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。

■ご開発マネージャ 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 fe 11r 11/2 (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	±ル	

[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ያወበቱ成	×
ንግንንェንኮマスጵ名(<u>M</u>):	sample01
プロジェクトマスタフォルダ名(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	leor.nmi			
ን°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	ウィントウ₩	ヘルフ°(<u>H</u>)
] 🔁 🖻 🔚 (🦉 ங 😼 😼		😳 🔛	× 🔍 🌮	🎒 🦢
	<u> </u>				
Sample(11 hmf					_
sample01					_
🖻 🗊 reidai1 -	reidai1.exe				
□ □ □ □ ソースファ	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	ole01.hmf (P) ビルド(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 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、シュージャーク	r COBOL2002 - samp ファイル(E) フロジェクト	ole01.hmf (P) ביאורי(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.OBL] オフ≫∍ン©) ツール(T)		. <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			-	
b7*i	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(E) ビ	f ルド(B) ツール(T) ウィ	₩ \IL7°(<u>H</u>)	<u></u>	1
	11 HQ 🛱 HR 🗽	ک 🌢 🛠		
				_
🚰 sample01.hmf				
in the second s	実行			
■ reidai1.cbl	実行可能ファイル名()	D: le01¥reidai1¥work	¥reidai1.exe▼ 参照(B)	
1.5%	作業フォルダ(<u>F</u>):		▼ 参照(R) ▼]
0/1	引数(<u>A</u>):			1.
			K キャンセル	
			2	
[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、 ■ ボタンをクリック	
実行が終了しました。			しまり。	

[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
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>	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 ታንኮወ/ፑ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。

		-			
ገንዥታՒመ追加:COBOLንኸንዥታՒ					×
最終生成物の種類を選択してください。					
● 実行形式ブログラム(E)					
 ダイナミックリンクライフ・ラリ(D) 					
○ 標準ライブラソ(L)					
	(百万(日)	 	5	キャントクリレ	-
		7,7 (1)	<u> </u>	47200	_
		٢En	iter]+	ーを押す。	
-					
ジェクトの追加:COBOLプロジェクト					2
プロシュークトの種類を選択してください。					
, H, 1,11, <u>F</u> WCM3, C C CCC C					
System指定メインプログラム					
V3指定メインブロクラム メインプログラムなし メリアカ ナーデポロトコナ・・パポロゲニー					
IULEアリトオフラロセスサーハラロクラム ICGIメインプログラム	A				
G01711774					
T				▼ ►	
					-
	< 戻る(<u>B</u>)	<u>次へ(N)</u>	>	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) ウィントウ(W) ヘルフ°(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。

w開発マネーシャ for COBOL20)2 - 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*(<u>B</u>) ツ	∽ル(፹) ሳ∂ንՒንጎየው) ∿ルንግ	<u>(H)</u>
🔁 🛥 🖬 🖉 🖪 😣	😥 🔰 🔛 🖗	3 比 🗽 🍳 🌋 🛛 🥔	۵
] 🖬 🔜 ?			
国 かセージウイントウ		_ 🗆 ×	
🕝 sample01.hmf			
sample01	il exe		
□ □ <u>□</u> <u>Y</u> -zファイル	①右クリック		
□□□ reidai1.cb □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□			
🗄 🗇 🖬 reidai2 - reidai2.e	選択中のプロジェクトのビルト	°E	
	「作業中プロジェクトの設定\」 ソースファイルの追加(F)…		
	ビルド(<u>B</u>)		
	プロジェクトの設定(S)		
作業中2年2月3月4月間完ま行う	14/17/17/060.20℃0/… 実行⊗		
TF# 17 47 47 0600 2017			
	プロパティ(<u>P</u>)		
		-	
■×開発スネージャ for COBOL2002	- sample01 hmf		
7 [°] ፲ジェクトマスタ(<u>M</u>) ファイル(<u>F</u>) フ	ʹロシシェクト(Ⴒ) ビルト(Β) ツー	⊮፹ ሳለን⊦ን₩ ∿⊮ን℃⊞	
😼 🛋 🔚 🗷 😼 ।	2 1 1 1 1 1 1	比 🗽 🍭 🖇 🗌 🥔 🎝	.
		t	
sample01.hmf			×
🔓 sample01			
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□	•		
☐ ☐ reidai1.cbl	^r rei	dai2」が	
	.exe < 太字	になる。	
T	7		
レディ		NUI	M ///

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。

輕閒発マネーシシャ for COBOL2002 - sample01.hmf	_ D ×
2^{1} $y = y = y = 0$ $y = y = 0$ $y = y = 0$ $y = $	
🐟 🛩 🖬 🗷 ங 🐼 😥 🔄 🏦 🐘 🖏 🍭 🖇 🌙 🆢	
国 メッセーシウィントウ	
🚰 sample01.hmf 📃 🗆 🗙	1
sample01 reidail - reidail.exe y-スファイル reidail.cbl 依存ファイル reidai2 - reidai2.exe reidai2 - reidai2.exe Image: Control of the second secon	
ντ ² γ	

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)			(E)
◆ 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 //.