
Micro Focus Enterprise Developer チュートリアル

メインフレーム COBOL 開発 : JCL

Visual Studio 2019 編

1. 目的

本チュートリアルでは、Visual Studio 2019 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 10 Enterprise
- 使用マシンに Microsoft Visual Studio 2019 がインストールされていること
- 使用マシンに Micro Focus Enterprise Developer 6.0 for Visual Studio 2019 がインストールされていること

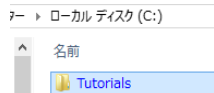
3. チュートリアル手順の概要

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. Enterprise Server インスタンスの設定
7. Enterprise Server インスタンスの開始と確認
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 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。



- 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ーを C:¥ 直下に作成します。

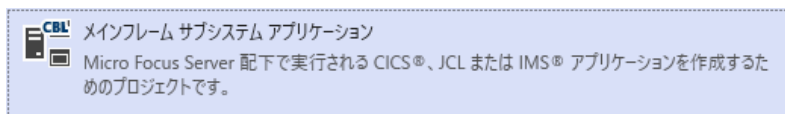
3.2 Visual Studio の起動

- 1) Visual Studio 2019 を起動します。



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、[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。

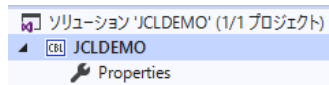


- 2) 下記項目を入力後、[作成] ボタンをクリックします。

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。

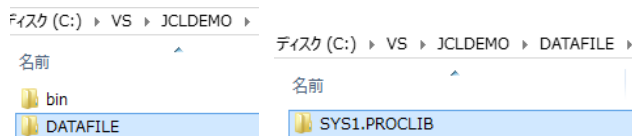


- 3) [ソリューション エクスプローラー] へ作成したプロジェクトが表示されます。



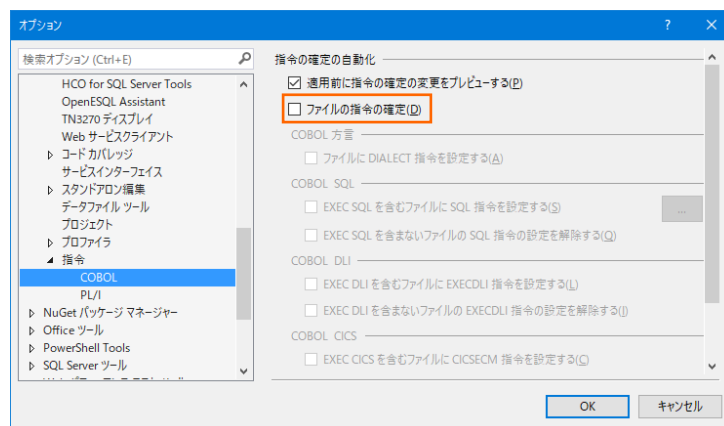
- 4) プロジェクトを作成したことにより C:\¥VS¥JCLDEMO フォルダが作成されていますので、このフォルダ配下にカタログファイルやスプールファイルを格納する C:\¥VS¥JCLDEMO¥DATAFILE フォルダを作成します。

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。

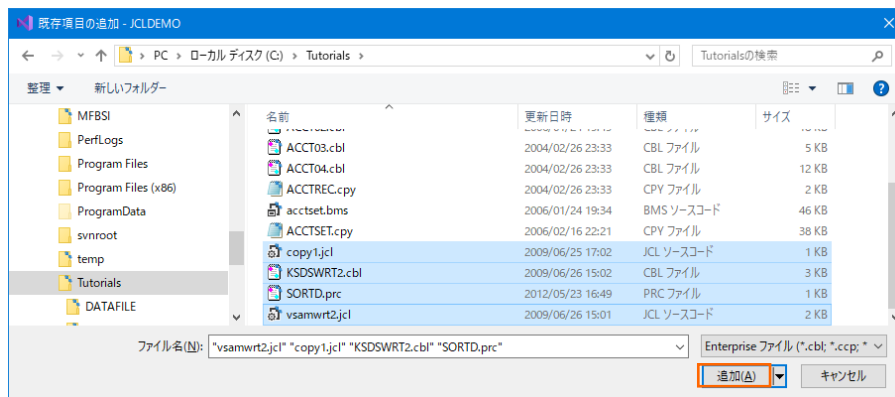


- 5) 既存ファイルのインポート時、自動的にコンパイル指令が指定される機能が用意されていますが、本チュートリアルではこれを解除します。[ツール] プロダクションメニューの [オプション] を選択してオプションウィンドウを表示します。

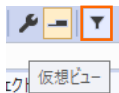
左側ツリービューの [Micro Focus Tools] > [指令] > [COBOL] > [ファイルの指令の確定] チェックボックスをオフにして [OK] ボタンをクリックします。



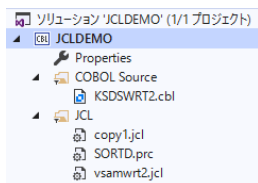
- 6) 用意した例題プログラム類をインポートします。JCLDEMO プロジェクトを右クリックして [追加] > [既存の項目] を選択し、既存項目の追加ウィンドウにて C:\¥Tutorials を指定すると内容が表示されますので、ファイル名先頭に [ACCT] が付いていない下部 4 ファイルを選択して [追加] ボタンをクリックします。この実行により、プロジェクトフォルダへ例題プログラムが配置されます。



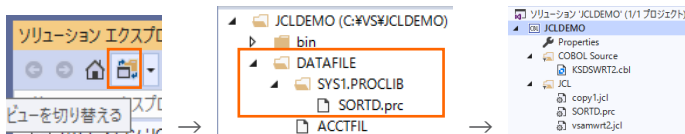
- 7) 種類別に表示するため、[ソリューション エクスプローラー] 内の [仮想ビュー] アイコンをクリックします。



- 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] をフォルダー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 [フォルダービュー] 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。

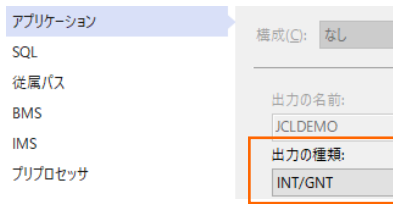


3.4 プロジェクトプロパティの設定

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。

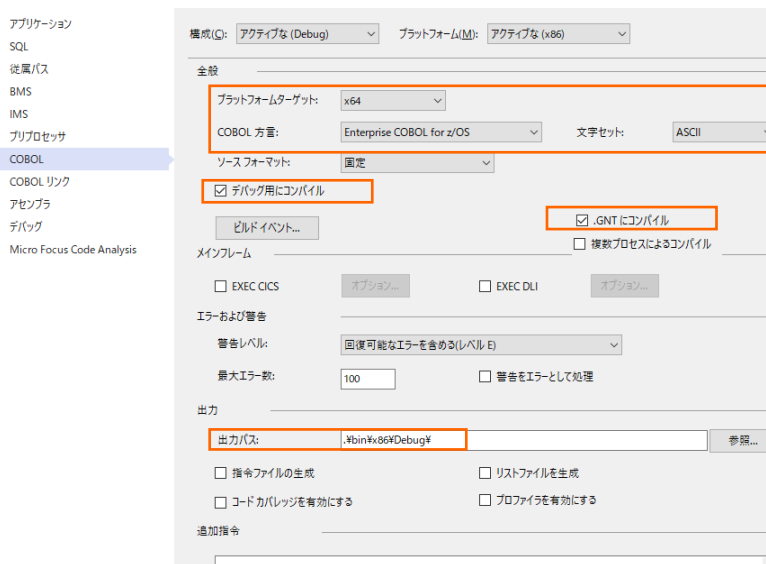


情報

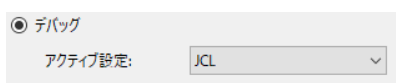
GNT は Micro Focus 独自のオブジェクトで、Micro Focus COBOL ランタイム環境下で実行可能となります。

- 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。



- 4) 左側ツリービューの [デバッグ] を選択して、[アクティブ設定] へ [JCL] を選択します。



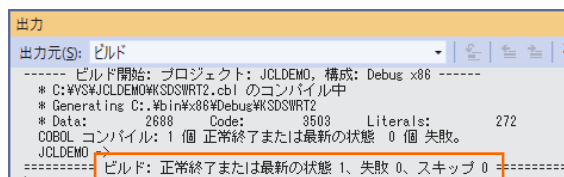
- 5) プロパティファイルを上書き保存します。



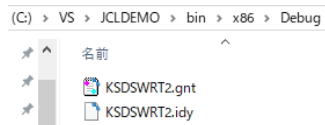
3.5 ビルドの実行

- 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 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 Servers] を右クリックして [管理] を選択します。



- 2) Enterprise Server Administration 画面が表示され、Enterprise Server インスタンス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。

サーバーの追加 (ページ 1 / 3):

サーバー名:

作業モード:

☐ 32ビット ☒ 64ビット

サーバーの作成後に、その作業モードの選択モードの変更は可能です。



重要

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。

- 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 インスタンスが追加されます。

生成オプション:

TN3270 リスナーの作成 ☐ 使用ポート:

→

編集...	MFES (MSS)	JCLDEMO	停止
	64		開始...

- 5) 左にある [編集] ボタンをクリックします。

編集...	MFES (MSS)	JCLDEMO	停止
	64		開始...

- 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。

- ① [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。

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 [適用] 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 イニシエータ (0) プリンター (0)			
ジョブ入力サブシステム有効: ☒			
JES プログラム パス: C:\VS\JCLDEMO\bin\x86\Debug			
システム カタログ: C:\VS\JCLDEMO\DATAFILE\catalog.dat			
データセットの省略時ロケーション: C:\VS\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:			



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。

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

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 イニシエータ (0) プリンター (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 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。

名前:
INITABC

Class:
ABC

説明:
A, B, Cクラスのイニシエータ

- 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

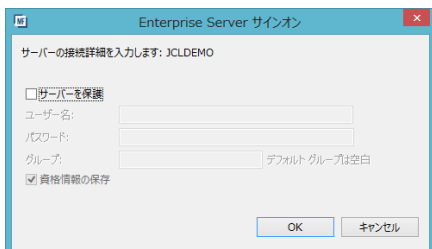
- 1) [サーバーエクスプローラー] 内に JCLDEMO インスタンスが表示されていることを確認します。表示されていない場合は [Micro Focus Servers] を右クリックし、[最新の情報に更新] を選択してリフレッシュしてください。
- 2) [サーバーエクスプローラー] 内の JCLDEMO インスタンスを右クリックし、[プロジェクトに関連付ける] > [JCLDEMO] を選択します。これにより JCLDEMO プロジェクトから実行されるアプリケーションは JCLDEMO インスタンスで処理されることになります。



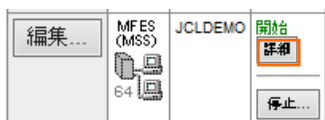
- 3) [JCLDEMO] インスタンスを右クリックして [開始] を選択します。



- 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。

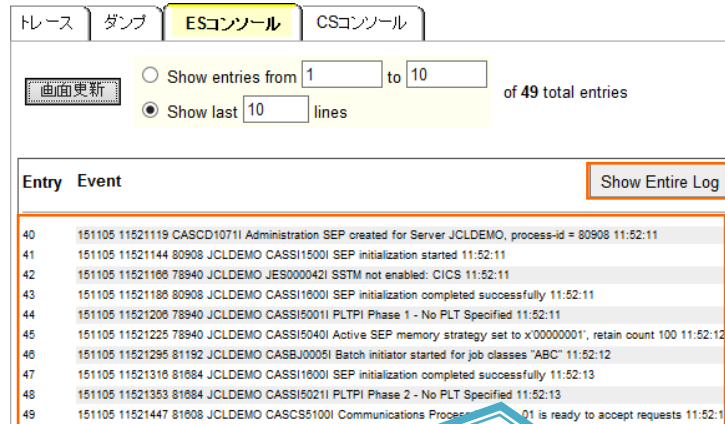


- 5) Enterprise Server Administration 画面へ移動して開始状態であることを確認後、[詳細] ボタンをクリックします。



- 6) [サーバー] > [診断] > [ES コンソール] で JCLDEMO インスタンス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

画面更新 ☐ Show entries from 1 to 10 of 49 total entries ☒ Show last 10 lines

Entry	Event
40	151105 11521119 CASCD10711 Administration SEP created for Server JCLDEMO, process-id = 80908 11:52:11
41	151105 11521144 80908 JCLDEMO CASSI15001 SEP initialization started 11:52:11
42	151105 11521168 78940 JCLDEMO JES0000421 SSTM not enabled: CICS 11:52:11
43	151105 11521188 80908 JCLDEMO CASSI16001 SEP initialization completed successfully 11:52:11
44	151105 11521208 78940 JCLDEMO CASSI50011 PLTPI Phase 1 - No PLT Specified 11:52:11
45	151105 11521225 78940 JCLDEMO CASSI50401 Active SEP memory strategy set to x'00000001', retain count 100 11:52:12
46	151105 11521235 81192 JCLDEMO CASBJ00051 Batch initiator started for job classes "ABC" 11:52:12
47	151105 11521318 81684 JCLDEMO CASSI16001 SEP initialization completed successfully 11:52:13
48	151105 11521353 81684 JCLDEMO CASSI50211 PLTPI Phase 2 - No PLT Specified 11:52:13
49	151105 11521447 81608 JCLDEMO CASC51001 Communications Process 01 is ready to accept requests 11:52:14

Show Entire Log

【JES 機能の正常開始ログ抜粋】

```
JES000051I Job Entry Subsystem (JES) services initialized 13:43:51
JES000059I JES 5 digit job numbering support enabled 13:43:51
CASCD1080I JES Initiator created for Server JCLDEMO, process-id = 3380 13:43:51
CASC5003I Communications interface 01 initialization complete 13:43:51
CASBJ0023I Batch initiator INITABC: class(es) "ABC" 13:43:51
```



注意

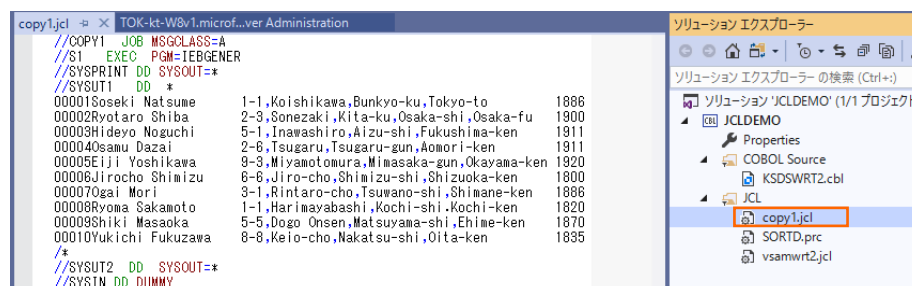
いくつかのサービス開始が失敗してもインスタンスは開始されますので、ログ内容を必ず確認してください。

- 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 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 TOK-kt-W&V1.microf...ver Administration

```
//COPY1 JOB MSGCLASS=A
//S1 EXEC PGM=IEBGENER
//SYSOUT DD SYSOUT=*
//SYST1 DD *
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
00004Osamu Dazai 2-6,Tsugaru,Tsugaru-gun,Aomori-ken 1911
00005Eiji Yoshikawa 9-3,Miyatomomura,Mimasaka-gun,Okayama-ken 1920
00006Jirocho Shimizu 6-6,Jiro-cho,Shimizu-shi,Shizuoka-ken 1880
00007Gai 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
/*
//SYSOUT DD SYSOUT=*
//SYSIN DD DUMMY
```

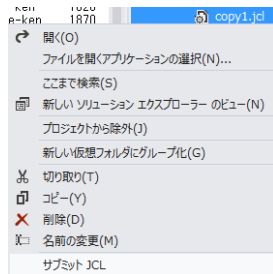
ソリューション エクスプローラー

ソリューション エクスプローラー の検索 (Ctrl+)

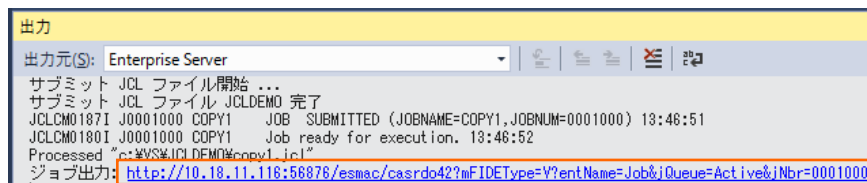
ソリューション 'JCLDEMO' (1/1 プロジェクト)

- JCLDEMO
 - Properties
 - COBOL Source
 - KSDSWRT2.cbl
 - JCL
 - copy1.jcl
 - SORTD.prc
 - vsamwrt2.jcl

- 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) [出力] タブの [出力元] に [Enterprise Server] を選択すると JOB 実行ログと JOB 番号が表示されますので、コントロールキーを押しながらリンクをクリックします。



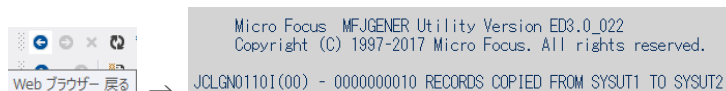
- 4) この JOB 番号にかかわるスプルー一覧が表示されます。先頭の [JESYSMSG] をクリックしてジョブログを確認します。

J0001000	Name: COPY1	Status: Complete				
Hold	Class: A	Priority: 00				
Update	User: JESUSER	COND: 0000				
Delete	File: \$TXRFDR/t000000013.t					
JCLCM0188I J0001000 COPY1 JOB STARTED 13:46:52						
JCLCM0182I J0001000 COPY1 JOB ENDED - COND CODE 0000 13:46:52						
Status	Class	DD Name	Step	Nbr.	Proc Step	Records
Details	Hold	A	JESYSMSG		0	31
Details	Ready	A	SYSPPRINT	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) [web ブラウザー 戻る] アイコンをクリックし、再度スプルー一覧から [SYSPPRINT] をクリックすると、IEBGENERユーティリティの実行ログが確認できます。



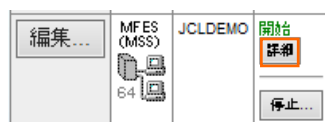
- 7) [web ブラウザー 戻る] アイコンをクリックし、再度スプルー一覧から [SYSUT2] をクリックすると、出力されたスプールの内容が確認できます。

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
00004Osamu 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
00007Ogai 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

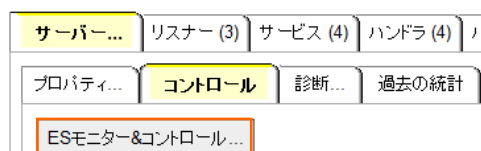
3.9 プロシージャライブラリの作成

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 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 というフォルダーを作成してプロシージャを配置しました。

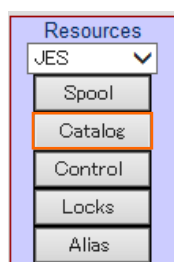
- 1) Enterprise Server Administration 画面へ移動して JCLDEMO インスタンスの [詳細] ボタンをクリックします。



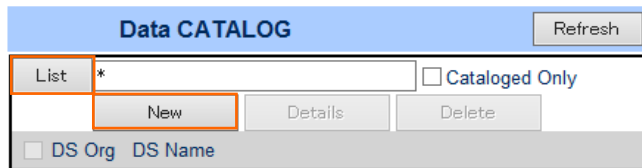
- 2) [サーバー] > [コントロール] > [ES モニター & コントロール] ボタンをクリックします。



- 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ついても [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。

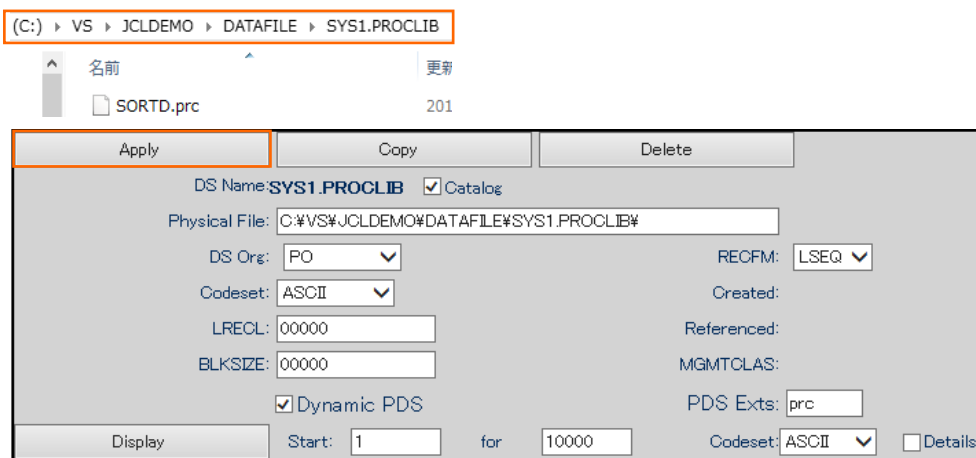


- 4) [List] ボタンをクリックして、カタログ情報の一覧を表示すると、現在は何も登録されていないことが確認できます。新規に作成するため [New] ボタンをクリックします。

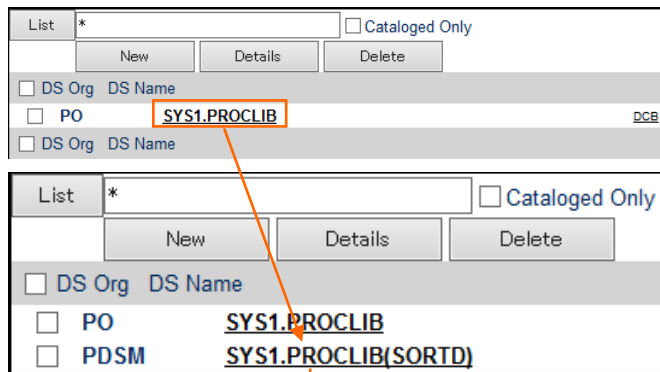


- 5) カタログエントリの画面が表示されますので、以下のように入力し [Apply] ボタンをクリックします。

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	区分データセットである PO を選択します。
RECFM	行順編成の 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。



- 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。



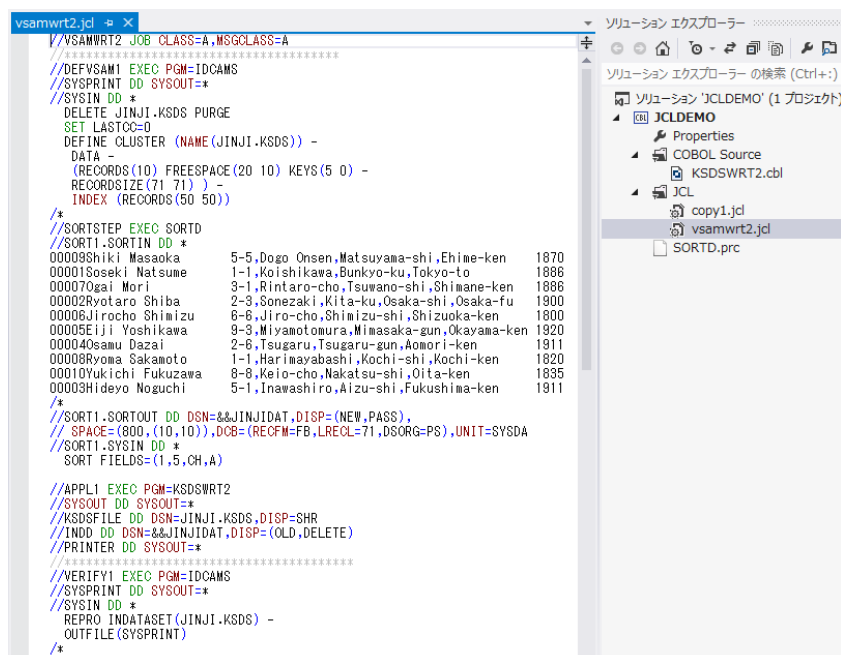
[SYS1.PROCLIB(SORTD)] をクリックすると内容が確認できます。

```
//SORTD PROC
//SORT1 EXEC PGM=SORT
//SYSOUT=*
//SORTWKD1 DD UNIT=SYSDA,SPACE=(CYL,(10,10))
//SORTWKD2 DD UNIT=SYSDA,SPACE=(CYL,(10,10))
// PEND
```

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) [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。



```
VSAMWRT2 JOB CLASS=A,MSGCLASS=A
//*****
//DEFVSAM1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE JINJI.KSDS PURGE
SET LASTOC=0
DEFINE CLUSTER (NAME(JINJI.KSDS)) -
DATA -
(RECORDS(10) FREESPACE(20 10) KEYS(5 0) -
RECORDSIZE(71 71)) -
INDEX (RECORDS(50 50))
/*
//SORTSTEP EXEC SORTD
//SORT1.SORTIN DD *
00009Shiki Masaoka 5-5,Dogo Onsen,Matsuyama-shi,Ehime-ken 1870
00001Soseki Natsume 1-1,Koishikawa,Bunkyo-ku,Tokyo-to 1886
00007Ogai Mori 3-1,Rintaro-cho,Tsuwano-shi,Shimane-ken 1886
00002Ryotaro Shiba 2-3,Sonezaki,Kita-ku,Osaka-shi,Osaka-fu 1900
00006Jirocho Shimizu 6-6,Jiro-cho,Shimizu-shi,Shizuoka-ken 1800
00005Eiji Yoshikawa 9-3,Miyamotomura,Mimasaka-gun,Okayama-ken 1920
00004Osamu Dazai 2-6,Tsugaru,Tsugaru-gun,Aomori-ken 1911
00008Ryoma Sakamoto 1-1,Harinayabashi,Kochi-shi,Kochi-ken 1820
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=(800,(10,10)),DCB=(RECFM=FB,LRECL=71,DSORG=PS),UNIT=SYSDA
//SORT1.SYSIN DD *
SORT FIELDS=(1,5,CH,A)
//APPL1 EXEC PGM=KSDSWRT2
//SYSOUT DD SYSOUT=*
//KSDSFILE DD DSN=JINJI.KSDS,DISP=SHR
//INDD DSN=JINJIDAT,DISP=(OLD,DELETE)
//PRINTER DD SYSOUT=*
//*****
//VERIFY1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
REPRO INDATASET(JINJI.KSDS) -
OUTFILE(SYSPRINT)
/*
```

① 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

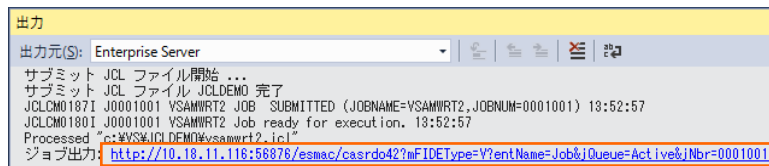
アプリケーション KSDSWRT2 を呼び出しステップ 2 でソートされたデータをステップ 1 で定義した VSAM クラスターに書き込みを行います。同時に書き込まれたデータを DD=PRINTER へ出力します。

④ ステップ 4 : VERIFY1

出力内容確認のため、IDCAMS の REPRO で内容を入力します。

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) [出力] タブの [出力元] に [Enterprise Server] を選択すると JOB 実行ログと JOB 番号が表示されますので、コントロールキーを押しながら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。

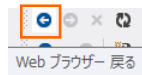
[COND CODE] には [0008] が返却されていますが、これは初回実行時、ステップ 1 で削除該当ファイルが見つからないためです。問題ありません。

J0001001		Name: VSAMWRT2		Status: Complete			
Hold	Class: A			Priority: 00			
Update	User: JESUSER		COND: 0008				
Delete	File: \$TXRFDIR/t000000030.t						
JCLCM0188I J0001001 VSAMWRT2 JOB STARTED 13:52:58							
JCLCM0182I J0001001 VSAMWRT2 JOB ENDED - COND CODE 0008 13:52:59							
	Status	Class	DD Name	Step	Nbr.	Proc Step	Records
Details	Hold	A	JESYSMSG		0		76
Details	Ready	A	SYSPRINT	DEFVSAM1	1		18
Details	Ready	A	SYSOUT	SORTSTEP	2	SORT1	12
Details	Ready	A	SYSOUT	APPL1	3		1
Details	Ready	A	PRINTER	APPL1	3		10
Details	Ready	A	SYSPRINT	VERIFY1	4		41

- 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。

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

- 6) [web ブラウザー 戻る] アイコンをクリックし、スプルー一覧から DEFVSAM1 ステップの [SYSPRINT] をクリックして内容を確認します。



```
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]
```

- 7) [web ブラウザー 戻る] アイコンをクリックし、スプルー一覧から SORTSTEP の [SYSOUT] をクリックしてソート内容を確認します。

```
SORT FIELDS=(1,5,CH,A)
SORT204I: ***** ソート結果 *****
SORT205I: INPUT   ファイル 'SORTIN'
           入力レコード          10 件
           使用レコード          10 件
SORT206I: OUTPUT  ファイル 'SORTOUT'
           入力レコード          10 件
           出力レコード          10 件
SORT399I: Micro Focus MFJSORT ユーティリティ終了
```

- 8) [web ブラウザー 戻る] アイコンをクリックし、スプルー一覧から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 [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

☐ DS Org DS Name

☐ VSAM	JINJI.KSDS	DCB
☐ PO	SYS1.PROCLIB	DCB

Apply Copy Delete

DS Name: JINJI.KSDS ☒ Catalog

Physical File: C:\VS\JCL\DEMO\DATAFILE\JINJI.KSDS.DAT

DS Org: VSAM RECFM: KS

Codeset: ASCII Created: 2015/12/15 16:06:12.87

LRECL: 00071 Referenced: 2015/12/15 16:06:13.12

BLKSIZE: 00000

VSAM Type: Cluster Key Start/Len: 00000/00005

VSAM Attr: Unique Key Max / Avg: 00071/00005

ShareOptions: Cross Region: 0 Cross System: 0

Display Start: 1 for 10000 Codeset: ASCII ☐ 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
00004Osamu Dazai        2-6,Tsugaru,Tsugaru-gun,Aomori-ken     1911
00005Eiji Yoshikawa     9-3,Miyatomomura,Mimasaka-gun,Okayama-ken 1920
00006Jirocho Shimizu    6-6,Jiro-cho,Shimizu-shi,Shizuoka-ken   1800
00007Ogai 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) [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。

プロファイラー

ステップ イン(L)	F11
ステップ オーバー(O)	F10

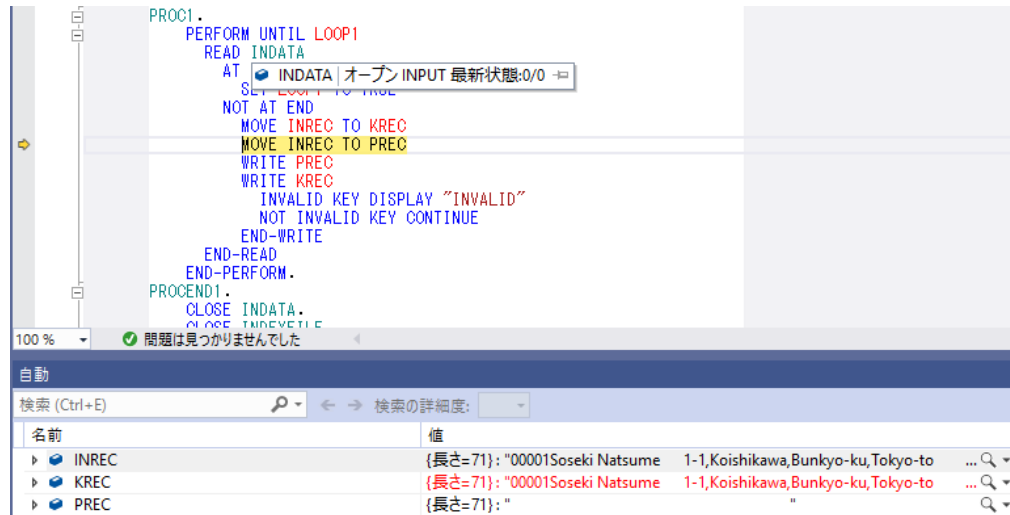
- 2) 画面左下がオレンジ色の [準備完了] となり、アタッチ待機状態になったことを確認します。

呼び出し履歴 イミディエイト ウィンドウ

準備完了

- 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] を選択して、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。



- 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。



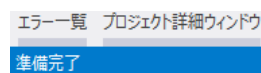
- 6) 先に進める場合は画面上部の [続行] アイコンをクリックします。



- 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。

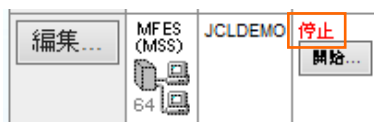


3.12 Enterprise Server インスタンスの停止

- 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。



WHAT'S NEXT

- メインフレーム COBOL 開発 : CICS Visual Studio 2019 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。