
Micro Focus 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 10
- Micro Focus Visual COBOL 7.0J 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

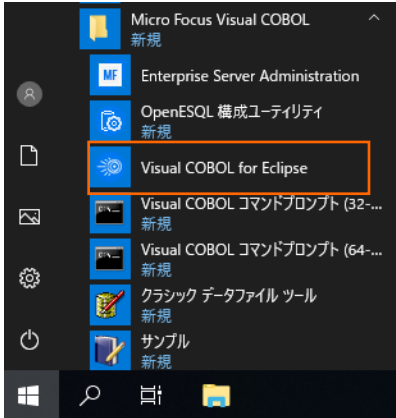
1. 前提
2. チュートリアル手順の概要
 - 2.1. IDE からの実行
 - 2.1.1. Eclipse の起動
 - 2.1.2. チュートリアルプロジェクトのインポート
 - 2.1.3. カバレッジを有効にしたアプリケーションの実行
 - 2.1.4. 既存カバレッジ情報ファイルの読み込み
 - 2.2. コマンドラインからの実行
 - 2.2.1. コマンドラインからのコードカバレッジ機能の利用
 - 2.2.2. ユーティリティ機能の紹介

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。

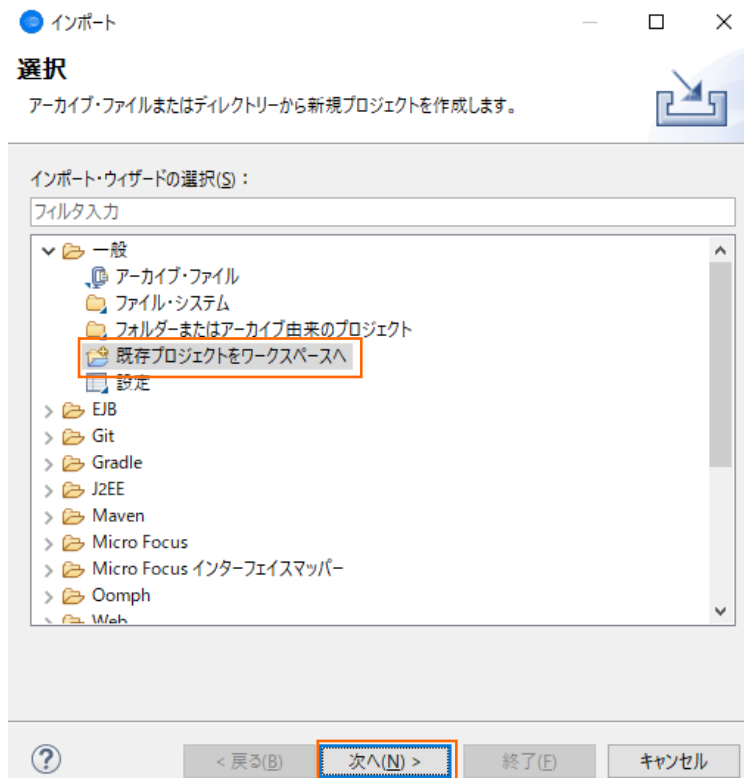


2.1.2. チュートリアルプロジェクトのインポート

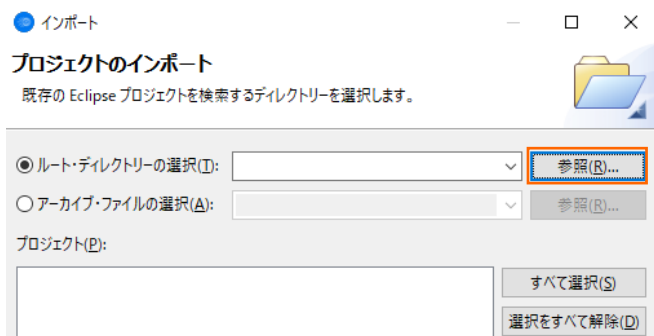
1) Eclipse IDE メニューの [ファイル(F)] > [インポート(I)] を選択します。



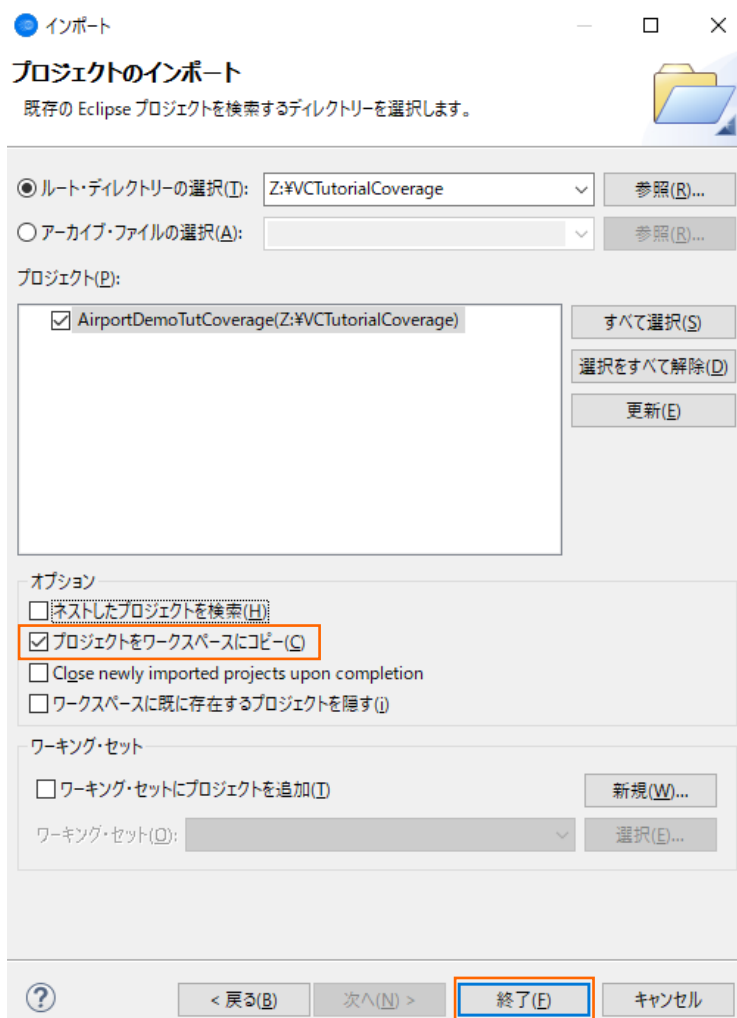
2) [一般] > [既存プロジェクトをワークスペースへ] を選択し、[次へ(N) >] ボタンをクリックします。



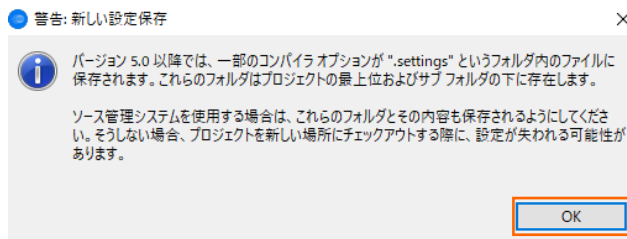
- 3) 「ルート・ディレクトリーの選択(T)」欄に、チュートリアルプロジェクトへのパスを指定します。[参照(R)] ボタンを押してダウンロードしたサンプルファイルを展開したフォルダを指定してください。



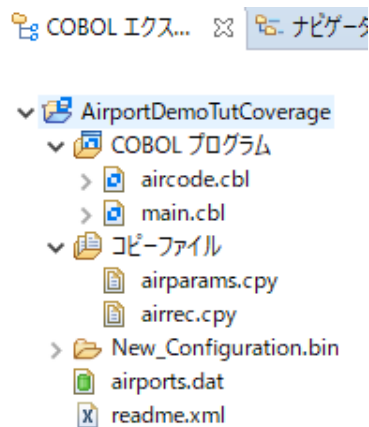
- 4) 「プロジェクトをワークスペースにコピー(C)」にチェックを入れたうえで、[終了(F)] ボタンをクリックします。



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。



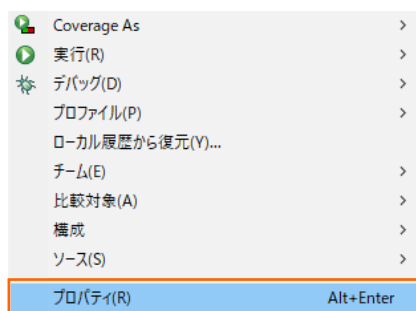
AirportDemoTutCoverage プロジェクトが作成されることを確認します。



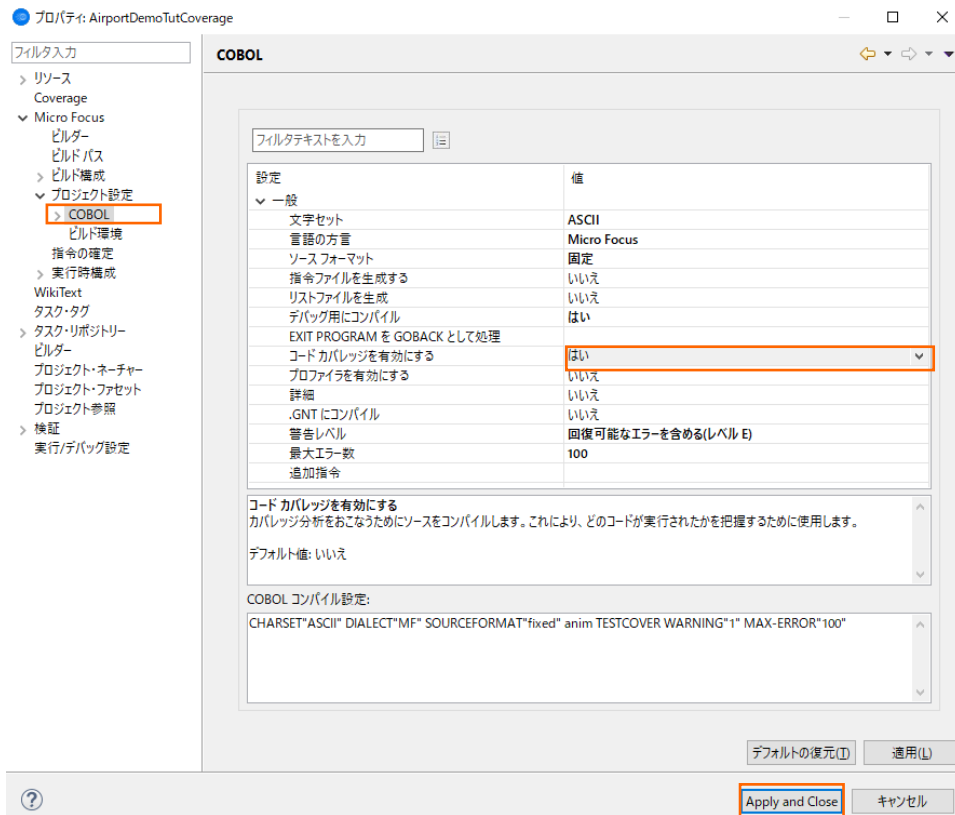
補足)

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。

- 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。

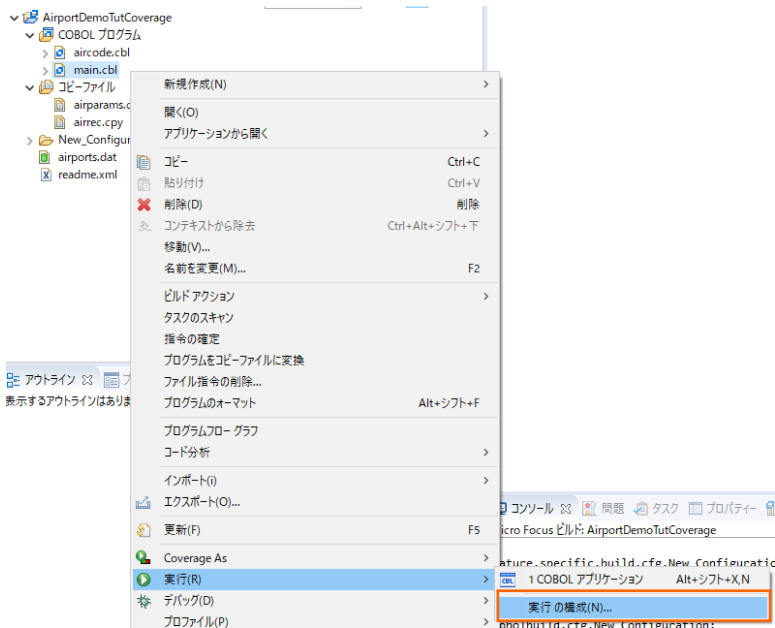


- 6) 画面左部のメニューより、[Micro Focus] > [プロジェクト設定] > [COBOL] を選択します。その後、「コードカバレッジを有効にする」を“はい”に変更して、[Apply and Close] ボタンをクリックします。

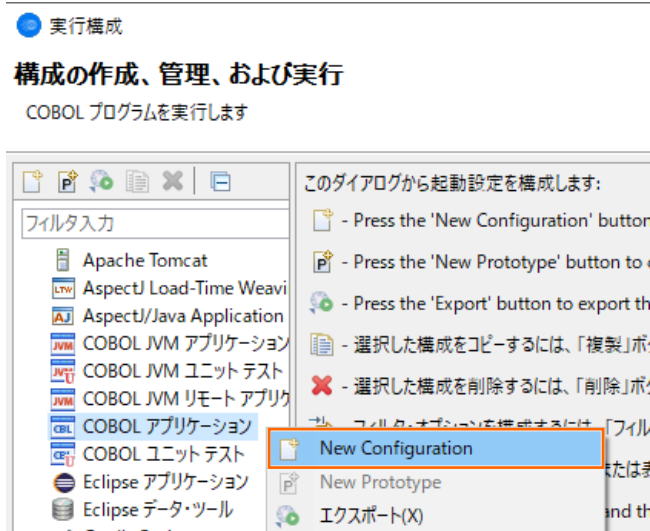


2.1.3. カバレッジを有効にしたアプリケーションの実行

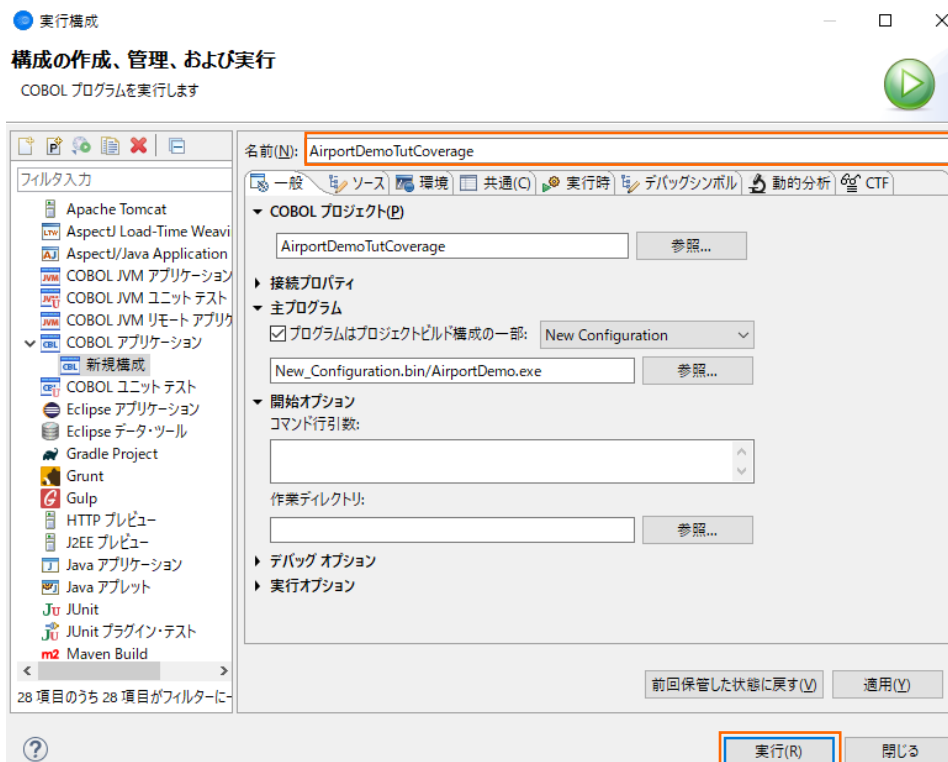
- 1) AirportDemoTutCoverage プロジェクト内の main.cbl を選択した上で、マウスの右クリックにてコンテキストメニューを表示し、[実行(R)] > [実行の構成(N)] を選択します。



- 「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、[New Configuration] を選択します。



- 「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。

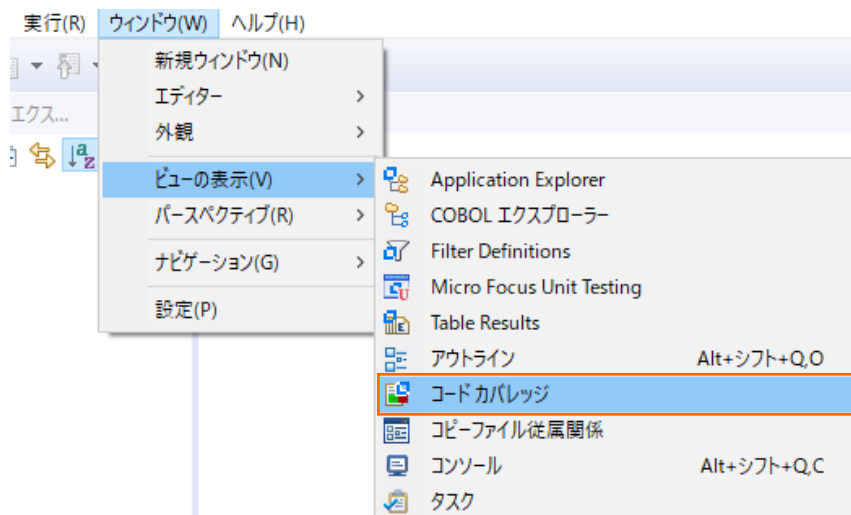
1 回目： “HND△LHR” を入力して Enter（羽田・ロンドンヒースロー空港間の距離を表示）

2 回目： 何も入力せず Enter（プログラムの終了）

```

c:\ 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。

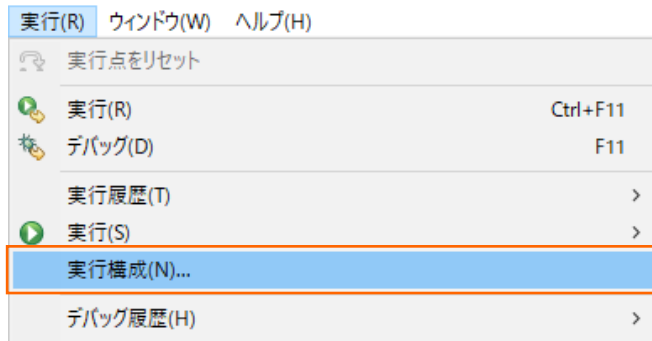


現時点では、カバレッジ情報が何も表示されていないことを確認します。

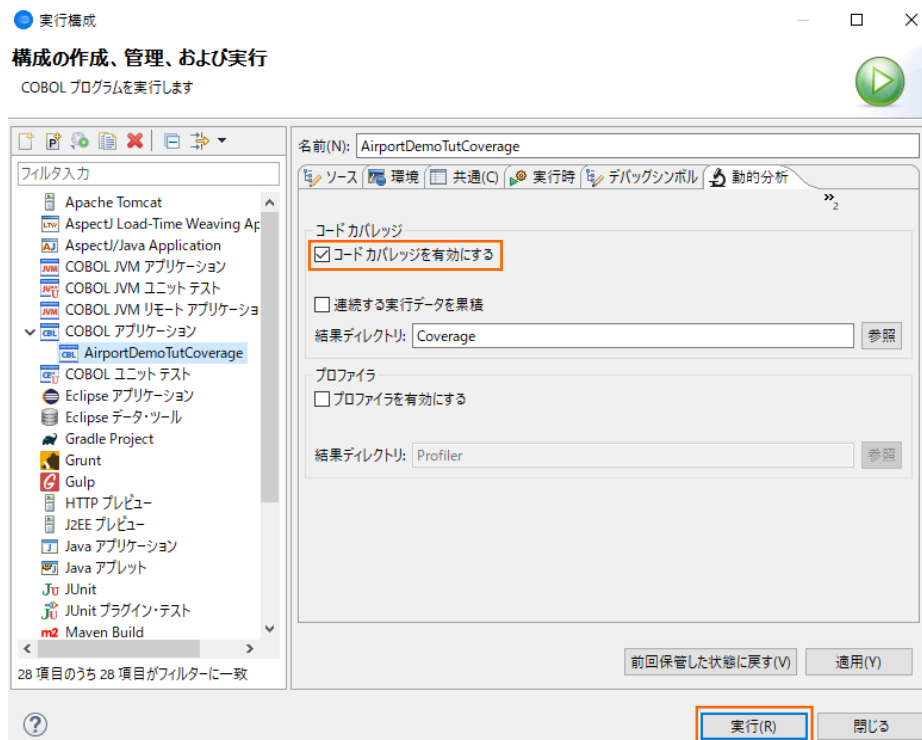
要素	カバレッジ	カバー済みブロック	未カバーブロック	ブロック数

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。

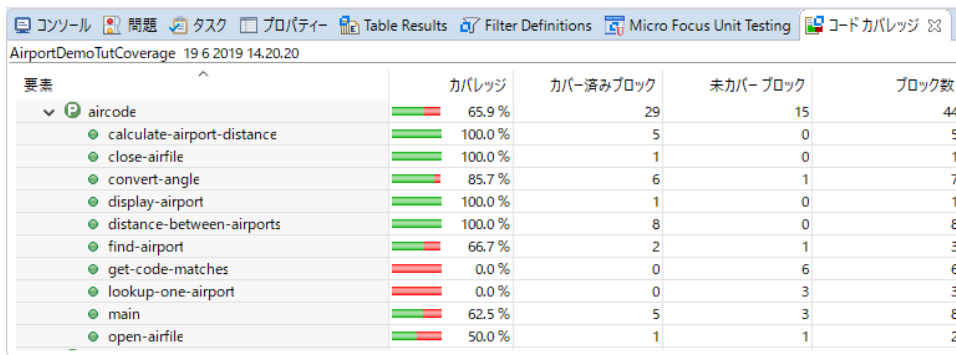


7) AirportDemoTutCoverage の実行構成が選択されている状態で、「動的分析」タブを選択し、「コードカバレッジを有効にする」をチェックします。その後、[実行(R)] ボタンをクリックします。



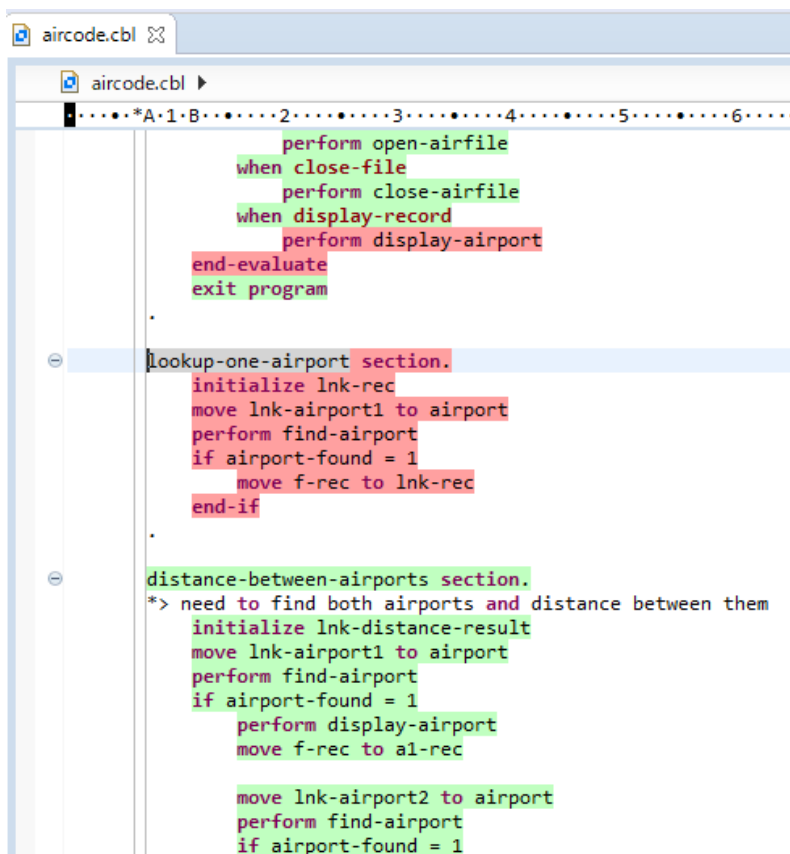
- 8) 再度、アプリケーション画面が起動されるため、前回同様、下記入力を行ってください。
- 1 回目： “HND△LHR” を入力して Enter（羽田・ロンドンヒースロー空港間の距離を表示）
 - 2 回目： 何も入力せず Enter（プログラムの終了）
- 補足) △は半角スペースを表します。

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
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
main	62.5%	5	3	8
open-airfile	50.0%	1	1	2

- 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

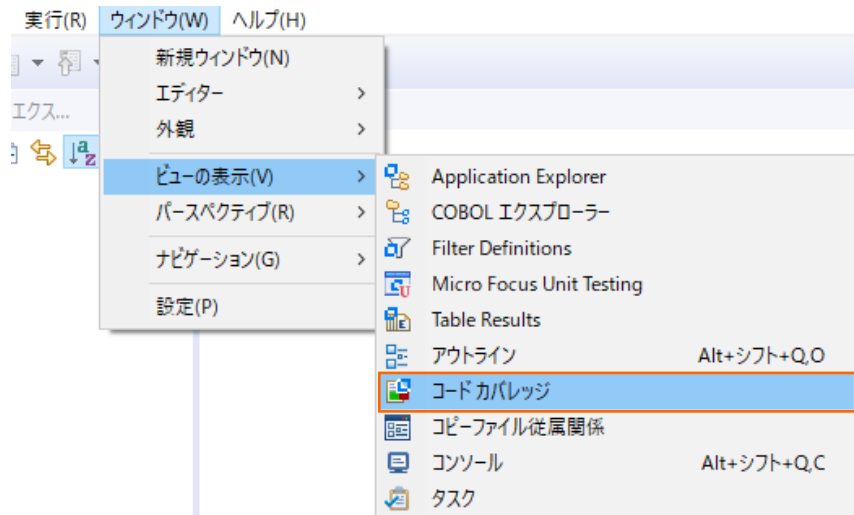
```

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. 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はなく、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。

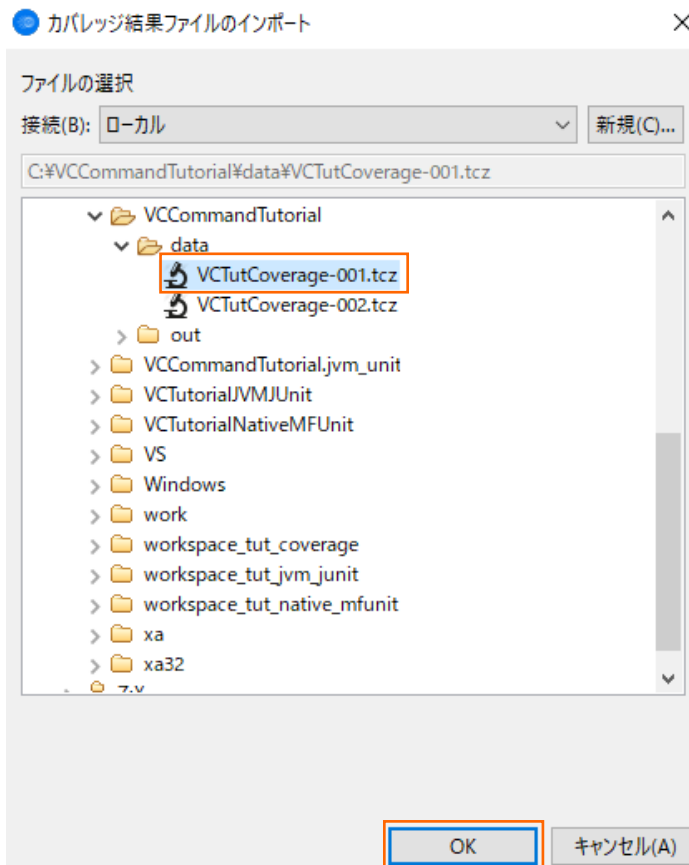
- 1) Eclipse IDE メニューより、[ウィンドウ(W)] > [ビューの表示(V)] > [コードカバレッジ] を選択して、コードカバレッジビューを開きます。



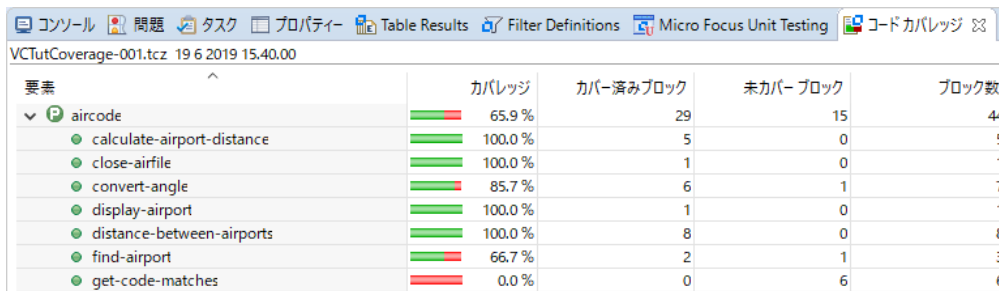
- 2) コードカバレッジビュー上の [セッションをインポート] アイコンをクリックします。



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。



VC TutCoverage-001.tcz 19 6 2019 15.40.00

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

2.2. 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。

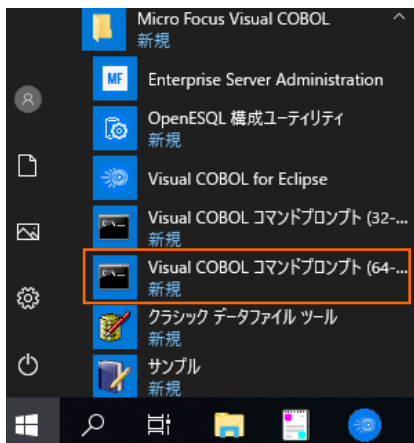
```
[TESTCOVER]
RESULT data¥VCTutCoverage-???.tcz
ECHOLOG NO
```

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 の Visual COBOL 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。

```
C:¥VCCommandTutorial>set ECLIPSE_WORKSPACE=c:¥workspace_tut_coverage
C:¥VCCommandTutorial>cobol %ECLIPSE_WORKSPACE%¥AirportDemoTutCoverage¥aircode.
cbl gnt,.¥ ANIM TESTCOVER COBIDY(.¥);

Micro Focus COBOL
Version 5.0 Copyright (C) Micro Focus 1984-2019. All rights reserved.
* 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:¥workspace_tut_coverage¥AirportDemoTutCoverage¥aircode
* Data:          1456      Code:          6920      Literals:          384
C:¥VCCommandTutorial>cobol %ECLIPSE_WORKSPACE%¥AirportDemoTutCoverage¥main.cbl
gnt,.¥ ANIM TESTCOVER COBIDY(.¥);

Micro Focus COBOL
Version 5.0 Copyright (C) Micro Focus 1984-2019. All rights reserved.
* 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:¥workspace_tut_coverage¥AirportDemoTutCoverage¥main
* Data:           208      Code:          1713      Literals:           192
C:¥VCCommandTutorial>dir
ドライブ C のボリューム ラベルがありません。
ボリューム シリアル番号は 6808-3539 です
C:¥VCCommandTutorial のディレクトリ
2019/06/19  15:32    <DIR>          .
2019/06/19  15:32    <DIR>          ..
2019/06/19  15:33                10,673 aircode.gnt
2019/06/19  15:33                59,648 aircode.idy
2019/06/19  15:33                3,571 main.gnt
```

```

2019/06/19 15:33          46,080 main.idy
2019/06/19 14:52          60 testcoverage.ini
          5 個のファイル          120,032 バイト
          2 個のディレクトリ 41,481,961,472 バイトの空き領域
  
```

6) アプリケーションを以下のコマンドで実行します。

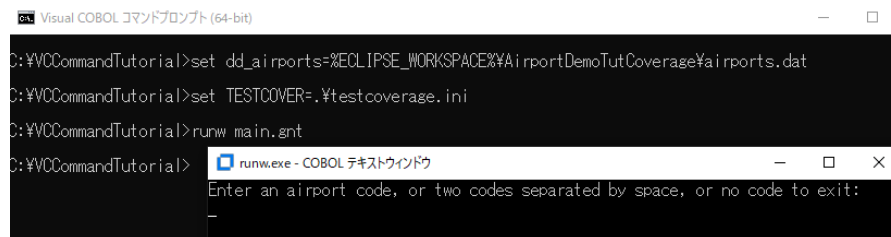
- set TESTCOVER=.¥testcoverage.ini
- set dd_airports=%ECLIPSE_WORKSPACE%¥AirportDemoTutCoverage¥airports.dat
- runw main.gnt

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

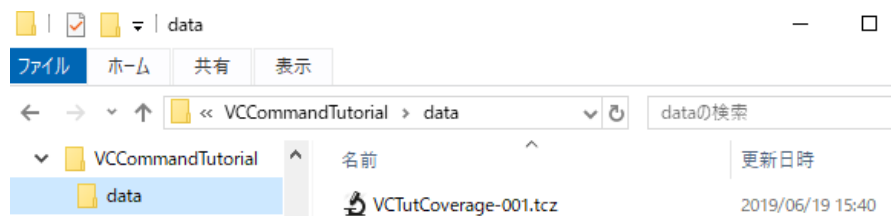
補足：△は半角スペースを表します。



```

Visual COBOL コマンドプロンプト (64-bit)
C:\¥VCCommandTutorial>set dd_airports=%ECLIPSE_WORKSPACE%¥AirportDemoTutCoverage¥airports.dat
C:\¥VCCommandTutorial>set TESTCOVER=.¥testcoverage.ini
C:\¥VCCommandTutorial>runw main.gnt
C:\¥VCCommandTutorial>
runw.exe - COBOL テキストウィンドウ
Enter an airport code, or two codes separated by space, or no code to exit:
  
```

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%VCTutCoverage-001.tcz browse html reportloc(out)

 Visual COBOL コマンドプロンプト (64-bit)

```
C:\VCCCommandTutorial>tcreport data%VCTutCoverage-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 とは異なり、未実行箇所のフォントが青色、実行済箇所のフォントが黒色で表示されます。

AIRCODE.GNT	
C:\VCCCommandTutorial\aircode.gnt	
コンパイル日時: 2019年06月19日: 15時33分17.84秒	
行番号	回数
195	4
196	4
197	4
198	-
199	-
200	1
201	1
202	-
203	-
204	1
205	1
206	2
207	2
208	-
209	-
210	4
211	-
212	-
213	-
214	-
215	-
216	-
217	-
218	-
219	-
220	-
221	-
222	1
223	1
224	1
225	1
226	1
227	1
228	1
229	1
230	1
231	1
232	1
233	1
234	1
235	1
236	1
237	1
238	1
239	1
240	1
241	-

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。

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。