

# 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 Enterprise Developer 11.0J 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

### 4. 免責事項

### 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。

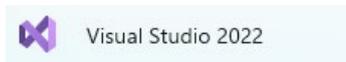
- 1) 使用する例題プログラムは、キットに添付されている Tutorials.zip に圧縮されています。これを C:¥直下に解凍します。



- 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

Visual Studio 2022 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、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。

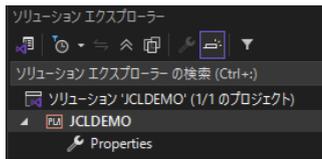


- 2) 下記項目を入力後、[作成] ボタンをクリックします。

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。



- 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。

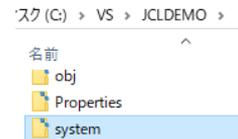


- 4) プロジェクトを作成したことにより C:\¥VS¥JCLDEMO フォルダが作成されますので、このフォルダ配下にカタログファイルやスプールファイルを格納する C:\¥VS¥JCLDEMO¥DATAFILE フォルダを作成します。

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。

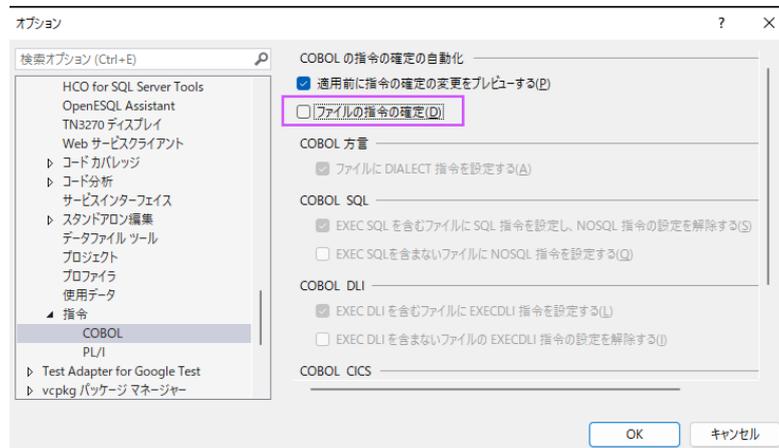


また、実行時に使用する system フォルダを新規作成してください。

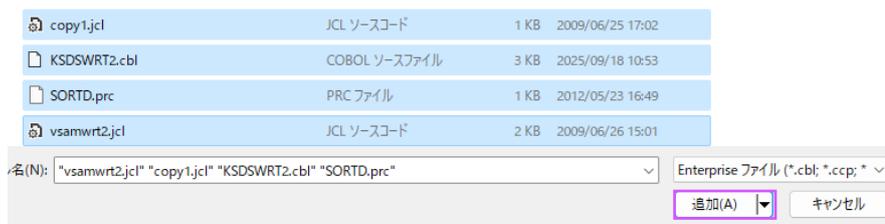


- 5) 既存ファイルのインポート時、自動的にコンパイル指令が指定される機能が用意されていますが、本チュートリアルではこれを解除します。Visual Studio の [ツール] プルダウンメニューの [オプション] を選択してオプションウィンドウを表示します。

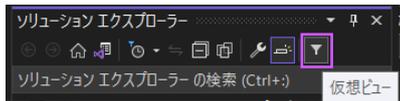
左側ツリービューの [Rocket ツール] > [指令] > [COBOL] > [ファイルの指令の確定] チェックボックスをオフにして [OK] ボタンをクリックします。



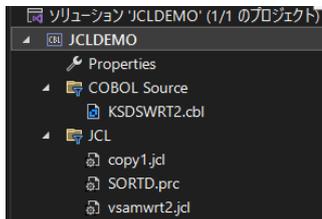
- 6) 用意した例題プログラム類をインポートします。JCLDEMO プロジェクトを右クリックして [追加] > [既存の項目] を選択し、既存項目の追加ウィンドウにて C:\¥Tutorials を指定すると内容が表示されますので、ファイル名先頭に [ACCT] が付いていない下部 4 ファイルを選択して [追加] ボタンをクリックします。この実行により、プロジェクトフォルダへ例題プログラムが配置されます。



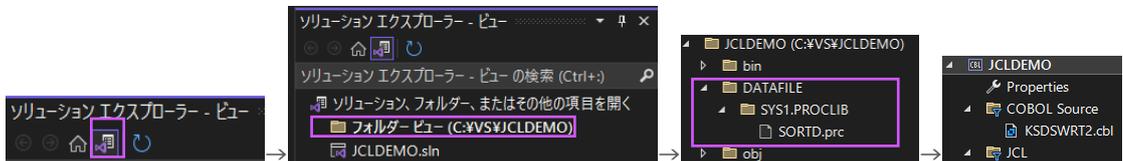
- 7) 種類別に表示するため、[ソリューション エクスプローラー] 内の [仮想ビュー] アイコンをクリックします。



- 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、ここへプロジェクトである 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] ^ [INT/GNT] を選択します。

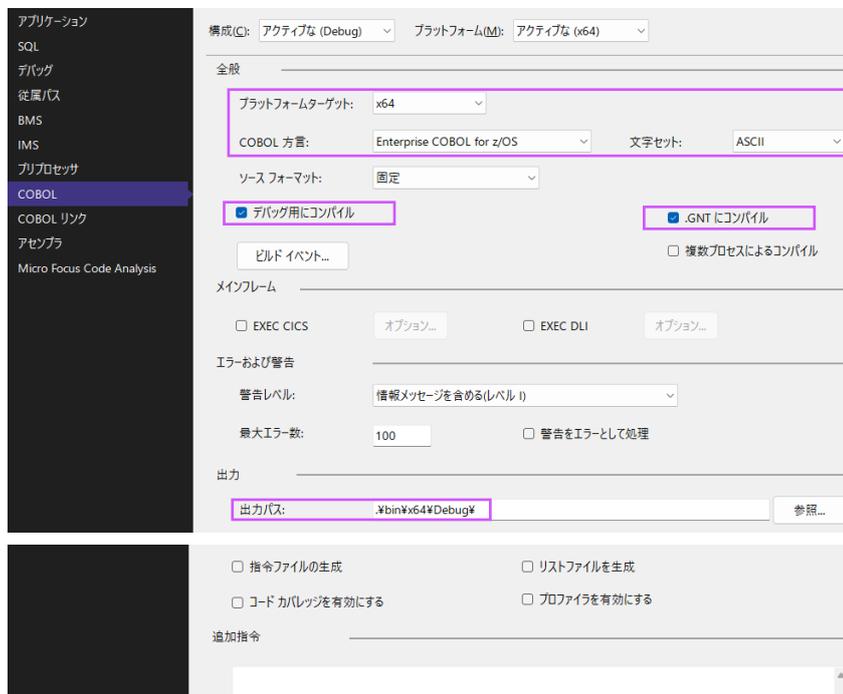


#### 情報

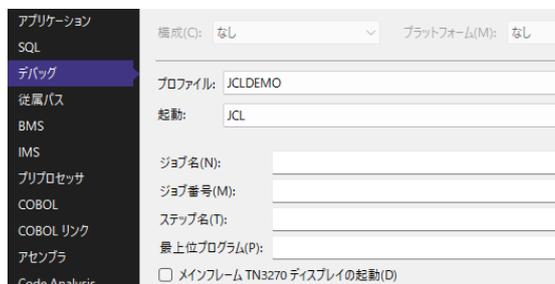
GNT は 製品独自の動的ロード実行可能ファイルを指します。

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、[メインフレーム TN3270 ディスプレイの起動] はオフに指定します。

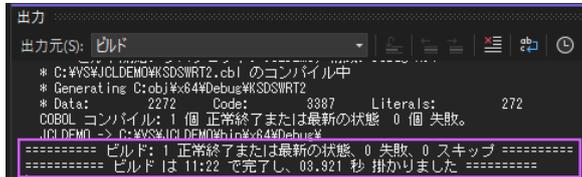


5) プロパティファイルを上書き保存します。



### 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

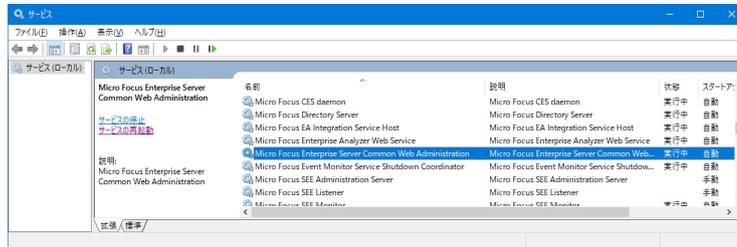
Enterprise Server インスタンスを運用、管理する Enterprise Server Common Web Administration (以降 ESCWA) では、スプールやデータ内容などに含まれる日本語を正しく表示させるために、事前に文字セットを所定のフォルダへ展開します。製品マニュアルの「リファレンス > コードセットの変換 > CCSID 変換テーブルのインストール > CCSID 変換テーブルをインストールするには」を参照しながら進めてください。

- 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。  
<https://download.boulder.ibm.com/ibmdl/pub/software/dw/java/cdctables.zip>
- 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)\Rocket Software\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。

名前	種類	サイズ
03A43480.MU-R-A2	MU-R-A2 ファイル	28 KB
03A43480.MU-R-A3	MU-R-A3 ファイル	28 KB
03A43480.MU-R-D	MU-R-D ファイル	28 KB
03A43480.PACKAGE	PACKAGE ファイル	5 KB
03A43480.TPMAP11A	TPMAP11A ファイル	329 KB
03A43480.TPMAP12A	TPMAP12A ファイル	329 KB
03A43480.TPMAP110	TPMAP110 ファイル	329 KB
03A43480.UPMAP12A	UPMAP12A ファイル	446 KB
03A43480.UPMAP13A	UPMAP13A ファイル	490 KB
03A43480.UPMAP120	UPMAP120 ファイル	447 KB
34B003A4.RPMAP12A	RPMAP12A ファイル	336 KB
34B003A4.RPMAP120	RPMAP120 ファイル	336 KB
34B003A4.UM-E-A21	UM-E-A21 ファイル	54 KB
34B003A4.UM-E-D12	UM-E-D12 ファイル	54 KB

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

Enterprise Server インスタンスには JES の互換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) ESCWA は、製品が提供する VSAM 外部セキュリティマネージャー (ESM) モジュールによるセキュリティがデフォルトで有効になっており、すべての処理において実行ユーザーの認証が行われます。

まずはデフォルトユーザーと初期パスワードを取得するために、スタートメニューから [Enterprise Developer] を選択し、[Enterprise Developer コマンドプロンプト] を起動します。



Enterprise Developer コマンドプロンプトから次のコマンドを実行して、デフォルトユーザーと初期パスワードを取得します。

```
mfsecretsadmin read microfocus/temp/admin
```

```
C:\Users\tarot\Documents>mfsecretsadmin read microfocus/temp/admin
{"mfUser": "SYSAD", "mfPassword": "CKfa+xWt"}
```

上記例の場合、SYSAD がデフォルトユーザー、CKfa+xWt が初期パスワードです。

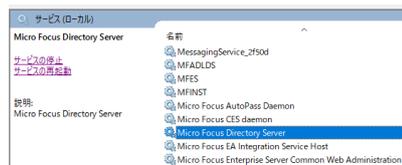
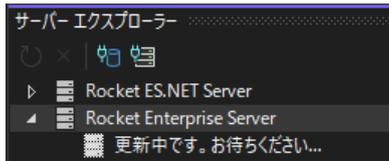
この情報は ESCWA のログオン時に使用しますので、記憶しておいてください。

デフォルトセキュリティを無効にすることもできますが、安全を確認後に実施してください。

詳しくは製品マニュアルの [ここからはじめよう] > [Getting Started] にある [デフォルトセキュリティの構成] チュートリアルをご参照ください。

- 2) Enterprise Server インスタンスを作成します。

Visual Studio の [サーバー エクスプローラー] タブの [Rocket Enterprise Server] を右クリックして [管理] を選択します。表示されていない場合は、Windows サービスの [Micro Focus Directory Server] が開始されているかを確認し、停止している場合は開始してください。



3) Web ブラウザが立ち上がり、ユーザー認証を求められます。

パスワードを変更してログオンします。

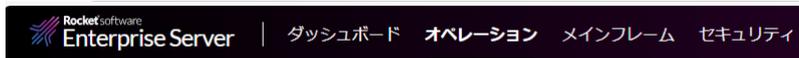
[パスワード変更] をクリックし、前項で取得したデフォルトユーザーと初期パスワード、新しいパスワードを入力して [サブミット] をクリックします。

変更後のパスワードはご自身の責任で管理してください。

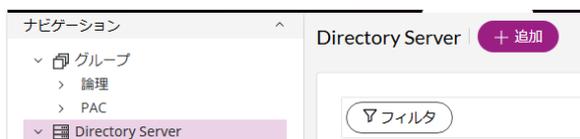
パスワード変更後、ESCWA がタイムアウトした場合は新しいパスワードを使用してログオンしてください。



7) メニューで [オペレーション] を選択後、[ナビゲーション] に [Default] という名前の接続が表示されます。表示されない場合は新規に作成します。表示されている場合はこの項目はスキップしてください。



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。



追加ウィンドウの [名前] は任意ですが、ここでは Default を、  
[ホスト] は ホスト名またはマシンの 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\*

ホスト\*

ポート\*

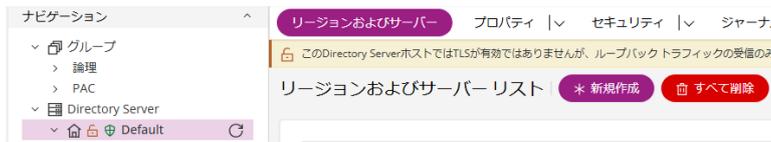
説明

\* 入力必須の項目です

保存 戻る

4) Enterprise Server インスタンスを追加します。

[Default] 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。



- 5) [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

Directory Serverとリージョンの作業モードが一致させ  
 ん。プラットフォームによっては、起動時に不具合が発生する  
 可能性があります。

名前\*

説明

64ビット作業モード

MSS有効

TN3270リスナーの作成

TN3270リスナーポート

\*入力必須の項目です 保存 戻る

**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。

- 6) 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。



- 7) 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。

一般的なプロパティ 適用 削除

開始オプション \*入力必須の項目です

名前\*

システムディレクトリ

リージョンの機能

MSS有効

JES有効

IMS有効

MQ有効

- 8) 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

- 9) [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 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]

MFACCCGI\_CHARSET=Shift\_JIS

### 追加設定

構成情報 ⓘ

[ES-Environment]  
MFACCCGI\_CHARSET=Shift\_JIS

- 10) 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。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。

JESの構成 | **適用**

JES プログラムパス ⓘ

システムカタログ ⓘ

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 ⓘ

- 11) [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 | **\* 新規作成**

- 12) 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 ⓘ

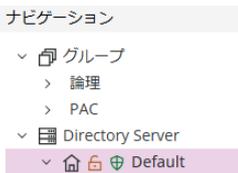
\* 入力必須の項目です

**保存** 戻る

- 13) 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。



- 14) 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。



### 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¥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 に戻り、[サーバーエクスプローラー] 内の [Rocket Enterprise Server] を展開します。ESCWA への認証ウィンドウが表示されますので、前述で指定した SYSAD ユーザーとそのパスワードを指定して [OK] ボタンをクリックします。



- 2) [サーバーエクスプローラー] 内に表示された [Default] を展開すると、再度認証ウィンドウが表示されますので、前述で指定した SYSAD ユーザーとそのパスワードを指定して [OK] ボタンをクリックします。

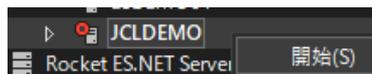


- 3) JCLDEMO インスタンスが表示されていることを確認します。表示されていない場合は [Rocket Enterprise Server] を右クリックし、[最新の情報に更新] を選択してリフレッシュしてください。

- 4) [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。



- 5) [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、[開始] を選択して JCLDEMO インスタンスを開始します。



- 6) 開始ユーザーの認証ウィンドウが表示されますので、前述で指定した SYSAD ユーザーとそのパスワードを指定して [OK] ボタンをクリックします。



- 7) 開始された JCLDEMO インスタンスを右クリックし、[コンソールログを表示] を選択して、[出力] タブの開始ログを確認します。

メッセージコードの最終桁のログレベルが I はインフォメーション、W は警告、S や E の場合はエラーです。



```
出力
出力元(S): Enterprise Server コンソールログ: 127.0.0.1/
251018 09512856 CASC001001 ES Threaded Daemon Initialized (Ver CAS 11.0.00) process-id = 11576 (09:51:26.21) 09:51:26
251018 09512856 CASC000981 ES Build Tag: ED11.0/20250723 09:51:26
251018 09512774 CASC001201 Server manager created for ES JCLDEMO, process-id = 10936 09:51:27
251018 09512812 10936 JCLDEMO CASSI00001 Server manager initialization started 09:51:28
251018 09512824 10936 JCLDEMO CASSI40001 Retrieving ES configuration from MFDS (127.0.0.1:38) 09:51:28
251018 09512857 10936 JCLDEMO CASSI00011 External security manager configured, external shutdown will require sign on 09:51:28
251018 09512855 10936 JCLDEMO CASSI90201 Region initializing in configured startup mode, 09:51:28
251018 09512865 10936 JCLDEMO CASSI90301 Monitor APIs supported on this platform, 09:51:28
251018 09512870 10936 JCLDEMO CASSI11001 Dynamic debug attachment permitted 09:51:28
251018 09512874 10936 JCLDEMO CASC00061 Setting up Configuration Store 09:51:28
251018 09512874 10936 JCLDEMO CASSI00111 Region running in 64 bit mode 09:51:28
251018 09512903 10936 JCLDEMO CASSI00111 ES support for IBM(R) CICS(R) WEBSERVICES is disabled. CCSID tables 00000 to Unicode(UCS) not found 09:51:29
251018 09512921 10936 JCLDEMO CASSI17401 Processing resources in startup list DFH$IVP 09:51:29
251018 09512922 10936 JCLDEMO CASSI00001 Region will use SYSID $I$VP, local CCSID 00001 and ascii CCSID 00001 09:51:29
251018 09512934 10936 JCLDEMO CASSI01111 Enterprise Server time keeper daemon created for Server JCLDEMO, process-id = 12136 09:51:29
251018 09512940 CASC010901 ES TRC Service Process created for Server JCLDEMO, process-id = 6036 09:51:29
251018 09512948 CASC010701 ES TSC Service Process created for Server JCLDEMO, process-id = 8180 09:51:29
251018 09512956 CASC010301 ES Communications Server created, ES JCLDEMO, process-id = 10808 09:51:29
251018 09512956 12136 JCLDEMO CASC06021 Enterprise Server time keeper daemon initialized 09:51:29
251018 09512958 6036 JCLDEMO CASC10011 ES TRC Service Process initialization started 09:51:29
251018 09512958 10936 JCLDEMO CASC10001 ES concurrent request limit: 0000000010 09:51:29
251018 09512965 10936 JCLDEMO CASSI10001 Server Manager initialization completed successfully 09:51:29
251018 09512974 10936 JCLDEMO CASSI40001 Single signon is not enforced for IBM(R) CICS(R) clients and is not enforced for IMS TM clients for this startup 09:51:29
251018 09512989 10936 JCLDEMO CASC00001 ES TRC Service Process initialization started 09:51:29
```

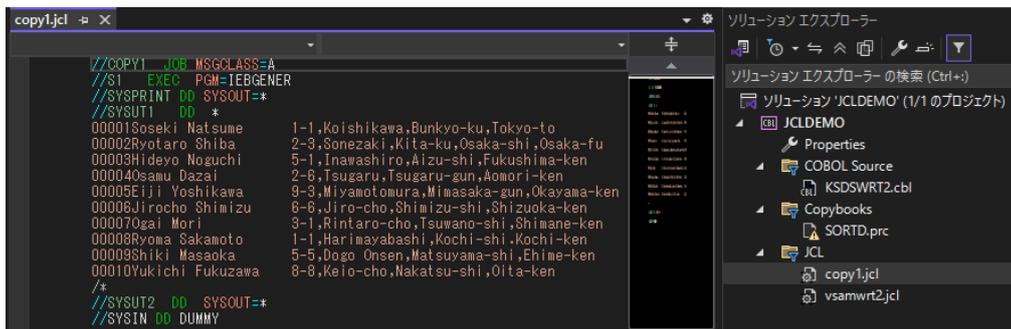
### 注意

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。

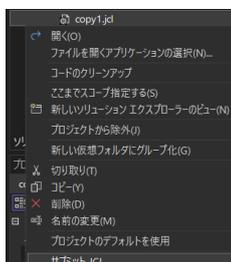
## 3.9 JCLの実行

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。

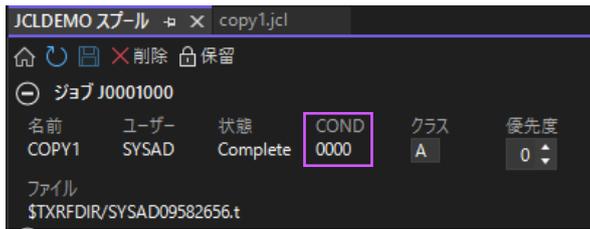
- 1) [ソリューション エクスプローラー] 内にある JCLDEMO プロジェクト配下の copy1.jcl をダブルクリックし、エディタで内容を確認します。この JCL は IEBGENER ユーティリティを使用して、JCL 内に書かれたインラインデータを SYSOUT に書き出しています。



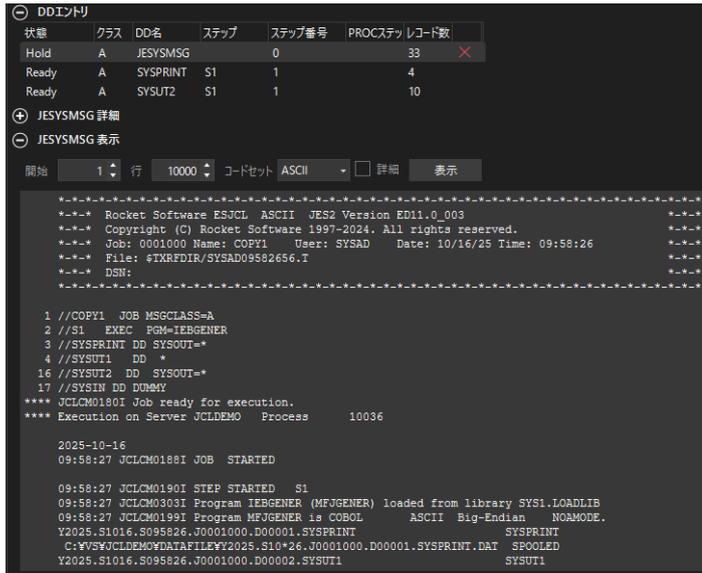
- 2) [ソリューション エクスプローラー] 内の copy1.jcl を右クリックして [サブミット JCL] を選択すると、この JCL が実行されます。



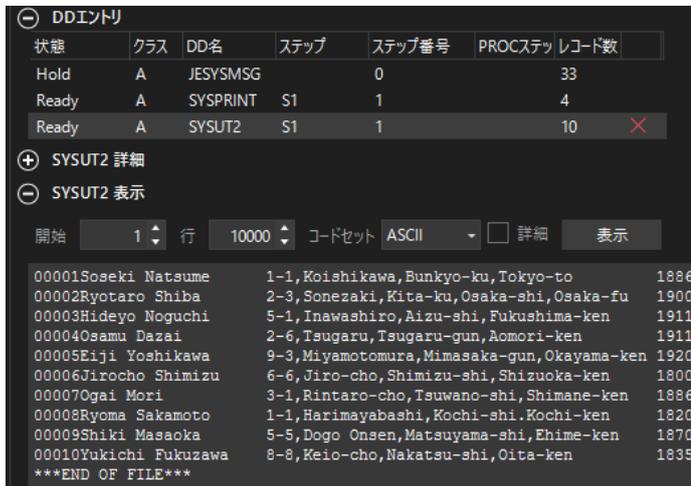
3) [JCLDEMO スプール] タブが表示されます。[COND] で正常終了したことが確認できます。



4) [JCLDEMO スプール] タブの [DDエントリ] で JESYSMSG を選択すると、[JESYSMSG 表示] に内容が表示されます。他のスプールファイルも同様に、選択すると内容が表示されます。

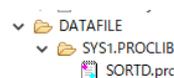


JCL のインラインデータが正常に出力されました。



### 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。

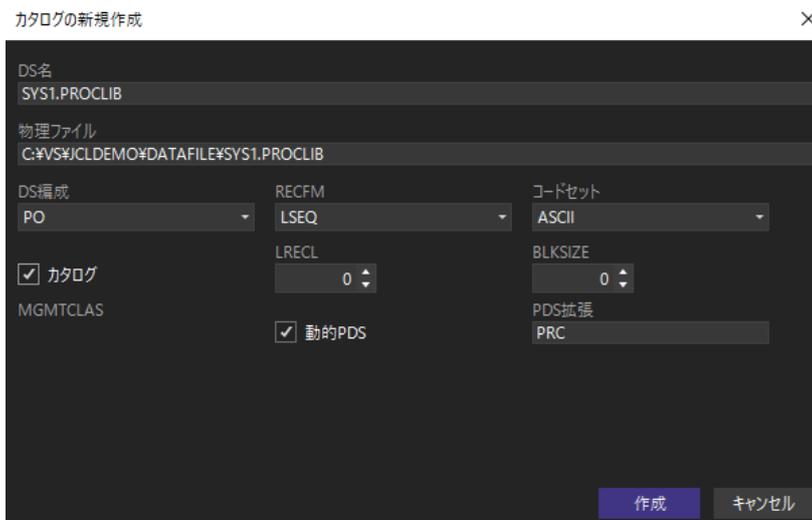


- 1) [サーバーエクスプローラー] 内の JCLDEMO 配下にある [Catalog] を右クリックし、[新規作成] を選択します。



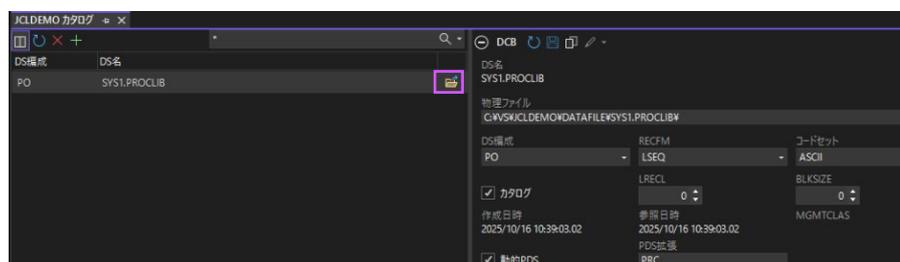
- 2) カタログの新規作成ウィンドウが表示されますので、以下のように入力し [作成] ボタンをクリックします。

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	チェックします。

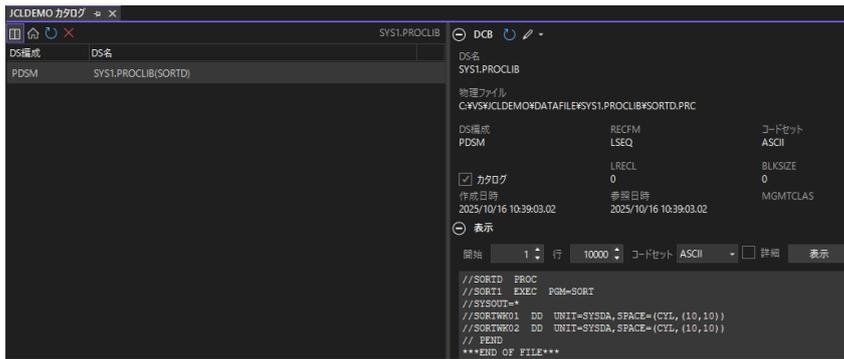


- 3) [サーバーエクスプローラー] 内の JCLDEMO 配下にある [Catalog] を右クリックし、[カタログを表示] を選択すると、現在登録されているカタログ一覧とその内容が表示されます。

[DS 構成] 行の右端にあるリストアイコンをクリックすると配置されているメンバーとその内容を確認することができます。



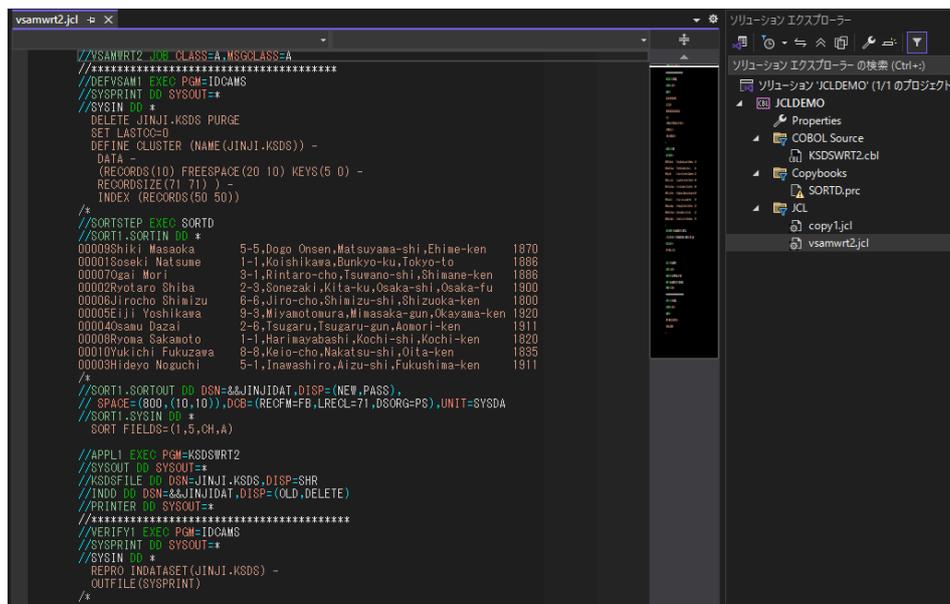
メンバーと内容が表示されます。



### 3.11 COBOL バッチプログラムの実行

COBOL プログラムを呼び出す JCL を実行します。

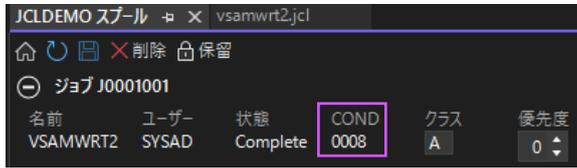
- 1) [ソリューションエクスプローラー] 内にある JCLDEMO プロジェクト配下の vsamwrt2.jcl をダブルクリックし、エディタで内容を確認します。



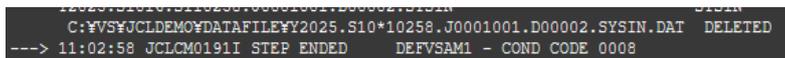
- ① ステップ 1:DEFVSAM1  
IDCAMS を使用して VSAM データセットの JINJI.KSDS を削除し、クラスターを持つ KSDS として再作成します。
- ② ステップ 2:SORTSTEP  
前項で登録した SORTD.prc を使用して、JINJI.KSDS ファイルへの書き込み用データをソートします。
- ③ ステップ 3:APPL1  
アプリケーション KSDSWRT2 を呼び出しステップ 2 でソートされたデータをステップ 1 で定義した VSAM クラスターに書き込みを行います。同時に書き込まれたデータを DD=PRINTER へ出力します。
- ④ 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、この JCL を実行します。

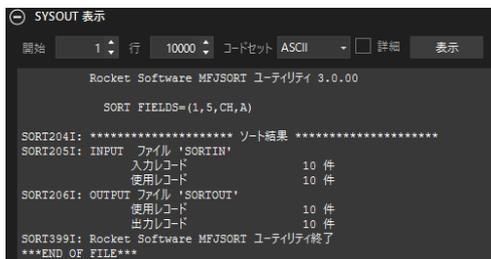
3) [JCLDEMO スプール] タブが表示されます。初回実行時はステップ 1 で削除する該当ファイルが見つからないため [COND] は 0008 となりますが、問題はありません。



4) [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。

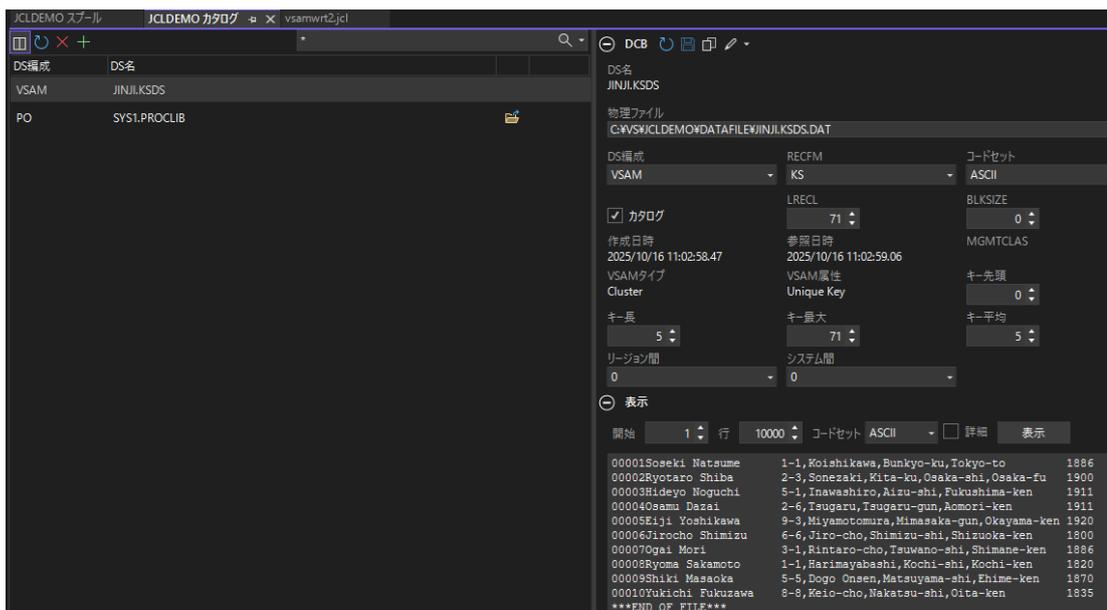


5) SORTSTEP の [SYSOUT] をクリックしてソート内容を確認します。



6) 他のスプールファイルの内容も確認してみてください。

7) 前項と同様の手順で、この JCL によってカタログされた情報を確認します。カタログ一覧では、VSAM ファイルの JINJI.KSDS が新たにカタログされていることが確認でき、その DCB 情報と内容も表示されています。



### 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。

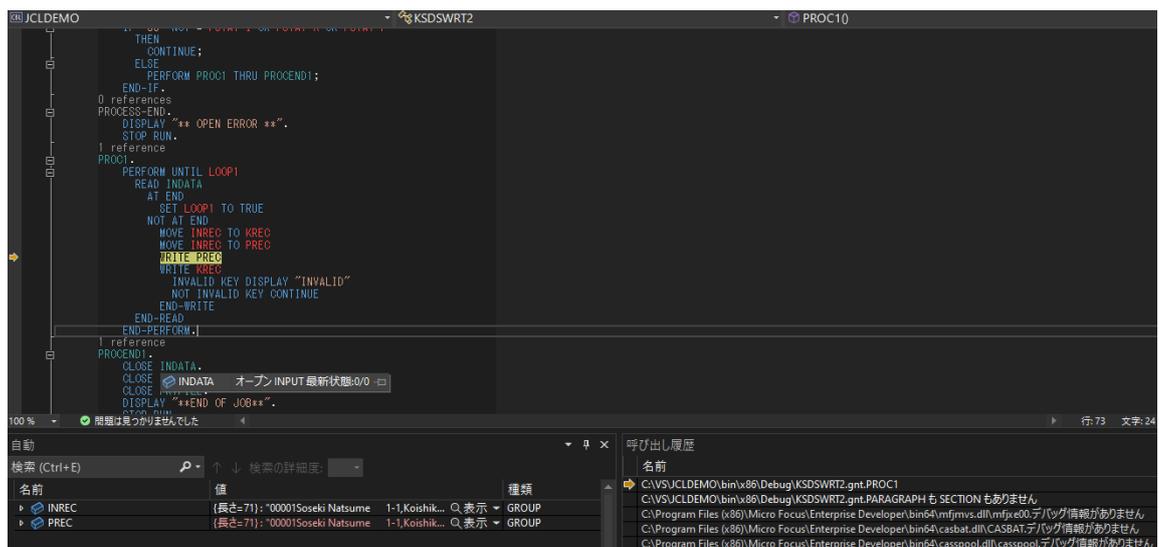
- 1) [デバッグ] プルダウンメニューの [ステップ イン] を選択します。



- 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。



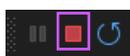
- 5) 希望のステップの左端をクリックすることにより、ブレークポイントを設定することも可能です。



- 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。

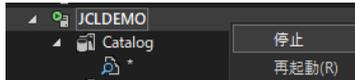


- 8) 画面左下が元の色の [準備完了] となり、アタッチ待機状態ではなくなったことを確認します。



### 3.13 Enterprise Server インスタンスの停止

- 1) [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、[停止] を選択して 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。

## 5. 免責事項

本チュートリアル of 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。