
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 7.0 for Visual Studio 2019 がインストールされていること

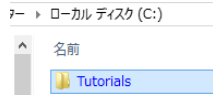
3. チュートリアル手順の概要

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. 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。

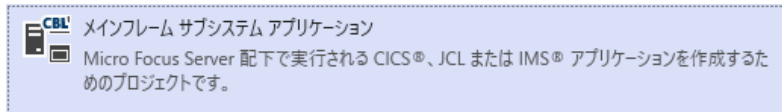
3.2 Visual Studio の起動

- 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。

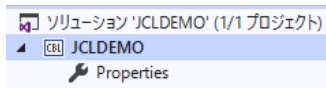


- 2) 下記項目を入力後、[作成] ボタンをクリックします。

| 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。 |

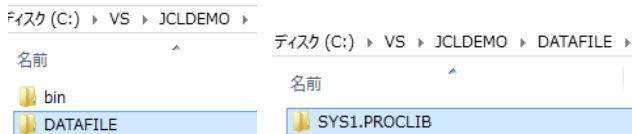


- 3) [ソリューション エクスプローラー] へ作成したプロジェクトが表示されます。

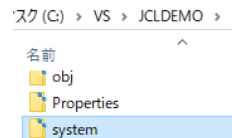


- 4) プロジェクトを作成したことにより C:\%VS%\JCLDEMO フォルダが作成されていますので、このフォルダ配下にカタログファイルやスプールファイルを格納する C:\%VS%\JCLDEMO\DATAFILE フォルダを作成します。

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。

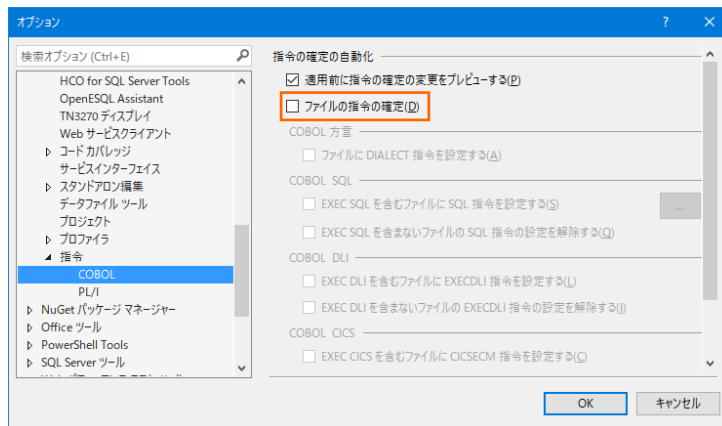


また、実行時に使用する system フォルダを新規作成してください。

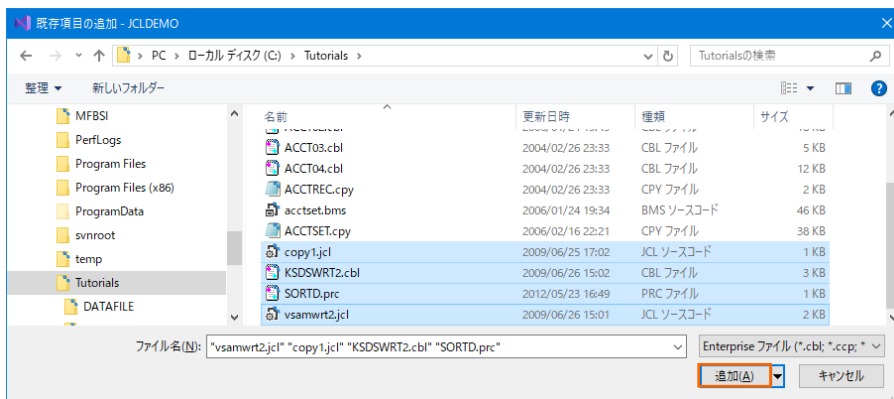


- 5) 既存ファイルのインポート時、自動的にコンパイル指令が指定される機能が用意されていますが、本チュートリアルではこれを解除します。[ツール] プロダクションメニューの [オプション] を選択してオプションウィンドウを表示します。

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。



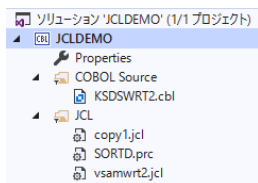
- 6) 用意した例題プログラム類をインポートします。JCLDEMO プロジェクトを右クリックして [追加] > [既存の項目] を選択し、既存項目の追加ウィンドウにて C:\%Tutorials を指定すると内容が表示されますので、ファイル名先頭に [ACCT] が付いていない下部 4 ファイルを選択して [追加] ボタンをクリックします。この実行により、プロジェクトフォルダへ例題プログラムが配置されます。



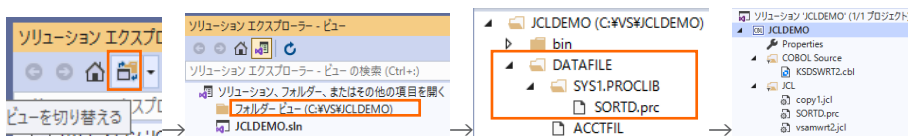
- 7) 種類別に表示するため、[ソリューション エクスプローラー] 内の [仮想ビュー] アイコンをクリックします。



- 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。

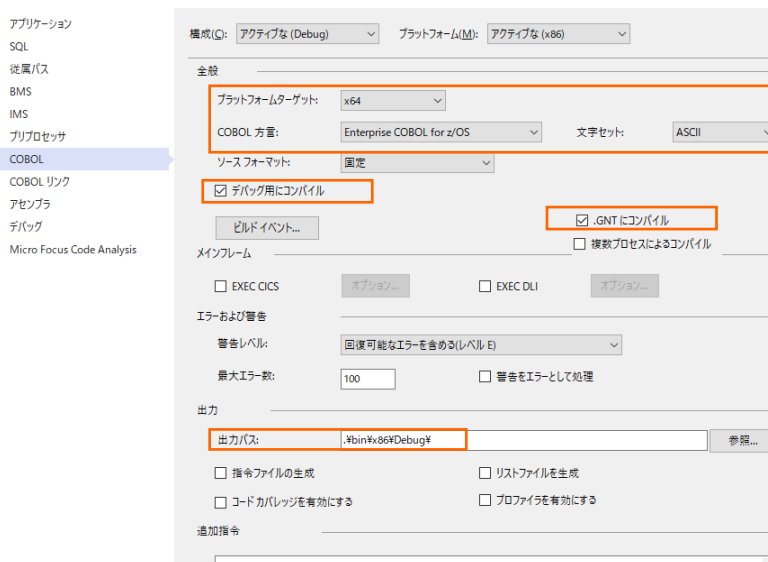


情報

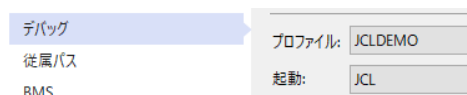
GNT は Micro Focus 独自の動的ロード実行可能ファイルを指します。

- 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。

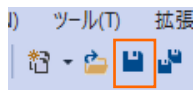
| 項目名 | 説明 |
|----------------|--|
| プラットフォーム ターゲット | 稼働ビット数を指定します。ここでは [x64] を指定します。 |
| COBOL 方言 | COBOL 言語方言を指定します。 例題プログラムは IBM Enterprise COBOL の方言を使用しているため、ここでは [Enterprise COBOL for z/OS] を指定します。 |
| 文字セット | EBCDIC または ASCII を指定します。ここでは [ASCII] を選択します。 |
| デバッグ用にコンパイル | デバッグ実行時に使用するファイルを生成するように指定します。 |
| .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、[プロファイル] には JCLDEMO、[起動] には JCL を指定します。



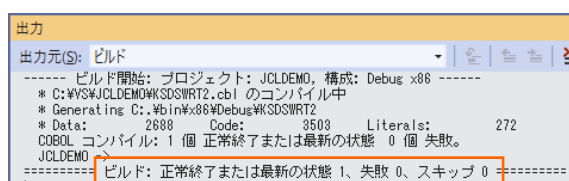
- 5) プロパティファイルを上書き保存します。



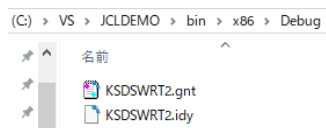
3.5 ビルドの実行

- 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 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へ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。

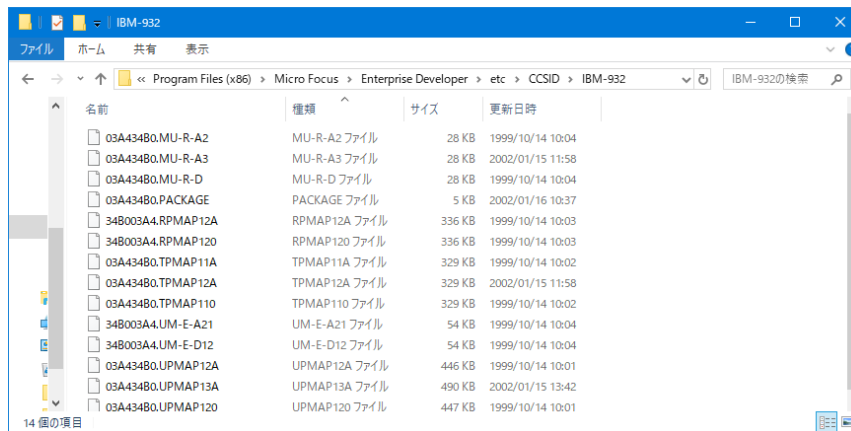
<http://www.microfocus.com/docs/links.asp?vc=cdctables>

- 2) 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

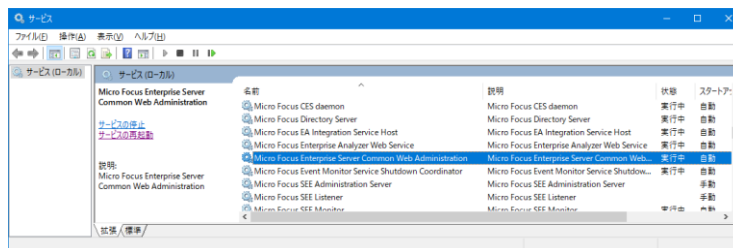
- 3) ダウンロードファイルに含まれている Package2.zip を展開します。
- 4) 展開した Package2 フォルダに含まれる IBM-932.zip を展開します。

- 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。



詳細については、製品マニュアルの「デプロイ > 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。

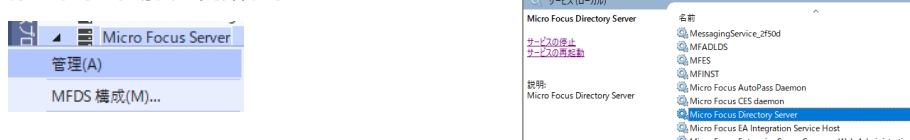
- 6) Windows サービスとして起動している Micro Focus Enterprise Server Common Web Administration を再起動し、インストールした CCSID をロードさせます。



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。

- 7) 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。

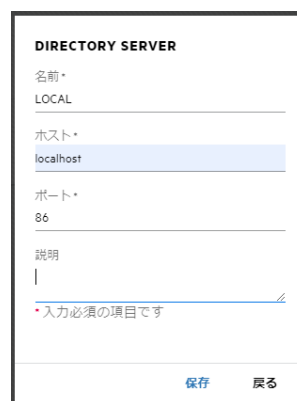


- 8) ブラウザが立ち上がり、Enterprise Server インスタンスを運用、管理する ESCWA が表示されます。ESCWA では Micro Focus Directory Server サービスのポートへ接続して登録された Enterprise Server インスタンスを管理するため、まず接続を作成します。既に存在している場合には、インスタンス作成に進んでください。

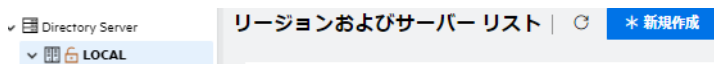
左側ペインで [Directory Server] をクリックし、右側ペインの [追加] ボタンをクリックします。



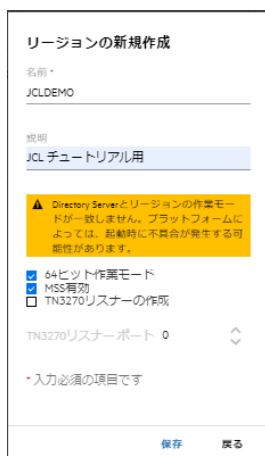
追加ウィンドウでは [名前] は任意ですが、ここでは LOCAL を、
[ホスト] は localhost またはマシンの IP アドレスを、
[ポート] は Micro Focus Directory Server の
デフォルトポートである 86 を入力して [保存] ボタンをクリックします。



- 9) 作成した [LOCAL] 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。



- 10) [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] に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。



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。

- 11) 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 | タイプ | ステータス | 64ビット | MSS有効 | セキュリティ | |
|---------|--------|---------|-------|-------|--------|--|
| JCLDEMO | Region | Stopped | ✓ | ✓ | デフォルト | <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> |

編集

- 12) 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: JCLDEMO システムディレクトリ: C:\VS\JCLDEMO\system | <input checked="" type="checkbox"/> MSS有効 <input checked="" type="checkbox"/> JES有効 <input type="checkbox"/> IMS有効 <input type="checkbox"/> PL/I有効 <input type="checkbox"/> MQ有効 |

- 13) 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。

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

- 14) [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

- 15) 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刻ーション | 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。 |

JESの構成 |

| | |
|---|---|
| JES プログラムパス <input type="text" value="\$proj\bin\x86\Debug"/> | システムカタログ <input type="text" value="\$proj\DATAFILE\catalog.dat"/> |
| データセットの省略時ロケーション <input type="text" value="\$proj\DATAFILE"/> | システムプロシージャ ライブラリ <input type="text" value="SYS1.PROCLIB"/> |

16) [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 |

17) 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

クラス

説明

*入力必須の項目です

18) 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 | ステータスの設定 <input type="text" value="Stopped"/> | 実際のアドレス |
| Disabled | | tcp:0.0.0.0:0 |

19) 画面左側ペインの [LOCAL] をクリックして一覧画面に戻ります。

▼

▼

重要

バージョン 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 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:¥work¥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

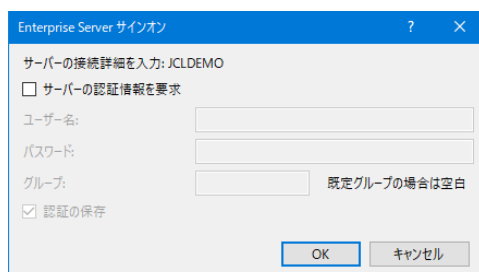
- 1) Visual Studio の [サーバーエクスプローラー] 内に JCLDEMO インスタンスが表示されていることを確認します。表示されていない場合は [Micro Focus Servers] を右クリックし、[最新の情報に更新] を選択してリフレッシュしてください。
- 2) [サーバーエクスプローラー] 内の JCLDEMO インスタンスを右クリックし、[プロジェクトに関連付ける] > [JCLDEMO] を選択します。これにより JCLDEMO プロジェクトから実行されるアプリケーションは JCLDEMO インスタンスで処理されることとなります。



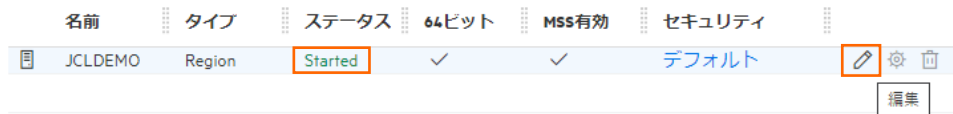
- 3) [JCLDEMO] インスタンスを右クリックして [開始] を選択します。



- 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。

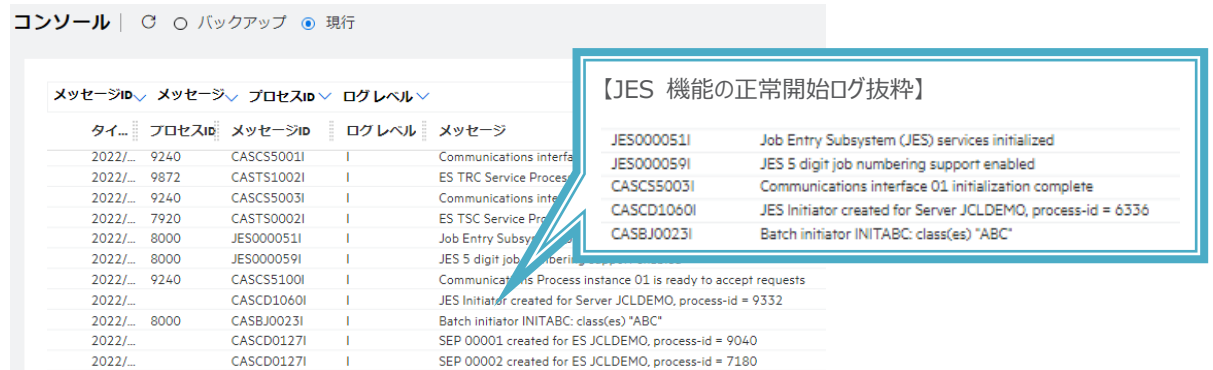


5) ESCWA へ移動して開始状態であることを確認後、[編集] アイコンをクリックします。



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 S や E の場合はエラー表示されます。

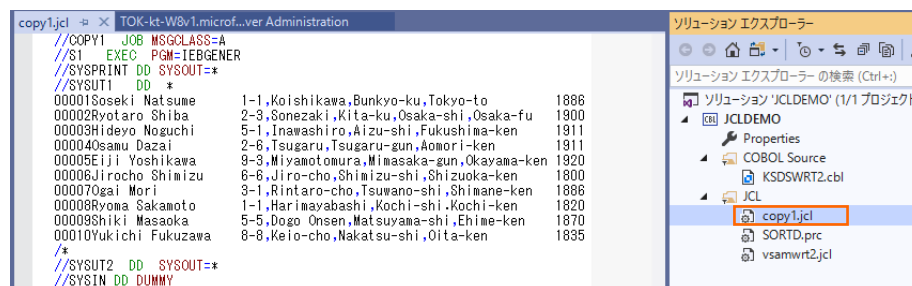


注意
いくつかのサービス開始が失敗してもインスタンスは開始されますので、ログ内容を必ず確認してください。

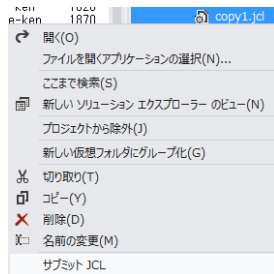
3.9 JCL の実行

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。

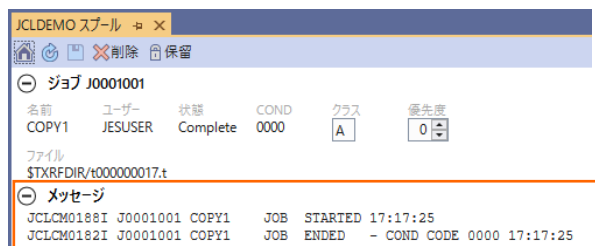
1) [ソリューション エクスプローラー] 内にある JCLDEMO プロジェクト配下の copy1.jcl をダブルクリックし、エディタで内容を確認します。この JCL は IEBGENER ユーティリティを使用して、JCL 内に書かれたインラインデータを SYSOUT に書き出しています。



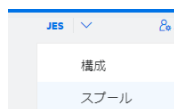
- 2) [ソリューション エクスプローラー] 内の copy1.jcl を右クリックして [サブミット JCL] を選択すると、この JCL が実行されます。



- 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] プログラムメニューから [スプール] を選択します。



- 5) スプルー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。



6) 実行した JOB 番号のスパ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 | DD名 | ステップ | PROCステップ | 状態 | クラス | レコード数 |
|--------------------------------|----------|-----|---------|-------|-------|------|----------|----|-----|-------|
| <input type="checkbox"/> COPY1 | J0001001 | A | JESUSER | 0000 | Hold | A | JESYSMSG | | | 31 |
| | | | | | Ready | A | SYSPRINT | S1 | 1 | 4 |
| | | | | | Ready | A | 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。

```
---> 17:17:25 JCLCM0191I STEP ENDED      S1 - COND CODE 0000
---> 17:17:25 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パール一覧に戻り、[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 ED7.0_PU2E
Copyright (C) Micro Focus 1997-2020. All rights reserved.

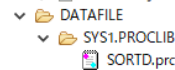
JCLGN0110I(00) - 000000010 RECORDS COPIED FROM SYSUT1 TO SYSUT2
```

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

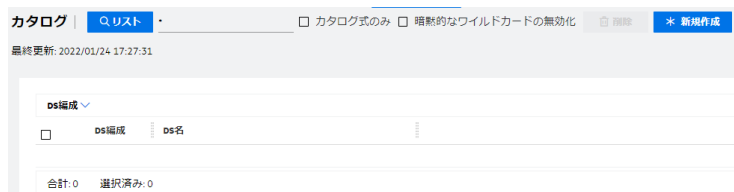
```
00001Soseki Natsume      1-1,Koishikawa,Bunkyo-ku,Tokyo-to      1886
00002Ryotaro Shiba      2-3,Sonezaki,Kita-ku,Osaka-shi,Osaka-fu 1900
00003Hideyo Noguchi    5-1,Inawashiro,Aizu-shi,Fukushima-ken  1911
00004Osamu Dazai       2-6,Tsugaru,Tsugaru-gun,Aomori-ken     1911
00005Eiji Yoshikawa    9-3,Miyatomomura,Mimasaka-gun,Okayama-ken 1920
00006Jirocho Shimizu   6-6,Jiro-cho,Shimizu-shi,Shizuoka-ken   1800
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
```

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。

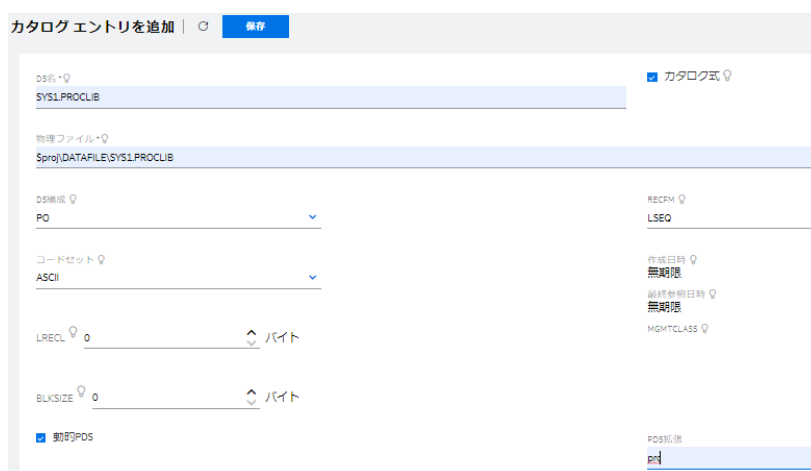


1) ESCWA へ移動して JCLDEMO インスタンスの画面上部にある [JES] プルダウンメニューの [カタログ] を選択し、[リスト] ボタンをクリックすると、現時点では何も登録されていないことがわかります。新しくカタログするために [新規作成] ボタンをクリックします












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。 |



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。

| PO | DS 名 | 操作 |
|--------------------------|--------------------------|---|
| <input type="checkbox"/> | PO SYS1.PROCLIB |    |
| <input type="checkbox"/> | PO SYS1.PROCLIB |    |
| <input type="checkbox"/> | PDSM SYS1.PROCLIB(SORTD) |    |

[SYS1.PROCLIB(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。

SYS1.PROCLIB(SORTD) | 表示 << > ページ

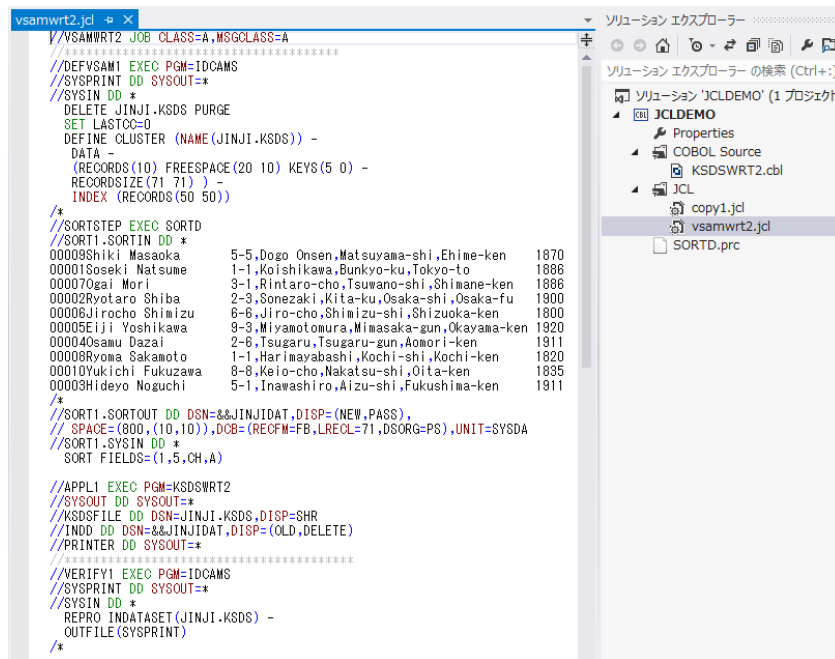
```

//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。



```

vsamwrt2.jcl
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
00007Ogaki Mori         3-1,Rintaro-cho,Tsuwano-shi,Shimane-ken 1886
00002Ryotaro Shiba      2-3,Somezaki,Kita-ku,Osaka-shi,Osaka-fu 1900
00006Jirocho Shimizu    6-6,Jiro-cho,Shimizu-shi,Shizuoka-ken 1800
00005Eiji Yoshikawa     9-3,Miyatomura,Mimasaka-gun,Okayama-ken 1920
00004Osamu Dazai        2-6,Tsugaru,Tsugaru-gun,Aomori-ken    1911
00008Ryoma Sakamoto     1-1,Harimayabashi,Kochi-shi,Kochi-ken  1820
00010Yukichi Fukuzawa   8-8,Keio-cho,Nakatsu-shi,Oita-ken     1835
00003Hideyo Naguchi     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 DD 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

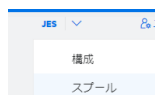
アプリケーション KSDSWRT2 を呼び出しステップ 2 でソートされたデータをステップ 1 で定義した VSAM クラスターに書き込みを行います。同時に書き込まれたデータを DD=PRINTER へ出力します。

- 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] プログラムメニューから [スプール] を選択します。



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

| DD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 | ステップ番 | PROCステ | レコード数 | |
| Hold | A | JESYSMSG | | 0 | | 76 | |
| Ready | A | SYSPRINT | DEFVSAM1 | 1 | | 18 | |
| Ready | A | SYSOUT | SORTSTEP | 2 | SORT1 | 12 | |
| Ready | A | SYSOUT | APPL1 | 3 | | 1 | |
| Ready | A | PRINTER | APPL1 | 3 | | 10 | |
| Ready | A | 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
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パール一覧に戻り、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
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]
```

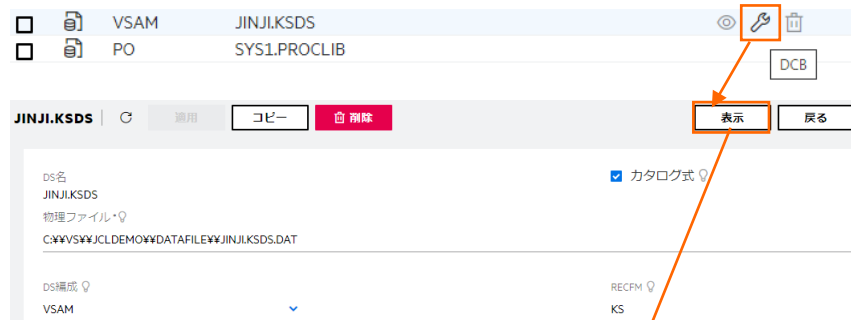
- 7) 右上にある [戻る] ボタンをクリックしてスパール一覧に戻り、SORTSTEP の [SYSOUT] をクリックしてソート内容を確
認します。

```
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'
              使用レコード          10 件
              出力レコード          10 件
SORT399I: Micro Focus MFJ3SORT ユーティリティ終了
```

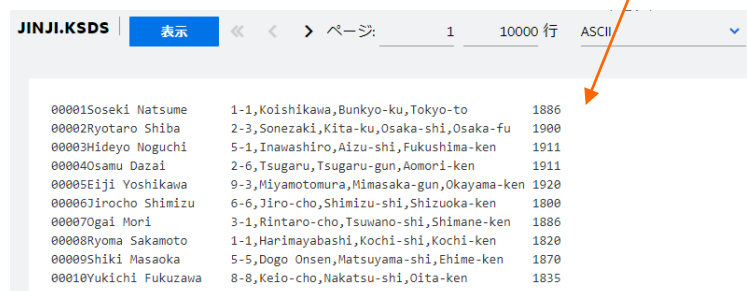
- 8) 右上にある [戻る] ボタンをクリックしてスパール一覧に戻り、他ステップに関しても確認してみてください。

- 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、[DCB] 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。



また、一覧の [表示] アイコンもしくは DCB 情報の [表示] 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。



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。

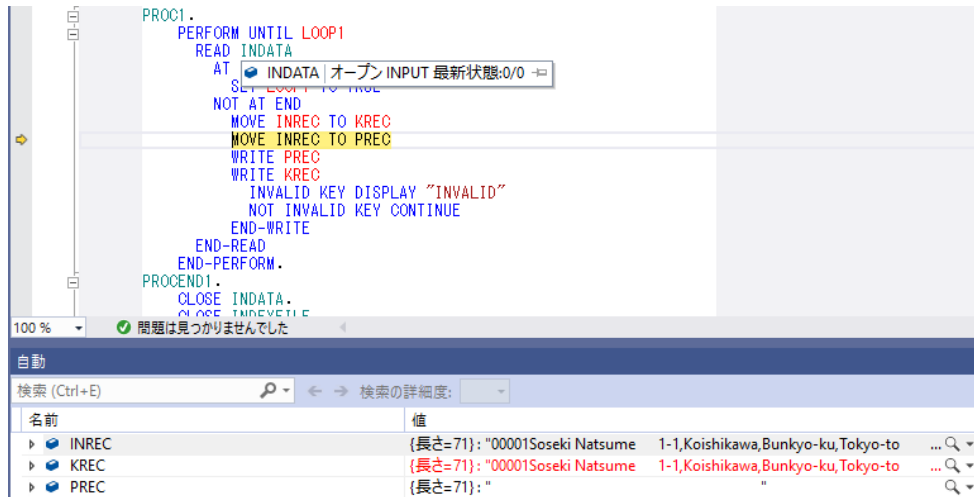
| プロファイター | |
|----------------|-----|
| ⚙️ ステップ イン(L) | F11 |
| 🔄 ステップ オーバー(O) | F10 |

- 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。



- 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。

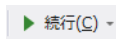
- 4) 少し待つとデバッグセッションが開始して、プログラムのステップ実行が可能になります。[F11] キーもしくは [デバッグ] プルダウンメニューから [ステップ イン] を選択してステップを進めることができ、マウスオーバーやウォッチタブを利用して変数の値が確認できます。



- 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。



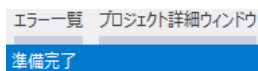
- 6) 先に進める場合は画面上部の [続行] アイコンをクリックします。



- 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。



3.13 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。

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。