

Enterprise Developer チュートリアル

メインフレーム PL/I 開発 : JCL Visual Studio 2022 編

1. 目的

本チュートリアルでは、PL/I 言語で書かれたソースをオープン環境へ移行後、Visual Studio 2022 を使用した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
- Windows 開発環境に Enterprise Developer 9.0 for Visual Studio 2022 がインストール済であること。

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. PL/I ソリューションのインポート
4. プロジェクトプロパティの確認
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 インスタンス開始と確認
9. JCL の実行
10. PL/I ソースのデバッグ
11. 終了処理

3.1 チュートリアルの準備

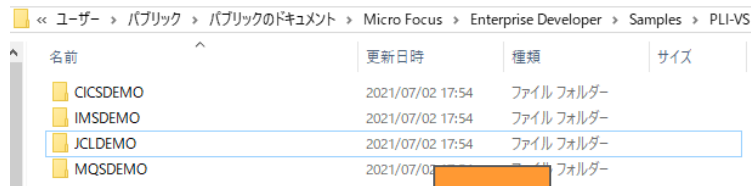
例題プログラムに関連するリソースを用意します。

- 1) Visual Studio のソリューションを保存するフォルダを C:¥ 直下に作成します。

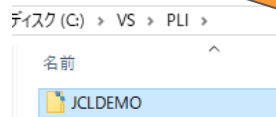
例)C:¥VS¥PLI

- 2) ユーザーのパブリックフォルダ配下に含まれている例題プログラム JCLDEMO フォルダを、作成した C:¥VS¥PLI へコピーします。

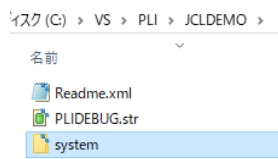
例)C:¥Users¥Public¥Documents¥Micro Focus¥Enterprise Developer¥Samples¥PLI-VS¥JCLDEMO



コピー先)C:¥VS¥PLI¥JCLDEMO

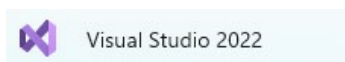


また、実行時に使用する system フォルダを C:¥VS¥PLI¥JCLDEMO 直下に新規作成してください。



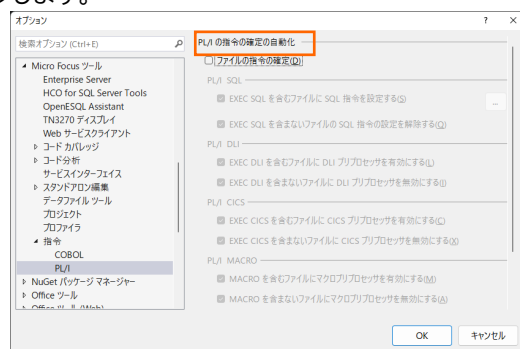
3.2 Visual Studio の起動

- 1) Visual Studio 2022 を起動します。



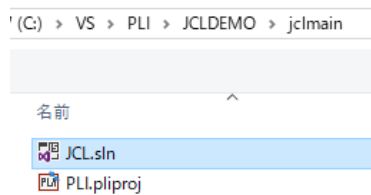
- 2) 既存ファイルのインポート時、自動的にコンパイル指令が指定される機能が用意されていますが、本チュートリアルではこれを解除します。[ツール] ブルダウンメニューの [オプション] を選択してオプションウィンドウを表示します。

左側ツリーメニューの [Micro Focus ツール] > [指令] > [PL/I] > [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。

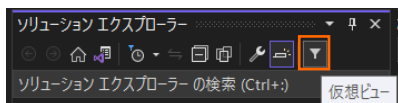


3.3 PL/I ソリューションのインポート

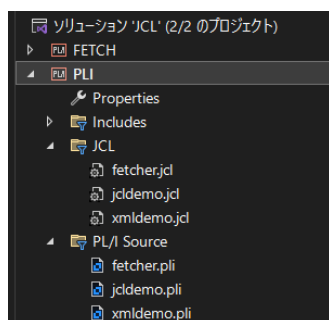
- 1) 用意した例題ソリューションを表示します。[ファイル] プルダウンメニューから [開く] > [プロジェクト/ソリューション] を選択し、[プロジェクト/ソリューションを開く] ウィンドウにて前項でコピーした C:\¥VS¥PLI¥JCLDEMO¥jclmain に存在する JCL.sln を選択後 [開く] ボタンをクリックします。



- 2) 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。



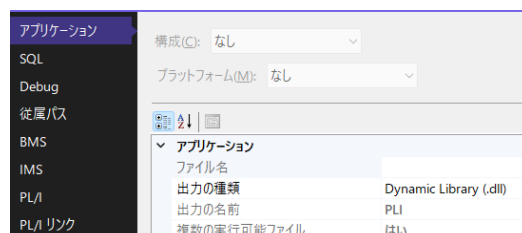
- 3) [ソリューション エクスプローラー] にインポートしたソリューションと 2 つのプロジェクトが表示され、[PLI] プロジェクトを展開すると PL/I ソースや JCL などが確認できます。



3.4 プロジェクトプロパティの確認

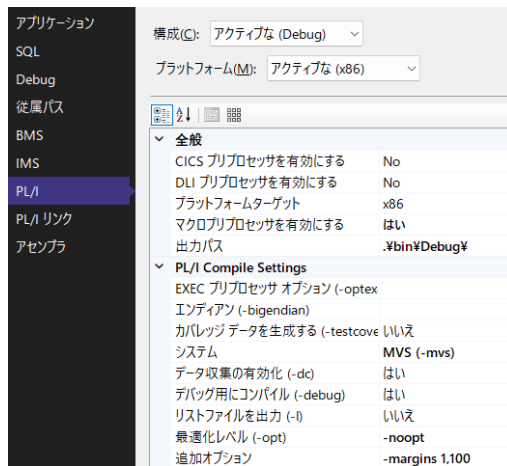
プロジェクトの設定値を確認していきます。

- 1) [ソリューション エクスプローラー] 内の [PLI] プロジェクトの [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) 左側メニュー [アプリケーション] を選択すると、実行ファイルとして DLL を指定していることが確認できます。

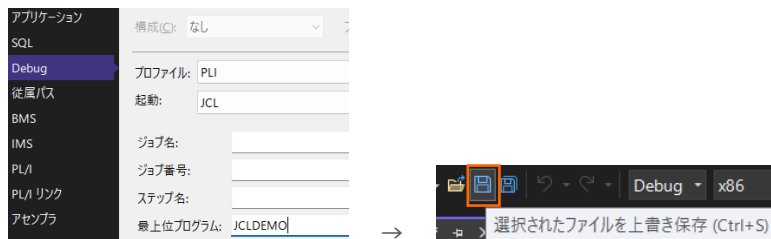


- 3) 左側メニュー [PL/I] を選択すると下記項目が確認できます。出力パスを変更して保存してください。

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。x86 が選択されており 32-bit 稼働が指定されています。
マクロプリプロセッサを有効にする	デバッグ用に指定する追加オプションのため、ここでは ”はい” を指定します。
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が、ここでは Enterprise Server インスタンスの設定値と合わせるために .¥bin¥Debug¥ を指定します。
システム	ここでは “MVS” が指定されているため、JCL を対象としています。
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。

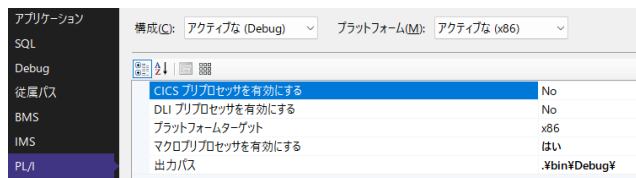


- 4) 左側メニュー [Debug] を選択し、[プロファイル] には PLI を、[起動] には JCL を、[最上位プログラム] には JCLDEMO を指定し、設定を保存します。



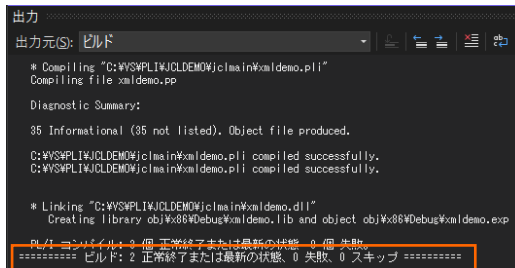
- 5) [ソリューション エクスプローラー] 内の [FETCH] プロジェクトの [Properties] をダブルクリックしてプロパティウィンドウを表示します。

[出力パス] に Enterprise Server インスタンスの設定値と合わせて .¥bin¥Debug¥ を指定します。

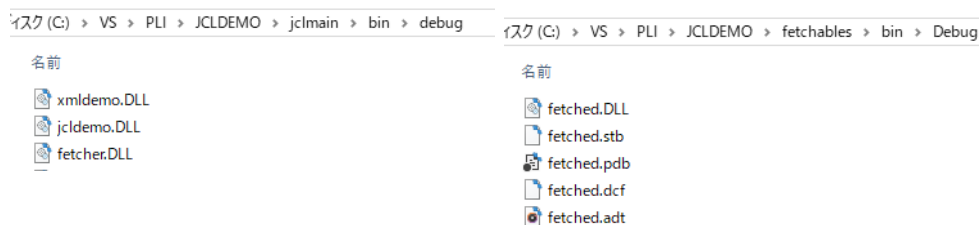


3.5 ビルドの実行

- 1) [ソリューション エクスプローラー] の JCL 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 DLL 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>

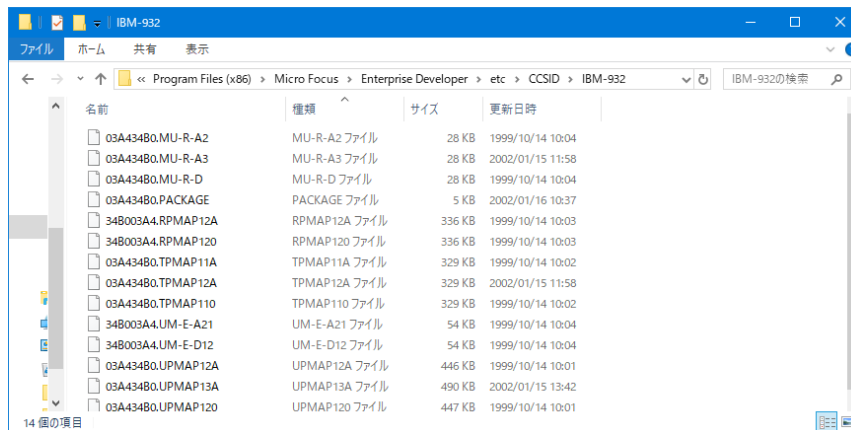
- 2) 製品インストールフォルダ配下の etc フォルダに CCSID フォルダがない場合はこれを作成します。

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。

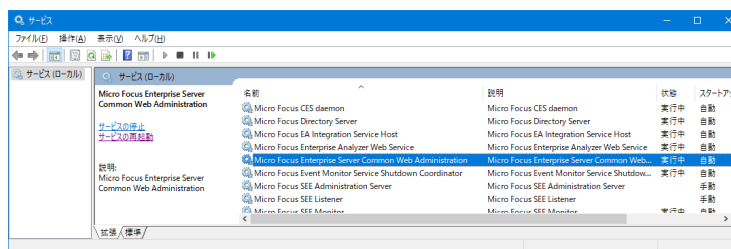
- 4) 展開した Package2 フォルダに含まれる IBM-932.zip を展開します。

- 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。



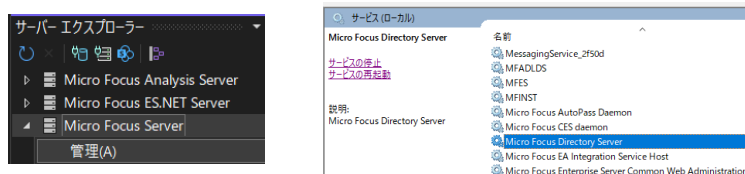
詳細については、製品マニュアルの「ディプロイ > 構成および管理 > Enterprise Server の構成および管理 > Enterprise Server Common Web Administration > [Native] > [Directory Servers] > リージョンとサーバー > リージョン > エンタープライズ サーバー リージョンの文字エンコーディングのサポート」をご参照ください。

- 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。



3.7 Enterprise Server インスタンスの設定

- 1) PL/I を実行するためのエンジンを搭載した 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。



- 2) 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 Default 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。

Directory Server

名前 *
Default

ホスト *
localhost

ポート *
86

説明

* 入力必須の項目です

保存 戻る

- 3) 例題のフォルダには Enterprise Server インスタンスのサンプルが含まれており、これをインポートします。
C:\¥VS¥PLI¥JCLDEMO¥JCL_SERVERS.xml がインポート対象のファイルです。
作成した【Default】接続をクリックし、右側ペインの【インポート】ボタンをクリックします。



- 4) 【インポートタイプの選択】では【XML】を選択して【次へ】をクリックします。

インポート

手順 1: インポートタイプ ✓

手順 2: アップロード

インポートタイプの選択

☐ JSON
クライアント ブラウザが実行されているホスト上のファイルを選択して、json ファイルをインポートします。

☒ XML
クライアント ブラウザが実行されているホスト上のファイルを選択して、xml ファイルをインポートします。

☐ レガシー
Directory Server が実行されているホスト上のディレクトリの場所を選択して、レガシー リポジトリ (.dat ファイルのディレクトリ) をインポートします。

- 5) 【ファイルのアップロード】をクリックし、XML ファイルを指定します。

インポート

手順 1: インポートタイプ ✓

手順 2: アップロード

ファイルのアップロード

クリックまたはドラッグ&ドロップしてファイルをアップロード

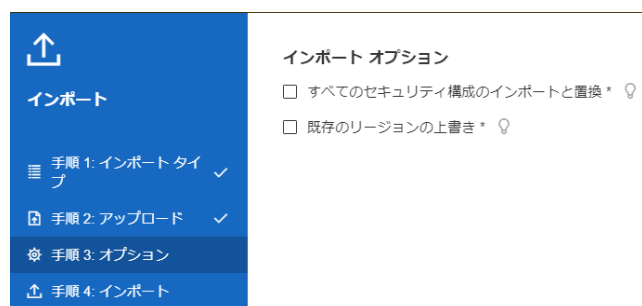
- 6) 【ファイルを受け付けました】メッセージを確認後、【次へ】ボタンをクリックします。

ファイルのアップロード

クリックまたはドラッグ&ドロップしてファイルをアップロード

ファイルを受け付けました。続行してください。

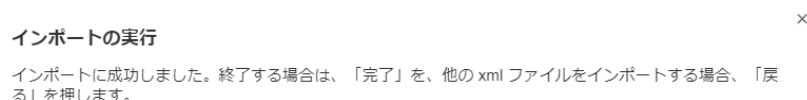
7) 手順 3 では [次へ] をクリックします。



8) 手順 4 では [インポート] をクリックします。



9) インポートの完了が通知されますので、[完了] ボタンをクリックします。



10) 32 ビットアプリケーション稼働用の PLIJCL と、64 ビットアプリケーション稼働用の PLIJCL64 インスタンスが追加されます。Visual Studio では 32 ビットアプリケーション稼働用にコンパイルしましたので、ビット数が一致する PLIJCL インスタンスを使用します。

	名前	タイプ	ステータス	64ビット	MSS有効	セキュリティ
📄	PLIJCL64	Region	Stopped	✓	✓	デフォルト
📄	PLIJCL	Region	Stopped		✓	デフォルト



重要

アプリケーション稼働ビット数 = 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) PLIJCL 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。



12) 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、[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。指定後は[適用] ボタンをクリックします。

一般的なプロパティ | 適用 削除

開始オプション

名前: PLI/JCL	システムディレクトリ: C:\VS\PLI\JCLDEMO\system	☒ 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	

- 13) [リージョンの機能] では、[MSS 有効]、[JES 有効]、[PL/I 有効] のチェックがオンになっていることを確認します。

リージョンの機能

☒ MSS 有効 ☒ JES 有効 ☐ IMS 有効 ☒ PL/I 有効

☐ MQ 有効

- 14) [追加設定] の [構成情報] 欄に環境変数を設定します。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。

変更前)

[ES-Environment]
JBASE=C:\Users\Public\Documents\Micro Focus\Enterprise Developer\Samples\PLI-VS or
PLI-Eclipse\JCLDEMO
JDEMO=\$JBASE\jclmain
JFETCH=\$JBASE\fetchables
JDEBINIT=\$JBASE\debinit
JCBLBASE=\$JBASE\cblmain

変更後)

[ES-Environment]
JBASE=C:\VS\PLI\JCLDEMO
JDEMO=\$JBASE\jclmain
JFETCH=\$JBASE\fetchables
JDEBINIT=\$JBASE\debinit
JCBLBASE=\$JBASE\cblmain
MFACCCGI_CHARSET=Shift_JIS

追加設定

構成情報

[ES-Environment]
JBASE=C:\VS\PLI\JCLDEMO
JDEMO=\$JBASE\jclmain
JFETCH=\$JBASE\fetchables
JDEBINIT=\$JBASE\debinit
JCBLBASE=\$JBASE\cblmain
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確認します。構成情報に指定した環境変数を使用して値が入力されています。

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可能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セッション	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。

JESの構成 | 適用

JES プログラム パス	システム カタログ
\$JDEMO\bin\debug;\$JFETCH\bin\debug;\$JDEBINIT\bin\debug	\$JBASE\plijcl_base\catalog.dat
データセットの省略時ロケーション	システム プロシージャ ライブラリ
\$JBASE\plijcl_base	
Fileshare 構成ロケーション	

16) [イニシエータ] には A ~ 9 までのクラスに対するイニシエータが設定されていることを確認します。

イニシエータ | * 新規作成

名前	クラス	説明
INIT1	abcdefghijklmnopq...	

JESイニシエータ

名前: INIT1

クラス: abcdefghijklmnopqrstuvwxyz0123456789

説明:

* 入力必須の項目です

17) 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] になっていることがあります。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。例題インスタンスが既に [Stopped] の場合はこれを確認します。

TLS設定

ステータス	ステータスの設定	実際のアドレス
Disabled	Stopped	tcp:0.0.0.0:0

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 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%\PLI\%JCLDEMO%\plijcl_base\catalog.dat
```

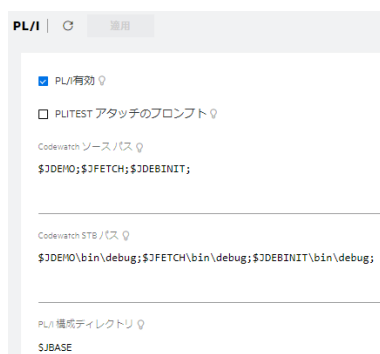
```
mfsplcnv -2
```

詳しくは製品マニュアルをご参照ください。

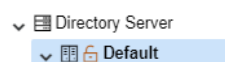
18) 画面上部の [一般] プルダウンメニューから [PL/I] を選択し、表示される画面の各項目を確認します。

項目名	説明
PL/I 有効	オン、オフ指定します。ここではオンを指定します。
Codewatch ソース パス	デバッグで使用するソースファイルのパスを指定します。

Codewatch STB パス	デバッグで使用するデバッグファイルのパスを指定します。例)XXX.stb
PL/I 構成ディレクトリ	プロジェクトのパスを指定します。

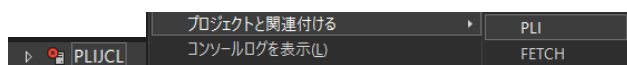


19) 画面左側ペインの [Default] をクリックして一覧画面に戻ります。



3.8 Enterprise Server インスタンスの開始と確認

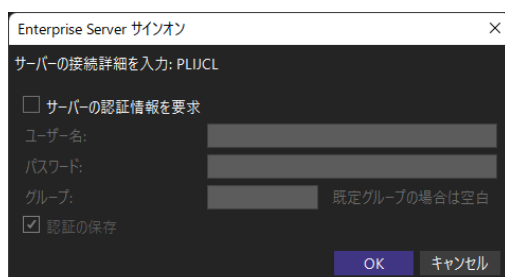
- 1) Visual Studio の [サーバー エクスプローラー] 内に PLIJCL インスタンスが表示されていることを確認します。表示されていない場合は [Micro Focus Servers] を右クリックし、[最新の情報に更新] を選択してリフレッシュしてください。
- 2) [サーバー エクスプローラー] 内の PLIJCL インスタンスを右クリックし、[プロジェクトと関連付ける] > [PLI] を選択します。これにより PLI プロジェクトから実行される JCL は PLIJCL インスタンスで処理されることになります。



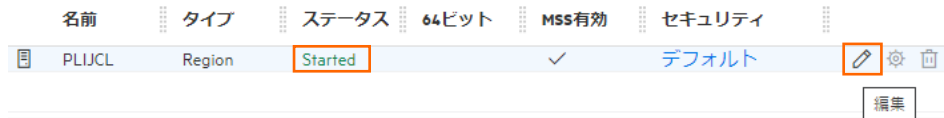
- 3) PLIJCL インスタンスを右クリックして [開始] を選択します。



- 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 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。

コンソール | 現行

メッセージID	メッセージ	プロセスID	ログレベル
2022...	10356	CASSI1500I	I
2022...	10692	CASCD1071I	I
2022...	10692	CASSI1500I	I
2022...	9060	CASSI1744I	I
2022...	10692	CASSI1744I	I
2022...	10692	CASSI1600I	I
2022...	9060	CASSI1600I	I
2022...	9060	CASSI4100I	I
2022...	9060	CASSI4103I	I

【PL/I, JES 機能の正常開始ログ抜粋】

CASSI1744I	I	MFPLI support loaded successfully
JES000051I	I	Job Entry Subsystem (JES) services initialized
JES000059I	I	JES 5 digit job numbering support enabled
CASCD1060I	I	JES Initiator created for Server PLIUCL, process-id = 11160
CASBJ0023I	I	Batch initiator INIT1: class(es) *abcdefghijklmnopqrstuvwxyz01:
		MFPLI support loaded successfully
		MFPLI support loaded successfully
		SEP initialization completed successfully
		SEP initialization completed successfully
		PLTPI Phase 1 Processing Starting
		PLTPI starting JES Phase 1

注意

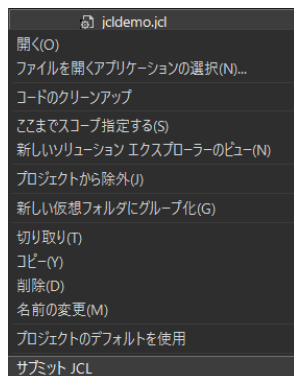
いくつかのサービス開始が失敗してもインスタンスは開始されますので、ログ内容を必ず確認してください。

3.9 JCL の実行

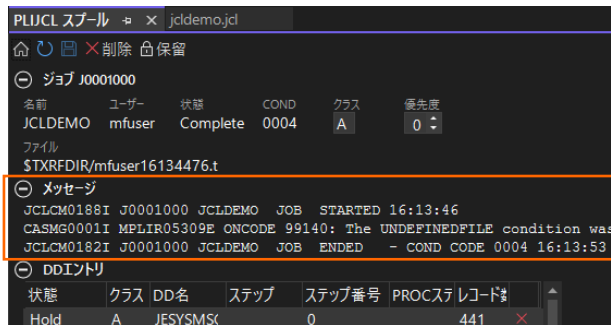
- 1) Visual Studio の[ソリューション エクスプローラー]内に存在する[PLI]プロジェクトの jcldemo.jcl をダブルクリックして内容を表示します。IDCAMS などのユーティリティを使用してファイルを操作したのち、JCLDEMO プログラムを実行していることがわかります。

```
*****
/* Run the JCLDEMO Program
*****
//STEP100 EXEC PGM=JCLDEMO
//SYSOUT DD SYSOUT=*,HOLD=Y
//SYSPRINT DD SYSOUT=*,HOLD=Y,DCB=(RECFM=LSEQ)
//B107925B DD DISP=(,CATLG),SPACE=(CYL,(5,5),RLSE),
//          DCB=(RECFM=FBA,LRECL=137,BLKSIZE=0),
//          DSN=SYSAD.STREAM.TEST
//FBFILE DD DSN=SYSAD.QSAM.TESTFILE,DISP=OLD
```

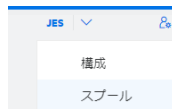
- 2) [ソリューション エクスプローラー] から jcldemo.jcl を選択して右クリック後、[サブミット JCL] を選択すると、この JCL が実行されます。



- 3) [PLIJCL 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。PLIJCL 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
JCLDEMO	J0001000	A	JESUSER	0004

DDエントリ

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		441		
 Ready	A	SYSPRINT	STEP00	1		59		
 Hold	A	SYSPRINT	STEP1	2		16		
 Hold	A	SYSPRINT	STEP2	3		7		
 Hold	A	SYSPRINT	STEP3	4		25		
 Hold	A	SYSPRINT	STEP4	5		57		
 Hold	A	SYSPRINT	STEP05	6		4		
 Ready	Y	SYSPRINT	STEP10	7		4		

- 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、ステップ名 STEP60 から STEP090 でリターンコードに 0004 が返却されていることがわかります。
- 8) 右上にある [戻る] ボタンをクリックしてスプルー一覧に戻り、[STEP60] の [SYSPRINT] をダブルクリックすると、最終行にワーニングが発生しており、JCL で指定した 100 件のレコードを下回ったため発生した警告と判断できます。

```

REPRO INFILE(IN) OUTFILE(OUT) COUNT(100)
JCLAM0134I(00) - 00000009 Records processed.
JCLAM0194W(04) - Number of records read was less than COUNT(00000100).

```

- 9) Jcldemo.jcl の STEP60 から STEP090 に記述されている COUNT(100) を COUNT(5) へ修正して保存し、JCL を再実行します。

```
//SYSIN DD *  
      REPRO INFILE(IN) OUTFILE(OUT) COUNT(5)  
/*
```

- 10) 前項同様の手順で [JESYSMSG] 内容を確認すると、全てのステップが正常に終了していることがわかります。

```
JCLCM0182I J0001001 JCLDEMO JOB ENDED - COND CODE 0000 15:10:32
```

- 11) STEP100 では jcldemo.pli ソースから出力された内容が参照できますので、ソースコードと合わせて確認してみてください。

	Hold	A	SYSPRINT	STEP100	16	110
	Hold	A	FBOUT	STEP100	16	9
	Hold	A	TABLE	STEP100	16	422

- 12) 実行された JCL から作成されたカタログ情報を確認します。[JES] プルダウンメニューの [カタログ] を選択し、[リスト] ボタンをクリックし、一覧を表示します。

[DCB] アイコンをクリックするとカタログされたファイルの情報が表示され、変更も可能です。

☐	DS編成	DS名	
☐	VSAM	SYSAD.CLUSTER.AIX	
☐	VSAM	SYSAD.CLUSTER.BASE	
☐	VSAM	SYSAD.CLUSTER.BASE.DATA	
☐	VSAM	SYSAD.CLUSTER.BASE.INDEX	
☐	VSAM	SYSAD.CLUSTER.PATH	
☐	PS	SYSAD.QSAM.TESTFILE	
☐	?	SYSAD.STREAM.TEST	
☐	PS	SYSAD.TABLE5	
☐	PS	SYSAD.TABLE6	
☐	PS	SYSAD.VBFILE	
☐	PS	SYSAD.VBOUT	
☐	VSAM	SYSAD.VSAM.ESDS.TESTFILE	
☐	VSAM	SYSAD.VSAM.KSDS.TESTFILE	
☐	VSAM	SYSAD.VSAM.KSDS2.TESTFILE	
☐	VSAM	SYSAD.VSAM.RRDS.TESTFILE	

SYSAD.CLUSTER.BASE.DATA     

* 入力必須の項目です

DS名
SYSAD.CLUSTER.BASE.DATA 

物理ファイル 
C:\VSW\PLI\JCLDEMO\PLI\JCL_BASE\SYSAD.CLUSTER.BASE.DAT

DS編成 
VSAM

コードセット 
ASCII

RECFM 
KS

作成日時 
Thu Sep 14 2023

- 13) 画面右上の [表示] ボタンをクリックするとデータが参照できます。

SYSAD.CLUSTER.BASE.DATA      **表示**

RECORD01 AIX9 DATA-010101010101
RECORD02 AIX4 DATA-020202020202
RECORD03 AIX5 DATA-030303030303
RECORD04 AIX4 DATA-040404040404
RECORD05 AIX7 DATA-050505050505
RECORD06 AIX2 DATA-060606060606

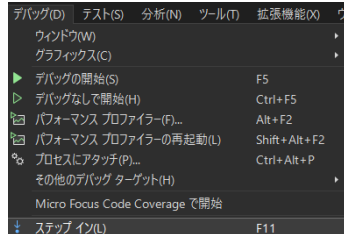
14) また、このカタログ機能からカタログの新規作成や削除も可能です。



3.10 PL/I ソースのデバッグ

JCL から実行される PL/I 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。

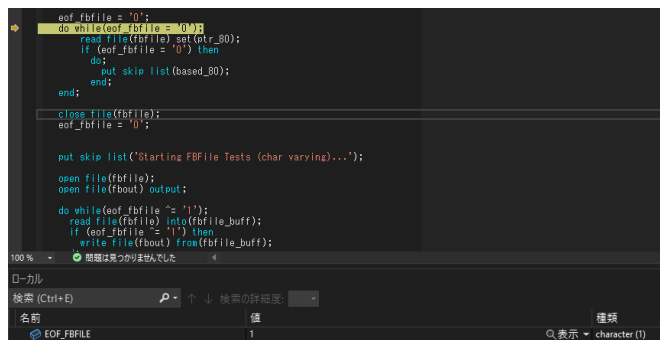


2) 画面左下が [準備完了] となり、アタッチ待機状態になったことを確認します。



3) [ソリューション エクスプローラー] 内の jcldemo.jcl を右クリックして [サブミット JCL] を選択し、JCL を再度実行します。

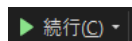
4) 少し待つとデバッグセッションが開始して、プログラムのステップ実行が可能になります。[F11] キーもしくは [デバッグ] プルダウンメニューから [ステップ イン] を選択してステップを進めることができ、マウスオーバーやウォッチタブを利用して変数の値が確認できます。



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。



3.11 終了処理

- 1) Visual Studio の [サーバー エクスプローラー] 内で PLIJCL インスタンスを右クリックして [停止] を選択し、開始中のインスタンスを停止します。



- 2) PLIJCL インスタンスの停止状態を確認後に、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。