HITACHI
Inspire the Next

COBOL2002 Professional Tool Kit 単体テスト支援ガイド

手引・操作書

3021-3-618-10

COBOL2002

前書き

■ 対象製品

P-2636-7344 COBOL2002 Developer Professional 04-10（適用 OS：Windows 7，Windows 8.1，Windows 10，Windows Server 2008 R2，Windows Server 2012，Windows Server 2012 R2，Windows Server 2016，Windows Server 2019）

P-2936-7344 COBOL2002 Developer Professional(64) 04-10（適用 OS：Windows 7(x64)，Windows 8.1(x64)，Windows 10(x64)，Windows Server 2008 R2，Windows Server 2012，Windows Server 2012 R2，Windows Server 2016，Windows Server 2019）

■ 輸出時の注意

本製品を輸出される場合には、外国為替及び外国貿易法の規制並びに米国輸出管理規則など外国の輸出関連法規をご確認の上、必要な手続きをお取りください。

なお、不明な場合は、弊社担当営業にお問い合わせください。

■ 商標類

HITACHI，HiRDB，XDM は，株式会社 日立製作所の商標または登録商標です。

Microsoft は，米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows は，米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows Server は，米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

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。

■ 発行

2019 年 4 月　3021-3-618-10

■ 著作権

All Rights Reserved. Copyright (C) 2017, 2019, Hitachi, Ltd.

変更内容

変更内容（3021-3-618-10）　COBOL2002 Developer Professional 04-10，COBOL2002 Developer Professional(64) 04-10

追加・変更内容	変更個所
適用 OS に「Windows Server 2019」を追加した。	－

単なる誤字・脱字などはお断りなく訂正しました。

はじめに

このマニュアルは，次に示す COBOL2002 Professional 製品の機能である，単体テスト支援の使い方について説明したものです。

- COBOL2002 Developer Professional
- COBOL2002 Developer Professional(64)

このマニュアルでは，上記の製品を総称して，COBOL2002 Developer Professional と表記します。

■ 対象読者

単体テスト支援を利用して，開発中の COBOL プログラムの単体テストを実行するプログラマーで，次に示す項目を理解，習得していることを前提とします。

- Windows の基本的な操作方法
- COBOL2002 の知識

■ このマニュアルで使用する記号

このマニュアルで使用する記号を次のように定義しています。

記号	意味
[　　]	メニュータイトル，メニュー項目，ボタン，キー，およびアイコンの名称を示します。 例：[ファイル] メニュー [OK] ボタン [Enter] キー など
[　　] － [　　]	前のメニューを選択し，続けて後ろの項目を選択することを示します。 例：[ファイル] － [開く] [ファイル] メニューから [開く] を選択することを示します。
[　　] ＋ [　　]	＋ の前のキーを押したまま，後ろのキーを押すことを示します。 例：[Ctrl] ＋ [C] キー [Ctrl] キーを押したまま [C] キーを押すことを示します。
△	半角の空白を入れること，またはスペースキーを指定された数だけ押すことを示します。
斜体	インストール先フォルダなど可変の個所を示します。
〔 〕	省略できることを示します。 例：cbluts -Test -Run 〔-Detail〕 -Detail は省略できることを示します。

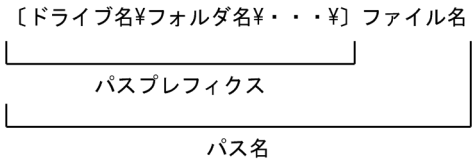
記号	意味
{ }	この記号で囲まれている複数の項目のうちから 1 つを選択することを示します。項目が縦に複数行にわたって記述されている場合は，そのうちの 1 行分を選択します。
|	横に並べられた複数の項目に対して項目間の区切りを示し，「または」を意味します。
下線	括弧で囲まれた複数の項目のうち 1 項目に対して使用され，括弧内のすべてを省略したときにシステムが仮定する標準値を意味します。

■ 用語の定義

このマニュアルで使用する用語の定義を次に示します。

パス名とパスプレフィクス

パス名とパスプレフィクスは，次のとおりです。



絶対パス名

ドライブ名で始まるパス名。

ドライブ名¥フォルダ名¥ … ¥ファイル名

相対パス名

カレントフォルダからの相対のパス名。

・カレントフォルダの 1 階層上位のフォルダを経由する場合

..¥フォルダ名¥フォルダ名¥ … ¥ファイル名

・カレントフォルダ下のフォルダを経由する場合

フォルダ名¥フォルダ名¥ … ¥ファイル名

絶対パスプレフィクス

ドライブ名で始まるパスプレフィクス。

相対パスプレフィクス

カレントフォルダからの相対のパスプレフィクス。

目次

前書き　2
変更内容　3
はじめに　4

1　単体テスト支援の紹介　11
1.1　単体テスト支援とは　12
1.2　単体テスト支援の利用方法　14
1.2.1　新規開発時のモジュールテストをする　14
1.2.2　プログラム修正後のデグレード有無を確認する　15
1.2.3　一括バッチを実行しデグレード有無を確認する　16

2　単体テスト支援の基本的な使い方　17
2.1　単体テスト支援を実行する　18
2.2　テスト結果を参照しプログラムを修正する　23

3　単体テスト支援の準備をする　24
3.1　テストできるプログラムかどうかを確認する　25

4　プログラムの単体テストをする　27
4.1　単体テスト支援の作業の手順　28
4.2　テストプロジェクトを作成する　29
4.2.1　開発マネージャからテストプロジェクトを作成する　30
4.2.2　Windows のプログラム一覧からテストプロジェクトを作成する　34
4.3　テストケースを作成する　37
4.3.1　テストケースを追加する　37
4.3.2　テストケースをコピーして追加する　37
4.3.3　テストケースの名前を変更する　38
4.3.4　テストケースを削除する　39
4.3.5　テストケースの有効と無効を切り替える　40
4.4　テストデータを設定する　41
4.4.1　プログラム開始時の値を設定する　41
4.4.2　プログラム終了時の値を設定する　43
4.4.3　ファイルシミュレーションのテストデータを設定する　44
4.4.4　中断点を設定する　49
4.4.5　中断点のテストデータを設定する　50

4.4.6 中断点を解除する 52
4.4.7 データ項目に値または期待値を設定する 52
4.4.8 値の入力規則 56
4.5 テストプロジェクトの環境を設定する 61
4.5.1 テストプロジェクトのプロパティを表示する 64
4.5.2 テストプロジェクトのスタブを編集する 66
4.5.3 テストケースごとに実行環境ファイルと実行時環境変数を設定する 67
4.6 単体テストを実行する 71
4.6.1 すべてのテストケースを実行する 72
4.6.2 1 つのテストケースを実行する 73
4.7 テストプロジェクトを保存する 75
4.8 単体テスト支援を終了する 76
4.9 テストプロジェクトを開く 77
4.9.1 単体テスト支援のメインウィンドウからテストプロジェクトを開く 77
4.9.2 開発マネージャからテストプロジェクトを開く 77
4.9.3 Windows のエクスプローラーから単体テスト支援を開始する 78
4.10 単体テストを実行したときの判定結果 79

5 テストを繰り返す 82
5.1 エラーになった COBOL プログラムを修正する 83
5.2 COBOL ソースファイルを解析する 84

6 テスト結果を参照する 89
6.1 テストプロジェクト結果レポートを参照する 90
6.2 テストケース結果レポートを参照する 91
6.3 カバレージレポートを参照する 96
6.4 エビデンスファイルを確認する 98

7 テスト結果を操作する 99
7.1 手動で判定結果を変更する 100
7.2 実行結果の名前を変更する 102
7.3 実行結果を削除する 103
7.4 実行結果をテストプロジェクトから除外する 104
7.5 カバレージ情報をクリアする 105

8 バッチモードで実行する 106
8.1 バッチモードでできる単体テスト支援の機能 107
8.2 テストケースの実行 108
8.2.1 すべてのテストケースを実行する 108
8.3 テストケースの操作 110

8.3.1 テストケースを削除する 110
8.3.2 テストケースを一覧表示する 111
8.3.3 CSV ファイルをインポートする（フォルダ指定） 111
8.3.4 CSV ファイルをインポートする（ファイル指定） 112
8.3.5 テストケースをエクスポートする（テストケース指定） 113
8.3.6 テストケースをエクスポートする（ひな形） 114
8.4 テストプロジェクトの操作 115
8.4.1 テストプロジェクトを作成する 115
8.4.2 プロジェクト情報ファイルを出力する 116
8.4.3 プロジェクト情報を設定する 117
8.5 実行結果の操作 124
8.5.1 カバレージ情報を表示する 124
8.5.2 カバレージ情報をクリアする 124
8.5.3 実行結果を削除する 125
8.5.4 実行結果を一覧表示する 126
8.6 バッチモードでできる単体テスト支援の使用例 127
8.6.1 大量の COBOL ソースファイルから一括でプロジェクトファイルを作成する 127
8.6.2 新しい環境でプログラムの動作確認をやり直す 128
8.6.3 プロジェクト情報を変更する 129
8.6.4 リグレッションテストなどのために繰り返しテスト実行する 130
8.6.5 ほかのテストプロジェクトにあるテストケースを流用する 132
8.6.6 サブフォルダにある CSV ファイルを一括でインポートする 134
8.6.7 複数のテストケースを一括でエクスポートする 135
8.6.8 複数のテストケースまたは実行結果を一括で削除する 136
8.6.9 テストケースごとに実行環境ファイルまたは実行時環境変数を設定する 138

9 環境をカスタマイズする 140
9.1 テストプロジェクトを移動する 141
9.2 テストケースを CSV ファイルにエクスポートしインポートする 142
9.2.1 テストケースをエクスポートする（すべてのテストケースを CSV ファイルにエクスポートする場合） 142
9.2.2 テストケースをエクスポートする（選択したテストケースを CSV ファイルにエクスポートする場合） 144
9.2.3 テストケースをエクスポートする（ひな形のテストケースをエクスポートする場合） 146
9.2.4 テストケースをインポートする（指定したフォルダの直下にあるすべての CSV ファイルをテストケースとしてインポートする場合） 147
9.2.5 テストケースをインポートする（CSV ファイルをテストケースとしてインポートする場合） 149
9.3 単体テスト支援の画面を変更する 152
9.3.1 色を変更する 152
9.3.2 フォントを変更する 153

9.4 データ項目検索の設定を変更する 154
9.5 テスト実行の設定を変更する 155

10 画面のリファレンス 156
10.1 ［新しいテストプロジェクトを作る］ダイアログ 157
10.2 単体テスト支援のメインウィンドウ 160
10.3 ［テストプロジェクト］画面 162
10.4 ［テスト設定］画面 164
10.5 ［スタートページ］ 165
10.6 ［プロパティページ］ 166
10.7 ［スタブ設定ページ］ 168
10.8 ［テストケースページ］ 170
10.8.1 ［ソーステキスト］画面 170
10.8.2 ［プログラム開始時］画面 172
10.8.3 ［プログラム終了時］画面 175
10.8.4 ［ファイルシミュレーション］画面 176
10.8.5 ［中断点］画面 179
10.8.6 ［テスト環境の設定］画面 182
10.9 ［出力］画面 184
10.9.1 ［メッセージ］画面 184
10.9.2 ［テストプロジェクト結果レポート］画面 185
10.9.3 ［テストケース結果レポート］画面 186
10.9.4 ［カバレージレポート］画面 192

11 ファイルのリファレンス 194
11.1 テストプロジェクトのファイル構成 195
11.2 プロジェクト情報ファイルの構成要素 197

12 コマンドリファレンス 200
12.1 コマンドの指定形式 201
12.2 バッチモードで単体テスト支援を実行するコマンド 204
cbluts コマンド － バッチモードで単体テスト支援を実行する 204

13 エラー時の対処 222
13.1 メッセージから探す 223
13.1.1 メッセージの出力形式 223
13.1.2 メッセージの一覧 223
13.2 エラー時に採取するファイル 243
13.2.1 エラー情報ファイルの格納先 243

付録　244

付録 A　単体テスト支援で使用するファイル　245
付録 A.1　単体テスト支援のファイル一覧　245
付録 A.2　テストケースの CSV ファイル　246
付録 B　単体テスト支援で使用できる正規表現　259
付録 C　単体テスト支援の制限値と限界値　260
付録 D　単体テスト支援の環境変数　262
付録 E　各バージョンの変更内容　264
付録 F　このマニュアルの参考情報　265
付録 F.1　関連マニュアル　265
付録 F.2　このマニュアルでの表記　265
付録 F.3　KB（キロバイト）などの単位表記について　266

索引　267

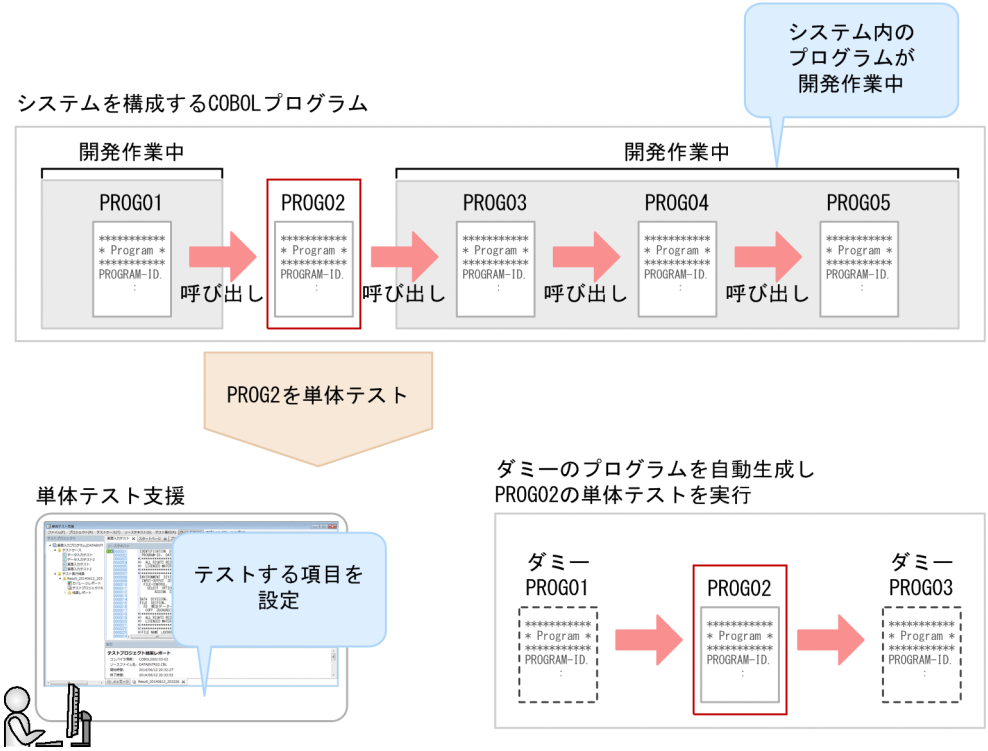
1 単体テスト支援の紹介

単体テスト支援の概要について説明します。

1.1　単体テスト支援とは

単体テスト支援とは，開発中の COBOL プログラムの単体テストを GUI の操作で実行できるようにする機能です。

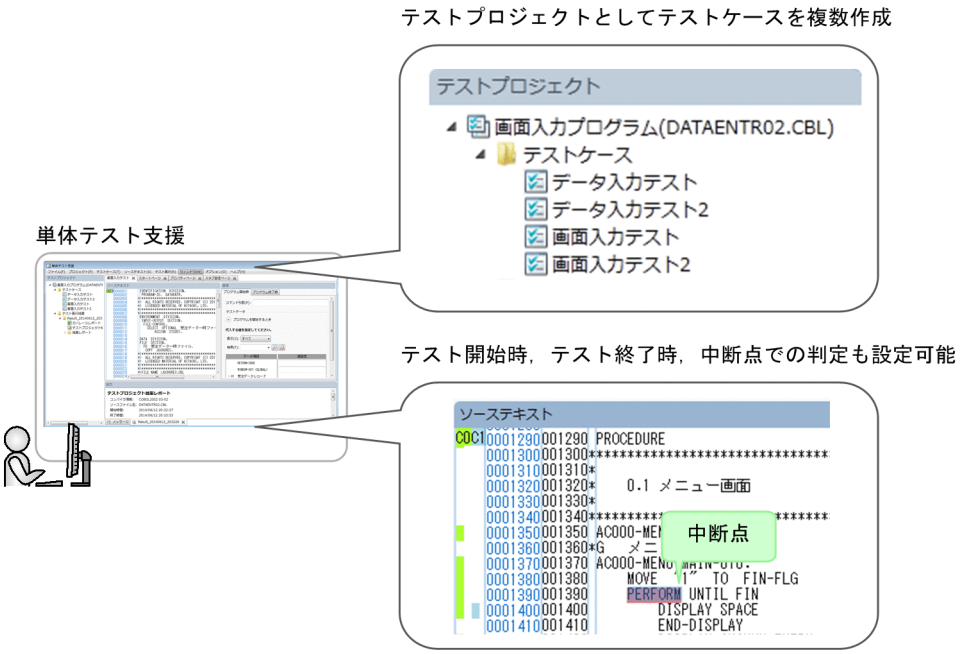
単体テスト支援を使えば，ダミーの主プログラム，副プログラムが自動的に作成されるため，連動するプログラムやファイルなどの環境が整わない段階でもプログラムをテストできます。さらに，ファイルへの入出力処理に代わる操作（シミュレーション）ができます。



テスト項目は，GUI でテストケースを作成し，複数のテストケースをテストプロジェクトとして管理できます。テスト項目を GUI で確認できるため，コマンド文字列の入力に比べるとテスト項目の設定やテスト結果の管理がしやすくなります。これによって，プログラムのデバッグ，カバレージ実行の手間を軽減できます。

単体テスト支援の画面で設定したテストケースの情報を CSV ファイルとしてエクスポートおよびインポートできます。多数のテストデータの設定を外部ツールで効率良く作成/編集するときに使用できます。

テストの実行では，必要な処理だけをテストできるように，プログラムの開始時と中断点で設定する値を指定します。また，中断点およびプログラム終了時でデータ項目の値が正しいかどうかを判定できます。

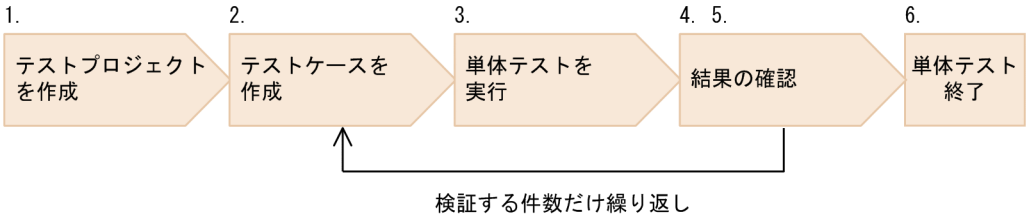

テストプロジェクトとしてテストケースを複数作成
テストプロジェクト
画面入力プログラム(DATAENTR02.CBL)
テストケース
データ入力テスト
データ入力テスト2
画面入力テスト
画面入力テスト2
単体テスト支援
テスト開始時，テスト終了時，中断点での判定も設定可能
ソーステキスト
中断点

1.2　単体テスト支援の利用方法

単体テスト支援の基本的な利用方法です。

1.2.1　新規開発時のモジュールテストをする

新規開発中のシステムに含まれる COBOL プログラム（.cbl）のコーディングが終了したので，プログラムの単体テストを実行する手順です。



作業の手順

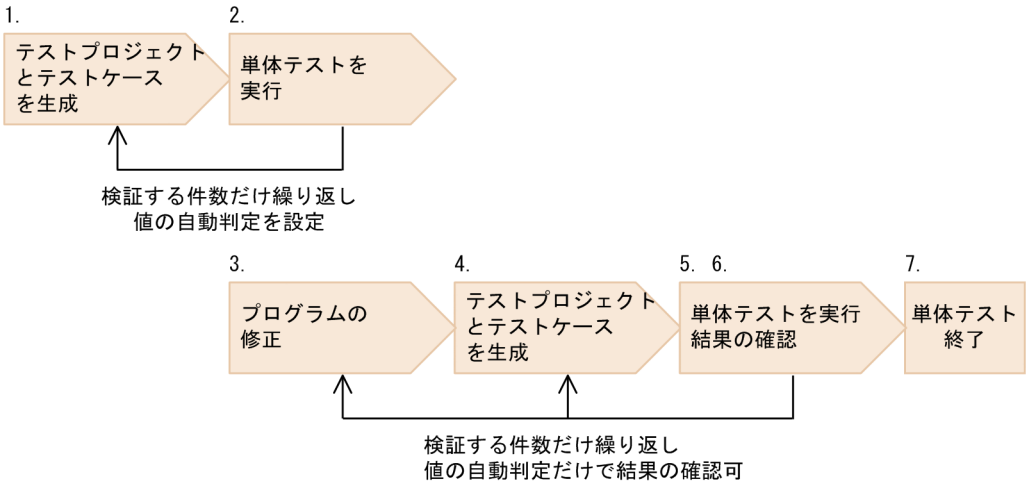
1. 単体テスト支援を起動し，COBOL プログラム（.cbl）に対応するテストプロジェクトを作成します。
2. テストプロジェクトに新規にテストケースを追加し，そのケースにテストデータを入力します。
   検証するテストケースを複数作成できます。
3. 単体テストを実行します。
4. 単体テストの結果レポートを参照して，COBOL プログラム（.cbl）の動作が正しいかどうかを判断します。
   正しくない場合は，COBOL プログラムを修正またはテストデータを修正し，再度テストします。
5. カバレージレポートを参照して，COBOL プログラム（.cbl）のカバレージ率を確認します。
   実行していない文があるときは，ソーステキスト画面で確認し，実行していない文が実行されるテストケースを追加し，再度テストします。
6. すべてのテストを実施したことを確認します。
   COBOL プログラム（.cbl）に対する単体テストは完了します。

テストプロジェクトを作成し，検証が必要な件数だけテストケースを追加し，動作に必要な個所だけにテストデータを設定して，単体テストを実行できます。

テスト対象プログラムを呼び出す主プログラムと，CALL 文でテスト対象プログラムから呼ばれる副プログラムは自動生成されるため，テストデータだけを設定すれば，単体テストができます。また，テスト結果は単体テスト支援が生成するレポートを GUI 上で参照できるため，レポートの作成と管理の手間を減らせます。

1.2.2　プログラム修正後のデグレード有無を確認する

システムのエンハンスで，既存の COBOL プログラム（.cbl）を修正することになったため，修正部分の検証と既存部分のデグレードを単体テストで確認する手順です。



作業の手順

1. 修正前の COBOL プログラム（.cbl）に対応するテストプロジェクトを作成し，必要なテストケースとテストデータを設定します。

2. 修正後のテストで結果を自動判定できるように期待値を設定し，単体テストを実行します。

   テストを繰り返して実行するときに，値を自動判定するように設定します。

   期待値が不明な場合は，単体テストの実行後に結果の値を設定します。

3. COBOL プログラム（.cbl）を修正します。

4. 単体テスト支援で COBOL プログラム（.cbl）用のテストプロジェクトを表示し，ソースコードの再読み込み（COBOL ソースファイルの解析）を実行して，プログラムの修正をテストプロジェクトへ取り込みます。

   COBOL プログラム（.cbl）の修正で変更されたデータ項目があり，テストデータの変更が必要な場合は，その部分を修正します。

5. 単体テストを実行し，結果レポートを参照して，COBOL プログラム（.cbl）の動作が正しいかどうかを判断します。

   結果の自動判定を設定しているため，判定結果は OK か NG かで判断できます。

   正しくない場合は，ソースの修正またはテストデータを修正し，再度テスト実行します。

6. カバレージレポートを参照して，COBOL プログラム（.cbl）のカバレージ率を確認します。

   実行していない文があるときは，ソーステキスト画面で確認し，実行していない文が実行されるテストケースを追加し，再度テストします。

7. すべてのテストを実施したことを確認します。

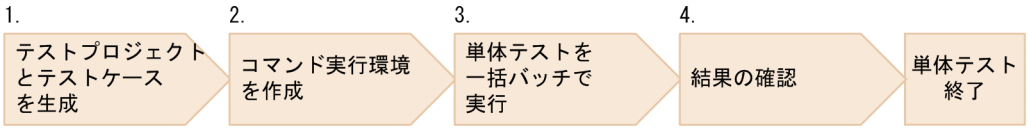
COBOL プログラム（.cbl）に対する修正部分の検証およびデグレードの確認が完了です。

一度テストプロジェクトを作成し，テストデータを設定すると，そのテストデータを繰り返し使用できるので，テスト実行の手間を低減できます。特に結果の自動判定を設定すると，結果が OK か NG かだけを確認するだけです。

COBOL ソースファイルを修正した場合も，テストプロジェクトに再読み込み（COBOL ソースファイルの解析）で反映させると，ソース修正部分以外のテストデータはそのまま利用できるため，エンハンス時にも単体テストの手間を大幅に低減できます。

1.2.3　一括バッチを実行しデグレード有無を確認する

システム開発中は， COBOL プログラムは頻繁に修正されます。各 COBOL プログラムの単体テストを定期的（例：夜間）に一括でバッチ実行すれば，修正件数が大幅に増えた場合でも，業務時間外に単体テストをまとめて実行できます。



作業の手順

1. 各 COBOL プログラムに対してテストプロジェクトを作成します。
2. bat ファイルなどのスクリプトファイルを用意し，作成済みのテストプロジェクトを実行するコマンドを記述します。
3. 手順 2.のスクリプトファイルを定期的に実行します。
4. 実行後に結果を参照し，デグレードしていないかを確認します。
   問題がなければ，デグレードの確認は完了です。

テストプロジェクトが作成済みであれば，テストの実行はコマンドラインで実行できます。このため，複数のテストプロジェクトをまとめてバッチで実行することで，個々に単体テストを実行する手間が省けます。結果の確認も標準出力に各テストプロジェクトのテスト結果のサマリ情報が出力されるため，GUI を起動しなくてもテスト結果を参照できます。

2 単体テスト支援の基本的な使い方

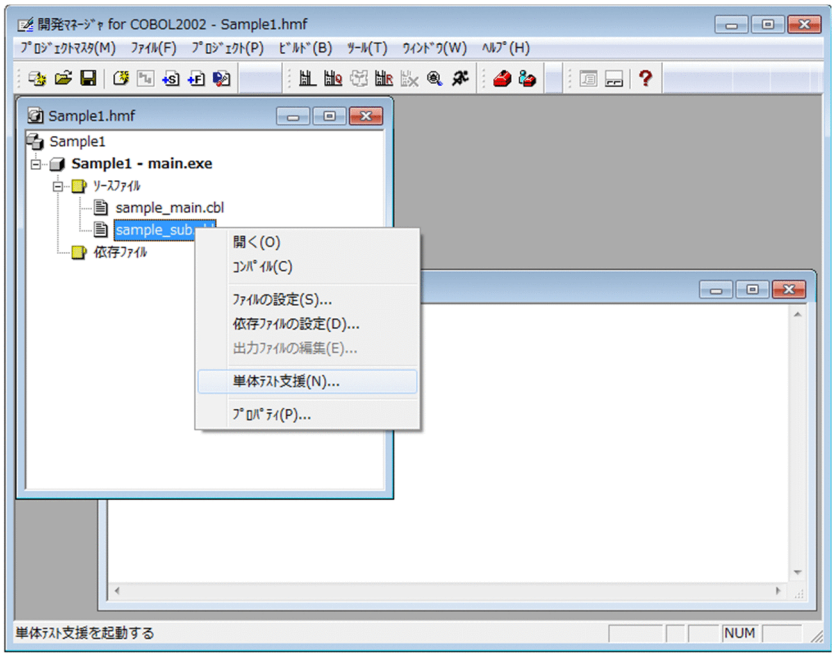
単体テスト支援の基本的な使い方について説明します。各手順の詳細については，3 章以降を参照してください。

2.1　単体テスト支援を実行する

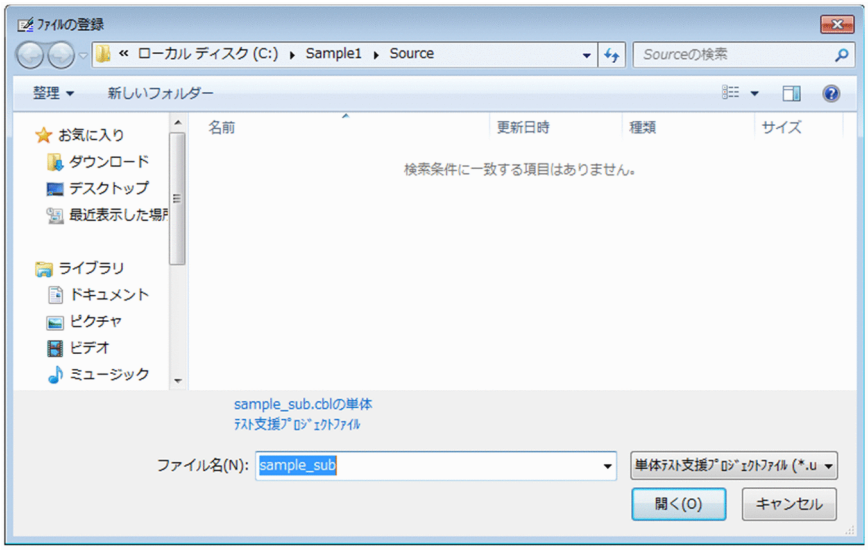
単体テスト支援を実行する，基本的な手順です。

作業の手順

1. 開発マネージャのツリービューウィンドウで，単体テストをする COBOL ソースファイルを右クリックし，ポップアップメニューから［単体テスト支援］メニューを選択します。



テストプロジェクトファイルの格納先フォルダを指定するダイアログが表示されます。格納先は，COBOL ソースファイルが格納されているフォルダが仮定されます。



1 つの COBOL ソースファイルに対して，複数のテストプロジェクトファイルを作成できます。複数のテストプロジェクトファイルを作成するときは，事前にワークフォルダを作成しておくことをお勧めします。

テストプロジェクトファイルを指定するときは，ほかのテストプロジェクトファイルが存在しないフォルダのファイルを指定してください。すでにあるテストプロジェクトファイルは指定できません。

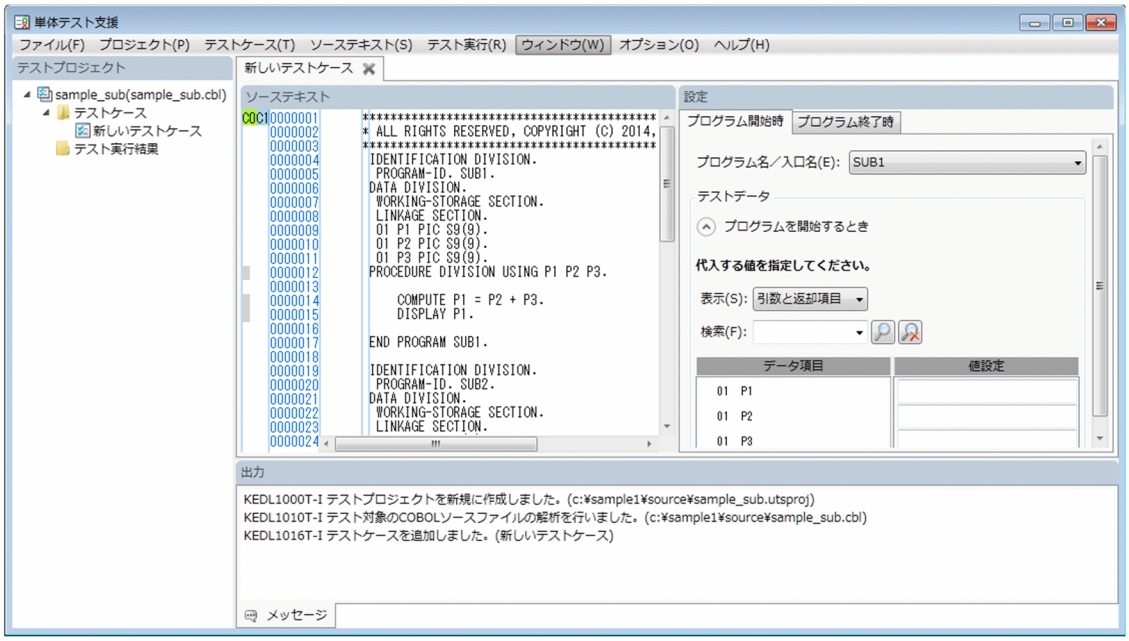
指定したフォルダにほかのテストプロジェクトファイルが存在する場合，またはすでにあるテストプロジェクトファイルを指定した場合は，エラーになります。この場合，COBOL ソースファイルの選択からやり直してください。

ここで指定したフォルダが，テストプロジェクト作業フォルダになります。テストプロジェクト作業フォルダは，単体テストの実行時のカレントフォルダになります。

2. テストプロジェクトファイル名を指定して［開く］ボタンをクリックします。

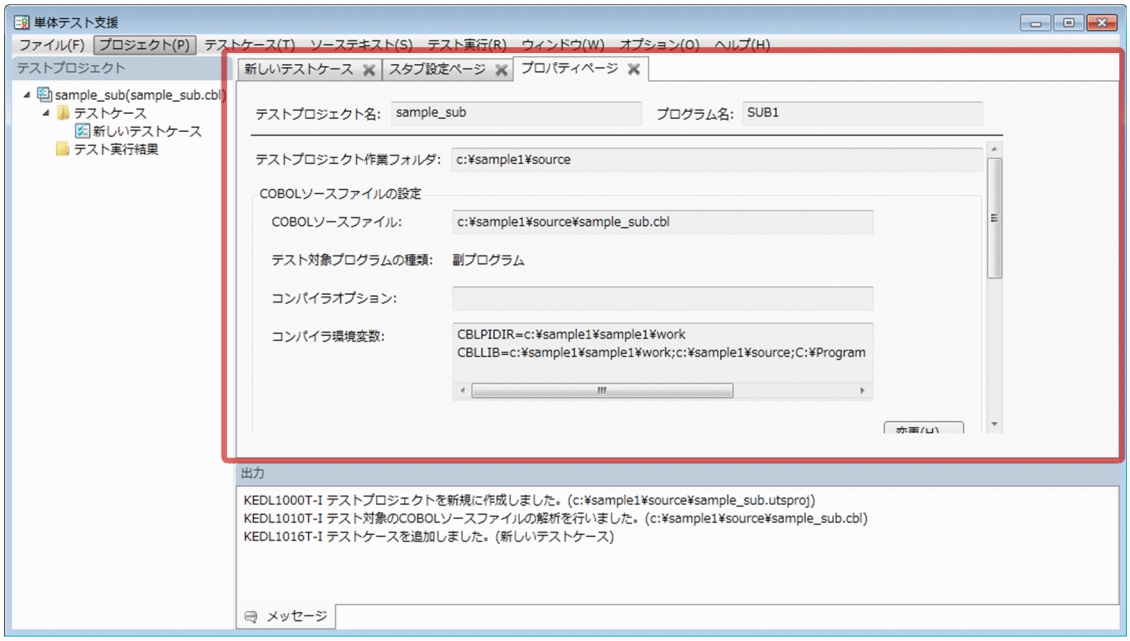
単体テスト支援のメインウィンドウが表示されます。

指定した COBOL ソースファイルに複数のプログラム定義がある場合は，プログラム名を選択するダイアログが表示されます。



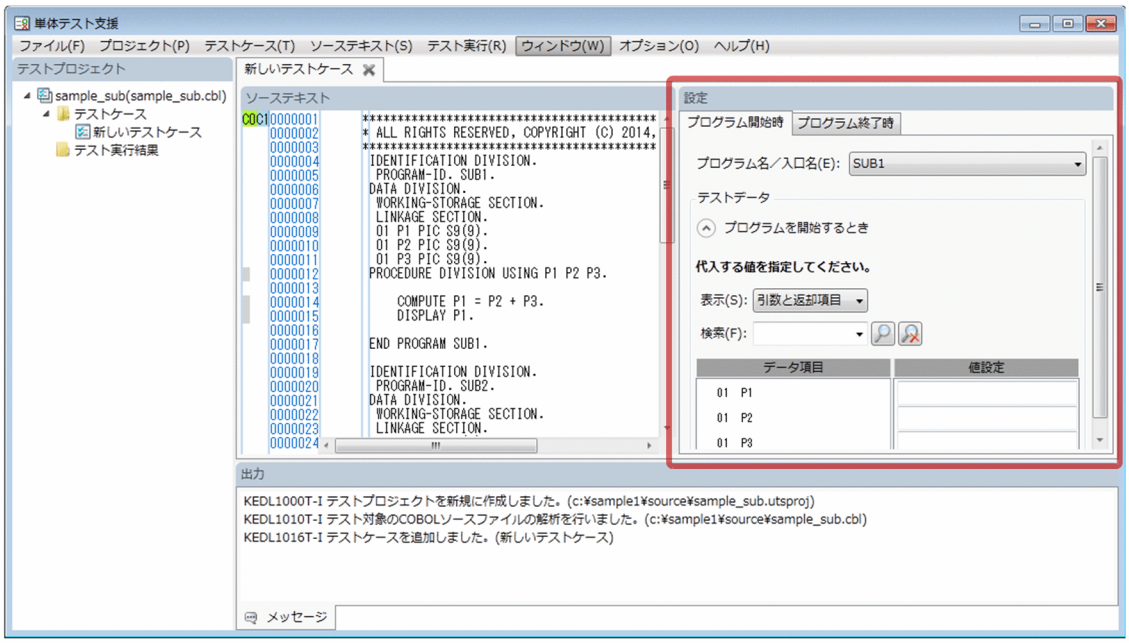
3. 必要に応じて，［プロパティページ］と［スタブ設定ページ］の値を見直してください。

［スタブ設定ページ］を表示するときはメインウィンドウの［プロジェクト］－［スタブ設定ページを開く］を選択，［プロパティページ］を表示するときはメインウィンドウの［プロジェクト］－［プロパティページを開く］を選択します。

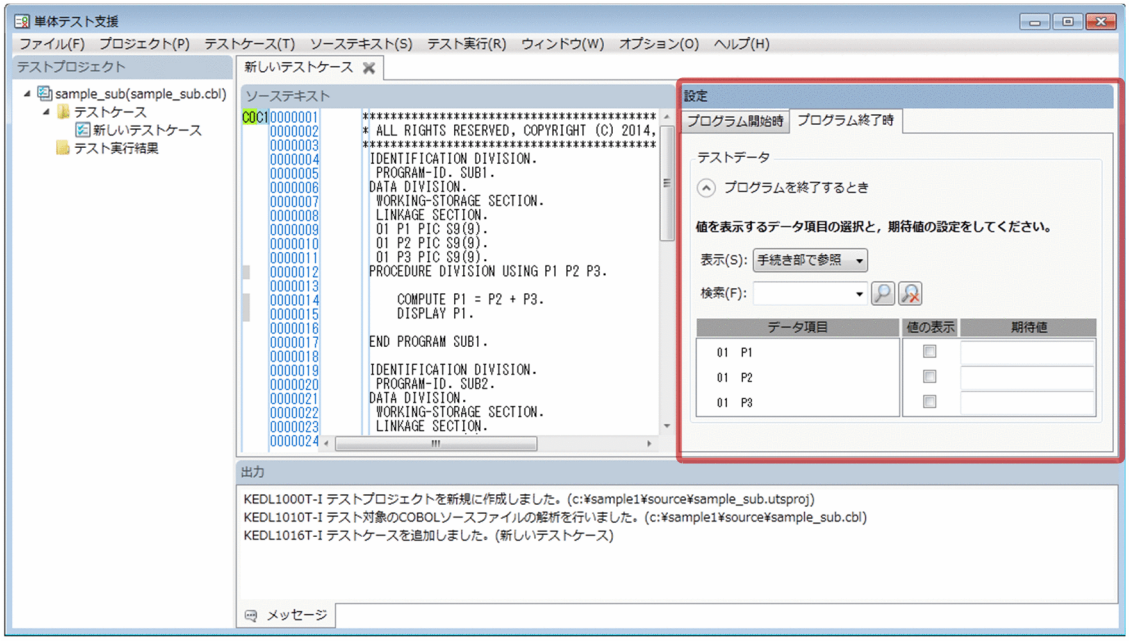


［プロパティページ］と［スタブ設定ページ］に設定する値の注意については，「4.5　テストプロジェクトの環境を設定する」の注意を参照してください。

4. メインウィンドウ右側の［プログラム開始時］画面に，テストするデータ項目に設定する値を入力します。

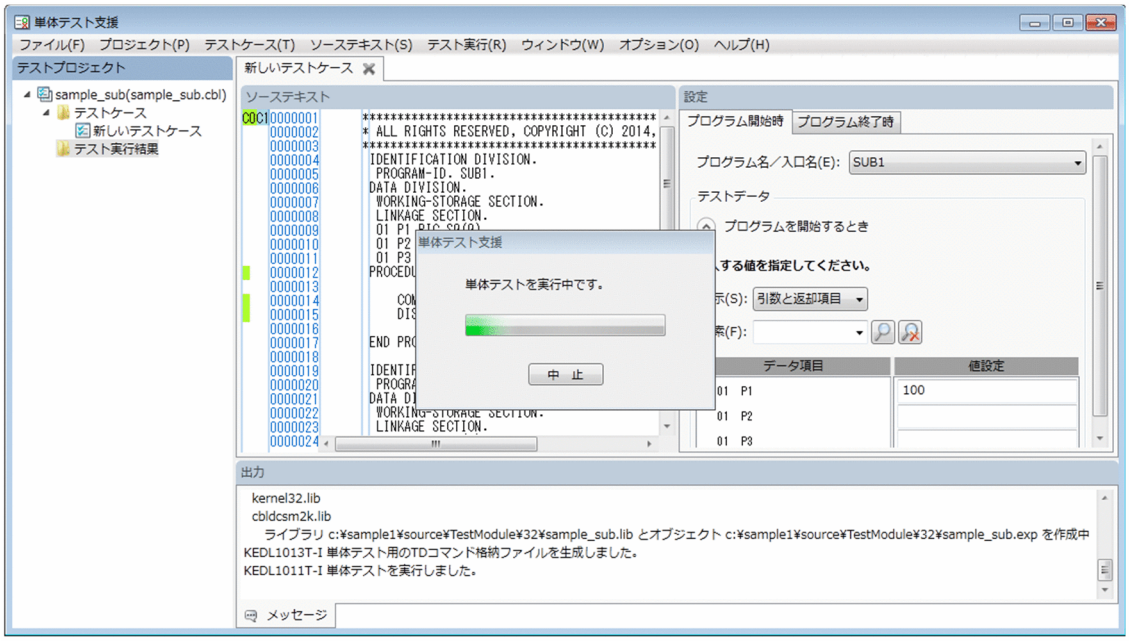


5. メインウィンドウ右側の［プログラム終了時］画面に，テスト結果を判定する値をデータ項目に入力します。

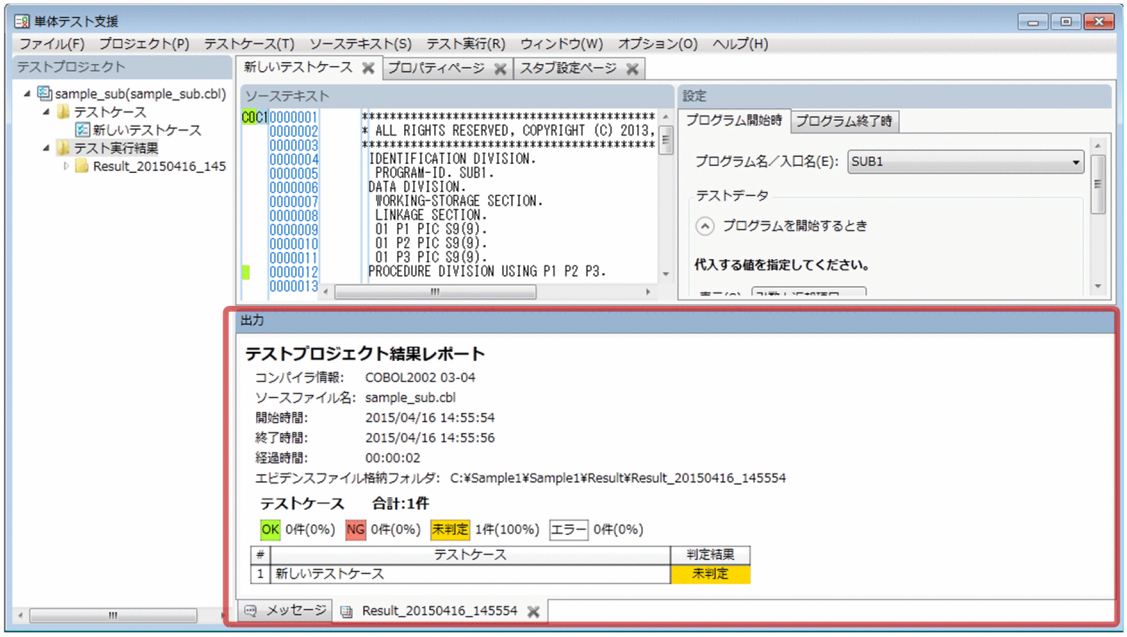


6. ［テスト実行］ － ［テストケースを実行］ メニューを選択します。

単体テストが実行されます。



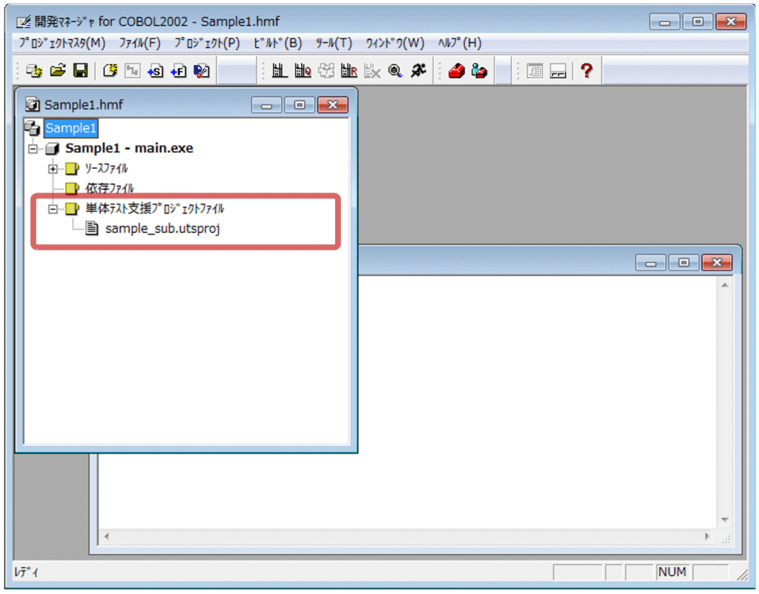
実行が完了すると，メインウィンドウ左側の［テストプロジェクト］画面のテスト実行結果の下に，テストを実行した日付のフォルダ「Result_*yyyymmdd_hhmmss*」（*yyyy*：西暦年，*mm*：月，*dd*：日，*hh*：時，*mm*：分，*ss*：秒）が作成され，メインウィンドウの下部にテストプロジェクト結果レポートが表示されます。



テストケースはテストプロジェクト内に複数作成できるので，テストケースごとに，異なるテスト内容を設定できます。

テストケースを実行すると，テスト結果はテストプロジェクトに自動的に保存されます。

テストプロジェクトを保存すると，開発マネージャのツリービューウィンドウには単体テスト支援プロジェクトファイルノードが追加され，テストプロジェクトファイルが追加されます。保存したテストプロジェクトファイルは，開発マネージャからも開けます。



2.2　テスト結果を参照しプログラムを修正する

単体テスト支援でのテスト結果を参照し，プログラムを修正する手順です。

作業の手順

1. メインウィンドウ左側の［テストプロジェクト］画面の［テスト実行結果］タグから，参照するテスト結果を選択します。

2. 次の目的別に，テスト結果レポートを参照します。
   - テストプロジェクト内のすべてのテストケースの結果を参照するとき
     テストプロジェクト結果レポートを開きます。
   - 個々のテストケースの結果を参照するとき
     ［結果レポート］タグの下にある，テストケース名で示すテストケース結果レポートを開きます。
   - プログラムのカバレージ率を参照するとき
     カバレージレポートを開きます。

3. プログラムに問題があった場合は，参照した内容を基に，COBOL エディタでプログラムを修正します。

4. COBOL プログラムの修正後，再度単体テストをする場合は，［プロジェクト］－［COBOL ソースファイルの解析］メニューを選択します。
   修正後の COBOL ソースファイルの内容がテストケースに反映されます。

3 単体テスト支援の準備をする

単体テスト支援を使う前の準備作業について説明します。

3.1　テストできるプログラムかどうかを確認する

単体テストの対象とする COBOL プログラムがテストできるかどうか確認する手順です。

単体テスト支援が対象とする COBOL プログラムは，プログラム定義の翻訳単位（内側のプログラム定義も含む）です。この翻訳単位をコンパイルして生成したプログラムのことをテスト対象のプログラムといいます。

単体テスト支援を使用するときの注意

- 単体テスト支援では，1 つのテストプロジェクトで 1 つの COBOL ソースファイルをテスト対象にできます。
- クラス定義，インタフェース定義，および関数定義の翻訳単位は，単体テスト支援での単体テストはできません。ただし，単体テストの対象とするプログラム定義の翻訳単位から，上記または別のプログラム定義の翻訳単位を呼び出すことはできます。
- 1 つの COBOL ソースファイルに複数の最外側のプログラムがあるときは，最外側のプログラムごとにテストプロジェクトを作成する必要があります。
- プログラム定義の翻訳単位が 1 つも含まれない COBOL ソースファイルは，単体テスト支援の対象にできません。
- 単体テストでは，ファイルへの入出力処理に代わる操作（シミュレーション）ができます。
- 次に示す呼び出しやアクセスでは，呼び出すプログラム定義やアクセスするファイルがなくても，対象となるプログラム定義の単体テストができます。
  - 副プログラム呼び出し（CALL 定数，CALL 一意名）

    単体テスト支援の指定で，呼び出す副プログラム定義を仮定します。副プログラムの手続きはシミュレーションできません。

    ただし，stdcall 呼び出し規約で呼び出すプログラムの場合は，呼び出す副プログラム定義は仮定されません。そのため，呼び出し先を準備してリンクするか，次に示す方法で cdecl 呼び出し規約に沿った単体テストをする必要があります。

    ・-StdCallFile コンパイラオプションで stdcall 呼び出し指示ファイルを指定している場合は，コンパイラオプションを削除する。

    ・COBOL プログラムで，環境部の外部プログラム節に指定している場合はコメントアウトする。
  - ファイルアクセス

    単体テスト支援の指定で，アクセスするファイルを仮定します。
- 次に示す呼び出しやアクセスは，単体テストの実行時にシミュレーションできません。単体テストの実行時には実呼び出しや実アクセスが必要です。
  - INVOKE 文またはオブジェクトプロパティの参照によるメソッド呼び起こし
  - サービスルーチン呼び出し
  - 利用者定義関数呼び出し

- 共通例外処理の例外引き起こし
- ODBC インタフェース機能によるデータベースアクセス
- HiRDB による索引ファイル入出力機能
- XDM によるデータベース操作
- OLE2 オートメーション機能
- データコミュニケーション機能によるメッセージ受け渡し

- 画面入出力があるプログラムは，単体テストの実行時に画面の操作が必要です。
- 埋め込み SQL 文によるデータベースアクセスをするプログラムでは，単体テスト支援の入力はプリプロセス後の COBOL ソースファイルである必要があります。
- 単体テスト支援で単体テストを実行する場合は，常にシングルスレッドで動作します。

COBOL ソースファイルの内容と単体テスト支援上での表示が異なる項目

等価規則や COPY 文の展開後の表示などで，COBOL ソースファイルの内容と単体テスト支援上での表示が異なる項目があります。異なる項目は次のとおりです。

- プログラム名，入口名

  COBOL2002 のプログラム名の構成規則，変換規則に従って変換した名称で扱います。プログラム名の構成規則，変換規則については，マニュアル「COBOL2002 言語 標準仕様編」を参照してください。

- データ項目を表示するツリーでのデータ項目名

  等価規則や，-EquivRule コンパイラオプションの指定がある場合は，その規則の適用後の名前で表示されます。項目名が明記されていない場合は，名称には FILLER と表示されます。

  等価規則や-EquivRule コンパイラオプションについては，マニュアル「COBOL2002 言語 標準仕様編」または「COBOL2002 ユーザーズガイド」を参照してください。

  なお，-EquivRule,StdCode コンパイラオプションには対応していません。指定した場合は，-EquivRule,StdCode コンパイラオプションを指定してコンパイルされますが，単体テストで使用する TD コマンドは，-EquivRule コンパイラオプションを指定していないときのものと同じになります。

- ソーステキスト

  原文展開後など，COBOL コンパイラで翻訳したあとの文字列が表示されます。

- 行番号

  原文展開後などの，COBOL コンパイラが付与した番号が表示されます。これはコンパイルリストの行番号と一致します。

4 プログラムの単体テストをする

単体テスト支援を使って COBOL プログラムの単体テストをするときの各手順について説明します。

4.1　単体テスト支援の作業の手順

単体テスト支援を使用するときの基本的な手順です。

テスト対象の COBOL プログラムのコーディングが完了したあとで，単体テスト支援を使用します。

1. テストプロジェクトを作成する
2. テストケースを作成する
3. テストデータを設定する
4. テストプロジェクトの環境を設定する
5. 単体テストを実行する
6. テストプロジェクトを保存する
7. 単体テスト支援を終了する

4.2　テストプロジェクトを作成する

単体テスト支援のテストプロジェクトを作成する手順です。

テストプロジェクトを作成するには，次の 2 つの方法があります。

- 開発マネージャからテストプロジェクトを作成する
- Windows のプログラム一覧からテストプロジェクトを作成する

注意

- 単体テスト支援のテストプロジェクトを作成するときは，COBOL ソースファイルを解析します。テストプロジェクトのプロパティに指定したコンパイラオプションに加え，次に示すコンパイラオプションが自動的に指定されます。
  - -SrcList,OutputAll
  - -TDInf
  - -CVInf
  - -SimMain
  - -SimSub
  - -SimIdent
  - -OutputFile
  - -Main,{System¦V3}

  上記のコンパイラオプションが無効になるコンパイラオプションは，テストプロジェクトのプロパティのコンパイラオプションには指定しないでください。指定した場合の単体テスト支援の動作は保証しません。コンパイラオプションの組み合わせによって有効または無効になるコンパイラオプションについては，マニュアル「COBOL2002 ユーザーズガイド」を参照してください。

  注意が必要なコンパイラオプションを次に示します。これらのコンパイラオプションはテストプロジェクトのプロパティには指定しないでください。

コンパイラオプション	指定すると起こる現象
-Dll,{Stdcall ｜ Cdecl}［32bit 版の COBOL2002 の場合］	「単体テストの実行」でエラーになります。
-Dll［64bit 版の COBOL2002 の場合］	
-Compile,CheckOnly	「単体テストの実行」でエラーになります。
-LiteralExtend,Alnum	次のどれかを［プロパティページ］または［スタブ設定ページ］で設定している場合，-LiteralExtend,Alnum コンパイラオプションが無効化されます。 • 「テスト対象プログラムの種類」で副プログラムを選択する • ［スタブ設定ページ］の「CALL 定数で呼び出す副プログラム名」で 1 つ以上有効にする

コンパイラオプション	指定すると起こる現象
	• ［スタブ設定ページ］の「CALL 一意名で呼び出す副プログラム名」に 1 つ以上のプログラム名を指定する

- 次の個所で設定した環境変数の合計値が 65,535 バイトを超えると，COBOL ソースファイルの解析をしたときに論理エラーになります。回避するには，不要な環境変数を削除してください。
  - ［新しくテストプロジェクトを作る］ダイアログ，［プロパティページ］または［COBOL ソースファイルの設定］ダイアログで設定したコンパイラ環境変数
  - ［プロパティページ］で設定した実行時環境変数
  - 開発マネージャで設定したコンパイラ環境変数（開発マネージャから起動した場合）
  - システムに設定した環境変数
- OCCURS 指定ありデータ項目には，1 次元当たり最大 500 までテストデータを設定できます。500 を超える分はテスト対象にできません（表示，値の設定ができません）。OCCURS 指定ありデータ項目の 1 次元当たりに設定できるテストデータの個数の上限値を変更するときは，環境変数 CBLUTS_ANALYZEMAXOCCURSNUM に上限値を設定してください。環境変数については，「付録 D　単体テスト支援の環境変数」を参照してください。
- -StdCall コンパイラオプション（32bit 版 COBOL2002 でだけ有効）を指定しても COBOL ソースファイルと同じフォルダに置いた stdcall 呼び出し指示ファイルは有効になりません。-StdCallFile コンパイラオプションで stdcall 呼び出し指示ファイルを絶対パスで指定してください。指定しない場合は，cdecl 呼び出し規約のプログラムとしてスタブ生成の対象になります。
- ファイル名にコンマ（,）を含む COBOL ソースファイルをテスト対象にすることはできません。また，パスにコンマを含むフォルダをテストプロジェクト作業フォルダに指定することはできません。

  どちらの場合も，指定するとエラーになりテストプロジェクトは生成されません。

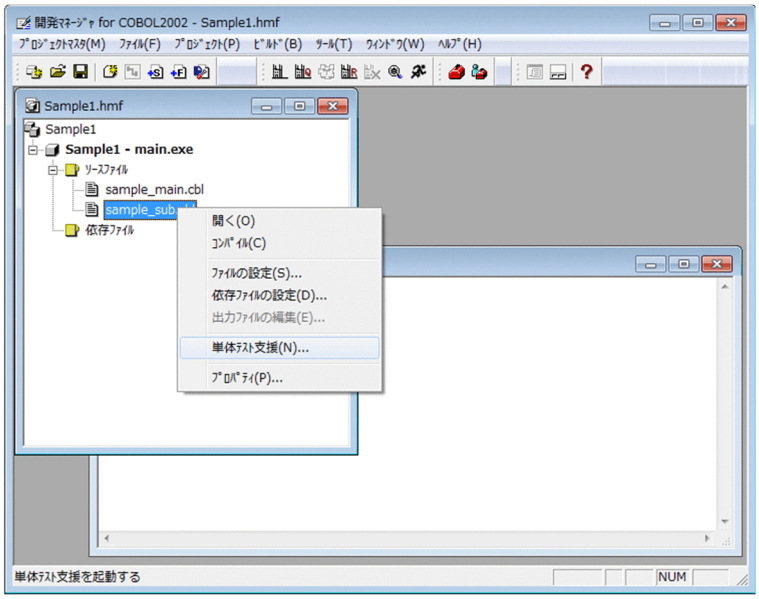
4.2.1　開発マネージャからテストプロジェクトを作成する

開発マネージャのツリービューウィンドウで表示した COBOL ソースファイルを選択して，単体テスト支援のテストプロジェクトを作成する手順です。

作業の手順

1. 開発マネージャで，単体テストを実行するプロジェクトマスタを開きます。
2. 開発マネージャのツリービューウィンドウから単体テストを実行する COBOL ソースファイルを選び，次のどちらかの手順で［単体テスト支援］を選択します。
   - 開発マネージャの［ツール］－［単体テスト支援］－［単体テスト支援］を選択
   - COBOL ソースファイルを右クリックし［単体テスト支援］を選択

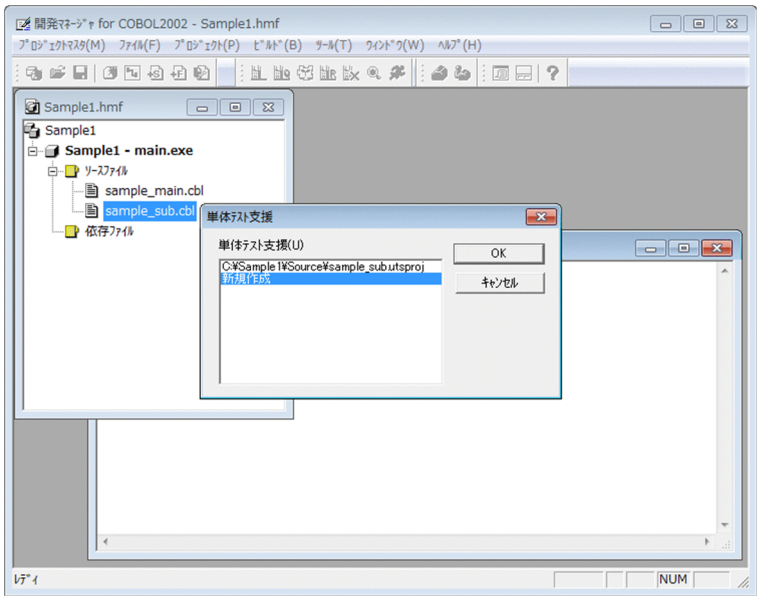
     ［ファイルの登録］ダイアログが表示されます。



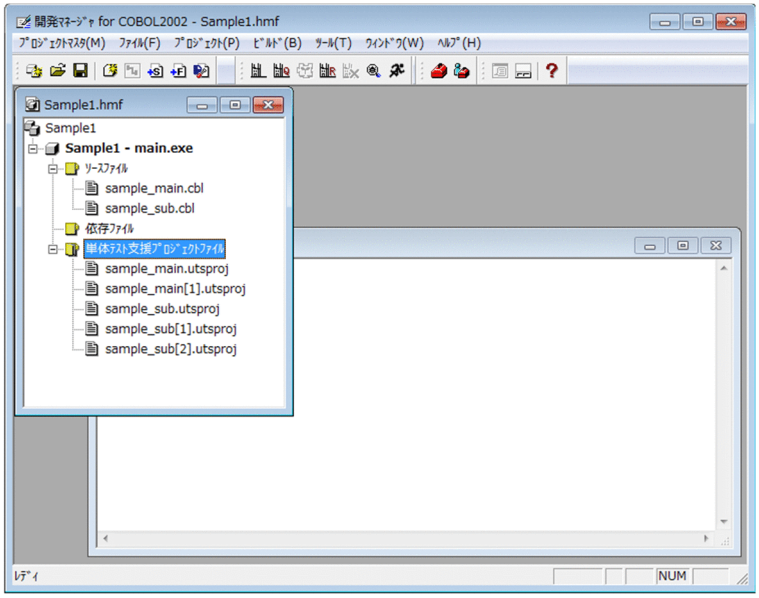
3. ［ファイルの登録］ダイアログで，単体テストプロジェクトファイルの保存場所と名前を指定します。

デフォルトでは，保存場所には COBOL ソースファイル名が格納されているフォルダが，名前には COBOL ソースファイル名が仮定されます。

1 つの COBOL ソースファイルに対して，複数のテストプロジェクトファイルを設定できます。COBOL ソースファイルに対してテストプロジェクトファイルがすでに登録されている場合は，既存のテストプロジェクトファイルを開くか，新規作成するのかを選択するダイアログが表示されます。



すでに登録されているテストプロジェクトファイルを選択すると，選択したテストプロジェクトが開きます。新規作成を選択すると，新規でテストプロジェクトファイルが作成されます。このとき，ファイルダイアログのファイル名の初期値は，COBOL ソースファイル名の後ろに数字［1］［2］が付いたファイル名です。sample_main.cbl に対して 2 つ，sample_sub.cbl に対して 3 つのテストプロジェクトファイルを追加した例を次に示します。



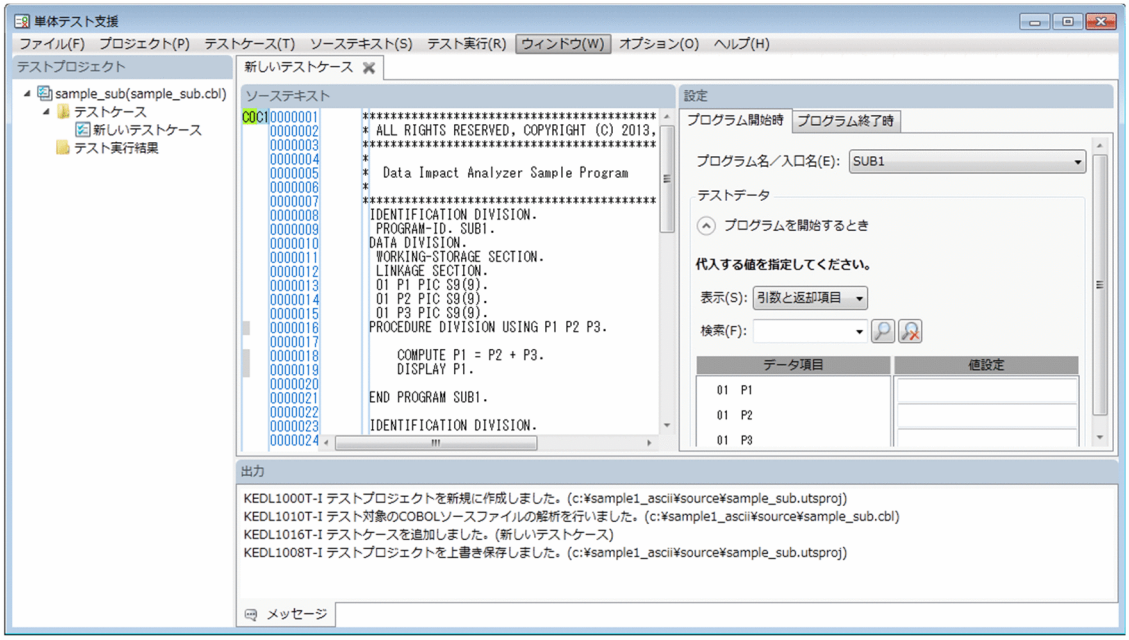
テストプロジェクトファイルを指定するときは，ほかのテストプロジェクトファイルが存在しないフォルダのファイルを指定してください。すでにあるテストプロジェクトファイルは指定できません。

指定したフォルダにほかのテストプロジェクトファイルが存在する場合，またはすでにあるテストプロジェクトファイルを指定した場合は，エラーになります。この場合，COBOL ソースファイルの選択からやり直してください。

4. ［ファイルの登録］ダイアログの［開く］ボタンをクリックします。

COBOL ソースファイル内に複数のプログラム名が定義されているときは，テスト対象プログラムを指定するダイアログが表示されます。

単体テスト支援が起動されます。



開発マネージャのツリービューウィンドウには，単体テスト支援プロジェクトファイルのノードが追加され，テストプロジェクトファイルはその下に追加されます。

注意

- 開発マネージャから起動して新しいテストプロジェクトを作成する場合，テストプロジェクトの作業フォルダは，開発マネージャのプロジェクトフォルダの下を指定することをお勧めします。開発マネージャのプロジェクトフォルダをコピーまたは移動するときに，同時にテストプロジェクトのコピーまたは移動ができます。
- 開発マネージャから COBOL ソースファイルを指定して単体テスト支援を起動すると，新しくテストプロジェクトが作成され，単体テスト支援のメインウィンドウが表示されます。新しいテストプロジェクトは，開発マネージャの「プロジェクトの設定」と「ファイルの設定」で指定したコンパイラオプション，コンパイラ環境変数およびテスト対象プログラムの種類を引き継ぎます。これらの設定は，単体テスト支援の［プロパティページ］で変更できます。
- 開発マネージャから起動して新しいテストプロジェクトを作成する場合は，開発マネージャで設定しているコンパイラオプションとコンパイラ環境変数を引き継いで，テストプロジェクトを生成します。このため，「新しいテストプロジェクトを作る」ダイアログは表示されません。
- 開発マネージャに登録したテストプロジェクトは，開発マネージャのツリービューウィンドウで削除できますが，テストプロジェクトファイルなど，テストプロジェクトを構成するファイルとそのフォルダは削除されません。テストプロジェクトが不要なときは，Windows のエクスプローラーからフォルダとファイルを削除してください。なお，開発マネージャのツリービューウィンドウからテストプロジェクトを削除しても，テストプロジェクトを構成するファイルとそのフォルダは削除されないため，単体テスト支援からテストプロジェクトを開けます。
- テストプロジェクトの作成後に開発マネージャの［プロジェクト設定］ダイアログの設定を変更しても，単体テスト支援のテストプロジェクトには反映されません。単体テスト支援の［プロパティページ］で，設定を変更してください。
- 開発マネージャから起動してテストプロジェクトを作成したあとで，単体テスト支援の［プロパティページ］を変更しても，開発マネージャのプロジェクトには反映されません。単体テスト支援の［プロパティページ］のコンパイラオプション・コンパイラ環境変数を変更した場合は，開発マネージャの［プロジェクト設定］ダイアログの設定を変更してください。
- 開発マネージャから起動してテストプロジェクトを作成したあとで，開発マネージャのプロジェクトファイルのコピーや移動で，COBOL ソースファイルのパスを変更した場合，単体テスト支援の［プロパティページ］で COBOL ソースファイルのパスを変更してください。上記以外で，開発マネージャから起動してテストプロジェクトを作成したあとで，単体テスト支援の［プロパティページ］で COBOL ソースファイルのパスを変更しても，開発マネージャのプロジェクトには反映されません。
- 既存のテストプロジェクトは，開発マネージャに直接登録できません。登録する手順を次に示します。
  1. 開発マネージャを起動して，COBOL ソースファイルを指定して，空のフォルダに既存のテストプロジェクトファイルと同じ名前のテストプロジェクトファイルを新しく作成します。
  2. Windows のエクスプローラーで，手順 1.で作成したテストプロジェクトのフォルダに既存のテストプロジェクトフォルダのファイルを上書きでコピーします。
- 単体テスト支援の環境変数は，開発マネージャ上で設定しないでください。

  単体テスト支援の環境変数については，「付録 D　単体テスト支援の環境変数」を参照してください。

- 次の個所で設定した環境変数の合計値が 65,535 バイトを超えると，新しくテストプロジェクトを作るときに論理エラーになります。回避するには，不要な環境変数を削除してください。
  - ［コンパイラ環境変数］テキストボックス
  - 開発マネージャで設定したコンパイラ環境変数（開発マネージャから起動した場合）
  - システムに設定した環境変数
- テストプロジェクト名に「-」（ハイフン）で始まる名前を指定しないでください。指定した場合，テスト実行時にエラーになります。エラーになった場合，テストプロジェクトを閉じ，テストプロジェクトファイルのファイル名を「-」（ハイフン）で始まらない名前に変更してからテストプロジェクトを開き直してください。

関連項目

- 「10.2　単体テスト支援のメインウィンドウ」

次の作業

「4.3　テストケースを作成する」

4.2.2　Windows のプログラム一覧からテストプロジェクトを作成する

Windows のプログラム一覧から単体テスト支援を開始する方法です。

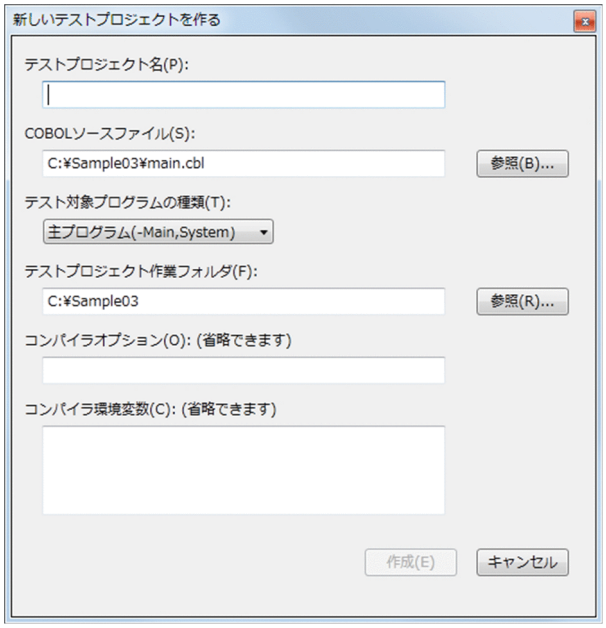
作業の手順

1. Windows のプログラム一覧から［COBOL2002 Professional Tool Kit］－［単体テスト支援］を選択します。

   単体テスト支援のスタートページが表示されます。

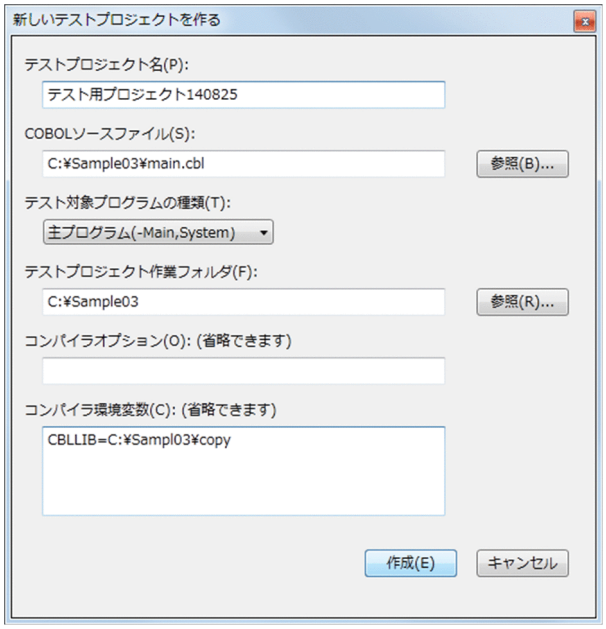
2. ［新しいテストプロジェクトを作る］ボタンをクリックします。

［新しいテストプロジェクトを作る］ダイアログが表示されます。



3. ［新しいテストプロジェクトを作る］ダイアログに項目を設定します。

例えば，COPY 文を使用しているプログラムの場合，コンパイラ環境変数に CBLLIB を指定してください。



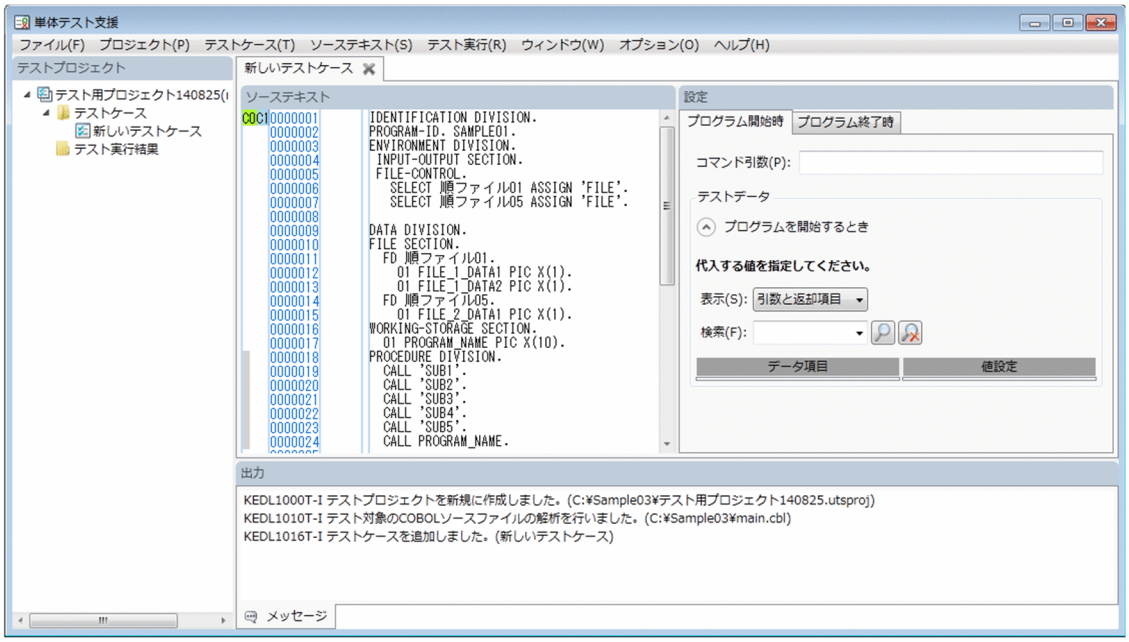
注意事項

テストプロジェクト名に「-」（ハイフン）で始まる名前を指定しないでください。指定した場合，テスト実行時にエラーになります。エラーになった場合，テストプロジェクトを閉じ，テストプロジェクトファイルのファイル名を「-」（ハイフン）で始まらない名前に変更してからテストプロジェクトを開き直してください。

4. ［新しいテストプロジェクトを作る］ダイアログの［作成］ボタンをクリックします。

COBOL ソースファイル内に複数のプログラム名が定義されているときは，テスト対象プログラムを指定するダイアログが表示されます。

［テスト設定］画面に，［新しいテストケース］としてテスト対象プログラムが表示されます。



関連項目

- 「10.1　［新しいテストプロジェクトを作る］ダイアログ」
- 「10.2　単体テスト支援のメインウィンドウ」
- 「10.3　［テストプロジェクト］画面」
- 「10.4　［テスト設定］画面」
- 「10.5　［スタートページ］」

次の作業

「4.3　テストケースを作成する」

4.3　テストケースを作成する

テストプロジェクトにテストケースを作成する手順です。

4.3.1　テストケースを追加する

テストプロジェクトにテストケースを追加する手順です。

作業の手順

1. ［テストケース］－［追加］メニューを選択します。

   テストプロジェクトにテストケースが追加されます。

   テストケースの名前を変更するときは，次のどちらかの操作をしてください。

   - ［テストプロジェクト］画面でテストケースを選択し，［テストケース］－［名前の変更］メニューを選択
   - ［テストプロジェクト］画面のテストケースを右クリックして，ポップアップメニューの［名前の変更］メニューを選択

注意

- コンパイルエラーなどで，テスト対象の COBOL ソースファイルの解析が完了していない場合は，テストケースに対する操作はできません。
- テストケースの個数や設定するテストデータの個数の総数が多過ぎると，メモリ不足で異常終了することがあります。個々の上限値や総数の目安は，「付録 C　単体テスト支援の制限値と限界値」を参照してください。

関連項目

- 「10.3　［テストプロジェクト］画面」

次の作業

「4.4　テストデータを設定する」

4.3.2　テストケースをコピーして追加する

すでにあるテストケースをコピーして追加する手順です。

設定済みのテスト項目の一部を修正して，新しいテストケースにするときは，すでにあるテストケースをコピーすると便利です。

作業の手順

1. ［テストプロジェクト］画面で，コピーを作成するテストケースを選択します。

2. 次のどちらかの操作をします。

   - ［テストケース］－［コピーして追加］メニューを選択
   - ［テストプロジェクト］画面のテストケースを右クリックして，ポップアップメニューの［コピーして追加］メニューを選択

   コピー元のテストケースの名前に_(1) が付いた名前で，コピーのテストケースが追加されます。

   テストケースの名前を変更するときは，次のどちらかの操作をしてください。

   - ［テストプロジェクト］画面でテストケースを選択し，［テストケース］－［名前の変更］メニューを選択
   - ［テストプロジェクト］画面のテストケースを右クリックして，ポップアップメニューの［名前の変更］メニューを選択

注意

- コンパイルエラーなどで，テスト対象の COBOL ソースファイルの解析が完了していない場合は，テストケースに対する操作はできません。

次の作業

「4.4　テストデータを設定する」

4.3.3　テストケースの名前を変更する

テストケースの名前を変更する手順です。

新しいテストケースの場合は，テストケースを識別できる名前に変更することをお勧めします。

テストケースの名前を変更したあとで，単体テスト支援を実行しても，変更する前の実行で作成されたテスト実行結果の名前は変わりません。

作業の手順

1. ［テストプロジェクト］画面で，名前を変更するテストケースを選択します。

2. 次のどちらかの操作をします。

   - ［テストケース］－［名前の変更］メニューを選択
   - ［テストプロジェクト］画面のテストケースを右クリックして，ポップアップメニューの［名前の変更］メニューを選択

   ［名前の変更］ダイアログが表示されます。

3. ［名前の変更］ダイアログに新しい名前を入力し，［OK］ボタンをクリックします。

   テストケースの名前が変更されます。

注意

- テストケースには，次に示す文字は使用できません。

  「/」「>」「<」「?」「:」「"」「¥」「*」「|」
- コンパイルエラーなどで，テスト対象の COBOL ソースファイルの解析が完了していない場合は，テストケースに対する操作はできません。
- テストケースの名前では，英大文字と英小文字は区別されません。また，1 つのテストプロジェクトの中に同じ名前のテストケースは作成できません。テストケースの追加，コピー，または名前の変更で同じ名前を指定すると，自動的にテストケースの名前の末尾に通し番号が付けられます。

関連項目

- 「10.3 ［テストプロジェクト］画面」

4.3.4 テストケースを削除する

テストケースを削除する手順です。

作業の手順

1. ［テストプロジェクト］画面で，削除するテストケースを選択します。

2. 次のどちらかの操作をします。

   - ［テストケース］－［削除］メニューを選択
   - ［テストプロジェクト］画面のテストケースを右クリックして，ポップアップメニューの［削除］メニューを選択

   「テストケースを削除します。」のメッセージボックスが表示されます。

3. ［OK］ボタンをクリックします。

   テストケースがプロジェクトから削除されます。さらに，テストケースのために生成された TD コマンド格納ファイルも削除されます。

注意

- コンパイルエラーなどで，テスト対象の COBOL ソースファイルの解析が完了していない場合は，テストケースに対する操作はできません。

関連項目

- 「10.3 ［テストプロジェクト］画面」

4.3.5　テストケースの有効と無効を切り替える

テストケースごとに有効にするか無効にするかを切り替える手順です。

テストケースを一括して実行する前に，実行の対象外にするときに使用します。

テストケースを無効にすると，バッチ実行など，テストケースを一括して実行した場合でも，無効にしたテストケースは実行されません。

作業の手順

1. ［テストプロジェクト］画面で，テストケースを選択します。
2. 次のどちらかの操作をします。
   - ［テストケース］－［有効／無効］メニューを選択
   - ［テストプロジェクト］画面のテストケースを右クリックして，ポップアップメニューの［有効／無効］メニューを選択

   有効なテストケースの場合は無効になり，アイコンがグレーに変わります。

   無効なテストケースの場合は有効になり，アイコンのグレー表示が解除されます。

注意

- コンパイルエラーなどで，テスト対象の COBOL ソースファイルの解析が完了していない場合は，テストケースに対する操作はできません。

関連項目

- 「10.3　［テストプロジェクト］画面」

4.4　テストデータを設定する

テストケースごとに，テストデータを設定する手順です。

テストデータとして，次の項目を設定します。テストケースの個数や設定するテストデータの個数の総数が多過ぎると，メモリ不足で異常終了することがあります。個々の上限値や総数の目安は，「付録 C　単体テスト支援の制限値と限界値」を参照してください。

- 代入する値
  プログラムの初期値など，テストを実行するときにデータ項目に代入する値です。
- 実行結果を判定するための期待値
  データ項目の値と比較して正しいかどうかを判断する基準の値です。

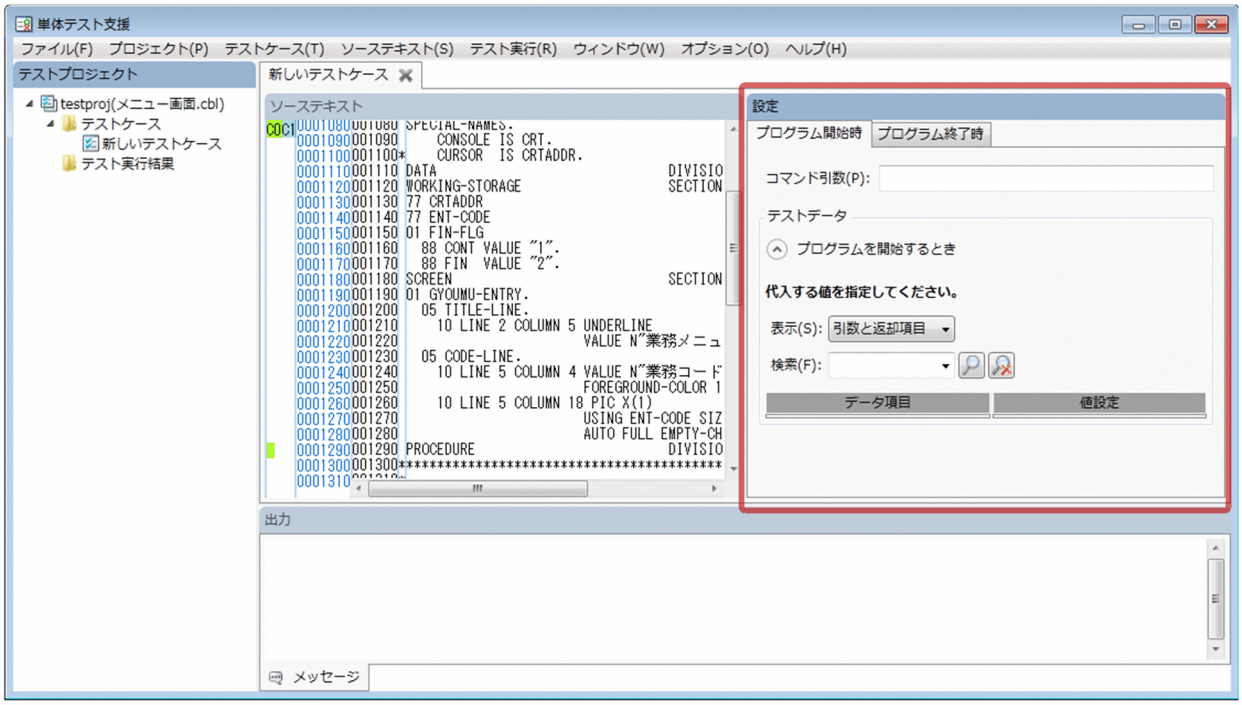
4.4.1　プログラム開始時の値を設定する

単体テストでプログラムを開始するときの値を設定します。

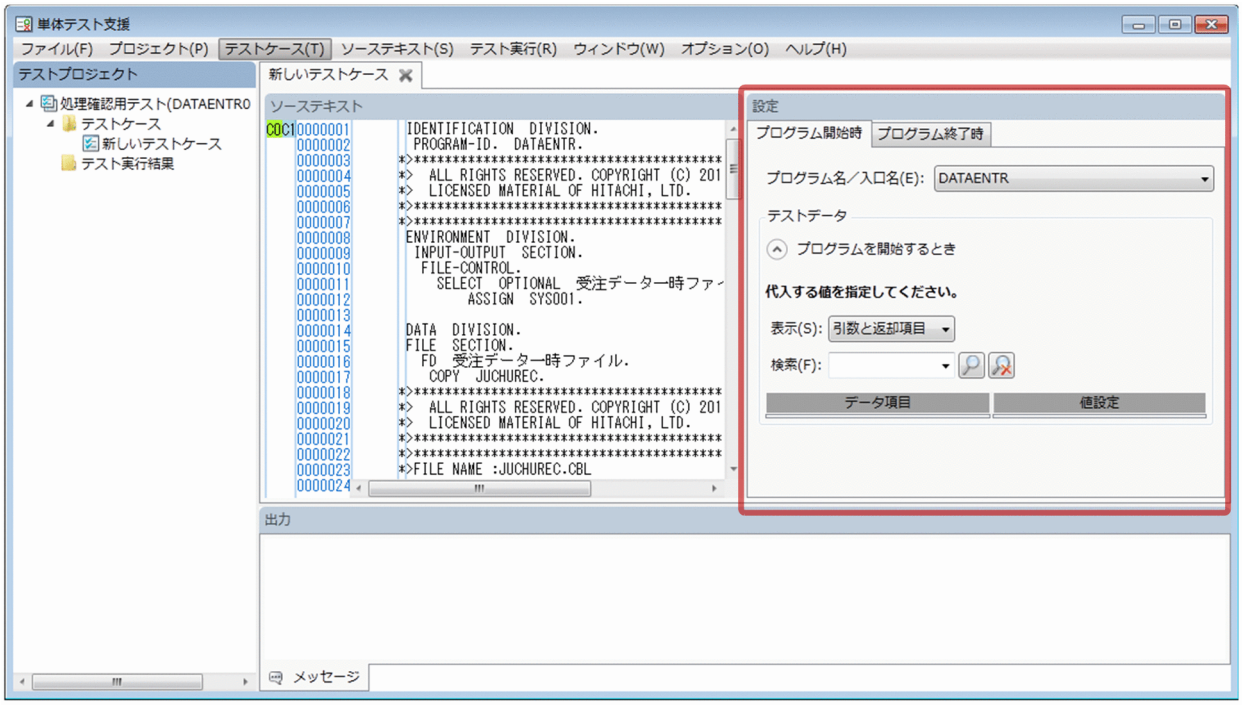
プログラム開始時の値は，単体テスト支援のメインウィンドウ右側の［プログラム開始時］画面に設定します。

テスト対象のプログラムの種類が主プログラムの場合と副プログラムの場合とでは，［プログラム開始時］画面に設定できる項目が異なります。主プログラムにはコマンド引数と初期値を，副プログラムには入口点と初期値を指定できます。

- テスト対象プログラムが「主プログラム（-Main,System)」「主プログラム（-Main,V3)」のとき



- テスト対象プログラムが「副プログラム」のとき



作業の手順

テスト対象プログラムが「主プログラム（-Main,System/-Main,V3)」のとき

1. ［テスト設定］画面の［プログラム開始時］画面を選択します。

   ［プログラム開始時］画面が表示されます。

2. ［コマンド引数］に，プログラムに渡す引数の値を設定します。

3. 「代入する値を指定してください。」の領域に，テストデータを設定します。

   テストデータの設定方法については，「4.4.7　データ項目に値または期待値を設定する」を参照してください。

テスト対象プログラムが「副プログラム」のとき

1. ［テスト設定］画面の［プログラム開始時］画面を選択します。

   ［プログラム開始時］画面が表示されます。

2. ENTRY 文があるプログラムのときは，［プログラム名／入口名］のリストから，プログラムを実行するときの入口点を選択できます。

   ENTRY 文がないプログラムの場合は，［プログラム名／入口名］にプログラム名が表示されます。

3. 「代入する値を指定してください。」の領域にテストデータを設定します。

   テストデータの設定方法については，「4.4.7　データ項目に値または期待値を設定する」を参照してください。

関連項目

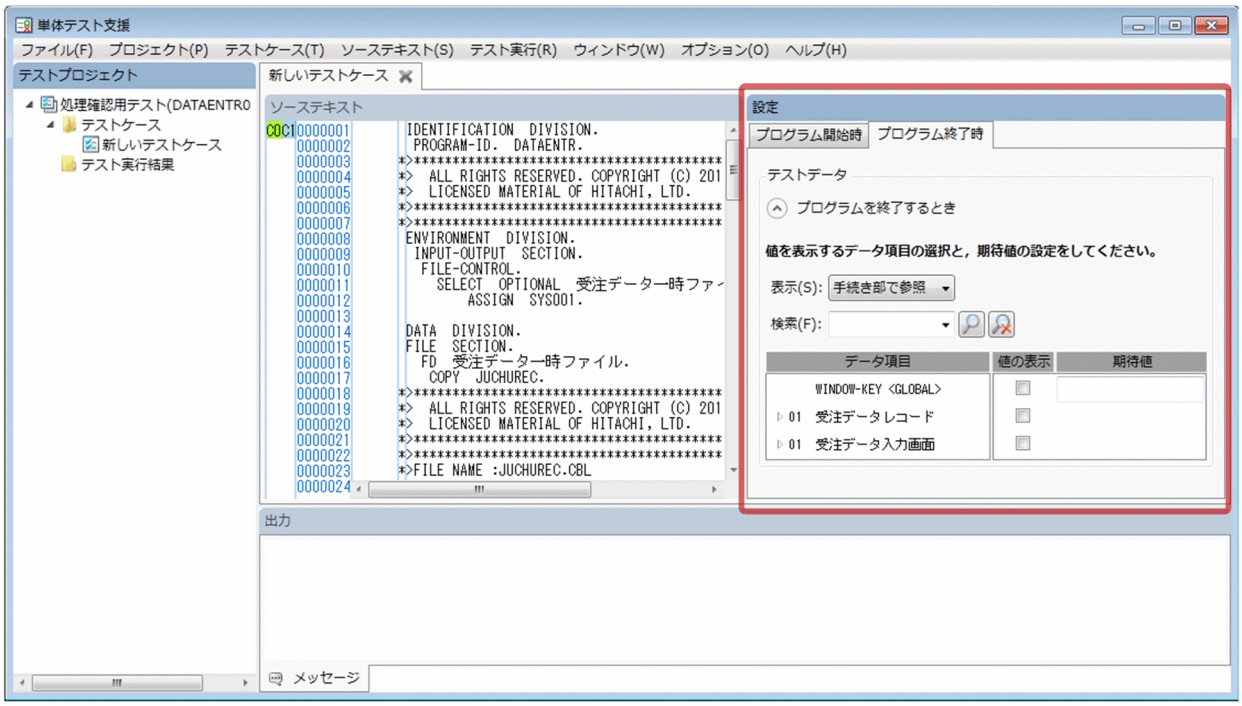
- 「9.4　データ項目検索の設定を変更する」
- 「10.8.2 ［プログラム開始時］画面」

4.4.2　プログラム終了時の値を設定する

プログラム終了時の値が正しいかどうかを判定するための期待値を［プログラム終了時］画面に設定する手順です。

作業の手順

1. ［テスト設定］画面の［プログラム終了時］画面を選択します。



2. 「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。」の「期待値」の領域に，期待値を設定します。

   期待値の設定方法については，「4.4.7　データ項目に値または期待値を設定する」を参照してください。

   一覧に表示されるデータ項目は，［表示］コンボボックスで切り替えられます。

注意

- テスト対象の COBOL プログラムが実行時に異常終了する場合は，プログラム終了時の処理は実行されません。
- 期待値を使って自動判定するには，［値の表示］チェックボックスをオンにする必要があります。オフだと判定は実行されません。

- STOP RUN 文や EXIT 文の実行で，中断点やプログラム終了時の期待値の判定が実行されない場合は，［値の表示］チェックボックスをオンにして［期待値］を設定しても，判定結果は未判定になります。

関連項目

- 「4.10　単体テストを実行したときの判定結果」
- 「9.4　データ項目検索の設定を変更する」
- 「10.8.3 ［プログラム終了時］画面」

4.4.3　ファイルシミュレーションのテストデータを設定する

ファイルシミュレーションのテストデータを設定する手順です。

ファイルシミュレーションには，ファイル入出力文が実行されたときに使用する，次のテストデータを設定します。

- 入力文が実行されたときに，レコード領域に代入する代入値
- 出力文が実行されたときに，レコード領域の値を検証するための期待値
- 出力文が実行されたときに，レコード領域の値を表示するかどうかの設定
- 入力文または出力文が実行されたときに発生する終了条件の設定

ファイルシミュレーション用の設定は，1 回の入出力で使用するテストデータ（テストデータセット）を複数準備しておきます。対象の入出力文が実行されたタイミングで，順次使用されます。

例

テスト対象プログラム

```
          :
002300  PROCEDURE DIVISION.
          :
003500  OPEN INPUT FILE001.
          :
003600  READ FILE001.
          :
005500  READ FILE001.
          :
```

003600 と 005500 どちらの READ 文で使われるかは，プログラムの実行順で決まります。READ 文が実行されるたびに，テストデータが，設定した順に使用されます。

設定したテストデータセットを順に使用するため，ファイルシミュレーションは次のような動作になります。

- 同一ファイルに対して WRITE 文，READ 文の順に実行するプログラムの場合に，WRITE 文で使ったテストデータが次の READ 文でテストデータとして設定されることはありません。WRITE 文で使用したテストデータではなく，READ 文用のテストデータが代入値として設定されます。

- ISAM ファイルなどで，検索キーに一致したテストデータが読み込まれるのではなく，検索キーとは関係なく READ 文用のテストデータが代入値として，指定した順に設定されます。

終了条件は，準備したテストデータをすべて使用したあとで判定されます。そのため，設定したテストデータや終了条件が，どこの入出力文で使用されるかは，設定したテストデータの個数やプログラムの実行順序に依存します。中断点の設定のように，特定の場所にある入出力文に対して，テストデータや終了条件は設定できません。

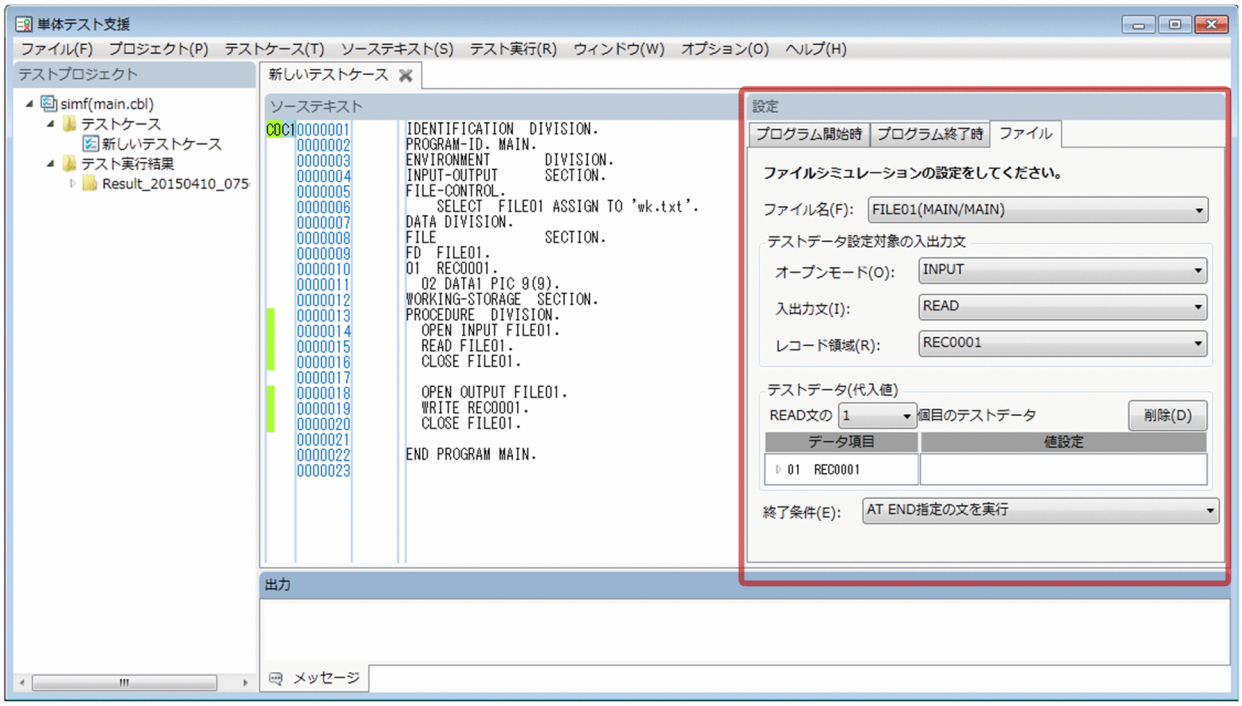
設定できるテストデータ

ファイルシミュレーションでは，次に示す条件をすべて満たすファイルの入出力文に対して，テストデータを設定できます。

- ［スタブ設定ページ］の「入出力するファイル名」欄のチェックボックスがオンになっている。
- テスト対象プログラム内（入れ子を含む）にファイル記述項がある。
- EXTERNAL 指定がない場合は，テスト対象プログラム内（入れ子を含む）に対象ファイルの OPEN 文がある。

作業の手順

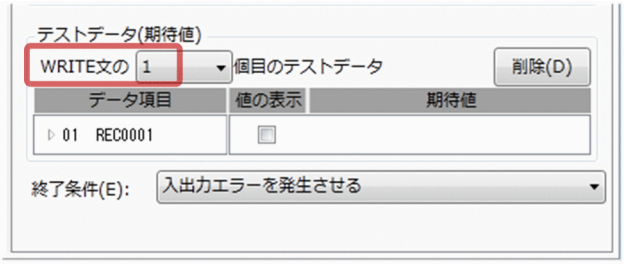
1. ［テスト設定］画面の［ファイル］タブを選択して［ファイルシミュレーション］画面を表示します。



2. ［ファイル名］の一覧から，シミュレーションの対象とするファイル名を選択します。

3. ［オープンモード］，［入出力文］および［レコード領域］から，テストデータを設定したい入出力文を選択します。

   入出力文に WRITE 文または REWRITE 文を指定すると，「テストデータ」グループの表示が次に変わります。



4. 入出力文の何個目のテストデータとするかを「テストデータ」グループのコンボボックスで指定します。

   新しいテストデータを作成する場合は，コンボボックスから「追加」を選択します。「追加」を選択すると，最後にテストデータが追加され（個数が 2，3，4...と増えていく），入力できる状態になります。

5. テストデータを設定します。

   設定方法については，「4.4.7　データ項目に値または期待値を設定する」および「4.4.8　値の入力規則」を参照してください。

6. 終了条件を設定する場合は［終了条件］コンボボックスから，判定する終了条件を選択します。終了条件が不要な場合は「なし」を選択します。

注意

- 条件に合致していても，ファイル記述項に複数のレコード領域の定義がある場合は，そのうちの 1 つのレコード領域にだけしかテストデータは設定できません。また，テストデータは，入出力文（READ 文，WRITE 文，REWRITE 文）に対してだけ設定できます。START 文と DELETE 文にはテストデータを設定できません。

  入出力文に対して設定したテストデータは，次に示す用途で使用されます。

入出力文	用途
READ	代入値
WRITE	期待値
REWRITE	期待値

- テストデータは，ファイル名，オープンモードおよび入出力文を 1 つの組み合わせとして，組み合わせごとに設定できます。OPEN 文ごとには設定できません。また，テストデータは，1 回の入出力文で使用するテストデータとしてデータ項目ごとに指定でき，これらをまとめてテストデータセットとします。その上で，各入出力文の想定する実行回数分のテストデータセットと，1 つの終了条件を設定できます。

項番	オープンモード	入出力文	設定できるテストデータセットの数	設定できる終了条件の数
1	INPUT	READ	入出力のテスト分	1 種類
2	OUTPUT	WRITE	入出力のテスト分	1 種類
3	I-O	READ	入出力のテスト分	1 種類
4		WRITE	入出力のテスト分	1 種類

項番	オープンモード	入出力文	設定できるテストデータセットの数	設定できる終了条件の数
5		REWRITE	入出力のテスト分	1 種類
6	EXTEND	WRITE	入出力のテスト分	1 種類

EXTERNAL 指定がないファイルの場合は，オープンモードは実際にテスト対象プログラム内で指定しているオープンモードだけが選択できます。EXTERNAL 指定があるファイルの場合は，すべてのオープンモードが表示されます。

また，入出力文は，テスト対象プログラム内で使用していなくても選択できます。そのため，例えば上記の場合で，実際にテスト対象プログラム内では，オープンモードは I-O だけ，入出力文は READ 文だけ使用している場合は，項番 3，4，5 の 3 種類のテストデータだけを設定できます。

- 終了条件は，EXTERNAL 指定がないファイルの場合は，入出力文およびファイルのアクセス種別に応じて，次に示す条件を指定できます。EXTERNAL 指定があるファイルの場合は，アクセス種別の影響は受けません。

入出力文	アクセス種別	終了条件
READ	順アクセス	AT END 指定の文を実行
	乱アクセス	INVALID KEY 指定の文を実行
	動的アクセス	AT END 指定の文を実行
		INVALID KEY 指定の文を実行
WRITE	順アクセス	AT END-OF-PAGE 指定の文を実行
	乱アクセス	INVALID KEY 指定の文を実行
	動的アクセス	INVALID KEY 指定の文を実行
REWRITE	順アクセス	－ この組み合わせで特別に選択できるものはありません。
	乱アクセス	INVALID KEY 指定の文を実行
	動的アクセス	INVALID KEY 指定の文を実行

上記の条件に加えて，入出力文およびファイルのアクセス種別に関係なく，次に示す条件も指定できます。

入出力文	アクセス種別	終了条件
すべて	すべて	入出力エラーを発生させる
		なし（終了条件を指定しない）

- テスト対象の COBOL ソースファイル中で OPEN を実行すると，ファイルシミュレーションが開始します。このため，OPEN 文がないプログラムではテストデータは設定できません。

- ファイルシミュレーションが有効な場合は，実体ファイルにはアクセスされません。テスト実行時に実体ファイルにアクセスするときは，［スタブ設定ページ］で対象のファイルをオフにし，実体ファイルを割り当ててください。
- ［スタブ設定ページ］で対象のファイルをオンにすると，テストデータが設定されていなくてもシミュレーションは実行されます。入出力文ごとにシミュレーションのオン/オフの切り替えはできません。
- 終了条件を設定した場合は，設定したテストデータがすべて使用されたあとで終了条件が判定されます。

  例えば，READ 文用に 3 つのデータを設定し，かつ，終了条件に「AT END 指定の文を実行」を指定すると，最初の 3 回の READ 文でテストデータが設定され，4 回目の READ 文で終了条件の「AT END 指定の文を実行」が判定されます。

  また，デフォルトでテストデータセットが 1 つ準備されるため，1 回目の入出力文で終了条件を判定させる場合は，テストデータセットを削除して，0 個（［テストデータセット番号］は「-」と表示される）にしてください。
- テストデータの設定後にレコード領域を切り替えると，設定済みのテストデータが削除されることを確認するダイアログが表示されます。［はい］ボタンをクリックするとレコード領域は切り替わりますが，それまでに設定したテストデータは削除されます。
- プログラム上にない入出力文に対しても，テストデータは設定できます。ただし，入出力文が実行されないので，該当するシミュレーションも実行されません。
- 入出力文よりもテストデータが少ない場合，テストデータの個数を超えた分の入出力文では，最後のテストデータが繰り返し使用されます。終了条件が指定されている場合は，終了条件が繰り返し判定されます。

  例 1：

出力文の実行回数	使用したテストデータ
1回目	1個目のテストデータ
2回目	2個目のテストデータ
3回目	2個目のテストデータ
4回目	2個目のテストデータ

最後に指定したテストデータが繰り返し使用されます。

  例 2：

出力文の実行回数	使用したテストデータ
1回目	1個目のテストデータ
2回目	2個目のテストデータ
3回目	終了条件が発生
4回目	終了条件が発生

終了条件が繰り返されます。

- 同じオープンモードが複数ある場合，2 回目以降のオープン後は，設定したテストデータの最初から使用されます。

  例：

出力文の実行回数	使用したテストデータ
1回目	1個目のテストデータ
2回目	2個目のテストデータ
3回目	1個目のテストデータ
4回目	2個目のテストデータ
5回目	3個目のテストデータ

2度目に開いたときには，
最初のテストデータから使用されます。
そのため，この条件では，4番目と5番目の
データは使われません。

出力文の回数は，2回目以降のオープンでも，
前の回数から継続でカウントされます。

- 手続き文中に USING 指定の SORT 文または MERGE 文がある場合，終了条件で「AT END 指定の文を実行」を必ず指定してください。指定しないと，実行時エラーまたはテスト実行が終了しなくなります。
- INTO 指定がある READ 文のシミュレーションを実行すると，設定したテストデータが，シミュレーションの実行後に INTO で指定したデータ項目にも転記されます。
- シミュレーションを実行すると，FILE STATUS 句に指定されたデータ項目の値が更新されます。

  更新される値については，マニュアル「COBOL2002 操作ガイド」の論理誤りチェックと入出力状態の値の説明を参照してください。
- ファイル記述項に EXTERNAL 指定がある場合，このファイルに対して，テスト対象プログラム以外で入出力文を実行すると，その入出力文に対してもシミュレーションが実行されます。この場合，［出力文の実行結果］一覧や［入力文の実行結果］一覧に，テスト対象プログラム以外で実行した結果が表示されます。EXTERNAL 指定の場合は，テストデータの個数や内容も，テスト対象プログラム以外の入出力文を考慮してください。

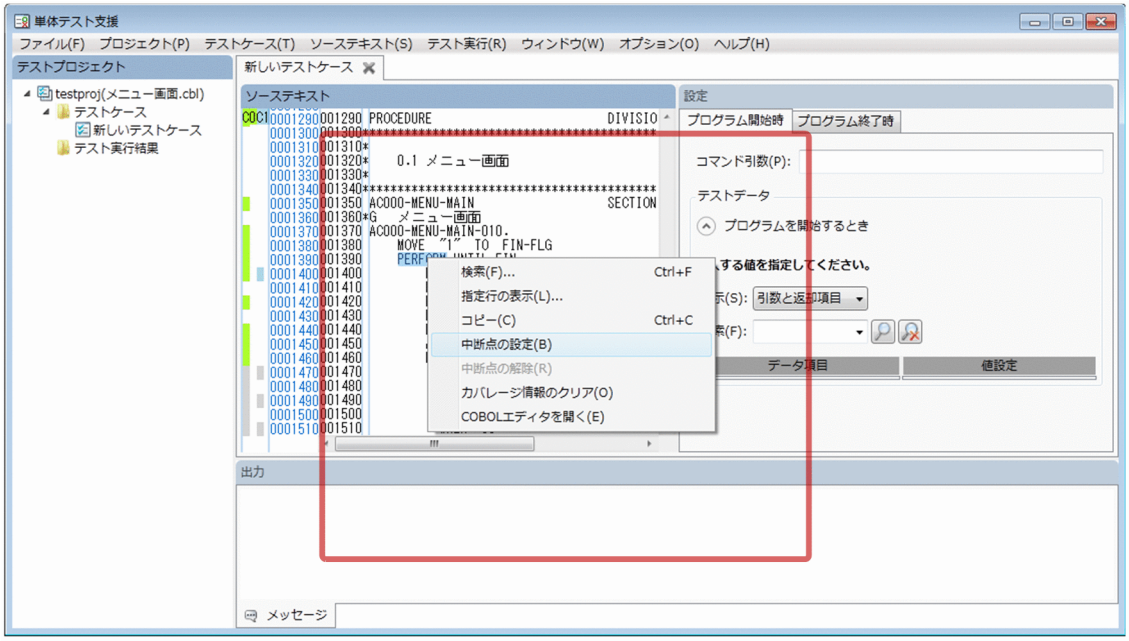
4.4.4　中断点を設定する

テスト対象のプログラムの手続き部（PROCEDURE DIVISION）に中断点を設定する手順です。

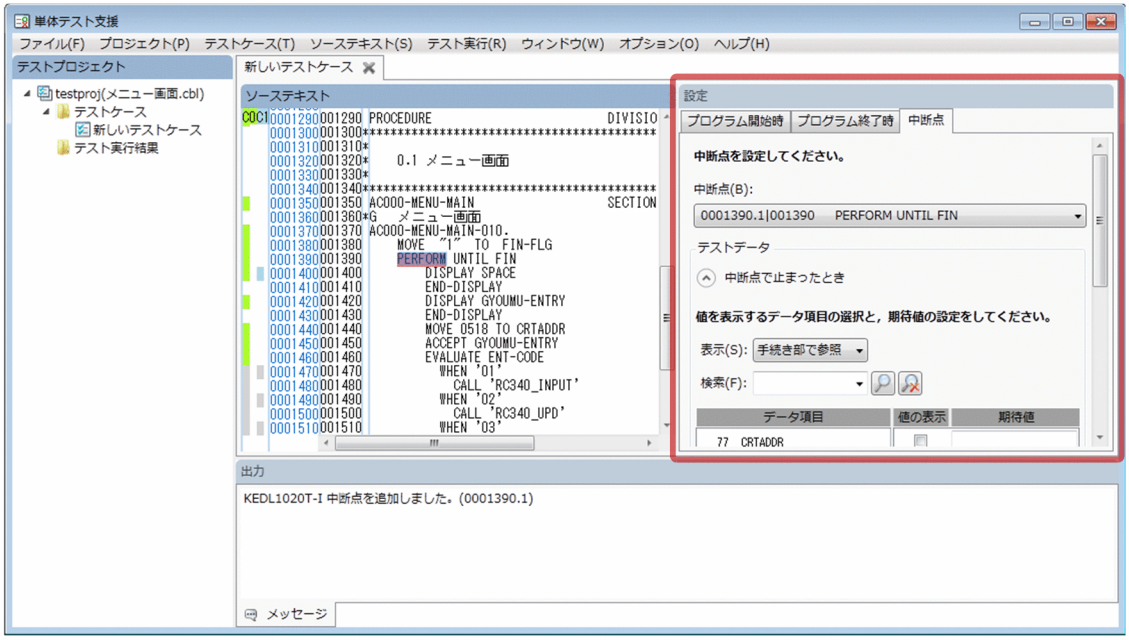
設定した中断点には「4.4.5　中断点のテストデータを設定する」の手順で，テストデータを設定できます。

作業の手順

1. ［ソーステキスト］画面で，中断点を設定する文（節または段落名を含む）の文キーワードを選択します。
2. ［ソーステキスト］－［中断点の設定］メニューを選択します。



［中断点］画面が表示されます。



注意

- テストデータを設定済みの中断点を解除した場合，設定したテストデータも削除されます。

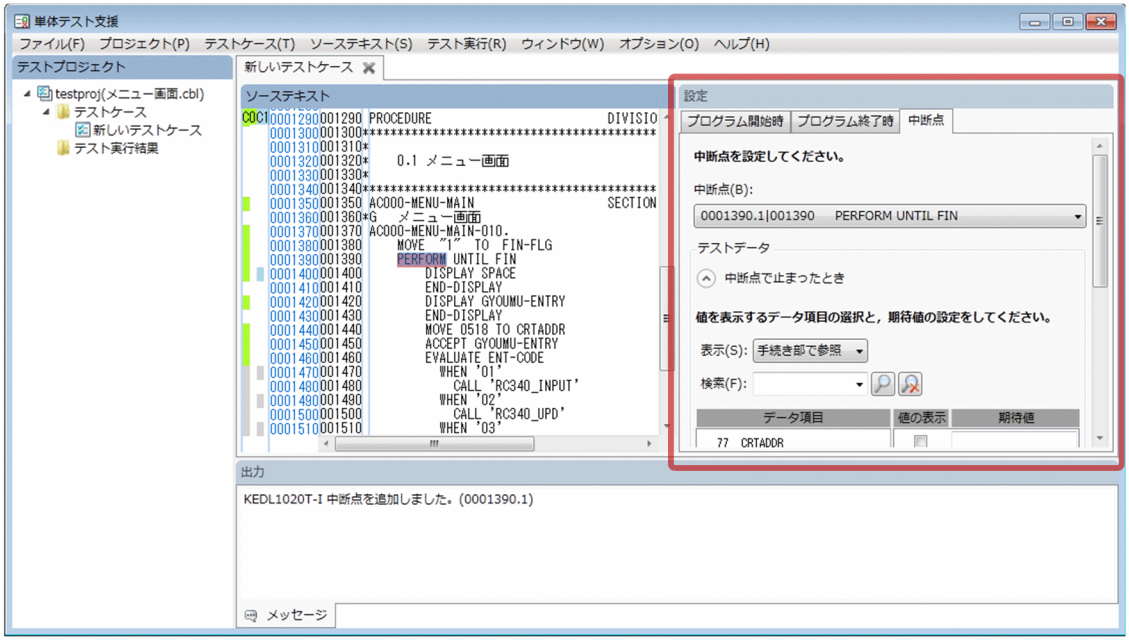
4.4.5　中断点のテストデータを設定する

プログラムを中断したときの値が正しいかどうかを判定するための期待値と，値の代入を［中断点］画面に設定する手順です。

作業の手順

1. ［テスト設定］画面の［中断点］画面を選択します。

   ［中断点］画面が表示されます。［中断点］コンボボックスを開くと，設定されている中断点の一覧が表示されます。



   先頭に，行番号と位置番号をピリオドでつないだ文番号が表示されます。位置番号は，行中の文に対して順に振られる番号であり，先頭の文が 1 になります。

2. ［中断点］画面の［中断点］コンボボックスから，対象とする中断点を選択します。

3. 「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。」の「期待値」の領域に，期待値を設定します。

   一覧に表示されるデータ項目は，［表示］コンボボックスで切り替えられます。

   期待値の設定方法については，「4.4.7　データ項目に値または期待値を設定する」を参照してください。

4. 中断点から実行するときは，「代入する値を指定してください。」の領域にテストデータを設定します。

   テストデータの設定方法については，「4.4.7　データ項目に値または期待値を設定する」を参照してください。

注意

- 中断点はテストケースごとに設定します。別のテストケースで同じ個所にテストデータを設定する場合には，そのテストケースで同じように中断点を設定してください。
- 中断点を設定したあとで COBOL ソースファイルを修正し，［COBOL ソースファイルの解析］を実行した場合，中断点は同一の文番号の COBOL の文に対して再設定されます。同一の文番号のソース内の文が削除された場合，またはコメントアウトされた場合は，中断点は解除されます。

関連項目

- 「4.10　単体テストを実行したときの判定結果」
- 「9.4　データ項目検索の設定を変更する」
- 「10.8.5　［中断点］画面」

4.4.6　中断点を解除する

設定した中断点を解除する手順です。

作業の手順

1. ［ソーステキスト］画面で中断点を解除する文（節および段落名を含む）の文キーワードを選択します。
2. ［ソーステキスト］－［中断点の解除］メニューを選択します。
   中断点が解除されます。

注意

- 中断点を解除するときは，中断点を解除する対象の文（節および段落名を含む）の文キーワードを選択して実行します。

4.4.7　データ項目に値または期待値を設定する

テストデータのデータ項目に値または期待値を設定する手順です。

［プログラム開始時］画面と［中断点］画面で値を代入できます。［プログラム終了時］画面と［中断点］画面で期待値を設定できます。値の入力規則については，「4.4.8　値の入力規則」を参照してください。

作業の手順

［プログラム開始時］画面でデータ項目に値を代入する場合

1. ［テスト設定］画面の［プログラム開始時］画面を選択します。
   ［プログラム開始時］画面が表示されます。
2. 「代入する値を指定してください。」の領域に，代入するデータ名または値を設定します。
   ［表示］または［検索］で，代入するデータ項目を表示して，［値設定］に値を指定します。

［中断点］画面でデータ項目に値を代入する場合

1. ［テスト設定］画面の［中断点］画面を選択します。
   ［中断点］画面が表示されます。
2. 「代入する値を指定してください。」の領域に，代入するデータ名または値を設定します。

［表示］または［検索］で，代入するデータ項目を表示して，［値設定］に値を指定します。

［中断点］画面で期待値を設定する場合

1. ［テスト設定］画面の［中断点］画面を選択します。

   ［中断点］画面が表示されます。

2. ［値を表示するデータ項目の選択と，期待値を設定してください。］の領域に，期待値として設定するデータ名または値を設定します。

   ［表示］または［検索］ボタンで期待値を設定するデータ項目を表示して，［期待値］に値を設定します。

［プログラム終了時］画面で期待値を設定する場合

1. ［テスト設定］画面の［プログラム終了時］画面を選択します。

   ［プログラム終了時］画面が表示されます。

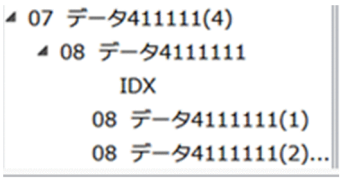
2. ［値を表示するデータ項目の選択と，期待値を設定してください。］の領域に，期待値として設定するデータ名または値を設定します。

   ［表示］または［検索］ボタンで期待値を設定するデータ項目を表示して，［期待値］に値を設定します。

［プログラム開始時］画面でデータ項目に値を代入する例を次に示します。集団項目は，データ項目をクリックして，基本項目を表示して代入してください。

注意

- ［値の代入の設定］一覧と［データ項目の値の表示/期待値の設定］一覧の，最後のデータ項目の後ろに「...」が表示されたときは，データ項目の数が上限値を超えています。この場合は，［検索文字］を指定し，表示するデータ項目を上限値以下に減らすと，目的のデータ項目を表示できます。



- 単体テスト支援がアドレスを参照できないデータ項目に対して，値の代入または期待値を指定して単体テストを実行した場合，［メッセージ］画面にエラーメッセージが表示され，失敗したテストケースの実行はスキップされます。アドレスを参照できないデータ項目を次に示します。
  - 連絡節で定義されているが，手続き部で参照されないデータ項目
  - 不正な値のアドレス名によって ADDRESSED 句の指定で参照されるデータ項目
- ［値設定］に入力した値は，次に示す順序で［値の代入の設定］一覧に表示されたデータ項目に代入されます。

  制御変数，指標名，アドレス名以外は，［値の代入の設定］一覧の表示順に代入

  例 1：

データ項目	値設定
01 DAT1	'AAA'
01 DAT2	DAT1

この例の場合，次の順序で転記が実行されたことになります。

MOVE 'AAA' TO DAT1.

MOVE DAT1 TO DAT2.

例 2：

データ項目	値設定
01 DAT1	DAT2
01 DAT2	'AAA'

この例の場合，次の順序で転記が実行されたことになります。

MOVE DAT2 TO DAT1.

MOVE 'AAA' TO DAT2.

制御変数，指標名，アドレス名は，それ以外の［値の代入の設定］一覧に表示されたデータ項目の代入よりも前で代入

例 1：

データ項目	値設定
01 DAT1	'AAA'
01 DAT2	
02 DAT3	
02 DAT4	
I<DEPENDING>	2
02 DAT4(1)	'BBB'
02 DAT4(2)	DAT1

データ項目	値設定
:	
02 DAT5	
01 I	2

上記の例の場合，次の順序で転記が実行されたことになります。

MOVE 2 TO I.

MOVE 'AAA' TO DAT1.

MOVE 'BBB' TO DAT4(1).

MOVE DAT1 TO DAT4(2).

例 2：

データ項目	値設定
01 DAT1	&'DAT3'
01 DAT2	
02 ADR1 <ADDRESSED>	DAT1
02 DAT21	
01 DAT3.	'BBB'

この例の場合，次の順序で転記が実行されたことになります。

SET ADR1 TO DAT1.

COMPUTE DAT1 = FUNCTION ADDR(DAT3).

MOVE 'BBB' TO DAT3

この場合，ADR1 のアドレス名には未初期化の DAT1 の値を設定する動作になります。このため，DAT21 への値の代入を設定すると，アドレスを解決できないで単体テストの実行でエラーになります。

次の操作は，同一のテストデータ設定個所ではなく，別のテストデータ設定個所で設定してください。

- アドレス項目の値をアドレス名への代入
- アドレスデータ項目の値をアドレス名へ代入
- アドレス名によって参照されるデータ項目へ値の代入

例えば，［プログラム開始時］画面でアドレスデータ項目にほかのデータ項目のアドレス，または @'ALLOCATE'で確保した領域のアドレスを設定し，［中断点］画面でアドレスデータ項目の値をアドレス名への代入，およびアドレス名によって参照されるデータ項目へ値の代入を設定してください。

- 期待値は，判定が必要な項目にだけ設定してください。期待値の設定がない場合は，実行結果の自動判定は未判定になります。実行結果の自動判定については「4.10　単体テストを実行したときの判定結果」を参照してください。
- 期待値に対して，自動判定対象のデータ項目と比較できない値を入力した場合には，［メッセージ］画面にエラーメッセージが表示されます。

- -EquivRule コンパイラオプションを指定した場合，期待値に設定する値やデータ項目も-EquivRule コンパイラオプションで指定する規則にあわせて指定する必要があります。ただし，-EquivRule,StdCode コンパイラオプションを指定した場合は，指定していない場合と同様の扱いになります。

関連項目

- 「10.8.2 ［プログラム開始時］画面」
- 「10.8.5 ［中断点］画面」

4.4.8 値の入力規則

代入値と期待値に指定する値には，データ名および COBOL の定数，テストデバッガで定義する拡張 16 進定数を指定できます。さらに，代入値には次の値も指定できます。

- データ項目のアドレス値
- 指定されたサイズで確保された領域のアドレス値
- アドレス名によって参照されるデータ項目と同じサイズで確保された領域のアドレス値

これらの値を指定するときは，単体テスト支援で固有の形式で入力してください。

表 4-1 代入値に指定する値の形式

入力先のデータ項目	指定する代入値	入力形式
アドレスデータ項目/ポインタ項目	データ項目のアドレス値	&'*データ項目名*' または&"*データ項目名*"※1
	指定されたサイズの領域を確保したアドレス値 ※2	@'ALLOCATE,*領域サイズ*（バイト）' または@"ALLOCATE,*領域サイズ*（バイト）" ※1 指定できる領域サイズは 1～2147483647 ※3
アドレス名	アドレス名によって参照されるデータ項目と同じサイズの領域を確保したアドレス値※2	@'ALLOCATE' または @"ALLOCATE" ※1

注※1

データ項目名以外は，基本英文字，基本数字，基本特殊文字で指定してください。ALLOCATE の英大文字と英小文字は区別しません。データ項目名は，等価規則が適用されます。

基本英文字，基本数字，基本特殊文字については，マニュアル「COBOL2002 言語 標準仕様編」の COBOL 文字集合の説明を参照してください。

注※2

確保した領域は，そのテストケースの実行が終了したときに解放されます。

注※3

領域サイズを指定する例を示します。

（例）

次に示すデータ項目の定義で，アドレスデータ項目 ARGV-1 に，確保した 120 バイトの領域のアドレスを設定するときは，「@'ALLOCATE,120'」と指定します。

01 ARGV.
  02 ARGV-1 ADDRESS.

複数のデータ項目が関連づいている場合の構造と設定

集団項目など，複数のデータ項目が関連づいてデータ項目を定義している場合の，データ項目の表示と設定方法について説明します。

- 集団項目

  集団項目は，各データ項目に設定されているレベル番号に合わせて，親子関係を持つ構造で扱います。例を次に示します。

  COBOL ソーステキスト

```
01 AAA.
  02 AAA-SUB PIC X.
```

  表示例

```
01  AAA
  02  AAA-SUB
```

  集団項目に対して直接値を設定できません。

- REDEFINES

  再定義するデータ項目の構造の例を次に示します。再定義するデータ項目のデータ名の後ろには，再定義であることを示す「<REDEFINES>」の文字列が表示されます。

  COBOL ソーステキスト

```
01 BASEAREA.
  02 BUFFER PIC X(10).
01 WORK REDEFINES BASEAREA.
  02 N PIC S9(5).
  02 S PIC X(5).
```

  表示例

```
01  BASEAREA
  02  BUFFER
01  WORK <REDEFINES>
  02  N
  02  S
```

- アドレス名によって参照されるデータ項目およびアドレス名

  アドレス名によって参照されるデータ項目の構造の例を次に示します。

  COBOL ソーステキスト

```
01 DATA-1 ADDRESSED BY ADDRESS_NAME.
  02  BUFFER PIC X(10).
```

表示例

```
01  DATA-1
      ADDRESS_NAME <ADDRESSED>
  02 BUFFER
```

- OCCURS/指標名

  指標名を含むデータ項目の構造の例を次に示します。指標名の名前の後ろには，指標名であることを示す「<INDEXED>」の文字列が表示されます。

  COBOL ソーステキスト

```
01 DATA-1.
  05 ELEM PIC 9(5) OCCURS 2 INDEXED BY IDX.
```

  表示例

```
01  DATA-1
    05  ELEM
          IDX <INDEXED>
    05  ELEM(1)
    05  ELEM(2)
```

  また，可変長項目の指標名を含むデータ項目の構造の例を次に示します。制御変数の名前の後ろには，可変長項目の制御変数であることを示す「<DEPENDING>」の文字列が表示されます。

  COBOL ソーステキスト

```
01 DATA-1.
    05 ELEM PIC 9(5) OCCURS 2 DEPENDING ON DEPNUM INDEXED BY IDX.
01 DEPNUM PIC S9(4).
```

  表示例

```
01  DATA-1
    05  ELEM
          IDX <INDEXED>
          DEPNUM <DEPENDING>
    05  ELEM(1)
    05  ELEM(2)
01  DEPNUM
```

  制御変数は，データ項目の表示部分には複数個所に表示されます。ただし，制御変数に対する値の設定（値の代入，値の表示，期待値）については，同じ制御変数のデータ項目については共通になります。

  どこか 1 か所で値の設定を変更した場合，その制御変数を表示しているすべての個所の表示が変わります。上記の例では，「DEPNUM <DEPENDING>」の値の代入の設定を変更した場合，「01 DEPNUM」の値の代入の設定についても連動して変更されます。

- TYPE/TYPEDEF

  型宣言を含むデータ項目の構造の例を次に示します。

  COBOL ソーステキスト

```
01 DATATYPE TYPEDEF.
  02 BUFFER PIC X(10).
01 DATAREF TYPE DATATYPE.
```

表示例

```
01  DATAREF
  02  BUFFER
```

- SAME AS

  SAME AS を含むデータ項目の構造の例を次に示します。

  COBOL ソーステキスト

```
01 DATATYPE.
  02 BUFFER PIC X(10).
01 DATAREFTYPE SAME AS DATATYPE.
```

表示例

```
01  DATATYPE
  02  BUFFER
01  DATAREFTYPE
  02  BUFFER
```

- GLOBAL 指定のデータ項目

  GLOBAL 指定のデータ項目の例を次に示します。データ項目のデータ名の後ろには，GLOBAL 指定であることを示す「<GLOBAL>」の文字列が表示されます。

  COBOL ソーステキスト

```
01 AAA IS GLOBAL.
  02 AAA-SUB PIC X.
```

表示例

```
01  AAA <GLOBAL>
  02  AAA-SUB <GLOBAL>
```

- RENAMES

  RENAMES 句で定義されたデータ項目は，それを含む最上位の集団項目から 1 段下がった場所に表示されます。

  COBOL ソーステキスト

```
01 AAA.
  03 BBB PIC X(1).
  66 RN RENAMES BBB.
```

表示例

```
01 AAA
  03 BBB
  66 RN
```

- LINAGE-COUNTER

LINAGE-COUNTER は，特殊レジスタが結びついているデータ名（ファイル名）を「(xxx)」の形で表示します。

表示例

```
LINAGE-COUNTER(FD-NAME)
```

注意

- 次に該当する，値の代入と期待値の設定をしたテストケースは，単体テストが実行されないで，判定結果が「エラー」となります。［メッセージ］画面のメッセージを確認して，設定を変更してください。
  - 指定したデータ名が存在しない，または参照できない。
  - 指定したデータ名のデータ項目が，対象のデータ項目に対して代入または比較ができない。
  - 指定した定数が，対象のデータ項目に対して代入または比較ができない。
  - 設定したデータ項目のアドレス値または領域を確保したアドレス値が，単体テスト支援固有の形式に合っていない。
  - 設定したデータ項目のアドレス値または領域を確保したアドレス値が，対象のデータ項目に対して代入または比較ができない。
- 次に示すデータ項目は，値の代入および値の表示/期待値のデータ名の一覧に表示されません。
  - オブジェクト参照データ項目
  - OLE オブジェクト参照データ項目
  - バリアントデータ項目
- 編集項目に対してテストデータを設定する場合は，次のことに気を付けてください。
  - 値の代入には，編集項目に入力する値を設定してください。

    （例）

    対象とするデータ項目の PICTURE 句が「¥¥.¥」のとき，値 2.5 を代入する場合は，値の代入として「2.5」を設定します。
  - 期待値には，編集項目を表示したときの文字列と比較する文字列を設定します。

    （例）

    対象とするデータ項目の PICTURE 句が「¥¥.¥」のとき，値 2.5 であるということを確認する場合は，期待値として「"¥2.5"」を設定します。
- ［中断点］画面で，大域名（GLOBAL 句が指定されたデータ項目）が中断点から参照できないとき，［値設定］［値の表示］［期待値］は指定できません。

4.5　テストプロジェクトの環境を設定する

テストプロジェクトの環境は，次の画面で確認または設定します。

- ［プロパティページ］
- ［スタブ設定ページ］

注意

- ENTRY 文の個所から単体テストを実行する場合は，［プロパティページ］の［テスト対象プログラムの種類］ で「副プログラム」を選択してください。
- ［プロパティページ］で指定できるコンパイラオプションは，ccbl2002 コマンドのコンパイラオプションだけです。ファイル名は指定しないでください。
- ［プロパティページ］には，コンパイラ環境変数 CBLPIDIR は指定しないでください。
- ［プロパティページ］の［変更］ボタンで表示される［COBOL ソースファイルの設定］ダイアログの指定は，「COBOL ソースファイルの解析」と「テスト実行」で有効になります。また，指定したコンパイラオプションに対して，「5.2　COBOL ソースファイルを解析する」の注意で示すコンパイラオプションが追加されます。
- ［COBOL ソースファイルの解析］で実行するのはコンパイルだけで，リンクは実行されません。［テスト実行］メニューはコンパイル・リンク・実行までが実行されます。［プロパティページ］の［実行環境］の指定は，「テスト実行」で有効になります。
- ［プロパティページ］の実行環境ファイルと実行時環境変数の両方を指定して，同じ環境変数に異なる値を設定した場合の優先順位を次に示します。
  1. テストプロジェクトのプロパティで設定した実行環境ファイル
  2. 共通実行環境ファイル
  3. テストプロジェクトのプロパティで設定した実行時環境変数
  4. 単体テスト支援起動時に有効になっているコマンドプロンプトやコントロールパネルで設定した環境変数
- ［COBOL ソースファイルの設定］ダイアログで［COBOL ソースファイルの解析］ボタンをクリックして，解析の結果としてエラーが発生すると，［スタブ設定ページ］，［テストケースページ］には入力できなくなります。メッセージ画面でエラーメッセージを確認して，次に示す操作でエラーを解消すると，設定できるようになります。
  - COBOL ソースファイルの内容に問題があるときは，［ソーステキスト］メニューの［COBOL エディタを開く］を選択して，COBOL エディタを開きます。ソーステキストを修正して［プロジェクト］メニューの［COBOL ソースファイルの解析］を選択します。
  - コンパイラオプションまたはコンパイラの環境変数に問題があるときは，［プロジェクト］－［プロパティページを表示］メニューを選択して，［プロパティページ］を開きます。［プロパティページ］の［変更］ボタンをクリックして，［COBOL ソースファイルの設定］ダイアログを開いて修正したあと，［COBOL ソースファイルの解析］ボタンをクリックします。

- 同一の COBOL ソースファイル内に，CALL 定数で呼び出すプログラムが定義されている場合は，［スタブ設定ページ］の「CALL 定数で呼び出す副プログラム名」に，そのプログラム名は表示されません。
- ［テスト実行］での結果の確認で，COBOL2002 の実行時エラーメッセージを確認するには，［プロパティページ］の実行時環境変数，または実行環境ファイルに，実行時環境変数 CBL_SYSERR と実行時環境変数 CBLABNLST を指定してください。

  テストケースごとに異なるファイルに出力したい場合，［テスト環境の設定］画面で実行時環境変数 CBL_SYSERR と実行時環境変数 CBLABNLST を指定してください。［テスト環境の設定］画面は，［プロパティページ］で「テストケースごとに実行環境ファイルまたは実行時環境変数を設定する」チェックボックスをオンにすると表示されます。
- 手続き文中に USING 指定の SORT 文または MERGE 文がある場合は，［ファイルシミュレーション］画面で，USING 指定に指定したファイルの READ 文に対して，終了条件で「AT END 指定の文を実行」を指定してください。または，USING 指定に指定したファイルに対する［スタブ設定ページ］に表示されるファイル名のチェックボックスをオフにして，［プロパティページ］の実行時環境変数または実行環境ファイルにファイルの実体を指定してください。

  終了条件または実体を指定しないと，実行時エラーになるか，テスト実行が終了しなくなります。
- 手続き文中の ACCEPT 文への入力または DISPLAY 文，定数指定の STOP 文による出力内容を確認する場合，それぞれ［プロパティページ］の実行時環境変数または実行環境ファイルに必要な環境変数を指定してください。ACCEPT 文，DISPLAY 文，STOP 文に対する注意を次に示します。
  - CUI モードで ACCEPT 文を実行する場合は，実行時環境変数 CBL_SYSIN または CBL_SYSTD にファイル名を指定してください。実行時環境変数にファイル名を指定しないと，制御が戻りません。
  - UPON CONSOLE 指定以外の DISPLAY 文の出力内容を確認する場合は，実行時環境変数 CBL_SYSOUT または CBL_SYSPUNCH にファイル名を指定してください。実行時環境変数にファイル名を指定しないと，出力内容を確認できません。
  - 定数指定の STOP 文の出力内容を確認する場合は，GUI モードで実行してください。CUI モードで実行した場合，出力内容を確認できません。
  - 手続き中に UPON CONSOLE 指定の DISPLAY 文がある場合は，出力内容を確認できません。確認する場合は CONSOLE 指定以外の指定に変更してから確認してください。

  GUI モード，CUI モード，ACCEPT 文，DISPLAY 文，および定数指定の STOP 文については，マニュアル「COBOL2002 ユーザーズガイド」を参照してください。
- ［プロパティページ］の実行時環境変数に，出力ファイルを指定する実行時環境変数を追加モード（ファイル名の末尾に「+」を付ける形式）で指定した場合，テストを実行したテストケースすべての出力内容が，実行時環境変数に指定したファイルに追加で出力されます。追加モードを指定しない場合は，最後に実行したテストケースの出力内容がファイルに出力されます。
- 次の個所で設定した環境変数の合計値が 65,535 バイトを超えると，［COBOL ソースファイルの設定］ダイアログで［COBOL ソースファイルの解析］ボタンをクリックしたときに論理エラーになります。回避するには，不要な環境変数を削除してください。
  - ［新しくテストプロジェクトを作る］ダイアログ，［プロパティページ］または［COBOL ソースファイルの設定］ダイアログで設定したコンパイラ環境変数

- ［プロパティページ］で設定した実行時環境変数
- 開発マネージャで設定したコンパイラ環境変数（開発マネージャから起動した場合）
- システムに設定した環境変数

- stdcall 呼び出し規約で呼び出すプログラムのダミーの副プログラムは作成されません。
- COBOL ソースファイル内に，stdcall 呼び出し規約のプログラムを呼び出す CALL 文があっても，そのプログラム名は［スタブ設定ページ］の一覧には表示されません。
- ダミーファイルに対する入出力文では，［ファイルシミュレーション］画面で終了条件を設定しないと，ファイル終了条件などの入出力条件が発生しません。そのため，入出力条件が発生しないと次の処理に進まないようなプログラムをテスト対象とした場合，テスト実行が終了しません。

  例 1：

```
開始.
  READ INPUT-FILE AT END GO TO 終了.
  GO TO 開始.
終了.
  CLOSE INPUT-FILE.
```

  終了条件が発生しないので，AT END 側の手続きの「GO TO 終了」が実行されず，無限ループします。この場合は，入出力対象のファイルに対して［スタブ設定ページ］に表示されるファイル名のチェックボックスをオフにして，［プロパティページ］の実行時環境変数または実行環境ファイルにファイルの実体を指定してください。

  例 2：

```
開始.
  READ INPUT-FILE AT END MOVE 0 TO 終了フラグ.
  IF 終了フラグ = 0
   THEN
   GO TO 終了
   ELSE
   GO TO 開始
  END-IF.
終了.
  CLOSE INPUT-FILE.
```

  終了条件が発生しないので，AT END 側の手続きの「MOVE 0 TO 終了フラグ.」が実行されないで，無限ループします。この場合は，IF 文に中断点を設定して，［テストケースページ］の［設定］タブの［中断点］画面でフラグに値を設定するか，例 1 と同様にファイルの実体を指定してください。

- ［プロパティページ］で［テストケースごとに実行環境ファイルまたは実行時環境変数を設定する］チェックボックスをオンにすると，テスト実行時間が増加します。必要のない場合はチェックボックスをオフにしてください。
- ファイルシミュレーションの設定は，［スタブ設定ページ］で「入出力するファイル名」欄のチェックボックスをオフにして，プロジェクトを上書き保存すると削除されます。チェックボックスをオフにする前に，設定内容を控えることをお勧めします。ただし，チェックボックスをオフにしても，プロジェクトを上書き保存する前に，再度チェックボックスをオンにすると，前回設定していた内容のまま，ファイルシミュレーションの設定が表示されます。

- テストケースごとに設定した実行環境ファイルまたは実行時環境変数の内容は，［プロパティページ］で［テストケースごとに実行環境ファイルまたは実行時環境変数を設定する］チェックボックスをオフにして，プロジェクトを上書き保存すると削除されます。チェックボックスをオフにする前に，設定内容を控えることをお勧めします。ただし，チェックボックスをオフにしても，プロジェクトを上書き保存する前に，再度チェックボックスをオンにすると，前回設定していた内容のまま，テストケースごとに設定した実行環境ファイルまたは実行時環境変数の内容が表示されます。
- ［スタブ設定ページ］の「入出力するファイル名」では，対象のファイルに EXTERNAL 指定がある場合は，どれか 1 つのチェックをオンにすると，ほかの EXTERNAL 付きのファイルは非活性になります。このとき，チェックはオフのまま非活性になります。
- 03-04，03-04-/A でテストプロジェクトを開いたとき，そのテストプロジェクトファイルに EXTERNAL 指定のファイルがあり，［スタブ設定ページ］で同名の EXTERNAL 指定のファイルが複数チェックオンであった場合の動作は次のようになります。

テストプロジェクトを作成したバージョン	03-04 で開く	03-04-/A 以降で開く
03-04 未満	• 複数チェックオンのまま開きます • 活性/非活性の制御は行いません	• 複数チェックオンのまま開きます※ • すべてのファイルについて，同名のファイルでチェックがオンのファイルが 1 つだけの状態になったら，活性/非活性の制御を行います→それ以降は複数チェックオンの状態にはなりません
03-04		

注※

KEDL1033T-W を出力します。同名の EXTERNAL のうちどれか 1 つのファイルのチェックをオンにしてください。どれか 1 つを選択しない（同名の他ファイルのチェックをオフにしない）ままテスト実行またはテストプロジェクトの保存をしようとした場合，KEDL0063T-E を出力して処理を中止します。

また，COBOL ソースファイルの解析時に複数チェックオンの状態である場合（複数チェックオンの状態を解消しないまま解析を行った場合）にも KEDL1033T-W を出力します。

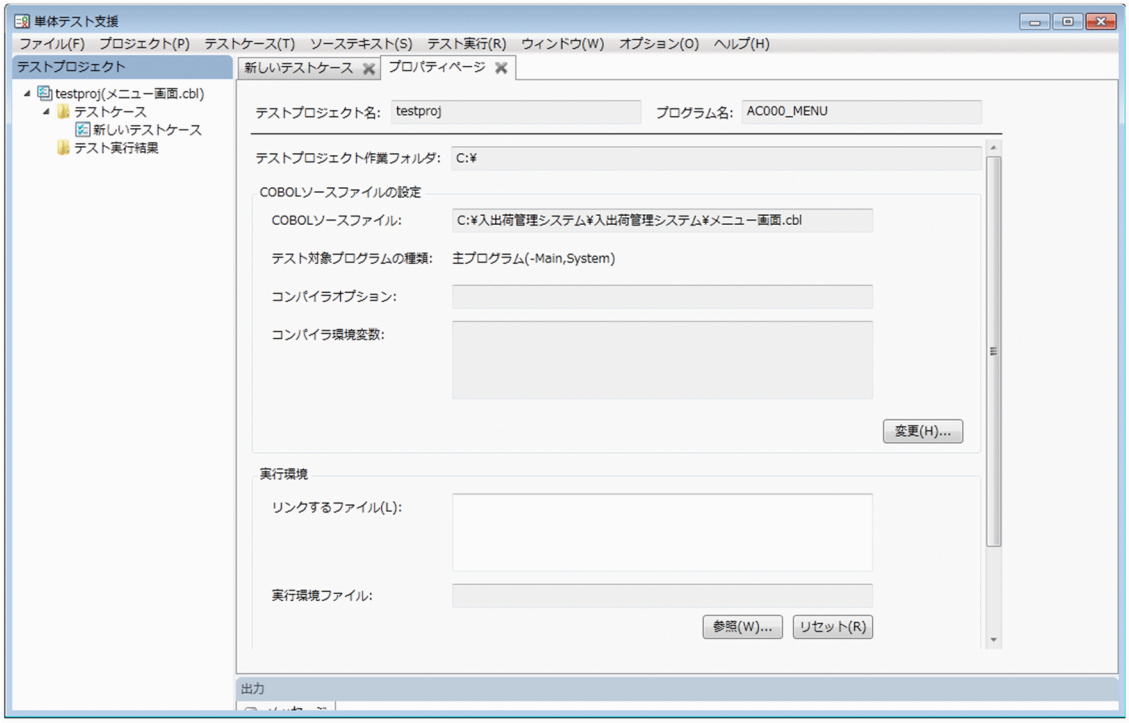
4.5.1 テストプロジェクトのプロパティを表示する

テストプロジェクトのプロパティを表示する手順です。

作業の手順

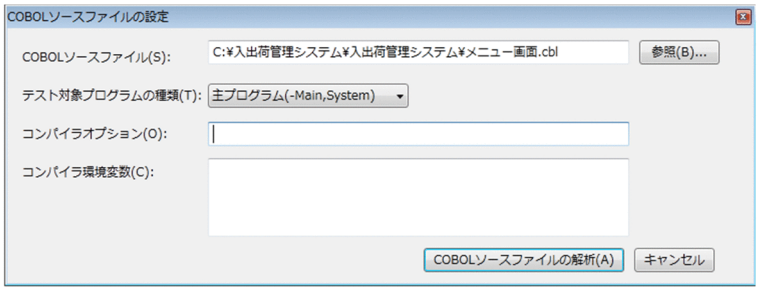
1. ［プロジェクト］－［プロパティページを表示］メニューを選択します。

   ［プロパティページ］が表示されます。



［プロパティページ］の「COBOL ソースファイルの設定」を変更するときは，［変更］ボタンをクリックします。

［COBOL ソースファイルの設定］ダイアログが表示されます。



変更できる項目を次に示します。変更しない場合は［キャンセル］ボタンをクリックし，ダイアログを閉じます。

- COBOL ソースファイル
- テスト対象プログラムの種類
- コンパイラオプション
- コンパイラ環境変数

COBOL ソースファイルについては，パス名は変更できますが，ファイル名は変更できません。

2. ［プロパティページ］の下にある実行環境も修正します。

   指定する項目については，「10.6 ［プロパティページ］」を参照してください。

3. ［プロパティページ］の内容を変更したら，［COBOL ソースファイルの解析］ボタンをクリックします。

「COBOL ソースファイルの解析」が実行され，新しい設定に合わせて，次に示す画面の情報が更新されます。

- ［スタブ設定ページ］
- ［テストケースページ］

関連項目

- 「4.5.3　テストケースごとに実行環境ファイルと実行時環境変数を設定する」
- 「10.6　［プロパティページ］」

4.5.2　テストプロジェクトのスタブを編集する

単体テストを実行するために使用する，スタブ（ダミーの副プログラムまたはファイル）を編集する手順です。

単体テスト支援では，テスト対象プログラムの CALL 文または入出力文に対し，ダミーの副プログラムやファイルを作成します。単体テストは次のように実行されます。

- 呼び出す副プログラムがなくても，定数指定の CALL 文，プログラム名が代入された一意名指定の CALL 文を実行します。
- ファイルの実体がなくても，ファイル入出力文を実行します。

［スタブ設定ページ］は，単体テスト支援が設定したダミーのプログラムやファイルをカスタマイズするときに使用します。

作業の手順

1. ［プロジェクト］－［スタブ設定ページを表示］メニューを選択します。
   ［スタブ設定ページ］が表示されます。

2. ［スタブ設定ページ］の，次に示す項目を設定します。

設定すると，ダミーの副プログラムまたはファイルが生成されます。ダミーの CALL 文または入出力文の前後に中断点を設定すると，引数またはレコードなどに対する値の設定や表示もできます。

- CALL 定数で呼び出す副プログラム名

  ダミーの副プログラムを作成する場合はチェックボックスをオンにします。呼び出す副プログラムの実体がある場合はオフにします。

- CALL 一意名で呼び出す副プログラム名

  ダミーの副プログラムを作成する場合はプログラム名を指定します。プログラム名は，1 行に 1 つの名前を入力します。

  呼び出す副プログラムの実体がある場合はプログラム名を指定しません。

- 入出力するファイル名

  ダミーのファイルを作成する場合はチェックボックスをオンにします。ファイルの実体がある場合はオフにします。

関連項目

- 「10.7　［スタブ設定ページ］」

4.5.3　テストケースごとに実行環境ファイルと実行時環境変数を設定する

テストケースごとに実行環境ファイルと実行時環境変数を設定する手順です。

作業の前に確認すること

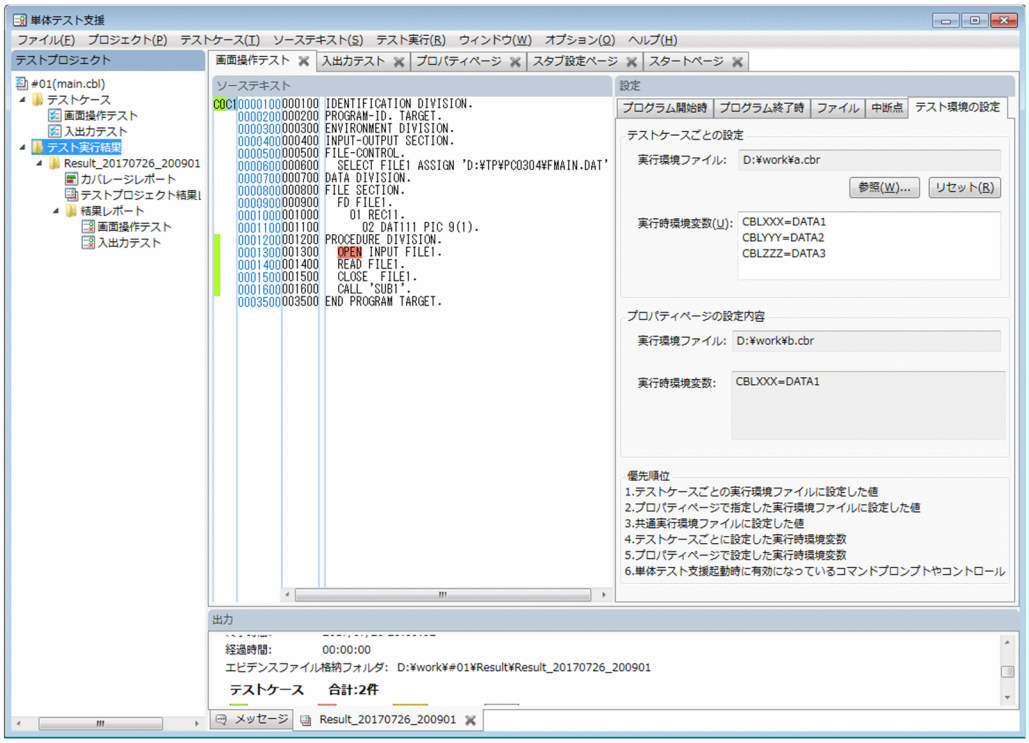
- ［プロパティページ］で［テストケースごとに実行環境ファイルまたは実行時環境変数を設定する］チェックボックスをオンにしている。

作業の手順

1. 設定したいテストケースを表示します。

2. ［テスト設定］画面の［テスト環境の設定］画面を選択します。

   ［テスト環境の設定］画面が表示されます。



3. テストケースごとに実行環境ファイルを指定する場合は，［テストケースごとの設定］の［参照］ボタンから実行環境ファイルを選択します。

   ［リセット］ボタンをクリックすると，実行環境ファイルの指定が解除されます。

4. テストケースごとに実行時環境変数を指定する場合は，［テストケースごとの設定］の［実行時環境変数］に直接設定します。

注意

実行時環境変数の優先順位は次のとおりです。

1. テストプロジェクトの［テスト環境の設定］画面で設定した実行環境ファイル※
2. テストプロジェクトの［プロパティページ］で設定した実行環境ファイル
3. 共通実行環境ファイル
4. テストプロジェクトの［テスト環境の設定］画面で設定した実行時環境変数※
5. テストプロジェクトの［プロパティページ］で設定した実行時環境変数
6. 単体テスト支援起動時に有効になっているコマンドプロンプトやコントロールパネルで設定した環境変数

注※

［プロパティページ］の［テストケースごとに実行環境ファイルまたは実行時環境変数を設定する］チェックボックスをオンにした場合，設定できます。

優先順位 1 の実行環境ファイルを設定した場合，優先順位 2 の実行環境ファイルは使用されません。

テスト実行時の実行環境ファイルの有無は優先順位ごとに確認されます。優先順位 1 の実行環境ファイルを設定した場合，優先順位 2 の実行環境ファイルの有無は確認されません。優先順位 2 の実行環境ファイルが存在しなくてもエラーになりません。

優先順位ごとの項目に設定した環境変数と，テストケースで設定される値の例を次に示します。

項目	優先順位	環境変数の設定内容			
		すべての項目で環境変数を設定		［テスト環境の設定］画面の実行環境ファイルは設定したが，環境変数の指定がない。	
［テスト環境の設定］画面で設定した実行環境ファイル	1	ENV1=A	ENV1=A	• 実行環境ファイルは設定済み • 環境変数の指定がない	
［テスト環境の設定］画面で設定した実行時環境変数	4	ENV1=B	ENV2=B	ENV1=B	ENV2=B
共通実行環境ファイル	3	ENV1=C	ENV3=C	ENV1=C	ENV3=C
［プロパティページ］で設定した実行環境ファイル	2	ENV1=D	ENV4=D	ENV1=D	ENV4=D
［プロパティページ］で設定した実行時環境変数	5	ENV1=E	ENV5=E	ENV1=E	ENV5=E
単体テスト支援起動時に有効になっているコマンドプロンプトやコントロールパネルで設定した環境変数	6	ENV1=F	ENV6=F	ENV1=F	ENV6=F
テストケースで設定される値		ENV1=A	ENV1=A ENV2=B ENV3=C ENV5=E ENV6=F	ENV1=C	ENV2=B ENV3=C ENV5=E ENV6=F

すべての項目で環境変数を設定した場合

- 各項目で ENV1 に異なる値を設定すると，優先順位 1 の ENV1=A がテストケースで設定されます。
- 各項目で異なる環境変数を設定すると，優先順位 1 の実行環境ファイルが設定されているので，優先順位 2 の値は設定されません。優先順位 2 以外の値はテストケースで設定されます。

［テスト環境の設定］画面の実行環境ファイルは設定したが，環境変数の指定がない場合

優先順位 1 には環境変数の指定がないため，優先順位 1 は設定されません。また，優先順位 1 の実行環境ファイルは設定されているので，優先順位 2 は設定されません。

- 各項目で ENV1 に異なる値を設定すると，優先順位 3 の ENV1=C がテストケースで設定されます。
- 各項目で異なる環境変数を設定すると，優先順位 1 と優先順位 2 以外の値はテストケースで設定されます。

テストケースで設定される値の次の情報は，［テストケース結果レポート］画面に，バッチモードの場合は「結果詳細」に出力されます。

- 実行環境ファイルの絶対パス
- ［テスト環境の設定］画面と［プロパティページ］で指定した実行時環境変数をマージした値

関連項目

- 「10.6　［プロパティページ］」
- 「10.8.6　［テスト環境の設定］画面」

4.6　単体テストを実行する

単体テスト支援で，単体テストを実行する手順です。

単体テストでコンパイルエラーになったときは，単体テストは実行されません。このとき，［スタブ設定ページ］と［テストケースページ］は操作できません。

COBOL ソーステキスト，コンパイラオプション，またはコンパイラ環境変数を変更したときは，単体テストを実行する前に，［プログラム開始時］画面，［プログラム終了時］画面，または［中断点］画面の設定内容を確認してください。不当なテスト結果が出力されることを防ぐため，次の場合はテストデータの設定を見直してください。

- COBOL ソーステキストを修正したことで，中断点を設定していた文の文番号がずれた。
- 値設定や期待値を設定したデータ項目が削除または変更されたため，テストデータの設定がなくなった。

単体テストの実行には，次のパターンがあります。

- すべてのテストケースを実行
  テストプロジェクトに存在するすべての有効なテストケースを一括で実行する。
- 1 つのテストケースを実行
  テストプロジェクトに存在する有効なテストケースのうち，任意の 1 つを実行する。

注意

- 次の個所で設定した環境変数の合計値が 65,535 バイトを超えると，すべてまたは 1 つのテストケースを実行したときに論理エラーになります。回避するには，不要な環境変数を削除してください。
  - ［新しくテストプロジェクトを作る］ダイアログ，［プロパティページ］または［COBOL ソースファイルの設定］ダイアログで設定したコンパイラ環境変数
  - ［プロパティページ］で設定した実行時環境変数
  - 開発マネージャで設定したコンパイラ環境変数（開発マネージャから起動した場合）
  - システムに設定した環境変数
- テストプロジェクトに含まれる実行結果の個数や実行結果で表示するデータ項目の個数の総数によってはメモリ不足となりテストプロジェクトファイルの保存に失敗するか，異常終了する場合があります。個々の上限値や総数の目安は「付録 C　単体テスト支援の制限値と限界値」を参照してください。
- -StdCall コンパイラオプションを指定しても COBOL ソースファイルと同じフォルダに置いた stdcall 呼び出し指示ファイルは有効になりません。有効にするには，-StdCallFile コンパイラオプションで stdcall 呼び出し指示ファイルを絶対パスで指定してください。指定しない場合は，cdecl 呼び出し規約のプログラムとしてスタブ生成の対象になります。
- テスト実行時に，対応が取れないテストデータが見つかった場合は，そのテストデータが未保存の状態で削除され，テスト実行が中止されます。

このとき，テストデータを見直して，問題がなければ再度テスト実行できます。問題がある場合は，テストデータを再設定してからテスト実行するか，または，テストプロジェクトを保存しないで開き直すことで，対応が取れなくなったテストデータを元に戻すことができます。

- テストプロジェクト名に「-」（ハイフン）で始まる名前を指定しないでください。指定した場合，テスト実行時にエラーになります。エラーになった場合，テストプロジェクトを閉じ，テストプロジェクトファイルのファイル名を「-」（ハイフン）で始まらない名前に変更してからテストプロジェクトを開き直してください。

4.6.1　すべてのテストケースを実行する

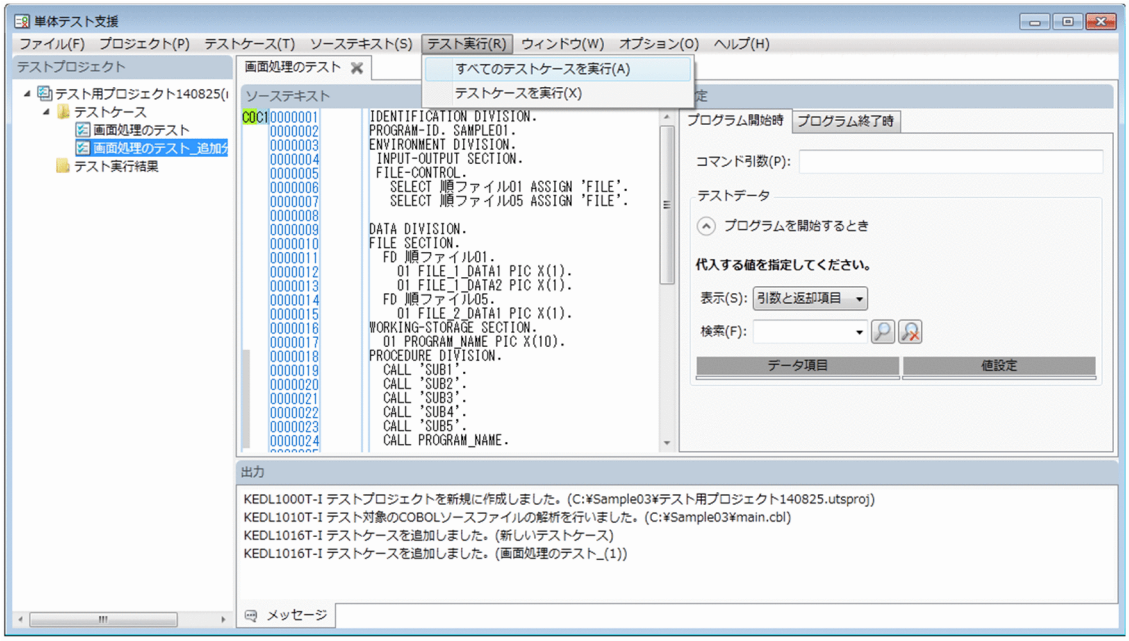
テストプロジェクト内のすべてのテストケースを実行する手順です。

作業の前に確認すること

- テストケースを作成し，テストデータが設定してある。
- COBOL ソースファイルを更新した場合，「COBOL ソースファイルの解析」を実行済みである。

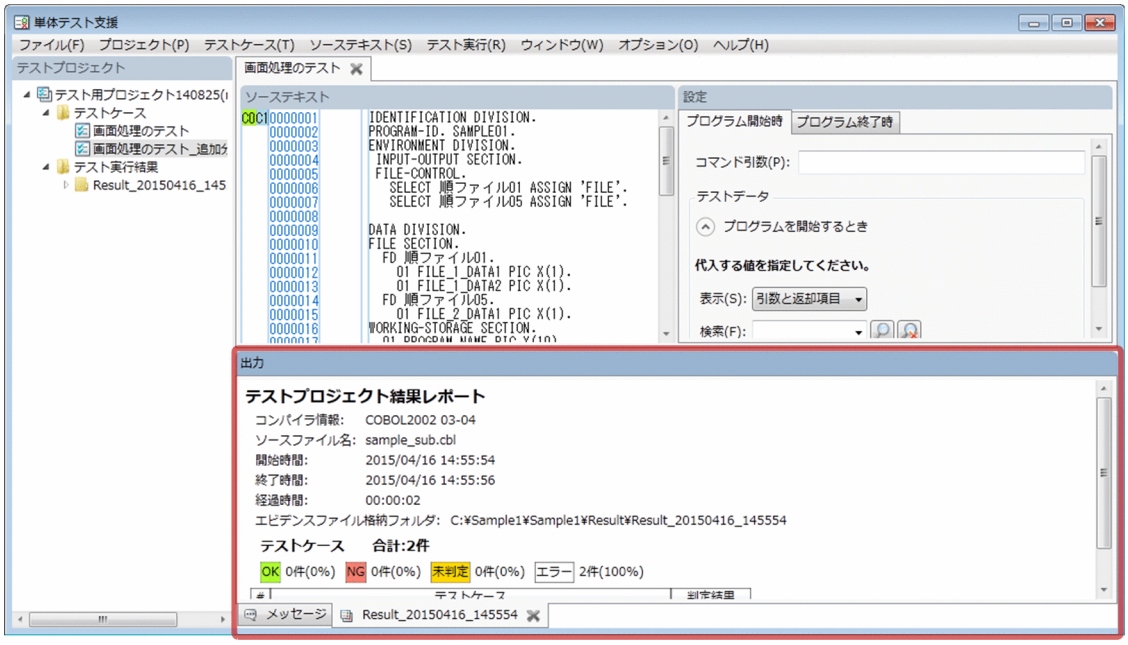
作業の手順

1. ［テスト実行］－［すべてのテストケースを実行］メニューを選択します。



テストプロジェクトの保存が必要な場合は，単体テストの実行を確認するダイアログが表示されます。［OK］ボタンをクリックすると，テストプロジェクトが保存され，すべてのテストケースが実行されます。

テストケースの実行が完了すると，［テストプロジェクト］画面の［テスト実行結果］に，テスト結果が「Result_*yyyymmdd*_*hhmmss*」（*yyyy*：西暦年，*mm*：月，*dd*：日，*hh*：時，*mm*：分，*ss*：秒）で追加され，［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。



注意

- すべてのテストケースを実行する操作をしても，無効にしたテストケースは実行されません。
- テストプロジェクトに含まれる実行結果の個数や実行結果で表示するデータ項目の個数の総数が多過ぎると，メモリ不足となりテストプロジェクトファイルの保存に失敗するか，異常終了することがあります。個々の上限値や総数の目安は「付録 C　単体テスト支援の制限値と限界値」を参照してください。

次の作業

「6. テスト結果を参照する」

4.6.2　1 つのテストケースを実行する

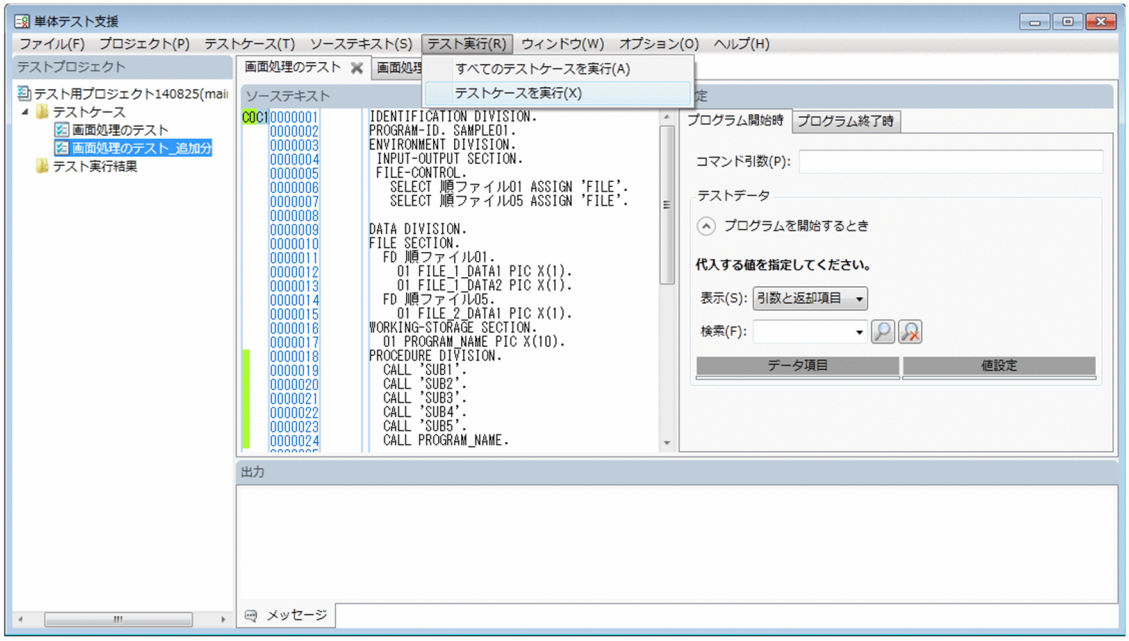
テストプロジェクト内の 1 つのテストケースを実行する手順です。

作業の前に確認すること

- テストケースを作成し，テストデータが設定してある。
- COBOL プログラムを更新した場合，「COBOL ソースファイルの解析」を実行済みである。

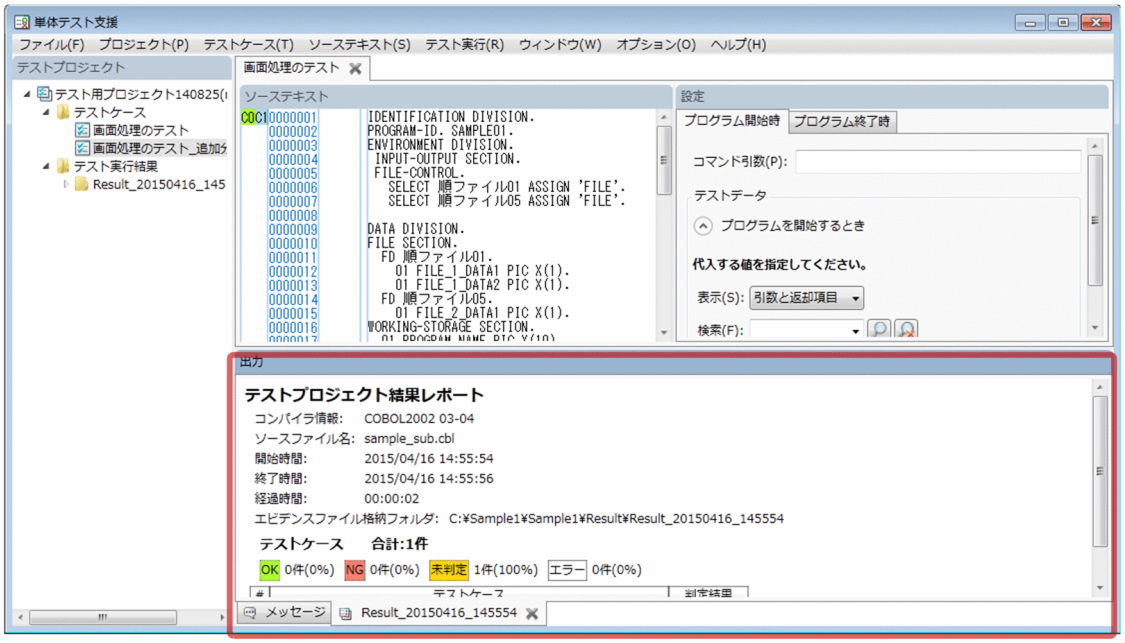
作業の手順

1. ［テストプロジェクト］画面で，単体テストをするテストケースを選択し，［テスト実行］－［テストケースを実行］メニューを選択します。



テストプロジェクトの保存が必要な場合は，単体テストの実行を確認するダイアログが表示されます。［OK］ボタンをクリックすると，テストプロジェクトが保存され，選択したテストケースが実行されます。

テストケースの実行が完了すると，［テストプロジェクト］画面の［テスト実行結果］に，テスト結果が「Result_*yyyymmdd*_*hhmmss*」（*yyyy*：西暦年，*mm*：月，*dd*：日，*hh*：時，*mm*：分，*ss*：秒）で追加され，［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。



次の作業

「6. テスト結果を参照する」

4.7　テストプロジェクトを保存する

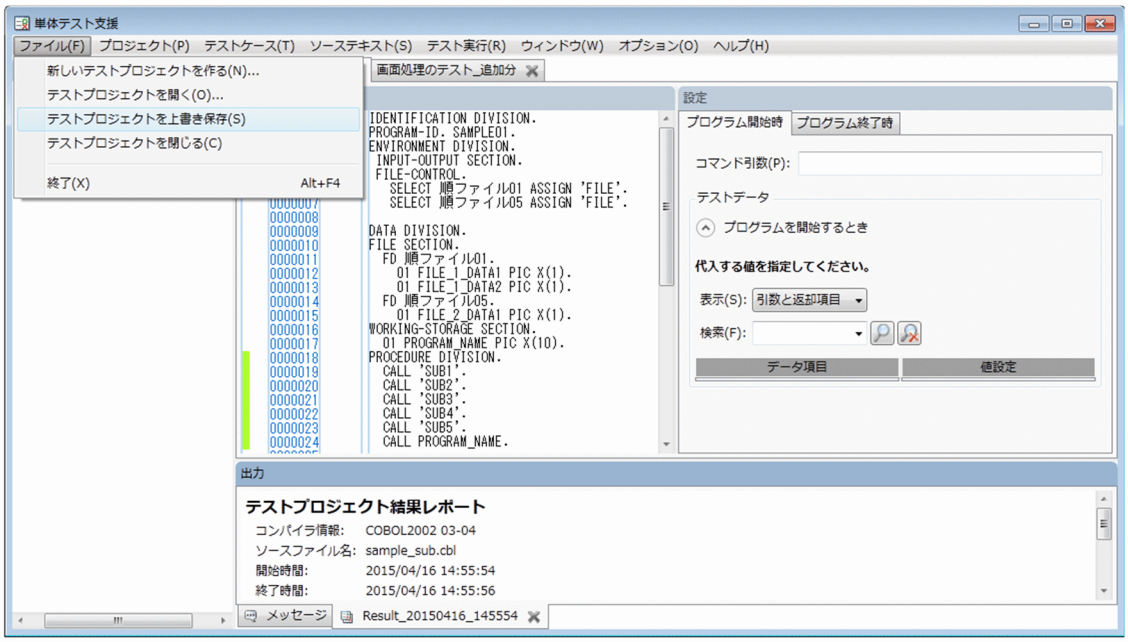
テストプロジェクトファイルに，テストプロジェクトの情報を保存する手順です。

なお，単体テストを実行すると，操作が完了した時点でテストプロジェクトは自動で上書き保存されます。

32bit 版の COBOL2002 Developer Professional で作成したテストプロジェクトファイルは，64bit 版の COBOL2002 Developer Professional では使用できません。また，64bit 版の COBOL2002 Developer Professional で作成したテストプロジェクトファイルは，32bit 版の COBOL2002 Developer Professional では使用できません。

作業の手順

1. ［ファイル］－［テストプロジェクトを上書き保存］メニューを選択します。



現在開いているテストプロジェクトが，テストプロジェクトファイルに保存されます。保存されるテストプロジェクトの設定は次のとおりです。

- ［プロパティページ］の設定内容
- ［スタブ設定ページ］の設定内容
- ［テストケースページ］の設定内容
- テスト実行結果

4.8　単体テスト支援を終了する

単体テスト支援を終了する手順です。

作業の手順

1. ［ファイル］－［終了］メニューを選択します。または単体テスト支援のメインウィンドウ右上の［×］ボタンをクリックします。

   単体テスト支援の終了を確認するダイアログが表示されます。

   ［OK］ボタンをクリックすると，単体テスト支援が終了します。

注意

- 単体テストの実行中は，単体テスト支援を終了できません。終了する場合は，単体テストの実行を中止するか，実行が完了したあとで終了させてください。

4.9　テストプロジェクトを開く

保存したテストプロジェクトを開く手順です。

保存したテストプロジェクトを開くには，次の方法があります。

- 単体テスト支援のメインウィンドウからテストプロジェクトを開く
- 開発マネージャからテストプロジェクトを開く
- Windows のエクスプローラーから単体テスト支援を開始する

32bit 版の COBOL2002 Developer Professional で作成したテストプロジェクトファイルは，64bit 版の COBOL2002 Developer Professional では使用できません。また，64bit 版の COBOL2002 Developer Professional で作成したテストプロジェクトファイルは，32bit 版の COBOL2002 Developer Professional では使用できません。

注意事項

ファイル名が「-」（ハイフン）で始まるテストプロジェクトファイルを指定しないでください。指定した場合，テスト実行時にエラーになります。エラーになった場合，テストプロジェクトを閉じ，テストプロジェクトファイルのファイル名を「-」（ハイフン）で始まらない名前に変更してからテストプロジェクトを開き直してください。

4.9.1　単体テスト支援のメインウィンドウからテストプロジェクトを開く

単体テスト支援のメインウィンドウからテストプロジェクトを開く手順です。

作業の手順

1. ［ファイル］－［テストプロジェクトを開く］メニューを選択します。
   ［テストプロジェクトを開く］ダイアログが表示されます。

2. ［テストプロジェクトを開く］ダイアログで，テストプロジェクトファイルを選択します。
   指定したテストプロジェクトファイルの内容が表示されます。

4.9.2　開発マネージャからテストプロジェクトを開く

開発マネージャからテストプロジェクトを開く手順です。

作業の手順

1. 開発マネージャのツリービューウィンドウの単体テスト支援プロジェクトファイルノードのテストプロジェクトファイルをクリックします。

   単体テスト支援が起動され，指定したテストプロジェクトファイルの内容が表示されます。

4.9.3 Windows のエクスプローラーから単体テスト支援を開始する

Windows のエクスプローラーで，作成済みのテストプロジェクトファイルをダブルクリックして開始する手順です。

作業の手順

1. エクスプローラーで，テストプロジェクトファイルを格納しているフォルダを開きます。

2. テストプロジェクトファイルをダブルクリックします。

   単体テスト支援が起動され，指定したテストプロジェクトファイルの内容が表示されます。

4.10　単体テストを実行したときの判定結果

単体テストを実行したときの判定結果についての説明です。

単体テスト支援には，判定結果として次の 2 種類の判定方法があります。単体テスト支援では，デフォルトでは自動判定による判定結果が表示されます。

- 自動判定
  テスト実行時に，データ項目の値と期待値を比較して自動で判定する方法です。
- 手動判定
  テスト結果を参照し，テストケースごとに判定結果を設定する方法です。

自動判定

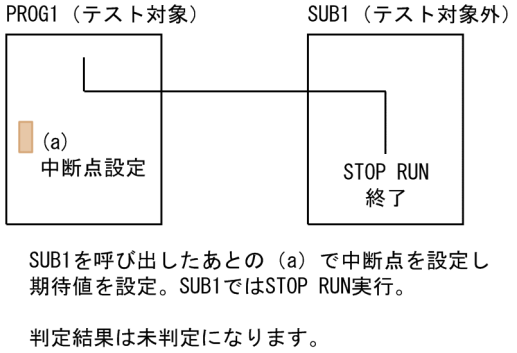
テストデータとして設定したデータ項目ごとに，設定した期待値と実行結果の値を比較して，単体テストの実行時に判定します。テストケース単位の結果は，そのテストケースに含まれるすべてのテストデータの自動判定の結果を基に判定します。

表 4-2　結果の判定条件

判定条件	判定結果
期待値との比較結果が OK となるデータ項目を 1 つ以上含み，かつ期待値との比較結果が NG およびエラーになるデータ項目がない。	OK
期待値との比較結果が NG になるデータ項目を 1 つ以上含む。さらにエラーになるデータ項目もない。	NG
すべてのデータ項目で，期待値との比較結果が未判定となる。	未判定
単体テスト支援が生成する TD コマンドにエラーがあり，テストプログラムを実行できなかった。または，テスト実行中に COBOL2002 の実行時エラーが発生し，結果を得られなかった。	エラー

単体テストの実行が完了していれば，期待値の設定を 1 つ以上設定したテストケースは必ず OK か NG の判定になります。逆に，期待値の設定がない場合は未判定になります。

また，次の例のように中断点やプログラム終了時の期待値の判定が 1 つも実行されないでプログラムが終了した場合は，判定結果は未判定になります。



自動判定では，テストデータとして期待値を設定します。設定方法については，「4.4.2　プログラム終了時の値を設定する」または「4.4.5　中断点のテストデータを設定する」を参照してください。

判定結果がエラーになるのは，次のどちらかの要因でテストプログラムを実行できなかった場合です。

- テストデータとして設定した値の代入や期待値の指定が間違っている。
- COBOL2002 の実行時エラーが起こる。

この場合，［メッセージ］画面に KEDL0031T-W のエラーメッセージが出力されます。値の代入や期待値の指定が間違っている場合は，KEDL0031T-W のあとに，KCCC で始まる COBOL2002 のテストデバッガのメッセージが出力されます。テストデバッガのメッセージに従って，値の代入や期待値の指定を見直し，正しい指定に修正してください。

また，テストケースに値の代入や期待値の指定が間違って設定されているために該当するエラーが複数存在した場合，1 回の「テストケースの実行」ですべてのエラーが出力されない場合があります。この場合は，KCCC ではじまる COBOL2002 のテストデバッガのメッセージが出力されなくなるまで，「テストケースの実行」を実行してください。

値の代入や期待値の設定については，「4.4.7　データ項目に値または期待値を設定する」を参照してください。

COBOL2002 の実行時エラーが起こる場合は，［メッセージ］画面にテストデバッガのメッセージ KCCC1207T-I が出力されるので，COBOL2002 の実行時エラーメッセージを参照して修正してください。COBOL2002 の実行時エラーメッセージは，単体テスト支援の［プロパティページ］で実行時環境変数 CBL_SYSERR を指定して取得します。

実行時環境変数 CBL_SYSERR については，マニュアル「COBOL2002 ユーザーズガイド」を参照してください。

手動判定

テストケースの結果を確認し，その結果を判定して設定する方法です。

表 4-3　結果の判定条件

判定条件	判定結果
結果を確認し，問題ないと判断した場合	OK（手動判定）
結果を確認し，問題ありと判断した場合	NG（手動判定）
結果をまだ確認していない場合	未判定（手動判定）
自動判定で判定結果を設定する場合	自動判定※

注※
　自動判定を選択した場合は，自動判定の結果を判定結果とします。

手動判定の結果については，その回の実行結果だけに適用されます。次回以降に単体テストを実行するごとに，判定結果を設定してください。

また，各データ項目で個別に判定結果は設定できません。

5 テストを繰り返す

単体テスト支援を使って COBOL プログラムの単体テストを繰り返すときの手順について説明します。

5.1　エラーになった COBOL プログラムを修正する

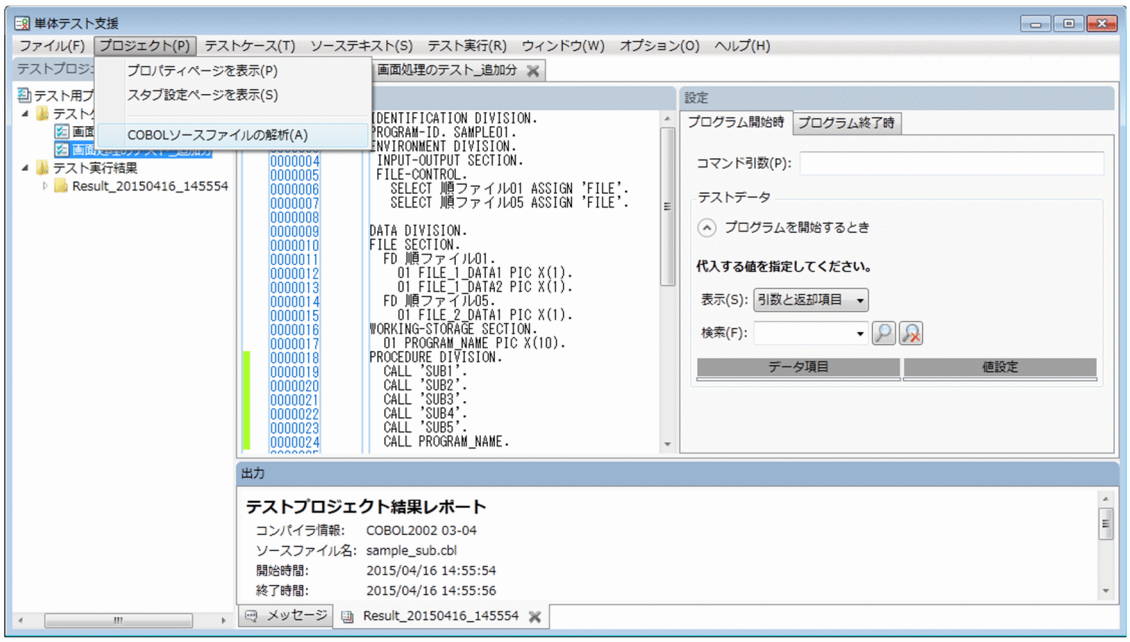
単体テストの結果，COBOL プログラムを修正する手順です。

作業の手順

1. ［ソーステキスト］－［COBOL エディタを開く］メニューを選択します。
   テスト対象の COBOL ソースファイルが COBOL エディタで表示されます。

2. 単体テスト支援の結果レポートを参照し，COBOL エディタで COBOL プログラムを編集し保存します。

3. ［プロジェクト］メニューの［COBOL ソースファイルの解析］を選択します。



   編集後のソーステキストがテストプロジェクトに反映されます。

注意

- COBOL エディタで開く COBOL ソースファイルは，テスト対象としてテストプロジェクトに指定した COBOL ソースファイルです。COBOL エディタでは，［ソーステキスト］画面で位置づいていた行には移動しません。また，COPY 文によって取り込んだ登録集原文を COBOL エディタで開けません。

5.2　COBOL ソースファイルを解析する

単体テスト支援で COBOL ソースファイルを解析する手順です。

COBOL ソースファイルの解析では，単体テストがダミーで作成する主プログラム，副プログラム，ファイルの情報，およびプログラム定義で定義されているデータ項目などの情報を収集します。

COBOL ソースファイルの解析が実行されるタイミングは次のとおりです。

- テストプロジェクトを新しく作成するとき
- ［COBOL ソースファイルの設定］ダイアログで［COBOL ソースファイルの解析］ボタンをクリックしたとき

また，次の場合は，手動で COBOL ソースファイルの解析を実行する必要があります。

- テスト対象の COBOL ソースファイルを編集したとき
- テスト対象の COBOL ソースファイルが取り込んでいる登録集原文を修正したとき

COBOL ソースファイルの解析を実行すると，解析前と解析後の情報が比較され，一致した情報に対するテストデータの設定が引き継がれます。

- 基本項目のデータ項目がレベル番号，データ項目名，ADDRESS 句の有無およびアドレス名と一致するかどうか。

  このときの名前の比較は，-EquivRule コンパイラオプションの指定の有無に関係なく，名前が完全に一致するものだけを引き継ぎます。

- 集団項目に所属するデータ項目は，基本項目の比較条件に加え，集団項目の ADDRESS 句の有無およびアドレス名，集団項目内の定義順，OCCURS 句の有無および繰り返し数，INDEXED 指定の有無および指標名，DEPENDING 指定の有無および制御変数名，RENAMES 指定したデータ項目の有無およびデータ項目名，REDEFINES 指定したデータ項目の有無およびデータ項目名も一致するかどうか。

  一致しない場合は，所属している集団項目の全体が引き継ぎ対象外になります。

  OCCURS 句の繰り返し数については，環境変数 CBLUTS_ANALYZEMAXOCCURSNUM の設定で，OCCURS 句に設定できるテストデータの個数の上限値が変わった場合も引き継ぎ対象外になる場合があります。詳細は，「付録 D　単体テスト支援の環境変数」を参照してください。

  制御変数は，DEPENDING 指定されている集団項目が引き継ぎ対象となった場合，レベル番号の変更や制御変数の所属する集団項目の変更で引き継ぎ対象外になった場合でも，制御変数自体は引き継ぎ対象になります。

  例：

  解析前のデータ定義

```
01  DAT1.
  02  DAT11 OCCURS 10 TIMES
      DEPENDING ON DAT1DO.
    03  DAT111 PIC X.
01  DATDO.
```

```
  02  DAT1DO PIC S9(9).
  02  DAT2DO PIC S9(9).
```

解析後のデータ定義

```
01  DAT1.
  02  DAT11 OCCURS 10 TIMES
      DEPENDING ON DAT1DO.
    03  DAT111 PIC X.
01  DATDO.
  02  DATXDO PIC S9(9).
  02  DAT1DO PIC S9(9).
  02  DAT2DO PIC S9(9).
```

上記は，COBOL ソースファイルを修正し，「02 DATXDO PIC S9(9).」を追加した例です。この場合，データ項目 DATDO 自体は集団項目内の定義順が一致しないため，比較の条件によって引き継ぎ対象外になりますが，データ項目 DAT1 は比較の条件によって引き継ぎ対象となるため，データ項目 DAT1DO は引き継ぎ対象になります。

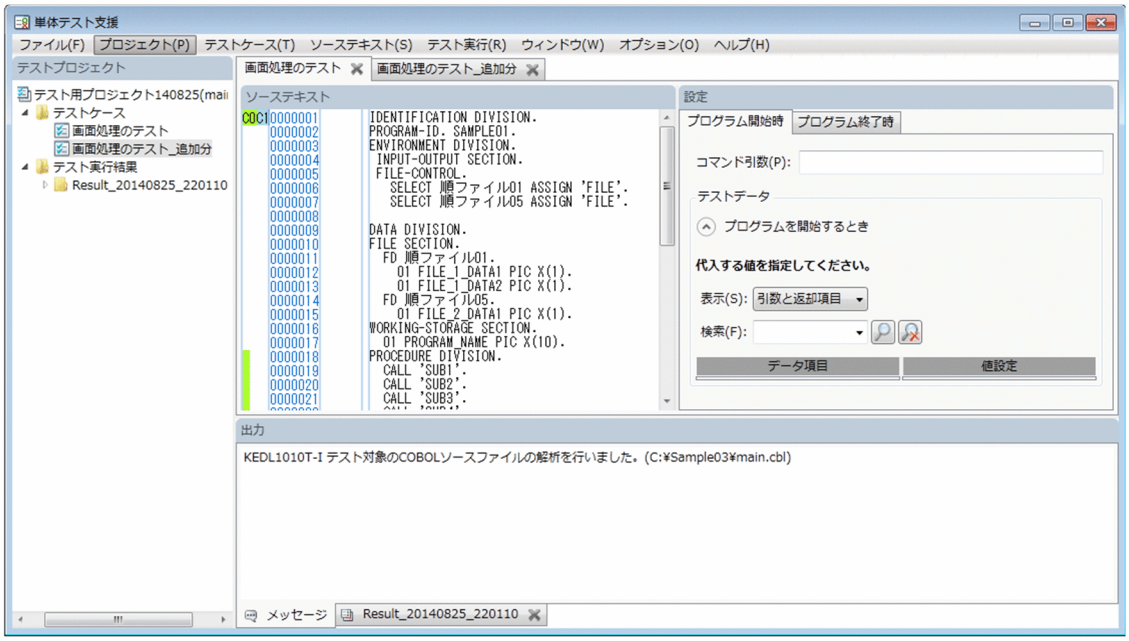
- 中断点の設定は，中断点を設定した文番号の値が一致するかどうか。

  中断点を設定した行に文が複数存在する場合には，中断点を設定した文の順番も一致するかどうかが比較されます。一致しない場合は，その中断点で設定するデータ項目も含めて引き継ぎ対象外になり，中断点は削除されます。

作業の手順

1. ［プロジェクト］－［COBOL ソースファイルの解析］メニューを選択します。

   ［メッセージ］画面に，コンパイラのメッセージなどの解析の結果が表示されます。



2. 解析がエラーになったときは，［メッセージ］画面でエラーメッセージを確認し，エラーの個所を修正します。

エラーが発生しているときは，［スタブ設定ページ］と［テストケースページ］は操作できません。修正方法を次に示します。

- COBOL プログラムに問題があるときは，［ソーステキスト］－［COBOL エディタを開く］メニューを選択して COBOL エディタを開きます。COBOL プログラムを修正し，［プロジェクト］－［COBOL ソースファイルの解析］メニューを選択して解析を再実行します。
- コンパイラオプションまたはコンパイラ環境変数に問題があるときは，［プロジェクト］－［プロパティページを表示］メニューを選択して，［プロパティページ］を開きます。［プロパティページ］の［変更］ボタンをクリックし，［COBOL ソースファイルの設定］ダイアログで修正したあと，［COBOL ソースファイルの解析］ボタンをクリックします。

注意

- COBOL ソースファイルまたは登録集原文を修正したあとに単体テストを実行する場合，「COBOL ソースファイルの解析」を実行し必要に応じてテストデータを再設定したあとで単体テストを実行する必要があります。「COBOL ソースファイルの解析」，またはテストデータを再設定しないで単体テストの実行した場合の結果は保証しません。テストデータの設定方法については，「4.4.7　データ項目に値または期待値を設定する」を参照してください。
- 「COBOL ソースファイルの解析」を実行する前に，すでに設定してあるテストデータは，解析前後のデータ項目の対応が検索され，値は引き継がれます。ただし，対応が取れないデータ項目はテストデータとして設定した値がクリアされます。この場合，対応が取れなかったデータ項目一覧が［メッセージ］画面に表示されます。［メッセージ］画面を参照し，必要な個所にテストデータを再設定してください。
- 「COBOL ソースファイルの解析」は解析処理のため，テストプロジェクトのプロパティに指定したコンパイラオプションに加え，次に示すコンパイラオプションが自動的に指定されます。
  - -SrcList,OutputAll
  - -TDInf
  - -CVInf
  - -SimMain
  - -SimSub
  - -SimIdent
  - -OutputFile
  - -Main,{System¦V3}

  上記のコンパイラオプションが無効になるコンパイラオプションは，テストプロジェクトのプロパティのコンパイラオプションには指定しないでください。指定した場合の単体テスト支援の動作は保証しません。コンパイラオプションの組み合わせによって有効または無効になるコンパイラオプションについては，マニュアル「COBOL2002 ユーザーズガイド」を参照してください。

  注意が必要なコンパイラオプションを次に示します。これらのコンパイラオプションはテストプロジェクトのプロパティには指定しないでください。

コンパイラオプション	指定すると起こる現象
-Dll,{Stdcall｜Cdecl}［32bit 版の COBOL2002 の場合］	「単体テストの実行」でエラーになります。
-Dll［64bit 版の COBOL2002 の場合］	
-Compile,CheckOnly	「単体テストの実行」でエラーになります。
-LiteralExtend,Alnum	次のどれかを［プロパティページ］または［スタブ設定ページ］で設定している場合，-LiteralExtend,Alnum コンパイラオプションが無効化されます。 • 「テスト対象プログラムの種類」で副プログラムを選択する • ［スタブ設定ページ］の「CALL 定数で呼び出す副プログラム名」で 1 つ以上有効にする • ［スタブ設定ページ］の「CALL 一意名で呼び出す副プログラム名」に 1 つ以上のプログラム名を指定する

- 次の個所で設定した環境変数の合計値が 65,535 バイトを超えると，COBOL ソースファイルを解析したときに論理エラーになります。回避するには，不要な環境変数を削除してください。
  - ［新しくテストプロジェクトを作る］ダイアログ，［プロパティページ］または［COBOL ソースファイルの設定］ダイアログで設定したコンパイラ環境変数
  - ［プロパティページ］で設定した実行時環境変数
  - 開発マネージャで設定したコンパイラ環境変数（開発マネージャから起動した場合）
  - システムに設定した環境変数
- OCCURS 指定ありデータ項目には，1 次元当たり最大で 500 までテストデータを設定できます。500 を超える分については，テスト対象にできません（表示，値の設定ができません）。
- OCCURS 指定ありデータ項目の 1 次元当たりに設定できるテストデータの個数の上限値を変える場合は，環境変数 CBLUTS_ANALYZEMAXOCCURSNUM に上限値にしたい値を設定してください。環境変数の詳細は，「付録 D　単体テスト支援の環境変数」を参照してください。
- -StdCall コンパイラオプションを指定しても COBOL ソースファイルと同じフォルダに置いた stdcall 呼び出し指示ファイルは有効になりません。有効にするには，-StdCallFile コンパイラオプションで stdcall 呼び出し指示ファイルを絶対パスで指定してください。指定しない場合は，cdecl 呼び出し規約のプログラムとしてスタブ生成の対象になります。
- 引き継ぎ対象外になった集団項目が，ファイル入出力文で参照するレコード領域の場合は，ファイルシミュレーションの設定画面でも引き継ぎ対象外です。
- ファイルシミュレーションの設定は，ファイル名とオープンモードが一致する場合は引き継がれます。一致しない場合は，引き継ぎ対象外です。

  同じファイルに対して，異なるオープンモードで開いていた場合は，一致しなかった組み合わせのデータだけ引き継ぎ対象外です。
- 中断点の設定は，中断点を設定した一連番号の値が一致する場合は引き継がれます。また，中断点を設定した行に文が複数存在する場合には，中断点を設定した文の順番も一致するかどうかが比較され，一致しない場合は，その中断点で設定するデータ項目も含んで引き継ぎ対象外です。

- ファイル記述項で，別のレコード領域を新たに追加した場合，元からある領域に対してのテストデータの設定は引き継がれます。また，複数あったレコード領域の 1 つを削除した場合，残ったレコード領域に対するテストデータの設定は引き継がれます。

  例：

  左のコードを右に修正した場合も，右のコードを左に修正した場合も，REC1DATA1 に対するテストデータの設定は引き継がれます。

```
FD  FILE1.
01  REC1.
  02  REC1DATA1 PIC 9(9).
```

```
FD  FILE1.
01  REC1.
  02  REC1DATA1 PIC 9(9).
01  REC2.
  02  REC2DATA1 PIC X(1).
  02  REC2DATA2 PIC X(1).
```

  制御変数は，DEPENDING 指定されている集団項目が引き継ぎ対象の場合，レベル番号の変更や制御変数の所属する集団項目が変更となり引き継ぎ対象外となった場合でも，制御変数自体は引き継ぎ対象です。

  例：

解析前のデータ定義

```
01  DAT1.
  02  DAT11 OCCURS 10 TIMES
        DEPENDING ON DAT1DO.
    03  DAT111 PIC X.
01  DATDO.
  02  DAT1DO PIC S9(9).
  02  DAT2DO PIC S9(9).
```

解析後のデータ定義

```
01  DAT1.
  02  DAT11 OCCURS 10 TIMES
        DEPENDING ON DAT1DO.
    03  DAT111 PIC X.
01  DATDO.
  02  DATXDO PIC S9(9).
  02  DAT1DO PIC S9(9).
  02  DAT2DO PIC S9(9).
```

  上記は，COBOL ソースファイルを修正し，「02 DATXDO PIC S9(9).」を追加した例です。この場合，データ項目 DATDO 自体は集団項目内の定義順が一致しないため，比較の条件で引き継ぎ対象外になりますが，データ項目 DAT1 は比較の条件で引き継ぎ対象となるため，データ項目 DAT1DO については引き継ぎ対象です。

6 テスト結果を参照する

単体テスト支援の実行結果を確認する方法について説明します。実行結果として出力するレポートを次に示します。

・テストプロジェクト結果レポート

・テストケース結果レポート

・カバレージレポート

6.1　テストプロジェクト結果レポートを参照する

テストプロジェクト結果レポートの参照方法の説明です。

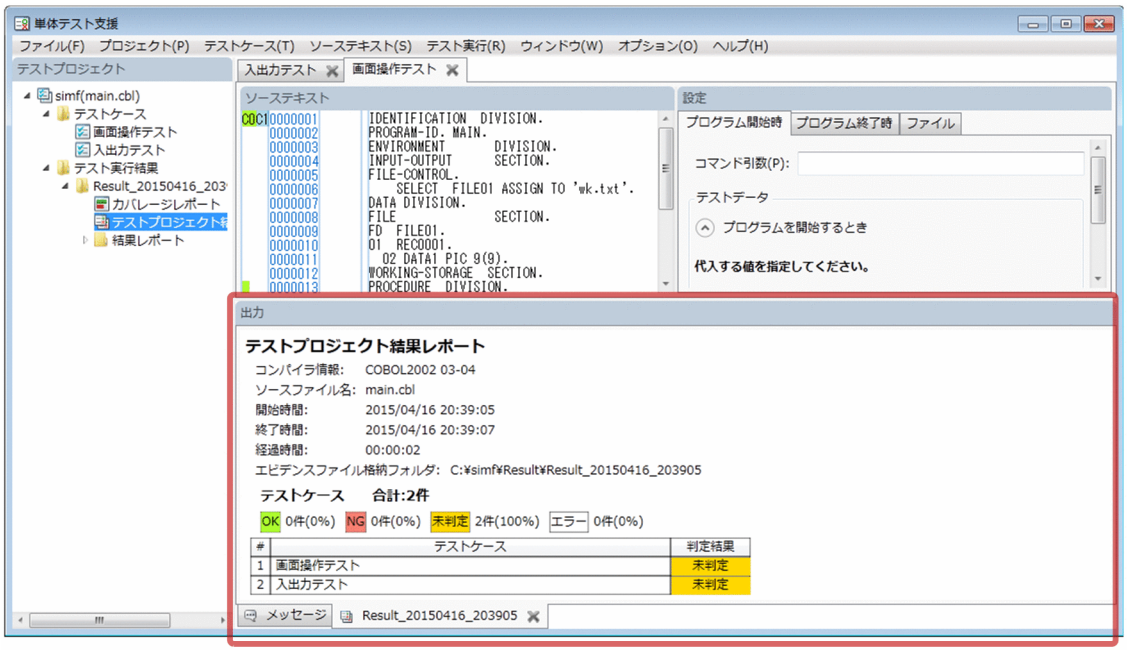
テストプロジェクト結果レポートには，単体テストの実行結果が一覧で表示されます。単体テストを実行したすべてのテストケースが対象です。

作業の前に確認すること

- テストケースの単体テストを実行し，テスト実行結果が作成されている。

作業の手順

1. ［テストプロジェクト］画面の［テスト実行結果］の下にあるタグ（デフォルトでは［Result_yyyymmdd_hhmmss］）から［テストプロジェクト結果レポート］をダブルクリックします。
   「10.9.2　［テストプロジェクト結果レポート］画面」が表示されます。



2. 表示された結果から，NG の項目を参照し COBOL プログラムを修正します。

関連項目

- 「4.10　単体テストを実行したときの判定結果」
- 「10.9.2　［テストプロジェクト結果レポート］画面」

6.2　テストケース結果レポートを参照する

テストケース結果レポートの参照方法の説明です。

テストケース結果レポートは，1 つのテストケースの単体テストを実行した結果です。

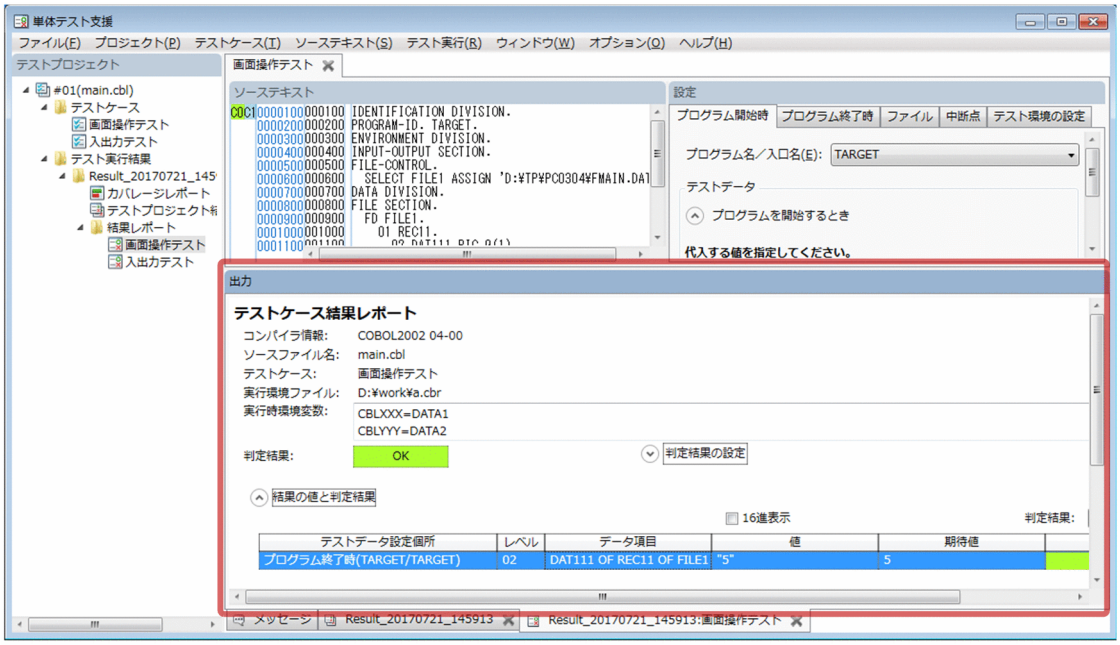
作業の前に確認すること

- テストケースを実行し，テスト実行結果が作成されている。

作業の手順

1. ［テストプロジェクト］画面の［テスト実行結果］の下にあるタグ（デフォルトでは［Result_yyyymmdd_hhmmss］）－［結果レポート］の下にある，テストケース名をダブルクリックします。

   「10.9.3　［テストケース結果レポート］画面」が表示されます。



2. 結果の値と判定結果を確認するため，［結果の値と判定結果］の左にあるボタンを選択して［結果の値と判定結果］の一覧を表示します。

   ［結果の値と判定結果］に表示される内容

項目	内容
テストデータ設定個所	データ項目がテストデータとして設定された位置です。
レベル	データ項目のレベル番号です。
データ項目	データ項目の名前です。
値	実行結果のデータ項目の値です。
期待値	データ項目の期待値です。

項目	内容
判定結果	実行結果の値と期待値を比較した判定結果です。

［値］を 16 進数で表示するときは，［16 進表示］のチェックボックスをオンにします。

判定結果で表示項目を絞り込むときは，［判定結果：］の一覧から OK，NG，または未判定のどれかを選択します。選択した判定結果だけが［結果の値と判定結果］一覧に表示されます。

任意の行を右クリックして，「値と期待値を表示」メニューを選択すると，［結果の値と期待値］ダイアログが表示され，一覧では収まりきらない大きなデータ値を確認できます。

3. ［プログラム開始時］画面の「値設定」で代入した値を確認するため，［代入した値］の左にあるボタンを選択して［代入した値］の一覧を表示します。

［代入した値］に表示される内容

項目	内容
テストデータ設定個所	データ項目がテストデータとして設定された位置です。
レベル	データ項目のレベル番号です。
データ項目	データ項目の名前です。集団項目に属するデータ項目のときは，修飾付きのデータ名で表示されます。
値	テストの実行で代入したデータ項目の値です。

［値］を 16 進数で表示するときは，［16 進表示］のチェックボックスをオンにします。

任意の行を右クリックして，「値を表示」メニューを選択すると，［代入した値］ダイアログが表示され，一覧では収まりきらない大きなデータ値を確認できます。

4. 出力文の実行結果を確認するため，［出力文の実行結果］の左にあるボタンを選択して［出力文の実行結果］の一覧を表示します。

［出力文の実行結果］に表示される内容

項目	内容
ファイル名	シミュレーションした出力文の対象ファイルの名前です。
オープンモード	シミュレーションした出力文の対象ファイルをオープンしたときのオープンモードです。
出力文	シミュレーションした出力文です。
出力文の場所	シミュレーションした出力文の場所の文番号です。
判定結果	実行結果の値と期待値を比較した判定結果です。 終了条件の実行の場合は，「終了条件」と表示されます。 データの組み合わせなどのエラーのときは「エラー」と表示されます。

5. 入力文の実行結果を確認するため，［入力文の実行結果］の左にあるボタンを選択して［入力文の実行結果］の一覧を表示します。

［入力文の実行結果］に表示される内容

項目	内容
ファイル名	シミュレーションした入力文の対象ファイルの名前です。
オープンモード	シミュレーションした入力文の対象ファイルをオープンしたときのオープンモードです。
入力文	シミュレーションした入力文です。
入力文の場所	シミュレーションした入力文の場所の文番号です。
判定結果	値の代入が行われたときは，「---」と表示されます。 終了条件が行われたときは，「終了条件」と表示されます。 データの組み合わせなどのエラーになったときは「エラー」と表示されます。

6. 判定結果を手動で変更するときは，［判定結果の設定］の左にあるボタンを選択します。開いたラジオボタンから， OK，NG，未判定のどれかを選択します。

［テストデータ設定個所］に表示される形式を次に示します。

- プログラム開始時

  形式：プログラム開始時（テスト対象のプログラム名/プログラム開始時の［プログラム名/入口名］で設定したプログラム名または入口名）

  例：テスト対象プログラム SAMPLE で開始時の［プログラム名/入口名］で ENT を指定したときは，「プログラム開始時（SAMPLE/ENT）」と表示されます。

- プログラム終了時

  形式：プログラム終了時（テスト対象のプログラム名/プログラム開始時の［プログラム名/入口名］で設定したプログラム名または入口名）

  例：テスト対象プログラム SAMPLE で開始時の［プログラム名/入口名］で SAMPLE を指定したときは，「プログラム終了時（SAMPLE/SAMPLE）」と表示されます。

- 中断点

  形式：中断点（文番号）

  例：文番号 0001000.1 の個所の中断点のときは，「中断点（0001000.1）」と表示されます。

注意

- テストケースの実行が正常に終了しなかった場合は，そのテストケースに対するテストケース結果レポートは生成されません。また，［出力文の実行結果］および［入力文の実行結果］に表示するデータがない場合は，テストケース結果レポートは生成されません。［出力文の実行結果］および［入力文の実行結果］では，組み合わせエラーなどが発生しても，テストケース結果レポートは表示されます。
- 判定結果を手動で設定した実行結果は，その後のテスト実行には引き継がれません。テストを再実行したときは，再度，手動で設定してください。
- テストケース結果レポートの［結果の値と判定結果］の一覧と［代入した値］の一覧のデータ項目の数は，上限値までが表示されます。上限値以上のデータ項目を確認する場合は，バッチモードで実行して結果を確認してください。バッチモードについては，「8.　バッチモードで実行する」を参照してください。上限値については，「付録 C　単体テスト支援の制限値と限界値」を参照してください。

- 終了条件を実行したときの結果は，判定結果を決めるための評価対象になりません。例えば，次の条件の場合でも，終了条件の結果は評価対象ではないので，テストケース結果レポートの結果は「未判定」になります。
  - 「結果の値と判定結果」に結果がない。
  - ファイルシミュレーションのテストデータは，終了条件だけ。
  - 上記終了条件の実行結果は「無視」。
- 終了条件を設定しても，終了条件を実行する入出力文に必要な文の記載がないと，終了条件は無視されます。例えば，WRITE 文の終了条件に「INVALID KEY 指定の文を実行」を指定しても，その WRITE 文に INVALID KEY 指定がないと，終了条件は無視されます。

  (例)

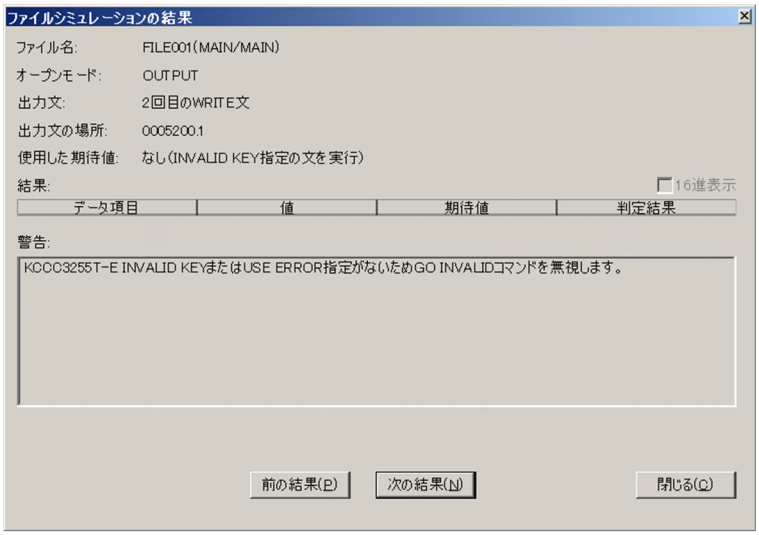
  [条件]
  - WRITE 文を 2 回
  - 設定したテストデータは 1 個
  - 終了条件に「INVALID KEY 指定の文を実行」を指定（テストデータが 1 つなので，2 回目の WRITE 文で実行される）
  - 2 つ目の WRITE 文に INVALID KEY 指定はない。

  [結果]

  テストケース結果レポートの一覧には，次のように表示されます。

ファイル名	オープンモード	入出力文	出力文の場所	判定結果
FILE001(MAIN/MAIN)	OUTPUT	WRITE	0005200.1	OK
FILE001(MAIN/MAIN)	OUTPUT	WRITE	0005200.1	終了条件

このとき，[判定結果] 欄は通常とは変わりませんが，[ファイルシミュレーションの結果] ダイアログで確認できます。[ファイルシミュレーションの結果] ダイアログは，[出力文の実行結果] または [入力文の実行結果] の表を右クリックして [結果を表示] を選択して表示します。



関連項目

- 「4.10　単体テストを実行したときの判定結果」
- 「10.9.3　［テストケース結果レポート］画面」

6.3　カバレージレポートを参照する

カバレージレポートの参照方法の説明です。

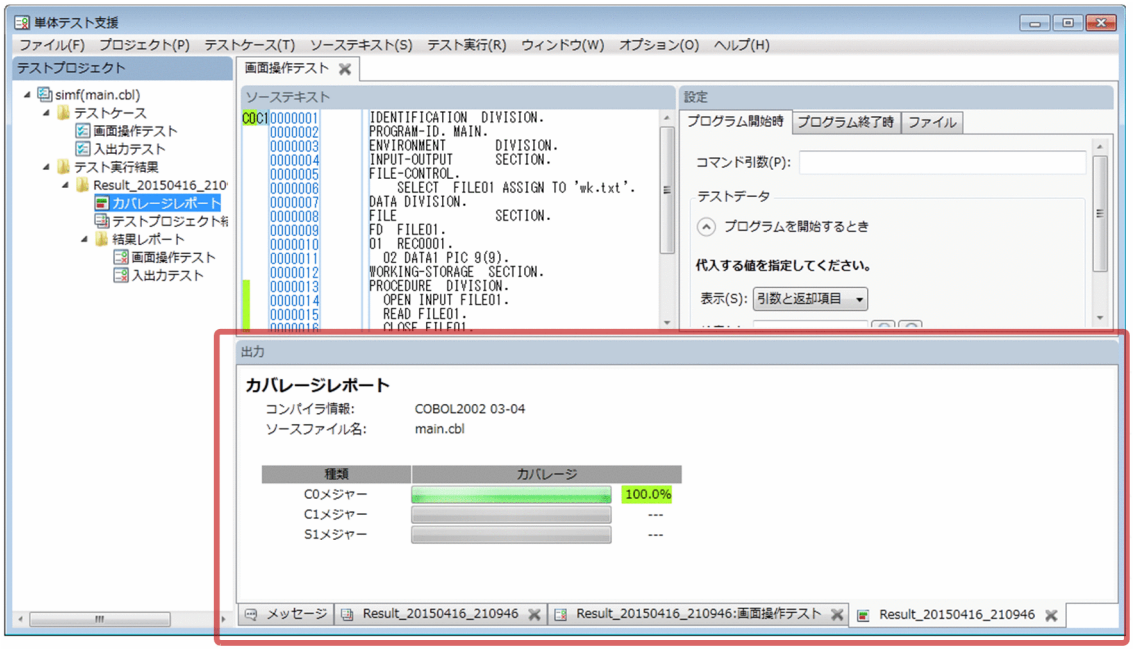
カバレージレポートには，単体テストを実行したときのカバレージ情報が表示されます。

作業の前に確認すること

- テストケースを実行し，テスト実行結果が作成されている。

作業の手順

1. ［テストプロジェクト］画面の［テスト実行結果］の下にあるタグ（デフォルトでは［Result_yyyymmdd_hhmmss］）－［カバレージレポート］をダブルクリックします。
   ［カバレージレポート］画面が表示されます。



注意

- 単体テスト支援では，テストプロジェクト単位でカバレージ情報を生成します。
- カバレージレポートは，単体テストを実行した時点でのカバレージ情報が表示されます。このため，カバレージ情報をクリアしても，カバレージレポートとその内容は更新されません。
- カバレージ情報の対象となる文が COBOL プログラムに含まれないときは，C0 メジャー，C1 メジャー，S1 メジャーの数値には「---」が表示されます。
- ［ソーステキスト］画面上に表示されるカバレージ対象の行は，エビデンスのカバレージ情報リストファイルの内容と一部違いがあるため，［ソーステキスト］画面上ではすべてカバレージ蓄積済みであってもカバレージレポートでは 100%になっていなかったり，［ソーステキスト］画面上ではカバレージ対象の行があってもカバレージレポートでは「---」と表示されたりする場合があります。

- ソーステキストに表示されるカバレージ情報は，カバレージ情報リストファイル（coverage.cll）とは次の違いがあります。
  - PROCEDURE DIVISION，節，段落は，ソーステキスト上では，カバレージ情報の対象になります。
  - 条件分岐で省略された部分は，ソーステキスト上には表示されません。
- 次の個所で設定した環境変数の合計値が 65,535 バイトを超えると，COBOL エディタを開いたときに論理エラーになります。回避するには，不要な環境変数を削除してください。
  - ［新しくテストプロジェクトを作る］ダイアログ，［プロパティページ］または［COBOL ソースファイルの設定］ダイアログで設定したコンパイラ環境変数
  - ［プロパティページ］で設定した実行時環境変数
  - 開発マネージャで設定したコンパイラ環境変数（開発マネージャから起動した場合）
  - システムに設定した環境変数
- COBOL エディタで開く COBOL ソースファイルは，テスト対象としてテストプロジェクトに指定した COBOL ソースファイルです。COBOL エディタでは，［ソーステキスト］画面で位置づいていた行には移動しません。また，COPY 文で取り込んだ登録集原文は COBOL エディタでは開けません。

関連項目

- 「10.9.4 ［カバレージレポート］画面」

6.4　エビデンスファイルを確認する

単体テストを実行した結果である，エビデンスファイルを参照する手順です。

単体テスト支援が出力するエビデンスファイルを次に示します。各エビデンスファイルは，単体テストを1回実行するごとに1種類ずつ生成されます。

出力先は，テストプロジェクト作業フォルダの下のResultフォルダです。

エビデンスファイル名	拡張子	内容
テストデバッガ結果出力ファイル	tdl	テストデバッガが出力する結果出力ファイルです。 ファイル名はlog.tdlで作成されます。
カバレージ情報リストファイル	cll	カバレージ情報を出力するファイルです。 ファイル名はcoverage.cllで作成されます。

作業の前に確認すること

- テストケースを実行し，テスト実行結果が作成されている。

作業の手順

1. テストプロジェクト作業フォルダの¥Result¥Result_*yyyymmdd_hhmmss* フォルダ（*yyyy*：西暦年，*mm*：月，*dd*：日，*hh*：時，*mm*：分，*ss*：秒）を開きます。

   エビデンスファイルであるテストデバッガ結果出力ファイル（log.tdl）とカバレージ情報リストファイル（coverage.cll）が格納されています。

2. テキストエディタでファイルを開いて，内容を確認します。

   テストデバッガ結果出力ファイルの内容については，マニュアル「COBOL2002　操作ガイド」のテストデバッガの説明を参照してください。カバレージ情報ファイルの内容については，マニュアル「COBOL2002　操作ガイド」のカバレージ情報の表示の説明を参照してください。

注意

- エビデンスファイルを直接編集，移動，または削除しないでください。参照とコピーはできます。
- ［テストプロジェクト］画面で，テストプロジェクトを右クリックしポップアップメニューで［除外］を選択した場合，エビデンスファイルは削除されません。［削除］を選択すると，エビデンスファイルは削除されます。

7 テスト結果を操作する

単体テスト支援のテスト結果を操作する手順について説明します。

7.1　手動で判定結果を変更する

テストケースの結果を確認し，その結果を判定して設定する手順です。

設定できる判定を次に示します。

判定	設定する内容
結果を確認し，問題ないと判断した場合	OK（手動判定）
結果を確認し，問題ありと判断した場合	NG（手動判定）
結果をまだ確認していない場合	未判定（手動判定）
自動判定で判定結果を設定する場合	自動判定※

注※
　自動判定の結果が判定結果になります。

手動判定の結果は，その回の実行結果にだけ適用されます。次回以降に単体テストを実行した結果を変更する場合は，再度判定し判定結果を設定してください。

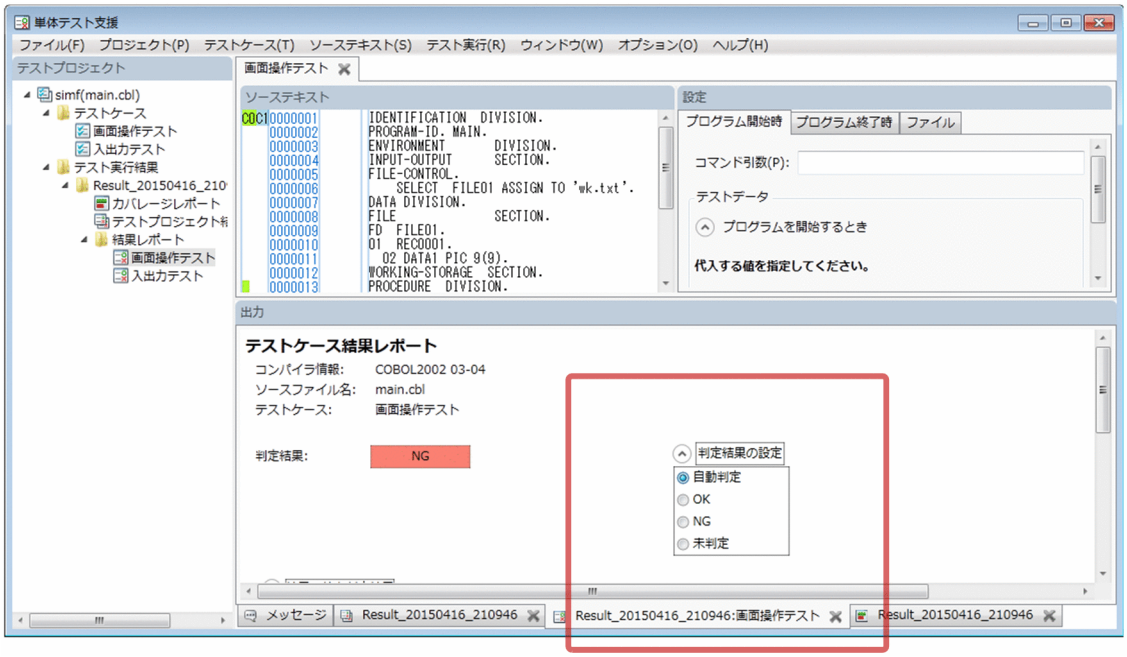
手動判定で変更できる判定結果はテストケース単位です。各データ項目で個別に判定結果を設定できません。

作業の前に確認すること

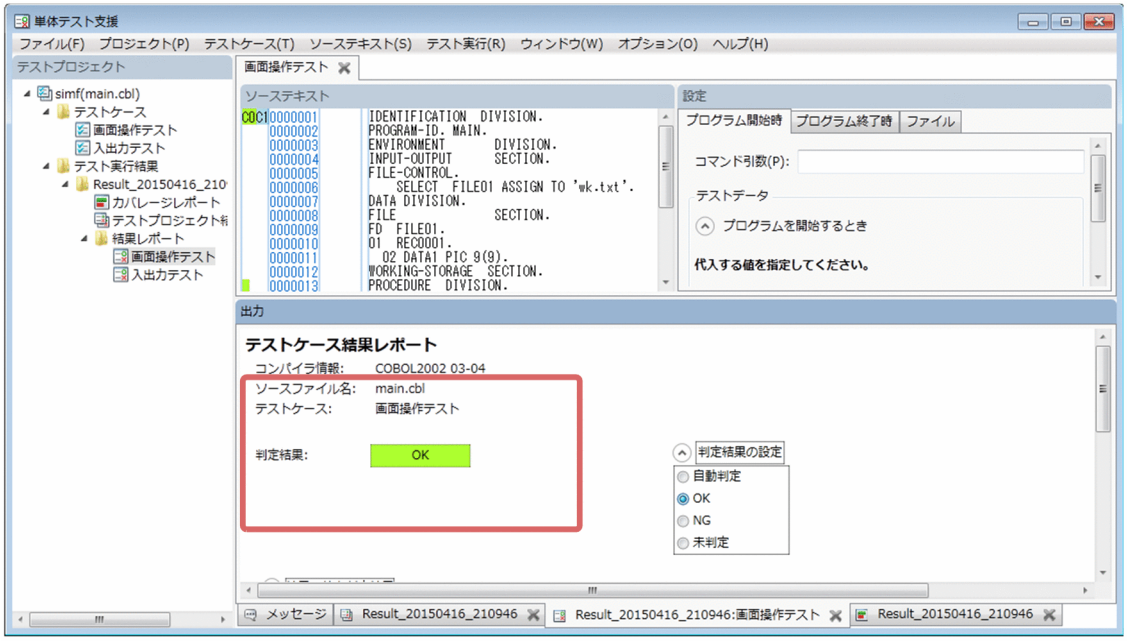
- テストケースを実行し，テスト実行結果が作成されている。

作業の手順

1. ［テストケース結果レポート］画面の［判定結果の設定］から判定結果とする値を選択します。



　選択した判定結果に変更されます。


単体テスト支援
ファイル(F) プロジェクト(P) テストケース(T) ソーステキスト(S) テスト実行(R) ウィンドウ(W) オプション(O) ヘルプ(H)
テストプロジェクト
simf(main.cbl)
テストケース
画面操作テスト
入出力テスト
テスト実行結果
Result_20150416_210
カバレージレポート
テストプロジェクト結
結果レポート
画面操作テスト
入出力テスト
画面操作テスト
ソーステキスト
COC 0000001 IDENTIFICATION DIVISION.
0000002 PROGRAM-ID. MAIN.
0000003 ENVIRONMENT DIVISION.
0000004 INPUT-OUTPUT SECTION.
0000005 FILE-CONTROL.
0000006 SELECT FILE01 ASSIGN TO 'wk.txt'.
0000007 DATA DIVISION.
0000008 FILE SECTION.
0000009 FD FILE01.
0000010 01 REC0001.
0000011 02 DATA1 PIC 9(9).
0000012 WORKING-STORAGE SECTION.
0000013 PROCEDURE DIVISION.
設定
プログラム開始時
プログラム終了時
ファイル
コマンド引数(P):
テストデータ
プログラムを開始するとき
代入する値を指定してください。
出力
テストケース結果レポート
コンパイラ情報: COBOL2002 03-04
ソースファイル名: main.cbl
テストケース: 画面操作テスト
判定結果: OK
判定結果の設定
自動判定
OK
NG
未判定
メッセージ
Result_20150416_210946
Result_20150416_210946:画面操作テスト
Result_20150416_210946

7.2　実行結果の名前を変更する

単体テストの実行結果の名前を変更する手順です。

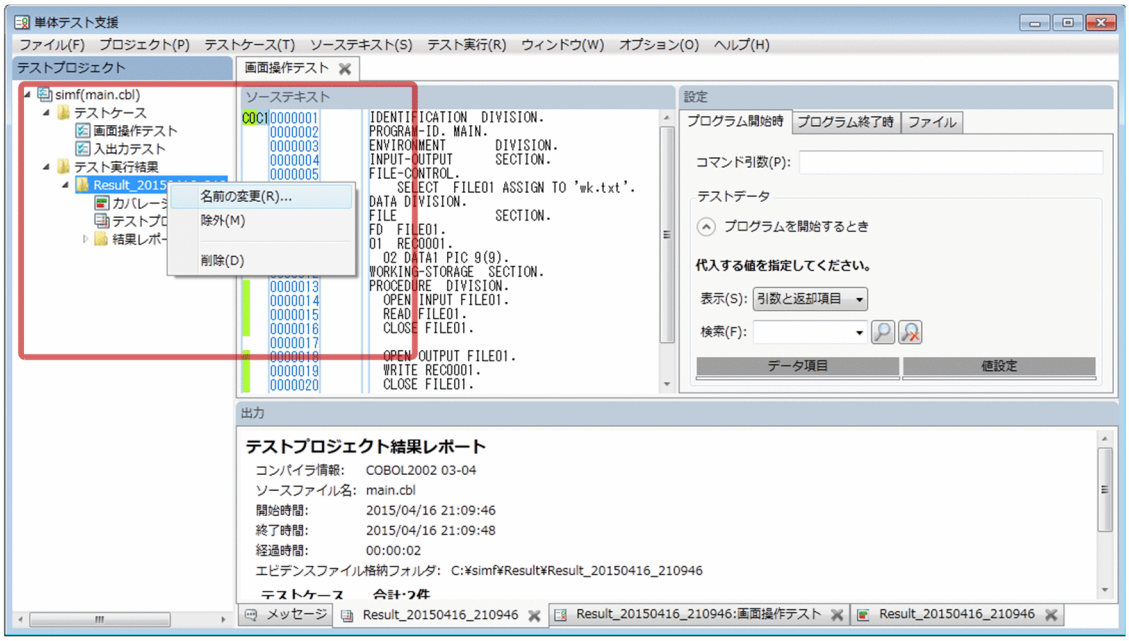
作業の前に確認すること

- テストケースを実行し，テスト実行結果が作成されている。

作業の手順

1. ［テストプロジェクト］画面で［テスト実行結果］の下にあるタグ（デフォルトでは［Result_yyyymmdd_hhmmss］）を右クリックし，ポップアップメニューから［名前の変更］を選択します。

   ［名前の変更］ダイアログが表示されます。



2. ［名前の変更］ダイアログに新しい名前を入力し，［OK］ボタンをクリックします。

   テスト実行結果の名前が変更されます。

注意

- テストの実行結果の名前を変更しても，エビデンスファイルを格納するフォルダ名は変更されません。
- 実行結果を格納するフォルダの場所については，テストプロジェクト結果レポートに表示されているフォルダのパスを参照してください。

7.3　実行結果を削除する

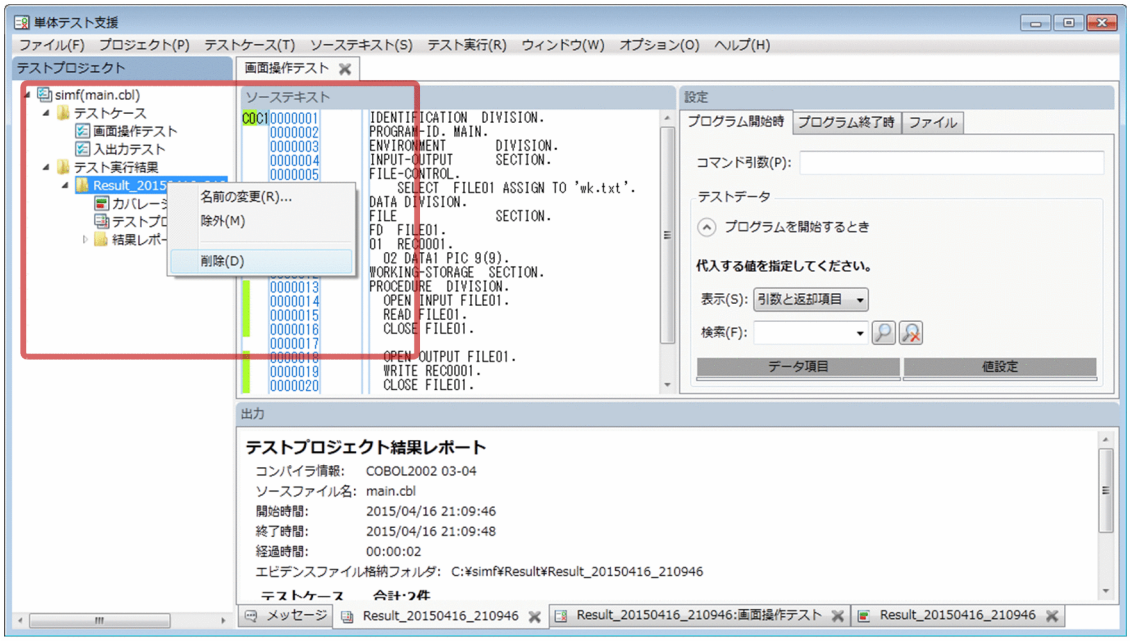
単体テストを実行した結果の情報が不要になった場合に，テストプロジェクトから削除する手順です。

作業の前に確認すること

- テストケースを実行し，テスト実行結果が作成されている。

作業の手順

1. ［テストプロジェクト］画面で［テスト実行結果］の下にあるタグ（デフォルトでは［Result_yyyymmdd_hhmmss］）を右クリックし，ポップアップメニューから［削除］を選択します。



削除して良いかどうかを確認する画面が表示されます。

2. ［OK］ボタンをクリックします。

テスト結果が削除されます。

注意

- テスト結果を削除すると，テストプロジェクトの保存時にエビデンスファイルも削除されます。

7.4　実行結果をテストプロジェクトから除外する

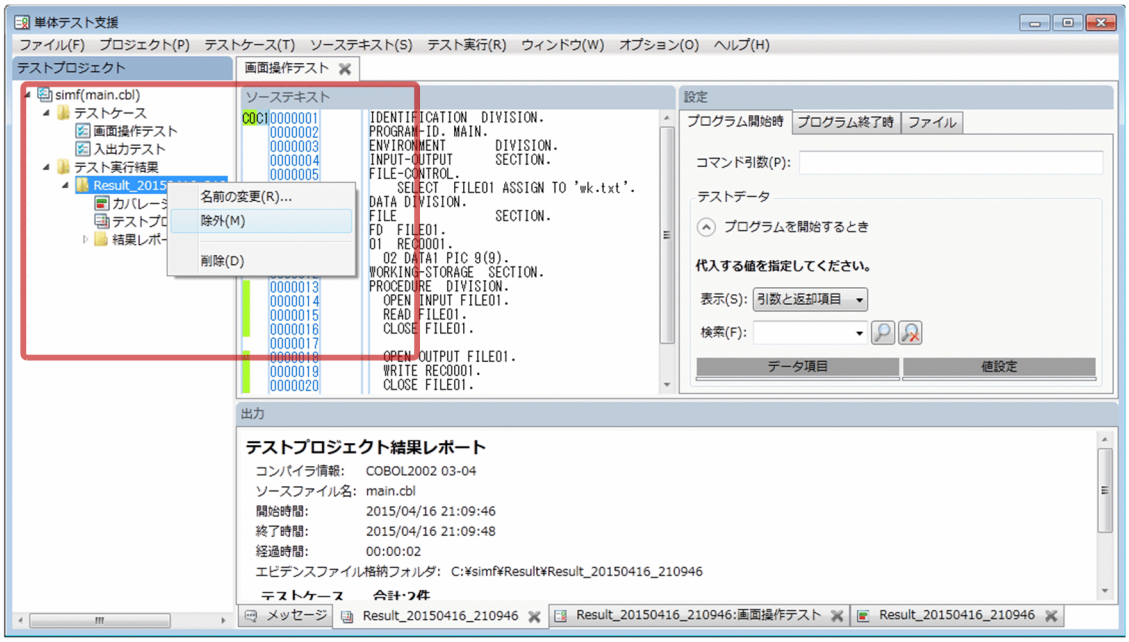
単体テストを実行した結果の情報が不要になったが，エビデンスファイルを残すときは，テストプロジェクトから除外します。

作業の前に確認すること

- テストケースを実行し，テスト実行結果が作成されている。

作業の手順

1. ［テストプロジェクト］画面で［テスト実行結果］の下にあるタグ（デフォルトでは［Result_yyyymmdd_hhmmss］）を右クリックし，ポップアップメニューから［除外］を選択します。



   除外して良いかどうかを確認する画面が表示されます。

2. ［OK］ボタンをクリックします。

   テスト結果が除外されます。

注意

- テスト結果を除外した場合，テストプロジェクトを保存してもエビデンスファイルは削除されません。

7.5　カバレージ情報をクリアする

カバレージ情報をクリアする手順です。

単体テスト支援では単体テストの実行ごとにカバレージを取得し，カバレージ情報として蓄積します。このため，テストケースをすべて再実行する場合にカバレージ情報を 0 から再確認するときは，カバレージ情報のクリアが必要です。

カバレージ情報をクリアすると，次に示す情報がクリアされます。

- C0 メジャー
- C1 メジャー
- S1 メジャー

作業の手順

1. ［ソーステキスト］－［カバレージ情報のクリア］メニューを選択します。
   カバレージ情報をクリアするかどうかを確認するダイアログが表示されます。
2. ［OK］ボタンをクリックします。
   蓄積されたカバレージ情報がクリアされます。
   次の単体テストの実行では，C0 メジャー，C1 メジャー，および S1 メジャーが 0%から蓄積されます。

注意

- カバレージ情報をクリアしても，それ以前に作成したテスト結果に含まれるカバレージレポートについては，クリアされないでテストを実行した時点のカバレージ情報が表示されます。
- クリアすると，クリアしたカバレージ情報は元に戻せません。
- 次の個所で設定した環境変数の合計値が 65,535 バイトを超えると，カバレージ情報をクリアしたときに論理エラーになります。回避するには，不要な環境変数を削除してください。
  - ［新しくテストプロジェクトを作る］ダイアログ，［プロパティページ］または［COBOL ソースファイルの設定］ダイアログで設定したコンパイラ環境変数
  - ［プロパティページ］で設定した実行時環境変数
  - 開発マネージャで設定したコンパイラ環境変数（開発マネージャから起動した場合）
  - システムに設定した環境変数

8 バッチモードで実行する

単体テスト支援をバッチモードで実行する手順について説明します。

8.1　バッチモードでできる単体テスト支援の機能

単体テスト支援の GUI 画面を表示しないで，単体テストを実行するバッチモードの手順です。

バッチモードでは，すでに GUI 画面でテストケースおよびテストデータを設定したテストプロジェクトに対し，コマンドラインから GUI 画面を表示しないで単体テストを実行できます。大量のテストプロジェクトを夜間バッチで実行する場合など，GUI での操作をしないで一括で実行するときに使用します。

また，テストプロジェクトの設定の変更や，テストケースのインポートなどを，複数のテストプロジェクトに対して，GUI 画面を表示しないで一括で操作できます。

バッチモードで使用できる単体テスト支援の機能を次の表に示します。

分類	機能	参照先
テスト実行	すべてのテストケースの実行	8.2.1
テストケース操作	テストケースの削除	8.3.1
	テストケースの一覧表示	8.3.2
	CSV ファイルのインポート（フォルダ指定）	8.3.3
	CSV ファイルのインポート（ファイル指定）	8.3.4
	テストケースのエクスポート（テストケース指定）	8.3.5
	テストケースのエクスポート（ひな形）	8.3.6
テストプロジェクト操作	テストプロジェクトの作成	8.4.1
	プロジェクト情報ファイルの出力	8.4.2
	プロジェクト情報の設定	8.4.3
実行結果操作	カバレージ情報の表示	8.5.1
	カバレージ情報のクリア	8.5.2
	実行結果の削除	8.5.3
	実行結果の一覧表示	8.5.4

8.2　テストケースの実行

8.2.1　すべてのテストケースを実行する

有効になっているすべてのテストケースを実行します。

作業の前に確認すること

- 単体テスト支援でテストケースおよびテストデータを設定している。

作業の手順

1. 単体テストの実行の対象となるテストプロジェクトを準備します。
2. Windows のコマンドプロンプトを起動します。
3. 次に示す cbluts コマンドで，バッチモードで単体テスト支援を起動して単体テストを実行します。

```
cbluts -Test -Run 〔-InputPath プロジェクト情報ファイルパス〕 〔-Detail〕 〔-Force〕 -P
roject 実行対象テストプロジェクトファイルパス
```

テスト実行する直前で，対応が取れないテストデータが見つかった場合は，テスト実行が中止されます。ただし，-Force オプションが指定されていた場合は，対応が取れないテストデータが見つかっても中止しないで，テスト実行します。この場合，対応が取れないテストデータはすべて削除されます。

4. 実行後の結果を確認します。

バッチモードでは，標準出力に実行結果のサマリ情報が表示されます。サマリ情報を次に示します。

- 総テストケース件数
- テストケースの判定結果数
- テストケースごとの判定結果一覧
- カバレージ情報

コマンドラインオプションに-Detail を指定した場合は，サマリ情報に続き，結果の詳細が表示されます。結果の詳細を次に示します。

- テストケースごとの判定結果
- 各テストケースの結果の値と判定結果の個数
- データ項目ごとの結果の値と判定結果の詳細
- 各テストケースの結果の代入した値の個数
- データ項目ごとの代入した値の詳細
- ファイルシミュレーションで出力文を実行したときの結果の値と判定結果
- ファイルシミュレーションで入力文を実行したときに代入した値

注意

- バッチモードで単体テストを実行中に，単体テスト支援の実行プロセスを強制終了すると，実行中のテストプロジェクトのデータに不整合が起こることがあります。強制終了した場合は，単体テストの結果やテストプロジェクトのデータは保証しません。
- バッチモードで単体テストを実行すると，実行結果はテストプロジェクトに追加して上書き保存されます。このため，テストプロジェクトファイルやテストプロジェクト作業フォルダに書き込み権限がない場合には，単体テストを実行しないでエラーになります。
- 結果詳細で表示するデータ項目について，COBOL ソースファイルで明示的に定義していないデータ項目は，レベル番号として「00」が表示されます。
- 古いバージョンの単体テスト支援で作成したテストプロジェクトファイルを対象にした場合，テスト実行後の保存では，新しいフォーマットで保存されます。そのため，古いバージョンの単体テスト支援で開いたり，バッチモードでのテストをしたりできなくなることがあります。
- 終了条件を実行した場合，「レベル」，「データ項目名」，「値」，「16 進表示の値」，「期待値」，「代入値」は空になります。

  「判定結果」は，正しく終了条件が機能すれば空になります。問題があった場合は，「無視」と表示されます。
- 03-05 より前のバージョンでサポートしていたオプションを使ってテスト実行することもできます。ただし，03-05 より前のバージョンでサポートしていたオプションと，03-05 以降のバージョンでサポートするオプションを混在して使うことはできません。例えば，次のような指定はできません。

```
cbluts -Run,All -Detail -Project C:¥work¥sample.utsproj
cbluts -Run,All -Force -Project C:¥work¥sample.utsproj
```

- ファイル名が「-」（ハイフン）で始まるテストプロジェクトファイルを-Project オプションの引数に指定しないでください。指定した場合はエラーになります。エラーになった場合，テストプロジェクトファイルのファイル名を「-」（ハイフン）で始まらない名前に変更してください。
- -Force オプションを指定しないでテスト実行した場合，対応の取れないテストデータが見つかってテスト実行が中止されたときにテストプロジェクトは保存されません。テストプロジェクトが保存されないため，テストプロジェクトファイルからテストデータは削除されません。

関連項目

- 「cbluts コマンド － バッチモードで単体テスト支援を実行する」

8.3　テストケースの操作

8.3.1　テストケースを削除する

指定したテストケースを削除します。

作業の前に確認すること

- テストケースの内容を確認し，どのテストケースを削除するか決めている。

作業の手順

1. 単体テストの実行の対象となるテストプロジェクトを用意します。
2. Windows のコマンドプロンプトを起動します。
3. 次に示す cbluts コマンドで，テストケースを削除します。
   - テストケースの名前を指定して削除する場合
     一度，テストケースの一覧を表示して，名前を確認します。

```
cbluts -TestCase -List -Project 実行対象テストプロジェクトファイルパス
```

     次のコマンドで，名前を指定して削除します。

```
cbluts -TestCase -Delete -Name 削除するテストケースの名前 -Project 実行対象テストプロジェクトファイルパス
```

     コマンドを実行すると次の確認が行われます。

```
KEDL3042T-I テストケースを削除します。(削除するテストケースの名前)
KEDL3032T-I 削除しますか？ (Y/N)
```

     この確認で，Y を入力すると，テストケースは削除されます。N を入力すると，テストケースの削除は中止されます。

   - すべてのテストケースを削除する場合
     次のコマンドで削除します。

```
cbluts -TestCase -Delete -All -Project 実行対象テストプロジェクトファイルパス
```

     コマンドを実行すると次の確認が行われます。

```
KEDL3022T-I すべてのテストケースを削除します。
KEDL3032T-I 削除しますか？ (Y/N)
```

     この確認で，Y を入力すると，テストケースは削除されます。N を入力すると，テストケースの削除は中止されます。

削除するときの確認を行いたくない場合は，-Quiet オプションを指定してください。

注意

- -All オプションを指定したときに対象が複数存在していた場合，どれかのテストケースの削除でエラーが発生しても，エラーメッセージが出力された上で処理が継続され，残りのテストケースの削除が行われます。

関連項目

- 「cbluts コマンド － バッチモードで単体テスト支援を実行する」

8.3.2　テストケースを一覧表示する

テストケースを一覧表示します。

作業の手順

1. 単体テストの実行の対象となるテストプロジェクトを用意します。
2. Windows のコマンドプロンプトを起動します。
3. 次に示す cbluts コマンドで，テストケースの一覧を表示します。

```
cbluts -TestCase -List -Project 実行対象テストプロジェクトファイルパス
```

関連項目

- 「cbluts コマンド － バッチモードで単体テスト支援を実行する」

8.3.3　CSV ファイルをインポートする（フォルダ指定）

指定したフォルダの直下またはそのサブフォルダの下にある CSV ファイルをテストケースとして取り込みます。

作業の前に確認すること

- インポートする CSV ファイル内で，テストデータの設定が完了している。
- CSV ファイルで設定するテストデータが正しい。

作業の手順

1. 単体テストの実行の対象となるテストプロジェクトを用意します。
2. Windows のコマンドプロンプトを起動します。
3. 次に示す cbluts コマンドで，CSV ファイルをインポートします。

```
cbluts -TestCase -Import -InputPath CSVが格納されているフォルダパス 〔-Skip | -OverWrite
| -Rename〕 〔-Recursive〕 -Project 実行対象テストプロジェクトファイルパス
```

-Recursive オプションを指定すると，-InputPath オプションで指定したフォルダのサブフォルダも含めてインポートされます。

CSV ファイル名と同じ名前のテストケースがすでにインポート先にあった場合，デフォルトでは，その CSV ファイルのインポートがスキップされます。

動作を変えたい場合は，-OverWrite オプションまたは-Rename オプションを指定してください。

注意

- -Recursive オプションを指定した場合は，フォルダ構成が深さ優先で探索されて，取り込まれます。

  同じ階層に複数のフォルダがあったときは，フォルダ名の順番で取り込まれます。例えば，次のフォルダ構成で，-InputPath オプションに TOPFOLDER を指定した場合は，TOPFOLDER，AAA，CCC，BBB の順に取り込まれます。

  ```
  TOPFOLDER
   ├ AAA
   │   └ CCC
   └ BBB
  ```

  同名のファイルが存在する場合の動作を「上書き」としている場合は，あとから取り込んだファイルで上書きされます。

- 取り込み中にエラーが発生した場合の処理の継続可否は，GUI の動作に準じます。
- コンパイルエラーが発生している場合や，プログラム情報ファイルに必要な情報が含まれていない場合，このコマンドは実行できません。

  その場合は，先に COBOL ソースファイルの内容や，コンパイルオプションを見直して，エラーを解消しておく必要があります。

関連項目

- 「cbluts コマンド － バッチモードで単体テスト支援を実行する」

8.3.4 CSV ファイルをインポートする（ファイル指定）

指定した CSV ファイルをテストケースとして取り込みます。

作業の前に確認すること

- インポートする CSV ファイル内で，テストデータの設定が完了している。
- CSV ファイルで設定するテストデータが正しい。

作業の手順

1. 単体テストの実行の対象となるテストプロジェクトを用意します。

2. Windows のコマンドプロンプトを起動します。

3. 次に示す cbluts コマンドで，CSV ファイルをインポートします。

```
cbluts -TestCase -Import -InputPath CSVファイルのパス 〔-Skip | -OverWrite | -Rename〕 -P
roject 実行対象テストプロジェクトファイルパス
```

CSV ファイル名と同じ名前のテストケースがすでにインポート先にあった場合，デフォルトでは，その CSV ファイルのインポートがスキップされます。

動作を変えたい場合は，-OverWrite オプションまたは-Rename オプションを指定してください。

取り込みたい CSV ファイルが複数ある場合は，上記のコマンドをファイルごとに実行するか，または取り込む CSV ファイルを一時的に 1 つのフォルダにまとめて，そのフォルダを指定して取り込んでください。

注意

- 取り込み中にエラーが発生した場合の処理の継続可否は，GUI の動作に準じます。
- コンパイルエラーが発生している場合や，プログラム情報ファイルに必要な情報が含まれていない場合，このコマンドは実行できません。

  その場合は，先に COBOL ソースファイルの内容や，コンパイルオプションを見直して，エラーを解消しておく必要があります。

関連項目

- 「cbluts コマンド － バッチモードで単体テスト支援を実行する」

8.3.5 テストケースをエクスポートする（テストケース指定）

指定したテストケースを CSV ファイルとしてエクスポートします。

作業の手順

1. 単体テストの実行の対象となるテストプロジェクトを用意します。

2. Windows のコマンドプロンプトを起動します。

3. 次の手順でテストケースをエクスポートします。

   - ファイル名を指定してエクスポートする場合

     一度，テストケースの一覧を表示して，名前を確認します。

```
cbluts -TestCase -List -Project 実行対象テストプロジェクトファイルパス
```

     次のコマンドで，名前を指定してエクスポートします。

```
cbluts -TestCase -Export -Name テストケース名 -OutputPath 出力先のフォルダパス -Project 実行対象テストプロジェク
トファイルパス
```

- すべてのテストケースをエクスポートする場合

  次のコマンドで，すべてのテストケースをエクスポートします。

```
cbluts -TestCase -Export -All -OutputPath 出力先のフォルダパス 〔-Skip | -OverWrite〕 -Project 実行対象テストプロジェクトファイルパス
```

テストケースと同じ名前の CSV ファイルがすでに出力先のフォルダにあった場合，デフォルトでは，そのテストケースのエクスポートがスキップされます。CSV ファイルを上書きしてよい場合は，-OverWrite オプションを指定してください。

関連項目

- 「cbluts コマンド － バッチモードで単体テスト支援を実行する」

8.3.6 テストケースをエクスポートする（ひな形）

テストケースのひな形を CSV ファイルとしてエクスポートします。

作業の手順

1. 単体テストの実行の対象となるテストプロジェクトを用意します。
2. Windows のコマンドプロンプトを起動します。
3. 次に示す cbluts コマンドで，ひな形をエクスポートします。

```
cbluts -TestCase -ExportTemplate -OutputPath 出力先のフォルダパス 〔-Skip | -OverWrite〕
-Project 実行対象テストプロジェクトファイルパス
```

ひな形は，「ひな形.csv」の名前で出力されます。

出力先にすでに「ひな形.csv」があった場合，デフォルトでは，ひな形のエクスポートがスキップされます。上書きしてよい場合は，-OverWrite オプションを指定してください。

関連項目

- 「cbluts コマンド － バッチモードで単体テスト支援を実行する」

8.4　テストプロジェクトの操作

8.4.1　テストプロジェクトを作成する

テストプロジェクトファイルを作成します。

作業の前に確認すること

- コンパイラオプション，リンクするファイル，実行時環境変数など，単体テストするときに必要なものを確認する。

作業の手順

1. 単体テストをしたい COBOL ソースファイルを用意します。
2. テストプロジェクトファイルを配置したいフォルダとテストプロジェクトファイルのファイル名を決めます。
3. Windows のコマンドプロンプトを起動します。
4. 次に示す cbluts コマンドで，テストプロジェクトを作成します。

```
cbluts -TestProject -Create -Source COBOLソースファイルのパス 〔-InputPath プロジェクト情報ファイルのパス〕 -Project 実行対象テストプロジェクトファイルパス
```

コンパイラオプションなどを付加したい場合は，付加する内容が記載されたプロジェクト情報ファイルを指定してください。プロジェクト情報ファイルの各項目については，「8.4.3 プロジェクト情報を設定する」を参照してください。

ただし，ここで指定するプロジェクト情報ファイルの次の要素は，プロジェクト情報の設定時と動作が異なります。

- <COBOLSourceFile>要素
  指定していても無視されます。また，この要素を空要素とすることもできます。
- <EnableTestCaseSetting>要素
  ON を指定すると，警告が出力され，OFF が仮定されます。
- <TestCaseSettings>要素
  <EnableTestCaseSetting>の値に OFF が仮定されるため，指定されている内容が無視されます。

注意

- 最外プログラムが複数存在する場合，定義順で最も上のプログラムがテスト対象プログラムとなります。
- プロジェクト情報ファイルを指定しなかった場合，テスト対象プログラムの種類は「副プログラム」で作成されます。
- プロジェクト名，テストプロジェクト作業フォルダは，次の規則で決まります。

プロジェクト名：

指定したテストプロジェクトファイルのファイル名（拡張子は含まない）

テストプロジェクト作業フォルダ：

指定したテストプロジェクトファイルを含むフォルダ

（例）

-Project オプションに c:¥work¥sample.utsproj を指定した場合，次のようになります。

プロジェクト名：sample

テストプロジェクト作業フォルダ：c:¥work

- テストプロジェクトファイルを出力するフォルダがない場合は，自動で作成されます。
- ファイル名が「-」（ハイフン）で始まるテストプロジェクトファイルを-Project オプションの引数に指定しないでください。指定した場合はエラーになります。エラーになった場合，テストプロジェクトファイルのファイル名を「-」（ハイフン）で始まらない名前に変更してください。

関連項目

- 「11.2 プロジェクト情報ファイルの構成要素」
- 「cbluts コマンド － バッチモードで単体テスト支援を実行する」

8.4.2 プロジェクト情報ファイルを出力する

プロジェクト情報ファイルを出力します。

作業の手順

1. 単体テストの実行の対象となるテストプロジェクトを用意します。
2. Windows のコマンドプロンプトを起動します。
3. 次に示す cbluts コマンドで，プロジェクト情報ファイルをエクスポートします。

```
cbluts -TestProject -Export -OutputPath プロジェクト情報ファイルのパス 〔-Skip | -OverWri
te〕 -Project 実行対象テストプロジェクトファイルパス
```

-OutputPath オプションに指定したファイルがすでにある場合，デフォルトではエクスポートされません。上書きで出力したい場合は，-OverWrite オプションを指定してください。

4. 出力されたプロジェクト情報ファイルの内容を確認したり，設定用に内容を変更したりします。

注意

- プロジェクト情報ファイルは xml で記述してください。文字コードはシフト JIS（Shift_JIS）とします。

関連項目

- 「11.2 プロジェクト情報ファイルの構成要素」

- 「cbluts コマンド － バッチモードで単体テスト支援を実行する」

8.4.3　プロジェクト情報を設定する

指定したプロジェクト情報ファイルの内容で，プロジェクト情報を更新します。プロジェクト情報を更新したあとは，COBOL ソースファイルの解析が行われます。なお，旧バージョンで作成済みのプロジェクト情報ファイルはそのまま使用できます。

作業の前に確認すること

- 設定で使用するプロジェクト情報ファイルに，変更する内容をすべて記載している。
- プロジェクト情報ファイルに記載した内容が正しい。

作業の手順

1. 単体テストの実行の対象となるテストプロジェクトを用意します。
2. Windows のコマンドプロンプトを起動します。
3. 次のどちらかの方法で，プロジェクト情報ファイルを準備します。
   - プロジェクト情報ファイルを出力して，その内容を編集します。
     出力方法は，「8.4.2 プロジェクト情報ファイルを出力する」を参照してください。
   - プロジェクト情報ファイルを作成します。
     作成するときのフォーマットは，次の表および「11.2 プロジェクト情報ファイルの構成要素」を参照してください。

表 8-1　プロジェクト情報ファイルに記載する内容

要素			要否	備考
<CblUnitTestSupport>			必須	－
	<Property>		任意	［プロパティページ］の内容を更新する場合に必要です。
		<ProjectName>	任意	指定しても無視されます。
		<ProgramName>	任意	指定しても無視されます。
		<WorkingFolder>	任意	指定しても無視されます。
		<COBOLSourceFile>	任意	COBOL ソースファイルのパスを指定します。 空要素にはできません。 相対パスで指定した場合，cbluts コマンドを実行したフォルダからの相対パスとして絶対パスに変換して設定されます。 COBOL ソースファイル名を変更することはできません。
		<TargetProgramType>	任意	テスト対象プログラムの種類を変更するときに指定します。

要素				要否	備考
					次のどれかの値を指定します。 • Main,System： 主プログラム（-Main,System）にする場合 • Main,V3： 主プログラム（-Main,V3）にする場合 • Sub： 副プログラムにする場合 大文字と小文字は区別されます。
		<CompilerOptions>		任意	コンパイラオプションを指定するとき，またはコンパイラオプションを削除するときに指定します。
			<CompilerOption>	任意	コンパイラオプションを指定します。 この要素がない場合，コンパイラオプションは変更されません。 コンパイラオプションを削除するときは，空要素を 1 つだけ指定します。 コンパイラオプションを複数指定するときは，1 つの要素にまとめて記載するか，または別の要素に分けて記載します。 （例）次の指定は同じ結果になります。 • 指定 1 <CompilerOption>-Option1 -Option2</CompilerOption> • 指定 2 <CompilerOption>-Option1</CompilerOption> <CompilerOption>-Option2</CompilerOption>
		<CompilerEnvironmentVariables>		任意	コンパイラ環境変数を指定するとき，またはコンパイラ環境変数を削除するときに指定します。
			<CompilerEnvironmentVariable>	任意	コンパイラ環境変数を指定します。 この要素がない場合，コンパイラ環境変数は変更されません。 コンパイラ環境変数を削除するときは，空要素を 1 つだけ指定します。 この要素には値がありません。 環境変数名および値は次の属性に指定します。 • 環境変数名：name 属性 • 値：value 属性 name 属性と value 属性には空文字を指定できます。※1
		<LinkFiles>		任意	リンクするファイルを指定するとき，またはリンクするファイルの設定を解除するときに指定します。

要素				要否	備考
			<LinkFile>	任意	リンクするファイルのパスを指定します。 相対パスで指定した場合，絶対パスに変換しないでそのまま設定されます。 この要素がないと，リンクするファイルは変更されません。 リンクするファイルの設定を解除するときは，空要素を1つだけ指定します。 1つの要素で，1つのファイルを指定します。
		<RuntimeEnvironmentFile>		任意	実行環境ファイルを指定します。 相対パスで指定した場合，cbluts コマンドを実行したフォルダからの相対パスとして絶対パスに変換して設定されます。 空要素にすると，実行環境ファイルの設定が解除されます。
		<RuntimeEnvironmentVariables>		任意	実行時環境変数を指定するとき，または実行時環境変数を削除するときに指定します。
			<RuntimeEnvironmentVariable>	任意	実行時環境変数を指定します。 この要素がないと，実行時環境変数は変更されません。 実行時環境変数を削除するときは，空要素を1つだけ指定します。 この要素には値がありません。 環境変数名および値は次の属性に指定します。 • 環境変数名：name 属性 • 値：value 属性 name 属性と value 属性には空文字が指定できます。※1
		<EnableTestCaseSetting>		任意	テストケースごとに実行環境ファイルまたは実行時環境変数を設定するかどうかを指定します。次のどちらかの値を指定します。 • ON：有効 • OFF：無効 大文字と小文字は区別されます。この要素は省略できます。省略すると OFF（無効）が設定されます。
	<Stub>			任意	［スタブ設定ページ］の内容を更新する場合に必要です。
		<ConstantNames>		任意	CALL 定数で呼び出す副プログラムに対して，ダミーのプログラムを作成するかどうかを変更するときに指定します。
			<ConstantName>	任意	CALL 定数で呼び出す副プログラムのうち，ダミーのプログラムを作成するかどうかを変更したいプログラム名を指定します。 この要素がないと CALL 定数呼び出しのダミープログラムの作成要否は変更されません。

要素				要否	備考
					この要素には値がありません。 プログラム名およびダミープログラムを作成するかどうかは次の属性に指定します。 • プログラム名： name 属性 • ダミープログラムを作成： value 属性 ・ON：作成します。 ・OFF：作成しません。 ・空文字：無視します。 name 属性と value 属性には空文字が指定できます。※2 この要素で指定しなかった副プログラムについては，ダミープログラムを作成するかどうかが変更されません。 指定した副プログラムの呼び出しがなかった場合は無視されます。
		<IdentNames>		任意	CALL 一意名で呼び出す副プログラムに対して，ダミーのプログラムを作成するかどうかを変更するときに指定します。
			<IdentName>	任意	CALL 一意名で呼び出す副プログラムのうち，ダミーのプログラムを作成するものを指定します。 この要素がないと設定されません。 ダミーのプログラムの作成をすべてやめるときは，空要素を 1 つだけ指定します。 対象のプログラムが複数ある場合は，1 つの要素で 1 つのプログラムを指定します。 （例） 「Prog1」と「Prog2」の 2 つのプログラムを登録する場合 正しい例： <IdentName>Prog1</IdentName> <IdentName>Prog2</IdentName> 誤った例： <IdentName>Prog1 Prog2</IdentName> この場合，「Prog1 Prog2」という名前で設定されます。
		<FileNames>		任意	入出力するファイル名に対して，ダミーのファイルを作成するかどうかかを変更するときに指定します。
			<FileName>	任意	入出力するファイル名のうち，ダミーのファイルを作成するかどうかを変更したいファイル名を指定します。 この要素がない場合は設定されません。 ファイル名およびダミーファイルを作成するかどうかは次の属性に指定します。 • ファイル名： name 属性 • ダミーファイルを作成： value 属性 ・ON：作成します。

要素					要否	備考
						・OFF：作成しません。 ・空文字：無視します。 name 属性と value 属性には空文字が指定できます。※2 この要素で指定しなかったファイル名については，ダミーのファイルを作成するかどうかが変更されません。 指定したファイル名の入出力文がなかった場合は無視されます。
	<TestCaseSettings>				任意	<EnableTestCaseSetting>の値または要素の有無によって，この要素の扱いが異なります。 • <EnableTestCaseSetting>の値が ON の場合 この要素の指定は必須です。ON の場合，この要素が指定されていないときはエラーとなります。 テストケースごとの設定が何もないときは，<TestCaseSetting>要素とその配下の要素が空で 1 つだけ出力されます。 • <EnableTestCaseSetting>の値が OFF の場合 この要素は無視されます。 • <EnableTestCaseSetting>の要素自体がない場合 <TestCaseSettings>要素で指定している内容は無視されます。
		<TestCaseSetting>			任意	テストケースごとの実行環境を設定する場合に指定します。 設定対象のテストケース名を次の属性に指定してください。 テストケース名：testcasename 属性 testcasename 属性が存在しない値のとき，警告が出力され，次の<TestCaseSetting>要素の読み取りに移ります。 testcasename 属性が空のときは警告を出力しないで，次の<TestCaseSetting>要素の読み取りに移ります。
			<RuntimeEnvironmentFile>		任意	実行環境ファイルを指定します。 相対パスで指定した場合，cbluts コマンドを実行したフォルダからの相対パスとして絶対パスに変換して設定されます。 空要素にすると，実行環境ファイルの設定が解除されます。
			<RuntimeEnvironmentVariables>		任意	テストケースごとの実行環境ファイルまたは実行時環境変数を設定するとき，または削除するときに指定します。
				<RuntimeEnviro	任意	テストケースに設定する実行時環境変数を指定します。 この要素がないと，実行時環境変数は変更されません。

要素					要否	備考
				nmentVariable>		実行時環境変数を削除するときは，空要素を 1 つだけ指定します。 この要素には値がありません。 環境変数名および値は次の属性に指定します。 • 環境変数名：name 属性 • 値：value 属性 name 属性と value 属性には空文字が指定できます。※1

注※1

各値の組み合わせによる動作は次に示すとおりです。

value 属性の値	name 属性の値	
	任意の文字列	空文字
任意の文字列	正常	エラー
空文字	正常（指定した環境変数に空文字を設定します）	無視

注※2

各値の組み合わせによる動作は次に示すとおりです。

value 属性の値	name 属性の値	
	任意の文字列	空文字
ON/OFF	正常	エラー
空文字	無視	無視

4. 次に示す cbluts コマンドで，手順 3 で準備したプロジェクト情報ファイルをインポートします。

```
cbluts -TestProject -Import -InputPath プロジェクト情報ファイルのパス 〔-Quiet〕 -Project
 実行対象テストプロジェクトファイルパス
```

指定したプロジェクト情報ファイルの内容でプロジェクト情報を更新したあと，COBOL ソースファイルの解析が行われます。この解析で，対応の取れなくなったテストデータを発見した場合，対応の取れなくなったテストデータに対するメッセージを出力したあとに，次の確認メッセージが表示されます。

```
…（対応の取れなくなったテストデータに対するメッセージ）
：
KEDL3029T-W テストデータ設定後に，COBOLソースファイルが更新されました。続行すると，対応の
取れないテストデータは削除されます。
KEDL3030T-I 続行しますか？（Y/N）
```

Y を入力すると，対応が取れないテストデータは削除されてインポートが続行されます。N を入力すると，インポートは中止されてプロジェクトの内容はコマンドを実行する前の状態のままになります。対応が取れなくなったテストデータに対するメッセージの内容を確認したあと，Y か N を入力してください。この確認を行いたくない場合は，-Quiet オプションを指定してください。

注意

- プロジェクト情報ファイルは xml で記述してください。文字コードはシフト JIS（Shift_JIS）とします。
- プロジェクト情報の設定前後で内容に変更がなくても，COBOL ソースファイルの解析と保存が行われます。

関連項目

- 「11.2 プロジェクト情報ファイルの構成要素」
- 「cbluts コマンド － バッチモードで単体テスト支援を実行する」

8.5　実行結果の操作

8.5.1　カバレージ情報を表示する

カバレージの蓄積状況を表示します。

作業の手順

1. 単体テストの実行の対象となるテストプロジェクトを用意します。
2. Windows のコマンドプロンプトを起動します。
3. 次に示す cbluts コマンドで，カバレージ情報をクリアします。

```
cbluts -TestResult -DisplayCoverage -Project 実行対象テストプロジェクトファイルパス
```

関連項目

- 「cbluts コマンド － バッチモードで単体テスト支援を実行する」

8.5.2　カバレージ情報をクリアする

カバレージ情報をクリアします。

作業の前に確認すること

- カバレージ情報をクリアすると，クリアしたカバレージ情報は元に戻せないが，問題ないか確認する。※

  注※　テスト実行したときに，0 から再蓄積することになります。

作業の手順

1. 単体テストの実行の対象となるテストプロジェクトを用意します。
2. Windows のコマンドプロンプトを起動します。
3. 次に示す cbluts コマンドで，カバレージ情報をクリアします。

```
cbluts -TestResult -ClearCoverage -Project 実行対象テストプロジェクトファイルパス
```

コマンドを実行すると，次の確認メッセージが表示されます。

```
KEDL3038T-I カバレージ情報をクリアしますか？ (Y/N)
```

Y を入力すると，カバレージ情報はクリアされます。N を入力すると，カバレージ情報のクリアは中止されます。この確認を行いたくない場合は，-Quiet オプションを指定してください。

関連項目

- 「cbluts コマンド － バッチモードで単体テスト支援を実行する」

8.5.3　実行結果を削除する

指定した実行結果を削除します。

作業の前に確認すること

- 実行結果の内容を確認し，どの実行結果を削除するか決めている。

作業の手順

1. 単体テストの実行の対象となるテストプロジェクトを用意します。
2. Windows のコマンドプロンプトを起動します。
3. 次の手順で実行結果を削除します。
   - 実行結果の名前を指定して削除する場合
     一度，実行結果の一覧を表示して，名前を確認します。

```
cbluts -TestResult -List -Project 実行対象テストプロジェクトファイルパス
```

     次のコマンドで，名前を指定して削除します。

```
cbluts -TestResult -Delete -Name 削除する実行結果の名前 -Project 実行対象テストプロジェクトファイルパス
```

     コマンドを実行すると次の確認が行われます。

```
KEDL3023T-I 実行結果を削除します。(削除する実行結果の名前)
KEDL3032T-I 削除しますか？ (Y/N)
```

     Y を入力すると，実行結果は削除されます。N を入力すると，実行結果の削除は中止されます。

   - すべてのテストケースを削除する場合
     次のコマンドで削除します。

```
cbluts -TestResult -Delete -All -Project 実行対象テストプロジェクトファイルパス
```

     コマンドを実行すると次の確認が行われます。

```
KEDL3036T-I すべての実行結果を削除します。
KEDL3032T-I 削除しますか？ (Y/N)
```

     Y を入力すると，実行結果は削除されます。N を入力すると，実行結果の削除は中止されます。

   削除するときの確認を行いたくない場合は，-Quiet オプションを指定してください。

注意

- -All オプションを指定した場合，どの実行結果の削除でエラーが発生しても，エラーメッセージを出力した上で処理は継続し，残りの実行結果の削除を行います。

関連項目

- 「cbluts コマンド － バッチモードで単体テスト支援を実行する」

8.5.4　実行結果を一覧表示する

実行結果の一覧を表示します。

作業の手順

1. 単体テストの実行の対象となるテストプロジェクトを用意します。
2. Windows のコマンドプロンプトを起動します。
3. 次に示す cbluts コマンドで，実行結果の一覧を表示します。

```
cbluts -TestResult -List -Project 実行対象テストプロジェクトファイルパス
```

注意

- 実行結果の名前を変更した場合は，変更後の名前が表示されます。
  出力時には名前で昇順に並べ替えられますが，このときに使用する名前も変更後の名前が使用されます。

関連項目

- 「cbluts コマンド － バッチモードで単体テスト支援を実行する」

8.6 バッチモードでできる単体テスト支援の使用例

8.6.1 大量の COBOL ソースファイルから一括でプロジェクトファイルを作成する

テスト対象の COBOL ソースファイルと，テストプロジェクトファイル（コマンドを実行することで作成される）を指定することで，テストプロジェクトを作成できます。大量の COBOL ソースファイルがあっても，簡単なスクリプトで作成できます。

また，作成するときに複数のプロジェクトで共有したい設定がある場合，プロジェクト情報ファイルをあらかじめ作成しておき，コマンドに指定することで，その内容を反映してテストプロジェクトを作成することができます。

作業の手順

1. テストプロジェクトファイルを置くフォルダを決めます。

   プロジェクトは最外プログラム単位で作成します。

   1 つのファイルに複数の最外プログラムがある場合は，いちばん上に定義されているプログラムがテスト対象になります。それ以外のプログラムをテスト対象にしたいときは，ソースを分けるか，単体テスト支援の GUI 画面で作成してください。

   テストプロジェクトは，すでにテストプロジェクトファイルが存在する場所には作成できません。テストプロジェクトごとに決める必要があります。

2. プロジェクト情報ファイルを作成します。

   特別なコンパイルオプションや，リンクするファイルなどがなければ，プロジェクト情報ファイルがなくてもテストプロジェクトは作成できます。

3. 次に示す cbluts コマンドで，テストプロジェクトファイルを作成します。

```
cbluts -TestProject -Create -Source C:¥work¥sample.cbl -InputPath C:¥work¥setting.xml -Pr
oject C:¥work¥sample.utsproj
```

4. 作成したい COBOL ソースファイルの分だけ，手順 1～3 を繰り返します。

関連項目

- 「8.4.1 テストプロジェクトを作成する」
- 「11.2 プロジェクト情報ファイルの構成要素」
- 「cbluts コマンド － バッチモードで単体テスト支援を実行する」

8.6.2　新しい環境でプログラムの動作確認をやり直す

新しい環境で動作確認をやり直すときは，これまでのテストで蓄積された情報がカバレージ情報として残っているため，これをクリアします。

また，動作させるパスなどが変わる場合もあるため，COBOL ソースファイルのパスなどの情報を変更する必要があります。

作業の手順

1. テストプロジェクトを，動作確認する場所にコピーします。
2. 移動先のプロジェクトを対象にして，プロジェクト情報ファイルを出力します。

```
cbluts -TestProject -Export -OutputPath c:¥work¥setting.xml -Project sample.utsproj
```

3. 出力したプロジェクト情報ファイルの内容を，動作確認する場所に合わせて修正します。
4. 修正したプロジェクト情報ファイルを入力します。

```
cbluts -TestProject -Import -InputPath c:¥work¥setting.xml -Project sample.utsproj
```

プロジェクト情報ファイルは，テスト実行するときに指定することもできます。

5. カバレージ情報をクリアします。

```
cbluts -TestResult -ClearCoverage -Project sample.utsproj
```

6. 確認のため，カバレージ情報を出力します。

```
cbluts -TestResult -DisplayCoverage -Project sample.utsproj
```

7. テスト実行します。

```
cbluts -Test -Run -Project sample.utsproj
```

8. 対象のプロジェクトが複数あるときは，プロジェクトごとに手順 1～7 を繰り返します。

関連項目

- 「8.2.1 すべてのテストケースを実行する」
- 「8.4.2 プロジェクト情報ファイルを出力する」
- 「8.4.3 プロジェクト情報を設定する」
- 「8.5.1 カバレージ情報を表示する」
- 「8.5.2 カバレージ情報をクリアする」
- 「8.6.3 プロジェクト情報を変更する」
- 「11.2 プロジェクト情報ファイルの構成要素」

- 「cbluts コマンド － バッチモードで単体テスト支援を実行する」

8.6.3　プロジェクト情報を変更する

次のプロジェクト情報を変更します。

- COBOL ソースファイル（ファイル名は変更できません）
- テスト対象プログラムの種類
- コンパイラオプション
- コンパイラ環境変数
- リンクするファイル
- 実行環境ファイル
- 実行時環境変数
- CALL 定数で呼び出す副プログラムのスタブ作成要否
- CALL 一意名で呼び出す副プログラムのスタブ作成要否
- 入出力するファイルのスタブ作成要否
- テストケースごとに実行環境ファイルまたは実行時環境変数を変更する

これらは，プロジェクト情報ファイルに変更する内容を記載し，そのファイルをテストプロジェクトに設定することで変更できます。

プロジェクト情報ファイルは，テストプロジェクトから出力して編集するか，または新規に作成して使用します。

コンパイラオプションを変更する手順を示します。

作業の手順

1. 次のコマンドで，プロジェクト情報ファイルを出力します。

```
cbluts -TestProject -Export -OutputPath c:¥work¥setting.xml -Project sample.utsproj
```

例えば，すでに「-Option1」と「-Option2」が設定されていた場合，次のように，「-Option1 -Option2」と記述されたファイルが出力されます。

```
<CblUnitTestSupport>
  <Property>
    <CompilerOptions>
      <CompilerOption>-Option1 -Option2</CompilerOption>
    </CompilerOptions>
  </Property>
  :
<CblUnitTestSupport>
```

2. プロジェクト情報ファイルを編集します。

新しく「-Option3」を加える場合は，<CompilerOption>の内容を次のように修正します。

```
<!-- 既存の行に追加 -->
<CompilerOption>-Option1 -Option2 -Option3</CompilerOption>
```

または

```
<CompilerOption>-Option1 -Option2</CompilerOption>
<!-- 行を追加 -->
<CompilerOption>-Option3</CompilerOption>
```

既存のオプションを「-Option3」に置き換える場合は，元の内容を次のように修正します。

```
<!-- 既存の行の内容を置き換える -->
<CompilerOption>-Option3</CompilerOption>
```

コンパイラオプションを削除したいときは，空要素を 1 つだけ記述します。

```
<!-- 既存の行を空要素にする。-->
<CompilerOption></CompilerOption>
```

3. 次のコマンドで，プロジェクト情報ファイルを設定します。

```
cbluts -TestProject -Import -InputPath c:¥work¥setting.xml -Project sample.utsproj
```

関連項目

- 「8.4.2 プロジェクト情報ファイルを出力する」
- 「8.4.3 プロジェクト情報を設定する」
- 「11.2 プロジェクト情報ファイルの構成要素」
- 「cbluts コマンド － バッチモードで単体テスト支援を実行する」

8.6.4 リグレッションテストなどのために繰り返しテスト実行する

繰り返しテスト実行する場合は，実行するコマンドをバッチファイル（.bat）にして実行すれば作業の手間が省けます。

また，実行結果か正しいどうかの判定もバッチファイルに追加しておけば，繰り返しテスト実行するたびに，正しい結果かどうか自動で判定することができます。

実行結果を自動で判定する方法としては，例えば次の方法があります。

- 実行結果のサマリ情報に，NG やエラーがないことを確認する
- 事前に出力しておいた正しい結果と，リグレッションテストの結果を比較する

作成するバッチファイルを test.bat として，次に手順を示します。

作業の手順

1. バッチファイル（test.bat）を作成します。

バッチファイルの例を次に示します。find などの DOS コマンドについては，使用する Windows のヘルプを参照してください。

実行結果のサマリ情報に，NG やエラーがないことを確認する場合

スクリプトの例を次に示します。この例では，テスト実行の結果を testresult.txt に出力します。出力した内容に「NG[0]」と「エラー[0]」があるか検索し，どちらかが見つからない場合にエラーとしています。

```
rem テスト実行して，結果をtestresult.txtに出力する。
cbluts -Test -Run -Project c:¥work¥sample.utsproj > testresult.txt
rem testresult.txtの内容を確認する。
find "NG[0]" testresult.txt | find "エラー[0]"
if ERRORLEVEL 1 goto ERROR
rem 正常時
rem ここに正常時の処理を追加
goto :FINALLY
rem エラー処理
:ERROR
rem ここにエラー時の処理を追加
:FINALLY
rem 最後に出力したtestresult.txtを削除する。
del testresult.txt
```

事前に出力しておいた正しい結果と，リグレッションテストの結果を比較する場合

スクリプトの例を次に示します。この例では，テスト実行の結果を testresult.txt に出力します。出力したファイルと，事前に作成しておいた c:¥work¥correct_result.txt を比較し，差分がある場合にエラーとしています。

```
rem テスト実行して，結果をtestresult.txtに出力する。
cbluts -Test -Run -Project c:¥work¥sample.utsproj > testresult.txt
rem 2つのファイルを比較する。
fc testresult.txt c:¥work¥correct_result.txt
if ERRORLEVEL 1 goto ERROR
rem 正常時
rem ここに正常時の処理を追加
goto :FINALLY
rem エラー処理
:ERROR
rem ここにエラー時の処理を追加
:FINALLY
rem 最後に出力したtestresult.txtを削除する。
del testresult.txt
```

この方法で判定する場合は，次のコマンドで，事前に正常終了した結果を保存しておく必要があります。

```
cbluts -Test -Run -Project c:¥work¥sample.utsproj > c:¥work¥correct_result.txt
```

2. バッチファイル（test.bat）を実行します。

次のどちらかの方法で，test.bat を実行します。

- test.bat ファイルをダブルクリックする
- Windows のコマンドプロンプト上，またはほかのバッチファイルから test.bat を呼び出す

```
call test.bat
```

注意事項

- 各コマンドにファイルを指定するときは，「指定するファイルがあるフォルダ」と「バッチファイルを実行するフォルダ」が同一であるかどうかを注意する必要があります。

  フォルダが同一の場合はファイル名だけで指定することもできます。同一ではない場合は，絶対パスか相対パスで指定する必要があります。

- テストプロジェクト内に登録された実行結果の数が上限に達すると，テスト実行できなくなります。

  この場合は，いちばん古い実行結果を自動で削除するように単体テスト支援の GUI 画面で設定しておくことで回避できます。詳細は，「9.5 テスト実行の設定を変更する」を参照してください。

  なお，この設定をしても，実行結果の登録数が上限に達していないときは，実行結果は削除されません。また，この設定をすると，ほかのテストプロジェクトのテスト実行でも有効になります。

関連項目

- 「8.2.1 すべてのテストケースを実行する」
- 「9.5 テスト実行の設定を変更する」
- 「cbluts コマンド － バッチモードで単体テスト支援を実行する」

8.6.5　ほかのテストプロジェクトにあるテストケースを流用する

ほかのテストプロジェクトで使用しているテストケースを流用するには，流用元のテストプロジェクトからテストケースを CSV ファイルとしてエクスポートし，その CSV ファイルを流用先のテストプロジェクトにインポートします。

作業の手順

1. 流用元のテストプロジェクトを対象にして，流用したいテストケースをエクスポートします。

エクスポートするには，次のどちらかの方法があります。操作を実行すると，csvfolder フォルダに CSV ファイルが出力されます。

- 特定のテストケースをエクスポートしたい場合

```
cbluts -TestCase -Export -Name 流用するテストケース -OutputPath csvfolder -Project from.utsproj
```

  -Name オプションの場合，一度に複数のテストケースは指定できません。

- 複数のテストケースをエクスポートしたい場合

  一括でエクスポートすることはできないため，「特定のテストケースをエクスポートしたい場合」の手順を繰り返すか，または「8.6.7 複数のテストケースを一括でエクスポートする」を参照してエクスポートしてください。

- すべてのテストケースをエクスポートしたい場合

```
cbluts -TestCase -Export -All -OutputPath csvfolder -Project from.utsproj
```

出力した CSV ファイルの名前は，「テストケースの名前.csv」になります。

2. 出力した CSV ファイルをインポートします。

- ファイルを指定してインポートしたい場合

```
cbluts -TestCase -Import -InputPath csvfolder¥流用するテストケース.csv -Project sample.utsproj
```

- フォルダを指定してインポートしたい場合

```
cbluts -TestCase -Import -InputPath csvfolder -Project sample.utsproj
```

  -InputPath オプションに CSV ファイルの格納先フォルダを指定すると，そのフォルダ直下の CSV ファイルがすべてインポートされます。

  -Recursive オプションを指定すると，さらに，サブフォルダにあるすべての CSV ファイルもインポートされます。

テストケース名は，CSV ファイルの名前（拡張子「.csv」を除いたもの）になります。

テストケース内で使用している中断点や，データ名がない場合，そのテストデータは，インポート時に削除されます。

3. 流用先のテストプロジェクトが複数ある場合は，手順 2 を繰り返します。

関連項目

- 「8.3.2 テストケースを一覧表示する」
- 「8.3.3 CSV ファイルをインポートする（フォルダ指定）」
- 「8.3.4 CSV ファイルをインポートする（ファイル指定）」
- 「8.3.5 テストケースをエクスポートする（テストケース指定）」
- 「8.6.7 複数のテストケースを一括でエクスポートする」
- 「cbluts コマンド － バッチモードで単体テスト支援を実行する」

8.6.6　サブフォルダにある CSV ファイルを一括でインポートする

CSV ファイルをインポートするときに-Recursive オプションを指定すると，サブフォルダにある CSV ファイルも含めてすべてインポートすることができます。

作業の手順

1. CSV ファイルが置かれたフォルダが複数ある場合は，どこかの階層で共通のフォルダを持つようにします。

   次の構成の場合を例にします。この例では，3 つの「テストケースがあるフォルダ」が，「共通のフォルダ」でつながっています。

```
共通のフォルダ
├ テストケースがあるフォルダ 1
│　└ テストケースがあるフォルダ 2
└ テストケースがあるフォルダ 3
```

2. -Recursive オプションを指定して，CSV ファイルをインポートします。

```
cbluts -TestCase -Import -InputPath 共通のフォルダ -Recursive -Project sample.utsproj
```

   -Recursive オプションを指定するときは，-InputPath オプションにファイルのパスは指定できません。

   -InputPath オプションで指定したフォルダの直下に CSV ファイルがなくても，サブフォルダを含めてどこかのフォルダに CSV ファイルがあれば構いません。

   フォルダの下に拡張子が「.csv」以外のファイルがあった場合，そのファイルは無視されます。

   - -InputPath オプションに，「共通のフォルダ」を指定した場合

     次のフォルダにある CSV ファイルがすべて取り込まれます。

     ・共通のフォルダ

     ・テストケースがあるフォルダ 1

     ・テストケースがあるフォルダ 2

     ・テストケースがあるフォルダ 3

     取り込まれる順番も上記の順番で取り込まれます。

   - -InputPath オプションに，「テストケースがあるフォルダ 1 」を指定した場合

     次のフォルダにある CSV ファイルがすべて取り込まれます。

     ・テストケースがあるフォルダ 1

     ・テストケースがあるフォルダ 2

     取り込まれる順番も上記の順番で取り込まれます。

     次のフォルダにある CSV ファイルは取り込まれません。

     ・共通のフォルダ

     ・テストケースがあるフォルダ 3

3. 対象のプロジェクトが複数あるときは，プロジェクトごとに手順 1，2 を実施します。

関連項目

- 「8.3.3 CSV ファイルをインポートする（フォルダ指定）」
- 「cbluts コマンド － バッチモードで単体テスト支援を実行する」

8.6.7　複数のテストケースを一括でエクスポートする

テストケースのエクスポートでは，1 つまたはすべてのテストケースを指定できます。そのため，複数のテストケースをエクスポートしたい場合は，次のどちらかの方法で実施します。

- テストケースの一覧をリダイレクトして保存し，その内容を簡単なスクリプトに変更する
- すべてを対象にしてエクスポートしたあと，不要な CSV ファイルを削除する

対象のプロジェクトが複数ある場合は，プロジェクトごとに次の手順が必要です。

作業の手順

テストケースの一覧をリダイレクトして保存し，その内容を簡単なスクリプトに変更する場合

1. 次のコマンドで，テストケースの一覧を「list.txt」にリダイレクトします。

```
cbluts -TestCase -List -Project sample.utsproj > list.txt
```

2. list.txt の内容のうち，エクスポートしないテストケース名を行ごと削除します。

   （例）

   次の内容で，「テストケース 2」をエクスポートしない場合

   - 修正前の list.txt

```
[ ON]テストケース1
[ ON]テストケース2
[OFF]テストケース3
:
```

   - 修正後の list.txt

```
[ ON]テストケース1
[OFF]テストケース3
:
```

3. 手順 2 で修正した list.txt のうち，テストケース名よりも前の部分（[ON]または[OFF]）を，cbluts コマンドに変更します。

   - 修正前の list.txt

```
[ ON]テストケース1
[OFF]テストケース3
```

```
:
```

- 修正後の list.txt

```
cbluts -TestCase -Export -OutputPath csvfolder -Project sample.utsproj -Name テストケース1
cbluts -TestCase -Export -OutputPath csvfolder -Project sample.utsproj -Name テストケース3
:
```

「-Name テストケース名」や「-Project プロジェクトファイルパス」の順番は必須ですが，ほかのオプションの順番は任意です。

上の例では，修正量を減らすため「-Project プロジェクトファイルパス」を-Name オプションの前に挿入しています。

4. list.txt の内容を実行します。

実行するには次の方法があります。

- list.txt を list.bat に変更してバッチファイルとして一括で処理させる
- list.txt の内容をコマンドプロンプトにコピー&ペーストして実行する

すべてを対象にしてエクスポートしたあと，不要な CSV ファイルを削除する場合

1. 次のコマンドで，テストケースをすべてエクスポートします。

```
cbluts -TestCase -Export -All -OutputPath csvfolder -Project sample.utsproj
```

上記のコマンドを実行すると，csvfolder にすべてのテストケースがエクスポートされます。

2. csvfolder に出力された CSV ファイルのうち，不要なものを削除します。

関連項目

- 「8.3.2 テストケースを一覧表示する」
- 「8.3.5 テストケースをエクスポートする（テストケース指定）」
- 「cbluts コマンド － バッチモードで単体テスト支援を実行する」

8.6.8 複数のテストケースまたは実行結果を一括で削除する

テストケースの削除では，1 つまたはすべてのテストケースを指定できます。同様に，実行結果の削除も，1 つまたはすべての実行結果を指定できます。そのため，複数のテストケースまたは実行結果を削除したい場合は，次の方法を実施します。

- テストケースまたは実行結果の一覧をリダイレクトして保存し，その内容を簡単なスクリプトに変更する

テストケースの場合にかぎり，次の方法でも削除できます。

- すべてのテストケースを対象にしてエクスポートと削除を実施し，必要なものだけをインポートする

対象のプロジェクトが複数ある場合は，プロジェクトごとに次の手順が必要です。

作業の手順

テストケースの一覧をリダイレクトして保存し，その内容を簡単なスクリプトに変更する場合

1. 次のコマンドで，テストケースの一覧を「list.txt」にリダイレクトします。

```
cbluts -TestCase -List -Project sample.utsproj > list.txt
```

2. list.txt の内容のうち，削除しないテストケース名を行ごと削除します。

（例）

次の内容で，「テストケース 2」を削除しない場合

- 修正前の list.txt

```
[ ON]テストケース1
[ ON]テストケース2
[OFF]テストケース3
:
```

- 修正後の list.txt

```
[ ON]テストケース1
[OFF]テストケース3
:
```

3. 手順 2 で修正した list.txt のうち，テストケース名よりも前の部分（[ON]または[OFF]）を，cbluts のコマンドに変更します。

- 修正前の list.txt

```
[ ON]テストケース1
[OFF]テストケース3
:
```

- 修正後の list.txt

```
cbluts -TestCase -Delete -Project sample.utsproj -Name テストケース1
cbluts -TestCase -Delete -Project sample.utsproj -Name テストケース3
:
```

「-Name テストケース名」や「-Project プロジェクトファイルパス」の順番は必須ですが，ほかのオプションの順番は任意です。

上の例では，修正量を減らすため「-Project プロジェクトファイルパス」を-Name オプションの前に挿入しています。

4. list.txt の内容を実行します。

実行するには次の方法があります。

- list.txt を list.bat に変更してバッチファイルとして一括で処理させる
- list.txt の内容をコマンドプロンプトにコピーペーストして実行する

すべてのテストケースを対象にしてエクスポートと削除を行い，必要なものだけをインポートする場合

1. 次のコマンドで，テストケースをすべてエクスポートします。

```
cbluts -TestCase -Export -All -OutputPath csvfolder -Project sample.utsproj
```

上記のコマンドを実行すると，csvfolder にすべてのテストケースがエクスポートされます。

2. csvfolder に出力された CSV ファイルのうち，不要なものを削除します。

3. すべてのテストケースを削除します。

```
cbluts -TestCase -Delete -All -Project sample.utsproj
```

4. csvfolder を指定して，テストケースをインポートします。

```
cbluts -TestCase -Import -InputPath csvfolder -Project sample.utsproj
```

関連項目

- 「8.3.1 テストケースを削除する」
- 「8.3.2 テストケースを一覧表示する」
- 「8.3.5 テストケースをエクスポートする（テストケース指定）」
- 「8.5.3 実行結果を削除する」
- 「8.5.4 実行結果を一覧表示する」
- 「cbluts コマンド － バッチモードで単体テスト支援を実行する」

8.6.9 テストケースごとに実行環境ファイルまたは実行時環境変数を設定する

プロジェクト情報ファイルを編集して，テストケースごとに実行環境ファイルまたは実行時環境変数を設定できます。

作業の手順

1. 次のコマンドで，プロジェクト情報ファイルを出力します。

```
cbluts -TestProject -Export -OutputPath c:¥work¥setting.xml -Project sample.utsproj
```

2. プロジェクト情報ファイルを編集します。

コマンド実行時に，テストケースごとに実行環境ファイルまたは実行時環境変数を設定している場合

実行環境ファイルを指定するには，RuntimeEnvironmentFile 要素に実行環境ファイルのパスを，実行時環境変数を指定するには，RuntimeEnvironmentVariable 要素に変数名と値を指定します。

コマンド実行時に，テストケースごとに実行環境ファイルまたは実行時環境変数を設定していない場合

次の内容をコピーし，プロジェクト情報ファイルの</Stub>と</CblUnitTestSupport>の間に貼り付けます。

```
<TestCaseSettings>
<TestCaseSettig testcasename="">
<RuntimeEnvironmentFile></RuntimeEnvironmentFile>
<RuntimeEnvironmentVariables>
<RuntimeEnvironmentVariable name="" value="" />
</RuntimeEnvironmentVariables>
</TestCaseSettig>
</TestCaseSettings>
```

testcasename 属性に設定対象のテストケース名を指定してください。

実行環境ファイルを指定する場合は，RuntimeEnvironmentFile 要素に実行環境ファイルのパスを，実行時環境変数を指定する場合は，RuntimeEnvironmentVariable 要素に変数名と値を指定してください。

EnableTestCaseSetting 要素の値を OFF から ON に変更してください。

3. 次のコマンドで，プロジェクト情報ファイルを設定します。

```
cbluts -TestProject -Import -InputPath c:¥work¥setting.xml -Project sample.utsproj
```

注意事項

設定対象のテストケースがテストプロジェクトにまだ存在しない場合は，CSV ファイルのインポートによって，設定対象のテストケースを先に作成しておいてください。

関連項目

- 「8.4.2　プロジェクト情報ファイルを出力する」
- 「8.4.3　プロジェクト情報を設定する」
- 「8.6.3　プロジェクト情報を変更する」
- 「11.2　プロジェクト情報ファイルの構成要素」
- 「cbluts コマンド － バッチモードで単体テスト支援を実行する」

9 環境をカスタマイズする

単体テスト支援の環境をカスタマイズする手順について説明します。

9.1　テストプロジェクトを移動する

テストプロジェクトを別のフォルダに移動する手順です。

単体テスト支援では，テストプロジェクトは上書き保存だけができます。単体テスト支援上で別のフォルダに別名保存や移動はできません。テストプロジェクトをコピーして使用する場合や，ほかのフォルダで使用する場合は，テストプロジェクト作業フォルダごとコピーまたは移動してください。

コピーまたは移動によって，次に示すフォルダの場所を変更する場合は，移動後に単体テスト支援でテストプロジェクトを読み込んだあとで，［プロパティページ］でテストプロジェクトの設定を変更してください。

- テスト対象の COBOL ソースファイルの絶対パス名
- 登録集原文の検索対象パス
- 実行環境ファイルパス

作業の前に確認すること

- テストプロジェクトを移動前に，対象のテストプロジェクトを単体テスト支援で開いていない。

作業の手順

1. テストプロジェクト作業フォルダごと，移動先にファイル一式をコピーまたは移動します。
2. 単体テスト支援を起動し，［ファイル］－［テストプロジェクトを開く］メニューを選択，または［テストプロジェクトを開く］ボタンをクリックし，テストプロジェクトを開きます。
3. ［プロジェクト］－［プロパティページを開く］メニューを選択します。
   テストプロジェクトの［プロパティページ］が表示されます。
4. テストプロジェクトのプロパティの設定値を確認し，移動先の環境に合わせた設定に変更します。
   テストプロジェクトのプロパティの編集については，「10.6　［プロパティページ］」および「10.7　［スタブ設定ページ］」を参照してください。

注意

テストプロジェクトを移動したあとのパスにコンマ（,）が含まれていた場合，テストプロジェクトを開くことはできません。テストプロジェクトを開こうとした場合，エラーメッセージが表示されます。

9.2 テストケースを CSV ファイルにエクスポートしインポートする

単体テスト支援のテストケースを CSV ファイルにエクスポートして，インポートする手順です。

単体テスト支援のテストプロジェクトには，複数のテストケースを設定できます。複数のテストケースを効率良く作成/編集できるように，テストケースの情報を CSV ファイルにエクスポートし，編集後にインポートできます。CSV ファイルを任意の編集ツールで一括して編集すれば，単体テスト支援の GUI で逐一設定する手間が省けます。

テストケースの CSV ファイルへの出力は，次のように使用できます。

- テストケースの一部の設定値を変更し，大量のテストケースのバリエーションを作成
- 共通の登録集原文の定義がある別のテスト対象ソースのテストデータを利用してテストケースを作成

単体テスト支援でできるエクスポートおよびインポートの操作を次に示します。

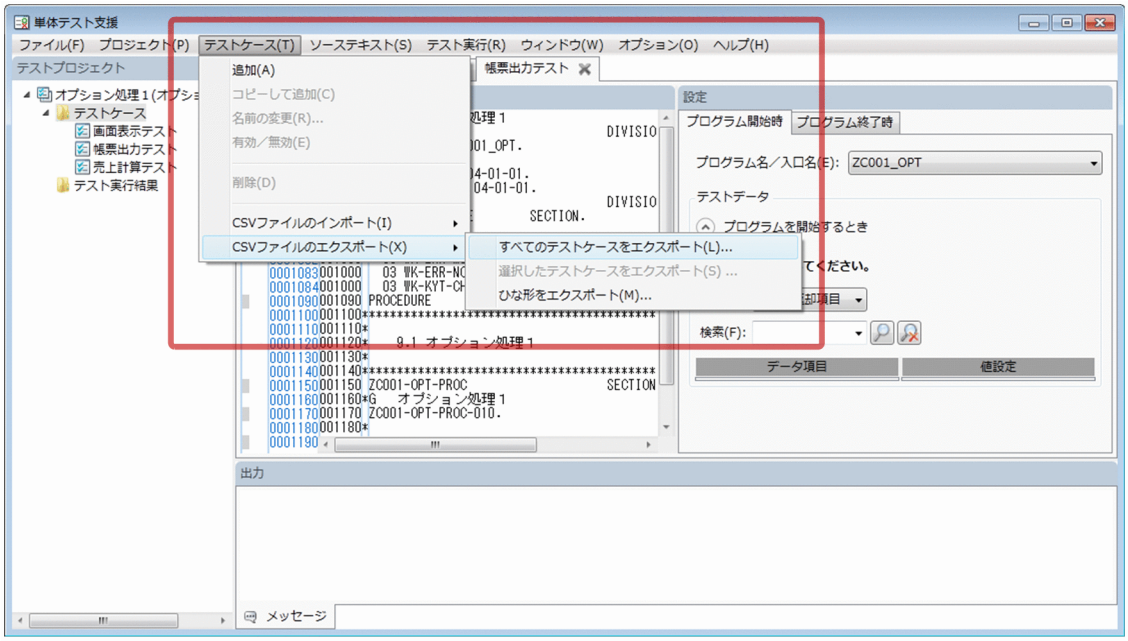
- すべてのテストケースを CSV ファイルにエクスポート
- 選択したテストケースを CSV ファイルにエクスポート
- ひな形のテストケースをエクスポート
- 指定したフォルダの直下にあるすべての CSV ファイルをテストケースとしてインポート
- CSV ファイルをテストケースとしてインポート

9.2.1 テストケースをエクスポートする（すべてのテストケースを CSV ファイルにエクスポートする場合）

すべてのテストケースを CSV ファイルにエクスポートする手順です。

作業の手順

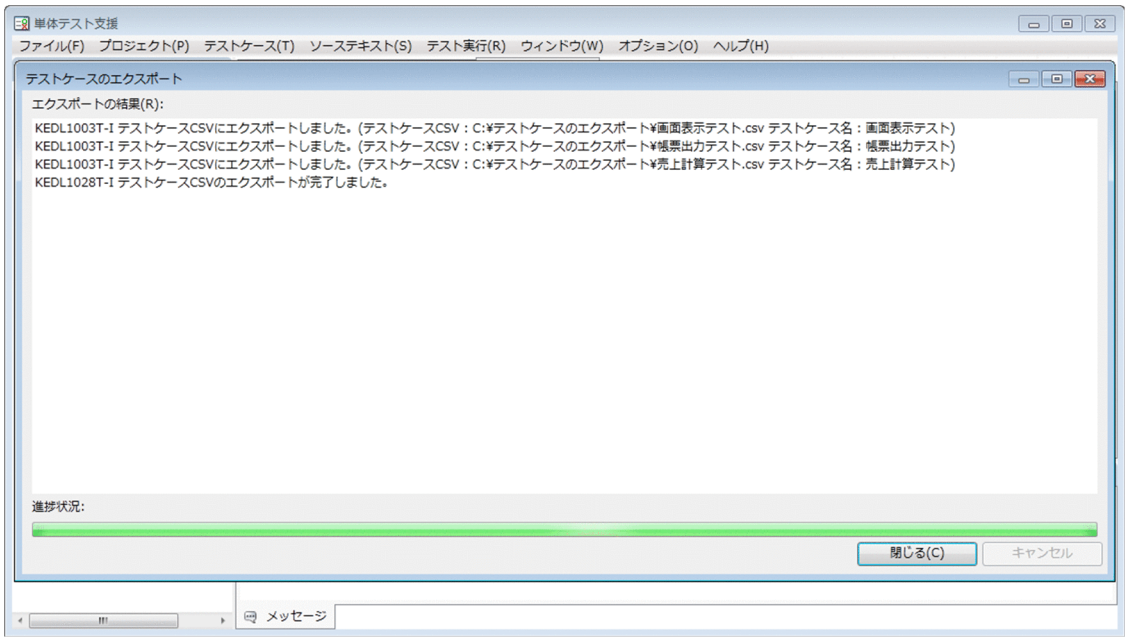
1. ［テストケース］メニューで［CSV ファイルのエクスポート］－［すべてのテストケースをエクスポート］を選択します。



［フォルダーの参照］ダイアログが表示されます。

2. ［フォルダーの参照］ダイアログに出力先のフォルダを指定して［OK］ボタンをクリックします。

エクスポートの状況を示すダイアログが表示されます。エクスポートで発生したエラーはメッセージ画面に出力されないで，代わりにこのダイアログの結果部分に出力されます。



指定したフォルダに CSV ファイルの出力が完了すると，そのことを通知するメッセージが出力されます。

3. エクスポートが完了したら［閉じる］ボタンをクリックします。

注意

複数のテストケースをエクスポートする場合，エクスポートするテストケース名と同じ CSV ファイルがあるときは，上書きするかどうかを確認するダイアログが表示されます。ボタンの動作を次に示します。

- ［上書きする］ボタン

  既存の CSV ファイルの内容を上書きします。既存の CSV ファイルの内容とはマージされません。元の CSV ファイルの内容はすべて削除されます。

- ［スキップする］ボタン

  該当のテストケースのエクスポートをスキップして，次のテストケースのエクスポートに移行します。

- ［キャンセル］ボタン

  残りのエクスポート処理をすべて中止します。［キャンセル］ボタンをクリックするまでに実行したエクスポートはキャンセルされません。

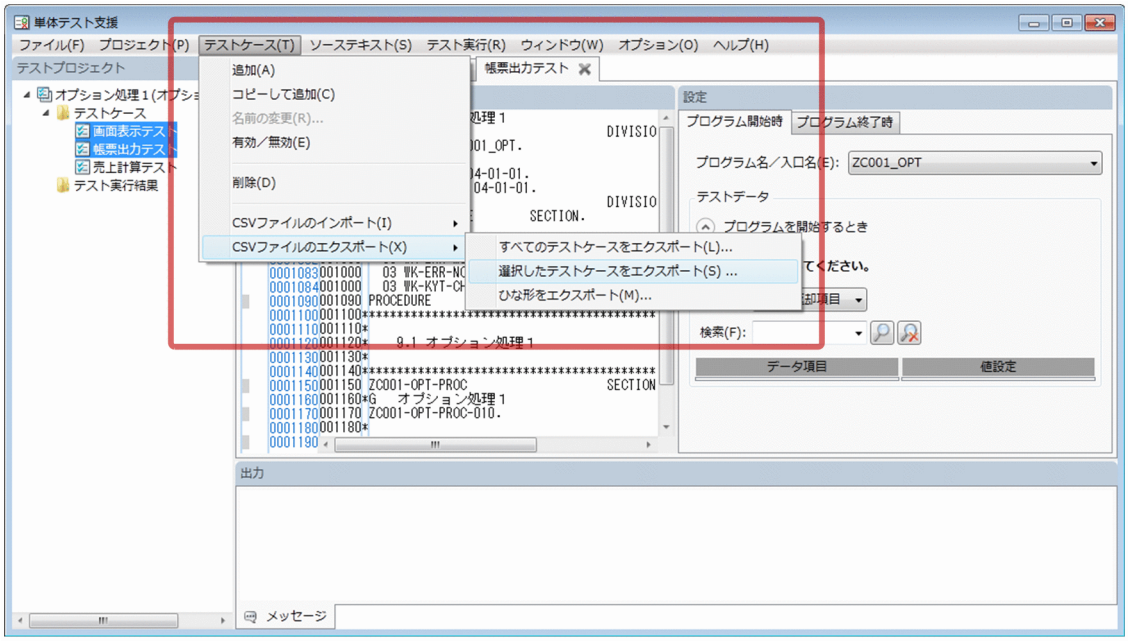
エクスポートの対象が複数の場合は，［すべてに適用する］チェックボックスが表示されます。チェックボックスをオンにすると，残りのエクスポートで名前が重複したときには，ダイアログで選んだ処理が適用されます。

9.2.2 テストケースをエクスポートする（選択したテストケースを CSV ファイルにエクスポートする場合）

選択したテストケースを CSV ファイルにエクスポートする手順です。

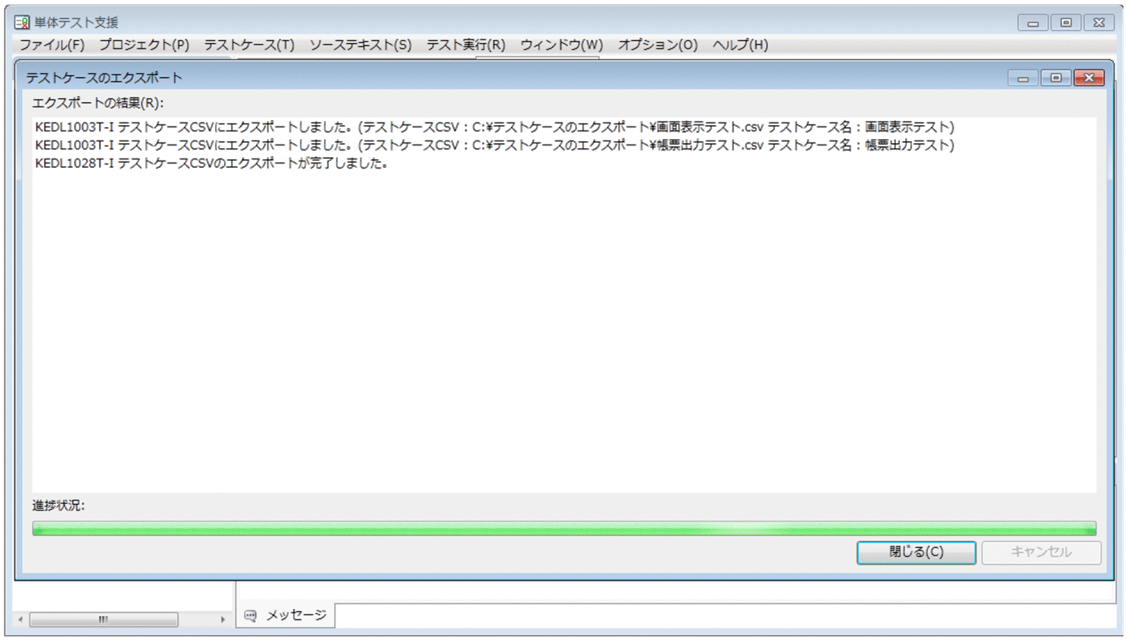
作業の手順

1. ［テストプロジェクト］画面で 1 つまたは複数の［テストケース］ノードを選択します。
2. ［テストケース］メニューで［CSV ファイルのエクスポート］－［選択したテストケースをエクスポート］を選択します。



3. 1.で選択した［テストケース］ノードが 1 つの場合は，［テストケースのエクスポート］ダイアログが表示されるので，CSV ファイル名と出力先を指定します。複数選択していた場合は，［フォルダーの参照］ダイアログが表示されるので，出力するフォルダを指定します。

   エクスポートの状況を示すダイアログが表示されます。エクスポートで発生したエラーはメッセージ画面に出力されないで，代わりにこのダイアログの結果部分に出力されます。



   指定したフォルダに CSV ファイルの出力が完了すると，そのことを通知するメッセージが出力されます。

4. エクスポートが完了したら［閉じる］ボタンをクリックします。

注意

複数のテストケースをエクスポートする場合，エクスポートするテストケース名と同じ CSV ファイルがあるときは，上書きするかどうかを確認するダイアログが表示されます。ボタンの動作を次に示します。

- ［上書きする］ボタン

  既存の CSV ファイルの内容を上書きします。既存の CSV ファイルの内容とはマージされません。元の CSV ファイルの内容はすべて削除されます。

- ［スキップする］ボタン

  該当のテストケースのエクスポートをスキップして，次のテストケースのエクスポートに移行します。

- ［キャンセル］ボタン

  残りのエクスポート処理をすべて中止します。［キャンセル］ボタンをクリックするまでに実行したエクスポートはキャンセルされません。

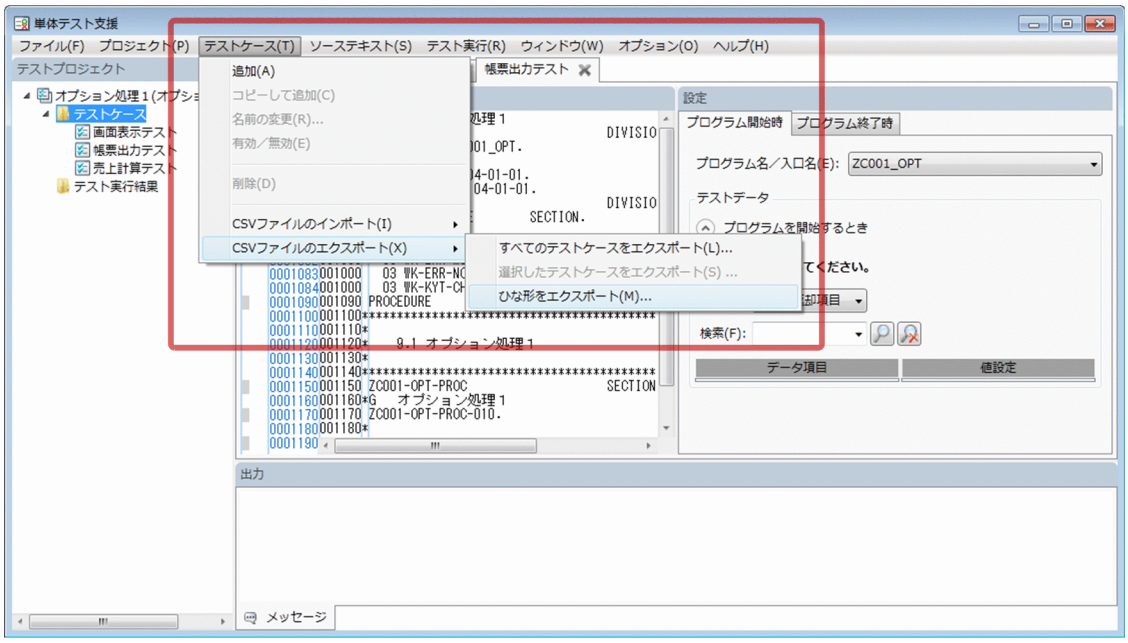
エクスポートの対象が複数の場合は，［すべてに適用する］チェックボックスが表示されます。チェックボックスをオンにすると，残りのエクスポートで名前が重複したときには，ダイアログで選んだ処理が適用されます。

9.2.3 テストケースをエクスポートする（ひな形のテストケースをエクスポートする場合）

ひな形のテストケースを CSV ファイルにエクスポートする手順です。

作業の手順

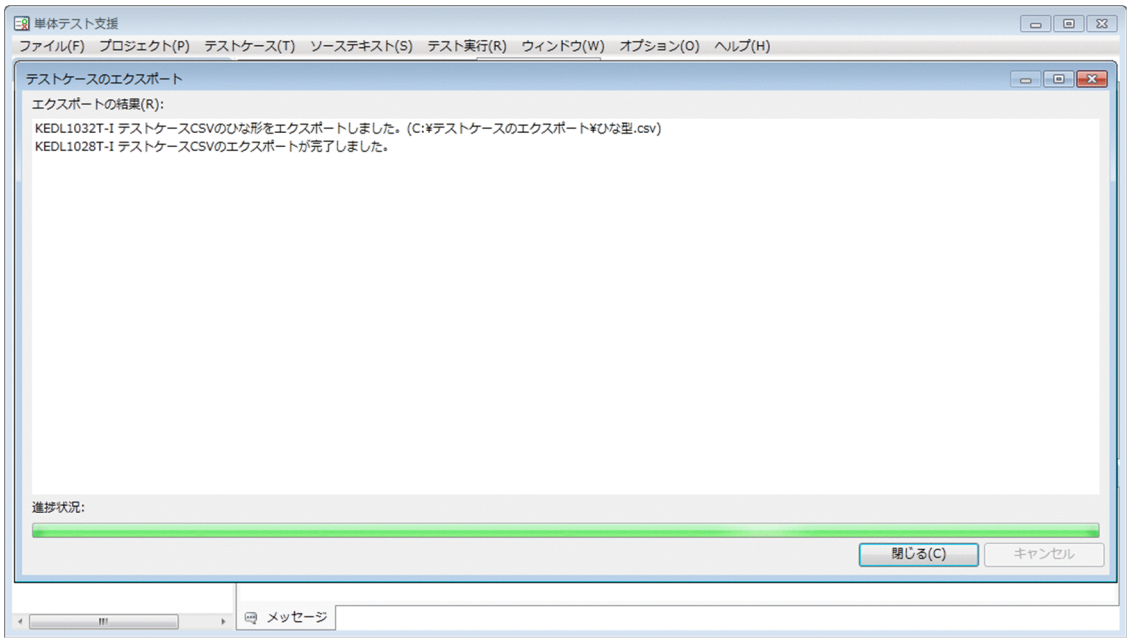
1. ［テストケース］メニューで［CSV ファイルのエクスポート］－［ひな形をエクスポート］を選択します。



2. ［ひな形をエクスポート］ダイアログが表示されるので，CSV ファイル名と出力先を指定します。

   ファイル名は，デフォルトで「ひな形.csv」と表示されます。

   エクスポートの状況を示すダイアログが表示されます。エクスポートで発生したエラーはメッセージ画面に出力されないで，代わりにこのダイアログの結果部分に出力されます。



   指定したフォルダに CSV ファイルの出力が完了すると，そのことを通知するメッセージが出力されます。

3. エクスポートが完了したら［閉じる］ボタンをクリックします。

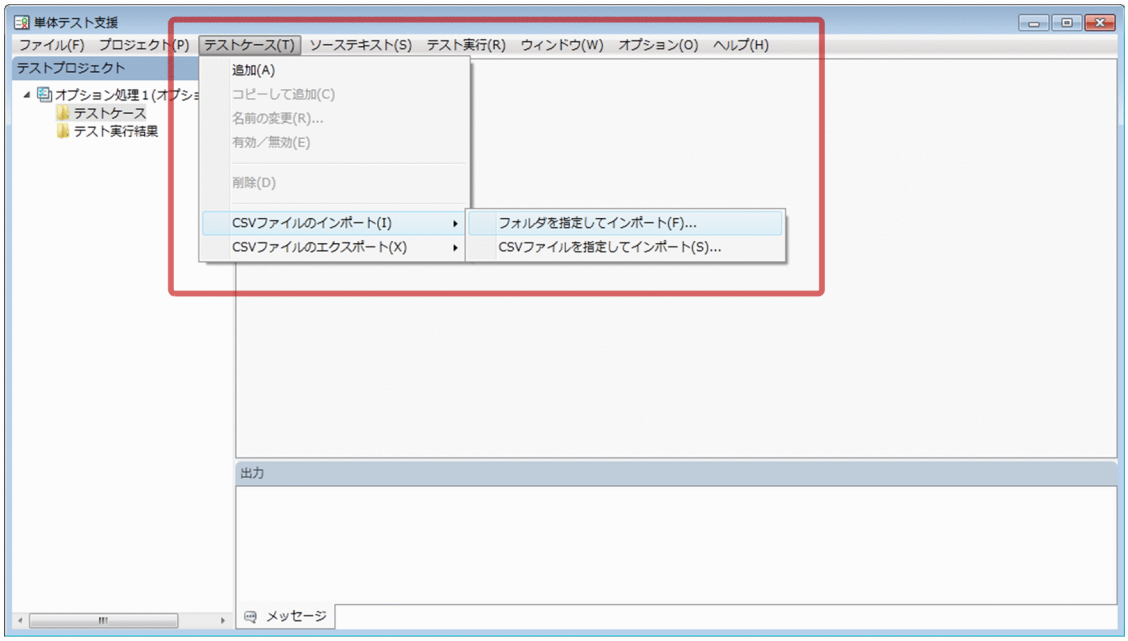
9.2.4 テストケースをインポートする（指定したフォルダの直下にあるすべての CSV ファイルをテストケースとしてインポートする場合）

指定したフォルダの直下にあるすべての CSV ファイルをテストケースとしてインポートする手順です。

インポートでは，CSV ファイルの拡張子を除いたファイル名がテストケース名になります。

作業の手順

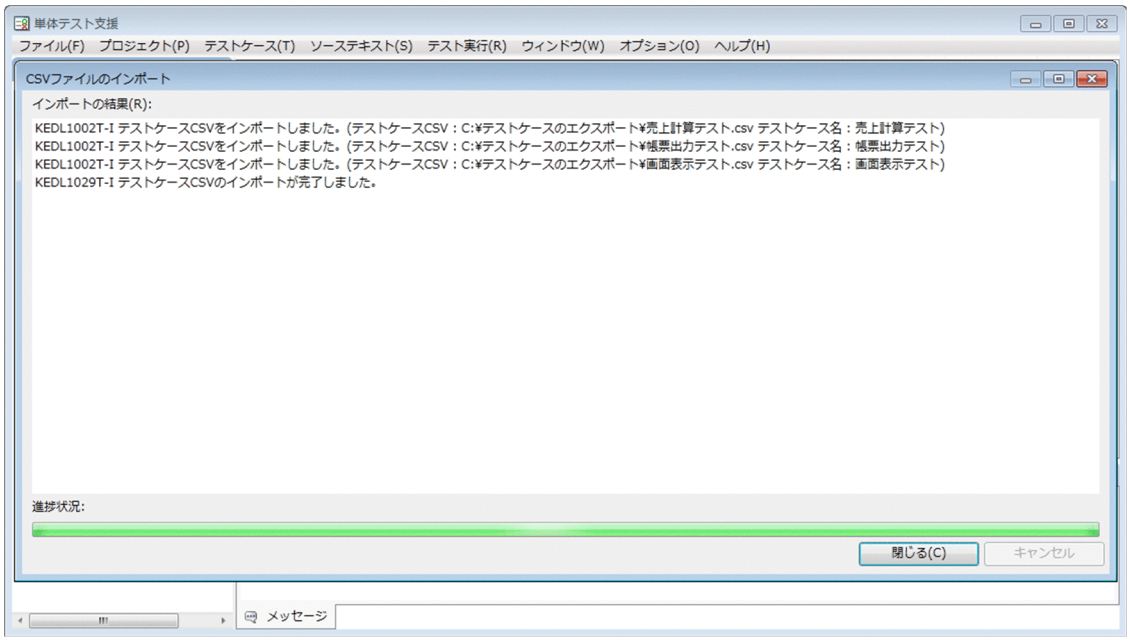
1. ［テストケース］メニューで［CSV ファイルのインポート］－［フォルダを指定してインポート］を選択します。



［フォルダーの参照］ダイアログが表示されます。

2. ［フォルダーの参照］ダイアログに入力元のフォルダを指定して［OK］ボタンをクリックします。

インポートの状況を示すダイアログが表示されます。インポートで発生したエラーはメッセージ画面に出力されないで，代わりにこのダイアログの結果部分に出力されます。



指定した CSV ファイルの内容が単体テスト支援のテストケースとしてインポートされると，そのことを通知するメッセージが出力されます。

3. インポートが完了したら［閉じる］ボタンをクリックします。

注意

インポートする CSV ファイル名と同じテストケースがある場合は，上書きするかどうかを確認するダイアログが表示されます。ボタンの動作を次に示します。

- ［上書きする］ボタン

  既存のテストケースの内容を上書きします。既存のテストケースの内容にはマージされません。元のテストケースの内容はすべて削除されます。ただし，インポート先のテストケースで，テストケースごとに実行環境ファイル，実行時環境変数を設定している場合※，インポートで上書きをしても，実行環境ファイル，実行時環境変数の情報は更新されません。テストケースごとの設定が有効となります。そのため，実行環境ファイル，実行時環境変数の設定を変更する場合は，［テスト環境の設定］画面で設定する必要があります。

注※

［テストケースごとに実行環境ファイルまたは実行時環境変数を設定する］チェックボックスをオフにしても，テストプロジェクトを保存するまでは，テストケースごとの実行環境ファイルまたは実行時環境変数の設定は保持されるため，実行環境ファイル，実行時環境変数を設定しているとみなされます。

- ［別名で取り込む］ボタン

  自動で名前を変更して，変更後の名前でテストケースを作成します。名前を変更するルールは，同じ名前のテストケースを追加する場合と同じく，自動的にテストケースの名前の末尾に通し番号が付けられます。

- ［スキップする］ボタン

  該当の CSV ファイルのインポートをスキップして，次の CSV ファイルのインポートに移行します。1 つの CSV ファイルをインポートする場合は，［スキップする］ボタンは表示されません。

- ［キャンセル］ボタン

  残りのインポート処理をすべて中止します。［キャンセル］ボタンをクリックするまでに実行されたインポートはキャンセルされません。

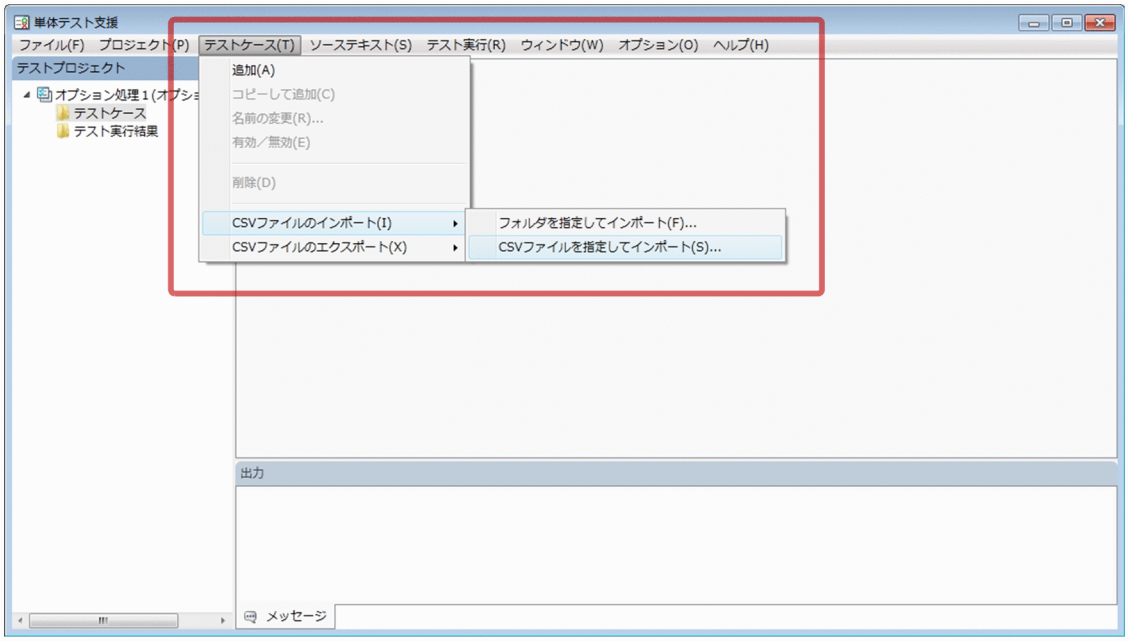
インポートの対象が複数の場合は，［すべてに適用する］チェックボックスが表示されます。チェックボックスをオンにすると，残りのインポートで名前が重複したときには，ダイアログで選んだ処理が適用されます。

9.2.5 テストケースをインポートする（CSV ファイルをテストケースとしてインポートする場合）

CSV ファイルをテストケースとしてインポートする手順です。

作業の手順

1. ［テストケース］メニューで［CSV ファイルのインポート］－［CSV ファイルを指定してインポート］を選択します。

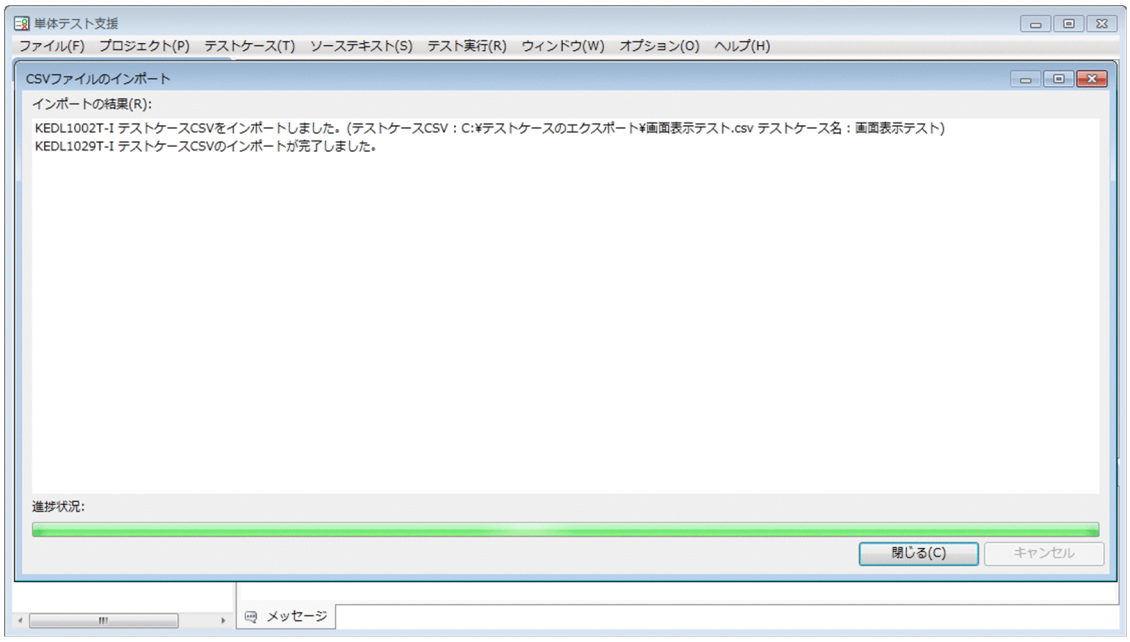


［テストケースのインポート］ダイアログが表示されます。

2. ［テストケースのインポート］ダイアログに入力元の CSV ファイルを指定して［OK］ボタンをクリックします。

   CSV ファイルは複数指定できます。

   インポートの状況を示すダイアログが表示されます。インポートで発生したエラーはメッセージ画面に出力されないで，代わりにこのダイアログの結果部分に出力されます。



   指定した CSV ファイルの内容が単体テスト支援のテストケースとしてインポートされると，そのことを通知するメッセージが出力されます。

3. インポートが完了したら［閉じる］ボタンをクリックします。

注意

複数の CSV ファイルをインポートする場合，インポートする CSV ファイル名と同じテストケースがあるときは，上書きするかどうかを確認するダイアログが表示されます。ボタンの動作を次に示します。

- ［上書きする］ボタン

  既存のテストケースの内容を上書きします。既存のテストケースの内容にはマージされません。元のテストケースの内容はすべて削除されます。ただし，インポート先のテストケースで，テストケースごとに実行環境ファイル，実行時環境変数を設定している場合※，インポートで上書きをしても，実行環境ファイル，実行時環境変数の情報は更新されません。テストケースごとの設定が有効となります。そのため，実行環境ファイル，実行時環境変数の設定を変更する場合は，［テスト環境の設定］画面で設定する必要があります。

注※

［テストケースごとに実行環境ファイルまたは実行時環境変数を設定する］チェックボックスをオフにしても，テストプロジェクトを保存するまでは，テストケースごとの実行環境ファイルまたは実行時環境変数の設定は保持されるため，実行環境ファイル，実行時環境変数を設定しているとみなされます。

- ［別名で取り込む］ボタン

  自動で名前を変更して，変更後の名前でテストケースを作成します。名前を変更するルールは，同じ名前のテストケースを追加する場合と同じく，自動的にテストケースの名前の末尾に通し番号が付けられます。

- ［スキップする］ボタン

  該当の CSV ファイルのインポートをスキップして，次の CSV ファイルのインポートに移行します。

- ［キャンセル］ボタン

  残りのインポート処理をすべて中止します。［キャンセル］ボタンをクリックするまでに実行されたインポートはキャンセルされません。

インポートの対象が複数の場合は，［すべてに適用する］チェックボックスが表示されます。チェックボックスをオンにすると，残りのインポートで名前が重複したときには，ダイアログで選んだ処理が適用されます。

9.3　単体テスト支援の画面を変更する

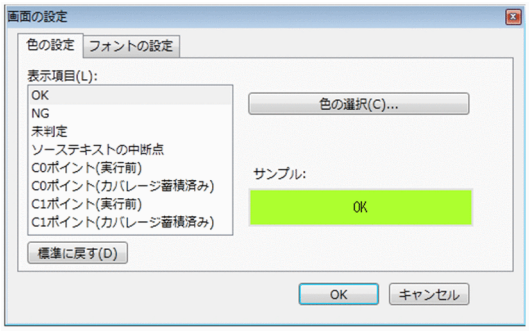
単体テスト支援の画面をカスタマイズする手順です。

9.3.1　色を変更する

単体テスト支援の画面で表示している項目の色を変更する手順です。

作業の手順

1. ［オプション］－［画面の設定］メニューを選択します。
   ［画面の設定］ダイアログが表示されます。



2. ［画面の設定］ダイアログの［色の設定］タブを選びます。

3. ［表示項目］から変更する項目を選択し［色の選択］ボタンをクリックし，色を選択します。

4. ［OK］ボタンをクリックします。
   画面の配色が変更されます。
   変更される項目

画面	対象	表示項目
テストプロジェクト結果レポート	次の項目の「表示項目」の文字列 • テスト結果サマリ • 結果一覧	OK
		NG
		未判定
テストケース結果レポート	次の項目の「表示項目」の文字列 • 判定結果 • データ項目の結果値と期待値の「判定」	OK
		NG
		未判定
ソーステキスト	中断点が設定された文キーワードを囲む枠の色	ソーステキストの中断点
	カバレージ蓄積前の C0 列の色	C0 ポイント（実行前）
	カバレージ蓄積後の C0 列の色	C0 ポイント（カバレージ蓄積済み）

画面	対象	表示項目
	カバレージ蓄積前の C1 列の色	C1 ポイント（実行前）
	カバレージ蓄積後の C1 列の色	C1 ポイント（カバレージ蓄積済み）

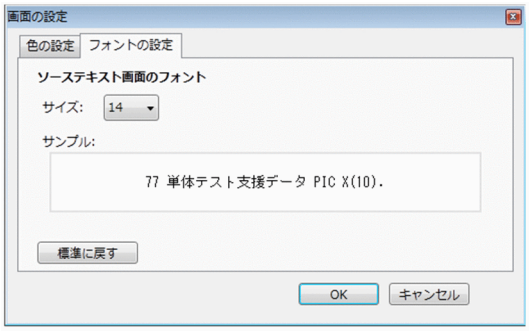
9.3.2　フォントを変更する

単体テスト支援の COBOL ソーステキストのフォントサイズを変更する手順です。

作業の手順

1. ［オプション］－［画面の設定］メニューを選択します。
   ［画面の設定］ダイアログが表示されます。

2. ［画面の設定］ダイアログの［フォントの設定］タブを選びます。



3. ［サイズ］で文字の大きさを選択します。

4. ［OK］ボタンをクリックします。
   フォントサイズが変更されます。

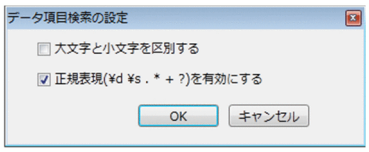
9.4　データ項目検索の設定を変更する

［プログラム開始時］画面，［プログラム終了時］画面，または［中断点］画面で，データ項目を検索するときの条件を指定する手順です。

作業の手順

1. ［オプション］－［データ項目検索の設定］メニューを選択します。

   ［データ項目検索の設定］ダイアログが表示されます。



2. データ項目の検索で英大文字と英小文字を区別するときは，［大文字と小文字を区別する］チェックボックスをオンにします。データ項目の検索で正規表現を利用するときは，［正規表現（¥d ¥s . * ＋ ?）を有効にする］チェックボックスをオンにします。

3. ［OK］ボタンをクリックします。

   次回のテストケースの設定時にデータ項目を検索するときから，設定した条件が有効になります。

関連項目

- 「付録 B　単体テスト支援で使用できる正規表現」

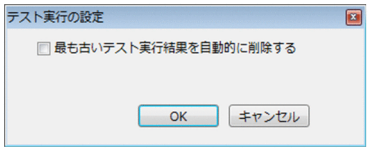
9.5　テスト実行の設定を変更する

テスト実行後にテスト実行結果が上限値を超える場合，最も古いテスト実行結果を自動的に削除するかどうかを設定する手順です。

作業の手順

1. ［オプション］－［テスト実行の設定］メニューを選択します。

   ［テスト実行の設定］ダイアログが表示されます。



2. 最も古いテスト実行結果を自動的に削除する場合は，［最も古いテスト実行結果を自動的に削除する］チェックボックスをオンにする。

3. ［OK］ボタンをクリックします。

   次回のテスト実行時から，設定した条件が有効になります。

10 画面のリファレンス

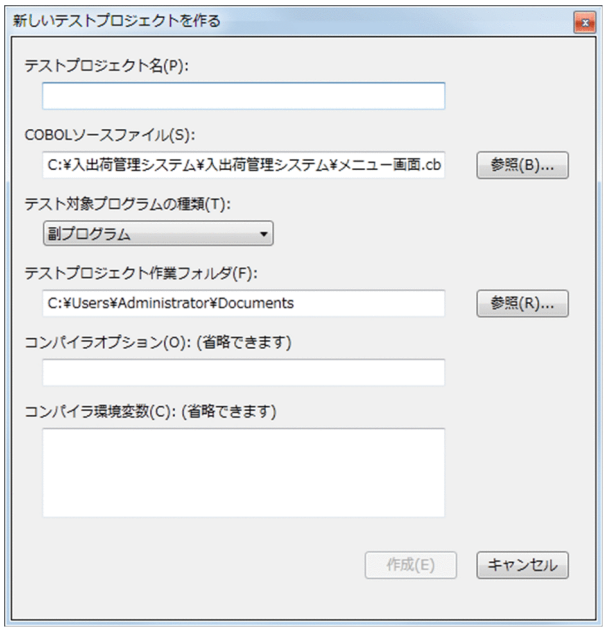
単体テスト支援の画面について説明します。

10.1 ［新しいテストプロジェクトを作る］ダイアログ

新規でテストプロジェクトを作成するときに表示されるダイアログです。

［新しいテストプロジェクトを作る］ダイアログは，次に示す操作で表示されます。

- ［ファイル］－［新しいテストプロジェクトを作る］を選択
- ［スタートページ］の［新しいテストプロジェクトを作る］ボタンをクリック



［テストプロジェクト名］

テストプロジェクトの名前を指定します。

［COBOL ソースファイル］

テストの対象とする COBOL ソースファイルの絶対パス名を指定します。

［テスト対象プログラムの種類］

テスト対象プログラムの呼び出し方法を次のどれかから選択します。

- 主プログラム（-Main,System）
- 主プログラム（-Main,V3）
- 副プログラム

［テストプロジェクト作業フォルダ］

テストプロジェクトを作成するフォルダの絶対パス名を指定します。

［コンパイラオプション］

テスト対象の COBOL ソースファイルをコンパイルするときのオプションを指定します。複数のオプションを指定する場合は，空白で区切ります。

［コンパイラ環境変数］

テスト対象の COBOL ソースファイルをコンパイルするときの環境変数を指定します。

環境変数は名前と値の間にイコール（=）を入れて指定します。複数の環境変数を指定するときは，改行で区切ります。次の場合は，何も設定されません。

- 空行を指定した場合
- イコール（=）を指定しなかった場合
- イコール（=）の前に何も指定しなかった場合

イコール（=）の後ろに何も指定しなかった場合は，空の値が設定されます。

同じ名前の環境変数を複数指定したときは，最後に指定した値が有効になります。

環境変数 CBLXXX，CBLYYY，CBLZZZ にそれぞれ値を設定する例を次に示します。

```
CBLXXX=DATA1
CBLYYY=DATA2
CBLZZZ=DATA3
```

テスト対象が COPY 文で登録集原文を複写している場合は，環境変数 CBLLIB を必ず指定してください。

［作成］ボタン

クリックすると，設定した内容でテストプロジェクトが作成されます。

［キャンセル］ボタン

クリックすると，［新しいテストプロジェクトを作る］ダイアログが閉じられます。

注意

- 1 つのテストプロジェクト作業フォルダに 2 つ以上のテストプロジェクトファイルを格納できません。テストプロジェクト作業フォルダは，1 つのテストプロジェクトに 1 つになるように指定してください。すでにテストプロジェクトファイルが存在するフォルダをテストプロジェクト作業フォルダに指定した場合はエラーになります。
- 指定したテストプロジェクト作業フォルダが存在しない場合は，作成されます。
- テストプロジェクトを作成すると，COBOL ソースファイルが解析されます。解析でエラーになると，テストケースは作成されません。［メッセージ］画面でエラーメッセージを確認し，エラーの個所を修正してください。
  - COBOL ソースファイルの内容に問題があるときは，［ソーステキスト］メニューの［COBOL エディタを開く］を選択して，COBOL エディタを開きます。COBOL ソースファイルを修正して［プロジェクト］メニューの［COBOL ソースファイルの解析］を選択します。
  - コンパイラオプションまたはコンパイラ環境変数に問題があるときは，［プロジェクト］－［プロパティページを表示］メニューを選択し，［プロパティページ］を開きます。［プロパティページ］の［変更］ボタンをクリックし，［COBOL ソースファイルの設定］ダイアログを表示して修正したあと，［COBOL ソースファイルの解析］ボタンをクリックします。

- テスト対象の COBOL ソースファイルに複数の翻訳単位が含まれている場合，テストプロジェクトの生成後にテスト対象のプログラムを選択するダイアログが表示されます。単体テストを実行したいプログラムを選択できるのはテストプロジェクト生成直後だけなので，別のプログラムを単体テストの対象にする場合には，新しくテストプロジェクトを作成してください。
- コンパイラオプションに指定できるのは，ccbl2002 コマンドのコンパイラオプションだけです。ファイル名は，コンパイラオプションの引数としてだけ指定できます。オプションの引数でなく，コンパイルの対象としてファイル名を指定した場合の動作は保証しません。
- コンパイラ環境変数 CBLPIDIR は指定しないでください。指定した場合の動作は保証しません。
- ［新しいテストプロジェクトを作る］ダイアログで指定するコンパイラオプションとコンパイラ環境変数は，COBOL ソースファイルの解析で使用します。COBOL ソースファイルの解析では，自動的にコンパイラオプションまたはコンパイラ環境変数を追加することがあります。追加するコンパイラオプションについては，「5.2　COBOL ソースファイルを解析する」を参照してください。
- テストプロジェクト名には，次に示す文字は使用できません。使用するとエラーになって，テストプロジェクトは作成されません。

  「/」，「>」，「<」，「?」，「:」，「"」，「¥」，「*」，「|」
- 次の個所で設定した環境変数の合計値が 65,535 バイトを超えると，新しくテストプロジェクトを作るときに論理エラーになります。回避するには，不要な環境変数を削除してください。
  - ［コンパイラ環境変数］テキストボックス
  - システムに設定した環境変数
- テストプロジェクト名に「-」（ハイフン）で始まる名前を指定しないでください。指定した場合，テスト実行時にエラーになります。エラーになった場合，テストプロジェクトを閉じ，テストプロジェクトファイルのファイル名を「-」（ハイフン）で始まらない名前に変更してからテストプロジェクトを開き直してください。

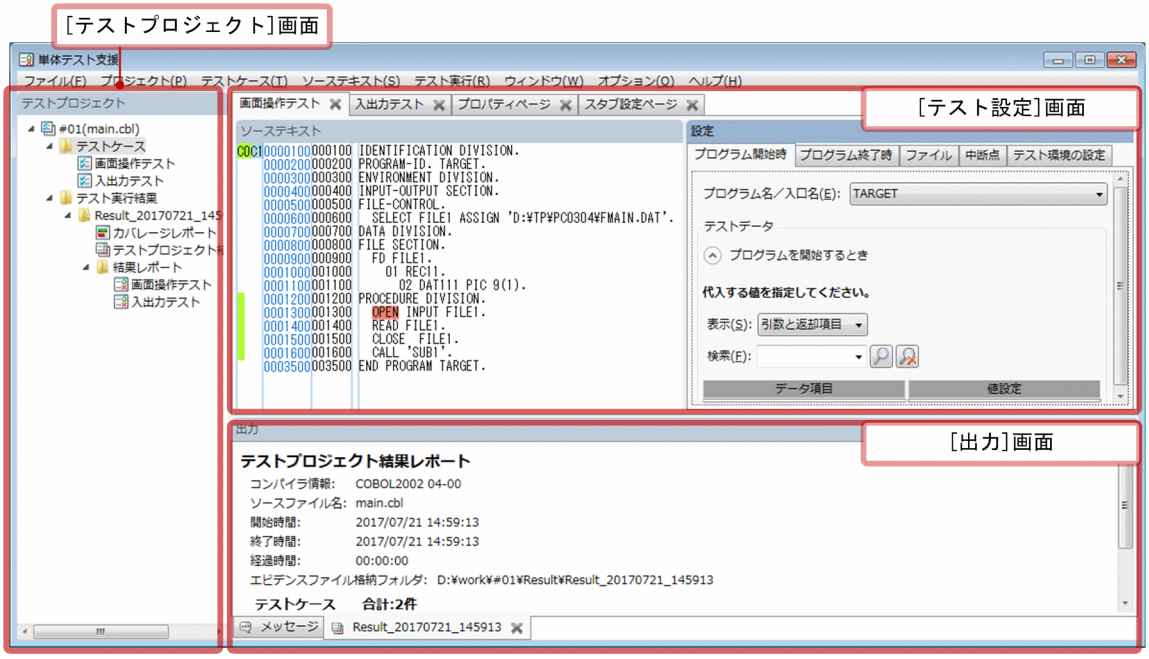
10.2　単体テスト支援のメインウィンドウ

単体テスト支援を操作するメインウィンドウです。

次に示す操作をすると，各画面にテストプロジェクトの内容が表示されます。

- 新規にテストプロジェクトを作成
- テストプロジェクトを開く

通常の画面



[テストプロジェクト］画面

　単体テスト支援の各画面を操作する，ツリー構造の画面です。

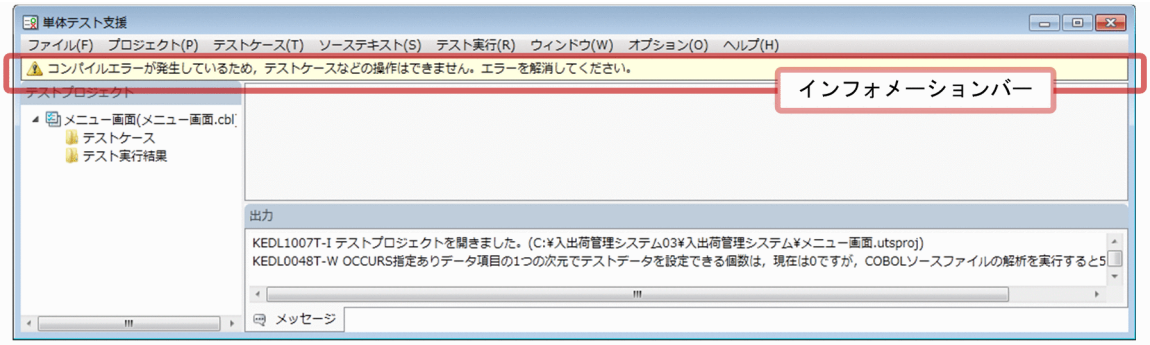
[テスト設定］画面

　単体テストの条件を設定する画面です。COBOL ソースファイルのテキスト，［プログラム開始時］画面，［プログラム終了時］画面，［ファイルシミュレーション］画面，［中断点］画面から構成されます。

[出力］画面

　単体テスト支援の実行で出力されるメッセージと，テスト実行結果の各種レポートが表示されます。表示項目は，画面下のタブで切り替えます。

重大な問題が発生しているときの画面



インフォメーションバー

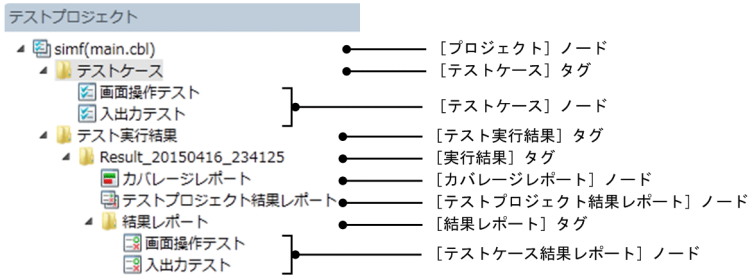
単体テスト支援の操作で，特に注意が必要な状態のときに表示されます。

出力されるメッセージに従って対処してください。

10.3 ［テストプロジェクト］画面

単体テスト支援の各画面を操作するための画面です。

■［テストプロジェクト］画面の詳細



［プロジェクト］ノード

テストプロジェクト名とテスト対象の COBOL ソースファイルです。

［テストケース］タグ

テストケースをまとめるタグです。

［テストケース］ノード

テストプロジェクトに属するテストケースです。ダブルクリックすると，画面右上の［テスト設定］画面にテストケースが表示されます。テストケース名は変更できます。

［テスト実行結果］タグ

［実行結果］タグをまとめるタグです。

［実行結果］タグ

単体テストの実行結果の各種レポートをまとめるタグです。単体テストを実行すると，「Result_*yyyymmdd*_*hhmmss*」（*yyyy*：西暦年，*mm*：月，*dd*：日，*hh*：時，*mm*：分，*ss*：秒）の名前の実行結果が生成されます。この名称は変更できます。

［カバレージレポート］ノード

実行結果のカバレージレポートです。ダブルクリックすると，画面右下の［出力］画面にカバレージレポートが表示されます。

［テストプロジェクト結果レポート］ノード

実行結果のテストプロジェクト結果レポートです。ダブルクリックすると，画面右下の［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。

［結果レポート］タグ

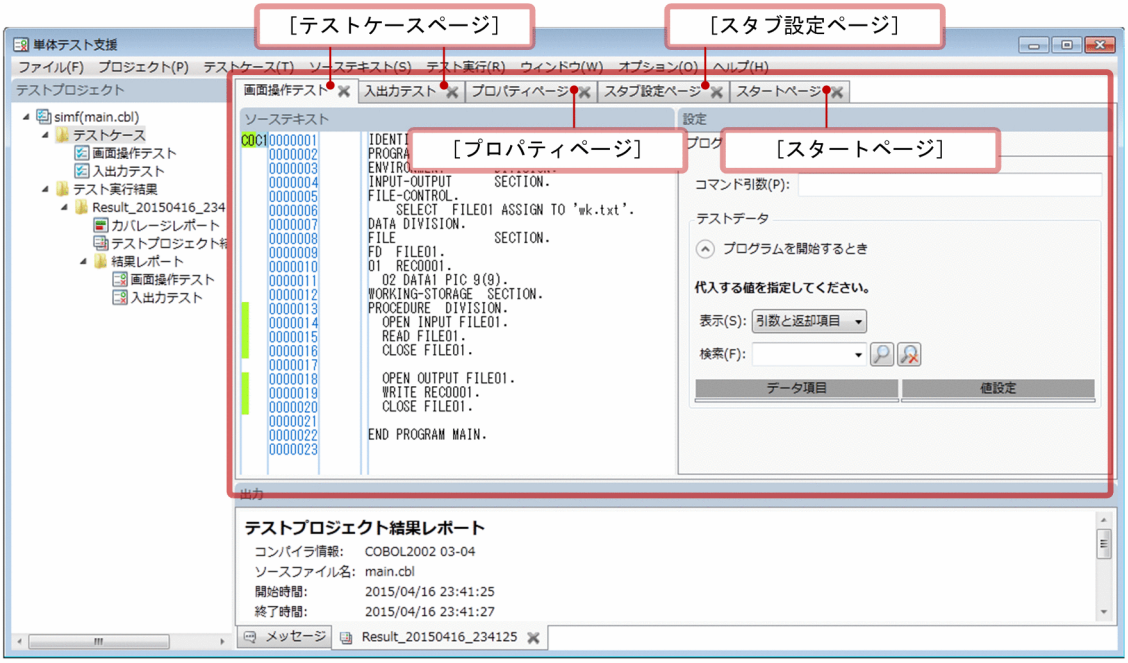
テストケースの結果レポートをまとめるタグです。

［テストケース結果レポート］ノード

テストケース結果レポートです。ダブルクリックすると，画面右下の［出力］画面にテストケース結果レポートが表示されます。

10.4 ［テスト設定］画面

単体テストを実行するためのテスト条件を設定する画面です。



次の画面から構成されます。

- ［スタートページ］
- ［プロパティページ］
- ［スタブ設定ページ］
- ［テストケースページ］

関連項目

- 「10.5　［スタートページ］」
- 「10.6　［プロパティページ］」
- 「10.7　［スタブ設定ページ］」
- 「10.8　［テストケースページ］」

10.5 ［スタートページ］

単体テスト支援を最初に使うときに表示される画面です。

［スタートページ］が表示されていないときは，［ウィンドウ］－［スタートページを表示］メニューを選択すると表示されます。

［新しいテストプロジェクトを作る］ボタン

　クリックすると，［新しいテストプロジェクトを作る］ダイアログが表示されます。

［テストプロジェクトを開く］ボタン

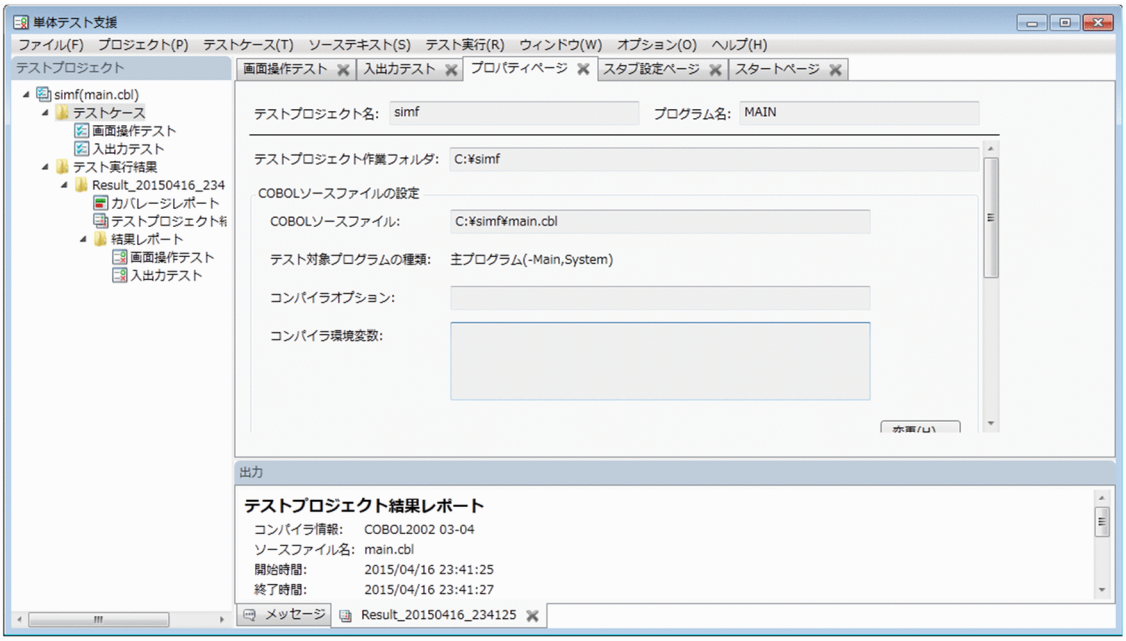
　クリックすると，［テストプロジェクトを開く］ダイアログが表示されます。

10.6 ［プロパティページ］

テストプロジェクトのプロパティを設定する画面です。

［プロジェクト］－［プロパティページを表示］メニューを選択すると表示されます。

COBOL ソースファイルの設定に関する項目を変更するときは，［プロパティページ］の［変更］ボタンをクリックし［COBOL ソースファイルの設定］ダイアログを表示します。



COBOL ソースファイルの設定

［COBOL ソースファイル］

テスト対象とする COBOL ソースファイルの絶対パスです。

［テスト対象プログラムの種類］

テスト対象プログラムの呼び出し方法を次のどれかから選択します。

- 主プログラム（-Main,System）
- 主プログラム（-Main,V3）
- 副プログラム

［コンパイラオプション］

テスト対象の COBOL ソースファイルをコンパイルするときのオプションです。複数のオプションを指定するときは，空白で区切ります。

［コンパイラ環境変数］

テスト対象の COBOL ソースファイルをコンパイルするときの環境変数です。

環境変数は名前と値の間にイコール（=）を入れて指定します。複数の環境変数を指定するときは，改行で区切ります。空行を指定，またはイコール（=）を指定しなかった場合は，何も設定されません。イコール（=）の後ろに何も指定しなかった場合は，空の値が設定されます。

同じ名前の環境変数を複数指定したときは，最後に指定した値が有効になります。

環境変数 CBLXXX，CBLYYY，CBLZZZ にそれぞれ値を設定する例を次に示します。

```
CBLXXX=DATA1
CBLYYY=DATA2
CBLZZZ=DATA3
```

テスト対象が COPY 文で登録集原文を複写している場合は，環境変数 CBLLIB を必ず指定してください。

実行環境

［リンクするファイル］

テスト対象プログラムにリンクするオブジェクトファイル（.obj）とライブラリファイル（.lib）の絶対パスを指定します。複数のファイルを指定するときは，改行で区切ります。

［実行環境ファイル］

テスト実行で利用する実行環境ファイルの絶対パスです。単体テスト支援が単体テストを実行する前に，毎回テストモジュールフォルダに実行環境ファイルをコピーします。

［実行時環境変数］

テスト実行するときの実行時環境変数を指定します。

［テストケースごとに実行環境ファイルまたは実行時環境変数を設定する］

チェックボックスをオンにすると，テストケースごとに実行環境ファイルまたは実行時環境変数を設定できる［テスト環境の設定］画面が表示されます。

テストケースごとに実行環境ファイルまたは実行時環境変数を設定する例を次に示します。

- ファイルシミュレーションではなく実体を使うときに，テストケースごとに入力／出力するファイルを切り替えるために，テストケースごとに環境変数 CBL_外部装置名を指定する。
- テスト実行時のログをテストケースごとに取得するために，テストケースごとに環境変数 CBL_SYSERR の値を変える。

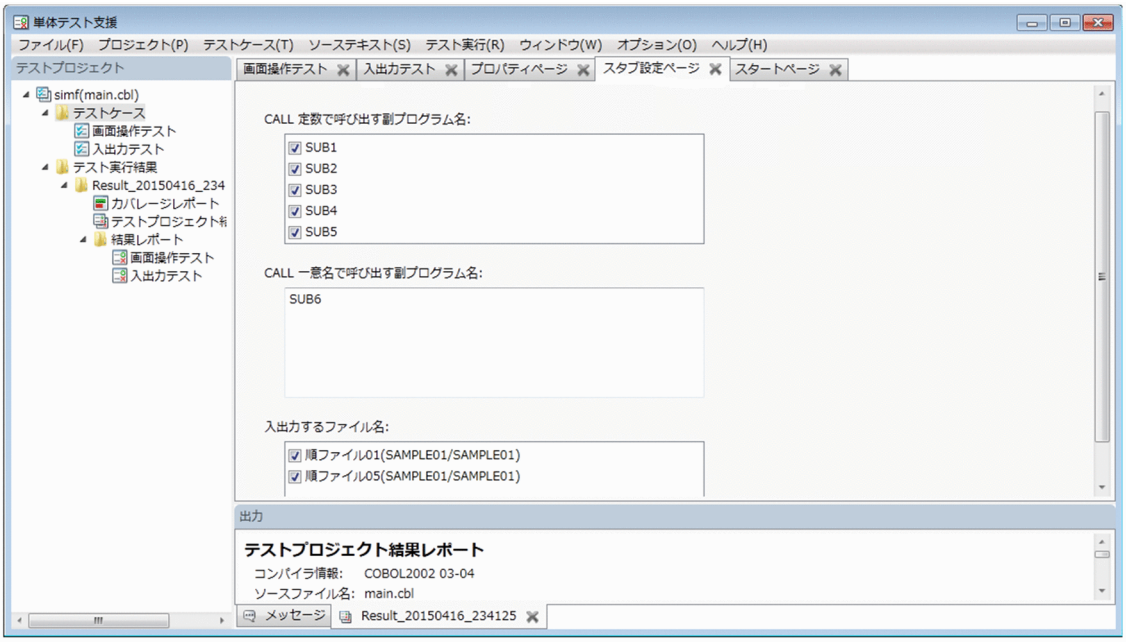
10.7 ［スタブ設定ページ］

単体テストを実行するときに必要になるスタブ（ダミーの副プログラムまたはファイル）を設定する画面です。

［プロジェクト］－［スタブ設定ページを表示］メニューを選択すると表示されます。

ダミーの副プログラムまたはファイルに対する CALL 文や入出力文の前後に中断点を設定すると，引数・レコードなどに対する値の設定や表示もできます。

単体テストでコンパイルエラーになったときは，単体テストは実行されません。このとき，［スタブ設定ページ］は操作できません。



［CALL 定数で呼び出す副プログラム名］

テスト対象の COBOL プログラムが呼び出す副プログラム名の一覧が表示されます。ダミーの副プログラムを作成する場合は，チェックボックスをオンにします。呼び出す副プログラムの実体がある場合はオフにします。

［CALL 一意名で呼び出す副プログラム名］

テスト対象の COBOL プログラムが一意名で呼び出す副プログラム名を入力する領域です。ダミーの副プログラムを作成する場合は，プログラム名を指定します。プログラム名は，1 行に 1 つの名前を入力します。

呼び出す副プログラムの実体がある場合はプログラム名を指定しないでください。

［入出力するファイル名］

テスト対象の COBOL プログラムが呼び出すファイル名と，定義されたプログラム名の一覧が，次の形式で表示されます。

ファイル名（最外側のプログラム名/最外側のプログラム名または内側のプログラム名）

ダミーのファイルを作成する場合はチェックボックスをオンにします。ファイルの実体がある場合はオフにします。

［入出力するファイル名］の表示規則を次に示します。

- 対象のファイルに EXTERNAL 指定がある場合は，ファイル名の後ろに「<EXTERNAL>」を付けて表示されます。

  （例）

  FILE1△<EXTERNAL>

  △：半角空白を示します。

- 表示するファイル名は，次に示す順番で表示されます。

  優先順位 1：ファイル名

  優先順位 2：定義場所（行番号の若い順）

- ［スタブ設定ページ］の［入出力するファイル名］では，対象のファイルに EXTERNAL 指定がある場合は，どれか 1 つのチェックボックスをオンにすると，ほかの EXTERNAL 付きのファイルは非活性になります。このとき，チェックボックスはオフのまま非活性になります。

  （例）

```
□ FILE1(MAIN/MAIN) <EXTERNAL>
□ FILE1(MAIN/IREKO) <EXTERNAL>
□ FILE1(STACK/STACK) <EXTERNAL>
```

上記の場合，FILE1 のチェックボックスをオンにすると次の状態になります。

```
■ FILE1(MAIN/MAIN) <EXTERNAL>
□ FILE1(MAIN/IREKO) <EXTERNAL>…（チェックがオフのまま非活性）
□ FILE1(STACK/STACK) <EXTERNAL>…（チェックがオフのまま非活性）
```

■：チェックオン，□：チェックオフ

- 03-04 以前のバージョンで作成したテストプロジェクトに EXTERNAL 指定のファイルが含まれ，［スタブ設定ページ］で同名の EXTERNAL 指定のファイルが複数チェックされていた場合，テストプロジェクトを開いたときの動作は次のようになります。

  ・同名の EXTERNAL 指定のファイルが複数チェックされたまま開く※

  ・すべてのファイルについて，同名のファイルでチェックボックスがオンのファイルが 1 つだけの状態になった場合，活性/非活性が制御される

注※

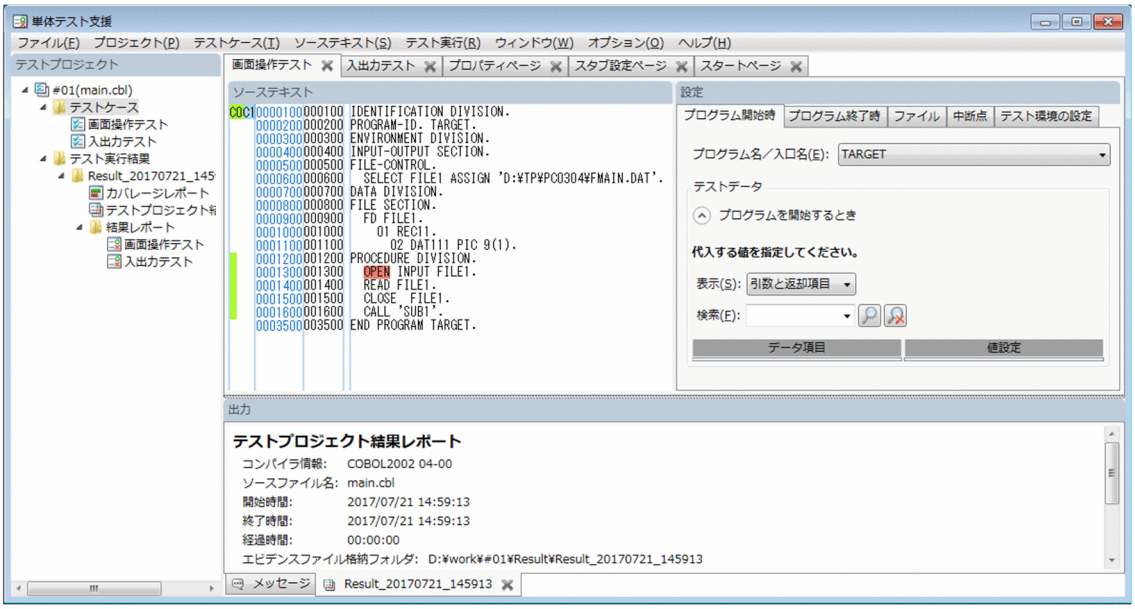
警告メッセージが出力され，同名の EXTERNAL 指定のファイルのうち，どれか 1 つのファイルのチェックボックスだけをオンにする必要があります。どれか 1 つを選択しない（同名のほかのファイルのチェックボックスをオフにしない）ままテスト実行またはテストプロジェクトの保存をしようとした場合，エラーメッセージを出力して処理が中止されます。

また，COBOL ソースファイルの解析時に複数のチェックボックスがオンの状態である場合（複数のチェックボックスがオンの状態を解消しないまま解析をした場合）に警告メッセージが出力されます。

10.8 ［テストケースページ］

テストプロジェクトのテストケースごとのテスト環境を設定する画面です。

単体テストでコンパイルエラーになったときは，単体テストは実行されません。このとき，［テストケースページ］は操作できません。



次の画面から構成されます。

［ソーステキスト］画面

テスト対象の COBOL プログラムのソーステキストを表示する画面です。

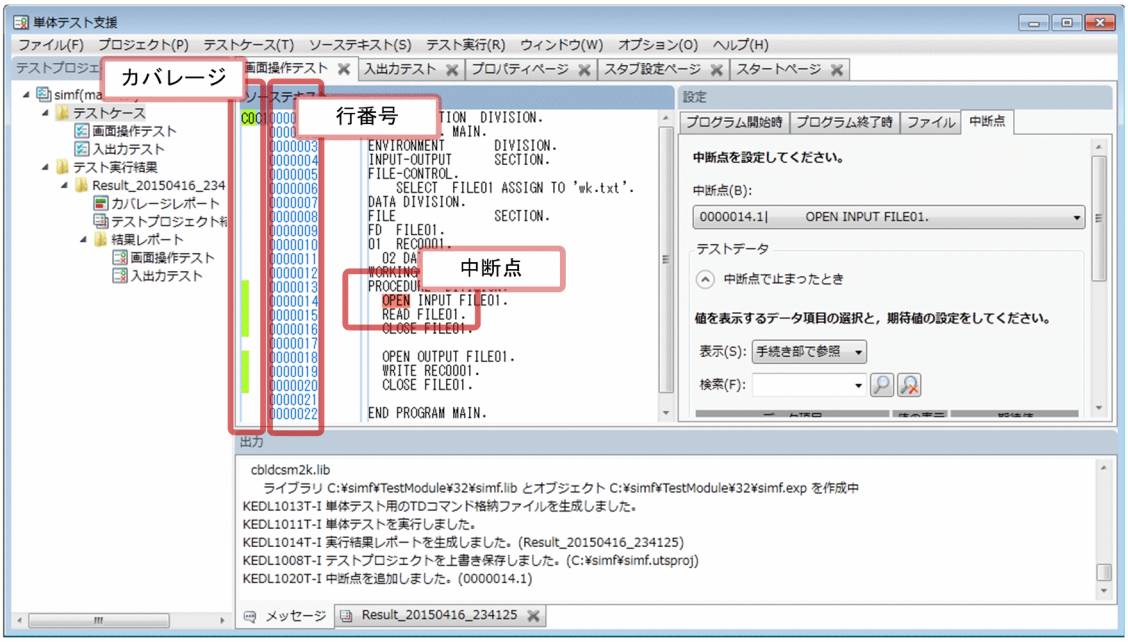
［設定］画面

［プログラム開始時］［プログラム終了時］［ファイル］［中断点］および［テスト環境の設定］の設定をする画面です。

10.8.1 ［ソーステキスト］画面

COBOL プログラムのソーステキストを表示する画面です。

［ソーステキスト］画面に表示するソーステキストは，コンパイラが翻訳したあとのソーステキストです。例えば，COPY 文による登録集原文の展開や，REPLACE 文による原始プログラムの置き換えが実行されたあとの内容になります。LISTING 指令，条件翻訳によってコンパイラが無視する文も表示されます。



[カバレージ]

単体テストの実行で蓄積したカバレージ情報が，ソーステキストの左端に表示されます。C0 メジャーと C1 メジャーを実行する前の行と実行したあとの行を示します。

[行番号]

コンパイラが翻訳時に振り直した行番号です。

[ソーステキスト]

COBOL ソースファイルの内容です。

[中断点]

設定した中断点の文キーワードに色が付きます。

[ソーステキスト] 画面の操作

[ソーステキスト] 画面は，[ソーステキスト] メニューで操作します。[ソーステキスト] メニューを次に示します。次に示すメニューは，ソーステキストを右クリックすると表示されるポップアップメニューからでも選択できます。

項目	機能
検索	ソーステキストの文字列を検索します。
指定行の表示	行番号を指定すると，該当する行が表示されます。
コピー	ソーステキスト内の文字列をコピーします。 ポップアップメニューだけで表示されるメニューです。
中断点の設定	中断点を設定します。 中断点を設定する文（節または段落名を含む）の文キーワードを選択してからメニューを選択してください。メニューを実行すると，選択した文（節または段落名を含む）の文キーワードに背景色が付き，[テスト設定] 画面に [中断点] 画面が表示されます。

項目	機能
中断点の解除	中断点を解除します。 中断点が設定された文（節または段落名を含む）の文キーワードを選択してからメニューを選択してください。
カバレージ情報のクリア	カバレージ情報をクリアします。
COBOL エディタを開く	テスト対象の COBOL ソースファイルを COBOL エディタで開きます。

注意

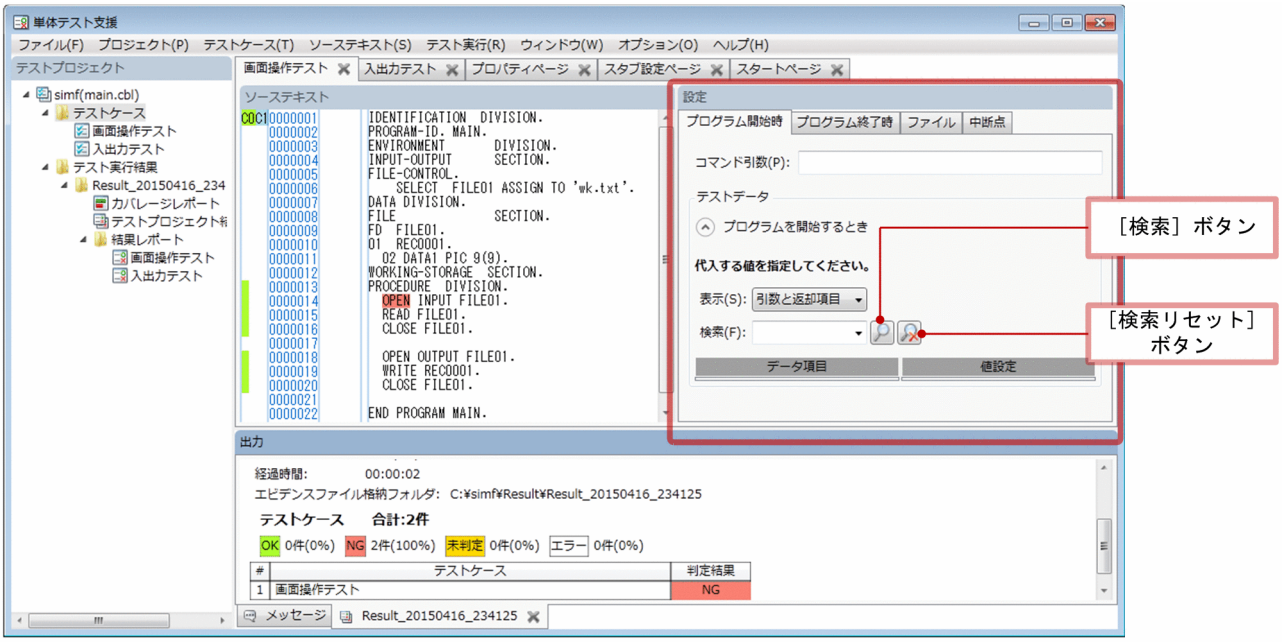
- ［ソーステキスト］画面上に表示されるカバレージ情報は，カバレージ情報リストファイル（coverage.cll）と次に示す違いがあります。
  - PROCEDURE DIVISION，節，段落は，［ソーステキスト］画面上では，カバレージ情報の対象になります。
  - 条件分岐によって省略された部分は，［ソーステキスト］画面上では表示されません。
- 次の個所で設定した環境変数の合計値が 65,535 バイトを超えると，COBOL エディタを開いたときに論理エラーになります。回避するには，不要な環境変数を削除してください。
  - ［新しくテストプロジェクトを作る］ダイアログ，［プロパティページ］または［COBOL ソースファイルの設定］ダイアログで設定したコンパイラ環境変数
  - ［プロパティページ］で設定した実行時環境変数
  - 開発マネージャで設定したコンパイラ環境変数（開発マネージャから起動した場合）
  - システムに設定した環境変数
- COBOL エディタで開く COBOL ソースファイルは，テスト対象としてテストプロジェクトに指定した COBOL ソースファイルです。そのとき，［ソーステキスト］画面で位置づいていた行には移動しません。

  また，COPY 文によって取り込んだ登録集原文を COBOL エディタでは開けません。

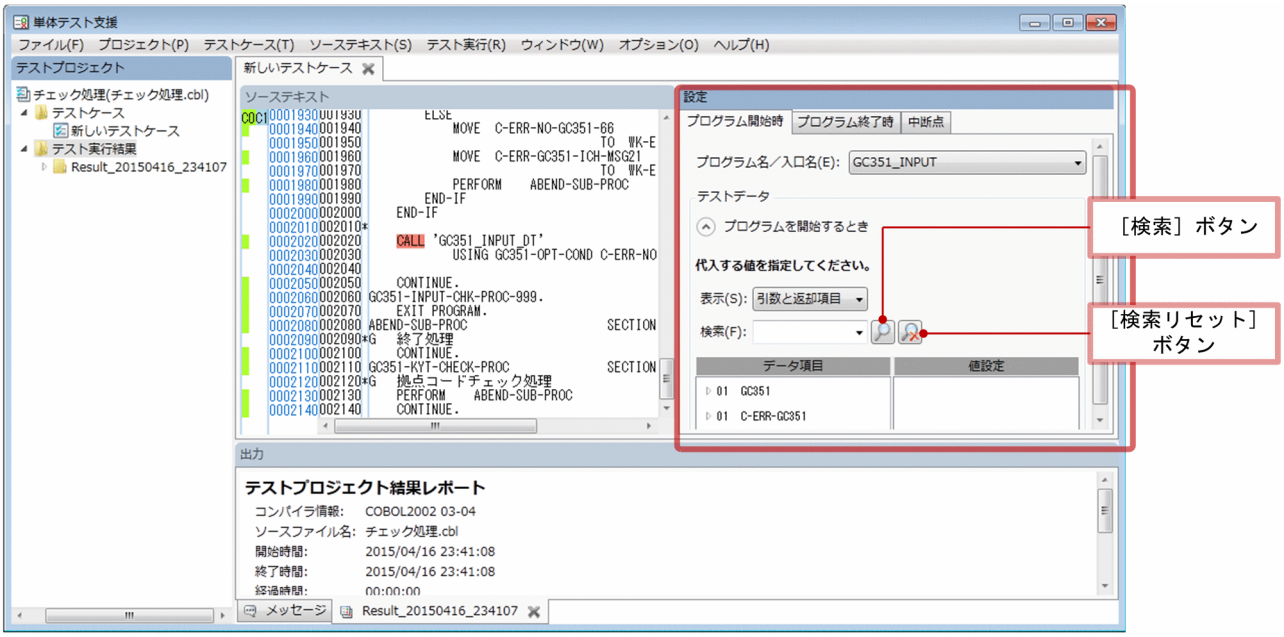
10.8.2 ［プログラム開始時］画面

プログラムの開始時のテストデータを設定する画面です。

テスト対象プログラムが「主プログラム（-Main,System/-Main,V3）」のとき



テスト対象プログラムが「副プログラム」のとき



［コマンド引数］

主プログラムをテスト対象とするときに表示されます。

コマンド引数を指定します。

［プログラム名/入口名］

副プログラムをテスト対象とするときに表示されます。

ENTRY 文がないプログラムでは，［プログラム名/入口名］にプログラム名が表示されます。

ENTRY 文があるプログラムのときは，［プログラム名/入口名］のリストから，プログラムを実行するときの入口点を選択します。

テストデータ

[プログラムを開始するとき]

左側の矢印ボタンで，表示と非表示を切り替えられます。

[表示]

検索対象のデータ項目の範囲を次に示す項目から選択します。

データ項目の範囲	表示するデータ項目
すべて	次に示すデータ項目が表示されます。 • テスト対象のプログラムの，次に示す個所で定義されているすべてのデータ項目 ・ファイル節のレコード記述項で定義されたデータ項目，および指標名 ・作業場所節のデータ記述項で定義されたデータ項目，指標名，およびアドレス名 ・局所場所節のデータ記述項で定義されたデータ項目，指標名，およびアドレス名 ・連絡節のデータ記述項で定義されたデータ項目，および指標名 • テスト対象のプログラムで参照している特殊レジスタ ただし，RETURN-CODE は常に参照しているものとして表示します。
手続き部で参照	次に示すデータ項目が表示されます。 • テスト対象のプログラムの，次に示す個所で定義されているデータ項目のうち，手続き部で参照しているもの ・ファイル節のレコード記述項で定義されたデータ項目，および指標名 ・作業場所節のデータ記述項で定義されたデータ項目，指標名，およびアドレス名 ・局所場所節のデータ記述項で定義されたデータ項目，指標名，およびアドレス名 ・連絡節のデータ記述項で定義されたデータ項目，および指標名 • テスト対象のプログラムで参照している特殊レジスタ
引数と返却項目	次に示すデータ項目が表示されます。 • テスト対象のプログラムの，次に示す個所で定義されているすべてのデータ項目 ・手続き部の見出し領域に指定された引数，および返却項目 ・ENTRY 文の USING 指定に指定されたデータ項目

[検索]

検索するデータ項目を入力します。

[検索］ボタン

クリックすると，［検索］テキストボックスの値に該当するデータ項目の候補だけに絞られます。検索結果の数は，ヒット件数として表示されます。

[検索リセット］ボタン

クリックすると，検索での絞り込みを解除します。

[データ項目]

検索してヒットしたデータ項目の一覧です。

[値設定]

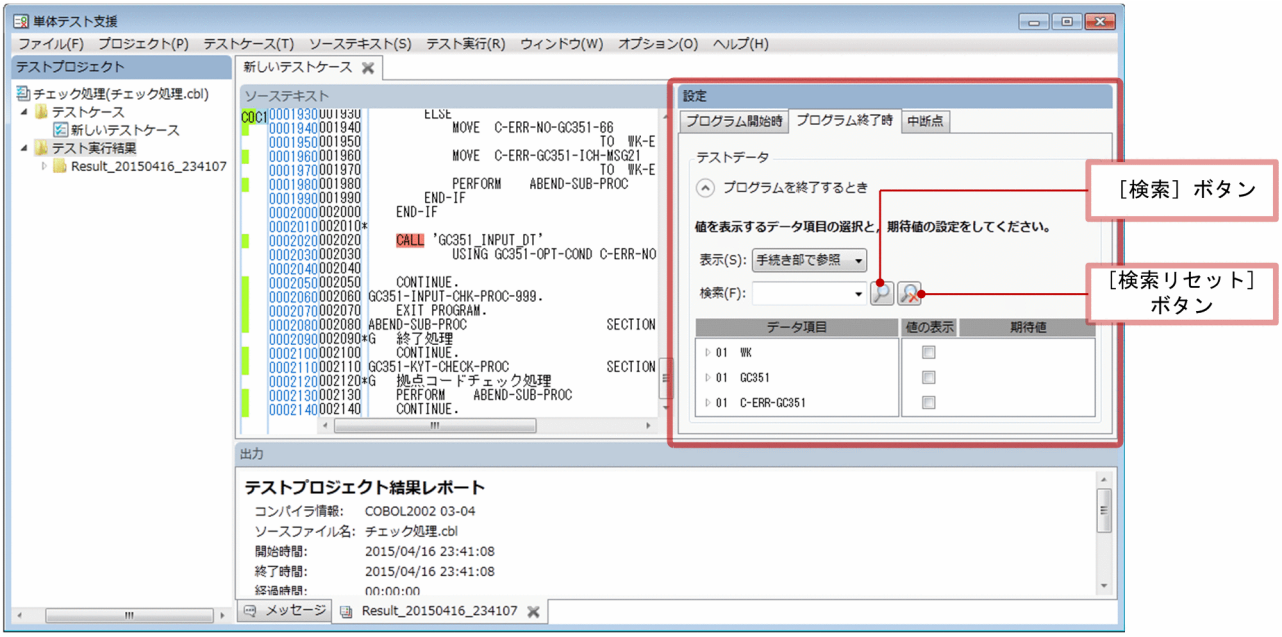
データ項目に設定する値を入力します。

関連項目

- 「9.4　データ項目検索の設定を変更する」

10.8.3　［プログラム終了時］画面

プログラムの終了時のテストデータを設定する画面です。



テストデータ

［プログラムを終了するとき］

　左側の矢印ボタンで，表示と非表示を切り替えられます。

［表示］

　検索対象のデータ項目の範囲を次に示す項目から選択します。

データ項目の範囲	表示するデータ項目
すべて	次に示すデータ項目が表示されます。 • テスト対象のプログラムの，次に示す個所で定義されているすべてのデータ項目 ・ファイル節のレコード記述項で定義されたデータ項目，および指標名 ・作業場所節のデータ記述項で定義されたデータ項目，指標名，およびアドレス名 ・局所場所節のデータ記述項で定義されたデータ項目，指標名，およびアドレス名 ・連絡節のデータ記述項で定義されたデータ項目，および指標名 • テスト対象のプログラムで参照している特殊レジスタ ただし，RETURN-CODE は常に参照しているものとして表示します。
手続き部で参照	次に示すデータ項目が表示されます。 • テスト対象のプログラムの，次に示す個所で定義されているデータ項目のうち，手続き部で参照しているもの

データ項目の範囲	表示するデータ項目
	・ファイル節のレコード記述項で定義されたデータ項目，および指標名 ・作業場所節のデータ記述項で定義されたデータ項目，指標名，およびアドレス名 ・局所場所節のデータ記述項で定義されたデータ項目，指標名，およびアドレス名 ・連絡節のデータ記述項で定義されたデータ項目，および指標名 • テスト対象のプログラムで参照している特殊レジスタ
引数と返却項目	次に示すデータ項目が表示されます。 • テスト対象のプログラムの，次に示す個所で定義されているすべてのデータ項目 ・手続き部の見出し領域に指定された引数，返却項目 ・ENTRY 文の USING 指定に指定されたデータ項目

［検索］

検索するデータ項目を入力します。

［検索］ボタン

クリックすると，［検索］テキストボックスの値に該当するデータ項目の候補だけに絞られます。検索結果の数は，ヒット件数として表示されます。

［検索リセット］ボタン

クリックすると，検索での絞り込みを解除します。

［データ項目］

検索してヒットしたデータ項目の一覧です。

［値の表示］

テストケース結果レポートに表示するデータ項目のチェックボックスをオンにします。

チェックボックスをオンにすると，単体テストを実行したときの実行時の値がテストケース結果レポートに表示されます。

［期待値］

単体テストを実行したときの期待値を設定します。期待値を設定する場合，値の表示のチェックボックスもオンにしてください。設定した期待値で比較処理が実行され，単体テストの実行結果が判定されます。

判定結果については「4.10　単体テストを実行したときの判定結果」を，期待値の設定については「4.4.7　データ項目に値または期待値を設定する」を参照してください。

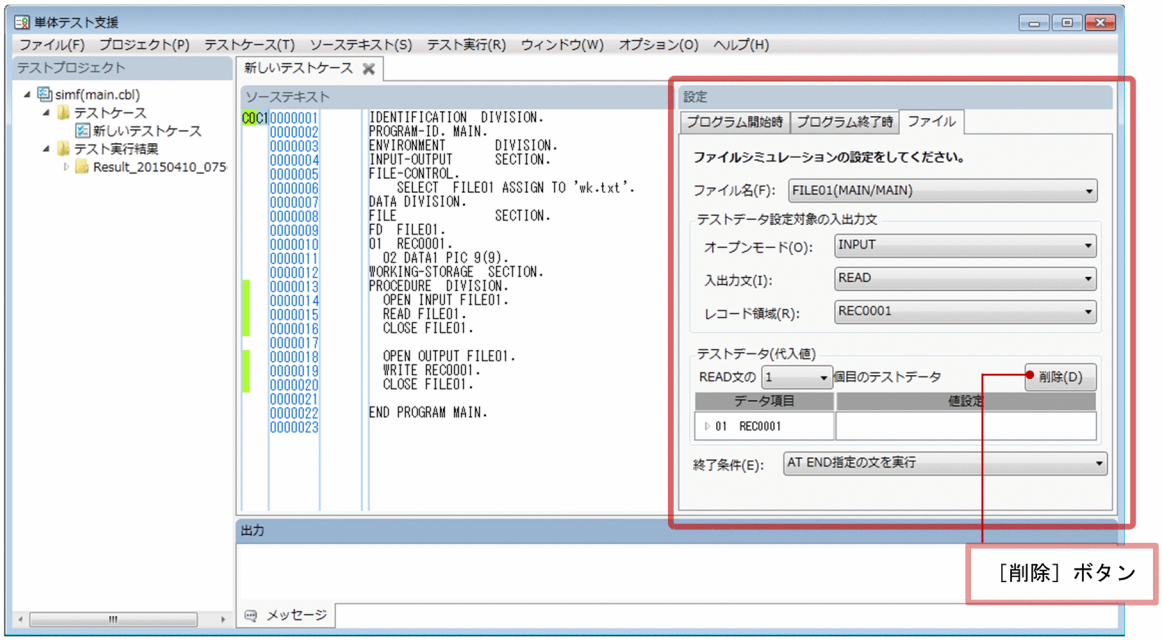
10.8.4　［ファイルシミュレーション］画面

テスト対象のプログラムの中断点で，テストデータを設定する画面です。

次の条件をすべて満たすファイルが 1 つ以上ある場合に，［ファイル］タブが表示されます。

- ［スタブ設定ページ］の「入出力するファイル名」欄のチェックボックスがオンになっている。
- テスト対象プログラム内（入れ子を含む）にファイル記述項がある。

- EXTERNAL 指定がない場合は，テスト対象プログラム内（入れ子を含む）に対象ファイルの OPEN 文がある。



［ファイル名］

ファイルシミュレーションの対象とするファイル名を選択します。

［オープンモード］

ファイルシミュレーションの対象とするオープンモードを選択します。

ファイル名で指定したファイルのオープンモードが表示されます。次に示すオープンモードのうち，実際にテスト対象プログラム内で使用しているモードが候補として表示されます。

- INPUT
- OUTPUT
- I-O
- EXTEND

［入出力文］

ファイルシミュレーションの対象とするオープンモードを選択します。

オープンモードに応じた入出力文が候補として表示されます。オープンモードと表示する入出力文の関係を次に示します。

項番	オープンモード	表示する値
1	INPUT	READ
2	OUTPUT	WRITE
3	I-O	READ
4		WRITE

項番	オープンモード	表示する値
5		REWRITE
6	EXTEND	WRITE

［レコード領域］

ファイルシミュレーションの対象とするレコード領域を選択します。

ファイルに対して，ファイル記述項で定義されているレコード領域名が表示されます。

［*入出力文の n* 個目のテストデータ］

設定しているテストデータが，該当する入出力文が何回目に実行されたときに使用するかを選択します。コンボボックスに登録される番号は 1 から最大 100 です。

データの個数が 100 個未満の場合は，コンボボックスの候補の最後に［追加］が表示されます。［追加］を選択すると，最後に追加されます。

［削除］ボタン

表示中のテストデータセットを削除します。

［終了条件］

ファイルシミュレーションの終了条件を選択します。

ここで設定した終了条件は，テストデータをすべて使用したあとで判定されます。例えば，READ 文用のテストデータが 3 つあった場合，4 回目の READ 文の実行で判定されます。

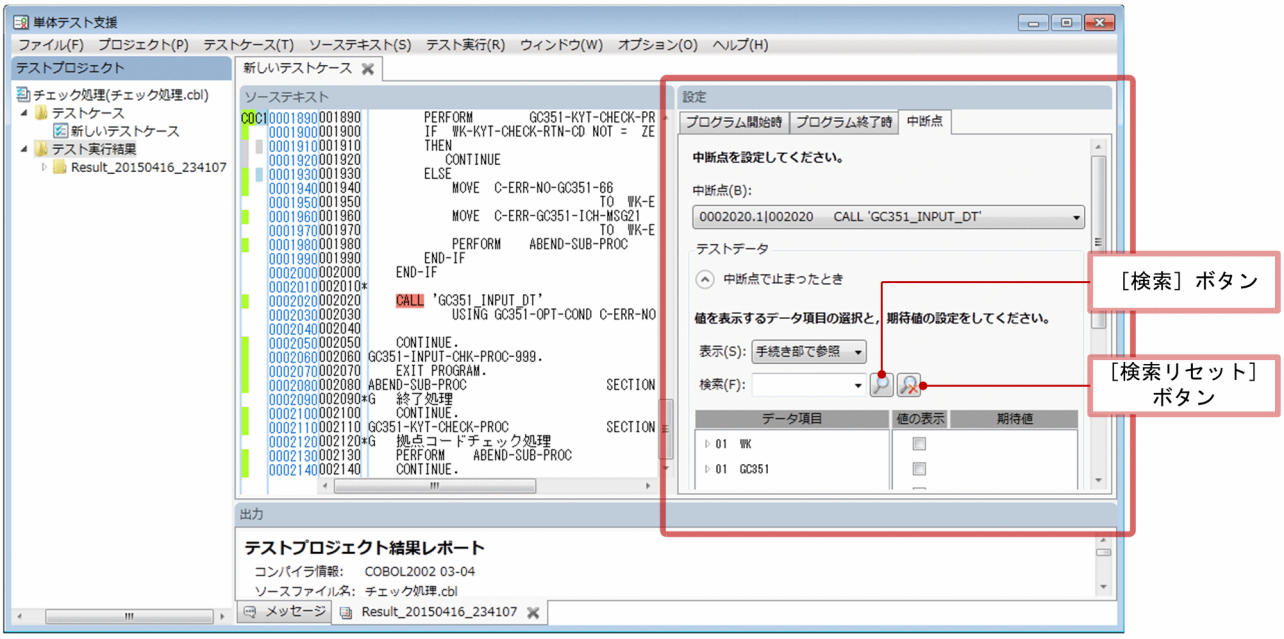
入出力文と終了条件の関係を次に示します。

入出力文	終了条件
READ	AT END 指定の文を実行
	INVALID KEY 指定の文を実行
	入出力エラーを発生させる
	なし
WRITE	AT END-OF-PAGE 指定の文を実行
	INVALID KEY 指定の文を実行
	入出力エラーを発生させる
	なし
REWRITE	INVALID KEY 指定の文を実行
	入出力エラーを発生させる
	なし

10.8.5 ［中断点］画面

テスト対象のプログラムの中断点で，テストデータを設定する画面です。

［ソーステキスト］画面に中断点を設定したときに表示されます。



［中断点］

ソーステキストに設定されている中断点の一覧が表示されます。

先頭に行番号と位置番号をピリオドでつないだ文番号が表示されます。

位置番号とは，行中の文に対する通し番号で，先頭の文を 1 とします。

テストデータ

中断点で止まったとき

以降で，中断点が止まったときの設定を入力します。

左側の矢印ボタンで，表示と非表示を切り替えられます。

［表示］

検索対象のデータ項目の範囲を次に示す項目から選択します。

データ項目の範囲	表示するデータ項目
すべて	次に示すデータ項目が表示されます。 • テスト対象のプログラムの，次に示す個所で定義されているすべてのデータ項目 ・ファイル節のレコード記述項で定義されたデータ項目，および指標名 ・作業場所節のデータ記述項で定義されたデータ項目，指標名，およびアドレス名 ・局所場所節のデータ記述項で定義されたデータ項目，指標名，およびアドレス名 ・連絡節のデータ記述項で定義されたデータ項目，および指標名 • テスト対象のプログラムで参照している特殊レジスタ

データ項目の範囲	表示するデータ項目
	ただし，RETURN-CODE は常に参照しているものとして表示します。
手続き部で参照	次に示すデータ項目が表示されます。 • テスト対象のプログラムの，次に示す個所で定義されているデータ項目のうち，手続き部で参照しているもの ・ファイル節のレコード記述項で定義されたデータ項目，および指標名 ・作業場所節のデータ記述項で定義されたデータ項目，および指標名，アドレス名 ・局所場所節のデータ記述項で定義されたデータ項目，および指標名，アドレス名 ・連絡節のデータ記述項で定義されたデータ項目，および指標名 • テスト対象のプログラムで参照している特殊レジスタ

内側のプログラムの手続き部に設定した中断点の場合は，次に示す項目から選択します。

データ項目の範囲	表示するデータ項目
すべて	次に示すデータ項目が表示されます。 • 内側のプログラムの，次に示す個所で定義されているすべてのデータ項目 ・ファイル節のレコード記述項で定義されたデータ項目，および指標名 ・作業場所節のデータ記述項で定義されたデータ項目，指標名，およびアドレス名 ・局所場所節のデータ記述項で定義されたデータ項目，指標名，およびアドレス名 ・連絡節のデータ記述項で定義されたデータ項目，および指標名 • テスト対象のプログラムの，次に示す個所で定義されているデータ項目のうち，GLOBAL 句のあるデータ項目 ・ファイル節のレコード記述項で定義されたデータ項目，および指標名 ・作業場所節のデータ記述項で定義されたデータ項目，指標名，およびアドレス名 ・局所場所節のデータ記述項で定義されたデータ項目，指標名，およびアドレス名 ・連絡節のデータ記述項で定義されたデータ項目，および指標名 • 内側のプログラムで参照している特殊レジスタ ただし，RETURN-CODE は常に参照しているものとして表示します。
手続き部で参照	次に示すデータ項目が表示されます。 • 内側のプログラムの，次に示す個所で定義されているデータ項目のうち，手続き部で参照しているもの ・ファイル節のレコード記述項で定義されたデータ項目，および指標名 ・作業場所節のデータ記述項で定義されたデータ項目，および指標名，アドレス名 ・局所場所節のデータ記述項で定義されたデータ項目，および指標名，アドレス名 ・連絡節のデータ記述項で定義されたデータ項目，および指標名 • テスト対象のプログラムの，次に示す個所で定義されているデータ項目データ項目のうち，GLOBAL 句のあり，かつ手続き部で参照しているもの ・ファイル節のレコード記述項で定義されたデータ項目，および指標名 ・作業場所節のデータ記述項で定義されたデータ項目，および指標名，アドレス名 ・局所場所節のデータ記述項で定義されたデータ項目，および指標名，アドレス名 ・連絡節のデータ記述項で定義されたデータ項目，および指標名 • 内側のプログラムで参照している特殊レジスタ

［検索］

検索するデータ項目を入力します。

［検索］ボタン

クリックすると，［検索］テキストボックスの値に該当するデータ項目の候補だけに絞られます。検索結果の数は，ヒット件数として表示されます。

［検索リセット］ボタン

クリックすると，検索での絞り込みを解除します。

［データ項目］

検索してヒットしたデータ項目の一覧です。

［値の表示］

テストケース結果レポートに表示するデータ項目のチェックボックスをオンにします。

チェックボックスをオンにすると，単体テストを実行したときの実行時の値がテストケース結果レポートに表示されます。

［期待値］

単体テストを実行したときの期待値を設定します。期待値を設定する場合，値の表示のチェックボックスもオンにしてください。設定した期待値で比較処理が実行され，単体テストの実行結果が判定されます。

中断点から実行するとき

以降で，中断点から実行するときに代入する値を入力します。

左側の矢印ボタンで，表示と非表示を切り替えられます。

［表示］

検索対象のデータ項目の範囲を次に示す項目から選択します。

データ項目の範囲	表示するデータ項目
すべて	次に示すデータ項目が表示されます。 • テスト対象のプログラムの，次に示す個所で定義されているすべてのデータ項目 ・ファイル節のレコード記述項で定義されたデータ項目，指標名 ・作業場所節のデータ記述項で定義されたデータ項目，指標名，アドレス名 ・局所場所節のデータ記述項で定義されたデータ項目，指標名，アドレス名 ・連絡節のデータ記述項で定義されたデータ項目，指標名 • テスト対象のプログラムで参照している特殊レジスタ ただし，RETURN-CODE は常に参照しているものとして表示します。
手続き部で参照	次に示すデータ項目が表示されます。 • テスト対象のプログラムの，次に示す個所で定義されているデータ項目のうち，手続き部で参照しているもの ・ファイル節のレコード記述項で定義されたデータ項目，指標名 ・作業場所節のデータ記述項で定義されたデータ項目，指標名，アドレス名 ・局所場所節のデータ記述項で定義されたデータ項目，指標名，アドレス名 ・連絡節のデータ記述項で定義されたデータ項目，指標名

データ項目の範囲	表示するデータ項目
	• テスト対象のプログラムで参照している特殊レジスタ

［検索］

検索するデータ項目を入力します。

［検索］ボタン

クリックすると，［検索］テキストボックスの値に該当するデータ項目の候補だけに絞られます。検索結果の数は，ヒット件数として表示されます。

［検索リセット］ボタン

クリックすると，検索での絞り込みを解除します。

［データ項目］

検索してヒットしたデータ項目の一覧です。

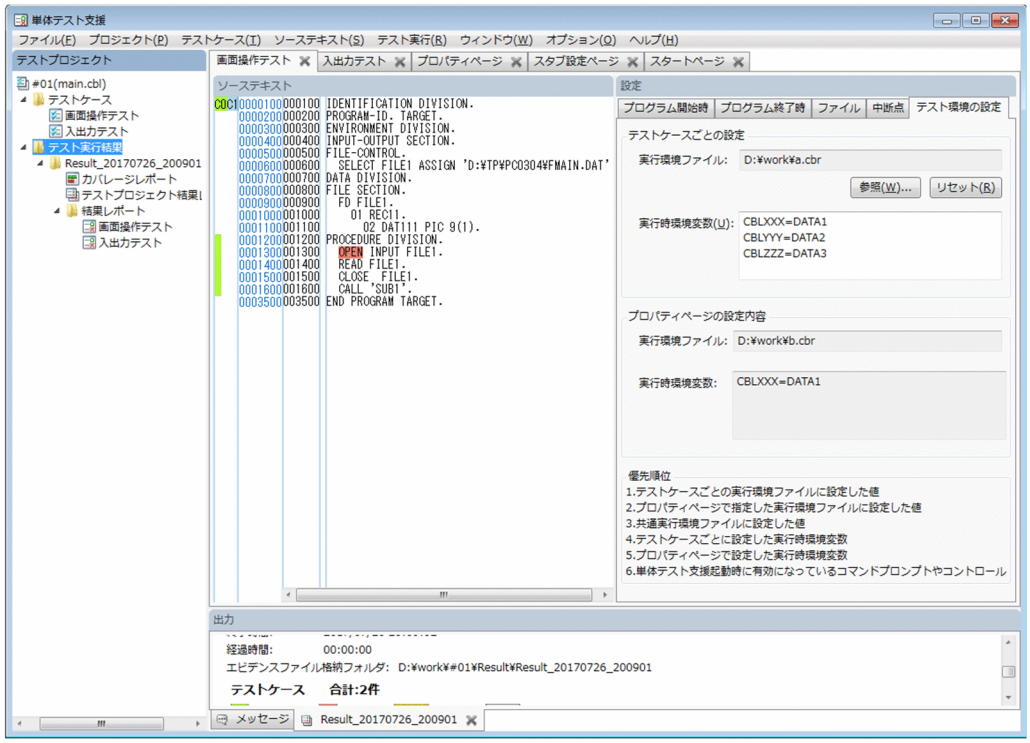
［値設定］

データ項目に設定する値を入力します。

10.8.6 ［テスト環境の設定］画面

テストケース固有の環境として，テスト実行時の実行環境ファイルまたは実行時環境変数の値を設定する画面です。

［プロパティページ］で［テストケースごとに実行環境ファイルまたは実行時環境変数を設定する］チェックボックスをオンにしたときに表示されます。



[実行環境ファイル]

テスト実行で利用する実行環境ファイルの絶対パスです。単体テスト支援が単体テストを実行する前に，毎回テストモジュールフォルダに実行環境ファイルをコピーします。

[実行時環境変数]

テスト実行するときの実行時環境変数を指定します。

関連項目

- 「4.5.3　テストケースごとに実行環境ファイルと実行時環境変数を設定する」

10.9 ［出力］画面

単体テスト支援が出力する項目を表示する画面です。

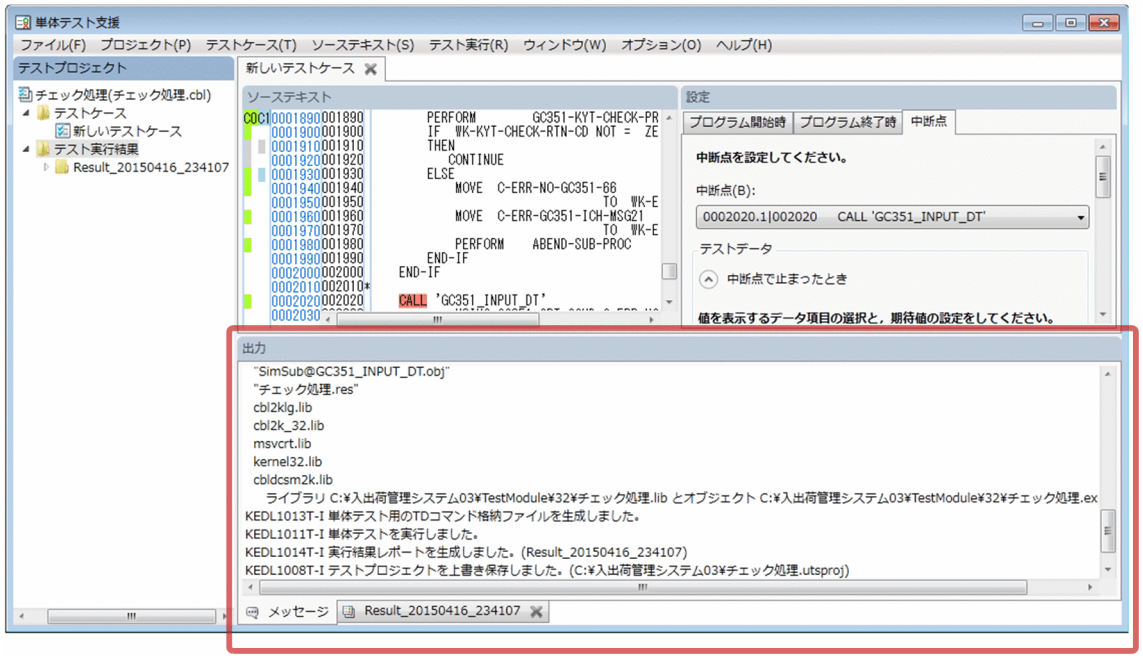
［出力］画面に表示される画面を次に示します。

画面	内容	画面を表示する条件
メッセージ	単体テスト支援のメッセージです。	常時
カバレージレポート	単体テストの実行によるカバレージの結果です。	単体テストの実行後
テストプロジェクト結果レポート	テストプロジェクトに属する，すべてのテストケースの実行結果です。	単体テストの実行後
テストケース結果レポート	テストケースごとの実行結果です。	単体テストの実行後

10.9.1 ［メッセージ］画面

単体テスト支援のメッセージを表示する画面です。

最新のメッセージが最下部に表示されます。表示されているメッセージを消すときは，画面を右クリックしてポップアップメニューから［クリア］メニューを選択します。

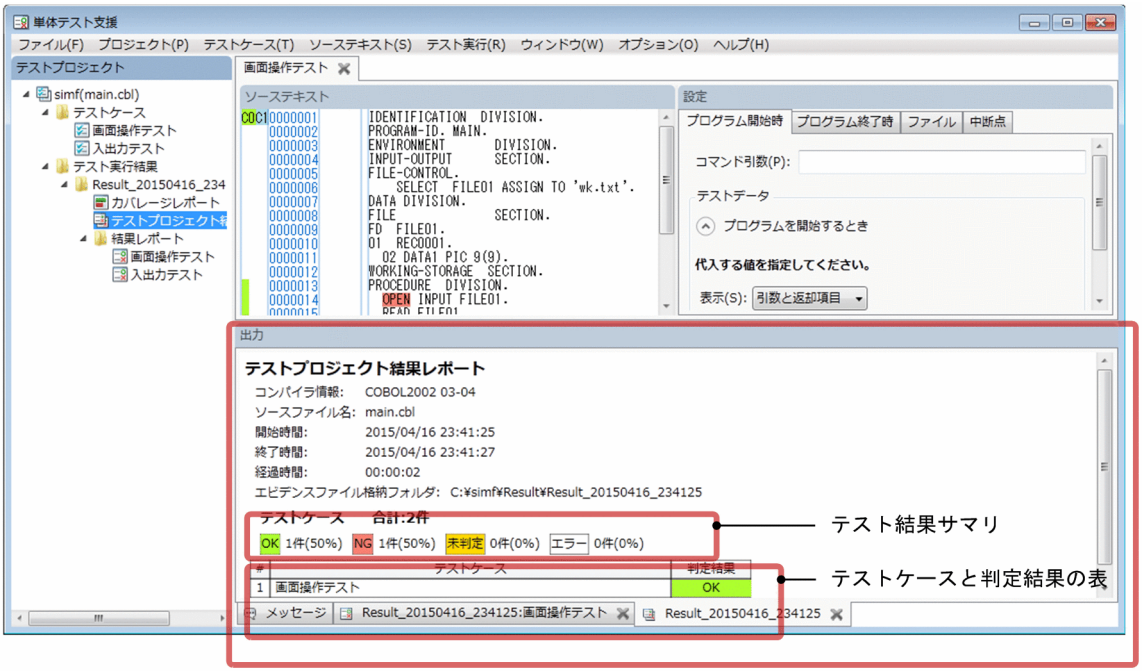


注意

- ［メッセージ］画面に表示するメッセージの行数は 1,000 行までです。1,000 行を超えたときは，古い行から順に消去されます。

10.9.2 ［テストプロジェクト結果レポート］画面

テストプロジェクト内の複数のテストケースに対して，一括で単体テストを実行した結果を表示する画面です。



［テストプロジェクト結果レポート］画面に表示される情報

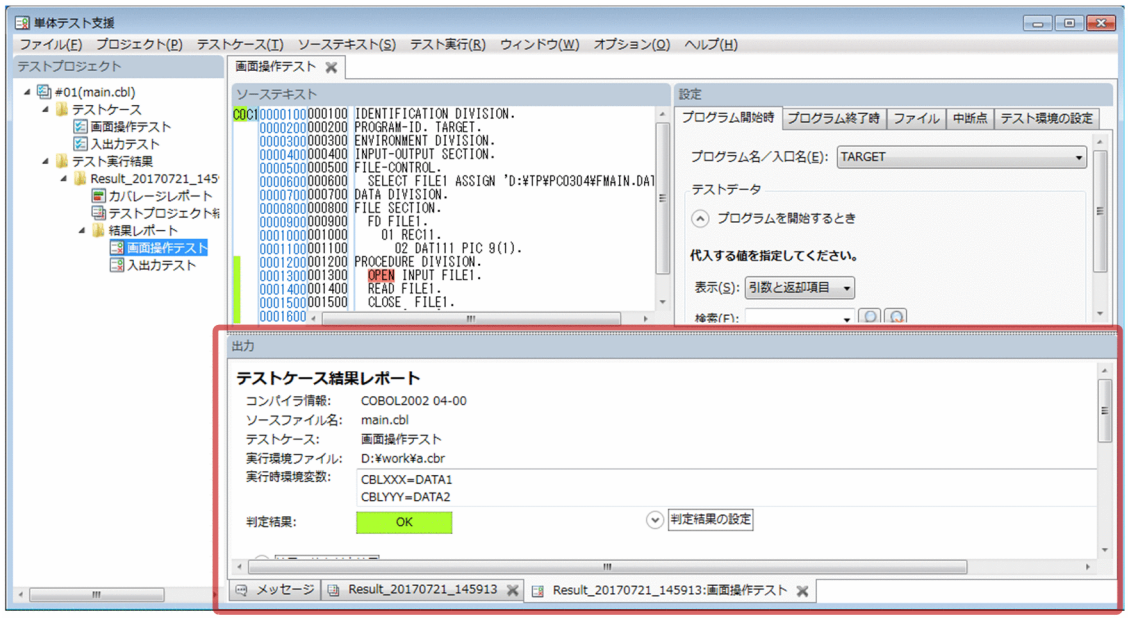
項目		内容
コンパイラ情報		COBOL2002 の製品情報です。
ソースファイル名		テスト対象の COBOL ソースファイルの名前です。
開始時間		テストを開始した時間です。
終了時間		テストを終了した時間です。
経過時間		テストの実行に掛かった時間です。
エビデンスファイル格納フォルダ		テストの実行結果を出力したエビデンスファイルの格納先フォルダのパスです。
テスト結果サマリ		実行したテストケースの総数です。 OK，NG，未判定，およびエラーの数と割合が表示されます。
テストケースと判定結果の表	テストケース	テストケースの名前です。
	判定結果	テストの判定結果です。

関連項目

- 「6. テスト結果を参照する」

10.9.3 ［テストケース結果レポート］画面

1 つのテストケースをテスト実行した結果を表示する画面です。



［テストケース結果レポート］画面に表示される情報

項目	内容
コンパイラ情報	COBOL2002 の製品情報です。
ソースファイル名	テスト対象の COBOL ソースファイルの名前です。
テストケース	テストを実行したテストケースの名前です。
実行環境ファイル	テスト実行時に使用した実行環境ファイルの絶対パスが表示されます。※
実行時環境変数	テスト実行時に使用した実行時環境変数が表示されます。※
判定結果	テストケースの実行結果が OK，NG，未判定またはエラーで表示されます。
判定結果の設定	判定結果を手動で設定するときに使用します。 自動判定，OK，NG，未判定のどれかを判定結果に設定します。
結果の値と判定結果	［プログラム終了時］画面と［中断点］画面で「値の表示」に指定したデータ項目の値と，「期待値」に指定したデータ項目の判定結果を表示するときに確認します。
代入した値	［プログラム開始時］画面と［中断点］画面の「値設定」で代入したテストデータの値を表示するときに指定します。
出力文の実行結果	［ファイルシミュレーション］画面で，出力文に対して「値の表示」を指定したデータ項目の値と，「期待値の設定」を指定したデータ項目の判定結果です。
入力文の実行結果	［ファイルシミュレーション］画面で，入力文に対して「値の設定」によって代入されたテストデータの値です。

注※

［プロパティページ］の［テストケースごとに実行環境ファイルまたは実行時環境変数を設定する］チェックボックスがオンになっている状態でテストを実行した場合，結果に項目と内容が表示されます。チェックボックスがオフになっている状態でテストを実行した場合は，表示されません。

チェックボックスがオンになっている状態で，実行環境ファイルや実行時環境変数の指定がない場合は，項目だけが表示されます。

［実行時環境変数］に表示される環境変数は，［テスト環境の設定］画面と［プロパティページ］で指定した実行時環境変数をマージした値が，環境変数名で昇順にソートされて表示されます。

［結果の値と判定結果］に表示される内容

項目	内容
テストデータ設定個所	データ項目がテストデータとして設定された位置です。
レベル	データ項目のレベル番号です。
データ項目	データ項目の名前です。集団項目に属するデータ項目のときは，修飾付きのデータ名で表示されます。
値	実行結果のデータ項目の値です。16 進数で表示するときは，［16 進表示］のチェックボックスをオンにします。
期待値	データ項目の期待値です。16 進数で表示するときは，［16 進表示］のチェックボックスをオンにします。
判定結果	実行結果の値と期待値を比較した判定結果です。

［代入した値］に表示される内容

項目	内容
テストデータ設定個所	データ項目がテストデータとして設定された位置です。
レベル	データ項目のレベル番号です。
データ項目	データ項目の名前です。集団項目に属するデータ項目のときは，修飾付きのデータ名で表示されます。
値	テストの実行で代入したデータ項目の値です。16 進数で表示するときは，［16 進表示］のチェックボックスをオンにします。

［出力文の実行結果］に表示される内容

項目	内容
ファイル名	シミュレーションした出力文の対象ファイルの名前です。
オープンモード	シミュレーションした出力文の対象ファイルをオープンしたときのオープンモードです。
出力文	シミュレーションした出力文です。
出力文の場所	シミュレーションした出力文の場所の文番号です。
判定結果	実行結果の値と期待値を比較した判定結果です。

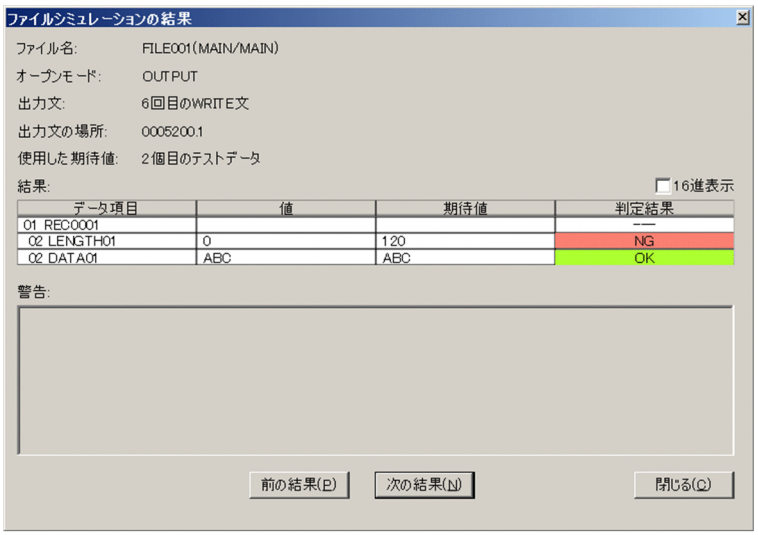
項目	内容
	終了条件の実行の場合は，「終了条件」と表示されます。 データの組み合わせなどのエラーのときは「エラー」と表示されます。

［入力文の実行結果］に表示される内容

項目	内容
ファイル名	シミュレーションした入力文の対象ファイルの名前です。
オープンモード	シミュレーションした入力文の対象ファイルをオープンしたときのオープンモードです。
入力文	シミュレーションした入力文です。
入力文の場所	シミュレーションした入力文の場所の文番号です。
判定結果	値の代入が実行されたときは，「---」と表示されます。 終了条件が実行されたときは，「終了条件」と表示されます。 データの組み合わせなどのエラーになったときは「エラー」と表示されます。

［ファイルシミュレーションの結果］に表示される内容（［出力文の実行結果］から表示した場合）

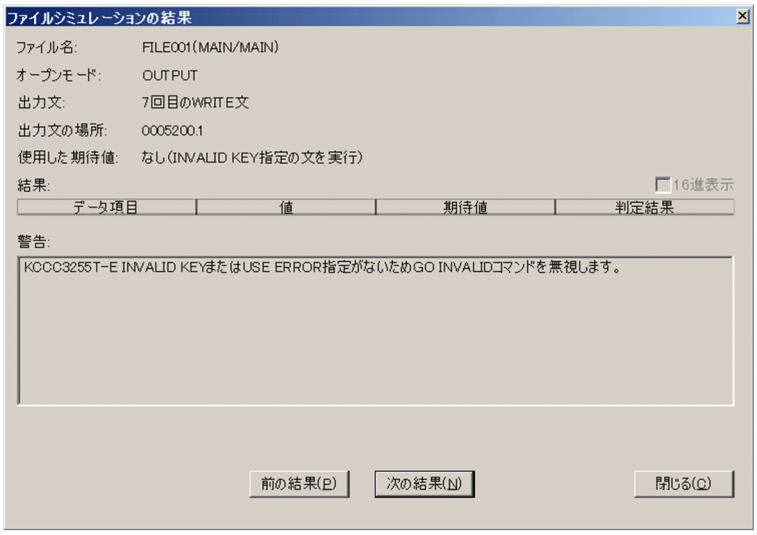
［出力文の実行結果］の任意の行を右クリックし，［結果を表示］メニューを選択すると，［ファイルシミュレーションの結果］ダイアログが表示されます。



項目	内容
ファイル名	シミュレーションした出力文の対象ファイルの名前です。
オープンモード	シミュレーションした出力文の対象ファイルをオープンしたときのオープンモードです。
出力文	シミュレーションした出力文です。
出力文の場所	シミュレーションした出力文の場所の文番号です。
使用した期待値	シミュレーションしたときに，何番目のテストデータと比較したか，です。

項目	内容
結果	データ項目単位のシミュレーションの結果です。 データ項目 データ項目の名前です。 値 実行結果のデータ項目の値です。 期待値 データ項目の期待値です。データの組み合わせなどのエラーになったときは空欄です。 判定結果 実行結果の値と期待値を比較した判定結果です。比較対象ではない場合は「---」と表示されます。データの組み合わせなどのエラーになったときは「エラー」と表示されます。
16 進表示	チェックボックスをオンにすると，［値］が 16 進表示になります。
警告	シミュレーションで問題があった場合，その内容が COBOL2002 のテストデバッガのメッセージで表示されます。 組み合わせ不正など，データ項目ごとでエラーになっている場合は，［結果］一覧で，エラーになっている行を選択すると，エラーメッセージが表示されます。メッセージについては，マニュアル「COBOL2002 メッセージ」を参照してください。
前の結果	前の結果です。表示される内容は，［出力文の実行結果］の一覧での絞り込みに合わせた結果です。例えば，OK で絞り込みをしていた場合は，判定結果が OK の結果が表示されます。 ［テストケース結果レポート］画面の「入力文の実行結果」で選択中の行が，このダイアログで表示した内容に変わります。
次の結果	次の結果です。表示される内容は，［入力文の実行結果］の一覧での絞り込みに合わせた結果です。例えば，OK で絞り込みをしていた場合は，判定結果が OK の結果が表示されます。 ［テストケース結果レポート］画面の「出力文の実行結果」で選択中の行が，このダイアログで表示した内容に変わります。
閉じる	ダイアログを閉じます。

終了条件を実行した場合は，次のように表示されます。



[結果:] の一覧の任意の行を選択しているときには，任意の行を右クリックして［値と期待値を表示］メニューを選択すると，［結果の値と期待値］ダイアログで一覧では収まりきらない大きなデータ値を確認できます。

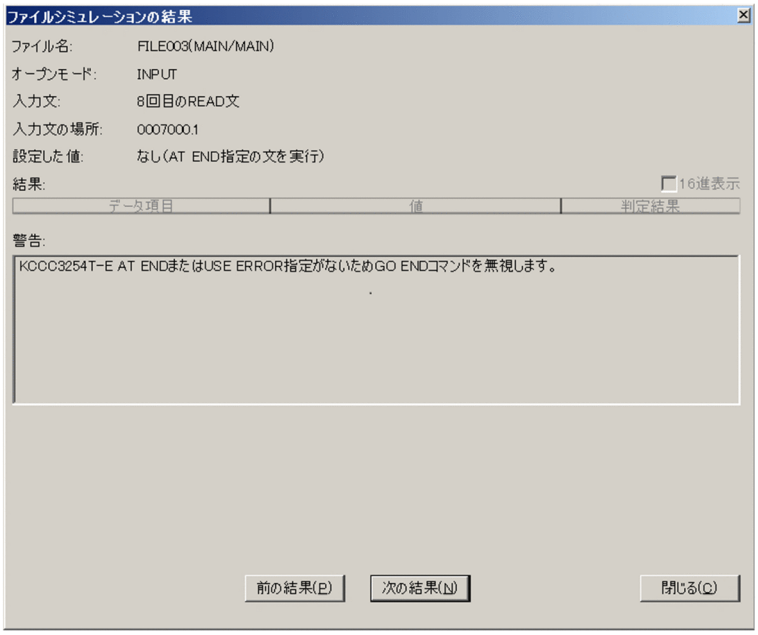
［ファイルシミュレーションの結果］に表示される内容（［入力文の実行結果］から表示した場合）

［入力文の実行結果］の任意の行を右クリックし，［結果を表示］メニューを選択すると，［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。

項目	内容
ファイル名	シミュレーションした入力文の対象ファイルの名前です。
オープンモード	シミュレーションした入力文の対象ファイルをオープンしたときのオープンモードです。
入力文	シミュレーションした入力文です。

項目	内容
入力文の場所	シミュレーションした入力文の場所の文番号です。
設定した値	シミュレーションしたときに，何番目のテストデータと比較したか，です。
結果	データ項目単位のシミュレーションの結果です。 データ項目 データ項目の名前です。 値 実行結果のデータ項目の値です。データの組み合わせなどのエラーになったときは空欄です。 判定結果 データの組み合わせなどのエラーになったときは「エラー」と表示されます。これ以外の場合は「---」と表示されます。
16 進表示	チェックボックスをオンにすると，［値］が 16 進表示になります。
警告	シミュレーションで問題があった場合，その内容が COBOL2002 のテストデバッガのメッセージで表示されます。 組み合わせ不正など，データ項目ごとでエラーになっている場合は，［結果］一覧で，エラーになっている行を選択すると，エラーメッセージが表示されます。メッセージについては，マニュアル「COBOL2002 メッセージ」を参照してください。
前の結果	前の結果です。 ［テストケース結果レポート］画面の「入力文の実行結果」で選択中の行が，このダイアログで表示した内容に変わります。
次の結果	次の結果です。 ［テストケース結果レポート］画面の「入力文の実行結果」で選択中の行が，このダイアログで表示した内容に変わります。
閉じる	ダイアログを閉じます。

終了条件を実行した場合は，次のように表示されます。

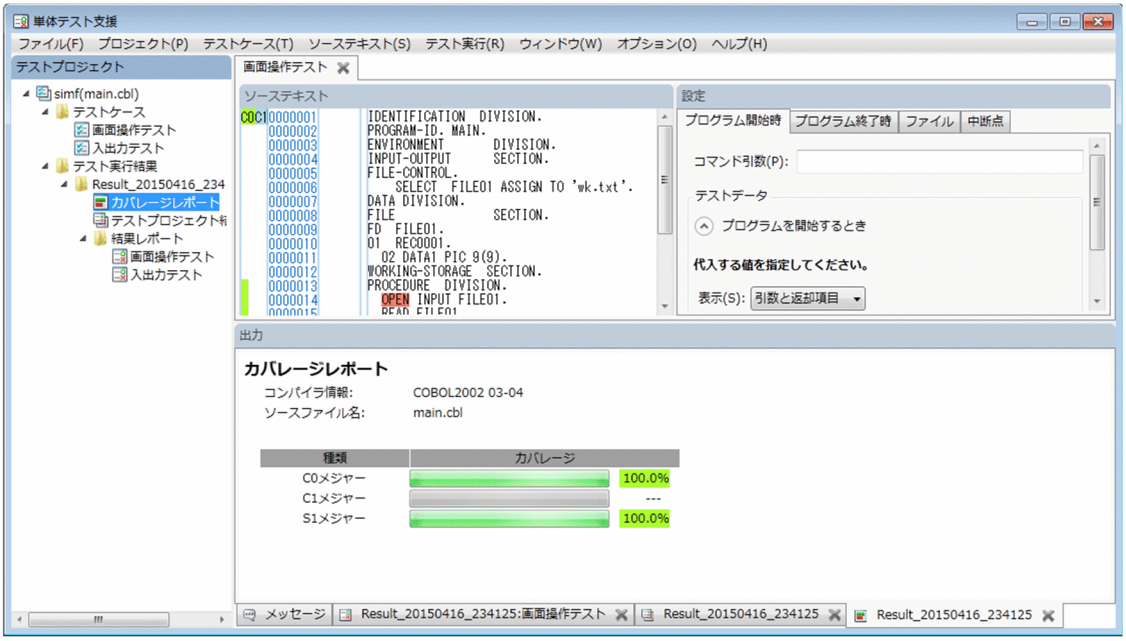


[結果:] の一覧の任意の行を選択しているときには，任意の行を右クリックして［値と期待値を表示］メニューを選択すると，［結果の値と期待値］ダイアログで一覧では収まりきらない大きなデータ値を確認できます。

10.9.4 ［カバレージレポート］画面

単体テストを実行したときのカバレージ情報を表示する画面です。

カバレージ情報には，テストプロジェクトで繰り返し実行した単体テストの累積値が表示されます。カバレージレポートには，それまでに実行した単体テストによるカバレージの値が累積して表示されます。



［カバレージレポート］画面に表示される情報

項目		内容
コンパイラ情報		COBOL2002 の製品情報です。
ソースファイル名		テスト対象の COBOL ソースファイルの名前です。
カバレージ情報		テストで実行された手続き部の割合です。
カバレージ情報の種類	C0 メジャー	実行した文の割合です。
	C1 メジャー	処理が分岐する個所で，実行した分岐先の数の割合です。
	S1 メジャー	実行した呼び出し文の数の割合です。

11 ファイルのリファレンス

単体テスト支援で使用するファイルについて説明します。

11.1　テストプロジェクトのファイル構成

テストプロジェクトのファイル構成の説明です。

単体テスト支援では，1 つの COBOL ソースファイルに対応して 1 つのテストプロジェクトを作成します。テストプロジェクトは，テストプロジェクトファイルと，単体テストの実行に必要なファイル群およびテスト実行結果のファイル群から構成されます。

テストプロジェクト作業フォルダは，単体テストの実行時のカレントフォルダになります。

テストプロジェクトのファイル	ファイルまたはフォルダ名	説明
テストプロジェクトファイル	*テストプロジェクト名*.utsproj	単体テストのためのテストデータおよびテスト結果の情報が格納されているファイルです。
テストモジュールフォルダ	TestModule	単体テスト用の実行可能ファイルなどのファイルを配置します。 テストモジュールフォルダは，32bit 版の COBOL2002 と 64bit 版の COBOL2002 とで下位フォルダが分かれます。
32bit 用テストモジュールフォルダ	TestModule¥32	32bit 版の COBOL2002 で単体テストを実行する場合に，COBOL ソースファイルなどが格納されます。
64bit 用テストモジュールフォルダ	TestModule¥64	64bit 版の COBOL2002 で単体テストを実行する場合に，COBOL ソースファイルなどが格納されます。
TD コマンド格納ファイルフォルダ	Tdi	テストデバッガで単体テストを実行するための TD コマンドを記述したファイルを格納するフォルダです。
実行結果フォルダ	Result	テストプロジェクトで実行した単体テストの結果のエビデンスファイルが格納されます。下位フォルダには，単体テストの実行ごとの結果が格納されます。
1 回分の実行結果フォルダ	Result ¥Result_*yyyymmdd_hhmmss*	1 回分の単体テストの実行結果のエビデンスファイルが格納されます。

注

テストプロジェクトファイルの*テストプロジェクト名*は，テストプロジェクトのプロジェクト名を示します。

1 回分の実行結果フォルダの *yyyymmdd_hhmmss* は，単体テストを実行した日時（*yyyy*：西暦年，*mm*：月，*dd*：日，*hh*：時，*mm*：分，*ss*：秒）を示します。

注意

- 1 つの COBOL ソースファイルに複数の最外側のプログラムがあるときは，プログラム名を一覧から選択して，テストプロジェクトを作成する必要があります。
- 1 つのテストプロジェクト作業フォルダに 2 つ以上のテストプロジェクトファイルは格納できません。

- テストプロジェクトを構成しているファイルは，テストプロジェクト作業フォルダからの相対パスで管理されます。
- テストプロジェクト作業フォルダ下のファイルの直接編集や移動はしないでください。

11.2　プロジェクト情報ファイルの構成要素

プロジェクト情報ファイルは XML 形式に従います。エンコードはシフト JIS（Shift_JIS）です。

プロジェクト情報ファイルの構成要素を次に示します。インデントは階層を表します。

要素			説明
<CblUnitTestSupport>			テストプロジェクトの情報を定義する。
	<Property>		［プロパティページ］の内容を定義する。
		<ProjectName>	プロジェクト名。
		<ProgramName>	プログラム名。
		<WorkingFolder>	テストプロジェクト作業フォルダ。
		<COBOLSourceFile>	COBOL ソースファイルのパス。
		<TargetProgramType>	テスト対象プログラムの種類。 次のどれかの値。 Main,System：主プログラム（-Main,System）を表す Main,V3：主プログラム（-Main,V3）を表す Sub：副プログラムを表す
		<CompilerOptions>	コンパイラオプションを定義する。
		<CompilerOption>	コンパイラオプション。
		<CompilerEnvironmentVariables>	コンパイラ環境変数を定義する。
		<CompilerEnvironmentVariable>	コンパイラ環境変数。 常に空要素。 次の属性に値を設定する。 name 属性：環境変数名 value 属性：環境変数の値
		<LinkFiles>	リンクするファイルを定義する。
		<LinkFile>	リンクするファイルのパス。
		<RuntimeEnvironmentFile>	実行環境ファイルのパス。
		<RuntimeEnvironmentVariables>	実行時環境変数を定義する。
		<RuntimeEnvironmentVariable>	実行時環境変数。 常に空要素。 次の属性に値を設定する。 name 属性：環境変数名 value 属性：環境変数の値
		<EnableTestCaseSetting>	テストケースごとに実行環境ファイルまたは実行時環境変数を設定しているかどうか。

要素					説明
					次のどちらかの値。 ON：有効 OFF：無効
	<Stub>				［スタブ設定ページ］の内容を定義する。
		<ConstantNames>			CALL 定数で呼び出す副プログラム名を定義する。
			<ConstantName>		CALL 定数で呼び出す副プログラム名。 常に空要素。 次の属性に値を設定する。 name 属性：副プログラム名 value 属性：ON，OFF または空文字
		<IdentNames>			CALL 一意名で呼び出す副プログラム名を定義する。
			<IdentName>		CALL 一意名で呼び出す副プログラム名。
		<FileNames>			入出力するファイル名に対して，ダミーのファイルを作成するか否かを定義する。
			<FileName>		入出力するファイル名。 常に空要素。 次の属性に値を設定する。 name 属性：ファイル名 value 属性：ON，OFF または空文字
	<TestCaseSettings>				テスト環境の設定ページの内容を定義する。
		<TestCaseSetting>			テストケースごとの設定を定義する。 次の属性に値を設定する。 testcasename 属性：テストケース名
			<RuntimeEnvironmentFile>		テストケースの実行環境ファイルのパス。
			<RuntimeEnvironmentVariables>		テストケースの実行時環境変数を定義する。
				<RuntimeEnvironmentVariable>	テストケースの実行時環境変数。 次の属性に値を設定する。 name 属性：実行時環境変数名 value 属性：任意の文字列，または空文字

記述例

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<CblUnitTestSupport>
  <Property>
    <ProjectName>テストプロジェクト名</ProjectName>
    <ProgramName>プログラム名</ProgramName>
    <WorkingFolder>テストプロジェクト作業フォルダのパス</WorkingFolder>
```

```
    <COBOLSourceFile>COBOLソースファイルのパス</COBOLSourceFile>
    <TargetProgramType>Main,System または Main,V3 または Sub</TargetProgramType>
    <CompilerOptions>
      <CompilerOption>コンパイラオプション</CompilerOption>
    </CompilerOptions>
    <CompilerEnvironmentVariables>
      <CompilerEnvironmentVariable name="コンパイラ環境変数名" value="値"/>
    </CompilerEnvironmentVariables>
    <LinkFiles>
      <LinkFile>リンクするファイル</LinkFile>
    </LinkFiles>
    <RuntimeEnvironmentFile>実行環境ファイル</RuntimeEnvironmentFile>
    <RuntimeEnvironmentVariables>
      <RuntimeEnvironmentVariable name="実行時環境変数名" value="値"/>
    </RuntimeEnvironmentVariables>
    <EnableTestCaseSetting>ON または OFF</EnableTestCaseSetting>
  </Property>
  <Stub>
    <ConstantNames>
      <ConstantName name="CALL 定数で呼び出す副プログラム名" value="ON, OFFまたは空文字"/>
    </ConstantNames>
    <IdentNames>
      <IdentName>CALL 一意名で呼び出す副プログラム名</IdentName>
    </IdentNames>
    <FileNames>
      <FileName name="入出力するファイル名" value="ON, OFFまたは空文字"/>
    </FileNames>
  </Stub>
  <TestCaseSettings>
    <TestCaseSetting testcasename="テストケース名">
    <RuntimeEnvironmentFile>実行環境ファイル</RuntimeEnvironmentFile>
    <RuntimeEnvironmentVariables>
      <RuntimeEnvironmentVariable name="実行時環境変数名" value="値"/>
    </RuntimeEnvironmentVariables>
    </TestCaseSetting>
  </TestCaseSettings>
</CblUnitTestSupport>
```

12 コマンドリファレンス

単体テスト支援のコマンドの文法について説明します。

12.1　コマンドの指定形式

cbluts コマンドの指定形式，および各項目とオプションについて説明します。

cbluts コマンドの指定形式を次に示します。

```
cbluts  ターゲット
        アクション
        アクションの引数
        動作などの指示
        プロジェクトファイルの指定
```

各項目の内容と指定の規則を次に示します。

項目	内容と指定の規則
ターゲット	テストケース，実行結果など，これから処理を行う対象です。 • 必須項目です。 • 1 回の処理で，1 種類の項目しか指定できません。 • 同じものを複数指定してもエラーになりません。 • 異なるターゲットを同時に指定した場合，すべて排他エラーになります。
アクション	追加，削除など，ターゲットに対して行う処理です。 • 必須項目です。 • 1 回の処理で，1 種類の項目しか指定できません。 • 同じものを複数指定してもエラーになりません。 • 異なるアクションを指定した場合，すべて排他エラーになります。
アクションの引数	インポートする CSV ファイルパスなど，アクションを行う上で必要な引数です。 • 必須項目，または任意項目です。 • 指定できる内容，および必須か任意かは，ターゲットとアクションの組み合わせで決定します。 • 同じオプションを複数指定した場合は，後から指定したものが有効になります。
動作などの指示	テストケース名が重複した場合にどうするかなど，アクション中の動作を指定します。 • 任意項目です。 • 指定できる内容は，ターゲットとアクションの組み合わせで決定します。 • 同じオプションを複数指定した場合は，後から指定したものが有効になります。
プロジェクトファイルの指定	処理を行う対象のテストプロジェクトです。 • 必須項目です。 • 1 回の処理で，1 つのプロジェクトしか指定できません。 • 同じオプションを複数指定した場合は，後から指定したものが有効になります。

各種別に属するオプションを次に示します。

種別	オプション名	用途
ターゲット	-Test	テスト実行を操作対象とする。

種別	オプション名	用途
	-TestCase	テストケースを操作対象とする。
	-TestProject	プロジェクト情報を操作対象とする。
	-TestResult	実行結果を操作対象とする。
アクション	-ClearCoverage	カバレージ情報をクリアする。
	-Create	テストプロジェクトを新規作成する。
	-Delete	テストケース，実行結果を削除する。
	-DisplayCoverage	カバレージ情報を表示する。
	-Export	CSV ファイルなどを出力する。
	-ExportTemplate	CSV のひな形をエクスポートする。
	-Import	CSV ファイルなどを取り込む。
	-List	テストケースなどを一覧表示する。
	-Run	テスト実行する。
アクションの引数	-InputPath	入力ファイルのフォルダパスまたはファイルパスを出力する。
	-Name	テストケース名，実行結果名を指定する。
	-All	削除などをするときに，すべてを対象にする。
	-OutputPath	出力ファイルのフォルダパスまたはファイルパスを出力する。
	-Source	COBOL ソースファイルのパスを指定する。
動作などの指示	-Detail	実行結果の詳細を表示する。
	-Force	強制的に処理を続行する。
	-OverWrite	ファイルの出力で，すでに出力先のファイルがあったときに上書きする。
	-Quiet	削除系などの操作で，処理を続行してよいかの確認を抑止する。
	-Recursive	テストケースのインポートで，サブフォルダも含めて，インポートする。
	-Rename	テストケース CSV のインポートで名前が重複したとき，別名でインポートする。
	-Skip	テストケース CSV のインポート／エクスポート，プロジェクト情報ファイルの出力で名前が重複したとき，当該ファイルの入出力をスキップする。
プロジェクトファイルの指定	-Project	対象とするテストプロジェクトを指定する。

注意事項を次に示します。

- cbluts コマンドのオプションは，大文字と小文字を区別しません。
- 「-Name テストケース名」など，オプションとその引数の順番を除いて，オプションは順不同で指定できます。
- cbluts コマンドに指定するオプションは，1 つ以上の半角空白またはタブで区切って指定してください。
- テストケース名，実行結果名，およびパスをコマンドに指定する場合，それらに空白やコマンドプロンプト上での特殊文字が含まれるときは，引用符（"）で囲む必要があります。

  コマンドプロンプト上での特殊文字については，OS のヘルプを参照してください。

  また，引用符（"）を含むときは，引用符（"）で囲んで上で，「""」のように 2 つ並べて書くか，または「¥"」のように引用符（"）の前に「¥」記号を挿入してください。

指定したい文字列	指定のしかた
testcase△1	"testcase△1"
result*1	"result*1"
result"1	"result""1"または"result¥"1"

（凡例）

△：半角空白

- テストケースの削除など，幾つかの操作では，処理を続行してよいかどうかの問い合わせが発生します。問い合わせに対しては，内容に応じて次のどちらかを指定する必要があります。

  処理してよい：

  「y」または「Y」

  処理をやめる：

  「n」または「N」

  上記以外の文字列，例えば「YES」，「△Y（△は半角空白）」，全角の「Ｙ」などは指定できません。

  -Quiet オプションを指定することで問い合わせを抑止できます。

12.2　バッチモードで単体テスト支援を実行するコマンド

バッチモードで単体テスト支援を実行する cbluts コマンドの説明です。

cbluts コマンド － バッチモードで単体テスト支援を実行する

cbluts コマンドは，単体テスト支援をコマンドプロンプトから操作するときに使います。

(1) すべてのテストケースを実行する場合

形式

```
cbluts  -Test
        -Run
         〔-InputPath プロジェクト情報ファイルのパス〕
         〔-Detail〕
         〔-Force〕
        -Project テストプロジェクトファイルのパス
```

オプション

-Test

テスト実行を操作対象とする場合に指定します。

-Run

テスト実行する場合に指定します。

-InputPath プロジェクト情報ファイルのパス

プロジェクト情報ファイルのパスを指定します。オプション指定時は，設定ファイルの情報を読み込み，内容を設定したあと，テストが実行されます。

-Detail

サマリ情報に続けて，結果の詳細を出力するときに指定します。

サマリ情報は次に示す内容が出力されます。

- 総テストケース件数
- テストケースの判定結果数
- テストケースごとの判定結果一覧
- カバレージ情報

このオプションを指定すると，サマリ情報に続いて，次の結果詳細が出力されます。

- テストケースごとの判定結果
- 各テストケースの結果の値と判定結果の個数
- データ項目ごとの結果の値と判定結果の詳細
- 各テストケースの結果の代入した値の個数
- データ項目ごとの代入した値の詳細
- ファイルシミュレーションで出力文を実行したときの結果の値と判定結果
- ファイルシミュレーションで入力文を実行したときに代入した値

-Force

対応が取れないテストデータを発見した場合に，テスト実行を続行するときに指定します。この場合，対応が取れないテストデータはすべて削除されます。このため，事前にバックアップを取っておくことを推奨します。

このオプションを指定しない場合，対応が取れないテストデータを発見したときはテスト実行が中止されます。

-Project テストプロジェクトファイルのパス

実行対象のテストプロジェクトファイルのパスを指定します。

出力形式

サマリ情報の出力フォーマットを次に示します。△は半角空白を表します。

```
実行したテストプロジェクト△実行したテストプロジェクト名
△総テスト件数[実行した総テスト件数]△OK[OK判定のテストケース件数]△NG[NG判定のテストケース
件数]△未判定[未判定のテストケース件数]△エラー[エラーのテストケース件数]
△△*△テストケース名△判定結果(OK|NG|未判定)
                          :
```

また，サマリ情報に続いて，カバレージ情報が出力されます。カバレージ情報のフォーマットを次に示します。

- 数値は，整数部分 3 けた，小数部分 1 けたで出力されます。
- 整数部分のうち，3 けたに満たない場合は，先頭が空白で埋められます。
- カバレージ対象外の場合は，「---」が出力されます。

```
△カバレージ情報
△△C0メジャー:△△75.0%  // 整数部分が3けたに満たないため，先頭に空白が入る。
△△C1メジャー:△---      // カバレージ対象外のため，「---」が表示される。
△△S1メジャー:△100.0%
```

結果の詳細の出力フォーマットを次に示します。△は半角空白を表します。

```
実行したテストプロジェクト△実行したテストプロジェクト名△結果詳細
△△*△テストケース名△判定結果(OK|NG|未判定)
△△△△実行環境ファイル△実行環境ファイル名
△△△△実行時環境変数
実行時環境変数
                                    :
△△△△結果の値と判定結果△件数件
△△△△テストデータ設定個所△レベル△データ項目名△値△16進表示の値△期待値△判定結果
                                    :
△△△△代入した値△件数件
△△△△テストデータ設定個所△レベル△データ項目名△値△16進表示の値
                                    :
△△△△出力文を実行したときの結果の値と判定結果
△△△△ファイル名△オープンモード△出力文△出力文の実行回数△出力文の場所△使用した期待値/
実行した終了条件△レベル△データ項目名△値△16進表示の値△期待値△判定結果
：
△△△△入力文を実行したときに代入した値
△△△△ファイル名△オープンモード△入力文△入力文の実行回数△入力文の場所△使用した代入値/
実行した終了条件△レベル△データ項目名△代入値△16進表示の値△判定結果
：
```

終了コード

終了コード	意味	メッセージとの対応
0	正常終了しました。	E レベルおよび U レベルのエラーはありません。
1	エラーが発生して終了しました。	E レベルまたは U レベルのエラーが 1 つ以上あります。

(2) テストケースを削除する場合

形式

```
cbluts  -TestCase
        -Delete
         {-Name テストケース名 | -All}
         〔-Quiet〕
        -Project テストプロジェクトファイルのパス
```

オプション

-TestCase

テストケースを操作対象とする場合に指定します。

-Delete

テストケースを削除する場合に指定します。

-Name テストケース名

削除対象のテストケース名を指定します。

このオプションは複数指定できません。

-All オプションを指定する場合はこのオプションを指定できません。

-All

すべてのテストケースを削除する場合に指定します。

-Name オプションを指定する場合はこのオプションを指定できません。

-Quiet

確認メッセージを抑止する場合に指定します。

-Quiet オプションを指定してコマンドを実行すると，確認メッセージが表示されないで処理が続行します。

-Project テストプロジェクトファイルのパス

実行対象のテストプロジェクトファイルのパスを指定します。

終了コード

終了コード	意味	メッセージとの対応
0	正常終了しました。	E レベルおよび U レベルのエラーはありません。
1	エラーが発生して終了しました。	E レベルまたは U レベルのエラーが 1 つ以上あります。

(3) テストケースを一覧表示する場合

形式

```
cbluts  -TestCase
        -List
        -Project テストプロジェクトファイルのパス
```

オプション

-TestCase

テストケースを操作対象とする場合に指定します。

-List

テストケースを一覧表示する場合に指定します。

-Project テストプロジェクトファイルのパス

実行対象のテストプロジェクトファイルのパスを指定します。

出力形式

左から順に，テストケースの有効／無効の情報，テストケース名がテストケース名の昇順で出力されます。

```
[△ON]テストケース名
[OFF]テストケース名
```

有効なテストケースの場合は「[△ON]」，無効なテストケースの場合は「[OFF]」で表示されます（△は半角空白）。

終了コード

終了コード	意味	メッセージとの対応
0	正常終了しました。	E レベルおよび U レベルのエラーはありません。
1	エラーが発生して終了しました。	E レベルまたは U レベルのエラーが 1 つ以上あります。

(4) CSV ファイルをインポートする場合（フォルダ指定）

形式

```
cbluts  -TestCase
        -Import
        -InputPath CSVファイルが格納されているフォルダパス
         〔-Skip｜-OverWrite｜-Rename〕
         〔-Recursive〕
        -Project テストプロジェクトファイルのパス
```

オプション

-TestCase

テストケースを操作対象とする場合に指定します。

-Import

CSV ファイルをインポートする場合に指定します。

-InputPath CSV ファイルが格納されているフォルダパス

テストケースの CSV ファイルが存在するフォルダのパスを指定します。

-Skip

インポートする CSV ファイルと同じ名前のテストケースがすでに存在する場合に，そのファイルのインポートをスキップします。

このオプションは，-OverWrite オプション，-Rename オプションと同時に指定できません。

-Skip オプション，-OverWrite オプション，-Rename オプションのどれも指定しなかった場合は，-Skip オプションが仮定されます。

-OverWrite

インポートする CSV ファイルと同じ名前のテストケースがすでに存在する場合に，CSV ファイルでそのテストケースを上書きします。

このオプションは，-Skip オプション，-Rename オプションと同時に指定できません。

-Skip オプション，-OverWrite オプション，-Rename オプションのどれも指定しなかった場合は，-Skip オプションが仮定されます。

-Rename

このオプションを指定すると，インポートする CSV ファイルと同じ名前のテストケースがすでに存在する場合に，テストケースの名前を変更（名前の末尾に通し番号を追加）してファイルが取り込まれます。

このオプションは，-Skip オプション，-OverWrite オプションと同時に指定できません。

-Skip オプション，-OverWrite オプション，-Rename オプションのどれも指定しなかった場合は，-Skip オプションが仮定されます。

-Recursive

-InputPath オプションで指定したフォルダのサブフォルダも含めてインポートする場合に指定します。

-Project テストプロジェクトファイルのパス

実行対象のテストプロジェクトファイルのパスを指定します。

出力形式

- ファイルの取り込み中に表示されるメッセージの横に，進捗状況が［完了したファイル数／対象のファイルの総数］の形式で表示されます。
- 「完了したファイル数」の桁数が，「対象のファイルの総数」の総数より少ない場合は，先頭を半角空白で埋めて表示されます。
- 1 つのファイルに対してメッセージが複数出力される場合は，すべてのメッセージの先頭に，［完了したファイル数／対象のファイルの総数］が付きます。
- すべての CSV のインポートが完了したときに表示されるメッセージ（KEDL1029T-I）の横には，進捗状況は表示されません。

出力フォーマットを次に示します。△は半角空白を表します。

```
[△△n/nnn]メッセージ
    :
KEDL1029T-I テストケースCSVのインポートが完了しました。
```

終了コード

終了コード	意味	メッセージとの対応
0	正常終了しました。	インポート処理を中断するようなEレベルおよびUレベルのエラーはありません。
1	エラーが発生してインポート処理が中断しました。	インポート処理を中断させるEレベルまたはUレベルのエラーが1つ以上あります。

(5) CSVファイルをインポートする場合（ファイル指定）

形式

```
cbluts  -TestCase
        -Import
        -InputPath CSVファイルのパス
         〔-Skip｜-OverWrite｜-Rename〕
        -Project テストプロジェクトファイルのパス
```

オプション

-TestCase

テストケースを操作対象とする場合に指定します。

-Import

CSVファイルをインポートする場合に指定します。

-InputPath CSVファイルのパス

テストケースのCSVファイルのパスを指定します。

-Skip

インポートするCSVファイルと同じ名前のテストケースがすでに存在する場合に，そのファイルのインポートをスキップします。

このオプションは，-OverWriteオプション，-Renameオプションと同時に指定できません。

-Skipオプション，-OverWriteオプション，-Renameオプションのどれも指定しなかった場合は，-Skipオプションが仮定されます。

-OverWrite

インポートするCSVファイルと同じ名前のテストケースがすでに存在する場合に，CSVファイルでそのテストケースを上書きします。

このオプションは，-Skipオプション，-Renameオプションと同時に指定できません。

-Skip オプション，-OverWrite オプション，-Rename オプションのどれも指定しなかった場合は，-Skip オプションが仮定されます。

-Rename

このオプションを指定すると，インポートする CSV ファイルと同じ名前のテストケースがすでに存在する場合に，テストケースの名前を変更（名前の末尾に通し番号を追加）してファイルが取り込まれます。

このオプションは，-Skip オプション，-OverWrite オプションと同時に指定できません。

-Skip オプション，-OverWrite オプション，-Rename オプションのどれも指定しなかった場合は，-Skip オプションが仮定されます。

-Project テストプロジェクトファイルのパス

実行対象のテストプロジェクトファイルのパスを指定します。

出力形式

「(4) CSV ファイルをインポートする場合（フォルダ指定）」の場合と同じです。

終了コード

終了コード	意味	メッセージとの対応
0	正常終了しました。	インポート処理を中断するような E レベルおよび U レベルのエラーはありません。
1	エラーが発生してインポート処理が中断しました。	インポート処理を中断させる E レベルまたは U レベルのエラーが 1 つ以上あります。

(6) テストケースをエクスポートする場合（テストケース指定）

形式

```
cbluts  -TestCase
        -Export
         {-Name テストケース名 | -All}
        -OutputPath 出力先フォルダパス
         〔-Skip | -OverWrite〕
        -Project テストプロジェクトファイルのパス
```

オプション

-TestCase

テストケースを操作対象とする場合に指定します。

-Export

テストケースをエクスポートする場合に指定します。

-Name テストケース名

エクスポートするテストケース名を指定します。

このオプションは複数指定できません。

-All オプションを指定する場合はこのオプションを指定できません。

-All

すべてのテストケースをエクスポートする場合に指定します。

-Name オプションを指定する場合はこのオプションを指定できません。

-OutputPath 出力先フォルダパス

出力先フォルダのパスを指定します。

-Skip

エクスポートするテストケースと同じ名前の CSV ファイルがすでに存在する場合に，そのテストケースのエクスポートをスキップします。

このオプションは，-OverWrite オプションと同時に指定できません。

-Skip オプション，-OverWrite オプションのどちらも指定しなかった場合は，-Skip オプションが仮定されます。

-OverWrite

エクスポートするテストケースと同じ名前の CSV ファイルがすでに存在する場合に，そのテストケースで CSV ファイルを上書きします。

このオプションは，-Skip オプションと同時に指定できません。

-Skip オプション，-OverWrite オプションのどちらも指定しなかった場合は，-Skip オプションが仮定されます。

-Project テストプロジェクトファイルのパス

実行対象のテストプロジェクトファイルのパスを指定します。

出力形式

CSV ファイルのインポートの場合と同じです。

終了コード

終了コード	意味	メッセージとの対応
0	正常終了しました。	エクスポート処理を中断するような E レベルおよび U レベルのエラーはありません。

終了コード	意味	メッセージとの対応
1	エラーが発生してエクスポート処理が中断しました。	エクスポート処理を中断させる E レベルまたは U レベルのエラーが 1 つ以上あります。

(7) テストケースをエクスポートする場合（ひな形）

形式

```
cbluts  -TestCase
        -ExportTemplate
        -OutputPath 出力先フォルダパス
         〔-Skip｜-OverWrite〕
        -Project テストプロジェクトファイルのパス
```

オプション

-TestCase

テストケースを操作対象とする場合に指定します。

-ExportTemplate

テストケースのひな形をエクスポートする場合に指定します。

-OutputPath 出力先フォルダパス

出力先フォルダのパスを指定します。

-Skip

ひな形.csv ファイルがすでに存在する場合に，エクスポートをスキップします。

このオプションは，-OverWrite オプションと同時に指定できません。

-Skip オプション，-OverWrite オプションのどちらも指定しなかった場合は，-Skip オプションが仮定されます。

-OverWrite

ひな形.csv ファイルがすでに存在する場合に，そのひな形.csv を上書きします。

このオプションは，-Skip オプションと同時に指定できません。

-Skip オプション，-OverWrite オプションのどちらも指定しなかった場合は，-Skip オプションが仮定されます。

-Project テストプロジェクトファイルのパス

実行対象のテストプロジェクトファイルのパスを指定します。

出力形式

CSV ファイルのインポートの場合と同じです。

終了コード

終了コード	意味	メッセージとの対応
0	正常終了しました。	エクスポート処理を中断するような E レベルおよび U レベルのエラーはありません。
1	エラーが発生してエクスポート処理が中断しました。	エクスポート処理を中断させる E レベルまたは U レベルのエラーが 1 つ以上あります。

(8) プロジェクトファイルを作成する場合

形式

```
cbluts  -TestProject
        -Create
        -Source COBOLソースファイルのパス
         〔-InputPath プロジェクト情報ファイルのパス〕
        -Project テストプロジェクトファイルのパス
```

オプション

-TestProject

テストプロジェクトが操作対象であることを示します。

-Create

テストプロジェクトを作成する場合に指定します。

-Source COBOL ソースファイルのパス

COBOL ソースファイルのパスを指定します。

-InputPath プロジェクト情報ファイルのパス

プロジェクト情報ファイルのパスを指定します。

-Project テストプロジェクトファイルのパス

実行対象のテストプロジェクトファイルのパスを指定します。ここで指定したパスでプロジェクトファイルが作成されます。

ファイル形式

プロジェクト情報ファイルに記述する各項目については，「8.4.3　プロジェクト情報を設定する」を参照してください。ただし，< COBOLSourceFile >要素は指定していても無視される点で，動作が異なります。

終了コード

終了コード	意味	メッセージとの対応
0	正常終了しました。	E レベルおよび U レベルのエラーはありません。
1	エラーが発生して終了しました。	E レベルまたは U レベルのエラーが 1 つ以上あります。

(9) プロジェクト情報ファイルを出力する場合

形式

```
cbluts  -TestProject
        -Export
        -OutputPath プロジェクト情報ファイルのパス
         〔-Skip | -OverWrite〕
        -Project テストプロジェクトファイルのパス
```

オプション

-TestProject

テストプロジェクトが操作対象であることを示します。

-Export

プロジェクト情報ファイルをエクスポートする場合に指定します。

-OutputPath プロジェクト情報ファイルのパス

出力先のプロジェクト情報ファイルのパスを指定します。

-Skip

プロジェクト情報ファイルがすでに存在する場合に，エクスポートをスキップします。

このオプションは，-OverWrite オプションと同時に指定できません。

-Skip オプション，-OverWrite オプションのどちらも指定しなかった場合は，-Skip オプションが仮定されます。

-OverWrite

プロジェクト情報ファイルがすでに存在する場合に，そのファイルを上書きします。

このオプションは，-Skip オプションと同時に指定できません。

-Skip オプション，-OverWrite オプションのどちらも指定しなかった場合は，-Skip オプションが仮定されます。

-Project テストプロジェクトファイルのパス

実行対象のテストプロジェクトファイルのパスを指定します。

出力形式

プロパティ情報は，設定されていないものも含めてすべて出力されます。

要素				備考
<CblUnitTestSupport>				−
	<Property>			−
		<ProjectName>		−
		<ProgramName>		−
		<WorkingFolder>		−
		<COBOLSourceFile>		−
		<TargetProgramType>		−
		<CompilerOptions>		コンパイラオプションがないときも出力されます。
			<CompilerOption>	コンパイラオプションの指定がないときは，空の要素が 1 つ出力されます。 複数のコンパイラオプションが指定されているときは，1 つの要素にまとめて出力されます。
		<CompilerEnvironmentVariables>		コンパイラ環境変数がないときも出力されます。
			<CompilerEnvironmentVariable>	コンパイラ環境変数の指定がないときは，空の要素が 1 つ出力されます。 複数のコンパイラ環境変数が指定されている場合は，1 つの設定につき 1 つの要素で出力されます。
		<LinkFiles>		リンクするファイルの指定がないときも出力されます。
			<LinkFile>	リンクするファイルが指定されていないときは，空要素が 1 つ出力されます。 複数のファイルが指定されている場合は，1 つのファイルにつき 1 つの要素で出力されます。
		<RuntimeEnvironmentFile>		実行環境ファイルの指定がない場合は，空要素として出力されます。
		<RuntimeEnvironmentVariables>		実行時環境変数がないときも出力されます。
			<RuntimeEnvironmentVariable>	実行時環境変数の指定がないときは，空の要素が 1 つ出力されます。

要素				備考
	<Stub>			スタブの情報がないときも出力されます。
		<ConstantNames>		「CALL 定数で呼び出す副プログラム名」の対象がない場合でも出力されます。
			<ConstantName>	「CALL 定数で呼び出す副プログラム名」の対象がない場合は，空要素が 1 つ出力されます。 対象のプログラムのチェックが ON，OFF に関係なく出力されます。1 つのプログラム名につき，1 つの要素で出力されます。
		<IdentNames>		「CALL 一意名で呼び出す副プログラム名」の対象がない場合でも出力されます。
			<IdentName>	「CALL 一意名で呼び出す副プログラム名」の対象がない場合は，空要素が出力されます。 対象のプログラムのチェックが ON，OFF に関係なく出力されます。1 つのプログラム名につき，1 つの要素で出力されます。
		<FileNames>		「入出力するファイル名」の対象がなくても，出力されます。
			<FileName>	「入出力するファイル名」の対象がない場合は，空要素が出力されます。 対象のファイルのチェックが ON，OFF に関係なく出力されます。1 つのファイル名につき，1 つの要素で出力されます。

（凡例）

－：特にありません。

各要素の説明は，「11.2 プロジェクト情報ファイルの構成要素」を参照してください。

終了コード

終了コード	意味	メッセージとの対応
0	正常終了しました。	E レベルおよび U レベルのエラーはありません。
1	エラーが発生して終了しました。	E レベルまたは U レベルのエラーが 1 つ以上あります。

(10) プロジェクト情報を設定する場合

形式

```
cbluts  -TestProject
        -Import
        -InputPath プロジェクト情報ファイルのパス
         〔-Quiet〕
        -Project テストプロジェクトファイルのパス
```

オプション

-TestProject

テストプロジェクトが操作対象であることを示します。

-Import

プロジェクト情報を設定する場合に指定します。

-InputPath プロジェクト情報ファイルのパス

設定する内容が記述されたプロジェクト情報ファイルのパスを指定します。

-Quiet

確認メッセージを抑止する場合に指定します。

-Quiet オプションを指定してコマンドを実行すると，確認メッセージが表示されないで処理が続行します。

-Project テストプロジェクトファイルのパス

実行対象のテストプロジェクトファイルのパスを指定します。

ファイル形式

各要素の説明は，「11.2 プロジェクト情報ファイルの構成要素」を参照してください。

終了コード

終了コード	意味	メッセージとの対応
0	正常終了しました。	E レベルおよび U レベルのエラーはありません。
1	エラーが発生して終了しました。	E レベルまたは U レベルのエラーが 1 つ以上あります。

(11) カバレージ情報を表示する場合

形式

```
cbluts  -TestResult
        -DisplayCoverage
        -Project テストプロジェクトファイルのパス
```

オプション

-TestResult

実行結果が操作対象であることを示します。

-DisplayCoverage

カバレージの情報を表示する場合に指定します。

-Project テストプロジェクトファイルのパス

実行対象のテストプロジェクトファイルのパスを指定します。

出力形式

カバレージ情報が出力されます。カバレージ情報のフォーマットを次に示します（△は半角空白）。

- 数値は，整数部分 3 けた，小数部分 1 けたで出力されます。
- 整数部分のうち，3 けたに満たない場合は，先頭が空白で埋められます。
- カバレージ対象外の場合は，「---」が出力されます。

```
△カバレージ情報
△△C0メジャー:△△75.0%  // 整数部分が3けたに満たないため，先頭に空白が入る。
△△C1メジャー:△---      // カバレージ対象外のため，「---」が表示される。
△△S1メジャー:△100.0%
```

終了コード

終了コード	意味	メッセージとの対応
0	正常終了しました。	E レベルおよび U レベルのエラーはありません。
1	エラーが発生して終了しました。	E レベルまたは U レベルのエラーが 1 つ以上あります。

(12) カバレージ情報をクリアする場合

形式

```
cbluts  -TestResult
        -ClearCoverage
         〔-Quiet〕
        -Project テストプロジェクトファイルのパス
```

オプション

-TestResult

実行結果が操作対象であることを示します。

-ClearCoverage

カバレージ情報をクリアする場合に指定します。

-Quiet

確認メッセージを抑止する場合に指定します。

-Quiet オプションを指定してコマンドを実行すると，確認メッセージが表示されないで処理が続行します。

-Project テストプロジェクトファイルのパス

実行対象のテストプロジェクトファイルのパスを指定します。

終了コード

終了コード	意味	メッセージとの対応
0	正常終了しました。	E レベルおよび U レベルのエラーはありません。
1	エラーが発生して終了しました。	E レベルまたは U レベルのエラーが 1 つ以上あります。

(13) 実行結果を削除する場合

形式

```
cbluts  -TestResult
        -Delete
         {-Name 実行結果名 | -All}
         〔-Quiet〕
        -Project テストプロジェクトファイルのパス
```

オプション

-TestResult

実行結果が操作対象であることを示します。

-Delete

実行結果を削除する場合に指定します。

-Name 実行結果名

削除対象の実行結果名を指定します。

このオプションは複数指定できません。

-All オプションを指定する場合はこのオプションを指定できません。

-All

すべての実行結果を削除する場合に指定します。

-Name オプションを指定する場合はこのオプションを指定できません。

-Quiet

確認メッセージを抑止する場合に指定します。

-Quiet オプションを指定してコマンドを実行すると，確認メッセージが表示されないで処理が続行します。

-Project テストプロジェクトファイルのパス

実行対象のテストプロジェクトファイルのパスを指定します。

終了コード

終了コード	意味	メッセージとの対応
0	正常終了しました。	E レベルおよび U レベルのエラーはありません。
1	エラーが発生して終了しました。	E レベルまたは U レベルのエラーが 1 つ以上あります。

(14) 実行結果を一覧表示する場合

形式

```
cbluts  -TestResult
        -List
        -Project テストプロジェクトファイルのパス
```

オプション

-TestResult

実行結果が操作対象であることを示します。

-List

実行結果を一覧表示する場合に指定します。

-Project テストプロジェクトファイルのパス

実行対象のテストプロジェクトファイルのパスを指定します。

出力形式

名前で昇順に並べ替えて，1 行で 1 つの実行結果が出力されます。

```
実行結果の名前
```

終了コード

終了コード	意味	メッセージとの対応
0	正常終了しました。	E レベルおよび U レベルのエラーはありません。
1	エラーが発生して終了しました。	E レベルまたは U レベルのエラーが 1 つ以上あります。

13 エラー時の対処

単体テスト支援でエラーが起こったときに対処について説明します。

13.1　メッセージから探す

単体テスト支援の操作で出力されるメッセージについての説明です。

コンパイラ，テストデバッガ，および COBOL2002 の実行時のエラーメッセージ（KCCC で始まるメッセージ）については，マニュアル「COBOL2002 メッセージ」を参照してください。

13.1.1　メッセージの出力形式

単体テスト支援を実行したときに出力されるメッセージの出力形式です。

形式

KEDL*nnnn*T-*i XX...XX*

形式の説明

- KEDL*nnnn*T-*i* ：メッセージ ID を示します。「*i* 」はメッセージの重要度を示します。メッセージの重要度には次の 4 種類があります。

  「U」：回復不能のエラーです。処理は中止されます。

  「E」：エラーメッセージです。正常に動作しないトラブルが起こったことを示します。

  「W」：警告メッセージです。リソースの使用状況などについての警告，またはコマンドの指定誤りに対して値を仮定して処理を続行することを示します。

  「I」：インフォメーションメッセージです。「U」「E」「W」に該当しない単純な動作状況を示します。
- *XX...XX* ：メッセージテキストを示します。

13.1.2　メッセージの一覧

単体テスト支援の操作で出力されるメッセージと対処方法です。

(1)　単体テスト支援共通のメッセージ（KEDL0000～KEDL0999）と対処

表 13–1　単体テスト支援共通のメッセージ（KEDL0000～KEDL0999）と対処

メッセージ ID	メッセージテキスト	対処
KEDL0001T-U	COBOL2002 Professional Tool Kit のレジストリ情報が不正です。	COBOL2002 Professional Tool Kit のレジストリ情報が取得できません。 単体テスト支援を終了します。 COBOL2002 Professional Tool Kit をアンインストールしてから，再インストールしてください。
KEDL0002T-U	前提の COBOL2002 Net Developer のレジストリ情報が不正です。	前提の COBOL2002 Net Developer のレジストリ情報が取得できません。

メッセージ ID	メッセージテキスト	対処
		単体テスト支援を終了します。 COBOL2002 Net Developer をアンインストールしてから，再インストールしてください。
KEDL0003T-U	前提の COBOL2002 Net Developer のバージョンが不正です。	前提 COBOL のバージョンが異なります。 単体テスト支援を終了します。 COBOL2002 Net Developer をアンインストールしてから，再インストールしてください。
KEDL0005T-U	前提の COBOL2002 Net Developer のツール(COBOL エディタ)を起動できませんでした。	COBOL エディタが起動できません。 単体テスト支援を終了します。 COBOL2002 Net Developer をアンインストールしてから，再インストールしてください。
KEDL0006T-U	前提の COBOL2002 Net Developer のツール(テストデバッガ)を起動できませんでした。	テストデバッガ（バッチモード）が起動できません。 単体テスト支援を終了します。 COBOL2002 Net Developer をアンインストールしてから，再インストールしてください。
KEDL0007T-U	前提の COBOL2002 Net Developer のツール(カバレージ)を起動できませんでした。	カバレージが起動できません。 単体テスト支援を終了します。 COBOL2002 Net Developer をアンインストールしてから，再インストールしてください。
KEDL0008T-E	指定された COBOL ソースファイルにアクセスできません。指定したパスが正しいか，またはアクセス権限が付与されているかを確認してください。(*** 1 ***)	COBOL ソースファイルにアクセスできません。 処理を中止します。 ファイルのパスが正しいか，またはファイルのアクセス権限が付与されているかを確認してください。
KEDL0009T-E	指定されたテストプロジェクト作業フォルダにアクセスできません。指定したパスが正しいか，またはアクセス権限が付与されているかを確認してください。(*** 1 ***)	テストプロジェクト作業フォルダにアクセスできません。 処理を中止します。 フォルダのパスが正しいか，またはフォルダのアクセス権限が付与されているかを確認してください。
KEDL0010T-E	指定されたテストプロジェクト作業フォルダを作成できません。指定したパスが正しいか，またはアクセス権限が付与されているかを確認してください。(*** 1 ***)	テストプロジェクト作業フォルダを作成できません。 処理を中止します。 フォルダのパスが正しいか，またはフォルダのアクセス権限が付与されているかを確認してください。
KEDL0011T-E	すでに別の単体テスト支援で開いているため，指定されたテストプロジェクトを開くことができません。(*** 1 ***)	指定したテストプロジェクトが別の単体テスト支援ですでに開かれています。 処理を中止します。 別の単体テスト支援で指定したテストプロジェクトをすでに開いていないかを確認してください。
KEDL0012T-E	指定されたテストプロジェクトのバージョンに対応していないため，開くことができません。(*** 1 ***)	指定したテストプロジェクトのバージョンがこの単体テスト支援では開けません。 処理を中止します。

メッセージ ID	メッセージテキスト	対処
		インストールしている COBOL2002 Professional Tool Kit のバージョンを確認してください。
KEDL0013T-E	テストプロジェクトの内容が不正です。(*** 1 ***)	指定したテストプロジェクトの内容が不正です。 処理を中止します。 テストプロジェクトファイルを再作成してください。再作成しても問題が解消しない場合には，当社保守員に連絡してください。
KEDL0014T-E	テストプロジェクト名に使用できない文字が入力されました。(*** 1 ***)	テストプロジェクト名に使用できない文字が指定されました。 処理を中止します。 使用できない文字を入力しないでください。
KEDL0015T-E	テストプロジェクト名の長さが長すぎます。(*** 1 ***)	テストプロジェクト名の長さが上限値を超えました。 処理を中止します。 テストプロジェクト名の長さを短くしてください。 *テストプロジェクト作業フォルダ*¥*テストプロジェクト名*.utsproj が，OS のファイルパス長の上限値を超えないようにしてください。
KEDL0016T-E	指定したテストプロジェクト作業フォルダに，すでにテストプロジェクトが存在します。テストプロジェクトが存在しないフォルダを指定してください。(*** 1 ***)	保存先のテストプロジェクト作業フォルダにすでにテストプロジェクトが存在します。 処理を中止します。 テストプロジェクトが存在しないフォルダをテストプロジェクト作業フォルダとして指定してください。
KEDL0017T-E	テストプロジェクトの保存に失敗しました。テストプロジェクト作業フォルダやテストプロジェクト作業フォルダに含まれるファイルにアクセス権限や書き込み権限があるかを確認してください。(*** 1 ***)	テストプロジェクトファイルが保存できません。 処理を中止します。 テストプロジェクト作業フォルダや，そのフォルダ/サブフォルダにあるファイルに対して書き込み権限があるかを確認してください。
KEDL0018T-E	テスト対象の COBOL ソースファイルの解析でコンパイルエラーが発生しました。	COBOL ソースファイルの解析でコンパイルエラーが発生しました。 処理を中止します。 コンパイルエラーのエラー内容を参照し，テスト対象の COBOL ソースファイルの内容を修正してください。
KEDL0019T-E	テストケース名に使用できない文字が入力されました。(*** 1 ***)	テストケース名で使用できない文字が指定されました。 処理を中止します。 使用できない文字を入力しないでください。
KEDL0020T-E	テストケース名の長さが長すぎます。(*** 1 ***)	テストケース名の長さが上限値を超えました。 処理を中止します。 *テストプロジェクト作業フォルダ*¥Tdi¥*テストケース名*.tdi が，OS のファイルパス長の上限値を超えないようにしてください。

メッセージ ID	メッセージテキスト	対処
KEDL0021T-E	テストプロジェクトに含められるテストケースの個数の上限を超えました。	テストケースの個数が上限値を超えました。 処理を中止します。 不要なテストケースを削除するか，新しくテストプロジェクトを作成してテストケースを追加してください。
KEDL0022T-E	テストケース TD コマンド格納ファイルを削除できません。テストプロジェクト作業フォルダ以下のファイルを別プログラムで開いていないか，もしくはテストプロジェクト作業フォルダ以下のフォルダやファイルにアクセス権限や削除権限が付与されているかを確認してください。(*** 1 ***)	テストケース TD コマンド格納ファイルを削除できません。 処理を中止します。 テストケース TD コマンド格納ファイルのアクセス権限が付与されているかを確認してください。
KEDL0023T-E	CSV インポートまたはエクスポート対象のファイルにアクセスできません。(*** 1 ***)	CSV インポートまたはエクスポート実行時に，指定したファイルにアクセスできません。 処理を中止します。 対象のファイルにアクセス権限が付与されているかを確認してください。
KEDL0024T-E	CSV ファイルを出力できません。(*** 1 ***)	CSV ファイルを出力できません。 処理を中止します。 対象のフォルダのパスが正しいか，またはそのフォルダのアクセス権限が付与されているかを確認してください。
KEDL0025T-E	不正なテストケース CSV ファイルが入力されました。(*** 1 ***)	テストケース CSV のフォーマットに合わない CSV ファイルが入力されました。 処理を中止します。 テストケース CSV ファイルの内容を確認し，正しくなるように修正してください。
KEDL0026T-E	適用できないテストデータがテストケース CSV ファイルに含まれていました。(テストケース CSV : *** 1 *** 行 : *** 2 *** 詳細情報 : *** 3 ***)	テストケースとして反映できないテストデータの設定が CSV ファイルに記述されています。 該当の行をスキップします。 テストケース CSV ファイルの内容を確認し，正しくなるように修正してください。
KEDL0027T-E	1 つのテストケースに設定可能な中断点の個数の上限を超えました。(*** 1 ***)	中断点の設定可能な上限値を超えました。 処理を中止します。 CSV ファイルのインポートのときは，該当の行をスキップします。 不要な中断点を削除して，中断点を設定してください。
KEDL0028T-E	テスト対象の実行可能ファイルの生成でコンパイルエラーが発生しました。	テスト対象の実行可能ファイルが生成できません。 処理を中止します。 コンパイルエラーのエラー内容を参照し，テスト対象の COBOL ソースファイルの内容を修正してください。

メッセージ ID	メッセージテキスト	対処
KEDL0029T-E	TD コマンド格納ファイルを生成できません。テストプロジェクト作業フォルダ以下のフォルダやファイルにアクセス権限や書き込み権限が付与されているかを確認してください。	TD コマンド格納ファイルが生成できません。 処理を中止します。 テストプロジェクト作業フォルダ以下のフォルダやファイルにアクセス権限が付与されているかを確認してください。
KEDL0030T-E	実行結果を格納するフォルダを生成できません。テストプロジェクト作業フォルダ以下のフォルダやファイルにアクセス権限や書き込み権限が付与されているかを確認してください。	実行結果を格納するフォルダが生成できません。 処理を中止します。 テストプロジェクト作業フォルダ以下のフォルダやファイルにアクセス権限が付与されているかを確認してください。
KEDL0031T-W	単体テストの実行が失敗しました。(*** 1 ***)	テストケースの単体テストの実行が正常に終了しませんでした。 失敗したテストケースの単体テストの実行をスキップします。 単体テストの実行に失敗したテストケースのテストデータの設定に間違いがないかを確認してください。原因不明の場合は，当社保守員に連絡してください。
KEDL0034T-E	エビデンスファイルを削除できません。テストプロジェクト作業フォルダ以下のファイルを別プログラムで開いていないか，もしくはテストプロジェクト作業フォルダ以下のフォルダやファイルにアクセス権限や削除権限が付与されているかを確認してください。(*** 1 ***)	エビデンスファイルを削除できません。 処理を中止します。 テストプロジェクト作業フォルダ以下のフォルダやファイルにアクセス権限が付与されているかを確認してください。
KEDL0035T-E	指定された実行環境ファイルにアクセスできません。指定したパスが正しいか，またはアクセス権限が付与されているかを確認してください。(*** 1 ***)	実行環境ファイルにアクセスできません。 処理を中止します。 ファイルのパスが正しいか，またはファイルのアクセス権限が付与されているかを確認してください。
KEDL0036T-E	指定された COBOL ソースファイルに，テスト対象になる翻訳単位がありません。テスト対象となる翻訳単位が 1 つ以上定義された COBOL ソースファイルを指定してください。	指定した COBOL ソースファイルを解析した結果，テスト対象となる翻訳単位が 1 つも存在しません。 処理を中止します。 テスト対象となる翻訳単位が 1 つ以上存在する COBOL ソースファイルを指定してください。
KEDL0037T-E	テストプロジェクトに含められる実行結果の個数の上限を超えました。	実行結果の個数が上限値を超えました。 処理を中止します。 不要な実行結果をクリアまたは削除してください。
KEDL0038T-E	プログラム情報ファイルにアクセスできません。テストプロジェクト作業フォルダ以下のフォルダやファイルにアクセス権限が付与されているかを確認してください。(*** 1 ***)	プログラム情報ファイルにアクセスできません。 処理を中止します。 ファイルが存在するか，またはファイルにアクセス権限が付与されているかどうかを確認してください。 解決できない場合は，当社保守員に連絡してください。
KEDL0039T-U	前提の COBOL2002 Net Developer のツール(コンパイラ)を起動できませんでした。	コンパイラが起動できません。 単体テスト支援を終了します。

メッセージ ID	メッセージテキスト	対処
		COBOL2002 Net Developer をアンインストールしてから，再インストールしてください。
KEDL0040T-E	COBOL ソースファイルにテスト対象のプログラム(*** 1 ***)が存在しません。テスト対象のプログラム名を変更する場合は，新規にテストプロジェクトを作成してください。	プログラム名に指定されたプログラム定義が COBOL ソースファイルにありません。 処理を中止します。 テスト対象の COBOL ソースファイルにテストプロジェクト新規作成時に設定したプログラムが含まれるように修正し，COBOL ソースファイルの解析を再実行してください。または，新規にテストプロジェクトを作成してください。
KEDL0041T-E	テストデータとして入力した形式が不正です。(*** 1 ***)	テストデータの入力形式が不正です。入力形式の不正チェックを行うのは次に示す文字から始まる入力形式の場合だけです。 • @ • & 処理を中止します。 該当するデータ項目の入力値を形式に合うように修正してください。
KEDL0042T-E	64bit 版の単体テスト支援で作成したテストプロジェクトを 32bit 版の単体テスト支援で開くことはできません。	テストプロジェクトが読み込めません（64bit 版の COBOL2002 Developer Professional の単体テスト支援で作成したテストプロジェクトを 32bit 版の COBOL2002 Developer Professional の単体テスト支援で読み込もうとした場合）。 処理を中止します。 32bit 版の COBOL2002 で新規にテストプロジェクトを作成してください。
KEDL0043T-E	32bit 版の単体テスト支援で作成したテストプロジェクトを 64bit 版の単体テスト支援で開くことはできません。	テストプロジェクトが読み込めません（32bit 版の COBOL2002 Developer Professional の単体テスト支援で作成したテストプロジェクトを 64bit 版の COBOL2002 Developer Professional の単体テスト支援で読み込もうとした場合）。 処理を中止します。 64bit 版の COBOL2002 で新規にプロジェクトを作成してください。
KEDL0044T-E	単体テスト用の実行可能ファイルにアクセスできません。コンパイラオプションに-Compile オプションを付与していないか，またはテストプロジェクト作業フォルダ以下のフォルダやファイルにアクセス権限が付与されているかを確認してください。(*** 1 ***)	単体テスト用の実行可能ファイルが存在しないか，アクセス権限がありません。 処理を中止します。 テストプロジェクト作業フォルダ以下のフォルダやファイルにアクセス権限が付与されているかを確認してください。または，コンパイラオプションに-Compile,{CheckOnly|NoLink}を付与していないかを確認してください。
KEDL0045T-E	テスト対象の COBOL ソースファイル名(*** 1 ***)は変更できません。別名の COBOL ソース	テスト対象の COBOL ソースファイル名を変更しようとしました。

メッセージ ID	メッセージテキスト	対処
	ファイルをテスト対象とする場合は，新しくテストプロジェクトを作成してください。	処理を中止します。 新しくテストプロジェクトを作成してください。
KEDL0046T-W	環境変数 CBLUTS_ANALYZEMAXOCCURSNUM の指定内容に誤りがあります。500 を仮定します。	環境変数 CBLUTS_ANALYZEMAXOCCURSNUM には，1～500 の範囲の数値以外は指定できません。 500 を仮定します。 環境変数 CBLUTS_ANALYZEMAXOCCURSNUM に，1～500 の範囲内の数値を指定して，単体テスト支援を再起動してください。
KEDL0047T-I	古いバージョンの単体テスト支援で作成したテストプロジェクトファイルを開きました。保存または単体テストを実行すると，古いバージョンの単体テスト支援では開けなくなります。	古いバージョンの単体テスト支援で作成したテストプロジェクトファイルを開きました。 保存または単体テストを実行すると，古いバージョンの単体テスト支援では開けなくなります。
KEDL0048T-W	OCCURS 指定ありデータ項目の 1 つの次元でテストデータを設定できる個数は，現在は*** 1 ***ですが，COBOL ソースファイルの解析を実行すると*** 2 ***になります。	OCCURS 指定ありデータ項目の 1 つの次元でテストデータを設定できる個数が，前回テストプロジェクトファイルを保存したときの値と，現在の値が異なります。 上限値を変更しない場合には，環境変数 CBLUTS_ANALYZEMAXOCCURSNUM に，前回テストプロジェクトファイルを保存したときの値を設定して，単体テスト支援を再起動してください。
KEDL0049T-W	stdcall 呼び出し規約で呼び出すプログラムが見つかりました。stdcall 呼び出し規約で呼び出すプログラムのスタブは生成できません。テスト対象プログラムにリンクするオブジェクトファイル(.obj)またはライブラリファイル(.lib)を，[プロパティページ]の[リンクするファイル]に指定してください。	テスト対象の COBOL ソースファイル内に，スタブを生成できない stdcall 呼び出し規約で呼び出すプログラムがありました。 stdcall 呼び出し規約で呼び出すプログラムのオブジェクトファイル（.obj）またはライブラリファイル（.lib）を［プロパティページ］の［リンクするファイル］に指定するか，単体テストの実行中は cdecl 呼び出し規約にして，単体テストの終了後に stdcall 呼び出し規約に変更してください。
KEDL0050T-E	代入値を設定するデータ項目数が上限値を超えました。(テストケース CSV : *** 1 *** 行 : *** 2 ***)	代入値を設定するデータ項目数が上限値を超えました。 該当の行をスキップします。 不要な代入値の設定を解除してから再設定してください。
KEDL0051T-E	期待値を設定するデータ項目数が上限値を超えました。(テストケース CSV : *** 1 *** 行 : *** 2 ***)	期待値を設定するデータ項目数が上限値を超えました。 該当の行をスキップします。 不要な期待値の設定を解除してから再設定してください。
KEDL0052T-E	テストプロジェクト作業フォルダのパスにコンマ(,)を使用しないでください。(*** 1 ***)	テストプロジェクト作業フォルダに利用できない文字を指定しています。 処理を中止します。 コンマ（,）を含まないパス名を指定してください。

メッセージ ID	メッセージテキスト	対処
KEDL0053T-E	COBOL ソースファイルのファイル名にコンマ(,)を使用しないでください。(*** 1 ***)	COBOL ソースファイルのファイル名に利用できない文字が指定しています。 処理を中止します。 ファイル名にコンマ（,）を含まないパス名を指定してください。
KEDL0054T-E	コンマ(,)を含むパスにあるテストプロジェクトは使えません。コンマ(,)を含まないフォルダに移動させてください。(*** 1 ***)	コンマ（,）を含むフォルダに移動したテストプロジェクトを指定しました。 処理を中止します。 コンマ（,）を含まないフォルダにテストプロジェクトを移動してから使用してください。
KEDL0055T-E	インポート後のテストケース名の長さが長すぎるため，テストケース CSV をインポートできません。(*** 1 ***)	インポート後のテストケース名の長さが上限値を超えました。 処理を中止します。 *テストプロジェクト作業フォルダ*¥Tdi¥*テストケース名*（*CSV ファイル名*）.tdi が，OS のファイルパス長の上限値を超えないようにしてください。
KEDL0056T-E	設定内容が長すぎます。(テストケース CSV : *** 1 *** 行 : *** 2 ***)	1 行のバイト数，またはデータ項目名欄の文字数が上限値を超えました。 該当の行をスキップします。 1 行のバイト数，またはデータ項目名欄の文字数を上限値以下にしてください。
KEDL0057T-E	テストデータを設定するための情報が足りません。(テストケース CSV : *** 1 *** 行 : *** 2 ***)	テストデータを設定するために必要な情報が不足しています。 該当の行をスキップします。 設定内容を見直し，必要な情報を追加してください。
KEDL0058T-E	項目の内容が不正です。(テストケース CSV : *** 1 *** 行 : *** 2 *** 内容 : *** 3 ***)	項目の内容が間違っています。 該当の行をスキップします。 設定内容を見直し，正しい内容に変更してください。
KEDL0059T-E	フォーマットが不正です。(テストケース CSV : *** 1 *** 行 : *** 2 ***)	ダブルコーテーションのペアが正しくないなど，CSV のフォーマットが間違っています。 該当の行をスキップします。 設定内容を見直し，正しい内容に変更してください。
KEDL0060T-E	ファイル名，オープンモード，入出力文または終了条件の組み合わせが不正です。(テストケース CSV : *** 1 *** 行 : *** 2 ***)	ファイル名，オープンモード，入出力文または終了条件の組み合わせが間違っています。 該当の行をスキップします。 設定内容を見直し，ファイル名，オープンモード，入出力文または終了条件の組み合わせを修正してください。
KEDL0061T-E	ひな形 CSV ファイルを出力できません。	ひな形 CSV ファイルを出力できません。 処理を中止します。

メッセージ ID	メッセージテキスト	対処
		対象のフォルダのパスが正しいか，またはそのフォルダのアクセス権限が付与されているかを確認してください。
KEDL0062T-E	プログラム情報ファイルを正しく生成できませんでした。コンパイラオプションを見直してください。	プログラム情報ファイルが正しく生成できませんでした。 単体テスト支援を終了します。 -StdCallFile コンパイラオプションの後ろにファイル名を指定していないかなど，コンパイラオプションの指定方法に誤りがないかどうかを見直してください。
KEDL0063T-E	スタブ設定ページの同名の EXTERNAL 指定のファイルが複数チェックオンになっているため，処理が行えません。同名の EXTERNAL 指定のファイルはどれかひとつのチェックをオンにし，他はチェックをオフにしてください。	［スタブ設定ページ］で同名の EXTERNAL ファイルが複数チェックオンになっています。 処理を中止します。 どれか 1 つのファイルのチェックをオンにし，他のファイルのチェックをオフにしてください。
KEDL0064T-E	テストデータ設定後に，COBOL ソースファイルが更新されました。テストデータの設定を見直してから再実行してください。	テストデータを設定した後に，COBOL ソースファイルを更新したため，テスト実行をするときに対応の取れないテストデータが見つかりました。 処理を中止します。 対応の取れないテストデータが不要な場合など，更新後の内容で問題がない場合は，テスト実行を行います。 問題がある場合は，次のどちらかを行ってください。 • テストデータの見直し，再設定する。 • テストプロジェクトを保存せずに開き直したり，COBOL ソースファイルを元に戻したりして，元の状態に戻す。
KEDL0065T-I	一番古い実行結果を削除しました。(***1***)	一番古い実行結果を削除しました。
KEDL0066T-U	COBOL2002 Net Developer の TEMPLATE フォルダまたはサービスルーチンファイルが存在しないか，アクセス権限がありません。	COBOL2002 Net Developer の TEMPLATE フォルダまたはサービスルーチンファイルが存在しないか，アクセス権限がありません。 単体テスト支援を終了します。 COBOL2002 Net Developer をアンインストールしてから，再度インストールしてください。
KEDL0067T-E	指定したファイルが存在しません。	存在しないファイルを指定しました。 処理を中止します。 存在するファイルを指定し直してください。
KEDL0068T-E	指定されたテストプロジェクトファイルにアクセスできません。アクセス権限が付与されているかを確認してください。(*** 1 ***)	テストプロジェクトファイルにアクセスできません。 処理を中止します。 テストプロジェクトファイルにアクセス権限が付与されているかを確認してください。
KEDL0069T-E	テストケース TD コマンド格納ファイルにアクセスできません。テストケース TD コマンド格納	テストケース TD コマンド格納ファイルにアクセスできません。

メッセージ ID	メッセージテキスト	対処
	ファイルを別プログラムで開いていないか，もしくはテストケース TD コマンド格納ファイルにアクセス権限が付与されているかを確認してください。(*** 1 ***)	処理を中止します。 テストケース TD コマンド格納ファイルを別プログラムで開いていないか，またはアクセス権限が付与されているかを確認してください。

(2) 単体テスト支援共通のメッセージ（KEDL1000～KEDL1999）と対処

表 13–2 メッセージ画面のメッセージ（KEDL1000～KEDL1999）と対処

メッセージ ID	メッセージテキスト	対処
KEDL1000T-W	テストプロジェクトを新規に作成しました。(*** 1 ***)	新規にテストプロジェクトを生成しました。
KEDL1001T-W	解析後，対応の取れないテストデータがありました。テストプロジェクトから削除します。	解析前後で対応が取れないテストデータが存在しました。 単体テストで必要なテストデータについては，再設定してください。
KEDL1002T-I	テストケース CSV をインポートしました。(テストケース CSV：*** 1 *** テストケース名：*** 2 ***)	テストケース CSV をインポートしました。
KEDL1003T-I	テストケース CSV にエクスポートしました。(テストケース CSV：*** 1 *** テストケース名：*** 2 ***)	テストケース CSV をエクスポートしました。
KEDL1004T-W	ファイルシミュレーションに設定済みのテストデータがシミュレーションの解除により無効化されました。このテストデータはテストプロジェクト保存時に削除されます。(*** 1 ***)	テストデータを設定しているファイルを［スタブ設定ページ］でオフにしています。 誤ってシミュレーションを解除した場合は，テストプロジェクトを保存する前に［スタブ設定ページ］でオンにしてください。
KEDL1005T-W	中断点に設定済みのテストデータが中断点の解除により無効化されました。このテストデータはテストプロジェクト保存時に削除されます。	テストデータを設定済みの中断点を解除しました。 誤って中断点を解除した場合は，テストプロジェクトを保存する前に再度同じところに中断点を設定してください。
KEDL1006T-I	単体テストの実行を中止しました。	単体テストの実行を中止しました。
KEDL1007T-I	テストプロジェクトを開きました。(*** 1 ***)	テストプロジェクトを開きました。
KEDL1008T-I	テストプロジェクトを上書き保存しました。(*** 1 ***)	開いているテストプロジェクトを上書き保存しました。
KEDL1009T-I	テストプロジェクトを閉じました。(*** 1 ***)	開いているテストプロジェクトを閉じました。
KEDL1010T-I	テスト対象の COBOL ソースファイルの解析を行いました。(*** 1 ***)	COBOL ソースファイルの解析を実行しました。
KEDL1011T-I	単体テストを実行しました。	単体テストを実行しました。

メッセージ ID	メッセージテキスト	対処
KEDL1012T-I	単体テスト用の実行可能ファイルを生成しました。	単体テスト用の実行可能ファイルを生成しました（テストケース実行時）。
KEDL1013T-I	単体テスト用の TD コマンド格納ファイルを生成しました。	単体テスト用の TD コマンド格納ファイルを生成しました。
KEDL1014T-I	実行結果レポートを生成しました。(*** 1 ***)	実行結果レポートを生成しました（テストケース実行完了時）。
KEDL1015T-I	カバレージ情報をクリアしました。	カバレージ情報をクリアしました。
KEDL1016T-I	テストケースを追加しました。(*** 1 ***)	テストケースを追加しました。
KEDL1017T-I	テストケースを削除しました。(*** 1 ***)	テストケースを削除しました。
KEDL1018T-I	実行結果を削除しました。エビデンスファイルはテストプロジェクト保存時に削除されます。(*** 1 ***)	実行結果を削除しました。
KEDL1019T-I	実行結果をテストプロジェクトから除外しました。エビデンスファイルは削除されません。(*** 1 ***)	実行結果をテストプロジェクトから除外しました。ファイルの実体は削除しません。
KEDL1020T-I	中断点を追加しました。(*** 1 ***)	中断点を追加しました。
KEDL1021T-I	中断点を削除しました。(*** 1 ***)	中断点を削除しました。
KEDL1022T-I	中断点に対応する文番号が存在しません。該当の中断点を削除します。(*** 1 ***)	設定済みの中断点に対応する文番号が存在しないため，該当する中断点を削除しました。
KEDL1023T-W	テスト対象の COBOL ソースファイルが指定されたパスにありません。プロパティページから COBOL ソースファイルの設定で正しいパスを設定してください。(*** 1 ***)	テストプロジェクトを開いたときに，テスト対象の COBOL ソースファイルのパスのところにファイルが存在しません。 テスト対象の COBOL ソースファイルのパスを正しく設定してください。
KEDL1024T-W	実行環境ファイルが指定されたパスにありません。プロパティページから実行環境ファイルの正しいパスを設定してください。(*** 1 ***)	テストプロジェクトを開いたときに，実行環境ファイルのパスのところにファイルが存在しません。 実行環境ファイルのパスを正しく設定してください。
KEDL1026T-I	テストケース CSV のエクスポートをスキップしました。(*** 1 ***)	名前の重複があるためスキップしました。 テストケースか，既存のテストケース CSV のファイル名を変更してください。または，違う場所にエクスポートしてください。
KEDL1027T-I	テストケース CSV のインポートをスキップしました。(*** 1 ***)	名前の重複があるためスキップしました。 テストケースか，既存のテストケース CSV のファイル名を変更してインポートしてください。
KEDL1028T-I	テストケース CSV のエクスポートが完了しました。	テストケース CSV のエクスポートが完了しました。
KEDL1029T-I	テストケース CSV のインポートが完了しました。	テストケース CSV のインポートが完了しました。

メッセージ ID	メッセージテキスト	対処
KEDL1030T-I	テストケース CSV のエクスポートをキャンセルしました。	テストケース CSV のエクスポートをキャンセルしました。
KEDL1031T-I	テストケース CSV のインポートをキャンセルしました。	テストケース CSV のインポートをキャンセルしました。
KEDL1032T-I	テストケース CSV のひな形をエクスポートしました。(*** 1 ***)	テストケース CSV のひな形をエクスポートしました。
KEDL1033T-W	スタブ設定ページで同名の EXTERNAL 指定のファイルが複数チェックオンになっています。同名の EXTERNAL 指定のファイルはどれかひとつのチェックをオンにし，他はチェックをオフにしてください。	［スタブ設定ページ］で同名の EXTERNAL 指定のファイルが複数チェックオンになっています。 どれか 1 つのファイルのチェックをオンにし，他のファイルのチェックをオフにしてください。
KEDL1034T-W	上書き前のテストケースに実行環境ファイルまたは実行時環境変数の設定がありました。これらの設定は，上書き後のテストケースでも有効です。(テストケース名：*** 1 ***)	上書き対象のテストケースに実行環境ファイル，実行時環境変数の設定があります。 テストケースごとの実行環境ファイル，実行時環境変数の設定を変更する場合は，［テスト環境の設定］画面で設定してください。
KEDL1035T-W	テストケースに設定済みの実行環境ファイルまたは実行時環境変数が無効化されました。テストケースに設定した実行環境ファイルまたは実行時環境変数はテストプロジェクト保存時に削除されます。(*** 1 ***)	テストケースに実行環境ファイルまたは実行時環境変数を設定しているときに，［プロパティページ］で「テストケースごとに実行環境ファイルまたは実行時環境変数を設定する」チェックボックスをオフにしました。 誤って「テストケースごとに実行環境ファイルまたは実行時環境変数を設定する」チェックボックスをオフにした場合は，テストプロジェクトを保存する前にオンにしてください。

(3) GUI 固有のメッセージ（KEDL2000～KEDL2999）と対処

表 13–3 GUI 固有のメッセージ（KEDL2000～KEDL2999）と対処

メッセージ ID	メッセージテキスト	対処
KEDL2000T-U	メモリ不足が発生しました。	メモリ不足が発生しました。 単体テスト支援を終了します。 次に示す対処で，メモリ使用量を減らしてください。 • テストケースの数を減らす。 • 設定しているテストデータの数を減らす。 • 実行結果の数を減らす。 • テスト対象のプログラムで繰り返し実行される文に中断点を設定している場合は，繰り返し回数を見直し，実行する回数を減らす。
KEDL2001T-U	COBOL2002 Professional Tool Kit の環境が不正です。	COBOL2002 Professional Tool Kit のレジストリまたは製品のファイルが存在しません。環境不正，またはインストールが失敗した可能性があります。

メッセージ ID	メッセージテキスト	対処
		単体テスト支援を終了します。 COBOL2002 Professional Tool Kit をアンインストールしてから，再インストールしてください。
KEDL2010T-U	A logical error has occurred. Parameter(*** 1 ***)	単体テスト支援を起動するコンポーネントとのインタフェースに不正があります。 単体テスト支援を終了します。 当社保守員に連絡してください。
KEDL2011T-U	A logical error has occurred. Method (*** 1 ***)	単体テスト支援のコンポーネント間のインタフェースに不正があります。 単体テスト支援を終了します。 当社保守員に連絡してください。
KEDL2012T-U	A logical error has occurred.	論理エラーが発生しました。 単体テスト支援を終了します。 当社保守員に連絡してください。
KEDL2013T-U	A logical error has occurred. Exception (*** 1 ***)	論理エラーが発生しました。 単体テスト支援を終了します。 当社保守員に連絡してください。
KEDL2014T-W	コンパイルエラーが発生しているため，テストケースなどの操作はできません。エラーを解消してください。	コンパイルエラーが発生しています。 プログラムの内容や設定を見直して，コンパイルエラーを解消してください。
KEDL2015T-W	プログラム情報ファイルに必要な情報が含まれていないため，テストケースなどの操作はできません。コンパイラオプションを見直してください。	プログラム情報ファイルが正しく作成できていません。 指定したコンパイラオプションを見直してください。 -StdCallFile コンパイラオプションの後ろにファイル名を指定していないかなど，コンパイラオプションの指定方法に誤りがないかどうかを見直してください。
KEDL2016T-W	対応の取れないテストデータがあります。対応の取れないテストデータは，プロジェクトの保存，またはテスト実行したときに削除されます。	テストデータを設定した後に，COBOL ソースファイルを更新したため，テスト実行をするときに対応の取れないテストデータが見つかりました。 対応の取れないテストデータが不要な場合など，更新後の内容で問題がない場合は，テスト実行を行います。 問題がある場合は，次のどちらかを行ってください。 • テストデータの見直し，再設定する。 • テストプロジェクトを保存せずに開き直したり，COBOL ソースファイルを元に戻したりして，元の状態に戻す。

(4) バッチモード固有のメッセージ（KEDL3000～KEDL3999）と対処

表 13–4 バッチモード固有のメッセージ（KEDL3000～KEDL3999）と対処

メッセージ ID	メッセージテキスト	対処
KEDL3000T-U	メモリ不足が発生しました。	メモリ不足が発生しました。 単体テスト支援を終了します。 次に示す対処で，メモリ使用量を減らしてください。 • テストケースの数を減らす。 • 設定しているテストデータの数を減らす。 • 実行結果の数を減らす。 • テスト対象のプログラムで繰り返し実行される文に中断点を設定している場合は，繰り返し回数を見直し，実行する回数を減らす。
KEDL3001T-U	論理エラーが発生しました。エラー詳細:	論理エラーが発生しました。 単体テスト支援を終了します。 当社保守員に連絡してください。
KEDL3002T-E	コマンドライン引数に指定したオプションが不正です。	不正な値の引数が指定されました。 処理を中止します。 正しいオプションを指定してください。
KEDL3003T-E	テストプロジェクトファイルが存在しません。指定したパスが正しいか，またはアクセス権限が付与されているかを確認してください。(*** 1 ***)	テストプロジェクトファイルとして指定されたファイルパスのところにファイルが存在しないか，そのファイルにアクセス権限が付与されていません。 処理を中止します。 ファイルが存在しない場合には，正しいファイルパスを指定してください。 ファイルにアクセス権限が付与されていない場合は，アクセス権限を付与してください。
KEDL3004T-E	ターゲットが複数指定されています。ターゲットは 1 つだけ指定してください。	ターゲットが複数指定されました。 処理を中止します。 ターゲットを 1 つにしてください。
KEDL3005T-E	ターゲットが指定されていません。ターゲットを指定してください。	ターゲットが指定されていません。 処理を中止します。 ターゲットを指定してください。
KEDL3006T-E	アクションが複数指定されています。アクションは 1 つだけ指定してください。	アクションが複数指定されました。 処理を中止します。 アクションを 1 つにしてください。
KEDL3007T-E	アクションが指定されていません。アクションを指定してください。	アクションが指定されていません。 処理を中止します。 アクションを指定してください。
KEDL3008T-E	***1***オプションに必要な引数が指定されていません。引数を指定してください。	引数が必要なオプションに，引数が指定されていません。

メッセージ ID	メッセージテキスト	対処
		処理を中止します。 引数を指定してください。
KEDL3009T-E	***1***オプションに引数は指定できません。	引数を指定できないオプションに引数が指定されました。 処理を中止します。 引数を指定しないでください。
KEDL3010T-E	コマンドラインが不正です。(***1***)	不正な文字列がコマンドラインに指定されました。 処理を中止します。 正しいオプションを指定してください。
KEDL3011T-E	***1***オプションと***2***オプションは同時に指定できません。	同時に指定できないオプションが同時に指定されました。 処理を中止します。 どちらかのオプションだけにしてください。
KEDL3012T-E	***1***オプションと***2***オプション指定した場合，***3***オプションは指定できません。	指定できないオプションが指定されました。 処理を中止します。 指定できるオプションを指定してください。
KEDL3013T-E	***1***を指定してください。	指定が必要なものが指定されていません。 処理を中止します。 必要なものを指定してください。
KEDL3014T-E	ファイルは指定できません。フォルダを指定してください。	フォルダを指定するところにファイルを指定しました。 処理を中止します。 フォルダを指定してください。
KEDL3015T-E	フォルダは指定できません。ファイルを指定してください。	ファイルを指定するところにフォルダを指定しました。 処理を中止します。 ファイルを指定してください。
KEDL3016T-E	ファイルの拡張子が違います。(*** 1 ***)	指定できません。 処理を中止します。 正しい拡張子を持つファイルを指定してください。
KEDL3017T-E	指定したフォルダにアクセスできません。 (***1***)	指定したフォルダにアクセスできませんでした。 処理を中止します。 フォルダのパスが正しいか，またはフォルダのアクセス権限が付与されているかを確認してください。
KEDL3018T-E	コンパイルエラーが発生しているため，コマンドを実行できません。エラーを解消してください。	コンパイルエラーが発生しています。 処理を中止します。 プログラムの内容や設定を見直して，コンパイルエラーを解消してください。
KEDL3019T-E	プログラム情報ファイルに必要な情報が含まれていないため，コマンドを実行できません。コンパイラオプションを見直してください。	プログラム情報ファイルが正しく作成できていません。 処理を中止します。

メッセージ ID	メッセージテキスト	対処
		-StdCallFile コンパイラオプションの後にファイル名を指定していないなど，コンパイラオプションの指定方法に誤りがないか見直してください。
KEDL3020T-W	テストケースがありません。	テストケースのエクスポート，削除または一覧を表示しようとしましたが，テストプロジェクトにテストケースがありませんでした。 処理を中止します。 テストケースがあるテストプロジェクトを対象にしてください。
KEDL3021T-E	指定したテストケースが見つかりません。(***1***)	指定したテストケースが見つかりませんでした。 処理を中止します。 存在するテストケースを指定してください。
KEDL3022T-I	すべてのテストケースを削除します。	すべてのテストケースを削除します。
KEDL3023T-I	実行結果を削除します。(***1***)	実行結果を削除します。
KEDL3024T-I	プロジェクト情報ファイルを出力しました。(***1***)	プロジェクト情報ファイルを出力しました。
KEDL3025T-I	プロジェクト情報ファイルの内容を設定しました。	プロジェクト情報ファイルの内容を設定しました。
KEDL3026T-E	プロジェクト情報ファイルの内容が不正です。後続するエラーメッセージを参考にして，エラーを取り除いてください。	プロジェクト情報ファイルの内容が不正です。 処理を中止します。 プロジェクト情報ファイルの内容を修正して，再設定してください。
KEDL3027T-E	要素は空にできません。(***1***)	プロジェクト情報ファイルの内容で，空を指定できない要素を空にしました。 処理を中止します。 空要素を削除してください。または空要素でなくしてください。
KEDL3028T-W	***1***の内容は無視します。	プロジェクト情報ファイルでは変更できない項目が，プロジェクト情報ファイルで変更されていました。
KEDL3029T-W	テストデータ設定後に，COBOL ソースファイルが更新されました。続行すると，対応の取れないテストデータは削除されます。削除しても良い場合は続行してください。そうでない場合は処理を中断し，COBOL ソースファイルの内容やテストデータの見直し，再設定をしてください。	テストデータを設定した後に，COBOL ソースファイルを更新したため，テスト実行をするときに対応の取れないテストデータが見つかりました。 一時中断し，ユーザに確認します。 対応の取れないテストデータが不要な場合など，更新後の内容で問題がない場合は，「Y」を入力して，処理を続行してください。 問題がある場合は，「N」を入力して処理を中断したあと，COBOL ソースファイルの内容や，テストデータの見直し，再設定してください。
KEDL3030T-I	続行しますか？ (Y/N)	直前の動作を続行するかどうか。 「Y」を入力：処理を続行します 「N」を入力：処理を中止します

メッセージ ID	メッセージテキスト	対処
		続行するか否かに合わせて，Y または N を入力してください。
KEDL3032T-I	削除しますか？ (Y/N)	直前に表示された項目を削除するかどうか。 「Y」を入力：削除します 「N」を入力：削除しません 削除するか否かに合わせて，Y または N を入力してください。
KEDL3033T-I	実行結果を削除しました。(*** 1 ***)	実行結果を削除しました。
KEDL3034T-W	実行結果がありません。	実行結果を削除，または一覧を表示しようとしましたが，テストプロジェクトに実行結果がありませんでした。 処理を中止します。 実行結果があるテストプロジェクトを対象にしてください。
KEDL3035T-E	指定した実行結果が見つかりません。(***1***)	指定した実行結果が見つかりませんでした。 処理を中止します。 存在する実行結果を指定してください。
KEDL3036T-I	すべての実行結果を削除します。	すべての実行結果を削除します。
KEDL3037T-E	CSV ファイルが存在しません。CSV ファイルがあるフォルダを指定するか，または，サブフォルダに CSV ファイルがあるフォルダと-Recursive オプションを指定してください。	指定したフォルダに CSV ファイルが存在しません。 処理を中止します。 CSV ファイルがあるフォルダを指定してください。 または，サブフォルダに CSV ファイルがあるのなら，-Recursive オプションを指定してください。
KEDL3038T-I	カバレージ情報をクリアしますか？ (Y/N)	カバレージ情報をクリアするかどうか。 「Y」を入力：クリアします 「N」を入力：クリアしません
KEDL3039T-E	有効なテストケースがありません。実行したいテストケースを有効にするか，または新たにテストケースを登録してからテスト実行してください。	テスト実行しようとしましたが，有効なテストケースがありませんでした。 処理を中止します。 実行したいテストケースを有効にするか，または新たにテストケースを登録してからテスト実行してください。
KEDL3040T-E	-InputPath オプションにファイルを指定した場合，-Recursive オプションは指定できません。-Recursive オプションの指定をやめるか，または，-InputPath オプションにフォルダパスを指定してください。	テストケースのインポートで，インポート元に CSV ファイルを指定しましたが，同時に-Recursive オプションも指定しました。 処理を中止します。 -Recursive オプションの指定をやめるか，または，-InputPath オプションにフォルダパスを指定するようにしてください。
KEDL3041T-E	指定されたプロジェクト情報ファイルにアクセスできません。指定したパスが正しいか，またはア	プロジェクト情報ファイルにアクセスできません。 処理を中止します。

メッセージ ID	メッセージテキスト	対処
	クセス権限が付与されているかを確認してください。(*** 1 ***)	ファイルが存在するか，もしくはファイルにアクセス権限が付与されているかどうかを確認してください。
KEDL3042T-I	テストケースを削除します。(***1***)	テストケースを削除します。
KEDL3044T-W	ConstantName 要素で指定した副プログラムを呼び出す文が見つからないか，または設定対象ではありません。無視します。(***1***)	次のどれかに該当するプログラム名を ConstantName 要素で指定しました。 • CALL 定数で呼び出していない。 • テスト対象の COBOL ソースファイル内に定義されている。 • INVOKE 文での呼び出しやサービスルーチンなど，ダミーの副プログラムの作成対象ではない。 無視します。 指定した内容が正しいか確認してください。誤っていた場合は，プロジェクト情報ファイルを修正して，テストプロジェクトに再設定してください。
KEDL3045T-W	FileName 要素で指定したファイル名の入出力文が見つかりませんでした。無視します。(***1***)	プロジェクト情報ファイルで，FileName 要素を使ってファイル名を指定しましたが，そのファイル名の入出力が見つかりませんでした。 無視します。 指定した内容が正しいか確認してください。誤っていた場合は，プロジェクト情報ファイルを修正して，テストプロジェクトに再設定してください。
KEDL3046T-W	同名の EXTERNAL 指定のファイルのうち，複数のファイルに対してスタブを作成する設定になっています。同名の EXTERNAL 指定のファイルはどれかひとつをスタブ作成の対象にしてください。	プロジェクト情報ファイルで，同名の EXTERNAL 指定のファイルのうち，複数のファイルがスタブ作成の対象になっています。 指定した内容で処理を続行します。 同名の EXTERNAL 指定のファイルの設定を見直して，1 つだけがスタブ作成の対象になるようにプロジェクト情報ファイルの内容を変更してください。 変更したあと，プロジェクト情報ファイルをテストプロジェクトに再設定してください。
KEDL3047T-W	一意名指定の CALL 文がありません。IdentName 要素の指定を無視します。(***1***)	プロジェクト情報ファイルで，IdentName 要素を使って副プログラムを指定しましたが，一意名指定の CALL 文がありませんでした。 無視します。 指定した内容が正しいか確認してください。誤っていた場合は，プロジェクト情報ファイルを修正して，テストプロジェクトに再設定してください。
KEDL3048T-W	指定されたプロジェクト情報ファイルはすでに存在します。存在しないファイルを指定するか，または-OverWrite オプションを指定してください。(***1***)	プロジェクト情報ファイルの出力で，すでに存在するファイルを指定しました。 処理を中止します。 別のプロジェクト情報ファイルを指定するか，または-OverWrite オプションを指定して再度プロジェクト情報ファイルを出力してください。

メッセージ ID	メッセージテキスト	対処
KEDL3049T-E	***1***要素の name 属性の値を空にした場合は，value 属性に値は指定できません。name 属性に値を設定するか，または value 属性を空にしてください。	name 属性の値に空を指定しているのに，value 属性に値を指定しました。 処理を中止します。 プロジェクト情報ファイルの内容を見直し，name 属性に値を設定するか，または value 属性を空にしてください。
KEDL3050T-E	作成対象のテストプロジェクトファイルのパスが不正です。ファイル名が拡張子だけや空白と拡張子だけになっていないか，または OS のパス長の上限値を超えていたりパスに使用できない文字を使っていないか確認してください。(***1***)	テストプロジェクトとして指定できないパスを，作成するテストプロジェクトファイルのパスに指定しました。 処理を中止します。 次の条件に該当するパスを指定していないか確認し，問題を取り除いてから再実行してください。 • ファイル名が拡張子だけから成る • ファイル名が空白と拡張子だけから成る • ファイルパス長が OS の上限値を超えている • ファイル名やパスに使用できない文字を含んでいる
KEDL3051T-E	encoding 属性には Shift_JIS を指定してください。	XML 宣言の encoding 属性に指定した値が不正です。 処理を中止します。 encoding 属性に Shift_JIS を指定してください。
KEDL3052T-E	プロジェクト情報ファイルの RuntimeEnvironmentFile 要素に指定した実行環境ファイルのパスが不正です。指定したパスに誤りがないか確認してください。(***1***)	プロジェクト情報ファイルの RuntimeEnvironmentFile 要素に，不正なパスを指定しました。 処理を中止します。プロジェクト情報ファイルの RuntimeEnvironmentFile 要素の内容を見直してください。
KEDL3053T-W	testcasename 属性に指定したテストケースが存在しません。TestCaseSetting 要素の内容を無視します。(***1***)	プロジェクト情報ファイルで，testcasename 属性に指定したテストケースがありませんでした。 TestCaseSetting の内容を無視します。 指定したテストケース名が正しいか確認してください。誤っていた場合は，testcasename 属性に正しい値を指定し，プロジェクト情報を再設定してください。
KEDL3054T-W	テストプロジェクトの作成で EnableTestCaseSetting 要素に ON が指定されました。EnableTestCaseSetting 要素を OFF に仮定し，TestCaseSettings 要素の内容は無視します。テストケースごとに実行環境ファイル，実行時環境変数を設定する場合は，設定したいテストケースを作成した後にプロジェクト情報を設定し直してください。	テストプロジェクトの作成時に，プロジェクト情報ファイルの EnableTestCaseSetting 要素に ON を指定しました。 処理を続行します。 テストケースごとに実行環境ファイルまたは実行時環境変数を設定する場合は，設定したいテストケースを作成したあとにプロジェクト情報を再設定してください。
KEDL3055T-E	TestCaseSettings 要素が指定されていません。TestCaseSettings 要素を指定してください。	EnableTestCaseSetting 要素に ON が指定されていますが，TestCaseSettings 要素が指定されていません。

メッセージ ID	メッセージテキスト	対処
		処理を中止します。 プロジェクト情報ファイルに TestCaseSettings 要素を指定し，プロジェクト情報を設定し直してください。

13.2　エラー時に採取するファイル

メッセージの内容を確認してもトラブルが解決しない場合，保守員に連絡してください。保守員に連絡するときには，次のフォルダおよびファイルを採取して，現象の詳細と発生日時をあわせてご連絡ください。

- テストプロジェクトファイルがあるフォルダ下のファイル一式
- ［プロパティページ］の［COBOL ソースファイル］で指定したソースファイル
- エラー情報ファイル

13.2.1　エラー情報ファイルの格納先

エラー情報ファイルは，次に示すフォルダに生成されます。

```
ドキュメントのフォルダ※ ¥Cbluts_yyyymmdd_hhmmss.txt
```

注※　Windows の「ドキュメント」または「マイ ドキュメント」のフォルダです。フォルダ名は Windows の種類によって異なります。

出力される各項目の意味を次に示します。

- *yyyymmdd*（*yyyy*：西暦年，*mm*：月，*dd*：日）はファイルが出力された日付です。
- *hhmmss*（*hh*：時，*mm*：分，*ss*：秒）はファイルが出力された時間です。

出力例

```
C:¥Documents and Settings¥MyUsername¥My Documents¥Cbluts_20150410_141116.txt
```

2015 年 4 月 10 日　14 時 11 分 16 秒に出力したエラー情報ファイルです。

不要になったエラー情報ファイルは削除できます。

付録

付録 A　単体テスト支援で使用するファイル

単体テスト支援で使用するファイルの説明です。

付録 A.1　単体テスト支援のファイル一覧

単体テスト支援で使用するファイルの一覧です。

ファイル名	拡張子	生成元	説明
テストプロジェクトファイル	.utsproj	○	単体テストのためのテストデータおよびテスト結果の情報が格納されているファイルです。
TD コマンド格納ファイル	.tdi	○	テストデバッガで単体テストを実行するための TD コマンドを記述したファイルです。
COBOL ソースファイル	※1	○	単体テスト対象の COBOL ソースファイルです。
実行可能ファイル	.exe	○	テスト対象ソースコードをコンパイルし，テストデバッガでテストが実行できる実行可能ファイルです。 このファイル自体は COBOL コンパイラが生成します。
プログラム情報ファイル	.cbp	○	テスト対象ソースコードのデータ項目の情報やカバレージの情報などを保持するファイルです。 このファイル自体は COBOL コンパイラが生成し，テストデバッガがカバレージ情報を設定します。 テストプロジェクトの初回生成時に，このプログラム情報ファイルを生成するためにコンパイルを実行します。
擬似プログラム用プログラム情報ファイル	.cbs	○	テストデバッガでテスト実行時に，主プログラムや副プログラムをシミュレーションするための情報を保持するファイルです。このファイル自体は COBOL コンパイラが生成します。
コンパイラが生成するファイル	※2	○	COBOL コンパイラがコンパイル時に生成するファイル群です。単体テスト支援では，TestModule フォルダに出力します。 単体テスト支援ではこれらのファイルを直接利用することはありません。
実行環境ファイル	.cbr	－	単体テストの実行時の環境変数の情報を設定したファイル。COBOL2002 の実行支援で作成します。 単体テスト支援では，COBOL2002 の実行支援で作成した実行環境ファイルをプロジェクトのプロパティに設定すると，テスト実行時に設定した実行環境ファイルが実行可能ファイルと同じフォルダにコピーされます。テストデバッガでのテスト実行では，自動的に実行環境ファイルが読み込まれます。

（凡例）

○：単体テスト支援によって生成されるファイルです。

－：ほかの COBOL2002 のコンポーネントが生成し，単体テスト支援が参照するファイルです。

注※1

COBOL ソースファイルの拡張子を次に示します。

固定形式正書法：

.cbl，.cbo，.ocb，または環境変数 CBLFIX で指定した拡張子

自由形式正書法：

.cbf，.ocf，または環境変数 CBLFREE で指定した拡張子

注※2

オブジェクトファイル（.obj）などがあります。コンパイラが生成するファイルについては，マニュアル「COBOL2002 ユーザーズガイド」を参照してください。

付録 A.2　テストケースの CSV ファイル

テストケースのインポート/エクスポートで使用する CSV ファイルの形式です。

(1)　テストケースの CSV ファイルの規則

テストケースの CSV ファイルの規則は次のとおりです。

- 行頭に「*」がある行はコメント行として扱います。
- 文字コードはシフト JIS です。シフト JIS 以外の文字コードで保存した場合の動作は保証しません。
- 1 行にはデータ項目 1 つの情報を格納します。
- 1 行の各項は半角のコンマ「,」で区切ります。1 つの項は設定する値をそのまま記載するか，半角のダブルコーテーション「"」で囲みます。ただし，設定する値に次の文字が含まれる場合は，その項全体を半角のダブルコーテーション「"」で囲む必要があります。
  - 半角のコンマ「,」
  - 半角のダブルコーテーション「"」
- 半角のダブルコーテーション「"」で囲んだ中に，さらに半角のダブルコーテーション「"」を入れる場合は，「""」のように 2 つ並べて記述します。

  例を次に示します。

項番	扱う文字	CSV ファイルの記載方法	備考
1	@'ALLOCATE,120'	"@'ALLOCATE,120'"	－
2	@"ALLOCATE"	"@""ALLOCATE"""	－
3	"AAA"	"""AAA"""	－
4	"AA,A"	"""AA,A"""	－
5	"AA""A"	"""AA""""A"""	代入値または期待値に指定すると，「AA"A」（文字列データ）として代入または比較されます。
6	"""B""C"	"""""""B""""C"""	代入値または期待値に指定すると，「"BB"C」（文字列データ）として代入または比較されます。

- 単体テスト支援が出力するCSVファイルでは，必要な場合だけ半角のダブルコーテーション「"」で囲んで出力されます。
- 項目の値として改行は含めないでください。改行を含んでいた場合の動作は保証しません。
- 1行に含まれる項目数は，テストデータの設定に必要な項目数を超えてもエラーにはなりません。超えた分の内容は無視されます。ただし，1行に書ける最大バイト数を超えるとエラーになります。
- 項目によっては空欄にできますが，空欄ではなくコンマで区切られていないことで項目数が足りない状態にはできません。この状態のCSVファイルはエラーになります。
- 項目によっては，定数を使用している場合があります。定数は半角の英数字で，英大文字と英小文字の区別はしません。ただし，単体テスト支援がエクスポートする場合は，各項目の説明の表記で出力されます。項目で共通に使用する定数を次に示します。

定数	意味
true	該当の項目を有効にします。GUI画面上ではチェックボックスがオンになります。
false	該当の項目を無効にします。GUI画面上ではチェックボックスがオフになります。

(2)　テストケースのCSVファイルの形式

単体テスト支援でエクスポートする，テストケースのCSVファイルの1行のフォーマットは次のとおりです。

<データ項目の設定個所の情報>,<参照フラグ>,<データ項目のレベル番号>,<データ項目名>,<値の表示>,<期待値>,<代入値>,<設定個所固有情報>

各項目の意味とエクスポート時のデフォルト値について説明します。デフォルト値は，CSVファイルにエクスポートするときに，項目に対応する値が設定されていなかった場合に出力される値のことです。また，インポート時に項目が空欄の場合の扱いは，「空欄の扱い」として説明します。空欄の扱いでエラーになった場合は，エラーメッセージが出力され，その行はスキップされます。

表A-1　テストケースのCSVファイルの形式

項番	項目	内容
1	データ項目の設定個所の情報	どのテストデータの設定個所かを示す定数です。定数と設定個所を次に示します。 StartPoint ［プログラムの開始時］画面用の設定情報です。 EndPoint ［プログラムの終了時］画面用の設定情報です。 FileSim ［ファイルシミュレーション］画面用の設定情報です。 BreakPoint ［中断点］画面用の設定情報です。 ［デフォルト値］ 対応する設定個所を表す定数が仮定されます。

項番	項目	内容
		[空欄および定数以外の値の扱い] エラーになります。
2	参照フラグ	プログラム中で参照されているデータ項目かどうかを示します。このデータ項目は，CSV 編集のための情報であり，インポート時にはテストプロジェクトに反映させることはなく，エクスポート時の値が編集されていてもエラーになりません。 • 参照されている場合：R • 参照されていない場合：空欄 [空欄の扱い] 無視されます。
3	データ項目のレベル番号	データ項目のレベル番号を表す半角の数字です。 エクスポート時は，01 と出力されます。また，特殊レジスタは，エクスポート時は，空欄で出力されます。 データ項目名は，OF 修飾子を使った完全な名前で指定するため，インポート時にこの項目をテストプロジェクトに反映させることはなく，エクスポート時の値が編集されていたとしてもエラーになりません。 [デフォルト値] データ項目のレベル番号文字列です。 [空欄の扱い] 無視されます。
4	データ項目名	テストデータの設定対象のデータ項目名です。デバッガのプログラム名指定（#PROGRAM など）は扱えません。指定するとエラーになります。 エクスポート時は，すべてのレベル階層を OF 修飾子でつなげた名前で出力されます。ただし，FILLER は OF 修飾子を付けないでそのまま出力されます。 ファイル記述項で定義されたデータ項目の場合は，ファイル名も OF 修飾子でつなげた名前で出力されます。また，データ項目が多重定義されていても出力されます。 [大文字小文字，全角半角の扱い] エクスポートのときは，等価規則に従った各設定画面に表示される名前で出力されます。 インポートでは，この項目の内容を使って対象のデータを検索します。そのときの検索では，入力したデータ項目名をそのまま使用します。そのため，データ名は，各設定画面に表示される名前として等価規則を意識して指定してください。 OF 修飾子，空白，括弧( )，コンマは，等価規則に関係なく半角文字だけを使用できます。全角文字を使用すると，データ名を正しく特定できないことがあります。 [添字の扱い] 添字は，データ名に付加されます。 例： DATA1(1) DATA1 OF DATA2 OF DATA3(1,1,1) [フォーマット] エクスポートのフォーマットを次に示します。OF，空白，括弧，コンマは半角文字です。 データ名△OF△データ名…△(n,n,…) △：半角空白

項番	項目	内容
		n：添字を示す数字 インポート時も，上記のフォーマットで，かつ，すべてのレベル階層を OF 修飾子でつなげた名前でデータ項目が特定されます。ただし，FILLER は OF 修飾子を付けないで，FILLER のままで検索されます。 OF 修飾子は IN 修飾子でもかまいません。また，添字を示す括弧の前の半角空白は省略できます。添字のコンマの後ろに半角空白があってもかまいません。さらに，半角空白は連続していてもかまいません。 [デフォルト値] 設定対象のデータ項目名の文字列です。 [空欄の扱い] ファイルシミュレーション用のデータで，かつ終了条件の指定であれば無視されます。これ以外の場合はエラーになります。
5	値の表示	データ項目の値を表示する設定です。 表示する場合は「true」，表示しない場合は「false」です。 [デフォルト値] false [空欄の扱い] false として扱います。
6	期待値	データ項目の期待値です。このデータ項目の値の表示が表示しない設定の場合は，期待値を設定していても，自動判定対象にはなりません。 この項目が設定なしの場合は，期待値は設定されません。値の入力規則については，「4.4.8　値の入力規則」を参照してください。 [デフォルト値] なし。 [空欄の扱い] 期待値はないものとして扱います。
7	代入値	データ項目に代入する値です。この項目が設定なしの場合，データ項目に対する値は設定されません。値の入力規則については，「4.4.8　値の入力規則」を参照してください。 [デフォルト値] なし。 [空欄の扱い] 代入値はないものとして扱います。
8	設定個所固有情報（可変長）	テストデータ設定個所ごとの固有の情報です。設定個所によって格納する情報が変わります。 StartPoint 　なし。 EndPoint 　なし。 FileSim 　「表 A-2　FileSim の設定個所固有情報」を参照してください。

項番	項目	内容
		BreakPoint 文番号です。「表 A-3　BreakPoint の設定個所固有情報」を参照してください。 [デフォルト値] 対応するテストデータ設定個所で格納する情報が変わります。 StartPoint なし。 EndPoint なし。 FileSim 「表 A-2　FileSim の設定個所固有情報」を参照してください。 BreakPoint 文番号です。「表 A-3　BreakPoint の設定個所固有情報」を参照してください。

表 A-2　FileSim の設定個所固有情報

項番	項目	意味
1	ファイル名	シミュレーションする入出力文の対象となるファイル名です。 [デフォルト値] シミュレーションする入出力文の対象となるファイル名 [空欄の扱い] エラーになります。
2	プログラム名	シミュレーションする入出力文の対象となるファイルが定義されているプログラム名です。 テスト対象プログラムで宣言されている場合は，テスト対象プログラム名です。テスト対象プログラムの内側のプログラムで宣言されている場合は，内側のプログラム名です。 [デフォルト値] ファイルが定義されたプログラム名 [空欄の扱い] テスト対象プログラムが仮定されます。 注意 指定したプログラムが，テスト対象プログラムでも，その内側のプログラムでもない場合はエラーになります。指定したプログラムがテスト対象 COBOL ソースファイル内に存在していてもエラーになります。 プログラム名は，コンパイル後の名前で指定する必要があります。
3	オープンモード	シミュレーションする入出力文の対象となるファイルに対するオープンモードです。次の値のどれかです。 • INPUT • OUTPUT • I-O • EXTEND [デフォルト値] テストデータ設定対象のオープンモード [空欄の扱い]

項番	項目	意味
		エラーになります。
4	入出力文	テストデータを設定する入出力文です。次の値のどれかです。 • READ • WRITE • REWRITE [デフォルト値] テストデータ設定対象の入出力文 [空欄の扱い] エラーになります。
5	テストデータセット番号	入出力文が何回目に実行されたときに，テストデータを使用するのかを示す番号です。すべてのテストデータセットを削除した場合は，「-」が表示されます。 [デフォルト値] エクスポートのときは，テストデータを設定した番号 ひな形のときは，1 [空欄の扱い] 終了条件を指定する場合は無視されます。 終了条件の指定がない場合はエラーになります。
6	終了条件	入出力文に対して実行する終了条件です。次の値のどれかです。 • AtEnd：AT END 指定の文を実行させる。 • AtEndOfPage：AT END-OF-PAGE 指定の文を実行させる。 • InvalidKey：INVALID KEY 指定の文を実行させる。 • IOError：入出力エラーを発生させる。 終了条件を設定する場合は，単独で 1 行に記述する必要があります。 インポートした CSV ファイルで，終了条件に値が設定されていた場合，次の項目は無視されます。 • データ項目名 • データ項目のレベル番号 • 値の表示 • 期待値 • 代入値 • データ番号 [デフォルト値] なし [空欄の扱い] 終了条件なしとして扱います。

表 A–3　BreakPoint の設定個所固有情報

項番	項目	意味
1	文番号	中断点を設定した文番号です。文番号は，中断点を設定した行番号と，設定した文の位置番号をピリオドでつなげた形式です。登録集原文の展開後など，COBOL コンパイラが付与した番号です。コンパイルリストの行番号と一致します。形式を次に示します。

項番	項目	意味
		行番号[.位置番号] ピリオド以降の位置番号を省略すると，「.1」が仮定されます。 行番号は最大で 7 けた，位置番号は最大で 3 けたです。どちらかが超えた場合はエラーになります（例：00000001 の場合，数値としては 1 けたですがエラーになります）。 行番号の先頭の 0 は省略できます。例えば，1000 行目に設定した中断点の場合は，次のどれも同じ意味になります。 0001000.1 0001000.001 1000.1 1000.001 0001000 1000 エクスポート時は，0001000.1 の形式で出力されます。 [デフォルト値] 中断点が設定された文番号 [空欄の扱い] エラーになります。

(3)　ひな形の内容

ひな形としてエクスポートした CSV ファイルについて，各設定画面でテストデータとして出力する内容です。

表 A-4　ひな形で出力する内容

項番	テストデータの設定先	内容
1	[プログラムの開始時] 画面	プログラムの引数，返却項目の有無に関係なく，テストデータを何も設定していない状態で出力されます。
2	[プログラムの終了時] 画面	プログラムの引数，返却項目の有無に関係なく，テストデータを何も設定していない状態で出力されます。
3	[ファイルシミュレーション] 画面	[ファイルシミュレーション] 画面でテストデータを設定できる条件に合致する場合，そのファイルおよびオープンモードに合わせて，可能な組み合わせがすべて出力されます。 レコード領域が複数定義されていた場合は，最初に定義されているレコード領域を対象にし，2 つ目以降のレコード領域はコメント行で出力されます。 [ファイルシミュレーション] 画面でテストデータを設定できる条件に合致しない場合は，出力されません。 [ファイルシミュレーション] 画面でテストデータを設定できる条件については，「4.4.3　ファイルシミュレーションのテストデータを設定する」を参照してください。
4	[中断点] 画面	文の有無に関係なく，コメント行で出力されます。そのときの文番号は「0001000.1」です。

例

COBOL プログラムのソーステキスト

```
IDENTIFICATION  DIVISION.
PROGRAM-ID. 'MAIN'.
ENVIRONMENT       DIVISION.
INPUT-OUTPUT      SECTION.
FILE-CONTROL.
    SELECT  FILE01 ASSIGN TO 'FILE-B2046M1.txt'.
DATA DIVISION.
FILE              SECTION.
FD  FILE01.
01  REC0001 PIC X(5).
01  REC0002 PIC X(10).
WORKING-STORAGE  SECTION.
  01 DATA1 PIC X(10).
PROCEDURE  DIVISION.
  OPEN OUTPUT FILE01.
  WRITE REC0001.
  CLOSE FILE01.
END PROGRAM 'MAIN'.
```

ひな形 CSV の内容

```
* Version: 03-04
* テストプロジェクト名: SAMPLE_PROJECT
* プログラム名: MAIN
* COBOLソースファイル: C:¥temp¥main.cbl
***** ［プログラム開始時］画面の設定 *****
* <データ項目の設定個所の情報>,<参照フラグ>,<データ項目のレベル番号>,<データ項目名>,<値の表示>,<期待値>,<代入値>
StartPoint,,,RETURN-CODE,false,,
StartPoint,R,01,REC0001 OF FILE01,false,,
StartPoint,,01,REC0002 OF FILE01,false,,
StartPoint,,01,DATA1,false,,
***** ［プログラム終了時］画面の設定 *****
* <データ項目の設定個所の情報>,<参照フラグ>,<データ項目のレベル番号>,<データ項目名>,<値の表示>,<期待値>,<代入値>
EndPoint,,,RETURN-CODE,false,,
EndPoint,R,01,REC0001 OF FILE01,false,,
EndPoint,,01,REC0002 OF FILE01,false,,
EndPoint,,01,DATA1,false,,
***** ［ファイルシミュレーション］画面の設定（FILE01(MAIN/MAIN)/OUTPUT/WRITE）*****
* <データ項目の設定個所の情報>,<参照フラグ>,<データ項目のレベル番号>,<データ項目名>,<値の表示>,<期待値>,<代入値>,<ファイル名>,<プログラム名>,<オープンモード>,<入出力文>,<テストデータセット番号>,<終了条件>
FileSim,R,01,REC0001 OF FILE01,false,,,FILE01,MAIN,OUTPUT,WRITE,1,
* FileSim,,01,REC0002 OF FILE01,false,,,FILE01,MAIN,OUTPUT,WRITE,1,
***** ［中断点］画面の設定（0001000.1）*****
* <データ項目の設定個所の情報>,<参照フラグ>,<データ項目のレベル番号>,<データ項目名>,<値の表示>,<期待値>,<代入値>,<文番号>
* BreakPoint,,,RETURN-CODE,false,,,0001000.1
* BreakPoint,R,01,REC0001 OF FILE01,false,,,0001000.1
* BreakPoint,,01,REC0002 OF FILE01,false,,,0001000.1
* BreakPoint,,01,DATA1,false,,,0001000.1
```

(4) テストケースの CSV ファイルに関する注意

1. テストケースの CSV ファイルには，単体テスト支援の画面では制限値・限界値としている値を超えた設定ができますが，このように設定した CSV ファイルをインポートしても，画面では制限値・限界値を超えた部分の値は参照できません。ただし，設定した内容はテスト実行時には反映され，実行結果としては参照できます。

   テストケースの CSV ファイルの制限値・限界値を超えた値を設定した場合は，該当する行の読み込みをスキップし，エラーを表示し，次の行以降の読み込みを継続します。テストケース CSV の制限値・限界値については，「付録 C　単体テスト支援の制限値と限界値」を参照してください。

2. 期待値のための情報が設定されていて，かつ「代入値」に値が設定されていた場合の動作は次のとおりです。

   期待値のための情報とは，「値の表示」の値が true で「期待値」が空，または，「値の表示」の値が true で「期待値」に値が設定されている場合のことです。

   StartPoint の場合

   「代入値」の内容だけを適用します。

   EndPoint の場合

   期待値用の情報だけを適用します。

   FileSim の場合

   設定個所固有情報の「入出力文」の内容によって使用するものが変わります。「入出力文」の値が READ の場合は，「代入値」の内容だけを適用します。「入出力文」の値が WRITE または REWRITE の場合は，期待値のための情報だけを適用します。

   BreakPoint の場合

   両方適用します。

   どの場合でも，採用しなかった値について，エラーメッセージは出力されません。

3. 集団項目に対する設定をインポートした場合，「期待値」と「代入値」の項目は無視されます。エクスポートのときは空欄で出力されます。

4. インポートした CSV ファイルで次の内容が設定されている場合，設定する順番（CSV ファイル内の行の順）に関係なく，集団項目に対する「値の表示」は true になります。

   - 集団項目に対して設定する行（CSV ファイル内の行）がない。または，集団項目に対する「値の表示」が false または空欄。
   - 集団項目の下位要素の「値の表示」が true。

5. CSV ファイル内で，同じデータ項目に対してテストデータを設定する行が複数ある場合は，最後の設定が有効になります。

6. 特殊レジスタのレベル番号は無視されます。エクスポートのときは，空欄で出力されます。

7. FILLER に対する設定をインポートした場合，その設定はすべて無視されます。

(5) テストケースの CSV ファイルの例

COBOL プログラムのソーステキストとテストケースの CSV ファイルの例の説明です。

例 1：集団項目を含むテストデータ設定

COBOL プログラムのソーステキスト（抜粋）

```
IDENTIFICATION DIVISION.
PROGRAM-ID. MAIN.
DATA DIVISION.
WORKING-STORAGE SECTION.
01 DATA01.
  05 SUBDATA01 PIC X(3).
  05 SUBDATA02 PIC S9(3).
```

テストケースの CSV ファイルの例（抜粋）

```
* Version: 03-04
* テストプロジェクト名: SAMPLE_PROJECT
* プログラム名: MAIN
* COBOLソースファイル: C:¥temp¥main.cbl
***** ［プログラム開始時］画面の設定 *****
* <データ項目の設定個所の情報>,<参照フラグ>,<データ項目のレベル番号>,<データ項目名>,<値の表示>,<期待値>,<代入値>
StartPoint,R,01,DATA01,,,
StartPoint,R,05,SUBDATA01 OF DATA01,,,"""ABC"""
StartPoint,R,05,SUBDATA02 OF DATA01,,,123
***** ［プログラム終了時］画面の設定 *****
* <データ項目の設定個所の情報>,<参照フラグ>,<データ項目のレベル番号>,<データ項目名>,<値の表示>,<期待値>,<代入値>
EndPoint,R,01,DATA01,,,
EndPoint,R,05,SUBDATA01 OF DATA01,true,"""ABC""",
EndPoint,R,05,SUBDATA02 OF DATA01,true,123,
```

解説

行	データ項目の設定個所の情報	参照フラグ	レベル番号	データ項目名	値の表示	期待値	代入値
7	StartPoint	R	01	DATA01	－	－	－
8	StartPoint	R	05	SUBDATA01 OF DATA01	－	－	"""ABC"""
9	StartPoint	R	05	SUBDATA02 OF DATA01	－	－	123
12	EndPoint	R	01	DATA01	－	－	－
13	EndPoint	R	05	SUBDATA01 OF DATA01	true	"""ABC"""	－
14	EndPoint	R	05	SUBDATA02 OF DATA01	true	123	－

CSV ファイルの 7，8，9 行目の設定で，テスト対象プログラム名（MAIN）の開始時用データとしてテストデータが設定されます。

- 7 行目は集団項目用の設定のため，行自体はなくてもかまいません。代入値があっても無視されます。
- 8 行目の内容で，SUBDATA01 に文字列 ABC が代入されます。また，データ構造から，SUBDATA01 だけで名前解決が可能なら，OF 修飾子以降は不要です。
- 9 行目の内容で，SUBDATA02 に数値 123 が代入されます。

CSV ファイルの 12，13，14 行目の設定で，テスト対象プログラム名（MAIN）の終了時用データとしてテストデータが設定されます。

- 12 行目は集団項目用の設定のため，行自体はなくてもかまいません。期待値があっても無視されます。
- 12 行目では「値の表示」は空（= false）ですが，13，14 行目で値の表示を true にしているため，GUI で確認したときは，DATA01 の「値の表示」にもチェックが入ります。
- 13，14 行目で，SUBDATA01 と SUBDATA02 を期待値と比較しています。

例 2：中断点の設定

COBOL プログラムのソーステキスト（抜粋）

```
000100 ID DIVISION.
000200 PROGRAM-ID. MAIN.
000300 DATA DIVISION.
000400 WORKING-STORAGE SECTION.
000500   01 SAMPLEDATA PIC 9(9) VALUE 1.
000600 PROCEDURE DIVISION.
000800   DISPLAY SAMPLEDATA.
000900   IF SAMPLEDATA = 2
001000   :
```

テストケースの CSV ファイルの例（抜粋）

```
* Version: 03-04
* テストプロジェクト名: SAMPLE_PROJECT
* プログラム名: MAIN
* COBOLソースファイル: C:¥temp¥main.cbl
***** ［中断点］画面の設定（0000800.1）*****
* <データ項目の設定個所の情報>,<参照フラグ>,<データ項目のレベル番号>,<データ項目名>,<値の表示>,<期待値>,<代入値>,<文番号>
BreakPoint,R,01,SAMPLEDATA,true,1,,0000800.1
***** ［中断点］画面の設定（0000900.1）*****
* <データ項目の設定個所の情報>,<参照フラグ>,<データ項目のレベル番号>,<データ項目名>,<値の表示>,<期待値>,<代入値>,<文番号>
BreakPoint,R,01,SAMPLEDATA,true,2,2,0000900.1
```

解説

行	データ項目の設定個所の情報	参照フラグ	レベル番号	データ項目名	値の表示	期待値	代入値	文番号
7	BreakPoint	R	01	SAMPLEDATA	true	1	－	0000800.1
10	BreakPoint	R	01	SAMPLEDATA	true	2	2	0000900.1

CSV ファイルの 7 行目で，プログラムの 800 行目の DISPLAY 文に中断点を設定して，SAMPLEDATA を期待値と比較しています。

CSV ファイルの 10 行目で，プログラムの 900 行目の IF 文に中断点を設定し，次を設定しています。

- SAMPLEDATA を期待値と比較している（中断してすぐ）。
- SAMPLEDATA に 2 を代入している（上記の比較が終わったあと）。

例 3：ファイルシミュレーション用テストデータの設定

COBOL プログラムのソーステキスト（抜粋）

```
000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. MAIN.
000300 ENVIRONMENT DIVISION.
000400 INPUT-OUTPUT SECTION.
000500 FILE-CONTROL.
000600 SELECT FILE001 ASSIGN TO SYS010.
001000 DATA DIVISION.
001100 FILE SECTION.
001200 FD FILE001.
001300 01  USER.
001400   02 USERID   PIC 9(9).
001500   02 USERNAME PIC X(30).
002300 PROCEDURE DIVISION.
002400   OPEN INPUT FILE001.
002500 FILE-READ.
002600   READ FILE001
002700   AT END
002800     GO TO END-OF-FILE-READ
002900   NOT AT END
           :
004000     GO TO FILE-READ
004100   END-READ.
004200 END-OF-FILE-READ.
004300   CLOSE FILE001.
```

テストケースの CSV ファイルの例（抜粋）

```
* Version: 03-04
* テストプロジェクト名: SAMPLE_PROJECT
* プログラム名: MAIN
* COBOLソースファイル: C:¥temp¥main.cbl
***** ［ファイルシミュレーション］画面の設定（FILE001(MAIN/MAIN)/INPUT/READ）*****
* <データ項目の設定個所の情報>,<参照フラグ>,<データ項目のレベル番号>,<データ項目名>,<値の表示>,<期待値>,<代入値>,<ファイル名>,<プログラム名>,<オープンモード>,<入出力文>,<テストデータセット番号>,<終了条件>
FileSim,R,01,USER OF FILE001,,,,FILE001,MAIN,INPUT,READ,1,
FileSim,R,02,USERID OF USER OF FILE001,,,1,FILE001,MAIN,INPUT,READ,1,
FileSim,R,02,USERNAME OF USER OF FILE001,,,"""user1""",FILE001,MAIN,INPUT,READ,1,
FileSim,R,01,USER OF FILE001,,,,FILE001,MAIN,INPUT,READ,2,
FileSim,R,02,USERID OF USER OF FILE001,,,2,FILE001,MAIN,INPUT,READ,2,
FileSim,R,02,USERNAME OF USER OF FILE001,,,"""user2""",FILE001,MAIN,INPUT,READ,2,
FileSim,,,,,,,FILE001,MAIN,INPUT,READ,,AtEnd
```

解説

行	データ項目の設定個所の情報	参照フラグ	レベル番号	データ項目名	値の表示	期待値	代入値	ファイル名	プログラム名	オープンモード	入出力文	テストデータセット番号	終了条件
7	FileSim	R	01	USER OF FILE001	－	－	－	FILE001	MAIN	INPUT	READ	1	－
8	FileSim	R	02	USERID OF USER OF FILE001	－	－	1	FILE001	MAIN	INPUT	READ	1	－
9	FileSim	R	02	USERNAME OF USER OF FILE001	－	－	"""user1"""	FILE001	MAIN	INPUT	READ	1	－
10	FileSim	R	01	USER OF FILE001	－	－	－	FILE001	MAIN	INPUT	READ	2	－
11	FileSim	R	02	USERID OF USER OF FILE001	－	－	2	FILE001	MAIN	INPUT	READ	2	－
12	FileSim	R	02	USERNAME OF USER OF FILE001	－	－	"""user2"""	FILE001	MAIN	INPUT	READ	2	－
13	FileSim	－	－	－	－	－	－	FILE001	MAIN	INPUT	READ	－	AtEnd

7～12 行目でテストデータを設定しています。

- 7，8，9 行目で 1 回目の READ 文で代入する値を指定しています。
- 10，11，12 行目で 2 回目の READ 文で代入する値を指定しています。

13 行目では，終了条件を指定しています。

この内容でテストすると，READ 文が 3 回実行され，最初の 2 回ではデータが設定され（入力文のシミュレーション），3 回目の READ 文で AT END 側の処理に移ります。

付録 B　単体テスト支援で使用できる正規表現

単体テスト支援で使用できる正規表現の説明です。

検索で使用できる正規表現の構文規則を次に示します。

パターン	検索する対象
¥d	半角数字の 0 から 9 の 1 文字。
¥s	空白文字。空白文字を次に示します。 • 半角スペース • タブ • 改ページ
¥	正規表現の要素で使われる文字を通常の文字としてエスケープする場合に使います。 例 「¥*」や「¥.」は，「*（アスタリスク）」や「.（ピリオド）」を検索します。
.	任意の 1 文字。
*	直前の文字の 0 回以上の繰り返しと一致。 「?」を付けると最短で一致。
+	直前の文字の 1 回以上の繰り返しと最長で一致。 「?」を付けると最短で一致。
?	直前の文字の 0 回または 1 回の繰り返しと最長で一致。 「?」を付けると最短で一致。

正規表現の使用例

「DATA」で始まり，半角数字が 0 回以上繰り返す文字列を検索する場合

検索パターン：DATA¥d*

検索対象文字列：MOVE DATA10 TO DATA20 DATA21

一致する文字列：DATA10，DATA20，DATA21

「MOVE」で始まり「DATA」で終わる最長の文字列を検索する場合

検索パターン：MOVE.*DATA

検索対象文字列：MOVE DATA10 TO DATA20

一致する文字列：MOVE DATA10 TO DATA

「MOVE」で始まり「DATA」で終わる最短の文字列を検索する場合

検索パターン：MOVE.*?DATA

検索対象文字列：MOVE DATA10 TO DATA20

一致する文字列：MOVE DATA

付録 C　単体テスト支援の制限値と限界値

単体テスト支援の制限値と限界値の説明です。

「文字」は，全角文字・半角文字ともに 1 文字と見なします。

分類	項目	制限値，限界値
GUI	［ソーステキスト］画面に表示する行数	999,999 行
	一度に表示する［テストケース］画面の数	25 個
	一度に表示する［出力］画面のタブの数	25 個
	［出力］画面の［メッセージ］画面に表示する数	1,000 行
	［テストプロジェクト画面］で同時に選択して操作できるテストケースの数	100 個
	［データ項目の値の表示/期待値の設定］一覧に表示するデータ項目の数	1,500 個
	［データ項目の値の表示/期待値の設定］一覧で設定できるデータ項目の数	1,500 個
	［値の代入の設定］一覧に表示するデータ項目の数	1,500 個
	［値の代入の設定］一覧で設定できるデータ項目の数	1,500 個
テストプロジェクト	1 つのテストプロジェクトに含まれるテストケースの数	5,000 個
	1 つのテストプロジェクトに含まれる実行結果の数	100 個
	1 つのテストケースで設定できる中断点の数	500 個
	［データ項目の値の表示/期待値の設定］および［値の代入の設定］で入力できる文字の数	1,024 文字
	実行結果の名前の長さ	255 文字
	1 つの OCCURS 指定ありデータ項目の 1 つの次元でテストデータを設定できる個数	500 個
テストケースの CSV ファイル	［データ項目名］に記載できるバイト数	4,096 バイト
	1 行に記述できるバイト数	8,192 バイト
	［代入値］および［期待値］に記述できる文字数	1,024 文字
	次の設定画面用に，期待値を設定できる数 • ［プログラム開始時］画面 • ［プログラム終了時］画面 • ［中断点］画面（中断点ごと）	それぞれ 1,500 個
	次の設定画面用に，代入値を設定できる数 • ［プログラム開始時］画面 • ［プログラム終了時］画面	それぞれ 1,500 個

分類	項目	制限値，限界値
	• ［中断点］画面（中断点ごと）	
ファイルシミュレーション	［ファイルシミュレーション］画面で，1 テストデータセットに設定できる期待値の数	100 個
	［ファイルシミュレーション］画面で，1 テストデータセットに設定できる代入値の数	100 個
	ファイルシミュレーション用のテストデータとして，「ファイル名」，「オープンモード」，「入出力文」の組み合わせで 1 つのキーとしたときに設定できるテストデータセットの個数	100 個

注意

各要素の上限値以内でも，データの組み合わせによってはメモリ不足になることがあります。この場合は，次に示す目安の数値以内となるように，テストデータの設定を見直すか，テストプロジェクト内に保存するテスト実行結果の数を削減してください。

1. 1 つのテストプロジェクトに含められるテストデータの個数：50,000 個
   テストケースごとに設定したテストデータの個数の合計です。
2. 1 つのテストプロジェクトに含められる実行結果のデータ項目の表示対象の個数：750,000 個
   中断点の部分が繰り返し実行されたときの表示の個数もカウントします。
3. テスト対象の COBOL ソースファイルに含まれるデータ項目の個数：20,000 個
   OCCURS 句の繰り返し分の数を含みます。ただし，OCCURS 句の繰り返し数は 1 次元につき最大 500 までとします。この上限値を変更する場合，環境変数 CBLUTS_ANALYZEMAXOCCURSNUM で設定してください。詳細は，「5.2　COBOL ソースファイルを解析する」および「付録 D　単体テスト支援の環境変数」を参照してください。また，OCCURS 句を含む集団項目のデータ項目数は，使用するメモリ量がほかのデータ項目より多いため，2 倍した値で計算します。
4. テスト対象の COBOL ソースファイルの行数：100,000 行（COPY 文などの展開後の値とします）

付録 D　単体テスト支援の環境変数

単体テスト支援の環境変数の説明です。

CBLUTS_ANALYZEMAXOCCURSNUM

[概要]

OCCURS 指定ありデータ項目の 1 次元当たりに設定できるテストデータの個数の上限値（以降，上限値とします）を設定します。

例えば，この環境変数に 300 を設定した場合，OCCURS 句で指定した繰り返し回数が 300 以下のときは繰り返し回数まで，300 を超えるときは繰り返し回数 300 までのデータ項目にテストデータを設定できます。

［テストケースページ］の各設定画面の［データ項目の値の表示/期待値の設定］の一覧での表示も，この環境変数で設定した値を超える分は表示されなくなります。

[使用目的]

OCCURS 指定ありデータ項目の繰り返し回数や階層が深いとメモリ不足になることがあるため，1 次元当たりに設定できるテストデータの個数が 500 に制限されています。しかし，それでもメモリ不足になったときに，この環境変数を指定してメモリ不足を回避します。

この環境変数を使用するには，単体テスト支援を起動する前（開発マネージャから単体テスト支援を起動する場合は，開発マネージャを起動する前）に，次のどちらかの環境変数に設定する必要があります。

- Windows の［ユーザ環境変数］
- Windows の［システム環境変数］

[設定できる値]

1～500 までの数値。

設定されていない場合，または不正な値を設定した場合は，500 が仮定されます。

[環境変数の値を上限値として使用するタイミング]

上限値として使用する値は，タイミングによって異なることがあります。

「テストプロジェクトファイルを開いた場合」と，「新しくテストプロジェクトファイルを作った場合」に分けて説明します。

テストプロジェクトファイルを開いた場合

既存のテストプロジェクトファイルを開いたときの上限値は，前回テストプロジェクトファイルを保存したときに使用していた値です。環境変数の値が上限値として使われるのは，「COBOL ソースファイルの解析」を実行したあとからです。

この環境変数がサポートされる前に保存したテストプロジェクトファイルを開いた場合は，上限値は 500 です。

新しくテストプロジェクトファイルを作った場合

新しくテストプロジェクトを作成したときの上限値は，環境変数の値です。

環境変数に値が設定されていない，または不正値が設定されていた場合は，上限値は 500 です。

［注意］

- バッチモードでは，この環境変数は使用しません。設定されていても無視します。
- この環境変数を開発マネージャ上で設定しないでください。
- この環境変数の変更で，実際に上限値が変わる OCCURS 指定ありデータ項目がある場合，そのデータ項目を含む集団項目下に設定したテストデータは，「COBOL ソースファイルの解析」をしたあとの引き継ぎ対象になりません。

  そのため，該当する集団項目下のデータ項目にテストデータを設定していた場合は，「COBOL ソースファイルの解析」をしたあとにテストデータを再設定してください。

例：

OCCURS 句で指定した繰り返し回数	「COBOL ソースファイルの解析」をする前		「COBOL ソースファイルの解析」をしたあと		引き継ぎ
	上限値※1	見え方※2	上限値※3	見え方※2	
200	200	200	500	200	○※4
600	400	400	500	500	×※5
200	200	200	100	100	×※5
600	500	500	300	300	×※5

（凡例）

○：引き継がれる

×：引き継がれない

注※1

前回テストプロジェクトファイルを保存したときに使用していた値，または 500。

注※2

［テストケースページ］の各設定画面の［データ項目の値の表示/期待値の設定］の一覧に表示される OCCURS 句ありデータ項目の繰り返し回数。

注※3

環境変数に設定した値，または 500。

注※4

「COBOL ソースファイルの解析」する前と後で上限値が変わりますが，実際にテストデータを設定できるデータ項目の数が変わらないため，テストデータは引き継がれます。

注※5

「COBOL ソースファイルの解析」する前と後で上限値が変わることで，実際にテストデータを設定できるデータ項目の数が変わるため，テストデータは引き継がれません。

付録 E　各バージョンの変更内容

各バージョンの変更内容を示します。

変更内容（3021-3-618）　COBOL2002 Developer Professional 04-00，COBOL2002 Developer Professional(64) 04-00

追加・変更内容
単体テスト支援のテストプロジェクトを作成する開発マネージャのメニュー階層を変更した。
ファイルシミュレーションの設定時の注意について記載を追加した。
テストケースごとに実行環境ファイルまたは実行時環境変数を設定できるように変更した。これに伴い，次に示す内容を変更した。 • ［プロパティページ］に［テストケースごとに実行環境ファイルまたは実行時環境変数を設定する］チェックボックスを追加した。 • ［テスト環境の設定］画面を追加した。
次に示すメッセージを追加した。 KEDL0068T-E，KEDL0069T-E，KEDL1034T-W，KEDL1035T-W，KEDL3053T-W，KEDL3054T-W，KEDL3055T-E

付録 F　このマニュアルの参考情報

このマニュアルを読むに当たっての参考情報を示します。

付録 F.1　関連マニュアル

関連マニュアルを次に示します。必要に応じてお読みください。

COBOL2002 Professional 製品

COBOL2002 Professional 製品 導入ガイド（3021-3-615）
COBOL2002 Professional Tool Kit COBOL ソース解析ガイド（3021-3-616）
COBOL2002 Professional Tool Kit データ影響波及分析ガイド（3021-3-617）

COBOL2002

COBOL2002 ユーザーズガイド（3021-3-600）
COBOL2002 操作ガイド（3021-3-601）
COBOL2002 言語 標準仕様編（3021-3-604）
COBOL2002 言語 拡張仕様編（3021-3-605）
COBOL2002 メッセージ（3021-3-609）

付録 F.2　このマニュアルでの表記

このマニュアルでは，マイクロソフト製品の名称を次のように表記しています。

マニュアルでの表記		製品名
Windows 7	Windows 7(x86)	Microsoft Windows 7 Professional 日本語版(32 ビット版)
		Microsoft Windows 7 Enterprise 日本語版(32 ビット版)
		Microsoft Windows 7 Ultimate 日本語版(32 ビット版)
	Windows 7(x64)	Microsoft Windows 7 Professional 日本語版(64 ビット版)
		Microsoft Windows 7 Enterprise 日本語版(64 ビット版)
		Microsoft Windows 7 Ultimate 日本語版(64 ビット版)
Windows 8.1	Windows 8.1(x86)	Windows 8.1 Pro 日本語版(32 ビット版)
		Windows 8.1 Enterprise 日本語版(32 ビット版)
	Windows 8.1(x64)	Windows 8.1 Pro 日本語版(64 ビット版)
		Windows 8.1 Enterprise 日本語版(64 ビット版)

マニュアルでの表記		製品名
Windows 10	Windows 10(x86)	Windows 10 Pro 日本語版(32 ビット版)
		Windows 10 Enterprise 日本語版(32 ビット版)
	Windows 10(x64)	Windows 10 Enterprise 日本語版(64 ビット版)
		Windows 10 Pro 日本語版(64 ビット版)
Windows Server 2008 R2		Microsoft Windows Server 2008 R2 Standard 日本語版
		Microsoft Windows Server 2008 R2 Enterprise 日本語版
		Microsoft Windows Server 2008 R2 Datacenter 日本語版
Windows Server 2012		Microsoft Windows Server 2012 Standard 日本語版
		Microsoft Windows Server 2012 Datacenter 日本語版
Windows Server 2012 R2		Microsoft Windows Server 2012 R2 Standard 日本語版
		Microsoft Windows Server 2012 R2 Datacenter 日本語版
Windows Server 2016		Microsoft Windows Server 2016 Standard 日本語版
		Microsoft Windows Server 2016 Datacenter 日本語版
Windows Server 2019		Windows Server 2019 Standard 日本語版
		Windows Server 2019 Datacenter 日本語版

- Windows 7，Windows 8.1，Windows 10，Windows Server 2008 R2，Windows Server 2012，Windows Server 2012 R2，Windows Server 2016，および Windows Server 2019 とで機能差がない場合，Windows と表記しています。

このマニュアルでは，製品名を次のように表記しています。

表記	製品名
COBOL2002	COBOL2002 Net Developer
	COBOL2002 Net Developer(64)

付録 F.3　KB（キロバイト）などの単位表記について

1KB（キロバイト），1MB（メガバイト），1GB（ギガバイト），1TB（テラバイト）はそれぞれ 1,024 バイト，$1{,}024^2$ バイト，$1{,}024^3$ バイト，$1{,}024^4$ バイトです。

索引

記号

-All　207, 212, 220
-ClearCoverage　219
-Create　214
-Delete　206, 220
-Detail　204
-DisplayCoverage　218
-Export　211, 215
-ExportTemplate　213
-Force　205
-Import　208, 210, 218
-InputPath CSV ファイルが格納されているフォルダパス　208
-InputPath CSV ファイルのパス　210
-InputPath プロジェクト情報ファイルのパス　204, 214, 218
-List　207, 221
-Name 実行結果名　220
-Name テストケース名　206, 212
-OutputPath 出力先フォルダパス　212, 213
-OutputPath プロジェクト情報ファイルのパス　215
-OverWrite　209, 210, 212, 213, 215
-Project テストプロジェクトファイルのパス　205, 207, 209, 211–215, 218–221
-Quiet　207, 218–220
-Recursive　209
-Rename　209, 211
-Run　204
-Skip　208, 210, 212, 213, 215
-Source COBOL ソースファイルのパス　214
-Test　204
-TestCase　206–208, 210, 211, 213
-TestProject　214, 215, 217
-TestResult　218–221

数字

1 つのテストケースを実行する　73
32bit 用テストモジュールフォルダ　195
64bit 用テストモジュールフォルダ　195

C

cbluts コマンド － バッチモードで単体テスト支援を実行する　204
COBOL ソースファイル　245
COBOL ソースファイルを解析する　84
CSV ファイルをインポートする（ファイル指定）　112
CSV ファイルをインポートする（フォルダ指定）　111

T

TD コマンド格納ファイル　245
TD コマンド格納ファイルフォルダ　195

W

Windows のエクスプローラーから単体テスト支援を開始する　78
Windows のプログラム一覧からテストプロジェクトを作成する　34

あ

新しい環境でプログラムの動作確認をやり直す　128
［新しいテストプロジェクトを作る］ダイアログ　157

い

一括バッチを実行しデグレード有無を確認する　16
色を変更する　152
インフォメーションバー　161

え

エビデンスファイルを確認する　98
エラー時に採取するファイル　243
エラー時の対処　222
エラー情報ファイル　243
エラー情報ファイルの格納先　243
エラーになった COBOL プログラムを修正する　83

か

開発マネージャからテストプロジェクトを作成する 30
カバレージ情報をクリアする 105, 124
カバレージ情報を表示する 124
［カバレージレポート］画面 192
カバレージレポートを参照する 96
画面のリファレンス 156
環境変数 262
環境をカスタマイズする 140

き

擬似プログラム用プログラム情報ファイル 245
期待値を設定する 52

こ

コマンドの指定形式 201
コマンドリファレンス 200
コンパイラが生成するファイル 245

さ

サブフォルダにある CSV ファイルを一括でインポートする 134

し

実行可能ファイル 245
実行環境ファイル 245
実行結果の操作 124
実行結果の名前を変更する 102
実行結果フォルダ 195
実行結果を一覧表示する 126
実行結果を削除する 103, 125
実行結果をテストプロジェクトから除外する 104
シミュレーション 12
［出力］画面 184
手動で判定結果を変更する 100
新規開発時のモジュールテストをする 14

す

［スタートページ］ 165
［スタブ設定ページ］ 168
すべてのテストケースを実行する 72, 108

せ

正規表現 259
制限値と限界値 260

そ

［ソーステキスト］画面 170

た

大量の COBOL ソースファイルから一括でプロジェクトファイルを作成する 127
単体テスト支援で使用するファイル 245
単体テスト支援で使用できる正規表現 259
単体テスト支援とは 12
単体テスト支援の画面を変更する 152
単体テスト支援の環境変数 262
単体テスト支援の基本的な使い方 17
単体テスト支援の作業の手順 28
単体テスト支援の準備をする 24
単体テスト支援の紹介 11
単体テスト支援の制限値と限界値 260
単体テスト支援のファイル一覧 245
単体テスト支援のメインウィンドウ 160
単体テスト支援の利用方法 14
単体テスト支援を実行する 18
単体テスト支援を終了する 76
単体テストを実行する 71

ち

［中断点］画面 179
中断点のテストデータを設定する 50
中断点を解除する 52
中断点を設定する 49

て

データ項目検索の設定を変更する 154
データ項目に値または期待値を設定する 52
［テスト環境の設定］画面 182

［テストケース結果レポート］画面　186
テストケース結果レポートを参照する　91
テストケースごとに実行環境ファイルと実行時環境変数を設定する　67
テストケースごとに実行環境ファイルまたは実行時環境変数を設定する　138
テストケースの CSV ファイル　246
テストケースの CSV ファイルの規則　246
テストケースの CSV ファイルの形式　247
テストケースの実行　108
テストケースの操作　110
テストケースの名前を変更する　38
テストケースの有効と無効を切り替える　40
［テストケースページ］　170
テストケースを CSV ファイルにエクスポートしインポートする　142
テストケースを一覧表示する　111
テストケースをインポートする（CSV ファイルをテストケースとしてインポートする場合）　149
テストケースをインポートする（指定したフォルダの直下にあるすべての CSV ファイルをテストケースとしてインポートする場合）　147
テストケースをエクスポートする（すべてのテストケースを CSV ファイルにエクスポートする場合）142
テストケースをエクスポートする（選択したテストケースを CSV ファイルにエクスポートする場合）144
テストケースをエクスポートする（テストケース指定）　113
テストケースをエクスポートする（ひな形）　114
テストケースをエクスポートする（ひな形のテストケースをエクスポートする場合）　146
テストケースをコピーして追加する　37
テストケースを削除する　39, 110
テストケースを作成する　37
テストケースを追加する　37
テスト結果を参照しプログラムを修正する　23
テスト結果を参照する　89
テスト結果を操作する　99
テスト実行の設定を変更する　155
［テスト設定］画面　164
テストデータを設定する　41
テストできるプログラムかどうかを確認する　25
［テストプロジェクト］画面　162
［テストプロジェクト結果レポート］画面　185
テストプロジェクト結果レポートを参照する　90
テストプロジェクトの環境を設定する　61
テストプロジェクトのスタブを編集する　66
テストプロジェクトの操作　115
テストプロジェクトのファイル構成　195
テストプロジェクトのプロパティを表示する　64
テストプロジェクトファイル　195, 245
テストプロジェクトを移動する　141
テストプロジェクトを作成する　29, 115
テストプロジェクトを保存する　75
テストモジュールフォルダ　195
テストを繰り返す　82

は

バッチモードで実行する　106
バッチモードでできる単体テスト支援の機能　107
バッチモードでできる単体テスト支援の使用例　127

ふ

［ファイルシミュレーション］画面　176
ファイルシミュレーションのテストデータを設定する　44
ファイルのリファレンス　194
フォントを変更する　153
複数のテストケースまたは実行結果を一括で削除する　136
複数のテストケースを一括でエクスポートする　135
［プログラム開始時］画面　172
プログラム開始時の値を設定する　41
プログラム修正後のデグレード有無を確認する　15
［プログラム終了時］画面　175
プログラム終了時の値を設定する　43
プログラム情報ファイル　245
プログラムの単体テストをする　27
プロジェクト情報ファイルの構成要素　197
プロジェクト情報ファイルを出力する　116

プロジェクト情報を設定する　117
プロジェクト情報を変更する　129
［プロパティページ］　166

ほ

ほかのテストプロジェクトにあるテストケースを流用する　132

め

［メッセージ］画面　184
メッセージから探す〔トラブルシュート〕　223
メッセージの一覧　223
メッセージの出力形式　223

り

リグレッションテストなどのために繰り返しテスト実行する　130