループに「LNK-B-STOCKNO」をドラッグ&ドロップします。

🐲 *B	ookrest ×					
LINK	AGE SECTION:		SEARCHBOOK オペレーション	/ • インタ・	-フェイス フィ・	ールド:
名前	i	PICTURE	名前	方向	型	OCC
-	LNK-FUNCTION	X	🛩 / in_group	入力		
\~ E	LNK-B-DETAILS		LNK_B_STOCKNO		string	
>	🗧 LNK-B-TEXT-DETAIL	S				
	LNK-B-STOCKNO	X(4)				
'	🗢 LNK-B-RETAIL	9(5) comp-3				
	LNK-B-ONHAND	9(5) comp-3				
	LNK-B-SOLD	9(5) comp-3				
-	LNK-FILE-STATUS	X(2)				

 ⑨ 同じく [SEARCHBOOK オペレーション - インターフェースフィールド] にて右クリックから[インターフェースグループの 新規作成]を選択し、[名前]に"out_group"と入力し、[方向]を「出力」に変更し、[OK]ボタンをクリックします。

名前:	out_group)
OCCURS	: 0	
方向:	〇入力(●出力
場所	◎ ボディ	○パス ○クエリ
	ОК	キャンセル

① 作成したグループに「LNK-B-DETAILS」と「out_group」をドラッグ&ドロップします。

最終的には下のイメージのような構成になります。

INKAGE SECTION:		SEARCHROOK + M			11.12 •		
LINKAGE SECTION:		SEARCHBOOK JAL	-232-129	-711,7,71	- <i>w</i> r:		
名前	PICTURE	名前	方向	型	OCC	場所	匿名ルート
LNK-FUNCTION	Х	🗸 🖉 in_group	入力			ボディ	Yes
✓		LNK_B_STOC	KNO	string			
> B LNK-B-TEXT-DETA	ILS	✓	出力			ボディ	Yes
LNK-B-STOCKNO	X(4)	✓	ILS				
LNK-B-RETAIL	9(5) comp-3	✓	XT_DE				
LNK-B-ONHAND	9(5) comp-3	omp-3 INK B TITLE					
LNK-B-SOLD	9(5) comp-3	⇔ LNK_B	_TYPE	string			
LNK-FILE-STATUS	X(2)	⇒ LNK_B	_AUTH	string			
		➡ LNK_B_ST	OCKN	string			
		➡ LNK_B_RE	TAIL	integer			
		➡ LNK_B_OI	NHANI	integer			
		➡ LNK_B_SC	DLD	integer			
		➡ LNK_FILE_ST	ATUS	string			
		COBOL 割当て:					
		名前	値				
		LNK-FUNCTION	1				

- 3) 書籍データ追加機能オペレーションの追加
 - 雨度、Web サービスプロファイル「BOOKREST」を右クリックし、コンテクストメニューから [新規作成(N)] > [オ ペレーション] を選択します。



② オペレーションプロパティウィンドウが表示されます。[オペレーション名] 欄に "ADDBOOK" を入力し、[OK] をクリ ックします。

一般	HTTP	ユーザ出口					
オペレー	ションは選	択されたエントリ	ポイント	インターフェイスを使用し	て COBOL プログ	ラムを起動する	のに使用されます
オペレー	ション名	ADDBOOK					
エントリ	ポイント:	BOOK					~

- ③ [BOOKREST] > [ADDBOOK] をダブルクリックします。
- ④ 「LNK-FUNCTION」を COBOL 割当てにドラッグ&ドロップし、[値] には "2" を指定します。
- ⑤ 書籍データ検索機能オペレーションの作成と同様の手順で「in_group」を作成し、「LNK-B-DETAILS」をドラッグ &ドロップします。

OK

キャンセル

最終的には、以下のようになります。

🐲 *BOOKREST ×							
LINKAGE SECTION:		ADDBOOK オ	ペレーション・インタ	ーフェイス フィール	۲ :		
名前	PICTURE	名前	方	向型	OCC	場所	匿名ルート
LNK-FUNCTION	х	v 🖉 in_grou	ip 入	. カ		ボディ	Yes
> 🗗 LNK-B-DETAILS		V 🖉 LNK	_B_DETAILS				
LNK-FILE-STATUS	X(2)	~ <i>[</i> 2] L	NK_B_TEXT_DE				
		F	INK_B_TITLE	string			
		×	LNK_B_TYPE	string			
		E	LNK_B_AUTH	string			
		<u>ا</u> ط	NK_B_STOCKN	string			
		L I I	NK_B_RETAIL	integer			
		⊨ L	NK_B_ONHAN[integer			
		L a	NK_B_SOLD	integer			
		✓	oup 出	カ		ボディ	Yes
		⇒ LNK	_FILE_STATUS	string			
		COBOL 割当	τ:				
		名前	値				
		LNK-FUNCTIO	ON 2				

⑦ Ctrl + S を押して保存します。

3.6 COBOL アプリケーションを Enterprise Server ヘディプロイ

- 1) Enterprise Server へのディプロイ情報を指定
 - ① COBOL エクスプローラーにて追加した Web サービス「BOOKREST」を右クリックし、コンテクストメニューから [プロパティ(P)]を選択します。



			~	E							
🗸 🐸 RemoteN	lativeCOBC)L [Linux:/ho	ome/tarot/R	еі							
V 🖉 COBO											
ספיפי פיי ר–%ר ⊈ו	して.こし レマイル										
√ 💫 Web †	ナービス										
> 😻 BO	OKREST		÷ / N IN								
🗦 🗁 dat		新井丸11Fル	戊(N)								
> 🗁 New_(Configura	¥ 削除									
> 🗁 repos		プロパティ	(P)								
ディプロイメン	トサーバー	〕 9 ブを選	択し、[変勇	• 更] ボタンる	をクリッ	りします	0				
ディプロイメント	サーバー ア	プリケーション	ワァイル エク	ブジット ポイン	ንት እን	ノドラー (ORS 7	゚ロパティ			
Enterprise Se	erver 名:										
									亦田		
									2.L.		
Enterprise	e Server 実	行時環境の	使用								
		Ente	erprise Serv	er 実行時環	環境の相	構成					
		~) <u></u>	5010	1 1 -				
追動済みの 1	interpris	se Server	- TESDEM	1064」を	選択	<u>, [</u> ОК] 不夕.	ンをクリ	ックします	•	
ディプロイ先の Ente	rprise Server	を選択してくださ	きしい:								
サーバー	サービス名	サービス状態	エンドポイント	リスナー状態	説明						
	Deploy	Available	00000	Stopped	De	1					
ESDEIVIO04	Deploy	Available	iocainost:	Started	De	1					
					1						
(?)								OK		キャンセル	
······								OK			
汪怠)		祭に、以下	のダイアログ	が表示され	った際	は、ユー	ザー名	5/パスワ	フードには	t、ESCV	VA 画面
注意) [変更] をク	つめつしてい										
注意) [変更] をク カレた mfs	ecretadr	min の情報	おを入 カレ	てください。							
注意) [変更] をク 力した mfs	ecretadr	min の情報	報を入力し	てください。							
注意) [変更] をク カした mfs	·ウックした。 ecretadr ·名/パスワ	min の情報 ルード	報を入力し	てください。							
注意) [変更] をク カした mfs	·ウックした。 ecretadr ·名/パスワ	min の情報 クード	報を入力し	てください。 ×							
注意) [変更] をク カした mfs ・ ユーザー名と	ecretadr -名/パスワ ニパスワート	<mark>min の情報</mark> ワード 「を入力しる	報を入力し	てください。 ×							
注意) [変更] をク カした mfs ユーザー名と コーザー名と	ecretadr ・名/パスワ ニパスワート	min の情報 ルード 「を入力し」	報を入力します:	てください。							
注意) [変更] をク カした mfs ユーザー名と ユーザー名と	-名/パスワート	min の情報 ロード 「を入力しる	報を入力します:	てください。 ×							
注意) [変更] をク カした mfs ユーザー名と ユーザー名と スーザー名: パスワード:	-名/パスワート - 名/パスワート	min の情報 ロード ^K を入力しる	<mark>服を入力し</mark> ます:	てください。 ×							
注意) [変更] をク カした mfs ユーザー名と ユーザー名: パスワード:	-名/パスワート -名/パスワート	min の情報 ルード 「を入力しる	<mark>服を入力し</mark> ます:								
注意) [変更] をク カした mfs ユーザー名と ユーザー名: パスワード:	・タックした。 ecretadu ・名/パスワート [min の情報 ルード 「を入力し、	報を入力し ます: ャンヤル	てください。 ×							
注意) [変更] をク カした mfs ユーザー名と ユーザー名と パスワード:	・タックした。 ecretadu - 名/パスワート [[min の情報 ルード *を入力し: ーキ	報を入力し ます: ャンセル								
注意) [変更] をク カした mfs ・ ユーザー ユーザー名と ユーザー名: パスワード:	ッタックレル ecretadr -名/パスワート [い い い こ らず、En	min の情報 ルード 「を入力し、 」 ま terprise	報を入力し ます: ャンセル Server 名	てください。 ×	示され	ない場合	合は、」	以下の=	手順のい	ずれかを	実施して
注意) [変更] をク カした mfs ユーザー名と ユーザー名: パスワード:	・クタクレルド ecretadr -名/パスワート レ パスワート	min の情報 レード 「を入力し、 「 terprise	^{服を入力し} ます: ャンセル Server 名	てください。 × 。 。 。 。 。 。 。	示され	ない場合	含は、 」	<u>ፈ</u> ጉወ≣	手順のい	ずれかを	実施して



\$COBDIR/bin/DisableESDefaultSecurity.sh を実行

2) \$COBDIR/etc/mf-client.dat を編集する

[mldap] セクションに username=SYSAD と userpassword=xxxxxxx 項目を追加

いずれも、mfds を再起動してください。

④ [ディプロイする場合はユーザー名/パスワードが必要] にチェックします。

ディプロイメントサーバー	アプリケーションファイル	エグジット ポイント ハンドラー	CORS プロパティ	
Enterprise Server 名	:			
ESDEMO64 (loc	alhost:35783)			変更
Enterprise Server	r実行時環境の使用			
	Enterprise	Server 実行時環境の構成		
EJB ステートフル サ・ サービス名:	-ビスの場合、一部の値	は無視されます		
BOOKREST				高度
トランザクション管理 アプリケーション管 コンテナ管理 	理			
✓ ディプロイする場合		《必要		

⑤ [アプリケーションファイル] タブを選択し、「レガシーアプリケーションをディプロイする」を選択したうえで、[ファイル追加]をクリックします。

ディプロイメントサーバー	アプリケーションファイル	エグジット ポイント ハンドラー	CORS プロパティ
レガシーアプリケーション 〇 レガシーアプリケーシ ディプロイされたアフ	・ パをディプロイ済みか、また パョンは既にディプロイ済み パリケーションのパス:	ー はサーバーにディプロイする必要 、	があるかを指定してください。
◉ レガシーアプリケーシ	/ヨンをディプロイする		
アプリケーションファイル	¢		
			ファイル追加 ファイル削除

プロジェクトディレクトリ配下の「New_Configuration.bin」に生成された「BOOK.gnt」および「BOOK.idy」 を選択し、 [OK] ボタンをクリックします。





選択したファイルが追加されます。

ディプロイメントサーバー アプリケーションファイル エグジット ポイン	トハンドラー CORS プロパティ
レガシーアプリケーションをディプロイ済みか、またはサーバーにディブ(〇 レガシーアプリケーションは既にディプロイ済み	コイする必要があるかを指定してください。
ディプロイされたアプリケーションのパス:	
◉ レガシーアプリケーションをディプロイする	
アプリケーションファイル:	
New_Configuration.bin/BOOK.gnt	ファイル追加
new_configuration.bit/beok.ldy	ファイル削除

2) オリジン間リソース共有(CORS)を許可

デフォルトではオリジン間リソース共有は許可されていません。もしこれに関するエラーが発生する場合、許可設定を行いま

- す。
- [CORS プロパティ]タブを選択し、運用用途に合わせた設定を行います。ここでは「アクセス制限チェックなしでクロス オリジンでリソースのフェッチを許可する(*)」を選択したうえで、[OK]をクリックします。



ディプロイメントサーバ・	- アプリケーションファイル	エグジット ポイント ハンドラー	CORS プロパティ	
オリジンを許可する				
○ クロスオリジン要	求を許可しない(null)		_	
● アクセス制御チェ	ックなしでクロスオリジンで!	リソースのフェッチを許可する (*)		
○ 特定のオリジンカ	いらクロスオリジンでリソースの	のフェッチを許可する		
Allowed Origin:				
公開するヘッダー				

🗌 資格情報を許可する



- 3) RESTful Web サービスを Enterprise Server ヘディプロイ
 - COBOL エクスプローラーにて作成した Web サービス「BOOKREST」を右クリックし、コンテクストメニューから [ディプロイ]を選択します。

ີ 🖧 COB 🗙 🖻 プロ 🧯	e A	opl 📑 サーバ 🛄 Anal
		 ✓ □ ♀ ↓
✓ I RemoteNativeCOB	0 L [L	inux:/home/tarot/RemoteNa
🗸 垣 COBOL プログラム		
> 🖻 BOOK.cbl		
> 追 コピーファイル		
🗸 词 Web サービス		
> 🗱 BOOKREST		新相作成(NI)
> 🗁 dat		4/175C1F782(19)
> 🗁 New_Configurat	×	削除
> 🗁 repos		プロパティ(P)
		ディプロイ

- Eclipse IDE 上からサーバーエクスプローラーを選択します。
 認証画面が表示された場合は、3.3 と同じ情報を入力します。
- ③ 「ESDEMO64」を選択し、右クリックでコンテクストメニューを表示して、[管理ベージを開く] をクリックし、ブラウザー で管理画面を開きます。



诸 COB	₽ <u></u> プロ	😤 Appl	🛛 📕 サーバ	Х	📇 Ar	nal			
						۷	÷	F	000
🗸 🗐 Lin	ux [172.22.2	242.111:100)86]						
v 📕	Default [12	7.0.0.1:86]							
>	🐁 ESDEMO	0							
>	🗏 ESDEMO	064	新規作成(N)					>
			管理ページ	を開	<				

④ [一般]メニューから[サービス]をクリックします。

ES	管理 ダッ	シュボード	ネイティブ	メインフレーム
ネイテ	イブナビゲーション	^	一般丨、	✓ モニター ✓
Υ Ē	♪ グループ		一般的	プロパティ
>	· m理 PAC			コントロール
~ E	Directory Server		月日十 4	検証
~	🖌 🔓 🕀 Default		IĦJXC	リスナー
	🗐 ESDEMO			++ビフ
	民 ESDEMO64	\triangleright	名前*	9-LA

画面を下にスクロールしていくと最下行にディプロイした RESTful Web サービスが表示されます。

8	#SEARCHBOOK	Available	Web Ser	/temppath	MFRHJSON	created 14	Ø	Ū
ଞ	#ADDBOOK	Available	Web Ser	/temppath	MFRHJSON	created 14	Ø	Ū



3.7 Enterprise Server インスタンスへの環境設定と有効化

 変更内容を反映させるために ESDEMO64 の再起動が必要となります。このため、まずは停止を行います。いったん、
 Eclipse IDE に戻り、[サーバーエクスプローラー] ビューの [ESDEMO64] を選択し、マウスの右クリックにてコンテクスト メニューを開き、「停止]をクリックします。

		L	-			
🔓 COB	Ĩ <u>₀</u> プロ	😤 Арр	ol 💄	サーバ	×	📇 Anal
						× =
🗸 🚳 Linu	ıx [172.22.2	242.111:1	0086]			
v 📕 [Default [12	7.0.0.1:8	5]			
>	🛓 ESDEMO	C				
> 2	SDEM	064				
			新	規作成(№	J)	
			管	理ページを	·聞<	
			停	ιĿ		

停止後、アイコンが赤色になったことを確認して、ESCWA 管理画面に戻ってください。

🔓 COB	🔁 プロ	😤 Appl	黒 サーバ
- 0			
🗸 💷 Lini	ux [172.22.2	242.111:100	86]
v 📕	Default [12	7.0.0.1:86]	
>	🐁 ESDEMO	C	
>	🛃 ESDEMO	D64	

2) データファイルへの環境変数設定

① [一般]をクリックします。

ES	管理	ダッシュ	ボード	ネイティブ
ネイテ	ィブナビゲー	ション	^	一般 丨~
、口 、 、	D グループ 論理			サービス
→ ~ 目	PAC Directory Ser	ver		サービス
~	r 🔓 🔓 Det ESDEI	fault MO		
	🗟 ESDEI	M064	\triangleright	▼ 名前
> Ę	SOR			

② [構成情報] に以下の情報を設定したうえで、画面上部の [適用] をクリックします。
 [ES-Environment]

BOOKINFO=/home/tarot/RemoteNativeCOBOL/dat/BOOKINFO.DAT



一般的なプロパティ 適用	<u></u> 前除
開始オプション	* 入力必須の項目です
名前* ♀ ESDEMO64	
共有メモリページ数 💡 512 🍾 ページ数(4k):	共有メモリ クッション 💡 32 ^ ページ数(4k):
SEP数 0 2 [^]	コンソール ログ サイズ 💡 0 k
🗌 ローカル コンソールを表示 💡	🗌 動的デバッグを許可 💡
🗌 システム起動時に開始する 💡	☑ 64ビット作業モード 💡
□ 以前のログを削除 🛛	

追加設定

構成情報 ♀ [ES-Environment] BOOKINFO=/home/tarot/RemoteNativeCOBOL/dat/BOOKINFO.DAT

- 3) リスナーポートの設定
 - ① [一般] > [リスナー] をクリックします。



 を下より [Web Services and J2EE] をクリックし、右側の [ポート] に "9003" を入力したうえで、[適用] を クリックします。



ES 管理 ダッ	シュボード	
ネイティブ ナビゲーション	^	
 ○ クループ > 論理 > PAC > 団 Directory Server < 合 ⊕ Default 		
> 🛱 SOR		
 ■信ジーパーの和扱作成 ● 通信ブロセス1 ● パ Web ■ パ Web Services ar リスナー プロパティ 	nd J2EE 適用	
* 入力必須の項目です 名前* Q Web Services and J2EE		🗌 レガシー Micro Focus アブリケーション形式 9
□ このエンドホイントはネットワ プロトコル ♀	ッーク経田でアクセス可能になり、TLSが無効になり。 ホスト名またはIP アドレス* ♀	kg。 ポート 9
tcp	*	9003

4) 不要ログの停止

2

3

~ 🔓 🖯 Default ESDEMO B ESDEMO64

① [Directory Server] > [Default] をクリックします。

ES 管理 / ダッシュボード
ネイティブ ナビゲーション ^
◇ □ グループ
> 論理
> PAC
✓
✓ 🔓 🕀 Default
ESDEMO
B ESDEMO64 ▷
[プロパティ] をクリックします。
ネイティブナビゲーション ^ リージョンおよびサーバー プロパティ 🗸 セキュリテ
✓ ☐ グループ このDirectory ServerホストではTLSが有効ではありません。データはローカ
> 論理 Sirectory ServerにMTO ライセンスがありません。ライセンス状態を確認し
 > PAC > III Directory Server Uージョンおよびサーバー リスト * 新規作成

▼ 名前

一番下にスクロールし、[モニター] 欄の [有効] のチェックを外し、[適用] をクリックします。

 \triangleright

▼ 説明



	リージョンおよびサーバー	プロパティ 🛛 🗸	セキュリティ	✓ ジャーナ	ナノレ	
£	このDirectory ServerホストではTLS	が有効ではありません	。データはローカルネ	ットワーク上で送	信されますが、他の接続が公開	され
es.	Directory ServerにMT0 ライセンス;	がありません。ライセン	ンス状態を確認してくた	ごさい。一部の機能	は使用できなくなります。	
D	irectory Serverの構成	適用				
	□ ルーノバック000%をシック 最大オブジェクト数 ♀			A	🗌 すべてのリスナーが	起動
	1024				□ サーバーの停止時に	動的
	デフォルトプロセスユーザーID(Ş			🔲 レガシー Micro Focu	sサ
	root				🗌 システム サーバー タ	い
	✓ タイムアウト UI セッショ	UIセッ 600	ションタイムアウト へ ↓ 利 ↓	B	✓ タイムアウト API セ	ッシ
	ジャーナル					
	□ ctf トレース §	দেশা বৃ⁄	✓ ✓	~	ファイルの最大サイズ 9 512 ^	kB
	モニター					
	口 有効 🖓	キーフ 60	プアライブ間隔 💡 へ ¹ ¹ ¹	þ	許容応答時間 💡 5 🍦 🤅	秒

- 5) Directory Server の開始
 - ④ Eclipse IDE に戻り、[サーバーエクスプローラー] ビューの [ESDEMO64] を右クリックにてコンテクストメニューを 開き、[開始] をクリックします。

🗄 COB	🔁 プロ	😤 Appl	🔳 サーバ	×	📇 Ar	al			E
						۷	÷	-	ş
🗸 🚳 Linu	x [172.22.2	42.111:10	086]						
~ 📕 C	efault [12	7.0.0.1:86]							
> 🚪	SDEMC)							
> 🚪	SDEMC	64	新規作成(N)						>
			管理ページを	, 盟く					
			開始	713 X					

3.8 RESTful Web サービスのテスト

- テストクライアントアプリケーションによる確認
 製品には、登録したサービスのテストクライアントアプリケーションを自動生成する機能が搭載されています。こちらを利用して、
 書籍データ検索機能オペレーションのテストを行います。
 - ① Eclipse IDE に戻り、[COBOL エクスプローラー] をクリックします。





② [BOOKREST] を選択し、マウスの右クリックにてコンテクストメニューを開き、[クライアント生成] をクリックします。



③ [ナビゲーター上の現在のプロジェクト]を選択し、[OK]をクリックします。

○ ノロシェクトなし	51	
● デビクータ上の現在のノロシエク		
 ・ 開いている Eclipse ノロシエクト ・ のののに、プロンドートレナが、提供に ・ 		
 COBOL JUジェクトを新規作り 	成	
生成されたクライアント用のプロジ:	ジェクト: RemoteNativeCOBOL 通	択
出力ディレクトリ:	/home/tarot/RemoteNativeC 参	照
□ JSON/YAML を使用		
JSON/YAML ファイル:		照
オペレーション		
すべてのオペレーション		
- 7 1 X		
文字列サイズ:	100	
インライン配列サイズの制限:	個別の構造体なし(0)	
配列サイズ:	100	
→文字集合	ビット モード	
	● 32 ○ 64	
	OK キャンセル	

クライアントプログラムが生成され、自動的にビルドされます。

BOOKREST-app.cbl がテストクライアントプログラムになります。



ľ	😪 COB × 陷 プロ 😤 Appl 📕 サーバ
ł	
Ŀ	 RemoteNativeCOBOL [Linux:/home/tarot
L	✓ / ✓ COBOL プログラム
Ł	✓ ➢ client
1	BOOKREST-app.cbl
L	BOOKREST-proxy.cbl
L	> 🖻 BOOK.cbl
Į.	> 🔑 コピーファイル
Ł	✓ G Web サービス
L	> 😻 BOOKREST
Ł	> 🗁 client
Ł	> 🗁 dat
L	✓ ➢ New_Configuration.bin
L	BOOK.gnt
L	BOOK.gnt.1.tlog
L	DOOK.idy
L	BOOK.objlist
	BOOKREST-app.gnt
i I	

④ リモートサーバーにターミナル上でログインを行い、以下のコマンドを実行します。

. /opt/microfocus/VisualCOBOL/bin/cobsetenv

\$. /opt/microfocus/VisualCOBOL/bin/cobsetenv COBDIR set to /opt/microfocus/VisualCOBOL

- ⑤ リモートサーバー上の RemoteNativeCOBOL/New_Confugiration.bin ディレクトリにターミナル上で移動します。
- ⑥ 以下のコマンドを実行します。

cobrun BOOKREST-app.gnt

次の入力を行ってください。

「Service Address (Enter = http://localhost:9003):」では、そのまま Enter キーを押します

```
「Supplemental Query String (optional):」では、そのまま Enter キーを押します
```

「Username (optional):」では、そのまま Enter キーを押します

「Password (optional):」では、そのまま Enter キーを押します

「Operation (1 = SEARCHBOOK, 2 = ADDBOOK):」では"1"を入力します

「.LNK_B_STOCKNO:」では、"1111" を入力します

```
Supplemental Query String (optional):
Username (optional):
Password (optional):
Operation (1 = SEARCHBOOK, 2 = ADDBOOK): 1
Body Parameters:
:
.LNK_B_STOCKNO: 1111
:
.LNK_B_DETAILS:
..LNK_B_DETAILS:
..LNK_B_TEXT_DETAILS:
..LNK_B_TITLE: LORD OF THE RINGS
...LNK_B_TYPE: FANTASY
...LNK_B_AUTHOR: TOLKIEN
```



まずは、既存データがない事を検索機能で確認します。

curl http://IP アドレス:9003/temppath/BOOKREST/1.0/SEARCHBOOK -d

@json¥search9999.txt





続いて、追加機能でデータを追加します。

curl http://IP アドレス:9003/temppath/BOOKREST/1.0/ADDBOOK -d @json¥add9999.txt



検索機能で、正しく追加されたことを確認します。

curl http://IP アドレス:9003/temppath/BOOKREST/1.0/SEARCHBOOK -d

```
@json¥search9999.txt
```



3.9 RESTful Web サービスのデバッグ

1) サーバー側のデバッグ設定



① Eclipse IDE に戻り、[サーバーエクスプローラー] をクリックします。



② [ESDEMO64] を選択し、マウスの右クリックにてコンテクストメニューを開き、[停止] をクリックします。

🔓 COB	陷 プロ	😤 Appl	💻	サーバ	\times	📇 Anal
						× .
🗸 🚳 Linu	ıx [172.22.2	42.111:10	0086]			
v 📕 [Default [12]	7.0.0.1:86]			
>	🛓 ESDEMC)				
> 1	SDEMC	64	£6.‡E	₩E FÜ (NI)		
			有用の	51F/25(1N) ∃∧°_≥%≠5		
				- / /	πIX	
			1루표			

停止すると、赤色のアイコンが表示されます。



③ ESCWA 画面に戻り、左下より [ESDEMO64] をクリックします。

この時、認証画面が再度表示されることがありますので、前回と同じ情報を入力してください。

ES	管理 ダッシュボード
ネイテ	イブナビゲーション ^
✓ É > >	♪ グループ › 論理 > PAC ■ Directory Server
~	🖌 🎧 🔓 Default
	ESDEMO
> 6	⊅ SOR

④ [動的デバッグを許可] にチェックを行い、[適用] をクリックします。



一般 🗸	
一般的なプロパティ 適用	<u></u> 削除
開始オプション	* 入力必須の項目です
_{名前*} ♀ ESDEMO64	
共有メモリページ数 💡	共有メモリ クッション 💡
512 ~ ページ数(4k):	32 、ページ数(4k):
sep数 🛛	コンソール ログサイズ 💡
2	0
🗌 ローカルコンソールを表示 💡	☑ 動的デバッグを許可 ♀

⑤ Eclipse IDE に戻り、[ESDEMO64] を選択し、マウスの右クリックにてコンテクストメニューを開き、[開始] をクリックします。

🔓 COB	🔁 プロ	e : A	Appl	📕 サーバ	×	📇 Ar	nal		
✓ @ Linu ✓ 📕 I	ıx [172.22.2 Default [12	42.11	1:1008 1:86]	36]		-	~	÷	000
>	ESDEMC)							
>	📇 ESDEMC	064	÷	新規作成 管理ページ	(N) を開く	(>
			[開始					

開始すると、緑のアイコンが表示されます。

🔓 COB 📔	プロ	🔁 Appl	黒 サ−バ×
🗸 🚳 Linux [172.22.2	242.111:100	861
v 📕 Def	ault [12	7.0.0.1:86]	-
> 📇 E	SDEM	C	
> 法 E	SDEM	D64	

- 2) デバッガーのアタッチ
 - ① [COBOL エクスプローラー] をクリックします。



② RemoteNativeCOBOL プロジェクトを選択し、マウスの右クリックにてコンテクストメニューを表示した上で、[デバッグ(D)] > [デバッグの構成(B)] をクリックします。



🕆 🕆 🖓 🖧 🖧 🖧 🕆 🔁 🔁 🔁		ATT2011 /00(14)		
		表示方法(W)	Alt+シフト+W >	
 ✓ COBOL プログラム ◇ ゆ COBOL プログラム ◇ ゆ コピーファイル ◇ 喩 Web サービス >> 部 BOOKREST 		 コピー 貼り付け 削除(D) コンテキストから除去 移動(V) 名前を変更(M) 	Ctrl+C Ctrl+V 削除 Ctrl+Alt+シフト+下 F2	
 Image: Second Sec		ビルド アクション タスクのスキャン 指令の確定 ファイル指令の削除 コード分析	>	-
BOOKREST-app.gnt BOOKREST-app.gnt.1.tlog	4	インボート(i) エクスポート(O)…	>	
 BOOKREST-app.idy BOOKREST-app.objlist BOOKREST-proxy.gnt POOKREST-proxy.gat 1 ligg 	\$	更新(F) プロジェクトを閉じる(S) 無関係なプロジェクトを閉じる(U)	F5	
<		Source	>	
語 アウトライン × 国プログラム アウト 国 コピ	Q. O	リモートシステムビューで表示 Coverage As 実行(R)	>	Diffe (- Po Tabla Dogulto - 87 Fi
アリトノコノを提供するアクティノはエアイターはめり	*	デバッグ(D)	>	デバッグ の構成(B)

③ 画面左側より、[COBOL Enterprise Server] をダブルクリックします。

構成の作成、管理、および実行

Enterprise Server アプリケーションへの接続とデバッグ



	このダイアログから起動設定を構成します:
フィルタ入力	『 - 選択した種類の構成を作成するには、「新規構成」ボタンを押します。
🗄 Apache Tomcat 🔨	- 選択した種類の起動構成プロトタイプ「新規プロトタイプ」ボタンを押します。
AspectJ/Java Application	◎ - 選択した構成をTクスポートするには、「Tクスポート」ボタンを押します。
🔤 AspectJ Load-Time Weaving Ap	
Chrome Debug	11回 - 選択した構成をコピーするには、「複製」ホタンを押します。
國 COBOL/Java 相互運用機能のアン	🔀 - 選択した構成を削除するには、「削除」ボタンを押します。
SCOBOL Enterprise Server	
🏧 COBOL JVM アプリケーション	

④ [Enterprise Server] セクションの [参照] をクリックして、Linux > ESDEMO64 を選択します。選択後は、以

下のように表示されます。

名前(N): 新規構	構成	
💦 一般 🦭 ソー	-ス 🗉 共通 🧤 デバッグシンボル 🥒 コンテナー	
Enterprise Serv	ver 上でデバッグセッションを開始して、COBOL プログラムの起動を待機します。	
▼ COBOL プロミ	ジェクト(P)	
RemoteNat	tiveCOBOL	参照
	Server	
接続: サーバ	「- エクスプローラ-	\sim
サーバー エ	エクスプローラーの設定	
ESCWA:	Linux	
MFDS:	Default	
リージョン	ESDEMO64	
	·	参照

⑤ 画面を下にスクロールを行い、[デバッグの種類] セクションで [Web サービス] タブを選択し、[Web サービス名] が空欄になっていることを確認したうえで、[デバッグ(D)] をクリックします。



名前(N): 新規	見構成		x. 1911 a my m 1					
▶ 一般 ▶	ノース 共う 重類	直 ≒∕/ デバックシ	/ンホル 🥔 コンテナー					^
タイプ: ヽ	Neb サービス							\sim
Web サ	ービス名(3	2白の場合はす/	べてのサービスをデバッ	グ)				1
完全修	節の Web t	ナービス名 例 htt	p://tempuri.org/pa	ackage#oper	ation			
クライア ① すべ	ントのリクエス て	ストにより起動され	いた Web サービスのう	デバッグ				
○ IP 7	יドレス:	0.0	. 0 . 0					
○ ホス	卜名:	localhost						~
					前回保管	した状態に戻す(\	/) 適用(Y)
						デバッグ(D)	閉じ	` る

以下のダイアログが表示された場合は、[はい(Y)]をクリックします。

この種類の起動では、開始時にデバッグ パースペクティブを開くように構成します。
このデバッグ・パースペクティブは、アブリケーションのデバッグをサポートするように設計されています。これに は、デバッグ・スタック、変数、およびブ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。
今パースペクティブを開きますか?
□ 設定を保存
はい(Y) いいえ(N)

⑥ さきほどの curl コマンドを用いたリクエストを行います。以下のコマンドをコマンドプロンプト上で実行します。
 curl http://IP アドレス:9003/temppath/BOOKREST/1.0/SEARCHBOOK -d
 @json¥search1111.txt
 「IP アドレス」は、環境に合わせて変更してください。

TIP プトレス」は、現現にロわせて変更してんてい。

 ⑦ Eclipse IDE に戻ると、デバッガーが起動し、プログラムが停止しています。
 通常のコンソールアプリケーションのデバッグと同様、[実行] メニュー配下の [再開] や [ステップイン] などといった 操作が利用できます。

除デバッグ × 迠 プロジェクト・エクスプローラー 🖏 サーバー 🗧 🗆	81	BOOKRE	ST 🖻 BOOK.cbl ×		(¥)= 変数 ×	🎭 ブレークポイント	📧 プログラム アウトラ.
E 🙀 🔛 🗄		BOC	K.cbl 🕨				
- 🔄 新規構成 [COBOL Enterprise Server]		·····*A·1·					值
✓ → Micro Focus デバッガ: (一時停止)				^	> 🔶 LNK	B-TEXT-DETAILS	
🗸 🧬 アプリケーション スレッド: 2422 (一時停止)					LS-C	ALL-STATUS	08224
/opt/mf/VC100PU01/deploy/BOOKREST.ANtz70	E	Θ	PROCEDURE DIVISION USING LNK-FUNCTION				
🔗 アプリケーション スレッド: 2421 (一時停止)		Θ	LNK-B-DETAILS				
🧬 アプリケーション スレッド: 2476 (一時停止)			LNK-FILE-STATUS.				
		9	MAIN SECTION.				
			CALL "CDL TOUDDED" HETNE LNK D TEXT DETATLE				
		CAL	CALL CBE_TOOPPER OSTING ENK-B-TEXT-DETAILS				
			BY VALUE LENGTH LNK-B-TEXT-DETAILS				
			RETURNING LS-CALL-STATUS				
			EVALUATE TRUE				
			WHEN READ-RECORD				
			PERFORM DO-READ-RECORD				

[ステップイン] にて、1 行毎にプログラムを進めることができます。

[再開] などで、デバッグが終了すると、コマンドラインに結果が戻されます。

C:¥vc-tutorial-rest02>curl http://172.22.252.244:9003/temppath/BOOKREST2/1.0/SEA RCHBOOK -d @json¥search1111.txt





8

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	ブロジェクト(P) 実行(R)	UP	冉開(M)	F8
≂ © ⊨ s ⊡ : ≥ ⊳ n ⊲ ∠ : ⊟ : ∂ □ − f=	🐨 : 🔏 🔲 🖏 😨 🗆 1		中断(S)	
			終了(T)	Ctrl+F2
除すハック× № ノロシェクト・エクスノローラー 報 サーハー レ	S BOOKREST	14	切断	
E 🖗 🗗 🗄	BOOK.cbl 🕨	3	ステップイン(I)	F5
 新規構成 [COBOL Enterprise Server] 	····•*A·1·		ステップイン選択	Ctrl+F5
豂 Micro Focus デバッガ: (アタッチ待機)		Q.	ステップ・オーバー(O)	F6

3.10 サーバーの停止

- 1) Enterprise Server インスタンスの停止
 - ① 画面右上のアイコンより、「COBOL エクスプローラー」ビューに切り替え、「サーバーエクスプローラー」に切り替えま す。





② ESDEMO64 上で右クリックし、コンテクストメニューから [停止] を選択し、Enterprise Server インスタンスを停



2) 以下のコマンドを実行します。

コマンド引数の情報は、先に取得・使用していた認証情報です。

> Enterprise Server Common Web Administration (ESCWA) サービスの停止

```
escwa -p SYSAD 6YaafasP
```

π cscwa p SISAD Oladia.	
2024-10-10 11:41:22.740	Loaded COBOL Run Time Environment Extension
GkCobExInit at 0x7fc041ca	2e29
2024-10-10 11:41:22.740	New thread high-water mark: 1 threads are now running
2024-10-10 11:41:22.740	MFCS server "ESCWA" running as process 9891
2024-10-10 11:41:22.740	GK-OS version 2.11.2
2024-10-10 11:41:22.740	GK-Utility version 2.11.3
2024-10-10 11:41:22.740	GkCobGetFuncAddr: 4
2024-10-10 11:41:22.741	ES Common Web Administration version: 6.1.1
2024-10-10 11:41:22.741	/opt/mf/VC100PU01/etc
2024-10-10 11.41.22 742	Shutting down ESCWA instance a

> Micro Focus Directory Server の停止

mfds /s 1 SYSAD 6YaafasP

```
# mfds /s 1 SYSAD 6YaafasP
Processing -s option...
Copyright 1991-2024 Micro Focus.
Micro Focus Directory Server daemon: Version 1.30.26
Request sent...[root@Ora9temp WITHXAES.deploy]# PR7007I Changing effective process uid
to "tarot"PR0007I Process is using UID 1000
[2]+ 終了 mfds (wd: /home/tarot/withxa/repos/WITHXAES.deploy)
(wd now: /home/tarot/withxa/repos/WITHXAES.deploy)
```

免責事項

ここで紹介したソースコードは、機能説明のためのサンプルであり、製品の一部ではございません。ソースコードが実際に動作するか、御社業務に適合するかなどに関しまして、一切の保証はございません。 ソースコード、説明、その他すべてについて、無謬性は保障されません。 ここで紹介するソースコードの一部、もしくは全部について、弊社に断りなく、御社の内部に組み込み、そのままご利用頂いても構いません。 本ソースコードの一部もしくは全部を二次的著作物に対して引用する場合、著作権法の精神に基づき、適切な扱いを行ってください。