スクールCOBOL2002

<u>スクール COBOL2002</u>

<u>操作入門書</u>



(2008年3月作成)

目 次

1. COBOL2002起動から実行まで	3
2. テストデバッガの使用方法	36
3. 関連資料	69
(a)ファイルの入出力処理	70
(b)テストデータの作成方法	80
(c)用紙の節約方法	90
(d)印刷書式の設定方法	96
(e)エディタ設定方法	101
(f)コンパイルリストの入手方法	114
(g)オンラインマニュアルの使用方法	124
(h)登録集原文の指定方法	132
(i)サブプログラムの追加方法	140
(j)索引ファイルを新規に作成する方法	144
(k)既存のプロジェクトマスタファイルの開	き方 147



一初めてCOBOL2002を使う方のために —



- 1. はじめに
- 2. COBOL2002の起動
- 3. プロジェクトマスタファイルの作成
- 4. プロジェクトの作成
- 5. COBOLソースプログラムの編集
- 6. コンパイル(実行用ファイルの生成)
- 7. 実行
- 8. プロジェクトの追加
- 9. 終わりに

1. はじめに

これから、COBOL2002の使い方を順を追って説明していきます。

スクールCOBOL2002には次の2種類の製品があります。

- ・Windows Vista[®]対応版スクールCOBOL2002
- ・Windows XP/2000 対応版スクールCOBOL2002

どちらの製品をお使いになっても操作手順は同じです。COBOL2002のメニュー画面 の形式も変わりません。スタートボタンやウインドウのデザインが異なる等の違いはありますが、手順そのものは同じです。

本資料では、スタートボタンからCOBOL2002を起動するまでの流れだけ、Windows Vista[®] 対応版とWindows XP/2000対応版に分けて説明します。起動後の操作につ きましては、Windows XP/2000の画面表示例を使って説明します。

なお、Windows Vista[®] 対応版には次の制約がありますので、ご注意ください。

《Windows Vista[®] 対応版の制約事項》

・プログラム名として日本語を使用することはできません。

(プログラム名段落のプログラム名, CALL文で呼び出すプログラム名等)

操作手順の説明に入る前に、まず次の注意事項をお読みください。

《注意事項》

COBOLのソースプログラムや実行形式ファイルは特定のフォルダの下に作ります。 後で削除等しやすいように、練習用のフォルダを作成してください。本説明書で は、Cドライブの下にtempという名称のフォルダを作成し、その下にsample01とい うフォルダを作成するものとして、説明を進めて行きます。

2. COBOL2002の起動

COBOL2002を起動するには、スタートボタンから行う方法と、COBOL2002の各種ツ ールをアイコン化しておいて、そのアイコンを(ダブル)クリックする方法があり ます。ここでは、スタートボタンから行う方法を説明します。

(1) Windows Vista[®] 対応版スクールCOBOL2002をお使いの場合



[手順3] COBOL2002の使用できるツールの中から、COBOL2002開発マネージャを 選んでクリックします。すると、COBOL2002開発マネージャが起動され ます。



[用語解説] COBOL2002開発マネージャ

日立COBOL2002を用いてコーディングからコンパイル、テスト、実行等の操作をコントロールするツールです。基本操作ではこのツールを使用します。

(2) Windows XP/2000 対応版スクールCOBOL2002をお使いの場合

[手順1] スタートボタンを押し、「プログラム(P)」の所にマウスポインタを移動 します。すると起動できるプログラムの一覧が出てきます。



[手順2] プログラムの一覧の中からCOBOL2002の所にマウスポインタを移動しま す。するとCOBOL2002の使用できるツール一覧が表示されます。



[手順3] COBOL2002の使用できるツールの中から、COBOL2002開発マネージャを 選んでクリックします。すると、COBOL2002開発マネージャが起動され ます。



3. プロジェクトマスタファイルの作成

COBOL2002を起動したあと、まず、ソースプログラムの作成からコンパイル・実行までの一連の作業を行うための環境を整えます。

ソースプログラムの作成から実行までを行うために必要な各種リソースやコンパ イラオプション等の管理を行うファイルをプロジェクトマスタファイルと言いま す。最初に「デフォルトオプションの設定」をしてからプロジェクトマスタファ イルを作成します。

[デフォルトオプションについて]

コンパイラに対するオプション情報をコンパイラオプションといいます。 コンパイラオプションは、「プロジェクトの設定」メニューから設定することが できますが、プログラムを作成するたびに設定しなければなりません。全てのプ ログラムに共通のコンパイラオプションは、デフォルトオプションで設定してお くと便利です。デフォルトオプションを設定しておくと、新規にプログラムを作 成するとき、コンパイラオプションの初期値(設定済の扱い)として有効になりま す。

[手順1] デフォルトオプションの設定

基本操作では、次のコンパイラオプションを設定します。

①コンパイルリストの出力

コンパイルした結果のリストが出力されます。

②デバッグ情報の出力

テストデバッガを使用してデバッグするために必要な情報が出力されます。

開発マネージャの画面から、プロジェクトマスタファイルを開いていない状態で 設定します。メニューバーの「プロジェクトマスタ(M)」-「デフォルトオプションの設定(D)」を クリックしてください。「デフォルトオプションの設定」画面が表示されます。

■愛開発マネージャ for COBOL2002 コケンジェクトフスタ(M) ツール(T) ヘルコ*(H)		
→ 1→ 1→ 1→ 1→ 1→ 1→ 1→ 1→ 1→ 1→ 1→ 1→ 1→	Ctrl+0	🕾 🗽 🔍 🎗 🌮 🛛 🥔 🏷
テフォルトオフ ションの設定(型)		

デフォルトオフ*ション設定
製品連携│実行│最適化│デバッグ│リンク│月格│移行◀▶
 -SQL: XDM/RD又はODBCIご準拠するSQLを使用可能にする ト XDM: リレーショナルデータへ~スペDM/RD)操作シュシレーション機能 L + ODBC: 埋込みSQL文をODBCインタフェース機能で使用す -SQLDisp: 用途が表示用のデータ項目をSQL文に指定可能と -RDBTran: ソース中に記述したCOMMIT/ ROLLBACK文をRD -IsamExtend: 索引ファイル機能にBtrieveを使用できる L Zone: Btrieveファイルで外部10進キー属性を有効にする
□-XMAP: XMAP3を使用する □ └ LinePrint:書式印刷機能を使用する □-OpenTP1: OpenTP1を使用したDC機能を使用する
▲ ▼
OK キャンセル

[手順2] 該当するコンパイラオプションの設定

コンパイラオプションは、カテゴリ別に分かれて表示されます。該当するカテゴリのタブをクリックしてからコンパイラオプションをチェックします。

「デバッグ」タブをクリック(下記画面参照)し、次のオプションをチェックする。
 「デバッグ」タブ: ロ-TDInf:テストデバッグ情報出力

②右矢印()) をクリックして「リスト出力」タブを表示し、次のオプションを チェックする。

「リスト出力」タブ: □ ├ OutputAll:全てのソースをリストに展開する

二つのオプションのチェックが完了したら、「Enter」キーを押すか「OK」ボタンを クリックします。



[手順3] プロジェクトマスタファイルを新規作成します。 開発マネージャのメニューバーの「プロジェクトマスタ(M)」-「新規作成(N)」を クリックしてください。すると新規作成画面が表示されます。

	<mark>፼</mark> 開発マネージャ for COBOL2002		
•	プロジェクトマスタ(M) ツール(T) ヘルプ(H)	
	→ 新規作成(N)		110 fil 11r 11/ (0, 🗩 🛛 🚄 🏠
	開(@)	Ctrl+O	
	テフォルトオフジョンの設定(<u>D</u>)		
	開発マネージャの終了♡		
	ν ντ [*] 1		

[手順4]新規作成画面の「参照(R)」ボタンを押します。 すると、「フォルダの参照」画面が表示されます。

泊ジュウトマスタの作成		×
プロジェクトマスタ名(<u>M</u>):		
プロジェクトマスタフォルダ名(<u>F</u>):	C:¥Program Files¥Hitachi¥Cobol2002¥E	
	完了	キャンセル
フォルダの参照		? ×
このフォルダでよ	ければ、OKを押して下さい。	
	APPGALL Cobol2002 Good 2002 Good 2002	
	OK	キャンセル

[手順5]「フォルダの参照」画面で、右側のスクロールバーを使って、Cドライブの tempフォルダを探します。

フォルダの参照	<u>? ×</u>	
このフォルダでよければ、OKを押して下さい。		表示位置を調整する。上下にスクロールし、
OK キャン1	b11	

[手順6]「tempフォルダ」を選び「Enter」キーを押すかまたは「OK」ボタンをクリックします。

フォルダの参照	<u>? ×</u>
このフォルタでよければ、OKを押して下さい。	
・	
Brogram Files	
	-
ОК	キャンセル

[手順7] プロジェクトマスタ名に「sample01」と入力し、「Enter」キーを押すかま たは「完了」ボタンをクリックしてください。

ንግንንድንትዋスタの作成	×
ንግንንェንኮマスጵ名(<u>M</u>):	sample01
フロジェクトマスタフォルが名(圧):	C:¥temp¥sample01 参照(B)
	完了キャンセル
	「Enter」キーを押す。

[手順8] 次の画面が表示されるので、「Enter」キーを押すかまたは「OK」ボタンを クリックしてください。

C:¥temp¥sample01は存在しません。作成しますか?
OK キャンセル
「Enter」キ

4. プロジェクトの作成

プロジェクトマスタは、プロジェクトという単位でリソースを管理します。 基本操作では、Cドライブのtempフォルダの下にsample01というプロジェクトマ スタを作成し、その中にreidai1というプロジェクト(プロジェクト名はソース ファイル名と同じにします)を作成します。このときのフォルダ構成は次のよう になります。



[概要] プロジェクトの作成 プロジェクトは、次の操作で作成します。

・入力要求には「プログラム名」を入力し、あとは全て「Enter」キーを押すだけで 作成できます。

詳細の手順を、次に示します。

[手順1] プロジェクトマスタファイルの設定に続いて次の画面が表示されます。 プロジェクト名「reidai1」を入力して「Enter」キーを押してください。 (「OK」ボタンをクリックしてもよい。)

プロジェクトの作成	
7泊ジュウト名(N): reidai1 「reidai1」と入力し、	「Enter」キーを押す。
7 [°] ロジェクトフォルダ′(<u>P</u>):	
C:¥temp¥sample01¥reidai1 参照(B)	[ワンポイントアドバイス]
作業フォルダ(<u>W</u>): work 参照(<u>F</u>)	プログラム名は3回入力します。 毎回 入力するとスペルミスをするこ
□ プロジェクトフォルダと作業フォルダを同一にする。Φ	とがあるので、カット&ペーストで貼
OK ++2211	り付けるとよいでしょう。

[手順2] 次の画面が順に出ますので、続けて「Enter」キーを押してください。 (「OK」ボタンをクリックしてもよい。)

開発マネーシ	°r for COBOL2002
⚠	C:¥temp¥sample01¥reidai1 は存在しません。作成しますか?
	OK キャンセル
	「Enter」キーを押す。
開発マネージャ	for COBOL2002
	C:¥temp¥sample01¥reidai1¥work は存在しません。作成しますか?
	OK キャンセル
	「Enter」キーを押す。

[手順3] プロジェクトの追加画面が表示されます。デフォルトの「実行形式プ ログラム(E)」を指定すればよいので、この画面も「Enter」キーを押し てください。(「次へ(N)」ボタンをクリックしてもよい。)

ንግንንድንՒመንፅከጠ: COBOLንግጉንንድንՒ	×
最終生成物の種類を選択してください。	
● 実行形式プログラム(E)	
○ ダイナミックリンクライフ [*] ラリ(<u>D</u>)	
◎ 標準ライブラリ(L)	
	_
〈戻る(8) 次へ(N) > キャンオ	216
「Enter」キーを押	す。

[手順4] プロジェクトの種類もデフォルトの「System指定メインプログラム」で よいので、「Enter」キーを押してください。 (「次へ(N)」ボタンを押してもよい。)

ንግንኙታՒመነፅ加:COBOLንግንኙታՒ	×
つかジェクトの種類を選択してください。	
System指定メインプログラム	
メインプログラムなし OLEアウトオブプロセスサーバプログラム	
CGIメインプログラム	
	1
	_
< 戻る(<u>B</u>) 次へ(<u>N</u>) > キャンセル	
「Enter」キーを押す	0

[手順5] 出力ファイル名に「プログラム名」(プロジェクト名と同じ名称)を入力 して「Enter」キーを押してください。(「次へ(N)」ボタンを押してもよ い。)

ንግንዥያኑውነይ加:COBOLንግንዥያኑ	×
出力ファイル名(EXE)を入れてください。(Q)	
reidai1 参照(B)	
「reidai1」と入力し、「Enter」キーを押す。	
	-
〈戻る(B) 次へ(N) 〉 キャンセル	

[手順6] メインファイル名も「プログラム名」(プロジェクト名と同じ名称)を入 カします。続いて「Enter」キーを押してください。 (「完了」ボタンをクリックしてもよい。)

プロジェクトの追加:COBOLプロジェクト	×
メインファイル名を入れてください。(M)	
reidai1 参照(R)	
「reidai1」と入力し、「Enter」キーを押す。	
,	
< 戻る(B) 完了 キャンセル	

[手順7] ファイルの種別もデフォルトの「COBOLソース(固定形式)」でよいので、 「Enter」キーを押してください。 (「OK」ボタンをクリックしてもよい。)

reidai1	×
ファイルの種別を指定して下さい。(E): COBOLソース(固定形式) COBOLソース(自由形式)	OK キャンセル
	「Enter」キーを押す。

これでプロジェクトの設定が完了です。プログラム名の入力が3回、あとは 「Enter」キーを押していく(8回)だけでできました。

[手順8] 開発マネージャの画面に戻り、実行ファイルとソースファイルフォル ダ、依存ファイルフォルダ、メッセージウィンドウが表示されます。

[ワンポイントアドバイス]

- ここで、一度保存しておくとよいでしょう。作業終了時に開発マネージャを閉じるとき、 保存するか否かを聞いてきますが、うっかり「いいえ」をクリックしてしまうとせっかく設 定してきた内容が失われます。作業終了時にも保存する必要はありますが、ここで保存し ておけば全てを失うことは避けられます。
- ②メッセージウインドウが表示されない場合は、開発マネージャのメニューバーの「ウインドウ(W)」をクリックし、プルダウンメニューの中の「並べて表示(T)」をクリックしてください。実行ファイルとメッセージウインドウが並べて表示されます。他に「重ねて表示(C)」もできます。

	COBOL2002 - sampl	ieur.nmi			
プロジェクトマスタ(<u>M</u>)	ファイル(E) プロジェクト((P) ビルド(B)	ツール①	ウィンドウ₩	ヘルフ°(<u>H</u>)
] 🔁 🖻 🔚 (🦉 ங 😼 😼		😳 🔛	× 🔍 🌮	🎒 🦢
	<u> </u>				
Sample(11 hmf					_
sample01					_
🖻 🗊 reidai1 -	reidai1.exe				
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□	P1ル pイル				
					_
V7 1					INOM
			1		
፩開発マネージャ fo	r COBOL2002 - samp	Die01.hmf			
■ Z ご 開発マネージャ fo フ [*] ロジェクトマスタ(<u>M</u>)	r COBOL2002 - samp ファイル(<u>E</u>) ጋ°ኳን՟ェዕኑ)le01.hmf (P) ۲٬ルト٬(B)	Ÿ−₩Ţ	<u> </u>	∿µ,∪7°(<u>H</u>)
■ご開発マネージャ fo フ [*] ロシ*ェクトマスタ(M) 新規作成(N)	r COBOL2002 - samp ファイル(E) ንግንንታንት	ble01.hmf (例 ビルド(風)	ッール(T) 留 脚	ንለንዮማ 🕨	(<u>H</u>)
図書記であった。 図書記では、ので、ので、ので、「「「「「「「」」」、 図目にのので、ので、ので、ので、ので、ので、ので、ので、ので、ので、ので、ので、ので、	r COBOL2002 - samp ファイル(<u>E</u>) ጋ°ኪን°ェクト	D <mark>le01.hmf</mark> (P) ビルド(B) Ctrl+O	ッ-ル① - 昭 証	ეიეჩულ ლაფა ჯ	^\L7°(<u>H</u>)
区間発マネージャ fo フプロジェクトマスタ(M) 新規作成(N) 閉(Q) 閉じる(Q)	r COBOL2002 - samp ファイル(<u>F</u>) ንግንንታንኑ	Die01.hmf (P) ביווגליש Ctrl+0	γ-μ(Ω (∰ ii k	ণে/১শিউ 🔣 🎕 🎜	^ルフ°(<u>H</u>)
■ご開発マネージャ fo フ [*] ロシェクトマスタ(M) 新規作成(N) 開く(Q) 閉じる(C) 保存(S)	r COBOL2002 - samp ファイル(E) ጋ°ロジェクト	ole01.hmf (P) ביאגרי(B) Ctrl+0 Ctrl+S	ッール(T) (学) 証	ንለንՒማ 😡 🗽 🎕 🎜	^,µ,⁊°(<u>H</u>)] 🍰 🏝
 ■経行ネージャ fo フロジェクトマスタ(M) 新規作成(N) 開く(Q) 閉じる(Q) (保存(S) フロハディ(P) 開発され、シューク	r COBOL2002 - samp ファイル(E) フロジェクト	ole01.hmf (P) ביאוּאַי(B) Ctrl+O Ctrl+S	ッール(1) (空 証)	ሳለሥን 🕨 🗽 🍭 🎗	 ^↓L7*(<u>H</u>) ↓ ▲ 🏂

5. COBOLソースプログラムの編集

開発マネージャで、環境を作りました。この後は実際にCOBOLソースプロ グラムをコーディングしていきます。

[手順1] 開発マネージャに表示されている、ソースファイルフォルダ配下にあ る.cblファイルをダブルクリックしてください。すると、自動的に COBOL2002の専用エディタが起動されます。

w 開発マネージャ for COBOL2002 - sample01.hmf	
7° $\gamma + \sqrt{C}$	
🛐 sample01.hmf	
sample01	
■ Feldan col	
COBOLエディタのウインドウの大き	さは
右上の 🔲 ボタン等で調整してくださ	い。
S [*] COBOLIディタ for COBOL2002 - C:¥temp¥¥reidai1.cbl ファイル(F) 編集(E) 検索(S) 構文(Y) オフ [*] ション(O) ツール(T) ウィント [*] ウ(W) ヘルフ [*] (H)	
Image: Second seco	
Image: Image	
Image: Image	
Image: Image	
Image: Image	
Image: Image	
Image: Set and	
Image: Set and Set an	19

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

[手順2] エディタを使用してCOBOLソースを編集(作成)します。練習用に以下の ようなプログラムをコーディングしてみてください。





COBOL専用エディタは、予約語・定数等の色分け表示、キーワード補完、構文テンプレート、構文チェック等のCOBOLの文法に対応した各種機能を用意しています。

[手順3] コーディングが終了したら、終了ボタンを押してエディタを終了してく ださい。すると、保存するかどうかの応答が返ってきますので、「はい」 を選択して保存してください。 ______



6. コンパイル(実行ファイルの生成)

いよいよコンパイルです。コーディングしたプログラムにエラー(フラグ)がなかったら、そのままリンケージ(結合・編集)が行われ、実行ファイルが生成されます。エラーがある場合は、エラーメッセージが出力されます。

[手順1] 開発マネージャの上の方にあるビルドボタン()をクリックします。



[用語解説] ビルド コンパイルとリンケージを一括して行うことを、「ビルド」といいます。

[手順2] ビルドが終了したら、メッセージウインドウを閉じてください。これ で、コンパイルは終わりです。メッセージウインドウに「KCCCXXXX」の エラーメッセージが出力されたときは、手順3以降を参照してくだ さい。

	右上 🗾 ボタンをクリック
/ENTRY:WinMainCRTStartup	
"/OUT:C:¥temp¥sample01¥reidai1¥work¥reidai1.exe"	
/NODEFAULTLIB	
″reidai1.OBJ″	
″reidai1.res″	
cbl2klg.lib	
cbl2k_32.lib	
msvcrt.lib	
kernel32.lib	
cbldcsm2k.lib	
■ ライブラリ C:¥temp¥sample01¥reidai1¥work¥reidai1.lib と	≤オブジョ <mark>ー</mark>
=== reidai1 のビルが処理が終了しました。===	
ν 7 %	NUM ///

[手順3] エラーがなくビルドが終了していたら、次の章の「実行」に進んでください。ここからは、コンパイルエラーが出たときのソース修正の方法を説明します。 メッセージウインドウ中に表示されているエラーメッセージをダブルクリックしてください。すると、エディタが自動的に起動されます。



エラーメッセージが見にくい場合は、スクロールバーを使ってスクロールするか、メッセージウインドウの大きさを調整することで、見やすいようにしてください。

[手順4] エディタが自動起動し、エラーに該当する行の先頭にカーソルが位置 付きます。先のエラーメッセージの内容と照らし合わせて、エラーを 修正してください。

国メッセージウィントウ				
=== reidai1 のビルが処理を開始します	₫. ===			•
C:¥temp¥sampleU1¥reidai1¥reidai1.cb				
"C:¥temp¥sample01¥reidai1¥reidai1.C	:BL", line 26: KC	CC3015C-	s ‴үүүмм	DD″は未定義です。 🗕
=== reidai1 のビ゙ルド処理でエラーが発生	ミしました。===	:		•
COBOLIデタ for COBOL2002 - [C:¥temp¥ ○ ファイル(E) 編集(E) 検索(S) 構文(Y)	¥reidai1.0BL] オフ≫ョン@) ッール(①		. <u> </u>	
<u> ヘルフ°(H)</u>			. <u>8 ×</u>	
🛛 🗅 🖨 🔚 🎒 👗 🖻 🕄 🝊 😗	<u>₩</u> 0	•	6† 6±	
		🕾 🄁		
				[]
				エラーメッセージの内 容と昭らし合わせて
	-+3++++++++++++++++++++++++++++++++++++		<u> </u>	プログラムを修正する。
002500 家/項规理 SECTION. 002600 ACCEPT YYYMMDD	FROM DATE.			この例では、スペルミ スであった。
				誤:YYYMMDD
			_	正:YYMMDD
003000 THEN			-	
לד ^י ז	行: 26 力	うちに 8 月	博入 //	
エラーの該当する行にカーソルが位置付きま	ミす。			

[手順5] エラーの修正が終ったら、エディタとメッセージウインドウを閉じて 手順1に戻り、コンパイルからやり直してください。 なお、エラーが複数ある場合は、メッセージウインドウ上のエラーメ ッセージを次々にダブルクリックすれば、エディタの該当位置に位置 付きます。また、一つのエラーのために複数のエラーが派生すること や、一つのエラーに隠れて他のエラーが検知できない場合もあります ので、ご注意ください。

7. 実行

次は実行です。今回の例題では特に入出力ファイルを用いていないので、簡単に 実行できます。入出力ファイルがある場合は、ファイルの割り当てを行う必要が あります。ファイルの割り当てについての説明は、「後述の3.関連資料」の「(a) ファイルの入出力処理」を参照してください。

[手順1] 開発マネージャの上の方にある実行ボタン(♪)をクリックします。 このボタンを「実行ボタン」と呼びます。実行ボタンをクリックすると、 実行確認画面が出ます。この画面の「OK」ボタンをクリックしてくだ さい。

		このボタンにマ けると「実行」	ウスポインタを位置づ と吹き出しが出ます。
■ご開発マネージャ for COBOL2002 - sample01.hm プロジェクトマスタ(M) ファイル(E) プロジェクト(P) ビ	f ルド個 ツール(T) ウ	W 117°H	
😳 🛎 🖬 🖉 ங 🚳 🖗	11 119 🕄 118 🕅	ه 🏖 🛛 🕹	
🚰 sample01.hmf			
in the second s	実行		× *
■ reidai1.cbl	実行可能ファイル名()	V: le01¥reidai1¥work	¥reidai1.exe 💌 参照(<u>B</u>)
1 -2.	作業フォルダ(<u>F</u>):		
V7 1	引数(<u>A</u>):		
			Kキャンセル
[CONSOLE画面]			
🔜 CONSOLE - C:¥temp¥sample01¥reidai1¥w	ork¥reidai1.exe		
**********Not September!!	****		今回の例題のDISPLAY文の 実行結果は、CONSOLE画面 に表示されます。
			この画面を終了させるに は、 <u>×</u> ボタンをクリック
実行が終了しました。			しより。

[ワンポイントアドバイス]

CONSOLE画面を閉じないうちは、実行ファイルは起動されたままの状態になっています。 このままにしておくと、再度コンパイルしたときエラーになります。実行結果を確認した ら、必ずCONSOLE画面を閉じるようにしましょう。

8. プロジェクトの追加

プログラムの作成から実行までの操作を一通り説明しました。ここでは、新た にプロジェクトを登録する方法と注意事項について説明します。

最初にも述べましたが、一つのプロジェクトマスタファイルの中に複数のプロ ジェクトを登録できます。もちろん、プロジェクト毎にプロジェクトマスタフ ァイルを作成してもかまいません。この場合は、これまで説明した手順にした がって作業してください。

プロジェクトを登録するときの二つの形態を以下の図に示します。

<二つの形態>

① プロジェクトマスタファイル1 ― プロジェクト1

プロジェクトマスタファイル2 ―― プロジェクト2

② プロジェクトマスタファイル1 → プロジェクト3

それでは、既にプロジェクトを作成したプロジェクトマスタファイルに新たな プロジェクトを追加する手順を示します。

[手順1] 新たなプロジェクト「reidai2」を追加します。開発マネージャの画面 から「プロジェクト(P)」-「プロジェクトの作成」の順にクリックします。

輕開発マネージ☆ for COBOL2002 - sample01.hmf	
7° $y = y = y = y = y = y = y = y = y = y $	
🐟 🛩 🖬 ૐ 🖼 🗞 🗞	
」コンゴン	
🛐 sample01.hmf 📃 🗆 🗙	
Sample01 「「」「reidail - reidail.exe 「」」」 「」」 「」」 「」」」 「」 「	<u>.</u>
ντ ² ή NUM	



jiwg開発マネージャ for COBOL200	02 - sample01.hmf	
プロジェクトマスタ(<u>M</u>) ファイル(<u>F</u>)	プロジェクト(Ⴒ) ビルド(B) ツール(① ウィンドウ(W) ヘルプ(Ή)
	フロジェクトの作成(A) フロジェクト名の変更(N)	هه
回 品 ?	プロジュウトのエウスポート⊗ プロジュウトのインポート@	1
sample01.hmf	作業中7泊ジェクトの設定(W) 7泊ジェクトの削除(E) 7泊ジェクトの関係設定(E)	
□ □ <u>-</u> <u>7</u> -27714	ソースファイルの追加(E)	
└────────────────────────────────────	フロジェクトの設定(<u>S</u>) 依存ファイルの設定(<u>D</u>) 依存ファイルの仮登録(<u>T</u>)	
	<u> </u>	
<u> </u>	ን [°] ኳパ [°] ティ(<u>P</u>)	
プロジェクトマスタにプロジェクトを作成	ag N	UM

[手順2]続いて、「プロジェクトの作成」画面が表示されます。

これは、「4. プロジェクトの作成」の手順1の画面と同じです。 もうおわかりですね。ここからは、これまで説明したプロジェクトの 作成手順にしたがってください。おさらいの意味で画面の遷移を以下 に示します。

ጋግንን ታንትወለፑ成	×
7泊ジェクト名(N):	
」 フペロジ [・] ェクトフォルダヾ(<u>P</u>):	
C:¥TEMP¥sample01 作業コッルないか	参照(<u>R</u>)
work	参照(E)
□ プロジェクトフォルタと作業フォ	オレタを同一にする。♡
0	K キャンセル
	プロジェクト名を入力する。
ንግንንድንትወ/ፑ成	×
フ [®] ロジェクト・名(<u>N</u>): reidai2	
, フ [°] ロジ [*] ェクトフォルタ ^{*(} <u>P</u>):	
C:#TEMP#sampleUT#reidal 作業フォルダ(W):	□2
work	参照(E)
□ プロジェクトフォルダと作業フォ	ォルタを同一にする。①
0	
	「Enter」キーを押す。
開発マネージゃ for COBOL2002	×
C:¥TEMP¥sample01¥reid	dai2 は存在しません。作成しますか?
OK	キャンセル
	「Enter」キーを押す。
開発マネージゃ for COBOL2002	×
C:¥TEMP¥sample01¥reidai	2¥work は存在しません。作成しますか?
(OK	キャンセル
	「Enter」キーを押す。

		-	
)ジェウトの追加:COBOLプロジェクト			3
最終生成物の種類を選択してください。			
○ 実行形式プログラム(E)			
○ ダイナミックリンクライブラリ(<u>D</u>) ○ 挿進ライブラリ(1)			
	< 戻る(8)	次へ(N) >	キャンセル
		[Enter]	キーを押す。
バェクトの追加:COBOLプロジェクト			2
プロジェクトの種類を選択してください。			
System指定メインプログラム V3指定メインプログラム			<u> </u>
メインプログラムなし OLEアウトオブプロセスサーバプログラム	4		
GUIプログラム			
त			▼ ▶
	< 戻る(B)	次へ(№) >	キャンセル

ንግንንድንኮለንነይከ加: COBOLንግንንድንኮ	×
出力ファイル名(EXE)を入れてください。(Q)	
reidai2 参照(<u>R</u>)	
「「「Inter」キーを押す。」	
〈戻る(B) 次へ(N) > キャンセル	
プロジェクトの追加:COBOLプロジェクト	×
メインファイル名を入れてください。(M)	
reidai2 参照(R)	
「reidai2」と入力し、「Enter」キーを押す。	
く戻る(B) 完了 キャンセル	_

Г









逐開発マネーツや for COBOL2002 - sample01	
プロジェクトマスタ(M) ファイル(E) プロジェクト(P)	ビルド(B) ツール(T) ウィンドウ(₩) ヘルプ(H)
	11 110 🕾 111 11X 🔍 🛠 🎒 🏠
国 かセージウィントウ	
🞯 sample01.hmf	
🔁 sample01	
□ reidai1 - reidai1.exe □ □ <u>1 - 27ァイル</u> □ □ <u>1 - 27ァイル</u> □ □ reidai1.cbl □ □ 依存ファイル □ □ reidai2.exe	
1 -8.	
V77	
	ブロジェクトの追加が完了しました。

続いて、複数プロジェクト環境で作業をするときに必要な「作業対象プロジェクトの選択」と「プロジェクトの削除」について説明します。

作業対象プロジェクトの選択

プロジェクトが複数ある場合、作業を開始するときに、どのプロジェクトの作 業をするかを明示的に指定する必要があります。 開発マネージャの画面で、一つだけ太字で表示されているプロジェクトがあり ます。これが、現在作業中のプロジェクトです。



プロジェクト「reidai2」の作業をしたい場合、「reidai2」をクリックします。 次に右クリックして表示されるプルダウンメニューの中から「作業中プロジェクト の設定(W)」を選択します。

■ 型開発マネージャ for COBOL200	02 - sample01.hmf	
7119191778(M) 77114(E)		
)
国がセージウイントウ	×	2
Sample01.hmf Sample01 Feidail - reida Feidail - reida Feidail - reidail Feidail.cb Feidail.cb K存ファイル Feidai2 - reidai2.e Feidai2 - reidai2.e fr 業中プロジェクトの設定を行う	il.exe 1 右クリック 選択中のブロジェクトのビレレド(E) <u>作業中プロジェクトの設定(W)</u> ソースファイルの追加(E) ビ゙レレド(B) プロンジェクトの設定(D) 実行(公 プロンジェクト名の変更(<u>い</u>) プロノヘティ(P)	×
壓開発マネーシシャ for COBOL2002	2 - sample01.hmf	
プロジェウトマスタ(<u>M</u>) ファイル(<u>F</u>) フ	1 T> 1 T>T>T>T>T>T>T>T>T>T>T>T>T>T>T>T>T>T>T>	
🔩 🚔 🖬 🦉 🐚 🚯 I	2 H. H. C. H. K. Q. X 42 ha	
「国 かセージウルトウ 「図 sample01.hmf		<u>1</u>
● Sampleon ● ● Preidai1 - reidai1.ext ● ● ● ソースファイル ■ reidai1.cbl ● ● ● 依存ファイル ● ● ● reidai2 - reidai2	e 「reidai2」が 太字になる。	
b7°₁	NUM	

Reidai2に対して、ソース編集やビルド,実行をしてみてください。 なお、ソースの編集は、該当ソースファイルをダブルクリックして編集するた め、プロジェクトを選択しなくても編集作業が可能です。しかし、ビルド等を するときには、作業中のプロジェクトを設定しておかなければなりません。

プロジェクトの削除

プロジェクトを削除したい場合は、次のように操作します。 削除したいプロジェクトをマウスで選択(クリック)し、「Delete」キーを押しま す。

開発マネージ≽ for COBOL2002 - sample01.hmf	_ 🗆 🗵
7° 7°	
🐟 🛩 🖬 ૐ № 🚳 № 📗 🔛 🗠 🛣 🔜 🛠 🍊 🎥	
国 かセージウイントウ	
🚰 sample01.hmf 📃 🗆 🔀	1
Sample01 → 「reidail - reidail.exe → 「ソースファイル → 「reidail.cbl → (依存ファイル ★ 「reidai2 - reidai2.exe ↓ 「Delete」キーを押す	
לא NUM	11.

9. 終わりに

以上で、プロジェクトの作成、及びプログラムの作成から、コンパイル、エラー 修正、実行までの一連の操作の説明は終わりです。

いろいろなプログラムを試して見てください。

テストデバッグ機能をお使いになりたい場合は後述の「2. テストデバッガの使用方法」をご覧ください。また、順ファイル等のファイルを使用するプログラムの実行については、後述の「3. 関連資料」の「(a)ファイルの入出力処理」を、 エディタの細かい設定方法については後述の「3. 関連資料」の「(e)エディタ設 定方法」をご覧ください。

最後に、reidai1のプログラムのコンパイルによって生成されたファイルを示し ますので、エクスプローラーで参照してみてください。コンパイラの生成物は、 「reidai1」フォルダの下の「work」フォルダに格納されています。

Reidai1.exe:実行可能ファイルです。「7.実行」で実行したのはこのファ イルです。

Reidai1. lst: コンパイルリストです。

Reidai1.cbp:デバッグ情報を格納したファイルです。

その他:各種管理情報を持つファイルです。

🔍 work				
ファイル(E) 編集(E) 表示(V) お気に入り((A) ツール(T) ヘルプ(H)			
◆ 戻る ▼ → ▼ 🖻 🛛 🖗検索 隆 フォルダ	3 1 to X 10			
アドレス(D) 🧰 work				▼ 🔗移動
フォルダ ×	名前 △	サイズ	種類	更新日時
	🗐 reidai1.cbo	1 KB	CBO ファイル	2005/06/27 18:44
🗄 🛅 Program Files 📃	🗐 reidai1.cbp	10 KB	CBP ファイル	2005/06/27 18:44
- D SOTEN	🔣 reidai1.exe	9 KB	アプリケーション	2005/06/27 18:44
	🖬 reidai1.exp	1 KB	EXP ファイル	2005/06/27 18:44
📃 🗄 🛅 GUI	💽 🖬 reidai1.lib	2 KB	LIB ファイル	2005/06/27 18:44
🖃 🛅 sample01	🗐 reidai1.lst	6 KB	LST ファイル	2005/06/27 18:44
🖻 💼 reidai1	🛛 🖬 reidai1.obj	6 KB	OBJ ファイル	2005/06/27 18:44
	📄 reidai1.rc	1 KB	RC ファイル	2005/06/27 18:44
⊕ 🛅 reidai2	🛛 🗃 reidai1.res	1 KB	RES ファイル	2005/06/27 18:44
🔄 🛄 WUTemp				
9 個のオブジェクトを選択(空きディスク領域: 4.72 0	iB)	33.3	KB 📃 🖓	イロンピュータ //.



一目次一

- 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。

開発マネージャ for COBOL200)2 - samp 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(P) ビルド(B) ツール(T)	ሳ/ኦኻ(₩)^ルプ(<u>H</u>)	
	プロジェクトの作成(<u>A</u>) プロジェクト名の変更(<u>N</u>)	🗽 🍳 🛠 🛛 🍊	
G sampleULhmr SampleU1 ⊡…☐ reidai1 - reidai1.€ ⊡…∏ У-2771№	(作業中プロジェクトの設定(\y) プロジェクトの削除(E) プロジェクトの関係設定(R)		<u>×u-</u>
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□	ソースファイルの追加(E) フ [^] ロジェクトの設定(<u>S</u>)	<2	
国メッセージウィントウ	依存ファイルの設定(<u>D</u>) 依存ファイルの仮登録(<u>T</u>)		
	カパレージ(Q)		
	ንኺኯ ^ۥ ፇ፟፞፞፞፞ <u>ጘ</u> ィ(<u>P</u>)		
プロジェクトの設定を行う			



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泊ジェ外設定 7泊ジェ外一覧(P): ● sample01 ● @ reidai1 - reidai1.exe	X Y コジェクト 製品連携 実行 最適化 アバック リンク 丸・ - DebugLine: デハック 「 たん・ - DebugInf: 異常終了時、エラー要約 情報を出力する - 「 - DebugCompati: 添字.指標名の反復回数の範囲外とつ い - DebugCompati: 添字.指標名の反復回数の範囲外とつ い - DebugData: データ例外検出 TDInf: テスト デハック 情報出力 CVInf: テスト カハ・レーシ 情報出力 CVInf: テストカハ・レーシ 情報出力 CVInf: テストカハ・レーシ 情報出力 DebugRange: 添字、指標名の反復回数範囲外チェック TestOmd: TDコマノト *格納ファイルの出力 - ト Full: 中断点情報。 シミュレーション 情報の TDコマノト *格納ファイル出力 - Sim: ジェュレーション 情報の TDコマノト *格納ファイル出力 - Sim Main: 主つ ログラムシミュレート機能
] 設定内容	
	▲ ▼

[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 「ロ」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 //



テストデハジック対象の設定		×
実行可能ファイル(<u>N</u>):		
C:¥temp¥sample01¥reidai1¥work¥reidai1.exe	-	参照(<u>B</u>)
DLLファイル(L):		
		削除(R)
		全削除())
' /定業つせしか/に)・		
		毎昭(の) 【
1		<u>- 2000 (0)</u>
3	OK	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: リック ニュー;	ストデバッガの制 して表示されるプ から「終了」をクリ	卸(F)」を? ルダウンン ックする。	7 *	G	2 クリック
🔮 テストデ・バッガ for COBOL	2002 - reidai	1.exe				
テストデバッガの制御(<u>F</u>) 編	譙/表示(E)	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①	ሳሪኑን 😡
テストデンバック対象の設定(<u>[</u>) Ctrl+O					
TDコマント [*] 格納ファイルの入力 初期化(N)	ታው <u></u> .) =X 💷 🕅			7	10 10 Part
🔷 テストデバッがの終了 😒						

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。

in COBOL2002 - reidai1.exe	
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
	と赤枠で囲ま れます。
	スクロールバー で調整して目
0000100 000100 IDENTIFICATION DIVISION.	的の場所に移動する
0000300 000300*	
10002800 00000000	
0003000 003000 THEN T	
■ モニタ IF命令をダブルクリックすると色が変わります。	
1	
<u>アープログラム起動前 NUM ///</u>	

[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。
😢 テストデバッガ 🔪 COBOL2002 - reidai1.exe
テストデバッガの制ム FL 編集/表示(E) プログラムの制御(C) 中断(B)
データの操作(D) < (T) ウィントや(W) ヘルフ°(H)
🛱 🎲 🐕 🕸 🕸 🎘 😄 💷 🐹 🔝 🔝
Set in the set
reidail
0000100 000100 IDENTIFICATION DIVISION
0000200 000200 PROGRAM-ID. reidail.
KCCC12051-1 COBOLフロクラムの実行を開始
KCCC12081-1 中断点です。2900 <reidait f="" td="" ユーザが与ぬ(p):<=""></reidait>
プログラム中断中
OK キャンセル
モニタ画面には、中断点で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)



 データ名「月」をダブル 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)。

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 [・] ハ [・] 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		b			
	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 [。] 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 ⁽ 」) ジャンプ ⁽ 」)			
*	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)			
加スなイズ(U)… IET3 KCCC1205T-I COBOLブロクラムの実行を開始しました。 KCCC1206T-I COBOLブ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。						

[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



[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/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 ^s 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





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。





🏪 カバレージ for COE	30L2002				_ 🗆 ×
カバレージの制御(工)	カバレーシ情報(C)	<u> かント情報(N</u>)	^ルフ°(<u>H</u>)		
לד ^י ז עדיי					

[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		//.



	カバレーショ情報の蓄積	×	
☆ カバレージ for COBOL2002 カバレージの制御(T) カバレーシ情報	実行可能ファイル(<u>N)</u> : ▼		
	DLLファイル(<u>D</u>):		
		削除(<u>P</u>)	
		全削除())	
	作業フォルダ(<u>F</u>):		
		参照(<u>S</u>)	
	ユーザ ^ッ パッラメタ(<u>P</u>):		
ן לאינ	ОК	キャンセル	



カバレージ。情報の蓄積	×	
このまたは、またしたが、おかいしージットの制御(T) カハッレージットは、 またしており、おいいしージットは、 またしており、おいいしージットは、 またしており、おいいしージットは、 またしており、またしており、 またしており、 またしてまり、 またしてまり、 またしてまり、 またしてまりいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいい		
実行可能ファイル	? ×	I I
ファイルの場所型: 🧲	🛿 work 🔽 🖛 🖻 💣 🎟 🕇	
reidai1.exe		
	ミ行可能ファイルをクリックします。	
」 ファイル名(N):		
ファイルの種類(1): [実	行可能ファイル(*.exe) ▼ キャンヤル	





🛗 カバレージ for COE	30L2002				- U ×
カバレージの制御(工)	カバレーショ情報(C)	がント情報(<u>N</u>)	∧μフ°(<u>H</u>)		
KCCC1200T-I KCCC1201T-I KCCC3405T-I KCCC0317T-I I	ユーザブログ ユーザブログ カバレージ情 カバレージ情	ラムの実行 ラムの実行 報の蓄積を 報の蓄積を	を開始しまし を除了しまし 行いました。 終了します。	,た。 ,た。 翻訳単位名(REIDAI1)) ファイ
<u>د</u>					×
レディ				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 [°] 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)	+++>++
□ Y-ス表示(S)	
□ 差分ソース表示(D)	
□ 未実行ソース表示(型)	
□ 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(型):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.c	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)			11
⇔戻る・⇒・ 🖻 🔍検索	B	フォルダ 🧭 🖺 🖻	$\times \mathfrak{o}$	*		
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	T					
		•				
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 情 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 ∞ 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 称 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。

			\sim				
🛗 カバレージ for COI	BOL2002	77					
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>)	¥7°(<u>H</u>)				
KCCC1200T-I KCCC1201T-I KCCC3405T-I KCCC0317T-I KCCC3427T-I	ューザプログ ユーザプログ 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 報リストを	を開始し を終いまし に 後 い て し し ま	ました。 ました。 た。翻訳単 す。 した。ファ	位名(REI イル(D:	DAI1) ¥sampli	ファイ e¥reid
•							►
カウント情報を表示する	5					NUM	11.

[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。

1989	かント情報の表示 実行可能ファイル(<u>N</u>):		1
mm ルバレージ カバレージの#	DLLファイル(<u>D</u>):		
KCCC120 KCCC340 KCCC031) ファイ
KUUU342	 作業フォレダ(<u>E</u>):	▲ #昭(S)	ple‡reid
▲ カウント情報を:	」 ユーザ [*] ハ [*] ラメタ(<u>P</u>):		
	出力ファイル名(Q):空白時は, 実行ファイル名.cnl	▼ 参照(E)	
	0	K ++>ZUL	



たのではまま たのではまま たので、 たので、 たので、 たので、 たので、 たので、 たので、 たので、	ノージ ジの新 19(実行 フ	<u>カウンド情報の表示</u> 実行可能ファイル DLLファイル(<u>D</u>): テ <mark>可能ファイル</mark> ァイルの場所型:	× N): ・ N): ・ ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	ーロメ メ ファイ ¥reid
	7 7	ァイル名(<u>N</u>): ァイルの種類(<u>T</u>):	 実行可能ファイル(*.exe) ▼ キャンセル OK キャンセル	



	カウント情報の表示	×
	実行可能ファイル(<u>N</u>):	
🎇 カバレージ f	D:¥sample¥reidai1¥work¥reidai1.exe	
カバレーシの制 KCCC1200	DLLファイル(<u>D</u>):	
KCCC1201)
KCCC3403 KCCC031 KCCC342		レーフアイ シー・ Le¥reid
	全的形余	
•	作業フォルダ(<u>F</u>):	
カウント情報を表		
	ユーサ [*] ハ [•] ラメタ(<u>P</u>):	
	出力ファイル名(②)空白時は,実行ファイル名.cnl	
)
	OK キャンセル	65

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>)	^/レプ(<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 ユーザフト 情報	ラムの実行 ラムの蓄積 報リスで マムの マムの マム の 実行 を を フム ト を 出 の ま に を を の ま で の ま の で る の ま の で る の ま の で る の ま の の ま の の ま の の ま の の ま の の ま の の ま の の こ の の ま の の こ の の こ の の こ の の ま の の こ の の こ の の ま の の こ の の こ ろ の こ ろ の の ま の の こ ろ の の ま の ろ の ろ の ろ の ろ の ま の う ろ の ろ の ろ の ろ ろ の ろ ろ の ろ ろ ろ ろ ろ の ろ ろ の ろ	を静終了した。 を終了しし始了しし。 を終うした。 を終してまししし	ました。 ました。 す。 した。ファイ しました。 た。ファイル	ル(D:¥sample¥re ・(D:¥sample¥reid	id lai
<u>・</u> レディ						•

[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(V) お気に入り(A) ツール(T) ヘルプ(H)					11		
今戻る・⇒・国 ◎検索 № 7ォルダ ③ 啓 啓 × ∞ 囲•							
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		
		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		🔛 <u>reidai1.exe</u>	9 KB	アプリケーション	/ 2004/04/01 19:10		
		📓 <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.obj</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		
□ □ □ ↓ □ ↓ □ ↓ □ ↓ □ ↓ □ ↓ □ ↓ □ ↓ □ ↓		neidai1.res	1 KB	RES ファイル	2004/04/01 19:10		
Address		reidai1.cnl	🔲 (1) 4 КВ	ONL ファイル	2004/04/06 18:53		
Agent	_						
種類: CNL ファイル サイズ: 3.06 KB 2010 2010 2010 2010 2010 2010 2010 201							

[カウント情報の表示例]

	* カウント情報 *
C0B0L2002	(X) 01-01 ****************************** 2004-04-06 18:53:00
	: 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:	REIDAI1
実行回数	
	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-1F.
	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。





- 1. はじめに
- 2. コーディング上の指定
- 3. 順ファイルの使用方法
- 4. プリンタへの出力方法
- 5. 索引ファイルの使用方法
- 6. 終わりに

1. はじめに

本説明書では、簡単なプログラム(ファイル等を使わないプログラム) の作成からコンパイル、実行までの使用方法は既に理解しているもの として、説明を進めます。

説明は、既にコンパイル&リンケージが終わり、実行ファイルが生成 されているところから始めます。フラグ消しなどの実行ファイルの生 成までは入門編他を参照ください。

通常、出力系で指定したファイルの実体が存在しない場合は、 COBOL2002が、自動的にファイルを割り当てて生成します。入力・更 新等で指定したファイルの実体がない場合は、実行時にエラーとなり ます。

なお、ここでは、ファイル等を用いたCOBOLプログラムを実行する上 での基本的な使い方を説明します。その他の詳細な使用方法について は、マニュアル「COBOL2002 操作ガイド」または、「COBOL2002 ユーザ ーズガイド」を参照ください。

2. コーディング上の指定

まず、ソースプログラム上のファイルの指定方法を説明します。 ファイルの割り当ては、プリンタに直接出力する場合などの特別な場 合を除いて、主に2つの方法があります。

①ソースプログラム中に、直接「ファイル実体名」を指定する方式
 ②外部装置名を指定して、プログラムを実行するときに実行時環境変数でファイル実体と結びつける方式

[ソースプログラム中に直接「ファイル実体名」を指定する方法]

ASSIGN句に、文字定数でファイル実体名(完全パス名)を指定します。パス名を省略すると、 実行可能ファイルがあるパスが仮定されます(実行可能ファイルと同じフォルダにファイル がある場合はパス名を省略できます)。



ファイル名を間違えるとプログラム実行時に、エラーとなります。(コンパイル時には チェックされないので、誤った名称であってもエラーにはなりません。)

<u>この指定方法では、3章以降で説明していくファイルの割り当て方法は必要ありません。</u>

[外部装置名を指定する方法]

ASSIGN句に「SYS010」などの外部装置名を指定します。



外部装置名は英数字の名称で「SYSXXX」などが一般的に使用されます。

<u>この指定方法では、3章以降で説明するファイルの割り当て方法によりファイル実体と結びつけます。</u>
3. 順ファイルの使用方法

COBOLプログラム中に、外部装置名「SYS010」を指定したものとして説 明します。また、コンパイル&リンケージが終わった状態から操作 方法を説明します。

[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 ウンメニューの中から「実行支援(L)」をクリックします。

☞開発マネーシャ for COBOL2002 - sample01	.hmf					<u>_ ×</u>
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>)	ビルド(B) ツール(T)	ሳብ≻ኮን₩	^/レフ°(<u>H</u>)			
🔹 🚔 🖶 🍊 ங 🍕 🖗 🗊 📰 📍	ビルド(B) クイックビルド(Q) コンパイル(C) リビルド(R)		Ctrl+B			
🚰 sample01.hmf		対する操作(P) •			
sample01 in reidai1 - reidai1.exe in reidai2 - reidai2.exe in ♪ ソースファイル in in reidai2.cbl in in reidai2.cbl in in reidai2.cbl	ビルドの中止(S) 依存関係更新(A) デバッガ(D) 実行(E)	~13 @J#TFV	Ctrl+E	2		
COBOL2002 実行支援を起動する	実行支援(1)				NUM	
またう支援 for COBOL2 ファイル(E) 設定(E) 実行 環境変数登録リスト(L) 「環境変数登録リスト(L) 「現象 少量デーー CBL_BATCH: プログ CBL_SYSERR: 実行時 CBLABNCODE: CBL CBLCOMCBR: 共通男 CBLEXVALUE: EXTE CBLLDLL: ダイナミック CBLLPROGDLL: DLL	002 - reidai2.CBR (R) ヘルフペH) マアイル マアイル マアイル マアイル ウンクトランパン マアイル ウンパン マアイル マアイン マアイン マアイン マアン マアン マアン マアン マアン マアン <th>「ユーザ設 ・ 指向 ・ 直面 セスを終了さ ファイル名を指 ードにする 指定する 脂値を指定する 脂値を指定する にする にする にする にする にする にする にする に</th> <th>定」タブで言 イベントログ 画面(XMAF せる 皆定する る</th> <th>-</th> <th></th> <th></th>	「ユーザ設 ・ 指向 ・ 直面 セスを終了さ ファイル名を指 ードにする 指定する 脂値を指定する 脂値を指定する にする にする にする にする にする にする にする に	定」タブで言 イベントログ 画面(XMAF せる 皆定する る	-		

[用語解説] COBOL2002実行支援

COBOL2002実行支援とは、プログラムの実行に必要な環境設定をするツールで、ファイルの割り当てやプリンタに対する印刷書式の設定などを行います。

[手順2] 実行支援画面の中の「ユーザ設定」タブをクリックし、変数名の入力エ リアに「CBL_外部装置名=」と半角で入力します。その後、参照ボタン をクリックします。すると、ファイルの参照画面が出ます。

≫実行支援 for COBOL2002 -	- reidai2.CBR				
ファイル(E) 設定(E) 実行(R)	∧μフ°(<u>H</u>)				
🗅 🛩 🖬 🕌 🖇	ファイルの参照				? ×
環境変数登録リスト(<u>L</u>)	ファイルの場所①:	🔁 reidai2	•	⇔ 🗈 📸 📰 •	
	◯ work i FILE02 eidai2.cbl				
	1) 外部装置名がSYSC)10の場合、「CBL_S	SYS010=」と入力	
一般 少量テータ 拡張機能 デバッグ	 ファイル名(<u>N</u>):	*.*		開公	0
変数名(N) CBL_SYS010=	ファイルの種類(工):	全てのファイル(*.*)		 キャン 	บ ม //
值(V) 个		1			
	印刷定義(P)				

[手順3] 必要であればフォルダの表示を調整して該当するファイルを探し、 ファイルのアイコンをダブルクリックします。

		該当フォルダに位置付ける	5.
ファイルの参照		2 7 7 ? ×	
ファイルの場所①:	🔁 reidai2	¥ ⇔ È r ⊞•	
work			
reidal2.cbi	該当するファイルのアイ コンをダブルクリック		
		·	
, ファイル名(N):	FILE02	開〈②〉	
ファイルの種類(工):	全てのファイル(*.*)	 キャンセル 	

[手順4] COBOL2002実行支援の画面に戻り、ファイルが完全パス指定で表示されます。ここで、登録ボタンをクリックください。

実行支援 for COBOL2002 - reidai2.CBR ファイル(E) 設定(E) 実行(B) ヘルフ [◦] (H)			ファ
🗅 😅 🖬 🖁 🖋			
環境変数登録リスト(L)			レレー
		削除(<u>D</u>)	完
			ポ
			名
 一般 少量データ ファイル 拡張機能 デバッグ オブジェク 	画面 画面(XMAP) フト指向 イベントログ	整列併合 ユーザ設定	示 さ
*** b 0 0			<i>n</i>
変数名(N) CBL_SYS010=			ま
(***			
1≝\⊻ C:¥temp¥sample01¥reidai2¥FILE02		登録(A)	
印刷定	:義(P) 参照(<u>B</u>)		登録ボタンを
			立 以 ホ ノ ン と と の の 、 、 、 、 、 、 、 、 、 、 、 、 、 、 、 、
			と垣境変数登 録リスト上に
■実行支援 for COBOL2002 - reidai2.CBR	(更新)		登録されます
ファイル(E) 設定(E) 実行(B) ヘルプ(H)			
🗅 🚅 🔒 🕌 🖋			
環境変数登録リスト(L)	1.500		
CBL_SYSUIU=C:#temp#sampleUI #reidai2#F1			
	環境変数を削除するとき	は、削除ボタンをク	フリックします。

[ワンポイントアドバイス]

- ①ファイル実体名が拡張子付きの場合は、拡張子も含めて指定してください。
 〈例〉CBL_SYS0101=C:¥temp¥samp1e¥reidai2¥F1LE02. dat
- ②本例題ではファイル実体名は完全パス名で指定しましたが、実行可能ファイル(.exe)と同じフォルダの中にファイルの実体があるときは、最後のファイル名だけですみます。 〈例〉CBL_SYS0101=FILE02
- ③新規に作成するファイル(OPEN OUTPUTのファイル)の場合、ファイル実体が存在しなく てもかまいません。この場合、参照ボタンで該当フォルダに位置付けて、ファイル名の 欄にファイル名を入力し、「開く(0)」ボタンをクリックします。実行可能ファイル(.exe) と同じフォルダに作成する場合は、ファイル名だけを入力します。

[手順5] COBOL2002実行支援を終了します。「保存しますか?」と聞いてくるの で、「はい(Y)」ボタンをクリックします。

	ダブルクリック [<u> </u>
「ファイル」「終了」の	実行支援 for COBOL2002 - reidai2.CBR (更新) ファイル(E) 設定(E) 実行(R) ヘルフ*(H) 新規作成(N) 開(@) 上書き保存(S) 名前を付けて(保存(A) 終了(※) 三つの終了方法があります どの方法で終了してもかます	
い順にクリックする	→般 少量データ ファイル 画面 画面(XMAF) 拡張機能 デバッグ オブジェクト指向 イベントログ 変数名(N)	 シ 整列併合 ユーザ設定
	印刷定義(2) 参照(8)	
	実行支援 for COBOL2002	×
	reidai2.CBRは更新されています。保存しますか (パパダ) パリンセル (パパズ)	?

[手順6] 開発マネージャに戻って、「実行」ボタンを押します。



4. プリンタへの出力方法

COBOLプログラムから印刷データをプリンタに出力する方法を説明します。

[印刷データを直接プリンタに出力する方法]

2章で説明したように、ソースプログラム中で指定することができます。 次に示すように、外部装置名の所に文字定数で「PRINTER」と指定します。



[印刷データを一旦フアイル上に出力する方法]

プリンタに直接出力するのでなく、一旦ディスク上に出力する方式も考えられます。この場合は、順ファイルに出力するのと同じやり方でできます。 順ファイルの出力内容が正しいことを確認したら、COBOLエディタ等を使って ディスク上のファイルを開いて印刷してください。

5. 索引ファイルの使用方法

索引ファイルの割り当ては、順ファイルと同様に次のどちらかで行い ます。

- ・ソースプログラム中のASSIGN句の外部装置名に文字定数でファイル 実体名を指定する。
- 外部装置名に「SYSXXX」を指定しておき、COBOL2002実行支援を起動して、「CBL_SYSXXX=ファイル実体名」を指定する。

順ファイルと異なるのは、次の点です。

- ・順ファイルは単独のファイルですが、索引ファイルは3種類のファ イルで構成されます。副キーは、最大98個まで指定可能です。
 ①キー定義ファイル(.kdf):キーとデータの対応を示す。
 ②キーファイル(主キー:.k01,副キー:.k02~.k99)
 ③データファイル(.drf)
- ・順ファイルでは拡張子を含めてファイル実体名を指定しましたが、
 索引ファイルの場合拡張子は指定しません。











[ワンポイントアドバイス]

順ファイルは任意の拡張子を付けることができます。また、拡張子がなくてもかまいません。拡張子を付ける場合は、ファイルの割り当て時に拡張子を含めて指定します。 索引ファイルは、拡張子が決められています。ファイルの割り当てをする際は、上記の例のように拡張子を除いて指定します。

6. 終わりに

ファイルを使用したプログラムの実行においては、ソースに直接フ ァイルの実体名を記述する方法と、実行支援でファイル実体と結び つける方法があることを説明しました。また、プリンタに直接出力 する方法も説明しました。

しかし、プログラムを実行するにあたっては、更に詳細な設定が必要な場合もあります。

詳細な設定方法については、マニュアル「COBOL2002 操作ガイド」または「COBOL2002 ユーザーズガイド」をご参照ください。

(b)テスト データの作成方法 - テストデータを容易に作成するために -

- 1. はじめに
- 2. COBOLエディタの起動方法
- 3. データの入力方法
- 4. COBOLエディタの終了
- 5. 改行コードの表示方法

1. はじめに

本説明書では、テキストファイルのデータの作成方法を説明します。 テキストファイルと順ファイルはほとんど同じ形式をしていますが、 改行コード付きのデータをテキストファイルと呼んでいます。改行コ ードまでを1レコードとして扱います。テキストファイルのデータは COBOLエディタやメモ帳で作成できます。

順ファイルとテキストファイルは、ソースプログラム上の定義が次の ように異なります。順ファイルの場合、「ORGANIZATION」句は省略でき ます。

順ファイル指定方法

SELECT	ファイル名	ASSIGN	Т0	SYS001	
	ORGANIZATIO	N IS S	SEQUE	NTIAL.	

テキストファイル指定方法

SELECT	ファイル名	ASSIGN	Τ0	SYS001
	ORGANIZATIO	N IS	LINE	SEQUENTIAL.

2. COBOLエディタの起動方法

[手順1]スタートボタンを押し(①)、「プログラム」の所にマウスポインタを移動します(②)。すると起動できるプログラムの一覧が出てきます。



[手順2] プログラムの一覧の中からCOBOL2002の所にマウスポインタを 移動します(①)。メニューから「COBOLエディタ」を選択します (②)。



[手順3] 起動されたCOBOLエディタ画面より「ファイル(F)」-「新規作成(N)」 の順に選択します(①-②)。すると「新規作成画面」が表示され ます(③)。

OLIディタ for COBOL200)2						_ [
νイル(E) 編集(E) 検索(S)	構文(Y) わりい	ル(型) ツール(⊒	ጋ ዕብሥትን 😡	^/レフ°(<u>H</u>)			
新規作成(N)	Ctrl+N Ctrl+O	5		■ 10 10	SB 38		
開いる(2) (2)	Otrl+F4						
上書き保存(<u>S</u>) 名前を付けて保存(<u>A</u>) すべて保存(<u>L</u>)	Ctrl+S	_	1			_	
ページレイアウトの設定(U) 印刷プレビュー(V) 印刷(<u>P</u>)	Ctrl+P						
プロパティ(民)							
COBOLIデ約の終了🗵	Alt+F4						
			_				
2							
	-		it:	J)カラム:			
) BOLIディ for COBOL2002 (F) 編(E) 検索(S) オ	基文(Y) オフ [⊗] /a ^γ /a	() y-l(T)	行: ウィントや (W)	<u>)</u> カラム: へルフ [*] (H)			
3 BOLIデ↓ for COBOL2002 (E) 編 (E) 検索(S) 材 (C) 復 文 (B)	携文(Y) オフジョン(21175 95) [美	() () () () () () () () () () () () () (「 市: ウィントウ ()))))))))))))))))))	カラム: へルフ [*] (日) ▼ ↓ ↓ ↓ ↓ ↓ ↓			
* BOLIディ for COBOL2002 (E) 編: (E) 検索(S) 相 译 【 / ● 湯 唱 [構文(⊻) わりうい 遣 仏 ℃]昏	0 7-1 T	パゴ: ウィントやう ₩0	אדּג: ∧ルז°(<u>H)</u> ערזינ <u>א</u> וי גטינטן	<u> </u>		
3 BOLIデ for COBOL2002 (E) 編 (E) 検索(S) 相 译 【 ● 人 ● 【 新規作成	#文(⊻) 打≫₃ン 己 (ひ�)量	0 7-4 (T) 3	済: ウ心トウ₩	אָדע: ∧וויס°(<u>H)</u> עוויס°(<u>H)</u>	<u>)</u>	X	!
* BOLIディ for COBOL2002 (E) 編 (E) 検索(S) 林 ご 「」」 参 」 み 唱 新規作成 愛 ファイルの種類:	構文☆ オパション ■ ひ �] 量	 ◎ ソール① ↓ 拡張子:	ነት י י י י י י י י י י י י י	カラム: ^↓レフ°(<u>H)</u> ▼ ±Ა キᲐ ₀		X	
* BOLIデ for COBOL2002 (E) 編 (E) 検索(S) 核 ご (E) 検索(S) 核 ご (E) 検索(S) 核 ご (E) 検索(S) 核 (E) 新規作成 で (COBOL)-ス(固定用)	構文(Y) オフ [°] ション/ 3 ひ ℃] 昏 (式正書法)(<u>C</u>)	 (2) ツール(T) (2) ブール(T) (3) 「 (4) ボ張子: (5) ▼ 	ןיד: לאיזי ₪ לאיזי ₪ לעליקע דייזיעי וו	カラム: ^ルフ°(<u>H</u>) ▼ ₺Ა १Ა -		×	
。 BOLIディ for COBOL2002 (E) 編 (E) 検索(S) (E) (E) (E) (E) (E) (E) (E) (E) (E) (E)	#文(Y) オフジョン 3 ひ �] 章 <u> ご書法)(C)</u> 試正書法)(E)	 (○) ツール(丁) (□) 	行: ウィントウ (W) し フ*ログラム テンフ*レー ロ	カラム: ∧ルフ°(Ⴞ) ▼ \$3 ₹3 -		×	
* BOLIデ for COBOL2002 (E) 編 E) 検索(S) 林 E) 人 (E)	構文(Y) オフジョン(3 ひ �] 章 「 ズ正書法)(E) 「ル(T)	 (○) ツール(丁) (□) 	行: ウィントウ (₩) フ*ロかラム テンフ*レ・ □ □	カラム: ^ルフ°(<u>H</u>) ▼ ±Ა ۴Ა /		×	
3 BOLIデ for COBOL2002 (E) 編 (E) 検索(S) 相 (E) 検索(S) 相 (E) 新規作成 (E) 77/ルの種類 (C) COBOLY-ス個定用 (C) 7キスト、その他の774	構文(Y) オフション 2 ひ ひ] 第 式正書注)(C) 試正書注)(F) い(T) OK	② ツール① ③ ツール① ③ 広張子: cbl ▼ cbf ▼	『行: ウィントッウ (₩) ワ・ロク・ラム テンフ・レ・ □	カラム: へルフ°(<u>H</u>) ・ ! -ト: 定値を登録(D)	<u></u>	×	
3 BOLIデ for COBOL2002 (E) 検索(S) 体 (E) (E) 検索(S) 体 (E) (E) (E) (E) (E) (E) (E) (E) (E) (E)	構文(Y) オフション(1) (J) 9) 【 【 【 二 書 法)(E) (ル(T) OK	② ツール① ③ ツール① ③ 広張子: cbl ▼ cbf ▼ 千ヤン	ドデ: ウィントゥ (₩) ワ・ロントゥ (₩) フ・ロク・ラム テンフ・レ・ 「」	カラム: へルフ°(<u>H</u>) ▼ ±3 ÷3 -ト: 定値を登録(<u>D</u>)	<u>√</u> √ √ // 7 '	×	

[手順4] ファイルの種類を「テキスト、その他のファイル(T)」にして(①) 「OK」ボタンを押す(②)と、新規にテキストファイルが表示され ます(③)。

■ COBOLIデሉ for COBOL2002 ファイル(E) 編集(E) 検索(S) 構文(Y) オフジョン(Q) ツール(T)	<u>- ロメ</u> ウィントウ (W) ヘルフ ^{*(} 出)
D 🛎 🗉 🧔 🐰 🖻 🖻 🕑 😗 🏷	I \$3 \$3 5€ 38
■ 新規作成	×
22 電 ファイルの種類: 拡張子: フ	ንግታንቻል ምንጋግሥት፦
C COBOLY-ス個定形式正書法)(C) cbl ▼	
С (1) ОК <u>+</u> еУ	セル 設定値を登録(D) ヘルフ°(H)
ν ⁷ [*] 1 (2)	行: 加ラム: //
	3
Second Life for COBOL2002 - [テキスト1] Read and the second secon	
	4+++++5+++++6+++++7+++++
لــــــــــــــــــــــــــــــــــــ	行: 1 万4: 1 挿入 //

3. データの入力方法

- [手順1] エディタ画面上のカラム目盛りの任意の位置をクリックすると 補助線が引かれます(①)。
 - ※ 補助線を消去するには、補助線のカラム目盛りの位置をダブル クリックします。
 - ※ 補助線は保存されません。

皆 COBOLエディタ for COBC	L2002 - [テキスト1]						<u> </u>
■ ファイル(E) 編集(E) :	検索(S) 構文(Y)	オフ°ション(<u>O</u>)	%-μD	<u>ሳሪ</u> ኦን∰	∧μフ°(<u>H</u>) ━		_ 8 ×
🗅 🚔 🔚 🎒 👗	B B J V	i] ≣\			- 1 9 49	ንይ %	
				🖏 🔁			
- CI 42							
	•••• 2 •••• ••	++ 3 ++++ +	4	+++++++ 5-		+6 ++++ +	7++
	>						
	1						
							•
							Þ
ν τ 'r			行	:1	<u> </u>	挿入	

[手順2] データをキーインします。各データの最後は改行します。

🛓 COBOLIデわ for COBOL2002 -	[テキスト1 *]						<u> </u>
□ ファイル(E) 編集(E) 検索(S)	構文化オプション@	ツール(①	<u> </u>	∿ルフ°(<u>H</u>)			<u>_ 8 ×</u>
🗅 🚅 🔚 🎒 👗 🖻 🗊	l 🗗 🕑 😼 🛛 🛼			- 79 ±9 9	6 🔏		
			👯 🏚				
G &							
	+++++	+++++4++	••• •	• • • • • • • • •	6++++	+++++7+	
02021301へ。ンタント	03600100						
0203130217977 020413037°¤~ 1	05500145 9		fの最後に	は改行コー	۰F		
02051304ネックレス	06890100						
020613052E17 02071306E977	10500050¶ 01920160¶						
02081307\$7E°>	05000080						
02091308*#F	p1000p99) #						
							-
							►
ν 7 °r		行	r: 8	カラム: 29	挿入	変更	11.

[手順3] 名前を付けてデータをセーブします。

- 「ファイル(F)」をクリックして、プルダウンメニューから「名前を付けて保存(A)」をクリックします(①)。
- すると、「名前を付けて保存画面」が表示されます(②)。 ・必要ならフォルダの表示を調整して、格納したいフォルダに位置づけ
- ・必要ならフォルダの表示を調発して、恰納したいフォルダに位直つけ ます(③)。
- ・フォルダに位置付いたらデータファイル名を指定します(④)。
- ・「保存(A)」ボタンをクリックします(⑤)。





4. COBOLエディタの終了

「ファイル(F)」をクリックして、プルダウンメニューから「COBOLエディタの終了(X)」をクリックします(①)。



5. 改行コードの表示方法

taba/25

- ・エディタのメニューバーの「オプション(0)」をクリックします(①)。
- ・プルダウンメニューから「カスタマイズ(C)」をクリックします(②)。
- ・「カスタマイズ画面」出たら、「表示」タブ中の「改行」をクリックします(③)。 ・「OK」ボタンをクリックします(④)。
 - $(\mathbf{1})$ 📴 COBOLIディタ for COBOL2002 - [C:¥temp¥DAT, - 🗆 × □ ファイル(E) 編集(E) 検索(S) 構文(Y) オフジョン(Q) ツール(T) ウィンドウ(W) ヘルフ°(H) _ 8 × ※

 国

 日

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10

 10 < 2 - 10 10 18 18 🗅 🗲 🔚 🎒 ツールバーの表示(S) **≣**+ ≣± ≣‡ ≣* 罷 61 62 02021301h°ンタ`ント 03600|100| 020313024ヤリング 02000/230| 020413037*ローチ 05500|145 020511304株ックレス 06890|100| 0206130524*7 10500050 ∎ Î Î COBOLIディタのカスタマイス 行: 8 カラム: 29 挿入



編集表示)色)画	
- ワインドワの設定 ▼ 選択マーシン(M)	▼ 縦ス加ールパー型
□ 行番号(L)	☑ 横スケロールパー(乙)
ルーラー(R)	☑ ウィンドウサイスを最大にする⊗
▶ トラッキングバー(①)	
┌編集文字の設定――	
□空白Ѡ: □	
□ \$7°@: [·	→ = ddī(P):
┌機能キャラクタの設定――	
 ● テキスト(T) 漢 	字開始(<u>S</u>): ^{*K} 漢字終了(<u>D</u>): ^{*E}
○ ヴラフィッウ(<u>G</u>) 漢	字開始: 漢字終了: 巨
🔲 テフォルトで機能キャラ	がタを表示する(Y)
	4
	OK キャンセル ヘルフ°(H)

89

V

(c)印刷用紙の節約方法

- 実行結果の確認と印刷方法 -

- 1. はじめに
- 2. ソースプログラムの変更
- 3. COBOLエディタの起動と印刷方法

1. はじめに

実行結果は、通常直接プリンタに出力して確認しますが、この方法では用 紙の使用量が多くなることがあります。下記のように、一旦ファイルに出 カして、出力結果が正しいことを確認してからプリンタに出力する手順に すると、用紙を節約することができます。

・実行結果をファイルに出力します。

・COBOLエディタを使用して出力結果を確認してから印刷します。

2. ソースプログラムの変更

次に示すように任意のフォルダのファイル名を指定します。 (ファイル実体が存在しなくても、指定されたファイルが作成されます)

直接プリンタに出力するときの指定 SELECT ファイル名 ASSIGN TO 'PRINTER'.



ファイルに出力するときの指定 SELECT ファイル名 ASSIGN TO 'C:¥Temp¥data¥OUTFILE'.

3. COBOLエディタの起動と印刷方法

[手順1] スタートボタンを押し(①)、「プログラム((P)」の所にマウスポイ ンタを移動します(②)。

すると起動できるプログラムの一覧が表示されます。

	•••	プログラム(型)	•	 7ウセサリ COBOL2002	*
$(\underline{2})$		最近使ったファイル(<u>D</u>)	·		-
		設定(S)	•		
		検索©)	•		
	2	ヘルプ(出)			
	1	ファイル名を指定して実行(R)			
	2	k997571 のログオフ(L)			
1		シャットダウン(山)			
	1 1 1	<u>Þ-</u> F			

[手順2] プログラムの一覧の中から「COBOL2002」の所にマウスポインタを 移動します(①)。プルダウンメニューから「COBOLエディタ」を選 択します(②)。



[手順3] 出力ファイル(C:¥temp¥data¥0UTFILE)の表示 起動されたCOBOLエディタ画面より「開く(0)」を選択します(①)。 すると「ファイルを開く」画面が表示されます(②)。

🛓 COBOLIデ約 for COBOL200)2		
ファイル(E) 編集(E) 検索(S)	構文(Y) 打ジョン(Q)	୬−ル(፲) ሳィンドウ(₩) ^ルプ(Ⴞ)	
新規作成(N)	Ctrl+N	▼ †? i	5 SB 38
開いる(の)	Otrl+O Otrl+F4		
上書き保存(<u>G</u>) 名前を付けて(保存(<u>A</u>) すべて保存(<u>L</u>)	Ctrl+S		
ページレイアウトの設定(U) 印刷プレビュー(y) 印刷(<u>P</u>)	Ctrl+P		
プロノパティ(<u>B</u>)			
COBOLIデ物の終了🖄	Alt+F4		
ν 7 °r		行: カラム:	





[手順4]「ファイルを開く」画面から

 ①「ファイルの場所」で該当フォルダ(C:¥temp¥data)を指定します。
 ②「ファイルの種類」で「すべてのファイル(*.*)」を選択し、表示された ファイルの一覧から「OUTFILE」を選択します。

- ③「開く(0)」ボタンをクリックします。
- ④ファイル中に特殊文字が含まれていると、「特殊文字の削除] 画面が表示されます。
- ⑤「はい(Y)」をクリックすると、「OUTFILE」が表示されます。
- * 印刷ファイルは改行コード(特殊文字)が含まれるため「特殊文字の削除」 画面が表示されますが、ファイルの表示内容には影響ありません。



[手順5] 表示された「OUTFILE」の内容が正しい場合は印刷します。

①エディタのメニューバーの「ファイル(F)」をクリックし、プルダウンメニューの中から「印刷(0)」を選択すると「印刷」画面が表示されます。

②「印刷」画面の「OK」ボタンをクリックします。

💕 COBOLIデ約 for COBOL2002 - [C:	¥temp¥data¥OU	TFILE]		
□ ファイル(E) 編集(E) 検索(S) 様	<u>乾∽</u> わѷョン	(D) ツール(T) ウィントウ	(₩) ^/レフ°(<u>H</u>)	_ & ×
0 🛩 🖬 🎒 👗 🖻 💼	ຜ ຈ 🛛 🔊 🛛		- 1 7 ±2 ±2 ?B	SE .
		82	E I I	
G &			<u> </u>	
	++++++ 3++	+++++++4++++++++	++ 2 +++++++++6+	+++++7+++++
商品名 1.エアコン(L-100) 2.冷蔵庫 3.ポータブルテレビ	個 数 1 O 1 5 3 5	金 額(万円) 250 300 350		
合 計	60	900		- 1
				Þ
ν 7 ² 4		行: 1	力ラム: 1 月	ŧλ //.
			印刷	



(d)印刷書式の設定方法

ーはじめにー

プリンタへ出力するときの印刷書式の設定は、開発マネージャから「実行支援」ツールを起動して設定します。

[ワンポイントアドバイス]

プリンタ出力時に次のエラーが出力されることがあります。

KCCC3907R-W 文字列の出力がページの右端を超えています。

このエラーの対処としては、次の方法が考えられます。

〈ソースプログラムの修正〉

・プリンタファイルに対するファイル管理記述項に次の指定をする。
 ORGANIZATION IS LINE SEQUENTIAL
 この指定をすると、後部の空白を詰めて印刷します。各行の後部に空白

できることがあります。 この指定をするこ、後部の生日を詰めて印刷しより。 名目の後部に生日 があるときは、空白の分だけ出力データが短くなるので、エラーを回避

・出力レコード長を印刷できる範囲に変更する。

〈実行支援ツールの設定〉

- ・用紙の向きを横に変更する。用紙の向きが縦になっていると、1行に印 刷できる文字数が少ないためにエラーになりやすいといえます。
- ・余白/字間値/文字サイズを調整して1行に収まるようにする。

1. 印刷書式の設定

[手順1]「開発マネージャ」のメニューバーから「ビルド(B)」をクリックし、プル ダウンメニューの中から「実行支援(L)」を選択します。

■<開発フネージを for COBOL2002 - sample01	hund		
フ [*] ロジ [*] ェクトマスタ(<u>M</u>) ファイル(<u>F</u>) フ [*] ロジ [*] ェクト(<u>P</u>)	ビルド(1) ツール(1) ウィントウ(1)	ヘルフ°(<u>H</u>)	
🔹 🛋 🖬 🧭 ங 🔹 🖗 II 😄 ?	ビルト*(B) クイックビルト*(Q) コンパイル(G) リビルト*(<u>R</u>)	Ctrl+B	
🚰 sample01.hmf 🚰 sample01	すべてのプロジェクトに対する操作(圧) •	
i reidai1 - reidai1.exe i	ビルドの中止(S) 依存関係更新(A)		
■ reidai1.cbl	デバッガ(<u>D</u>)		
🖅 🗇 reidai2 - reidai2.exe	実行(E) 実行支援(L)	Ctrl+E	2
COBOL2002 実行支援を起動する			

[手順2]「実行支援」のメニューバーの「設定(E)」をクリックし、プルダウンメ ニューの中から「印刷書式(P)」-「デフォルト(D)」の順に選択します。

$\sqrt{1}$	
▶実行支援 fo BOL2002 - reidai1.CBR	
ファイル(E) 設定(E) 実行(R) ヘルプ(H)	
□ 日刷書式(2) → 773/04(2)	
π ² π ² χ ² μ	削除(D)
│ 拡張機能 │ デバッグ │ オブジェクト指向 │ イベントログ │	ユーザ設定
一般 少量データ ファイル 画面 画面(XMAP)	│ 整列併合 │
■ CBL BATCH:プログラム終了と同時にプロセスを終了させる	
□ CBLCOMCBR: 共通実行環境ファイル名を指定する	
□ CBLEXVALUE: EXTERNAL指定項目の初期値を指定する	
□ CBLLDLL:ダイナミックリンクするDLLの名称を指定する	
□ CBLLPROGDLL:DLL自動ロード機能を使用する	-

[手順3]余白/行間隔/フォント/文字サイズ等の設定をします。 用紙のサイズや印刷の向きは「プリンタの設定(P)」で設定します。

詳細設定	×
- プリンタ 名前: ¥¥AT21P¥PRINTER1 場所: LPT1:	ОК キ ャンセル
□ 通常使うブリンタに出力(D) 余白/行/文字 ┃書体 ┃	
余白(インチ) 上①: 0.5 左①: 0.5 下M): 0.207 右(B): 0.207	印刷行数(N): 自動 📑
行間隔①: 4 lpi (18 pt) ▼ 間隔②:	□ 半角文字と全角文字の間隔を調整する(C)
文字 7#ント名(E): 7#/レ{ゾ: 明朝 サイズ(S): 9 pt 9 pt ア揃え マ	-7 [®] Ut [*] ュー <u>AaBbYyZzあぁアァ亜宇</u> <u>AaBbYyZzあぁアァ亜宇</u> AaBbYyZzあぁアァ亜宇



[手順4] 「プリンタの設定」画面で、用紙サイズのリストボックスから該当する用 紙サイズを選択(①)します。印刷の向きは、「縦」、「横」のラジオボタ ンから選択(②)します。必要な情報を設定したら「OK」ボタンをクリッ ク(③)して終了します。

プリ	レタの設定			×	
	-プリンター				
	プリンタ名(<u>N</u>):	¥¥AT21P¥PRINTER1	•	プロパティ(<u>P</u>)	
	状態:	準備完了			
	種類	Canon LASER SHOT LBP-730			
	場所:	LPT1:			
	אַכאָב	Canon LBP-730			
	-用紙			の向き	
	サイズ(Z):	B4 (JIS)	.	r ⊙‱⊙ <	⊐②
	給紙方法(<u>S</u>):	A3 A4 A5 B4 (JIS)			
	ネットワーク(<u>W</u>)	B5 (JIS) Executive Legal	ок∠	キャンセル	
					98

[手順5]「詳細設定」画面に戻ったら、「OK」ボタンをクリックして終了します。 「実行支援」画面に戻ったら、メニューバーの「ファイル(F)」をクリックし、 プルダウンメニューの中から「終了(X)」を選択します。

詳細設定	×
フツンター 名前: ¥¥AT21P¥PRINTER1 場所: LPT1:	OK
□ 通常使うフツンタに出力(D) 余白/行/文字 書体 余白(インチ)	[7975の設定で]]
上①: 0.5 左①: 0.5 下(M): 0.207 右(B): 0.207	印刷行数(N): 自動
	□ 半角文字と全角文字の間隔を調整する(©) -7°レビュー A+ P+Yu7= 第 ★ マ = 第 字
明朝 サイス ⁽ S): 9 pt → 「揃え ▼	AaBb1y22ののファ亜ナ AaBbYyZzあぁアァ亜宇 AaBbYyZzあぁアァ亜宇

★援 for COBOL2002 - reidai1.CBR (更新)	_ 🗆 🗙
ファイル(E) 設定(E) 実行(R) ヘルプ(H)	
新規作成(<u>N</u>)	
開(@)	
上書き保存(S)	8/124/03
名前を付けて1米存(A)	用112年(世)
終了☆ (2)	
拡張機能 デバッグ オブジェクト指向 イベントログ	ユーザ設定
 一般 少量データ ファイル 画面 画面(XMAP) 	整列併合
■ CBL BATCH: プログラム終了と同時にプロセスを終了させる	
□ CBL SYSERR・実行時メッセージの出力失力マイルタを指定する	
□ CBLCOMCBR: 共通美行環境 Jアイル 名を指定する	
CBLEXVALUE: EXTERNAL指定項目の初期値を指定する	
□ CBLLDLL:ダイナミックリンクするDLLの名称を指定する	
□ CBLLDLL:ダイナミックリンクするDLLの名称を指定する □ CBLLPROGDLL:DLL自動ロード機能を使用する	
□ CBLLDLL:ダイナミックリンクするDLLの名称を指定する □ CBLLPROGDLL:DLL自動ロード機能を使用する	

[手順7] すると以下のダイアログが表示されます。必ず「はい」ボタンをクリックしてください。「はい」ボタンをクリックすると設定した内容が保存され、開発マネージャの画面に戻ります。





(e)エディタ設定方法

- COBOL専用エディタを使いこなすために -



- 1. はじめに
- 2. マーカーとシーケンスの指定
- 3. 表示のカスタマイズ
- 4. キーワード補完
- 5. 構文テンプレート
- 6. 構文チェック

1. はじめに

本説明書では、COBOL2002専用エディタの起動方法を理解しているものとして説明を進めます。

使用する例題プログラムは入門編で用いたプログラムです。

ここでは、COBOL言語でコーディングするときに特に便利な機能について説明します。更に詳細を知りたい場合は、マニュアル 「COBOL2002操作ガイド」を参照ください。



マーカーは、COBOLプログラムのコーディングの際に、インデンテーション を揃えるために使用します。

[手順1] エディタのメニューバーの「オプション(0)」をクリックし、プルダウンメ ニューの「カスタマイズ(C)」をクリックします。

COBOLIデタ for COBOL2002 - [Ci¥temp¥¥reidai1.
□ ファイル(E) 編集(E) 検索(S) 構文(Y) オブション(Q) ツール(D) ウィントウ(W) ヘリコペル)
□□ 🖙 🖬 😂 🛯 🕹 🔨 🔨ルバーの表示(S) 🕨 🔄 抗 🚯

IDENTIFICATION DIVISION.
TRUGRAMTID. reidail.
カフタフィブ画面で
を選択します。
編集 表示 色 画面フォント ファイルの種類 保存 予約語 COBOLソース
ウィントウの設定
✓ 固定形式正書法のマーカ(M) □ ルーラーのB領域識別子(B)
☑ 自由形式正書法のマーカ(E)
□ 一連番号の設定
初期値(S): 100 増分値(N): 100
☑ 保存時にオプションを適用する(B): 一連番号を空白に置換
■ 保存時に不当な一連番号を知らせる(型)
■ 見出し領域の設定
 ● 現在の日付に置換(D)
○ 指定された文字列と置換(T):
「行指定の背景色分け表示の設定
□ 行指定の背景色分け表示を使用する(Q)
開始キー(近): 終了キー(E):
OK キャンセル ヘルフ*(H)

[手順2]「COBOLソース」画面で必要な設定をします。 設定したら「OK」ボタンをクリックしてください。



3. 表示のカスタマイズ

表示を見易くするための設定です。

[手順1] エディタのメニューバーの「オプション(0)」をクリックし、プルダウ ンメニューの「カスタマイズ(C)」を選択します。「カスタマイズ」画面が出 たら、「色」または「画面フォント」タブをクリックします。



[手順2] 予約語(動詞やコメントや定数等)の色の指定を行います。



[ワンポイントアドバイス]

予約語が色分け表示されるので、コーディングの際にスペルミスをチェックすることが可能です。 見分けがつきやすい色を選ぶと一目でスペルミスがわかります。

[手順2] フォントの指定を行います。

「デフォルトのイメージが表示されます。



4. キーワード補完

本機能は語の綴りを忘れたとき、語の一部を指定してフルスペルに 補完する機能です。

[手順1] 例えばINSPECT文の「INS」までの綴りしか思い出せない場合、記述したい箇所に「INS」を入力し、「INS」の部分にカーソルを位置付けるかまたはリバース表示させます。

🖹 COBOLIन्दे⁄a for COBOL2002 - [C:¥temp¥¥re	idai1.cbl *]		
📑 ファイル(E) 編集(E) 検索(S) 構文(Y) オフ	ション(ロ) ツール	🛈 ሳለኦኮウ🛯	0
^ルフ°(<u>H</u>)			_ 8 ×
D 🖨 🖬 🎒 🌡 🖻 🖻 🝊 😗]	5		<u>-</u> ₹9 ±9
▶ ▶ 計 計 🗘 🕸		📇 I	
	 2	1 ++++	5++++
001700 PROCEDURE DIVISION.			
001800			
001900 Mein-Sec SECTION.			
002000 PERFORM 初期処理.			
002100 <u>PERFORM</u> 比較処理.			
00220011 PEREORM 山-5hn I用			
ปรั้ง	行: 22	カラム: 15	挿入

[手順2] カーソルを位置付けて、あるいはリバース表示させた箇所で右クリ ックし、「キーワード補完″INS″」を選択します。



[手順3] キーワード補完の画面中の「INSPECT」を選択し、「貼り付け」をクリック します。



COBULT7 1% for COBULZUUZ - [C:#temp##re	idal1.cb1 *j	
📑 ファイル(E) 編集(E) 検索(S) 構文(Y) オフ	≫ョン(_) ツール(_) ሳለ/ኑን(₩)
^ルフ°(<u>H</u>)		_ 8 ×
🗅 🚅 🖬 🎒 👗 🖻 🖻 🝊 😗]	5	₹7 ±7 ±7
]▶ क़ क़ ऻ 🗘 🕸		🔀 🔁 🗮 🎞
++++ + 1++++ + 2++++ + 3	+++++++++++++++++++++++++++++++++++++++	++++++++5+++++
001700 PROCEDURE DIVISION.		
001800		
001900 Mein-Sec SECTION.		
002000 PERFORM 初期処理.		
002100 PERFORM 比較処理.		
INSPECT		
		<u></u>
bディ	行: 22	カラム:19 挿入 //


本機能は文の書き方を知りたいとき、マニュアルを見なくても書き方を参照できる機能です。

[手順1] 例えばINSPECT文の書き方を知りたいとき、エディタ画面上の記述したい箇所にキーワード「INSPECT」を入力し、その一部にカーソルを位置付けるか、またはリバース表示させます。

皆 COBOLIデ水 for COBOL2002 - [C:¥temp¥¥re	idai1.cbl :	*]		>	<
📑 ファイル(E) 編集(E) 検索(S) 構文(Y) オフ	ション(0)	ツール①	<u>ሳ</u> ረንኑን (N)	
^μフ°(<u>H</u>)				_8/2	4
🛛 🗅 🚅 🖬 🎒 👗 🖻 🖻 🖉 😗	1			- 19 ↓	5
▶ 幹 計 計 ① 磁] 🛃	₽ [] I	Ĩ
++++++++ 1 +++++ 2 +++++ 3		+++++4+++	++++++	+5++++	
001700 PROCEDURE DIVISION.					
001800					
001900 Mein-Sec SECTION.					
002000 PERFORM 初期処理.					
002100 PERFORM 比較処理.					
INSPECT					
					1
	K= 00				
V7'Y	T: 22	ות	74: 19	挿人	11.

[手順2] カーソルを位置付けた箇所またはリバース表示させた箇所で右クリ ックし、「構文テンプレートの"INSPECT"」を選択します。

🛓 COBOLIデ水 for COBOL2002	- [C:¥temp¥¥reidai1.cbl *]	
📑 ファイル(E) 編集(E) 検索(S	〉 構文(Y) オプション(Q) ツール	⊕ ሳለ⊁ኻ₩
^/レフ [*] (<u>H</u>)		
0 🛩 🖬 🎒 👗 🖻	🛍 🝊 👽 🛼 📃	₹19 <u>±</u> 9
]▶ क क क 🛱 🕸		8. 🔁 🗖 🎞
+++++++++++++++++++++++++++++++++++++++	2 +++++++++ 3 ++++++++ ,	4++++ ++++5+++++
001700 PROCEDURE [DIVISION.	
001800		
001900 Mein-Sec SEC	CTION.	
002000 PERFORM	初期処理.	
002100 <u>PERFORM</u>	比較処理.	
INSPECT	រ– រ–ាម≣ាម/=)	
	νου 4χου <u>τ</u> ο ημ ^ο =(C)	
雄立ショットの差昭		
1年又77717-1700多頭	削除(D)	14/
-	(画:逆:5)ばの(化)(二)(か)	
	1点酸积极(1)16年(位) 目中1.領域の苦境(1)	
I	構文テンプレート(<u>B</u>) "INSPECT"	
	キーワート 補完 (E) "INSPECT"	

[手順3]「構文テンプレート」画面が表示されます。

ここで、例えば書き方1(計数)をクリックすると、プレビュー画面 に使用方法が表示されます。

構文デンフプレート		×
INSPECTO - 書き方1 (計数) - 書き方2 (置換) - 書き方3 (計数と置換) 書き方4 (変換) E INVOKE文 - MERGE文 - MULTIPLY文 - OPEN文 - PERFORM文 - RAISE文 E READ文 - READ文 - READ文	Bibl付け(P) El号削除貼り(付け(P) El号削除貼り(付け(P) キャンセル ペルフ*(H)	
検索する文字列(<u>N</u>) INSPECT	下方向② 上方向④	1

[手順4] プレビュー画面へ使用方法が表示され、「記号削除貼り付け」をクリ ックすると記号が削除されて該当する箇所へ貼り付けられます。

文テンフ℃ート 構文		
□- INSPECT文 □- 書き方2(置換) □- 書き方2(置換) □- 書き方3(計数と置換) □- 書き方4(変換) □- INVOKE文 □- MERGE文 □- MOVE文 □- MULTIPLY文 □- OPEN文 □- PERFORM文 □- RAISE文 □- RAISE文 □- READ文 BECCETVE+	COBOL2002 VOS3 COBOL85	<u>貼り付け(P)</u> 記号前J除貼り付け(E) キャンセル ヘルフ ^o (H)
INSPECT 一意名1 TALLYING TALLYING指定 TALLYING指定: {一意名2 FOR {{CHARACTERS [AFTER-BEFORE指定]} {ALL {{一意名3 定数1 } [AFTER-BEFORE指定]}····} {LEADING {{一意名3 定数1 } [AFTER-BEFORE指定]}····} }···} AFTER-BEFORE指定:		
検索する文字列(N) INSPECT		下方向② 上方向(B)
•		11

「記号削除貼り付け」の結果を以下に示します。「張り付け」をクリックすると記号も 一緒に貼り付けられてしまいます。なお、手順4の画面に表示されている構文を直 接書き換えることもできます。書き換えが完了したら、「張り付け」をクリックします。



6. 構文チェック

構文チェック機能は、エディタ画面でプログラム編集中に、文法チ ェックを行う機能です。

[手順1] エディタのメニューバーの「構文(Y)」をクリックし、プルダウンメ ニューの「構文チェック(S)」を選択します。

🛃 COBOLIデ約 for COBOL2002 - [C	:¥temp¥¥reidai1.cbl *]		_	
📑 ファイル(E) 編集(E) 検索(S) 💈	構文(Y) オプション(Q) ツーノ	⊮ጠ ታሪን⊦ጛ₩ ^	μフ°(H)	. 8 ×
	構文チェック(S) F 構文チェックを中止(C) (C 構文チェックオフジョン(Q)	5 Dtrl+F5	1 Ⅲ 1 11 10 10 10 10 10 10 10 10 10 10 10 10	
002300 STOP RUN	構文テンプ レート(B) C キーワート [*] 補完(E) C ユーザキーワート [*] 補完(U) C	Dtrl+B Dtrl+E Dtrl+K	•••	+++
002500 初期処理 SECT 002600 ACCEPT YY 002700*	ION. /MMDD FROM DATE			
002800 比較処理 SECT: 002900 IF 月 = 003000 THEN	ION. 9			
003100 MOVE 'Se 003200 ELSE 003300 MOVE 'N	eptember!!' TO I	DATA2 TO DATA2		
003400 END-IF. 003500*		10 00102		
003600 出力処理 SECT 003700 DISPLAY D/ 003800 003900	ION. ATAO.			-
【▲】 COBOL ソースの構文チェック	行: 17) መንገር መንገር መንገር መንገር መንገር መንገር መንገር መንገር	挿入 変更	

構文チェック結果がCOBOLエディタ画面の下に表示されます。 エラーがある場合は、通常のコンパイルと同様にエラーメッセージが表示 され、エラーメッセージ部分をダブルクリックすると、エラーの箇所にカ ーソルが位置付きます。

📴 COBOLIディタ for COBOL2002 - 構文チェック結果	X
ファイル(E) 編集(E) 検索(S) 構文(Y) オプション(Q) ツール(T) ウィントウ(W) ヘルプ(H)	
D 🛸 🖬 🚭 X 🖻 📾 🖉 O 🕫 👀 💽 💽 ±0 ∓0 № 38	
₩₩₩₩₩₩₩₩	
C¥temp¥sample01¥reidai1¥reidai1.cbl	×
	G
II 002600 ACCEPT TTIMMDD FROM DATE.	
002800 比較如理 SECTION.	
002900 IF 月 = 9	
UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU	
003200 LESE Not September !!' TO DATA2	-1
003400 END-IF.	
	Ľ.
	X
· C:¥temp¥sample01¥reidai1¥reidai1.cbl'を構立チェック中	
"C:¥temp¥sample01¥reidai1¥reidai1.CBL", line 26: KCCC3015C-S "	-
YYYMMDD″は未定義です。	-1
1個、エラーが発生しました。	•
レディ 行: 4 加ラム: 1 挿入 書禁	

(f)コンパイルリストの入手方法

(1) コンパイルリストの種類

コンパイラが出力するリストの種類を次に示します。

(a)情報リスト

プログラム情報やエラーの総数等のコンパイル時の情報を要約して 出力したものです。

- (b) 原始プログラムリスト
 コンパイル時に入力した原始プログラムのリストです。相互参照情報やコンパイル時にエラーが検出されたときのエラーメッセージなども出力されます。
- (c) エラーリスト
 コンパイルエラーのエラーレベルやエラーメッセージを出力したものです。

(2) リストの出力方法

(a) コンパイラオプションの指定
 コンパイルリストを出力するための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を指定しない場合、コンパイルリストは出力されません。

-SrcList, NoCopy

COPY文で複写した登録集原文の内容を原始プログラムリスト中に展開しません。

-SrcList, CopySup

SUPPRESS指定のあるCOPY文で複写した登録集原文の内容は原始 プログラムリスト中に展開しません。 SUPPRESS指定のない COPY文の場合はすべて展開します。

-SrcList, CopyAll

COPY文で複写した登録集原文の内容をすべて原始プログラムリ スト中に展開します。

-SrcList, OutputAll

COPY文の指定や条件翻訳、LISTING翻訳指令にかかわらず、強制的にすべてのソース原文をコンパイルリスト中に展開します。SUPPRESS指定のあるCOPY文も展開します。

115

-SrcList, XXXXX, NoFalsePath

条件翻訳結果の無効行はコンパイルリストに出力されません。 XXXXXには、CopyAll, CopySup, NoCopyのどれかを指定します。

NoCopy, CopySup, CopyAll, およびOutputAllサブオプションは同時に は指定できません。同時に指定した場合、最後に指定したオプショ ンが有効になります。

NoFalsePathサブオプションは、その他のオプション(OutputAll以 外)と同時に指定する必要があります。

全ての情報を表示したいときは、OutputAllオプションだけを指定 します。

(b) コンパイルリストの出力先

情報リストと原始プログラムリストは、コンパイルリストファイル (.lst)に出力されます。また、エラーリストは標準エラー出力 (stderr)、すなわちCOBOL2002開発マネージャのメッセージウィン ドウに表示されます。) コンパイルリストの出力手順および出力例は、次ページ以降で説明 します。 下記プログラムは「reidai1」プログラムのデータ定義部の「DATAO」と「YYMMDD」 を登録集原文として別ファイルに登録したものです。DATAO. cblとYYMMDD. cbl は、ソースファイルと同じフォルダに作成します。このプログラムをコンパイラオ プション「-SrcList, NoCopy」を指定してコンパイルしてみます。

출 COBOLIディタ for COBOL2002 - C:¥temp¥s	ample01¥reidai2¥r	eidai2.cbl		
ファイル(E) 編集(E) 検索(S) 構文(Y) オフ	^ション© ツール①	<u>ሳሪ</u> ኑን∰ ′	\/µフ°(<u>H</u>)	
D ≌ 🖬 ቆ X 🖻 🖻 Ø ∿	i] #V 		- 79 19 9	3 38
▶ ▶ ₩ ₩ ₩ 🗰 📾		🖳 E		
C¥temp¥sample01¥reidai2¥reidai2.cbl				
	····3·····	+4+++++++++++++++++++++++++++++++++++++	+5+++++++++++++++++++++++++++++++++++++	<u>6+++++</u>
000500 DATA DIVISION.				
000700 WORKING-STORAGE SE	CTION.			
000900 COPY DATAU.	ESS.			
001000				
	行: 1	π 5λ: 8	 插λ	
	1.1.0	p:/4.0		
COBOLIデタ for COBOL2002 - Citempts コーイル(に) (写集(に) たま(に) 様本(な) たっ コーイル(に) (写集(に) たま(に)) (様本(な) たっ)	ample01¥reidai1¥[ATA0.cbl	× 11 → 9/11)	
		9421593000		<u> </u>
] =0	l ei=a F		7 02
] 🛱 ។		
Ci¥temp¥sampleUI¥reidarI¥DATAU.cbl				
000100 01 DATA0.		*4****	·)···· ····	<u></u>
000200 02 DATA1 PIC X(10)	VALUE ALL	*'		
000400 02 DATA2 PIC X(20)	VALUE SPACE	' *'		
	行:			
	1.4.	,))	
STOBOLIデタ for COBOL2002 - Citempts	ample01¥reidai1¥\ ©->v(c) v=v(T)	∩YMMDD.cbl	λ.μ.⊐%(⊔)	
		9121 9 <u>w</u> y		B
		11 1 -		2 002
C:¥temp¥sample01¥reidai1¥YYMMDD.cbl		+4++++++++++++++++++++++++++++++++++++		
C:¥temp¥sample01¥reidai1¥YYMMDD.cbl	••••3••••• •••	+4 ++++ +++	•5••••	<u>×</u>
C¥temp¥sample01¥reidai1¥YYMMDD.cbl 000100 01 YYMMDD. 000200 02 年 PIC 9(2). 000300 02 日 PIC 9(2).	<u>+++3++++</u> ++++	+4++++++++++++++++++++++++++++++++++++	•5••••	6+++++
C¥temp¥sample01¥reidai1¥YYMMDD.cbl 000100 01 YYMMDD. 000200 02 年 PIC 9(2). 000300 02 月 PIC 9(2). 000400 02 日 PIC 9(2).	<u>+++3+++++</u>	+4+++++++++++++++++++++++++++++++++++++	•5••••	6+++++
C¥temp¥sample01¥reidai1¥YYMMDD.cbl 000100 01 YYMMDD. 000200 02 年 PIC 9(2). 000300 02 日 PIC 9(2). 000400 02 日 PIC 9(2).	+++3+++++	+4+++++++++++++++++++++++++++++++++++++	₽ ₽	
C¥temp¥sample01¥reidai1¥YYMMDD.cbl O00100 01 YYMMDD. 000200 02 年 PIC 9(2). 000300 02 月 PIC 9(2). 000400 02 日 PIC 9(2). レディ	<u>+++</u> 3+++++ ↑↑; 1	+4+++++ +2+++++++++++++++++++++++++++++	● □ Ⅲ	

[手順1] COBOL2002開発マネージャのメニューバーの「プロジェクト(P)」、「プロジェク トの設定(S)」の順にクリックします。すると「プロジェクト設定」画面が表 示されます。



[手順2] 「プロジェクト設定」画面の「リスト出力」タブの中から「-SrcList, NoCopy」オ プションをクリックし、「OK」ボタンをクリックします(口内にレ印が 付き、「設定内容」の欄に表示されます)。



118

[手順3] 開発マネージャの画面に戻ったら、「ビルド」を行います。

www.communications.com/www.communications.com/www.communications.com/www.communications.com/www.communications.com/www.communications.com/www.co	_ 🗆 ×
ר אָראָדעראָאָ(M) אין דערער די דיאריר אין	
3₂ ≥ 🖬 (3º № 4₂ №) 1⊥ 1№ 55 1№ 1№ 0€ 🖇 22 №	
🚰 sample01.hmf	
Sample01 ■ ■ reidai1 - reidai1.exe ■ ■ reidai2 - reidai2.exe ■ ■ reidai2.cbl ■ ■ 依存ファイル	

[手順4] ビルドが終了すると「依存ファイル」が表示されます。+ボタンを押して登録されている登録集原文を見てください。依存関係が自動的に確立されていることがわかります。また、workフォルダの下には、拡張子が. |stのファイルが生成されていますので開いて見てください。

■■ 開発マネージャ for COBOL2002 - sample01.hmf
ר \mathfrak{T}^{T} יבאלקאא(M) און דיראל(E) דיראל(P) אין דיראל(B) אין אין דיראל(M) און דיראל(H) און דיראל(P) אין ד
•≥ ≈ ■ ○ • • • • • • • ▲ ₩ ∞ * ₩ ‰ • <i>*</i> △ *
🗃 sample01.hmf
🔁 sample01
💼 🗇 reidai1 - reidai1.exe
📄 🗇 reidai2 – reidai2.exe
📗 🛱 🔁 Y-z774u
reidai2.cbl
□ □···································
┃
DATAO.CBL
Ⅲ ライブラリ C:¥temp¥sample01¥reidai2¥work¥reidai2.lib とオブジェクト C:¥temp
=== roidai2 のと"しい"加理が終了しました。 ===

[手順5] エクスプローラによりworkフォルダの下に実行可能ファイル等と共 にコンパイルリストファイル「reidai2.lst」が生成されます。 コンパイルリストファイルは、COBOLエディタ、メモ帳等で開いて 見ることができます。



コンパイルリストを以下に示します(メモ帳で開いています)。 「-SrcList, NoCopy」オプションなので、COPY文の内容は展開されません。

✓ reidai2.lst - 以无帳	l ×
ファイル(ビ) 編集(ビ) 書式(U) ヘルフ(ビ)	
	*:
* # : INITIALIZE又(はCORRESPONDINGで更新される下位項目	
* A: ALTERで参照	
* D:データ部又は環境部で参照	
▲ G:GUTUC参照 * P·PEREORMで参照	
* Q: IF/EVALUATE/PERFORMUNTIL/SEARCHWHEN/探索条件で参	82
* S:添字で参照	
* なし : その他	
***************************************	*:
A 000100 IDENTIFICATION DIVISION	
000200 PROGRAM-ID. reidai2.	
000300*	
000400 ENVIRONMENT DIVISION.	
UUUBUU DATA DIVISION. 000700 WORKING-STORACE SECTION	
000900 COPY YYMMDD SUPPRESS. されません。	
001000*	
001100 PROCEDURE DIVISION.	
001200* 001200 Main-Sec SECTION	
001400 PERFORM 27耳的九理	
001500 PERFORM 比較処理.	
001600 PERFORM 出力処理.	
001700 STOP RUN.	•
	• //

「-SrcList, CopySup」オプションを指定してコンパイルすると、コンパイルリストは次のようになります。



「-SrcList, CopyAll」オプションを指定してコンパイルすると、コンパイルリストは次のようになります。

6	reidai2.ls	st - メモ帳			
7	ァイル(E)	編集(<u>E</u>)	書式(<u>O</u>)	ヘルプ(円)	
		A	000100 000200 000300: 000400 000500: 000600	IDENTIFICATION DIVISION. PROGRAM-ID. reidai2. * ENVIRONMENT DIVISION. * DATA DIVISION.	•
	801 802 803 804 901 902 903 904	C1 C1 C1 C1 C1 C1 C1	000700 000800 000100 000200 000300 000400 000400 000100 000200 000300 000400	WORKING-STORAGESECTION.COPYDATA0.01DATA0.02DATA1 PIC X(10) VALUE ALL '*'.02DATA2 PIC X(20) VALUE SPACE.02DATA3 PIC X(10) VALUE ALL '*'.COPYYYMMDD SUPPRESS.01YYMMDD.02年02J02FIC 9(2).02J02FIC 9(2).02FIC 9(2).02FIC 9(2).02FIC 9(2).02FIC 9(2).02FIC 9(2).02FIC 9(2).02FIC 9(2).02FIC 9(2).02FIC 9(2).03FIC 9(2).04FIC 9(2).05FIC 9(2).06FIC 9(2).07FIC 9(2).	な 展
			001000 001100 001200 001300 001400 001500 001600 001600 001800 001800 001900 002000 002200	* PROCEDURE DIVISION. * Mein-Sec SECTION. PERFORM 初期処理. PERFORM 比較処理. PERFORM 出力処理. STOP RUN. * 初期処理 SECTION. ACCEPT YYMMDD FROM DATE. * 比較処理 SECTION.	

(g)オンラインマニュアルの使用方法

オンラインマニュアルにはCOBOL2002の全機能が掲載されています。

・キーワード検索等により、探したい項目を瞬時に検索することができます。

〈オンラインマニュルの構成〉

オンラインマニュアルは、次のマニュルから構成されています。

- ・「総合目次」
- ・「COBOL2002言語 標準仕様編」
- •「COBOL2002 言語 拡張仕様編」
- ・「COBOL2002操作ガイド」
- ・「COBOL2002 ユーザーズガイド」
- ・「COBOL2002 メッセージ」
- ·「COBOL2002操作入門」
- ・「索引順編成ファイル管理ISAM」

1. オンラインマニュアルの参照方法

[手順1]スタートボタンを押し、「プログラム(P)」の所にマウスポインタを移動し ます。すると起動できるプログラムの一覧が出てきます。



[手順2] プログラムの一覧の中から「COBOL2002」の所にマウスポインタを移動し ます。するとCOBOL2002の中の使用できるツール一覧が表示されます。



[手順3] COBOL2002の中の使用できるツールの中から、「COBOLマニュアル」の所にマ ウスポインタを移動すると、COBOLマニュアルの一覧が表示されます。 ここで参照したいマニュアルを選択します。



下記のようなCOBOL2002操作入門マニュアル画面が表示されます。 「キーワート(N)」、「検索(S)」機能により探したい項目、文字列を素早く検索すること ができます。次ページ以降は検索手順について説明します。

💕 COBOL20	02 操作入門						
1211 非表示	(中 戻る	→ 進む	6	日朝	₫- オプション@		
目次©) ⊡·◆ COE	キーワード(<u>N</u>) 30L2002 操作)	│ 検索⑤ <u> ◀</u> 入門		CO	BOL2002	操作入門	
					入門	書	
					3020-3	-D46	
					E	次	
			 <u>前書</u> き				
			(まじめ	<u>(2</u>			
			<u>1. CO</u> <u>1.1 C</u> <u>1.2 [</u>	BOLプロ COBOL20 開発マネ	1グラムを作成T 002に添付されて ージャを起動す	するための準備をしま こいる例題について るには	<u>د لیک</u>

2. キーワードによる検索方法

[手順1] 例えば「COBOLエディタ」に関する項目を探したい場合は、キーワード入 カ画面で「COBOLエディタ」というキーワードを入力するか、選択画面 から該当するキーワードを選択し、「表示」ボタンをクリックします。

울 COBOL2002 携	解入門						
- 1913 非表示	(中 戻る	> 進む			ご- オプション(①)		
目次(C) キー キーワードを入力し	ワード(<u>N</u>) てください(<u>W</u>); ;;	C	юво	L2002	操作入門	Î
COBOLエディタ COBOLエディタ COBOLプログラル COBOL原始プロ エラーメッセージ	< Эзц				入門書 3020-3-D4	≇ 46−01	
コーティンク コンパイラオプシ コンパイル コンパイルリスト ソーステキストウ テストデバッガ テストデバッグ	ョン いンドウ		前書考		日义	ب	
 デストナハック一 ビルド ファイル編成 プロジェクト プロジェクトの作り プロジェクトマス2 プロジェクトマス2 	覧 ワインド5 成 3 5の作成	_	<u>新国</u> 変更内容				
ボップアップメニ: リンケージ 開発マネージャ 拡張子 環境変数 警告メック。	- -	T	<u>(まじめに</u> <u>1. COBOL</u>	<u> プログラ</u>	ムを作成す	るための準備をしまし	<u>/4)</u>
	> 表示		<u>1.1_COBC</u> <u>1.2_開発</u> 型) <u>L2002</u> (こ マネージャ	<u>添付されてし</u> を起動する	<u>いる例題について</u> (<u>こ(す</u>	-

[手順2] すると、トピック画面が表示されますので、該当するトピックをクリ ックしてから、「表示」をクリックします。



該当する箇所が表示されます。



3. 検索語句入力による文字列検索方法

[手順1]例えば「COBOL2002」という文字列を探したい場合は、検索項目入力画面 で検索文字列「COBOL2002」を入力し、「検索開始」ボタンを押してください。



[手順2] すると、該当するトピック画面が表示されますので、トピックをクリッ クしてから、「表示」をクリックしてください。



以下のように指定した語句「COBOL2002」がリバース表示されます。「キーワート (N)」と 「検索(S)」の違いを次に示します。

- ・項目の内容を検索したい場合は「キーワート、(N)」を使用します。
- ・項目の文字列を検索したい場合は「検索(S)」を使用します。



(h) 登録集原文の指定方法

1. 登録集原文の指定方法

複数のCOBOLプログラムに共通の記述を別のソースファイルとしてお き、COPY文で取り込むことができます。登録集原文の概念図を下欄に 示します。このようにすると、コーディング量を削減でき、記述ミス もなくなるため、開発効率を高めることができます。ここでは、第1 章で実習した「reidai1.cbl」というソースファイルの「DATAO」と 「YYMMDD」を登録集原文としたものとして説明します。このソースファ イルを「reidai2.cbl」とします。

登録集原文はソースファイルとは別のフォルダにまとめて格納するの が一般的です。コンパイルする際には、「CBLLIB」というコンパイル時 環境変数で登録集原文が格納されているフォルダを指定します。



フォルダの構成の例を次に示します。 ここでは、「COPY」というフォルダに登録集原文を格納しています。



[手順1] reidai2というプロジェクトをこれまでと同様の手順で作成します。 ソースファイル「reidai2.cbl」を次に示します。 「reidai2.cbl」は、「reidai1.cbl」の「DATAO」と「YYMMDD」をCOPY文に変 更したものです。

000100 IDENTIFICATION DIVISION. 000200 PROGRAM-ID. reidai2. 000300* 000400 ENVIRONMENT DIVISION. 000500* 000600 DATA DIVISION. 000700 WORKING-STORAGE SECTION. 000800* 000900 COPY DATAO. 001000* 001100 COPY YYMMDD. 001200* 001300 PROCEDURE DIVISION. [手順2] プロジェクト「reidai2」を作成したら、作業中のプロジェクトを 「reidai2」に設定します。「reidai2」をポイントし右クリックし、プ ルダウンメニューの中から「作業中プロジェクトの設定(W)」を選択します。

with a set of the se	DL2002 - sample01.hmf	_ 🗆 🗵
- プロジェクトマスタ(<u>M</u>) - ファイル((E) プロジェクト(P) ビルト(B) ツール(T) ウィントウ(W) ヘルフ(H)	
🔹 🛥 🖬 🖉 ங	s 🕺 🔰 🔛 🔛 🐨 🏗 🗽 🍭 🛠 🛛 🍊 🏠	
] 🗊 🚍 ?		
🕝 sample01.hmf		
🔓 sample01		
📗 🗐 reidai1 - reida	hi1.exe	
😟 🕀 🗇 reidai2 - reidai2.e	選択中のプロジェクトのビルド(E)	
	作業中ブロジェクトの設定(W) 🤇 🔄 3	
	ソースファイルの追加(E) マン・シー	
	CUNK(B)	
②右クリック	プロジェクトの設定(S)	
	依存ファイルの設定(D)	
	美行巡	
	7泊ジェクト名の変更(ハ)	
	プロノパティ(<u>P</u>)	
11		

[手順3]次に環境変数「CBLLIB」で登録集原文のフォルダを指定します。 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)」を選択します。

変開発マネージャ for COBOL200	02 - sample01.hmf	_ 🗆 🗙
プロジェクトマスタ(<u>M</u>) ファイル(<u>F</u>)	プロラジェウト(Ⴒ) ビルド(B) ツール(T) ウィンドウ(W) ヘルプ(H)	
	 7 ロジェクトの作成(A) 7 ロジェクトの作成(A) プロジェクト名の変更(N) 	
□	7°ロジェクトのエクスホ°ートᢗᢓ フ°ロジェクトのインホ°ートℚ	
sample01 ⊕~⊡ reidai1 - reidai1.exe ⊕~⊡ reidai2 - reidai2.e	作業中プロジュクトの設定(W) プロジュクトの削除(E) プロジュクトの関係設定(B)	
	ソースファイルの追加(E) フロジェクトの設定(S)	
	依存ファイルの設定(D) 依存ファイルの仮登録(T)	
	<u>カハッレーシップの</u>	
	プロパティ(<u>P</u>)	
<u> </u>		



[手順4] 「プロジェクトの設定」画面が表示されるので、「環境変数」タブをクリック し、「CBLL IB」をチェックします。「環境変数」タブが見えないときは、 ▶ ボタンで表示位置を調整してください。



[手順5]「環境変数の設定」画面が表示されるので、「参照(B)」ボタンをクリックします。

環境変数の設定	×
設定内容(<u>S</u>):	
追加(A) 変更(C) 削除(D)	
	$\langle -$
OK \$6'/77.	1

[手順6]「フォルダの参照」画面が表示されるので、「COPY」フォルダを指定して 「OK」ボタンをクリックします。

フォルダの参照	? ×
このフォルダでよければ、OKを押して下さい。 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	
CK Arrybeol Englished Sampleol Englished Fridai2 Long Work OK Arrybeol CK CK	•
理境変数の設定	Y
我現象数(M設定 設定内容(<u>S</u>): c¥temp¥copy	
<u>追加(A)</u> 値(⊻): ● ● ○ ● ○ ● ● ○ ● ● ○ ● ● ○ ● ○ ● ○ ● ○	

「OK」ボタンをクリックします。

[手順7] 「プロジェクトの設定」画面に戻るので、「OK」ボタンをクリックします。

プロジェクト設定	×
7泊シェウト→覧(P): ・ sample01 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	リンク 規格 移行 リスト出力 その他 環境変数 ユー▲ ▶
	 □CBLCOP12002:10/N1737/>3/2を設定する □CBLERRMAX:10/N17477/32を設定する □CBLERRMAX:10/N14を打ち切るSLへルのIラーの数を設定す □CBLFREE:自由形式正書法で書かれたCOBOL原始710/54の □CBLFN2:固定形式正書法で書かれたCOBOL原始710/54の □CBLTAB:9710-トを空白に変換するときの値を設定する □CBLTAB:9710-トを空白に変換するときの値を設定する □CBLTAB:9710-トを空白に変換するときの値を設定する □CBLVALUE:データ項目の初期値を計算機固有文字集合中の □CBL_RDBSYS:HiRDBIこよる索引編成774ルの操作対象にな □CBLREP:リホットリ774ルを出力し、参照時に検索する734ルタを指定 □CBLSYSREP:リホットリ774ルの参照時に検索する734ルタを指定 ■登録集名:登録集名を検索する734ルタ(ディレクトリ)を指定する
] 設定内容	
CBLFIX=txt CBLLIB=c:¥temp¥copy	 ▼
	OK ++>>te/

[手順8] ビルドします。



[手順9] プロジェクト「reidai2」を全て表示すると、依存ファイルとして登録 集原文「DATAO.cbl」と「YYMMDD.cbl」が登録されているのがわかります。 ビルドすると、開発マネージャがCOPY文を認識して自動的に依存関係 を構築するのです。

www.geweinet.com/www.com	
ר איז	
🚰 sample01.hmf	
🔁 sample01	
📑 🗄 🕣 reidai1 - reidai1.exe	
□	



1. プロジェクトの設定方法

ここでは、一つのメインプログラムと複数のサブプログラムから構成 される場合のプロジェクトの設定方法について説明します。 (プロジェクト「reidai3」を例題として説明します。)

第1章で実習した「reidai1」のプログラムの「初期処理」、「比較処理」、 「出力処理」を各サブプログラムとして、メインプログラムから呼ばれ るようにした例で設定方法を示します。 各サブプログラムの名称は次のとおりです。

■メインプログラム : reidai3.cbl

PROCEDURE	: DIVISION.		
Mein-Sec	SECTION.		
CALL	'REIDAI3A'	USING	YYMMDD.
CALL	'REIDAI3B'	USING	DATAO YYMMDD.
CALL	'REIDAI3C'	USING	DATAO.
STOP	RUN.		

■サブプログラム「比較処理」: reidai3b.cbl

IDENTIFICATION DIVISION. PROGRAM-ID. reidai3b. ENVIRONMENT DIVISION. DATA DIVISION. LINKAGE SECTION. 01 DATAO. 02 DATA1 PIC X(10). 02 DATA2 PIC X(20). 02 DATA3 PIC X(10). 01 YYMMDD. 02 年 PIC 9(2). 02 月 PIC 9(2). 02 ⊟ PIC 9(2). PROCEDURE DIVISION USING DATAO YYMMDD. 比較処理 SECTION. IF 月 = 9 THEN MOVE 'September!!' TO DATA2 ELSE MOVE 'Not September!!' TO DATA2 END-IF. EXIT PROGRAM.

■サブプログラム「初期処理」: reidai3a.cbl

```
IDENTIFICATION DIVISION.

PROGRAM-ID. reidai3a.

ENVIRONMENT DIVISION.

DATA DIVISION.

LINKAGE SECTION.

01 YYMMDD.

02 年 PIC 9(2).

02 月 PIC 9(2).

02 日 PIC 9(2).

PROCEDURE DIVISION USING YYMMDD.

初期処理 SECTION.

ACCEPT YYMMDD FROM DATE.

EXIT PROGRAM.
```

■サブプログラム「出力処理」: reidai3c.cbl

IDENTIFICATION DIVISION. PROGRAM-ID. reidai3a. ENVIRONMENT DIVISION. DATA DIVISION. LINKAGE SECTION. 01 DATAO. 02 DATA1 PIC X(10). 02 DATA2 PIC X(20). 02 DATA3 PIC X(20). 02 DATA3 PIC X(10). PROCEDURE DIVISION USING DATAO. 出力処理 SECTION. DISPLAY DATAO. EXIT PROGRAM.

メインプログラムのプロジェクト「reidai3」は、これまで示した手順 で作成し、各サブプログラムはあらかじめ「reidai3.cbl」と同じフォ ルダに格納してあるものとして手順を示します。

141

[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 ウンメニューの中の「ソースファイルの追加(F)」をクリックします。 すると、ファイルの追加画面が表示されます。

	$\sqrt{1}$	
	002 - sa	
プロジェウトマスタ(<u>M</u>) ファイル(<u>F</u>)	「プロジェクト(Ⴒ) ビルド(B) ツール(T)	ウィンドウ W2 ヘルフ [*] (<u>H</u>)
😼 🛋 📕 🗷 🍬	7 コジェクトの作成(<u>A</u>) 7 コジェクト名の変更(<u>N</u>)	ix a 🛠 🛛 🥔 🦢
刘 🗔 🚍 🥐	プ᠋᠋᠋᠋᠊ジュፇトのエクスポート⊗ プ᠋ジュፇトのインポートℚ	
sample01 ⊕@ reidai1 - reidai1.exe ⊕@ reidai2 - reidai2.exe	作業中プロジェクトの設定(\\/) プロジェクトの削除(E) プロジェクトの関係設定(R)	
i i i i i i i i i i i i i i i i i i i	ソースファイルの追加(<u>F</u>) く	
□□□ · · ···□] reidai3.cbl	プロジェクトの設定(S) 依存ファイルの設定(D) 依存ファイルの仮登録(T)	
プロジェクト(こファイルを追加する	カバレージ(<u>O</u>)	
	フ°ロノı°ティ(<u>P</u>)	

[手順2] ここで追加するファイルを選択し、開くボタンを押下します。 複数のサブプログラムを追加できます。

ファイルの追加		? ×
ファイルの場所型:	🔁 reidai3 📃 🖛 🗈 💣 🎟	
i work i reidai3.cbl i reidai3a.cbl i reidai3b.cbl i reidai3c.cbl i reidai3c.cbl		
ファイル名(N):	「"reidai3c.cbl" "reidai3b.cbl" "reidai3a.cbl" 開く(Q	
ファイルの種類(工):	COBOLソース個定形式)(*.cbl;*.cob;*.ocb) 💌 キャンセ	JL I
追加先プロジェクト(<u>P</u>):	reidai3	

[手順3] reidai3のプロジェクトに3つのサブプログラムが追加されました。 この後、ビルドを行います。ビルドすると4つのプログラムが結合されて一つの実行可能ファイル(.exe)になります。エラーがなければ続いて実行してください。

www.commercedimensional and the sampleO1.hmf	
\mathcal{T}^{T}	
😳 🛎 🖬 ૐ № 💩 🖗 L L 🛍 🛱 L k 🗞 @ 🆇 🍊 🍅	
🚰 sample01.hmf	
Sample01 ■ ■ reidai1 - reidai1.exe ■ ■ reidai2 - reidai2.exe ■ ■ reidai3 - reidai3.exe ■ ■ reidai3.cbl ■ reidai3a.cbl ■ reidai3b.cbl ■ reidai3c.cbl ■ reidai3c.cbl ■ reidai3c.cbl	
ντ²ι ΝυΜ	//

[注意事項]

- ①メインプログラムの終了は「STOP RUN」文、サブプログラムの終了は 「EXIT PROGRAM」文を使用します。
- ②メイン側のプログラムからサブプログラムを呼び出す「CALL」文では、サブプログラム名をアポストロフィで囲んで指定します。

[例] CALL 'REIDAI3A' USING DATAO YYMMDD.

プログラム名は基本的に大文字を使用します。これは、サブプログラムの 「PROGRAM-ID」段落で定義したプログラム名が大文字と解釈されるからです。

[例] PROGRAM-ID. reidai3a. <-- プログラム名は「REIDAI3A」と解釈

なお、「PROGRAM-ID」段落のプログラム名をアポストロフィで囲んで指定する と、指定したとおりに解釈されます。

[例] PROGRAM-ID. 'reidai3a'. <-- プログラム名は「reidai3a」と解釈 ③サブプログラムで受け取るデータは「LINKAGE SECTION」として定義します。 「LINKAGE SECTION」では、「VALUE」句は使用できません。

(j) 索引ファイルを新規に 作成する方法

144
索引ファイルを新規に作成する方法について説明します。

「OPEN OUTPUT」で開いてWRITE文でレコードを書き出すと、通常は新規作成になり ますが、索引ファイルの場合は追加モードでレコードが書き出されます。すなわ ち、一度索引ファイルを作成した後、同じキーで索引ファイルを作成し直そうと しても、「重複キー」エラーになってしまいます。

索引ファイルの新規作成をするときは、実行時環境変数「CBLISAMDL=YES」を指定 します。これにより、既存の索引ファイルが削除され、新規作成することができ 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「実行支援(L)」をクリックします。すると、実行支援 画面が表示されます。

🗾 「「「」 Sample01 → Weight for COBOL2002 – sample01	.hmf	
プロジェクトマスタ(<u>M</u>) ファイル(<u>F</u>) プロジェクト(<u>P</u>)	ビルト*(B) ツール(D) ウィント*ウ(W) ヘルフ*(H)	
	ビルド(B) Ctrl+B クイックビルド(Q) コンパイル(O) リビルド(B)	
🚰 sample01.hmf		, <u> </u>
sample01 È…,, reidai1 - reidai1.exe È…,, reidai2 - reidai2.exe	 ビルドの中止(S) 依存関係更新(<u>A</u>)	_
😟 🗇 reidai3 - reidai3.exe	デンドッガ(<u>D</u>)	
□	実行(<u>E</u>) Ctrl+E 実行支援(<u>L</u>) ◀	2
COBOL2002 実行支援を起動する		

[手順2] 実行時環境変数は、「ユーザ設定」タブで設定します。 変数名に「CBLISAMDL」、値に「YES」と入力して「登録」ボタンを押すと、 「環境変数登録リスト(L)」に登録されます。閉じる(≥)ボタンを押すと 「保存しますか?」と聞いてきますので、必ず「はい」をクリックして ください。

素実行支援 for COBOL2002 - reidai4.CBR (更新)	
ファイル(E) 設定(E) 実行(R) ヘルフ [*] (H)	
環境変数登録リスト(L)	
	`
- 一般 少量データ ファイル 画面 画面(MAP) 整列併合 拡張機能 - デバッグ オージン・クト指向 COBOL GUIT2%ay イベントログ ユーザ設定く	ノ コ
	_
変数名(N) CBLISAMDL 、 / /	
	2
値(V) YES 登録(A)	
印刷定義(P) 参照(B)	
	1)
■ 実行支援 for COBOL2002 - reidai4.CBR (更新)	
□	
CBLISAMDL=YES 削除(D)	
実行支援 for COBOL2002	1
変数名(N) (1()え(N) をおいけれし	
值(V) (2)	
登錄(<u>A</u>)	
印刷定義(2) 参照(8)	





1. 既存のプロジェクトマスタファイルの開き方

ここでは、既に作成済みのプロジェクトマスタファイルの開き方について説明します。

[手順1] 開発マネージャのメニューバーの「プロジェクトマスタ(M)」、「開く(0)」をク リックします。すると「ファイルを開く」画面が表示されます。

wg開発マネ ∳ for COBOL2002		×
プロジェクトマスタ(M) ツール(T) ヘルプ(H)		
新規作成(N)	No 🕾 Nr No 🔍 🏈 🦾	
テウォルトオフションの設定(型) (2)		
1 C:¥temp¥sample01¥sample01.hmf		
開発マネージャの終了 🖄		
νīτκ		//

[手順2] ここで開きたいプロジェクトマスタファイルをクリックし、「開く (0)」ボタンを押します。すると選択したプロジェクトマスタファイ ルが開きます。下記の例では「SAMPLE01」というプロジェクトマスタ ファイルを開きます。

ファイルを開く			? ×
ファイルの場所型:	🔄 sample01	- 🗧 🖆 📰	
ireidai1 reidai2 reidai3 reidai4 reidai4			
 ファイル名(N): ファイルの種類(II):	ጋግንንድንኮマスタファイル (*.hmf)	開() マ キャン	2 2 2

[手順3] プロジェクトマスタファイル「sample01」が開かれました。

☞ 開発マネージゃ for COBOL2002 - sample01.hmf	
ר איז	
43 ⊯ 🖬 65 № 43 № 🗎 🛍 🛱 🗽 🖄 🦀	
🕝 sample01.hmf	
sample01 reidai1 - reidai1.exe reidai2 - reidai2.exe reidai3 - reidai3.exe reidai4 - reidai4.exe	
レディ	NUM //

[補足] 手順1で、「プ ロジ エクトマスタ(M)」をクリックしたとき、最近使用したプロジェクトマスタファイルが表示されます。開きたいファイルが表示されているときは、そのファイルをクリックしてください。 あるいは、エクスプローラで直接プロジェクトマスタファイル(.hmf)をダブルクリックして開くこともできます。

www.commercedeline.	
- プロジェウトマスタ(M) ツール(T) ヘルプ(H)	
新規作成(N) 🐘 🕄 🐘 🖏 🖉 🐴 🗞	
テウォルトオフ ションの設定(D)	
1 C:¥temp¥sample01¥sample01.hmf	
開発マネージャの終了公グ	
ν̄τ̄'i	

《他社所有名称に対する表示》

- Btrieveは、米国Pervasive Software Inc.の米国およびその他の国における商標または登録 商標です。
- ・ODBCは、米国Microsoft Corp.が提唱するデータベースアクセス機構です。
- ・OLEは, 米国Microsoft Corp.が開発したソフトウェア名称です。OLEは, Object Linking and Embeddingの略です。
- ・Windowsは、米国およびその他の国における米国Microsoft Corp.の登録商標です。
- Windows Vistaは、米国Microsoft Corporationの米国及びその他の国における登録商標です。
- ・その他記載の会社名・製品名は、それぞれの会社の商標もしくは登録商標です。