

Visual COBOL チュートリアル

1

Eclipse - ネイティブ COBOL プログラムの単体テスト

1 目的

本チュートリアルでは、ネイティブ COBOL プログラムに対するテスト作成、実行方法、および、テスト結果を表示させる方 法の習得を目的としています。

MFUnit は、Visual COBOL に搭載された xUnit 系の単体テストフレームワークです。xUnit はオブジェクト指向型の 単体テストフレームワーク SUnit に起源を持つ JUnit や RUnit 等の単体テストフレームワークの総称です。MFUnit は xUnit の設計アーキテクチャーや仕組みは取り入れつつも COBOL 開発者にとって扱いやすい手続き型の COBOL を対象とした単体テストフレームワークという設計思想の下、開発されました。

MFUnit は COBOL 開発作業に以下の利点を提供します。

- テストを繰返し実行させることができるため、修正作業時などのテスト工数の削減が見込める
- Jenkins などの継続的インテグレーション (Continuous Integration) ツールと連携によりテストの自動化が行え、DevOps サイクルの導入が足がかりを作れる

2 前提条件

- 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

本資料は、ネイティブ COBOL に対する単体テストフレームワークの利用方法を記載したチュートリアルです。JVM COBOL の単体テスト実現方法については、別チュートリアルを参照ください。

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。 サンプルファイルのダウンロード



内容

- 1 目的
- 2 前提条件
- 3 チュートリアルについて
- 3.1 IDE からの実行
 - 3.1.1 前準備
 - 3.1.2 基本的なテスト
 - 3.1.3 データ駆動型テスト
 - 3.1.4 自己完結型テスト
- 3.2 コマンドラインからの実行



3 チュートリアルについて

MFUnit は、単一処理の結果を判定する基本的なテストプログラムからデータ駆動型、自己完結型といった複数のテストタイプを用意しており、順に説明していきます。特定のテストタイプのみを実施したい場合においても、「3.1.1 前準備」は先に実施してください。

3.1 IDE からの実行

3.1.1 前準備

3.1.1.1 Eclipse IDE 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)] 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_native_mfunit	~	参照(B)
_		
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(R)		
	起動(L) キ	ヤンセル

3.1.1.2 チュートリアルプロジェクトのインポート

1) Eclipse IDE メニューより、[ファイル(F)] > [インポート(I)] を選択してください。

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	プロジ
	新規(ファイ) ファイ) Recen	N) レを開く(.) レ・システム; nt Files	からプロジェクトを	AI 開<	t+シフト	+N >
	閉じる すべて	o(C) 閉じる(L)		Ctrl	Ctrl・ +シフト・	+W +W
	保存(名前 すべて 前回(S) を付けて保ィ 保存(E) 呆存した状	字(A) 態に戻す(T)	Ct	Ctr rl+シフト	I+S `+S
2 2	移動(名前教 更新(行区t	V) E変更(M) F) のり文字の	 変換(D)			F2 F5 >
Ð	印刷(P)			Ctr	I+P
è	インポ	ート(I)				
4	エクス	ポート(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) >] をクリックします。



インポート・ウィザート	の選択(S):			
フィルタ入力				
 ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	ブ・ファイル システム -またはアーカイブ由来 ジェクトをワークスペー	ŧのプロジェクト スへ		

3) 以下の入力を行い、[終了(F)] をクリックします。

ルートディレクトリの選択:

[参照(R)] をクリックし、サンプルファイルを展開したフォルダー内の AirportDemoMFUnit フォルダーを指定

プロジェクトをワークスペースにコピー: 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するディレクトリーを選択します。	
● ルート・ディレクトリーの選択(T) C:¥vc-elnativetestframework	✓ 参照(R)
○ アーカイブ・ファイルの選択(A):	~ 参照(R)
プロジェクト(P):	
AirportDemoTutNativeMFUnit(C:¥vc-eInativetestframework¥AirportDer	すべて選択(S)
	選択をすべて解除(D)
	更新(E)
< >	
オプション	
□ ネストしたプロジェクトを検索(H)	
✓ ノロシェクトをリークスペースにコヒー(C) □ 完了次第 新しくインボートしたプロジェクトを閉じる(o)	
□ ワークスペースに既に存在するプロジェクトを隠す(i)	
ワーキング・セット	
□ ワーキング・セットにプロジェクトを追加(T)	新規(W)
	選択(E)
	, <u> </u>
? < 戻る(B) 次へ(N) > 終了(F)	キャンセル



🔓 COB × 눱 プロ 🐁 Appl 🥌 サーバ 💻 Anal
✓ I AirportDemoTutNativeMFUnit
✓ 個 COBOL プログラム
> 🖻 aircode.cbl
> 🖻 main.cbl
∽ 🔑 コピーファイル
airparams.cpy
airrec.cpy
🗸 🗁 New_Configuration.bin
aircode.idy
🗎 aircode.obj
📄 aircode.obj.1.tlog
🗎 aircode.objlist
AirportDemo.exe
🗖 main.idy
📄 main.obj
main.obj.1.tlog
🖹 main.objlist
airports.dat
🖹 readme.xml

AirportDemoTutNativeMFUnit プロジェクトが作成されます。

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。

4) 単体テストを行なうために、出力形式を int 形式に変更します。以下の手順を実行してください。

AirportDemoTutNativeMFUnit 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、[プロパティ(R)]を選択します。

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

ツリーメニューより、[Micro Focus] > [ビルド構成] > [リンク] を選択し、「ターゲットの種類」を"すべて INT/GNT ファイル"に変更した上で、[適用して閉じる] をクリックします。



フィルタ入力	リンク	⇔ ▼ ⇔ ▼ 8
> リソース ^	New Configuration (在中中)	
Coverage	New Configuration [[史用中]	◇ (将风の旨理
✓ 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	設定	值 ^
> COBOL	Linkage	-
イベント	+ Linkage 出力の友益	(imatDama
ディプロイ	4.5.67	New Configuration bin
ビルド環境		New Conliguration.bin
> リンク	12597775	
> プロジェクト設定	ターゲットの種類	すべて INT/GNT ファイル
指会の確定	ビット数	64 Ľツト
「日口の確定」	IDD I- Part Still	1112
/ 天门时俩成	ターゲットの種類	
WIKIIEXt	作成する出力ノアイルの種類、および単一ノアイノ	レを作成するか、ソームことに1個のノアイルを作成するかを指定します
サーハー	(必須)	
タスク・タグ		•
> タスク・リポジトリー		
ビルダー 🗸 🗸 🗸		デコナルトの復元(エ) (東田(1)
< >		アフォルドの復元(1) 通用(L)
?		適用して閉じる キャンセル

注意)

上記画面では、「ビット数」に "64 bit" を指定しています。 "32 bit" 指定も可能ですが、その場合、以降の 手順でも "32 bit" を選択してください。

3.1.2 基本的なテスト

この方式では、1つのテストを1つのテストブロックで記述していきます。

3.1.2.1 MFUnit テストの作成

1) Eclipse IDE メニューより、[ファイル(F)] > [新規(N)] > [COBOL ユニットテストプロジェクト] を選択します。

ファ	イル(F)	編集(E)	リファクタリング	ナビゲート(N)	検索	プロジ	ェクト	·(P) 実行(R)	ウィンドウ(W)	ヘルプ(H)
	新規(N)		Al	t+シフト	+N >	鬯	COBOL プロジ	IDF	
	ファイルを開く(.)						2	COBOL コピーファイル プロジェクト		
	ファイル・システムからプロジェクトを開く						ê	リモート COBOL プロジェクト		
	Recent Files >				>	ê	🗯 リモート COBOL コピーファイル プロジェクト			
						알	COBOL/Java	相互運用機能	のプロジェクト	
	閉じる	(C)			Ctrl	+W	€Û	COBOL ユニッ	ト テスト プロジェ	<u> </u>
	すべて	閉じる(L)		Ctrl	+シフト	+W	B	COBOL JVM ;	フロジェクト	
	保存(S)			Ctr	I+S	R	リモート COBO	L ユニット テスト	プロジェクト

 プロジェクト名」に "AirportDemoTutNativeMFUnitTest" を入力し、実行環境に合わせたプロジェクトテン プレートを選択した上で、[終了(F)] をクリックします。



COBOL ユニット テスト プロジェクト			7
ワークスペースに COBOL ユニット テスト ブロシ	エクトを新規作成しま	す。	\sim
プロジェクト名(P): AirportDemoTutNative	MFUnitTest		
プロジェクト テンプレートを選択			
 ➢ Micro Focus テンプレート [32 ビット] ➢ Micro Focus テンプレート [64 ビット] 			
		=>>-	1. トの乳中た#
□ テンプレートの参昭		<u>))</u>	/レートの設定を供
場所:			参照
ファイルシステムを選択: default 、	·		
✓ デフォルト・ロケーションの使用(D)			
ロケーション(L): C:¥workspace_tut_native_	mfunit¥AirportDem	oTutNativeMFUnitTe	。参照(R)
ファイル・システムを選択(Y):	デフォルト 〜		
0		終了(F)	キャンセル
主意)			

プロジェクトが作成されます。 COB... × C プロ... ? Appl... ... ? * COB... × C プロ... ? Appl... ... ? * COB... × C プロ... ? * COB.C プログラム > @ aircode.cbl > @ main.cbl * DL-ファイル @ airparams.cpy @ airrec.cpy > > New_Configuration.bin @ airports.dat X readme.xml > C AirportDemoTutNativeMFUnitTest

3) AirportDemoTutNativeMFUnitTest プロジェクトを選択した上で、Eclipse IDE メニューより、[ファイル(F)]
 > [新規(N)] > [COBOL ユニットテスト] を選択します。

7



ファ	イル(F) 編集(E) リファクタリング ナ	└ビゲート(N) 検索 プロ:	ジェク	ト(P) 実行(R) ウィンドウ(W) ヘルプ(H)
	新規(N)	Alt+シフト+N >	ø	COBOL プロジェクト
	ファイルを開く(.) ファイル・システムからプロジェクトを開 Recent Files	< >	1000 1000 1000 1000 1000 1000 1000 100	COBOL コピーファイル プロジェクト リモート COBOL プロジェクト リモート COBOL コピーファイル プロジェクト COBOL Jean 把写演用機能のプロジェクト
	閉じる(C) すべて閉じる(L)	Ctrl+W Ctrl+シフト+W	9 10 10 10 10 10 10 10 10 10 10 10 10 10	COBOL フニット テスト プロジェクト COBOL ユニット テスト プロジェクト COBOL JVM プロジェクト
	保存(S) 名前を付けて保存(A) すべて保存(E) 前回保存した状態に戻す(T)	Ctrl+S Ctrl+シフト+S		リモート COBOL ユニット テスト プロジェクト COBOL JVM ユニット テスト プロジェクト リモート COBOL JVM プロジェクト プロジェクト(R)
1	移動(V) 名前を変更(M) 更新(F) 行区切り文字の変換(D)	F2 F5 >		Enterprise Server Common Web Administration 接続 Directory Server 接続 フォルダー ファイル
۵	印刷(P)	Ctrl+P		COBOL フロクラム COBOL コピーファイル
	インポート(I) エクスポート(O)			スタンドアロン ファイル リモート スタンドアロン ファイル
	プロパティ(R)	Alt+Enter		Enterprise Server
	ワークスペースの切り替え(W)	>	eñ Cô	

4) [COBOL ユニット テストの新規作成] ウィンドウが表示されるので [プログラム ユニット テスト] 項目を選択し、

COBOL ユニット テスト 作成するテストのタイプを選択してください… プログラム ユニット テスト プログラムを直接呼び出して、入力と出力をアサートできるようにします。CSV ファイルから読み込んだデータを使用して、プログラムを繰り返し実行できます。
 自己完結型ユニット テスト Micro Focus Unit Test Preprocessor を使用してテスト ケースを既存のソースコードにコンパイルし、セクションと段落を直接テストできるようにします。

5) 以下の項目を入力し、[終了(F)] をクリックします。

[次へ(N)] をクリックします。

- [プログラムのユニットテストを作成する]を選択
- [参照] をクリックして、AirportDemoTutNativeMFUnitTest プロジェクト内の「aircode.cbl」を選択



COBOL ユニット テスト エディタで開くことができる COBOL ユニット テスト ファイルを新規	作成します。			D
含まれるプロジェクト: AirportDemoTutNativeMFUnitTest				参照
新規ファイル名: TestProgram1.cbl				
 プログラムのユニット テストを作成する テスト対象のプログラム: AirportDemoTutNativeMFUnit/air 	code.cbl			参照
○ テンプレートからユニット テストを作成する				
テンプレートを選択:				
📄 Micro Focus テンプレート				
			<u>テンプレ</u>	ートの設定を構成
□ テンブレートの参照 場所: ファイルシステムを選択: default ~				参照
?	< 戻る(B)	次へ(N) >	終了(F)	キャンセル

単体テストプログラムが作成されたことを確認します。

🔓 COB	× 陷 プロ	😤 Appl	🔜 サーバ			
> 😕 Airpe	ortDemoTutN	lativeMFUni	it			
🗸 🔡 Airpo	ortDemoTutN	lativeMFUni	itTest			
✓ 個 COBOL プログラム						
> 🧕 TestProgram1.cbl						
> 🗁 New_Configuration.bin						

6) テストプログラムを確認します。

AirportDemoTutNativeMFUnitTest プロジェクト内の「TestProgram1.cbl」をダブルクリックして、コード を表示します。



TestProgram1.cbl ×
TestProgram1.cbl
•···•*A·1·B··•···2····•3····•4····•5···
*> Test Fixture for aircode, aircode
<pre>copy "mfunit_prototypes.cpy".</pre>
program-id. TestProgram1.
working-storage section.
COPY MTUNIT.CPY . 78 TEST TESTATRCODE value "TestATRCODE"
01 np procedure-pointer.
of bb browning between
*> Program linkage data
01 lnk-function PIC X.
88 get-matches value '1'.
88 get-distance value '2'.
88 get-details value '3'.
88 close-file value 4.
88 display-record value '6'.
01 lnk-airport1 PIC X(3).
01 lnk-airport2 PIC X(3).
01 lnk-prefix-text PIC X(3).

以下のコードが記述されていることが分かります。

- entry MFU-TC-PREFIX & TEST-TESTAIRCODE
- entry MFU-TC-SETUP-PREFIX & TEST-TESTAIRCODE

MFUnit では、テストを下記のように決められた手順で実行しています。

テスト名 test1 を実行する場合、以下の順序で動作します。

- ① entry MFU-TC-SETUP-PREFIX & "test1"
- ② entry MFU-TC-PREFIX & "test1"
- ③ entry MFU-TC-TEARDOWN-PREFIX & "test1"
- ④ 次のテストを実行・・・

MFU-TC-SETUP-PREFIX で始まる entry にて、テストの前処理を定義できます。前処理の例として、ファイル の事前オープン処理などが考えられます。一方、MFU-TC-TEARDOWN-PREFIX で始まる entry では、テスト 実行後の処理を定義できます。前処理でオープンしたファイルをクローズするような処理が該当します。前処理、後 処理ともに省略可能です。

自動生成されるテストプログラムはテンプレートであり、実際には、上記ルールに従い、テストを記述する必要があります。

7) 新規のテストケース(羽田・ロンドンヒースロー空間間の距離 (km) のテスト)を追加した上で、実行を行ないま す。

サンプルファイルを展開したフォルダー内の Basic¥TestProgram1.cbl の内容で、TestProgram1.cbl を上書 きしてください。 これは、テストケース "testDistance"を途中まで作成しています。前述した MFU-TC-SETUP-PREFIX, MFU-TC-PREFIX, MFU-TC-TEARDOWN-PREFIX の 3 entry が追加されています が、肝心な結果判定を記述していません。

結果判定処理を実装するため、82 行目に、以下のコードを追加して保存します。





3.1.2.2 MFUnit テストの実行

本テスト対象のプログラムは、環境変数で設定された空港情報が保存されたデータファイルを参照するため、手順 内で設定を行います。

「TestProgram1.cbl」を選択した状態で、マウスの右クリックでコンテクストメニューを表示し、[実行(R)] > [実行の構成(N)] をクリックします。

Q.	Coverage As	>	१ 🔲 プロパティ・	 Table Results 	🔊 🖓 Filter Definiti
0	実行(R)	>	💷 1 COBOL	アプリケーション	Alt+シフト+X,N
*	デバッグ(D)	>	2 COBOL	ユニット テスト	Alt+シフト+X,F
8 ∎	プロファイル(P) ☆ m	>	実行 の構	成(N)	

 2) 画面左側より「COBOL ユニットテスト」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、 [新規構成(W)]を選択します。



構成の作成、管理、および実行

🖹 🖻 🖚 🖹 🗶 🖻 🍸 🗕	このダイアログから起動設定を構成します:
フィルタ入力	『 - 選択した種類の構成を作成するには、「新規構成」ボタンを押します。
Apache Tomcat	- 選択した種類の起動構成プロトタイプを…、「新規プロトタイプ」ボタンを押
AspectJ/Java Application	👔 🕫 - 選択した構成をエクスポートするには、「エクスポート」ボタンを押します。
Market Construction Aspect Load-Time Weaving A	□□□ - 選択した構成をコピーするには、「複製」ボタンを押します。
■ COBOL JVM アプリケーション	メ - 選択した構成を削除するには、「削除」ボタンを押します。
🚾 COBOL JVM ユニット テスト	☆ - フィルタ・オプションを構成するには、「フィルタ」ボタンを押します。
M COBOL JVM リモート アプリケーシ	- プロトタイプをリンクするにけ、記動構 リンクリメニュー項目を選択してくだ
COBOL フニット テスト COBOL ユニット テスト	
⊕ Debug Adapter Laur □ 新規構 □ 新規構 □ 新規構	
	ローンコン(ア) りるには、起…ビット」メーユー項日を選択してくた

3) 「名前」に "AirtportDemoTutNativeMFUnitTest" を入力し、「環境」 タブを選択した後、[追加(A)] を クリックします。

1	新(N): AirportDemoTutNativeMFUnitTest		
	😞 一般 🔄 ソーズ 🚾 環境 📘 共通 🔎 実行時 🕫	▶ デバッグシンボル 🕹 動的分析 🔮 🕻	TF 🥒 コンテナー
	(注: ここで定義された変数は、任意の現在の設定値 環境スクリプト内の設定値を上書きします。)	、または任意の指定された	
	変数	値	追加(A)
			編集(E)
			削除(R)

4) 以下の情報を入力し、[OK] をクリックします。

環境変数を追加または変更します

変数: "dd_airports"

值: "..¥..¥AirportDemoTutNativeMFUnit¥airports.dat"

_airports
.¥AirportDemoTutNativeMFUnit¥airports.dat
OK キャンセル

5) さきほど追加した dd_airports 環境変数が表示されていることを確認して、[実行(R)] をクリックします。

(



名前(N): AirportDemoTutNativeMFUnitTest		
🕟 一般 🦻 ソース 🌄 環境 🔲 共通 🔎 実行時 🕏		
(注:ここで定義された変数は、任意の現在の設定値、 環境スクリプト内の設定値を上書きします。)	または任意の指定された	^
変数 dd_airports	値 ¥¥AirportDemoTutNativeMFUnit¥airport	追加(A) 編集(E) 削除(R)
実行する環境スクリプト: 場所: ファイルがプロジェクト内にある場合、絶対/	パスは相対パスになります。	参照
パラメータ:	お同保障した生態に言うの	
	前回休官した状態に戻す(実行(R)	v) 適用(Y) 閉じる

6) [Micro Focus Unit Testing] ビューをクリックすると、2つのテストケースが緑色で表示されています。緑色は、テ ストが正常に終了したことを表しています。

🖳 コンソール 🔝 問題 🧔 タスク 🔲 プロパティー 🏫 Table Results 🖉 Filter Definitions 🖫	Micro Focus Unit Testing 🗙 📴 コード カバレッジ 速 コード参
実行: 2/2 ロブー: 0 ロ失敗: 0	
✓ ➡ AirportDemoTutNativeMFUnitTest (0 ms)	テスト結果
Verify TestProgram1.cbl MFUT_TESTAIRCODE (0 ms) MFUT_TESTDISTANCE (0 ms)	

7) エラーケースを確認します。

「TestProgram1.cbl」をエラーとなるように修正します。本例では、テストプログラム内に記載されていた期待値 9591 を 9592 に修正しています。





ツールバーより、[実行] アイコンの矢印をクリックし、「AirportDemoTutNativeMFUnitTest」をクリックします。



MFUT_TESTDISTANCE のテストで、エラーが発生したことが一覧から判断できます。

🗟 コンソール 🖹 問題 🧔 タスク 🔲 プロパティー 🏗 Table Results 🖉 Filter Definitions 🖾	Micro Focus Unit Testing 🗙 💕 コード カパレッジ 💁 コード検索結果 嘿 直
実行: 2/2	
✓ arportDemoTutNativeMFUnitTest (0 ms)	テスト結果
 InstProgram1.cbl InstProgram1.cbl	FAILED: expected 9,592, but 9,591
	HND Tokyo Intl Japan Lat:+035.552258 Lon:+139.077969 LHR Heathrow

3.1.3 データ駆動型テスト

この方式では、複数のテストデータをデータファイルに設定しておくことで、データによって結果が異なるテストを効率よく行う ことができます。

3.1.3.1 MFUnit テストの作成

1) Eclipse IDE メニューより、[ファイル(F)] > [新規(N)] > [COBOL ユニットテストプロジェクト] を選択します。

ファイル(F) 編集(E) リファクタリング ナビゲート(N) 検索 プロジェクト(P) 実行(R) ウィンドウ(W) ヘルプ(H)

新規(N)	Alt+シフト+N >	2	COBOL プロジェクト
ファイルを開く(.)		2	COBOL コピーファイル プロジェクト
ファイル・システムからプロジェクトを開く		Ŷ	リモート COBOL プロジェクト
Recent Files	>	ġ	リモート COBOL コピーファイル プロジェクト
881/17 (0)		썉	COBOL/Java 相互運用機能のプロジェクト
閉じる(C)	Ctrl+W	E	COBOL ユニット テスト プロジェクト
すべて閉じる(L)	Ctrl+シフト+W	B	COBOL JVM プロジェクト
保存(S)	Ctrl+S	R	リモート COBOL ユニット テスト プロジェクト

プロジェクト名」に "AirportDemoTutNativeMFUnitTest2" を入力し、実行環境に合わせたプロジェクトテンプレートを選択した上で、[終了(F)] をクリックします。



プリジンクト キンブレート (34 ビット) アンブレートの設定を構成。 アンブレートの設定を構成。 アンブレートの設定を構成。 アンブレートの設定を構成。 アンブレートの設定を構成。 アンブレートの設定を構成。 アンブレートの設定を構成。 アンブレークション() ビダンブレートの設定を構成。 アンブレークション() ビダンブレートの設定を構成。 アンブレークション() ビダンブレートの設定を構成。 アンブレークション() ビダンブレートの参照 専用 アンブレークション() アンブレークション() Calva on the full Calva on		-			
プリジンクト テジブレート [64 ビクト] ● Micro Focus デブブレート [64 ビクト] ● Micro Focus デブブレート [64 ビクト] ● アナイレトの参照 ● デンブレートの参照 ● デンブレートの参照 ● デンオレト・ロケーションの使用(D) D* フランコントと注意(): D* ブナイレンステムを選択(): アイル・システムを選択(): アイル・システムを当び): アイル・システムを当び): アイル・システム・Bを選択(): アイル・システム・システム・システム・システム・システム・システム・システム・システム	プロ	ジェクト名(P) AirportDemoTutNative	MFUnitTest2		
● Micro Focus 7770F (64 E 97) ● Micro Focus 7770F (64 E 97) ● アブル-Fooget # #m: ● アブル-Fooget # #m: ● アブル-Fooget # #m: P アブル - 5074 # #m: P アブル - 5074 # #m: P アブル - 5074 # #m: P アブル - 5074 # #m: P アブル - 5074 # #m: P アブル - 5074 # #m: <t< th=""><th>プロ</th><th>コジェクト テンプレートを選択</th><th></th><th></th><th></th></t<>	プロ	コジェクト テンプレートを選択			
アンプレートの設定を構成。 アンプレートの登録を構成。 電子: アンプレートの設定を構成。 アンパレ・システムを選択(*) 宇宙 アイルシステムを選択(*) アフォル・システムを選択(*) アンパレ・システムを選択(*) アフォル・システムを選択(*) アンパレ・システムを選択(*) アフォルト・ロー・ションの使用(0) ア・システムを選択(*) アフォルト・ロー・フォルト・ロー・フォルト・ロー・フォルト・システムを選択(*) アオル・システムを選択(*) アフォルト・ロー・アフォルト・ロー・アオルト * (************************************		 ➢ Micro Focus テンプレート [32 ビット] ➢ Micro Focus テンプレート [64 ビット]]		
□ テンプレートの参照 場所: ● 原風. フィイルシステムを選択: default ② デフォルト・ロケーションの使用(D) ア・ク・コンジント(D): ロケーション(D): C4Workspace_tut_native_mfunit¥AirportDemoTutNativeMFUnitTe 参照(B). ア・ク・コンジンテムを選択(Y): デフォルト・ アンイル・システムを選択(Y): デフォルト ~ ② アンイル・システムを選択(Y): デフォルト ~ * 経7(f) キャンセル 注意) 先行作業にて指定したビット数と同じテンプレートを選択した上で、Eclipse IDE メニューより、[ファイル・ * 新規(N)] > [COBOL ユニットテスト] アンイル・ジステムからプロジェクトを選択します。 アンイル・ジステムからプロジェクトを選択した上で、Eclipse IDE メニューより、[ファイル・ * 「新規(N)] > [COBOL ユニットテスト] アンイル・ジステムからプロジェクトを選択します。 アンイル・ジステムからプロジェクトを選択します。 アンイル・ジステムからプロジェクトを選択してください。 * 新規(N)] > [COBOL ユニットテスト] * アンイル・グンプランクト * 「新見(N) * アンイル・グ・ファイル・プロジェクト * 「新してのしてい」かび アンドン・ジェーキャー * アンドレートン・クレー・シンジェート	L				<u>テンプレートの設定を構成</u>
ダボ]テンプレートの参照			
ファイルシステムを選択() マテスルシークションの使用(D) ローションパい: CWorkspace_tut_native_mfunit¥AirportDemoTutNativeMFUnitTe 学習(f) ファイル・ウステムを選択(Y): デフォルト・ アイル・ウステムを選択(Y): デフォルト・ * (************************************	鳺	易所:			参照
 ○ デフォルト・ロケーションの使用(0) □ ケーション(1): 〔Cłworkspace_tut_native_mfunit¥AirportDemoTutNativeMFUnitTe 参照(R) ファイル・システムを選択(Y): デフォルト ②		ファイルシステムを選択: default 、	~		
ビビファスキルド・ロンーションの映用(U) ロケーション(L): C:¥workspace_tut_native_mfunit¥AirportDemoTutNativeMFUnitTe 参照(R) ファイル・システムを選択(Y): アフォルト 注意) 先行作業にて指定したビット数と同じテンプレートを選択してください。 AirportDemoTutNativeMFUnitTest2 プロジェクトやを選択した上で、Eclipse IDE メニューより、[ファイ, シント・シートを通知した] > [新規(N)] > [COBOL ユニットテスト] を選択します。 アイル(P) 編集(E) リファクタリング ナビゲート(N) 検索 プロジェクト(P) 実行(R) ウィンドウ(W) ヘルブ(H) 新規(N) Alt+シフト・N> ジロシェクト・システムからプロジェクトを選択してください。 アイル(P) 編集(E) リファクタリング ナビゲート(N) 検索 プロジェクト(P) 実行(R) ウィンドウ(W) ヘルブ(H) 新規(N) Alt+シフト・N> COBOL プロジェクト マフィル・システムからプロジェクトを開く (COBOL ゴビーファイル プロジェクト アイル・システムからプロジェクトを開く (COBOL コピーファイル プロジェクト (COBOL コピーファイル プロジェクト (COBOL コピーファイル プロジェクト マスイ局でしてい (CH+W) (COBOL コピーファイル プロジェクト マスで同じる(L) Ctrl+ジフト+W (COBOL コピーファイル プロジェクト (COBOL パM ゴロジェクト (COBOL パM ゴロジェクト マスで同じる(L) Ctrl+ジフト (COBOL パM ゴロジェクト マスで同じる(L) Ctrl+ジフト (COBOL パM ゴロジェクト マスで同じな(L) Ctrl+ジフト (CH+V) (COBOL パM ゴロジェクト マスで属ないた (大都に戻す(T)) (CH+V)					
アイル・システムを選択(Y): アフォルト ② 後了(F) 生ヤソセル 注意) 先行作業にて指定したビット数と同じテンプレートを選択してください。 NirportDemoTutNativeMFUnitTest2 プロジェクトを選択した上で、Eclipse IDE メニューより、[ファイ. > [新規(N)] > [COBOL ユニットテスト]を選択します。 ファイル・システムを選択した」で、Eclipse IDE メニューより、[ファイ. > [新規(N)] > [COBOL ユニットテスト]を選択します。 ファイル・システムからプロジェクトを選択します。 ファイル・システムからプロジェクトを選択します。 アイル・システムからプロジェクトを選択します。 アイル・システムからプロジェクトを聞く ● ファイル・プロジェクトを聞く ● マーイルを聞く() ● スァイルを聞く() ● スァイルを聞く() ● オイス府じる(L) ● オイス府じる(L) ● マーイル・プロジェクト ■ 保存(S) ● オイス保存(S) ● オイス保存(C) ● オイス保存(C) ● オイス保存(C) ● オイス保存(C) ● オイス保存(C) ● マ新(F) ● 安新(F) ● マー ● 安新(F) ● 安新(F) ● 安新(F) ● マー ● 安新(F) ● 安新(F) ● 安新(F) ● 安新(F) <t< td=""><td>☑</td><td>テノオルト・ロクーンヨンの使用(D)</td><td>mfunit¥AirportDemoTut</td><td>lative</td><td>MELInitTe 参昭(R)</td></t<>	☑	テノオルト・ロクーンヨンの使用(D)	mfunit¥AirportDemoTut	lative	MELInitTe 参昭(R)
アイル・ワイル と (1) アイル・ワイル と (1) ②	Ш.).		_manit#AnportDemoida	vativei	
		ファイル・ンステムを迭め(1);	J JANK ~		
注意) 先行作業にて指定したビット数と同じテンプレートを選択してください。 AirportDemoTutNativeMFUnitTest2 プロジェクトを選択した上で、Eclipse IDE メニューより、[ファイ, > [新規(N)] > [COBOL ユニットテスト] を選択します。 ファイル(F) 編集(E) リファクタリング ナビゲート(N) 検索 プロジェクト(P) 実行(R) ウィンドウ(W) ヘルブ(H) 新規(N) Alt+シフト+N> アイル・ジステムからプロジェクトを開く COBOL プロジェクト アイル・システムからプロジェクトを開く COBOL プロジェクト アイル・システムからプロジェクトを開く COBOL コピーファイル プロジェクト アイル・システムからプロジェクトを開く COBOL コピーファイル プロジェクト 開じる(C) Ctrl+W すべて閉じる(L) Ctrl+シフト+W 保存(S) Ctrl+シフト+W 保存(S) Ctrl+シフト+W 電 応告だ付けて保存(A) マレート COBOL コニット テスト プロジェクト 素前を付けて保存(A) ビロ+シフト+S 東新(F) FS 東新(F) FS 東新(F) FS 東新(F) FS 東新(F) FS 「石区切り文字の変換(D) アイル					
注意) 先行作業にて指定したビット数と同じテンプレートを選択してください。 AirportDemoTutNativeMFUnitTest2 プロジェクトを選択した上で、Eclipse IDE メニューより、[ファイ, > [新規(N)] > [COBOL ユニットテスト] を選択します。 ファイル(F) 編集(E) リファクタリング ナビゲート(N) 検索 プロジェクト(P) 実行(R) ウィンドウ(W) ヘルプ(H) 新規(N) Alt+シフト+N> COBOL プロジェクト アイル・システムからプロジェクトを開く Recent Files アイル・システムからプロジェクトを開く Recent Files 同じる(C) Ctrl+W すべて閉じる(L) Ctrl+シフト+W COBOL ユニット テスト プロジェクト 保存(S) Ctrl+S Ainを付けて保存(A) テオペに保存(E) Ctrl+シフト+S 前回保存した状態に戻す(T) を動い(N 名前を変更(M) を動い(N を動い(N たまりをした状態に戻す(D) とののし、MATON Alt+シフト+N たいののし、	(?))		終了	て(F) キャンセル
先行作業にて指定したビット数と同じテンプレートを選択してください。 AirportDemoTutNativeMFUnitTest2 プロジェクトを選択した上で、Eclipse IDE メニューより、[ファイ, シ > [新規(N)] > [COBOL ユニットテスト] を選択します。 ファイル(F) 編集(E) リファクタリング ナビゲート(N) 検索 プロジェクト(P) 実行(R) ウィンドウ(W) ヘルプ(H) 新規(N) Alt+シフト+N> アイルを聞く(.) COBOL プロジェクト ファイル(F) 編集(E) リファクタリング ナビゲート(N) 検索 プロジェクト(P) 実行(R) ウィンドウ(W) ヘルプ(H) 新規(N) Alt+シフト+N> アイルを聞く(.) COBOL プロジェクト ファイル、システムからプロジェクトを開く COBOL プロジェクト Recent Files リモート COBOL コピーファイル プロジェクト 開じる(C) Ctrl+W すべて閉じる(L) Ctrl+シフト+W 保存(S) Ctrl+シフト+W 保存(S) Ctrl+シフト+S 前回保存した状態に戻す(T) COBOL JVM プロジェクト 移動(V) Fahを変更(M) Fahを変更(M) そ約を変更(M) Fabを変更新(F) Fs アイル GoBOL JUM プロジェクト Directory Server 接続 フォルダー COBOL JUM プログラム Directory Server 接続	Ŷ)	L	終了	了(F) キャンセル
AirportDemoTutNativeMFUnitTest2 プロジェクトを選択した上で、Eclipse IDE メニューより、[ファイ, > [新規(N)] > [COBOL ユニットテスト] を選択します。 ファイル(F) 編集(E) リファクタリング ナビゲート(N) 検索 プロジェクト(P) 実行(R) ウィンドウ(W) ヘルブ(H) 新規(N) Alt+シフト+N COBOL プロジェクト アフィルを開く(.) ファイルを開く(.) ファイル・システムからプロジェクトを開く Recent Files PIE- COBOL ユニットテスト プロジェクト 閉じる(C) Ctrl+W すべて閉じる(L) Ctrl+シフト+W COBOL ユニーナクイル プロジェクト 保存(S) Ctrl+S 名前を付けて保存(A) 保存(S) Ctrl+S ジャイ(保存(E) Ctrl+シフト+S 前回保存した状態に戻す(T) を動(V) 名前を変更(M) 名前を変更(M) アコンテの変換(D) 、 COBOL プログラム	⑦ 注意) 意)	L	終了	了(F) キャンセル
AirportDemoTutNativeMFUnitTest2 プロジェクトを選択した上で、Eclipse IDE メニューより、[ファイ, > [新規(N)] > [COBOL ユニットテスト] を選択します。 ファイル(F) 編集(E) リファクタリング ナビゲート(N) 検索 プロジェクト(P) 実行(R) ウインドウ(W) ヘルプ(H) 新規(N) Alt+シフト+N> ② COBOL プロジェクト ファイルを開く(.) ファイルを開く(.) ファイルを開く(.) マアイルを開く(.) Recent Files >> 同じる(C) Ctrl+W すべて同じる(L) Ctrl+シフト+W 保存(S) Ctrl+シフト+W 名前をなけて保存(A) マイ保存(E) Ctrl+シフト+S 前回保存した状態に戻す(T) ** 移動(V) 名前を変更(M) 名前を変更(M) 予備区切り文字の変換(D) >> Aite シフト・Aite Signature (Composition of the sector o	⑦ 注意 先ぞ) 意) テ作業にて指定したビット数と	同じテンプレートを選	終了	了(F) キャンセル てください
> [新規(N)] > [COBOL ユニットテスト] を選択します。 ファイル(F) 編集(E) リファクタリング ナビゲート(N) 検索 ブロジェクト(P) 実行(R) ウィンドウ(W) ヘルプ(H) 新規(N) Alt+シフト+N> アイルを開く(.) COBOL プロジェクト ファイル・システムからプロジェクトを開く COBOL プロジェクト マアイル・システムからプロジェクトを開く ビート COBOL プロジェクト マアイル・システムからプロジェクトを開く ビート COBOL プロジェクト Recent Files アイ・COBOL コピーファイル プロジェクト 間じる(C) Ctrl+W すべて閉じる(L) Ctrl+シフト+W 保存(S) Ctrl+シフト+W 保存(S) Ctrl+シフト+W 第 名前を付けて保存(A) ビート COBOL コニット テスト プロジェクト 「回 保存した状態に戻す(T) ビート COBOL コニット テスト プロジェクト 移動(V) F2 夏新(F) F5 更新(F) F5 東新(F) F5 夏新(F) F5 夏新(F) F5 夏新(F) F5 「日区切り文字の変換(D) アイル) 注意 先行	う 意) テ作業にて指定したビット数と	同じテンプレートを選	終了	7(F) キャンセル てください。
ファイル(F) 編集(E) リファクタリング ナビゲート(N) 検索 プロジェクト(P) 実行(R) ウィンドウ(W) ヘルブ(H) 新規(N) Alt+シフト+N> ど COBOL プロジェクト ファイルを開く(.) ジ COBOL プビジェクト ファイル・システムからプロジェクトを開く ビ COBOL フピーファイル プロジェクト Recent Files ジ リモート COBOL コピーファイル プロジェクト 間じる(C) Ctrl+W COBOL ユニット テスト プロジェクト すべて閉じる(L) Ctrl+シフト+W ご 保存(S) Ctrl+S W) リモート COBOL ユニット テスト プロジェクト 4 高前を付けて保存(A) ビ ジモイ保存(E) すべて保存(E) Ctrl+シフト+S ジ 前回保存した状態に戻す(T) ビ COBOL JWM プロジェクト 移動(V) F2 ジェクト(R) ビ 私前を変更(M) F2 F5 更新(F) F5 F5 アイルダー 「口区切り文字の変換(D) デ ア ア	了 注意 先行 Airp) 意) う作業にて指定したビット数と portDemoTutNativeMFU	し 同じテンプレートを選 nitTest2 プロジェク	終了 択して トを選	ア(F) キャンセル てください。 選択した上で、Eclipse IDE メニューより、[ファイル
新規(N) Alt+シフト+N> ど COBOL プロジェクト ファイルを開く() マアイルを開く() ビ COBOL プロジェクト アイル・システムからプロジェクトを開く ロ リモート COBOL プロジェクト Recent Files アイル・システムからプロジェクトを開く ロ アイル・システムからプロジェクトを開く ロ リモート COBOL プロジェクト 閉じる(C) Ctrl+W COBOL ユニット テスト プロジェクト マベて閉じる(L) Ctrl+シフト+W COBOL コニット テスト プロジェクト 名前を付けて保存(A) ビ Uモート COBOL コニット テスト プロジェクト ネ前を付けて保存(A) ビ Uモート COBOL コニット テスト プロジェクト ボロ保存した状態に戻す(T) ビ COBOL JVM プロジェクト 移動(V) ビ Uモート COBOL コニット テスト プロジェクト 移動(V) ビ フロジェクト(R) 移動(V) ビ COBOL JVM プロジェクト シベて保存(E) Ctrl+シフト+S ジ 前回保存した状態に戻す(T) ビ Enterprise Server Common Web Administration 接近 シボルダー アイル アイル 「 ファイル アイル	了 注意 先行 Airp > [新) <mark>ま) テ作業にて指定したビット数と</mark> portDemoTutNativeMFUr 新規(N)] > [COBOL ユニン	同じテンプレートを選 nitTest2 プロジェク ットテスト] を選択し	終了 択して トを選 ます。	ア(F) キャンセル てください。 選択した上で、Eclipse IDE メニューより、[ファイノ
ファイルを開く(.) ご COBOL コピーファイル プロジェクト アイル・システムからプロジェクトを開く (*) リモート COBOL ブビウト Recent Files (*) リモート COBOL コピーファイル プロジェクト 閉じる(C) Ctrl+W (*) COBOL コピーファイル プロジェクト オベて閉じる(L) Ctrl+シフト+W (*) COBOL コピーファイル プロジェクト (*) (*) COBOL コピーファイル プロジェクト (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*)	② 注意 先行 Airp > [系 ファ・) テ作業にて指定したビット数と portDemoTutNativeMFUr 新規(N)] > [COBOL ユニ イル(F) 編集(E) リファクタリング	同じテンプレートを選 nitTest2 プロジェク ットテスト] を選択し ナビゲート(N) 検索 プ	終了 択して トを遅 ます。 コジェク	7(F) キャンセル てください。 選択した上で、Eclipse IDE メニューより、[ファイノ , ウト(P) 実行(R) ウィンドウ(W) ヘルプ(H)
□ ファイル・システムからプロジェクトを開く 「リモート COBOL フロジェクト Recent Files > 「リモート COBOL コピーファイル プロジェクト 閉じる(C) Ctrl+W COBOL ノニット テスト プロジェクト すべて閉じる(L) Ctrl+シフト+W ご 保存(S) Ctrl+S リモート COBOL コニット テスト プロジェクト このBOL ユニット テスト プロジェクト ご COBOL ノンハ プロジェクト このBOL コニット テスト プロジェクト ご COBOL コニット テスト プロジェクト このBOL コニット テスト プロジェクト ご ジェクト COBOL コニット テスト プロジェクト 「ロ保存した状態に戻す(T) ご フロジェクト(R) 移動(V) F2 「ロジェクト(R) 「回日保存した状態に戻す(T) 「 フロジェクト(R) 「回日保存した状態に戻す(T) 「 「ロジェクト(R) 「回日保存した状態に戻す(T) 「 「 「回日保存した状態に戻す(T) 「 「 「回日保存した状態に戻す(T) 「 「 「回日保存した状態に戻す(T) 「 「 「日日中国会会会会(M) 「 「 「日日中国会会会会(M) 「 「 「日日中国会会会会(D) 「 「 「日日中国会会会会(D) 「 「 「日日中国会会会会(D) 「 「 「日日中国会会会会会会(D) 」 「	で 注意 先行 Airp > [系 ファ・) う作業にて指定したビット数と portDemoTutNativeMFU 新規(N)] > [COBOL ユニ イル(F) 編集(E) リファクタリング 新規(N)	同じテンプレートを選 nitTest2 プロジェク ットテスト] を選択し ナビゲート(N) 検索 プ Alt+シフト+N	終了 択して トを選 ます。 コジェク	7(F) キャンセル てください。 選択した上で、Eclipse IDE メニューより、[ファイル , ッ アト(P) 実行(R) ウィンドウ(W) ヘルプ(H) © COBOL プロジェクト
Recent Files 今日 Pt Problem L T Print Ph D T Print 開じる(C) Ctrl+W COBOL/Java 相互運用機能のプロジェクト すべて閉じる(L) Ctrl+YW COBOL ユニット テスト プロジェクト (C) Ctrl+シフト+W COBOL ユニット テスト プロジェクト (C) Ctrl+シフト+W COBOL ユニット テスト プロジェクト (C) Ctrl+シフト+W COBOL コニット テスト プロジェクト (C) Ctrl+シフト+W COBOL コニット テスト プロジェクト (C) Ctrl+シフト+S リモート COBOL ユニット テスト プロジェクト すべて保存(E) Ctrl+シフト+S リモート COBOL コニット テスト プロジェクト (C) Ctrl+シフト+S リモート COBOL コニット テスト プロジェクト 前回保存した状態に戻す(T) 「 プロジェクト(R) 移動(V) F2 Enterprise Server Common Web Administration 接) ショ新(F) F5 Directory Server 接続 フォルダー 「 ファイル 「C) アーイル (C) COBOL プログラム	⑦ 注意 先行 Airp > [新 ファ・) テ作業にて指定したビット数と portDemoTutNativeMFUr 新規(N)] > [COBOL ユニッ イル(F) 編集(E) リファクタリング 新規(N) ファイルを開く(.)	同じテンプレートを選 nitTest2 プロジェク ットテスト] を選択し ナビゲート(N) 検索 プ Alt+シフト+N	終了 択して トを選 ます。 コジェク	7(F) キャンセル てください。 選択した上で、Eclipse IDE メニューより、[ファイJ ・ クト(P) 実行(R) ウィンドウ(W) ヘルプ(H) © COBOL プロジェクト © COBOL コピーファイル プロジェクト
閉じる(C) Ctrl+W COBOL ユニット テスト プロジェクト すべて閉じる(L) Ctrl+シフト+W COBOL ユニット テスト プロジェクト 保存(S) Ctrl+S リモート COBOL ユニット テスト プロジェクト 名前を付けて保存(A) COBOL Jビット テスト プロジェクト COBOL Jビット テスト プロジェクト すべて保存(E) Ctrl+シフト+S リモート COBOL JVM プロジェクト 前回保存した状態に戻す(T) プロジェクト(R) Tロジェクト(R) 移動(V) F2 Enterprise Server Common Web Administration 接近 夏新(F) F5 アオルダー 行区切り文字の変換(D) アイル COBOL プログラム	② 注意 先行 Airp > [系 ファ・) テ作業にて指定したビット数と portDemoTutNativeMFUr 新規(N)] > [COBOL ユニ イル(F) 編集(E) リファクタリング 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開	<mark>同じテンプレートを選</mark> nitTest2 プロジェク ットテスト] を選択し ナビゲート(N) 検索 プ Alt+シフト+N	終了 択して トを選 ます。 コジェク 警 警	7(F) キャンセル てください。 選択した上で、Eclipse IDE メニューより、[ファイJ ・ クト(P) 実行(R) ウインドウ(W) ヘルプ(H) © COBOL プロジェクト © COBOL コピーファイル プロジェクト © リモート COBOL プピュファト
すべて閉じる(L) Ctrl+シフト+W ご COBOL JVM プロジェクト 日 保存(S) Ctrl+S 第 Uモート COBOL ユニット テスト プロジェクト 日 名前を付けて保存(A) ご COBOL JVM ユニット テスト プロジェクト マベ保存(E) Ctrl+シフト+S ご フロジェクト(R) 市回保存した状態に戻す(T) ご プロジェクト(R) ご 移動(V) ドシフト+S ご Directory Server Common Web Administration 接近 2 A前を変更(M) F2 ご フォルダー 行区切り文字の変換(D) アイル COBOL プログラム	② 注意 先行 Airp > [系 ファ・) テ作業にて指定したビット数と portDemoTutNativeMFUr 新規(N)] > [COBOL ユニ イル(F) 編集(E) リファクタリング 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開 Recent Files	<mark>同じテンプレートを選</mark> nitTest2 プロジェク ットテスト] を選択し ナビゲート(N) 検索 プ Alt+シフト+N	終了	ア(F) キャンセル てください。 選択した上で、Eclipse IDE メニューより、[ファイ) ウト(P) 実行(R) ウィンドウ(W) ヘルプ(H) COBOL プロジェクト COBOL コピーファイル プロジェクト リモート COBOL コピーファイル プロジェクト リモート COBOL コピーファイル プロジェクト COBOL ノビーファイル プロジェクト COBOL ノビーファイル プロジェクト
保存(S) Ctrl+S 第 リモート COBOL ユニット テスト プロジェクト 名前を付けて保存(A) 谷前を付けて保存(A) 〇 COBOL JVM ユニット テスト プロジェクト すべて保存(E) Ctrl+シフト+S リモート COBOL JVM プロジェクト 前回保存した状態に戻す(T) ゴ プロジェクト(R) 移動(V) ビ F1 移動(V) ビ Enterprise Server Common Web Administration 接近 夏新(F) F5 Directory Server 接続 可知以夕一 フォルダー 〇 ファイル 〇 COBOL プログラム	② 注意 先行 Airp > [系 ファ・) 意) う作業にて指定したビット数と portDemoTutNativeMFU 新規(N)] > [COBOL ユニ イル(F) 編集(E) リファクタリング 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開 Recent Files 閉じる(C)	同じテンプレートを選 nitTest2 プロジェク ットテスト]を選択し ナビゲート(N) 検索 プ Alt+シフト+N 軟、	終了 択して 沢して ジェク シ の の の の の の の の の の の の の	ア(F) キャンセル てください。 選択した上で、Eclipse IDE メニューより、[ファイノ クト(P) 実行(R) ウィンドウ(W) ヘルプ(H) * COBOL プロジェクト * COBOL コピーファイル プロジェクト * リモート COBOL コピーファイル プロジェクト * リモート COBOL コピーファイル プロジェクト * COBOL ノピーファイル プロジェクト * COBOL ノピーファイル プロジェクト * COBOL ノピーファイル プロジェクト * COBOL ノニット テスト プロジェクト
■ 名前を付けて保存(A) ピ COBOL JVM ユニット テスト プロジェクト すべて保存(E) Ctrl+シフト+S リモート COBOL JVM プロジェクト 前回保存した状態に戻す(T) ゴ プロジェクト(R) 移動(V) ビ Enterprise Server Common Web Administration 接続 夏新(F) F5 ゴ 有区切り文字の変換(D) ア ア	② 注意 先行 Airp > [新 ファ・) う 作業にて指定したビット数と portDemoTutNativeMFUr 新規(N)] > [COBOL ユニ) イル(F) 編集(E) リファクタリング 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開 Recent Files 閉じる(C) すべて閉じる(L)	同じテンプレートを選 nitTest2 プロジェク ットテスト] を選択し ナビゲート(N) 検索 プ Alt+シフト+N 駅く Ctrl+W Ctrl+シフト+W	終了 振して ます。 シ 警 警 警 警 警 警 警 警 警 警 警 警 警	ア(F) キャンセル てください。 選択した上で、Eclipse IDE メニューより、[ファイノ クト(P) 実行(R) ウィンドウ(W) ヘルプ(H) ウト(P) 実行(R) ウィンドウ(W) ヘルプ(H) * COBOL プロジェクト * COBOL コピーファイル プロジェクト * リモート COBOL ゴピーファイル プロジェクト * COBOL/Java 相互運用機能のプロジェクト * COBOL ユニット テスト プロジェクト * COBOL JVM プロジェクト
■ すべて保存(E) Ctrl+シフト+S プ リモート COBOL JVM プロジェクト プロジェクト(R) 移動(V) 移動(V) プロジェクト(R) 移動(V) を動を変更(M) ア2 ア新(F) F5 ア5 行区切り文字の変換(D) ア ア	② 注意 先行 Airpp > [系 ファ・) う作業にて指定したビット数と portDemoTutNativeMFUr 新規(N)] > [COBOL ユニッ イル(F) 編集(E) リファクタリング 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開 Recent Files 閉じる(C) すべて閉じる(L) 保存(S)	同じテンプレートを選 nitTest2 プロジェク ットテスト] を選択し ナビゲート(N) 検索 プ Alt+シフト+N 私t+シフト+N な	終了 採して 選 ます。	ア(F) キャンセル てください。 選択した上で、Eclipse IDE メニューより、[ファイノ アト(P) 実行(R) ウィンドウ(W) ヘルブ(H) * アト(P) 実行(R) ウィンドウ(W) ヘルブ(H) *
前回保存した状態に戻す(T) 移動(V) 移動(V) 名前を変更(M) 更新(F) 行区切り文字の変換(D) 「 日 ての変換(D) 」 「 」 「 」 「 」 「 」 ののの変換(D) 」 「 」 「 」 「 」 「 」 」 のののの変換(D) 」 「 」 「 」 」 「 」 」 のののの変換(D) 」 「 」 「 」 」 」 「 」 」 」 」 」 」 」 「 」 」 」 」 」 」 』 『 」 「 」 」 」 」 「 」 」 」 」 」 」 」 「 」 」 」 」 」 」 」 」 』 」 「 」 」 」 」 」 」	「注意 先行 Airp > [系 ファ・) う作業にて指定したビット数と portDemoTutNativeMFUr 新規(N)] > [COBOL ユニッ 1ル(F) 編集(E) リファクタリング <mark>新規(N)</mark> ファイルを開く(.) ファイル・システムからプロジェクトを開 Recent Files 閉じる(C) すべて閉じる(L) 保存(S) 名前を付けて保存(A)	同じテンプレートを選 nitTest2 プロジェク ットテスト] を選択し ナビゲート(N) 検索 プ Alt+シフト+N 駅 Ctrl+W Ctrl+シフト+W	終了	ア(F) キャンセル てください。 選択した上で、Eclipse IDE メニューより、[ファイノ タト(P) 実行(R) ウィンドウ(W) ヘルプ(H) ア(P) 実行(R) ウィンドウ(W) ヘルプ(H) COBOL プロジェクト COBOL コピーファイル プロジェクト リモート COBOL コピーファイル プロジェクト U リモート COBOL コピーファイル プロジェクト COBOL/Java 相互運用機能のプロジェクト COBOL JAVA プロジェクト COBOL コニット テスト プロジェクト U リモート COBOL ユニット テスト プロジェクト COBOL JVM プロジェクト マOBOL JVM プロジェクト U リモート COBOL ユニット テスト プロジェクト COBOL JVM ユニット テスト プロジェクト マOBOL JVM ユニット テスト プロジェクト
移動(V) で Enterprise Server Common Web Administration 接近 営 名前を変更(M) F2 夏新(F) F5 行区切り文字の変換(D) 下	② 注意 先行 Airp > [系 ファ・ ■ ■ ■ ■ ■) う 行作業にて指定したビット数と portDemoTutNativeMFUr 新規(N)] > [COBOL ユニ: イル(F) 編集(E) リファクタリング <u>新規(N)</u> ファイルを開く(.) ファイル・システムからプロジェクトを開 Recent Files 閉じる(C) すべて閉じる(L) 保存(S) 名前を付けて保存(A) すべて保存(E)	同じテンプレートを選 nitTest2 プロジェク ットテスト] を選択し ナビゲート(N) 検索 プ Alt+シフト+N 覧く Ctrl+シフト+W Ctrl+シフト+S	終了	ア(F) キャンセル てください。 選択した上で、Eclipse IDE メニューより、[ファイ) タト(P) 実行(R) ウィンドウ(W) ヘルプ(H) COBOL プロジェクト ウモンドウ(W) ヘルプ(H) COBOL プロジェクト リモート COBOL ブロジェクト リモート COBOL ブレーファイル プロジェクト COBOL Jとーファイル プロジェクト COBOL JンW プロジェクト COBOL JンW プロジェクト UTート COBOL コニット テスト プロジェクト UTート COBOL JVM プロジェクト マロジ・トレーア
✓ 名前を変更(M) F2 ▲ Directory Server 接続 ② 更新(F) F5 〇 フォルダー 行区切り文字の変換(D) > 〇 ファイル 〇 COBOL プログラム	② 注意 先行 Airp > [系 ファ・) う 作業にて指定したビット数と portDemoTutNativeMFU 新規(N)] > [COBOL ユニ: 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開 Recent Files 閉じる(C) すべて閉じる(L) 保存(S) 名前を付けて保存(A) すべて保存(E) 前回保存した状態に戻す(T)	同じテンプレートを選 nitTest2 プロジェク ットテスト]を選択し ナビゲート(N)検索 プ Alt+シフト+N 駅く Ctrl+シフト+W Ctrl+シフト+S	終了	ア(F) キャンセル てください。 選択した上で、Eclipse IDE メニューより、[ファイ) タト(P) 実行(R) ウィンドウ(W) ヘルプ(H) * COBOL プロジェクト * COBOL プロジェクト * リモート COBOL プロジェクト * リモート COBOL プロジェクト * COBOL ノロジェクト * COBOL ノロジェクト * COBOL ノロジェクト * ワート COBOL コピーファイル プロジェクト * COBOL ノンM オロジェクト * COBOL コニット テスト プロジェクト * リモート COBOL コニット テスト プロジェクト * リモート COBOL コニット テスト プロジェクト * リモート COBOL コント テスト プロジェクト * リモート COBOL JVM プロジェクト * リモート COBOL JVM プロジェクト * リモート COBOL JVM プロジェクト * ブロジェクト(R)
 ● 更新(F) 行区切り文字の変換(D) 行区切り文字の変換(D) COBOL プログラム 	② 注意 先行 Airpp > [系 □ □) 意) う作業にて指定したビット数と portDemoTutNativeMFU 新規(N)] > [COBOL ユニ) 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開 Recent Files 閉じる(C) すべて閉じる(L) 保存(S) 名前を付けて保存(A) すべて保存(E) 前回保存した状態に戻す(T) 移動(V)	同じテンプレートを選 nitTest2 プロジェク ットテスト]を選択し ナビゲート(N) 検索 プ Alt+シフト+N 駅く Ctrl+シフト+W Ctrl+シフト+S	終了	ア(F) キャンセル てください。 選択した上で、Eclipse IDE メニューより、[ファイ) タト(P) 実行(R) ウィンドウ(W) ヘルブ(H) * COBOL プロジェクト * COBOL プロジェクト * Uモート COBOL ブロジェクト * Uモート COBOL ブロジェクト * COBOL ノニット テスト プロジェクト * COBOL ノニット テスト プロジェクト * COBOL コニット テスト プロジェクト * COBOL コニット テスト プロジェクト * Uモート COBOL コニット テスト プロジェクト * UTート COBOL コニット テスト プロジェクト * Tロジェクト(R)
行区切り文字の変換(D) > □ ノパ1ル ぼ COBOL プログラム	② 注意 先行 Airp > [系 □ □) う作業にて指定したビット数と portDemoTutNativeMFU 新規(N)] > [COBOL ユニ) が規(N) ファイルを開く(.) ファイルを開く(.) ファイル・システムからプロジェクトを開 Recent Files 閉じる(C) すべて閉じる(L) 保存(S) 名前を付けて保存(A) すべて保存(E) 前回保存した状態に戻す(T) 移動(V) 名前を変更(M)	同じテンプレートを選 nitTest2 プロジェク ットテスト] を選択し ナビゲート(N) 検索 プ Alt+シフト+N 私t+シフト+N Ctrl+シフト+W Ctrl+シフト+S Ctrl+シフト+S	終了	ア(F) キャンセル てください。 選択した上で、Eclipse IDE メニューより、[ファイノ クト(P) 実行(R) ウィンドウ(W) ヘルブ(H) クト(P) 実行(R) クィンドウ(W) ヘルブ(H) クト(P) 実行(R) ウィンドウ(W) ヘルブ(H) クト(P) 実行(R) ウィンドウ(W) ヘルブ(H) クロジェクト COBOL Jビーファイル プロジェクト マレート COBOL Jビーファイル プロジェクト クロシレート COBOL Jニット テスト プロジェクト COBOL JUM プロジェクト マレート マレート COBOL JUM プロジェクト フロジェクト フロジェクト(R) アロジェクト(R) Enterprise Server Common Web Administration 接続 Directory Server 接続 マレーダ マレーダ マレーダ マレーダ
	② 注意 先行 Airp > [系 □ □ □) う作業にて指定したビット数と DoortDemoTutNativeMFUr 新規(N)] > [COBOL ユニ: イル(F) 編集(E) リファクタリング 新規(N) ファイルを開く(.) ファイルを開く(.) ファイル・システムからプロジェクトを開 Recent Files 閉じる(C) すべて閉じる(L) 保存(S) 名前を付けて保存(A) すべて保存(E) 前回保存した状態に戻す(T) 移動(V) 名前を変更(M) 更新(F)	同じテンプレートを選 nitTest2 プロジェク ットテスト] を選択し ナビゲート(N) 検索 プ Alt+シフト+N 私t+シフト+N Ctrl+シフト+W Ctrl+シフト+S Ctrl+シフト+S	終了	ア(F) キャンセル てください。 選択した上で、Eclipse IDE メニューより、[ファイノ クト(P) 実行(R) ウィンドウ(W) ヘルブ(H) アト(P) 実行(R) ウィンドウ(W) ヘルブ(H) * COBOL プロジェクト * OBOL プロジェクト * Uモート COBOL ブロジェクト * Uサート COBOL コピーファイル プロジェクト * COBOL ノニット テスト プロジェクト * COBOL ユニット テスト プロジェクト * Uサート COBOL ユニット テスト プロジェクト * Uサート COBOL コニット テスト プロジェクト * Uサート COBOL ノVM プロジェクト * Uサート COBOL JVM プロジェクト * フォルダー * Directory Server 接続 * フォルダー * マスペリ
○ 印刷(P) Ctrl+P 除 COBOL 1ピーファイル	② 注意 先行 Airp > [系 ファ・ ■ ■ ■ ■ ■ ■ ■ ■	 	同じテンプレートを選 nitTest2 プロジェク ットテスト] を選択し ナビゲート(N) 検索 プ Alt+シフト+N 私t+シフト+N てtrl+シフト+W Ctrl+シフト+S Ctrl+シフト+S	終了 択して たまごびび の びび の ぎ 回 ぎ 回 ※	ア(F) キャンセル てください。 選択した上で、Eclipse IDE メニューより、[ファイ) タト(P) 実行(R) ウィンドウ(W) ヘルプ(H) ア(P) 実行(R) ウィンドウ(W) ヘルプ(H) ア(P) 実行(R) ウィンドウ(W) ヘルプ(H) ア(P) 実行(R) ウィンドウ(W) ヘルプ(H) ア(P) 実行(R) ウィンドウ(W) ヘルプ(H) COBOL プロジェクト COBOL コピーファイル プロジェクト マOBOL コピーファイル プロジェクト マOBOL コピーファイル プロジェクト COBOL コニット テスト プロジェクト COBOL コニット テスト プロジェクト マOBOL コニット テスト プロジェクト マロジェクト マロジェクト マロジェクト マスト プロジェクト アロジェクト アロジェクト マロジェクト(R) Enterprise Server Common Web Administration 接続 Directory Server 接続 フォルダー ファイル COBOL プログラ()
≥ インポート(I) ≥ スタンドアロン ファイル	② 注意 先行 Airp > [系 ファ・ ■ ■ ■ ■ ■ ■ ■	 う 「	同じテンプレートを選 nitTest2 プロジェク ットテスト] を選択し ナビゲート(N) 検索 プ Alt+シフト+N 駅 Ctrl+シフト+W Ctrl+シフト+S Ctrl+シフト+S F2 F5	終了	ア(F) キャンセル てください。 選択した上で、Eclipse IDE メニューより、[ファイ) タート(P) 実行(R) ウインドウ(W) ヘルブ(H) * </td
	② 注意 先行 Airp > [系 □ □ □ □ □ □ □ □ □ □ □ □ □	 う テ作業にて指定したビット数と portDemoTutNativeMFUr 新規(N)] > [COBOL ユニッ イル(F) 編集(E) リファクタリング 新規(N) ファイルを開く(.) ファイルを開く(.) ファイル・システムからプロジェクトを開 Recent Files 閉じる(C) すべて保存(S) 名前を付けて保存(A) すべて保存(E) 前回保存した状態に戻す(T) 移動(V) 名前を変更(M) 更新(F) 行区切り文字の変換(D) 印刷(P) インボート(I) 	同じテンプレートを選 nitTest2 プロジェク ットテスト] を選択し ナビゲート(N) 検索 プ Alt+シフト+N 駅、 Ctrl+シフト+W Ctrl+シフト+S Ctrl+シフト+S Ctrl+シフト+S Ctrl+シフト+S	終了	ア(F) キャンセル てください。 選択した上で、Eclipse IDE メニューより、[ファイ) タート(P) 実行(R) ウィンドウ(W) ヘルプ(H) COBOL プロジェクト COBOL プロジェクト リモート COBOL ブロジェクト リモート COBOL ブレーファイル プロジェクト マロBOL フレーファイル プロジェクト マロBOL コニット テスト プロジェクト COBOL コニット テスト プロジェクト マOBOL コニット テスト プロジェクト COBOL JVM プロジェクト フロBOL コニット テスト プロジェクト アロBOL コニット テスト プロジェクト アロジェクト(R) Enterprise Server Common Web Administration 接続 Directory Server 接続 フォルダー ファイル COBOL コピーファイル COBOL コピーファイル COBOL コピーファイル
☆ エクフポート(∩) C◆ リモート フタンドプロン ファイリ	② 注意 先行 Airp > [系 □ □ □ □ □ □ □ □ □ □ □ □ □	 う テ作業にて指定したビット数と portDemoTutNativeMFUr 新規(N)] > [COBOL ユニ・ イル(F) 編集(E) リファクタリング 新規(N) ファイルを開く(.) ファイルを開く(.) ファイルを開く(.) ファイル・システムからプロジェクトを開 Recent Files 閉じる(C) すべて保存(E) 前回保存した状態に戻す(T) 移動(V) 名前を変更(M) 更新(F) 行区切り文字の変換(D) 印刷(P) インボート(I) Tクスポート(I) TO、 TO、	同じテンプレートを選 nitTest2 プロジェク ットテスト] を選択し ナビゲート(N) 検索 プ Alt+シフト+N 私t+シフト+N てtrl+シフト+S Ctrl+シフト+S Ctrl+シフト+S F2 F5	終了 採了 F <td>ア(F) キャンセル てください。 選択した上で、Eclipse IDE メニューより、[ファイノ タート(P) 実行(R) ウィンドウ(W) ヘルプ(H) *<!--</td--></td>	ア(F) キャンセル てください。 選択した上で、Eclipse IDE メニューより、[ファイノ タート(P) 実行(R) ウィンドウ(W) ヘルプ(H) * </td

 (COBOL ユニット テストの新規作成) ウィンドウが表示されるので [プログラム ユニット テスト] 項目を選択し、 [次へ(N)] をクリックします。



COBOL ユニットテスト 作成するテストのタイプを選択してください	Ů
● プログラム ユニット テスト プログラムを直接呼び出して、入力と出力をアサートできるようにします。CSV ファイルから読み込んだデータを使用して、プログラムを繰り返し実行でき	ます。
○ 自己完結型ユニット テスト Micro Focus Unit Test Preprocessor を使用してテスト ケースを既存のソースコードにコンパイルし、セクションと段落を直接テストできるようにします。	

- 5) 以下の項目を入力し、[終了(F)] をクリックします。
 - [プログラムのユニットテストを作成する]を選択
 - [参照] をクリックして、AirportDemoTutNativeMFUnitTest プロジェクト内の「aircode.cbl」を選択

COBOL ユニット エディタで開くことが	テスト できる COBOL ユニット テスト ファイルを新規	作成します。			DU
含まれるプロジェクト:	AirportDemoTutNativeMFUnitTest2				参照
新規ファイル名:	TestProgram1.cbl				
◎ プログラムのユニッ	/ト テストを作成する				
テスト対象のプログラ	5ム: AirportDemoTutNativeMFUnit/air	code.cbl			参照
○ テンプレートからコ	ニット テストを作成する				
テンプレートを選択:					
Micro Foci	<i>ıs テンプレ−</i> ト				
				<u> - </u>	ートの設定を構成
 」 テンプレートの 場所: 	多照				参照
ファイルシス	テムを選択: default >				
?		< 戻る(B)	次へ(N) >	終了(F)	キャンセル

- 6) AirportDemoTutNativeMFUnitTest2 プロジェクト配下の TestProgram1.cbl を削除してください。
- 7) AirportDemoTutNativeMFUnitTest2 プロジェクトを選択したうえで、Eclipse IDE メニューから [ファイル (F)] > [インポート(I)] をクリックします。



ファ・	イル(F) 編集(E) ソース リファクタリング	ナビゲート(N) 検引
۵,	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開く Recent Files	Alt+シフト+N > >
	閉じる(C) すべて閉じる(L)	Ctrl+W 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)	
2	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)	

8) [一般] > [ファイル・システム] を選択し、[次へ(N)] をクリックします。

選択

ローカル・ファイル・システムから既存のプロジェクトへリソースをインポートします。

インポート・ウイザードの選択(S):	
71ルタ入力	
▼ 🗁 一般	^
- [®] アーカイブ・ファイル [©] ファイル・システム [©] フィル・システム	
◎ 既存プロジェクトをワークスペースへ	
> 🗁 EJB	
> 🗁 Git	
> 🗁 Gradle	\sim
? < 戻る(B) 次へ(N) > 終了(F) キャンセル	

 9) [参照(R)] をクリックした後、サンプルファイルを展開したフォルダー内の DataDriven を選択したうえで、 「input.csv」,「TestProgram1.cbl」を選択、[終了(F)] をクリックします。



ファイル・システム ローカル・ファイル・システムからリソースをインポートします。		
次のディレクトリーから(Y): C:¥vc-elnativetestframework¥	{DataDriven	~ 参照(R)
🗹 🗁 DataDriven	input.csv ☑	
タイプをフィルター(T) すべて選択(S) 選択をす	べて解除(D)	
インポート先フォルダ(L): AirportDemoTutNativeMFUnitT	est2	参照(W)
オプション * 営告を出さずに既存リソースを上書き(O) hップ・レベルのフォルダーを作成(C) 拡張 >>(A) 		
?	< 戻る(B) 次へ(N) >	終了(F) キャンセル

プロジェクトは以下のようになります。



- ✓ ☐ AirportDemoTutNativeMFUnitTest2
 - 🗸 📴 COBOL プログラム
 - IestProgram1.cbl
 - > 🗁 New_Configuration.bin
 - input.csv
- 10) TestProgram1.cbl を開き、プログラムを確認します。

行数	説明
$10 \sim 12$	01 mfu-dd-airport1 is MFU-DD-VALUE external.
	01 mfu-dd-airport2 is MFU-DD-VALUE external.
	01 mfu-dd-distance-km is MFU-DD-VALUE external.
	テストデータファイル "input.csv" から取得する各テストデータの値が格納されます。
53 entry MFU-TC-PREFIX & TEST-TESTAIRCODE.	
	基本的な単体テストプログラムと同様の entry 句の構成ですが、テストデータ1行ごとに呼
	び出されるようになります。
	また、基本的な単体テストプログラムでは、テスト失敗時には MFU-ASSERT-FAIL-Z を
	使用してエラー情報を戻していましたが、データ駆動型テストではこちらは使用できません。
	このため、70 行目では goback での戻り値に MFU-FAIL-RETURN-CODE を設定し
	ています。



77	entry MFU-TC-METADATA-SETUP-PREFIX & TEST-TESTAIRCODE.
	テストデータファイルを読み込みます。

3.1.3.2 MFUnit テストの実行

本テスト対象のプログラムは、環境変数で設定された空港情報が保存されたデータファイルを参照するため、手順内で設 定を行います。

 AirportDemoTutNativeMFUnitTest2 プロジェクト配下の「TestProgram1.cbl」を選択した状態で、マウ スの右クリックでコンテクストメニューを表示し、[実行(R)] > [実行の構成(N)] をクリックします。

Q	Coverage As	>	\vdash		
0	実行(R)	>	CBL	1 COBOL アプリケーション	Alt+シフト+X,N
*	デパッグ(D)	>	œ	2 COBOL ユニット テスト	Alt+シフト+X,F
	プロファイル(P)			実行の構成(N)	
\checkmark	快証				

 直面左側より「COBOL ユニットテスト」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、
 [新規構成(W)]を選択します。

構成の作成、管理、および実行

📑 🖻 🗫 🖹 E 🏹 🗸		このダイアログから起
フィルタ入力		📑 - 選択した種類
🥅 COBOL JVM リモート アプリケーション	^	▶ 💣 - 選択した種類
■ COBOL アプリケーション		💁 - 選択した構成
✓ ■ COBOL ユニット テスト	□ 新規構成()	N)
AirportDemoTutNativeMFUnitTest	■ 新規プロトタ	・ 7イブ(P)
Debug Adapter Launcher	エクスポート	(X)

3) [名前] に "AirtportDemoTutNativeMFUnitTest2"を入力し、 [環境] タブを選択した後、[追加(A)]

をクリックします。

(注:ここで) 環境スクリ	定義された変数は、任意の現在の設定値 「プト内の設定値を上書きします。)	「、または任意の指定された		
変数		値		追加(<u>A</u>)
				編集(<u>E</u>)
				削除(<u>R</u>)
実行する	環境スクリプト:			
場所:				参照
	ファイルがプロジェクト内にある場合、絶対	オパスは相対パスになります。		
パラメータ	:			
	• · · • •			
			前回保管した状態に戻す(V) 適用()

© 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.



:	变数:"dd_airports"				
	直: "¥¥AirportDemoTutNa	tiveMFL	Jnit¥airports.dat"		
ļ	環境変数を追加または変更します				L
	変数: dd_airports				
	值:¥¥AirportDemoTutNativeMFUnit¥air	ports.dat			
	0		ОК	キャンセノ	k
5	きほど追加した dd_airports 環境変	数が表示	示されていることを確認	して、[適月	引(Y)] をクリックし
(R)] をクリックします。				
1		15			1
		♥/ T ハック ま ++ いん	シンハル 2 動的分析 管	CIF / 207.) –
	(注:ここで定義された変数は、任息の現在の設定1) 環境スクリプト内の設定値を上書きします。)	≜、よ/こは1±	息の損走された		
	変数	値			追加(<u>A</u>)
	alal aima anta	V VA:	ant Dama Tut Mativa MEUra	it Vainnant	

5) た後、[実行

🗟 一般 🦻 ソース 🚾 環境 🗉 共通 🔎 実行時 🎙	🥢 デバッグシンボル 🛃 動的分析	i 🧉 CTF 🥒 コンテナー		
(注: ここで定義された変数は、任意の現在の設定値 環境スクリプト内の設定値を上書きします。)	、または任意の指定された			^
変数	値		追加(<u>A</u>)	
dd_airports	¥¥AirportDemoTutNative	//FUnit¥airport	編集(<u>E</u>)	
			削除(<u>R</u>)	
実行する環境スクリプト:				
場所:			参照	
ファイルがプロジェクト内にある場合、絶対	パスは相対パスになります。			
パラメータ:				
				~
	前	回保管した状態に戻す(\	() 適用(<u>Y</u>)	
		実行(<u>R</u>)	閉じる	

実行後に [Micro Focus Unit Testing] ビューを表示すると、以下のように1つテストが失敗しています。

📮 コンソール 🖹 問題 🧔 タスク 🔲 プロパティー 🏠 Table Results 🗗 Filter Definitions 💽 Micro F	ocus Unit Testing × 📴
実行: 4/4 ロフラー: 0 ロ 失敗: 1	
 AirportDemoTutNativeMFUnitTest2 (0 ms) TestProgram1.cbl 	テスト結果
✓ 📾 MFUT_TESTAIRCODE (0 ms)	
🖻 Row #1 (LHR,HND,9591)	
ᠠ Row #2 (LHR,NRT,9592)	
📰 Row #3 (LTN,LGW,820)	



テストデータファイルの 4 行目 LTN, LGW の距離がエラーとなっています。これは、テストデータファイルの 820 が 誤りであり、正しい値は 82 です。テストデータファイルの 820 を 82 に修正したうえで、テストを再実行すると、以下 のように全て成功します。

🖳 コンソール 🔝 問題 🧔	タスク 🔲 プロパティー ኬ Table	Results 🛛 Filter Definitions 🔄	Micro Focus Unit Testing $ imes$
実行: 4/4	⊠ Iラ-: 0	☑ 失敗: 0	
✓ Is AirportDemoTut	NativeMFUnitTest2 (4 ms)		テスト結果
✓	.cbl		
🗸 🛃 MFUT_TEST	TAIRCODE (4 ms)		
🚰 Row #1	(LHR,HND,9591)		
🖅 Row #2	(LHR,NRT,9592)		
🚰 Row #3	(LTN,LGW,82)		
🖅 Row #4	(STN,LCY,47)		

3.1.4 自己完結型テスト

この方式では、テスト対象となるプログラム内にテストコードをコンパイル時に埋め込むことで、より粒度の小さなテストを行 えます。

3.1.4.1 MFUnit テストの作成

1) Eclipse IDE メニューより、[ファイル(F)] > [新規(N)] > [COBOL ユニットテストプロジェクト] 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	プロジ	ェクト	·(P) 実行(R)	ウィンドウ(W)	ヘルプ(H)
	新規(N)	Alt	+シフト-	+N >	2	COBOL プロジ	ェクト	
	ファイルを開く(.)				2	COBOL JL-7	ファイル プロジェク	7 ト
	ファイル・システムからプロジェクトを開	司く			Ê	リモート COBO	L プロジェクト	
	Recent Files			>	ġ	リモート COBO	Lコピーファイル:	プロジェクト
					썉	COBOL/Java	相互運用機能	のプロジェクト
	閉じる(C)		Ctrl+	·W	Ê	COBOL ユニッ	ト テスト プロジェ	クト
	すべて閉じる(L)	Ctrl+	・シフト+	W	B	COBOL JVM 7	フロジェクト	
	保存(S)		Ctrl	+S		リモート COBO	L ユニット テスト	プロジェクト

プロジェクト名」に "AirportDemoTutNativeMFUnitTest3" を入力し、実行環境に合わせたプロジェクトテンプレートを選択した上で、[終了(F)] をクリックします。





📓 リモート スタンドアロン ファイル

🖄 Enterprise Server

📑 COBOL ユニット テスト

Alt+Enter

4) [自己完結型ユニットテスト] を選択して[次へ(N)]をクリックします。

🖆 エクスポート(O)...

プロパティ(R)



COBOL ユニット テスト 作成するテストのタイプを選択してください	Ĵ
○ プログラム ユニット テスト プログラムを直接呼び出して、入力と出力をアサートできるようにします。CSV ファイルから読み込んだデータを使用して、プログラムを繰り返し実行でき	きます。
● 自己完結型ユニットテスト Micro Focus Unit Test Preprocessor を使用してテスト ケースを既存のソースコードにコンパイルし、セクションと段落を直接テストできるようにします。	2

5) テスト対象のプログラム欄の右にある [参照...]をクリックしたうえで、AirportDemoTutNativeMFUnit プロジェク

YEL トの AIFCOOR COBOL ユニットテ 自己完結型ユニットラ	E.CDI を選択し、「終」 スト Fスト用 COBOL ファイルを新規作	(F)] をソリソンしより。 F成します。			O
含まれるプロジェクト:	AirportDemoTutNativeMFU	nitTest3			参照
テスト対象のプログラム:	AirportDemoTutNativeMFU	nit/aircode.cbl			参照
?		< 戻る(B)	次へ(N) >	終了(F)	キャンセル

以下のようなコピーブックファイルが作成されます。

MFUPD_aircode.cpy ×
aircode.cbl 🕨 🖺 MFUPD_aircode.cpy
•••••*A•1•B••••••2•••••3••••
entry "MFUT_TESTAIRCODE". goback

自己完結型のテストでは、このコピーブックがテスト対象プログラム(本手順では aircode.cbl)内にコンパイル 時に埋め込まれます。このため、テストコードに必要なデータ項目は LINKAGE SECTION で定義された項目で はなく、WORKING-STORAGE SECTION 内で定義された実体のある項目で記述されている必要があります。 サンプルファイルを展開したフォルダー内の Self¥MFUPD_aircode.cpy の内容で、MFUPD_aircode.cpy を 上書きしてください。



entry "MFUT_TESTAIRCODE".		
perform open-airfile		
if fstat-1 not= 0		
goback returning 1		
end-if		
goback		
entry "MFUT_TESTAIRCODE2".		
perform close-airfile		
goback		

この例では、open-airfile と close-airfile section のテストを行っています。open-airfile のテストコードを 確認すると、fstat-1 項目の値を直接参照して処理結果を評価していることが確認できます。一方、closeairfile は特に判定材料がないため、常にテストは成功します。

3.1.4.2 MFUnit テストの実行

本テスト対象のプログラムは、環境変数で設定された空港情報が保存されたデータファイルを参照するため、手順内で設 定を行います。

 AirportDemoTutNativeMFUnitTest3 プロジェクト配下の「aircode.cbl」を選択した状態で、マウスの右ク リックでコンテクストメニューを表示し、[実行(R)] > [実行の構成(N)] をクリックします。

Q	Coverage As	>	\vdash		
0	実行(R)	>	CBL	1 COBOL アプリケーション	Alt+シフト+X,N
栨	デパッグ(D)	>	œŋ	2 COBOL ユニット テスト	Alt+シフト+X,F
	プロファイル(P)	>		実行の構成(N)	
\checkmark	快証				

2) 画面左側より「COBOL ユニットテスト」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、

[新規構成(W)]を選択します。



3) [名前] に "AirtportDemoTutNativeMFUnitTest3"を入力し、 [環境] タブを選択した後、[追加(A)]

をクリックします。

名前(N) AirtportDemoTutNativeMFUnitTest3						
🕓 一般 🦭 ソース 写 環境 🗉 共通 🔊 実行時 🦉	✓デバッグシンボル ⑤ 動的分析 營 CTF ┙ コンテ;	+-				
(注:ここで定義された変数は、任意の現在の設定値、または任意の指定された 環境スクリプト内の設定値を上書きします。)						
変数	値	追加(A)				
		編集(E)				
		削除(R)				
実行する環境スクリプト:						



4) 以下の情報を入力し、[OK]をクリックします。

変数:	"dd_	_airports"
-----	------	------------

值: "..¥..¥AirportDemoTutNativeMFUnit¥airports.dat"

環境家	数を追加または変更します
変数:	dd_airports
値:	¥¥AirportDemoTutNativeMFUnit¥airports.dat
?	OK キャンセル

5) さきほど追加した dd_airports 環境変数が表示されていることを確認して、[適用(Y)] をクリックした後、[実行

(R)] をクリックします。

名前(N): AirtportDemoTutNativeMFUnitTest3		
🕓 一般 🦻 ソース 🚾 環境 🔲 共通 👂 実行時 🕏	🦻 デバッグシンボル 🛃 動的分析 🦉 CTF 🥑 コンテナ	-
(注:ここで定義された変数は、任意の現在の設定値 環境スクリプト内の設定値を上書きします。)	こ、または任意の指定された	^
変数	値	追加(A)
dd_airports	¥¥AirportDemoTutNativeMFUnit¥airport	編集(E)
		削除(R)
実行する環境スクリプト:		
場所:		参照
ファイルがプロジェクト内にある場合、絶対	けパスは相対パスになります。	
パラメータ:		
		~
	前回保管した状態に戻す	で(V) 適用(Y)
	実行(R)	閉じる

実行後に [Micro Focus Unit Testing] ビューを表示すると、以下のように2つのテストがともに成功し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 🗗 Fi	ilter Definitions $\[\] {\[\] { { I } } } { { { I } } } { { I } } { { I } } { { o } } { { I } } { { o } } { { I } } { I } } { I }$
実行: 2/2 ロラー: 0 ロラー: 0	5敗: 0
 AirportDemoTutNativeMFUnitTest3 (0 ms) aircode.cbl MFUT_TESTAIRCODE (0 ms) MFUT_TESTAIRCODE2 (0 ms) 	テスト結果



3.2 コマンドラインからの実行

MFUnit によるテストは、Eclipse 上の画面からではなく、コマンドライン上からも行なうことができます。従来のスタイルでのテスト 作業の効率化を図ることができ、Jenkins などの CI ツールと連携する事で、テストの自動実行を行なえるため、品質担保や 作業工数の削減が見込めます。

どのテスト方式でも実行方法は同じであるため、ここでは 3.1.2 で作成したテストプログラムをコマンドラインから実行する方法について学びます。

1) スタートメニューより、Micro Focus Visual COBOL 配下の Visual COBOL コマンドプロンプト をクリックします。

	м		
	Micro Focus	License Manager	
	Micro Focus	Visual COBOL	
8	Enterprise	Server Administration	
Ľ	GpenESQL	構成ユーティリティ	
23	🚿 Visual COB	BOL for Eclipse	
	Visual COB	3OL コマンドプロンプト (32-	·
203	Visual COB	30L コマンドプロンプト (64-	·
Ф	📝 クラシック デ	ータファイル ツール	
4	ا 🗧 م		
注意)			

3.1.2 で指定したビット数と同じテンプレートを選択してください。

2) 作業フォルダを作成し、作成したフォルダに移動します。

C:¥>mkdir VCCommandTutorial C:¥>cd VCCommandTutorial C:¥VCCommandTutorial>

- 3) 3.1 で利用したワークスペースフォルダを ECLIPSE_WORKSPACE に指定した上で、下記コマンドを実行します。
 - set ECLIPSE_WORKSPACE=c:\u00e4workspace_tut_native_mfunit
 - cobol %ECLIPSE_WORKSPACE%¥AirportDemoTutNativeMFUnit¥aircode.cbl;
 - cobol %ECLIPSE_WORKSPACE%¥AirportDemoTutNativeMFUnitTest¥TestProgram1.cbl sourceformat(variable);
 - cbllink -D aircode.obj
 - cbllink -D TestProgram1.obj

C: ¥VCCommandTutorial>set ECLIPSE_WORKSPACE=c: ¥workspace_tut_native_mfunit

C:¥VCCommandTutorial>cobol %ECLIPSE_WORKSPACE%¥AirportDemoTutNativeMFUnit¥aircod e.cbl;

(実行中のログを省略)

C:¥VCCommandTutorial>cobol %ECLIPSE_WORKSPACE%¥AirportDemoTutNativeMFUnitTest¥T estProgram1.cbl sourceformat(variable);

(実行中のログを省略)

C:¥VCCommandTutorial>cbllink -D aircode.obj

(実行中のログを省略)

C: ¥VCCommandTutorial > cbllink -D TestProgram1.obj



(実行中のログを省略)			
C:¥VCCommandTutorial>			
以下のファイルが作成されていることを確認します。			
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 722D-D996 です			
C:¥VCCommandTutorial のディレクトリ			
2024/08/06 15:59 <dir> .</dir>			
2024/08/06 15:59 <dir></dir>			
2024/08/06 15:59 17,408 aircode.dll			
2024/08/06 15:59 10,159 aircode.obj			
2024/08/06 15:59 16,384 TestProgram1.dll			
2024/08/06 15:59 8,164 TestProgram1.obj			
4 個のファイル 52,115 バイト			
2 個のディレクトリ 103,178,993,664 バイトの空き領域			
C:¥VCCommandTutorial>			

- 4) プロンプト上で下記コマンドを実行し、MFUnit を実行します。
 - set dd_airports=%ECLIPSE_WORKSPACE%¥AirportDemoTutNativeMFUnit¥airports.dat
 - mfurun -report:junit -outdir:result TestProgram1.dll

C:¥VCCommandTutoria	l>set dd_airports=%ECLIPSE_WORKSPACE%¥AirportDemoTutNativeMF
Unit¥airports.dat	
C:¥VCCommandTutoria	l>mfurun -report:junit -outdir:result TestProgram1.dll
Micro Focus COBOL - I	mfurun Utility
Unit Testing Framewor	k for Windows/Native/64
Fixture : TestProgram1	
Test Run Summary	
Overall Result	Passed
Tests run	2
Tests passed	2
Tests failed	0
Total execution time	0
C:¥VCCommandTutoria	>

outdir オプションにより、出力結果が result フォルダーに保存されます。

ファイル ホーム 共有	表示	
$\leftarrow \rightarrow \checkmark \uparrow \blacksquare \rightarrow$ PC :	> ローカル ディスク (C:) > VCCommandTutorial > result	~ Ū
> ProgramData ^ VCCommandTu	TEST-MFUT_TESTAIRCODE.xml TestProgram1-report.txt	TEST-MFUT_TESTDISTANCE.xml
📕 result		
> 📙 vc-elnativetesti		



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。