



単体テスト支援 ファイルシミュレーション機能の活用

COBOL2002 Developer Professional 単体テスト支援

株式会社 日立製作所

本資料はCOBOL2002製品群をより便利に活用するためのガイドです。

本資料は以下の製品が対象です。

- ◆ COBOL2002 Developer Professional 03-04以降
- ◆ COBOL2002 Developer Professional(64) 03-04以降

本資料は「COBOL活用ガイド 単体テスト支援ファーストステップガイド」の内容を前提としています。

「単体テスト支援ファイルシミュレーション機能の活用」ではファイルアクセスを行うCOBOLプログラムの単体テスト方法を紹介します。

演習内容は以下です。

- ◆ ファイルシミュレーション機能を用いた単体テスト

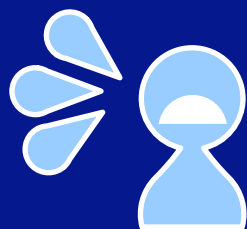
演習にかかる時間の目安は約30分程度です。

Contents

1. 単体テスト支援
2. ファイルシミュレーション機能で単体テストを実行してみよう
3. まとめ

1. 単体テスト支援





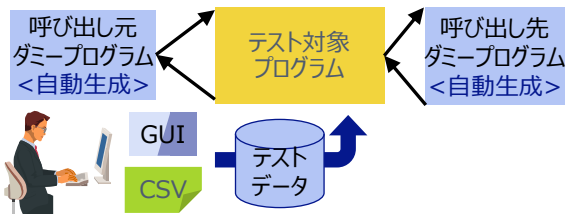
バグを早期に発見したい！

手作業のテストではもう対応できない！

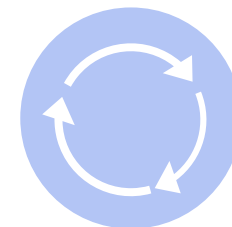
そのお悩み 単体テスト支援が解決します！



準備が簡単に！



結果確認が簡単に！

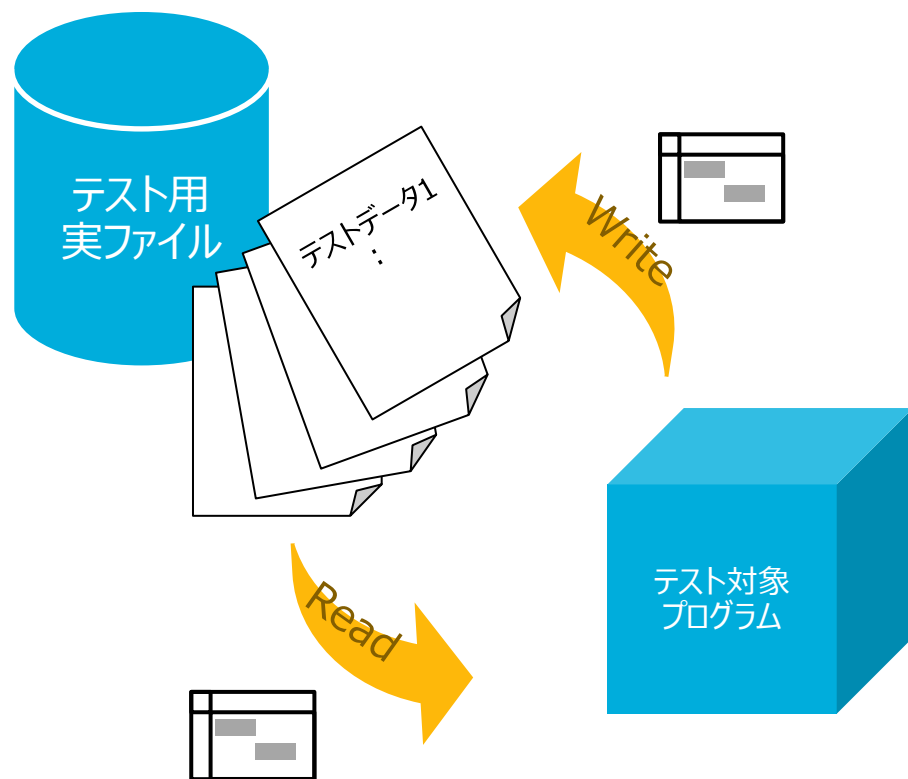


繰り返しテストが簡単に！



1-2. ファイル入出力を行うプログラムの単体テスト

従来、ファイル入出力を行うプログラムに対して単体テストを実施するには準備などが大変でした。

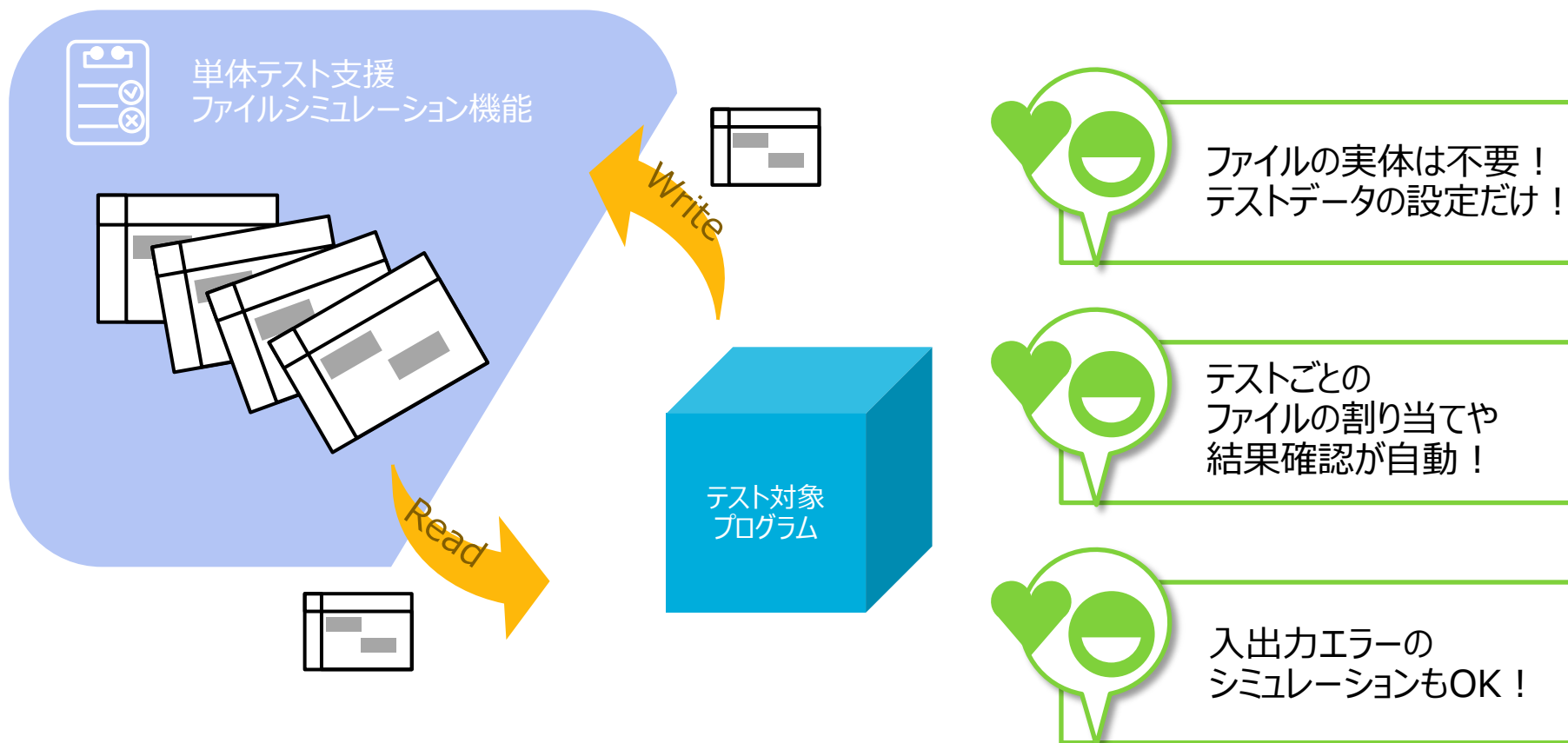


テストケースの分だけ
ファイルの実体を用意
するのが大変！

テストケースごとに
ファイルの割り当てや
結果の確認が面倒！

入出力エラーの
再現が大変！

ファイルシミュレーション機能を使うことで準備などの手間が削減でき、開発効率の向上に役立ちます。

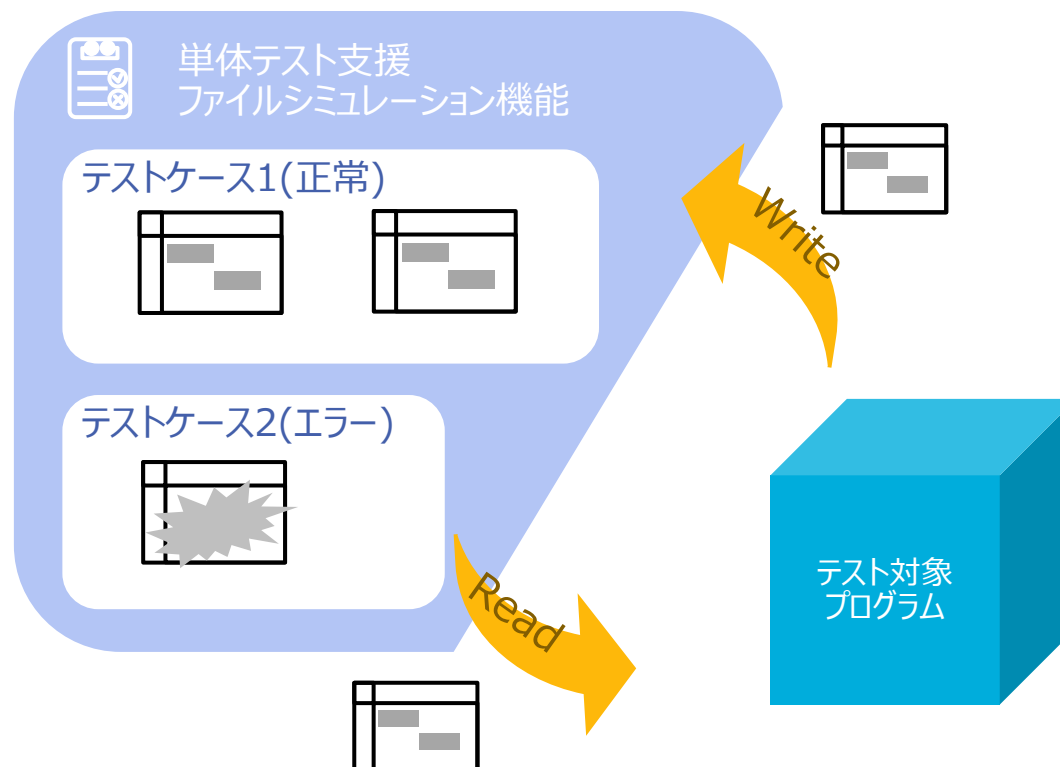


2. ファイルシミュレーション機能で 単体テストを実行してみよう



ファイルシミュレーション機能を用いてファイルアクセスを行うプログラムの単体テストを行います。

本演習ではファイルの読み込み/書き込みを行うプログラムに対して、「正常に読み書きができたケース」と「入出力エラーが発生したケース」の2つのケースをテストします。

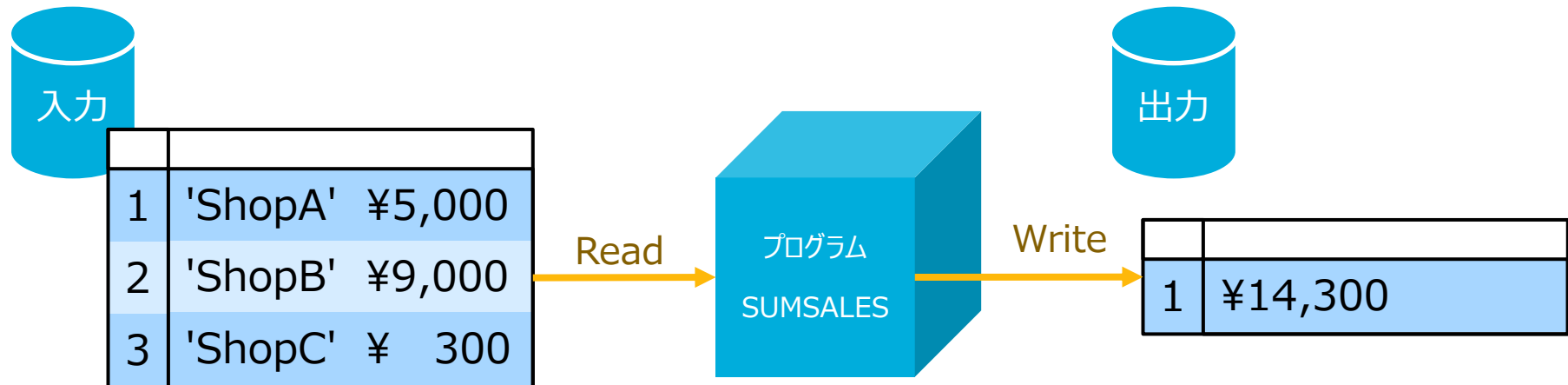


テスト対象のプログラム[SUMSALES]について説明します。

機能仕様：

各店舗の売り上げデータを集計するプログラムです。

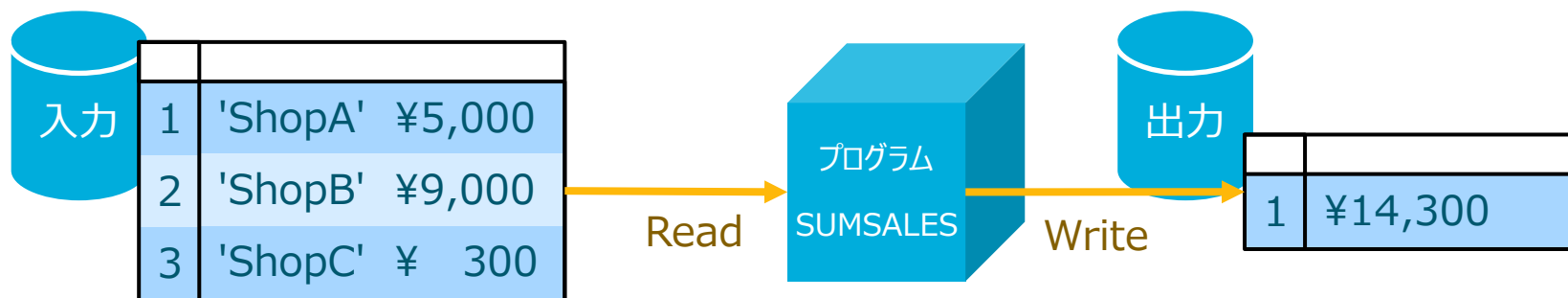
1つの店舗の売り上げデータは1つのレコードに記載され
1つのファイルに全店舗の売り上げデータが記載されます。
これを入力ファイルとして合計値を計算し、その値を出力ファイルに出力します。



この演習では2つのテストケースを作成します。

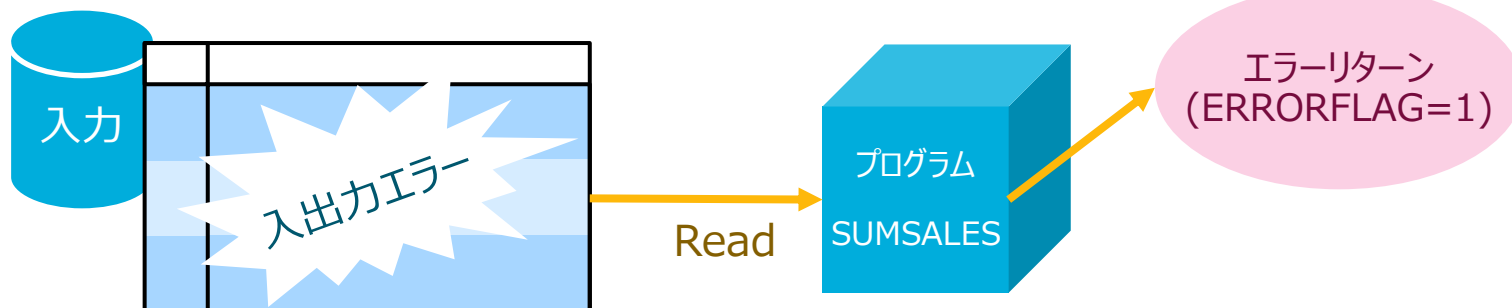
テストケース1 [正常ケース]

3店舗の売り上げデータをテストデータとして入力ファイルのレコードに設定し、その集計結果を期待値として確認します。

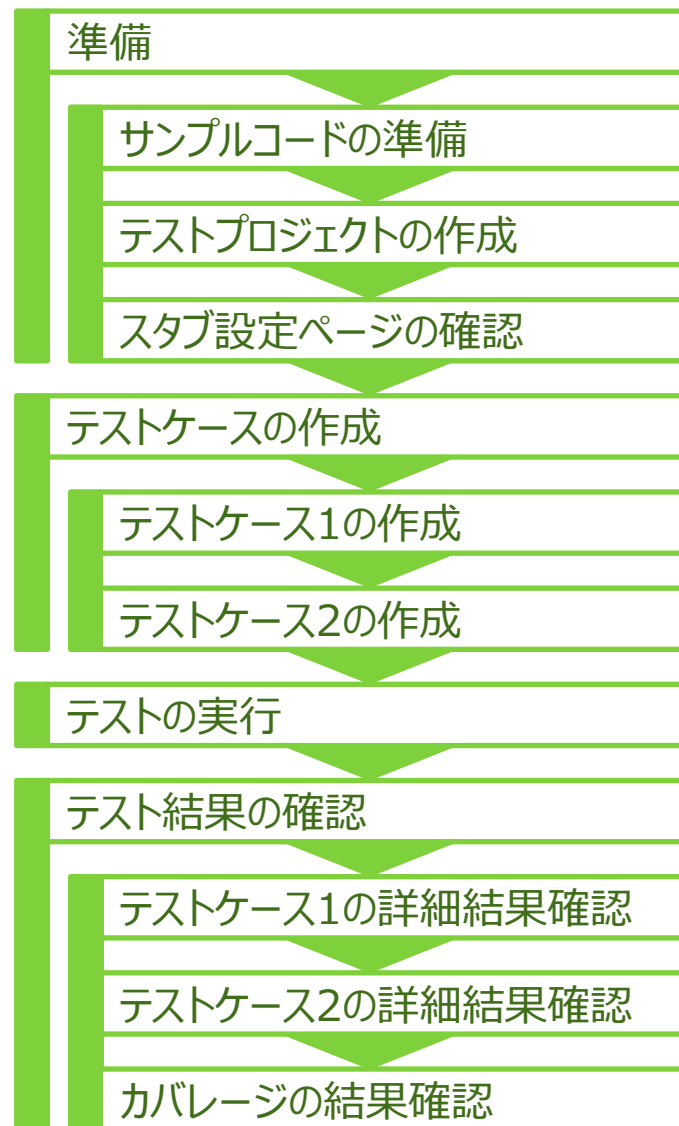
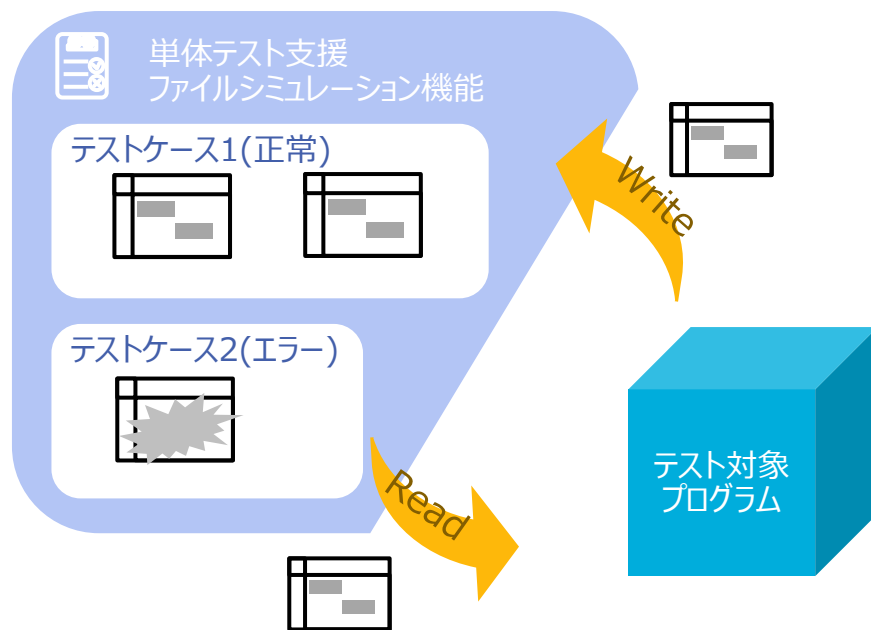


テストケース2 [入出力エラーケース]

入力ファイルをReadしたときに入出力エラーが発生するようにテストの設定を行い、プログラムがエラーリターンすることを確認します。



大まかな手順です。



Progress

準備

テストケース1作成

テストケース2作成

実行

結果確認

サンプルコードの準備

サンプルコードのファイルを任意のフォルダに作成します。

以降は例として以下に置いたものとして説明します。

D:¥cblsample

準備するファイルは以下の1つです。

D:¥cblsample¥sumsales.cbl

TIPS

サンプルコードは「付録1. サンプルコード」に記載しています。

テキストをコピー&ペーストして、サンプルコードのファイルを作成してください。

2-6. テストプロジェクトの作成

Progress

準備

テストケース1作成

テストケース2作成

実行

結果確認

単体テスト支援を起動します。
スタートメニューから「COBOL2002 Professional Tool Kit」→「単体テスト支援」をクリックします。

スタートページから「新しいプロジェクトを作る」をクリックします。

「新しいプロジェクトを作る」ダイアログで以下の値を入力して、「作成」をクリックします。

テストプロジェクト名

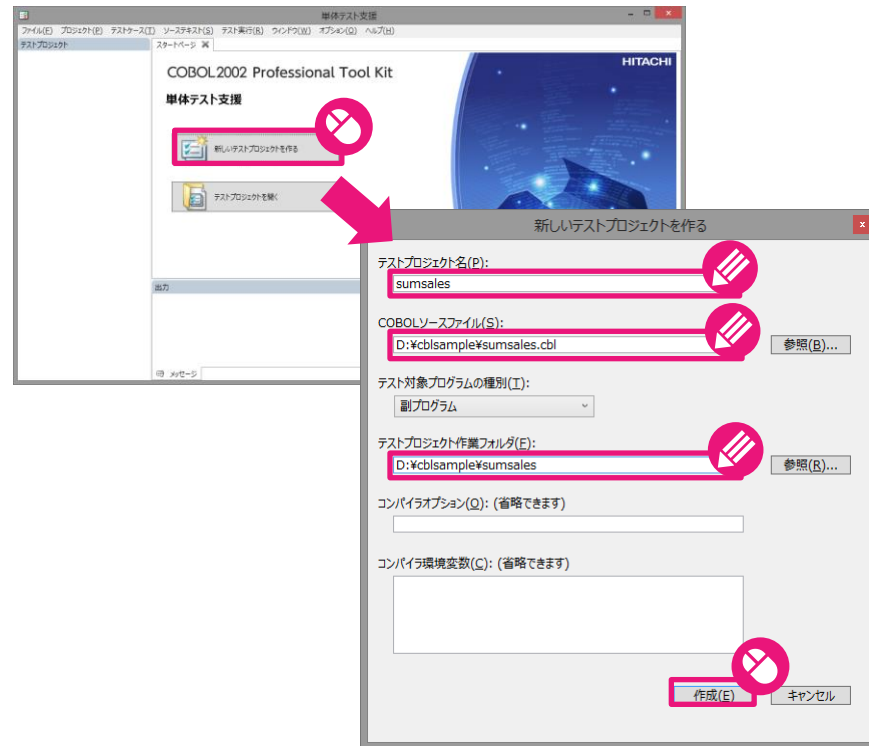
sumsales

COBOLソースファイル

D:¥cblsample¥sumsales.cbl

テストプロジェクト作業フォルダ

D:¥cblsample¥sumsales



TIPS

テストプロジェクト作成時に、テストプロジェクト作業フォルダが存在しない場合には、作成するかどうかを問い合わせるダイアログが表示されるので、「OK」をクリックしてください。

2-7. スタブ設定ページの確認

Progress

準備

テストケース1作成

テストケース2作成

実行

結果確認

ファイル入出力処理を含むプログラムでは自動的にファイルシミュレーションを行うよう設定されます。ここではその設定がされていることを確認します。

メニューから「プロジェクト」→「スタブ設定ページを表示」をクリックします。

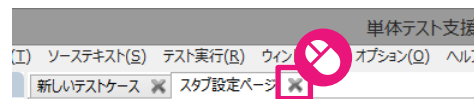
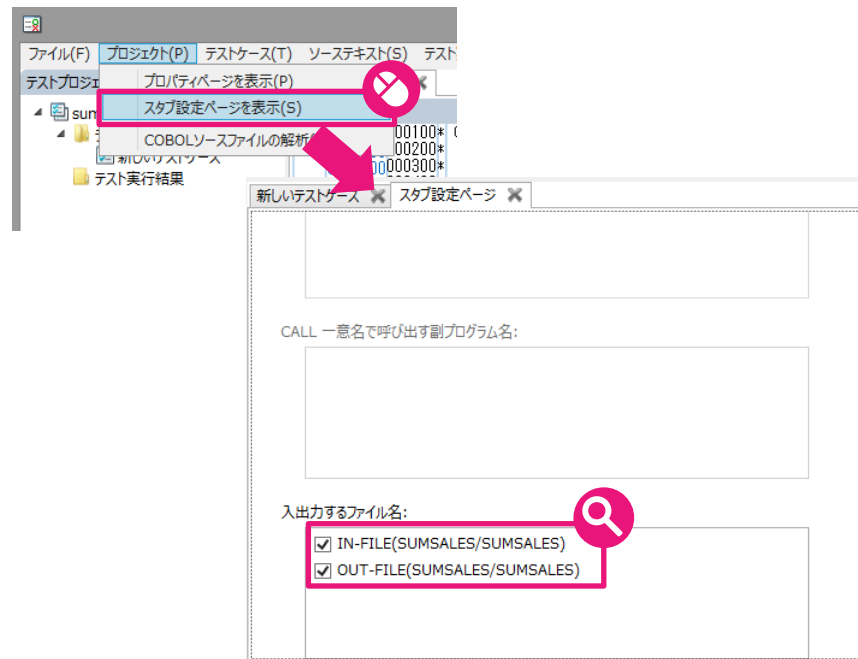
「スタブ設定ページ」では副プログラムやファイルのシミュレーションの設定や確認を行えます。

「入出力するファイル名」に"IN-FILE"と"OUT-FILE"のチェックボックスがオンになっていることを確認します。

このチェックボックスがオンになっているファイル名に対して、単体テスト実行時にファイルシミュレーションが行われます。

確認が完了したら「スタブ設定ページ」画面を閉じます。

「スタブ設定ページ」のタブのバツマークをクリックします。



TIPS

単体テスト実行時に実ファイル入出力を行いたい場合には「スタブ設定ページ」の「入出力するファイル名」欄から該当するファイルのチェックボックスをオフにします。

2-8. テストケース1の作成(1)

Progress

準備

テストケース1作成

テストケース2作成

実行

結果確認

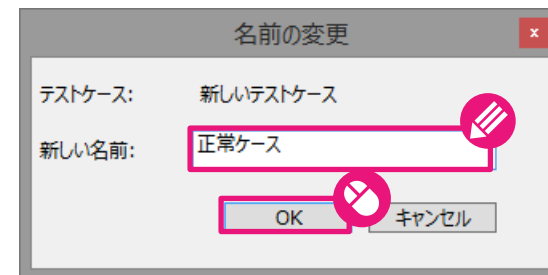
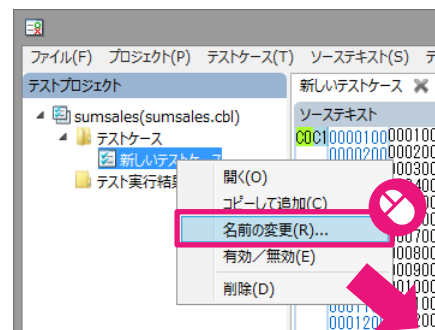
これからテストケースを2つ作成します。
まずは正常な処理を確認するテストケースです。

画面左側の「テストプロジェクト」のツリーから"新しいテストケース"を右クリックして、右クリックメニューから「名前の変更」をクリックします。

「名前の変更」ダイアログで新しい名前に以下の値を入力して「OK」をクリックします。

新しい名前

正常ケース



2-9. テストケース1の作成(2)

Progress

準備

テストケース1作成

テストケース2作成

実行

結果確認

次にこのテストケースにテストデータを設定します。
このテストケースでは入力ファイルおよび出力ファイルのテストデータを設定します。

画面右側の「設定」の「ファイル」タブを選びます。

まず入力のIN-FILEのテストデータを設定します。
ここでは3つのレコードと終了条件を設定します。

1つ目のレコードのテストデータを設定します。
「テストデータ(代入値)」のREAD文の"1"個目の
テストデータとして以下のデータを入力します。

SHOPNAME

'ShopA'

SALES

5000

設定

プログラム開始時 プログラム終了時 **ファイル**

ファイルシミュレーションの設定をしてください。

ファイル名(E): IN-FILE(SUMSALES/SUMSALES)

テストデータ設定対象の入出力文

オープンモード(O): INPUT

入出力文(I): READ

レコード領域(B): INDATA

テストデータ(代入値)

READ文の 1 個目のテストデータ 削除(D)

データ項目	値設定
1 INDATA	

終了条件(E): なし

テストデータ(代入値)

READ文の 1 個目のテストデータ 削除(D)

データ項目	値設定
01 INDATA	
05 SHOPNAME	'ShopA'
05 SALES	5000

終了条件(E): なし

2-10. テストケース1の作成(3)

Progress

準備

テストケース1作成

テストケース2作成

実行

結果確認

2つ目のレコードのテストデータを設定します。
まずREAD文の"1"個目のテストデータのプルダウンをクリックし、リストの末尾の"追加"をクリックします。

"2"個目のテストデータを設定する画面が表示されるので、以下のデータを入力します。

SHOPNAME

'ShopB'

SALES

9000

3つ目のレコードのテストデータを設定します。
2つ目のレコードの設定と同様に、"3"個目テストデータを追加して、以下のデータを入力します。

SHOPNAME

'ShopC'

SALES

300

テストデータ(代入値)
READ文の 1 個目のテストデータ 削除(D)

データ項目	値設定
01 INI	
05 SHOPNAME	'ShopA'
05 SALES	5000

追加

テストデータ(代入値)
READ文の 2 個目のテストデータ 削除(D)

データ項目	値設定
01 INDATA	
05 SHOPNAME	'ShopB'
05 SALES	9000

終了条件(E): なし

テストデータ(代入値)
READ文の 3 個目のテストデータ 削除(D)

データ項目	値設定
01 INDATA	
05 SHOPNAME	'ShopC'
05 SALES	300

終了条件(E): なし

2-11. テストケース1の作成(4)

Progress

準備

テストケース1作成

テストケース2作成

実行

結果確認

終了条件を設定します。
このテストケースでは正常に読み込みが完了することを確認するため、「終了条件」のプルダウンから以下を選択します。

終了条件

AT END指定の文を実行

入力ファイルIN-FILEに対するテストデータの設定は完了しました。
次は出力ファイルOUT-FILEへの設定を行います。

テストデータ(代入値)

READ文の 3 個目のテストデータ 削除(D)

データ項目	値設定
01 INDATA	
05 SHOPNAME	'ShopC'
05 SALES	300

終了条件(E):

- AT END指定の文を実行
- AT END指定の文を実行
- 入出力エラーを発生させる
- なし

1sales.utsproj)
:umsales.cbl)

2-12. テストケース1の作成(5)

Progress

準備

テストケース1作成

テストケース2作成

実行

結果確認

OUT-FILEへテストデータを設定します。
「ファイル名」のプルダウンから"OUT-FILE"を選択
します。

「テストデータ(期待値)」のWRITE文の"1"個目
のテストデータに、以下の値を入力します。

TOTALSALES
14300

設定

プログラム開始時 プログラム終了時 ファイル

ファイルシミュレーションの設定をしてください。

ファイル名(F): IN-FILE(SUMSALES/SUMSALES)

テストデータ設定対 IN-FILE(SUMSALES/SUMSALES)

オープンモード(O): OUT-FILE(SUMSALES/SUMSALES)

入出力文(I): READ

レコード領域(R): INDATA

テストデータ(期待値)

WRITE文の 1 個目のテストデータ 削除(D)

データ項目	値の表示	期待値
01 OUTDATA	<input checked="" type="checkbox"/>	
05 TOTALSALES	<input checked="" type="checkbox"/>	14300

終了条件(E): なし

2-13. テストケース2の作成(1)

Progress

準備

テストケース1作成

テストケース2作成

実行

結果確認

2つ目のテストケースを作成します。
このテストケースでは入出力エラーを確認します。

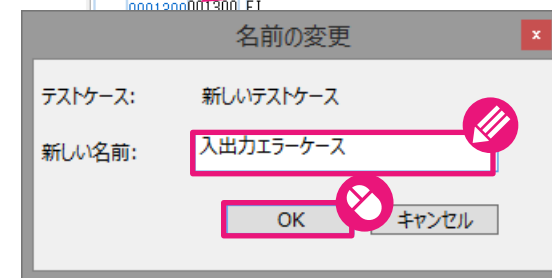
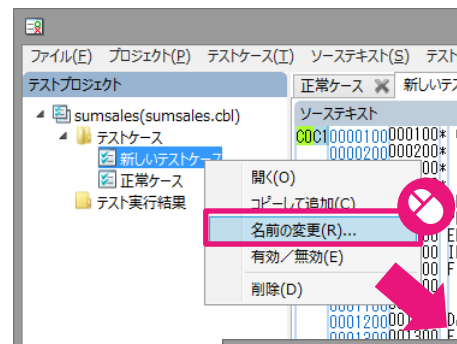
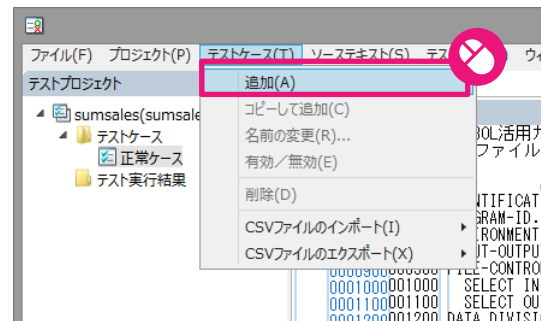
メニューから「テストケース」→「追加」をクリックします。これで「テストプロジェクト」のツリーに"新しいテストケース"が1つ追加されます。

画面左側の「テストプロジェクト」のツリーから"新しいテストケース"を右クリックして、右クリックメニューから「名前の変更」をクリックします。

「名前の変更」ダイアログで新しい名前に以下の値を入力して「OK」をクリックします。

新しい名前

入出力エラーケース



2-14. テストケース2の作成(2)

Progress

準備

テストケース1作成

テストケース2作成

実行

結果確認

このテストケースにテストデータを設定します。
画面右側の「設定」の「ファイル」タブを選択します。

入力ファイルで入出力エラーを発生させるので、
ファイル名が"IN-FILE"であることを確認します。

1回目のREAD文で入出力エラーを発生させるため、
レコードの設定は不要となります。
そのため「削除」ボタンをクリックし、レコードの設定
を削除します。
削除するとREAD文の"- "個目のテストデータと表示
されます。

「終了条件」を設定します。
プルダウンから以下の値を選択します。

終了条件

入出力エラーを発生させる

設定

プログラム開始時 プログラム終了時 **ファイル**

ファイルシミュレーションの設定をしてください。

ファイル名(F): **IN-FILE(SUMSALES/SUMSALES)**

テストデータ設定対象の入出力文

オープンモード(O): INPUT

入出力文(I): READ

レコード領域(R): INDATA

テストデータ(代入値)

READ文の 1 個目のテストデータ **削除(D)**

データ項目	値設定
▷ 01 INDATA	

終了条件(E): なし

テストデータ(代入値)

READ文の - 個目のテストデータ **削除(D)**

データ項目	値設定
▷ 01 INDATA	

終了条件(E): なし

AT END指定の文を実行

入出力エラーを発生させる

なし

tsproj)

2-15. テストケース2の作成(3)

Progress

準備

テストケース1作成

テストケース2作成

実行

結果確認

入出力エラーが発生するとデータ項目の ERRORFLAGに1が設定されプログラムの返却値となります。この値を確認します。

画面右側の「設定」の「プログラム終了時」タブを選択します。

「期待値」として以下の値を設定します。

ERRORFLAG

1

これでテストケースの設定は全て完了しました。

設定

プログラム開始時 **プログラム終了時** ファイル

テストデータ

プログラムを終了するとき

値を表示するデータ項目の選択と、期待値の設定をしてください。

表示(S): 手続き部で参照

検索(E):

データ項目	値の表示	期待値
▷ 01 INDATA	<input type="checkbox"/>	
▷ 01 OUTDATA	<input type="checkbox"/>	
01 LOOPFLAG	<input type="checkbox"/>	
01 SUMSALES	<input type="checkbox"/>	
01 ERRORFLAG	<input checked="" type="checkbox"/>	1

2-16. テストの実行

Progress

準備

テストケース1作成

テストケース2作成

実行

結果確認

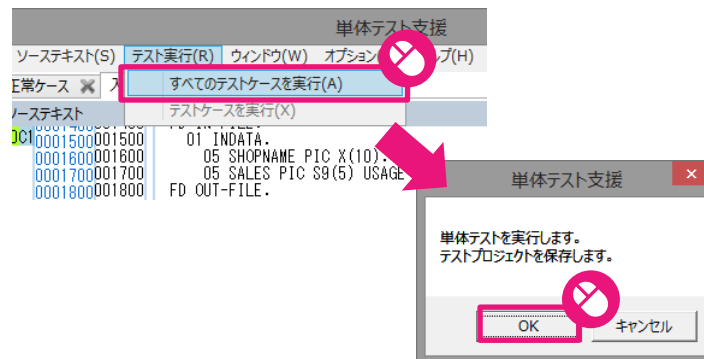
単体テストを実行します。
メニューから「テスト実行」→「すべてのテストケース
を実行」をクリックし、表示されるダイアログで
「OK」をクリックします。

単体テストが実行されますので少し待ちます。

すべてのテストケースの実行が完了すると画面下
部の出力画面にテストプロジェクト結果レポートが
表示されます。
このレポートには実行したテストケースの結果が一
覧で表示されます。

判定結果がすべて"OK"となりテスト完了です。

テスト自体はこれで完了ですが、実行したテスト
ケースの詳細結果を見てみましょう。



出力

テストプロジェクト結果レポート

コンパイラ情報: COBOL2002(64) 03-04
ソースファイル名: sumsales.cbl
開始時間: 2016/06/30 12:19:03
終了時間: 2016/06/30 12:19:03
経過時間: 00:00:00
エビデンスファイル格納フォルダ: D:\%cblsample%\sumsales%\Result%\Result_20160630_121902

テストケース 合計:2件

OK 2件(100%) NG 0件(0%) 未判定 0件(0%) エラー 0件(0%)

#	テストケース	判定結果
1	正常ケース	OK
2	入出力エラーケース	OK

メッセージ Result_20160630_121902

2-17. テストケース1の詳細結果確認(1)

Progress

準備

テストケース1作成

テストケース2作成

実行

結果確認

詳細な結果ではテストケースの各データの値などの情報を表示できます。
まず1つ目の"正常ケース"の詳細結果を見てみます。

画面左側の「テストプロジェクト」ツリーのテスト実行結果から、先ほど実行した結果のツリーを展開して結果レポートの"正常ケース"をダブルクリックします。

画面下部に"正常ケース"のテストケース結果レポートが表示されます。

The screenshot shows the Hitachi test environment interface. The top part displays a tree view of the test project 'sumsales(sumsales.cbl)'. Under 'テスト実行結果', the 'Result_20160630_1219' folder is expanded, showing '結果レポート' (Result Report) and '正常ケース' (Normal Case). A red circle highlights the '正常ケース' item, with a red arrow pointing to the '出力' (Output) button below it.

The bottom part of the screenshot shows the '出力' (Output) window displaying the 'テストケース結果レポート' (Test Case Result Report) for the '正常ケース' (Normal Case). The report shows the test result as 'OK'.

判定結果: OK

ファイル名	オープンモード	出力文	出力文の場所	判定結果
OUT-FILE(SUMSALES/SUMSALES)	OUTPUT	WRITE	0005500.1	OK

ファイル名	オープンモード	入力文	入力文の場所	判定結果
IN-FILE(SUMSALES/SUMSALES)	INPUT	READ	0004500.1	---
IN-FILE(SUMSALES/SUMSALES)	INPUT	READ	0004500.1	---
IN-FILE(SUMSALES/SUMSALES)	INPUT	READ	0004500.1	---
IN-FILE(SUMSALES/SUMSALES)	INPUT	READ	0004500.1	終了条件

2-18. テストケース1の詳細結果確認(2)

Progress

準備

テストケース1作成

テストケース2作成

実行

結果確認

出力ファイルに出力されたデータの値を確認します。
「出力文の実行結果」から項目を右クリックして、
右クリックメニューから「結果を表示」をクリックします。

「ファイルシミュレーションの結果」ダイアログが表示
されます。
このダイアログではシミュレーションを行ったレコード
の各要素の値を見ることができます。
"正常ケース"ではデータ項目"TOTALSALES"の
値が14300で設定した期待値と一致していること
から判定結果が"OK"となっていることが分かります。

TIPS

結果の値を表示するダイアログは「出力文の実行結果」だけでなく「入力文の実行結果」の各項目についても表示できます。

出力文の実行結果

ファイル名	オープンモード	出力文	出力文の場所	判定結果
OUT-FILE(SUMSALES/SUMSALES)	OUTPUT	WRITE	0005500.1	OK

結果を表示(R)

ファイルシミュレーションの結果

ファイル名: OUT-FILE(SUMSALES/SUMSALES)
オープンモード: OUTPUT
出力文: 1回目のWRITE文
出力文の場所: 0005500.1
使用した期待値: 1個目のテストデータ

結果: 16進表示

データ項目	値	期待値	判定結果
01 OUTDATA			---
05 TOTALSALES	+000014300	14300	OK

警告:

前の結果(P) 次の結果(N) 閉じる(C)

2-19. テストケース2の詳細結果確認(1)

Progress

準備

テストケース1作成

テストケース2作成

実行

結果確認

2つ目のテストケース"入出力エラーケース"の詳細結果を確認します。

画面左側の「テストプロジェクト」ツリーのテスト実行結果から、先ほど実行した結果のツリーを展開して結果レポートの"入出力エラーケース"をダブルクリックします。

画面下部に"入出力エラーケース"のテストケース結果レポートが表示されます。

The screenshot shows the Hitachi test environment interface. On the left, the 'テストプロジェクト' (Test Project) tree is expanded to show the '入出力エラーケース' (Input/Output Error Case) under the '結果レポート' (Result Report) folder. A red arrow points from this folder to the detailed result report below.

The detailed result report is titled 'テストケース結果レポート' (Test Case Result Report) and shows the following information:

- コンパイラ情報: COBOL2002(64) 03-04
- ソースファイル名: sumsales.cbl
- テストケース: 入出力エラーケース
- 判定結果: OK

The report also includes a table for 'プログラムの終了時(SUMSALES/SUMSALES)' (At program termination) and a table for '入出力の実行結果' (Input/Output Execution Result).

テストデータ設定箇所	レベル	データ項目	値	期待値	期待値	判定結果
プログラム終了時(SUMSALES/SUMSALES)	01	ERRORFLAG	"1"	1		OK

ファイル名	オープンモード	出力文	出力文の場所	判定結果
IN-FILE(SUMSALES/SUMSALES)	INPUT	READ	0004500.1	終了条件

2-20. テストケース2の詳細結果確認(2)

Progress

準備

テストケース1作成

テストケース2作成

実行

結果確認

入出力ファイルで入出力エラーを発生させたときの情報を確認します。
「出力文の実行結果」から項目を右クリックして、右クリックメニューから「結果を表示」をクリックします。

「ファイルシミュレーションの結果」ダイアログが表示されます。
"入出力エラーケース"では1回目のREAD文をプログラムが実行したタイミングで"入出力エラーを発生させる"動作がシミュレーションとして行われたことが確認できます。

また「結果の値と判定結果」のデータ項目"ERRORFLAG"も期待値と値が一致し、判定結果が"OK"となっていることが分かります。

① 入力文の実行結果

ファイル名	オープンモード	入力文	入力文の場所	判定結果
IN-FILE(SUMSALES/SUMSALES)	INPUT		0004500.1	終了条件

② ファイルシミュレーションの結果

ファイル名: IN-FILE(SUMSALES/SUMSALES)
 オープンモード: INPUT
 入力文: 1回目のREAD文
 入力文の場所: 0004500.1
 設定した値: なし(入出力エラーを発生させる)

結果: 16進表示

データ項目	値	判定結果
警告:		

前の結果(P) 次の結果(N) 閉じる(C)

出力

テストケース結果レポート

コンパイル情報: COBOL2002(64) 03-04
 ソースファイル名: sumsales.cbl
 テストケース: 入出力エラーケース

判定結果: OK

③ 結果の値と判定結果

テストデータ設定箇所	レベル	データ項目	値	期待値	判定結果
プログラム終了時(SUMSALES/SUMSALES)	01	ERRORFLAG	"1"	1	OK

テストデータ設定箇所	レベル	データ項目	値	期待値	判定結果
プログラム終了時(SUMSALES/SUMSALES)	01	ERRORFLAG	"1"	1	OK

2-21. カバレッジの結果確認

Progress

準備

テストケース1作成

テストケース2作成

実行

結果確認

最後にカバレッジの結果を確認します。
画面中央の「ソーステキスト」の左側にカバレッジ情報が表示されています。実行された文は色がつきます。

今回のテスト実行の結果、通常の入出力処理および入出力エラー処理の全ての処理がテスト実行されたことがわかります。

また、「テストプロジェクト」ツリーのテスト実行結果からカバレッジレポートをダブルクリックすると、「出力」にカバレッジレポートが表示されます。
こちらからでもC0/C1メジャーが共に100%となっていることが確認できます。

ファイルシミュレーション機能により、入出力エラーの処理も簡単にテストすることができ、カバレッジ網羅性を上げるのにも効果的です。

これで詳細結果も一通り確認できました。
この演習もここで完了です！

テスト完了!

正常ケース

ソーステキスト

```
000100000100* COBOL活用ガイド#002
000200000200*   ファイルシミュレーション機能の活用 サンプルコード
000300000300*
000400000400*
000500000500*
000600000600*
000700000700*
000800000800*
000900000900*
001000001000*
001100001100*
001200001200*
001300001300*
001400001400*
001500001500*
001600001600*
001700001700*
001800001800*
001900001900*
002000002000*
002100002100*
002200002200*
002300002300*
002400002400*
002500002500*
002600002600*
002700002700*
002800002800*
002900002900*
003000003000*
003100003100*
003200003200*
003300003300*
003400003400*
003500003500*
003600003600*
003700003700*
003800003800*
003900003900*
004000004000*
004100004100*
004200004200*
004300004300*
004400004400*
004500004500*
004600004600*
004700004700*
004800004800*
004900004900*
005000005000*
005100005100*
005200005200*
005300005300*
005400005400*
005500005500*
005600005600*
005700005700*
005800005800*
005900005900*
```

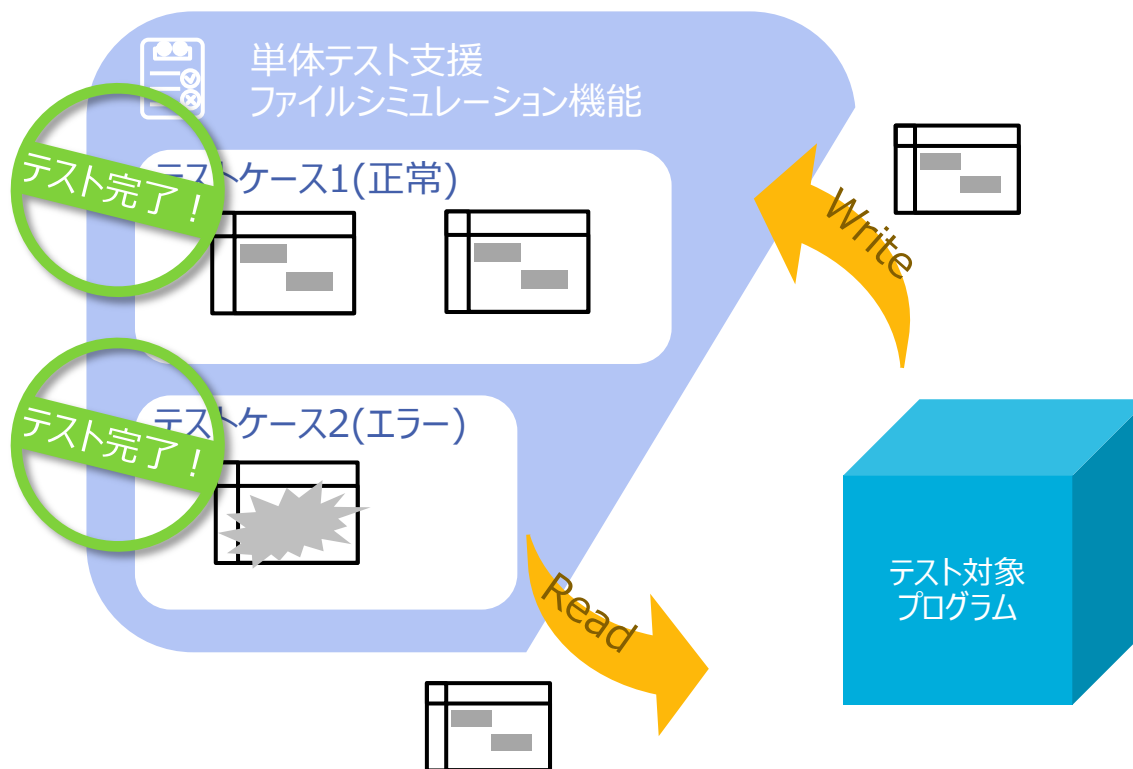
カバレッジレポート

コンパイル情報: COBOL2002(64) 03-04
ソースファイル名: sumsales.cbl

種類	カバレッジ
C0メジャー	100.0%
C1メジャー	100.0%
S1メジャー	---

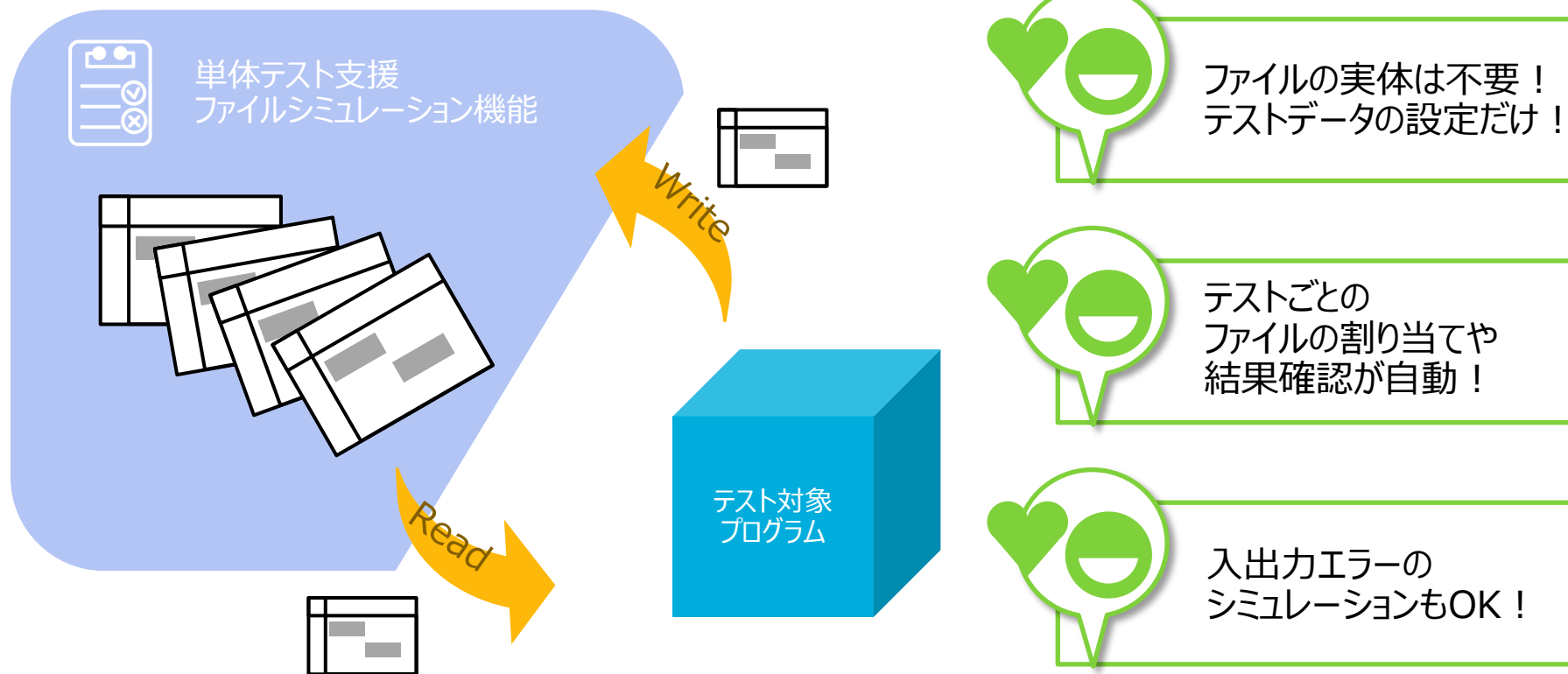
3.まとめ





ファイルシミュレーション機能を利用した単体テストについて演習を行いました。演習では、基本的な入力と出力のテストデータの設定方法、および入出力エラーの発生のシミュレーションの設定方法や実行結果の確認について手順を紹介しました。

単体テスト支援のファイルシミュレーション機能を活用することで、ファイル入出力を行うプログラムの単体テストの手間を削減でき、開発効率の向上に役立ちます。



ファイルシミュレーション機能以外にも、単体テスト支援は単体テストに有用な機能を取り揃えています。それらについてはCOBOL活用ガイドなどで紹介していますので、ご参照ください。

- 他の活用ガイドの紹介

COBOL2002活用ガイドシリーズは随時拡充/更新する予定です。
詳細はCOBOL2002のWebサイトまたは問い合わせください。

- お問い合わせ先

URL:<http://www.hitachi.co.jp/soft/cobol/>

sumsales.cbl

```

000100* COBOL活用ガイド#002
000200* ファイルシミュレーション機能の活用 サンプルコード
000300*
000400* All Rights Reserved. Copyright (C) 2016, Hitachi, Ltd.
000500 IDENTIFICATION DIVISION.
000600 PROGRAM-ID. SUMSALES.
000700 ENVIRONMENT DIVISION.
000800 INPUT-OUTPUT SECTION.
000900 FILE-CONTROL.
001000 SELECT IN-FILE ASSIGN TO FILE001.
001100 SELECT OUT-FILE ASSIGN TO FILE002.
001200 DATA DIVISION.
001300 FILE SECTION.
001400 FD IN-FILE.
001500 01 INDATA.
001600 05 SHOPNAME PIC X(10).
001700 05 SALES PIC S9(5) USAGE PACKED-DECIMAL.
001800 FD OUT-FILE.
001900 01 OUTDATA.
002000 05 TOTALSALES PIC S9(9) USAGE BINARY.
002100 WORKING-STORAGE SECTION.
002200 01 LOOPFLAG PIC 9.
002300 01 SUMSALES PIC 9(9).
002400 LINKAGE SECTION.
002500 01 ERRORFLAG PIC 9.
002600
002700 PROCEDURE DIVISION RETURNING ERRORFLAG.
002800
002900*ファイル入出力エラー処理
003000 DECLARATIVES.
003100 IO-ERR SECTION.
003200 USE AFTER STANDARD ERROR PROCEDURE IN-FILE.
003300 MOVE 1 TO ERRORFLAG.
003400 EXIT PROGRAM.
003500 END DECLARATIVES.
003600

003700*初期化処理
003800 MOVE 0 TO ERRORFLAG.
003900 MOVE 0 TO SUMSALES.
004000 MOVE 0 TO LOOPFLAG.
004100
004200*店舗データの読み込みと集計
004300 OPEN INPUT IN-FILE.
004400 PERFORM UNTIL LOOPFLAG = 1
004500 READ IN-FILE
004600 AT END MOVE 1 TO LOOPFLAG
004700 NOT AT END COMPUTE SUMSALES = SUMSALES + SALES
004800 END-READ
004900 END-PERFORM.
005000 CLOSE IN-FILE.
005100
005200*集計結果の出力
005300 OPEN OUTPUT OUT-FILE.
005400 MOVE SUMSALES TO TOTALSALES.
005500 WRITE OUTDATA.
005600 CLOSE OUT-FILE.
005700
005800 EXIT PROGRAM.
005900 END PROGRAM SUMSALES.

```

■ 商標類

HITACHIは株式会社 日立製作所の商標または登録商標です。

その他記載の会社名，製品名などは，それぞれの会社の商標もしくは登録商標です。