# Enterprise Developer チュートリアル

メインフレーム COBOL 開発: JCL Visual Studio 2022 編

# 1. 目的

本チュートリアルでは、Visual Studio 2022 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 11 Pro
- 使用マシンに Microsoft Visual Studio 2022 がインストールされていること
- 使用マシンに Micro Focus<sup>™</sup> Enterprise Developer 9.0 for Visual Studio 2022 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. Visual Studio 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. 文字エンコーディングの設定
- 7. Enterprise Server インスタンスの設定
- 8. Enterprise Server インスタンスの開始と確認
- 9. JCL の実行
- 10. プロシージャライブラリの作成
- 11. COBOL バッチプログラムの実行
- 12. COBOL バッチプログラムのデバッグ
- 13. Enterprise Server インスタンスの停止

#### 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。

1) 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。

7— →	ローカル ディスク (C:)
^	名前
	퉬 Tutorials

2) Visual Studio のソリューションを保存する VS という名前のフォルダを C:¥ 直下に作成します。

#### 3.2 Visual Studio の起動

Visual Studio 2022 を起動します。



Visual Studio 2022

## 3.3 メインフレーム サブシステム アプリケーション プロジェクトの作成

1) 新しいソリューションとプロジェクトを作成します。[ファイル] プルダウンメニューから [新規作成] > [プロ ジェクト] を選択して [新しいプロジェクト] ウィンドウを表示し、COBOL が含まれる [メインフレーム サブ システム アプリケーション]を選択後、[次へ] ボタンをクリックします。



2) 下記項目を入力後、[作成] ボタンをクリックします。

項目名	説明
プロジェクト名	任意ですが、ここでは JCLDEMO を入力します。
場所	前項で作成した C:¥VS を指定します。
ソリューションとプロジェクトを 同じディレクトリに配置する	ここではチェックをオンにします。

新しいプロジ	ェクトを構成しま	す			
メインフレーム サブシスラ	テム アプリケーション cosol	Windows Enterpris	e Server		
プロジェクト名仏					
JCLDEMO					
場所し					
ソリューション名(M) 💿					
☑ ソリューションとプロジェクトを同!	じディレクトリに配置する(D)				
				戻る(B)	作成(C)

3) [ソリューション 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。



 プロジェクトを作成したことにより C:¥VS¥JCLDEMO フォルダが作成されていますので、このフォルダ配 下にカタログファイルやスプールファイルを格納する C:¥VS¥JCLDEMO¥DATAFILE フォルダを作成し ます。

また、例題にはプロシージャが含まれておりますので、これを格納するため C:¥VS¥JCLDEMO¥DATAFILE¥SYS1.PROCLIB フォルダも作成します。

ティスク (C:) → VS	▶ JCLDEMO ▶			
/7 <del>**</del>	×	ディスク (C:) → N	/S → JCLDEMO → DAT	AFILE →
名則		A7 min	*	-
鷆 bin		石則		5
퉬 DATAFILE		퉬 SYS1.PRC	OCLIB	2

また、実行時に使用する system フォルダを新規作成してください。

·スク(C:) → VS →	JCLDEMO >
名前	^
📑 obj	
Properties	
system	

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。Visual Studio の [ツール] プルダウンメニューの [オプション] を選択してオプションウィンドウを表示します。

左側ツリービューの [Micro Focus ツール] > [指令] > [COBOL] > [ファイルの指令の確定] 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。

06-#477737 (CUITE)	P	
Live Share		☑ 適用前に指令の確定の変更をフレビューする(P)
▲ Micro Focus ツール		ファイルの指令の確定(D)
Enterprise Server		COBOL 方言
OpenESOL Assistant		ファイルに DIALECT 指令を設定する(A)
TN3270 ディスプレイ		COBOL SQL
vveb リービスノンパノント・ ▷ コードカバレッジ		📴 EXEC SQL を含むファイルに SQL 指令を設定し、NOSQL 指令の設定を解 🔤
▷ コード分析 サードス (1) ローマー (7)		□ EXEC SQLを含まないファイルに NOSQL 指令を設定する(Q)
リービス1 ノツーノエ1 ス  ト フタンドアロン編集		COBOL DLI
データファイル ツール		EXEC DLIを含むファイルに EXECDLI 指令を設定する(L)
プロジェクト		
/□//1/		COBOL CICS
COBOL		EXEC CICS を含むファイルに CICSECM 指令を設定する(C)
PL/I ▶ NuGet パッケージ マネージャー		<ul> <li>EXEC CICS を含まないファイルに NOCICSECM 指令を設定する(N)</li> </ul>
> 06 W		

6) 用意した例題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 [追加] > [既存の項目] を選択し、既存項目の追加ウィンドウにて C:¥Tutorials を指定すると内容が表示されますので、ファイル名先頭に [ACCT] が付いていない下部4ファイルを選択して [追加] 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ーへ例題プログラムが配置されます。

→ × ♠ В х рс х п-	カルディフ	27 (C) > Tutorials >		Tutorial	:の検売	
	110 7.17			♥ O Intoniai:	9719C 370	
<ul> <li>新しいフォルダー</li> </ul>						
MFBSI	^	名前	更新日時	種類	サイズ	
PerfLogs			2000/07/211010	CRL 77 ()	E KB	
Program Files			2004/02/26 23:33 2004/02/26 23:33	CBL ファイル	5 KB 12 KB	
Program Files (x86)		ACCTREC.cpv	2004/02/26 23:33	CPY ファイル	2 KB	
ProgramData		acctset.bms	2006/01/24 19:34	BMS ソースコード	46 KB	
synroot	- 64	ACCTSET.cpy	2006/02/16 22:21	CPY ファイル	38 KB	
temp		🗟 copy1.jcl	2009/06/25 17:02	JCL ソースコード	1 KB	
Trataviale	- 1	SDSWRT2.cbl	2009/06/26 15:02	CBL ファイル	3 KB	
Tutonais		SORTD.prc	2012/05/23 16:49	PRC ファイル	1 KB	
DATAFILE	~	🖏 vsamwrt2.jcl	2009/06/26 15:01	JCL ソースコード	2 KB	
ファイルタ(N)		ut2 is!" "coput is!" "KCDSWPT2 sh!" "S	SOPTD are"	Entern	ise 774 II. (* chl: *	ccm *
ファイル・日(四)・	vsditiv	падет соруплет карачикталерг з	SOKTOPIC	<ul> <li>Enterprise</li> </ul>	ise yr fyr ( iebi, i	cep,

7) 種類別に表示するため、[ソリューション エクスプローラー] 内の [仮想ビュー] アイコンをクリックします。



8) [ソリューション エクスプローラー] 内に表示されている JCLDEMO プロジェクト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ます。



9) [ソリューション エクスプローラー] をフォルダビューに切り替えます。右上の [ビューを切り替える] アイコン から [フォルダビュー] を選択すると、前項で作成した SYS1.PROCLIB フォルダが表示されますので、こ こへプロシージャである SORTD.prc ファイルをドラッグアンドドロップします。移動後はソリューションビュ ーに戻します。



# 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) [ソリューション エクスプローラー] 内の [Properties] をダブルクリックしてプロパティウィンドウを表示 します。 2) 左側ツリービューの [アプリケーション] を選択して、生成する実行ファイルを GNT にするため [出力の種類] へ [INT/GNT] を選択します。

アプリケーション	構成(C): なし	▲ 情報
SQL		
デバッグ	山市の友益。	GNT は Micro Focus™
従属パス	山川の石前:	独白の動的ロード実行可能ファイルを指します
BMS	したのであ	
IMS	田 月の種類:	
	INT/GNT	

3) 左側ツリービューの [COBOL] を選択して、下記項目を入力します。

項目名	説明
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 [x64] を指定します。
COBOL 方言	COBOL 言語方言を指定します。 例題プログラムは 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。
文字セット	EBCDIC または ASCII を指定します。ここでは [ASCII] を選択します。
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 に指定するためにチェックをオンにします。
出力パス	実行ファイルが出力されるパスを指します。任意に指定可能です。
追加指令	ここでは何も指定しません。

アプリケーション	構成( <u>C</u> ): アクティブな (Debug)	✓ プラットフォーム( <u>M</u> ):	アクティブな (x64)	$\sim$	
SQL ≓ufautt					
ブハック	主权				
	プラットフォームターゲット:	x64 ~			
IMS	COBOL 方言:	Enterprise COBOL for z/C	s ~	文字セット:	ASCII ~
プリプロセッサ	1-7 7+-7%h	周定			
COBOL	<i>y=x yx=x y</i> ::	回 <b>た</b>			
COBOL リンク	デバッグ用にコンパイル			.GNT (23)	ンパイル
アセンプラ	120112 2 4054			□ 複数プロセ	スによるコンパイル
Micro Focus Code Analysis	באר ז איז די				
	メインフレーム				
		オプション		オプション	
	エラーおよび警告				
	警告レベル:	情報メッヤージを含めるしべ	ll n	~	
		INTROPIC DE LONGIO O	v 1)		
	最大エラー数:	100	□ 警告をエラーとして	処理	
	# <b>+</b>				
	щЛ				
	出力パス:	.¥bin¥x64¥Debug¥			参照
	□ 指令ファイルの生成		□ リストファイルを生)	戓	
	□ コード カバレッジを有効に	する	□ プロファイラを有効	にする	
	这句话会				
	20/11月170				

4) 左側ツリービューの [デバッグ] を選択して、[プロファイル] には JCLDEMO、[起動] には JCL を指定します。

アプリケーション SQL	構成( <u>C</u> ): なし
デバッグ	プロファイル: JCLDEMO
従属パス	起動: JCL

5) プロパティファイルを上書き保存します。

・ 🖆 🖪 🛛 🏷 ・ 🤆 ・	Debug 🝷	x86
; → 、 選択されたファイルを	上書き保存	(Ctrl+S)

#### 3.5 ビルドの実行

- 1) [ソリューション エクスプローラー] の JCLDEMO ソリューションを右クリックして [ソリューションのビル ド] を選択すると、コンパイル指定に沿ったビルドが実行されます。
- 2) [出力] ウィンドウで成功を確認します。



3) 前項で確認した出力パスへ実行ファイルに指定した gnt ファイルが作成されていることを確認します。



#### 3.6 文字エンコーディングの設定

Enterprise Server インスタンスを運用、管理する Enterprise Server Common Web Administration (以降 ESCWA)では、スプールやデータ内容などに含まれる日本語を正しく表示させるために、事前に文字セット を所定のフォルダへ展開します。製品マニュアルの「リファレンス > コードセットの変換 > CCSID 変換テーブ ルのインストール > CCSID 変換テーブルをインストールするには」を参照しながら進めてください。

1) CCSID 変換テーブルをインストールします。

製品マニュアルにリンクされている下記の IBM CCSID 変換テーブルを、Web ブラウザから任意のフォル ダヘダウンロードします。アドレスは変更される可能性がありますので、製品マニュアルにてご確認ください。 <u>http://www.microfocus.com/docs/links.asp?vc=cdctables</u>

- 製品インストールフォルダ配下の etc フォルダに CCSID フォルダがない場合はこれを作成します。
   例)C:¥Program Files (x86)¥Micro Focus¥Enterprise Developer¥etc¥CCSID
- 3) ダウンロードファイルに含まれている Package2.zip を展開します。

- 4) 展開した Package2 フォルダに含まれる IBM-932.zip を展開します。
- 5) 展開した IBM-932 フォルダを切り取り、作成した CCSID フォルダ配下へ貼り付け、14 ファイルが含ま れていることを確認します。

	<del>↓</del>   IBM-932				_	
ファイル	ホーム 共有 表示					~ <b>(</b> )
$\leftarrow \rightarrow$	<ul> <li></li></ul>	→ Micro Focus → Enterpri	se Developer >	etc > CCSID > IBM-932	v ♂ IBM-	932の検索 🔎
^	名前	2種類 へ	サイズ	更新日時		^
	03A434B0.MU-R-A2	MU-R-A2 ファイル	28 KB	1999/10/14 10:04		
	03A434B0.MU-R-A3	MU-R-A3 ファイル	28 KB	2002/01/15 11:58		
	03A434B0.MU-R-D	MU-R-D ファイル	28 KB	1999/10/14 10:04		
	03A434B0.PACKAGE	PACKAGE ファイル	5 KB	2002/01/16 10:37		
	34B003A4.RPMAP12A	RPMAP12A ファイル	336 KB	1999/10/14 10:03		
	34B003A4.RPMAP120	RPMAP120 ファイル	336 KB	1999/10/14 10:03		
	03A434B0.TPMAP11A	TPMAP11A ファイル	329 KB	1999/10/14 10:02		
	03A434B0.TPMAP12A	TPMAP12A ファイル	329 KB	2002/01/15 11:58		
, i	03A434B0.TPMAP110	TPMAP110 ファイル	329 KB	1999/10/14 10:02		
4	34B003A4.UM-E-A21	UM-E-A21 ファイル	54 KB	1999/10/14 10:04		
2	34B003A4.UM-E-D12	UM-E-D12 ファイル	54 KB	1999/10/14 10:04		
7	03A434B0.UPMAP12A	UPMAP12A ファイル	446 KB	1999/10/14 10:01		
	03A434B0.UPMAP13A	UPMAP13A ファイル	490 KB	2002/01/15 13:42		
<b>~</b>	3434B0.UPMAP120	UPMAP120 ファイル	447 KB	1999/10/14 10:01		~
14 個の項	<b>頁目</b>					

詳細については、製品マニュアルの「ディプロイ > 構成および管理 > Enterprise Server の構成および 管理 > Enterprise Server Common Web Administration > [Native] > [Directory Servers] > リージョンとサーバー > リージョン > エンタープライズ サーバー リージョンの文字エンコーディングの サポート」をご参照ください。

6) Windows サービスとして起動している Micro Focus Enterprise Server Common Web Administration を再起動し、インストールした CCSID をロードさせます。

Q #-EZ						
ファイル(E) 操作(A)	表示(1) ヘルブ(日)					
🗢 🔿 📅 🖾 🖉	à 📴 📓 📷 🕨 🗰 H 🕪 👘					
サービス (ローカル)	④ サービス (ローカル)					
	Micro Focus Enterprise Server	- 名前 ^ ^	説明	状態	79-	h7: ^
	Common Web Administration	Canal Micro Focus CES daemon	Micro Focus CES daemon	実行中	自動	
	#	Micro Focus Directory Server	Micro Focus Directory Server	実行中	自動	
	サービスの再起動	Micro Focus EA Integration Service Host	Micro Focus EA Integration Service Host	実行中	自動	
		Micro Focus Enterprise Analyzer Web Service	Micro Focus Enterprise Analyzer Web Service	実行中	自動	
	en 00	🐘 Micro Focus Enterprise Server Common Web Administration	Micro Focus Enterprise Server Common Web	実行中	自動	
	Micro Focus Enterprise Server	Micro Focus Event Monitor Service Shutdown Coordinator	Micro Focus Event Monitor Service Shutdow	実行中	自動	
	Common Web Administration	Micro Focus SEE Administration Server	Micro Focus SEE Administration Server		手動	
		Children Andrew See Listener	Micro Focus SEE Listener		手動	
		Micro Encur SEE Monitor	Micen Encur SEE Monitor	常行曲	0.55	. ×
	 	<b>x</b>				,

#### 3.7 Enterprise Server インスタンスの設定

Enterprise Server インスタンスには JES をエミュレーションする機能が搭載されており、この開発用インスタ ンスを使用してメインフレームアプリケーションのテスト実行やデバッグを行います。本番環境には実行環境製品で ある Enterprise Server をインストールし、本番用インスタンス上でアプリケーションを稼働させます。

1) Enterprise Server インスタンスを作成します。[サーバー エクスプローラー] タブの [Micro Focus Server] を右クリックして [管理] を選択します。

Windows の Micro Focus Directory Server サービスが開始されているか確認し、停止している場合は開始してください。



 ブラウザが立ち上がり、Enterprise Server インスタンスを運用、管理する ESCWA が表示されます。 ESCWA では Micro Focus Directory Server サービスのポートへ接続して登録された Enterprise Server インスタンスを管理するため、まず接続を作成します。既に存在している場合には、インスタンス作成 に進んでください。

左側ペインで [Directory Server] をクリックし、右側ペインの [追加] ボタンをクリックします。

ES	管理	ダッシュボード	
✓ 回グ, > 論	ループ 理	DIRECTORY SERVER   C + 追加	
> PA	ectory Server C	名前〜 説明〜 ホスト〜 ポー	
			Directory Server
追加さ	フィンドウでは	[名前] は任意ですが、ここでは Default を、	名前 *
「ホス	h] It localh	nost またはマシンの IP アドレスを	Default
			ホスト*
Lボー	ト」はMicro	Focus Directory Server の	localhost
デフォ	ルトポートであ	5る 86 を入力して [保存] ボタンをクリックします。	86
			* 入力必須の項目です
			保存

3) 作成した [Default] 接続をクリックし、右側ペインの [新規作成] ボタンをクリックします。

> FAU			
🗸 🗐 Directory Server	<b>リージョンおよびサーバー リスト</b>   C	* 新規作成	直 すべて削除
✓ 🖽 🔂 Default			

4) [リージョンの新規作成] 項目の [名前]、[説明] は任意ですが、ここでは名前に JCLDEMO、説明に JCL チュートリアル用と入力します。実行可能ファイルは 64 ビットを指定してコンパイルしたため、稼働させる Enterprise Server インスタンスも同様に [64 ビット作業モード] ヘチェックを入れます。これにより警告 が表示されますが無視して先に進んでください。[MSS 有効] にチェックが入っていることを確認し、 [TN3270 リスナーの作成] のチェックを外して [保存] ボタンをクリックします。

リージョンの新規作成	
名前。 JCLDEMO	
成明 JCL チュートリアル用	€ 重要
▲ Directory Serverとリージョンの作業モー ドが一致しません。プラットフォームに よっては、記絶時に不具合が発生する可 能性があります。	実行ファイル生成に指定した稼働ビット数 = Enterprise Server インスタンス稼働ビット数 である必要があります。
2 64ビット作業モード ○ MSS有効 □ TN3270リスナーの作成 TN3270リスナーボート 0 ↓	
・入力必須の項目です	
保存 戻る	

5) 64 ビットアプリケーション稼働用の JCLDEMO インスタンスが作成され、一覧に表示されます。 JCLDEMO インスタンスにカーソルを合わせ、[編集] アイコンをクリックします。

	名前	タイブ	ステータス	64ビット	MSS有効	セキュリティ	
Ξ.	JCLDEMO	Region	Stopped	$\checkmark$	$\checkmark$	デフォルト	🧷 🔅 🗓
							編集

6) JCLDEMO インスタンスのログなどが出力される [システムディレクトリ] には前項で作成した system フ ォルダを指定して、[リージョンの機能] の [JES 有効] をチェックします。

開始オプション		リージョンの機	能		
名前∗♀	システムディレクトリ 🛿	✓ MSS有効(	♀ JES有効 ♀	□ IMS有効 🖓	□ PL/I有効 🖓
JCLDEMO	C:\VS\JCLDEMO\system	☐ MQ有効 ♀	1		

7) 表示画面の下にある [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、Visual Studio からの動的デバッグが可能になります。

□ □-カル ♀	✓ 動的デバ ♀	<ul> <li>システム</li></ul>
コンソー	ッグを許	起動時に
ルを表示	可	開始する
■ 64ビット 8 作業モー ド	□ 以前の□	

8) [追加設定] の [構成情報] 欄に、文字エンコーディングを指定する MFACCCGI\_CHARSET 環境変数に IBM-932 を認識させるための値である Shift\_JIS と、 プロジェクトのパスを指定する環境変数を設定し、 最後に [適用] ボタンをクリックします。

入力値) [ES-Environment] proj=C:¥VS¥JCLDEMO MFACCCGI\_CHARSET=Shift\_JIS

追加設定

構成情報 🖓

[ES-Environment]
proj=C:\VS\JCLDEMO
MFACCCGI\_CHARSET=Shift\_JIS

9) 画面上部の [JES] プルダウンメニューから [構成] を選択し、表示される画面の各項目を設定します。構成 情報に指定した proj 環境変数を使用して値を入力後、 [適用] ボタンをクリックします。

項目名	説明
JES プログラム パス	COBOL アプリケーションの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	カタログファイルを出力するパスと、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。

Sの構成   C 適用	Q
JESプログラムバス Q \$proj\bin\x64\Debug	システム カタログ ② SprojDATAFILE\catalog.dat
	データセットの省端時ロケーション 🔇 \$proj\DATAFILE
システムプロシージャ ライブラリ 〇	Fileshare 構成ロケーション ♀
SYS1.PROCLIB	4

10) [イニシエータ] の [新規作成] ボタンをクリックします。

イニシエータ  〇	* 新規作成
-----------	--------

11) 下記画面のように入力して [保存] ボタンをクリックします。この指定により JCLDEMO インスタンス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。

JESイニシエータ	
名前 • Q INITABC	
クラス Q ABC	
説明♀ A,B,C クラスのイニシエーター	/,
・入力必須の項目です	
保存	戻る

12)セキュリティ観点から、Web リスナーのデフォルトステータスは [Disabled] になっています。安全を確認 したうえで、 [一般] プルダウンメニューから [リスナー] を選択し、表示された Web リスナーのステータス を [Stopped] へ変更後、 [適用] ボタンをクリックします。

ステータスの設定 🖓		実際のアドレス
Stopped	~	tcp:0.0.0.0:0
	ステータスの設定 V Stopped	ステーダスの設定 V Stopped ・

- 13) 画面左側ペインの [Default] をクリックして一覧画面に戻ります。
  - ✓ I Directory Server
    ✓ I Content of the server

#### 😢 重要

 パージョン 7.0 では、パフォーマンス向上の観点から JES 関連ファイルである SPLJOB.DAT のフォ ーマットが改善されています。そのため、旧バージョンのファイルを 7.0 以降で利用する場合は mfsplcnv コマンドを使用して新フォーマットにコンバートする必要があります。コンバートを実行すると、 古いフォーマットのファイルは SPLJOB.bak として保存されます。
 対象ファイルの特定には MFSYSCAT 環境変数を利用して、カタログファイルを指定します。
 例)
 set MFSYSCAT=C:¥VS¥JCLDEMO¥DATAFILE¥catalog.dat
 mfsplcnv -2
 詳しくは製品マニュアルをご参照ください。

#### 3.8 Enterprise Server の開始と確認

- Visual Studio の [サーバーエクスプローラー] 内に JCLDEMO インスタンス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 [Micro Focus Server] を右クリックし、[最新の情報に更新] 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 インスタンスを右クリックし、[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 インスタンスで処理されることになります。

JCLDEMO	プロジェクトと関連付ける	JCLDEMO

3) [JCLDEMO] インスタンスを右クリックして [開始] を選択します。

Þ	•	JCLDEMO
l	開始	i(S)

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。

Enterprise Server サインオン			×
サーバーの接続詳細を入力: JCLD	emo		
🗌 サーバーの認証情報を要求			
ユーザー名:			
パスワード:			
グループ:			
☑ 認証の保存			
		ОК	キャンセル

5) ESCWA へ移動して開始状態であることを確認後、[編集] アイコンをクリックします。

	名前	タイプ	ステータス	<u>64</u> ビット	MSS有効	セキュリティ	
Ξ.	JCLDEMO	Region	Started	$\checkmark$	$\checkmark$	デフォルト	🖉 🕸 🗓
							編集

6) 画面上部の [モニター] プルダウンメニューから [ログ] > [コンソールログ] を選択し、正常に開始されたこ とを確認します。

ログレベルが I はインフォメーション、S や E の場合はエラー表示されます。

ッセージロ〜	メッセージ	◇ プロセスι▷ ◇	ログレベルン		【JES 機能の	D正常開始ログ抜粋】
91 J	ロセスル	メッセージル	ログレベル	メッセージ	JES0000511	Job Entry Subsystem (JES) services initialized
2022/ 9	240	CASCS5001I	1	Communications interfa	JES0000591	JES 5 digit job numbering support enabled
2022/ 9	872	CASTS1002I	1	ES TRC Service Proces	CASCS5003I	Communications interface 01 initialization complete
2022/ 9	240	CASCS5003I	1	Communications inte	CASCD1060	IES Initiates constant for Service ICI DEMO, accesses id a 42
2022/ 7	920	CASTS0002I	1	ES TSC Service Pro	CASCD10001	JES Initiator created for Server JCLDEMO, process-Id = 0.
2022/ 8	000	JES000051I	1	Job Entry Subsyr	CASBJ0023I	Batch initiator INITABC: class(es) "ABC"
2022/ 8	000	JES0000591	1	JES 5 digit job		
2022/ 9	240	CASCS5100I	1	Communicat us Process i	instance 01 is ready to ac	ccept requests
2022/		CASCD1060I	1	JES Initiator created for Se	erver JCLDEMO, process-	-id = 9332
2022/ 8	000	CASBJ0023I	1	Batch initiator INITABC: cl	ass(es) "ABC"	
2022/		CASCD0127I	1	SEP 00001 created for ES	JCLDEMO, process-id =	9040
2022/		CASCD0127I	1	SEP 00002 created for ES	JCLDEMO, process-id =	7180

\rm \rm 注意

いくつかのサービス開始が失敗してもインスタンスは開始されますので、ログ内容を必ず確認してください。

#### 3.9 JCL の実行

現在 JCLDEMO インスタンスが稼働していますので、例題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) [ソリューション 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。

copy1.jcl 🕫 🗙	<b>-</b> ¢	ソリューション エクスプローラー 🔹 👎 🗙
V/COPY1_JOB_MSGCLASS=A //S1_EXEC_POM=EBGENER //SUBJECTIONED	÷	ⓒ ☺  │"⊙ ▪ ≒ ᄏ ☞│ ⊁ ≓│ ▼
//SYSPRINI DD SYSOUT=* //SYSUT1 DD *	_	ソリューション エクスプローラー の検索 (Ctrl+:) - 🔑 🖊 🗸
00001šoseki Natsume 1-1,Koishikawa,Bunkyo-ku,Tokyo-to 1886 00002Ryotaro Shiba 2-3,Sonezaki,Kita-ku,Osaka-shi,Osaka-fu 1900 00009Hideyo Noguchi 5-1,Inawashiro,Aizu-shi,Fukushima-ken 1911 000040samu Dazai 2-6,Isugaru,Isugaru-gun,Aomori-ken 1911 00005Liji 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,Shimizu-shi,Shimane-ken 1820 00008Shiki Masaoka 5-5,Dogo Onsen,Matsuyama-shi,Ehima-ken 1870 000019Vukichi Fukuzawa 8-8,Keio-cho,Nakatsu-shi,Oita-ken 1835 /* /XSYSUT2 00 SYSOUT=*	4.0 4.1 4.2 4.3 4.4 4.4 4.4 4.4 4.4 4.4 4.4 4.4 4.4	

2) [ソリューション エクスプローラー] 内の copy1.jcl を右クリックして [サブミット JCL] を選択すると、この JCL が実行されます。



3) [JCLDEMO スプール] タブが表示されますので、[メッセージ] で COND CODE を確認します。



4) ESCWA で 詳細を確認してみます。JCLDEMO インスタンスを選択後、[JES] プルダウンメニューから [スプール] を選択します。



5) スプール一覧が表示されます。フィルタ機能で [完了] が指定されていることを確認します。

JES 出力キュー  C	アフィルタ	フィルタ		>
		名前*♡		
		ユーザー・0 ・		-
		ジョブ番号 * 🖗		_
		出力タイプ ○ 入力 ♡ ○ 入力の保留 ♡ ○ ディスパッチ ♡ アクティブ ♡ 完了 ♡	キュー ✓ 出力 ♀ ✓ 出力の保留 ♀ ✓ 印刷中 ♀	2

6) 実行した JOB 番号のスプールをダブルクリックして内容を表示します。

名前	ジョブID	クラス	ユーザー	条件コード			
COPY1	J0001000	А	mfuser	0000			
状態	クラス	DD名	ステップ	ステッ	ブ番号	PROCステップ	レコード数
Hold	А	JESYSMSG	i	0			32
Ready	А	SYSPRINT	S1	1			4
Ready	А	SYSUT2	S1	1			10

7) 先頭の [JESYSMEG] をダブルクリックしてジョブログを確認すると、正常に終了していることが確認できます。

---> 14:09:13 JCLCM0191I STEP ENDED S1 - COND CODE 0000

---> 14:09:13 JCLCM0182I JOB ENDED - COND CODE 0000

8) 右上にある [戻る] ボタンをクリックしてスプール一覧に戻り、[SYSPRINT] をダブルクリックすると、 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 ED8.0\_GA\_1 Copyright (C) Micro Focus 1997-2020. All rights reserved.

JCLGN0110I(00) - 0000000010 RECORDS COPIED FROM SYSUT1 TO SYSUT2

有上にある [戻る] ボタンをクリックしてスプール一覧に戻り、[SYSUT2] をダブルクリックすると、出力されたスプールの内容が確認できます。

0001Soseki Natsume	1-1,Koishikawa,Bunkyo-ku,Tokyo-to	1886
00002Ryotaro Shiba	2-3,Sonezaki,Kita-ku,Osaka-shi,Osaka-fu	1900
0003Hideyo Noguchi	5-1,Inawashiro,Aizu-shi,Fukushima-ken	1911
0004Osamu Dazai	2-6,Tsugaru,Tsugaru-gun,Aomori-ken	1911
0005Eiji Yoshikawa	9-3,Miyamotomura,Mimasaka-gun,Okayama-ken	1920
0006Jirocho 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

#### 3.10 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 インスタンスではプロシージャを区分データセットのメンバーとして配置します。このため、前項で SYS1.PROCLIB というフォルダを作成してプロシージャを配置しました。 、 Datafile 、 SYS1.PROCLIB

SORTD.prc

 ESCWA へ移動して JCLDEMO インスタンスの画面上部にある [JES] プルダウンメニューの [カタログ] を選択し、[リスト] ボタンをクリックすると、現時点では何も登録されていないことがわかります。新しくカタ ログするために [新規作成] ボタンをクリックします

メインフレーム COBOL 開発:JCL Visual Studio 2022 編

カタログ   <mark>ヘリスト</mark> ・ 最終更新: 2022/01/24 17:27:31	□ カタログ式のみ □ 暗黙的なワイルドカードの無効化	前 別除	* 新規作成
DS編成 ✓ □ DS編成 DS名			
合計:0 選択済み:0			

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

項目名	説明
DS 名	SYS1.PROCLIB を入力します。
物理ファイル	物理パスを指定します。ここでは前項で作成したフォルダパスを入力します。
DS 編成	区分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。
動的 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 拡張	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。

カタログ エントリを追加 📋 🏾 🛛 🦛

os‰•Q		🗾 カタロク式 💡
SYS1.PROCLIB		
勿理ファイル・♀		
<pre>\$proj\DATAFILE\SYS1.PROCLIB</pre>	1	
DS編成 ♀		RECFM 😨
PO	<b>*</b>	LSEQ
コードセット 🖗		作成日時 🛛
ASCII	<b>~</b>	無期限
		最終参照日時 🛛
LRECL 8 0	<u>्र</u> ./.न.म	MGMICLASS V
0		
BLKSIZE 🗸 o	<u>्र</u> ी राजन	
☑ 動的PDS		PDS拡張
		pro

3) カタログ一覧に戻り [リスト] 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] の [表示] アイコン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。

	ð	PO	SYS1.PROCLIB	◎ 必 前
				表示
	ð	PO	SYS1.PROCLIB	💁 🖉 🛅
	6)	PDSM	SYS1.PROCLIB(SORTD)	◎ 29 位
				表示
[SY	S1.	PROC	LIB(SORTD)]の[表示]	アイコンをクリックすると内容が確認できます。
201	0			
			SYS1.PROCLIB(SORTD) 表示 《	< 🗡 rt-5%
			//SORTD PROC	
			//SORT1 EXEC PGM=SORT //SYSOUT=*	
			//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(10,10	))
			//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(10,10	
			// PEND	

# 3.11 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) [ソリューションエクスプローラー] 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。

V/VSAMWRT2 JOB CLASS=A,MSGCLASS=A		÷	0 0 A 🚚 🐚 • 与 🖪	
//####################################				
//SYSPRINT DD SYSOUT=*	G 16	ソリューション エクスフローラー の	)検索 (Ctrl+:)	
//SYSIN DD * DELETE JINJI KSDS PURGE		20 D	🗖 אין	)' (1/1 のプロジェクト)
SET LASTOC=0		NUME -	▲ CBL JCLDEMO	
DEFINE CLUSTER (NAME(JINJI.KSDS)) -		ALCONO.	🔑 Properties	
(RECORDS(10) FREESPACE(20 10) KEYS(5 0) -			A E COBOL Source	
RECORDSIZE(71 71) ) -		NGA Taklarlar 2 Nata Taklarlar 2 Nati Sectorian 2		pl
INDEX (RECORDS(50 50))		Roda casteriaria Roda chalactaria di Roda chalactaria	KSDSWRT2.cbl	
//SORTSTEP EXEC SORTD		Main Taplophi I Main Taplophi I Main Taplophi I		
//SORTI.SORTIN DD ★ DDDD9Shiki Masanka 5-5.Dovo Onsen Matsuvama-shi.Ehime-kan I	1870	AND MALES	copy i.jci	
00001Soseki Natsume 1-1,Koishikawa,Bunkyo-ku,Tokyo-to	1886	440 Million 4	SORID.prc	
000070gai Mori 3-1,Rintaro-cho,Tsuwano-shi,Shimane-ken	1886		S vsamwrt2.jcl	
00002Ky0taro sniba 2-5,50nezaki,krta-ku,osaka-sni,osaka-tu 00006Jirocho Shimizu 6-6,Jiro-cho,Shimizu-shi,Shizuoka-ken	1800			
00005Eiji Yoshikawa 9-3,Miyamotomura,Mimasaka-gun,Okayama-ken	1920			
000040samu Dazai 2-6,Isugaru,Isugaru-gun,Aomori-ken 00008Ryoma Sakamoto 1-1.Harimayabashi.Kochi-shi.Kochi-ken	1911 1820	inn.		
00010Yukichi Fukuzawa 8-8,Keio-cho,Nakatsu-shi,Oita-ken	1835			
00003Hideyo Noguchi 5-1,Inawashiro,Aizu-shi,Fukushima-ken /*	1911			
/*SORT1.SORTOUT DD DSN=&&JINJIDAT,DISP=(NEW,PASS), // SPACE-(RDO (10 10)) DCR-(RECEM-ER LEFC)-71 DSORC-PS) UNIT-SYSDA			ソリューション エクスプローラー	Git 変更 クラス ビュー
//SORTI.SYSIN DD * SORTI.SYSIN DD *			プロパティ	-
			vsamwrt2.jcl JCL 項目プロ/	パティ
//WHILE LOC FOR ADDONNES			🔡 🛃 🌽	
//NDD_DD_DSN=&&JINJIDAT,DISP=(OLD,DELETE)			□ その他	
//PRINTER DD SYSOUT=*			JCL ファイルの種類	JES2
//************************************			ファイル名	vsamwrt2.jcl
//SYSPRINT DD SYSOUT=*			完全パス	C:¥VS¥JCLDEMO¥v
REPRO INDATASET(JINJI.KSDS) -			□ 高度	
OUTFILE (SYSPRINT)			カスタムッール	
/*				

ステップ 1:DEFVSAM1

IDCAMS を使用して VSAM データセット JINJI.KSDS を削除し、クラスターを持つ KSDS とし て再作成します。

② 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します。

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います。同時に書き込まれたデータを DD=PRINTER へ出力しま す。

④ ステップ 4:VERIFY1

出力内容確認のため、IDCAMS の REPRO で内容を出力します。

- 2) [ソリューション エクスプローラー] 内の vsamwrt2.jcl を右クリックして [サブミット JCL] を選択して、 この JCL を実行します。
- 3) ESCWA で JCLDEMO インスタンスを選択後、[JES] プルダウンメニューから [スプール] を選択しま す。



4) 実行した JOB 番号のスプールをダブルクリックして内容を表示します。

D	□名∨	ステップ~	PROCステップ√	状態∨	クラス〜		
	状態	クラス	DD名	ステップ	ステップ都	ŧ procステ	・ レコード数
	Hold	А	JESYSMSG		0		76
	Ready	А	SYSPRINT	DEFVSAM1	1		18
Ľ	Ready	А	SYSOUT	SORTSTEP	2	SORT1	12
Ľ	Ready	А	SYSOUT	APPL1	3		1
Ľ	Ready	А	PRINTER	APPL1	3		10
Ľ	Ready	А	SYSPRINT	VERIFY1	4		41

[COND CODE] に [0008] が返却された場合は、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。

5) [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。

---> 13:23:50 JCLCM0191I STEP ENDED STEP04 - COND CODE 0000

---> 13:23:50 JCLCM0182I JOB ENDED - COND CODE 0000

6) 右上にある [戻る] ボタンをクリックしてスプール一覧に戻り、DEFVSAM1 ステップの [SYSPRINT] を ダブルクリックして内容を確認します。

JCLAM0114I(00) - ENTRYNAME DELETED [JINJI.KSDS]

SET LASTCC=0 JCLAM01401(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)) JCLAM01131(00) - ENTRYNAME DEFINED [JINJI.KSDS]

7) 右上にある [戻る] ボタンをクリックしてスプール一覧に戻り、SORTSTEP の [SYSOUT] をクリックして ソート内容を確認します。

SORT2051:	INPUT ファイル 'SORTIN'	
	入力レコード	10件
	使用レコード	10件
SORT2061:	OUTPUT ファイル 'SORTOUT'	
	使用レコード	10件
	出力レコード	10件
SORT3991:	Micro Focus MFJSORT ユーティリティ	修了

- 8) 右上にある [戻る] ボタンをクリックしてスプール一覧に戻り、他ステップに関しても確認してみてください。
- 9) 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 [リスト] ボタン をクリックすると、VSAM ファイル JINJI.KSDS が新たにカタログされていることが確認できます。

このカタログ情報にカーソルを合わせ、[DCE	3] をクリックすると登録情報が表示されますので、内容を確認し
てみてください。	

	ð	VSAM	JINJI.KSDS				0 <i>P</i>	۵.			
	ð	PO	SYS1.PROCLIB				_/г	DCB			
								DCD			
ונאונ	SDS	C MA	コピー 道 削除				表示	戻る			
DS: JIN 物3 C:¥	名 IJI.KSDS 里ファイノ ¥VS¥¥JC	ル・2 CLDEMO¥¥DATAFILE	¥JINJI.KSDS.DAT			☑ カタログ式 🤇					
DS	////////////////////////////////////		~			RECFM Q			-		
_						····			-		
また	、—『	覧の [表示	:] アイコンもしくは	t DCB 情	報の[剥	 表示】ボタ	ンをクリ	ノックする	ると、ファ	ァイルのダ	容が表示
また され	、一覧 ます	覧の [表示 。	] アイコンもしくは	t DCB 情	報の[ヲ	表示] ボタ	ンをクリ	リックする	ると、ファ	ァイルのダ	容が表示
また され JINJI	、一 ます .кsds	覧の[表示 。 s   <sub>表示</sub>	:] アイコンもしくに « 〈 〉 ページ:	は DCB 情	報の [す <sup>2000 行</sup> AS	表示】ボタ	ンをク!	リックする	ると、ファ	ァイルのダ	容が表示

# 3.12 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) [デバッグ] プルダウンメニューの [ステップ イン] を選択します。



2) 画面左下が [準備完了] となり、アタッチ待機状態になったことを確認します。



3) [ソリューション エクスプローラー] 内の vsamwrt2.jcl を右クリックして [サブミット JCL] を選択し、 JCL を実行します。 4) 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。[F11] キーもしくは [デバッグ] 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、マウスオー バーやウォッチタブを利用して変数の値が確認できます。

ICLDEMO	- 🎯 KSDSWRT2	- 🗇 PROC10
C      C     C      C	IN - TORN - W CONTRUCTION CONTRUCTION       WP ROOT THRU PROCENDI;       ** OPEN ERROR ***.       INTL LOOPI       INTA       INDATA       INDATA       INDATA       INTA       INDATA       INPUT BATCHEROO       INTA       INDATA       INTA       INDATA       INDATA	▶
自動		▼ # × 呼び出し履歴
検索 (Ctrl+E)	▶ ↑ ↓ 検索の詳細度: -	名前
名前	値	種類 C:\VS\VCLDEMO\bin\x86\Debug\KSDSWRT2.gnt.PROC1 C:\VS\ICLDEMO\bin\x86\Debug\KSDSWRT2.gnt PARAGRAPH =\SECTION ====+4.
<ul> <li>Ø INREC</li> <li>Ø PREC</li> </ul>	(長さ=71): "00001Soseki Natsume 1-1,Koishik Q 表示 {長さ=71}: "00001Soseki Natsume 1-1,Koishik Q 表示 •	GROUP     ChProgram Files (b8)/Micro Focus/Enterprise Developer/bin6Anspond/microsoft/0/行構築/がありません     ChProgram Files (b8)/Micro Focus/Enterprise Developer/bin6Anspond/microsoft/0/行構築/かりません     ChProgram Files (b8)/Micro Focus/Enterprise Developer/bin6Anspond/microsoft/0/行構築/がかりません     ChProgram Files (b8)/Micro Focus/Enterprise Developer/bin6Anspond/microsoft/0/行振が/がかりません     ChProgram Files (b8)/Micro Focus/Enterprise Developer/bin6Anspond/microsoft/0/行振が/がわません     ChProgram Files (b8)/Micro Focus/Enterprise Developer/bin6Anspond/microsoft/0/行振ががかません     ChProgram Files (b8)/Micro Focus/Enterprise Developer/bin6Anspond/microsoft/0/行振ががかません     ChProgram Files (b8)/Micro Focus/Enterprise Developer/bin6Anspond/microsoft/0/行振ががかません     ChProgram Files (b8)/Micro Focus/Enterprise Developer/bin6Anspond/microsoft/0/files/file//files/fil

5) 希望のステップの左端を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。



6) 先に進める場合は画面上部の [続行] アイコンをクリックします。



7) デバッグを終了させるため、画面上部の [デバッグ停止] アイコンをクリックします。



8) 画面左下が元の色の [準備完了] となり、アタッチ待機状態ではなくなったことを確認します。
 □ 準備完了

#### 3.13 Enterprise Server インスタンスの停止

1) JCLDEMO インスタンスを停止します。



2) JCLDEMO インスタンスの停止を確認後、Visual Studio を終了します。

## 4. 免責事項

本チュートリアルの例題ソースコードは機能説明を目的としたサンプルであり、無謬性を保証するものではありません。例題ソ ースコードは弊社に断りなくご利用いただけますが、本チュートリアルに関わる全てを対象として、二次的著作物に引用する場 合は著作権法の精神に基づき適切な扱いを行ってください。

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。