

Visual COBOL チュートリアル

COBOL 開発 : Visual Studio – コードカバレッジ機能

1 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、マネージ 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。

2 前提

- 本チュートリアルで使用したマシン OS : Windows 11
- Visual COBOL 11.0 for Visual Studio 2022 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

- 1 目的
- 2 前提
- 3 チュートリアル手順
 - 3.1 IDE からの実行
 - 3.1.1 チュートリアルプロジェクトの作成
 - 3.1.2 カバレッジを有効にしたアプリケーション実行
 - 3.1.3 複数のカバレッジ情報のマージ
 - 3.1.4 既存カバレッジ情報ファイルの読み込み
 - 3.2 コマンドラインからの実行
 - 3.2.1 コマンドラインからのコードカバレッジ機能の利用
 - 3.2.2 ユーティリティ機能の紹介

3 チュートリアル手順

3.1 IDE からの実行

3.1.1 チュートリアルプロジェクトの作成

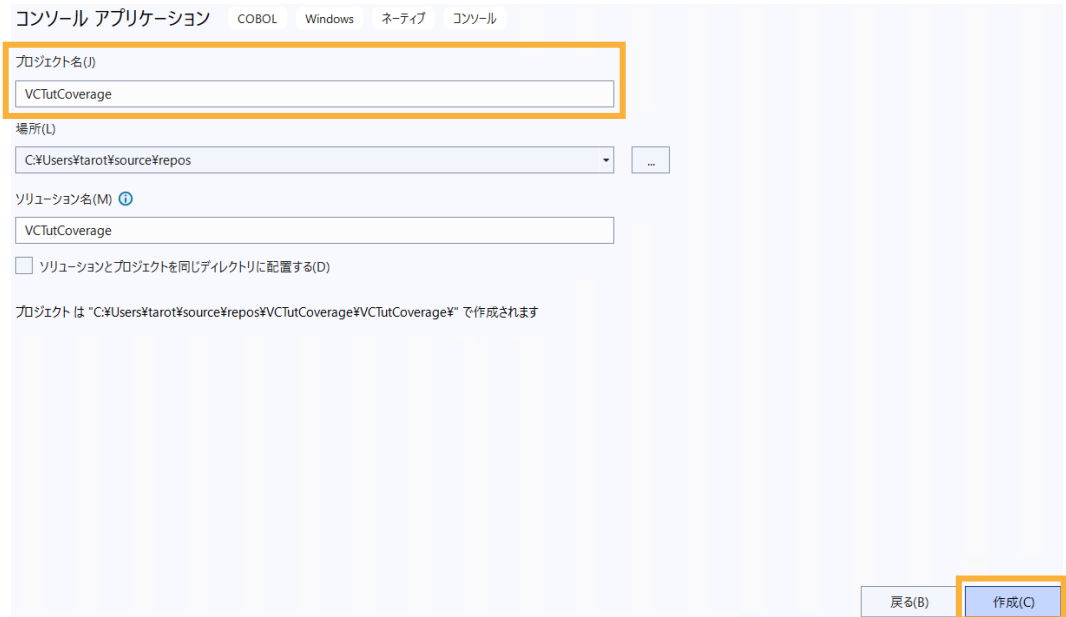
- 1) スタートメニューより、Visual Studio 2022 を起動します。
- 2) [新しいプロジェクトの作成] をクリックします。



- 3) 言語に “COBOL”、プラットフォームに “Windows”、プロジェクトタイプに “ネイティブ” を選択し、[コンソール アプリケーション] を選択した上で、[次へ(N)] をクリックします。



- 4) プロジェクト名に “VCTutCoverage” を入力し、[作成(C)] をクリックします。



コンソール アプリケーション COBOL Windows ネーティブ コンソール

プロジェクト名(J)
VCTutCoverage

場所(L)
C:\Users\tarot\source\repos

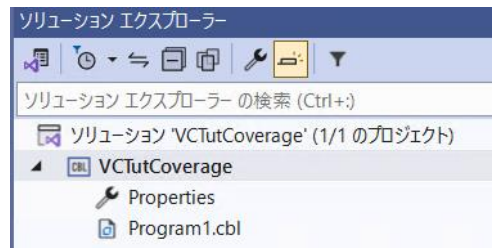
ソリューション名(M) ⓘ
VCTutCoverage

☐ 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(D)

プロジェクトは "C:\Users\tarot\source\repos\VCTutCoverage\VCTutCoverage" で作成されます

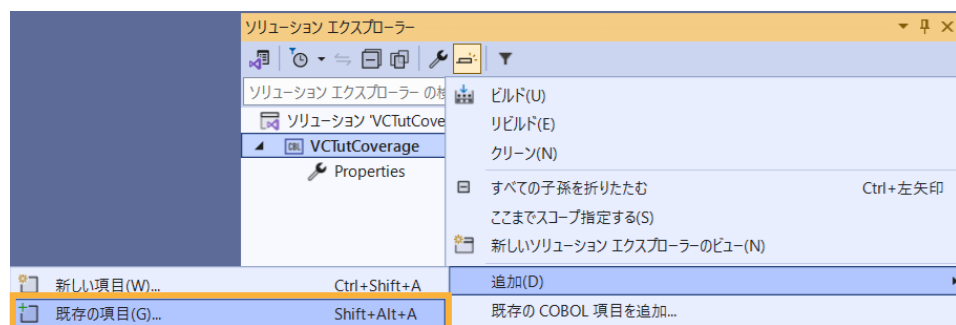
戻る(B) 作成(C)

プロジェクトが作成されます。

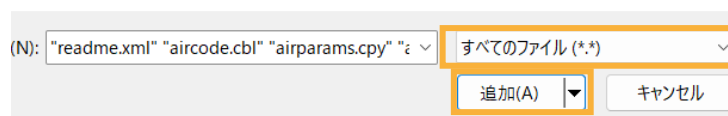
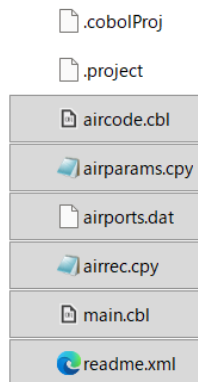


この Program1.cbl は不要なため、削除してください。

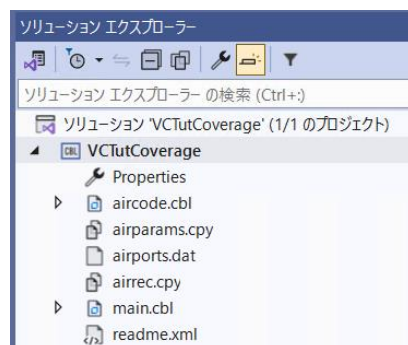
- 5) VCTutCoverage プロジェクト名を選択した状態で、マウスの右クリックにてコンテキストメニューを表示し、[追加(D)] > [既存の項目(G)] を選択します。



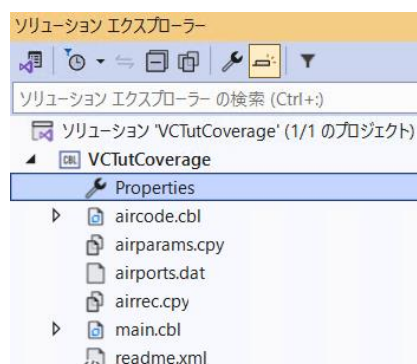
- 6) サンプルプログラムを展開したフォルダを選択し、“すべてのファイル (*.*)” を選択した後、.cobolProj, .project を除く全てのファイルを選択した上で、[追加(A)] をクリックします。



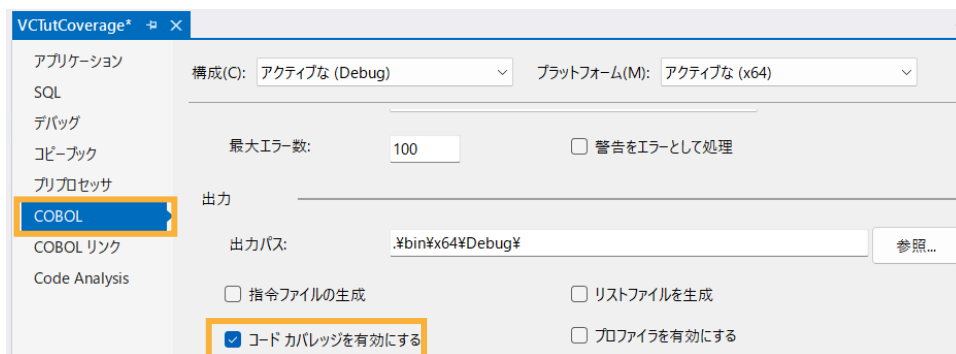
プロジェクトが以下ようになります。



- 7) VC TutCoverage プロジェクト配下の [Properties] をダブルクリックします。

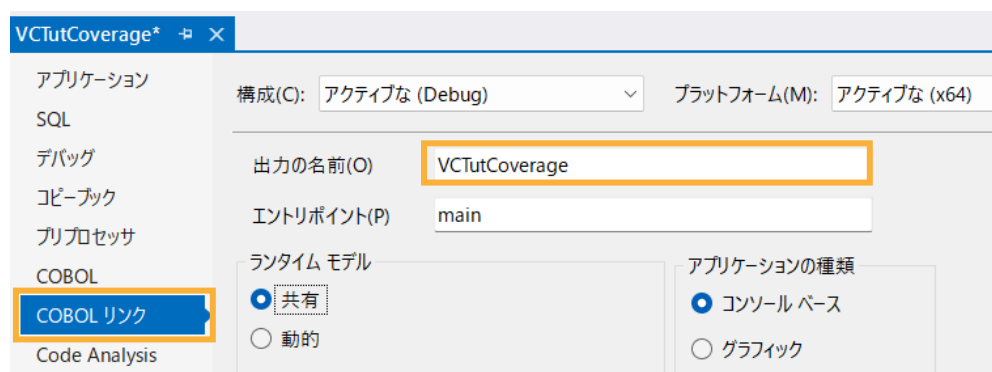


- 8) COBOL タブを選択し、「コードカバレッジを有効にする」にチェックを付けます。



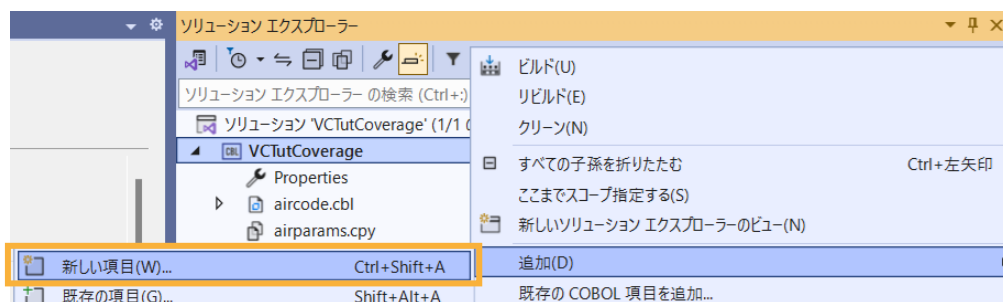
- 9) COBOL リンクタブを選択し、以下の入力を行った後、保存します。

エントリポイント: “main”

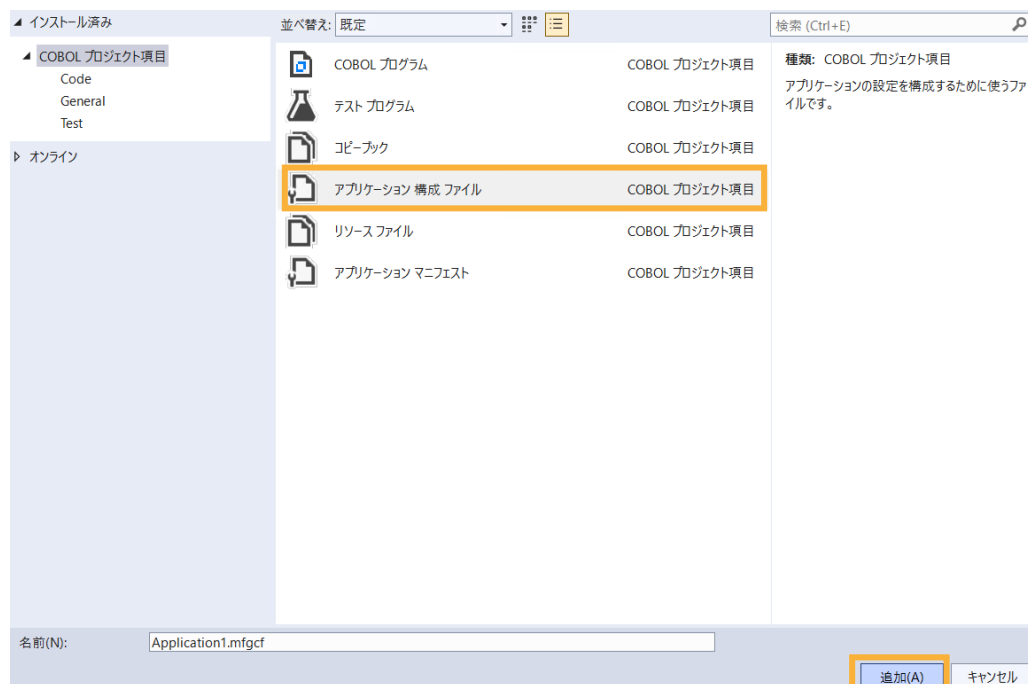


3.1.2 カバレッジを有効にしたアプリケーション実行

- 1) VCTutCoverage プロジェクト名を選択した状態で、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、[追加(D)] > [新しい項目(W)] を選択します。



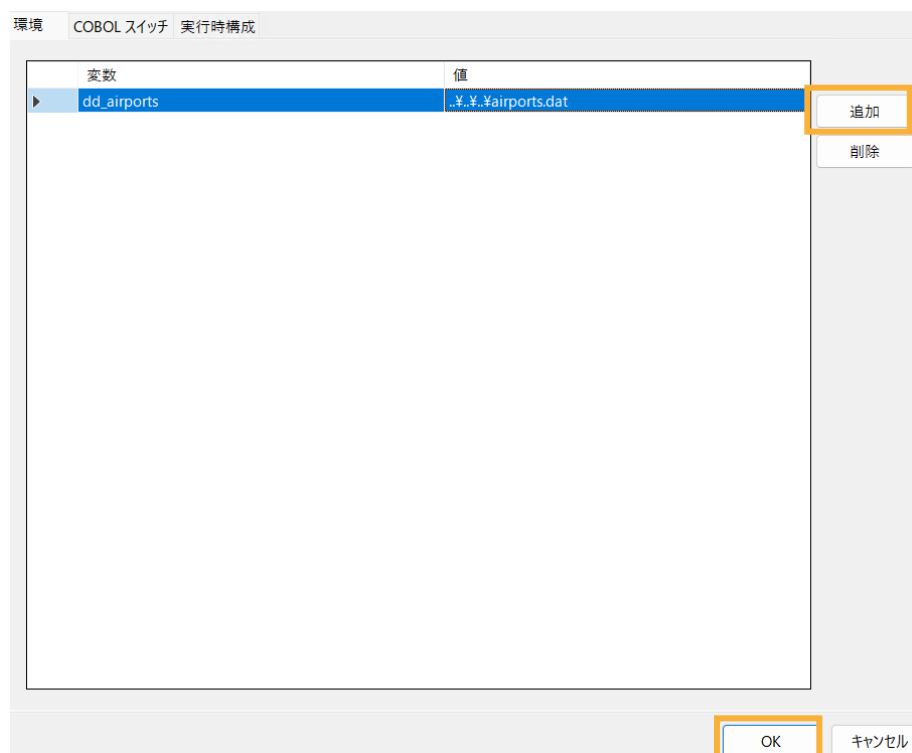
- 2) [アプリケーション構成ファイル] を選択し、[追加(A)] をクリックします。



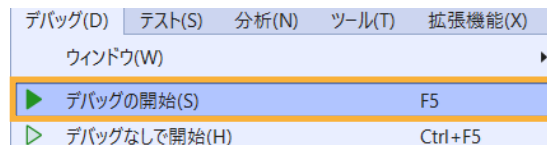
- 3) 追加された「Application1.mfgcf」をクリックし、表示されたダイアログ上で、[追加] をクリックしたうえで、以下の入力を行い、[OK] をクリックします。

変数: "dd_airports"

値: "..¥..¥..¥airports.dat"



- 4) Visual Studio IDE メニューより、[デバッグ(D)] > [デバッグの開始(S)] を選択します。



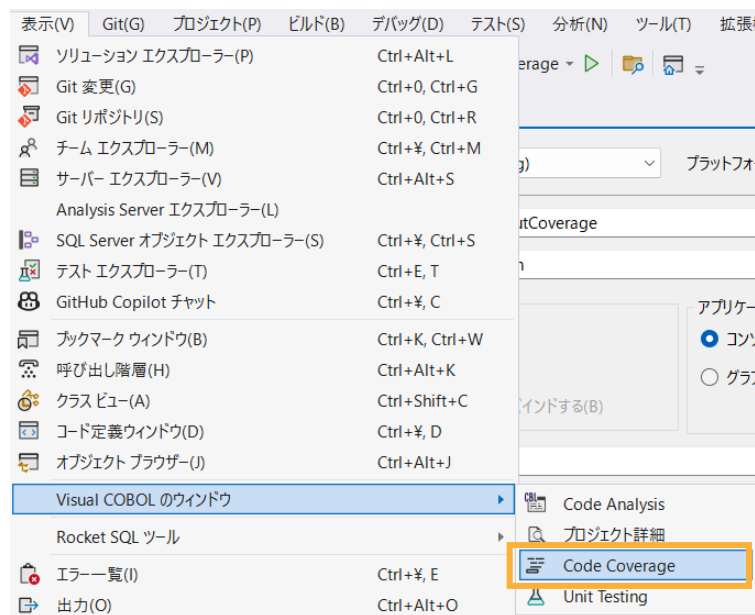
アプリケーション起動後、以下の入力を行います。

- 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）

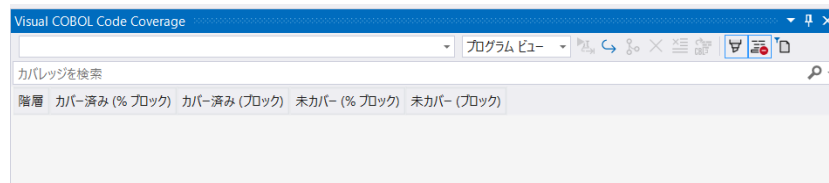
補足）△は半角スペースを表します。

```
Enter an airport code, or two codes separated by space, or no code to exit:
HND LHR
HND Tokyo Intl
      Japan                Lat:+035.552258 Lon:+139.077969
LHR Heathrow
      United Kingdom       Lat:+051.004775 Lon:-000.461389
Distance: 9,591kms 5,959miles
Enter an airport code, or two codes separated by space, or no code to exit:
```

- 5) Visual Studio IDE メニューより、[表示(V)] > [Visual COBOL のウィンドウ] > [Code Coverage] を選択します。

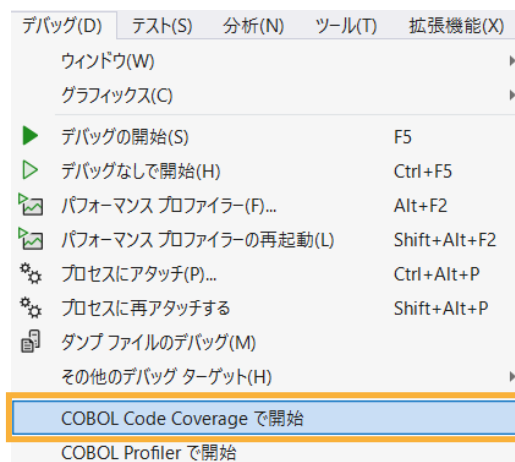


現時点では、カバレッジ情報が何も表示されていないことを確認します。



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) Visual Studio IDE メニューより、[デバッグ(D)] > [COBOL Code Coverage で開始] を選択します。



再度、アプリケーションが起動するため、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）

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。

Visual COBOL Code Coverage

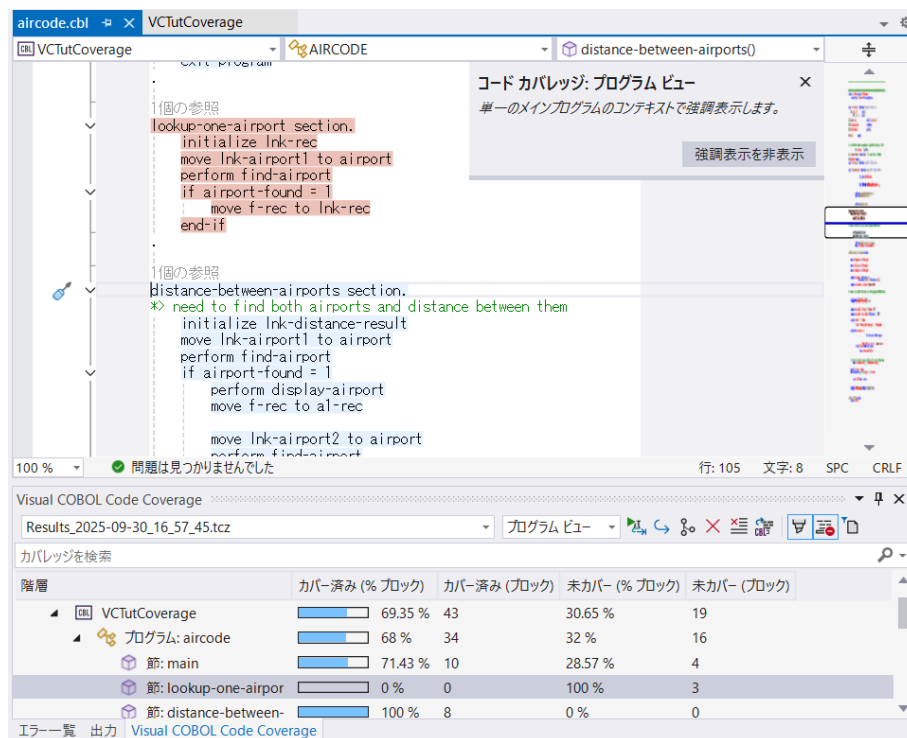
Results_2025-09-30_16_57_45.tcz

▼ プログラム ビュー

カバレッジを検索

階層		カバー済み (% ブロック)	カバー済み (ブロック)	未カバー (% ブロック)	未カバー (ブロック)
▼ Results_2025-09-30_16_57_45.tcz		69.35 %	43	30.65 %	19
▶ VCTutCoverage		69.35 %	43	30.65 %	19

- 7) 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、カバレッジ結果が表示されます。



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。

3.1.3 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。テスト全体のカバレッジ情報を確認するため、マージ機能を利用します。

1) 再度、コードカバレッジを実行し、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)

```
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:
```

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。

Visual COBOL Code Coverage

Results_2025-09-30_16_59_22.tcz プログラムビュー

カバレッジを検索

階層	カバレッジ (%)	カバレッジ (ブロック)	未カバレッジ (%)	未カバレッジ (ブロック)
Results_2025-09-30_16_59_22.tcz	48.39 %	30	51.61 %	32
VCutCoverage	48.39 %	30	51.61 %	32
プログラム: aircode	40 %	20	60 %	30
節: main	85.71 %	12	14.29 %	2
節: lookup-one-airpor	100 %	3	0 %	0
節: distance-between-	0 %	0	100 %	8
節: calculate-airport-d	0 %	0	100 %	5
節: convert-angle	0 %	0	100 %	7

2) コードカバレッジビュー上の「カバレッジをマージする」アイコンをクリックします。

Visual COBOL Code Coverage

Results_2025-09-30_16_59_22.tcz プログラムビュー

カバレッジを検索

階層	カバレッジ (%)	カバレッジ (ブロック)	未カバレッジ (%)	未カバレッジ (ブロック)
Results_2025-09-30_16_59_22.tcz	48.39 %	30	51.61 %	32
VCutCoverage	48.39 %	30	51.61 %	32

3) マージ対象となる履歴項目にチェックした上で、[OK] をクリックします。

カバレッジ実行回数によって、表示される項目数が異なります。

マージするカバレッジ結果を選択:

☒ Results_2025-09-30_16_59_22.tcz

☒ Results_2025-09-30_16_57_45.tcz

OK キャンセル

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。

Visual COBOL Code Coverage

マージされたカバレッジ (マージ元: 2個のファイル) プログラムビュー

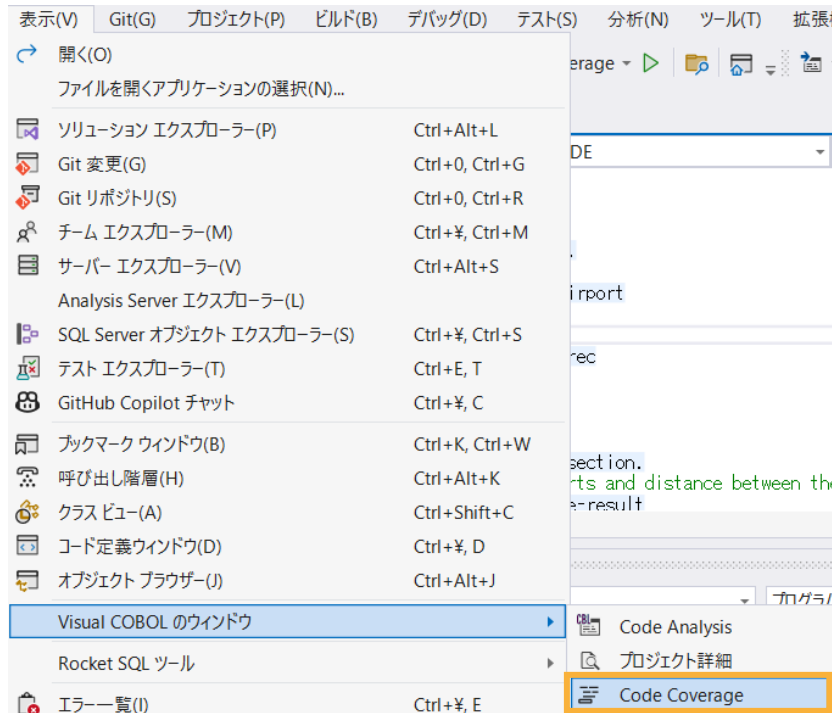
カバレッジを検索

階層	カバレッジ (%)	カバレッジ (ブロック)	未カバレッジ (%)	未カバレッジ (ブロック)
マージされたカバレッジ (マージ元: 2個のファイル)	83.87 %	52	16.13 %	10
VCutCoverage	83.87 %	52	16.13 %	10
プログラム: aircode	80 %	40	20 %	10
節: main	92.86 %	13	7.14 %	1
節: lookup-one-airpor	100 %	3	0 %	0
節: distance-between-	100 %	8	0 %	0
節: calculate-airport-d	100 %	5	0 %	0
節: convert-angle	85.71 %	6	14.29 %	1
節: display-airport	100 %	1	0 %	0

3.1.4 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、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も表示できます。ここでは、コマンドラインからの実行で作成した結果を使用するため、先に 3.2.1 を実行してください。

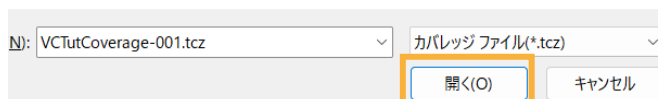
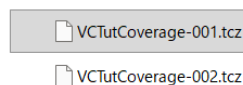
- 1) Visual Studio IDE メニューより、[表示(V)] > [Micro Focus Code Coverage] を選択して、コードカバレッジビューを開きます。



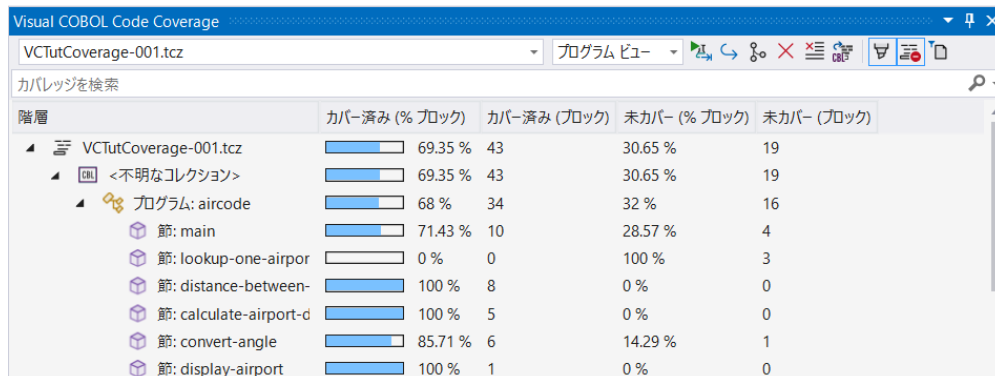
- 2) コードカバレッジビュー上の [カバレッジデータをインポートする] アイコンをクリックします。



- 3) ファイルダイアログ上で、インポートするカバレッジファイルを選択し、[開く(O)] をクリックします。



カバレッジ情報が更新されます。



階層	カバー済み (% ブロック)	カバー済み (ブロック)	未カバー (% ブロック)	未カバー (ブロック)
VCTutCoverage-001.tcz	69.35 %	43	30.65 %	19
<不明なコレクション>	69.35 %	43	30.65 %	19
プログラム: aircode	68 %	34	32 %	16
節: main	71.43 %	10	28.57 %	4
節: lookup-one-airpor	0 %	0	100 %	3
節: distance-between-	100 %	8	0 %	0
節: calculate-airport-d	100 %	5	0 %	0
節: convert-angle	85.71 %	6	14.29 %	1
節: display-airport	100 %	1	0 %	0

3.2 コマンドラインからの実行

コードカバレッジ機能は、Visual Studio 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。

3.2.1 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
C:¥>
```

- 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。

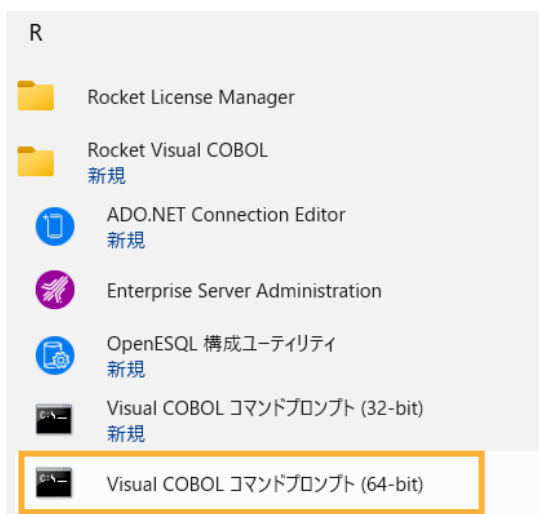
```
C:¥VCCommandTutorial>type testcoverage.ini
[TESTCOVER]
RESULT data¥VCTutCoverage-???.tcz
ECHOLOG NO
C:¥VCCommandTutorial>
```

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、「Visual COBOL コマンドプロンプト(64-bit)」を選択します。



- 4) プロンプト画面上で、さきほど用意した作業フォルダに移動します。

- 5) 以下のコマンドを実行します。

- set VS_SOLUTION_PATH=c:¥vs_solution_path
- cobol %VS_SOLUTION_PATH%¥VCTutCoverage¥aircode.cbl gnt,.¥ ANIM TESTCOVER COBIDY(.¥);
- cobol %VS_SOLUTION_PATH%¥VCTutCoverage¥main.cbl gnt,.¥ ANIM TESTCOVER COBIDY(.¥);

注意)

VS_SOLUTION_PATH は、Visual Studio のソリューションフォルダーです。ご自分の環境に合わせて修正してください。

```
C:¥VCCCommandTutorial>set VS_SOLUTION_PATH=c:¥vs_solution_path
C:¥VCCCommandTutorial>cobol %VS_SOLUTION_PATH%¥VCTutCoverage¥aircode.cbl gnt,.¥ ANIM TESTCOVER COBIDY(.¥);
Rocket (R) COBOL
Version 11.0 (C) 1984-2025 Rocket Software, Inc. or 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:¥Users¥tarot¥source¥repos¥VCTutCoverage¥VCTutCoverage¥aircode
* Data:      1456      Code:      6978      Literals:      400

C:¥VCCCommandTutorial>cobol %VS_SOLUTION_PATH%¥VCTutCoverage¥main.cbl gnt,.¥ ANIM TESTCOVER COBIDY(.¥);
Rocket (R) COBOL
Version 11.0 (C) 1984-2025 Rocket Software, Inc. or 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:¥Users¥tarot¥source¥repos¥VCTutCoverage¥VCTutCoverage¥main
* Data:      208      Code:      1677      Literals:      192

C:¥VCCCommandTutorial>
```

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=%VS_SOLUTION_PATH%¥VCTutCoverage¥airports.dat
- runw main

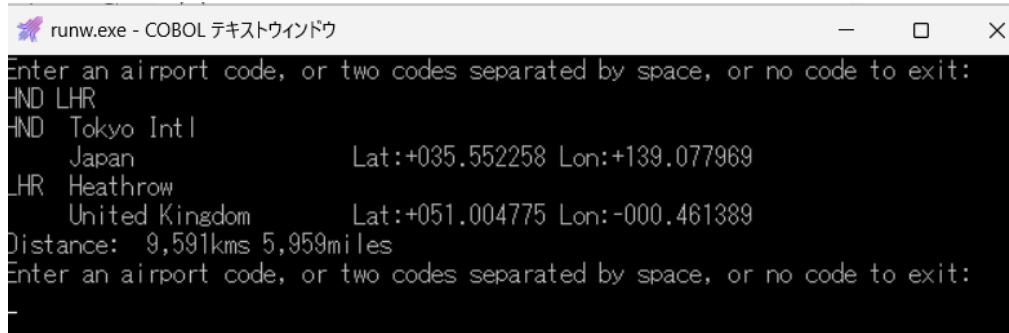
```
C:¥VCCCommandTutorial>set TESTCOVER=.%testcoverage.ini
C:¥VCCCommandTutorial>set dd_airports=%VS_SOLUTION_PATH%¥VCTutCoverage¥airports.dat
C:¥VCCCommandTutorial>runw main
C:¥VCCCommandTutorial>
```

表示された画面にて、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。



```
runw.exe - COBOL テキストウインドウ
Enter an airport code, or two codes separated by space, or no code to exit:
HND LHR
HND Tokyo Intl
      Japan           Lat:+035.552258 Lon:+139.077969
LHR Heathrow
      United Kingdom   Lat:+051.004775 Lon:-000.461389
Distance:  9,591kms 5,959miles
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ます。

```
C:¥VCCCommandTutorial>dir data
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 3837-A4A9 です

C:¥VCCCommandTutorial¥data のディレクトリ

2025/09/30  17:07    <DIR>          .
2025/09/30  17:06    <DIR>          ..
2025/09/30  17:07                1,429 VCTutCoverage-001.tcz
                1 個のファイル                1,429 バイト
                2 個のディレクトリ 81,035,120,640 バイトの空き領域

C:¥VCCCommandTutorial>
```

3.2.2 ユーティリティ機能の紹介

3.2.2.1 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)`

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。

aircode - TEST COVERAGE		
ファイル C:/VCCommandTutorial/out/aircode.htm		
201	when get-details	-
202	perform lookup-one-airport	-
203	when open-file	-
204	perform open-airfile	1
205	when close-file	2
206	perform close-airfile	2
207	when display-record	-
208	perform display-airport	-
209	end-evaluate	-
210	exit program	4
211	.	4
212		
213	lookup-one-airport section.	-
214	initialize lnk-rec	-
215	move lnk-airport1 to airport	-
216	perform find-airport	-
217	if airport-found = 1	-
218	move f-rec to lnk-rec	-
219	end-if	-
220	.	-
221		
222	distance-between-airports section.	1
223	*> need to find both airports and distance between them	
224	initialize lnk-distance-result	
225	move lnk-airport1 to airport	

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、3.1.4 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:¥VCCommandTutorial>type resultfiles.txt

data¥VCTutCoverage-001.tcz

data¥VCTutCoverage-002.tcz

C:¥VCCommandTutorial>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:¥VCCommandTutorial¥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:¥VCCommandTutorial>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

3.2.2.2 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
```

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。