Micro Focus Enterprise Developer チュートリアル

メインフレーム COBOL 開発: JCL

Eclipse 編

1. 目的

本チュートリアルでは、Eclipse を使用したメインフレーム COBOL プロジェクトの作成、コンパイル、JCL の実行、デバッグまでを行い、その 手順の習得を目的としています。

2. 前提

- 本チュートリアルで使用したマシン OS : Windows 8 Enterprise
- 使用マシンに Micro Focus Enterprise Developer 2.3 for Eclipse がインストールされていること

3. チュートリアル手順の概要

- 1. チュートリアルの準備
- 2. Eclipse の起動
- 3. メインフレーム COBOL プロジェクトの作成
- 4. プロジェクトプロパティの設定
- 5. ビルドの実行
- 6. Enterprise Server の設定
- 7. Enterprise Server の開始と確認
- 8. JCL の実行
- 9. プロシージャライブラリの作成
- 10. COBOL バッチプログラムの実行
- 11. COBOL バッチプログラムのデバッグ
- 12. Enterprise Server の停止



3.1 チュートリアルの準備

例題プログラムに関連する資源を用意します。

- 使用する例題プログラムは、キットに添付されている Tutorials.zip に圧縮されています。これを C:¥ 直下に解凍します。
 > ローカル ディスク (C:)
 - ▲ 名前
- 2) Eclipse のワークスペースで使用する「work」フォルダを C:¥ 直下に作成します。

3.2 Eclipse の起動

1) Micro Focus Enterprise Developer for Eclipse を起動します。

Enterprise Developer for Eclipse	👃 Adobe Reader XI
Enterprise Developer for Visual Studio Enterpr	ise Developer for Eclipse の起動
「「「キュメント - Eclipse	Micro Focus License Manager
😯 ドキュメント - Visual Studio 2012	🌗 Micro Focus Enterprise Developer 💦

2) 前項で作成した「C:¥work」をワークスペースへ指定して、[OK] ボタンをクリックします。

M.	ワークスペース・ランチャー		×
ワークスペースの選択			
Eclipse は、ワークスペースと呼ばれるフォル このセッションに使用するワークスペース・フォノ	ダにプロジェクトを保存します。 レダを選択してください。		
ワークスペース(<u>W</u>): C:¥work		✓ 参照	l(<u>B</u>)
□この選択をデフォルトとして使用し、今後。	の質問を表示しない(<u>U</u>)	OK ‡P	ンセル

3) [ようこそ] タブが表示されたら [Open COBOL Perspective] をクリックして、COBOL パースペクティブを開きます。

Ш		Team D	eveloper - Eclij	pse 💶 🗖 🗙
771	「ル(E) 編集(E) ナビゲート(<u>N</u>) 検索 プロジェクト(<u>P</u>) 実行(<u>B</u>) ウインドウ(<u>W</u>) ハ	∨レプ(圧)	
8	83 3554 88			
60	D M	CRO		ß
	100	LUS		
		Enterprise Team Develo	per for	Eclipse にようこそ
		概要		ファースト・ステップ
		フィーチャーの概要		ファースト・ステップの開始
		Web リソース		マイグレーション
	(E)	Web 上の詳細情報		新規リリースへのマイグレーション
			1	
	ID.	Open COBOL Perspective COBOL パースペクティブを開きます。このパースペクティブでは、	D	Open Team Developer Perspective Team Developer パースペクティブを開きます。このパースペクティブア
		COBOLアプリケーションを開発するためのツールを提供します。		は、メインフレームのリソースへのアクセスを物略化するためのシールを提 (単)ます。
			1	
	-	Open PL/I Perspective		
	- u	PL/Iパースペクティブを開きます。このパースペクティブでは、PL/I リケーションを開発するためのツールを提供します。	アブ	
ME I				

MICRO[®] FOCUS

4) パースペクティブ表示後、[プロジェクト] プルダウンメニューの [自動的にビルド] を選択して、これをオフにします。



3.3 メインフレーム COBOL プロジェクトの作成

1) 用意したサンプルソースをインポートします。 [ファイル] プルダウンメニューから [新規] > [メインフレーム COBOL プロジェ クト] を選択します。

MF								
ファイ	Ί μ(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)	へルプ(H)
	新規(N)			Alt+シフト+N	۲	COBOL JVM プロ	コジェクト
	ファイル	を開く(.)				12	አፈንጋቦ-7 co	BOL プロジェクト

2) [プロジェクト名] は任意ですが、ここでは "JCLDEMO" を入力して [終了] ボタンをクリックします。

🖻 メインフレーム COBOL プロジェクトの新規作成 – 🗆 🗙
メインフレーム cobol プロジェクト
CICS®、JCL または IMS® アプリケーションを作成するためのプロジェクトです。
70岁ェクト名(P): JCLDEMO
プロジェクト テンプレートを選択
🖾 Micro Focus テンプレート
テンフレートの設定を構成
場所:
✓ デフォルト・ロケーションの使用(D)
ロケーション(上): C:¥work¥JCLDEMO 参照(R)
ファイル・システムを選択(Y): デフォルト V
ファイル・システムを選択(い): デフォルト >
ファイル・システムを選択(Y): デフォルト \vee
ファイル・システムを選択(Y): デフォルト \vee
ファイル・システムを選択(Y): デフォルト ∨ (?) 終7(F) キャンヤル

- 3) [COBOL エクスプローラー] へ作成したプロジェクトが表示されます。
 - 🔓 COBOL エクスプローラ 👷

🐸 JCLDEMO



- 4) プロジェクトを作成したことにより C:¥work¥JCLDEMO フォルダが作成されています。このフォルダ配下に JES 機能で使用するフォルダをあらかじめ用意しておきます。
 - ① カタログファイルやスプールファイルを配置するため "DATAFILE" フォルダを C:¥work¥JCLDEMO 配下へ作成します。
 - ② プロシージャファイルを配置するため、プロシージャライブラリとして "SYS1.PROCLIB" フォルダを C:¥work¥JCLDEMO¥DATAFILE 配下へ作成します。

ク(C:)	$\mapsto \text{ work } \mapsto \text{ JCLDEMO } \mapsto \text{ DATAFILE } \mapsto$
^	名前
	SYS1.PROCLIB

5) 既存ファイルのインポート時、自動的にコンパイル指令が指定される機能が用意されていますが、本チュートリアルではこれを 解除します。 [ウィンドウ] プロダウンメニューの [設定] > [Micro Focus] > [COBOL] > [指令の確定] > [指令の 確定を行う] チェックボックスをオフにして [OK] ボタンをクリックします。

ME		設定 -
フィルタ入力		指令の確定
 Data Management Java Java EE Java Persistence Java Script JDT Weaving Maven Micro Focus COBOL デバッグ エディタ ニド分析 プDアイラ 指令の確定 単一アテイルをコンド COBOL デバッグ 	^	指令の確定の設定 ファイルのスキャン時に設定する指令を選択します。 ファイルには、プシェクト設定と異なる指令のみが設定されます。 □指令の確定を行う 万言 ▼ ファイルは DIALECT 指令を設定する ■ アンテイルに DIALECT 指令を設定する ■ EXEC SQL を含むファイルに SQL 指令の設定を解除する ■ EXEC SQL を含むファイルに SQL 指令の設定を解除する ■ EXEC CICS を含むファイルに CICS 指令の設定を解除する ■ EXEC CICS を含まないファイルの CICS 指令の設定を解除する

6) 用意したサンプルプログラム類をインポートします。 [JCLDEMO] プロジェクトを右クリックして [インポート] > [インポート] > [インポート] > を選択し、インポートウィンドウにて [General] > [ファイル・システム] を選択後 [次へ] ボタンをクリックします。

「厨」 インポート	– 🗆 🗙
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。	Ľ
インポート・ソースの選択(<u>S</u>): フィルタ入力	
Ceneral	^
◎ 照存 ブロシェクトをワークスペースへ	×
(P) (P) 終了(E)	キャンセル



 "C:¥Tutorials"を [次のディレクトリーから] へ指定すると内容が表示されますので、最後から 4 ファイルをオンにして [終了] ボタンをクリックします。この実行により、プロジェクトフォルダヘサンプルプログラムが配置されます。

Te:	インボート	- 🗆 🗙
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:¥Tutorials		▼ 参照(<u>R</u>)
P ● Tutoriels タイプを2イルター(1) ダイて進択(5) インボート先カルダ(1): IンEDEMO オブラコ> 警告を出まずに取得リソースを上書き(2) トップ・レベルのフォルターを作成(2) 証拠 >> (A)	ACCT04.cbl ACCT04.cbl ACCT08.cc.py AcCT08.cov ACCT08.cov	◆ 毎年(⊻)
0	< 戻る(<u>B</u>) 次へ(<u>N</u>) > 終了 (<u>(E)</u> キャンセル

8) [COBOL エクスプローラー] 内に表示されている [JCLDEMO] にインポートしたファイルが表示されていることを確認して、 [SORTD.prc] ファイルを作成した [SYS1.PROCLIB] ヘドラッグしてドロップします。プロシージャファイルが [SYS1.PROCLIB] フォルダ配下へ移動しました。

৳ COBOL エクスプローラ ⋈ 🚪	<mark>ೀ</mark> COBOL ±クスプローラ ೫ 📃
 ✓ JCLDEMO ▷ (○) COBOL プログラム ▷ (○) JCL ファイル △ (○) DATAFILE ○ SYS1.PROCLIB ○ CODED prote 	 ■ JCLDEMO ■ COBOL プログラム ● 図してファイル ■ DATAFILE ■ SYS1.PROCLIB
SORTD.prc	SORTD.prc

3.4 プロジェクトプロパティの設定

プログラム内容に沿ったプロジェクトのプロパティを設定します。

1) [COBOL エクスプローラー] 内の [JCLDEMO] プロジェクトを右クリックして [プロパティ] を選択します。



 左側メニューの [Micro Focus] > [ビルド構成] > [COBOL] を選択して、下記項目を指定します。指定後は [OK] ボタンをクリックしてください。

項目名	説明
ターゲットの種類	実行ファイル形式を指定します。ここでは [全て INT/GNT ファイ ル] を選択します。
プラットフォーム ターゲット	稼働ビット数を指定します。ここでは [64 ビット] を指定します。

T <u>WF</u>	プロパティ: JCLDEMO	- 🗆 🗙
フィルタ入力	COBOL	↓ ↓ ↓ ↓
▷ リソース ▲ Micro Focus ビルドパス ▲ ビルド構成 BMS	New Configuration [使用中]	✓ 構成の管理
COBOL CICS プリプロ SQL プリプロセ コード分析	出力パス: New Configuration.bin エントルポイント:	参照
」 アセンブラ コンパイ・ アセンブラ コンパイ・ アセンブラ リンカ イベント > リンク	ターゲット設定 ターゲットの種類 すべて INT/GNT ファイル ✓ ○ 32 ビット ● 64 ビット	
 > プロジェクト設定 > 実行時構成 Project Facets > Task Repository Task Taos 	▶ プロジェクトの COBOL の設定の上書き	
Validation		デフォルトの復元(I) 適用(L)
?		OK キャンセル

3) 再度プロパティウィンドウを開き、左側メニューの [Micro Focus] > [プロジェクト設定] > [COBOL] を選択して、下記 項目を指定します。指定後は [OK] ボタンをクリックしてください。

項目名	説明
文字集合	EBCDIC または ASCIIを指定します。ここでは [ASCII] を選択します。
	COBOL 言語方言を指定します。
言語方言	サンプルプログラムは IBM Enterprise COBOL の方言を使用しているため、ここでは
	[Enterprise COBOL for z/OS] を指定します。
デバッグ用にコンパイル	デバッグ実行時に使用するファイルを生成するように指定します。
.GNT にコンパイル	実行ファイル形式を GNT に指定します。
追加指令	ここでは指定しません。

ME .	プロパティ:	JCLDEMO	- 🗆 🗙
フィルタ入力 トリソース	COBOL		<> ▼ <> ▼ ▼
 D J へ Source So	 文字集合: (注語方言: ンースフォーマット: メインフレームのコピー処理: (□リストプァイルを生成 (□リストプァイルを生成 (□ブラバッグ用にコンパイル(D)) (マ) 「GNT (ニコンパイル) (マ) 「GNT (ニコンパイル) (マ) 「(マ) 「(マ) 「(マ) 「(マ) 「(マ) (マ) (マ) (マ) (マ) (マ) (マ) (マ) (マ) (マ)	ASCII Enterprise COBOL for z/OS 回走 COPY □ コード カバレッジを有効にする ブロファイラを有効にする ブロファイラを有効にする 回復可能なエラーを含める(レベル E) 100	
?		ОК	キャンセル

3.5 ビルドの実行

1) [プロジェクト] プルダウンメニューの [自動的にビルド] を選択して、これをオンすると自動的にビルドが実行されます。



2) [コンソール] タブで成功を確認します。



- 3) [COBOL エクスプローラー] のプロジェクト内に存在する [New_Configuration.bin] フォルダ配下に実行ファイル (.gnt ファイル) が作成されていることを確認してください。
 - ■ JCLDEMO
 ■ GCOBOL プログラム

 ■ JCL ファイル

 ■ DATAFILE

 ■ New_Configuration.bin

 ■ KSDSWRT2.gnt

 ■ KSDSWRT2.gnt.1.tlog



3.6 Enterprise Server の設定

Enterprise Server には JCL をエミュレーションする機能が搭載されており、この開発用サーバーを使用してメインフレームアプリケーションの実行やデバッグを行います。マイグレーションにおいては本番実行用の Enterprise Server 製品を使用します。

1) Enterprise Server を作成します。 [サーバー エクスプローラー] タブの [ローカル] を右クリックして [Administration ページを開く] を選択します。 デフォルトポート番号は 86 です。

🔓 COBOL エクスプローラ	💻 サーバー エクスプローラー 🛙	
⊿ 📃 ローカル [localhos	t:86]	
LCWPROV	新規作成(N)	۰.
	Administration ページを開く	Ctrl+F3

2) Enterprise Server Administration 画面へ遷移して、Enterprise Server 一覧が表示されますので、画面の左下 にある [追加] ボタンをクリックします。



3) サーバー名には [JCLDEMO] を入力、動作モードは 64-bit を指定して [次へ] ボタンをクリックします。

助作モード: 〇 32-bit	● 64-bit			
You cannot o	hange your choice of work			
キャンセル	次へ >>			

4) 画面の Page 2/3 ではそのまま [次へ] ボタンを、Page 3/3 では [TN3270 リスナーの作成] チェックボックスをオフに して [追加] ボタンをクリックすると、[JCLDEMO] という名前の 64 ビットアプリケーション稼働用 Enterprise Server が追加されます。"JCL リージョン" と同意語です。

生成オプション:		編集	MFES (MSS)	JCLDEMO	<u>停止</u> 開始…
TN3270リスナーの作成 🗌 using port	\rightarrow	ļ	64 <mark>9</mark>		

5) 左にある [編集] ボタンをクリックします。





- 6) [サーバー] > [プロパティ] > [一般] タブ内の下記項目を設定します。
 - [動的デバッグを許可] チェックボックスをオンにします。この指定により、Eclipse からの動的デバッグが可能になります。

開始オプション:		
共有メモリページ数:	512	サービス実行プロセス: 2
共有メモリクッション:	32	要求ライセンス: 10
ローカルコンソールを表示:		動的デバッグを許可: 🗹
Start on System Start:		64-Bit Working Mode: 🗹
以前のログを削除:		コンソールログサイズ (K): 0

- ② [Apply] ボタンをクリックします。
- 7) [サーバー] > [プロパティ] > [MSS] > [JES] タブで表示される画面の各項目を設定します。入力後は [Apply] ボタ ンをクリックします。

項目名	説明
メインフレーム サブシステム サポート有効	[MSS] タブ配下の設定をオン、オフ指定します。ここではオンに指定します。
ジョブ入力サブシステム 有効	[JES] タブ配下の設定をオン、オフ指定します。ここではオンに指定します。
JES プログラム パス	COBOL アプリケーション実行ファイルが存在するパスを指定します。
システムカタログ	カタログファイルが存在するパスと、そのファイル名称を指定します。
データセットの省略時ロケーション	ジョブ実行時に生成されるスプールデータやカタログされるデータセットのデフォル トパスを指定します。
システムプロシージャライブラリ	プロシージャライブラリの名前を指定します。 ここでは "SYS1.PROCLIB" を入力します。

メインフレーム サブシステム サポート有効: 🔽

CICS (✔) JES IMS PL/I	
一般 Initiators (0) Printers (0)	
ジョブ入力サブシステム有効: 🗹	
JESプログラムパス: C:\work\JCLDEMO\New_Configuration.bin	● 重要
	入力値は全て半角英数字で指定してください。
	これらのフィールドでは改行を入れないように注意してください。
C:\work\JCLDEMO\DATAFILE	
システムプロシージャライブラリ:	
SYS1.PROCLIB	
Fileshare 構成 ロケーション:	
Apply	



8) [サーバー] > [プロパティ] > [MSS] > [JES] > [Initiators] タブを表示し、左下の [追加] ボタンをクリックします。

CICS (🗸)	JES (✔)	IMS	PL/I
一般	nitiators (0)	Printers (0	
追加			

9) 下記画面のように入力して [追加] ボタンをクリックします。この指定により [JCLDEMO] サーバーが開始時にイニシエー タが稼働し、ジョブクラス A,B,C のジョブが実行可能になります。

名前:	
INITABC	
Class:	
ABC	
説明:	
A、B、Cクラスのイニシェータ	

10) 画面左上の [Home] をクリックして一覧画面に戻ります。



3.7 Enterprise Server の開始と確認

- 1) [サーバー エクスプローラー] 内に [JCLDEMO] サーバーが表示されていることを確認します。表示されていない場合は [ローカル] を右クリックし、[更新] を選択してリフレッシュしてください。
- 2) [サーバー エクスプローラー] 内の [JCLDEMO] サーバーを右クリックし、[プロジェクトに関連付ける] > [JCLDEMO] を 選択します。これにより [JCLDEMO] プロジェクトから実行されるアプリケーションは [JCLDEMO] サーバーで処理される ことになります。

プロジェクトに関連付ける
・
JCLDEMO

3) [JCLDEMO] サーバーを右クリックして [開始] を選択します。

⊳	📒 JCLDEMO
⊳	新規作成(N)
⊳	開始

4) 下記ウィンドウが表示された場合は、ここではユーザーによる制限を行わないため [OK] ボタンをクリックします。

M	Enterprise Server サインオン	×
サーバーの接続詳細を	と入力します: JCLDEMO	
ロ サーバーを保護 ユーザー名: パスワード: グループ:	デフルトグループは空白	
☑ 貝恰旧報0/床仔	ОК ‡ †УФЛ	

5) Enterprise Server Administration 画面へ移動して開始状態であることを確認後、[詳細] ボタンをクリックします。



6) [サーバー] > [診断] > [ES コンソール] で [JCLDEMO] サーバーのコンソールログをリアルタイムにチェックすることができ ます。また [Show Entire Log] をクリックしてログ全体を表示させることも可能です。

正常に開始されたことを確認します。

トレース	ダンプ ESコンソール CSコンソール
曲直	Show entries from 1 to 10 of 49 total entries Show last 10 lines
Entry	Event Show Entire Log
40	151105 11521119 CASCD10711 Administration SEP created for Server JCI DEMO process-id = 80908 11-52-11
41	151105 11521144 80908 JCLDEMO CASSI1500I SEP initialization started 11:52:11
42	151105 11521166 78940 JCLDEMO JES000042I SSTM not enabled: CICS 11:52:11
43	151105 11521188 80908 JCLDEMO CASSI1600I SEP initialization completed successfully 11:52:11
44	151105 11521208 78940 JCLDEMO CASSI5001I PLTPI Phase 1 - No PLT Specified 11:52:11
45	151105 11521225 78940 JCLDEMO CASSI5040I Active SEP memory strategy set to x'00000001', retain count 100 11:52:12
46	151105 11521295 81192 JCLDEMO CASBJ0005I Batch initiator started for job classes "ABC" 11:52:12
47	151105 11521318 81684 JCLDEMO CASSI1600I SEP initialization completed successfully 11:52:13
48	151105 11521353 81684 JCLDEMO CASSI5021I PLTPI Phase 2 - No PLT Specified 11:52:13
	151105 (1531117 01000 IOLDENO CAROSSION Commission Provide Instance Of Instants Associated Association (1551)



7) 画面左上の [Home] をクリックして一覧画面に戻ります。



3.8 JCL の実行

現在 [JCLDEMO] サーバーが稼働していますので、サンプルプログラムを実行することができます。まずは簡単な JCL を実行 してみます。

1) [COBOL エクスプローラー] 内にある [JCLDEMO] プロジェクト配下の [copy1.jcl] をダブルクリックし、エディタで内容 を確認します。この JCL は IEBGENER ユーティリティを使用して、JCL 内に書かれたインラインデータを SYSOUT に 書き出しています。

▲ 2 JCLDEMO ▶ COBOL プログラム ▲ 2 JCL ファイル ● COPULjcl ● vsamwrt2.jcl ■ COBOL プログラム ■ //COPY1 10B MSGCLASS=A ● //S1 EXEC PGM=IEBGENER //SYSUT1 DD *	· · · ·	
 ▶ (配 COBOL プログラム ● //COPY1 JOB MSGCLASS=A ● //S1 EXEC PGM=IEBGENER //SYSUT1 DD * 	JCLDEMO	••••••6•••••7•
<pre></pre>	© COBOL プログラム ② JCL ファイル ● copy1.jcl ● vsamwrt2.jcl DATAFILE New_Configuration.bin 楽 KSDSWRT2.gnt @ KSDSWRT2.gnt1.tlog ■ KSDSWRT2.idy	<pre>shikawa,Bunkyo-ku,Tokyo-to 1886 ezaki,Kita-ku,Osaka-shi,Osaka-fu 1900 washiro,Aizu-shi,Fukushima-ken 1911 garu,Tsugaru-gun,Aomori-ken 1911 imotomura,Mimasaka-gun,Okayama-ken 1920 >-cho,Shimizu-shi,Shimane-ken 1886 mayabashi,Kochi-shi,Kochi-ken 1820 > Onsen,Matsuyama-shi,Ehime-ken 1870 p-cho,Nakatsu-shi,Oita-ken 1835</pre>

2) [COBOL エクスプローラー] 内の [copy1.jcl] を右クリックして [Enterprise Server へのサブミット] を選択すると、この JCL が実行されます。



3) [コンソール] タブに JOB 実行ログと JOB 番号が表示されますので、リンクをクリックします。



4) この JOB 番号にかかわるスプール一覧が表示されます。先頭の [JESYSMEG] をクリックしてジョブログを確認します。

JOB01000 Release Update	Name Class Use	e: COI s: A r: JES	PY1 USER	Status Priority COND	: Output Hold : 00		
JCLCM01881	JOB010	00 CO	PY1 JOB STAL	RTED 13:39:50			
JCLCM01821	Status	Class	PY1 JOB END	ED - COND CODE 0000 13 Sten	:39:51 Nbr	Proc Sten	Records
	otatus	oluse	De Marrie	otop	1401.	The otep	Records
Details	Hold	A	JESYSMSG		0		31
Details	Ready	A	SYSPRINT	S1	1		4
Details	Ready	A	SYSUT2	S1	1		10

5) ジョブログの内容を確認すると、この JOB が正常に終了していることが確認できます。

JCLCM0182I JOB ENDED - COND CODE 0000

6) 右クリックで [前へ戻る] を選択し、スプール一覧から [SYSPRINT] をクリックすると、IEBGENER ユーティリティの実行ロ グが記録されていることが確認できます。

Micro Focus MFJGENER Utility Version ED2.3_HF1C Copyright (C) 1997-2015 Micro Focus. All rights reserved. JCLGN0110I(00) - 0000000010 RECORDS COPIED FROM SYSUT1 TO SYSUT2

7) 右クリックで [前へ戻る] を選択し、スプール一覧から [SYSUT2] をクリックすると、出力されたスプールの内容が確認できます。

00001Soseki Natsume	1-1,Koishikawa,Bunkyo-ku,Tokyo-to	1886
00002Ryotaro Shiba	2-3,Sonezaki,Kita-ku,Osaka-shi,Osaka-fu	1900
00003Hideyo Noguchi	5-1, Inawashiro, Aizu-shi, Fukushima-ken	1911
000040samu Dazai	2-6,Tsugaru,Tsugaru-gun,Aomori-ken	1911
00005Eiji Yoshikawa	9-3, Miyamotomura, Mimasaka-gun, Okayama-ken	1920
00006Jirocho Shimizu	6-6, Jiro-cho, Shimizu-shi, Shizuoka-ken	1800
000070gai Mori	3-1,Rintaro-cho,Tsuwano-shi,Shimane-ken	1886
00008Ryoma Sakamoto	1-1,Harimayabashi,Kochi-shi.Kochi-ken	1820
00009Shiki Masaoka	5-5,Dogo Onsen,Matsuyama-shi,Ehime-ken 👘	1870
00010Yukichi Fukuzawa	8-8,Keio-cho,Nakatsu-shi,Oita-ken	1835

3.9 プロシージャライブラリの作成

プロシージャを使用する JCL を実行するために、プロシージャライブラリを作成します。Enterprise Server ではプロシージャを パーティションドデータセットのメンバーとして配置します。このためまずプロシージャライブラリを作成し、プロシージャを配備しておきま す。



1) Enterprise Server Administration 画面へ移動して [JCLDEMO] サーバーの [詳細] ボタンをクリックします。



2) [サーバー] > [コントロール] > [ES モニター & コントロール] ボタンをクリックします。

サーバー	リスナー <mark>(3)</mark>	サ <i>ー</i> ビス (4)	ハンドラ (4)	パ
プロパティ		レ 診断…)過去の統計	
ESモニター&	ביירם-איני]		

3) 画面左の中央部にある [Resources] 直下のコンボボックスから [JES] を選択後、表示された [Catalog] ボタンをク リックします。前項で確認したスプールに関しても [Spool] ボタンをクリックすることにより、全てが参照可能になります。



4) [List] ボタンをクリックして、カタログ情報の一覧を表示すると、現在は何も登録されていないことが確認できます。新規に作成するため [New] ボタンをクリックします。

	Data CATA	LOG	[Refresh		
List	*		Cataloged	Only		
	New	Details	Delete			
DS C	DS Org DS Name					

5) カタログエントリの画面が表示されますので、以下のように入力し [Apply] ボタンをクリックします。

項目名	説明
DS Name	"SYS1.PROCLIB"を入力します。
Physical File	存在するパスを指定。ここでは前項で作成したフォルダパスを入力します。
DS Org	パーティションドデータセットである "PO"を選択します。
RECFM	行順である"LSEQ"を選択します。
Dynamic PDS	プロシージャファイルをフォルダ配下に保持する動的 PDS の場合にオンにします。ここではオンを指定します。
PDS Exts	プロシージャファイル拡張子を指定します。ここでは "prc" を入力します。



	Apply	Сору	Delete	
	DS Name: S	YS1.PROCLIB		✓ Catalog
	Physical File: C	¥work¥JCLDEMO¥D/	ATAFILE¥SYS1.PRO	CLIB¥
	DS Org: P	•• •	RECFM	LSEQ 🗸
> 📠 COBOL プログラム	Codeset: A	SCI 🗸	Created	2015/11/05 14:09:54.15
▲ DCL ファイル copy1.jcl copy1.jcl copy1.jcl		0000	Referenced	2015/11/05 14:09:54.15
 Vsaniwr 2. Jci DATAFILE SYS1.PROCLIB SORTD.prc 	✓ Display Sta	Dynamic PDS	PDS Exts	ASCII V Details

6) カタログ一覧に戻り [List] ボタンをクリックすると作成した PO が表示されます。[SYS1.PROCLIB] をクリックすると配置 されているメンバーが確認できます。

List	*		Cataloged	Only				
	New	Details	Delete					
	DS Org DS Name							
	D <u>SYS</u>	.PROCLIB		DCB				
	Drg DS Name							
List	*			Cataloged Only				
	Nev	v	Details	Delete				
	S Org DS N	ame						
	PO	SYS1 PR	<u>OCLIB</u>					
	PDSM	SYS1.PR	OCLIB(SOR	TD)				
[SYS1	[SYS1.PROCLIB(SORTD)] をクリックすると内容が確認できます。							
//sol	RTD PROC	•						
//sol	RT1 EXEC	PGM=SOF	!T					
77SY8	SOUT=*							
//SOF	RTWKO1 D	D UNIT=S	YSDA,SPA	E=(CYL,(10,10))				
//SOF // PE	RTWKO2 D END	D UNIT=8	YSDA,SPA	E=(CYL,(10,10))				

3.10 COBOL バッチプログラムの実行

COBOL プログラムを含む JOB を実行してみます。

1) [COBOL エクスプローラー] 内にある [JCLDEMO] プロジェクト配下の [vsamwrt2.jcl] をダブルクリックし、エディタで 内容を確認します。

	■サーバー: ローカル @ KSDSWRT2.cbl @ copy1.id ■サーバー ローカル/JCLDEM	0
 ► Sorta Sector ► Sorta Sector	<pre> vsamwrt2.jcl //vsamwrt2.jcl //vsamwrt2.jcl //vsamwrt2 JOB CLASS=A,MSGCLASS=A //ervsamt Exec PGM=IDCAMS //SYSIN DD * DELETE JINJI.KSDS PURGE SET LASTCC=0 DEFINE CLUSTER (NAME(JINJI.KSDS)) - DAT - (RECORDS(10) FREESPACE(20 10) KEYS(5 0) - RECORDSIZE(71 71)) - INDEX (RECORDS(56 50)) /* //SORTSTEP Exec SORTD </pre>	
	<pre>//SORT1.SORTIN DD * 000095hiki Masaoka 5-5,Dogo Onsen,Matsuyama-shi,Ehime-ken 1870 000095nseki Natsume 1-1,Koishikawa,Bunkyo-ku,Tokyo-to 1886 0000270gai Mori 3-1,Rintaro-cho,Tsuwano-shi,Shimane-ken 1886 0000270gai Mori 3-1,Rintaro-cho,Shimizu-shi,Shimane-ken 1886 000051irocho Shimizu 6-6,Jiro-cho,Shimizu-shi,Shizuoka-ken 1800 00005Eiji Yoshikawa 9-3,Miyamotomura,Mimasaka-gun,Okayama-ken 1920 000040samu Dazai 2-6,Tsugaru,Tsugaru-gun,Aomori-ken 1921 000080Syoma Sakamoto 1-1,Harimayabashi,Kochi-shi,Kochi-ken 1820 00009Hideyo Noguchi 5-1,Inawashiro,Aizu-shi,Fukushima-ken 1911 /* //SORT1.SORTOUT DD DSN=&&JIHJIDAT,DISP=(NEW,PASS), // SPACE=(800,(10,10)),DCB=(RECFM=FB,LRECL=71,DSORG=PS),UNIT=SYSDA //SORT1.SYSIN DD * SORT FIELDS=(1,5,CH,A) //MAPL1 EXEC PGM=KSDSWRT2 //YSOUT DD DSN=&&JIHJIAT,DISP=SHR //IND DD DSN=&&JIHJIAT,DISP=SHR //IND DD DSN=&&JIHJIAT,DISP=(NEW,PASS) //PRINTER DD SYSOUT=* //VERIFY1 EXEC PGM=IDCAMS</pre>	

ステップ1:DEFVSAM1

IDCAMS を使用して KSDS クラスター "JINJI.KSDS" を削除し再作成。

ステップ 2 : SORTSTEP

前項で登録した SORTD.prc を使用して、JINJI.KSDS ファイルへの書き込み用データをソート。

③ ステップ 3: APPL1

アプリケーション KSDSWRT2 を呼び出しステップ2でソートされたデータをステップ1で定義した VSAM クラスター に書き込みを行う。同時に書き込まれたデータを DD=PRINTER へ出力。

④ ステップ 4 : VERIFY1

出力内容確認のため、IDCAMS の REPRO で内容を出力。

- 2) [COBOL エクスプローラー] 内の [vsamwrt2.jcl] を右クリックして [Enterprise Server へのサブミット] を選択して、 この JCL を実行します。
- 3) [コンソール] タブに JOB 実行ログと JOB 番号が表示されますので、リンクをクリックします。

📃 コンソール	🗙 🔝 問題	. 🧟 970	🔲 プロパティー	🗙 🚉 🚮 🛃 🖵 🖛
Enterprise Se	erver			
JCLCM0187I	JOB01006	VSAMWRT2	JOB SUBMITTED	(JOBNAME=VSAMWRT2, JOBNUM=01006)
JCLCM0180I	JOB01006	VSAMWRT2	Job ready for	execution. 14:58:44
Processed '	'C:\work\J	CLDEMO\vs	samwrt2.jcl"	



4) この JOB 番号にかかわるスプール一覧が表示されます。

[COND CODE] には [0008] が返却されていますが、これは初回実行時、ステップ 1 で削除該当ファイルが見つからないためですので問題ありません。

JOB01006	Name	: VSA	MWRT2	Status:	Output Hold		
Release	Class	: A		Priority:	00		
Update	Use	I JES	USER	COND:	80000		
JCLCM0188I JCLCM0182I	JOB010 JOB010	06 VS	AMWRT2 JOB AMWRT2 JOB	STARTED 14:58:44 ENDED - COND CODE 0008	14:58:44		
	Status	Class	DD Name	Step	Nbr.	Proc Step	Records
Details	Hold	A	JESYSMSG		0		76
Details	Ready	Α	SYSPRINT	DEFVSAM1	1		18
Details	Ready	Α	SYSOUT	SORTSTEP	2	SORT1	12
Details	Ready	Α	SYSOUT	APPL1	3		1

8) [JESYSMSG] の内容を確認すると、各ステップの COND CODE が確認できます。ジョブが異常終了した場合にはここでエラーの原因を調査することができます。

JCLCM0191I STEP ENDED VERIFY1 - COND CODE 0000 JCLCM0182I JOB ENDED - COND CODE 0008

9) 右クリックで [前へ戻る] を選択し、スプール一覧から DEFVSAM1 ステップの [SYSPRINT] をクリックして内容を確認 します。

Content-Type: text/plain							
Micro Focus MFJAMS Utility Version ED2.3_HF1C Copyright (C) 1997-2015 Micro Focus. All rights reserved.							
DELETE JINJI.KSDS PURGE							
JCLAM0115E(08) - ENTRYNAME NOT CATALOGED [JINJI.KSDS]							
SET LASTCC=0 JCLAM0140I(00) - LASTCC set to 0.							
DEFINE CLUSTER (NAME(JINJI.KSDS)) - DATA - (RECORDS(10) FREESPACE(20 10) KEYS(5 0) - RECORDSIZE(71 71)) - INDEX (RECORDS(50 50)) JCLAM0113I(00) - ENTRYNAME DEFINED [JINJI.KSDS]							

10) 右クリックで [前へ戻る] を選択し、スプール一覧から SORTSTEP の [SYSOUT] をクリックしてソート内容を確認しま す。

Content-Type: text/plain							
	Micro Focus MFJSORT ユーディリテ	ત ૩.(0.00				
	SORT FIELDS=(1,5,CH,A)						
SORT204I: SORT205I:	************************************	****	KKK KKK				
	入力レコード 使用レコード	10 10	件 件				
SORT206I:	OUTPUT ファイル 'SORTOUT' 使用レコード 出力レコード	10 10	件件				
SORT399I:	Micro Focus MFJSORT ユーティリテ	ィ終了	· · ·				



11) 右クリックで [前へ戻る] を選択し、スプール一覧から他ステップに関しても確認してみてください。

12) 前項と同様の手順で、この JOB によってカタログされた情報を確認します。カタログ一覧で [List] ボタンをクリックすると、 VSAM ファイル JINJI.KSDS がカタログされていることが確認できます。

右端の [DCB] をクリックすると登録情報が表示されます。

	Data (CATALOG	[Refresh	Interval (Secs)
List	*		Cataloged	Only	
	New	Details	Delete		
	Drg DS Na	ame			
	SAM O	JINJI.KSDS SYS1.PROCLIB			DCB DCB
0	OS Name:	JINJI.KSDS 🔽 Ca	atalog 🛛 🔶		
Phy	sical File:	C:¥WORK¥JCLDEMC	D¥DATAFILE¥JIN	IJI.KSDS.D	۹T
	DS Org:	VSAM 🗸	RE	ECFM: KS	3 🗸
	Codeset:	ASCII 🗸	Ci	eated: 201	5/11/05 14:58:44.4
	LRECL:	00071	Refer	enced: 201	5/11/05 14:58:44.4
1	BLKSIZE:	00000			
VS	AM Type:	Cluster	Key Sta	rt/Len: 000	000/00005
V	SAM Attr:	Unique Key	Max	/ Avg: 000	071/00005
Shar	eOptions:	Cross Region: • 🗸	Cross Syster	n: 0 🗸	
Dis	play	Start: 1 for	10000 Co	deset: ASC	🗉 🗸 🗌 Details

[Display] ボタンをクリックすると、ファイルの内容が表示されます。

Content-Type: text/plain

00001Soseki Natsume	1-1,Koishikawa,Bunkyo-ku,Tokyo-to	1886
00002Ryotaro Shiba	2-3,Sonezaki,Kita-ku,Osaka-shi,Osaka-fu	1900
00003Hideyo Noguchi	5-1,Inawashiro,Aizu-shi,Fukushima-ken	1911
000040samu Dazai	2-6,Tsugaru,Tsugaru-gun,Aomori-ken	1911
00005Eiji Yoshikawa	9-3,Miyamotomura,Mimasaka-gun,Okayama-ken	1920
00006Jirocho Shimizu	6-6,Jiro-cho,Shimizu-shi,Shizuoka-ken	1800
000070gai Mori	3-1,Rintaro-cho,Tsuwano-shi,Shimane-ken	1886
00008Ryoma Sakamoto	1-1,Harimayabashi,Kochi-shi,Kochi-ken	1820
00009Shiki Masaoka	5-5,Dogo Onsen,Matsuyama-shi,Ehime-ken	1870
00010Yukichi Fukuzawa	8-8,Keio-cho,Nakatsu-shi,Oita-ken	1835

3.11 COBOL バッチプログラムのデバッグ

JCL から実行される COBOL プログラムをデバッグします。

1) [実行] プルダウンメニューの [デバッグの構成] を選択します。

_y.	- Z	ナビゲート	(N) 🔅	検索	プロジェクト(P)	実行(R)
2	実行	テ点をリセッ	٢			
Q,	実行	7 (R)			Ct	rl+F11
检	デバ	ッグ(D)				F11
	実行	テ履歴(T)				•
	実行	ī (S)				•
	実行構成(N)					
	デバ	ッグ履歴(H	I)			+
	デバ	ッグ(G)				•
	デバ	ッグの構成	(B)			



2) 左側のメニューから [COBOL Enterprise Server] を選択して、左上の [新規の起動構成] アイコンをクリックします。



3) [COBOL プロジェクト] へ対象となる「JCLDEMO」を入力し、[Enterprise Server] へ実行させる「JCLDEMO」 サーバーを指定します。

TWE	デバッグ構成	×
構成の作成、管理、および Enterprise Server アプリケーション	実行 への接続とデバッグ	Ť
○ ● ● ● ● >7/1/5入力 第規構成 ^ ○ COBOL JMM アプリケー ○ ○ COBOL JMM アプリケー ○ ○ COBOL JMM アプリケー ○ ○ COBOL アプリケー ○ ○ COBOL アプリケー ○ ○ COBOL のプリセスにア ○ ○ COBOL のプリケーション ○ ○ COBOL のプリケーション ○ ○ CEIpse アカリケーション ○ ○ Generic Server ○ ○ Generic Server(Ext HTTP Preview □ Java アプリケーション ○ □ Java アプリケーション □ □ Junit プラヴィン・テスト m ▲ Draven Build ◆ ◆ DSI フレークフーク ● ● P P	名前(U): 新規構成	
< >> フィルター一致: 27 / 27 項目	適用(<u>Y</u>) 前回保管した	状態に戻す(⊻)
?	デバッグ(<u>D</u>)	閉じる

[デバッグの種類]は「JCL」タブを選択した状態で、[デバッグ]ボタンをクリックします。

4) パースペクティブの切り替え確認ウィンドウでは [いいえ] ボタンをクリックします。

ME	パースペクティブの切り替えの確認				
\bigcirc	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、アプリケーション・デバッグをサポートするために設計されています。 これには、デバッグ・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(<u>Y</u>) いいえ(<u>N</u>)				



5) デバッグタブで [アタッチ待機] 状態になったことを確認します。

☆ デバッグ ☆
 ▲ 新規構成 [COBOL Enterprise Server]
 ② COBOL デバッガ: (アタッチ待機)

- 6) [COBOL エクスプローラー] 内の [vsamwrt2.jcl] を右クリックして [Enterprise Server へのサブミット] を選択して、 JCL を実行します。
- 7) 再度、パースペクティブの切り替え確認ウィンドウが表示されますので、ここでは [はい] ボタンをクリックし、デバッグ用のパー スペクティブを開きます。

MF	パースペクティブ切り替えの確認	×				
\bigcirc	この種類の起動は、中断時に デバッグ パースペクティブが開くように構成されています。					
•	このデバッグ・パースペクティブは、アプリケーション・デバッグをサポートするために設計されています これには、デバッグ・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 ます。	เม				
	このパースペクティブを開きますか?					
□常(:	□ 常にこの設定を使用する(<u>R</u>)					
	(<u>まい(Y)</u> いいえ(<u>N</u>)					

8) 少し待つとデバッグセッションが開始して、プログラムのステップ実行が可能になります。[F5] キーもしくは [実行] プルダウン メニューから [ステップイン] を選択してステップを進めることができ、変数タブでは使用している変数の値が確認できます。





9) 希望のステップの左端をダブルクリックすることにより、ブレークポイントを設定することも可能です。



10) 先に進める場合は画面上部の再開アイコンをクリックします。



11) デバッグを終了させるため、画面上部の終了アイコンをクリックします。



3.12 Enterprise Server の停止

1) [JCLDEMO] サーバーを停止します。



2) [JCLDEMO] サーバーの停止を確認後、Eclipse を終了します。



WHAT'S NEXT

- リモート メインフレーム COBOL 開発: JCL Eclipse 編
- 本チュートリアルで学習した技術の詳細については製品マニュアルをご参照ください。