

一日次一

- 1. はじめに
- 2. コンパイル時のオプション
- 3. テストデバッガの起動と終了
- 4. ソースの表示方法と中断点の設定
- 5. データの内容表示と代入
- 6. データのトレース
- 7. エディタとの連携
- 8. デバッガの色などの変更
- 9. カバレージ情報の蓄積と表示
- 10. カウント情報の表示
- 11. 終わりに

### 1. はじめに

この章では、COBOL2002専用のテストデバッグツールの使用方法について説明します。

既に「1. COBOL2002の起動から実行まで」を読み終わっているものとして、説明を行います。

例題プログラムは1章で作成したプログラム(reidail)を使用します。

ここでは、COBOL2002専用のテストデバッグツールの機能のうち、特に 知っておいて頂きたい基本的な機能について説明します。COBOL教育で は、この基本的な操作を理解すれば十分と考えますが、更に詳細を知 りたいという場合には、マニュアル「COBOL2002操作ガイド」、 「COBOL2002ユーザーズガイド」を参照ください。

# 2. コンパイル時のオプション

テストデバッグツールを使用するためには、COBOLプログラムのコンパ イル時に「コンパイラオプション」を指定する必要があります。

1章で説明したように、デフォルトオプションの設定で「-TDInf」を指定 してあれば、ここでオプションを指定する必要はありません。ここで は、デフォルトオプションを設定していないものとして手順を示しま す。

[手順1] 開発マネージャのメニューバーの「プロジェクト(P)」をクリックし、プ ルダウンメニューの「プロジェクトの設定(S)」をクリックします。 すると、コンパイラオプションの一覧が出力されます。

w 開発マネージャ for COBOL2002 - samp hmf	I XI
	ا السالة
$\frac{7^{\circ}}{2}\sqrt{2}\sqrt{2}\sqrt{2}\sqrt{2}\sqrt{2}\sqrt{2}\sqrt{2}\sqrt{2}2$	
🤹 🚔 🔚   🦉 ங 🤞 7泊ジェ外の作成(Д) 7泊ジェ外名の変更(N)	
□     □     7°ロジェウトのエウスホ°-トQ0       ⑦*ロジェウトのインボ°-トQ     □	xI
Generation           (作業中ブロンジェクトの設定\w)             Generation           (作業中プロンジェクトの設定\w)             Generation           (作業中プロンジェクトの設定\w)             Generation           アロジェクトの設定\w)             Generation           アロジェクトの関係設定(R)	
■ reidai1.cbl ソースファイルの追加(F)	
7位ジェ外の設定(2) 2	
国 かセージウィントウ     依存ファイルの設定(D)       低存ファイルの仮登録(T)	4
カハ <sup>*</sup> レージ <sup>*</sup> ( <u>O</u> )	
フ <b>゚</b> ロノペティ( <u>P</u> )	
ा ।	Iلح
7 U / I / I / I / I / NUM / I / I / NUM / I / I / NUM / I / I / I / I / NUM / I / I / I / I / I / I / I / I / I /	//



オプションの一覧が出力されます。
-TDInfをチェックして「OK」ボタンをクリックします。

プロジェクト設定	×
7泊シュークト設定 7泊シューŷト一覧(P): ・ ■ reidai1 - reidai1.exe 設定内容	<ul> <li>アカジェクト 製品連携 実行 最適化 アバックア リック 丸く ト</li> <li>DebugLine: デバックバラを有効にする</li> <li>DebugInf: 異常終了時、エラー要約1情報を出力する</li> <li>Trace: 異常終了時、エラー要約1情報を出力する</li> <li>Trace: 異常終了時、エラー要約1情報を出力する</li> <li>DebugCompati: 添字:指標名の反復回数の範囲外とフロ</li> <li>DebugData: データ例外検出</li> <li>TDInf: デスト方 バック 1情報出力</li> <li>ObbugRange: 添字、指標名の反復回数範囲外チェック</li> <li>TestOmd: TDコマント 18納ファイルの出力</li> <li>Full: 中断点情報、ジュレーション情報のTDコマント 18納ファイル出力</li> <li>Sim: ジュレーション情報のTDコマント 18納ファイル出力</li> <li>Sim: ジュレーション情報のTDコマント 18納ファイル出力</li> </ul>

[用語解説] コンパイラオプション

コンパイラオプションは、コンパイラがオプションでサポートしている機能を使うときに 指定します。背反する仕様を使い分けるために用意されているオプション等もあります。

デバッグが完了すると、そのプログラムにはデバッグ情報は不要になります。デバッグ情報の出力がオプション機能になっているのは、完成したプログラムに余分な情報を持たな くてもよいよう配慮している意味もあります。 [手順2] コンパイラオプションの一覧の中から「デバッグ」タブの「-TDInf」の「□」をクリックして「レ」印をつけます。 「レ」印が設定されたことを確認して、「0K」ボタンをクリックします。



#### [ワンポイントアドバイス]

オプション設定の画面を見てわかるように、多数のコンパイラオプションがあります。 オプションの意味はマニュアルを参照し、必要のないオプションは指定しないようにしま しょう。

### [手順3] 開発マネージャ画面に戻りますので、改めてコンパイルを行ってく ださい。

輕閒発マネージャ for COBOL2002 - sample01.hmf	
$\mathcal{T}^{T}$	
🚰 sample01.hmf	
🔁 sampleO1	
📄 🗊 reidai1 - reidai1.exe	I
$\prod_{n=1}^{\infty} \frac{1}{n} \sum_{k=1}^{\infty} \frac{1}{n} \sum_{k$	I
III initial.cbl	I
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□	I
$b\bar{\tau}^{*}$	NUM //

# 3. テストデバッガの起動と終了

-TDInfオプションを指定して、コンパイル&リンケージが終わると、 テストデバッグツールを使用することができます。(「実行」ボタンを クリックすれば、実行だけすることもできます。)

[手順1] 開発マネージャのメニューバーの「ビルド(B)」をクリックし、プルダウ ンメニューの中の「デバッガ(D)」をクリックします。すると、デバッグ するプログラムの確認画面が出ますので、「OK」ボタンをクリックしま す。

延開発マネージャ for COBOL2002 - sample01	hmf	
プロジェクトマスタ( <u>M</u> ) ファイル( <u>E</u> ) プロジェクト( <u>P</u> )	ビルト(国) ツール(田) ウィントウ(畑) ヘルフ(田)	
•3 ≥ ■   3 ™ § №   ■ ?	ビルド(B) くううう ① Ctrl+B りイックビルド(Q) コンパイル(G) リトゴルド(B)	
Sample01.hmf		X
Sampleur □···· □· reidai1 - reidai1.exe □··· □· V-スファイル	ビルドの中止(S) 依存関係更新( <u>A</u> )	
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□	デバッガ( <u>D</u> )	2
	実行(E) Ctrl+E	
COBOL2002 テストデンバッカを起動する	実行支援(1)	NUM //



テストデバッグ対象の設定		×
実行可能ファイル( <u>N</u> ):		
C:¥temp¥sample01¥reidai1¥w	ork¥reidai1.exe 🔽 💌	参照( <u>B</u> )
		·疑垠(A)
		<u></u>
		削除( <u>R</u> )
		全削除(W)
作業フォルダ( <u>F</u> ):		
	•	参照(0)
	3 ОК	キャンセル

[手順2]以下のような画面が出力されます。主な各部の機能は以下の説明の通 りです。基本的には速度調整して実行すれば、デバッグを行っていき ます。次の章以降は、デバッグの基本的な操作方法を説明します。



[手順3] デバッグの方法を説明する前に終了方法を説明しておきます。 終了方法は、Windowsの基本操作に沿った3通りの方法があります。

③ ダブルクリック	①「テス リックし ニューか	トデバッガの制術 て表示されるプ/ ら「終了」をクリ、	卸(F)」をク レダウンス ソクする。	7	Ċ	2 クリック
🔗 テストテッハッカ for COBOL200	)2 - reidai1.e	exe				
テストデバッガの制御(E) 編集	/表示(E) フ	プログラムの制御( <u>C</u> )	中断( <u>B</u> )	データの操作( <u>D</u> )	ツール(工)	ሳሪኑን 🖤
テストデンバック対象の設定(工)	Ctrl+O					
TDコマント、格納ファイルの入力(』 初期化(N)	)	) =x =x 🕅 🔊	B) [		7	10 10   Part
🏷 テストデバッガの終了 🖄						

# 4. ソースの表示方法と中断点の設定

さて、いよいよプログラムのテストデバッグの開始ですが、まず、プ ログラムの実行部と定義部の表示方法から説明します。

実行部に関しては先に表示しておかなくても、プログラムを実行すれ ば自動的に表示されます。ただし、事前に中断点等を設定したい場合 には、予め表示しておく必要があります。

[手順1] テストデバッグ画面の一覧ウィンドウの中のプログラム名をクリックし、そのままマウスの右ボタンをクリックすると「実行部」と「定義部」のプルダウンメニューが表示されます。これらのどちらかをクリックすると選択したものがそれぞれの画面に表示されます。



[手順2] 中断点を設定します。実行画面のスクロールバーを調整して中断したい文に位置付けます。この例では2900行目のIF文の所まで移動しています。ここで、文字列の「IF」をダブルクリックすると、赤色の枠で囲まれます。これで、中断点を設定できました。この文に制御が渡ってきた時点で(この文を実行する直前で)止まります。

マストデットがのがの制御(E) 編集/表示(E) プログラムの制御(C) 中断(E)       F: 大の(操作(E)) ツール(D) ウィルウ(W) ヘルプ(H)       IF 文をダブル クリックする と赤枠で囲まれます。         アータの操作(E) ツール(D) ウィルウ(W) ヘルプ(H)       アール(D) ウィルウ(W) ヘルプ(H)       アール(D) クィルウ(W) ヘルプ(H)         アータの操作(E) ツール(D) ウィルウ(W) ヘルプ(H)       アール(D) ロックする と赤枠で囲まれます。         アータの操作(E) ツール(D) ウィルウ(W) ヘルプ(H)       アール(D) ロックする         アータの操作(E) ツール(D) ウィルウ(W) ヘルプ(H)       アール(D) ロックする         アータの(P) マール(D) ワール(D) マール(D) ロックする       アール(D) ロックする         アータの(P) (P) アール(D) ロション(P) ロション(P	ピラストデヘシッガ for COBOL2002 - reidai1.exe	
C	テストデバッガの制御(E) 編集/表示(E) プロゲラムの制御(C) 中断(B) データの操作(D) ツール(T) ウィンドウ(W) ヘルフ°(H)	IF文をダブル クリックする
Prime Part of the former control of t	◎ • • • • • • • • • • • • • • • • • •	と赤枠で囲ま れます。
Image: Control Contr	Tei tei tiei tixi	
Image: cidail       スクロールバーで調整して目的の場所に移動の場所に移動する。         0000200       000200 PROGRAM-ID. reidail.         0000300       000300*         0002800       002800 比較処耳         0002900       002900         0002900       002900         0003000       IF         回 E=3       IF命令をダブルクリックすると色が変わります。		
0000200 0000200 000200 PROGRAM-ID. reidail. 0000300 000200 PROGRAM-ID. reidail. 0000300 000300 × 1 00002900 002800 比較処理 SECTION. 0002900 002900 IF 月 = 9 0003000 002900 THEN ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	Image: reidail     Image: reidail       0000100     000100     IDENTIFICATION DIVISION	スクロールバー で調整して目 めの場所に移
0000300 000300 0002800 002800 比較処町 SECTION. 0002900 002900 IF 月 = 9 0003000 003000 THEN IF命令をダブルクリックすると色が変わります。	0000200 000200 PROGRAM-ID. reidail.	動する。
0002800 002800 比較処理 SECTION. 0002900 002900 IF 月 = 9 0003000 003000 THEN ■ EIS IF命令をダブルクリックすると色が変わります。		
□ 0002300 002300 II ) / - 3 0003000 003000 THEN ■ E <sup>-</sup> ■ E <sup>-</sup> IF命令をダブルクリックすると色が変わります。	0002800 002800 比較処理 SECTION. ▲	
■ IF命令をダブルクリックすると色が変わります。	0003000 003000 THEN	
□□□-> IF命令をタフルクリックすると色が変わります。		
	□□□□-> □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□	
<u>ドーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーー</u>		1

### [ワンポイントアドバイス]

このように文やタグ名称をダブルクリックすると、赤色に変わって中断点が設定されま す。ここでは、ダブルクリックをして設定しましたが、文等を黒く反転させそのままマ ウスの右ボタンをクリックするとプルダウンメニューが表示されます。ここで、「中断 点の設定/解除」を選んでもできます。なお、設定を解除する場合はもう一度ダブルクリ ックして、色を戻せば解除されます。 [手順3]中断点はいくつ設けてもかまいませんが、必要最小限にした方がよい と思います。中断点の設定ができたら、「連続実行」ボタンをクリックして、実行 してみましょう。

「連続実行」ボタンをクリック(①)すると「実行」画面が表示されますので、「OK」ボ (1) タンをクリック(②)してください。
データの操作(D) (D ウィントゥ) 小リフ°(H)
🗗 😳 🐘 🐕 🕸 🕸 🍇 😄   ≕ 🖾   🔊   🔝   実行をしようとしている文 の色が変わります。先に中
「「「」」「」」「」」「」」「」」」「」」」「」」」「」」」」」」」」」」
reidail
0000100 000100 IDENTIFICATION DIVISION. 0000200 000200 PROGRAM-ID. reidail. 0000300 000300*
* 0002800 002800 比較処理 SECTION. 0002900 002900 IF 月 = 9 0003000 003000 THEN
KCCC1205T-I COBOLブログラムの実行を開始 KCCC1208T-I 中断点です。2900 <reidai1 f<br="">エーザンパラメタ(P):</reidai1>
7℃17~5ム中断中
モータ回面には、 甲断点  じ止まったことを示す   メッセージが出力されます。

## 5. データの内容表示と代入

### データの内容を表示したり、データに値を設定することができます。

# [手順1] データ名称をダブルクリックすると、モニタ画面にデータの内容が表示されます。



#### [ワンポイントアドバイス]

このようにデータ名をダブルクリックすると、データの内容が表示されます。ここでは、 ダブルクリックをして表示しましたが、データ名をクリックし、マウスの右ボタンをク リックして表示されるプルダウンメニューから「データ値の表示」を選んでもかまいま せん。 [手順2] データ名に値を設定するには、データ名称をクリックしてそのままマ ウスの右ボタンをクリックすると、プルダウンメニューが出ますので、 その中の「代入」を選択してください。





# [手順3] データ名に正しく設定されたかチェックして、1ステップだけ実行してみます。(この操作は確認のためにやっているだけです)



クリック(①)して、表 示してみます。 値が「09」である(②)こ とを確認します。 1ステップだけ実行し てみます(③)。 実行状態を示す色が1 ステップだけ、移動し

ます(④)。

データ名「月」をダブル

14

6. データのトレース

次は、データの状態を常に表示する方法を説明します。

[手順1] 定義画面を適当にスクロールして、表示したいデータ名を見つけます。 表示するデータ名をクリックして、そのままマウスの右ボタンをクリ ックして表示されるプルダウンメニューから「常時データ表示に設定」 をクリックします。この例では、データ名「YYMMDD」を常時データ表示 します。

1997 テストラ データ	(トテ <sup>・</sup> ハ <sup>・</sup> ッカ・for( デハ・ッカ・の制御( の操作( <u>D</u> ) ッ	COBOL2002 - reidai1.exe E) 編集/表示(E) プログラム/ ール(T) ウィンドウ(W) ヘルプ(圧	 の制御(C) 中断(B) )			
10	• 睽 鄂 鄂	x=   🗢 🖟 🗶 🖉 🐛	E E 🔊 🕅			
#->I	er lei xi		5			
	L DATA PRO					
	eidai1					
*	0001200 0001300 0001400 0001500 0001600 0001700	001200 01 YYMMDE 001300 02 年 PI 001400 02 月 PI 001500 02 日 PI 001600* 001700 PPOCEDURE	コピ <sup>。</sup> ー(Q) すべて選択( <u>A</u> ) ソースファイルの編集(E) 文字列の検索(E) ジャンプ <sup>(</sup> 」) ジャンプ <sup>(</sup> 」)			
*	0001700 0001800 0001900	001700 PROCEDURE 001800 001900* 002000 Main-Sac S	データ値の表示(D) データ値の代入( <u>V</u> ) 中断点の設定/解除(B) データ監視条件の設定(S) 業時デーム表示(ご設定(T)			
*	0002000	002000 Merri Sec 3 002100 PERFOF	定義デーゲーの表示(P)			
カスタマイズ心し… PIETタ RCCC1205T-I COBOLプロクラムの実行を開始しました。 KCCC1206T-I COBOLプログラムの実行を終了しました。 KCCC1201T-I ユーザブログラムの実行を終了しました。 常時データ表示ウィントウにデータ名を設定する						

[ワンポイントアドバイス] 常時表示を行うデータ名はいくつ指定してもかまいませんが、必要最小限にとどめた方がよいと思います。 [手順2]常時表示画面が表示されてその中に先に指定したデータ名称が表示されます。データ名の+の箇所をクリックすると、下位項目まで表示されます。この常時表示画面のデータ名の右にデータ名の値が変化した際に値が表示されます。



### 7. エディタとの連携

デバッグ中にプログラムを修正したくなった場合は、以下のやり方で簡単に エディタを自動起動できますので、そこで修正してください。

[手順1] デバッグ画面中の実行画面・定義画面の該当する行にカーソルを位置 付け、右クリックをするとプルダウンメニューが出ます。 そこで、「ソースファイルの編集(E)」を選択してください。エディタが起



17

[手順2] 修正が完了したら、「上書き保存」してエディタを終了します。 デバッグ画面に戻りますが、デバッグは一旦終了してください。とい うのは、エディタで修正した部分は現在のデバッグ情報には反映され ていませんので、改めてコンパイルし直す必要があるからです。した がって、デバッグを終了して開発マネージャに戻り、再コンパイルし てから再度デバッグをしてください。

www.ligetheter.com/www.com/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/states/state	_ 🗆 ×
ר איז דער איז	
Sample01.hmf	
Sample01 I reidail.exe I 用コンパイル I 承示ンパイル I 体存ファイル	
メッセージウィントウ	
νī <sup>s</sup> i	

# 8. デバッガの色などの変更

ここでは、デバッガの画面の配色と、フォントの変更方法を説明します。

[手順1] デバッグ画面のメニューバーの「ツール(T)」をクリックするとプルダ ウンメニューが出ます。この中の「カスタマイズ(U)」をクリックする とカスタマイズ画面が出ますので、そこで、色やフォントの設定を 行ってください。 △①



19



れます。

# 9. カバレージ情報の蓄積と表示

カバレージとは、テスト進捗状況を定量的に把握する機能です。カバレージ情 報を採取するには、コンパイラオプション-CVInfを指定してコンパイルします。 カバレージ情報には、次の3種類の指標があります。

①COメジャー:実行した文の割合を示します。

COメジャー=(実行が済んだ文の数) / (実行文の数)×100 (%)

②C1メジャー:分岐する個所で、実行した分岐先の割合を示します。

C1メジャー=(実行が済んだ分岐先の数)/(分岐先の数)×100(%)

③S1メジャー:実行した呼び出し文(CALL文やINVOKE文)の割合を示します。

S1メジャー=(実行が済んだ呼び出し文の数)/(呼び出し文の数)×100(%) これらの情報は、プログラム情報ファイルに蓄積されます。プログラム情報ファ イルは、実行可能ファイルと同じフォルダに作成されます。

### [ワンポイントアドバイス]

プログラム情報ファイルを実行可能ファイルとは別のフォルダに作成したい場合は、コン パイル時の環境変数CBLPIDIRでフォルダを指定します。コンパイル時の環境変数は、コン パイラオプションを指定するときと同様に「プロジェクトの設定(S)」をクリックし、コン パイラオプション一覧の中から「環境変数」タブをクリックして設定します。

[手順1] カバレージ情報の蓄積

開発マネージャのメニューバーの「プロジェクト(P)」をクリックし、プルダウンメニューの 中の「カバレージ(0)」をクリックします。すると、カバレージのウインドウが開かれます。



21



🛗 カバレーシ <sup>®</sup> for COE	30L2002				_ 🗆 ×
カバレージの制御(工)	カバレージ情報(C)	カウント情報( <u>N</u> )	∧μフ°( <u>H</u> )		
Π					
לד <b>י</b> ז					NUM //

[手順2] カバレージ情報の蓄積

カバレージウインドウのメニューバーの「カバレージ情報(C)」をクリックし、 プルダウンメニューの「蓄積(A)」をクリックすると、「カバレージ情報の蓄積」 画面が表示されます。「実行可能ファイル(N)」の参照ボタンで実行可能ファイル を指定します。「OK」ボタンをクリックすると、プログラムが実行されカバレー ジ情報が蓄積されます。

### [ワンポイントアドバイス] デバッガからの実行でもカバレージ情報を蓄積することができます。

(1)		
たまたのでは、「The Cobole and Cob		×
カバレージの制御(T) カバレージ情報(C) カウント情報(N)	^/l/7°( <u>H</u> )	
カバレージ情報を蓄積する		//.



	カパレーショ情報の蓄積	×	
カパレージ for COBOL2002 カパレージの制御(T) カパレージ情報	実行可能ファイル( <u>N</u> ):		-0×
1	DLLファイル( <u>D</u> ):		
		削除( <u>P</u> )	
		全削除())	
	作業フォルダ( <u>E</u> ):		
		参照( <u>S</u> )	
	ユーザ <sup>*</sup> ハ <sup>*</sup> ラメタ( <u>P</u> ):		
לד <sup>י</sup> ז	ОК	キャンセル	



カバレーシット情報の蓄積	
憲力パレージ for COBOL2002     ま行可能ファイル(N):     アパレージでの制御(T) カパレージで     すいじージで     すいしージで     すいしージ     すいしージで     すいしージ     すいしージで     すいしージで     すいしージ      すいしージ      むいしージ      むいしージ      むいしージ      むいしージャ      むいしージ      むいし	×
実行可能ファイル ? メ	
ファイルの場所型: 🔂 work 🔽 🖛 🗈 💣 囲*	
reidail.exe (2)	
ファイル名(N): 開((Q)	
ファイルの種類(I): 実行可能ファイル(*.exe) <	
νσ <sup>2</sup> 1 ΟΚ ++ντελ	//.





ዀ カパレージ for COBOL2002	
カバレージの制御(工) カバレージ・情報(C) カウント情報(N) ヘルフ・(H)	
KCCC1200T-I ユーザブログラムの実行を開始しました。 KCCC1201T-I ユーザプログラムの実行を終了しました。 KCCC3405T-I カバレージ情報の蓄積を行いました。翻訳単位名(REIDA KCCC0317T-I カバレージ情報の蓄積を終了します。 ┃	NII) ファイ
	Þ
b7*1	NUM ///

[手順3] カバレージ情報の表示

カバレージウインドウのメニューバーの「カバレージ情報(C)」をクリックし、 プルダウンメニューの「表示(D)」をクリックします。すると、「カバレージ情報 の表示」画面が表示されるので、プログラム情報ファイル(.cbp)を指定します。





かルージ情報の表示。	?	×
ファイルの場所①:	🔄 work 🔽 🗲 🗈 💣 🎟	
reidai1.cbp		
	プログラム情報ファイルをクリックします。	
, ファイル名(N):	開(()	
ファイルの種類(工):	7 <sup>°</sup> ログラム情報ファイル(*.cbp) マンマンセル	

[手順4] カバレージ情報の表示

「カバレージ情報の表示」画面の中の表示したい項目をクリックします。ここでは、「翻訳単位の一覧表示」と「まとめ表示」をクリックし、「OK」ボタンをクリックします。

カバレージ情報の表示	×
☞ 翻訳単位の一覧表示(1)	
▼ まとめ表示(M)	+++)++
□ ソース表示(S)	
□ 差分ソース表示(D)	
□ 未実行ソース表示(U)	
□ 差分未実行ソース表示(№)	
□ 呼び出し文ソース表示(A)	
出力ファイル名(Q):空白時は, プログラム情報ファイル名.cll	
D:¥MATUNO¥sample¥COBOL2002¥sample01¥reidai1¥work¥reidai1.	参照( <u>B</u> )



🎆 カバレージ for COI	BOL2002				
カバレージの制御(工)	カバレーショ情報(C)	カウント情報( <u>N</u> )	^//フ°( <u>H</u> )		
KCCC1200T-I KCCC1201T-I KCCC3405T-I KCCC0317T-I KCCC3427T-I	ユーザプログ ユーザプログ カバレージ情 カバレージ情	ラムの実行 ラムの審積 の審 報リストを	を開始しま を終了しま にま しまし	ました。 ました。 な。翻訳単位名 す。 した。ファイル	(REIDAI1) ファイ (D:¥sample¥reid
•					Þ
レディ					

[手順5] カバレージ情報の表示

実行可能ファイルと同じフォルダに.cllという拡張子のファイルが生成されてい ます。このファイルをCOBOLエディタやメモ帳で開いて、カバレージ情報を見る ことができます。カバレージ情報の表示例を次ページに示します。

### [ワンポイントアドバイス]

同じ条件で複数回実行してもテスト回数は1回として扱われます。実行ルートが異なるテ ストをする度にカバレージ情報は蓄積されます。

🔍 D:¥sample¥reidai1¥work						_ D ×
ファイル(E) 編集(E) 表示(⊻)	お気	に入り( <u>A</u> ) ツール(①)	ヘルプ(円)			<b>11</b>
⇔戻る・⇒・ 🖻 🔍検索	<b>B</b>	フォルダ 🧭 🖺 🖻	$\times \mathfrak{o}$	<b>*</b>		
アドレス(D) 🔂 work						▼ 🔗移動
フォルダ	×	名前 △		サイズ	種類	更新日時
📄 💼 💼 R DRV		📓 <u>reidai1.cbo</u>		1 KB	CBO ファイル	2004/04/01 19:10
📃 🗔 sample		Y reidai1.cbp		12 KB	CBP ファイル	2004/04/06 18:37
📄 💼 reidai1		reidai1.cll		3 KB	CLL ファイル	2004/04/06 18:37
work		reidai1.exe	1	9 KB	アプリケーション	2004/04/01 19:10
E ⊕ _ TEMP		neidai1.exp	$\odot$	1 KB	EXP ファイル	2004/04/01 19:10
TEMP1		🔊 <u>reidai1.lib</u>		2 KB	LIB ファイル	2004/04/01 19:10
TEMP2		🔊 <u>reidai1.obi</u>		6 KB	OBJ ファイル	2004/04/01 19:10
ТЕМРЗ		🛋 <u>reidai1.rc</u>		1 KB	RC ファイル	2004/04/01 19:10
□		🔊 <u>reidai1.res</u>		1 KB	RES ファイル	2004/04/01 19:10
- Address						
Agent						
🗄 🛅 Desktop	Ţ					
		•				
種類: CLL ファイル サイズ: 2.87 KB					2.87 KB	🖳 דר בטעב אד 🏿

[カバレージ情報の表示例] 「まとめ情報」の例を次に示します。 0.0%\* 0.0% 差分01 0 0 実行済数 差分01-2004-04-06 18:37:15 0 0 対象総数 50.0% 50.0% 5 実行済数 ς̈́ < 差分31> 0 0.0% 0 0 2004-04-06 18:37:15 0 2 2 対象総数 髧 テスト回数 変更回数 I 0.0% 0 0 0 < S1 > 差分00 0.0% 0.0% 韔 틁 0 0 実行済数 ") < 港分01> 0 0.0% 差分00-0 С I \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* د 対象総数 0 0 ぐ 50.0% 2 カバレージ情報 < C1 > R 88.8% 88.8% 8 × < 進分 (0) > 0 0.0% C 0 ω ω 実行済数 コンパイル日時: 2004-04-01 19:10:39 : 2004-04-06 18:37:03 ş 88.8% 6 6  $\infty$ 6 対象総数 <00> 01-01 : REIDAI1 01-01 種別 ۲ COBOL2002 (X) COBOL2002 (X) カバレージ率 プログラム名 実行済数 対象総数 未実行数 テスト日時 첛 REIDAI1 包 佑 蕃昻 × × × × × ×

# 10. カウント情報の表示

カウント情報は、プログラム中の文の実行回数を示します。

[手順1] カウント情報の表示

カバレージウインドウのメニューバーの「カウント情報(N)」をクリックし、プル ダウンメニューの「表示(D)」をクリックすると、「カウント情報の表示」画面が表 示されます。

🛗 カバレージ for CO	BOL2002	77				
カバレージの制御(工)	カバレージを情報(C)	<u>カウント情報(N)</u>	€7°( <u>H</u> )			
KCCC1200T-I KCCC1201T-I KCCC3405T-I KCCC0317T-I KCCC3427T-I	ユーザプログ ユーザブログ カバレージ情 カバレージ情	<u>表示(D)</u> フムの実行 報の蓄積を 報りストを	を開始しま を終了しま 行いました 出力しまし	€した。 €した。 ⊑。翻訳単位= 「。 ,た。ファイ,	名(REIDAI1) ル(D:¥sampl	ファイ e¥reid
•						
カウント情報を表示する	5				NUM	

[手順2] カウント情報の表示

「カウント情報の表示」画面で、「実行可能ファイル(N)」の参照ボタンをクリック し実行可能ファイルを指定します。

🎬 ከለጌታን	かント情報の表示 実行可能ファイル( <u>N</u> ):		
カバレージの制	DLLファイル(D):		
KCCC120			
KCCC340		育(B余( <u>R</u> )	)ファイ
KCCC031 KCCC342		全前時余(量)	ple¥reid
	作業フォルダ( <u>F</u> ):		
		▼ 参照( <u>S</u> )	
	ユーサ <sup>*</sup> ハ <sup>*</sup> ラメタ( <u>P</u> ):		
カワンド)音報版を:	[	•	M
	出力ファイル名(Q):空白時は,実行ファイル名.cnl		
		OK キャンセル	



<mark>គ្គី                                   </mark>	かンド情報の 実行可能 パージ パの 1107 DLLファイル 実行可能ファイル	表示 ファイル(Ŋ): (D):	× ▼ 菱照四	
KCC( KCC( KCC( ・ カウント情	ファイルの場所・ Indiai1.exe	): 🔁 work / ②   実行可能ファイルをクリック	←	ファイ +¥reid
	」 ファイル名(N): ファイルの種類の	D: 実行可能ファイル(*.exe)	「開く(①)     「     キャンセル     OK     キャンセル     ・	



	がい情報の表示		×	
	実行可能ファイル( <u>N)</u> :			
🎆 カバレージー	D:¥sample¥reidai1¥work¥reidai1.exe	▼	( · · · · · · · · · · · · · · · · · · ·	
カバレージの制 レビンク120	DLLファイル( <u>D</u> ):			
KCCC120			選択( <u>A</u> )	
KCCC340 KCCC031			削除( <u>R</u> )	ファイ
KCCC342	·		全削除())	le¥reid
•	」 作業フォルダ( <u>E</u> ):			Þ
カウント情報を表		<b>_</b>	参照( <u>S</u> )	
	ユーザ <sup>*</sup> ハ°ラメタ( <u>P</u> ):			
		<b>_</b>		
	出力ファイル名(型):空白時は,実行ファイル名.cnl			
	1	<b>_</b>	参照( <u>E)</u>	
		ОК	キャンセル	30

All Rights Reserved. Copyright © 2008, Hitachi, Ltd.



🎬 カバレージ for COI	BOL2002				
カバレージの制御(工)	カバレーショ情報(C)	カウント情報( <u>N</u> )	^ルフ°( <u>H</u> )		
KCCC1200T-I KCCC1201T-I KCCC0317T-I KCCC3427T-I KCCC1200T-I KCCC1201T-I KCCC3428T-I	ユーザプログ プログ サブージレージ カバレザプログ オウント情報	ラムの実行 ラムの蓄積 報リスの ラムの ラムの ミリスト を 出	を静終了した。それの時代の時代では、それの時代では、それの時代です。それでは、それの時代では、このでは、そので、このでは、そので、こので、こので、こので、こので、こので、こので、こので、こので、こので、	ました。 ました。 す。 した。ファイ した。 ました。 た。ファイル	ル(D:¥sample¥reid (D:¥sample¥reidai
ντ. ντ <sup>*</sup> ί					

[手順3] カウント情報の表示

実行可能ファイルと同じフォルダに. cnlという拡張子のファイルが生成されています。このファイルをCOBOLエディタやメモ帳で開いて、カウント情報を見ることができます。

🔍 D:¥sample¥reidai1¥work					
ファイル(E) 編集(E) 表示(⊻)	お気	に入り(A) ツール(T)	ヘルプ(円)		10 A
⇔戻る・⇒・ 🖻 🔇検索	6	フォルダ 🎯 🖺 🖻	X 🛛 🗐 🎟 -		
アドレス(D) 🔂 work					▼ 🔗移動
フォルダ	x	名前 △	サイズ	種類	更新日時
📄 💼 💼 R DRV		😫 <u>reidai1.cbo</u>	1 KB	CBO ファイル	2004/04/01 19:10
□		reidai1.cbp	12 KB	CBP ファイル	2004/04/06 18:37
📄 🛄 reidai1		i <u>reidai1.cll</u>	3 KB	CLL ファイル	2004/04/06 18:37
work		reidai1.exe	9 KB	アプリケーション	/ 2004/04/01 19:10
E TEMP		🛋 <u>reidai1.exp</u>	1 KB	EXP ファイル	2004/04/01 19:10
TEMP1		🔊 <u>reidai1.lib</u>	2 KB	LIB ファイル	2004/04/01 19:10
		🔊 <u>reidai1.obi</u>	6 KB	OBJ ファイル	2004/04/01 19:10
ТЕМРЗ		🛋 <u>reidai1.rc</u>	1 KB	RC ファイル	2004/04/01 19:10
□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □		reidai1.res	1 KB	RES ファイル	2004/04/01 19:10
Address		reidai1.cnl	🔲 (1) 4 КВ	CNL ファイル	2004/04/06 18:53
Agent	그				
種類: CNL ファイル サイズ: 3.06 KB				3.06 KB	🖳 マイ コンピュータ 🛛 🎢

### [カウント情報の表示例]

	******	
	* カウント情報 *	
C0B0L2002	(X) 01-01 *********************************	004-04-06 18:53:00
プログラム名	: REIDAI1	
コンパイル日	時: 2004-04-01 19:10:39	
実行日時	: 2004-04-06 18:53:00	
プログラム名 : 実行回数	RE I DA I 1	
	0001700 PROCEDURE DIVISION. aa	
	0001701	
	0001900 Mein-Sec SECTION.	
1	0002000 PERFORM 初期処理.	
1	0002100 PERFORM 比較処理.	
1	0002200 PERFORM 出力処理.	
1	0002300 STOP RUN.	
	0002500 初期処理 SECTION.	
1	0002600 ACCEPT YYMMDD FROM DATE.	
	0002800 比較処理 SECTION.	
1	0002900 IF 月 = 9	
	0003000 THEN	
0	0003100 MOVE 'September!!' TO DATA2	
	0003200 ELSE	
1	0003300 MOVE 'Not September!!' TO DATA2	
	0003400 END-IF.	
	0003600 出力処理 SECTION.	
1	0003700 DISPLAY DATAO.	
	0003800	
	0003900	

# 11. 終わりに

テストデバッグツールにおける基本的な使用方法は以上の説明で終わりです。一通りのデバッグを行う場合には、今までの説明の機能だけで十分であると思います。

しかし、テストデバッグツール自身には、その他の機能も備わっていますので、それらをお知りになりたい方は、マニュアル「COBOL2002操作ガイド」を参照ください。