

Visual COBOL チュートリアル

COBOL 開発 : Eclipse – コードカバレッジ機能

目的

本チュートリアルは、ネイティブ COBOL プログラムに対するコードカバレッジを表示させる手順の習得を目的としています。コードカバレッジ機能は、COBOL 開発作業に以下の利点を提供します。

- テスト未実施箇所の検出
追加テストを適切に行う事でプログラム品質の向上が見込まれます
- テスト終了後のカバレッジ率算出
品質指標値を満たしているかの確認ができ、その証左として利用できます

なお、コードカバレッジ機能は ネイティブ COBOL でのみ利用可能です。また、JVM COBOL プロジェクトに追加されたネイティブ COBOL プログラムへの利用もできません。

1. 前提

- 本チュートリアルで使用したマシン OS : Windows 11
- Visual COBOL 9.0J for Eclipse がインストール済みであること

下記のリンクから事前にチュートリアル用のサンプルファイルをダウンロードして、任意のフォルダに解凍しておいてください。

[サンプルプログラムのダウンロード](#)

内容

目的

1. 前提
2. チュートリアル手順の概要
 - 2.1. IDE からの実行
 - 2.1.1. Eclipse の起動
 - 2.1.2. チュートリアルプロジェクトのインポート
 - 2.1.3. カバレッジを有効にしたアプリケーションの実行
 - 2.1.4. 複数のカバレッジ情報のマージ
 - 2.1.5. 既存カバレッジ情報ファイルの読み込み
 - 2.2. コマンドラインからの実行
 - 2.2.1. コマンドラインからのコードカバレッジ機能の利用
 - 2.2.2. ユーティリティ機能の紹介

2. チュートリアル手順の概要

2.1. IDE からの実行

2.1.1. Eclipse の起動

- 1) スタートメニューより、Visual COBOL for Eclipse を起動します。
- 2) ワークスペースを指定し、[起動(L)] ボタンをクリックします。

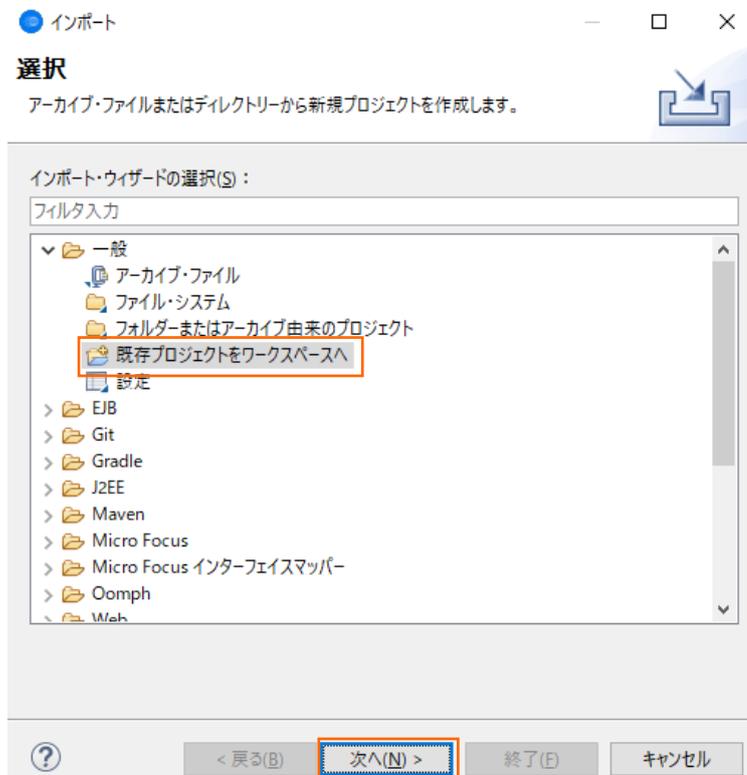


2.1.2. チュートリアルプロジェクトのインポート

1) Eclipse IDE メニューの [ファイル(F)] > [インポート(I)] を選択します。



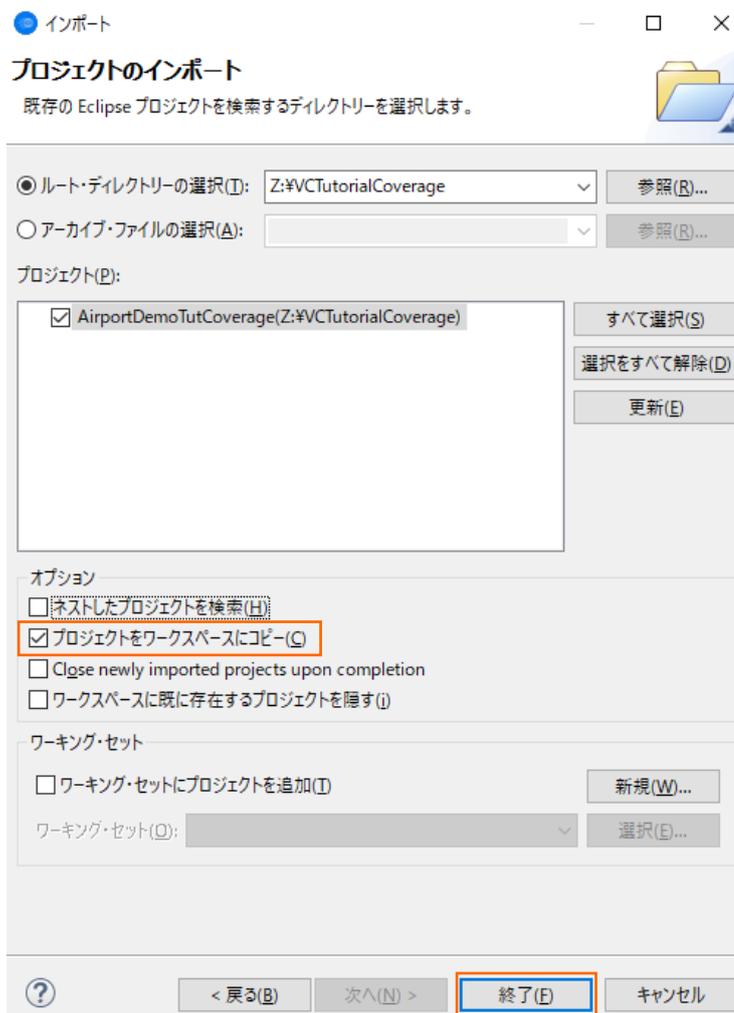
2) [一般] > [既存プロジェクトをワークスペースへ] を選択し、[次へ(N) >] ボタンをクリックします。



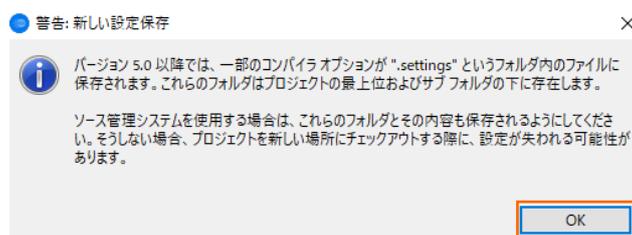
- 3) 「ルート・ディレクトリーの選択(T)」欄に、チュートリアルプロジェクトへのパスを指定します。[参照(R)] ボタンを押してダウンロードしたサンプルファイルを展開したフォルダを指定してください。



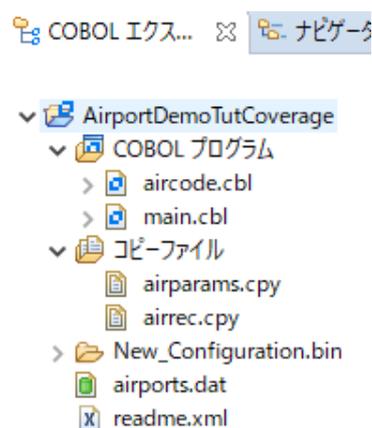
- 4) 「プロジェクトをワークスペースにコピー(C)」にチェックを入れたうえで、[終了(F)] ボタンをクリックします。



以下のダイアログが表示された場合、そのまま [OK] ボタンをクリックします。



AirportDemoTutCoverage プロジェクトが作成されることを確認します。



補足)

COBOL 開発を行うためには、COBOL パースペクティブという画面レイアウトを使用します。異なるパースペクティブを開いている場合、Eclipse IDE メニューの [ウィンドウ(W)] > [パースペクティブ(R)] > [パースペクティブを開く(O)] > [その他(O)] をクリックした上で、COBOL をクリックすることで、COBOL パースペクティブを開くことができます。

- 5) COBOL エクスプローラービュー上で AirportDemoTutCoverage プロジェクトを選択した状態で、マウスの右クリックにてコンテキストメニューを表示し、[プロパティ(R)] をクリックします。



- 6) 画面左部のメニューより、[Micro Focus] > [ビルド構成] > [COBOL] を選択します。その後、「コードカバレッジを有効にする」を “true” に変更します。

Micro Focus
ビルダー
ビルドパス
ビルド構成
COBOL
イベント
ディプロイ
ビルド環境
リンク
プロジェクト設定
指令の確定
実行時構成
サーバー
タスク・タグ
ビルダー
プロジェクト・ネーチャ
プロジェクト・ファセット
プロジェクト参照
検証
実行/デバッグ設定

構成の固有な設定を可能にする(C)

上位レベルの設定を上書きする (マージしない)

フィルタテキストを入力

設定	値
一般	
文字セット	ASCII
COBOL 方言	Micro Focus
ソース フォーマット	固定
デバッグ用にコンパイル	はい
EXIT PROGRAM を GOBACK として処理	いいえ
詳細	いいえ
.GNT にコンパイル	いいえ
出力	
指令ファイルを生成する	いいえ
リストファイルを生成	いいえ
コード カバレッジを有効にする	true
プロファイルを有効にする	

- 7) 次に画面左部のメニューより、[Micro Focus] > [ビルド構成] > [リンク] を選択します。エントリポイントに “main” と入力し、[適用して閉じる] ボタンをクリックします。

Micro Focus
ビルダー
ビルドパス
ビルド構成
COBOL
イベント
ディプロイ
ビルド環境
リンク
プロジェクト設定
指令の確定
実行時構成
サーバー
タスク・タグ
ビルダー
プロジェクト・ネーチャ
プロジェクト・ファセット
プロジェクト参照
検証
実行/デバッグ設定

フィルタテキストを入力

設定	値
Linkage	
出力の名前	AirportDemo
出力パス	New_Configuration.bin
エントリポイント	main
ターゲットの種類	単一実行可能ファイル
ビット数	32 ビット
.LBR にパッケージ化	いいえ
COBOL 以外のアプリケーションから呼び出し可能	いいえ
サービスを COBOL アーカイブ (.car) ファイルにパッケージ化	いいえ
マルチスレッド	はい
実行時モデル	共有
現在の実行時システムだけにバインドする	いいえ
出力の種類	コンソール
ターゲット オペレーティング システム	Windows
詳細	いいえ
cpp ライブラリを含める	いいえ
未定義シンボルでエラー	いいえ
エントリポイント アドレスを読み込む	いいえ
Windows リソースファイル	
追加指令	
カスタム コマンド	

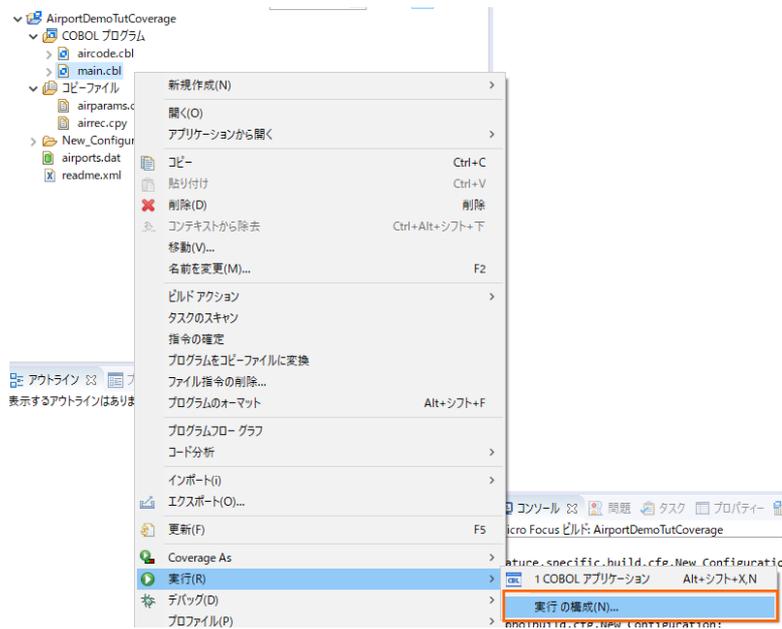
エントリポイント
ビルドファイル内のデフォルトのエントリポイントを指定します

デフォルトの復元(T) 適用(L)

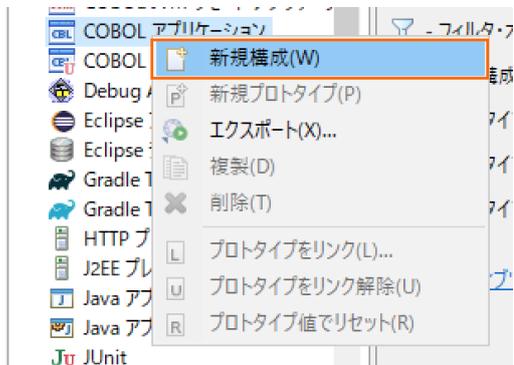
適用して閉じる キャンセル

2.1.3. カバレッジを有効にしたアプリケーションの実行

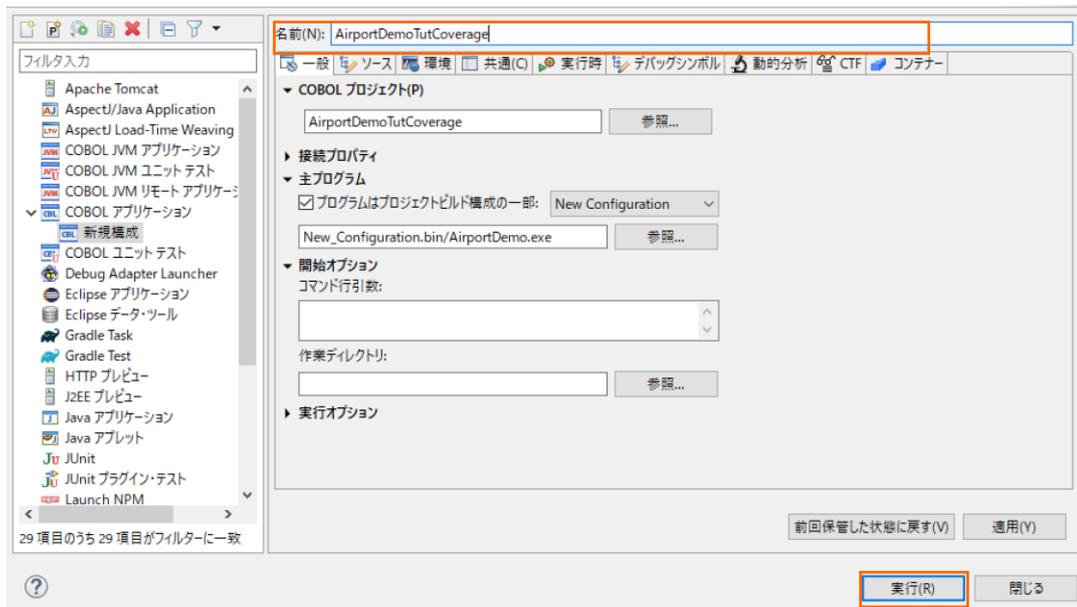
- 1) AirportDemoTutCoverage プロジェクト内の main.cbl を選択した上で、マウスの右クリックにてコンテキストメニューを表示し、[実行(R)] > [実行の構成(N)] を選択します。



- 2) 「COBOL アプリケーション」を選択した状態で、マウスの右クリックにてコンテキストメニューを表示し、[新規構成] を選択します。



- 3) 「名前(N)」欄に "AirportDemoTutCoverage" を入力した上で、[実行(R)] ボタンをクリックします。



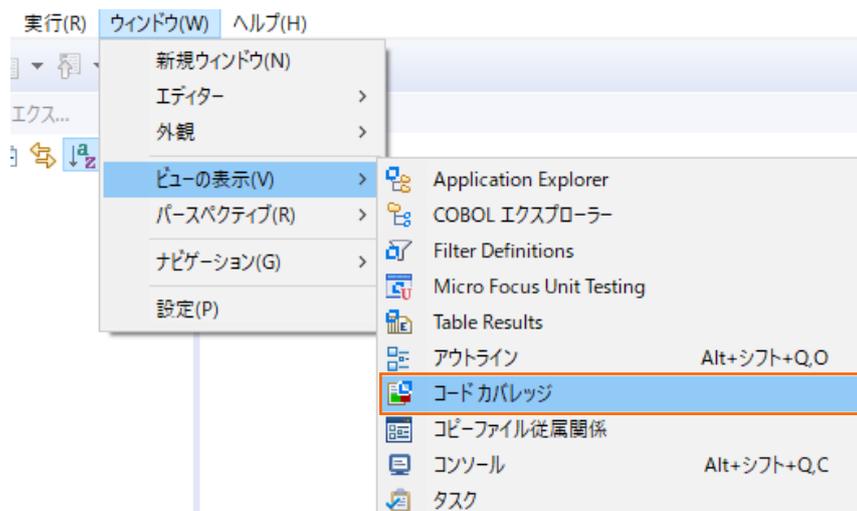
- 4) アプリケーション画面上に以下の入力を行います。
- 1 回目 : “HND△LHR” を入力して Enter (羽田・ロンドンヒースロー空港間の距離を表示)
 - 2 回目 : 何も入力せず Enter (プログラムの終了)

```

AirportDemo
Enter an airport code, or two codes separated by space, or no code to exit:

```

- 5) Eclipse IDE メニューより、[ウィンドウ(W)] > [ビューの表示(V)] > [コードカバレッジ] を選択します。

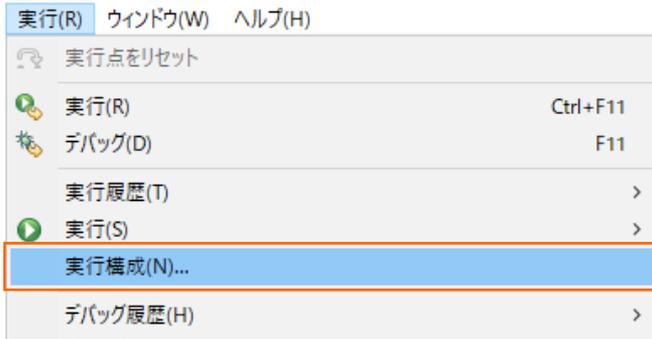


現時点では、カバレッジ情報が何も表示されていないことを確認します。

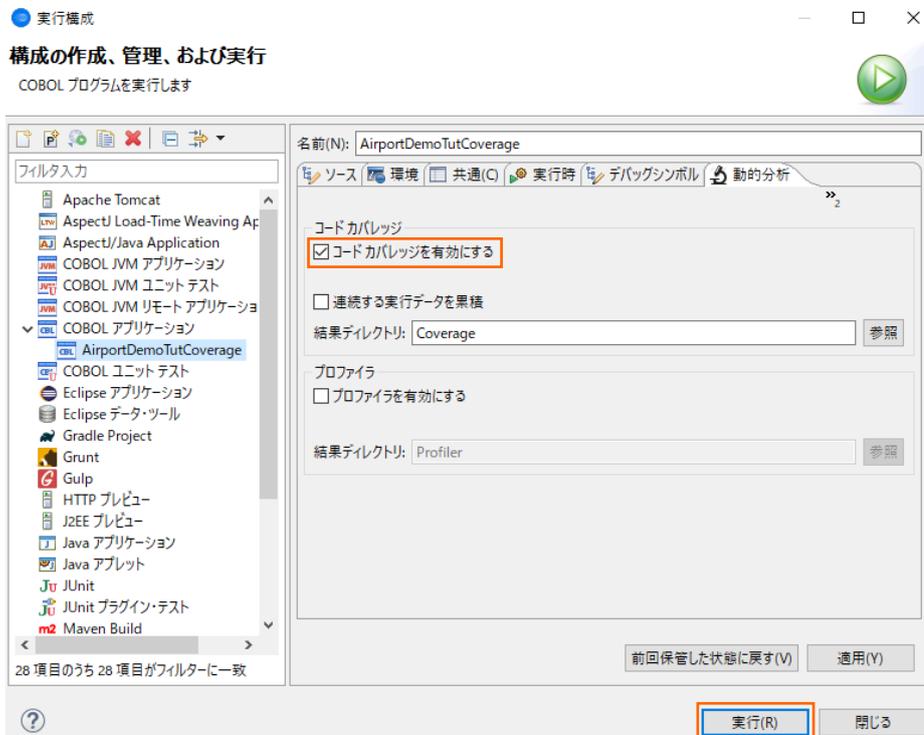
要素	カバレッジ	カバー済みブロック	未カバーブロック	ブロック数

コードカバレッジ機能を有効化するためには、アプリケーション実行時にも指示が必要となります。

- 6) Eclipse IDE メニューより、[実行(R)] > [実行構成(N)] を選択します。



- 7) AirportDemoTutCoverage の実行構成が選択されている状態で、「動的分析」タブを選択し、「コードカバレッジを有効にする」をチェックします。その後、[実行(R)] ボタンをクリックします。



- 8) 再度、アプリケーション画面が起動されるため、前回同様、下記入力を行ってください。
- 1 回目： “HND△LHR” を入力して Enter（羽田・ロンドンヒースロー空港間の距離を表示）
 - 2 回目： 何も入力せず Enter（プログラムの終了）
- 補足) △は半角スペースを表します。

実行後、コードカバレッジビューに、カバレッジ情報が表示されることを確認します。

要素	カバレッジ	カバー済みブロック	未カバーブロック	ブロック数
AirportDemoTutCoverage	67.9 %	38	18	56
aircode	65.9 %	29	15	44
calculate-airport-distance	100.0 %	5	0	5
close-airfile	100.0 %	1	0	1
convert-angle	85.7 %	6	1	7
display-airport	100.0 %	1	0	1
distance-between-airports	100.0 %	8	0	8
find-airport	66.7 %	2	1	3
get-code-matches	0.0 %	0	6	6
lookup-one-airport	0.0 %	0	3	3

- 9) コードカバレッジビュー上から「lookup-one-airport」をダブルクリックすることで、エディターが開き、カバレッジ結果を表示します。

```

.....*A.1.B.....2.....3.....4.....5.....6.....
        perform open-airfile
        when close-file
        perform close-airfile
        when display-record
        perform display-airport
    end-evaluate
    exit program
.
lookup-one-airport section.
    initialize lnk-rec
    move lnk-airport1 to airport
    perform find-airport
    if airport-found = 1
        move f-rec to lnk-rec
    end-if
.
distance-between-airports section.
*> need to find both airports and distance between them
    initialize lnk-distance-result
    move lnk-airport1 to airport
    perform find-airport
    if airport-found = 1
        perform display-airport
        move f-rec to a1-rec

        move lnk-airport2 to airport
        perform find-airport
        if airport-found = 1

```

背景色が赤色の箇所が未実行箇所、緑色の箇所が実行済み箇所となります。

2.1.4. 複数のカバレッジ情報のマージ

一般的に、1回のテスト実行で品質指標として十分なカバレッジ情報を得ることはできません。本節では、複数回のアプリケーション実行によるカバレッジ情報をマージする方法を学びます。

- 再度、AirportDemoTutCoverage アプリケーションを実行し、以下の入力を行ってください。
 - 1回目: "HND" と入力した後で Enter (羽田空港の情報を表示)
 - 2回目: 何も入力せず Enter (プログラムの終了)

```

C:\workspace_tut_coverage\AirportDemoTutCoverage\New_Configuration.bin\AirportDemo.exe
Enter an airport code, or two codes separated by space, or no code to exit:
HND
HND Tokyo Intl
Japan Lat:+035.552258 Lon:+139.077969
Enter an airport code, or two codes separated by space, or no code to exit:
続行するには何かキーを押してください . . .

```

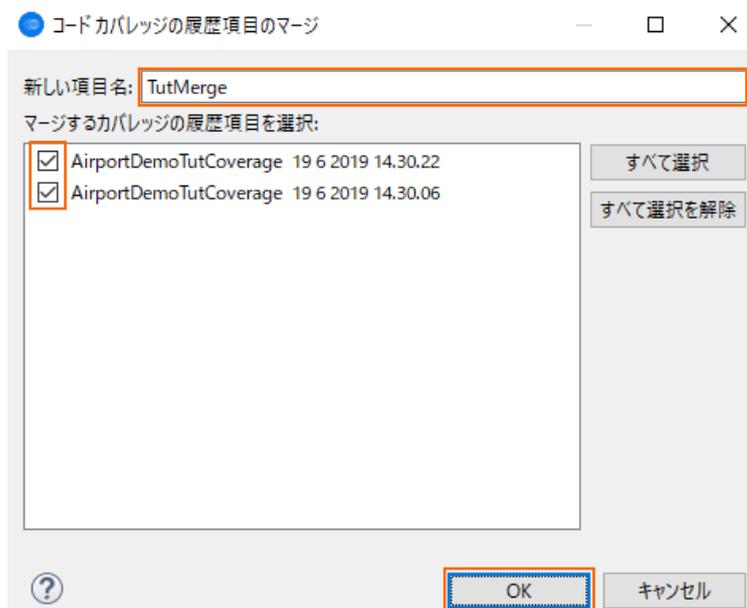
コードカバレッジビューより、さきほどの実行結果と異なっていることを確認します。例えば、lookup-one-airport や distance-between-airports 節のカバレッジ率が真逆になっていることが確認できます。

要素	カバレッジ	カバー済みブロック	未カバーブロック	ブロック数
aircode	31.8%	14	30	44
calculate-airport-distance	0.0%	0	5	5
close-airfile	100.0%	1	0	1
convert-angle	0.0%	0	7	7
display-airport	100.0%	1	0	1
distance-between-airports	0.0%	0	8	8
find-airport	66.7%	2	1	3
get-code-matches	0.0%	0	6	6
lookup-one-airport	100.0%	3	0	3
main	75.0%	6	2	8

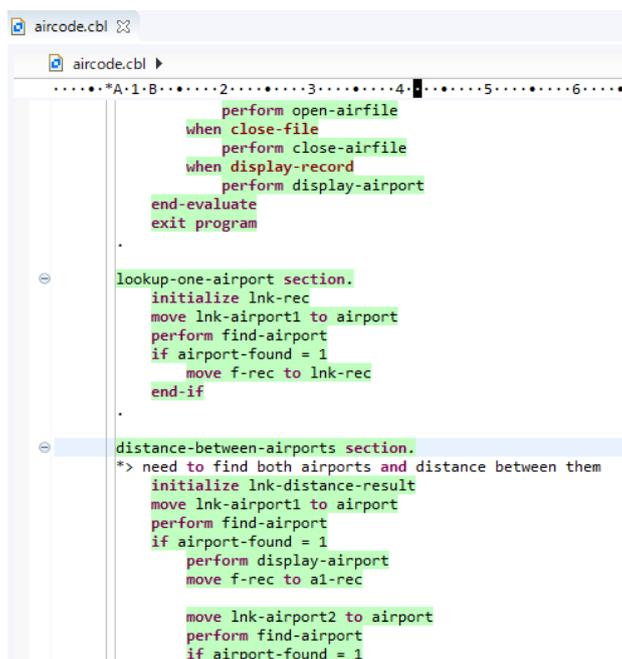
- コードカバレッジビュー上の「履歴項目のマージ」アイコンをクリックします。

未カバーブロック	ブロック数
30	44
5	5
0	1

3) 新しい項目名に “TutMerge” を入力し、マージ対象となる履歴項目にチェックした上で、[OK] ボタンをクリックします。



カバレッジ情報がマージされ、情報が更新されていることを確認してください。

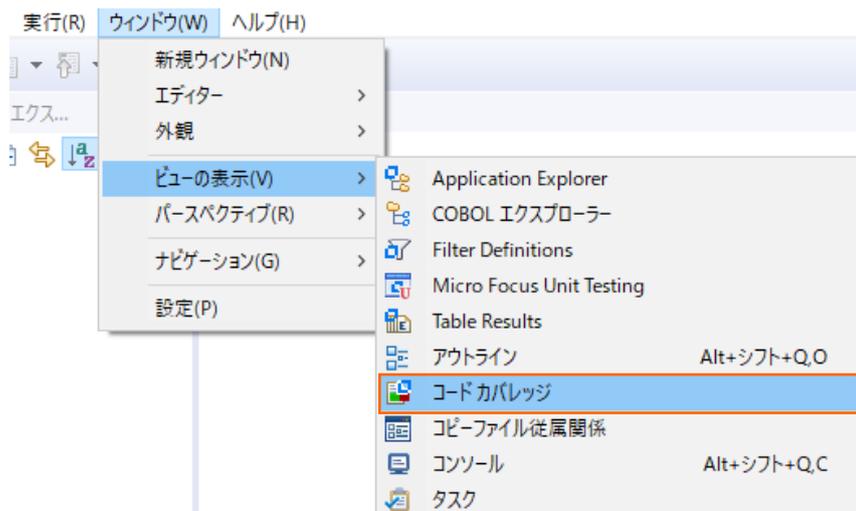


要素	カバレッジ	カバー済みブロック	未カバー ブロック	ブロック数
aircode	77.3 %	34	10	44
calculate-airport-distance	100.0 %	5	0	5
close-airfile	100.0 %	1	0	1
convert-angle	85.7 %	6	1	7
display-airport	100.0 %	1	0	1
distance-between-airports	100.0 %	8	0	8
find-airport	66.7 %	2	1	3
get-code-matches	0.0 %	0	6	6
lookup-one-airport	100.0 %	3	0	3
main	87.5 %	7	1	8

2.1.5. 既存カバレッジ情報ファイルの読み込み

過去に作成したカバレッジ情報ファイルを IDE 上に読み込む方法について学びます。なお、読み込み対象となるカバレッジ情報ファイルは、IDE 上からの実行だけでなく、後述するコマンドラインから実行された結果にも対応しています。

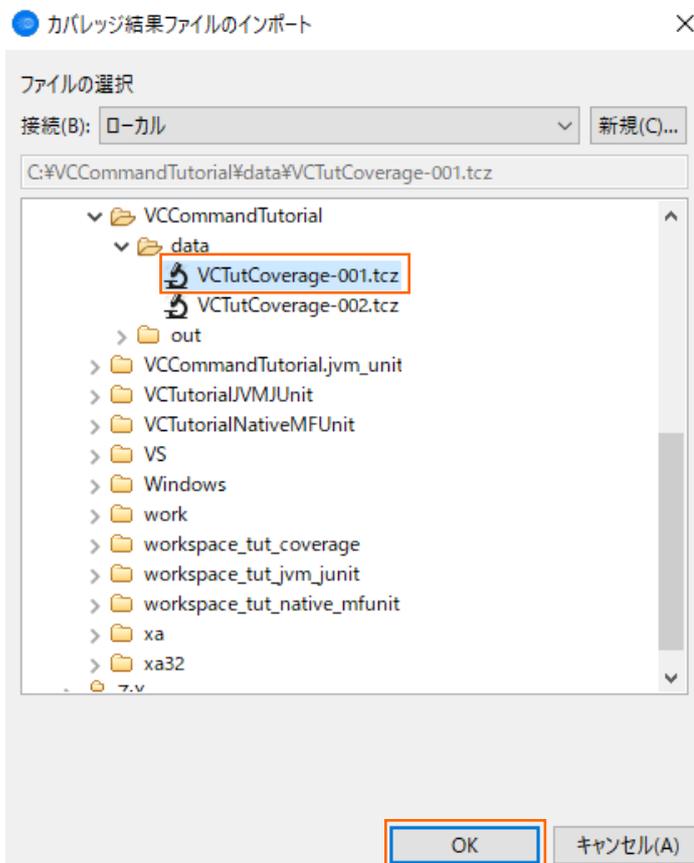
- 1) Eclipse IDE メニューより、[ウィンドウ(W)] > [ビューの表示(V)] > [コードカバレッジ] を選択して、コードカバレッジビューを開きます。



- 2) コードカバレッジビュー上の [セッションをインポート] アイコンをクリックします。

カバレッジ	カバー済みブロック	未カバー ブロック	ブロック数
1.8 %	14	30	44
0.0 %	0	5	5
0.0 %	1	0	1
0.0 %	0	7	7
0.0 %	1	0	1
0.0 %	0	8	8
6.7 %	2	1	3

- 表示したいカバレッジ情報ファイルを選択し、[OK] ボタンをクリックします。



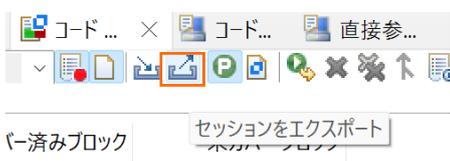
コードカバレッジビュー情報が更新され、カバレッジ情報が読み込まれていることを確認します。

要素	カバレッジ	カバー済みブロック	未カバーブロック	ブロック数
aircode	65.9 %	29	15	44
calculate-airport-distance	100.0 %	5	0	5
close-airfile	100.0 %	1	0	1
convert-angle	85.7 %	6	1	7
display-airport	100.0 %	1	0	1
distance-between-airports	100.0 %	8	0	8
find-airport	66.7 %	2	1	3
get-code-matches	0.0 %	0	6	6

2.1.6. カバレッジ情報ファイルのエクスポート

カバレッジ情報はエクスポートすることでファイルに保存しておくことができます。

- Eclipse IDE メニューより、[ウィンドウ(W)] > [ビューの表示(V)] > [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開きます。
- カバレッジのエクスポートアイコンをクリックすると保存先が選択できます。



2.2. コマンドラインからの実行

コードカバレッジ機能は、Eclipse 上からだけではなく、コマンドラインから実行できます。Jenkins などの CI ツールと連携することで、他言語開発と同様、プログラム開発 > コンパイル > テスト実行 > カバレッジ結果の確認 > プログラム開発 > ... といった開発サイクルを、COBOL 開発に導入することができます。

本節では、コマンドラインからコードカバレッジ機能を利用する手順を習得します。

2.2.1. コマンドラインからのコードカバレッジ機能の利用

- 1) コマンドライン実行用のフォルダを用意します。

```
C:¥>mkdir VCCommandTutorial
```

- 2) 以下の内容で、カバレッジ設定ファイルを VCCommandTutorial¥testcoverage.ini で作成します。

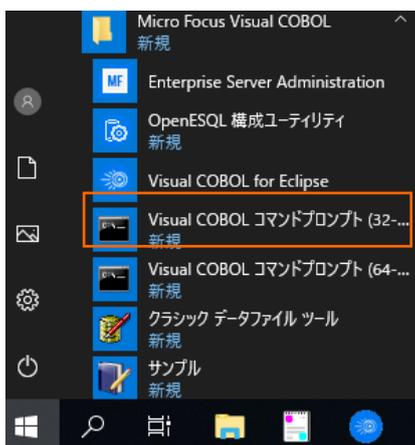
```
[TESTCOVER]
RESULT data¥VCTutCoverage-???.tcz
ECHOLOG NO
```

補足)

実行毎にカバレッジファイルが data フォルダ配下に VCTutCoverage-001.tcz, VCTutCoverage-002.tcz, ... というように作成されます。テスト形態によっては、複数回の実行結果を 1 ファイルに最初からマージしたい場合があります。これを行うには、カバレッジファイル名に続いて、ACCUMULATE 指令を追加します。

```
RESULT data¥VCTutCoverage.tcz ACCUMULATE
```

- 3) Windows メニューより、[Micro Focus Visual COBOL] > [Visual COBOL コマンドプロンプト] を選択します。



- 4) さきほど用意した作業フォルダに移動します。

```
C:¥Users¥Public¥Documents>cd ¥
C:¥>cd VCCommandTutorial
C:¥VCCommandTutorial>
```

- 5) 以下のコマンドを実行します。

- set ECLIPSE_WORKSPACE=c:¥workspace_tut_coverage
- cobol %ECLIPSE_WORKSPACE%¥AirportDemoTutCoverage¥aircode.cbl gnt,.¥ ANIM TESTCOVER COBIDY(.¥);
- cobol %ECLIPSE_WORKSPACE%¥AirportDemoTutCoverage¥main.cbl gnt,.¥ ANIM TESTCOVER COBIDY(.¥);

注意)

ECLIPSE_WORKSPACE は各環境に合わせて修正してください。

```
set ECLIPSE_WORKSPACE=C:¥Work¥Coverage

C:¥Work¥Coverage¥AirportDemoTutCoverage>cobol %ECLIPSE_WORKSPACE%¥AirportDemo
TutCoverage¥aircode.cbl gnt,.¥ ANIM TESTCOVER COBIDY(.¥);

Micro Focus COBOL
Version 9.0 (C) Copyright 1984-2023 Micro Focus or one of its affiliates.
* チェック終了 : エラーはありません- コード生成を開始します
* Generating C:¥Work¥Coverage¥AirportDemoTutCoverage¥aircode
* Data:          1456      Code:          6921      Literals:          352

C:¥Work¥Coverage¥AirportDemoTutCoverage>cobol %ECLIPSE_WORKSPACE%¥AirportDemo
TutCoverage¥main.cbl gnt,.¥ ANIM TESTCOVER COBIDY(.¥);

Micro Focus COBOL
Version 9.0 (C) Copyright 1984-2023 Micro Focus or one of its affiliates.
* チェック終了 : エラーはありません- コード生成を開始します
* Generating C:¥Work¥Coverage¥AirportDemoTutCoverage¥main
* Data:           208      Code:          1695      Literals:          192
```

- 6) アプリケーションを以下のコマンドで実行します。

- set TESTCOVER=.¥testcoverage.ini
- set dd_airports=%ECLIPSE_WORKSPACE%¥AirportDemoTutCoverage¥airports.dat
- runw main.gnt

前回同様の入力を行ってください。

1 回目：“HND△LHR” を入力後 Enter

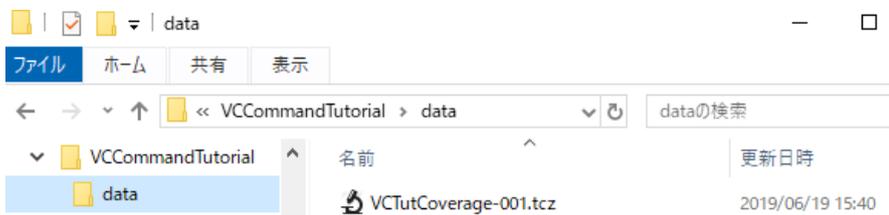
2 回目：何も入力せず Enter

補足：△は半角スペースを表します。



The image shows two overlapping windows. The background window is titled "Visual COBOL コマンドプロンプト (64-bit)". It contains the following command sequence:
C:¥VCCommandTutorial>set dd_airports=%ECLIPSE_WORKSPACE%¥AirportDemoTutCoverage¥airports.dat
C:¥VCCommandTutorial>set TESTCOVER=.%¥testcoverage.ini
C:¥VCCommandTutorial>runw main.gnt
C:¥VCCommandTutorial>
The foreground window is titled "runw.exe - COBOL テキストウインドウ". It displays the prompt "Enter an airport code, or two codes separated by space, or no code to exit:" followed by a blank line.

testcoverage.ini の RESULT 項目で指定した data フォルダ配下にカバレッジ情報ファイルが作成されていることを確認してください。



2.2.2. ユーティリティ機能の紹介

カバレッジ情報ファイルの HTML 形式でのレポート出力

1) 以下のコマンドを実行し、HTML レポート形式で出力を行います。

- tcreport data¥VC¥TutCoverage-001.tcz browse html reportloc(out)

 Visual COBOL コマンドプロンプト (64-bit)

```
C:¥VCCCommandTutorial>tcreport data¥VC¥TutCoverage-001.tcz browse html reportloc(out)
プログラム aircode.gnt 正常に処理されました
プログラム main.gnt 正常に処理されました
生成されたレポートのリストは以下にあります:
C:¥VCCCommandTutorial¥out¥TCIndex.htm
```

IDE と同様に通過した行は緑、未通過は赤色で表示されます。

行番号		回数
70	set open-file to true	
71	perform call-aircode-program	1
72		
73	perform until exit	1
74	display "Enter an airport code, or two codes " &	
75	"separated by space, or no code to exit:"	
76	accept user-input	
77	unstring user-input delimited by space	2
78	into ls-airport1, ls-airport2	
79		
80	if ls-airport1 = spaces	
81	exit perform	1
82	end-if	
83		
84	if ls-airport2 not = spaces	1
85	set get-distance to true	
86	perform call-aircode-program	1
87	display "Distance: " distance-km "kms"	1
88	distance-miles "miles"	
89	else	
90	move " " to ap-code of ls-rec	-
91	set get-details to true	
92	perform call-aircode-program	
93		
94	if ap-code OF ls-rec <> " "	-
95	then	
96	set display-record to true	
97	perform call-aircode-program	
98	end-if	
99	end-if	-
100	end-perform	

補足 1)

コマンドラインからの実行により作成されたカバレッジ情報を IDE 上に表示する方法は、2.1.5 を参照ください。

補足 2)

IDE 同様、複数カバレッジ情報ファイルのマージした結果レポートを出力することができ、以下の 2 通りの方法があります。

リストファイルを用いてマージ対象ファイルを指定する

```

C:¥VCCCommandTutorial>type resultfiles.txt
data¥VCTutCoverage-001.tcz
data¥VCTutCoverage-002.tcz
C:¥VCCCommandTutorial>tcreport @resultfiles.txt html reportloc(out)
プログラム aircode.gnt 正常に処理されました
プログラム main.gnt 正常に処理されました
生成されたレポートのリストは以下にあります:
C:¥VCCCommandTutorial¥out¥TCIndex.htm

tcreport コマンド実行時引数に半角カンマで対象ファイルを指定する
C:¥VCCCommandTutorial>tcreport data¥VCTutCoverage-001.tcz,data¥VCTutCoverage-002.tcz
html reportloc(out)
プログラム aircode.gnt 正常に処理されました
プログラム main.gnt 正常に処理されました
生成されたレポートのリストは以下にあります:
C:¥VCCCommandTutorial¥out¥TCIndex.htm

```

カバレッジ率の到達・未達判定

1) 以下のコマンドを実行します。

しきい値に達している場合)

```
- tcutil -t 60 data¥VCTutCoverage-001.tcz
```

```
C:¥VCCCommandTutorial>tcutil -t 60 data¥VCTutCoverage-001.tcz
```

```
C:¥VCCCommandTutorial>
```

しきい値に達していない場合)

```
- tcutil -t 70 data¥VCTutCoverage-001.tcz
```

```
C:¥VCCCommandTutorial>tcutil -t 70 data¥VCTutCoverage-001.tcz
```

```
カバレッジのしきい値に達しませんでした
```

```
C:¥VCCCommandTutorial>
```

補足)

しきい値の達成有無を判定するには、tcutil コマンドの終了コードを確認します。%ERRORLEVEL% に終了コードが設定されているため、この値が 0 の場合に達している、0 以外の場合に達していないと判断できます。

```
C:¥VCCCommandTutorial>tcutil -t 60 data¥VCTutCoverage-001.tcz
```

```
C:¥VCCCommandTutorial>echo %ERRORLEVEL%
```

```
0
```

```
C:¥VCCCommandTutorial>tcutil -t 70 data¥VCTutCoverage-001.tcz
```

```
カバレッジのしきい値に達しませんでした
```

```
C:¥VCCCommandTutorial>echo %ERRORLEVEL%
```

```
-1
```

WHAT'S NEXT

- 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。

免責事項

ここで紹介したソースコードは、機能説明のためのサンプルであり、製品の一部ではございません。ソースコードが実際に動作するか、御社業務に適合するかなどに関しまして、一切の保証はございません。ソースコード、説明、その他すべてについて、無謬性は保障されません。

ここで紹介するソースコードの一部、もしくは全部について、弊社に断りなく、御社の内部に組み込み、そのままご利用頂いても構いません。

本ソースコードの一部もしくは全部を二次的著作物に対して引用する場合、著作権法に基づき、適切な扱いを行ってください。