

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 10.0 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

Rocket software

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。

開始する

↓	リポジトリのクローン(C) GitHub や Azure DevOps などのオンライン リポジトリ からコードを取得します
ď	プロジェクトやソリューションを開く(P) ローカルの Visual Studio プロジェクトまたは .sln ファイル を開きます
Ľ	ローカル フォルダーを開く(F) 任意のフォルダー内のコードに移動して編集します
韵	新しいプロジェクトの作成(N) 開始するには、コード スキャフォールディング付きのプロジェ クト テンプレートを選択します
	コードなしで続行(W) →

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 "ネーティブ" 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。

COBOL	•	Windows		•	ネーティブ		•
	Windows アプリケーション Windows アプリケーションを COBOL Windows	作成するための ネーティブ	ネイティブ プロ]ジェクトて	۶ġ.		
CBL C:\	コンソール アプリケーション ネイティブ コマンドライン アプ COBOL Windows	リケーションを作り ネーティブ	成するための コンソール	プロジェクト	ヽ です。		
	空のプロジェクト ローカル アプリケーションを作 COBOL Windows	成するための空(ネーティブ	のプロジェクト	です。			
	リンク ライブラリ 他のアプリケーションで使用す COBOL Windows	するクラスを作成 ネーティブ	するためのネ・ ライブラリ	イティブ プ	ロジェクトです。		
	ユニット テスト ライプラリ MFUnit テスト ライプラリを作 COBOL Windows	■成するためのネ ネーティブ	イティブ プロミ テスト	ジェクトです ライブラリ	t.		
C	Micro Focus INT/GNT Micro Focus INT または G	NT コードを作成	けるためのブ	ロジェクト	रुंच.		
				[戻る(B)	次へ(N)	





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ク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。

	ソリューション エクスプローラー			▼ ₽×
	√ ™ - ⇒ ⊡ ⊡ ≯	, er	T	
	ソリューション エクスプローラー の枝		ビルド(U)	
	😽 ソリューション 'VCTutCove		リビルド(E)	
	▲ WCTutCoverage		クリーン(N)	
	🔑 Properties	8	すべての子孫を折りたたむ ここまでスコープ指定する(S) 新しいソリューション エクスプローラーのビュー(N)	Ctrl+左矢印
1 新しい項目(W)	Ctrl+Shift+A		追加(D)	•
1 既存の項目(G)	Shift+Alt+A		既存の COBOL 項目を追加	



 5) 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。

	更新日時	種類
.cobolProj	2024/08/28 10:05	COBOLPROJ
D.project	2024/08/28 10:05	PROJECT ファ・
aircode.cbl	2024/08/28 10:05	COBOL ソース
airparams.cpy	2024/08/28 10:05	COBOL JĽ-:
airports.dat	2024/08/28 10:05	DAT ファイル
airrec.cpy	2024/08/28 10:05	COBOL JĽ-:
D main.cbl	2024/08/28 10:05	COBOL ソース
readme.xml	2024/08/28 10:05	XML ファイル
N): "aircode.cbl" "airparams.cpy" "airports.dat" "	'& 〜 すべてのファイル (*.*)	~
	追加(A) ►	キャンセル

プロジェクトが以下のようになります。



7) VCTutCoverage 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。

VCTutCoverage* キ × アプリケーション SOI	構成(C): アクティブな (Debu	g)	~
デバッグ コピーブック	エラーおよび警告		
プリプロセッサ	警告レベル:	回復可能なエラーを含める(レベル E) ~	
COBOL COBOL リンク Micro Focus Code Analysis	最大エラー数: 出力 —————	100 🗌 警告をエラーとして処理	
	出力パス:	.¥bin¥x64¥Debug¥	参照
	□ 指令ファイルの生成	□ リストファイルを生成	
	■ コード カバレッジを有効	にする フロファイラを有効にする	
	追加指令		

- 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"

VCTutCoverage 😕 🗙		
アプリケーション	構成(C)・ アクティブな (Debug)	プラットフォーム(M)・アクティブか(y64) ~
SQL		
デバッグ	出力の名前(O) VCTutCoverage	
コピーブック	T:\bll#d:\b(D) main	
プリプロセッサ	171-271-271-(F)	
COBOL	ランタイム モデル	アプリケーションの種類
COBOL リンク	● 共有	● コンソール ベース
Mic o Focus Code Analysis	○ 動的	 グラフィック
	 現在のランタイムだけにバインドする(B) 	

3.1.2 カバレッジを有効にしたアプリケーション実行

 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ク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。

	~ ‡	ソリューション エクスプローラー		- ₽ ×
		 ● ・ 二 ● ● ● ● ● ● ● ● ● ● ●	ビルド(U) リビルド(E) クリーン(N) すべての子孫を折りたたむ ここまでスコープ指定する(S) 新しいソリューション エクスプローラーのビュー(N)	Ctrl+左矢印
°	新しい項目(W)	Ctrl+Shift+A	追加(D)	•
to	既存の項目(G)	Shift+Alt+A	既存の COBOL 項目を追加	

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。



▲ インストール済み	並べ替え: 既定	■ # =	検索 (Ctrl+E)
▲ COBOL プロジェクト項目 Code	COBOL プログラム	COBOL プロジェクト項目	種類: COBOL プロジェクト項目
General Test	デスト プログラム	COBOL プロジェクト項目	アフリケーションの設定を構成するために使うファ イルです。
▶ オンライン	בעל-באר	COBOL カジェクト項目	
	🔎 アプリケーション 構成 ファイル	COBOL プロジェクト項目	
	עריע די איז איז איז איז איז איז איז איז איז אי	COBOL プロジェクト項目	
	アプリケーション マニフェスト	COBOL プロジェクト項目	
名前(N): Applicatio	n1.mfgcf		
			追加(A) キャンセル

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"

值: "..¥..¥..¥airports.dat"

環境	COBOL スイッチ 実行時構成		
	変数	值	
•	dd_airports	¥¥airports.dat	追加
			削除
		ОК	キャンセル



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。

デバ	、ッグ(D)	テスト(S)	分析(N)	ツール(T)	拡張機能(X)	
	ウィンドゥ	ל(W)				۲
	デバッグ	の開始(S)			F5	

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)] > [Micro Focus Code Coverage] を選択します。

表示	₹(V)	Git(G)	プロジェクト(P)	ビルド(B)	デバッグ(D)	テスト(
	צעצ	1-ション I	Ctrl+Alt+L			
	Git 3	変更(G)	Ctrl+0, Ctrl+	+G		
\$ 7	Git	リポジトリ(S)		Ctrl+0, Ctrl+	+R
RR	<i>₹−1</i>	→ エクスプロ·	-ラ-(M)		Ctrl+¥, Ctrl-	+M
	サーノ	(ー エクスプ	ローラー(V)		Ctrl+Alt+S	
20	SQL	Server オフ	ジェクト エクスプロ・	-ラ-(S)	Ctrl+¥, Ctrl-	+S
π×	テスト	- エクスプロ・	Ctrl+E, T			
83	GitH	lub Copilo	t チャット		Ctrl+¥, C	
	Micr	ro Focus C	ode Analysis			
<u>à</u>	プロシ	バクト詳細				
11	Micr	o Focus C	ode Coverage			

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。

Micro	Micro Focus Code Coverage 🔹 🖣 🗙						
	 プログラムビュー ・ 塩 ら ‰ × 準 滞 						
カバレ	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)			
T5	一覧 出力 Micro Foo	us Code Coverage					
	Se HJJ Micro roc	us coue coverage					

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)] > [Micro Focus 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。

Micro Focus Code Coverage				- ₽ ×
Results_2024-08-28_10_23_22.tcz		 プログラム 	Ľı- → 🖳 🔾	‰ × ≍≣ 穿 🛛 🖌
カバレッジを検索				- م
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_2024-08-28_10_23_22.tcz 	67.86 %	38	32.14 %	18
▷ CBL VCTutCoverage	67.86 %	38	32.14 %	18
I エラー一覧 出力 Micro Focus Code Covera	ge			

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。

aircode.cbl 😐 🗙 VCTutCoverage						
DIL VCTutCoverage	👻 🖓 AIRC	ODE		- 6	🖯 lookup-one-airport()	•
when open-fil perform o when close-fi perform c when display-	e pen-airfile le lose-airfile record				コード カバレッジ: プログラム ビュ ・ <i>単一のメインプ</i> ログラムのコンテキストで、	- × 強調表示します。
perform d end-evaluat exit program 1個の参照	Isplay-airport					如何不可能
 Iookupronerainport se initialize Inkrainport perform findrainport if ainport found move firec to end=if 1個の参照 	ction. c ort = 1 Ink-rec					
100 % 🔹 🕑 問題は見つかりませんでした	4				•	行: 96 文字: 8 5
Micro Focus Code Coverage						
Results_2024-08-28_10_23_22.tcz		 プログラム 	- El- 🔸 🏹 🔾	‰ × ≝ 🖁	7 😽 👪 🖸 👘	カバレッジを検索
階層	カバー済み (% ブロック)	カバー済み (プロック)	未カバー (% プロック	7) 未カバー (プロ・	ック)	
 Results_2024-08-28_10_23_22.tcz 	67.86 %	38	32.14 %	18		
 VCTutCoverage 	67.86 %	38	32.14 %	18		
🔺 🔩 プログラム: aircode	65.91 %	29	34.09 %	15		
😚 節: main	62.5 %	5	37.5 %	3		
😚 節: lookup-one-airpor	0 %	0	100 %	3		
😚 節: distance-between-	100 %	8	0 %	0		
⑦ 節: calculate-airport-d	100 %	5	0 %	0		
エラー一覧 出力 Micro Focus Code Coverage	ge 05.71.%	c	14 20 9/	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。

Rocket software

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。

Micro Focus Code C	Coverage				
Results_2024-08-2	8_10_30_38.tcz			・ プログラム ビュー	 ➡ ➡ ↓ ↓
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
٢	節: close-airfile		100 %	1	0 %
Ô	節: convert-angle		0 %	0	100 %
Û	節: display-airport		100 %	1	0 %
٢	節: distance-between-airports		0 %	0	100 %
٢	節: find-airport		66.67 %	2	33.33 %
٢	節: get-code-matches		0 %	0	100 %
٢	節: lookup-one-airport		100 %	3	0 %
Micro Focus Unit Te		ocus Code Co	verage	c	DE ₽/

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。

Micro Focus Code Coverage				
Results_2024-08-28_10_30_38.tcz		 プログラム ビュー 	- 🖳 🖕 🗞 🕻	× 🚈 🎘 🛛 🙀 🔁
階層	カバー済み (% ブロック)	カバー済み (ブロック)	未カバー (% ブレ	カバレッジをマージする 7)
☆ 節: close-airfile	100 %	1	0 %	0
☆ 節: convert-angle	0 %	0	100 %	7
😚 節: display-airport	100 %	1	0 %	0
😚 節: distance-between-airports	0 %	0	100 %	8



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_2024-08-28_10_30_38.tcz	
Results_2024-08-28_10_29_06.tcz	
Results_2024-08-28_10_23_22.tcz	
	0K +++>/17/1
	47700

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。

Micro Focus	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	► № S S S S	¥ 🖀 🕅 🐺 🖸
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)
	🕥 節: close-airfile		100 %	1	0 %	0
	🕥 節: convert-angle		85.71 %	6	14.29 %	1
	😚 節: display-airport		100 %	1	0 %	0
	😚 節: distance-between-airports		100 %	8	0 %	0
	😚 節: find-airport		66.67 %	2	33.33 %	1
	😚 節: get-code-matches		0 %	0	100 %	6
	😚 節: lookup-one-airport		100 %	3	0 %	0
Micro Focu	s Unit Testing エラー一覧 出力 Micro	Focus Code Co	overage	7	10 5 9/	1

3.1.4 既存カバレッジ情報ファイルの読込み

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?を実行してください。

 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。



	開く(O)						
アプリケーションの選打	沢(N)						
ソリューション エクスプローラー(P) Ctrl+Alt+L							
		Ctrl+0, Ctrl	+G				
(S)		Ctrl+0, Ctrl	+R				
チーム エクスプローラー(M) Ctrl+¥, Ctrl+M							
サーバー エクスプローラー(V) Ctrl+Alt+S							
SQL Server オブジェクト エクスプローラー(S) Ctrl+							
ローラー(T)		Ctrl+E, T					
ilot チャット		Ctrl+¥, C					
Code Analysis							
細							
Code Coverage							
	アプリケーションの選打 エクスプローラー(P) J(S) プローラー(M) スプローラー(V) オブジェクト エクスプロ プローラー(T) bilot チャット s Code Analysis 細 s Code Coverage	アプリケーションの選択(N) エクスプローラー(P) J(S) プローラー(M) スプローラー(V) オブジェクト エクスプローラー(S) プローラー(T) billot チャット & Code Analysis 細 & Code Coverage	アプリケーションの選択(N) エクスプローラー(P) Ctrl+Alt+L Ctrl+0, Ctrl グローラー(M) Ctrl+3, Ctrl ズプローラー(V) Ctrl+4, Ctrl オプジェクト エクスプローラー(S) Ctrl+4, Ctrl カーラー(T) Ctrl+4, Ctrl カーラー(T) Ctrl+4, Ctrl かーラー(T) Ctrl+4, Ctrl なてのde Analysis 細 s Code Coverage				

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。

Micro Focus 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 	- 🛛 🔾	> × 🏣 🎘 🗕 🎫 🚺
階層	カバー済み (% ブロック)	カバー済み (ブロック)	未カバー (カバレッジ データをインポートする
😚 節: close-airfile	100 %	1	0 %	0
😚 節: convert-angle	85.71 %	6	14.29 %	1
😚 節: display-airport	100 %	1	0 %	0

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。

	更新日時	種類
VCTutCoverage-001.tcz	2024/08/28 10:54	TCZ ファイル
VCTutCoverage-002.tcz	2024/08/28 11:02	TCZ ファイル

i(N)	: VCTutCoverage-001.tcz ~		カバレッジ ファイル(*.tcz)	~
		l	開く(O) ▼ キャンセル	

カバレッジ情報が更新されます。



Micro Focus Code Coverage				
VCTutCoverage-001.tcz			・ プログラム ビュー	► №
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)
✓		67.86 %	38	32.14 %
▲ 💷 <不明なコレクション>		67.86 %	38	32.14 %
🔺 % プログラム: aircode		65.91 %	29	34.09 %
😚 節: calculate-airport-distance		100 %	5	0 %
⑦ 節: close-airfile		100 %	1	0 %
😚 節: convert-angle		85.71 %	6	14.29 %
😚 節: display-airport		100 %	1	0 %
😚 節: distance-between-airports		100 %	8	0 %
😚 節: find-airport		66.67 %	2	33.33 %
😚 節: get-code-matches		0 %	0	100 %
😚 節: lookup-one-airport		0 %	0	100 %
😚 節: main		62.5 %	5	37.5 %
Micro Focus Unit Testing エラー一覧 出力 Micro	Focus Code Co	verage		

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。



М	
	Micro Focus License Manager
	Micro Focus Visual COBOL
1	ADO.NET Connection Editor
MF	Enterprise Server Administration
	OpenESQL 構成ユーティリティ
G:5_	Visual COBOL コマンドプロンプト (32-bit)
0:5_	Visual COBOL コマンドプロンプト (64-bit)

- 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:¥VCCommandTutorial>set VS_SOLUTION_PATH=c:¥vs_solution_path					
C:¥VCCommandTutorial>cobol %VS_SOLUTION_PATH%¥VCTutCoverage¥aircode.cbl gn					
t,.¥ ANIM TESTCOVER COBIDY(.¥);					
Micro Focus COBOL					
Version 10.0 (C) Copyright 1984-2024 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:¥Users¥tarot¥source¥repos¥VCTutCoverage¥VCTutCoverage¥aircode					
* Data: 1456 Code: 6925 Literals: 400					
C:¥VCCommandTutorial>cobol %VS_SOLUTION_PATH%¥VCTutCoverage¥main.cbl gnt,.					
¥ ANIM TESTCOVER COBIDY(.¥);					
Micro Focus COBOL					
Version 10.0 (C) Copyright 1984-2024 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:¥Users¥tarot¥source¥repos¥VCTutCoverage¥VCTutCoverage¥main					
* Data: 208 Code: 1699 Literals: 192					
C:¥VCCommandTutorial>					



- 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: ¥VCCommandTutorial>set TESTCOVER=. ¥testcoverage.ini

C:¥VCCommandTutorial>set dd_airports=%VS_SOLUTION_PATH%¥VCTutCoverage¥airp orts.dat

C:¥VCCommandTutorial>runw main

C:¥VCCommandTutorial>

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。



2024/08/28 10:55 <DIR> . 2024/08/28 10:52 <DIR> . 2024/08/28 10:52 <DIR> .. 2024/08/28 10:54 1,405 VCTutCoverage-001.tcz 1 個のファイル 1,405 バイト C:¥VCCommandTutorial>

© Rocket Software, Inc. or its affiliates 1990–2024. All rights reserved. Rocket and the Rocket Software logos are registered trademarks of Rocket Software, Inc. Other product and service names might be trademarks of Rocket Software or its affiliates.

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 x +	
\leftarrow	С	① 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	•
	206	perform close-airfile	2
	207	when display-record	-
	208	perform display-airport	_
	209	end-evaluate	
	210	exit program	
	211	•	4
	212		
	213	lookup-one-airport section.	
	214	Initialize Ink-rec	-
	210	move ink-airporti to airport	
	210	if aiment found = 1	
	217	move fames to lokanes	
	210	end-if	
	220		-
	221	•	
	222	distance-between-airports section.	
	223	*> need to find both airports and distance between them	
	224	initialize lnk-distance-result	1
	225	<pre>move lnk-airport1 to airport</pre>	

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:¥VCCommandTutorial¥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:\VCCommandTutorial>tcutil -t 60 data\VCTutCoverage-001.tcz

C:¥VCCommandTutorial>

しきい値に達していない場合)

tcutil -t 70 data¥VCTutCoverage-001.tcz

C: ¥VCCommandTutorial>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:¥VCCommandTutorial>

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:¥VCCommandTutorial>tcutil -t 60 data¥VCTutCoverage-001.tcz

C:¥VCCommandTutorial>echo %ERRORLEVEL%

0

C:¥VCCommandTutorial>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:¥VCCommandTutorial>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、著作権法の精神に基づき、適切な扱いを行ってください。