(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。

➡ ■ 開発マネージャ for COBO	L2002 – sample01.hmf	
プロジェクトマスタ(<u>M</u>) ファイル(E) プロジェウト(P) ビルド(B) ツール(T) ウィンドウ(W)	ヘルフ ^{°(} <u>H</u>)
🔹 🛋 🖉 🖪	😼 🗞 📃 🔛 🏙 🖏 🍭 🛠	🗳 🆢
] 🗊 🖃 📍		
🚰 sample01.hmf		
🔓 sampleO1		
🗄 🖅 🗾 reidai1 - reida	i1.exe	
😟 🗐 reidai2 - reidai2.(選択中のプロジェクトのビルド(E)	
	作業中プロジェクトの設定(20) 🤸 🗔 (3)	
	ソースファイルの追加(<u>F</u>) マリン	
(2)右クリック	7℃ロジェクトの設定(S)	
	依存7ァイルの設定(U) 実行へへ	
	プロジェクト名の変更(Ŋ)	
	プロノパティ(Ⴒ)	
<u> </u>		

[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。

🗾 🗾 🗾 🕺 🗾 🕺 🗾 🕺 🗾 📝 🖉	02 - sample01.hmf	
プロジェクトマスタ(<u>M</u>) ファイル(<u>F</u>)	7° Dŷʿェクト(P) Ľĺルト ⁽ (B) ツール(T) ウィントゥウ(W) ヘルフ ^o (H)	
3 🚔 🖬 🗷 🍕	7℃ジェクトの作成(A) 7℃ジェクト名の変更(N)	
□	プ᠋᠋ᠴジュウトのエクスポートᇲ プ᠋ュジュウトのインポートℚ	_ [] X
G sample01 ⊕~⊡ reidai1 - reidai1.exe	作業中7泊ジュクトの設定(W) プ泊ジュクトの削除(E)	
⊞ ∏ reidai2 - reidai2.e	プロジェクトの開催系設定(PD ソースファイルの追加(E)	
	7 [°] 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>)	
	カパレーシ [*] (<u>O</u>)	
	プロパティ(<u>P</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(S):	
追加(A) 変更(C) 削除(D)	
値♡:	
参照(<u>B</u>)	
OK ++v)セル	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。	•
⊡ reidai2	
work	•
OK キャンセル	
環境変数の設定	×
設定内容(S):	
C:¥temp¥copy	
追加(A) 変更(C) 削除(D)	
値(⊻): 参照(<u>B</u>)	
OK ++>tell	

「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泊ジェクト一覧(P): 「「a sample01	リンク 規格 移行 リスト出力 その他 環境変数 ユ-▲ ▶
tereidai1 - reidai1.exe tereidai2 - reidai2.exe	 ✓ CBLLIB: 登録集原文を格納するフォルダ(ディレクトリ)を設定する CBLCOPT2002: コンパイラオブションを設定する CBLERRMAX: コンパイルを打ち切るSレベルのエラーの数を設定す CBLFREE: 自由形式正書法で書かれたCOBOL原始プログラム ✓ CBLFIX: 固定形式正書法で書かれたCOBOL原始プログラムの CBLTAB: タブコードを空白に変換するときの値を設定する CBLTAB: タブコード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 CBLRDBSYS: HiRDBIこよる索引編成ファイルの操作対象にな CBLREP: リボジトリファイルを出力し、参照時に検索するフォルダを指定 CBLSYSREP: リボジトリファイルの参照時に検索するフォルダを指定 登録集名: 登録集名を検索するフォルダ(ディレクトリ)を指定する
CBLFIX=txt CBLLIB=c:¥temp¥copy	 ▼
	OK ++>tell

[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.開発マネージゃ for COBOL2002 - sample01.hmf	
ጋ°ロジェクトマスタ(M) ファイル(E) フ°ロジェクト(P) ヒ`ルトヾ(B) ツール(T) ウィント゛ウ(W) ヘルフ°(H)	
32 ⊯ 3 10 10 10 10 10 10 10 10 10 10 10 10 10	
🚰 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	E 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。

10

[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 - sa01.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(T)	ウィントウ W2 ヘルフ (H)
🗞 🛋 📕 😃 🐱	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 🛠 🛛 🥔 🦢
🛛 🖬 🥐	プ᠋᠋᠋₽ジュታトのエクスポート⊗ プ᠋ジュታ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(<u>R</u>)	
⊡ reidai3 - reidai3. ⊡ Y77⊋∕II.	ソースファイルの追加(<u>F</u>) <	
□□□ / ////////////□□ 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	カパレージ(@)	

[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 📃 🖛 🗈 💣 🎟	
─ work meidai3.cbl reidai3a.cbl reidai3b.cbl reidai3b.cbl meidai3c.cbl		
ファイル名(N):	「"reidai3c.cbl" "reidai3b.cbl" "reidai3a.cbl" 開く()	
ファイルの種類(工):	COBOLソース個定形式)(*.cbl;*.cob;*.ocb) 💌 キャンセ	11
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.gwww.com/com/com/com/com/com/com/com/com/com/	<u>- 🗆 ×</u>
ר אָראָדאָלאָדאָאָשאָראָאָשאָראָאָשאָראָאָשאָראָראָראָראָראָראָראָראָראָראָראָאָשאָראָראָאָשאָראָאָראָאָשאָראָראָראָאָשאָראָאָראָאָאַראָאָראָאָ	
🐟 🛩 🖬 🗷 № 🕸 👌 🔛 🔟 🔛 😚 🖿 🐘 🗞 ֎ 🌋 🌙 🆢	
🚰 sample01.hmf	
Sample01 ■ ■ reidai1 - reidai1.exe ■ ■ reidai2 - reidai2.exe ■ ■ reidai3 - reidai3.exe ■ ■ reidai3.cbl ■ reidai3a.cbl ■ reidai3b.cbl ■ reidai3c.cbl ■ reidai3c.cbl ■ reidai3c.cbl	
ν ⁷ '	

[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

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。

🗾 「「「」 「」 「」 「」 「」 「」 「」 「」 「」 「」 「」 」 「」 」 「」 」 「」 」 「」 」 「」 」 「」 」 「」 」 「」 」 」 」 」 」 」 」 」 」 」 」 」 」 」 」 」 」 」 」	.hmf	
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(13) ツール(17) ウィンドウ(14) ヘルフペ(14)	
	ビルド(B) Ctrl+B ウイックビルド(Q) コンパイル(Q) リビルド(B)	
🕝 sample01.hmf		, <u> </u>
sample01 ⊕…@ reidai1 - reidai1.exe ⊕…@ reidai2 - reidai2.exe	ビルドの中止(S) 依存関係更新(A)	
😟 🗇 reidai3 - reidai3.exe	デバッガ(<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(E) ヘルプ(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(XMAP) │ 整列併合 │ 拡張機能 デバッグ │ オブジェクト指向 │ COBOL GUIオプション │ イベントログ │ ユーザ設定 	
変数名(N) CBLISAMDL てて	
	3
	(1)
▶ 実行支援 for COBOL2002 - reidai4.CBR (更新)	×
環境変数登録リスト(L)	-
実行支援 for COBOL2002	×
ー般 少量データ ファイル 拡張機能 デバッグ オブミ ♪ reidai4.CBRは更新されています。保存し	ますか?
変数名(N) (はいび) いいえ(N) キャント	211
値♥2	
	_

15



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(<u>T</u>):	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。

www.開発マネージゃ for COBOL2002 - sample01.hmf	
ר אָראָלאַראָאָשאָראָאָשאָראָאָשאָראָאָשאָראָאָשאָראָאָאָראָאָראָאָשאָראָאָאָראָאָשאָראָאָאָשאָראָאָאָאַראָאָראָאָ	
🐟 🖆 🖬 🗳 ங 🚳 🗞 L L L 🖄 🖏 🐘 🗞 🍭 烯 🥔 🍃	
🕝 sample01.hmf	
sample01 reidai1 - reidai1.exe reidai2 - reidai2.exe reidai3 - reidai3.exe reidai4 - reidai4.exe	
לד ^י י	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.com/second	
フプロジェクトマスタ(<u>M</u>) ツール(<u>T</u>) ヘルフ [°] (<u>H</u>)	
新規作成(N) 🐘 🖓 🐘 🕄 🖉 🐴 🗞	
テウォルトオフジョンの設定(D)	
1_C:¥temp¥sample01¥sample01.hmf	
開発マネージャの終了 図 クリック	
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。