

# Enterprise Developer チュートリアル

## メインフレーム COBOL 開発：コードカバレッジの実施 Eclipse 編

### 1. 目的

本チュートリアルは、JCL から実行された COBOL ソースを例としたコードカバレッジの取得や結果表示を行う手順とその方法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 Enterprise Developer 11J for Eclipse がインストールされていること
- JCL チュートリアルのプログラムと Enterprise Server インスタンスを使用するため、JCL チュートリアルが実施済であること

**補足)** 完了していない場合は、先に JCL チュートリアルを実施してください。

### 3. チュートリアル手順の概要

1. Eclipse の起動
2. プロジェクトプロパティの設定
3. IDE を使用した JCL 実行
4. コードカバレッジの表示
5. IDE を利用しない JCL 実行

### 4. 免責事項

### 3.1 Eclipse の起動

- Enterprise Developer for Eclipse を起動します。



- JCL チュートリアルで作成した C:¥work をワークスペースへ指定して、[OK] ボタンをクリックします。

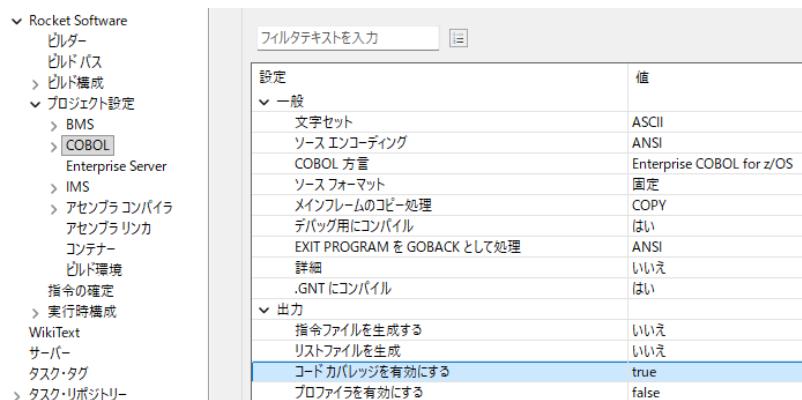


### 3.2 プロジェクトプロパティの設定

プロジェクトのプロパティを設定します。

- COBOL エクスプローラー内の JCLDEMO プロジェクトを右クリックして [プロパティ] を選択します。
- 左側メニューの [Rocket Software] > [プロジェクト設定] > [COBOL] を選択して、[コード カバレッジを有効にする] に true を指定後 [適用して閉じる] ボタンをクリックしてください。

保存後は再度プロジェクトのビルドを実施してください。



#### 補足)

個別ソースファイルへ指定したい場合は対象ソースファイルを右クリックし、左側メニューの [COBOL] から表示される画面で [ファイルの固有な設定を可能にする] チェックをオンにした後、コードカバレッジを有効にします。指定後は [適用して閉じる] ボタンをクリックしてください。



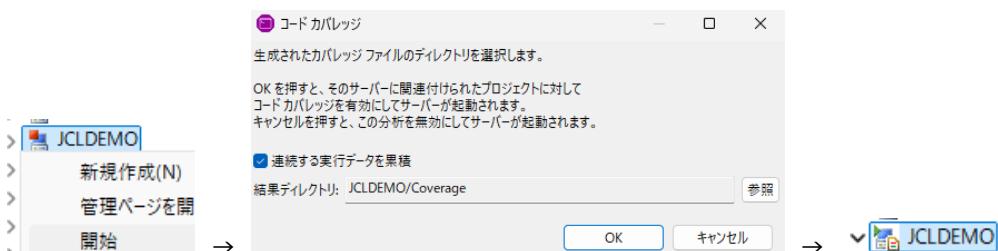
### 3.3 IDE を使用した JCL 実行

- 1) JCLDEMO プロジェクトに関連付けられている、JCL チュートリアルで作成した JCLDEMO インスタンスへコードカバレッジ分析を指定します。

Eclipse の [サーバーエクスプローラー] > [JCLDEMO] を右クリックし、[分析を設定] > [コード カバレッジ] を選択して有効にします。



- 2) 再度 [サーバーエクスプローラー] > [JCLDEMO] を右クリックし [開始] を選択するとログファイルと結果ファイルの出力パスを指定するウィンドウが表示されますので、[連続する実行データを累積] にチェックし、パスを確認後 [OK] ボタンをクリックします。

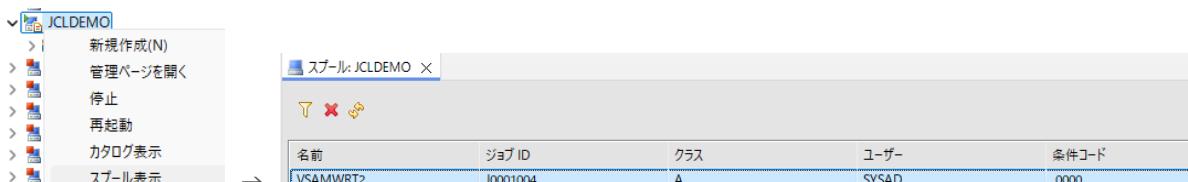


[コンソール] タブに以下のように表示されます。

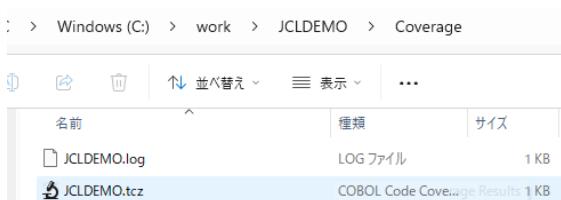
サーバー: JCLDEMO 正常に起動されました - コード カバレッジ有効

- 3) COBOL エクスプローラー内の vsamwrt2.jcl を右クリックして [Enterprise Server へのサブミット] を選択し、この JCL を実行します。

- 4) [サーバーエクスプローラー] > [JCLDEMO] を右クリックし [スプール表示] を選択して JCL の正常終了を確認します。



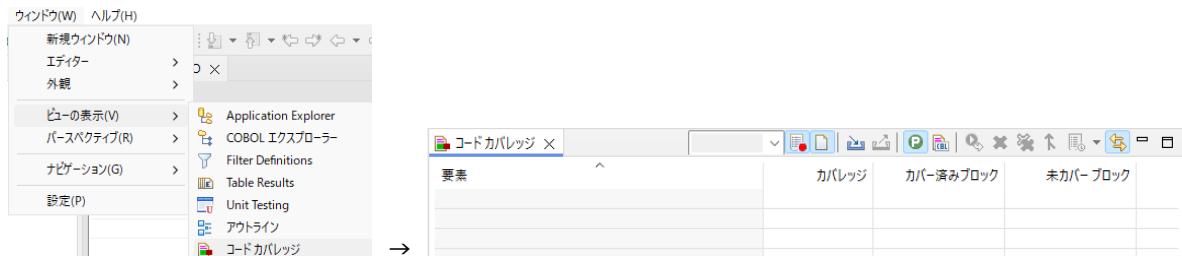
- 5) Windows エクスプローラーから、前項の出力パスにコードカバレッジ用のフォルダとファイルが作成されていることを確認します。



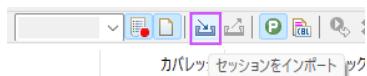
### 3.4 コードカバレッジの表示

Eclipse に戻り、下記の操作を行います。

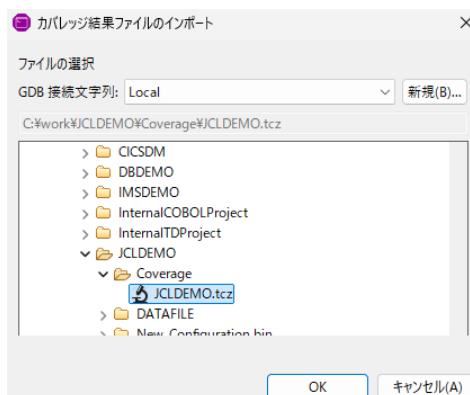
- 1) [ウィンドウ] プルダウンメニューから [ビューの表示] > [コード カバレッジ] を選択してコードカバレッジビューを表示します。



- 2) [コードカバレッジ] タブの [セッションをインポート] アイコンをクリックしてファイル選択ウィンドウを表示します。



- 3) [ドライブ] 配下から前項で生成した JCLDEMO.tcz ファイルを選択し、[OK] ボタンをクリックします。



- 4) 実行されたコードのカバレッジ率が表示されます。[PROC1] をダブルクリックしてみます。

要素	カバレッジ	カバー済みブロック	未カバー ブロック
JCLDEMO	72.7 %	8	3
KSDSWRT2	72.7 %	8	3
PROC1	83.3 %	5	1
PROCEND1	100.0 %	1	0
PROCESS-END	0.0 %	0	1
Procedure Division	72.7 %	8	3

- 5) 該当箇所が表示され、通過したコードは緑で表示され、見通過のコードは赤で表示されます。

```

PROCEDURE DIVISION.
  OPEN INPUT INDATA.
  OPEN OUTPUT INDEXFILE.
  OPEN OUTPUT PRTFILE.
  IF "00" NOT = FSTAT-I OR FSTAT-K OR FSTAT-P
    THEN
      CONTINUE;
    ELSE
      PERFORM PROC1 THRU PROCEND1;

  END-IF.
PROCESS-END.
  DISPLAY "*** OPEN ERROR ***".
  STOP RUN.
PROC1.
  PERFORM UNTIL LOOP1
    READ INDATA
    AT END
      SET LOOP1 TO TRUE
    NOT AT END
      MOVE INREC TO KREC
      MOVE INREC TO PREC
      WRITE PREC
      WRITE KREC
      INVALID KEY DISPLAY "INVALID"
      NOT INVALID KEY CONTINUE
    END-WRITE
  END-READ
END-PERFORM.
PROCEND1.
  CLOSE INDATA.
  CLOSE INDEXFILE.

```

- 6) カバレッジが 0% の [PROCESS-END] を見ると赤で表示されており、今回の実行ではカバーされていないことがわかります。

```

PROCESS-END.
  DISPLAY "*** OPEN ERROR ***".
  STOP RUN.
PROC1.

```

- 7) [PROCESS-END] を通すため、INDATA ファイルをオープン時にエラーとなるよう意図的に操作して JCL を再実行します。

- 8) 再度、JCLDEMO.tcz ファイル取り込むと累積された結果が表示されます。



- 9) [PROCESS-END] を通過し、ソースコードも緑になりました。

```

PROCEDURE DIVISION.
  OPEN INPUT INDATA.
  OPEN OUTPUT INDEXFILE.
  OPEN OUTPUT PRTFILE.
  IF "00" NOT = FSTAT-I OR FSTAT-K OR FSTAT-P
    THEN
      CONTINUE;
    ELSE
      PERFORM PROC1 THRU PROCEND1;

  END-IF.
PROCESS-END.
  DISPLAY "*** OPEN ERROR ***".
  STOP RUN.

```

- 10) カバレッジ出力フォルダにはログが生成されますので、こちらでも結果を確認することができます。

```

=====
"C:\work\JCLDEMO\Coverage\JCLDEMO.tcz"           Run: 1 22 Dec 25 15:37
-----
          Basic Blocks
Program    Calls Total done left
-----
KSDSWRT2      1    11    8    3  72.72%
All programs  11
=====
"C:\work\JCLDEMO\Coverage\JCLDEMO.tcz"           Run: 2 22 Dec 25 15:37
-----
          Basic Blocks          Cumulative Result
Program    Calls Total done left      calls done left
-----
KSDSWRT2      1    11    3    8  27.27%      2    10    1  90.90%
All programs  11

```

- 11) JCLDEMO インスタンスを停止します。

### 3.5 IDE を利用しない JCL 実行

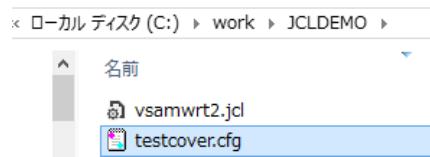
次に、コマンドラインから JCL を実行する際のコードカバレッジを実施します。

- 1) 結果の出力先を指定するコードカバレッジ用の構成ファイルを作成します。

下記内容をテキストファイルへコピーして JCLDEMO フォルダ配下へ保存します。ここではファイル名を testcover.cfg とします。RESULT 文字に続く jclcover.tcz ファイルが実行により出力される結果ファイルです。

#### 【構成ファイル内容】

```
[TESTCOVER]
RESULT C:\work\JCLDEMO\jclcover.tcz ACCUMULATE
ECHOLOG NO
LOGNAME C:\work\JCLDEMO\testcover.log
```

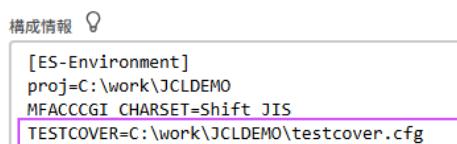


- 2) ESCWA から [JCLDEMO] インスタンスの [編集] アイコンをクリックして設定画面を表示します。

- 3) [構成情報] へ環境変数を設定し、構成ファイル名までのパスを指定します。

TESTCOVER=C:\work\JCLDEMO\testcover.cfg

#### 追加設定



構成情報

[ES-Environment]  
proj=C:\work\JCLDEMO  
MFACCGI CHARSET=Shift\_JIS  
TESTCOVER=C:\work\JCLDEMO\testcover.cfg

- 4) Windows のアプリケーションメニューから [Enterprise Developer] > [Enterprise Developer コマンドプロンプト(64-bit)] を右クリックし [管理者として実行] を選択します。



- 5) [JCLDEMO] インスタンスを開始するコマンドを、権限のあるユーザーを指定して実行します。

コマンド)casstart /rJCLDEMO /uSYSAD /pSYSAD

```
C:¥Users¥tarot¥Documents>casstart /rJCLDEMO /uSYSAD /pSYSAD
CASCD0167I ES Daemon successfully auto-started 16:16:23
CASCD1005I C:¥work¥JCLDEMO¥system¥console.log 16:16:23
CASCD0050I ES "JCLDEMO" initiation is starting 16:16:23
```

- 6) C:¥work¥JCLDEMO フォルダに存在する JCL を、権限のあるユーザーでコマンドから実行します。

コマンド)cassub /rJCLDEMO /jc:¥work¥jcldemo¥vsamwrt2.jcl /uSYSAD /pSYSAD

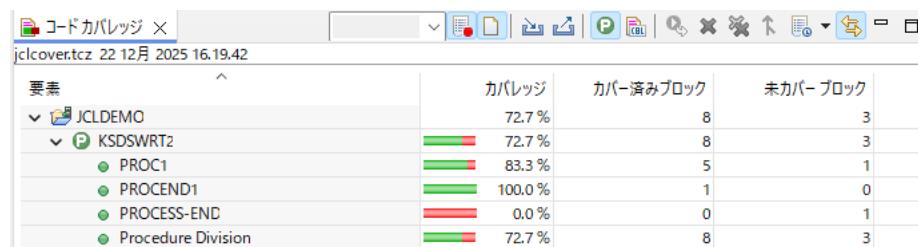
```
C:¥Users¥tarot¥Documents>cassub /rJCLDEMO /jc:¥work¥jcldemo¥vsamwrt2.jcl /uSYSAD /pSYSAD
JCLCM0187I J0001013 VSAMWRT2 JOB SUBMITTED (JOBNAME=VSAMWRT2,JOBNUM=0001013) 16:19:42
JCLCM0180I J0001013 VSAMWRT2 Job ready for execution. 16:19:42
Processed "c:¥work¥jcldemo¥vsamwrt2.jcl"
```

- 7) Windows エクスプローラーから指定パスにコードカバレッジ用のフォルダとファイルが作成されていることを確認します。

ディスク (C:) > work > JCLDEMO >

名前  
 jclcover.tcz  
 testcover.log

- 8) Eclipse の [コードカバレッジ] タブの [セッションをインポート] アイコンをクリックして生成されたファイル内容を確認すると前項と同じ結果であることがわかります。



- 9) 出力された testcover.log ファイルをテキストエディターで表示すると、ブロック単位のカバレッジ結果を確認できます。

Program	Calls	Total	done	left	Basic Blocks
KSDSWRT2	1	11	8	3	72.72%
All programs		11			

- 10) [JCLDEMO] インスタンスを停止するコマンドを、権限のあるユーザーを指定して実行します。

コマンド)casstop /rJCLDEMO /uSYSAD /pSYSAD

```
C:¥Users¥tarot¥Documents>casstop /rJCLDEMO /uSYSAD /pSYSAD
CASST0005I Shutdown of ES JCLDEMO starting 16:24:22
Return code: 0
```

#### 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。