

Visual COBOL チュートリアル

Eclipse – 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、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。

2 前提

- 本チュートリアルで使用したマシン OS : Windows 10
- Visual COBOL 10.0 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、任意のフォルダー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 Eclipse の起動
 - 3.1.2 チュートリアルプロジェクトのインポート
 - 3.1.3 カバレッジを有効にしたアプリケーションの実行
 - 3.1.4 複数カバレッジ情報のマージ
 - 既存カバレッジ情報ファイルの読み込み
 - 3.1.5 カバレッジ情報のエクスポート
- 3.2 コマンドラインからの実行
- 3.3 ユーティリティ機能の紹介
 - 3.3.1 カバレッジ情報ファイルの HTML 形式でのレポート出力
 - 3.3.2 カバレッジ率の評価

Rocket software

3 チュートリアル手順

3.1 IDE からの実行

3.1.1 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。

ディレクトリーをワークスペースとして選択

Eclipse は、ワークスペースディレクトリを使用して、環境設定と開発成果物を保存します。

c:¥workspace_tut_coverage	\sim	参照(<u>B</u>)
□この選択をデフォルトとして使用し、今後この質問を表示しない(U)		
▶ 最近のワークスペース(<u>R</u>)		
	起動(<u>L</u>) キ	ャンセル

3.1.2 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。

ファイ	イル(F) 編集(E) リファクタリング	ナビゲート(N) 検索 プロシ
_	新規(N) ファイルを開く(.) ファイル・システムからプロジェクトを問	Alt+シフト+N > 墹<
	Recent Files	>
	閉じる(C)	Ctrl+W
	すべて閉じる(L)	Ctrl+シフト+W
	保存(S) 名前を付けて保存(A)	Ctrl+S
¢	すべて保存(E) 前回保存した状態に戻す(T)	Ctrl+シフト+S
	移動(V)	
P	名前を変更(M)	F2
8	更新(F)	F5
	行区切り文字の変換(D)	>
Ð	印刷(P)	Ctrl+P
è	インポート(I)	
4	エクスボート(O)	

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。



45.18.1.18.14.19.100.1			
インボート・ウイザードの選択(<u>S</u>):			
、 アーカイブ・ファイル			
C ファイル・システム	イゴナナのゴロジェクト		
□ フォルターまたはアーカ	17日来の7日919下 -クスペースへ		
🗔 設定			
> 🗁 EJB			
> 🗁 Gradle			
2	· 豆ろ(R) 次(N) > 約7	7 (E)	キャンク
		(<u>r</u>)	7170
以下の設定を行い、[終了([F)] をクリックします。		
ルート・ディレクトリーの	選択(T)		
[参照(R)] をクリックし	,て、ダウンロードしたサンプルファイルを	展開した	フォルダーを
プロジェクトをワークスペ	ペースにコピー		
チェックをつける			
チェックをつける			
チェックをつける プロシェクトをインポート			
チェックをつける プロシェクトをインポート 既存の Eclipse プロジェクトを検察	索するディレクトリーを選択します。		
チェックをつける プロシェクトをインポート 既存の Eclipse プロジェクトを検察 ・・ディレクトリーの選択(T):	をするディレクトリーを選択します。 「C:¥vc-coverage	~	参照(R)
チェックをつける プロシェクトをインボート 既存の Eclipse プロジェクトを検察 ・ルート・ディレクトリーの選択(L): ・ アーカイブ・ファイルの選択(A):	索するディレクトリーを選択します。 C:¥vc-coverage	~	参照(<u>R</u>) 参照(R)
チェックをつける プロシェクトをインポート 既存の Eclipse プロジェクトを検察 ・ ・ ルート・ディレクトリーの選択(I): ・ アーカイブ・ファイルの選択(<u>A</u>): プロジェクト(P)・	索するディレクトリーを選択します。 C:¥vc-coverage	~	参照(<u>R</u>)
チェックをつける プロシェクトをインボート 既存の Eclipse プロジェクトを検察 ④ ルート・ディレクトリーの選択(L): 〇 アーカイブ・ファイルの選択(A): プロジェクト(P):	をするディレクトリーを選択します。 C:¥vc-coverage	~ ~	参照(B) 参照(R)
チェックをつける プロシェクトをインポート 既存の Eclipse プロジェクトを検す ● ルート・ディレクトリーの選択(I): ○ アーカイブ・ファイルの選択(A): プロジェクト(P): ☑ AirportDemoTutCovera	なするディレクトリーを選択します。 C:¥vc-coverage age(C:¥vc-coverage)		参照(R) 参照(R) すべて選択(S)
チェックをつける プロシェクトをインボート 既存の Eclipse プロジェクトを検察 ● ルート・ディレクトリーの選択(<u>L</u>): ○ アーカイブ・ファイルの選択(<u>A</u>): プロジェクト(<u>P</u>): ☑ AirportDemoTutCovera	索するディレクトリーを選択します。 C:¥vc-coverage age(C:¥vc-coverage)	×	参照(B) 参照(R) すべて選択(<u>S)</u> 訳をすべて解防 東充(て)
チェックをつける プロシェクトをインポート 既存の Eclipse プロジェクトを検す ● ルート・ディレクトリーの選択(I): ○ アーカイブ・ファイルの選択(A): プロジェクト(P): ☑ AirportDemoTutCovera	なするディレクトリーを選択します。 C:¥vc-coverage age(C:¥vc-coverage)		参照(R) 参照(R) すべて選択(<u>S</u>) 訳をすべて解除 更新(<u>E</u>)
チェックをつける プロシェクトをインボート 既存の Eclipse プロジェクトを検察 ルート・ディレクトリーの選択(L): アーカイブ・ファイルの選択(A): プロジェクト(P): AirportDemoTutCovera 	索するディレクトリーを選択します。 C:¥vc-coverage age(C:¥vc-coverage)	× ×	参照(B) 参照(R) すべて選択(<u>S</u>) 訳をすべて解除 更新(E)
チェックをつける プロシェクトをインポート 既存の Eclipse プロジェクトを検す ● ルート・ディレクトリーの選択(I): ○ アーカイブ・ファイルの選択(A): プロジェクト(P): ☑ AirportDemoTutCovera	をするディレクトリーを選択します。 C:¥vc-coverage age(C:¥vc-coverage)	v V	参照(R) 参照(R) すべて選択(<u>S</u> 訳をすべて解附 更新(E)
チェックをつける プロシェクトをインボート 既存の Eclipse プロジェクトを検察 ルート・ディレクトリーの選択(L): アーカイブ・ファイルの選択(<u>A</u>): プロジェクト(<u>P</u>): ✓ AirportDemoTutCovera 	索するディレクトリーを選択します。 C:¥vc-coverage age(C:¥vc-coverage)	· · · · · · · · · · · · · · · · · · ·	参照(B) 参照(R) すべて選択(<u>S</u>) 訳をすべて解除 更新(E)
チェックをつける プロシェクトをインポート 既存の Eclipse プロジェクトを検索 ● ルート・ディレクトリーの選択(I): ○ アーカイブ・ファイルの選択(A): プロジェクト(P): ☑ AirportDemoTutCovera	索するディレクトリーを選択します。 C:¥vc-coverage age(C:¥vc-coverage)	v W	参照(R) 参照(R) すべて選択(<u>S</u> 訳をすべて解時 更新(E)
チェックをつける プロシェクトをインポート 既存の Eclipse プロジェクトを検す ● ルート・ディレクトリーの選択(L): ○ アーカイブ・ファイルの選択(A): プロジェクト(P): ☑ AirportDemoTutCovera オプション □ ネストしたプロジェクトを検索(H	索するディレクトリーを選択します。 C:¥vc-coverage age(C:¥vc-coverage)	× ×	参照(B) 参照(R) すべて選択(<u>S</u>) 訳 をすべて解 り 更新(E)
 チェックをつける プロシェクトをインポート 既存の Eclipse プロジェクトを検索 ルート・ディレクトリーの選択(Δ): アーカイブ・ファイルの選択(Δ): プロジェクト(P): ✓ AirportDemoTutCovera オブション ネストしたプロジェクトを検索(ト ✓ プロジェクトをワークスペースにコ 	茶するディレクトリーを選択します。 C:¥vc-coverage age(C:¥vc-coverage)	v J	参照(R) 参照(R) すべて選択(<u>S</u> 訳をすべて解附 更新(E)
チェックをつける プロシェクトをインポート 既存の Eclipse プロジェクトを検索 ● ルート・ディレクトリーの選択(L): ○ アーカイブ・ファイルの選択(A): プロジェクト(P): ☑ AirportDemoTutCovera オプション □ ネストしたプロジェクトを検索(H ☑ プロジェクトをワークスペースにコ □ 完了次第、新しくインポートし; □ ワークスペースに可じたなすまる	をするディレクトリーを選択します。 C:¥vc-coverage ige(C:¥vc-coverage) ige(C:¥vc-coverage) i) i/)		参照(<u>R</u>) 参照(<u>R</u>) すべて選択(<u>S</u>) 現をすべて解除 更新(<u>E</u>)
 チェックをつける プロシェクトをインポート 既存の Eclipse プロジェクトを検索 ルート・ディレクトリーの選択(Δ): アーカイブ・ファイルの選択(Δ): プロジェクト(P): ✓ AirportDemoTutCovera オプション ネストしたプロジェクトを検索(ト ゾブロジェクトをワークスペースにご 完了次第、新しくインボートし; ワークスペースに既に存在する; ワーキング・セット 	をするディレクトリーを選択します。 C:¥vc-coverage age(C:¥vc-coverage) 1) ビー(C) たプロジェクトを閉じる(の) プロジェクトを隠す(i)		参照(R) 参照(R) すべて選択(<u>S</u> 訳をすべて解閉 更新(E)
チェックをつける プロシェクトをインポート 既存の Eclipse プロジェクトを検索 ● ルート・ディレクトリーの選択(L): ○ アーカイブ・ファイルの選択(A): プロジェクト(P): ☑ AirportDemoTutCovera オプション □ ネストしたプロジェクトを検索(ト ☑ プロジェクトをワークスペースにコ □ 完了次第、新しくインポートし; □ ワークスペースに既に存在する; ワーキング・セット □ ワーキング・セット	をするディレクトリーを選択します。 C:¥vc-coverage age(C:¥vc-coverage) hge(C:¥vc-coverage) hge(C:¥vc-coverage)		参照(B) 参照(R) すべて選択(<u>S</u>) 訳をすべて解除 更新(E)
 チェックをつける プロシェクトをインポート 既存の Eclipse プロジェクトを検索 ● ルート・ディレクトリーの選択(Δ): ○ アーカイブ・ファイルの選択(Δ): プロジェクト(P): ✓ AirportDemoTutCovera オブション ネストしたプロジェクトを検索(ト ジ プロジェクトをワークスペースにご 完了次第、新しくインボートし; ワークスペースに既に存在する; ワーキング・セットにプロジェクト ワーキング・セットにプロジェクト 	を追加(①)		参照(R)… 参照(R)… すべて選択(<u>S</u> 択をすべて解財 更新(<u>E</u>) 新規(<u>W</u>)…



以下のダイアログが表示された場合は、そのまま [OK] をクリックします。





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 リックにてコンテクストメニューを表示し、[プロパティ(R)]をクリックします。

~	検証	
	チーム(E)	>
	比較対象(A)	>
	構成	>
	ソース(S)	>
	プロパティ(R)	Alt+Enter

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 "true" に変更したうえで、[適用(L)] をクリックします。



YHルタ入力 > リソース	COBOL	¢	▼ ⇔ ▼
Coverage			
 Micro Focus 			
ビルダー			
ビルド パス	設定	値	^
> ビルド構成	▶ 一般		
✓ プロジェクト設定	文字セット	ASCII	
> COBOL	ソース エンコーディング		
コンテナー	COBOL 方言	Micro Focus	
ビルド環境	ソース フォーマット	固定	
指令の確定	デバッグ用にコンパイル	はい	
> 実行時構成	EXIT PROGRAM を GOBACK として処理	ANSI	
WikiText	詳細	いいえ	
サーバー	.GNT にコンパイル	いいえ	
タスク・タグ	✓ 出力		
タスク・リポジトリー	指令ファイルを生成する	いいえ	
ビルダー	リストファイルを生成	1.11.1.2	
プロジェクト・ネーチャー	コード カバレッジを有効にする	true	
プロジェクト・ファヤット	プロファイラを有効にする		
プロジェクト参照	✓ Iラ-/警告		
検証	警告レベル	回復可能なエラーを含める(レベル E)	
実行/デバッグ設定	コード カバレッジを有効にする カバレッジ分析をおこなうためにソースをコンパイルし COBOL コンパイル設定: CHARSET*ASCII* DIALECT*MF* SOURCEFOR WARNING*1* MAX-ERROR*100*	ます。カバレッジ分析により、どのコードが実行されたかを表示できます MAT"fixed" NOLIST anim EXITPROGRAM"ANSI" TESTCOVE	- ~
		デフォルトの復元(T)	適用(L)

 6) 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。

Coverage	New Configuration [使用中]		~ 構成の管理
Micro Focus			
ビルダー			
ビルド パス	フィルタテキストを入力		
∨ ビルド構成	· 2.4 定	信	^
> COBOL	ax / Linkago	I <u>P</u>	
イベント	◆ Linkage 山力の々前	AirportDomo	
ディプロイ	山力の石削	New Configuration bin	
ビルド環境		main	
> リンク	エノトリホイノト	main 単一史行可能ファイル	
✓ フロジェクト設定		半一天11可能ノバ1ル 22 ビット	
> COBOL	$L \mathcal{D} \Gamma \mathcal{B} \mathcal{K}$	52 L 9 P	
コンテナー	COROL 以外のアプリケーションから呼びかり 可能	11/17	
ビルド環境		1117	
指令の確定	Z//	(#1)	
> 実行時構成	「「「「「「」」」	出右	
WikiText	現在の実行時システムだけにバインドする	/1/17	
サーバー	出力の種類		
タスク・タグ	ターゲットオペレーティング システム	Windows	
タスク・リポジトリー	詳細		
ビルダー		11/13	
プロジェクト・ネーチャー	未定義シンボルでTラー	1.1.1.1.2	
プロジェクト・ファセット	エバルリポノバー アビーフカ陸アパイナ	1117	~
フロジェクト参照 検証 実行/デバッグ設定	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()

3.1.3 カバレッジを有効にしたアプリケーションの実行

 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クストメ ニューを表示し、[実行(R)] > [実行の構成(N)] を選択します。

😤 COB 🗙 陷 プロ 🛛 😤 Appl	<u>_</u>	サーバ 🛄 Anal 🗖 🗖		
		新規作成(N)	>	
✓ ¹ × ¹ ✓ ¹ ×		表示方法(W)	Alt+シフト+W >	
> aircode.cbl		כול -	Ctrl+C	
> 🖻 main.cbl	Ē	貼り付け	Ctrl+V	
∽ 🔑 コピーファイル	₩	削除(D) コンテキストから除去	削除 Ctrl+Alt+シフト+下	
airparams.cpy		移動(V)	Currait	
> 🗁 New_Configuration.bin		名前を変更(M)	F2	
airports.dat		ビルド アクション	>	
🖹 readme.xml		タスクのスキャン		
		指令の確定		
		コート分析	>	
	1	インボート(i) エクフポート(O)	>	
🏗 アウトライン × 🔤 プログラム アウト	5	五头(F)	E5	
고리! = /\;+!=/#+?고ስ= /『*\포ゔ /	٢	プロジェクトを閉じる(S)	15	
アウトフィンを提供するアクティノなエティ		無関係なプロジェクトを閉じる(U)		
		Source	>	
		リモートシステムビューで表示		タスク 🔲 プロパティー 🏗 Table I
	Q	Coverage As	>	
	U	実行(R)	>	実行 の構成(N)

 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クストメニューを表示し、[新規構成 (W)]を選択します。



3) [名前(N)」に "AirportDemoTutCoverage" を入力した上で、[実行(R)] をクリックします。



名前(N) AirportDemoTutCoverage	
 ▼ COBOL プロジェクト(P) 	^
AirportDemoTutCoverage 参照	
▶ 接続プロパティ	
▼ 主プログラム	
✓ プログラムはプロジェクトビルド構成の一部: New Configuration ∨	
New_Configuration.bin/AirportDemo.exe 参照	
▼ 開始オプション	
コマンド行引数:	
\sim	
作業ディレクトリ:	~
<u> 前</u> 回保管した状態に戻す(V) 適用(Y)	
実行(R) 閉じる	

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)

	ov. Ai	rport	Demo									
1	Enter	an	airport	code,	or two	o 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。 実行(R) ウィンドウ(W) ヘルプ(H)

×11(1)					
=> ← ▼	新規ウィンドウ(N) エディター 外観	> >			
	ビューの表示(V)	>	ا دور م	Application Explorer	
	ナビゲーション(G)	>	ar G	Filter Definitions Micro Focus Unit Testing	
	設定(P)			Table Results アウトライン	Alt+>/76+00
				コード カバレッジ コビーファイル従属関係	

現時点では、カバレッジ情報が何も表示されていません。

שרעב 🗐	🔝 問題	<u>ହ</u> 922	🔲 プロパティー	12 Table Results	s 🛛 Filter Definiti	ons 🔄 Micro Foci	us Unit Testing 😫	コード カバレッジ ×
m =		^		+ 101	もじ 次っゴロック	ナナバ ブロック		
安东				ガハレッシ	カハー済みノロック	木カハー ノロック	ノロック数	



```
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。



 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)] をクリックします。

構成の作成、管理、および実行 COBOL プログラムを実行します		
 ○ ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	 名前(N): [AirportDemoTutCoverage 名前(N): [AirportDemoTutCoverage □ - 除 futCoverage □ - R futCoverage<th>- - - - - - - - - - - - - - - - - - -</th>	- - - - - - - - - - - - - - - - - - -
	→ 前回保管した状態に戻す(V) 適用(Y)
?	実行(R)	閉じる

8) 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。

🖳 コンソール 🖹 問題 🧔 タスク 🔲 プロパティー	💼 Table Results	🕯 🕅 Filter Definitio	ons 🔄 Micro Focus	s Unit Testing 📴	コード カバレッジ ×
AirportDemoTutCoverage 07 8月 2024 15.31.	22				
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	
✓ I AirportDemoTutCoverage	67.9 %	38	18	56	
✓	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	
Iookup-one-airport	0.0 %	0	3	3	
main	62.5 %	5	3	8	
 open-airfile 	50.0 %	1	1	2	
🗸 🕑 main	75.0 %	9	3	12	
Procedure Division	72.7 %	8	3	11	
 call-aircode-program 	100.0 %	1	0	1	

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。



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。

3.1.4 複数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。

1) 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)



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。



🖳 コンソール 🔝 問題 🧔 タスク 🔲 プロパティー	🗈 Table Results	🗟 Filter Definitio	ons 🔄 Micro Focu	s Unit Testing 😫	コード カバレッジ ×
AirportDemolutCoverage 07 8/ 2024 15.41.4	+1				
	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 	42.9 %	24	32	56	
✓ ♀ 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	
Iookup-one-airport	100.0 %	3	0	3	
 main 	75.0 %	6	2	8	

2) 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。

💷 コン 🖹 問題 🗐 タスク 🔲 プロ 🏦 Tab	🗗 Filt 🗔	Micr 📴 🗆 – 🔅	× 🖳 I 🖳 ē	[接 5
		V 🖪 🗋 🗠 👍	🕑 🖻 🔍 🗙 💥	1 🕆 🖪 י
AirportDemoTutCoverage 07 8月 2024 15.41.4	1			
要素	カバレッジ	カバー済みブロック	未カバー ブロック	-
✓ Isoto v State v	42.9 %	24	32	
✓ ♀ aircode	31.8 %	14	30	
calculate-airport-distance	0.0 %	0	5	
 close-airfile 	100.0 %	1	0	
 convert-angle 	0.0 %	0	7	

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。

新しい項目名 TutMerge	
マージするカバレッジの履歴項目を選択:	
☑ AirportDemoTutCoverage 07 8月 2024 15.41.41	すべて選択
AirportDemoTutCoverage 07 8月 2024 15.31.22	「べて選択を解除
? Ок	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。





🗟 コンソール 🖹 問題 🖉 タスク 🔲 プロパティー	Table Results	🛛 🕶 Filter Definitio	ns 🔄 Micro Focus	Unit Testing 🔒	コード カバレッジ ×
TutMerge					
	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	82.1 %	46	10	56	
✓	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	
Iookup-one-airport	100.0 %	3	0	3	
main	87.5 %	7	1	8	
 open-airfile 	50.0 %	1	1	2	
> 🕑 main	100.0 %	12	0	12	

© 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.



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。 本手順では、後述する 3.2 のカバレッジ結果を読み込みます。

 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。



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。

🖳 コンソール 🖹 問題 🧧 タスク 🔲 プロパティー	記 Table Results	🛙 🕶 Filter Definitio	ons 📴 Micro Focu	s Unit Testing 😫	コードカバレッジ 🗙 🖳 コード検索結果 🚨 直接参照
TutMerge					
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 	82.1 %	46	10	56	
 B 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	

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対象のカバレッジ情報ファイルのものに更新されます。



💷 コンソール 🖹 問題 🧔 タスク 🔲 プロパティー	able Result	s 🔊 Filter Definiti	ons 📴 Micro Focu	us Unit Testing 🔋	コード カバレッジ ×
VCTutCoverage-001 toz 07 8 E 2024 16 10 4	,				
Venucoverage=001.tcz 07 8/3 2024 10.10.4/					
~	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	
> 🕒 main	75.0 %	9	3	12	

3.1.5 カバレッジ情報のエクスポート

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を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。
- 2) 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。

, 😰 I-	ド カバレッジ ×	🖪 コード検索結果	📲 直接参照	照の結果を取	又得
		v 🖡 🗋	20	2 🔍 🛪	۴ 🙀
ク数					
56					

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、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。

- コマンドライン実行用のフォルダーを用意します。
 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 フォルダー配下に 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 (32-bit)] を選択 します。



	М
	Micro Focus License Manager 🛛 🗸
8	Micro Focus Visual COBOL ^
Ľ	Enterprise Server Administration
	GpenESQL 構成ユーティリティ
	Visual COBOL for Eclipse
ŝ	Visual COBOL コマンドプロンプト (32
Ф	■ Visual COBOL コマンドプロンプト (64
	P 📄 🐵

- 4) さきほど用意した作業フォルダーに移動します。
 C:¥Users¥tarot¥Documents>cd ¥VCCommandTutorial
 C:¥VCCommandTutorial>
- 5) 以下のコマンドを実行します。
 - set ECLIPSE_WORKSPACE=c:\u00e4workspace_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:\vorkspace tut coverage
C:\VCcommandTutorial>cobol %ECLIPSE_WORKSPACE%\AirportDemoTutCoverage\aircode.c
bl 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: workspace_tut_coverage
 Data:
             1824
                      Code:
                                  5272
                                          Literals:
                                                         771
C: ¥VCCommandTutorial > cobol % ECLIPSE_WORKSPACE% ¥ AirportDemoTutCoverage ¥ 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:\u00e4workspace_tut_coverage\u00e4AirportDemoTutCoverage\u00e4main
  Data:
              576
                      Code:
                                 1328
                                          Literals:
                                                         320
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=%ECLIPSE_WORKSPACE%¥AirportDemoTutCoverage¥airports.dat



runw main.gnt

- 以下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



testcoverage.ini の RESULT 項目で指定した data フォルダー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。



3.3 ユーティリティ機能の紹介

3.3.1 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)

C:¥VCCommandTutorial>tcreport data¥VCTutCoverage-001.tcz browse html reportloc(o ut)

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

C:¥VCCommandTutorial>

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。



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、0 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

3.3.2 カバレッジ率の評価

1) 以下のコマンドを実行します。

tcutil -t <しきい値> カバレッジファイルへのパス

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。